SIEMENS

STAR-CCM+® Documentation

Version 12.06

Product Overview

User Guide

Theory

Tutorials

STAR-View+

STAR-CAT5

STAR-NX

STAR-Creo

STAR-Inventor

© 2017 Siemens PLM Software

SIEMENS
Ingenuity for life

Siemens PLM Software

STAR-CCM+
Product Overview

Version
12.06

www.siemens.com/mdx

© 2017 Siemens Product Lifecycle Management Software Inc.

Any and all Siemens Product Lifecycle Management Software Inc. brand, product, service and feature names, logos and slogans are registered trademarks or trademarks of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries. All other brand, product, service and feature names or trademarks are the property of their respective owners.

Product Overview

STAR-CCM+ is a Computational Aided Engineering (CAE) solution for solving multidisciplinary problems in both fluid and solid continuum mechanics, within a single integrated user interface.

STAR-CCM+ Simulation Environment

The STAR-CCM+ simulation environment offers all stages required for carrying out engineering analyses, including:

- Import and creation of geometries
- Mesh generation
- Solution of the governing equations
- Analysis of the results
- Automation of the simulation workflows for design exploration studies
- Connection to other CAE software for co-simulation analysis

Contents:

Geometry
Mesh
Physics
Solvers
Data Analysis
Design Exploration
Interaction with CAE Software
Application-Specific Tools
Usability and Productivity

Geometry

STAR-CCM+ imports geometries from, and integrates within, leading CAD and PLM systems. STAR-CCM+ also offers a built-in capability for modifying and creating CAD geometries directly.

Geometry Import

STAR-CCM+ reads geometric data as follows:

- Native formats: CATIA V4 and V5[1], Siemens NX[1], SolidWorks[1], Autodesk Inventor[1], PTC Creo[1], Rhino 3D[1], and Solid Edge[1]
- Neutral formats: IGES, STEP, Parasolid, IDF, and JTOpen[1]
- Triangulated formats: STL, CATIA graphics .cgr[1], pro-STAR .dbs/.inp, Nastran and Patran shells .nas & .pat, Medina .bif[1], FELISA .fro, VDA-FS[1]
- PLM formats: 3DXML[1], PLMXML
- Automatic error check and healing

CAD Clients

CAD Clients are available as STAR-CCM+ add-ons embedded within leading CAD systems. CAD Clients provide bi-directional and associative geometry data transfer. The supported CAD systems are: CATIA V5, Siemens NX, PTC Creo, and Autodesk Inventor.

Geometry Modification and Creation

3D-CAD is a parametric feature-based modeler that is built upon the Parasolid kernel. 3D-CAD works on both Linux and Windows, and is fully integrated within the client-server architecture of STAR-CCM+. The extensive suite of CAD functions include:

- All relevant operations for creating parametric feature-based solid and sheet bodies
  - Sketching, extrusions, revolving, lofting, sweeping, and imprinting
- Body operations: patterning, transforming, Boolean actions, and imprinting
- Import and edit 2D and 3D curves
- Defeaturing and sewing tools for imported geometries
- Automatic blade slicing tool for rotating machine geometries
- Automatic removal of small bodies or surfaces irrelevant to simulation
- Flow domain extraction

[1] Requires add-on module

Mesh

STAR-CCM+ provides a complete set of capabilities for both surface and volume meshing operations.

Mesh Topologies

- Surface: triangles, quadrilaterals, polygons
- Volume: tetrahedra, hexahedra, prisms, arbitrary polyhedra

Mesh Framework

The STAR-CCM+ mesh framework provides a flexible environment and repeatable processes. The general features of the framework are:

- Sequencing and re-ordering of mesh operations
- Serial, concurrent per part, and parallel mesh generation
- Local mesh modification wapped and remeshed surfaces
- Conformal polyhedral and tetrahedral meshes across multi-part assemblies

Surface Meshing

The surface meshing tools in STAR-CCM+ are designed to provide high quality triangulations on arbitrarily complex and dirty geometries. Available surface meshing tools include:

- Surface preparation: to manually repair and prepare surfaces ready for meshing
- Surface wrapper: to provide a closed, manifold, non-intersecting surface when starting from poor quality CAD data
- Surface remesher: to provide a high quality starting surface for all volume mesh types
- 2D mesher: to create triangular and polygonal meshes on 2D geometries

Volume Meshing

- **Core Meshers**
  - Trimmer
  - Polyhedral, tetrahedral
  - Prism layer mesher
- **Specialized Meshers**
  - Thin part mesher
  - Generalized cylinder mesher
  - Extruder mesher
  - Offset mesher
- **Directed mesher for swept CAD geometries**
  - Manual or automatic creation of patches for sweeping
  - Control of layer count and distribution

Physics

STAR-CCM+ is a multiphysics platform that solves systems of equations derived from the fundamental laws of physics.

Fluid Mechanics

STAR-CCM+ core capabilities for modeling fluid flow and energy are:

Flow characteristics

- Inviscid, laminar, or turbulent
- Newtonian and Non-Newtonian viscosities
- Incompressible and compressible
- Multi-component mixtures
- Multiphase mixtures
- Porous resistance
- Gravitational acceleration
- Passive scalars

Temporal discretization

- Steady
- Unsteady (explicit and implicit)

Equation of state

- Constant density
- Ideal gas law
- Real gas laws (Peng-Robinson, Redlich-Kwong, Standard and Modified Soave-Redlich-Kwong, Van der Waals, Equilibrium Air, IAPWS-IF97)
- Polynomial density
- User-defined

Materials

STAR-CCM+ comes with a database of common materials in categories of solid, liquid, gas, and electrochemical species.

- STAR-CCM+ installation comes with standard and extended databases
- Add or modify user-defined materials and material properties within the user interface
- XML format allows direct creation of user-defined databases

Heat Transfer

STAR-CCM+ can simulate all modes of heat transfer in both fluid and solid materials.

- Conduction

- Convection
- Radiation (Surface to surface (S2S) with view factors for both gray and multiband, solar loads, discrete ordinate model (DOM) for participating media)

Turbulence

STAR-CCM+ provides a choice of turbulence models for modeling different flow behaviors.

Reynolds-Averaged Navier Stokes (RANS) models

- Spalart-Allmaras (standard and high-Reynolds number)
- K-Epsilon (standard, realizable, two-layer, low-Reynolds number, elliptic blending, lag elliptic blending, Abe-Kondoh-Nagano, and V2F)
- K-Omega (standard and SST)

Reynolds Stress Transport models

- Linear pressure strain (standard and two-layer)
- Quadratic pressure strain
- Elliptic blending

Large Eddy Simulation

- Subgrid scale models (Smagorinsky, dynamic Smagorinsky, and WALE)
- Synthetic turbulence initial and boundary conditions

Detached Eddy Simulation (delayed and improved delayed)

- SST K-Omega
- Spalart-Allmaras
- Elliptic blending

Laminar-turbulent transition

- User-defined turbulence suppression
- Gamma ReTheta model
- Gamma model

Associated Models

- Temperature heat flux model
- Wall treatments (high-y+, low-y+, and all-y+)

Multiphase Flow (Eulerian Description)

STAR-CCM+ provides several multiphase solvers that account for distinct fluid phases within a simulation.

Eulerian Multiphase (EMP) Segregated Flow

- Distinct conservation equations are solved for each phase
- Supported phase types are gas, liquid, and particle (granular)
- Supports interphase energy, momentum, and mass transfer

- Multiple flow regime model for capturing differing regimes in the same simulation
  - Dispersed – bubbles/droplets of one phase dispersed in another
  - Large interface – resolved free surfaces
  - Symmetric/mixed – no clear distinction between continuous and dispersed phases
- Granular phase models: granular pressure, and granular temperature transport
- Population balance models: S-Gamma or A-MUSIG
- Single component crystal growth
- Wall and bulk boiling
- Phase-level turbulence modeling
- Phase chemical reactions

Volume of Fluid (VOF)

- Solves single set of conservation equations plus one for volume fraction
- Tracks the free surface between immiscible fluids
- High-resolution interface capturing (HRIC) scheme
- Phase-change processes: evaporation and condensation, boiling, gas dissolution, melting and solidification

Fluid Film Model

- Solves flow of thin film on a surface
- Film formation through user initialization, injection through a boundary, or impingement from other phases
- Edge or wave stripping
- Phase-change processes: evaporation and condensation, boiling, melting and solidification
- Mass removal from frozen film in conjunction with mesh morphing to simulate ice accretion

Dispersed multiphase (DMP)

- Lightweight approach that solves for the dispersed phase in an Eulerian manner with one-way or two-way coupling
- Can form, or be formed from, fluid films

Mixture Multiphase (MMP)

- Simplified multiphase model for modeling mixture flows
- Reduces computational effort by assuming mixture is a homogeneous single-phase system

Multiphase Flow (Lagrangian Description)

Lagrangian models can simulate a wide variety of flow processes involving the transport of solid particles, liquid droplets, or gas bubbles—known as dispersed phases—by gaseous or liquid continuous phases.

Lagrangian Multiphase

- Particle equation of motion: drag, pressure gradient, virtual mass, gravity, and body forces
- Multi-component liquid and solid particles

- Heat and mass transfer
- Impingement erosion
- Atomization and breakup
- Collision and coalescence
- Passive scalar interactions
- Mass transfer into VOF phases
- Coulomb forces

Discrete Element Method (DEM)

- Simulation of large numbers of interacting discrete objects
- Particle equation of motion—resolve rotation of particles
- Models for particle-particle contact and particle-boundary interaction
- Particle bonding and breakup
- Spheres, cylinders, capsules, polyhedral, composite, clumped, or coarse grain
- Particle-particle, particle-boundary, and particle-continuous phase heat transfer
- Erosion of walls due to particles: abrasive and impact wear

Injectors for introducing particles into the Lagrangian phase

- Point, part, and surface
- Tabular
- Size distribution functions
- Hollow and solid cones (excluding DEM)
- Pressure swirl (excluding DEM)
- Lattice (only available for DEM)
- Random (only available for DEM)

Motion

STAR-CCM+ can model moving and deforming meshes and multiple frames of reference. Specific features that are related to motion are:

Motion specification

- User-defined rigid body motion
  - Rotation and/or translation
- Dynamic fluid body interaction (DFBI)
  - Model motion of bodies in six degrees of freedom under the action of fluid forces
  - Superimposed motion
  - Body-body couplings
  - External forces
  - Contact prevention

Moving mesh techniques

- Sliding interfaces
- Overlapping meshes using the overset (chimera) technique
- Moving reference frames (MRF)
- Mesh morphing
- Blade flutter for Harmonic Balance
- User-defined vertex motion

Reacting Flows

STAR-CCM+ offers models that simulate chemical processes that occur in combustion, polymerization, and other chemical reactions.

Fuel types

- Solid, liquid, gas

Types of reaction

- Premixed, partially premixed, and non-premixed combustion
- General gas phase chemistry
- Liquid-liquid chemistry
- Interphase reactions
- Surface reactions
- Particle reactions and coal combustion
- Polymerization

Reaction and combustion models

- Flamelet
  - Flamelet Generated Manifold model (FGM)
  - Steady Laminar Flamelet (SLF)
  - Chemical Equilibrium
  - Flame propagation models
    - Coherent Flame model (CFM)
    - Turbulent Flame Speed Closure model (TFC)
- Reacting Species Transport
  - Complex Chemistry
  - Eddy Break-Up model (EBU)
  - Thickened Flame Model (TFM)
  - Eddy Contact Mictromixing (ECM)

Additional models

- NOx emissions
- Soot emissions

- 1D reacting channel
- Spark Ignition

Solid Mechanics

STAR-CCM+ provides an integral finite element solver for the solution of solid mechanics, fluid-structure interaction, and thermal stress problems.

- Static, dynamic, and quasi-static analysis
- Linear and non-linear geometry with full and modified Newton matrix update
- 3D linear and quadratic solid elements: hexahedra, tetrahedra, wedge, and pyramid
- Isotropic, orthotropic, and anisotropic linear elastic materials
- Rayleigh damping
- Loads and constraints for points and surfaces; body loads
- Thermal stresses from mapped temperature field
- Cyclic symmetry boundary condition
- Bonded and small sliding frictionless contacts for conformal and non-conformal meshes
- One-way and two-way Fluid-Structure Interaction (FSI) with added mass pre-conditioner

Adjoint Solver

The adjoint method is an efficient means for predicting the influence of many design parameters and physical inputs on some engineering quantity of interest.

- Available for the coupled implicit solver for both solids and fluids
- Many cost functions available including combination cost functions and user defined
- Provides gradients with respect to mesh error estimates, mesh morpher control points (shape changes), surface deformation, and flow parameters

Electrochemistry

Electrochemistry models in STAR-CCM+ support studies of ion transport along with chemical and electrochemical reactions.

- Ion transport in fluids and solids, including coulombic effects
- Surface electrochemical reactions and heating
- Bulk ion chemical reactions
- 3D micro-structural model for Li-Ion battery cells
- Electro-deposition coating

Plasma

STAR-CCM+ includes electron transport to support studies of cold, non-thermal plasmas.

- Electron transport and energy
- Ion transport

Electromagnetism

STAR-CCM+ contains electric potential and magnetic potential models that solve for the electric and magnetic fields.

- Electrostatics
- Low-Frequency Electrodynamics
  - Electric currents
  - Magnetic fields induced by electric currents, permanent magnets, and excitation coils
  - Transverse magnetic modes
  - Magnetic fields with harmonic time-dependence
- Ohmic heating
- Magnetohydrodynamics (MHD)

Aeroacoustics

STAR-CCM+ aeroacoustic models predict noise generation in fluid systems.

- Broadband noise source models
  - Curle, Goldstein axisymmetric, Proudman, Lilley, and Linearized Euler Equation (LEE)
- Acoustic Perturbation Equation (APE) solver
- Far-field propagation using Ffowcs Williams-Hawkings
- Time to frequency domain analysis tools including FFT, filters, and spectral data

Computational Rheology

Finite element solver for modeling flow and energy problems in which rheological physics dominate.

Material Characteristics

- Newtonian (highly viscous) and Non-Newtonian for either generalized or viscoelastic materials
- Generalized: Power Law, Cross, and Carreau-Yasuda
- Viscoelastic, supporting up to 8 modes: Oldroyd-B, Giesekus-Leonov, Phan Thien-Tanner/JS (linear and exponential forms), and Extended Pom-Pom

Modeling Capabilities

- Axisymmetric, 2D, or 3D
- Steady or implicit unsteady
- Viscous energy for problems with significant heat generation
- Film casting
- Free surface for die-swelling extrusion problems
- Passive scalar for simple mixing
- Residence time
- Partial fill

Solvers

Numerical algorithms solve the systems of equations that STAR-CCM+ constructs from the chosen models and their boundary conditions.

Precision

STAR-CCM+ is available as mixed or double precision.

Finite Volume Method

- Segregated flow solver
  - Automatic continuity initialization
- Coupled flow solver
  - Automatic grid-sequencing initialization
  - Expert driver for solution process
  - Convergence accelerators
- SIMPLE (steady and unsteady) and PISO (unsteady only) solution algorithms
- Algebraic multigrid linear solver
- Harmonic balance method for periodic unsteady flows and single frequency electromagnetics
- Discretization schemes
  - For convection: first-order upwind, second-order upwind, central-differencing (regular and bounded), hybrid second-order upwind/CD (regular and bounded), hybrid MUSCL 3rd-order/CD
  - For time: first-order (Euler implicit), second-order (with 3, 4, and 5 time levels)

Finite Element Method

- Shared or distributed memory linear direct solvers
- In-core or out-of-core memory storage
- Memory estimation
- Multi-threading
- Applies to:
  - Solid mechanics
  - Viscous flow
  - Magnetic Vector Potential

Parallel Execution

- Automatic ParMetis decomposition
- The STAR-CCM+ installation provides IBM Platform MPI and Intel MPI
  - Compatible with Open MPI, Microsoft MPI, SGI MPI, and Cray MPI
- Parallel disk I/O

Data Analysis

STAR-CCM+ provides tools for analyzing simulation results both numerically and visually.

Reports

A report presents a computed summary of the current simulation or CPU data. STAR-CCM+ provides three types.

System Reports

Provide information related to performance and memory

Statistical reports

Apply statistical functions to data within the simulation

Specific Reports

Provide information relevant to, and only available with, specific physical models

Includes reports for harmonic balance, DFBI, reacting flows, and solid mechanics

Monitors

Monitors facilitate sampling and saving of summary information from the simulation during the solution.

Report-based

Provide the evolution of report values as the simulation progresses

Field monitors

Collect data samples from all or part of the solution domain

Types are: maximum, minimum, mean, sum, variance, and co-variance

Plots

Plots graphically display data either from STAR-CCM+ or imported using tables. The appearance and layout of the plots are fully customizable.

Monitor plots

Displays values that are collected by report-based monitors as the solution proceeds

XY plots

Two-dimensional graphs that can display scalar values from the set of available solution variables or from table data

Histogram plots

Two-dimensional bar charts that can display data distribution

Visualization

Visualization of results within STAR-CCM+ is achieved through graphical scenes and associated displayers. Specific rendering of results depends on the type of displayer being used. Any simulation can have any number of scenes each with any number of displayers.

Default scene types

- Geometry, mesh, scalar, vector, or empty

Global scene controls

- Lighting, background colors and images, environment maps, object clipping, annotations, measurement grids, scene update and export, transparency (fast or high quality)
- Ray tracing for realistic visual presentation

Post-processing entities

- Scalar field display
- 2D and 3D vector glyphs
- Many pre-defined colormaps with tools for creating new ones
- Streamlines
- Volume rendering of scalar fields
- Line Integral Convolution (LIC) for vector fields
- Section slices
- Iso-surfaces
- Display of DFBI bodies, forces, and couplings
- Point, line, grid, and arbitrary probes
- Scalar and vector warps for showing exaggerated deformations

Simulation Summaries

STAR-CCM+ can generate user-customizable HTML reports that detail the settings chosen for a simulation.

Solution History

During a simulation run, a solution history stores chosen simulation data on specific analysis surfaces. You can create solution history content for:

- Boundaries
- Selected derived parts: probes, streamlines, sections, iso-surfaces, and warp parts
- Lagrangian/DEM phases

Simulation data can come from any field function (including user-defined) and is written at specified intervals. Intervals are defined by iteration, time-step, delta time, or a specific update event.

- After the simulation ends, Solution Views allow you to interact with the recorded data
- Interactions include advancing to any stored state instantaneously, and creating animations

Data Focus

Method for exploring and interrogating simulation data interactively in order to focus on critical engineering results.

- XY plots provide the canvas for creating data focus selections
- Heatmap XY plots style supports large amounts of data
- Data focus selection filters connect to 3D scenes for interactive data exploration

STAR-View+

A license-free scene viewer for three-dimensional STAR-CCM+ scene files. STAR-View+ allows you to:

- Deliver sets of scenes for post-processing review
- Pan, roll, rotate, and zoom within a scene
- Animate objects within a scene, or a series of scenes

Design Exploration

In STAR-CCM+, Design Manager provides an automated approach for running design exploration studies. Design exploration covers both performance assessment and optimization. In addition to Design Manager, all functions within the STAR-CCM+ environment can be automated using Java macros.

Available Study Types

- Performance Assessment
  - Manual Studies
  - Design Sweeps
- Design Optimization (STAR-Innovate)
  - Design of Experiments
    - 2 Level Full Factorial
    - 3 Level Full Factorial
  - Stochastic Analysis
  - Optimization
    - Weighted sum of all objectives (SHERPA)
    - Multiple objective tradeoff study—Pareto front (MO-SHERPA)

Study Inputs

- 3D-CAD design parameters
- CAD Client design parameters
- Global parameters

Study Outputs

- Responses (based on reports)
  - Information only
  - Objectives
  - Constraints
- Scenes and Plots

Simulation Run Options

- Local jobs
- Remote jobs on a cluster

Study Analysis

- Output Tables
- Snapshots
- Project Plots

- Export to HEEDS|Post

Interaction with CAE Software

STAR-CCM+ provides data mapping to other tools, file-based coupling, and tightly coupled co-simulation

Co-simulation between STAR-CCM+ and 1D tools

- GT-SUITE (Gamma Technologies, LLC)
- LMS Imagine.Lab Amesim (Siemens PLM Software)
- RELAP5-3D (Idaho National Laboratory, INL)
- WAVE (Ricardo Software)

Co-simulation between STAR-CCM+ and 3D tools

- STAR-CCM+ (co-simulation between two STAR-CCM+ simulations)
- Abaqus (Dassault Systèmes)
- Third-party/user-specific tools using the STAR-CCM+ co-simulation API

File-based coupling between STAR-CCM+ and CAE tools

- Abaqus, using .inp files
- Ansys, using .cdb files
- Nastran, using .nas, .bdf, or .dat files
- TAITherm, using .ntl or .neu files
- RCAS, using .in files

Data mappers (interpolate data to, or from, different meshes)

- Surface to surface, volume, or beam
- Volume to surface, or volume
- Beam to surface
- Table to surface, volume, or beams

Application-Specific Tools

STAR-CCM+ includes tools for specific industry applications. These tools shorten the time required for creating and analysing typical cases that those industries require.

STAR-ICE

STAR-ICE provides a dedicated interface within STAR-CCM+ for simulating unfired internal combustion engines (ICE) using transient moving-mesh methods.

Geometry

Import engine CAD and map to recognized parts: full or half-engine (with symmetry)

Physics

STAR-ICE uses a subset of physics models that are relevant for unfired ICE cases:

- Time: transient calculations
- Time-step: constant or stepped table
- Solution domain: single-domain (gas)
- Material: nitrogen
- Heat transfer modes: conduction and convection (forced and buoyancy driven)
- Flow regime: turbulent (Realizable K-Epsilon Two-Layer All Y+)
- Equation of state: ideal gas

Conditions

- Geometry: full or half-engine (with symmetry)
- Initialization: import tabular pressure and temperature data
- Flow boundaries: pressure outlets using tabular pressure and temperature data
- Wall boundaries: thermal and wall surface specification

Motion

- Cylinder motion: calculates piston position from engine geometry and operating conditions
- Valves motion: import valve lift curves

Mesh

- Volume mesh: trimmed cell mesher (predominantly hexahedral cells)
- Prism layers: prismatic cells on walls for boundary flow accuracy
- Cell size: based on best practices
- Mesh refinement zones for chamfer and valves

Solver

- Segregated solver for flow and energy
- SIMPLE algorithm for pressure-velocity coupling

- Solver under-relaxation
- Live monitoring of solution data

Results

- Visualization of geometry, mesh, and solution data
- Access to solution data through plane sections that are positioned and oriented based on the engine geometry
- Plots for characteristics of the in-cylinder flow, such as swirl, x-tumble, and y-tumble

Electronics Cooling

The Electronics Cooling Toolset provides a dedicated interface within STAR-CCM+ for thermal management simulation of electronics devices.

Geometry

- Geometry generation that is based on simple shape templates
- Import of IDF (Intermediate Data Format) files
- Import of CAD models from other CAD packages
- Transfer of CAD data from the STAR-CCM+ CAD modeler

Physics

- Time: Steady-state calculations
- Solution domains: multi-domain (fluid and solid)
- Heat transfer modes: conduction, convection (forced and buoyancy driven), and radiation (wavelength independent Surface-to-Surface (S2S) radiative heat transfer)
- Flow regime: laminar or turbulent (Realizable K-Epsilon Two-Layer All Y+)
- Equation of state: constant density or ideal gas

Conditions

- Environment: ambient boundary conditions
- Thermal specification: specification of material or specification of layers (PCB)
- Compact thermal models: resistor networks for chips
- Heat dissipation: volumetric heat sources

Mesh

- Volume Mesh: Trimmed cell mesher generates predominantly hexahedral cells
- Prism Layers: prismatic cells on walls for boundary flow accuracy
- Cell size: automatic calculation of target and minimum cell size that is based on overall model size
- Mesh Refinement: integrated features for mesh refinement to detect and capture complex geometries
- Local mesh controls: cell size customization on surfaces and within volumetric zones

Solver

- Segregated solver for flow and energy

- Solver under-relaxation
- Live monitoring of solution data

Results

- Visualization of geometry, scalar, and vector quantities
- Access to solution data through streamlines, isosurfaces, and plane sections
- Residuals and XY plots
- Summary reports

Usability and Productivity

The STAR-CCM+ client provides several features for an efficient and extensible simulation framework.

Custom Simulation Trees

Allows you to create smaller, more manageable simulation trees from selected objects in the main tree. These custom trees are saved with the simulation file.

Simulation Tree Comparison

Provides a side-by-side comparison between two simulation files so that you can identify any differences.

Field Functions

STAR-CCM+ supports an expression language by which you can create scalar, vector, or tensor functions that vary in space and time.

- Includes an expression editor with built-in functions, predictive typing, auto-complete, and a preview panel

Parameters

User-defined, spatially invariant scalar and vector constants. These parameters can be used in field functions and can be referenced from any expression.

Filters

Filters, created from one or more logic-based queries, facilitate automation through repeatable object selection.

- Queries are built from predicates (conditional statements) that are applied to a set of objects
- Queries support Boolean operators (AND, OR) and can be nested

Tagging

Tags provide a mechanism for grouping objects in almost every part of the simulation. You can create custom trees and complex filters using tags.

Siemens PLM Software

Headquarters
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asia-Pacific
Suites 4301-4302, 43/F
AIA Kowloon Tower,
Landmark East
100 How Ming Street
Kwun Tong, Kowloon
Hong Kong
+852 2230 3308

About Siemens PLM Software

Siemens PLM Software, a business unit of the Siemens Digital Factory Division, is a leading global provider of product lifecycle management (PLM) and manufacturing operations management (MOM) software, systems and services with over 15 million licensed seats and more than 140,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software works collaboratively with its customers to provide industry software solutions that help companies everywhere achieve a sustainable competitive advantage by making real the innovations that matter. For more information on Siemens PLM Software products and services, visit www.siemens.com/plm.

© 2017 Siemens Product Lifecycle Management Software Inc. Siemens and the Siemens logo are registered trademarks of Siemens AG. D-Cubed, Femap, Fibersim, Geolus, GO PLM, I-deas, JT, NX, Parasolid, Polarion, Solid Edge, Syncrofit, Teamcenter and Tecnomatix are trademarks or reg istered trademarks of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries. All other logos, trademarks, registered trademarks or service marks belong to their respective holders.

SIEMENS
Ingenuity for life

Siemens PLM Software

STAR-CCM+
User Guide

Version
12.06

www.siemens.com/mdx

© 2017 Siemens Product Lifecycle Management Software Inc.

Any and all Siemens Product Lifecycle Management Software Inc. brand, product, service and feature names, logos and slogans are registered trademarks or trademarks of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries. All other brand, product, service and feature names or trademarks are the property of their respective owners.

Contents

Getting Started 1

Navigating Help 2

Using Online Help 3

Browsing the Online Help 4

Accessing Context Sensitive Help 6

Using the Navigation Frame 8

Using the Toolbar Frame 13

Changing the Default Browser 14

Understanding Typographic Conventions 15

STAR-CCM+ Simulation Process 16

Simulation Categories 17

STAR-CCM+ Licensing 24

Power-on-Demand Licensing 28

STAR-CCM+ Add-on Modules 41

Power Tokens 46

Getting Support 50

Gathering Information on the Issue 53

Preparing a Case for Support 59

Suggesting Enhancements to STAR-CCM+ 61

STAR-CCM+ Default Options 63

Resetting STAR-CCM+ Default Options 70

Controlling Graphics Performance 72

Troubleshooting Graphics 74

Resolving Hardware Issues 75

Working Without Graphics Hardware 76

A Scene Will Not Open 78

Improving the Rendering of Concave Faces 80

Z-Fighting Appears 82

STAR-CCM+ Fails to Detect Graphics Card (Linux Only) 83

Credits 84

User Interface 94

Using the STAR-CCM+ Workspace 96

Using the Workspace 97

Using the Menus 114

Using the Toolbars 121

Using the Explorer Pane 131

Using the Table Editor 134

Using the Properties Window 142

Using the Graphics Window 148

Using the Output Window 199

Using Keyboard Shortcuts 213

Conditions, Values, and Profiles 221

Tables....268
What Are the Important Concepts?....287
What Are Simulations?....288
What Are 3D-CAD Models?....289
What Are Parts?....290
What Are Continua?....292
What Are Models?....295
What Are Regions?....298
What Are Boundaries?....301
What Are Interfaces?....306
What Are Solvers?....307
What Are Reports, Monitors and Plots?....309
What Are Scenes?....310
Working with Simulation Objects....311
Working with the Object Tree....312
The Object Selection Dialog....325
Comparing Objects and Simulation Trees....342
Using Custom Simulation Trees....344
Tagging Simulation Objects....347
Object Filters....350
Using Query-Based Selection....394
Global Parameters....397
Field Functions....399
Field Function Types....401
Creating a User Field Function....405
Copying User Field Functions and User Field Function Properties....412
Ignoring Boundary Values....413
Interpolating Field Functions Using Inverse Distance Weighting....415
User Field Functions Syntax....417
User Field Function Examples....437
Field Functions Reference....440
Engineering Units....446
Working with Units....448
Using Units While Setting Values....452
Units Reference....456
Scripting the Application....458
Recording a Macro....460
Editing the Macro....463
Playing the Macro....474
Debugging a Macro Using an IDE....479
Running Multiple Simulations Consecutively....485
Playing Macros Within Macros....487
Adding Custom Macros to the Toolbar....488
Using the Java API....490
Working with Default Macros....493
Macro API Changes in Past Releases....497
Working with User Code....736
What Are User Libraries?....738
Working With User Functions....740
Creating a New User Library....743
Accessing a User Library....745

User Function Interface Reference (C)....749
Fortran User Interface Reference....753
Variables Available to User Functions....757
User Code Examples....758
Using User Code on Particles and Parcels....764
Working with the Plugins Manager....776
Simulation Assistants....781
The Simulation Assistant User Interface....784
Setting Up a Simulation Assistant Project in NetBeans....794
Developing a Simulation Assistant....807
Wizard Guide....834
Using the Turbo Wizard....835
Using the Fire and Smoke Wizard....879
Using the Thermal Comfort Wizard....901
Introduction to the SPEED Motor Tool Import Wizard....962
Using The IDF Import Wizard....1005

Client-Server Setup....1009
Client-Server Connections....1012
Providing Remote Server Access....1014
Server Processes Reference....1015
Connect to Server Dialog Reference....1016
Connecting a Client from the Command Line....1017
Running the Client in Batch....1018
Configuring the Server to Shutdown When Idle....1020
Closing the Server Automatically Upon Solver Completion....1022
Working with Remote Servers....1023
Starting a Proxy Server on a Host....1025
Starting a Server without a Client....1027
Displaying Diagnostic Messages from the Server and Client....1028
Setting Network Binding Order for Failed Visualization Connections....1030
Mixed or Double Precision....1031
Launching STAR-CCM+ and Working With Simulations....1032
Launching STAR-CCM+....1033
Interactive Startup Reference....1034
Creating and Opening Simulations from the Command Line in Serial Mode....1037
Starting Remote Servers from the Command Line....1039
Restrictions on Names of Files and Paths in STAR-CCM+....1041
Using the Simulation Pop-Up Menu....1043
Saving Simulations to File....1044
Reverting Unsaved Changes to the Simulation....1053
Closing the Simulation....1054
Closing the Client Workspace....1055
Viewing and Editing the Log File....1057
Using Summaries....1060
Working With Parallel Servers....1063
Parallel Servers and Requirements....1064
Supported MPI Implementations....1066
Using a Local Multi-Core Workstation....1074
Using a Remote Multi-Core Workstation....1077

Using a Group of Machines....1080
Troubleshooting....1089
Balancing....1091
Getting Parallel Help through the Command Line....1092
Viewing the Partitioning....1093
Using Batch Management Systems....1094
Cluster Node Communication....1101
Expert MPI Options....1106
Running Distributed Parallel Simulations on Multiple Windows Machines....1107
Command-Line Reference....1109
Working with .ini Files....1118

Pre-Processing....1120
Modeling Geometry....1121
Preparing the Geometry....1123
Using 3D-CAD....1124
Importing Geometry....1486
Understanding Geometry Parts....1517
Manipulating Parts....1583
Part Contacts....1655
Repairing CAD Geometry....1668
The Surface Repair Tool....1674
Displaying the Geometry....1964
Defining the Regions Layout....1966
Regions Layout Workflow....1967
Assigning Geometry Parts to Regions....1969
Initial Assignment of Boundary Types....1990
Interfaces Overview....1991
Region Manipulation....2056
Manipulating Boundaries....2084
Performing Surface Boolean Operations....2127
Meshing....2141
Finite Volume Mesh Construction....2144
Meshers in STAR-CCM+....2147
Advantages of Parts-Based Meshing....2156
The Meshing Workflow....2157
Creating a Conformal Mesh....2159
Importing Volume Meshes....2161
Parts-Based Meshing....2164
Region-Based Meshing....2302
Surface Meshers....2389
Volume Meshers....2438
Mesh Refinement....2570
Directed Meshing....2593
Two Dimensional and Axisymmetric Meshes....2653
Mesh Visualization....2665
Checking the Surface Mesh....2672
Checking the Volume Mesh....2676
Meshing Field Functions Reference....2722
Working With Representations....2727
Scaling the Entire Mesh....2787

Working with Baffles 2790
Exporting Mesh and Solution Data 2800
Multi-Region Meshing 2808

Simulating Physics 2817

Defining Physics Workflow 2819

Setting Up Physics 2822
Physics Continua 2823
Selecting Physics Models 2825
Setting Types 2832
Region Types Reference 2834
Boundary Types Reference 2836
Applying Quantities by Subgroup 2843
Extensive and Intensive Quantities in Values 2846

Running Physics Simulations 2847
Using Initial Conditions 2848
Using Solvers 2855
Setting Up Stopping Criteria 2856
Obtaining a Solution Interactively 2875
Troubleshooting the Solution 2898

Using Overset Meshes 2904
Understanding the Overset Workflow 2908
Preparing Overset Regions 2909
Coupling Regions 2911
Enforcing Mass Conservation 2917
Dynamic Overset Behavior 2918
Overset Topology 2924
Setting up Overset Mesh with Fluid Film 2927
Generating Suitable Meshes 2931
Visualizing and Analyzing Solution 2933
Activating Load Balancing for Parallel Overset Mesh Simulations 2936
Overset Mesh Guidelines 2937
Overset Setup Examples and Best Practices 2943
Overset Mesh Methodology 2946
Overset Mesh Reference 2950
Overset Mesh Field Function Reference 2955
Troubleshooting Overset Mesh 2959
Overset Mesh Bibliography 2963

Modeling Space, Time, and Motion 2964
Modeling Space 2965
Coordinate Systems 2977
Modeling Time 3001
Reference Frames 3011
Modeling Motion 3025

Modeling Materials 3079
Modeling a Single-Component Material 3081
Modeling a Multi-Component Mixture 3085
Modeling Multi-Component Solids 3090
Modeling a Multiphase Mixture 3103
Modeling Linear Isotropic Elastic Stress 3104

Setting Material Properties Methods....3106
Material Databases....3194
Modeling Flow and Energy....3213
Flow and Energy Workflow....3217
High-Speed Flow Regimes....3222
Pressure....3224
Modeling the Equation of State....3231
Modeling Flow and Energy Using a Coupled Approach....3248
Coupled Flow Controls and General Setup Recommendations....3275
Segregated Flow Guidelines....3278
Segregated Flow Model Reference....3282
Initializing Flow....3298
Initial Conditions....3300
Flow Boundaries Reference....3301
Partial Slip....3318
Boundary Conditions Formulation....3321
Reporting Mass Flowrate....3336
Modeling Passive Scalars....3337
Modeling Gravity....3351
Modeling Axisymmetric Swirl....3354
Modeling Fully Developed Flow and Energy....3357
Tracking Vortices in the Flow....3364
Vorticity Confinement Formulation....3370
Region Values....3372
Flow Node Reference....3373
Fluid Flow and Energy Field Functions Reference....3378
Fluid Flow Formulation....3391
Modeling Viscous Flow....3394
Modeling Rheology....3396
Setting Up Viscous and Viscoelastic Fluid Flow....3397
Extrusion and Free Surface Flow....3400
Setting Up Extrusion and Free Surface Flow....3402
Setting Up a Film Casting Simulation....3404
Viscous Flow Models....3406
Mesh Requirements and Guidelines....3421
Guidelines for Co-Extrusion....3423
Guidelines for the eXtended Pom-Pom Model (XPP)....3424
Viscous Flow Field Function Reference....3425
Modeling Heat Transfer....3427
Heat Conduction....3429
Convective Heat Transfer....3430
Thermal Radiation....3432
Verification and Validation of Results....3434
Designing a Heat Transfer Optimization Study....3435
Convective Heat Transfer Coefficients....3437
Setting an Environment Boundary Condition for Heat Transfer....3444
Modeling Boiling....3452
Modeling Moisture Boiling in Solids....3456
Modeling Thin Film: Deicing and Defogging....3460
Modeling Heat Exchangers....3474
Modeling Energy in Solids....3493

Energy Reference....3504
Heat Transfer Field Functions Reference....3520
Energy Formulation....3526
Modeling Species....3543
Working with Species....3545
What Is the Segregated Species Solver?....3554
Setting a Species Region Source....3555
Species Transport Formulation....3558
Species Transport Bibliography....3561
Species Transport Field Functions Reference....3562
Modeling Porous Media....3563
Porous Regions Workflow....3567
Porous Media Models....3569
Porous Media Model Workflow....3573
Examples of Setting Porous Resistance Properties....3575
Influence of Porous Resistance on Temperature for Incompressible Flow....3584
Using Porous Media to Model Flow in Tubes....3585
Solving the Adjoint....3586
Adjoint Analysis Workflow....3588
Enhancing the Coupled Flow Solver Convergence Behavior....3592
Creating the Adjoint Cost Functions....3594
Running the Adjoint Solver....3596
Computing Adjoint Error Estimates....3599
Troubleshooting the Adjoint Solver....3601
Setting Up a Shape Optimization....3602
Creating Initial Control Point Positions....3603
Specifying Mesh Deformation Boundary Conditions....3604
Computing the Shape Sensitivity....3607
Obtaining New Control Point Positions....3609
Computing the Mesh Deformation....3610
Obtaining Boundary Condition Value Sensitivity....3612
Visualizing the Adjoint Solution....3613
Visualizing the Mesh Adjoint....3615
Visualizing the Surface Sensitivity....3616
Adjoint Cost Functions Reference....3618
Point Set Reference....3621
Adjoint Field Functions....3623
Modeling Fans and Blowers....3624
Overview of Fan Performance Curves....3628
Fan Pressure Components....3629
Fan Configurations....3630
Modeling Fan Performance Curves....3633
Using the Fan Interface....3634
Creating a Fan Interface....3635
Fan Interface Properties....3636
Specifying the Fan Performance Curve Type....3638
Specifying the Fan Performance Curve as a Table....3639
Specifying the Fan Performance Curve as a Polynomial Function....3642
Specifying the Fan Performance Curve Polynomial....3644
Specifying the Fan Curve Temperature Scaling....3646
Modeling the Fan Swirl....3647

Orientation of Parts-Based Interfaces....3649
Setting the Fan Interface Verbosity....3650
Adjusting the Fan Interface Intersection Tolerance....3651
Creating a Blower Interface....3652
Blower Interface Properties....3653
Modeling the Blower Swirl....3656
Modeling Multiple Fans and Blowers in a Single Interface....3657
Using the Fan Momentum Source....3658
Importing the Fan Performance Data Table....3659
Activating the Fan Momentum Source....3660
Specifying the Fan Performance Curve....3662
Fan Momentum Source Properties....3664
Scaling the Fan Performance Curve....3666
Working with the Fan Solver....3667

Virtual Disk Model....3668
Selecting the Virtual Disk Model....3669
Body Force Propeller Method....3671
Blade Element Method....3687
1D Momentum Method....3713
User Defined Method....3722
Reporting Virtual Disk Results....3729
Virtual Disk Properties....3731
Inflow Specification Properties Reference....3733
Virtual Disk Reports Properties....3735
Virtual Disk Field Functions....3738

Modeling Turbulence....3740
Selecting a Turbulence Modeling Approach....3741
Reynolds-Averaged Navier-Stokes (RANS) Turbulence Models....3746
Scale-Resolving Simulations....3824
Wall Treatment....3862

Modeling Transition....3884
Turbulence Suppression Model....3885
Gamma ReTheta Transition Model....3893
Gamma Transition Model....3900
Transition Guidelines....3904
Transition Formulation....3906

Modeling Wall Distance....3923
Wall Distance Models Reference....3924
Poisson Formulation....3928

Modeling Radiation....3929
Radiation Characterization....3931
Radiation Model Workflow....3933
Modeling Surface-to-Surface Radiation....3935
Modeling Participating Media Radiation....3952
The Gray Thermal Radiation Model....3958
The Multiband Thermal Radiation Model....3960
Radiation Properties on Boundaries....3963
Modeling Solar Radiation Loads....3968
Radiation Field Functions Reference....3974

Modeling Aeroacoustics....3979
What Is Aeroacoustics?....3980

How Do I Conduct an Aeroacoustics Analysis?....3982
Selecting an Aeroacoustics Model....3984
Using Broadband Noise Source Models....3985
Using the Acoustic Wave Model....3996
Using Direct Noise Calculation....4019
Using the Acoustic Suppression Zone Model....4020
Using the Ffowcs Williams-Hawkings Aeroacoustics Models....4024
Using the Lighthill Stress Tensor....4056
Aeroacoustics Bibliography....4060
Modeling Hydroacoustics....4064
Combustion and Other Reacting Flows....4065
Reacting Flow Applications....4068
Reacting Flow General Workflow....4075
Reacting Flow General Reference....4081
Reacting Species Transport....4099
Flamelet....4138
Emissions....4183
Interphase....4205
Surface Chemistry....4221
Reacting Channels....4233
Common Reacting Flow Actions....4247
Modeling Multiphase Flow....4251
What Are Multiphase Flows?....4253
Using the Lagrangian Multiphase Model....4263
Modeling Fluid Film....4442
Using the Discrete Element Method....4511
Using the Eulerian Multiphase Model....4618
Eulerian Multiphase Mixture Model....4645
Using the Multiphase Segregated Flow Model....4675
Dispersed Multiphase Model....4872
Using the Volume Of Fluid (VOF) Multiphase Model....4889
Modeling VOF Waves....5007
Modeling Dynamic Fluid Body Interaction....5039
What Is a Rigid Body?....5041
What Is a DFBI Rotation and Translation Motion?....5042
What Is a DFBI Embedded Rotation Motion?....5043
What Is a DFBI Morphing Motion?....5045
Setting up Superposing Motions....5048
What Is the Six DOF Solver?....5054
What Is the 6-DOF Motion Solver?....5058
Working with 6-DOF Bodies....5059
Working with 6-DOF Body Couplings....5096
Setting up a Two-Dimensional DFBI Simulation....5114
DFBI Formulation....5116
DFBI Bibliography....5134
Modeling with Harmonic Balance....5135
Setting Up a Harmonic Balance Simulation....5137
What Is the Harmonic Balance Model?....5138
What Is the Harmonic Balance Flow and Energy Model?....5142
What Are the Harmonic Balance Turbulence Models?....5159
Defining a Wake....5162

What Is the Harmonic Balance Flutter Motion?....5174
What Is the Implicit Harmonic Balance Solver?....5183
Post-Processing Harmonic Balance Solution Data....5184
What Is the HB Solution View?....5186
Filtered Mode Field Functions....5190
Harmonic Balance Guidelines....5193
Harmonic Balance Formulation....5196
Harmonic Balance Field Functions Reference....5204
Solid Stress....5207
General Workflow for Stress Analysis....5209
Modeling Fluid-Structure Interaction (FSI)....5231
Solid Stress Reference....5235
Mesh Requirements and Guidelines....5260
Solution Analysis Guidelines....5264
Bibliography....5267
Modeling Finite Volume Solid Stress....5268
Selecting the Finite Volume Solid Stress Model....5269
Applying Boundary and Initial Conditions....5270
Applying a Body Force....5272
Modeling Dynamic Problems....5273
Modeling Rayleigh Damping....5274
Modeling Large Displacements....5276
Working with Interfaces in Solid Stress....5278
Modeling FSI Using Finite Volume Stress....5280
Using the Summary Report for Bad Cells....5284
Setting Parameters of Other Solvers....5285
Finite Volume Solid Stress Model Reference....5286
FV Stress Solver Reference....5292
Finite Volume Stress Analysis Guidelines....5294
Solid Stress Formulation....5298
Electromagnetism....5305
Modeling Electrostatics....5309
Modeling Electric Currents....5311
Modeling Magnetic Fields....5318
Modeling Ohmic Heating....5334
Modeling Magnetohydrodynamics (MHD)....5336
Physics Models Reference....5338
Using the Circuit Model....5397
Accessing Electromagnetism Constants through the Java API....5399
Troubleshooting....5400
Modeling Electrochemistry....5401
Electrochemistry Applications....5403
Li-Ion Battery Cells....5409
Ionic Species Flux....5428
Electrochemical Surface Reactions....5443
Bulk Ion Chemical Reactions....5473
Electro-Deposition Coating....5479
Electrochemistry Formulation....5485
Electrochemistry Bibliography....5502
Electrochemistry Nomenclature....5503
Modeling Plasma....5508

Modeling Cold, Non-Thermal Plasma....5511
Modeling Plasma Arcs....5521
Plasma Nomenclature....5525
Plasma Bibliography....5527
Modeling Batteries....5528
What Is Battery Modeling?....5530
Selecting the Battery Model....5539
What Is the Battery Solver?....5540
Working with Battery Design Studio....5541
Preparing Program (.prg) Files....5558
Working with a Battery Cell....5561
Setting Up a Battery Module....5577
Generating a Battery Module....5608
Generating a Battery Mesh....5614
Setting Up Connector Ohmic Heating....5617
Defining External Battery Casing....5622
Defining an Electrical Load....5630
Defining Electrical Circuits....5636
Exporting Electrochemistry Data....5657
Evaluating Electrode Structure Tortuosity....5659
Battery Modeling Formulation....5660
Battery Field Functions Reference....5670
Modeling Region Sources....5672
Marking Cells for Region Sources....5673
Setting Fluid Sources or Sinks....5676
Prescribing Constant Temperature in Fluid Cells....5677
Remedying Cell Quality....5681
Troubleshooting....5682
Cell Quality Remediation Field Functions Reference....5683
Controlling Domain Decomposition....5685
The Partitioning Solver....5686
Guidelines for Applying STAR-CCM+....5691
Guidelines for Fluid Flow and Energy....5692
Guidelines for Aeroacoustics Calculations....5700
Guidelines for Aerodynamics Calculations....5726
Guidelines for Simulating Air Flow in Electronic Systems....5735
Modeling Fluid-Structure Interaction....5741
Modeling Atmospheric Boundary Layers....5747
Preparing CAD Models for Import....5753
Migrating from Other Codes....5757
Solving Transport Equation....5760
Gradients Model Reference....5761
AMG Linear Solver Reference....5763
Under-Relaxation Factor Ramp Reference....5769
Transport Equation Solvers Reference....5770

Post-Processing....5771
Analysis Workflow....5773
Accessing Solution Data through Derived Parts....5775
Defining Derived Parts: General Workflow....5778

Using the Visual Tools....5782
Reading an External File....5790
Exporting to an External File....5792
Calculating the Surface Area of a Part....5794
Defining Implicit Sections....5796
Defining an Isosurface....5805
Defining a Threshold....5807
Defining a Scalar Warp Surface....5809
Defining a Vector Warp Surface....5810
Defining a Streamline....5812
Defining an (m',theta) Warp....5815
Defining a Separation/Attachment Line....5819
Defining a Vortex Core Part....5822
Defining Probes....5825
Defining a Cell Surface....5832
Defining a Resampled Volume....5833
Derived Parts Reference....5843
Derived Parts Bibliography....5856
Visualizing the Solution....5857
Working with Scenes....5859
Changing the Attributes of a Scene....5873
Displaying Geometry and Mesh....5912
Displaying Scalar Quantities....5923
Displaying Vector Quantities....5945
Displaying Streamlines....5974
Displaying Dynamic Fluid Body Interaction (DFBI)....5988
Animating Scenes....5996
Using Displayers....6011
Advanced Rendering....6063
Saving and Printing Scene Data....6075
Controlling Display Formats....6082
Annotations....6086
Color Maps....6111
Using Transforms....6121
Working with Views....6135
Obtaining Benchmark Information for a Scene....6151
Linking Scene Views....6153
Recording and Viewing Solutions....6156
Understanding the Simulation History File....6157
Recording Solution Data....6159
Retrieving Results from the Simulation History Files....6165
Using Solution Snapshots....6171
Animating Solution Views....6172
Sampling Data Values During and After Running....6175
Reporting Results....6179
What Is a Report?....6180
System Reports....6182
Statistical Reports....6185
Specific Reports....6199
Expression Report....6243
Understanding Report Errors....6245

Using Reports as Field Functions.... 6246
Reporting Additional Field Functions at Maximum or Minimum.... 6247
Using Coordinate Systems with Force and Moment Reports.... 6248
Using Bands with Force and Moment Reports.... 6250
Working with Parts-Based Interfaces in Reports.... 6251
Reports Bibliography.... 6256
Monitoring the Solution.... 6257
What Is a Monitor?.... 6258
Customizing Monitors.... 6261
Using Report-Based Monitors.... 6263
Using Field Function Monitors.... 6265
Properties of the Monitors Manager.... 6271
Pop-Up Menu of the Monitors Manager.... 6272
Plotting Results.... 6273
Plotting Workflow.... 6277
Working with Monitor Plots.... 6278
Working with XY Plots.... 6287
Working with Histogram Plots.... 6305
Plotting with Multiple Axes.... 6311
Updating Plots.... 6313
Deleting Plots.... 6316
Renaming Plots.... 6317
Copying Plots.... 6318
Customizing Plots.... 6319
Adding and Removing Data in a Plot.... 6336
What Is a Tabular Data Set?.... 6337
What Is a Derived Data Set?.... 6339
Tabulating Data.... 6341
Exporting Data.... 6343
Printing and Saving Plot Images.... 6344
Plot File Formats Reference.... 6346
Pop-Up Menu of the Plots Manager.... 6348
Exploring Solutions with Data Focus.... 6350
Data Focus Workflow.... 6352
Data Focus Objects Reference.... 6355
Triggers and Update Events.... 6356
Setting the Trigger for an Update.... 6357
Trigger Types.... 6359
Trigger Properties.... 6360
User-Defined Update Events.... 6361
Using Data Set Functions.... 6376
Creating Data Set Functions.... 6379
What Are Point Time Histories?.... 6381
What Are Point Time Fourier Transforms?.... 6383
Using Point Time Fourier Transforms.... 6391
What Are Point-to-Point Time Fourier Transforms?.... 6392
What Are Point-to-Point Time Correlations?.... 6396
What Are Point Inverse Time Fourier Transforms?.... 6398
What Are Line Time Histories?.... 6402
What Are Line Time Fourier Transforms?.... 6405
Using Line Time Fourier Transforms.... 6408

What Are Line Spatial Fourier Transforms?....6409
Using Line Spatial Fourier Transforms.... 6411
What Are Surface Time Histories?....6412
What Are Surface Time Fourier Transforms?....6415
FIR Filtering to Compute Band Filtered Surface Time Histories....6418
What Is Derived Data?....6419
Post-Processing Data Set Functions....6447
Pop-Up Menu of the Data Set Functions Manager....6470
Data Set Functions Formulation....6471

Design Manager....6481
Study Inputs....6484
Study Outputs....6486
Study Types....6489
The SHERPA Algorithm....6493
Study Runs....6495
Study Analysis....6497
Design Manager Licensing....6500
Design Manager Workflow....6505
Creating a Design Manager Project....6507
Setting Up a Design Study....6511
Running a Design Study....6523
Monitoring a Design Study....6539
Analyzing the Results of a Design Study....6541
Design Manager Reference....6555
Simulations Reference....6557
Design Studies Reference....6558
Snapshots Reference....6577
Plots Reference....6580
Tools Reference....6588
Toolbars Reference....6590
Command Line Options Reference....6593

Interacting with CAE Software....6597
Importing into STAR-CCM+....6599
Importing Mesh and Solution Data from STAR-CD....6600
Importing CAE Models....6602
Exporting Mesh and Solution Data....6623
Automatically Exporting Mesh and Solution Data at Regular Intervals....6627
File-Based Coupling....6631
Understanding File-Based Coupling....6632
Mapping STAR-CCM+ Solution Data to a CAE Model....6636
Exporting Mapped Data....6646
Mapping and Exporting Transient Surface Data....6651
Importing CAE Solution Data....6654
Mapping CAE Solution Data to a STAR-CCM+ Model....6661
Exchanging Heat Transfer Coefficients....6667
Working with Specific CAE Products....6675
Working with Abaqus....6676

Working with ANSYS....6679
Coupling with NASTRAN....6685
Working with TAITherm....6693
Working with es-ice or pro-STAR....6698
Working with Ensight....6709
Working with the STAR-CCM+ Boundary Data (.sbd) Format....6711
Working with the Visualization Toolkit (VTK) Format....6716
Coupling with RCAS....6717

STAR-CCM+ to STAR-CCM+ Co-Simulation....6722
Co-Simulation Interface and Exchanged Fields....6725
STAR-CCM+ Co-Simulation Workflow....6727
STAR-CCM+ Co-Simulation Models Reference....6741
Partitioned Coupling Model Reference....6753
Uncoupled Mesh Entities Treatment Model Reference....6756

LMS Amesim Co-Simulation....6758
Specifying Co-Simulation Settings in LMS Amesim....6760
Specifying Co-Simulation Settings in STAR-CCM+....6762
Coupling STAR-CCM+ to LMS Amesim....6765
LMS Amesim Co-Simulation Model Reference....6766

Abaqus Co-Simulation....6771
General Requirements....6773
Co-Simulation Interface and Exchanged Fields....6775
General Workflow for Co-Simulation with Abaqus....6777
Specifying Co-Simulation Settings in Abaqus....6778
Specifying Co-Simulation Settings in STAR-CCM+....6780
Running the Co-Simulation and Terminating Abaqus....6787
Restarting from a Previous Abaqus Simulation....6788
Remeshing in STAR-CCM+....6790
Post-Processing the Co-Simulation Results....6791
Allocating Cores to Abaqus in Batch Queuing Systems....6792
Abaqus Co-Simulation Models Reference....6793
Abaqus Link Reference....6795
STAR-CCM+ Properties and Equivalent Abaqus Keywords....6806
Additional Considerations....6807

GT-SUITE Co-Simulation....6810
Preparing the GT-SUITE Simulation for Co-Simulation....6815
Preparing the STAR-CCM+ Simulation for Co-Simulation....6824
Coupling to GT-SUITE....6830
Setting Up STAR-CCM+ and GT-SUITE for a Restart Run....6833
GT-SUITE Co-Simulation Model Reference....6835

WAVE Co-Simulation....6842
Preparing the WAVE Simulation for Co-Simulation....6844
Preparing the STAR-CCM+ Simulation for Co-Simulation....6847
Coupling STAR-CCM+ to WAVE....6852
WAVE Co-Simulation Model Reference....6853

RELAP Coupling....6857
What is RELAP Coupling?....6858
What is a RELAP Zone or Flow Zone?....6859
Preparing the RELAP Case....6861
Setting Up the STAR-CCM+ Simulation....6862
Coupling STAR-CCM+ to RELAP....6868

Reacting Channel Coupling 6875
Modeling Reacting Channels 6877
Reacting Channel Coupling Formulation 6887
Reacting Channel Coupling Nomenclature 6890
Co-Simulation API 6892
Co-Simulation API Overview 6894
Preparing Your Code for Co-Simulation with STAR-CCM+ 6899
Running a Co-Simulation between Partner Code and STAR-CCM+ 6908
Data Mappers 6910
Mapping Fields Using Data Mappers 6912
Data Mappers Reference 6920
Data Mapper Formulation 6931
Examples of Using Data Mappers 6937
The STAR-CCM+ Component for Isight 6939

Add-ons 6942
Electronics Cooling Toolset 6943
Accessing the Electronics Cooling Toolset 6946
Setting up an Electronics Cooling Simulation 6948
Electronics Cooling Toolset Reference 6969
Resistor Networks Formulation 7049
Bibliography 7050
Optimate User Guide 7051
Capabilities of Optimate and Optimate+ 7052
How Does Optimate Work? 7053
Optimate Run Modes 7054
Optimate Variables 7057
Optimate Responses 7059
The Optimate Workflow 7060
Saving and Loading Optimate Projects 7080
Post-Processing Optimate Results 7081
Customizing the Optimate Workflow Using Java Macros 7082
Using Design Sets 7083
Running Optimate Jobs 7085
Monitoring Designs 7102
Exporting Scenes from Optimate to STAR-View+ 7105
Running a CAD Robustness Study 7106
Optimate Licensing 7107
Coupling Optimate to External Optimization Codes 7108
Optimate Options 7109
File Descriptions 7112
Troubleshooting 7114
Optimate Tutorials 7116
Design Sweeps Tutorial 7117
Design of Experiments Tutorial 7122
Pareto Optimization Tutorial 7129
Use Existing Designs Tutorial 7136
Optimization with Macro Tutorial 7141
STAR-ICE 7144
ICE Mechanism 7146

Geometry Requirements....7148
Valves Motion....7153
Piston Motion....7156
Mesh Strategy....7158
Solution Analysis....7161
STAR-ICE Workflow....7165
STAR-ICE Troubleshooting....7185
STAR-ICE Reference....7186
Casting Simulation with STAR-Cast....7227
About the Casting Module....7228
Casting Workflow....7229
Selecting the Physics Models for the Melt....7231
Selecting the Physics Models for the Mold....7234
Selecting the Pure Thermal Model....7235
Selecting the Variable Density Compensation Model....7236
Working With the Casting Material Database....7237
Using Imported Casting Materials and Properties in STAR-CCM+....7283
Detecting Shrinkage-Related Defects....7288
Defect Prediction Using Criteria Functions....7297
Automated Time-Step Control....7310
Artificial Compressibility....7315
The Casting User Interface....7317
Casting Field Functions....7348
Casting Bibliography....7351

Theory....7352
Fundamental Equations....7353
Fluid Flow....7356
Equations of State....7358
Newtonian Fluids....7363
Non-Newtonian Fluids....7364
Boundary Conditions....7375
Gravity....7385
Fluid Flow Bibliography....7387
Numerical Flow Solution....7391
Finite Volume Discretization....7392
Finite Element Method for Viscoelastic Fluid Flow....7424
Numerical Flow Solution Bibliography....7429
~Moving Reference Frames~....7434
Turbulence....7435
Velocity Gradient and Invariants....7436
Non-Dimensional Quantities....7437
Reynolds-Averaged Navier-Stokes (RANS) Turbulence Models....7439
Scale-Resolving Simulations....7489
Turbulent Time and Length Scales....7508
Turbulence Bibliography....7510
~Transition~....7515
Heat Transfer....7516
Conduction....7518
Convection....7519

Thermal Radiation....7523
Solar Radiation....7549
Heat Transfer Bibliography....7553
Porous Media....7555
Superficial Velocity Formulation....7557
Physical Velocity Formulation....7559
Porous Baffle....7561
Porous Media Bibliography....7563
Species and Passive Scalars....7564
Species Transport....7565
Multi-Component Mixture Material Properties....7568
Passive Scalars Transport....7571
Eulerian Multiphase Flow....7573
Multiphase Segregated Flow....7576
Volume of Fluid Method....7715
VOF Bibliography....7743
Fluid Film....7746
Lagrangian Multiphase Flow....7766
Particle Equations of Motion....7769
Particle Heat and Mass Transfer....7779
Two-Way Coupling with the Continuous Phase....7784
Particle Injection....7787
Sprays and Droplets....7790
Particle Porosity....7828
Discrete Element Method....7829
Erosion....7845
Passive Scalar Interactions....7850
Solution Methodology....7853
Lagrangian Multiphase Bibliography....7856
Reacting Flow....7861
Basic Definitions....7865
Governing Equations for Reacting Flows....7870
Chemical Kinetics....7871
Reactions—Species Transport....7880
Gas Phase Combustion—Flamelet Models....7894
Fluid-Fluid Interphase Reactions....7910
Surface Chemistry....7911
Ignitors....7914
NOx....7917
Soot....7925
Particle Reactions....7934
Polymerization....7939
Reacting Flow Nomenclature....7943
Reacting Flow Bibliography....7949
Electrochemistry 7954
Transport of Electrochemical Reaction Components....7955
Conversion of Electrochemical Reaction Components....7964
Model Simplifications / Engineering Approaches....7975
Electrochemistry Bibliography....7978
Electrochemistry Nomenclature....7979
Electromagnetism....7984

Constitutive Relations....7985
Potential Formulation of the Governing Equations....7988
Harmonic Time Dependence....7991
Electrostatic Potential Model....7993
Electrodynamic Potential Models....7994
Magnetic Vector Potential Models....7998
Ohmic Heating....8007
Magnetohydrodynamics (MHD)....8008
Electromagnetism Bibliography....8010
Solid Mechanics....8011
Displacement, Stress, Strain....8012
Governing Equations....8019
Constitutive Laws....8022
Finite Element Discretization....8025
Rayleigh Damping....8028
Static and Dynamic Solutions....8031
Bibliography....8035
Aeroacoustics....8036
Broadband Noise Source Models....8038
Acoustic Wave Model....8050
Acoustic Suppression Zone Model....8056
Ffowcs Williams-Hawkings Model....8058
Aeroacoustics Bibliography....8065
Finite Element Method....8068
Shape Functions....8070
Integration Rules....8077
Sparse Direct Solver....8078
Finite Element Bibliography....8079
~Mesh Motion~....8081
Morphing....8082
Rotating Flow....8085
Virtual Disk Model....8086
Fan Modeling....8103
~Harmonic Balance~....8110
Adjoint....8111
Sensitivity with Respect to Design Parameters....8113
Sensitivity with Respect to Boundary Parameters....8120
Surface Sensitivity....8121
Adjoint Error Estimation....8124
Vector and Tensor Algebra....8125

Tutorials....8132
Using Tutorial Macros and Files....8134
Downloading the Tutorial Files from the Customer Success Portal....8135
Understanding the Directory Structure and Naming Conventions....8136
Playing the Macro....8137
Using the Final Simulation Files....8139
Introduction....8141
Starting a STAR-CCM+ Simulation....8143
Saving and Naming a Simulation....8149

Importing the Geometry.... 8150
Visualizing the Imported Geometry.... 8151
Defining Boundary Surfaces.... 8153
Assigning Parts to Regions.... 8160
Setting Boundary Types.... 8162
Mesh Generation.... 8167
Selecting the Physics Models.... 8175
Setting Initial Conditions.... 8181
Defining the Region Continuum.... 8182
Setting Boundary Conditions and Values.... 8183
Setting Solver Parameters and Stopping Criteria.... 8187
Visualizing the Solution.... 8188
Monitoring Simulation Progress.... 8193
Running the Simulation.... 8197
Adjusting Solver Parameters and Continuing.... 8198
Visualizing the Results.... 8202
Plotting Data on a Slice.... 8211
Adding Streamlines.... 8218
Closing and Reopening the Simulation.... 8225
Summary.... 8226

Foundation Tutorials.... 8227
3D-CAD.... 8229
Geometry Parts.... 8269
Mesh.... 8275
Physics - Flow.... 8293
Usability.... 8299

Geometry.... 8308
3D-CAD: Cyclone Separator.... 8309
Geometry Parts: CAD Assembly.... 8358

Mesh.... 8396
Surface Wrapper: Intake Manifold.... 8397
Surface Repair Tools Brake Assembly.... 8438
Parts-Based Meshing External Aerodynamics.... 8468
Mesh Operations: Imported Control Valve.... 8486
Meshing: Multi-Part Heat Exchanger.... 8501
Directed Meshing: Electrical Machine.... 8512

Incompressible Flow.... 8528
Steady Flow: Lid-Driven Cavity Flow.... 8529
Steady Flow: Channel Flow with Multiple Meshes.... 8543
Steady Flow: Laminar and Turbulent in an S-Bend.... 8569
Steady Flow: Backward Facing Step.... 8604
Adjoint Flow Solver: External Flow over a Dual Element Wing.... 8636
Adjoint Flow: Wing Shape Optimization.... 8649
Porous Resistance: Isotropic Media.... 8659
Porous Resistance: Orthotropic Media.... 8685
Solution Recording and Playback: Vortex Shedding.... 8693
Generalized Newtonian Fluid: Flow in a Static Mixer.... 8714
Viscoelastic Flow: Basic Extrusion.... 8723

Compressible Flow.... 8736
Subsonic Flow: NACA-Type Intake.... 8737
Transonic Flow: RAE2822 Airfoil.... 8772

Transonic Flow: RAE2822 Airfoil Using Overset Mesh....8789
Adjoint Flow Solver: Flow through an S-Bend....8811
Turbo Wizard: Single Row....8825
Turbo Wizard: Multiple Rows....8847
Harmonic Balance: Single Stage Periodic Flow....8886
Heat Transfer and Radiation....8912
Conjugate Heat Transfer: Heated Fin Introduction....8913
Multi-Part Solid: Graphics Card Cooling....8928
Natural Convection: Concentric Cylinders Introduction....8955
Electronics Cooling Toolset: Natural Convection JEDEC....8966
Electronics Cooling Toolset: Graphics Card Cooling....8993
Dual Stream Heat Exchanger: Car Radiator....9011
Surface-to-Surface Radiation: Thermal Insulator....9029
Multiband Surface-to-Surface Radiation: Solar Collector....9050
Thermal Comfort Wizard: Single Occupant in a Cabin....9065
Multiphase Flow....9091
VOF: Gravity-Driven Flow....9092
VOF: Capillary Effects....9107
VOF: Cavitation....9124
VOF: Boiling....9142
VOF: Melting-Solidification....9164
Lagrangian: Particle-Laden Flow....9189
Lagrangian: Solid Particle Erosion....9219
Eulerian: Hibiki's Bubble Column....9261
Eulerian: Mixture Settling....9295
Eulerian: Bubble Formation in a Fluidized Bed....9315
Eulerian: Degassing Boundary....9333
Eulerian: Wall Boiling....9353
Eulerian: Conjugate Heat Transfer Wall Boiling....9394
Eulerian Multiple Flow Regimes: Pressurized Water Reactor....9426
Eulerian: Pipe Flow with Multi-Speed AMUSIG....9445
Fluid Film: Liquid Film Flow....9465
Fluid Film: Binary Liquid Film Flow with Evaporation and Edge Stripping....9487
Fluid Film - VOF: Rivulets on an Inclined Plane....9523
Dispersed Multiphase: Airfoil Icing....9545
Discrete Element Method....9562
DEM Particles in a Conveyor....9563
DEM Particle Settling....9592
Cylindrical Particles in a Rotating Drum....9622
Motion....9635
Moving Reference Frames: Rotating Fan....9636
Rigid Body Motion: Rotating Fan....9647
DFBI: Boat in Head Waves....9655
DFBI: Lifeboat with Overset Mesh....9677
Marine Resistance Prediction: KCS Hull with a Rudder....9703
Moving Reference Frames: Marine Propeller in Open Water....9740
Body Force Propeller Method: Marine Self-Propulsion....9759
Blade Element Method: Helicopter Rotor-Fuselage Interaction....9779
Morphing: Cylinder with Boundary Motion....9796
Reacting Flow....9812
3-Step Eddy Break-Up: Non-Premixed Propane....9813

Complex Chemistry: Methane-Air Jet Flame....9828
Spray Combustion: Jet A Fuel....9838
Steady Laminar Flamelet: Non-Premixed Hydrogen....9854
Surface Chemistry: Methane on Platinum Oxidation....9861
Reacting Channels: Steam Cracking Furnace....9871
Fire and Smoke Wizard: Steckler Room....9889

Solid Stress....9909
Linear Stress Analysis: Cantilever I Beam....9910
Plane Stress: Plate with Circular Hole in a Tensile Field....9930
Thermal Strain: Exhaust Manifold....9945
Fluid-Structure Interaction: Vibrating Pipe....9966

Aeroacoustics....9981
Broadband Models: Noise from a Cylinder (Preparation)....9982
DES and FW-H On-The-Fly: Noise from a Cylinder (Unsteady Analysis)....10004
Ffowcs Williams-Hawkings: Sound Propagation....10028
Signal Post-Processing: FFT and Wavenumber....10043
Acoustic Wave Modeling: Noise from a Cylinder....10057

Electromagnetism....10079
Ohmic Heating: Domestic Fuse....10080
Magnetic Vector Potential: Magnetic Flux....10104

Electrochemistry....10117
Electrochemistry: Electroplating....10118
Electrochemistry: Solid Oxide Fuel Cell....10130

Battery....10165
Battery Simulation Module: Cell Thermal Analysis....10166
Cylindrical Cells: Cell Thermal Analysis....10216
Li-Ion Battery Cell Model: Cell Electrochemistry Analysis....10256

Casting....10277
Basic Gravity Sand Casting....10278
STAR-Cast: Investment Casting....10304

Automation....10328
Simple Java Macros: Post-Processing Objects....10329
Intermediate Java Macros: Run Multiple Simulations....10356
Simulation Assistant: Internal Flow Assistant....10397

Coupling with CAE Codes....10439
STAR-CCM+ to STAR-CCM+ Coupling: Heat Transfer in a Chimney....10440
LMS Amesim Co-Simulation: 1D Coupling....10461
Abaqus File-Based Coupling: Exhaust Manifold....10473
Abaqus Co-Simulation: Thermal Coupling....10500
Abaqus Co-Simulation: Mechanical Coupling....10534
GT-SUITE Co-Simulation: 1D Coupling....10557
Co-Simulation API: Spindle Valve....10597

Analysis Methods....10612
Volume Rendering: Steckler Room....10613
Data Focus: Thermal Comfort....10628
Simulation History: Lagrangian DEM on an Auger....10638

Design Manager....10650
Optimizing a Static Mixer....10651
Design Sweep of a Static Mixer....10664

STAR-ICE....10669

Gasoline Engine..... 10670

Getting Started

STAR-CCM+ provides the world's most comprehensive engineering physics simulation inside a single integrated package. Much more than just a CFD solver, STAR-CCM+ is an entire engineering process for solving problems involving flow (of fluids or solids), heat transfer, and stress. It provides a suite of integrated components that combine to produce a powerful package that can address a wide variety of modeling needs.

Contents:

Navigating Help
Using Online Help
STAR-CCM+ Simulation Process
STAR-CCM+ Licensing
Getting Support
Suggesting Enhancements to STAR-CCM+
STAR-CCM+ Default Options
Resetting STAR-CCM+ Default Options
Controlling Graphics Performance
Troubleshooting Graphics
Credits

Navigating Help

New Features

Quick links to documentation for the latest features

How to use STAR-CCM+

Launch STAR-CCM+; understand the workflow; learn about the interface.

Working with Geometry

Learn how to create geometry in 3D-CAD, import geometry, manipulate geometry, and repair surfaces before meshing.

Simulation Topology

Transfer the geometry model to the CFD model using regions, boundaries and interfaces.

Generate Mesh

Understand mesh continua, mesh sizing, models for surface and volume meshing, and how to refine in chosen locations.

Modeling Physics

Explore all the physics models and options, accessed through physics continua.

Running the Simulation

Set initial conditions and solvers, apply stopping criteria, and run the simulation from the GUI or in batch.

Analyze the Solution

Create derived parts, reports and scenes; plot XY data; record solutions and play back later.

Using Tools

Learn how to create your own field functions, define local coordinate systems and reference frames, and work with tables and units.

Wizard Guide

STAR-CCM+ provides several wizards: Turbo, Fire and Smoke, Thermal Comfort, Wing Mesher, and SPEED Import.

Interacting with CAD/CAE

Integrate a wide range of CAD/CAE tools into your STAR-CCM+ simulation.

Working with CAD Clients

Embedded solutions to let you define your STAR-CCM+ simulation within your CAD interface.

Tutorials

Get step-by-step instructions on setting up many different cases with our models.

STAR-View+

A freely distributable and independent application for viewing 3D scenes from STAR-CCM+.

Getting Support

Access the Knowledge Base for extra help; prepare queries for CD-adapco Support Engineers.

Any and all Siemens Product Lifecycle Management Software Inc. brand, product, service and feature names, logos and slogans are registered trademarks or trademarks of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries. All other brand, product, service and feature names or trademarks are the property of their respective owners.

Using Online Help

The online help is accessed through the **Help** menu for browsing, and through the `<F1>` key for context-sensitive information.

The STAR-CCM+ User Guide is provided in two formats, HTML and PDF. The PDF file can be found within the `doc` directory of your STAR-CCM+ installation.

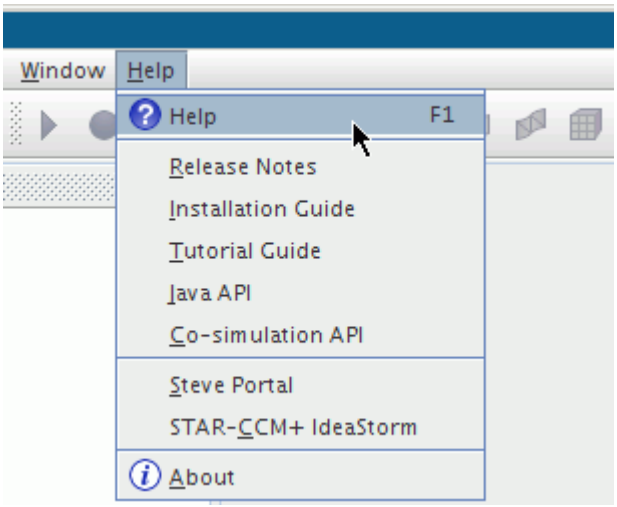
If the **Help** menu action fails to access the locally installed User Guide, STAR-CCM+ accesses the copy on the customer success portal. If you are prompted, log in to the customer success portal at https://thesteveportal.plm.automation.siemens.com.

Contents:

Browsing the Online Help
Accessing Context Sensitive Help
Using the Navigation Frame
Using the Toolbar Frame
Changing the Default Browser
Understanding Typographic Conventions

Browsing the Online Help

To access the online Help for browsing, select **Help** > **Help** from the top menus, or press the `<F1>` key.

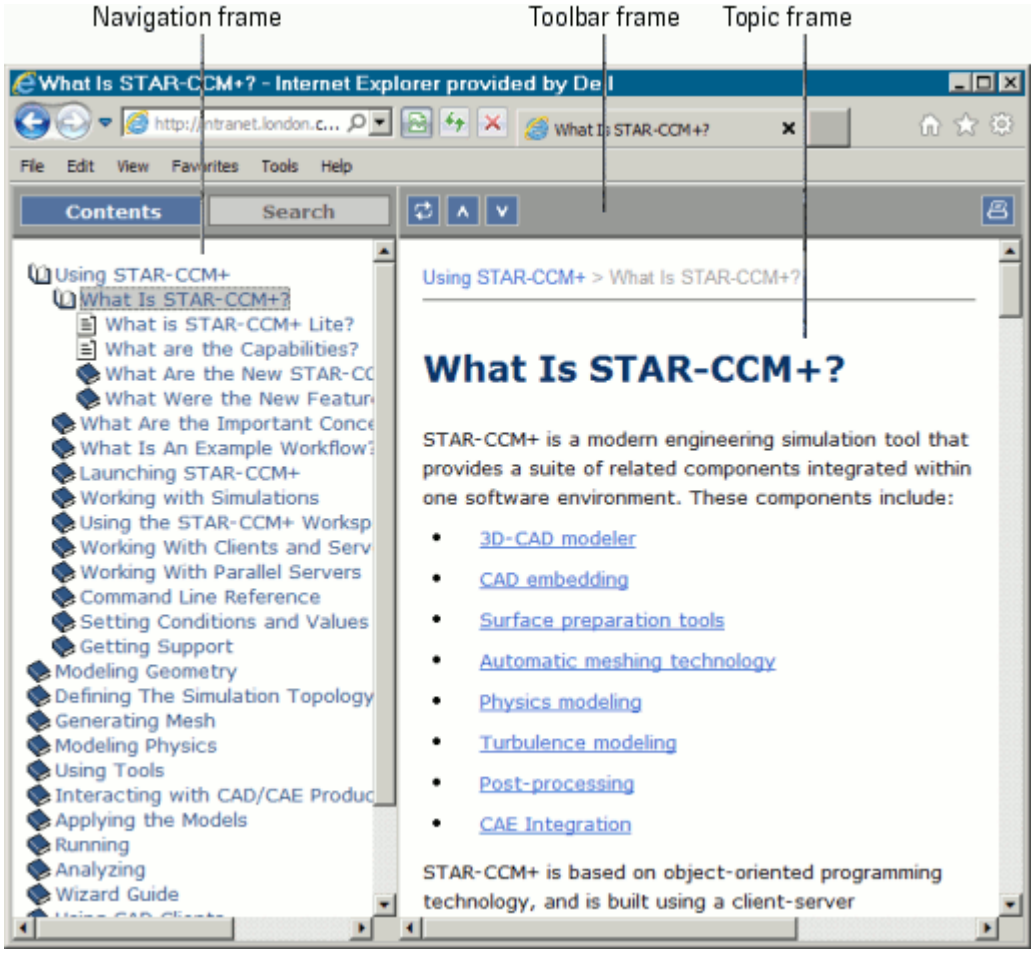


STAR-CCM+ will launch an HTML browser at the title page of the online Help. You can also jump directly to other parts of the online Help using the same menu:

- New Features
- Tutorial Guide
- Wizard Guide

The Help system is based entirely on JavaScript and displays help topics with the frameset view, which divides the browser window into three areas:

- A navigation frame on the left that displays the Contents tab (with a Table of Contents pane) and the Search tab
- A toolbar frame across the top that displays a set of buttons to navigate around the Help system
- A topic frame on the right to display the actual Help topics included in the Help system



This view gives the most control and is used to display Help requested from the Help menu. It will also display if you open `$STAR_HOME/doc/online/index.html` directly into your browser.

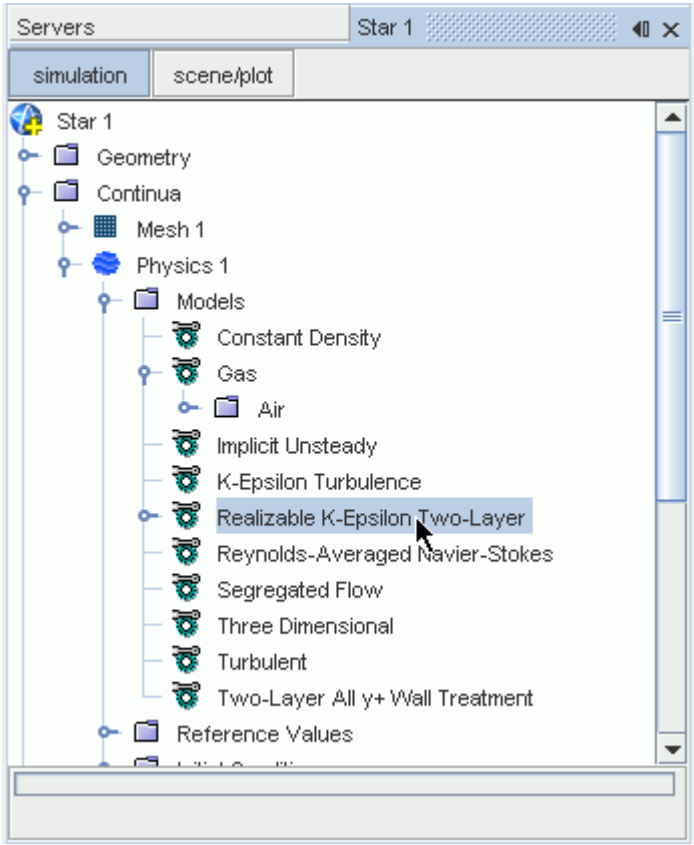
You can also read useful information about:

- The navigation frame
- The toolbar frame
- Changing the default browser
- Typographic conventions used in this documentation

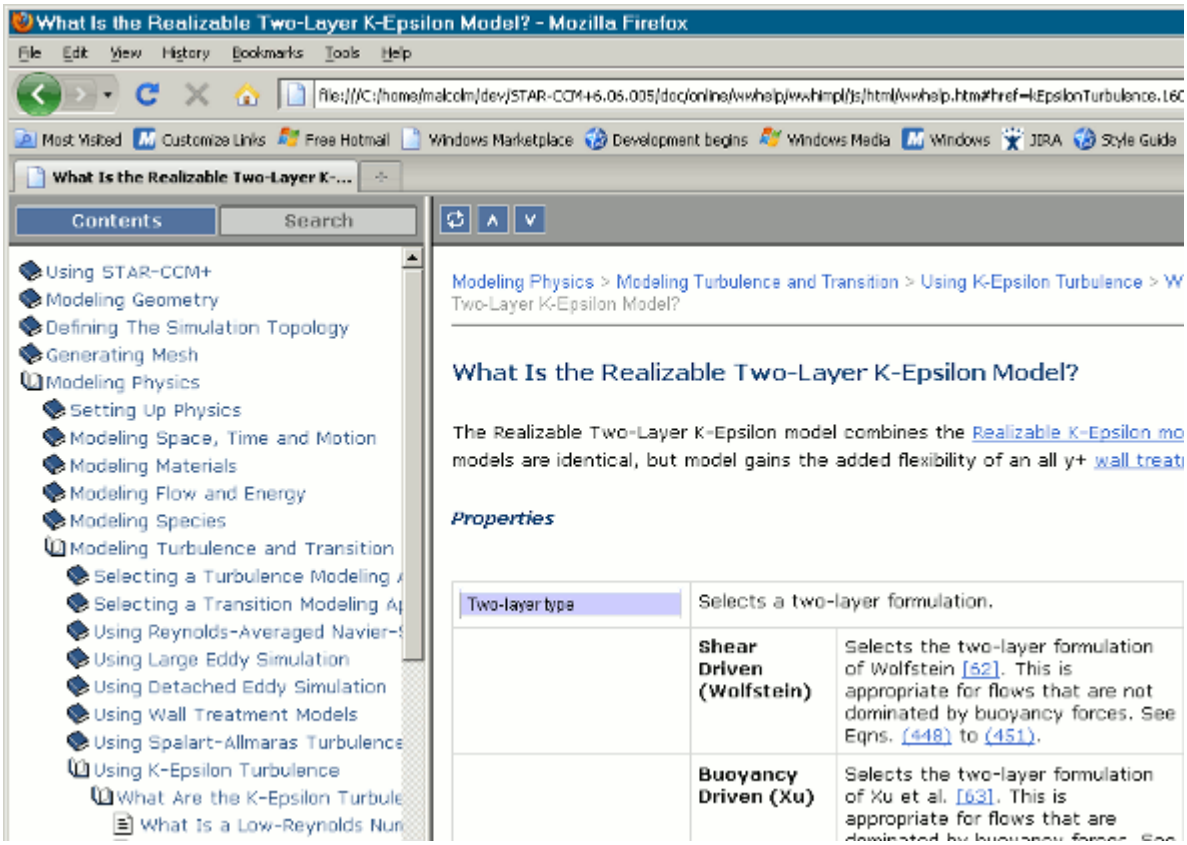
Accessing Context Sensitive Help

You can jump directly to help topics that relate to specific objects in the simulation tree by pressing `<F1>` while the object is selected.

For example, you could press `<F1>` on the **Realizable K-Epsilon Two-Layer** model node within the **Models** manager of a physics continuum.



This will launch a browser showing the Help system open at the location of the documentation for the Realizable K-Epsilon Two-Layer model.

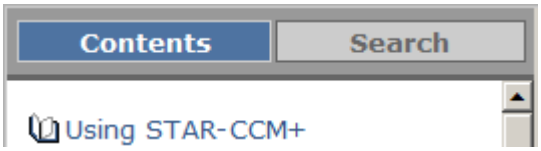

What Is the Realizable Two-Layer K-Epsilon Model? - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Most Visited Customize Links Free Hotmail Windows Marketplace Development begins Windows Media Windows JIRA Style Guide
What Is the Realizable Two-Layer K-...
Contents
Search
Using STAR-CCM+
Modeling Geometry
Defining The Simulation Topology
Generating Mesh
Modeling Physics
Setting Up Physics
Modeling Space, Time and Motion
Modeling Materials
Modeling Flow and Energy
Modeling Species
Modeling Turbulence and Transition
Using Large Eddy Simulation
Using Detached Eddy Simulation
Using Wall Treatment Models
Using K-Epsilon Turbulence
Modeling Physics > Modeling Turbulence and Transition > Using K-Epsilon Turbulence
What Is the Realizable Two-Layer K-Epsilon Model?
Properties
Two-layer type
Selects a two-layer formulation.
Shear Driven (Wolfstein)
Selects the two-layer formulation of Wolfstein [62]. This is appropriate for flows that are not dominated by buoyancy forces. See Eqns. (448) to (451).
Buoyancy Driven (Xu)
Selects the two-layer formulation of Xu et al. [63]. This is appropriate for flows that are

Using the Navigation Frame

By default, the navigation frame displays two tabs:

- Contents
- Search

These tabs provide the core navigational features in the Help system.



In addition, you can make the best use of space by resizing.

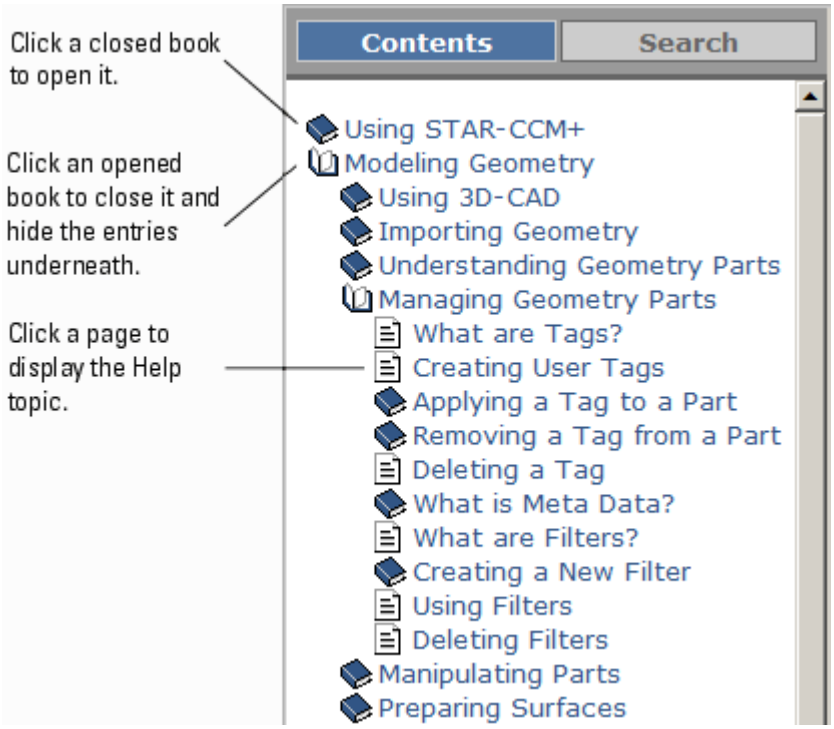
Contents:

Navigating Contents
Searching the Help System
Resizing the Navigation Frame

Navigating Contents

The Contents tab displays the table of contents in the form of an expandable/collapsible tree view.

Closed book icons represent TOC entries that have subentries. Click a closed book to open it and see its contents. When you expand a book, the closed book icon is replaced by an open book icon. You can click the book again to close it. The page icons represent individual Help topics. Click a page icon or the tree text to display the contents of that page in the Contents pane. Clicking the text of an element with subentries will open the folder and display the selected page.

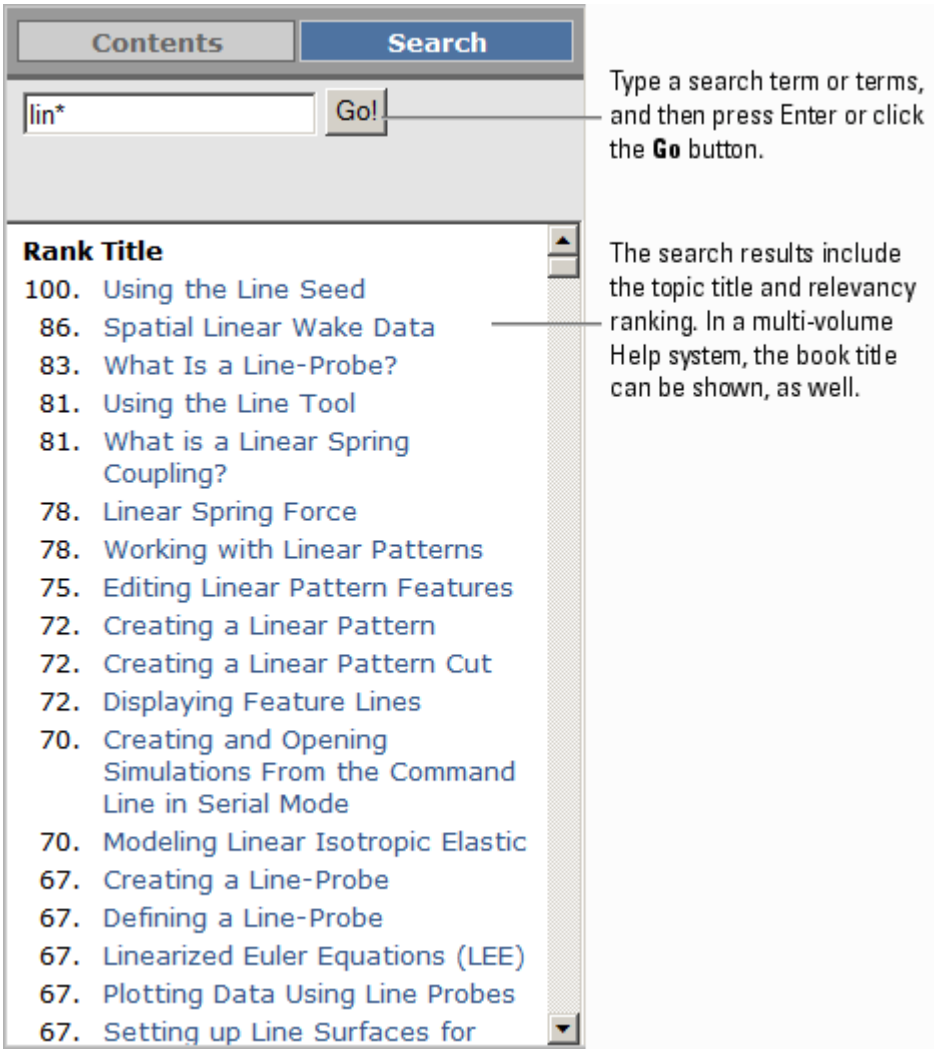


Searching the Help System

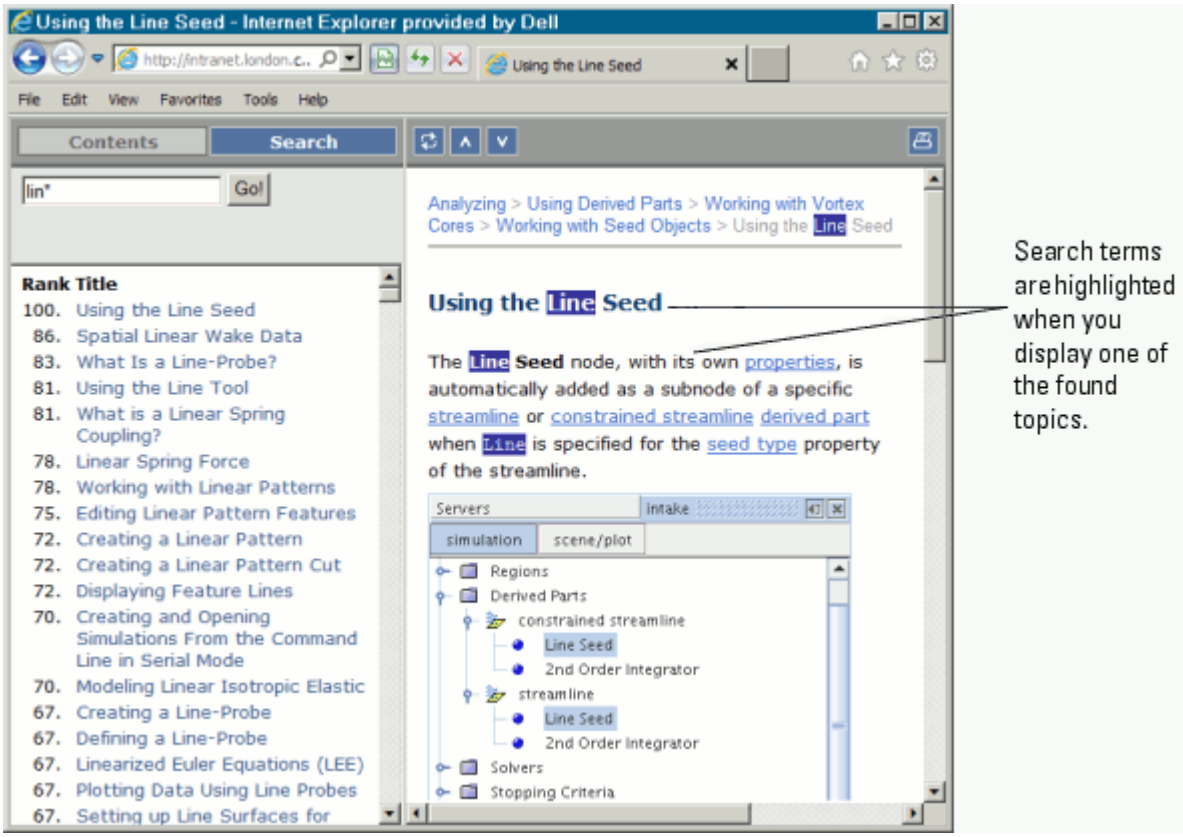
Using the *Search* tab, you can search the full text of the Help system.

Type the word or phrase to search for, and then press <`Enter`> or click **Go**. The search is case insensitive, and by default searches on whole words. So searching for “lin” would not find “Linux”. Use an asterisk to perform partial word searches, so for example, use “lin*” to find “Linux” as well as all other words beginning with “lin”.

The *Search* tab displays a list of all the topics in the Help system that contain the word or phrase you entered. If you search for multiple words, the search finds Help topics that contain all the words you entered. The topics found are ranked in order of relevance. The higher the ranking, the more likely the topic includes the word or phrase you searched for.



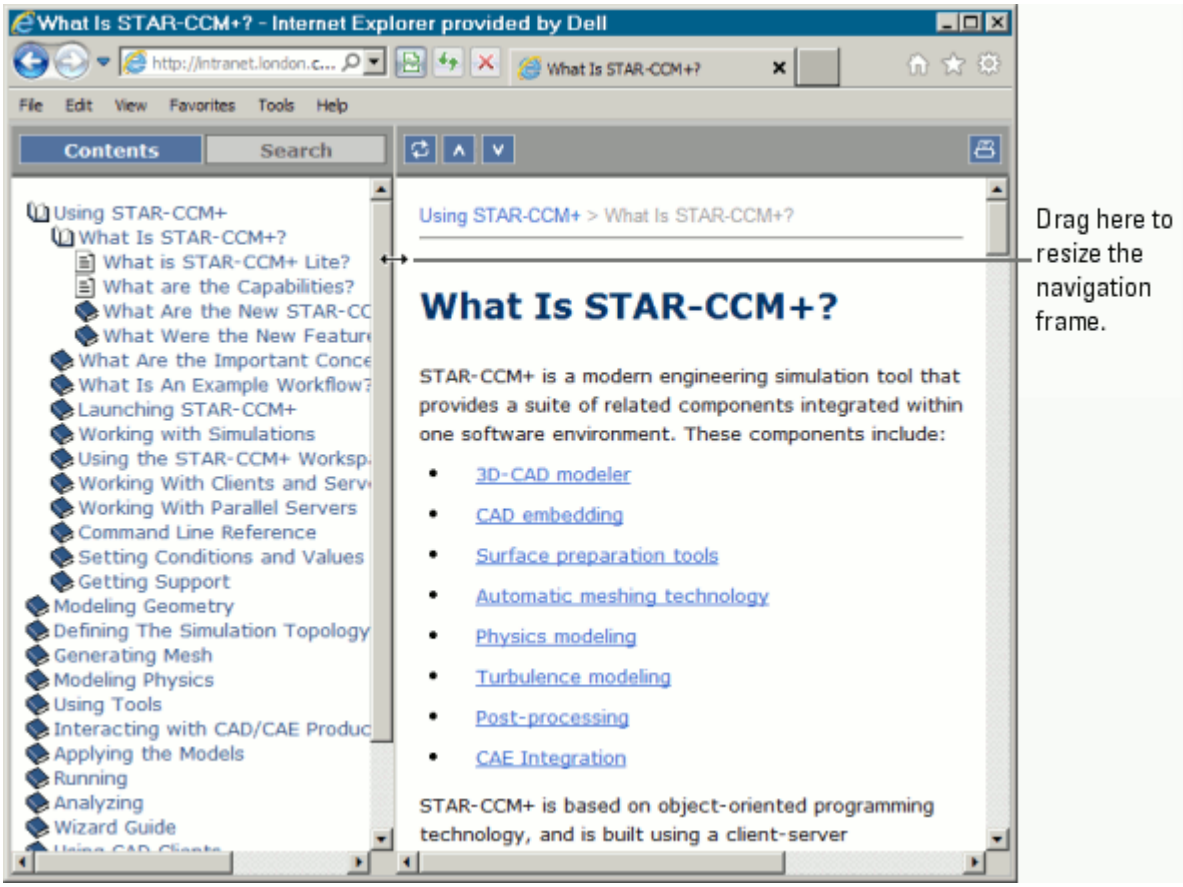
When you click one of the topics found by search, the page is loaded into the topic frame. If you are using Internet Explorer on Microsoft Windows, each occurrence of the search term(s) is/are highlighted as shown in the following screenshot. (The search highlighting feature does not work in other browser environments.)



Resizing the Navigation Frame

You can resize the navigation frame by dragging the vertical line that separates the navigation frame from the topic frame.

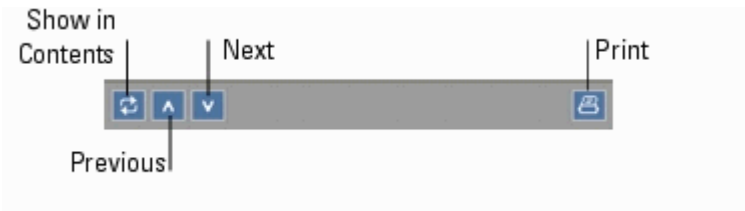
As you reduce the width of the navigation frame, the buttons will reduce in size until they start being hidden after they reach a minimum width, starting with the right-hand one.



By default, the width of the navigation frame is set to 300 pixels.

Using the Toolbar Frame

The toolbar frame displays a set of buttons to give you additional functionality, as shown below.



Show in Contents

Locates the currently displayed topic in the *Contents* tab. When you click `Show in Contents`, Help highlights the entry in the *Contents* tab that corresponds to the currently displayed topic. If the *Contents* tab is not currently the front-most tab, it will be brought to the front.

In the topic-only view, the **Show in Contents** button uses a different icon. When you click the **Show in Contents** button in the topic-only view, the display switches to frameset view, and the navigation frame is displayed.

Previous

Displays the topic that precedes the currently displayed topic in book sequence. This is not the same as the browser's back button.

Next

Displays the topic that follows the currently displayed topic in book sequence. This is not the same as the browser's forward button.

Print

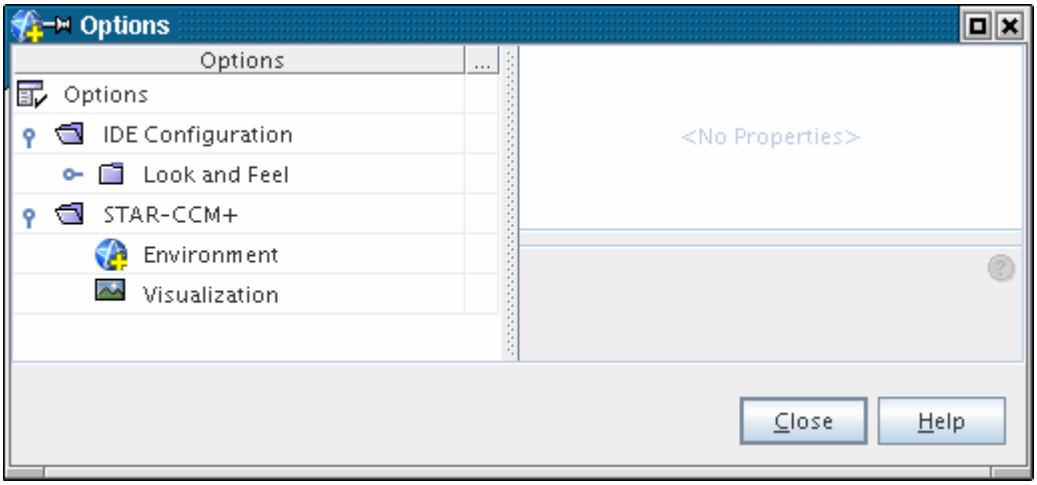
Prints the currently displayed topic.

Changing the Default Browser

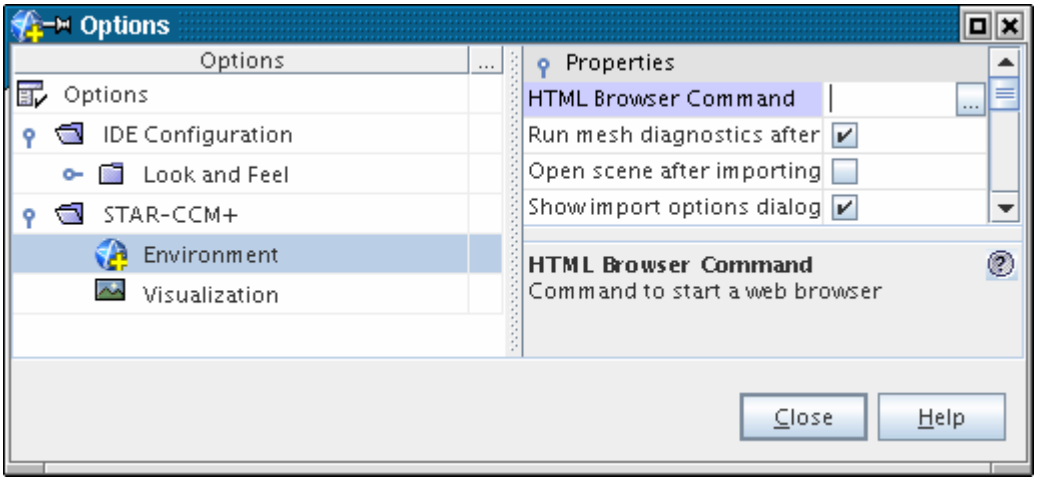
You change the command that STAR-CCM+ uses to launch a browser.

This can be useful, for example, if the `<F1>` keyboard command fails to bring up the context-sensitive Help.

Select **Tools** > **Options**. This activates the *Options* dialog.



Open the **STAR-CCM+** node, then select the **Environment** node to access its properties, which include the *HTML Browser Command* property.



Select that property and enter the full path of the HTML browser into the text field.

Click **Close**. STAR-CCM+ will use that new path when it launches the online Help.

Understanding Typographic Conventions

This printed documentation online Help system uses the following conventions:

Convention	What it Means
BoldArial	Items you can select in the STAR-CCM+ interface, including buttons and menu items.
Italic	Components in the STAR-CCM+ interface such as windows, dialogs, and menus.
>	The greater-than sign indicates one of the following: • A menu choice. For example, “select `File > Open`” means “click the **File** menu, and then click **Open**”. • A node of a tree that has been opened. For example, "select the **Scenes** > **Part Scene** > **Part Displayer** node" means "expand the Scenes node, then expand the **Part Scene** node under that and finally click on the **Part Displayer** node under that".
`Code`	A fixed space font indicates text that you would type, such as on the command line or in macros.
`<Enter>`	Fixed space font between angled brackets indicates a single keystroke on the keyboard.

STAR-CCM+ Simulation Process

STAR-CCM+ provides you with a simulation tool that allows you to test solution designs under virtual conditions. Although STAR-CCM+ supports a wide range of modeling scenarios, there is a general workflow that you can follow in most situations.

When defining a simulation in STAR-CCM+, you initially work with objects that divide into three broad categories:

1. Geometry-level objects. Geometry-level objects describe the spatial construction of the design you are testing. The main geometry objects are geometry parts, part surfaces, part curves, and part contacts. These objects reside beneath the **Geometry** node in the STAR-CCM+ interface.
2. Region-level objects. These objects support the physics, initial conditions, boundary conditions, and other sources that you apply to the materials under test. The main region-level objects are regions, boundaries, and interfaces. These objects reside beneath the **Regions** and **Interfaces** nodes.
3. Physics-model objects. Physics-model objects are defined within physics continua. A physics continuum represents a material, which can be multi-component or multiphase, and the physics models that act on that material.

In general practice, you create and order the geometry objects before mapping them to corresponding objects within the region hierarchy. However, in order to prepare an appropriate arrangement of geometry objects, you must first plan the arrangement of region objects. Although STAR-CCM+ supports a wide range of possible arrangements, there are some broad categories that you can consider in making an initial plan.

General Workflow

STAR-CCM+ applies physics models to fluid and solid materials that belong to physics continua. A physics continuum is an object in the STAR-CCM+ simulation hierarchy that connects fluid or solid materials to the models that govern their behavior. Physics continua in turn apply to regions

The most general view of the STAR-CCM+ workflow is depicted in the illustration below.

Contents:

Simulation Categories

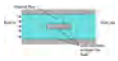
Simulation Categories

Before starting a simulation project in STAR-CCM+, decide how many regions you require to simulate the engineering scenario, and how those regions connect. For some scenarios, you are best to have two simulations that connect through the co-simulation framework.

To decide which simulation category applies in your analysis, prepare a sketch that identifies the materials, physics, and conditions that are present in the real situation. Then, scan the engineering examples in the sections below to identify one that is similar to your analysis. As a general rule for optimal performance, choose the layout that minimises the number of regions, boundaries, and interfaces in the simulation.

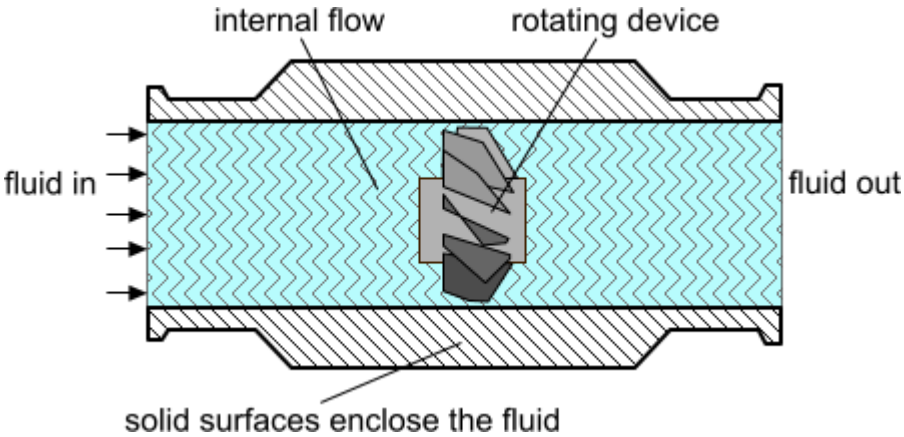
Single Material with Same Physics Throughout

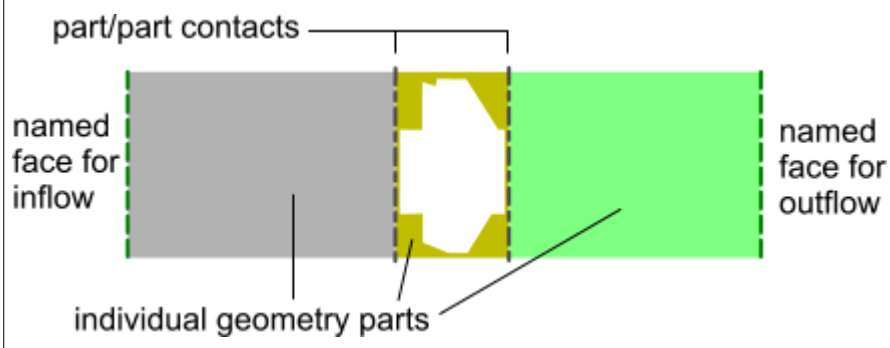
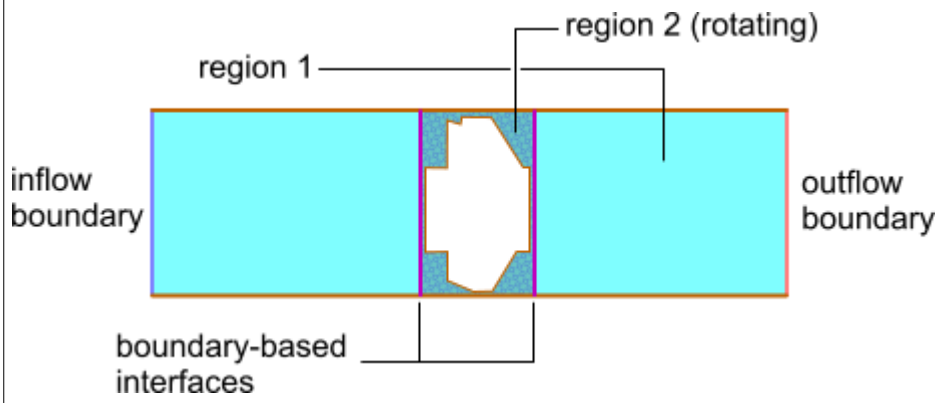
For a flow of a single fluid, or heat conduction through a single solid, you require only one geometry part that maps to one region.

Engineering Example	
Geometry Objects	
Region Objects	
Tutorial Example	Tutorial Guide > Incompressible Flow > Steady Flow: Backward Facing Step

Single Material with Modified Physics in Specific Locations

In this class of problems, some additional physics models are applied to specific areas or volumes within the simulation domain. Typical examples are moving parts and porous media. These sections must have their own region in which the additional physics applies. The same continuum applies to all regions within the domain.

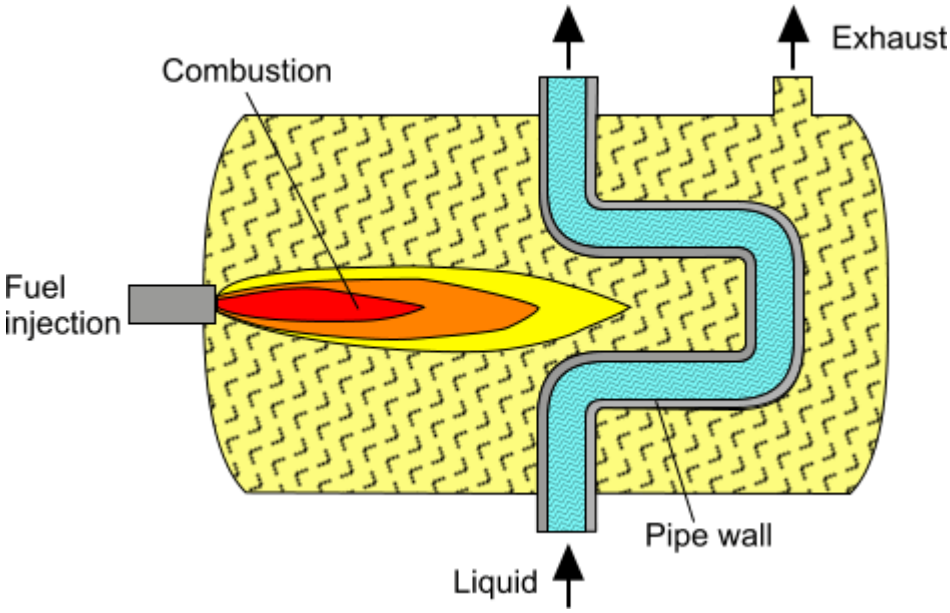
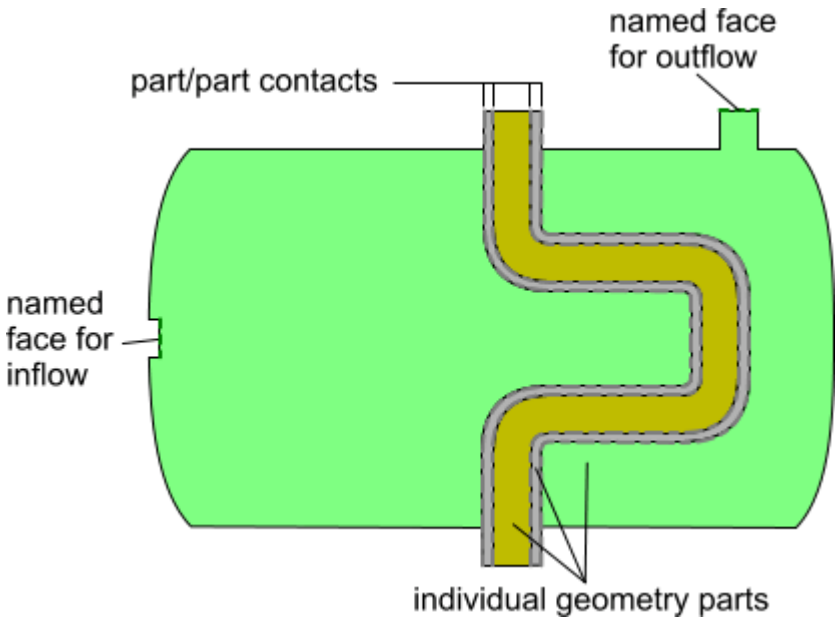
Engineering Example	

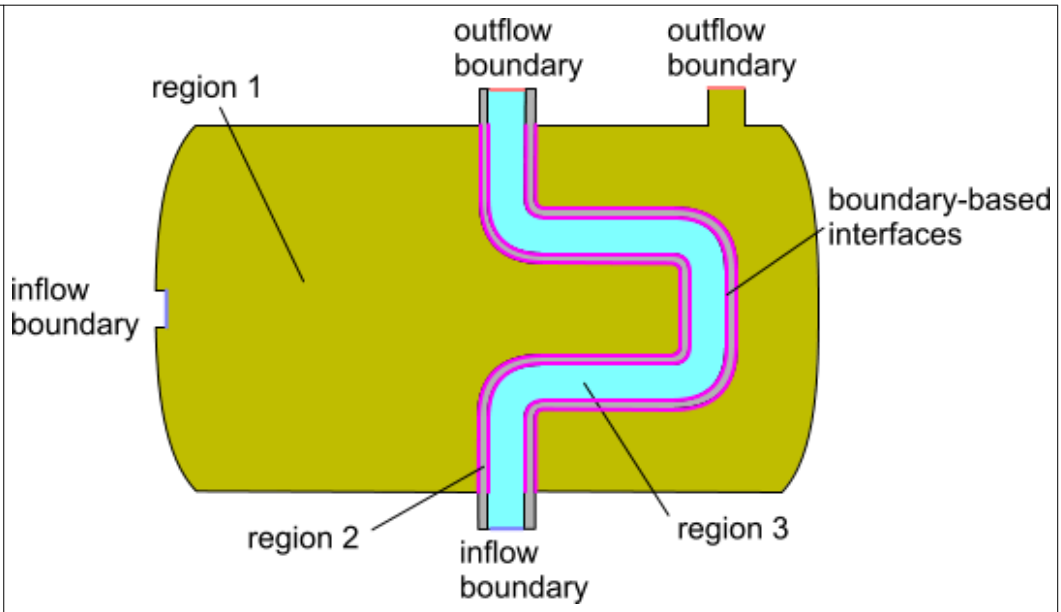
Geometry Objects	
Region Objects	
Tutorial Example	Tutorial Guide > Motion > Rigid Body Motion: Rotating Fan

Multiple Connected Materials (Low Number of Parts)

When you want to model heat passing through a combination of fluid and solid materials, you must prepare for interfaces between solid and fluid regions. If the number of parts is low, you can use boundary-based interfaces where one pair of boundaries are matched per interface.

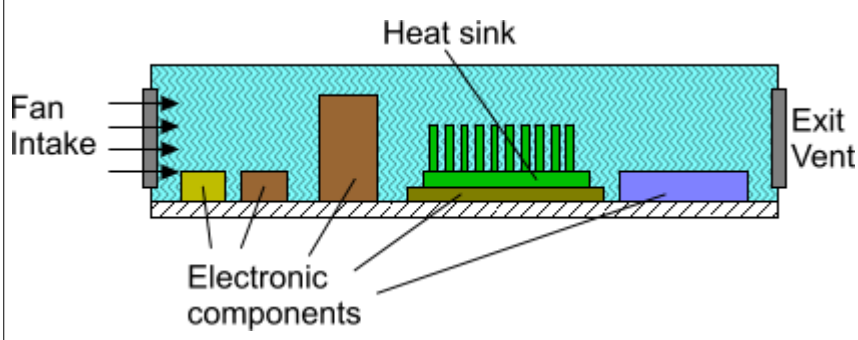
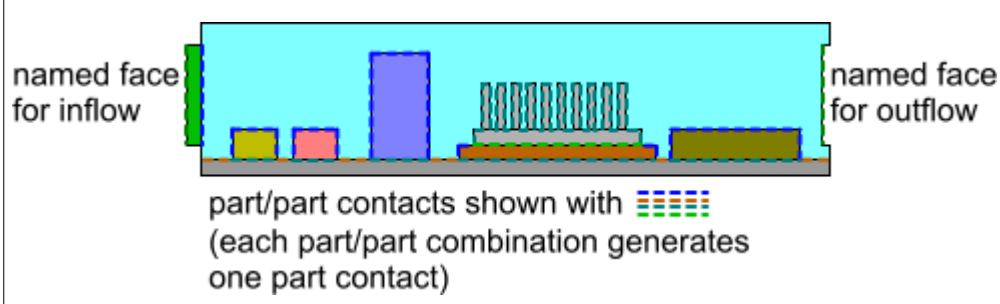
Engineering Example	

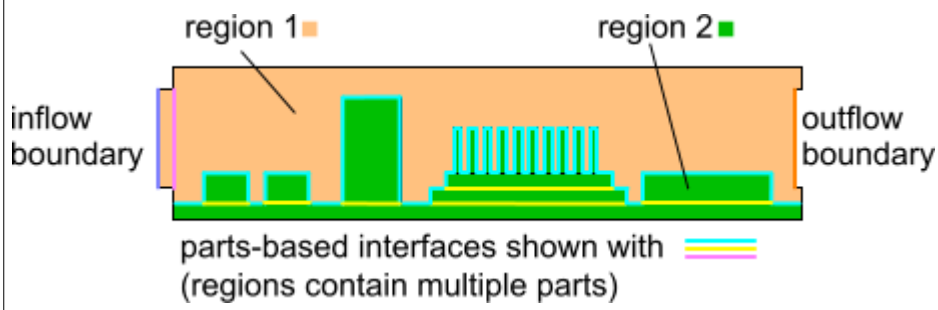
	
Geometry Objects	
Region Objects	

	
Tutorial Example	Tutorial Guide > Heat Transfer and Radiation > Conjugate Heat Transfer: Heated Fin Introduction

Multiple Connected Materials (Medium to High Number of Parts)

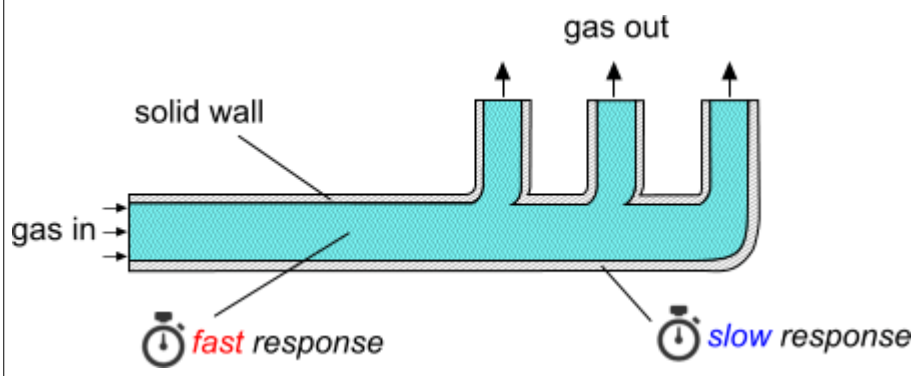
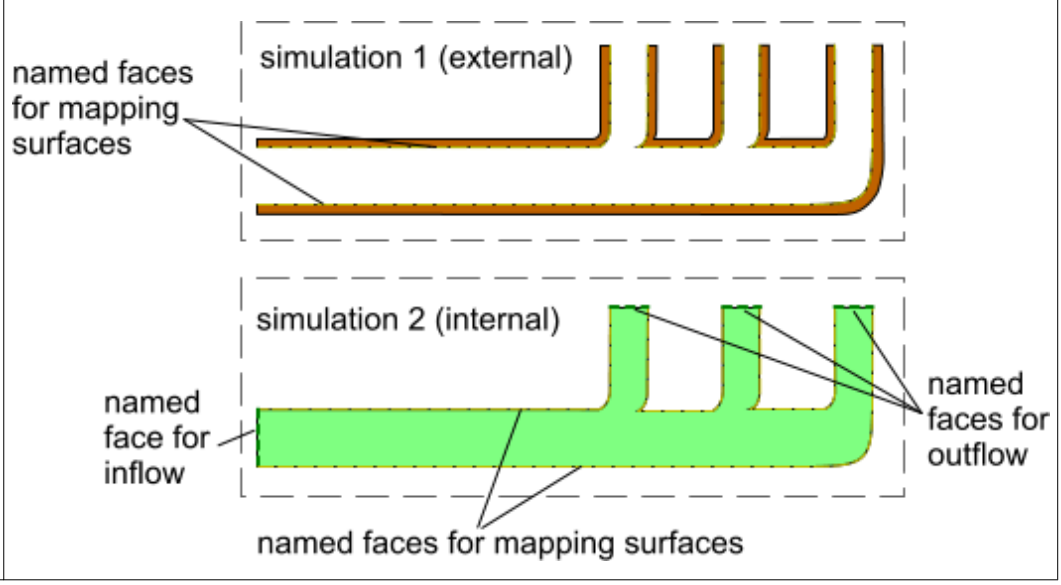
When you are modeling a scenario that contains multiple materials in contact, such as heat transfer across a high number of parts in contact with the same fluid, you can create interfaces based on part contacts rather than regions. This approach gives significant performance benefits as the overall number of interfaces is low.

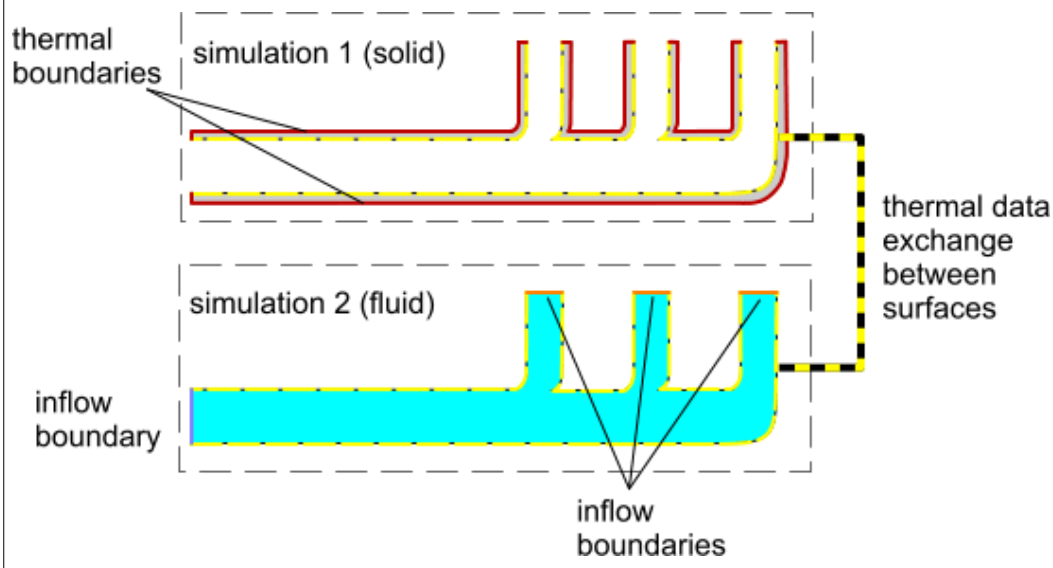
Engineering Example	
Geometry Objects	

Region Objects	
Tutorial Example	Tutorial Guide > Heat Transfer and Radiation > Multi-Part Solid: Graphics Card Cooling

Simulations with Multiple Timescales or Incompatible Phenomena (Co-Simulation Method)

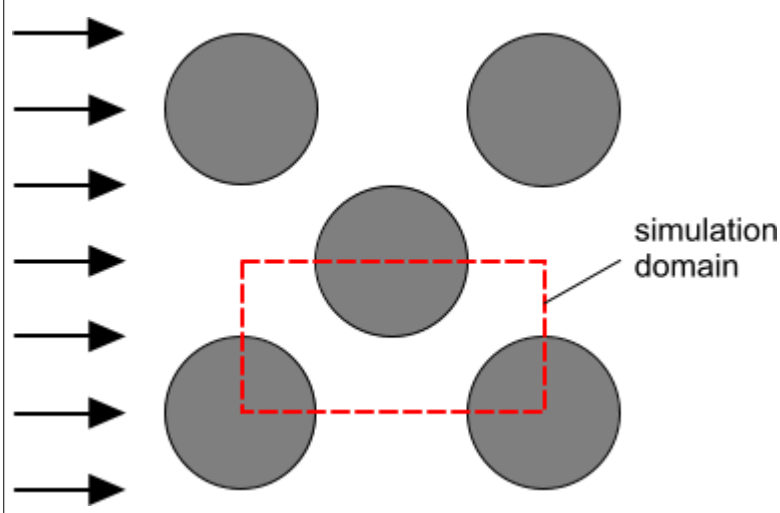
For scenarios where the time scales of thermal conduction in a solid are much longer than the time scales in the fluid, or where different solver techniques are required in one part of the scenario, STAR-CCM+ offers the co-simulation method. In this method, you define two separate simulations, each with their own region-level layouts, and configure the simulations to exchange data with each other during runtime.

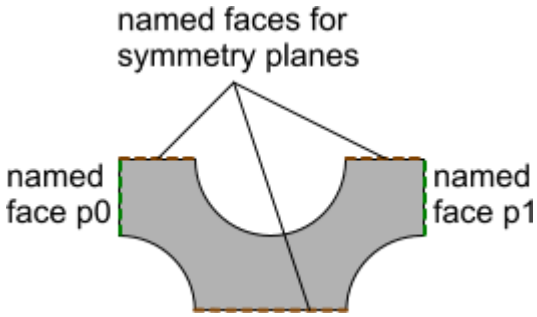
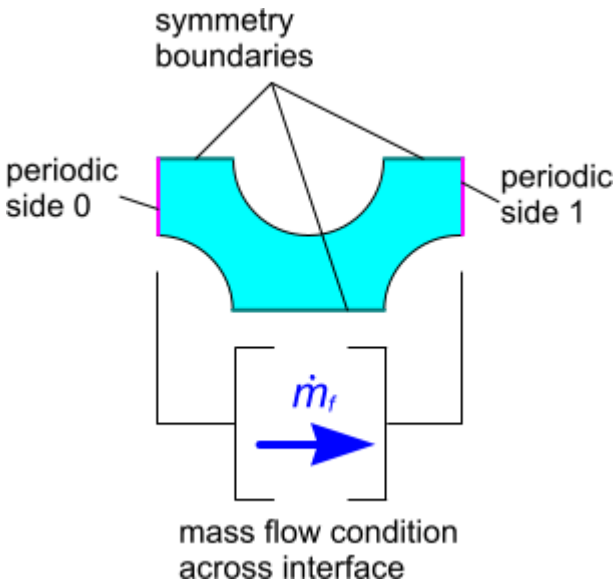
Engineering Example	
Geometry Objects	

Region Objects	
Tutorial Example	Tutorial Guide > Coupling with CAE Codes > STAR-CCM+ to STAR-CCM+ Coupling: Heat Transfer in a Chimney

Repeating Geometries and Cyclic Conditions

In some scenarios, repeating flow patterns are observed across similar geometries, or along a tube. In these situations, you can simplify the problem definition by modeling only part of the scenario and applying a periodic interface across two surfaces on which you expect to see identical flow profiles.

Engineering Example	
Geometry Objects	

	
Region Objects	
Tutorial Example	Tutorial Guide > Compressible Flow > Harmonic Balance: Single Stage Periodic Flow

STAR-CCM+ Licensing

STAR-CCM+ is licensed software that operates according to license feature keys that it checks out from the license management system.

STAR-CCM+ has multiple product configurations. Each product configuration may require a different type of license, depending on what add-on modules you want to use.

Product Configurations

The following STAR-CCM+ product configurations are available.

Product Configurations

Product Configuration	License Feature	Description
STAR-CCM+	ccmpsuite	The STAR-CCM+ product license contains all the standard features in STAR-CCM+. The default `ccmpsuite` feature license is used for this configuration.
STAR-CCM+ Lite	ccmplite	STAR-CCM+ Lite is a version of STAR-CCM+ that provides a limited set of features. STAR-CCM+ Lite configuration uses the `ccmplite` feature license. See Capabilities of STAR-CCM+ Lite.
STAR-CCM+ Power Session	ccmppower	STAR-CCM+ Power provides all the features of STAR-CCM+ with the added benefit of running one job on an unlimited number of cores. When this type of license is used, other licenses such as `ccmpsuite` remain inactive. If you start a job using the Power Session, only one copy of the `ccmppower` license is checked out.
STAR-CCM+ Power-On-Demand	ccmppower	The Power-On-Demand (PoD) session is a licensing model that enables you to use a STAR-CCM+ Power Session from a Siemens PLM Software public license server rather than a license server on your own network. For more information, refer to the Power-On-Demand section.

A comparison table listing the STAR-CCM+ add-ons is provided below. A short description of what the configuration includes, along with any pre-requisites is also provided for each configuration. For more information, see STAR-CCM+ Add-on Modules.

STAR-CCM+ Add-on	Description	Pre-requisites	Feature LicenseName
STAR-CAT5	Enables the STAR-CAT5 client	STAR-CCM+ or STAR-CCM+ Lite and CATIA V5 with ME2 or MD2 + GPS configuration license	star-cat5
STAR-Pro/E	Enables the STAR-Pro/E client	STAR-CCM+ or STAR-CCM+ Lite and Pro/E license	star-proe
STAR-Creo	Enables the STAR-Creo client	STAR-CCM+ or STAR-CCM+ Lite and Creo or Pro/E license	star-proe
STAR-NX	Enables the STAR-NX client	STAR-CCM+ or STAR-CCM+ Lite and NX license	star-nx
STAR-Inventor	Enables the STAR-Inventor client	STAR-CCM+ or STAR-CCM+ Lite and Inventor license	star-inventor
CAD Exchange	Adds native CAD reading capability to STAR-CCM+	STAR-CCM+ or STAR-CCM+ Lite license	cadexchange
JTOpen Reader	Adds a reader for JT Open files	STAR-CCM+ or STAR-CCM+ Lite license	jtopen
MEDINA Reader	Adds a reader for MEDINA files	STAR-CCM+ or STAR-CCM+ Lite license	medina-import
IC Engine Module	Enables internal combustion engine modeling in STAR-CCM+	STAR-CCM+ or STAR-CCM+ Lite license	StarIce
Battery Simulation Module	Enables Li-Ion Battery modeling in STAR-CCM+	STAR-CCM+ license	batterysim
Optimate	Enables Optimate wizard and HEEDS post	STAR-CCM+, STAR-CCM+ Lite, or STAR-CCM+ Power license	doesuite
Optimate+	Enables Optimate wizard, HEEDS post, and SHERPA	STAR-CCM+, STAR-CCM+ Lite, or STAR-CCM+ Power license	doesuiteplus
Power Tokens	Run time tokens for Optimate and Optimate+	Optimate or Optimate+ license	doetokens
STAR-Cast	Enables casting modeling	STAR-CCM+ or STAR-CCM+ Power license	star-cast

Note: You can configure STAR-CCM+ to release a license automatically when it detects no activity for a set amount of time.

Capabilities of STAR-CCM+ Lite

STAR-CCM+ Lite is a version of STAR-CCM+ that provides a limited set of features. In STAR-CCM+ Lite, when you view models for a physics continuum in the selection dialog, models that are not available with the STAR-CCM+ Lite license are grayed out.

STAR-CCM+ Lite gives you access to all the pre-processing features of STAR-CCM+ such as 3D-CAD, meshing, and surface repair. STAR-CCM+ Lite also give you access to the post-processing features including scenes, streamlines, rendering features, and animation.

For physics, STAR-CCM+ Lite is focused on providing you with enough capabilities to solve basic fluid flow and heat transfer problems using the segregated flow solver. Specifically, STAR-CCM+ Lite provides access to:

Flow characteristics

- Inviscid, laminar, or turbulent
- Newtonian and Non-Newtonian viscosities
- Porous resistance
- Gravitational acceleration
- Passive scalars

Temporal discretization

- Steady
- Unsteady (implicit and PISO)

Equation of state

- Constant density
- Ideal gas law
- Polynomial density

Heat Transfer

- Conduction
- Convection

Turbulence

K-Epsilon (RANS) models

- Standard
- Realizable
- Two-layer
- Low-Reynolds number

- Elliptic blending
- Lag elliptic blending
- Abe-Kondoh-Nagano
- V2F

Materials

STAR-CCM+ Lite comes with a database of common materials in categories of solid, liquid, and gas.

Finite Volume Solid Stress

Suitable for certain types of analysis in solid mechanics, including some fluid-structure interactions.

You can use STAR-CCM+ Lite to post-process the results of a simulation that was solved using STAR-CCM+. It is also possible for STAR-CCM+ Lite to run the simulation for further iterations providing all models are available in STAR-CCM+ Lite. STAR-CCM+ Lite can run Java macros that were generated by STAR-CCM+, but if the macro activates any models that are not permitted under the STAR-CCM+ Lite license, the simulation fails to run.

Contents:

Power-on-Demand Licensing
STAR-CCM+ Add-on Modules
Power Tokens

Power-on-Demand Licensing

The Power-On-Demand (PoD) session allows you to use a STAR-CCM+ Power Session from a Siemens PLM Software public license server rather than a license server on your own network.

You can purchase Real Time credit to use with the public license server. When you purchase Real Time credit, this credit is added to your account where it is available for you to use at any time. Real Time credit has a 12-month validity period. A unique key is used for authenticating the session before it starts. STAR-CCM+ requires this key when you launch a new session with PoD licensing. You can create multiple keys to manage and track your time credit within your company.

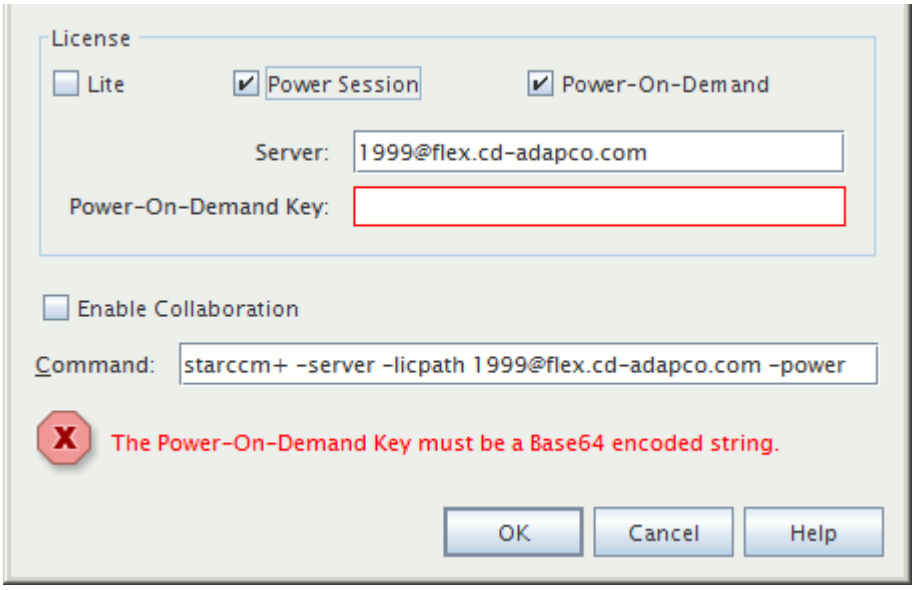
STAR-CCM+ requires some ports to be open to facilitate communication between your machine and the PoD license server. Also define the `CDLMD_LICENSE_FILE` environment variable.

In the *Create a New Simulation* or *Load Simulation* dialog, the correct format for the license path (the **Server** text box) is:

```
1999@flex.cd-adapco.com
```

The Power-On-Demand Key must be a Base64-encoded string.

Verify all your license information before entering it. If the **Power-On-Demand** checkbox is activated and if either of these text boxes is empty or incorrectly formatted, an error message appears. In addition to the error message, the invalid text box is highlighted with a red border.



Additionally, a list of potential error messages with a brief explanation of what they mean is provided in the section on Power-On-Demand server errors.

Contents:

Purchasing Real Time Credit

Viewing Your Power-On-Demand Account Online
Managing License Keys
Which Ports Are Used in a Power-On-Demand Session?
Controlling the Default Behavior of Power-on-Demand Licensing
Launching STAR-CCM+ with the Power-On-Demand Option
Power-On-Demand Deposits
Power-on-Demand Server Errors Reference Page

Purchasing Real Time Credit

Real Time is defined as the difference between the time at which a STAR-CCM+ Power Session ends and the time when it starts. It accounts for the summation of I/O times, CPU times, multiple concurrent sessions, communication channel delays, and all other types of wait.

An example of how Real Time works is outlined below.

- You have 500 hours of time credit in your account. You could run:
  - 1 Power Session job for 500 hours
  - 5 Power Sessions jobs for 100 hours each
  - Or 500 Power Sessions jobs for 1 hour each

Important: when you start a STAR-CCM+ server using PoD licensing, two things happen:

- STAR-CCM+ reserves a number of hours from the available credit. This action is known as a deposit.
- Real Time credit counting starts.

Viewing Your Power-On-Demand Account Online

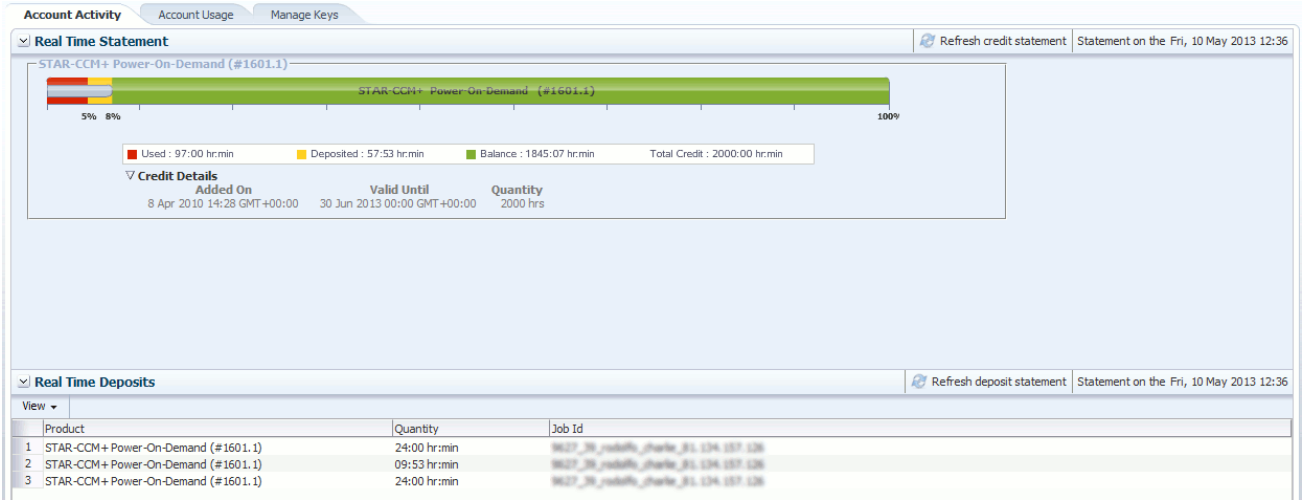
The details of your PoD account can be checked at any time online.

Log in to your customer success portal account (https://thesteveportal.plm.automation.siemens.com).

The remaining amount of PoD credit is displayed on the left of the page.

To get a more detailed view of your PoD account, click **Products** > **Licensing**. If you do not have a customer success portal account, contact your local support office.

The *Account Activity* tab is selected by default. Here, you can check your Real Time statement.



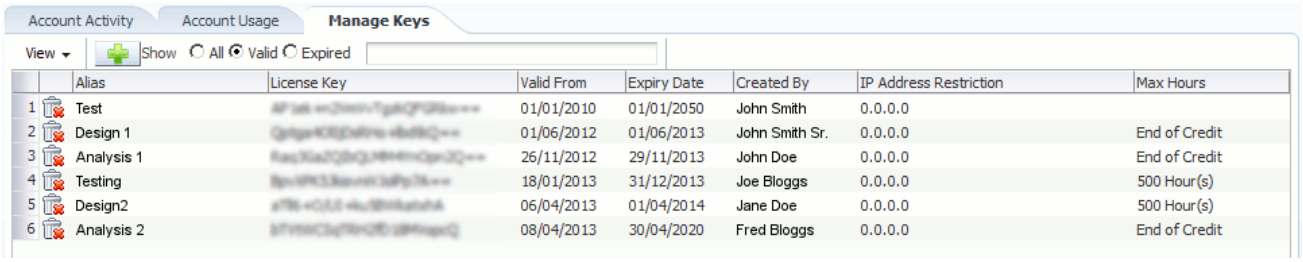
The available balance is shown in green on the graph, the used time is shown in red, while the deposits appear in yellow.

In the *Account Usage* tab, you can create and view reports to monitor the Real Time license usage in your company.

Managing License Keys

You can manage existing keys or generate new ones for each STAR-CCM+ session you wish to run.

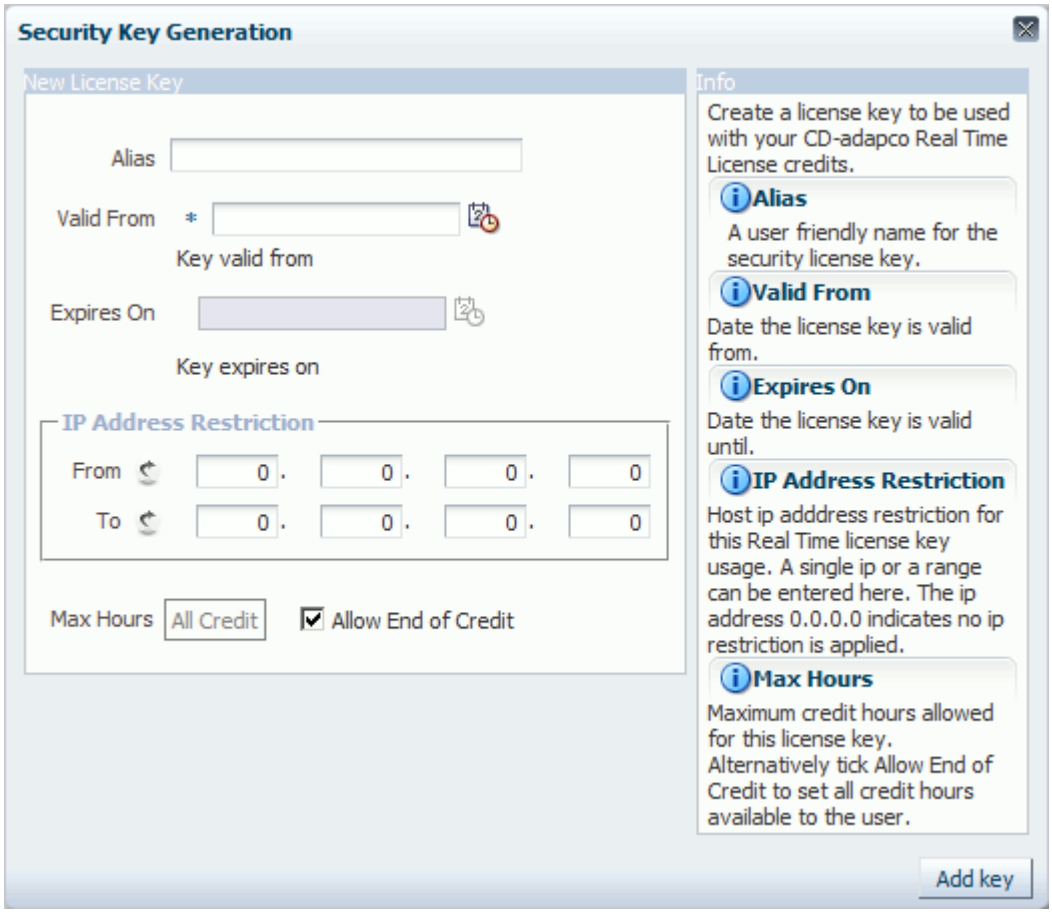
The *Manage Keys* tab provides the relevant action.



To generate more keys:

1. Click (**Create License Key**) at the top of the screen.

   The *Security Key Generation* dialog appears.



Here, you can select an alias for the key and the dates you want it to be valid for. IP address restrictions that you want to place on the key, and the number of hours you want to allocate to it, are also specified in this dialog.

The default starting point for the key is the current date and time. The PoD servers use UTC (Coordinated Universal Time) as a reference time, so the time you set when creating a key must be in UTC to match the server. If you create a key in Japan at 21:00 local time (UTC+9), the key is set to start at 21:00 UTC, not 21:00 in Japan local time. Hence, you are not able to use the key for another 9 hours. To have the key start immediately, set the start time to `00:01`.

2. Click **Add key** to create the key and add it the list.

3. To delete a key, click (**Delete the key on this row**) next to the key you want to delete.

Which Ports Are Used in a Power-On-Demand Session?

Ports 1999 and 2099 must be open to allow communication traffic between your machine and the PoD license server.

On Linux, you can use the Nmap utility to verify that these ports are open. On Windows, you can use the Port Query tool that is provided in the FLEXlm directory of your STAR-CCM+ installation directory.

Verifying Ports on Linux

On Linux, you can use the Nmap utility to check that ports `1999` and `2099` are open.

To check your ports:

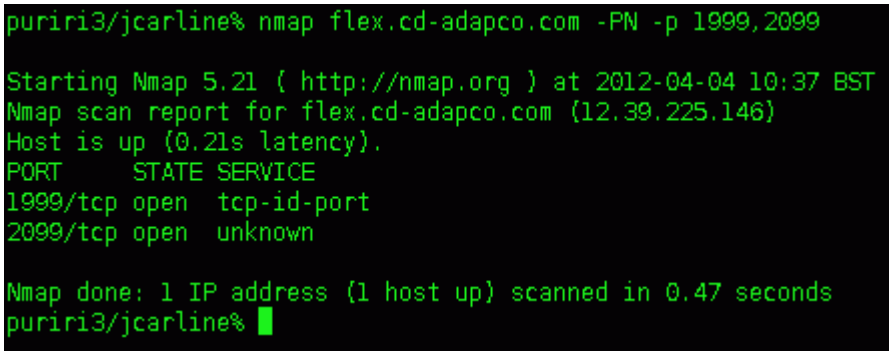
1. Enter the following in the command prompt:

```
% nmap flex.cd-adapco.com -PN -p 1999,2099
```

   where:

   - `[PATH]` is the full path to the nmap.exe executable file.
   - `flex.cd-adapco` is the host that you want to check.
   - `-PN` is the command for Nmap to ping the host.
   - `-p 1999,2099` are the ports that you wish to check.

   A report showing the results is displayed.



   In this example, both ports are `open`. If your report shows that the ports are anything but `open`, contact your system administrator and request that they open these ports for your machine. For more information on what the various *STATE* results mean, visit the Nmap website.

Verifying Ports on Windows

On Windows, you can use Port Query or Nmap to check that ports `1999` and `2099` are open.

Using Port Query

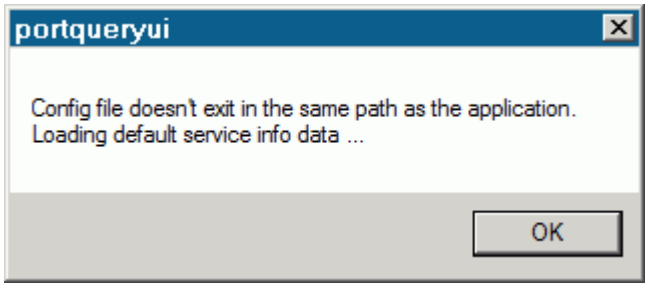
Port Query is included with your STAR-CCM+ installation. The default location is:

```
C:\Program Files\CD-adapco\FLEXlm\11_9_1_0\bin\portqueryui.exe
```

To check your ports:

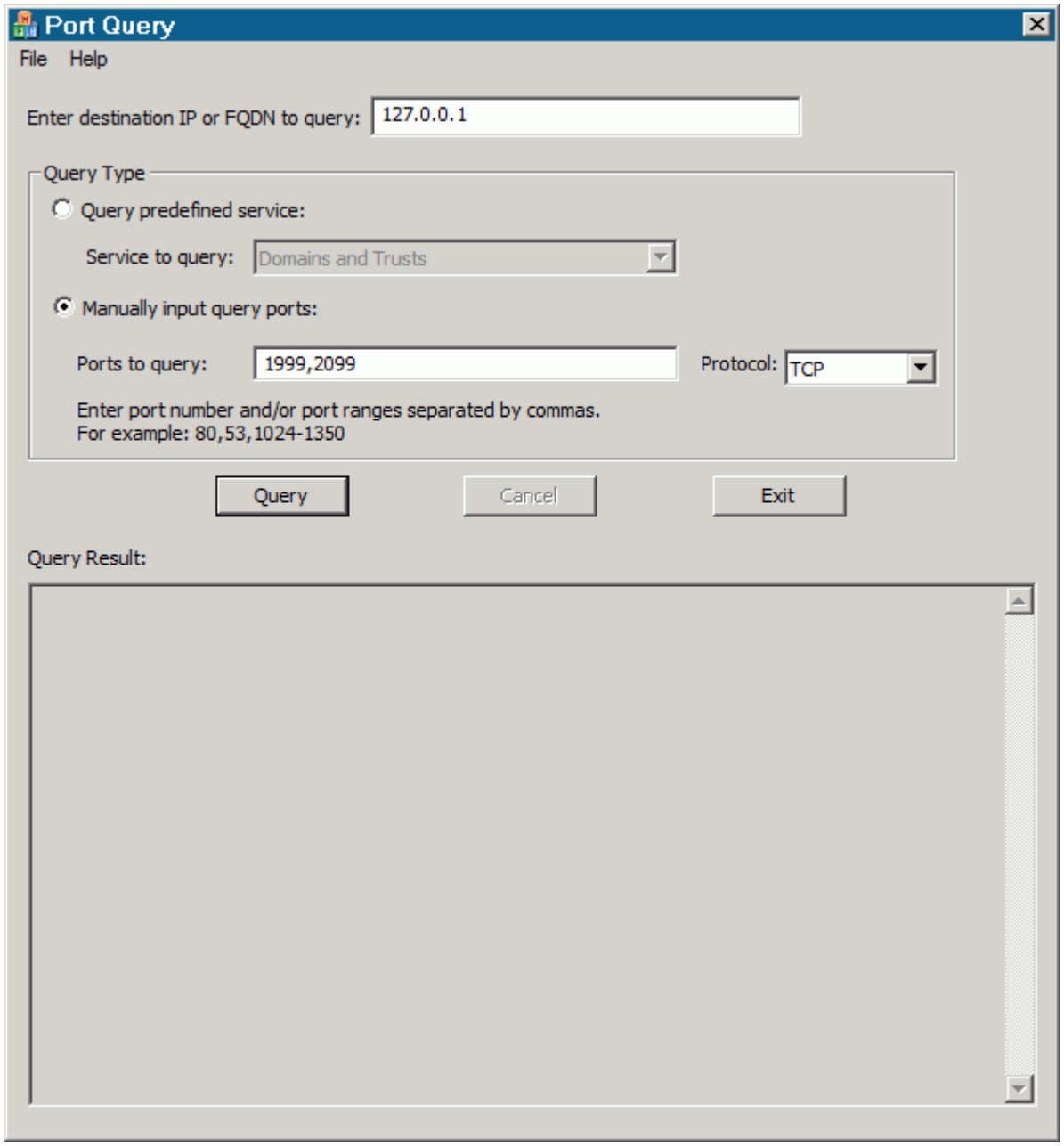
2. Launch `portqueryui.exe`.

   The following message may appear, warning you that no configuration file could be found.



3. Click **OK** to ignore this message.

   The *Port Query* dialog appears.
4. In the *Enter destination IP or FQDN to query:* entry box, enter `flex.cd-adapco.com`
5. Select the **Manually input query ports:** radio button.
6. In the *Ports to query:* entry box, enter `1999,2099`
7. Ensure *Protocol:* is set to **TCP**.



8. Click **Query**.

   The output displays the query results.

```
Query Result:
==============================================

 Starting portqry.exe -n flex.cd-adapco.com -e 1999,2099 -p TCP ...


Querying target system called:

 flex.cd-adapco.com

Attempting to resolve name to IP address...

Name resolved to 12.39.225.146

querying...

TCP port 1999 (unknown service): LISTENING

TCP port 2099 (unknown service): LISTENING
portqry.exe -n flex.cd-adapco.com -e 1999,2099 -p TCP exits with return code 0x00000000.
```

In this example, port 1999 and 2099 are open. If your ports are reported as anything but `LISTENING`, contact your system administrator and request that they open these ports for your machine.

Using Nmap on Windows

Nmap is a free utility that you can download from the Nmap download page. To check that the ports are open:

9. Download the stable command-line zip file.
10. Extract the contents of the file to a location of your choice.
11. Open a command prompt window.
12. Enter the following command:

```
[PATH]\nmap.exe flex.cd-adapco.com -PN -p 1999,2099
```

where:

- `[PATH]` is the full path to the `nmap.exe` executable file.
- `flex.cd-adapco` is the host that you want to check.
- `-PN` is the command for Nmap to ping the host.
- `-p 1999,2099` are the ports that you wish to check.

A report showing the results is displayed.

```
C:\Users\jcarline>C:\nmap-5.51\nmap.exe flex.cd-adapco.com -PN -p 1999,2099

Starting Nmap 5.51 ( http://nmap.org ) at 2012-04-04 13:29 GMT Daylight Time
Nmap scan report for flex.cd-adapco.com (12.39.225.146)
Host is up (0.085s latency).
PORT     STATE SERVICE
1999/tcp open  tcp-id-port
2099/tcp open  h2250-annex-g

Nmap done: 1 IP address (1 host up) scanned in 0.32 seconds

C:\Users\jcarline>_
```

In this example, both ports are `open`. If your report shows that the ports are anything but `open`, contact your system administrator and request that they open these ports for your machine. For more information on what the various *STATE* results mean, visit the Nmap website.

Controlling the Default Behavior of Power-on-Demand Licensing

You can set up and control the default behavior of PoD licensing using environment variables.

The `CDLMD_LICENSE_FILE` environment variable specifies the PoD license server. This environment variable must be present on your machine before you can use PoD licensing.

The `LM_PROJECT` environment variable specifies a specific PoD key to use when using the PoD license server. This optional environment variable is useful if you want to always use a specific key when working with STAR-CCM+.

Setting Variables on Linux

1. Modify the script that sets your Linux environment:

Shell	Variable	Example
Bourne Shell	`CDLMD_LICENSE_FILE`	`export CDLMD_LICENSE_FILE=1999@flex.cd-adapco.com`
	`LM_PROJECT`	`export LM_PROJECT=<PoD_key>` Where `<PoD_key>` is one of the Power-on-Demand keys that are defined in the Power-on-Demand database. See Managing License Keys.
C Shell	`CDLMD_LICENSE_FILE`	`setenv CDLMD_LICENSE_FILE 1999@flex.cd-adapco.com`
	`LM_PROJECT`	`setenv LM_PROJECT <PoD_key>`

Shell	Variable	Example
		Where `<PoD_key>` is one of the Power-on-Demand keys that are defined in the Power-on-Demand database. See Managing License Keys.

Setting Variables on Windows

Use the *Environment Variables* dialog to create the PoD variables:

2. Click **Start** and type `env` into the search field near the bottom of the **Start** menu.
3. Select **Edit environment variables for your account** from the list of results.

   The *Environment Variables* dialog appears.
4. In the *System variables* group box, click **New...**.

   The *New System Variable* dialog appears.
5. Enter the details for the variables:
   - For the `CDLMD_LICENSE_FILE` variable, set the following properties:

Property	Setting
Variable name	`CDLMD_LICENSE_FILE`
Variable value	`1999@flex.cd-adapco.com`
   - For the `LM_PROJECT` variable, set the following properties:

Property	Setting
Variable name	`LM_PROJECT`
Variable value	`<PoD_key>`

     Where `<PoD_key>` is one of the Power-on-Demand keys that are defined in the Power-on-Demand database. See Managing License Keys.
6. Click **OK** to close the *New System Variable* dialog.
7. Click **OK** to close the *Environment Variables* dialog.

Launching STAR-CCM+ with the Power-On-Demand Option

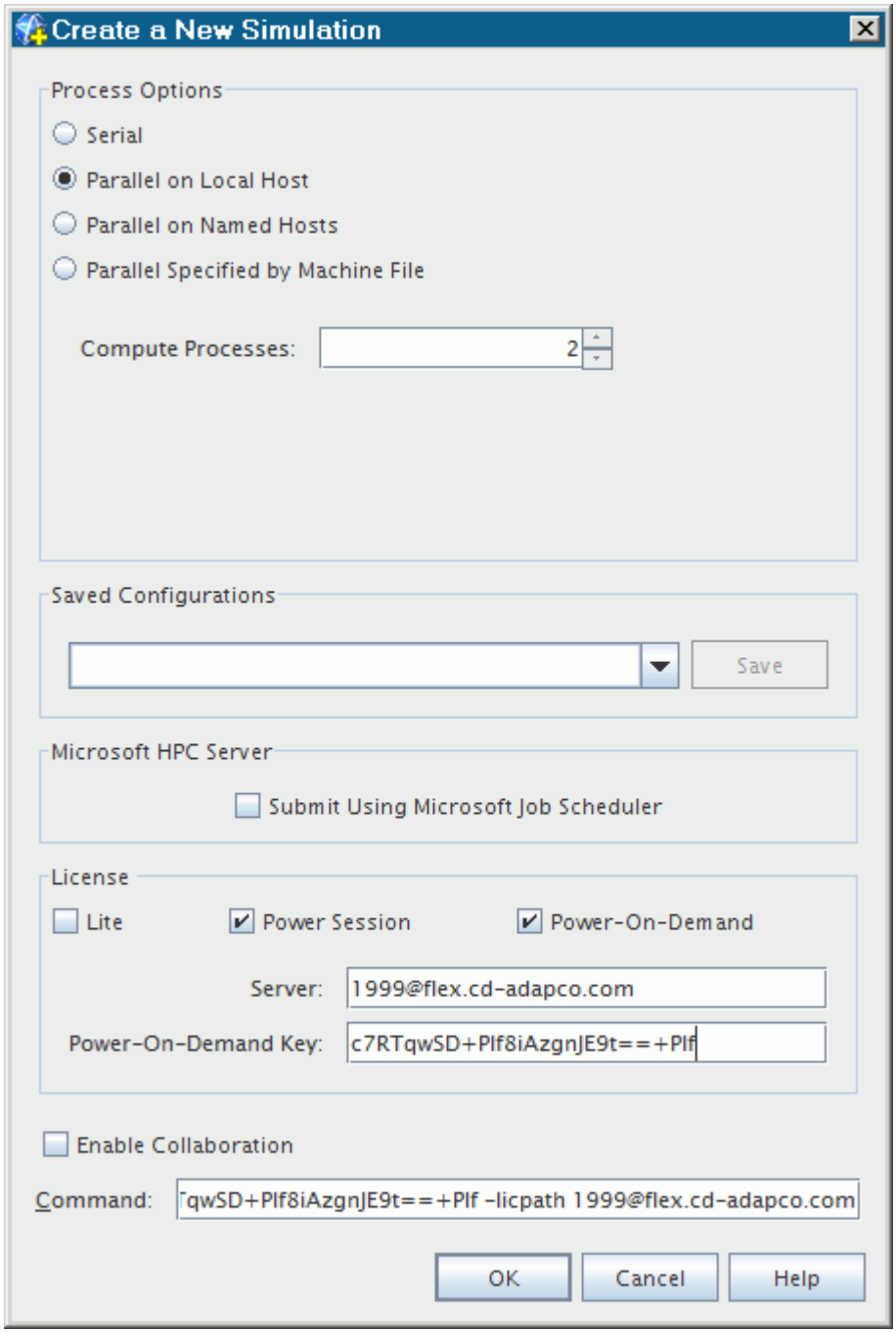
Launch STAR-CCM+ with the Power-On-Demand option.

To enable PoD licensing:

1. Tick the PoD checkbox in the *Create a New Simulation* dialog.

   STAR-CCM+ automatically enables the **Power Session** option, and presents entry boxes for the Server and PoD Key. By default, the STAR-CCM+ public license server is entered in the **Server** entry box: `1999@flex.cd-adapco.com`.
2. Enter your PoD key in the **PoD Key** entry box.

   You can copy and paste this line from the *License Key* column of the PoD license management webpage.



Power-On-Demand Deposits

When you start a PoD session, a deposit of 24 hours is automatically taken out of your account.

If your job runs beyond this first 24 hours, a further deposit of 24 hours is taken. This procedure is repeated until the job completes. When you end the simulation, any hours unused in the final deposit are returned to your overall credit account.

Note: If you run out of credits while your job is running, this job continues to run until completion. Purchase more credit to cover the hours used.

Power-on-Demand Server Errors Reference Page

A list of PoD server error messages—with an explanation of what they mean—is provided below. These error messages are displayed in the *Output* window of STAR-CCM+ when the error occurs.

Error Message	Description
`FAILED_TOKEN_NOT_RECOGNIZED -20001`	This error shows that the key that is passed via `LM_PROJECT` or `-podkey` does not match any keys that are stored in the PoD database. Make sure that the key you entered is valid. A key is a binary string and is converted to text using base64 encoding, so it is typically 24 or 28 characters long. Valid characters are: any of the 52 upper and lower case letters, the numbers 0–9, and the + and / characters.
`FAILED_TOKEN_EXPIRED -20002`	This error means that the keys that you created in your account on the customer success portal at https://thesteveportal.plm.automation.siemens.com have expired. To resolve this issue, create new keys.
`FAILED_TOKEN_NO_CREDIT -20003`	In your customer success portal account, you can define a cap when creating PoD keys. This error means that the capped usage of the key has been exceeded and the key is no longer valid. To resolve this issue, create a new key.
`FAILED_TOKEN_IP_ADDRESS_NOT_IN_RANGE -20004`	This error appears when your IP address does not match the allowed range that you set up in the keys that are stored in your customer success portal account. Your organization probably uses NAT (network address translation), so a machine can have one IP address on internal networks but another one when it accesses external networks. As the PoD server is external, make sure that you use the right IP range if you choose this option when creating a key. Alternatively, create a new key and do not enable the IP range restriction option.
`FAILED_NO_CREDIT_AT_ALL -20005`	This error means that there is no credit left in your PoD account. Credit is valid for a 12 month period, so this error can mean that your unused credit has expired. To resolve this issue, purchase more credit.
`FAILED_NO_PROJECT -30001`	This error means that you have not set `LM_PROJECT` either via the environment variable or via the `-podkey` command. To resolve this issue, create an environment variable named `LM_PROJECT` and enter your PoD key as the variable value. Alternatively, use the `-podkey <yourkey>` command when launching STAR-CCM+.
`FAILED_NO_CONNECTION -30002`	This error means that the PoD server cannot connect to the license database. This error indicates a problem with the PoD server.

Error Message	Description
`FAILED_REFUSED_CONNECTION -30003`	This error means that the connection from the PoD server to the license database has been refused. This error indicates a problem with the PoD server.
`FAILED_EXPECTED_FAILURE -40001`	This error only appears when the PoD license server undergoes routine maintenance.
`FAILED_UNCHARGED_CHECKOUT -50001`	This error appears for STAR-CD PoD and occurs when the `hpcdomains` license checks out before the `starsuite` license. This error can mean that you are checking out `starsuite` from a local server and the `hpcdomains` from the PoD server. To resolve this issue, look at the license output and check the order of the checkouts; the `starsuite` license must check out before the `hpcdomains` license.

STAR-CCM+ Add-on Modules

STAR-CCM+ automatically checks out any license that it needs for add-on modules.

A list of the add-on modules, along with the feature licenses that they use, is provided below:

- Battery Simulation Module (which uses the `batterysim` feature license)
- STAR-Cast (which uses the `star-cast` feature license)
- IC Engine Module (which uses the `proice` feature license)
- JTOpen Reader (which uses the `JTOpen` feature license)
- CAD Exchange Reader (which uses the `cadexchange` feature license)
- MEDINA Reader (which uses the `medina-import` feature license)

For example if you load a simulation file that contains items from the battery module, STAR-CCM+ checks out an available `batterysim` feature license.

Add-on licenses are checked out when:

- an add-on feature is used in a simulation
- the license is requested from the command line
- it is specified in the Add-on Licensing dialog
- it is reserved in the options dialog

Add-on licenses are checked in when:

- a simulation is closed
- a requiring feature is removed
- it is deactivated from the *Add-on Licensing* dialog

Note: By default, the JTOpen Reader and CAD Exchange Reader licenses are automatically released after importing a CAD geometry into STAR-CCM+. The licensing options for the STAR-CCM+ add-on modules can be manually controlled through the add-on licensing manager.

You can set up your simulation in your native CAD package environment using one of the STAR-CCM+ CAD client add-ons.

Contents:

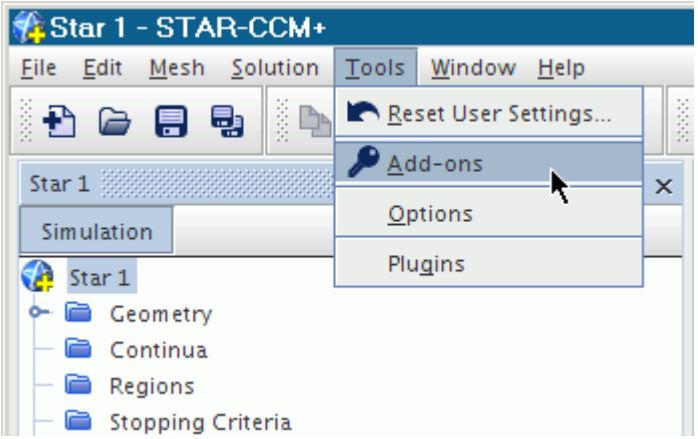
Working with the Add-on Licensing Manager
CAD Client Add-ons

Working with the Add-on Licensing Manager

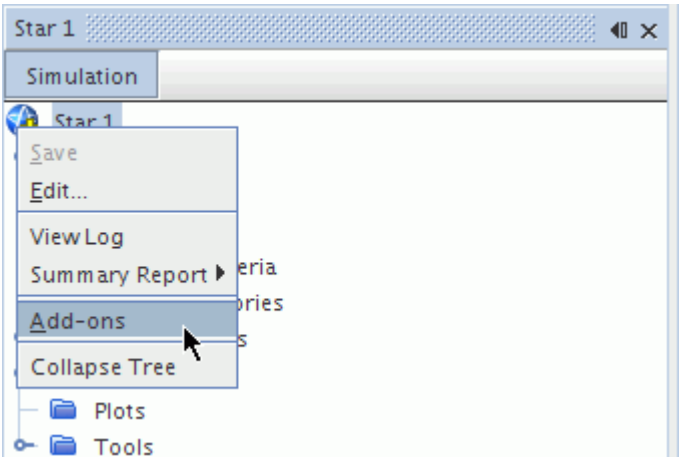
Licenses for the STAR-CCM+ add-on modules can be managed through the *Add-on Licensing* dialog.

To access this dialog:

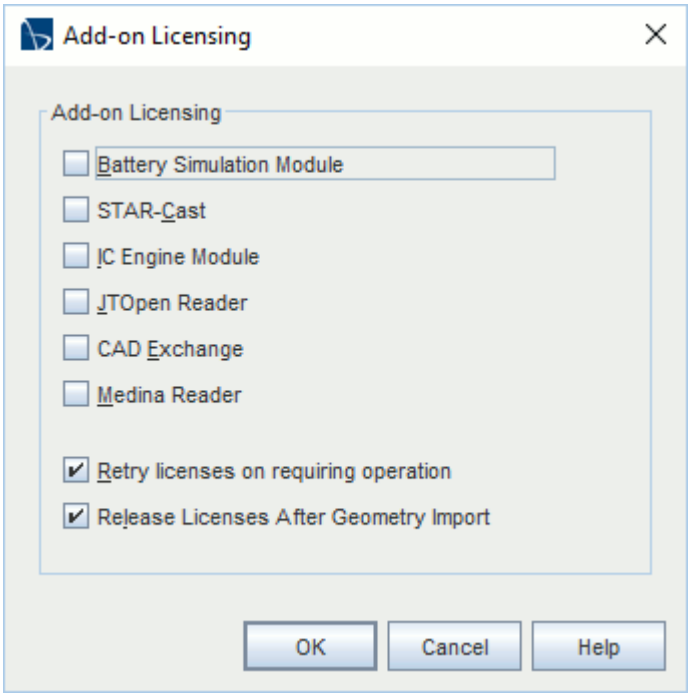
1. Select **Tools** > **Add-ons** from the menu bar.



2. Alternatively, right-click on the root node in the STAR-CCM+ object tree and select **Add-ons**.



The *Add-on Licensing* dialog appears.



3. You can manually check out licenses by activating the checkbox next to the respective add-on, and clicking **OK**.

   A message then appears in the *Output* window informing you that the license was checked out successfully. Checked out licenses can be checked back in by deactivating the checkbox next to the respective add-on, and clicking **OK**. A message appears in the *Output* window informing you of this action.

The *Retry licenses on requiring operation* option controls whether STAR-CCM+ tries to check out the required license (if it is not already checked out). If this option is cleared, STAR-CCM+ does not try and check out any of the add-on licenses.

The *Release licenses after geometry import* option controls whether the JTOpen Reader and CAD Exchange licenses are released after importing a CAD geometry into STAR-CCM+. If this option is cleared, these two licenses will remain checked out until you manually release them or the simulation file is closed.

Contents:

Reserving a License

Reserving a License

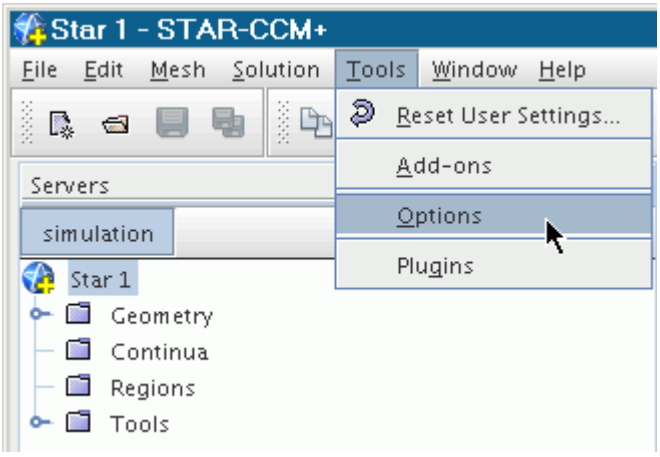
Add-on licenses can be configured so that they are automatically checked out when a STAR-CCM+ session is started.

This option is useful when a certain add-on module is regularly used and can ensure that a license is always available when you need it.

Note: A reserved license is released when the simulation file is closed.

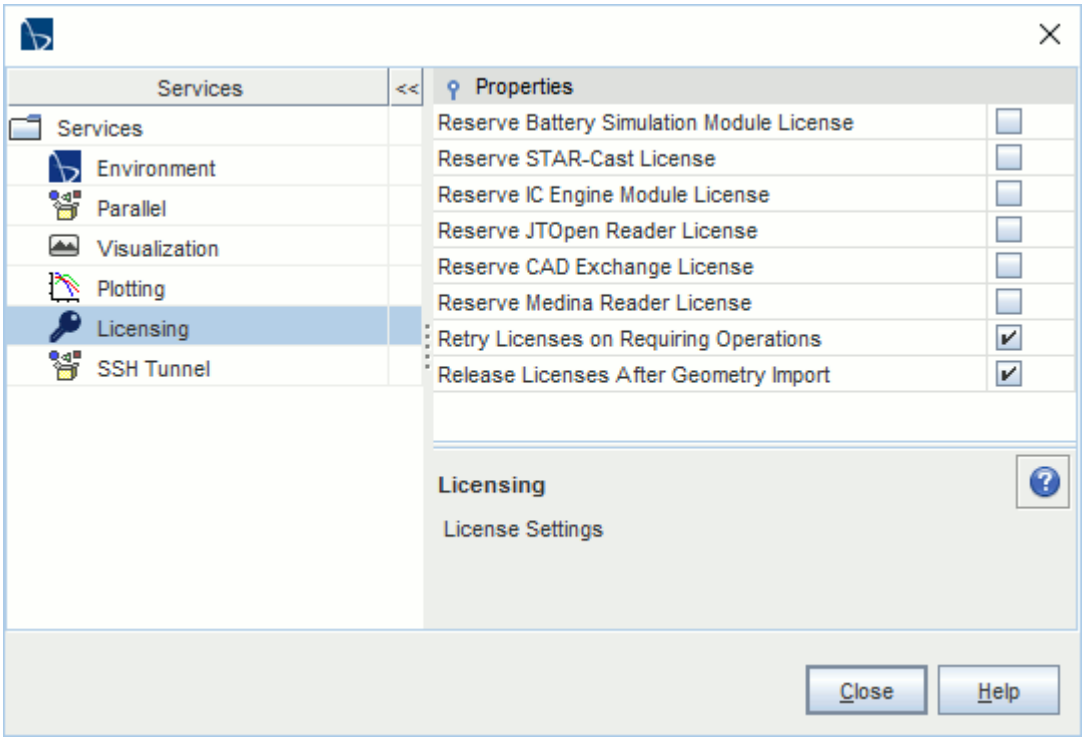
To reserve an add-on license:

1. Select **Tools** > **Options** from the menu bar.



A dialog displaying the various options for STAR-CCM+ appears. This dialog can also be accessed when no STAR-CCM+ session is active.

2. Expand the STAR-CCM+ node and select **Licensing**.



The pane in the right-hand side of the dialog displays the reserve options for each add-on.

3. Ticking these checkboxes and clicking **Close** will place a reserve on the respective add-on modules, so that the corresponding licenses will be automatically checked out next time a STAR-CCM+ session is started.

4. To release a license, manually check it in by clearing the respective checkbox in the Add-on Licensing dialog.

CAD Client Add-ons

The CAD clients plug into your native CAD work package and allow you to set up a simulation within its workspace.

You can also export your CAD geometries from your CAD package directly into STAR-CCM+ by clicking the **Transfer Geometry** button.

The following CAD client add-ons are available.

- STAR-CAT5
- STAR-Inventor
- STAR-Pro/E
- STAR-Creo
- STAR-NX

Power Tokens

If you have an Optimate or Optimate+ license, you can use power tokens (`DOEtoken` feature licenses) with certain features of STAR-CCM+.

Contents:

Using Power Tokens for Pre- and Post-Processing Sessions
Using Power Tokens as a Parallel License Feature for HPC Runs
Controlling the Check Out Order of DOE Parallel Licensing

Using Power Tokens for Pre- and Post-Processing Sessions

If you have an Optimate or Optimate+ license, with power tokens, you can use 10 power tokens to perform pre- and post-processing in STAR-CCM+.

You use power tokens instead of a STAR-CCM+ license. You can run the pre- and post-processing session interactively or in batch mode. During the pre- and post-processing session, the STAR-CCM+ solver is deactivated.

To use power tokens with STAR-CCM+:

1. Launch STAR-CCM+ and load or create a simulation with the `-powerpre` argument.

STAR-CCM+ checks out 10 copies of DOEtoken.

Using Power Tokens as a Parallel License Feature for HPC Runs

If you have an Optimate or Optimate+ license, with power tokens, you can use power tokens as parallel license features.

Prior to STAR-CCM+ v9.02, the license check-out procedure for a HPC run was as follows:

1. Try to satisfy the parallel requirement with `hpcdomains`.
2. If insufficient licenses, try to satisfy the remaining requirement with `ccmplite` (if using STAR-CCM+ lite).
3. If insufficient licenses, and the runtime option `-nosuite` is not specified, try to satisfy the remaining requirement with `ccmpsuite`.

From STAR-CCM+ v9.02 onwards, you can use power tokens to fulfill requirements for a parallel license.

The new procedure is as follows:

4. Try to satisfy the parallel requirement with `hpcdomains`.
5. If insufficient licenses, try to satisfy the remaining requirement with `DOEtoken`.

6. If insufficient licenses, try to satisfy the remaining requirement with `ccmplite` (if using STAR-CCM+ lite).
7. If insufficient licenses, and the runtime option `-nosuite` is not specified, try to satisfy the remaining requirement with `ccmpsuite`.

Controlling the Check Out Order of DOE Parallel Licensing

When running Optimate cases, you can control the check out order of design exploration session licenses.

The standard process for checking out design exploration session parallel licenses is as follows:

1. Try to satisfy the parallel requirement with `DOEtoken` (power tokens).
2. If insufficient licenses, try to satisfy the remaining requirement with `hpcdomains`.
3. If insufficient licenses, and the runtime option `-nosuite` is not specified, try to satisfy the remaining requirement with `ccmpsuite`.

You can switch the order that the `DOEtoken` (power tokens) and `hpcdomains` license features are checked out using the `-doe-prefer-hpcdomains` runtime option. The license features are then checked out in the following order:

4. Try to satisfy the parallel requirement with `hpcdomains`.
5. If insufficient licenses, try to satisfy the remaining requirement with `DOEtoken` (power tokens).
6. If insufficient licenses, and the runtime option `-nosuite` is not specified, try to satisfy the remaining requirement with `ccmpsuite`.

Example Scenario

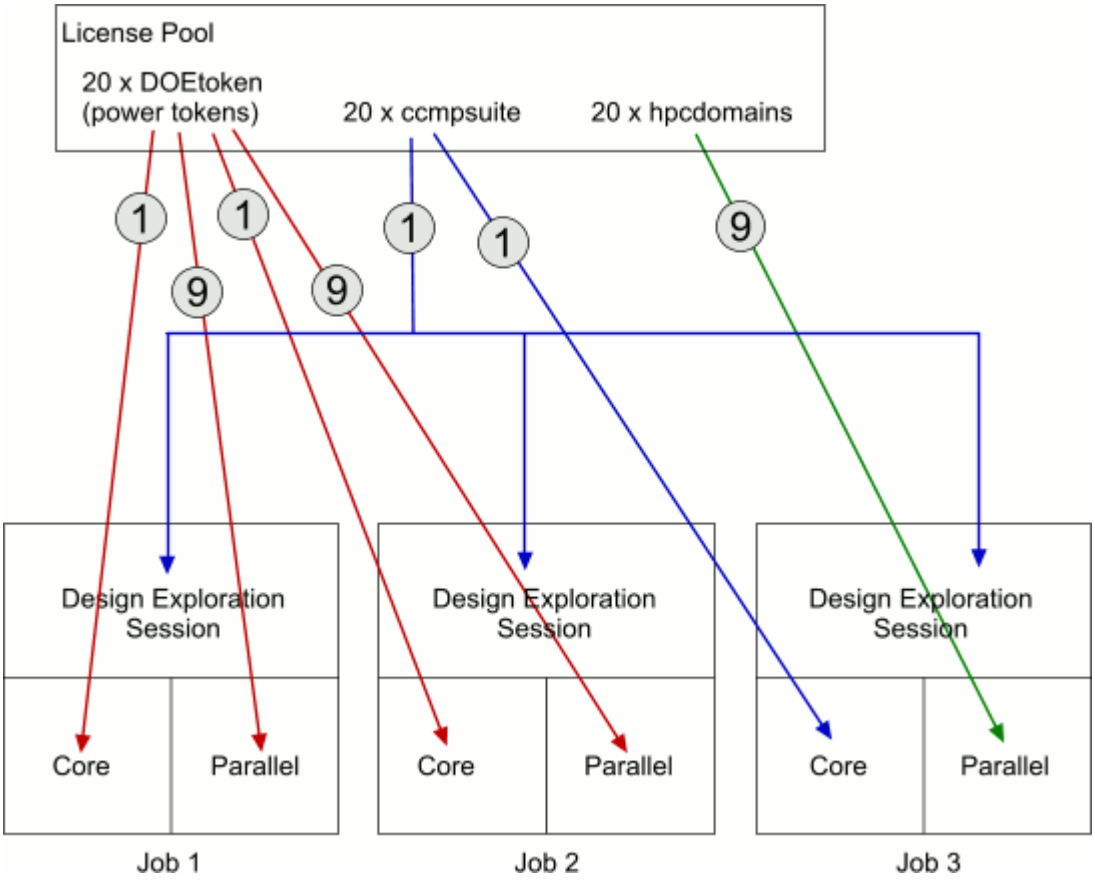
You have the following feature licenses available:

- 20 `ccmpsuite`
- 20 `DOEtoken`
- 20 `hpcdomains`

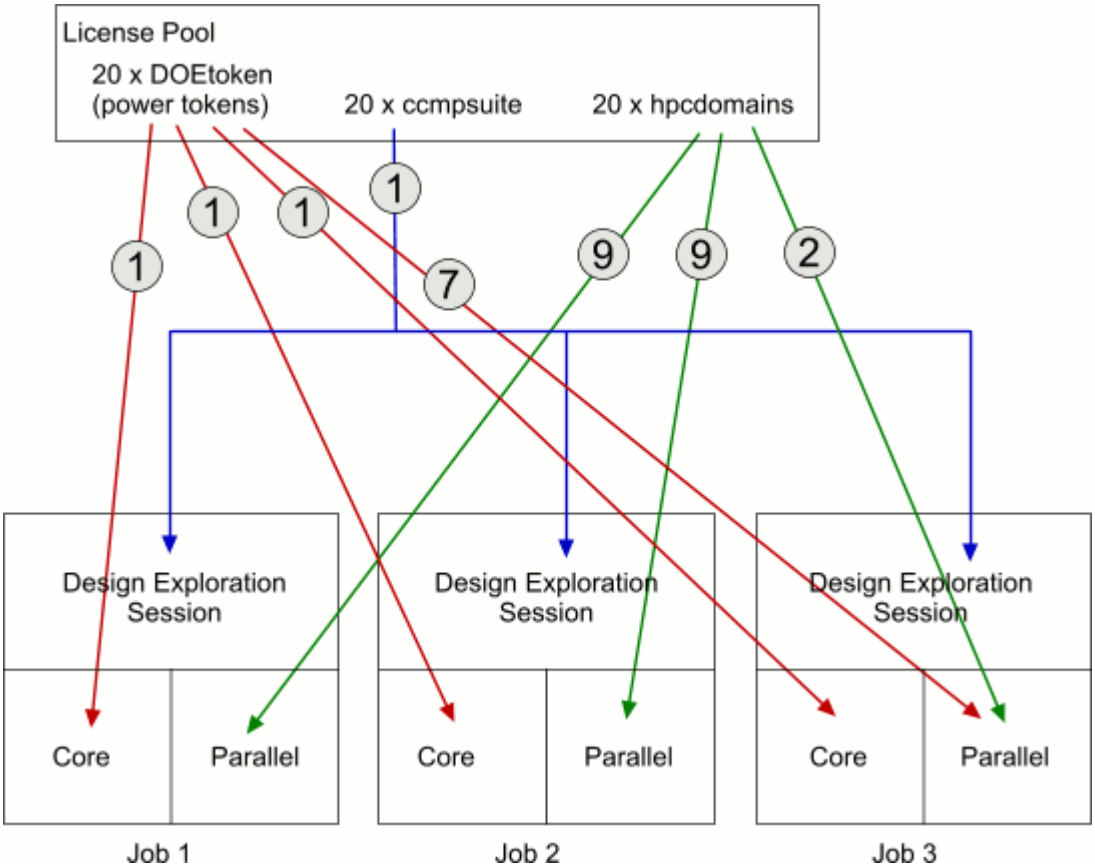
You want to run 3 jobs simultaneously, with 10 processors for each job. Job 1 is run first, then Job 2, followed by Job 3. Each job requires the following licenses:

- 1 design exploration session license
- 1 core license
- 9 parallel licenses

For each design exploration study, a single `ccmpsuite` feature license is checked out for the design exploration session feature. All STAR-CCM+ jobs that are spawned by a design exploration sessions study will share a single `ccmpsuite` license. Each job also requires a license for the core process. The default behavior is to use `DOEtoken` licenses (power tokens) to fulfill this requirement. If there are insufficient `DOEtoken` licenses for the core process, a `ccmpsuite` license is checked out instead. Similarly, `DOEtoken` licenses are used for each parallel process. If there are no `DOEtoken` licenses available for the parallel processes, `hpcdomains` are used instead.



Using the `-doe-prefer-hpcdomains` runtime option makes `hpcdomains` licenses the preferred license for fulfilling the parallel license requirement. If there are insufficient `hpcdomains` licenses available for the parallel processes, `DOEtoken` licenses are used to fulfill the remaining requirement. `DOEtoken` is still used to satisfy the license requirements for the core process.


License Pool
20 x DOEtoken
(power tokens)
20 x ccmpsuite
20 x hpcdomains
1
1
1
1
7
9
9
2
Design Exploration Session
Design Exploration Session
Design Exploration Session
Core
Parallel
Core
Parallel
Core
Parallel
Job 1
Job 2
Job 3

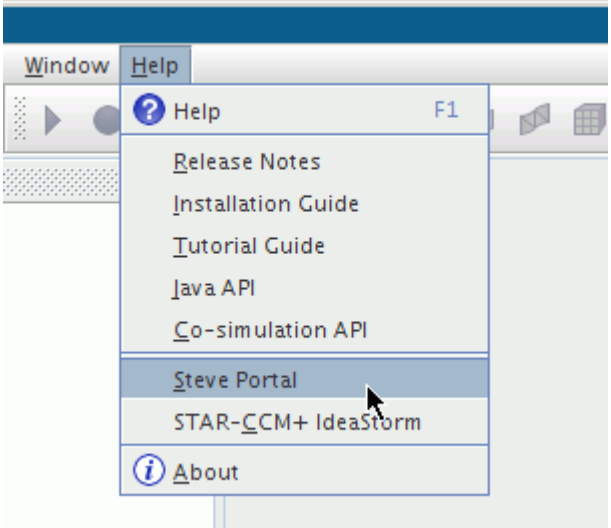
Getting Support

When you need assistance with using STAR-CCM+, you can first check the dedicated support site, the customer success portal.

The customer success portal lets you search information and open a support case. Access the customer success portal at:

https://thesteveportal.plm.automation.siemens.com

Alternatively, select **Help** > **Steve Portal**.



Choose from among the following features on this site:

- A Knowledge Base of Frequently Asked Questions, Videos, Best Practices, and other types of articles that cover a wide range of topics. You can search thoroughly for materials that relate to your problem. Perhaps your question has already been answered.
- A searchable library of documentation and training materials.
- The Macro Hut forum for sharing macros with the community.
- IdeaStorm to suggest new features, and vote on others that are submitted by the community, for STAR-CCM+.
- Access the status of your Power-on-Demand license and the next training sessions, as well as other technical resources.
- Tools for opening and tracking support cases.

The customer success portal can guide you through the process of opening a support case. In addition, it can suggest possible solutions that are based on the details of your case, and help you to track the progress of the case through to resolution.

The screenshot below shows the Home page of the customer success portal. To switch between different areas of the site, use the tabs on the top-level navigation menu.

If you cannot find the answer to your question by searching the customer success portal resources, you can open a support case for your support representative by clicking **OPEN SUPPORT CASE**. Alternatively, if Chat is available, select **Chat with Support**.

To help your Siemens PLM Software support representative, please prepare appropriately for reporting the issue. Further instructions on this subject are:

Contents:

Gathering Information on the Issue

Preparing a Case for Support

Gathering Information on the Issue

Accurate information is vital for dealing with your issue effectively. Your affected simulation file, if you can provide it, would be a key source of this information. See Preparing a Case for Support. Discuss the details of this file transfer with your Siemens PLM Software support representative.

In addition, there are three ways to obtain information, depending on the issue:

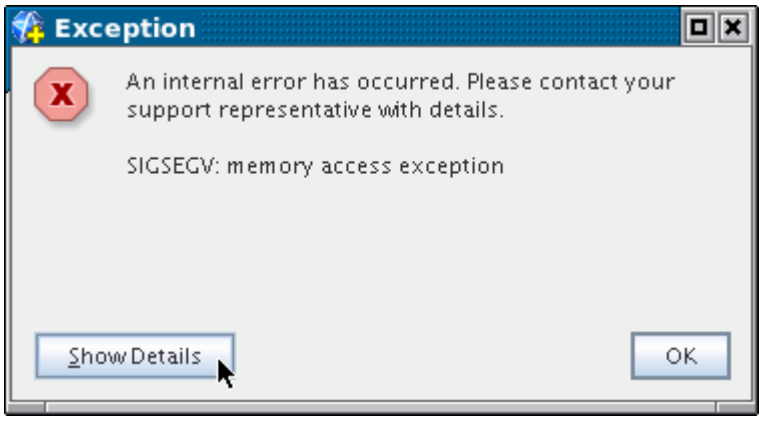
- Copying data from the workspace
- Getting an external data file
- Retrieving graphics driver information

Copying Data from the Workspace

Error messages in STAR-CCM+ commonly appear in an exception dialog, which contains extensive information.

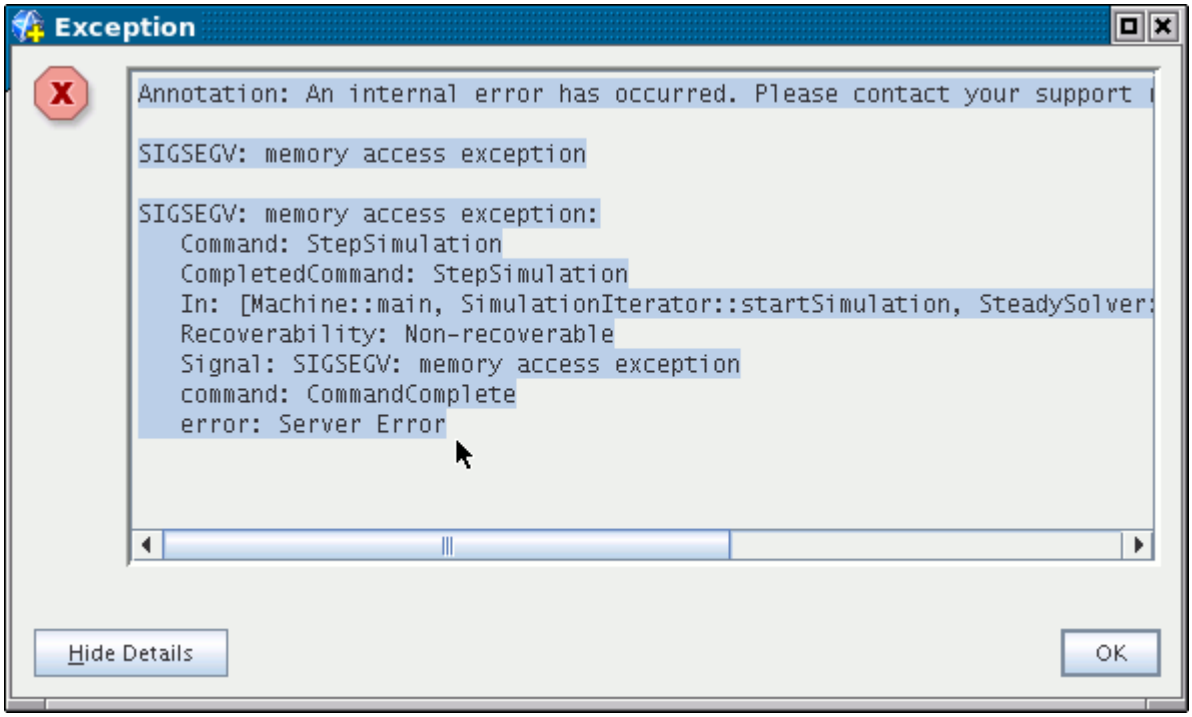
To access this information:

1. Click the **Show Details** button.



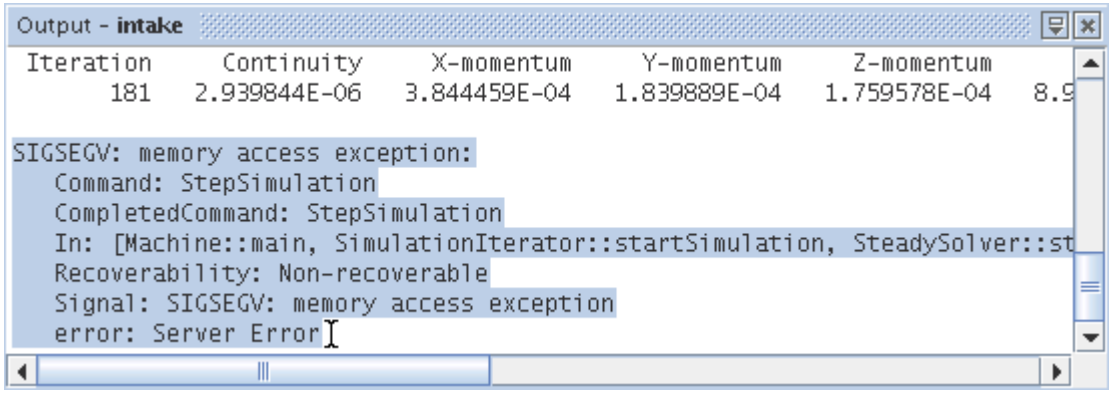
   The details appear in the dialog.

2. Simply highlight the information with your cursor and, when you have reached the end of the text, copy it to the clipboard and paste it into an e-mail message.

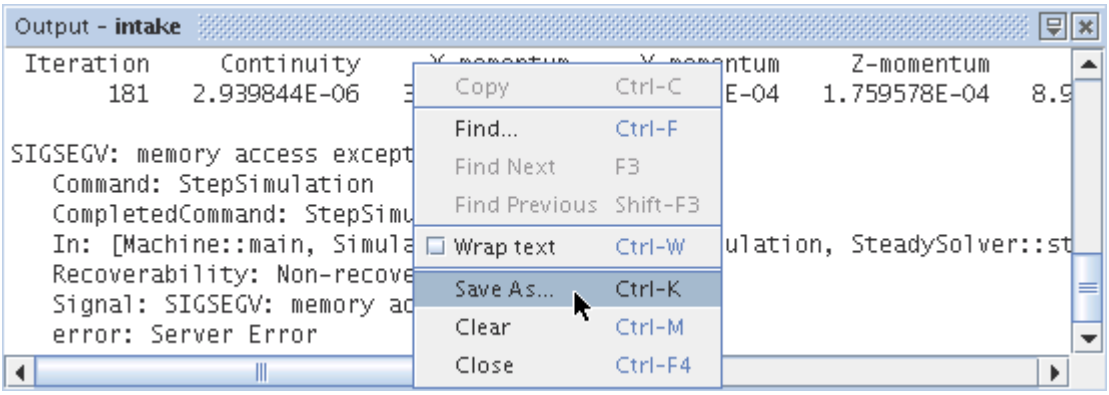


Click the **Hide Details** button to restore the brief dialog if you prefer.

3. In some cases, the error message also appears in the *Output* window, where you can also highlight and copy information.



4. Alternatively, you can use the pop-up menu of the *Output* window to save the full content as a text file.

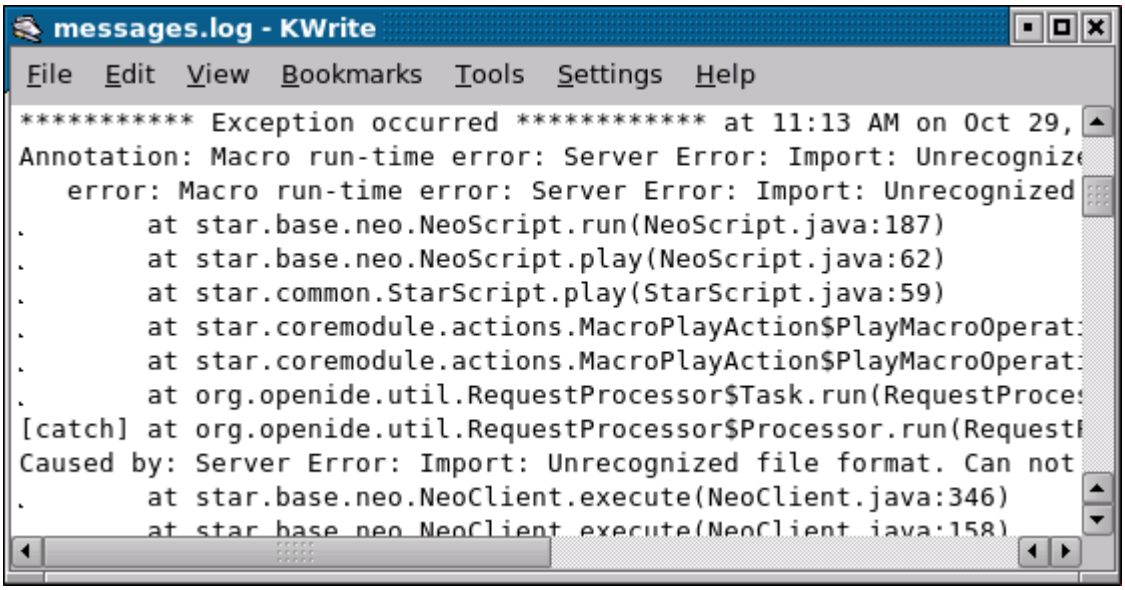


If the resulting text file contains substantial information that does not pertain to the error message, you may want to edit it.

Getting an External Data File

Various errors cause STAR-CCM+ to generate separate reports. You should send these reports to Siemens PLM Software support along with your query. The details they contain can help resolve your issue much sooner.

The most common external source of information on STAR-CCM+ errors is the `messages.log` file.

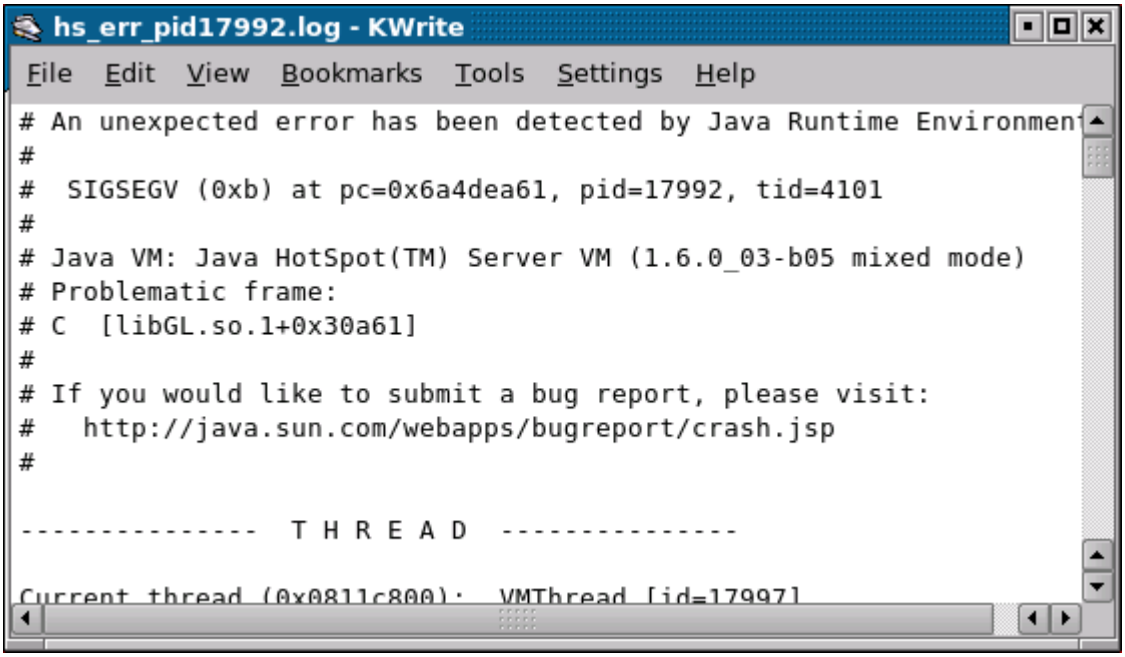


This file can be found in the following locations:

- On Linux: `~/.star-X.XX.XXX/var/log/messages.log`, where ~ is your `home` directory and `X.XX.XXX` is the version of STAR-CCM+, for example 9.04.006
- On Windows: `C:\Users\username\AppData\Local\CD-adapco\STAR-CCM+ X.XX.XXX\var\log\`

The file also contains prior error messages, so you may want to delete the earlier material for brevity.

Some errors are known as Java HotSpot errors because they are related to the Java functionality. As a result, a special log file may be generated called `hs_err_pidXXXX.log`, where `XXXX` is the process ID.



5. Look for the report in your main directory:
   - Your `home` directory on Linux
   - `C:\Users\user\Documents\` on Windows

Unlike `messages.log`, this type of report is generated one time due to a single error. Therefore all of its information pertains to that event.

Retrieving Graphics Driver Information

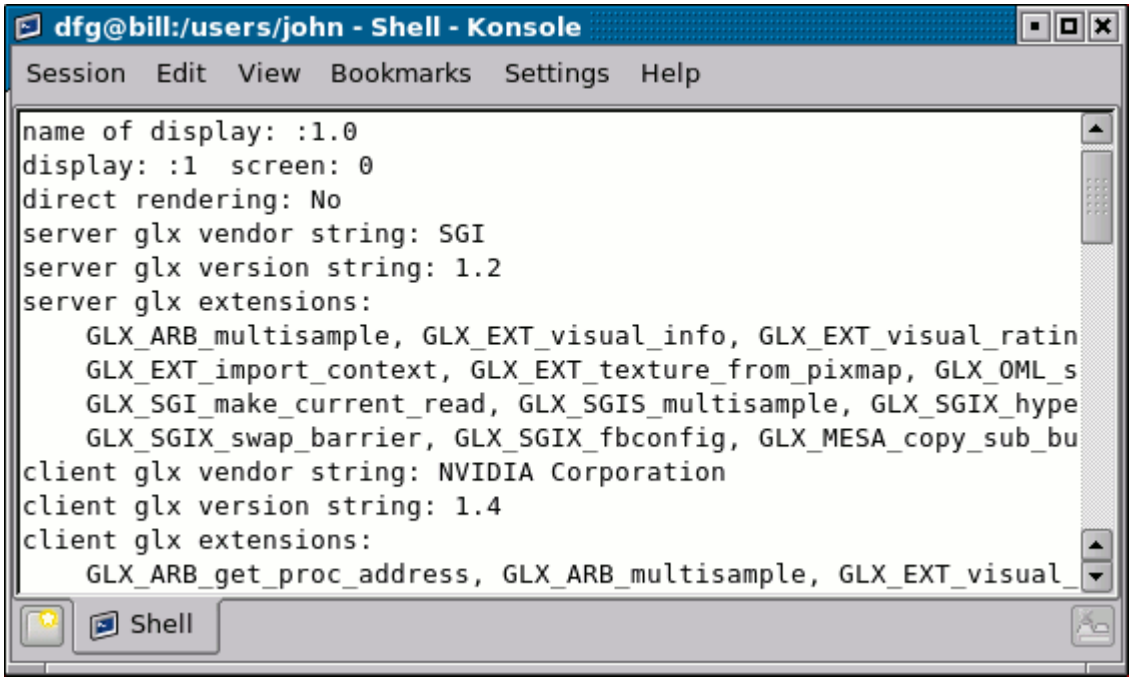
Some issues may originate with your graphics driver. You may want to get detailed information about that driver and send it to Siemens PLM Software support. At a minimum you should provide the driver version, card model and operating system.

The technique for obtaining this information depends on your operating system:

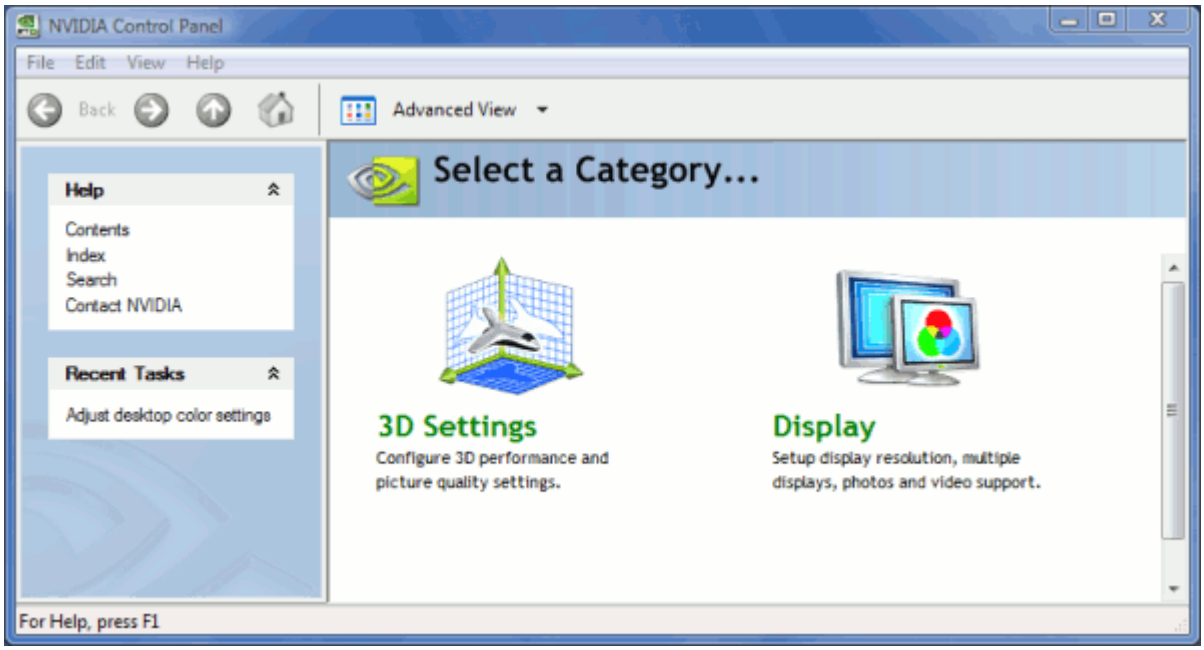
6. On Linux, use the line command:

   ```
   % glxinfo
   ```

   This results in detailed output, which you can either paste in the body of your e-mail message or save as a separate text file.

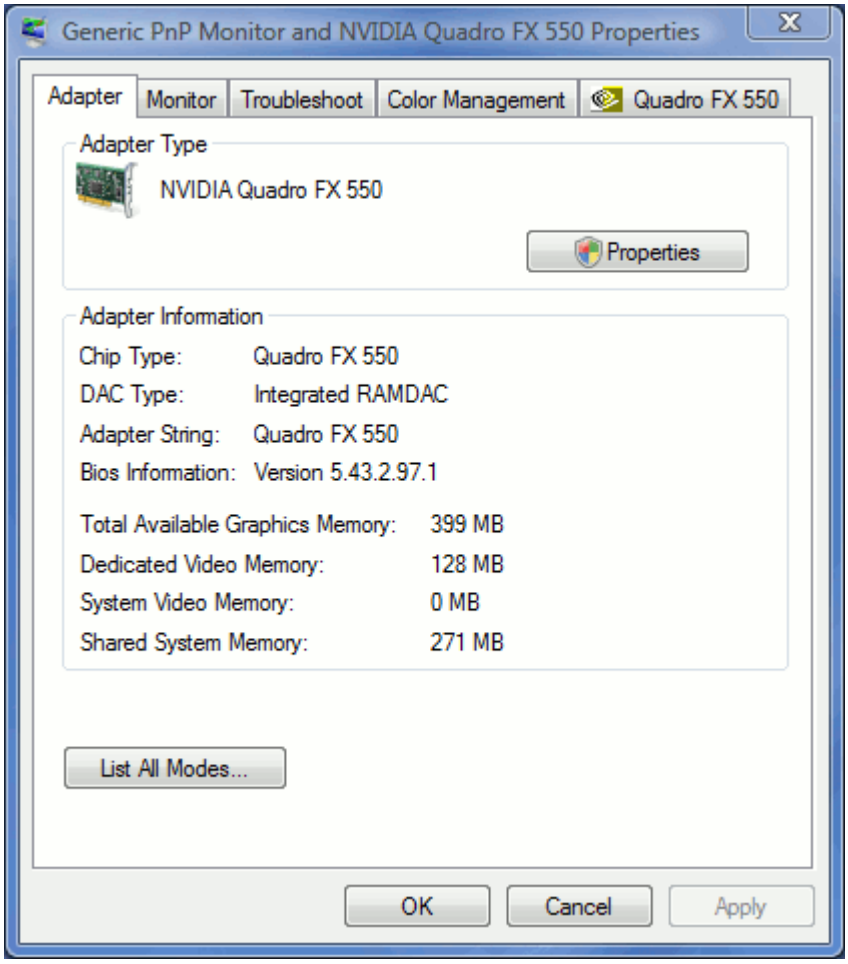


7. On Windows, some drivers might have a special all-in-one interface. The example shown in the following screenshot is accessible on Windows by right-clicking the desktop and selecting **NVIDIA Control Panel**.



8. Alternatively on Windows, search for `display settings` in the Windows search tool and select **Display Settings** from the list of results.
9. In the *Display Settings* dialog, click the **Advanced display settings** link at the bottom of the dialog.
10. In the *Advanced display settings* dialog, click the **Display adapter properties** link at the bottom of the dialog.

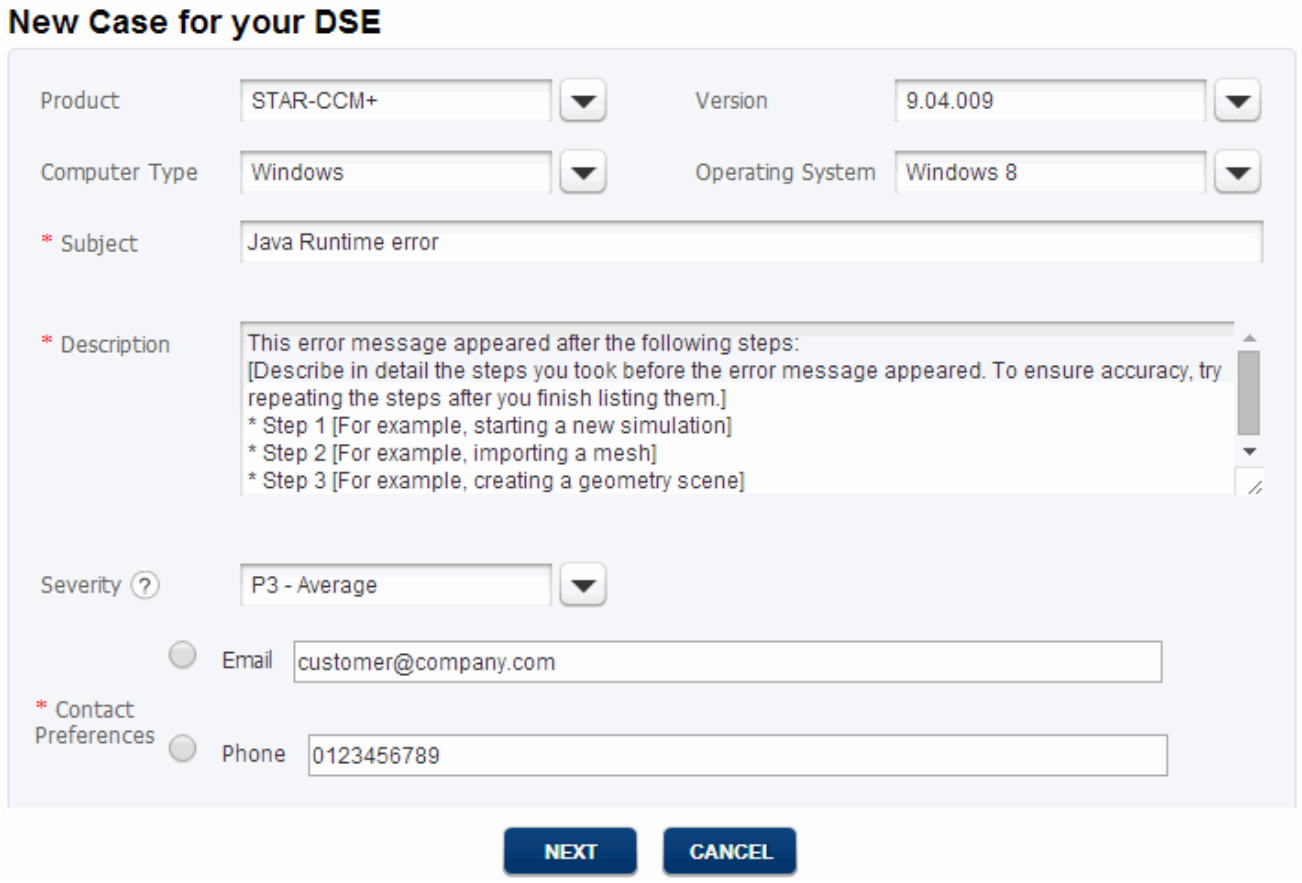
The following dialog appears with tabs that provide detailed information on the graphics hardware and drivers.



Preparing a Case for Support

This section explains how to get your case information to Siemens PLM Software support.

To create a case that is routed directly to your Dedicated Support Engineer (DSE), when in the customer success portal, click **OPEN SUPPORT CASE**, and fill the details as shown in the following image.



Specify the product and the exact version, as well as the computer type and operating system if they are relevant. The subject can include the error message that you obtained. Write the description thoroughly.

If you have any files to attach, such as screenshots of scenes, or error message text files, you can do so once the case has been created.

Sending a Simulation Summary Report

You can send a summary report of your simulation to Siemens PLM Software support. For more information about simulation summaries, see Using Summaries. To send the report:

1. In STAR-CCM+, select **File** > **Summary Report** > **Save & upload summary report**.
2. In the *Save Summary Report To XML File* dialog, select the location on your hard drive to create an XML file.

   As soon as you click **Save**, STAR-CCM+ automatically uploads that XML file to the Siemens PLM Software server.
3. Look in the *Output* window for an *upload reference code*. Include that reference code in your correspondence with your support representative.
4. The support representative uses the code to access that report and view the settings in your simulation file.
5. If requested, Siemens PLM Software can delete the report based on the reference code or supply it back to you.

Suggesting Enhancements to STAR-CCM+

Siemens PLM Software provides an innovation forum, STAR-CCM+ IdeaStorm, on which you can submit new ideas for STAR-CCM+. The forum allows you to comment on ideas from other users and add your votes.

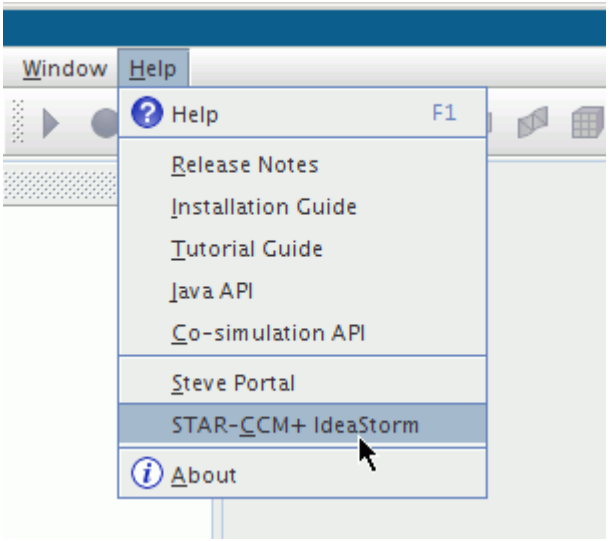
As a STAR-CCM+ user, you can participate in this forum in the following ways:

- Adding ideas
- Searching for ideas
- Commenting and voting on ideas
- Tracking ideas by subscribing to them

Access STAR-CCM+ IdeaStorm at:

http://thesteveportal.brightidea.com

Alternatively, select **Help** > **STAR-CCM+ IdeaStorm**.



STAR-CCM+ IdeaStorm is not intended for the following input:

- Discussions on support issues or guidelines -- visit the customer success portal (see Getting Support).
- Reporting possible bugs -- contact your Dedicated Support Engineer (DSE).

The screenshot below shows the Home page of STAR-CCM+ IdeaStorm.

Detailed information on participating in STAR-CCM+ IdeaStorm is available in the IdeaStorm User Guide.

To access it:

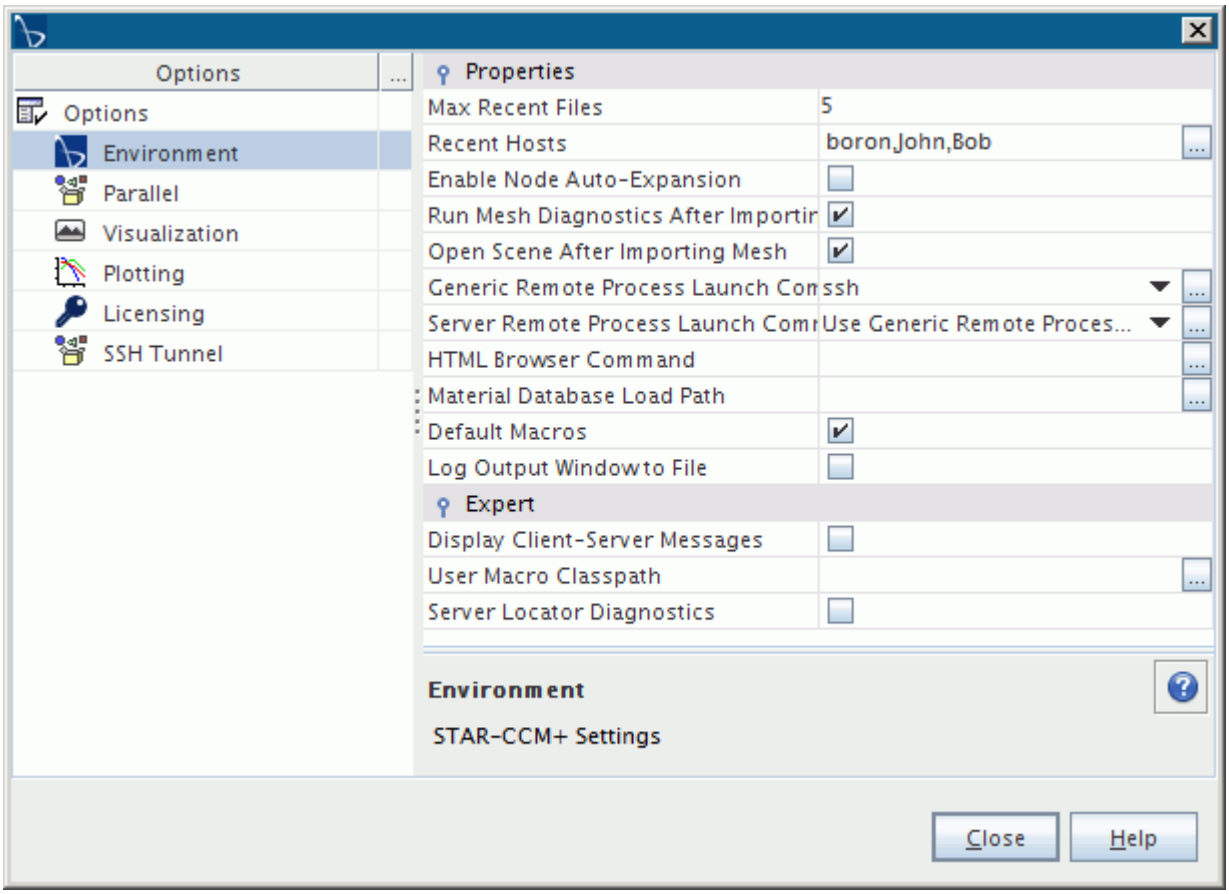
1. Enter STAR-CCM+ IdeaStorm.
2. Click **FAQ** on the top navigation menu.
3. Under the *Extra Links* heading, click **IdeaStorm User Guide**.

STAR-CCM+ Default Options

The *Options* dialog is used to control a number of global options in STAR-CCM+.

To access this dialog, select **Tools** > **Options** in the menu bar.

The *Options* dialog appears.



The options are divided into categories:

- Environment
- Parallel
- Visualization
- Plotting
- Licensing
- SSH Tunnel

These nodes are listed below with their properties.

Environment

Use **Environment** properties to control features such as mesh diagnostics, the path of the material database, and where information in the *Output* window goes.

Properties	
Max Recent Files	Sets a limit on the number of files listed in the **File** > **Recent Files** sub-menu.
Recent Hosts	Comma-separated list of host names for STAR-CCM+ to query for running servers.
Enable Node Auto Expansion	When this option is activated, the most commonly used nodes in the object tree are expanded automatically.
Run Mesh Diagnostics After Importing Mesh	When activated, STAR-CCM+ runs mesh diagnostics when you import a mesh.
Open Scene After Importing Mesh	When activated, STAR-CCM+ automatically creates a geometry scene when you import a mesh.
Generic Remote Process Launch Command	Sets the system command that STAR-CCM+ uses for launching remote servers from the client. Choose from the following options: • **rsh** • **ssh** (default) • **PuTTY** • **Custom** To further configure each option, click ... (**Custom Editor**) to reveal an associated *Environment* dialog. In this secondary dialog, depending on the launch option, you can set the *Remote Process Launch Command*, the *Command Path*, any *Username Specification Option*, and other *Command Options*. If the *Command Path* is empty, STAR-CCM+ assumes that the command is accessible within your environment. STAR-CCM+ retains any changes that you make within the *Environment* dialog for a launch option. You can reset the *Environment* entries by clicking **Reset Default Options.**
Server Remote Process Launch Command	Sets the system command that STAR-CCM+ uses to launch remote processes from the main server (such as when running in parallel or for co-simulation). The default option is to use **ssh**. You can use the same command as defined for the generic remote process, or choose an alternative (see *Generic Remote Process Launch Command*).
HTML Browser Command	Sets the path to the web browser that STAR-CCM+ launches when you access the user guide, either by pressing **F1** on an object, or from the **Help** menu.
Material Database Load Path	Specifies one or more paths from which to load material databases. To enter multiple paths, separate them with semicolons.

Properties	
Default Macros	When activated, STAR-CCM+ saves a default macro for every interactive session.
Log Output Window to File	When this option is activated, saves the contents of the *Output* window to a file.

Expert Properties	
Display Client-Server Messages	When this option is activated, the *Output* window includes client-server messages.
User Macro Classpath	Specifies one or more paths from which to load `.jar` files for compiling and running macros. To enter multiple paths on Windows, separate them with semicolons. On other systems, use colons.
Server Locator Diagnostics	When this option is activated, the *Servers* window includes network diagnostics output.

Parallel

This category contains the options that control the settings for parallel computing in STAR-CCM+.

Properties		
Parallel Hosts	A list of host machines that are selected by default for parallel jobs.	
Default Machine File	The name of a machine file that is selected by default for computing with a cluster.	
Parallel Host Specification Method	The default method for specifying the host computers for parallel jobs.	
	Local	The host process is on the local machine.
	Specify Hosts	Uses the list of hosts that you enter in the *Parallel Hosts* property.
	Machine File	Uses the machine file that you specify in the *Default Machine File* property.

Expert Properties	
Parallel Log File	When this option is activated, STAR-CCM+ generates log files for parallel events.
Parallel Output	When this option is activated, the *Output* window includes information from parallel worker processes. Output only appears if the client started the server. If the server was started independently of the client the output is displayed in the console window where the server was started.

Visualization

This category contains the options that control how the model is displayed in the *Graphics* window, and how you interact with it.

Properties		
Interactor Style	Selects a style for interacting with the *Graphics* window.	
	STAR-Default	For more information on these interactor options, see Using the Mouse in the Visualization Display.
	Pro-E/NX	
	SolidWorks	
	Trackball	
	Joystick	
Refresh Rate	Specifies the rate (frames per second) of refreshing the display.	
Show Framerate	When this option is activated, an annotation in the scene indicates the framerate.	
Default Camera Projection Mode	Choose between Perspective and Parallel modes for rendering objects in the *Graphics* window.	
	Perspective	Displays the object with a vanishing point.
	Parallel	Displays the object with edges parallel to each other.
Highlight Parts in Scene	When this option is activated, any part you select in the object tree is also highlighted in the graphics scene.	
Include Captions With Highlights	When this option is activated, STAR-CCM+ displays the names of highlighted parts in the graphics scene.	
Maximum Highlight Captions	Sets the maximum number of labels that can appear in the graphics scene at any one time.	
Highlight Caption Font	Selects the font to use for captions. To access a dialog that specifies font and style, click ... (**Custom Editor**).	
Annotate Scene Interaction	When this option is activated, annotations appear that describe your keyboard and mouse interaction in the scene.	
Logo Opacity	Controls the opacity of the STAR-CCM+ logo in the scene.	
Enable 3Dconnexion Devices	Allows 3Dconnexion 3D mice to control the display view. This feature is available on Windows only.	
3Dconnexion Sensitivity	Enter a value between 0.0 and 1.0. 0.0 is not very sensitive, while 1.0 is very sensitive.	
GPU Utilization	Lets you control globally what loads you put on your GPU. The lowest setting tries to use as few features on the GPU as	

Properties		
	possible, whereas the highest setting tries to use all the GPU capacity that is available.	
	Compatibility (0)	For more information see Controlling Graphics Performance.
	Default (1)	
	Unmanaged (2)	
	Opportunistic (3)	
	Custom	
Default Colormap	Specifies which colormap is used in new displayers. For details about the colormaps that are available by default, see Pre-Defined Colormaps.	

Expert Properties	
Edge Picking Tolerance	Tolerance used for edge picking, specified as a fraction of the diagonal size of the display.
Software Picking Tolerance	Tolerance used for surface picking when not using hardware, specified as a fraction of the diagonal size of the display.
Triangulate Part Surfaces	When this option is activated, triangulation eliminates visual artifacts that result from non-convex polygons in a part surface.
Use Hardware Picking	When this option is activated, graphics hardware is used for picking -- this is the recommended technique. Using this option usually results in faster picking, though it may not be fully supported on all systems.
Rotation Tool Step Size (degrees)	Sets the amount of degrees in one step by the scene rotation tool.
Use Implicit Representation	When this option is activated, certain shapes (for example, spheres and cylinders) are rendered using their implicit representations. This technique has the potential to increase performance, increase image quality, and reduce memory requirements. Otherwise, these shapes are represented using an approximation method such as triangulation.
Disable Gradient Background	When this option is activated, all scenes appear with no gradient background. Otherwise, each scene controls its own background.

Plotting

This category controls the default behavior of plots.

Properties	
Antialiasing	When activated, smooths the edges of text, lines, and symbols for improved on-screen viewing. Otherwise leaves edges sharp, which is appropriate for printing.
Line Width	Defines the default line width for data series within the plot display.
Symbol Size	Defines the default symbol size for data series within the plot display.
Title Font	Selects the default font for the title of a plot.
Legend Font	Selects the default font for the legend of a plot.
Axis Titles Font	Selects the default font for the titles of the plot axes.
Axis Label Font	Selects the default font for the labels of the plot axes.
Y-Axis Title Rotation	Specifies the default rotation for the title of the y-axis.
Major Grid Width	Specifies the default line width for the major grid lines of the plot.
Major Grid Style	Specifies the default line style for the major grid lines of the plot.
Major Grid Color	Specifies the default line color for the major grid lines of the plot.
Tick Count	Specifies the number of minor grid ticks between the major grid lines of the plot.
Minor Grid Width	Specifies the default line width for the minor grid lines of the plot.
Minor Grid Style	Specifies the default line style for the minor grid lines of the plot.
Minor Grid Color	Specifies the default line color for the minor grid lines of the plot.

Licensing

This category contains the add-on licensing options. From here, you can reserve the desired add-on licenses.

Properties	
Reserve Battery Simulation Module License	These options reserve the respective licenses when you create or open a simulation that requires any of them.
Reserve STAR-Cast License	
Reserve IC Engine Module License	
Reserve JTOpen Reader License	
Reserve CAD Exchange License	

Properties	
Retry Licenses on Requiring Operations	Makes additional attempts to access the license when needed.
Release Licenses After Geometry Import	Allows other users to access the license once you have finished importing geometry into the simulation.

SSH Tunnel

With this category, you can pre-configure the SSH options so that they are automatically applied each time that you want to use SSH tunneling with STAR-CCM+.

Properties		
SSH Tunnel Host	The host used for tunneling client-server connections. This machine is typically the head node of a cluster.	
User Name on Tunnel Host	Your user name on the tunnel host.	
SSH Tunnel Command	Selects a command for launching the ssh tunnel.	
	ssh	For more specifications of each option, click ... (**Custom Editor**).
	PuTTY	
	Custom	

Expert Properties	
SSH Tunnel Verbosity	Controls the verbosity level of output for the ssh tunnel. These settings can be used for debugging.

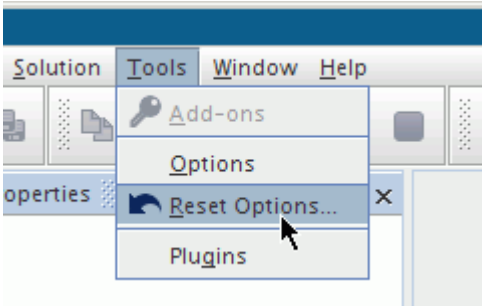
See Also:

Changing the Appearance of the Scene Highlight Font

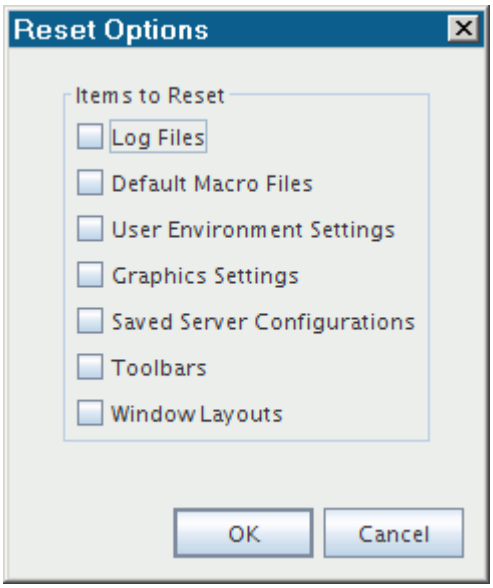
Resetting STAR-CCM+ Default Options

There may be occasions when changes to the STAR-CCM+ workspace are no longer needed, such as when a project ends or when the client computer passes to a different user. There are multiple methods for restoring the default settings.

In the **Tools** menu, select **Reset Options**.



In the *Reset Options* dialog, you can choose to revert changes to log files, the appearance of windows, and other features.



Select what to reset and click **OK**.

Alternatively, the full range of items can be reset by using the `-reset` line command as described in the Command-Line Reference.

Finally, this action can also be achieved by deleting the user settings directory in either Linux or Windows. You may need to take this step in case you attempt to launch STAR-CCM+ with an unsupported version of Java that is installed on your computer. This attempt can corrupt user settings files, causing the STAR-CCM+ workspace to appear incomplete.

Resetting Defaults on Linux

Exit the program, delete the hidden folder `.star-*-*-*` (for example, `.star-12.06.007` for STAR-CCM+ Version 12.06.007) on your `home` directory, and then restart the program.

To find the complete path to this directory, select **Help** > **About...** and look at the User Settings Directory: line in the *About STAR-CCM+* dialog.

Resetting Defaults on Windows

Delete the folder `STAR-CCM+ *-*-*` (for example, `STAR-CCM+ 12.06.007` for STAR-CCM+ Version 12.06.007) in `C:\Users\[login name]\AppData\Local\CD-adapco`.

To find the complete path to this directory, select **Help** > **About...** and look at the User Settings Directory: line in the *About STAR-CCM+* dialog.

Controlling Graphics Performance

STAR-CCM+ offers various options for using graphics hardware for your simulation.

The GPU Utilization feature lets you control globally what loads you put on your GPU. The lowest setting tries to use as few features on the GPU as possible, whereas the highest setting tries to use all the GPU capacity that is available.

Note: This is a client-side setting which is not stored in a particular simulation file.

To set the GPU Utilization:

1. Select **Tools** > **Options**.
2. In the *Options* dialog, select the **STAR-CCM+** > **Visualization** node.
3. To set the *GPU Utilization* property, do either of the following:
   - Select an option from the drop-down list of the *GPU Utilization* property. The numbering after the name of each option ranks the level of demand on the GPU:
     - **Compatibility (0)** -- see Rendering from CPU Memory.
     - **Default (1)** -- see Rendering with Managed GPU Memory.
     - **Unmanaged (2)** -- see Rendering with Unmanaged GPU Memory.
     - **Opportunistic (3)** -- see Rendering with Opportunistic GPU Memory.
   - Create a customized version of one of the GPU utilization options:
     1. Set the *GPU Utilization* property to one of the options in the drop-down list.
     2. Click ... (**Custom Editor**) for the *GPU Utilization* property. In the dialog that appears, activate the **Use custom settings** option.
     3. Activate the settings that you want for the custom option.
     4. Click **OK**.

        A **Custom** option appears in the drop-down list.

Rendering from CPU Memory

The **Compatibility (0)** option is the equivalent of activating the *Immediate Mode* expert property of displayers in STAR-CCM+ 9.06 and earlier versions. It is recommended for older graphics cards.

On Linux platforms, if Mesa rendering is detected, STAR-CCM+ automatically uses this option.

Rendering with Managed GPU Memory

With the **Default (1)** option, graphics data is stored on the GPU, but the STAR-CCM+ client manages GPU memory allocation. This option is a compromise between robustness and speed.

Default (1) is the functional equivalent of deactivating the *Immediate Mode* expert property of displayers in STAR-CCM+ 9.06 and earlier versions. However, the GPU memory management is designed to provide the most stable rendering of the GPU utilization options. This option is especially recommended if you use `-hardwarebatch`.

Rendering with Unmanaged GPU Memory

With the **Unmanaged (2)** option, graphics data is stored on the GPU, but only the GPU driver attempts to handle the graphics memory allocation. This option is faster than the default as soon as a large amount of GPU memory is required, such as for volume rendering or large isosurfaces. However, the unmanaged setting is a little less reliable than the default setting since older GPUs or drivers can crash when taxed too strongly by an application.

Caution: Consider this option only if you have the latest graphics card, or at least a recent one, along with up-to-date drivers for it.

Rendering with Opportunistic GPU Memory

The **Opportunistic (3)** option attempts to use a GPU to its maximum capacity. Currently this option only makes a difference with how volume rendering allocates memory.

Caution: Consider this option only if you have the latest graphics card, or at least a recent one, along with up-to-date drivers for it.

Troubleshooting Graphics

This section provides techniques for dealing with the following visualization problems:

- Resolving hardware issues
- Working without hardware acceleration
- A scene does not open
- Concave faces display badly
- Z-fighting appears

Contents:

Resolving Hardware Issues
Working Without Graphics Hardware
A Scene Will Not Open
Improving the Rendering of Concave Faces
Z-Fighting Appears
STAR-CCM+ Fails to Detect Graphics Card (Linux Only)

Resolving Hardware Issues

If you experience problems in displaying graphics within the STAR-CCM+ *Graphics* window, try the solutions listed here.

- Install the latest driver for your graphics hardware. The process for obtaining and installing the latest driver depends on your graphics hardware. Refer to the manufacturers instructions for details.
- On Windows, decrease the hardware acceleration slider in the *Troubleshooting* tab of the advanced display properties dialog. If the graphics display becomes slow as a result, contact your support representative.

Some specific issues reported in the past are:

- With some older NVIDIA drivers, memory leaks can occur when using OpenGL display lists to render scene graphics. OpenGL display lists can be used when the *GPU Utilization* feature is set to **Compatibility (0)** and the settings are customized. To fix the memory leaks, try upgrading to the latest driver, or alternatively switch to **Default (1)** for GPU utilization. (See section, Controlling Graphics Performance.)
- On Linux systems, the 169.XX series of NVIDIA drivers are unstable and are not supported. Upgrade to the latest series of drivers.
- On Windows laptops using certain NVIDIA graphics chips (such as the Quadro FX770M), STAR-CCM+ can crash, hang, or show unexpected artifacts in the graphics display. If you experience problems:
  - upgrade to the latest graphics driver for your graphics card
  - choose the **Dassault Systemes CATIA - compatible** option for the global 3D setting in the NVIDIA control panel

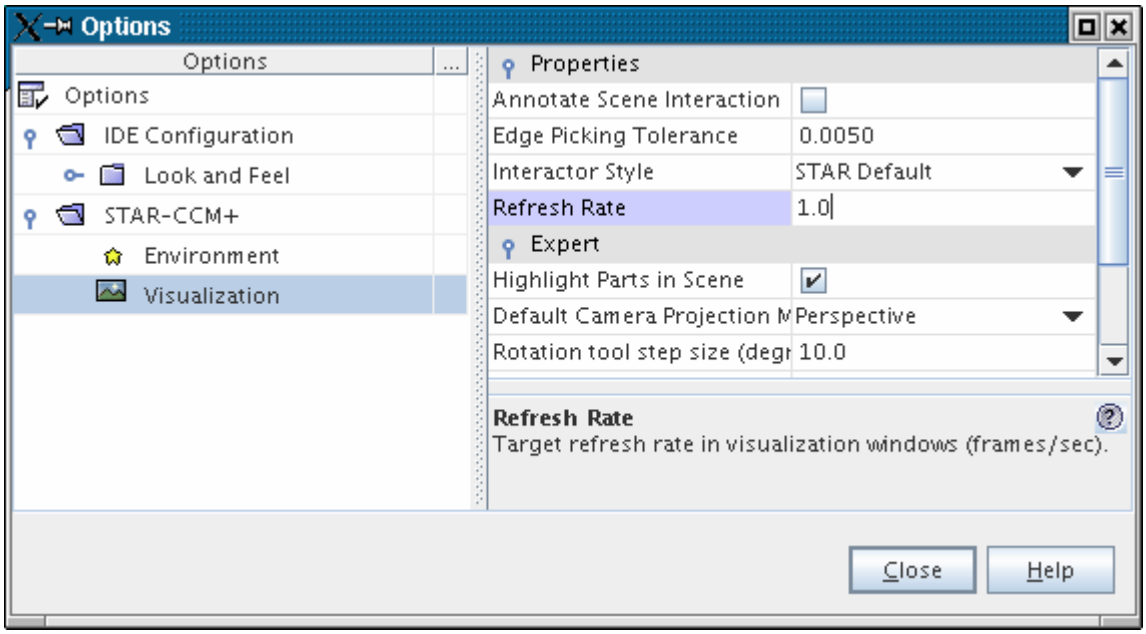
Working Without Graphics Hardware

When hardware-accelerated drivers are not available on your machine, or when you must run STAR-CCM+ remotely through an X-server (such as VNC), you can activate Mesa graphics libraries.

The Mesa graphics libraries provide software rendering without requiring hardware acceleration. Graphics performance is much slower when using Mesa than when using graphics hardware with up-to-date hardware-accelerated drivers. To explicitly force the use of software rendering with Mesa, launch STAR-CCM+ from a command prompt with the `-mesa` option appended.

To increase responsiveness:

1. Select **Tools** > **Options**. In the *Options* dialog that appears, open the **Options** and **STAR-CCM+** nodes, and select the **Visualization** node.



2. Increase the *Refresh Rate* property.

   This adjustment demotes surfaces to a simpler representation (and hence drawn faster) when interaction occurs in the scene.

Limitations of Software Rendering with Mesa

Software rendering is known to be much slower than hardware rendering. When rendering with advanced visualization options, such as Line Integral Convolution (LIC), the time to render a scene can be 100 times longer than with hardware rendering. Consequently, you are strongly advised to run STAR-CCM+ with hardware rendering whenever graphics hardware is present.

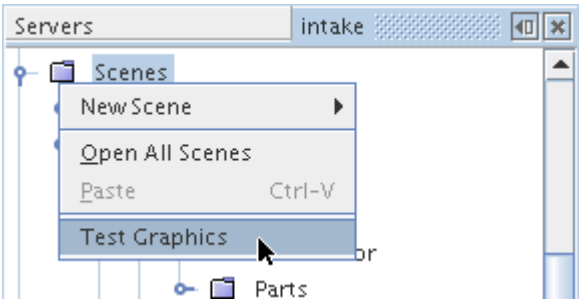
For volume rendering, if your graphics card does not support it, activate advanced rendering for the relevant scene.

Note: You can enable hardware rendering in batch (assuming that you have the appropriate graphics hardware) with the `-hardwarebatch` option.

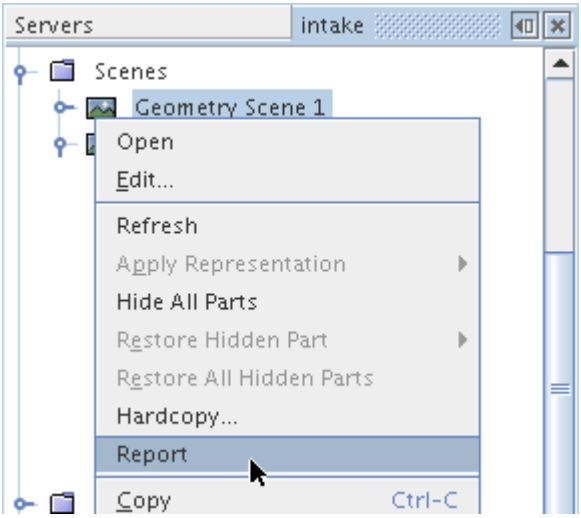
A Scene Will Not Open

If you are unable to open a scene, you can obtain information for troubleshooting.

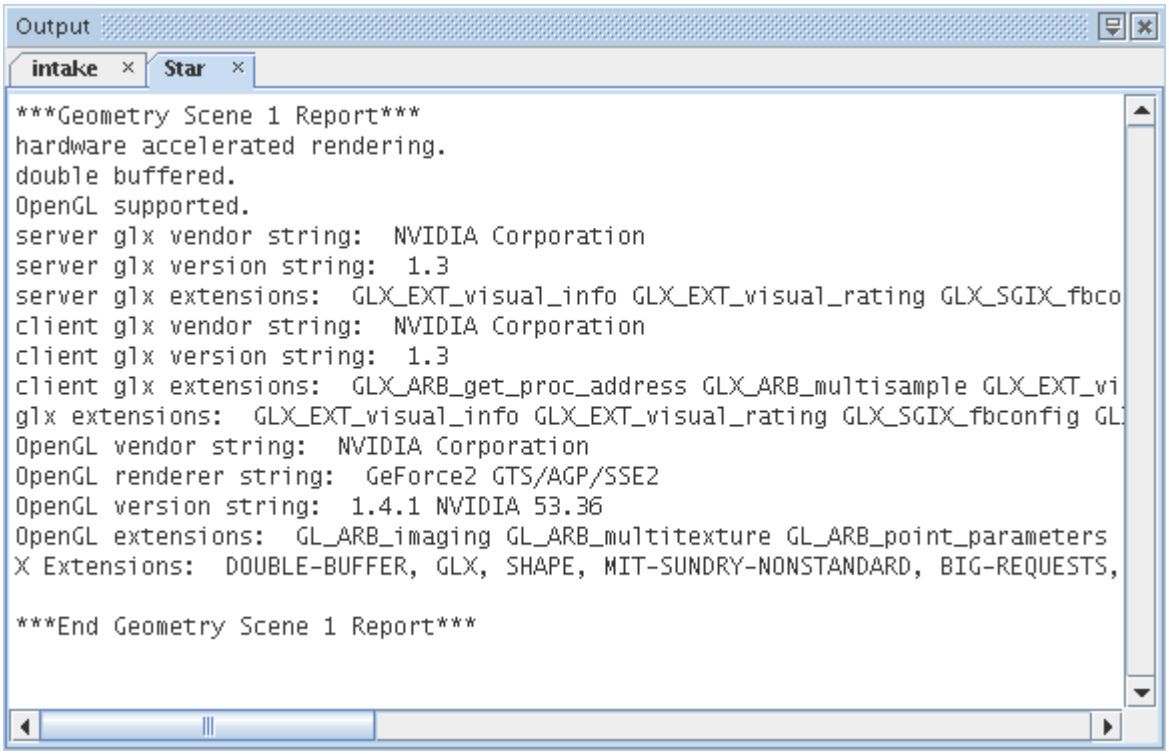
1. Right-click on the Scenes node and select **Test Graphics**.



2. Otherwise, right-click on the node of the particular scene and select **Report**.



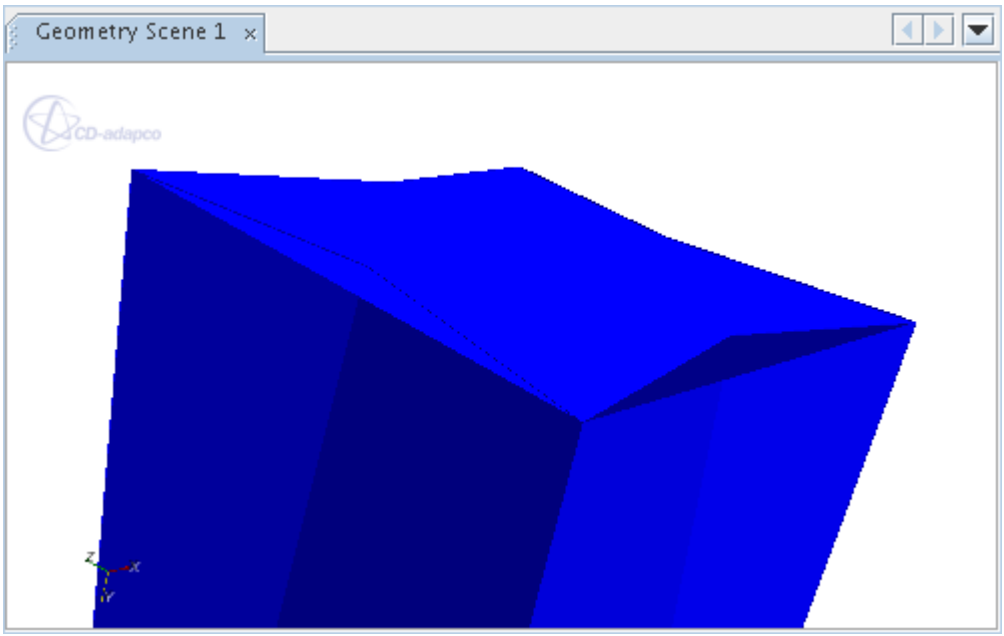
Either command produces a report in the *Output* window containing technical details on the graphics hardware.



If you wish, simply send the information to support by selecting the text in the *Output* window, copying it and pasting it into an e-mail message.

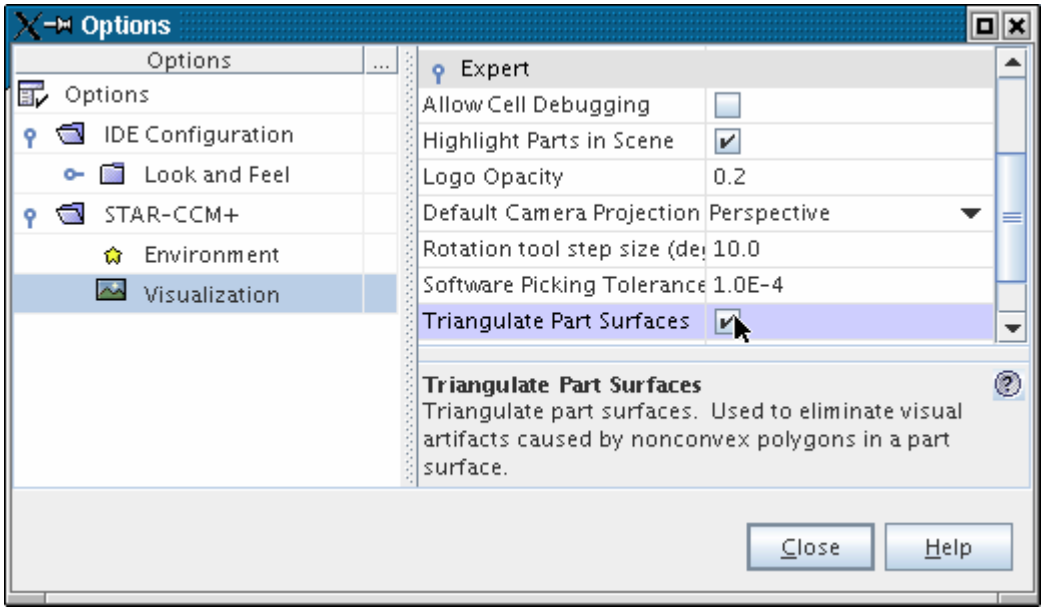
Improving the Rendering of Concave Faces

Some objects with highly non-planar polygons show artifacts. This problem is because STAR-CCM+ attempts to triangulate these kinds of surfaces--a process that requires extra memory.



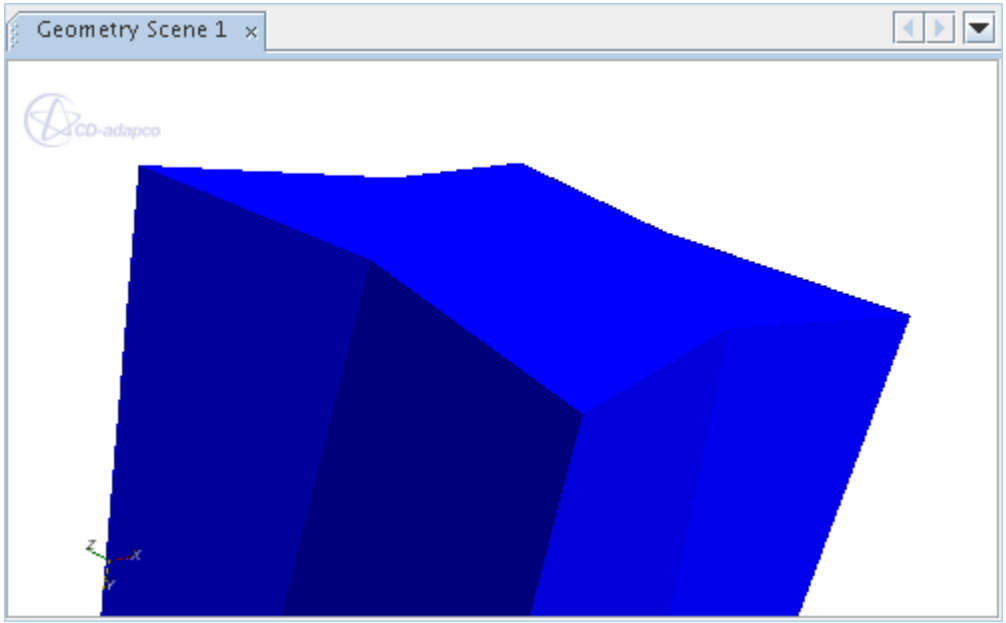
To activate triangulation:

1. Select **Tools** > **Options**. In the *Options* dialog that appears, select the **Options** > **STAR-CCM+** > **Visualization** node.
2. Activate the checkbox of the *Triangulate Part Surfaces* property.



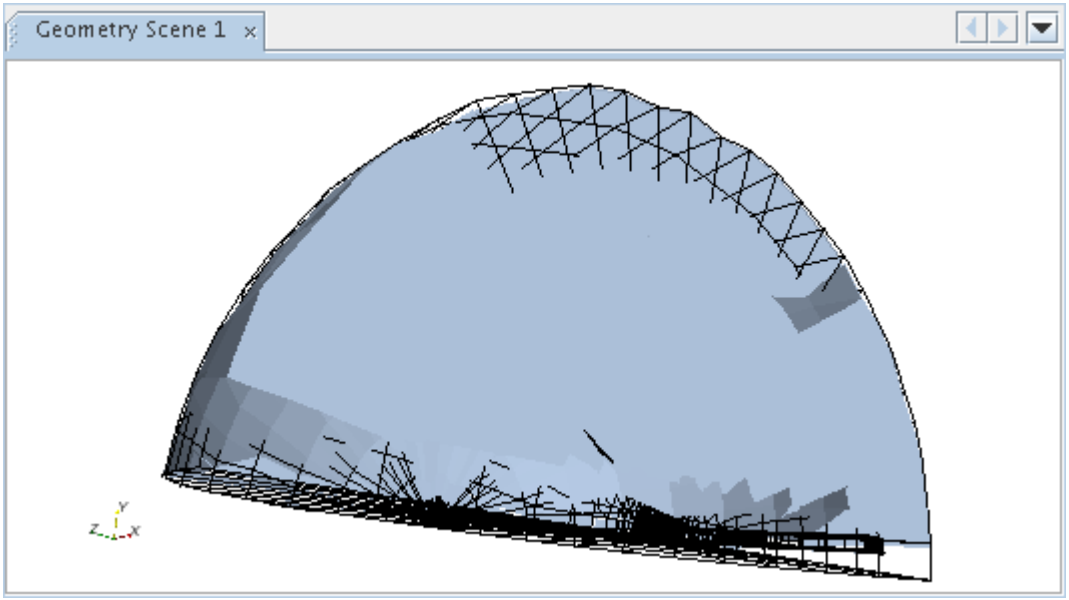
Use this option sparingly, since increased memory usage slows graphics rendering.

The triangulated surfaces appear in the scene display.



Z-Fighting Appears

The model shows rear polygons “bleeding” from behind the front ones.



This problem is known as z-fighting. The “fighting” occurs in three-dimensional rendering when multiple polygons are in the same plane or nearly so, and their pixels displace one another as you move the image in the scene. The competing portions of these polygons are called artifacts.

In general the cause of z-fighting is insufficient precision in the z-buffer, the element of the computer hardware/software that is supposed to manage the depth of the image (in the z-direction). Yet there are multiple aspects to this problem that require different solutions:

- Hardware/software requirements

  If your depth buffer size is less than 32 bits, then an upgrade of your graphics hardware is necessary to manage the depth imaging properly.
- Multiple surfaces in the same plane (creating a shimmering effect)

  Review the distances among the various parts of your model. If they are close enough, they could be treated as being in the same plane. Also, make sure that if a part is in multiple displayers in the same scene, then only one of those displayers (at most) has surface mode activated.
- Excessive viewing distance

  In rare cases the problem can be due to the model being shifted too far from the origin. Typically distances of millions of units are required to cause this effect, although these distances can be shorter if you are using obsolete graphics hardware. To eliminate excessive distance, shift the model back to the origin.

STAR-CCM+ Fails to Detect Graphics Card (Linux Only)

On Linux, STAR-CCM+ relies on a utility, *glxinfo*, to verify that the computer has a graphics card installed with OpenGL support. If STAR-CCM+ fails to find an OpenGL graphics card and driver then it renders plots and scenes using the mesa software rendering library.

When glxinfo is not available, STAR-CCM+ assumes that hardware rendering is not supported and reverts to mesa, a software rendering library. The following message indicates this change:

```
Warning: glxinfo unable to obtain openGL support. Rendering plots with
mesa.
```

Note: This rendering technique can cause plots to display slower.

If you know that your Linux machine has a graphics card installed, but that STAR-CCM+ is not detecting it, you can try two possible remedies:

- Make glxinfo available by installing the software mesa-utils.
- Force STAR-CCM+ to circumvent the check for glxinfo using the line command `-clientldpreload` as in the following example:

  ```
  ./bin/starccm+ <simfile> -clientldpreload .
  ```

  The `-clientldpreload` command is used normally for specifying an OpenGL library file of your choice. In this instance, the period (".") takes the place of the file name.

  Using the command in this fashion yields several instances of the following error message during the STAR-CCM+ startup:

  ```
  ERROR: ld.so: object '.' from LD_PRELOAD cannot be preloaded:
  ignored.
  ERROR: ld.so: object '.' from LD_PRELOAD cannot be preloaded:
  ignored.
  ```

  Once the STAR-CCM+ client is running, the error message stops. This message does not interfere with the launch or function of the STAR-CCM+ client.

Credits

STAR-CCM+ makes use of several third party software components to provide certain features within its code.

Some of these components are as follows:

- HOOPS Exchange translator v10.1.20 from Tech Soft 3D
- Parasolid Kernel Modeler v30.00.167 by Siemens PLM Software, Inc
- DCM v66.6 by Siemens PLM Software, Inc.
- IBM Platform MPI

lp_solve

STAR-CCM+ references the `lp_solve` library within its code. This library is governed by the GNU Lesser General Public Licence. Full citation data and disclaimer notice is given as follows:-

```
lpsolve citation data----------------------

Description : Open source (Mixed-Integer) Linear Programming system

Language : Multi-platform, pure ANSI C / POSIX source code, Lex/Yacc
based parsing

Official name : lp_solve (alternatively lpsolve)

Release data : Version 5.1.0.0 dated 1 May 2004

Co-developers : Michel Berkelaar, Kjell Eikland, Peter Notebaert

Licence terms : GNU LGPL (Lesser General Public Licence)

Citation policy : General references as per LGPL. Module specific
references as specified therein
```

The `lp_solve` library is distributed "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

RSyntaxTextArea and RSyntaxTextArea: Autocomplete

Copyright (c) 2012, Robert Futrell

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

sundials

Copyright © 2002, The Regents of the University of California. Produced at the Lawrence Livermore National Laboratory. All rights reserved.

This file is part of SUNDIALS.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the disclaimer below.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the disclaimer (as noted below) in the documentation and/or other materials provided with the distribution.

3. Neither the name of the UC/LLNL nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OF THE UNIVERSITY OF CALIFORNIA, THE U.S. DEPARTMENT OF ENERGY OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Additional BSD Notice

1. This notice is required to be provided under our contract with the U.S. Department of Energy (DOE). This work was produced at the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-ENG-48 with the DOE.

2. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately-owned rights.

3. Also, reference herein to any specific commercial products, process, or services by trade name, trademark, manufacturer or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

SuperLU

Copyright (c) 2003, The Regents of the University of California, through Lawrence Berkeley National Laboratory (subject to receipt of any required approvals from U.S. Dept. of Energy)

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

(1) Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

(2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

(3) Neither the name of Lawrence Berkeley National Laboratory, U.S. Dept. of Energy nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LAPACK

LAPACK++ 1.1a Linear Algebra Package 1.1a

University of Tennessee, Knoxvilee, TN.

Oak Ridge National Laboratory, Oak Ridge, TN.

Authors: J. J. Dongarra, E. Greaser, R. Pozo, D. Walker

(C) 1992-1996 All Rights Reserved

NOTICE

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice appear in supporting documentation.

Neither the Institutions (University of Tennessee, and Oak Ridge National Laboratory) nor the Authors make any representations about the suitability of this software for any purpose. This software is provided ``as is'' without express or implied warranty.

LAPACK++ was funded in part by the U.S. Department of Energy, the National Science Foundation and the State of Tennessee.

MUMPS

This version of MUMPS is provided to you free of charge. It is public domain, based on public domain software developed during the Esprit IV European project PARASOL (1996-1999). Since this first public domain version in 1999, research and developments have been supported by the following institutions: CERFACS, CNRS, ENS Lyon, INPT(ENSEEIHT)-IRIT, INRIA, and University of Bordeaux.

The MUMPS team at the moment of releasing this version includes Patrick Amestoy, Maurice Bremond, Alfredo Buttari, Abdou Guermouche, Guillaume Joslin, Jean-Yves L'Excellent, Francois-Henry Rouet, Bora Ucar and Clement Weisbecker.

We are also grateful to Emmanuel Agullo, Caroline Bousquet, Indranil Chowdhury, Philippe Combes, Christophe Daniel, Iain Duff, Vincent Espirat, Aurelia Fevre, Jacko Koster, Stephane Pralet, Chiara Puglisi, Gregoire Richard, Tzvetomila Slavova, Miroslav Tuma and Christophe Voemel who have been contributing to this project.

Up-to-date copies of the MUMPS package can be obtained from the Web pages: http://mumps.enseeiht.fr/ or http://graal.ens-lyon.fr/MUMPS

THIS MATERIAL IS PROVIDED AS IS, WITH ABSOLUTELY NO WARRANTY EXPRESSED OR IMPLIED. ANY USE IS AT YOUR OWN RISK.

User documentation of any code that uses this software can include this complete notice. You can acknowledge (using references [1] and [2]) the contribution of this package in any scientific publication dependent upon the use of the package. You shall use reasonable endeavours to notify the authors of the package of this publication.

[1] P. R. Amestoy, I. S. Duff, J. Koster and J.-Y. L'Excellent, A fully asynchronous multifrontal solver using distributed dynamic scheduling, SIAM Journal of Matrix Analysis and Applications, Vol 23, No 1, pp 15-41 (2001).

[2] P. R. Amestoy and A. Guermouche and J.-Y. L'Excellent and S. Pralet, Hybrid scheduling for the parallel solution of linear systems. Parallel Computing Vol 32 (2), pp 136-156 (2006).

fastapprox

Copyright (C) 2011 Paul Mineiro

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

(1) Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

(2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

(3) Neither the name of Paul Mineiro nor the names of other contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Contact: Paul Mineiro <paul@mineiro.com>

JSON

Copyright (c) 2002 JSON.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The Software shall be used for Good, not Evil.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

ANTLR 4

Copyright (c) 2012 Terence Parr and Sam Harwell

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

JFreeChart

This software incorporates JFreeChart, (C)opyright 2000-2009 by Object Renery Limited and Contributors.

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the

Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

ISAT-CK7

The ISAT-CK7 package contains several Fortran libraries developed by Ithaca Combustion Enterprise, LLC and by the Turbulence and Combustion Group at Cornell University. This package is made available under the BSD 3-Clause license. STAR-CCM+ uses only ISATAB and CEQ which were written by Steve Pope and whose copyright is held by Ithaca Combustion Enterprise, LLC. The full citation for this library is:

S.B. Pope, V. Hiremath, S.R. Lantz, Z. Ren and L. Lu (2012) ISAT-CK7: "A Fortran 90 library to accelerate the implementation of combustion chemistry." http://tcg.mae.cornell.edu/ISATCK7

Copyright © 2012, Ithaca Combustion Enterprise, LLC All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the Ithaca Combustion Enterprise, LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Textures

Twelve texture image products are supplied with STAR-CCM+ as part of its Environment Maps feature:

- **Desert Road**
- **Dirt Road**
- **Fall Road**
- **Heroes Square**
- **Lake At Sunset**
- **Meadow At Night**
- **Mountain Road**
- **Pebbles**
- **Snow Road**
- **Vancouver Convention Centre**
- **Warehouse**
- **Warehouse Interior**

Two of these, **Heroes Square** and **Vancouver Convention Centre**, are provided by Humus (http://www.humus.name/index.php?page=About) under a Creative Commons Attribution 3.0 license (links: summary | license).

The others are provided under license by HDRMAPS (http://hdrmaps.com/).

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code Guidelines Provided by Pharr, M., Jakob, W., and Humphreys, G. 2016. *Physically Based Rendering*, 3rd Ed. Morgan Kaufmann.

Copyright (c) 1998-2015, Matt Pharr, Greg Humphreys, and Wenzel Jakob.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Embree

STAR-CCM+ references the Embree library within its code. This library is released under the Apache License v2.0 (http://www.apache.org/licenses/LICENSE-2.0).

```
Copyright [yyyy] [name of copyright owner]
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
See the License for the specific language governing permissions and
limitations under the License.

Permission is hereby granted, free of charge, to any person obtaining
a copy
of this software and associated documentation files (the "Software"),
to deal
in the Software without restriction, including without limitation the
rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell
copies or substantial portions of the Software.

The above copyright notice and this permission notice shall be
included in
```

```
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT
SHALL
PAWEL W. OLSZTA BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR
IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

Except as contained in this notice, the name of Pawel W. Olszta shall
not be
used in advertising or otherwise to promote the sale, use or other
dealings
in this Software without prior written authorization from Pawel W.
Olszta.

Copyrights:
(c) is the Copyright property of CompuServe Incorporated.
Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.
Copyright (c) 1996-1997 Andreas Dilger
Copyright (c) 1998-2000 Glenn Randers-Pehrson
Copyright (c) 1999-2000 Pawel W. Olszta. All Rights Reserved.
Copyright (c) 2000-2002, 2004, 2006-2016 Glenn Randers-Pehrson
Copyright (c) 2007-2012 The Khronos Group Inc.
Copyright 1999-2016 ImageMagick Studio LLC
Copyright 2009-2012 Intel Corporation
Copyright 2009-2015 Intel Corporation
Copyright 2009-2016 Intel Corporation
Copyright by M.I.T. but is also freely distributable.
copyright (C) 1991-1998, Thomas G. Lane. All Rights Reserved
© 2009 2016 Intel Corporation Intel, the Intel logo, Xeon, Intel Xeon
Phi, and Intel Core are trademarks of Intel Corporation in the U.S.
and/or other countries.
```

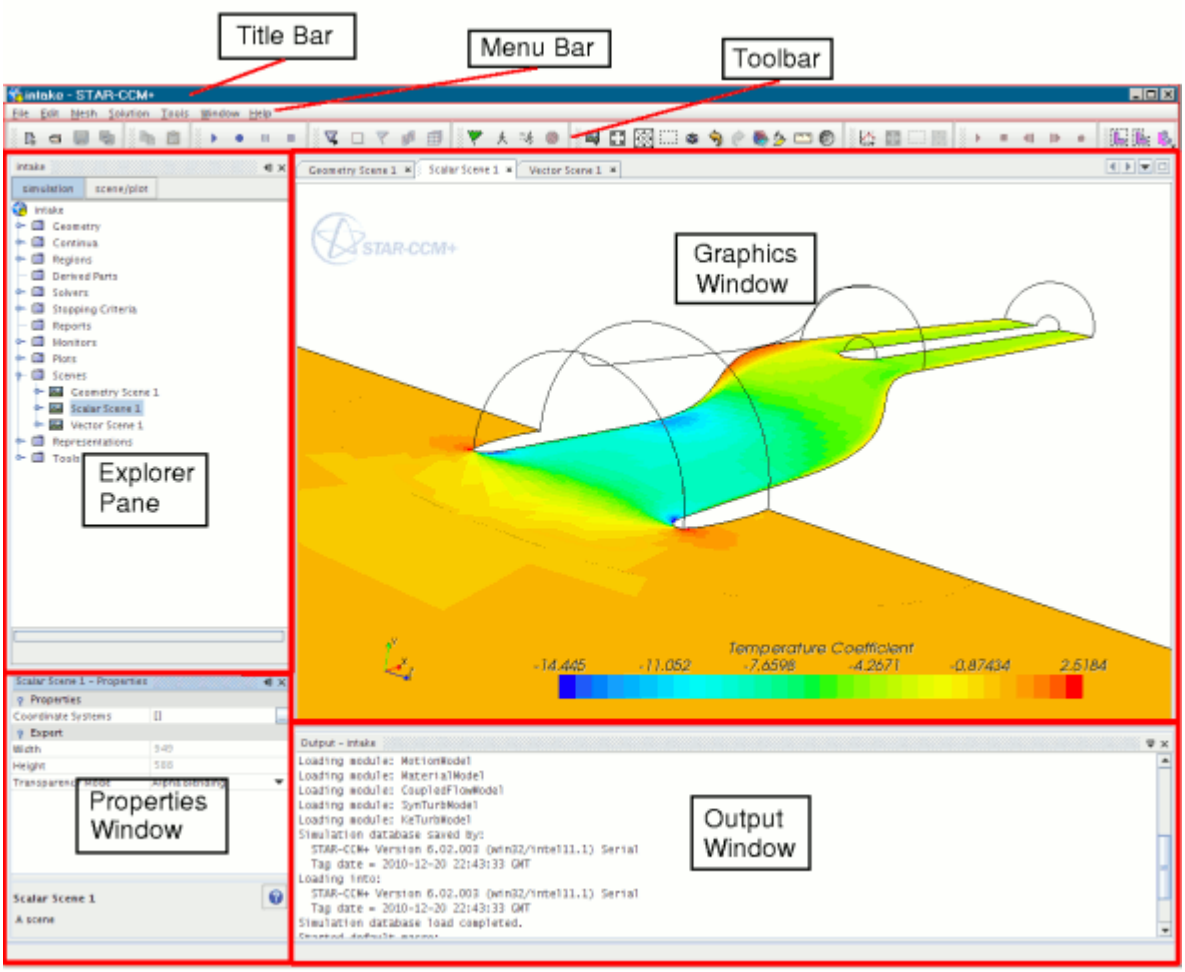
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING

NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

User Interface

STAR-CCM+ provides a Graphical User Interface generally known as the client workspace. This interface appears when you launch STAR-CCM+ interactively.

The key elements of this interface are shown in the clickable image below. You can jump to the documentation for different elements by clicking them.



1. Using the Explorer Pane
2. Using the Graphics Window
3. Using the Properties Window

4. Using the Output Window
5. Using the Toolbars
6. Using the Menus

Contents:

Using the STAR-CCM+ Workspace
What Are the Important Concepts?
Working with Simulation Objects
Global Parameters
Field Functions
Engineering Units
Scripting the Application
Working with User Code
Working with the Plugins Manager
Simulation Assistants
Wizard Guide

Using the STAR-CCM+ Workspace

This chapter shows you how to use the various components of STAR-CCM+, with a focus on the user interface.

Alternatively, if you prefer to run applications with line commands, see the Command Line Reference.

Note: The STAR-CCM+ workspace lets you display with unicode characters, change the scene highlight font, and change the workspace font size.

Specific topics addressed here are:

Contents:

Using the Workspace
Using the Menus
Using the Toolbars
Using the Explorer Pane
Using the Table Editor
Using the Properties Window
Using the Graphics Window
Using the Output Window
Using Keyboard Shortcuts
Conditions, Values, and Profiles
Using Tables

Using the Workspace

If you have used any GUI-based applications (such as Microsoft Word), you will have little difficulty using the STAR-CCM+ workspace.

It includes many widgets and operations that have become commonplace, such as tree views, toolbars, menu bars, tabs, docking windows, drag-and-drop and pop-up menus. Assuming you are familiar with basic operations, such as working with windows and opening files, the following sections provide information on:

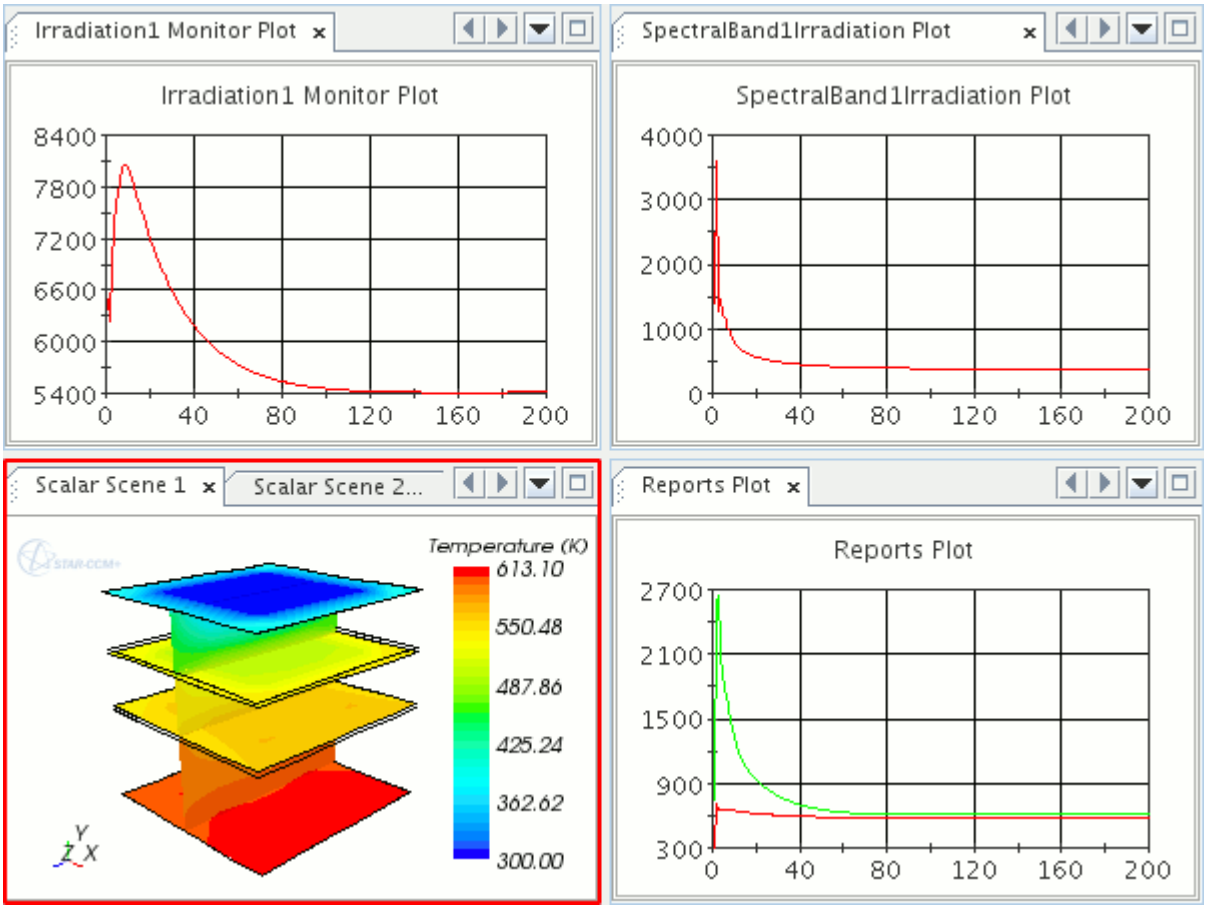
Contents:

Preserving the Window Layout
Defining Colors
Pasting Coordinate Sets Directly into Dialog Fields
Identifying the Version of STAR-CCM+
Working with Unicode Characters
Changing the Appearance of Decimals
Changing the Appearance of the Scene Highlight Font
Changing the Workspace Font Size
Navigating the Workspace with the Keyboard

Preserving the Window Layout

A window layout is an arrangement of windows in the graphics display, which consists of one or more modes. They can be used to save, restore, and edit a specified visual display.

A layout mode is an individual section of the graphics display, which consists of scenes and plots. Modes can be edited, to add, or remove scenes and plots from the graphics display. As an example, the following graphics display has four modes, with one of them highlighted.

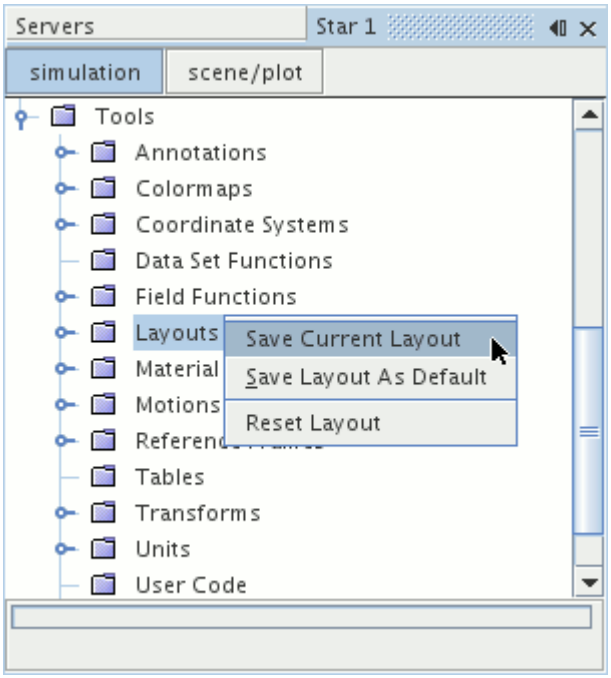


Modes can also be undocked from the graphics display. You can save the position on the screen with the layout.

Saving a Window Layout

Saving a window layout makes the current graphics display available for restore. When saved, a window layout can be edited to meet visualization requirements.

To save the current window layout as a new layout, right-click the **Layouts** manager, and select **Save Current Layout**.



A layout is added to the **Layouts** manager.

To save the current window layout as the default layout, right-click the **Layouts** manager, and select **Save Layout As Default**.

Alternatively, from the menu bar, select **Window** > **Save Layout As Default**.

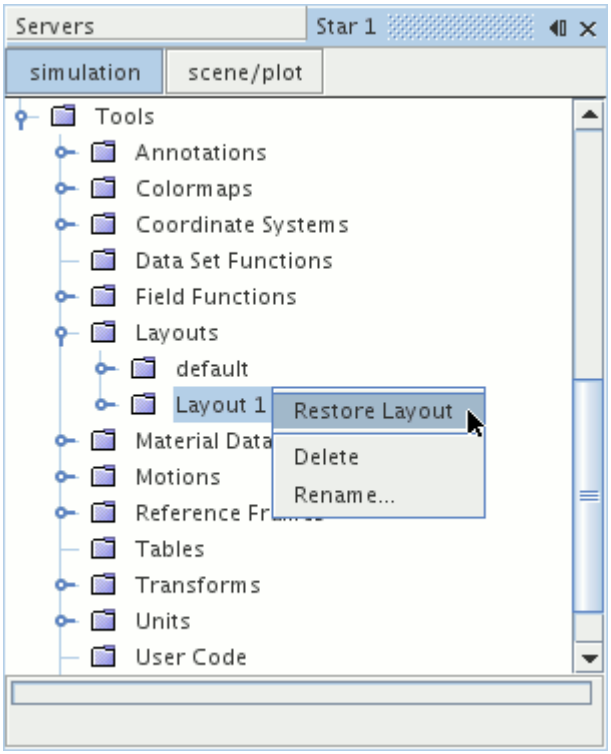
This action overwrites the default layout under the **Layouts** manager.

To save the layout as the default layout when the simulation is saved, from the menu bar, select **Tool** > **Options**. Then in the *Options* dialog, select the **Visualization** node, and activate the **Save layout on sim save** option.

Restoring a Window Layout

Restoring a window layout restores a saved layout in the graphics display.

To restore a layout, right-click a layout from within the **Layouts** manager, and select **Restore Layout**.



The saved window layout is shown in the Graphics window.

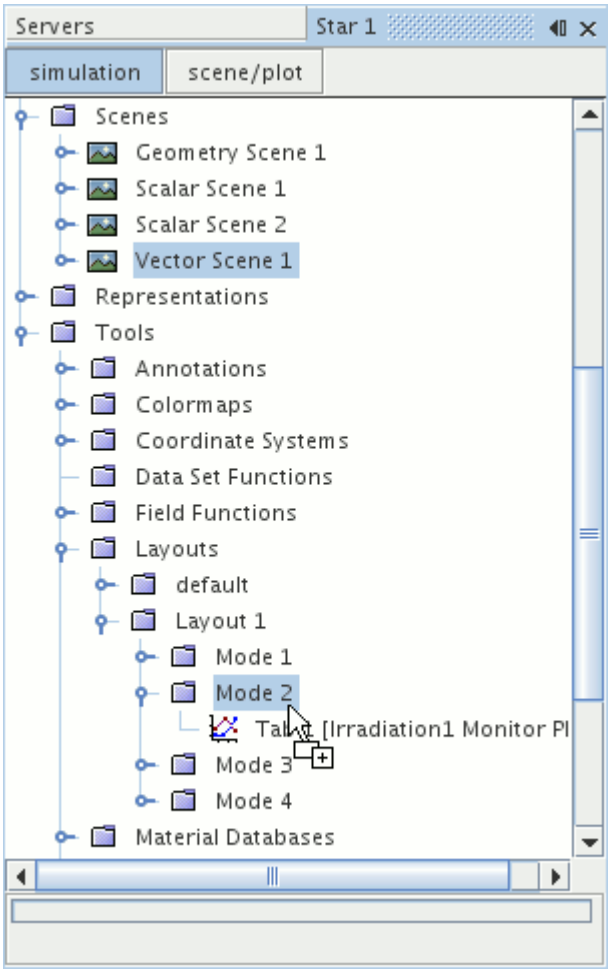
Alternatively, to restore only the default layout, from the menu bar, select **Window** > **Restore Default Layout**.

Editing a Window Layout

Once saved, window layouts can be edited to meet visualization requirements. Editing consists of adding, replacing, or removing scenes and plots from the layout modes.

Adding a Scene or Plot to Layout Modes

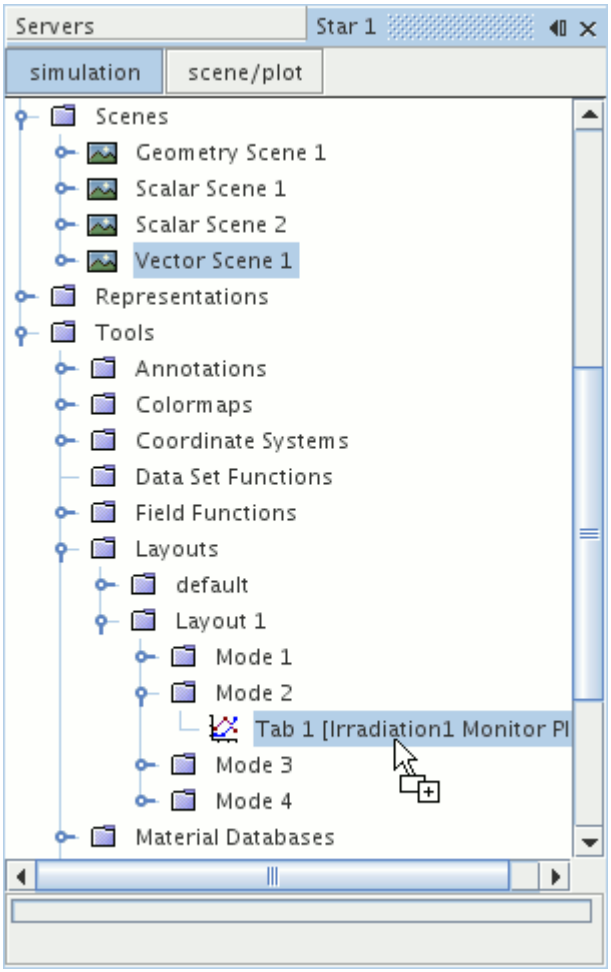
To add a scene or plot to a layout mode, select a scene or plot and drag the item to the **Mode** manager.



A node representing the scene or plot is then added under the **Modes** manager. The graphics display is updated when the scene is restored.

Replacing a Scene or Plot in Layout Modes

To replace a scene or plot in a layout mode, select a scene or plot and drag the item to the node that you want to replace.



A node representing the scene or plot then replaces the original node. The graphics display is updated when the scene is restored.

Removing a Scene or Plot from Layout Modes

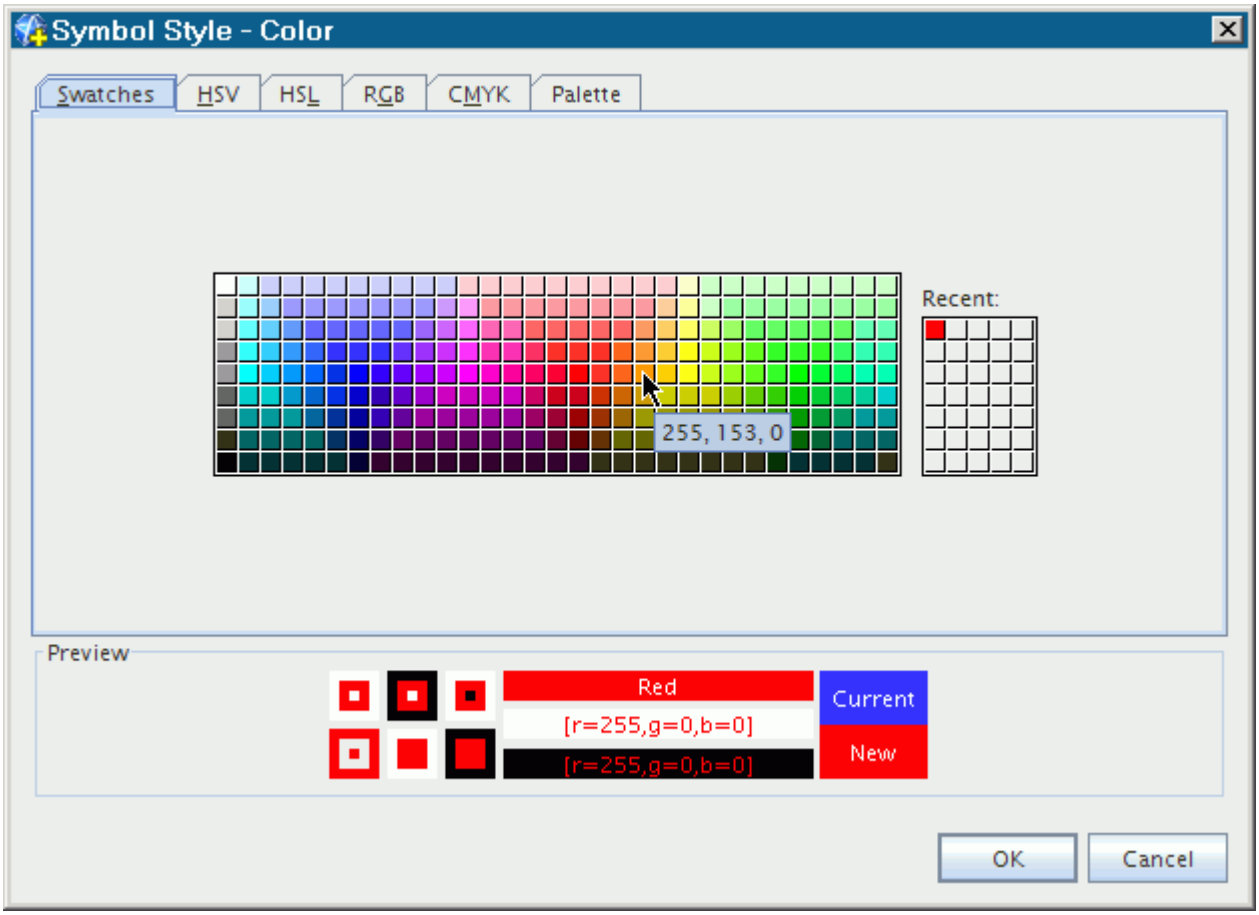
To remove a scene or plot from a layout mode, right-click the node in the layout mode representing the scene that you want to remove, and select **Remove from Layout**.

The node representing the scene or plot is then removed from the **Modes** manager. The graphics display is updated when the scene is restored.

Defining Colors

The dialog for color control is used for a wide variety of objects within STAR-CCM+.

The most common method for activating this dialog is clicking the property customizer ( ) in the color property of an object. The following example controls the color of symbols in the plot of a data set.

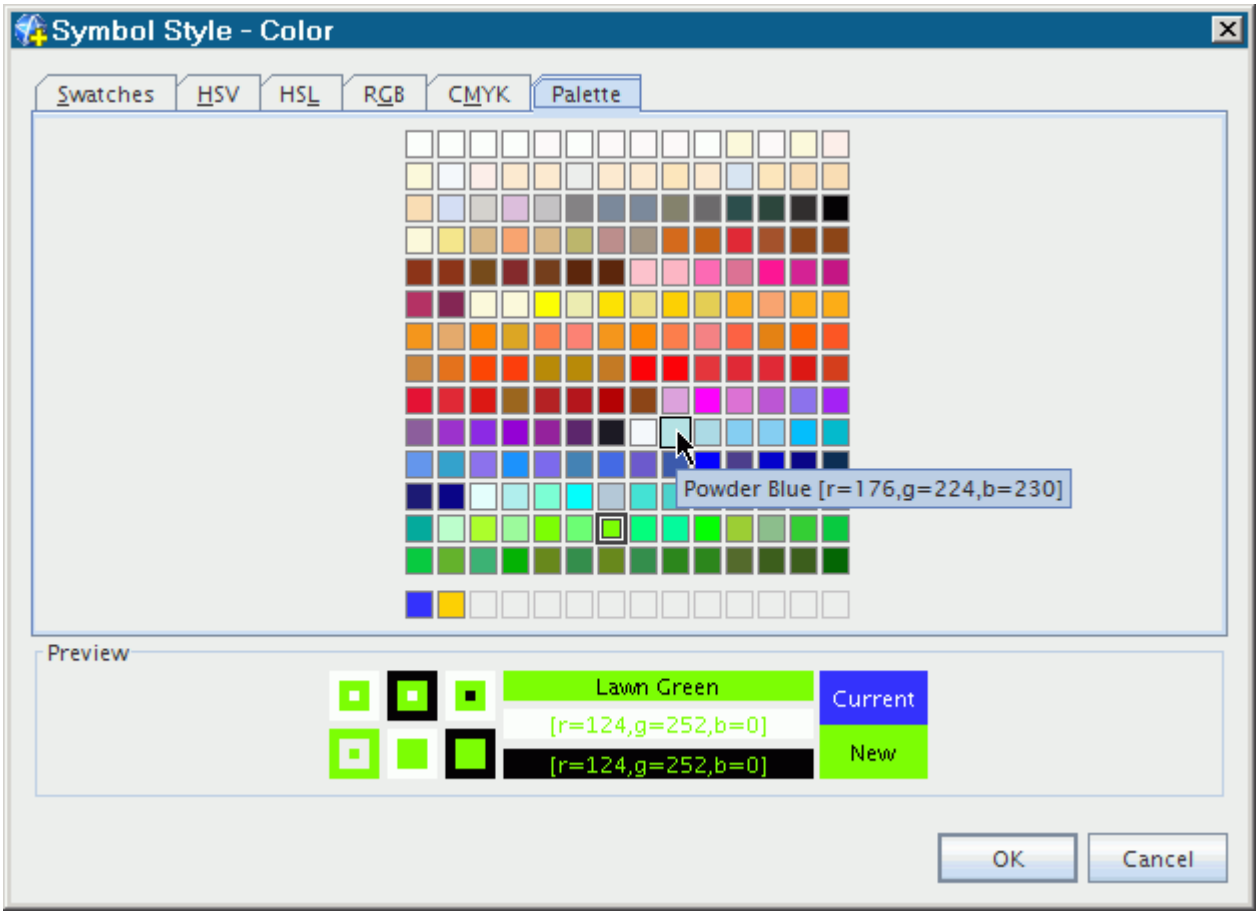


The *Swatches* tab provides a set of colors to choose from. As you hover with the pointer over the color squares, the RGB (Red, Green, and Blue) values appear in a tooltip. The smaller swatch that is labeled Recent shows colors that you selected. In the *Preview* group box, you can see how the color that you choose would appear in various forms before you confirm the change.

The following tabs provide you with the tools to customize your color selections with the precision of a graphics tool. They can be useful to generate output for a professional-quality printed report.

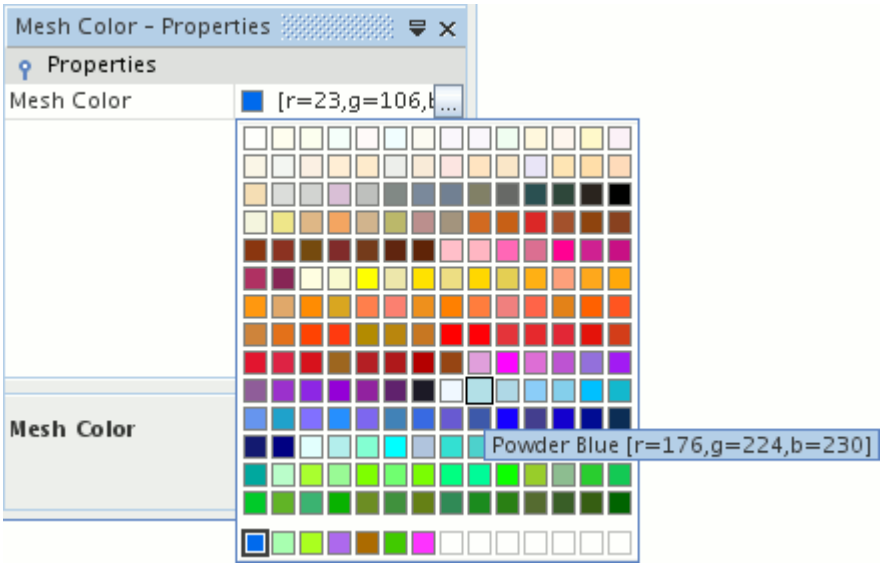
- *HSV* (Hue, Saturation, and Value) -- works like mixtures of paint
- *HSL* (Hue, Saturation, and Lightness) -- resembles perceptual color combinations
- *RGB* (Red, Green, and Blue) -- the color model commonly used in video displays
- *CMYK* (Cyan, Magenta, Yellow, and Black) -- the color model commonly used in printing

The *Palette* tab lets you choose from the colors that are used to design the STAR-CCM+ GUI.



Colors are added to the custom color history palette, below the standard color palette, when you select a color that is not in the palette. This history contains up to 14 colors and is saved during the session.

To access the color selection of the *Palette* tab directly in the *Properties* window, click in the right column of the color property.



Pasting Coordinate Sets Directly into Dialog Fields

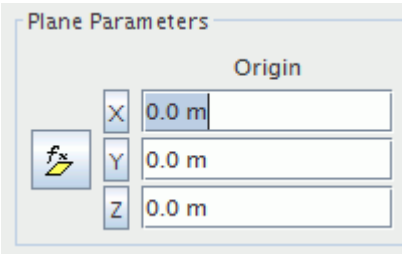
If you have a string of comma-separated coordinates, for example of a vector, you can paste them directly into a dialog with a single paste action.

You can paste the coordinates in any of the following formats:

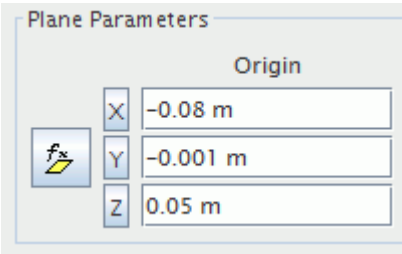
```
2.5 0.18 0.05
2.5, 0.18, 0.05
[2.5 0.18 0.05]
[2.5, 0.18, 0.05]
2.5 0.18 0.05 m
2.5, 0.18, 0.05 m
[2.5 0.18 0.05] m
[2.5, 0.18, 0.05] m
```

To use this feature:

1. Select the desired values, for example in the *Output* window or the *Properties* window, and copy them. In the example that is illustrated here, the values are `-0.08, -0.001, 0.05`.
2. Place the cursor in the first coordinate text box in the dialog (typically **X**) and paste.



All three values appear in the appropriate text boxes. The existing unit selection is applied to the values.

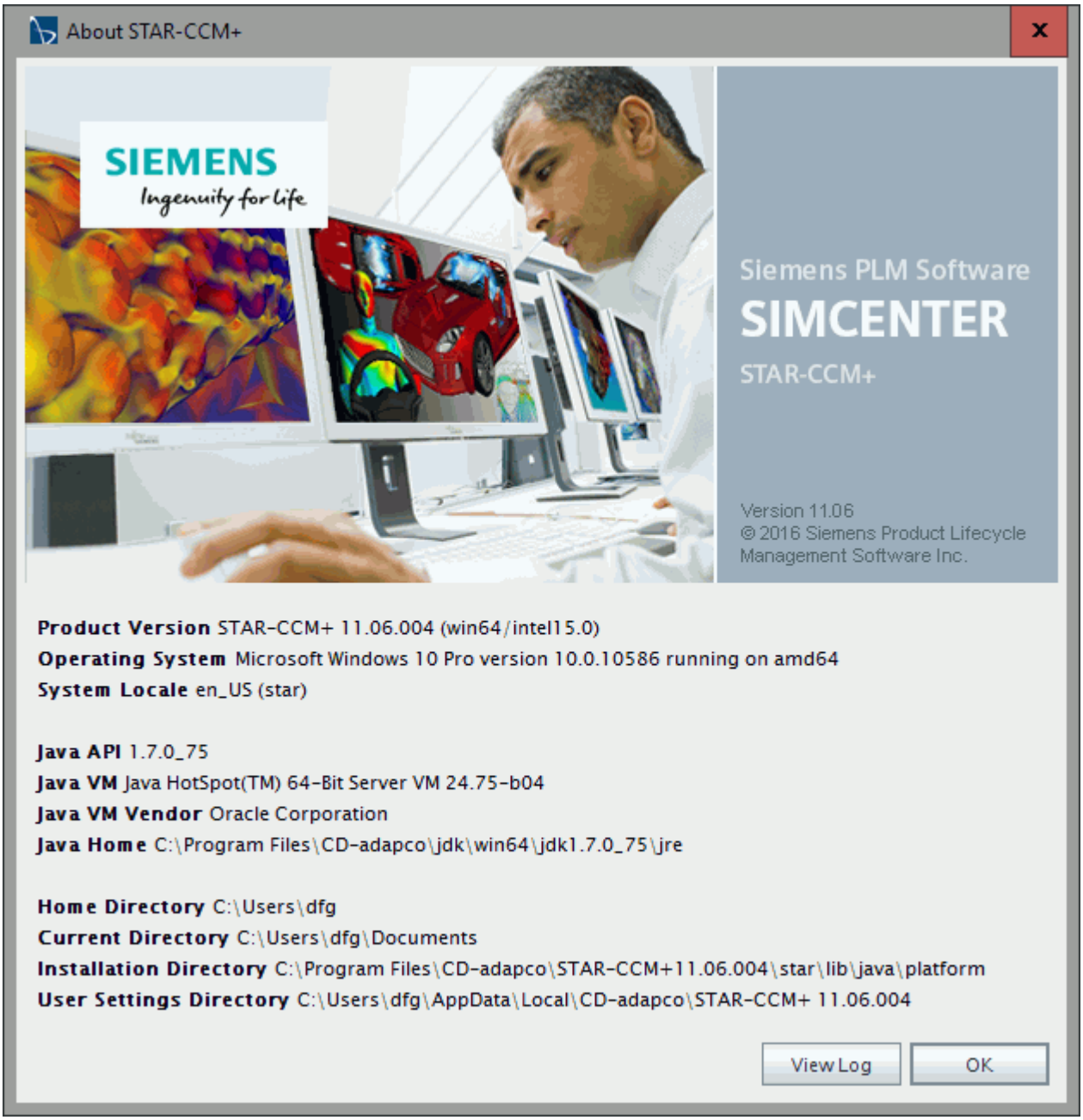


Note: In the current version of STAR-CCM+, strings with multiple units are not acceptable for pasting in this manner. To paste the string, simply omit the additional instances of units from what you paste. For example, change the string `[2.5, 0.18, 0.05] m,m,m` to `[2.5, 0.18, 0.05] m`.

Identifying the Version of STAR-CCM+

To find out which version of STAR-CCM+ you are running, select **Help** > **About**.

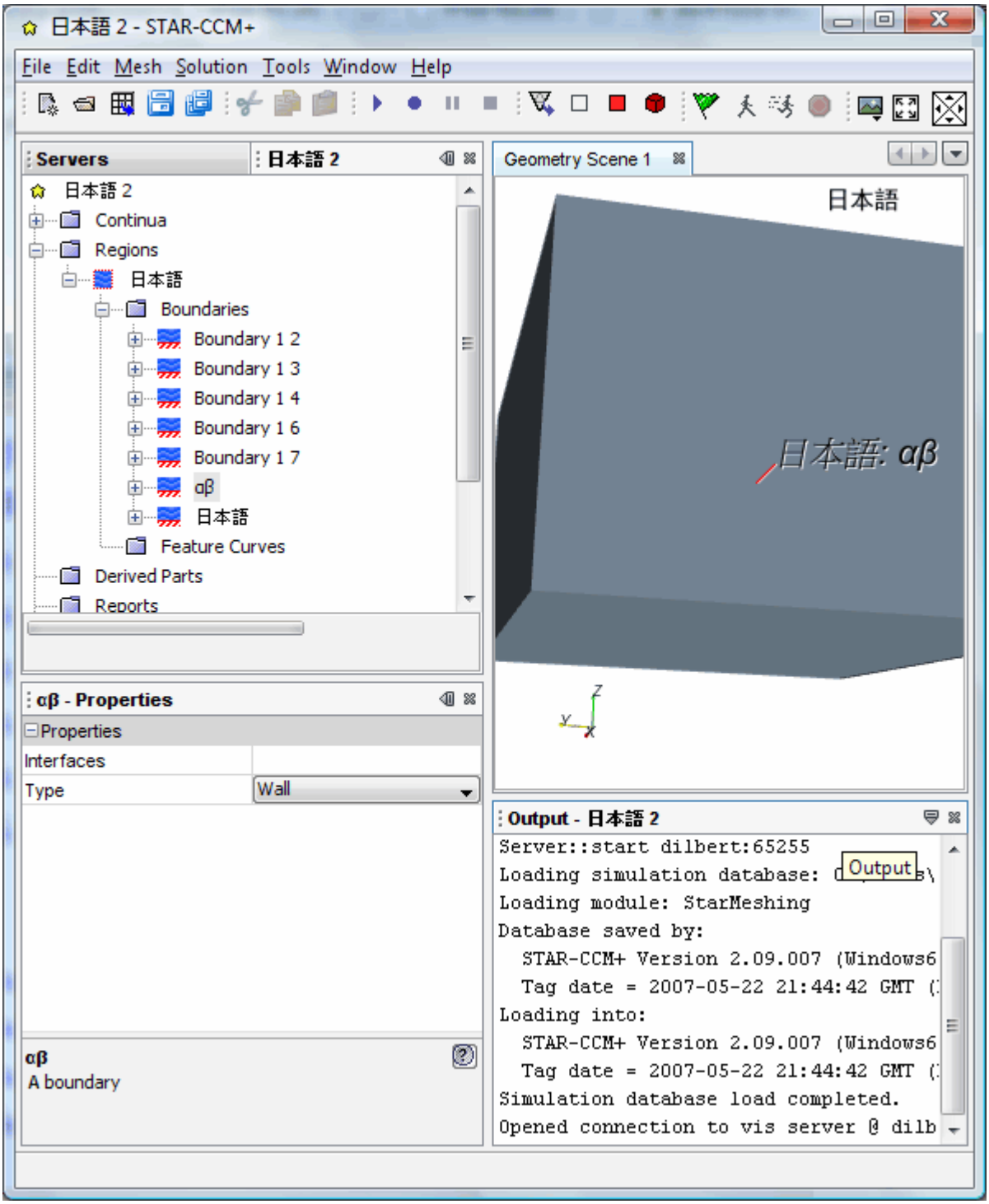
This action opens an information dialog with details about the code version.



You can click the **View Log** button to display an error log file in your default web browser. The information in this log can be useful for support purposes.

Working with Unicode Characters

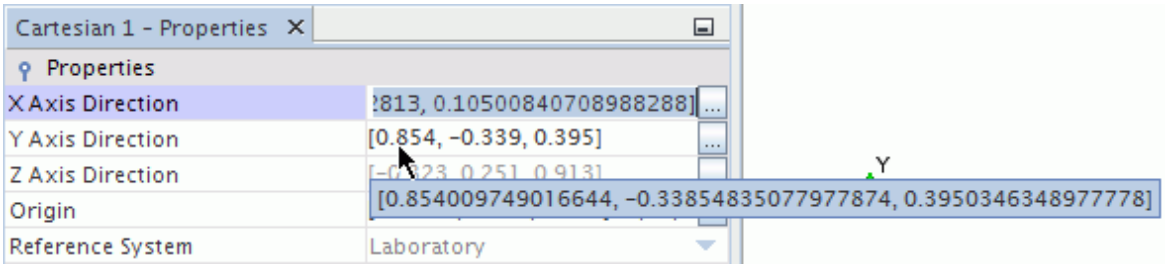
On systems that support unicode, named objects on the tree as well as scene annotations can include non-Western (UCS-16) character sets as shown below.



Changing the Appearance of Decimals

When you browse large amounts of data, it may seem cumbersome to view values in their lengthy double-precision format. For easier data viewing, change the decimal display.

Note: The decimal value is *not necessarily the actual value*. To check the actual value, hover with the mouse pointer for the tooltip or, where applicable, click in the value field.



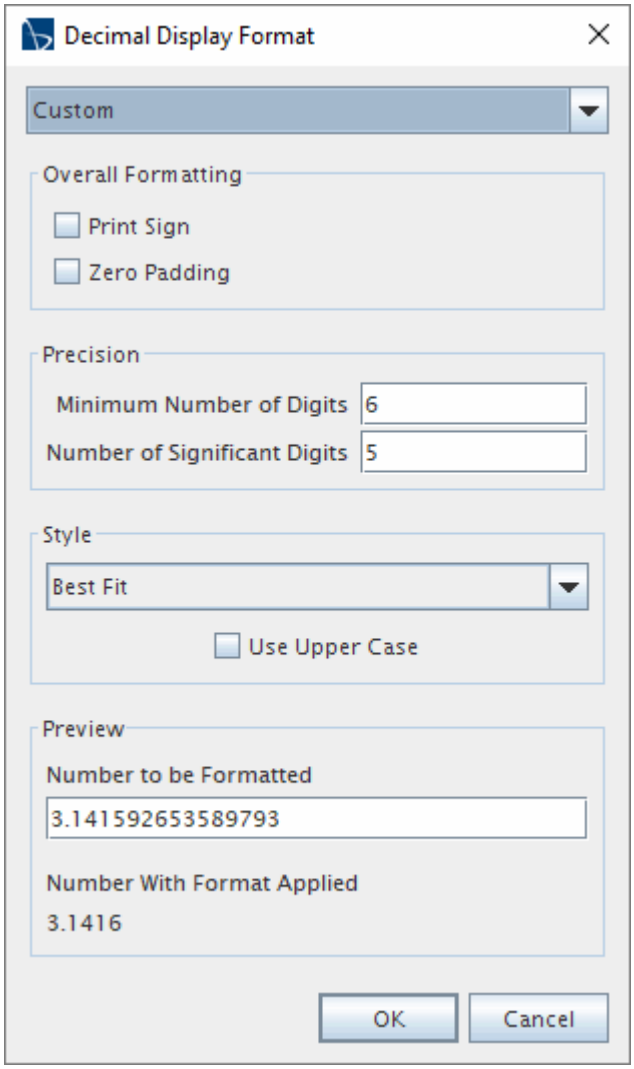
In the current version of STAR-CCM+, this feature only works with the *Properties* window value entries and the tabular data of a Design Manager study. For details about properties entries, see Common Editing Techniques. For details about using tabular data with Design Manager, see Creating Manual Designs and Tabulating Designs.

Choose from a number of standard formats, or create your own.

1. Select **Tools** > **Options**.
2. In the options dialog, select the **Services** > **Environment** node.
3. To set the *Decimal Display Format* property, do either of the following:
   - Use the drop-down list of the property:
     1. Select one of the following options (for these examples, Pi = 3.141592653589793).
        - **None**: the full double-precision numerical values are displayed; Pi is displayed as `3.141592653589793`.
        - **Short**: `0.###`; Pi is displayed as `3.142`.
        - **Long**: `0.#######`; Pi is displayed as `3.1415927`.
        - **Short Fixed Decimal**: `0.000`; Pi is displayed as `3.140`.
        - **Long Fixed Decimal**: `0.0000000`; Pi is displayed as `3.1400000`.
        - **Short Scientific**: `0.###E00`; Pi is displayed as `3.142e00` (converted to lower case).
        - **Long Scientific**: `0.#######E00`; Pi is displayed as `3.1415927e00` (converted to lower case).

        Scientific notation is applied when the value is greater than or equal to 100,000, or less than the decimal places of the format:

        - `Pi*100000.0` displayed in Short format appears as `3.142e05`.
        - `Pi/10000.0` displayed in Short format appears as `3.142e-04`.
     2. To preview the format, click ... (**Custom Editor**), and use the *Decimal Display Format* dialog.



In the *Preview* group box, compare the full value of the example in the **Number to be Formatted** text box to the value under **Number With Format Applied**. You can also enter your own value as the **Number to be Formatted**.

Note: Settings in the dialog such as **Precision** and **Style** are available for editing only when you select **Custom** in the drop-down list of the property or the dialog.

- Create a customized decimal display format option:
  1. Set the *Decimal Display Format* property to **Custom** in the drop-down list.
  2. Click ... (**Custom Editor**). In the *Decimal Display Format* dialog, specify the settings that you want for the custom option:
     - To indicate whether the values are positive or negative, activate the **Print Sign** checkbox.
     - To include zeros at the left of the labels (for a uniform number of digits), activate the **Zero Padding** checkbox.
     - To specify the minimum number of digits overall, and the number of digits after the decimal point, use the text boxes in the *Precision* group box. Press `<Enter>` after making an entry in each of these text boxes.

- In the *Style* group box, select a style from the drop-down list:
  - Best Fit—this default option presents the data in a visually optimal manner.
  - Decimal—this option displays the full decimal values based on the entry in the **Number of Decimal Digits** text box.
  - Scientific—this option displays the values in scientific notation. To use an uppercase “e”, activate the **Use Upper Case** checkbox.
- In the *Preview* group box, preview the effect of the format.

3. Click **OK**.

Changing the Appearance of the Scene Highlight Font

You can control the appearance of the font that is used to highlight objects in scene displays.

To change the font that appears in a scene display:

1. Select **Tools** > **Options**.
2. In the *Options* dialog, select the **STAR-CCM+** > **Visualization** node.
3. To set the *Highlight Caption Font* property, do either of the following:
   - Enter the name of the font that you require.
   - Click the property customizer button ( ). In the dialog that appears, select the font that you require.

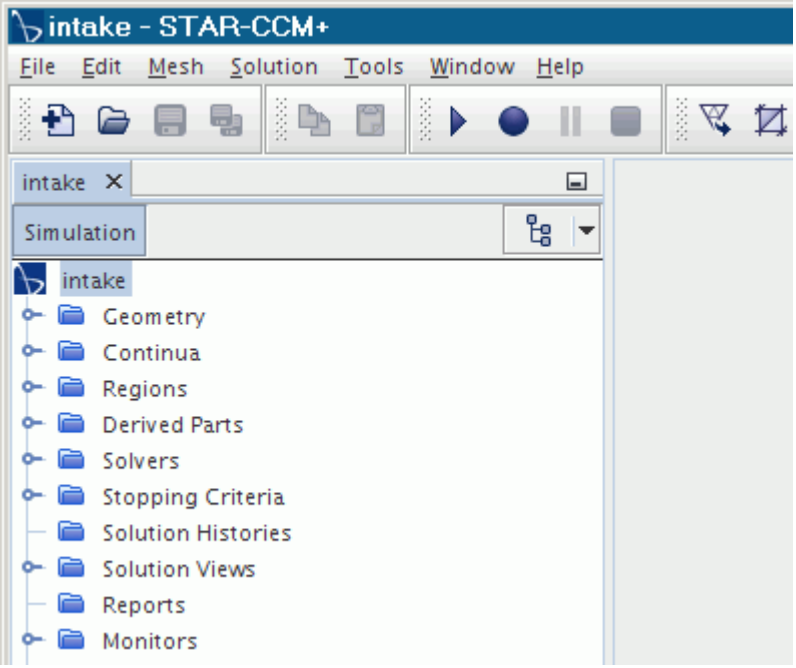
See Also:

STAR-CCM+ Default Options

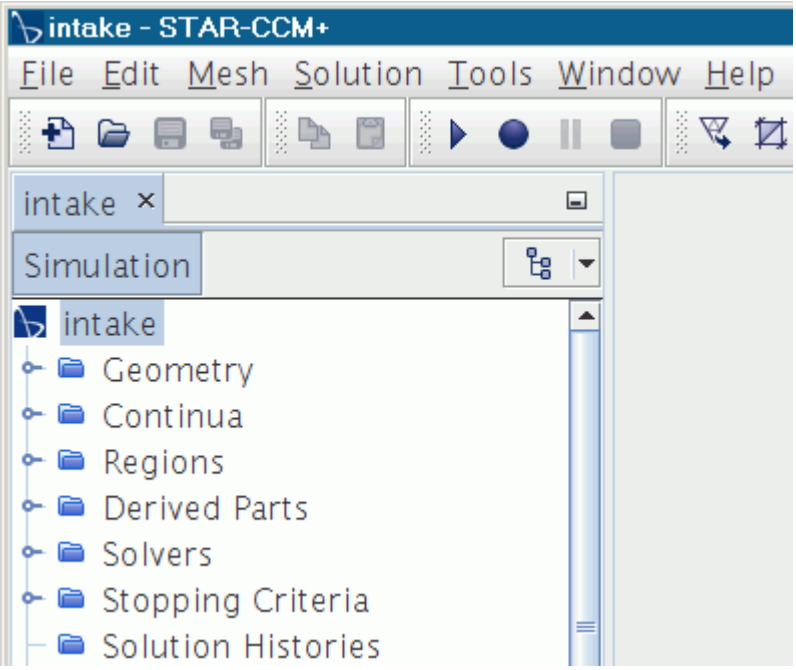
Changing the Workspace Font Size

In Linux or Windows you can adjust the size of the font that is used in the STAR-CCM+ client workspace, such as dialogs and the object tree, for that session.

A typical startup of the client activates the workspace in its standard font size.



Using the `-fontsize` option activates the workspace with the specified font size. The example in the following screenshot shows the workspace with a 16-point font size setting.



Linux

The line command for specifying the font size of the workspace is as follows:

```
% starccm+ -fontsize X
```

where `X` equals the point size of the fonts. For example, `-fontsize 16` would specify a 16-point font size.

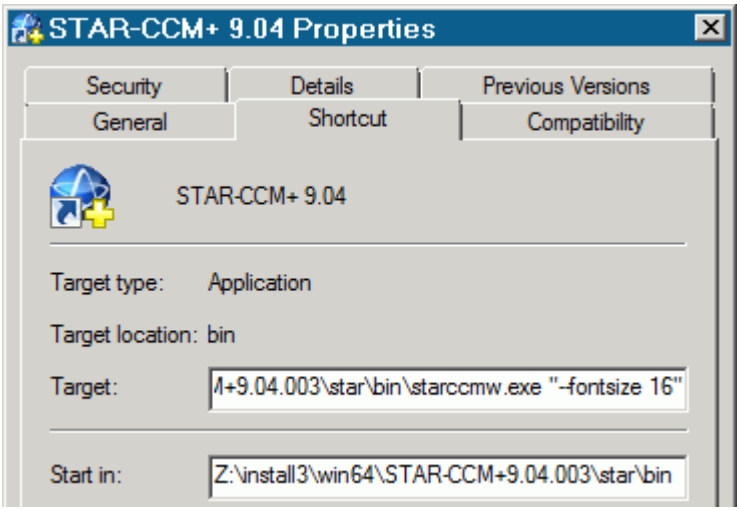
Windows

The line command for specifying the font size of the workspace is as follows:

```
% starccm+.exe "-fontsize X"
```

where `X` equals the point size of the fonts. For example, `"-fontsize 16"` would specify a 16-point font size.

If you typically work with Windows shortcuts, you can create a copy of your STAR-CCM+ shortcut and modify the contents of the *Target* text box in the shortcut properties with the `-fontsize` option:

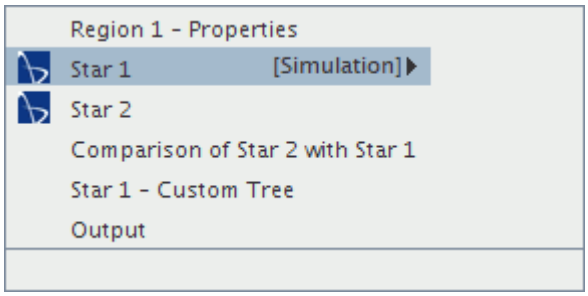


(This example may differ from the syntax for your shortcut, depending on your installation of Windows.)

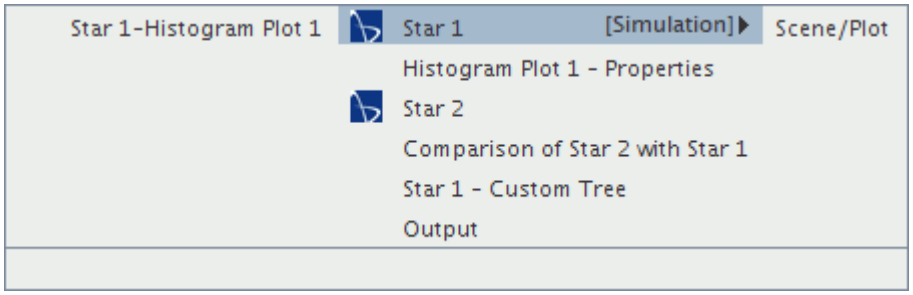
Navigating the Workspace with the Keyboard

You can use the keyboard to navigate among open top-level components of the STAR-CCM+ client workspace, including the simulation object tree, properties window, and scene and plot displays.

1. With the workspace active, press `<Ctrl>+<Tab>`, and continue to hold down the `<Ctrl>` key.
2. In the small pop-up window that appears, use the `<Tab>` and arrow keys to navigate among the components.



The contents of the window reflect the components of the workspace that are currently open. As you open other windows, the contents of the window change. The following example shows that a histogram plot was opened in the simulation `Star 1`.



3. To access the highlighted component, release the `<Ctrl>` key or press `<Enter>`.

Using the Menus

The menu bar is typical of GUI-based applications. It can be navigated with the mouse or with the keyboard.

The STAR-CCM+ menus provide a function-oriented view of some capabilities that are not bound to objects. They also duplicate a few of the double-click functions where it was deemed to improve user efficiency. The menus do not provide all of the available functionality. Rather, most functionality is accessed by right-clicking, double-clicking, and dragging objects around the workspace.

You can access them using the mouse or keyboard as in any other application.

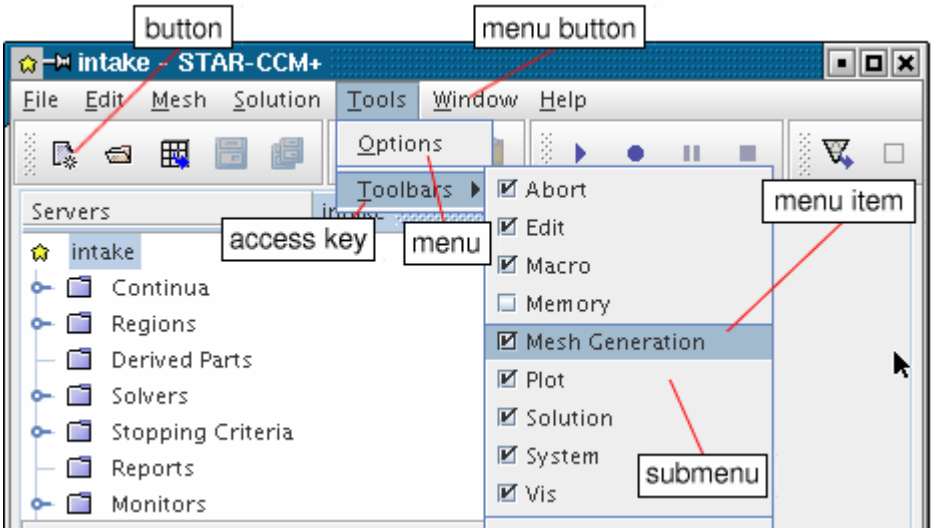
Contents:

Menu Reference

Menu Reference

This section provides a reference to all the menu items that appear in the supplied menu system.

The menu bar is always located immediately under the title bar and cannot be moved or hidden. The terms that are used with the menu bar are shown below. The following sections describe each of the menu items in the workspace and provide links to reference information where necessary.



If a menu item is grayed, it means that function is not available at the moment. Typically the function is not available because the workspace is not in a suitable mode, or the relevant object is not selected.

If a menu item has "..." appended to the name, it indicates that this function opens a dialog.

File

The File menu items apply to an entire server, simulation, or other file.

New Simulation...

Executes the new simulation action. This menu item is always activated.

Load Simulation...

Executes the load simulation action. This menu item is always activated.

Connect to Server...

Executes the connect to server action. This menu item is always activated.

Recent Files

Executes the recent files action. This menu item is always activated.

Save

Executes the save simulation action. This menu item is deactivated when there is no simulation open, when there are no changes to the simulation whose window is active, or when the active simulation is running. It becomes activated when you change the simulation, including opening a scene in the *Graphics* window.

Save As...

Executes the save as action. This option is deactivated when the simulation is running.

Save All

Executes the save all action. This menu item is activated when changes are made to at least one simulation (including the opening of a scene in the *Graphics* window).

Auto Save...

Executes the auto save action. This menu item is activated when a simulation window is active.

Macro

This menu item contains the following items in a submenu:

- Play Macro... -- Executes the play macro action. This menu item is activated when a simulation window is active.
- Start Recording... -- Executes the start recording action. This menu item is only activated when a simulation window is active.
- Pause Recording -- executes the pause recording action. This menu item is activated when a macro is recording.
- Stop Recording -- Executes the stop recording action. This menu item is activated when a macro is recording.
- Play Recent -- Executes the play recent macro action. This menu item is activated with submenu items once one macro has been run in any simulation.

Import

This menu item contains the following items in a submenu:

- Import Surface Mesh... -- Executes the import surface mesh action. This menu item is activated when a simulation window is active.
- Recent Surface Files -- Provides a list of the most recently imported surface files in a submenu. This menu item is activated with submenu items once one surface file has been imported into any simulation.

- Import Volume Mesh... -- Executes the import volume mesh action. This menu item is activated when a simulation window is active.
- Recent Volume Mesh Files -- Provides a list of the most recently imported volume mesh files in a submenu. This menu item is activated with submenu items once one volume mesh file has been imported into any simulation.
- Import CAE Model... -- Executes the import CAE model action. This menu item is activated when a simulation window is active.
- Recent CAE Files -- Provides a list of the most recently imported CAE files in a submenu. This menu item is activated with submenu items once one CAE file has been imported into any simulation.

Export...

Executes the export solution action, which opens a typical Save to File dialog. This menu item is activated when a simulation window is active.

Auto Export...

Executes the auto export action. This menu item is activated when a simulation window is active.

Page Setup...

Executes the page setup action. This menu item is always available.

Print...

Executes the print action. This menu option is deactivated if there is no display active in the *Graphics* window.

Disconnect

Executes the disconnect action. This menu item is activated when a simulation window is active.

Exit

Executes the exit application action. This menu item is always activated.

Edit

The following menu items let you edit objects using the clipboard of the operating system, and to navigate the selection history of the object tree.

Copy

Executes the copy action. This menu item is deactivated when there is no selection of an object that can be copied.

Paste

Executes the paste action. This menu item is deactivated when the clipboard is empty, or when there is no selection of an object into which the material on the clipboard can be pasted.

Delete

Executes the delete action. This menu item is deactivated when the selected object cannot be deleted, such as manager nodes of a simulation tree.

Previous Tree Selection

Executes the action to select the previous object in the range of object selections. This menu item is deactivated when there are no previous selections.

Next Tree Selection

Executes the action to select the next object in the range of object selections. This menu item is deactivated when there are no selections after the present one.

Mesh

The Mesh menu allows you to carry out a wide variety of tasks with elements of the mesh, making it unnecessary to adjust the original mesh file.

Generate Surface Mesh

Executes the generate surface mesh action. This menu item is activated when a simulation window is active, if the simulation contains a mesh that is generated in STAR-CCM+, and that no mesh generation process is running.

Generate Volume Mesh

Executes the generate volume mesh action. This menu item is activated when a simulation window is active, if the simulation contains a mesh that is generated in STAR-CCM+, and that no mesh generation process is running.

Clear Generated Meshes

Executes the clear generated meshes action. This menu item is activated when a simulation window is active, if the simulation contains a mesh that is generated in STAR-CCM+, and that no mesh generation process is running.

Scale Mesh...

Executes the scale mesh action. This menu item is always activated when a simulation window is active.

Diagnostics...

Executes the mesh diagnostics action. This menu item is always activated when a simulation window is active.

Convert to 2D...

Executes the convert mesh to 2D action. This menu item is always activated when a simulation window is active.

Solution

The Solution menu provides control over solution data and over running the simulation.

Clear Solution...

Executes the clear solution action. This menu item is always activated when a simulation window is active.

Initialize Solution

Executes the initialize solution action. This menu item is always activated when a simulation window is active.

Run

Executes the run action. This menu item is always activated when a simulation window is active.

Step

Executes the step action. This menu item is always activated when a simulation window is active.

Step Control

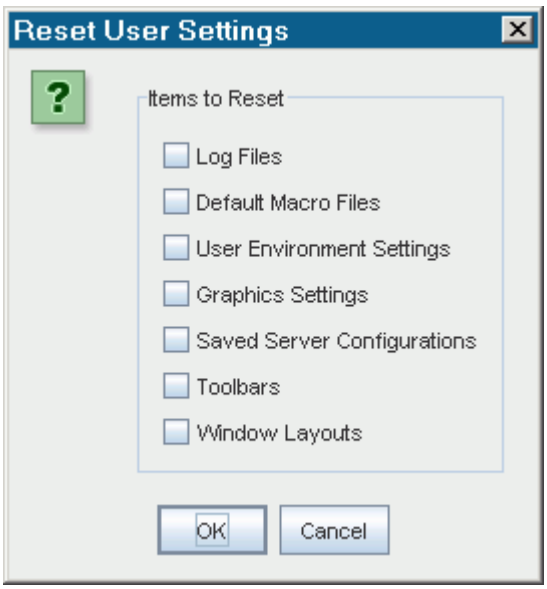
Executes the step control action. This menu item is always activated when a simulation window is active.

Tools

The Tools menu contains items for customizing your client workspace.

Reset User Settings

Opens a *Reset User Settings* dialog where you can select items to restore to their default settings.



Activate the checkboxes of the items to reset and click **OK**. You are prompted to close STAR-CCM+ to complete the reset.

Add-ons

Launches the Add-on Licensing dialog. This dialog manages the use of more licensing features.

Options

Opens a dialog with a host of advanced settings, which are covered under their respective topics. This menu item is always activated.

Plugins

Launches the Plugins dialog. This dialog manages the use of NetBeans modules in STAR-CCM+, which are plugins that permit the STAR-CCM+ GUI to be extended dynamically.

Window

The Window menu lets you control the appearance and position of windows.

Only three windows are accessible through the Window menu: *Properties*, *Servers*, and *Output*. These three windows always appear as the system defaults; others, such as a named simulation window or the *Graphics* window, appear as a result of user actions.

Close Window

Executes the close window action. This menu item is always activated unless all windows have been closed.

Maximize Window

Executes the maximize window action. This menu item is activated when a window in the *Explorer* pane has the focus and is hidden when one of the windows has been maximized.

Restore Window

Executes the restore window action. This menu item is activated when any maximized window has the focus and is hidden if no windows are maximized.

Properties

Executes the view Properties window action. This menu item is always activated.

Servers

Executes the view servers window action. This menu item is always activated.

Output

Executes the view output window action. This menu item is always activated.

Toolbars

Provides a submenu for controlling the display and customization of toolbars.

- Macro -- An option that displays (activated) or hides (deactivated) the macro toolbar. This menu item is always available.
- Solution -- An option that displays (activated) or hides (deactivated) the solution toolbar. This menu item is always available.
- Edit -- An option that displays (activated) or hides (deactivated) the edit toolbar. This menu item is always available.
- Memory -- An option that displays (activated) or hides (deactivated) the memory toolbar. This menu item is always available.
- System -- An option that displays (activated) or hides (deactivated) the system toolbar. This menu item is always available.
- View -- An option that displays (activated) or hides (deactivated) the view toolbar. This menu item is always available.
- Small Toolbar Icons -- An option that displays small (activated) or large (deactivated) versions of the images on the toolbar buttons. This menu item is always available.
- Standard -- A radio button menu item that is only visible when a second toolbar configuration has been saved.
- Customize -- Executes the customize toolbar action. This menu item is always available.
- Save Configuration -- Executes the save toolbar configuration action. This menu item is always available.

Help

The Help menu provides access to the online documentation, the Java API resources, and product information.

Help...

Executes the help action. For Help pages on specific topics, select the item and press <F1>. This menu item is always activated.

Java API...

Accesses STAR-CCM+ Java API resources through a web browser. This menu item is always activated.

Steve Portal

Accesses the customer success portal, https://thesteveportal.plm.automation.siemens.com, through a web browser. This menu item is always activated.

STAR-CCM+ IdeaStorm

Accesses the innovation forum, STAR-CCM+ IdeaStorm, through a web browser. This menu item is always activated.

About

Executes the about action. This menu item is always activated.

Using the Toolbars

Like the menu bars, the toolbars are typical of GUI-based applications, providing easy access to the most common actions.

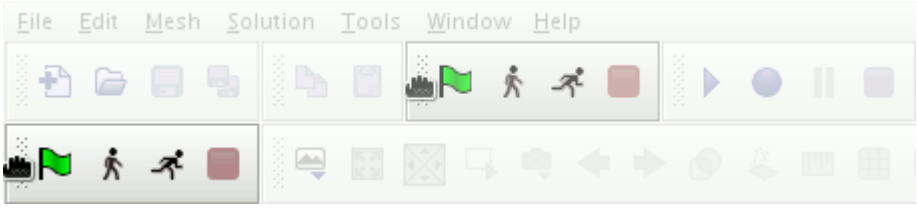
You can control the appearance of toolbars with various methods, including moving, showing and hiding, and creating them.

When you start a simulation, the full set of toolbars becomes visible. Otherwise the STAR-CCM+ workspace shows only the toolbars that are common to all simulations.

Moving a Toolbar

The toolbars always remain near the top of the workspace, just below the menu bar. They cannot be floated freely, or docked on the sides or bottom. However, they can be moved relative to each other.

To move a toolbar, position the mouse over the dimpled end (left side) of the bar and then drag it to a new position.

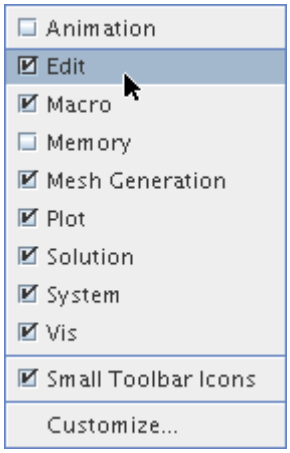


Dragging downwards creates a row for the toolbar, dragging left and right repositions it relative to the other bars.

Hiding a Toolbar

You can hide a toolbar using the menu or the right-mouse click:

- Select **Window** > **Toolbars** to display the submenu.



All the visible toolbars have the checkbox activated. Select the toolbar name to hide and click the left mouse button to deactivate the checkbox and hide that toolbar. The deactivated checkbox appears only if you bring up the submenu a second time.

- Right-click over any of the gray space of the toolbar (not directly over a button) to display a pop-up menu identical to the **Tools** > **Toolbars** submenu.

Displaying a Toolbar

You can display a toolbar using the menu or the right-mouse click:

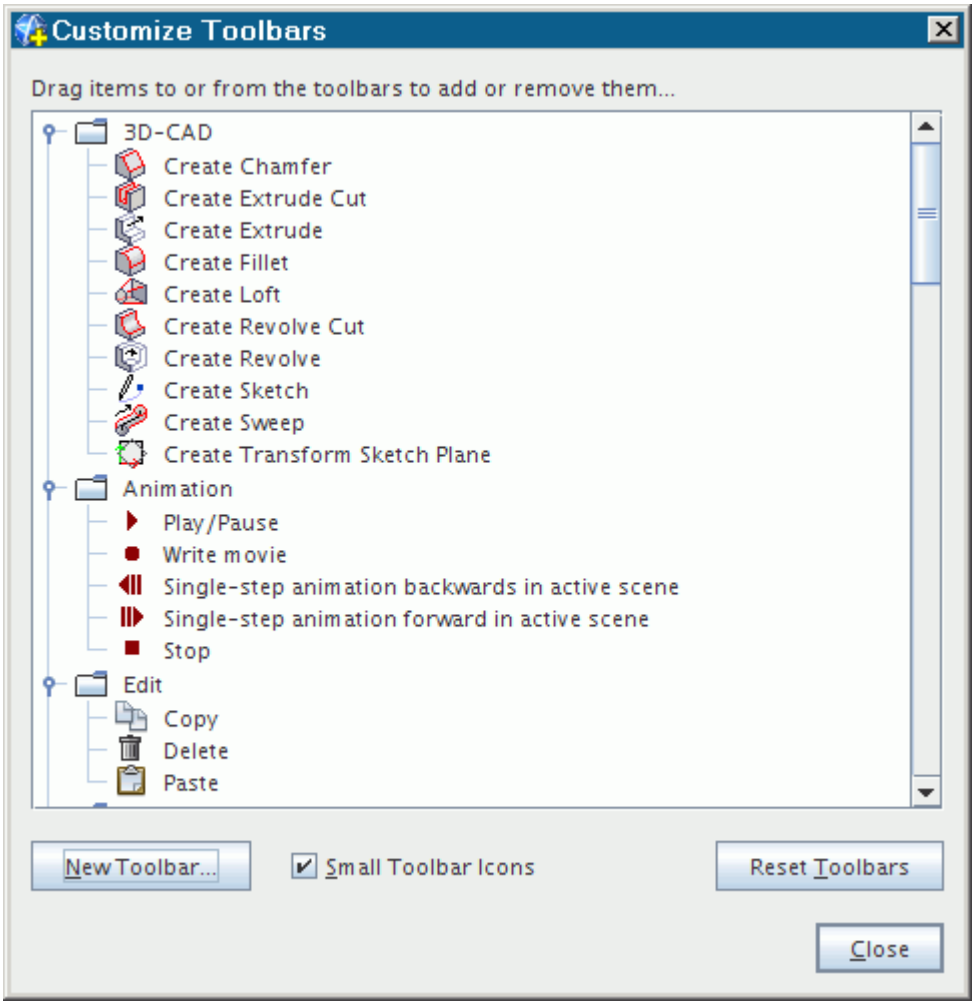
- Select **Window** > **Toolbars**.

  A hidden toolbar has the checkbox deactivated. Select the toolbar name to show and click the left mouse button to activate the checkbox and show that toolbar. The activated checkbox appears only if you bring up the submenu a second time.
- Right-click over any of the gray space of the toolbar (not directly over a button) to display a pop-up menu identical to the **Window** > **Toolbars** submenu.

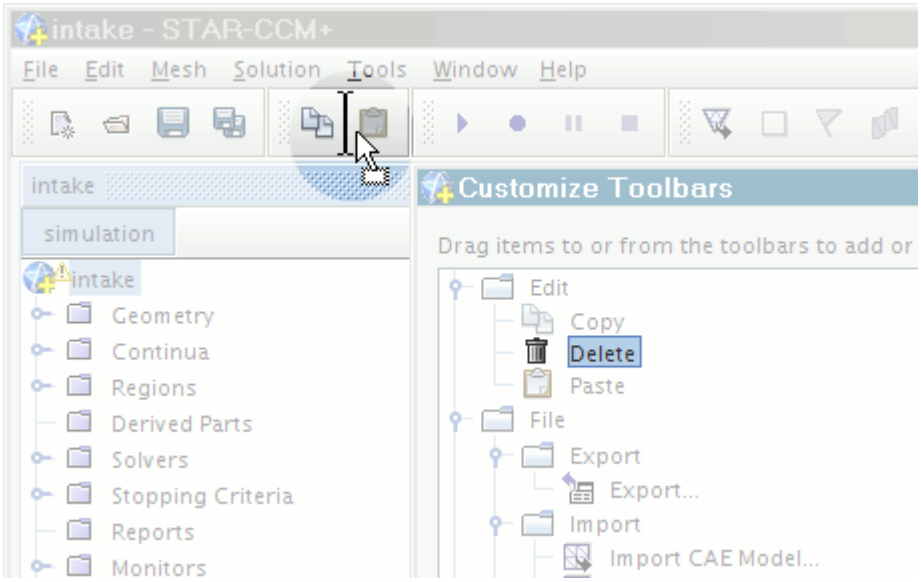
Customizing a Toolbar

You can add, delete, or relocate buttons to toolbars.

In the **Window** menu, select **Toolbars** > **Customize...**An explorer tree view of the current toolbar opens with definitions, as shown below.



This feature uses a drag-and-drop operation between the dialog and the toolbars themselves in the workspace. In the example below, the **Delete** button is being added to the Edit toolbar, into the position that is shown:



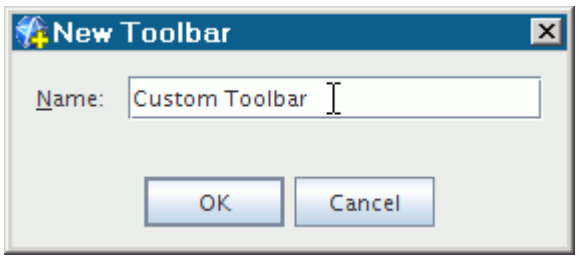
This method allows you to make any of the following changes to toolbars:

- Adding new buttons by dragging them from the dialog to the toolbar
- Deleting existing buttons by dragging them from the toolbar to the dialog
- Relocating existing buttons (by dragging them from the dialog to the new location) either within a toolbar or between toolbars

Creating a Toolbar

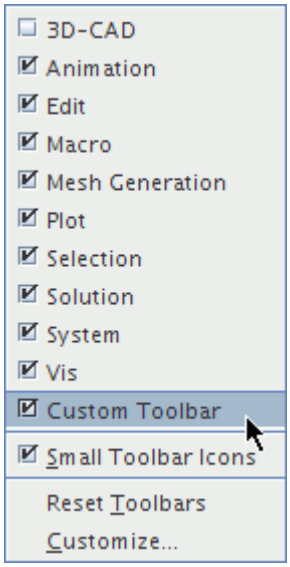
To add a toolbar to your workspace:

1. Click the **New Toolbar** button at the bottom of the *Customize Toolbars* dialog.
2. In the *New Toolbar* dialog, name the new toolbar.



3. After you click **OK**, customize your new toolbar before you close the *Customize Toolbars* dialog.

   Once a new toolbar has been saved, it appears along with other toolbars on the **Tools** > **Toolbars** menu.



It is also possible for a NetBeans plugin to create a toolbar when it is installed. To learn how to create your own modules, contact Siemens PLM Software support.

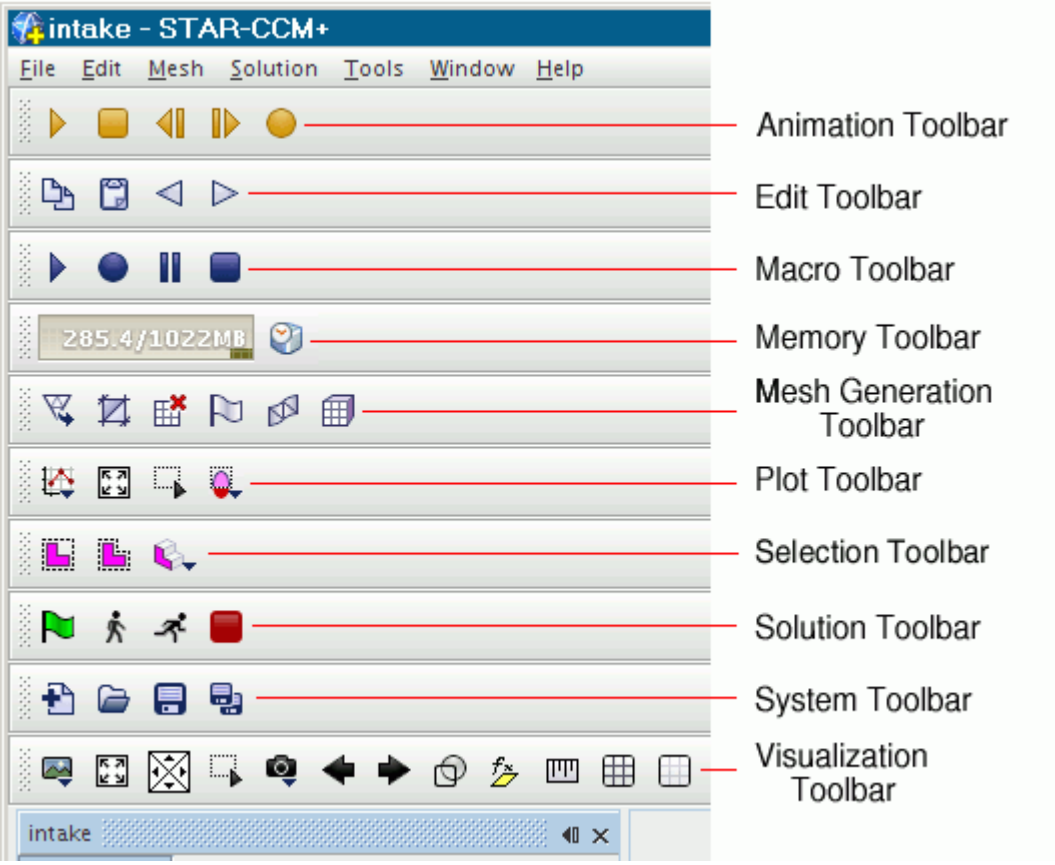
Contents:

Toolbar Reference

Toolbar Reference

Toolbar buttons are duplicates of menu item functions, and are provided for easy mouse access to common functions.

All the buttons are grouped onto several toolbars. In the figure below, the toolbars have been arranged so that they lie under each other.



Animation

The Animation toolbar buttons resemble the controls of a common video player.

(Play/Pause) -- this toggles between playing the scene animation and pausing it.

(Stop) -- Stops the animation in the scene.

(Single Step Forward) -- Moves the paused animation forward one frame.

(Single Step Backward) -- Moves the paused animation backward one frame.

(Write Movie) -- Records the animation onto a video file.

Edit

The Edit toolbar lets you copy and paste certain objects, as well as navigate the selection history of the object tree.

(Copy) -- Stores a copy of the selected object on the clipboard.

(Paste) -- Places the contents of the clipboard into the active area.

(Previous Selection) -- Moves backward in the range of object selections.

(Next Selection) -- Moves forward in the range of object selections.

Macro

The Macro toolbar lets you record and run macros.

(Play Macro) -- Launches an existing macro.

(Start Recording) -- Begins recording a new macro.

(Pause Recording) -- Interrupts the recording of the macro until you click the button again.

(Stop Recording) -- Finishes recording the macro.

Memory

285.4/1022MB (Garbage Collect) -- this recovers application memory that is no longer in use.

(Profile the Application) – when activated, this debugging tool samples data on the computing performance of the STAR-CCM+ client. Click the button once to begin the data gathering, and after a brief interval, click again to stop; STAR-CCM+ generates a message about where you can retrieve the NetBeans `.npss` data file.

Mesh Generation

The Mesh Generation toolbar provides quick access to important meshing operations.

(Import Surface) -- Imports surface data for generating a mesh.

(Launch Surface Repair) -- Opens the *Surface Preparation* dialog for regions or parts.

(Clear Generated Meshes) -- Deletes meshes generated in STAR-CCM+.

(Initialize Meshing) -- Provides the starting surface for the meshing models.

(Generate Surface Mesh) -- Begins the generation of a surface mesh.

(Generate Volume Mesh) -- Begins the generation of a volume mesh.

Plot

The Plot toolbar lets you open plot displays and navigate within them.

(Create/Open Plots)	Opens a drop-down menu that allows you to add a plot or open an existing one.	
	XY Plot	Creates a XY plot and opens a display for it.
	Monitor Plot	Creates a monitor plot and opens a display for it.
	Histogram Plot	Creates a histogram plot and opens a display for it.
	Open	Contains a submenu with names of existing plots.
	Open All Plots	Displays all existing plots in the *Graphics* window.

(Fit) -- Returns the data portion of the plot display to the default zoom and pan position.

(Zoom Selected Plot) -- A toggle button that, when selected, zooms the data portion of the plot approximately one order of magnitude each time the left mouse button is clicked. The click point is centered on the axes, so you could use this point to determine more accurately the x and y values of a particular point. A drag operation begins a rubber-band zoom that is effected when the button is released. If the **Pan** button was selected, then it is deselected as you press this button.

(Create Data Focus Selection)	Opens a drop-down menu that allows you to add a data focus selection or intersect with an existing one. See Exploring Solutions with Data Focus.	
	Add Selection to Data Focus	Lets you draw a box in the plot for defining data in the data focus.
	Intersect Selection with Data Focus	Lets you draw a box in the plot that intersects with an existing data focus box. This intersection creates a compound data focus that uses the AND/OR rule for filtering data.

Selection

The Selection toolbar lets you refine your selection of objects in the scene display.

(Rubberband Select) -- Selects a specific area in the display by clicking two points that define the opposing corners of the area. Press `<Esc>` to cancel.

(Zone Select) -- Lets you draw a zone by clicking and dragging to create a series of connected points. Objects within that zone are selected. Press <Esc> to cancel.

(Depth Select) -- Allows you to select objects that are obscured from view in the scene.

(Selection Filter) -- Provides options for selecting volumes, surfaces, or curves with either the rubberband or zone tool.

Solution

The Solution toolbar gives you quick access to basic solution controls.

(Initialize Solution) -- Initializes the solution.

(Step) -- Adds a user-specified number of iterations to a solution.

(Run) -- Begins running the simulation.

(Stop) -- Stops the running of the simulation.

System

The System toolbar provides controls for starting a simulation and saving it to file.

(New Simulation) -- Creates a simulation.

(Load Simulation) -- Opens an existing simulation.

(Save) -- Saves the simulation.

(Save All) -- Saves all open simulations at once.

Visualization

The Visualization toolbar lets you open scene displays, work with them interactively, and store view settings.

(Create/Open Scenes)	Opens a drop-down menu that allows you to add a scene or open an existing one.	
	Geometry	Creates a geometry scene and opens a display for it.
	Mesh	Creates a mesh scene and opens a display for it.
	Scalar	Creates a scalar scene and opens a display for it.
	Vector	Creates a vector scene and opens a display for it.
	Empty	Creates an empty scene and opens a display for it.

	Open	Contains a submenu with names of existing scenes.
	Open All Scenes	Displays all existing scenes in the *Graphics* window.

(Reset View) -- Restores the view to a zoom factor such that the model takes up most of the display. The view angle is unchanged. Use this function if the model has somehow zoomed or panned to such a position that you can no longer see it on the display.

(Rotate) -- Rotates the view around a focal point that your most recent mouse click in the display sets.

(Rubberband Zoom) -- Allows you to zoom to a specific area in the display by clicking two points that define the opposing corners of the zoomed area. Press `<Esc>` to cancel.

(View)	Opens a drop-down menu that allows you to set, save or restore a view.	
	Store Current View	Records properties about the projection mode, zoom factor, pan position and view angle and adds a node to the **Views** node of the simulation tree. It is named View N, where N is the first available number that is not used by existing views with the same prefix. So if you have a view, View 1, then this operation creates a view, View 2.
	Restore View	Contains a submenu of saved views, including a thumbnail of the view. Select one of the submenu items to restore that view.
	Projection Mode	Contains a submenu that toggles between perspective and parallel mode for the current view.
	Look Down	Contains a submenu of standard view angles using the axes as reference vectors.
	Standard Views	Contains a submenu of standard views using the aspects of the object in the scene.
	View Coordinate System	Contains a submenu of coordinate systems in the simulation that a view can use.

(Previous View) -- Reverses the latest change to the view.

(Next View) -- Repeats a change that was subjected to the `Previous View` command.

(Make Scene Transparent) -- Makes all parts transparent so that you can see all of them from any angle, and overrides the opacity settings of all the displayers in the scene. Restores the displayer settings when you click the button again.

(Create Planar Section) -- Allows you to create a plane section derived part, which is normal to the view, by clicking two points in the display.

(Measure Distance) -- Lets you click two points in the display to measure the distance. After your second click, the information on point coordinates and distance appears in the *Output* window. If you want to cancel this process after clicking the button, simply press `<Esc>`.

(Show All Meshes) -- Adds mesh lines to the display. Clicking this button once adds the mesh lines, and overrides the settings of all the displayers in the scene. Clicking it a second time reverts to the displayer settings.

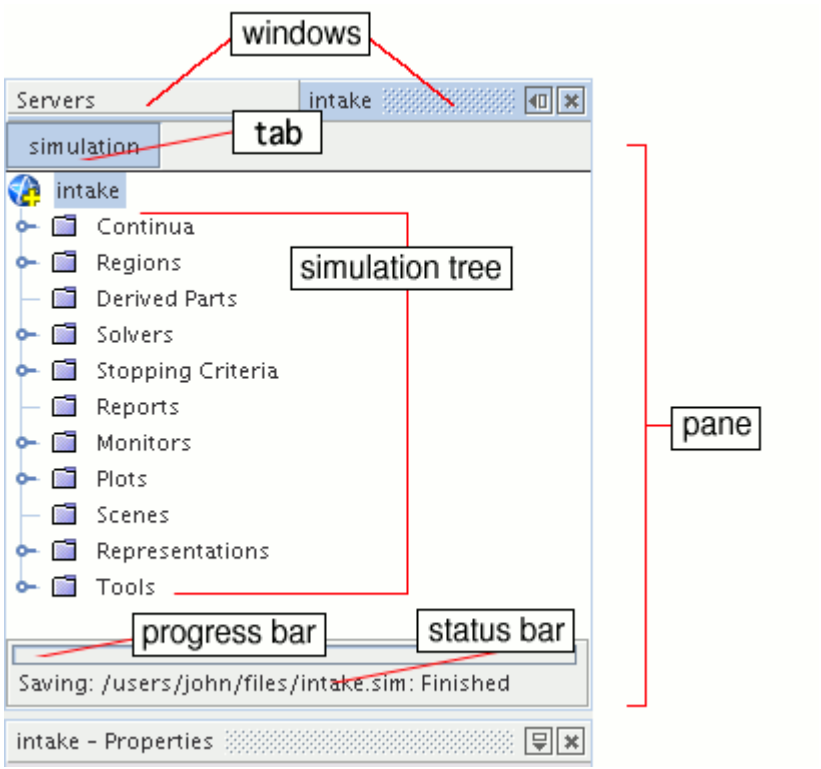
(Hide All Meshes) -- Removes mesh lines from the display. Clicking this button once removes the mesh lines, and overrides the settings of all the displayers in the scene. Clicking it a second time reverts to the displayer settings.

Using the Explorer Pane

The *Explorer* pane is typically docked on the left of the workspace.

This pane can hold either of the following kinds of window:

- Zero or one Servers window (this window indicates servers on the network and is closed by default)
- Zero or more named simulation windows (each running a simulation; *intake* in the example below)



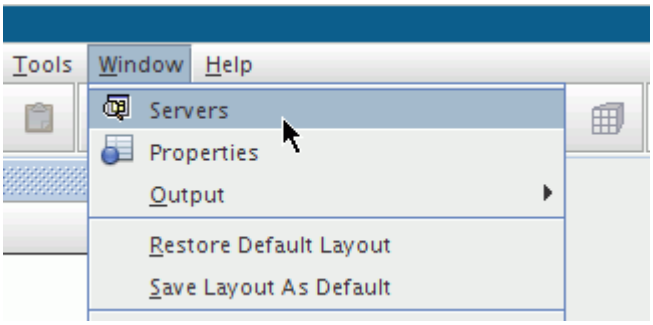
Each window type holds a tree view and, sometimes, a status bar of some kind. There are general rules for navigating the trees (such as opening and closing nodes) that apply to all the trees, and also general rules for using the windows.

The tabs near the top of the simulation window allow you to toggle among views of simulation objects, scene or plot objects, or editing panels when applicable.

Servers Window

The nodes in this object tree represent STAR-CCM+ server processes running on your local network.

The *Servers* window is closed by default. To open it, select **Window** > **Servers** from the top menus.

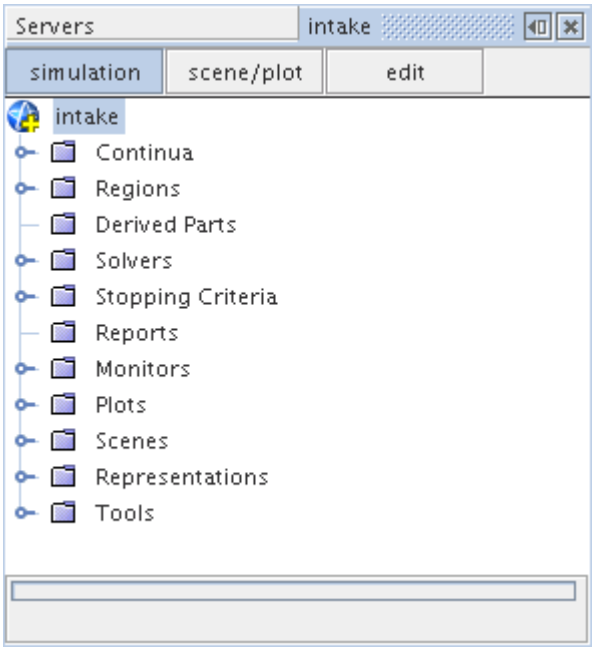


By default, only processes that you start are displayed. STAR-CCM+ server processes that others start can be displayed by entering different user names in the *User* property of the **Servers** node. This feature can be useful if your user name on a remote machine differs from your local user name.

Simulation Windows

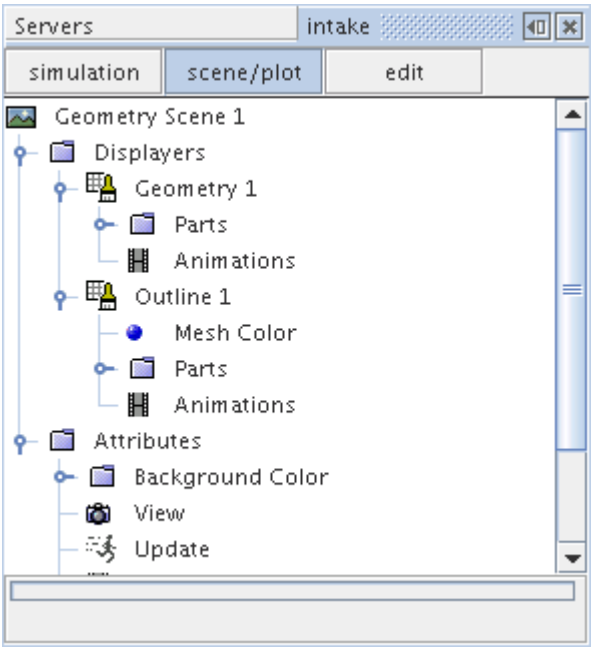
This window shows the named simulation tree, which contains nodes that represent the various objects of the open simulation.

Multiple windows of this kind can be present at the same time--one for each open simulation.

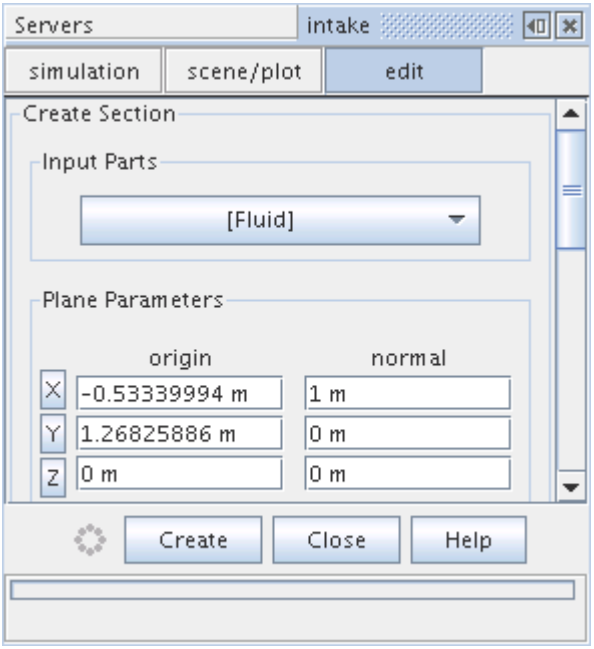


Near the top of a simulation window are one or more tabs that allow you to use this window for specific tasks, where applicable:

- Viewing the objects of a scene or plot, when its display is open and active



- Editing an object, particularly creating a derived part



The *simulation* tab is always present in a simulation window.

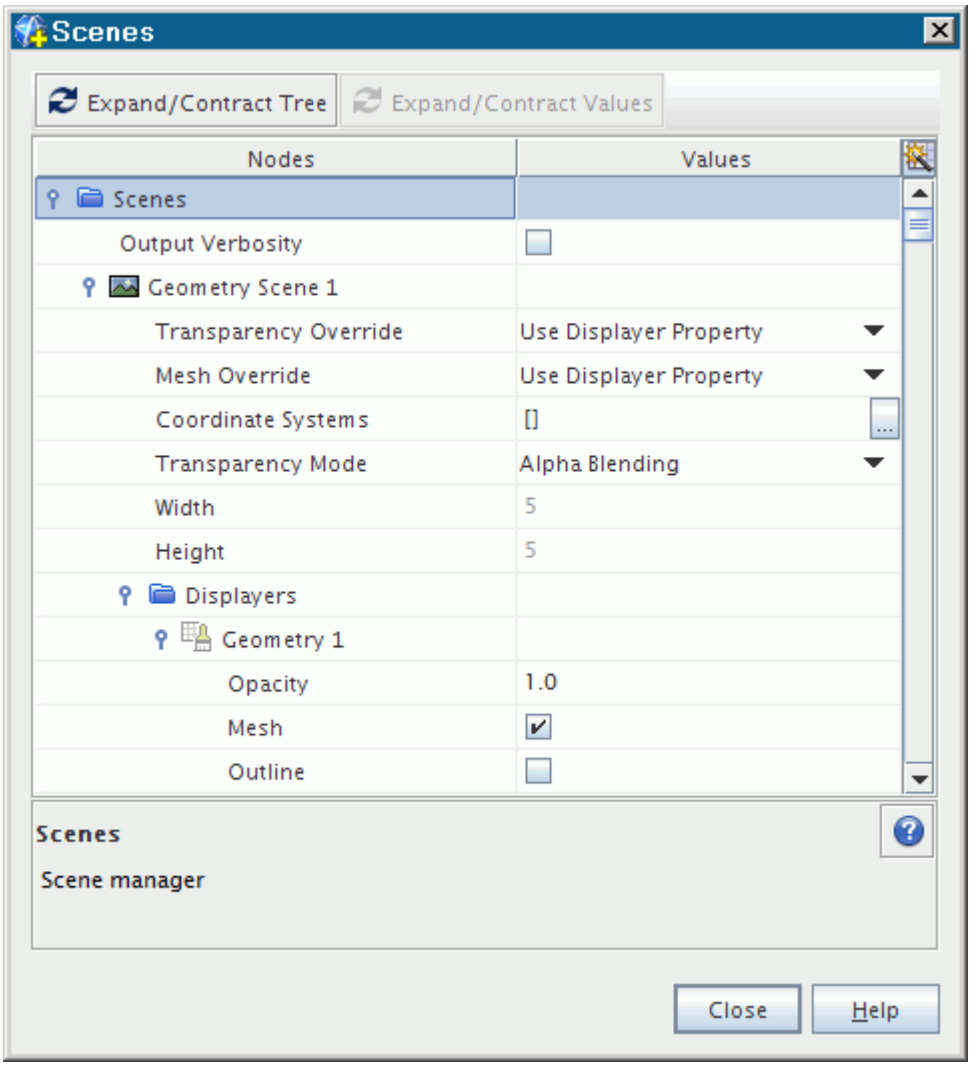
Closing the Explorer Pane

To close the *Explorer* pane with a single action, right-click any of the windows and select **Close Group**.

Using the Table Editor

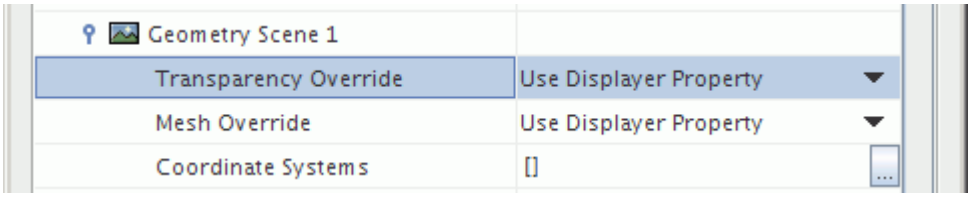
The table editor combines the functionality of the simulation tree and *Properties* window.

It displays the selected node, all of its child nodes, and their properties in a single dialog.



The left-hand column includes nodes and their properties underneath. The right-hand column contains the settings of those properties.

When a row in the table has the focus, its cells have a colored background. The cell in the row that has the focus has a colored background and border.



Note: There is no function available to right-click on a node within the edit dialog. To access pop-up menu options for a node, exit the edit dialog and right-click the appropriate node in the simulation tree.

Combining the Object Tree and Properties Window

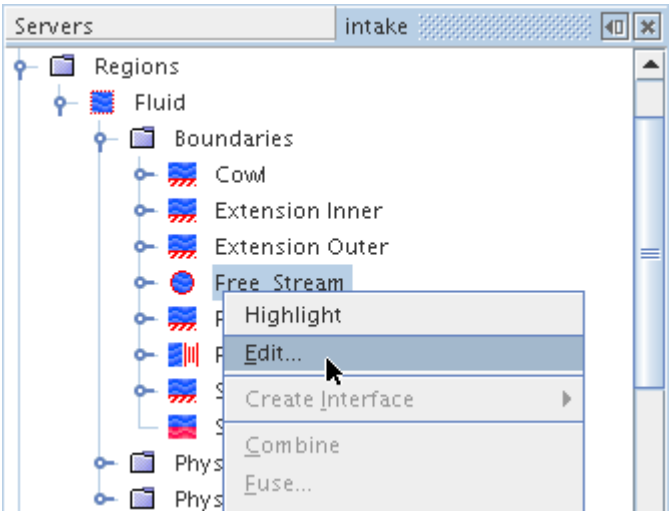
The table editor combines much of the functionality of the object tree and *Properties* window in a single dialog.

- The description area near the bottom left of the table editor summarizes the currently selected object or property, just like the properties description area does.
- You can access the online Help for objects within the table editor just as you would from the *Properties* window. When a node has the focus, pressing the question mark button in the lower right corner of the table has the same effect as the `<F1>` key.
- As with the object tree, *Properties* window and toolbars, hovering with the mouse activates a brief description of the item of interest.

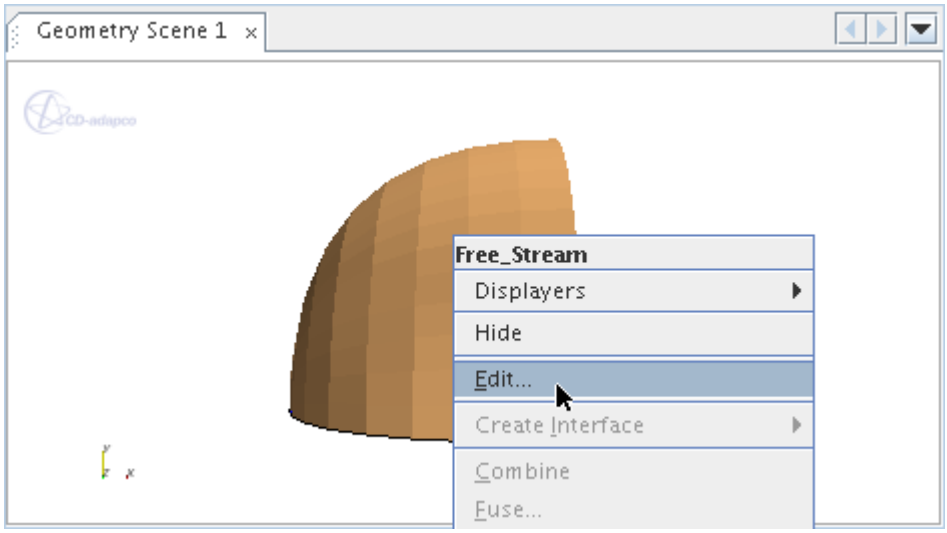
Activating the Table Editor

The table editor can be activated from several places in the workspace.

- Right-clicking on a node in the simulation tree and selecting **Edit...**



- Right-clicking on an object in the *Graphics* window while in interact mode, and selecting one of the edit options from the pop-up menu

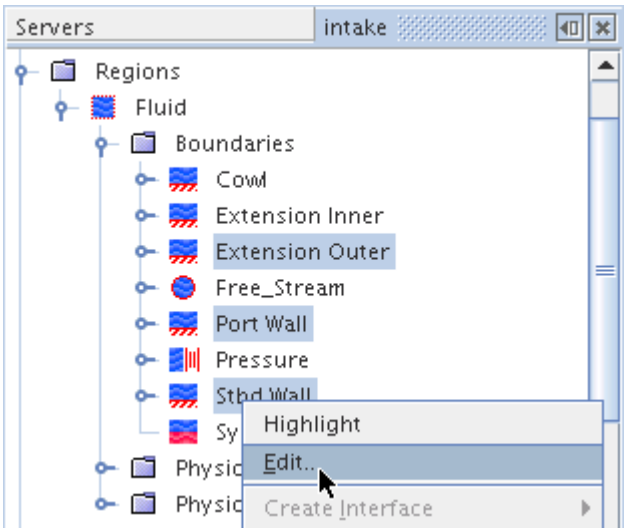


These techniques can also be used to activate the multi-edit mode.

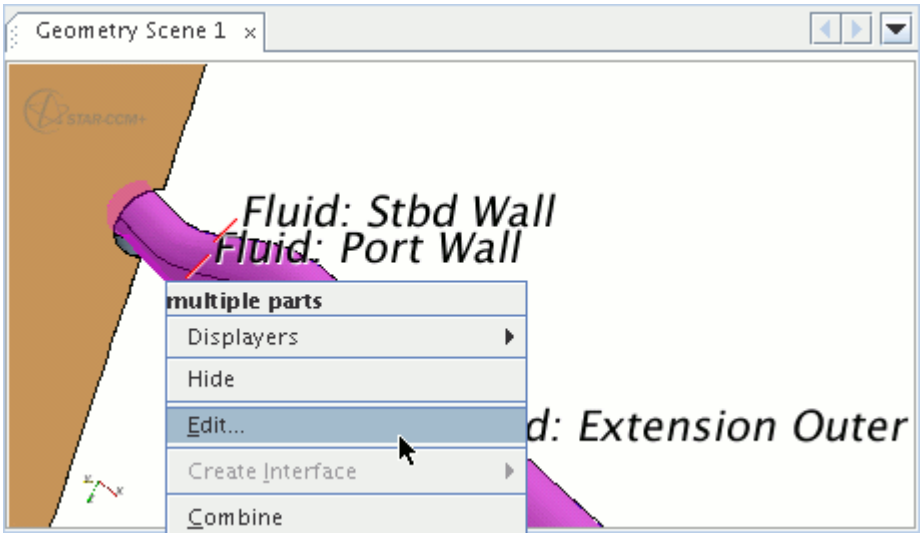
Activating the Table Editor in Multi-Edit Mode

The multi-edit mode of the table editor is a useful way to work with multiple objects at once, provided all of those objects are of the same general type, such as boundaries.

To activate this mode, select all the objects you wish to view in the table editor using the `<Shift><Click>` method, or `<Ctrl><Click>` for non-contiguous items. Right-click on these nodes and select **Edit...**.



You can also use the `<Ctrl><Click>` selection technique in the *Graphics* window while in interact mode.

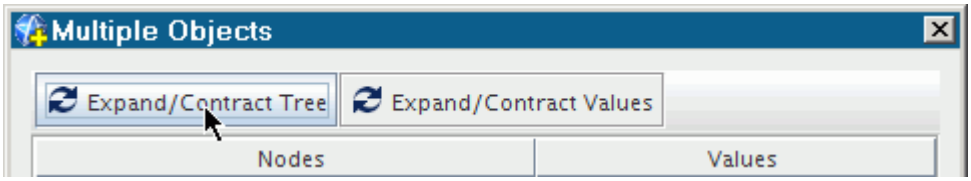


Note: When you attempt to multi-edit objects with properties that differ extensively, the **Edit...** context menu item is unavailable (grayed out). This restriction includes the editing of monitors, plots, and scenes. To multi-edit monitors, plots, or scenes, multi-select only the ones of the same type.

Working with Multiple Objects at a Glance

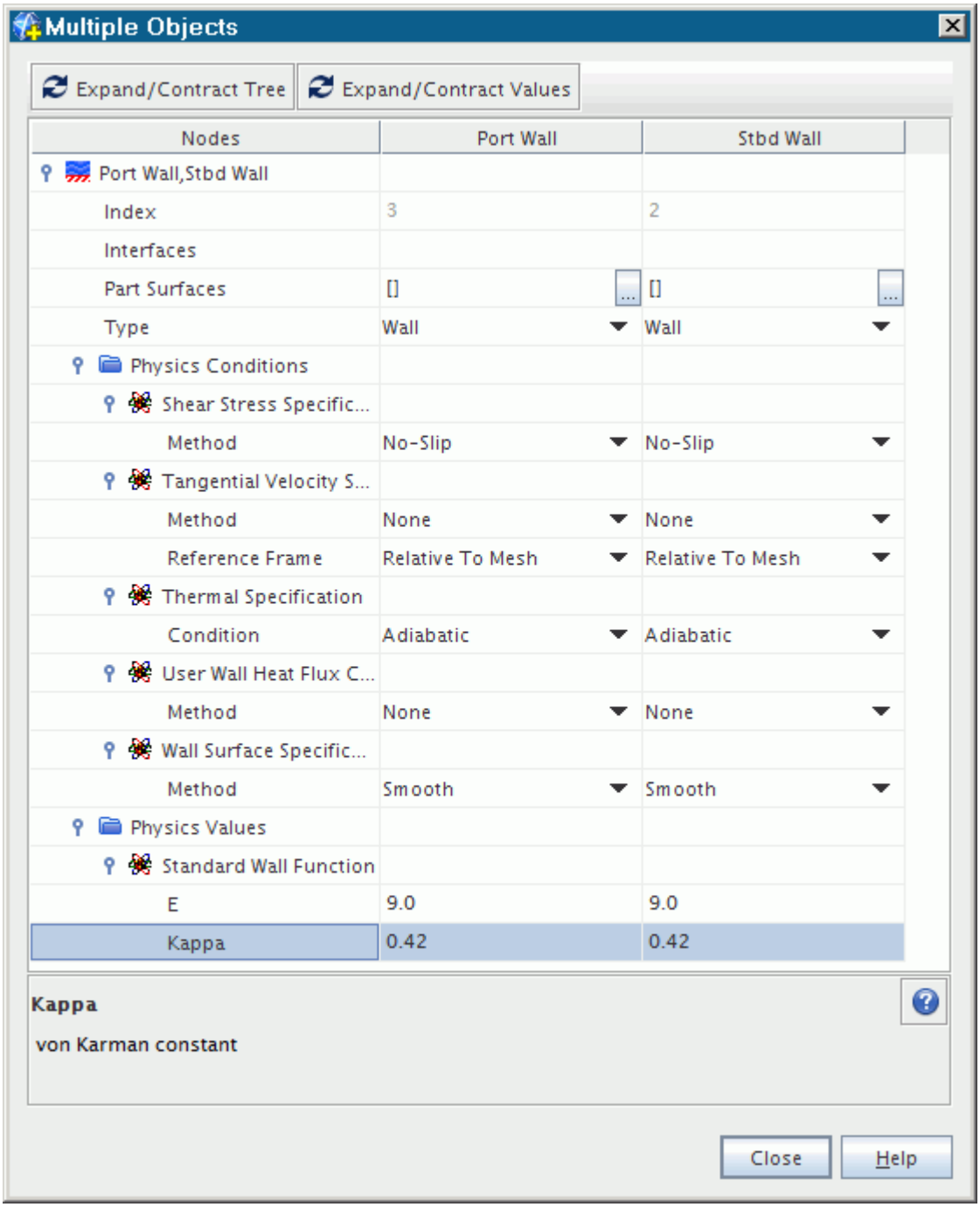
The table editor lets you apply settings to multiple separate objects easily since you can view them all at once, provided their parent object has the focus when you activate this feature.

You can expand the node hierarchy in the table editor by clicking the **Expand/Contract Tree** toolbar button in the top of the dialog. In addition, if you want to enter different values for a property that is common to multiple nodes, click the **Expand/Contract Values** toolbar button.



This will give you access to the properties of the subnodes. These buttons toggle the rows and columns open and closed.

For example, if you wish to apply the same setting to multiple boundaries, you can do so by referring to a previous entry elsewhere in the table.

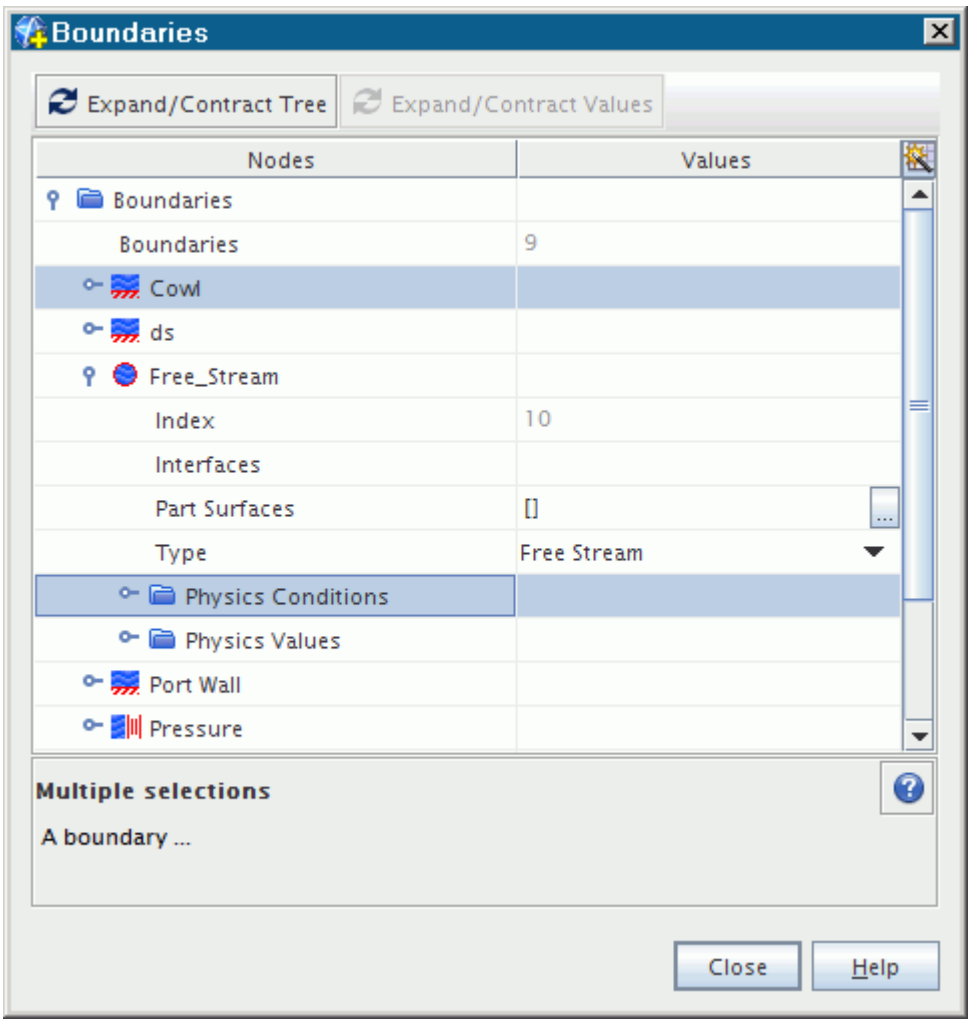


Note: There is no function available to right-click on a node within the Multi-edit dialog. To access pop-up menu options for a node, exit the Multi-edit dialog and right-click the appropriate node in the simulation tree.

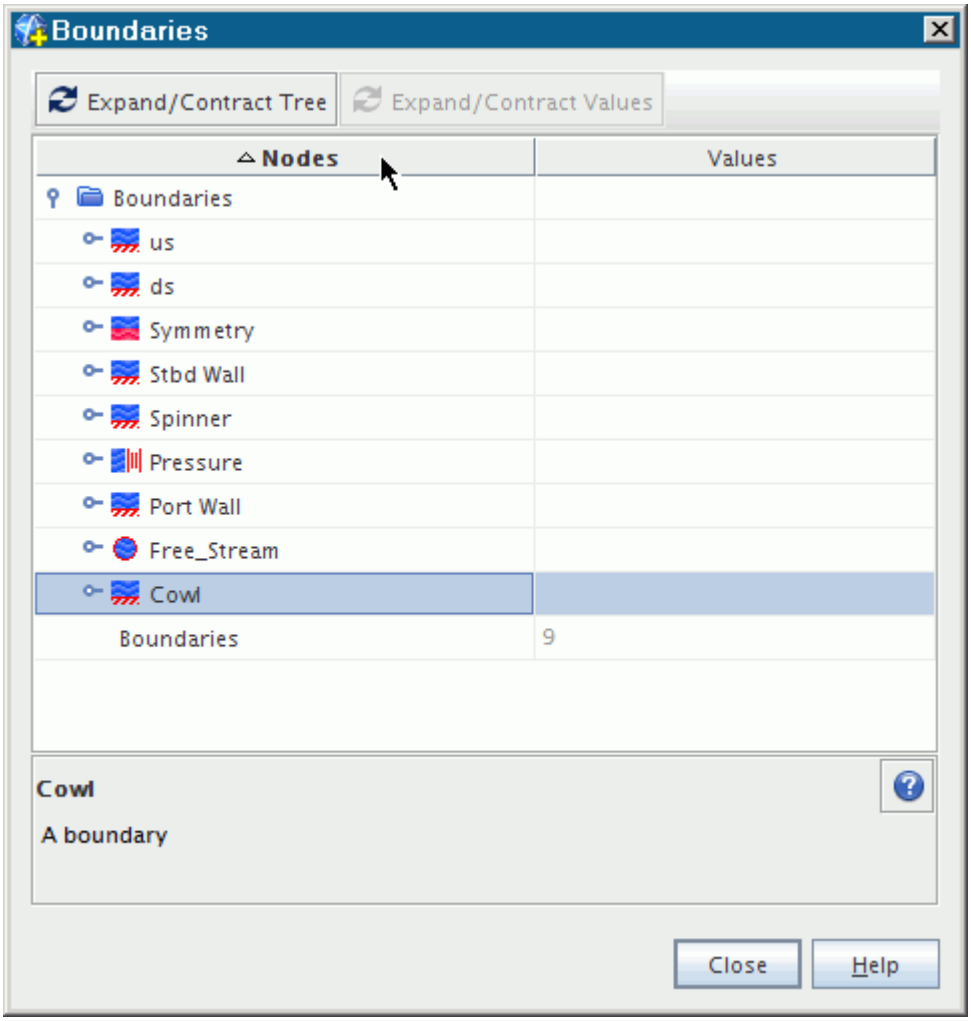
Adjusting the Table Editor

You can adjust the table editor for easier access to the objects of interest.

The handles on the nodes work just like the ones in the object tree, allowing you to show or hide subnodes or properties.



The column headers, **Nodes** and **Values**, are buttons that let you sort the visible objects in the respective column by name, toggling between ascending and descending order.



Navigating with the Keyboard

If you prefer using the keyboard, you can use keystrokes to access various parts of the table editor.

For navigating among the cells in the body of the table, the following keys are supported:

- `<Right>` - shifts the focus from one column to the one right of it. When the focus is in the tree column, pressing this arrow opens the node of the row (assuming it is closed).
- `<Ctrl>+<Right>` - When the focus is in the tree column, pressing this combination opens all nodes and subnodes within the selected node of the row.
- `<Left>` - shifts the focus from one column to the one left of it. When the focus is in the tree column, pressing this arrow closes the node of the row (assuming it is open).
- `<Ctrl>+<Left>` - When the focus is in the tree column, pressing this combination closes all nodes and subnodes within the selected node of the row.
- `<Tab>` - works like the `<Right>` arrow.
- `<Shift><Tab>` - works like the `<Left>` arrow.
- `<Up>` - shifts the focus from one row to the one above it.
- `<Down>` - shifts the focus from one row to the one below it.

- <Home> - shifts the focus to the left cell of the row.
- <End> - shifts the focus to the right cell of the row.
- <Space bar> - goes to edit mode in a property cell.
- <Esc> - goes to view mode in a property cell, assuming that cell is in edit mode; it does not change the value if you have changed it.

  This key closes the dialog if pressed when there is no property cell in edit mode.

To access a drop-down list, navigate with the arrows until the cell of that list has the focus, then press the <Space> bar to display the list. Use the <Up> or <Down> arrow keys to reach the desired item in the list, and then press <Enter>. To exit the list without entering any change, press <Esc>.

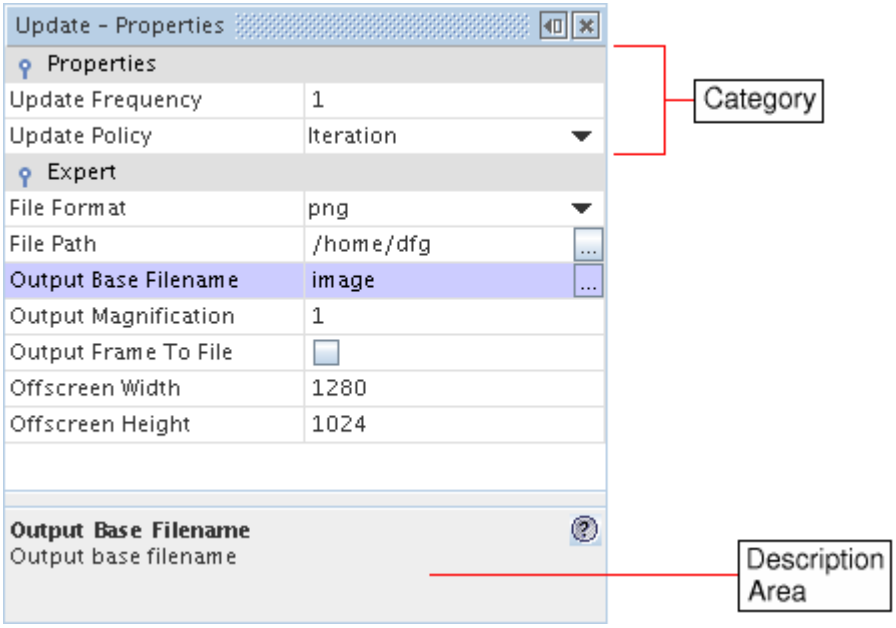
To move to the top row of the table body, press <Ctrl>+<Home>. To move to the bottom row, press <Ctrl>+<End>.

To select a contiguous block of nodes, select the first node of the block using the <Up> or <Down> arrow keys, hold down the <Shift> key, then press the <Down> arrow until you reach the last node of the block.

To select a non-contiguous block of nodes, select the first node of the block using the <Up> or <Down> arrow keys, hold down the <Ctrl> key, then press the <Down> arrow until you reach the next node of the block and press the <Space> bar. Continue navigating up or down and pressing <Space> on required nodes until you are done.

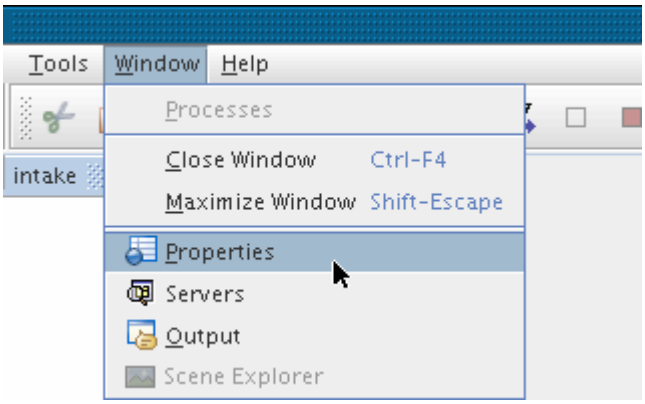
Using the Properties Window

The *Properties* window is a single location where all of the properties for a selected object can be viewed and manipulated.



It is designed for easy navigation with the mouse or the keyboard, and includes a description area at the bottom of the window.

To open this window, select **Window** > **Properties**.



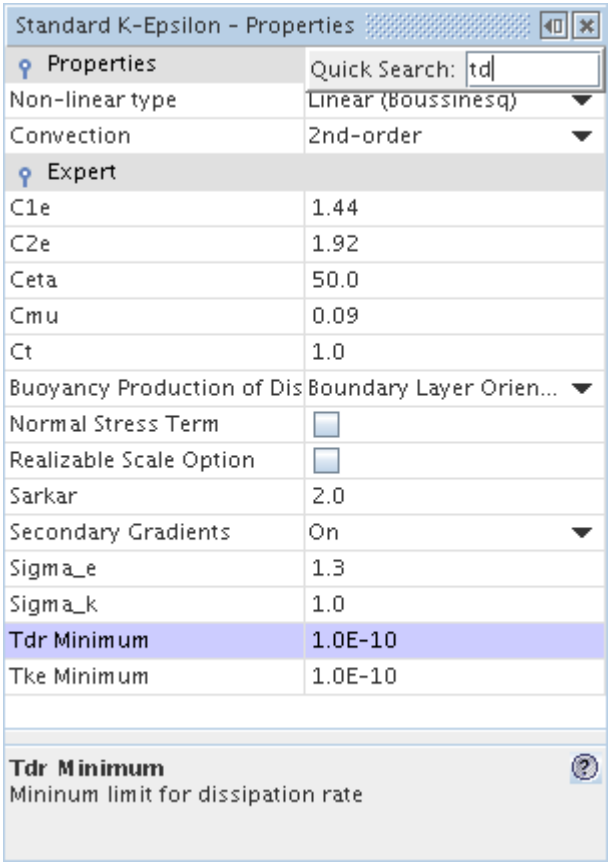
The dialog that appears under the *edit* tab of the *Explorer* pane, typically used for derived parts and surface repair, causes the *Properties* window to disappear from the workspace. It reappears when the *edit* tab is closed.

A set of editing techniques is common to virtually all STAR-CCM+ objects.

A special kind of property, usually labeled *Parts*, activates the object selection dialog. See Activating the Object Selection Dialog.

Using the Search Field

When the *Properties* window has the focus, typing a keystroke causes a search field to appear.



If you type one or more letters that are in the name of one of the properties, that property gets the focus.

Common Editing Techniques

Put the property into edit mode using one of the following techniques:

- Clicking the property value (in the right-hand column)
- Using the keyboard to navigate to the property and press the `<Space>` bar
- Clicking the ellipsis button in the right-hand column (if one is there) to activate the property customizer

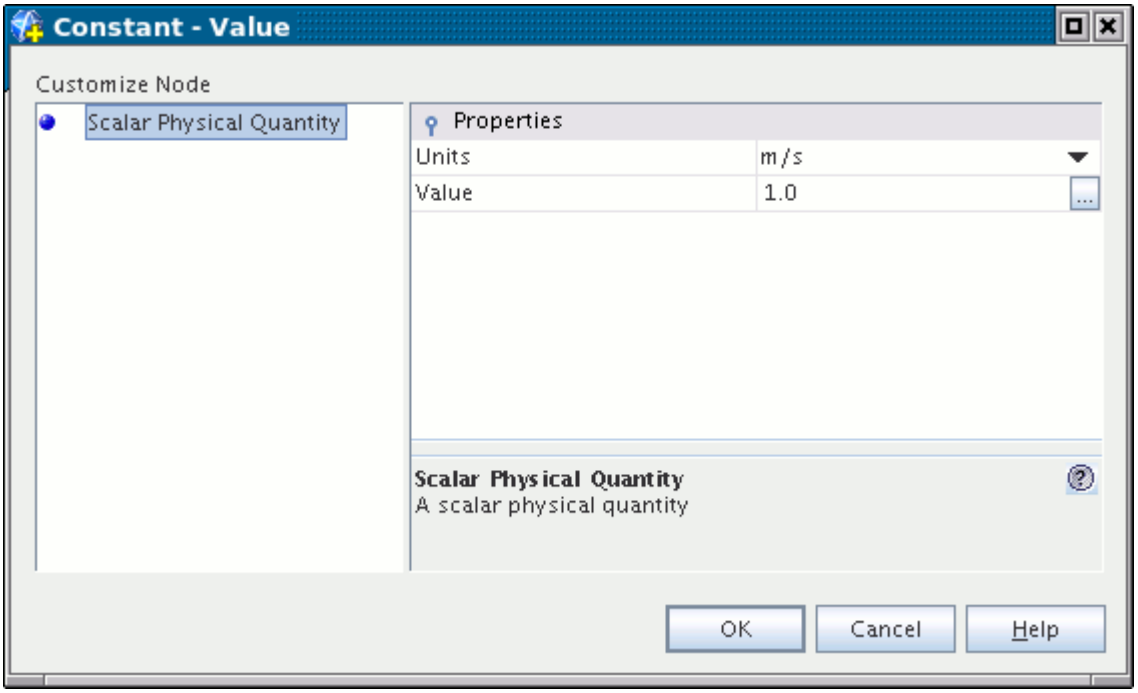
Change the value by typing, making a selection from a drop-down list or making the changes within the property customizer (if one is activated).

Accept the new value by pressing `<Enter>` in the inline editor or clicking **OK** on the *Customizer* dialog. The property is then put into view mode and retains the focus. If the value that you entered was invalid for some reason, a warning box is displayed and the value reverts to its previous value.

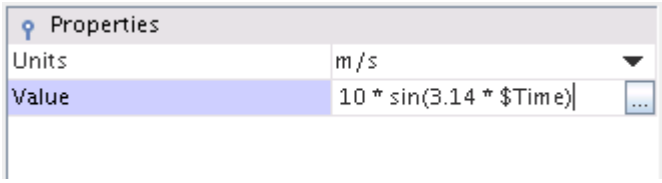
Cancel your editing by pressing `<Esc>` or clicking **Cancel** on the *Customizer* dialog.

Using a Property Customizer

Use the property customizer for greater control over a property's value, typically with control over units. The most common method for activating this dialog is clicking the ... (property customizer) in an object's value property.



For some objects, you can also enter an expression directly into the value field.

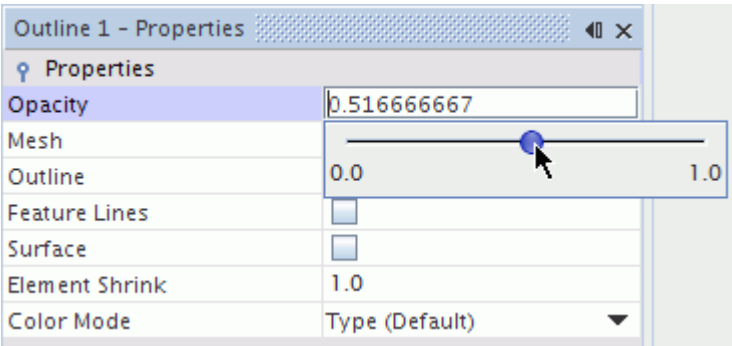


Alternatively, if you click the ... (property customizer) in that field, it activates a larger text box (helpful for longer entries).

Using a Slider

For any property with a finite range of values, a slider is available for adjustments. To use this feature:

1. Click the property value. This action makes the slider available.
2. Drag the slider button along the range until you reach the value you want.



If the relevant property is displayed in the current scene, the scene automatically updates to the new value.

When the property specifies a range, the slider provides two buttons. While you can slide either button past the other, the button at left always specifies the minimum, while the one at right always specifies the maximum.

3. To close the tool, click anywhere else in the STAR-CCM+ window.

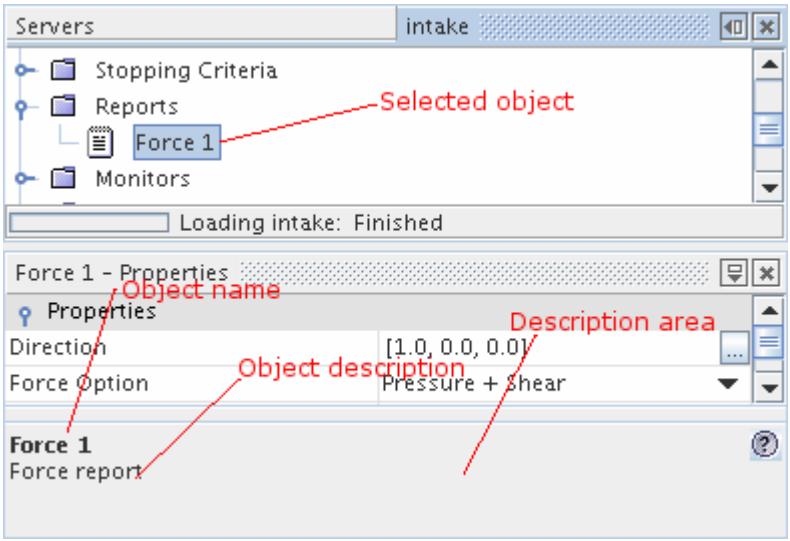
   The new value is automatically applied.

   Alternatively, you can enter a value by typing it.

Showing and Hiding the Description Area

At the bottom of the *Properties* window is a description area that summarizes the currently selected object or property by displaying the name of that object or property along with a brief description.

The example below shows the description area when an object on the simulation tree has the focus.

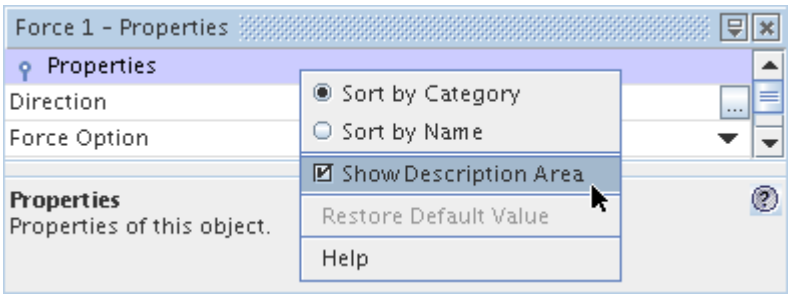


To hide the description area:

1. Right-click anywhere within the *Properties* window, except on the title bar or within a property value that is in edit mode.

2. In the pop-up menu that appears, select **Show Description Area** so that the checkbox of that menu item is cleared.

To restore the description area, repeat the step so that the checkbox is ticked.



Contents:

Properties Window Reference

See Also:

The Surface Repair Tool

The Object Selection Dialog

Accessing the Properties Window with the Keyboard

Conditions, Values, and Profiles

Setting Values Using an Expression

Properties Window Reference

This section describes the structure of the properties window.

The *Properties* window contains one or more collapsible lists, or categories, each of which has a category bar at the top. This window also has its own pop-up menu.

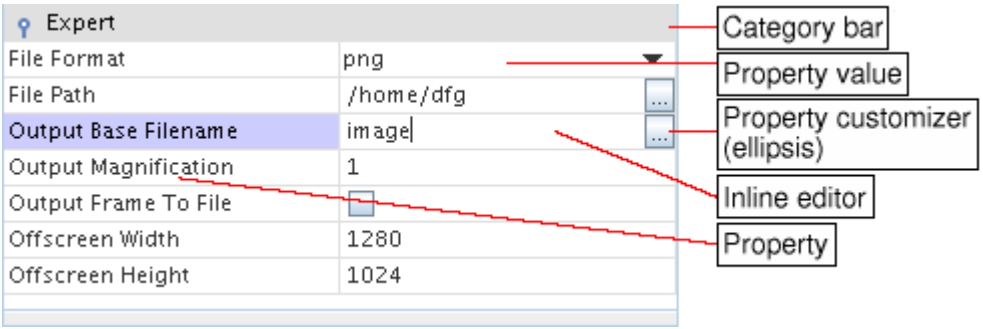
Using Properties Categories

Properties are grouped in different categories for ease of use, depending on the tree that is visible in the *Explorer* pane, with the most commonly accessed properties in the top category, which is visible by default.

- A simulation tree always has a *Properties* category, and may optionally have an *Expert* category depending on which node is selected.

  You are typically not required to visit the *Expert* category, as this holds properties that usually do not need to be modified.
- The Servers tree always has a *Properties* category, and may optionally have an *Expert* category depending on which node is selected.
- A handle controls each category inside its category bar. The handle can be:
  - closed (the properties are not visible); or
  - open (the properties are visible).

Each category in the *Properties* window contains a list of properties for the selected node in the *Explorer* pane arranged in two columns. The property name is on the left and the property value is on the right.



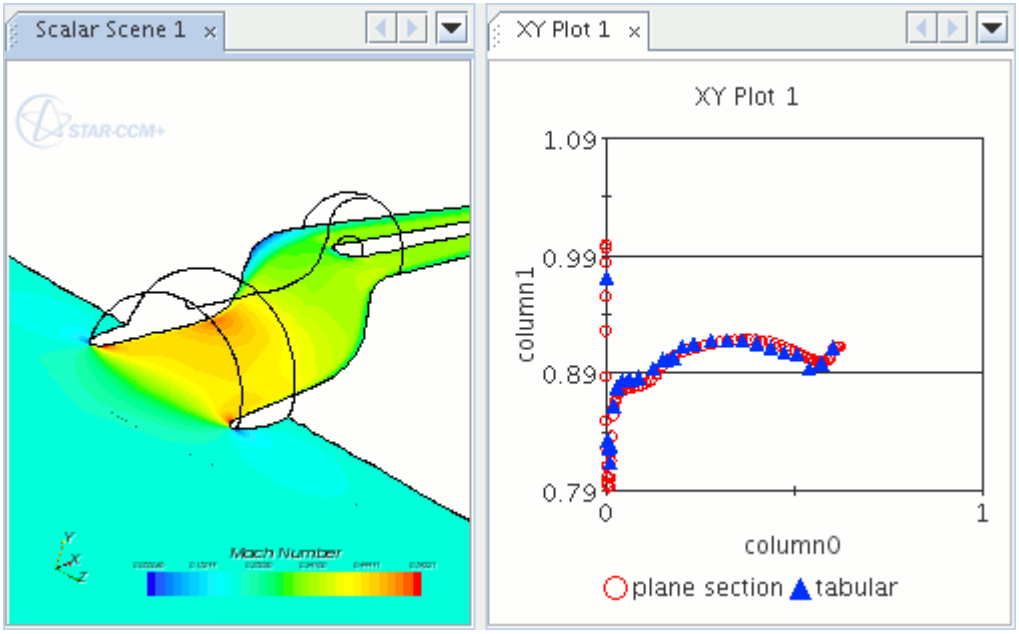
Pop-Up Menu

Pop-up menu items give you more control over the information in the *Properties* window.

Sort by Category	Sorts the properties by type. This feature acts as a radio button.
Sort by Name	Sorts the properties in ascending alphabetical order. This feature acts as a radio button.
Show Description Area	Controlled by a checkbox: Shows the description area when ticked, and hides it when cleared.

Using the Graphics Window

The Graphics window can contain Graphics windows and/or plot windows -- powerful tools that let you view developments in the simulation in real time, as well as plot data.



Continuous Updating

The scenes in STAR-CCM+ always display an up-to-date view of the server objects.

When the solution changes (after an iteration for example), all objects that are viewed in the GUI are updated to reflect the new values. This feature is valuable for monitoring the progress of a solution and being able to make corrections to the settings and conditions to allow you to get to a faster solution.

The downside of this feature is that continuous updating can be expensive and can slow down the server in some situations. The solution would be to close any open scene displays, or at least scenes which are no longer essential.

Contents:

Working with Displays
Splitting the Window
Using Scenes to Interact with the Object Tree
Working with Graphics Windows
Selecting Parts by Type in the Display
Changing a Three-Dimensional Visualization Display
Changing a Two-Dimensional Visualization Display
Interacting with the Plot Display

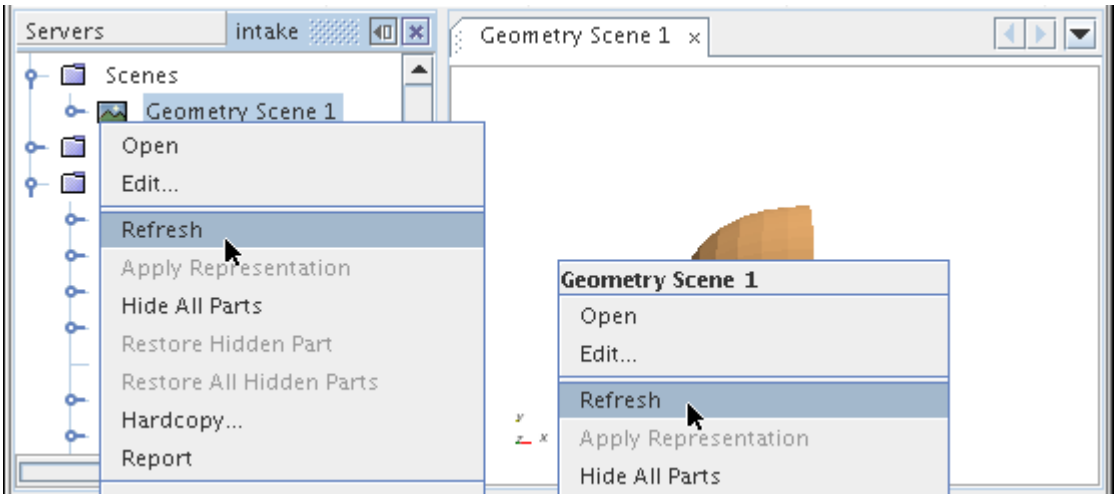
Plot Tab Reference

Working with Displays

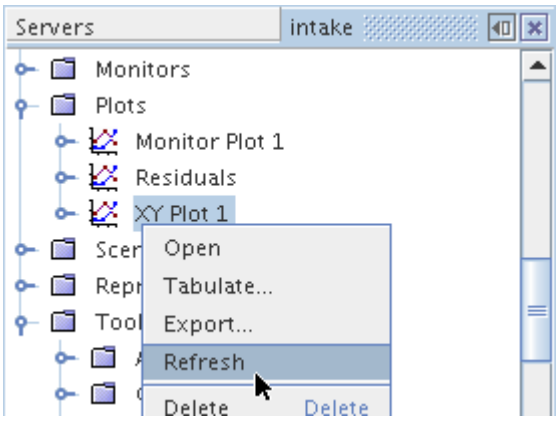
Visualization or plot displays are a way to make the best use of the STAR-CCM+ GUI.

Refreshing the Display

Typically the display of a scene or XY plot updates automatically to reflect the latest changes to the simulation. However, in case it fails to do so, you can force an update. To force a scene to update, right-click the scene node, or in the background of its display, and select **Refresh**.



To force an XY plot to update, right-click the XY plot and select **Refresh**.



If you select this menu item when the display of this scene or plot is not open or active in the *Graphics* window, the open scene or open plot action is triggered.

Rearranging Displays

You can reposition displays inside the *Graphics* window with a drag-and-drop mouse movement. To rearrange displays:

1. Click the tab of the display that you wish to move. As you begin to drag the tab, the display has a red outline.
2. Drag toward the desired position until you see the red outline jump to that position.
3. Release the mouse button. The display and its tab are in the new location.

Bringing a Display Tab to the Foreground

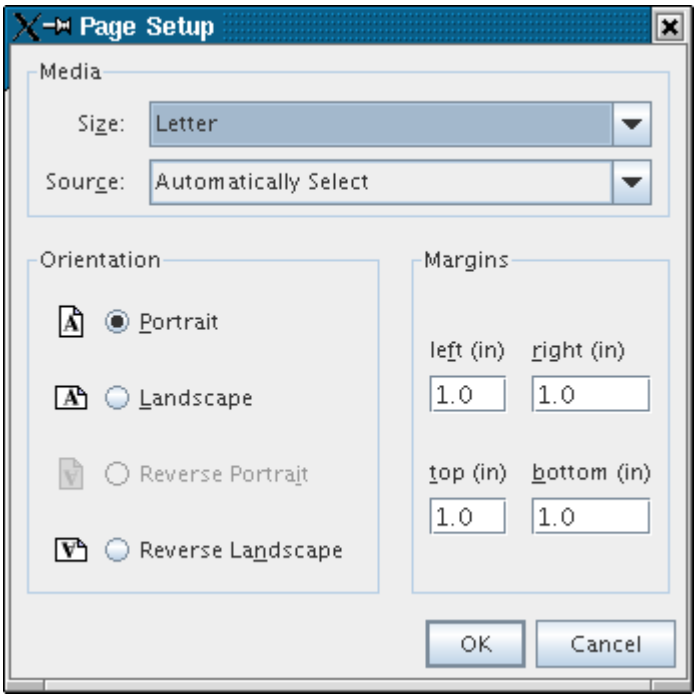
Click the tab with the left mouse button.

There are no keyboard alternatives for navigating through the tabs.

Controlling the Page Setup of a Display

To adjust the print settings of a display, select **File** > **Page Setup...**.

The operating system-specific page setup dialog appears.



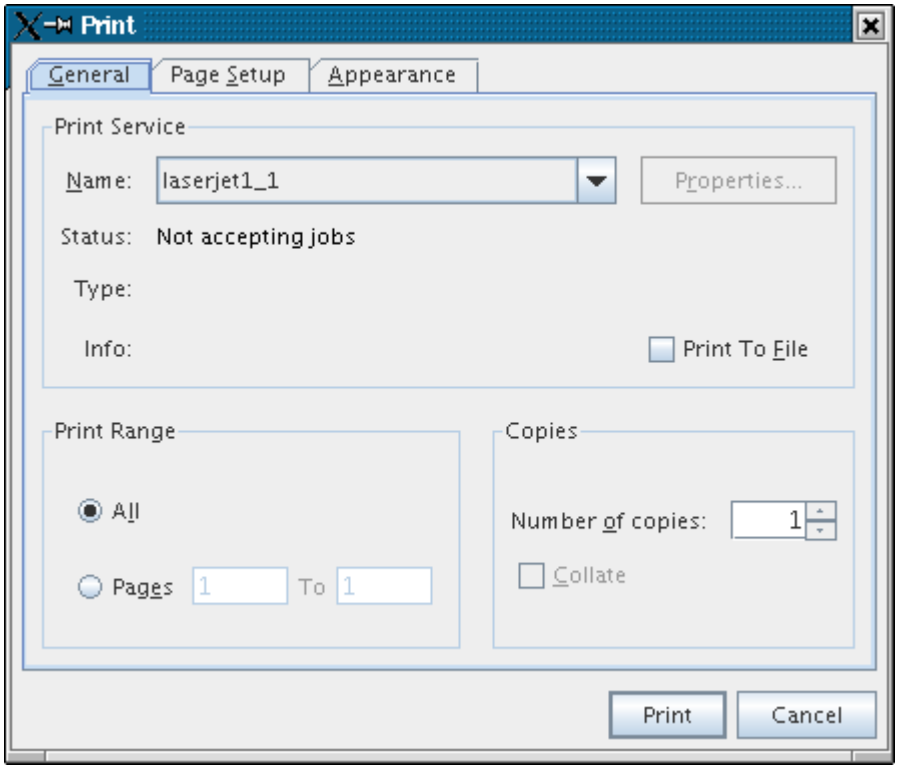
Here, you can specify detailed aspects of the print job, such as margins and page orientation.

Printing a Display

To print a display:

1. Select **File** > **Print...** or press `<Ctrl><P>` on the keyboard.

   The operating system-specific standard print dialog appears.



2. Click **Print** (or **OK**) to print the contents of the active display in the *Graphics* window.

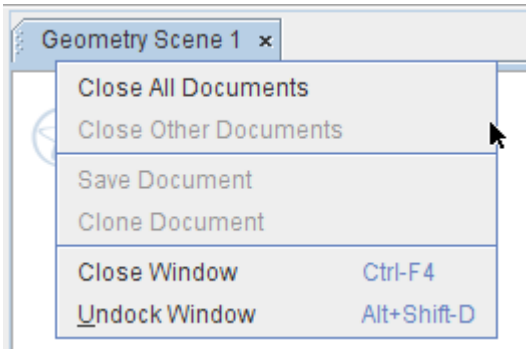
   The contents of this dialog are not documented further. Refer to your operating system documentation for details of using this dialog.

Note: This action is only available if a simulation is visible and a display (scene or plot) is visible in the *Graphics* window (it does not have to have the focus).

Using Pop-Up Menus

The pop-up menu that appears on each display tab gives you an extra measure of control over these windows.

To activate the pop-up menu, right-click on the display tab.



The menu items refer to these displays as “documents” because they can be printed, saved to a graphic, and updated in real time.

See Also:

Plot Tab Reference
Working with Graphics Windows
Interacting with the Scene
Plotting Results
What Is an XY Plot?
What Is a Scene?

Splitting the Window

STAR-CCM+ lets you display two or more scenes simultaneously, as well as revert to displaying a single scene.

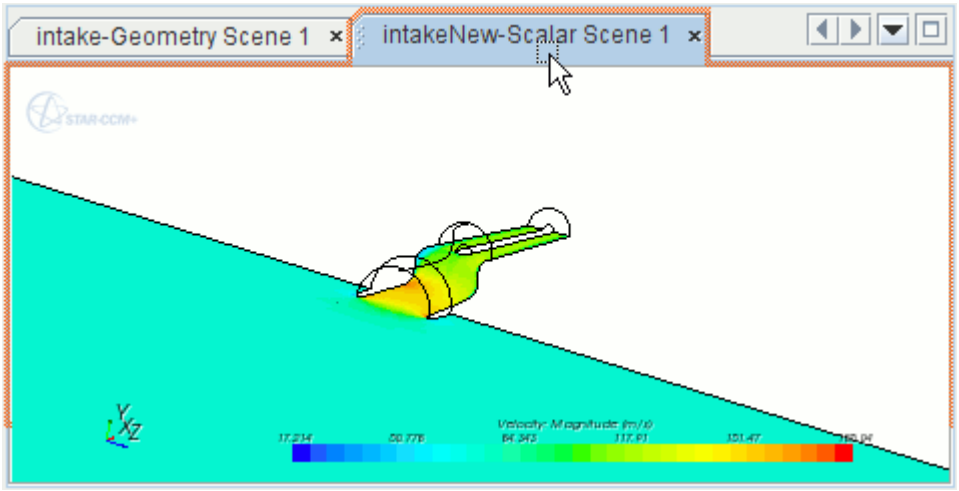
You may need to compare subtle patterns in two different displays, say, scalars and vectors. For this reason, it would be helpful to look at multiple scenes at the same time, rather than constantly working with tabs.

Viewing Two Scenes Side-by-Side

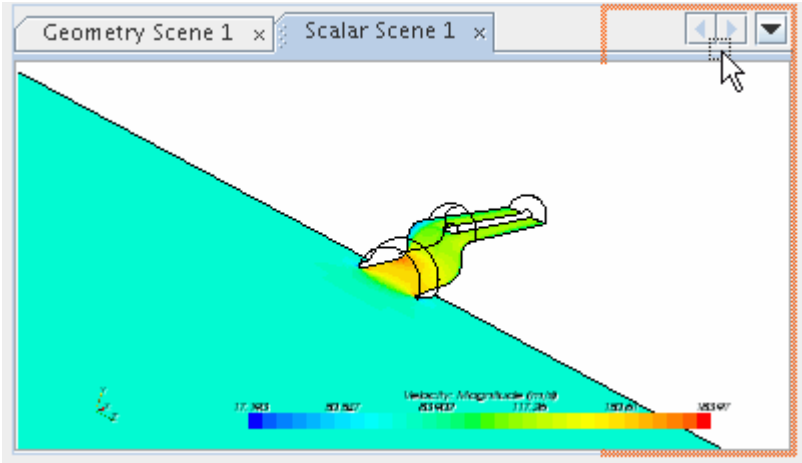
Scenes can share space in the *Graphics* window.

Using the left mouse button, drag the tab of one of the displays in the direction where you want it to appear. In the illustrations below, the *Geometry Scene 1* display tab is dragged to the right so that the display occupies the right half of the *Graphics* window.

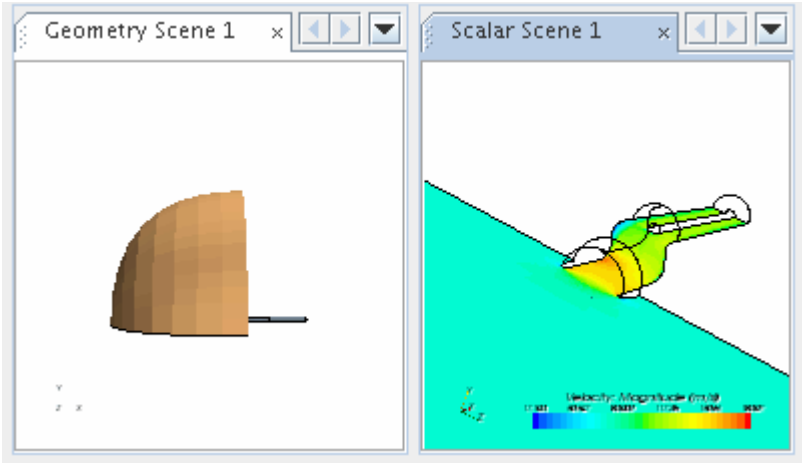
A red outline appears around the scene display as you begin to move it.



As you continue to move the mouse in that direction, the outline changes to show how the *Graphics* window will be split between the scenes.



Let go of the mouse button. The scenes share space in the *Graphics* window.



You can use the same technique to divide the *Graphics* window further among more scenes.

Viewing Scenes from Two Simulations Side-by-Side

At times, you may need to view scenes from two or more simulations simultaneously to compare data. This can be done with the window-splitting capability in STAR-CCM+.

1. With one or more scene displays open in one simulation, open a second simulation in the same STAR-CCM+ client. This results in two open simulation windows in the *Explorer* pane that you can switch between.
2. Open a scene display in the second simulation.
3. Split the *Graphics* window.

   The two scene displays from the separate simulations are side-by-side for easy comparison.

   When more than one simulation is open, the tab includes the name of the corresponding simulation. This feature lets you know at a glance which simulation a display represents.

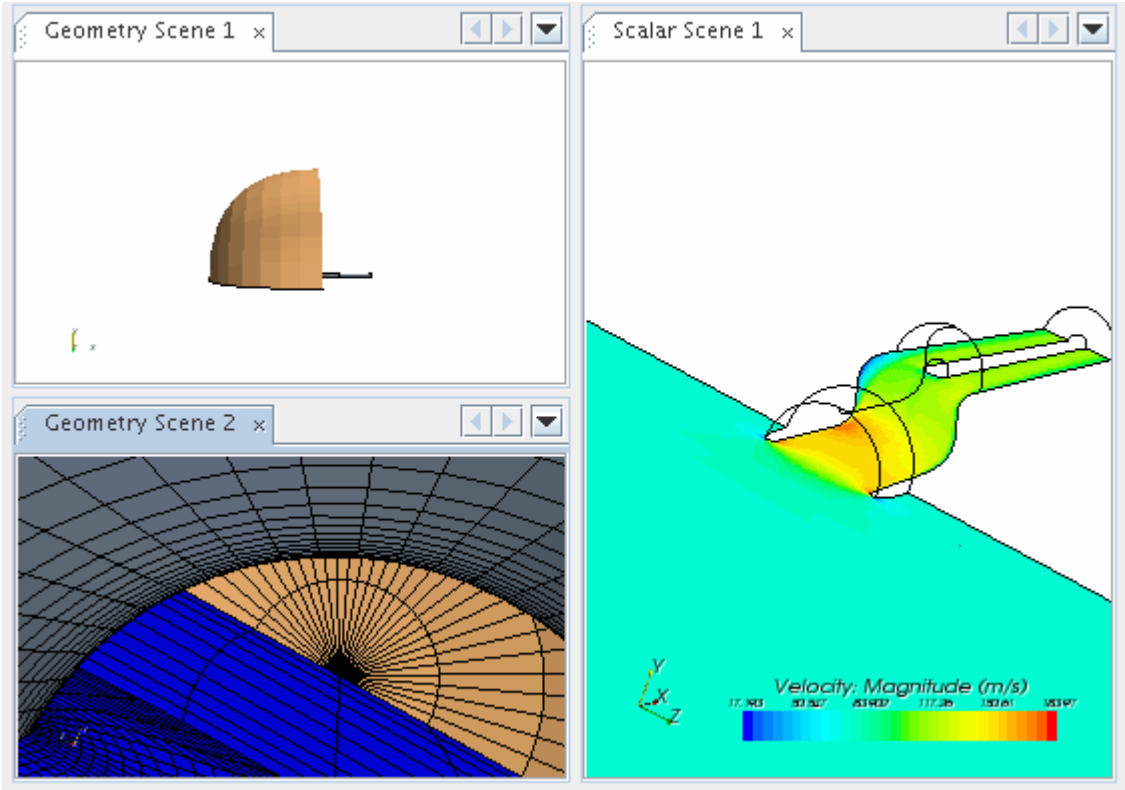
Showing More Scenes Simultaneously

You can arrange more scenes in a column alongside a main scene.

To do this, drag the tab of the additional scene downward over the open scene with which it should be stacked.

As before, keep dragging until the outline changes to indicate a split.

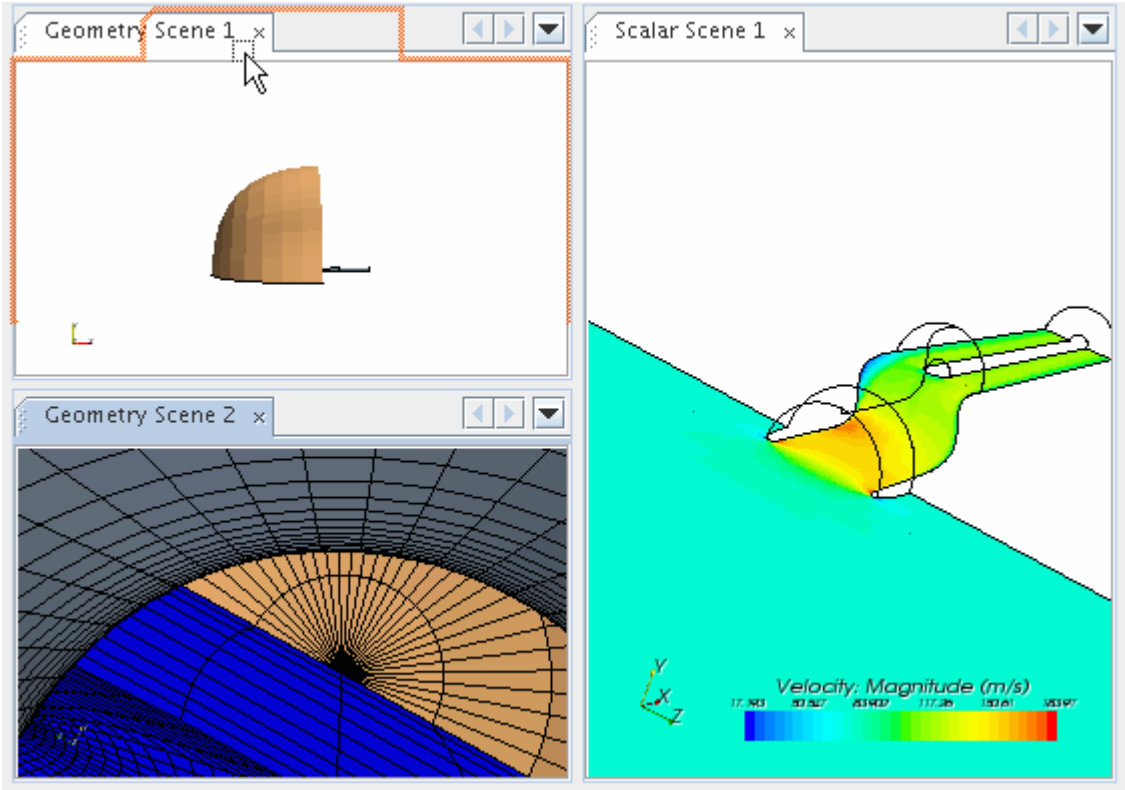
This action creates the stack.



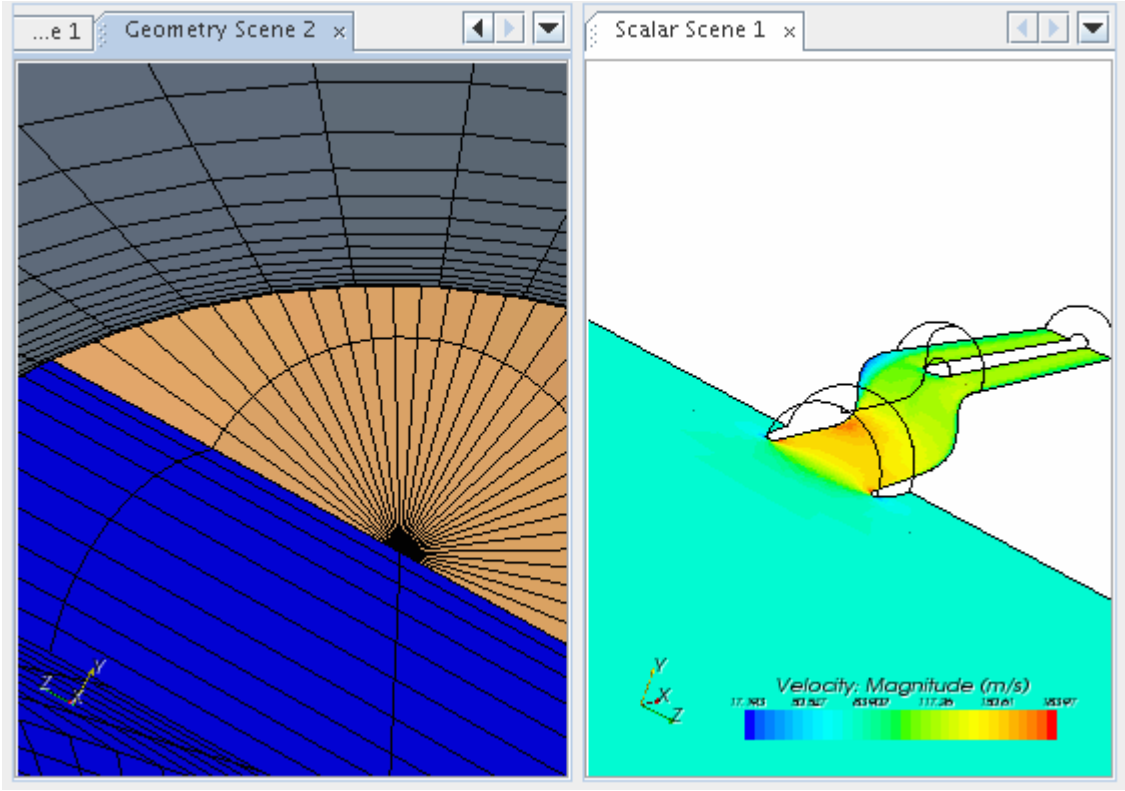
Reverting to Displaying One Scene

The drag-and-drop technique that is used for splitting scenes can also be used to reverse the splits.

To reverse a split, drag one tab over another. In the illustrations below, the scene in the lower left corner of the *Graphics* window is being dragged over the one in the upper left corner.



As a result, the scene that you drag over the other now takes the entire left half of the *Graphics* window. Only two of the three scene displays are active, while the third is open but inactive.



You can keep following this procedure until only one scene display is active at a time.

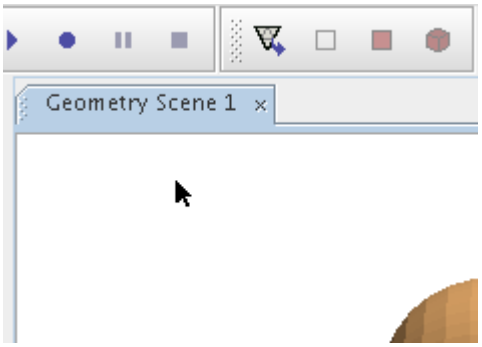
Using Scenes to Interact with the Object Tree

The sets of instructions below show how you can use the various connections between the *Graphics* window and the object tree.

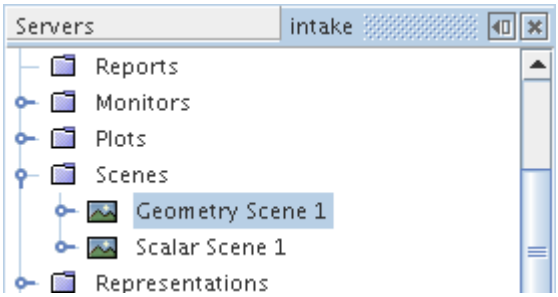
Finding a Scene in the Tree

This section explains how to determine which node in the object tree corresponds to a display window.

Click the background of a display window.



The label of the node corresponding to that window is highlighted.

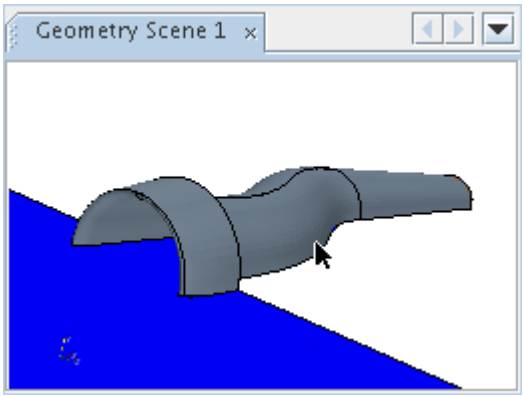


When more than one simulation is open, the tab includes the name of the corresponding simulation. This feature lets you know at a glance which simulation a display represents.

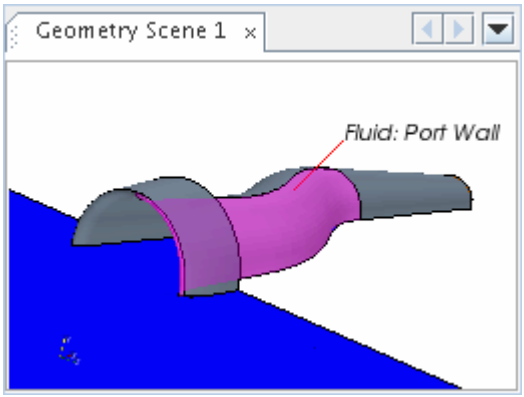
Highlighting Objects

When you click an object in either the display window or on the object tree, that object is highlighted in both locations simultaneously.

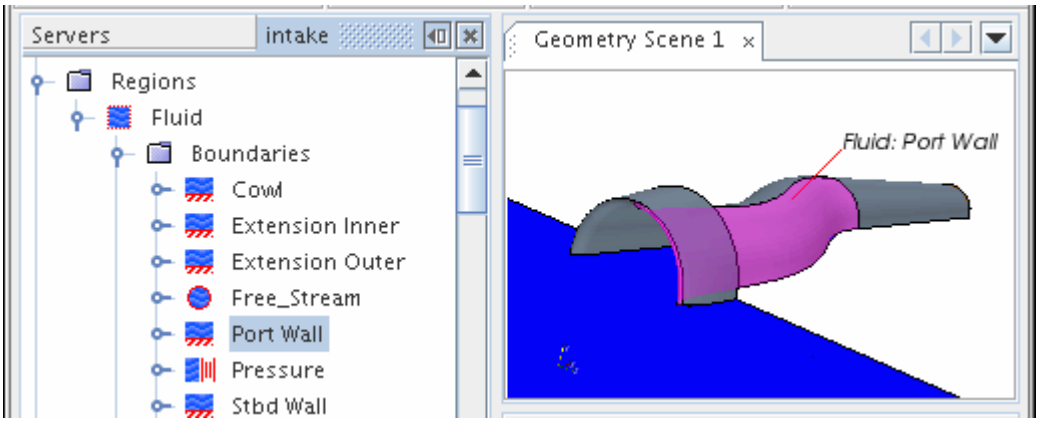
Click a part inside a window.



The part becomes labeled and highlighted.



The label of the boundary node corresponding to that part is also highlighted in the simulation tree.



Working Without the Default Highlighting Behavior

The default behavior can be stopped by deactivating the *Highlight Parts in Scene* option in the **Visualization** node of the *Options* dialog. Deactivating this option can improve the speed of response in the *Graphics* window when working with models that contain many surfaces. It is also possible to highlight objects in a scene without the labels.

To highlight parts, part contacts, part surface contacts, and objects in custom mesh controls in the *Graphics* window when the *Highlight Parts in Scene* function is deactivated, do one of the following:

- Right-click the node you wish to highlight and select **Highlight**.
- Double-click the object in the scene.

Controlling the Number of Highlight Captions

To improve the readability and graphics performance of the scene, you can set the maximum number of captions that appear with highlighted objects.

To change this setting:

1. In the *Options* dialog, expand the **Options** > **Visualization** node.
2. Set *Maximum Highlight Captions* to the maximum value you wish to allow.

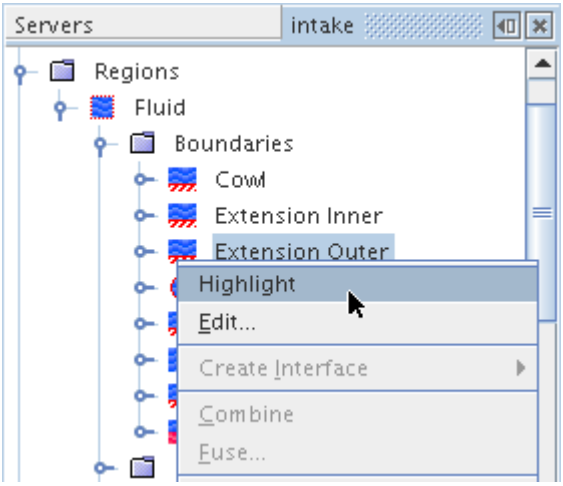
   When the total number of highlight captions in the scene exceeds this value, all captions are omitted from the scene.

Highlighting an Object Using the Tree Node

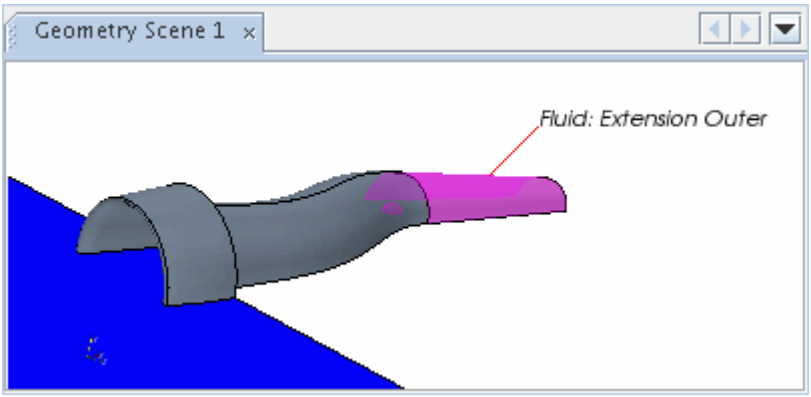
You can highlight an object in a scene by right-clicking on its node in the object tree.

The same works for a region or part node. This functionality also works for multiple objects. Additionally, it is possible to highlight objects in a scene without the labels.

1. In the object tree, right-click the node of a boundary and select **Highlight**.



   The object is highlighted in the display.

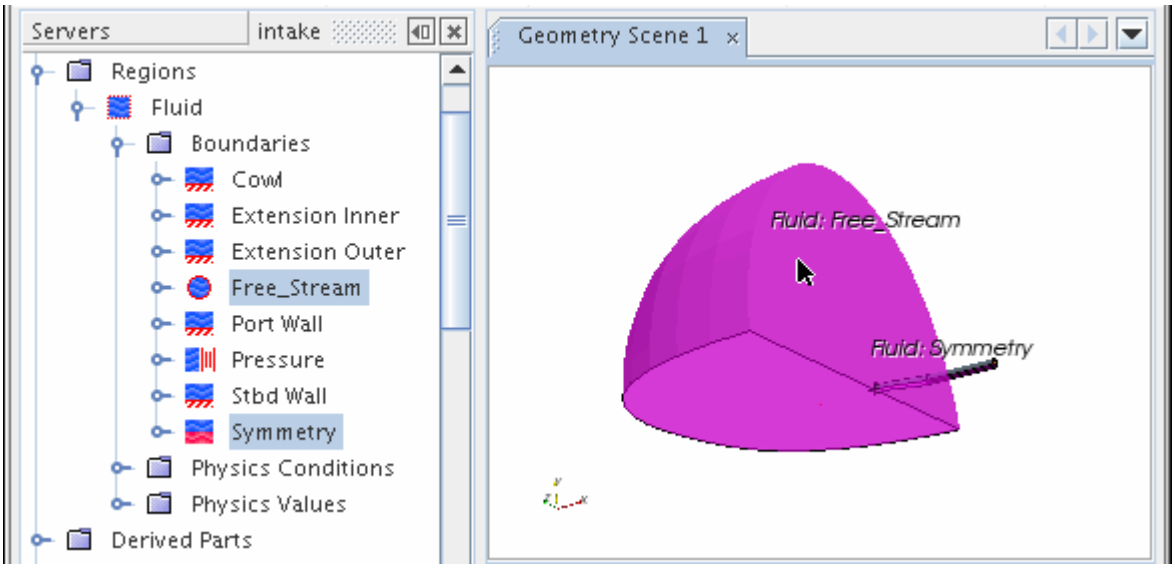


2. Use the (**Reset**) button in the toolbar to view the entire shape.

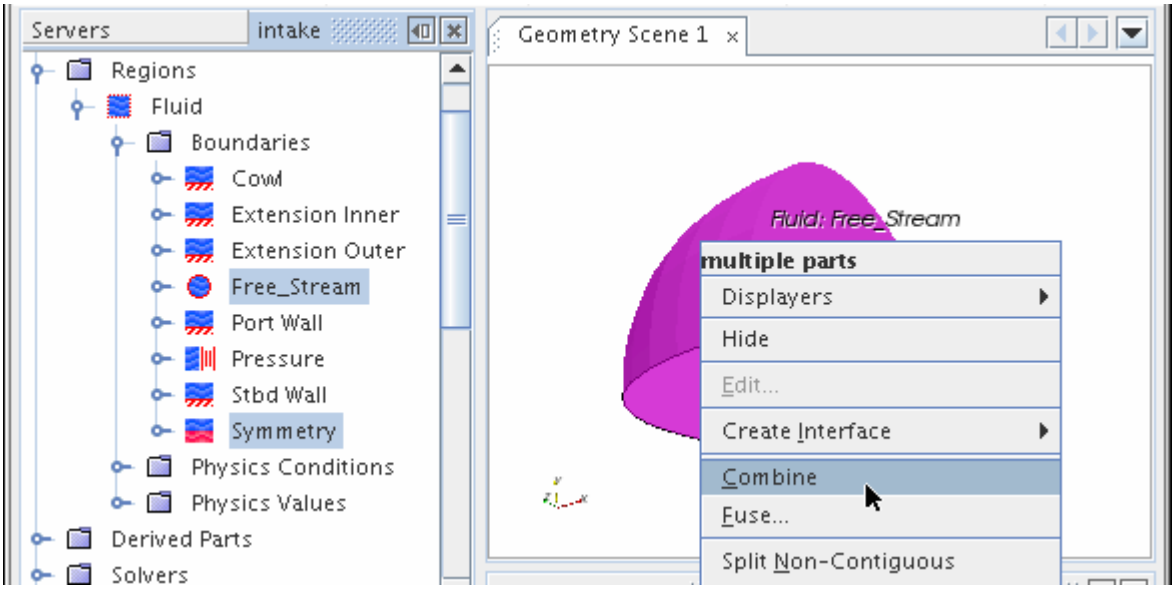
Synchronizing Multiple Objects

It is possible to work visually with multiple objects by selecting them in a scene.

To select multiple objects in a scene, hold down the `<Ctrl>` key while clicking them. The labels of the corresponding nodes in the object tree are also highlighted.



Depending on what type of objects you select, right-clicking on one of the selected objects activates certain pop-up menu options. For example, right-clicking on one of two selected boundaries in a scene allows you to combine those boundaries.



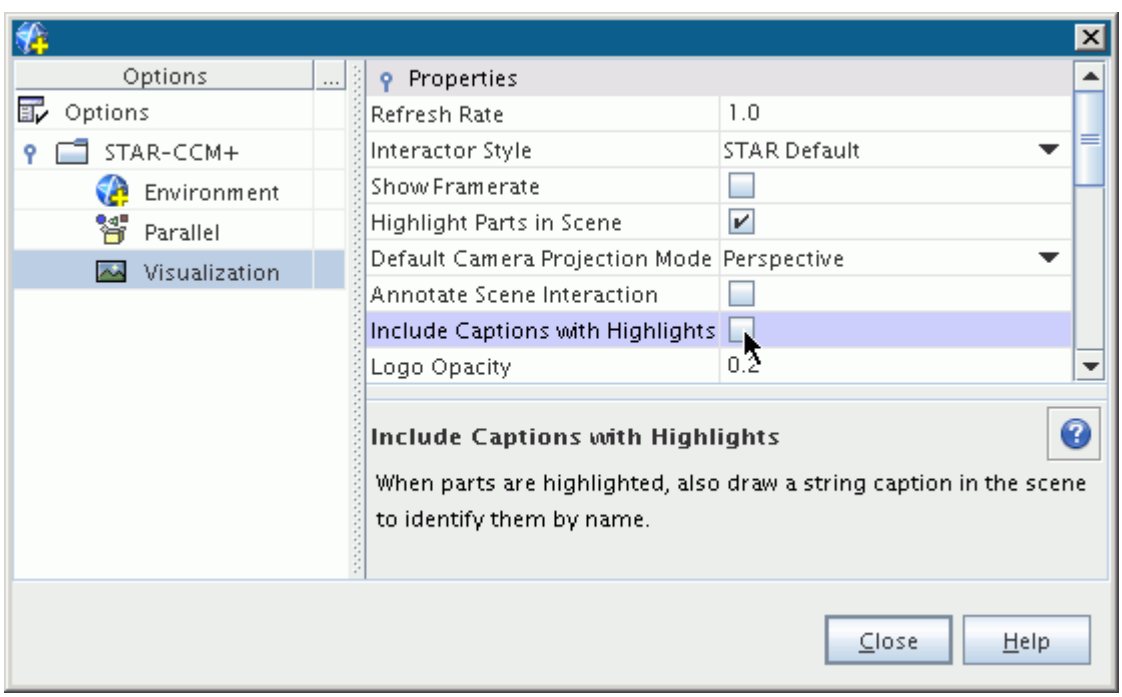
Highlighting Objects Without Labels

It is possible to highlight a boundary in a scene without the label of the boundary appearing. This facility is useful if your simulation contains many boundaries, when the display of the labels can take a long time to load.

To change the setting:

1. Select **Tools** > **Options**.

   The *Options* dialog appears.



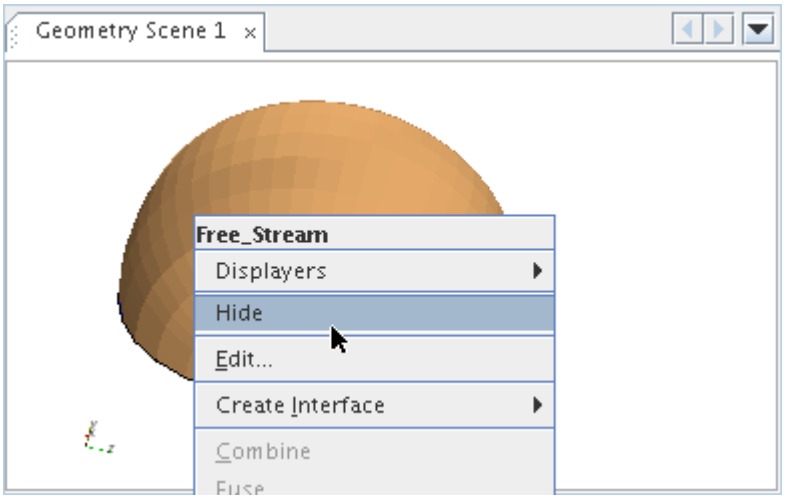
2. Expand the **STAR-CCM+** node in the **Options** node.
3. Select the **Visualization** node and deactivate the *Include Captions with Highlights* option to omit labels from highlighted objects in the display.
4. Click **Close**.

Accessing Objects Obscured from View

There are several ways to get access to an object in a scene when it is obscured by other objects.

Hiding Objects

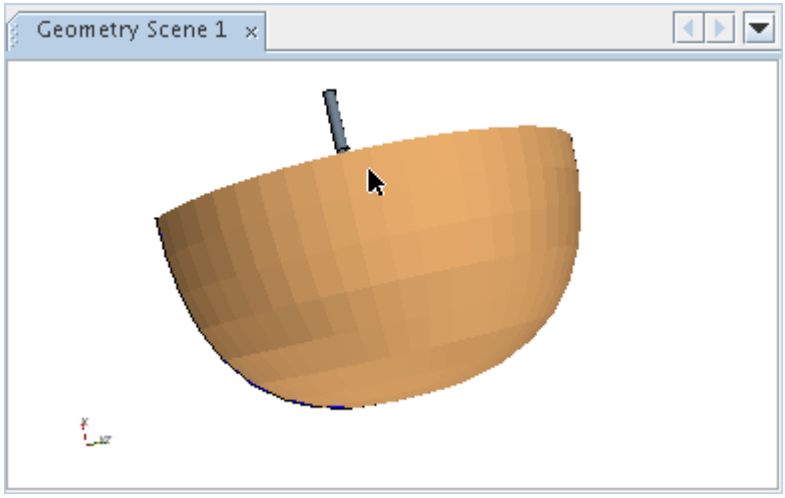
The most common technique is to hide any objects that are in the way.



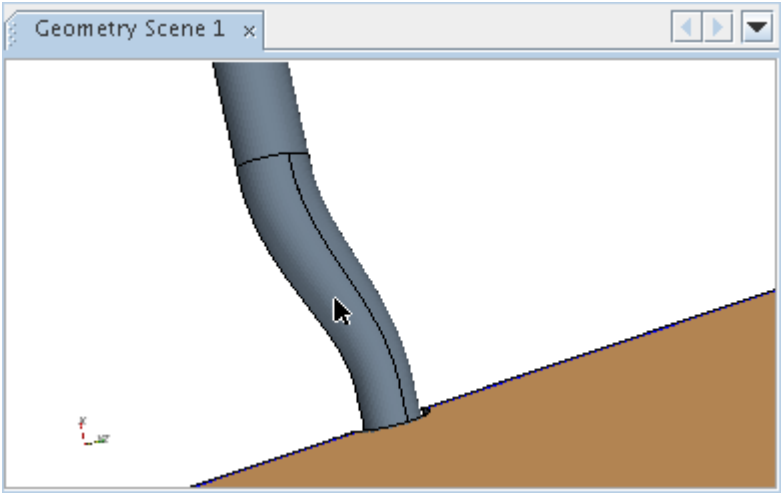
Rotating and Zooming Toward Objects

Rotating the object slightly can give you access to distant points.

1. Move the mouse until you begin to see the object that you want to access.



2. Click that part of the object and hold down the mouse button. This action establishes the focal point at that distant location; you can either rotate the object around that point, or zoom toward it.

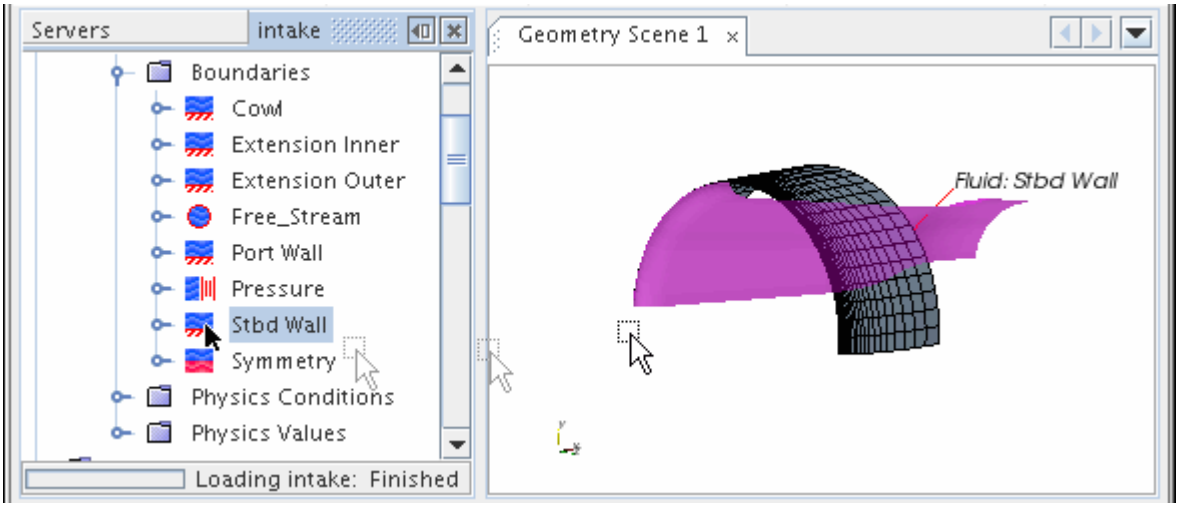


You can also use nodes in the object tree.

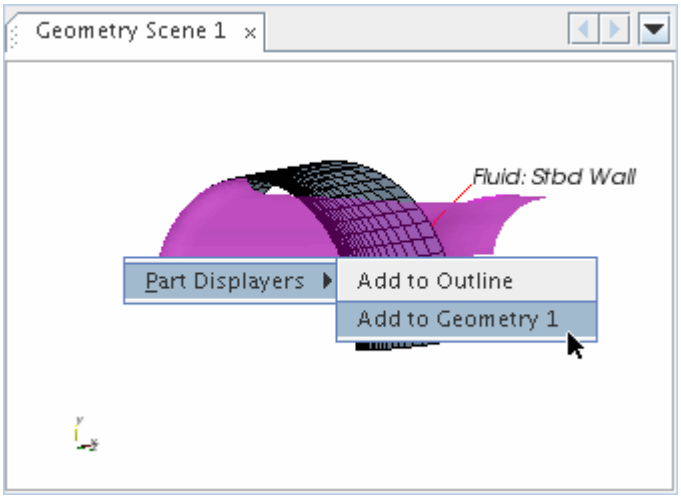
Dragging Parts onto a Scene

You can use the drag-and-drop technique to add parts to an active scene display. The nodes in the object tree function as tangible, viewable objects.

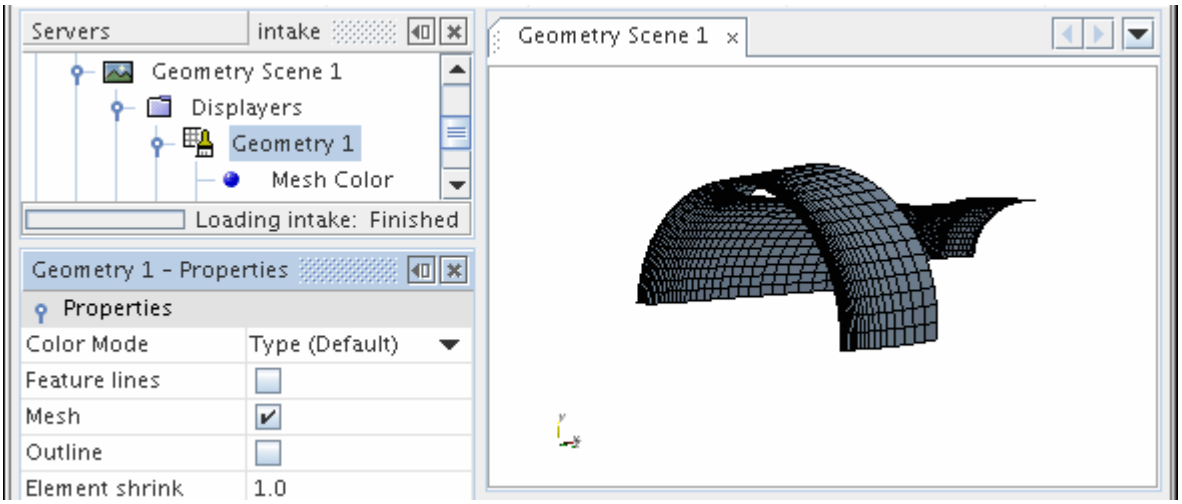
Drag the boundary node into the *Graphics* window.



A pop-up menu appears letting you choose which part displayer receives the part.



In the display window, the part that you added has the properties that are specified in the part displayer that you selected from the pop-up menu.



See Also:

Combining Boundaries

Using Tree Nodes to Access Objects Obscured from View

Hiding a Part in a Scene

Changing a Three-Dimensional Visualization Display

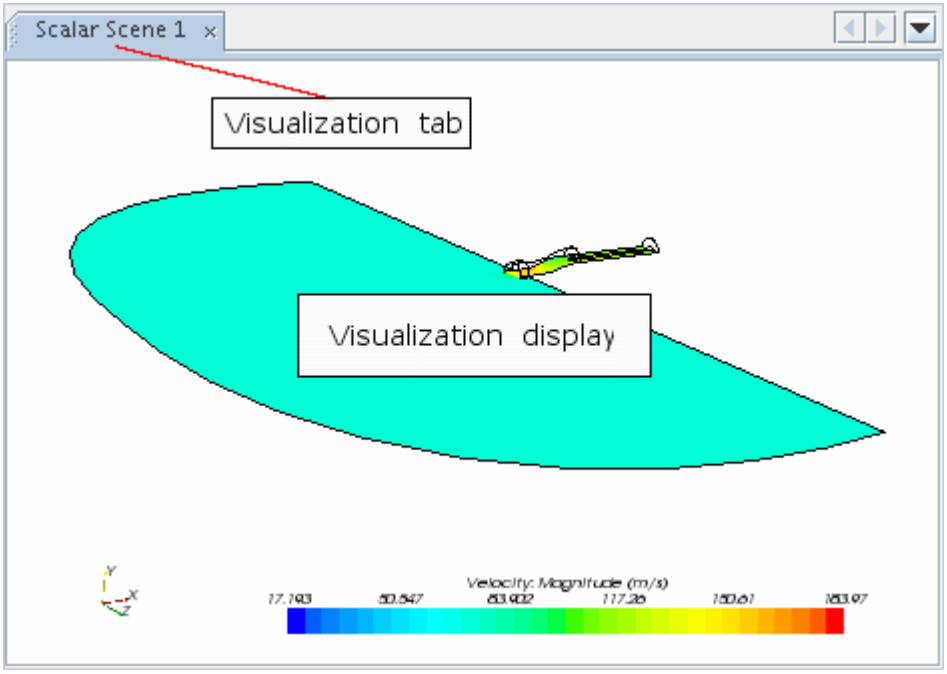
Changing a Three-Dimensional Visualization Display

Selecting Parts by Type in the Display

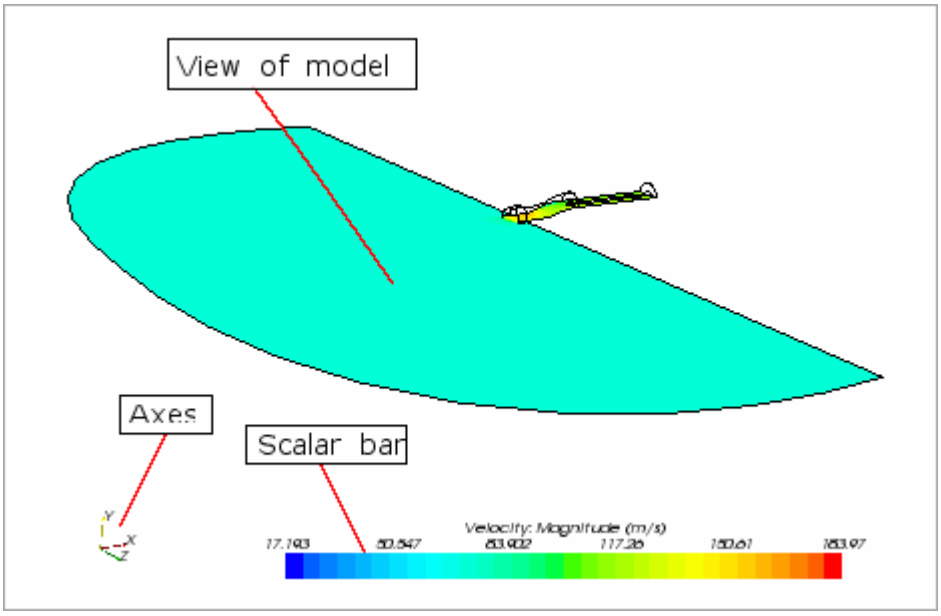
Working with Graphics Windows

A Graphics window contains a tab and a display (the white space that holds the image and other widgets).

The tab is named the same as the node in the object tree. If multiple simulations are open at once, then the tab includes the name of the corresponding simulation. You can open a Graphics window by double-clicking the corresponding scene node in the tree. If the window for a particular scene has been opened, but is not active, you can double-click the relevant simulation tree node or click its tab in the *Graphics* window.



The visualization display is the space inside the Graphics window where the model is shown with various display objects such as the axes, scalar and/or vector bars, and textual legend.



In addition to the selection capabilities, the following operations let you get the most out of the Graphics window:

- Renaming parts in the scene
- Using the mouse
- Activating menus by right-clicking
- Accessing hidden objects using tree nodes
- Selecting scalars and vectors using scalar and vector bars
- Using the distance measuring tool
- Changing the view
- Moving the image around the x, y or z-axis
- Obtaining scalar data with the cell probe

Contents:

Renaming Parts in the Scene
Using the Mouse in the Visualization Display
Using Tree Nodes to Access Objects Obscured from View
Using the Scalar and Vector Bars
Using the Distance Measuring Tool
Changing the View
Obtaining Scalar Data with the Cell Probe

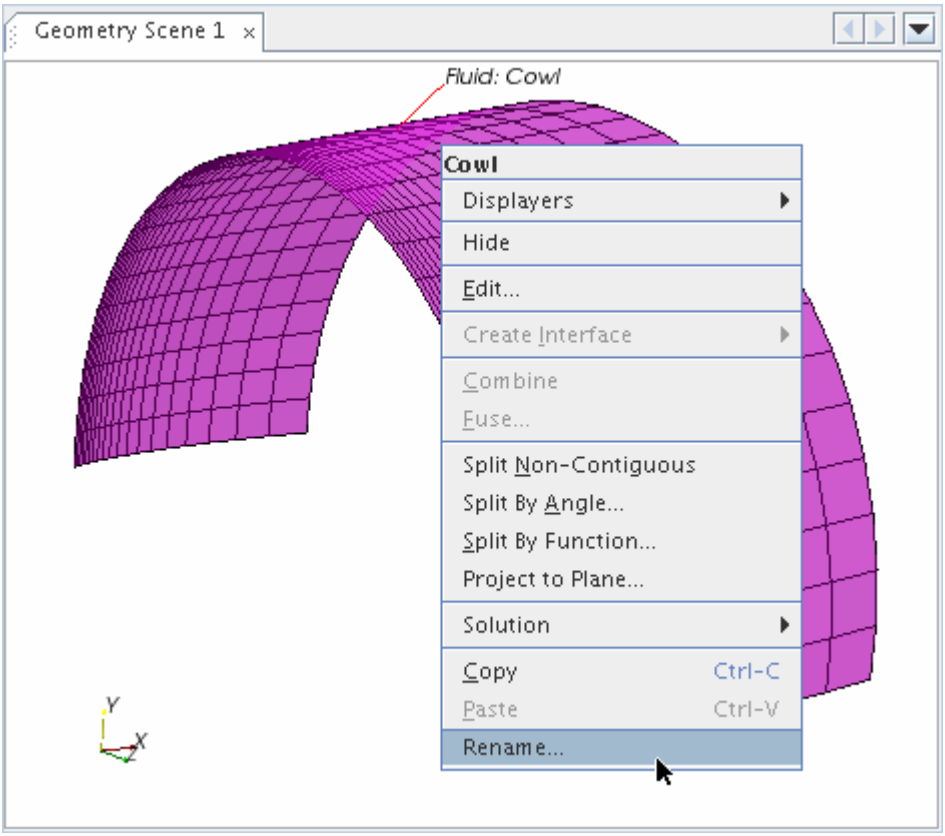
See Also:

Working with Displays

Renaming Parts in the Scene

This section explains how to change the name of a part in the scene.

1. To change the name of a part in the scene, click once on it in the visualization display.
   This highlights that part.
2. Right-click on it to activate the pop-up menu and select **Rename...**



3. Enter the new name of the part in the inline editor.
4. Press `<Enter>`.

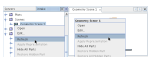
   The new name appears in both the simulation tree and the label of the part (when you select it).
5. Alternatively, right-click the boundary node in the object tree and select **Rename...**

Using the Mouse in the Visualization Display

Each mouse button, sometimes combined with a keystroke, affects the view.

Right-Clicking

If you right-click inside the display, you get a pop-up menu identical to that of the node associated with that display.



A similar functionality exists for boundaries in the object tree and related objects in the scene display. The pop-up menu for a part that is selected in the visualization display contains all of the boundary-related options that exist in the corresponding node's pop-up menu.

Mouse Configurations

In the following text:

- LMB = left mouse button.
- MMB = middle mouse button.
- RMB = right mouse button.

The default configuration for mouse interaction in STAR-CCM+ is shown below. The rotate, roll, and zoom functions operate around the point where the mouse was first clicked. So in zoom, for example, the point on the model where the mouse was clicked remains at the same point on the display as the rest of the model shrinks or expands around that point.

<LMB>	Rotates the view. Moving the mouse horizontally rotates the view around the vertical axis. Moving the mouse vertically rotates the view around the horizontal axis. Moving the mouse at an angle, combines the two rotations.
<MMB>	Zooms the view in or out. Moving the mouse up, makes the view larger, moving the mouse down, makes the view smaller. Moving the mouse horizontally has no effect.
<RMB>	Pans the model across the screen in the direction of the mouse movement.
<Ctrl><LMB>	Rolls the view, that is, rotates it around the axis perpendicular to the screen. Moving the mouse down rotates clockwise, and moving it up rotates the view counter-clockwise. Moving the mouse in the left or right direction has no effect.

Within STAR-CCM+, it is possible to change the configuration of the mouse in order to suit your preferred working environment. You can change the mouse configuration in the *Options* dialog (**Tools** > **Options**) under the **Visualization** node by selecting an alternative *Interactor Style*. The default option, **STAR Default**, is described above, and the rest of the styles are described below.

Also in the *Options* dialog, you can add an annotation in scenes that indicates the mouse action in progress. To do this, activate the *Annotate Scene Interaction* property.

Pro-E / NX

This interactor style matches the mouse configuration in the Pro-Engineer and NX CAD packages. The rotate, roll, and zoom functions operate around the point where the mouse was first clicked.

<MMB>	Rotates the view. Moving the mouse horizontally rotates the view around the vertical axis. Moving the mouse vertically rotates the view around the horizontal axis. Moving the mouse at some angle, combines the two rotations.

<Ctrl><MMB>	Zooms the view in or out. Moving the mouse up, makes the view larger, moving the mouse down, makes the view smaller. Moving the mouse horizontally has no effect.
<Shift><MMB>	Pans the model across the screen in the direction of the mouse movement.
<Ctrl><LMB>	Rolls the view, that is, rotates it around the axis perpendicular to the screen. Moving the mouse down rotates clockwise, and moving it up rotates the view counter-clockwise. Moving the mouse in the left or right directions has no effect.

SolidWorks

This interactor style matches the mouse configuration in the SolidWorks CAD package. The rotate, roll, and zoom functions operate around the point where the mouse was first clicked.

<MMB>	Rotates the view. Moving the mouse horizontally rotates the view around the vertical axis. Moving the mouse vertically rotates the view around the horizontal axis. Moving the mouse at some angle, combines the two rotations.
<Shift><MMB>	Zooms the view in or out. Moving the mouse up, makes the view larger, moving the mouse down, makes the view smaller. Moving the mouse horizontally has no effect.
<Ctrl><MMB>	Pans the model across the screen in the direction of the mouse movement.
<Ctrl><LMB>	Rolls the view, that is, rotates it around the axis perpendicular to the screen. Moving the mouse down rotates clockwise, and moving it up rotates the view counter-clockwise. Moving the mouse in the left or right direction has no effect.

Joystick

All the operations are executed simply by clicking the mouse, no mouse movement is necessary. The function of each mouse button is the same as for **STAR Default**, described above.

The rotate, roll, and zoom functions operate around the center of the model. So in zoom, for example, if the center of the model is at the right of the display, it will appear that the model is moving towards a point out to your right, rather than coming towards you, as you zoom in.

Trackball

All the operations are executed by dragging the mouse with the relevant mouse button held down. The function of each mouse button is the same as for **STAR Default**, described above.

The rotate, roll, and zoom functions operate around the center of the model. So in zoom, for example, if the center of the model is at the right of the display, it will appear that the model is moving towards a point out to your right, rather than coming towards you, as you zoom in.

3Dconnexion (Windows Only)

Currently STAR-CCM+ supports 3Dconnexion hardware only on Windows. Only basic view navigation through the controller cap/handle is supported. Its function keys and other special keys are not supported.

To use this hardware with STAR-CCM+:

1. Download and install the 64-bit version of the 3Dconnexion driver from http://www.3dconnexion.com/service/drivers.html.
2. From the main menu, select **Tools** > **Options**.

   The *Options* dialog appears.
3. Select the **STAR-CCM+** > **Visualization** node and activate the **Enable 3Dconnexion devices** option.

   When this option is active, STAR-CCM+ periodically polls to see if a 3Dconnexion device has been added. If it has, manipulating the device changes the view in the currently active scene.
4. Adjust the sensitivity of the device in *3Dconnexion sensitivity*.

   This value ranges from 0.0 to 1.0 and controls how much the view changes.

   You can also adjust the sensitivity of the device through the 3Dconnexion Control Panel.

See Also:

Changing a Three-Dimensional Visualization Display
Changing a Two-Dimensional Visualization Display

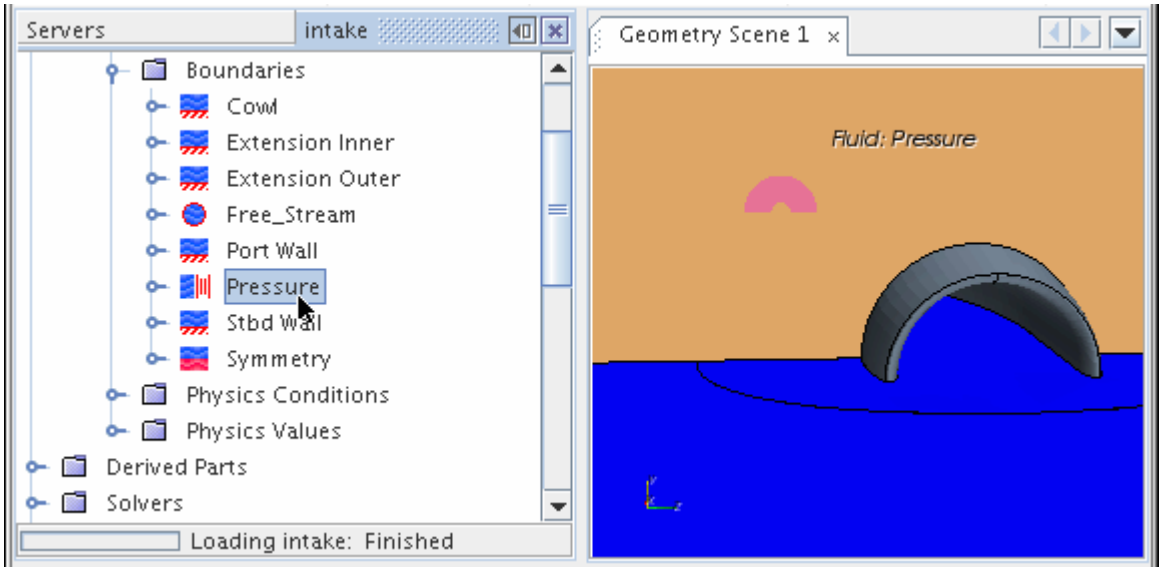
Using Tree Nodes to Access Objects Obscured from View

This section explains how to access an object behind other objects in the visualization display.

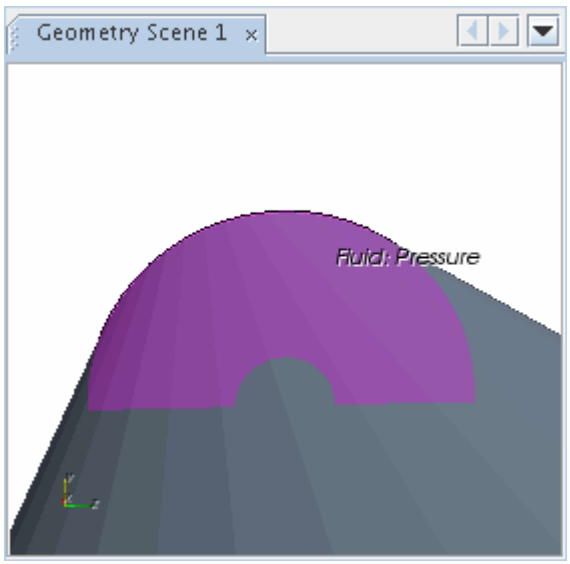
To access an object behind other objects in the visualization display

1. Click the tree node of that object.

   The object is highlighted in the display.



2. Click the (**Reset**) button in the toolbar (or the <R> key when the scene is in focus).
   The view focuses directly to that object, allowing you to click it for rotating, zooming, or other actions.



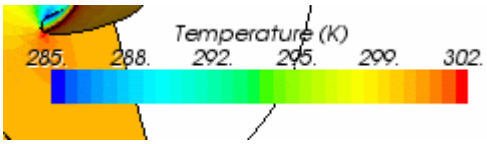
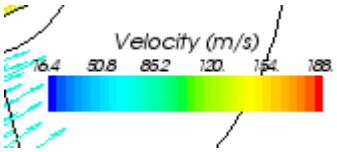
See Also:

Using Scenes to Interact with the Object Tree

Using the Scalar and Vector Bars

The scalar and vector bars appear in scalar and vector scenes, respectively. Each type of bar maps values to colors.

The bar contains value indicators and a title reflecting the scalar or vector type. The Color Bar node of the corresponding displayer controls the position, colors, and other visual characteristics of the bar.

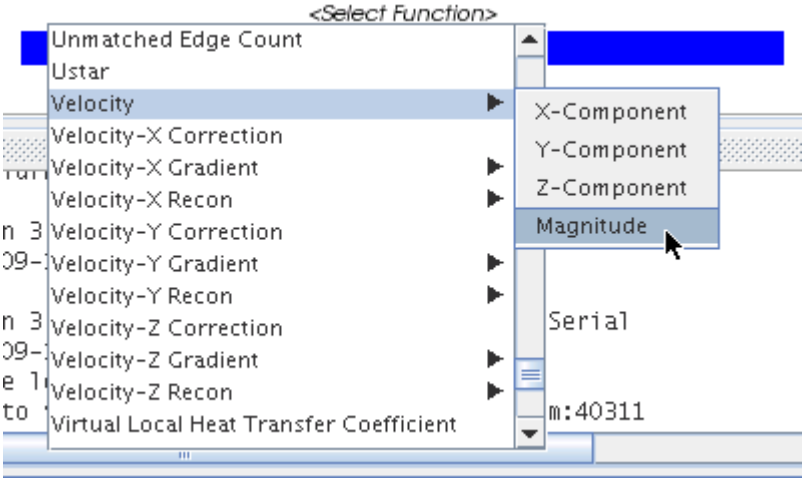
	
Scalar bar	Vector bar

When a scalar bar is used in a display of scalars on streamlines, it may be described as a “streamline bar”.

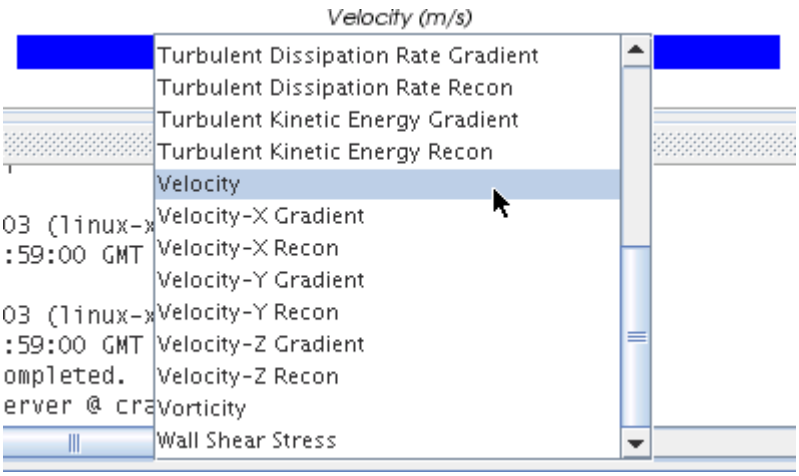
The displayer of the parent scene controls the types and ranges of scalar quantities and vector quantities. You can control scalar and vector types from pop-up menus that are activated by right-clicking the bars themselves.

The following operations can be carried out on the scalar or vector bars:

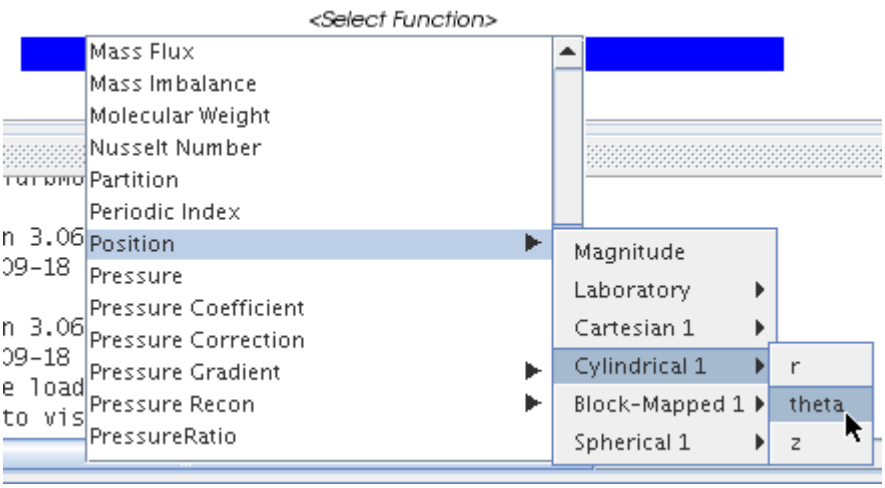
- Repositioning by dragging the bar -- change the orientation between vertical and horizontal by dragging it near a horizontal or vertical border.
- Resizing by dragging one of the edges of the bar -- the text resizes accordingly.
- Right-clicking to activate a drop-down list for the following:
  - Selecting scalar functions



  - Selecting vector functions



- Selecting the individual components of certain functions, such as positions and centroids, in local coordinate systems



You can use keystrokes to search drop-down lists.

See Also:

Accessing the Properties Window with the Keyboard
Displaying Vector Quantities
Displaying Scalar Quantities
Using the Color Bar

Using the Distance Measuring Tool

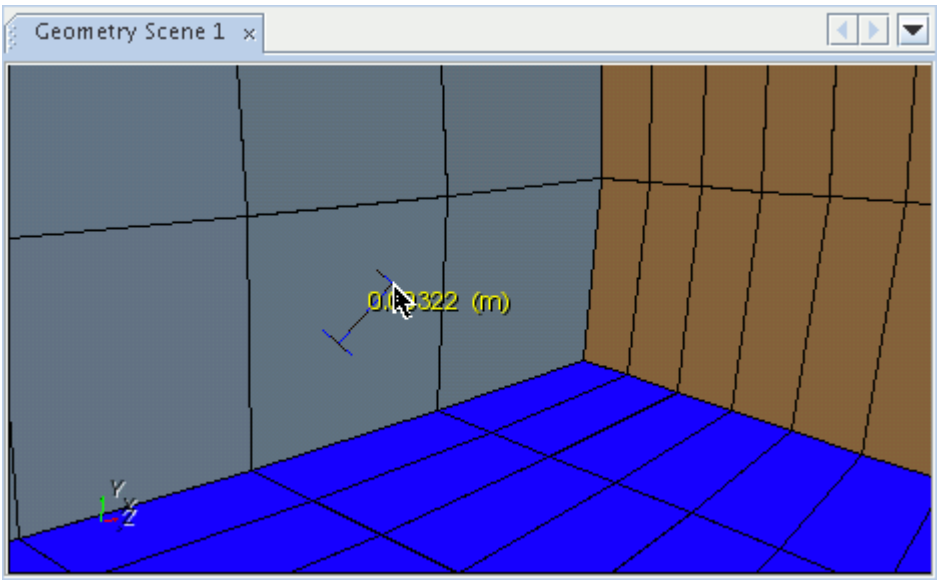
You can measure distances in the scene. Depending where in the scene you click, these distances can be among vertices, cells and/or the background of the scene.

1. To begin using this feature, click the (**Measure Distance**) toolbar button.
2. If you want to cancel this process after clicking the button, simply press `<Esc>`.

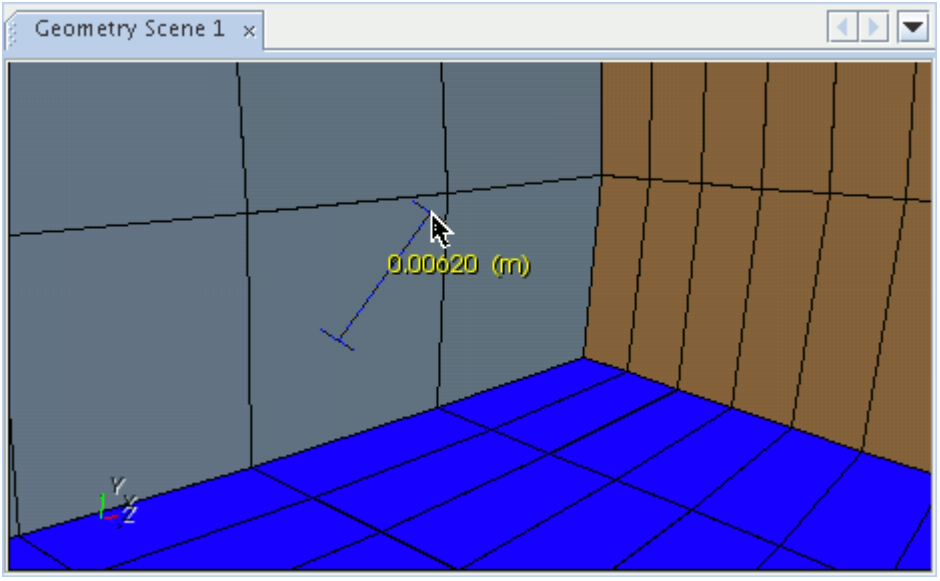
You can rotate, pan, and make other standard movements with the scene while measuring. You can also choose one of the standard or stored views using the **View** menus in the Visualization toolbar. The measurement is not final until you complete the second click and release the left mouse button.

3. Click the starting point in the scene for your measurement, and then drag the mouse in the direction you want to measure.

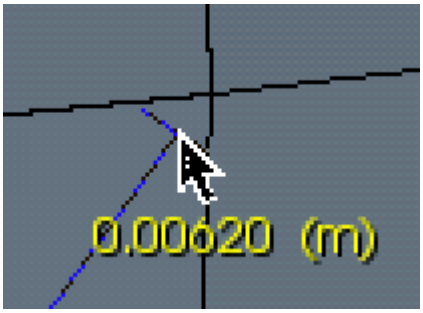
   A line (the “length bar”) with the distance appears in the scene.

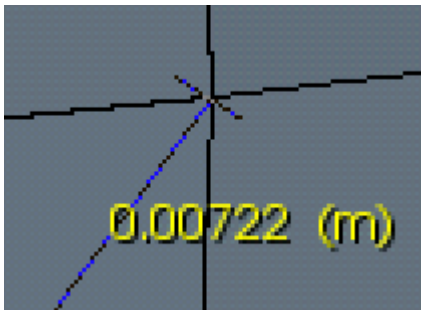


4. To complete the measurement, click again at the end of the distance.



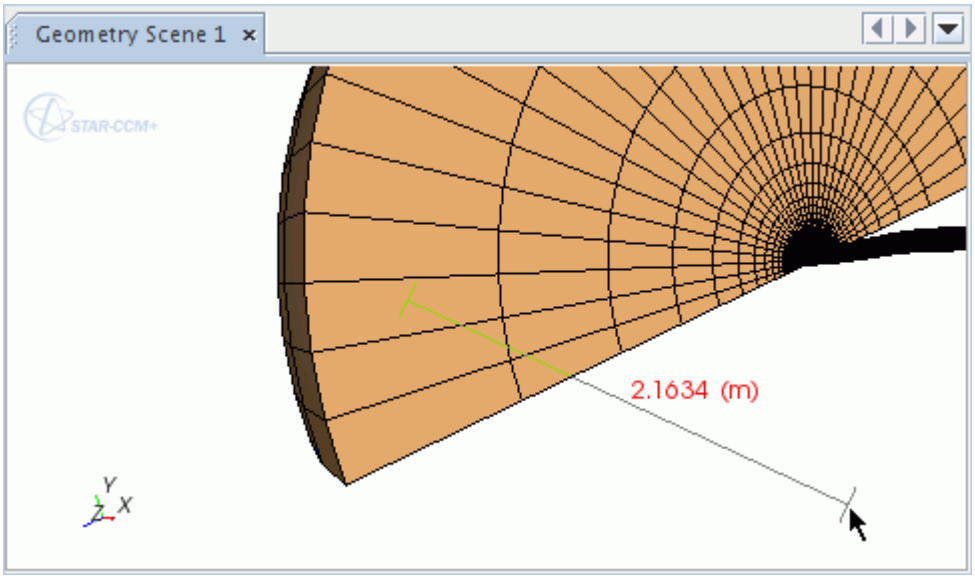
If you click near a vertex at the beginning or the end, the point that you create snaps to that vertex.





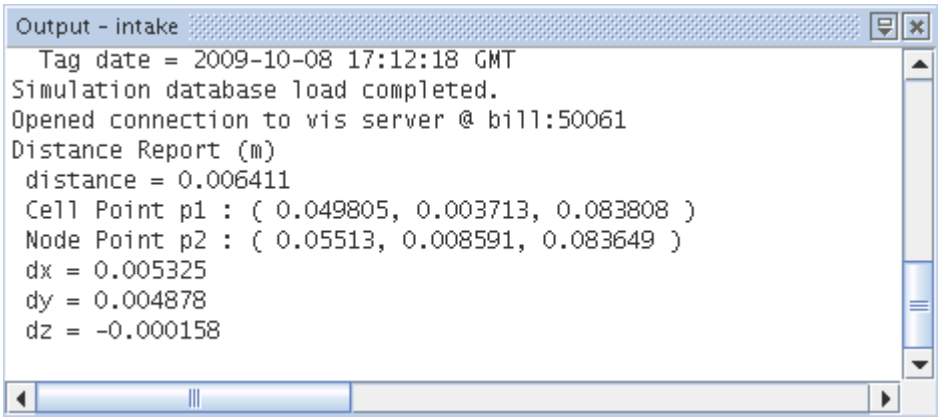
The actual distance for snapping depends on the zoom level.

If you move the mouse anywhere in the scene other than inside a cell or vertex, the distance appears in red. The red color indicates that the reported distance is indeterminate.



Note: Do not use the measuring tool with the **Make Scene Transparent** feature activated. Transparency interferes with depth measurement, possibly resulting in incorrect distance reports.

Detailed information on your measurement also appears in the *Output* window.



In addition to the distance itself, this information includes:

- The coordinates of the end points.
- The type of each end point -- if the point is on a vertex (or mesh node) (`Node Point`), in a cell (`Cell Point`), or off the model (`Scene Point`).
- Distances in the x, y, and z planes.

5. The length bar remains in the display, and the toolbar button for this feature is deactivated, until you do any of the following:
   - Click anywhere in the scene.
   - Click elsewhere in the GUI.
   - Press `<Esc>`.

See Also:

Restoring a View

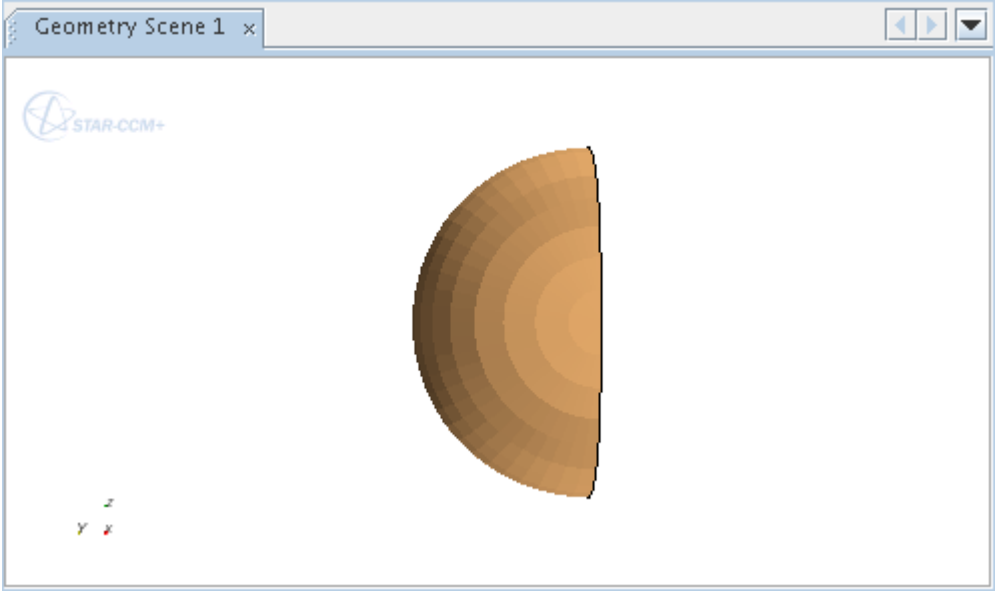
Setting a Predefined View

Changing a Three-Dimensional Visualization Display

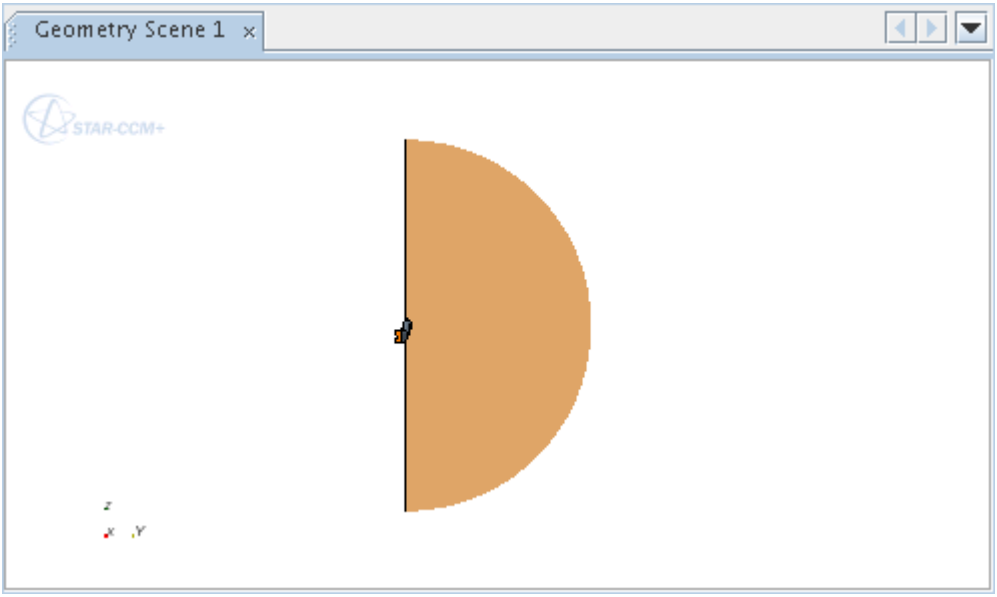
Changing the View

You can change the view in the scene display using keystrokes.

- `<F>` - brings the view to the front of the object.

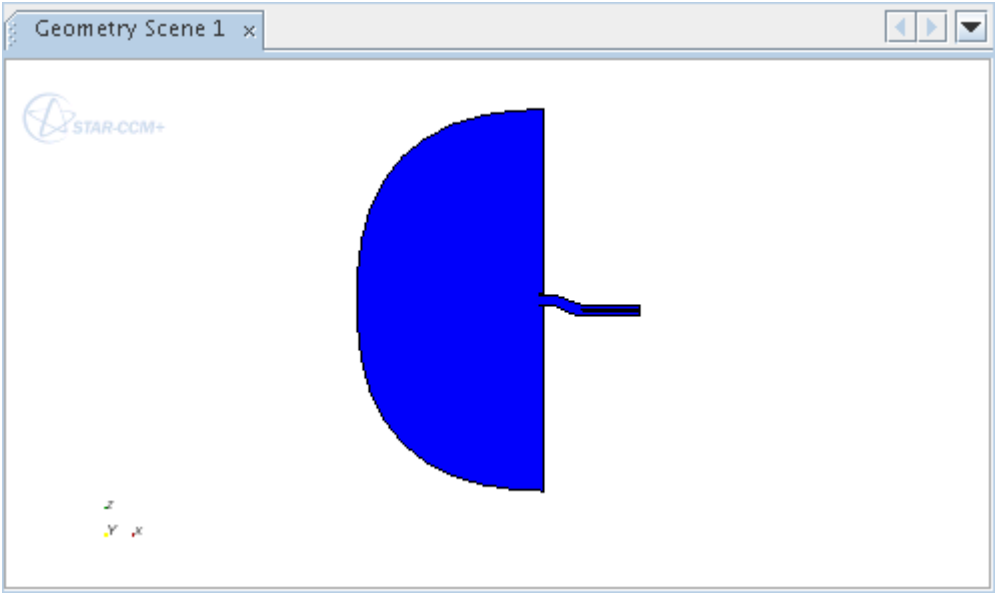


Pressing this key again flips to the back.

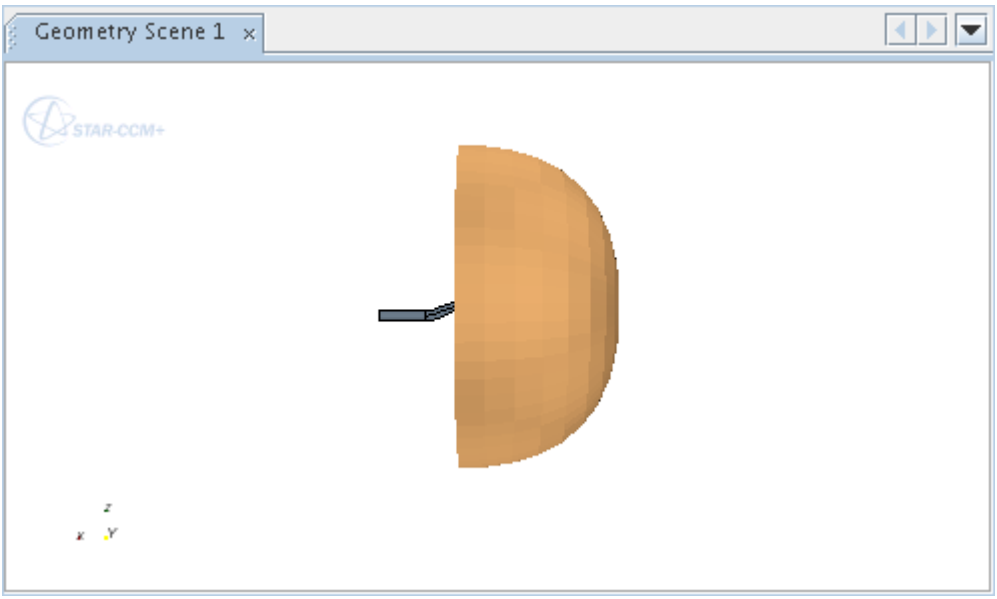


Additional strokes of this key toggle between front and back.

- <S> - brings the view to the side of the object.

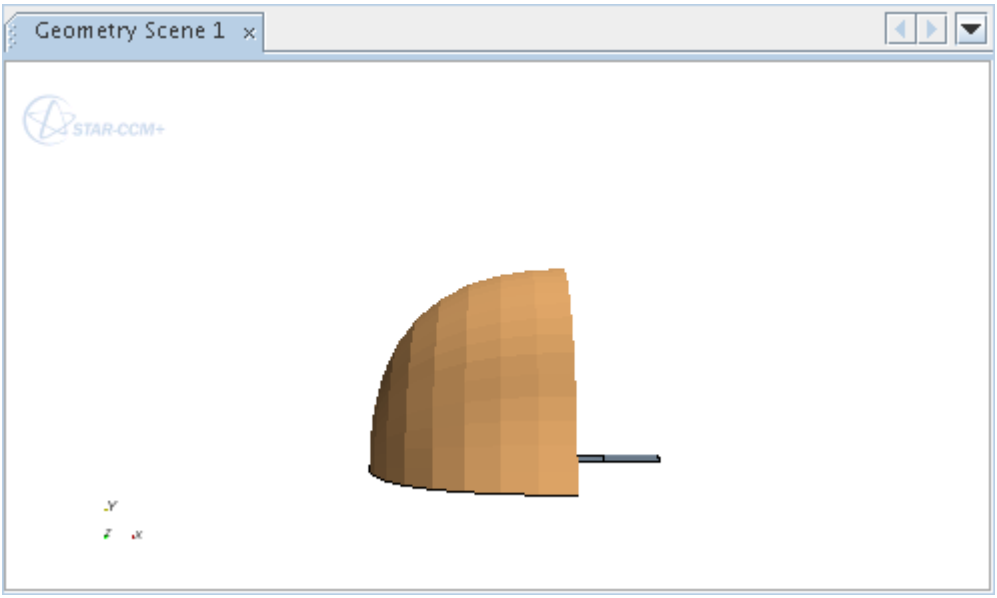


Pressing this key again flips to the other side.

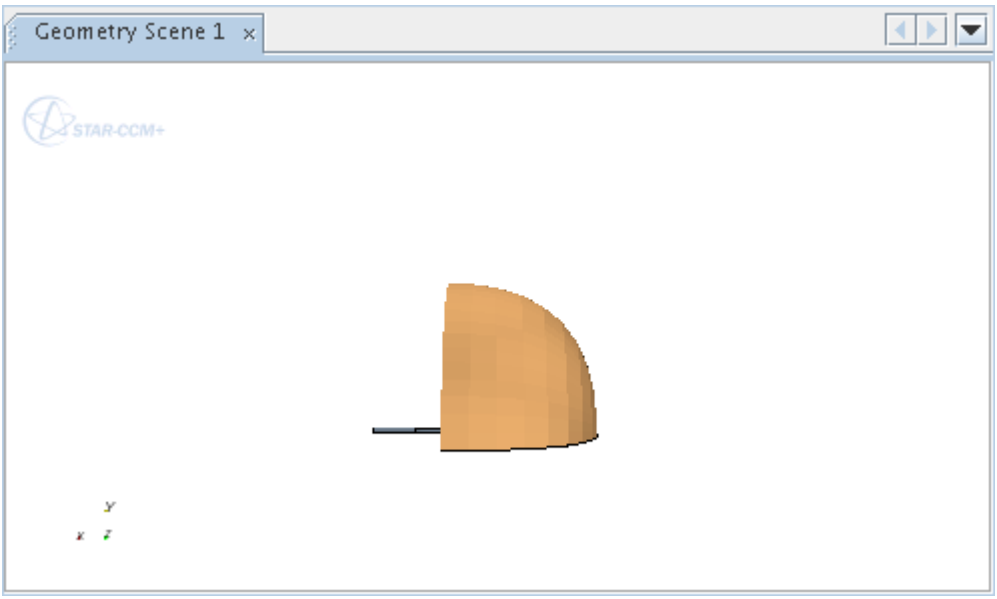


Additional strokes of this key toggle from one side to another.

- Similarly, the <C> key flips the picture horizontally along a vertical axis in the center of the view.
- <T> - brings the view to the top of the object.



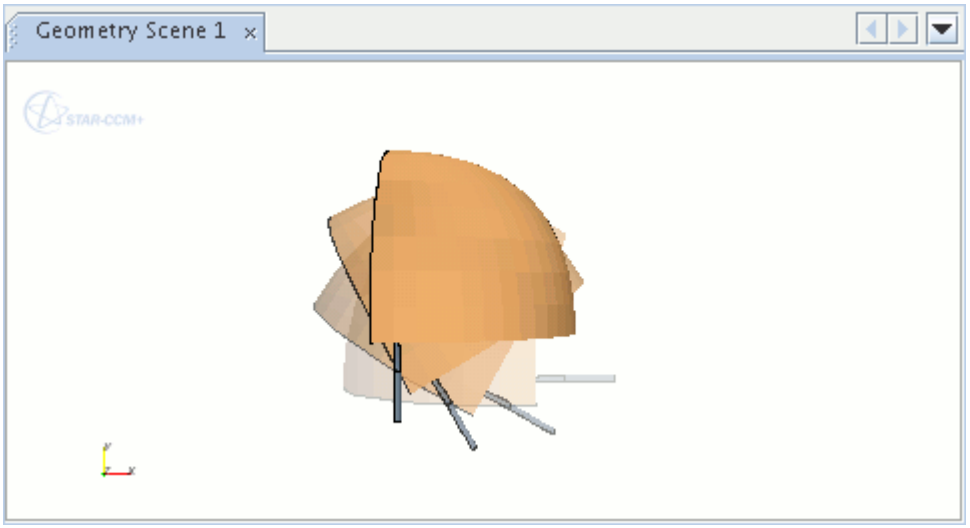
Pressing this key again flips to the bottom.



Additional strokes of this key toggle between top and bottom.

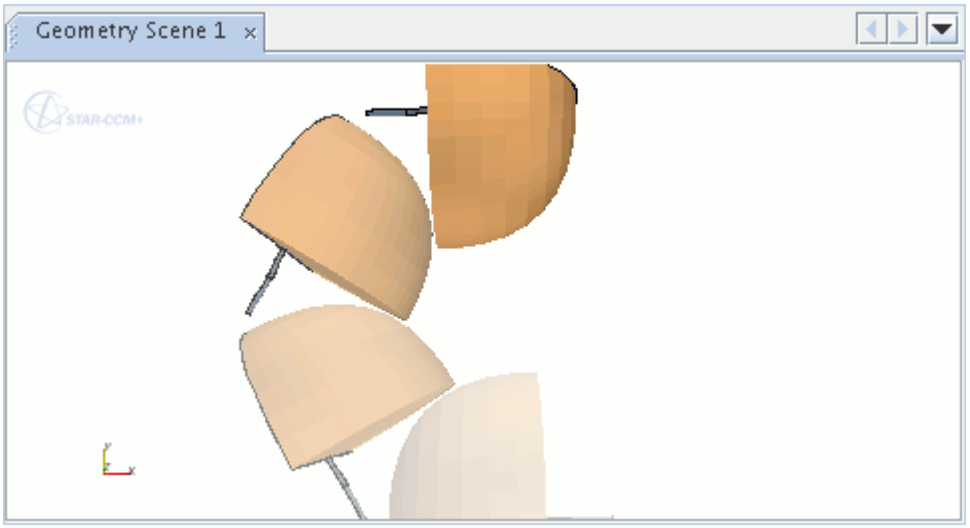
You can use the `<X>`, `<Y>`, and `<Z>` keys to move the image around the corresponding axis.

For example, `<Z>` rotates the object in a negative direction around the z-axis, while `<Shift><Z>` rotates the object in the opposite direction.



Each keystroke turns the image 10 degrees.

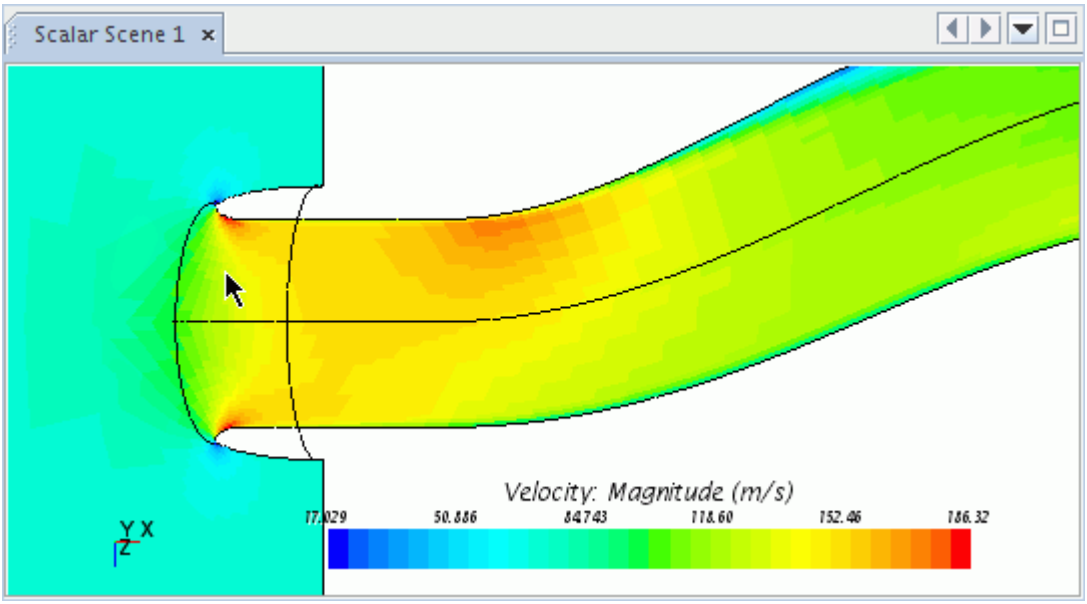
The shifts are in relation to the focal point of the display.



Obtaining Scalar Data with the Cell Probe

You can obtain scalar data for a particular cell in an open scalar scene.

1. To use the cell probe in a scalar scene, click in the display of that scene.
2. To obtain the scalar value for a cell, hover with the pointer over the location of that cell in the scene and press the period (“.”) key.



A report appears in the *Output* window that includes the scalar value and position.

```
Output - intake
Loading/configuring connectivity (old|new partitions: 1|1)
  Fluid : 58800 cells, 171560 faces
Configuring finished
Reading material property database "Z:\install2\win64\STAR-CCM+9.03.080\st
***cell probe
  entity:  Fluid: Symmetry
  position in Laboratory:  [0.005243390661976419, -1.295167441703604E-4, -
  field:  Velocity: Magnitude
  value:  129.204  (m/s)
```

The reported position is based on the coordinate system, which is Laboratory by default. To use a different coordinate system, see Setting the Coordinate System of the View.

If you point outside the mesh (in the background of the scene display), the *Output* window simply shows the message: `No cell picked`.

Note: In other types of scenes, such as geometry, this action only results in the name of the boundary and the position.

Selecting Parts by Type in the Display

STAR-CCM+ provides a toolbar that lets you select objects in a scene with precision.

The selection tools come in three varieties: rubberband, zone, and depth. You can control each one by a filter that lets you select parts by curve, surface, or volume.

Add any volumes, surfaces, or curves that you wish to select using these tools to the scene displayers. For example, when importing CAD files, part curves are not included in the geometry displayer by default, so are not available for selection.

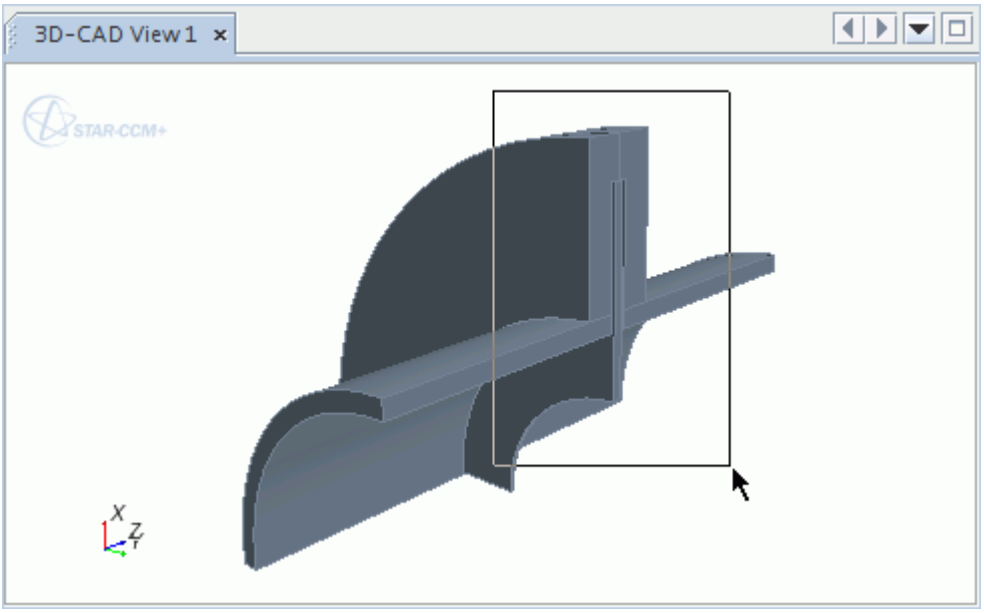
The Depth Select tool lets you select objects that are hidden from view in the scene.

Selecting in a Rubberband

This technique lets you define a specific area in the display by clicking two points that define the opposing corners. Objects within that area are selected. You can use the selection filter to control whether this selection includes volumes, surfaces and/or curves.

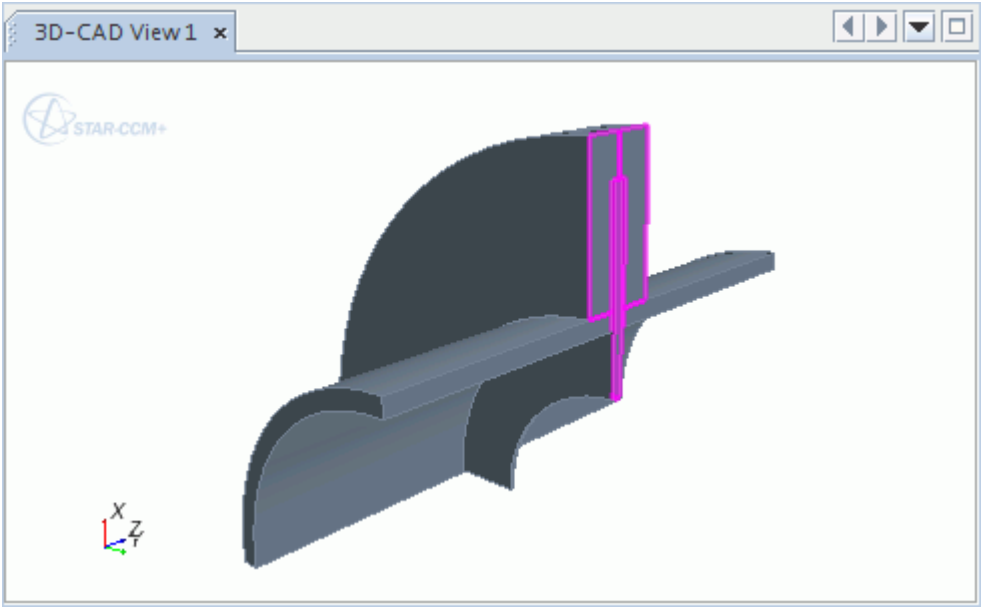
1. Click the (**Rubberband Select**) button to activate this tool.

   After you click the first point in the display, the tool draws a rectangle and finishes it once you click the second point.



2. Press <Esc> to cancel the drawing process.

   The resulting selection is highlighted in the display. In the example that illustrated here, the setting is for curves in a 3D-CAD view scene.

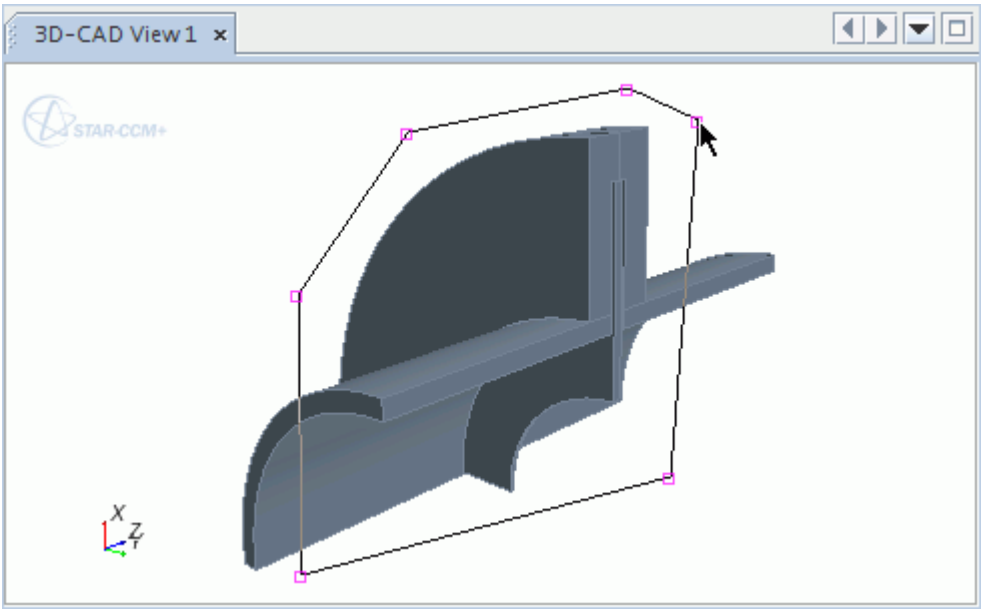


Selecting by Zone

This technique lets you draw a zone by clicking and dragging to create a series of connected points. Objects within that zone are selected.

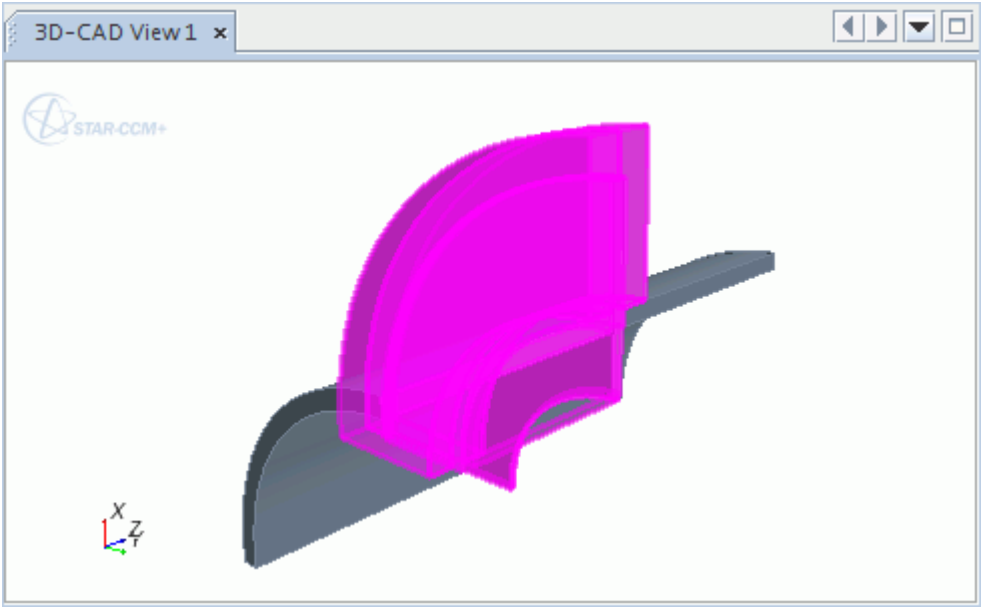
You can use the selection filter to control whether this selection includes volumes, surfaces and/or curves.

1. Click the (**Zone Select**) button to activate this tool. Each click in the display adds a point that is connected by a segment to the previous point.



2. Right-click the last point to finish the zone.
3. Press <Esc> to cancel the drawing process.

The resulting selection is highlighted in the display. In the example that illustrated here, the setting is for surfaces and curves in a 3D-CAD view scene.



Selecting Behind Other Objects

This technique lets you select objects that are obscured from view. It is available in CAD and in regular visualization scenes. To select an object behind other objects:

1. Click the "Select Hidden Object" button in the **Selection** toolbar.
   - To add the hidden object to an existing selection of objects, hold down the `<Ctrl>` key before you click in the scene.
2. Click in the scene at the approximate location of the part that you wish to select.

   This action does the equivalent of "shooting a ray" that is perpendicular to your view. The only objects available for selection are the ones that the ray hits.
3. Use the mouse wheel or arrow keys to iterate through the choice of objects.

   Green highlighting of an object means that the particular part is available for selection. Yellow highlighting means that the object already has been selected previously, and selecting it again would deselect it.

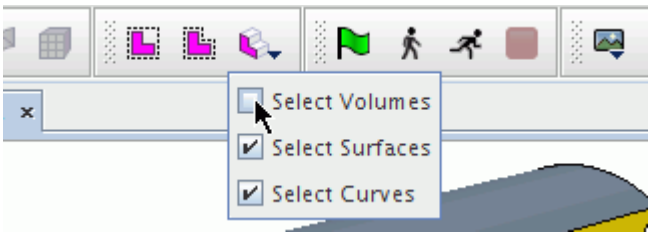
   Note: Rotating the scene is allowed when in iteration mode. This capability lets you check the order of the results along the ray.
4. To select the object, press the `<Space>` bar.
   - To add objects to your selection, hold down the `<Ctrl>` key while shooting another ray.

Filtering the Selections

If you have a wide variety of parts in the object being simulated, you can filter the selection based on curve, surface, or volume.

The result will be that, as you carry out mouse actions in the display, only the type of part you specify is selected.

Click the (**Selection Filter**) button. It activates a menu.



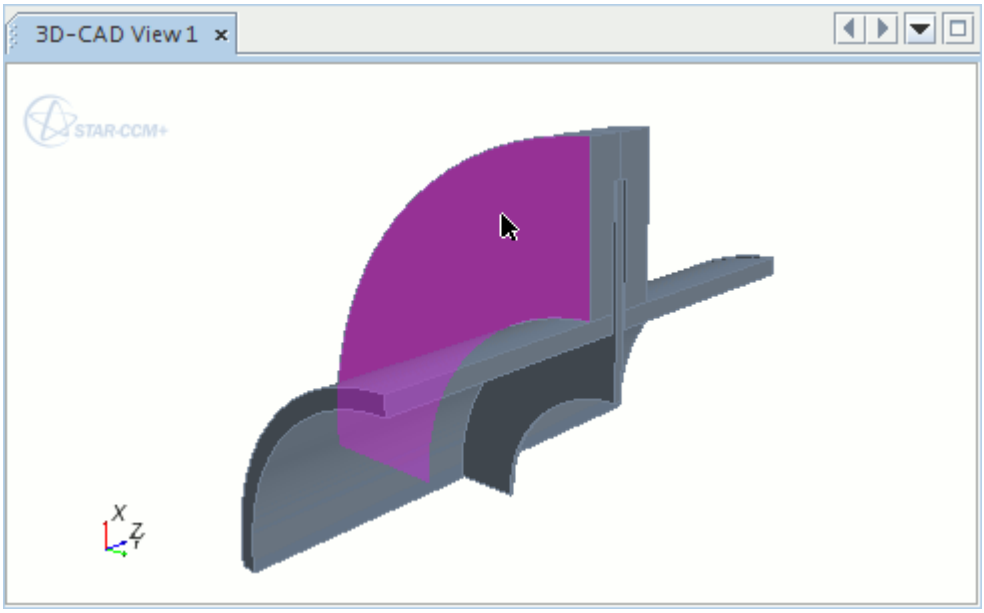
By ticking the checkbox next to an option, you can specify the type of part selected. Since selecting volumes always includes the selection of curves and surfaces, the options in the menu change their selection automatically depending on which checkbox you tick, providing the following set of possibilities:

- Volumes
- Surfaces
- Curves
- Surfaces and curves

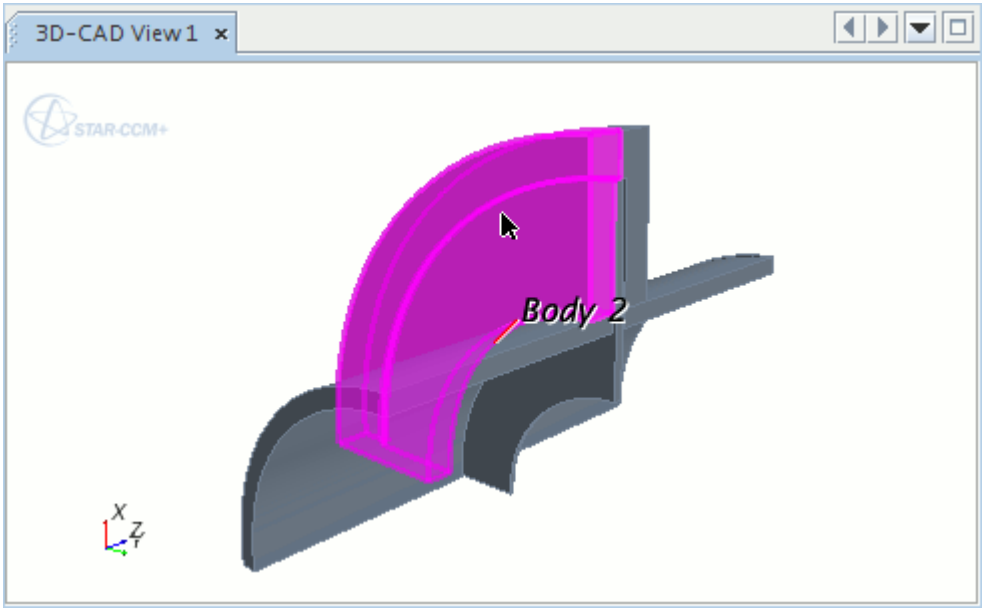
The meaning of these terms for the specific part type varies depending on what type of display you are accessing:

- For 3D-CAD views:
  - “Volumes” are 3D-CAD bodies.
  - “Surfaces” are the faces of the 3D-CAD bodies.
  - “Curves” are the edges of the 3D-CAD faces.
- For geometry views displaying geometry parts:
  - “Volumes” are geometry parts.
  - “Surfaces” are part surfaces.
  - “Curves” are part curves.
- For views displaying regions and boundaries:
  - “Volumes” are regions.
  - “Surfaces” are boundaries.
  - “Curves” are feature curves.

The filter also controls how the left-click selects objects. For example, if it is set to select surfaces only, a click on a part selects only the surface.



However, if selection of volumes is specified, then a click in the exact same place selects the volume.



See Also:

Using Scenes to Interact with the Object Tree

Changing a Three-Dimensional Visualization Display

This section explains various methods of changing the three-dimensional visualization display.

The following operations let you manipulate a three-dimensional visualization display:

- Shifting the image by panning
- Moving the part in the visualization display around a specific axis by rotating and rolling

  (You can also rotate the image around the x, y, or z-axis using the keyboard.)
- Getting a close-up view by zooming
- Aligning the view to a part normal

It is also possible to reset the scene to its original view.

These techniques are based on the default interactor style. Within STAR-CCM+, it is possible to change the mouse configuration to suit your preferred working environment. Available styles include Pro-Engineer/NX and SolidWorks. It is also possible to specify joystick, trackball, and 3Dconnexion interactor styles.

Panning

To pan the image in the visualization display, hold down the right mouse button and drag the mouse in the direction where you want to move the object.

The image appears to change its view angle as you pan. This is because the scene's view is in the `Perspective` projection mode, which depicts three-dimensional space. In the `Parallel` projection mode, the view angle stays the same.

You can achieve the same effect with the middle mouse button if you hold down the `<Shift>` key at the same time.

Rotating

STAR-CCM+ uses a point on a part to rotate. If you select a point somewhere other than on a part, STAR-CCM+ uses the depth to that last point; the depth is normal to the view plane.

The scene can be rotated around various axes, vertical, horizontal, or arbitrary, depending on the direction you move the mouse.

To rotate, choose a point on or near the object in the visualization display, and click and drag (holding down the left mouse button) from that point.

This technique rotates the object around the axis perpendicular to the direction of your drag. You set the location of the axis when you choose a point to begin clicking and dragging.

Rolling

To roll the scene, hold down the `<Ctrl>` key, and click and drag the mouse upward or downward.

Using the `<Ctrl>` key rotates the object along an axis perpendicular to your screen.

If you try clicking and dragging left or right in this mode, there is no effect.

Zooming

There are different ways to zoom with the mouse:

Choosing a focal point

Choose a point inside the display, then click and drag downward using the middle mouse button.

This action brings your view closer to that point.

Moving the mouse straight up has the opposite effect.

Rubber banding

To zoom to a specific area in the display, click the (Rubberband Zoom) button, and then click two points that define the opposing corners of the zoomed area.

Using the mouse wheel

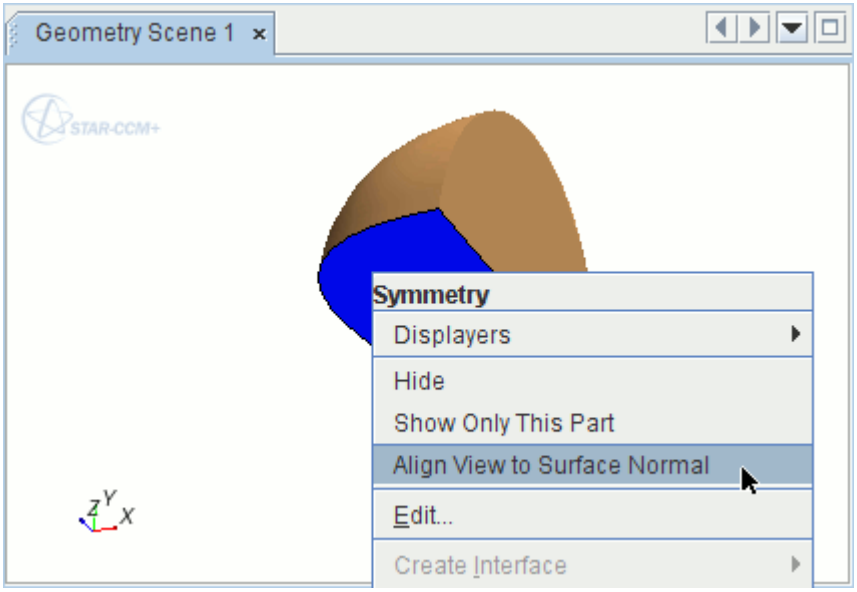
Turning the mouse wheel toward you brings your view closer to the visualization display.

The opposite wheel movement takes your view farther away.

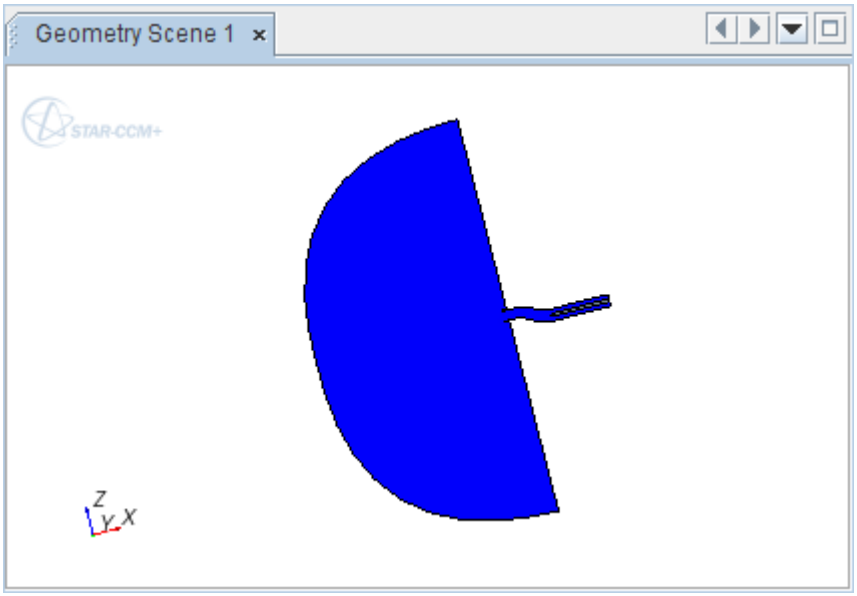
The mouse pointer must be within the visualization display for this operation to work. Its particular position within the display determines the direction of this type of zooming.

Aligning to a Normal

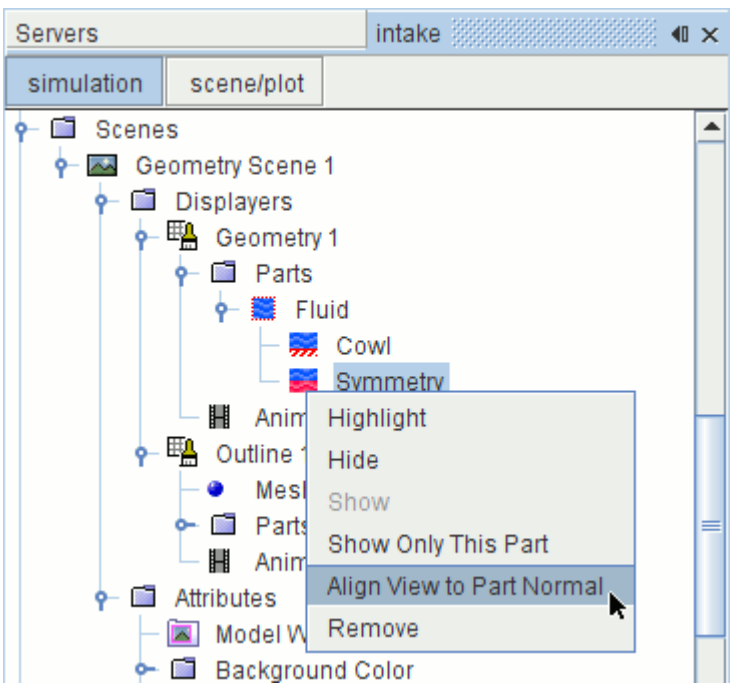
To align the view to the surface normal of a part, right-click on the part in the scene display and select **Align View to Surface Normal**.



The view aligns accordingly.



Similar functionality exists in the pop-up menu of the scene's part node: the **Align View to Part Normal** menu item.



Resetting

To restore the original view of the scene, click the (**Reset**) button in the toolbar.

The <R> key behaves the same as the (**Reset View**) toolbar button.

See Also:

Using Scenes to Interact with the Object Tree
Using Scenes to Interact with the Object Tree
Using the Distance Measuring Tool
Using the Mouse in the Visualization Display
Changing a Two-Dimensional Visualization Display

Changing a Two-Dimensional Visualization Display

This section explains various methods of changing the two-dimensional visualization display.

A two-dimensional mesh appears in the visualization display as perfectly flat; therefore the following display operations differ somewhat from the display options for three-dimensional displays:

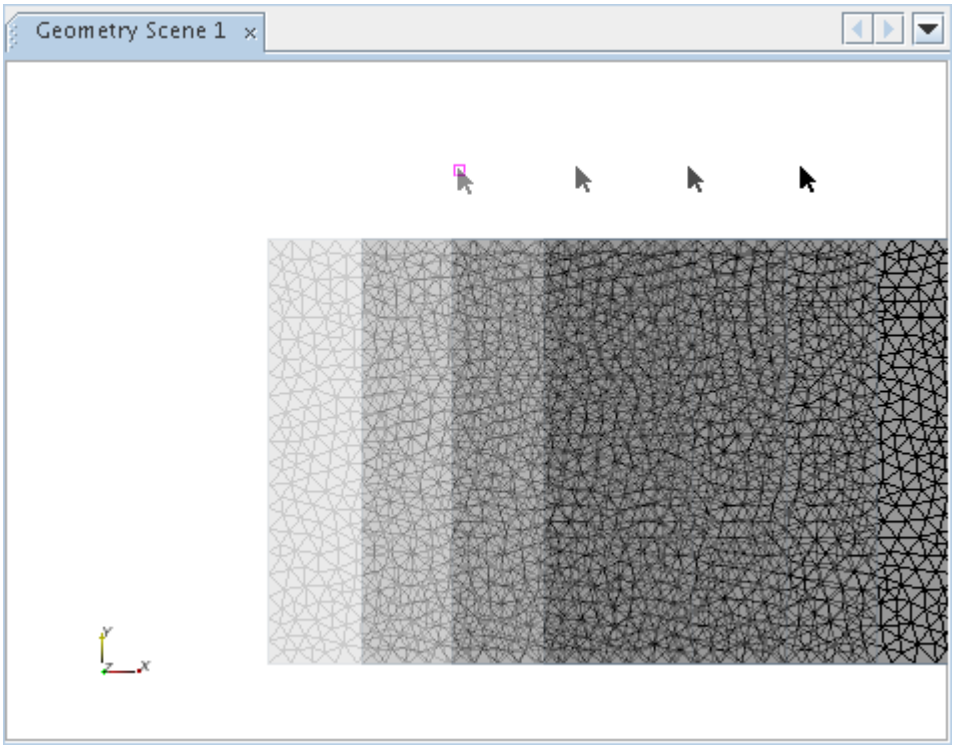
- Panning
- Rotating
- Zooming

Resetting the scene to its original view can be done the same as for a three-dimensional display.

These techniques are based on the default interactor style. Within STAR-CCM+, it is possible to change the mouse configuration to suit your preferred working environment. Available styles include Pro-Engineer/NX and SolidWorks. It is also possible to specify joystick, trackball, and 3Dconnexion interactor styles. See Mouse Configurations.

Panning

Panning a two-dimensional image is similar to doing so with a three-dimensional one: holding down the left mouse button and dragging the mouse in the direction where you want to move the mesh.



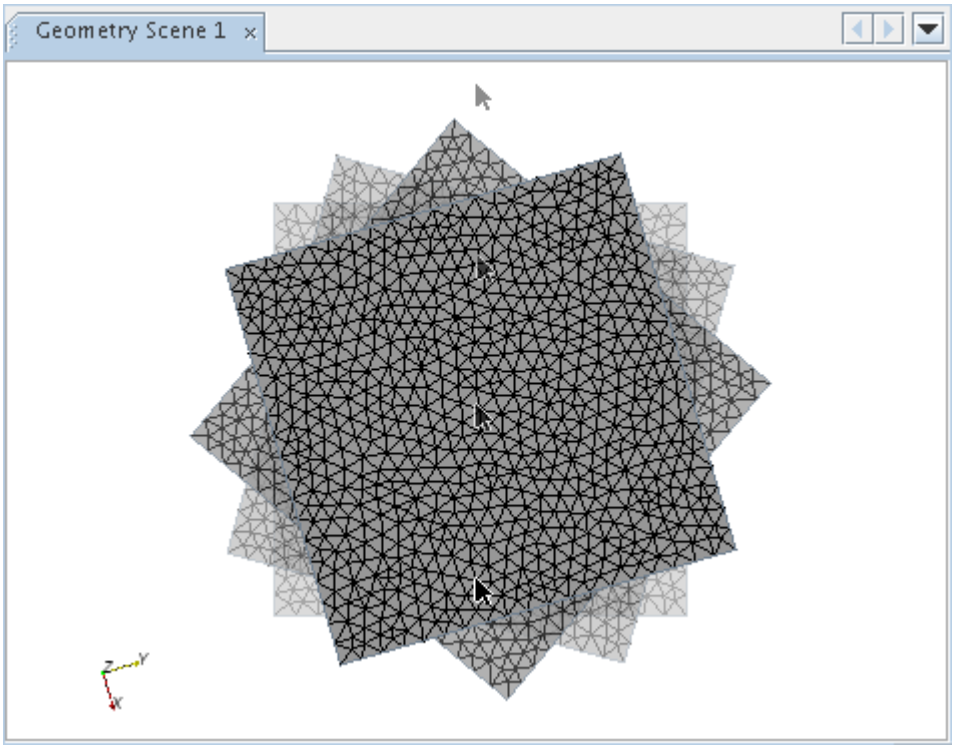
However, there are two important differences:

- The right mouse button is also available for panning, because no rotation is possible.
- There is no change in perspective as you pan, since there is no third dimension in the view.

Rotating

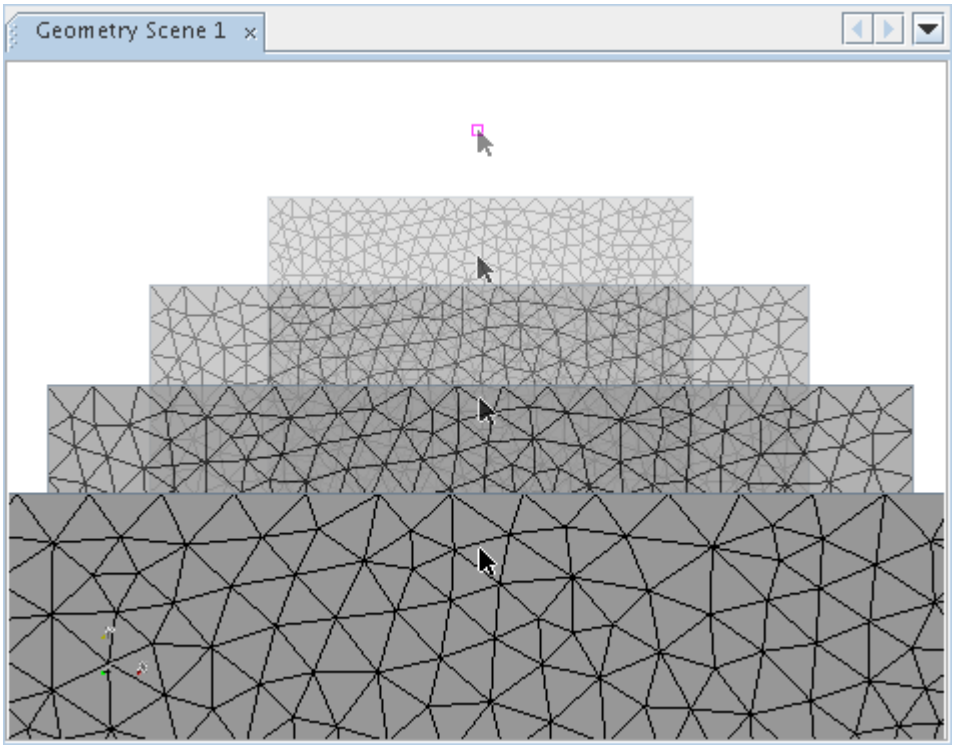
A two-dimensional mesh cannot be rotated, since it has no third dimension to display. However, it is possible to roll it.

To roll a two-dimensional mesh, hold down the `<Ctrl>` key, and click and drag the mouse toward you (downward).



Zooming

The zooming functionality matches that for a three-dimensional mesh.



See Also:

Changing a Three-Dimensional Visualization Display

Using the Mouse in the Visualization Display

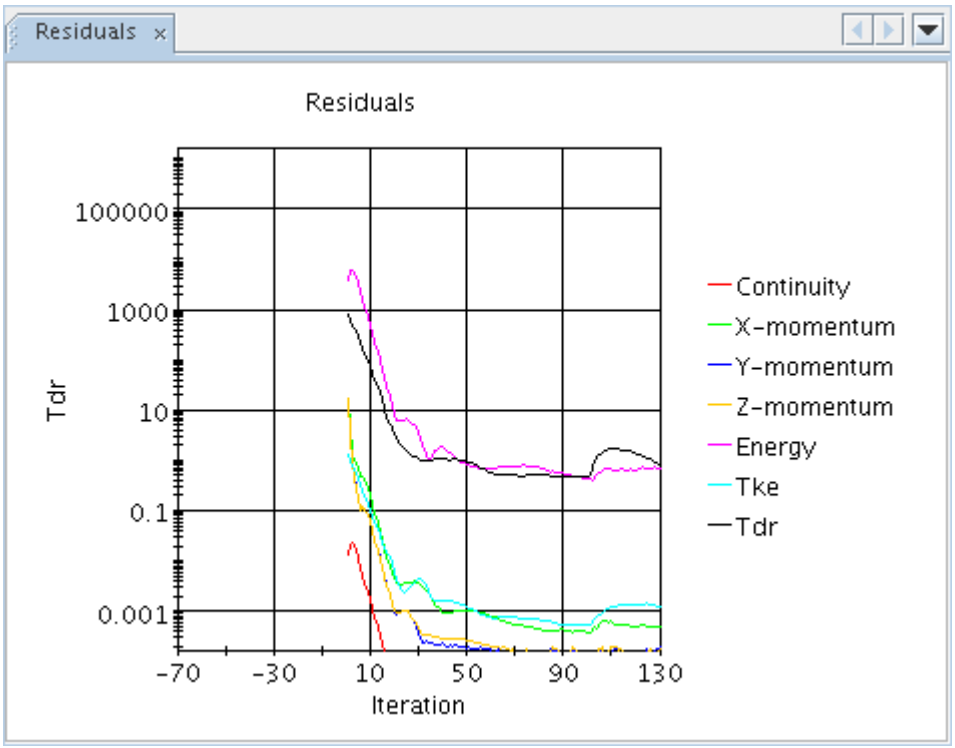
Interacting with the Plot Display

This section explains the features of the plot display.

The typical plot display contains a wealth of information. Therefore, it can be difficult to read all of it from the default appearance. That is why the plot display in STAR-CCM+ includes panning and zooming tools to navigate within the plot, as well as a feature for resetting the original plot view. In addition, you can navigate the object tree to plot objects. You can also change the plot characteristics by right-clicking.

Panning and Zooming

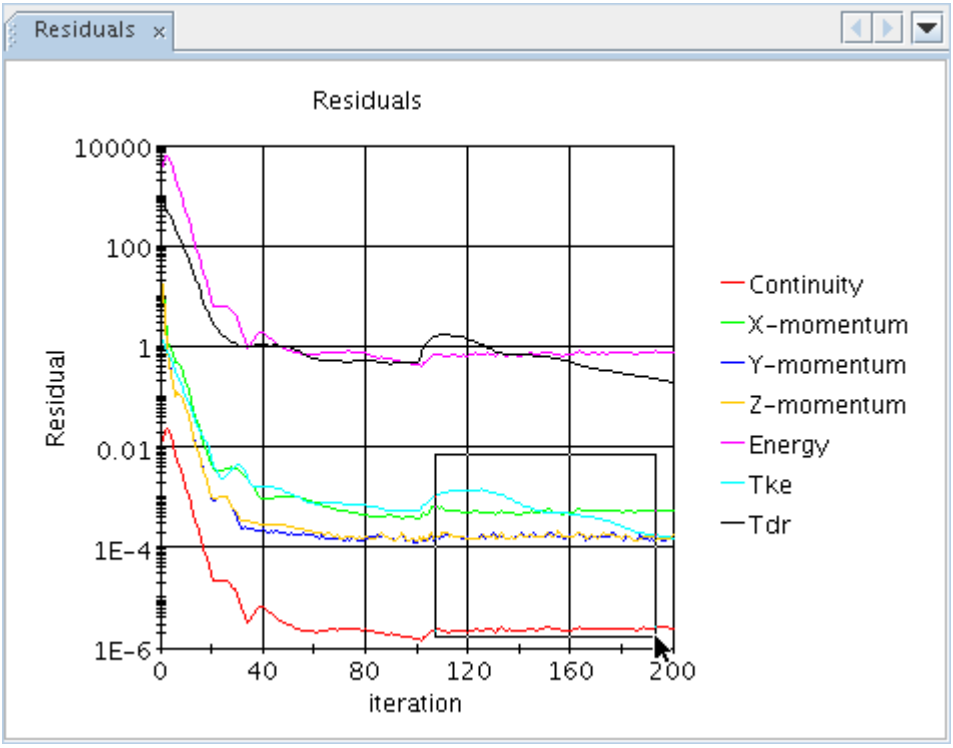
By clicking the (**Pan**) button in the toolbar, you allow the mouse to shift the plot within the confines of the axes. When you release the mouse, the values of the axes shift in response to your mouse movement.



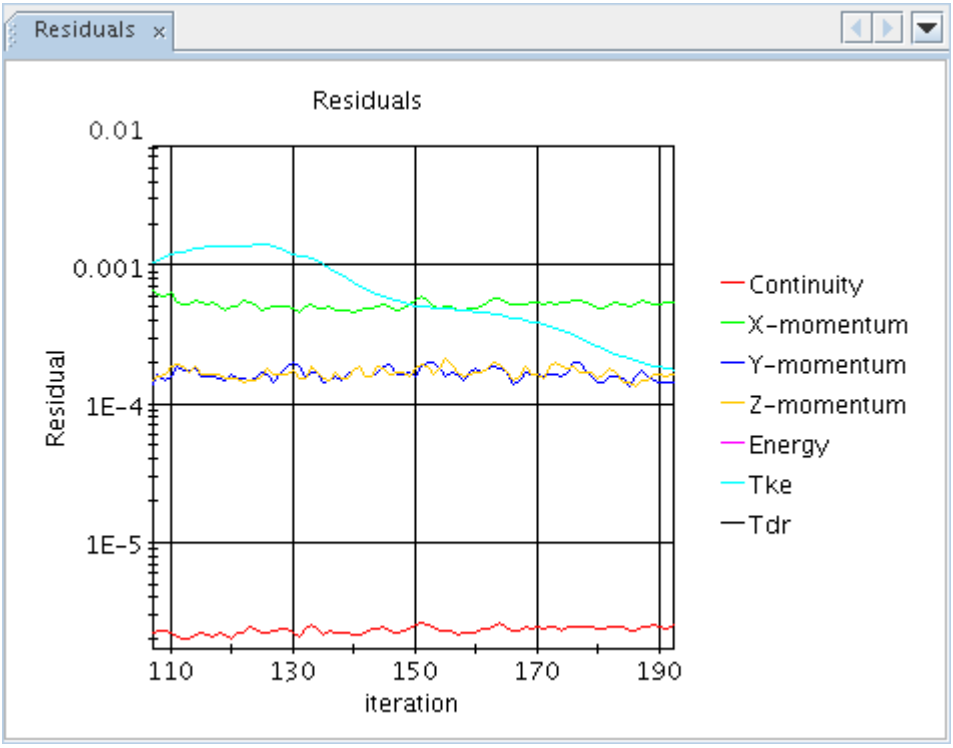
However, the pan feature is especially useful when you zoom in on (magnify) a portion of the plot to look at smaller changes in values, and then pan to examine adjacent portions.

1. Click the (**Zoom**) button in the toolbar.

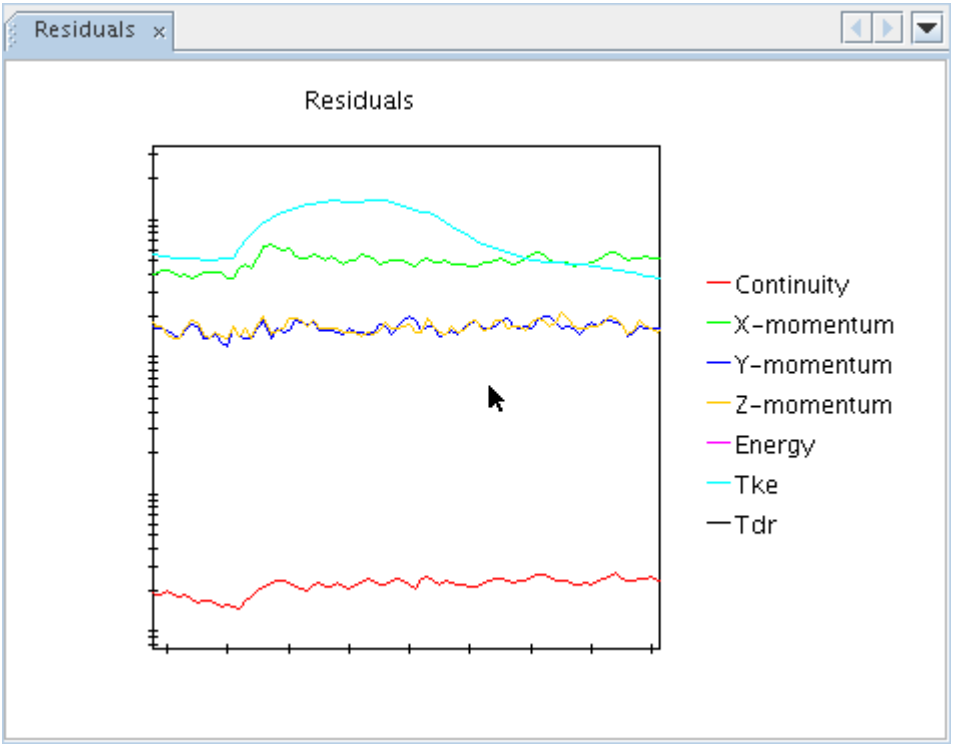
   This action activates the plot zooming feature, which operates on the rubber-banding principle.
2. Drag the mouse through the area of the plot that you want to magnify. This will show a quadrangle expanding like a rubber band.



That quadrangle bounds the zoomed area.



3. Click the **Pan** button again. As you move the plot with the mouse, you get a close-up view of other parts of the plot.

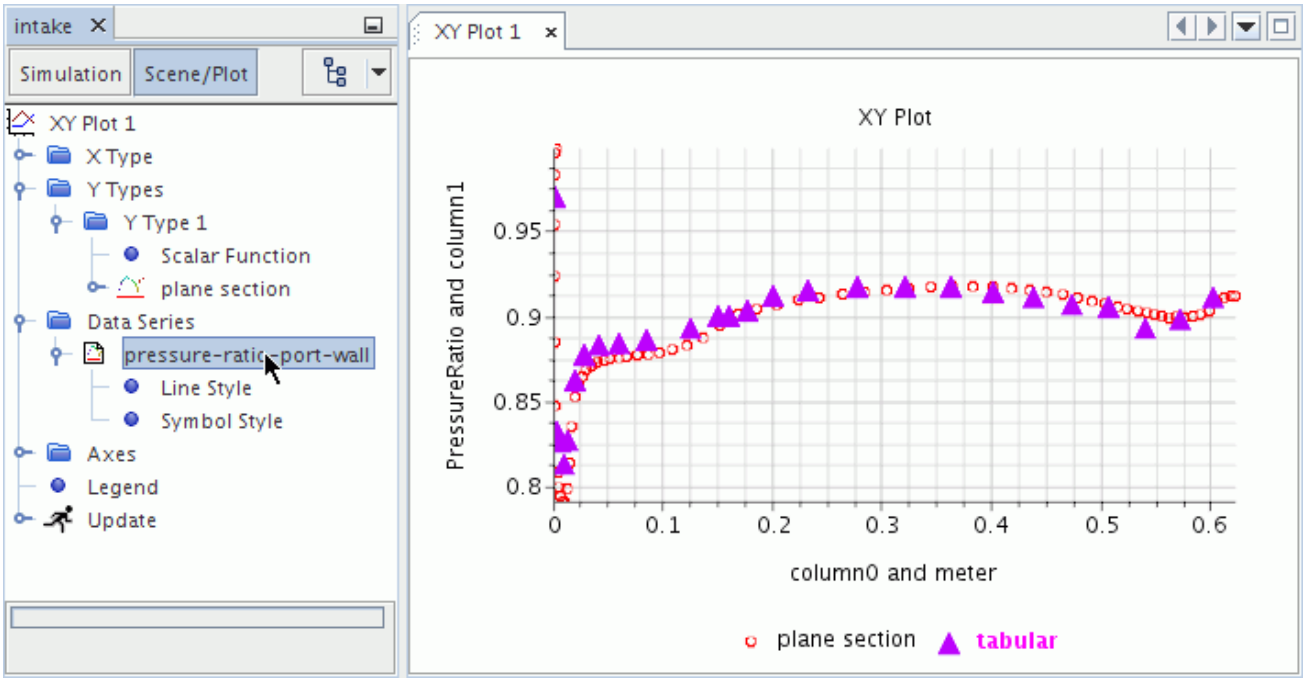


To restore the original view of the plot, click the (**Fit**) button in the toolbar.

Highlighting Objects

Certain plot objects are highlighted when the *Show Highlights* property of the plot is activated (by default).

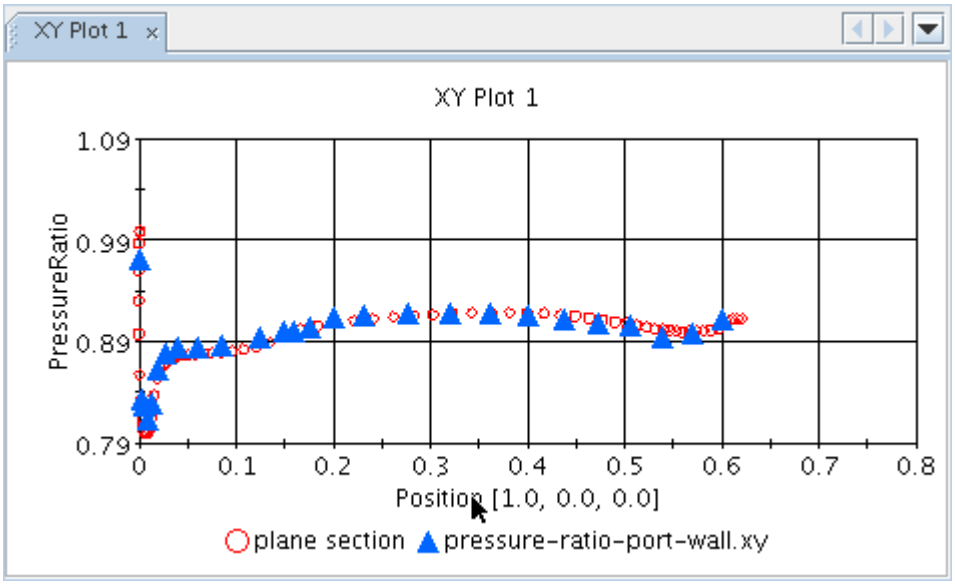
When you select a plot object, such as a data series, axis or legend, in the object tree or the plot display, it is highlighted in the display.



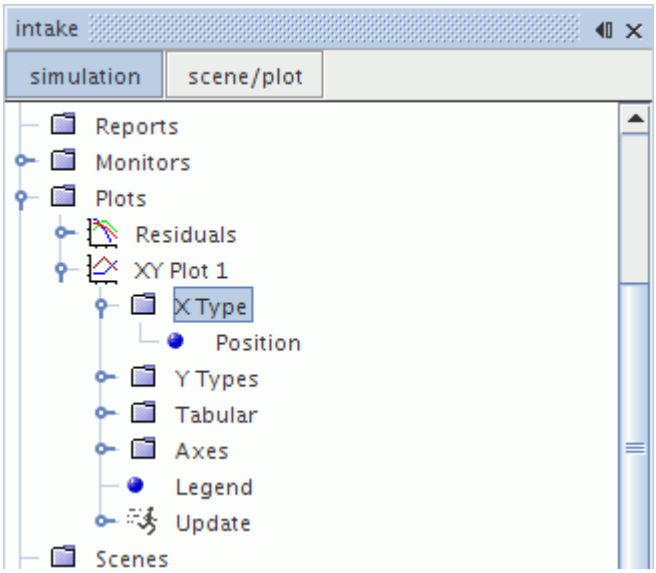
In addition, if you select an input object in the tree, such as a monitor or a table, the corresponding data series is also highlighted in the plot.

Navigating the Object Tree

To select a variable type in a plot, click its label in the plot display.



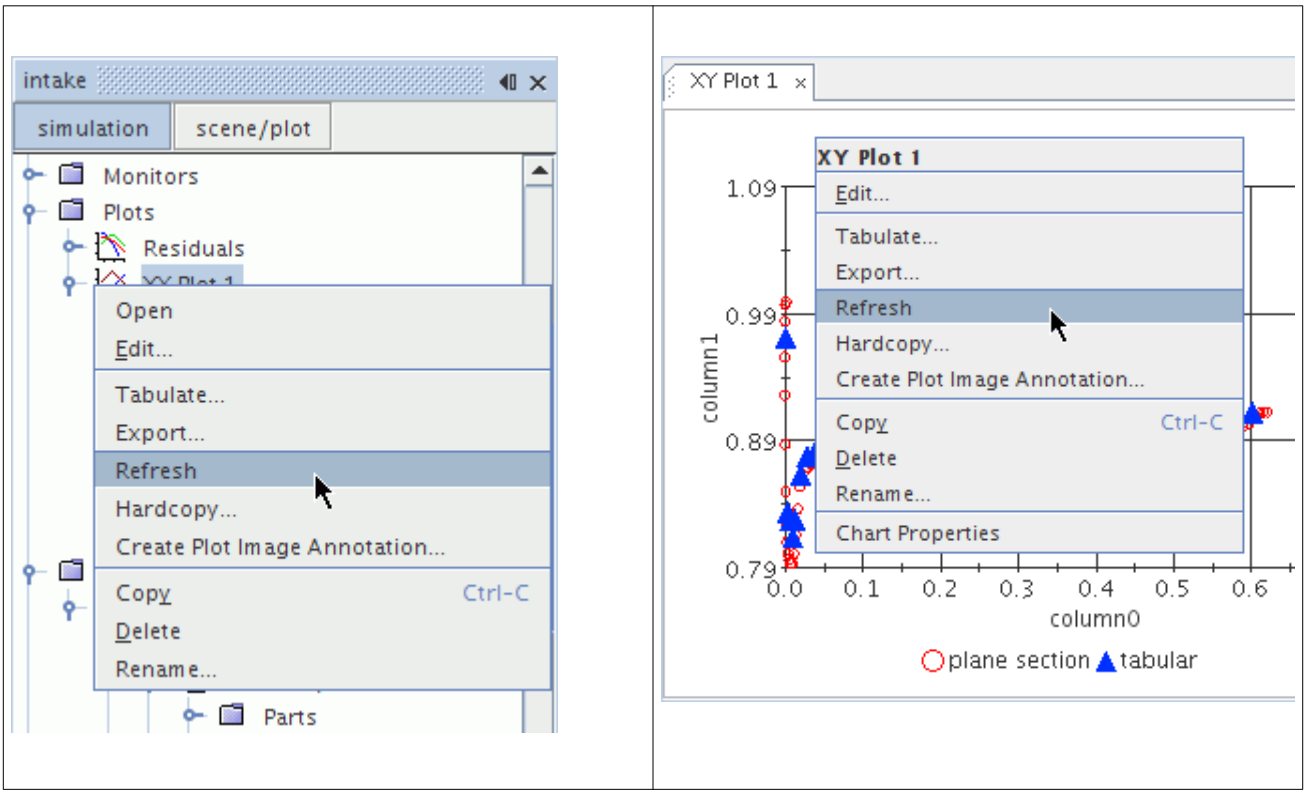
The tree navigates to the corresponding node.



For XY plots, this action takes you to the **X Type** or **Y Type** nodes. For other plots, it navigates to the corresponding node under the **Axes** node.

Right-Clicking

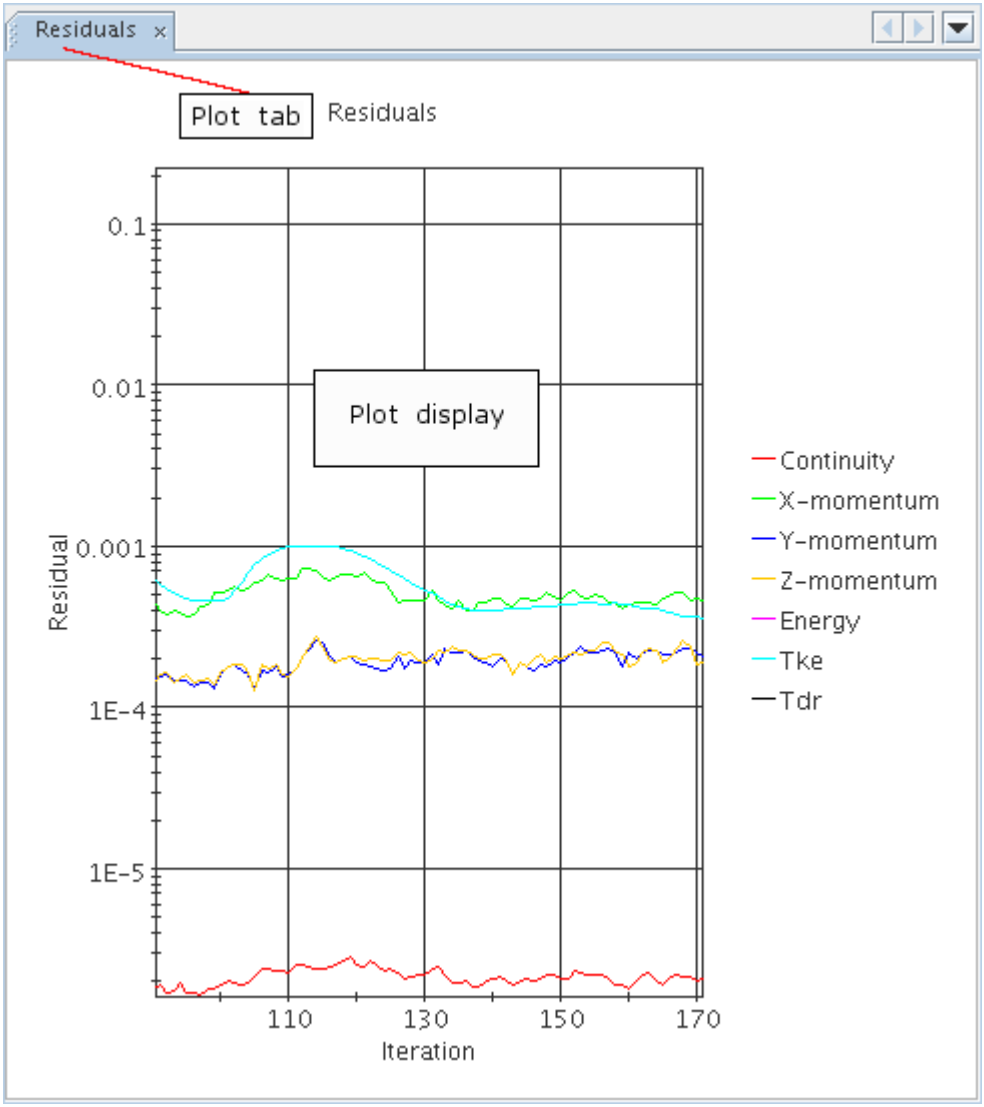
If you right-click inside the plot display, you get a pop-up menu almost identical to that of the node associated with that display.



Plot Tab Reference

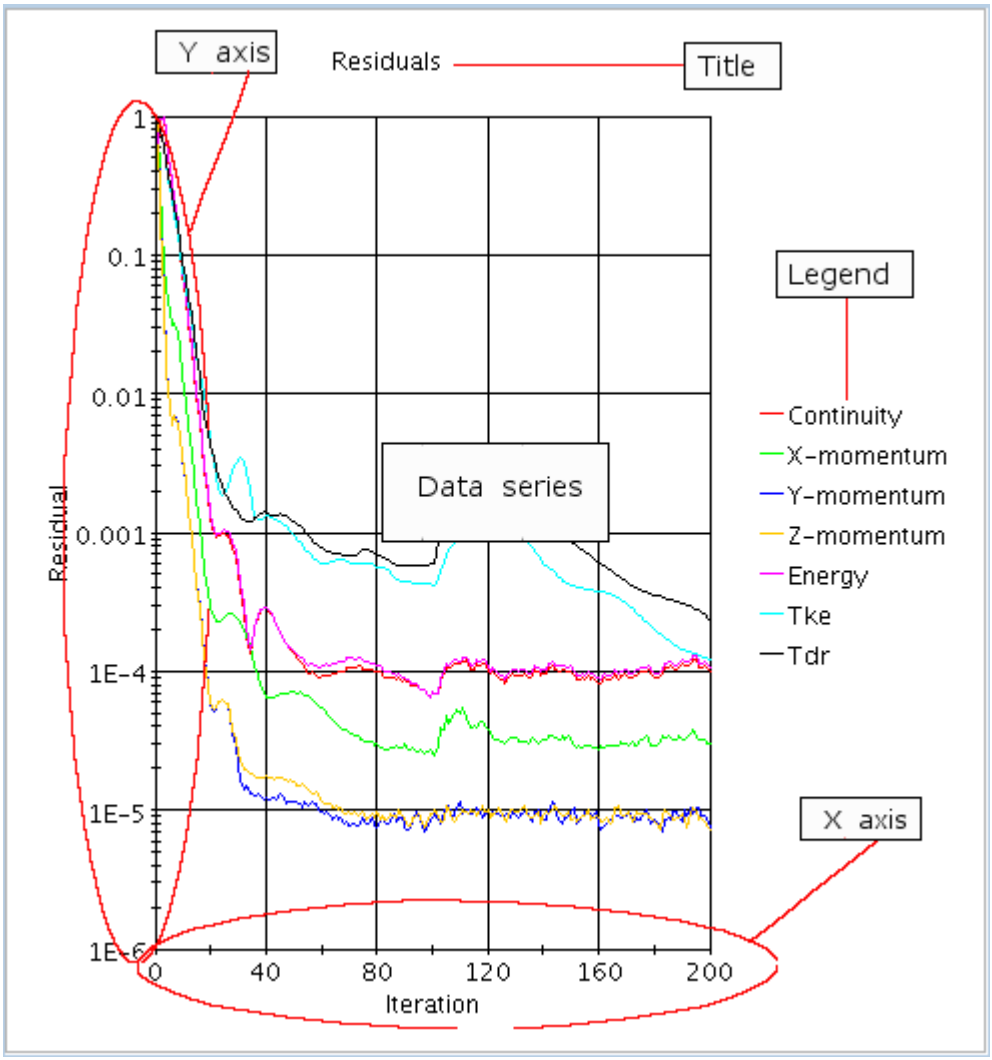
A plot tab consists of a plot display. The tab provides a representation of a plot node in the simulation tree and is named the same as the node.

You can create a plot tab by double-clicking the corresponding node in the tree. If the tab for a particular plot has been created, but is not active, you can double-click the relevant simulation tree node or click its tab in the *Graphics* window.



Plot Display

The key elements of a plot display are shown below.



You can modify these elements (series, X and Y axes, legend, symbols and line styles) by customizing their common properties.

See Also:

Working with Displays

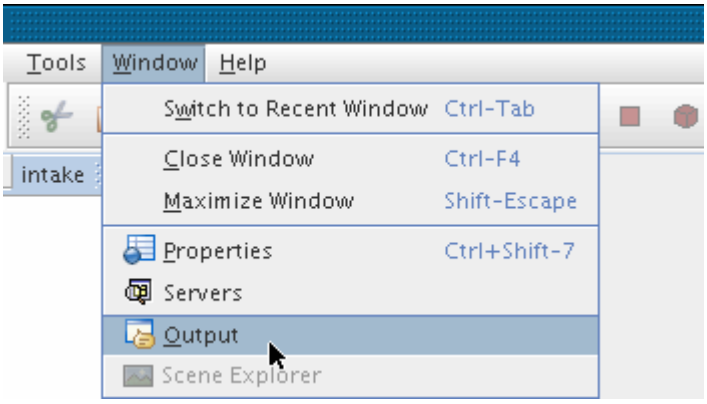
Customizing Plots

Using the Output Window

The *Output* window gives a precise text description of all developments within the simulation.

This output can be saved to a file, and handled like a text document in other ways.

This window is open in the STAR-CCM+ workspace by default. However, to open the *Output* window, select **Window** > **Output**.



The *Output* window can have multiple tabs open at once, for example if you have multiple simulations open. To close all tabs of the *Output* window at once, right-click any of the tabs and select **Close All Tabs**.

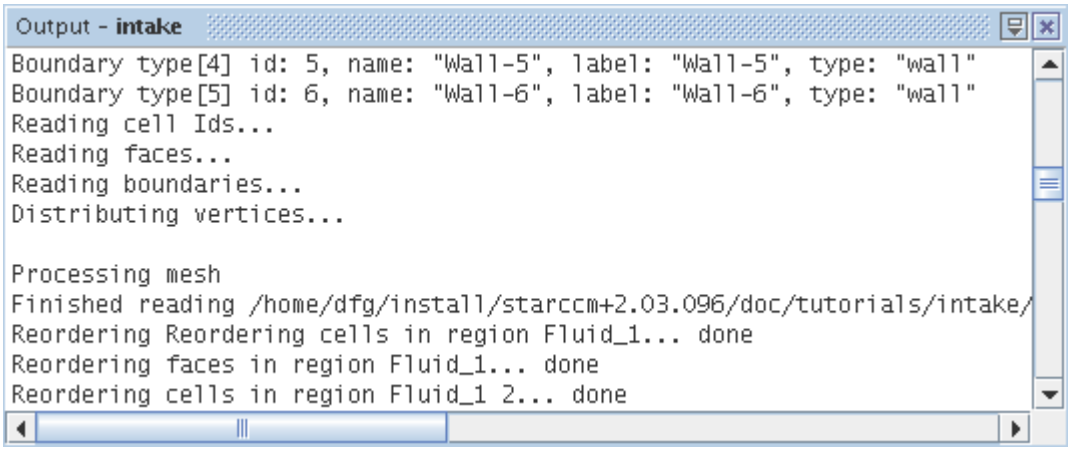
The illustration below shows the output during the loading of an existing simulation.

```
Output - intake
Starting local server: /home/dfg/install/starccm+2.03.096/star/bin/starcc

STAR-CCM+ Version 2.03.096 (linux-x86-2.2.5/gnu4.1)
License version 14-dec-2004
Checking license file: /usr/local/STAR-CD/v3150/license/license.dat
1 copies of prostar checked out from /usr/local/STAR-CD/v3150/license/lic
Feature prostar expires in 96 days

Server::start zx6r:56026
Loading simulation database: /home/dfg/install/starccm+2.03.096/doc/tutor
Loading module: CoupledFlowModel
Loading module: KeTurbModel
Loading module: MaterialModel
Loading module: MotionModel
Database saved by:
  STAR-CCM+ Version 2.03.072 (linux-x86_64-2.2.5/gnu4.1)
  Tag date = 2006-08-21 13:35:18 GMT
Loading into:
  STAR-CCM+ Version 2.03.096 (linux-x86-2.2.5/gnu4.1) Serial
  Tag date = 2006-09-22 23:16:59 GMT

Number of partitions stored in database: 1
Simulation database load completed.
```

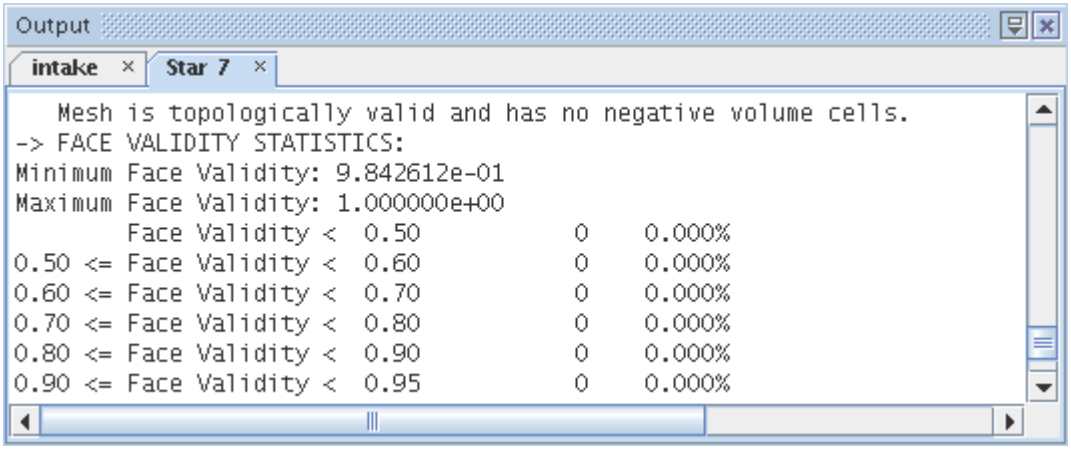
The example below is of running a simulation 1000 iterations.

```
Output - intake
    995   1.276528e-02 7.746027e+00 4.657106e-01 4.458370e-01 3.843451e+0
    996   1.275689e-02 7.750090e+00 4.712672e-01 4.377354e-01 3.840665e+0
    997   1.276532e-02 7.746091e+00 4.656880e-01 4.458260e-01 3.843463e+0
    998   1.275709e-02 7.750113e+00 4.712879e-01 4.377322e-01 3.840719e+0
    999   1.276538e-02 7.746105e+00 4.657168e-01 4.458107e-01 3.843474e+0
   1000   1.275703e-02 7.750091e+00 4.712908e-01 4.377354e-01 3.840699e+0
Stopping criterion Maximum Steps satisfied.
```

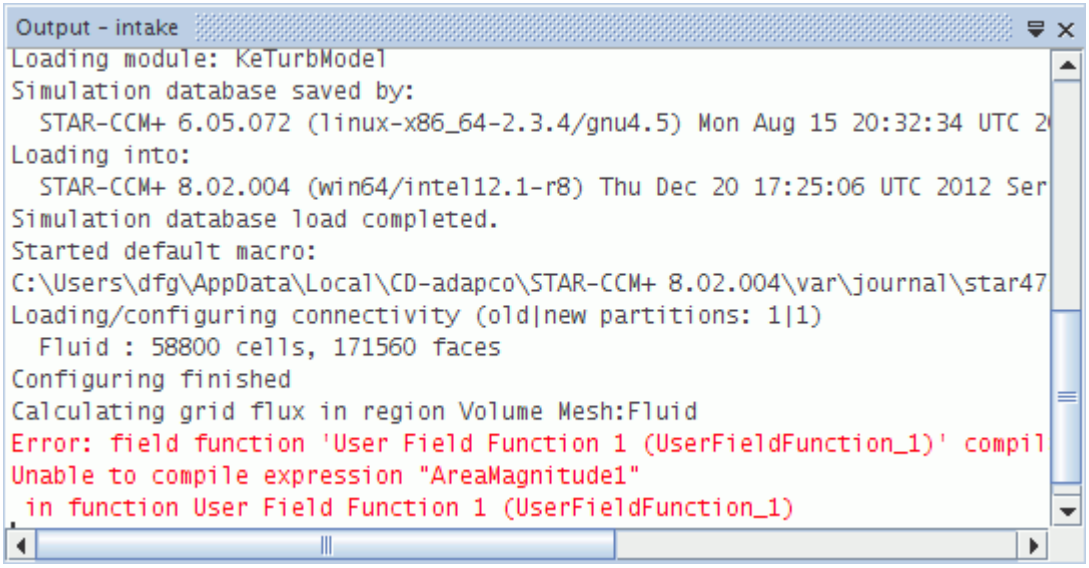
The following is an example of importing a mesh.



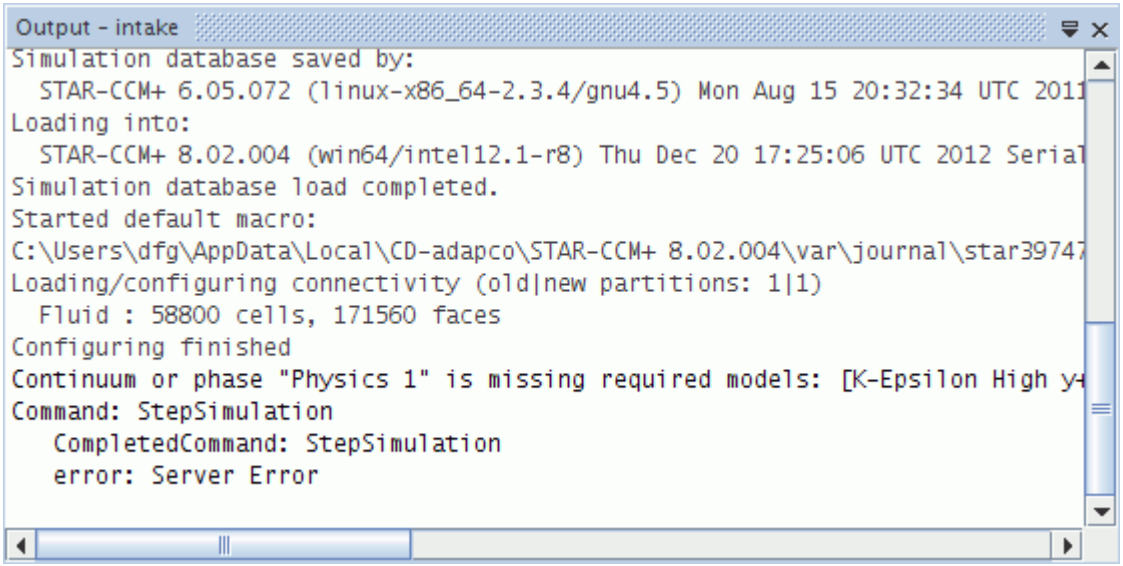
The illustration below shows the *Output* window with separate tabs, each with output for an open simulation.



The *Output* window displays errors in red text, as shown in the following screenshot.



The following example is of a warning, with text colored in black rather than the usual gray.



Contents:

Saving the Output to a File
Working with Output Text

Saving the Output to a File

You can save the contents of the *Output* window to an external file.

The automated saving approach retains the entire session's data in a `.log` file. Alternatively, you can right-click in the *Output* window to save the contents as a standard text (`.txt`) file in a location of your choice, but such a file would only retain the current contents of the *Output* window.

Contents:

Automatically Retaining the Output of the Entire Session
Saving the Current Output as a Text File

Automatically Retaining the Output of the Entire Session

You can save the content of the *Output* window to a file.

In the *Options* dialog, select the **STAR-CCM+** > **Environment** node. Make sure that the *Log output window to file* property has its checkbox ticked.

This file will be named `output****.log`, with "`****`" representing a series of numeric characters. It can be found in the following locations:

- On Linux/Unix: `~/.star-X.XX.XXX/var/log/`, where ~ is your `home` directory and `X.XX.XXX` is the version of STAR-CCM+, for example 9.04.006
- On Windows: `C:\Users\username\AppData\Local\CD-adapco\STAR-CCM+ X.XX.XXX\var\log\`

Once you access this file, be sure to save a copy of it in a location where you keep data related to the simulation project.

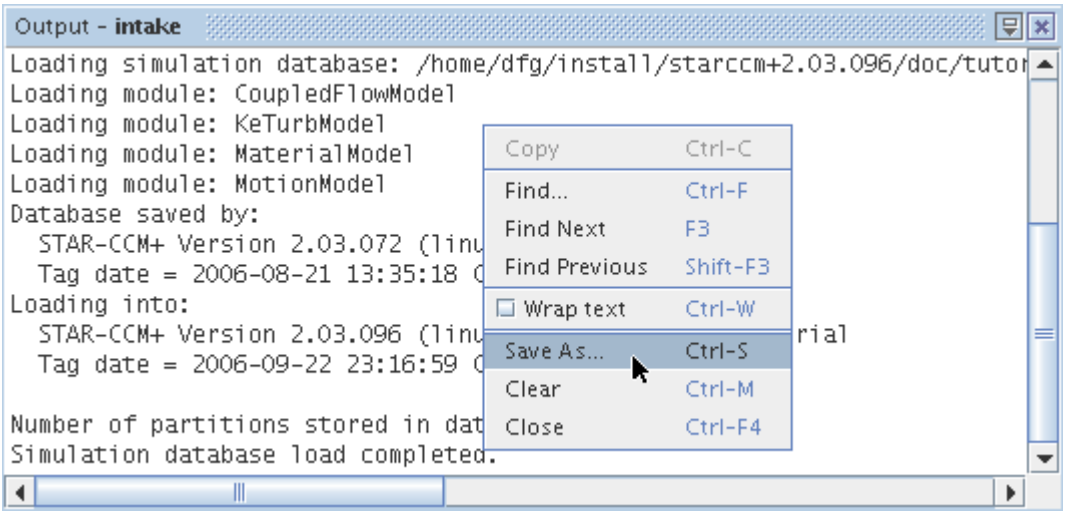
Saving the Current Output as a Text File

This technique only captures what still appears in the *Output* window.

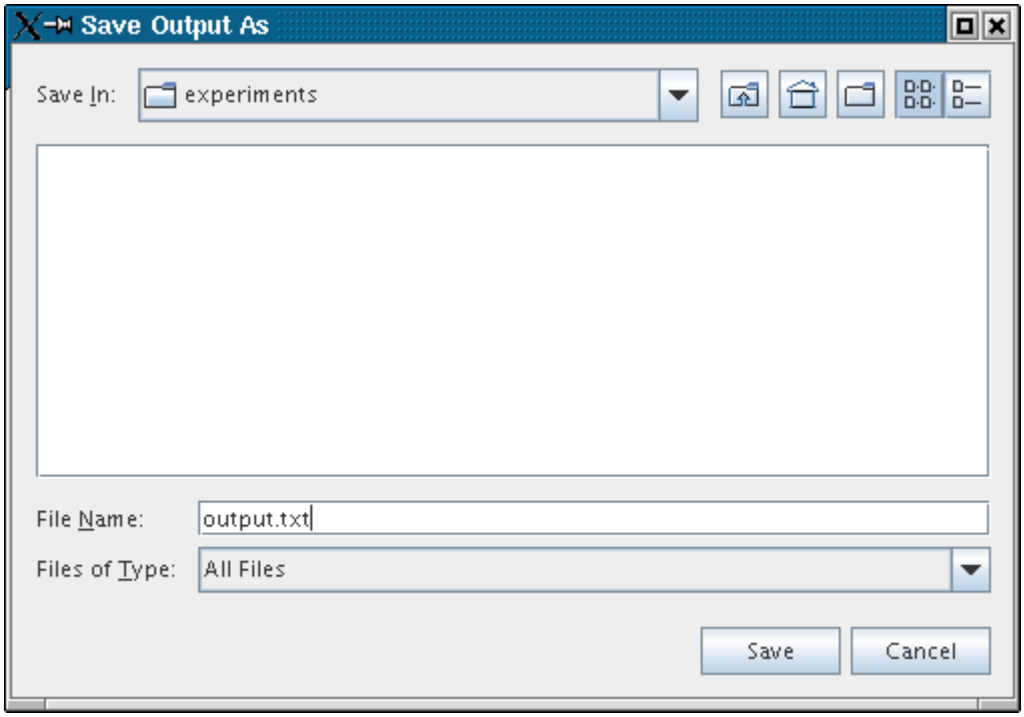
Earlier portions of the output may be lost unless you have automated saving enabled.

To save the contents of the *Output* window in a standard text file:

1. Right-click in the window and select **Save As...**



A *Save Output As* dialog will appear allowing you to navigate to where you want to save the output.

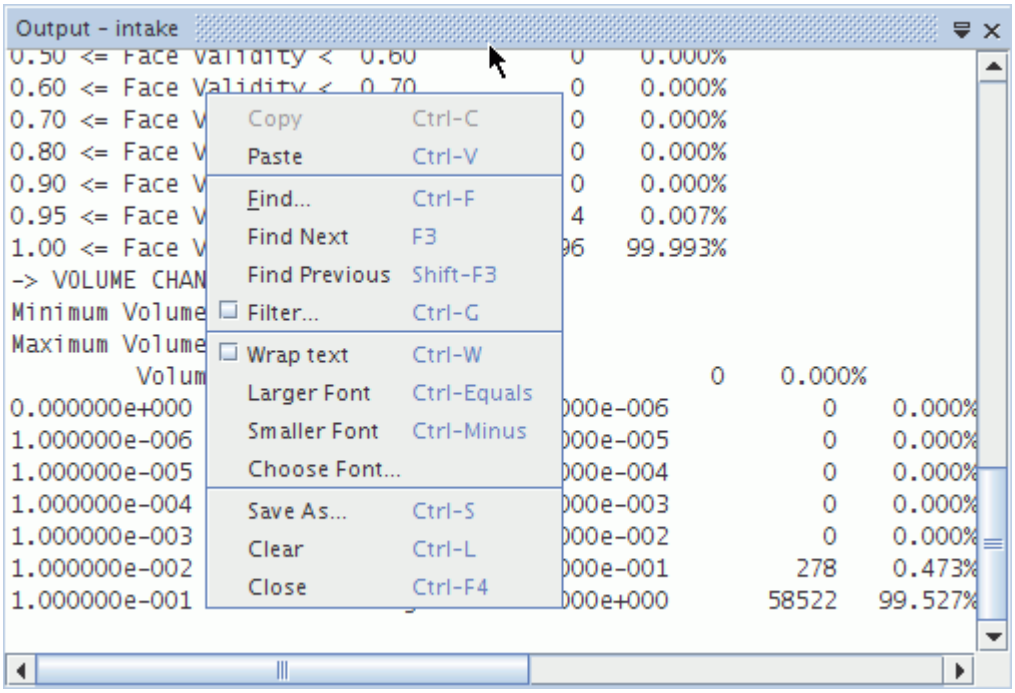


2. Click **Save** to create the text file.

Working with Output Text

You can work with the contents of the *Output* window as you would with a typical text document.

Right-clicking in the window reveals a set of options similar to those that exist in common word processors.



These options are described in the following sections.

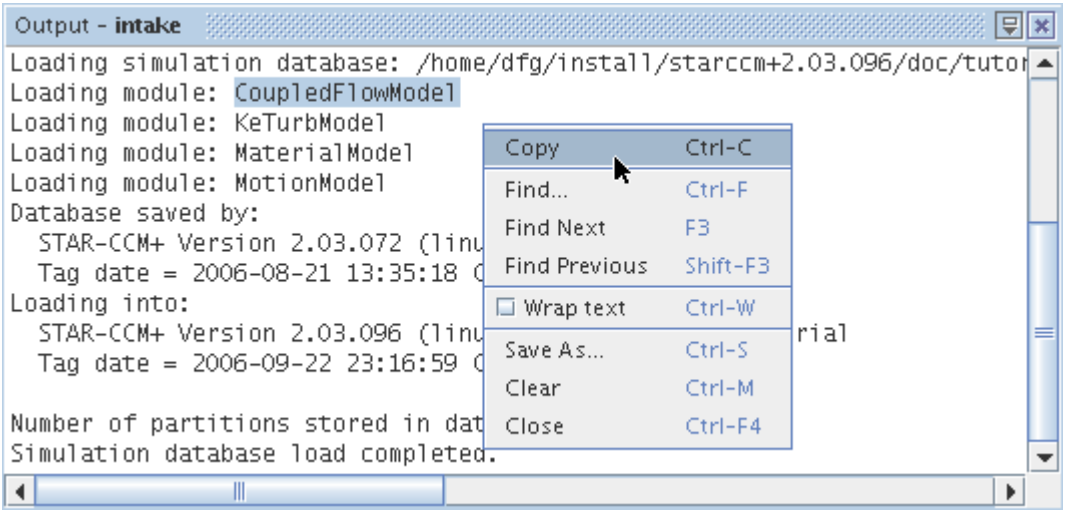
Contents:

Copying Output Text to the Clipboard
Finding a Term in the Output Text
Filtering the Output Text
Wrapping the Output Text
Changing the Font Size in the Output Text
Customizing Fonts
Clearing the Output Text

Copying Output Text to the Clipboard

To copy material from the window to the clipboard, highlight that material:

1. Right-click in the window and select **Copy** from the pop-up menu.

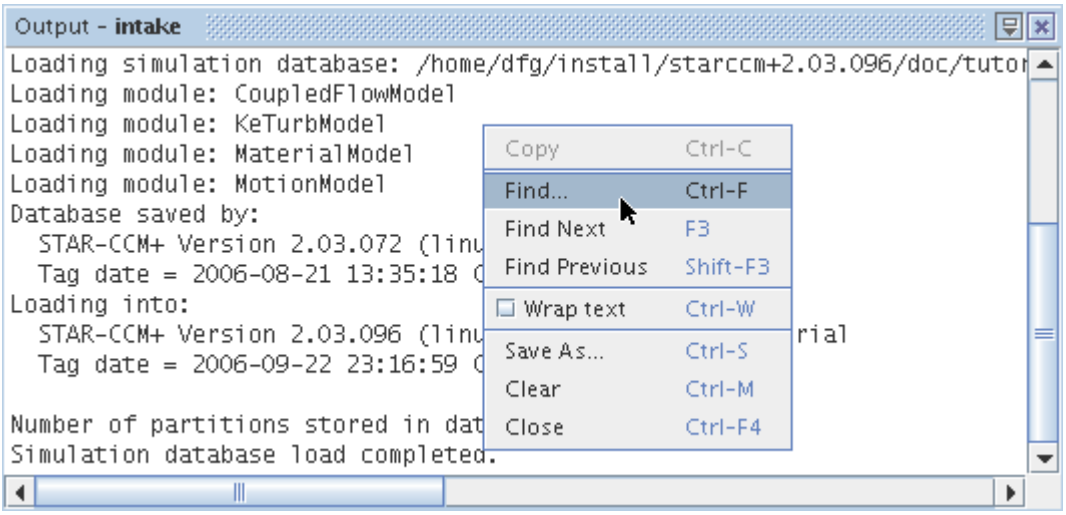


This menu item is disabled (grayed out) when no text in the *Output* window is highlighted.

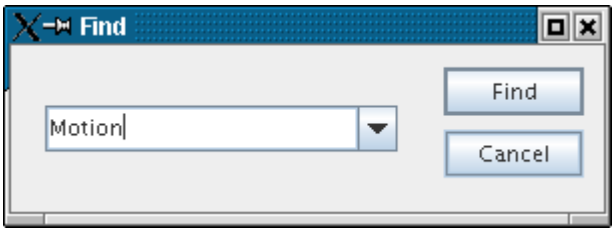
Finding a Term in the Output Text

To find a term in the text:

1. Right-click in the window and select **Find...** from the pop-up menu.

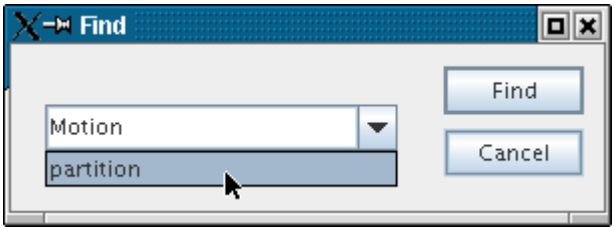


A separate dialog appears in which you can enter a search term.



2. Click **Find** to locate that term.

The dialog also contains a drop-down list of all your search entries from the current STAR-CCM+ session.

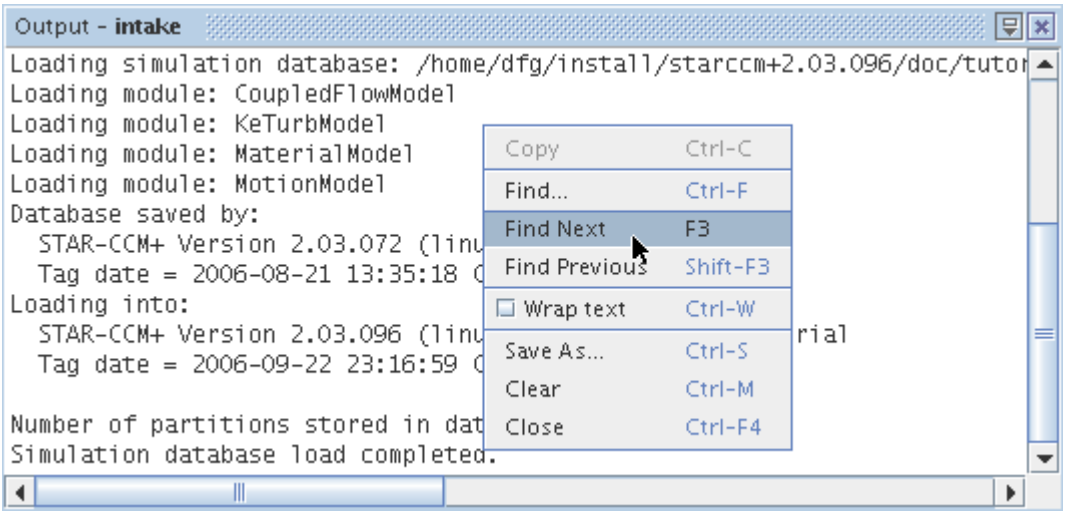


Finding the Next Occurrence of a Term in the Output Text

Sometimes a term appears many times in a text.

To repeat your search for a term efficiently:

1. Right-click in the window and select **Find Next** from the pop-up menu.

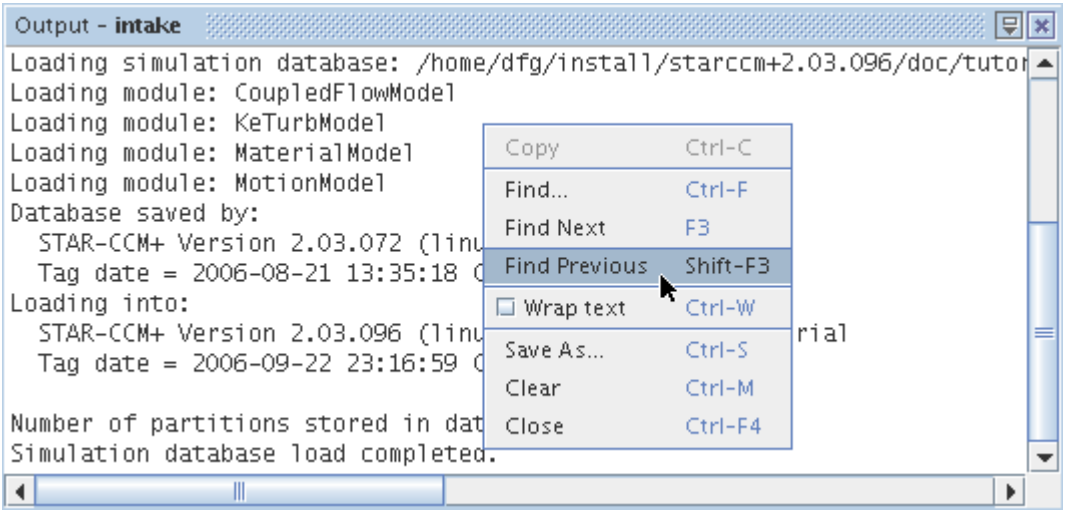


Finding the Previous Occurrence of a Term in the Output Text

You can search for a term backwards.

To search for a term backwards:

1. Right-click in the window and select **Find Previous** from the pop-up menu.



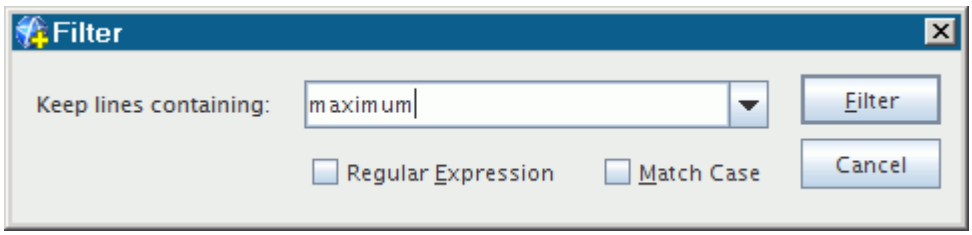
Filtering the Output Text

You can control the contents of the *Output* window so that it shows only certain types of information or messages.

To view only the lines with a certain text string in them:

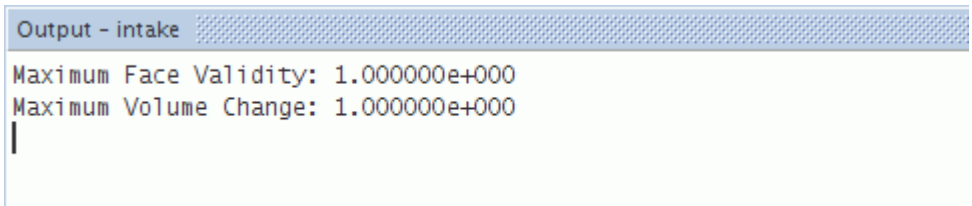
1. Right-click in the window and select **Filter...** from the pop-up menu.

   In the *Filter* dialog, enter the string that you want and click **Filter**.

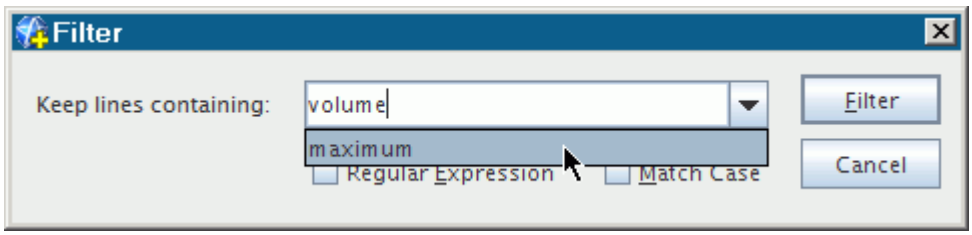


2. Activate the **Regular Expression** checkbox to include variations of the string.

   The window shows only those lines of text.



The dialog also contains a drop-down list of all your entries from the current STAR-CCM+ session.



3. To restore the full content of the window, right-click and select **Filter...** again.

To select the severity level of messages:

4. Right-click in the window and activate or deactivate any combination of the following filter options from the pop-up menu. All these options are activated by default.
   - **Errors** -- includes error messages in the output text.
   - **Warnings** -- includes warnings in the output text.
   - **Information** -- includes all contents of the *Output* window other than error messages and warnings.

     Caution: If you deactivate this option, you lose the output unless you have automated saving activated.

   You can, for example, choose to omit errors and warnings from the window. Alternatively, you can deactivate the **Information** option so that the window is blank except when errors or warnings appear.

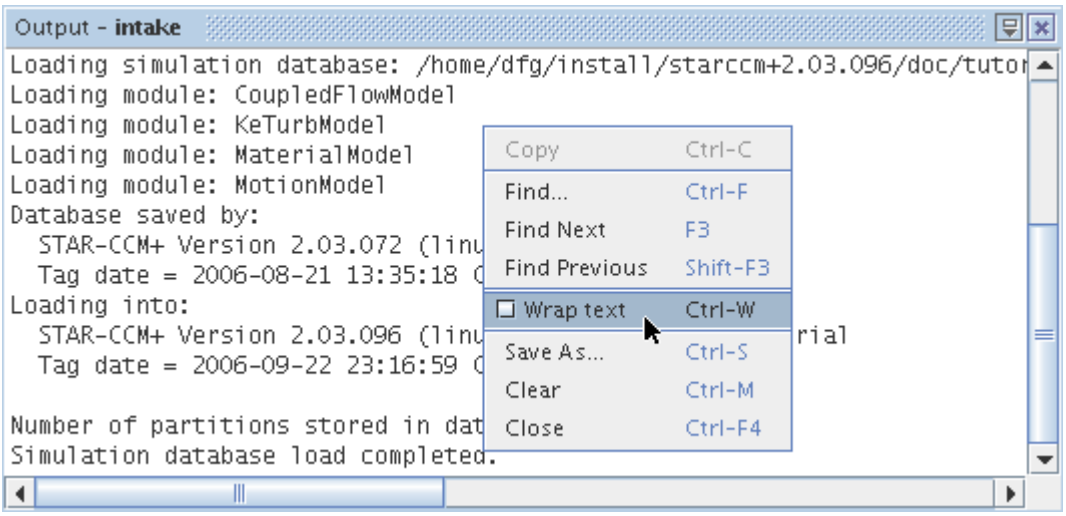
   Note: A message can contain the word `error` without necessarily having the message severity of an error, which is displayed in red.

Wrapping the Output Text

By default the text lines appear to run off the side of the *Output* window.

To make the text “wrap”, or break lines according to the window width:

1. Right-click in the window and select **Wrap text** from the pop-up menu.



2. To restore the default setting, select the same menu item again.

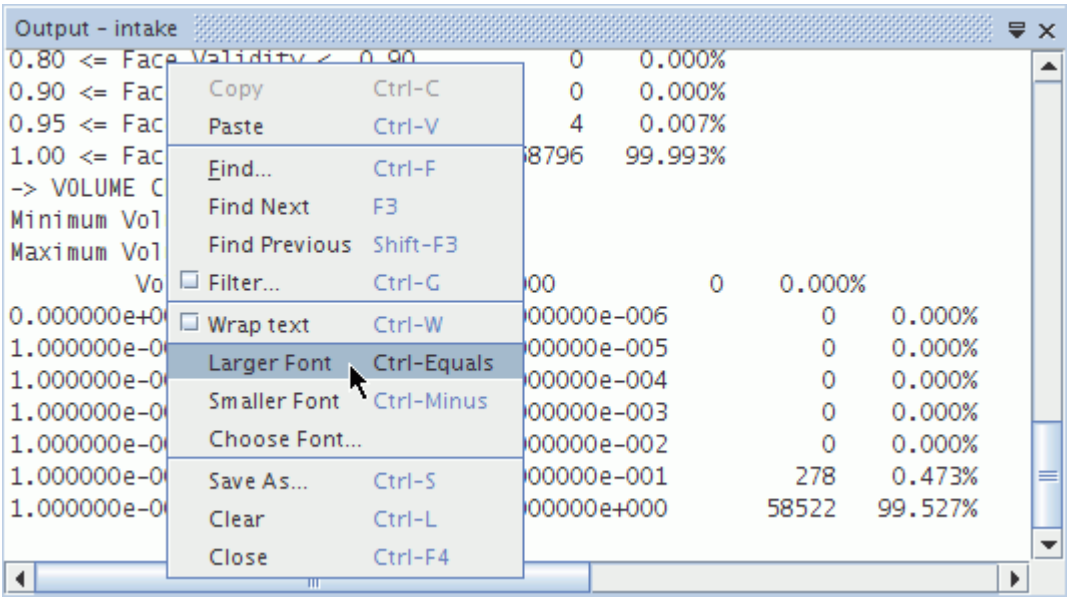
Changing the Font Size in the Output Text

STAR-CCM+ offers the ability to change the font size of the output text.

Increasing Font Size in the Output Text

To increase the font size incrementally:

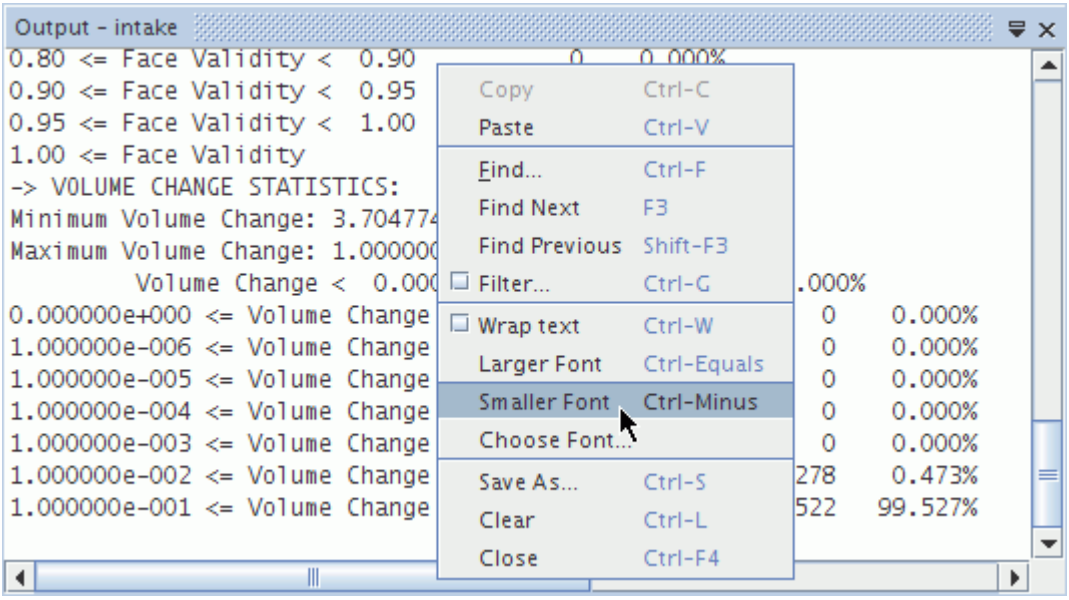
1. Right-click in the window and select **Larger Font** from the pop-up menu.



Decreasing Font Size in the Output Text

To decrease the font size incrementally:

1. Right-click in the window and select **Smaller Font** from the pop-up menu.

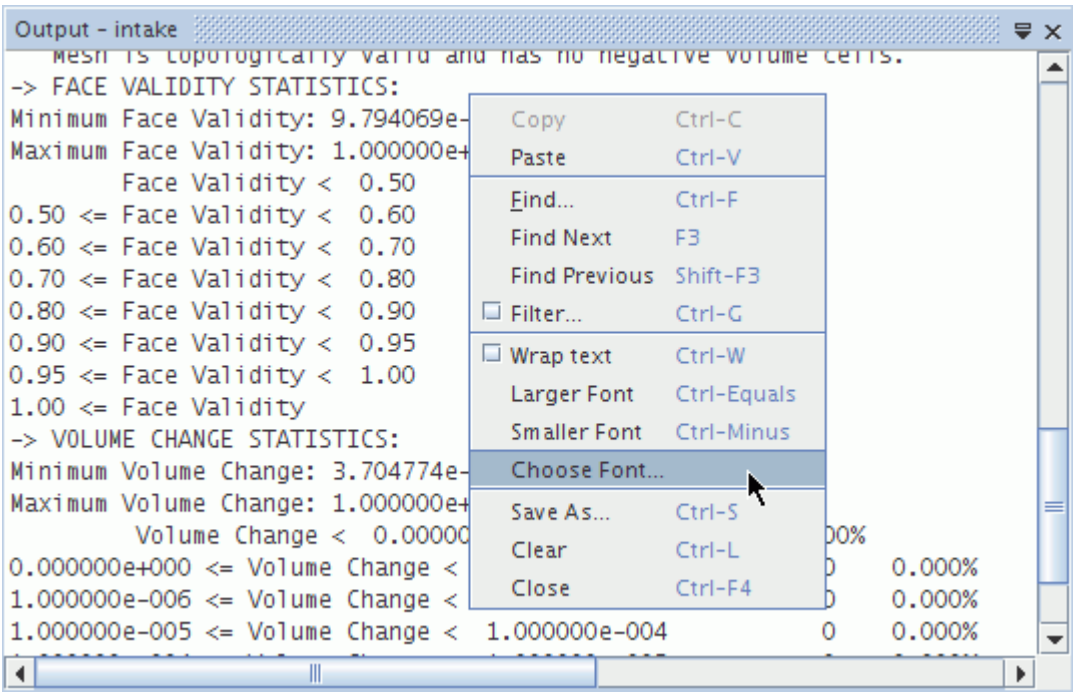


It is also possible to customize fonts.

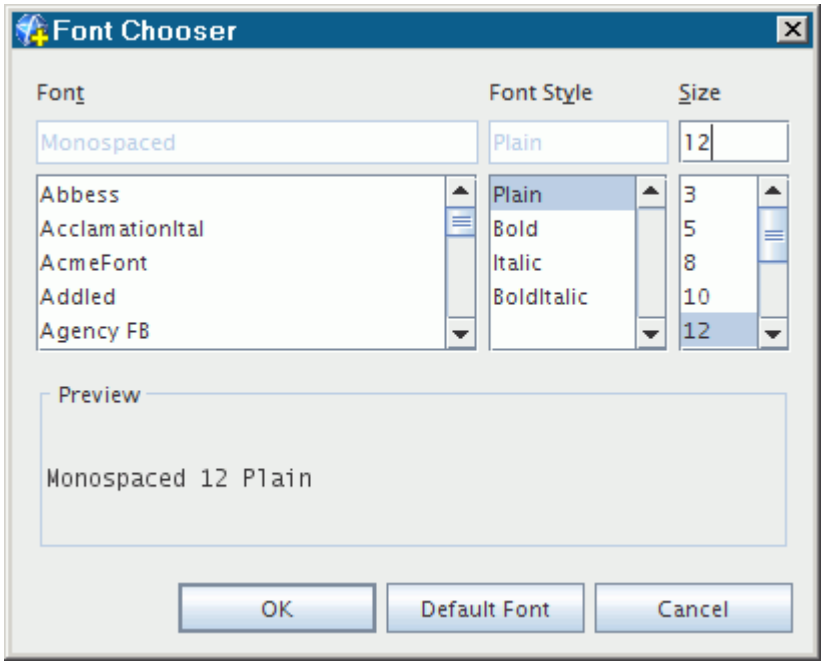
Customizing Fonts

It is possible to customize the appearance of fonts in the *Output* window.

1. Right-click in the window and select **Choose Font...** from the pop-up menu.



2. In the *Font Chooser* dialog, you can specify the font family, font style, and font size. Alternatively, you can revert any customized settings by clicking **Default Font**.
3. Click **OK** to apply your changes.

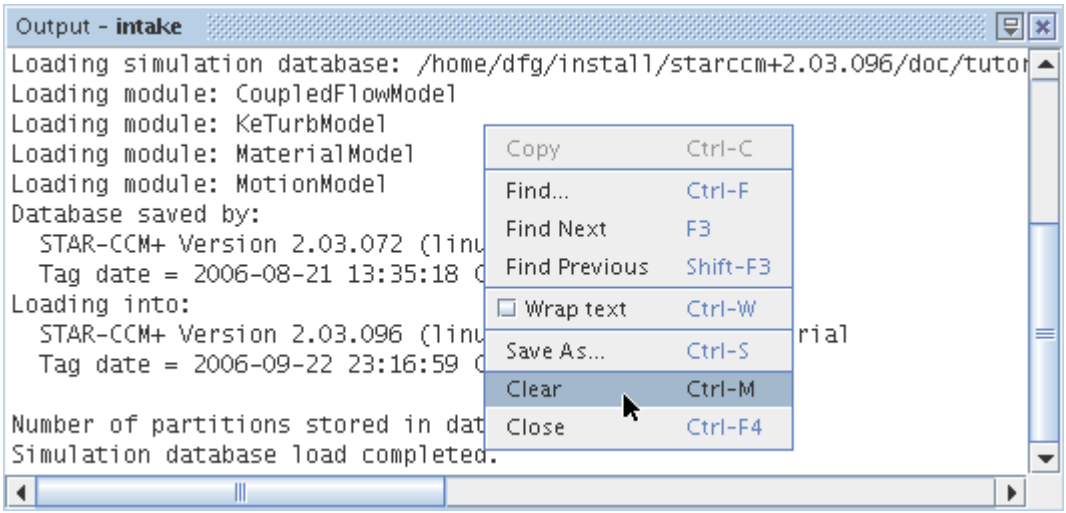


Clearing the Output Text

To remove the contents of the window for the current simulation:

1. Right-click in the window and select **Clear** from the pop-up menu.

This action results in the loss of these contents if you did not save them as a file previously.



If multiple simulations are open, the *Output* window contains multiple tabs with output for each simulation, and this action in one tab will not affect the contents of any other tab.

Using Keyboard Shortcuts

STAR-CCM+ accommodates keystrokes for a host of activities.

You can use the keyboard to navigate the menu bar, work with the table editor, access elements of the *Properties* window, manipulate the scene display.

Keyboard shortcuts for surface preparation are also provided in the Preparing Surfaces section.

Contents:

Using the Keyboard to Navigate the Menu Bar
Navigating the Table Editor with the Keyboard
Accessing the Properties Window with the Keyboard
Using Keyboard Shortcuts in the Scene Display

Using the Keyboard to Navigate the Menu Bar

The keyboard can be used to access the menu bar when you hold down the `<Alt>` key and simultaneously press the underscored character of the menu button of interest. You can also use the `<Right>` and `<Left>` arrow keys to move from one menu to the next.

File

Action	Keyboard Shortcut
Create a new simulation.	`<Ctrl>+<N>` or `<Alt>+<F>+<N>`
Load an existing simulation.	`<Ctrl>+<L>` or `<Alt>+<F>+<L>`
Reload the simulation.	`<Alt>+<F>+<O>`
Connect to a server.	`<Alt>+<F>` then select with `<Down>` arrow key and press `<Enter>`.
Select a recently opened simulation file.	`<Alt>+<F>+<R>` then select with `<Up>` or `<Down>` arrow key and press `<Enter>`.
Save changes to the current simulation.	`<Alt>+<F>+<S>`
Save a copy of the current simulation file with a new name and/or in a new location.	`<Ctrl>+<S>`
Save changes to all currently open simulations.	`<Alt>+<F>` then select with `<Down>` arrow key and press `<Enter>`.
Save changes to the current simulation at regular intervals automatically (auto-save).	`<Alt>+<F>` then select with `<Down>` arrow key and press `<Enter>`.
Play a macro.	`<Alt>+<F>+<C>+<P>`

Action	Keyboard Shortcut
Replay a recently played macro.	<Alt>+<F>+<C>+<T> then select with <Up> or <Down> arrow key and press <Enter>.
Start recording a macro.	<Alt>+<F>+<C>+<R>
Stop recording a macro.	<Alt>+<F>+<C>+<S>
Edit a custom macro.	<Alt>+<F>+<C>+<E>
Save the default macro.	<Alt>+<F>+<C>+<D>
Import a surface mesh.	<Alt>+<F>+<I> then press <Enter>.
Select a recently imported surface mesh.	<Alt>+<F>+<I>, select with <Down> then <Right> arrow key; select with <Up> or <Down> arrow key and press <Enter>.
Import a volume mesh.	<Alt>+<F>+<I> then select with <Down> arrow key and press <Enter>.
Select a recently imported volume mesh.	<Alt>+<F>+<I>, select with <Down> then <Right> arrow key; select with <Up> or <Down> arrow key and press <Enter>.
Import a CAE model.	<Alt>+<F>+<I> then select with <Down> arrow key and press <Enter>.
Select a recently imported CAE model.	<Alt>+<F>+<I>, select with <Down> then <Right> arrow key; select with <Up> or <Down> arrow key and press <Enter>.
Export simulation data.	<Alt>+<F>+<E>
Export simulation data at regular intervals during the run automatically (auto-export).	<Alt>+<F> then select with <Down> arrow key and press <Enter>.
Generate a summary report.	<Alt>+<F> then select with <Down> then <Right> arrow key; select with <Up> or <Down> arrow key and press <Enter>.
Load the Simulation Assistant.	<Alt>+<F>+<A>
Specify the page setup of display for printing.	<Alt>+<F>+<G>
Print contents of display.	<Alt>+<F>+<P>
Disconnect from the server.	<Alt>+<F>+<D>

Edit

Action	Keyboard Shortcut
Copy the selected object.	<Alt>+<E>+<Y>

Action	Keyboard Shortcut
Paste the contents of the clipboard into the selected location in the object tree.	<Ctrl>+<V> or <Alt>+<E>+<P>
Delete the selected object.	<Alt>+<E>+<D>

Mesh

Action	Keyboard Shortcut
Scale the mesh.	<Alt>+<M>+<S>
Run mesh diagnostics.	<Alt>+<M>+<D>
Extract a two-dimensional mesh.	<Alt>+<M>+<2>

Solution

Action	Keyboard Shortcut
Clear the solution data of the simulation.	<Alt>+<S>+<C>
Run the simulation.	<Alt>+<S>+<R>
Step the solution.	<Alt>+<S>+<S>
Set step size.	<Alt>+<S>+<T>
Stop the solution.	<Alt>+<S>+<P>

Tools

Action	Keyboard Shortcut
Access the add-ons.	<Alt>+<T>+<A>
Open the *Options* dialog.	<Alt>+<T>+<O>
Reset options.	<Alt>+<T>+<R>
Access the plugins.	<Alt>+<T>+<G>

Window

Action	Keyboard Shortcut
Open the *Output* window, or make it active if it is already open.	<Ctrl>+<4>
Restore the default layout of the workspace.	<Alt>+<W>+<R>
Save the existing layout as the default.	<Alt>+<W>+<S>
Close the selected window (*Properties* or windows in the *Explorer* pane only).	<Ctrl>+<F4> or <Alt>+<W>+<C>

Action	Keyboard Shortcut
Maximize (or restore when maximized) the selected window (*Properties* or windows in the *Explorer* pane only).	<Shift>+<Esc> or <Alt>+<W>+<Z>
Undock the selected window.	<Alt>+<Shift>+<D>
Show or hide a toolbar. This action toggles between showing and hiding the selected toolbar.	<Alt>+<W>+<T> then select with <Up> or <Down> arrow key and press <Enter>.
Switch to or from small toolbar icons.	<Alt>+<W>+<T>+<S>
Reset toolbars to default configuration.	<Alt>+<W>+<T>+<T>
Open a dialog to customize toolbars.	<Alt>+<W>+<T>+<C>

Help

Action	Keyboard Shortcut
Access the online Help.	<F1>
Get information about the current version of STAR-CCM+.	<Alt>+<H>+<A>

Navigating the Table Editor with the Keyboard

For navigating among the cells in the body of the table, the following keys are supported:

Keyboard Shortcuts

Action	Keyboard Shortcut
Shift the focus from one column to the one right of it. When the focus is in the tree column, pressing this arrow opens the node of the row (assuming it is closed).	<Right> arrow
When the focus is in the tree column, pressing this combination opens all nodes and subnodes within the selected node of the row.	<Ctrl>+<Right> arrow
Shift the focus from one column to the one left of it. When the focus is in the tree column, pressing this arrow closes the node of the row (assuming it is open).	<Left> arrow
When the focus is in the tree column, pressing this combination closes all nodes and subnodes within the selected node of the row.	<Ctrl>+<Left> arrow
Shift the focus from one column to the one right of it.	<Tab>

Action	Keyboard Shortcut
Shift the focus from one column to the one left of it.	<Shift>+<Tab>
Shift the focus from one row to the one above it.	<Up> arrow
Shift the focus from one row to the one below it.	<Down> arrow
Shift the focus to the left cell of the row.	<Home>
Shift the focus to the right cell of the row.	<End>
Go to edit mode in a property cell.	<Space bar>
Go to view mode in a property cell, assuming that cell is in edit mode. This does not change the value if you have changed it. This key closes the dialog if pressed when there is no property cell in edit mode.	<Esc>
Display drop-down list -- when the cell of that list has the focus.	<Space bar>
Move from a drop-down list item to the one above it.	<Up> arrow
Move from a drop-down list item to the one below it.	<Down> arrow
Select a drop-down list item.	<Enter>
Exit a drop-down list without entering any change.	<Esc>
Move to the top row of the table body.	<Ctrl>+<Home>
Move to the bottom row of the table body.	<Ctrl>+<End>

To select a contiguous block of nodes, select the first node of the block using the <Up> or <Down> arrow keys, hold down the <Shift> key, then press the <Down> arrow until you reach the last node of the block.

To select a non-contiguous block of nodes, select the first node of the block using the <Up> or <Down> arrow keys, hold down the <Ctrl> key, then press the <Down> arrow until you reach the next node of the block and press the <Space> bar. Continue navigating up or down and pressing <Space> on required nodes until you are done.

Accessing the Properties Window with the Keyboard

Once the *Properties* window has the focus, the following keys are supported:

Navigating

Action	Keyboard Shortcut
Close an open category.	<Left> arrow
Open a closed category.	<Right> arrow
Move the focus to the previous property in the list, stopping at the category bar of the top list.	<Up> arrow

Action	Keyboard Shortcut
Move the focus to the next property in the list, stopping at the last item of an open category or the category bar of a closed category.	<Down> arrow
Move to the next property.	<Up> arrow or <Down> arrow
Put the property into edit mode if it is not read-only (with the text selected if the inline editor is a text box).	<Space bar>

In edit mode, the keys available depend on the type of inline editor.

Text Box

Action	Keyboard Shortcut
Accept the new value (subject to error checking) and go to view mode. If the error checking fails, the value reverts to the previous value.	<Enter>
Revert the property to the previous value, going to view mode, provided you have not typed <Enter>.	<Esc>

Drop-Down List

Action	Keyboard Shortcut
Open the list.	<Space bar>
Change the selection to the item above.	<Up> arrow
Change the selection to the item below.	<Down> arrow
Accept the list selection, reverting to view mode.	<Enter>
Go to view mode, not changing the value if you have changed it.	<Esc>

When you have the drop-down list open, pressing any letter will always be interpreted as the first letter of the required item. Repeated presses of that key will continue to cycle through the set of items that begin with that letter.

For example, suppose you open a drop-down list with scalar functions, and you want to select Temperature. When you press <T>, you may highlight Tangential Velocity (depending on the simulation). If you try pressing <T> followed by <E>, you may highlight Tangential Velocity and then Effective Conductivity. Instead, keep pressing <T> until Temperature is highlighted. Then press <Enter> to complete the selection.

Mini-Dialog

There is no keyboard support for mini-dialogs.

See Also:

Using the Properties Window

Using the Scalar and Vector Bars

Using Keyboard Shortcuts in the Scene Display

Keyboard shortcuts allow you to work with the scene display as follows:

Changing the View

Action	Keyboard Shortcut
Bring the view to the front of the object. Flip it to the back with a second press. Additional strokes of this key toggle between front and back.	<F>
Bring the view to the side of the object. Flip it to the other side with a second press. Additional strokes of this key toggle between front and back.	<S>
Flip the picture horizontally along a vertical axis in the center of the view.	<C>
Bring the view to the top of the object. Flip it to the bottom with a second press. Additional strokes of this key toggle between front and back.	<T>

Moving the Image on an Axis

Action	Keyboard Shortcut
Move the image in a negative direction around the x-axis. Each keystroke turns the image 10 degrees. The shifts are in relation to the focal point of the display.	<X>
Move the image in a positive direction around the x-axis.	<Shift>+<X>
Move the image in a negative direction around the y-axis.	<Y>
Move the image in a positive direction around the y-axis.	<Shift>+<Y>
Move the image in a negative direction around the z-axis.	<Z>
Move the image in a positive direction around the z-axis.	<Shift>+<Z>

Obtaining Scalar Data for a Cell

Action	Keyboard Shortcut
Determine the scalar value for a cell when hovering with the pointer over the location (scalar scenes only).	<.>

Resetting the View

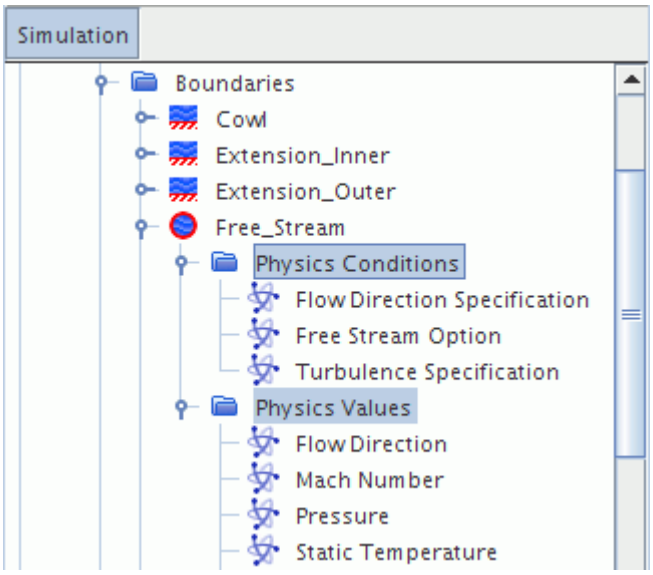
Action	Keyboard Shortcut
Restore the original view of the scene.	<R>

Setting the Focal Point

Action	Keyboard Shortcut
Set the focal point of the camera to the position where you hover with the mouse.	<H>

Conditions, Values, and Profiles

Throughout the simulation tree, STAR-CCM+ presents groups of condition nodes and their matching value nodes. For example, a boundary object has child nodes on which you choose physics conditions, and child nodes on which you set physics values. These values are quantities such as velocity, temperature, or density. The particular values available for change depend on what you set in the conditions.



It is possible to do the following:

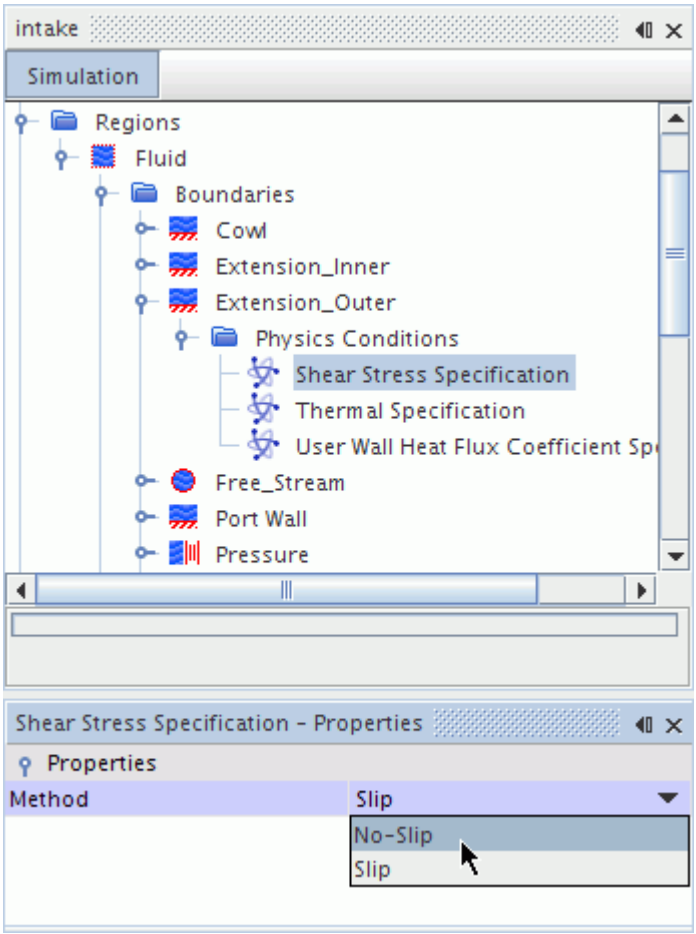
- Copy properties from one value or condition node to another
- Define dimensionality of a value or unit
- Set several conditions or several values of the same type at the same time.

What Are Conditions?

Conditions specify additional information about simulation objects. For example, with a boundary of type wall, the conditions tell you whether this wall is to be a no-slip wall, a slip wall, or a moving wall. The conditions also tell you whether you want to apply a specified temperature (Dirichlet) thermal boundary condition or a specified heat flux (Neumann) thermal boundary condition.

Setting Conditions

To set the conditions for a region, boundary or interface, open the node corresponding to the specific region, boundary, or interface. Then open the **Conditions** node and select the required condition. In the *Properties* window, select one of the available options from the drop-down list corresponding to the *Method* property.



What Are Values?

The types and conditions inform models how to deal with a boundary, a region, or an interface, but they do not specify actual numerical input. Values provide these numbers. Some examples of how certain conditions require values follow:

- The region condition specifies that a volumetric heat source must be applied to a region with an energy model, but the value specifies the magnitude of the volumetric heat source.
- The boundary condition specifies that a wall is to have a specified heat flux, but the value specifies the magnitude of the heat flux.
- The interface condition specifies that a thermal contact resistance (say due to a layer of paint) is associated with a contact interface between a solid region and a fluid region, but the value specifies the value of the contact resistance.

What Are Profiles?

Profiles are a STAR-CCM+ concept that defines the methods by which you can provide values to the simulation. For example, you may wish to define an inlet temperature as a function of time, rather than as a constant value. Several profile types provide a field function option that allows you to do this. The basic types of profiles are:

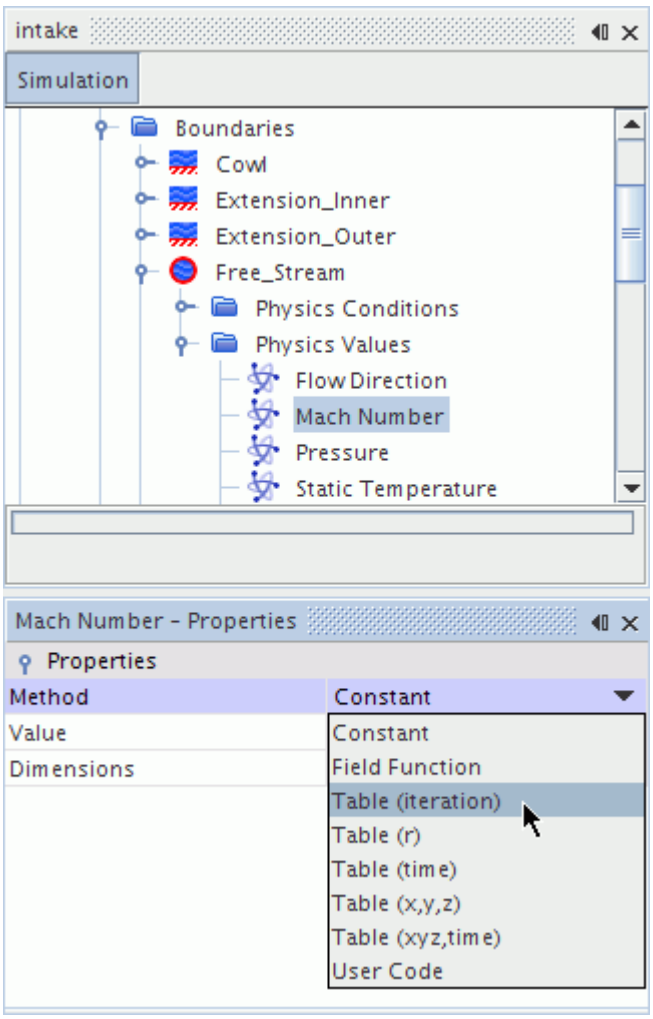
- Scalar

- Scalar Array
- Vector
- Tensor

If you want to use the same value at all points on the boundary, then you can enter constant as the profile. You also have the flexibility of setting conditions that are dependent on other values (using field functions), or of reading in a table of values from a file.

Setting Profile Methods

To set a value for a region, boundary or interface, first set the condition. Once the condition is set, open the **Values** node and select the node of the required profile. By default the simplest method is assumed, a constant method for most profile types. To change this method, select one of the items (`Table (iteration)` in the following screenshot) in the drop-down list corresponding to the *Method* property in the *Properties* window.



Contents:

Scalar and Vector Quantities

Arrays of Scalar Quantities
Tensor Quantities
Complex Quantities
Setting Values Using the Composite Vector Profile Method
Setting Values Using an Expression
Setting Values Using Distributions
Setting Values Using a Table
Setting Values Using a Field of Direction Vectors
Setting Values Using a Polynomial
Copying Properties
Defining Dimensionality
Setting Multiple Corresponding Conditions Simultaneously
Setting Multiple Corresponding Values Simultaneously

See Also:

Using the Properties Window
Setting Up Physics
Physics Continua
Setting Types

Scalar and Vector Quantities

Scalar and vector quantities are common forms of profiles. Pressure profiles are a special case of scalar profile.

Using Scalar Profiles

Scalar profiles are used to specify scalar quantities such as Mach number, temperature, and mass flow.

In some cases, the node exists in the **Values** node independent of the corresponding **Conditions**. However, in other cases, the node is added to the **Values** node according to the method selected in the **Conditions** node. For example, mass flow can be entered as mass flux or as mass flow rate.

Some profile nodes have dynamic properties. The setting of the **Method** property determines which other properties are present. For example, when you set the **Method** property of a scalar profile to **Constant**, the *Properties* window includes the **Value** property. But when **Method** is set to **Field Function**, then the **Scalar Function** property appears instead of the **Value** property.

Scalar Profile Properties

Method	Selects the method to use for entering the scalar value.	
	Constant	Specifies the scalar value as a single, constant value.

	Field Function	Defines the scalar value using a field function (typically user defined).
	Normally distributed	Defines the scalar value randomly using a normal distribution. For details, see Setting Values Using Distributions.
	Table (iteration)	Defines the scalar value as a function of iteration number.
	Table (r)	Defines the scalar value as a function of radius.
	Table (time)	Defines the scalar value as a function of physical time.
	Table (x,y,z)	Defines the scalar value as a function of position.
	Table (xyz, time)	Defines the scalar value as a function of position and time.
	Tabular	Defines the value using tabular data.
	Uniformly distributed	Defines the scalar value randomly using a uniform distribution. For details, see Setting Values Using Distributions.
	User Code	Defines the scalar value using a user subroutine.
Value	Specifies the value with a text entry. You can also enter an expression, such as `10 * sin(3.14 * $Time)`, directly. This property is only available when **Method** is set to **Constant**.	
Scalar Function	Activates an object selector dialog from which you can choose a field function. This property is only available when **Method** is set to **Field Function**.	
Function	Provides a drop-down list from which you can choose a user function. This property is only available when **Method** is set to **User Code**.	
Dimensions	The dimensionality of the value (read-only). Clicking the customizer executes the define dimensionality action.	

Using Pressure Profiles

Pressure profiles are similar to scalar profiles except that they have an additional option in the *Method* property—**Radial Equilibrium**. This option defines the pressure using a radial equilibrium condition.

There is no corresponding node in the **Conditions** node.

Pressure Profile Properties

Hub Pressure	Specifies the pressure. This property is only available when **Method** is set to **Radial Equilibrium**.

Using Vector Profiles

A vector profile is used to specify a vector quantity.

Typical vector quantities are momentum source, flow direction, and velocity. This profile's *Coordinate System* property lets you switch from the default laboratory coordinate system to one of multiple local coordinate systems, depending on how you want to set the conditions in your simulation.

Vector Profile Properties

Method	Selects the method to use for entering the vector.	
	Composite	Defines the vector as three (for example, X, Y, Z) components (or two (for example, X, Y) for two-dimensional problems).
	Constant	Specifies the vector as a single three-part (or two-part for two-dimensional), comma-separated, number.
	Feature Curve	Specifies the curve parts, representation, and orientation from which the local direction is calculated internally.
	Field Function	Defines the vector using a field function (typically user defined).
	Normally distributed	Specifies the vector randomly using a normal distribution. For details, see Setting Values Using Distributions.
	Table (iteration)	Defines the vector as a function of iteration number.
	Table (r)	Defines the vector as a function of radius.
	Table (time)	Defines the vector as a function of physical time.
	Table (x,y,z)	Defines the vector as a function of position.
	Table (xyz, time)	Defines the vector as a function of position and time.

	Uniformly distributed	Specifies the vector randomly using a uniform distribution. For details, see Setting Values Using Distributions.
	User Code	Defines the vector using a user subroutine.
Value	Specifies the x, y, and z values. These values are separated by commas in a text entry. You can also enter an expression, such as `10 * sin(3.14 * $Time)`, directly. The property customizer opens a dialog which contains separate fields for these values. This property is only available when **Method** is set to **Constant**.	
Vector Function	Activates an object selector dialog from which you can choose a field function. This property is only available when **Method** is set to **Field Function**.	
Function	Provides a drop-down list from which you can choose a user function. This property is only available when **Method** is set to **User Code**.	
Dimensions	The dimensionality of the value (read-only). Clicking the customizer executes the define dimensionality action.	
Coordinate System	The coordinate system to use.	
	Cartesian	Specifies the Cartesian coordinate system.
	Cylindrical	Specifies the cylindrical coordinate system.
	Laboratory	Specifies the laboratory coordinate system.
	Spherical	Specifies the spherical coordinate system.

Arrays of Scalar Quantities

A scalar array is used to specify multiple values of mass-fraction or volume-fraction where each value corresponds to one component of the mixture.

Properties

Method	Selects the method to use for entering the scalar value.	
	Composite	Defines the method as a composite array profile, which serves as a placeholder for separate scalar nodes.

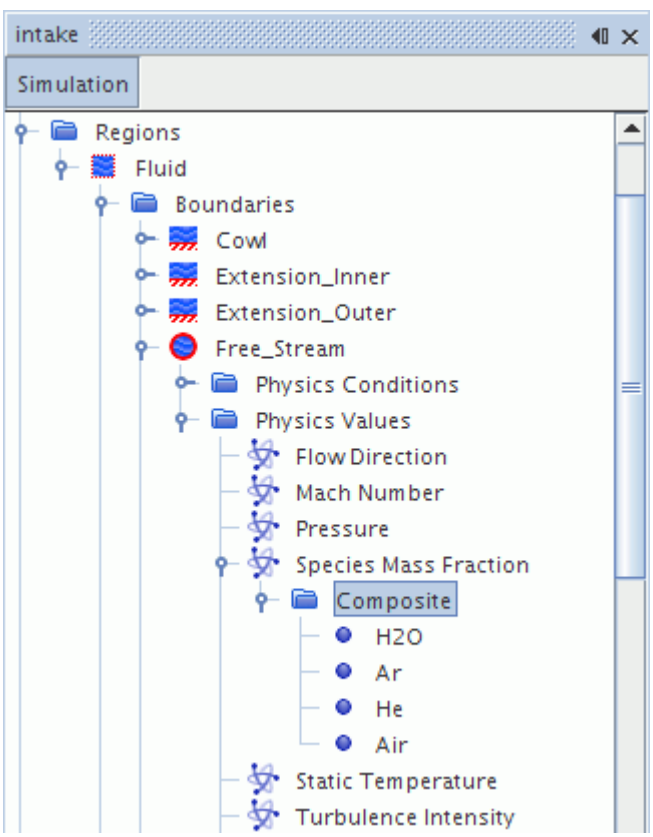
	Constant	Defines the method as a single constant array. This constant method differs from a typical scalar constant method in that you provide it with a series of values. The number and sequence of these values correspond to the number and sequence of the components of the array.
Value	Specifies a series of comma-separated values with a text entry. This property is only available when **Method** is set to **Constant**.	
Dimensions	The dimensionality of the value (read-only). Clicking the customizer executes the define dimensionality action.	

Setting Values Using a Composite Array

The composite array profile method node serves as a placeholder for separate nodes, each of which represents a selected component in a mixture.

The component nodes let you specify profile methods for each component separately:

1. Open the **Composite** node.
2. Select a component node.
3. Select a setting for the **Method** property of the component node.



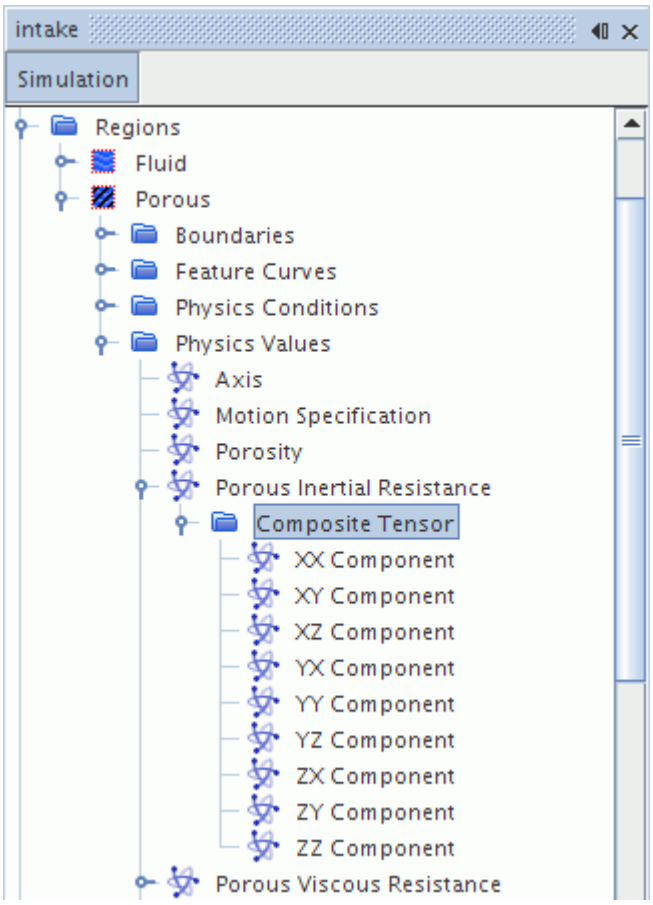
Properties of a Composite Array

Method	Selects the method to use for entering the scalar component. For a list of available methods, see the Methods for a Scalar Profile.
Dimensions	The dimensionality of the value (read-only). Clicking the customizer executes the define dimensionality action.

Tensor Quantities

A tensor profile is used to specify tensor quantities such as porous resistances.

A tensor node serves as a placeholder for separate nodes representing independent values for components and for any axes. In the example in the following screenshot, there is a node for each component of a 3-by-3 matrix.



Properties

Dimensions	The dimensionality of the value (read-only). Clicking the customizer executes the define dimensionality action.

Method	Selects the method to use for entering the tensor.	
	Axisymmetric Tensor	Defines the method as an axisymmetric tensor profile, which serves as a placeholder for separate scalar nodes.
	Composite Symmetric Tensor	Defines the method as a composite symmetric tensor profile, which serves as a placeholder for separate scalar nodes.
	Composite Tensor	Defines the method as a composite tensor profile, which serves as a placeholder for separate scalar nodes.
	Isotropic Tensor	Defines the method as an isotropic tensor profile, which serves as a placeholder for separate scalar nodes.
	Principal Tensor	Defines the method as a principal tensor profile, which serves as a placeholder for separate scalar and vector nodes.

Setting Values Using an Axisymmetric Tensor

In the axisymmetric tensor profile method, a tensor is specified using its components along a user-specified axis and those normal to the axis. The node serves as a placeholder for separate component and axis nodes.

The Axis node represents the axisymmetric direction. There are two component nodes: **Axial component** which is along the user-specified Axis, and **Cross-Stream Component** which is along the direction normal to the Axis. These nodes function just like the scalar profile quantity nodes in a typical composite vector profile method.

For example, to specify the tensor

$$\begin{bmatrix} 20 & 0 & 0 \\ 0 & 30 & 0 \\ 0 & 0 & 30 \end{bmatrix}$$

specify the **Axis** as a **Constant** value of `[1, 0, 0]`. Then set the **Axial Component** to `20`, and the **Cross-Stream Component** to `30`.

Setting Values Using a Composite Symmetric Tensor

The composite symmetric tensor profile method node serves as a placeholder for separate nodes.

Each node represents a component of a symmetric tensor: three in two-dimensional problems (XX, XY &YX and YY) or six in three-dimensional problems (XX, XY & YX, XZ & ZX, YY, YZ & ZY, and ZZ). The component

nodes function just like the scalar profile quantity nodes in a typical composite vector profile method. This feature lets you specify profile methods for each component separately.

To do this, open the **Composite Symmetric Tensor** node and work with the scalar profile quantities as you did with the profile node.

For example, to specify the tensor

$$\begin{bmatrix} 10 & 15 \\ 15 & 20 \end{bmatrix}$$

set the **XX Component** to a *Constant* value of `10`, the **XY & YX Component** to `15`, and the **YY Component** to `20`.

Properties of a Composite Symmetric Tensor

Coordinate System	The coordinate system in which the tensor components are defined.

Setting Values Using a Composite Tensor

The composite tensor profile method node serves as a placeholder for separate nodes.

Each node represents a component of a tensor: four in two-dimensional problems (XX, XY, YX, and YY) or nine in three-dimensional problems (XX, XY, XZ, YX, YY, YZ, ZX, ZY, and ZZ). The component nodes function just like the scalar profile quantity nodes in a typical composite vector profile method. This feature lets you specify profile methods for each component separately.

To do this, open the **Composite Tensor** node and work with the scalar profile quantities as you did with the profile node.

For example, to specify the tensor

$$\begin{bmatrix} 10 & 15 \\ 25 & 20 \end{bmatrix}$$

set the **XX Component** to a *Constant* value of `10`, the **XY Component** to `15`, the **YX Component** to `25`, and the **YY Component** to `20`.

Properties of a Composite Tensor

Coordinate System	The coordinate system in which the tensor components are defined.

Setting Values Using an Isotropic Tensor

In the isotropic tensor profile method, a tensor is specified using a single component which remains invariant in all the spatial directions. The node serves as a placeholder for the isotropic component.

The isotropic component represents the single invariant component of the tensor. This functions just like the scalar profile quantity nodes in a typical composite vector profile method.

For example, to specify the tensor

$$\begin{bmatrix} 20 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 20 \end{bmatrix}$$

specify the **Isotropic Component** to a *Constant* value of `20`.

Setting Values Using a Principal Tensor

In the principal tensor profile method, a tensor is specified using its diagonal (principal) components with respect to its principal axes. The node serves as a placeholder for separate component and axis nodes.

The component nodes represent the diagonal components of a tensor: two in two-dimensional problems (XX and YY) or three in three-dimensional problems (XX, YY and ZZ). These nodes function just like the scalar profile quantity nodes in a typical composite vector profile method.

The axis nodes represent the directions of the principal axes of the tensor. Only two principal axes need to be specified (XX and YY), which must not be parallel to each other. The third principal axis is deduced internally as being orthogonal to the other two. The magnitude of the axis vectors is ignored; only their directions are important. If the entered axes are parallel to each other or are zero length (the degenerate condition), the principal axes fall back to the axes of the standard coordinate system. The axis nodes function just like vector profile quantity nodes, including the ability to specify a coordinate system for each axis vector. The coordinate system may often be the same for both axes but need not be.

The design allows you to specify profile methods for each component and axis separately.

To do this, open the **Principal Tensor** node and work with the scalar and vector profile quantities as you did with the profile node.

For example, to specify the tensor

$$\begin{bmatrix} 20 & 0 & 0 \\ 0 & 30 & 0 \\ 0 & 0 & 30 \end{bmatrix}$$

set the **XX Component** to a *Constant* value of `20`, and the **YY Component** to `30`. Then specify the **XX Axis** as `[1, 0, 0]` and the **YY Axis** to `[0, 1, 0]`.

Complex Quantities

In STAR-CCM+, you use complex profiles to specify quantities that are defined by complex numbers.

For example, when Maxwell's equations are expressed in the frequency domain, the electromagnetic fields and material properties, such as the electrical conductivity, become complex quantities. See Harmonic Time Dependence.

In general, you can write a complex number $\mathbf{A}$ in terms of its real and imaginary parts:

$$\mathbf{A} = \mathrm{Re}[\mathbf{A}] + i\mathrm{Im}[\mathbf{A}] \tag{1}$$

where i is the imaginary unit, defined by $i^2 = -1$.

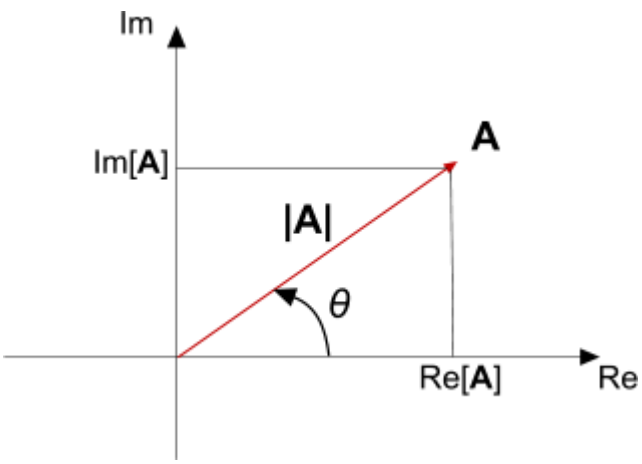
You can also write $\mathbf{A}$ in polar form:

$$\mathbf{A} = |\mathbf{A}| \cdot e^{i\theta} = |\mathbf{A}|(\cos\theta + i\sin\theta) \tag{2}$$

where $|\mathbf{A}|$ is the magnitude:

$$|\mathbf{A}| = \sqrt{(\mathrm{Re}[\mathbf{A}])^2 + (\mathrm{Im}[\mathbf{A}])^2} \tag{3}$$

and θ is the phase.



Complex Profile Methods

STAR-CCM+ allows you to define complex profiles using the following methods:

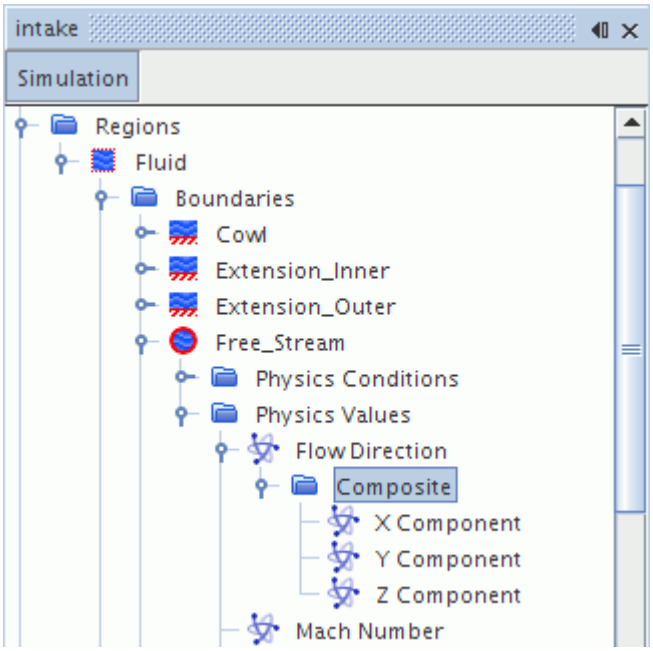
Method	Associated Profile Child Nodes
Composite Rectangular Allows you to specify the complex profile in terms of its real and imaginary parts (see Eqn. (1)).	**Composite Rectangular > Real** Specifies the real part as a scalar or vector profile. **Composite Rectangular > Imaginary** Specifies the imaginary part as a scalar or vector profile.
Composite Polar Allows you to specify the complex profile in polar form (see Eqn. (2)).	**Composite Polar > Magnitude** Specifies the magnitude as a scalar or vector profile. **Composite Polar > Phase** Specifies the phase as a scalar or vector profile.

When you define a complex profile, STAR-CCM+ creates four field functions to represent its real part, imaginary part, magnitude, and phase.

Setting Values Using the Composite Vector Profile Method

The composite vector profile method differs from the others, as its node serves as a placeholder for the scalar profile quantity nodes. This feature lets you specify profile methods for X, Y, Z components separately.

To do this, open the **Composite** node and work with the scalar profile quantities as you did with the profile node.



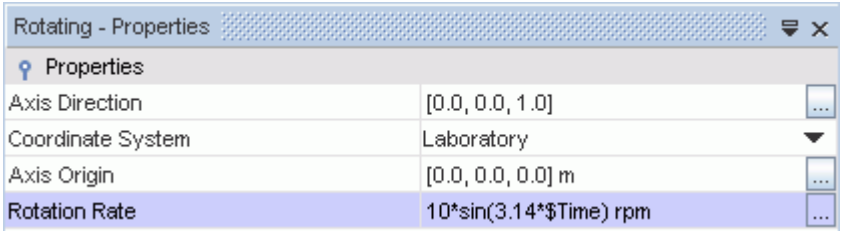
Setting Values Using an Expression

In addition to constants, you can enter expressions that use field function syntax, such as `10 * sin(3.14 * $Time)`, directly. This is an easy alternative to creating a macro for calculations of non-constant physical values.

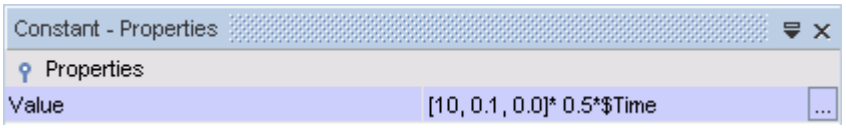
Examples of applications include:

- Valve openings or closings
- Tank sloshing
- A vehicle starting at rest and accelerating to full speed
- Prescribed rigid body motion
- A varying gravity vector
- A solar radiation vector

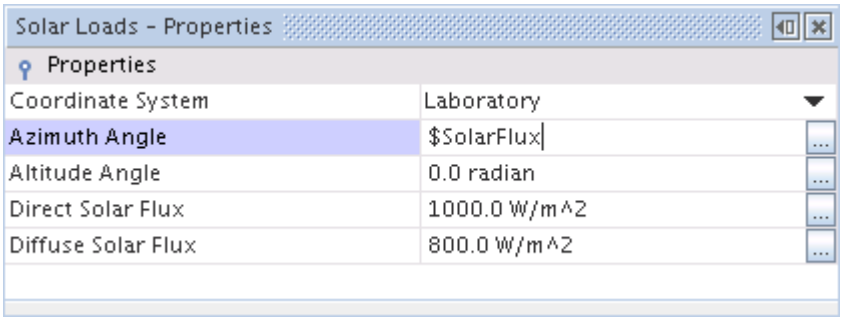
Simply type the function definition into the property value as you would a constant. The example in the following screenshot specifies oscillation with a period of 2 seconds.



This technique can also be used for vector entries.



Alternatively, you can create a field function that uses the expression, and then copy the name of the field function into the property.



The name comes from the *FunctionName* property of the field function. When you enter it into a constant value property, prefix it with `$`. In the preceding example, the function name of the field function is `SolarFlux`.

At the same time, other objects in the simulation can reference these field functions.

The field functions that can be used to define the expressions cannot be a function of space; they have to evaluate to a single number as the expressions are constants in space. Variables such as `$Time`, `$TimeStep` and `$Iteration`, as well as the field functions associated with reports, are valid. Variables such as `$Temperature` or `$Pressure` have different values at every different point in space and cannot be used in expressions.

Using Units

You can include units in these expressions, for example:

- `$Time - <1 min>`
- `<180 deg>`
- `<65 mph>`

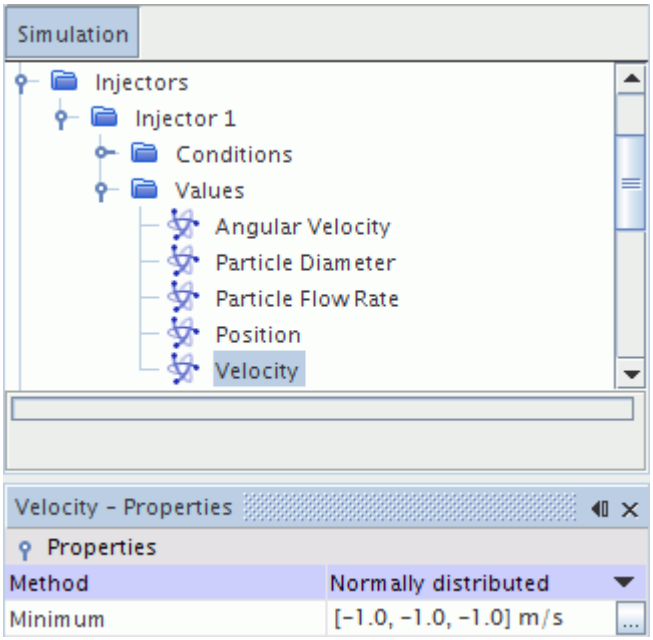
See Also:

Using the Properties Window

Setting Values Using Distributions

Set the value of a scalar or vector randomly using a uniform or a normal distribution.

In the *Properties* window, select the distribution method for the profile and enter the appropriate values for the distribution.



In a uniform distribution, each possible value has the same probability of being set, $\frac{1}{n}$where n is the number of possible values.

In a normal distribution, each possible value x has a probability of being set of $p(x)$

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{4}$$

where σ is the standard deviation and μ is the mean value.

Properties of a Uniformly Distributed Scalar Profile

Minimum	Specifies the minimum value of the scalar.
Maximum	Specifies the maximum values of the scalar.

Properties of a Uniformly Distributed Vector Profile

Minimum	Specifies the minimum values of the x, y, and z components of the vector. These values are separated by commas in a text entry.
Maximum	Specifies the maximum values of the x, y, and z components of the vector. These values are separated by commas in a text entry.

Properties of a Normally Distributed Scalar Profile

Mean	Specifies the mean value of the scalar.

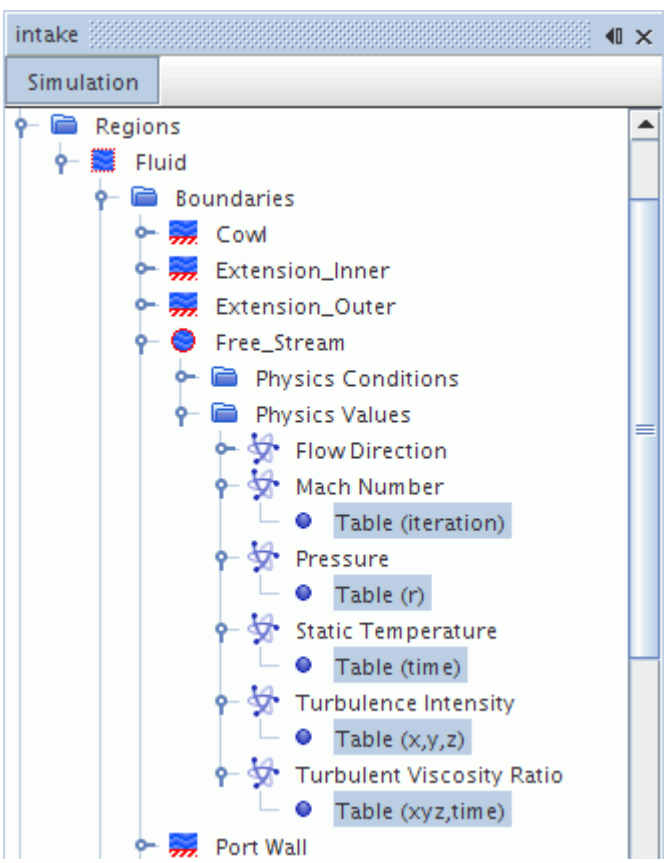
Deviation	Specifies the deviation of the scalar.
Minimum	Specifies the minimum value of the scalar.
Maximum	Specifies the maximum value of the scalar.

Properties of a Normally Distributed Vector Profile

Mean	Specifies the mean values of the x, y, and z components of the vector. These values are separated by commas in a text entry.
Deviation	Specifies the deviations of the x, y, and z components of the vector. These values are separated by commas in a text entry.
Minimum	Specifies the minimum values of the x, y, and z components of the vector. These values are separated by commas in a text entry.
Maximum	Specifies the maximum values of the x, y, and z components of the vector. These values are separated by commas in a text entry.

Setting Values Using a Table

There are several common types of tabular methods, each represented by nodes.



These types are:

- Table (iteration)
- Table (r)
- Table (time)
- Table (x,y,z)
- Table (xyz, time)
- Table (Equivalence Ratio)
- Table (particle size CDF)
- Tabular

The first three have similar properties. The properties of the Table (x,y,z) and the Table (xyz, time) are similar in that they are structured to accommodate the three coordinates. However, the Table (xyz, time) also takes simulation time into account.

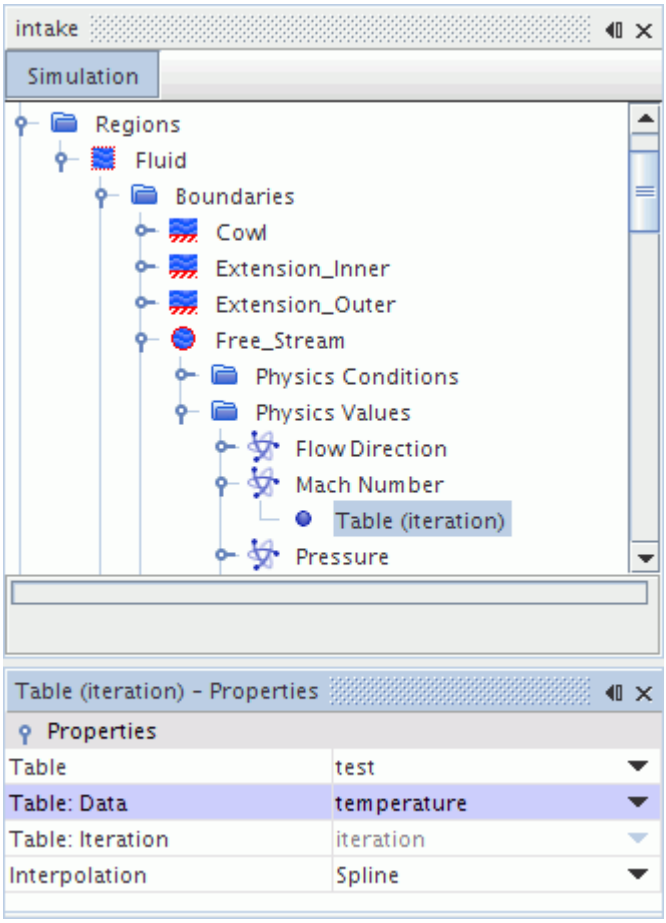
Additionally, there are two specific table types, Table (Equivalence Ratio), and Table (particle size CDF), which use the same properties as the first three types that are listed previously.

For all tabular methods, if the point lies inside the table range, the interpolation method specified by user is used to compute the function value. If, however, the point lies outside the table range, the function value is computed at the nearest data point in the table.

See File Format Reference.

Although a table data file is treated simply as a table when it is read into STAR-CCM+, each of these methods looks for a particular type of column in the table that accompanies the data values.

After specifying the method of a profile as a table, open the profile node (**Mach Number** in the following screenshot). If necessary, select the tabular node. Set the appropriate properties from the drop-down lists corresponding to *Table: Data* and *Table*. You must have read a table into STAR-CCM+ to make this selection; otherwise these drop-down lists are empty and the *Table: Data* and *Table* properties show as `Null`. The other property (*Table (iteration)* in the following screenshot), is read-only since it simply indicates the title of the column that the tabular method expects.



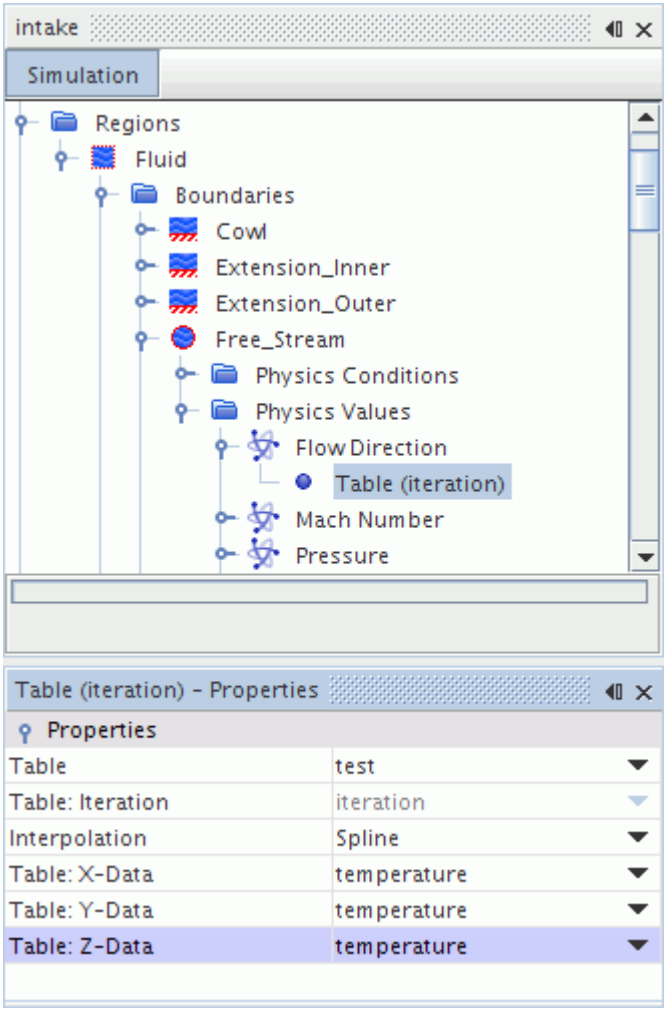
Separate table value nodes exist for setting vector profiles.

Contents:

Setting Vector Profiles Using a Table
What Is Table (iteration)?
What Is Table (r)?
What Is Table (time)?
What Is Table (x,y,z)?
What Is Table (xyz, time)?
What Is Table (Equivalence Ratio)?
Example of Creating a Table for a Profile
Example of Extracting and Exporting a Table
Example of Using Data with Sharp Discontinuities
Table Properties Iteration Radius and Time
Properties of Table (x,y,z) and Table (xyz, time)
Vector Table Properties Iteration Radius and Time
Properties of Vector Table (x,y,z) and Vector Table (xyz, time)

Setting Vector Profiles Using a Table

You can also use a single table method to specify a vector profile.



Separate *Data* properties exist for the x-, y-, and z-components. As is the case with the table methods for scalar and pressure profiles, properties are customized for iteration, radius, and time, and for (x,y,z) and (xyz, time).

What Is Table (iteration)?

An (iteration) table has a column of 'iteration' values that specify the iteration of each point (row) in the table.

The points must be ordered by increasing values of iteration. Multiple methods are available for interpolating between data points in the table, as described in the properties section. A profile in iteration could also be specified analytically with a user-defined field function that uses the $Iteration variable.

The following is a simple example of how an (iteration) table might look with actual data:

```
iteration,Value
5,0.5
40,1.0
100,1.5
```

What Is Table (r)?

An (r) table has a column of ‘R’ values that specify the radius, in default units, of each point (row) in the table.

The points must be ordered by increasing values of radius. A B-spline is used on the data to calculate face and cell values, requiring extra measures for data with sharp discontinuities.

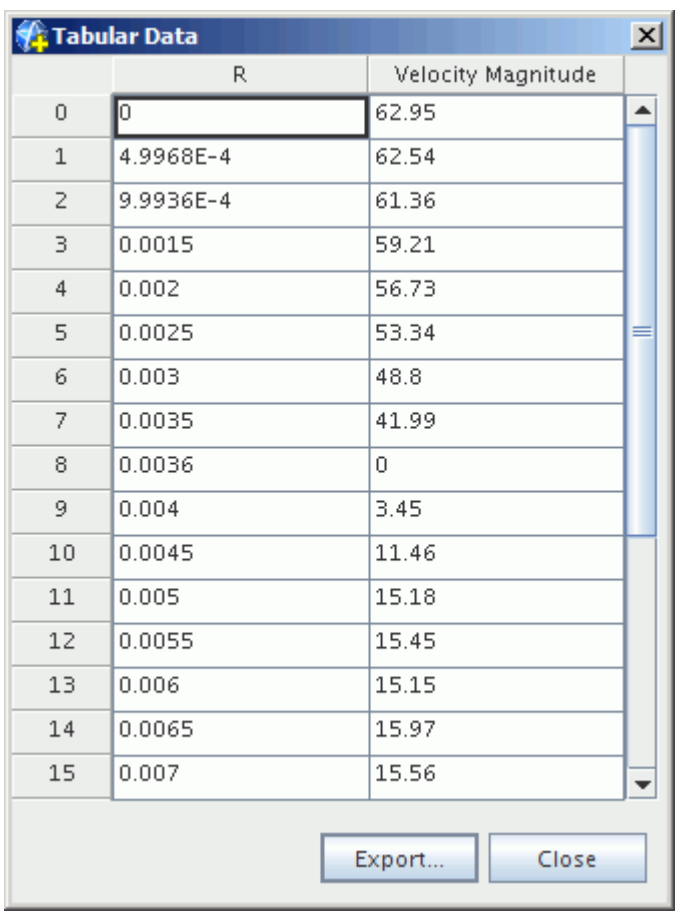
Radius is computed relative to the frame of reference of each region, which is specified in the region values. The axis from which the radius is computed is specified using the Axis node of the region.

If a Rotation or Rotation and Translation reference frame is specified under the **Motion Specification** node, the axis that is used to define that reference frame is used to calculate the radial coordinate for the region.

For example, to initialize velocity using an (r) table:

- Import the table.

  The tabular data can look as follows:



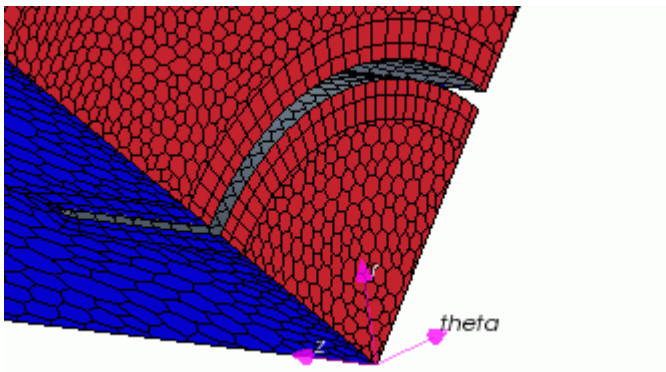
	R	Velocity Magnitude
0	0	62.95
1	4.9968E-4	62.54
2	9.9936E-4	61.36
3	0.0015	59.21
4	0.002	56.73
5	0.0025	53.34
6	0.003	48.8
7	0.0035	41.99
8	0.0036	0
9	0.004	3.45
10	0.0045	11.46
11	0.005	15.18
12	0.0055	15.45
13	0.006	15.15
14	0.0065	15.97
15	0.007	15.56

Specify the axis for the region that is used to define the radial coordinate:

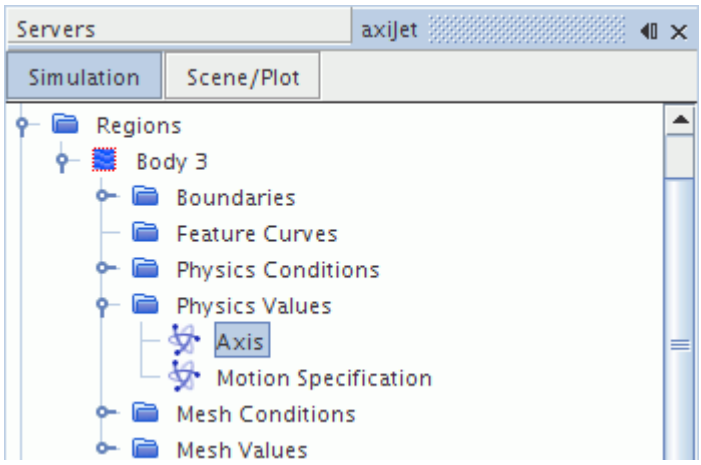
1. Select the **Regions** > **[Region]** > **Physics Values** > **Motion Specification** node and make sure that *Reference Frame* is set to **Lab Reference Frame**.

   The **Physics Values** > **Axis** node for the region appears only when the *Reference Frame* is set to **Lab Reference Frame**.

   In this example, a local cylindrical coordinate system is chosen to define the axis for the region. The *Direction* `[0.0, 0.0, 1.0]` in the **Laboratory-** > **Cylindrical 1** system is rooted at *Position* `[0.0, 0.0, 0.0]`, as shown in the following screenshot.

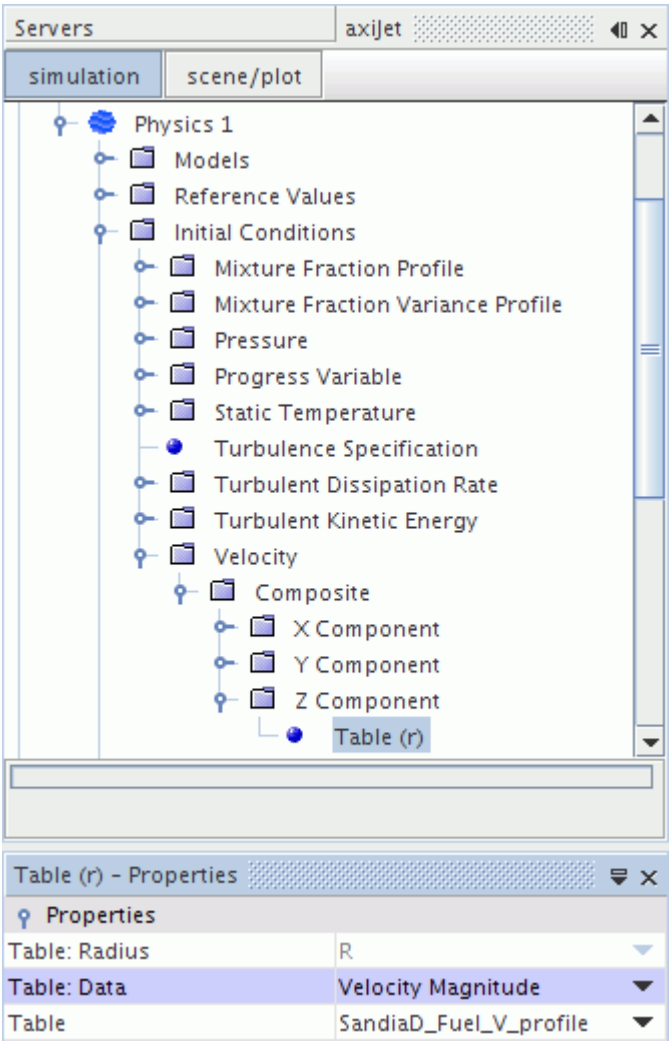


2. Select the **Regions** > **[Region]** > **Physics Values** > **Axis** node and set the *Coordinate System* property.



In this example, the initial jet velocity component is set using a composite specification. A composite specification allows you to specify each component of the velocity using a different method. To set a component using an imported (r) table:

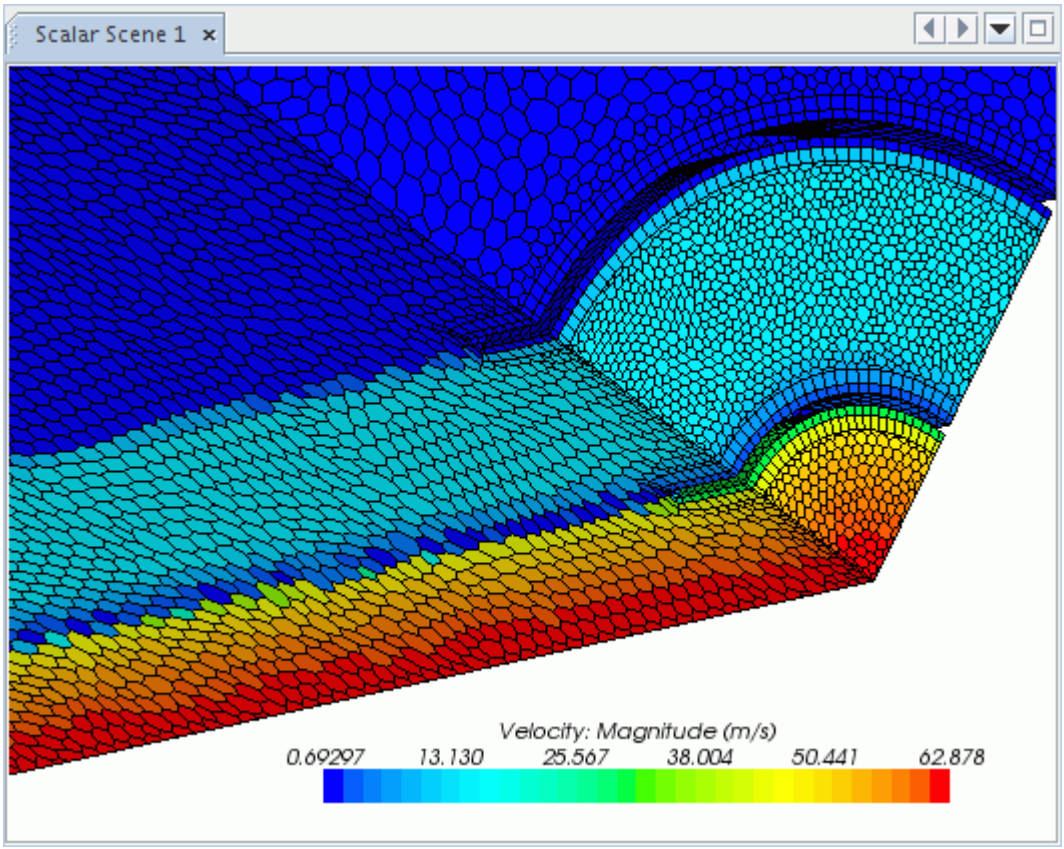
1. Select the **Continua** > **[Physics Continuum]** > **Initial Conditions** > **Velocity** node and set Method to **Composite**.
2. Select the **Velocity** > **Composite** > **[Component]** node and set *Method* to **Table (r)**.
3. Select the **[Component]** > **Table (r)** node and set the *Table* and *Table:Data* properties.



Similarly, to set the boundary condition:

1. Select the **Regions** > **[Region]** > **Boundaries** > **[Boundary]** > **Physics Values** > **Velocity Magnitude** node and set *Method* to **Table (r)**.
2. Select the **Velocity Magnitude** > **Table (r)** node and set the *Table* and *Table:Data* properties.

The initialized velocity for the flow field in this example looks as follows.



What Is Table (time)?

A (time) table has a column of 'time' values that specify the physical time, in seconds, of each point (row) in the table.

The points must be ordered by increasing values of time. Multiple methods are available for interpolating between data points in the table, as described in the properties section. This type of table is only applicable to unsteady calculations where "time" is defined.

Note: A profile in time can also be specified analytically with a user-defined field function that uses the `$Time` variable.

What Is Table (x,y,z)?

An (x,y,z) table has a column of 'X' values, 'Y' values and, in three-dimensional cases, 'Z' values that specify the position, in default units, of each point (row) in the table:

```
X Y Z VAR1 ... VARn
```

where, for each column header, `VARn` is the name of a particular variable.

The points are considered unordered in the file, and no interpolation is done to calculate face and cell values. Each face or cell is assigned the value from the closest point in the table.

What Is Table (xyz, time)?

Similar to the (x,y,z) table, an (xyz, time) table has a column of 'X' values, 'Y' values and, in three-dimensional cases, 'Z' values that specify the position, in default units, of each point (row) in the table. The points are considered unordered in the file.

In addition, this method causes simulation time to be considered when STAR-CCM+ reads the table. The data columns would appear as follows:

```
X Y Z VAR1[t=T1] VAR1[t=T2] VAR1[t=T3] VAR2[t=TT1] VAR2[t=TT2]
```

where, for each column header, `VARn` is the name of a particular variable and `x` in `[t=x]` is the time stamp for the column. The file that is shown above contains two time series, `VAR1` and `VAR2`. `VAR1` has data points for all X,Y,Z at times `T1`, `T2` and `T3`, and `VAR2` has data points for times `TT1` and `TT2`. (The values of `Tn` and `TTn` for each `n` can be different.)

The following is a simple example of how an (xyz, time) table might look with actual data:

```
X Y Z Temperature(K)[t=1s] Temperature(K)[t=2s] Temperature(K)[t=3s]
```

```
1 2 3 300 400 500
```

In the STAR-CCM+ GUI, the time stamps are removed and you are able to select a variable name for the data property in the method. As the simulation iterates, the profile is filled with values from the table based on the time, using the interpolation method that is set for the *Interpolation* property.

What Is Table (Equivalence Ratio)?

An equivalence ratio table has a column of 'er' (equivalence ratio) values, and 'lfs' (laminar flame speed) values given in m/s. This method is available with the *User Defined Laminar Flame Speed* option of the Coherent Flame model, Turbulent Flame Speed Closure model, or Thickened Flame model.

When defining the equivalence ratio table, the first column, which contains the values for the equivalence ratio, must be labeled `er`. However, the second column, which contains the laminar flame speed values, can be labeled with any title.

This example shows the contents of a file with equivalence ratio data:

```
er      lfs
0.6     0.146586604
0.7     0.231662295
0.8     0.329079651
0.9     0.42048291
1       0.483519189
1.1     0.500554535
1.2     0.466632794
1.3     0.391806544
1.4     0.296351954
1.5     0.201946692
1.6     0.12399419
```

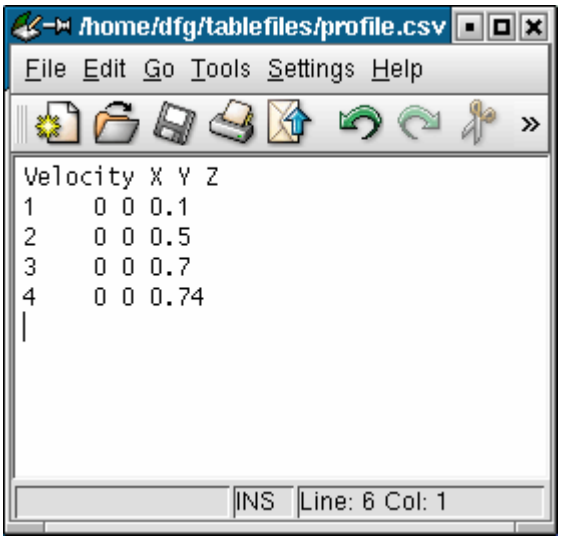
```
1.7     0.068601963
1.8     0.03420356
1.9     0.015368442
```

Example of Creating a Table for a Profile

This section illustrates how you would go about creating a table for a profile.

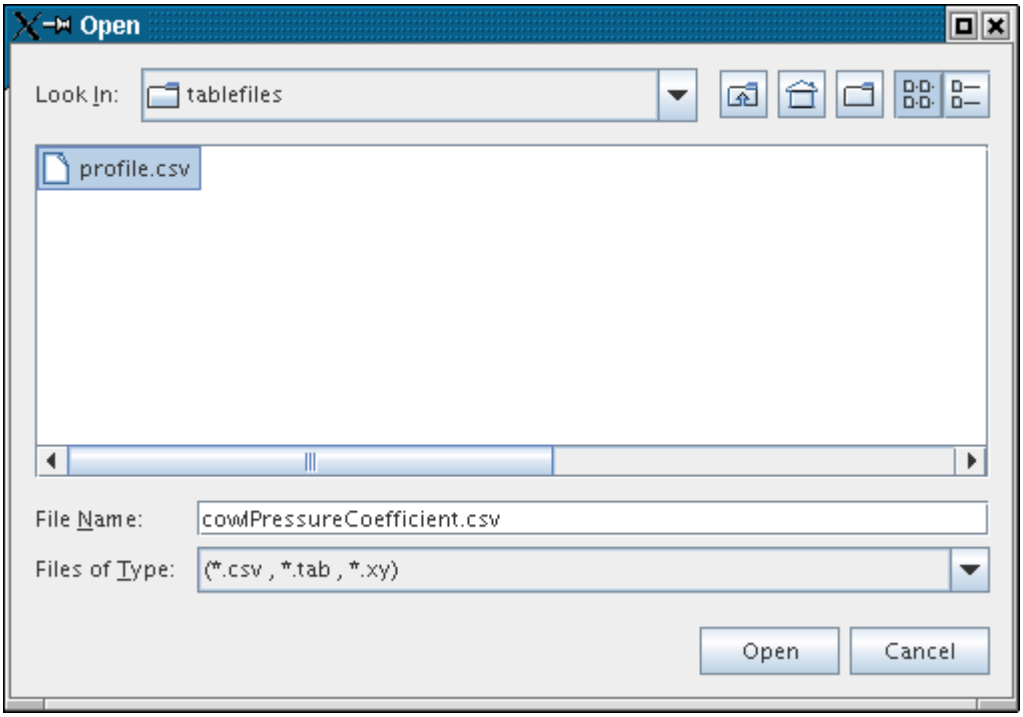
Suppose that you must specify a velocity inlet as a function of Z. If you have a few points of data, you can create a table and use it to specify the inlet.

First, create, and save your table file in a typical text editor, specifying the .csv format as the file extension.

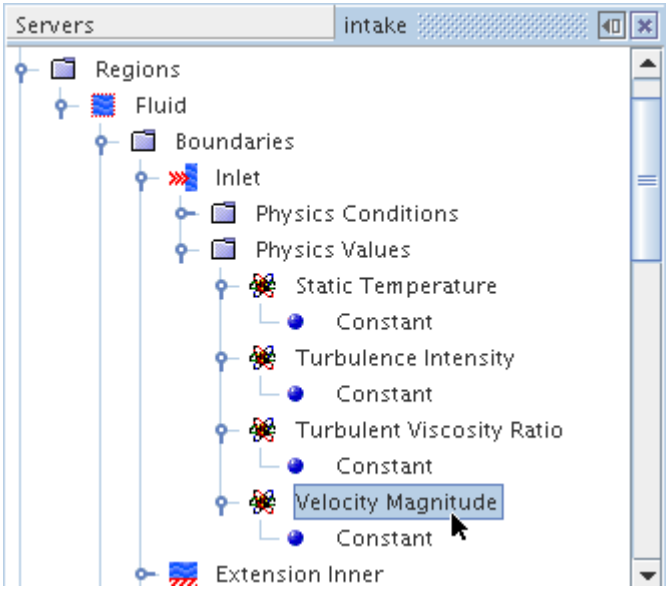


Even if you only want calculations in relation to one direction, you need columns that correspond to the two others (or the other in a two-dimensional case). Hence “empty” columns must contain entries of zero.

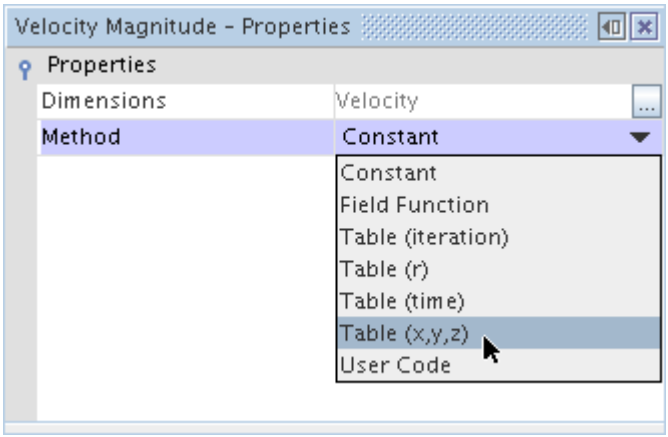
Next, read the table file into STAR-CCM+ using the **Tools** > **Tables** node. In this example, the table file is named `profile.csv`.



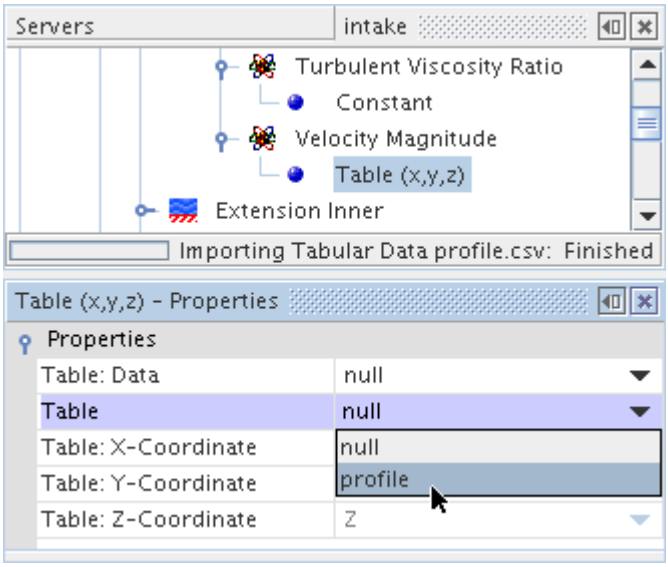
Open the **Regions** > **[individual region node]** > **Boundaries** > **[individual boundary node]** > **Physics Values** nodes, and select the profile node (**Velocity Magnitude** in the following screenshot).



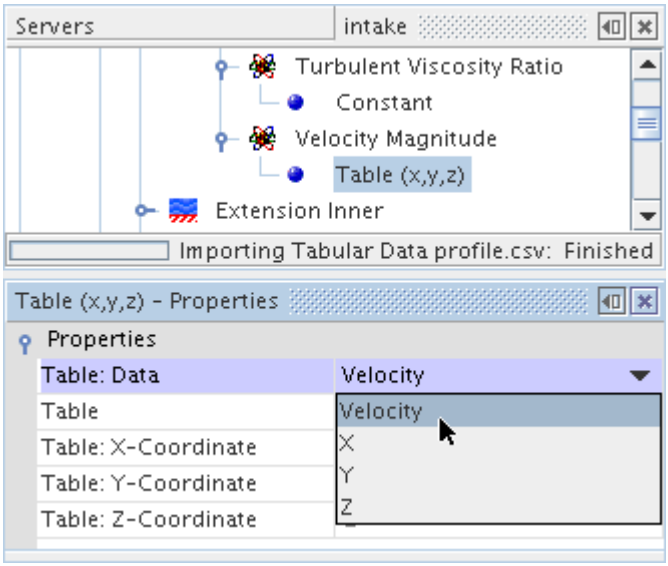
Set the *Method* property to a tabular method (`Table (x,y,z)` in the following screenshot).



As a result of this action, the method node inside the profile node changes to a tabular method node. Its label matches the selection in the drop-down list, in this case **Table (x,y,z)**. Select that node and set the *Table* property to the name of your table file. In this example, the name is `profile`.



Set the *Table:Data* property to the column of the scalar value, in this case `Velocity`.



When you use `Table (x,y,z)` as the tabular method, once you set the *Table* and *Table:Data* properties, the other properties for the X, Y, and Z coordinates are read-only, since they map automatically to the columns in your table file with those headings.

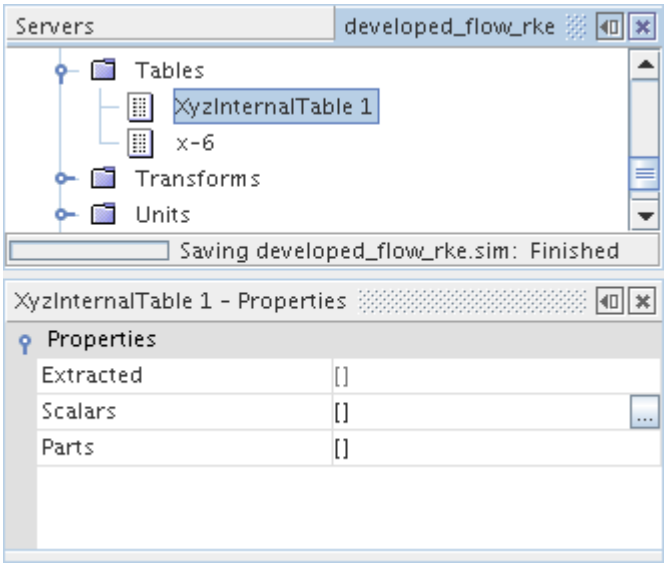
This completes the example of setting a profile with tables.

Example of Extracting and Exporting a Table

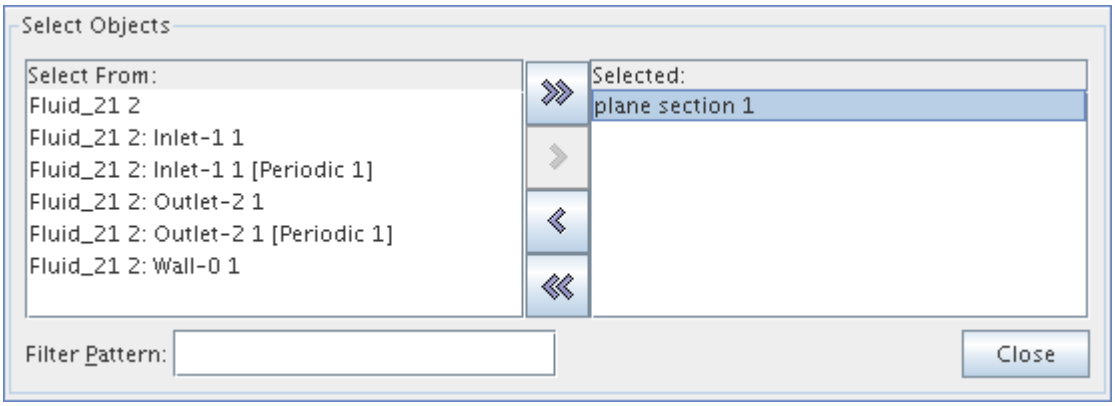
This section illustrates how you would extract a table from one simulation to use as a boundary condition in another.

In this example, a channel flow simulation with periodic boundaries is used to create a fully developed channel flow profile. Values from an x-y plane cut across the fully developed channel are extracted into a table, exported and then used at the inlet of another simulation. This obviates the need to create a long inlet section in the second simulation to obtain the required fully developed conditions.

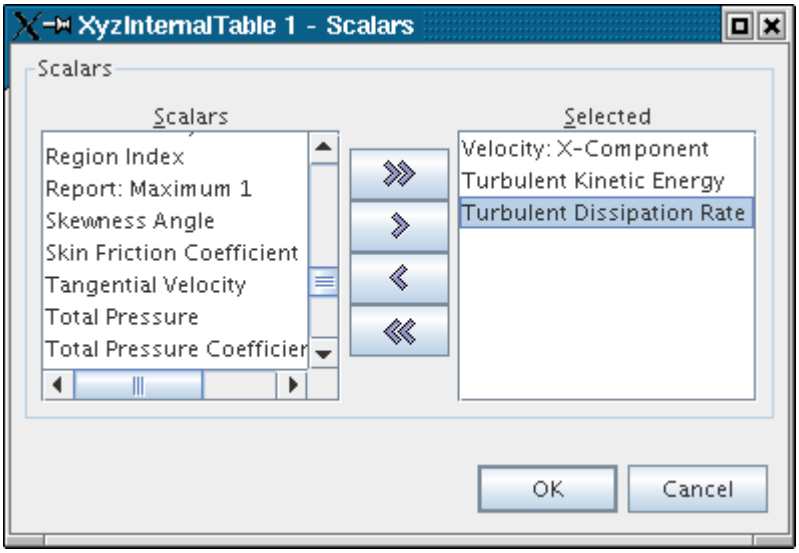
Create an xyz internal table. Make sure that the node of the new table is selected.



In the properties, set the *Parts* property by clicking the right half to activate the in-place object selector. In this example, `plane section 1` is used.

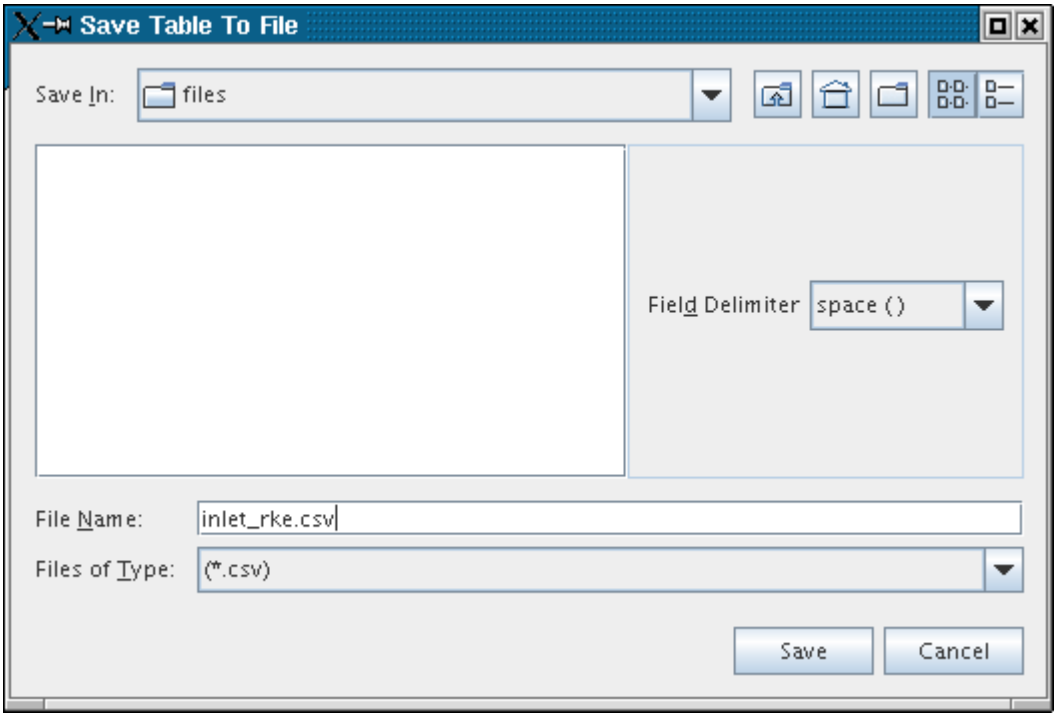


Once you select the part, set the *Scalars* property. While the property allows text entries, it is much more convenient to use the property customizer ( ), which activates a scalars dialog.



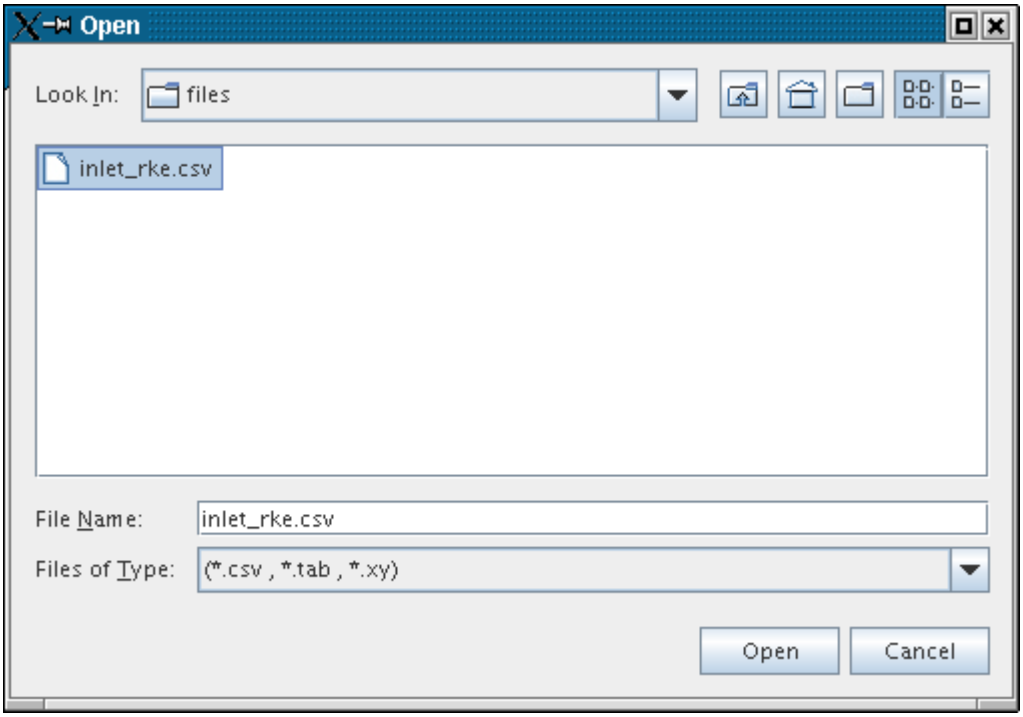
This dialog works the same as the one you used for the part selection. In this example, `Velocity: X-Component`, `Turbulent Kinetic Energy`, and `Turbulent Dissipation Rate` are selected.

Extract the table data, then export the data to a `.csv` file.



In this example, the table is named `inlet_rke.csv`.

Open the simulation in which you want to use the profile. Then read the table file that you created in the other simulation.

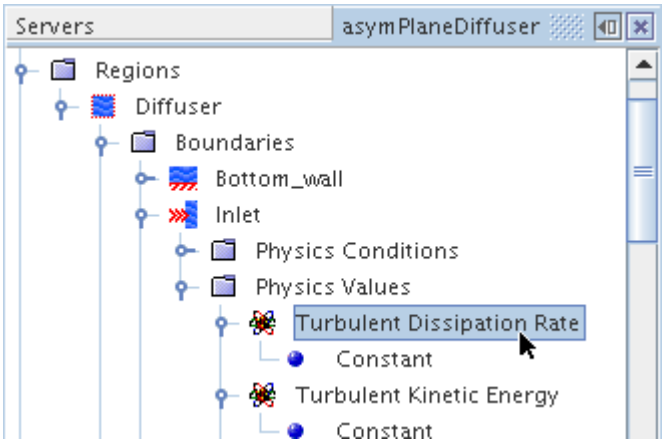


In this example, the profiles that receive the data of this file are:

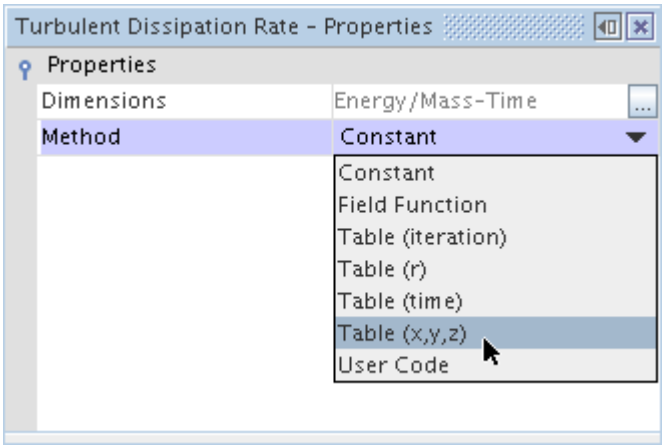
- Turbulent Dissipation Rate
- Turbulent Kinetic Energy
- Velocity: X-Component

They correspond to the scalars selected for the xyz table that was created earlier.

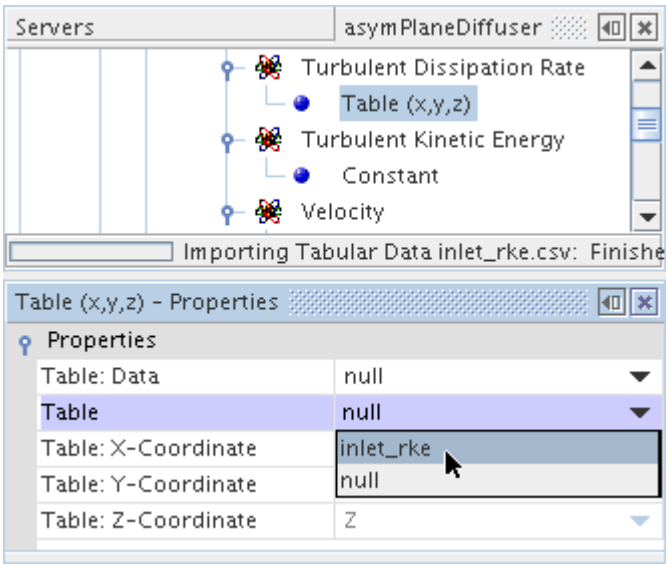
Open the **Regions** > **[individual region node]** > **Boundaries** > **[individual boundary node]** > **Physics Values** nodes, and select the profile node (**Turbulent Dissipation Rate** in the following screenshot).



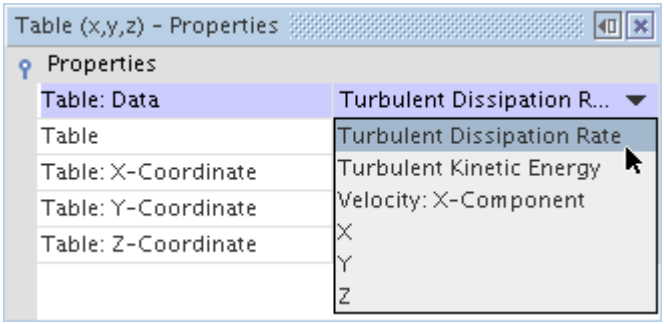
Set the *Method* property to a tabular method (`Table (x,y,z)` in the following screenshot).



As a result of this action, the method node inside the profile node changes to a tabular method node. Its label matches the selection in the drop-down list, in this case **Table (x,y,z)**. Select that node and set the *Table* property to the name of your table file. In this example, the name is `inlet_rke`.

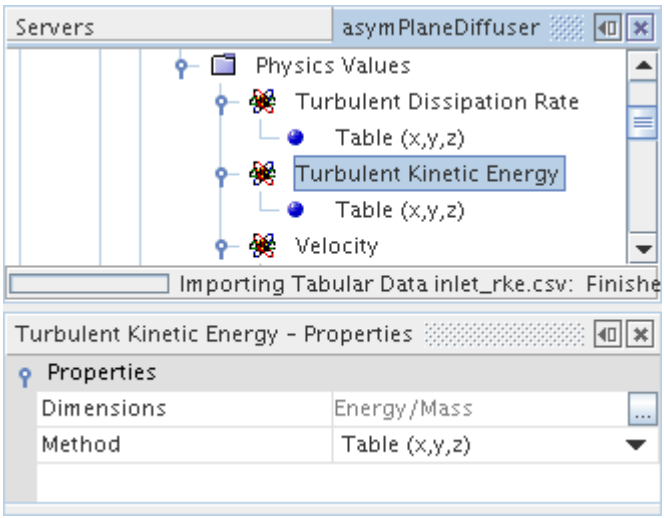


Set the *Table:Data* property to the column of the scalar value, in this case `Turbulent Dissipation Rate`.

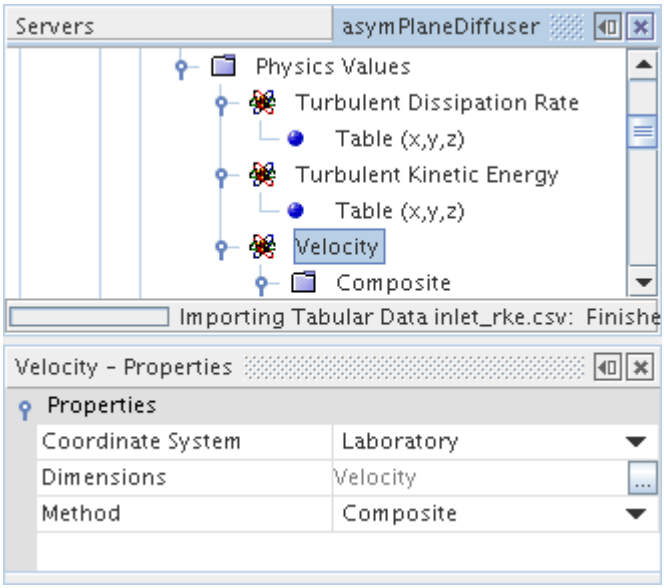


When you use `Table (x,y,z)` as the tabular method, once you set the *Table* and *Table:Data* properties, the other properties for the X, Y, and Z coordinates are read-only, since they map automatically to the columns in your table file with those headings.

In this example, in the same **Physics Values** node, these steps are repeated for **Turbulent Kinetic Energy**, its node depicted below.

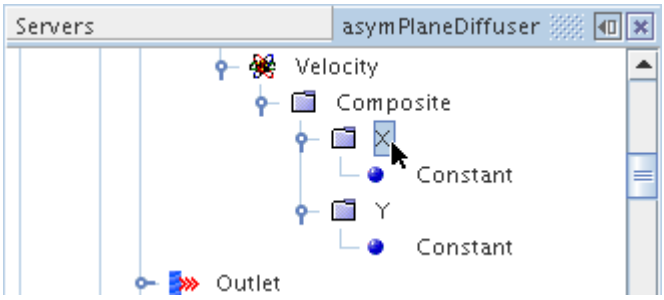


In the same **Physics Values** node, a profile node for velocity is selected.



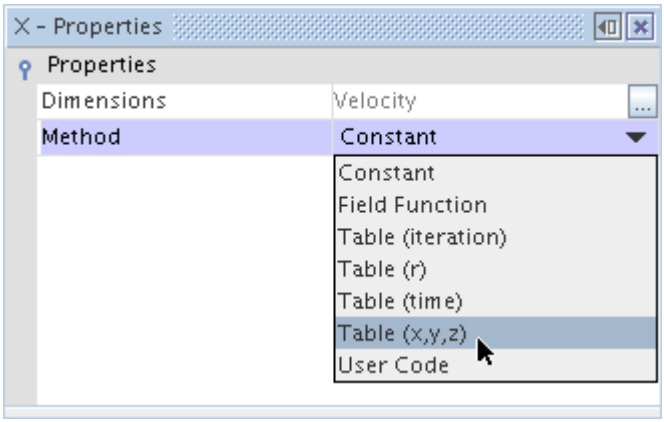
In this example, the velocity specification boundary condition has been set to `Components`, since only the X-component is being used in the table file. Hence, for this **Velocity** node, the *Method* property must be set to `Composite`, to use the composite vector profile method.

The next step, therefore, is to open the **Velocity** and **Composite** nodes. To use this approach, select the node of the components for which you want to plot data.



In this example, the X-component is selected, while the Y-component is a constant. (There is no Z-component because this case is two-dimensional.)

Set the *Method* property (`Table (x,y,z)` in the following screenshot).



As before, set the *Table* property to the name of your table file. Then set the *Table:Data* property to the column of the scalar value, in this example it is `Velocity X-Component`.

This completes the example of using a table file to transfer boundary conditions from one simulation to another.

Example of Using Data with Sharp Discontinuities

In earlier versions of STAR-CCM+, it was not possible to choose the Step interpolation option for tables based on iteration and time; all interpolation was based on a B-spline approach.

The B-spline approach is still used for tables based on a radial profile, and so some care must be taken to deal with any sharp discontinuities across the radial profile.

- The essential approach is to add additional data points before and after the sharp discontinuity so that STAR-CCM+ can more accurately interpolate data in that interval.
- It is also possible to generate data with sharp discontinuities by using a user-defined field function, and so avoiding the use of a table altogether.

```
($Time < t) ? 0 : 1e-4
```

Table Properties Iteration Radius and Time

Table: Data	Identifies the selected column in the table file as the data column.	
Table: [heading]	Looks for the column in the table with the heading `Iteration`, `R`, or `time` (read-only).	
Table	Selects a table from among the ones that are represented by nodes in the **Tables** node of the **Tools** node.	
Interpolation	(Iteration and Time table types). Selects the method that is used to interpolate between data points in the table.	
	Spline	Uses B-spline interpolation between table data values. This method is set automatically for Radial profiles.
	Step	Uses the table data value corresponding to the highest iteration or time that is less than or equal to the current iteration or time.
	Linear	Specifies a linear interpolation between table data values.

Properties of Table (x,y,z) and Table (xyz, time)

Table: X-Coordinate	Looks for the column in the table with the heading `X` (read-only).	
Table: Y-Coordinate	Looks for the column in the table with the heading `Y` (read-only).	
Table: Z-Coordinate	Looks for the column in the table with the heading `Z` (read-only).	
Coordinate System	Specifies the coordinate system (laboratory by default). The drop-down list of this property includes any local coordinate systems that exist in the simulation.	
Table: Data	Identifies the selected column in the table file as the data column.	
Table	Selects a table from among the ones that are represented by nodes in the **Tables** node of the **Tools** node.	

Expert Properties (Table (xyz, time) only)

Interpolation	Selects the method that is used to interpolate between data points in the table.

	Spline	Uses B-spline interpolation between table data values.
	Step	Uses the table data value corresponding to the highest time that is less than or equal to the current time.
	Linear	Specifies a linear interpolation between table data values.

Vector Table Properties Iteration Radius and Time

Table: [heading]	Looks for the column in the table with the heading `Iteration`, `R`, or `time` (read-only).	
Table	Selects a table from among the ones that are represented by nodes in the **Tables** node of the **Tools** node.	
Table: X-Data	Identifies the selected column in the table file as the data column for the x-component.	
Table: Y-Data	Identifies the selected column in the table file as the data column for the y-component.	
Table: Z-Data	Identifies the selected column in the table file as the data column for the z-component.	
Interpolation	(Iteration and Time table types). Selects the method that is used to interpolate between data points in the table.	
	Spline	Uses B-spline interpolation between table data values. This method is set automatically for radial profiles.
	Step	Uses the table data value corresponding to the highest iteration or time that is less than or equal to the current iteration or time.
	Linear	Specifies a linear interpolation between table data values.

Properties of Vector Table (x,y,z) and Vector Table (xyz, time)

Table: X-Coordinate	Looks for the column in the table with the heading `X` (read-only).	

Table: Y-Coordinate	Looks for the column in the table with the heading `Y` (read-only).
Table: Z-Coordinate	Looks for the column in the table with the heading `Z` (read-only).
Coordinate System	Specifies the coordinate system (laboratory by default). The drop-down list of this property includes any local coordinate systems that exist in the simulation.
Table: X-Data	Identifies the selected column in the table file as the data column for the x-component.
Table: Y-Data	Identifies the selected column in the table file as the data column for the y-component.
Table: Z-Data	Identifies the selected column in the table file as the data column for the z-component.
Table	Selects a table from among the ones that are represented by nodes in the **Tables** node of the **Tools** node.

Contents:

Expert Properties (Vector Table (xyz, time) only)

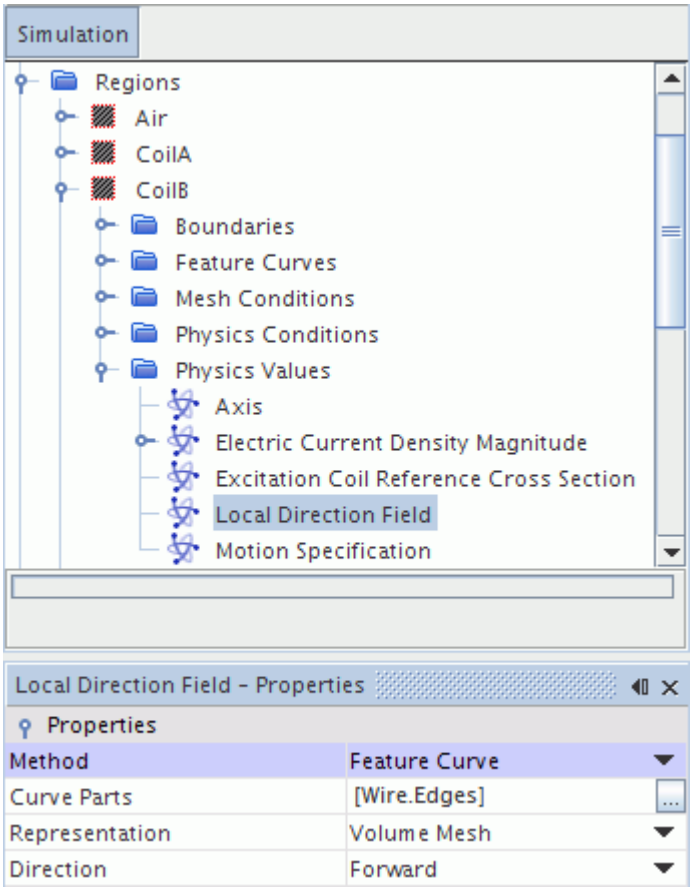
Expert Properties (Vector Table (xyz, time) only)

Interpolation	Selects the method that is used to interpolate between data points in the table.	
	Spline	Uses B-spline interpolation between table data values.
	Step	Uses the table data value corresponding to the highest time that is less than or equal to the current time.
	Linear	Specifies a linear interpolation between table data values.

Setting Values Using a Field of Direction Vectors

Use this method to specify a field of direction vectors using line curves or surfaces.

In the *Properties* window, set the method of the profile to **Feature Curve** and set the appropriate values for the other properties.



This method is similar to the **interpolateDirection** field function.

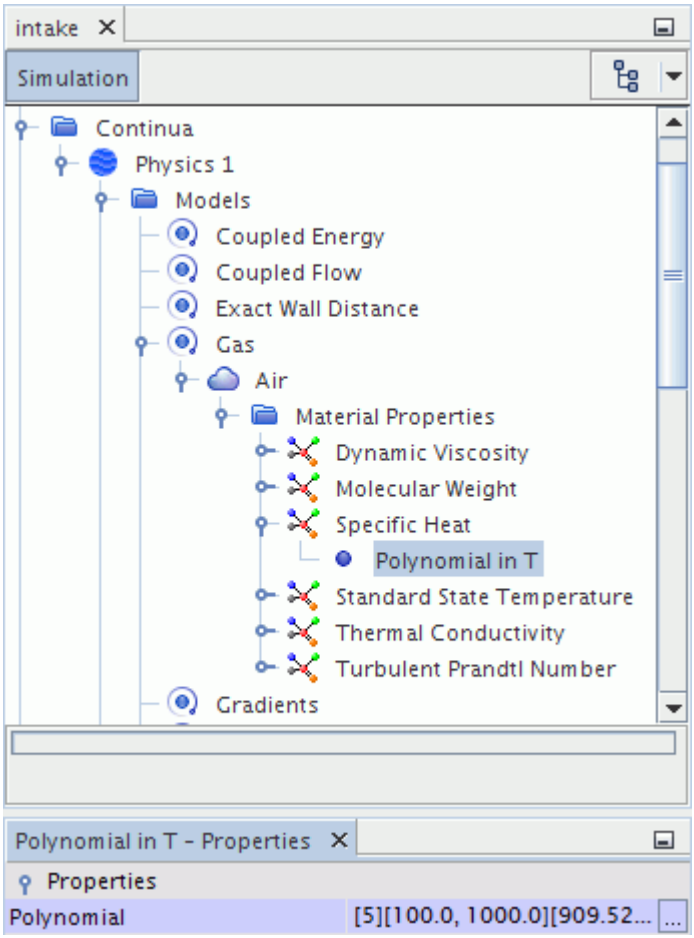
Properties of a Profile with a Feature Curve Method

Curve Parts	A surface or a line curve. For a surface, the direction field is given using the normals of that part. For a line curve, the direction field is given using the line segment directions.	
Representation	The representation that is used to extract the part.	
Direction	Specifies the orientation of the direction vectors.	
	Forward	Set the orientations of the vectors based on the direction of the normals of the input surface or the direction of the line segment of the input line.
	Backward	Reverses the direction vectors.

Setting Values Using a Polynomial

STAR-CCM+ allows a number of quantities to be specified using a polynomial.

Examples are density and C_p (illustrated below) as polynomial functions of temperature. The properties for a polynomial are set in a customizer dialog that is activated with the property customizer (...) of the *Polynomial* property.



A polynomial $y(x)$ is specified in piece-wise fashion where each interval i spans the range x_i to x_{i+1}, has m_i number of terms, and is of the form:

$$y(x) = \sum_{k=0}^{m_i} a_{i,k} \cdot x^{b_{i,k}};\ x_i \le x \le x_{i+1} \tag{5}$$

The polynomial has n intervals with interval ranges: $x_0, x_1, x_2, \ldots, x_{n+1}$

The number of terms in each interval are: $m_0, m_1, m_2, \ldots, m_n$

The exponents are:

$$\begin{matrix} b_{0,0}, & b_{0,1}, & b_{0,2}, \ldots & b_{0,m_0} \\ b_{1,0}, & b_{1,1}, & b_{1,2}, \ldots & b_{1,m_1} \\ \vdots & \vdots & \vdots & \vdots \\ b_{3,0}, & b_{3,1}, & b_{3,2}, \ldots & b_{3,m_3} \end{matrix}$$

The coefficients for the polynomial are:

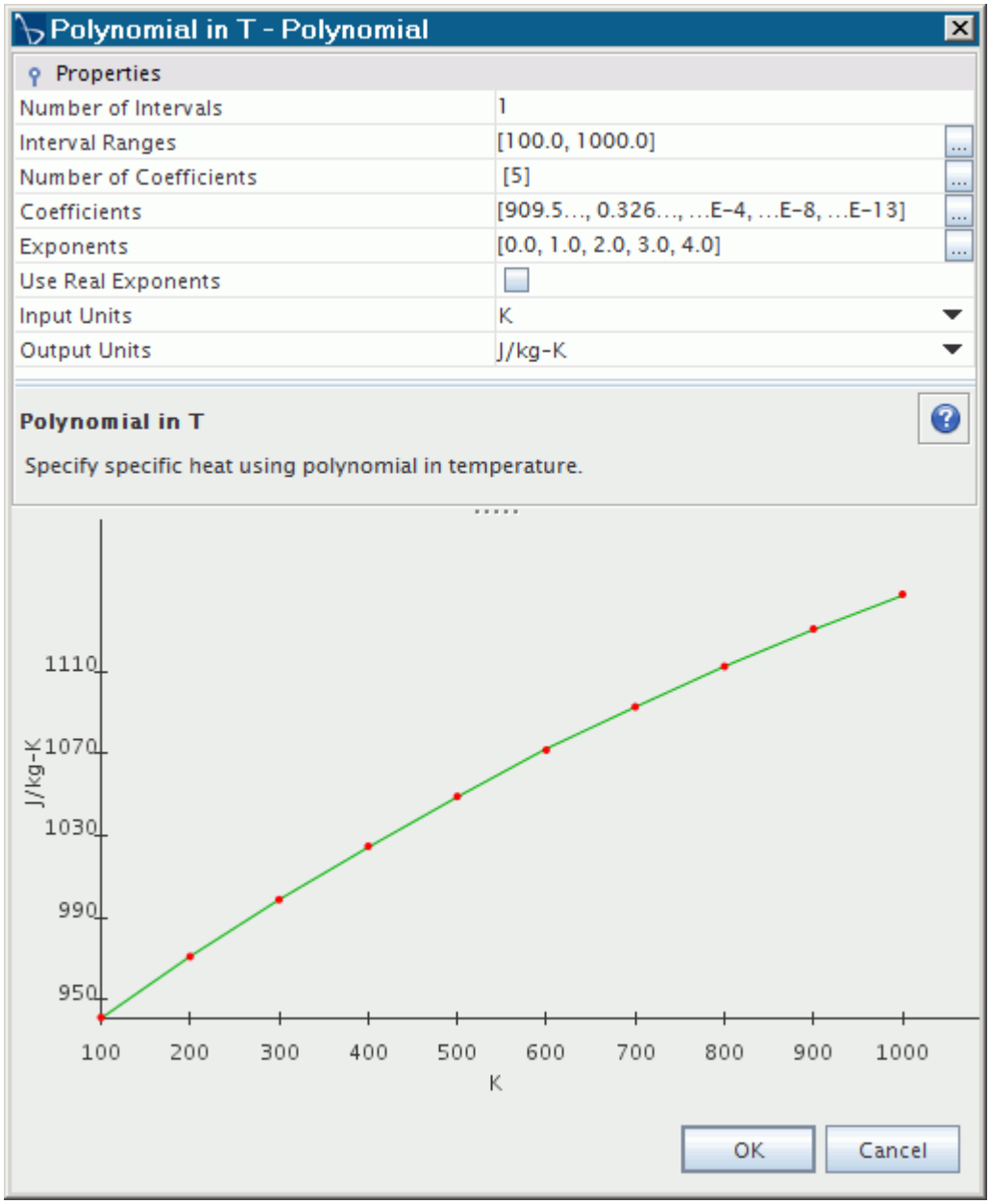
$$\begin{matrix} a_{0,0}, & a_{0,1}, & a_{0,2}, \cdots & a_{0,m_0} \\ a_{1,0}, & a_{1,1}, & a_{1,2}, \cdots & a_{1,m_1} \\ \vdots & \vdots & \vdots & \vdots \\ a_{3,0}, & a_{3,1}, & a_{3,2}, \cdots & a_{3,m_3} \end{matrix}$$

Specify a range of the polynomial sufficiently large such that the polynomial is not evaluated outside the range during the analysis. If the polynomial is ever evaluated outside the range, the property is assigned a value of the polynomial at the respective range limit.

Example of a Polynomial Workflow

This section illustrates how you would use a STAR-CCM+ polynomial: the Polynomial in T.

1. In the *Properties* window, a single property that is called *Polynomial* includes a property customizer (...). Click that button to activate a customizer dialog.



The plot in this dialog gets updated instantly with your changes.

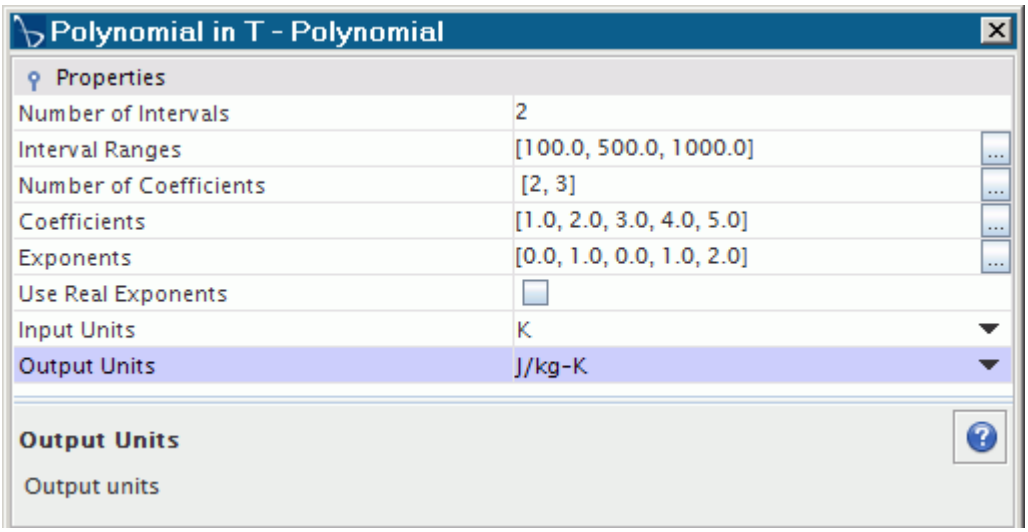
Suppose that you have the following, which is defined with two intervals:

$$C_p = 1 + 2T,\ 100 \leq T < 500$$

$$C_p = 3 + 4T + 5T^2,\ 500 \leq T < 1000$$

2. First specify the *Number of Intervals*, in this case `2`.
3. Enter the *Interval Ranges*, `100,500,1000`.
4. Specify the number of coefficients for each interval. In this example, the first interval has two coefficients and the second interval has three, so enter `2,3` in the *Number of Coefficients* property.
5. For the *Coefficients* property, specify all for each interval in succession. In this example, the entries are both coefficients for the first interval, `1,2`, followed by all three for the second, `3,4,5`.

6. Enter the *Exponents* using the same technique as for the coefficients, in this example `0,1,0,1,2`.
7. Decide whether to activate the *Use Real Exponents* option. When activated, this option lets you specify exponents other than integers, but it takes considerably more computing time.
8. Work with the units as needed. You can select units separately for each axis of the graph: *Input Units* for the x-axis, and *Output Units* for the y-axis.



9. Click **OK**. When you do so, STAR-CCM+ checks for consistency among the properties. This step completes the example of using polynomials.

Contents:

Polynomial Properties

Polynomial Properties

The properties in the following table are in a customizer dialog that is activated with the property customizer of the *Polynomial* property.

These properties correspond to the components of Eqn. (5). Properties that accommodate a series of entries have their own customizer buttons (...). When one of these buttons is clicked, it opens a text box with more space for entering values.

Number of Intervals	The number of intervals in the equation.
Interval Ranges	The minimum and maximum values that bound the intervals.
Number of Coefficients	The number of coefficients that appear in each interval in succession, and that are entered in the *Coefficients* property.
Coefficients	For the *Coefficients* property, the entries are separated by commas and the coefficients are in the following order: $$y(x) = a_0 + (a_1 \cdot x) + (a_2 \cdot x^2) + \ldots + (a_n \cdot x^n)$$

Exponents	The exponents that appear in the successive intervals.	
Use Real Exponents	Controls how polynomials are calculated.	
	Activated	You can enter any values for exponents. However, entering exponent values manually can increase computing time considerably, because when coefficients are arbitrary, fast polynomial evaluation cannot be used.
	Deactivated	The exponents cannot be edited; they are determined using the number of coefficients in each range (default).
Input Units	Specifies units for the x-axis. Unlike constant values, the polynomial coefficients and polynomial ranges (and therefore the x- and y-axes on the polynomial plot) do not rescale when you change the units. These unit selections serve as an indicator of the units that STAR-CCM+ uses when calculating quantities with this polynomial.	
Output Units	Specifies units for the y-axis.	

Copying Properties

The conditions and values for a region, boundary, or interface can be copied from a region, boundary, or interface of the identical type using the clipboard or by using drag-and-drop.

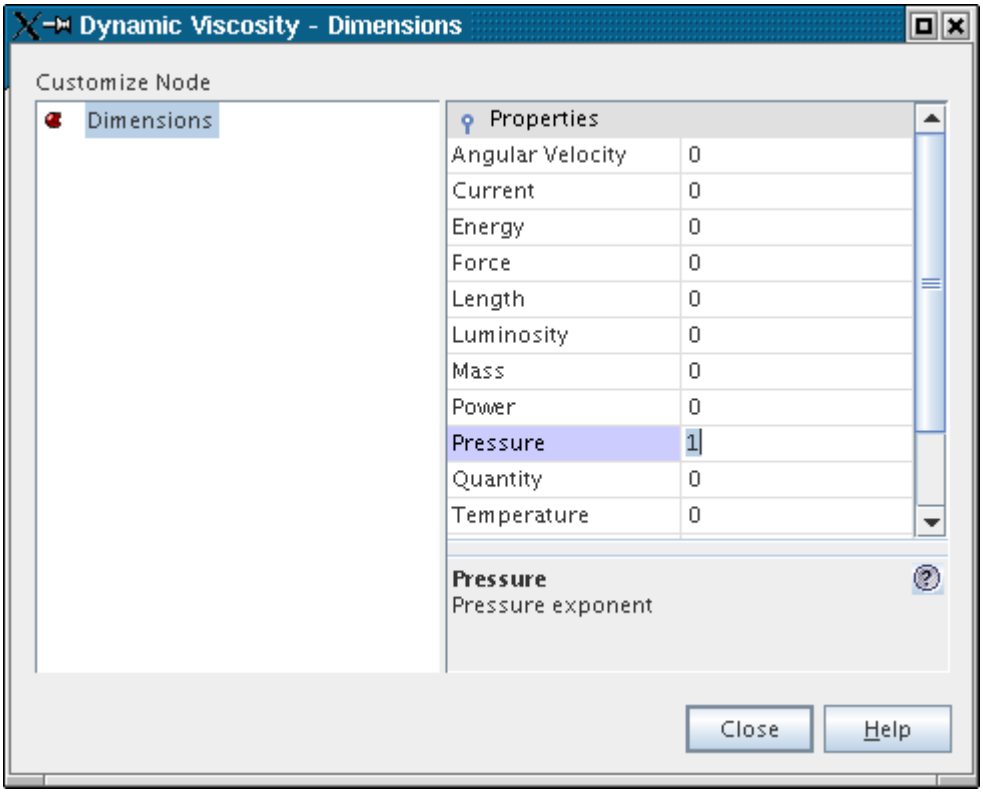
The copy action causes all the conditions and values in the destination region, boundary, or interface to be identical to the source.

This copy feature can be used between regions, boundaries, and interfaces of the same simulation or between two different simulations, provided that both the simulations have been opened in the same STAR-CCM+ session.

Defining Dimensionality

For various objects in the simulation tree, clicking the property customizer (...) of the *Dimensions* property opens a *Dimensions* dialog.

It allows you to set the dimensionality of a value or unit by setting the exponent of each of the base dimensions.



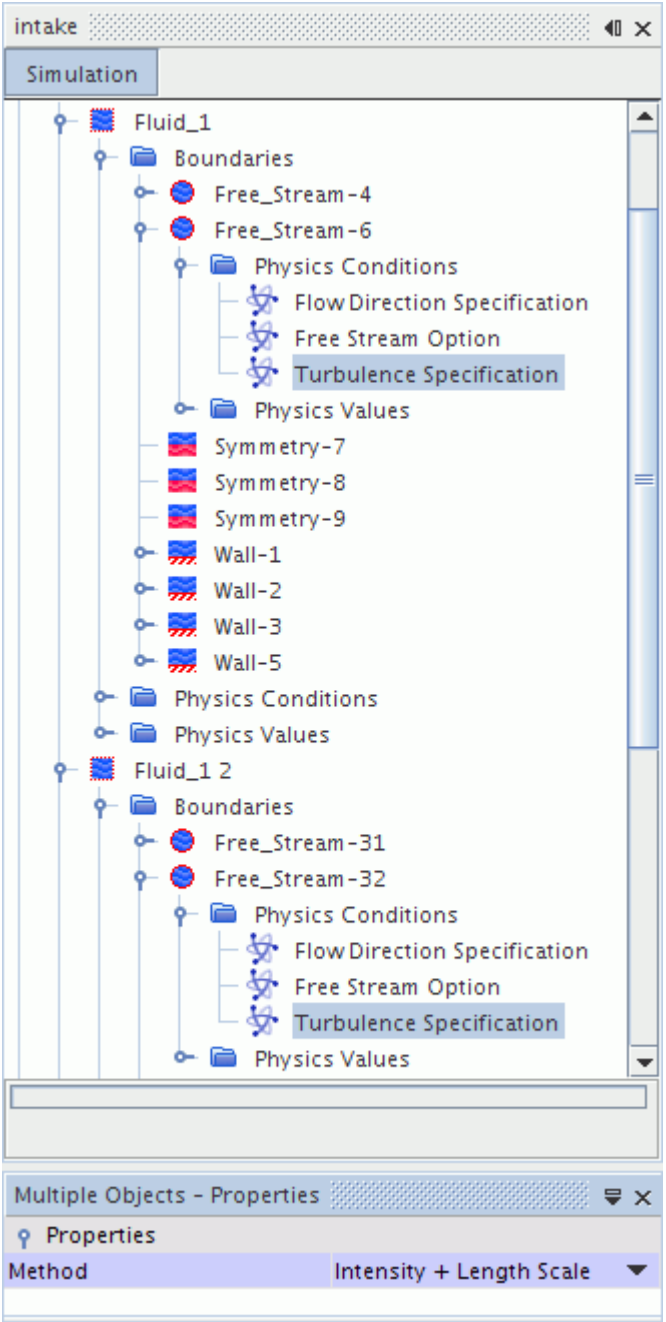
Setting Multiple Corresponding Conditions Simultaneously

Corresponding conditions for different regions, boundaries, and interfaces can be set at the same time.

To do this:

1. Open the nodes for the corresponding regions, boundaries, or interfaces.
2. Open the **Conditions** nodes.
3. Select the required conditions using the mouse to do multiple selection.
4. In the *Properties* window, select one of the available options from the drop-down list corresponding to the *Method* property.

   This option is applied to all the selected conditions.



Setting Multiple Corresponding Values Simultaneously

Corresponding conditions for different regions, boundaries, and interfaces can be set at the same time.

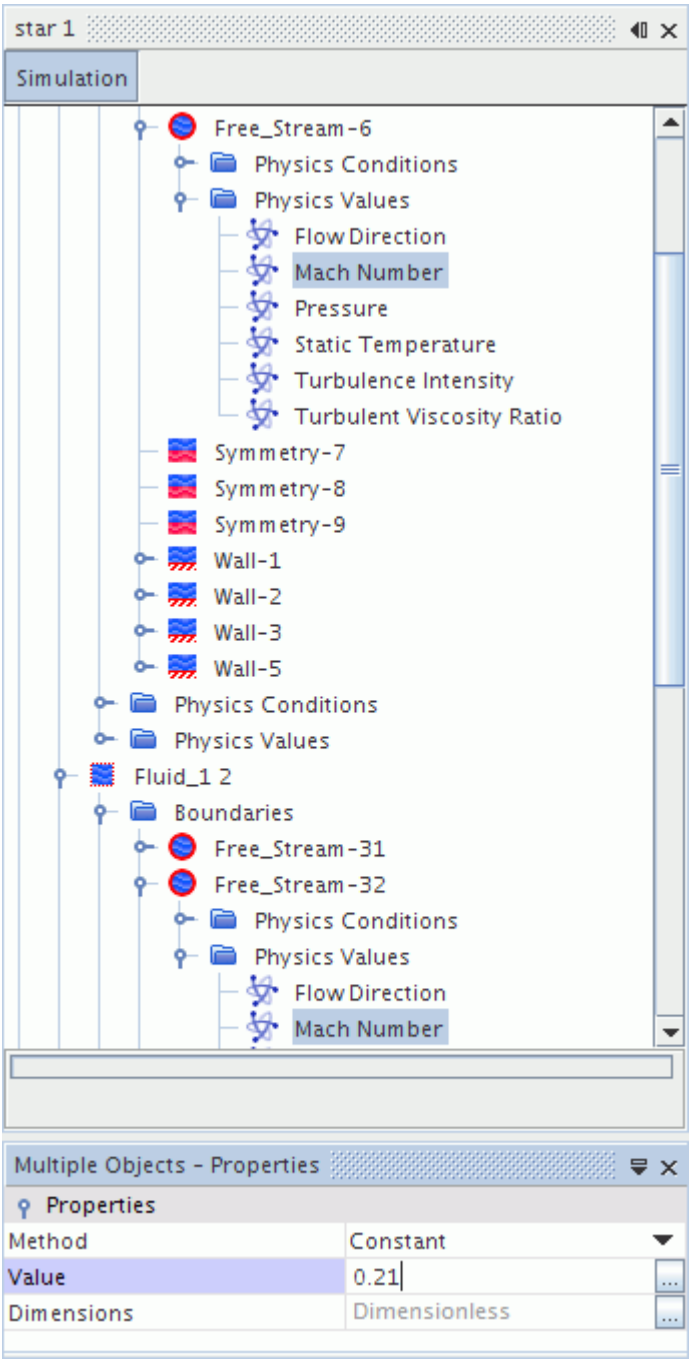
This section describes how to enter multiple values in the object tree. However, depending on the parent node, you can also use the table editor.

This technique is illustrated by setting a constant value, but the principle would hold true for other methods such as field functions or tables.

1. Open the nodes for the corresponding regions, boundaries, or interfaces.

2. Open the **Values** nodes in each of the corresponding regions, boundaries, or interfaces.
3. Select the required profile nodes using the mouse to do multiple selection.
4. In the *Properties* window, enter the appropriate values.

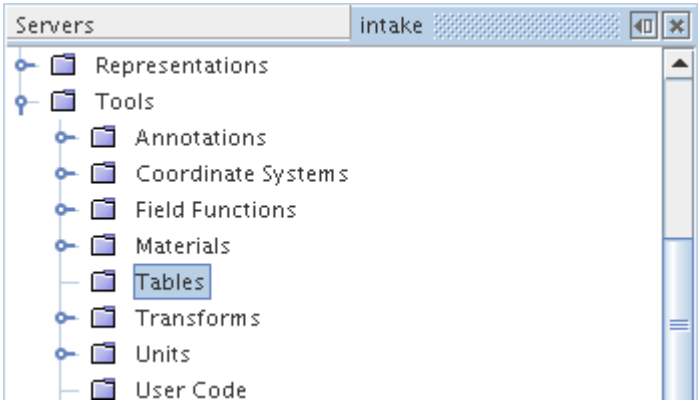
   This entry is applied to all the selected values.



Tables

Tables refer to objects in STAR-CCM+ containing tabular data.

All the tables that are created in a simulation are listed under the **Tools** > **Tables** node. The **Tables** node exists even when empty to allow creation of the first table. The tables manager can hold any number of tables up to the memory limits of the computer.



Tables are multi-variable entities akin to spreadsheets. They can be used for:

- Specifying tabular data for the simulation, particularly for boundary conditions and/or initial conditions. If you are providing table data based on experiment or analysis, for example, material property data, be aware that data organized in a Cartesian manner gives the best performance. See Cartesian and Structured Tables.
- Extracting tabular data from the simulation, for instance to use as boundary conditions for other simulations

The table contains a combination of location and scalar data. Normally, a file table is created using external data such as from experiments or an analysis, but it could also be extracted and exported from a simulation.

The types of table that are available in STAR-CCM+ are listed below. For more information on each type and on the available properties and actions, see Tables Reference.

Note: To import properly into STAR-CCM+, an input file that contains special characters (including accented characters) requires UTF-8 encoding rather than ANSI.

Table Types

File Table	A file table contains data that STAR-CCM+ imports from a table data file of format `.csv`, `.tab`, or `.xy`. It is possible to access the table data from a field function.
Histogram Table	Histogram tables show the distribution of a selected quantity with optional weighting.

Accumulated Force Table	Accumulated force tables provide a convenient way to plot accumulated pressure and shear forces (or both) over the surface of selected parts.
Radial Internal Table	A radial internal table specifies location in terms of radius, and the data set must contain an identifier specifying that it is a radial table. The radial internal table also needs a list of identifiers corresponding to the “columnar data” in the file. The columnar data must contain the radial coordinate as well as the scalar data. An example of the file format is given in File Format Reference. The interpolation method for radial tabular data is a cubic spline. Hence, it is assumed that the radial data are sorted and single-valued.
XYZ Internal Table	An xyz internal table specifies location in Cartesian coordinates corresponding to cell or face centroid positions in the Laboratory coordinate system. The data set must contain an identifier specifying that it is an xyz table. It also needs a list of identifiers corresponding to the “columnar data” in the file. The columnar data must contain the Cartesian coordinates (x and y are only required for two-dimensional simulations) as well as the scalar data. An example of the file format is given in File Format Reference. The interpolation method for xyz tabular data is zeroth order. When interpolating the data to a specific cell or face location, STAR-CCM+ finds the coordinate location in the tabular data that is closest to the specific cell or face centroid. STAR-CCM+ then applies the appropriate scalar data from the table to that cell or face.

Contents:

Reading a Table Data File
Creating and Filling a Table
Exporting a Table
Copying Part Data Between Simulations
Displaying a Table
Automatically Extracting and Exporting Tables
Copying, Deleting, and Renaming Tables
Tables Reference

Reading a Table Data File

To read a table data file:

1. Right-click the **Tools** > **Tables** node and select **New Table** > **File Table**.
2. In the *Open* dialog, select one or more tabular data files, then click **Open**.

   STAR-CCM+ creates a table object for each selected table file. All the table files that you select must be in the same directory. For more table files in another directory, repeat this action.

If you modify a table file that has been already read into the simulation, you can reread the table file:

1. Right-click its table node and select **Reload File**.

   STAR-CCM+ updates the table data for this object in the simulation.

If you wish to use a different set of data in the same table object, you can change the file path of the table:

- Select the table node. In the *Path* property, enter the relative path to the new file.
- Alternatively, click ... (**Custom Editor**) and use the *Open* dialog to specify a new file.

Once you have changed the path, reload the table file. For best results, ensure that your new table file has the same titles, numbers of columns, and rows as the old one.

Creating and Filling a Table

Create an xyz internal table or a radial internal table that is filled with the data within an existing simulation.

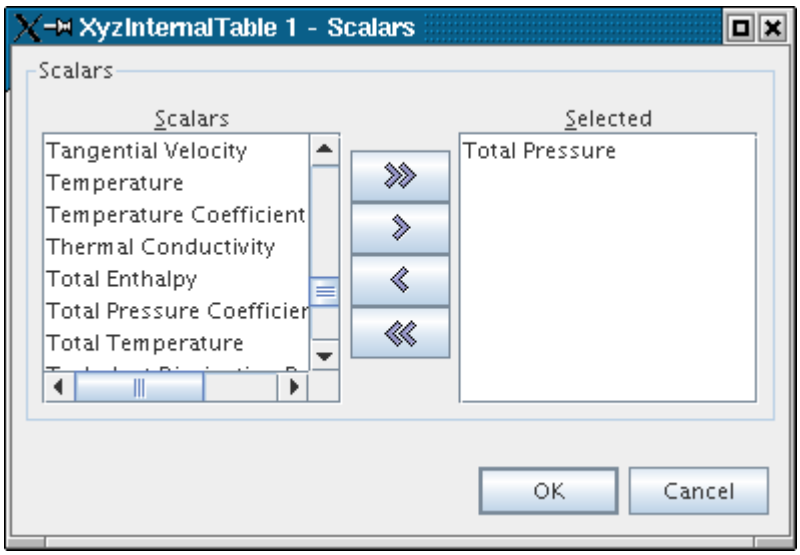
To create a new (empty) table:

1. Right-click the **Tools** > **Tables** node, and select **New Table** > **XYZ Internal Table** or **Radial Internal Table**.

   STAR-CCM+ adds a table node under **Tables**.
2. Specify the parts from which the data are extracted, by setting the table node *Parts* property.

To specify the scalar field function values to extract.

3. Click ... (**Custom Editor**) next to the *Scalars* property.
4. In the *[Table] - Scalars* dialog, select the required scalars.



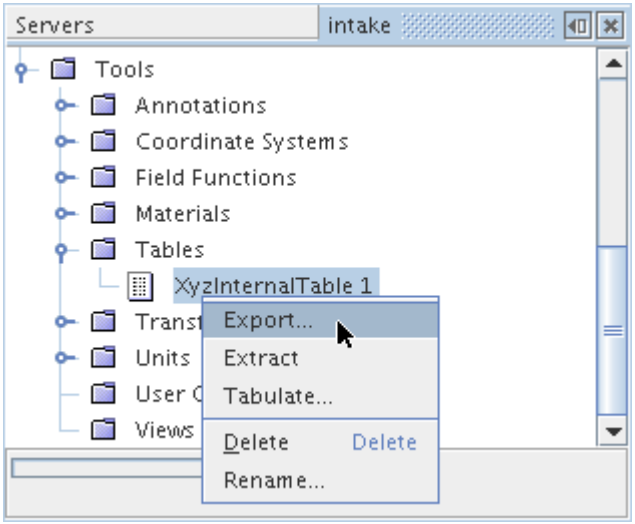
5. Once the parts and the scalars are specified, right-click the table node and select **Extract**.

   STAR-CCM+ fills the table with data. Use this action to update the table object when the external table file is updated with new data.

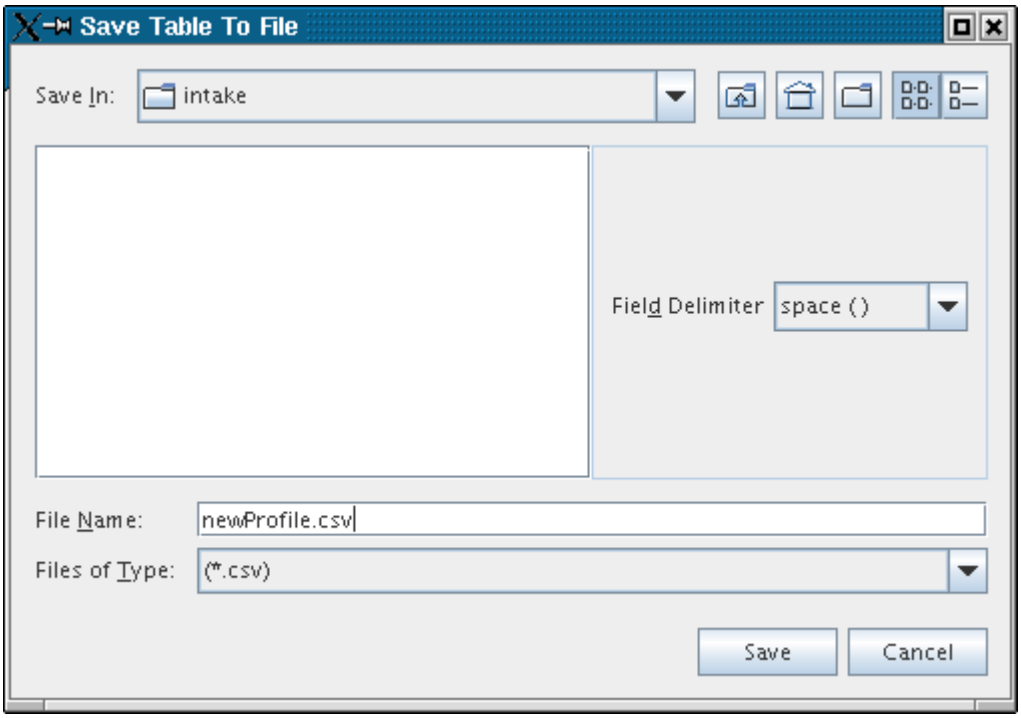
A table that is created in this way can either be used in the current simulation or exported to a file.

Exporting a Table

To export an existing table to a file, right-click the table name and select **Export...**



In the *Save Table to File* dialog, enter the file name (with the extension `.csv`) and click **Save**.



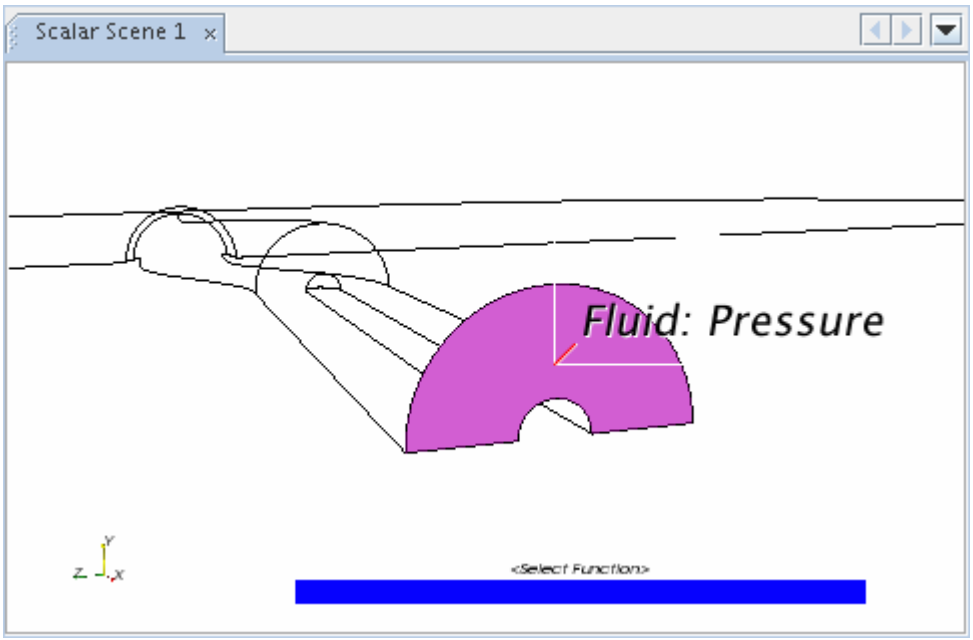
Copying Part Data Between Simulations

You can use tables to copy solution data of a single part from one simulation to another.

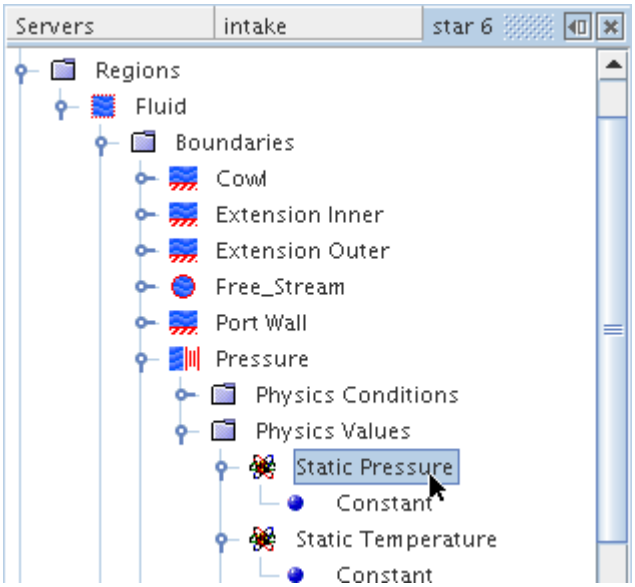
Begin by creating a table in the simulation that contains the solution data. In that new table, specify the part from which you want to copy the data. Select the scalar that you want to copy and then extract the data. See Creating and Filling a Table.

Then you can export the table to a file. Finally, read the table into the simulation file where you want to add the data.

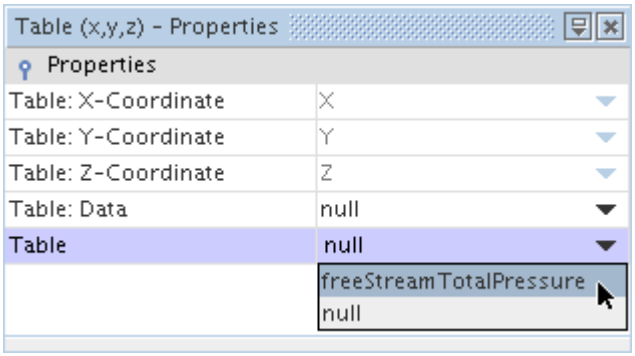
In the destination file, set up a scalar scene with the part on which you want the table data applied.



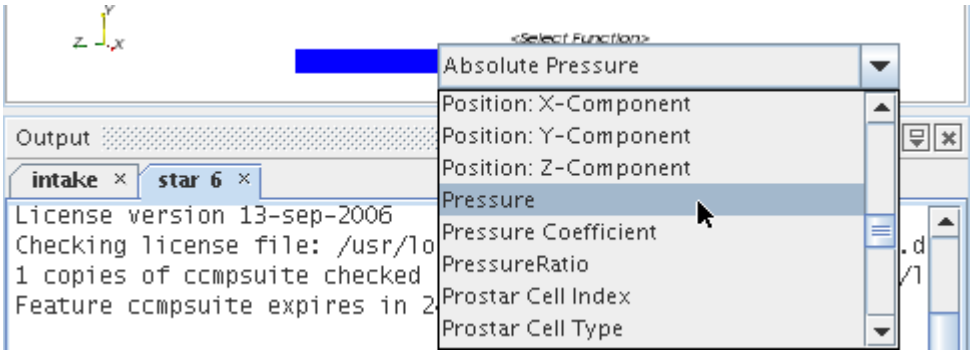
Suppose that the destination part is a boundary. Open the **Regions** node, the node of the region containing the boundary, the **Boundaries** node, the node of the boundary and the **Physics Values** node. Select the profile node that corresponds to the table values that you are copying.



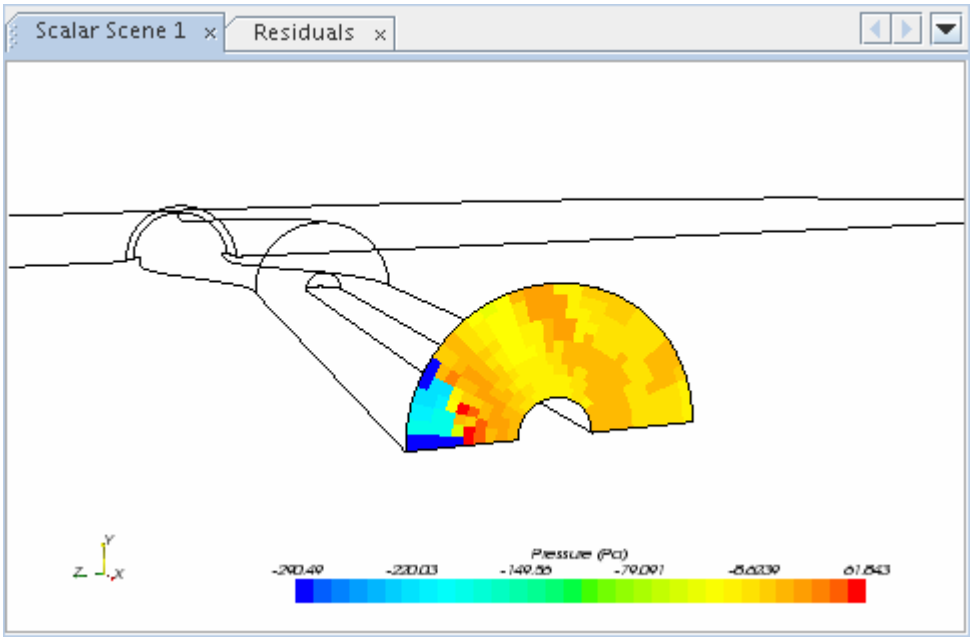
Select a tabular method for the profile. Then set the tabular method properties to use the values in the table file.



Set the value in the scalar scene to the one that corresponds to the table values that you are copying.

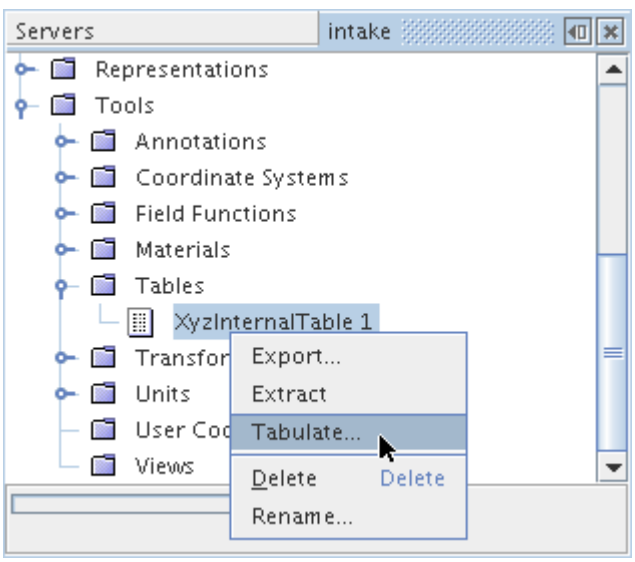


Next, step the solution. (When the *Residuals* display appears, click the tab of the scalar scene display to bring it back into focus.) The table values are displayed on the part in the scalar scene.

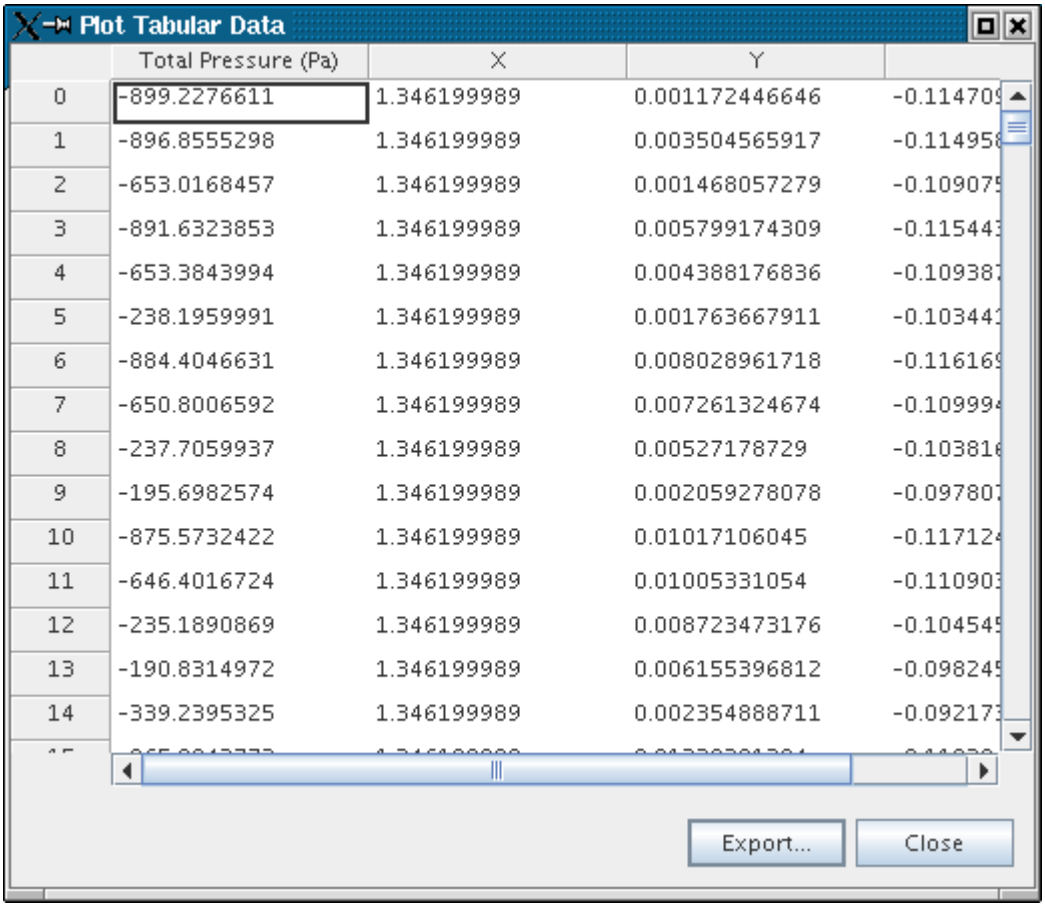


Displaying a Table

To display the data contained in a table, right-click the table node and select **Tabulate...**



The data from the plot are presented in the *Plot Tabular Data* dialog.



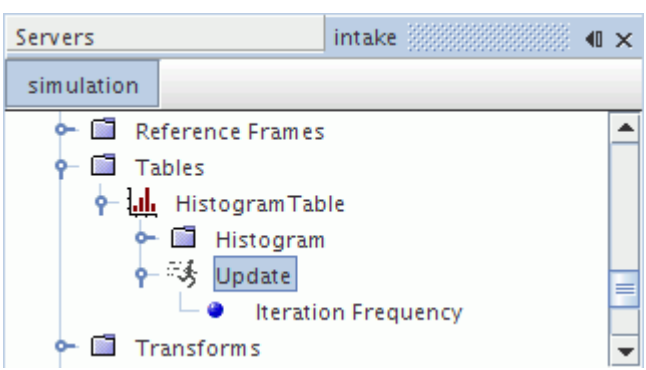
	Total Pressure (Pa)	X	Y	
0	-899.2276611	1.346199989	0.001172446646	-0.114709
1	-896.8555298	1.346199989	0.003504565917	-0.114958
2	-653.0168457	1.346199989	0.001468057279	-0.109075
3	-891.6323853	1.346199989	0.005799174309	-0.115443
4	-653.3843994	1.346199989	0.004388176836	-0.109387
5	-238.1959991	1.346199989	0.001763667911	-0.103441
6	-884.4046631	1.346199989	0.008028961718	-0.116169
7	-650.8006592	1.346199989	0.007261324674	-0.109994
8	-237.7059937	1.346199989	0.00527178729	-0.103816
9	-195.6982574	1.346199989	0.002059278078	-0.097807
10	-875.5732422	1.346199989	0.01017106045	-0.117124
11	-646.4016724	1.346199989	0.01005331054	-0.110903
12	-235.1890869	1.346199989	0.008723473176	-0.104545
13	-190.8314972	1.346199989	0.006155396812	-0.098245
14	-339.2395325	1.346199989	0.002354888711	-0.092173

Units are displayed in the preferred unit for the selected scalar.

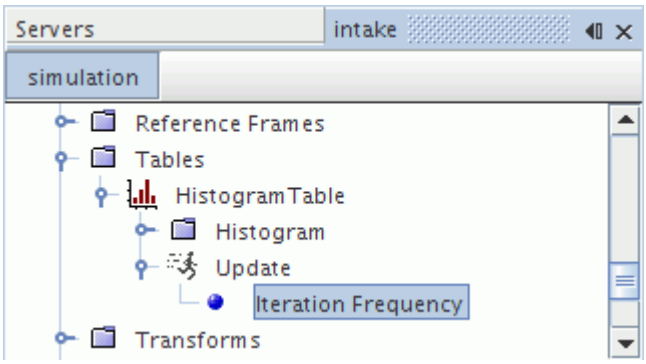
Automatically Extracting and Exporting Tables

You can update, extract, and export tables based on a trigger event.

To select the trigger event, expand the **Tools** >**Tables** node, and choose the table (internal table or histogram table). The **Update** node customizes how tables are updated as the simulation iterates. See Trigger Types.



The **Update** node has a sub-node with which to control the frequency of the action (update, extraction, and export). The properties of the **Update Frequency** child node are used to change the frequency of updates. See Trigger Properties.



For more information on selecting the trigger type and defining the update frequency, go to Setting the Trigger for an Update.

Update Properties

Trigger	Determines how the table is updated. Select the desired trigger type from the drop-down list. See Trigger Types.	
Auto Extract	Controls automatic extraction.	
	Activated	Extracts automatically.
	Deactivated	Does not use this feature.
Save to File	Controls automatic export.	
	Activated	Exports automatically.
	Deactivated	Does not use this feature.
Output Directory	Path to where the files specified in the *Base Filename Append Tag* property are saved.	
Base Filename Append Tag	The prefix of the file name to use when saving tables to file.	

Copying, Deleting, and Renaming Tables

Copying Tables

When properties are copied from one table to another, the *Dimensions* and *binsNumber* properties are changed. (The *binsNumber* property appears in radial tables.) The *Type* property remains unchanged. You would still specify the parts and extract the data of the destination table. See Creating and Filling a Table.

It is not possible to copy properties between different types of tables.

Deleting Tables

Tables can be deleted from an existing simulation as long as they are not in use. To prevent tables that are in use from being deleted, the **Delete** menu item for that table are grayed out.

Renaming Tables

Similar to other objects, a table can be renamed within STAR-CCM+.

Tables Reference

Tables Action Menu

These pop-up menu items are available when you right-click the **Tables** manager node.

New Table	Creates a new tabular data set of the type that is selected in the submenu.	
	File Table	Executes the open file action and reads in an existing tabular data file.
	Histogram Table	Executes the Create Internal Tabular Data Set action and creates an empty histogram tabular data set.
	Accumulated Force Table	Executes the Create Internal Tabular Data Set action and creates an empty accumulated force tabular data set.
	XYZ Internal Table	Executes the Create Internal Tabular Data Set action and creates an empty xyz tabular data set.
	Radial Internal Table	Executes the Create Internal Tabular Data Set action and creates an empty radial tabular data set.

[Table] Action Menu

These pop-up menu items are available when you right-click a table node, **Tables** > **[Table]**.

Export...	Saves the content of the table to a separate `.tab` file.
Extract	Fills a table with data extracted from specified scalar functions.
Reload file	Updates the table data from a modified file (file tables only).
Tabulate...	Opens a dialog displaying tabular data.

File Table Properties

Extracted	A read-only list of the names of the scalar field in the file. This property is blank if a new tabular data set has been created but no scalars have yet been selected.
File Path	The path to the table file from which this table object gets its data. This path is relative to the location of the simulation file.

Internal XYZ Table Properties

Extracted	A read-only list of the names of the scalar field in the file. This property is blank if a new tabular data set has been created but no scalars have yet been selected.	
Scalars	The property customizer activates a list box functionally similar to the *Select Objects* dialog, with which you can select scalars.	
Parts	Editing the property activates the *Select Objects* dialog for parts.	
Coordinate System	Specifies the coordinate system (laboratory by default). The drop-down list of this property includes any local coordinate systems that exist in the simulation (xyz internal table only).	
Data on Vertices	Controls whether to extract field data to cell centers or vertices. Depending on what field data you are using, use the appropriate option to facilitate direct mapping rather than interpolation or extrapolation.	
	Activated	Extracts data at vertices. Fields associated with vertex coordinates will be mapped without interpolation and fields associated with cell centers will be interpolated (or extrapolated) to the vertex coordinates.
	Deactivated	Extracts data at cell centers. Fields associated with cell centers will be mapped without interpolation and fields associated with vertex coordinates will be interpolated (or extrapolated) to the cell centers.
Representation	Selects a representation from among the available ones in the simulation.	

Radial Internal Table Properties

Number of Bins	Number of radial stations.
Parts	Editing the property activates the *Select Objects* dialog for parts.
Scalars	The property customizer activates a list box functionally similar to the *Select Objects* dialog, with which you can select scalars.
Extracted	A read-only list of the names of the scalar field in the file. This property is blank if a new tabular data set has been created but no scalars have yet been selected.

Histogram Table Properties

Parts	Parts from which the table data is extracted.
Extracted	A read-only list of the names of the data sets extracted for the table.
Representation	Selects a representation from among the available ones in the simulation.

The properties of the **Histogram Table** > **Histogram Base** node, and its sub-nodes (some of which are optional) control the definition of the histogram table:

Histogram Base

Bins	Number of discrete intervals in the histogram. Selects the scalar field function to split into separate intervals (or bins), using the **Histogram Base** > **Bin** node.	
Smooth Values	When activated, instructs STAR-CCM+ to use node values to populate the histogram, instead of cell values.	
Generate Profile	When activated, STAR-CCM+ evaluates a specified field function, for each set of points that is allocated to a bin, and returns the maximum and minimum values of the function. Specify the field function and its units using the **Histogram Base** > **Profile** node.	
Weighting	STAR-CCM+ weights the count of each point that is allocated to a bin by the amount that is selected here.	
	Frequency	Equivalent to a weighting of one. The sum for each bin is the frequency with which points fall into the bin.
	Specified Scalar	Allows you to specify the field function that is used to weight each data point allocated to a bin. Specify the field function and its units using the **Histogram Base** > **Weighting** node.
	Specified Scalar with Accumulation	Same as **Specified Scalar**, but with the additional property that values for successive bins are summed cumulatively along the bin direction. Specify the field function and its units using the **Histogram Base** > **Weighting** node.
Range Mode	Specifies how to find the maximum and minimum range of bin values.	
	Auto	STAR-CCM+ automatically calculates the maximum and minimum range of values for the bin field function, and sets the range from this calculation.
	Manual	Allows you to manually specify the minimum and maximum values of the binning quantity. Specify the field function and its units using the **Histogram Base** > **Range** node.

Accumulated Force Table Properties

Parts	Editing the property activates the *Select Objects* dialog for parts.

Extracted	A read-only list of the names of the data sets extracted for the table.

Use the properties of the **Accumulated Force Table** > **Force Histogram** node to set the parameters that are required for building the histogram of forces and accumulated force across the chosen parts. The forces that are extracted are the same as the forces given for the Force. When force normalization is activated, the force coefficients, and accumulated force coefficients, are returned.

Force Histogram

Bins	Number of discrete intervals (or bands) across the parts.	
Coordinate System	Coordinate system in which the bin, force, and profile directions are defined. The drop-down list of this property includes any local coordinate systems that exist in the simulation. The default is laboratory. This property does not control the *Position* values given in the displayed table. The *Position* values given in the table are always for the centroids of the bins, which are given in the laboratory coordinate system along the specified bin direction.	
Bin Direction	Direction in which the intervals (or bands) are distributed.	
Force Direction	Direction in which the force is computed	
Profile Direction	Direction to use in extracting the profile from the parts.	
Force Option	Choose which force components are included in the reported data.	
	Pressure	Only the force due to pressure, $\mathbf{f}_f^{\text{pressure}}$ is considered. See Eqn. (1242) for the Force report.
	Shear	Only the force due to shear, $\mathbf{f}_f^{\text{shear}}$ in Eqn. (1242) is considered.
	Pressure and Shear	Both pressure and shear forces, $\mathbf{f}_f^{\text{pressure}}$ and $\mathbf{f}_f^{\text{shear}}$ in Eqn. (1242) is considered.
Normalization Mode	When activated, instructs STAR-CCM+ to normalize the force as a force coefficient. See Eqn. (1249). Use the **Force Histogram** > **Force Normalization** node to specify the reference density, velocity, and area that are used to calculate the force coefficients (see Eqn. (1249), Eqn. (1249), and Eqn. (1249)).	
Range	Maximum and minimum band values (read only).	
Reference Pressure	The reference pressure for the computation of Eqn. (1243).	
Range Mode	Sets how band positions are calculated.	
	Auto	Use this option if you want STAR-CCM+ to calculate the start and end positions of the bands.
	Manual	Use this option if you want to specify where the first band begins and the last band ends.

		Note: When you specify limits manually, any data outside those limits is ignored.
Spacing	The width of the bands across the selected parts (read only)	

Contents:

File Format Reference
Cartesian and Structured Tables

File Format Reference

This section details the formats of files when tables are saved to disk and the formats of files from which you can read a table.

The following principles apply to using variables in either internal table format:

- You can use any variable name that you like in the file as a heading row, and then assign it to the scalar value in the boundary condition. See Setting Values Using A Table.
- The file can contain any number of variables, and they can be used or unused.
- However, the xyz file must have the fields X, Y and (if three-dimensional) Z, while the radial file needs the field R.
- A variable name can contain a units specification. The units are specified within a set of parentheses () as part of the variable name:

  ```
  "Volume (in^3)", "Density (lb/ft^3)"
  ```

  Imported units are converted to the preferred units for a dimension that is based on the *Preferred System* property, which is specified in the **Units** node. See Working with Units.

  If the provided units are not known to the units manager, the *Output* window displays that unit conversion is not possible.
- The layout of data in imported tables can impact STAR-CCM+ performance. See Cartesian and Structured Tables for details.

Note: To import properly into STAR-CCM+, an input file that contains special characters (including accented characters) requires UTF-8 encoding rather than ANSI.

Xyz Internal Table

An example of a two-dimensional xyz internal table file is shown below, with columns for 'X' and 'Y' and the field functions 'U' and 'SaNut'.

Critical aspects of the file format are:

- The specific parentheses used
- The use of quotation marks to identify text strings
- The literal identifiers including 'Type', 'xyz' and 'DataSets'.

The columnar data inside the brackets are free format, and delimiters are used to separate the individual numbers. Each column of data must have the same number of elements. There must be columns of data that are identified using ’X’, ’Y’ (and ’Z’ in three-dimensional space).

```
{’Type’: ’xyz’
’DataSets’: [ ’X’  ’Y’  ’U’  ’SaNut’]
’X’:
[-4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4]
’Y’:
[1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0]
’U’:
[0.04 0.123 0.209 0.291 0.359 0.409 0.445 0.471 0.491 0.509 0.523]
’SaNut’:
[1.14e-05 3.51e-05 6.078e-05 8.84e-05 0.00012 0.00015 0.00019 0.00022
0.00026 0.00030 0.00035] }
```

Radial Internal Table

An example of a radial internal table file is shown below.

Critical aspects of the file are the specific parentheses used, the use of quotation marks to identify text strings, and the literal identifiers such as ’Type’, ’radial’ and ’DataSets’. The columnar data inside the brackets are free format, and any delimiters can be used to separate the individual numbers. Each column of data must have the same number of elements. There must be a column of data identified as ’R’.

```
{’Type’: ’radial’
’DataSets’: [ ’R’  ’axial-vel’]
’R’: [0 0.01 0.02 0.03 0.05 0.07 0.09 0.1 0.12 0.13 0.14 0.15]
’axial-vel’: [0.9700 0.9756 0.9697 0.9878 0.9955 1.0223
1.0404 1.0519 1.059543 1.065043 1.064233 1.071543] }
```

.csv Table File

A simplified example of a `.csv` (comma-separated values) table file is shown below.

The individual numbers can be separated by commas or white space. Each column of data must have the same number of elements. Labels at the top are optional, but if a label contains multiple strings it must use quotation marks, for example 'Laminar Flame Speed'.

```
Vol   Press
0.01  458.58
0.10  417.85
0.20  286.26
0.30  278.52
0.40  267.68
0.50  254.59
0.60  239.05
 ...   ...
2.20 -204.01
```

This format is the best way to bring in user data (`.usr`) files from pro-STAR that were written in ASCII format using the SAVU command.

.tab Table File

An example of a `.tab` table file is shown below.

Each column of data must have the same number of elements.

Critical aspects of the file format are:

- The specific parentheses used.
- The use of quotation marks to identify text strings.
- The literal identifiers such as ’DataSets’, ’Modified Diffusivity’ and ’Velocity: X-Component’.

```
{’DataSets’: [’Modified Diffusivity’, ’Velocity: X-
Component’, ’Velocity: Y-Component’, ’X’, ’Y’, ’Z’],
’Modified Diffusivity’: [0.0005044474964961409568 8,
0.0015556754078716039658, ...],
’Type’: ’xyz’,
’Velocity: X-Component’: [0.5540286302566528320 3,
0.68478977680206298828,...],
’Velocity: Y-Component’: [-7.3806604632409289479e-06,
-3.4772736398736014962e-05, ...],
’X’: [-4.0000000000000000000, -4.0000000000000000000,...],
’Y’: [8.9650001525878906250, 8.8903274536132812500,...],
’Z’: [0.0000000000000000000, 0.0000000000000000000,...]}
```

Although `.csv` is the standard format for exporting table data from STAR-CCM+, `.tab` is still supported as a legacy format for reading into a simulation.

.xy Table File

An example of a .xy table file is shown below.

Each column of data must have the same number of elements. There must be columns of data identified as ’X’, ’Y’ (and ’Z’ in three-dimensional space).

```
Y X Z U
0.0 0.0 0.0 0.00000
0.0005 0.0 0.0 0.11408
0.001 0.0 0.0 0.22230
0.002 0.0 0.0 0.42120
0.003 0.0 0.0 0.59670
0.004 0.0 0.0 0.74880
0.005 0.0 0.0 0.87750
 ...  ...  ...  ...
0.3 0.0 0.0 0.00000
```

Although `.csv` is the standard format for exporting table data from STAR-CCM+, `.xy` is still supported as a legacy format for reading into a simulation.

Cartesian and Structured Tables

Whenever possible, organize tables to be "Cartesian" or at least "Structured." These tables are much faster to search than "Unstructured" tables.

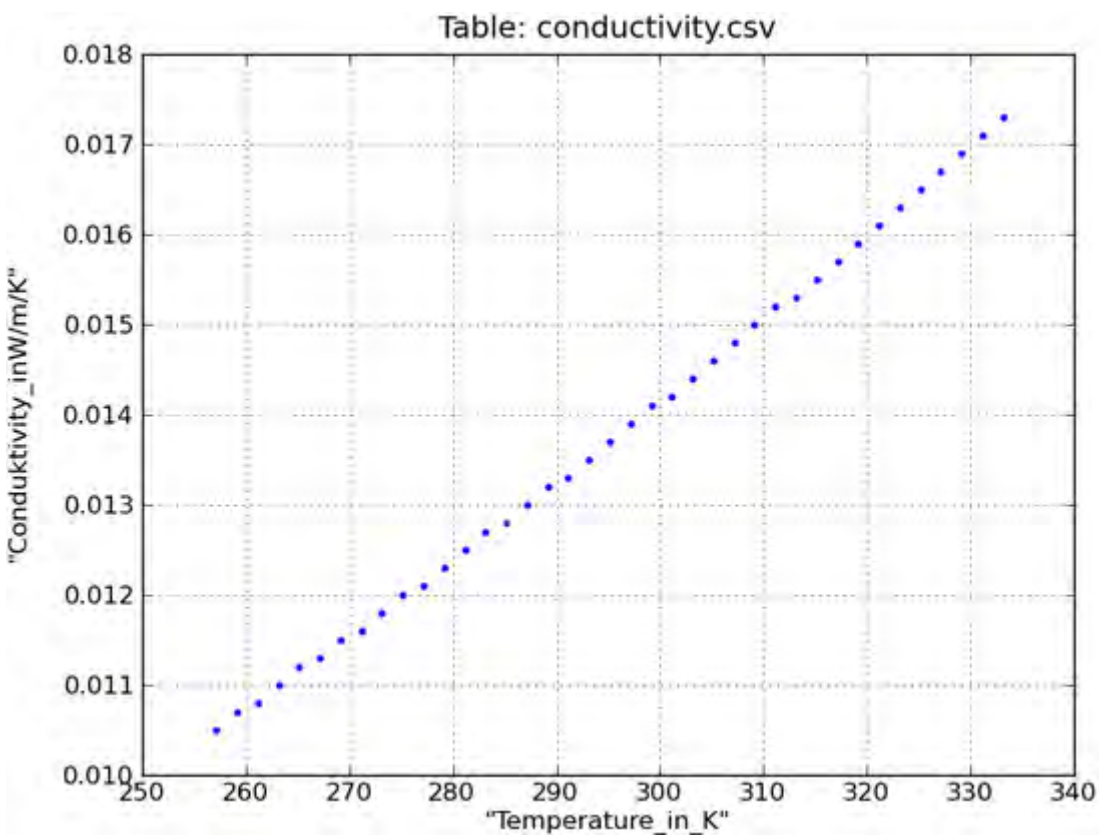
In a Cartesian table, the entries for the independent variable are equally spaced. In a Structured table, the spacing is predictable but not uniform. The organization of a table is reported in the read-only *Table Format* property for **Table(T)**, **Table(T,P)**, and **Table(Reaction Time)** tables.

When STAR-CCM+ reads a table, it first sorts the data points in each dimension and then computes the spacing. For example, in a 2D table let there be Nx points in the first dimension and Ny points in the second dimension. If the total number of specified points is Nx*Ny and if the spacing is uniform in each dimension then the table is tagged *Cartesian*. If the spacing is non-uniform then it is tagged *Structured*. If the total number of points does not match Nx*Ny, then it is tagged *Unstructured*. (Note that even if the spacing is uniform, but the total number of points does not match Nx*Ny, the table is tagged *Unstructured*, since this situation means that some points are missing in the Nx*Ny grid.)

Examples:

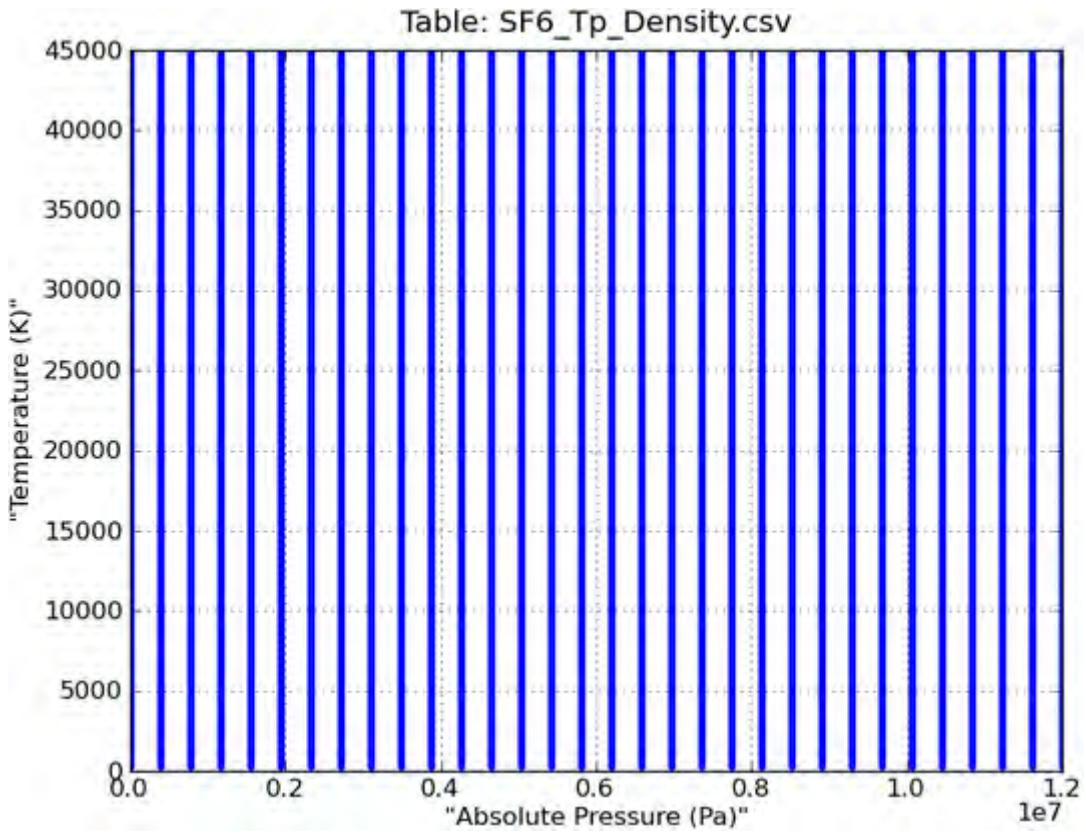
1D Cartesian

This table is a 1D Cartesian table since the sample points are equally spaced. 1D tables can only be Cartesian or Structured; they cannot be Unstructured.



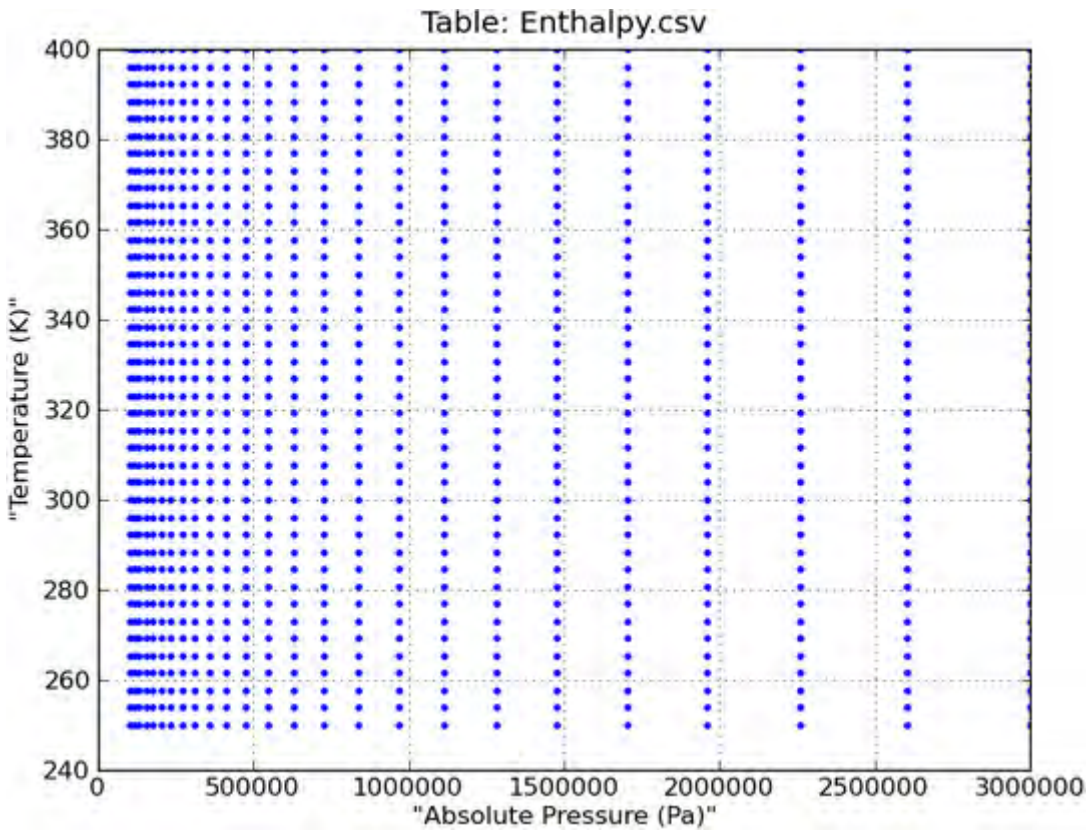
2D Cartesian

This table is a 2D Cartesian table since the points are equally spaced along the independent variable.



2D Structured

This table is a 2D Structured table since the points lie on a Structured grid, although the spacing is not uniform.



2D Unstructured

This tale is a 2D Unstructured table since the points are irregularly spaced along the independent variable.

These categories also apply to higher-dimensional tables. If the independent points lie on an N-dimensional Structured grid, then it is a Structured table or (if the spacing is uniform) a Cartesian table. If points are irregularly spaced in the N-dimensional space, then it is an Unstructured table.

What Are the Important Concepts?

This section presents the key concepts you come across when using STAR-CCM+.

It begins with an introduction to the client-server architecture, which is the foundation on which the software is built. Further concepts are:

Contents:

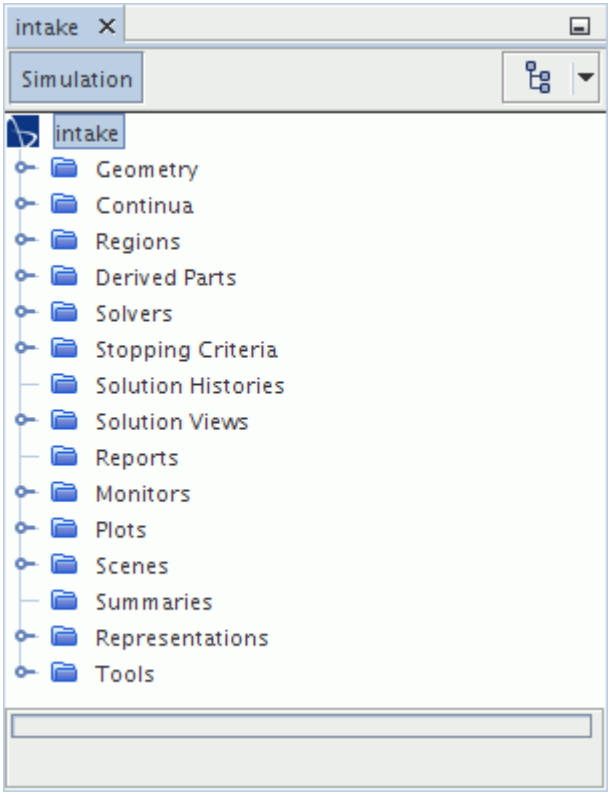
What Are Simulations?
What Are 3D-CAD Models?
What Are Parts?
What Are Continua?
What Are Models?
What Are Regions?
What Are Boundaries?
What Are Interfaces?
What Are Solvers?
What Are Reports, Monitors and Plots?
What Are Scenes?

What Are Simulations?

A simulation contains the solution state and data for a STAR-CCM+ analysis.

It exists on a server, and can only be accessed through a client connected to the server. When a client connects to a server, the object tree for the simulation on the server is shown in its own explorer pane in the client workspace. If the client is connected to multiple servers, the simulation object tree for each server is shown in its own explorer pane within the client workspace.

The root object within each simulation object tree is known as the simulation node. This node has its own properties and pop-up menu.



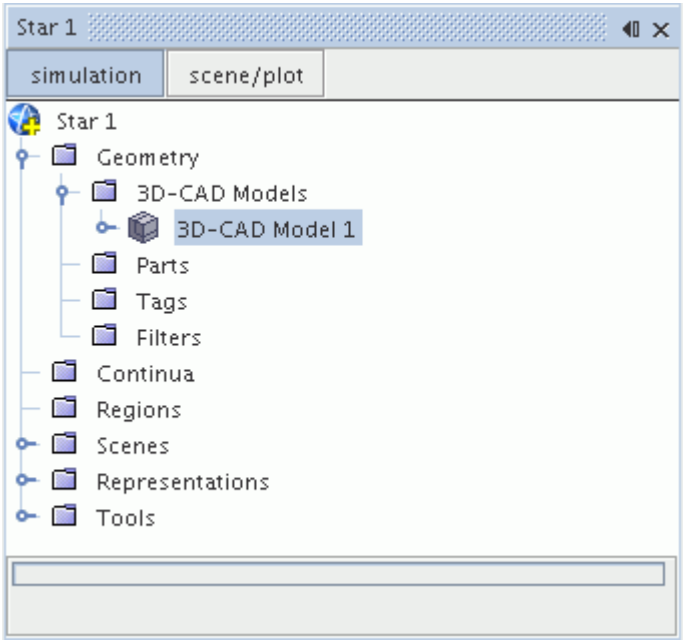
Simulation Properties

Name	The name of the simulation (read-only). To change the name of the simulation, use the save as action.
Connection Mode	Controls whether and how other users may connect to the server of this simulation. See Server Connection Mode.

What Are 3D-CAD Models?

STAR-CCM+ contains its own feature-based parametric solid modeler that allows you to create geometry for your analysis.

This integrated solid modeler is called the 3D-CAD module, and solid models generated from it are called 3D-CAD models. The screenshot below shows a single 3D-CAD model object within the simulation object tree.



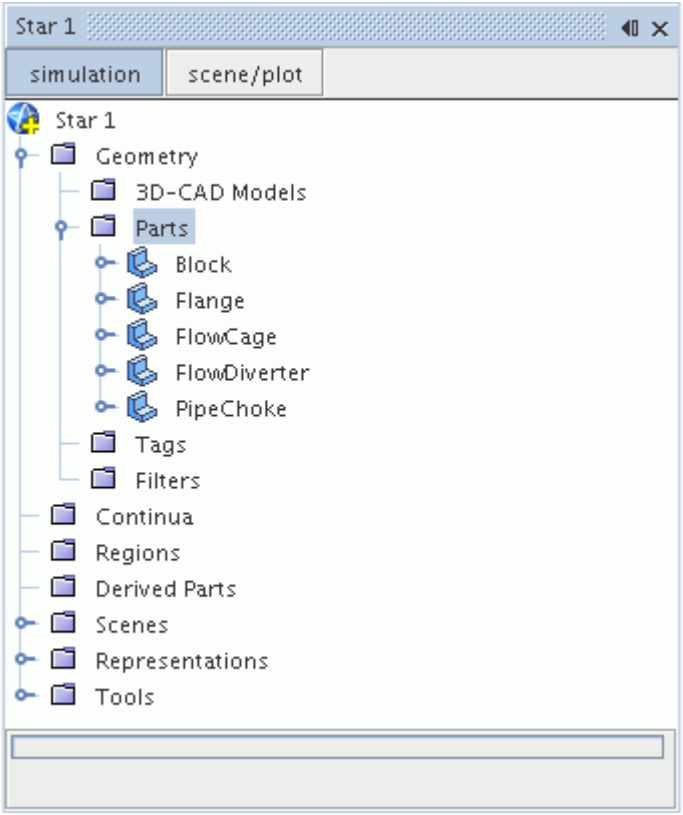
Complete documentation on using the 3D-CAD solid modeler can be found in Using 3D-CAD.

What Are Parts?

There are three different types of part that can be used in a STAR-CCM+ simulation:

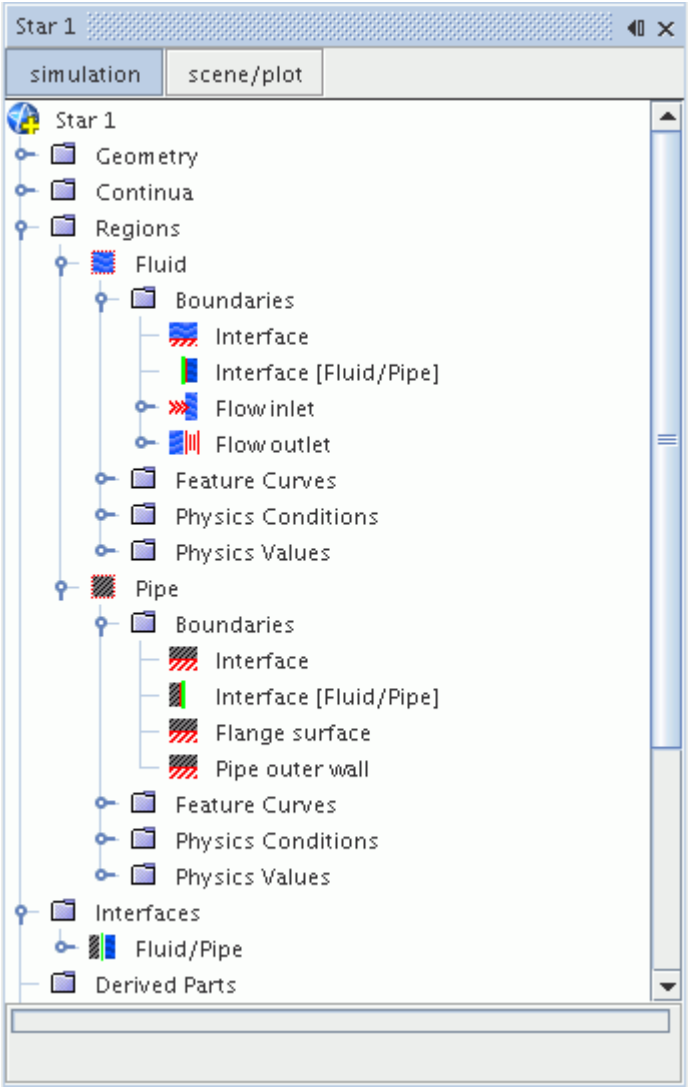
- Geometry parts

  These parts represent the model geometry, and are contained within the **Geometry** > **Parts** manager node of the object tree.



- Model parts, including regions, boundaries, and interfaces

  These represent the discretized portions of the geometry to be analyzed, on which physics models are applied. These parts are contained within the **Regions**, **Boundaries**, and **Interfaces** manager nodes.



- Derived parts

  These parts are created in a number of ways. They can be used in the definition of analysis reports and for visualizing solution data in scalar and vector scenes. All derived parts are contained within the **Derived Parts** manager node of the object tree.

Parts can be used as inputs to many other objects in STAR-CCM+, such as displayers, reports, tabular data sets, and x-y plots.

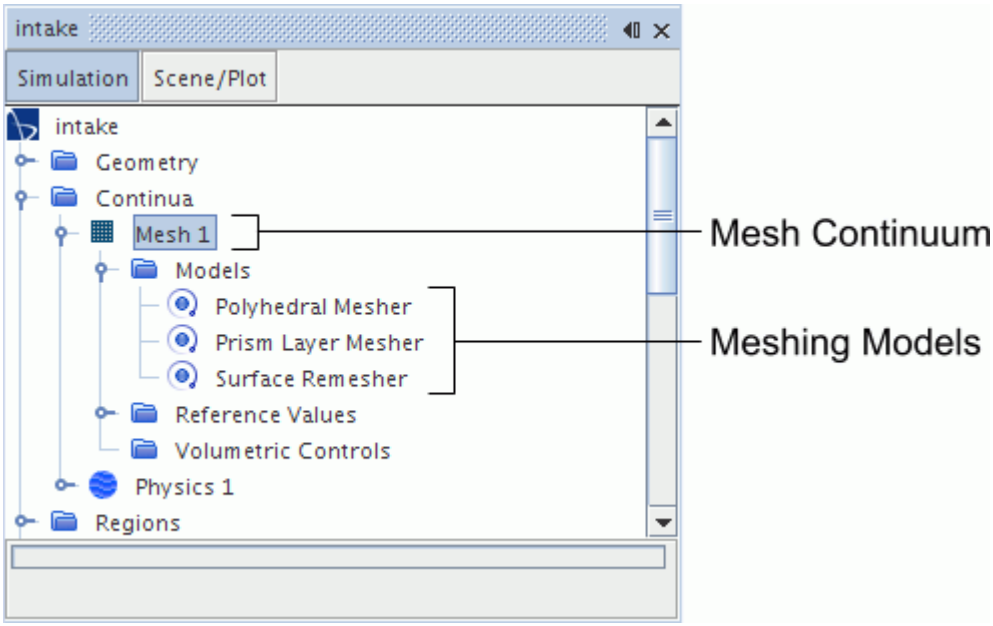
What Are Continua?

Continua are used to contain selections of physics or meshing models that are subsequently applied to one or more regions. Continua have no geometric definition associated with them.

The **Continua** manager node, which has its own pop-up menu, contains all of the continuum nodes in the object tree. Two types of continua may be present within this manager node: mesh continua and physics continua.

Mesh Continuum

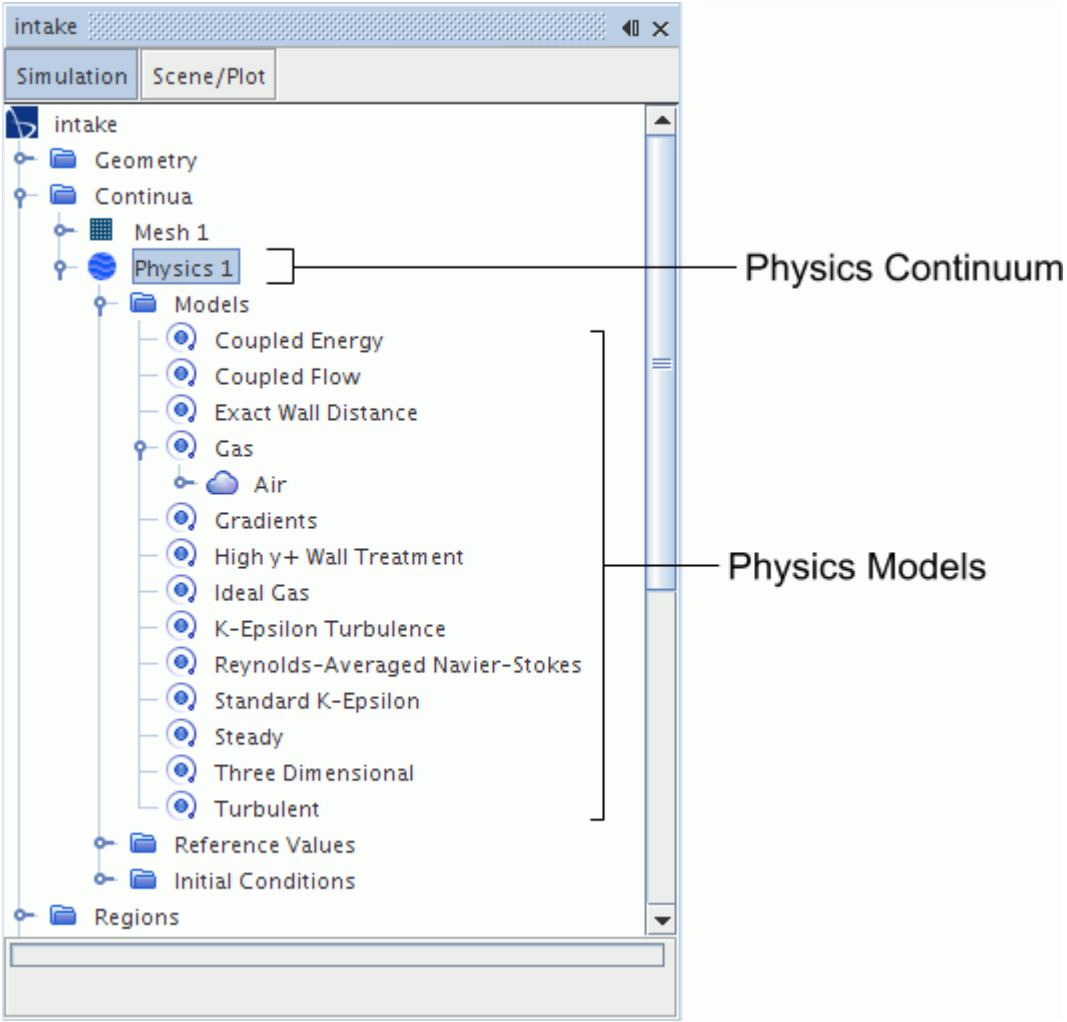
A mesh continuum contains a selection of meshing models, such as the surface remesher, the prism layer mesher, and the polyhedral mesher. When a mesh continuum is applied to a region, the region is discretized according to the meshing models selected for the mesh continuum. An example of a mesh continuum with its selection of models is shown below.



See the section, Mesh Continua, for further information.

Physics Continuum

A physics continuum contains a selection of physics models, such as a chosen flow solver, material models, steady or transient time model, a turbulence model, and so on. Each physics continuum represents a single substance that is present in all regions to which the physics continuum applies.



See the section, Using Physics Continua, for more information.

Managing Continua

Similar to other objects, a continuum may be renamed within STAR-CCM+. They can be deleted from an existing simulation as long as they are not in use. To prevent a continuum that is in use from being deleted, the **Delete** menu item for that continuum is grayed out. The contents of a continuum may also be copied from one continuum to another using the clipboard or by using drag-and-drop. The copy action causes all the models from the source continuum to be activated in the destination continuum, with the properties of each model also being copied. For physics continua, all reference values and initial conditions and values are copied from the source continuum to the destination continuum. For mesh continua, all reference values and volumetric control settings are copied from the source continuum to the destination one.

The continuum copy feature can be used within the same simulation or between two different simulations, if both simulations have been opened in the same STAR-CCM+ session.

Further documentation is available on mesh continua, and on physics continua. Some documentation common to both types of continua is included in the following pages.

Storing Commonly Used Continua

Suppose that you repeatedly perform simulations using the same continua, that is, simulations involving the same models and materials. Rather than setting up the continuum each time a simulation is created on a new mesh, you can set up the continuum once only, save it, then reuse it multiple times.

To set up the continuum:

1. Launch STAR-CCM+ and start a new simulation.
2. Open the **Continuum** node, and select the required models using the model selection dialog.
3. Set the material properties and model coefficients.
4. Save the simulation.

To reuse this continuum:

1. Run STAR-CCM+ and open the existing simulation that contains only the saved continuum.
2. Import the required mesh(es).
3. Save the simulation with a new name.
4. Proceed with the simulation as normal.

Pop-Up Menu of the Continua Manager

These pop-up menu items are available when you right-click the **Continua** manager node.

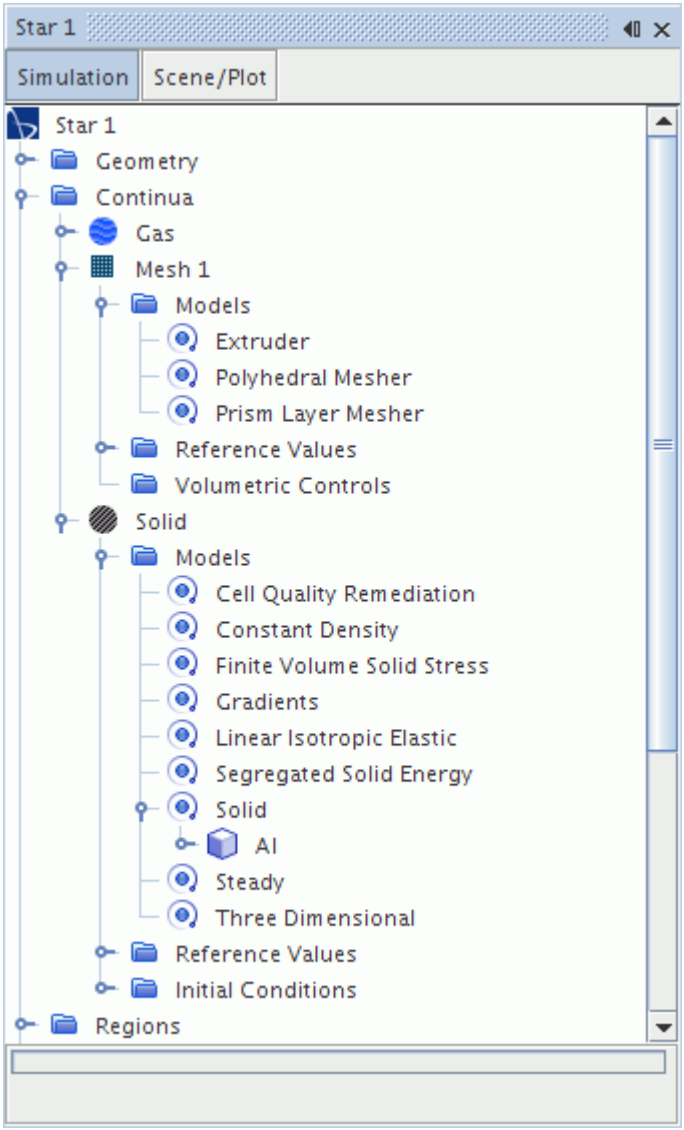
New Continuum	Adds a continuum with a default name to the model and three manager subnodes (models, reference values, and initial conditions).	
	Mesh Continuum	Creates a mesh continuum.
	Physics Continuum	Creates a physics continuum.

See also: common menu items.

What Are Models?

Model nodes in the simulation object tree represent simulation features that are active in the current case.

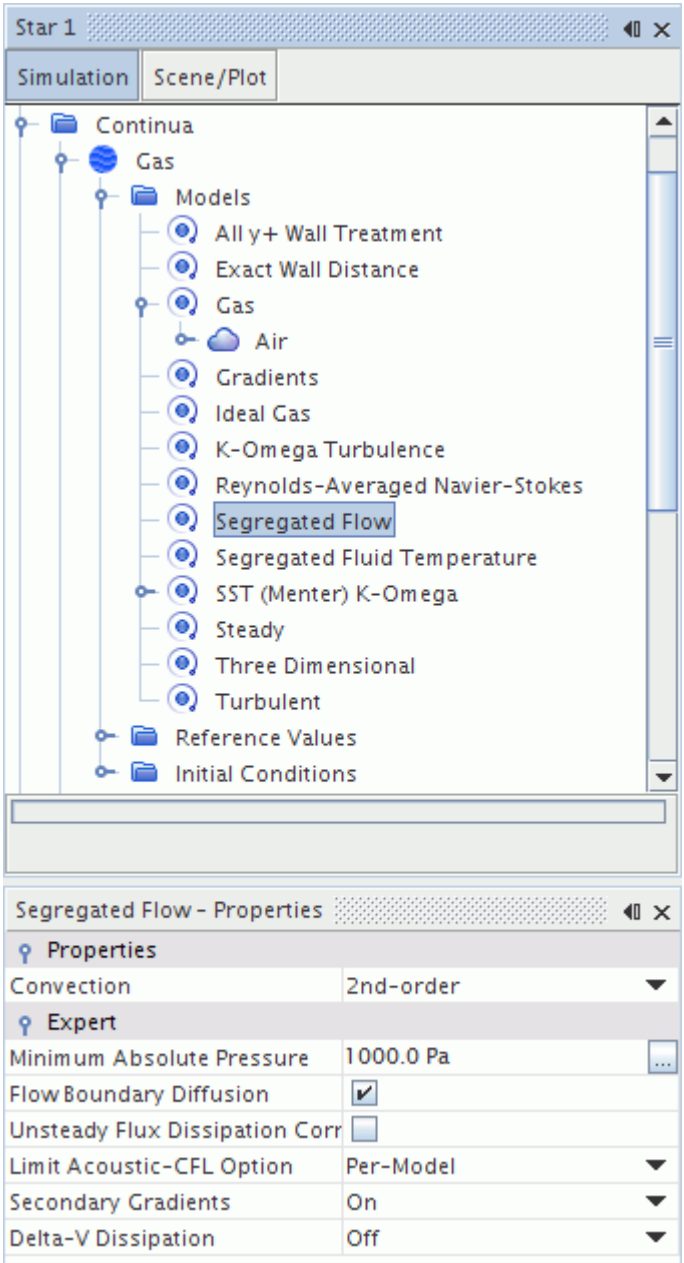
Model nodes are contained within the Models manager nodes of meshing and physics continua. Meshing models control the construction of surface and volume meshes, and physics models define the behavior of the specified material in response to the conditions imposed upon it. Examples of both types of model node are shown below.



Model nodes are added in response to selections made using the model selection dialogs that can be accessed using the pop-up menus of the Models manager nodes. Further information on selecting and reviewing models is given in the following sections:

- Selecting meshing models
- Selecting physics models

Some model nodes have properties that can be set in the *Properties* window. These properties set the default behavior of the model in all regions to which the parent continuum is applied. It may be possible to override this behavior at region or boundary levels if appropriate controls are provided. An example of a physics model with properties is shown below.



Some model nodes have additional options on their right-click menus, in addition to common menu options.

Models Manager Pop-Up Menu

Select Meshing Models...	In the mesh continuum, executes the select models action to allow selection of the models to be activated.

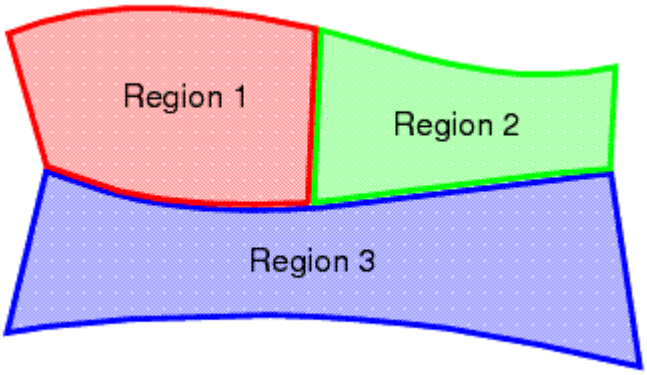
Select Models...	In the physics continuum, executes the select models action to allow selection of the models to be activated.

See also: common menu items.

What Are Regions?

Regions are volume domains (or areas in a two-dimensional case) in space that are completely surrounded by boundaries.

They are not necessarily contiguous, and are discretized by a conformal mesh consisting of connected faces, cells and vertices.



One region is joined to another using an interface, so that information can be passed between them. Boundaries are not shared between regions.

There may be multiple regions in a simulation, each with a unique name and properties, as well as a pop-up menu. Region nodes are created when you:

- Import a mesh -- from native CAD, neutral format, surface mesh (without using geometry parts), or volume mesh formats
- Split a non-contiguous region
- Assign parts to regions, see Assigning Geometry Parts to Regions

Each region node is given a unique name during the import process, but it can be renamed.

The node icon is based on the region type. You can also work with region nodes using drag-and-drop actions.

Using Drag-and-Drop with a Region Node

- Dragging a region node onto another region node will copy all the properties of the source region to the destination region, including region condition methods, but excluding the region name.
- Dragging a region onto the **Parts** node of a scene displayer will add all the boundaries to the scene.
- Dropping a continuum onto a region will associate the region with that continuum (the same as setting the Continuum property).

Region Properties

Index	Index reference for region (read only).

Mesh Continuum	Defines the mesh continuum associated with this region. Select from a drop down list of all the available continua in the simulation.	
Physics Continuum	Defines the physics continuum associated with this region. Select from a drop down list of all the available continua in the simulation.	
Parts	Specifies the parts included in the region, using the parts selection dialog.	
Type	Defines the type of the region depending on the material model selected. This is primarily used to define porosity in a region associated with a gas or liquid physics continuum. A region associated with a solid continuum only allows one option. This property will be changed if the material model of the associated continuum is changed from gas or liquid to solid (and vice versa).	
	Fluid Region	The region is associated with a gas or liquid continuum. No porous modeling is applied.
	Solid Region	The region is associated with a solid continuum.
	Porous Region	The region is associated with a gas or liquid continuum, and porous media modeling assumptions apply.
Allow Per-Part Values	When activated, creates the Parts Groupings node under the region node. All parts in a given group under this node have the same profile assigned to them. See Usage Details.	
	Activated	Creates the **Parts Groupings** node.
	Deactivated	No **Parts Groupings** node is created. This is the default.

Region Pop-Up Menu

Highlight	Highlights a region in a scene. This option is only available if the scene is open.	
Edit...	Opens the table editor, which combines the functionality of the simulation tree and *Properties* window. It displays the selected region, all of its child nodes, and their properties in a single dialog.	
Combine	Executes the combine regions action to combine the selected regions (two or more) into one region.	
Boolean	Applies Boolean operations in the submenu to the selected regions.	
	Unite Regions	Performs a Boolean unite operation.

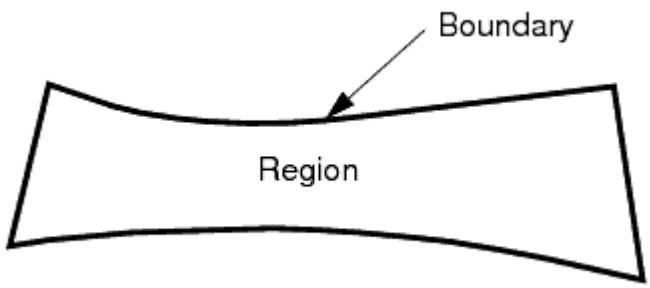
	Intersect Regions	Performs a Boolean intersect operation.
	Subtract...	Performs a Boolean subtract operation.
Split by Surface Topology	Executes the split by surface topology action to split closed, semi-manifold surface meshes that belong to a single region into their constituent regions.	
Split Non-Contiguous	Executes the split non-contiguous region action to split the selected regions into multiple contiguous regions if possible. If more than one region is selected, the action will operate on each region in sequence.	
Split by Function...	Executes the split by function action to split a region according to the rounded integer value of a selected field function.	
Remove Invalid Cells...	Executes the remove invalid cells action to move cells below a user defined threshold quality into another region so that they are not included as part of the analysis.	
Transform	Scales, translates or rotates the selected regions and all associated boundaries. These actions operate on one or multiple regions, and if more than one region is selected, the transform action will be applied to each region in sequence.	
	Scale...	Executes the scale region action.
	Translate...	Executes the translate regions action.
	Rotate...	Executes the rotate region action.
	Reflect...	Executes the reflect region action.
Reorder Mesh	Reorders the internal references to cells in the volume mesh region in order to optimize the solver performance in terms of memory usage.	
Replace Mesh...	Executes the replace mesh action to replace the mesh of the selected region with that of another.	
Solution	Applies operations in the submenu to the individual region.	
	Clear	Executes the clear region solution action.
Create Interface	Creates a heat exchanger interface between selected regions.	

See also: common menu items.

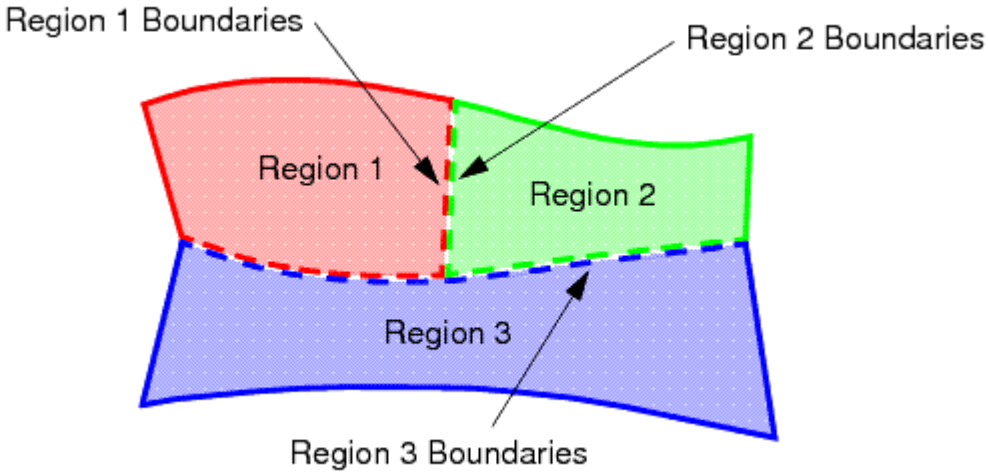
What Are Boundaries?

Boundaries are surfaces (or lines in a two-dimensional case) that completely surround and define a region.

Each boundary has a corresponding node in the simulation tree, and has its own properties and pop-up menu. The figure below shows a simple region where the lines surrounding the region represent the boundaries. In this example, the region could be surrounded by a single boundary or multiple boundaries. The choice depends on what conditions and values need be assigned.



The discretization of a boundary corresponds to the surface bounding the volume mesh for the region. Boundaries are never shared between regions; a boundary can belong to only one region. The figure below shows an example of three separate regions. The dotted lines represent the boundaries that separate each region. The boundaries at the interface of one region and its neighbor are coincident in space.



Boundaries are created by:

- Volume mesh import or surface mesh import;
- Interface creation; or
- Boundary splitting

Only boundaries that belong to a surface mesh can be deleted.

Each boundary is given a unique name during the mesh import process, but it can be renamed, unless it is of the interface type.

Boundary Properties

Index	Index reference for boundary (read only). This index is available through the `Boundary Index` field function.	
Interfaces	The interfaces that were created from this boundary (read-only). Blank if this boundary has not been in any interface creation.	
Part Surfaces	Specifies the surfaces included in the boundary, using the parts selection dialog.	
Type	The boundary type. The full list of possible boundary types is listed below, but the actual list you will see depends on the models you have selected. The list of types changes depending on whether the boundary is a regular one, or one created by an interface.	
	Axis	The axis of an axisymmetric region. The two icons are for a fluid and solid region respectively.
	Flow Split Outlet	A flow split outlet boundary.
	Free-Stream	A free-stream boundary.
	Mass Flow Inlet	A mass flow inlet.
	Pressure Outlet	A pressure outlet.
	Stagnation Inlet	A stagnation inlet.
	Symmetry Plane	A symmetry plane.
	Velocity Inlet	A velocity inlet.
	Wall	A wall.
	Displacement	A solid stress displacement boundary
	Traction	A solid stress traction boundary
	Fixed	A solid stress fixed boundary
	Free	A solid stress free boundary
	Pressure	A solid stress pressure boundary
	Normal Displacement	A solid stress normal displacement boundary

	Force	A solid stress force boundary
	The list of possible types, and the icons, for boundaries generated by interfaces are listed below.	
	Baffle	Generated from a baffle interface.
	Contact	Generated from a contact interface. The two icons are for the master slave boundaries respectively.
	Fan	Generated from a fan interface
	Fully Developed	Generated from a fully developed interface.
	Internal Interface	Generated from an internal interface.
	Porous Baffle	Generated from a porous interface.
	Mixing Plane	Generated from a mixing plane interface.
Parent Interface	The name of the interface that created the boundary for internal use (read-only). Only available on a boundary created by an interface.	

Boundary Pop-Up Menu

Highlight	Highlights a region in a scene. This option is only available if the scene is open.	
Edit...	Opens the table editor, which combines the functionality of the simulation tree and *Properties* window. It displays the selected region, all of its child nodes, and their properties in a single dialog.	
Create Interface	Creates an interface of a particular topology as defined in the submenu. The topology can be changed after creation.	
	In-Place	Creates an in-place interface.
	Periodic	Creates a periodic interface.
	Repeating	Creates a repeating interface.
	Indirect	Create an indirect interface.
Create Shell Region	Creates a shell region from the selected boundary.	

Combine	Executes the combine boundaries action to combines two or more boundaries into one. This is only active when more than one boundary is selected.
Fuse...	Executes the fuse boundaries action to fuse the boundaries together. The selected boundaries from the tree are placed in the Selected list of the *Fuse Boundaries* dialog.
Split Non-Contiguous	Executes the split non-contiguous boundary action to split non-contiguous boundaries (which may not be contiguous in space), into multiple boundaries each of which are contiguous. This will operate on all selected boundaries.
Split By Angle...	Executes the split boundary by angle action to allow you to split a boundary based on the angle between adjacent faces. All the selected boundaries from the simulation tree are placed in the Selected list of the dialog.
Split By Function...	Executes the split boundary by function action to split a boundary based on the value of a field function. All the selected boundaries from the tree are placed in the Selected list of the *Split Boundaries by Function* dialog.
Split by Patch...	Executes the split boundary function action to split a boundary based on the underlying patch definitions of the surface mesh. This function can only be used if more than one patch definition exists in the imported surface.
Split by Feature Curves...	Executes the split boundary function action to split a boundary based on the feature curves in the surface mesh region. Feature curves must exist for the region for this option to be active. All the selected boundaries from the simulation tree are placed in the Selected list of the dialog.
Split Interactively...	Executes the split boundary function action to split a boundary interactively in the current scene. A scene must be open for the option to be active.
Project to Plane...	Executes the project boundary to plane action allowing you project a boundary onto a user-defined plane. All the selected boundaries from the tree are placed in the Selected list of the *Split Boundaries by Function* dialog.
Convert to Interface(s)	Converts the boundary into an interface by creating an in-place interface between the original boundary and a copy in order to allow additional regions to be made. The boundary copy is automatically made by the process.

Create New Region from Boundaries	Create a new region from boundaries that make up a closed volume. Boundaries are removed from the original region and moved into the new region definition.	
Solution	Applies operations in the submenu to the individual region.	
	Clear	Executes the clear region solution action.

See also: common menu items.

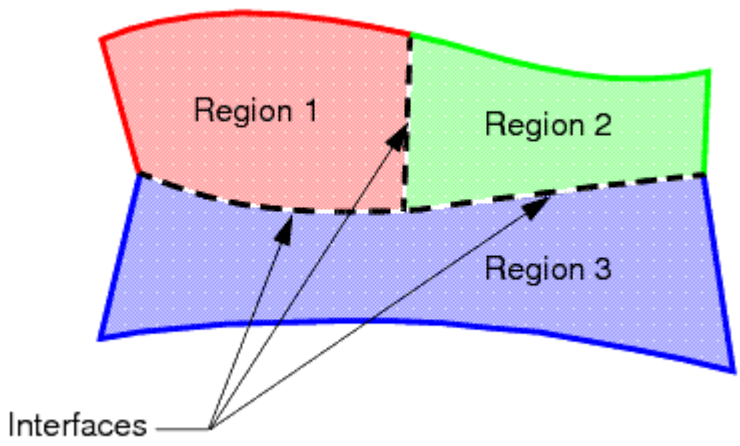
What Are Interfaces?

Interfaces provide a connection between boundaries during the simulation meshing setup and/or analysis process.

They are always composed of boundary pairs, either in the same region or in different regions. There can be many (or zero) interfaces in a simulation.

When using the surface remesher, tetrahedral, polyhedral or prism layer meshers, interfaces may be defined to ensure that a conformal mesh is obtained between regions or within the same region if a baffle plane is required.

In a volume mesh, interfaces permit mass, energy and other continuum quantities to pass from one region to another. In the illustration below, the dotted black line indicates the location of the interfaces that are created using the coincident boundaries between each region:



Interfaces have a topology and a type associated with them. In general, the topology refers to the physical arrangement of the interface boundaries. For example, in-place and repeating interfaces normally have coincident boundaries while a periodic interface has separated boundary pairs.

The interface type determines the modeling behavior of the interface. For example, depending on the topology selected, you can:

- Allow mass and energy to be exchanged between fluid and porous regions that have coincident boundary planes.
- Simulate baffles/porous baffles within a region.
- Provide a contact interface for heat transfer between fluid-solid and solid-solid regions.
- Specify a fully developed flow condition to simulate the pressure drop or mass-flow across a periodic (cyclic) across a region.

See Also:

What Is a Generalized Cylinder?

What Are Solvers?

In STAR-CCM+, solvers control the solution and are activated once per iteration (or once per time-step for explicit unsteady simulations).

Typically, a model elects the solvers that are required. Different models can use the same solver and sometimes models need more than one solver.

Some models control other models and some only perform tasks when the solution is initialized. For this reason, not all models elect solvers. Models can also elect solvers that perform subsidiary tasks in addition to the solver that the model elected to control it.

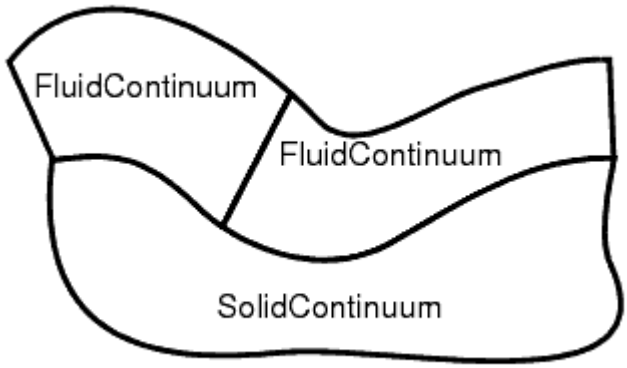
Contents:

Why Are Solvers Different From Models?

Why Are Solvers Different From Models?

The essential difference between a solver and a model is that the scope of a model is limited to the continuum in which is defined, while the solver can span continua.

This distinction between solvers and models facilitates the simultaneous solution of physical phenomena across regions that are associated with different continua, such as conjugate heat transfer. The figure below, which shows two continua, illustrates this distinction:



- one region that is called Fluid Continuum, representing a fluid
- one region that is called Solid Continuum, representing a solid

Suppose the following are defined:

- a flow model and a fluid energy model in a Fluid Continuum
- a solid energy model in a Solid Continuum

A solver would be activated in the Fluid Continuum to control the flow solution in that continuum and another solver would be activated in Solid Continuum to control the energy solution. In this situation, a single solver would control two different models (fluid energy and solid energy) in both continua, which allows the implicit solution of the conjugate heat transfer.

What Are Reports, Monitors and Plots?

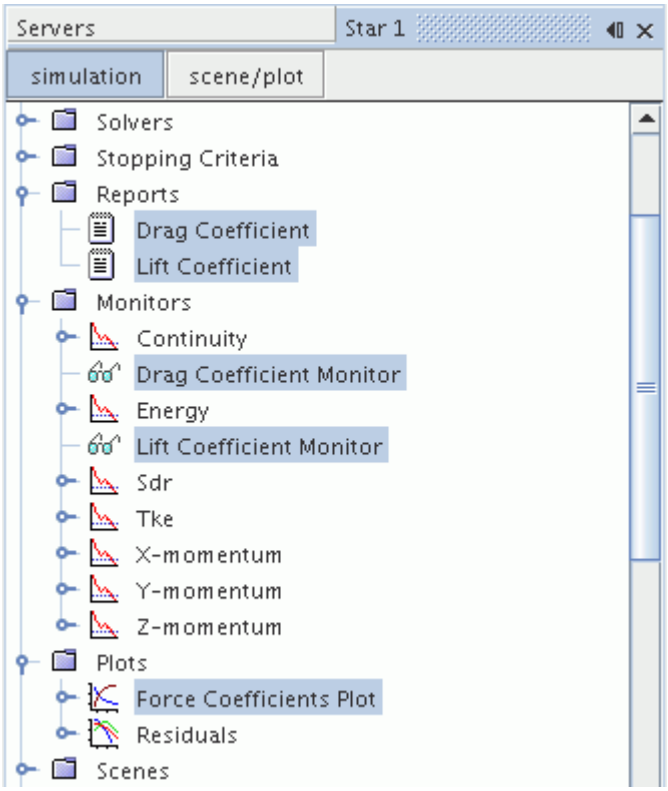
Reports provide a computed summary of simulation or system data.

They are contained within the **Reports** manager node in the object tree, and have varying properties that depend on their type.

Monitors provide a mechanism for sampling summary information from the simulation during runtime. Some built-in monitors are provided for residuals that the various solvers compute. User-defined monitors can be created based on reports that are contained within the Reports manager node.

Plots provide line graphs containing information that is supplied from; monitors, extracted simulation data, tables, or derived data sets. These plots can be customized with different line and point properties, axes, scales, titles, and so on.

An example object tree showing all three of these object types is shown below. In this particular example, the Drag and Lift Coefficient reports have been used to create monitors, which in turn have been used within a single plot called Force Coefficients Plot.



What Are Scenes?

Scenes are used to visualize the geometry, mesh, and solution data that is contained within a simulation.

They are viewed through the *Graphics* window in general, and can also be written to file for a hard copy. Four general types of scene can be created:

- Geometry scenes to view outlines and surfaces of geometry parts and regions.
- Mesh scenes to view the surface and volume mesh grids on the surfaces of parts and cell sets.
- Scalar scenes to view contours of scalar quantities on part surfaces.
- Vector scenes to view arrow vectors generated from vector quantities on part surfaces.

Scenes are often used with derived parts to obtain solution quantities at particular locations within the simulation. Streamlines can also be created through the streamline derived part.

Working with Simulation Objects

Objects are the key components of STAR-CCM+. They represent various aspects of a simulation and have properties that you can set as required for your particular problem.

Specific topics addressed here are:

Contents:

Working with the Object Tree
The Object Selection Dialog
Comparing Objects and Simulation Trees
Using Custom Simulation Trees
Tagging Simulation Objects
Object Filters
Using Query-Based Selection

Working with the Object Tree

The objects in STAR-CCM+ are presented as nodes on a tree view, and their properties are presented in a spreadsheet-like view tightly linked to the selected node in the tree.

This section includes explanations of how to navigate around and use the tree view, as well as how to create, copy, paste, delete, rename, sort, and group objects in the simulation tree, and copy properties among them. Information about object dependencies is included.

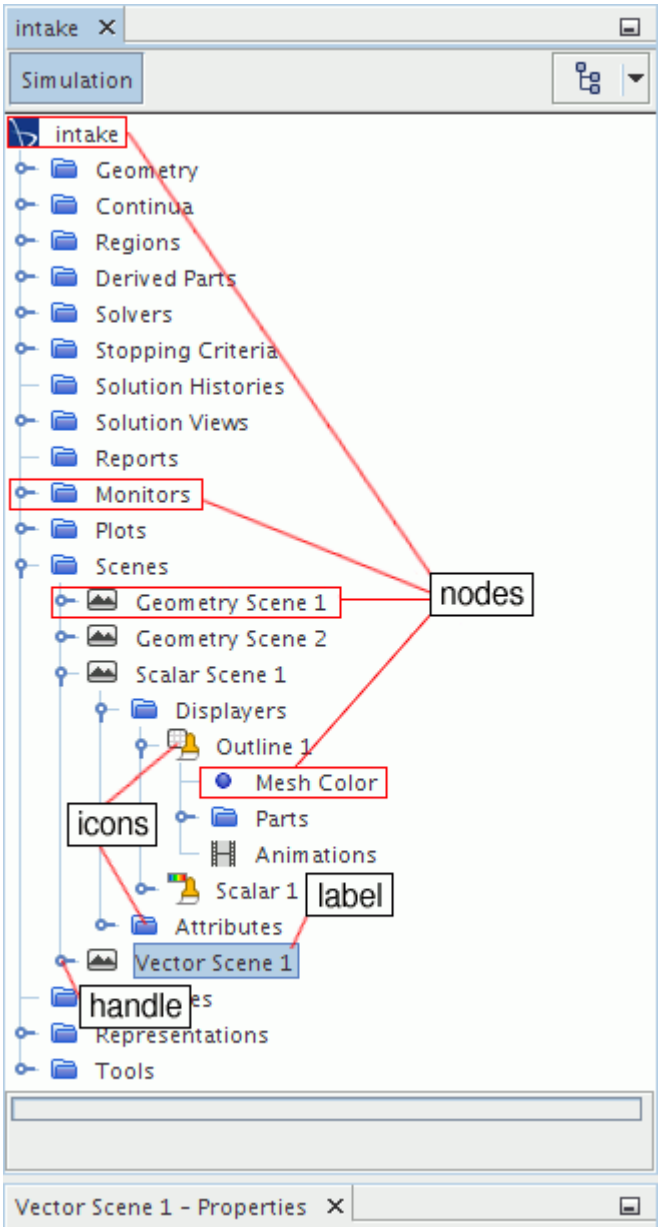
Contents:

Interacting with the Object Tree
Object Dependencies
Copying and Pasting Objects
Deleting Objects
Renaming Objects
Sorting Objects
Grouping Objects
Common Pop-Up Menu Items

Interacting with the Object Tree

This section gives guidance on how to work with and navigate around object trees.

Tree views appear in several places in STAR-CCM+.

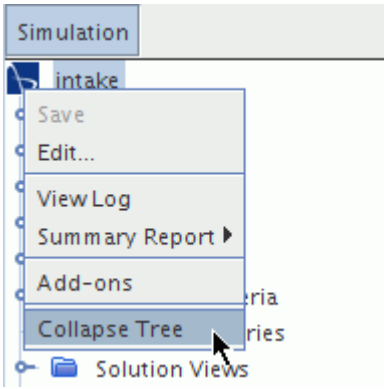


A folder ( ) icon indicates that the node is a “manager” or “collection” of several nodes of the same type. For example, the **Scenes** node is a manager for several individual scene nodes.

You can expand or collapse nodes by clicking the handle to the left of the node icon. STAR-CCM+ provides keyboard shortcuts for the same actions: `<Left>` arrow closes nodes, `<Right>` arrow opens nodes. Adding `<Ctrl>` to either action applies the command to all child nodes and their descendants.

If a tree node has any associated actions, right-clicking the node opens a pop-up menu. If multiple nodes are selected and the actions do not apply to all selected nodes, some menu items are grayed.

To shrink the tree back to its default size, right-click the simulation node at the root of the object tree and select **Collapse Tree** from the pop-up menu.



Drag-and-Drop

You can drag-and-drop objects in STAR-CCM+ to perform **copy operations**.

The mouse cursor changes to indicate whether a drop is valid.

Scrolling Up or Down

Move the mouse cursor near the top or bottom of the tree window. If necessary, move slightly outside the border to start scrolling.

Opening Nodes with Children

Hold the mouse over the node for a second or so, and the node opens.

Navigating the Selection History of the Tree

After selecting a variety of nodes in the tree, you can navigate the selection history as you would navigate the browsing history in a web browser: moving forward or backward in the range of selections.

Note: If you go back in the selection history and then make a different selection, that selection becomes the end of the selection history. The selections through which you moved back are lost.

The available navigation techniques are as follows:

Workspace Context	Previous Selection	Next Selection
Edit menu	**Previous Tree Selection**	**Next Tree Selection**
Edit toolbar	◁ (**Previous Selection**)	▷ (**Next Selection**)
Keyboard shortcuts	`<Alt>+<Shift>+<Left>` arrow key	`<Alt>+<Shift>+<Right>` arrow key

Note: If you are at the beginning or end of the selection history, or if you delete an object somewhere within the selection history, these options can be unavailable, and the affected menu item and toolbar button are grayed out.

Additionally, this feature has the following characteristics:

- It works with custom tree views in addition to the main simulation tree.
- Each tree view has its own selection history.

- Multi-selection of objects is treated as a single selection step. When you navigate to that action, all nodes in the multi-selection will appear selected.

Searching the Tree

You can search the tree using keystrokes.

If the focus is on the tree, that is, on one of its nodes, pressing an alphanumeric key (lower case only) has the following effects:

- Moving the focus to a node beginning with the typed character (if one exists)
- Opening a search edit box with that character

If no match is found, then no node is selected.

The search facility is not case-sensitive.

Object Dependencies

When an object in STAR-CCM+ requires data from another, a dependency relationship is formed between the objects.

Dependency relationships affect which objects you can delete and which you cannot, as well as the consequences of modifying an object. Examples include:

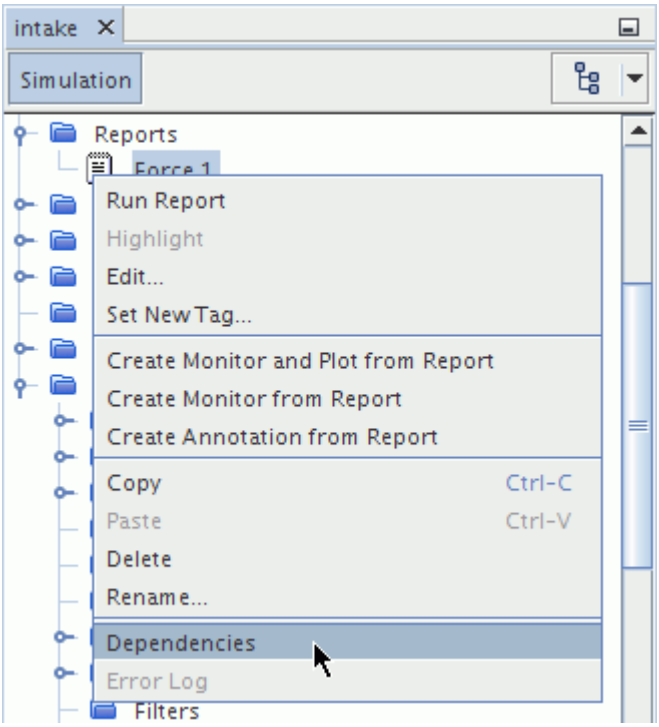
- A monitor plot showing data from a monitor
- A scalar scene displaying data from a field function
- A report using data from a coordinate system

Most types of objects in STAR-CCM+ have dependency relationships.

Checking an Object for Dependencies

To check whether an object has dependencies:

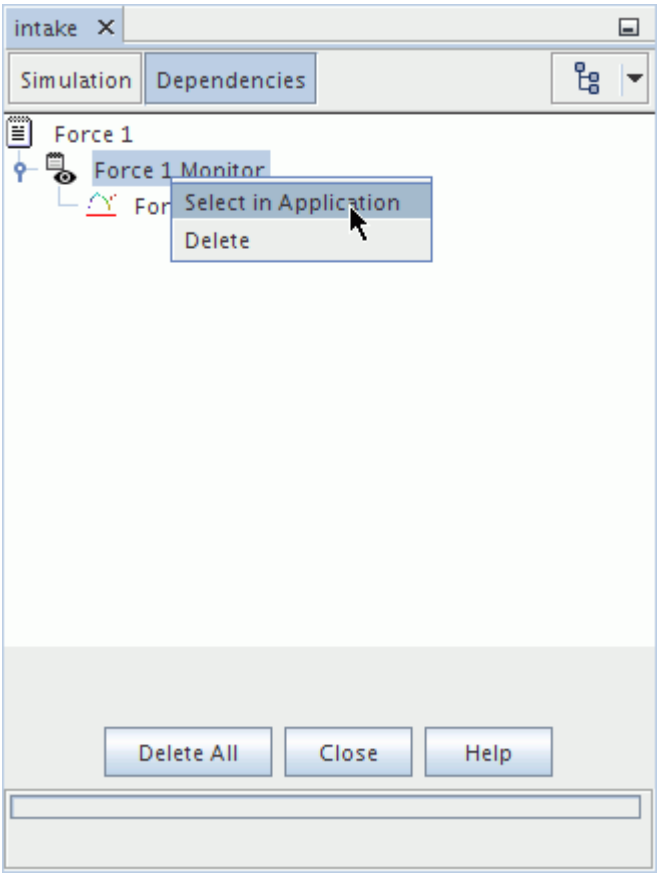
1. Right-click the node of that object.
2. If the pop-up menu includes a **Dependencies** item, select it.



Note: This menu item only appears in the pop-up menu of an object when that object has dependencies.

Before you delete or modify an object, you can assess the potential impact of your change by looking at the dependent objects. To examine an object under the *Dependencies* tab:

1. Right-click its node and choose **Select in Application**.



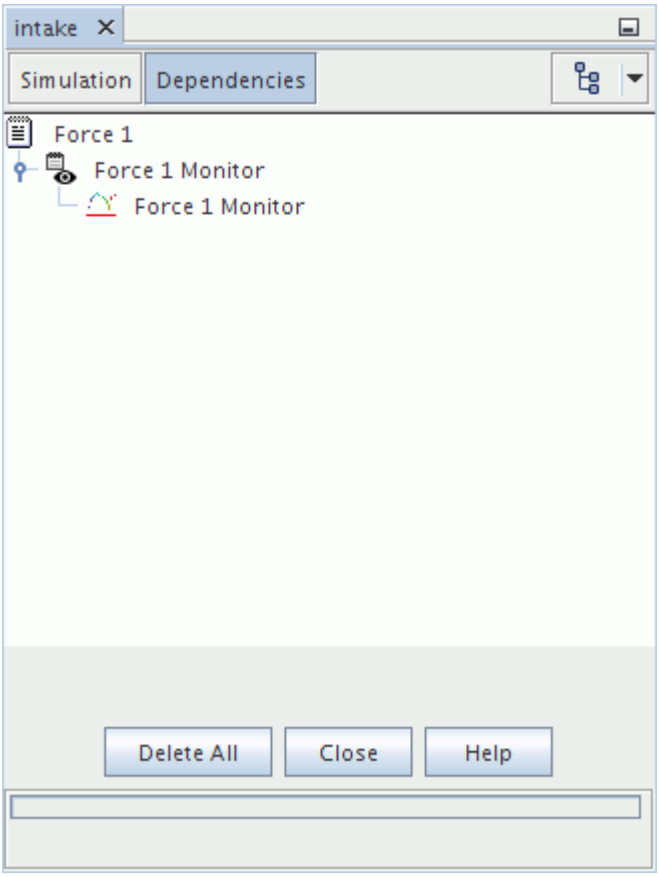
STAR-CCM+ locates the selected object in the object tree.

2. When you are finished working with the *Dependencies* tab, click **Close**.

Deleting Objects with Dependencies

To delete an object under the *Dependencies* tab:

- Examine the node hierarchy. Each parent node is referenced for data by its immediate sub-node.



- To delete all the objects, click **Delete All**. Click **Yes** in the confirmation dialog.
- To delete the object at the bottom of the hierarchy (which has no dependencies), right-click its node and select **Delete**. Click **Yes** in the confirmation dialog.

  Deleting any node in the hierarchy other than the one at the bottom also deletes all of its dependencies.
- To delete an object and all of its dependencies, right-click its node and select **Delete**. The confirmation dialog will ask whether you want to delete all of those objects. If so, click **Yes**.

Copying and Pasting Objects

To give one node the same characteristics (such as properties or child nodes) as another of the same type, copy one node over the other using the clipboard or drag-and-drop.

Copying objects does not always transfer dependent objects automatically. In particular, when you copy an object from one simulation to another, copy all of its dependent objects first. For example, to copy a scene displaying derived parts into another simulation, you would have to copy the derived parts beforehand so that the scene could access them properly.

The specifics about what gets copied (some properties, all properties, or child objects) are detailed in the user guide explanation of the specific object.

Copying and pasting objects using the Clipboard

Select the node from which you want to copy properties and/or child objects, and press `<Ctrl><C>` on the keyboard, select the **Edit** > **Copy** menu item or click the (**Copy**) toolbar button.

With some nodes, you can also right-click on the node and select **Copy** from the pop-up menu.

Creating a new Object by Pasting

Select a manager and press `<Ctrl><V>`, click the (**Paste**) toolbar button, or right-click on the node and select **Paste** from the pop-up menu. This will create a new object named Copy of <copied node>, with the same properties as the copied object.

Copying Properties to an Existing Object

Select the existing object you want to copy the properties to and **Paste**. The node where you applied the **Paste** command resembles the node from which you copied properties.

Using Drag-and-Drop

Click the node that has been set up and, while still holding down the left mouse button, drag the mouse until the pointer hovers over the target node.

To finish copying, release the mouse button.

If the target node is a manager, then a copy of the initial object is created. Otherwise, the node properties are simply copied from the first to the target node.

Deleting Objects

Many of the objects on the tree that are added through importing a mesh or setting up the simulation can be deleted.

To delete one or multiple objects select them and then select **Edit** > **Delete**, use the pop-up menu by right-clicking and selecting **Delete**, or use the `<Delete>` key on the keyboard.

If the delete action worked, then a confirmation dialog is displayed.

If nothing happens, then one or more of the selected objects currently cannot be deleted for some reason.

Use this option with care, since it removes the object permanently; the deleted objects are not moved to the Trash or Recycle Bin of your computer.

Objects That Cannot Be Deleted

Some objects cannot be deleted. These include:

- All the objects that are automatically created on the tree of an empty simulation
- Boundaries that are used as interfaces
- Objects that other objects use, such as when a monitor plot receives data from a monitor, or a continuum is assigned to a region

When an object uses another object, there is a dependency. When you attempt to delete an object that has a dependency, STAR-CCM+ opens a warning dialog.

To find out what the dependency is, click **Details**.

Renaming Objects

Most objects on the tree can be renamed.

Typically manager ( ) objects cannot be renamed. Bear in mind that most objects within one simulation must have unique names. The uniqueness rules are case-sensitive, so `Wall-1` is different from `wall-1`. If you do inadvertently use a duplicate name for an object, STAR-CCM+ appends a space and a digit to what you typed so that the result is a name that is still unique.

One major exception to this rule, however, is with region and boundary names. You can have a region named `Element` and a boundary named `Element` in the same simulation. Also, you can give the same name to multiple boundaries, provided each of those boundaries is within a different region.

To change the name of an object, you have three options:

- Selecting the node and pressing `<F2>`
- Slowly clicking the label of the node
- Right-clicking the node and selecting **Rename...**

Any of these options activates the inline editor in the label of the node.

To apply your change, press `<Enter>`

To keep the existing name, even if you have changed the name in the inline editor, press `<Esc>` or click outside the node label.

When you rename a node, it is not sorted with its siblings. To alphabetize the nodes, use the sort action.

Sorting Objects

You can sort child nodes of a selected object alphabetically.

This feature is typically available to manager nodes and would be used if any of the sub-nodes were renamed. Nodes are not sorted automatically after a name change, and this action will force a sort. You can sort by pressing `<F5>` or by right-clicking a manager node and selecting **Sort**.

Linux (Unix) and Windows systems sort differently:

- Linux systems sort taking case into account with upper case being “smaller” than lower case (Wall, cyclic).
- Windows systems sort by ignoring case (cyclic, Wall).

Grouping Objects

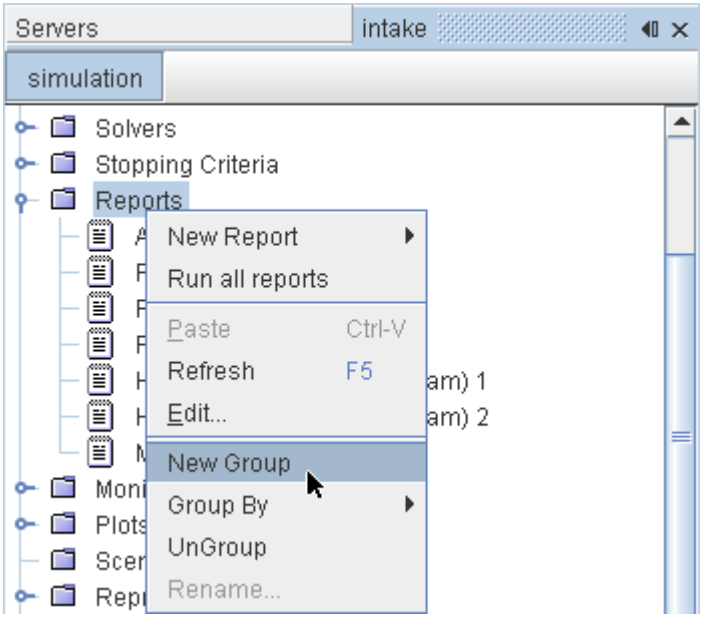
When an object tree contains many nodes of the same type, such as interfaces, it helps to be able to organize them into groups.

The following types of objects can be handled this way by creating subfolder nodes in their manager nodes:

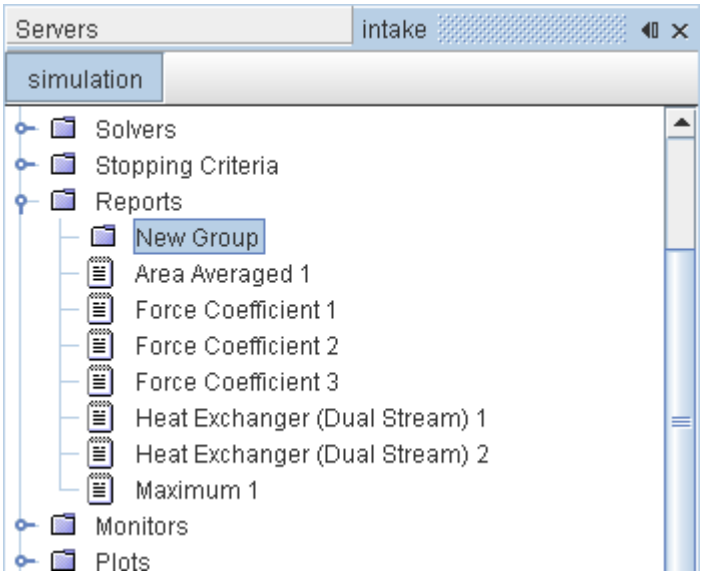
- Regions
- Boundaries
- Interfaces
- Derived parts
- Reports
- Monitors
- Plots
- Scenes

Placement of objects in a group has no effect on how they function in the simulation.

To use the feature right-click, for example, the **Reports** node and select **New Group**.

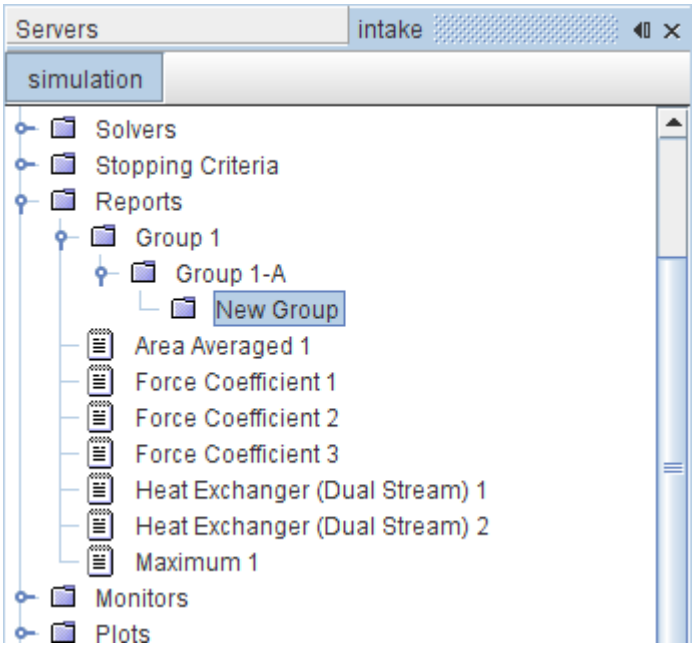


A **New Group** node appears in the object tree.



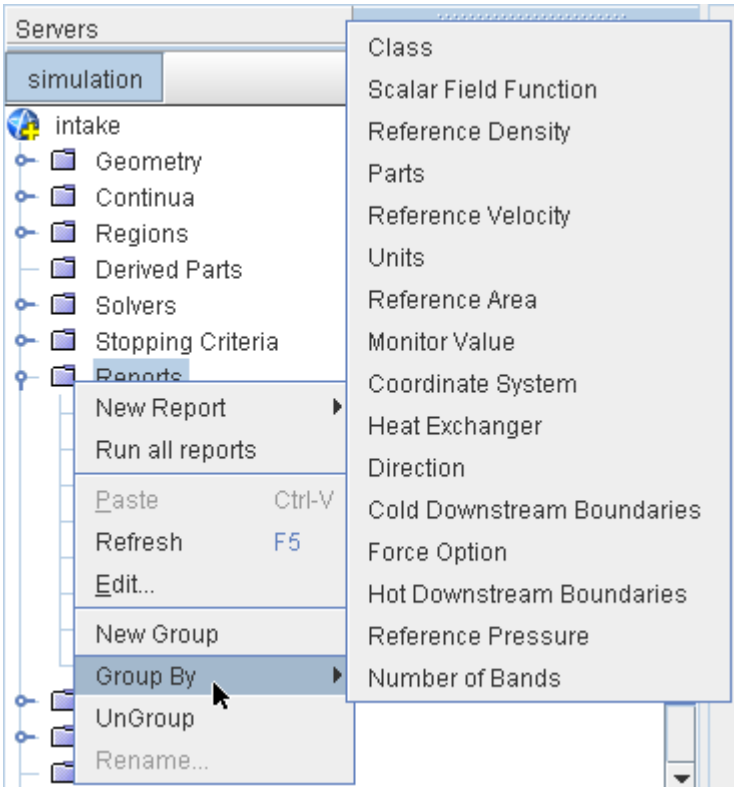
You can then drag-and-drop nodes into this group node. It is possible to rename the group node to represent its contents.

If you have a set of items so large that you want subcategories, it is also possible to create groups within groups.



To reverse the process, right-click the manager node (**Reports** in this example) and select **UnGroup**. This action moves the objects (the report nodes in this example) back to their original location, and automatically deletes any empty group nodes.

The **Group By** pop-up menu item is an alternative way to use this feature. It takes all the properties of the objects in that manager node and makes them available as group categories.



Group nodes that are created through this automated, property-based approach can also have subfolder nodes that are created within them.

Groupings are only possible based on one property at a time. Pre-existing group nodes, whether they were created manually or automatically, are deleted as the objects are moved into the new groups.

Common Pop-Up Menu Items

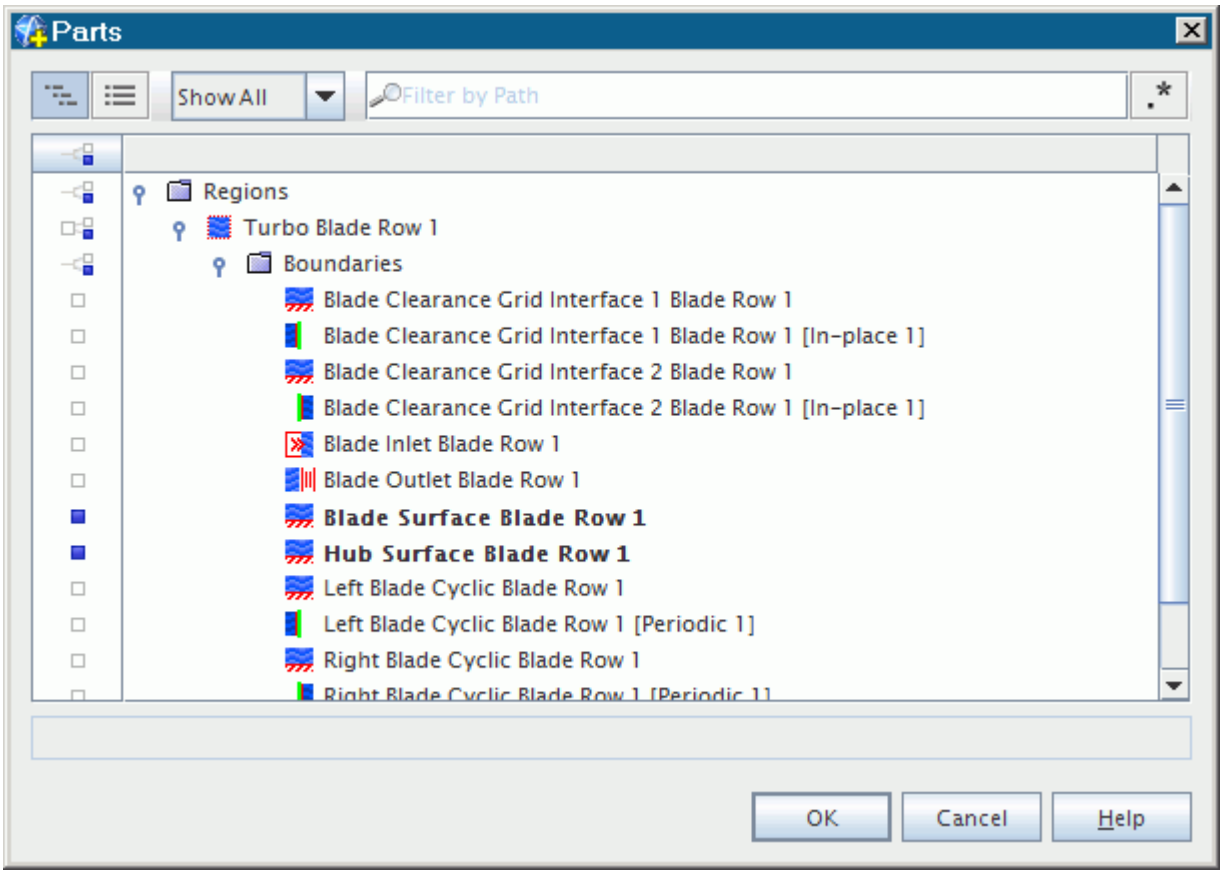
Many objects on the simulation tree have items in the pop-up menu that are common. This section lists all those items, although any single object does not necessarily use all of the menu items.

Edit	Allows customizing of the selected object and its children using a separate modeless stay-on-top dialog.
Copy	Copies the object and selected properties onto the clipboard.
Delete	Removes the selected objects.
Highlight	Highlights an object in a scene. This option is only available if the scene is open.
Paste	Replaces the properties of the current object by pasting the ones from the clipboard.

Refresh	Updates the selected object or display so it reflects the latest changes to the simulation.
Rename	Allows you to change the name of the currently selected object.
Sort	Sorts child nodes in alphabetical order. This item is typically available on manager nodes.

The Object Selection Dialog

Certain options in STAR-CCM+ allow selection of objects from an object selection dialog.



Using this dialog includes the following operations:

Contents:

Activating the Object Selection Dialog
Accessing Objects in the Dialog
Selecting Objects
Filtering Displayed Objects
Regular Expressions
Single-Selection Mode
Alternate Formats

See Also:

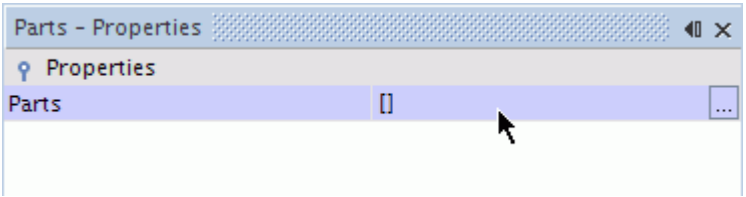
Using the Properties Window

Activating the Object Selection Dialog

For the specific method to open the object selection dialog, see the instructions of the particular STAR-CCM+ feature that you are using.

In general, to open the dialog:

- Click the right half of a *Parts* property (or a similar property by another name), or click ... (**Custom Editor**).

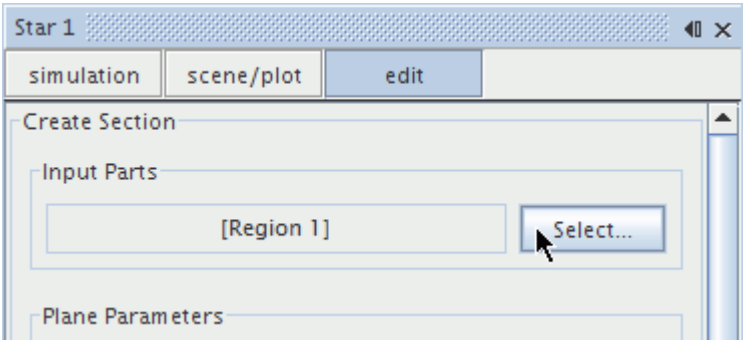


  When you click the property field for small sets of objects (10 or less), the object selector appears as a drop-down list by default.

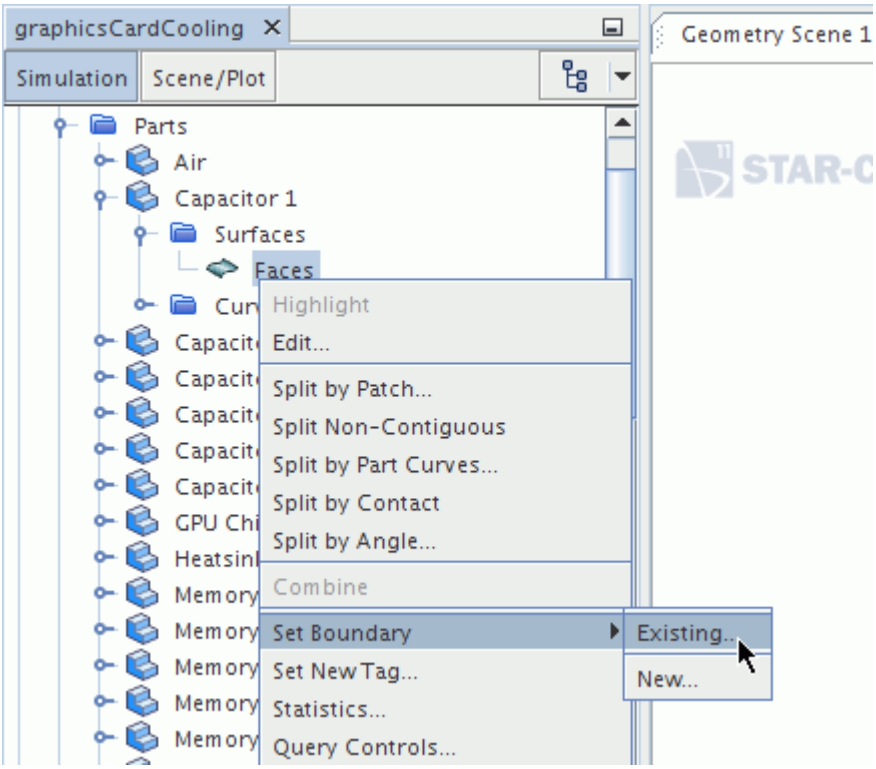
  In single-select mode, the list disappears when you make a selection. In multi-select mode, the list disappears when you press `<Enter>` or when the list no longer has the focus. In either mode, the list disappears and the changes are cancelled when you press `<Esc>`.

  To access the dialog for a small set, click ... (**Custom Editor**).

- Click a specific button in a dialog, such as **Select** for *Input Parts*.



- Select a menu item that accesses a set of objects in the simulation, such as **Set Boundary** > **Existing**.



When you select such a menu item for small sets of objects (10 or less), the object selector appears as a set of submenu items by default.

In single-select mode, the list disappears when you make a selection. In multi-select mode, the list disappears when you press `<Enter>` or when the list no longer has the focus. In either mode, the list disappears and the changes are cancelled when you press `<Esc>`.

Accessing Objects in the Dialog

Use the Tree view or the List view. In either view, you can use the `<Ctrl><Click>` technique to highlight objects without selecting them.

The Tree View

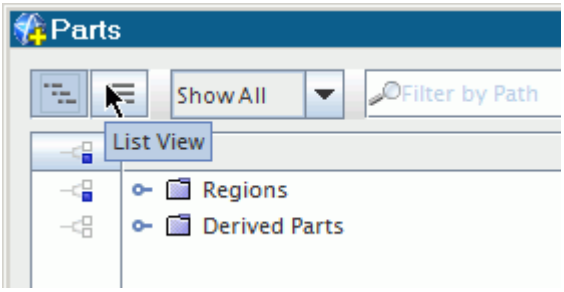
By default, the objects in the dialog are arranged in a hierarchical node tree similar to what you see in a simulation window.

- To open the tree view of an object and all its children at once, select **Expand** from the pop-up menu of the object.
- To open objects selectively, use the handle ( ).
- To identify where in the node hierarchy an object is, hover over it with the mouse pointer in the view panel. The path field, a read-only text box near the bottom of the dialog, shows the location.

  The text that is directly above the tree window displays the portion of the path that all objects in the tree share in common and is not duplicated in the path field beneath the tree.

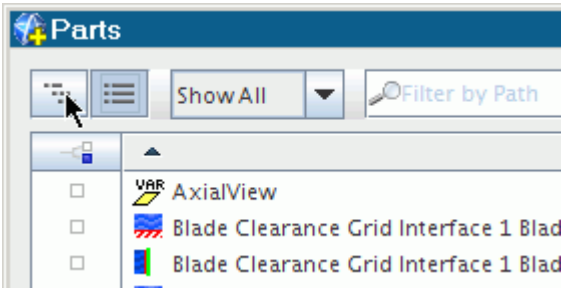
The List View

If you know the name of the object that you want to select, switch to the **List View** by clicking its button.



With the objects in an alphabetical list, scroll until you find the desired object. If the list is long, filter the displayed objects.

To return to the tree view, click the **Tree View** button.



Contents:

Highlighting Objects

Highlighting Objects

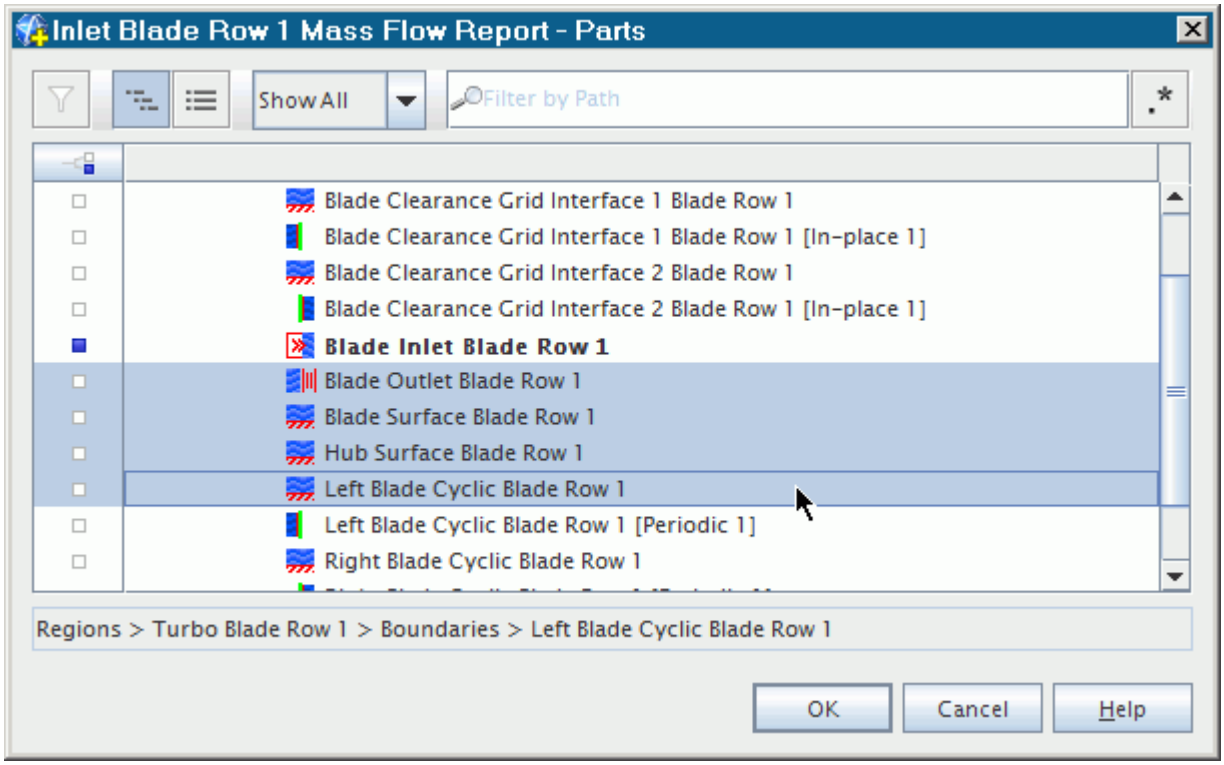
You can highlight one or more objects in the view panel without selecting them.

To highlight one object or a non-contiguous group of objects:

1. Click in the row of each object while holding down the `<Ctrl>` key.

To highlight a contiguous group of objects:

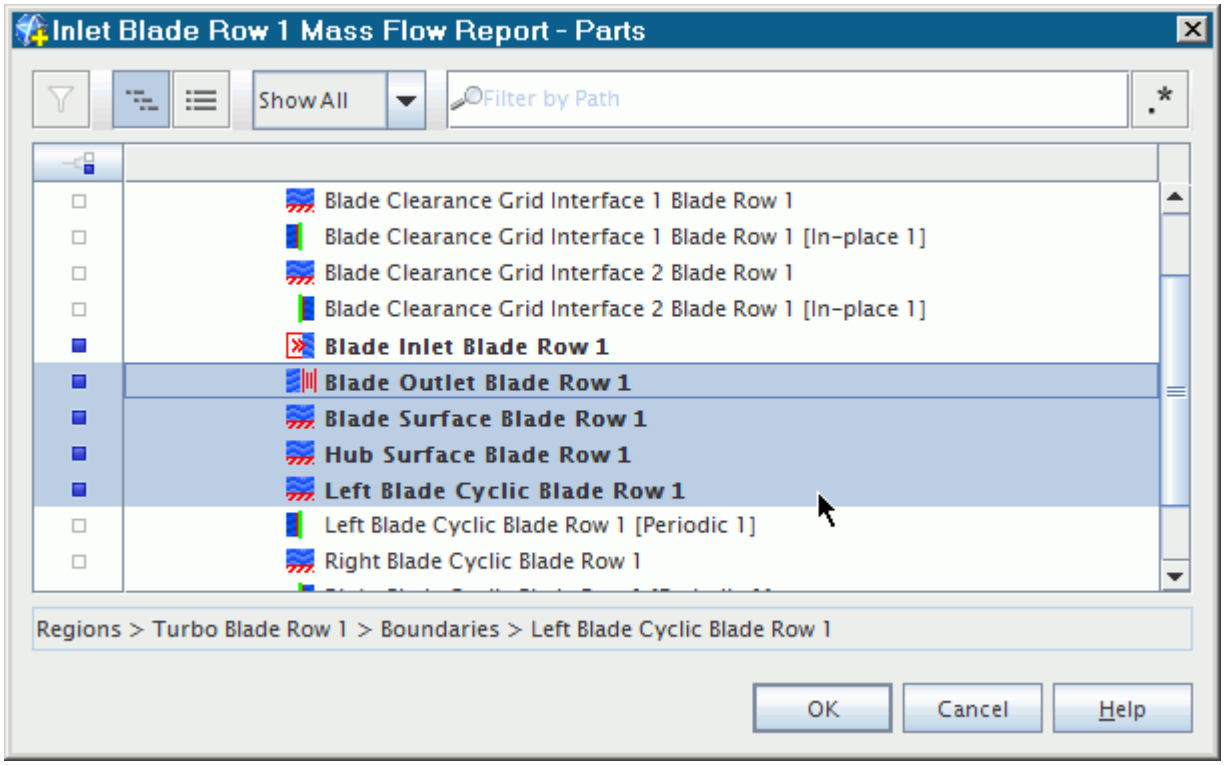
2. Click the first object in the group while holding down the `<Shift>` key or the `<Ctrl>` key.
3. Click the last object in the group while holding down the `<Shift>` key.



To remove the highlighting, use one of the following techniques:

4.
   - Click anywhere in the view panel outside the highlighted group of objects.

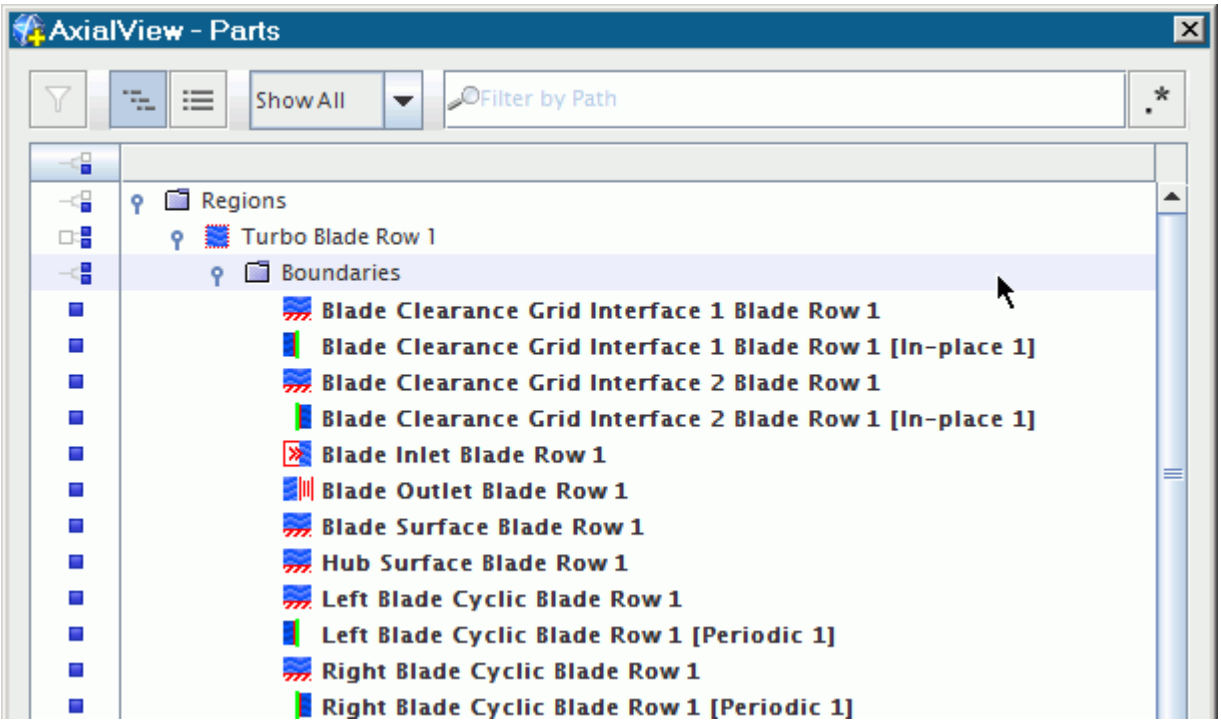
     If you click an object in this situation, the selection state of the object does not change.
   - Right-click the highlighted group and select **Remove Highlight** from the pop-up menu.
5. To select the highlighted items, click any one of them.



Selecting Objects

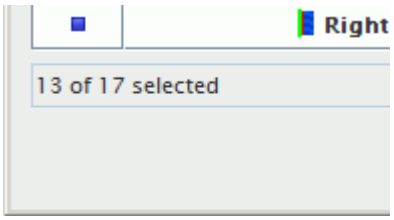
You can select objects in an object selector dialog using a variety of techniques, depending on the hierarchy of the object that you want to select.

- To select an object, click anywhere in the row of that object.
- To deselect the object, click in the row of the object again.
- To select or deselect all the children of manager objects that have no solution or mesh data (such as a **Boundaries** manager), click anywhere in the row of the manager object.

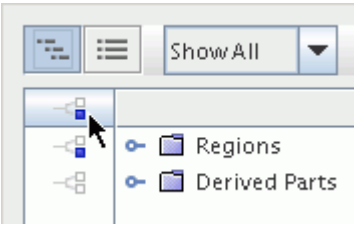


Note: A different procedure exists when parent objects, such as regions, are available to select. Selection transitions that occur in manager or parent objects can only affect visible children. Children that are currently filtered out of the view are not affected.

A counter at the bottom left of the object selector dialog indicates the number of objects selected. It only appears when your mouse pointer is not hovering over an object or highlighting it.



- To select all objects in the dialog, click the **Column Selection** button in the upper left portion of the dialog. This global selection also applies to all objects that are contained within objects.



- To deselect all selections, click the **Column Selection** button a second time.
- To apply your selections, click **OK**.

- To close the dialog without applying the selections, click **Cancel**.

Note: When you choose multiple nodes before accessing the object selector dialog on a common property of the nodes, the dialog displays the intersection of any objects that are available or already selected. For example, if you choose multiple table nodes and open the object selector dialog to set the *Parts* property, STAR-CCM+ displays the intersection of any objects that were already selected for the individual nodes.

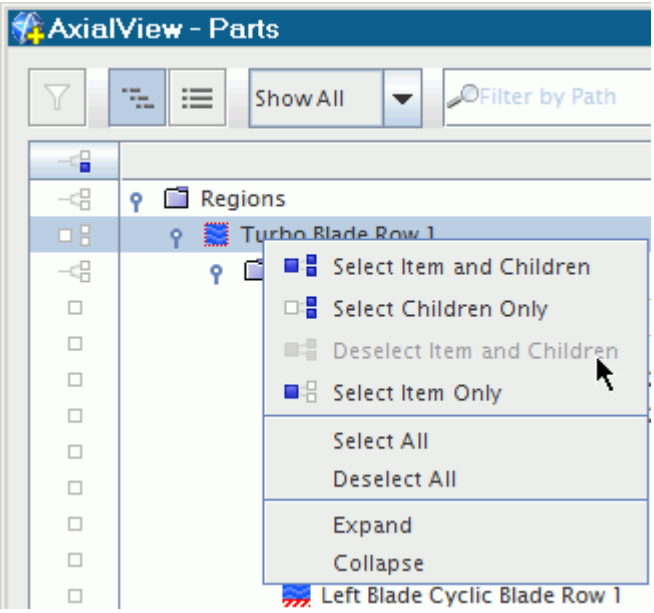
Contents:

Selecting Certain Parent and Child Objects (Tree View Only)
Selecting Objects Using the Clipboard

Selecting Certain Parent and Child Objects (Tree View Only)

Parent objects that can contain solution or mesh data (such as a region) are available for selection, and there are four different states of selection possible.

To choose one of the selection states, right-click the parent object. The current state of the object is unavailable (grayed out).



Alternatively, you can click the row of the parent object to cycle through the following states.

1. Click once to select the parent only, leaving the selection status of the children unchanged.
2. Click a second time to select all (children and parent).
3. Click a third time to deselect the parent (leaving only the children selected).
4. Click a fourth time to deselect all (children and parent).
5. On the fifth click, the sequence repeats to select the parent only.

Selecting Objects Using the Clipboard

While working on the setup of a simulation, you can collect simulation objects, such as parts, on the fly for later use in, for example, reports, derived parts, regions, or summaries. You copy these objects directly from a scene or the simulation tree to a clipboard. The collected clipboard items can then populate the list in an object selection dialog.

For example:

- While navigating a scene, you detect two cylinders for which you want to run a report.
- While navigating the simulation tree, you detect ten lug nut parts to assign to a region.

To select objects using the clipboard:

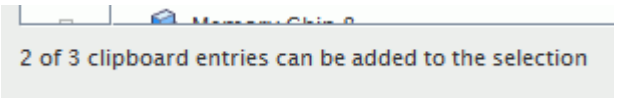
1. Within a scene or the simulation tree, select the objects that you plan to use further in an object selection dialog.

   Note: To select parts in a scene display, make sure that you select the proper type of part. For example, if you select part surfaces and a derived part, those objects together cannot be used as inputs for an action such as combining. To filter objects that you select in the scene, see Filtering the Selections.

2. Copy the objects to the clipboard (using the context menu item, the toolbar button, or `<Ctrl><C>`).
3. Activate the object selector in the property (for example, *Parts*) of the node that is supposed to access the objects on the clipboard (see Activating the Object Selection Dialog).
4. To paste a selection of the objects contained on the clipboard according to a filter, apply the filter before you activate the clipboard selection. See Filtering Using Filter Objects.
5. In the object selection dialog, right-click in the view panel and choose **Select via Clipboard**.

   Alternatively, click (**Select via Clipboard**).

   As you hover with the mouse pointer over the **Select via Clipboard** button, a message in the lower left corner of the dialog indicates how many objects are available on the clipboard.

   The available number of parts depends on the domain of the object selector. For example, if you click the *Parts* property of a region, the object selector expects parts. If you select two parts and one region in your clipboard, only two of the clipboard items (the two parts) are considered compatible. The region cannot be added to the *Parts* property of a region.

   After you click the **Select via Clipboard** button, the objects on the clipboard appear in the list of selected objects.

Filtering Displayed Objects

STAR-CCM+ provides multiple ways of filtering the displayed objects in order to make selection easier.

To view only the objects that are selected or unselected, use the Filter menu. To filter specific objects by path, enter text into the **Filter by Path** text box near the top of the dialog. To see results without the need to open parent objects, use the List view.

You can use either plain text, regular expressions, or query filters.

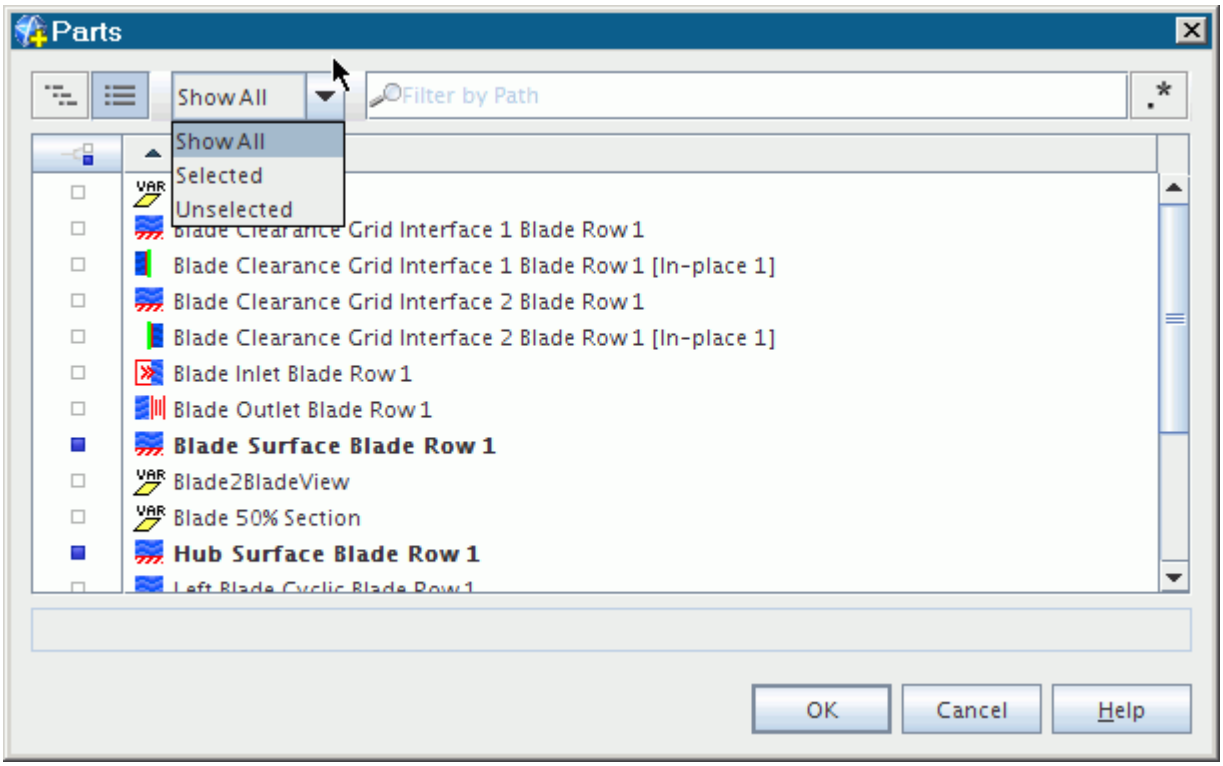
Contents:

Filtering Views
Filtering by Path Using Plain Text
Filtering Using Regular Expressions
Filtering Using Filter Objects

Filtering Views

To view only the objects that are selected or unselected, use the menu in the upper left part of the dialog.

To view results without opening parent objects, use the List view. By default, the menu is set to **Show All**.



Note: In the Tree view, if an item with children matches the filter, the item and all related children are displayed regardless of whether they themselves match the criteria. In the List view, only items that match the criteria are shown.

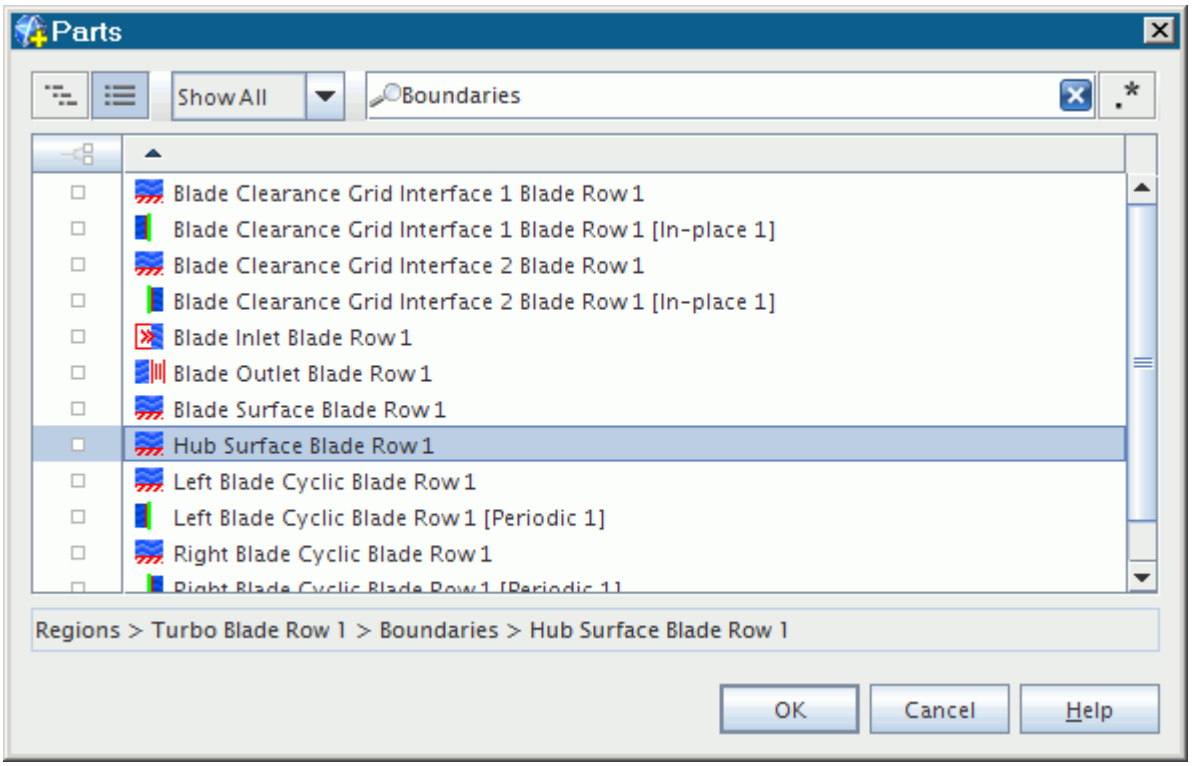
- To view only the objects that are selected, use the **Selected** option.
- To view only the unselected objects, use the **Unselected** option.

Filtering by Path Using Plain Text

To filter the objects of interest based on their node paths, enter text that is a part of the paths of the objects. Plain text filtering is not case-sensitive, but regular expressions are case-sensitive by default.

The full path is shown at the bottom of the dialog when you hover the mouse over an object in the list.

The following example is a search for boundary objects using the text `Boundaries`. Each of the objects in this list has a path that includes `Boundaries`.



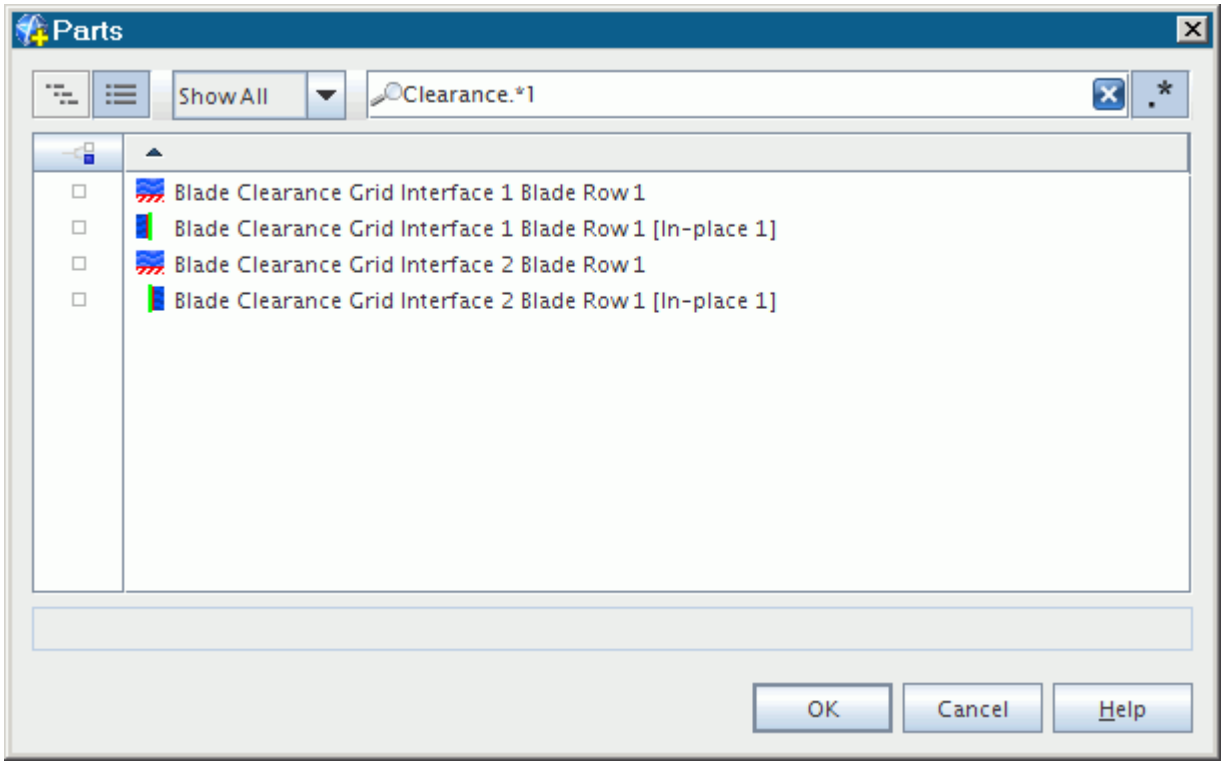
To refine the search results, type more text of the relevant path. For example, to view only the boundaries that have names starting with `Blade`, enter `Boundaries > Blade` in the path field.

Filtering Using Regular Expressions

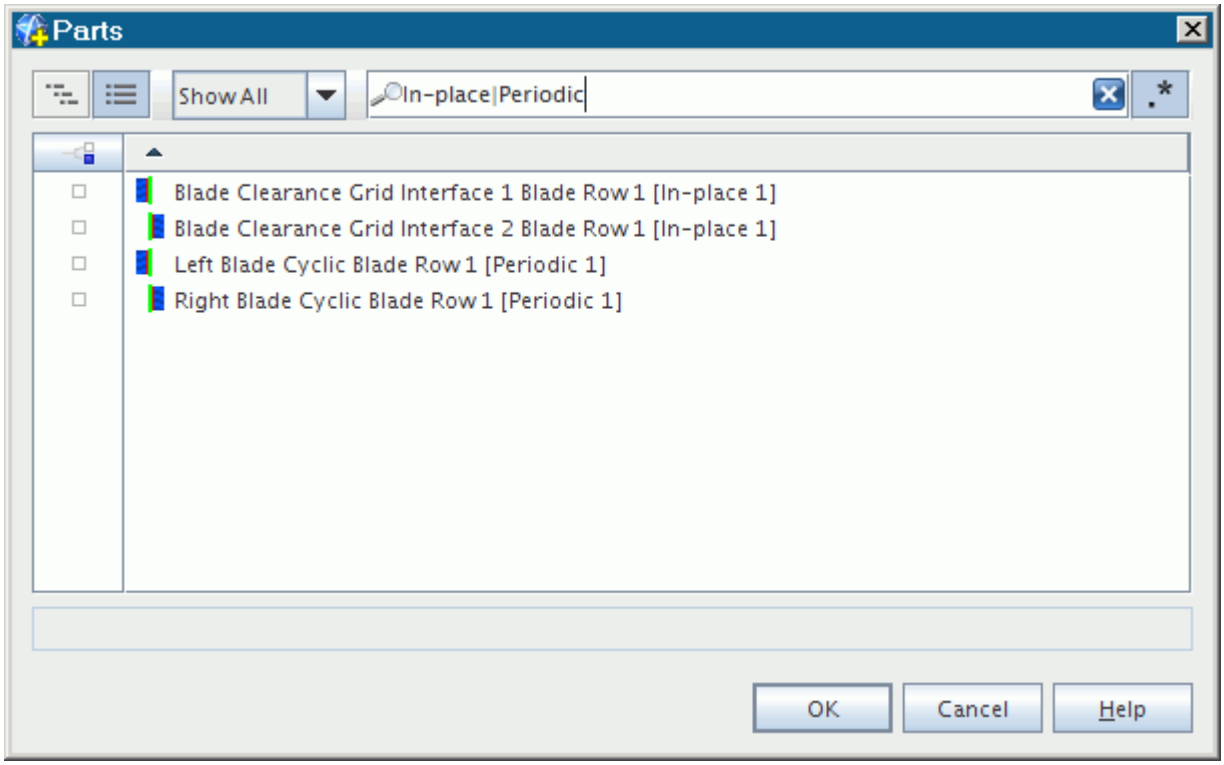
These brief instructions presume prior familiarity with the syntax of regular expressions. For more details about using this technique with the object selection dialog, see the section Regular Expressions.

- To enter regular expressions, click the **Use Regular Expressions** button to the right of the **Filter by Path** text box.

- To find objects with paths that include the same text strings, use the `.*` metacharacter. The example below shows items that have the word `Clearance`, followed by the `.*` portion (which means any number of characters, possibly no characters), followed by a `1`.



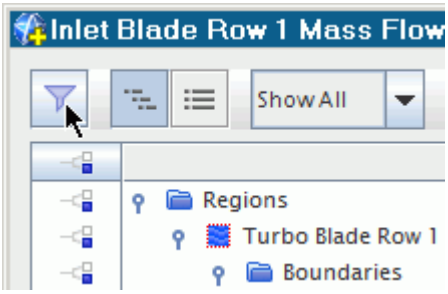
- To find objects using an OR relationship, use the `|` metacharacter. The example below contains the results for interface boundaries.



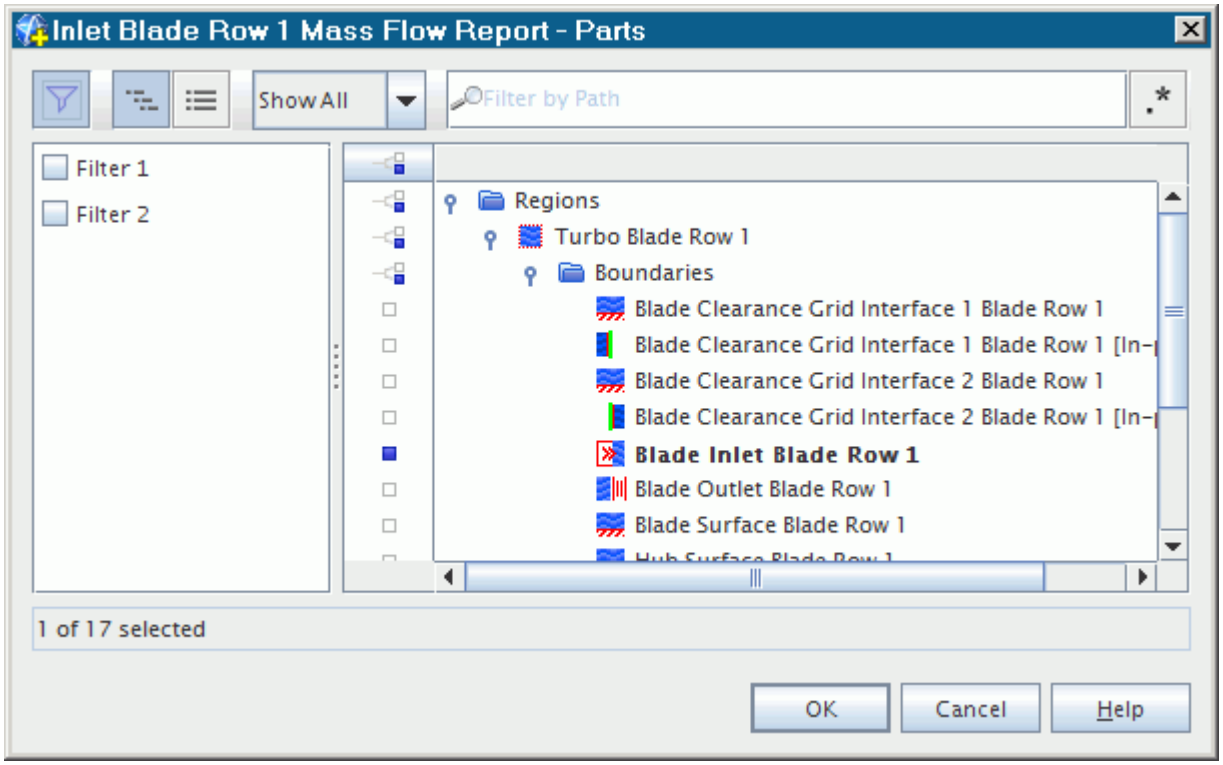
- To switch back to searching using plain text, click the **Use Regular Expressions** button.

Filtering Using Filter Objects

If you created filters in your simulation, the object selection dialog detects them and shows an additional button in the top left corner.



- To make these filters available in the dialog, click (**Show/Hide Filters**). A new panel with a list of available filters appears in the left portion of the dialog.



- To apply one of the available filters, activate the checkbox next to its name.

  The filters are applied in addition to any filters that you have already applied to the view.

You can also select a group of objects that are listed in the search results of a filter. See Selecting Nodes through a Filter.

Regular Expressions

In Regular Expressions mode, any text that is entered in the text box is interpreted as a regular expression.

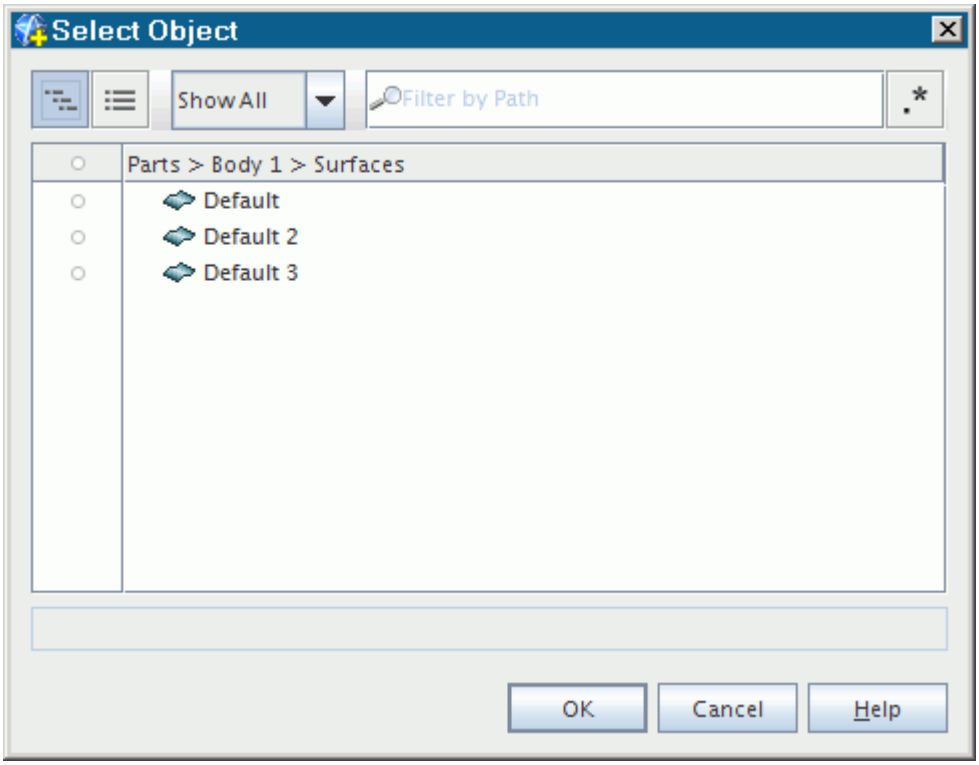
Regular expressions are case-sensitive by default. Objects are filtered on whether their paths, or paths of their parents, contain the regular expression. Regular expressions behave quite differently from simple text fragments, in that they follow a particular set of rules and conventions that must be learned. For example, the period, '.', in a regular expression represents any character, rather than just a period. In the context of a regular expression, the period can be called a metacharacter, as it has a programmatic function. For more information on using regular expressions, consult a suitable textbook. The full syntax permissible within the object selection dialog is detailed in the help page of the Java Pattern class. Some examples are given in the table below.

Example	Description
`mir.*rt`	Search for any object whose name contains text that starts with `mir` and ends with `rt`, and has any number of characters in the middle. The '*' metacharacter indicates zero or more times.

Example	Description
`bar|log`	The '|' metacharacter indicates an OR relationship. This search finds any object whose name contains either `bar` or `log`.
`[0-9]+`	The `[0-9]` represents a single digit in the range 0 to 9, and the '+' metacharacter indicates one or more times. This search finds any object that contains one or more digits in its name.
`[0-9]{2}`	The `{2}` after the digit range indicates two times. This search finds any object that contains at least two consecutive digits in its name. Any number can be used within the `{}` brackets.

Single-Selection Mode

The object selection dialog appears in a special single-selection mode for STAR-CCM+ features that are not designed to have multiple objects selected at one time.



Single-selection mode works the same as the typical multiple-selection mode, with the following exceptions:

- You can select only one object in the panel at a time.
- Clicking a selection button toggles between selection and deselection of an object.

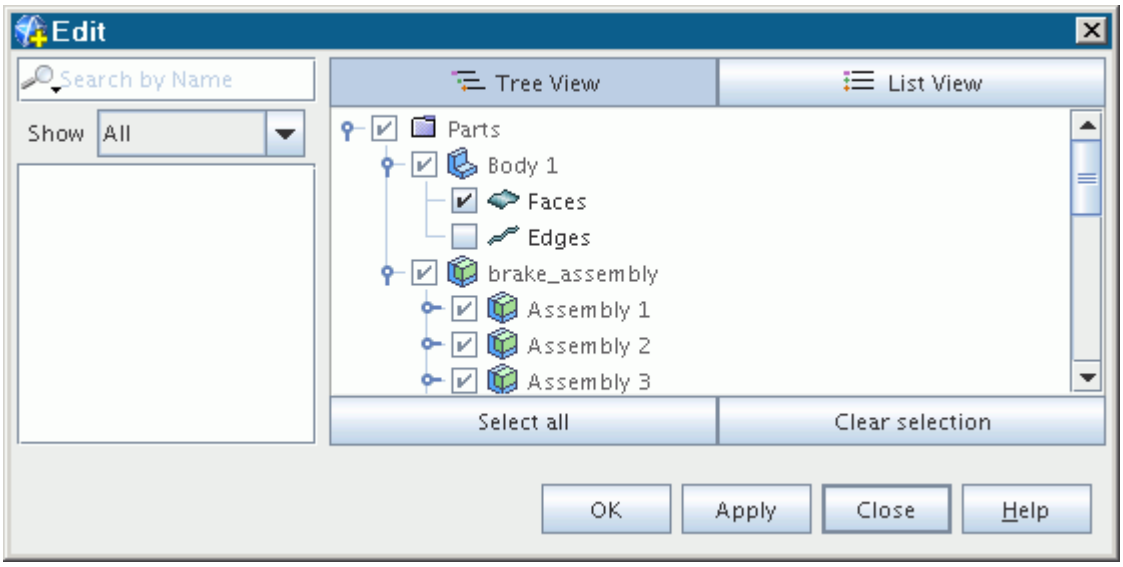
- If you select an object when another object in the panel is already selected, that other object becomes deselected automatically.
- The selection button will only be active for objects that are available for selection. Therefore the button is not active for manager objects, which cannot be selected.
- The header of the selection column is not a button but an indicator. Its icon appears colored when any object is selected, and clear when there is no selection.

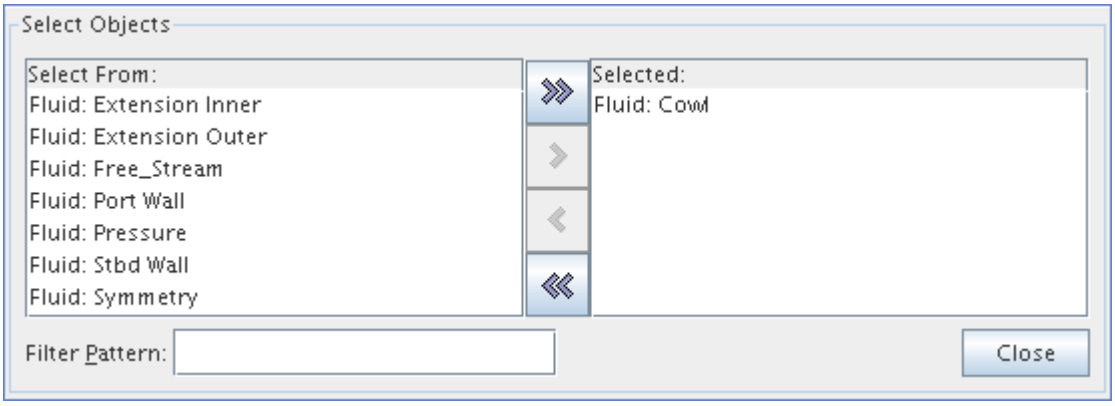
Examples of features using single-selection mode include:

- Defining a battery module configuration
- Selecting parts for an external battery casing
- Magnetic fields -- controls for:
  - The excitation coil reference cross section
  - The local direction field
- Defining topology -- controls for:
  - Assigning parts to regions
  - Assigning part surfaces to boundaries
  - Assigning part curves to feature curves
- Surface repair -- controls for:
  - Editing part surfaces
  - Editing part curves

Alternate Formats

Other formats of the object selection dialog, commonly used in earlier versions of STAR-CCM+, appear as follows:





In the inline format immediately above, the buttons between the two lists function as follows:

- Transfers the part that has the focus from the Select From list to the Selected list.

- Transfers the part that has the focus from the Selected list to the Select From list.

- Transfers all parts in the Select From list to the Selected list.

- Transfers all parts in the Selected list to the Select From list.

You can select multiple parts at the same time by holding down the <`Ctrl`> key while selecting them.

Comparing Objects and Simulation Trees

STAR-CCM+ provides a comparison tool that allows you to identify differences visually between individual objects or between simulations (in the same STAR-CCM+ session). You can observe differences in object properties such as model settings, boundary conditions, and meshing parameters.

This tool appears as a separate window. When you select a highlighted child object, the property values are shown in a pair of *Properties* windows below the *Explorer* windows. You can edit property values within these windows.

Comparing Objects

To compare exactly two nodes in the same simulation, do the following:

1. Multi-select the nodes.
2. Right-click one of the highlighted nodes and select **Compare**.
3. View the nodes side-by-side in the *Comparison* window.

Alternatively, use the following technique which lets you compare objects within a simulation or between simulations:

1. Select **File** > **Open Empty Comparison View**.
2. Drag any pair of nodes from the simulation tree into the *Comparison* window.

   Note: If a comparison window already has a node, the node that you drag replaces it.

3. To compare objects between simulations, drag one node at a time from each simulation tree into a different half of the *Comparison* window.

Comparing Simulations

When comparing simulations, the comparison tool duplicates the object trees of the two simulations that you want to compare. In these duplicate object trees, highlighted root objects guide you to the child objects that are different between simulations.

A typical workflow for comparing simulations is as follows:

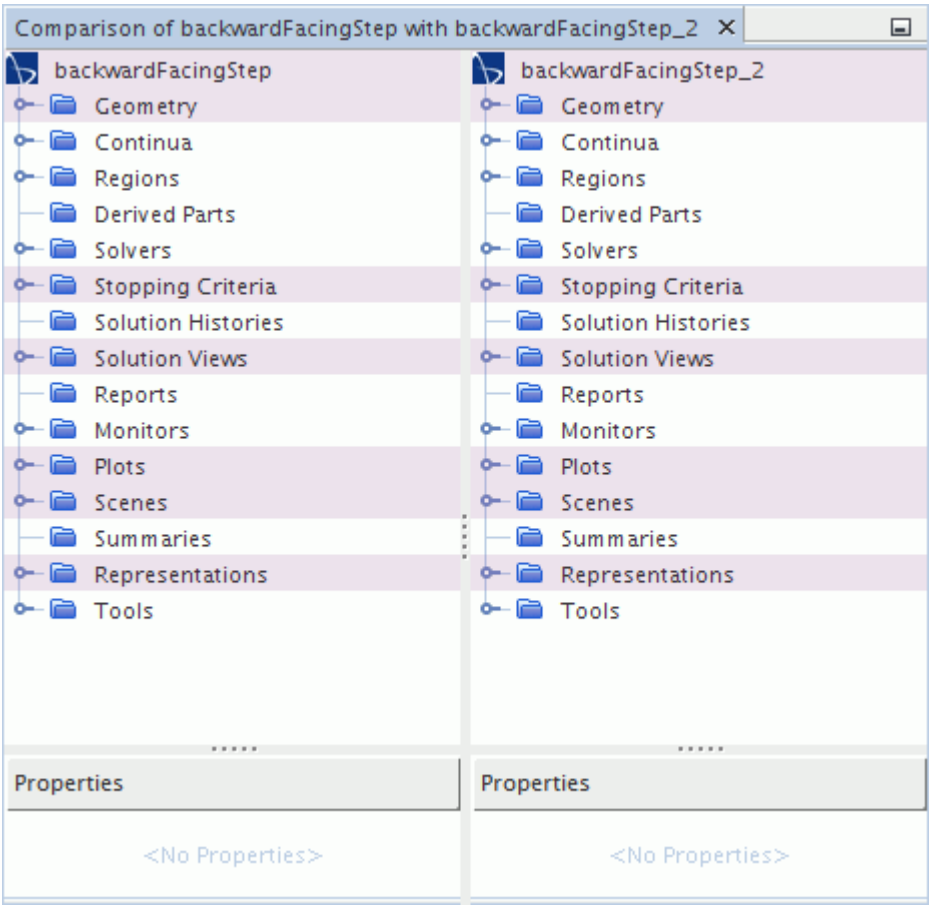
1. In the same session of STAR-CCM+, open the two simulation files that you want to compare.

   Remember that STAR-CCM+ checks out one license per simulation.

2. With either simulation window active, access the comparison in one of the following ways:
   - Select **File** > **Compare with [simulationName]**.
   - Right-click the root node of the simulation tree and select **Compare with [simulationName]**.

   The appearance of the menu varies with the number of open simulations. If you have more than two simulations open, the simulation names are available for selection in a submenu of the **Compare with** menu item. However, if you have only one simulation open, this menu item is inactive (grayed out).

   After you execute this menu action, the comparison window appears in the workspace.



3. Navigate through the object trees. To find differences consecutively, use the following techniques:
   - Next difference: right-click on a node and choose **Select Next Difference**, or press `<Shift>+<Alt>+<Down>` arrow.
   - Previous difference: right-click on a node and choose **Select Previous Difference**, or press `<Shift>+<Alt>+<Up>` arrow.

   Observe the following conventions for node highlighting:

   - No highlighting: this node, and its children (if any), are identical across simulations
   - Light purple: one or more children of this node differ between simulations
   - Dark purple: this node or property is different between simulations
   - Green: this node or property appears only in one simulation and not the other
4. Compare and change properties of nodes in the comparison windows.
5. To locate a node in the main simulation tree, right-click it in the comparison tree and choose **Select in Simulation**.

Note: You can only have one comparison window open at a time. Suppose that you have three or more simulations open, and you launch the **Compare with** action with the third simulation. The existing comparison window closes and a new one appears with your latest comparison choices.

Using Custom Simulation Trees

If you have a large simulation with many objects in the tree, add custom trees for quick access to the objects that you use most.

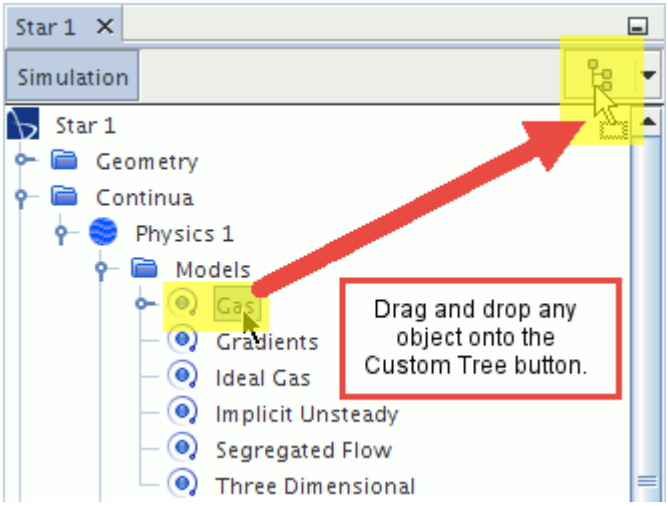
For complex simulations that have numerous regions, reports, scenes, monitors, injectors, and field functions, browsing in the simulation tree can be time-consuming. To change settings requires a lot of scrolling and clicking.

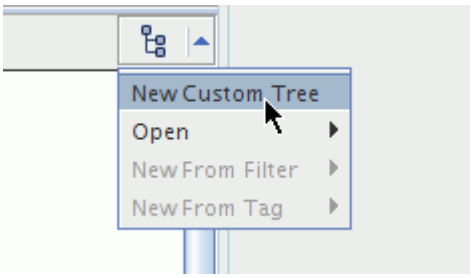
To reduce the complexity of the tree, use custom simulation trees that contain only the nodes that you choose. Examples of uses are:

- With only the physics of relevance to a given problem, complete the physics continua specification on the reduced tree.
- With only the meshing nodes, create a custom mesh with minimum mouse moves.
- View only those nodes that have been modified from default property values.

A custom tree can consist of the objects that are associated with a filter or a tag. For details about filters, see Defining Logical Groups of Objects. For details about tags, see Tagging Simulation Objects.

To use custom trees, set them up and manage them with the following techniques:

Action	Instructions
Create an empty custom tree.	1. In an open simulation, create the custom tree object in one of the following ways: • In the explorer window, drag any object from the simulation tree and drop it on the (**Create a Custom Tree**) button.  • Click (**Create a Custom Tree**).

Action	Instructions
	• Click the drop-down arrow on the **Create a Custom Tree** button and select **New Custom Tree**.  A new *Custom Tree* tab appears next to the explorer window, and a new **Custom Tree** node appears in the **Tools** > **Custom Trees** node. 2. To label the contents of your custom tree, rename the **Custom Tree** node with a label such as `Physics` or `Post`. 3. To populate the custom tree, drag the nodes from the *Simulation* tab and drop them in the window of the tree.
Create a custom tree from a filter.	To create a custom tree from a filter, do either of the following: • Click the drop-down arrow on the **Create a Custom Tree** button and select one or more filters from the **New From Filters** submenu. • In the **Tools** > **Filters** node, right-click one or more filter nodes and select **New Custom Tree**. If you create a custom tree from multiple filters, the resulting custom tree contains objects that are included in any of those filters. For example, suppose the geometry part **Block** is included in **Filter 1** and the geometry part **Cylinder** is included in **Filter 2**. Creating a custom tree from **Filter 1** and **Filter 2** results in a custom tree that contains **Block** and **Cylinder**. **Note:** These techniques are available only if your simulation contains at least one filter. **Note:** If you modify a filter after creating a custom tree from it, the custom tree does not update itself automatically. For an updated custom tree, create another one from the modified filter.
Create a custom tree from a tag.	To create a custom tree from a tag, do either of the following: • Click the drop-down arrow on the **Create a Custom Tree** button and select one or more items from the **New From Tags** submenu. • In the **Tools** > **Tags** node, right-click one or more tag nodes and select **New Custom Tree**. If you create a custom tree from multiple tags, the resulting custom tree contains objects that are tagged with any of those tags. For example, suppose the geometry part **Block** is tagged with **Tag 1** and the geometry part **Cylinder** is

Action	Instructions
	tagged with **Tag 2**. Creating a custom tree from **Tag 1** and **Tag 2** results in a custom tree that contains **Block** and **Cylinder**. **Note:** These techniques are available only if your simulation contains at least one tag.
Open an existing custom tree window.	To open an existing custom tree window, do either of the following: • Click the drop-down arrow on the **Create a Custom Tree** button and select an item from the **Open** submenu. • Double-click the node of that custom tree in the **Tools** > **Custom Trees** node. As you hover with the mouse over the nodes, the node hierarchy appears in the path field at the lower left of the window.
Control the view of the objects in the custom tree.	To control the view of the objects in the custom tree, use the button at the bottom of the custom tree window: • To show the position of the nodes in the main simulation tree, click (**Show Full Tree Hierarchy**). With this setting activated, other nodes that you add to the custom tree appear with the full hierarchy of parent nodes, just like in the main simulation tree. However, nodes in the hierarchy that have been explicitly added to the custom tree have labels appearing in bold text, while the other labels appear in normal text. In hierarchy mode, the added nodes appear in the same order as in the main simulation tree. • To restore the view so that only the placed nodes are visible, click (**Show Full Tree Hierarchy**) again. **Note:** This action also closes the placed nodes, so that any sub-node hierarchies that you were displaying are no longer visible.
Find a node in the main simulation tree.	To locate a node in the main simulation tree, right-click it in the custom tree and choose **Select in Simulation**.
Remove a node from the custom tree.	To remove a node from the custom tree, right-click it and select **Remove from this Tree**.

Custom tree objects can also be copied, renamed, or deleted using either of the following techniques:

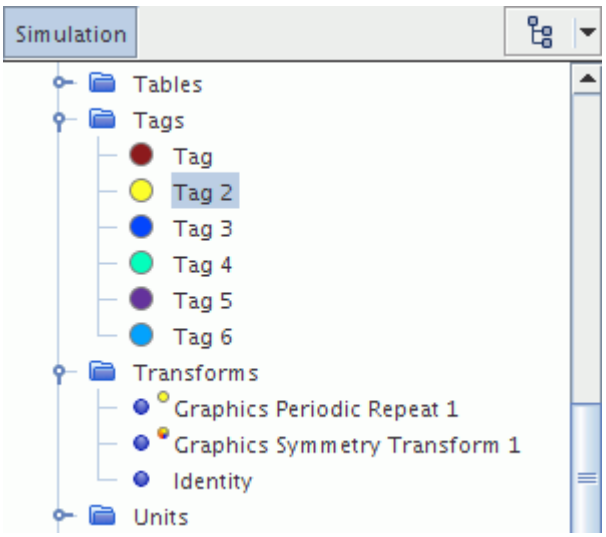
- Right-click the custom tree node in the simulation tree.
- Right-click the background of the custom tree.

Tagging Simulation Objects

Tags are like graphical labels that you apply to objects in the simulation tree. You can then reference objects by tag when using filters.

Once a tag has been created, it can be applied to one or more objects to distinguish them from other objects. A filter can be created based on the tag. The filter can then be used to limit the objects that are available for selection in various dialogs and tools within STAR-CCM+.

A colored circle, with a unique color assigned automatically by STAR-CCM+, represents the tag. An example of a tag that is applied to two different transforms is shown in the following screenshot:



Two of the transforms in this image have tags applied to them, as indicated by the small circles that accompany their nodes. One of the transforms has multiple tags assigned to it, therefore its circle tag has a blend of colors.

Tags can be added to most objects in STAR-CCM+.

Contents:

Using Tags

Using Tags

Any number of user tags can be created for flagging STAR-CCM+ objects.

1. To create a user tag:
   a) Right-click the **Tools** > **Tags** manager node and select **New Tag** from the pop-up menu.

      A new node named **Tag** appears.

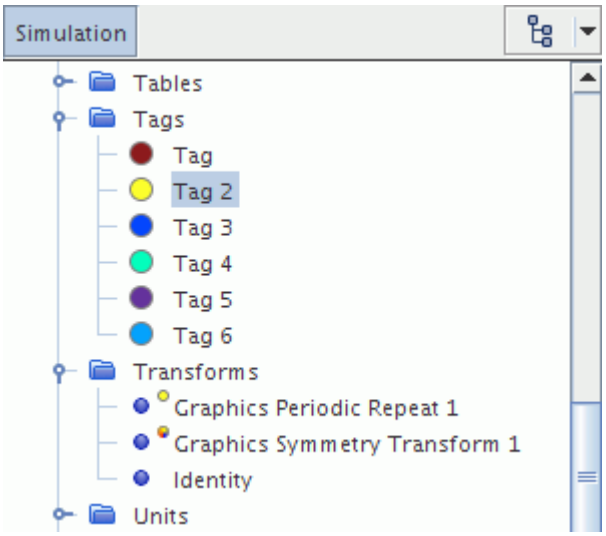
Note: To create a tag and apply it immediately to a particular object, right-click the node of that object and select **Set Tags** > **New...**. A user tag is automatically created and applied to the object at the same time.

b) To give the tag a descriptive and unique name, rename it before you apply it.

2. To apply a tag to an object, use one of the following techniques:
   - To apply the tag through the properties of the object node, edit its *Tags* property which activates the object selector (see Activating the Object Selection Dialog).
   - Drag the tag node onto the object.
   - To apply the tag through the context menu of the object node, right-click the object node and select **Set Tags** and then the name of the tag (**Tag 2** for example).

     For a larger set of tags, the list of tag names is replaced with an **Existing** submenu item.

     A miniature version of the tag icon is placed next to the object node indicating that it has been tagged, as shown (for transforms) in the following screenshot:



3. To apply a new tag to filter search results, use one of the following techniques, depending on whether you want the full set of search results or a subset:
   - To create and apply a tag to the full set of search results of a filter, right-click the filter node and select **Tag Result**.
   - To create and apply a tag to selected search results, in the open dialog of a filter, do the following:
     1. Select the objects in the search results window of the dialog.
     2. Right-click in that group and choose **Select in Simulation**.
     3. In the main simulation window, right-click and select **Set Tags**.
4. To search for a set of objects having specific tags:
   a) Create a filter. See Creating a Filter.
   b) Specify a tag predicate, `[Tag contains MyObjects]`, where **MyObjects** is the name of the tag.
   c) To search for additional tags in the same query, use a compound predicate.
5. To remove a tag from an object, do one of the following:

- Edit the *Tags* property of the object.
- Right-click the object node, select **Set Tags**, and deselect the name of the tag in the submenu.

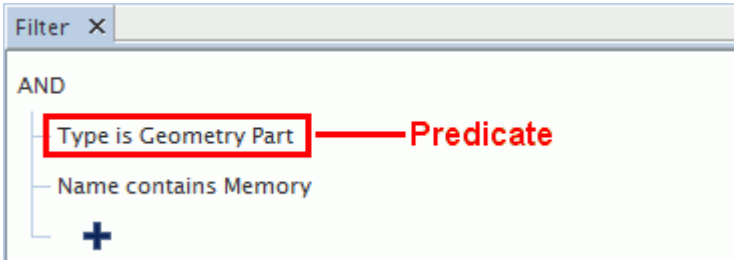
6. To change the color of a tag (both the node and the miniature icon on the objects with that tag), select the node of that tag and do one of the following:
   - Click in the right half of the *Color* property and select a new color from the drop-down swatch.
   - To access the color control dialog, click ... (**Custom Editor**) in the *Color* property. See Defining Colors.
7. To delete a tag node, right-click the tag node and select **Delete Tag...** from the context menu.

   All objects that are associated with the tag are also updated so that the link is removed from each.

Object Filters

In the STAR-CCM+ simulation tree, a filter is an object that uses queries to define groups of STAR-CCM+ objects.

A query consists of one or more conditional statements, called *predicates*, by which STAR-CCM+ searches among the objects in the simulation. The example below shows a Type predicate.

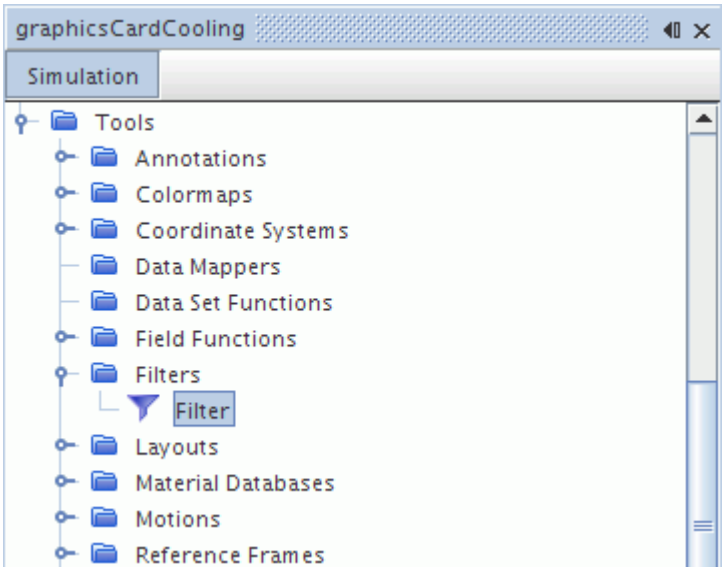


A query is executed and returns results.

Some uses of filters include:

- finding objects in the tree
- query-based dynamic selection in mesh operations, derived parts, and scene displayers
- creating custom trees
- constructing nested queries
- limiting the set of selectable objects in an object selector

Filters appear for selection in the **Tools** > **Filters** node of the STAR-CCM+ object tree.



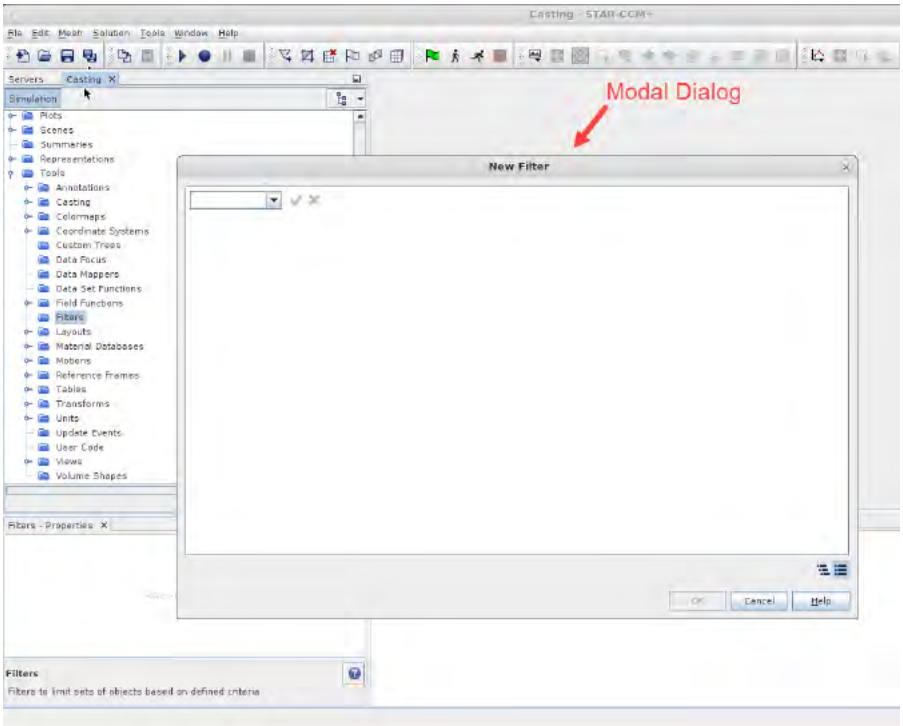
In addition to the **Tools** > **Filters** node, filters are accessible through the following parts of the STAR-CCM+ interface:

- The object selection dialog; for details, see Filtering Using Filter Objects.
- The *Properties* window (for dynamic queries); for details, see Activating a Dynamic Query.

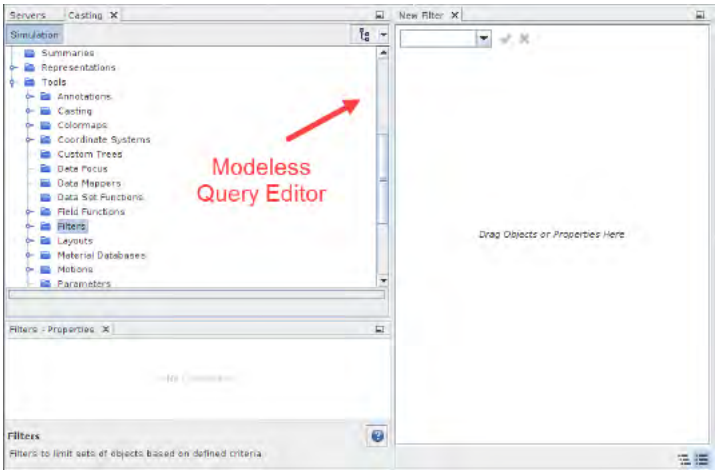
The Query Editor

In 11.06, the query editor has undergone a significant change. Although it is still launched from **Tools** > **Filters**, it is no longer a modal dialog.

In other words, in versions prior to 11.06, the query editor was a separate dialog that took the focus and did not allow interaction with the main STAR-CCM+ workspace.



The 11.06 query editor exists within the main window in a new panel.



The main advantage of this arrangement is that items from the object tree can be dragged and dropped into the query editor.

Contents:

Predicates
Property Predicates
Nested Queries
Relationship Predicates
Working with Filters
Filter Dialog Reference
Predicate Reference

Predicates

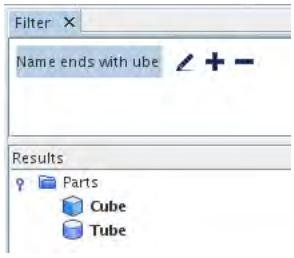
A predicate is a conditional statement that is applied to a set of objects. An object either satisfies the condition or it does not; if it does, it is included in the results.

The query editor allows you to create and modify predicates through drop-down menus and object selectors. You can also drag and drop objects or properties from the simulation tree into the query editor.

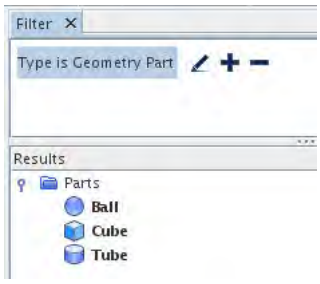
Although there are a few small variations, a predicate generally consists of a predicate type, a predicate operator, and a predicate value. For example, in the predicate, `[Name Contains MyPart]`, the type is `Name`, the operator is `Contains`, and the value is `MyPart`.

Consider an example. Suppose that a predicate is applied to three parts: a cylinder (**Tube**), a block part (**Cube**), and a sphere (**Ball**). Some different predicates applied to this set of parts are as follows:

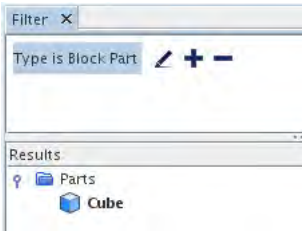
1. `[Name ends with ube]` returns both the cylinder (**Tube**) and the block (**Cube**)



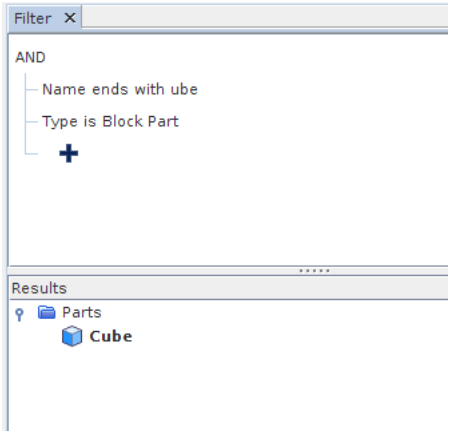
2. `[Type is Geometry Part]` has results consisting of all three parts



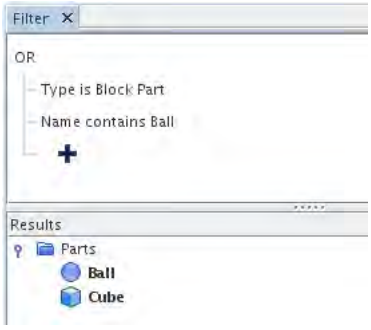
3. `[Type is Block Part]` has results consisting only of the block (**Cube**)



Predicates can be combined using the 'AND' and 'OR' operators. When two predicates are combined using AND, the filter displays the intersection of their results. For instance, by combining Query 1 with Query 3 using AND, you receive the result, **Cube**.



When two predicates are combined with OR, the filter displays the union of their results. For instance, combining `[Type is Block Part]` with `[Name contains Ball]` returns both the cube and the sphere.



When creating or editing filters, all changes are automatically committed to the server. The Discard Changes (X) button restores the predicate as it existed before any edits. Undo is available using the right-click menu or CTRL+<z>.

The 'Pre-11.06 Results' Predicate

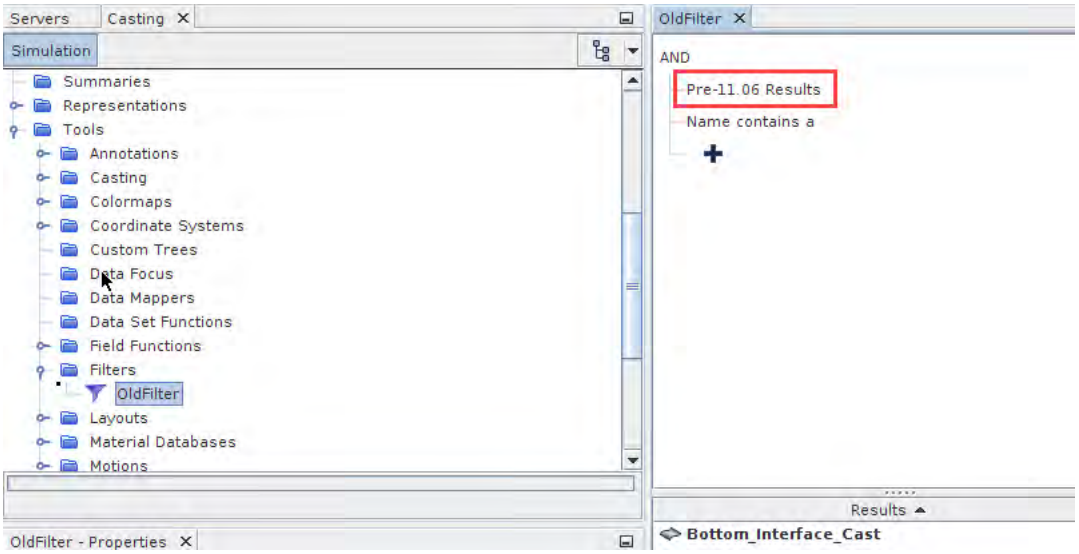
In STAR-CCM+, a named object is an object that can be renamed. To rename, you select, for example, a part, press <F2> (or right-click and choose **Rename...**), and then give the object a new name. Objects such as parts, part surfaces, part curves, regions, continua, reports, scenes, and plots are all named objects.

Objects such as meshers, default controls, models, physics conditions, physics values, and initial conditions cannot be renamed. These objects are termed *non-named*.

Generally, you can consider named objects as physical objects that you incorporate into your simulation, while non-named objects represent the attributes of those physical objects.

Prior to STAR-CCM+ v11.06, queries returned only named objects. Since v11.06, queries return both named objects and non-named objects. This change in behavior became necessary to make some of the relationship and property-based nested queries work properly.

As this change in behavior could cause backwards compatibility problems for some cases, you can include the predicate, `Pre-11.06 Results`, that restricts queries to return only named objects. Queries that contain `Pre-11.06 Results` behave in the same manner as queries made in versions of STAR-CCM+ prior to v11.06.



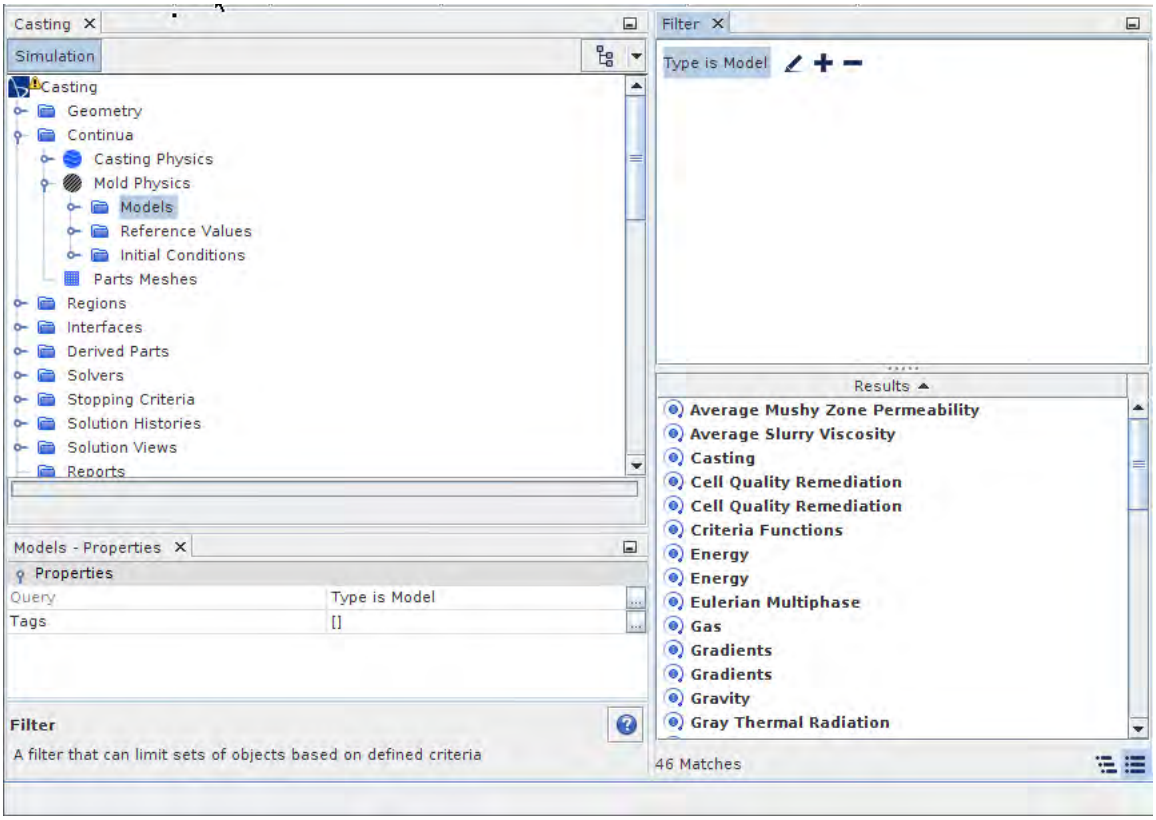
Hence, when restoring a query written before v11.06, STAR-CCM+ automatically adds the `Pre-11.06 Results` predicate with an AND operator.

Returning Only Named Objects (without Using `Pre-11.06 Results`)

If you want to restrict results to return only named objects, you can add the predicate `[Name contains ]`. This predicate has the effect of limiting the results to only named objects—the `Pre-11.06 Results` predicate is essentially `[Name contains ]`.

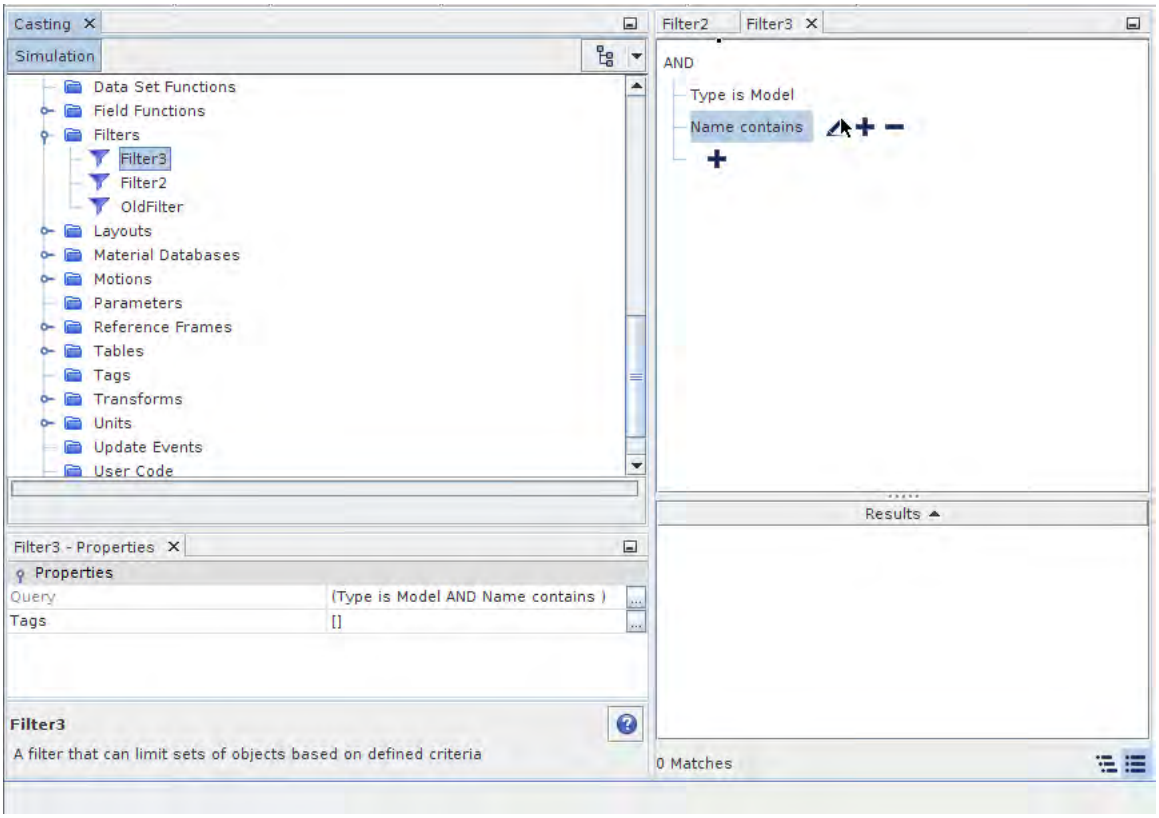
To illustrate the difference between named and non-named objects, consider the following example in which the **Continua** > **Mold Physics** > **Models** node is dragged into the query editor to create `[Type is Model]`. All the models in the continuum appear as results:

`[Type is Model]` returns models.



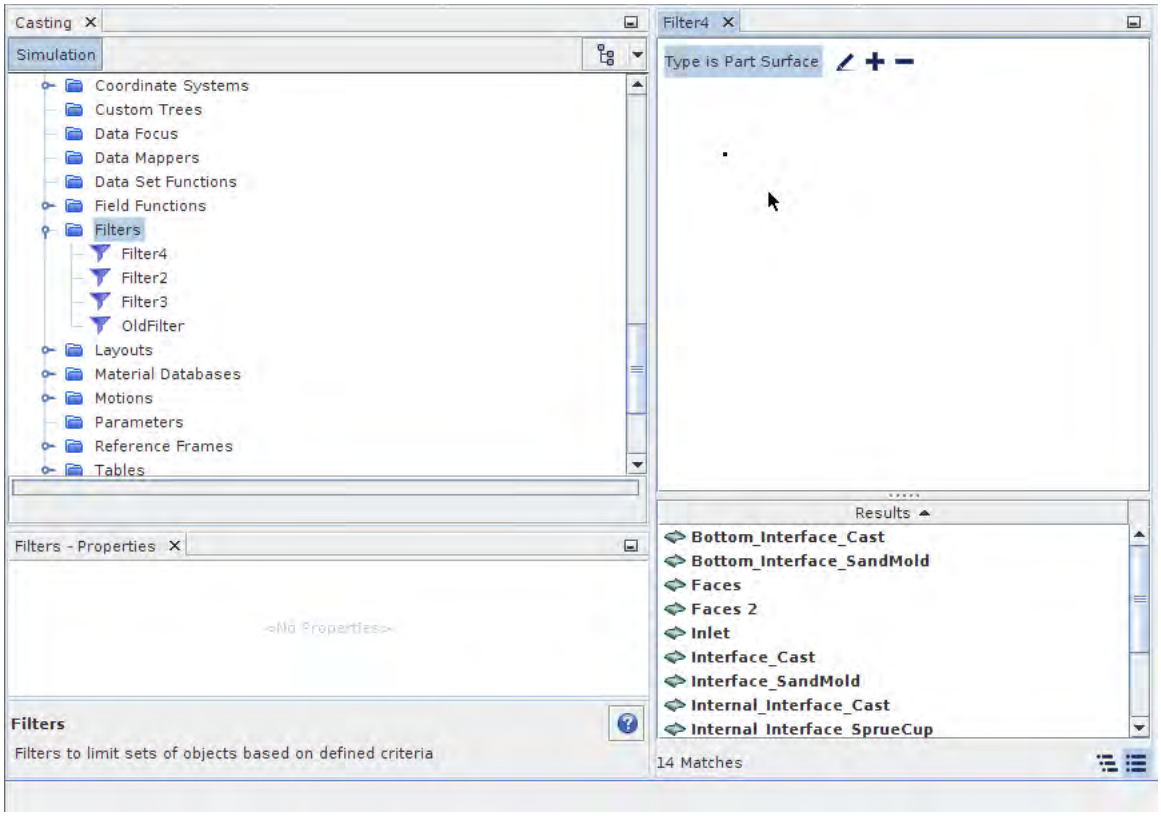
Now, when the predicate `[Name contains ]` is added using AND, all the results disappear. As the names of model objects are fixed by STAR-CCM+ and cannot be modified, the `[Name contains ]` predicate ignores them:

`[Type is Model] AND [Name contains ]` returns nothing.

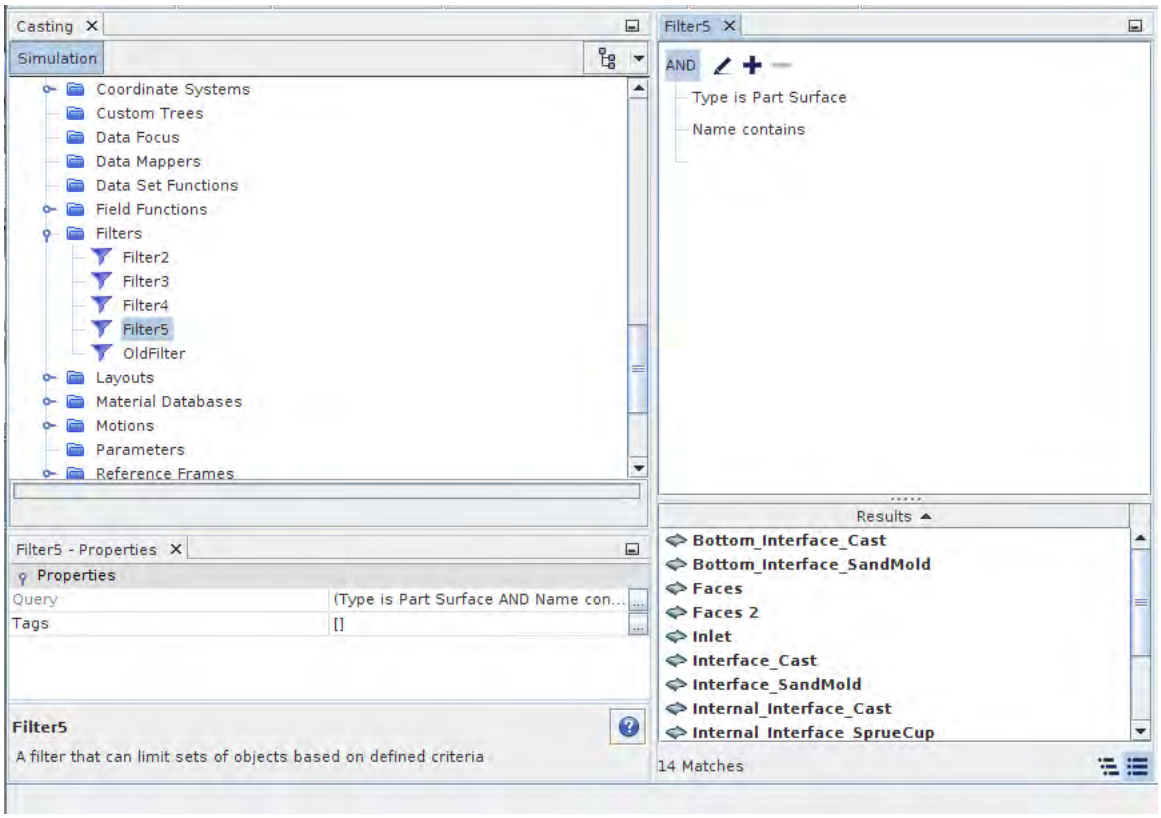


Two further examples illustrate the behavior of named objects in a query:

- `[Type is Part Surface]` returns all part surfaces. Part surfaces depend on the geometry of the model and are objects that you create when building a simulation.



- `[Type is Part Surface] AND [Name contains ]` returns the identical results, because part surfaces are also named objects.



Property Predicates

Many important objects in the simulation tree have properties that can be used to generate useful queries.

For instance, you can use property predicates to identify all velocity inlet boundaries or all fluid regions.

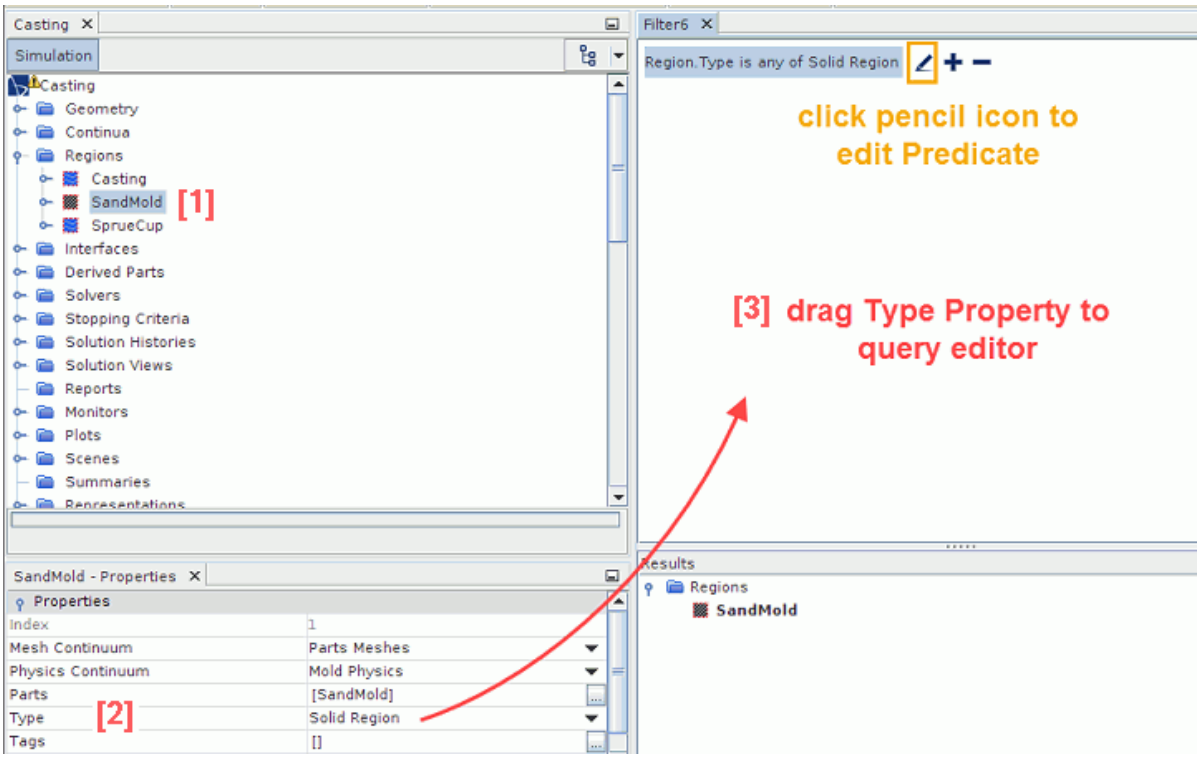
To create a property predicate, select the property in the *Properties* window and drag it into the query editor.

The search results from a property predicate query contain the same type of object as the one to which the property belongs (the *owning object*).

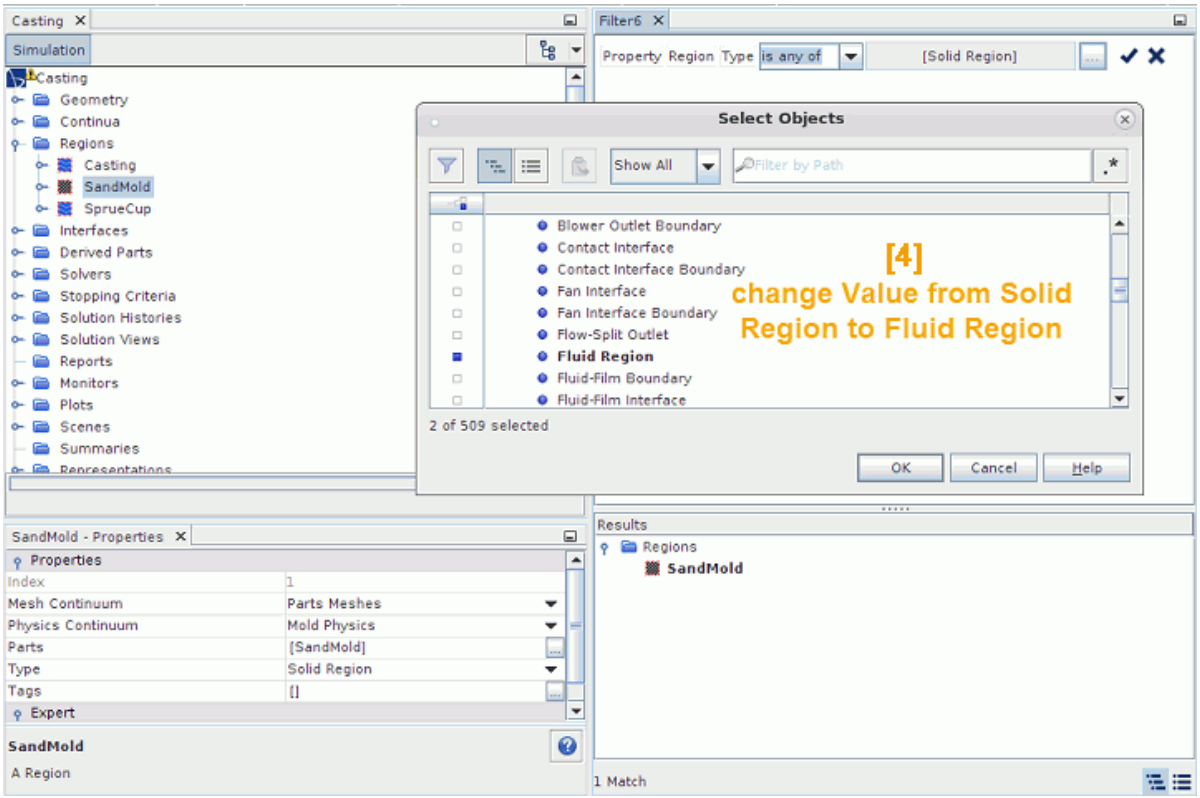
The following examples illustrate how you can use this feature:

- To find all regions of type Fluid:
  1. Select a region.
  2. Find the *Type* property.
  3. Drag the property into the query editor.

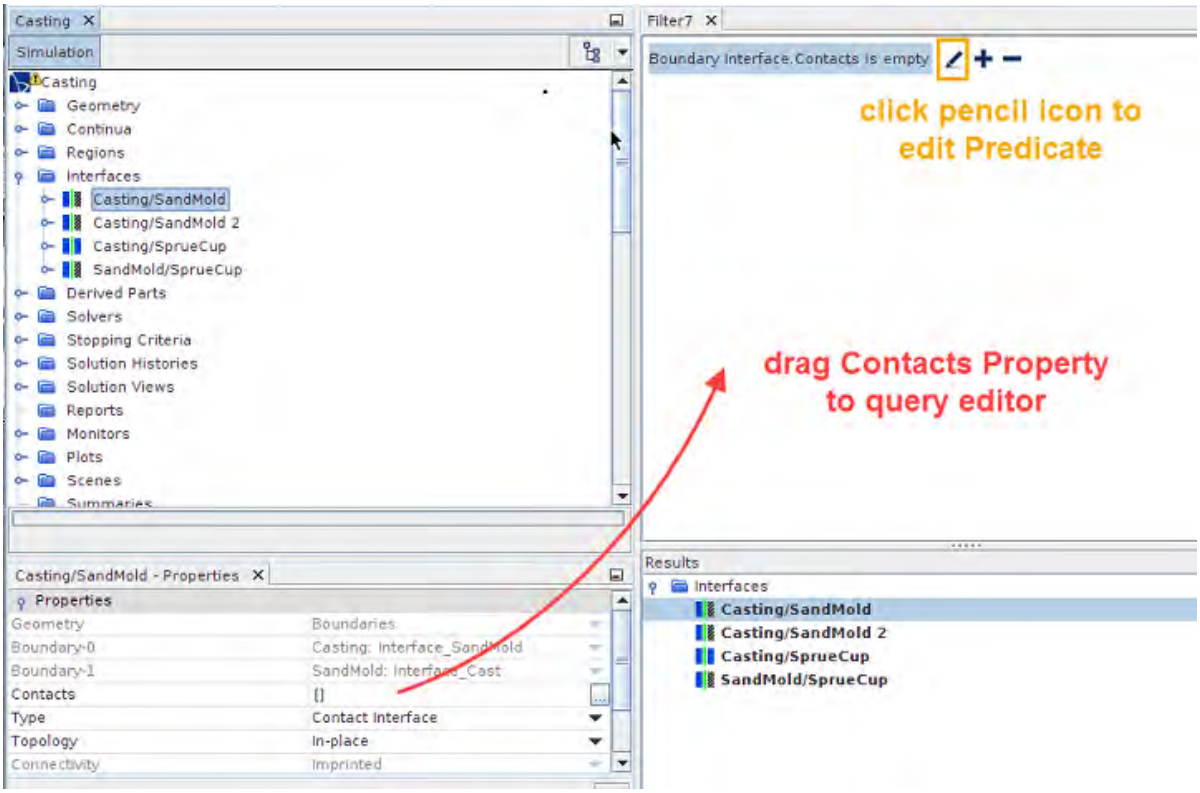
     If *Type* is set to **Fluid Region**, the query provides the desired search results. Otherwise, you can edit the predicate.



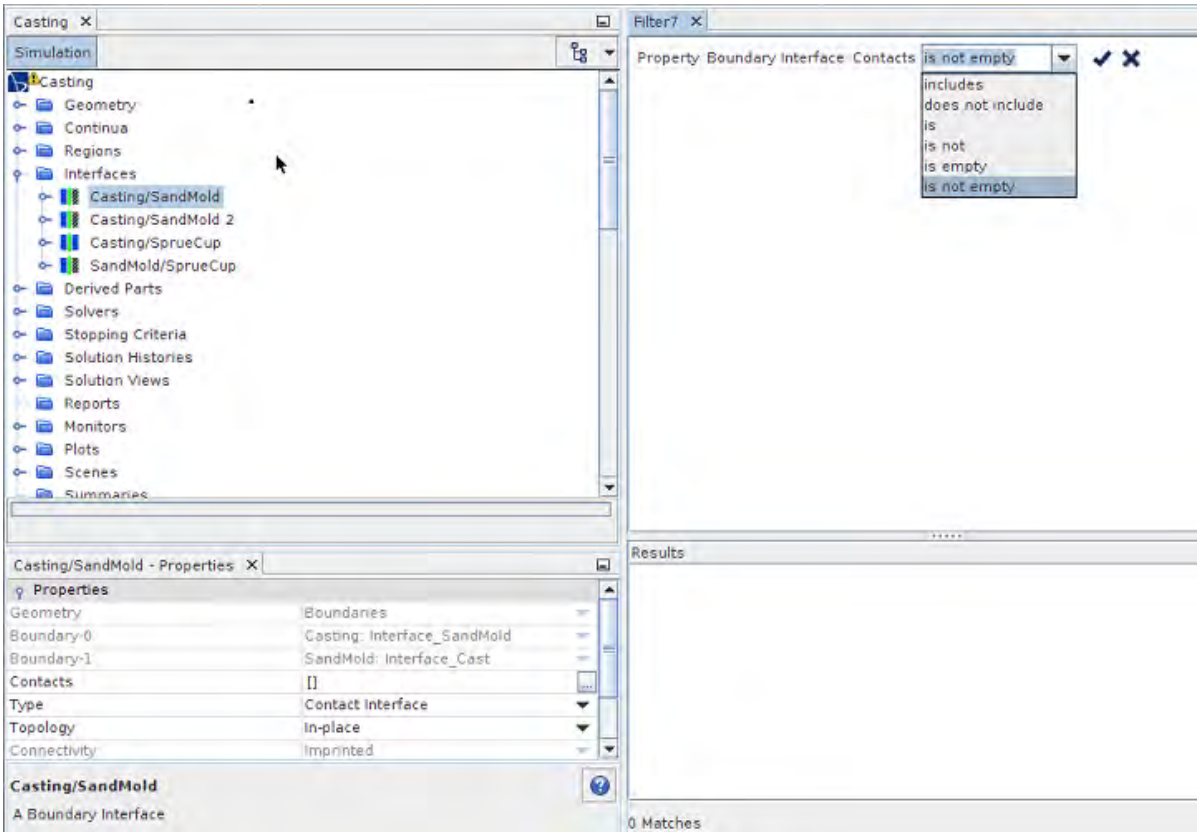
  4. In the edit mode of the predicate, set the *Type* property as desired.



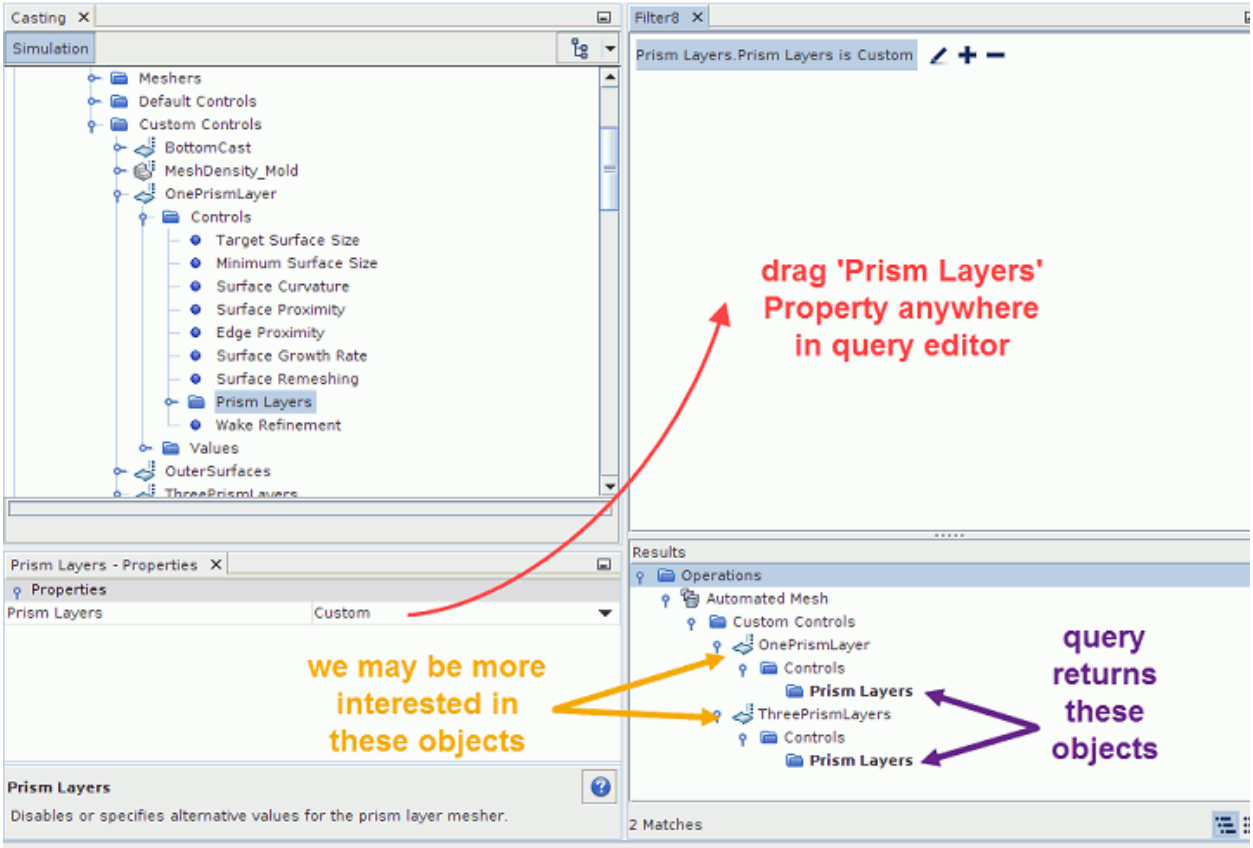
- To find interfaces with contacts, and then the ones without them:
  1. Drag the *Contacts* property into the query editor to obtain `[Boundary Interface.Contacts is empty]`.



2. Change the predicate to `[Boundary Interface.Contacts is not empty]`.



- To find all the regions that have not been assigned parts, drag the *Parts* property of a region into the query editor to get `[Region.Parts is empty]`.
- To find all prism layers that are set to **Custom**, drag the *Prism Layers* property of the **Prism Layers** node under a custom mesh control.



The results are unnamed objects that have that property. However, you may be more interested in the custom controls themselves. For details on how to modify the query to return those objects, see Example: Finding Custom Controls with Prism Layers of Type Parent.

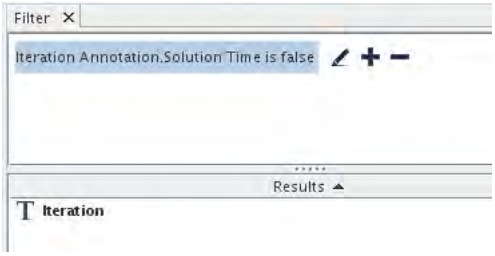
How Common Properties Affect Predicates

In STAR-CCM+, shared properties among objects usually reflect the fact that those objects belong to a certain class.

For example, the annotation types iteration, logo, and simple text have different sets of properties, but they also share the *Opacity* property. This sharing reflects the common class of the objects as *annotations*. Therefore, when you drag and drop the property of an annotation into the query editor, the appearance of the predicate depends on which property you choose.

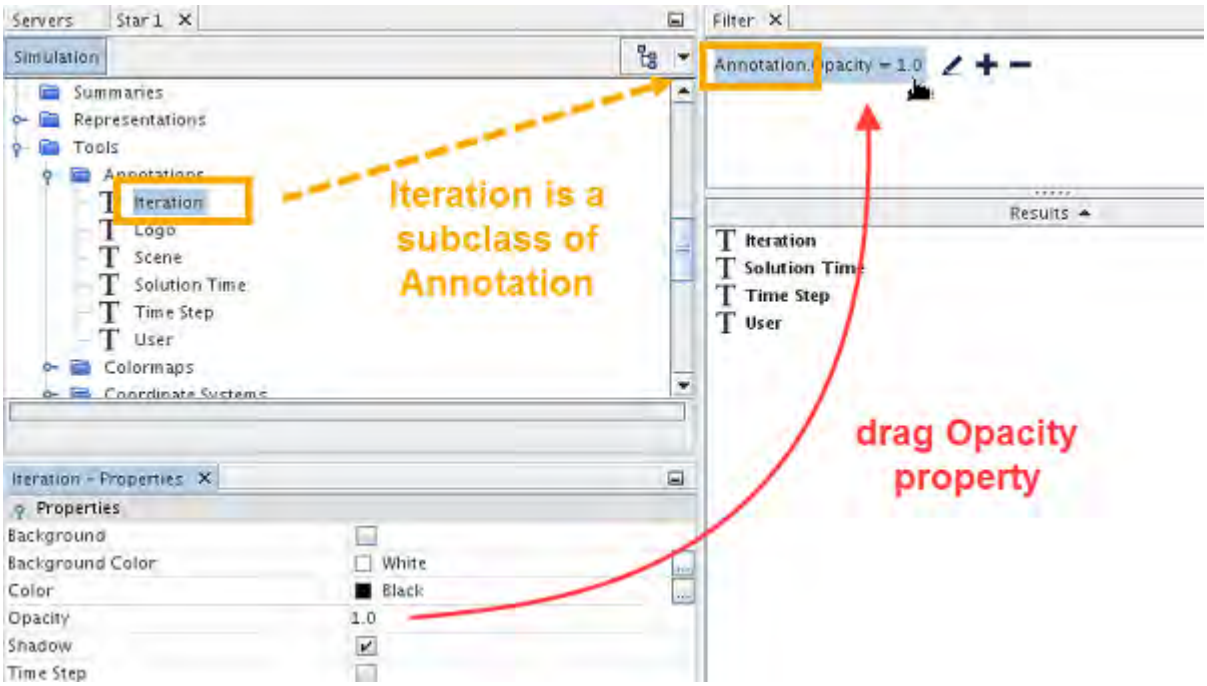
For example, when you drag and drop the *Solution Time* property of the iteration annotation, you get the following predicate:

```
[Iteration Annotation.Solution Time is false]
```



However, when you drag and drop the *Opacity* property of the same type of annotation, you get a different predicate:

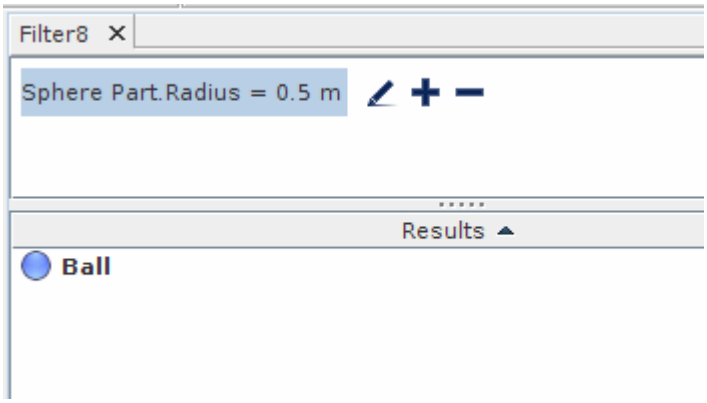
```
[Annotation.Opacity = 1.0]
```



As another example, consider cylinder, cone, and sphere parts, which belong to the Shape Part class. Each class of objects can in turn be a subclass of other classes. For example, the Shape Part class belongs to the Mesh Part class, which belongs to the Geometry Part class.

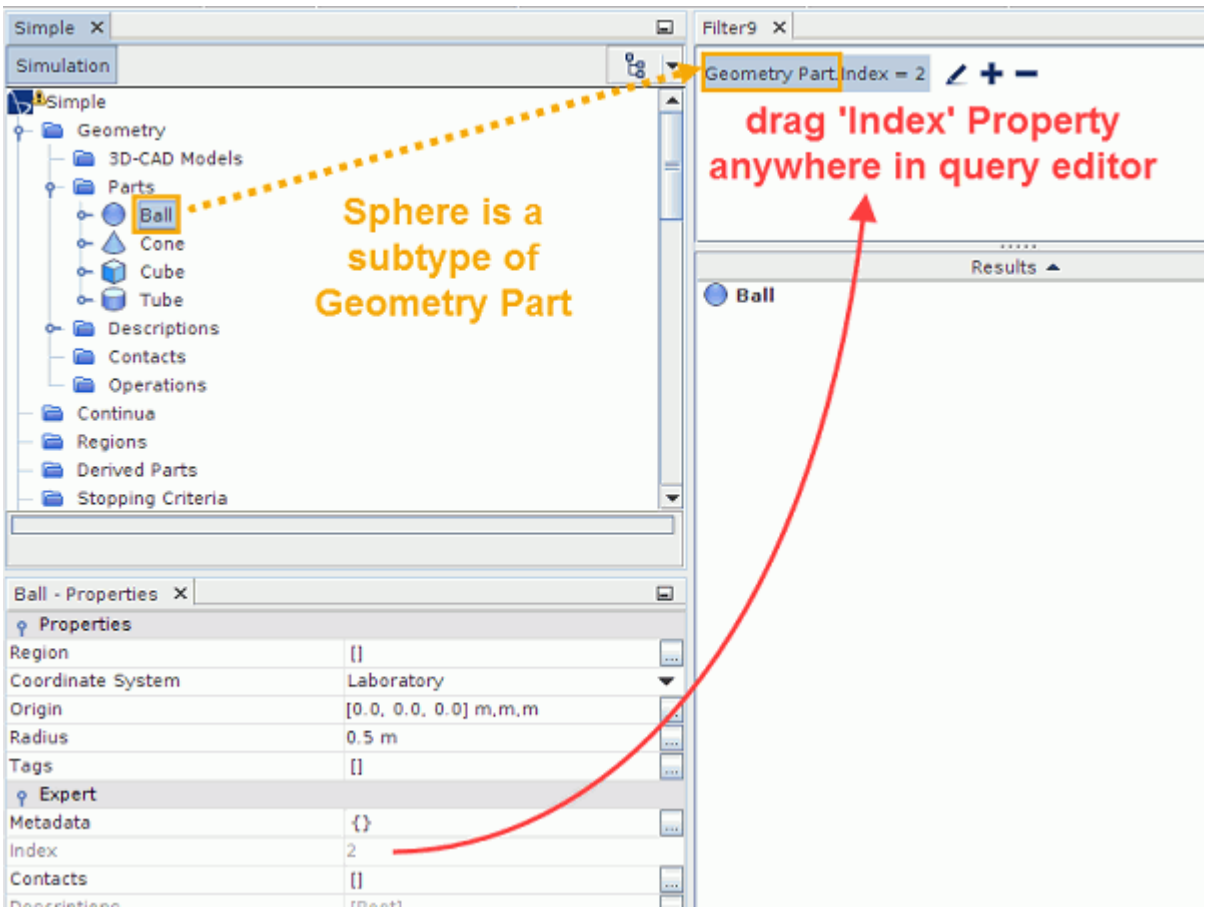
The sphere has a *Radius* property that is unique to it (geometrically). As a result, the specific sphere type appears in a predicate when you drag and drop that property into the query editor:

```
[Sphere Part.Radius = 0.5 m]
```



However, if you drag and drop the *Index* property of the sphere, the predicate reflects the fact that the sphere shares that property with other parts in the *Geometry Parts* class:

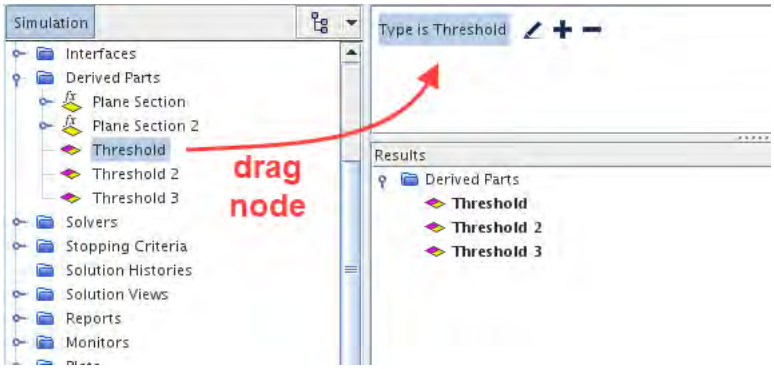
```
[Geometry Part.Index = 2]
```



Controlling Object Types in Predicates

Suppose that you have a threshold derived part and want to create a predicate that limits the search results to the type threshold. This option is not available in the drop-down lists for manual creation of predicates.

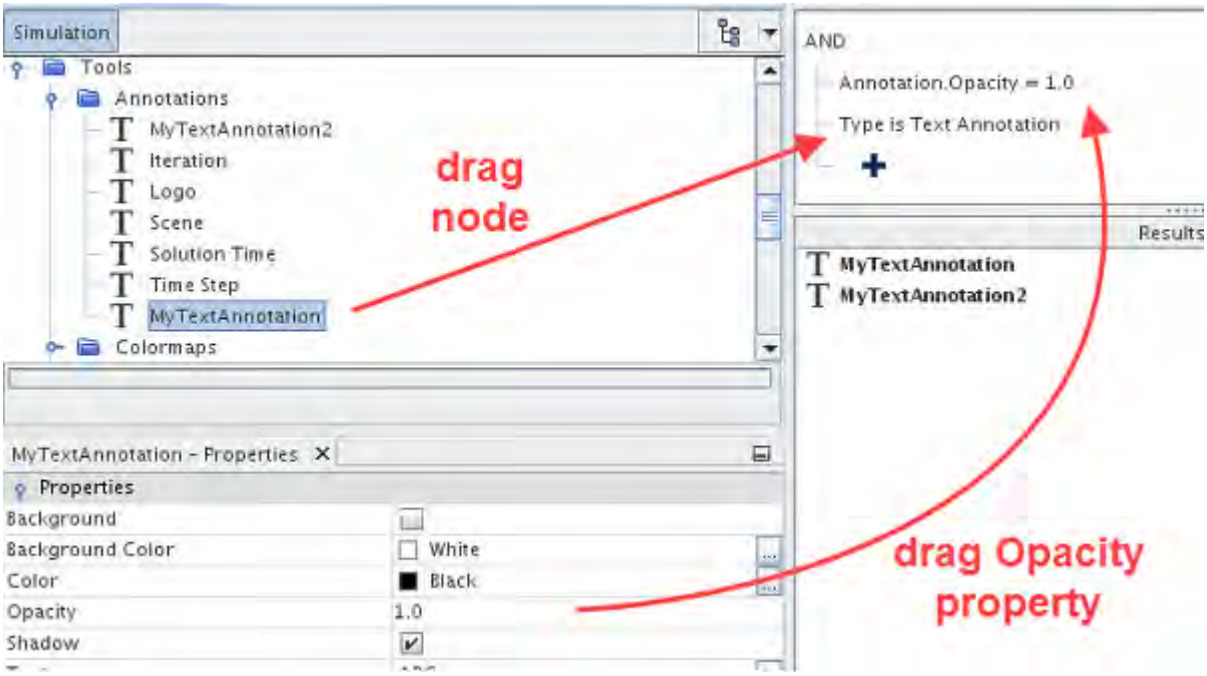
However, you can drag any node into the query editor and a Type query for that node appears. In the example that follows, if you drag and drop your **Threshold** node, a `[Type is Threshold]` predicate appears in the query editor.



This principle of type restrictions can apply to the example of annotations. If you drag the *Opacity* property of a simple text annotation into the query editor, the results of the query include objects of type iteration, logo, and simple text, since all of them are annotations with the *Opacity* property.

To restrict the results to simple text annotations, drag a simple text annotation node into the query editor where the initial predicate `[Annotations.Opacity = 1.0]` is defined. You get a *compound* predicate:

```
[Annotation.Opacity = 1.0] AND [Type is Text Annotation]
```



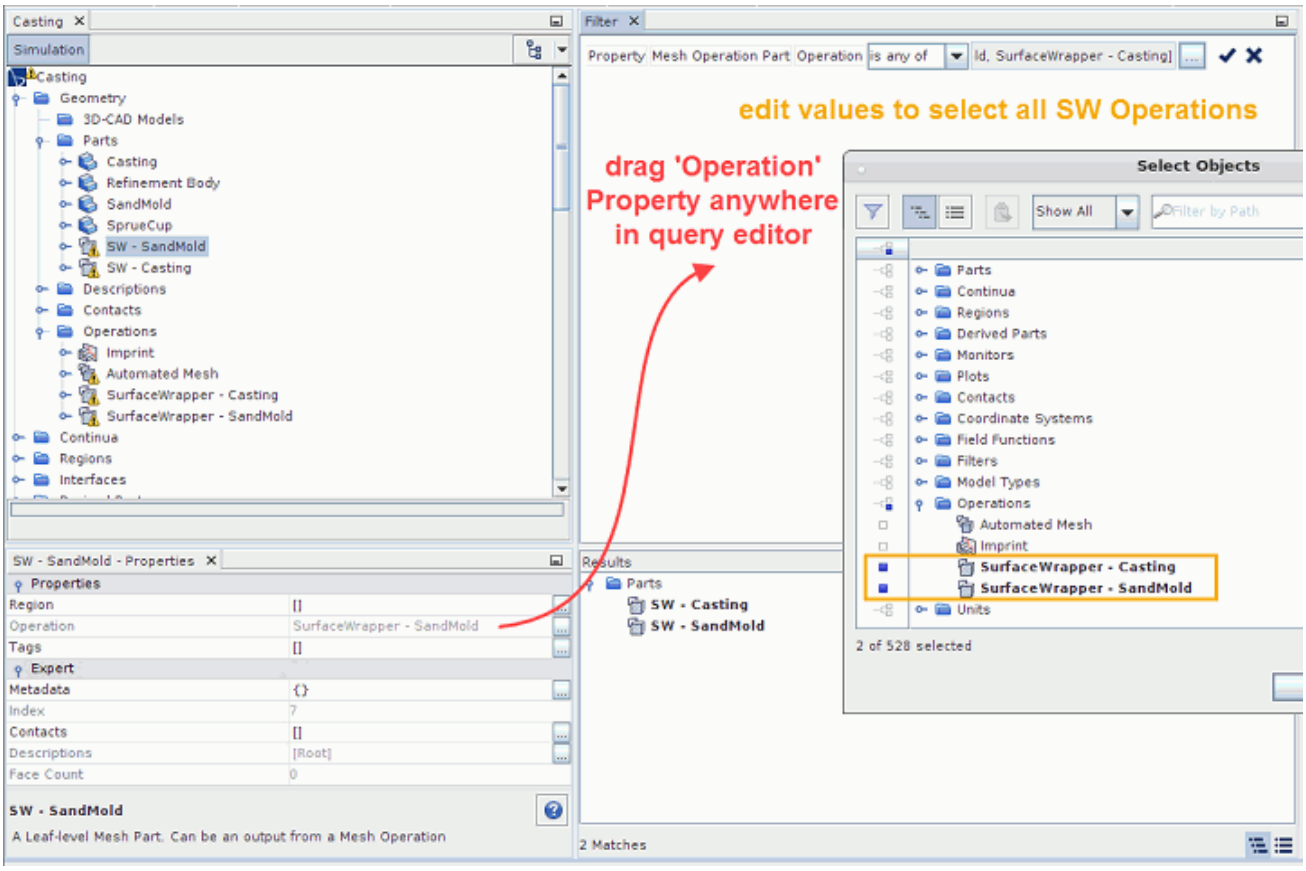
The second predicate restricts the results to text annotations.

Nested Queries

For more powerful queries, use the results of a filter as the value of a predicate. This technique is called a *nested query*.

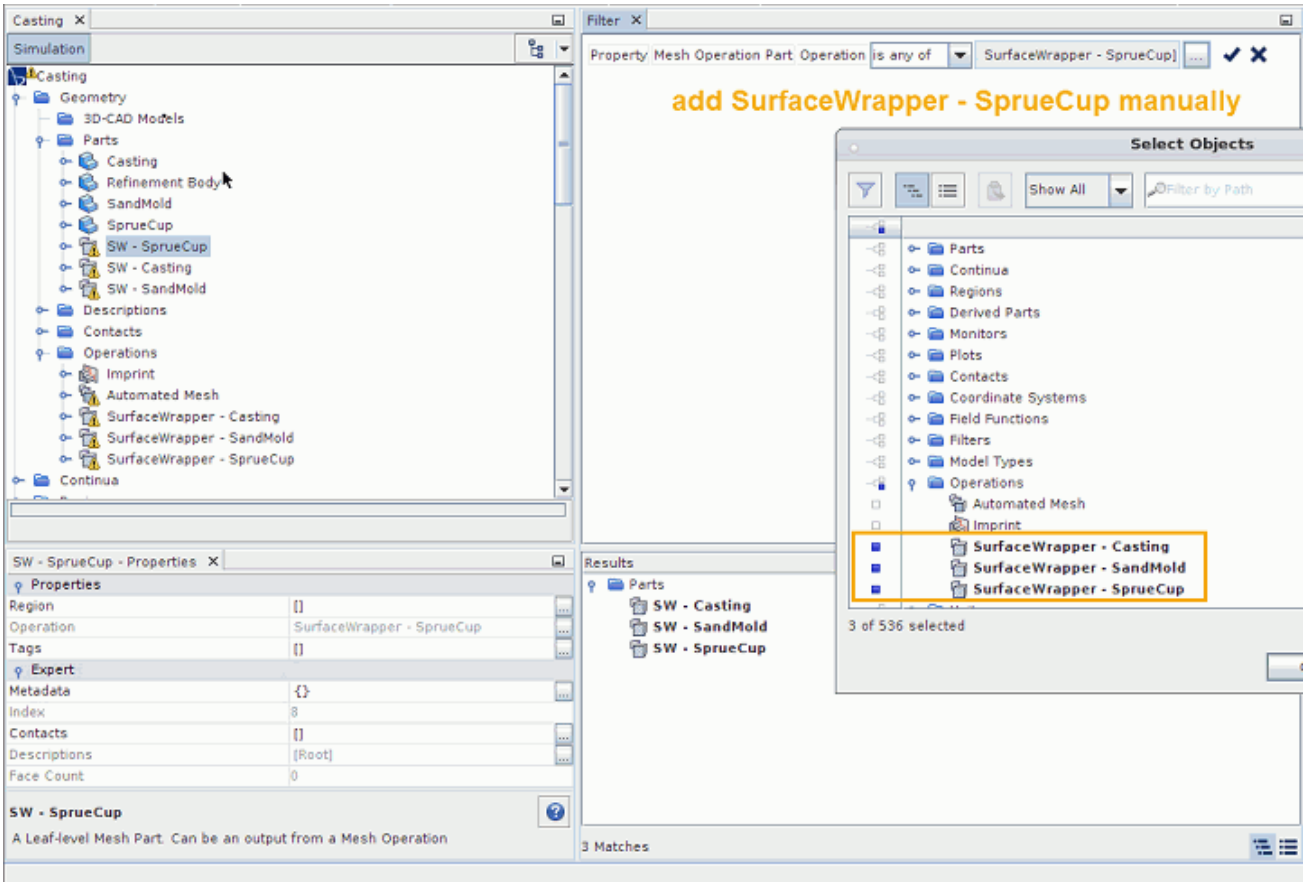
For example, suppose that you want to identify all the parts that originated from surface wrapper operations.

The initial query would resemble what is shown below. The query uses a property predicate from the *Operation* property of a mesh operation part, with all surface wrapper operations selected. The query results list all parts that are generated by surface wrapper operations.



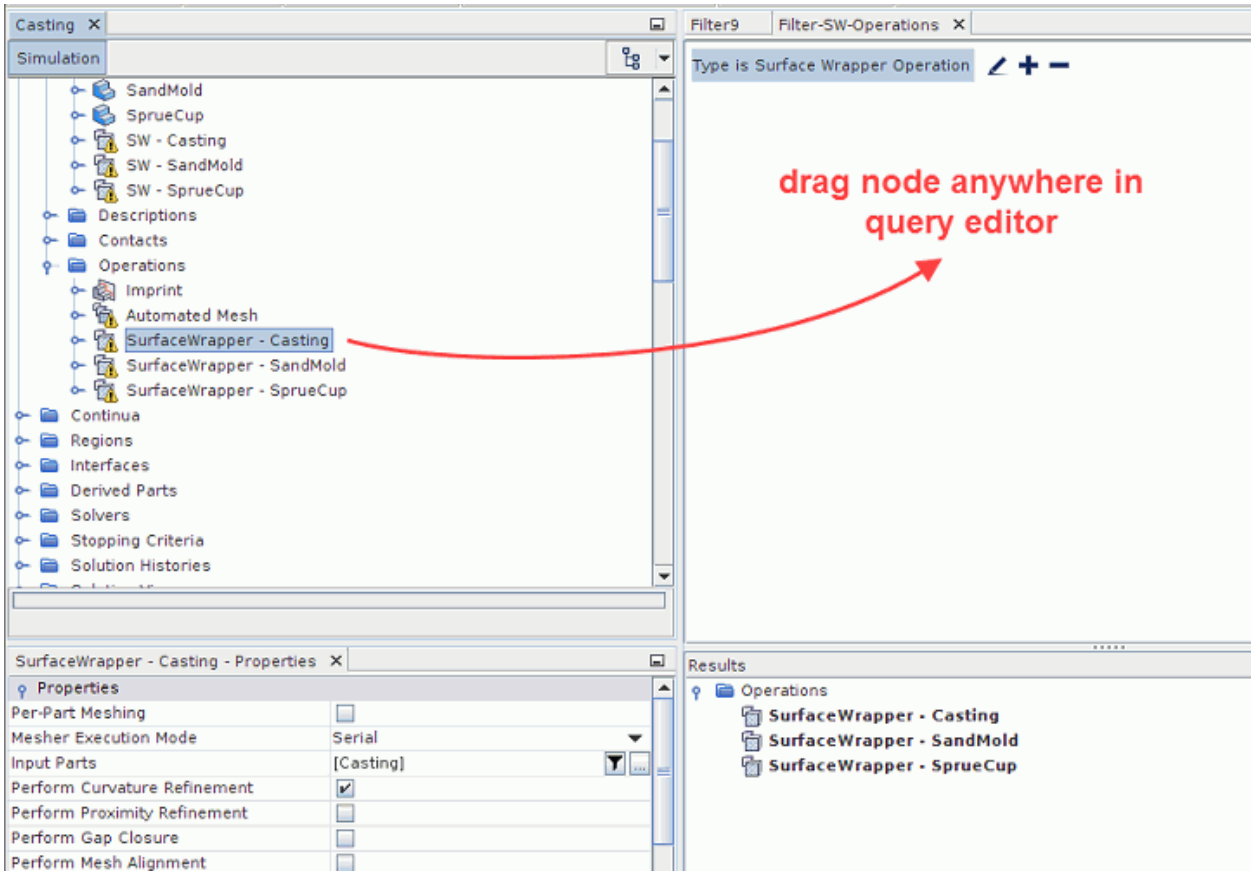
However, if you create another surface wrapper operation, it would not appear in the results automatically. Select it manually in the value of the property predicate.

In the example below, **SurfaceWrapper - SprueCup** has been added manually to the query.

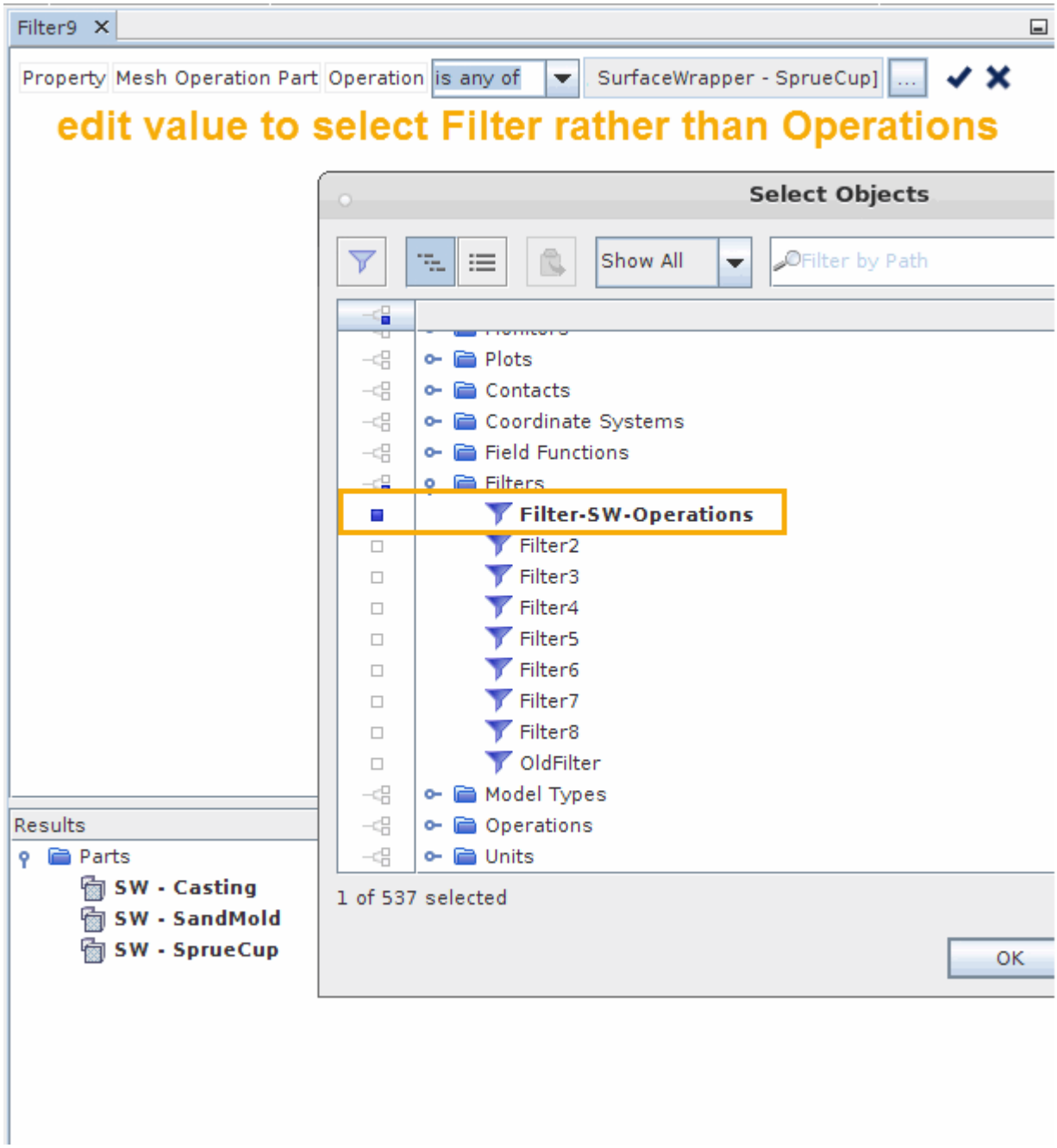


To avoid manual adjustments to filters, use nested filters as shown in the following example:

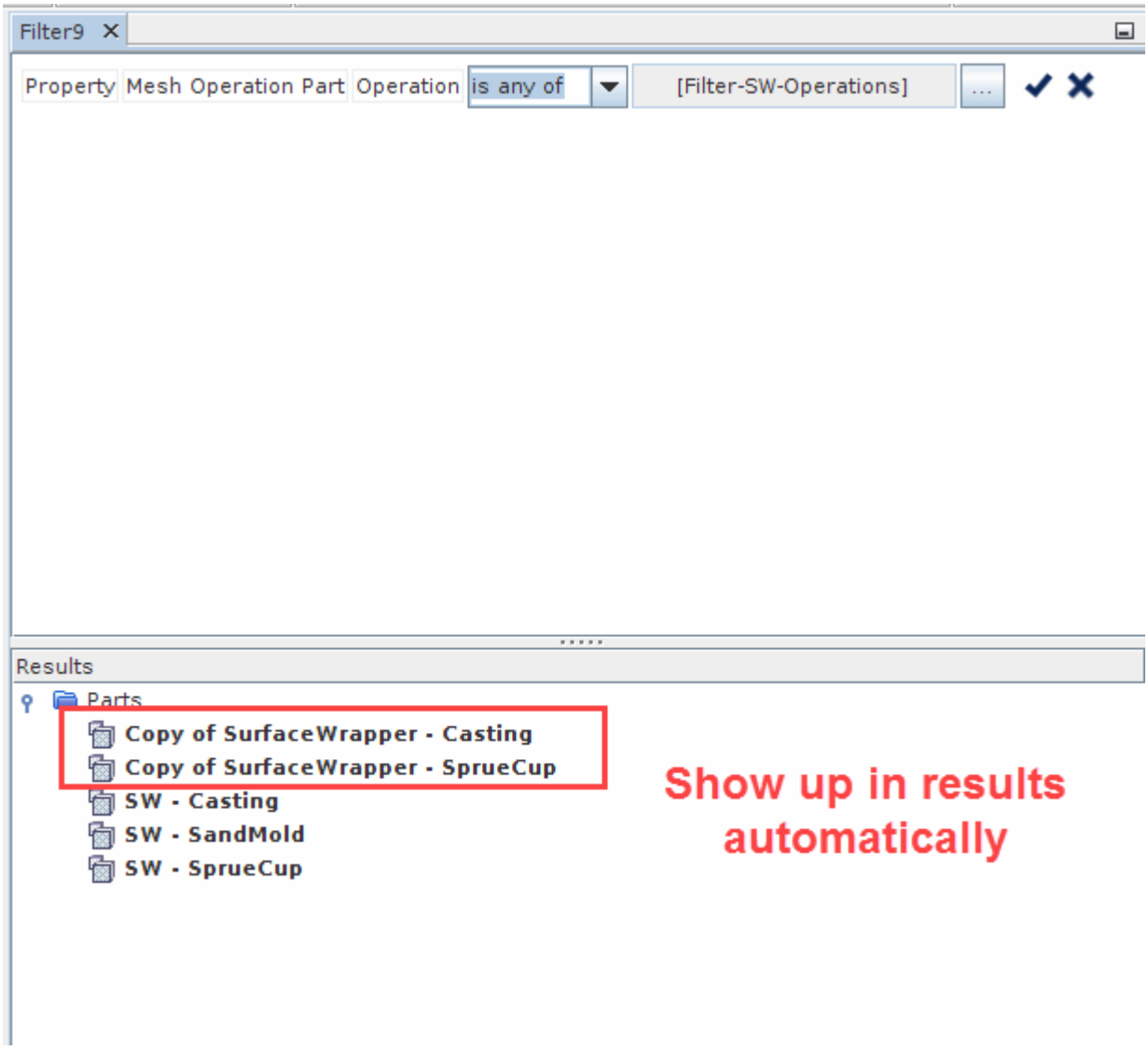
1. Create a filter called `Filter-SW-Operations`, using a Type predicate by dragging a surface wrapper operation node into the query filter.



2. In the original query (named **Filter 9** in this example), change the value from a set of selected operations to the filter **Filter-SW-Operations**.



The filter **Filter-SW-Operations** is nested within **Filter 9**.



To test if this filter automatically picks up new surface wrapper operations, make a few copies of the existing operations and see that the new ones show up immediately.

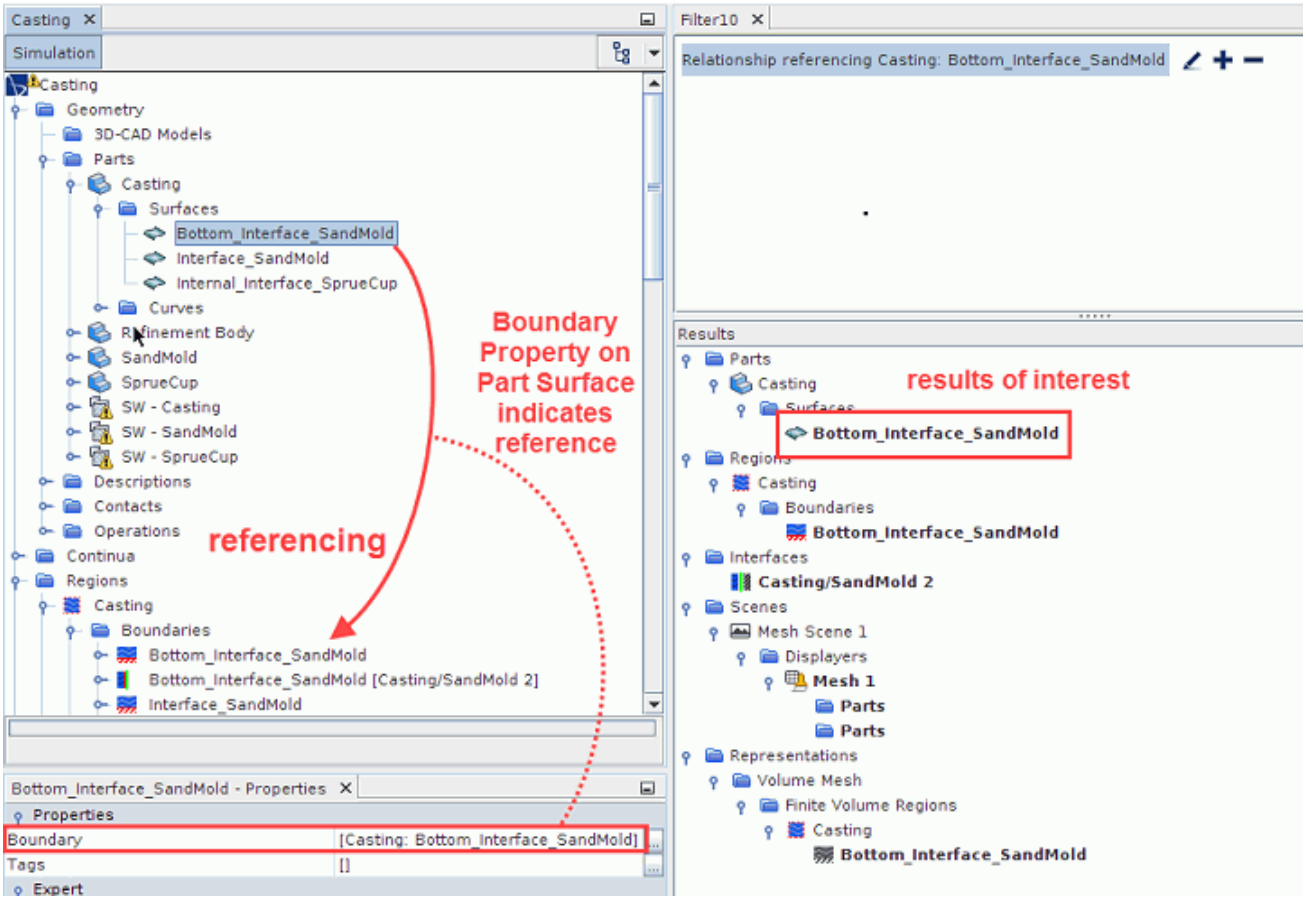
Relationship Predicates

This type of predicate includes tree hierarchy functionality (from versions of STAR-CCM+ prior to v11.06) and current referencing operators.

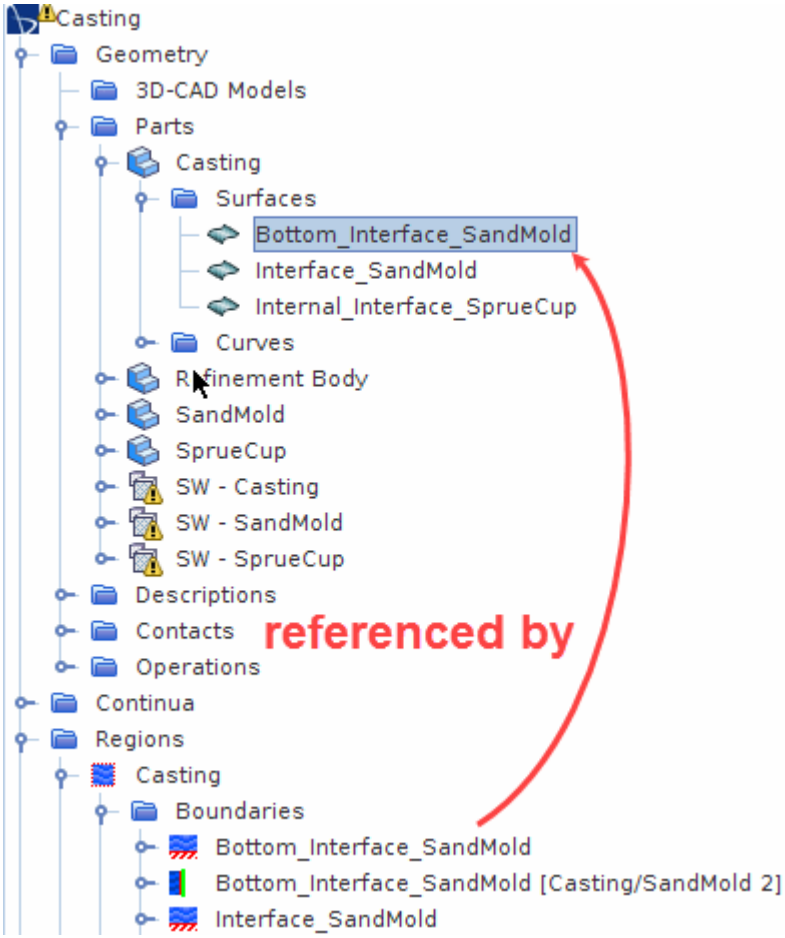
For a list of the available operators for the relationship predicate, see Predicate Reference.

Suppose a part surface **Bottom_Interface_SandMold** has a *Boundary* property **Casting: Bottom_Interface_SandMold**. That part surface is *referencing* the boundary **Bottom_Interface_SandMold**.

If you create the predicate `[Relationship referencing Casting: Bottom_Interface_SandMold]`, the results include all objects that reference the boundary.



Similarly, the boundary **Bottom_Interface_SandMold** is *referenced by* the part surface **Bottom_Interface_SandMold**.

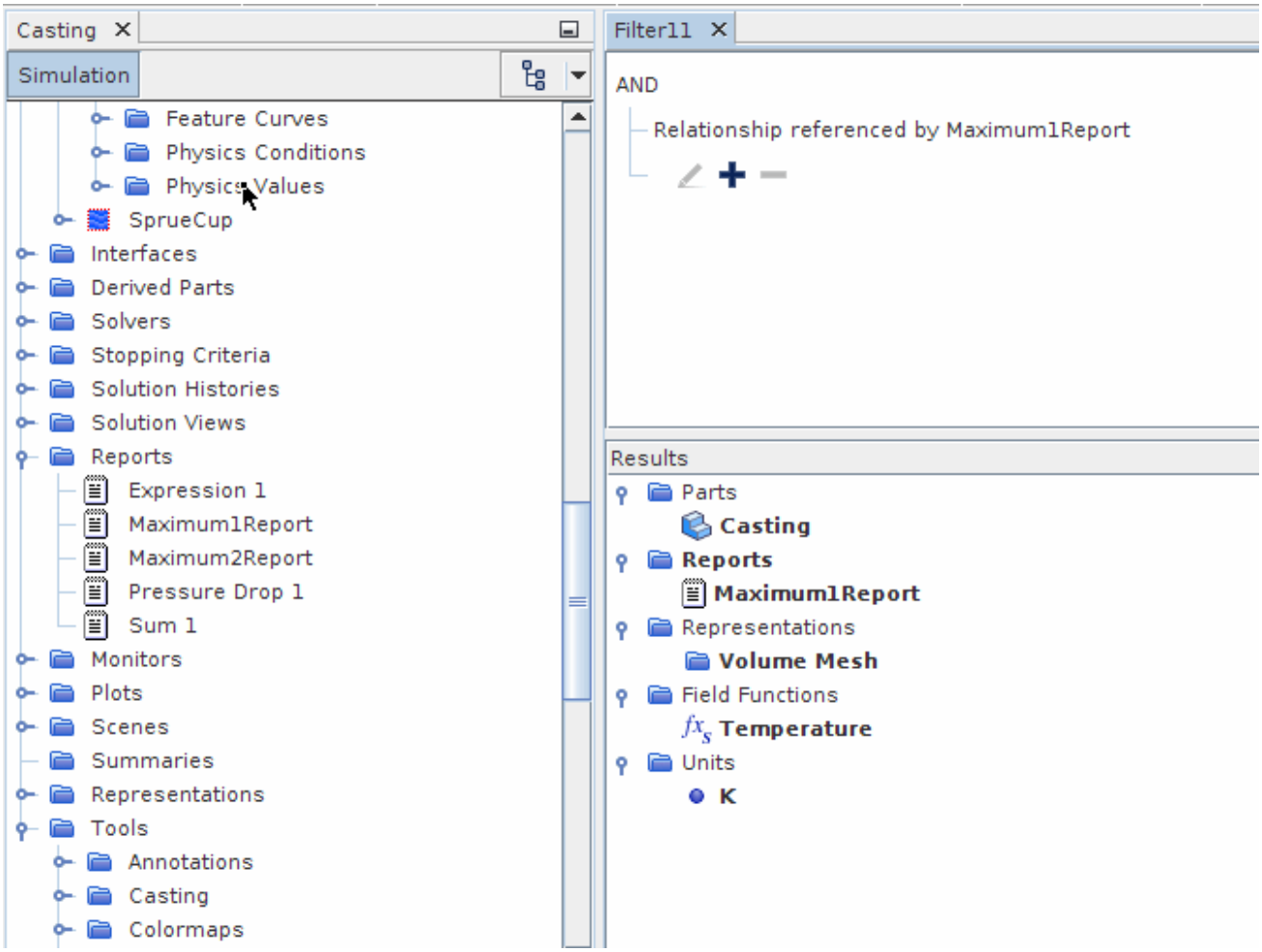


Since the *Boundary* property of a part surface already provides the referencing information, why is it useful to have a query for it? The following examples clarify this and other aspects of relationship predicates.

Example: Finding Geometry Parts in a Set of Reports

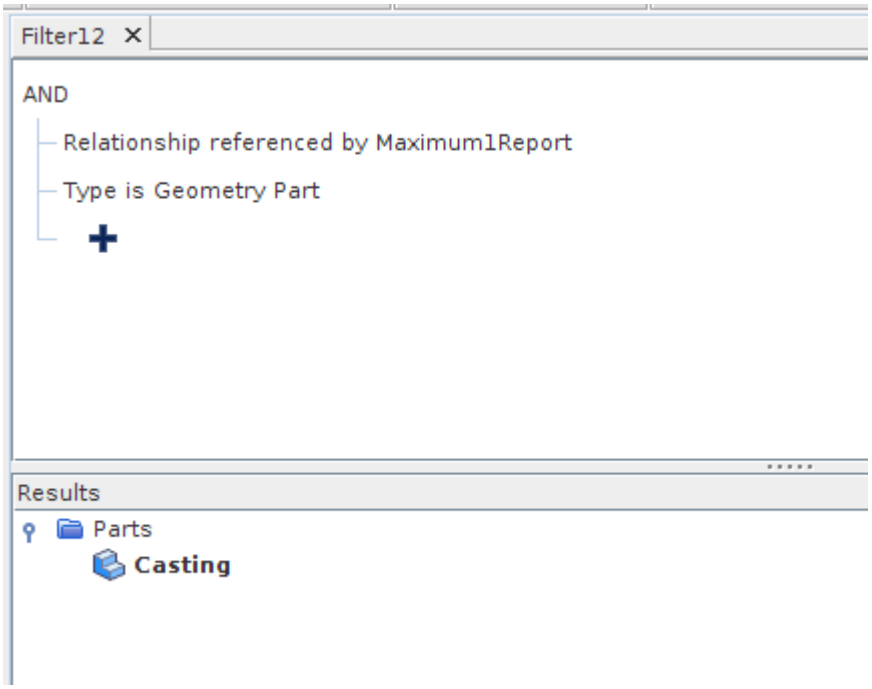
Suppose that a report called **Maximum1Report** has a part called **Casting** and **Maximum2Report** has a part called **SandMold**. A relationship predicate can make a query work as follows:

```
[Relationship referenced by Maximum1Report]
```



This query returns the **Casting** node, as well several other objects that are also referenced by **Maximum1Report** (for example the **Units** node and the **Representations** node). To restrict the search results to the geometry part **Casting**, add a type predicate:

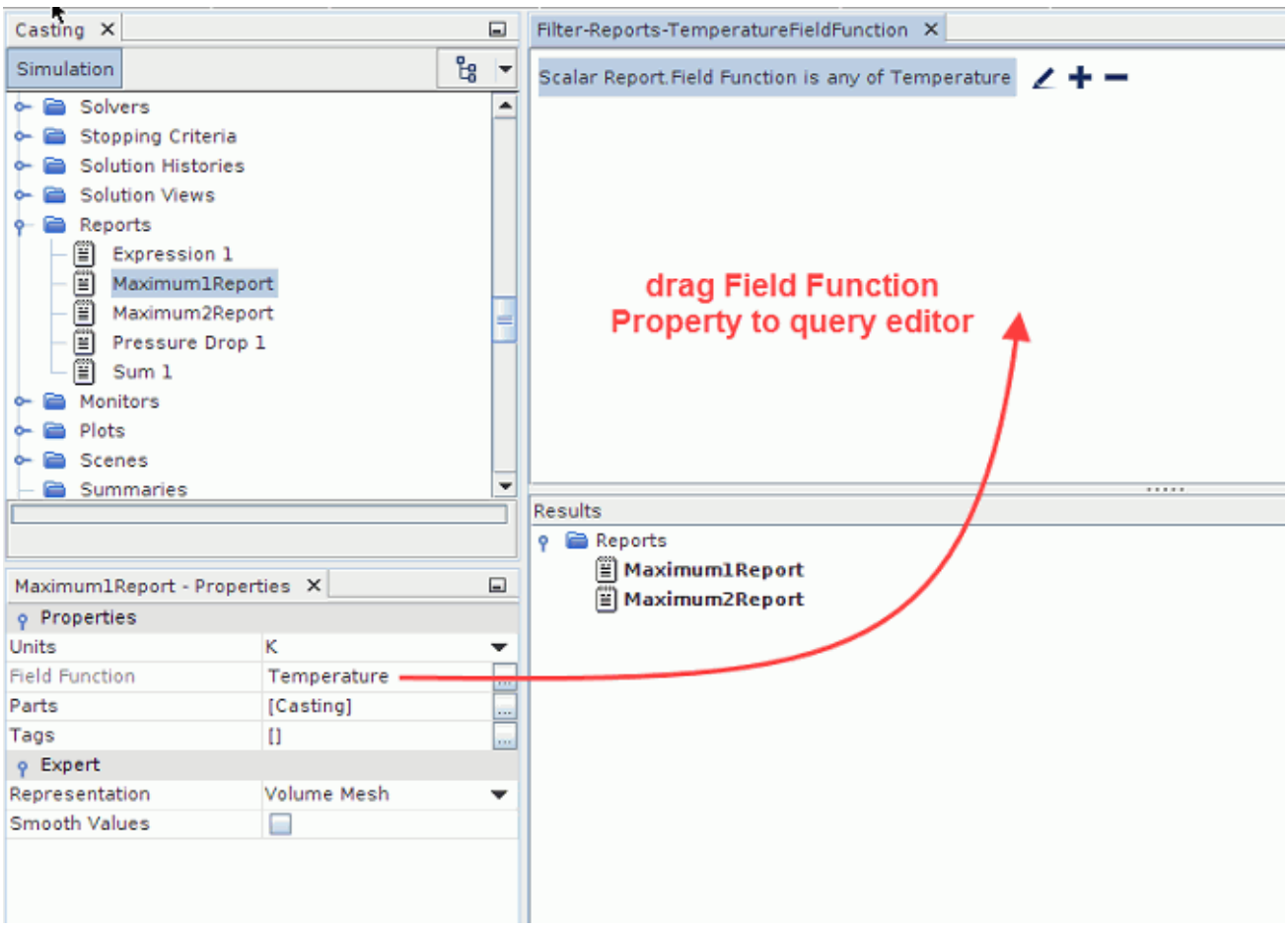
```
[Relationship 'referenced by' Maximum1Report] AND [Type is Geometry Part]
```



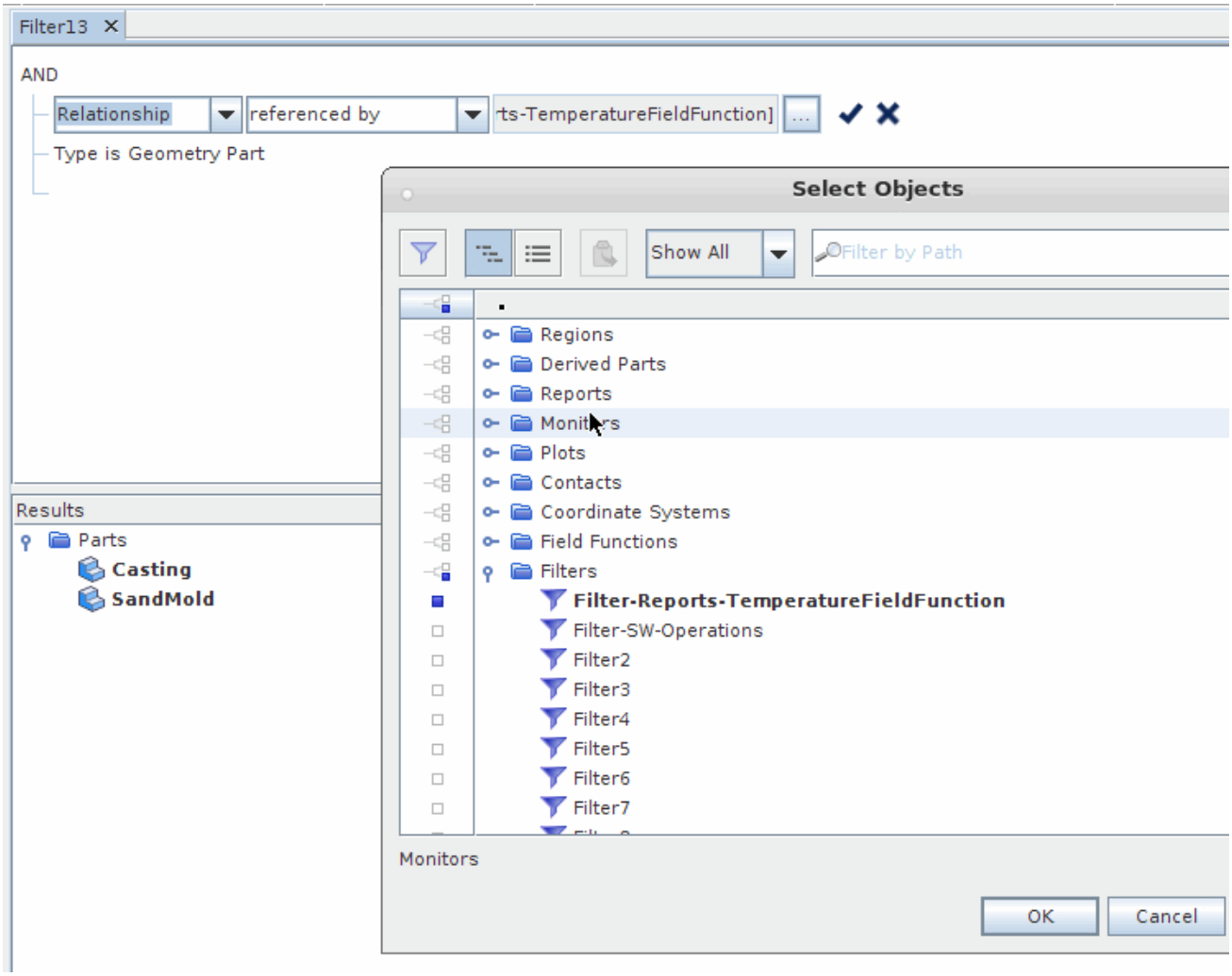
This query represents all geometry parts, in a particular report (**Maximum1Report**); in this case, it returns only **Casting**.

Suppose that you are interested in all reports that utilize the field function **Temperature**, and want to find all geometry parts that are used in that class of reports. A tedious and error-prone approach would be selecting each report node, looking at the *Field Function* property on each one, and writing down the ones that use the **Temperature** field function. You could then create a predicate with a static list consisting of the items you wrote down.

A better approach is to use a nested query together with the relationship/referenced by predicate. First, create the inner filter using a property predicate. This step gives you all reports utilizing the field function **Temperature**.



Next, to create the outer filter, modify the initial filter by replacing **Maximum1Report** with the filter **Filter-Reports-TemperatureFieldFunction**.

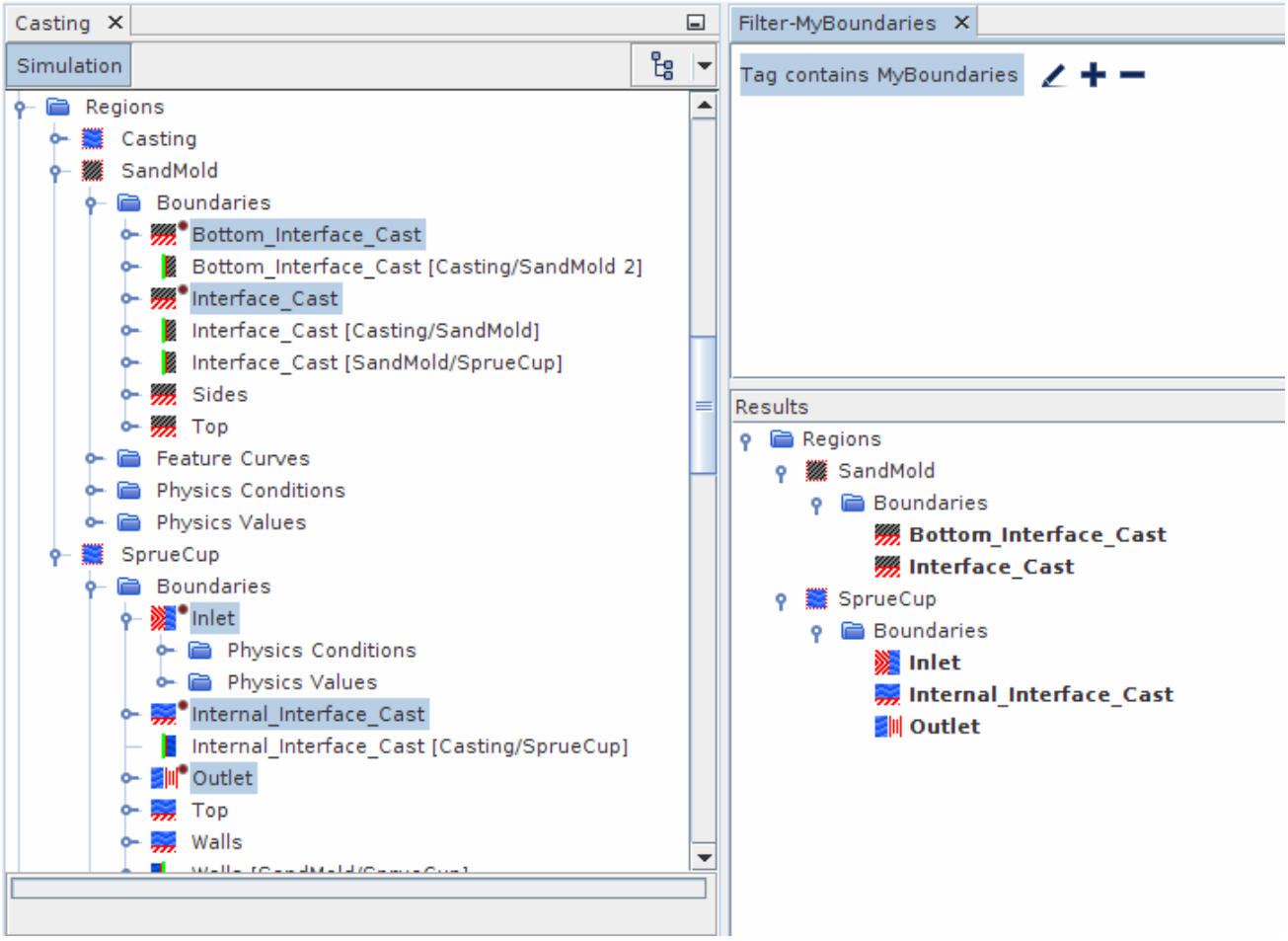


This filter is dynamic in the sense that if any other reports with the **Temperature** field function are added, the results get updated automatically with those reports.

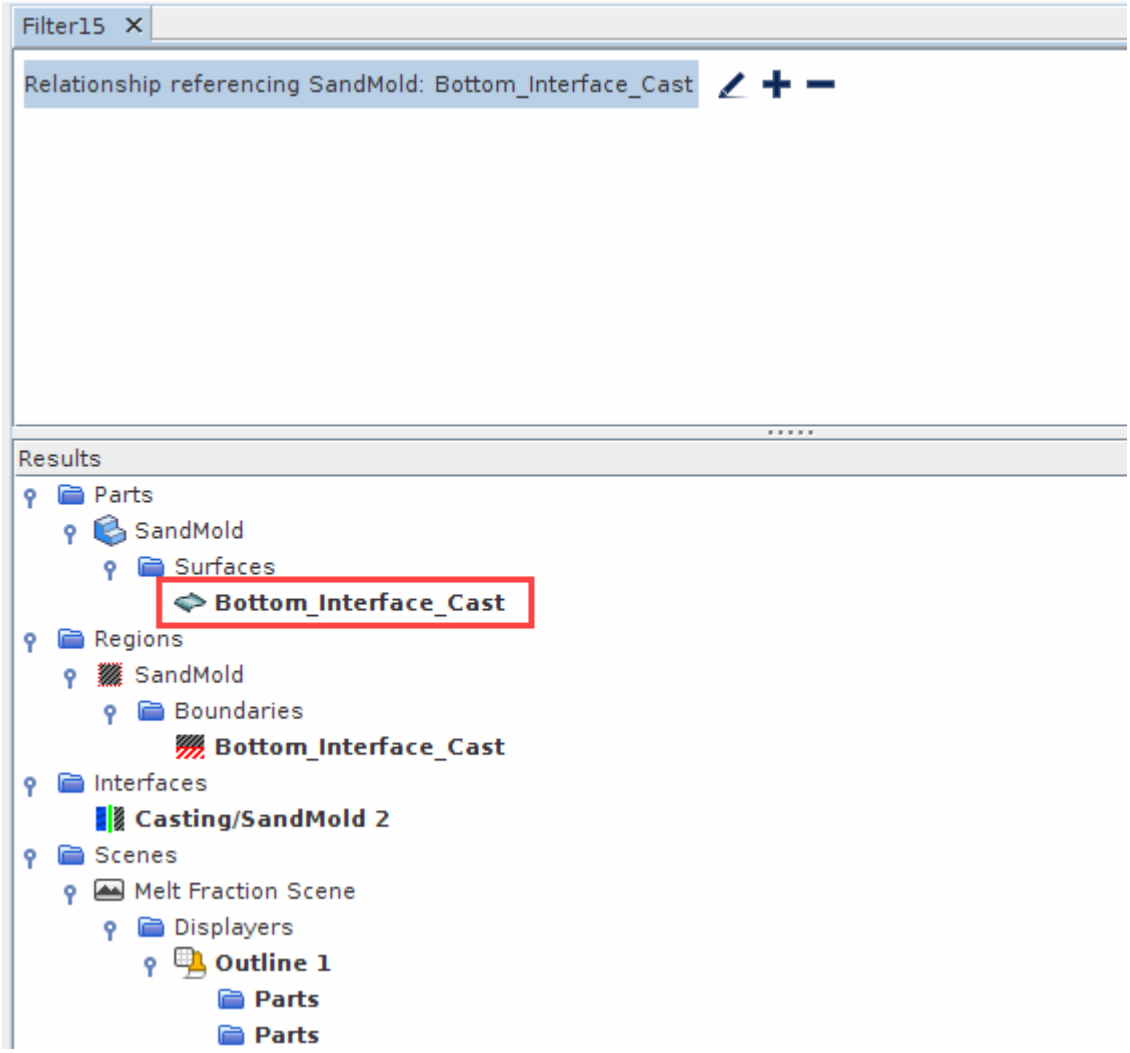
Example: Finding Part Surfaces Assigned to Tagged Boundaries

Suppose that you have specified some boundaries of interest with the tag **MyBoundaries**. To find all part surfaces that are assigned to these boundaries, use a relationship/referencing nested query that is combined with a tag predicate.

In this example, the tag query is `[Tag contains MyBoundaries]`, in a filter called **Filter-MyBoundaries**. As shown in the following screenshot, five tagged boundaries are selected in the tree, and the query results appear at right as expected.



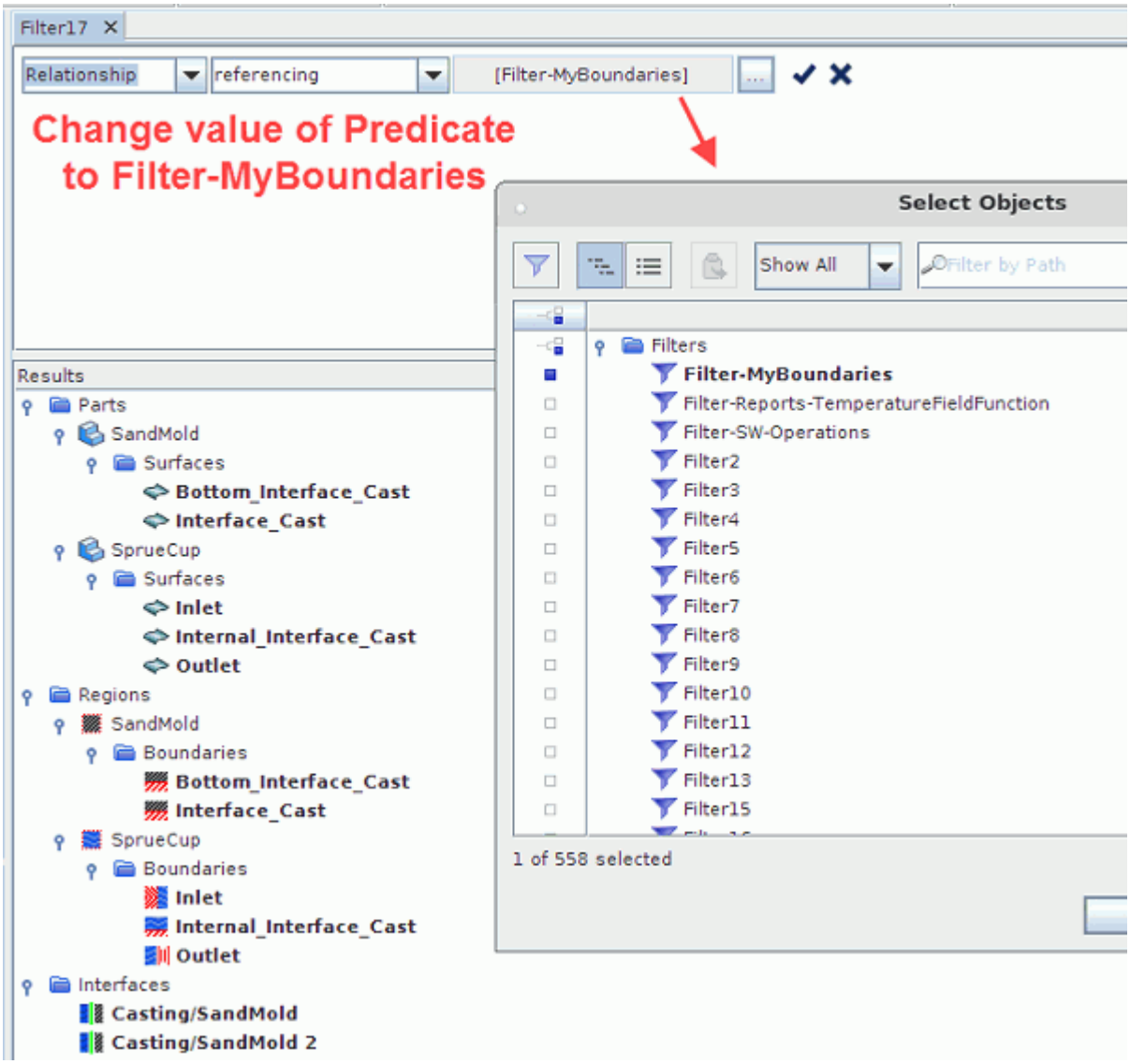
Each of these boundaries can have its own set of referencing objects. For instance, below are the results for **Bottom_Interface_Cast**. The part surface **Bottom_Interface_Cast** appears first.



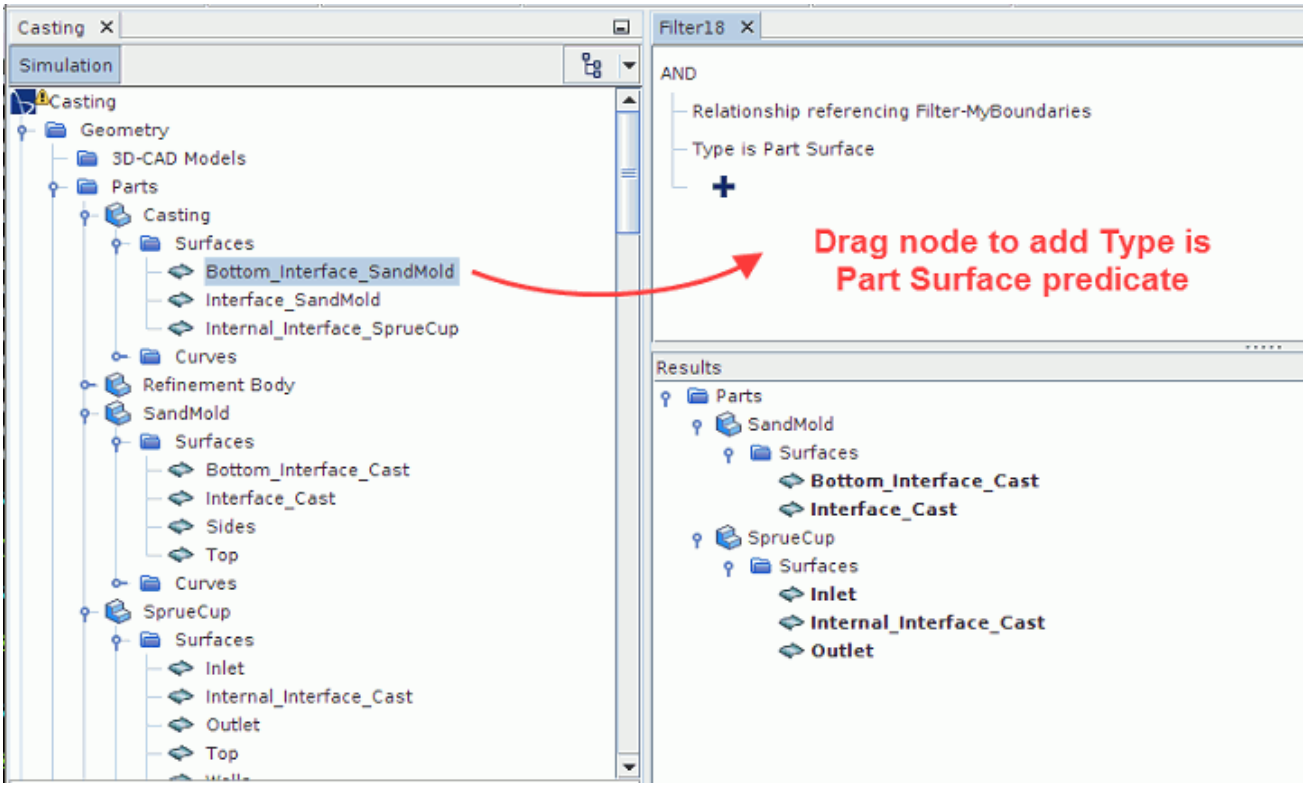
You can generalize this query to find all part surfaces that reference boundaries that are tagged with **MyBoundaries**. The query result contains all the objects that reference the boundaries, including objects besides the part surfaces. There are two possible approaches for generalizing the query: static and dynamic.

The static approach, which is tedious and error-prone, is to modify the relationship/referencing predicate to include all tagged boundaries, selecting them one at a time in the object selector.

The dynamic approach, which is more flexible and automatic, is to modify the relationship/referencing predicate to have a filter value of **Filter-MyBoundaries**. This option returns all objects that reference the items listed in the **Filter-MyBoundaries** filter.

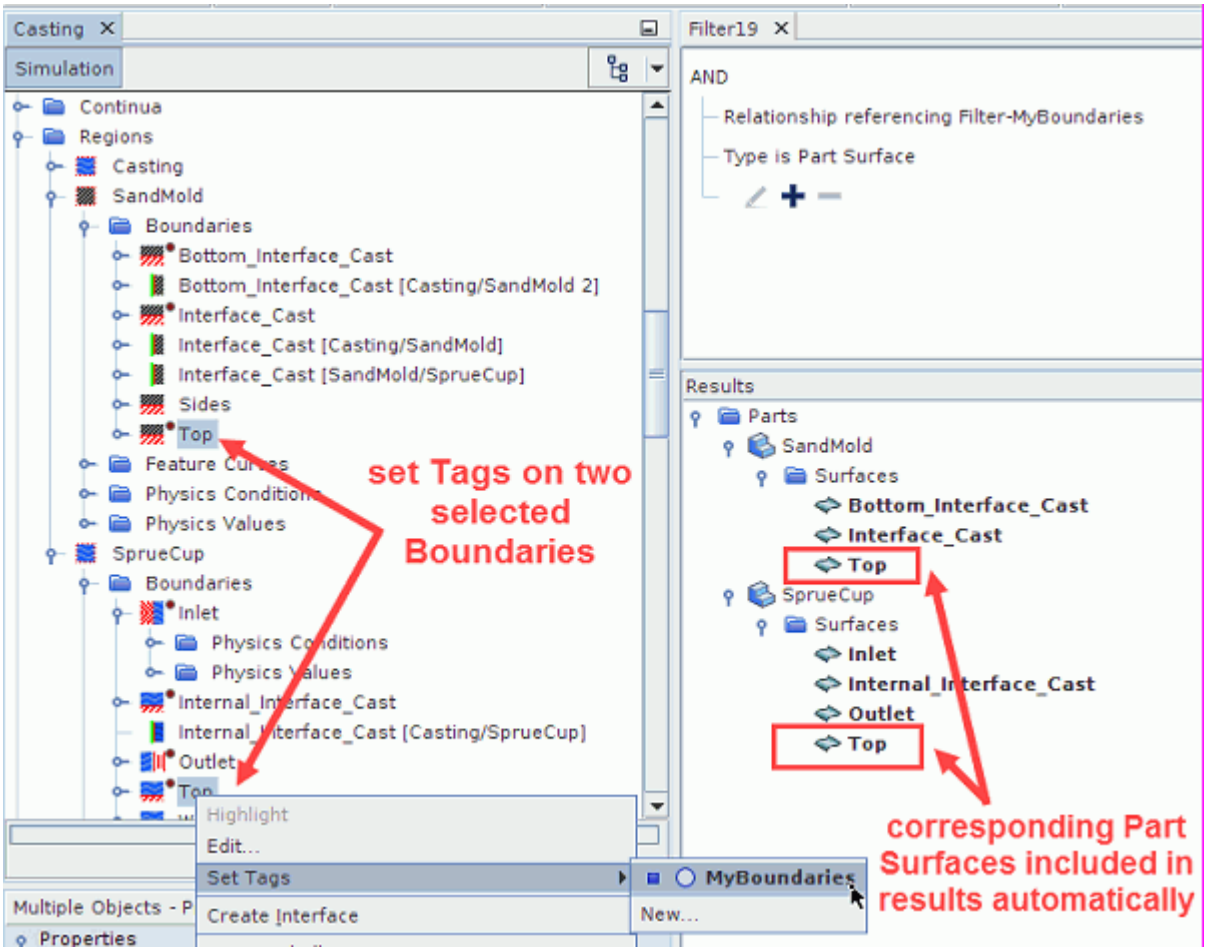


To narrow the scope of the query results to part surfaces, combine this predicate with a `[Type is Part Surface]` predicate.



The resulting compound predicate is `[Relationship referencing MyBoundariesFilter] AND [Type is Part Surface]` where `MyBoundariesFilter` is `[Tag contains MyBoundaries]`.

If you tag more boundaries with the **MyBoundaries** tag, the query results get updated automatically, which makes this approach dynamic.



Example: Finding Regions with a Particular Continuum

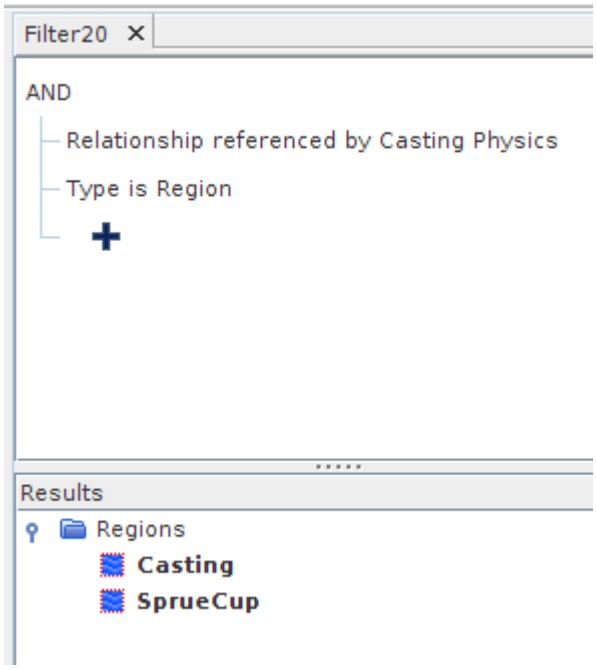
There are instances where property predicates are not available. For example, the **Physics Continuum** property of a region cannot be dragged into the query editor.

To find all regions that are associated with a particular physics continuum:

1. Create a relationship predicate with the **referenced by** operator.
2. For the value of the query, select the continuum of interest (in the example below, **Casting Physics**).

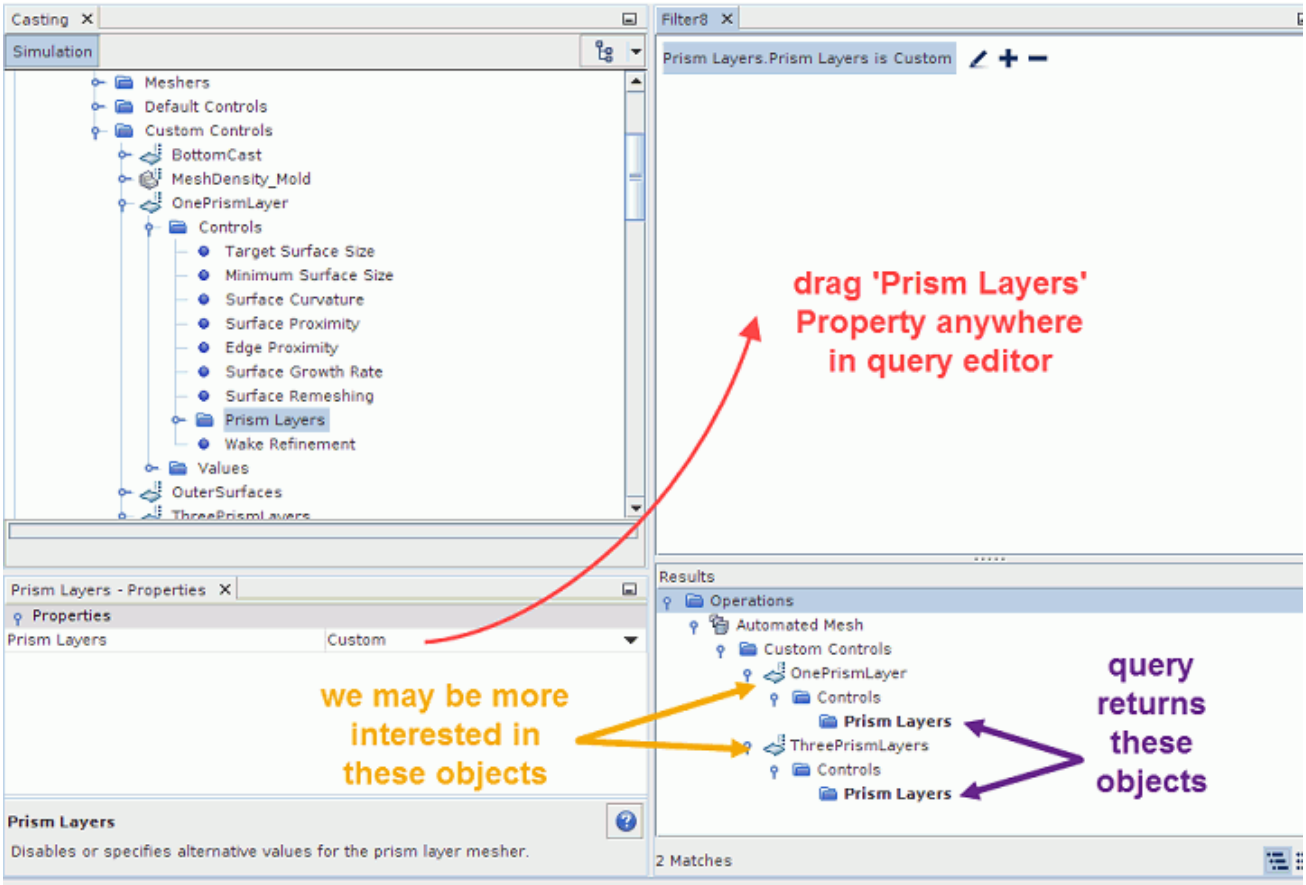
   The initial results include various types of objects such as regions and interfaces.
3. To restrict the query results to regions, drag the **Regions** node into the query editor, creating a compound predicate:

   ```
   [Relationship referenced by Casting Physics] AND [Type is Region]
   ```



Example: Finding Custom Controls with Prism Layers of Type Parent

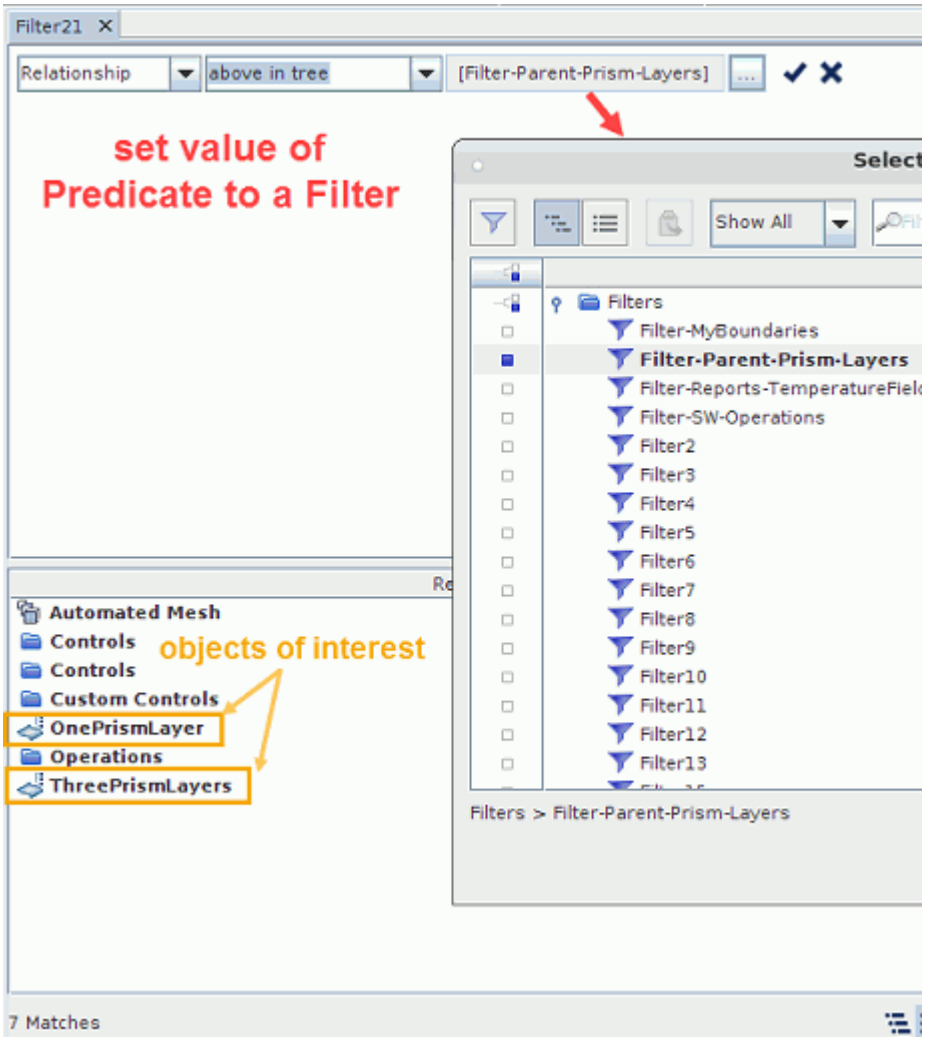
In Property Predicates, one example involves finding prism layers that were set to **Custom**. As it is, the resulting query gives non-named **Prism Layers** nodes, which are the same type as the owning object, **Controls**. However, the objects of interest are the actual named custom control nodes (in this example **OnePrismLayer** and **ThreePrismLayers**), which are parents of the nodes listed in the query results.



To retrieve these named objects, use a nested query with the technique in the following example:

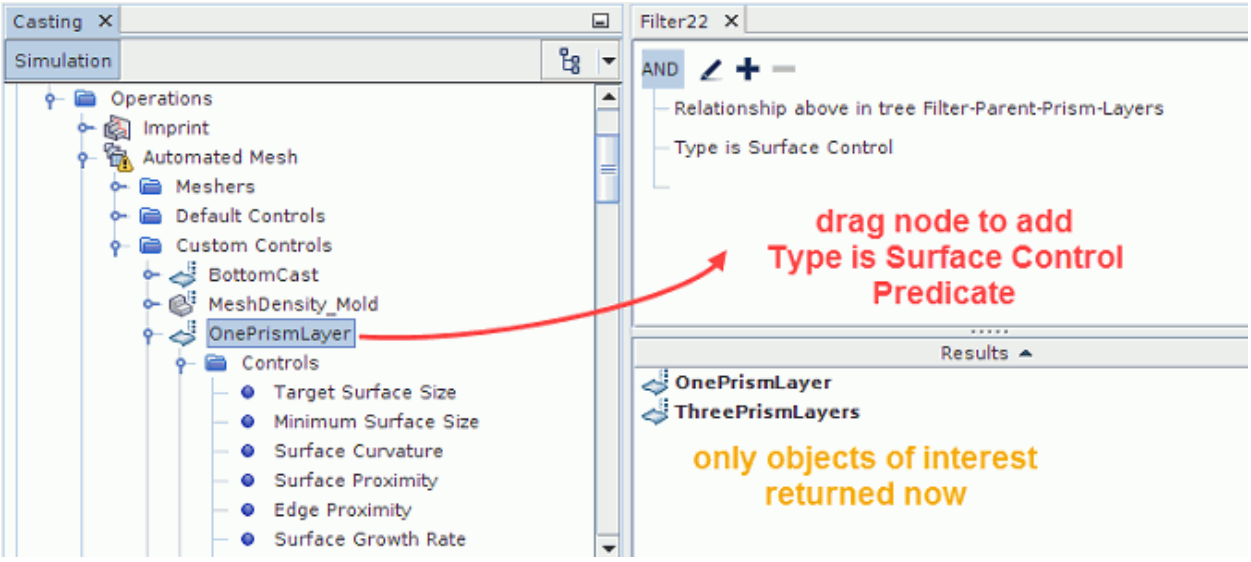
1. Name the existing query `Filter-Parent-Prism-Layers`. Observe that the custom controls of interest are hierarchically above the **Prism Layers** nodes that the first filter found.
2. To redirect search results and capture the actual objects of interest, create a query with the following:
   - a **Relationship** predicate
   - an **above in tree** operator
   - a value that selects the filter **Filter-Parent-Prism-Layers**

   Although the query results include the custom controls, they also include other objects that are not of interest.



3. To restrict the results, combine this predicate with a `[Type is Surface Control]` predicate.

   This modification provides a list of all custom controls that contain custom prism layers. As objects of this type are added to the tree, the query is updated automatically.



Working with Filters

This section explains how to create filters, modify existing ones, and use them to select certain nodes.

Contents:

Creating a Filter
Selecting Nodes through a Filter

Creating a Filter

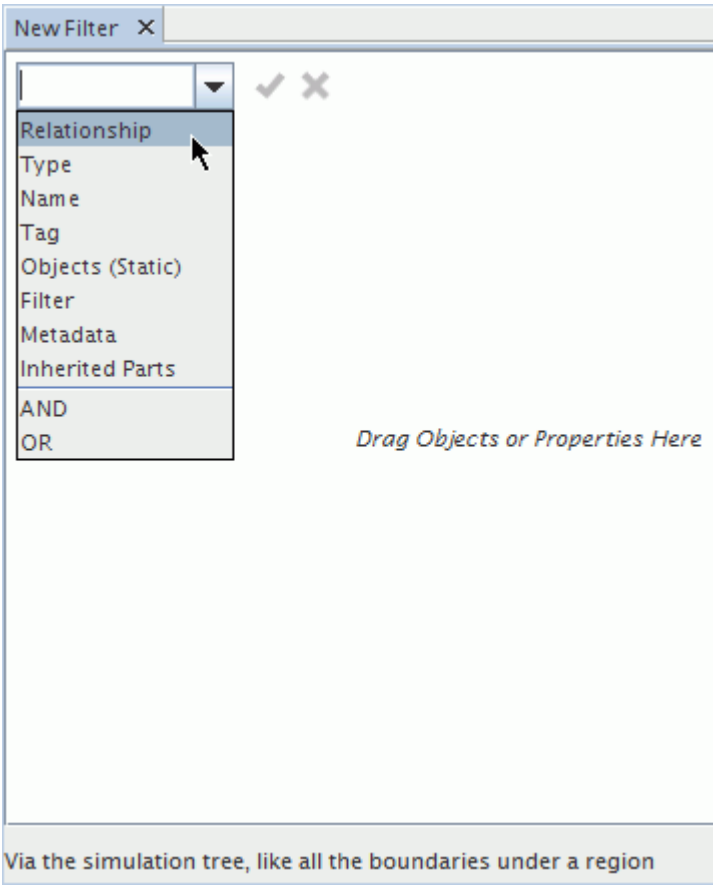
You can add a filter and edit existing filters.

To create a filter:

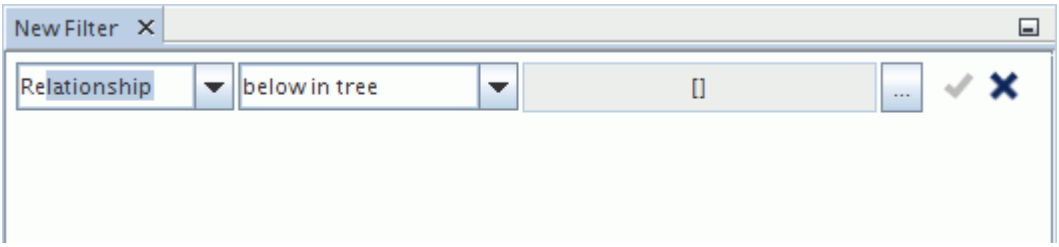
1. Right-click the **Tools** > **Filters** node and select **New**.

   A new node named **Filter** appears.
2. In the dialog, do one of the following by making a selection in the drop-down list:
   - Begin creating a predicate by selecting a type.
   - To create a compound predicate in which objects match all of the contained values, select **AND**.
   - To create a compound predicate in which objects match any of the contained values, select **OR**.

   As you hover with the mouse over the options in the list, text appears at the bottom left of the dialog to identify the option.



You can also type into the text box of the drop-down list.



To select the desired entry, navigate to it with the `<Up>` or `<Down>` arrow keys and press `<Enter>`.

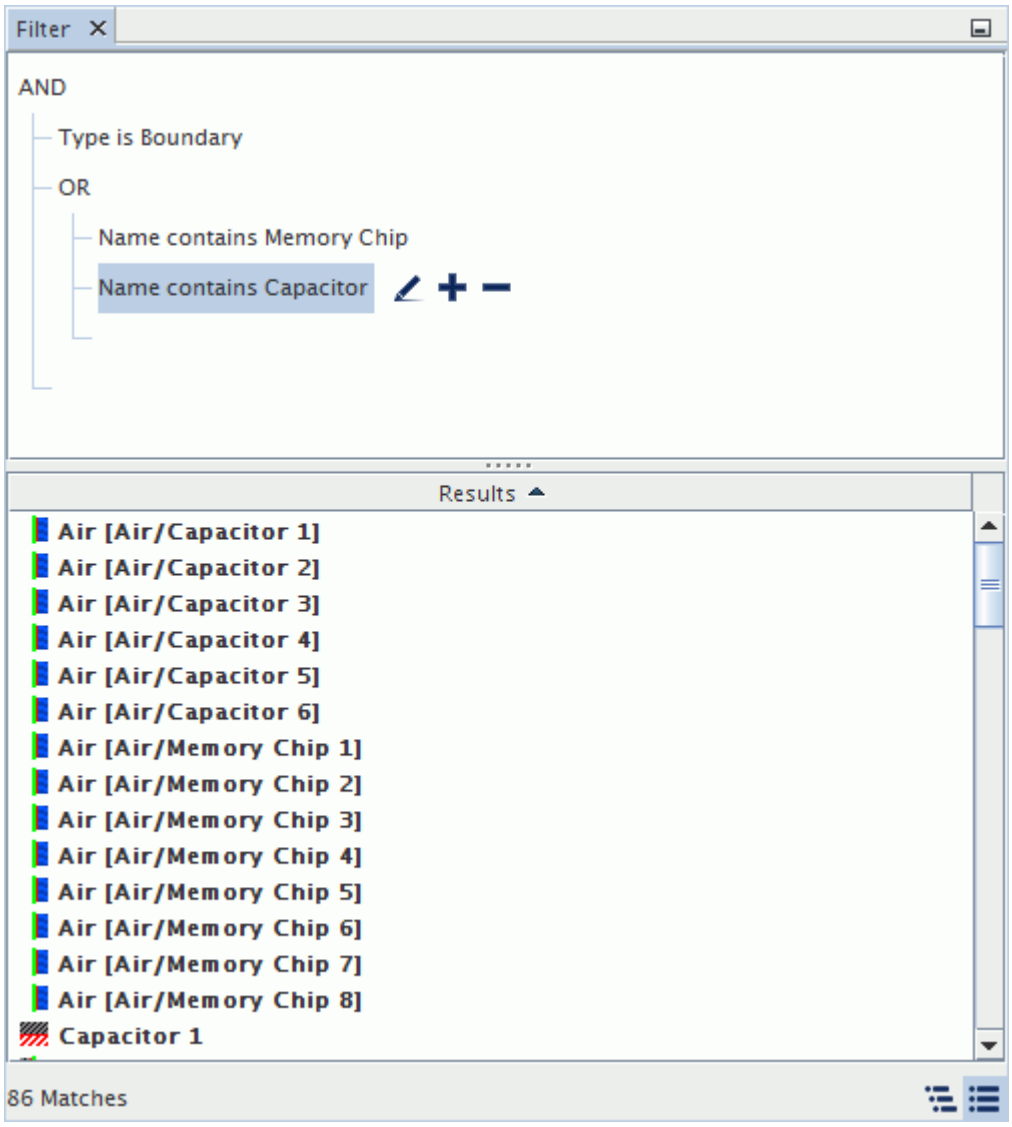
3. Complete the predicate by selecting the other components.

4. To finish specifying the predicate, click ✔ (**Commit changes**). Alternatively, click anywhere in the view outside the components of the predicate, or press `<Enter>`. To cancel the creation of the predicate, click ✖ (**Discard changes**).

5. Use the actions available on the dialog to create further predicates. See Filter Dialog Reference for more information.

In the following example, the filter contains results of this search:

```
(Type is Boundary) AND ((Name contains "Memory Chip") OR (Name
contains "Capacitor"))
```



At the bottom left of the dialog, text indicates the number of matches.

6. Click **OK**.
7. If necessary, rename the filter to an appropriate mnemonic.
8. To apply the filter in an object selector, see Filtering Using Filter Objects.

To view and modify the query definition of an existing filter:

9. Select the node of the filter.
10. Hover over the right half of the *Query* property with the pointer. The *Query* definition appears in a tooltip.
11. To modify an existing filter, do one of the following:

- Select the filter node and click the right half of the *Query* property.
- Right-click the filter node and select **Edit**.

The filter dialog appears with the query parameters of the filter.

You can copy and paste multiple predicates from one filter to another, even between simulations. To copy and paste predicates:

12. Right-click anywhere in the filter view outside the components of the predicate, and select **Copy**.
13. Make the window of the destination filter active.
14. Right-click in the view and select **Paste**.
15. If you do not want to include some of the copied predicates in the destination filter, you can modify or delete them.

Selecting Nodes through a Filter

You can use a filter to access a group of nodes in the object tree directly.

To select the nodes in the tree:

1. Right-click the desired filter and choose **Select Results**.

   An object selection dialog appears with all the search results of the filter. This feature gives you the opportunity to omit some of the nodes from your selection in the object tree.
2. Click **OK**.

   The nodes that were included among the search results appear selected in the object tree.

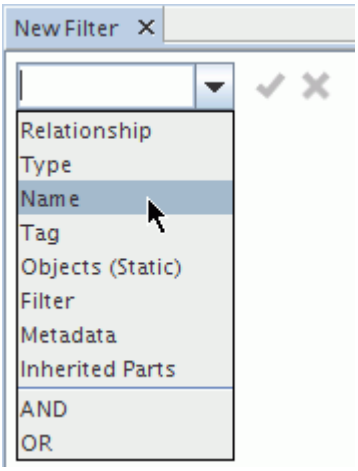
Filter Dialog Reference

The filter dialog contains editors and tools for creating predicates, working with compound predicates, and modifying predicates.

Editors of Predicate Components

Three editors exist for creating or modifying a predicate. For details pertaining to specific predicate types, see Predicate Reference.

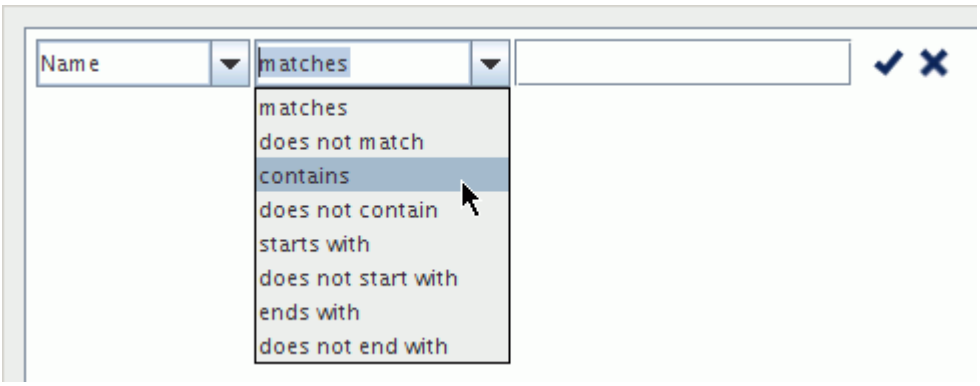
- The type editor lets you select a type from the drop-down list.



You can also type into the text box of the drop-down list.

To select the desired entry, navigate to it with the `<Up>` or `<Down>` arrow keys and press `<Enter>`.

- The operator editor becomes available after you select a type.



The set of available operators varies with the type.

Note: Operators that exclude items from a search, such as `is not` and `does not match`, are the equivalent of the search term NOT.

- The value editor lets you specify what to search for. The value editor varies in its form, depending on the type that you select.

Compound Predicate Editors

Compound predicate editors are accessible as soon as you begin to create a filter:

- **AND** -- objects matching all of the contained values
- **OR** -- objects matching any of the contained values

To access the compound predicate editor when a complete predicate exists, click (**Add**). The compound predicate editor appears near the top of the view as an **AND**.

You can change **AND** to **OR** using a drop-down list. To activate the drop-down list, click **AND** or click (**Edit**).

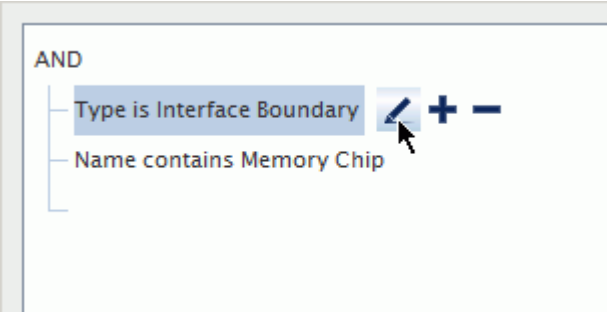
You can also use the following techniques for combining predicates into a compound one, or transferring a predicate from one compound predicate to another:

- Drag-and-drop
- Using the (**Cut**) and (**Paste**) tools

Query Editing Tools

The following table lists the tools that are available, depending on the situation, in the filter dialog and/or in a context menu when you right-click in the view.

Some of these tools "float" with your mouse when you hover over a predicate or a compound predicate editor.



(**Edit**), also `<Space>`	Accesses a predicate for modification.
(**Add**), also `<Insert>`	Creates a predicate.
(**Delete**), also `<Delete>`	Deletes a predicate.
(**Undo**), also `<Ctrl><Z>`	Reverses the latest change.
(**Redo**), also `<Ctrl><Y>` or `<Ctrl><Shift><Z>`	Repeats a change that was subjected to the Undo command.
(**Cut**), also `<Ctrl><X>`	Removes the selected object and stores it on the clipboard.
(**Copy**), also `<Ctrl><C>`	Stores a copy of the selected object on the clipboard.
(**Paste**), also `<Ctrl><V>`	Places the contents of the clipboard into the active area.

(**Display query results as a tree**)	Lets you view search results in a hierarchical node tree. The tree view resembles what you see in a simulation window.
(**Display query results as a list**)	Lets you view search results in a list (default setting).
(**Commit changes**)	Completes the specification of a predicate.
(**Discard changes**)	Cancels the specification of a predicate.

Predicate Reference

Predicates have a wide variety of types, operators, and values to help you create a precise search in STAR-CCM+.

Note: Only objects that you can rename are included in the search results. For example, a search for `Plot` yields individual plots, but not the **Plots** manager object.

Predicate Type	Available Operators	Effect
Relationship	below in tree	Matches all objects that are contained within the selected object (that is, sub-objects, sub-objects of sub-objects, and so on).
	exclude children of	Disregards all objects that are contained within the selected object (that is, sub-objects, sub-objects of sub-objects, and so on).
	above in tree	For the selected object, matches the hierarchy of parent objects (that is, the parent object, the parent of the parent object, and so on).
	exclude parents of	Disregards all objects that are above the selected object (that is, the parent object, the parent of the parent object, and so on).
	directly below in tree	Matches the immediate sub-objects of the selected object.
	directly above in tree	Matches the immediate parent object of the selected object.
	referencing	Matches all objects that are referencing the selected object.
	not referencing	Matches all objects other than the ones that are referencing the selected object.
	referenced by	Matches all objects that are referenced by the selected object.

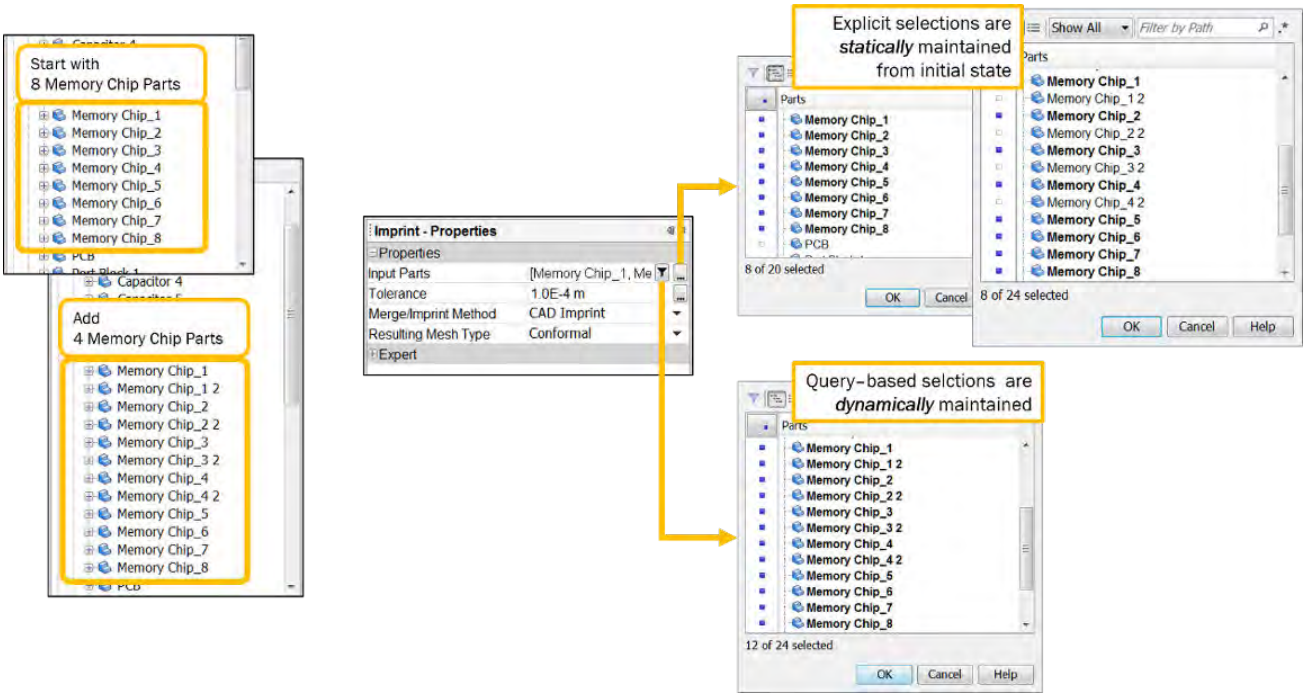
Predicate Type	Available Operators	Effect
	not referenced by	Matches all objects other than the ones that are referenced by the selected object.
Type	is	Matches the selected object type. The available object types are: • Geometry Part • Part Surface • Part Curve • Part Surface Contact • Region • Boundary • Feature Curve • Interface Boundary • Derived Part • Report • Monitor • Plot • Coordinate System • Field Function • Units • Composite Part • Mesh Part
	is not	Matches all objects other than the selected object type.
Name	contains	Matches all objects with names that include the specified text string.
	does not contain	Matches all objects with names that do not include the specified text string.
	matches	Matches all objects with the specified name.
	does not match	Matches all objects with names other than the specified name.
	starts with	Matches all objects with names that begin with the specified text string.
	does not start with	Matches all objects with names that do not begin with the specified text string.
	ends with	Matches all objects with names that end with the specified text string.

Predicate Type	Available Operators	Effect
		Note: A name can include a trailing number which is considered part of the name, such as **Cartesian 1**. Therefore, for example, an entry of `ian` does not find the **Cartesian 1** node.
	does not end with	Matches all objects with names that do not end with the specified text string.
Tag For more information, see Tagging Simulation Objects.	contains	Matches all objects that are tagged with the selected tag.
	does not contain	Matches all objects that are not tagged with the selected tag.
Objects (Static) **Note:** This type makes a static query. Use it to include or exclude specific objects explicitly.	are	Matches the selected objects.
	are not	Matches all objects other than the selected objects.
Filter Use this type to reuse portions of a filter without duplicating predicates.	matches	Matches all objects that match the referenced filter.
	does not match	Matches all objects that do not match the referenced filter.
Metadata -- for more information, see What Is Metadata?. The values are two text components: key and value. The key is always a string, but the value can be either a string or a number.	contains	Matches all objects that contain the specified metadata. If you enter a numeric value, it only finds objects with that value as a number. If you enter a number in single quotes, it only finds objects with that value as a string of text.
	does not contain	Matches all objects that do not contain the specified metadata.
Inherited Parts Searches for parts that are inherited from geometry objects: parts, composites, part surfaces, part curves, and part points. For more information, see Inherited Parts.	derive from	Matches all part surfaces, part curves and part point objects below the selected objects in the tree, either the actual objects themselves or their occurrence under the Inherited Parts folder of another part. Only the leaf level objects at the bottom of the tree are selected. Intermediate objects, such as parts composites, are not selected.
	do not derive from	Matches all objects other than the selected inherited parts.

Using Query-Based Selection

For properties that require a selection of objects, you can apply dynamic queries that use filter criteria to define which objects are included in the selection. STAR-CCM+ automatically evaluates the selection when objects are added, removed, or modified.

A dynamic query uses the same set of predicates as filters. However, when you use a filter for selecting objects, the list of objects defined by the filter is a static list. STAR-CCM+ does not subsequently update the list of objects in the property. By comparison, a dynamic query defines a list of objects that change according to whether objects in the simulation meet the criteria or not. This difference is illustrated in the following diagram.



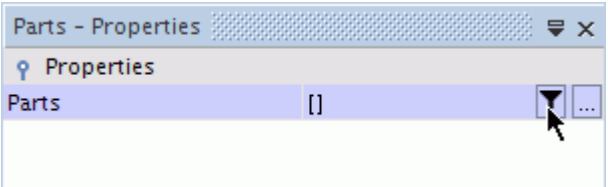
In the current version of STAR-CCM+, dynamic queries are available with the following items:

- input parts for derived parts, mesh operations, and reports
- parts on regions
- part surfaces on boundaries
- part curves on feature curves
- contacts on an interface
- parts on scene displayers
- objects on groupings

Activating a Dynamic Query

To activate a dynamic query:

1. Click the query editor button ( ) in the right half of the property (typically the **Parts** property).



2. Define the query for selecting objects. The techniques for defining this query are the same as those for a filter. For details, see Using Filters.

Note:

- If your object (scene, operation, or derived part) is using a dynamic query, changing the selection through the standard object selector causes the dynamic query to be eliminated and replaced by a static list.
- In a scene that uses dynamic queries, when you manually remove a part from the scene, those parts return to the scene as soon as you interact with the dynamic query.

Prioritizing Assignment of Parts

When you use dynamic query selection to assign parts across a number of similar objects, you can find that an additional level of prioritization is necessary. STAR-CCM+ allows you to define the order of priority in which objects receive parts through dynamic query selection. Objects that come higher in the list receive parts in preference to objects that come after them.

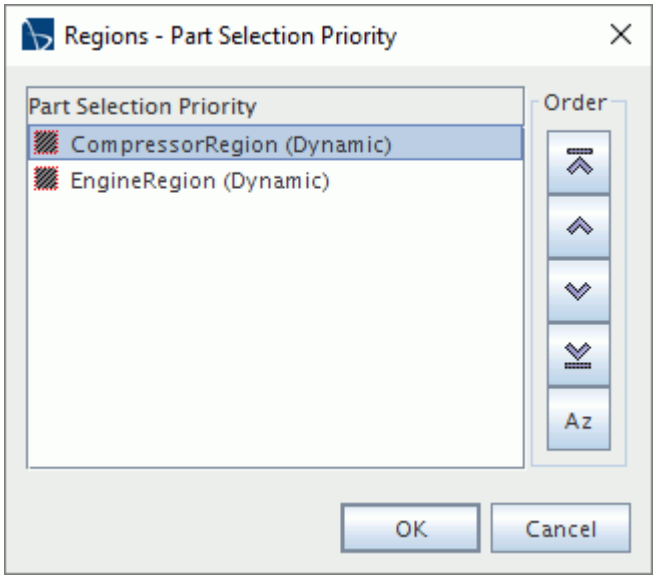
The types of objects to which you can prioritize assignment are regions, boundaries, feature curves, subgroupings (on regions and boundaries), and boundary interfaces. For these types of objects, the manager node that contains them exposes a property, *Part Selection Priority*, that contains the priority list. The order in which objects appear on the priority list sets the order in which STAR-CCM+ assigns parts to those object properties that require them—when dynamic queries are used in place of a static list of parts.

For example, suppose that you have two regions existing within the **Regions** node, **EngineRegion** and **CompressorRegion**. You require that **EngineRegion** is automatically assigned all parts that have `Engine` in the name, and that **CompressorRegion** is automatically assigned all parts with `Compressor` in the name. For the *Parts* property of the **EngineRegion**, you define the dynamic query, `Name contains Engine`, and for the *Parts* property of **CompressorRegion**, you define the dynamic query, `Name contains Compressor`.

These queries successfully assign parts such as **Compressor-Duct-1**, **Compressor-Chamber-1**, **Engine-Port-1**, **Engine-Port-2**, and so on. However, if you then add a part, **Engine-Compressor-Inlet1**, you cannot tell in advance whether **EngineRegion** or **CompressorRegion** receives the part.

To make sure that such a part gets assigned to **CompressorRegion**, use the following technique:

1. Select the **Regions** manager node.
2. In the *Part Selection Priority* property, click ... (**Custom Editor**).
3. In the *Regions - Part Selection Priority* dialog, specify the order of priority. In this example, you make sure that **CompressorRegion** appears at the top of the list. .



Use any of the buttons (top to bottom) as follows:

- **Move to top**: move the region to the top of the list
- **Move up**: move the region up one position
- **Move down**: move the region down one position
- **Move to bottom**: move the region to the bottom of the list
- **Alphabetize entire list**: reorder the items alphabetically

4. Click **OK**.

Now, when STAR-CCM+ detects that the dynamic queries for both **CompressorRegion** and **EngineRegion** can accept the part, **Engine-Compressor-Inlet1**, STAR-CCM+ gives priority to **CompressorRegion** over **EngineRegion**.

Consider the following as you work with these objects:

- Regions with statically assigned parts (that is, using the standard technique for manually assigning parts to regions) will take priority over regions with dynamically assigned parts, regardless of the order specified in the part selection priority dialog. They appear in the dialog to allow setting priority before or after you assign queries. However, once you assign a part statically, the only way to remove it is to deselect it manually or assign it to another region.
- Boundaries, boundary interfaces, and feature curves work the same way as regions. However, since only boundary interfaces are assigned part contacts, non-boundary interfaces do not get listed in the part selection priority dialog.

Global Parameters

A global parameter is a scalar or vector physical quantity (for example, 120.0 K or [0.0, 1.0, 0.0] m) that can be defined once and used in multiple objects.

One application of this feature is using a key physical characteristic as a reference value. Other values can point directly to this reference value or be defined in terms of it. For example, in a boat simulation, the hull length could be used (directly or multiplied by a scale factor) in numerous other nodes.

Similarly, you can define a global constant like the mathematical value pi (3.14...). By defining a global parameter (for example, `${PI_CONSTANT}`), you can use it repeatedly without entering the value again. This technique lets you keep the setting consistent; you do not have 3.14 specified for one object and 3.14159 for another. Because this parameter value is not defined on every cell, it can save significant system resources.

Another important use is design optimization – think of the global parameters as controls that can be conveniently adjusted over the course of multiple iterations of a simulation.

Note: Only SI units are available for this feature.

Note: Simulation parameter expressions can only reference other simulation parameters. References to reports or field functions are not allowed.

To set up a global parameter:

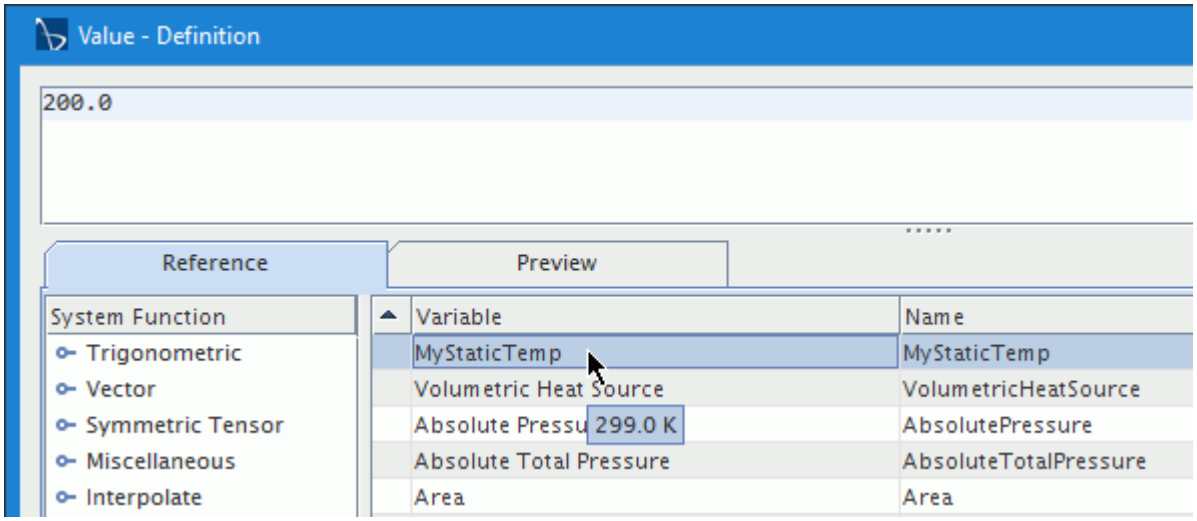
1. Right-click the **Tools** > **Parameters** node and select **Scalar** or **Vector** from the **New** submenu.
2. Rename the new node based on the type of quantity, for example `MyStaticTemp` for a scalar parameter.
3. Set a value in the *Dimensions* property. Click ... (**Custom Editor**) and enter values in the dialog. These values are exponents.

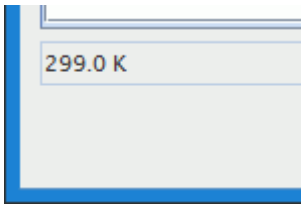
   For example, in the *Scalar - Dimensions* dialog for a scalar parameter, an entry of `1` for *Temperature* gives you the units K. An entry of `2` for *Length* gives you the units m^2.

To apply a global parameter, follow a procedure similar to the example outlined below:

1. Select a node that stores a scalar physical quantity, such as **Regions** > **Air** > **Air.Inlet 1** > **Physics Values** > **Static Temperature**.
2. Click ... (**Custom Editor**) next to the *Value* property.
3. In the dialog that appears, for example *Static Temperature - Value*, click ... (**Custom Editor**) next to the *Definition* property.
4. In the definition dialog, apply the global parameter, which is listed in the *Variable* column:

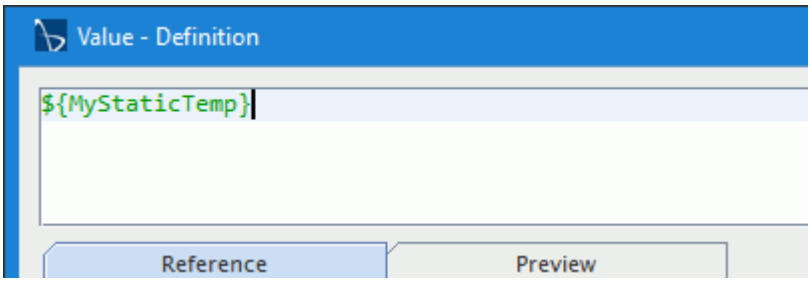
When you hover over the parameter with the mouse pointer, the value of the parameter appears in a tooltip and in the bottom left corner of the dialog.





a. Clear the existing definition from the expression text panel near the top of the definition dialog.

b. Double-click the parameter. Its definition appears in the expression text panel as `${MyStaticTemp}`.



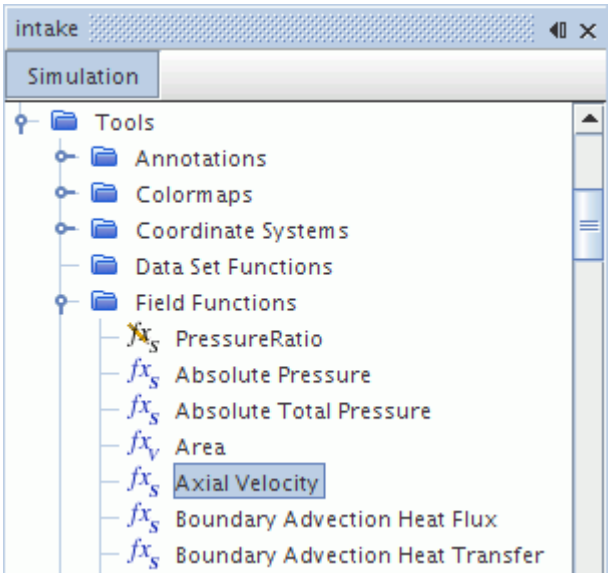
c. Click **OK** in the definition dialog and then in the value input dialog.

The definition of the parameter appears in the property of the object, such as the *Value* property of the **Static Temperature** node.

Field Functions

Field functions allow you to access fields (scalar or vector data that are evaluated at cells, vertices, or boundary faces) in STAR-CCM+. You can use field functions to visualize the computed fields, to specify boundary and region values, or to define initial conditions.

Field functions are represented by nodes under the **Tools** > **Field Functions** node. The **Field Functions** manager node can hold any number of field functions up to the memory limits of the computer.



The letter in the lower right corner of the field function icon indicates the type of field function: Scalar (S), Vector (V), Array (A), Position (P), or Tensor (T). See Field Function Types.

STAR-CCM+ provides system-defined field functions, that are associated with physics models, and user-defined field functions, that you completely define using expressions. When you activate a physics model, STAR-CCM+ automatically adds the predefined field functions for that model. For example, when you activate the Solid Stress model, STAR-CCM+ automatically activates the system field functions that are relevant to Solid Stress, like the Stress Tensor and Displacement field functions.

The most common use of field functions is examining the solution. You can, for example, display pressure on the wall of a body by using the system field function, Pressure, in a scene. A more advanced use of field functions is the specification of boundary and region values as well as initial conditions. In this case, you typically define a new field function and apply it. An example would be writing a function for a parabolic velocity profile that is a function of the y-coordinate and using it for the inlet velocity at a boundary.

Once the field function has been made available, it is provided in scalar and/or vector drop-down lists for any operation that requires it: notably visualization, XY plotting, reporting, monitoring, and creating table data. In addition, the following objects are typically included in these lists:

- Field function monitors
- Field functions corresponding to reports
- Cell sets

A field function remains active in the GUI as long as any other object is still accessing it. When objects such as reports, monitors, or displayers access field functions, that connection forces the field function to remain present in the simulation. For system field functions, this is true even after you deactivate the related physics model. Therefore, when you are changing continuum models and/or choosing to delete a user field function, take care to examine your simulation for objects that keep an obsolete field function active. Failure to do so can lead to the inability to delete user field functions, as well as confusion as to which visible field functions are actually being calculated.

Field functions can only be accessed on a part if the data used to define the field function exists on that part. For the data to exist it must be allocated on the specific part and must have been initialized. For example, wall shear stress is a quantity that is stored only on walls in viscous-flow simulations. Therefore the wall shear stress field function, and related field function quantities such as Wall y+, can only be displayed on wall boundaries, and only after the solution has been initialized.

Contents:

Field Function Types
Creating a User Field Function
Copying User Field Functions and User Field Function Properties
Ignoring Boundary Values
Interpolating Field Functions Using Inverse Distance Weighting
User Field Functions Syntax
User Field Function Examples
Field Functions Reference

Field Function Types

Scalar Field Functions

Scalar field functions are field functions that correspond to just one number (zero, positive or negative) at a specific location.

Vector Field Functions

Vector field functions define a vector that can be expressed in different coordinate systems.

They have the following components when presented in the scalar selection list, depending on the coordinate system with which they are associated:

- `Magnitude`
- `Cartesian` or `Laboratory`:
  - `i`
  - `j`
  - `k`
- `Cylindrical`:
  - `Radial`
  - `Tangential`
  - `Axial`
- `Spherical`:
  - `Radial`
  - `Polar`
  - `Azimuthal`

Array Field Functions

Array field functions are field functions that are defined as three numbers at each location.

No coordinate system transformations are possible. They have the following components when presented in the scalar selection list:

- `Magnitude`
- `[X-component value]`
- `[Y-component value]`
- `[Z-component value]`

Position Field Functions

Position field functions are field functions that are defined as a position at each location.

They can be transformed into different coordinate systems using position transformations. They have the following components when presented in the scalar selection list, depending on the coordinate system with which they are associated:

- `Magnitude`
- `Cartesian` or `Laboratory`:
  - `X`
  - `Y`
  - `Z`
- `Cylindrical`:
  - `r`
  - `theta`
  - `z`
- `Spherical`:
  - `r`
  - `theta`
  - `phi`

Symmetric Tensor Field Functions

See Symmetric Tensor Field Functions.

Contents:

Symmetric Tensor Field Functions

Symmetric Tensor Field Functions

STAR-CCM+ provides support for symmetric tensor-valued field functions. These functions are stored internally as six floating point values for a 3 x 3 symmetric matrix.

Symmetric tensor-valued field functions are currently only produced by the finite-element solid stress module, but they can be constructed and used for general calculations. Similar to vector-valued field functions, you can access individual components of symmetric tensors, perform coordinate transformations on them, and compute common algebraic operations on them.

Construction

In the field function expression editor, a symmetric tensor-valued variable must be prefixed with `$$$`.

Symmetric tensor variables are constructed in the following way:

Operation	Expression Syntax	Result
3 x 3, lower-triangular specification	`[1 ; 2,3 ; 4,5,6]`	$\mathbf{A} = \begin{bmatrix} 1 & 2 & 4 \\ 2 & 3 & 5 \\ 4 & 5 & 6 \end{bmatrix}$
3 x 3, upper-triangular specification	`[1,2,3 ; 4,5 ; 6]`	$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}$

Note: Symmetric tensors support all basic arithmetic operations, except for multiplication of two symmetric tensors. Multiplication of two symmetric tensors can result in a non-symmetric tensor and so the current version of STAR-CCM+ does not support this operation.

Coordinate System Transformations

Symmetric tensors are transformed into a local coordinate system through *similarity transformation*. For example, the user-defined field function expression

```
$$$A( @CoordinateSystem("Laboratory.Sph1") )
```

transforms the symmetric tensor $\mathbf{A}$ into the local spherical coordinate system **Sph1**, whose parent is the Laboratory coordinate system. Let $\mathbf{R}$ denote the 3 x 3 orthogonal matrix which transforms a vector $\mathbf{v}_s$ from **Sph1** into Laboratory coordinates. Then the mathematical form of this transformation is:

$$\mathbf{A}_s = \mathbf{R}^T\mathbf{A}\mathbf{R} \tag{6}$$

The columns of $\mathbf{R}$ are the (position-dependent) local basis in **Sph1**.

The components of $\mathbf{A}_s$ are now with respect to the local basis at that particular location within **Sph1**. (Recall that the unit basis vectors of a local curvilinear coordinate system depend on the coordinate within that local frame.) The matrix-vector product of $\mathbf{A}_s$ with $\mathbf{v}_s$,

$$\mathbf{A}_s\mathbf{v}_s = \mathbf{R}^T\mathbf{A}\mathbf{R}\mathbf{v}_s \tag{7}$$

is therefore equivalent to transforming $\mathbf{v}_s$ into the Laboratory frame, applying $\mathbf{A}$, and then transforming the result back into **Sph1**. In STAR-CCM+, it is possible to verify that for any symmetric tensor $\mathbf{A}$ and any vector $\mathbf{v}$ (both in the Laboratory frame), the expression

```
$$$A(@CoordinateSystem("Laboratory.Sph1")) *
$$v(@CoordinateSystem("Laboratory.Sph1"))
```

produces values that are identical to the ones that are produced by

```
($$$A*$$v)(@CoordinateSystem("Laboratory.Sph1"))
```

since

$$\left(\mathbf{R}^T\mathbf{A}\mathbf{R}\right)\left(\mathbf{R}^T\mathbf{v}\right) = \mathbf{R}^T(\mathbf{A}\mathbf{v}) \tag{8}$$

Coordinate System Components

When selecting a component of a symmetric tensor in each coordinate system, the following labels correspond to the appropriate matrix entry:

Coordinate System Type	Tensor Component
Laboratory	$ii = a_{00}$ $ij = a_{01}$ $ik = a_{02}$ $jj = a_{11}$ $jk = a_{12}$ $kk = a_{22}$

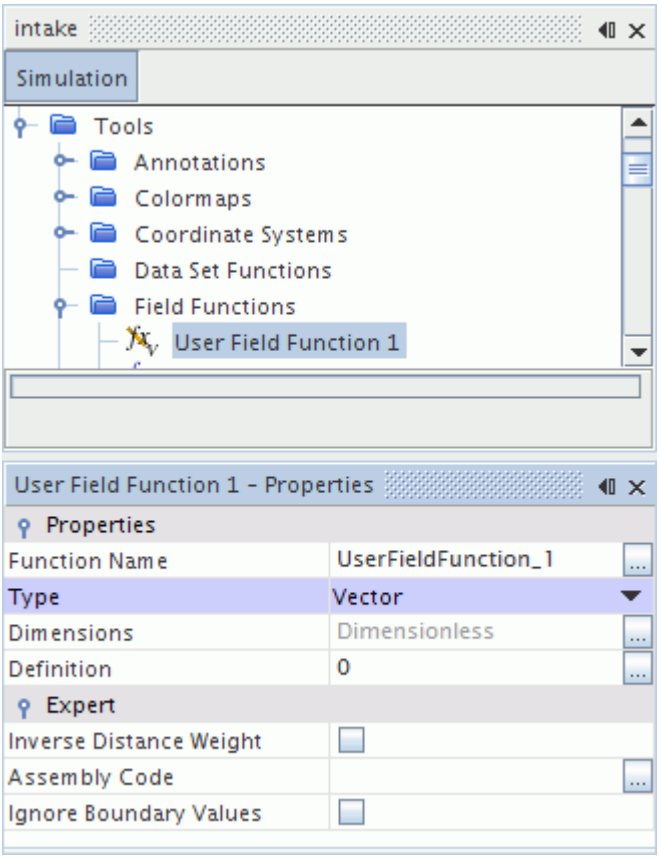
Coordinate System Type	Tensor Component
Local Cartesian	$ii = a_{00}$ $ij = a_{01}$ $ik = a_{02}$ $jj = a_{11}$ $jk = a_{12}$ $kk = a_{22}$
Cylindrical	$rr = a_{00}$ $r\theta = a_{01}$ $rz = a_{02}$ $\theta\theta = a_{11}$ $\theta z = a_{12}$ $zz = a_{22}$
Spherical	$rr = a_{00}$ $r\theta = a_{01}$ $r\phi = a_{02}$ $\theta\theta = a_{11}$ $\theta\phi = a_{12}$ $\phi\phi = a_{22}$

Creating a User Field Function

To create a user field function:

1. Right-click the **Tools** > **Field Functions** node and select **New** followed by the type of field function you wish to create:
   - **Scalar**
   - **Array**
   - **Vector**
   - **Position**
   - **Symmetric Tensor**

   A new field function appears in the object tree. The type that you selected in the context menu is displayed in the *Type* property of the field function.



Initially the node has a default name such as **User Field Function 1**. However, you can rename the field function.

A new field function also appears automatically with the creation of a report. You cannot rename such field functions directly; their names contain the names of their respective reports.

Using Programming Resources

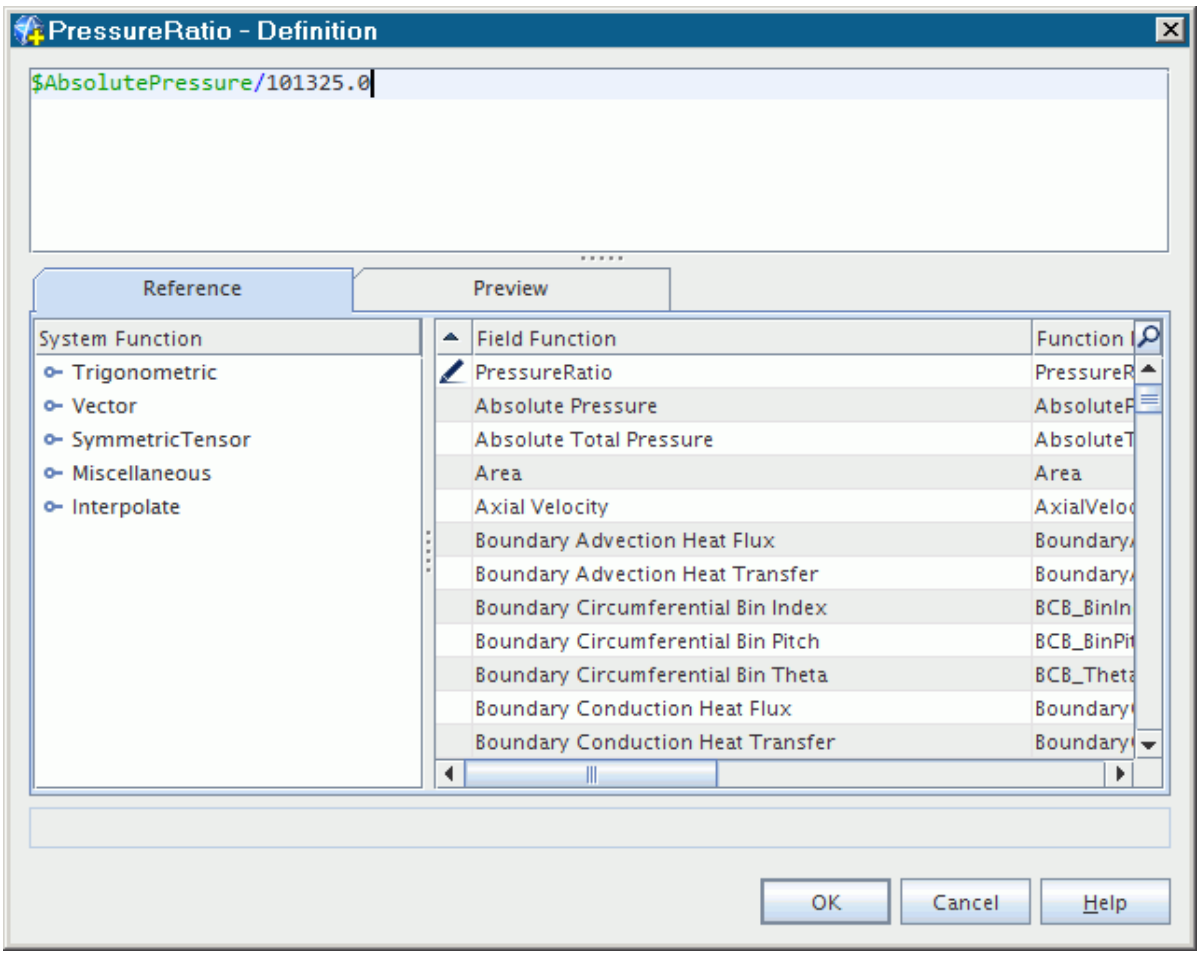
STAR-CCM+ offers a wide range of programming resources with which you can define your field functions. See the following sections for details:

- User Field Functions Syntax -- This reference section contains a comprehensive set of "building blocks" (operators and functions) for your custom field functions, as well as techniques for making them achieve your purpose. It includes:
  - Working with Table Values Through Field Functions -- This subsection provides the technical specs for interpolating values by accessing data from tables.
- User Field Function Examples -- Here you can refer to examples of common user field functions. These examples could include the very field function that you wish to create.

1. To define the user field function, do either of the following:
   - In the *Definition* expert property, enter the definition directly.
   - For longer definitions, use the definition dialog.

     Click ... (**property customizer**) to activate the dialog.

     An example definition, using the dialog, is shown below.



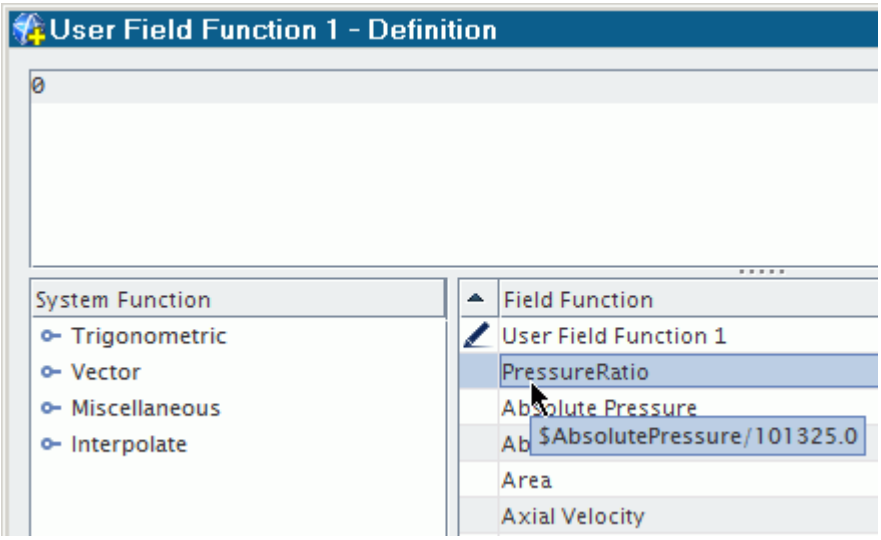
2.

3. For longer definitions, use the definition dialog.

Click ... (**property customizer**) to activate the dialog.

In the bottom right panel, a pencil icon appears alongside the name of the field function that you are defining.

4. To see the definition of another user field function in the simulation, hover over it with the pointer. The definition appears in a tooltip.



5. In the expression text panel at the top of the dialog, define the field function using any combination of the following techniques:
   - Typing the definition directly. As you type, candidate functions that start with your string appear in a pop-up list. The pop-up list includes system-provided field functions such as `LocalCellIndex`, user-defined field functions, as well as system functions such as `floor()`.

     To make your selection, use the `<Up>` or `<Down>` arrow keys and press `<Enter>`. Alternatively, double-click the desired entry.

     Note: This auto-complete feature does not recognize the following characters:

     ```
     ~ ` ! @ # $ % ^ & * ( ) - = + { [ } ] : ; " ' | \ < ,
     > . ? /
     ```

   - Inserting expressions interactively from the two bottom panels.
   - Using common editing techniques, such as copying and pasting:
     - Apply standard keyboard commands such as `<Ctrl><Z>` (Undo), `<Ctrl><Y>` (Redo, also with `<Ctrl><Shift><Z>`), and `<Ctrl><C>` (Copy).
     - To access the commands from a context menu, right-click in the expression text panel.

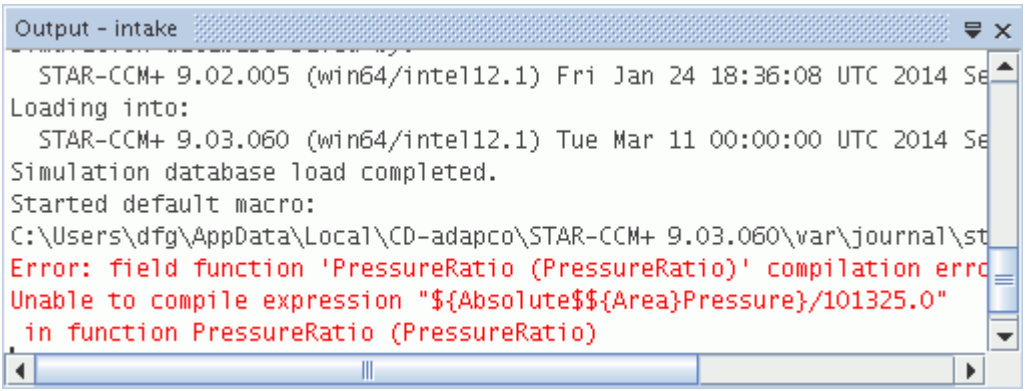
       Note: The syntax that is used in the definition for user field functions is a subset of the C programming language.

Space between tokens is not significant, so you can use spacing to make the definition easier to read. You can even include line breaks. If you indent on one line, subsequent lines are automatically indented. To maintain readability of deeply nested parenthetical expressions, use this indenting capability.

The expression text panel uses colors to help guide you through the editing process:

- Green indicates a syntactically correct reference to an existing field function:
- Parentheses appear in yellow to indicate the matching parenthesis:

  This color technique also applies to square brackets.
- Red indicates an incorrect entry: However, if a field function reference has red text (but no black text or red background) the color represents a syntactically correct reference to a field function that does not yet exist.
- If you click **OK** when the definition includes an error, dialogs appear with error messages. The messages are repeated in the *Output* window.



When such an error occurs, examine and debug the definition until it is accepted without any error notifications.

Note: Field function definitions are always done in SI units, independent of the units that you use for input or display.

Contents:

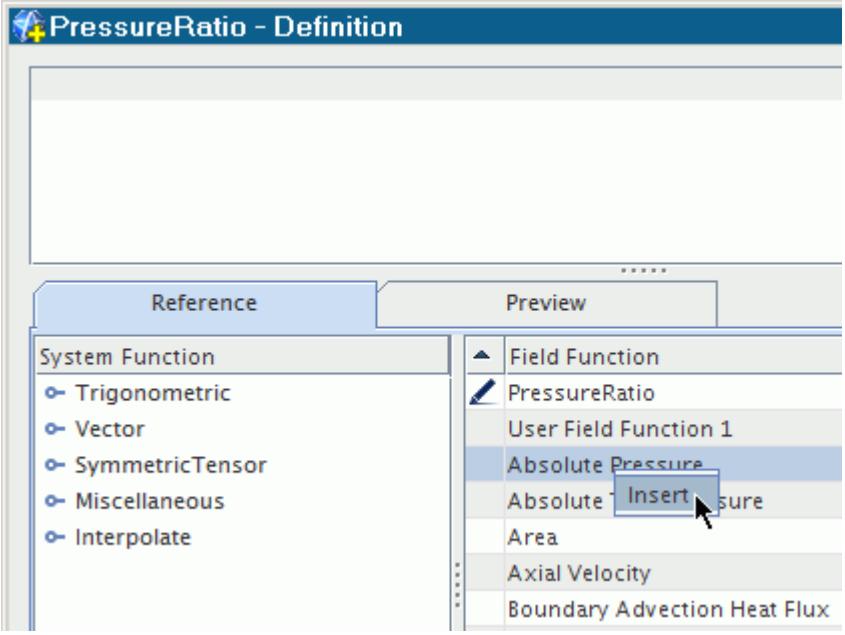
Inserting Expressions Interactively
Previewing Expressions

Inserting Expressions Interactively

To add complete expressions (field functions or system functions) into the expression text panel, make selections from the two bottom panels of the dialog.

To insert a field function expression:

- Right-click on one of the names in the bottom right panel and select **Insert**.



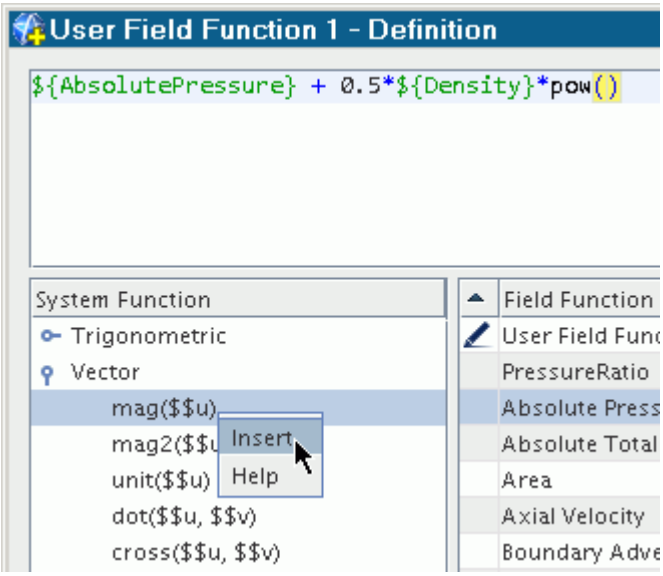
- Alternatively you can double-click the row, or highlight the row and press the spacebar.

  When you insert a field function, the field function reference replaces any selected text in the expression text panel.

  For details on including field functions within a field function, see Referencing Field Functions.

To insert a system function expression:

- Open any of the objects with the handle ( ) in the bottom left panel. Right-click one of the child objects and select **Insert**.



- Alternatively you can double-click the row, or highlight the row and press the spacebar.

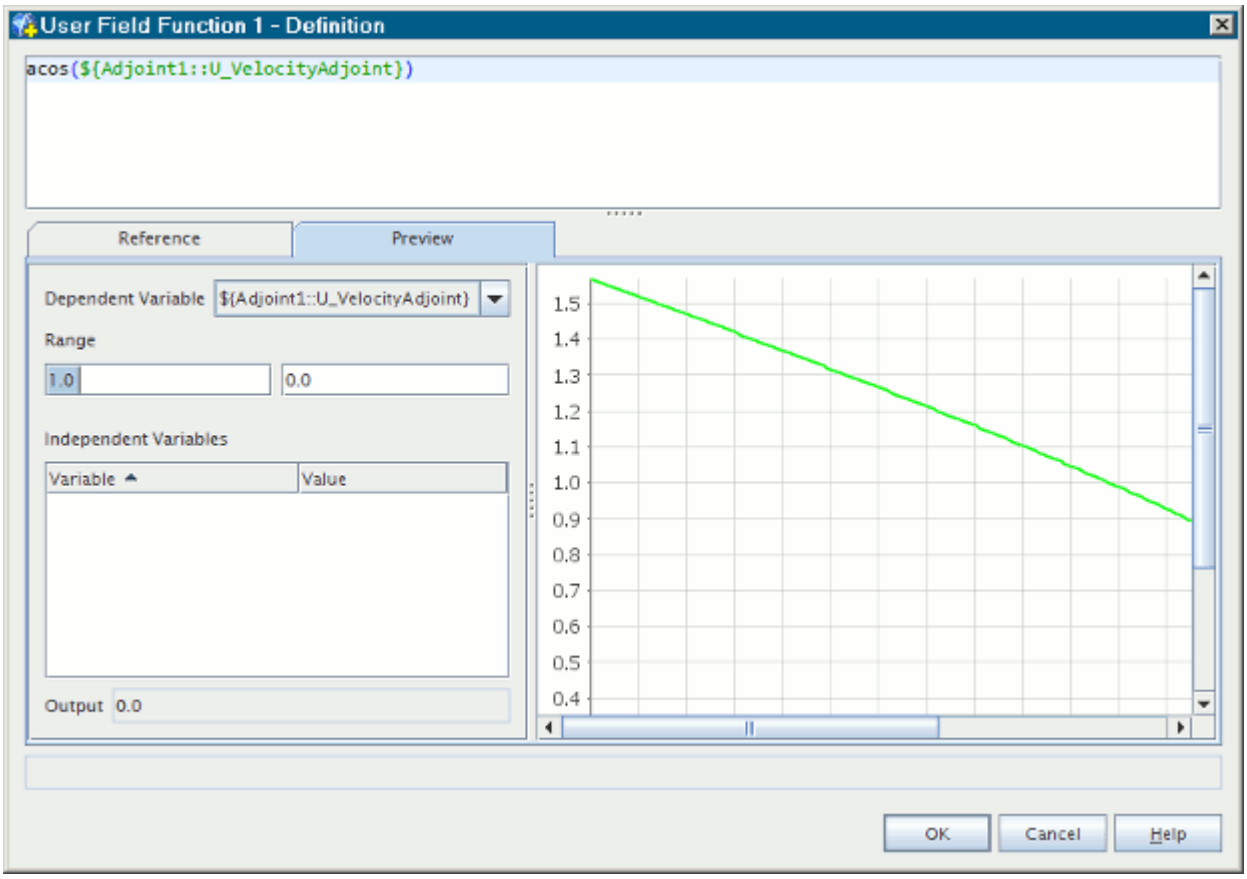
When you insert a system function, any selected text becomes enclosed by the function. For example, if you have `$$Velocity` selected and you insert the system function `mag($$u)`, `mag($$Velocity)` appears

in the expression text panel. If there is no selection, the system function gets inserted at the cursor location with no parameters, for example `mag()`.

Previewing Expressions

You can preview the calculations of your user field functions without actually running the simulation. Scalar values appear in a plot which gets updated instantly with your changes. The output plot is calculated by substituting the values of each of the independent variables and sweeping through the range of values that are provided for the dependent variable.

An example of a plot preview is shown below. In this example, the output is `acos(1) = 0`.

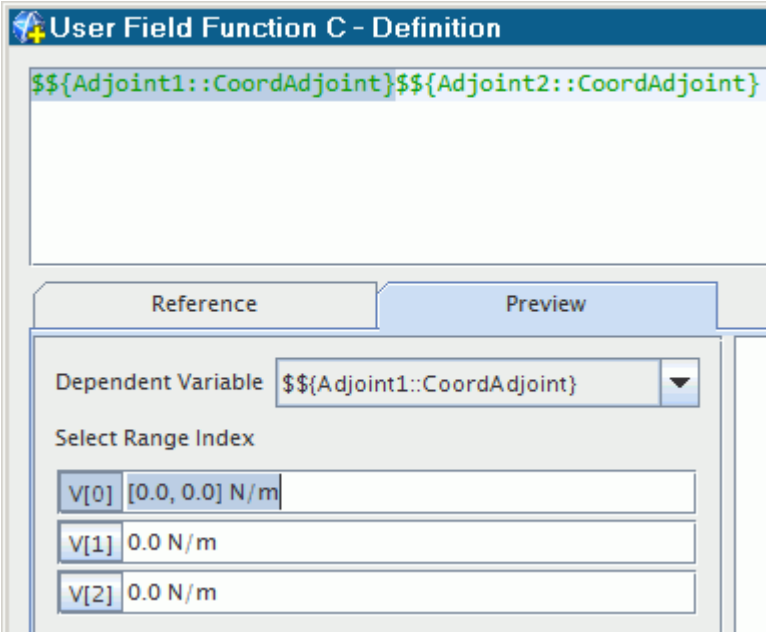


Note: Plot previews are available for scalar functions only.

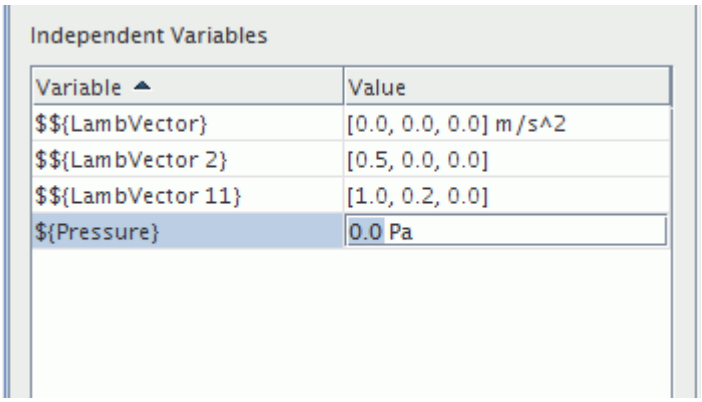
To preview a user field function:

1. In the definition dialog, click the *Preview* tab.
2. Select the *Dependent Variable* and enter the *Range and Values* to use in the preview. As you select the dependent variable, STAR-CCM+ highlights the corresponding text in the definition window.

   For a vector component, click the label of the component whose range you wish to set; the right column value becomes available for that component. You can plot a range for only one of the vector components at a time.



3. Within the table of *Independent Variables*, enter values for the field function to use in generating the preview plot.



Note: If you make errors, such as dividing by zero, STAR-CCM+ informs you by stating the nature of the error in the status field at the bottom of the definition dialog.

Copying User Field Functions and User Field Function Properties

There are several ways to copy a user field function:

- Copy the user field function to the clipboard and paste it into the **Field Functions** node.
- Drag and drop the user field function onto the **Field Functions** node.
- Copy the properties of the user field function onto another.

See Copying and Pasting Objects.

When properties are copied from one user field function to another, the following properties and expert properties are changed:

- *Dimensions*
- *Assembly code*
- *Definition*
- *Ignore Boundary Values*

Change the *Function Name* expert property by entering text or renaming it.

It is not possible to copy properties between other types of field functions and user field functions.

Ignoring Boundary Values

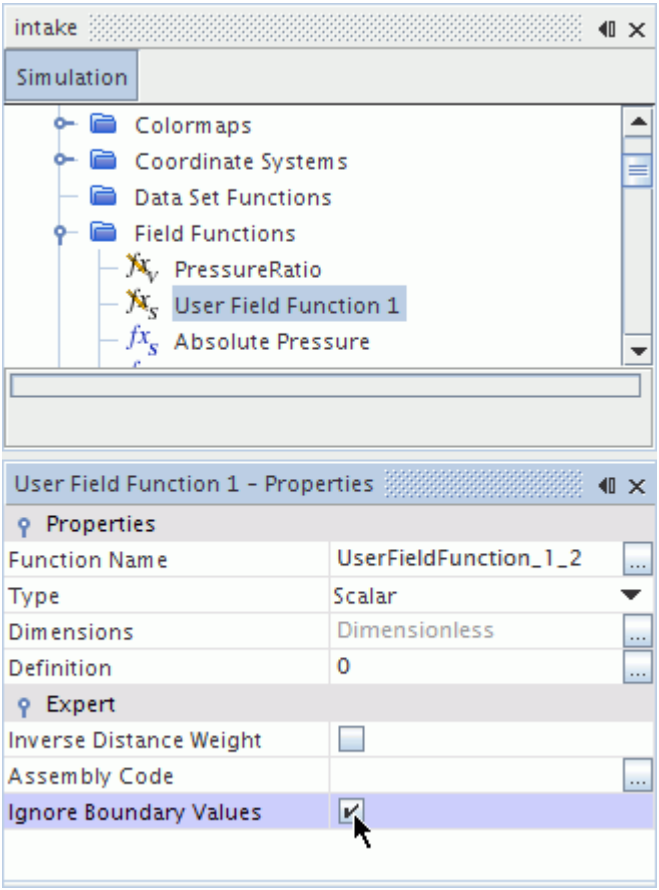
Field functions access data on specific parts that are defined in the simulation. When field functions are accessed on boundary parts, two rules are used by default:

- If the data is stored on the boundary, the boundary data are accessed.
- If the data is not stored on the boundary, the cell data from the cells in the adjacent region are accessed.

In specific situations where data is stored on the boundary and in the adjacent cells, access the adjacent cell data:

1. Create a user field function. See Creating a User Field Function.
2. Define the new user field function such that it is defined in terms of an existing field function that is stored in both cells and on boundaries, but the values in each location differ.

   An example is velocity magnitude, which has the wall velocity values stored in wall boundaries, and the wall-adjacent fluid velocity in the cells. In this case, the field function is defined as: `mag($${Velocity}).`
3. Activate *Ignore Boundary Values* in the *Expert* category.



If this user field function is accessed on a boundary, it contains a scalar value corresponding to the velocity magnitude in the cell that is next to the boundary.

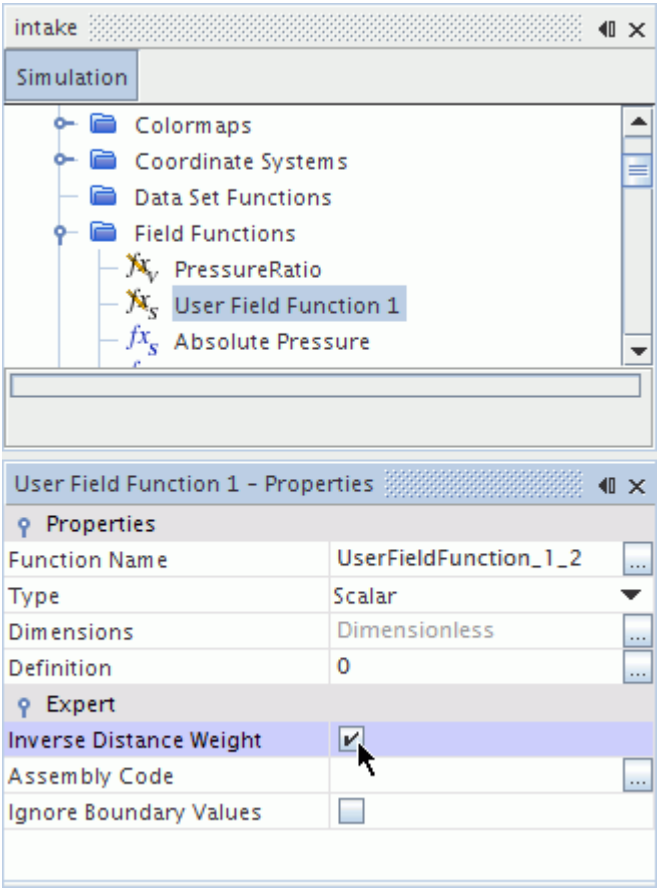
If the field function is applied on an interface, and the *Ignore Boundary Values* option is activated, the field function is computed using data from the cell next to the boundary. This data is identified using the *Boundary-0* property of the interface.

Interpolating Field Functions Using Inverse Distance Weighting

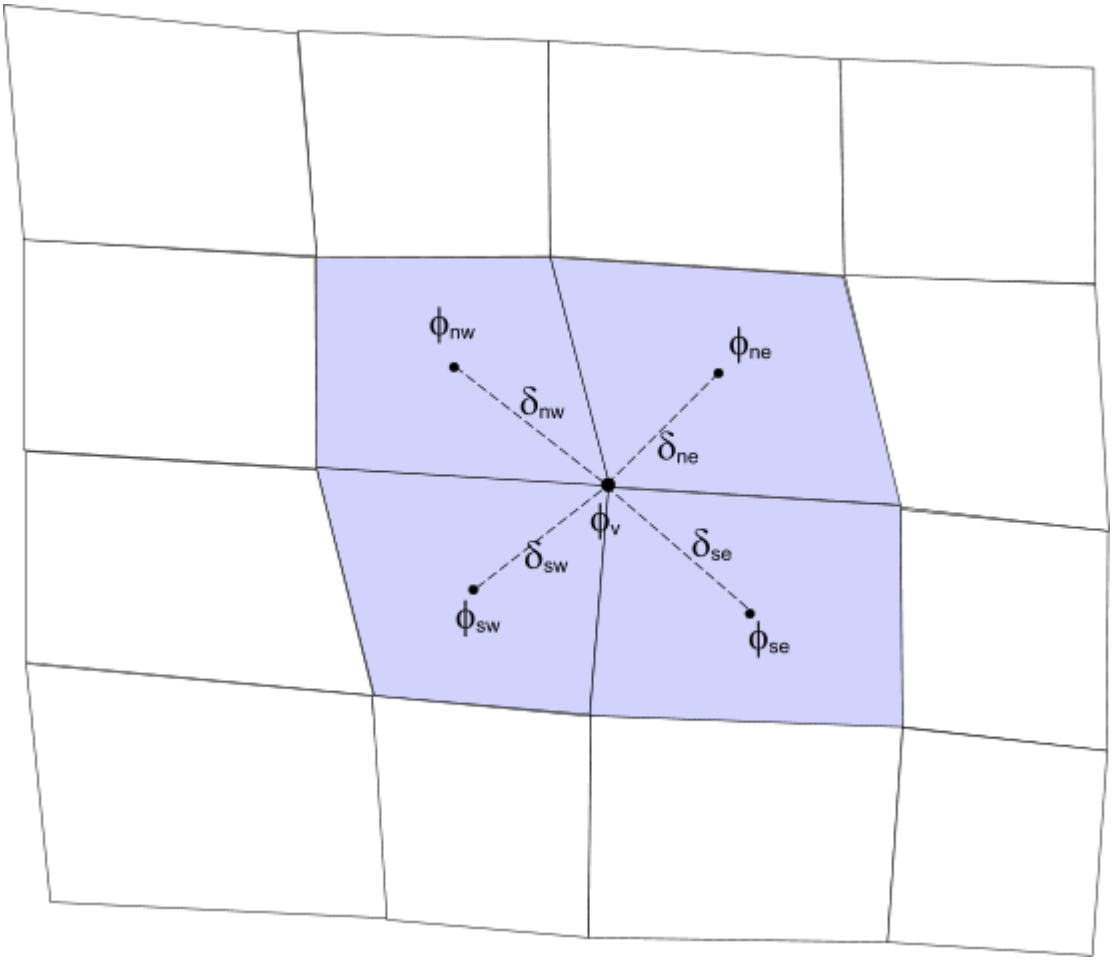
In STAR-CCM+, most variables are solved and stored at cell centers. To smooth these values, STAR-CCM+ uses a first order method by default to interpolate the data that are stored in cell centers to vertices. When using a field function in a derived part, the value of the field function at the extraction points is interpolated from these vertex values.

You can interpolate field functions from cell centers to vertices:

1. Create a user field function. See Creating a User Field Function.
2. Define the new user field function.
3. Activate *Inverse Distance Weight* in the *Expert* category.



When *Inverse Distance Weight* is activated, STAR-CCM+ interpolates vertex values from the nearest cell centers using inverse distance weighting.



The field function ϕ_v at a node is calculated as:

$$\phi_v = \sum_i \frac{1}{\delta_i} \phi_i \tag{9}$$

where δ_i is the distance between the node and the centroid i (cell center for a cell value and face center for a face value) and ϕ_i is the variable that is stored at the centroid.

User Field Functions Syntax

This section details the syntax that must be followed when writing your own user field functions.

Contents:

Working with Table Values Through Field Functions
Referencing Field Functions, Coordinate Systems, and Reference Frames
Referencing Reports and Global Parameters
Accessing Report Results on a Per-Part Basis
Operators and Functions
Working with Units in Field Functions

Working with Table Values Through Field Functions

You can access table data from user field functions within STAR-CCM+ and use these values for post-processing.

As mentioned in Creating a User Field Function, column table data can be interpolated as functions of other scalar field functions (for example `$Time`), and position tables can be spatially interpolated and accessed as fields. Both scalar and vector tables can be interpolated.

Interpolating Data as a Function of Another Table Column

The syntax for doing this involves the use of a file table's name and the names of its columns. For this technique to work properly, column names must not include any units associated with that column.

Using the field function table interpolation syntax to generate a scalar expression:

```
interpolateTable(@Table("<tablename>"), "<x-column>", {LINEAR | STEP |
SPLINE}, "<phi-column>", <scalar_expression>)
```

Using the field function table interpolation syntax to generate a vector expression:

```
interpolateTable(@Table("<tablename>"), "<x-column>", {LINEAR | STEP},
"<phi_0-column>", "<phi_1-column>", "<phi_2-column>",
<scalar_expression>)
```

where:

- `<tablename>` is a table name from the **Tools** > **Tables** node in the object tree.
- `<x-column>` is the name of that table's column to be used as the `x` data for the interpolation.
- `<scalar_expression>` is the scalar field function that is used as the `x` value for the interpolation of `y`.
- For scalar functions:
  - `<phi-column>` is the table column of scalar y-values to be interpolated as a function of `x`.

- For vector functions:
  - `<phi_0-column>` is the table column of component [0] values to be interpolated as a function of `x`.
  - `<phi_1-column>` is the table column of component [1] values to be interpolated as a function of `x`.
  - `<phi_2-column>` is the table column of component [2] values to be interpolated as a function of `x`.

  The operator returns a vector-valued field from the interpolated `0`, `1`, `2` component columns.

The following interpolation methods are supported:

- `LINEAR` -- uses piecewise-linear interpolation between `y` table values.
- `STEP` -- uses `y` table values at the largest `x` table value that is less than the value of `$scalar_expression`.
- `SPLINE` -- uses B-spline interpolation between `y` table values.

For example, assuming a table `"DataTable"` with columns `"Time"` and `"Phi"` the following:

```
interpolateTable(@Table("DataTable"), "Time", LINEAR, "Phi", $Time)
```

linearly interpolates `"Phi"` as a function of `"Time"`, and provides the interpolated value at the simulation time.

Interpolating Data as a Function of Position

You can use field functions to access table data as a function of position.

The syntax for doing this involves the use of a table file's name and the names of its columns. For this technique to work properly, column names must not include any units associated with that column.

Using the field function table position interpolation syntax to generate a scalar expression:

```
interpolatePositionTable(@Table("<tablename>"), "<phi-column>")
```

or, with a coordinate system:

```
interpolatePositionTable(@Table("<tablename>"),
@CoordinateSystem("<coordinatesystem>"), "<phi-column>")
```

or, with a nested coordinate system:

```
interpolatePositionTable(@Table("<tablename>"),
@CoordinateSystem("<coordinatesystem>.<coordinatesystem>.<coordinatesys
tem>"), "<phi-column>")
```

Using the field function table position interpolation syntax to generate a vector expression:

```
interpolatePositionTable(@Table("<tablename>"), "<phi_0-
column>","<phi_1-column>","<phi_2-column>")
```

or, with a coordinate system:

```
interpolatePositionTable(@Table("<tablename>"),
@CoordinateSystem("<coordinatesystem>"), "<phi_0-column>","<phi_1-
column>","<phi_2-column>")
```

or, with a nested coordinate system:

```
interpolatePositionTable(@Table("<tablename>"),
@CoordinateSystem("<coordinatesystem>.<coordinatesys
tem>"), "<phi_0-column>","<phi_1-column>","<phi_2-column>")
```

where:

- `<coordinatesystem>` is the name of a coordinate system in the **Tools** > **Coordinate Systems** node in the object tree. Multiple instances of this name, which dots separate, represent layers of nesting. When a coordinate system is not specified, the laboratory system is used and assumed to be Cartesian.
- `<tablename>` is a table in the **Tools** > **Tables** node, which contains three columns which specify the coordinates corresponding to the type of `<coordinatesystem>` specified. Column names must be those that the coordinate system dictates, that is, Cartesian (`"X"`, `"Y"`, `"Z"`), cylindrical (`"r"`, `"theta"`, `"z"`) or spherical (`"r"`, `"theta"`, `"phi"`).
- For scalar functions:
  - `<phi-column>` is the table column for spatial interpolation using the nearest neighbor, corresponding to the position coordinate columns.
- For vector functions:
  - `<phi_0-column>`, `<phi_1-column>`, and `<phi_2-column>` represent the column names of the three values of the components of the vector in `<coordinatesystem>`.

Nearest-neighbor (closest point) interpolation is used, based on the coordinate values in each row of the table. Data points in the table do not need to follow an particular spatial ordering (the point data is considered to be scattered).

Interpolating Direction from Parts

The interpolateDirection field function generates a field of direction vectors based on line curves or surfaces that you specify. Having a field of direction vectors can be useful when you model anisotropic behavior in materials.

The field function returns a field of orientation vectors, interpolated from the lines or surfaces to every cell.

The syntax for this field function is:

```
interpolateDirection(@Part("<partname>"),
@Representation("<representation>"),FORWARD)
```

or, with a coordinate system:

```
interpolateDirection(@Part("<partname>"),
@Representation("<representation>"),
@CoordinateSystem("<coordinatesystem>"),FORWARD)
```

where:

- `<coordinatesystem>` is the name of a coordinate system in the **Tools** > **Coordinate Systems** node in the object tree. The direction vectors are returned based on this coordinate system. When a coordinate system is not specified, the laboratory system is used and assumed to be Cartesian.
- `<partname>` is a surface or a line curve. For a surface, the direction field is given using the normals of that part. For a line curve, the direction field is given using the line segment directions. If using a surface or a curve name of a part, the syntax is `<partname.surface/curve name>`. For a region, it is `<regionname.boundary/feature curve name>`.
- `<representation>` is the representation that is used to extract the part. It is set in the **Representation** node in the object tree.
- `FORWARD` is used to set the orientation of the vectors based on the direction of the normals of the input surface or the direction of the line segment of the input line. Use `BACKWARD` to reverse the direction vectors.

The syntax for this field function with multiple parts is:

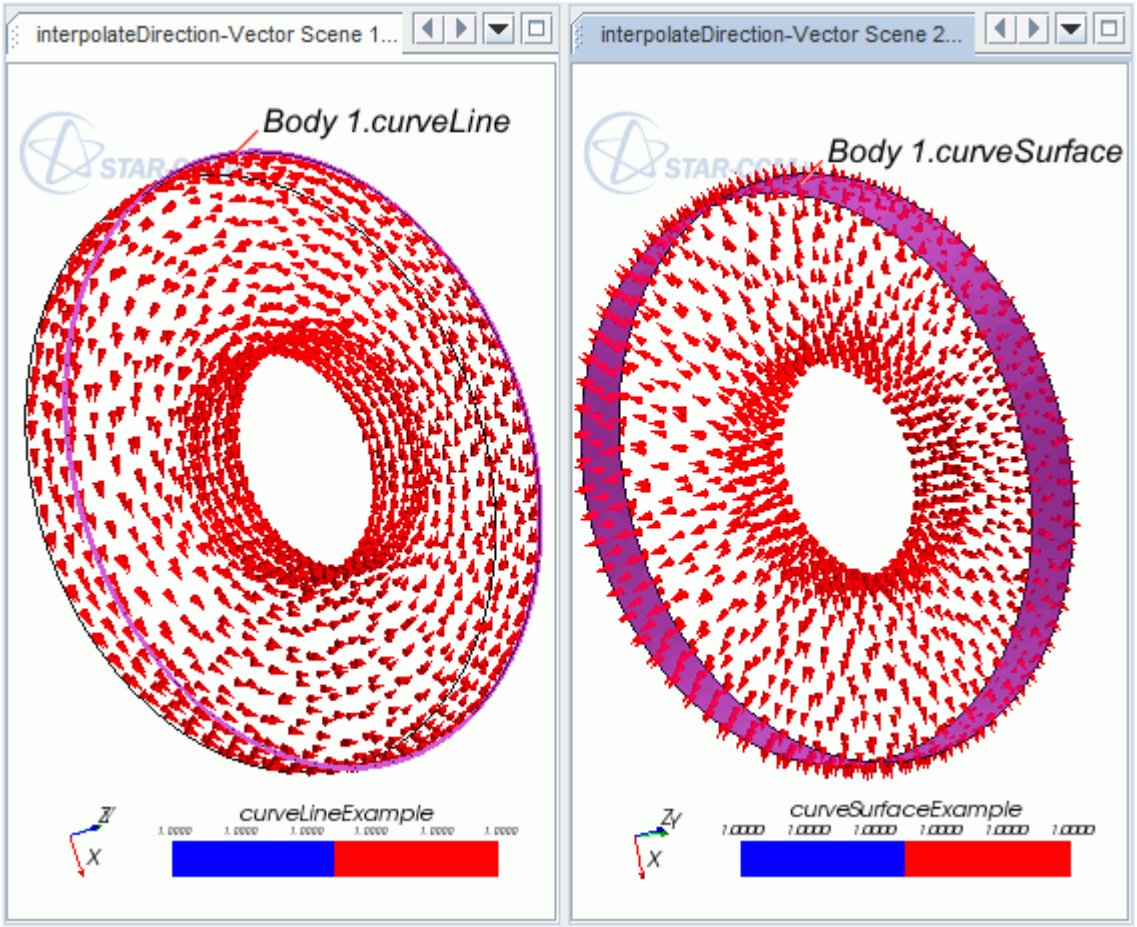
```
interpolateDirection(@Part("<partname1>::<partname2>::<partname3>"),
@Representation("<representation>"),FORWARD)
```

or, with a coordinate system:

```
interpolateDirection(@Part("<partname1>::<partname2>::<partname3>"),
@Representation("<representation>"),
@CoordinateSystem("<coordinatesystem>"),FORWARD)
```

The parts are delimited using `::`

The following is an example of using this field function. The left image is a tangential field function that uses a line curve input. The right image shows a radial field function that uses a surface input for the given geometry.



Referencing Field Functions, Coordinate Systems, and Reference Frames

When defining an expression or a field function, you can reference other field functions within it by using the name from the *FunctionName* property prefixed with `$`, such as `$MachNumber`.

If the name contains any non-alphanumeric characters apart from '_', Enclose the field function name in curly braces, '{}'. Also use curly braces if the name begins with a number, or if you are referencing object names that include characters from non-Western alphabets. Some examples of referencing such names are:

- ${Five Things}
- ${Temp@X}
- ${3rdInletMassflow}
- ${test-Temperature}

Field functions used to define expressions cannot be a function of space; they have to evaluate to a single number as the expressions are constants in space. Variables such as `$Time`, `$TimeStep`, and `$Iteration`, as well as the field functions associated with reports, are valid. Variables such as `$Temperature` or `$Pressure` have different values at every different point in space and cannot be used in expressions.

The syntax for referencing field functions, coordinate systems, and reference frames is presented below.

Note: When you reference a field function in an expression, use the function name that is specified in the *Function Name* property. Since the names are case-sensitive, they must be entered in the field function definition exactly as they are shown in the function name.

Referencing Scalar Field Functions

To reference an existing scalar field function, the variable name in the new definition is the existing function name (as defined in the *Function Name* property of the *Expert* category) prepended by a dollar ($) symbol. For example:

```
$Temperature
```

Referencing Vector Field Functions

To reference an existing vector, array, or position field function, the variable name in the new definition is the existing function name (as defined in the *Function Name* property of the *Expert* category) prepended by two dollar symbols ($$). For example:

```
$$Velocity
```

Referencing Components of a Vector Field Function

If you are defining a new user field function that uses components of a vector, array, or position field function, your function is clearer and executes efficiently if you write the function directly in terms of the original vector function, rather than going through the derived scalar.

For example, to define a new scalar function that is equal to the sum of the Cartesian velocity components, you can define it in terms of the velocity vector as:

```
$$Velocity[0] + $$Velocity[1] + $$Velocity[2]
```

Referencing Local Coordinate Systems

In STAR-CCM+, a distinction is made between vector and position field functions so that you can transform individual field values between coordinate frames appropriately.

For example, let C denote a local Cartesian coordinate frame with origin (position) O and orientation matrix R, whose columns are the orthonormal basis vectors of C with respect to the Laboratory frame. The transformation of a vector v, from laboratory coordinates into C, is given by:

$$v_C = R^T v \tag{10}$$

However, transforming a point p from the Lab frame into C requires subtraction of the local origin O from p, then computing the coordinates of p within the local frame:

$$p_C = R^T\left(p - O\right) \tag{11}$$

The transformations of a vector and a position from C into the Laboratory frame are:

$$v = Rv_C \tag{12}$$

and:

$$p = Rp_C + O \tag{13}$$

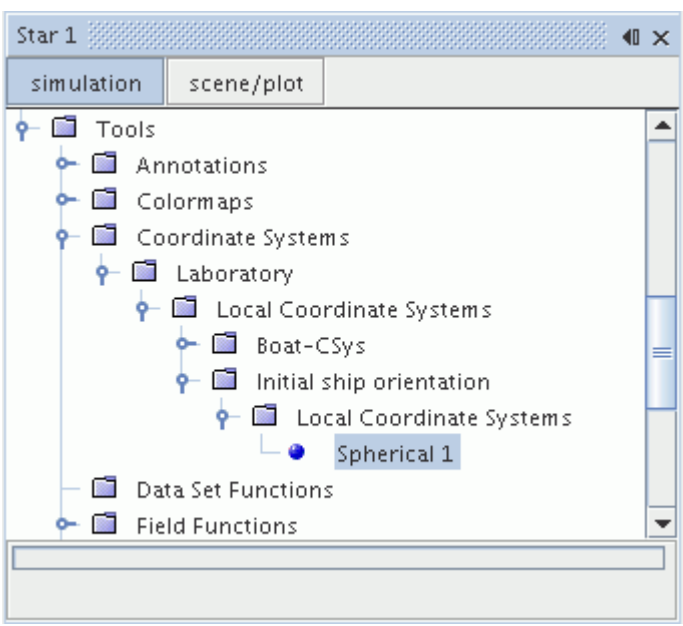
There is a syntax to transform position-type and vector-type fields to different local coordinate systems. For example:

```
$$Position(@CoordinateSystem("Laboratory.Coordinate System
1"))
```

transforms the `Position` field from the `Laboratory` frame to the `Coordinate System 1` frame. For positions, components of the vector are in the units of the coordinate system specified:

- If the coordinate system is Cartesian, `$$Position(@CoordinateSystem("Laboratory.CartesianSystem"))` is `[X,Y,Z]`.
- If the coordinate system is cylindrical, `$$Position(@CoordinateSystem("Laboratory.CylindricalSystem"))` is `[radius, theta, z]`.
- If the coordinate system is spherical, `$$Position(@CoordinateSystem("Laboratory.SphericalSystem"))` is `[radius, theta, phi]`.

For cases where local coordinate systems have one or more levels of nesting, use the qualifier `@CoordinateSystem` and separate each coordinate system from its parent using the `.` character. As an example, consider a field function that gives the position in the **Spherical 1** local coordinate system for the simulation tree below:



The field function would be written as:

```
$$Position(@CoordinateSystem("Laboratory.Initial ship
orientation.Spherical 1"))
```

The **Boat-CSys** coordinate system would be referred to as:

```
$$Position(@CoordinateSystem("Laboratory.Boat-CSys"))
```

Note: The differences between the `Position` and `Centroid` fields are important:

- `Position` gives the coordinates of either the vertices or the cell centers, depending on the `smooth` flag.
- `Centroid` specifically gives the geometric centroid of the cells of the simulation mesh.

In general, `Position` should be used within user field function expressions since it provides the appropriate geometric location *without any interpolation*, even on derived parts. In post-processing operations, such as reporting and plotting, `Position` uses the appropriate geometric location on the geometry with respect to the `smooth` flag. If the `smooth` flag is set, the vertex locations are used. If the `smooth` flag is not set, the cell, face, or line segment centers are used. The `Centroid` field, however, specifically refers to the *geometric centroid* of the cells or faces of the simulation mesh. If the `smooth` flag is set during a visualization or analysis operation, `Centroid` is interpolated like any other field to the vertex locations of the simulation mesh or the derived part, possibly introducing small interpolation errors.

Referencing Velocity in a Local Reference Frame

Using field function syntax, you can convert velocity in the laboratory coordinate system to the equivalent velocity in a local reference frame. For example:

```
$$Velocity(@ReferenceFrame("Rotating"))
```

This function yields a vector of `Velocity` in the local reference frame named `Rotating`.

Referencing Reports and Global Parameters

Reports and global parameters are referenced directly by their names in expressions. For example a Sum report named **Sum 1** is referenced in an expression as follows:

```
sqrt(${Sum 1})
```

For expressions prior to v12.06, referencing reports by their associated field functions works:

```
sqrt(${Sum1Report})
```

Accessing Report Results on a Per-Part Basis

In an expression report, you can use the `getReportValue` field function operator to extract scalar values for a subset of parts that are included in a scalar report, in addition to the overall report results.

For more information about expression reports, see Expression Report.

The operator syntax is:

```
getReportValue(${Scalar Report}, "name", {LITERAL | PATTERN})
```

where:

- `Scalar Report` is the name of a scalar report field function.

  Note: For simulations created in versions of STAR-CCM+ prior to 12.06, the function name of the report can still be used. So in this example, `${Scalar Report}` could be written as `$scalarReport`.

- `name` can be the literal name of a part that is referenced by the scalar report or a string pattern (following the rules of a regular expression).
- `name` must specify a subset of the list of parts that are included in `Scalar Report` itself.
- The third argument `LITERAL` or `PATTERN` denotes the type of input of the second argument (and is optional). If it is omitted, `getReportValue(${Scalar Report}, "name")` is equivalent to `getReportValue(${Scalar Report}, "name", PATTERN)`.

Note: This feature is available for scalar values only.

Arithmetic, Logical, and Trigonometric Operators

- Arithmetic binary operators (+, -, *, /...) are supported and the defined behavior is as follows:

  ```
  getReportValue($report1, "nameA", LITERAL) +
  getReportValue($report2, "name")
  ```

  or

  ```
  getReportValue($report2, "name") + getReportValue($report1,
  "nameA", LITERAL)
  ```

  results in the value of the `nameA` part value of the report `report1` being added to the part value(s) of `report2` that correspond to the string pattern `name`, and the overall value of

  ```
  getReportValue($report2, "name")
  ```

  ```
  getReportValue($report1, "nameA") + getReportValue($report2,
  "nameB")
  ```

  results in part values of the common parts being added and also the overall values being added. The part list of the result consists of the union of parts `nameA` and `nameB`. A per-part error `nonexistent report part` is output for each part that is not present in both, which results in an overall error status of `incomplete computation`.

- Logical operators are supported and the behavior is similar to binary operations with regard to per-part values, overall values, and errors for missing parts.
- Trigonometric functions and other miscellaneous functions like `sqrt`, `log`, `floor`, and `ceil` are supported.

Conditional and Alternate Value Operators

- Conditional expression is supported. In the example

  ```
  getReportValue($report1, "nameA") > getReportValue($report2,
  "nameB") ? getReportValue($report3, "nameC") :
  getReportValue($report4, "nameD", LITERAL)
  ```

  the overall value of

  ```
  getReportValue($report1, "nameA") > getReportValue($report2,
  "nameB")
  ```

  dictates whether the output of the expression is

  `getReportValue($report3, "nameC")` or `getReportValue($report4, "nameD", LITERAL)`.

- The alternate value function is supported. In the example

  ```
  alternateValue(getReportValue($report1, "nameA"), 1.0)
  ```

  if `getReportValue($report1, "nameA")` cannot be evaluated for some reason, the alternate value `1.0` is used. Note that an error during the evaluation of `getReportValue($report1, "nameA")` does not result in the alternate value `1.0`.

Example of Using the getReportValue Operator

Consider a report `MaximumA` with the following list of parts: `part`, `part 1`, `part surface 1`, `part 2`, `plane section 1`, `region 1`, `boundary 1`, `boundary 2`, and `boundary 3`:

```
getReportValue($MaximumAReport, "part.*", PATTERN)
```

results in part values of:

- `part`
- `part 1`
- `part surface 1`
- `part 2`

and an overall value that is maximum of the above four part values (not the `MaximumA` report value).

```
 getReportValue($MaximumAReport, "part", LITERAL)
```

results in no part values but an overall value of report value of `part`.

```
getReportValue($MaximumAReport, "part", PATTERN)
```

results in both a part value and an overall value which are one and the same. You get a different output from the LITERAL option.

```
getReportValue($MaximumAReport, "part.*", LITERAL)
```

produces an error with `nonexistent report part` since there is no part `part.*`.

```
getReportValue($MaximumAReport, ".* 1", PATTERN)
```

results in part values of:

- `part 1`
- `part surface 1`
- `plane section 1`
- `region 1`
- `boundary 1`

and an overall value of maximum of the above five part values.

Common examples of string patterns used in regular expressions are:

- `.*` - all parts
- `.*surface` - all parts that end in "surface"

So you can use the following syntax:

```
getReportValue($reportFF, ".*")
```

This technique generates output of per-part value of all the parts along with the overall report value, similar to the **Run Report** action.

When you include the `getReportValue` operator in an expression report, you obtain a subset of part report values of a scalar report. The expression report output has been modified to resemble the output of a scalar report, if the expression report has part values. So

```
getReportValue(${Scalar Report}, "name", LITERAL)
```

results in no part values but just an overall value equal to the part report value. However,

```
getReportValue(${Scalar Report}, "name")
```

or

```
getReportValue(${Scalar Report}, "name", PATTERN)
```

results in part values and an overall value.

Operators and Functions

Arithmetic Operators

Operator	Function
+	plus
-	minus
*	multiply

Operator	Function
/	divide

Logical Operators

Operator	Function
==	equal to
!=	not equal to
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
&&	and
||	or

Trigonometric Functions

Operator	Function
acos($x)	arccosine of x
asin($x)	arcsine of x
atan($x)	arctangent of x
atan2($y,$x)	arctangent of y/x
cos($x)	cosine of x
cosh($x)	hyperbolic cosine of x
sin($x)	sine of x
sinh($x)	hyperbolic sine of x
tan($x)	tangent of x
tanh($x)	hyperbolic tangent of x

Vector Functions

These operators work with vector, array, and position field functions.

Operator	Function
mag($$u)	vector magnitude
mag2($$u)	vector magnitude squared
unit($$u, $x)	unit vector, where $x is the magnitude tolerance

	When the magnitude of a vector is smaller than the specified tolerance, the vector is omitted from the display.
dot($$u,$$v)	dot product of u and v
cross($$u,$$v)	cross product of u and v
curl($$v)	curl of vector v In vector field notation, the curl of vector v is defined as: $$\vec{\nabla} \times v = \left(\frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z}\right)i + \left(\frac{\partial v_x}{\partial z} - \frac{\partial v_z}{\partial x}\right)j + \left(\frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y}\right)k$$
div($$v)	divergence of vector v returned as a scalar In vector field notation, the divergence of vector v is defined as: $$\nabla \cdot v = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z}$$

Symmetric Tensor Functions

In addition to basic arithmetic operations, the following tensor-specific operations are supported. Letting $\mathbf{A}$ denote any 3x3 symmetric tensor:

Operator	Function
`$$$A.eigValue(<index>)`, `eigValue($$$A, <index>)`	i^{th} eigenvalue of $\mathbf{A}$: $\lambda_0 \geq \lambda_1 \geq \lambda_2$, *signed* order
`$$$A.eigVector(<index>)`, `eigVector($$$A, <index>)`	i^{th} eigenvector of $\mathbf{A}$ The ith eigenvector corresponds to the ith eigenvalue of $\mathbf{A}$. The eigenvectors are normalized to unit vectors. The eigenvalue decomposition will produce a set of orthonormal eigenvectors.
`$$$A.trace()`, `trace($$$A)`	Trace of $\mathbf{A}$: $$tr\left(\mathbf{A}\right) = \sum_{i=0}^{n-1} a_{ii} = \sum_{i=0}^{n-1} \lambda_i$$
`$$$A.norm("infinity")`, `norm($$$A, "infinity")`	Infinity norm of $\mathbf{A}$: $\lVert\mathbf{A}\rVert_\infty$ = maximum absolute *row* sum of $\mathbf{A}$
`$$$A.norm(1)`, `norm($$$A, 1)`	1-norm of $\mathbf{A}$: $\lVert\mathbf{A}\rVert_1$ = maximum absolute *column* sum of $\mathbf{A}$
`$$$A.norm(2)`, `norm($$$A, 2)`	2-norm of $\mathbf{A}$: $\lVert\mathbf{A}\rVert_2$ = maximum $\lvert\lambda_i\rvert$ of $\mathbf{A}$
`$$$A.norm("frobenius")`,	Frobenius norm of $\mathbf{A}$:

Operator	Function
`norm($$$A, "frobenius")`	$\|\mathbf{A}\|_F = \sqrt{\sum_{i,j=0}^{n-1} \left\|a_{i,j}\right\|} = \sqrt{tr\left(\mathbf{A}^2\right)}$
`$$$A.mag()`, `mag($$$A)`	Magnitude of $\mathbf{A}$: $\|\mathbf{A}\|_F$
`$$$A.mag2()`, `mag2($$$A)`	Squared magnitude of $\mathbf{A}$: $\|\mathbf{A}\|_F^2$
`$$$A.dot($$$B)`, `dot($$$A, $$$B)`	Double-dot (double-inner) product of $\mathbf{A}$ and $\mathbf{B}$: $\mathbf{A}:\mathbf{B} = \sum_{i,j=0}^{n-1} a_{i,j} b_{i,j}$
`$$$A.dotVector($$v)`, `$$v.dotVector($$$A)`, `dotVector($$$A, $$v)`, `dotVector($$v, $$$A)`	Matrix-vector product of $\mathbf{A}$ and $\mathbf{v}$: $\mathbf{A}\mathbf{v} = \mathbf{v}^T\mathbf{A}$, for $\mathbf{A}$ symmetric

Miscellaneous Functions

Operator	Function
ceil($x)	smallest integer not less than x, as a real
exp($x)	exponential function, e to the power x
abs($x)	absolute value of x
floor($x)	largest integer not greater than x, as a real
log($x)	base e logarithm of x, natural log
log10($x)	base 10 logarithm of x
sqrt($x)	square root of x
pow($x,$y)	x raised to the power of y
fmod($x,$y)	x modulus y; the floating point remainder of x/y
mod($x,$y)	equivalent to fmod($x,$y)
min($x,$y)	the minimum of x and y
max($x,$y)	the maximum of x and y
grad($phi)	gradient of scalar phi returned as a vector

Operator	Function
	The gradient of scalar ϕ is defined as: $$\nabla\phi = \frac{\partial\phi}{\partial x}i + \frac{\partial\phi}{\partial y}j + \frac{\partial\phi}{\partial z}k$$ Values calculated using this approach are not necessarily the same as gradients returned from the solvers. This is due to different approaches used in calculating the gradients. In the solvers, a two-pass Gauss algorithm is used, whereas the function provided here uses a least squares approach. Solvers also have more information on boundary values than is available to this function.

Conditional Expressions

The ternary operator “?:” defines a condition operation if-then-else similar to the C language. For example:

```
(a > b) ? a : b
```

is equivalent to

```
z = max(a, b)
```

You can also use parentheses to nest conditional expressions.

Nested conditional expressions allow you to specify different values for a user field function over different time intervals:

```
($Time <= t_1) ? Value 1 :
(
($Time > t_1 && $Time <= t_2) ? Value 2 : Value 3
)
)
```

For example, consider a case where the value alternates between 125 and 0 in five-second intervals, and remains at 0 after 15 seconds:

```
($Time <= 5) ? 125 :
(
($Time > 5 && $Time <= 10) ? 0 :
(
($Time > 10 && $Time <= 15) ? 125 : 0
)
)
```

The ternary “?:” operator, on the other hand, requires all fields to be available on the part (such as the boundary or volume). If one of the fields is not available on the part, then the entire user-defined expression involving the “?:” operator is judged to be invalid, the field function fill fails, and, for example, you see gray in a scene.

Alternate Value Functions

If the value for a scalar or vector expression on a part is not available or computable, you can provide an alternate value for that expression.

Normally STAR-CCM+ requires all terms in an expression to exist for the expression to be computable. For example, in the following expression:

```
$Pressure * $Volume
```

if `Pressure`, `Volume`, or both do not exist for a part, this expression is not computable on that part.

To provide an alternate value, use the `alternateValue()` operator:

```
alternateValue($Pressure, 0) * $Volume
```

In this example, STAR-CCM+ substitutes zero for Pressure if Pressure is not available.

The alternate value operator is not restricted to constants; it can also work with another expression. The syntax for this use of the operator is `alternateValue(<scalar/vector expression 1>, <scalar/vector expression 2>)`, for example:

```
alternateValue($Pressure1, $Pressure2) * $Volume
```

In this example, if both Pressure`1` and Pressure`2` are not computable, then the `alternateValue` expression is not computable.

Contents:

Coordinate System Averaging

Coordinate System Averaging

Coordinate system averaging combines the three-dimensional STAR-CCM+ solution results with a typical two-dimensional meridional view. While the meridional view is commonly used in turbomachinery, it can also be applied to certain other applications that involve axisymmetric regions.

This technique can provide an average of results in the θ direction, although you can choose any direction for projecting this average. The calculation is defined as:

$$\overline{\phi_\theta} = \frac{\int_{\theta_1}^{\theta_2} \phi \, \partial\theta}{\int_{\theta_1}^{\theta_2} \partial\theta} = \frac{1}{(\theta_2 - \theta_1)} \int_{\theta_1}^{\theta_2} \phi \, \partial\theta \tag{14}$$

This definition can be implemented in STAR-CCM+ with a `csavg()` field function, in a coordinate system and direction that you specify:

```
csavg($Pressure, @CoordinateSystem("Block-Mapped 1"), 2, 0, 20, 20)
```

where, in this example:

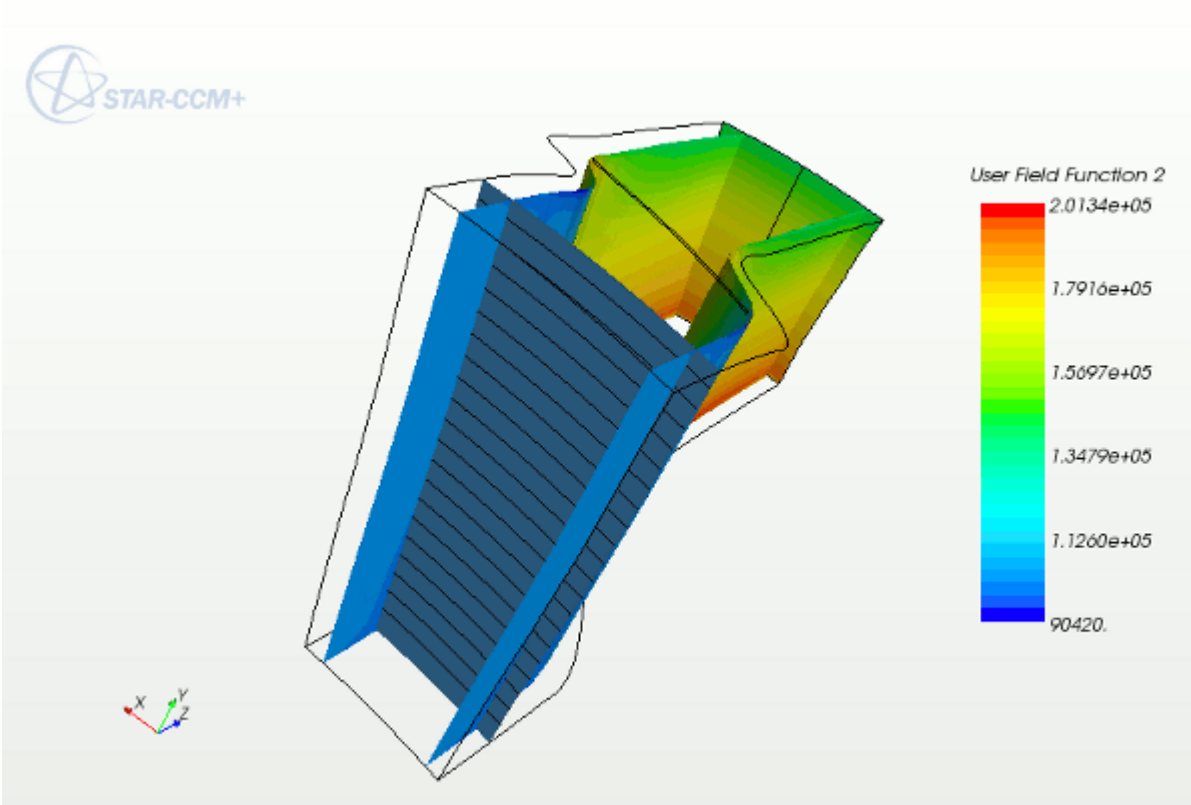
- `$Pressure` is the scalar being defined. (A vector can also be defined in this type of field function.)

- `CoordinateSystem("Block-Mapped 1")` is the (block-mapped) coordinate system being used for the averaging.
- `2` is the averaging direction.
- `0` is the slice direction.
- `20, 20` are the number of slices and the integrations per slice, respectively.

Using a two-dimensional grid, the approach involves projecting region vertices to the grid and interpolating the average to the vertices.

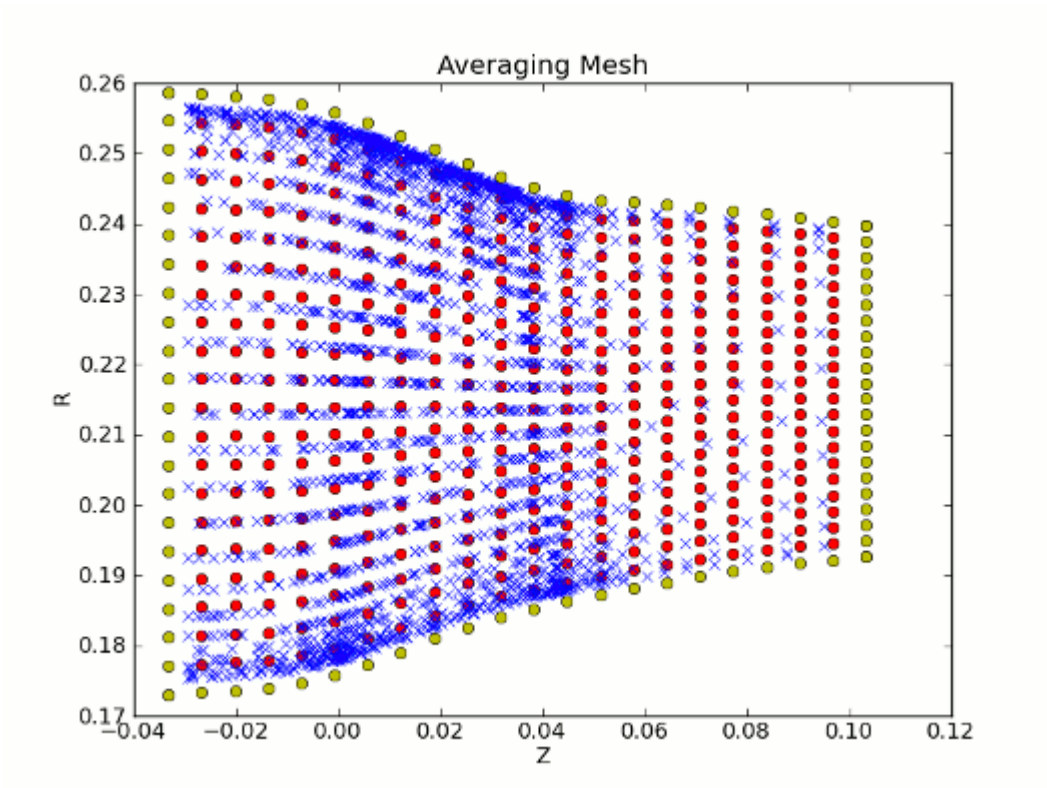
Examples in this section are from the Rotor 37 case.

This scalar scene illustrates projection of vertices to the grid:



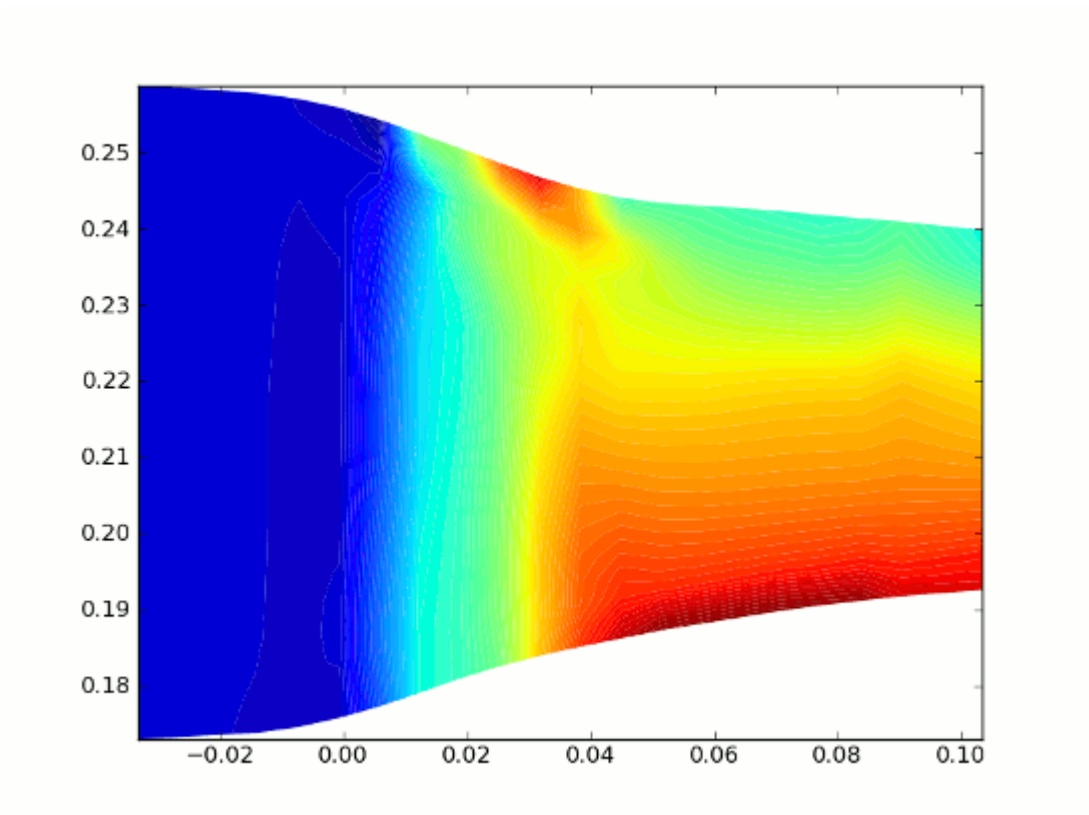
- The blue-gray plane shows a constant z slice.
- Black lines are averaging lines of constant r.
- The results from multiple slices appear in a two-dimensional average mesh.

The following screenshot shows these values in a plot:



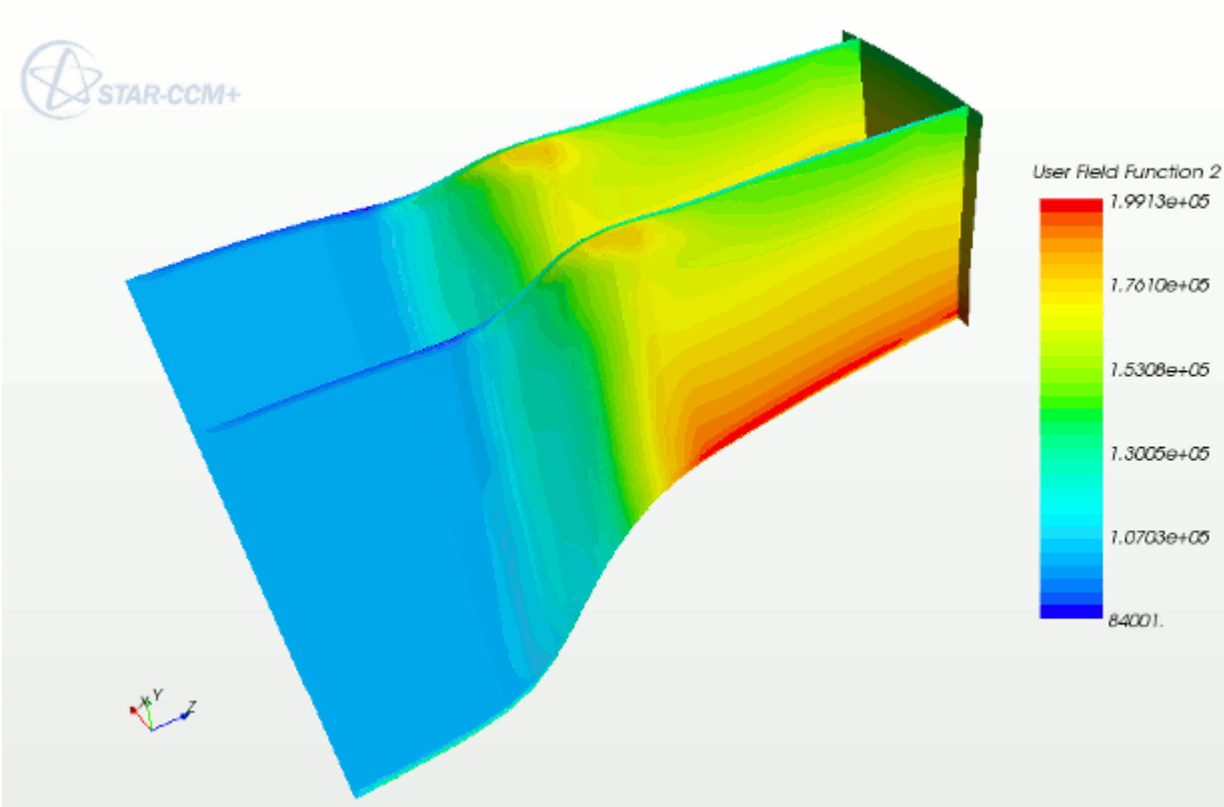
- Red dots are the resulting average mesh.
- Blue “x” symbols are the (z, r) vertices from the region projected onto this mesh.
- Green dots are extrapolated (halo) nodes from the average mesh.

For comparison, look at the scalar-scene representation of the average mesh.



The scalar scene in the following screenshot shows the Rotor 37 case using the following user-defined field function:

```
csavg($AbsoluteTotalPressure, @CoordinateSystem("Cylindrical 1"), 1,
2, 20, 60)
```



This field function takes the average of Absolute Total Pressure, using the Cylindrical 1 coordinate system in the θ direction, and slicing in the z direction. It uses a 20 x 60 averaging mesh -- that is, this mesh has 20 z-slices and, for each z-slice, 60 averaging lines (or r-slices).

Working with Units in Field Functions

You can include units when defining a field function. STAR-CCM+ assumes all physical quantities to be in SI units by default.

However, literal constants can have a unit specification in a different system. The syntax for the quantities is `<number units>`, for example:

```
<32 ft/s^2>

$Pressure - <32 psi>

<27 in> + <3 ft>
```

The units are parsed at run time, so they do not use the Preferred units setting and do not necessarily need to match any of the predefined units that the objects in the **Tools** > **Units** manager node represent.

Additionally, "-" binds tighter than "/", so `kg/m-s^2` equals `(kg)/(m-s^2)` not `(kg/m)*(s^2)`.

User Field Function Examples

This section provides some examples of user field functions.

Scaling an Existing Field Function

Suppose that you wish to create a user field function that defines the distance in the x direction in inches, rather than in the SI units of the computation. You could create a field function that is called, say, “X inches” with the following definition:

```
$$Centroid[0]/0.0254
```

Setting a Transient Boundary Condition

Suppose that you want to set a transient boundary condition, such that temperature ramps up linearly from 300K to a value of 1000K at a time of 0.01 seconds. Your user field function definition (to be used in the definition of the boundary temperature) would be:

```
($Time >= 0.01)  ?  1000 : 300+70000*$Time
```

Specifying Components of a Vector Field Function

To define a vector, array or position field function in terms of its components, specify the three scalar components inside square brackets separated using commas. For example, to set up a field function to initialize the velocity in a reducing pipe, where the velocity is 2 m/s in the larger diameter and 10 m/s in the smaller diameter, and the diameter changed at x=4. The field function definition would be:

```
[($$Centroid[0] < 4) ? 2 : 10, 0, 0]
```

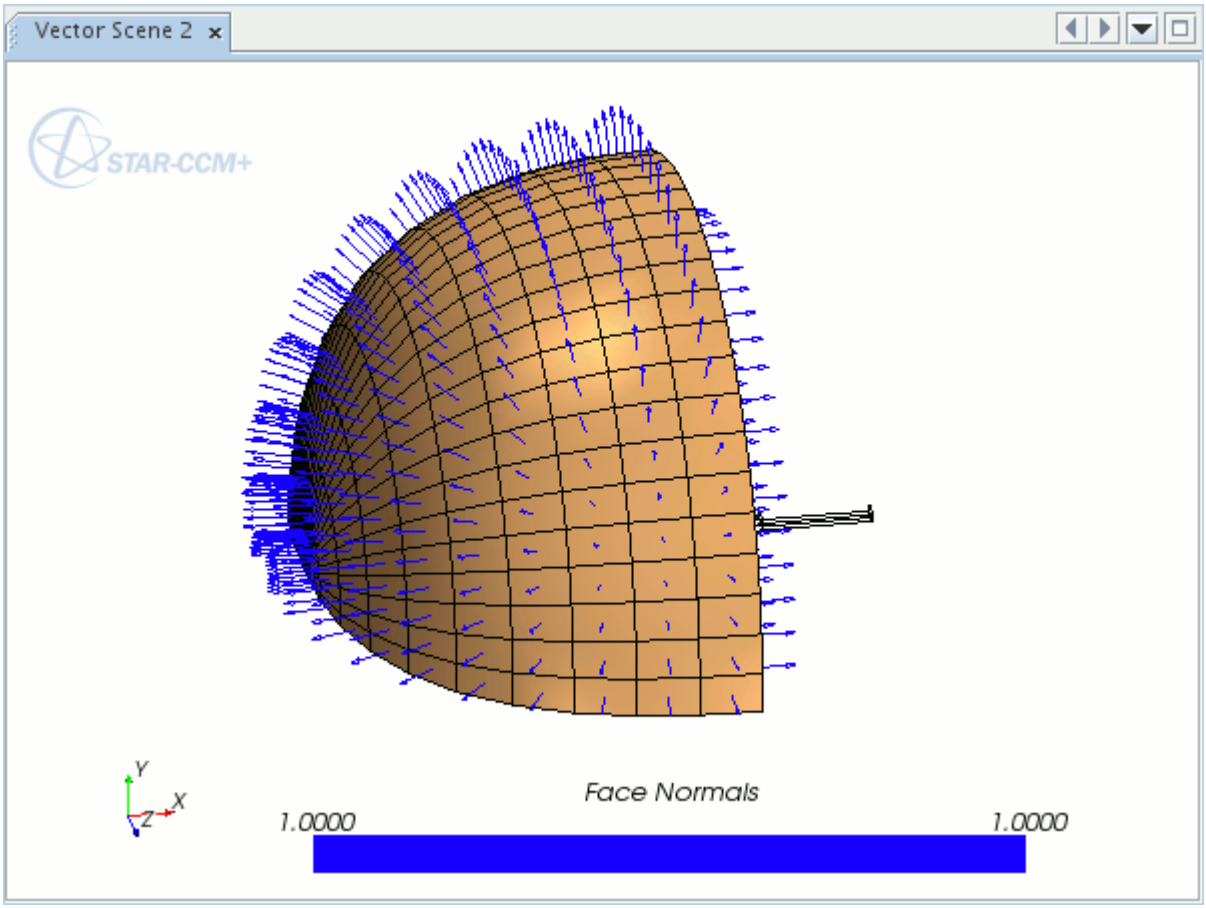
Calculating and Visualizing Face Normals

In STAR-CCM+, the Area field function is a vector field function of the area components, that is the area magnitude multiplied by the unit normal. Therefore, to calculate the unit face normals, a field function can be created to normalize the Area field function by the area magnitude:

- Create a field function that is called, for example, `Face Normals`.
- Set its type to `Vector` and the definition to:

  ```
  $$Area / $$Area.mag()
  ```

This field function can be used to visualize the face normals by creating a vector scene, and setting the vector quantity to your Face Normals field function. You can also add a vector displayer to an existing scene. An example scene is shown below.



Creating a Velocity Field Function Based on Grid Flux

To create a field function that is based on Grid Flux that has the same units as velocity:

1. Create a user-defined field function and rename it to `Grid Flux Velocity`.
2. Set its dimensions to Length = `1` and Time = `-1`.
3. Set its type to `Vector`.
4. Set its definition to:

```
$$GridFlux * $$Area / $$Area.mag2()
```

In this expression, `$$GridFlux / $$Area.mag()` is the magnitude of the velocity, and `$$Area / $$Area.mag()` is the direction of the flux, normal to the face area.

Calculating Derivatives with Vector Field Functions

Typical user field functions involving derivatives are:

- Divergence of velocity $\nabla \cdot v$ is specified as:

```
div($$Velocity)
```

- Divergence of density, velocity $\nabla\cdot(\rho v)$ is specified as:

```
div($Density * $$Velocity)
```

- Gradient of pressure ∇p is specified as:

```
grad($Pressure)
```

- Curl of velocity $\nabla\times v$ is specified as:

```
curl($$Velocity)
```

A cell-based least squares gradient algorithm is used for the computation of the field gradients. At boundaries, zero-order extrapolation is used. This algorithm differs from the gradients that the various solvers compute. Therefore, the gradients computed for post-processing are likely to differ slightly from solver gradients that are viewed by using the *Temporary Storage Retained* option in specific solvers. However, they are sufficiently close for most practical purposes.

Field Functions Reference

Field Functions Manager Right-Click Actions

These context menu items are available when you right-click the **Field Functions** manager node.

New	Creates a field function of the type you select in the submenu.	
	Scalar	Creates a scalar field function.
	Array	Creates a array field function.
	Vector	Creates a vector field function.
	Position	Creates a position field function.

Primitive Field Function Properties

Type	The type of field function: scalar, vector, array, or position (read-only).	
Function Name	The name of this field function (read-only). Use this property in the definition of other field function definitions, not the node name. Use this property to remind you of the actual field function if you have renamed the node in the tree.	
Dimensions	The physical dimensions of this field function (read-only).	
Inverse Distance Weight	Whether to use inverse distance weighting to interpolate node values.	
	Activated	Interpolate node values using inverse distance weighting.
	Deactivated	Use nearest neighbor as node values.

System Field Function Properties

Type	The type of field function: scalar, vector, array, or position (read-only).
Dimensions	The physical dimensions of this field function (read-only).
Function Name	The name of this field function (read-only). Use this property in the definition of other field function definitions, not the node name. Use this property to remind you of the actual field function if you have renamed the node in the tree.
Definition	The definition of the field function in the STAR-CCM+ field function syntax (read-only).
Assembly Code	An unambiguous tree-based (but humanly readable) form of the field function formula (read-only). This property is empty until the field is used, for example in a contour plot.
Inverse Distance Weight	Whether to use inverse distance weighting to interpolate node values.

	Activated	Interpolate node values using inverse distance weighting.
	Deactivated	Use nearest neighbor as node values.

User-Defined Field Function Properties

Type	Defines the type of field function. See Field Functions Types.	
	Scalar	Defines the field function as scalar which means that the field function corresponds to just one number (zero, positive or negative) at a specific location.
	Array	Defines the field function as an array of three numbers at each location.
	Vector	Defines the field function as a vector which can be expressed in different coordinate systems. See Referencing Local Coordinate Systems.
	Position	Defines the field function as a position at each location. See Referencing Local Coordinate Systems.
	Symmetric Tensor	Defines the field function as a second-order symmetric tensor at each location.
Dimensions	The physical dimensions of this field function, which you can customize.	
Function Name	The name of this field function. Use this property in the definition of other field function definitions, not the node name. Use this property to remind you of the actual field function if you have renamed the node in the tree. If you inadvertently use a function name that another field function uses, STAR-CCM+ appends an underscore (_) and a digit to what you typed. The result is a name that is still unique.	
Definition	The formulation of this field function using the special STAR-CCM+ field function syntax. Entering an invalid definition causes the error: `Error compiling Field Function: parse error` to be displayed in the *Output* window, and the *Assembly Code* property to be populated with `error`.	
Ignore Boundary Values	Determines whether the field functions are computed based on the data that are stored in the cells that are next to the boundaries rather than the field data stored on boundaries (if available).	
	Activated	Use adjacent cell values.
	Deactivated	Use boundary values.
Assembly Code	An unambiguous tree-based (but humanly readable) form of the field function formula (read-only). Empty until the *Definition* is entered or until the field is used, for example in a contour plot. The assembly code is updated each time that you change the *Definition* and press `<Enter>`. This property can be useful to make sure that STAR-CCM+ interprets your definition as expected (for example, operator precedence).	

Inverse Distance Weight	Whether to use inverse distance weighting to interpolate node values.	
	Activated	Interpolate node values using inverse distance weighting.
	Deactivated	Use nearest neighbor as node values.

Contents:

Heat Transfer Coefficient Field Function Properties
Nusselt Number Field Function Properties
Pressure Coefficient Field Function Properties
Skin Friction Coefficient Field Function Properties
Temperature Coefficient Field Function Properties
Total Pressure Coefficient Field Function Properties

Heat Transfer Coefficient Field Function Properties

Reference Temperature	The reference temperature, which can be specified with a text entry.
Type	The type of field function (scalar; read-only).
Dimensions	The physical dimensions of this field function (read-only).
Function Name	The name of this field function (read-only). Use this property in the definition of other field function definitions, not the node name. Use this property to remind you of the actual field function if you have renamed the node in the tree.

Expert Properties

Inverse Distance Weight	Whether to use inverse distance weighting to interpolate node values.	
	Activated	Interpolate node values using inverse distance weighting.
	Deactivated	Use nearest neighbor as node values.

Nusselt Number Field Function Properties

Dimensions	The physical dimensions of this field function (read-only).
Reference Length	The reference length, which can be specified with a text entry.
Reference Temperature	The reference temperature, which can be specified with a text entry.
Reference Thermal Conductivity	The reference thermal conductivity, which can be specified with a text entry.
Type	The type of field function (scalar; read-only).

Function Name	The name of this field function (read-only). Use this property in the definition of other field function definitions, not the node name. Use this property to remind you of the actual field function if you have renamed the node in the tree.

Expert Properties

Inverse Distance Weight	Whether to use inverse distance weighting to interpolate node values.	
	Activated	Interpolate node values using inverse distance weighting.
	Deactivated	Use nearest neighbor as node values.

Pressure Coefficient Field Function Properties

Dimensions	The physical dimensions of this field function (read-only).
Reference Density	The reference density, which can be specified with a text entry.
Reference Pressure	The reference pressure, which can be specified with a text entry.
Reference Velocity	The reference velocity, which can be specified with a text entry.
Type	The type of field function (scalar; read-only).
Function Name	The name of this field function (read-only). Use this property in the definition of other field function definitions, not the node name. Use this property to remind you of the actual field function if you have renamed the node in the tree.

Expert Properties

Inverse Distance Weight	Whether to use inverse distance weighting to interpolate node values.	
	Activated	Interpolate node values using inverse distance weighting.
	Deactivated	Use nearest neighbor as node values.

Skin Friction Coefficient Field Function Properties

Dimensions	The physical dimensions of this field function (read-only).
Reference Density	The reference density, which can be specified with a text entry.
Reference Velocity	The reference velocity, which can be specified with a text entry.
Type	The type of field function (scalar; read-only).

Function Name	The name of this field function (read-only). Use this property in the definition of other field function definitions, not the node name. Use this property to remind you of the actual field function if you have renamed the node in the tree.

Expert Properties

Inverse Distance Weight	Whether to use inverse distance weighting to interpolate node values.	
	Activated	Interpolate node values using inverse distance weighting.
	Deactivated	Use nearest neighbor as node values.

Temperature Coefficient Field Function Properties

Dimensions	The physical dimensions of this field function (read-only).
Reference Temperature Delta	The reference temperature difference, which can be specified with a text entry.
Reference Temperature	The reference temperature, which can be specified with a text entry.
Type	The type of field function (scalar; read-only).
Function Name	The name of this field function (read-only). Use this property in the definition of other field function definitions, not the node name. Use this property to remind you of the actual field function if you have renamed the node in the tree.

Expert Properties

Inverse Distance Weight	Whether to use inverse distance weighting to interpolate node values.	
	Activated	Interpolate node values using inverse distance weighting.
	Deactivated	Use nearest neighbor as node values.

Total Pressure Coefficient Field Function Properties

Dimensions	The physical dimensions of this field function (read-only).
Reference Density	The reference density, which can be specified with a text entry.
Reference Pressure	The reference pressure, which can be specified with a text entry.
Reference Velocity	The reference velocity, which can be specified with a text entry.
Type	The type of field function (scalar; read-only).

Function Name	The name of this field function (read-only). Use this property in the definition of other field function definitions, not the node name. Use this property to remind you of the actual field function if you have renamed the node in the tree.

Expert Properties

Inverse Distance Weight	Whether to use inverse distance weighting to interpolate node values.	
	Activated	Interpolate node values using inverse distance weighting.
	Deactivated	Use nearest neighbor as node values.

Engineering Units

STAR-CCM+ has flexible and customizable support for engineering units. The standard SI and USCS systems are built in, and you can add new units as desired. Physical quantities can be specified and displayed in any of these units. Individual quantities can be specified with new units on the fly by simply typing something like “80 mph”.

A unit is a reference amount of a physical quantity used to measure quantities of the same dimensions. Each unit is represented by a node under **Tools** > **Units**.

A unit has an associated set of dimensions. For example, a meter has dimensions of length. A foot, inch, and millimeter also have dimensions of length. Any quantity with dimensions of length can therefore be expressed in meters, feet, inches, and millimeters.

The fundamental dimensions used in STAR-CCM+ are:

- Length
- Time
- Current
- Mass
- Temperature
- Luminosity
- Quantity
- Angle

and the derived dimensions are:

- Force
- Energy
- Power
- Pressure
- Velocity
- Angular velocity
- Volume
- Moment
- Stress

Every physical quantity has a set of dimensions which are one of these dimensions, or a combination of these basic dimensions. For example gravity has dimensions of acceleration, which is dimensions of length per time squared.

Contents:

Working with Units
Using Units While Setting Values
Units Reference

Working with Units

This section explains the following:

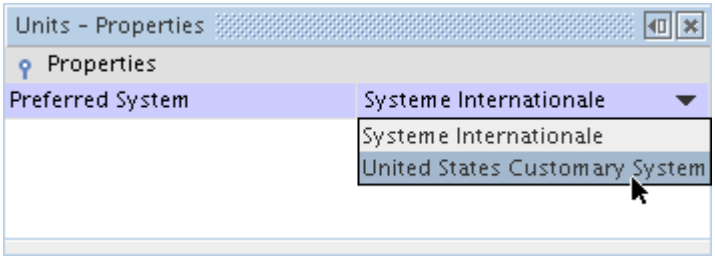
- selecting a standard system
- specifying the default units
- defining your own units
- entering mixed units
- copying the properties of units

Selecting a Standard System

There are two primary units systems available for selection in STAR-CCM+:

- Systeme Internationale (SI), which includes meter, kilogram and second; and
- United States Common System (USCS), which includes foot, pound and second.

To set the SI or USCS systems as your preferred system, simply select it in the **Units** node in the **Tools** node:



What happens when you change the system of units is that STAR-CCM+ runs through all the units and sets the preferred ones to be the SI or USCS selection.

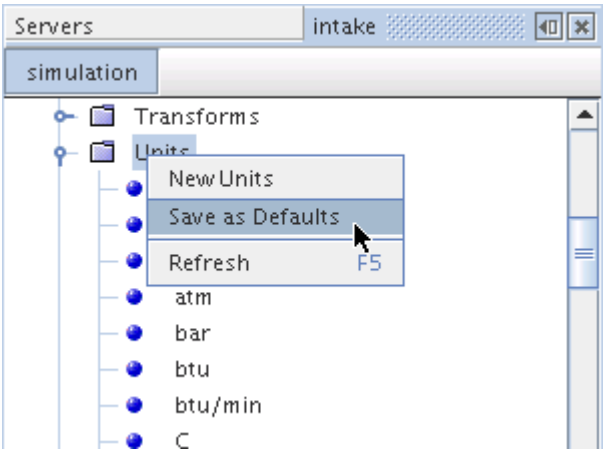
Keep in mind that any time a new preferred unit is selected it will reset the units everywhere in the system to that new value. Thus, it will reset any manual overrides you may have set in specific quantities.

Specifying the Default Units

You can specify which system and unit STAR-CCM+ uses as its default for every new simulation.

To specify the default units:

1. Either:
   - Select a standard system. See Selecting a Standard System.
   - Set a preferred unit. See Unit Node Properties.
2. Right-click the **Units** manager node in the **Tools** node and select **Save as Defaults**.



Your defaults will be applied when a new simulation is created.

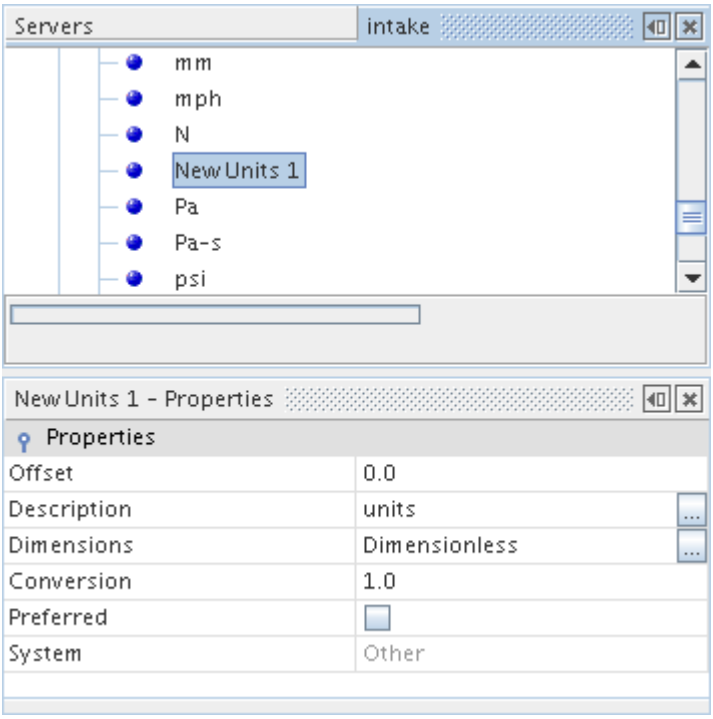
Defining Your Own Units

It is easy to create a new custom unit. You will give this unit a certain set of dimensions and conversion information, and then you can use the unit anywhere a quantity with those dimensions is used.

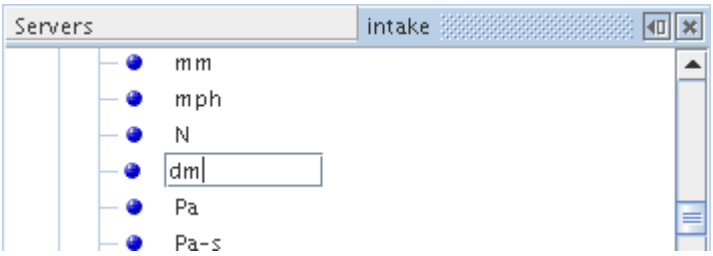
To define you own units:

1. Right-click the **Tools** > **Units** node and select **New**.

   This will result in a new blank unit:

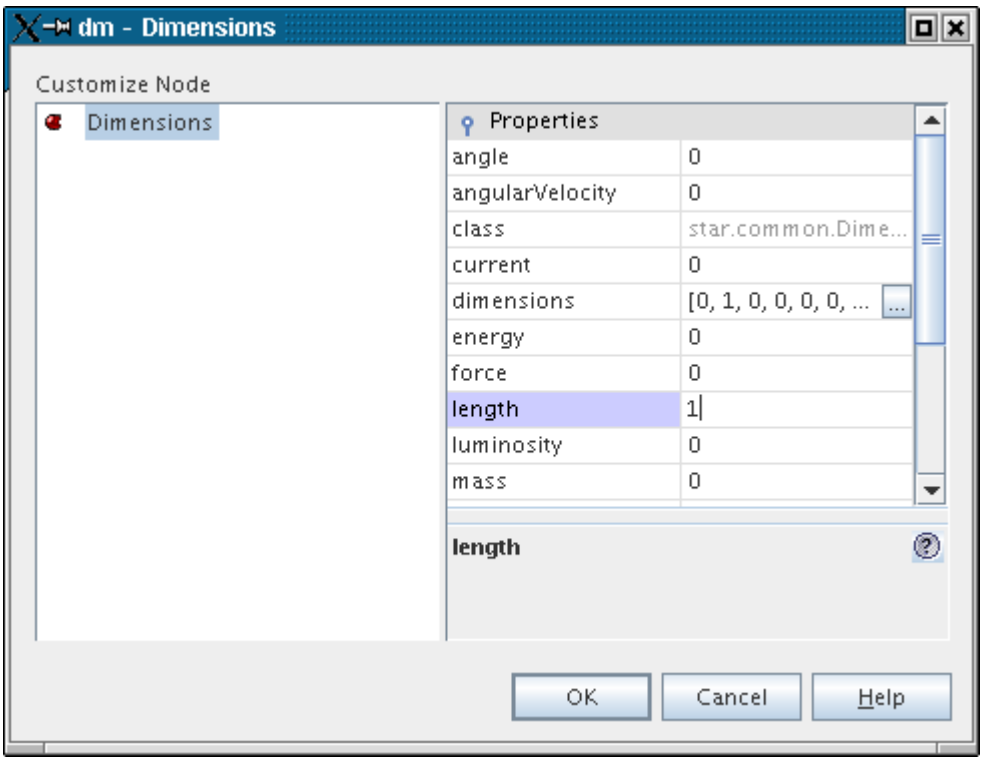


2. Rename the unit.

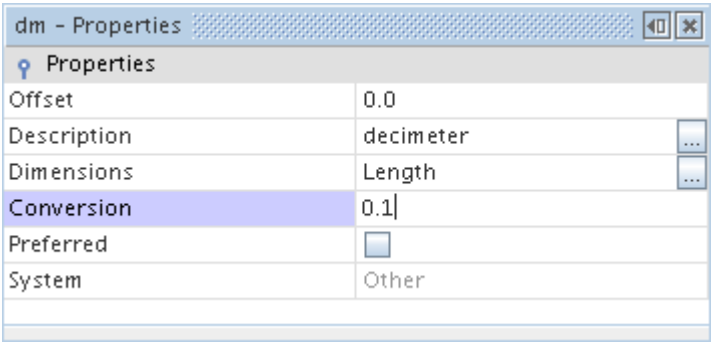


3. Click the property customizer button ( ) of the *Dimensions* property to set the dimensions.

   In this case the dimensions are length, so the length dimension exponent should be set to 1.



4. Click **OK**
5. Set the *Conversion* and *Description* properties for the new unit:



You can now use this unit anywhere a quantity with dimensions of length is required.

Entering Mixed Units

You can enter mixed unit strings into coordinate editor panel fields or properties.

Including commas between items is irrelevant, as long as there are spaces separating each item from its neighbor. For example:

- `[2, 0.2222, 6] in yd ft`
- `[3 2.4 0.001] in cm m`

Copying Properties of Units

When properties are copied from one unit to another, all of the properties other than *System* are applied.

It is not possible to copy properties between built-in and user-defined units.

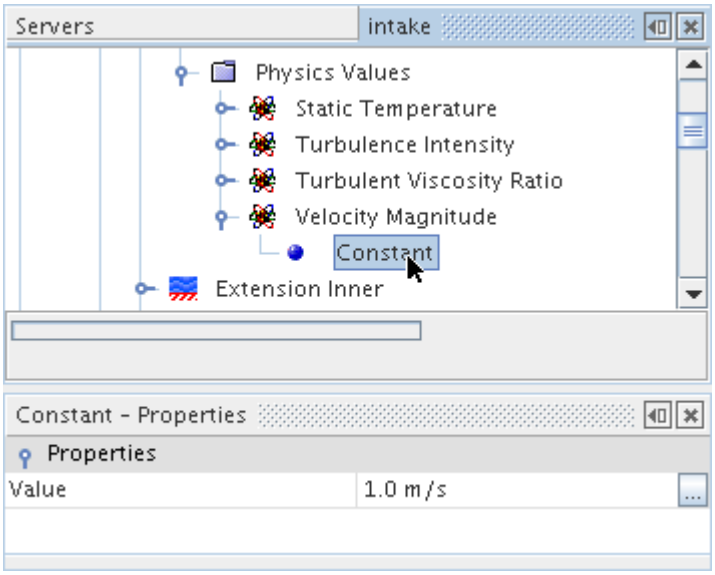
Using Units While Setting Values

This section explains techniques for adjusting units at the same time that you change values:

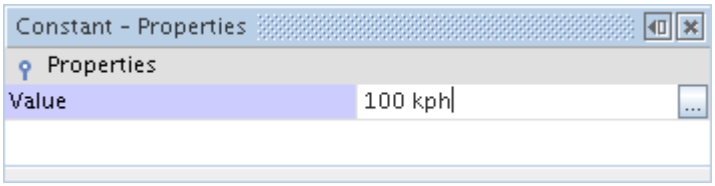
- Using the Keyboard to Specify Different Units
- Using the Keyboard to Convert Units
- Using the Editor to Convert Units

Using the Keyboard to Specify Different Units

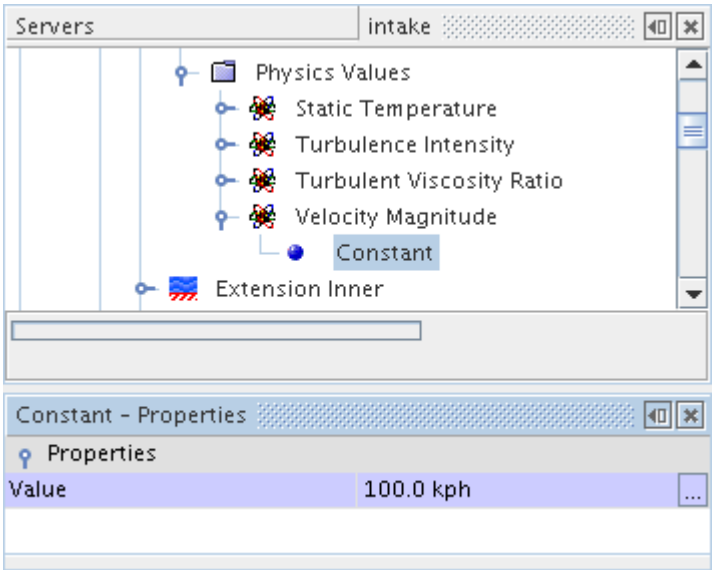
You can specify a constant in non-default units simply by typing the units along with the number. For example, if you use the default SI unit system, the units for velocity will be meters per second:



If you want to change it to a velocity of 100 kilometers per hour, simply type 100 kph into the Value field:

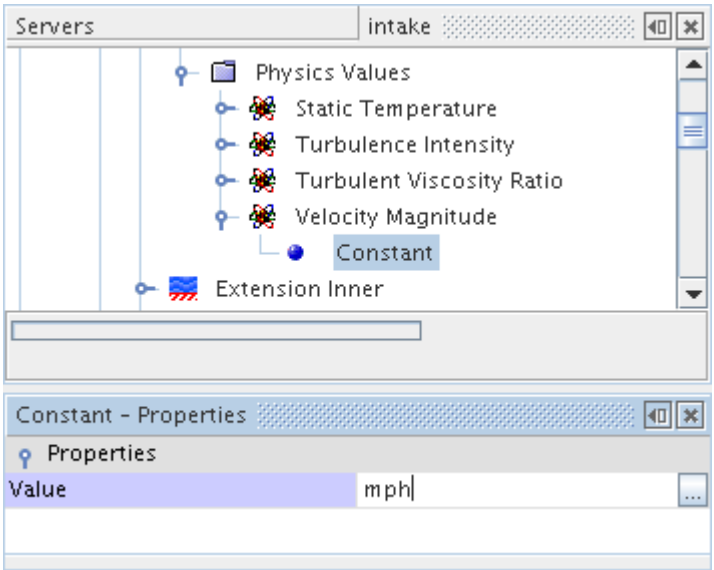


and press <`Enter`>:

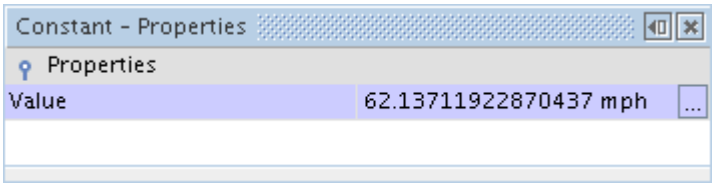


Using the Keyboard to Convert Units

You can convert units in an entry by typing the new units without typing a number. Following from the example above, let's say we wanted to convert the 100 kilometers per hour to miles per hour. In this case simply type mph in the Value field:



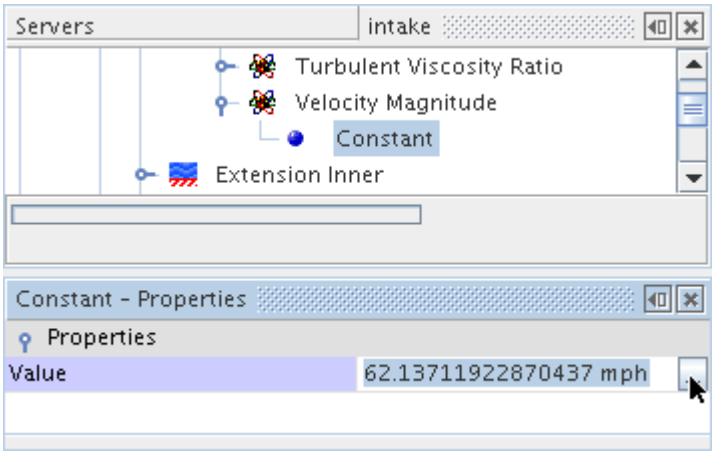
and hit <`Return`>:



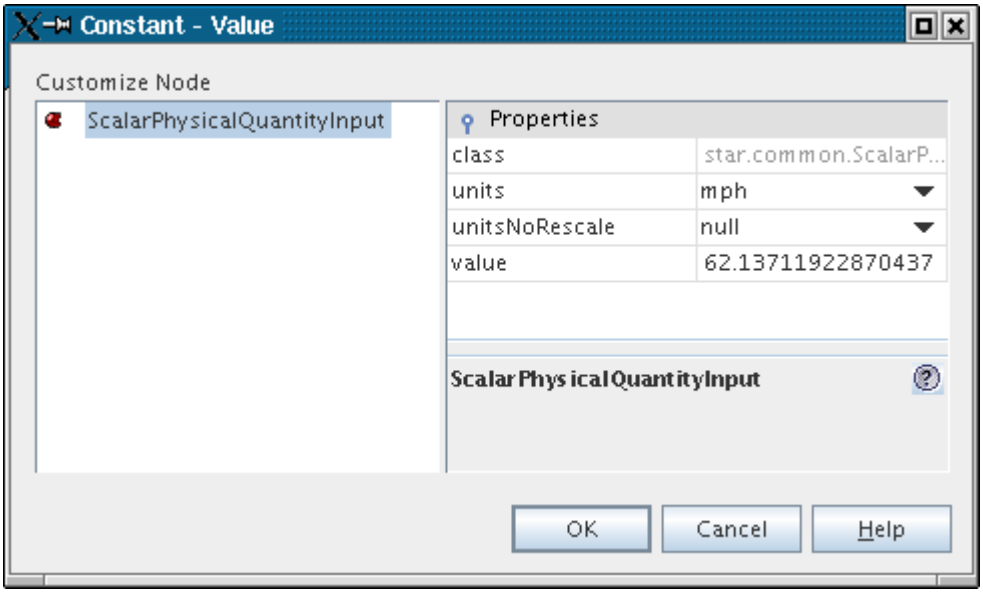
The velocity value is converted from 100 kph to 62.137 mph.

Using the Editor to Convert Units

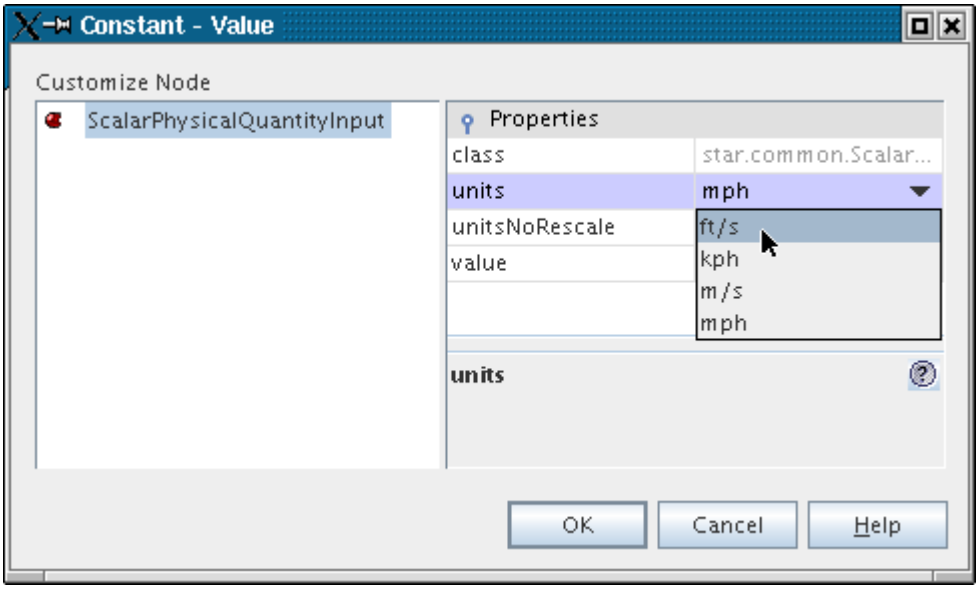
You can also convert units using the graphics editor. Continuing with the example above, let's convert back feet per second. Select the Value field and press the property customizer ( ):



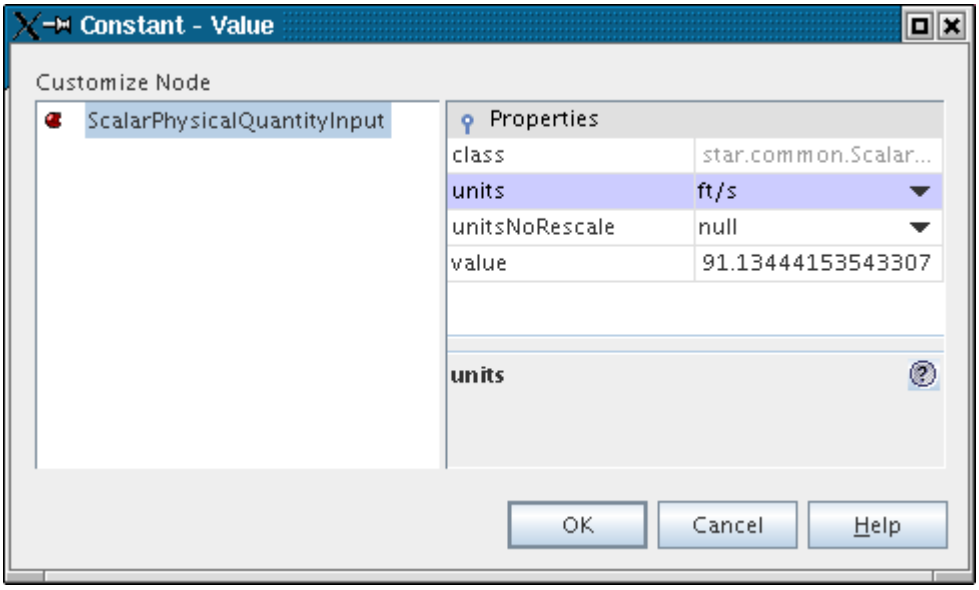
This will open up the editor for the quantity:



For the Units field, just select feet per second:



The velocity value will change from 62 miles per hour to 91 feet per second:



You can then press **OK** to accept or **Cancel** if you do not want to save the change.

Units Reference

Units Node Properties

Preferred System	Selects a preferred system of units.	
	Systeme Internationale	Includes kilogram, meter, Pascal, Newton, Joule, Watt, second.
	United States Customary System	Includes pound, foot, second.

Units Node Action Menu

New	Executes the create units action to add a user-defined unit type to the **Units** node. See Defining Your Own Units.
Save as Default	Executes the save as default action so that all new simulations use the specified unit settings. See Specifying the Default Units.

Unit Node Properties

Conversion	The conversion factor that converts the quantity from this unit to SI units. Read-only except for user-defined units.	
Description	The full name of the unit. Read-only except for user-defined units.	
Dimensions	The physical dimensions of the units. Read-only except for user-defined units.	
Offset	The offset factor used to convert the quantity from this unit to SI units. Used for Celsius and Fahrenheit temperatures. Read-only except for user-defined units.	
Preferred	For each dimensional combination in use in the simulation, STAR-CCM+ marks exactly one unit to be the preferred unit for that dimension. For example the preferred unit for quantities with the dimension of length might be meters and the preferred unit for quantities with dimensions of length per time might be kph (kilometers per hour). The preferred units are the ones that are used by default for any new physical quantity. You can change the preferred unit for an individual dimensional combination, and you can change the preferred units for all quantities to a standard system. Although the preferred unit is used by default, you can override it at any time you like in specific situation. For example, you might have the preferred unit for dimensions of velocity to be the SI standard of meters per second, but for a specific velocity vector display you could change the units to kilometers per hour. Note: Field functions default to SI units, without using to the *Preferred* property.	
	Ticked	Makes this unit the preferred one for these dimensions.

	Cleared	Causes the preferred unit for this dimension to revert back to the default for the current preferred system (SI or USCS).
System	The systems (SI, USCS, or Other) to which this unit belongs. “s” (seconds), for example, is part of both SI and USCS. User-defined units belong to “Other”.	

Scripting the Application

Scripting allows you to work more efficiently by automating repetitive tasks.

In many situations, a java file is not needed. If the properties in your simulation file are correctly set, many batch simulations can be run using the default operations with no java file:

```
starccm+ -batch [SIMULATION]
```

For example, reports, auto-save, auto-export, and scene hardcopies can all be specified in the simulation properties. You would need a macro to customize extensively, or to automate the setting of the relevant properties so they are the same across simulations.

What Is a Java Macro?

A STAR-CCM+ macro is a Java program that is compiled and executed within the STAR-CCM+ workspace.

The Java SDK (Software Development Kit) is required for this key reason - to compile the Java macros. Unlike other macro systems, STAR-CCM+ does not record every mouse movement, button click, and keystroke, but rather it translates your actions from the workspace into meaningful Java code.

The macros that you write are standard Java so you have access to all the programming constructs of that language, such as loops and conditional constructs. In addition, the STAR-CCM+ server exposes a number of objects that you instantiate and manipulate to carry out the sequence of tasks that you require. You can activate a macro with the `Play Macro...` command.

You can write a macro from scratch, but that would require up-front knowledge about all the objects, attributes, and methods that the server exposes. It is more effective to use the workspace to record the actions you want to perform. Then you can edit the Java file using your favorite text editor to get the exact effect you require. You can then run the macro as often as you need.

The macro is just a Java program, so it does not compile if there are syntax errors. In this instance, error messages are displayed so that you have specific file names, line numbers, and issues to fix.

Macro Files

The macros are stored in a `.java` file that contains a set of tasks for the program to perform.

Macro files are Java language files that can be recorded as a journal of your interaction with the application. While it is most convenient to create the macro files from within a STAR-CCM+ session, they can be manually edited in a text editor. For example, they can be specialized for specific applications using the full power of the Java programming language.

The two most significant macro commands are play and record. As their names suggest, play causes an existing macro file to be executed and record causes the file to be opened for recording. Recording occurs

until the stop recording command is executed. A pause option is also available to preclude certain interactions form being recorded into a given macro file while it is still open for recording.

Contents:

Recording a Macro
Editing the Macro
Playing the Macro
Debugging a Macro Using an IDE
Running Multiple Simulations Consecutively
Playing Macros Within Macros
Adding Custom Macros to the Toolbar
Using the Java API
Working with Default Macros
Macro API Changes in Past Releases

Recording a Macro

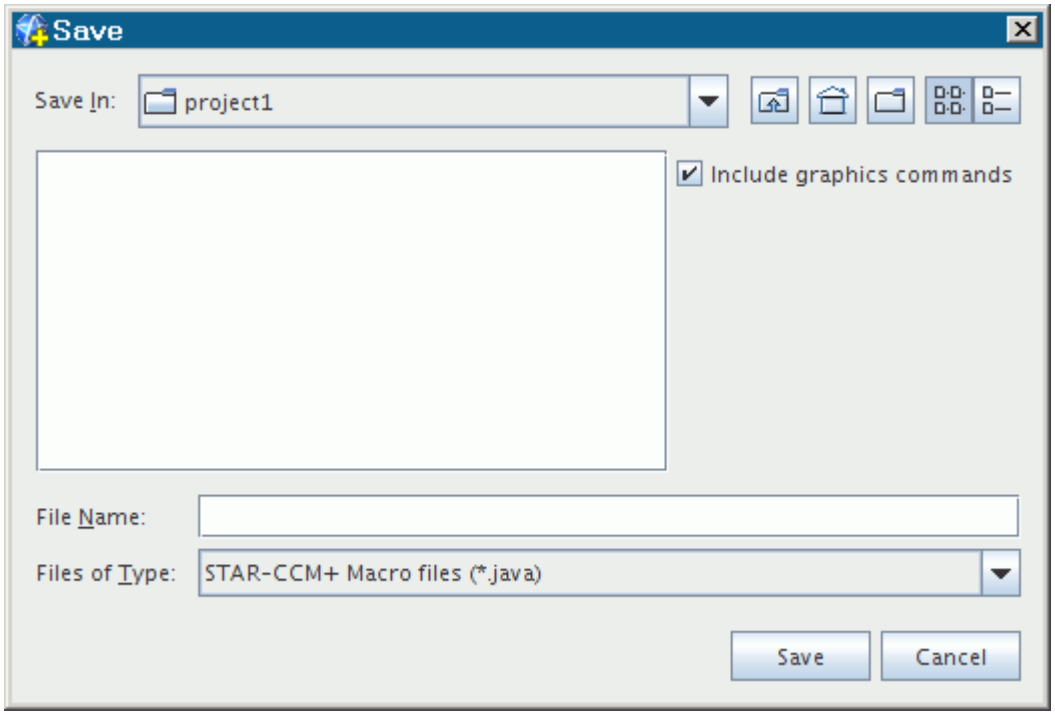
The easiest way to create your Java macro file is to record one in STAR-CCM+ while you perform all the steps that belong in the macro file.

To record a Java macro:

1. Select **File** > **Macro** > **Start Recording...**.

   Alternatively, click the (**Start Recording**) toolbar button.

   This action opens a variant of the standard, operating system-specific Save dialog that is set up for creating macro files.



2. Navigate to the directory where you want to place this macro and give it a name (the `.java` extension is automatically added to the file name).
3. If the purpose of the macro does not involve specific adjustments to graphical elements such as scenes and displayers, deactivate the **Include Graphics Commands** option near the top of the dialog.

   This option excludes graphics code from the macro.

When the macro is recording, each action you perform in STAR-CCM+ adds an associated command to the macro file. You can see which commands are being added to the macro by selecting the macro tab in the *Output* window; the tab has the same name as the macro file. If you wish to perform some steps in the simulation, but do not want them to be included in the macro, it is possible to pause the recording. When you are ready to record more steps for the macro, you can resume recording. The macro is saved when you stop recording.

It is recommended that you plan which steps to include in the Java macro before you begin recording. This planning ensures that the macro is not cluttered with superfluous commands, and is easier to edit.

Pausing and Resuming the Recording

To interrupt recording of the macro:

1. To interrupt recording of the macro, select **File** > **Macro** > **Pause Recording**.
2. Alternatively, click the (**Pause Recording**) toolbar button.

   Actions that you trigger in the workspace are not added to the macro file while recording is paused. The pause icon in the toolbars becomes highlighted ( ) when macro recording is paused. This action is only available if a macro is being recorded.
3. To resume recording, select the same menu item or toolbar button.

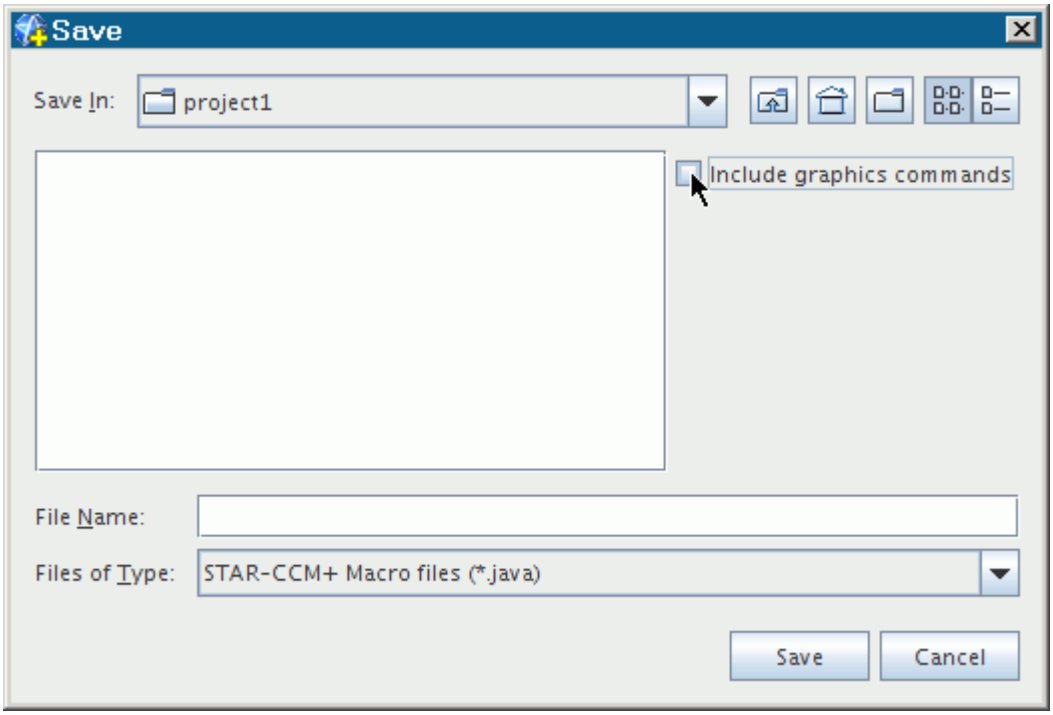
   Alternatively, click the (**Pause Recording**) toolbar button.

   The pause icon in the toolbars reverts to normal ( ) once this action has been triggered.

Excluding Visualization Code from the Macro

By default, macros include code for all adjustments to displayers, scenes, and other graphical elements in the STAR-CCM+ workspace. The purpose of a macro can even be just to add an item like a derived part. This extra code can be inconvenient if you plan to run this macro in batch, or if you want to edit the macro.

To use this feature, deactivate the **Include Graphics Commands** checkbox in the *Save* dialog.



This action causes the macro recording feature to ignore changes to scenes, displayers, view, animation, and part groups for the objects affected.

Saving the Macro

To finish recording and save a macro:

- Select **File** > **Macro** > **Stop Recording**.

  Alternatively, click the (**Stop Recording**) toolbar button.

  This action is only available if a macro recording is in progress or is paused.

  The macro file that you had named in the previous step contains the Java code that maps to the actions you carried out while the recorder was active.

Editing the Macro

The macro file is a regular text file containing Java code that can be edited and customized to suit your task.

For example, if you want to change the number of iterations that the batch job runs from 1 to 1000. Open the `.java` file in your favorite text editor and look for a line that looks something like:

```
simulation_0.getSimulationIterator().step(1);
```

and change the iteration count to whatever you need:

```
simulation_0.getSimulationIterator().step(1000);
```

or better still, use the `run` function:

```
simulation_0.getSimulationIterator().run();
```

Save the file and close it. Some examples of other typical edits are:

- Loading in a different case file
- Adding loops to change properties to simulate a rotor being spun up to speed
- Varying properties as a function of time

Path names in java macros can be made relative to the java file using the `resolvePath` function (see the Java API for details). It is also possible to edit the macro so that it pauses to prompt you for input.

You can also exclude graphics code from the macro.

You can design a macro to prompt you for input at a given point in the simulation process.

This feature gives you the powerful capability of automating some parts of a simulation while having real-time discretion over settings in other parts.

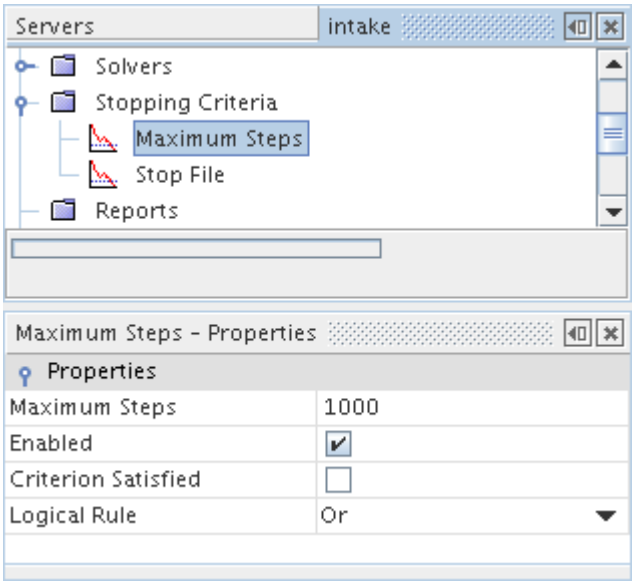
Contents:

Prompting for an Integer Entry
Prompting for a Double-Precision Real Number Entry
Prompting for a Text Entry
Providing an Interactive Dialog

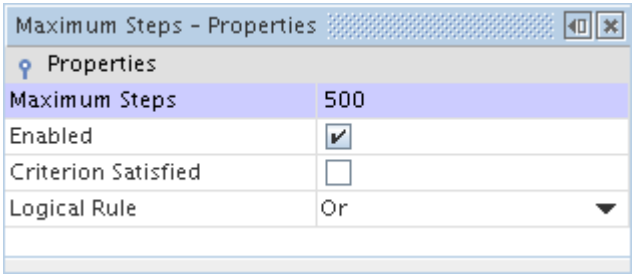
Prompting for an Integer Entry

You can edit the macro to prompt for an integer entry.

Integer entries are used in various STAR-CCM+ features, such as stopping criteria.



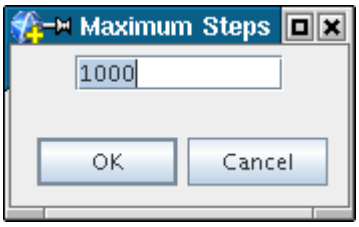
For this example, a simple macro was recorded to change the *Maximum Steps* property of the corresponding stopping criterion from `1000` to `500`.



The macro, named `integer.java`, contains the following code:

```
// STAR-CCM+ macro: integer.java
package macro;
import java.util.*;
import star.common.*;
import star.base.neo.*;
public class integer extends StarMacro {
  public void execute() {
    Simulation simulation_0 =
      getActiveSimulation();
    StepStoppingCriterion stepStoppingCriterion_0 =
      ((StepStoppingCriterion)
simulation_0.getSolverStoppingCriterionManager().
         getSolverStoppingCriterion("Maximum Steps"));
    stepStoppingCriterion_0.setMaximumNumberSteps(500);
  }
}
```

This typical macro code contains the value for the stopping criterion that is entered during recording—it remains constant. However, when the macro contains code for user input, it pauses to activate a dialog in which you can specify that value.



Once you click **OK**, the macro continues.

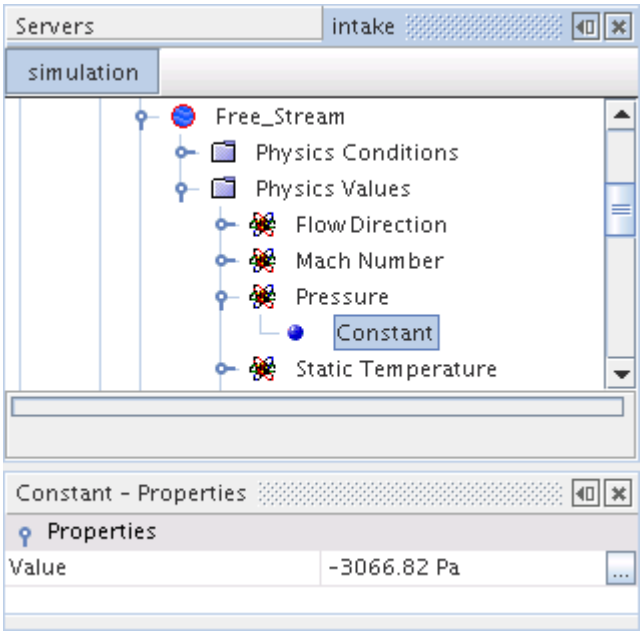
The code of `integer.java` is modified as follows to make this work:

```
// STAR-CCM+ macro: integer.java
package macro;
import java.util.*;
import star.common.*;
import star.base.neo.*;
public class integer extends StarMacro {
  public void execute() {
    Simulation simulation_0 =
      getActiveSimulation();
    StepStoppingCriterion stepStoppingCriterion_0 =
      ((StepStoppingCriterion)
simulation_0.getSolverStoppingCriterionManager().
        getSolverStoppingCriterion("Maximum Steps"));
    int ret = promptUserForInput("Maximum Steps", 1000); /* <---- NEW
CODE */
    stepStoppingCriterion_0.setMaximumNumberSteps(ret); /* <---- "ret"
replaces
         * the numeric entry that was recorded automatically */
  }
}
```

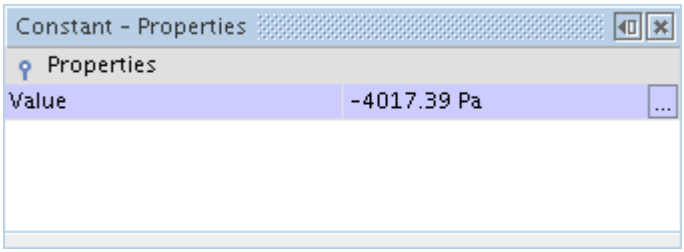
Prompting for a Double-Precision Real Number Entry

You can edit a macro to prompt for a double-precision real number.

Double-precision real numbers are often used for specifying values in STAR-CCM+ such as boundary conditions.



For this example, a simple macro was recorded to change the pressure from `-3066.82` to `-4017.39`.

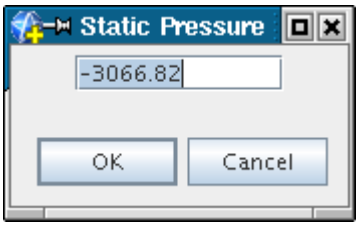


The macro, named `doublePrecision.java`, contains the following code:

```
// STAR-CCM+ macro: doublePrecision.java
package macro;
import java.util.*;
import star.common.*;
import star.base.neo.*;
import star.flow.*;
public class doublePrecision extends StarMacro {
  public void execute() {
    Simulation simulation_0 =
      getActiveSimulation();
    Region region_0 =
      simulation_0.getRegionManager().getRegion("Fluid");
    Boundary boundary_0 =
region_0.getBoundaryManager().getBoundary("Free_Stream");
    StaticPressureProfile staticPressureProfile_0 =
boundary_0.getValues().get(StaticPressureProfile.class);
    ((ConstantScalarProfileMethod)
```

```
staticPressureProfile_0.getMethod()).getQuantity().
      setValue(-4017.39);
  }
}
```

This typical macro code contains the value for the static pressure that is entered during recording—it remains constant. However, when the macro contains code for user input, it pauses to activate a dialog in which you can specify that value.



Once you click **OK**, the macro continues.

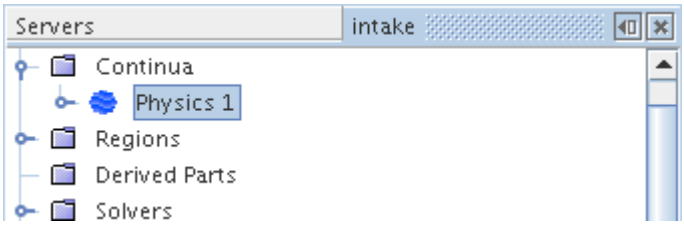
The code of `doublePrecision.java` is modified as follows to make this work:

```
// STAR-CCM+ macro: doublePrecision.java
package macro;
import java.util.*;
import star.common.*;
import star.base.neo.*;
import star.flow.*;
public class doublePrecision extends StarMacro {
  public void execute() {
    Simulation simulation_0 =
      getActiveSimulation();
    Region region_0 =
      simulation_0.getRegionManager().getRegion("Fluid");
    Boundary boundary_0 =
region_0.getBoundaryManager().getBoundary("Free_Stream");
    StaticPressureProfile staticPressureProfile_0 =
boundary_0.getValues().get(StaticPressureProfile.class);
    double ret = promptUserForInput("Static Pressure", -3066.82); /*
<---NEW CODE */
    ((ConstantScalarProfileMethod)
staticPressureProfile_0.getMethod()).getQuantity().setValue(ret); /*
<---- "ret"
         * replaces the numeric entry that was recorded automatically
*/
  }
}
```

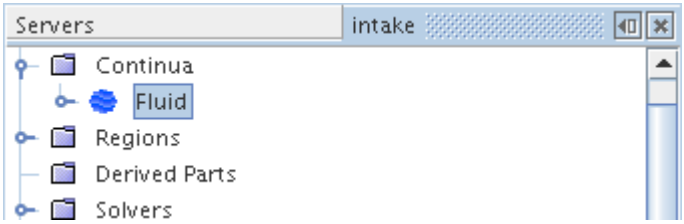
Prompting for a Text Entry

You can edit a macro to prompt for a text entry.

Text entries (strings) are used for various purposes in a STAR-CCM+ simulation, such as renaming objects.



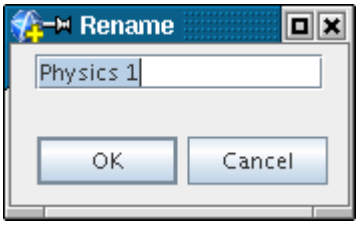
For this example, a simple macro was recorded to change the name of a continuum from `Physics 1` to `Fluid`.



The macro, named `string.java`, contains the following code:

```
// STAR-CCM+ macro: string.java
package macro;
import java.util.*;
import star.common.*;
import star.base.neo.*;
public class string extends StarMacro {
  public void execute() {
    Simulation simulation_3 =
      getActiveSimulation();
    PhysicsContinuum physicsContinuum_0 =
      ((PhysicsContinuum) simulation_3.getContinuumManager().
          getContinuum("Physics 1"));
    physicsContinuum_0.setPresentationName("Fluid");
  }
}
```

This typical macro code contains the value for the stopping criterion that is entered during recording—it remains constant. However, when the macro contains code for user input, it pauses to activate a dialog in which you can specify that value.



Once you click **OK**, the macro continues.

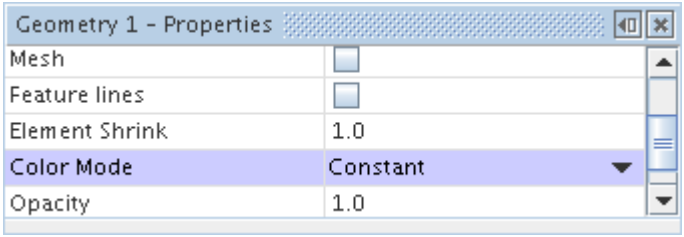
The code of `string.java` is modified as follows to make this work:

```
// STAR-CCM+ macro: string.java
package macro;
import java.util.*;
import star.common.*;
import star.base.neo.*;
public class string extends StarMacro {
  public void execute() {
    Simulation simulation_3 =
      getActiveSimulation();
    PhysicsContinuum physicsContinuum_0 =
      ((PhysicsContinuum) simulation_3.getContinuumManager().
         getContinuum("Physics 1"));
    String ret = promptUserForInput("Rename", "Physics 1"); /* <----
NEW CODE */
    physicsContinuum_0.setPresentationName(ret); /* <---- "ret"
replaces
         * the text entry that was recorded automatically */
  }
}
```

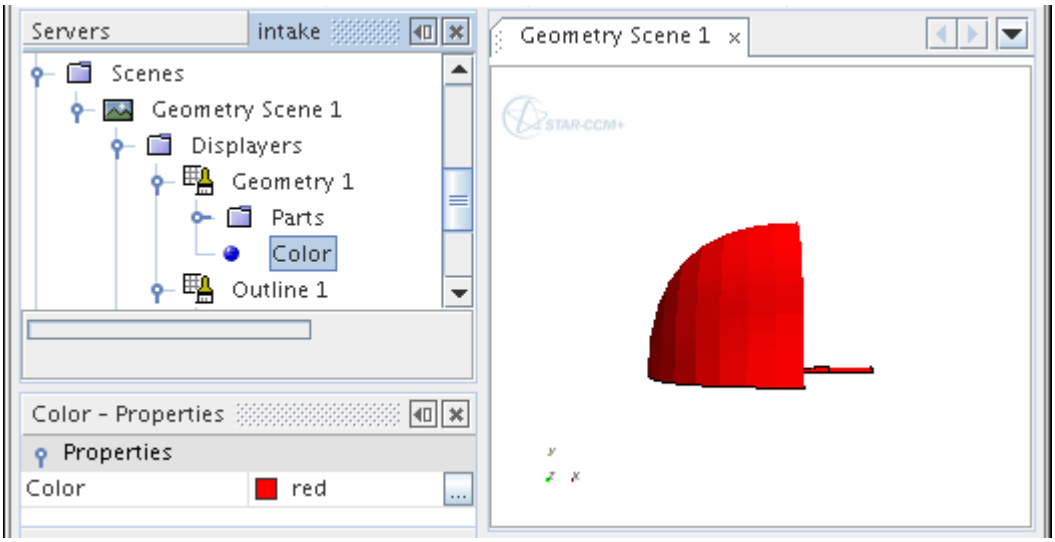
Providing an Interactive Dialog

You can edit a macro to provide an interactive dialog.

STAR-CCM+ offers a great many visualization controls that let you customize the appearance of the displays in your simulation. Settings for these controls are captured in the recording of macros. For example, a simple macro was created to add a geometry scene to a simulation and change the *Color Mode* property of the geometry displayer to `Constant`.



The next step in editing the macro involves changing the constant color to red.



The macro, named `partcolor.java`, contains the following code:

```
// STAR-CCM+ macro: partcolor.java
package macro;
import java.util.*;
import star.common.*;
import star.base.neo.*;
import star.vis.*;
public class partcolor extends StarMacro {
  public void execute() {
    Simulation simulation_1 =
      getActiveSimulation();

simulation_1.getSceneManager().createGeometrySceneAndWait("Geometry
Scene", "Outline", "Geometry", 1);
    Scene scene_1 =
      simulation_1.getSceneManager().getScene("Geometry Scene 1");
    scene_1.initializeAndWait();
    PartDisplayer partDisplayer_3 =
      ((PartDisplayer)
scene_1.getDisplayerManager().getDisplayer("Outline 1"));
    partDisplayer_3.initialize();
    PartDisplayer partDisplayer_4 =
      ((PartDisplayer)
scene_1.getDisplayerManager().getDisplayer("Geometry 1"));
    partDisplayer_4.initialize();
    PartDisplayer partDisplayer_5 =
      ((PartDisplayer) scene_1.getHighlightDisplayer());
```

```
    partDisplayer_5.initialize();
    CurrentView currentView_1 =
      scene_1.getCurrentView();
    currentView_1.setInput(new DoubleVector(new double[]
{-0.5333999395370483, 1.268258864962263, 0.0}), new DoubleVector(new
double[] {-0.5333999395370483, 1.268258864962263,
14.471860805953908}), new DoubleVector(new double[] {0.0, 1.0, 0.0}),
-1.0, 0);
    partDisplayer_4.setColorMode(1);
    partDisplayer_4.setDisplayerColor(new DoubleVector(new double[]
{1.0, 0.0, 0.0}));
  }
}
```

This typical macro code contains the one color selection that is made during recording—it remains the same. However, when the macro contains code for user input, it pauses to activate a dialog in which you can make your own selection.

Once you click **OK**, the macro continues.

To make this work, the code of `partcolor.java` has extensive new and modified code, as follows:

```
// STAR-CCM+ macro: partcolor.java
package macro;
import java.util.*;
import star.common.*;
import star.base.neo.*;
import star.vis.*;
import java.io.File;
import java.awt.*;
import javax.swing.*;
import star.base.neo.NeoMacro;
import star.base.neo.NeoScriptFile;
```

```
import star.base.neo.NeoDoubleField;
import star.base.neo.NeoIntegerField;
import org.openide.DialogDescriptor;
import org.openide.NotifyDescriptor;
import org.openide.DialogDisplayer;
public class partcolor extends StarMacro {
    public boolean promptUserForInput(String title, JComponent panel) {
        if (System.getProperty("java.awt.headless") == null) {
            if (title == "") {
                title = "Input values";
            }
            DialogDescriptor dd = new DialogDescriptor(panel, title,
true, NotifyDescriptor.OK_CANCEL_OPTION, NotifyDescriptor.OK_OPTION,
null);
            return DialogDisplayer.getDefault().
                notify(dd) == NotifyDescriptor.OK_OPTION;
        }
        return false;
    }
    public class ColorChoice extends JPanel {
        ButtonGroup group;
        String[] labels = { "Red", "Green", "Blue", "Black", "White",
"Gray", "Yellow" };
        Color[] colors = { Color.RED, Color.GREEN, Color.BLUE,
Color.BLACK, Color.WHITE, Color.GRAY, Color.YELLOW };
        public ColorChoice() {
            super(new GridLayout(0, 1));
            setBackground(Color.lightGray);
            setBorder(BorderFactory.
                createTitledBorder("Colors"));
            group = new ButtonGroup();
            JRadioButton option;
            for (int i = 0; i < labels.length; i++) {
                option = new JRadioButton(labels[i]);
                if(i==0) option.setSelected(true);
                option.setForeground(colors[i]);
                group.add(option);
                add(option);
            }
        }
        public Color getColor() {
            Enumeration elements = group.getElements();
            while (elements.hasMoreElements()) {
                AbstractButton button =
(AbstractButton)elements.nextElement();
                if (button.isSelected()) {
                    String btext=button.getText();
                    for (int i = 0;i < labels.length; i++) {
```

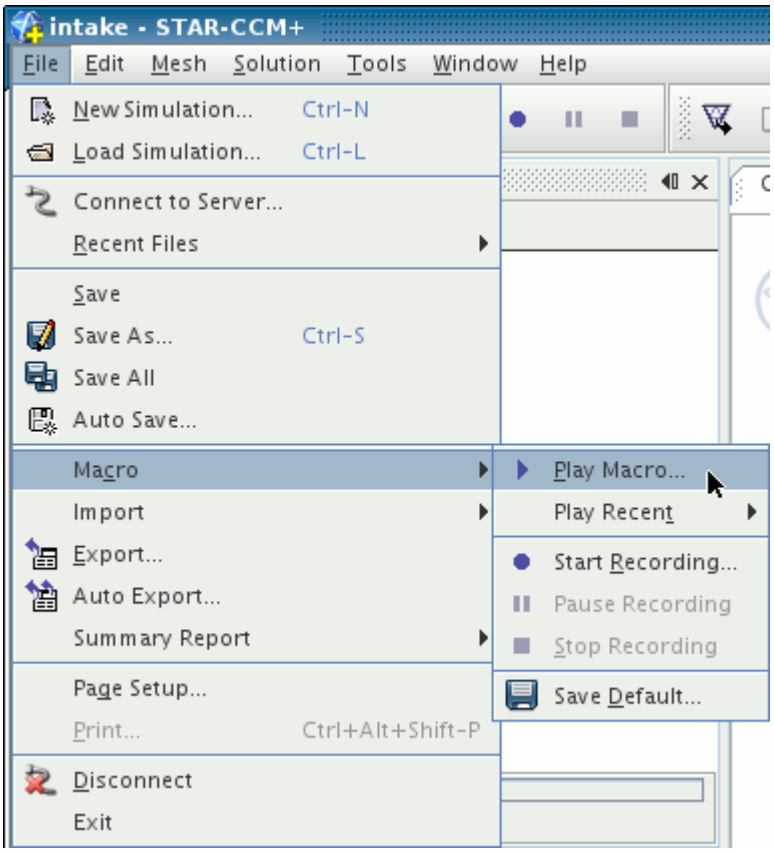
```
                        if(btext==labels[i]) return colors[i];
                    }
                }
            }
            return Color.RED;
        }
    }
    public void execute() {
        Simulation simulation_3 =
          getActiveSimulation();
simulation_3.getSceneManager().createGeometrySceneAndWait("Geometry
Scene", "Outline", "Geometry", 1);
        Scene scene_3 =
simulation_3.getSceneManager().getScene("Geometry Scene 1");
        scene_3.initializeAndWait();
        PartDisplayer partDisplayer_9 =
          ((PartDisplayer)
scene_3.getDisplayerManager().getDisplayer("Outline 1"));
        partDisplayer_9.initialize();
        PartDisplayer partDisplayer_10 =
          ((PartDisplayer)
scene_3.getDisplayerManager().getDisplayer("Geometry 1"));
        partDisplayer_10.initialize();
        PartDisplayer partDisplayer_11 =
          ((PartDisplayer) scene_3.getHighlightDisplayer());
        partDisplayer_11.initialize();
        CurrentView currentView_3 =
          scene_3.getCurrentView();
        currentView_3.setInput(new DoubleVector(new double[]
{-0.5333999395370483, 1.268258864962263, 0.0}), new DoubleVector(new
double[] {-0.5333999395370483, 1.268258864962263,
14.471860805953908}), new DoubleVector(new double[] {0.0, 1.0, 0.0}),
-1.0, 0);
        partDisplayer_10.setColorMode(1);
        ColorChoice p = new ColorChoice();
        if (promptUserForInput("Set Color", p)) {
        Color mycolor = p.getColor();
        partDisplayer_10.setDisplayerColorColor(mycolor);
        }
    }
}
```

Playing the Macro

A macro can be played from the workspace and stopped or, maybe more typically, can be run in batch mode. It is also possible to replay recent macros.

Running a Macro from the Workspace

To replay a macro from the workspace, select **File** > **Macro** > **Play Macro** from the menus.



Alternatively, click the (**Play Macro**) toolbar button.

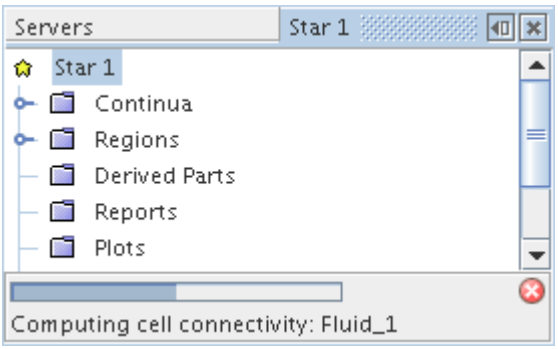
This action opens the operating system-specific standard file open dialog, with the **Files of Type** drop-down list is automatically set to `.java` which allows you to select a Java macro file. Once you click **Open** in the dialog, STAR-CCM+ attempts to compile the Java code, and then run it.

In addition, you can use the command line to open the STAR-CCM+ workspace, start a new simulation, and play a macro:

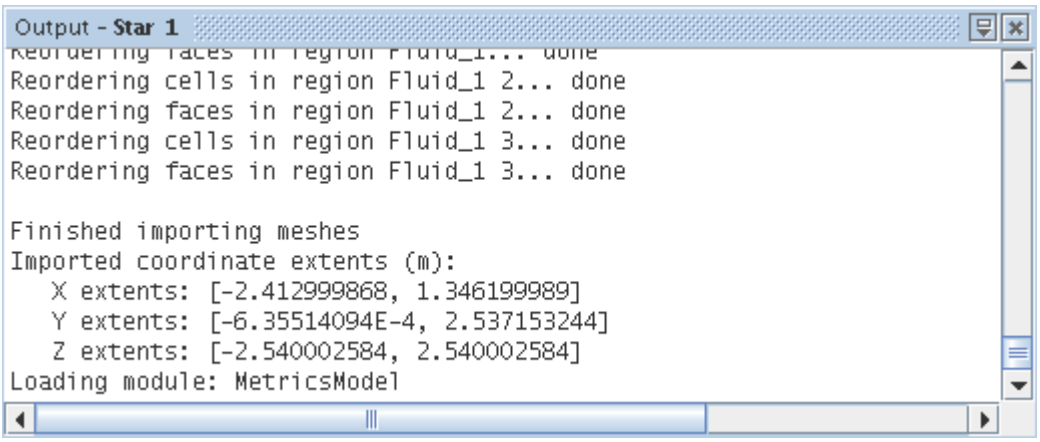
```
% starccm+ -new -m macro.java
```

Progress is reported in different ways:

- In the workspace status bar:



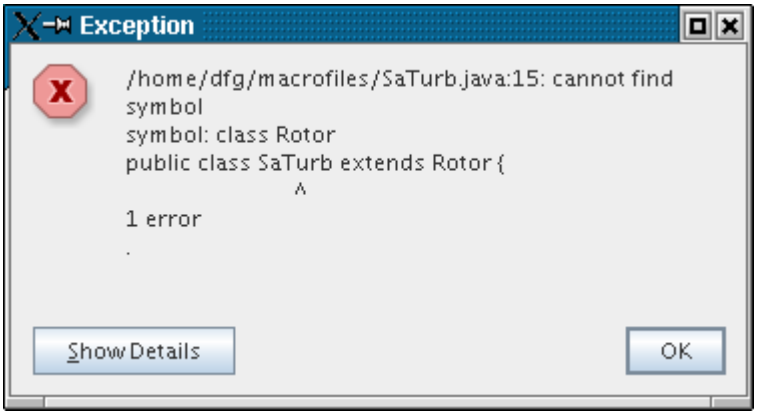
- In the *Output* window (including any output that manual inputs generate):



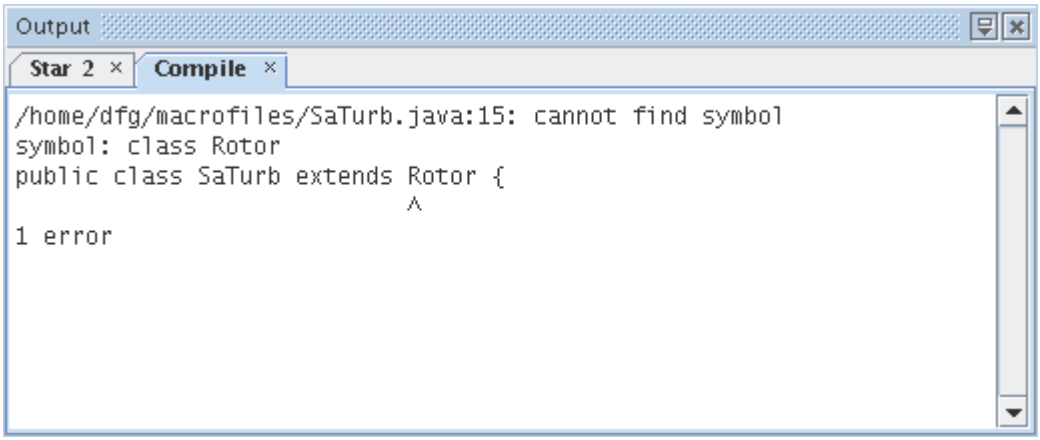
If the compilation does not work, you get errors reported in a dialog and in the *Output* window. These errors must be corrected before the script can run.

Resolving Errors

If there are any compile errors, you receive an exception error dialog:



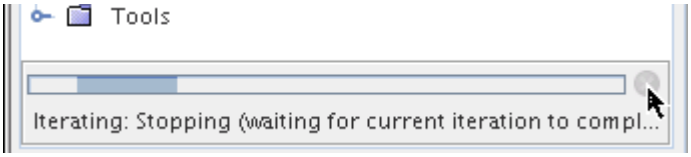
Click **OK**. You can see the error messages reported in the *Output* window under a new tab that is labeled `Compile`.



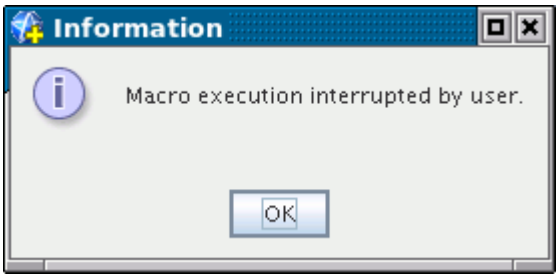
Use this information and edit the file until there are no further errors. In the example above, there is no such class called `Rotor`.

Stopping the Macro

You can stop the playing of a macro by clicking the (**Cancel Macro**) button in the workspace status bar.



STAR-CCM+ stops the operations and shows acknowledgement by displaying an information dialog.



Contents:

Running the Macro in Batch Mode
Running Using the Recent Macros List

Running the Macro in Batch Mode

You can run a macro in batch mode.

Batch mode for macros is run from the command line using the `-batch` option:

```
% starccm+ -batch cavity.java file.sim
```

This option runs STAR-CCM+ with a script (`cavity.java` in the above example). The `-batch` option ensures that no GUI is displayed. If you do not want the server process to exit once the batch file has completed, then add the `-noexit` option to the command line. For example, maybe you want to connect to it to analyze results. If you wish to monitor the progress of a batch run, an additional client can access the server process interactively.

Alternatively, you can start a new simulation and run a macro in batch mode:

```
% starccm+ -new -batch macro.java
```

The following command-line arguments can be used in a batch run macro:

- Pass in property definitions:
  - On Linux: `-jvmargs -Dmyprop=myvalue`
  - On Windows: `-J-Dmyprop=myvalue`
- Include `System.getProperty("myprop")` in the macro.

Contents:

Adding Arguments to a Macro in Batch Mode

Adding Arguments to a Macro in Batch Mode

You can use command-line arguments in a batch run macro.

The following command-line arguments can be used in a batch run macro:

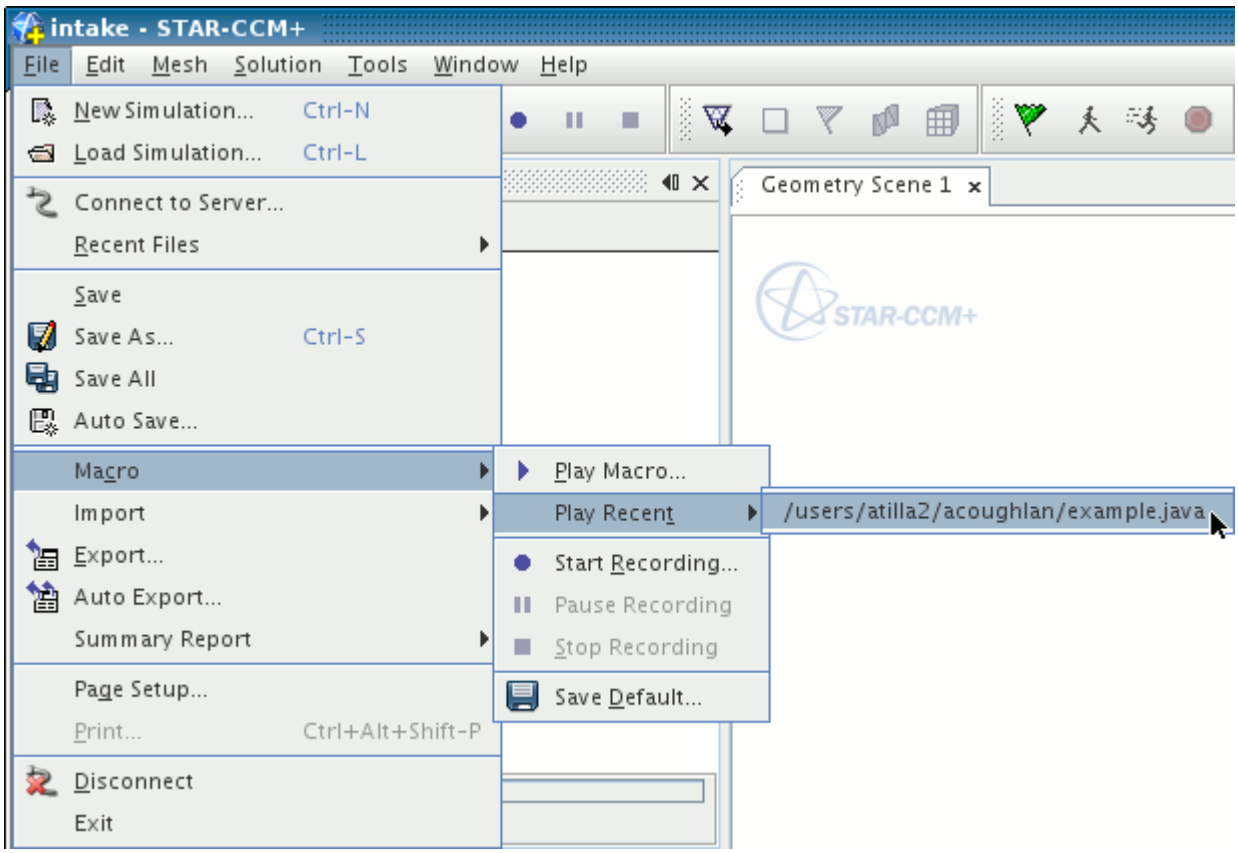
- Pass in property definitions:
  - On Linux: `-jvmargs -Dmyprop=myvalue`
  - On Windows: `-J-Dmyprop=myvalue`
- Include `System.getProperty("myprop")` in the macro.

Running Using the Recent Macros List

STAR-CCM+ keeps a record of macros that were run previously in any simulation and makes them available in a list.

To replay a recently run macro:

1. Click the **File** > **Macro** > **Play Recent** menu item to display the list of recently run macros.
2. Select the macro that you want to play.



The selected macro begins running immediately.

You can access the recent macros from the toolbar once you make the associated toolbar button available.

3. In the STAR-CCM+ workspace, add the (**Play Recent Macro**) button to the **Macro** toolbar. See Customizing a Toolbar.
4. Click the button.

   The same list of recently run macros appears as the list from the **File** > **Macro** > **Play Recent** menu item.

Debugging a Macro Using an IDE

You can debug a STAR-CCM+ macro using a special command-line option and an Integrated Development Environment (IDE) program.

There are several IDEs available for free download including the following:

- BlueJ: www.bluej.org
- Eclipse: www.eclipse.org
- JDeveloper: www.oracle.com/technetwork/developer-tools/jdev
- NetBeans: www.netbeans.org

The instructions below are for debugging a macro using NetBeans 7.1. Similar techniques can be used for other IDEs.

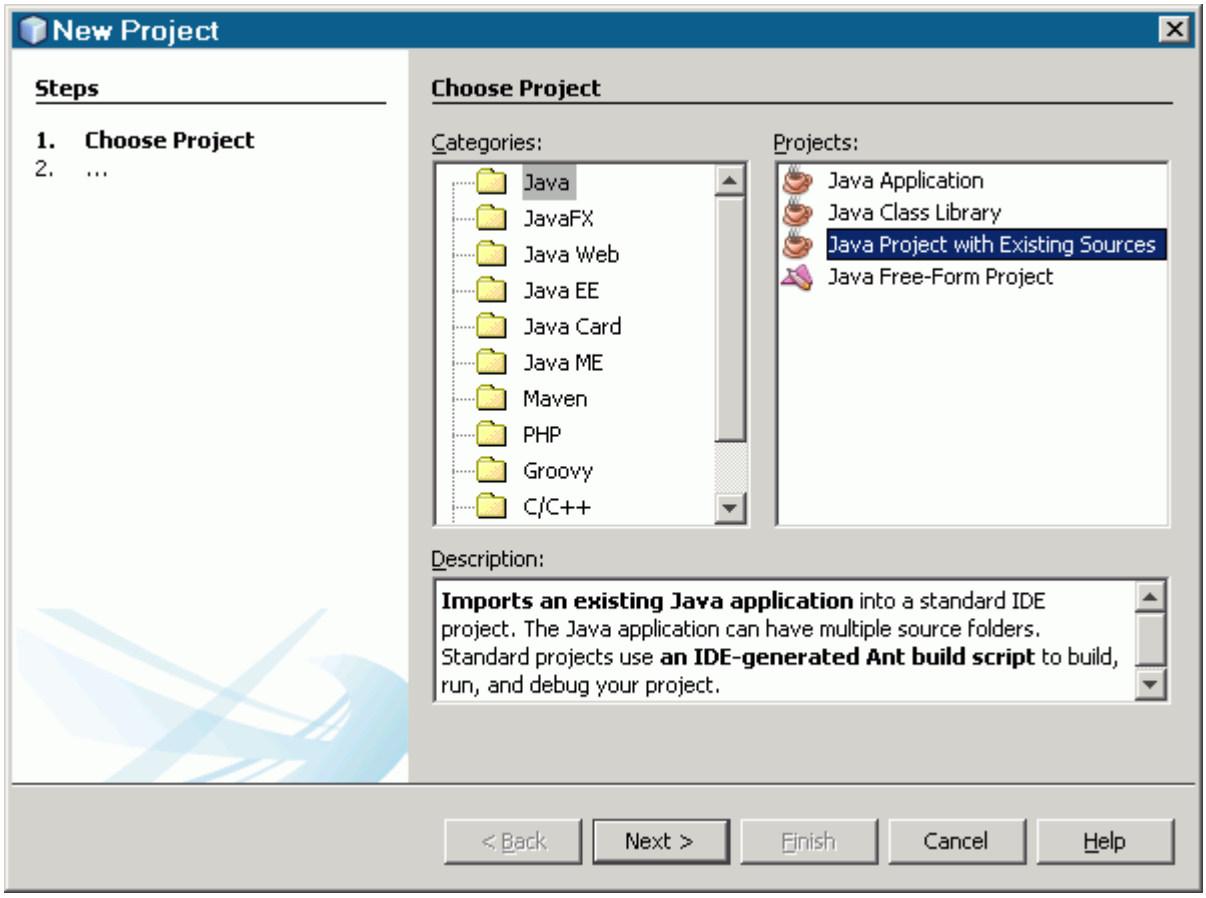
The process for debugging a macro involves the following steps:

1. Loading the macro in the IDE
2. Running STAR-CCM+ in debugging mode
3. Stepping through the macro

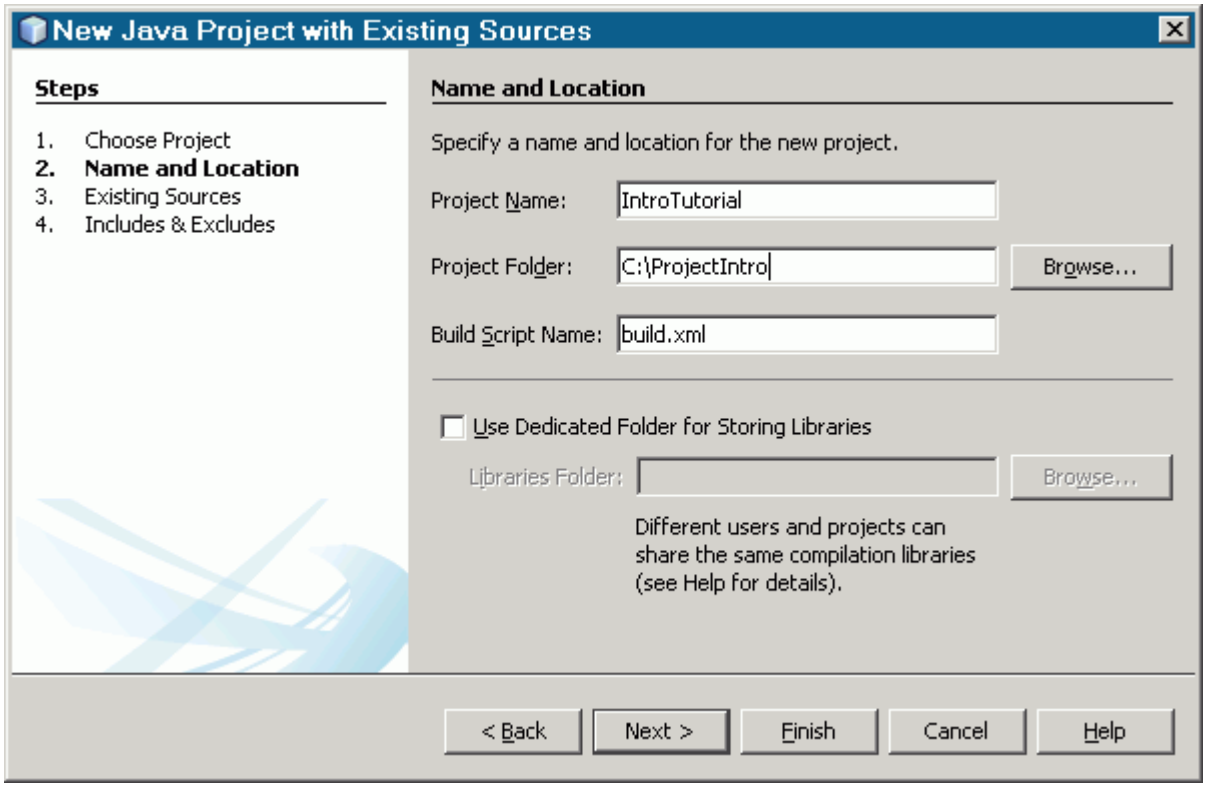
Loading the Macro in the IDE

NetBeans requires the macro to be in a directory structure `<ProjectFolder>/<SourceFolder>/<Macro>.java`, so you must organize the macro and any input files to be in the appropriate locations before starting.

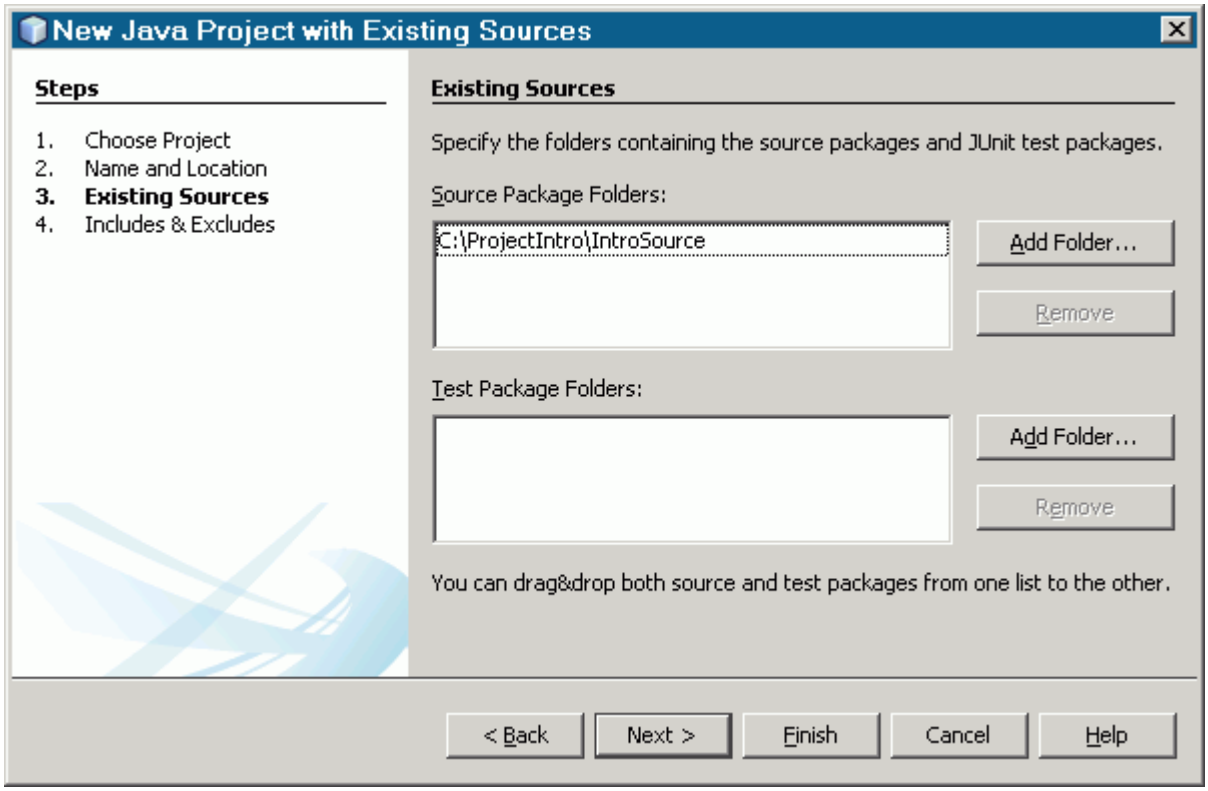
1. In NetBeans, select **File** > **New Project**.
2. In the *New Project* wizard, select **Java Project with Existing Sources** and click **Next**.



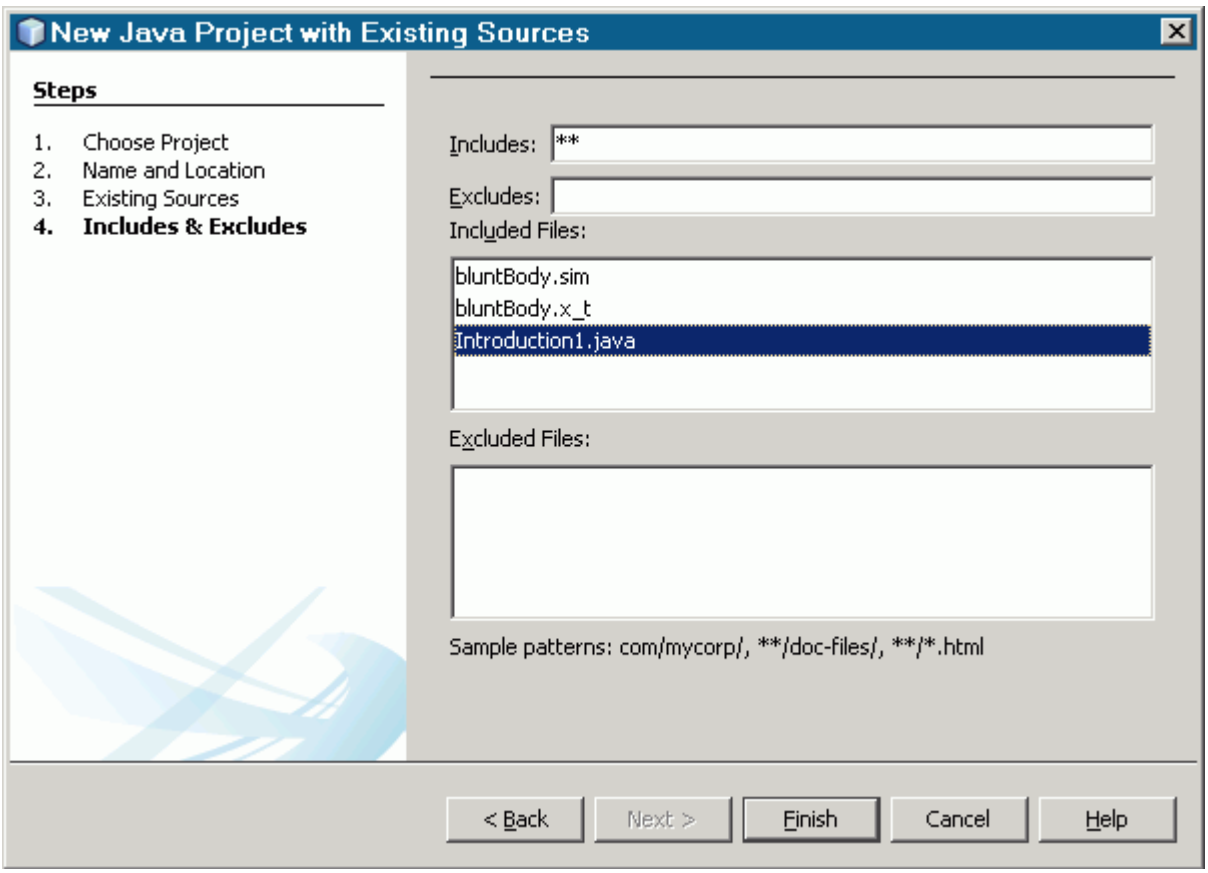
3. In the *Name and Location* panel specify a name for your project in Project Name, provide the location of your project directory in Project Folder, and click **Next**.



4. In the *Existing Sources* panel, click the **Add Directory** button for the **Source Package Folders**, add the source directory that you created earlier and click **Next**.



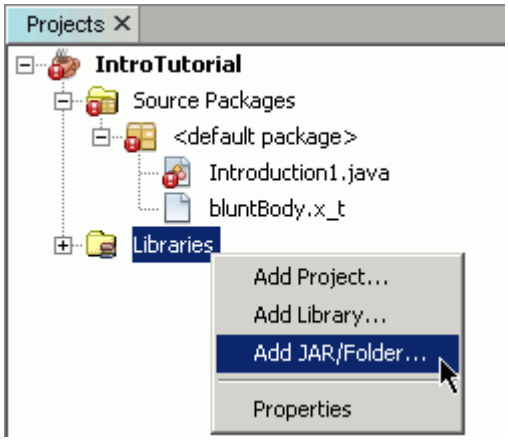
5. In the *Includes* panel, ensure that the required macro is highlighted.



6. Click **Finish** to create the project.

The next step is to add the STAR-CCM+ JAR files to the project so that the IDE can interpret the STAR-CCM+ Java classes. Add the JAR files that are held in the `ext/` directory that is created as part of the installation of STAR-CCM+.

7. In the *Projects* panel in NetBeans, right-click **Libraries** and select **Add JAR/Folder**.

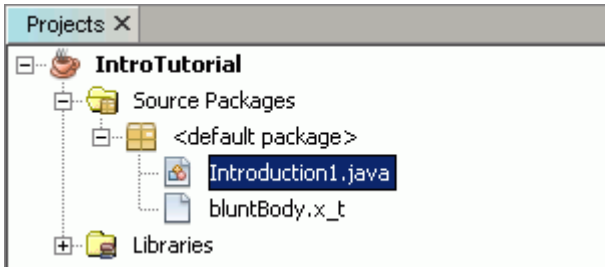


8. In the STAR-CCM+ installation directory, select every JAR file in the `ext/` directory.

On Linux and Windows, this directory is located in:

```
[INSTALLDIR]/star/lib/java/platform/modules/ext/
```

The error symbols disappear from the macro.



You are now ready to start STAR-CCM+ in debugging mode.

Running STAR-CCM+ in Debugging Mode

STAR-CCM+ can pass messages directly to the Java Virtual Machine on which the client runs. You can use this function to attach NetBeans to the instance of STAR-CCM+ through a specified port. In the examples below the port number 8765 is used but you can use any unused port number.

1. Do one of the following:
   - On Windows, in a command prompt, enter the following command on a single line:

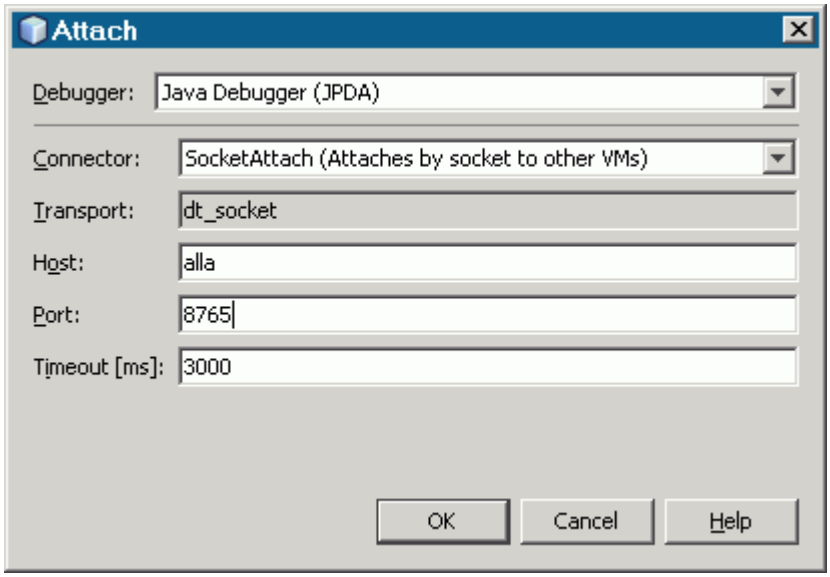
     ```
     <InstallationDirectory>\star\bin\starccm+.exe -jvmargs "-Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=8765"
     ```
   - On Linux, in a terminal window, enter the following command on a single line:

     ```
     <InstallationDirectory>/star/bin/starccm+ -jvmargs '-Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=8765'
     ```

   STAR-CCM+ opens.

Now attach the IDE to STAR-CCM+ through the port:

2. In NetBeans, select **Debug** > **Attach Debugger**, specify the correct port, and click **OK**.



You are now ready to run the macro in debugging mode.

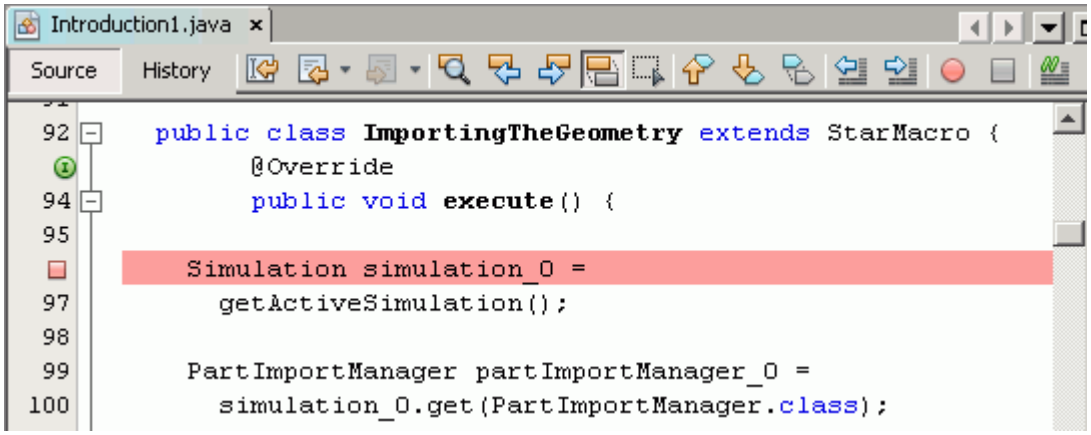
The macro executes to the specified breakpoint. You can use the IDEs stepping tools to work through each command from this point.

Stepping Through the Macro

Connecting STAR-CCM+ to an IDE lets you execute a macro up to any line and step through individual commands to understand each. Once the IDE is attached to STAR-CCM+, you can create one or more breakpoints in the macro before running.

To step through the macro, do the following:

1. In NetBeans, place a breakpoint at the start of your macro by clicking the line number.



2. In STAR-CCM+, load or create a case as necessary.
3. In STAR-CCM+, run the macro.

Running Multiple Simulations Consecutively

You can load and run multiple simulation files consecutively using a single macro. You do not need a separate shell or DOS script.

The following sample macro looks for all the `.sim` files in a specified directory, then for each simulation file it starts a server, iterates, and saves to a new filename. Starting new simulations requires the appropriate licenses.

```
// STAR-CCM+ macro: runMultiple.java
package macro;
import java.io.*;
import star.base.neo.*;
import star.common.*;
public class runMultiple extends StarMacro {
  public class SimFileFilter implements FilenameFilter {
    public boolean accept(File dir, String name) {
      return name.endsWith(".sim");
    }
  }
  public void execute() {
    File simDir = new File("c:\\users\\john\\documents\\testMultiple");
    Simulation sim_0 = getActiveSimulation();
    sim_0.kill();
    for (File f : simDir.listFiles(new SimFileFilter())) {
      startAndRun(f);
    }
  }
  public void startAndRun(File f) {
    System.out.println("\n Starting "+f);
    String fileName = f.getAbsolutePath();
    Simulation sim = new Simulation(fileName);
    // Uncomment next line to clear simulation before running
    // sim.getSolution().clearSolution();
    sim.getSimulationIterator().run();
    String newFileName = fileName.replaceAll("\\.sim","-new.sim");
    sim.saveState(newFileName);
    sim.kill();
  }
}
```

To use the sample macro:

1. Save the macro to a Java file and name it `runMultiple.java`.
2. In the macro, provide the correct path to the working directory. That is, replace "C:\\users\\john\ \documents\\testMultiple" with the path to the actual working directory. The working directory contains the simulation files to run in sequence.

3. In the working directory, issue the following command:

   Windows:

   ```
   [INSTALL_DIR]\star\bin\starccm+.exe -batch runMultiple.java
   ```

   Linux:

   ```
   [INSTALL_DIR]/star/bin/starccm+ -batch runMultiple.java
   ```

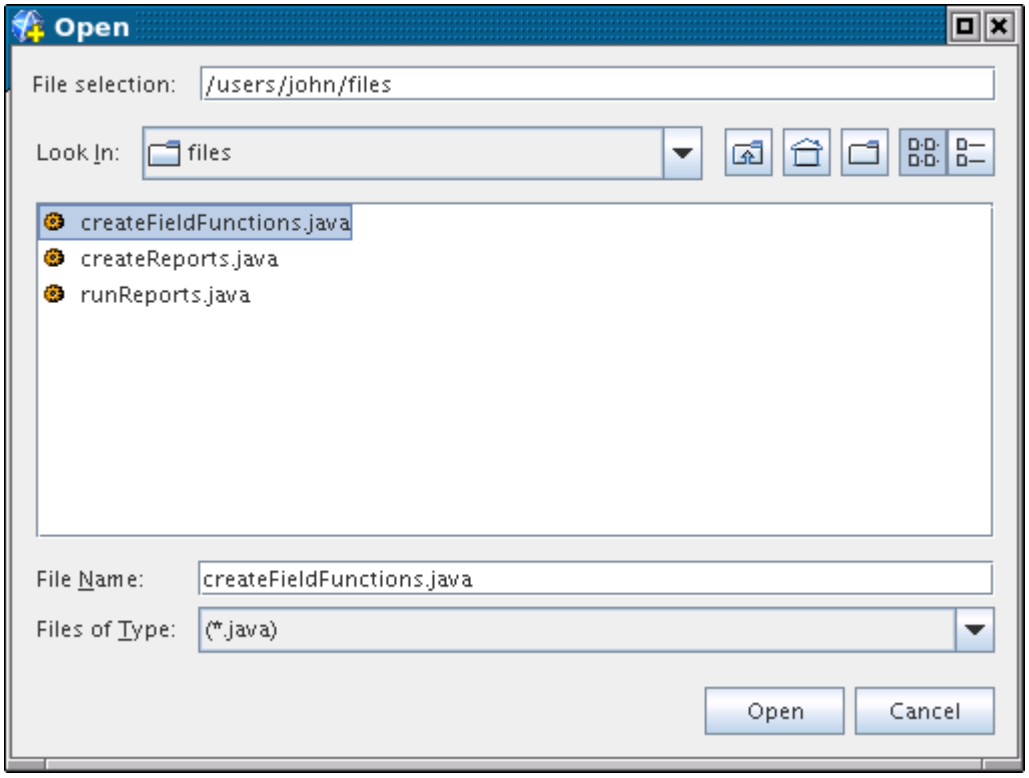
Playing Macros Within Macros

For greater automation of your simulation work, you can create a macro that calls other macros.

To create this kind of macro, play a macro while you record one. For example, suppose that you have a group of simulations that involve similar subtasks, such as working with the same types of field functions and reports.

Do the following:

1. Record a macro for each subtask, such as `createFieldFunctions.java` or `createReports.java`.
2. As you record the simulation macro, play the subtask macro.



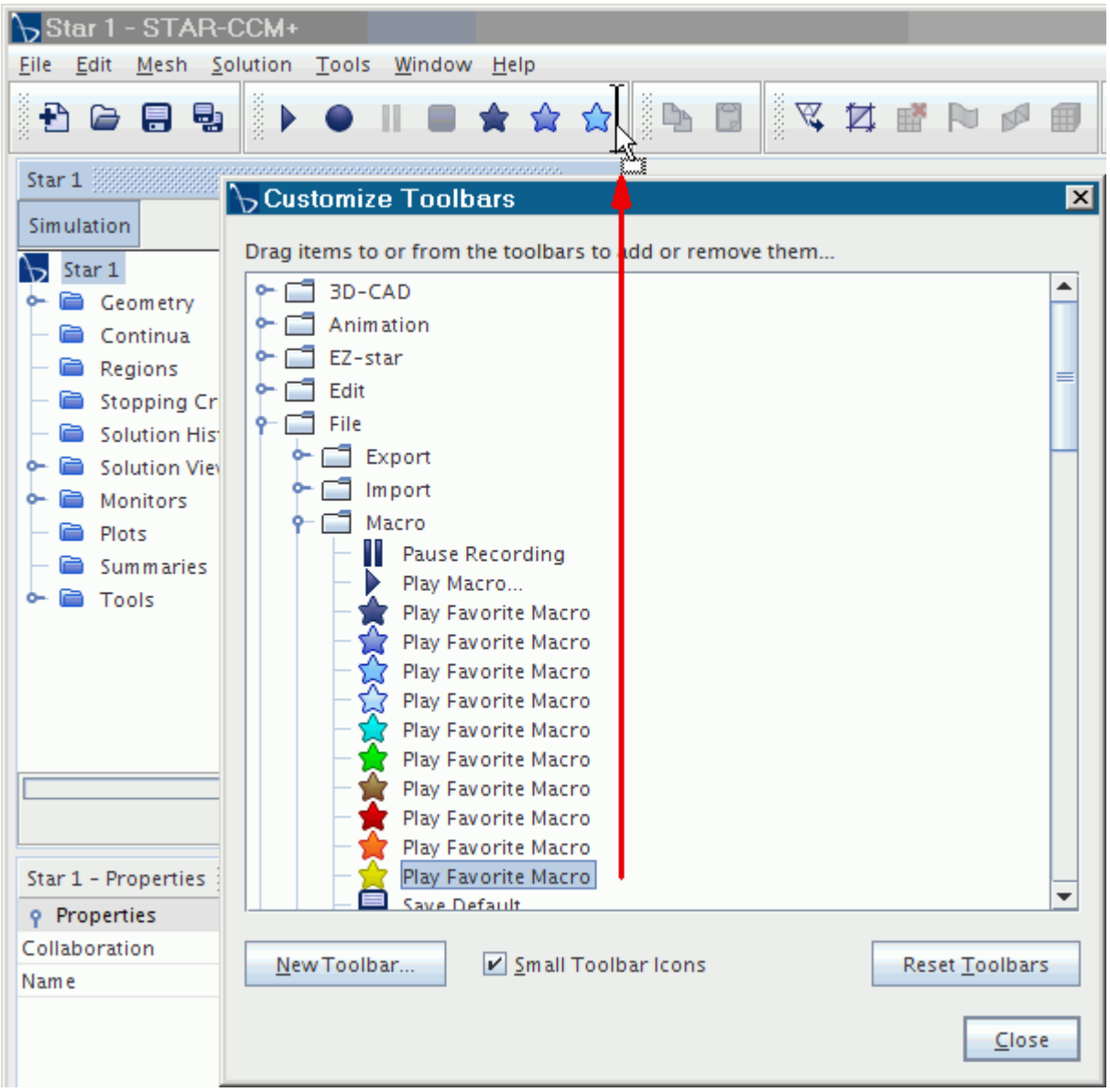
Alternatively, for your subsequent simulations, you can manually edit the macro files to copy and paste the references to the macros of the subtasks.

Adding Custom Macros to the Toolbar

You can add toolbar buttons that instantly launch the macros of your choice.

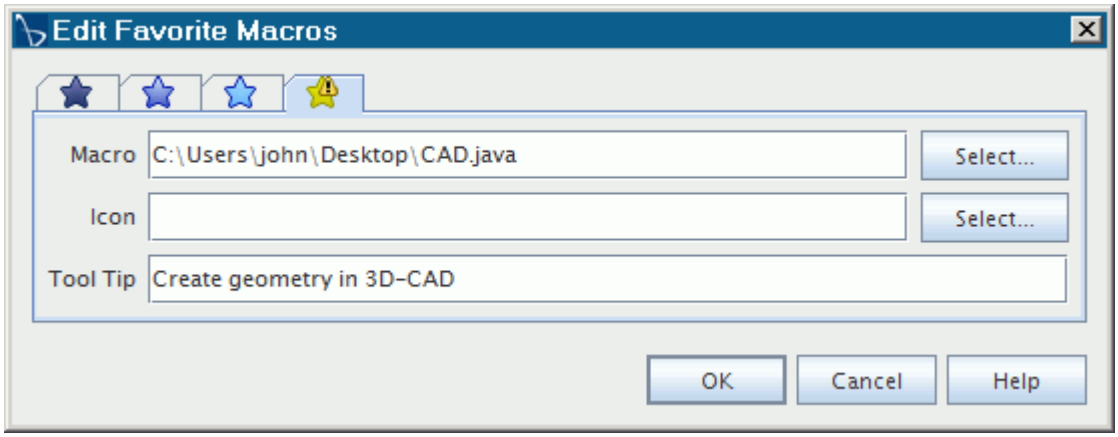
To set up a toolbar button for a macro:

1. Load a new or existing simulation.
2. Select **Window** > **Toolbars** > **Customize**.
3. In the *Customize Toolbars* dialog, expand the **File** > **Macro** node.
4. Drag and drop one of the **Play Favorite Macro** icons next to existing icons on the toolbar of your choice.



A new icon, marked with a yellow warning symbol, is added to the toolbar. The yellow warning symbol indicates that the icon definition is incomplete.

5. To complete the icon definition, click the icon.
6. In the *Edit Favorite Macros* dialog, enter the following information:



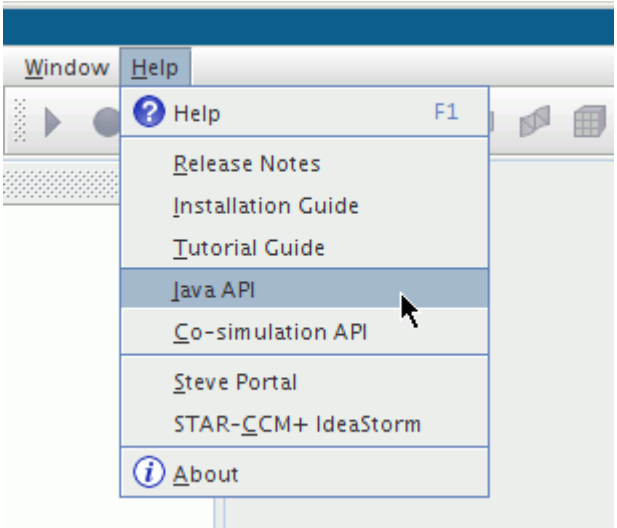
   - **Macro** -- the path to the `.java` file.
   - **Icon** -- the user-defined icon for the macro in the form of image file. Acceptable formats include `.jpg`, `.jpeg`, `.png`, `.tif`, `.tiff`, and `.gif`.
   - **Tool Tip** -- a short description about the task that the macro performs.
7. Click **OK**.

   The icon for the macro is added to the toolbar. When you click it, the macro starts.
8. To modify the button, do one of the following:
   - Right-click and select **Edit**.
   - Select **File** > **Macro** > **Edit Favorites**.
9. To remove the button, open the *Customize Toolbars* dialog and drag the icon back into it.

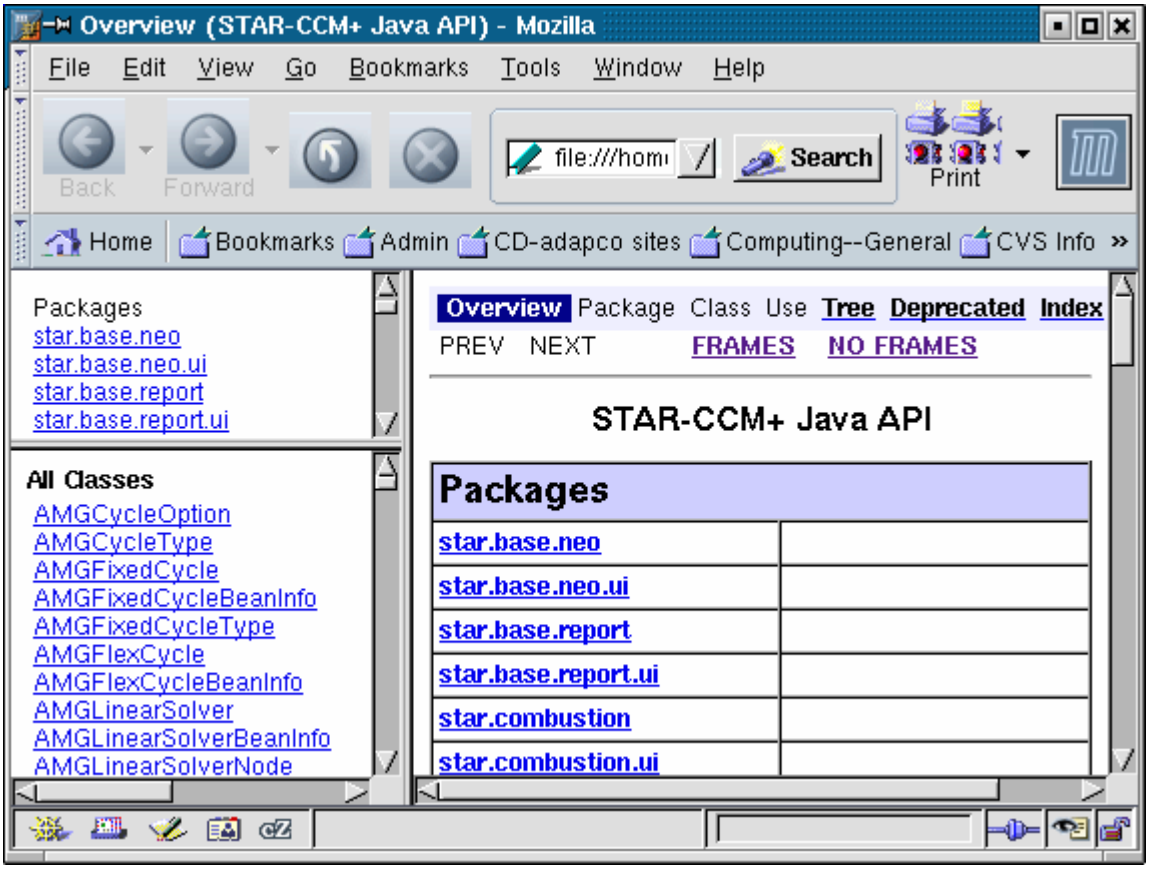
Using the Java API

The Java API, or Application Programmer Interface, contains a comprehensive range of Java code that functions in STAR-CCM+ macros.

To access this material, select the **Help** > **Java API** menu item.



This action makes the STAR-CCM+ API resources available through your web browser.



This set of Java resources only includes Java code that applies to STAR-CCM+ macros. For a more extensive Java API, visit the Oracle Java page: http://www.oracle.com/technetwork/java/index.html.

Do not reuse any of the class names belonging to the STAR-CCM+ Java API, as it can cause conflicts at compile time.

If your browser fails to respond, change the browser path or change the default browser that STAR-CCM+ uses.

Programming in Java

This section contains links to several tutorials on Java programming.

These tutorials are especially useful if you already have some experience with another programming language, or at least some familiarity with basic programming concepts.

http://www.mindview.net/Books/TIJ/ Free Electronic Book: Thinking in Java, 3rd Edition, by Bruce Eckel. An electronic book that you can download to your local drive. It is one of the better learning tools for Java programming, though it presumes prior programming knowledge.

http://www.dickbaldwin.com/tocint.htm Java and JavaScript Programming, by Richard G. Baldwin. An excellent, comprehensive tutorial that makes the subject easy to learn.

http://www.ibiblio.org/javafaq/books/jdr/chapters/ The Java Developer's Resource by Elliotte Rusty Harold. An online version of the book by the same name. The book was developed from the same author's Brewing Java: A Tutorial, which can be found at http://www.ibiblio.org/javafaq/javatutorial.html.

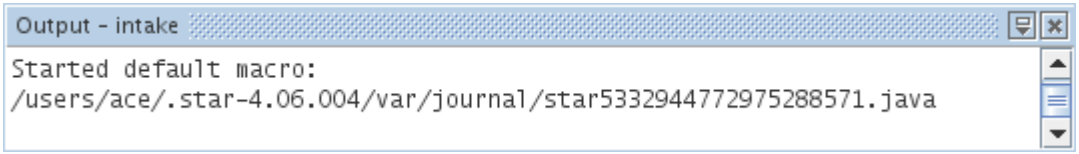
http://docs.oracle.com/javase/tutorial/ The Java Tutorial by Oracle. A well-structured introduction to the Java language.

http://www.oracle.com/technetwork/java/langenv-140151.html White Paper - The Java Language Environment by Oracle. A thorough description of the Java language with a useful feature: comparison of Java with C and C++.

Working with Default Macros

The default macro is a Java macro that is automatically created at the start of each interactive session.

During the interactive session, STAR-CCM+ automatically records the Java code for every action in the default macro.



You can use the default macro to repeat the sequence of actions that you performed during the session when you play it using the Play Macro command. A copy of the default macro can be saved through the STAR-CCM+ GUI at any point in the simulation.

Default macros make it easier for you to submit cases to Siemens PLM Software support engineers for debugging. However, when sending a macro to a support engineer, also send any initial simulation file or input files that the macro requires.

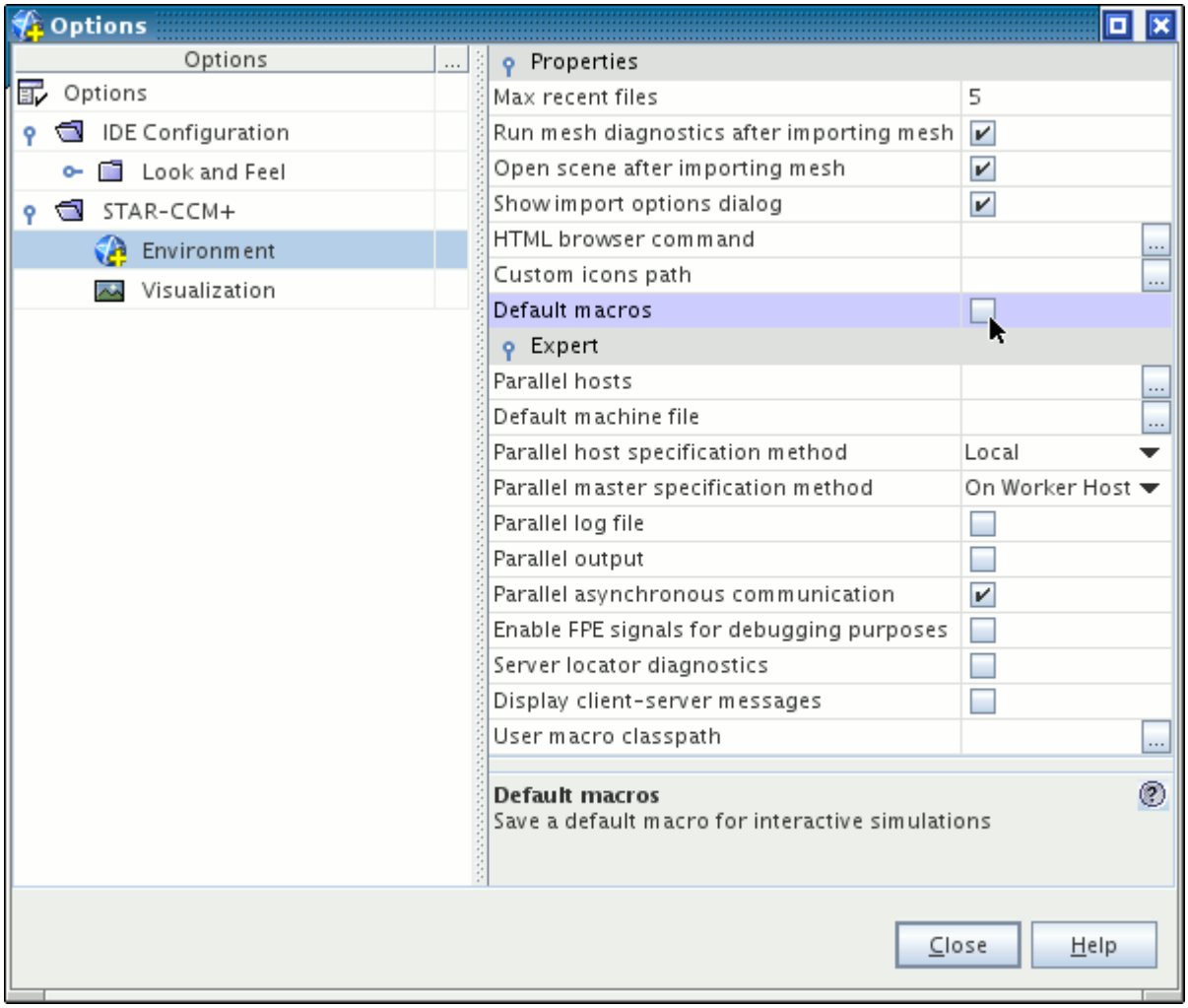
The path to the directory in which default macros are stored on your system is displayed in the *Output* window whenever a new session is started, as shown in the preceding screenshot.

Disabling Default Macros

The default macros feature is activated as standard in STAR-CCM+, but can be deactivated if desired. As a default macro is generated for each interactive session, it is good practice to empty the default macros directory occasionally to make sure that it does not become cluttered.

To stop the creation of default macros:

1. Open the *Options* dialog by selecting **Tools** > **Options** in the menu bar.
2. On the left-hand side of the dialog, select STAR-CCM+ > **Environment**, then on the right-hand side of the dialog, deactivate the *Default macros* property.



3. Finally, click **Close**.

If you need to reactivate the creation of default macros, repeat the above procedure and activate the *Default macros* property.

Changes that are made to this setting only come into effect in subsequent interactive sessions. For example, when a new simulation is started, or an existing `.sim` file is opened. The setting is maintained if STAR-CCM+ is closed and reopened.

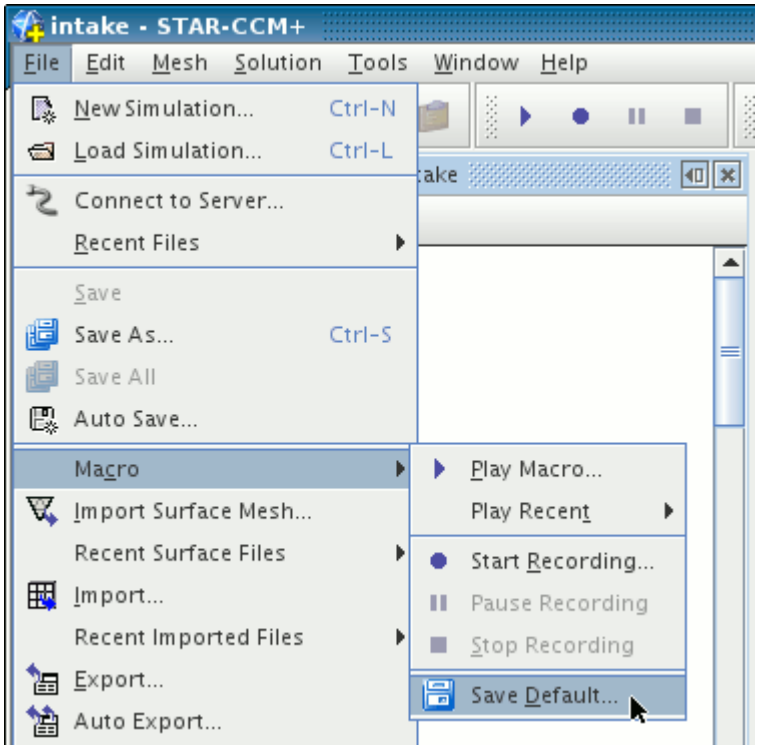
Saving a Default Macro

A copy of the default macro can be saved at any point during the interactive session.

The macro contains every action that is performed in the session up to the point of saving.

To save the default macro:

1. Select **File** > **Macro** > **Save Default...**.



This action opens the standard, operating system-specific *Save* dialog.

2. Navigate to the directory where you want to save a copy of the default macro and give it a name.

   The `.java` extension is automatically added to the file name.

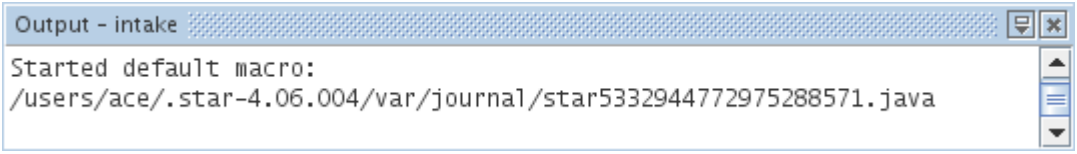
Contents:

Emptying the Default Macros Folder

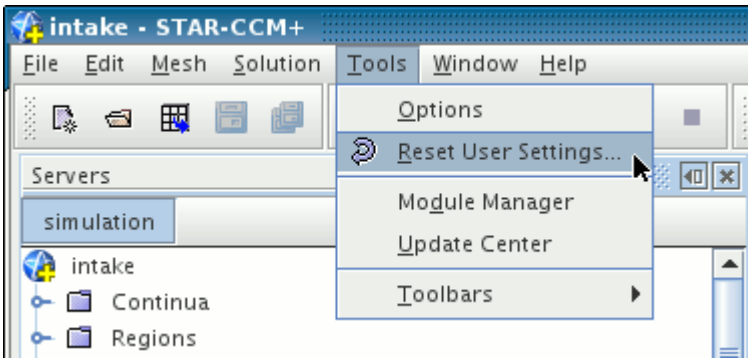
Emptying the Default Macros Folder

It is good practice to empty the default macros folder occasionally to ensure that it does not become large and cluttered.

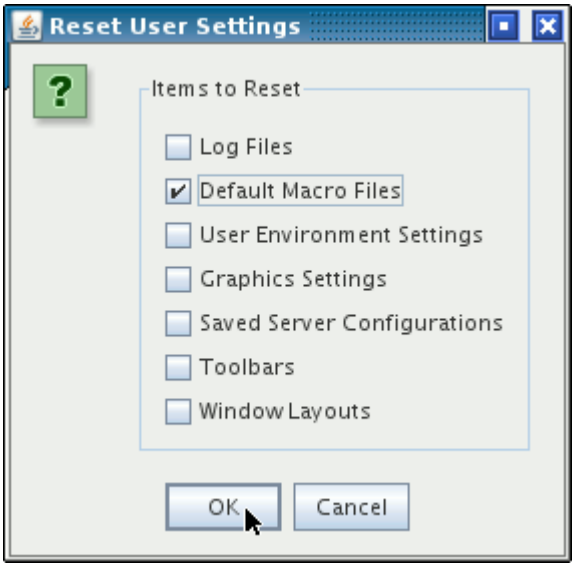
All default macros are created and stored in a dedicated folder that is generated during the installation of STAR-CCM+. The location of this folder is given in the *Output* window at the start of each interactive session as shown below.



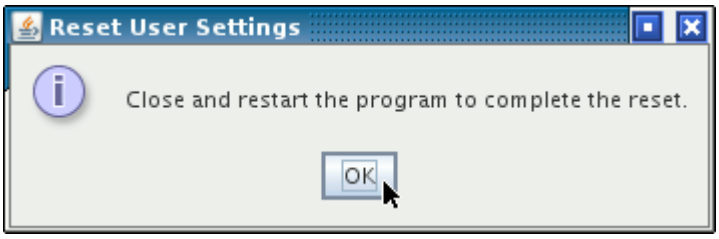
It is good practice to empty the default macros folder occasionally to ensure that it does not become large and cluttered. You can empty the default macros folder in the STAR-CCM+ GUI through the *Reset User Settings* dialog. This dialog is opened by selecting **Tools** > **Reset User Settings...** in the menu bar.



In the *Reset User Settings* dialog, tick the *Default Macro Files* checkbox and click **OK** to schedule the deletion of all default macros in the folder.



A message is displayed that prompts you to close and restart the program for the action to be completed.



Macro API Changes in Past Releases

The following sections provide a trail of macro API changes for the previous releases of STAR-CCM+.

Contents:

Macro API Changes 12.04
Macro API Changes 12.02
Macro API Changes 11.06
Macro API Changes 11.04
Macro API Changes 11.02
Macro API Changes 10.06
Macro API Changes 10.04
Macro API Changes 10.02
Macro API Changes 9.06
Macro API Changes 9.04
Macro API Changes 9.02
Macro API Changes 8.06
Macro API Changes 8.04
Macro API Changes 8.02
Macro API Changes 7.06
Macro API Changes 7.04
Macro API Changes 7.02
Macro API Changes 6.06
Macro API Changes 6.04
Macro API Changes 6.02
Macro API Changes 5.06
Macro API Changes 5.04
Macro API Changes 5.02
Macro API Changes 4.06
Macro API Changes 4.04
Macro API Changes 4.02
Macro API Changes 3.04
Macro API Changes 3.02
Macro API Changes 2.10

Macro API Changes 12.04

In STAR-CCM+ 12.04, the macro API changed for surface preparation, part operations, dynamic queries, DFBI, finite element analysis, reacting flows, Eulerian multiphase, electromagnetics, electrochemistry, and co-simulation.

Surface Preparation

Changes to Extruder User Interface

The user interface of the Surface Extruder operation has changed to accommodate non-Cartesian coordinate systems, resulting in changes to the macro code.

Previous Release	STAR-CCM+ v12.04

`surfaceExtruderOperation_0.getSurfaceExtruderControls().get(SurfaceExtruderDistance.class).setUnits(units_0);`
`surfaceExtruderOperation_0.getSurfaceExtruderControls().get(SurfaceExtruderDistance.class).setValue(0.1);`
`surfaceExtruderOperation_0.setOutputMode(SurfaceExtruderOperation.OutputMode.SINGLE_NEW_PART);`
`surfaceExtruderOperation_0.setPreExecuteMode(SurfaceExtruderOperation.PreExecuteMode.ALL_PERTINENT);`
`SurfaceExtruderDirection surfaceExtruderDirection_0 =`
 	
`surfaceExtruderOperation_0.getSurfaceExtruderControls().get(SurfaceExtruderDirection.class);`
`surfaceExtruderDirection_0.setDirectionMode(SurfaceExtruderDirection.DirectionMode.SPECIFIED);`
`SurfaceExtruderUserSpecifiedDirection surfaceExtruderUserSpecifiedDirection_0 =`
 	
`surfaceExtruderDirection_0.getUserSpecifiedDirection();`
`surfaceExtruderUserSpecifiedDirection_0.getDirection().setComponents(2.0, 3.0, 4.0);`
	`SurfaceExtruderDirection surfaceExtruderDirection_0 =`
	
	`surfaceExtruderOperation_0.getSurfaceExtruderControls().get(SurfaceExtruderDirection.class);`
	`surfaceExtruderDirection_0.getExtrusionDistance().setUnits(units_0);`
	`surfaceExtruderDirection_0.getExtrusionDistance().setValue(0.1);`
	`surfaceExtruderOperation_0.setOutputMode(SurfaceExtruderOperation.OutputMode.SINGLE_NEW_PART);`
	`surfaceExtruderOperation_0.setPreExecuteMode(SurfaceExtruderOperation.PreExecuteMode.ALL_PERTINENT);`
	`surfaceExtruderDirection_0.setDirectionMode(SurfaceExtruderDirection.DirectionMode.SPECIFIED);`
	`SurfaceExtruderUserSpecifiedDirection surfaceExtruderUserSpecifiedDirection_0 =`
	
	`surfaceExtruderDirection_0.getUserSpecifiedDirection();`
	`Coordinate coordinate_0 =`
	
	`surfaceExtruderUserSpecifiedDirection_0.getDirection();`
	`coordinate_0.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {2.0, 3.0, 4.0}));`

Part Operations: Change to Duplication of Composite Parts

The **Duplicate** command on composite parts has been changed, resulting in changes to the macro code.

Previous Release	STAR-CCM+ v12.04

`// Composite part "top" contains a nested composite part named "middle"`
`CompositePart compositePart_0 = ((CompositePart) simulation_0.get(SimulationPartManager.class).getPart("top"));`
`CompositePart compositePart_1 = ((CompositePart) compositePart_0.getChildParts().getPart("middle"));`
 	
`// duplicate the entire assembly by duplicating "top"`
`CompositePart compositePart_2 = (CompositePart) compositePart_0.duplicatePart(simulation_0.get(SimulationPartManager.class));`
 	
`// The duplicatePart API has modified the nested composite part's name to "middle 1"`
`CompositePart compositePart_3 = ((CompositePart) compositePart_2.getChildParts().getPart("middle 1"));`
	`// Composite part "top" contains a nested composite part named "middle"`
	`CompositePart compositePart_0 = ((CompositePart) simulation_0.get(SimulationPartManager.class).getPart("top"));`
	`CompositePart compositePart_1 = ((CompositePart) compositePart_0.getChildParts().getPart("middle"));`
	
	`// duplicate the entire assembly by duplicating "top"`
	`CompositePart compositePart_2 = (CompositePart) compositePart_0.duplicatePart(simulation_0.get(SimulationPartManager.class));`
	
	`// The duplicatePart API no longer modifies the name of the nested composite part`
	`CompositePart compositePart_3 = ((CompositePart) compositePart_2.getChildParts().getPart("middle"));`

Dynamic Queries: Changes to Region Assignments

Changes to dynamic queries in STAR-CCM+ have resulted in changes to the macro code. If your macro includes the object `ManagerWithQuery`, update your macros as explained here.

When a dynamic query is set, `ManagerWithQuery` generates an exception when attempting to call the objects `addObjects`, `removeObjects`, or `eraseObjects`. When no query is set, these objects operate correctly.

As an example, assume you have a manager with a query set whose objects are currently `A` and `B`:

```
manager.getQuery(); // == true
manager.getObjects(); // == [A, B]
```

Calling `addObjects`, `removeObjects`, or `eraseObjects` in the previous version would not have changed the objects:

```
manager.addObjects(C); // or manager.removeObjects(B)
                       // or manager.eraseObjects(B)
manager.getObjects(); // == [A, B]
```

The current version generates the `IllegalStateException` error:

```
manager.addObjects(C); // IllegalStateException
manager.removeObjects(B); // IllegalStateException
manager.eraseObjects(B); // IllegalStateException
```

It is recommended that you either set the query to `null` explicitly, or check that no query is set before calling the objects:

```
manager.setQuery(null);
manager.addObjects(C); // objects will now be [A, B, C]
```

Alternatively:

```
if (!manager.hasQuery() {
    manager.removeObjects(B); // objects will now be [A]
```

DFBI

Changes to Creation of Bodies

With the introduction of 2D bodies, the way in which bodies are created has changed. In the current release, the `createBody` method takes an additional boolean argument which is `true` if the new body is 3D. For backward compatibility the old `createBody` method, without any argument, still works (it creates a 3D body). The following example shows the change between the previous and current techniques of creating a 3D body:

Previous Release	STAR-CCM+ v12.04

`star.sixdof.Body sixdofBody_0 =`
 	
`simulation_0.get(star.sixdof.Body Manager.class).createBody();`
	`star.sixdof.Body sixdofBody_0 =`
	
	`simulation_0.get(star.sixdof.Body Manager.class).createBody(true);`

Change to Superposed Motion

A change was implemented to the mechanism of DFBI superposed motion, resulting in changes to the macro code. For backward compatibility, the old mechanism can still work in the current version of STAR-CCM+.

Previous Release	STAR-CCM+ v12.04
`Simulation simulation_0 =` `  getActiveSimulation();`	`Simulation simulation_0 =` `  getActiveSimulation();`

Previous Release	STAR-CCM+ v12.04

`SixDofPlusRotatingMotion sixDofPlusRotatingMotion_0 =`
 	
`simulation_0.get(MotionManager.class).createMotion(SixDofPlusRotatingMotion.class, "DFBI Superposed Rotation");`
`sixDofPlusRotatingMotion_0.getAxisVector().setComponents(1.0, 0.0, 0.0);`
`sixDofPlusRotatingMotion_0.getOriginVector().setComponents(0.0, 0.0, -0.1);`
`Units units_0 =`
`  ((Units) simulation_0.getUnitsManager().getObject("rps"));`
`superposingRotatingMotion_0.getRotationRate().setUnits(units_0);`
`sixDofPlusRotatingMotion_0.getRotationRate().setValue(20.0);`
`Region region_0 =`
 	
`simulation_0.getRegionManager().getRegion("PropellerRegion");`
`MotionSpecification motionSpecification_0 =`
 	
`region_0.getValues().get(MotionSpecification.class);`
`motionSpecification_0.setMotion(sixDofPlusRotatingMotion_0);`
	`star.sixdof.Body sixdofBody_0 =`
	`  ((star.sixdof.Body) simulation_0.get(star.sixdof.BodyManager.class).getObject("Body 1"));`
	`SixDofBodyMotion sixDofBodyMotion_0 =`
	`  (SixDofBodyMotion) sixdofBody_0.createSixDofBodyMotion();`
	`LabCoordinateSystem labCoordinateSystem_0 =`
	
	`simulation_0.getCoordinateSystemManager().getLabCoordinateSystem();`
	`CartesianCoordinateSystem cartesianCoordinateSystem_0 =`
	`  ((CartesianCoordinateSystem) labCoordinateSystem_0.getLocalCoordinateSystemManager().getObject("Body 1-CSys"));`
	`CartesianCoordinateSystem cartesianCoordinateSystem_1 =`
	
	`cartesianCoordinateSystem_0.getLocalCoordinateSystemManager().createLocalCoordinateSystem(CartesianCoordinateSystem.class, "Cartesian");`
	`SuperposingRotatingMotion superposingRotatingMotion_0 =`
	
	`sixDofBodyMotion_0.getSuperposingMotionManager().createSuperposingMotion(SuperposingRotatingMotion.class, "Superposing Rotation");`
	`superposingRotatingMotion_0.setCoordinateSystem(cartesianCoordinateSystem_1);`
	`superposingRotatingMotion_0.getAxisDirection().setComponents(1.0, 0.0, 0.0);`
	`superposingRotatingMotion_0.getAxisOrigin().setComponents(0.0,`

Previous Release	STAR-CCM+ v12.04

	`0.0, -0.1);`
	`Units units_0 =`
	`  ((Units)`
	`simulation_0.getUnitsManager().ge`
	`tObject("rps"));`
	`superposingRotatingMotion_0.getRo`
	`tationRate().setUnits(units_0);`
	`superposingRotatingMotion_0.getRo`
	`tationRate().setValue(20.0);`
	`Region region_0 =`
	
	`simulation_0.getRegionManager().g`
	`etRegion("PropellerRegion");`
	`MotionSpecification`
	`motionSpecification_0 =`
	
	`region_0.getValues().get(MotionSp`
	`ecification.class);`
	`motionSpecification_0.setMotion(s`
	`uperposingRotatingMotion_0);`

Finite Element Analysis: Changes to Integration Order Option

The *Integration Order Option* property of the Finite Element Magnetic Vector Potential model has been refactored, resulting in changes to the macro code. The class `star.electromagnetism.magneticpotential.fem.FeIntegrationOrderOption` has been renamed to `star.fea.common.models.FeIntegrationOrderOption`. So, for example, the `import` statement changes as follows:

Previous Release	STAR-CCM+ v12.04

`import`
`star.electromagnetism.magneticpot`
`ential.fem.FeIntegrationOrderOpti`
`on`
	`import`
	`star.fea.common.models.FeIntegrat`
	`ionOrderOption;`

Reacting Flows

Changes to Multiphase Reactions

The techniques for calculating multiphase reactions for liquid-solid interfaces and volume of fluid have changed, resulting in changes to the macro code. The class `InterphaseReactionRateModel` has changed to `EmpInterphaseReactionModel`.

Previous Release	STAR-CCM+ v12.04
`phaseInteraction_0.enable(InterphaseReactionRateModel.class);`	`phaseInteraction_0.enable(EmpInterphaseReactionModel.class);`
`InterphaseReactionRateModel interphaseReactionRateModel_0 = phaseInteraction_0.getModelManager().getModel(InterphaseReactionRateModel.class);`	`EmpInterphaseReactionModel empInterphaseReactionModel_0 = phaseInteraction_0.getModelManager().getModel(EmpInterphaseReactionModel.class);`
`((ReactionManager) interphaseReactionRateModel_0.getReactions()).createReaction();`	`((ReactionManager) empInterphaseReactionModel_0.getReactions()).createReaction();`

For Eulerian multiphase modeling with continuous-dispersed interaction, when using an interphase reaction with a user-defined reaction rate from a previous version of STAR-CCM+, change the field function. Specifically, the earlier reaction rate `[kmol/s]` should now be `[kmol/m2/sec]`.

An example of the changes to the macro code follows:

Previous Release	STAR-CCM+ v12.04
`userFieldFunction_0.setFunctionName("ReactionRate");` `userFieldFunction_0.setDefinition("($TemperatureAir> $TemperatureWater)? 2.e-5*($TemperatureAir- $TemperatureWater):0");`	`userFieldFunction_0.setFunctionName("ReactionRate");` `userFieldFunction_0.setDefinition("($TemperatureAir> $TemperatureWater)? 2.e-5*($TemperatureAir- $TemperatureWater)/${InteractionAreaDensityPhase Interaction 1}:0");`

Changes to Soot Two-Equation Model

The techniques of soot modeling have changed, resulting in the following change to the macro code:

Previous Release	STAR-CCM+ v12.04
`sootTwoEquationModelProperties_0.getSootNucleationModelOption().setSelected(SootNucleationModelOption.PAH_BASED);`	`sootTwoEquationModelProperties_0.getSootNucleationModelOption().setSelected(SootTwoEquationNucleationOption.PAH_BASED);`

Eulerian Multiphase

Changes to Phase Interaction Model for Dispersed Multiphase (DMP) and Volume of Fluid (VOF)

To accommodate the new DMP-VOF Phase Interaction Model structure, the `DispersedPhasePressureGradientModel` class has been removed.

Previous Release	STAR-CCM+ v12.04
`PhaseInteraction phaseInteraction_1 = multiPhaseInteractionModel_0.createPhaseInteraction(); phaseInteraction_1.setPresentationName("DMP-Continuum"); phaseInteraction_1.enable(DispersedPhysicsContinuumInteractionModel.class); phaseInteraction_1.enable(DispersedPhasePressureGradientModel.class); phaseInteraction_1.enable(ParticleDiameterInteractionLengthScaleModel.class); phaseInteraction_1.enable(DispersedPhaseHeatTransferModel.class); phaseInteraction_1.enable(DispersedPhaseDragForceModel.class);`	`PhaseInteraction phaseInteraction_1 = multiPhaseInteractionModel_0.createPhaseInteraction(); phaseInteraction_1.setPresentationName("DMP-Continuum"); phaseInteraction_1.enable(DispersedPhysicsContinuumInteractionModel.class); phaseInteraction_1.enable(ParticleDiameterInteractionLengthScaleModel.class); phaseInteraction_1.enable(DispersedPhaseHeatTransferModel.class); phaseInteraction_1.enable(DispersedPhaseDragForceModel.class);`

Renaming of VOF Model Classes

The `VofCavitationSingleComponentModel` class has been renamed to `VofSchnerrSauerCavitationSingleComponentModel`, and the `VofCavitationMultiComponentModel` class has been renamed to `VofSchnerrSauerCavitationMultiComponentModel`.

The following example shows the change to the single-component class:

Previous Release	STAR-CCM+ v12.04
`phaseInteraction_0.enable(VofPhaseInteractionModel.class); VofPhaseInteractionModel vofPhaseInteractionModel_0 = phaseInteraction_0.getModelManager().getModel(VofPhaseInteractionModel.class); EulerianMultiPhaseModel eulerianMultiPhaseModel_0 = physicsContinuum_0.getModelManager().getModel(EulerianMultiPhaseModel.class);`	`phaseInteraction_0.enable(VofPhaseInteractionModel.class); VofPhaseInteractionModel vofPhaseInteractionModel_0 = phaseInteraction_0.getModelManager().getModel(VofPhaseInteractionModel.class); EulerianMultiPhaseModel eulerianMultiPhaseModel_0 = physicsContinuum_0.getModelManager().getModel(EulerianMultiPhaseModel.class);`

Previous Release	STAR-CCM+ v12.04
`EulerianPhase eulerianPhase_0 = ((EulerianPhase) eulerianMultiPhaseModel_0.getPhaseManager().getPhase("H2O")); vofPhaseInteractionModel_0.setPrimaryPhase(eulerianPhase_0); EulerianPhase eulerianPhase_1 = ((EulerianPhase) eulerianMultiPhaseModel_0.getPhaseManager().getPhase("H2O(G)")); vofPhaseInteractionModel_0.setSecondaryPhase(eulerianPhase_1); phaseInteraction_0.enable(VofCavitationSingleComponentModel.class); phaseInteraction_0.enable(PhaseInteractionMaterialModel.class);`	`EulerianPhase eulerianPhase_0 = ((EulerianPhase) eulerianMultiPhaseModel_0.getPhaseManager().getPhase("H2O")); vofPhaseInteractionModel_0.setPrimaryPhase(eulerianPhase_0); EulerianPhase eulerianPhase_1 = ((EulerianPhase) eulerianMultiPhaseModel_0.getPhaseManager().getPhase("H2O(G)")); vofPhaseInteractionModel_0.setSecondaryPhase(eulerianPhase_1); phaseInteraction_0.enable(VofSchnerrSauerCavitationSingleComponentModel.class); phaseInteraction_0.enable(PhaseInteractionMaterialModel.class);`

Similar to the previous example, the corresponding lines would need to be replaced for cavitation between **Multi-Component Liquid** and **Multi-Component Gas**:

Previous Release	STAR-CCM+ v12.04
`phaseInteraction_0.enable(VofCavitationMultiComponentModel.class);`	`phaseInteraction_0.enable(VofSchnerrSauerCavitationMultiComponentModel.class);`

Changes to Heat Transfer Coefficient Method

In the Boiling/Condensation model of a Multiple Flow Regime Phase Interaction, the internal heat transfer coefficient method manager is now accessible for the first and second regime Nusselt number, resulting in changes to the macro code.

The object `getPrimaryNusseltNumber()` is now `getFirstRegimeExternalNuMethodManager()`, and the object `getSecondaryNusseltNumber()` is now `getSecondRegimeExternalNuMethodManager()`. The following example shows the effects of these changes:

Previous Release	STAR-CCM+ v12.04
`LsiBulkBoilingModel lsiBulkBoilingModel_0 = phaseInteraction_0.getModelManager().getModel(LsiBulkBoilingModel.class);`	`LsiBulkBoilingModel lsiBulkBoilingModel_0 = phaseInteraction_0.getModelManager().getModel(LsiBulkBoilingModel.class);`

Previous Release	STAR-CCM+ v12.04
`ConstantInteractionCoefficientMethod constantInteractionCoefficientMethod_0 = lsiBulkBoilingModel_0.getPrimaryNusseltNumber().getMethod(ConstantInteractionCoefficientMethod.class); constantInteractionCoefficientMethod_0.getQuantity().setValue(2.0); ConstantInteractionCoefficientMethod constantInteractionCoefficientMethod_1 = lsiBulkBoilingModel_0.getSecondaryNusseltNumber().getMethod(ConstantInteractionCoefficientMethod.class); constantInteractionCoefficientMethod_1.getQuantity().setValue(2.0);`	`ConstantInteractionCoefficientMethod constantInteractionCoefficientMethod_0 = lsiBulkBoilingModel_0.getFirstRegimeExternalNuMethodManager().getMethod(ConstantInteractionCoefficientMethod.class); constantInteractionCoefficientMethod_0.getQuantity().setValue(2.0); ConstantInteractionCoefficientMethod constantInteractionCoefficientMethod_1 = lsiBulkBoilingModel_0.getSecondRegimeExternalNuMethodManager().getMethod(ConstantInteractionCoefficientMethod.class); constantInteractionCoefficientMethod_1.getQuantity().setValue(2.0);`

Note: In v12.02 and earlier, the expression internal heat transfer used in the Boiling/Condensation model of a Multiple Flow Regime Phase Interaction was evaluated within STAR-CCM+. With v12.04, the internal heat transfer coefficient method manager is accessible and you have to specify the Nu / HTC value for internal heat transfer. In the likely scenario where you want to use the old expression for the specification of internal heat transfer, Nu / HTC, you may do so by defining a field function, the details of which are provided in the User Guide.

The object `star.multiphase.FunctionInterphaseHTCxAreaMethod` is no longer available. In v12.02 and earlier, the `FunctionInterphaseHTCxAreaMethod` worked with the product of the `HeatTransferCoefficient` and the interaction area density as a single field function. In v12.04, a new method, `star.multiphase.FunctionInterphaseHTCMethod`, is now available. You can specify `HeatTransferCoefficient` as a field function and the interaction area density is calculated by the Interaction Area Density model.

Changes to Wall Boiling

The Wall Boiling model has been extended to work with Large Scale Interface (LSI) models, resulting in changes to the macro code.

Select the Model

Previous Release	STAR-CCM+ v12.04

`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Coolant"));`
`MultiPhaseInteractionModel multiPhaseInteractionModel_0 = physicsContinuum_0.getModelManager().getModel(MultiPhaseInteractionModel.class);`
`PhaseInteraction phaseInteraction_0 = ((PhaseInteraction) multiPhaseInteractionModel_0.getPhaseInteractionManager().getPhaseInteraction("Liquid Vapour Interface"));`
`phaseInteraction_0.enable(WallBoilingModel.class);`
`phaseInteraction_0.enable(LemmartChawlaNucleationSiteNumberDensityModel.class);`
`phaseInteraction_0.enable(KurulPodowskiBubbleInfluenceWallAreaFractionModel.class);`
`phaseInteraction_0.enable(DelValleKenningBubbleInducedQuenchingHeatTransferCoefficientModel.class);`
`phaseInteraction_0.enable(TolubinskyKonstanchukBubbleDepartureDiameterModel.class);`
`phaseInteraction_0.enable(ColeBubbleDepartureFrequencyModel.class);`
`phaseInteraction_0.enable(BasicWallDryoutAreaFractionModel.class);`
	`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Coolant"));`
	`MultiPhaseInteractionModel multiPhaseInteractionModel_0 = physicsContinuum_0.getModelManager().getModel(MultiPhaseInteractionModel.class);`
	`PhaseInteraction phaseInteraction_0 = ((PhaseInteraction) multiPhaseInteractionModel_0.getPhaseInteractionManager().getPhaseInteraction("Liquid Vapour Interface"));`
	`phaseInteraction_0.enable(WallBoilingModel.class);`
	`phaseInteraction_0.enable(BubbleNucleationModel.class);`
	`phaseInteraction_0.enable(TransientConductionModel.class);`

Changing Settings

Previous Release	STAR-CCM+ v12.04
`WallBoilingModel wallBoilingModel_0 =`	`WallBoilingModel wallBoilingModel_0 =`

Previous Release	STAR-CCM+ v12.04

`phaseInteraction_0.getModelManager().getModel(WallBoilingModel.class);`
`wallBoilingModel_0.getRelaxationFactor().setValue(0.4);`
`BasicWallDryoutAreaFractionModel basicWallDryoutAreaFractionModel_0 =`
 	
`phaseInteraction_0.getModelManager().getModel(BasicWallDryoutAreaFractionModel.class);`
 	
`//change the Bubbly Layer Relaxation Factor`
`basicWallDryoutAreaFractionModel_0.getRelaxationFactor().setValue(0.4);`
 	
`//change the Bubbly Layer Option to Fixed Yplus and set the value to 150`
`basicWallDryoutAreaFractionModel_0.getBubblyLayerThicknessOption().setSelected(BubblyLayerThicknessOption.Type.FIXED_YPLUS);`
`BubblyLayerThicknessFixedYplus bubblyLayerThicknessFixedYplus_0 =`
 	
`basicWallDryoutAreaFractionModel_0.getBubblyLayerThicknessFixedYplus();`
`bubblyLayerThicknessFixedYplus_0.setFixedYplus(150.0);`
 	
`//change the Quenching Temperature Option to Fixed Yplus and set the value to 150`
`BubbleInducedQuenchingTemperatureModel bubbleInducedQuenchingTemperatureModel_0 =`
 	
`phaseInteraction_0.getModelManage`
	`phaseInteraction_0.getModelManager().getModel(WallBoilingModel.class);`
	`wallBoilingModel_0.getRelaxationFactor().setValue(0.4);`
	`WallDryoutAreaFractionProfile wallDryoutAreaFractionProfile_0 =`
	
	`wallBoilingModel_0.getWallDryoutAreaFractionProfile();`
	
	`//change the Bubbly Layer Relaxation Factor`
	`wallDryoutAreaFractionProfile_0.getBubblyLayerRelaxationFactor().setValue(0.4);`
	
	`//change the Bubbly Layer Option to Fixed Yplus and set the value to 150`
	`wallDryoutAreaFractionProfile_0.getBubblyLayerThicknessOption().setSelected(BubblyLayerThicknessOption.Type.FIXED_YPLUS);`
	`BubblyLayerThicknessFixedYplus bubblyLayerThicknessFixedYplus_0 =`
	
	`wallDryoutAreaFractionProfile_0.getBubblyLayerThicknessFixedYplus();`
	`bubblyLayerThicknessFixedYplus_0.setFixedYplus(150.0);`
	
	`//change the Quenching Temperature Option to Fixed Yplus and set the value to 150`
	`TransientConductionModel transientConductionModel_0 =`
	
	`phaseInteraction_0.getModelManager().getModel(TransientConductionModel.class);`
	`transientConductionModel_0.getBub`

Previous Release	STAR-CCM+ v12.04

`r().getModel(BubbleInducedQuenchingTemperatureModel.class);`
`bubbleInducedQuenchingTemperatureModel_0.getBubbleInducedQuenchingTemperatureOption().setSelected(BubbleInducedQuenchingTemperatureOption.Type.FIXED_YPLUS);`
`BubbleInducedQuenchingFixedYplus bubbleInducedQuenchingFixedYplus_0 =`
 	
`bubbleInducedQuenchingTemperatureModel_0.getBubbleInducedQuenchingFixedYplus();`
`bubbleInducedQuenchingFixedYplus_0.setFixedYplus(150.0);`
	`bleInducedQuenchingTemperatureOption().setSelected(BubbleInducedQuenchingTemperatureOption.Type.FIXED_YPLUS);`
	`BubbleInducedQuenchingFixedYplus bubbleInducedQuenchingFixedYplus_0 =`
	
	`transientConductionModel_0.getBubbleInducedQuenchingFixedYplus();`
	`bubbleInducedQuenchingFixedYplus_0.setFixedYplus(150.0);`

Remove Standard Submodels and Use Constants

This technique applies to the specification of Constant methods only.

Previous Release	STAR-CCM+ v12.04

`PhaseInteraction phaseInteraction_0 =`
`  ((PhaseInteraction) multiPhaseInteractionModel_0.getPhaseInteractionManager().getPhaseInteraction("Liquid Vapour Interface"));`
`BasicWallDryoutAreaFractionModel basicWallDryoutAreaFractionModel_1 =`
 	
`phaseInteraction_0.getModelManager().getModel(BasicWallDryoutAreaFractionModel.class);`
`phaseInteraction_0.disableModel(basicWallDryoutAreaFractionModel_1);`
`phaseInteraction_0.enable(UserDefinedWallDryoutAreaFractionModel.class);`
`UserDefinedWallDryoutAreaFractionModel userDefinedWallDryoutAreaFraction`
	`PhaseInteraction phaseInteraction_0 =`
	`  ((PhaseInteraction) multiPhaseInteractionModel_0.getPhaseInteractionManager().getPhaseInteraction("Liquid Vapour Interface"));`
	`WallBoilingModel wallBoilingModel_1 =`
	
	`phaseInteraction_0.getModelManager().getModel(WallBoilingModel.class);`
	`WallDryoutAreaFractionProfile wallDryoutAreaFractionProfile_0 =`
	
	`wallBoilingModel_1.getWallDryoutAreaFractionProfile();`
	`wallDryoutAreaFractionProfile_0.setMethod(ConstantScalarProfileMethod.class);`
	`wallDryoutAreaFractionProfile_0.g`

Previous Release	STAR-CCM+ v12.04

Model_0 =
 	
phaseInteraction_0.getModelManager().getModel(UserDefinedWallDryoutAreaFractionModel.class);
WallDryoutAreaFractionProfile
wallDryoutAreaFractionProfile_0
=
 	
userDefinedWallDryoutAreaFractionModel_0.getWallDryoutAreaFractionProfile();
wallDryoutAreaFractionProfile_0.getMethod(ConstantScalarProfileMethod.class).getQuantity().setValue(1.0);
LemmartChawlaNucleationSiteNumberDensityModel
lemmartChawlaNucleationSiteNumberDensityModel_1 =
 	
phaseInteraction_0.getModelManager().getModel(LemmartChawlaNucleationSiteNumberDensityModel.class);
phaseInteraction_0.disableModel(lemmartChawlaNucleationSiteNumberDensityModel_1);
phaseInteraction_0.enable(UserDefinedNucleationSiteNumberDensityModel.class);
UserDefinedNucleationSiteNumberDensityModel
userDefinedNucleationSiteNumberDensityModel_0 =
 	
phaseInteraction_0.getModelManager().getModel(UserDefinedNucleationSiteNumberDensityModel.class);
NucleationSiteNumberDensityProfile
nucleationSiteNumberDensityProfile_0 =
 	
userDefinedNucleationSiteNumberDensityModel_0.getNucleationSiteNum
	etMethod(ConstantScalarProfileMethod.class).getQuantity().setValue(1.0);
	BubbleNucleationModel
	bubbleNucleationModel_1 =
	
	phaseInteraction_0.getModelManager().getModel(BubbleNucleationModel.class);
	NucleationSiteNumberDensityProfile
	nucleationSiteNumberDensityProfile_0 =
	
	bubbleNucleationModel_1.getNucleationSiteNumberDensityProfile();
	nucleationSiteNumberDensityProfile_0.setMethod(ConstantScalarProfileMethod.class);
	nucleationSiteNumberDensityProfile_0.getMethod(ConstantScalarProfileMethod.class).getQuantity().setValue(1.0);
	BubbleDepartureDiameterProfile
	bubbleDepartureDiameterProfile_0
	=
	
	bubbleNucleationModel_1.getBubbleDepartureDiameterProfile();
	bubbleDepartureDiameterProfile_0.setMethod(ConstantScalarProfileMethod.class);
	bubbleDepartureDiameterProfile_0.getMethod(ConstantScalarProfileMethod.class).getQuantity().setValue(1.0);
	BubbleDepartureFrequencyProfile
	bubbleDepartureFrequencyProfile_0
	=
	
	bubbleNucleationModel_1.getBubbleDepartureFrequencyProfile();
	bubbleDepartureFrequencyProfile_0.setMethod(ConstantScalarProfileMethod.class);

Previous Release	STAR-CCM+ v12.04

berDensityProfile();
nucleationSiteNumberDensityProfile_0.getMethod(ConstantScalarProfileMethod.class).getQuantity().setValue(1.0);
TolubinskyKonstanchukBubbleDepartureDiameterModel
tolubinskyKonstanchukBubbleDepartureDiameterModel_1 =
 	
phaseInteraction_0.getModelManager().getModel(TolubinskyKonstanchukBubbleDepartureDiameterModel.class);
phaseInteraction_0.disableModel(tolubinskyKonstanchukBubbleDepartureDiameterModel_1);
phaseInteraction_0.enable(UserDefinedBubbleDepartureDiameterModel.class);
UserDefinedBubbleDepartureDiameterModel
userDefinedBubbleDepartureDiameterModel_0 =
 	
phaseInteraction_0.getModelManager().getModel(UserDefinedBubbleDepartureDiameterModel.class);
BubbleDepartureDiameterProfile
bubbleDepartureDiameterProfile_0 =
 	
userDefinedBubbleDepartureDiameterModel_0.getBubbleDepartureDiameterProfile();
bubbleDepartureDiameterProfile_0.getMethod(ConstantScalarProfileMethod.class).getQuantity().setValue(1.0);
ColeBubbleDepartureFrequencyModel
 	
coleBubbleDepartureFrequencyModel_1 =
 	
phaseInteraction_0.getModelManage
	bubbleDepartureFrequencyProfile_0.getMethod(ConstantScalarProfileMethod.class).getQuantity().setValue(1.0);
	TransientConductionModel
	transientConductionModel_1 =
	
	phaseInteraction_0.getModelManager().getModel(TransientConductionModel.class);
	BubbleInfluenceWallAreaFractionProfile
	bubbleInfluenceWallAreaFractionProfile_0 =
	
	transientConductionModel_1.getBubbleInfluenceWallAreaFractionProfile();
	bubbleInfluenceWallAreaFractionProfile_0.setMethod(ConstantScalarProfileMethod.class);
	bubbleInfluenceWallAreaFractionProfile_0.getMethod(ConstantScalarProfileMethod.class).getQuantity().setValue(1.0);
	TransientConductionHeatTransferCoefficientProfile
	transientConductionHeatTransferCoefficientProfile_0 =
	
	transientConductionModel_1.getTransientConductionHeatTransferCoefficientProfile();
	transientConductionHeatTransferCoefficientProfile_0.setMethod(ConstantScalarProfileMethod.class);
	transientConductionHeatTransferCoefficientProfile_0.getMethod(ConstantScalarProfileMethod.class).getQuantity().setValue(1.0);

Previous Release	STAR-CCM+ v12.04

r().getModel(ColeBubbleDepartureFrequencyModel.class);
phaseInteraction_0.disableModel(coleBubbleDepartureFrequencyModel_1);
phaseInteraction_0.enable(UserDefinedBubbleDepartureFrequencyModel.class);
UserDefinedBubbleDepartureFrequencyModel
userDefinedBubbleDepartureFrequencyModel_0 =
 	
phaseInteraction_0.getModelManager().getModel(UserDefinedBubbleDepartureFrequencyModel.class);
BubbleDepartureFrequencyProfile
bubbleDepartureFrequencyProfile_0 =
 	
userDefinedBubbleDepartureFrequencyModel_0.getBubbleDepartureFrequencyProfile();
bubbleDepartureFrequencyProfile_0.getMethod(ConstantScalarProfileMethod.class).getQuantity().setValue(1.0);
KurulPodowskiBubbleInfluenceWallAreaFractionModel
kurulPodowskiBubbleInfluenceWallAreaFractionModel_1 =
 	
phaseInteraction_0.getModelManager().getModel(KurulPodowskiBubbleInfluenceWallAreaFractionModel.class);
phaseInteraction_0.disableModel(kurulPodowskiBubbleInfluenceWallAreaFractionModel_1);
phaseInteraction_0.enable(UserDefinedBubbleInfluenceWallAreaFractionModel.class);
UserDefinedBubbleInfluenceWallAreaFractionModel
userDefinedBubbleInfluenceWallAre	

Previous Release	STAR-CCM+ v12.04

aFractionModel_0 =
 	
phaseInteraction_0.getModelManager().getModel(UserDefinedBubbleInfluenceWallAreaFractionModel.class);
BubbleInfluenceWallAreaFractionProfile bubbleInfluenceWallAreaFractionProfile_0 =
 	
userDefinedBubbleInfluenceWallAreaFractionModel_0.getBubbleInfluenceWallAreaFractionProfile();
bubbleInfluenceWallAreaFractionProfile_0.getMethod(ConstantScalarProfileMethod.class).getQuantity().setValue(1.0);
DelValleKenningBubbleInducedQuenchingHeatTransferCoefficientModel delValleKenningBubbleInducedQuenchingHeatTransferCoefficientModel_1 =
 	
phaseInteraction_0.getModelManager().getModel(DelValleKenningBubbleInducedQuenchingHeatTransferCoefficientModel.class);
phaseInteraction_0.disableModel(delValleKenningBubbleInducedQuenchingHeatTransferCoefficientModel_1);
phaseInteraction_0.enable(UserDefinedBubbleInducedQuenchingHeatTransferCoefficientModel.class);
UserDefinedBubbleInducedQuenchingHeatTransferCoefficientModel userDefinedBubbleInducedQuenchingHeatTransferCoefficientModel_0 =
 	
phaseInteraction_0.getModelManager().getModel(UserDefinedBubbleInducedQuenchingHeatTransferCoefficientModel.class);
BubbleInducedQuenchingHeatTransfe	

Previous Release	STAR-CCM+ v12.04

`rCoefficientProfile bubbleInducedQuenchingHeatTransferCoefficientProfile_0 =`
 	
`userDefinedBubbleInducedQuenchingHeatTransferCoefficientModel_0.getBubbleInducedQuenchingHeatTransferCoefficientProfile(); bubbleInducedQuenchingHeatTransferCoefficientProfile_0.getMethod(ConstantScalarProfileMethod.class).getQuantity().setValue(1.0);`	

Select Basic Wall Dryout Area Fraction Model and Change Its Parameters

Previous Release	STAR-CCM+ v12.04
`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Coolant")); MultiPhaseInteractionModel multiPhaseInteractionModel_0 =` `physicsContinuum_0.getModelManager().getModel(MultiPhaseInteractionModel.class); PhaseInteraction phaseInteraction_0 = ((PhaseInteraction) multiPhaseInteractionModel_0.getPhaseInteractionManager().getPhaseInteraction("Liquid Vapour Interface")); phaseInteraction_0.enable(BasicWallDryoutAreaFractionModel.class); BasicWallDryoutAreaFractionModel basicWallDryoutAreaFractionModel_1 =`  `phaseInteraction_0.getModelManager().getModel(BasicWallDryoutAreaFractionModel.class); basicWallDryoutAreaFractionModel_`	`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Coolant")); MultiPhaseInteractionModel multiPhaseInteractionModel_0 =` `physicsContinuum_0.getModelManager().getModel(MultiPhaseInteractionModel.class); PhaseInteraction phaseInteraction_0 = ((PhaseInteraction) multiPhaseInteractionModel_0.getPhaseInteractionManager().getPhaseInteraction("Liquid Vapour Interface")); WallBoilingModel wallBoilingModel_0 =` `phaseInteraction_0.getModelManager().getModel(WallBoilingModel.class); WallDryoutAreaFractionProfile wallDryoutAreaFractionProfile_0 =`

Previous Release	STAR-CCM+ v12.04

`1.getWallDryoutBreakpoint().setValue(0.8);`	`wallBoilingModel_0.getWallDryoutAreaFractionProfile();`
	`wallDryoutAreaFractionProfile_0.setMethod(BasicWallDryOutAreaFractionProfileMethod.class);`
	`wallDryoutAreaFractionProfile_0.getMethod(BasicWallDryOutAreaFractionProfileMethod.class).getWallDryoutBreakpoint().setValue(0.8);`

Select Cole Model

Previous Release	STAR-CCM+ v12.04

`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Coolant"));`
`MultiPhaseInteractionModel multiPhaseInteractionModel_0 = physicsContinuum_0.getModelManager().getModel(MultiPhaseInteractionModel.class);`
`PhaseInteraction phaseInteraction_0 = ((PhaseInteraction) multiPhaseInteractionModel_0.getPhaseInteractionManager().getPhaseInteraction("Liquid Vapour Interface"));`
`phaseInteraction_0.enable(ColeBubbleDepartureFrequencyModel.class);`
	`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Coolant"));`
	`MultiPhaseInteractionModel multiPhaseInteractionModel_0 = physicsContinuum_0.getModelManager().getModel(MultiPhaseInteractionModel.class);`
	`PhaseInteraction phaseInteraction_0 = ((PhaseInteraction) multiPhaseInteractionModel_0.getPhaseInteractionManager().getPhaseInteraction("Liquid Vapour Interface"));`
	`BubbleNucleationModel bubbleNucleationModel_0 = phaseInteraction_0.getModelManager().getModel(BubbleNucleationModel.class);`
	`BubbleDepartureFrequencyProfile bubbleDepartureFrequencyProfile_0 = bubbleNucleationModel_0.getBubbleDepartureFrequencyProfile();`
	`bubbleDepartureFrequencyProfile_0`

Previous Release	STAR-CCM+ v12.04
	`.setMethod(ColeBubbleDepartureFrequencyProfileMethod.class);`

Select Del Valle Kenning and Change Its Parameters

Previous Release	STAR-CCM+ v12.04

`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Coolant"));`
`MultiPhaseInteractionModel multiPhaseInteractionModel_0 = physicsContinuum_0.getModelManager().getModel(MultiPhaseInteractionModel.class);`
`PhaseInteraction phaseInteraction_0 = ((PhaseInteraction) multiPhaseInteractionModel_0.getPhaseInteractionManager().getPhaseInteraction("Liquid Vapour Interface"));`
`phaseInteraction_0.enable(DelValleKenningBubbleInducedQuenchingHeatTransferCoefficientModel.class);`
`DelValleKenningBubbleInducedQuenchingHeatTransferCoefficientModel delValleKenningBubbleInducedQuenchingHeatTransferCoefficientModel_1 = phaseInteraction_0.getModelManager().getModel(DelValleKenningBubbleInducedQuenchingHeatTransferCoefficientModel.class);`
`delValleKenningBubbleInducedQuenchingHeatTransferCoefficientModel_1.getWaitCoefficient().setValue(0.800001);`
	`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Coolant"));`
	`MultiPhaseInteractionModel multiPhaseInteractionModel_0 = physicsContinuum_0.getModelManager().getModel(MultiPhaseInteractionModel.class);`
	`PhaseInteraction phaseInteraction_0 = ((PhaseInteraction) multiPhaseInteractionModel_0.getPhaseInteractionManager().getPhaseInteraction("Liquid Vapour Interface"));`
	`TransientConductionModel transientConductionModel_0 = phaseInteraction_0.getModelManager().getModel(TransientConductionModel.class);`
	`TransientConductionHeatTransferCoefficientProfile transientConductionHeatTransferCoefficientProfile_0 = transientConductionModel_0.getTransientConductionHeatTransferCoefficientProfile();`
	`transientConductionHeatTransferCoefficientProfile_0.setMethod(DelValleKenningBubbleInducedQuenchingHeatTransferCoefficientProfileMethod.class);`
	`transientConductionHeatTransferCo`

Previous Release	STAR-CCM+ v12.04
	`efficientProfile_0.getMethod(DelValleKenningBubbleInducedQuenchingHeatTransferCoefficientProfileMethod.class).getWaitCoefficient().setValue(0.800001);`

Select Hibiki Ishii Submodel and Change Its Parameters

Previous Release	STAR-CCM+ v12.04

`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Coolant"));`
`MultiPhaseInteractionModel multiPhaseInteractionModel_0 =`
 	
`physicsContinuum_0.getModelManager().getModel(MultiPhaseInteractionModel.class);`
`PhaseInteraction phaseInteraction_0 = ((PhaseInteraction) multiPhaseInteractionModel_0.getPhaseInteractionManager().getPhaseInteraction("Liquid Vapour Interface"));`
`phaseInteraction_0.enable(HibikiIshiiNucleationSiteNumberDensityModel.class);`
`HibikiIshiiNucleationSiteNumberDensityModel hibikiIshiiNucleationSiteNumberDensityModel_0 =`
 	
`phaseInteraction_0.getModelManager().getModel(HibikiIshiiNucleationSiteNumberDensityModel.class);`
`hibikiIshiiNucleationSiteNumberDensityModel_0.getAverageCavityDensity().setValue(472000.00001);`
`hibikiIshiiNucleationSiteNumberDensityModel_0.getWallContactAngle().setValue(0.7219999);`
	`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Coolant"));`
	`MultiPhaseInteractionModel multiPhaseInteractionModel_0 =`
	
	`physicsContinuum_0.getModelManager().getModel(MultiPhaseInteractionModel.class);`
	`PhaseInteraction phaseInteraction_0 = ((PhaseInteraction) multiPhaseInteractionModel_0.getPhaseInteractionManager().getPhaseInteraction("Liquid Vapour Interface"));`
	`BubbleNucleationModel bubbleNucleationModel_0 =`
	
	`phaseInteraction_0.getModelManager().getModel(BubbleNucleationModel.class);`
	`NucleationSiteNumberDensityProfile nucleationSiteNumberDensityProfile_0 =`
	
	`bubbleNucleationModel_0.getNucleationSiteNumberDensityProfile();`
	`nucleationSiteNumberDensityProfile_0.setMethod(HibikiIshiiNucleationSiteNumberDensityProfileMethod.class);`

Previous Release	STAR-CCM+ v12.04

`hibikiIshiiNucleationSiteNumberDensityModel_0.getWallContactAngleScale().setValue(0.7219999);`
`hibikiIshiiNucleationSiteNumberDensityModel_0.getCavityLengthScale().setValue(2.4999E-6);`
`hibikiIshiiNucleationSiteNumberDensityModel_0.getDensityFunctionConstantC0().setValue(-0.010639999);`
`hibikiIshiiNucleationSiteNumberDensityModel_0.getDensityFunctionConstantC1().setValue(0.482459999);`
`hibikiIshiiNucleationSiteNumberDensityModel_0.getDensityFunctionConstantC2().setValue(-0.227119999);`
`hibikiIshiiNucleationSiteNumberDensityModel_0.getDensityFunctionConstantC3().setValue(0.054679999);`
`hibikiIshiiNucleationSiteNumberDensityModel_0.getMaximumSuperheat().setValue(25.000001);`
	`nucleationSiteNumberDensityProfile_0.getMethod(HibikiIshiiNucleationSiteNumberDensityProfileMethod.class).getAverageCavityDensity().setValue(472000.00001);`
	`nucleationSiteNumberDensityProfile_0.getMethod(HibikiIshiiNucleationSiteNumberDensityProfileMethod.class).getWallContactAngle().setValue(0.7219999);`
	`nucleationSiteNumberDensityProfile_0.getMethod(HibikiIshiiNucleationSiteNumberDensityProfileMethod.class).getWallContactAngleScale().setValue(0.7219999);`
	`nucleationSiteNumberDensityProfile_0.getMethod(HibikiIshiiNucleationSiteNumberDensityProfileMethod.class).getCavityLengthScale().setValue(2.4999E-6);`
	`nucleationSiteNumberDensityProfile_0.getMethod(HibikiIshiiNucleationSiteNumberDensityProfileMethod.class).getDensityFunctionConstantC0().setValue(-0.010639999);`
	`nucleationSiteNumberDensityProfile_0.getMethod(HibikiIshiiNucleationSiteNumberDensityProfileMethod.class).getDensityFunctionConstantC1().setValue(0.482459999);`
	`nucleationSiteNumberDensityProfile_0.getMethod(HibikiIshiiNucleationSiteNumberDensityProfileMethod.class).getDensityFunctionConstantC2().setValue(-0.227119999);`
	`nucleationSiteNumberDensityProfile_0.getMethod(HibikiIshiiNucleationSiteNumberDensityProfileMethod.class).getDensityFunctionConstantC3().setValue(0.054679999);`
	`nucleationSiteNumberDensityProfile_0.getMethod(HibikiIshiiNucleationSiteNumberDensityProfileMethod.class).getMaximumSuperheat().setValue(25.000001);`

Select Kocamustafaogullari Model and Change Its Parameters

Previous Release	STAR-CCM+ v12.04

`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Coolant"));`
`MultiPhaseInteractionModel multiPhaseInteractionModel_0 = physicsContinuum_0.getModelManager().getModel(MultiPhaseInteractionModel.class);`
`PhaseInteraction phaseInteraction_0 = ((PhaseInteraction) multiPhaseInteractionModel_0.getPhaseInteractionManager().getPhaseInteraction("Liquid Vapour Interface"));`
`phaseInteraction_0.enable(KocamustafaogullariBubbleDepartureDiameterModel.class);`
`KocamustafaogullariBubbleDepartureDiameterModel kocamustafaogullariBubbleDepartureDiameterModel_0 = phaseInteraction_0.getModelManager().getModel(KocamustafaogullariBubbleDepartureDiameterModel.class);`
`kocamustafaogullariBubbleDepartureDiameterModel_0.getCalibrationConstant().setValue(0.0015125999);`
`kocamustafaogullariBubbleDepartureDiameterModel_0.getWallContactAngle().setValue(0.72200001);`
	`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Coolant"));`
	`MultiPhaseInteractionModel multiPhaseInteractionModel_0 = physicsContinuum_0.getModelManager().getModel(MultiPhaseInteractionModel.class);`
	`PhaseInteraction phaseInteraction_0 = ((PhaseInteraction) multiPhaseInteractionModel_0.getPhaseInteractionManager().getPhaseInteraction("Liquid Vapour Interface"));`
	`BubbleNucleationModel bubbleNucleationModel_0 = phaseInteraction_0.getModelManager().getModel(BubbleNucleationModel.class);`
	`BubbleDepartureDiameterProfile bubbleDepartureDiameterProfile_0 = bubbleNucleationModel_0.getBubbleDepartureDiameterProfile();`
	`bubbleDepartureDiameterProfile_0.setMethod(KocamustafaogullariBubbleDepartureDiameterProfileMethod.class);`
	`bubbleDepartureDiameterProfile_0.getMethod(KocamustafaogullariBubbleDepartureDiameterProfileMethod.class).getCalibrationConstant().setValue(0.0015125999);`
	`bubbleDepartureDiameterProfile_0.getMethod(KocamustafaogullariBubbleDepartureDiameterProfileMethod.class).getWallContactAngle().setValue(0.72200001);`

Select Kurul Podowski Model and Change Its Parameters

Previous Release	STAR-CCM+ v12.04

`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Coolant"));`
`MultiPhaseInteractionModel multiPhaseInteractionModel_0 = physicsContinuum_0.getModelManager().getModel(MultiPhaseInteractionModel.class);`
`PhaseInteraction phaseInteraction_0 = ((PhaseInteraction) multiPhaseInteractionModel_0.getPhaseInteractionManager().getPhaseInteraction("Liquid Vapour Interface"));`
`phaseInteraction_0.enable(KurulPodowskiBubbleInfluenceWallAreaFractionModel.class);`
`KurulPodowskiBubbleInfluenceWallAreaFractionModel kurulPodowskiBubbleInfluenceWallAreaFractionModel_1 = phaseInteraction_0.getModelManager().getModel(KurulPodowskiBubbleInfluenceWallAreaFractionModel.class);`
`kurulPodowskiBubbleInfluenceWallAreaFractionModel_1.getAreaCoefficient().setValue(2.000001);`
	`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Coolant"));`
	`MultiPhaseInteractionModel multiPhaseInteractionModel_0 = physicsContinuum_0.getModelManager().getModel(MultiPhaseInteractionModel.class);`
	`PhaseInteraction phaseInteraction_0 = ((PhaseInteraction) multiPhaseInteractionModel_0.getPhaseInteractionManager().getPhaseInteraction("Liquid Vapour Interface"));`
	`TransientConductionModel transientConductionModel_0 = phaseInteraction_0.getModelManager().getModel(TransientConductionModel.class);`
	`BubbleInfluenceWallAreaFractionProfile bubbleInfluenceWallAreaFractionProfile_0 = transientConductionModel_0.getBubbleInfluenceWallAreaFractionProfile();`
	`bubbleInfluenceWallAreaFractionProfile_0.setMethod(KurulPodowskiBubbleInfluenceWallAreaFractionProfileMethod.class);`
	`bubbleInfluenceWallAreaFractionProfile_0.getMethod(KurulPodowskiBubbleInfluenceWallAreaFractionProfileMethod.class).getAreaCoefficient().setValue(2.000001);`

Select Lemmert Chawla Submodel and Change Its Parameters

Previous Release	STAR-CCM+ v12.04

`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Coolant"));`
`MultiPhaseInteractionModel multiPhaseInteractionModel_0 =`
 	
`physicsContinuum_0.getModelManager().getModel(MultiPhaseInteractionModel.class);`
`PhaseInteraction phaseInteraction_0 = ((PhaseInteraction) multiPhaseInteractionModel_0.getPhaseInteractionManager().getPhaseInteraction("Liquid Vapour Interface"));`
`phaseInteraction_0.enable(LemmartChawlaNucleationSiteNumberDensityModel.class);`
`LemmartChawlaNucleationSiteNumberDensityModel lemmartChawlaNucleationSiteNumberDensityModel_1 =`
 	
`phaseInteraction_0.getModelManager().getModel(LemmartChawlaNucleationSiteNumberDensityModel.class);`
`lemmartChawlaNucleationSiteNumberDensityModel_1.getCalibrationConstantN0().setValue(12366.448);`
`lemmartChawlaNucleationSiteNumberDensityModel_1.getCalibrationConstantDT0.setValue(1.0001);`
`lemmartChawlaNucleationSiteNumberDensityModel_1.getCalibrationConstantA().setValue(1.80499);`
`lemmartChawlaNucleationSiteNumberDensityModel_1.getCalibrationConstantB().setValue(1.0E-6);`
`lemmartChawlaNucleationSiteNumberDensityModel_1.getMaximumSuperheat().setValue(25.000001);`
	`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Coolant"));`
	`MultiPhaseInteractionModel multiPhaseInteractionModel_0 =`
	
	`physicsContinuum_0.getModelManager().getModel(MultiPhaseInteractionModel.class);`
	`PhaseInteraction phaseInteraction_0 = ((PhaseInteraction) multiPhaseInteractionModel_0.getPhaseInteractionManager().getPhaseInteraction("Liquid Vapour Interface"));`
	`BubbleNucleationModel bubbleNucleationModel_0 =`
	
	`phaseInteraction_0.getModelManager().getModel(BubbleNucleationModel.class);`
	`NucleationSiteNumberDensityProfile nucleationSiteNumberDensityProfile_0 =`
	
	`bubbleNucleationModel_0.getNucleationSiteNumberDensityProfile();`
	`nucleationSiteNumberDensityProfile_0.setMethod(LemmartChawlaNucleationSiteNumberDensityProfileMethod.class);`
	`nucleationSiteNumberDensityProfile_0.getMethod(LemmartChawlaNucleationSiteNumberDensityProfileMethod.class).getCalibrationConstantN0().setValue(12366.448);`
	`nucleationSiteNumberDensityProfile_0.getMethod(LemmartChawlaNucleationSiteNumberDensityProfileMethod.class).getCalibrationConstantDT0().setValue(1.0001);`

Previous Release	STAR-CCM+ v12.04

	`nucleationSiteNumberDensityProfile_0.getMethod(LemmartChawlaNucleationSiteNumberDensityProfileMethod.class).getCalibrationConstantA().setValue(1.80499);`
	`nucleationSiteNumberDensityProfile_0.getMethod(LemmartChawlaNucleationSiteNumberDensityProfileMethod.class).getCalibrationConstantB().setValue(1.0E-6);`
	`nucleationSiteNumberDensityProfile_0.getMethod(LemmartChawlaNucleationSiteNumberDensityProfileMethod.class).getMaximumSuperheat().setValue(25.000001);`

Select Tolubinsky Kostanchuk Model and Change Its Parameters

Previous Release	STAR-CCM+ v12.04

`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Coolant"));`
`MultiPhaseInteractionModel multiPhaseInteractionModel_0 = physicsContinuum_0.getModelManager().getModel(MultiPhaseInteractionModel.class);`
`PhaseInteraction phaseInteraction_0 = ((PhaseInteraction) multiPhaseInteractionModel_0.getPhaseInteractionManager().getPhaseInteraction("Liquid Vapour Interface"));`
`phaseInteraction_0.enable(TolubinskyKonstanchukBubbleDepartureDiameterModel.class);`
`TolubinskyKonstanchukBubbleDepartureDiameterModel tolubinskyKonstanchukBubbleDepartureDiameterModel_1 =`
	`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Coolant"));`
	`MultiPhaseInteractionModel multiPhaseInteractionModel_0 = physicsContinuum_0.getModelManager().getModel(MultiPhaseInteractionModel.class);`
	`PhaseInteraction phaseInteraction_0 = ((PhaseInteraction) multiPhaseInteractionModel_0.getPhaseInteractionManager().getPhaseInteraction("Liquid Vapour Interface"));`
	`BubbleNucleationModel bubbleNucleationModel_0 = phaseInteraction_0.getModelManager().getModel(BubbleNucleationModel.class);`
	`BubbleDepartureDiameterProfile`

Previous Release	STAR-CCM+ v12.04

`phaseInteraction_0.getModelManager().getModel(TolubinskyKonstanchukBubbleDepartureDiameterModel.class);`
`tolubinskyKonstanchukBubbleDepartureDiameterModel_1.getReferenceDiameter().setValue(6.000001E-4);`
`tolubinskyKonstanchukBubbleDepartureDiameterModel_1.getReferenceDiameterSubcooling().setValue(45.000001);`
`tolubinskyKonstanchukBubbleDepartureDiameterModel_1.getMinimumDiameter().setValue(2.500001E-5);`
`tolubinskyKonstanchukBubbleDepartureDiameterModel_1.getMaximumDiameter().setValue(0.001400001);`
	`bubbleDepartureDiameterProfile_0 =`
	
	`bubbleNucleationModel_0.getBubbleDepartureDiameterProfile();`
	`bubbleDepartureDiameterProfile_0.setMethod(TolubinskyKostanchukBubbleDepartureDiameterProfileMethod.class);`
	`bubbleDepartureDiameterProfile_0.getMethod(TolubinskyKostanchukBubbleDepartureDiameterProfileMethod.class).getReferenceDiameter().setValue(6.00001E-4);`
	`bubbleDepartureDiameterProfile_0.getMethod(TolubinskyKostanchukBubbleDepartureDiameterProfileMethod.class).getReferenceDiameterSubcooling().setValue(45.00001);`
	`bubbleDepartureDiameterProfile_0.getMethod(TolubinskyKostanchukBubbleDepartureDiameterProfileMethod.class).getMinimumDiameter().setValue(2.500001E-5);`
	`bubbleDepartureDiameterProfile_0.getMethod(TolubinskyKostanchukBubbleDepartureDiameterProfileMethod.class).getMaximumDiameter().setValue(0.001400001);`

Electromagnetics: Changes to Magnetic Flux Reporting

The Magnetic Flux Linkage Surface Integral report is not available in v12.04. As a result, the command `createMagneticFluxLinkageReport` is no longer valid.

Update your macros as follows:

- For Finite Element cases, use the `MagneticFluxLinkageReport` based on a vector integral.
- For Finite Volume cases, use the `VolumeIntegralReport` of the field function `SpecificMagneticFluxLinkage`.

Alternatively, to recreate the original behavior of the Magnetic Flux Linkage Surface Integral report manually, use a Surface Integral report and a user-defined field function that calculates the dot product of the `MagneticFluxDensity` and the unit surface element.

Porous Media: Changes to Profile

With the code-restructuring involved in the implementation of Porous Media model, the location of `ScalarVolumeFractionProfile` has changed from `star.multiphase.ScalarVolumeFractionProfile` to `star.flow.ScalarVolumeFractionProfile`. This affects the `import` action near the beginning of a macro, as shown in the following example:

Previous Release	STAR-CCM+ v12.04

`import star.multiphase.ScalarVolumeFractionProfile;`
 	
`public void execute() {`
	`import star.flow.ScalarVolumeFractionProfile;`
	
	`public void execute() {`

Electrochemistry: Changes to Tafel Slope (log 10) Reacting Method

Input parameters for **Tafel Slope (log10)** have been changed. For the macro code, in the previous version, if the `TafelSlopeReactionMethod` was used, a `UserSpecificExchangeCurrentMethod` needed to be populated. This method has been replaced with a `SpecificExchangeCurrentLogMethod`. For consistent results, feed the value for the `SpecificExchangeCurrentLogMethod` with the decadic logarithm of the value of the `UserSpecificExchangeCurrentMethod`.

Previous Release	STAR-CCM+ v12.04

`UserSpecificExchangeCurrentMethod iMethod =`
 	
`reaction.getReactionProperties().get(SpecificExchangeCurrentProperty.class).getMethodObject(UserSpecificExchangeCurrentMethod.class);`
 	
`UserSpecificExchangeCurrentProfile iProfile = iMethod.getUserSpecificExchangeCurrentProfile();`
`iProfile.getMethod(ConstantScalarProfileMethod.class).getQuantity().setValue(legacyValue);`
	`SpecificExchangeCurrentLogMethod yilcMethod =`
	
	`reaction.getReactionProperties().get(SpecificExchangeCurrentLogProperty.class).getMethodObject(SpecificExchangeCurrentLogMethod.class);`
	
	`SpecificExchangeCurrentLogProfile yilcProfile = yilcMethod.getSpecificExchangeCurrentLogProfile();`
	`yilcProfile.getMethod(ConstantScalarProfileMethod.class).getQuantity().setValue(Math.log10(legacyValue);`

Solid Stress: New Material Law Model

A new material law model has been introduced in solid stress modeling. As a result, macro code that includes `FeSolidStressModel` needs updating. The following example illustrates the changes:

Previous Release	STAR-CCM+ v12.04

`physicsContinuum_0.enable(SteadyModel.class);`
`physicsContinuum_0.enable(FeSolidStressModel.class);`
`physicsContinuum_0.enable(LinearIsotropicElasticModel.class);`
`SolidModel solidModel_0 = physicsContinuum_0.getModelManager().getModel(SolidModel.class);`
`Solid solid_0 = ((Solid) solidModel_0.getMaterial());`
`ConstantMaterialPropertyMethod constantMaterialPropertyMethod_0 =`
 	
`((ConstantMaterialPropertyMethod)`
 	
`solid_0.getMaterialProperties().getMaterialProperty(ConstantDensityProperty.class).getMethod());`
`constantMaterialPropertyMethod_0.getQuantity().setValue(8000.0);`
`ConstantMaterialPropertyMethod constantMaterialPropertyMethod_1 =`
 	
`((ConstantMaterialPropertyMethod)`
 	
`solid_0.getMaterialProperties().getMaterialProperty(PoissonRatioProperty.class).getMethod());`
`constantMaterialPropertyMethod_1.getQuantity().setValue(0.3);`
`ConstantMaterialPropertyMethod constantMaterialPropertyMethod_2 =`
 	
`((ConstantMaterialPropertyMethod)`
 	
`solid_0.getMaterialProperties().getMaterialProperty(YoungsModulusP`
	`physicsContinuum_0.enable(SteadyModel.class);`
	`physicsContinuum_0.enable(SolidModel.class);`
	`physicsContinuum_0.enable(FeSolidStressModel.class);`
	`physicsContinuum_0.enable(FeMaterialLawsModel.class);`
	`FeMaterialLawsModel feMaterialLawsModel_1 =`
	
	`physicsContinuum_0.getModelManager().getModel(FeMaterialLawsModel.class);`
	`MaterialLaw materialLaw_0 = feMaterialLawsModel_1.createMaterialLaw();`
	`materialLaw_0.enable(LinearElasticMaterialModel.class);`
	`materialLaw_0.enable(IsotropicLinearElasticMaterialModel.class);`
	`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum(getContinuumName()));`
	`SolidModel solidModel_0 = physicsContinuum_0.getModelManager().getModel(SolidModel.class);`
	`FeMaterialLawsModel feMaterialLawsModel_1 = physicsContinuum_0.getModelManager().getModel(FeMaterialLawsModel.class);`
	`MaterialLaw materialLaw_0 = ((MaterialLaw)feMaterialLawsModel_1.getPhaseManager().getPhase("Material Law 1"));`
	`Solid solid_0 = ((Solid) solidModel_0.getMaterial());`

Previous Release	STAR-CCM+ v12.04

`roperty.class).getMethod());`
`constantMaterialPropertyMethod_2.`
`getQuantity().setValue(200.0);`
	`solid_0.getMaterialProperties().g`
	`etMaterialProperty(MaterialLawPro`
	`perty.class).setMaterialLaw(mater`
	`ialLaw_0);`
	`ConstantMaterialPropertyMethod`
	`constantMaterialPropertyMethod_Yo`
	`ungsModulus =`
	
	`((ConstantMaterialPropertyMethod)`
	
	`solid_0.getMaterialProperties().g`
	`etMaterialProperty(YoungsModulusP`
	`roperty.class).getMethod());`
	`constantMaterialPropertyMethod_Yo`
	`ungsModulus.getQuantity().setValu`
	`e(200.0);`
	`ConstantMaterialPropertyMethod`
	`constantMaterialPropertyMethod_Po`
	`issonsRatio =`
	
	`((ConstantMaterialPropertyMethod)`
	
	`solid_0.getMaterialProperties().g`
	`etMaterialProperty(PoissonRatioPr`
	`operty.class).getMethod());`
	`constantMaterialPropertyMethod_Po`
	`issonsRatio.getQuantity().setValu`
	`e(0.3);`
	`ConstantMaterialPropertyMethod`
	`constantMaterialPropertyMethod_De`
	`nsity =`
	
	`((ConstantMaterialPropertyMethod)`
	
	`solid_0.getMaterialProperties().g`
	`etMaterialProperty(ConstantDensit`
	`yProperty.class).getMethod());`
	`constantMaterialPropertyMethod_De`
	`nsity.getQuantity().setValue(8000`
	`.0);`

Co-Simulation

Changes to Java Classes

The Java classes `StarccmplusCoSimulation` and `AbaqusCoSimulation` have both been replaced by a single class, `CoSimulation`. Within macro code, all occurrences of these classes must be renamed accordingly.

If the macro is setting up a new co-simulation, the default name for the co-simulation object will now be `Link 1` instead of `STAR-CCM+ Co-Simulation 1` or `Abaqus Co-Simulation 1`. This is not needed if the macro is used on a simulation that had the co-simulation object already set up.

Previous Release	STAR-CCM+ v12.04
`StarccmplusCoSimulation abaqusCoSimulation_0 = ((StarccmplusCoSimulation) simulation_0.get(CoSimulationManager.class).getObject("STAR-CCM+ Co-Simulation 1"));`	`CoSimulation CoSimulation_0 = ((CoSimulation) simulation_0.get(CoSimulationManager.class).getObject("Link 1"));`
`AbaqusCoSimulation abaqusCoSimulation_0 = ((AbaqusCoSimulation) simulation_0.get(CoSimulationManager.class).getObject("Abaqus Co-Simulation 1"));`	`CoSimulation CoSimulation_0 = ((CoSimulation) simulation_0.get(CoSimulationManager.class).getObject("Link 1"));`

Change to Object for 1D Co-Simulation

The object `star.cosimulation.onedcoupling` has been renamed to `star.cosimulation.common`. This affects the import action near the beginning of a macro, as shown in the following example:

Previous Release	STAR-CCM+ v12.04
`import star.cosimulation.onedcoupling.OneDScalarFieldProfileMethod;`	`import star.cosimulation.common.OneDScalarFieldProfileMethod;`

Package Hierarchy Restructured for 1D Co-Simulations

The package hierarchy for 1D co-simulations has been restructured, resulting in changes to the macro code.

<u>Amesim</u>

For any macro files containing Amesim co-simulation client-server objects the following line must be included to import them:

```
import star.cosimulation.amesim.*;
```

Alternatively, if there are specific class imports, they need to be changed accordingly. The following example is for `AmesimModel`:

Previous Release	STAR-CCM+ v12.04
`import star.cosimulation.onedcoupling.AmesimModel;`	`import star.cosimulation.amesim.AmesimModel;`

Affected classes: `AmesimModel`, `AmesimCoSimulationExplicitUnsteadyModel`, `Amesim`, `AmesimPortAssociationManager`, and `AmesimSolver`.

GT-SUITE

For any macro files containing GT-SUITE co-simulation client-server objects the following line must be included to import them:

```
import star.cosimulation.gtpower.*;
```

Alternatively, if there are specific class imports, they need to be changed accordingly. The following example is for `GtVersion`:

Previous Release	STAR-CCM+ v12.04
`import star.cosimulation.onedcoupling.GtVersion;`	`import star.cosimulation.gtpower.GtVersion;`

Affected classes: `GtGasModel`, `GtLiquidModel`, `GtMultiComponentGasModel`, `GtPowerCoSimulationExplicitUnsteadyModel`, `GtSuiteLocalInputFile`, `GtSuiteRemoteInputFile`, `GtInformation`, `GtPower`, `GtSolver`, `GtVersion`, and `GtZoneData`.

OLGA

For any macro files containing OLGA co-simulation client-server objects the following line must be included to import them:

```
import star.cosimulation.olga.*;
```

Alternatively, if there are specific class imports, they need to be changed accordingly. The following example is for `OlgaVersion`:

Previous Release	STAR-CCM+ v12.04
`import star.cosimulation.onedcoupling.OlgaVersion;`	`import star.cosimulation.olga.OlgaVersion;`

Affected classes: `Olga`, `Olga7Location`, `Olga7LocationsManager`, `OlgaAssumeCircularBoundaries`, `OlgaCertRootDir`, `OlgaCoSimulationExplicitUnsteadyModel`, `OlgaCommunicationProtocol`,

`OlgaDensityMethod`, `OlgaDrdpMethod`, `OlgaDrdtMethod`, `OlgaDynamicViscosityMethod`, `OlgaEmpModel`, `OlgaEmpPhaseModel`, `OlgaEosModel`, `OlgaLocalInputFile`, `OlgaModelBase`, `OlgaNodeType`, `OlgaPhaseModel`, `OlgaPositionName`, `OlgaPropertiesFile`, `OlgaPropertyMethods`, `OlgaRemoteInputFile`, `OlgaSnapFiles`, `OlgaSolver`, `OlgaSpecificHeatMethod`, `OlgaSurfaceTensionMethod`, `OlgaSurfaceTensionModel`, `OlgaThermalConductivityMethod`, `OlgaVersion`, and `OlgaVofModel`.

Changes to WAVE Coupling Objects

WAVE co-simulation has been refactored for the current release.

WAVE Client Server Objects (CSOs) Moved

These are changes due to moving classes from `star.cosimulation.onedcoupling` to `star.cosimulation.wave`.

For any macro files containing WAVE co-simulation client-server objects the following line must be included to import them:

```
import star.cosimulation.wave.*;
```

Alternatively, if there are specific class imports, they need to be changed accordingly. The following example is for `WaveModel`:

Previous Release	STAR-CCM+ v12.04
`import star.cosimulation.onedcoupling.WaveModel;`	`import star.cosimulation.wave.WaveModel;`

Affected classes: `WaveModel.java`, `WaveCoSimulationExplicitUnsteadyModel.java`, and `WaveControlPinAssociationManager.java`.

WaveNode Removed

The CSO `WaveNode` has been replaced with the `CoSimulation` class.

Import line:

Previous Release	STAR-CCM+ v12.04

`import star.cosimulation.onedcoupling.*;`
`// or`
`import star.cosimulation.onedcoupling.WaveNode;`
	`import star.cosimulation.common.*;`
	`// or`
	`import star.cosimulation.common.CoSimulation;`

Getting the `WaveNode` replaced by getting the `CoSimulation`:

Previous Release	STAR-CCM+ v12.04
`WaveNode waveNode_0 = ((WaveNode) simulation_0.get(CoSimulationManager.class).getObject("WAVE 1"));`	`CoSimulation coSimulation_0 = ((CoSimulation) simulation_0.get(CoSimulationManager.class).getObject("WAVE 1"));`

Editing the WAVE input file property:

Previous Release	STAR-CCM+ v12.04

`waveNode_0.setWaveCaseName("WaveFile");`	`WaveLocalInputFile waveLocalInputFile_0 =`
	
	`coSimulation_0.getCoSimulationValues().get(WaveLocalInputFile.class);`
	`waveLocalInputFile_0.setFilePath("sims/wave/tube/tubechild.wvm");`

Editing the WAVE executable command property:

Previous Release	STAR-CCM+ v12.04

`waveNode_0.setWaveCommand("wave");`	`CoSimCommandLine coSimCommandLine_0 =`
	
	`coSimulation_0.getCoSimulationValues().get(CoSimCommandLine.class);`
	`coSimCommandLine_0.setCommandLine("wave");`

Editing the `RSimlink` libraries path property:

Previous Release	STAR-CCM+ v12.04

`waveNode_0.setWaveLibPath("RsimlinkLibraryPath");`	`RSimlinkLibPath rSimlinkLibPath_0 =`
	
	`coSimulation_0.getCoSimulationValues().get(RSimlinkLibPath.class);`
	`rSimlinkLibPath_0.setFilePath("RsimlinkLibraryPath");`

WaveSpecies Removed

`WaveSpecies` and `WaveSpeciesManager` CSO classes were replaced with `CoupledSpeciesComponentMap` and `CoupledSpeciesManager`.

Import line:

Previous Release	STAR-CCM+ v12.04

`import star.cosimulation.onedcoupling.*;`
`// or`
`import star.cosimulation.onedcoupling.WaveSpecies;`
`import star.cosimulation.onedcoupling.WaveSpeciesManager;`
	`import star.cosimulation.common.*;`
	`// or`
	`import star.cosimulation.common.CoupledSpecies;`
	`import star.cosimulation.common.CoupledSpeciesManager;`

Getting a `WaveSpecies` replaced by getting a `CoupledSpeciesComponentMap`:

Previous Release	STAR-CCM+ v12.04

`WaveSpecies waveSpecies_0 = ((WaveSpecies) waveNode_0.getWaveSpeciesManager().getObject("air"));`	`CoupledSpeciesComponentMap coupledSpeciesComponentMap_0 =`
	
	`coSimulation_0.getCoSimulationValues().get(CoupledSpeciesManager.class).getCoupledSpeciesComponentMap("air");`

Setting a coupled species:

Previous Release	STAR-CCM+ v12.04
`waveSpecies_0.setMaterialPropertiesHolder(passiveScalarMaterial_0);`	`coupledSpeciesComponentMap_0.setCoupledComponent(passiveScalarMaterial_0);`

WaveZones Removed

`WaveZone` and `WaveZoneManager` CSO classes were replaced with `CoSimulationZone` and `CoSimulationZoneManager`.

Import line:

Previous Release	STAR-CCM+ v12.04
`import star.cosimulation.onedcoupling.*;` `// or`	`import star.cosimulation.common.*;` `// or`

Previous Release	STAR-CCM+ v12.04
`import star.cosimulation.onedcoupling.WaveZone;` `import star.cosimulation.onedcoupling.WaveZoneManager;`	`import star.cosimulation.common.CoSimulationZone;` `import star.cosimulation.common.CoSimulationZoneManager;`

Getting the `WaveZone` replaced by getting the `CoSimulationZone`:

Previous Release	STAR-CCM+ v12.04

`WaveZone waveZone_0 =`
`  ((WaveZone) waveNode_0.getWaveZoneManager().getObject("WAVE-1"));`
	`CoSimulationZone coSimulationZone_0 =`
	
	`coSimulation_0.getCoSimulationZoneManager().getCoSimulationZone("WAVE-1");`

Coupling a boundary to a `WaveZone` (previous release) and to a `CoSimulationZone` (current release):

Previous Release	STAR-CCM+ v12.04

`boundary_0.getConditions().get(OneDOption.class).setSelected(OneDOption.Type.COUPLED);`
`waveZone_0.getCcmBoundaryPartGroup().setQuery(null);`
`waveZone_0.getCcmBoundaryPartGroup().setObjects(boundary_0);`
	`CoupledParts coupledParts_0 =`
	
	`coSimulationZone_0.getCoSimulationZoneConditions().get(CoupledParts.class);`
	`coupledParts_0.clear();`
	`coupledParts_0.add(boundary_0);`

Uncoupling a boundary from a `WaveZone` (previous release) and from a `CoSimulationZone` (current release):

Previous Release	STAR-CCM+ v12.04

`boundary_0.getConditions().get(OneDOption.class).setSelected(OneDOption.Type.UNCOUPLED);`
`waveZone_0.getCcmBoundaryPartGroup().setQuery(null);`
`waveZone_0.getCcmBoundaryPartGroup().setObjects();`
	`coupledParts_0.erase(boundary_0);`

Changes to Coupling Options

Co-simulation coupling options have been refactored, resulting in changes to the macro code.

Concurrency Option

The `ConcurrencyMethodManager` condition has been replaced by the `CoSimulationConcurrencyOption`.

Previous Release	STAR-CCM+ v12.04

`// get cosimulation StarccmplusCoSimulation starccmplusCoSimulation_0 = ((StarccmplusCoSimulation) simulation_0.get(CoSimulationManager.class)`
`.getObject("STAR-CCM+ Co-Simulation 1"));`
 	
`// Set this as leading cosimulation starccmplusCoSimulation_0.getCoSimulationConditions()`
`.get(ConcurrencyMethodManager.class)`
`.setMethod(LeadMethod.class);`
 	
`// Set this as lagging cosimulation starccmplusCoSimulation_0.getCoSimulationConditions()`
`.get(ConcurrencyMethodManager.class)`
`.setMethod(LagMethod.class);`
	`// get cosimulation StarccmplusCoSimulation starccmplusCoSimulation_0 = ((StarccmplusCoSimulation) simulation_0.get(CoSimulationManager.class)`
	`.getObject("STAR-CCM+ Co-Simulation 1"));`
	
	`// Set this as leading cosimulation starccmplusCoSimulation_0.getCoSimulationConditions()`
	`.get(CoSimulationConcurrencyOption.class)`
	`.setSelected(CoSimulationConcurrencyOption.Type.LEAD);`
	
	`// Set this as lagging cosimulation starccmplusCoSimulation_0.getCoSimulationConditions()`
	`.get(CoSimulationConcurrencyOption.class)`
	`.setSelected(CoSimulationConcurrencyOption.Type.LAG);`

Coupling Option

The `NegotiationMethodManager` condition has been replaced by the `CoSimulationCouplingOption`.

Previous Release	STAR-CCM+ v12.04
`// get cosimulation StarccmplusCoSimulation starccmplusCoSimulation_0 = ((StarccmplusCoSimulation)`	`// get cosimulation StarccmplusCoSimulation starccmplusCoSimulation_0 = ((StarccmplusCoSimulation)`

Previous Release	STAR-CCM+ v12.04

`simulation_0.get(CoSimulationManager.class)`
 	
`.getObject("STAR-CCM+ Co-Simulation 1"));`
`//--------`
`// Set constant coupling method`
`starccmplusCoSimulation_0.getCoSimulationConditions()`
`.get(NegotiationMethodManager.class)`
`.setMethod(ConstantNegotiationMethod.class);`
`// set constant coupling timestep value to 0.1s`
`((ConstantNegotiationMethod) starccmplusCoSimulation_0.getCoSimulationConditions()`
 	
`.get(NegotiationMethodManager.class).getMethod())`
 	
`.getConstantTimeStep()`
 	
`.setValue(0.1);`
`//--------`
`// Set minimum coupling method`
`starccmplusCoSimulation_0.getCoSimulationConditions()`
`.get(NegotiationMethodManager.class)`
`.setMethod(MinNegotiationMethod.class);`
`//--------`
`// Set maximum coupling method`
`starccmplusCoSimulation_0.getCoSimulationConditions()`
`.get(NegotiationMethodManager.class)`
`.setMethod(MaxNegotiationMethod.class);`
	`simulation_0.get(CoSimulationManager.class)`
	
	`.getObject("STAR-CCM+ Co-Simulation 1"));`
	`//--------`
	`// Set constant coupling method`
	`starccmplusCoSimulation_0.getCoSimulationConditions()`
	`.get(CoSimulationCouplingOption.class)`
	`.setSelected(CoSimulationCouplingOption.Type.CONSTANT);`
	`// set constant coupling timestep value to 0.1s`
	`starccmplusCoSimulation_0.getCoSimulationValues()`
	`.get(CoSimCouplingTimeStep.class)`
	`.getCouplingTimeStep()`
	`.setValue(0.1);`
	`//--------`
	`// Set minimum coupling method`
	`starccmplusCoSimulation_0.getCoSimulationConditions()`
	`.get(CoSimulationCouplingOption.class)`
	`.setSelected(CoSimulationCouplingOption.Type.MINIMUM);`
	`//--------`
	`// Set maximum coupling method`
	`starccmplusCoSimulation_0.getCoSimulationConditions()`
	`.get(CoSimulationCouplingOption.class)`
	`.setSelected(CoSimulationCouplingOption.Type.MAXIMUM);`
	`//--------`
	`// Set import coupling method`
	`starccmplusCoSimulation_0.getCoSi`

Previous Release	STAR-CCM+ v12.04

`//--------`
`// Set import coupling method`
`starccmplusCoSimulation_0.getCoSimulationConditions()`
`.get(NegotiationMethodManager.class)`
`.setMethod(ImportNegotiationMethod.class);`
`//--------`
`// Set export coupling method`
`starccmplusCoSimulation_0.getCoSimulationConditions()`
`.get(NegotiationMethodManager.class)`
`.setMethod(ExportNegotiationMethod.class);`
`//--------`
`// Set independent coupling method`
`starccmplusCoSimulation_0.getCoSimulationConditions()`
`.get(NegotiationMethodManager.class)`
`.setMethod(IndependentNegotiationMethod.class);`
`// set independent coupling timestep value to 0.1s`
`((IndependentNegotiationMethod)`
`starccmplusCoSimulation_0.getCoSimulationConditions()`
`.get(NegotiationMethodManager.class).getMethod())`
`.getConstantTimeStep()`
`.setValue(0.1);`
	`mulationConditions()`
	`.get(CoSimulationCouplingOption.class)`
	`.setSelected(CoSimulationCouplingOption.Type.IMPORT);`
	`//--------`
	`// Set export coupling method`
	`starccmplusCoSimulation_0.getCoSimulationConditions()`
	`.get(CoSimulationCouplingOption.class)`
	`.setSelected(CoSimulationCouplingOption.Type.EXPORT);`
	`//--------`
	`// Set independent coupling method`
	`starccmplusCoSimulation_0.getCoSimulationConditions()`
	`.get(CoSimulationCouplingOption.class)`
	`.setSelected(CoSimulationCouplingOption.Type.INDEPENDENT);`
	`// set independent coupling timestep value to 0.1s`
	`starccmplusCoSimulation_0.getCoSimulationValues()`
	`.get(CoSimCouplingTimeStep.class)`
	`.getCouplingTimeStep()`
	`.setValue(0.1);`

Mapping Option

The `MappingMethodManager` condition has been replaced by the use of the pre-existing `MappingMethodOption`, originally only used by Abaqus co-simulations.

Previous Release	STAR-CCM+ v12.04

`// get cosimulation StarccmplusCoSimulation starccmplusCoSimulation_0 = ((StarccmplusCoSimulation) simulation_0.get(CoSimulationManager.class)`
 	
`.getObject("STAR-CCM+ Co-Simulation 1"));`
 	
`// Set Map Before Export starccmplusCoSimulation_0.getCoSimulationConditions() .get(MappingMethodManager.class) .setMethod(Mode1MappingMethod.class);`
 	
`// Set Map After Import starccmplusCoSimulation_0.getCoSimulationConditions() .get(MappingMethodManager.class) .setMethod(Mode2MappingMethod.class);`
 	
`// Set Map Both Directions starccmplusCoSimulation_0.getCoSimulationConditions() .get(MappingMethodManager.class) .setMethod(Mode3MappingMethod.class);`
 	
`// Set Partner Maps Both Directions starccmplusCoSimulation_0.getCoSimulationConditions() .get(MappingMethodManager.class) .setMethod(Mode4MappingMethod.class);`
	`// get cosimulation StarccmplusCoSimulation starccmplusCoSimulation_0 = ((StarccmplusCoSimulation) simulation_0.get(CoSimulationManager.class)`
	
	`.getObject("STAR-CCM+ Co-Simulation 1"));`
	
	`// Set Map Before Export starccmplusCoSimulation_0.getCoSimulationConditions() .get(MappingMethodOption.class) .setSelected(MappingMethodOption.Type.MAP_BEFORE_EXPORT);`
	
	`// Set Map After Import starccmplusCoSimulation_0.getCoSimulationConditions() .get(MappingMethodOption.class) .setSelected(MappingMethodOption.Type.MAP_AFTER_IMPORT);`
	
	`// Set Map Both Directions starccmplusCoSimulation_0.getCoSimulationConditions() .get(MappingMethodOption.class) .setSelected(MappingMethodOption.Type.MAP_BOTH_DIRECTIONS);`
	
	`// Set Partner Maps Both Directions starccmplusCoSimulation_0.getCoSimulationConditions() .get(MappingMethodOption.class)`

Previous Release	STAR-CCM+ v12.04
	`.setSelected(MappingMethodOption.Type.PARTNER_MAPS_BOTH_DIRECTIONS);`

Macro API Changes 12.02

In STAR-CCM+ 12.02, the macro API changed for morphing motion, Eulerian multiphase, meshing, and reacting flows.

Morphing Motion: External STL File Replaced with Part Surface

For boundaries, the **Morpher Constraint Specification** constraint option **STL Surface** has been replaced with a more general **Part Surface** option.

In the macro code, the java class option `ConstraintSpecification.Type.SLIDE_ON_STL_SURFACE` has been replaced with `ConstraintSpecification.Type.SLIDE_ON_PART_SURFACE)`.

An example of the changes to the macro code follows:

Previous Release	STAR-CCM+ v12.02

`boundary_0.getConditions()`
`.get(MorpherSpecification.class)`
`.setSelected(MorpherSpecification`
`.Type.CONSTRAINT);`
 	
`boundary_0`
`.getConditions()`
`.get(ConstraintSpecification.class)`
`.setSelected(ConstraintSpecification.Type.SLIDE_ON_STL_SURFACE);`
 	
`GuideSurfaceFileSettings`
`guideSurfaceFileSettings_0 =`
`boundary_0`
`.getValues()`
`.get(GuideSurfaceFileSettings.class);`
 	
`guideSurfaceFileSettings_0`
`.setFilePath("/users/iridium2/`
`dmitry/guide_surface.stl");`
	`// Step 1. Import STL file as a`
	`Part`
	`Units units_0 =`
	`simulation_0`
	`.getUnitsManager()`
	`.getPreferredUnits(new`
	`IntVector(new int[] {0, 1, 0, 0,`
	`0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,`
	`0, 0, 0, 0, 0, 0, 0, 0, 0, 0}));`
	
	`PartImportManager`
	`partImportManager_0 =`
	`simulation_0`
	`.get(PartImportManager.class);`
	
	`partImportManager_0`
	`.importStlPart(resolvePath("/`
	`users/iridium2/dmitry/`
	`guide_surface.stl"),`
	`"OneSurfacePerPatch", units_0,`
	`true, 1.0E-5);`
	
	`Boundary boundary_0 =`

Previous Release	STAR-CCM+ v12.02

	`region_0`
	`.getBoundaryManager()`
	`.getBoundary("car-bdy");`
	
	`// Step 2. Use Part surface`
	
	`boundary_0.getConditions()`
	`.get(MorpherSpecification.class)`
	`.setSelected(MorpherSpecification`
	`.Type.CONSTRAINT);`
	
	`boundary_0`
	`.getConditions()`
	`.get(ConstraintSpecification.clas`
	`s)`
	`.setSelected(ConstraintSpecificat`
	`ion.Type.SLIDE_ON_PART_SURFACE);`
	
	`PartGuideSurfaceSettings`
	`partGuideSurfaceSettings_0 =`
	`boundary_0.getValues()`
	`.get(PartGuideSurfaceSettings.cla`
	`ss);`
	
	`MeshPart meshPart_0 =`
	`((MeshPart) simulation_0`
	`.get(SimulationPartManager.class)`
	`.getPart("guide_surface"));`
	
	`PartSurface partSurface_0 =`
	`((PartSurface) meshPart_0`
	`.getPartSurfaceManager()`
	`.getPartSurface("Surface"));`
	
	`partGuideSurfaceSettings_0`
	`.getGeometryObjectGroup()`
	`.setObjects(partSurface_0);`

Eulerian Multiphase: Changes to Suspension and Emulsion Rheology Models

For the Suspension and Emulsion Rheology models, phase interaction material properties have been restructured, resulting in changes to the macro code. The objects that were named based on their method are now named for the type of phase model.

The following example is the Krieger and Dougherty Relative Viscosity method used with the Emulsion Rheology model.

Previous Release	STAR-CCM+ v12.02

`KriegerRelativeViscosityMethod kViscosityMethod_0 = ((KriegerRelativeViscosityMethod) phaseInteractionMaterial_0 .getMaterialProperties() .getMaterialProperty(RelativeViscosityProperty.class) .getMethod());`
 	
`kViscosityMethod_0 .getMaximumPacking() .setValue(maxPacking);`
	`EmulsionRheologyModel emulsionRheologyModel_0 = phaseInteraction_0 .getModelManager() .getModel(EmulsionRheologyModel.class);`
	
	`emulsionRheologyModel_0 .getMaxPackingRheology() .setValue(maxPacking);`

The following example is the Morris and Boulay Relative Viscosity method used with the Suspension Rheology model.

Previous Release	STAR-CCM+ v12.02

`MorrisAndBoulayRelativeViscosityMethod morrisAndBoulayRelativeViscosityMethod_0 = ((MorrisAndBoulayRelativeViscosityMethod) phaseInteractionMaterial_0 .getMaterialProperties() .getMaterialProperty(RelativeViscosityProperty.class) .getMethod());`
 	
`morrisAndBoulayRelativeViscosityMethod_0 .getMaximumPacking() .setValue(maxPacking);`
	`SuspensionRheologyModel suspensionRheologyModel_0 = phaseInteraction_0 .getModelManager() .getModel(SuspensionRheologyModel .class);`
	
	`suspensionRheologyModel_0 .getMaxPackingRheology() .setValue(phimax);`

Meshing: Changes to Boolean Operation Macros

Macro commands for Boolean operations in surface repair have been enhanced to allow retention of tool faces for subtract operations.

Note: Boolean operation macros that worked in the previous version of STAR-CCM+ are still supported. For long-term reliability, it is recommended that you upgrade your macros with the methods of the current version.

Examples of the changes to the macro code follow:

Boolean Unite

Previous Release	STAR-CCM+ v12.02

`surfaceMeshWidgetIntersectOptions_0.setIntersectMode(1);`
`surfaceMeshWidgetIntersectOptions_0.setBooleanType(0); // Unite`
`surfaceMeshWidgetIntersectController_0.intersectFaces();`
	`surfaceMeshWidgetIntersectOptions_0.setIntersectMode(1);`
	`surfaceMeshWidgetIntersectOptions_0.setBooleanType(0);`
	`surfaceMeshWidgetIntersectOptions_0.setKeepToolFaces(false);`
	`surfaceMeshWidgetIntersectOptions_0.setRetainedFacesMode(0);`
	`surfaceMeshWidgetIntersectController_0.booleanFaces(null);`

Boolean Intersect

Previous Release	STAR-CCM+ v12.02

`surfaceMeshWidgetIntersectOptions_0.setIntersectMode(1);`
`surfaceMeshWidgetIntersectOptions_0.setBooleanType(1); // Intersect`
`surfaceMeshWidgetIntersectController_0.intersectFaces();`
	`surfaceMeshWidgetIntersectOptions_0.setIntersectMode(1);`
	`surfaceMeshWidgetIntersectOptions_0.setBooleanType(1);`
	`surfaceMeshWidgetIntersectOptions_0.setKeepToolFaces(false);`
	`surfaceMeshWidgetIntersectOptions_0.setRetainedFacesMode(0);`
	`surfaceMeshWidgetIntersectController_0.booleanFaces(null);`

Boolean Subtract

Previous Release	STAR-CCM+ v12.02

`surfaceMeshWidgetIntersectOptions_0.setIntersectMode(1);`
`surfaceMeshWidgetIntersectOptions_0.setBooleanType(2); // Subtract`
`surfaceMeshWidgetIntersectController_0.intersectFaces();`
	`surfaceMeshWidgetIntersectOptions_0.setIntersectMode(1);`
	`surfaceMeshWidgetIntersectOptions_0.setBooleanType(2);`
	`surfaceMeshWidgetIntersectOptions_0.setKeepToolFaces(false);`
	`surfaceMeshWidgetIntersectOptions_0.setRetainedFacesMode(0);`
	`surfaceMeshWidgetIntersectController_0.booleanFaces(null);`

Reacting Flows

Changes Due to Restructuring

The objects of certain reacting flow models have been restructured, resulting in changes to the macro code.

Previous Release	STAR-CCM+ v12.02

`import star.segregatedflow.*;`
`import star.segregatedenergy.*;`
`import star.combustion.*;`
	`import star.segregatedflow.*;`
	`import star.segregatedenergy.*;`
	`import star.combustion.*;`
	`import star.reactions.*;`

Changes to Table Generation

The process of generating tables for soot models has been merged into the process of generating tables for combustion. Therefore remove options with `ppdfSootFlameletTable` from your macros. An example follows:

Previous Release	STAR-CCM+ v12.02

`SootMomentTableModel sootMomentModel_0 =`
 	
`physicsContinuum_0.getModelManager().getModel(SootMomentTableModel.class);`
`sootMomentModel_0.setThreshold(100);`
`PpdfSootFlameletTable ppdfSootFlameletTable_0 =`
`((PpdfSootFlameletTable) sootMomentModel_0.getPpdfSootFlameletTable());`
`PpdfTableAxis ppdfTableAxis_2 =`
`((PpdfTableAxis) ((PpdfTableAxisManager) ppdfSootFlameletTable_0.getPpdfTableAxisManager()).getComponent("Mixture Fraction Variance"));`
`ppdfTableAxis_2.setDimensionSize(1);`
`ppdfSootFlameletTable_0.importTable(resolveDataPath("case/sootFlameletLibrary.tbl"));`
	`SootMomentTableModel sootMomentModel_0 =`
	
	`physicsContinuum_0.getModelManager().getModel(SootMomentTableModel.class);`
	`sootMomentModel_0.setThreshold(100);`

There were additional changes to the Soot Moment model in 12.02, in particular some model options were deprecated, that may affect existing macros. Please consult the "Restoring the 11.06 Soot Moments Model" section in the 12.02 User Guide for more information.

Changes to Setting of Number of Flamelet Grid Points

The method has changed for setting the number of points for a flamelet grid (such as for Mixture Fraction, Mixture Fraction Variance, Heat loss, and Progress Variable), resulting in changes to the macro code. You can now set the number of points for grids as fixed (similar to v11.06) or adaptive. Adaptive is now the default option.

The following example is for the Chemical Equilibrium Combustion model:

Previous Release	STAR-CCM+ v12.02

`EquilibriumTableGenerator equilibriumTableGenerator_0 = physicsContinuum_0.get(EquilibriumTableGenerator.class);`
`PpdfEquilibriumTable ppdfEquilibriumTable_0 = ((PpdfEquilibriumTable) equilibriumTableGenerator_0.getPpdfEquilibriumTable());`
`PpdfTableAxis ppdfTableAxis_0 = ((PpdfTableAxis) ((PpdfTableAxisManager) ppdfEquilibriumTable_0.getPpdfTableAxisManager()).getComponent("Mixture Fraction"));`
`ppdfTableAxis_0.setDimensionSize(61);`
`PpdfTableAxis ppdfTableAxis_1 = ((PpdfTableAxis) ((PpdfTableAxisManager) ppdfEquilibriumTable_0.getPpdfTableAxisManager()).getComponent("Mixture Fraction Variance"));`
`ppdfTableAxis_1.setDimensionSize(31);`
	`EquilibriumTableGenerator equilibriumTableGenerator_0 = physicsContinuum_0.get(EquilibriumTableGenerator.class)`
	`EquilibriumTableParameters equilibriumTableParameters_0 = ((EquilibriumTableParameters) equilibriumTableGenerator_0.getTableParameters());`
	`EquilibriumNumericalSettings equilibriumNumericalSettings_0 = ((EquilibriumNumericalSettings)equilibriumTableParameters_0.getEquilibriumNumericalSettings());`
	`TableAxisParameters tableAxisParameters_1 = ((TableAxisParameters) equilibriumNumericalSettings_0.getTableAxisParametersManager().getComponent ("MixtureFraction"));`
	`// Set number of points to adaptive based on tolerance`
	`tableAxisParameters_0.setAdapt(true);`
	`AdaptiveGridParameters adaptiveGridParameters_0 = tableAxisParameters_0.getAdaptiveGridParameters();`
	`adaptiveGridParameters_0.setMaxDimensionSize(61);`
	`adaptiveGridParameters_0.setRangeTol(0.001);`
	`// Set number of points to non-adaptive (fixed)`
	`tableAxisParameters_0.setAdapt(false);`
	`FixedGridParameters`

Previous Release	STAR-CCM+ v12.02
	`fixedGridParameters_0 = tableAxisParameters_1.getFixedGridParameters();` `fixedGridParameters_0.setDimensionSize(61);`

Macro API Changes 11.06

In STAR-CCM+ 11.06, the macro API changed for electrochemistry, Eulerian multiphase, reacting flows, directed meshing, mesh deformation, client-server, and meshing.

Electrochemistry: Replacement of Tafel Method Classes

Electrochemical reactions modeling has been modified to account for the consumption of multi-component gas species that are involved in reactions. In the macro code, the `Combined` classes in the previous version have been replaced by their corresponding `Potential` classes:

Tafel

Previous Release	STAR-CCM+ v11.06
`TafelCombinedReactionMethod.class`	`TafelPotentialReactionMethod.class`

Tafel Slope

Previous Release	STAR-CCM+ v11.06
`TafelSlopeCombinedReactionMethod.class`	`TafelSlopePotentialReactionMethod.class`

Transport Limited Tafel Slope

Previous Release	STAR-CCM+ v11.06
`TransportLimitedTafelSlopeCombinedReactionMethod.class`	`TransportLimitedTafelSlopePotentialReactionMethod.class`

All occurrences of the combined variant have to be replaced with the potential variant. For example:

Previous Release	STAR-CCM+ v11.06
`TafelCombinedReactionMethod tafelReactionMethod = rs0.getReactionProperties().get(E`	`TafelPotentialReactionMethod tafelReactionMethod = rs0.getReactionProperties().get(E`

Previous Release	STAR-CCM+ v11.06
`lectrochemistryReactionFormulationProperty.class).getMethodObject(TafelCombinedReactionMethod.class);`	`lectrochemistryReactionFormulationProperty.class).getMethodObject(TafelPotentialReactionMethod.class);`

Note: Remove the following statements if they are present in your Java macro:

```
import
star.electrochemistry.electrochemicalreactions.TafelCombinedReactionMethod
import
star.electrochemistry.electrochemicalreactions.TransportLimitedTafelSlopeCombinedReactionMethod
import
star.electrochemistry.electrochemicalreactions.TafelSlopeCombinedReactionMethod
```

Eulerian Multiphase

- The objects of the Volume of Fluid (VOF) and Two-Phase Equilibrium models have been restructured, resulting in changes to the macro code for importing:
  - For VOF, `star.vof.AlphaFilmBoilingProfile` has changed to `star.mixturemultiphase.AlphaFilmBoilingProfile`.

    To accommodate this change in your macro, you can insert an extra import statement:

    ```
    import star.mixturemultiphase.AlphaFilmBoilingProfile;
    ```

    Note: Remove `import star.vof.AlphaFilmBoilingProfile` if it is present in your macro.
  - For Two-Phase Equilibrium, `mixturemultiphase` has changed to `twophaseequilibrium`.

    To accommodate this change in your macro, you can insert an extra import statement:

    ```
    import star.twophaseequilibrium.AlphaFilmBoilingProfile;
    ```
  - To ensure backward compatibility, all occurrences of `SegregatedVofPhaseSpeciesModel` should be changed to `MmpPhaseSpeciesModel`, and all occurrences of `VofPhaseEnergyModel` should be changed to `MmpPhaseEnergyModel`. Examples of the changes to the macro code follow:

Previous Release	STAR-CCM+ v11.06

`SegregatedVofPhaseSpeciesModel`
 	
`segregatedVofPhaseSpeciesModel`
	`MmpPhaseSpeciesModel`
	`mmpPhaseSpeciesModel_0 =`

Previous Release	STAR-CCM+ v11.06

`_0 =`
 	
`eulerianPhase_0.getModelManager().getModel(SegregatedVofPhaseSpeciesModel.class);`
`segregatedVofPhaseSpeciesModel_0.getUpwindOption().setSelected(ConvectiveFluxOption.Type.MUSCL_3RD_ORDER);`
	`eulerianPhase_0.getModelManager().getModel(MmpPhaseSpeciesModel.class);`
	`mmpPhaseSpeciesModel_0.getUpwindOption().setSelected(ConvectiveFluxOption.Type.MUSCL_3RD_ORDER);`

`VofPhaseEnergyModel vofPhaseEnergyModel_0 =`
 	
`eulerianPhase_0.getModelManager().getModel(VofPhaseEnergyModel.class);`
`vofPhaseEnergyModel_0.getUpwindOption().setSelected(ConvectiveFluxOption.Type.MUSCL_3RD_ORDER);`
	`MmpPhaseEnergyModel mmpPhaseEnergyModel_0 =`
	
	`eulerianPhase_0.getModelManager().getModel(MmpPhaseEnergyModel.class);`
	`mmpPhaseEnergyModel_0.getUpwindOption().setSelected(ConvectiveFluxOption.Type.MUSCL_3RD_ORDER);`

- Two field functions that were available with the Macro Porosity (pure thermal) model have been removed: **Void Phase Source of XY** and **Liquid Zone Volume Change Rate**.

  Macros from previous versions of STAR-CCM+ that activate the Macro Porosity (pure thermal) model must select the Field Function method for the Solid Cell Indicator to achieve the same behavior, as shown in the following example of the new macro code:

```
MacroPorosityPureThermalModel macroPorosityPureThermalModel_0 =

phaseInteraction_0.getModelManager().getModel(MacroPorosityPureThermalModel.class);
SolidCellIndicatorProfile solidCellIndicatorProfile_0 =
    ((SolidCellIndicatorProfile)
macroPorosityPureThermalModel_0.getSolidCellIndicatorProfile());
solidCellIndicatorProfile_0.setMethod(FunctionScalarProfileMethod.class);
```

Reacting Flows

Changes to Soot Moment Model

The Soot Moment model has been extended to work with complex chemistry, resulting in changes to the macro code.

Previous Release	STAR-CCM+ v11.06

`physicsContinuum_0.enable(SootModel.class);`
 	
`physicsContinuum_0.enable(SootMomentModel.class);`
 	
`physicsContinuum_0.enable(RadiationModel.class);`
`@@ -174,8 +174,8 @@ public class CoppalleFlameletMomentSoot extends Soot {`
 	
`constantMaterialPropertyMethod_0.getQuantity().setValue(0.1);`
`  SootMomentModel sootMomentModel_0 =`
 	
`physicsContinuum_0.getModelManager().getModel(SootMomentModel.class);`
 	
`ppdfSootFlameletTable_0.importTable(resolveDataPath("case/sootC2H2FlameletLibrary.tbl"));`
`  SootMomentModelProperties sootMomentModelProperties_0 =`
 	
`sootMomentModel_0.getSootMomentModelProperties();`
	`physicsContinuum_0.enable(SootModel.class);`
	
	`physicsContinuum_0.enable(SootMomentTableModel.class);`
	
	`physicsContinuum_0.enable(RadiationModel.class);`
	`@@ -174,8 +174,8 @@ public class CoppalleFlameletMomentSoot extends Soot {`
	
	`constantMaterialPropertyMethod_0.getQuantity().setValue(0.1);`
	`  SootMomentTableModel sootMomentModel_0 =`
	
	`physicsContinuum_0.getModelManager().getModel(SootMomentTableModel.class);`
	
	`ppdfSootFlameletTable_0.importTable(resolveDataPath("case/sootC2H2FlameletLibrary.tbl"));`
	`  SootMomentTableModelProperties sootMomentModelProperties_0 =`
	
	`sootMomentModel_0.getSootMomentTableModelProperties();`

Removal of UnburntTemperatureControlOption

The `UnburntTemperatureControlOption` object is no longer available. Therefore remove statements such as the following:

```
cfmCombustionModel_0.getUnburntTemperatureControlOption().setSelected(UnburntTemperatureControlOption.ISENTROPIC_RELATION);
```

New User Interface

For further details about the v11.06 changes, see New Features > Physics > CFD > Reacting Flows.

The classifications have been changed from:

```
PremixedCombustionFlameTypeModel
PartiallyPremixedCombustionFlameTypeModel
NonPremixedCombustionFlameTypeModel
```

to:

```
ReactingSpeciesTransportBasedModel
FlameletBasedModel
```

Therefore when your macro activates the Eddy Break-Up, Complex Chemistry, or Eddy Contact Micromixing model, make the following change in your macro code:

Previous Release	STAR-CCM+ v11.06
`physicsContinuum_0.enable(NonPremixedCombustionFlameTypeModel.class);` `physicsContinuum_0.enable(PartiallyPremixedCombustionFlameTypeModel.class);`	`physicsContinuum_0.enable(ReactingSpeciesTransportBasedModel.class);`

When your macro activates the PPDF Equilibrium (now Chemical Equilibrium), PPDF Flamelet (now Steady Laminar Flame, or SLF), or Flamelet Generated Manifold (FGM), make the following change in your macro code:

Previous Release	STAR-CCM+ v11.06
`physicsContinuum_0.enable(NonPremixedCombustionFlameTypeModel.class);` `physicsContinuum_0.enable(PartiallyPremixedCombustionFlameTypeModel.class);`	`physicsContinuum_0.enable(FlameletBasedModel.class);`

Changes to Flamelet Generation

The species order has been changed. The new species list is sorted alphanumerically.

Previous Release	STAR-CCM+ v11.06
`MassFractionProfile massFractionProfile_2 = boundary_1.getValues().get(MassFractionProfile.class);` `((ConstantArrayProfileMethod) massFractionProfile_2.getMethod()).getQuantity().setArray(new`	`MassFractionProfile massFractionProfile_2 = boundary_1.getValues().get(MassFractionProfile.class);` `((ConstantArrayProfileMethod) massFractionProfile_2.getMethod()).getQuantity().setArray(new`

Previous Release	STAR-CCM+ v11.06
`DoubleVector(new double[] {0.667, 0.0, 0.0,  0.0, 0.0, 0.0, 0.0,  0.0, 0.0,0.0, 0.333}));`	`DoubleVector(new double[] {0.333, 0.0, 0.0,  0.0, 0.0, 0.0, 0.0,  0.0, 0.667,0.0, 0}));`

The name of the Third Body Coefficient object has changed.

Previous Release	STAR-CCM+ v11.06
`ThirdBodyCoefficient thirdBodyCoefficient_0 = reaction_0.getReactionProperties().get(ThirdBodyCoefficients.class).addThirdBodyCoefficient(CH4); thirdBodyCoefficient_0.setValue(1.0);`	`ThirdBodyCoefficientMixtureComponent thirdBodyCoefficient_0 = reaction_0.getReactionProperties().get(ThirdBodyCoefficients.class).addThirdBodyCoefficient(CH4); thirdBodyCoefficient_0.setValue(1.0);`

Enabling the Thermal NOx model: previous models are now obsolete and there is only one Nox model that models thermal nox, Nox Zeldovich (`NoxThreeEquationZeldovichModel`).

Previous Release	STAR-CCM+ v11.06
`physicsContinuum_0.enable(ThermalNoxModel.class); physicsContinuum_0.enable(NoxPpdfEquilibriumModel.class); physicsContinuum_0.enable(EmissionsPpdfCombustionModel.class);`	`physicsContinuum_0.enable(ThermalNoxModel.class); physicsContinuum_0.enable(NoxThreeEquationZeldovichModel.class);`

Chemical Equilibrium

For the Chemical Equilibrium model, you do not need to set the default material database.

Previous Release	STAR-CCM+ v11.06

`PpdfFlameletTable ppdfFlameletTable_0 = ((PpdfFlameletTable) ppdfFlameletModel_0.getPpdfFlameletTable());`
 	
`ppdfFlameletTable_0.setDefaultMdb(materialDataBase_0); PpdfTableAxis ppdfTableAxis_0 =((PpdfTableAxis) ((PpdfTableAxisManager)`
	`PpdfFlameletTable ppdfFlameletTable_0 = ((PpdfFlameletTable) ppdfFlameletModel_0.getPpdfFlameletTable());`
	
	`PpdfTableAxis ppdfTableAxis_0 =((PpdfTableAxis) ((PpdfTableAxisManager) ppdfFlameletTable_0.getPpdfTableAxisManager()).getComponent("Mixtu`

Previous Release	STAR-CCM+ v11.06
`ppdfFlameletTable_0.getPpdfTableAxisManager()).getComponent("Mixture Fraction 0"));` `ppdfTableAxis_0.setDimensionSize(20);`	`re Fraction 0"));` `ppdfTableAxis_0.setDimensionSize(20);`

For the Chemical Equilibrium model, the equilibrium algorithm selection is automated.

Previous Release	STAR-CCM+ v11.06

`PpdfEquilibriumTable ppdfEquilibriumTable_0 =((PpdfEquilibriumTable) ppdfEquilibriumModel_0.getPpdfEquilibriumTable());`
`ppdfEquilibriumTable_0.getAlgorithmTypeOption().setSelected(PpdfEquilibriumAlgorithmType.DARS);`
`ppdfEquilibriumTable_0.setDeleteNonExistSpecEntries(true);`
`ppdfEquilibriumTable_0.setNumExtra(11);`
	`PpdfEquilibriumTable ppdfEquilibriumTable_0 =((PpdfEquilibriumTable) ppdfEquilibriumModel_0.getPpdfEquilibriumTable());`
	
	`ppdfEquilibriumTable_0.setNumExtra(11);`

The following example shows the specification of species for Chemical Equilibrium.

Previous Release	STAR-CCM+ v11.06

`MultiComponentGasModel multiComponentGasModel_0 =((MultiComponentGasModel) physicsContinuum_0.getModelManager().getModel(MultiComponentGasModel.class));`
`MixtureComponent CH4 = createMixtureComponent(multiComponentGasModel_0, "CH4");`
`MixtureComponent O2 = createMixtureComponent(multiComponentGasModel_0, "O2");`
`MixtureComponent N2 = createMixtureComponent(multiComponentGasModel_0, "N2");`
	`EquilibriumTableGenerator equilibriumTableGenerator_0 =physicsContinuum_0.get(EquilibriumTableGenerator.class);`
	`TableChemistryDefinition tableChemistryDefinition_0 = ((TableChemistryDefinition) equilibriumTableGenerator_0.getTableChemistryDefinition());`
	`MaterialDataBase materialDataBase_0 = simulation_0.get(MaterialDataBaseManager.class).getMaterialDataBase("props");`
	`DataBaseMaterialManager dataBaseMaterialManager_0 =materialDataBase_0.getFolder("Gases");`

Previous Release	STAR-CCM+ v11.06
	`DataBaseGas dataBaseGas_0 =((DataBaseGas) dataBaseMaterialManager_0.getMaterial("N2_Gas")); tableChemistryDefinition_0.addComponents(new NeoObjectVector(new Object[] {dataBaseGas_0}));`

Tables

When generating or using flamelets, include `tablegenerator` and `ppdftable` packages.

Previous Release	STAR-CCM+ v11.06

`import star.combustion.*;`
 	
`import star.common.*;`
	`import star.combustion.*; import star.combustion.ppdftable.*; import star.combustion.tablegenerators.*; import star.common.*;`

The object `constructTable` has been renamed to `constructFlameletTable`.

Previous Release	STAR-CCM+ v11.06
`PpdfFlameletModel ppdfFlameletModel_0 = ((PpdfFlameletModel) physicsContinuum_0.getModelManager().getModel(PpdfFlameletModel.class)); PpdfFlameletTable ppdfFlameletTable_ = ((PpdfFlameletTable) ppdfFlameletModel_0.getPpdfFlameletTable()); ppdfFlameletTable_0.constructTable(new StringVector(new String[] {resolvePath(path + "flamelet01.lib01"), resolvePath(path + "flamelet02.lib01"), resolvePath(path + "flamelet03.lib01")}));`	`PpdfFlameletModel ppdfFlameletModel_0 = ((PpdfFlameletModel) physicsContinuum_0.getModelManager().getModel(PpdfFlameletModel.class)); PpdfFlameletTable ppdfFlameletTable_ = ((PpdfFlameletTable) ppdfFlameletModel_0.getPpdfFlameletTable()); ppdfFlameletTable_0.constructFlameletTable(new StringVector(new String[]{resolvePath(path + "flamelet01.lib01"), resolvePath(path + "flamelet02.lib01"), resolvePath(path + "flamelet03.lib01")}));`

Fluid Streams

The Thickened Flame Model (TFM) requires additional commands for fluid streams.

Previous Release	STAR-CCM+ v11.06

`physicsContinuum_0.get(FluidStreamManager.class).createFluidStream();`
`FluidStream fluidStream_0 = ((FluidStream)fluidStreamManager.getFluidStream("Fluid Stream 1"));`
`fluidStream_0.getTemperature().setValue(314.0);`
	`TfmCombustionModel tfmCombustionModel_0 =((TfmCombustionModel) physicsContinuum_0.getModelManager().getModel(TfmCombustionModel.class));`
	`FluidStreamManager fluidStreamManager = tfmCombustionModel_0.getFluidStreamManager();`
	`fluidStreamManager.createFluidStream();`
	` FluidStream fluidStream_0 = ((FluidStream)fluidStreamManager.getFluidStream("Fluid Stream 1"));`
	`fluidStream_0.getTemperature().setValue(314.0);`

Fluid stream objects are now accessed through the `EquilibriumTableGenerator`, `FlameletTableGenerator` and `FGMTableGenerator` objects. The following example shows the specification of stream temperature:

Previous Release	STAR-CCM+ v11.06

`FluidStream fluidStream_0 =((FluidStream) physicsContinuum_0.get(FluidStreamManager.class).getFluidStream("Fuel Stream"));`
`fluidStream_0.getTemperature().setValue(314.0);`
	`EquilibriumTableGenerator equilibriumTableGenerator_0 =physicsContinuum_0.get(EquilibriumTableGenerator.class);`
	`TableFluidStreamCollection tableFluidStreams_0 =((TableFluidStreamCollection) equilibriumTableGenerator_0.getTableFluidStreamCollection());`
	`TableFluidStream tableFluidStream_1 =((TableFluidStream) tableFluidStreams_0.getFuelStream());`
	`tableFluidStream_1.getTemperature().setValue(314.0);`

The following example of fluid stream objects shows the specification of stream compositions:

Previous Release	STAR-CCM+ v11.06

`FluidStreamManager fsm = ((FluidStreamManager) physicsContinuum_0.get(FluidStreamManager.class));`
`MultiComponentGasModel multiComponentGasModel_0 =((MultiComponentGasModel) physicsContinuum_0.getModelManager().getModel(MultiComponentGasModel.class));`
`GasMixture gasMixture_0 =((GasMixture) multiComponentGasModel_0.getMixture());`
`GasComponent N2 =(GasComponent) gasMixture_0.getComponents().getComponent("N2");`
`FluidStream str2 = fsm.getFluidStream("Oxidizer Stream");`
`FluidStreamComponent oxidizerO2 = str2.getFluidStreamComponentManager().addFluidStreamComponent(O2);`
`oxidizerO2.setMassFraction(0.233);`
	`EquilibriumTableGenerator equilibriumTableGenerator_0 = physicsContinuum_0.get(EquilibriumTableGenerator.class);`
	`TableFluidStreamCollection tableFluidStreams_0 =((TableFluidStreamCollection) equilibriumTableGenerator_0.getTableFluidStreamCollection());`
	`TableFluidStream tableFluidStream_1 =((TableFluidStream) tableFluidStreams_0.getiOxidizerStream());`
	`tableFluidStream_1.getFluidStreamComposition().setArray(new DoubleVector(new double[] {0.0, 0.233, 0.0, 0.0, 0.0, 0.0, 0.767,, 0.0}));`

Setting the fluid stream for the Soot Two-Equation model:

Previous Release	STAR-CCM+ v11.06

`FluidStreamManager fsm = ((FluidStreamManager) physicsContinuum_0.get(FluidStreamManager.class));`
 	
` FluidStream str1 = fsm.getFluidStream("Fuel Stream");`
	`SootTwoEquationModel sootTwoEquationModel_0 = ((SootTwoEquationModel) physicsContinuum_0.getModelManager().getModel(SootTwoEquationModel.class));`
	`FluidStreamManager fsm = sootTwoEquationModel_0.getFluidStreamManager();`
	` FluidStream str1 = fsm.getFluidStream("Fuel Stream");`

Inert Stream manipulation:

Previous Release	STAR-CCM+ v11.06

`FluidStream fluidStream_0 =((FluidStream) physicsContinuum_0.get(FluidStreamManager.class).getFluidStream("Inert Stream "));`	`InertStreamModel inertStreamModel_0 =((InertStreamModel) physicsContinuum_0.getModelManager().getModel(InertStreamModel.class));`
	`FluidStreamManager fsm = inertStreamModel_0.getFluidStreamManager();`
	`FluidStream fluidStream_0 =((FluidStream) fsm.getFluidStream("Inert Stream "));`

Changes to Thickened Flame Model

Previous Release	STAR-CCM+ v11.06

`physicsContinuum_0.enable(NonPremixedCombustionFlameTypeModel.class);`
 	
`physicsContinuum_0.enable(TfmNotPremixedCombustionModel.class);`
	`physicsContinuum_0.enable(ReactingSpeciesTransportBasedModel.class);`
	
	`physicsContinuum_0.enable(TfmCombustionModel.class);`

`physicsContinuum_0.enable(PremixedCombustionFlameTypeModel.class);`
 	
`physicsContinuum_0.enable(TfmPremixedCombustionModel.class);`
	`physicsContinuum_0.enable(ReactingSpeciesTransportBasedModel.class);`
	
	`physicsContinuum_0.enable(TfmCombustionModel.class);`

Time-Step Control: Renaming of Solvers

The Time-Step Control solvers have been renamed, resulting in changes to the macro code.

Convective CFL Time-Step Control

Previous Release	STAR-CCM+ v11.06
`ConvectiveCflTimeStepControlSolver convectiveCflTimeStepControlSolver =`	`ConvectiveCflTimeStepControlSolver convectiveCflTimeStepControlSolver =`

Previous Release	STAR-CCM+ v11.06
((ConvectiveCflTimeStepControlSolver) ((TimeStepControlSubSolverManager) timeStepControlSolver.getTimeStepControlSubSolverManager()).getSolver("Convective CFL Time Step Control"));	((ConvectiveCflTimeStepControlSolver) ((TimeStepControlSubSolverManager) timeStepControlSolver.getTimeStepControlSubSolverManager()).getSolver("Convective CFL Time-Step Control"));

Energy-Based Time-Step Control

Previous Release	STAR-CCM+ v11.06
((EnergyBasedTimeStepControlSolver) ((TimeStepControlSubSolverManager) timeStepControlSolver.getTimeStepControlSubSolverManager()).getSolver("Energy Based Time Step Control"));	((EnergyBasedTimeStepControlSolver) ((TimeStepControlSubSolverManager) timeStepControlSolver.getTimeStepControlSubSolverManager()).getSolver("Energy-Based Time-Step Control"));

The following code example is for the Convective CFL Time-Step Control solver:

Previous Release	STAR-CCM+ v11.06

TimeStepControlSolver timeStepControlSolver = ((TimeStepControlSolver)getSimulation().getSolverManager().getSolver(TimeStepControlSolver.class));
timeStepControlSolver.getMinTimeStep().setValue(0.0005);
timeStepControlSolver.setVerbose(true);
 	
ConvectiveCflTimeStepControlSolver convectiveCflTimeStepControlSolver = ((ConvectiveCflTimeStepControlSolver)
((TimeStepControlSubSolverManager)
timeStepControlSolver.getTimeStep
	TimeStepControlSolver timeStepControlSolver = ((TimeStepControlSolver)getSimulation().getSolverManager().getSolver(TimeStepControlSolver.class));
	timeStepControlSolver.getMinTimeStep().setValue(0.0005);
	timeStepControlSolver.setVerbose(true);
	
	ConvectiveCflTimeStepControlSolver convectiveCflTimeStepControlSolver = ((ConvectiveCflTimeStepControlSolver)
	((TimeStepControlSubSolverManager)
	timeStepControlSolver.getTimeStep

Previous Release	STAR-CCM+ v11.06
`ControlSubSolverManager()).getSolver("Convective CFL Time Step Control")); convectiveCflTimeStepControlSolver.getTargetMeanCfl().setValue(1.0); convectiveCflTimeStepControlSolver.getTargetMaxCfl().setValue(1.5);`	`ControlSubSolverManager()).getSolver("Convective CFL Time-Step Control")); convectiveCflTimeStepControlSolver.getTargetMeanCfl().setValue(1.0); convectiveCflTimeStepControlSolver.getTargetMaxCfl().setValue(1.5);`

Directed Meshing: Changes to Definitions of Closed-Loop Part Collections

In closed-loop part collections, for example toroidal geometry with multiple parts, the technique for defining the source mesh has changed to ensure accuracy. If you use a simulation file or a Java macro from a previous version and notice inconsistency in the mesh definition, consider recording a new macro.

Mesh Deformation: Change from Control Points to Point Sets

For mesh deformation, point sets have replaced control points. Unlike control points, point sets are not linked to regions. The following examples show the types of changes to the macro code.

Create Each Type of Control Point/Point Set

Previous Release	STAR-CCM+ v11.06
`Simulation simulation_0 = getActiveSimulation(); ControlPointRegion controlPointRegion_0 = simulation_0.get(ControlPointRegionManager.class).createLatticeControlPoint("Lattice Control Points", new DoubleVector(new double[] {1.5, 2.4843961000442505E-5, 0.49999989196658134}), new DoubleVector(new double[] {3.0, 0.1973467841744423, 1.1979536786675453}), new DoubleVector(new double[] {0.0, 0.0, 1.0}), 0.0, 2, 2, 2); ControlPointRegion controlPointRegion_1 = simulation_0.get(ControlPointRegi`	`Simulation simulation_0 = getActiveSimulation(); PointSet pointSet_0 = simulation_0.get(PointSetManager.class).createLatticePointSet("Lattice Control Points", new DoubleVector(new double[] {1.5, 2.4843961000442505E-5, 0.49999989196658134}), new DoubleVector(new double[] {3.0, 0.1973467841744423, 1.1979536786675453}), new DoubleVector(new double[] {0.0, 0.0, 1.0}), 0.0, 2, 2, 2); PointSet pointSet_1 = simulation_0.get(PointSetManager.class).createLinePointSet("Line Control Points", new DoubleVector(new double[] {0.0,`

Previous Release	STAR-CCM+ v11.06

`onManager.class).createLineControlPoint("Line Control Points", new DoubleVector(new double[] {0.0, 0.0, 0.0}), new DoubleVector(new double[] {1.0, 0.0, 0.0}), 20);`
`Region region_0 =`
 	
`simulation_0.getRegionManager().getRegion("Region");`
`Boundary boundary_0 =`
 	
`region_0.getBoundaryManager().getBoundary("Wall");`
`FvRepresentation fvRepresentation_0 =`
`((FvRepresentation) simulation_0.getRepresentationManager().getObject("Volume Mesh"));`
`ControlPointRegion controlPointRegion_2 =`
 	
`simulation_0.get(ControlPointRegionManager.class).createPartControlPoint("Part Control Points", new NeoObjectVector(new Object[] {boundary_0}), 0.1, 1000, false, 0.0, fvRepresentation_0);`
`Table cpTable_0 =`
 	
`simulation_0.getTableManager().getTable("ControlPoints");`
`ControlPointRegion controlPointRegion_4 =`
`simulation_0.get(ControlPointRegionManager.class).createTableControlPoint("Table Control Points", cpTable_0, "X", "Y", "Z");`
	`0.0, 0.0}), new DoubleVector(new double[] {1.0, 0.0, 0.0}), 20);`
	`Region region_0 =`
	
	`simulation_0.getRegionManager().getRegion("Region");`
	`Boundary boundary_0 =`
	
	`region_0.getBoundaryManager().getBoundary("Wall");`
	`FvRepresentation fvRepresentation_0 =`
	`((FvRepresentation) simulation_0.getRepresentationManager().getObject("Volume Mesh"));`
	`PointSet pointSet_2 =`
	
	`simulation_0.get(PointSetManager.class).createPartPointSet("Part Control Points", new NeoObjectVector(new Object[] {boundary_0}), 0.1, 1000, false, 0.0, fvRepresentation_0);`
	`Table cpTable_0 =`
	
	`simulation_0.getTableManager().getTable("ControlPoints");`
	`PointSet pointSet_3 =`
	
	`simulation_0.get(PointSetManager.class).createTablePointSet("Table Control Points", cpTable_0, "X", "Y", "Z");`

Edit Lattice Point Generator

Previous Release	STAR-CCM+ v11.06
`ControlPointRegion controlPointRegion_0 =`	`PointSet pointSet_0 =` `((PointSet)`

Previous Release	STAR-CCM+ v11.06
```((ControlPointRegion) simulation_0.get(ControlPointRegionManager.class).getObject("Lattice Control Point Region")); LatticePointGenerator latticePointGenerator_0 = ((LatticePointGenerator) controlPointRegion_0.getPointGenerator()); latticePointGenerator_0.setNumberX(5); latticePointGenerator_0.regenerateControlPoints();``` ```simulation_0.get(PointSetManager.class).getObject("Lattice Point Set")); LatticePointGenerator latticePointGenerator_0 = ((LatticePointGenerator) pointSet_0.getPointGenerator()); latticePointGenerator_0.setNumberX(5); latticePointGenerator_0.regeneratePointSet

Previous Release	STAR-CCM+ v11.06
`partPointGenerator_0 = ((PartPointGenerator) controlPointRegion_2.getPointGenerator()); partPointGenerator_0.getSize().setValue(0.5); partPointGenerator_0.regenerateControlPoints();`	`((PartPointGenerator) pointSet_2.getPointGenerator()); partPointGenerator_0.getSize().setValue(0.5); partPointGenerator_0.regeneratePointSet();`

Edit Table Point Generator

Previous Release	STAR-CCM+ v11.06
`ControlPointRegion controlPointRegion_3 = ((ControlPointRegion) simulation_0.get(ControlPointRegionManager.class).getObject("Table Control Point Region")); TablePointGenerator tablePointGenerator_0 = ((TablePointGenerator) controlPointRegion_3.getPointGenerator()); Table newTable_0 = simulation_0.getTableManager().getTable("NewControlPoints")); tablePointGenerator_0.setTable(newTable_0); tablePointGenerator_0.regenerateControlPoints();`	`PointSet pointSet_3 = ((PointSet) simulation_0.get(PointSetManager.class).getObject("Table Point Set")); TablePointGenerator tablePointGenerator_0 = ((TablePointGenerator) pointSet_3.getPointGenerator()); Table newTable_0 = simulation_0.getTableManager().getTable("NewControlPoints")); tablePointGenerator_0.setTable(newTable_0); tablePointGenerator_0.regeneratePointSet();`

Activate/Deactivate

Previous Release	STAR-CCM+ v11.06
`ControlPointRegion controlPointRegion_3 = ((ControlPointRegion) simulation_0.get(ControlPointRegionManager.class).getObject("Table Control Point Region")); controlPointRegion_3.setIsActive(false); //Deactivate`	`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Physics 1")); PointSet pointSet_3 = ((PointSet) simulation_0.get(PointSetManager.`

Previous Release	STAR-CCM+ v11.06
`controlPointRegion_3.setIsActive(true); //Activate`	`class).getObject("Table Point Set")); physicsContinuum_0.erase(pointSet_3); //Deactivate by removing from continuum physicsContinuum_0.add(pointSet_3); // Activate by adding to continuum`

Client-Server: Changes to StarCommandParser API

The `StarCommandParser` is used to parse and modify an instance of a command line used to start the STAR-CCM+ server. Previously the API permitted you to modify the internal container that the `StarCommandParser` uses. In the current version of STAR-CCM+, changes to the API prevent you from directly modifying the internal data; however you may still modify the command that the `StarCommandParser` manages.

Instantiating the Command Parser

The `StarCommandParser` is instantiated in the same way:

```
StarCommandParser cmdParser = new StarCommandParser("starccm+ -server");
```

Adding a Single Option

The command

```
cmdParser.addLast("-cpubind");
```

has been deprecated. Use the existing alternative:

```
cmdParser.addOption("-cpubind");
```

Adding an Option and a Value

The following statement

```
cmdParser.addLast("-licpath");
cmdParser.addLast("/my/license/path");
```

has been replaced with the existing alternative:

```
cmdParser.addOptionAndValue("-licpath","/my/license/path");
```

Query Option

The command

```
cmdParser.indexOf("-cpubind");
```

can be replaced with

```
cmdParser.hasOption("-cpubind");
```

Removing a Single Option

The statement

```
int idx = cmdParser.indexOf("-cpubind");
if ( idx != -1 ) cmdParser.remove(idx);
```

can be replaced with

```
cmdParser.remove("-cpubind");
```

Removing an Option and a Value

To remove the first instance of `-licpath <value>` from the command line, the statement

```
int idx = cmdParser.indexOf("-licpath");
if ( idx != -1 ) {
  cmdParser.remove(idx);
  cmdParser.remove(idx);
}
```

can be replaced with

```
if ( cmdParser.hasOption("-licpath") ) {
  cmdParser.remove("-licpath", cmdParser.getArg("-licpath"));
}
```

To remove all instances of `-licpath <value>` from the command line:

```
cmdParser.removeAll("-licpath");
```

Replacing an Option

The statement

```
int idx = cmdParser.indexOf("-licpath");
if ( idx != -1 ) {
  cmdParser.set(idx+1, "/a/different/license/path");
}
```

can be replaced with

```
cmdParser.replaceOptionAndValue("-licpath", "/a/different/license/
path");
```

Meshing: Changes to Surface Preparation Macros

The Java macros produced when launching surface repair and wrapper leak detection have been shortened by removing unnecessary commands and introducing new commands where appropriate. The end result is that these macros are easier to read and modify while still maintaining functionality for older macros.

Note: Macros that worked in the previous version of STAR-CCM+ are still supported.

A summary of the changes for each tool is provided below.

Surface Repair Launch Macro Changes

The following new command has been introduced:

```
partSurfaceMeshWidget_0.startSurfaceRepairControllers();
```

The above command replaces the following commands:

```
partSurfaceMeshWidget_0.startSurfaceMeshDiagnostics();
partSurfaceMeshWidget_0.startSurfaceMeshRepair();
partSurfaceMeshWidget_0.startMergeImprintController();
partSurfaceMeshWidget_0.startIntersectController();
partSurfaceMeshWidget_0.startLeakFinderController();
partSurfaceMeshWidget_0.startSurfaceMeshQueryController();
```

The following commands are no longer included in the macro for launching surface repair:

Removal of Feature Diagnostics Default Display State (Reason: Now Done on Server)

```
SurfaceMeshWidgetDiagnosticsController
surfaceMeshWidgetDiagnosticsController_0 =
partSurfaceMeshWidget_0.getControllers().getController(SurfaceMeshWidge
tDiagnosticsController.class);
surfaceMeshWidgetDiagnosticsController_0.setSoftFeatureErrorsActive(tru
e);
surfaceMeshWidgetDiagnosticsController_0.setHardFeatureErrorsActive(tru
e);
```

See also Macro Changes Common to Both Tools.

Wrapper Leak Detection Launch Macro Changes

The following new command has been introduced:

```
partSurfaceMeshWidget_0.startWrapperLeakDetectionControllers();
```

The above command replaces the following commands:

```
partSurfaceMeshWidget_0.startSurfaceMeshRepair();
partSurfaceMeshWidget_0.startMergeImprintController();
partSurfaceMeshWidget_0.startLeakFinderController();
partSurfaceMeshWidget_0.startSurfaceMeshDiagnostics();
```

The following commands are no longer included in the macro for launching wrapper leak detection:

Removal of Mesh Override Mode (Reason: Duplicate)

```
scene_0.setMeshOverrideMode(2);
```

Removal of Disabling of Default Diagnostics (Reason: Now Done Silently via the `setCheckAll()` Command)

```
PiercedFacesThreshold piercedFacesThreshold_0 =
      ((PiercedFacesThreshold)
surfaceMeshWidgetDiagnosticsController_0.getOptions().getThresholdDiagn
osticsManager().getObject("Pierced faces (default) 1"));
piercedFacesThreshold_0.setIsEnabled(false);
FaceQualityThreshold faceQualityThreshold_0 =
      ((FaceQualityThreshold)
surfaceMeshWidgetDiagnosticsController_0.getOptions().getThresholdDiagn
osticsManager().getObject("Face quality (default) 1"));
faceQualityThreshold_0.setIsEnabled(false);
FaceProximityThreshold faceProximityThreshold_0 =
      ((FaceProximityThreshold)
surfaceMeshWidgetDiagnosticsController_0.getOptions().getThresholdDiagn
osticsManager().getObject("Face proximity (default) 1"));
faceProximityThreshold_0.setIsEnabled(false);
FreeEdgesThreshold freeEdgesThreshold_0 =
      ((FreeEdgesThreshold)
surfaceMeshWidgetDiagnosticsController_0.getOptions().getThresholdDiagn
osticsManager().getObject("Free edges (default) 1"));
freeEdgesThreshold_0.setIsEnabled(false);
NonManifoldEdgesThreshold nonManifoldEdgesThreshold_0 =
      ((NonManifoldEdgesThreshold)
surfaceMeshWidgetDiagnosticsController_0.getOptions().getThresholdDiagn
osticsManager().getObject("Non-manifold edges (default) 1"));
nonManifoldEdgesThreshold_0.setIsEnabled(false);
NonManifoldVerticesThreshold nonManifoldVerticesThreshold_0 =
      ((NonManifoldVerticesThreshold)
surfaceMeshWidgetDiagnosticsController_0.getOptions().getThresholdDiagn
osticsManager().getObject("Non-manifold vertices (default) 1"));
nonManifoldVerticesThreshold_0.setIsEnabled(false);
```

Note: The `setCheckAll()` command now disables *all* diagnostics, not just the default ones.

Macro Changes Common to Both Tools

Removal of Units Objects (Reason: Not Used)

```
Units units_0 = simulation_0.getUnitsManager().getPreferredUnits(new
IntVector(new int[]
{0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}));
Units units_1 = simulation_0.getUnitsManager().getPreferredUnits(new
```

```
IntVector(new int[]
{0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}));
```

Removal of Displayer Initialization (Reason: Has No Effect)

```
SurfaceMeshWidgetDisplayer surfaceMeshWidgetDisplayer_0 =
((SurfaceMeshWidgetDisplayer)scene_0.getDisplayerManager().getDisplaye
r("Widget displayer 1"));
surfaceMeshWidgetDisplayer_0.initialize();
```

Macro API Changes 11.04

In STAR-CCM+ 11.04, the macro API changed for electrochemistry, simulation retrieval, server connections, co-simulation, VOF, DEM, the Segregated Time Step, material models, combustion, meshing, and morphing.

Electrochemistry: Renaming of Electrochemical Species Solver

The java class `ElectrochemicalSpeciesSolver` has been renamed to `SegregatedElectrochemicalSpeciesSolver`.

Within macro code, all occurrences of `ElectrochemicalSpeciesSolver` must be renamed accordingly. For example:

Previous Release	STAR-CCM+ v11.04
`import star.electrochemicalspecies.ElectrochemicalSpeciesSolver;` `ElectrochemicalSpeciesSolver echemSolver = _sim.getSolverManager().getSolver(ElectrochemicalSpeciesSolver.class);`	`import star.electrochemicalspecies.SegregatedElectrochemicalSpeciesSolver;` `SegregatedElectrochemicalSpeciesSolver echemSolver = _sim.getSolverManager().getSolver(SegregatedElectrochemicalSpeciesSolver.class);`

Electrochemistry: Change to Default Electroneutrality Option

The new default setting for the *Electroneutrality* option of the **Electrochemical Species** model is **Scaling**. Therefore macros which have previously set the `ElectroneutralityOption` to `ENFORCED` must set it to `SCALED`. For example:

Previous Release	STAR-CCM+ v11.04

`Simulation simulation_0 = getActiveSimulation();`
 	
`PhysicsContinuum physicsContinuum_0 =`
	`Simulation simulation_0 = getActiveSimulation();`
	
	`PhysicsContinuum physicsContinuum_0 =`

Previous Release	STAR-CCM+ v11.04
((PhysicsContinuum) simulation_0.getContinuumManager() .getContinuum("Physics 1")); ElectrochemicalSpeciesModel electrochemicalSpeciesModel_0 = physicsContinuum_0.getModelManager() .getModel(ElectrochemicalSpeciesModel.class); electrochemicalSpeciesModel_0 .getElectroneutralityOption() .setSelected(ElectroneutralityOption.Type.ENFORCED);	((PhysicsContinuum) simulation_0.getContinuumManager() .getContinuum("Physics 1")); ElectrochemicalSpeciesModel electrochemicalSpeciesModel_0 = physicsContinuum_0.getModelManager() .getModel(ElectrochemicalSpeciesModel.class); electrochemicalSpeciesModel_0 .getElectroneutralityOption() .setSelected(ElectroneutralityOption.Type.SCALED);

Simulations: New Method for Retrieval of Multiple Simulations

A new convenient method is available for retrieving a collection of simulations.

Previous Release	STAR-CCM+ v11.04
Set<Simulation> sl = ApplicationManager.getSingleton() .getApplicationAs(SimulationApplication.class, SimulationApplication.APPLICATION_TYPE).getRootObjects()	Collection<Simulation> sl = SimulationApplication.getSimulations()

Server Connections: Upgrades to Methods

For reasons of compatibility, the methods of the previous version are still available, but marked as deprecated. It is recommended that you upgrade your macros with the methods of the current version for long-term reliability.

Previous Release	STAR-CCM+ v11.04
static String ServerConnection.getRshCommand() static void ServerConnection.setRshCommand(String rshCommand)	String StarSettings.getSingleton() .getGenericRemoteShellSettings() .getRemoteShell().getCommandAndOptions() void StarSettings.getSingleton()

Previous Release	STAR-CCM+ v11.04
	`.getGenericRemoteShellSettings().setRemoteShell(String)`

`static String ServerConnection.getServerCmd()`
 	
`static String ServerConnection.getServerCmd(String sessionFile)`
 	
`static void ServerConnection.setServerCmd(String cmd)`
	`static String ServerProcess.getServerCommand()`
	
	`static String ServerProcess.getServerCommand(String sessionFile)`
	
	`static void ServerProcess.setServerCommand(String cmd)`

`static boolean ServerConnection.submitBatchJob()` `static boolean ServerConnection.submitBatchJob(String command)`	`static boolean ServerProcess.isWindowsHpcServerSubmission()` `static boolean ServerProcess.isWindowsHpcServerSubmission(String command)`
`static String ServerConnection.getDefaultRemoteShellOptions(String rshCmd)`	`static String RemoteShellFactory.getRemoteShell(String rshCmd).getOptions()`
`static String ServerConnection.getRemoteCommand(String host, String user, String command)`	`static ServerProcess.getRemoteCommand(String host, String user, String command)`

`static String ServerConnection.getDefaultHost()`
 	
`static void ServerConnection.setDefaultHost(String host)`
	`static String ServerProcess.getDefaultHost()`
	
	`static void ServerProcess.setDefaultHost(String host)`

`static String ServerConnection.getDefaultUser()`
 	
`static void`
	`static String ServerProcess.getDefaultUser()`
	
	`static void`

Previous Release	STAR-CCM+ v11.04
`ServerConnection.setDefaultUser(String user)`	`ServerProcess.setDefaultUser(String user)`

`static String ServerConnection.getSimFile()`
 	
`static void ServerConnection.setSimFile(String file)`
	`static String ServerProcess.getSimFile()`
	
	`static void ServerProcess.setSimFile(String file)`

Co-Simulation: Introduction of Explicit and Implicit Co-Simulation Models

To facilitate code reuse and enforce coupling strategies, a set of explicit and implicit co-simulation models has been implemented, resulting in changes to the macro code.

The following code examples are for Abaqus co-simulation.

Previous Release	STAR-CCM+ v11.04

`physicsContinuum_0.enable(CoSimulationModel.class);`
`physicsContinuum_0.enable(AbaqusCoSimulationModel.class);`
	`physicsContinuum_0.enable(CoSimulationOptionModel.class);`
	`physicsContinuum_0.enable(AbaqusCoSimulationModel.class);`
	`physicsContinuum_0.enable(AbaqusCoSimulationImplicitModel.class);`

VOF: Changes Due to Compatibility with Combustion Models

The Volume of Fluid (VOF) model is now compatible with combustion models, resulting in the use of `NonReactingModel.class`. For example:

Previous Release	STAR-CCM+ v11.04

`EulerianPhase eulerianPhase = getEulerianMultiPhaseModel().createPhase();`
`eulerianPhase.enable(SinglePhaseMultiComponentGasModel.class);`
`eulerianPhase.enable(SegregatedVofPhaseSpeciesModel.class);`
	`EulerianPhase eulerianPhase = getEulerianMultiPhaseModel().createPhase();`
	`eulerianPhase.enable(SinglePhaseMultiComponentGasModel.class);`
	`eulerianPhase.enable(NonReactingModel.class);`
	`eulerianPhase.enable(SegregatedVofPhaseSpeciesModel.class);`

DEM: Particle Size Field Function Renamed to Parcel Size

The field function `ParticleSize` has been renamed to `ParcelSize`. Previously the `ParticleSize` field function was available only when the Discrete Element Method (DEM) continuum model was activated. Now `ParcelSize` is available for both Lagrangian and DEM phases.

Previous Release	STAR-CCM+ v11.04
`PrimitiveFieldFunction primitiveFieldFunction_1 = ((PrimitiveFieldFunction) simulation_0 .getFieldFunctionManager().getFunction("ParticleSize"));`	`PrimitiveFieldFunction primitiveFieldFunction_1 = ((PrimitiveFieldFunction) simulation_0 .getFieldFunctionManager().getFunction("ParcelSize"));`

Segregated Time Step Model: Changes Due to Restructuring

The objects of the Segregated Time Step model have been restructured, resulting in changes to the macro code.

Previous Release	STAR-CCM+ v11.04

`import star.casting.*;`
`import star.common.*;`
 	
`public void execute() {`
`  Simulation simulation_0 =`
`  getActiveSimulation();`
`  PhysicsContinuum physicsContinuum_0 =`
`  ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Physics 1"));`
`  SegregatedFluidTimeStepModel segregatedFluidTimeStepModel_0 =`
 	
`physicsContinuum_0.getModelManager().getModel(SegregatedFluidTimeStepModel.class);`
 	
`segregatedFluidTimeStepModel_0.setVerbose(true);`
`  SegregatedFluidTimeStepControl segregatedFluidTimeStepControl_0 =`
 	
`(SegregatedFluidTimeStepControl) ((SegregatedFluidTimeStepControlM`
	`import star.base.neo.*;`
	`import star.casting.*;`
	`import star.common.*;`
	
	`public void execute() {`
	`  Simulation simulation_0 =`
	`  getActiveSimulation();`
	`  PhysicsContinuum physicsContinuum_0 =`
	`  ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Physics 1"));`
	`  SegregatedFluidTimeStepModel segregatedFluidTimeStepModel_0 =`
	
	`physicsContinuum_0.getModelManager().getModel(SegregatedFluidTimeStepModel.class);`
	
	`segregatedFluidTimeStepModel_0.setVerboseFlag(true);`
	
	`segregatedFluidTimeStepModel_0.getStartTime().setValue(0.0);`
	
	`segregatedFluidTimeStepModel_0.ge`

Previous Release	STAR-CCM+ v11.04

`anager)`
`segregatedFluidTimeStepModel_0.getControlManager()).createControl();`
 	
`segregatedFluidTimeStepControl_0.getStartTime().setValue(0.0);`
 	
`segregatedFluidTimeStepControl_0.getEndTime().setValue(0.0);`
 	
`segregatedFluidTimeStepControl_0.getStartTimeStep().setValue(5.0E-5);`
 	
`segregatedFluidTimeStepControl_0.getMinTimeStep().setValue(5.0E-6);`
 	
`segregatedFluidTimeStepControl_0.setMaxTimeStepChangeFactor(1.5);`
`}`
	`tEndTime().setValue(0.0);`
	
	`segregatedFluidTimeStepModel_0.getStartTimeStep().setValue(5.0E-5);`
	
	`segregatedFluidTimeStepModel_0.getMinTimeStep().setValue(5.0E-6);`
	
	`segregatedFluidTimeStepModel_0.setMaxTimeStepChangeFactors(new DoubleVector(new double[]{1.5}));`
	`}`

The new Segregated Time Step methods are as follows:

- `SegregatedFlowTimeStepMethod`
- `SegregatedVofTimeStepMethod`
- `SegregatedEnergyTimeStepMethod`
- `EnthalpyMeltingTimeStepMethod`

These methods are available via macro through the `getMethod` function provided by the model, for example:

```
  SegregatedFlowTimeStepMethod segregatedFlowTimeStepMethod_0 =

segregatedFluidTimeStepModel_0.getMethod(SegregatedFlowTimeStepMethod.class));
```

The method's shared properties are accessible through steps such as the following:

```
  segregatedFlowTimeStepMethod_0.getMaxConditionLimit().setValue(0.6);
  segregatedFlowTimeStepMethod_0.getSmoothingSteps().setValue(3.0);
```

Material Models: Change to Activation of Multi-Part Solid Model

Among the multi-component solid options, the Multi-Layer Solid model is now available in addition to the Multi-Part Solid model. This new option has resulted in changes to the macro code for activating the Multi-Part Solid model.

Previous Release	STAR-CCM+ v11.04
`physicsContinuum_0.enable(MultiPartSolidModel.class);`	`physicsContinuum_0.enable(MultiComponentSolidMaterialModel.class);` `physicsContinuum_0.enable(MultiPartSolidModel.class);`

Reactions: Chemistry ADI Removed from Complex Chemistry

Chemistry ADI is not available in the current release of STAR-CCM+.

To update your macros, remove instances of `complexChemistryModel_0.getChemistrySolverOption().setSelected(ComplexChemistrySolverOption.CADI);`.

Reactions: Extension of Choice of Solvers with ISAT

ISAT is now available with the CVODE solver and the Complex Chemistry user interface has been updated, resulting in changes to the macro code for the following situations:

Activating ISAT

Previous Release	STAR-CCM+ v11.04

`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Physics 1"));`
 	
`ComplexChemistryCombustionModel complexChemistryModel_0 =`
 	
`((ComplexChemistryCombustionModel) physicsContinuum_0.getModelManager().getModel(ComplexChemistryCombustionModel.class));`
 	
`complexChemistryModel_0.getIsatOption().setSelected(IsatOrEqTsmOption.ISAT);`
`DarsCfdIsatCalculationMethod darsCfdIsatCalculationMethod_0 =`
	`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Physics 1"));`
	
	`ComplexChemistryCombustionModel complexChemistryModel_0 =`
	
	`((ComplexChemistryCombustionModel) physicsContinuum_0.getModelManager().getModel(ComplexChemistryCombustionModel.class));`
	`ChemistryAccelerationOptions chemistryAccelerationOptions_0 = complexChemistryModel_0.getChemistryAccelerationOptions();`
	
	`chemistryAccelerationOptions_0.se`

Previous Release	STAR-CCM+ v11.04

`complexChemistryModel_0.getDarsCfdIsatCalculationMethod();`
 	
`darsCfdIsatCalculationMethod_0.getIsatPressureSpecificationOption().setSelected(IsatPressureSpecificationOption.CONST);`
	`tIsat(true);`
	`DarsCfdIsatCalculationMethod darsCfdIsatCalculationMethod_0 = complexChemistryModel_0.getDarsCfdIsatCalculationMethod();`
	
	`darsCfdIsatCalculationMethod_0.getIsatPressureSpecificationOption().setSelected(IsatPressureSpecificationOption.CONST);`

Deactivating Clustering

Previous Release	STAR-CCM+ v11.04

`ComplexChemistryCombustionModel complexChemistryModel_0 =`
 	
`((ComplexChemistryCombustionModel) physicsContinuum_0.getModelManager().getModel(ComplexChemistryCombustionModel.class));`
 	
`complexChemistryModel_0.getComplexChemistryOdeSolverOption().setSelected(OdeSolverOption.DARSCFD);`
 	
`complexChemistryModel_0.setClustering(false);`
	`ComplexChemistryCombustionModel complexChemistryModel_0 =`
	
	`((ComplexChemistryCombustionModel) physicsContinuum_0.getModelManager().getModel(ComplexChemistryCombustionModel.class));`
	
	`complexChemistryModel_0.getComplexChemistryOdeSolverOption().setSelected(OdeSolverOption.DARSCFD);`
	`ChemistryAccelerationOptions chemistryAccelerationOptions_0 =`
	
	`complexChemistryModel_0.getChemistryAccelerationOptions();`
	
	`chemistryAccelerationOptions_0.setClustering(false);`

Activating the Equilibrium Time Scale Model

Previous Release	STAR-CCM+ v11.04

`ComplexChemistryCombustionModel complexChemistryCombustionModel_0 =`
 	
`physicsContinuum_0.getModelManage`
	`ComplexChemistryCombustionModel complexChemistryCombustionModel_0 =`
	
	`physicsContinuum_0.getModelManage`

Previous Release	STAR-CCM+ v11.04

`r().getModel(ComplexChemistryCombustionModel.class);`
 	
`complexChemistryCombustionModel_0.getDarsCfdLibrary().importCase("darscfdstarccm+ch4sk30", resolveDataPath(""));`
 	
`complexChemistryCombustionModel_0.getComplexChemistryOdeSolverOption().setSelected(OdeSolverOption.Type.DARSCFD);`
 	
`complexChemistryCombustionModel_0.getIsatOption().setSelected(IsatOrEqTsmOption.Type.EQTSM);`
	`r().getModel(ComplexChemistryCombustionModel.class);`
	
	`complexChemistryCombustionModel_0.getDarsCfdLibrary().importCase("darscfdstarccm+ch4sk30", resolveDataPath(""));`
	
	`complexChemistryCombustionModel_0.getComplexChemistryOdeSolverOption().setSelected(OdeSolverOption.Type.DARSCFD);`
	`  ChemistryApproximationOptions chemistryApproximationOptions_0 =`
	
	`complexChemistryCombustionModel_0.getChemistryApproximationOptions();`
	
	`chemistryApproximationOptions_0.setEqTsm(true);`

Combustion: Premixed Variants of EBU and FGM Removed

The premixed variants of the Eddy Break-Up (EBU) and Flamelet Generated Manifold (FGM) models have been discontinued. When you select Premixed Combustion > Eddy Break-Up or Premixed Combustion > Flamelet Generated Manifold, STAR-CCM+ v11.04 uses the partially premixed variants.

The following are examples of macro code lines that have changed:

Converting Premixed EBU to EBU

Previous Release	STAR-CCM+ v11.04

`physicsContinuum_0.enable(PebuCombustionModel.class);`
`physicsContinuum_0.enable(PebuReactionModel.class);`
`physicsContinuum_0.enable(IdealGasModel.class);`
	`physicsContinuum_0.enable(EbuGasCombustionModel.class);`
	`physicsContinuum_0.enable(SegregatedSpeciesModel.class);`
	`physicsContinuum_0.enable(IdealGasModel.class);`

Converting Premixed FGM (formerly known as PPVM) to FGM

Previous Release	STAR-CCM+ v11.04

`physicsContinuum_0.enable(PpvmCombustionModel.class);`
`physicsContinuum_0.enable(PpvmReactionModel.class);`
`physicsContinuum_0.enable(PpvmIdealGasModel.class);`
	`physicsContinuum_0.enable(FgmCombustionModel.class);`
	
	`physicsContinuum_0.enable(FgmReactionModel.class);`
	`physicsContinuum_0.enable(FgmIdealGasModel.class);`
	`physicsContinuum_0.enable(FgmFiniteRateChemistryModel.class);`

PPDF Combustion: Change to Model Selection

The choice between Adiabatic and Non-Adiabatic PPDF models has been moved down from the top level of the selection of Non-Premixed Combustion models. First, the choice between PPDF Equilibrium and Flamelet Model is presented along with other combustion models, then either Adiabatic or Non-Adiabatic PPDF models can be chosen. The latter is selected automatically, since most practical use cases are non-adiabatic. However, you can deactivate the automatic selection and choose adiabatic if needed.

The following are examples of macro code lines that have changed:

PPDF Equilibrium, Non-Adiabatic Case

Previous Release	STAR-CCM+ v11.04
`physicsContinuum_0.enable(NonAdiabaticPpdfCombustionModel.class);` `physicsContinuum_0.enable(PpdfEquilibriumModel.class);`	`physicsContinuum_0.enable(PpdfEquilibriumModel.class);` `physicsContinuum_0.enable(NonAdiabaticPpdfCombustionModel.class);`

PPDF Equilibrium, Adiabatic Case

Previous Release	STAR-CCM+ v11.04
`physicsContinuum_0.enable(AdiabaticPpdfCombustionModel.class);` `physicsContinuum_0.enable(PpdfEquilibriumModel.class);`	`physicsContinuum_0.enable(PpdfEquilibriumModel.class);` `physicsContinuum_0.enable(AdiabaticPpdfCombustionModel.class);`

PPDF Flamelet, Non-Adiabatic Case

Previous Release	STAR-CCM+ v11.04
`physicsContinuum_0.enable(NonAdiabaticPpdfCombustionModel.class);` `physicsContinuum_0.enable(PpdfFlameletModel.class);`	`physicsContinuum_0.enable(PpdfNonAdiabaticFlameletModel.class);` `physicsContinuum_0.enable(NonAdiabaticPpdfCombustionModel.class);`

PPDF Flamelet, Adiabatic Case

Previous Release	STAR-CCM+ v11.04
`physicsContinuum_0.enable(AdiabaticPpdfCombustionModel.class);` `physicsContinuum_0.enable(PpdfFlameletModel.class);`	`physicsContinuum_0.enable(PpdfAdiabaticFlameletModel.class);` `physicsContinuum_0.enable(AdiabaticPpdfCombustionModel.class);`

Meshing: Changes to Creation of Part Surfaces and Part Curves Due to Split Actions

For split actions, which are typically executed through context menus in the user interface, the order in which part surfaces and part curves are created has changed. As a result, the names that are used to refer to these entities in macros have also changed. Expect to update your macro code for meshing with new names for part surfaces and part curves.

The following are examples of macro code lines that have changed:

Part Surface > Split Non-Contiguous

Previous Release	STAR-CCM+ v11.04

`MeshPart meshPart_0 = ((MeshPart) simulation_0.get(SimulationPartManager.class).getPart("Part"));`
`PartSurface partSurface_0 = ((PartSurface) meshPart_0.getPartSurfaceManager().getPartSurface("Surface"));`
`meshPart_0.getPartSurfaceManager().splitNonContiguousPartSurfaces(new NeoObjectVector(new Object[]{partSurface_0}));`
`partSurface_0.setPresentationName("Special Surface");`
	`MeshPart meshPart_0 = ((MeshPart) simulation_0.get(SimulationPartManager.class).getPart("Part"));`
	`PartSurface partSurface_0 = ((PartSurface) meshPart_0.getPartSurfaceManager().getPartSurface("Surface"));`
	`meshPart_0.getPartSurfaceManager().splitNonContiguousPartSurfaces(new NeoObjectVector(new Object[]{partSurface_0}));`
	`PartSurface partSurface_1 = ((PartSurface) meshPart_0.getPartSurfaceManager().getPartSurface("Surface 2"));`
	`partSurface_1.setPresentationName("Special Surface");`

Part Surface > Split by Part Curves

Previous Release	STAR-CCM+ v11.04
`MeshPart meshPart_0 = ((MeshPart) simulation_0.get(SimulationPartManager.class).getPart("Part"));`	`MeshPart meshPart_0 = ((MeshPart) simulation_0.get(SimulationPartManager.class).getPart("Part"));`

Previous Release	STAR-CCM+ v11.04

`PartSurface partSurface_0 = ((PartSurface) meshPart_0.getPartSurfaceManager().getPartSurface("Surface"));`
`PartCurve partCurve_0 =`
 	
`meshPart_0.getPartCurveManager().getPartCurve("Curve");`
`meshPart_0.getPartSurfaceManager().splitPartSurfacesByPartCurves(new NeoObjectVector(new Object[]{partSurface_0}), new NeoObjectVector(new Object[]{partCurve_0}));`
`PartSurface partSurface_1 = ((PartSurface) meshPart_0.getPartSurfaceManager().getPartSurface("Surface 6"));`
`partSurface_1.setPresentationName("Special Surface");`
	`PartSurface partSurface_0 = ((PartSurface) meshPart_0.getPartSurfaceManager().getPartSurface("Surface"));`
	`PartCurve partCurve_0 =`
	
	`meshPart_0.getPartCurveManager().getPartCurve("Curve");`
	`meshPart_0.getPartSurfaceManager().splitPartSurfacesByPartCurves(new NeoObjectVector(new Object[]{partSurface_0}), new NeoObjectVector(new Object[]{partCurve_0}));`
	`PartSurface partSurface_1 = ((PartSurface) meshPart_0.getPartSurfaceManager().getPartSurface("Surface 4"));`
	`partSurface_1.setPresentationName("Special Surface");`

Part Surface > Split by Angle

Previous Release	STAR-CCM+ v11.04

`MeshPart meshPart_0 = ((MeshPart) simulation_0.get(SimulationPartManager.class).getPart("Part"));`
`PartSurface partSurface_0 = ((PartSurface) meshPart_0.getPartSurfaceManager().getPartSurface("Surface"));`
`meshPart_0.getPartSurfaceManager().splitPartSurfacesByAngle(new NeoObjectVector(new Object[]{partSurface_0}), 89.0);`
`PartSurface partSurface_1 = ((PartSurface) meshPart_0.getPartSurfaceManager().getPartSurface("Surface 7"));`
`partSurface_1.setPresentationName("Special Surface");`
	`MeshPart meshPart_0 = ((MeshPart) simulation_0.get(SimulationPartManager.class).getPart("Part"));`
	`PartSurface partSurface_0 = ((PartSurface) meshPart_0.getPartSurfaceManager().getPartSurface("Surface"));`
	`meshPart_0.getPartSurfaceManager().splitPartSurfacesByAngle(new NeoObjectVector(new Object[]{partSurface_0}), 89.0);`
	`PartSurface partSurface_1 = ((PartSurface) meshPart_0.getPartSurfaceManager().getPartSurface("Surface 2"));`
	`partSurface_1.setPresentationName("Special Surface");`

Part Curve > Split Non-Contiguous

Previous Release	STAR-CCM+ v11.04

`MeshPart meshPart_0 = ((MeshPart) simulation_0.get(SimulationPartManager.class).getPart("Part"));`
`PartCurve partCurve_0 = meshPart_0.getPartCurveManager().getPartCurve("Curve");`
`meshPart_0.getPartCurveManager().splitNonContiguousPartCurves(new NeoObjectVector(new Object[] {partCurve_0}));`
`PartCurve partCurve_1 = meshPart_0.getPartCurveManager().getPartCurve("Curve 3");`
`partCurve_1.setPresentationName("Special Curve");`
	`MeshPart meshPart_0 = ((MeshPart) simulation_0.get(SimulationPartManager.class).getPart("Part"));`
	`PartCurve partCurve_0 = meshPart_0.getPartCurveManager().getPartCurve("Curve");`
	`meshPart_0.getPartCurveManager().splitNonContiguousPartCurves(new NeoObjectVector(new Object[] {partCurve_0}));`
	`PartCurve partCurve_1 = meshPart_0.getPartCurveManager().getPartCurve("Curve 2");`
	`partCurve_1.setPresentationName("Special Curve");`

Directed Meshing: Change to Ordering of Patch Vertices and Patch Curves

During directed mesh operations, the order in which patch vertices and patch curves are created in patch meshing has changed. As a result, the names that are used to refer to these entities in macros have also changed. Expect to update your macro code for patch meshing with new patch vertex and curve names. In more complex patch meshes, consider recording entirely new macros for patch meshing. For increased robustness during pipeline-based changes to part geometry, use Auto Source Mesh rather than Patch Mesh for source mesh type.

The following are examples of macro code lines from the previous version of STAR-CCM+ that are obsolete:

- Code that refers to patch vertices and patch curves by name:

```
PatchVertex patchVertex_3 = ((PatchVertex)

directedPatchSourceMesh_0.getPatchVertexManager().getObject("PatchVertex 99"));
PatchCurve patchCurve_36 = ((PatchCurve)

directedPatchSourceMesh_0.getPatchCurveManager().getObject("PatchCurve 177"));
```

- Code that operates on those entities:

```
directedPatchSourceMesh_0.splitPatchCurve(patchCurve_31, new DoubleVector(new double[]
  {-0.008895140013122065, 0.024398976690245322, -0.025}));
```

Directed Meshing: Change to Naming of Source Mesh Nodes

In previous versions of STAR-CCM+, the directed mesh operation named new source mesh nodes using either part or part collection names. Now that source mesh nodes can apply to more than one part collection, the name that is given to each source mesh node by default has also been changed. New source mesh nodes are now named after the source mesh type (Use Existing, Auto Source Mesh, or Patch Mesh), with sequence numbers added to maintain uniqueness. This naming technique is similar to the one that is used for mesh operations. Source mesh nodes can still be renamed. For an overview of changes to directed meshing in this release, see "Directed Meshing Improvements" in the list of new Volume Mesh features.

Update your macro code when it refers to source mesh nodes by name using the old naming convention:

Previous Release	STAR-CCM+ v11.04

`DirectedAutoSourceMesh directedAutoSourceMesh_0 = ((DirectedAutoSourceMesh)`
 	
`directedMeshOperation_0.getGuided SurfaceMeshBaseManager().  getObject("Connected Part"));`
	`DirectedAutoSourceMesh directedAutoSourceMesh_0 = ((DirectedAutoSourceMesh)`
	
	`directedMeshOperation_0.getGuided SurfaceMeshBaseManager().  getObject("Auto Mesh"));`

Morpher: Changes to Boundary Conditions

Morpher boundary conditions have been refactored, resulting in changes to the macro code.

MovingMeshOption

The class `MovingMeshOption` was renamed to `MorpherSpecification`.

Previous Release	STAR-CCM+ v11.04

`boundary_0.getConditions().get(Mo vingMeshOption.class).`
 	
`setSelected(MovingMeshOption.DISP LACEMENT);`
	`boundary_0.getConditions().get(Mo rpherSpecification.class).`
	
	`setSelected(MorpherSpecification. Type.DISPLACEMENT);`

Changes to related elements are as follows:

Previous Release	STAR-CCM+ v11.04
`MovingMeshOption`	`MorpherSpecification`
`FIXED`	No changes
`FLOATING`	
`DISPLACEMENT`	Adds `DisplacementSpecification` to the Physics Condition Manager
`GRID_VELOCITY`	Moved to `DisplacementSpecification`

Previous Release	STAR-CCM+ v11.04
`INTERNAL`	
`SOLIDSTRESS`	
`IN_PLANE`	Moved to `ConstraintSpecification`
`FIXEDPLANE`	
`SLIDE_ON_SURFACE`	
`RESTRAINED`	Renamed to `CONSTRAINT`. Adds `ConstraintSpecification` to the Physics Condition Manager

DofMorpherOption

Analogously the class `DofMorpherOption` is moved to `DofMorpherSpecification`.

Previous Release	STAR-CCM+ v11.04

`import star.sixdof.DofMorpherOption;`
 	
`boundary_1.getConditions().get(DofMorpherOption.class).`
 	
`setSelected(DofMorpherOption.Type.SIXDOFBODY);`
	`import star.sixdof.DofMorpherSpecification;`
	
	`boundary_1.getConditions().get(DofMorpherSpecification.class).`
	
	`setSelected(DofMorpherSpecification.Type.SIXDOFBODY);`

MorphPlanef

The class `MorphPlanef` is called now `MorpherPlaneOptions`.

Previous Release	STAR-CCM+ v11.04

`import star.morpher.MorphPlanef;`
`  MorphPlanef morphPlanef_0 =`
 	
`boundary_0.getValues().get(MorphPlanef.class);`
	`import star.morpher.MorpherPlaneOptions;`
	`  MorpherPlaneOptions morphPlaneOpt_0 =`
	
	`boundary_0.getValues().get(MorpherPlaneOptions.class);`
	
	`morphPlaneOpt_0.getDampFactor().setValue(0.01);`

DisplacementOption

The class `DisplacementOption` is renamed to `DisplacementSpecification`.

Previous Release	STAR-CCM+ v11.04

`boundary_0.getConditions().get(DisplacementOption.class).`
 	
`setSelected(DisplacementOption.Type.INCREMENTAL);`
	`boundary_0.getConditions().get(DisplacementSpecification.class).`
	
	`setSelected(DisplacementSpecification.Type.INCREMENTAL);`

Changes to related elements are as follows:

Previous Release	STAR-CCM+ v11.04
`DisplacementOption`	`DisplacementSpecification`
`TOTAL`	No changes
`INCREMENTAL`	
New elements	`GRID_VELOCITY`
	`SOLID_STRESS`
	`INTERNAL`
	`COORDINATE_OFFSET`

ConstraintSpecification

The new class `ConstraintSpecification` groups all the constraining methods that were previously located in `MovingMeshOption`.

Previous Release	STAR-CCM+ v11.04

`boundary_2.getConditions().get(MovingMeshOption.class).`
 	
`setSelected(MovingMeshOption.Type.FIXEDPLANE);`
	`boundary_2.getConditions().get(MorpherSpecification.class).`
	
	`setSelected(MorpherSpecification.Type.CONSTRAINT);`
	`boundary_2.getConditions().get(ConstraintSpecification.class).`
	
	`setSelected(ConstraintSpecification.Type.FIXEDPLANE);`

Changes to related elements are as follows:

Previous Release	STAR-CCM+ v11.04
`MovingMeshOption`	`ConstraintSpecification`
`IN_PLANE`	No changes
`FIXEDPLANE`	

Previous Release	STAR-CCM+ v11.04
`SLIDE_ON_SURFACE` with `GuideSurfaceOption.Type.ExternalFile`	`SLIDE_ON_STL_SURFACE`
`SLIDE_ON_SURFACE` with `GuideSurfaceOption.Type.InitialSurface`	`SLIDE_ON_INITIAL_BOUNDARY`
`SLIDE_ON_SURFACE` with `GuideSurfaceOption.Type.StayPlanar`	`IN_PLANE` with `PlaneConstraintSpecification.setTriangulatedPlane(true)`

The following table contains examples of these changed elements:

Previous Release	STAR-CCM+ v11.04

`boundary_0.getConditions().get(MovingMeshOption.class).`
 	
`setSelected(MovingMeshOption.Type.SLIDE_ON_SURFACE);`
`boundary_0.getConditions().get(GuideSurfaceSettings).getSurfaceOption().`
 	
`setSelected(GuideSurfaceOption.Type.ExternalFile);`
	`boundary_0.getConditions().get(MorpherSpecification.class).`
	
	`setSelected(MorpherSpecification.Type.CONSTRAINT);`
	`boundary_0.getConditions().get(MovingMeshOption.class).`
	
	`setSelected(MovingMeshOption.Type.SLIDE_ON_STL_SURFACE);`

`boundary_0.getConditions().get(MovingMeshOption.class).`
 	
`setSelected(MovingMeshOption.Type.SLIDE_ON_SURFACE);`
`boundary_0.getConditions().get(GuideSurfaceSettings).getSurfaceOption().`
 	
`setSelected(GuideSurfaceOption.Type.InitialSurface);`
	`boundary_0.getConditions().get(MorpherSpecification.class).`
	
	`setSelected(MorpherSpecification.Type.CONSTRAINT);`
	`boundary_0.getConditions().get(MovingMeshOption.class).`
	
	`setSelected(MovingMeshOption.Type.SLIDE_ON_INITIAL_SURFACE);`

`boundary_0.getConditions().get(MovingMeshOption.class).`
 	
`setSelected(MovingMeshOption.Type.SLIDE_ON_SURFACE);`
	`boundary_0.getConditions().get(MorpherSpecification.class).`
	
	`setSelected(MorpherSpecification.Type.CONSTRAINT);`

Previous Release	STAR-CCM+ v11.04

`boundary_0.getConditions().get(GuideSurfaceSettings).getSurfaceOption().`
 	
`setSelected(GuideSurfaceOption.Type.StayPlanar);`
	`boundary_0.getConditions().get(MovingMeshOption.class).`
	
	`setSelected(MovingMeshOption.Type.IN_PLANE;`
	`boundary_0.getConditions().get(PlaneConstraintSpecification.class).`
	`  setTriangulatedPlane(true);`

Macro API Changes 11.02

In STAR-CCM+ 11.02, the macro API changed for electrochemistry, combustion, erosion, GT-SUITE co-simulation, data mapping, and virtual disk modeling.

Electrochemistry: Renaming of Solid Electric Potential Profile

The profile `SolidElectricPotentialProfile` has been renamed to `ElectrodeElectricPotentialProfile`, resulting in changes to the macro code.

Within import sections of the macro code, as well as program instruction sections, all occurrences of `SolidElectricPotentialProfile` must be renamed accordingly. For example:

Previous Release	STAR-CCM+ v11.02
`smo.setActiveMechanism("SurfaceMechanism");` `SolidElectricPotentialProfile spp = _metalBoundary.getValues().get(SolidElectricPotentialProfile.class);`	`smo.setActiveMechanism("SurfaceMechanism");` `ElectrodeElectricPotentialProfile spp = _metalBoundary.getValues().get(ElectrodeElectricPotentialProfile.class);`

Electrochemistry: Renaming of Derivative Profile

The profile `MolarConcentrationFluxDerivativeProfile` has been renamed to `ConcentrationFluxDerivativeProfile`, resulting in changes to the macro code.

Within import sections of the macro code, as well as program instruction sections, all occurrences of `MolarConcentrationFluxDerivativeProfile` must be renamed accordingly. For example:

Previous Release	STAR-CCM+ v11.02
`chlorideAnionFluxProfile.getMethod(FunctionScalarProfileMethod.class`	`chlorideAnionFluxProfile.getMethod(FunctionScalarProfileMethod.class)`

Previous Release	STAR-CCM+ v11.02
```).setFieldFunction(BoundarySpecifiedFluxClMinusFF); MolarConcentrationFluxDerivativeProfile mcfdP = fluxBoundary .getValues() .get(MolarConcentrationFluxDerivativeProfile.class);``` 	```.setFieldFunction(BoundarySpecifiedFluxClMinusFF); ConcentrationFluxDerivativeProfile mcfdP = fluxBoundary .getValues() .get(ConcentrationFluxDerivativeProfile.class);```

Combustion: Transformation to Complex Chemistry

Combustion that involves DARS-CFD and complex chemistry has undergone extensive changes. Hence you are advised to re-record your macros, or at least the portions of those macros that affect the defining of models and properties.

As a rule, for macros defining complex chemistry:

- Occurrences of `import star.combustion.*;` should be changed to:

```
import star.combustion.*;
import star.combustion.complexchemistry.*;
```

- Occurrences of `import star.combustion.ui.*;` should be changed to:

```
import star.combustion.ui.*;
import star.combustion.complexchemistry.ui.*;
```

A general example of the changes to the macro code follows:

Previous Release	STAR-CCM+ v11.02

``physicsContinuum_0.enable(HomogeneousCombustionModel.class);
physicsContinuum_0.enable(OperatorSplittingGasModel.class);
OperatorSplittingGasModel operatorSplittingGasModel_0 =
 	
physicsContinuum_0.getModelManager().getModel(OperatorSplittingGasModel.class);
operatorSplittingGasModel_0.setThreshold(50);
physicsContinuum_0.enable(DarsCfdGasReactionModel.class);
physicsContinuum_0.enable(DarsCfdIdealGasModel.class);
DarsCfdIdealGasModel darsCfdIdealGasModel_0 =```
 	
	``physicsContinuum_0.enable(ComplexChemistryCombustionModel.class);
	physicsContinuum_0.enable(IdealGasModel.class);
	ComplexChemistryCombustionModel complexChemistryCombustionModel_0
	 =
	
	physicsContinuum_0.getModelManager().getModel(ComplexChemistryCombustionModel.class);
	complexChemistryCombustionModel_0.setThreshold(50);```

Previous Release	STAR-CCM+ v11.02
`physicsContinuum_0.getModelManager().getModel(DarsCfdIdealGasModel.class);` `darsCfdIdealGasModel_0.getIdealGasEquationOption().setSelected(IdealGasEquationOption.BUILTIN);`	

Examples of macro steps follow for specific instances of combustion modeling:

Activating the Chemistry ADI gas phase model

Previous Release	STAR-CCM+ v11.02

`physicsContinuum_0.enable(ReactingModel.class);`
`physicsContinuum_0.enable(NonPremixedCombustionFlameTypeModel.class);`
`physicsContinuum_0.enable(HomogeneousCombustionModel.class);`
`physicsContinuum_0.enable(ChemistryADIGasModel.class);`
`physicsContinuum_0.enable(DarsCfdGasReactionModel.class);`
	`physicsContinuum_0.enable(ReactingModel.class);`
	`physicsContinuum_0.enable(NonPremixedCombustionFlameTypeModel.class);`
	`physicsContinuum_0.enable(ComplexChemistryCombustionModel.class);`
	`ComplexChemistryCombustionModel complexChemistryCombustionModel_0 =`
	`physicsContinuum_0.getModelManager().getModel(ComplexChemistryCombustionModel.class);`
	`complexChemistryCombustionModel_0.getChemistrySolverOption().setSelected(ComplexChemistrySolverOption.Type.CADI);`

Activating the Operator Splitting gas phase model

Previous Release	STAR-CCM+ v11.02

`physicsContinuum_0.enable(ReactingModel.class);`
`physicsContinuum_0.enable(NonPremixedCombustionFlameTypeModel.class);`
`physicsContinuum_0.enable(HomogeneousCombustionModel.class);`
`physicsContinuum_0.enable(OperatorSplittingGasModel.class);`
	`physicsContinuum_0.enable(ReactingModel.class);`
	`physicsContinuum_0.enable(NonPremixedCombustionFlameTypeModel.class);`
	`physicsContinuum_0.enable(ComplexChemistryCombustionModel.class);`
	`ComplexChemistryCombustionModel complexChemistryCombustionModel_0 =`

Previous Release	STAR-CCM+ v11.02
`physicsContinuum_0.enable(DarsCfdGasReactionModel.class);`	`physicsContinuum_0.getModelManager().getModel(ComplexChemistryCombustionModel.class);` `complexChemistryCombustionModel_0.getComplexChemistryOdeSolverOption().setSelected(OdeSolverOption.Type.DARSCFD);`

Activating the Ideal Gas model for complex chemistry

Previous Release	STAR-CCM+ v11.02

`physicsContinuum_0.enable(SegregatedSpeciesModel.class);`
`physicsContinuum_0.enable(DarsCfdIdealGasModel.class);`
`physicsContinuum_0.enable(SegregatedFluidEnthalpyModel.class);`
`physicsContinuum_0.enable(LaminarModel.class);`
	`physicsContinuum_0.enable(SegregatedSpeciesModel.class);`
	`physicsContinuum_0.enable(SegregatedFluidEnthalpyModel.class);`
	`physicsContinuum_0.enable(IdealGasModel.class);`
	`physicsContinuum_0.enable(LaminarModel.class);`

Activating the Chemistry ADI surface/gas phase model

Previous Release	STAR-CCM+ v11.02

`physicsContinuum_0.enable(NonPremixedCombustionFlameTypeModel.class);`
`physicsContinuum_0.enable(HomogeneousCombustionSurfaceChemistryModel.class);`
`physicsContinuum_0.enable(SurfaceChemistryModel.class);`
`physicsContinuum_0.enable(ChemistryADISurfaceGasModel.class);`
`physicsContinuum_0.enable(DarsCfdSurfaceGasReactionModel.class);`
`physicsContinuum_0.enable(SegregatedFlowModel.class);`
	`physicsContinuum_0.enable(NonPremixedCombustionFlameTypeModel.class);`
	`physicsContinuum_0.enable(ComplexChemistryCombustionModel.class);`
	`physicsContinuum_0.enable(SegregatedFlowModel.class);`
	`physicsContinuum_0.enable(SurfaceChemistryModel.class);`
	`ComplexChemistryCombustionModel complexChemistryCombustionModel_0`
	
	`=physicsContinuum_0.getModelManager().getModel(ComplexChemistryCombustionModel.class);`
	`complexChemistryCombustionModel_0.getChemistrySolverOption().setSelected(ComplexChemistrySolverOption.Type.CADI);`

Activating the Operator Splitting surface/gas phase model

Previous Release	STAR-CCM+ v11.02

`physicsContinuum_0.enable(NonPremixedCombustionFlameTypeModel.class);`
`physicsContinuum_0.enable(HomogeneousCombustionSurfaceChemistryModel.class);`
`physicsContinuum_0.enable(SurfaceChemistryModel.class);`
`physicsContinuum_0.enable(OperatorSplittingSurfaceGasModel.class);`
`physicsContinuum_0.enable(DarsCfdSurfaceGasReactionModel.class);`
`physicsContinuum_0.enable(SegregatedFlowModel.class);`
	`physicsContinuum_0.enable(NonPremixedCombustionFlameTypeModel.class);`
	`physicsContinuum_0.enable(ComplexChemistryCombustionModel.class);`
	`physicsContinuum_0.enable(SegregatedFlowModel.class);`
	`physicsContinuum_0.enable(SurfaceChemistryModel.class);`
	`ComplexChemistryCombustionModel complexChemistryCombustionModel_0`
	
	`=physicsContinuum_0.getModelManager().getModel(ComplexChemistryCombustionModel.class);`
	`complexChemistryCombustionModel_0.getComplexChemistryOdeSolverOption().setSelected(OdeSolverOption.Type.DARSCFD);`

Setting the running trigger of complex chemistry models

Previous Release	STAR-CCM+ v11.02

`OperatorSplittingGasModel operatorSplittingGasModel_0 = physicsContinuum_0.getModelManager().getModel(OperatorSplittingGasModel.class);`
`operatorSplittingGasModel_0.getCombustionObjectEnabledOption().setSelected(CombustionObjectEnabledOption.Type.AFTER_ITERATION);`
`operatorSplittingGasModel_0.setThreshold(20);`
	`ComplexChemistryCombustionModel complexChemistryCombustionModel_0`
	
	`=physicsContinuum_0.getModelManager().getModel(ComplexChemistryCombustionModel.class);`
	`complexChemistryCombustionModel_0.getCombustionObjectEnabledOption().setSelected(CombustionObjectEnabledOption.Type.AFTER_ITERATION);`
	`complexChemistryCombustionModel_0.setThreshold(20);`

`OperatorSplittingSurfaceGasModel operatorSplittingSurfaceGasModel_0 = physicsContinuum_0.getModelManager().getModel(OperatorSplittingSurfaceGasModel.class);` `operatorSplittingSurfaceGasModel_`	

Previous Release	STAR-CCM+ v11.02
`0.getCombustionObjectEnabledOption().setSelected(CombustionObjectEnabledOption.Type.AFTER_ITERATION);` `operatorSplittingSurfaceGasModel_0.setThreshold(20);`	

`ChemistryADIGasModel chemistryADIGasModel_0 = physicsContinuum_0.getModelManager().getModel(ChemistryADIGasModel.class);`
`chemistryADIGasModel_0.getCombustionObjectEnabledOption().setSelected(CombustionObjectEnabledOption.Type.AFTER_ITERATION);`
`chemistryADIGasModel_0.setThreshold(20);`	

`ChemistryADISurfaceGasModel chemistryADISurfaceGasModel_0 = physicsContinuum_0.getModelManager().getModel(ChemistryADISurfaceGasModel.class);`
`chemistryADISurfaceGasModel_0.getCombustionObjectEnabledOption().setSelected(CombustionObjectEnabledOption.Type.AFTER_ITERATION);`
`chemistryADISurfaceGasModel_0.setThreshold(20);`	

Dars-CFD properties

Previous Release	STAR-CCM+ v11.02

`DarsCfdGasReactionModel darsCfdGasReactionModel_0 =physicsContinuum_0.getModelManager().getModel(DarsCfdGasReactionModel.class);`
`darsCfdGasReactionModel_0.setRunTimeCompile(false);`
`darsCfdGasReactionModel_0.setModuloProblemDescription(false);`
	`ComplexChemistryCombustionModel complexChemistryCombustionModel_0 =physicsContinuum_0.getModelManager().getModel(ComplexChemistryCombustionModel.class);`
	`DarsCfdLibrary darsCfdLibrary_0 = complexChemistryCombustionModel_0`

Previous Release	STAR-CCM+ v11.02
`DarsCfdProperties darsCfdProperties_0 =darsCfdGasReactionModel_0.getDarsCfdProperties(); darsCfdProperties_0.setUseReferenceValsTMinMax(false);`	`.getDarsCfdLibrary(); DarsCfdBinaryRunOptions darsCfdBinaryRunOptions_0 =darsCfdLibrary_0.getDarsCfdBinaryRunOptions(); darsCfdBinaryRunOptions_0.setRunTimeCompile(false); darsCfdLibrary_0.setModuloProblemDescription(false); darsCfdLibrary_0.setUseReferenceValsTMinMax(false);`

Dars-CFD model properties

Previous Release	STAR-CCM+ v11.02
`DarsCfdGasReactionModel darsCfdGasReactionModel_0 =physicsContinuum_0.getModelManager().getModel(DarsCfdGasReactionModel.class); darsCfdGasReactionModel_0.getColdFlowTempLimitScalar().setValue(400.0);`	`ComplexChemistryCombustionModel complexChemistryCombustionModel_0 =physicsContinuum_0.getModelManager().getModel(ComplexChemistryCombustionModel.class); complexChemistryCombustionModel_0 .getColdFlowTempLimitScalar().setValue(400.0);`
`DarsCfdSurfaceGasReactionModel darsCfdSurfaceGasReactionModel_0 =physicsContinuum_0.getModelManager().getModel(DarsCfdSurfaceGasReactionModel.class); darsCfdSurfaceGasReactionModel_0. getColdFlowTempLimitScalar().setValue(400.0);`	

Dynamic Load Balancing for complex chemistry

Previous Release	STAR-CCM+ v11.02
`DarsCfdIsatApproximationMethod darsCfdIsatApproximationMethod_0 =((DarsCfdIsatApproximationMethod) darsCfdProperties_0.getDarsCfdApproximation().getMethod());`	`ComplexChemistryCombustionModel complexChemistryCombustionModel_0 =physicsContinuum_0.getModelManager().getModel(ComplexChemistryCombustionModel.class);`

Previous Release	STAR-CCM+ v11.02
`darsCfdIsatApproximationMethod_0.setParallelRandomization(false);`	`complexChemistryCombustionModel_0.setParallelRandomization(false);`
`DarsCfdEqTsmApproximationMethod darsCfdEqTsmApproximationMethod_0 =((DarsCfdEqTsmApproximationMethod) darsCfdProperties_0.getDarsCfdApproximation().getMethod()); darsCfdEqTsmApproximationMethod_0.setParallelRandomization(false);`	
`DarsCfdNoneApproximationMethod darsCfdNoneApproximationMethod_0 =((DarsCfdNoneApproximationMethod) darsCfdProperties_0.getDarsCfdApproximation().getMethod()); darsCfdINoneApproximationMethod_0.setParallelRandomization(false);`	

Deleting a complex chemistry case

Previous Release	STAR-CCM+ v11.02
`DarsCfdGasReactionModel darsCfdGasReactionModel_0 = physicsContinuum_0.getModelManager().getModel(DarsCfdGasReactionModel.class); darsCfdGasReactionModel_0.deleteCase(true);`	`ComplexChemistryCombustionModel complexChemistryCombustionModel_0 =physicsContinuum_0.getModelManager().getModel(ComplexChemistryCombustionModel.class); complexChemistryCombustionModel_0.deleteCase(true);`
`DarsCfdSurfaceGasReactionModel darsCfdSurfaceGasReactionModel_0 = physicsContinuum_0.getModelManager().getModel(DarsCfdSurfaceGasReactionModel.class); darsCfdSurfaceGasReactionModel_0.deleteCase(true);`	

Importing a complex chemistry definition (Dars-CFD format)

Previous Release	STAR-CCM+ v11.02
`DarsCfdGasReactionModel darsCfdGasReactionModel_0 =physicsContinuum_0.getModelManager().getModel(DarsCfdGasReactionModel.class); darsCfdGasReactionModel_0.importCase("darscfdstarccm+", resolvePath("macroChanges/DarsAsciiInput"));`	`ComplexChemistryCombustionModel complexChemistryCombustionModel_0 =physicsContinuum_0.getModelManager().getModel(ComplexChemistryCombustionModel.class); DarsCfdLibrary darsCfdLibrary_0 = complexChemistryCombustionModel_0.getDarsCfdLibrary(); darsCfdLibrary_0.importCase("darscfdstarccm+", resolvePath("macroChanges/DarsAsciiInput"));`

Importing a complex chemistry definition (Chemkin format), gas phase

Previous Release	STAR-CCM+ v11.02
`DarsCfdGasReactionModel darsCfdGasReactionModel_0 =physicsContinuum_0.getModelManager().getModel(DarsCfdGasReactionModel.class); darsCfdGasReactionModel_0.importCaseFromChemkin("chem.inp", "therm.dat", "tran.dat", "StarccmProject", true, "darscfdstarccm+");`	`ComplexChemistryCombustionModel complexChemistryCombustionModel_0 =physicsContinuum_0.getModelManager().getModel(ComplexChemistryCombustionModel.class); complexChemistryCombustionModel_0.importCaseFromChemkin("chem.inp", "therm.dat", "tran.dat", "", "");`

Importing a complex chemistry definition (Chemkin format), surface/gas phase

Previous Release	STAR-CCM+ v11.02
`DarsCfdSurfaceGasReactionModel darsCfdSurfaceGasReactionModel_0 =physicsContinuum_0.getModelManager().getModel(DarsCfdSurfaceGasReactionModel.class); darsCfdSurfaceGasReactionModel_0.importCaseFromChemkin("chem.inp,surfChem.inp", "surfTherm.dat,therm.dat", "tran.dat", "StarccmProject", true, "darscfdstarccm+");`	`ComplexChemistryCombustionModel complexChemistryCombustionModel_0 =physicsContinuum_0.getModelManager().getModel(ComplexChemistryCombustionModel.class); complexChemistryCombustionModel_0.importCaseFromChemkin("chem.inp", "therm.dat", "tran.dat", "surfChem.inp", "surfTherm.dat");`

Activating ISAT and ISAT parameters

Previous Release	STAR-CCM+ v11.02

`darsCfdProperties_0.getDarsCfdApproximation().setMethod(DarsCfdIsatApproximationMethod.class);`
`DarsCfdIsatApproximationMethod darsCfdIsatApproximationMethod_0 =((DarsCfdIsatApproximationMethod) darsCfdProperties_0.getDarsCfdApproximation().getMethod());`
`darsCfdIsatApproximationMethod_0.getDarsCfdPressureSpecificationOption().setSelected(DarsCfdPressureSpecificationOption.Type.CONST);`
`DarsCfdIsatTable darsCfdIsatTable_0 =((DarsCfdIsatTable) darsCfdIsatApproximationMethod_0.getDarsCfdIsatTableManager().getDarsCfdIsatTable("1"));`
`darsCfdIsatTable_0.getUpperTimeStep().setValue(1.0E-4);`
	`complexChemistryCombustionModel_0.getIsatOption().setSelected(IsatOrEqTsmOption.Type.ISAT);`
	`DarsCfdIsatCalculationMethod darsCfdIsatCalculationMethod_0 = complexChemistryCombustionModel_0.getDarsCfdIsatCalculationMethod();`
	`darsCfdIsatCalculationMethod_0.getIsatPressureSpecificationOption().setSelected(IsatPressureSpecificationOption.Type.CONST);`
	`DarsCfdIsatTable darsCfdIsatTable_0 =((DarsCfdIsatTable) darsCfdIsatCalculationMethod_0.getDarsCfdIsatTableManager().getDarsCfdIsatTable("1"));`
	`darsCfdIsatTable_0.getUpperTimeStep().setValue(1.0E-4);`

Activating the Relax to Equilibrium method and setting properties

Previous Release	STAR-CCM+ v11.02

`darsCfdProperties_0.getDarsCfdApproximation().setMethod(DarsCfdEqTsmApproximationMethod.class);`
`DarsCfdEqTsmApproximationMethod darsCfdEqTsmApproximationMethod_0`
 	
`=((DarsCfdEqTsmApproximationMethod) darsCfdProperties_0.getDarsCfdApproximation().getMethod());`
`darsCfdEqTsmApproximationMethod_0.setIncludeChemTime(true);`
	`complexChemistryCombustionModel_0.getIsatOption().setSelected(IsatOrEqTsmOption.Type.EQTSM);`
	`DarsCfdEqTsmCalculationMethod darsCfdEqTsmCalculationMethod_0 =complexChemistryCombustionModel_0.getDarsCfdEqTsmCalculationMethod();`
	`darsCfdEqTsmCalculationMethod_0.setIncludeChemTime(true);`

Combustion: Changes to Names for Post-Processing

The changes to complex chemistry include the renaming of various items from `DarsCfd` to `ComplexChemistry`.

The following example shows changes to the macro code for creating reports:

Previous Release	STAR-CCM+ v11.02

`DarsCfdSumQuantityReport darsCfdSumQuantityReport_0 =`
 	
`simulation_0.getReportManager().c reateReport(DarsCfdSumQuantityRep ort.class);`
`darsCfdSumQuantityReport_0.getDar sCfdQuantityOption().setSelected( DarsCfdQuantityOption.Type.CPUTIM E);`
`darsCfdSumQuantityReport_0.getDar sCfdQuantityAccumulationOption().. setSelected(DarsCfdQuantityAccumu lationOption.Type.PERITER);`
	`ComplexChemistryMaxQuantityReport`
	
	`complexChemistryMaxQuantityReport _1 =`
	
	`simulation_0.getReportManager().c reateReport(ComplexChemistryMaxQu antityReport.class);`
	`complexChemistryMaxQuantityReport _1.getComplexChemistryReportQuant ityOption().setSelected(ComplexCh emistryReportQuantityOption.Type. CPUTIME);`
	`complexChemistryMaxQuantityReport _1.getComplexChemistryReportQuant ityAccumulationOption().setSelect ed(ComplexChemistryReportQuantity AccumulationOption.Type.PERITER);`

Combustion: Changes to Ideal Gas

The range of choices for ideal gas in combustion has been simplified. As a result, the Java macro classes `IdealGasWithCombustionModel.class` and `DarsCfdIdealGasModel.class` no longer exist. Update your macros as follows, depending on the combustion models that you are using:

- Replace `IdealGasWithCombustionModel.class` as follows:
  - For Eddy Break-Up, Thickened Flame Model, Coherent Flame Model, and Turbulent Flame Speed Closure, use `IdealGasModel.class`.
  - For Non-Premixed PPDF Equilibrium Model, use `PpdfIdealGasModel.class`.
  - For Partially-Premixed CFM, use `PcfmIdealGasModel.class`.
  - For Non-Adiabatic PPDF Flamelet Model, use `IdealGasWithNonAdiabaticFlameletModel.class`.
- For Complex Chemistry models, replace `DarsCfdIdealGasModel.class` with `IdealGasModel.class`.

An example from a macro with Eddy Break-Up follows:

Previous Release	STAR-CCM+ v11.02

`(NonPremixedCombustionFlameTypeMo del.class);`
 	
`physicsContinuum_0.enable(EbuGasC`
	`(NonPremixedCombustionFlameTypeMo del.class);`
	
	`physicsContinuum_0.enable(EbuGasC`

Previous Release	STAR-CCM+ v11.02

`ombustionModel.class);`
 	
`physicsContinuum_0.enable(SegregatedFlowModel.class);`
 	
`physicsContinuum_0.enable(SegregatedSpeciesModel.class);`
 	
`physicsContinuum_0.enable(IdealGasWithCombustionModel.class);`
 	
`physicsContinuum_0.enable(SegregatedFluidEnthalpyModel.class);`
	`ombustionModel.class);`
	
	`physicsContinuum_0.enable(SegregatedFlowModel.class);`
	
	`physicsContinuum_0.enable(SegregatedSpeciesModel.class);`
	
	`physicsContinuum_0.enable(SegregatedFluidEnthalpyModel.class);`
	
	`physicsContinuum_0.enable(IdealGasModel.class);`

Combustion: Changes to PPDF Tables

The way in which STAR-CCM+ handles PPDF tables has changed. Modify your existing macros accordingly.

Previous Release	STAR-CCM+ v11.02

`PpdfFlameletTable ppdfFlameletTable_0 = ((PpdfFlameletTable) ppdfNonAdiabaticFlameletModel_0.getPpdfFlameletTable());`
`NonPpdfVar nonPpdfVar_0 = ((NonPpdfVar) ((NonPpdfVarManager) ppdfFlameletTable_0.getNonPpdfVarManager()).getComponent("Scalar Dissipation Rate"));`
`nonPpdfVar_0.setDimensionSize(21);`
`PpdfTableAxis ppdfTableAxis_0 = ((PpdfTableAxis) ((PpdfTableAxisManager) ppdfFlameletTable_0.getPpdfTableAxisManager()).getComponent("Mixture Fraction"));`
	`PpdfFlameletTable ppdfFlameletTable_0 = ((PpdfFlameletTable) ppdfNonAdiabaticFlameletModel_0.getPpdfFlameletTable());`
	`PpdfTableAxis ppdfTableAxis_0 = ((PpdfTableAxis) ((PpdfTableAxisManager)ppdfFlameletTable_0.getPpdfTableAxisManager()).getComponent("Scalar Dissipation Rate"));`
	`ppdfTableAxis_0.setDimensionSize(21);`
	`PpdfTableAxis ppdfTableAxis_0 = ((PpdfTableAxis) ((PpdfTableAxisManager) ppdfFlameletTable_0.getPpdfTableAxisManager()).getComponent("Mixture Fraction"));`

Change for Erosion Model

The Erosion model is now available with Eulerian granular modeling. As a result, the **Erosion Ratio** scalar profile has been changed from `star.lagrangian.erosion.ErosionRatio` to `star.multiphase.erosion.ErosionRatio`. Modify your existing macros accordingly.

Previous Release	STAR-CCM+ v11.02

`import star.common.*;`
`import star.lagrangian.*;`
`import star.lagrangian.erosion.ErosionRatio;`
`DefaultBoundaryConditions defaultBoundaryConditions = (DefaultBoundaryConditions) getPhase().getDefaultBoundaryConditionsManager().getBoundaryConditions(WallBoundary.class);`
`ErosionRatio erosionRatio = defaultBoundaryConditions.getValues().get(ErosionRatio.class);`
`erosionRatio.setMethod(AhlertCorrelation.class);`
	`import star.common.*;`
	`import star.lagrangian.*;`
	`import star.multiphase.erosion.ErosionRatio;`
	`DefaultBoundaryConditions defaultBoundaryConditions = (DefaultBoundaryConditions) getPhase().getDefaultBoundaryConditionsManager().getBoundaryConditions(WallBoundary.class);`
	`ErosionRatio erosionRatio = defaultBoundaryConditions.getValues().get(ErosionRatio.class);`
	`erosionRatio.setMethod(AhlertCorrelation.class);`

GT-SUITE Co-simulation: Changes Due to Refactoring

GT-SUITE 1D Co-simulation has been refactored for greater ease of use and maintainability. The changes to the macros for this co-simulation are extensive, so you are advised to re-record them.

Data Mapping: 2D Regions Now Treated as Surfaces

Data mappers now treat 2D regions as surfaces rather than as volumes, resulting in changes to the macro code.

Previous Release	STAR-CCM+ v11.02

`VolumeDataMapper vdm = sim.get(DataMapperManager.class).createMapper( VolumeDataMapper.class, "Volume Data Mapper" );`
`vdm.setSourceStencil(2);`
`vdm.getSourceParts().setObjects(source2Dregion_0, source2Dregion_1);`
`vdm.setUpdateAvailableFields(true);`
`vdm.setScalarFieldFunctions( new NeoObjectVector(new Object[] { getScFieldFunction(sim,"SCLR000`
	`SurfaceDataMapper sdm = sim.get(DataMapperManager.class).createMapper( SurfaceDataMapper.class, "Surface Data Mapper" );`
	`sdm.setSourceStencil(1);`
	`sdm.getSourceParts().setObjects( source2Dregion_0, source2Dregion_1);`
	`sdm.setUpdateAvailableFields(true);`
	`sdm.setScalarFieldFunctions( new NeoObjectVector(new Object[] { getScFieldFunction(sim,"SCLR000`

Previous Release	STAR-CCM+ v11.02

`1")})`
`);`
`VolumeTargetSpecification vts =`
`((VolumeTargetSpecification)`
`vdm.getTargetSpecificationManage`
`r().getObject("Volume 1"));`
`vts.setInterpolationMethod(1);`
`vts.setTargetStencil(2);`
`vts.getTargetParts().setObjects(t`
`arget2Dregion_0,`
`target2Dregion_1);`
`vdm.mapData();`
	`1")})`
	`);`
	`SurfaceTargetSpecification sts =`
	`(`
	`(SurfaceTargetSpecification)`
	`sdm.getTargetSpecificationManage`
	`r().getObject("Surface 1")`
	`);`
	`sts.setInterpolationMethod(1);`
	`sts.setTargetStencil(1);`
	`sts.getTargetParts().setObjects(t`
	`arget2Dregion_0,`
	`target2Dregion_1);`
	`sdm.mapData();`

Virtual Disk: Changes to Body Force Propeller and 1D Momentum Methods

Options have been added to the Body Force Propeller and 1D Momentum methods, resulting in changes to the macro code.

Previous Release	STAR-CCM+ v11.02

`PropellerInflowVelocityPlane`
`inflowPlane =`
`disk.getComponentsManager().get(P`
`ropellerInflowVelocityPlane.class`
`);`
`inflowPlane.getRadius().setValue(`
`0.6);`
`inflowPlane.getOffset().setValue(`
`0.22);`
	`VdmInflowSpecification`
	`inflowSpec =`
	`disk.getComponentsManager().get(V`
	`dmInflowSpecification.class);`
	`inflowSpec.setActiveInflowMethod(`
	`VdmAverageInflowPlaneMethod.class`
	`);`
	`VdmAverageInflowPlaneMethod`
	`velPlane =`
	`(VdmAverageInflowPlaneMethod)`
	`inflowSpec.getActiveInflowMethod(`
	`);`
	`velPlane.getRadius().setValue(0.6`
	`);`
	`velPlane.getOffset().setValue(0.2`
	`2);`

Changes to Methods Related to Creation of the Simulation Process Object

The API related to creating a server and opening the simulation, as well as connecting to the running server, was changed. `SimulationProcessObject` is no longer created using its constructor directly but through the `ApplicationManager` and `SimulationApplicaton` management objects. If the

`SimulationProcessObject` is created directly, the registration with the `ApplicationManager` is not properly done and the behavior is undefined.

Examples of recommended macro changes follow for specific instances of creating a simulation process object:

To create a simulation process object and simulation using the default server connection:

Previous Release	STAR-CCM+ v11.02
`SimulationProcessObject spo = new SimulationProcessObject()`	`SimulationProcessObject spo = (SimulationProcessObject)ApplicationManager.getSingleton().getApplication(SimulationApplication.APPLICATION_TYPE).createRoot()`

To create a simulation process object and simulation for the given file:

Previous Release	STAR-CCM+ v11.02
`SimulationProcessObject spo = new SimulationProcessObject(filename)`	`SimulationProcessObject spo = (SimulationProcessObject)ApplicationManager.getSingleton().getApplication(SimulationApplication.APPLICATION_TYPE).createRoot(filename)`

To create a simulation process object by connecting to the server process at the specified host and port:

Previous Release	STAR-CCM+ v11.02
`SimulationProcessObject spo = new SimulationProcessObject(host, port)`	`SimulationProcessObject spo = (SimulationProcessObject)ApplicationManager.getSingleton().getApplication(SimulationApplication.APPLICATION_TYPE).connect(host, port)`

To create a simulation process object by connecting to the server process at the specified host and port through an SSH tunnel:

Previous Release	STAR-CCM+ v11.02
`SimulationProcessObject spo = new SimulationProcessObject(host, port, sshHost, sshUserName)`	`SimulationProcessObject spo = (SimulationProcessObject)ApplicationManager.getSingleton().getApplication(SimulationApplication.APP`

Previous Release	STAR-CCM+ v11.02
	`LICATION_TYPE).connect(host, port, sshHost, sshUserName)`

Change to Management of Open Simulations

Management of the open simulations through the `SimulationManager` class has been deprecated. The class still exists and all its methods are functional, but going forward the preferred methods of managing open simulations are through the new `ApplicationManager` and `SimulationApplication` classes.

Examples of recommended macro changes follow for specific instances of managing open simulations:

To get all simulations open in the client:

Previous Release	STAR-CCM+ v11.02
`Collection<Simulation> sl = SimulationManager.getSingleton().simulations();`	`Set<Simulation> sl = ApplicationManager.getSingleton().getApplicationAs(SimulationApplication.class, SimulationApplication.APPLICATION_TYPE).getRootObjects()`

To get the simulation for the existing client server object:

Previous Release	STAR-CCM+ v11.02
`Simulation s = SimulationManager.getSingleton().getSimulation(cso)`	`Simulation s = Simulation.getInstance(cso)`

To check whether the application is exiting:

Previous Release	STAR-CCM+ v11.02
`SimulationManager.getSingleton().isExiting()`	`ApplicationManager.getSingleton().isExiting()`

To add an observer to be notified when a new simulation appears:

Previous Release	STAR-CCM+ v11.02
`SimulationManager.getSingleton().addSimulationObserver(observer)`	`ApplicationManager.getSingleton().addRootObjectsChangedObserver(observer)`

Previous Release	STAR-CCM+ v11.02
	Note: To make sure that you respond only to additions or removals of simulations, check the type of object in the handler. For example: `// Watch for new root objects private static class RootObjectObserver extends RootObjectsChangedNotifier.NotifierReceiver { @Override protected void rootObjectAdded(RootObject root) { if (root instanceof Simulation) { // handle the new simulation added } } }`

Changes for Accessing the List of Simulation Processes

Functionality related to the list of active simulation process objects and the modified registry was moved to the new `ApplicationManager` class.

For reasons of compatibility, the methods of the previous version are still available, but marked as deprecated. It is recommended that you upgrade your macros with the methods of the current version for long-term reliability.

Examples of recommended macro changes follow for specific instances of accessing simulation processes:

To get the list of active simulation process objects:

Previous Release	STAR-CCM+ v11.02
`SimulationProcessObject[] list = SimulationProcessObject.activeSimulationProcessObjects()`	`Set<RootProcessObject> set = ApplicationManager.getSingleton().getApplication(SimulationApplication.APPLICATION_TYPE).getActiveRootProcessObjects();` **Note:** At this point the set contains all active `SimulationProcessObjects` (designated as the `RootProcessObject` base class).

To get the list of modified simulation process objects:

Previous Release	STAR-CCM+ v11.02
`Set<SimulationProcessObject> set = SimulationProcessObject.getModifiedRegistry().getModifiedSet()`	`Set<RootProcessObject> set = ApplicationManager.getSingleton().getApplication(SimulationApplication.APPLICATION_TYPE).getModifiedRootProcessObjects()`

To add an observer for changes in the list of modified simulation process objects:

Previous Release	STAR-CCM+ v11.02
`SimulationProcessObject.getModifiedRegistry().addChangeListener(changeListener)`	`ApplicationManager.getSingleton().addModifiedRootProcessObjectsChangedListener(observer)` **Note:** To make sure that you respond only to these changes, check the type of `RootProcessObject` that is passed to the observer.

Macro API Changes 10.06

In STAR-CCM+ 10.06, the macro API changed for units, shell thickness, simple shape parts, combustion, material properties, turbulence, and heat transfer.

Units: Replacement of Special Character

The special character for Angstroms, **Å**, has been replaced with **Ang** in the STAR-CCM+ interface, resulting in changes to the macro code.

Previous Release	STAR-CCM+ v10.06

`Simulation simulation_0 = getActiveSimulation();`
`MaxReport maxReport_0 = ((MaxReport) simulation_0.getReportManager().getReport("Maximum 1"));`
`Units units_0 = ((Units) simulation_0.getUnitsManager().getObject("\u212B"));`
`maxReport_0.setUnits(units_0);`
	`Simulation simulation_0 = getActiveSimulation();`
	`MaxReport maxReport_0 = ((MaxReport) simulation_0.getReportManager().getReport("Maximum 1"));`
	`Units units_0 = ((Units) simulation_0.getUnitsManager().getObject("Ang"));`
	`maxReport_0.setUnits(units_0);`

Change for Shell Thickness

The way in which STAR-CCM+ handles shell thickness has changed. Modify your existing macros accordingly.

Previous Release	STAR-CCM+ v10.06
`PrimitiveFieldFunction primitiveFieldFunction_0 = ((PrimitiveFieldFunction) simulation_0.getFieldFunctionManager().getFunction("ShellThickness"));`	`PrimitiveFieldFunction primitiveFieldFunction_0 = ((PrimitiveFieldFunction) simulation_0.getFieldFunctionManager().getFunction("SolidShellThickness"));`

Changes for Simple Shape Parts

The macro code for simple shape parts has changed.

Examples of macro steps follow for a block part, a cone, a cylinder, and a sphere.

Block Part

Previous Release	STAR-CCM+ v10.06

`SimpleBlockPart simpleBlockPart_0 =`
 	
`meshPartFactory_0.createNewBlockPart(simulation_0.get(SimulationPartManager.class));`
`simpleBlockPart_0.setDoNotRetessellate(true);`
`LabCoordinateSystem labCoordinateSystem_0 =`
 	
`simulation_0.getCoordinateSystemManager().getLabCoordinateSystem();`
`simpleBlockPart_0.setCoordinateSystem(labCoordinateSystem_0);`
`Coordinate coordinate_0 =`
`  simpleBlockPart_0.getCorner1();`
`coordinate_0.setCoordinateSystem(labCoordinateSystem_0);`
`coordinate_0.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {-1.0, -1.0, -1.0}));`
`coordinate_0.setValue(new`
	`SimpleBlockPart simpleBlockPart_0 =`
	
	`meshPartFactory_0.createNewBlockPart(simulation_0.get(SimulationPartManager.class));`
	`simpleBlockPart_0.setDoNotRetessellate(true);`
	`LabCoordinateSystem labCoordinateSystem_0 =`
	
	`simulation_0.getCoordinateSystemManager().getLabCoordinateSystem();`
	`simpleBlockPart_0.setCoordinateSystem(labCoordinateSystem_0);`
	`Coordinate coordinate_0 =`
	`  simpleBlockPart_0.getCorner1();`
	`coordinate_0.setCoordinateSystem(labCoordinateSystem_0);`
	`coordinate_0.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {0.0, 0.0, 4.0}));`
	`Coordinate coordinate_1 =`

Previous Release	STAR-CCM+ v10.06

`DoubleVector(new double[] {0.0, 0.0, 4.0}));`
`Coordinate coordinate_1 =`
`  simpleBlockPart_0.getCorner2();`
`coordinate_1.setCoordinateSystem(labCoordinateSystem_0);`
`coordinate_1.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {1.0, 1.0, 1.0}));`
`coordinate_1.setValue(new DoubleVector(new double[] {1.0, 1.0, 5.0}));`
`simpleBlockPart_0.rebuildSimpleShapePart();`
`simpleBlockPart_0.setDoNotRetessellate(false);`
	`  simpleBlockPart_0.getCorner2();`
	`coordinate_1.setCoordinateSystem(labCoordinateSystem_0);`
	`coordinate_1.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {1.0, 1.0, 5.0}));`
	`simpleBlockPart_0.rebuildSimpleShapePart();`
	`simpleBlockPart_0.setDoNotRetessellate(false);`

Cone

Previous Release	STAR-CCM+ v10.06

`SimpleConePart simpleConePart_0 =`
 	
`meshPartFactory_0.createNewConePart(simulation_0.get(SimulationPartManager.class));`
`simpleConePart_0.setDoNotRetessellate(true);`
`simpleConePart_0.setCoordinateSystem(labCoordinateSystem_0);`
`Coordinate coordinate_2 =`
 	
`simpleConePart_0.getStartCoordinate();`
`coordinate_2.setCoordinateSystem(labCoordinateSystem_0);`
`coordinate_2.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {0.0, 0.0, 1.0}));`
`coordinate_2.setValue(new DoubleVector(new double[] {0.0, 0.0, 3.0}));`
	`SimpleConePart simpleConePart_0 =`
	
	`meshPartFactory_0.createNewConePart(simulation_0.get(SimulationPartManager.class));`
	`simpleConePart_0.setDoNotRetessellate(true);`
	`simpleConePart_0.setCoordinateSystem(labCoordinateSystem_0);`
	`Coordinate coordinate_0 =`
	
	`simpleConePart_0.getStartCoordinate();`
	`coordinate_0.setCoordinateSystem(labCoordinateSystem_0);`
	`coordinate_0.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {0.0, 0.0, 3.0}));`
	`simpleConePart_0.getStartRadius().setUnits(units_0);`
	`simpleConePart_0.getStartRadius()`

Previous Release	STAR-CCM+ v10.06

`simpleConePart_0.getStartRadius()`
`.setUnits(units_0);`
`simpleConePart_0.getStartRadius()`
`.setValue(1.0);`
`Coordinate coordinate_3 =`
 	
`simpleConePart_0.getEndCoordinat`
`e();`
`coordinate_3.setCoordinateSystem(`
`labCoordinateSystem_0);`
`coordinate_3.setCoordinate(units_`
`0, units_0, units_0, new`
`DoubleVector(new double[] {0.0,`
`0.0, 0.0}));`
`coordinate_3.setValue(new`
`DoubleVector(new double[] {0.0,`
`0.0, 4.0}));`
`simpleConePart_0.getEndRadius().s`
`etUnits(units_0);`
`simpleConePart_0.getEndRadius().s`
`etValue(1.0);`
`simpleConePart_0.getTessellationD`
`ensityOption().setSelected(Tessel`
`lationDensityOption.MEDIUM);`
`simpleConePart_0.rebuildSimpleSha`
`pePart();`
`simpleConePart_0.setDoNotRetessel`
`late(false);`
	`.setValue(1.0);`
	`Coordinate coordinate_1 =`
	
	`simpleConePart_0.getEndCoordinat`
	`e();`
	`coordinate_1.setCoordinateSystem(`
	`labCoordinateSystem_0);`
	`coordinate_1.setCoordinate(units_`
	`0, units_0, units_0, new`
	`DoubleVector(new double[] {0.0,`
	`0.0, 4.0}));`
	`simpleConePart_0.getEndRadius().s`
	`etUnits(units_0);`
	`simpleConePart_0.getEndRadius().s`
	`etValue(1.0);`
	`simpleConePart_0.getTessellationD`
	`ensityOption().setSelected(Tessel`
	`lationDensityOption.Type.MEDIUM);`
	`simpleConePart_0.rebuildSimpleSha`
	`pePart();`
	`simpleConePart_0.setDoNotRetessel`
	`late(false);`

Cylinder

Previous Release	STAR-CCM+ v10.06

`SimpleCylinderPart`
`simpleCylinderPart_0 =`
 	
`meshPartFactory_0.createNewCylind`
`erPart(simulation_0.get(Simulatio`
`nPartManager.class));`
`simpleCylinderPart_0.setDoNotRete`
`ssellate(true);`
`simpleCylinderPart_0.setCoordinat`
`eSystem(labCoordinateSystem_0);`
`Coordinate coordinate_4 =`
	`SimpleCylinderPart`
	`simpleCylinderPart_0 =`
	
	`meshPartFactory_0.createNewCylind`
	`erPart(simulation_0.get(Simulatio`
	`nPartManager.class));`
	`simpleCylinderPart_0.setDoNotRete`
	`ssellate(true);`
	`simpleCylinderPart_0.setCoordinat`
	`eSystem(labCoordinateSystem_0);`
	`Coordinate coordinate_2 =`

Previous Release	STAR-CCM+ v10.06

`simpleCylinderPart_0.getStartCoordinate();`
`coordinate_4.setCoordinateSystem(labCoordinateSystem_0);`
`coordinate_4.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {0.0, 0.0, 1.0}));`
`coordinate_4.setValue(new DoubleVector(new double[] {0.0, 0.0, 6.0}));`
`Coordinate coordinate_5 =`
 	
`simpleCylinderPart_0.getEndCoordinate();`
`coordinate_5.setCoordinateSystem(labCoordinateSystem_0);`
`coordinate_5.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {0.0, 0.0, 0.0}));`
`coordinate_5.setValue(new DoubleVector(new double[] {0.0, 0.0, 7.0}));`
`simpleCylinderPart_0.getRadius().setUnits(units_0);`
`simpleCylinderPart_0.getRadius().setValue(1.0);`
`simpleCylinderPart_0.getTessellationDensityOption().setSelected(TessellationDensityOption.MEDIUM);`
`simpleCylinderPart_0.rebuildSimpleShapePart();`
`simpleCylinderPart_0.setDoNotRetessellate(false);`
	`simpleCylinderPart_0.getStartCoordinate();`
	`coordinate_2.setCoordinateSystem(labCoordinateSystem_0);`
	`coordinate_2.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {0.0, 0.0, 6.0}));`
	`Coordinate coordinate_3 =`
	
	`simpleCylinderPart_0.getEndCoordinate();`
	`coordinate_3.setCoordinateSystem(labCoordinateSystem_0);`
	`coordinate_3.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {0.0, 0.0, 7.0}));`
	`simpleCylinderPart_0.getRadius().setUnits(units_0);`
	`simpleCylinderPart_0.getRadius().setValue(1.0);`
	`simpleCylinderPart_0.getTessellationDensityOption().setSelected(TessellationDensityOption.Type.MEDIUM);`
	`simpleCylinderPart_0.rebuildSimpleShapePart();`
	`simpleCylinderPart_0.setDoNotRetessellate(false);`

Sphere

Previous Release	STAR-CCM+ v10.06

`SimpleSpherePart simpleSpherePart_0 =`
 	
`meshPartFactory_0.createNewSpherePart(simulation_0.get(SimulationP`
	`SimpleSpherePart simpleSpherePart_0 =`
	
	`meshPartFactory_0.createNewSpherePart(simulation_0.get(SimulationP`

Previous Release	STAR-CCM+ v10.06

`artManager.class));`
`simpleSpherePart_0.setDoNotRetessellate(true);`
`LabCoordinateSystem labCoordinateSystem_0 =`
 	
`simulation_0.getCoordinateSystemManager().getLabCoordinateSystem();`
`simpleSpherePart_0.setCoordinateSystem(labCoordinateSystem_0);`
`Coordinate coordinate_0 =`
`  simpleSpherePart_0.getOrigin();`
`coordinate_0.setCoordinateSystem(labCoordinateSystem_0);`
`coordinate_0.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {0.0, 0.0, 0.0}));`
`coordinate_0.setValue(new DoubleVector(new double[] {0.0, 0.0, 9.0}));`
`simpleSpherePart_0.getRadius().setUnits(units_0);`
`simpleSpherePart_0.getRadius().setValue(1.0);`
`simpleSpherePart_0.getTessellationDensityOption().setSelected(TessellationDensityOption.MEDIUM);`
`simpleSpherePart_0.rebuildSimpleShapePart();`
`simpleSpherePart_0.setDoNotRetessellate(false);`
	`artManager.class));`
	`simpleSpherePart_0.setDoNotRetessellate(true);`
	`LabCoordinateSystem labCoordinateSystem_0 =`
	
	`simulation_0.getCoordinateSystemManager().getLabCoordinateSystem();`
	`simpleSpherePart_0.setCoordinateSystem(labCoordinateSystem_0);`
	`Coordinate coordinate_0 =`
	`  simpleSpherePart_0.getOrigin();`
	`coordinate_0.setCoordinateSystem(labCoordinateSystem_0);`
	`coordinate_0.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {0.0, 0.0, 9.0}));`
	`simpleSpherePart_0.getRadius().setUnits(units_0);`
	`simpleSpherePart_0.getRadius().setValue(1.0);`
	`simpleSpherePart_0.getTessellationDensityOption().setSelected(TessellationDensityOption.Type.MEDIUM);`
	`simpleSpherePart_0.rebuildSimpleShapePart();`
	`simpleSpherePart_0.setDoNotRetessellate(false);`

Combustion: Changes in PPDF and Nox

The following changes have been made to PPDF and Prompt Nox models, resulting in changes to the macro code:

- Removal of PPDF update options
- Change in use of Weighted Sum of Gray Gases (WSGG) method
- Replacement of PPDF_TABLE with SPECIES LIST for the following: OxygenAtomModelOption, OHGroupModelOption, and AcetyleneModelOption
- Removal of the redundant Prompt Nox Species Model option

Examples of macro steps follow for these changes.

PPDF Update

Previous Release	STAR-CCM+ v10.06
`PpdfNonAdiabaticFlameletPartiallyPremixedModel ppdfFlameletPartiallyPremixedModel_0 = continuum.getModelManager().getModel(PpdfNonAdiabaticFlameletPartiallyPremixedModel.class); PpdfSpeciesUpdate ppdfSpeciesUpdate_0 = ppdfFlameletPartiallyPremixedModel_0.getPpdfSpeciesUpdate(); PpdfSpeciesUpdateFrequency ppdfSpeciesUpdateFrequency_0 = ppdfSpeciesUpdate_0.getPpdfSpeciesUpdateFrequency(); ppdfSpeciesUpdateFrequency_0.setThreshold(100); PpdfFlameletTable ppdfFlameletTable_0 = ((PpdfFlameletTable) ppdfFlameletPartiallyPremixedModel_0.getPpdfFlameletTable()); ppdfFlameletTable_0.importTable(resolveDataPath("case/PpdfFlameletDropComb.tbl"));`	`PpdfNonAdiabaticFlameletPartiallyPremixedModel ppdfFlameletPartiallyPremixedModel_0 = continuum.getModelManager().getModel(PpdfNonAdiabaticFlameletPartiallyPremixedModel.class); PpdfFlameletTable ppdfFlameletTable_0 = ((PpdfFlameletTable) ppdfFlameletPartiallyPremixedModel_0.getPpdfFlameletTable()); ppdfFlameletTable_0.importTable(resolveDataPath("case/PpdfFlameletDropComb.tbl"));`

WSGG

Previous Release	STAR-CCM+ v10.06
`gasMixture_0.getMaterialProperties().getMaterialProperty(AbsorptionCoefficientProperty.class).setMethod(PpdfWsggMethod.class);`	`gasMixture_0.getMaterialProperties().getMaterialProperty(AbsorptionCoefficientProperty.class).setMethod(WsggMethod.class);`

SPECIES LIST

Previous Release	STAR-CCM+ v10.06
`noxGeneralZeldovichModel_0.getOHGroupModelOption().setSelected(OHGroupModelOption.PPDF_TABLE); noxGeneralZeldovichModel_0.getOxy`	`noxGeneralZeldovichModel_0.getOHGroupModelOption().setSelected(OHGroupModelOption.SPECIES_LIST); noxGeneralZeldovichModel_0.getOxy`

Previous Release	STAR-CCM+ v10.06
`genAtomModelOption().setSelected(OxygenAtomModelOption.PPDF_TABLE);`	`genAtomModelOption().setSelected(OxygenAtomModelOption.SPECIES_LIST);`
`speciesConcentrationForSoot_0.getAcetyleneModelOption().setSelected(AcetyleneModelOption.PPDF_TABLE);`	`speciesConcentrationForSoot_0.getOxygenAtomModelOption().setSelected(OxygenAtomModelOption.SPECIES_LIST);`

Species Model

Previous Release	STAR-CCM+ v10.06

`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Physics 1"));`
`PromptNoxModel promptNoxModel_0 = physicsContinuum_0.getModelManager().getModel(PromptNoxModel.class);`
`promptNoxModel_0.getPromptNoxSpeciesModelOption().setSelected(PromptNoxSpeciesModelOption.PPDF_TABLE);`
`promptNoxModel_0.setCorrectionFactor(2.43);`
	`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Physics 1"));`
	`PromptNoxModel promptNoxModel_0 = physicsContinuum_0.getModelManager().getModel(PromptNoxModel.class);`
	`promptNoxModel_0.setCorrectionFactor(2.43);`

Combustion: Changes in Specific Heat Method Registration

The macro code has changed for setting the method of the **Specific Heat** material property to **PPDF Flamelet Table**.

Note: You cannot select this method unless you have constructed or imported a PPDF table.

Previous Release	STAR-CCM+ v10.06
`PpdfAdiabaticFlameletPartiallyPremixedModel PpdfAdiabaticFlameletPartiallyPremixedModel_0 = physicsContinuum_0.getModelManager().getModel(PpdfAdiabaticFlameletPartiallyPremixedModel.class);`	`PpdfAdiabaticFlameletPartiallyPremixedModel PpdfAdiabaticFlameletPartiallyPremixedModel_0 = physicsContinuum_0.getModelManager().getModel(PpdfAdiabaticFlameletPartiallyPremixedModel.class);`

Previous Release	STAR-CCM+ v10.06
`PpdfFlameletTable ppdfFlameletTable_0 = ((PpdfFlameletTable) PpdfAdiabaticFlameletPartiallyPremixedModel_0.getPpdfFlameletTable());` `MultiComponentGasModel multiComponentGasModel_0 = ((MultiComponentGasModel) physicsContinuum_0.getModelManager().getModel(MultiComponentGasModel.class));` `((SpecificHeatProperty) gasComponent_0.getMaterialProperties().getMaterialProperty(SpecificHeatProperty.class)).setMethod(PpdfFlameletSpecificHeatMethod.class);` `ppdfFlameletTable_0.importTable(resolveDataPath("case/CH4_Flamelet_29_16.tbl"));`	`PpdfFlameletTable ppdfFlameletTable_0 = ((PpdfFlameletTable) PpdfAdiabaticFlameletPartiallyPremixedModel_0.getPpdfFlameletTable());` `ppdfFlameletTable_0.importTable(resolveDataPath("case/CH4_Flamelet_29_16.tbl"));` `MultiComponentGasModel multiComponentGasModel_0 = ((MultiComponentGasModel) physicsContinuum_0.getModelManager().getModel(MultiComponentGasModel.class));` `((SpecificHeatProperty) gasComponent_0.getMaterialProperties().getMaterialProperty(SpecificHeatProperty.class)).setMethod(PpdfFlameletSpecificHeatMethod.class);`

Material Properties: Change to Lennard-Jones Material Property

The **Lennard-Jones Potential** material property has been divided into the **Lennard-Jones Energy** and **Lennard-Jones Characteristic Length** material properties, resulting in changes to the macro code.

Previous Release	STAR-CCM+ v10.06
`ConstantLennardJonesPotentialMethod constantLennardJonesPotentialMethod_0 =` `((ConstantLennardJonesPotentialMethod) gas_0.getMaterialProperties().getMaterialProperty(LennardJonesPotentialProperty.class).getMethod());` `constantLennardJonesPotentialMethod_0.getSigmaQuantity().setValue(5.0);`	`ConstantMaterialPropertyMethod constantMaterialPropertyMethod_0 =` `((ConstantMaterialPropertyMethod)` `gas_0.getMaterialProperties().getMaterialProperty(LennardJonesCharacteristicLengthProperty.class).getMethod());` `constantMaterialPropertyMethod_0.getQuantity().setValue(5.0);` `ConstantMaterialPropertyMethod constantMaterialPropertyMethod_1 =`

Previous Release	STAR-CCM+ v10.06

`constantLennardJonesPotentialMethod_0.getEpsilonQuantity().setValue(100.0);`	`((ConstantMaterialPropertyMethod)`
	
	`gas_0.getMaterialProperties().getMaterialProperty(LennardJonesCharacteristicEnergyProperty.class).getMethod());`
	
	`constantMaterialPropertyMethod_1.getQuantity().setValue(100.0);`

Turbulence Response Models: Change to Phase Interaction

The Turbulence Response models now select a reference phase interaction in their properties instead of a reference continuous phase.

The following example uses the Tchen Turbulence Response model. However, similar macro changes apply to the Issa and user-defined Turbulence Response models.

Previous Release	STAR-CCM+ v10.06

`tchenTurbulenceResponseModel_1.setReferencePhase(eulerianPhase_0);`	`MultiPhaseInteractionModel multiPhaseInteractionModel_0 = physicsContinuum_0.getModelManager().getModel(MultiPhaseInteractionModel.class);`
	`PhaseInteraction phaseInteraction_0 = ((PhaseInteraction) multiPhaseInteractionModel_0.getPhaseInteractionManager().getPhaseInteraction("solid-water"));`
	`tchenTurbulenceResponseModel_1.setReferencePhaseInteraction(phaseInteraction_0);`

Material Properties: Change in Molecular Weight Material Property Method

The **Read-only constant** method of the **Molecular Weight** material property has been renamed, resulting in changes to the macro code.

Previous Release	STAR-CCM+ v10.06

`GasComponent gasComponent_0 =`
`  ((GasComponent)`
`gasMixture_0.getComponents().getC`
	`GasComponent gasComponent_0 =`
	`  ((GasComponent)`
	`gasMixture_0.getComponents().getC`

Previous Release	STAR-CCM+ v10.06

`omponent("CH4"));`
`gasComponent_0.getMaterialPropert`
`ies().getMaterialProperty(Molecular`
`WeightProperty.class).setMetho`
`d(ElementalCompositionMolecularWe`
`ightMethod.class);`
`gasComponent_0.getMaterialPropert`
`ies().getMaterialProperty(Molecular`
`WeightProperty.class).setMetho`
`d(ReadOnlyConstantMolecularWeight`
`Method.class);`
	`omponent("CH4"));`
	`gasComponent_0.getMaterialPropert`
	`ies().getMaterialProperty(Molecular`
	`WeightProperty.class).setMetho`
	`d(ElementalCompositionMolecularWe`
	`ightMethod.class);`
	`gasComponent_0.getMaterialPropert`
	`ies().getMaterialProperty(Molecular`
	`WeightProperty.class).setMetho`
	`d(ReadOnlyConstantMaterialPropert`
	`yMethod.class);`

Heat Transfer: Change to Heat Transfer Coefficient Field Function

The Heat Transfer Coefficient field function has been changed to work more reliably with harmonic balance and co-simulation. This change affects the macro code.

Previous Release	STAR-CCM+ v10.06

`HeatTransferCoefficientFunction`
`heatTransferCoefficientFunction_0`
`=`
 	
`((HeatTransferCoefficientFunction`
`)`
`simulation_0.getFieldFunctionMana`
`ger().getFunction("HeatTransferCo`
`efficient"));`
`HeatTransferCoefficientUserYPlusF`
`unction`
`heatTransferCoefficientUserYPlusF`
`unction_0 =`
`((HeatTransferCoefficientUserYPlu`
`sFunction)`
`sim.getFieldFunctionManager().get`
`Function("HeatTransferCoefficient`
`UserYPlus"));`
	`HtcFunction htcFunction_0 =`
	`  ((HtcFunction)`
	`simulation_0.getFieldFunctionMana`
	`ger().getFunction("HeatTransferCo`
	`efficient"));`
	`HtcUserYPlusFunction`
	`htcUserYPlusFunction_0 =`
	`  ((HtcUserYPlusFunction)`
	`simulation_0.getFieldFunctionMana`
	`ger().getFunction("HeatTransferCo`
	`efficientUserYPlus"));`

Macro API Changes 10.04

In STAR-CCM+ 10.04, the macro API changed for battery modeling, aeroacoustics, finite-volume solid stress, granular particle phases, motions, pressure boundaries, and meshing.

Battery Modeling: Update to Circuit Solver for Multiple Circuits

The Circuit Solver has been updated to support multiple circuits. This added capability results in new macro code.

Previous Release	STAR-CCM+ v10.04
`protected void setCircuit(Simulation sim) { CircuitTool circuitTool = sim.get(CircuitTool.class); CircuitSolver circuitSolver = circuitTool.getCircuitSolver(); ScalarCircuitElement circuitElementR = circuitSolver.getElements().creat eCircuitElement(ScalarCircuitElem ent.class); circuitElementR.getLoadOption().s etSelected(LoadOption.RESISTANCE) ;`	`protected void setCircuit(Simulation sim) { CircuitManager cMgr = sim.get(CircuitManager.class); Circuit circuit = cMgr.getObject("Circuit 1"); ScalarCircuitElement circuitElementR = circuit.getElements().createCircu itElement(ScalarCircuitElement.cl ass); circuitElementR.getLoadOption().s etSelected(LoadOption.RESISTANCE) ;`

Aeroacoustics Modeling: Changes to the Curle and Proudman Models

You can now select the Curle and Proudman aeroacoustics models simultaneously. This added capability results in new macro code.

Previous Release	STAR-CCM+ v10.04
`continuum.enable(CurleModel.class );`	`continuum.enable(BroadbandNoiseMo del.class); continuum.enable(CurleModel.class );`
`continuum.enable(ProudmanModel.cl ass);`	`continuum.enable(BroadbandNoiseMo del.class); continuum.enable(ProudmanModel.cl ass);`

Finite-Volume Solid Stress: Field Function Name Changes

Certain field functions that relate to finite-volume solid stress have been renamed. Use the new field function names in your macros.

Previous Release	STAR-CCM+ v10.04
`PrimitiveFieldFunction primitiveFieldFunction_1 = ((PrimitiveFieldFunction) simulation_0.getFieldFunctionManager().getFunction("Solid Stress Displacement"));`	`PrimitiveFieldFunction primitiveFieldFunction_1 = ((PrimitiveFieldFunction) simulation_0.getFieldFunctionManager().getFunction("Displacement"));`
`PrimitiveFieldFunction primitiveFieldFunction_1 = ((PrimitiveFieldFunction) simulation_0.getFieldFunctionManager().getFunction("Solid Stress Velocity"));`	`PrimitiveFieldFunction primitiveFieldFunction_1 = ((PrimitiveFieldFunction) simulation_0.getFieldFunctionManager().getFunction("Solid Velocity"));`
`PrimitiveFieldFunction primitiveFieldFunction_1 = ((PrimitiveFieldFunction) simulation_0.getFieldFunctionManager().getFunction("Solid Stress Acceleration"));`	`PrimitiveFieldFunction primitiveFieldFunction_1 = ((PrimitiveFieldFunction) simulation_0.getFieldFunctionManager().getFunction("Acceleration"));`

Granular Particle Phases: Turbulence Available

The Standard K-Epsilon Two-Layer phase model is now available with the granular particle phase. When you set up a granular particle phase, make an explicit choice between the Laminar and Turbulent viscous regimes.

Previous Release	STAR-CCM+ v10.04

`EulerianPhase eulerianPhase_1 =`
 	
`eulerianMultiPhaseModel_1.createPhase();`
 	
`eulerianPhase_1.setPresentationName("Particle");`
 	
`eulerianPhase_1.enable(SinglePhaseParticleModel.class);`
 	
`eulerianPhase_1.enable(EulerianPhaseLaminarModel.class);`
 	
`eulerianPhase_1.enable(ConstantDensityModel.class);`
	`EulerianPhase eulerianPhase_1 =`
	
	`eulerianMultiPhaseModel_1.createPhase();`
	
	`eulerianPhase_1.setPresentationName("Particle");`
	
	`eulerianPhase_1.enable(SinglePhaseParticleModel.class);`
	`EulerianPhaseTurbulentModel eulerianPhaseTurbulentModel_0 =`
	
	`eulerianPhase_1.getModelManager().getModel(EulerianPhaseTurbulentModel.class);`

Previous Release	STAR-CCM+ v10.04

	`eulerianPhase_1.disableModel(eulerianPhaseTurbulentModel_0);`
	
	`eulerianPhase_1.enable(EulerianPhaseLaminarModel.class);`
	
	`eulerianPhase_1.enable(ConstantDensityModel.class);`

Motions: Change in Specification of Morpher Displacement

In cases with a morpher motion, the morpher displacement option is now selected in the properties of a boundary Physics Conditions manager node. This change in the user interface results in new macro code.

Previous Release	STAR-CCM+ v10.04

`package macro;`
`import java.util.*;`
`import star.morpher.*;`
`import star.common.*;`
`import star.base.neo.*;`
`public class make_total_old`
`extends StarMacro {`
`  public void execute() {`
`  execute0();`
`  }`
`  private void execute0() {`
`  Simulation simulation_0 =`
`  getActiveSimulation();`
`  Region region_0 =`
 	
`simulation_0.getRegionManager().getRegion("Block");`
`  Boundary boundary_0 =`
 	
`region_0.getBoundaryManager().getBoundary("Block Surface");`
 	
`boundary_0.getConditions().get(MovingMeshOption.class).setSelected(MovingMeshOption.TOTAL_DISPLACEMENT);`
`  TotalDisplacementProfile totalDisplacementProfile_0 =`
	`package macro;`
	`import java.util.*;`
	`import star.morpher.*;`
	`import star.common.*;`
	`import star.base.neo.*;`
	`public class make_total_new`
	`extends StarMacro {`
	`  public void execute() {`
	`  execute0();`
	`  }`
	`  private void execute0() {`
	`  Simulation simulation_0 =`
	`  getActiveSimulation();`
	`  Region region_0 =`
	
	`simulation_0.getRegionManager().getRegion("Block");`
	`  Boundary boundary_0 =`
	
	`region_0.getBoundaryManager().getBoundary("Block Surface");`
	
	`boundary_0.getConditions().get(MovingMeshOption.class).setSelected(MovingMeshOption.DISPLACEMENT);`
	`  // Set total`
	
	`boundary_0.getConditions().get(DisplacementOption.class).setSelect`

Previous Release	STAR-CCM+ v10.04

`boundary_0.getValues().get(TotalDisplacementProfile.class);`
`    // Set total`
 	
`totalDisplacementProfile_0.setTotalDisplacement(true);`
`    // Set incremental`
 	
`totalDisplacementProfile_0.setTotalDisplacement(false);`
`  }`
`}`
	`ed(DisplacementOption.TOTAL);`
	`    // Set incremental`
	
	`boundary_0.getConditions().get(DisplacementOption.class).setSelected(DisplacementOption.INCREMENTAL);`
	`  }`
	`}`

Pressure Boundaries: Redesign of Options

The interface for pressure boundary specification has been improved, resulting in changes to the macro code.

Previous Release	STAR-CCM+ v10.04

`boundary_1.getConditions().get(ReversedFlowPressureOption.class).setSelected(ReversedFlowPressureOption.STATIC);`
 	
`boundary_0.getConditions().get(InletBoundaryPressureJumpOption.class).setSelected(InletBoundaryPressureJumpOption.FAN);`
	`boundary_1.getConditions().get(BackflowSpecification.class).getReversedFlowPressureOption().setSelected(ReversedFlowPressureOption.STATIC);`
	
	`boundary_0.getConditions().get(PressureBoundaryOption.class).setSelected(PressureBoundaryOption.PRESSURE_JUMP);`
	`boundary_0.getConditions().get(InletBoundaryPressureJumpOption.class).setSelected(InletBoundaryPressureJumpOption.FAN);`

Meshing: Surface Growth Rate in Surface Remesher

If you are customizing the surface growth rate of the Surface Remesher, update your macro code. However, you can use the old macro code for the surface growth rate of the Trimmer Mesher.

Previous Release	STAR-CCM+ v10.04
`SurfaceCustomMeshControl surfaceCustomMeshControl_0 = autoMeshOperation_0.getCustomMeshControls().createSurfaceControl()`	`SurfaceCustomMeshControl surfaceCustomMeshControl_0 = autoMeshOperation_0.getCustomMeshControls().createSurfaceControl()`

Previous Release	STAR-CCM+ v10.04
`; surfaceCustomMeshControl_0.getCustomConditions().get(PartsSurfaceGrowthRateOption.class).setSelected(PartsSurfaceGrowthRateOption.CUSTOM_VALUES);`	`; surfaceCustomMeshControl_0.getCustomConditions().get(PartsResurfacerSurfaceGrowthRateOption.class).setSelected(PartsResurfacerSurfaceGrowthRateOption.CUSTOM_VALUES);`

Derived Parts: Change to Displayer Method

The method `partDisplayer_0.getParts().addParts(planeSection_1)` has been changed to `partDisplayer_0.getInputParts().add(planeSection_1)`. Modify your existing macros accordingly.

Macro API Changes 10.02

In STAR-CCM+ 10.02, the macro API changed for multiphase segregated flow, meshing, and combustion.

Multiphase Segregated Flow: Changes to the Explicit URF for the Volume Fraction and for S-Gamma

For macros that work with the explicit under-relaxation factor either for Volume Fraction or for S-Gamma, the macro code has changed. The method `setExplicitRelaxationFactor()` has been shortened to `setExplicitUrf()`.

Previous Release	STAR-CCM+ v10.02
`volumeFractionSolver_0.setExplicitRelaxationFactor(0.3);`	`volumeFractionSolver_0.setExplicitUrf(0.3);`

Multiphase Segregated Flow: Changes to Drag Specification in Eulerian Multiphase Modeling

The macro code for Drag Correction specification has changed.

Previous Release	STAR-CCM+ v10.02
`((DragCoefficientCorrectionMethodManager) eulerianDragForceModel_0.getCcInteractionCoefficient()).setMethod(VolumeFractionExponentMethod.class);`	`((DragCoefficientCorrectionMethodManager) eulerianDragForceModel_0.getCoefficientCorrectionMethods()).setMethod(VolumeFractionExponentMethod.class);`

Macros that used Linearized specification for drag now use a new method. The drag methods that were available under linearized specification were: Gidaspow, Syamlal, Arastoopour, Symmetric, Suspension, and Emulsion.

Gidaspow

Previous Release	STAR-CCM+ v10.02
`eulerianDragForceModel_0.getCoefficientOption().setSelected(DragForceCoefficientOption.LINEARIZED);` `((LinearizedDragCoefficientMethodManager) eulerianDragForceModel_0.getInteractionCoefficient()).setMethod(GidaspowDragCoefficientMethod.class);`	`((EulerianDragCoefficientMethodManager) eulerianDragForceModel_0.getInteractionCoefficient()).setMethod(GidaspowDragCoefficientMethod.class);`

Syamlal

Previous Release	STAR-CCM+ v10.02
`eulerianDragForceModel_0.getCoefficientOption().setSelected(DragForceCoefficientOption.LINEARIZED);` `(((LinearizedDragCoefficientMethodManager) eulerianDragForceModel_0.getInteractionCoefficient()).setMethod(SyamlalOBrienDragCoefficientMethod.class);`	`((EulerianDragCoefficientMethodManager) eulerianDragForceModel_0.getInteractionCoefficient()).setMethod(SyamlalOBrienDragCoefficientMethod.class);`

Arastoopour

Previous Release	STAR-CCM+ v10.02
`eulerianDragForceModel_0.getCoefficientOption().setSelected(DragForceCoefficientOption.LINEARIZED);` `((LinearizedDragCoefficientMethodManager) eulerianDragForceModel_0.getInteractionCoefficient()).setMethod(ArastoopourDragCoefficientMethod.class);`	`((EulerianDragCoefficientMethodManager) eulerianDragForceModel_0.getInteractionCoefficient()).setMethod(ArastoopourDragCoefficientMethod.class);`

Symmetric

Previous Release	STAR-CCM+ v10.02
`eulerianDragForceModel_0.getCoefficientOption().setSelected(DragForceCoefficientOption.LINEARIZED);` `((LinearizedDragCoefficientMethodManager) eulerianDragForceModel_0.getInteractionCoefficient()).setMethod(SymmetricDragCoefficientMethod.class);`	`((EulerianDragCoefficientMethodManager) eulerianDragForceModel_0.getInteractionCoefficient()).setMethod(SymmetricDragCoefficientMethod.class);`

Suspension

Previous Release	STAR-CCM+ v10.02
`eulerianDragForceModel_0.getCoefficientOption().setSelected(DragForceCoefficientOption.LINEARIZED);` `((LinearizedDragCoefficientMethodManager) eulerianDragForceModel_0.getInteractionCoefficient()).setMethod(SuspensionDragCoefficientMethod.class);`	`((EulerianDragCoefficientMethodManager) eulerianDragForceModel_0.getInteractionCoefficient()).setMethod(SuspensionDragCoefficientMethod.class);`

Emulsion

Previous Release	STAR-CCM+ v10.02
`eulerianDragForceModel_0.getCoefficientOption().setSelected(DragForceCoefficientOption.LINEARIZED);` `((LinearizedDragCoefficientMethodManager) eulerianDragForceModel_0.getInteractionCoefficient()).setMethod(EmulsionInversionDragCoefficientMethod.class);`	`((EulerianDragCoefficientMethodManager) eulerianDragForceModel_0.getInteractionCoefficient()).setMethod(EmulsionInversionDragCoefficientMethod.class);`

Multiphase Segregated Flow: Changes to the Virtual Mass Coefficient

The Virtual Mass Coefficient model needs to be activated under relevant phase interactions for simulations that use any of the following:

- Virtual Mass Force
- Turbulent Dispersion Force

- Tchen Particle Induced Turbulence
- Tchen Turbulent Response (in the dispersed phase of the interaction)

Previous Release	STAR-CCM+ v10.02
`phaseInteraction_0.enable(TurbulentDispersionForceModel.class);`	`phaseInteraction_0.enable(TurbulentDispersionForceModel.class);` `phaseInteraction_0.enable(VirtualMassCoefficientModel.class);`

If a non-default virtual mass coefficient was previously defined under the Virtual Mass Force model, then this specification should now be applied to the Virtual Mass Coefficient model.

Previous Release	STAR-CCM+ v10.02

`VirtualMassForceModel virtualMassForceModel_0 =`
 	
`phaseInteraction_0.getModelManager().getModel(VirtualMassForceModel.class);`
`virtualMassForceModel_0.getVirtualMassCoefficient().setMethod(ConstantInteractionCoefficientMethod.class);`
	`VirtualMassCoefficientModel virtualMassCoefficientModel_0 =`
	
	`phaseInteraction_0.getModelManager().getModel(VirtualMassCoefficientModel.class);`
	`virtualMassCoefficientModel_0.getVirtualMassCoefficient().setMethod(ConstantInteractionCoefficientMethod.class);`

Meshing: Changes to Part Surface Contact Location

The way in which STAR-CCM+ handles part surface contacts has changed. Modify your existing macros accordingly.

Previous Release	STAR-CCM+ v10.02
`import star.meshing.PartSurfaceContact;` `import star.meshing.PartSurfaceContactManager;`	`import star.common.PartSurfaceContact;` `import star.common.PartSurfaceContactManager;`

Combustion: Changes to Multiple Site Mixtures for One Surface Reaction Mechanism

A surface reaction mechanism can use detailed surface reactions with multiple sites and site species. This added capability results in new macro code.

Previous Release	STAR-CCM+ v10.02
`BulkSurfaceFractionProfile bulkSurfaceFractionProfile_0 = boundary_0.getValues().get(BulkSurfaceFractionProfile.class);`	`BulkSurfaceFractionProfileManager bulkSurfaceFractionProfileManager_0 = boundary_0.getValues().get(BulkSurfaceFractionProfileManager.class); BulkSurfaceFractionProfile bulkSurfaceFractionProfile_0 = bulkSurfaceFractionProfileManager_0.getProfile(0);`
`SiteSurfaceFractionProfile siteSurfaceFractionProfile_0 = boundary_0.getValues().get(SiteSurfaceFractionProfile.class);`	`SiteSurfaceFractionProfileManager siteSurfaceFractionProfileManager_0 = boundary_0.getValues().get(SiteSurfaceFractionProfileManager.class); SiteSurfaceFractionProfile siteSurfaceFractionProfile_0 = siteSurfaceFractionProfileManager_0.getProfile(0);`

Combustion: Changes to Zone Manager for Reacting Channel

Reacting channel zones are boundaries in the CFD region where data is transferred between the Plug Flow Reactor (PFR) and STAR-CCM+. The changes to the macros for reacting channels are extensive, so you are advised to re-record them.

Macro API Changes 9.06

In STAR-CCM+ 9.06, the macro API changed for the On-The-Fly FW-H model, the Circuit Solver, the Dars-CFD Reaction model, the Fan Interface model, and simple shape parts.

Changes to the On-The-Fly FW-H Model

The FW-H Field Functions: `FwhLoadingSurfaceTerm`, `FwhThicknessSurfaceTerm`, `FwhTotalSurfaceTerm`, and `FwhQuadrupoleVolumeTerm` now use a unique and persistent index for each Receiver. Consequently, any old macro that sets these On-the-Fly FW-H Model field functions will fail to compile.

Previous Release	STAR-CCM+ v9.06
```((PrimitiveFieldFunction) getSimulation().getFieldFunctionManager() .getFunction("LoadingSurfaceTermstar.acoustics.PointReceiver_0")); ((PrimitiveFieldFunction) getSimulation().getFieldFunctionManager() .getFunction("ThicknessSurfaceTermstar.acoustics.PointReceiver_0")); ((PrimitiveFieldFunction) getSimulation().getFieldFunctionManager() .getFunction("TotalSurfaceTermstar.acoustics.PointReceiver_0")); ((PrimitiveFieldFunction) getSimulation().getFieldFunctionManager() .getFunction("QuadrupoleVolumeTermstar.acoustics.PointReceiver_0"));```	```((PrimitiveFieldFunction) getSimulation().getFieldFunctionManager() .getFunction("Receiver0::LoadingSurfaceTerm")); ((PrimitiveFieldFunction) getSimulation().getFieldFunctionManager() .getFunction("Receiver0::ThicknessSurfaceTerm")); ((PrimitiveFieldFunction) getSimulation().getFieldFunctionManager() .getFunction("Receiver0::TotalSurfaceTerm")); ((PrimitiveFieldFunction) getSimulation().getFieldFunctionManager() .getFunction("Receiver0::QuadrupoleVolumeTerm"));```

Changes to the Circuit Solver

In STAR-CCM+ v9.06, the circuit solver has been reorganized to have its own library. Macros that were recorded in previous versions of STAR-CCM+ must be modified to import the circuit solver library: `import star.circuit`.

Previous Release	STAR-CCM+ v9.06
```package base.battery; import star.common.MonitorPlot; import star.common.Simulation; import star.common.StarMacro; import java.io.File; import star.material.*; import star.common.*; import star.base.neo.*;```	```package base.battery; import star.common.MonitorPlot; import star.common.Simulation; import star.common.StarMacro; import java.io.File; import star.material.*; import star.common.*; import star.base.neo.*;```

Previous Release	STAR-CCM+ v9.06

`import star.segregatedenergy.*;`
`import star.base.report.*;`
`import star.battery.*;`
`import star.solidmesher.*;`
`import star.flow.*;`
`import star.meshing.*;`
`import star.metrics.*;`
`import star.energy.*;`
	`import star.segregatedenergy.*;`
	`import star.base.report.*;`
	`import star.battery.*;`
	`import star.circuit.*;`
	`import star.solidmesher.*;`
	`import star.flow.*;`
	`import star.meshing.*;`
	`import star.metrics.*;`
	`import star.energy.*;`

Changes to the Dars-CFD Reaction Model

In STAR-CCM+ v9.06, the *Allow Switch to ASCII* property has been removed from the Dars-CFD Reaction model. Remove all references to this property from your existing macros.

Previous Release	STAR-CCM+ v9.06
`DarsCfdGasReactionModel darsCfdGasReactionModel_0 = physicsContinuum_0.getModelManager().getModel(DarsCfdGasReactionModel.class);` **`darsCfdGasReactionModel_0.setModeSwitch(true);`**	`DarsCfdGasReactionModel darsCfdGasReactionModel_0 = physicsContinuum_0.getModelManager().getModel(DarsCfdGasReactionModel.class);`

Introduction of Fan Curve Scaling for the Fan Interface Model

In STAR-CCM+ v9.06, the method that you use to set the rotation rate in the fan interface has changed due to the addition of fan curve scaling. In previous versions of STAR-CCM+ there was no fan curve scaling for the fan interface. In v9.06, you define the fan curve scaling on the performance curve by setting two values: Operating Rotating Rate and Data Rotation Rate. Modify your existing macros to use the updated method for setting the rotation rate.

Previous Release	STAR-CCM+ v9.06

`Simulation simulation_0 = getActiveSimulation();`
 	
`BoundaryInterface boundaryInterface_0 = ((BoundaryInterface) simulation_0.getInterfaceManager() .getInterface("Block/Block 2"));`
	`Simulation simulation_0 = getActiveSimulation();`
	
	`DirectBoundaryInterface directBoundaryInterface_0 = ((DirectBoundaryInterface) simulation_0.getInterfaceManager() .getInterface("Block/Block 2"));`

Previous Release	STAR-CCM+ v9.06

`FanInterfaceSwirlProperties fanInterfaceSwirlProperties_0 = boundaryInterface_0.getValues().get(FanInterfaceSwirlProperties.class);`
 	
`fanInterfaceSwirlProperties_0.getRotationRate().setValue(50.0);`
	`FanCurveTable fanCurveTable_0 = directBoundaryInterface_0.getValues().get(FanCurveTable.class);`
	
	`fanCurveTable_0.getOperatingRotationRate().setValue(50.0);`
	`fanCurveTable_0.getDataRotationRate().setValue(50.0);`

New Method for Simple Shape Parts

Newly created simple shape parts are now associated with CAD objects. Rather than numbering faces sequentially from 1, use the ID numbers from the CAD surfaces.

Simple shape parts in earlier simulations are not associated with CAD objects. If you change any parameter related to the shape in an old simulation file (for example the radius of a sphere), STAR-CCM+ generates a CAD body for the part and then re-tessellates the part.

To perform the split-by-patch action, use one of the following functions to get the patch ID of a specific face and pass it to the `splitPartSurfaceByPatch` function as an argument:

- For a block part:
  - `getMinXFace()`
  - `getMinYFace()`
  - `getMinZFace()`
  - `getMaxXFace()`
  - `getMaxYFace()`
  - `getMaxZFace()`
- For cylinder/cone parts:
  - `getStartFace()` – patch corresponding to the face that is close to the start point
  - `getEndFace()` – patch corresponding to the face that is close to the end point
  - `getSideFace()` – patch corresponding to the cylindrical face

Examples of macro steps for a block part and a cylinder follow.

<u>Block Part</u>

The value used in this example is `minXFace`.

Previous Release	STAR-CCM+ v9.06
`SimpleBlockPart simpleBlockPart_0 = ((SimpleBlockPart) simulation_0.get(SimulationPartMa`	`SimpleBlockPart simpleBlockPart_0 = ((SimpleBlockPart) simulation_0.get(SimulationPartMa`

Previous Release	STAR-CCM+ v9.06

`nager.class).getPart("Block"));`
 	
`PartSurface partSurface_0 = ((PartSurface) simpleBlockPart_0.getPartSurfaceManager().getPartSurface("Block Surface"));`
 	
`simpleBlockPart_0.splitPartSurfaceByPatch(partSurface_0, new IntVector(new int[] {1}), "Block Surface");`
	`nager.class).getPart("Block"));`
	
	`PartSurface partSurface_0 = ((PartSurface) simpleBlockPart_0.getPartSurfaceManager().getPartSurface("Block Surface"));`
	
	`int minXFace = simpleBlockPart_0.getMinXFace();`
	
	`simpleBlockPart_0.splitPartSurfaceByPatch(partSurface_0, new IntVector(new int[] {minXFace}), "Block Surface");`

Cylinder Part

The value used in this example is `startFace`.

Previous Release	STAR-CCM+ v9.06

`SimpleCylinderPart simpleCylinderPart_0 = ((SimpleCylinderPart) simulation_0.get(SimulationPartManager.class).getPart("Cylinder"));`
 	
`PartSurface partSurface_0 = ((PartSurface) simpleCylinderPart_0.getPartSurfaceManager().getPartSurface("Cylinder Surface"));`
 	
`simpleCylinderPart_0.splitPartSurfaceByPatch(partSurface_0, new IntVector(new int[] {1}), "Cylinder Surface");`
	`SimpleCylinderPart simpleCylinderPart_0 = ((SimpleCylinderPart) simulation_0.get(SimulationPartManager.class).getPart("Cylinder"));`
	
	`PartSurface partSurface_0 = ((PartSurface) simpleCylinderPart_0.getPartSurfaceManager().getPartSurface("Cylinder Surface"));`
	
	`int startFace = simpleCylinderPart_0.getStartFace();`
	
	`simpleCylinderPart_0.splitPartSurfaceByPatch(partSurface_0, new IntVector(new int[] {startFace}), "Cylinder Surface");`

Macro API Changes 9.04

In STAR-CCM+ 9.04, the macro API changed for the Granular Pressure model, Eulerian multiphase modeling, co-simulation, 1D coupling, STAR-Cast, and the Ffowcs Williams-Hawkings models.

Changes to the Frictional and Granular Pressure Models

In STAR-CCM+ v9.04, the frictional model selection is done at the granular pressure model. Update your old macros as follows:

Previous Release	STAR-CCM+ v9.04

`//Activating the Granular Pressure Model`
`PhysicsContinuum physicsContinuum_0 =`
`((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Physics 1"));`
`physicsContinuum_0.enable(GranularPressureModel.class);`
`physicsContinuum_0.enable(GranularTemperatureProviderGroup.class);`
`physicsContinuum_0.enable(GranularTemperatureTransportModelProvider.class);`
	`//Activating the Granular Pressure Model`
	`PhysicsContinuum physicsContinuum_0 =`
	`((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Physics 1"));`
	`physicsContinuum_0.enable(GranularPressureModel.class);`
	`physicsContinuum_0.enable(GranularTemperatureTransportModelProvider.class);`

`//Setting frictional regime to Modified Johnson`
`GranularPressureModel granularPressureModel_0 = physicsContinuum_0.getModelManager().getModel(GranularPressureModel.class);`
`FrictionalTheoryProvider frictionalTheoryProvider_0 = granularPressureModel_0.getFrictionalTheory();`
`frictionalTheoryProvider_0.getFri`
	`//Setting frictional regime to Modified Johnson`
	`PhysicsContinuum physicsContinuum_0 =`
	`((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Physics 1"));`
	`GranularPressureModel granularPressureModel_0 = physicsContinuum_0.getModelManage`

Previous Release	STAR-CCM+ v9.04
`ctionalSolidPressureOption().setSelected(FrictionalSolidPressureOption.MOD_JOHNSON);`	`r().getModel(GranularPressureModel.class);` `granularPressureModel_0.getFrictionalSolidPressureOption().setSelected(FrictionalSolidPressureOption.MOD_JOHNSON);`

`//Setting frictional regime to Schaeffer GranularPressureModel`
`granularPressureModel_0 = physicsContinuum_0.getModelManager().getModel(GranularPressureModel.class);`
`FrictionalTheoryProvider frictionalTheoryProvider_0 = granularPressureModel_0.getFrictionalTheory();`
`frictionalTheoryProvider_0.getFrictionalSolidPressureOption().setSelected(FrictionalSolidPressureOption.SCHAEFFER);`
	`//Setting frictional regime to Schaeffer GranularPressureModel`
	`GranularPressureModel granularPressureModel_0 = physicsContinuum_0.getModelManager().getModel(GranularPressureModel.class);`
	`granularPressureModel_0.getFrictionalSolidPressureOption().setSelected(FrictionalSolidPressureOption.SCHAEFFER);`

`//Setting angle property on Schaeffer model`
`SchaefferFrictionalMethod schaefferFrictionalMethod_0 = frictionalTheoryProvider_0.getFrictionalMethod();`
`schaefferFrictionalMethod_0.getAngInternalFriction().setValue(26.0);`
	`//Setting angle property on Schaeffer model`
	`SchaefferFrictionalMethod schaefferFrictionalMethod_0 = granularPressureModel_0.getFrictionalMethod();`
	`schaefferFrictionalMethod_0.getAngInternalFriction().setValue(26.0);`

Changes to the Eulerian Multi Phase Interaction Length Scale and Interaction Area Density

In STAR-CCM+ v9.04, the different interaction length scale (ILS) models are consolidated into a single model. Similarly, the different interaction area density (IAD) models are consolidated into one model. The previous ILS and IAD models are now made available as ILS and IAD methods. The constant interaction length scale method is set as the default ILS method with a value of 0.001 m while the symmetric interaction area density method is set as the default IAD method.

Previous Release	STAR-CCM+ v9.04
`//Selecting Interaction Length Scale as Constant and setting its value to 1.0E-4`	`//Selecting Interaction Length Scale as Constant and setting its value to 1.0E-4`
`eulerianPhaseInteraction_0.enable(ConstantInteractionLengthScaleModel.class);`	`eulerianPhaseInteraction_0.enable(EulerianInteractionLengthScaleModel.class);`
`ConstantInteractionLengthScaleModel constantInteractionLengthScaleModel_0 =`	`EulerianInteractionLengthScaleModel eulerianInteractionLengthScaleModel_0 =`
`eulerianPhaseInteraction_0.getModelManager().getModel(ConstantInteractionLengthScaleModel.class);`	`eulerianPhaseInteraction_0.getModelManager().getModel(EulerianInteractionLengthScaleModel.class);`
`constantInteractionLengthScaleModel_0.getInteractionLengthScale().setValue(1.0E-4);`	`ConstantInteractionCoefficientMethod constantInteractionCoefficientMethod_0 =`
`//Selecting Symmetric Interaction Area Density`	`eulerianInteractionLengthScaleModel_0.getEulerianInteractionLengthScale().getMethod(ConstantInteractionCoefficientMethod.class);`
`eulerianPhaseInteraction_0.enable(SymmetricInteractionAreaDensityModel.class);`	`constantInteractionCoefficientMethod_0.getQuantity().setValue(1.0E-4);`
	`//Selecting Symmetric Interaction Area Density (default)`
	`eulerianPhaseInteraction_0.enabl`

Previous Release	STAR-CCM+ v9.04
	`e(InteractionAreaDensityModel.class);`
`//Using a Field Function as Interaction Length Scale`	`//Using a Field Function as Interaction Length Scale`
`eulerianPhaseInteraction_0.enable(ConstantInteractionLengthScaleModel.class);`	`eulerianPhaseInteraction_0.enable(EulerianInteractionLengthScaleModel.class);`
`ConstantInteractionLengthScaleModel constantInteractionLengthScaleModel_0 =`	`EulerianInteractionLengthScaleModel eulerianInteractionLengthScaleModel_0 =`
`phaseInteraction_0.getModelManager().getModel(ConstantInteractionLengthScaleModel.class);`	`phaseInteraction_0.getModelManager().getModel(EulerianInteractionLengthScaleModel.class);`
`phaseInteraction_0.disableModel(constantInteractionLengthScaleModel_0);`	`eulerianInteractionLengthScaleModel_0.getEulerianInteractionLengthScale().setMethod(FunctionInteractionCoefficientMethod.class);`
`phaseInteraction_0.enable(UserDefinedInteractionLengthScaleModel.class);`	`FunctionInteractionCoefficientMethod functionInteractionCoefficientMethod_0 =`
`UserDefinedInteractionLengthScaleModel userDefinedInteractionLengthScaleModel_0 =`	`eulerianInteractionLengthScaleModel_0.getEulerianInteractionLength`
`phaseInteraction_0.getModelManage`	

Previous Release	STAR-CCM+ v9.04

`r().getModel(UserDefinedInteractionLengthScaleModel.class);`
`InteractionLengthScaleProfile interactionLengthScaleProfile_0 =`
`userDefinedInteractionLengthScaleModel_0.getInteractionLengthScaleProfile();`
`UserFieldFunction userFieldFunction_0 =`
`((UserFieldFunction) simulation_0.getFieldFunctionManager().getFunction(desiredFunctionName));`
`interactionLengthScaleProfile_0.getMethod(FunctionScalarProfileMethod.class).setFieldFunction(userFieldFunction_0);`
	`Scale().getMethod(FunctionInteractionCoefficientMethod.class);`
	`UserFieldFunction userFieldFunction_0 =`
	`((UserFieldFunction) simulation_0.getFieldFunctionManager().getFunction(desiredFunctionName));`
	`functionInteractionCoefficientMethod_0.setFieldFunction(userFieldFunction_0);`

`//Selecting Spherical Particle Interaction Area Density`
`phaseInteraction_0.enable(SymmetricInteractionAreaDensityModel.class);`
`SymmetricInteractionAreaDensityModel`
	`//Selecting Spherical Particle Interaction Area Density`
	`phaseInteraction_0.enable(InteractionAreaDensityModel.class);`
	`InteractionAreaDensityModel interactionAreaDensityModel_0 =`
	`phaseInteraction_0.getModelManage`

Previous Release	STAR-CCM+ v9.04
`symmetricInteractionAreaDensityModel_0 =`	`r().getModel(InteractionAreaDensityModel.class);`
`phaseInteraction_0.getModelManager().getModel(SymmetricInteractionAreaDensityModel.class);`	`interactionAreaDensityModel_0.getInteractionAreaDensity().setMethod(SphericalParticleInteractionAreaDensityMethod.class);`
`phaseInteraction_0.disableModel(symmetricInteractionAreaDensityModel_0);`	
`phaseInteraction_0.enable(SphericalParticleInteractionAreaDensityModel.class);`	

Changes to the Eulerian Multi Phase Flow Profiles and Flow Options

In STAR-CCM+ v9.04, EMP velocity and volume-fraction profiles that are used to set initial conditions, boundary conditions, mass, and momentum source have been migrated to each individual phase.

Previous Release	STAR-CCM+ v9.04
`//Setting the Initial conditions for velocity and volume fraction`	`//Setting the Initial conditions for velocity and volume fraction`
`MultiPhaseVelocityProfile multiPhaseVelocityProfile_0 =`	`VelocityProfile velocityProfile_0 =`
`physicsContinuum_0.getInitialConditions().get(MultiPhaseVelocityProfile.class);`	`eulerianPhase_0.get(ContinuumInitialConditionManager.class).get(VelocityProfile.class);`
`VectorProfile vectorProfile_0 =`	`velocityProfile_0.getMethod(ConstantVectorProfileMethod.class).get`
`multiPhaseVelocityProfile_0.getMe`	

Previous Release	STAR-CCM+ v9.04
`thod(CompositeArrayVectorProfileMethod.class).getProfile(0);`	`Quantity().setComponents(2.02, 0.0, 0.0);`
`vectorProfile_0.getMethod(ConstantVectorProfileMethod.class).getQuantity().setComponents(2.02, 0.0, 0.0);`	`VelocityProfile velocityProfile_1 =`
`VectorProfile vectorProfile_1 =`	`eulerianPhase_1.get(ContinuumInitialConditionManager.class).get(VelocityProfile.class);`
`multiPhaseVelocityProfile_0.getMethod(CompositeArrayVectorProfileMethod.class).getProfile(1);`	`velocityProfile_1.getMethod(ConstantVectorProfileMethod.class).getQuantity().setComponents(2.0, 0.0, 0.0);`
`vectorProfile_1.getMethod(ConstantVectorProfileMethod.class).getQuantity().setComponents(2.0, 0.0, 0.0);`	`ScalarVolumeFractionProfile scalarVolumeFractionProfile_0 =`
`VolumeFractionProfile volumeFractionProfile_0 =`	`eulerianPhase_0.get(ContinuumInitialConditionManager.class).get(ScalarVolumeFractionProfile.class);`
`physicsContinuum_0.getInitialConditions().get(VolumeFractionProfile.class);`	`scalarVolumeFractionProfile_0.getMethod(ConstantScalarProfileMethod.class).getQuantity().setValue(0.999);`
`volumeFractionProfile_0.getMethod(ConstantArrayProfileMethod.class).getQuantity().setArray(new DoubleVector(new double[]`	`ScalarVolumeFractionProfile scalarVolumeFractionProfile_1 =`
`{0.999, 0.0010}));`	`eulerianPhase_1.get(ContinuumInitialConditionManager.class).get(ScalarVolumeFractionProfile.class);`
	`scalarVolumeFractionProfile_1.getMethod(ConstantScalarProfileMetho`

Previous Release	STAR-CCM+ v9.04
	`d.class).getQuantity().setValue(0 .0010);`

`//Setting boundary conditions for two-phases (Water and Steam)`
`MultiPhaseVelocityMagnitudeProfil e multiPhaseVelocityMagnitudeProfil e_0 =`
`boundary_1.getValues().get(MultiP haseVelocityMagnitudeProfile.clas s);`
`multiPhaseVelocityMagnitudeProfil e_0.getMethod(ConstantArrayProfil eMethod.class).getQuantity().setA rray(new DoubleVector(new double[] {2.0, 2.0}));`
`VolumeFractionProfile volumeFractionProfile_1 =`
`boundary_1.getValues().get(Volume FractionProfile.class);`
`volumeFractionProfile_1.getMetho d(ConstantArrayProfileMethod.clas s).getQuantity().setArray(new DoubleVector(new double[] {0.999, 0.0010}));`
	`//Setting boundary conditions for two-phases (Water and Steam)`
	`PhaseConditions phaseConditions_0 =`
	`((PhaseConditions) boundary_1.get(PhaseConditionsMan ager.class).getPhaseConditions("W ater"));`
	`PhaseConditions phaseConditions_1 =`
	`((PhaseConditions) boundary_1.get(PhaseConditionsMan ager.class).getPhaseConditions("S team"));`
	`VelocityMagnitudeProfile velocityMagnitudeProfile_0 =`
	`phaseConditions_0.getPhaseValueMa nager().get(VelocityMagnitudeProf ile.class);`
	`velocityMagnitudeProfile_0.getMet hod(ConstantScalarProfileMethod.c`

Previous Release	STAR-CCM+ v9.04
	`lass).getQuantity().setValue(2.0);`
	`VelocityMagnitudeProfile velocityMagnitudeProfile_1 =`
	`phaseConditions_1.getPhaseValueManager().get(VelocityMagnitudeProfile.class);`
	`velocityMagnitudeProfile_1.getMethod(ConstantScalarProfileMethod.class).getQuantity().setValue(2.0);`
	`ScalarVolumeFractionProfile scalarVolumeFractionProfile_2 =`
	`phaseConditions_0.getPhaseValueManager().get(ScalarVolumeFractionProfile.class);`
	`scalarVolumeFractionProfile_2.getMethod(ConstantScalarProfileMethod.class).getQuantity().setValue(0.999);`
	`ScalarVolumeFractionProfile scalarVolumeFractionProfile_3 =`
	`phaseConditions_1.getPhaseValueManager().get(ScalarVolumeFractionProfile.class);`
	`scalarVolumeFractionProfile_3.getMethod(ConstantScalarProfileMetho`

Previous Release	STAR-CCM+ v9.04
	`d.class).getQuantity().setValue(0.0010);`

`//Turning on mass and momentum user-source options`
`MultiPhaseMassUserSourceOption multiPhaseMassUserSourceOption_0 =`
`region_0.getConditions().get(MultiPhaseMassUserSourceOption.class);`
`multiPhaseMassUserSourceOption_0.setUserSourceOption(true);`
`MultiPhaseMomentumUserSourceOption multiPhaseMomentumUserSourceOption_1 =`
`region_0.getConditions().get(MultiPhaseMomentumUserSourceOption.class);`
`multiPhaseMomentumUserSourceOption_1.setUserSourceOption(true);`
	`//Turning on mass and momentum user-source options`
	`MassUserSourceOption massUserSourceOption_0 =`
	`phaseConditions_0.getPhaseConditionManager().get(MassUserSourceOption.class);`
	`massUserSourceOption_0.setUserSourceOption(true);`
	`MassUserSourceOption massUserSourceOption_1 =`
	`phaseConditions_1.getPhaseConditionManager().get(MassUserSourceOption.class);`
	`massUserSourceOption_1.setUserSourceOption(true);`
	`phaseConditions_0.getPhaseConditionManager().get(MomentumUserSourceOption.class).setSelected(MomentumUserSourceOption.SPECIFIED);`
	`phaseConditions_1.getPhaseConditionManager().get(MomentumUserSourceOption.class).setSelected(MomentumUserSourceOption.SPECIFIED);`

Previous Release	STAR-CCM+ v9.04
`//Setting the field-functions as mass and momentum user source`	`//Setting the field-functions as mass and momentum user source`
`MultiPhaseMassUserSource multiPhaseMassUserSource_0 =`	`MassUserSource massUserSource_0 =`
`region_0.getValues().get(MultiPhaseMassUserSource.class);`	`phaseConditions_0.getPhaseValueManager().get(MassUserSource.class);`
`multiPhaseMassUserSource_0.setMethod(CompositeArrayProfileMethod.class);`	`massUserSource_0.setMethod(FunctionScalarProfileMethod.class);`
`MultiPhaseMassUserSourceVolumeFractionDerivative multiPhaseMassUserSourceVolumeFractionDerivative_0 =`	`MassUserSource massUserSource_1 =`
`region_0.getValues().get(MultiPhaseMassUserSourceVolumeFractionDerivative.class);`	`phaseConditions_1.getPhaseValueManager().get(MassUserSource.class);`
`multiPhaseMassUserSourceVolumeFractionDerivative_0.setMethod(CompositeArrayProfileMethod.class);`	`massUserSource_1.setMethod(FunctionScalarProfileMethod.class);`
	`MassUserSourceVolumeFractionDerivative massUserSourceVolumeFractionDerivative_0 =`
	`phaseConditions_0.getPhaseValueManager().get(MassUserSourceVolumeFractionDerivative.class);`
	`massUserSourceVolumeFractionDerivative_0.setMethod(FunctionScalarProfileMethod.class);`

Similarly, replace the old class names that are listed in the following table with their updated equivalents:

Previous Release	STAR-CCM+ v9.04
`MultiPhaseVelocityProfile`	`VelocityProfile`
`VolumeFractionProfile`	`ScalarVolumeFractionProfile`
`MultiPhaseMassUserSourceOption`	`MassUserSourceOption`
`MultiPhaseMassUserSource`	`MassUserSource`
`MultiPhaseMomentumUserSourceOption`	`MomentumUserSourceOption`
`MultiPhaseMomentumUserSource`	`MomentumUserSource`
`MultiPhaseMomentumUserSourceJacobianVector`	`MomentumUserSourceJacobianVector`
`MultiPhaseMassUserSourceVolumeFractionDerivative`	`MassUserSourceVolumeFractionDerivative`

Changes to the STAR-CCM+ Co-Simulation API due to the Introduction of the Implicit Model and Solver

In STAR-CCM+ v9.04, the `StarccmplusCoSimulationImplicitUnsteadyModel` model and `StarccmplusCoSimulationImplicitUnsteadySolver` solver are introduced for STAR-CCM+ co-simulation.

To accommodate these changes, the previous `StarccmplusCoSimulationModel` model has been renamed to `StarccmplusCoSimulationExplicitUnsteadyModel` for transient simulations and `StarccmplusCoSimulationSteadyModel` for steady simulations.

For transient simulations, the previous `StarccmplusCoSimulationUnsteadySolver` solver has been renamed to `StarccmplusCoSimulationExplicitUnsteadySolver`. If your simulation contains an older model or solver, these will be updated to use the renamed variants.

Previous Release	STAR-CCM+ v9.04
`PhysicsContinuum physicsContinuum_0 =`	`PhysicsContinuum physicsContinuum_0 =`
`((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Physics 1"));`	`((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Physics 1"));`
`physicsContinuum_0.enable(CoSimulationModel.class);`	`physicsContinuum_0.enable(CoSimulationModel.class);`
`physicsContinuum_0.enable(StarccmplusCoSimulationModel.class);`	`physicsContinuum_0.enable(StarccmplusCoSimulationExplicitUnsteadyModel.class);`
`StarccmplusCoSimulationUnsteadySolver starccmplusCoSimulationUnsteadySolver_0 = ((StarccmplusCoSimulationUnsteadySolver) simulation_0.getSolverManager().getSolver(StarccmplusCoSimulationUnsteadySolver.class));`	`StarccmplusCoSimulationExplicitUnsteadySolver starccmplusCoSimulationExplicitUnsteadySolver_0 =`
`((ConstantNegotiationMethod) starccmplusCoSimulationUnsteadySolver_0.getNegotiationMethodManager().getMethod()).getConstantTimeStep().setValue(10.0);`	`((StarccmplusCoSimulationExplicitUnsteadySolver) simulation_0.getSolverManager().getSolver(StarccmplusCoSimulationExplicitUnsteadySolver.class));`
	`((ConstantNegotiationMethod) starccmplusCoSimulationExplicitUnsteadySolver_0.getNegotiationMethodManager().getMethod()).getConstantTimeStep().setValue(10.0);`
`PhysicsContinuum physicsContinuum_0 =`	`PhysicsContinuum physicsContinuum_0 =`
`((PhysicsContinuum)`	`((PhysicsContinuum)`

Previous Release	STAR-CCM+ v9.04
`simulation_0.getContinuumManager().getContinuum("Physics 1"));`	`simulation_0.getContinuumManager().getContinuum("Physics 1"));`
`physicsContinuum_0.enable(CoSimulationModel.class);`	`physicsContinuum_0.enable(CoSimulationModel.class);`
`physicsContinuum_0.enable(StarccmplusCoSimulationModel.class);`	`physicsContinuum_0.enable(StarccmplusCoSimulationSteadyModel.class);`
`StarccmplusCoSimulationSteadySolver starccmplusCoSimulationSteadySolver_0 =`	`StarccmplusCoSimulationSteadySolver starccmplusCoSimulationSteadySolver_0 =`
`((StarccmplusCoSimulationSteadySolver) simulation_0.getSolverManager().getSolver(StarccmplusCoSimulationSteadySolver.class));`	`((StarccmplusCoSimulationSteadySolver) simulation_0.getSolverManager().getSolver(StarccmplusCoSimulationSteadySolver.class));`
`NumIterationsPerExchange numIterationsPerExchange_0 =`	`NumIterationsPerExchange numIterationsPerExchange_0 =`
`starccmplusCoSimulationSteadySolver_0.getNumIterationsPerExchange();`	`starccmplusCoSimulationSteadySolver_0.getNumIterationsPerExchange();`
`numIterationsPerExchange_0.setNumIterationsPerExchange(10);`	`numIterationsPerExchange_0.setNumIterationsPerExchange(10);`

Changes to How Zones are Assigned to Boundaries in 1D Coupling

In STAR-CCM+ v9.04, zones are set using an object selector dialog. In previous versions, you selected boundaries for the zones using a drop-down menu. Any macros that refer to the old selection method for the boundaries must be updated as follows:

Previous Release	STAR-CCM+ v9.04
`OlgaZone olgaZone_1 =`	`OlgaZone olgaZone_1 =`
`((OlgaZone) olga5Node_0.getOlgaZoneManager().getObject("Zone 1"));`	`((OlgaZone) olga5Node_0.getOlgaZoneManager().getObject("Zone 1"));`
`Boundary boundary_0 =`	`Boundary boundary_0 =`
`region_0.getBoundaryManager().getBoundary("Inlet");`	`region_0.getBoundaryManager().getBoundary("Inlet”);`
`olgaZone_1.setCcmBoundary(boundary_0);`	`olgaZone_1.getCcmBoundaryPartGroup().setObjects(boundary_0);`

Changes to the Dendrite Arm Spacing models in STAR-Cast

In STAR-CCM+ v9.04, the `Primary Dendrite Arm Spacing` material property is transformed into the `Primary Dendrite Arm Spacing` model. Similarly, the Secondary Dendrite Arm Spacing material property is transformed to the `Secondary Dendrite Arm Spacing`/`Secondary Dendrite Arm Spacing (LST)`/`Secondary Dendrite Arm Spacing (CR)` models. Any old macros that contain these functions must be updated as follows:

Previous Release	STAR-CCM+ v9.04
`package macro;`	`package macro;`
`import java.util.*;`	`import java.util.*;`
`import star.common.*;`	`import star.material.*;`
`import star.casting.*;`	`import star.common.*;`
`public class castingModel extends StarMacro {`	`import star.casting.*;`
`public void execute() {`	`public class castingModel extends StarMacro {`
`execute0();`	`public void execute() {`
`}`	`execute0();`
`private void execute0() {`	`}`
`Simulation simulation_0 =`	`private void execute0() {`
`getActiveSimulation();`	`Simulation simulation_0 =`
`PhysicsContinuum physicsContinuum_0 =`	`getActiveSimulation();`
`((PhysicsContinuum)`	`PhysicsContinuum physicsContinuum_0 =`
	`((PhysicsContinuum)`

Previous Release	STAR-CCM+ v9.04
`simulation_0.getContinuumManager().getContinuum("Physics 1"));`	`simulation_0.getContinuumManager().getContinuum("Physics 1"));`
`EulerianMultiPhaseModel eulerianMultiPhaseModel_0 =`	`EulerianMultiPhaseModel eulerianMultiPhaseModel_0 =`
`physicsContinuum_0.getModelManager().getModel(EulerianMultiPhaseModel.class);`	`physicsContinuum_0.getModelManager().getModel(EulerianMultiPhaseModel.class);`
`EulerianPhase eulerianPhase_0 =`	`EulerianPhase eulerianPhase_0 =`
`((EulerianPhase) eulerianMultiPhaseModel_0.getPhaseManager().getPhase("Phase 1"));`	`((EulerianPhase) eulerianMultiPhaseModel_0.getPhaseManager().getPhase("Phase 1"));`
`eulerianPhase_0.enable(CriteriaFunctionModel.class);`	`eulerianPhase_0.enable(CriteriaFunctionModel.class);`
`}`	`eulerianPhase_0.enable(PrimaryDendriteArmSpacingModel.class);`
`}`	`eulerianPhase_0.enable(SecondaryDendriteArmSpacingModel.class);`
	`eulerianPhase_0.enable(SecondaryDendriteArmSpacingLSTModel.class);`
	`}`
	`}`

Changes to the Ffowcs Williams-Hawkings Models

The *Far-field Density* and *Far-field Sound Speed* properties have moved from being properties of the *On-the-Fly FW-H* model to the *FW-H Receivers* manager. As a result, any old macros that sets these properties explicitly, rather than by default, will fail to compile.

Previous Release	STAR-CCM+ v9.04

`FwhModel fwhModel_0 =`
`physicsContinuum_0.getModelManager().getModel(FwhModel.class);`
`fwhModel_0.getRhoFarField().setValue(1.225);`
`fwhModel_0.getSoundSpeedFarField().setValue(340.0);`
	`simulation_0.get(ReceiverManager.class).getFarFieldDensity().setValue(1.225);`
	`simulation_0.get(ReceiverManager.class).getFarFieldSoundSpeed().setValue(340.0);`

The *Acoustic Correlation Length* has moved from being a property located on the *Reference Values* node, for two-dimensional simulations when *On-The-Fly FW-H Models* and *Post FW-H Models* are enabled, to the *FW-H Receivers* manager (for *On-the-Fly FW-H Model*) and *Post FW-H Receivers* manager (for *Post FW-H Model*) respectively. As a result, any old macros that sets this property explicitly, rather than by default, will fail to compile.

Previous Release	STAR-CCM+ v9.04
`continuum.getReferenceValues().get(AcousticCorrelationLengthRV.class).setValue(0.19);`	`simulation_0.get(ReceiverManager.class).getAcousticCorrelationLength().setValue(0.19);` `simulation_0.get(FwhPostProcessingReceiverManager.class).getAcousticCorrelationLength().setValue(0.19);`

The *Far-field Density* and *Far-field Sound Speed* properties have moved from being properties of the *Post FW-H* model to the *Post FW-H Receivers* manager. As a result, any old macros that sets these properties explicitly, rather than by default, will fail to compile.

Previous Release	STAR-CCM+ v9.04
`FwhCollectTrnPostProcessingModel`	`simulation_0.get(FwhPostProcessingReceiverManager.class).getFarFieldDensity().setValue(1.225);`
`fwhCollectTrnPostProcessingModel_0 =`	`simulation_0.get(FwhPostProcessingReceiverManager.class).getFarFieldSoundSpeed().setValue(340.0)`
`physicsContinuum_0.getModelManager().getModel(FwhCollectTrnPostProcessingModel.class);`	
`fwhCollectTrnPostProcessingModel_0.getRhoFarField().setValue(1.225);`	
`fwhCollectTrnPostProcessingModel_0.getSoundSpeedFarField().setValue(340.0);`	

Macro API Changes 9.02

In STAR-CCM+ 9.02, the macro API changed for OLGA coupling, the Phase Coupled Isothermal model, the Haugen Erosion Correlation, part surface contacts, the plane tool, the Molecular Diffusivity model, DFBI, and simple shape part creation.

OLGA 5

In STAR-CCM+ v9.02, you can couple with OLGA 5 and OLGA 7 (for Linux versions of STAR-CCM+ only). The name of OLGA objects in STAR-CCM+ have changed due to the addition of OLGA 7. In macros for OLGA 5, replace all instances of `OlgaNode` with `Olga5Node`. Note that macros written for OLGA 5 cannot be used with OLGA 7, as the coupling process is different.

Previous Release	STAR-CCM+ v9.02
`Simulation simulation_0 =`	`Simulation simulation_0 =`
`getActiveSimulation();`	`getActiveSimulation();`
`OlgaNode olgaNode_0 =`	`Olga5Node olga5Node_0 =`
`((OlgaNode) simulation_0.get(CoSimulationManager.class).getObject("OLGA-5 1"));`	`((Olga5Node) simulation_0.get(CoSimulationManager.class).getObject("OLGA-5 1"));`
`olgaNode_0.setOlgaPort(12340);`	`olga5Node_0.setOlgaPort(12340);`
`olgaNode_0.setOlgaHostName("olga-server");`	`olga5Node_0.setOlgaHostName("olga-server");`
`olgaNode_0.setOlgaInputFile("");`	`olga5Node_0.setOlgaInputFile("");`
`olgaNode_0.setOlgaInputFile("C:\\EXAMPLE\\OLGA\\");`	`olga5Node_0.setOlgaInputFile("C:\\EXAMPLE\\OLGA\\");`
`olgaNode_0.setOlgaInputFile("C:\\EXAMPLE\\OLGA\\OLGAInput");`	`olga5Node_0.setOlgaInputFile("C:\\EXAMPLE\\OLGA\\OLGAInput");`
`olgaNode_0.setOlgaInputFile("C:\\Work\\OLGAInput");`	`olga5Node_0.setOlgaInputFile("C:\\Work\\OLGAInput");`
`olgaNode_0.setOlgaPropsFile("3phase.tab");`	`olga5Node_0.setOlgaPropsFile("3phase.tab");`

Change to the Phase Coupled Isothermal Model

In STAR-CCM+ v9.02, the old Phase Coupled Fluid Isothermal Model has been removed and is replaced by something more focussed. The Segregated Fluid Isothermal option is used only to support the equation of

state models that require temperature, such as the Ideal Gas density model, when you are not modelling heat transfer. This option becomes available under each phase when such equation of state models are selected for that phase only if Phase Coupled Heat Transfer is turned off at the phase level. With this new structure, all phases do not need the `SegregatedFluidIsothermalModel` when only one phase may require it.

Previous Release	STAR-CCM+ v9.02
`PhysicsContinuum physicsContinuum_0 =`	`PhysicsContinuum physicsContinuum_0 =`
`((PhysicsContinuum) simulation_0.getContinuumManager( ).getContinuum("Physics 1 2D"));`	`((PhysicsContinuum) simulation_0.getContinuumManager( ).getContinuum("Physics 1 2D"));`
`physicsContinuum_0.enable(SteadyModel.class);`	`physicsContinuum_0.enable(SteadyModel.class);`
`physicsContinuum_0.enable(MultiPhaseMaterialModel.class);`	`physicsContinuum_0.enable(MultiPhaseMaterialModel.class);`
`physicsContinuum_0.enable(SegregatedMultiPhaseModel.class);`	`physicsContinuum_0.enable(SegregatedMultiPhaseModel.class);`
`physicsContinuum_0.enable(MultiPhaseLaminarModel.class);`	`physicsContinuum_0.enable(MultiPhaseLaminarModel.class);`
`physicsContinuum_0.enable(GravityModel.class);`	`physicsContinuum_0.enable(GravityModel.class);`
`physicsContinuum_0.enable(PhaseCoupledFluidIsothermalModel.class);`	

Previous Release	STAR-CCM+ v9.02
`EulerianPhase eulerianPhase_1 =`	`EulerianPhase eulerianPhase_1 =`
`eulerianMultiPhaseModel_0.createPhase();`	`eulerianMultiPhaseModel_0.createPhase();`
`eulerianPhase_1.setPresentationName("gas");`	`eulerianPhase_1.setPresentationName("gas");`
`eulerianPhase_1.enable(SinglePhaseGasModel.class);`	`eulerianPhase_1.enable(SinglePhaseGasModel.class);`
`eulerianPhase_1.enable(IdealGasModel.class);`	`eulerianPhase_1.enable(IdealGasModel.class);`
`SinglePhaseGasModel singlePhaseGasModel_0 =`	`eulerianPhase_1.enable(SegregatedFluidIsothermalModel.class);`
`eulerianPhase_1.getModelManager().getModel(SinglePhaseGasModel.class);`	`SinglePhaseGasModel singlePhaseGasModel_0 =`
`SinglePhaseGas singlePhaseGas_0 =`	`eulerianPhase_1.getModelManager().getModel(SinglePhaseGasModel.class);`
`((SinglePhaseGas) singlePhaseGasModel_0.getMaterial());`	`SinglePhaseGas singlePhaseGas_0 =`
`ConstantMaterialPropertyMethod constantMaterialPropertyMethod_2 =`	`((SinglePhaseGas) singlePhaseGasModel_0.getMaterial());`
`((ConstantMaterialPropertyMethod) singlePhaseGas_0.getMaterialProp`	`ConstantMaterialPropertyMethod`

Previous Release	STAR-CCM+ v9.02
`rties().getMaterialProperty(DynamicViscosityProperty.class).getMethod());`	`constantMaterialPropertyMethod_2 =`
`constantMaterialPropertyMethod_2.getQuantity().setValue(2.0E-5);`	`((ConstantMaterialPropertyMethod) singlePhaseGas_0.getMaterialProperties().getMaterialProperty(DynamicViscosityProperty.class).getMethod());`
`ConstantMaterialPropertyMethod constantMaterialPropertyMethod_3 =`	`constantMaterialPropertyMethod_2.getQuantity().setValue(2.0E-5);`
`((ConstantMaterialPropertyMethod) singlePhaseGas_0.getMaterialProperties().getMaterialProperty(MolecularWeightProperty.class).getMethod());`	`ConstantMaterialPropertyMethod constantMaterialPropertyMethod_3 =`
`constantMaterialPropertyMethod_3.getQuantity().setValue(24.95);`	`((ConstantMaterialPropertyMethod) singlePhaseGas_0.getMaterialProperties().getMaterialProperty(MolecularWeightProperty.class).getMethod());`
	`constantMaterialPropertyMethod_3.getQuantity().setValue(24.95);`

Name Change for the Haugen Erosion Correlation

In STAR-CCM+ v9.02, the `HaugenCorrelation` class is renamed to `DNVCorrelation`. Any macros you have that contain this erosion model will require updating.

Previous Release	STAR-CCM+ v9.02
`ErosionRatio er =`	`ErosionRatio er =`
`defaultBoundaryConditions_0.getValues().get(ErosionRatio.class);`	`defaultBoundaryConditions_0.getValues().get(ErosionRatio.class);`
`er.setMethod(HaugenCorrelation.class);`	`er.setMethod(DNVCorrelation.class);`
`er.getMethod(HaugenCorrelation.class).getReferenceVelocity().setValue(1.2);`	`er.getMethod(DNVCorrelation.class).getReferenceVelocity().setValue(1.2);`
`er.getMethod(HaugenCorrelation.class).setVelocityExponent(2.4);`	`er.getMethod(DNVCorrelation.class).setVelocityExponent(2.4);`
`er.getMethod(HaugenCorrelation.class).getMaterialCoefficient().setValue(2.3E-9);`	`er.getMethod(DNVCorrelation.class).getMaterialCoefficient().setValue(2.3E-9);`
`er.getMethod(HaugenCorrelation.class).getAngleFunction().setCoefficients(new DoubleVector(new double[] {0.0594, -0.00105707, 5.18692E-6}));`	`Polynomial poly = erosionRatio_0.getMethod(DNVCorrelation.class).getAngleFunction();` `poly.setCoefficients(new DoubleVector(new double[] {0.062, -0.00105707, 5.18692E-6}));`

Change in Part Surface Contacts

In STAR-CCM+ v9.02, the macro API for part surface contacts were not journaled with sufficient information to identify the correct contact. Update your macros as follows:

Previous Release	STAR-CCM+ v9.02
`PartContact partContact_1 =`	`PartContact partContact_1 =`
`simulation_0.get(PartContactManager.class).getPartContact(simpleBlockPart_1,simpleBlockPart_0);`	`simulation_0.get(PartContactManager.class).getPartContact(simpleBlockPart_1,simpleBlockPart_0);`
`InPlacePartSurfaceContact inPlacePartSurfaceContact_0 =`	`PartSurface partSurface_0 =`
`((InPlacePartSurfaceContact) partContact_1.get(PartSurfaceContactManager.class).getObject("In Place Contact"));`	`((PartSurface) simpleBlockPart_0.getPartSurfaceManager().getPartSurface("PartSurface_Body 1_Body 2"));`
`partContact_1.get(PartSurfaceContactManager.class).remove(inPlacePartSurfaceContact_0);`	`PartSurface partSurface_1 =`
	`((PartSurface) simpleBlockPart_1.getPartSurfaceManager().getPartSurface("PartSurface_Body 2_Body 1"));`
	`InPlacePartSurfaceContact inPlacePartSurfaceContact_0 =`
	`((InPlacePartSurfaceContact) partContact_1.get(PartSurfaceContactManager.class).getPartSurfaceContact(partSurface_0,partSurface_1));`
	`partContact_1.get(PartSurfaceContactManager.class).remove(inPlacePartSurfaceContact_0);`

Changes to the Vector and Point Input for Planes

In STAR-CCM+ v9.02, a new unified plane tool panel is introduced to the Surface Repair tool, so as to consolidate and simplify the options required to define a plane.

Previous Release	STAR-CCM+ v9.02
`surfaceMeshWidgetRepairOptions_0.setProjectVerticesType(0);`	`surfaceMeshWidgetRepairOptions_0.setProjectVerticesType(0);`
`surfaceMeshWidgetRepairOptions_0.setProjectPlaneType(0);`	`surfaceMeshWidgetRepairOptions_0.setPlaneNormalType(0);`
`surfaceMeshWidgetRepairOptions_0.setPlaneNormalType(0);`	`surfaceMeshWidgetRepairOptions_0.setProjectionPlaneOrigin(coordinate_0);`
`surfaceMeshWidgetRepairOptions_0.setProjectionPlaneOrigin(coordinate_0);`	`surfaceMeshWidgetRepairOptions_0.setUseOriginForPlanePoint(false);`
`surfaceMeshWidgetRepairOptions_0.setProjectionPlaneNormal(new DoubleVector(new double[] {0.0, 0.0, 1.0}));`	`surfaceMeshWidgetRepairOptions_0.setProjectionPlaneNormal(new DoubleVector(new double[] {0.0, 0.0, 1.0}));`
`surfaceMeshWidgetRepairController_0.projectSelectedVertices();`	`surfaceMeshWidgetRepairController_0.projectSelectedVertices();`

Previous Release	STAR-CCM+ v9.02
`surfaceMeshWidgetRepairOptions_0.setProjectVerticesType(0);`	`surfaceMeshWidgetRepairOptions_0.setProjectVerticesType(0);`
`surfaceMeshWidgetRepairOptions_0.setProjectPlaneType(0);`	`surfaceMeshWidgetRepairOptions_0.setPlaneNormalType(1);`
`surfaceMeshWidgetRepairOptions_0.setPlaneNormalType(1);`	`surfaceMeshWidgetRepairOptions_0.setUseOriginForPlanePoint(false);`
`surfaceMeshWidgetRepairOptions_0.setProjectionPlaneOrigin(coordinate_0);`	`surfaceMeshWidgetRepairOptions_0.setProjectionPlaneOrigin(coordinate_0);`
`surfaceMeshWidgetRepairOptions_0.setNormalSpecificationPoint1(coordinate_1);`	`surfaceMeshWidgetRepairOptions_0.setProjectionPlanePoint1(coordinate_1);`
`surfaceMeshWidgetRepairOptions_0.setNormalSpecificationPoint2(coordinate_2);`	`surfaceMeshWidgetRepairOptions_0.setProjectionPlanePoint2(coordinate_2);`
`surfaceMeshWidgetRepairOptions_0.setNormalSpecificationPoint3(coordinate_3);`	`surfaceMeshWidgetRepairOptions_0.setProjectionPlanePoint3(coordinate_3);`
`surfaceMeshWidgetRepairController_0.projectSelectedVertices();`	`surfaceMeshWidgetRepairController_0.projectSelectedVertices();`

Changes to the Molecular Diffusivity Model

In STAR-CCM+ v9.02, the UserMultiComponentDiffusivityMethod.class has been renamed to ProfileMolecularDiffusivityMethod.class and the MultiComponentKineticTheoryDiffusivityMethod.class been renamed to KineticTheoryDiffusivityMethod.class.

Previous Release	STAR-CCM+ v9.02
`PhysicsContinuum physicsContinuum_0 =`	`PhysicsContinuum physicsContinuum_0 =`
`((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Physics 1"));`	`((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Physics 1"));`
`MultiComponentGasModel multiComponentGasModel_0 =`	`MultiComponentGasModel multiComponentGasModel_0 =`
`physicsContinuum_0.getModelManager().getModel(MultiComponentGasModel.class);`	`physicsContinuum_0.getModelManager().getModel(MultiComponentGasModel.class);`
`GasMixture gasMixture_0 =`	`GasMixture gasMixture_0 =`
`((GasMixture) multiComponentGasModel_0.getMixture());`	`((GasMixture) multiComponentGasModel_0.getMixture());`
`gasMixture_0.getMaterialProperties().getMaterialProperty(MolecularDiffusivityProperty.class).setMethod(UserMultiComponentDiffusivityMethod.class);`	`gasMixture_0.getMaterialProperties().getMaterialProperty(MolecularDiffusivityProperty.class).setMethod(ProfileMolecularDiffusivityMethod.class);`
`PhysicsContinuum physicsContinuum_0 =`	`PhysicsContinuum physicsContinuum_0 =`
`((PhysicsContinuum)`	`((PhysicsContinuum)`

Previous Release	STAR-CCM+ v9.02
`simulation_0.getContinuumManager().getContinuum("Physics 1"));`	`simulation_0.getContinuumManager().getContinuum("Physics 1"));`
`MultiComponentGasModel multiComponentGasModel_0 =`	`MultiComponentGasModel multiComponentGasModel_0 =`
`physicsContinuum_0.getModelManager().getModel(MultiComponentGasModel.class);`	`physicsContinuum_0.getModelManager().getModel(MultiComponentGasModel.class);`
`GasMixture gasMixture_0 =`	`GasMixture gasMixture_0 =`
`((GasMixture) multiComponentGasModel_0.getMixture());`	`((GasMixture) multiComponentGasModel_0.getMixture());`
`gasMixture_0.getMaterialProperties().getMaterialProperty(MolecularDiffusivityProperty.class).setMethod(MultiComponentKineticTheoryDiffusivityMethod.class);`	`gasMixture_0.getMaterialProperties().getMaterialProperty(MolecularDiffusivityProperty.class).setMethod(KineticTheoryDiffusivityMethod.class);`

In addition to the changes in class names, macros that previously used `ComponentMolecularDiffusivityProperty` to specify the molecular diffusivity of each component must be changed because that property is no longer exposed. The specification is now done at the mixture level and the individual diffusivities are input directly into an array profile.

Previous Release	STAR-CCM+ v9.02
`gasMixture_0.getMaterialPropertie`	`gasMixture_0.getMaterialProperties().getMaterialProperty(MolecularDiffusivityProperty.class).`
	`setMethod(ProfileMolecularDiffusivityMethod.class);`
	`ProfileMolecularDiffusivityMethod`

Previous Release	STAR-CCM+ v9.02
`s().getMaterialProperty(MolecularDiffusivityProperty.class).`	`profileMolecularDiffusivityMethod_0 =`
`setMethod(MixtureUserDiffusivityMethod.class);`	`((ProfileMolecularDiffusivityMethod) gasMixture_0.getMaterialPropertie s().`
`UserDiffusivityMethod userDiffusivityMethod_0 =`	`getMaterialProperty(MolecularDiffusivityProperty.class).getMethod( ));`
`((UserDiffusivityMethod) gasComponent_0.getMaterialPropert ies().`	`ComponentDiffusivityArray componentDiffusivityArray_0 =`
`getMaterialProperty(ComponentMolecularDiffusivityProperty.class).getMethod());`	`profileMolecularDiffusivityMethod_0.getComponentDiffusivityArray() ;`
`UserDiffusivity userDiffusivity_0 =`	`componentDiffusivityArray_0.setMethod(CompositeArrayProfileMethod.class);`
`userDiffusivityMethod_0.getUserDiffusivity();`	`ScalarProfile componentDiffusivity_0 =`
`userDiffusivity_0.setMethod(UserScalarProfileMethod.class);`	`componentDiffusivityArray_0.getMethod(CompositeArrayProfileMethod.class).`
	`getProfile(gasComponent_0.getArrayIndex());`
	`componentDiffusivity_0.setMethod(UserScalarProfileMethod.class);`

Changes to DFBI

In STAR-CCM+ v9.02, the body property, `Parts`, has been renamed to `Body Surface` so as to avoid confusion with geometry parts. Macros that refer to this property require updating as follows:

Previous Release	STAR-CCM+ v9.02
`Body body_0 =`	`Body body_0 =`
`((Body) simulation_0.get(BodyManager.class).getObject("Boat"));`	`((Body) simulation_0.get(BodyManager.class).getObject("Boat"));`
`Region region_0 =`	`Region region_0 =`
`simulation_0.getRegionManager().getRegion("Region_1");`	`simulation_0.getRegionManager().getRegion("Region_1");`
`Boundary boundary_0 =`	`Boundary boundary_0 =`
`region_0.getBoundaryManager().getBoundary("hull");`	`region_0.getBoundaryManager().getBoundary("hull");`
`body_0.getParts().setObjects(boundary_0);`	`body_0.getBodySurface().setObjects(boundary_0);`

The body properties *Initial Center of Mass*, *Initial Velocity*, *Initial Angular Velocity*, *Initial Coordinate System*, *Moments of Inertia* and *Off Diagonal Moments of Inertia* have been moved to a new **Initial Values Manager** node of the body. Each of these properties can now be specified in a user-defined coordinate system. Corresponding macros need to be changed as follows:

Previous Release	STAR-CCM+ v9.02
`Body body_0 =`	`Body body_0 =`
`((Body)`	`((Body)`

Previous Release	STAR-CCM+ v9.02

`simulation_0.get(BodyManager.class).getObject("Boat"));`
 	
`body_0.getInitialPosition().setComponents(1.0, 0.0, 0.0);`
	`simulation_0.get(BodyManager.class).getObject("Boat"));`
	
	`CenterOfMass centerOfMass_0 =`
	
	`((CenterOfMass) body_0.getInitialValueManager().getObject("Center of Mass"));`
	
	`Coordinate coordinate_0 =`
	
	`centerOfMass_0.getPosition();`
	
	`Units units_0 =`
	
	`((Units) simulation_0.getUnitsManager().getObject("m"));`
	
	`coordinate_0.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {1.0, 0.0, 0.0}));`

`Body body_0 =`
 	
`((Body) simulation_0.get(BodyManager.class).getObject("Boat"));`
 	
`body_0.getInitialVelocity().setComponents(0.0, 0.0, 1.0);`
	`Body body_0 =`
	
	`((Body) simulation_0.get(BodyManager.class).getObject("Boat"));`
	
	`Velocity velocity_0 =`
	
	`((Velocity)`

Previous Release	STAR-CCM+ v9.02
	`body_0.getInitialValueManager().getObject("Velocity"));` `velocity_0.getValue().setComponents(0.0, 0.0, 1.0);`

`Body body_0 =`
`((Body) simulation_0.get(BodyManager.class).getObject("Boat"));`
`body_0.getInitialAngularVelocity().setComponents(1.0, 0.0, 0.0);`
	`Body body_0 =`
	`((Body) simulation_0.get(BodyManager.class).getObject("Boat"));`
	`AngularVelocity angularVelocity_0 =`
	`((AngularVelocity) body_0.getInitialValueManager().getObject("Angular Velocity"));`
	`angularVelocity_0.getValue().setComponents(1.0, 0.0, 0.0);`

`Body body_0 =`
`((Body) simulation_0.get(BodyManager.class).getObject("Boat"));`
`LabCoordinateSystem labCoordinateSystem_0 =`
`simulation_0.getCoordinateSystemM`
	`Body body_0 =`
	`((Body) simulation_0.get(BodyManager.class).getObject("Boat"));`
	`Orientation orientation_0 =`
	`((Orientation)`

Previous Release	STAR-CCM+ v9.02

`anager().getLabCoordinateSystem();`
`body_0.setInitialCoordinateSystem(labCoordinateSystem_0);`
	`body_0.getInitialValueManager().getObject("Orientation"));`
	`LabCoordinateSystem labCoordinateSystem_0 =`
	`simulation_0.getCoordinateSystemManager().getLabCoordinateSystem();`
	`orientation_0.setCoordinateSystem(labCoordinateSystem_0);`

`Body body_0 =`
`((Body) simulation_0.get(BodyManager.class).getObject("Boat"));`
`body_0.getMomentsOfInertia().setComponents(100.0, 100.0, 100.0);`
`body_0.getOffDiagonalMomentsOfInertia().setComponents(50.0, 20.0, 20.0);`
	`Body body_0 =`
	`((Body) simulation_0.get(BodyManager.class).getObject("Boat"));`
	`MomentOfInertia momentOfInertia_0 =`
	`((MomentOfInertia) body_0.getInitialValueManager().getObject("Moment of Inertia"));`
	`momentOfInertia_0.getDiagonalComponents().setComponents(100.0, 100.0, 100.0);`
	`momentOfInertia_0.getOffDiagonalComponents().setComponents(50.0, 20.0, 20.0);`

In previous versions of STAR-CCM+, it was only possible to specify the body's moment of inertia with respect to the initial coordinate system and the body's center of mass. The new **Moment of Inertia** node lets you specify the moment of inertia with respect to an arbitrary coordinate system and its origin. To get the old behavior, you may need to add the following to existing macros:

```
Body body_0 =
```

```
  ((Body) simulation_0.get(BodyManager.class).getObject("Boat"));
```

```
Orientation orientation_0 =
```

```
  ((Orientation)
body_0.getInitialValueManager().getObject("Orientation"));
```

```
CoordinateSystem orientationCoordinateSystem_0 =
```

```
  orientation_0.getCoordinateSystem();
```

```
MomentOfInertia momentOfInertia_0 =
```

```
  ((MomentOfInertia) body_0.getInitialValueManager().getObject("Moment
of Inertia"));
```

```
momentOfInertia_0.setCoordinateSystem(orientationCoordinateSystem_0);
```

```
momentOfInertia_0.setUseCenterOfMass(true);
```

Parts: Changes to Simple Shape Part Creation

The way in which STAR-CCM+ creates simple shape parts has changed. Modify your existing macros accordingly.

Previous Release	STAR-CCM+ v9.02

`MeshPartFactory meshPartFactory_0 =`
 	
`simulation_0.get(MeshPartFactory.`
	`MeshPartFactory meshPartFactory_0 =`
	
	`simulation_0.get(MeshPartFactory.`

Previous Release	STAR-CCM+ v9.02

class);
SimpleBlockPart simpleBlockPart_0 =
 	
meshPartFactory_0.createNewBlockPart(simulation_0.get(SimulationPartManager.class));
LabCoordinateSystem labCoordinateSystem_0 =
 	
simulation_0.getCoordinateSystemManager().getLabCoordinateSystem();
 	
simpleBlockPart_0.setCoordinateSystem(labCoordinateSystem_0);
Coordinate coordinate_0 =
 	
simpleBlockPart_0.getCorner1();
 	
coordinate_0.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {-1.0, -1.0, -1.0}));
Coordinate coordinate_1 =
 	
simpleBlockPart_0.getCorner2();
 	
coordinate_1.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {1.0, 1.0, 1.0}));
coordinate_0.setValue(new DoubleVector(new double[] {1.0, 1.0, 1.0}));
coordinate_1.setValue(new DoubleVector(new double[] {2.0, 2.0, 2.0}));
	class);
	SimpleBlockPart simpleBlockPart_0 =
	
	meshPartFactory_0.createNewBlockPart(simulation_0.get(SimulationPartManager.class));
	
	simpleBlockPart_0.setDoNotRetessellate(true);
	LabCoordinateSystem labCoordinateSystem_0 =
	
	simulation_0.getCoordinateSystemManager().getLabCoordinateSystem();
	
	simpleBlockPart_0.setCoordinateSystem(labCoordinateSystem_0);
	Coordinate coordinate_0 =
	
	simpleBlockPart_0.getCorner1();
	
	coordinate_0.setCoordinateSystem(labCoordinateSystem_0);
	
	coordinate_0.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {-1.0, -1.0, -1.0}));
	coordinate_0.setValue(new DoubleVector(new double[] {1.0, 1.0, 1.0}));
	Coordinate coordinate_1 =
	
	simpleBlockPart_0.getCorner2();
	
	coordinate_1.setCoordinateSystem(labCoordinateSystem_0);
	
	coordinate_1.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {1.0, 1.0, 1.0}));
	coordinate_1.setValue(new

Previous Release	STAR-CCM+ v9.02

	`DoubleVector(new double[] {2.0, 2.0, 2.0}));`
	
	`simpleBlockPart_0.rebuildSimpleShapePart();`
	
	`simpleBlockPart_0.setDoNotRetessellate(false);`

Macro API Changes 8.06

In STAR-CCM+ 8.06, the macro API changed for dataset functions, the Turbulent Dispersion model, part contacts, planar motion, the Granular Temperature model, the Solid Stress model, and mesh operations.

Surface Time-History Dataset Functions

The `TimeHistoryImportedModelDerivedData` class has been renamed to `ProbeTimeHistoryImportedModelDerivedData`.

Previous Release	STAR-CCM+ v8.06
`public void testhL(){`	`public void testhL(){`
`double[] verificationValue = {17762, 1156700};`	`double[] verificationValue = {17762, 1156700};`
`TimeHistoryImportedModelDerivedData h_l1 = (TimeHistoryImportedModelDerivedData) getSimulation().get(DerivedDataManager.class).hasObject("h_l1");`	`ProbeTimeHistoryImportedModelDerivedData h_l1 = (ProbeTimeHistoryImportedModelDerivedData)`
`DerivedDataView3D derivedDataView3D = h_l1.getDerivedDataView3D();`	`getSimulation().get(DerivedDataManager.class).hasObject("h_l1");`
`SingleSliderValue`	`DerivedDataView3D`

Previous Release	STAR-CCM+ v8.06
`singleSliderValue = derivedDataView3D.getSliderVal();`	`derivedDataView3D = h_l1.getDerivedDataView3D();`
`singleSliderValue.setValue(0.5);`	`SingleSliderValue singleSliderValue = derivedDataView3D.getSliderVal();`
`double[] range = createScalarScene(h_l1);`	`singleSliderValue.setValue(0.5);`
`setMessage(FourierTransformsUtils .verifyFieldFunctionValue(range, verificationValue));`	`double[] range = createScalarScene(h_l1);`
`}`	`setMessage(FourierTransformsUtils .verifyFieldFunctionValue(range, verificationValue));`
	`}`

Changes to the Turbulent Dispersion Model

The Turbulent Prandtl Number has moved from being a property of the Turbulent Dispersion Model to having its own manager, now supporting Constant, Field Function, and specific models as alternative methods. As a result, any old macro that sets this property explicitly, rather than by default, will fail to compile.

Previous Release	STAR-CCM+ v8.06
`PhaseInteraction phaseInteraction_0 =`	`PhaseInteraction phaseInteraction_0 =`
`((PhaseInteraction) multiPhaseInteractionModel_0.getP haseInteractionManager().getPhase`	`((PhaseInteraction) multiPhaseInteractionModel_0.getP haseInteractionManager().getPhase`

Previous Release	STAR-CCM+ v8.06
`Interaction("Phase Interaction 1"));`	`Interaction("Phase Interaction 1"));`
`TurbulentDispersionForceModel turbulentDispersionForceModel_0 =`	`TurbulentDispersionForceModel turbulentDispersionForceModel_0 =`
`phaseInteraction_0.getModelManager().getModel(TurbulentDispersionForceModel.class);`	`phaseInteraction_0.getModelManager().getModel(TurbulentDispersionForceModel.class);`
`turbulentDispersionForceModel_0.setPrandtlNo_t(0.9);`	`ConstantInteractionCoefficientMethod constantInteractionCoefficientMethod_0 =`
	`turbulentDispersionForceModel_0.getTurbulentPrandtlNumber().getMethod(ConstantInteractionCoefficientMethod.class);`
	`constantInteractionCoefficientMethod_0.getQuantity().setValue(0.9);`

Part Contacts

In previous versions of STAR-CCM+, contacts were not journaled with sufficient information to identify the correct contact. For example, when composite parts were involved, two contacts could have the same name. In STAR-CCM+ v8.06, the macro API for part contacts has changed to improve journaling, and now contacts are correctly identified. Update these macros as follows:

Previous Release	STAR-CCM+ v8.06
`Body body_0 =`	`Body body_0 =`
`((Body)`	`((Body)`

Previous Release	STAR-CCM+ v8.06
`cadModel_0.getBodyManager().getObject("Body 1"));`	`cadModel_0.getBodyManager().getObject("Body 1"));`
`Body body_1 =`	`Body body_1 =`
`((Body) cadModel_0.getBodyManager().getObject("Body 2"));`	`((Body) cadModel_0.getBodyManager().getObject("Body 2"));`
`cadModel_0.createParts(new NeoObjectVector(new Object[] {body_0, body_1}), "SharpEdges", 30.0, 2, true, 1.0E-5);`	`cadModel_0.createParts(new NeoObjectVector(new Object[] {body_0, body_1}), "SharpEdges", 30.0, 2, true, 1.0E-5);`
`PartContact partContact_0 =`	`SolidModelPart solidModelPart_0 =`
`((PartContact) simulation_0.get(PartContactManager.class).getObject("Part Contact"));`	`((SolidModelPart) simulation_0.get(SimulationPartManager.class).getPart("Body 1"));`
	`SolidModelPart solidModelPart_1 =`
	`((SolidModelPart) simulation_0.get(SimulationPartManager.class).getPart("Body 2"));`
	`PartContact partContact_0 =`
	`simulation_0.get(PartContactManager.class).getPartContact(solidModelPart_0,solidModelPart_1);`

Planar Motion

Some properties of the Planar Motion Mechanism in DFBI have been renamed for clarity. Java macros that were recorded in STAR-CCM+ v8.04 will still work in v8.06, however support for these macros may be dropped in the future.

Previous Release	STAR-CCM+ v8.06
`Simulation simulation_0 =`	`Simulation simulation_0 =`
`getActiveSimulation();`	`getActiveSimulation();`
`Body body_0 =`	`Body body_0 =`
`((Body) simulation_0.get(BodyManager.class).getObject("Boat"));`	`((Body) simulation_0.get(BodyManager.class).getObject("Boat"));`
`((PlanarMotionCarriage) body_0.getMotionType()).getPlanarMotionOption().setSelected(PlanarMotionOption.PLANAR_MOTION_MECHANISM);`	`((PlanarMotionCarriage) body_0.getMotionType()).getPlanarMotionOption().setSelected(PlanarMotionOption.PLANAR_MOTION_MECHANISM);`
`((PlanarMotionMechanism) ((PlanarMotionCarriage) body_0.getMotionType()).getPlanarMotionType()).getForwardVelocity().setValue(1.0);`	`((PlanarMotionMechanism) ((PlanarMotionCarriage) body_0.getMotionType()).getPlanarMotionType()).getVelocityX().setValue(1.0);`
`((PlanarMotionMechanism) ((PlanarMotionCarriage) body_0.getMotionType()).getPlanarMotionType()).getSwayAmplitude().setValue(2.0);`	`((PlanarMotionMechanism) ((PlanarMotionCarriage) body_0.getMotionType()).getPlanarMotionType()).getAmplitudeY().setValue(2.0);`
`((PlanarMotionMechanism) ((PlanarMotionCarriage) body_0.getMotionType()).getPlanar`	`((PlanarMotionMechanism) ((PlanarMotionCarriage) body_0.getMotionType()).getPlanar`

Previous Release	STAR-CCM+ v8.06
`MotionType()).getSwayFrequency().setValue(0.1);`	`MotionType()).getOscillationFrequency().setValue(0.1);`

Changes to the Granular Temperature Model in EMP

In STAR-CCM+ v8.04, the Granular Temperature Model was selected automatically after selecting the Granular Pressure Model. In STAR-CCM+ v8.06, the `GranularTemperatureModel` class is replaced by the `GranularTemperatureProviderGroup` class. This group provides options to select any of the 3 types of granular temperature models: Algebraic (`AlgebraicGranularTemperatureModelProvider`), Transport (`GranularTemperatureTransportModelProvider`), and User-defined (`UserDefinedGranularTemperatureModelProvider`).

The Kinetic Regime of the `GranularPressureModel` class has been removed. The `KineticTheoryProvider` class has also been removed, meaning that the `getKineticTheory()` and `getKineticRegimeOption()` methods of the `GranularPressureModel` class have been removed in STAR-CCM+ v8.06.

Previous Release	STAR-CCM+ v8.06
`Simulation simulation_0 = getActiveSimulation();`	`Simulation simulation_0 = getActiveSimulation();`
`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Physics 1"));`	`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Physics 1"));`
`physicsContinuum_0.enable(GranularTemperatureModel.class);`	`physicsContinuum_0.enable(GranularTemperatureProviderGroup.class);`
	`physicsContinuum_0.enable(AlgebraicGranularTemperatureModelProvider.class);`

Previous Release	STAR-CCM+ v8.06
`Simulation simulation_0 = getActiveSimulation();`	`Simulation simulation_0 = getActiveSimulation();`
`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Physics 1"));`	`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0.getContinuumManager().getContinuum("Physics 1"));`
`physicsContinuum_0.enable(GranularTemperatureModel.class);`	
`GranularPressureModel granularPressureModel_0 = physicsContinuum_0.getModelManager().getModel(GranularPressureModel.class);`	
`granularPressureModel_0.getKineticRegimeOption().setSelected(KineticRegimeOption.NONE);`	

Solid Stress Model Name Change

In STAR-CCM+ v8.06, the `SolidStress` class has been renamed to `FvSolidStressModel`.

Previous Release	STAR-CCM+ v8.06
`private void execute0() {`	`private void execute0() {`
`Simulation simulation_0 =`	`Simulation simulation_0 =`
`getActiveSimulation();`	`getActiveSimulation();`
`PhysicsContinuum physicsContinuum_0 =`	`PhysicsContinuum physicsContinuum_0 =`
`simulation_0.getContinuumManager().createContinuum(PhysicsContinuum.class);`	`simulation_0.getContinuumManager().createContinuum(PhysicsContinuum.class);`
`physicsContinuum_0.enable(ThreeDimensionalModel.class);`	`physicsContinuum_0.enable(ThreeDimensionalModel.class);`
`physicsContinuum_0.enable(SteadyModel.class);`	`physicsContinuum_0.enable(SteadyModel.class);`
`physicsContinuum_0.enable(SolidModel.class);`	`physicsContinuum_0.enable(SolidModel.class);`
`physicsContinuum_0.enable(SolidStressModel.class);`	`physicsContinuum_0.enable(FvSolidStressModel.class);`

Previous Release	STAR-CCM+ v8.06
`physicsContinuum_0.enable(LinearIsotropicElasticModel.class);`	`physicsContinuum_0.enable(LinearIsotropicElasticModel.class);`
`physicsContinuum_0.enable(ConstantDensityModel.class);`	`physicsContinuum_0.enable(ConstantDensityModel.class);`
`physicsContinuum_0.enable(SegregatedSolidEnergyModel.class);`	`physicsContinuum_0.enable(SegregatedSolidEnergyModel.class);`
`physicsContinuum_0.enable(CellQualityRemediationModel.class);`	`physicsContinuum_0.enable(CellQualityRemediationModel.class);`
`}`	`}`

Removal of Flat List and Flat List & Merge Options from the Output Part Options Property in Mesh Operations

In STAR-CCM+ v8.06, the **Flat List** and *Flat List & Merge by Name* options are removed from the *Output Parts Options* property. This property was found in the *Extract Volume*, *Unite*, *Intersect*, and *Subtract* mesh operation dialogs. Any reference to these options must be removed from your old macros.

Previous Release	STAR-CCM+ v8.06
`ExtractVolumeOperation extractVolumeOperation_2 =`	`ExtractVolumeOperation extractVolumeOperation_1 =`
`(ExtractVolumeOperation) simulation_0.get(MeshOperationManager.class).createExtractVolumeOperation();`	`(ExtractVolumeOperation) simulation_0.get(MeshOperationManager.class).createExtractVolumeOperation();`
`Units units_0 =`	`Units units_0 =`

Previous Release	STAR-CCM+ v8.06
`simulation_0.getUnitsManager().getPreferredUnits(new IntVector(new int[] {0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}));`	`simulation_0.getUnitsManager().getPreferredUnits(new IntVector(new int[] {0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}));`
`SolidModelPart solidModelPart_0 =`	`SolidModelPart solidModelPart_0 =`
`((SolidModelPart) simulation_0.get(SimulationPartManager.class).getPart("Body 1"));`	`((SolidModelPart) simulation_0.get(SimulationPartManager.class).getPart("Body 1"));`
`SolidModelPart solidModelPart_1 =`	`SolidModelPart solidModelPart_1 =`
`((SolidModelPart) simulation_0.get(SimulationPartManager.class).getPart("Body 2"));`	`((SolidModelPart) simulation_0.get(SimulationPartManager.class).getPart("Body 2"));`
`SolidModelPart solidModelPart_2 =`	`SolidModelPart solidModelPart_2 =`
`((SolidModelPart) simulation_0.get(SimulationPartManager.class).getPart("Body 3"));`	`((SolidModelPart) simulation_0.get(SimulationPartManager.class).getPart("Body 3"));`
`extractVolumeOperation_1.getInputGeometryObjects().setObjects(solidModelPart_0, solidModelPart_1, solidModelPart_2);`	`extractVolumeOperation_1.getInputGeometryObjects().setObjects(solidModelPart_0, solidModelPart_1, solidModelPart_2);`
`extractVolumeOperation_1.getVolum`	`extractVolumeOperation_1.getVolum`

Previous Release	STAR-CCM+ v8.06
`eModeOption().setSelected(ExtractVolumeOption.AUTOMATIC);`	`eModeOption().setSelected(ExtractVolumeOption.AUTOMATIC);`
`extractVolumeOperation_1.execute();`	`extractVolumeOperation_1.execute();`
`extractVolumeOperation_1.getDisplayMode().setSelected(MeshOperationDisplayModeOption.FLAT_UNMERGED);`	

Macro API Changes 8.04

In STAR-CCM+ 8.04, the macro API changed for VOF, the Ffowcs Williams-Hawkings models, multi-component fluids, battery meshing, and fluid film.

VOF Wall Porosity

Before setting the alpha and beta profiles for a phase, the *Wall Porosity Specification* option needs to be set to **Specified**. In the macro API, this option is represented by the `USER_DEFINED` option for `VofWallPorosityOption`.

Previous Release	STAR-CCM+ v8.04
`PhaseConditions phaseConditions_0 =`	`PhaseConditions phaseConditions_0 =`
`((PhaseConditions) boundary_0.get(PhaseConditionsManager.class).getPhaseConditions("Gas"));`	`((PhaseConditions) boundary_0.get(PhaseConditionsManager.class).getPhaseConditions("Gas"));`
`VofWallPorosityAlphaProfile vofWallPorosityAlphaProfile_0 = phaseConditions_0.getPhaseValueMa`	`phaseConditions_0.getPhaseConditionManager().get(VofWallPorosityOp`

Previous Release	STAR-CCM+ v8.04
`nager().get(VofWallPorosityAlphaProfile.class);`	`tion.class).setSelected(VofWallPorosityOption.USER_DEFINED);`
`vofWallPorosityAlphaProfile_0.getMethod(ConstantScalarProfileMethod.class).getQuantity().setValue(100000.0);`	`VofWallPorosityAlphaProfile vofWallPorosityAlphaProfile_0 =`
`VofWallPorosityBetaProfile vofWallPorosityBetaProfile_0 =`	`phaseConditions_0.getPhaseValueManager().get(VofWallPorosityAlphaProfile.class);`
`phaseConditions_0.getPhaseValueManager().get(VofWallPorosityBetaProfile.class);`	`vofWallPorosityAlphaProfile_0.getMethod(ConstantScalarProfileMethod.class).getQuantity().setValue(100000.0);`
`vofWallPorosityBetaProfile_0.getMethod(ConstantScalarProfileMethod.class).getQuantity().setValue(100000.0);`	`VofWallPorosityBetaProfile vofWallPorosityBetaProfile_0 =`
	`phaseConditions_0.getPhaseValueManager().get(VofWallPorosityBetaProfile.class);`
	`vofWallPorosityBetaProfile_0.getMethod(ConstantScalarProfileMethod.class).getQuantity().setValue(100000.0);`

Ffowcs Williams-Hawkings Aeroacoustics

An additional model is required when selecting this method in the physics continuum.

Previous Release	STAR-CCM+ v8.04
`PhysicsContinuum`	`PhysicsContinuum`

Previous Release	STAR-CCM+ v8.04
`physicsContinuum_0 = ((PhysicsContinuum) simulation_0 .getContinuumManager() .getContinuum("Fluid"));`	`physicsContinuum_0 = ((PhysicsContinuum) simulation_0 .getContinuumManager() .getContinuum("Fluid"));`
`physicsContinuum_0 .enable(AcousticsTopModel.class);`	`physicsContinuum_0 .enable(AcousticsTopModel.class);`
`physicsContinuum_0 .enable(FwhModel.class);`	`physicsContinuum_0 .enable(FwhUnsteadyBaseModel.class);`
	`physicsContinuum_0 .enable(FwhModel.class);`

Volume Weighted Mixture Methods for Multi-Component Fluids

From STAR-CCM+ v8.04 onwards, there are no volume weighted mixture methods for conductivity and dynamic viscosity for multi-component gases and liquids.

For simulation files created in previous releases and restored in STAR-CCM+ v8.04, these methods are still provided for multi-component gases and liquids that used them. This provision does not apply to Lagrangian multi-component droplets, where the methods are removed entirely and are not restored.

Macros containing these methods should be re-recorded with appropriate alternatives.

Meshing in the Battery Simulation Module

From STAR-CCM+ v8.04 onwards, the Battery Mesher model has been removed and replaced with a new Battery Mesh continuum with the Surface Remesher and Thin Mesher models.

Previous Release	STAR-CCM+ v8.04
`batteryModule_0.generateRegions(new NeoObjectVector(new Object[]{}));`	`batteryModule_0.generateRegions(n`
`MeshContinuum meshContinuum_0 =`	
`simulation_0.getContinuumManager(`	

Previous Release	STAR-CCM+ v8.04

```).createContinuum(MeshContinuum.class);```
```meshContinuum_0.enable(BatteryMesherModel.class);```
```BatteryMesherModel batteryMesherModel_0 =```
```meshContinuum_0.getModelManager().getModel(BatteryMesherModel.class);```
```BatteryMeshContinuum batteryMeshContinuum_0 =```
```batteryMesherModel_0.getBatteryMeshContinuum();```
```batteryMeshContinuum_0.add(batteryModule_0);```
```MeshPipelineController meshPipelineController_0 =```
```simulation_0.get(MeshPipelineController.class);```
```meshPipelineController_0.generateVolumeMesh();```
```ew NeoObjectVector(new Object[]{}));```
``batteryModule_0.setupMeshing();

Changes to the Fluid Film Thickness Field Function

In STAR-CCM+ v8.04, the property for the Fluid Film Thickness field function has changed.

Previous Release	STAR-CCM+ v8.04
`PrimitiveFieldFunction primitiveFieldFunction_0 =`	`PrimitiveFieldFunction primitiveFieldFunction_0 =`
`((PrimitiveFieldFunction) simulation_0.getFieldFunctionManager().getFunction("ShellThickness"));`	`((PrimitiveFieldFunction) simulation_0.getFieldFunctionManager().getFunction("FluidFilmThickness"));`

Macro API Changes 8.02

In STAR-CCM+ 8.02, the macro API changed for VOF, directed meshing, battery modeling, melting, plots, and vaporization.

VOF Cavitation Model

The model is now called `VofCavitationSingleComponentModel` (previously `CavitationModel`).

The method for setting seed diameter and density has changed for consistency with other models that also use seeds.

Previous Release	STAR-CCM+ 8.02
`Simulation simulation = getActiveSimulation();`	`Simulation simulation = getActiveSimulation();`
`PhysicsContinuum physicsContinuum =`	`PhysicsContinuum physicsContinuum =`
`((PhysicsContinuum) simulation.getContinuumManager().getContinuum("Physics 1"));`	`((PhysicsContinuum) simulation.getContinuumManager().getContinuum("Physics 1"));`
`MultiPhaseInteractionModel multiPhaseInteractionModel = physicsContinuum.getModelManager(`	`MultiPhaseInteractionModel multiPhaseInteractionModel = physicsContinuum.getModelManager(`

Previous Release	STAR-CCM+ 8.02

```).getModel(MultiPhaseInteractionModel.class);```
```PhaseInteraction phaseInteraction = ((PhaseInteraction) multiPhaseInteractionModel.getPhaseInteractionManager().getPhaseInteraction("Phase Interaction 1"));```
```phaseInteraction.enable(CavitationModel.class);```
```CavitationModel cavitationModel = phaseInteraction.getModelManager().getModel(CavitationModel.class);```
```ConstantInteractionCoefficientMethod constantInteractionCoefficientMethod_0 = cavitationModel.getCavitationSeedDensity().getMethod(ConstantInteractionCoefficientMethod.class);```
```constantInteractionCoefficientMethod_0.getQuantity().setValue(1.0E13);```
```ConstantInteractionCoefficientMethod constantInteractionCoefficientMethod_1 = cavitationModel.getCavitationSeed```
```).getModel(MultiPhaseInteractionModel.class);```
```PhaseInteraction phaseInteraction = ((PhaseInteraction) multiPhaseInteractionModel.getPhaseInteractionManager().getPhaseInteraction("Phase Interaction 1"));```
```phaseInteraction.enable(VofCavitationSingleComponentModel.class);```
```PhaseInteractionMaterialModel phaseInteractionMaterialModel_0 = phaseInteraction.getModelManager().getModel(PhaseInteractionMaterialModel.class);```
```PhaseInteractionMaterial phaseInteractionMaterial_0 = ((PhaseInteractionMaterial) phaseInteractionMaterialModel_0.getMaterial());```
``ConstantMaterialPropertyMethod constantMaterialPropertyMethod_0 = ((ConstantMaterialPropertyMethod) phaseInteractionMaterial_0.getMaterialProperties().getMaterialProp

Previous Release	STAR-CCM+ 8.02
`Diameter().getMethod(ConstantInteractionCoefficientMethod.class);`	`erty(VofSeedDensityProperty.class).getMethod());`
`constantInteractionCoefficientMethod_1.getQuantity().setValue(1.0E-5);`	`constantMaterialPropertyMethod_0.getQuantity().setValue(1.0E13);`
	`ConstantMaterialPropertyMethod constantMaterialPropertyMethod_1 = ((ConstantMaterialPropertyMethod) phaseInteractionMaterial_0.getMaterialProperties().getMaterialProperty(VofSeedDiameterProperty.class).getMethod());`
	`constantMaterialPropertyMethod_1.getQuantity().setValue(1.0E-5);`

Directed Meshing

Due to multiple changes in the API for Directed meshing, you are recommended to re-record the macros. The Patch mesh code has been refactored and macro definitions have been improved.

In the previous version of directed meshing, you were required to enter an array of points that intersects the triangle mesh on which the patch curve lies. In the current version, you just enter the starting and ending point of the patch curve, not all the intermediate intersection points.

Replace the following items in your macros:

Previous Release	STAR-CCM+ 8.02
`PatchCurve patchCurve_1 =`	`PatchCurve patchCurve_1 =`
`((PatchCurve) directedPatchSourceMesh_0.getPatchCurveManager().getObject("PatchCurve 17"));`	`((PatchCurve) directedPatchSourceMesh_0.getPatchCurveManager().getObject("PatchCurve 17"));`

Previous Release	STAR-CCM+ 8.02
`directedPatchSourceMesh_0.splitPatchCurveAtPickedPointMacro(patchCurve_1, new DoubleVector(new double[] {-0.0029531574676181487, 0.04985268359976114, 0.025}), new IntVector(new int[] {0, 36, 2, 3, 2}));`	`directedPatchSourceMesh_0.splitPatchCurve(patchCurve_1, new DoubleVector(new double[] {-0.0025257023394583402, 0.049875541256511474, 0.02500000037252903}));`

`PatchVertex patchVertex_1 =`
`((PatchVertex) directedPatchSourceMesh_0.getPatchVertexManager().getObject("PatchVertex 18"));`
`directedPatchSourceMesh_0.createPatchCurveMacro(new DoubleVector(new double[] {-0.009581887163221836, 0.03874162957072258, 0.02500000037252903, -0.007917706854641438, 0.0392085425555706, 0.02500000037252903}), new IntVector(new int[] {0, 0}), new IntVector(new int[] {30, 27}), new IntVector(new int[] {2, 1}), new IntVector(new int[] {0, 346}), new IntVector(new int[] {}), patchVertex_1, patchVertex_0, 2, 0, new BooleanVector(new boolean[] {false, true}));`
	`PatchVertex patchVertex_1 =`
	`((PatchVertex) directedPatchSourceMesh_0.getPatchVertexManager().getObject("PatchVertex 19"));`
	`directedPatchSourceMesh_0.createPatchCurve(patchVertex_0, patchVertex_1, new DoubleVector(new double[] {}), new StringVector(new String[] {}));`
	`directedPatchSourceMesh_0.initializePatchCurveCreation();`
	`currentView_1.setInput(new DoubleVector(new double[] {-0.001121880186180152, 0.037023192911687784, 0.024997045777624932}), new DoubleVector(new double[] {-0.001121880186180152, 0.037023192911687784, 0.09296125688392529}), new DoubleVector(new double[] {0.0,`

Previous Release	STAR-CCM+ 8.02
	1.0, 0.0}), 0.013750805976838906, 1); currentView_1.setInput(new DoubleVector(new double[] {-0.001121880186180152, 0.037023192911687784, 0.024997045777624932}), new DoubleVector(new double[] {-0.001121880186180152, 0.037023192911687784, 0.09296125688392529}), new DoubleVector(new double[] {0.0, 1.0, 0.0}), 0.013750805976838906, 1);
directedPatchSourceMesh_0.DefineMeshMultiplePatchCurvesMacro(new IntVector(new int[] {20, 8, 35, 37, 7, 22, 14, 33, 34, 36, 38, 18}), 14, false);	directedPatchSourceMesh_0.defineMeshMultiplePatchCurves(new NeoObjectVector(new Object[] {patchCurve_16, patchCurve_17, patchCurve_18, patchCurve_19, patchCurve_20, patchCurve_21, patchCurve_22, patchCurve_23, patchCurve_24, patchCurve_25, patchCurve_26, patchCurve_27}), 14, false);
directedPatchSourceMesh_0.DefineMeshPatchCurveMacro(6, patchCurve_6.getStretchingFunction(), 0.1, 0.1, 5, false, false);	directedPatchSourceMesh_0.defineMeshPatchCurve(patchCurve_6, patchCurve_6.getStretchingFunction(), 0.1, 0.1, 5, false, false);
directedPatchSourceMesh_0.splitPatchCurveAtConstructionPointMacro(patchVertex_1, patchCurve_13);	directedPatchSourceMesh_0.splitPatchCurve(patchVertex_1, patchCurve_13);
directedPatchSourceMesh_0.createPatchCurveVertexModificationMacro(	directedPatchSourceMesh_0.movePatchVertex(patchVertex_0, new

Previous Release	STAR-CCM+ 8.02
`patchVertex_0, new DoubleVector(new double[] {-0.06413809210062027, -0.0012215570313856006, 0.5}), new DoubleVector(new double[] {0.004088500048965216, ..., 0.5}), new IntVector(new int[] {0, 36}), new DoubleVector(new double[] {}), new IntVector(new int[] {0, 0}), new NeoObjectVector(new Object[] {patchVertex_1, patchVertex_0, patchVertex_0, patchVertex_2}), new IntVector(new int[] {2, 2}), new NeoObjectVector(new Object[] {patchCurve_0, patchCurve_1}), new IntVector(new int[] {2, 2}), null);`	`DoubleVector(new double[] {-0.06413809210062027, -0.0012215570313856006, 0.5}), "ON_SURFACE");`
`directedPatchSourceMesh_0.DefineMeshMultiplePatchCurvesMacro(new IntVector(new int[] {2, 6, 7, 10}), 20, false);`	`directedPatchSourceMesh_0.defineMeshMultiplePatchCurves(new NeoObjectVector(new Object[] {patchCurve_0, patchCurve_1, patchCurve_2}), 20, false);`

Battery Modeling

Macros that use the multi-module model in the batteries simulation module require updating.

Previous Release	STAR-CCM+ 8.02
`SolidModelPart solidModelPart_0 =`	`CadPart solidModelPart_0 =`
`((SolidModelPart) simulation_0.get(SimulationPartManager.class).getPart("Battery Module: Battery Module Cell 0,0: +Ve Tab Root"));`	`((CadPart) simulation_0.get(SimulationPartManager.class).getPart("Battery Module: Battery Module Cell 0,0: +Ve Tab Root"));`

Melting

Macros that use the melting models require updating. Add `import star.multiphase` to the import section of your macros, as shown below.

Previous Release	STAR-CCM+ 8.02
`package base.multiphaseFlow;`	`package base.multiphaseFlow;`
`import java.io.File;`	`import java.io.File;`
`import java.util.*;`	`import java.util.*;`
`import star.base.neo.*;`	`import star.base.neo.*;`
`import star.base.report.*;`	`import star.base.report.*;`
`import star.common.*;`	`import star.common.*;`
`import star.energy.*;`	`import star.energy.*;`
`import star.segregatedenergy.*;`	`import star.segregatedenergy.*;`
`import star.flow.*;`	`import star.flow.*;`
`import star.keturb.*;`	`import star.keturb.*;`
`import star.material.*;`	`import star.material.*;`
`import star.metrics.*;`	`import star.metrics.*;`
`import star.turbulence.*;`	`import star.turbulence.*;`
`import star.vis.*;`	`import star.vis.*;`
`import star.vof.*;`	`import star.vof.*;`
`import star.common.StarMacro;`	`import star.multiphase.*;`
	`import star.common.StarMacro;`

Getting Plot Samples

The methods, `getPlotSamples` and `getAllSamples`, return the samples that are stored by a monitor. Changes to the code have decoupled the monitor trigger type from the data series. In previous releases, the x-values were defined by the monitor trigger. In the current release, the x-values are specified by the monitor. The caller now has to specify the monitor used to convert the returned x-values.

Previous Release	STAR-CCM+ 8.02

`// NOTE: The returned xvalues`	`// NOTE: The returned xvalues are now`
	`// specified by the specified monitor.`
	`MonitorData allIterations =`
	`<plotableMonitor>.getAllSamples(<monitorManager>.getIterationMonit`
	`or() );`
	`MonitorData plotIterations =`
	`<plotableMonitor>.getPlotSamples(`

Previous Release	STAR-CCM+ 8.02

used to be // defined by the monitor trigger.
 	
<plotableMonitor>.setTrigger(MonitorTriggerOption.ITER
 	
ATION);
 	
MonitorData allIterations =
 	
<plotableMonitor>.getAllSamples();
 	
MonitorData plotIterations =
 	
<plotableMonitor>.getPlotSamples();
 	
// NOTE: The data previously is cleared
 	
// for a monitor when the triggers are
 	
// changed. This is NO LONGER true.
 	
<plotableMonitor>.setTrigger(MonitorTriggerOption.TIME
 	
STEP);
 	
MonitorData allTime =
	<monitorManager>.getIterationMoni
	
	tor());
	
	// NOTE: The trigger does not need to be
	
	// changed, just the specified monitor.
	
	MonitorData allTime =
	
	<plotableMonitor>.getAllSamples(
	<monitorManager>.getPhysicalTimeMo
	
	nitor());
	
	MonitorData plotTime =
	
	<plotableMonitor>.getPlotSamples(

Previous Release	STAR-CCM+ 8.02

`<plotableMonitor>.getAllSamples() ;`
`MonitorData plotTime =`
`<plotableMonitor>.getPlotSamples( );`
	`<monitorManager>.getPhysicalTimeM`
	`onitor() );`
	`// NOTE: This supports a general`
	`// monitorvsmonitor data requests.`
	`PlotableMonitor xMonitor =`
	`<monitorManager>.getMonitor("Monitor 2");`
	`MonitorData allOtherX =`
	`<plotableMonitor>.getAllSamples( xMonitor );`
	`MonitorData plotOtherX =`
	`<plotableMonitor>.getPlotSamples( xMonitor );`

Vaporization Latent Heat

Since STAR-CCM+ 7.06, `VaporizationLatentHeat` (and classes that are related to it) have moved from `star.lagrangian` to `star.energy`. Macros that refer to a particular class might fail to compile. Macros that already have both `star.lagrangian.*` and `star.energy.*` are unaffected.

Previous Release	STAR-CCM+ 8.02
`ConstantMaterialPropertyMethod`	`ConstantMaterialPropertyMethod`

Previous Release	STAR-CCM+ 8.02

`constantMaterialPropertyMethod_0 =`
 	
`((ConstantMaterialPropertyMethod)`
 	
`singleComponentDropletMaterial_0.getMaterialProperties().getMaterialProperty(star.lagrangian.VaporizationLatentHeatProperty.class).getMethod());`
	`constantMaterialPropertyMethod_0 =`
	
	`((ConstantMaterialPropertyMethod)`
	
	`singleComponentDropletMaterial_0.getMaterialProperties().getMaterialProperty(star.energy.VaporizationLatentHeatProperty.class).getMethod());`

`singleComponentDropletMaterial_1.getMaterialProperties().getMaterialProperty(star.lagrangian.VaporizationLatentHeatProperty.class).setMethod(star.lagrangian.ClausiusClapeyronLatentHeatMethod.class);`	`singleComponentDropletMaterial_1.getMaterialProperties().getMaterialProperty(star.energy.VaporizationLatentHeatProperty.class).setMethod(star.energy.ClausiusClapeyronLatentHeatMethod.class);`

Macro API Changes 7.06

In STAR-CCM+ 7.06, the macro API changed for battery modeling, VOF, and directed meshing.

Battery: Generating Boundaries from Parts

The recorded boundary name in a macro now includes the part name.

Previous Release	STAR-CCM+ 7.06

`region_0.`
`getBoundaryManager().`
`getBoundary("Stack Front")`
	`region_0.`
	`getBoundaryManager().`
	`getBoundary("Battery Module:`
	`Battery Module Cell 0,0:`
	`Stack.Stack Front")`

VOF Melting: Heat of Phase Change

Selecting the **Latent Heat of Fusion** material node now uses `FusionLatentHeatProperty` instead of `LatentHeatProperty`.

Previous Release	STAR-CCM+ 7.06
`(ConstantMaterialPropertyMethod)`	`((ConstantMaterialPropertyMethod)`

Previous Release	STAR-CCM+ 7.06

`singlePhaseLiquid`
`.getMaterialProperties()`
`.getMaterialProperty(`
`LatentHeatProperty.class)`
`.getMethod());`
	`singlePhaseLiquid`
	`.getMaterialProperties()`
	`.getMaterialProperty(`
	`FusionLatentHeatProperty.class)`
	`.getMethod());`

Directed Mesh

The journaling behavior for the Directed Mesh feature has changed between the current and previous releases. If you recorded macros for Directed Mesh in version 7.04, re-record them in version 7.06.

Macro API Changes 7.04

In STAR-CCM+ 7.04, the macro API changed for the Melting-Solidification model, cavitation, DFBI, and Eulerian Multiphase modeling.

Melting-Solidification: Solidification Path

The solidification path required by the Melting-Solidification model no longer belongs to the model node itself. Instead, you set the path using the **Fraction Solid Curve** material property node.

Previous Release	STAR-CCM+ 7.04.006

`VofEnthalpyMeltingModel`
`vofEnthalpyMeltingModel_0 = eulerianPhase_0`
`.getModelManager() .getModel( VofEnthalpyMeltingModel.class);`
`vofEnthalpyMeltingModel_0 .getSolidificationPathOption()`
`.setSelected( SolidificationPathOption.TABULAR);`
	`SinglePhaseLiquidModel singlePhaseLiquidModel_0 =`
	`eulerianPhase_0 .getModelManager() .getModel(`
	`SinglePhaseLiquidModel.class);`
	`SinglePhaseLiquid singlePhaseLiquid_0 =`
	`((SinglePhaseLiquid) singlePhaseLiquidModel_0 .getMaterial());`
	`singlePhaseLiquid_0 .getMaterialProperties() .getMaterialProperty(FractionSolidCurveProperty.class)`
	`.setMethod(FractionSolidCurveTemperatureTableMethod.class);`
	`singlePhaseLiquid_0 .getMaterialProperties() .getMaterialProperty(FractionSolidCurveProperty.class)`

Previous Release	STAR-CCM+ 7.04.006
	`.setMethod(FractionSolidCurveLinearMethod.class);`

Cavitation: Seed Density and Seed Diameter

The Cavitation phase interaction model now allows you to define the Seed Density and Seed Diameter properties with a field function or constant profile. Therefore, definition of a constant seed density or seed diameter has changed to include both options.

Previous Release	STAR-CCM+ 7.04.006

`cavitationModel_0`
`.getSeedDensity()`
`.setValue(1.0E12);`
	`cavitationModel_0`
	`.getCavitationSeedDensity()`
	`.setMethod(ConstantInteractionCoefficientMethod.class);`
	`ConstantInteractionCoefficientMethod constantInteractionCoefficientMethod_0 =`
	`cavitationModel_`
	`.getCavitationSeedDensity()`
	`.getMethod(ConstantInteractionCoefficientMethod.class);`
	`constantInteractionCoefficientMethod_0`
	`.getQuantity()`
	`.setValue(1.0E12);`

`cavitationModel_0`
`.getSeedDiameter()`
`.setValue(1.0E-6);`
	`cavitationModel_0`
	`.getCavitationSeedDiameter()`
	`.setMethod(ConstantInteractionCoefficientMethod.class);`
	`ConstantInteractionCoefficientMethod constantInteractionCoefficientMethod_1 =`
	`cavitationModel_0`
	`.getCavitationSeedDiameter()`
	`.getMethod(ConstantInteractionCoefficientMethod.class);`

Previous Release	STAR-CCM+ 7.04.006

	`constantInteractionCoefficientMethod_1`
	`.getQuantity()`
	`.setValue(1.0E-6);`

Dynamic Fluid-Body Interaction: External Moment

For the Dynamic Fluid-Body Interaction physics model, the External Moment has been extended so that it is relative to the body position, as opposed to the Center of Mass. Depending on the selected body motion type, the body position can be different from the Center of Mass. Therefore, External Moment has been renamed.

Previous Release	STAR-CCM+ 7.04.006

`ExternalMomentCM`
`externalMomentCM_0 = body_0`
`.getExternalForceAndMomentManager()`
`.createExternalForceAndMoment(ExternalMomentCM.class);`
	`ExternalMoment`
	`externalMoment_0 = body_0`
	`.getExternalForceAndMomentManager()`
	`.createExternalForceAndMoment(ExternalMoment.class);`

Eulerian Multiphase: Phase Turbulence Models

The Eulerian Multiphase model now uses an individual k-epsilon turbulence model for each Eulerian phase. This change affects the following:

- Initial and boundary conditions properties for Eulerian phases
  - Turbulence Intensity
  - Turbulent Viscosity Ratio
  - Turbulent Dissipation Rate
  - Turbulent Kinetic Energy
- k-epsilon turbulence solver
  - Renamed classes
  - Include the AMG Linear Solver with the V-Cycle option
- Enabling k-epsilon turbulence models
  - Remove the redundant turbulence models from the physics continuum
  - Add the new turbulence models to each Eulerian phase
  - Setting model options

Previous Release	STAR-CCM+ 7.04.006
Initial Conditions	

`MultiPhaseTurbulenceIntensityProfile`	`EulerianMultiPhaseModel`
	`eulerianMultiPhaseModel_0 =`
	`physicsContinuum_0`

Previous Release	STAR-CCM+ 7.04.006
multiPhaseTurbulenceIntensityProfile_0 = physicsContinuum_0 .getInitialConditions() .get(MultiPhaseTurbulenceIntensityProfile.class); multiPhaseTurbulenceIntensityProfile_0 .getMethod(ConstantArrayProfileMethod.class) .getQuantity().setArray(new DoubleVector(new double[] {0.06})); MultiPhaseTurbulentViscosityRatioProfile multiPhaseTurbulentViscosityRatioProfile_0 = physicsContinuum_0 .getInitialConditions(.get(MultiPhaseTurbulentViscosityRatioProfile.class); multiPhaseTurbulentViscosityRatioProfile_0 .getMethod(ConstantArrayProfileMethod.class) .getQuantity() .setArray(new DoubleVector(new double[] {255.0}));	.getModelManager() .getModel(EulerianMultiPhaseModel.class); EulerianPhase eulerianPhase_0 = ((EulerianPhase) eulerianMultiPhaseModel_0 .getPhaseManager() .getPhase("Water")); TurbulenceIntensityProfile turbulenceIntensityProfile_0 = eulerianPhase_0 .get(ContinuumInitialConditionManager.class) .get(TurbulenceIntensityProfile.class); turbulenceIntensityProfile_0 .getMethod(ConstantScalarProfileMethod.class) .getQuantity() .setValue(0.06); TurbulentViscosityRatioProfile turbulentViscosityRatioProfile_0 = eulerianPhase_0 .get(ContinuumInitialConditionManager.class) .get(TurbulentViscosityRatioProfile.class); turbulentViscosityRatioProfile_0 .getMethod(ConstantScalarProfileMethod.class) .getQuantity() .setValue(255.0);
Boundary Conditions	
MultiPhaseTurbulenceIntensityProfile	Boundary boundary_0 =

Previous Release	STAR-CCM+ 7.04.006
multiPhaseTurbulenceIntensityProfile_0 = boundary_0 .getValues() .get(MultiPhaseTurbulenceIntensityProfile.class); multiPhaseTurbulenceIntensityProfile_0 .getMethod(ConstantArrayProfileMethod.class) .setArray(new DoubleVector(new double[] {0.05}));	region_0 .getBoundaryManager() .getBoundary("Inlet"); PhaseConditions phaseConditions_0 = ((PhaseConditions) boundary_0 .get(PhaseConditionsManager.class) .getPhaseConditions("Water")); TurbulenceIntensityProfile turbulenceIntensityProfile_0 = phaseConditions_0 .getPhaseValueManager() .get(TurbulenceIntensityProfile .class); turbulenceIntensityProfile_0 .getMethod(ConstantScalarProfileMethod.class) .getQuantity() .setValue(0.05);
k-epsilon Turbulence Solver	
MultiPhaseKeTurbSolver multiPhaseKeTurbSolver_0 = ((MultiPhaseKeTurbSolver) simulation_0 .getSolverManager() .getSolver(MultiPhaseKeTurbSolver.class)); multiPhaseKeTurbSolver_0 .setUrf(0.5);	KeTurbSolver keTurbSolver_0 = ((KeTurbSolver) simulation_0 .getSolverManager() .getSolver(KeTurbSolver.class)); keTurbSolver_0 .setUrf(0.5); AMGLinearSolver aMGLinearSolver_0 = keTurbSolver_0 .getAMGLinearSolver(); aMGLinearSolver_0 .getCycleOption()

Previous Release	STAR-CCM+ 7.04.006
	`.setSelected(` `AMGCycleOption.V_CYCLE);`
Enabling k-epsilon Turbulence Models	

`PhysicsContinuum`
`physicsContinuum_0 =`
`simulation_0`
`.getContinuumManager()`
`.createContinuum(`
`PhysicsContinuum.class);`
`physicsContinuum_0`
`.enable(`
`RansTurbulenceModel.class);`
`physicsContinuum_0`
`.enable(`
`KEpsilonTurbulence.class);`
`physicsContinuum_0`
`.enable(`
`MultiPhaseSkeTurbModel.class);`
`physicsContinuum_0`
`.enable(`
`MultiPhaseKeHighYplusWallTreatment`
`.class);`
	`EulerianMultiPhaseModel`
	`eulerianMultiPhaseModel_0 =`
	`physicsContinuum_0`
	`.getModuleManager()`
	`.getModel(`
	`EulerianMultiPhaseModel.class);`
	`EulerianPhase eulerianPhase_0 =`
	`eulerianMultiPhaseModel_0`
	`.createPhase();`
	`eulerianPhase_0`
	`.enable(`
	`RansTurbulenceModel.class);`
	`eulerianPhase_0`
	`.enable(`
	`KEpsilonTurbulence.class);`
	`eulerianPhase_0`
	`.enable(`
	`SkeTurbModel.class);`
	`eulerianPhase_0`
	`.enable(`
	`KeHighYplusWallTreatment.class)`

Setting k-epsilon Turbulence Model Options	

`MultiPhaseSkeTurbModel`
`multiPhaseSkeTurbModel_0 =`
`physicsContinuum_0`
`.getModelManager()`
`.getModel(`
`MultiPhaseSkeTurbModel.class);`
`multiPhaseSkeTurbModel_0`
`.getSecondaryGradientOption()`
	`EulerianMultiPhaseModel`
	`eulerianMultiPhaseModel_0 =`
	`physicsContinuum_0`
	`.getModelManager()`
	`.getModel(`
	`EulerianMultiPhaseModel.class);`
	`EulerianPhase eulerianPhase_0 =`

Previous Release	STAR-CCM+ 7.04.006

`.setSelected(`
`SecondaryGradientOption.OFF);`
	`((EulerianPhase)`
	`eulerianMultiPhaseModel_0`
	`.getPhaseManager()`
	`.getPhase("Water"));`
	`SkeTurbModel SkeTurbModel_0 =`
	`eulerianPhase_0`
	`.getModelManager()`
	`.getModel(SkeTurbModel.class);`
	`SkeTurbModel_0`
	`.getSecondaryGradientOption()`
	`.setSelected(`
	`SecondaryGradientOption.OFF);`

Macro API Changes 7.02

In STAR-CCM+ 7.02, the macro API changed for energy source options, battery modeling, thermal resistance, derived parts, harmonic balance, and thermal comfort.

Changes to Class Names for Energy Source Options

Various API changes have been made to Energy Source Options for regions (solid, porous and fluid). Additional changes to the Energy Source Options have been made for contact and baffle interfaces.

Previous Release	STAR-CCM+ 7.02.011
Changes to Energy Source Options: the class name for `FluidEnergyUserSourceOption` (fluid and porous regions) and `EnergyUserSourceOption` (solid regions) is now `EnergyUserVolumeSourceOption`.	

`region.getConditions()`
`.get(FluidEnergyUserSourceOption.class)`
`.setSelected(EnergyUserSourceOption.SPECIFIED);`
	`region.getConditions()`
	`.get(EnergyUserVolumeSourceOption.class)`
	`.setSelected(EnergyUserVolumeSourceOption.VOLUMETRIC_HEAT_SOURCE);`

`region.getConditions()`
`.get(EnergyUserSourceOption.class)`
`.setUserSourceOption(true);`
	`region.getConditions()`
	`.get(EnergyUserVolumeSourceOption.class)`
	`.setSelected(EnergyUserVolumeSourceOption.VOLUMETRIC_HEAT_SOURCE);`

Previous Release	STAR-CCM+ 7.02.011
Changes to Energy Source Options for solid regions: the class name for `EnergyUserSource` has changed to `VolumetricHeatSourceProfile`.	

`EnergyUserSource src = region`
`.getValues()`
`.get(EnergyUserSource.class);`
	`VolumetricHeatSourceProfile src = region`
	`.getValues()`
	`.get(VolumetricHeatSourceProfile.class);`

Changes to Energy Source Options for contact and baffle Interfaces: the class name for `EnergyUserSourceOption` is now `EnergyUserSurfaceSourceOption`.	

`interface.getConditions() .get(EnergyUserSourceOption.class) .setUserSourceOption(true);`
`HeatFluxProfile src = interface`
`.getValues() .get(HeatFluxProfile.class);`
	`interface.getConditions()`
	`.get(EnergyUserSurfaceSourceOption.class)`
	`.setSelected(EnergyUserSurfaceSourceOption.HEAT_FLUX);`
	`HeatFluxProfile src =`
	`interface`
	`.getValues()`
	`.get(HeatFluxProfile.class);`

Changes in Boundary Names in the Battery Model

The boundary names in the battery model have changed format. The new names are now cleaner, dropping the extra information that used to appear before the boundary name. Consider the example shown below where the boundary is called `Stack Front`.

Previous Release	STAR-CCM+ 7.02.011

`Boundary boundary_2 =`
`region_0`
`.getBoundaryManager()`
`.getBoundary("Battery Module: Battery Module Cell 0,0: Stack.Stack Front");`
	`Boundary boundary_2 =`
	`region_0`
	`.getBoundaryManager()`
	`.getBoundary("Stack Front");`

All boundary names will need to be updated so that only the name appears in `.getBoundary()`.	

Baffle Interface Thermal Resistance Option

A new wall thermal resistance interface has been added to the code and contains an effective resistance and a multi-layer resistance option. In contrast to the baffle multi-layer resistance, the wall multi-layer resistance is implemented as a `Profile` method on a general `ThermalResistance` profile.

Previous Release	STAR-CCM+ 7.02.011
Setting effective baffle resistance:	

`directBoundaryInterface_0 .getConditions() .get(BaffleThermalOption.class) .setSelected(BaffleThermalOption.CONDUCTING);`
`directBoundaryInterface_0 .getConditions() .get(BaffleThermalResistanceOption.class) .setSelected(BaffleThermalResistanceOption.EFFECTIVE_RESISTANCE);`
`BaffleConductivityProfile baffleConductivityProfile_0 = directBoundaryInterface_0 .getValues() .get(BaffleConductivityProfile.class);`
`((ConstantScalarProfileMethod) baffleConductivityProfile_0.getMethod()).getQuantity().setDefinition("0");`
	`directBoundaryInterface_0`
	`.getConditions()`
	`.get(BaffleThermalOption.class)`
	`.setSelected(BaffleThermalOption.CONDUCTING);`
	`ThermalResistanceProfile thermalResistanceProfile_0 =`
	`directBoundaryInterface_0 .getValues()`
	`.get(ThermalResistanceProfile.class);`
	`thermalResistanceProfile_0`
	`.getMethod(ConstantScalarProfileMethod.class)`
	`.getQuantity()`
	`.setValue(0.0);`

Setting multi-layer baffle resistance:	

`directBoundaryInterface_1`
`.getConditions()`
`.get(BaffleThermalOption.class)`
`.setSelected(BaffleThermalOption.CONDUCTING);`
`directBoundaryInterface_1.getConditions()`
`.get(BaffleThermalResistanceOption.class)`
`.setSelected(BaffleThermalResistanceOption.MULTI_LAYER);`
`MultiLayerBaffleResistance multiLayerBaffleResistance_0 = directBoundaryInterface_1 .getValues() .get(MultiLayerBaffleResistance.class);`
`multiLayerBaffleResistance_0 .setNLayers(1);`
	`directBoundaryInterface_1`
	`.getConditions()`
	`.get(BaffleThermalOption.class)`
	`.setSelected(BaffleThermalOption.CONDUCTING);`
	`ThermalResistanceProfile thermalResistanceProfile_1 =`
	`directBoundaryInterface_1`
	`.getValues()`
	`.get(ThermalResistanceProfile.class);`
	`thermalResistanceProfile_1`
	`.setMethod(MultiLayerThermalResistanceMethod.class);`
	`MultiLayerThermalResistanceMethod multiLayerMethod =`
	`thermalResistanceProfile_1`

Previous Release	STAR-CCM+ 7.02.011

`multiLayerBaffleResistance_0 .setLayerThickness(new DoubleVector(new double[] {2.0}));`
`multiLayerBaffleResistance_0 .setLayerConductivity(new DoubleVector(new double[] {1.2}));`
	`.getMethod(MultiLayerThermalResistanceMethod.class);`
	`multiLayerMethod`
	`.setNLayers(1);`
	`multiLayerMethod .getLayerThicknes s()`
	`.setArray(new DoubleVector(new double[] {2.0}));`
	`multiLayerMethod .getLayerConductiv ity()`
	`.setArray(new DoubleVector(new double[] {1.2}));`

Units Support for Derived Parts

In order to add the ability to supply units for the values used in derived parts, it was necessary to change the APIs to create and set values for these objects. The existing APIs will be kept in place, assuming that the values specified are in internal units.

Previous Release	STAR-CCM+ 7.02.011
The `PartManager.createIsoPart` no longer accepts the number of contour levels followed by a set of values. Values are now supplied using the `SingleIsoValue`, `MultiIsoValue`, and `MultiRangeIsoValue` APIs which use `ScalarPhysicalQuantity` and `ArrayPhysicalQuantity` objects to specify values with units. The `PartManager.createImplicitPart` will be changed similarly.	

`IsoPart isoPart_0 =`
`simulation_0 .getPartManager() .creat eIsoPart(new NeoObjectVector(new Object[] {region_0}), 1, new DoubleVector(new double[] {50.0}), primitiveFieldFunction_0);`
	`IsoPart isoPart_0 =`
	`simulation_0 .getPartManager() .cre ateIsoPart(new NeoObjectVector(new Object[] {region_0}), primitiveFieldFunction_0);`
	`isoPart_0 .setMode(0);`
	`SingleIsoValue singleIsoValue_0 =`
	`isoPart_0 .getSingleIsoValue();`
	`singleIsoValue_0 .getValueQuantity( ) .setValue(50.0);`
	`singleIsoValue_0 .getValueQuantity( ) .setUnits(units_1);`
	`isoPart_0 .setMode(0);`

Previous Release	STAR-CCM+ 7.02.011
	`singleIsoValue_0 .getValueQuantity() .setUnits(units_1);` `singleIsoValue_0 .getValueQuantity() .setValue(50.0);`
The `PartManager.createThreshold` continues to specify two values in a range parameter, but it is now followed by units.	

`ThresholdPart thresholdPart_0 =`
`simulation_0 .getPartManager() .createThresholdPart(new NeoObjectVector(new Object[] {region_0}), new DoubleVector(new double[] {-25.0, 112.0}), primitiveFieldFunction_0, 0);`
	`ThresholdPart thresholdPart_0 =`
	`simulation_0`
	`.getPartManager()`
	`.createThresholdPart(new NeoObjectVector(new Object[] {region_0}), new DoubleVector(new double[] {-25.0, 112.0}), units_1, primitiveFieldFunction_0, 0);`

Simplified the Number of Blades Input for the Harmonic Balance Model

In the past, the number of blades were specified as equal to (2π/Pitch). While this is acceptable for 3D cases, where it corresponds to the physical number of blades on the wheel, it is prone to errors in 2D cases, since this is a non-integral number. The current version now considers the number of blades input to be an integer value for both 3D and 2D cases. In the case of 2D, it now corresponds to the number of passages being modeled.

Previous Release	STAR-CCM+ 7.02.011
For 2D cases the `NumberOfBlades` object needs to be set to `1`. The `Pitch` object needs to be set to (2 π/old `NumberOfBlades`). In the example below, the old `NumberOfBlades` was `12.56637061`, resulting in a `Pitch` of `0.5`.	

`HbBladeRow hbBladeRow_0 = harmonicBalanceFlowModel_0 .createBladeRow();`
`hbBladeRow_0 .setNumberOfBlades(12.56637061);`
	`HbBladeRow hbBladeRow_0 = harmonicBalanceFlowModel_0 .createBladeRow();`
	`hbBladeRow_0 .getBladesPerPitch() .setNumberOfBlades(1);`
	`hbBladeRow_0 .getBladesPerPitch() .getPitch() .setValue(0.5);`

`FourierCoefficientsWakeSpecifier fourierCoefficientsWakeSpecifier_0 =` `boundary_0 .getValues() .get(FourierCoefficientsWakeSpecifier.class);`	`FourierCoefficientsWakeSpecifier fourierCoefficientsWakeSpecifier_0 =` `boundary_0 .getValues() .get(FourierCoefficientsWakeSpecifier.class);`

Previous Release	STAR-CCM+ 7.02.011
`fourierCoefficientsWakeSpecifier_0 .setWakeBlades(12.56637061);`	`fourierCoefficientsWakeSpecifier_0 .getWakeBladesPerPitch() .setNumberOfBlades(1);` `fourierCoefficientsWakeSpecifier_0 .getWakeBladesPerPitch() .getPitch() .setValue(0.5);`
For 3D cases, the `NumberOfBlades` needs to be set to the old value (`12` in the example below). The `Pitch` needs to be set so that it equals 2π radians (`6.283185307` shown below).	

`HbBladeRow hbBladeRow_0 = harmonicBalanceFlowModel_0 .createBladeRow();`
`hbBladeRow_0 .setNumberOfBlades(12);`
	`HbBladeRow hbBladeRow_0 = harmonicBalanceFlowModel_0 .createBladeRow();`
	`hbBladeRow_0 .getBladesPerPitch() .setNumberOfBlades(12);`
	`hbBladeRow_0 .getBladesPerPitch() .getPitch() .setValue(6.283185307);`

`FourierCoefficientsWakeSpecifier fourierCoefficientsWakeSpecifier_0 = boundary_0 .getValues() .get(FourierCoefficientsWakeSpecifier.class);`
`fourierCoefficientsWakeSpecifier_0 .setWakeBlades(12);`
	`FourierCoefficientsWakeSpecifier fourierCoefficientsWakeSpecifier_0 = boundary_0 .getValues() .get(FourierCoefficientsWakeSpecifier.class);`
	`fourierCoefficientsWakeSpecifier_0 .getWakeBladesPerPitch() .setNumberOfBlades(12);`
	`fourierCoefficientsWakeSpecifier_0 .getWakeBladesPerPitch() .getPitch() .setValue(6.283185307);`

This change applies to all other wake options: Gaussian Wake, Sine Wake, Hodson Wake.	

Thermal Comfort Model Now Only Contains Relevant Boundary Conditions

Depending on the boundary type, STAR-CCM+ now only holds the specific boundary conditions, as opposed to all possible boundary conditions. For example a TCM inlet boundary only allows specification of mass flux and inlet temperature. As a result, the boundary type for each boundary needs to be specified in the macro with an additional line.

Previous Release	STAR-CCM+ 7.02.011
In the example below, the left `InletDashLeft` boundary will have the mass flux and inlet temperature conditions set to `0.007` and `12.0` respectively. A new line specifying the boundary type should be added after defining these values:	

Previous Release	STAR-CCM+ 7.02.011

`TcmBoundary tcmBoundary_3 = simulation_0 .get(TcmBoundaryManager .class) .createTcmBoundary();`
`Boundary boundary_10 = region_0 .getBoundaryManager() .getBoundary("InletDashLeft");`
`tcmBoundary_3 .setTcmBoundaryName(boundary_10);`
`tcmBoundary_3 .setPresentationName("Fluid: InletDashLeft: 0");`
`tcmBoundary_3 .getTcmBoundaryTime() .setValue(0.0);`
`tcmBoundary_3 .getTcmBoundaryInletFlux() .setValue(0.007);`
`tcmBoundary_3 .getTcmBoundaryInletT() .setUnits(units_2);`
`tcmBoundary_3 .getTcmBoundaryInletT() .setValue(12.0);`
	`TcmBoundary tcmBoundary_3 = simulation_0 .get(TcmBoundaryManager.class) .createTcmBoundary();`
	`Boundary boundary_10 = region_0 .getBoundaryManager() .getBoundary("InletDashLeft");`
	`tcmBoundary_3 .setTcmBoundaryName(boundary_10);`
	`tcmBoundary_3 .setPresentationName("Fluid: InletDashLeft: 0");`
	`tcmBoundary_3 .getTcmBoundaryTime() .setValue(0.0);`
	`tcmBoundary_3 .getTcmBoundaryInletFlux() .setValue(0.007);`
	`tcmBoundary_3 .getTcmBoundaryInletT() .setUnits(units_2);`
	`tcmBoundary_3 .getTcmBoundaryInletT() .setValue(12.0);`
	`tcmBoundary_3 .getTcmBoundaryOption() .setSelected(TcmBoundaryOption.INLET);`

Other boundary conditions that are affected are: • Convection (`CONVECTION`) • Wall Temperature (`WALL_TEMPERATURE`) • Wall Heat Flux (`WALL_HEAT_FLUX`) • Convection with Conduction (`CONVECTION_CONDUCTION`) • Convection and Radiation (`CONVECTION_RADIATION`) • Convection and Radiation with Conduction (`CONVECTION_CONDUCTION_RADIATION`)	

Macro API Changes 6.06

In STAR-CCM+ 6.06, the macro API changed for DFBI, VOF, melting and solidification, the Solid Stress solver, battery modeling, and OLGA coupling.

Changes for External and Propulsion Forces in DFBI

Propulsion Force and External Force have been extended so that the position of the force can be specified in a coordinate system other than the body coordinate system. The following macro changes result from this:

Previous Release	STAR-CCM+ 6.06.011
Propulsion Force: the type of property `Position` was changed from `VectorPhysicalQuantity` to `Coordinate`	

`propulsionForce_0`
`.getPosition()`
`.setComponents(1.0, 0.0, 0.0);`
	`Units units_0 =`
	`((Units) simulation_0`
	`.getUnitsManager()`
	`.getObject("m"));`
	`propulsionForce_0`
	`.getPositionAsCoordinate()`
	`.setCoordinate( units_0, units_0, units_0, new DoubleVector( new double[] {1.0, 0.0, 0.0} ));`

`propulsionForce_0`
`.getPosition()`
`.setVector( new DoubleVector( new double[] {1.0, 0.0, 0.0} ));`
	`Units units_0 = ((Units) simulation_0`
	`.getUnitsManager()`
	`.getObject("m"));`
	`propulsionForce_0`
	`.getPositionAsCoordinate()`
	`.setCoordinate( units_0, units_0, units_0, new DoubleVector( new double[] {1.0, 0.0, 0.0} ));`

External Force: the type of property `Position` was changed from `VectorPhysicalQuantity` to `Coordinate`	

`externalForce_0`
`.getPosition()`
`.setComponents(1.0, 0.0, 0.0);`
	`Units units_0 = ((Units) simulation_0`
	`.getUnitsManager()`
	`.getObject("m"));`
	`externalForce_0`
	`.getPositionAsCoordinate()`
	`.setCoordinate( units_0, units_0, units_0, new DoubleVector( new double[] {1.0, 0.0, 0.0} ));`

`externalForce_0`
`.getPosition()`
`.setVector( new DoubleVector( new double[] {1.0, 0.0, 0.0} ));`
	`Units units_0 = ((Units) simulation_0`
	`.getUnitsManager()`
	`.getObject("m"));`

Previous Release	STAR-CCM+ 6.06.011

	`externalForce_0`
	`.getPositionAsCoordinate()`
	`.setCoordinate( units_0, units_0, units_0, new DoubleVector( new double[] {1.0, 0.0, 0.0} ));`

For safety reasons, the method `getPosition()` has been declared as private in both classes `PropulsionForce` and `ExternalForce`. A compile error will be issued if this method is still used. The method `getPositionAsCoordinate().setCoordinate(...)` should be used instead.	

Change to Class Name for VOF Multiphase Model

If you activate the VOF Multiphase model from a macro, you will need to activate the `SegregatedVofFlowModel` instead of the `SegregatedFlowModel`.

Previous Release	STAR-CCM+ 6.06.011
`physicsContinuum_0 .enable( SegregatedVofModel.class);` `physicsContinuum_0 .enable( SegregatedFlowModel.class);`	`physicsContinuum_0 .enable( SegregatedVofModel.class);` `physicsContinuum_0 .enable( SegregatedVofFlowModel.class);`

Activating Solidification-Melting

The Solidification-Melting model is no longer activated on a physics continuum, but on a phase interaction.

Previous Release	STAR-CCM+ 6.06.011
`physicsContinuum_0 .enable(VofEnthalpyMeltingModel.class);`	`phaseInteraction_1 .enable(VofEnthalpyMeltingModel.class);`

Activating Carman-Kozeny Mushy Permeability

The Carman-Kozeny Mushy Permeability model is no longer activated on a physics continuum, but on a phase interaction.

Previous Release	STAR-CCM+ 6.06.011
`physicsContinuum_0 .enable(MushyZoneCarmanKozenyPermeabilityModel.class);`	`phaseInteraction_1 .enable(MushyZoneCarmanKozenyPermeabilityModel.class);`

Setting Latent Heat, Liquidus and Solidus Temperatures

The Latent Heat, Liquidus Temperature and Solidus Temperature are no longer set for a material within the Eulerian Phase manager, but for the multiphase mixture within the Phase Interaction manager.

Previous Release	STAR-CCM+ 6.06.011
SinglePhaseLiquidModel singlePhaseLiquidModel_0 = eulerianPhase_1 .getModelManager() .getModel(SinglePhaseLiquidModel.class); SinglePhaseLiquid singlePhaseLiquid_0 = ((SinglePhaseLiquid) singlePhaseLiquidModel_0 .getMaterial()); ConstantMaterialPropertyMethod constantMaterialPropertyMethod_0 = ((ConstantMaterialPropertyMethod) singlePhaseLiquid_0 .getMaterialProperties() .getMaterialProperty(LatentHeatProperty.class) .getMethod()); constantMaterialPropertyMethod_0 .getQuantity() .setValue(4800000.0);	PhaseInteractionMaterialModel phaseInteractionMaterialModel_0 = phaseInteraction_1 .getModelManager() .getModel(PhaseInteractionMaterialModel.class); PhaseInteractionMaterial phaseInteractionMaterial_0 = ((PhaseInteractionMaterial) phaseInteractionMaterialModel_0 .getMaterial()); ConstantMaterialPropertyMethod constantMaterialPropertyMethod_0 = ((ConstantMaterialPropertyMethod) phaseInteractionMaterial_0 .getMaterialProperties() .getMaterialProperty(LatentHeatProperty.class) .getMethod()); constantMaterialPropertyMethod_0 .getQuantity() .setValue(4800000.0);
ConstantMaterialPropertyMethod constantMaterialPropertyMethod_1 = ((ConstantMaterialPropertyMethod) singlePhaseLiquid_0 .getMaterialProperties() .getMaterialProperty(LiquidusTemperatureProperty.class) .getMethod()); constantMaterialPropertyMethod_1.getQuantity().setValue(300.0);	ConstantMaterialPropertyMethod constantMaterialPropertyMethod_1 = ((ConstantMaterialPropertyMethod) phaseInteractionMaterial_0 .getMaterialProperties() .getMaterialProperty(LiquidusTemperatureProperty.class) .getMethod()); constantMaterialPropertyMethod_1 .getQuantity().setValue(300.0);
ConstantMaterialPropertyMethod constantMaterialPropertyMethod_2 = ((ConstantMaterialPropertyMethod) singlePhaseLiquid_0 .getMaterialProperties() .getMaterialProperty(SolidusTemperatureProperty.class) .getMethod());	ConstantMaterialPropertyMethod constantMaterialPropertyMethod_2 = ((ConstantMaterialPropertyMethod) phaseInteractionMaterial_0 .getMaterialProperties() .getMaterialProperty(SolidusTemperatureProperty.class)

Previous Release	STAR-CCM+ 6.06.011
`constantMaterialPropertyMethod_2 .getQuantity().setValue(400.0);`	`.getMethod());` `constantMaterialPropertyMethod_2 .getQuantity().setValue(400.0);`

Change of Class Name for Solid Stress Solver

The class name for the solid stress solver is now SolidStressIterativeSolver.

Previous Release	STAR-CCM+ 6.06.011

`SolidStressSolver solidStressSolver_0 = ((SolidStressSolver) simulation_0`
`.getSolverManager()`
`.getSolver(`
`SolidStressSolver.class));`
	`SolidStressIterativeSolver solidStressIterativeSolver_0 = ((SolidStressIterativeSolver) simulation_0`
	`.getSolverManager()`
	`.getSolver( SolidStressIterativeSolver.class));`

Changes to the Battery Module

The process of generating battery module objects has changed since the previous release. The previous single action has been broken into three individual actions.

Previous Release	STAR-CCM+ 6.06.011

`batteryModule_0.generate();`	`batteryModule_0 .generateParts();`
	`batteryModule_0 .generateConnectorParts();`
	`batteryModule_0 .generateRegions( new NeoObjectVector( new Object[] {}), true);`

Changes to OLGA Coupling

For OLGA coupled simulations the dynamic viscosity of an Eulerian phase is set to use the OLGA Dynamic Viscosity method by default. If you wish to set a constant dynamic viscosity, you will need to choose this material property method before setting the constant value.

Previous Release	STAR-CCM+ 6.06.011
`ConstantMaterialPropertyMethod constantMaterialPropertyMethod_0 = ((ConstantMaterialPropertyMethod) singlePhaseGas_1 .getMaterialProperties() .getMaterialProperty(DynamicViscosityProperty.class) .getMethod())`	`SinglePhaseLiquid singlePhaseLiquid_0 = ((SinglePhaseLiquid) singlePhaseLiquidModel_0 .getMaterial());` `singlePhaseLiquid_0 .getMaterialProp`

Previous Release	STAR-CCM+ 6.06.011
; constantMaterialPropertyMethod_0 .getQuantity() .setValue(1.704E-5);	erties() .getMaterialProperty(DynamicViscosityProperty.class) .setMethod(ConstantMaterialPropertyMethod.class); ConstantMaterialPropertyMethod constantMaterialPropertyMethod_0 = ((ConstantMaterialPropertyMethod) singlePhaseLiquid_0 .getMaterialProperties() .getMaterialProperty(DynamicViscosityProperty.class) .getMethod()); constantMaterialPropertyMethod_0 .getQuantity() .setValue(1.704E-5);

Macro API Changes 6.04

In STAR-CCM+ 6.04, the macro API changed for Multiphase Segregated flow, Eulerian Multiphase turbulence, surface tension, cavitation, boiling, VOF, 3D-CAD, Abaqus co-simulation, thermal comfort, and OLGA coupling.

Multiphase Interactions for Multiphase Segregated Flow (Eulerian Multiphase)

The interface for setting up Eulerian phase interactions has changed. In particular the pair of phases represented by a single phase interaction are no longer set on the phase interaction object; they are set on the Eulerian Phase Interaction Model object. This is exemplified below.

Previous Release	STAR-CCM+ 6.04.014

EulerianPhasePair eulerianPhasePair_0 = multiPhaseInteractionModel_0 .createPhasePair();
eulerianPhasePair_0 .setPresentationName("Liquid Bubble Interactions");
eulerianPhasePair_0 .setContinuousPhase(eulerianPhase_0);
eulerianPhasePair_0 .setDispersedPhase(eulerianPhase_1);
eulerianPhasePair_0 .setDispersedPhase(eulerianPhase_1);
	PhaseInteraction eulerianPhaseInteraction_0 = multiPhaseInteractionModel_0 .createPhaseInteraction();
	eulerianPhaseInteraction_0 .enable(EulerianPhaseInteractionModel.class);
	EulerianPhaseInteractionModel phaseInteractionModel = eulerianPhaseInteraction_0 .getModelManager() .getModel(EulerianPhaseInteractionModel.class);
	phaseInteractionModel .setContinuousPhase(eulerianPhase_0);
	phaseInteractionModel .setDispersedPhase(eulerianPhase_1);

Change to Turbulence Model Selection for Eulerian Multiphase

Previous Release	STAR-CCM+ 6.04.014
`physicsContinuum_0 .enable(TurbulentModel.class);`	`physicsContinuum_0 .enable( MultiPhaseTurbulentModel.class);`

Activating the Surface Tension Model

The surface tension model is no longer set on a physics continuum, but on a phase interaction.

Previous Release	STAR-CCM+ 6.04.014

`continuum 0 .enable( SurfaceTensionModel.class) ;`	`MultiPhaseInteractionModel multiPhaseInteractionModel_0 = continuum 0 .getModelManager() .getModel( MultiPhaseInteractionModel .class);`
	`PhaseInteraction phaseInteraction_0 = multiPhaseInteractionModel_0 .createPhaseInteraction();`
	`phaseInteraction_0 .enable( VofPhaseInteractionModel.class);`
	`phaseInteraction_0 .enable( SurfaceTensionModel.class);`
	`phaseInteraction_0 .enable( PhaseInteractionMaterialModel .class);`

Set Surface Tension Coefficient

Previous Release	STAR-CCM+ 6.04.014
`ConstantMaterialPropertyMethod constantMethod_0 = ((ConstantMaterialPropertyMethod) ((SurfaceTensionProperty) singlePhaseLiquid_0 .getMaterialProperties() .getMaterialProperty( SurfaceTensionProperty.class)) .getMethod( )); constantMethod 0 .getQuantity() .setValue(0.0010);`	`PhaseInteractionMaterial phaseIntMaterial_0 = ((PhaseInteractionMaterial) phaseIntMaterialModel_0 .getMaterial());` `ConstantMaterialPropertyMethod constantMethod_0 = ((ConstantMaterialPropertyMethod) phaseIntMaterial_0 .getMaterialProperties() .getMaterialProperty( SurfaceTensionProperty.class) .getMethod()) ;`

Previous Release	STAR-CCM+ 6.04.014
	`constantMethod_0 .getQuantity() .setValue(0.082);`

Set Contact Angle

The contact angle is set on boundaries when the surface tension model is active. Boundaries have a manager object, PhaseConditions, on which such phase specific quantities are set.

Previous Release	STAR-CCM+ 6.04.014

`ContactAngleProfile contactAngleProfile_0 = ~ boundary_0 .getValues() .get(ContactAngleProfile.class);`
`contactAngleProfile_0 .getMethod( ConstantScalarProfileMethod.class) .getQuantity() .setValue(3.2);`
	`PhaseConditions phaseConditions_0 = ((PhaseConditions) boundary_0 .get(PhaseConditionsManager.class) .getPhaseConditions("Phase Interaction 1"));`
	`ContactAngleProfile contactAngleProfile_0 = phaseConditions_0 .getPhaseValueManager() .get(ContactAngleProfile.class);`
	`contactAngleProfile_0 .getMethod(ConstantScalarProfileMethod.class) .getQuantity() .setValue(0.33);`

Activating the Cavitation Model

This is now set on a phase interaction, rather than on the physics continuum as previously.

Previous Release	STAR-CCM+ 6.04.014
`physicsContinuum_0 .enable(CavitationModel.class);`	`PhaseInteraction phaseInteraction_0 = ((PhaseInteraction) multiPhaseInteractionModel_0 .getPhaseInteractionManager() .getPhaseInteraction( "Phase Interaction 1"));` `phaseInteraction_0 .enable(CavitationModel.class);`

Activating the Boiling Model

This is now set on a phase interaction, rather than on the physics continuum as previously.

Previous Release	STAR-CCM+ 6.04.014
`physicsContinuum_0 .enable(BoilingTopModel.class);`	`phaseInteraction_0 .enable(BoilingTopModel.class);`

Previous Release	STAR-CCM+ 6.04.014
`physicsContinuum_0.enable(VofBoilingModel.class);`	`phaseInteraction_0.enable(VofBoilingModel.class);`

Specify the Phase Pair (VOF Phase Interaction)

The pair of interacting phases is set on the VofPhaseInteractionModel.

Previous Release	STAR-CCM+ 6.04.014

Not applicable, as only one phase pair could be treated in the previous release.	`VofPhaseInteractionModel vofPhaseInteractionModel_0 = phaseInteraction_0.getModelManager().getModel(VofPhaseInteractionModel.class);`
	`vofPhaseInteractionModel_0.setPrimaryPhase(eulerianPhase_1);`
	`vofPhaseInteractionModel_0.setSecondaryPhase(eulerianPhase_0);`

3D-CAD Rotate and Mirror Body Actions

Axes and positions are now set using `CadModelCoordinate` objects. Some examples of code fragments are given below.

Previous Release	STAR-CCM+ 6.04.014
Setting the feature axis is now done through a `CadModelCoordinate`.	

`rotateBodyFeature_1.getAxis().setComponents(0.0, 0.0, 1.0);`
`rotateBodyFeature_1.getAxis().setUnits(units_0);`
	`CadModelCoordinate cadModelCoordinate_0 = rotateBodyFeature_1.getAxis();`
	`cadModelCoordinate_0.setCoordinateSystem(cartesianCoordinateSystem_0);`
	`cadModelCoordinate_0.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {0.0, 0.0, 1.0}));`

The position is also set using a `CadModelCoordinate` object.	
`Coordinate coordinate_1 = mirrorBodyFeature_0.getPosition();` `coordinate_1.setCoordinateSystem(cartesianCoordinateSystem_0);`	`CadModelCoordinate cadModelCoordinate_2 = mirrorBodyFeature_0.getPosition();` `cadModelCoordinate_2.setCoordinateSystem(cartesianCoordinateSystem_0);`

Previous Release	STAR-CCM+ 6.04.014
`coordinate_1.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {0.4, 0.4, 0.0}));`	`cadModelCoordinate_2.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {0.4, 0.4, 0.0}));`

3D-CAD Circular and Linear Patterns

Changes have been made to the way that the number of copies, the direction vector, and the position coordinates are set. Examples are given below for a circular pattern.

Previous Release	STAR-CCM+ 6.04.014
`bodyCircularPattern_1.setNumberCopy(5);`	`bodyCircularPattern_1.getNumberCopy().setValue(5.0);`

`bodyCircularPattern_1.setDirection(new DoubleVector(new double[] {1.0, 0.0, 0.0}));`	`CadModelCoordinate cadModelCoordinate_3 = bodyCircularPattern_1.getDirection();`
	`cadModelCoordinate_3.setCoordinateSystem(cartesianCoordinateSystem_0);`
	`cadModelCoordinate_3.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {1.0, 0.0, 0.0}));`

`Coordinate coordinate_2 = bodyCircularPattern_1.getPosition();`
`coordinate_2.setCoordinateSystem(cartesianCoordinateSystem_0);`
`coordinate_2.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {1.0, 0.0, 0.0}));`
	`CadModelCoordinate cadModelCoordinate_4 = bodyCircularPattern_1.getPosition();`
	`cadModelCoordinate_4.setCoordinateSystem(cartesianCoordinateSystem_0);`
	`cadModelCoordinate_4.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {1.0, 0.0, 0.0}));`

3D-CAD Extrusion

Should you choose to set a specified direction, the direction vector is now set as a `CadModelCoordinate`.

Previous Release	STAR-CCM+ 6.04.014
`extrusionMerge_1`	`CadModelCoordinate cadModelCoordinate_0 =`

Previous Release	STAR-CCM+ 6.04.014

`.setDirection(new DoubleVector(new double[] {0.0, 1.0, 0.5}));`	`extrusionMerge_1.getDirectionAxis();`
	`cadModelCoordinate_0.setCoordinateSystem(cartesianCoordinateSystem_0);`
	`cadModelCoordinate_0.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {0.0, 1.0, 0.0}));`

3D-CAD Move Body

The translation vector of a Move Body feature is now set using a `CadModelCoordinate`.

Previous Release	STAR-CCM+ 6.04.014

`Coordinate coordinate_2 = moveBodyFeature_0.getTranslationVector();`
`coordinate_2.setCoordinateSystem(cartesianCoordinateSystem_0);`
`coordinate_2.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {0.0, 1.0, 0.0}));`
	`CadModelCoordinate cadModelCoordinate_2 = moveBodyFeature_0.getTranslationVector();`
	`cadModelCoordinate_2.setCoordinateSystem(cartesianCoordinateSystem_0);`
	`cadModelCoordinate_2.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {0.0, 1.0, 0.0}));`

3D-CAD Transform Sketch Plane

The angle properties of the Transform Sketch Plane are set using a simplified function.

Previous Release	STAR-CCM+ 6.04.014
`transformSketchPlane_0.getAngleYZ().setValue(30.0);` `transformSketchPlane_0.getAngleZX().setValue(-10.0);`	`transformSketchPlane_0.getAngle().setComponents(30.0, -10.0, 0.0);`

Abaqus Co-simulation

Include paths for ABAQUS co-simulation modules have changed.

Previous Release	STAR-CCM+ 6.04.014
In general, `common` or `abaqus` have been added to the paths. The particular form required will depend on the classes you want to use.	

Previous Release	STAR-CCM+ 6.04.014
`import star.cosimulation.*;`	`import star.cosimulation.common.*;` `import star.cosimulation.abaqus.*;`
When referencing individual classes:	
`import star.cosimulation.CoSimulationManager;`	`import star.cosimulation.common.CoSimulationManager;`
`import star.cosimulation.CoSimulationModel;`	`import star.cosimulation.common.CoSimulationModel;`
In a similar way the Abaqus related classes are prefixed by `abaqus`:	
`import star.cosimulation.AbaqusCoSimulation;`	`import star.cosimulation.abaqus.AbaqusCoSimulation;`

Thermal Comfort Wizard

Previous Release	STAR-CCM+ 6.04.014
For the ThermalComfortSolver class:	
`solver.getMaximumTemperatureChange().setValues();`	`solver.setMaximumTemperatureChange();`
For the TcmBoundary class:	

`tcmboundary.setPresentationName();`
// has to be set after
`tcmboundary.setTcmBoundaryName();`	

OLGA Coupling

`OlgaZone` used to be a `ClientServerObject` carrying a field called `OlgaZoneName`. Now `OlgaZone` is a named object, and no longer has a `OlgaZoneName` field. The name is changed by calling `setPresentationName()` on the zone object. The default name for the newly created `OlgaZone` is also changed.

Previous Release	STAR-CCM+ 6.04.014
`OlgaZone olgaZone_0 = ((OlgaZone) olgaNode_0.getOlgaZoneManager().getObject("OLGA Zone:NewOlgaZone"));` `olgaZone_0.setOlgaPosName("POS-OUTLET");`	`olgaNode_0.getOlgaZoneManager().addOlgaZone();` `OlgaZone olgaZone_0 = ((OlgaZone) olgaNode_0.getOlgaZoneManager().getObject("NewOlgaZone"));`

Previous Release	STAR-CCM+ 6.04.014

`olgaZone_0 .setOlgaZoneName("OUTLET" );`
`OlgaZone olgaZone_1 = ((OlgaZone) olgaNode_0 .getOlgaZoneManager() .get Object( "OLGA Zone:OUTLET"));`
`olgaZone_1 .setOlgaZoneName("OUTLET" );`
`OlgaZone olgaZone_2 = ((OlgaZone) olgaNode_0 .getOlgaZoneManager() .get Object( "OLGA Zone:OUTLET"));`
`olgaZone_2 .setCcmBoundary(boundary_ 2);`
`OlgaZone olgaZone_3 = ((OlgaZone) olgaNode_0 .getOlgaZoneManager() .get Object( "OLGA Zone:NewOlgaZone"));`
	`olgaZone_0 .setOlgaPosName("POS-OUTLET");`
	`olgaZone_0 .setPresentationName("OUT LET");`
	`OlgaZone olgaZone_1 = ((OlgaZone) olgaNode_0 .getOlgaZoneManager() .get Object("OUTLET"));`
	`olgaZone_1 .setPresentationName("OUT LET");`
	`OlgaZone olgaZone_2 = ((OlgaZone) olgaNode_0 .getOlgaZoneManager() .get Object("OUTLET"));`
	`olgaZone_2 .setCcmBoundary(boundary_ 2);`
	`olgaNode_0 .getOlgaZoneManager() .add OlgaZone();`
	`OlgaZone olgaZone_3 = ((OlgaZone) olgaNode_0 .getOlgaZoneManager() .get Object("NewOlgaZone"));`

Macro API Changes 6.02

In STAR-CCM+ 6.02, the macro API changed for Abaqus co-simulation, CAE importing, 3D-CAD, and combustion.

Abaqus Co-simulation

1. Abaqus is now launched automatically by the co-simulation. It will be necessary to remove the command for launching Abaqus from macros recorded in STAR-CCM+ 5.06. An example would be:

   ```
   abaqusCoSimulation_0.start();
   ```

2. The "Synchronization Type" option is no longer available in the Co-Simulation tool. Therefore all macro commands relating to this option will need to be removed. Examples are:

   ```
   abaqusCoSimulation_0   .getSynchronizationTypeOptions()   .setSelect
   ed(     AbaqusSynchronizationTypeOptions.LOCK_STEP   );
   ```

   and

   ```
   abaqusCoSimulation_0   .getSynchronizationTypeOptions()   .setSelect
   ed(     AbaqusSynchronizationTypeOptions.SUB_CYCLE   );
   ```

3. The macro commands for setting the "Pressure Ramping Parameters" and "Traction Clipping Parameters" have changed.

Previous Release	STAR-CCM+ 6.02.007
`abaqusCoSimulation_0 .setPressureRampParams( "1.0, 2.0, 3.0" );`	`abaqusCoSimulation_0 .setPressureRampingParams( new DoubleVector( new double[] { 1.0, 2.0, 3.0 } ) );`
`abaqusCoSimulation_0 .setTractionClipParams( "6.0, 5.0, 4.0" );`	`abaqusCoSimulation_0 .setTractionClippingParams( new DoubleVector( new double[] { 6.0, 5.0, 4.0 } ) );`

CAE Import/Mapping/Export

1. The macro command for importing CAE geometry (both as an "Imported Model" and as a STAR-CCM+ region) has changed. It is also necessary to import `star.cae.common.*` in Java macros containing these actions.

Previous Release	STAR-CCM+ 6.02.007
	`import star.cae.common.*;`
`ImportManager importManager_0 = simulation_0.getImportManager();`	`CaeImportManager caeImportManager_0 = simulation_0.get(CaeImportManager .class);`

2. The macro command for importing data from CAE models has changed.

Previous Release	STAR-CCM+ 6.02.007

`simulation_0`
`.get(ImportedModelManager.class)`
`.importVertexDataForImportedSurface(`
`new NeoObjectVector(`
`new Object[] { importedSurface_0, importedSurface_1, importedSurface_2} ),`
`"Displacement",`
`"analytical_disp.odb",`
`"abaqus",`
`"Step-1",`
`"",`
`units_1`
	`caeImportManager_0 .importAbaqusData( resolvePath("analytical_disp.odb"),`
	`new NeoObjectVector(`
	`new Object[] { importedModel_0 }`
	`),`
	`1,`
	`Displacement", 1,`
	`"abaqus",`
	`"Step-1",`
	`"ALLMODES", units_1 );`

Previous Release	STAR-CCM+ 6.02.007
`);`	

3. The names of the Imported variables have changed:
   - The prefix given to field function names for imported CAE models has been changed from “Imported Surface” to “Imported”. For example, if Temperature is imported from an Abaqus model, its field function will now be called `ImportedTemperature`, as opposed to `ImportedSurfaceTemperature`.
   - The prefix given to mapped field functions has changed to `MappedImported` instead of `MappedImportedSurface`.

   It may be necessary to update any Java macros that explicitly rely on these field function names.

3D-CAD

There is a change in the function that sets the bodies to skip for the series of objects that handle linear patterns, linear pattern cuts, circular patterns and circular pattern cuts. The new function takes a vector of integer numbers that define the body indices to skip in the pattern.

Previous Release	STAR-CCM+ 6.02.007
`BodyLinearPattern* .setSkipFlags( new BooleanVector( new boolean[] { true, false,` `true, false } ) )`	`BodyLinearPattern*. .setSkipPositio n( new IntVector( new int[] {1, 3} ) )`
Similarly for objects beginning with `BodyCircularPattern*`	Similarly for objects beginning with `BodyCircularPattern*`

Reacting Flows - PPDF

Previous Release	STAR-CCM+ 6.02.007
`PpdfFlameletPartiallyPremixedModel`	`PpdfAdiabaticFlameletPartiallyPremi xedModel`

Reacting Flows - Dars-CFD

Previous Release	STAR-CCM+ 6.02.007
The `DarsCfdReactionBaseModel` class had the following methods:	These methods are now moved to the `DarsCfdProperties` class and should be replaced as shown:
`model.getATol()`	`model .getDarsCfdProperties() .getATo l()`
`model.setATol()`	`model .getDarsCfdProperties() .setATo l()`

Previous Release	STAR-CCM+ 6.02.007
`model.getRTol()`	`model .getDarsCfdProperties() .getRTol()`
`model.setRTol()`	`model .getDarsCfdProperties() .setRTol()`
`model.getTLimitLow()`	`model .getDarsCfdProperties() .getTLimitLow()`
`model.setTLimitLow()`	`model .getDarsCfdProperties() .setTLimitLow()`
`model.getTLimitHigh()`	`model .getDarsCfdProperties() .getTLimitHigh()`
`model.setTLimitHigh()`	`model .getDarsCfdProperties() .setTLimitHigh()`

Macro API Changes 5.06

In STAR-CCM+ 5.06, the macro API changed for scene and plot updates, and emissions modeling.

Changes to various items within the macro API are detailed below. Macros created using previous releases of the software will not necessarily run in STAR-CCM+ 5.06.010 if any of the items in the Previous Release columns below have been accessed.

Update Frequency for Scenes and Plots

The method for specifying update frequency has changed, in order to accommodate the delta time method. Previously, Update Frequency, which is only meaningful for number of iterations or number of time steps, was just a property of the SceneUpdate and PlotUpdate objects. Now, the specific object related to each type of update should be obtained from the PlotUpdate object.

Previous Release	STAR-CCM+ 5.06.010

`PlotUpdate plotUpdate_0 = xYPlot_0.getPlotUpdate();`
`plotUpdate_0.setUpdateMode(1);`
`// Iterations Update Policy`
`plotUpdate_0.setUpdateFrequency(10);`
	`PlotUpdate plotUpdate_0 = xYPlot_0.getPlotUpdate();`
	`plotUpdate_0.setUpdateMode(1);`
	`// Iterations Update Policy`
	`IterationUpdateFrequency iterationUpdateFrequency_0 = plotUpdate_0 .getIterationUpdateFrequency();`
	`iterationUpdateFrequency_0 .setIterations(10);`

	`plotUpdate_0.setUpdateMode(2);`

Previous Release	STAR-CCM+ 5.06.010
	`// Time Step Update Policy TimeStepUpdateFrequency timeStepUpdateFrequency_0 = plotUpdate_0 .getTimeStepUpdateFrequency(); timeStepUpdateFrequency_0 .setTimeSteps(5);`
	`plotUpdate_0.setUpdateMode(3); // Delta Time Update Policy DeltaTimeUpdateFrequency deltaTimeUpdateFrequency_0 = plotUpdate_0 .getDeltaTimeUpdateFrequency(); deltaTimeUpdateFrequency_0 . getDeltaTime() .setValue(1.1);`

Activating PCFM and NOx Emissions

In previous releases, activating PCFM and NOx Emissions had the effect of turning on NOx PPDF automatically. However, this is not required in the case of PCFM Emissions as PCFM already has a PPDF solver. Previous macros would have had the line:

```
physicsContinuum_0.enable(NoxPpdfCombustionModel.class);
```

This line is no longer required for PCFM Emissions.

Macro API Changes 5.04

In STAR-CCM+ 5.04, the macro API changed for the materials database, NASA polynomials, surface repair, DFBI, and clearing solutions.

Changes to various items within the macro API are detailed below. Macros created using previous releases of the software will not necessarily run in STAR-CCM+ 5.04.006 if any of the items in the Previous Release columns below have been accessed.

Materials Database

The function `setPath()` on `MaterialDataBase` is no longer valid. The purpose of this command was to allow STAR-CCM+ to automatically import dbs files from the paths in the list. These files must now be converted in to the new `.mdb` format, as explained in the user guide.

NASA Polynomials

In previous versions of STAR-CCM+, users setting thermodynamic polynomial data through a macro had to first multiply the coefficients by $R_u/M_r(X)$, where R_u is the universal gas constant [J/(kmol K)] and $M_r(X)$ is the molecular weight [kg/kmol] of the substance X. From STAR-CCM+ 5.04.006, this scaling is not required: coefficients should be entered as they are found in the literature.

Any existing macros should be revised to account for this change.

Automatic Surface Repair Option

The following methods have been removed from class AutomaticSurfaceRepair:

```
getDoFixPiercedFaces
```

```
setDoFixPiercedFaces
```

```
setDoFixCloseProximityFaces
```

```
setDoFixPoorQualityFaces
```

Also, the behavior of the following methods has changed as these now return values based on their settings in the Surface Remesher panel:

```
getDoFixCloseProximityFaces
```

```
getDoFixPoorQualityFaces
```

DFBI Linear Spring Force

The main functionality of linear springs has been transferred from the Linear Spring Force (located in the External Force and Moment Manger of a 6-DOF body) to the new Linear Spring Coupling (in the new Body Coupling Manager located beneath the DFBI parent top-level node). This means that Java code related to linear springs has changed. Macros as implemented in the previous release are given below, along with their latest counterparts.

Previous Release	STAR-CCM+ 5.04.006

`// Create Linear Spring Force`
`Body body_0 =`
`((Body)`
`simulation_0 .get(BodyManager.class)`
`.getObject("Boat"));`
	`// Create Linear Spring Coupling`
	`LinearSpringCoupling`
	`linearSpringCoupling_0 =`

Previous Release	STAR-CCM+ 5.04.006

LinearSpringForce linearSpringForce_0 =
body_0 .getExternalForceAndMomentManager() .createExternalForceAndMoment(LinearSpringForce.class);
// Set position of first end point
linearSpringForce_0 .getPosition1() .setComponents(1.0, 0.0, 0.0);
// Set position of second end point
linearSpringForce_0 .getPosition2() .setComponents(2.0, 0.0, 0.0);
// Set remaining properties
linearSpringForce_0 .getSpringConstant() .setValue(20.0);
linearSpringForce_0 .getRelaxationLength() .setValue(5.0);
	simulation_0 .get(SixDofBodyCouplingManager.class) .createSixDofBodyCoupling(LinearSpringCoupling.class);
	// Set position of first end point
	Coordinate coordinate_0 =
	.linearSpringCoupling_0.getPosition1();
	Units units_0 =
	((Units) simulation_0 .getUnitsManager() .getObject("m"));
	coordinate_0.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {1.0, 0.0, 0.0}));
	// Set position of second end point
	Coordinate coordinate_1 =
	linearSpringCoupling_0.getPosition2();
	coordinate_1.setCoordinate(units_0, units_0, units_0, new DoubleVector(new double[] {2.0, 0.0, 0.0});

linearSpringForce_0.setEnabled(false);
linearSpringForce_0.setNoRepellingForce(true);
// Set coordinate system of second end point
LabCoordinateSystem labCoordinateSystem_0 =
((LabCoordinateSystem) simulation_0 .getCoordinateSystemManager() .getObject("Laboratory"));
CartesianCoordinateSystem cartesianCoordinateSystem_0 =
	// Set remaining properties
	linearSpringCoupling_0 .getSpringConstant() .setValue(20.0);
	linearSpringCoupling_0 .getRelaxationLength() .setValue(5.0);
	linearSpringCoupling_0.setEnabled(false);
	linearSpringCoupling_0.setNoRepellingForce(true);
	// Set coordinate system of second end point
	LabCoordinateSystem labCoordinateSystem_0 =

Previous Release	STAR-CCM+ 5.04.006

`((CartesianCoordinateSystem) labCoordinateSystem_0 .getLocalCoordinateSystemManager() .getObject("Cartesian 1"));`
`linearSpringForce_0 .setCoordinateSystem(cartesianCoordinateSystem_0);`
	`((LabCoordinateSystem) simulation_0 .getCoordinateSystemManager() .getObject("Laboratory"));`
	`CartesianCoordinateSystem cartesianCoordinateSystem_0 =`
	`((CartesianCoordinateSystem) labCoordinateSystem_0 .getLocalCoordinateSystemManager() .getObject("Cartesian 1"));`
	`coordinate_1 .setCoordinateSystem(cartesianCoordinateSystem_0);`

Note that the former Linear Spring Forces still exist, but they are managed entirely by their corresponding Linear Spring Couplings. The creation, copying, deleting and renaming of Linear Spring Forces via macros is no longer possible. Creation and deletion is done automatically by selecting or deselecting a specific body in the corresponding Linear Spring Coupling. Copying is done by copying the corresponding Linear Spring Coupling. There is no substitute for renaming because the presentation name of the Linear Spring Force is constructed automatically by the Linear Spring Coupling.

DFBI Spring Elongation Reports

Following on from the previous section, spring elongation reports have also changed.

Previous Release	STAR-CCM+ 5.04.006

`Body body_0 =`
`((Body) simulation_0 .get(BodyManager.class) .getObject("Boat"));`
`LinearSpringForce linearSpringForce_0 =`
`((LinearSpringForce) body_0 .getExternalForceAndMomentManager() .getObject("Linear Spring Force 1"));`
`springElongationReport_0 .setLinearSpringForce(linearSpringForce_0);`
	`LinearSpringCoupling linearSpringCoupling_0 =`
	`((LinearSpringCoupling) simulation_0 .get(SixDofBodyCouplingManager.class) .getObject("Linear Spring 1"));`
	`springElongationReport_0 .setLinearSpringCoupling(linearSpringCoupling_0);`

New Framework for Clearing Solutions

The revised syntax for clearing solutions is shown below.

Previous Release	STAR-CCM+ 5.04.006

`Solution solution_0 =`
`simulation_0.getSolution();`
`solution_0.clearSolution();`
`solution_0.resetForRestart();`
	`Solution solution_0 =`
	`simulation_0.getSolution();`
	`//`
	`// clear history, clear fields and reset mesh`
	`solution_0.clearSolution();`
	`solution_0 .clearSolution(Solution.Clear.History, Solution.Clear.Fields);`
	`// replaces reset for restart`
	`solution_0.clearSolution(Solution.Clear.History);`
	`// clear history + reset the mesh`
	`solution_0.clearSolution(Solution.Clear.History, Solution.Clear.Mesh);`

Macro API Changes 5.02

In STAR-CCM+ 5.02, the macro API changed for motion and reports (Body Force and Body Moment).

Changes to various items within the macro API are detailed below. Macros created using previous releases of the software will not necessarily run in STAR-CCM+ 5.02.009 if any of the items in the Previous Release columns below have been accessed.

Motion Changes

Significant changes have been implemented in motion models, reference frames and coordinate systems. These changes are reflected both in the GUI and in the Java macro API. Macros written for previous versions of STAR-CCM+ will no longer work in the current release. It will be necessary for users to re-write their macros according to the following guide.

There are two steps in general:

- Create the new motion or reference frame and set its properties
- Apply the motion or reference frame object on the new region value, Motion Specification.

Create Rigid Body Motions

`// Rotation`
`RotatingMotion rotatingMotion_0 =`
`simulation_0.get(MotionManager.class) .createMotion(RotatingMotion.class, "Rotation");`

Create Rigid Body Motions

```
// Rotation and Translation
RotatingAndTranslatingMotion rotatingAndTranslatingMotion_0 =
simulation_0.get(MotionManager.class) .createMotion( RotatingAndTranslat
ingMotion.class, "Rotation and Translation");
```

Create Morphing Motion

```
MorphingMotion morphingMotion_0 =
simulation_0.get(MotionManager.class) .createMotion(MorphingMotion.class
, "Morphing");
```

DFBI (6-DOF) Motions

```
// DFBI Rotation and Translation
SixDofMotion sixDofMotion_0 =
simulation_0.get(MotionManager.class) .createMotion( SixDofMotion.class,
"DFBI Rotation and Translation");
// DFBI Embedded Rotation
SixDofEmbeddedMotion sixDofEmbeddedMotion_0 =
simulation_0.get(MotionManager.class) .createMotion( SixDofEmbeddedMotio
n.class, "DFBI Embedded Rotation");
// DFBI Morphing
SixDofMorphingMotion sixDofMorphingMotion_0 =
simulation_0.get(MotionManager.class) .createMotion( SixDofMorphingMotio
n.class, "DFBI Morphing");
// DFBI Superposed Rotation
SixDofPlusRotatingMotion sixDofPlusRotatingMotion_0 =
simulation_0.get(MotionManager.class) .createMotion( SixDofPlusRotatingM
otion.class, "DFBI Superposed Rotation");
```

Create Harmonic Balance Flutter Motion

```
HarmonicBalanceFlutterMotion harmonicBalanceFlutterMotion_0 =
simulation_0.get(MotionManager.class) .createMotion( HarmonicBalanceFlut
terMotion.class, "Harmonic Balance Flutter");
```

Create Solid Displacement Motion

```
LargeDisplacementMotion largeDisplacementMotion_0 =
```

Create Solid Displacement Motion

```
simulation_0.get(MotionManager.class) .createMotion( LargeDisplacementMo
tion.class, "Solid Displacement");
```

Assign Motion to a Region

```
// Use rotatingMotion_0 as example
MotionSpecification motionSpecification_0 =
region_0.getValues().get(MotionSpecification.class);
motionSpecification_0.setMotion(rotatingMotion_0);
```

A similar process is followed for reference frames.

Create and Assign a Reference Frame

```
UserRotatingReferenceFrame userRotatingReferenceFrame_0 =
simulation_0.get(ReferenceFrameManager.class) .createReferenceFrame( Use
rRotatingReferenceFrame.class, "Rotating");
userRotatingReferenceFrame_0 .getRotationRate() .setValue(3000.0);
MotionSpecification motionSpecification_0 =
region_0.getValues().get(MotionSpecification.class);
motionSpecification_0.setMotion(rotatingMotion_0);
motionSpecification_0 .setReferenceFrame(userRotatingReferenceFrame_0);
```

Coordinate systems make use of the new local coordinate system manager.

Create a Local Coordinate System

```
LabCoordinateSystem labCoordinateSystem_0 =
((LabCoordinateSystem) simulation_0.
getCoordinateSystemManager() .getObject("Laboratory"));
CartesianCoordinateSystem cartesianCoordinateSystem_0 =
labCoordinateSystem_0 .getLocalCoordinateSystemManager() .createLocalCoo
rdinateSystem( CartesianCoordinateSystem.class, "Cartesian");
CylindricalCoordinateSystem cylindricalCoordinateSystem_0 =
labCoordinateSystem_0 .getLocalCoordinateSystemManager() .createLocalCoo
rdinateSystem( CylindricalCoordinateSystem.class, "Cylindrical");
// Create a nested spherical system
SphericalCoordinateSystem sphericalCoordinateSystem_0 =
cartesianCoordinateSystem_0 .getLocalCoordinateSystemManager() .createLo
calCoordinateSystem( SphericalCoordinateSystem.class,
```

Create a Local Coordinate System

```
"Spherical");
```

Body Force and Body Moment Reports

Previous Release

```
// Body Force Report
BodyForceReport bodyForceReport_0 =
((BodyForceReport) simulation_0.getReportManager() .getReport("Active
Force X"));
Body body_0 =
((Body) simulation_0.get(BodyManager.class) .getObject("Body 1"));
bodyForceReport_0.setBody(body_0);
bodyForceReport_0 .getBodyForceReportOption() .setSelected(BodyForceReport
Option.ACTIVE_FORCE);
// Alternative 1
bodyForceReport_0 .getBodyForceReportOption() .setSelected(BodyForceReport
Option.RAWCFD_FORCE);
// Alternative 2
bodyForceReport_0 .getBodyForceReportOption() .setSelected(BodyForceReport
Option.RAW_FORCE);
// Body Moment Report
BodyMomentReport bodyMomentReport_0 =
((BodyMomentReport) simulation_0.getReportManager() .getReport("Active
Moment X"));
bodyMomentReport_0.setBody(body_0);
bodyMomentReport_0 .getBodyMomentReportOption() .setSelected(BodyMomentRep
ortOption.ACTIVE_MOMENT);
// Alternative
bodyMomentReport_0 .getBodyMomentReportOption() .setSelected(BodyMomentRep
ortOption.RAW_MOMENT);
```

For reasons of compatibility, the above methods are still available, but marked as deprecated. The new way of setting the group of forces or moments, and the use of raw forces, is shown below.

STAR-CCM+ 5.02.009

```
// Setting the group of forces in the Body Force Report
BodyForceReport bodyForceReport_0 =
```

STAR-CCM+ 5.02.009

((BodyForceReport) simulation_0 .getReportManager() .getReport("Active Force X"));
Body body_0 =
((Body) simulation_0.get(BodyManager.class).getObject("Body 1"));
ExternalFluidForceAndMoment externalFluidForceAndMoment_0 =
((ExternalFluidForceAndMoment) body_0 .getExternalForceAndMomentManager() .getObject("Fluid Force and Moment 1"));
ExternalGravityForce externalGravityForce_0 =
((ExternalGravityForce) body_0 .getExternalForceAndMomentManager() .getObject("Gravity Force 1"));
LinearSpringForce linearSpringForce_0 =
((LinearSpringForce) body_0 .getExternalForceAndMomentManager() .getObject("Linear Spring Force 1"));
bodyForceReport_0.setBody(body_0);
bodyForceReport_0 .getForces() .setObjects(externalFluidForceAndMoment_0,
externalGravityForce_0,
linearSpringForce_0);
bodyForceReport_0.setUseRawForces(true);

// Setting the group of moments in the Body Moment Report
BodyMomentReport bodyMomentReport_0 =
((BodyMomentReport) simulation_0 .getReportManager() .getReport("Active Moment X"));
bodyMomentReport_0.setBody(body_0);
bodyMomentReport_0 .getMoments() .setObjects(externalFluidForceAndMoment_0, linearSpringForce_0);
bodyMomentReport_0.setUseRawMoments(false);

Macro API Changes 4.06

In STAR-CCM+ 4.06, the macro API changed for passive scalar, thermal porosity, and cavitation.

Changes to various items within the macro API are detailed below. Macros created using previous releases of the software will not run in STAR-CCM+ 4.06.011 if any of the items in the Previous Release column below have been accessed.

	Previous Release	STAR-CCM+ 4.06.011

Passive Scalar	`physContinuum_0 .enable( PassiveScalarModel.class);`	`physContinuum_0 .enable( PassiveScalarModel.class);`
		`PassiveScalarModel pScalMod_0 = physContinuum_0 .getModelManager() .getModel(PassiveScalarModel.class);`
		`PassiveScalarMaterial pScalMat_0 =`
		`pScalMod_0 .getPassiveScalarManager() .createPassiveScalarMaterial(PassiveScalarMaterial.class);`

Region Thermal Porosity
This class has been removed and replaced by tensor profiles PorousConductivityProfile and PorousSpecificHeatProfile
	`RegionThermalPorosity rts = region .getValues() .get(RegionThermalPorosity .class);`
	`rts.setPorousConductivity(0.0260305);`
	`rts.setPorousSpecificHeat(1023.3);`
		`PorousConductivityProfile pCondProf = region .getValues() .get(PorousConductivityProfile.class);`
		`pCondProf .setMethod(IsotropicTensorProfileMethod.class);`
		`ScalarProfile scalProf = ((IsotropicTensorProfileMethod) pCondProf .getMethod()) .getIsotropicProfile();`
		`((ConstantScalarProfileMethod) scalProf .getMethod()) .getQuantity() .setValue(0.0260305);`
		`PorousSpecificHeatProfile porousSpecHeatProf = region .getValues() .get(PorousSpecificHeatProfile.class);`

		`porousSpecHeatProf .setMethod(IsotropicTensorProfileMethod.class);`
		`ScalarProfile scalProf = ((IsotropicTensorProfileMethod) porousSpecificHeatProfile.getMethod()) .getIsotropicProfile();`
		`((ConstantScalarProfileMethod) scalProf .getMethod()) .getQuantity() .setValue(1023.3);`

Saturation Pressure for Cavitation Model	`cavitationModel_0 .getSaturationPressure() .setValue(-97000.0);`	`ConstantMaterialPropertyMethod conMatPropMethod_0 = ((ConstantMaterialPropertyMethod) singlePhaseLiquid_0 .getMaterialProperties() .getMaterialProperty(SaturationPressureProperty.class) .getMethod());` `conMatPropMethod_0 .getQuantity() .setValue(2338.0);`

Macro API Changes 4.04

In STAR-CCM+ 4.04, the macro API changed for Flow-Split Outlet boundaries, VOF, system reports, porous energy, 6-DOF bodies, and the Turbo Wizard.

Changes to various items within the macro API are detailed below. Macros created using previous releases of the software will not run in STAR-CCM+ 4.04.011 if any of the items in the Previous Release column below have been accessed.

	Previous Release	STAR-CCM+ 4.04.011
Split Ratio in Flow-Split Outlet	`f.setSplitFactor(dv);`	`f.getSplitFactor() .setValue(dv);`
	where `f` is the `OutletSplitFactor`.	

HTCxArea in the VOF Boiling model	`vofBoilingModel_0 .getHTCxArea() .setValue(5000045.0);`	`VofBoilingHTCxArea vofBoilingHTCxArea_0 = vofBoilingModel_0 .getVofBoilingHTCxArea();`
		`VofBoilingHTCxAreaConstantMethod vofBoilingHTCxAreaConstantMethod_0 = vofBoilingHTCxArea_0 .getHTCxAreaConstantMethod();`
		`vofBoilingHTCxAreaConstantMethod_0 .getHTCxArea() .setValue(5000556.0);`

IterationElapsedTimeReport and IterationCpuTimeReport	`IteratorCpuTimeReport iteratorCpuTimeReport_0 = ((IteratorCpuTimeReport) simulation_0 .getReportManager() .getReport("Solver CPU Time per Iteration"));`	`IteratorCpuTimeReport iteratorCpuTimeReport_0 = ((IteratorCpuTimeReport) simulation_0 .getReportManager() .getReport("Solver Iteration CPU Time"));`
	`IteratorElapsedTimeReport iteratonElapsedTimeReport_0 = ((IteratorElapsedTimeReport) simulation_0 .getReportManager() .getReport("Solver Elapsed Time per Iteration"));`	`IteratorElapsedTimeReport iteratorElapsedTimeReport_0 = ((IteratorElapsedTimeReport) simulation_0 .getReportManager() .getReport("Solver Iteration Elapsed Time"));`
Porous energy specification	`RegionThermalPorosity rts = region.getValues() .get(RegionThermalPorosity.class);` `rts .setPorousConductivity(0.0260305);`	`PorousConductivityProfile porousConductivityProfile = region .getValues() .get( PorousConductivityProfile.class);`

		porousConductivityProfile.setMethod(IsotropicTensorProfileMethod.class); ScalarProfile scalarProfile = ((IsotropicTensorProfileMethod) porousConductivityProfile.getMethod()).getIsotropicProfile(); ((ConstantScalarProfileMethod) scalarProfile.getMethod()).getQuantity().setValue(0.0260305);
	RegionThermalPorosity rts = region.getValues().get(RegionThermalPorosity.class); rts.setPorousSpecificHeat(1023.3);	PorousSpecificHeatProfile porousSpecificHeatProfile = region.getValues().get(PorousSpecificHeatProfile.class); porousSpecificHeatProfile.setMethod(IsotropicTensorProfileMethod.class); ScalarProfile scalarProfile = ((IsotropicTensorProfileMethod) porousSpecificHeatProfile.getMethod()).getIsotropicProfile(); ((ConstantScalarProfileMethod) scalarProfile.getMethod()).getQuantity().setValue(1023.3);
6-DOF Body Properties	Body body_0 = ((Body) simulation_0.get(BodyMa	Body body_0 = ((Body) simulation_0.get(BodyMa

The former 6-DOF body properties ExternalForce, ExternalLocalForce, ExternalMoment and ExternalLocalMoment have been removed. Instead, an ExternalForceAndMomentManager has been introduced which can be used to add or delete different kinds of forces, including the old ones. As a consequence, access to the forces/moments via macros has changed.	`nager.class) .getObject( "ship"));`	`nager.class) .getObject( "ship"));`

	`// 1) Set external force:`
	`body_0.getExternalForce() .setVector( new DoubleVector( new double[] {1.0, 0.0, 0.0}));`
		`// 1) Set external force:`
		`ExternalForceCM externalForceCM_0 = body_0 .getExternalForceAndMomentManager() .createExternalForceAndMoment( ExternalForceCM.class );`
		`externalForceCM_0 .getForce() .setComponents(1.0 , 0.0, 0.0);`

	`// 2) Set external local force:`
	`body_0 .getExternalLocalForce() .setVector( new DoubleVector( new double[] {2.0, 0.0, 0.0}));`
		`// 2) Set external local force:`
		`ExternalForceCM externalForceCM_1 = body_0 .getExternalForceAndMomentManager() .createExternalForceAndMoment( ExternalForceCM.class );`
		`externalForceCM_1 .getForce() .setComponents(2.0 , 0.0, 0.0);`
		`CartesianCoordinateSystem cartesianCoordinateSystem_0 = ((CartesianCoordinateSystem) simulation_0 .getCoordinateSystemManager() .getObject("ship-CSys"));`
		`externalForceCM_1 .setCoordinateSystem( cartesianCoordinateSystem_0);`

	`// 3) Set external moment:`	`// 3) Set external moment:`

	`body_0.getExternalMoment().setVector(new DoubleVector(new double[] {3.0, 0.0, 0.0}));`	`ExternalMomentCM externalMomentCM_0 = body_0.getExternalForceAndMomentManager().createExternalForceAndMoment(ExternalMomentCM.class);` `externalMomentCM_0.getMoment().setComponents(3.0, 0.0, 0.0);`

	`// 4) Set external local moment:`
	`body_0.getExternalLocalMoment().setVector(new DoubleVector(new double[] {4.0, 0.0, 0.0}));`
		`// 4) Set external local moment:`
		`ExternalMomentCM externalMomentCM_1 = body_0.getExternalForceAndMomentManager().createExternalForceAndMoment(ExternalMomentCM.class);`
		`externalMomentCM_1.getMoment().setComponents(4.0, 0.0, 0.0);`
		`externalMomentCM_1.setCoordinateSystem(cartesianCoordinateSystem_0);`

Turbo Wizard	Macros involving the `importTurboBlade` method on the `ImportManager` should be re-recorded using the Turbo Wizard, as several changes have been made to the argument list for this method.	

Macro API Changes 4.02

In STAR-CCM+ 4.02, the macro API changed for combustion, the materials database, passive scalar, and LES turbulence.

- Combustion

  `DarsCfdReactionModel` is now called `DarsCfdReactionBaseModel`.

- Material Database

  Names and symbols were not consistent with the convention for other materials in `props.dbs`. Fixing this issue means that macros that were using the names as symbols will fail.

  For example, the symbol `Carbon Steel` has now been changed to `UNSG101000`, and the name for the material is now `Carbon Steel`.

Before:

```
Solid solid_1 = (Solid) solidModel_0.replaceMaterial(solid_0,
((DataBaseSolid) materialDataBase_0.getSolids().getObject("Carbon
Steel")));
```

After:

```
Solid solid_1 = (Solid) solidModel_0.replaceMaterial(solid_0,
((DataBaseSolid)
materialDataBase_0.getSolids().getObject("UNSG101000")));
```

- Passive Scalar

  Previous versions of passive scalar required inputting the molecular diffusivity and turbulent Schmidt number in the gas or liquid material properties panel, now it is done within passive scalar. This also allows passive scalar to now work for multi-component flows.

  Before:

```
// STAR-CCM+ macro: oldprop.java
package macro;

import java.util.*;
import star.turbulence.*;
import star.material.*;
import star.common.*;
import star.base.neo.*;
import star.flow.*;

public class oldprop extends StarMacro {

  public void execute() {
    Simulation simulation_0 = getActiveSimulation();
    PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum)
simulation_0.getContinuumManager().getContinuum("Physics 1"));
    SingleComponentGasModel singleComponentGasModel_0 =
physicsContinuum_0.getModelManager().getModel(SingleComponentGasMode
l.class);
    Gas gas_0 = ((Gas) singleComponentGasModel_0.getMaterial());
    SchmidtNumberDiffusivityMethod schmidtNumberDiffusivityMethod_0
= ((SchmidtNumberDiffusivityMethod)

gas_0.getMaterialProperties().getMaterialProperty(MolecularDiffusivi
tyProperty.class).getMethod());
    schmidtNumberDiffusivityMethod_0.setSchmidtNumber(0.111);
    ConstantMaterialPropertyMethod constantMaterialPropertyMethod_0
= ((ConstantMaterialPropertyMethod)

gas_0.getMaterialProperties().getMaterialProperty(TurbulentSchmidtNu
mberProperty.class).getMethod());
```

```
    constantMaterialPropertyMethod_0.getQuantity().setValue(0.222);
  }
}
```

After:

```
// STAR-CCM+ macro: newprop.java
package macro;

import java.util.*;
import star.material.*;
import star.common.*;
import star.base.neo.*;
import star.flow.*;
import star.passivescalar.*;

public class newprop extends StarMacro {
  public void execute() {
    Simulation simulation_0 = getActiveSimulation();
    PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum)
simulation_0.getContinuumManager().getContinuum("Physics 1"));
    PassiveScalarModel passiveScalarModel_0 =
physicsContinuum_0.getModelManager().getModel(PassiveScalarModel.cla
ss);
    PassiveScalarMaterial passiveScalarMaterial_0 =
passiveScalarModel_0.getPassiveScalarMaterial();
    SchmidtNumberDiffusivityMethod schmidtNumberDiffusivityMethod_0
= ((SchmidtNumberDiffusivityMethod)

passiveScalarMaterial_0.getMaterialProperties().getMaterialProperty(
PassiveScalarDiffusivityProperty.class).getMethod());
    schmidtNumberDiffusivityMethod_0.setSchmidtNumber(0.111);
    ConstantMaterialPropertyMethod constantMaterialPropertyMethod_0
= ((ConstantMaterialPropertyMethod)

passiveScalarMaterial_0.getMaterialProperties().getMaterialProperty(
PassiveScalarTurbulentSchmidtNumberProperty.class).getMethod());
    constantMaterialPropertyMethod_0.getQuantity().setValue(0.222);
  }
}
```

- LES

  Order of the field functions has changed

  Before:

```
  public void setInitialCondition() {
    Simulation simulation_0 =
      getActiveSimulation();
```

```
    PhysicsContinuum physicsContinuum_0 =
      ((PhysicsContinuum)
simulation_0.getContinuumManager().getContinuum("Physics 1"));

    TurbulenceIntensityProfile turbulenceIntensityProfile_0 =
!      ((TurbulenceIntensityProfile)
physicsContinuum_0.getInitialConditions().get(TurbulenceIntensityPro
file.class));


turbulenceIntensityProfile_0.setMethod(FunctionScalarProfileMethod.c
lass);

    UserFieldFunction userFieldFunction_0 =
      ((UserFieldFunction)
simulation_0.getFieldFunctionManager().getFunction("UserFieldFunctio
n_1"));

!    ((FunctionScalarProfileMethod)
turbulenceIntensityProfile_0.getMethod()).setFieldFunction(userField
Function_0);

    TurbulentLengthScaleProfile turbulentLengthScaleProfile_0 =
!      ((TurbulentLengthScaleProfile)
physicsContinuum_0.getInitialConditions().get(TurbulentLengthScalePr
ofile.class));


turbulentLengthScaleProfile_0.setMethod(FunctionScalarProfileMethod.
class);

-    UserFieldFunction userFieldFunction_1 =
-      ((UserFieldFunction)
simulation_0.getFieldFunctionManager().getFunction("UserFieldFunctio
n_3"));
-
!    ((FunctionScalarProfileMethod)
turbulentLengthScaleProfile_0.getMethod()).setFieldFunction(userFiel
dFunction_1);

    VelocityProfile velocityProfile_0 =
      ((VelocityProfile)
physicsContinuum_0.getInitialConditions().get(VelocityProfile.class)
);

    velocityProfile_0.setMethod(CompositeVectorProfileMethod.class);

    ScalarProfile scalarProfile_0 =
```

```
      ((CompositeVectorProfileMethod)
velocityProfile_0.getMethod()).getProfile(0);

    scalarProfile_0.setMethod(FunctionScalarProfileMethod.class);

    UserFieldFunction userFieldFunction_2 =
      ((UserFieldFunction)
simulation_0.getFieldFunctionManager().getFunction("UserFieldFunctio
n_2"));

    ((FunctionScalarProfileMethod)
scalarProfile_0.getMethod()).setFieldFunction(userFieldFunction_2);
  }
```

After:

```
  public void setInitialCondition() {
    Simulation simulation_0 =
      getActiveSimulation();

    PhysicsContinuum physicsContinuum_0 =
      ((PhysicsContinuum)
simulation_0.getContinuumManager().getContinuum("Physics 1"));

+
physicsContinuum_0.getInitialConditions().get(PseudoTurbulenceSpecOp
tion.class).setSelected(PseudoTurbulenceSpecOption.INTENSITY_LENGTH_
SCALE);
+
    TurbulenceIntensityProfile turbulenceIntensityProfile_0 =
!
physicsContinuum_0.getInitialConditions().get(TurbulenceIntensityPro
file.class);


turbulenceIntensityProfile_0.setMethod(FunctionScalarProfileMethod.c
lass);

    UserFieldFunction userFieldFunction_0 =
+      ((UserFieldFunction)
simulation_0.getFieldFunctionManager().getFunction("UserFieldFunctio
n_3"));
+
+    ((FunctionScalarProfileMethod)
turbulenceIntensityProfile_0.getMethod()).setFieldFunction(userField
Function_0);
+
+    UserFieldFunction userFieldFunction_1 =
```

```
      ((UserFieldFunction)
simulation_0.getFieldFunctionManager().getFunction("UserFieldFunctio
n_1"));

!    ((FunctionScalarProfileMethod)
turbulenceIntensityProfile_0.getMethod()).setFieldFunction(userField
Function_1);

    TurbulentLengthScaleProfile turbulentLengthScaleProfile_0 =
!
physicsContinuum_0.getInitialConditions().get(TurbulentLengthScalePr
ofile.class);


turbulentLengthScaleProfile_0.setMethod(FunctionScalarProfileMethod.
class);

!    ((FunctionScalarProfileMethod)
turbulentLengthScaleProfile_0.getMethod()).setFieldFunction(userFiel
dFunction_0);

    VelocityProfile velocityProfile_0 =
      ((VelocityProfile)
physicsContinuum_0.getInitialConditions().get(VelocityProfile.class)
);

    velocityProfile_0.setMethod(CompositeVectorProfileMethod.class);

    ScalarProfile scalarProfile_0 =
      ((CompositeVectorProfileMethod)
velocityProfile_0.getMethod()).getProfile(0);

    scalarProfile_0.setMethod(FunctionScalarProfileMethod.class);

    UserFieldFunction userFieldFunction_2 =
      ((UserFieldFunction)
simulation_0.getFieldFunctionManager().getFunction("UserFieldFunctio
n_2"));

    ((FunctionScalarProfileMethod)
scalarProfile_0.getMethod()).setFieldFunction(userFieldFunction_2);
  }
```

Macro API Changes 3.04

In STAR-CCM+ 3.04, the macro API changed for reflectivity, volume shapes, and meshing.

- **Specified Reflectivity Now Has Auto-Calculate as Default**

  Specified reflectivity has changed for macros. Old macros that explicitly set reflectivity need to be revised. With the auto-calculate method as the default, a constant reflectivity cannot be set in a macro unless the method type is first changed to `constant`. The easiest thing to do is to remove the setting of reflectivity from any macros; the reflectivity will then be calculated automatically.

- **Volume Shapes Now Recognize Local Coordinate Systems**

  Volume shapes have been enhanced to allow the use of the local coordinate system and user-specified units.

  To accommodate these changes, a new `Coordinate` class has been designed which has a reference to the local coordinate system and knows the units of its components. This class replaces the pre-existing `VectorPhysicalQuantity` class. This change affects the following data of each volume shape:

  - `BrickVolumeShape` - `Corner1` and `Corner2`
  - `CylinderConeVolumeShape` - `Start Coordinate` and `End Coordinate`
  - `SphereVolumeShape` - `Origin`

  So when you access these data elements of the volume shape in the macro, changes might be required in the macro depending on the way the data is accessed:

  The macro need not be changed if it accesses the data as follows:
  `brickVolumeShape_0.getCorner1().setVector(new DoubleVector(new double[] {0.1, 0.0108, 0.0}));`

  If the macro accesses the data and type casts it to `VectorPhysicalQuantity`, and calls its methods `setVector` or `getVector`, then it is necessary to replace the type cast of `VectorPhysicalQuantity` with the `Coordinate` class:

  `// old - ((VectorPhysicalQuantity) brickVolumeShape_0.getCorner2()).setVector(new DoubleVector(new double[] {0.2, 0.0, 0.0}));`

  `// new - ((Coordinate) brickVolumeShape_0.getCorner2()).setVector(new DoubleVector(new double[] {0.2, 0.0, 0.0}));`

- **Volumetric Controls Can Be Enabled Per Meshing Model**

  Volumetric Controls (previously known as Volume Sources) have an additional step of specifying which meshing models the control applies to and the size is a generic property of the control.

  A 3.02 macro would look like:

```
// create VolumeSource VolumeSource volumeSource_0 =
meshContinuum_0.getVolumeSources().createVolumeSource();
// stick shape in the source SphereVolumeShape sphereVolumeShape_0
= ((SphereVolumeShape)
simulation_0.get(VolumeShapeManager.class).getObject("Sphere 1"));
volumeSource_0.getVolumeShapeGroup().setObjects(sphereVolumeShape_0)
;
// set the size of the volume source (applies to all meshers)
```

```
VolumeSourceSize volumeSourceSize_0 = volumeSource_0.getSize();
GenericRelativeSize genericRelativeSize_0 = ((GenericRelativeSize)
volumeSourceSize_0.getRelativeSize());
genericRelativeSize_0.setPercentage(20.0);
```

On the other hand the exact same macro in 3.04 would require:

```
// create volume source
VolumeSource volumeSource_1 =
meshContinuum_0.getVolumeSources().createVolumeSource();
// enable volume source for poly mesher (new step)
VolumeSourceDualMesherSizeOption volumeSourceDualMesherSizeOption_0
=
volumeSource_1.get(MeshConditionManager.class).get(VolumeSourceDualM
esherSizeOption.class);
volumeSourceDualMesherSizeOption_0.setVolumeSourceDualMesherSizeOpti
on(true);
// enable volume source for surface remesher (new step)
VolumeSourceResurfacerSizeOption volumeSourceResurfacerSizeOption_0
=
volumeSource_1.get(MeshConditionManager.class).get(VolumeSourceResur
facerSizeOption.class);
volumeSourceResurfacerSizeOption_0.setVolumeSourceResurfacerSizeOpti
on(true);
// enable volume source for wrapper (new step)
VolumeSourceSurfaceWrapperSizeOption
volumeSourceSurfaceWrapperSizeOption_0 =
volumeSource_1.get(MeshConditionManager.class).get(VolumeSourceSurfa
ceWrapperSizeOption.class);
volumeSourceSurfaceWrapperSizeOption_0.setVolumeSourceSurfaceWrapper
SizeOption(true);
// set the size for the source (now done through
// MeshValueManager not directly from the source)
VolumeSourceSize volumeSourceSize_0 =
volumeSource_1.get(MeshValueManager.class).get(VolumeSourceSize.clas
s); GenericRelativeSize genericRelativeSize_0 =
((GenericRelativeSize) volumeSourceSize_0.getRelativeSize());
genericRelativeSize_0.setPercentage(20.0);
```

If a user tries to run the 3.02 macro through 3.04 it will work but not do what the user expects – the source will be created and associated with the sphere shape but it will not be activated for any mesh models. The reason that the 3.02 macro will work in 3.04 is because we kept method `VolumeSource::getSize()` around for backward compatibility.

Macro API Changes 3.02

In STAR-CCM+ 3.02, the macro API changed for transmissivity.

The two-sided interface transmissivity may require two changes to macros:

- If you have set the thermal or solar transmissivity on an interface in a macro, you need to change the macro to set the transmissivity on the two associated interface boundaries. An example of the change for thermal transmissivity is:

```
// old - set transmissivity on the interface DirectBoundaryInterface
directBoundaryInterface_0 = ((DirectBoundaryInterface)
simulation_0.getInterfaceManager().getInterface("RegionInterface_1"));
TransmissivityProfile transmissivityProfile_0 =
((TransmissivityProfile)
directBoundaryInterface_0.getValues().get(TransmissivityProfile.class)
); transmissivityProfile_0.setTransmissivity(1.0);
```

```
// new - set transmissivity on the interface boundaries (using a copy
for the second boundary) InterfaceBoundary interfaceBoundary_0 =
((InterfaceBoundary)
region_0.getBoundaryManager().getBoundary("interface-14
[RegionInterface_1]")); InterfaceBoundary interfaceBoundary_1 =
((InterfaceBoundary)
region_1.getBoundaryManager().getBoundary("interface-11
[RegionInterface_1]")); TransmissivityProfile transmissivityProfile_0 =
((TransmissivityProfile)
interfaceBoundary_0.getValues().get(TransmissivityProfile.class));
((ConstantScalarProfileMethod)
transmissivityProfile_0.getMethod()).getQuantity().setValue(1.0);
interfaceBoundary_1.copyProperties(interfaceBoundary_0);
```

- If you have set the thermal transmissivity on any external (non-interface) boundaries for the S2S model, then you need to change the name of the profile from BoundaryTransmissivityProfile to TransmissivityProfile. This does not affect solar S2S or discrete ordinates (DOM does not have transmissivity on external boundaries). An example of the change is:

```
// old BoundaryTransmissivityProfile boundaryTransmissivityProfile_1 =
((BoundaryTransmissivityProfile)
boundary_1.getValues().get(BoundaryTransmissivityProfile.class));
```

```
// new TransmissivityProfile transmissivityProfile_1 =
((TransmissivityProfile)
boundary_1.getValues().get(TransmissivityProfile.class));
```

Macro API Changes 2.10

In STAR-CCM+ 2.10, the macro API changed for stopping criteria and VOF.

Units added to stopping criteria:

- `monitorIterationStoppingCriterionMinLimitType_0.setLimit(1.0E-6);`

  becomes

`monitorIterationStoppingCriterionMinLimitType_0.getLimit().setValue(1.0E-6);`

- `monitorIterationStoppingCriterionAsymptoticType_0.setMaxWidth(0.02);`

  becomes

  `monitorIterationStoppingCriterionAsymptoticType_0.getMaxWidth().setValue(0.02);`

Due to changes in VOF:

- `MultiPhaseMixtureModel` has been removed. It is replaced by `EulerianMultiPhaseModel` and `MultiPhaseMaterialModel`, so where you used to enable `MultiPhaseMixtureModel.class` , now you must enable `EulerianMultiPhaseModel.class` and `MultiPhaseMaterialModel.class`, for example:

  ```
  physicsContinuum_0.enable(MultiPhaseMixtureModel.class);
  ```

  becomes

  ```
  physicsContinuum_0.enable(EulerianMultiPhaseModel.class);
  physicsContinuum_0.enable( MultiPhaseMaterialModel.class);
  ```

- `VofEquationOfStateModel` has been removed. It is replaced by `MultiPhaseEquationOfStateModel`, so where you used to enable `VofEquationOfStateModel.class`, now you must enable `MultiPhaseEquationOfStateModel.class`, for example:

  ```
  physicsContinuum_0.enable(VofEquationOfStateModel.class);
  ```

  becomes

  ```
  physicsContinuum_0.enable(MultiPhaseEquationOfStateModel.class);
  ```

- `MultiPhaseMixture` has been removed. Where you used to access the `MultiPhaseMixture` from the `MultiPhaseMixtureModel` and use `MultiPhaseMixture.addComponent()` to add Gas/LiquidComponent objects to the mixture, now you use `EulerianMultiPhaseModel.createPhase()` to create new EulerianPhase objects.

  ```
  // access EulerianMultiPhaseModel from PhysicsContinuum
  EulerianMultiPhaseModel eulerianMultiPhaseModel_0 =
  physicsContinuum_0.getModelManager().getModel(EulerianMultiPhaseMode
  l.class);
  ```

  ```
  // create phase-0 -- gas Oil Vapor
  ```

```
EulerianPhase eulerianPhase_0 =
eulerianMultiPhaseModel_0.createPhase();
```

```
// create phase-1 -- liquid Oil
EulerianPhase eulerianPhase_1 =
eulerianMultiPhaseModel_0.createPhase();
```

- New `SinglePhaseLiquidModel` and `SinglePhaseGasModel` material models have been created. In each `EulerianPhase` enable either a `SinglePhaseLiquidModel` or a `SinglePhaseGasModel` along with an equation of state model as appropriate for the phase, for example:

```
// phase-0 -- gas Oil Vapor
eulerianPhase_0.enable(SinglePhaseGasModel.class);
eulerianPhase_0.enable(ConstantDensityModel.class);
```

```
// phase-1 -- liquid Oil
eulerianPhase_1.enable( SinglePhaseLiquidModel.class);
eulerianPhase_1.enable(ConstantDensityModel.class);
```

- New `SinglePhaseGas` and `SinglePhaseLiquid` material classes have been created. Where you used to set material properties in each `Gas/LiquidComponent` material of the (now defunct) `MultiPhaseMixture`, now you set material properties in the `SinglePhaseGas/Liquid` material of each phase. The material in each phase is accessed via the material model in that phase, for example:

```
// phase-0 -- gas Oil Vapor
// get SinglePhaseGasModel from ModelManager in phase-0
SinglePhaseGasModel singlePhaseGasModel_0 =
eulerianPhase_0.getModelManager().getModel(SinglePhaseGasModel.class
 );
```

```
// get SinglePhaseGas material from SinglePhaseGasModel in phase-0
SinglePhaseGas singlePhaseGas_0 =  ((SinglePhaseGas)
singlePhaseGasModel_0.getMaterial());
singlePhaseGas_0.setPresentationName("Oil vapor");
```

```
// set value for constant density in phase-0
ConstantMaterialPropertyMethod constantMaterialPropertyMethod_0 =
((ConstantMaterialPropertyMethod) ((ConstantDensityProperty)
singlePhaseGas_0.getMaterialProperties().getMaterialProperty( Consta
ntDensityProperty.class)).getMethod());
constantMaterialPropertyMethod_0.getQuantity().setValue(0.2);
```

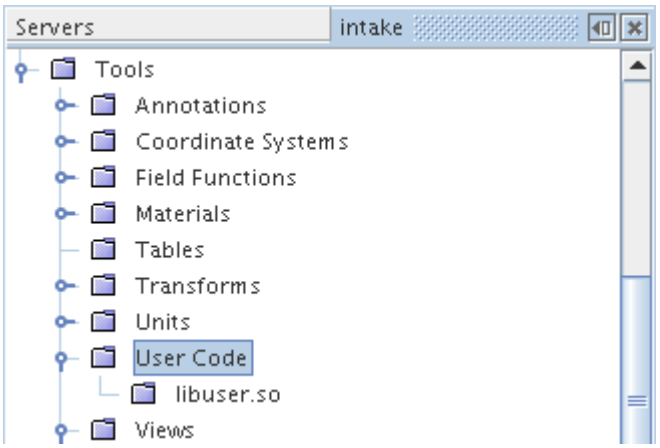
Working with User Code

User code allows STAR-CCM+ to be customized with functions written in a compiled language such as C, C++ or Fortran.

User code takes the form of one or more user libraries that are attached to STAR-CCM+, each of which contains one or more user functions and a library registration function. When a user library has been attached, its user functions are available for use and are provided in drop-down lists for any operation that needs them. You can use these functions to specify values on a region or a boundary, initialization, and visualization.

User functions can only be accessed on a part if the data used to define the user function exists on that part. For the data to exist, it must be allocated on the specific part and must have been initialized.

You can view the status of the user code where the currently attached user libraries are shown, under the **Tools** > **User Code** node.



The **User Code** node, which has a pop-up menu, is the manager object for all user code libraries that have been saved in the simulation.

The pop-up menu has the following option:

New User Library: Executes the create user library action and adds a user code library to the **User Code** node.

It exists even when empty to allow creation of the first user code library. The user code manager can hold any number of user code libraries up to the memory limits of the computer.

User functions and library registration functions can be coded in any language, as long as that language is able to bind like C functions or Fortran subroutines. For convenience, the C language interface and the Fortran language interface are documented separately, although conceptually they are identical. Functions in

other languages should follow one of these interfaces. A list of the variables available from STAR-CCM+ and some examples of user code are also provided.

FORTRAN is not supported on Windows.

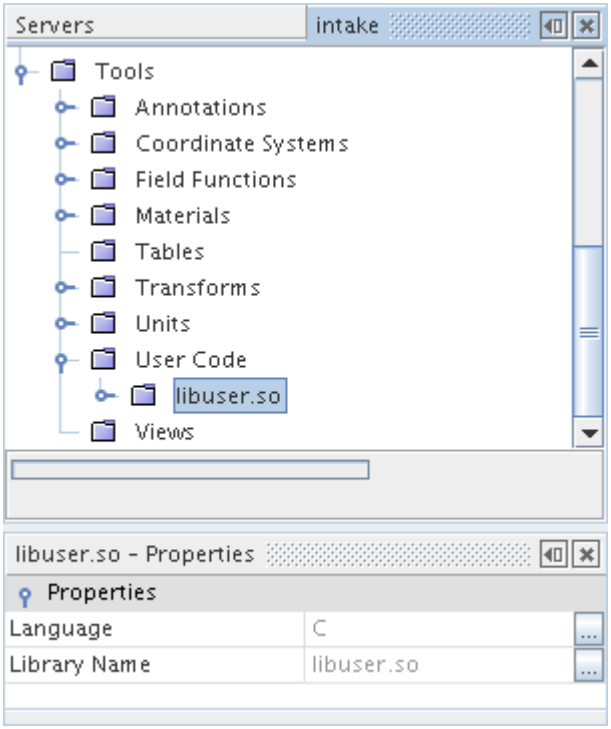
Contents:

What Are User Libraries?
Working With User Functions
Creating a New User Library
Accessing a User Library
User Function Interface Reference (C)
Fortran User Interface Reference
Variables Available to User Functions
User Code Examples
Using User Code on Particles and Parcels

What Are User Libraries?

User libraries are shared-object libraries (files with the platform-dependent extension `.so`, `.sl` or `.dll`) containing one or more user functions and a library registration function.

They are created by compiling and linking the user functions. Each of the user libraries that have been attached to STAR-CCM+ have a node in the simulation tree. Each of these nodes has its own properties and a pop-up menu.



The path to the user library is relative to the location of the simulation file.

A user library that is compiled on one platform is not binary compatible with another platform. For example, you are not able to compile a user library on Windows and then run the simulation on Linux. Compile user libraries for each platform on which you run the simulation.

The *Language* of the library is deduced from the library registration function. It represents the style of binding which the library functions require, not necessarily their language. For example, make C++ functions bind like C.

User Library Properties

These properties determine the link between the user library object in STAR-CCM+ and the file with your user code.

Language	The programming language of the library. STAR-CCM+ determines this language when it reads the library (read-only).

Library Name	The path to the file containing the user coding. This path is relative to the location of the simulation file.

User Library Pop-Up Menu

Reload Library	Executes the reload library action.

Working With User Functions

User functions are functions or subroutines that are written in a compiled language such as C, C++ or Fortran.

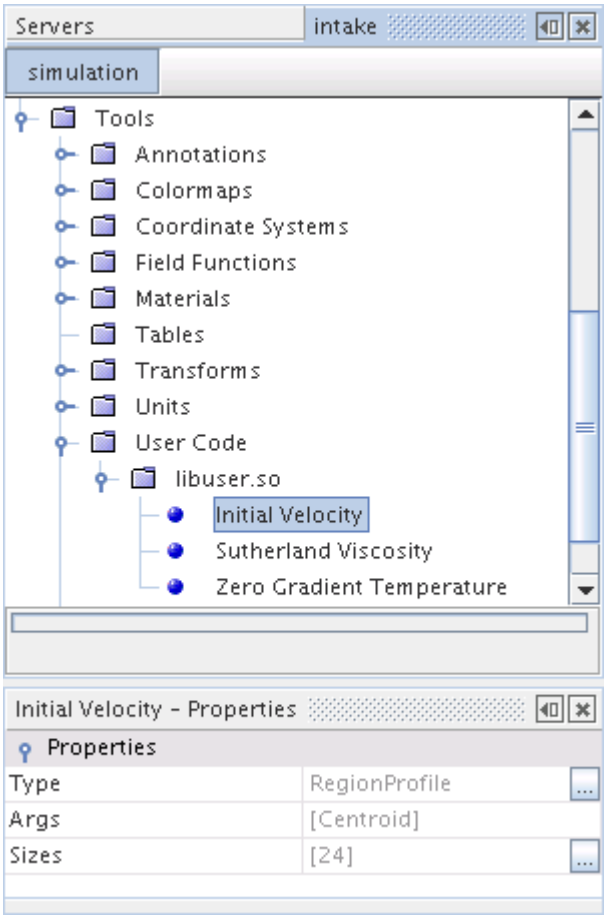
Each function can request arguments from STAR-CCM+ and must return a result consistent with its type. The two principal categories of user functions are:

- User-coded profile
- User-coded field function

The most common use of user functions is the specification of profiles of values on regions or boundaries. Examples are initial conditions, boundary conditions, and source terms. User functions which are registered as scalar field functions or vector field functions can also be used in the same applications as field functions, such as visualization. The principal reason for choosing a user function instead of a field function is power: user functions can implement more complex functionality than field functions.

STAR-CCM+ also allows specific user functions with fixed argument lists, such as user-coded reaction rates for stiff chemistry solvers.

In order for STAR-CCM+ to know which arguments to pass to a user function and what type of result to expect, register the user function using a library registration function. Available user functions can then be viewed in the user library nodes in the **User Code** node under **Tools** in the object tree.



The display name for the user function, together with its *Type*, its *Args*, and their *Sizes*, is obtained from the library registration function.

What Are User-Coded Profiles?

A user-coded profile is a profile that is implemented through a user function. Either of the following are possibilities:

- A boundary profile
- A region profile

Once a user-coded profile has been compiled and its library is loaded, it is available for use in boundary and region definitions, which are shown in drop-down lists of user functions.

What Are User-Coded Field Functions?

A user-coded field function is a field function that is implemented through a user function. As with all field functions, a user-coded field function can be either of the following:

- A scalar field function
- A vector field function

Once a user-coded field function has been compiled and its library is loaded, it is available for use. This availability is the same as with any other field function that is shown in scalar or vector drop-down lists of field functions.

Using User Functions in Parallel Simulations

Every array which is passed into a user function is given a size which depends on the number of relevant entities of the region or boundary on that partition. Depending on the partitioning, some partitions can end up with 0 relevant entities.

Never assume that there is at least one relevant entity present on all partitions. For example, the following code causes a segmentation fault when used in a partition that does not contain any relevant entities:

```
// This will seg fault if the iter array is sized 0

int currentIteration = iter[0];
```

What is a Library Registration Function?

Each user library must contain a library registration function, `uclib` (for C), or `uflib` (for Fortran). STAR-CCM+ calls this function when the user library is first attached. The purpose of calling this function is to:

- Discover which user functions are contained in the library
- Determine which arguments they need
- What type of result they return

Ensure that the library registration functions follow the C template or Fortran template, either by coding in those languages or by replicating their bindings.

Creating a New User Function

Ensure that the user functions follow the C template or Fortran template, either by coding in those languages or by replicating their bindings. For each user function in a library, add code to the library registration function to register it. STAR-CCM+ does not impose restrictions on how many user functions can be contained in a source file, nor on the names of the source files. Once a set of user function source files has been completed, together with a library registration function, the next stage is to create a user library from them.

Using User-Coded Reaction Rates

For the reacting species transport models which use non-stiff numerics to calculate species source from reactions (such as the Eddy Break-Up or Eddy Contact Micromixing models), you can use field functions or user-coded field functions to specify arbitrary reactions. However, for combustion models which use an ODE solver to handle numerical stiffness (in particular the Complex Chemistry and Surface Chemistry models), specific user-coded reaction rate functions must be used. These user functions are only for calculating reaction rates—the function signatures are fixed. See Reacting System Properties.

Creating a New User Library

This section provides information about creating a user library on Linux or Windows.

A user library is created from its constituent user functions and a library registration function.

Creating a User Library on Linux

On Linux, STAR-CCM+ does not require a particular compiler to be used for this process. For a selection of generic compilers, the table below gives example commands to create a library with base-name `user` from C or Fortran source files. Optional compilation flags, such as for optimization, can also be used; consult the compiler manual for more information.

Compiler	Precision	Compilation Command
GCC	mixed	`gcc -fPIC -shared *.c -o libuser.so`
GCC	double	`gcc -DDOUBLE_PRECISION -fPIC -shared *.c -o libuser.so`
GNU Fortran 77	mixed	`g77 -fPIC -shared *.f -o libuser.so`
G95	mixed	`g95 -fPIC -shared *.f -o libuser.so`

Creating a User Library on Windows

FORTRAN is not supported on Windows.

These instructions are for Microsoft Visual C++ 2013. Older versions may not work and if so, in a different manner. Only 64 bit versions are supported on Windows. Link against `UserFunctions.lib` found in the compiler subdirectory of your STAR-CCM+ installation directory, for example:

```
C:\Program Files\CD-adapco\STAR-CCM+10.02.008\star\lib
\win64\intel12.1\lib
```

To create a user library on Windows, follow the steps below:

1. Open a **VS2013 x64 Native Tools Command Prompt** to compile a 64-bit library.
2. Compile the source code into object files using the command:

   ```
   cl /MD /D_WINDOWS -c *.c
   ```

   For `*.cpp` files, substitute `*.c` with `*.cpp`.

   For double precision code, include the compiler option `/DDOUBLE_PRECISION`.
3. Link the objects into a Dynamic Link Library (DLL) using the `UserFunctions.lib` found in the compiler subdirectory of your STAR-CCM+ installation directory:

   ```
   link -dll /out:libuser.dll *.obj "C:\Program Files\CD-adapco\STAR-
   CCM+10.02.008\star\lib\win64\intel12.1\lib
   \UserFunctions.lib"
   ```

4. Check that you have a function that is called `uclib` exported from the DLL. STAR-CCM+ looks for this function in the DLL:

```
dumpbin/exports libuser.dll
```

An example of a dump output is given below:

```
Microsoft (R) COFF/PE Dumper Version 12.00.30501.0
Copyright (C) Microsoft Corporation.  All rights reserved.
Dump of file libuser.dll
File Type: DLL
  Section contains the following exports for libuser.dll

    00000000 characteristics
    54D1FF8D time date stamp Wed Feb 04 11:16:29 2015
        0.00 version
           1 ordinal base
           1 number of functions
           1 number of names
    ordinal hint RVA      name
          1    0 00001000 uclib
  Summary
        1000 .data
        1000 .pdata
        1000 .rdata
        1000 .reloc
        1000 .text
```

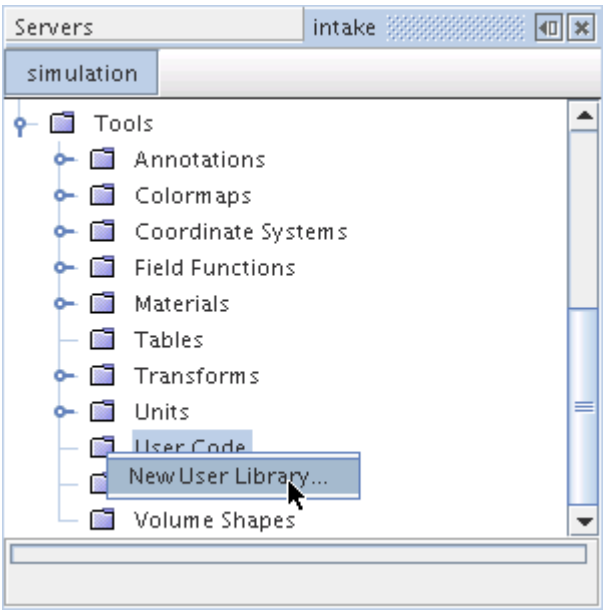
With Visual Studio 2013, you do not need to embed a manifest file. The DLL that is built has a dependency on the visual studio runtime (`MSVCR120.DLL`). The visual studio runtime must be stored on disk so that STAR-CCM+ can find it. Typically, if visual studio is installed, `MSVCR120.DLL` is already on the machine. If not, you can install the re-distributable or simply copy the DLL to the same location as the user coding `.dll`.

Accessing a User Library

This section describes how to load or reload a user library in a STAR-CCM+ simulation.

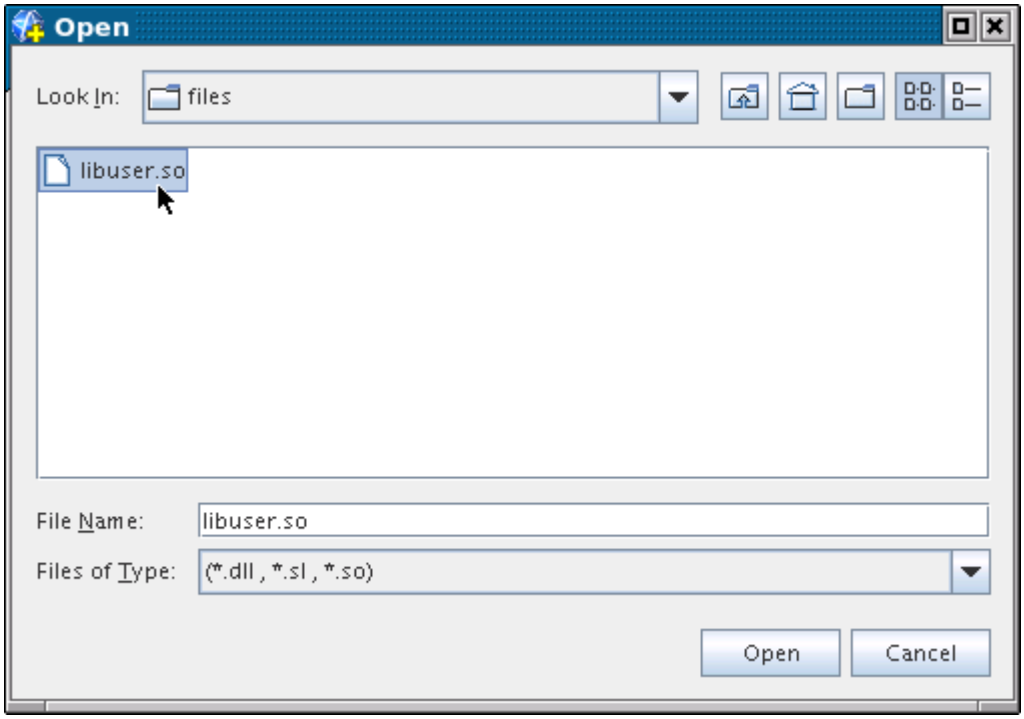
To load a new user library:

1. Right-click on the **User Code** node under the **Tools** object tree, then select **New User Library**.



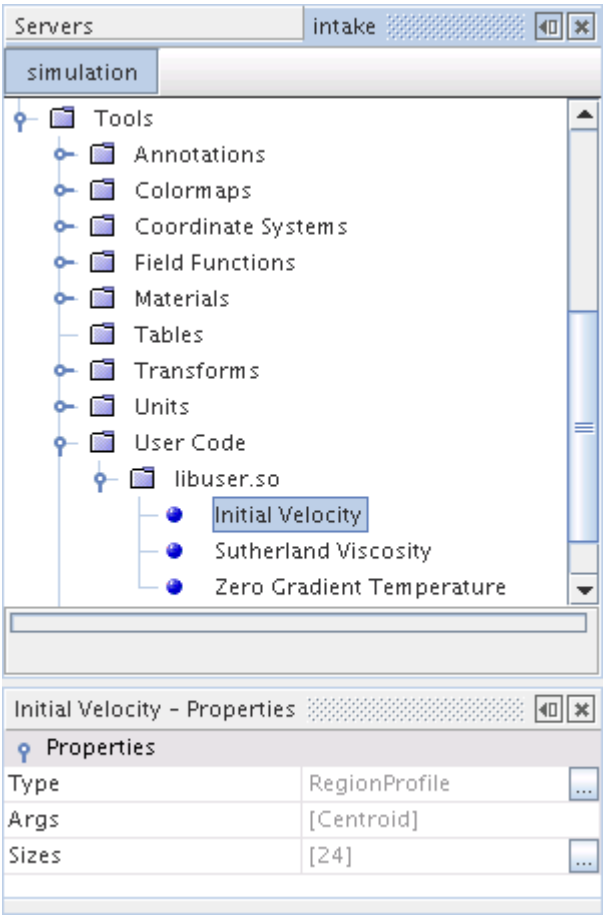
   A standard *Open* dialog appears, with the filter **Files of Type:** set to compatible formats.

2. Navigate to the file and open it.



The path to the user library is relative to the location of the simulation file.

The requested library is loaded and its registration function is invoked to set the *Language* of the user functions. Each user function has a node within its parent user library folder.

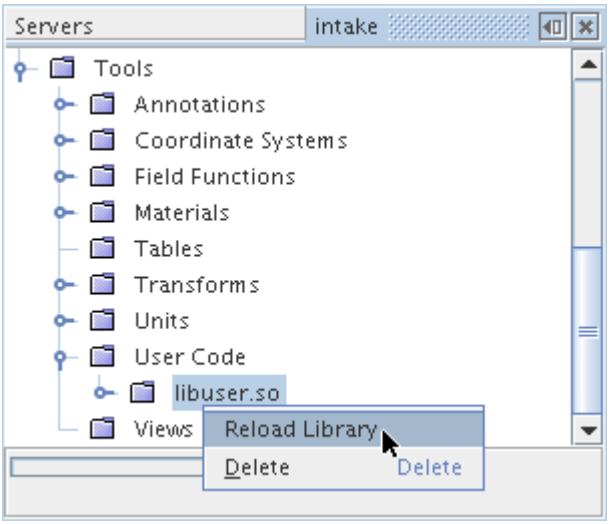


If you are working in parallel, it is also possible to reset the library path with a command line.

Note: User libraries can be reloaded, for example if they have been recompiled since they were last loaded.

To reload a user library:

3. Right-click the node of the user library and select **Reload Library**.



User libraries can be deleted from an existing simulation as long as they are not in use.

To prevent a user library that is in use from being deleted, the **Delete** menu item for that user library is grayed out.

User Function Interface Reference (C)

This section introduces the interface between STAR-CCM+ and C user functions.

The following topics cover the interface between STAR-CCM+ and C user functions:

- Type definitions
- The template user function
- The template library registration function (uclib)
- Function calls in `uclib` to register the user functions (ucfunc and ucarg)
- The variables available from STAR-CCM+

The usage of this interface is illustrated in the examples.

Contents:

Type Definitions (C)
Template User Function (C)
Template Library Registration Function uclib (C)
User Function Name Registration ucfunc (C)
User Function Argument Registration ucarg (C)

Type Definitions (C)

The types of variables and functions that are used in C user functions are defined in the file `uclib.h`.

This file is not part of the standard STAR-CCM+ installation. Create it in the same directory where you are creating your user functions. The contents of `uclib.h` is displayed as shown here:

```
#ifndef UCLIB_H
#define UCLIB_H
#ifdef DOUBLE_PRECISION
typedef double Real;
#else
typedef float Real;
#endif
typedef double CoordReal;

#ifdef __cplusplus
extern "C" {
#endif
#if defined(WIN32) || defined(_WINDOWS) || defined(_WINNT)
# define USERFUNCTION_EXPORT __declspec(dllexport)
# define USERFUNCTION_IMPORT __declspec(dllimport)
#else
```

```
# define USERFUNCTION_EXPORT
# define USERFUNCTION_IMPORT
#endif

extern void USERFUNCTION_IMPORT ucarg(void *, char *, char *, int);
extern void USERFUNCTION_IMPORT ucfunc(void *, char *, char *);
extern void USERFUNCTION_IMPORT ucfunction(void *, char *, char *,
int, ...);

void USERFUNCTION_EXPORT uclib();
#ifdef __cplusplus
}
#endif
#endif
```

Template User Function (C)

This section provides a template that you can use when creating a user function.

The generic purpose of a user function is to populate a `result` array with values obtained by operating on `args`, as shown in the examples, using this template:

```
#include "uclib.h"
void name(result, int size, args ...);
```

`name`: the function name. This name is arbitrary: STAR-CCM+ does not impose particular function names for particular functionality. There are, however, some protected names which must be avoided: `ucarg, ucfunc, ucfunction, uclib, ufarg, uffunc, uflib`.

`result`: the array of values that the user function returns. It is declared as:

```
Real *result
```

for a scalar function, or:

```
Real (*result)[3]
```

for a vector function. Each element of the `result` array represents a cell or face, depending on the type of the user function.

`size`: the number of elements in the `result` array.

`args`: the arguments which STAR-CCM+ passes to the function, which is chosen from the list of available variables and requested through ucarg. The form of declaration for an array `arg` with elements of various types is given in the table below.

Elemental Type	Declaration
`int`	`int *arg`
`Real`	`Real *arg`

Elemental Type	Declaration
`unsigned int`	`int *arg`
`Vector<2, unsigned int>`	`int (*arg)[2]`
`Vector<3, CoordReal>`	`CoordReal (*arg)[3]`
`Vector<3, Real>`	`Real (*arg)[3]`

For example,

```
CoordReal (*centroid)[3], Real *temperature
```

would declare centroid position and temperature as arguments.

The `Real` and `CoordReal` types define the precision of floating-point variables, and are defined in the uclib.h file.

Template Library Registration Function uclib (C)

This section provides a template that you can use when creating a library registration function.

Register each user function in the library with calls to `ucfunc` and `ucarg` in `uclib`, as shown in the examples. The template `uclib` is:

```
#include "uclib.h"
void uclib()
{
/* Register user functions here */
}
```

Create the header file `uclib.h` in the same directory as `uclib.c` to declare the functions that are used in `uclib`.

User Function Name Registration ucfunc (C)

This section describes the initial registration of a C user function.

The initial registration of a C user function is made with a call in uclib to `ucfunc`:

```
ucfunc(void *func, char *type, char *name);
```

`func`: a pointer to the user function.

`type`: a character string giving the type of the user function, one of the following:

- "BoundaryProfile"
- "RegionProfile"
- "ScalarFieldFunction"
- "VectorFieldFunction"

`name`: a character string giving the name with which the function is displayed as in drop-down lists in STAR-CCM+.

The arguments that the user function requires are registered by calling ucarg.

User Function Argument Registration ucarg (C)

This section describes the user function argument registration.

Each argument of a C user function which has been initially registered with ucfunc is requested with a call in uclib to `ucarg`:

```
ucarg(void *func, char *type, char *variable, int size);
```

`func`: a pointer to the user function.

`type`: a character string giving the type of the argument, either of the following:

- `"Cell"` for a cell field
- `"Face"` for a face field
- `"Vertex"` for a vertex field

`variable`: a character string naming the requested argument, which must be a valid STAR-CCM+ variable.

`size`: the storage in bytes required for one element of the `variable` array. This size is used to ensure that the precision of the user function matches the precision of STAR-CCM+. Examples are:

- `sizeof(Real)` if `variable` is a `Real` scalar per element
- `sizeof(CoordReal[3])` if `variable` is a `CoordReal` vector per element
- `sizeof(int[2])` if `variable` is a pair of integers per element

The `Real` and `CoordReal` types define the precision of floating-point variables, and are defined in the uclib.h file.

Calls to `ucarg` must be made in the order the user function needs the arguments in.

Fortran User Interface Reference

This section introduces the interface between STAR-CCM+ and Fortran user functions.

The interface between STAR-CCM+ and Fortran user functions is covered in the topics below:

- Type definitions
- The template user function
- The template library registration function (uflib)
- Function calls in `uflib` to register the user functions (uffunc and ufarg)
- The variables available from STAR-CCM+

The usage of this interface is illustrated in the examples. Although the interface is compatible with Fortran 77, code that is presented in this documentation assumes Fortran 90 as a minimum.

Contents:

Type Definitions (Fortran)
Template User Function (Fortran)
Template Library Registration Function uflib (Fortran)
User Function Name Registration uffunc (Fortran)
User Function Argument Registration ufarg (Fortran)

Type Definitions (Fortran)

The types of variables and functions that are used in Fortran user functions are defined in the file `StarReal.f`.

This file is not part of the standard STAR-CCM+ installation. Create it in the same directory where you are creating your user functions. The contents of `StarReal.f` appear as shown here:

```
module StarRealMod
        integer, parameter :: StarInt = kind(1)
        integer, parameter :: StarReal = kind(1.0)
        integer, parameter :: CoordReal = kind(1D0)
        integer, parameter :: StarIntSize = StarInt
        integer, parameter :: StarRealSize = StarReal
        integer, parameter :: CoordRealSize = CoordReal
      end module StarRealMod
```

When working with double precision, declare `StarReal` with:

```
integer, parameter :: StarReal = kind(1D0)
```

Template User Function (Fortran)

This section provides a template to help you create a user function.

The generic purpose of a user function is to populate a `result` array with values obtained by operating on `args`, as shown in the examples, using this template:

```
subroutine name(result, size, args ...)
use StarRealMod
integer, intent(in) :: size
```

`name`: the function name. This name is arbitrary: STAR-CCM+ does not impose particular function names for particular functionality. There are some protected names however which must be avoided: `ucarg, ucfunc, ucfunction, uclib, ufarg, uffunc, uflib`.

`result`: the array of values that the user function returns. It is declared as:

```
real(StarReal), intent(out) :: result(size)
```

for a scalar function, or:

```
real(StarReal), intent(out) :: result(3,size)
```

for a vector function. Each element of the `result` array represents a cell or face, depending on the type of the user function.

`size`: the number of elements in the `result` array.

`args`: the arguments which STAR-CCM+ passes to the function are chosen from the list of available variables and requested through `ufarg`. The form of declaration for an array `arg` with elements of various types is given in the table below.

Elemental Type	Declaration (all `intent(in)`)
`int`	`integer arg(*)`
`Real`	`real(StarReal) arg(*)`
`unsigned int`	`integer arg(*)`
`Vector<2, unsigned int>`	`integer arg(2,*)`
`Vector<3, CoordReal>`	`real(CoordReal) arg(3,*)`
`Vector<3, Real>`	`real(StarReal) arg(3,*)`

For example,

```
real(CoordReal), intent(in) :: centroid(3,*)
real(StarReal), intent(in) :: temperature(*)
```

would declare centroid position and temperature as arguments.

The `StarReal` and `CoordReal` types define the precision of floating-point variables, and are defined in the `StarReal.f` file.

Template Library Registration Function uflib (Fortran)

This section describes how to register each user function in the library.

Register each user function in the library with calls to `uffunc` and `ufarg` in `uflib`, as shown in the examples. The template `uflib` is:

```
subroutine uflib()
      use StarRealMod
C Register user functions here
      return
      end
```

User Function Name Registration uffunc (Fortran)

This section describes the initial registration of a Fortran user function.

The initial registration of a Fortran user function is made with a call in `uflib` to `uffunc`:

```
external func
call uffunc(func, type, name)
```

`func`: a reference to the user function (called `func` in this case).

`type`: a character string giving the type of the user function, one of the following:

- ‘BoundaryProfile’
- ‘RegionProfile’
- ‘ScalarFieldFunction’
- ‘VectorFieldFunction’

`name`: a character string giving the name with which the function is shown as in drop-down lists in STAR-CCM+.

The arguments that the user function requires are registered by calling `ufarg`.

User Function Argument Registration ufarg (Fortran)

This section describes user function argument registration.

Each argument of a Fortran user function which has been initially registered with `uffunc` is requested with a call in `uflib` to `ufarg`:

```
external func
integer size
call ufarg(func, type, variable, size)
```

`func`: a reference to the user function (called `func` in this case).

`type`: a character string giving the type of the argument, either of the following:

- `'Cell'` for a cell field
- `'Face'` for a face field

`variable`: a character string naming the requested argument, which must be a valid STAR-CCM+ variable.

`size`: the storage in bytes required for one element of the `variable` array. This value is used to ensure that the precision of the user function matches the precision of STAR-CCM+. Examples are:

- `StarRealSize` if `variable` is a `real(StarReal)` scalar per element
- `3*CoordRealSize` if `variable` is a `real(CoordReal)` vector per element
- `2*StarIntSize` if `variable` is a pair of integers per element

The `StarReal` and `CoordReal` types define the precision of floating-point variables, together with StarRealSize, CoordRealSize, and StarIntSize, and are defined in the `StarReal.f` file.

Calls to `ufarg` must be made in the order the user function requires the arguments.

Variables Available to User Functions

This section introduces the variables that are available to user functions.

User functions can access variables of two principal types:

- Field Functions
- Stored Variables

Accessing Field Functions in User Functions

Field functions can be accessed in user functions by requesting them with their names enclosed in double-quotes; the syntax depends on whether they are scalar or vector field functions. Temperature would be referred to as "`$Temperature`", for example, and velocity as "`$$Velocity`". Additionally:

- Field function arguments are always arrays of cell or face values, even when the fundamental data that they represent is a single value.

  For example "`$Iteration`" is a cell or face array, each element of which is the iteration number.
- Scalar field functions have `CoordReal` elemental type, even when the fundamental data that they represent has a different type.

  For example, whereas the iteration number is an integer, each element of "`$Iteration`" is the iteration number in `CoordReal` precision.
- Similarly, vector field functions have `Vector<3, CoordReal>` type, even when the fundamental data that they represent has a different type.
- The report can be accessed as a field function.

  For example, the **Cell Count** report spawns a **CellCountReport** field function, which can be requested in a user function as "`$CellCountReport`".

Accessing Stored Variables in User Functions

Some quantities in STAR-CCM+ are not available as field functions, and must be accessed directly. For example, `FaceCellIndex` is a stored variable of type `Vector<2, unsigned int>`.

User Code Examples

This section provides some examples of user functions.

The examples include:

- Defining a user-coded boundary profile
- Defining a user-coded region profile
- Defining a user-coded field function
- Registering user functions with STAR-CCM+
- Requesting and using a Cell field argument
- Requesting and using a Face field argument

The examples form a single library and are presented as both C and Fortran 90 coding.

Contents:

A User-Coded Boundary Profile
A User-Coded Region Profile
A User-Coded Field Function
Defining a Library Registration Function

A User-Coded Boundary Profile

This section contains an example of a user-coded boundary profile.

Suppose that you wish to specify the temperature on a boundary as the same temperature that is in the neighboring cell. You could create a user function that is called `zeroGradT` which takes the temperature and face-cell index fields as arguments and returns the boundary temperature.

In C, the following could be coded in a file `zeroGradT.c`, using the uclib.h file:

```
  #include "uclib.h"

  /* Set boundary temperature equal to cell temperature */
  void
  USERFUNCTION_EXPORT zeroGradT(Real *result, int size, int (*fc)[2],
Real *T)
  {
    int i;

  /* Loop through all entities applying T_boundary = T_cell *
   * fc[i][0] is the cell next to i                          */
    for (i = 0; i != size; ++i)
    {
      result[i] = T[fc[i][0]];
```

```
    }
  }
```

The equivalent in Fortran 90 could be a file `zeroGradT.f`, using the StarReal.f file:

```
C Set boundary temperature equal to cell temperature
      subroutine zeroGradT(result,size,fc,T)
      use StarRealMod
      implicit none
      integer, intent(in) :: size
      real(StarReal), intent(out) :: result(size)
      integer, intent(in) :: fc(2,*)
      real(StarReal), intent(in) :: T(*)
      integer i
C Loop through all entities applying T_boundary = T_cell
C fc(1,i) is the cell next to i
      do i = 1,size
        result(i) = T(fc(1,i))
      end do

      return
      end
```

This user function can then be registered with STAR-CCM+.

A User-Coded Region Profile

This section contains an example of a user-coded region profile.

Suppose that you wish to initialize the velocity to a constant swirl condition. You could create a user function that is called `initVelocity` which takes the centroids as an argument and returns the velocity vector at those centroid locations.

In C, the following could be coded in a file `initVelocity.c`, using the uclib.h file:

```
  #include "uclib.h"

  /* Initial velocity based on uniform swirl */
  void USERFUNCTION_EXPORT
  initVelocity(Real (*result)[3], int size, CoordReal (*centroid)[3])
  {
  /* Angular velocity and origin of rotation */
    CoordReal omega[3] = {0.0, 0.0, 100.0};
    CoordReal origin[3] = {0.0, 0.0, 0.0};
    CoordReal dr[3];
    int i;
```

```
  /* Loop through all entities applying u = omega x (centroid -
origin) */
    for (i = 0; i != size; ++i)
    {
      dr[0] = centroid[i][0] - origin[0];
      dr[1] = centroid[i][1] - origin[1];
      dr[2] = centroid[i][2] - origin[2];
      result[i][0] = (Real)(omega[1]*dr[2] - omega[2]*dr[1]);
      result[i][1] = (Real)(omega[2]*dr[0] - omega[0]*dr[2]);
      result[i][2] = (Real)(omega[0]*dr[1] - omega[1]*dr[0]);
    }
  }
```

The equivalent in Fortran 90 could be a file `initVelocity.f`, using the StarReal.f file:

```
C Initial velocity based on uniform swirl
      subroutine initVelocity(result,size,centroid)
      use StarRealMod
      implicit none
      integer, intent(in) :: size
      real(StarReal), intent(out) :: result(3,size)
      real(CoordReal), intent(in) :: centroid(3,*)
      integer i
      real(CoordReal) dr(3)
C Angular velocity and origin of rotation
      real(CoordReal), parameter :: omega(3) = (/0.0,0.0,100.0/)
      real(CoordReal), parameter :: origin(3) = (/0.0,0.0,0.0/)
C Loop through all entities applying u = omega x (centroid - origin)
      do i = 1,size
        dr(1) = centroid(1,i) - origin(1)
        dr(2) = centroid(2,i) - origin(2)
        dr(3) = centroid(3,i) - origin(3)
        result(1,i) = omega(2)*dr(3) - omega(3)*dr(2)
        result(2,i) = omega(3)*dr(1) - omega(1)*dr(3)
        result(3,i) = omega(1)*dr(2) - omega(2)*dr(1)
      end do

      return
      end
```

This user function can then be registered with STAR-CCM+.

A User-Coded Field Function

This section provides an example of a user-coded field function.

Suppose that you would like to define the Sutherland viscosity law as a field function. Defining this field function means that you can visualize the Sutherland viscosity for a given temperature field without having to apply it as the dynamic viscosity in your region of interest. The law is simple enough to be implemented as a user field function, but here you can see how it can be implemented as a user function. The function could be called `sutherlandViscosity`, taking the temperature as an argument and returning the viscosity.

In C, the following could be coded in a file `sutherlandViscosity.c`, using the uclib.h file:

```
  #include <math.h>
  #include "uclib.h"
  /* Dynamic viscosity based on Sutherland's law */
  void USERFUNCTION_EXPORT
  sutherlandViscosity(Real *result, int size, Real *T)
  {
  /* Reference viscosity, Sutherland constant and reference
temperature */
    Real v0 = 1.716E-5;
    Real Cs = 110.0;
    Real T0 = 273.15;
    int i;
  /* Loop through all entities applying Sutherland's law */
    for (i = 0; i != size; ++i)
    {
      result[i] = v0 * pow(T[i]/T0, 1.5) * (T0 + Cs)/(T[i] + Cs);
    }
  }
```

The equivalent in Fortran 90 could be a file `sutherlandViscosity.f`, using the StarReal.f file:

```
C Dynamic viscosity based on Sutherland's law
      subroutine sutherlandViscosity(result,size,T)
      use StarRealMod
      implicit none
      integer, intent(in) :: size
      real(StarReal), intent(out) :: result(size)
      real(StarReal), intent(in) :: T(*)
      integer i
C Reference viscosity, Sutherland constant and reference temperature
      real(StarReal), parameter :: v0 = 1.716E-5
      real(StarReal), parameter :: Cs = 110.0
      real(StarReal), parameter :: T0 = 273.15

C Loop through all entities applying Sutherland's law
      do i = 1,size
        result(i) = v0 * (T(i)/T0)**1.5 * (T0 + Cs)/(T(i) + Cs)
      end do
```

```
      return
      end
```

This user function can then be registered with STAR-CCM+.

Defining a Library Registration Function

This section provides an example of a library registration function.

The example functions zeroGradT, initVelocity, and sutherlandViscosity must be registered with STAR-CCM+ through the library registration function.

In C, this function could be achieved by coding the following in `uclib.c`, using the uclib.h file:

```
  #include "uclib.h"

  void zeroGradT(Real*, int, int*, Real*);
  void initVelocity(Real*, int, CoordReal*);
  void sutherlandViscosity(Real*, int, Real*);

  void
  USERFUNCTION_EXPORT uclib()
  {
    /* Register user functions here */
    ucfunc(zeroGradT, "BoundaryProfile", "Zero Gradient Temperature");
    ucarg (zeroGradT, "Face", "FaceCellIndex", sizeof(int[2]));
    ucarg (zeroGradT, "Cell", "Temperature", sizeof(Real));

    ucfunc(initVelocity, "RegionProfile", "Initial Velocity");
    ucarg(initVelocity, "Cell", "Centroid", sizeof(CoordReal[3]));

    ucfunc(sutherlandViscosity, "ScalarFieldFunction", "Sutherland
Viscosity");
    ucarg(sutherlandViscosity, "Cell", "Temperature", sizeof(Real));
  }
```

The equivalent library registration function in Fortran 90 could be coded in `uflib.f`, using the StarReal.f file:

```
      subroutine uflib()
      use StarRealMod
      implicit none
C Register user functions here
      external zeroGradT,initVelocity,sutherlandViscosity
      call uffunc(zeroGradT, "BoundaryProfile",
     &            "Zero Gradient Temperature")
      call ufarg (zeroGradT, "Face",
     &            "FaceCellIndex", 2*StarIntSize)
```

```
      call ufarg (zeroGradT, "Cell",
     &            "Temperature", StarRealSize)

      call uffunc(initVelocity, "RegionProfile",
     &            "Initial Velocity")
      call ufarg(initVelocity, "Cell",
     &           "Centroid", 3*CoordRealSize)

      call uffunc(sutherlandViscosity, "ScalarFieldFunction",
     &            "Sutherland Viscosity")
      call ufarg(sutherlandViscosity, "Cell",
     &           "Temperature", StarRealSize)
      return
      end
```

Using User Code on Particles and Parcels

This section gives examples of cases where user coding applies to particles and parcels in multiphase modeling.

The first example covers the topic in detail, while the subsequent examples build on the first. The examples are listed below:

Contents:

Changing the Distribution of Diameters in a Transient Injection
Changing the Mass Flow Rate in a Transient Injection
Defining a User-Coded Energy Source in a Parcel
Defining a User-Coded Body Force on a Parcel Cloud

Changing the Distribution of Diameters in a Transient Injection

This example examines the case of particles such as droplets being injected into the simulation.

STAR-CCM+ uses the Rosin-Rammler distribution to describe droplet diameters. The probability of finding a particle with diameter D in a given set of particles (the probability density function, or PDF) is:

$$PDF\left(D\right) = \exp\left[\left(\frac{-D}{D_{ref}}\right)^{q}\right] \tag{15}$$

where D_{ref} is the Rosin-Rammler diameter and q is the Rosin-Rammler exponent.

On a log-log graph, this distribution is a straight line; D_{ref} is the intercept and q is the slope. The cumulative probability density function (or CDF) is the integrated probability:

$$Y\left(D\right) = 1 - \exp\left[\left(\frac{-D}{D_{ref}}\right)^{q}\right] \tag{16}$$

where $Y(D)$ is the mass fraction of droplets with diameter less than D and is bounded between zero and one. This relation can be solved for D to obtain a sample of diameters at different mass fraction points:

$$D = D_{ref}[-\ln(1-Y)]^{\frac{1}{q}} \tag{17}$$

This relation can be used to return diameters for a given random sample of Y in the range [0,1]. Writing a routine to return the diameter is straightforward in Fortran or C, but you must provide some additional coding to link the routine into STAR-CCM+.

This example requires four files:

- A file that defines the size of variables in a module for a shared Fortran object library, StarReal.f
- A file containing definitions that link the variables inside STAR-CCM+ to the external library created for the case, uflib.f

- Source code, returning the appropriate diameter, rosinRammlerDiameter.f
- A makefile, for convenience

Module File StarReal.f

The first file, StarReal.f, defines a module which denotes the standard variable sizes that are used in STAR-CCM+. Compiling StarReal.f creates the Fortran shared object library. The file is:

```
module StarRealMod
     integer, parameter :: StarInt = kind(1)
     integer, parameter :: StarReal = kind(1.0)
     integer, parameter :: CoordReal = kind(1D0)
     integer, parameter :: StarIntSize = StarInt
     integer, parameter :: StarRealSize = StarReal
     integer, parameter :: CoordRealSize = CoordReal
 end module StarRealMod
```

Note: When using double precision, define `StarReal` as `kind(1.0d0)`.

Library File uflib.f

The library file, uflib.f, registers the user routine and defines the internal variables to be passed into your code. The file is:

```
subroutine uflib()
     use StarRealMod
     implicit none

  c Register user functions here

     external rosinRammlerDiameter

     call uffunc(rosinRammlerDiameter,
     &    "ParcelProfile",
     &    "Rosin-Rammler Diameter")

     return

  end
```

In this case, the internal routine uffunc looks for a module named rosinRammerDiameter, which is being applied internally to a ParcelProfile.

The third argument in the list, `Rosin-Rammler Diameter`, is the title that appears in the interface when the library is loaded.

Subroutine rosinRammlerDiameter.f

This subroutine calculates the diameters by sampling the CDF uniformly. The example here uses the built-in Fortran random number generator `rand()`, which means that this implementation depends on the machine

and compiler combination. Other suitable pseudorandom number generators are available from several sources.

```
subroutine rosinRammlerDiameter(result,size)

  c particle diameter example

     use StarRealMod

     implicit none
     integer, intent(in) :: size
     real(StarReal), intent(out) :: result(size)

     real(StarReal), parameter :: meandiam = 1e-5
     real(StarReal), parameter :: rrexpon = 1.0/3.5

     integer i

     do i = 1,size
      result(i) = meandiam*(-log(1.0-rand()))**rrexpon
     end do

     return
     end
```

The makefile

Below is an example of a makefile that contains the compilation flags for gnu Fortran95:

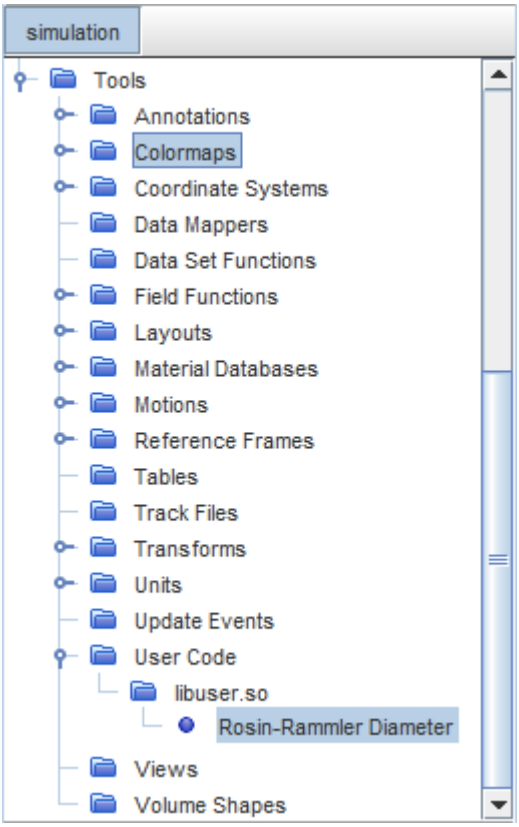
```
default: libuser.so

  libuser.so: StarReal.f uflib.f rosinRammlerDiameter.f
          f95 StarReal.f uflib.f rosinRammlerDiameter.f -fPIC -shared -
o $@
```

Running the makefile produces a shared object library libuser.so. After building the library, you connect the function to STAR-CCM+.

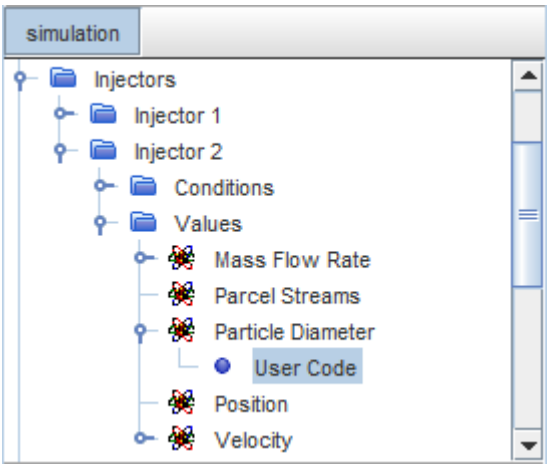
Connecting User Code to STAR-CCM+

To connect the user code:

1. Right-click the **Tools** > **User Code** node and select **New User Library...**.
2. Browse for libuser.so in the file system and expand the **User Code** node:



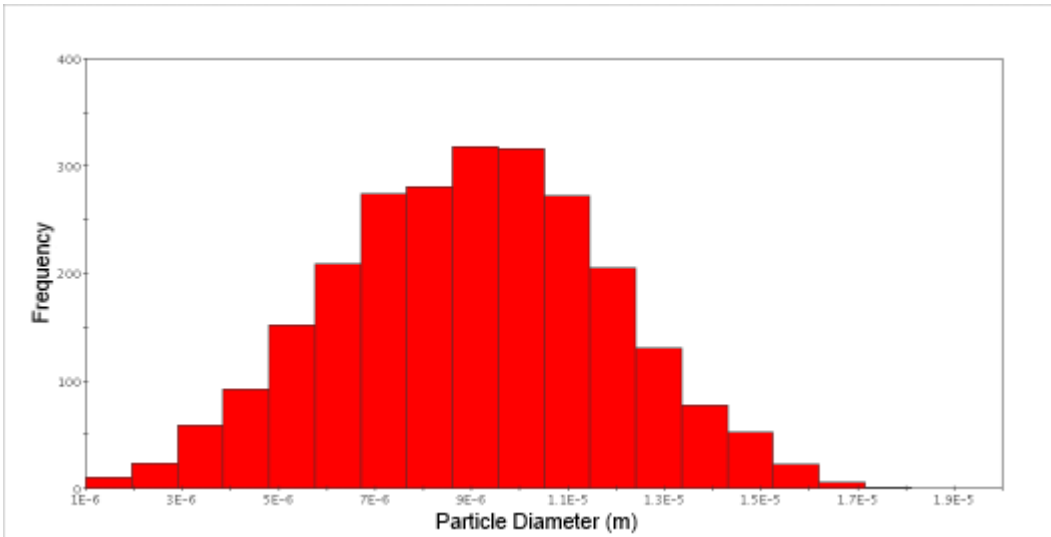
Assuming an appropriate injection has been defined for the case, the **User Code** option appears as an option for the **Particle Diameter** definition.



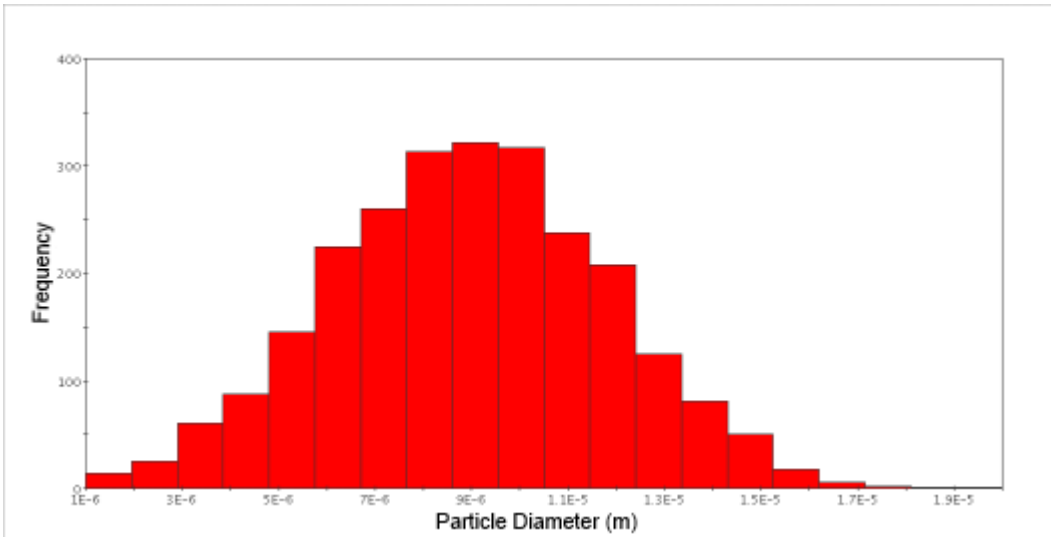
3. To select the Rosin-Rammler distribution, edit the *Function* property of the **User Code** node.

The case that is shown here defines two injectors. In one, the particle diameter is given by the Rosin-Rammler particle size distribution as defined internally by STAR-CCM+. In the other, the user coding defines the Rosin-Rammler distribution. The injectors are otherwise the same.

The following graph is the histogram of particle diameter for the internally defined Rosin-Rammler diameter distribution:



The following graph is the histogram of particle diameter for the user-coded Rosin-Rammler diameter distribution:



There are slight differences in mean diameter and actual number distribution for the two definitions. These differences are due mostly to the choice of sampling for the random numbers and clipping of maximum and minimum values between the methods. In both cases, the Rosin-Rammler diameter D_{ref} was 1.0E-5 meters and the Rosin-Rammler exponent q was set to 3.5. The differences become smaller for a larger number of samples.

Changing the Mass Flow Rate in a Transient Injection

This section provides an example of changing the mass flow rate in a transient injection.

By default, each parcel in STAR-CCM+ represents the same amount of mass. The mass of a parcel is the product of the mass of an individual particle $\left(\rho\left(\frac{4}{3}\right)\pi r^3\right)$ and the parcel count. The parcel count is the number of particles contained in the parcel. The parcel count is a floating point number and can also be considered the probability of finding a particle with that mass and diameter at that spatial location.

It is straightforward to change the amount of mass in each parcel as a function of diameter with user code. In this example, the mass flow rate of each parcel increases linearly with diameter.

To pass an argument to a subroutine, the argument must be registered with a call to the library routine ufarg():

```
      subroutine uflib()
      use StarRealMod
      implicit none

c Register user functions

      external linearDiameter
      external parcelMdot

      call uffunc(linearDiameter,
      &     "ParcelProfile",
      &     "Linear Diameter")

      call uffunc(parcelMdot,
      &     "ParcelProfile",
      &     "Parcel Mass Flow")

      call ufarg(parcelMdot,
      &     "Parcel",
      &     "$ParticleDiameter",
      & CoordRealSize)

      return
      end
```

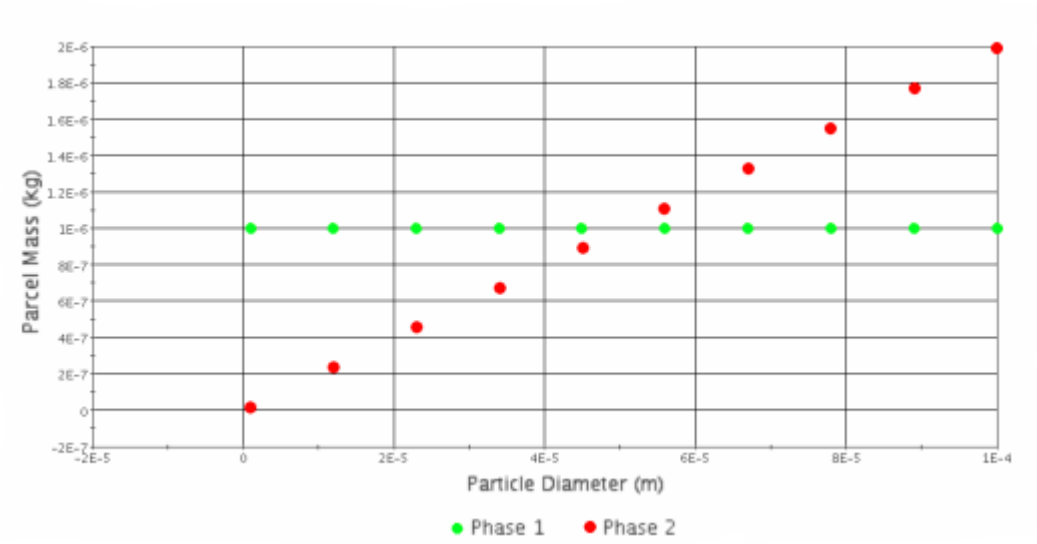
To set a linear diameter distribution, modify the rosinRammlerDiameter.f routine by replacing the line that sets the result. In this example, the minimum diameter is set to 1 micron and the maximum is set to 100 microns. These diameters can be set as parameters. The linear distribution can be coded as:

```
      delta = (maxdiam-mindiam)/(size-1)
      do i = 1, size
        result(i) = mindiam+(i-1)*delta
      enddo
```

As the example routine below sets the mass flow rate for each parcel in the injection, you must set the mass flow rate condition to Per Parcel Stream. This setting is valid only if the injector is on a single partition— it does not work in parallel if the partition cuts through the injector.

```
      subroutine parcelMdot(result,size,D)

      use StarRealMod

      implicit none
      integer, intent(in) :: size
      real(StarReal), intent(out) :: result(size)
      real(CoordReal), intent(in) :: D(*)

c parcel mass is count*rho*volume, so for an exponent of
c -3 parcel mass is constant (default behavior)
c -2 parcel mass increases with diameter linearly
c -1 parcel mass increases with diameter quadratically
c etc.

      real(StarReal), parameter :: expon = -2.0

      real(StarReal) sum, ratio, mdot
      integer i
c a constant mass flow rate for all particles
      mdot = 1.0e-3*size
c set the Parcel Count
      sum = 0.0
      do i = 1, size
       result(i) = D(i)**expon
       sum = sum + result(i)*D(i)**3
      end do

c scale for flow rate per parcel
      if (sum > 0.0) then
       ratio = mdot/sum
       do i = 1, size
         result(i) = ratio*result(i)*D(i)**3
       enddo
     endif

     return
     end
```

An XY plot of the parcel mass as a function of particle diameter shows the default behavior of STAR-CCM+ (Phase 1) and the results of the user code (Phase 2) with the parcel mass increasing linearly with diameter.



Defining a User-Coded Energy Source in a Parcel

In simple cases, internal heating of particles can be modeled using field functions, but more complex cases can be modeled with user coding. This example compares the two methods for a transient simulation injecting aluminum particles into a square box.

A constant energy source of 1.0E8 W/m^3 heats the particles during the 0.05 seconds of simulation time, raising their temperature from 300 to 301.12 degrees. Since the heat source is constant, the temperature increases linearly with residence time.

Parts of the parcel profile can become invalid due to other physical processes occurring during a substep of the Lagrangian solution procedure. The user code must check the `parcelId` for validity before operating on the profile. The profile must have the `parcelId` passed in as an argument in addition to the arguments necessary to calculate the heat source.

The subroutine uflib.f registers the `parcelId` and the temperature:

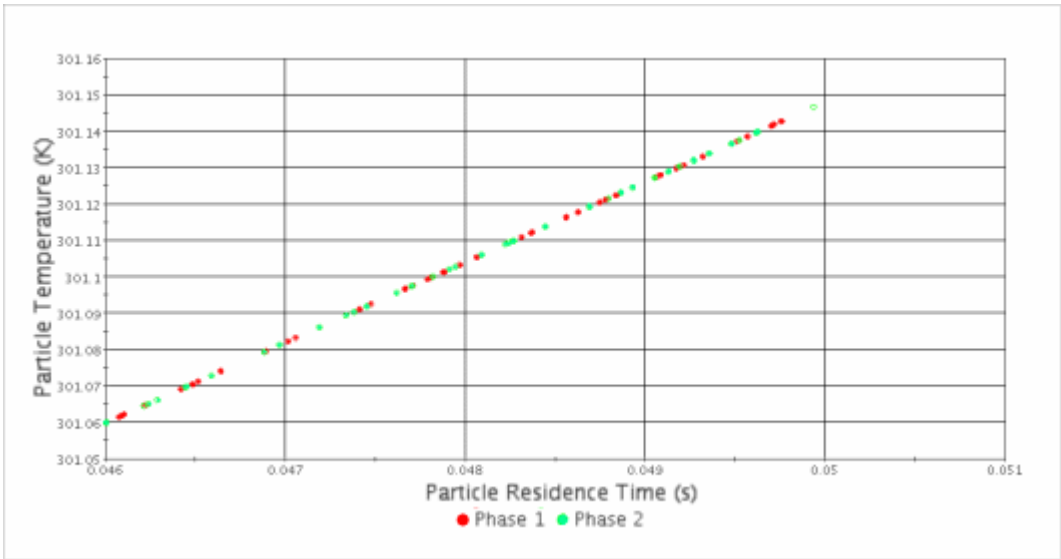
```
subroutine uflib()
     use StarRealMod
     implicit none

 c Register user functions

    external parcelQdot

    call uffunc(parcelQdot,
     &    "ParcelProfile",
     &    "Parcel Heat Source")

    call ufarg(parcelQdot,
     & "Parcel",
```

```
     & "ParcelId",
     & StarInt)

    return
     end
```

The user code that applies the heat source to the parcels in the domain is:

```
subroutine parcelQdot(result,size,Id)

 c parcel heat source example

    use StarRealMod

    integer, intent(in) :: size
     real(StarReal), intent(out) :: result(size)
     real(StarInt), intent(in) :: Id(*)

    real(StarInt), parameter :: NULL_INDEX = -1
     integer i

    do i = 1, size
       if (Id(i) .gt. NULL_INDEX) then
         result(i) = 1e8
       endif
     end do

    return
     end
```

When compared to the internal user-defined heat source, the field functions and the user code above have identical results:



In the plot above, Phase 1 corresponds to the internally calculated parcel heat flux and Phase 2 corresponds to the user-coded parcel heat flux. Since the particle times are staggered slightly during the injection process for this case, the residence times differ slightly. In both cases, however the particle temperature is linearly proportional to residence time.

Defining a User-Coded Body Force on a Parcel Cloud

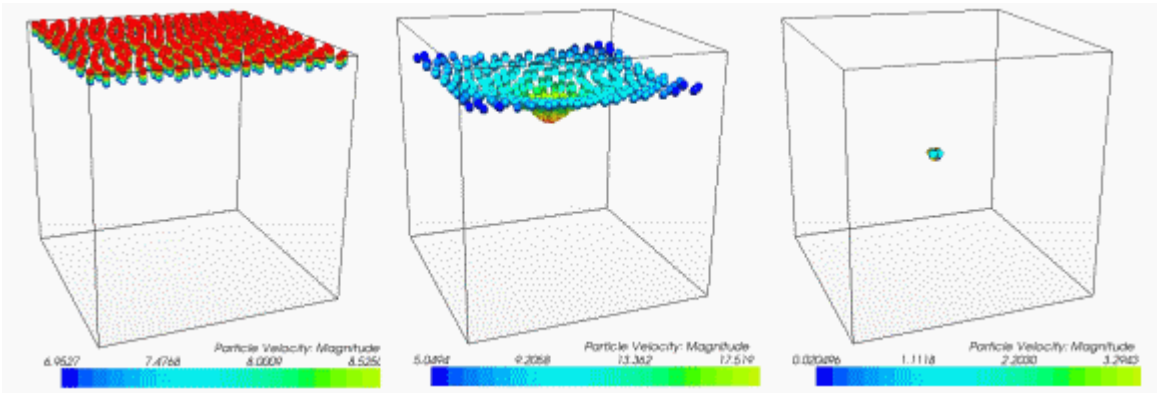
Defining a vector body force on a parcel cloud provides considerable flexibility in modeling phenomena acting on a parcel cloud. This example demonstrates such a body force, simulated first with field functions and then with user code.

In this simple case, the user code is longer and more difficult than the field functions, but in more elaborate situations, user code can be easier to write and debug.

In both methods:

- The body force on the parcel cloud is inversely proportional to the cube of the distance from the origin to the parcel position.
- The minimum radial distance is clipped to 1 cm to eliminate the singularity.
- Particles are injected for 10 ms from the top surface with a velocity of -10 m/s.
- On the time-step after injection, the body force begins to act, attracting the particles to the origin at the center of the domain.
- After 250 ms, the parcels are (unphysically) gathered at the origin.

Both the field function method and the user code method produce the same results.



For both field functions and user code, the injection mass flow rate is set to a field function that turns the injection off after 10 ms (that is, using a field function of: `($Time < 0.01) ? 0.001 : 0.0)` in the injection definition.

Field Function Method

Two field functions are defined to describe the body force:

- **rCubed**, defined as:

```
max(0.01,pow($$ParcelCentroid.mag(),3))
```

- **UserParticleBodyForce**, defined as:

```
($Time > 0.01) ? -9.81e5*($$ParcelCentroid)/$rCubed : [0,0,0]
```

User Code Method

The user code defining the same body force is:

```
subroutine parcelBodyForce(result,size,centroid,Id,Time)
 c parcel body force example
   use StarRealMod
   integer, intent(in) :: size
   real(StarReal), intent(out), dimension(3,size) :: result
   real(CoordReal), intent(in), dimension(3,*) :: centroid
   real(StarInt), intent(in), dimension(*) :: Id
   real(CoordReal), intent(in) :: Time
 c body force is proportional to 1/r^2 after 0.01 seconds
   real(StarReal), parameter:: aScale = -9.81e5
   real(StarReal), parameter:: center(3) = (/0.0, 0.0, 0.0/)
   real(StarInt), parameter :: NULL_INDEX = -1
   real(StarReal) radius(3), rmag
   integer i
 c set the magnitude and direction for valid parcels
   do i = 1, size
     if ((Id(i) .gt. NULL_INDEX) .and. (Time > 0.01)) then
       radius(:) = centroid(:,i)-center(:)
       rmag = max(0.01,sqrt(sum(radius**2))**3)
       result(:,i) = aScale*(radius)/rmag
     endif
   end do
   return
   end
```

The user code specifies a vector force on the parcel cloud, returned in the result vector. The parcel centroid is passed into the user code as a vector field function with a size of `3*CoordReal`. The uflib.f function that registers this function and argument list is:

```
subroutine uflib()
    use StarRealMod
 c Register user functions
    external parcelBodyForce
    call uffunc(parcelBodyForce,
    &    "ParcelProfile",
    &    "Parcel Body Force")
    call ufarg(parcelBodyForce,
    &    "Parcel",
```

```
    &    "$$ParcelCentroid",
    &    3*CoordRealSize)
    call ufarg(parcelBodyForce,
    &    "Parcel",
    &    "ParcelId",
    &    StarInt)
    call ufarg(parcelBodyForce,
    &    "Parcel",
    &    "$Time",
    &    CoordRealSize)
    return
    end
```

In this case, both the ParcelCentroid and Time field functions are required in the routine.

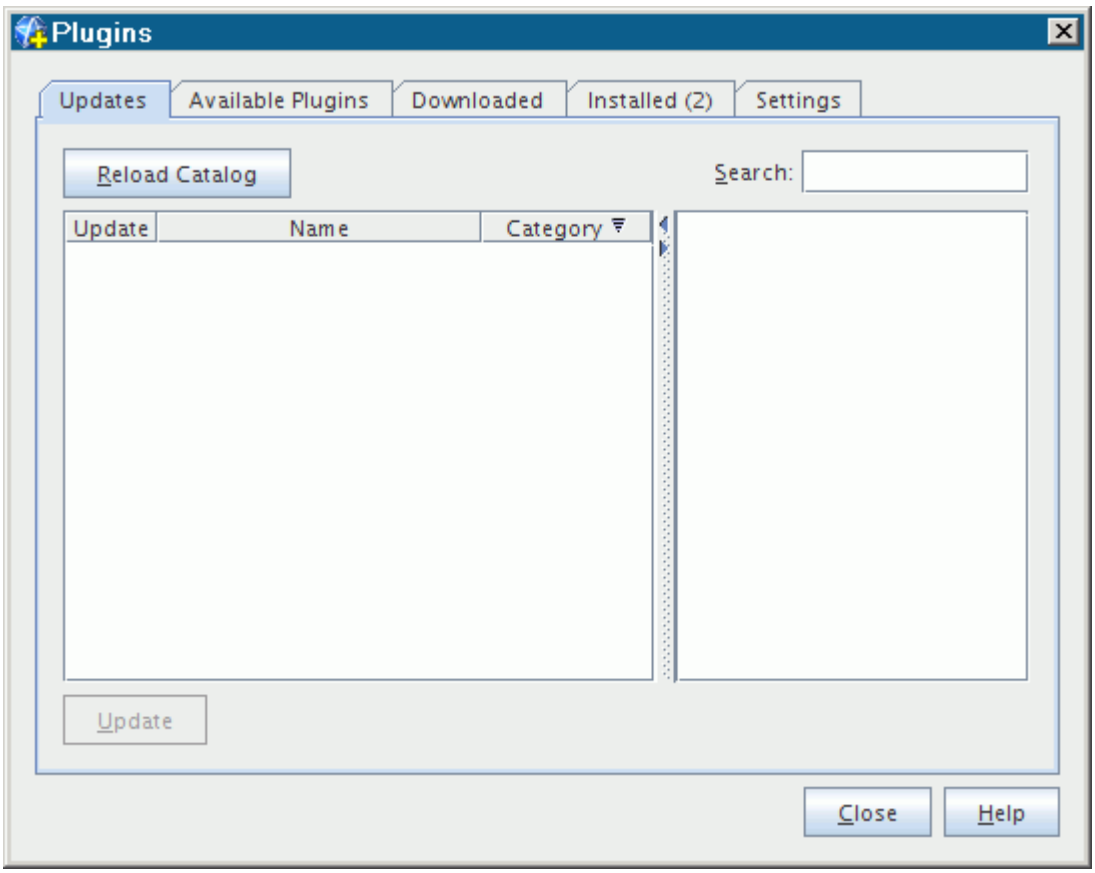
Note: The registration calls that register the arguments in STAR-CCM+ must be performed in the same order as the arguments appear in the parcelBodyForce function.

Note: These examples are not coupled with the flow field.

Working with the Plugins Manager

The Plugins Manager makes it possible to install and manage NetBeans modules for STAR-CCM+.

The Plugins Manager is opened when you select **Tools** > **Plugins** from the top menus.



The NetBeans modules (`.nbm` files) can be thought of as plugins that permit you to extend the STAR-CCM+ GUI dynamically. Modules range in complexity from a single Java class that executes a single operation to a group of classes to control the generation of application-specific models.

The Macro Hut site contains examples of NetBeans modules. To learn how to create your own modules, contact Siemens PLM Software support.

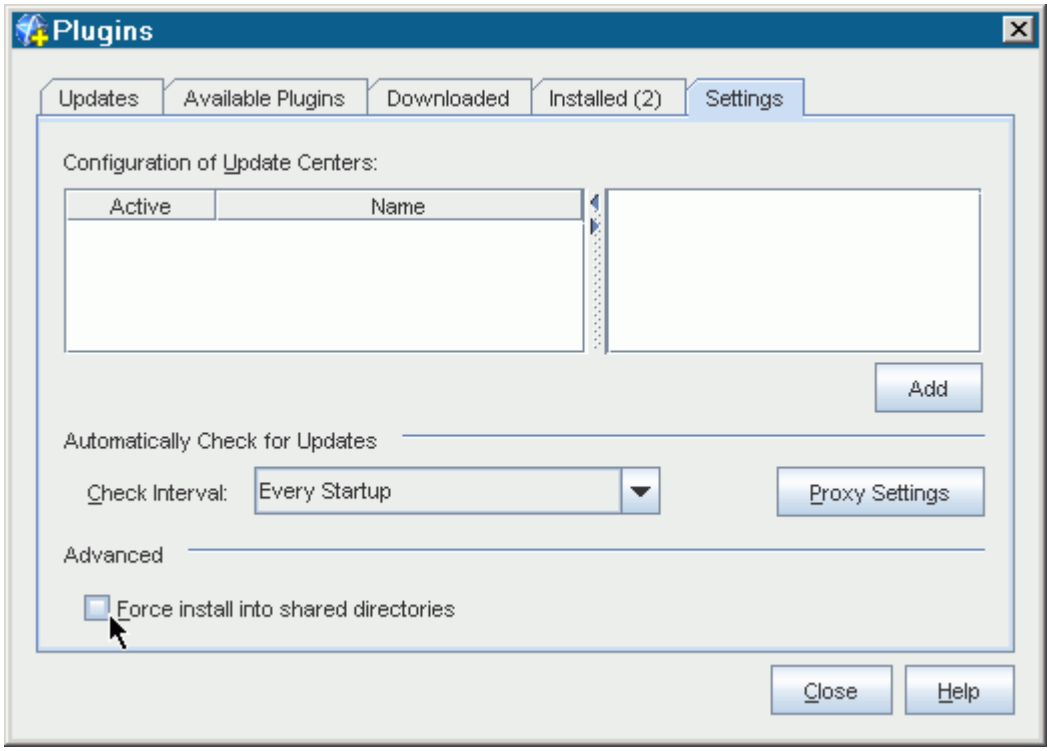
This section provides information on the following:

- Installing a NetBeans Module file
- Deactivating an Installed Plugin

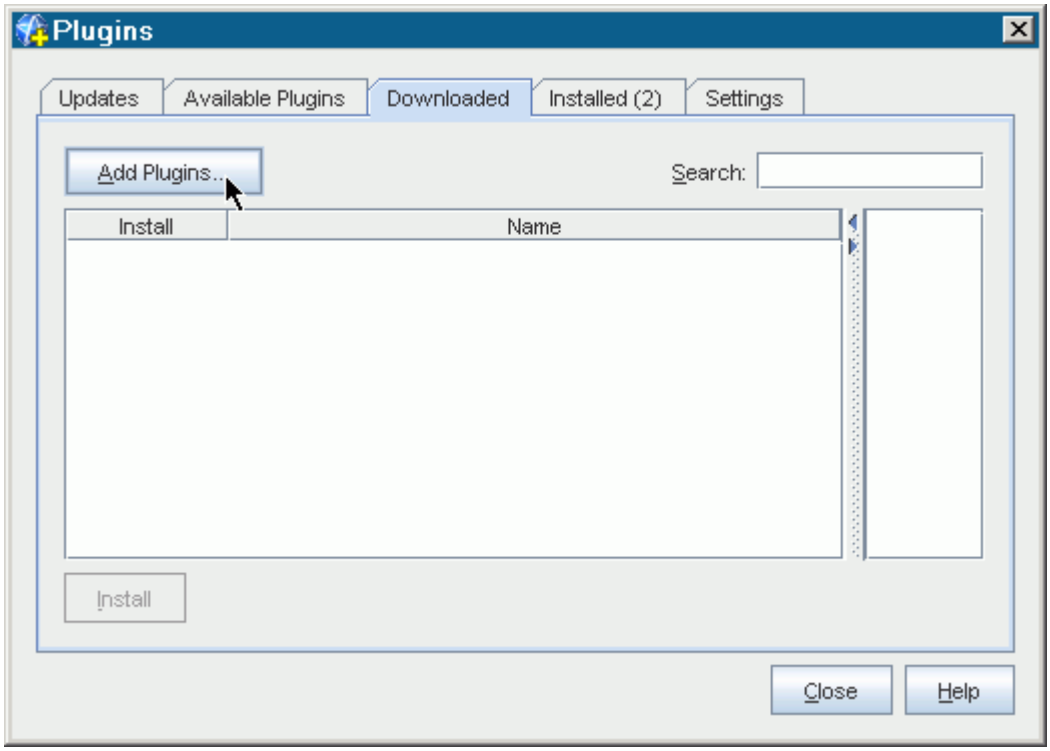
Installing a NetBeans Module File

To install a plugin, it is necessary to modify the STAR-CCM+ installation as follows:

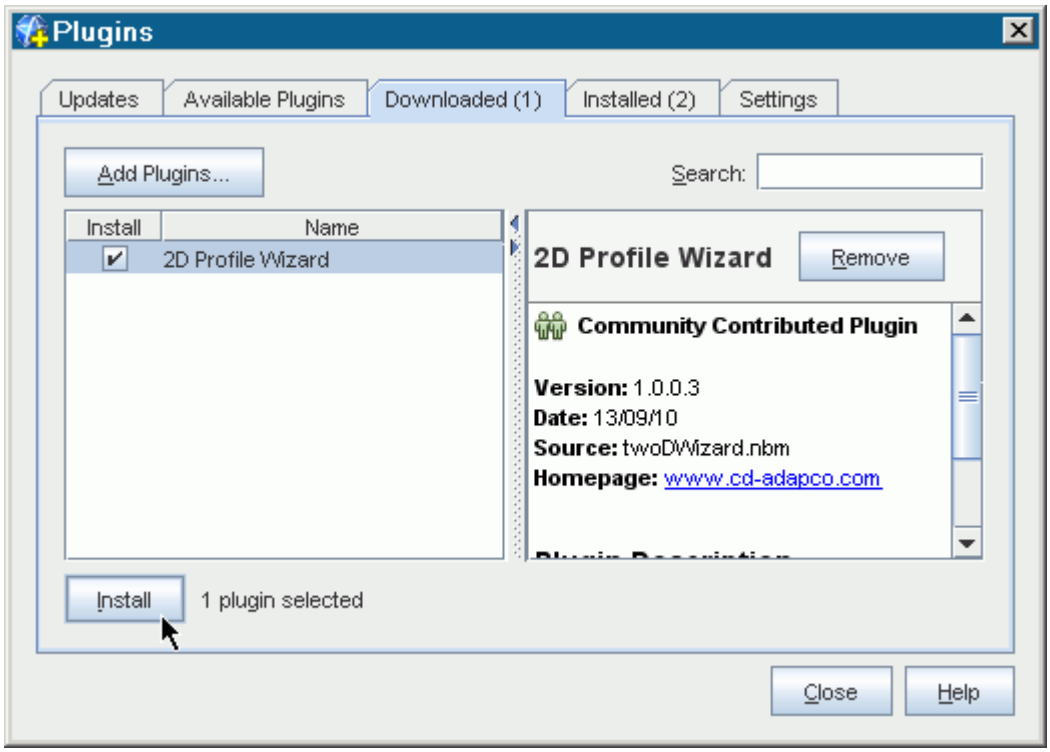
1. Choose **Tools** > **Plugins** from the main menu and select the *Settings* tab.
2. If you want to make the plugin available for all users, activate *Force install into shared directories*, otherwise deactivate this function.



3. Select the *Downloaded* tab.
4. Click **Add Plugins** and browse to the location of your created or downloaded .nbm file.

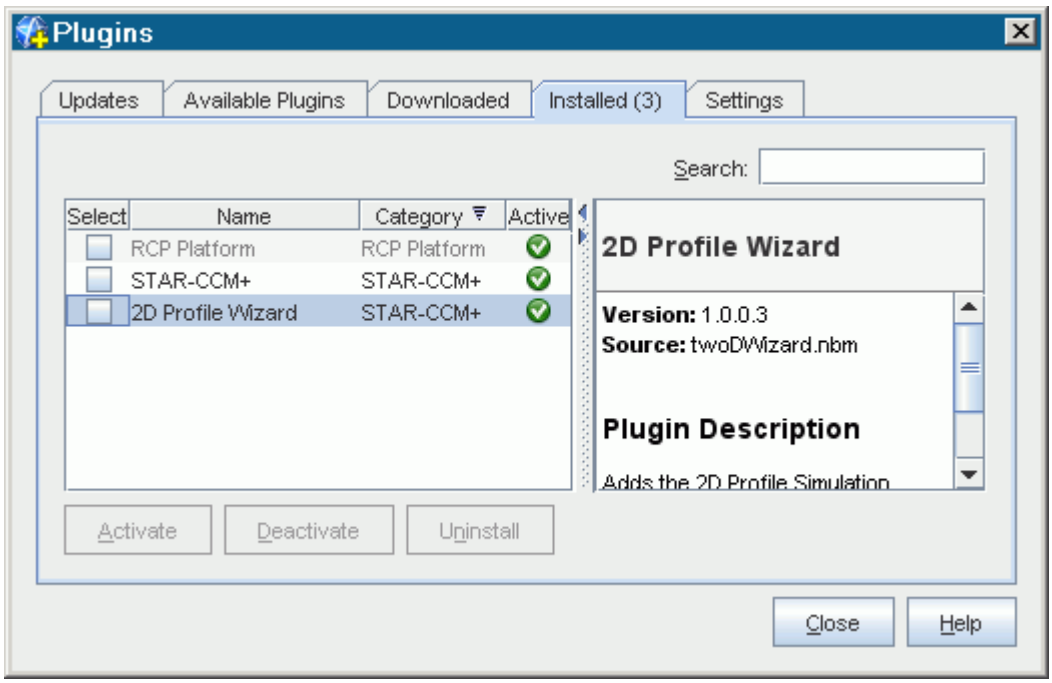


5. Click **Install** and complete the installation.



6. Restart STAR-CCM+ if prompted to do so.

   When the installation is complete, the plugin appears in the *Installed* tab as activated.



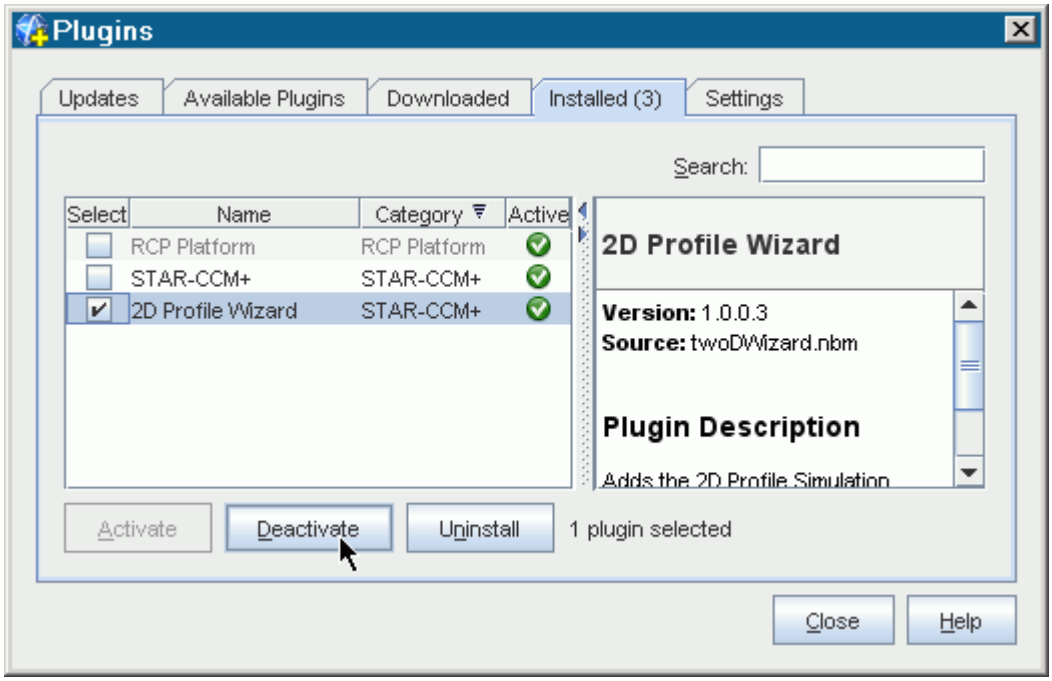
Deactivating an Installed Plugin

STAR-CCM+ allows you to deactivate installed plugins that you do not need to minimize startup time and save memory.

Note: Do not deactivate the STAR-CCM+ plugin.

To deactivate a plugin:

1. Choose **Tools** > **Plugins** from the main menu and select the *Installed* tab.
2. Select the plugin.
3. Click **Deactivate**.



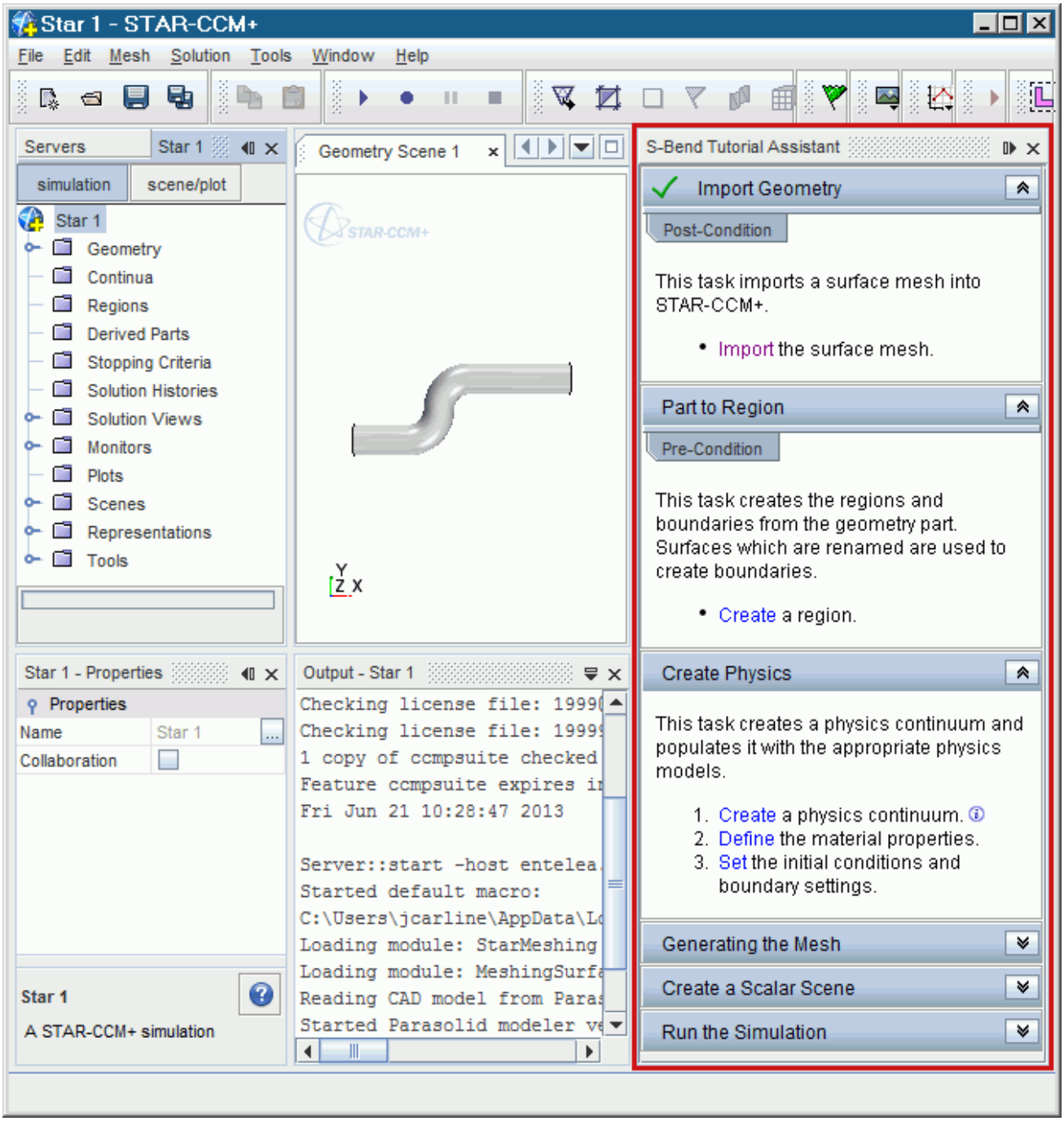
4. Restart STAR-CCM+ if prompted to do so. If you require it later, the plugin can be reactivated using the **Activate** button.

Uninstalling a plugin follows a similar workflow to deactivating with the exceptions that you click the **Uninstall** button and the plugin cannot be reactivated. The STAR-CCM+ plugin cannot be uninstalled.

Simulation Assistants

A Simulation Assistant provides a visual workflow within the STAR-CCM+ user interface (UI).

By default, this visual workflow appears as a panel to the right-hand side of the Graphics window. The panel is composed of a series of steps, or tasks, that you work through in sequence. You can reposition the Simulation Assistant panel in STAR-CCM+ as if it were any other scene.



STAR-CCM+ provides a Java framework that allows you to create your own Simulation Assistants. You create a Simulation Assistant using a combination of Java classes from the Simulation Assistant framework and Java macros that you record from STAR-CCM+. You require a working knowledge of Java in order to create a Simulation Assistant. When complete, you can share Simulation Assistant with other users, who can load them directly into STAR-CCM+. You do not require any knowledge of Java to use a Simulation Assistant.

If you are provided with an existing Simulation Assistant, you can load this into the STAR-CCM+ UI.

Benefits of Using a Simulation Assistant

There are numerous benefits to having an assistant in your simulation:

- New users can learn how to use STAR-CCM+ faster by focusing on the topic of interest, and understand the process workflow.
- Existing users can save time by following a pre-defined workflow that only exposes the areas that are relevant to their work.
- Expert users can create workflows that make it easy for non-expert users to accomplish complex tasks.

Assistant Workflow

The workflow of a Simulation Assistant is made up of a hierarchy of tasks. Each task is presented in its own panel in the assistant. The aim of the workflow is to help you set up a simulation in STAR-CCM+ without forcing you to perform tasks in a linear order.

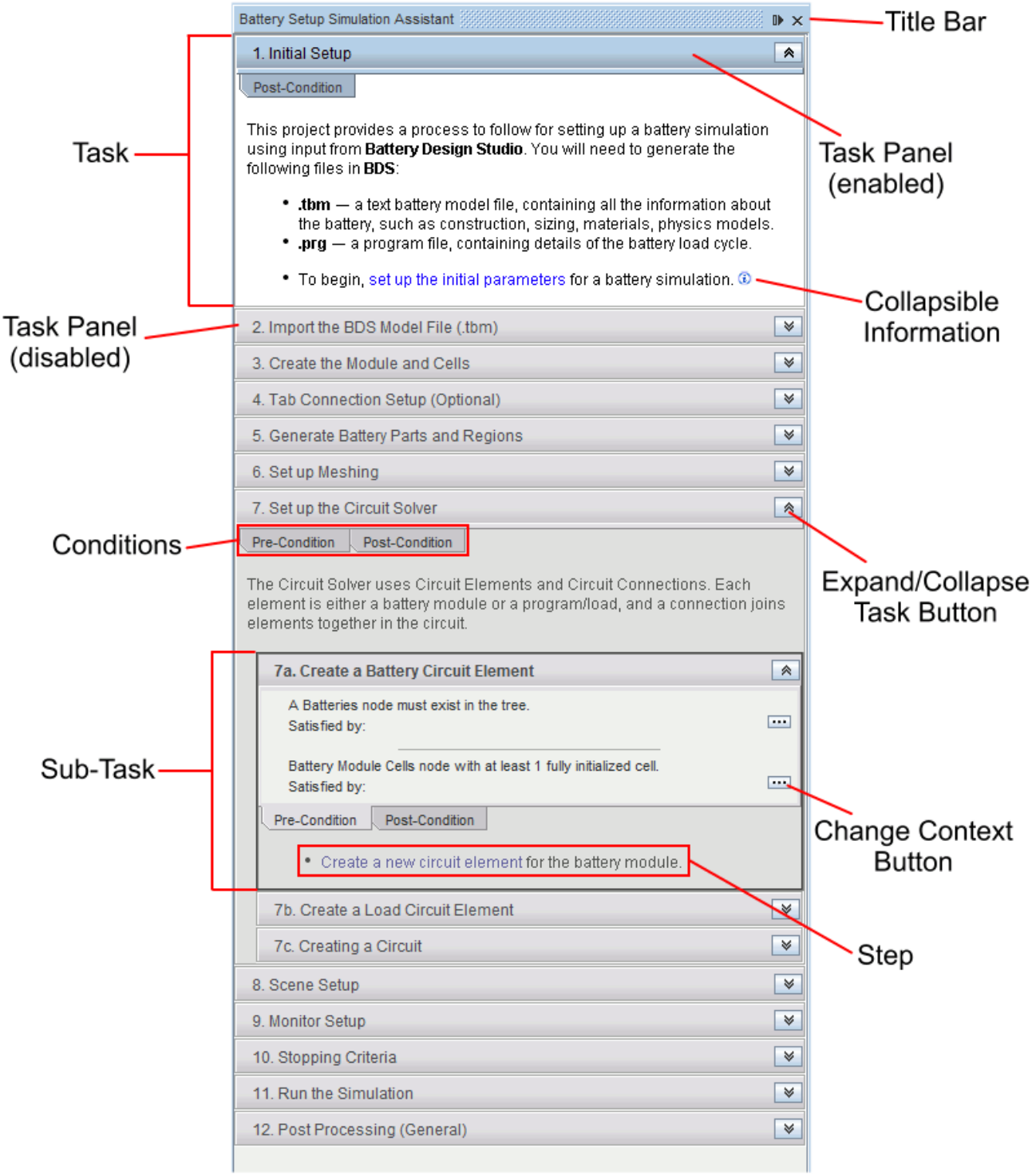
Note: It is recommended that you follow a top-to-bottom approach when working with assistants. Jumping back and forth between tasks, especially if the tasks have no conditions, can lead to errors.

Contents:

The Simulation Assistant User Interface
Setting Up a Simulation Assistant Project in NetBeans
Developing a Simulation Assistant

The Simulation Assistant User Interface

Every assistant has various user interface items.



Contents:

Tasks
Collapsible Information
Object Tree Focus
Conditions

Tasks

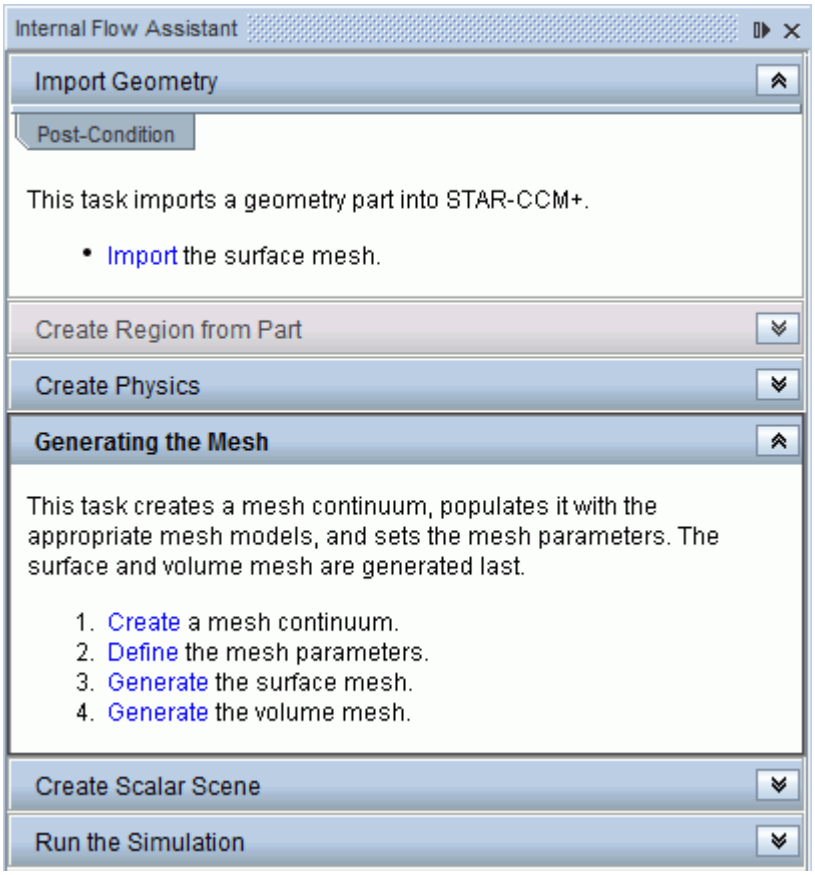
Tasks are hierarchical groups of steps that you perform in a simulation.

For example, a task could contain the steps for setting up a physics continuum. In the Simulation Assistant, each task is represented in a separate panel, called the Task Panel.

A task can contain any number of steps which are related to the task. Typically, the steps are list items which contain a combination of plain text and hyperlinks that are linked to an action. These actions are known as Star Actions, and can drive any operation in the STAR-CCM+ User Interface. Clicking on Star Actions typically causes a change in the STAR-CCM+ simulation. A Star Action can also execute code written by the author of the Simulation Assistant. The code can interact with STAR-CCM+ User Interface elements or the STAR-CCM+ API. The task is disabled while a Star Action executes. Star Actions appear as hyperlinks in steps.

Tasks can contain conditions, which determine when a task is ready to use, or when a task is complete. If tasks do not have conditions, you are best to follow a top-to-bottom approach through the workflow, without moving backwards and forwards between the tasks.

You can control which tasks and sub-tasks are visible in the assistant by expanding and collapsing them.



To expand a task:

- Click (or anywhere else) in the task panel that you want to expand. The selected task expands.

To collapse a task:

- Click (or anywhere else) in the task panel that you want to collapse. The selected task collapses.

Sub-Tasks

Sub-tasks are tasks that are nested under a single task. Typically, sub-tasks are groups of steps that are related to the parent task.

Steps and Star Actions

Steps and Star Actions appear in tasks and are the smallest unit in the Simulation Assistant.

A step is typically represented as a bullet point, or a numbered item, and is expected to perform a single action. A step can also contain a description of what the user must do.

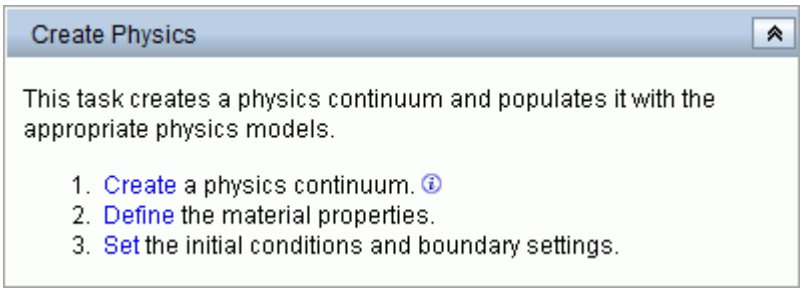
See Also:

Conditions

Collapsible Information

Collapsible information contains additional information that is hidden by default in the assistant.

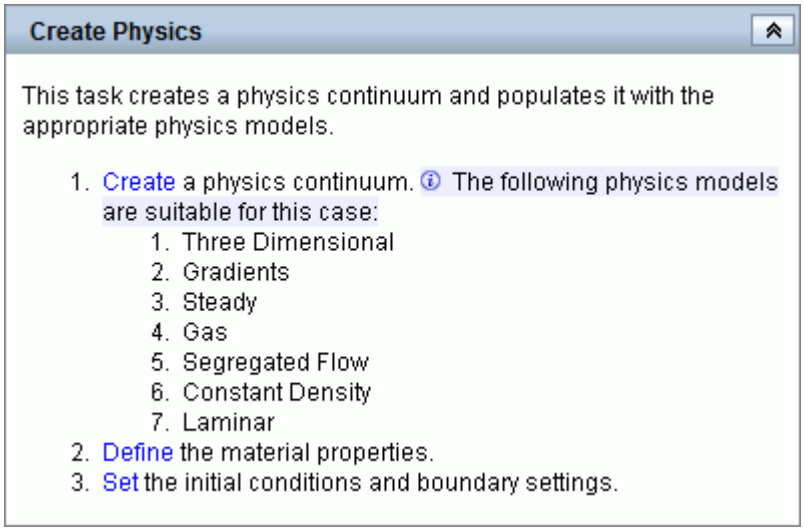
Collapsible information is marked by the icon. Typically, information that you store here is not essential for the completion of the task, but is there to provide extra information when necessary.



To view the additional information:

- Click the icon.

  The additional information is displayed in the task.



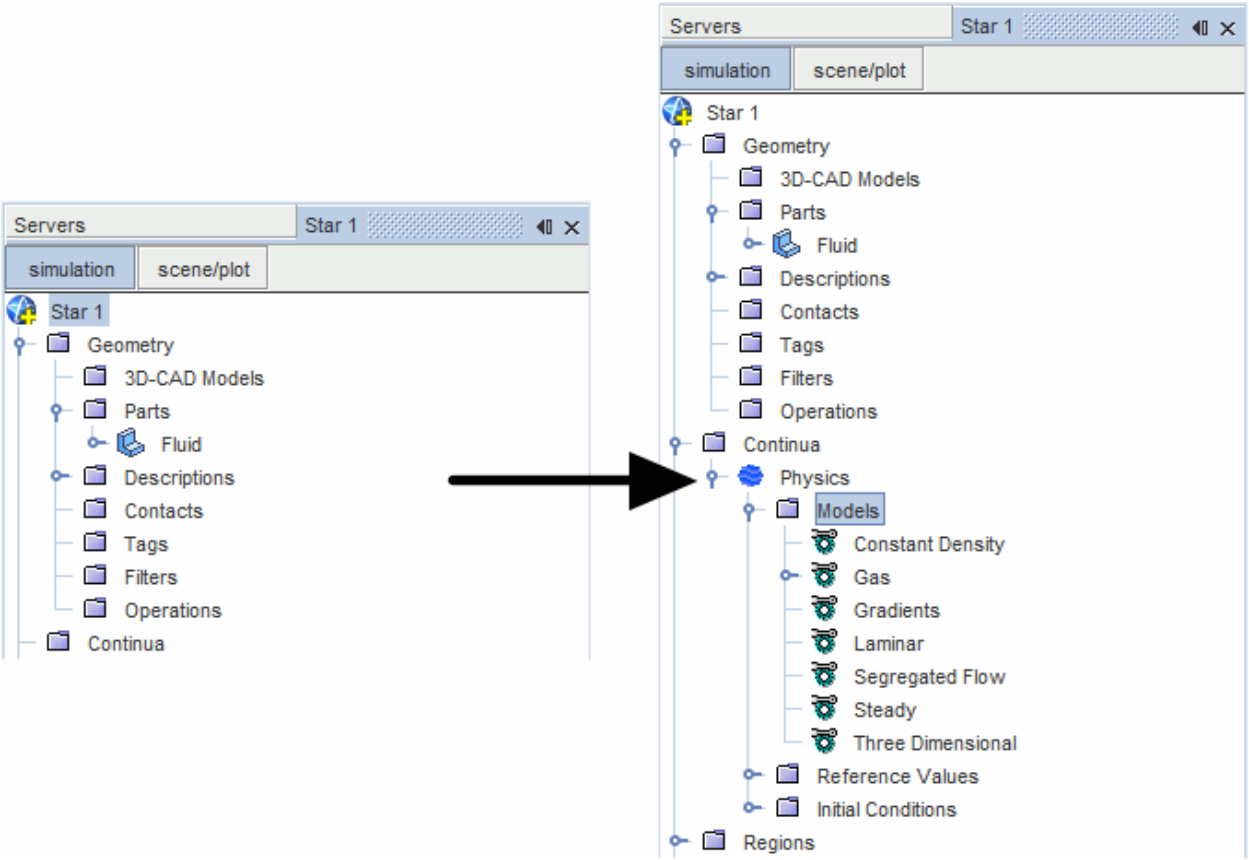
Object Tree Focus

Simulation assistants can provide an object tree focus—that is, when you click a step, the relevant node in the STAR-CCM+ object tree expands and is highlighted.

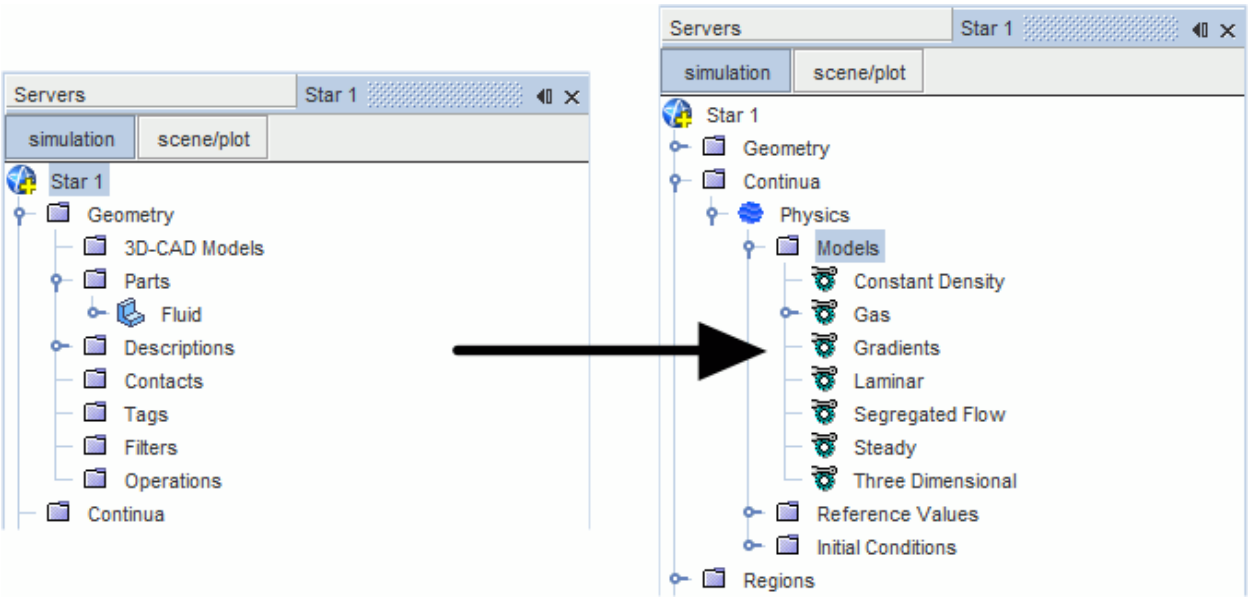
This facility is useful for users that are not familiar with the STAR-CCM+ object tree. Object tree focus can also act as a hint to what the corresponding Star Action does, such as expanding the physics continuum models node when the task adds a model to the physics continuum.

There are two types of object tree focus:

- *Inclusive* — the relevant node in the object tree is expanded, leaving all other expanded nodes as they were.



- *Exclusive* — the relevant node in the object tree is expanded, and all other nodes are closed.



Conditions

Conditions add an extra layer of control and guidance to the assistant, and act as a safeguard when you move between tasks. In the Simulation Assistant, conditions are represented by tabs in each task panel.

There are two types of conditions:

- *Pre-conditions* — are conditions that must be satisfied before you can start the current task.
- *Post-conditions* — are conditions that notify you if the constraints for the current task have been satisfied.

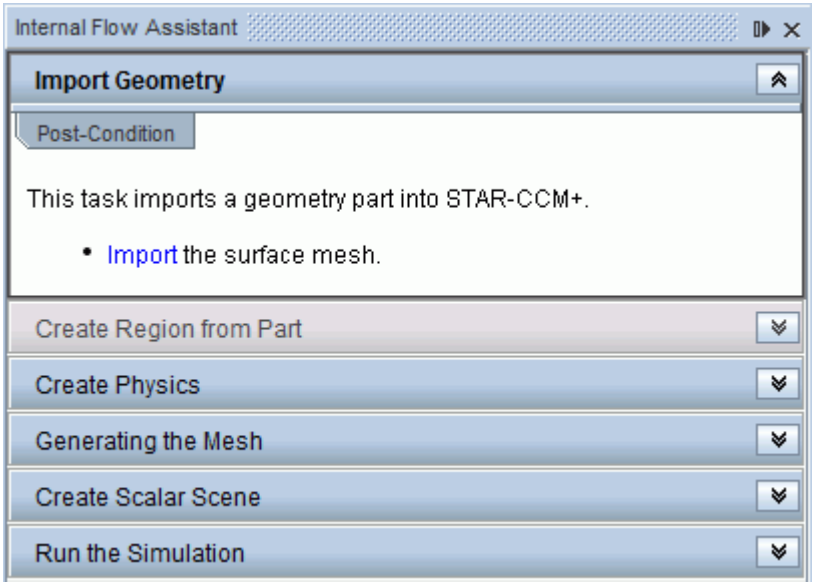
Contents:

Satisfying Conditions
Changing the Context of a Condition
Loading a Simulation Assistant into STAR-CCM+

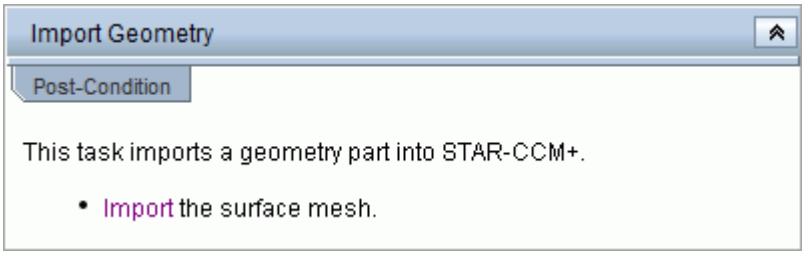
Satisfying Conditions

Conditions control whether you can perform a particular task or not.

If a task has a pre-condition that has not been satisfied, the task is grayed out and you cannot perform the task.



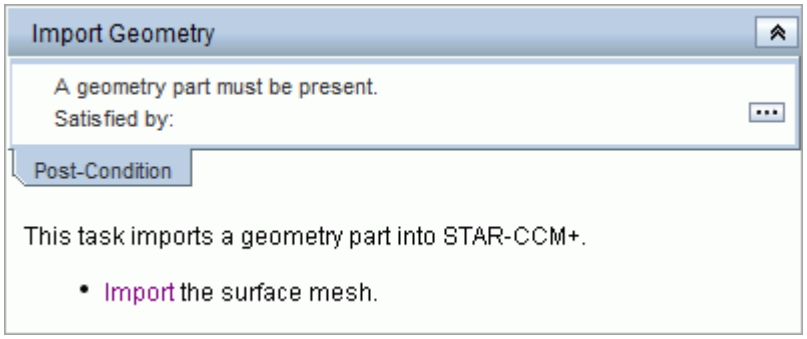
By default, condition tabs are minimized.



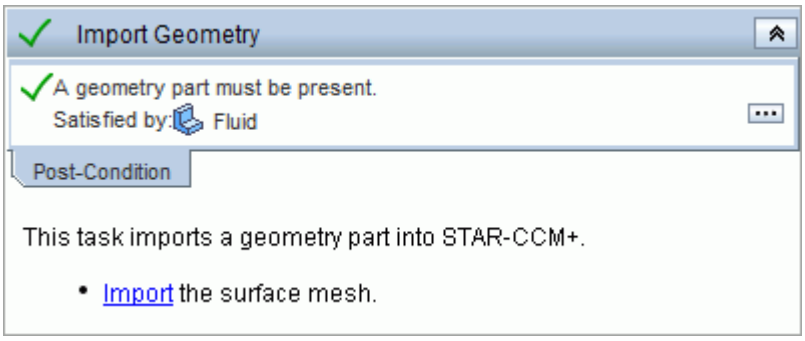
To view more information about a particular condition:

- Click the condition tab in the task panel.

  The tab expands to reveal more information, such as what is required in order to satisfy the condition. In the following example, a geometry part must be present in the simulation in order to satisfy the condition.



When a condition is satisfied, a green checkmark appears next to the condition. Tasks that have satisfied their post-conditions also have a green checkmark next to the title in the task panel.



Note: Tasks that have only satisfied their pre-conditions do not have a green checkmark in their task panel.

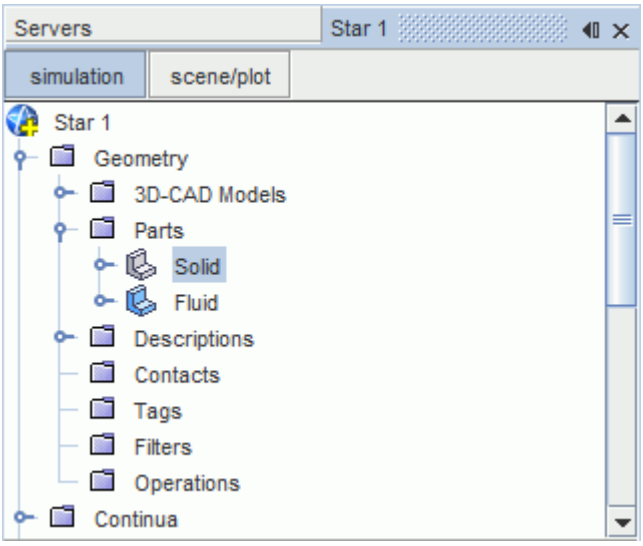
Changing the Context of a Condition

The context of a condition is the object in STAR-CCM+ that satisfies the condition—for example, a geometry part. You can change which object satisfies a condition by changing the context of the condition.

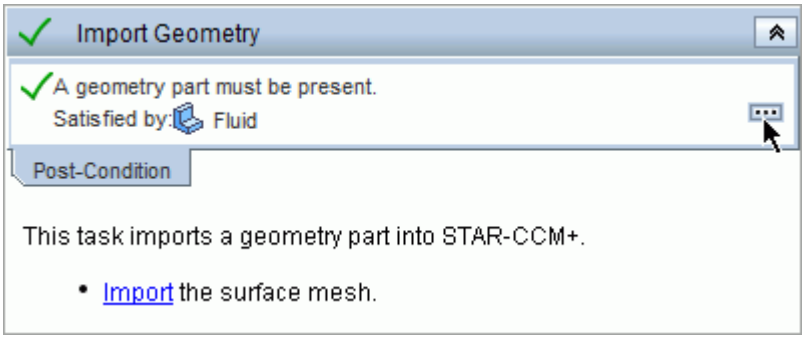
In the following example, a condition is already satisfied by a geometry part, however the part that satisfied the condition will be changed.

To change the context of a condition:

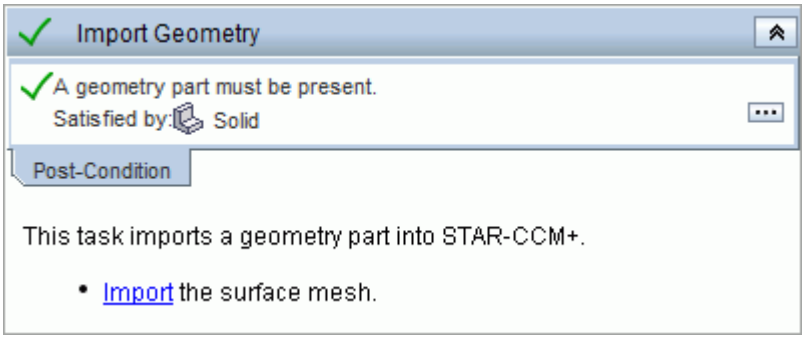
1. On the STAR-CCM+ object tree, select the object that you want to use to satisfy the condition.



2. Expand the tab of the condition that you want to change and click ...



The selected object is now used to satisfy the condition.



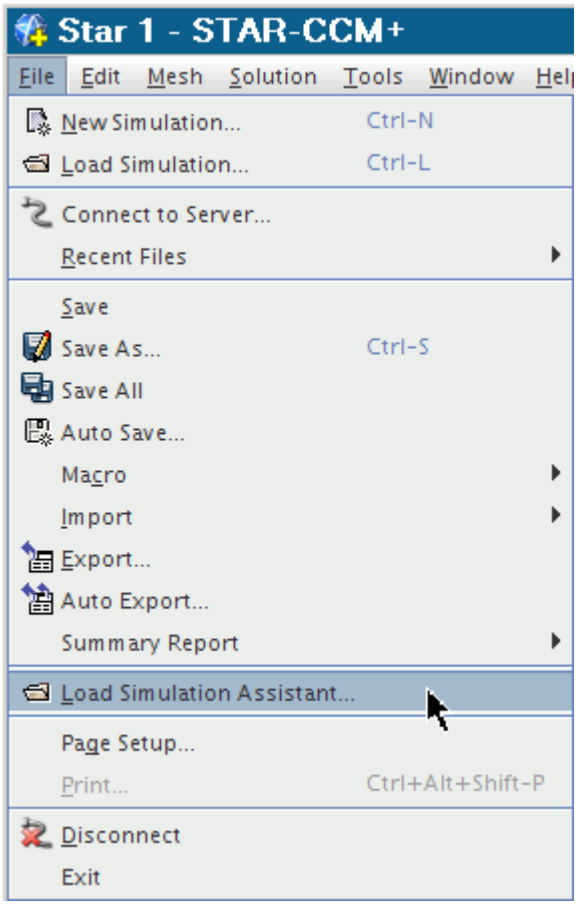
Note: You can only replace the context of a condition with an object of the same type. For example, you can replace a CAD part with a Mesh part, as both are geometry parts—but you cannot replace a CAD part with a 3D-CAD model. The object may also need to match other parameters that are typically specified in the condition description.

Loading a Simulation Assistant into STAR-CCM+

Simulation assistants are stored in precompiled `.java` or `.jar` files.

To load a Simulation Assistant into your simulation:

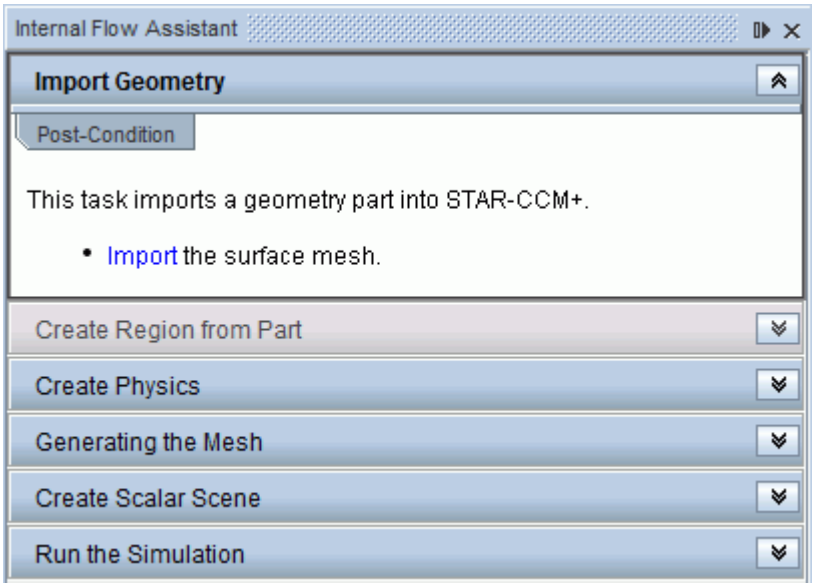
1. Launch STAR-CCM+ and open or create a simulation.
2. Select **File** > **Load Simulation assistant...**



The *Open* dialog appear.

3. Navigate to the location where the Simulation Assistant is stored and select it.
4. Click **OK**.

The assistant loads into the simulation. By default, the assistant appears on the right-hand side of the user interface.


Internal Flow Assistant
Import Geometry
Post-Condition
This task imports a geometry part into STAR-CCM+.
Import the surface mesh.
Create Region from Part
Create Physics
Generating the Mesh
Create Scalar Scene
Run the Simulation

Setting Up a Simulation Assistant Project in NetBeans

To develop STAR-CCM+ Simulation Assistants, it is recommended that you use an IDE (Integrated Development Environment).

IDEs offer many benefits compared to standard text editors, such as auto-indenting your code and offering auto-completion for method and class names (when you import the appropriate application libraries). This guide uses NetBeans 7.3.1 IDE. You can get a free copy of NetBeans SE from the official website.

This section contains the following topics:

Contents:

Creating A Simulation Assistant Project In NetBeans
Adding Java Packages, Java Classes, and XHTML Files to the Simulation Assistant Project
Adding STAR-CCM+ Libraries to the Simulation Assistant Project

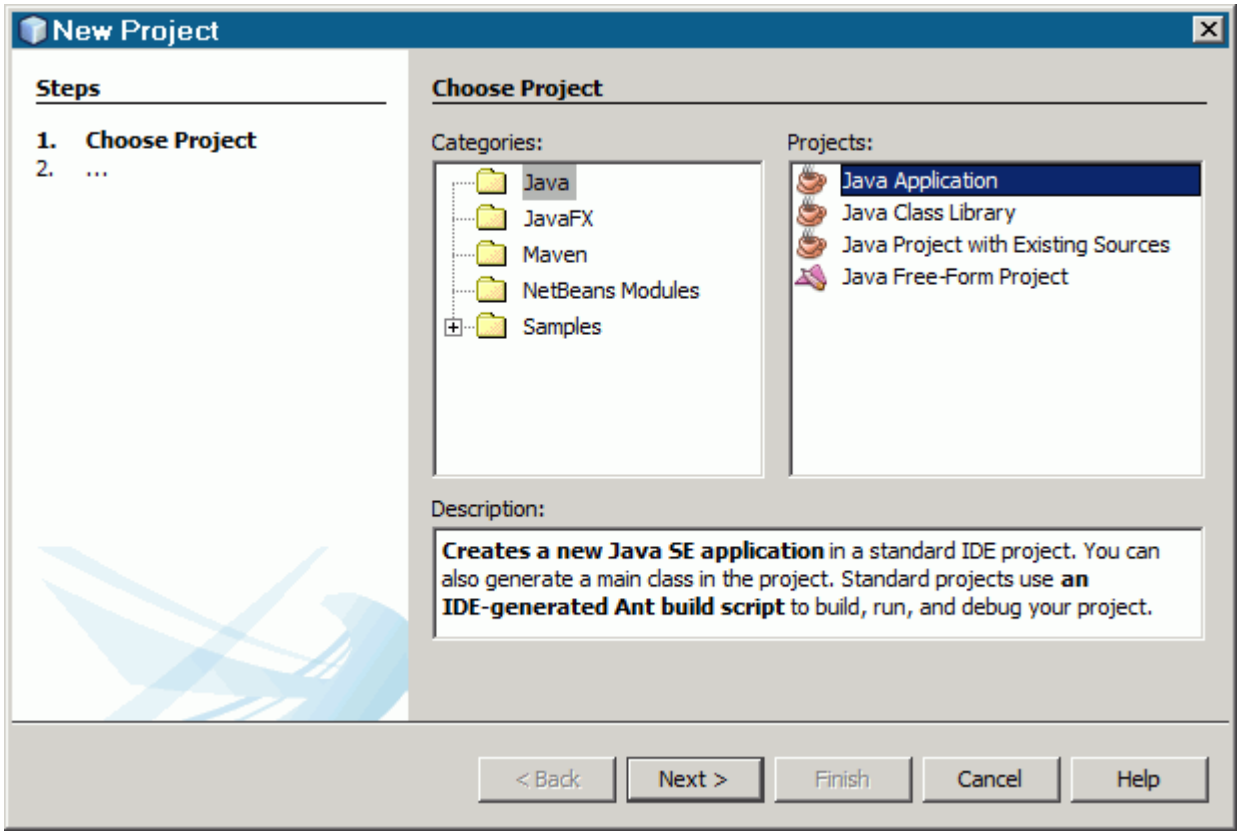
Creating A Simulation Assistant Project In NetBeans

To set up a NetBeans project:

1. Download and install the latest version of NetBeans from the official website.
2. Launch NetBeans.
3. Select **File** > **New Project...**.

The *New Project* dialog appears.

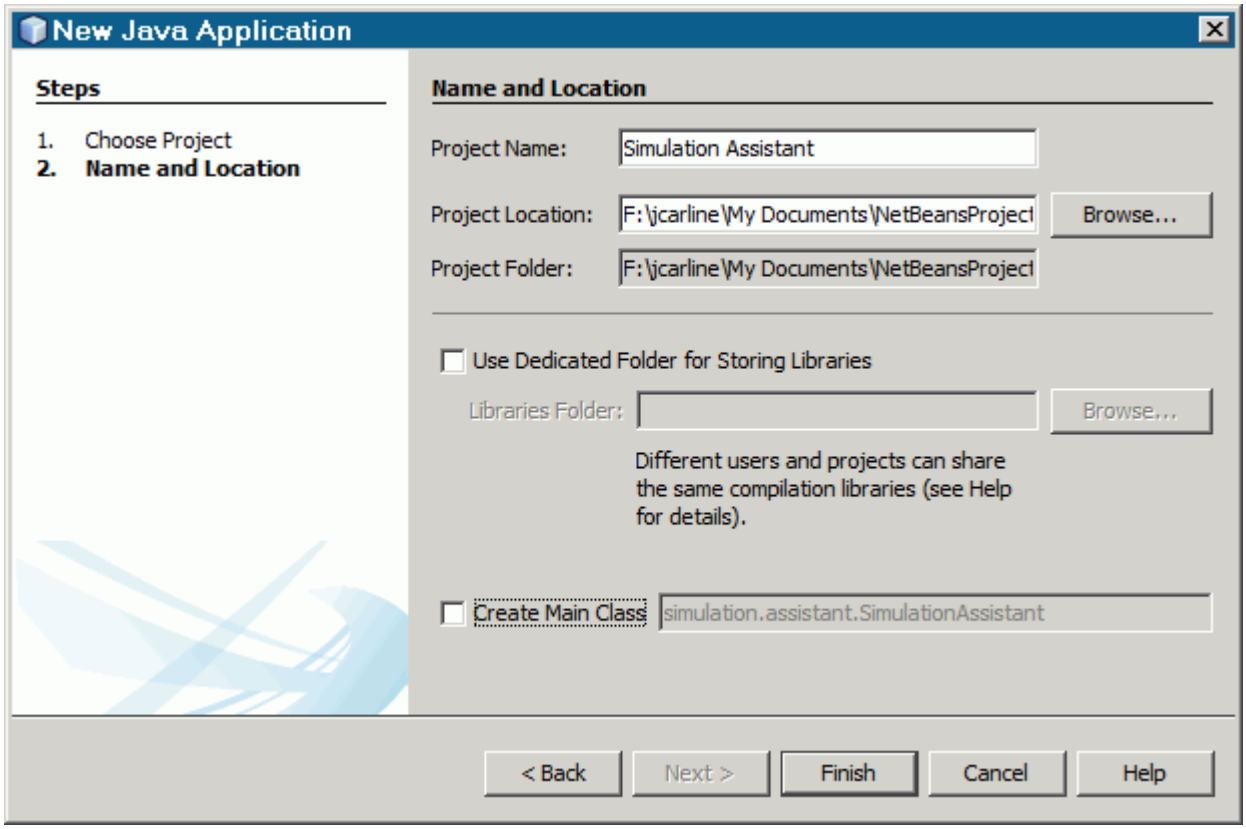
4. Select **Java**from the *Categories* list.
5. Select **Java Application** from the *Projects* list.



6. Click **Next**.

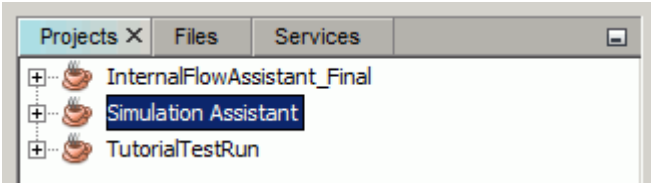
The *New Java Application* dialog appears.

7. In the **Project Name** field, enter the project name.
8. Clear the *Create Main Class* option.



9. Click **Finish**.

The project is added to your projects list. Continue by adding the Java packages and Java classes.



Adding Java Packages, Java Classes, and XHTML Files to the Simulation Assistant Project

Typically, Simulation Assistant projects contain two Java packages, one for the assistant code, and the other for the XHTML text that is displayed in the user interface of the assistant.

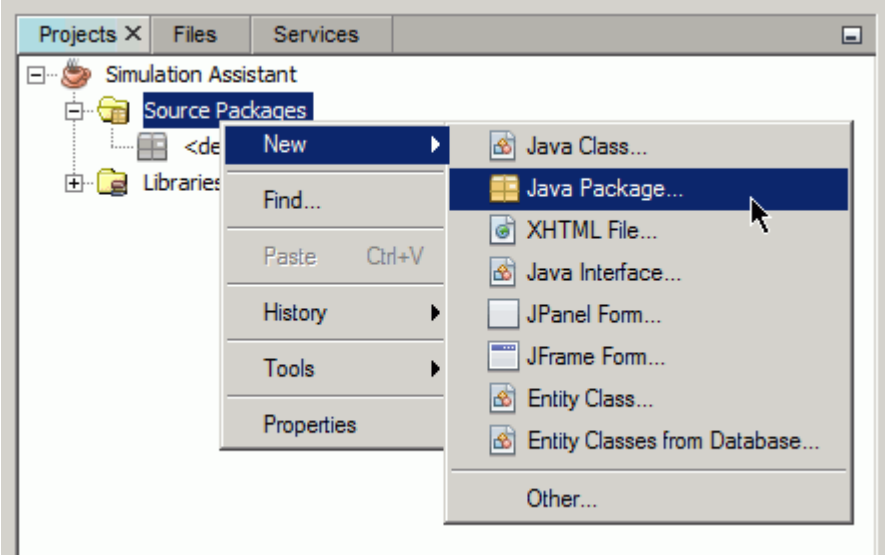
This topic contains the following sections:

- Adding Java Packages to the Project
- Adding Java Classes to the Project
- Adding XHTML Files to the Project

Adding Java Packages to the Project

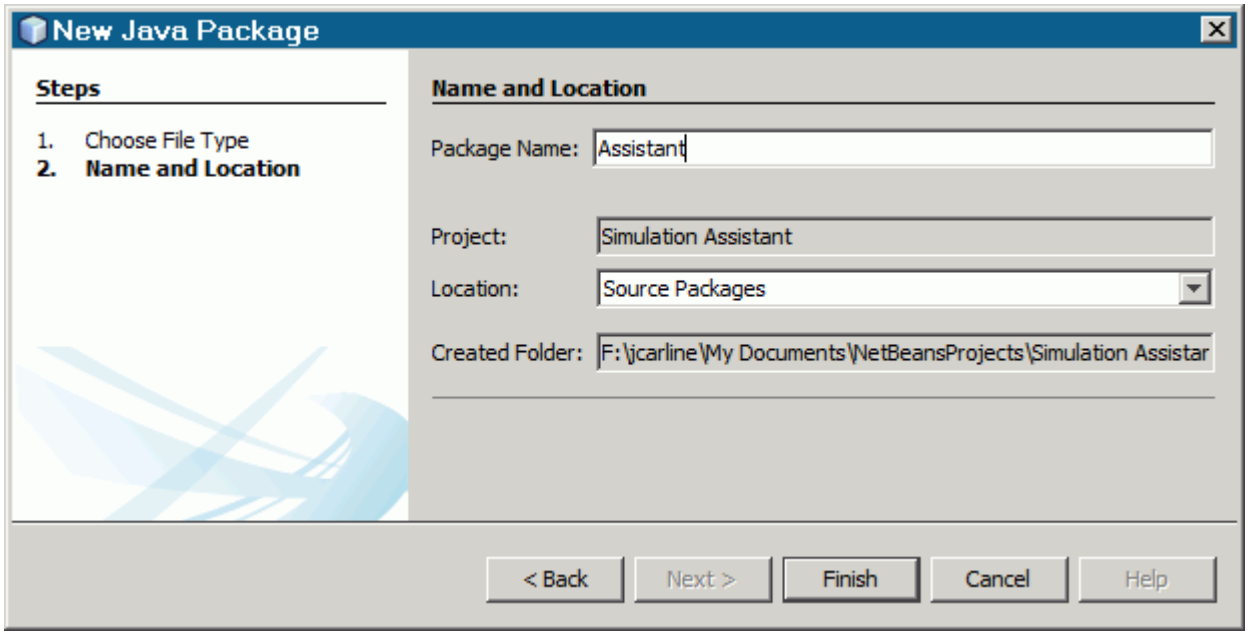
To add Java packages to the Simulation Assistant Project:

1. Right-click on **[project_name]** > **Source Packages** and select **New Java Package...**



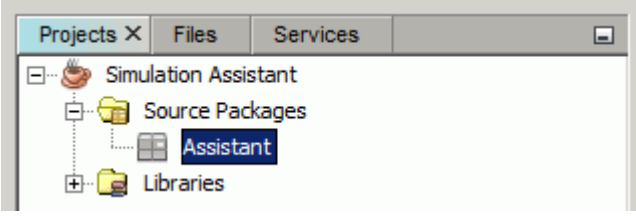
The *New Java Package* dialog appears.

2. In the **Package Name** field, enter the package name.



3. Click **Finish**.

   The Java package is added to the project.

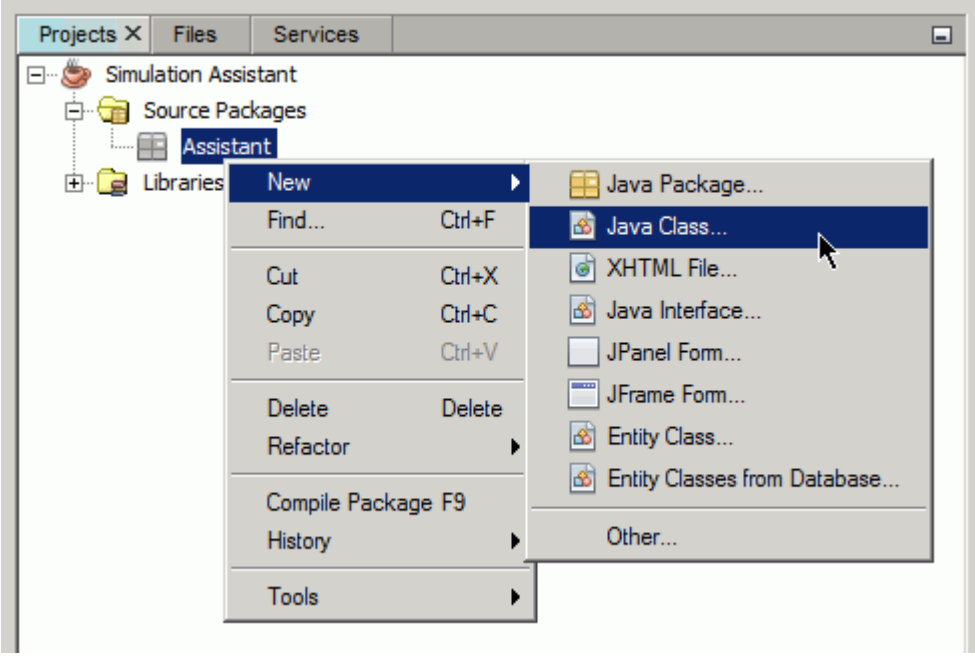


Adding Java Classes to the Project

Java classes contain the code that made up the Simulation Assistant.

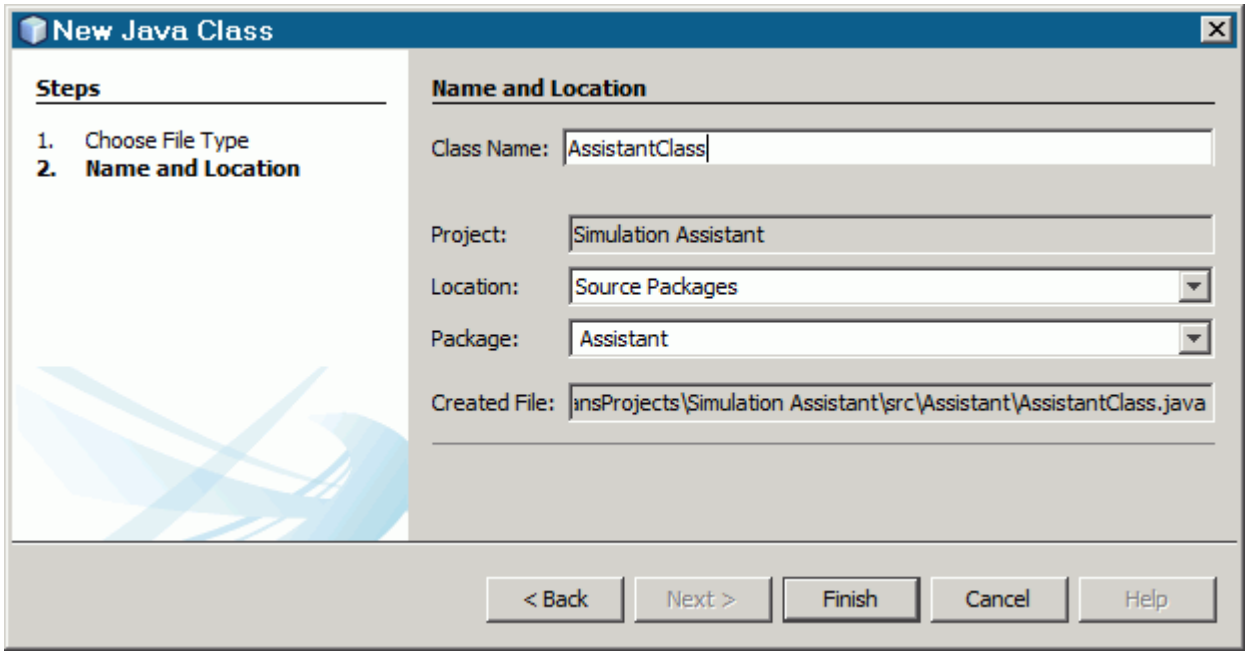
To add a Java class to the Simulation Assistant project:

1. Right-click **[project_name]** > **Source Packages** > **[package_name]** and select **New Java Class...**



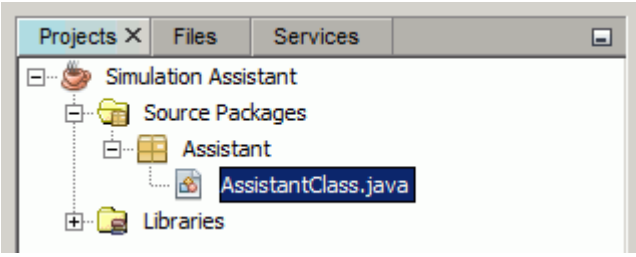
   The *New Java Class* dialog appears.

2. In the **Class Name** field, enter the class name.



3. Click **Finish**.

   The Java class is added to the Java package. Add as many classes as required for your workflow.

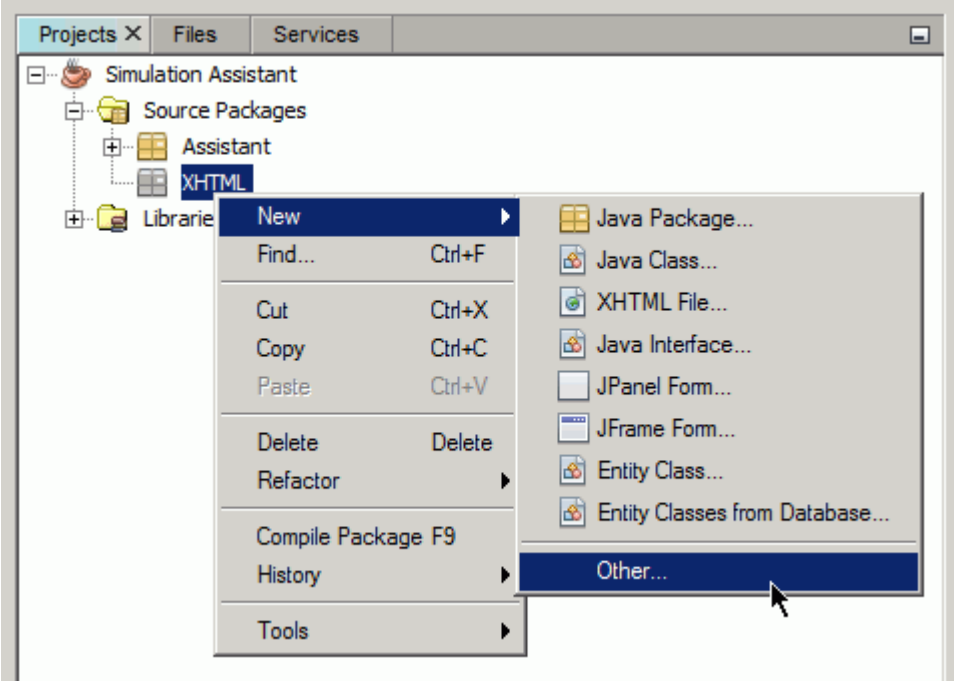


Adding XHTML Files to the Project

Typically, all the text that is displayed in the Simulation Assistant user interface is stored in XHTML files, and is contained in a separate Java class.

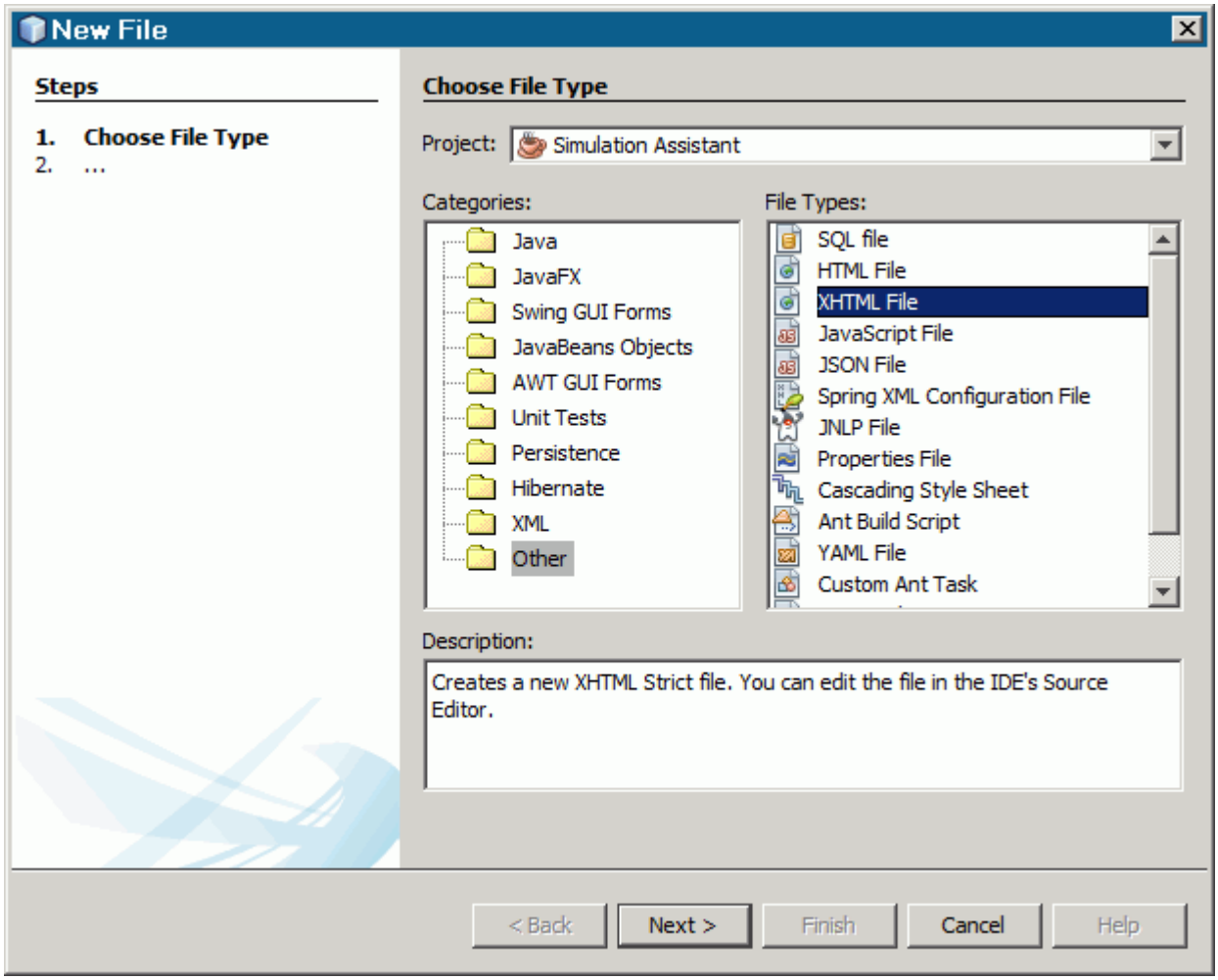
To add an XHTML file to the simulation assistant project:

1. Right-click on **[project_name]** > **Source Packages** > **[package_name]** and select **New** > **Other...**



The *New File* dialog appears.

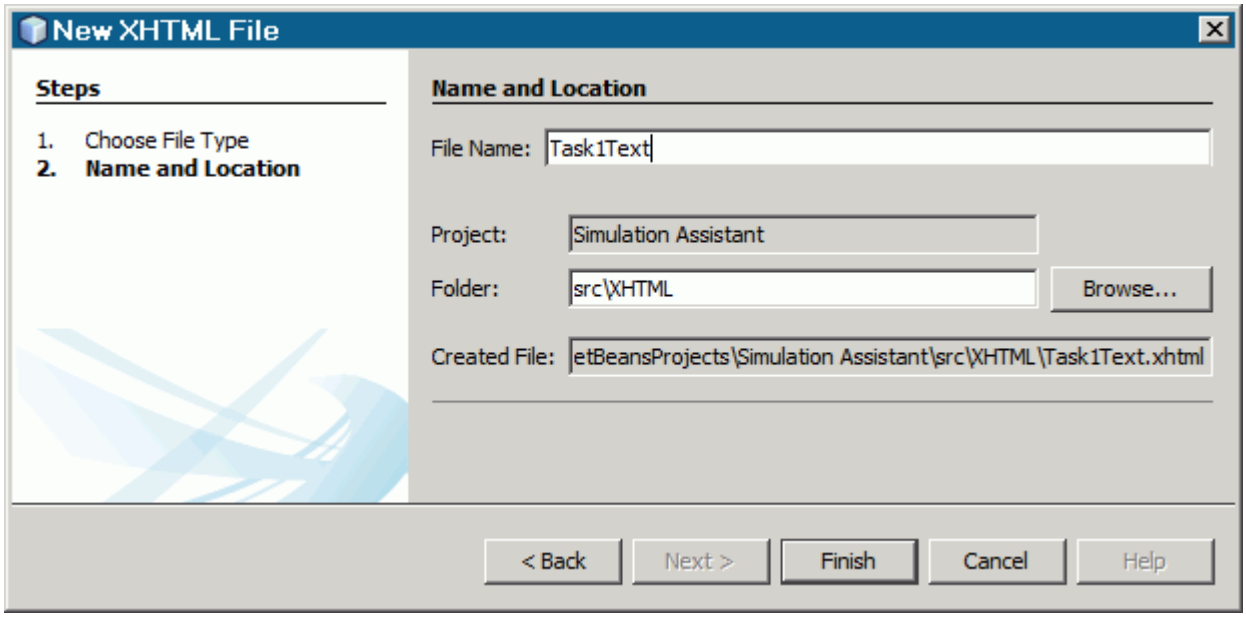
2. Select **Other** from the *Categories* list.
3. Select **XHTML File** from the *File Types* list.



4. Click **Next**.

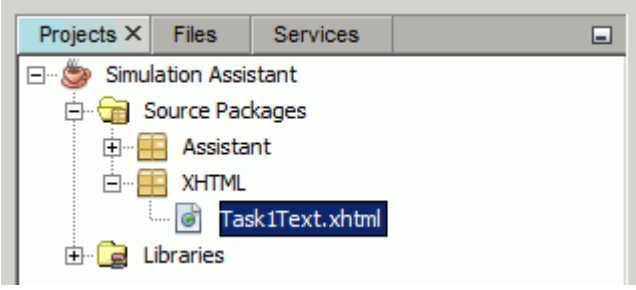
   The *New XHTML File* dialog appears.

5. In the **File Name** field, enter the file name.



6. Click **Finish**.

   The XHTML file is added to the Java package. Add as many XHTML files as required for your workflow.



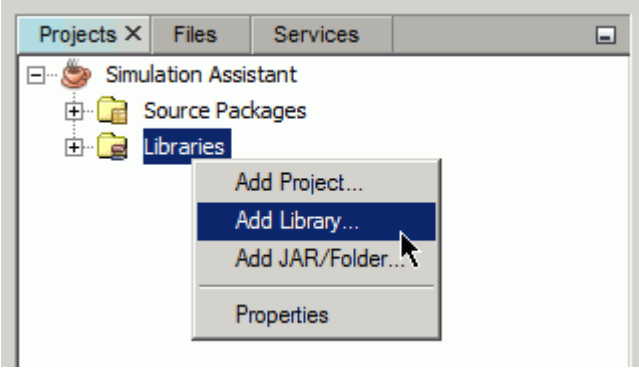
Adding STAR-CCM+ Libraries to the Simulation Assistant Project

You can import the STAR-CCM+ libraries into NetBeans so that you can use the auto-check and auto-complete features of the IDE.

Caution: If you are creating the project on a Windows platform, it is recommended that you copy and paste the STAR-CCM+ libraries to the root directory of your system, for example: `C:\STAR-CCM+ Libraries\`. When building code, Windows has a limitation of how large the path to the libraries can be. In the default STAR-CCM+ installation directory, the STAR-CCM+ libraries would contain paths that are too long. When building the project, these long paths would cause problems.

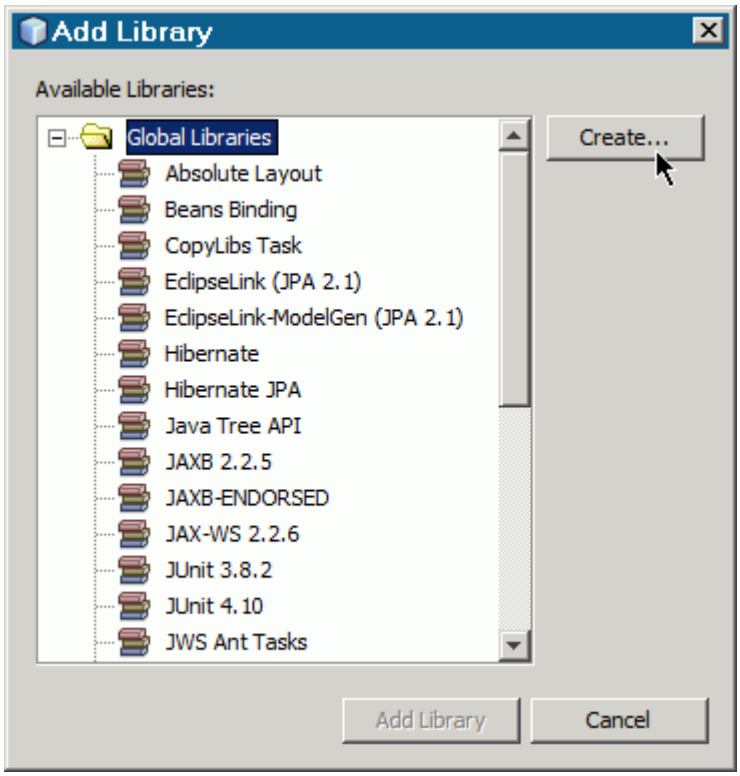
To import the STAR-CCM+ libraries:

1. Right-click on **[project_name]** > **Libraries** and select **Add Library...**.



The *Add Library* dialog appears.

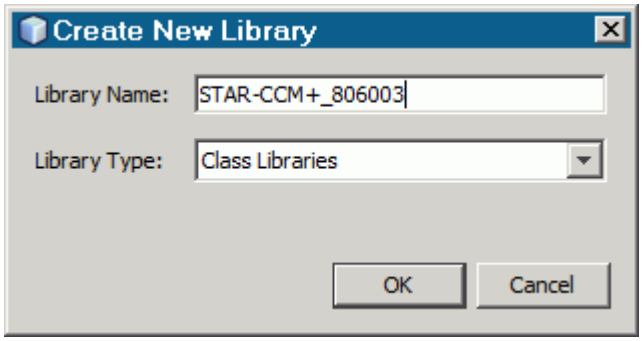
2. Click **Create...**.



The *Create New Library* dialog appears.

3. In the *Library Name* field, enter the name of the library.

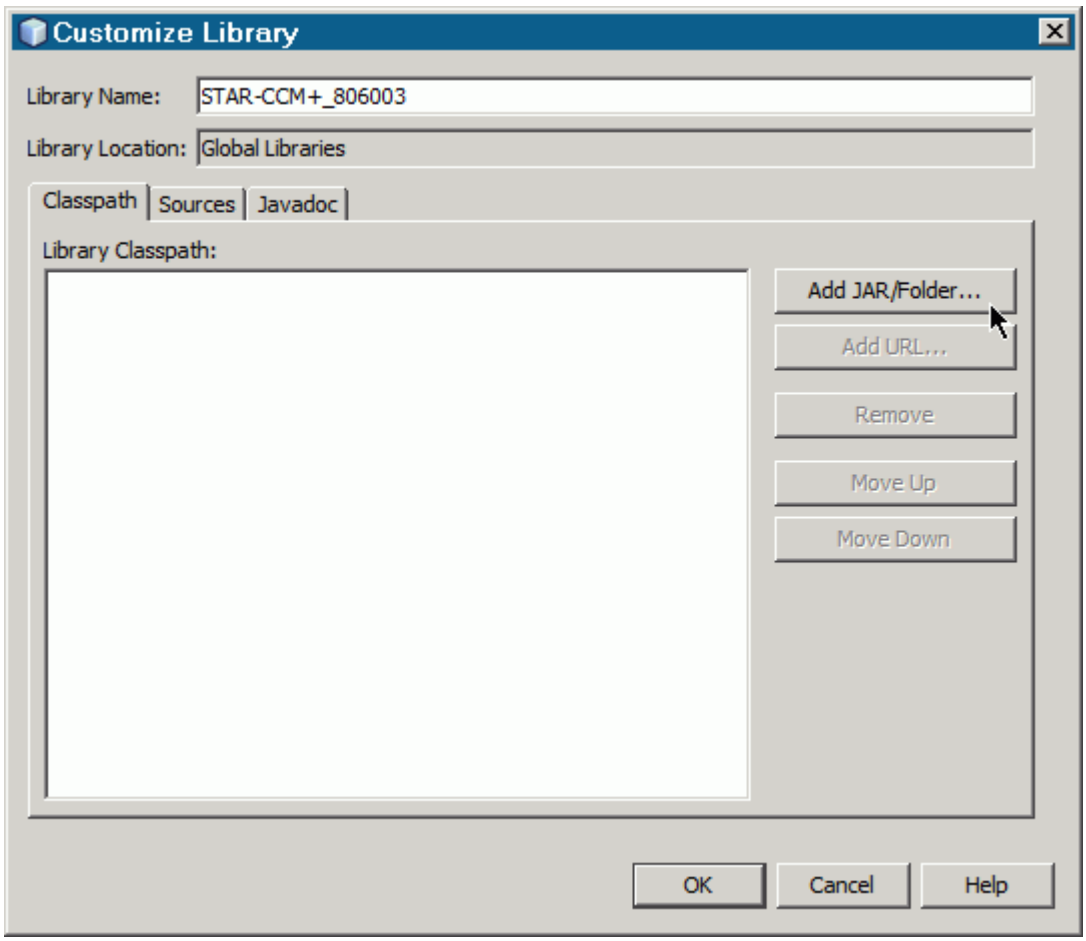
   It is recommended you name the STAR-CCM+ libraries according to the version number, for example `starccm+_804010`.



4. Click **OK**.

The *Customize Library* dialog appears.

5. Click **Add JAR/Folder...**



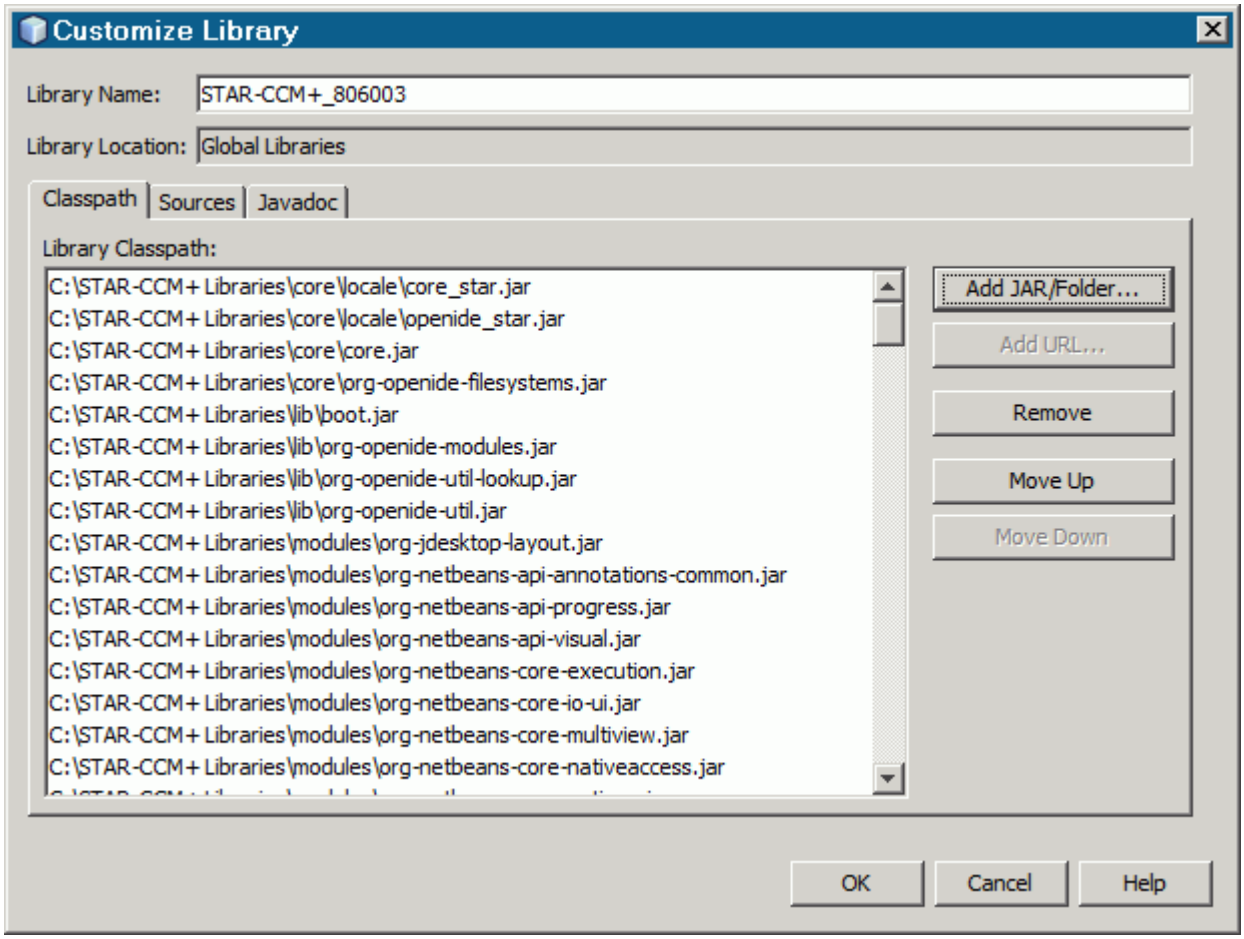
The *Browse JAR/Folder* dialog appears.

6. Navigate to your `[STAR-CCM+_install_dir]\star\lib\java` folder.

Note: If you are going to build projects on Windows platforms, it is recommended that you copy this folder to your root directory, for example: `C:\STAR-CCM+ Libraries\`.

7. Add all the `.jar` files that are located in the following subfolders:
   - `/platform/core`
   - `/platform/core/locale`
   - `/platform/lib`
   - `/platform/modules`
   - `/platform/modules/ext`
   - `/platform/modules/locale`

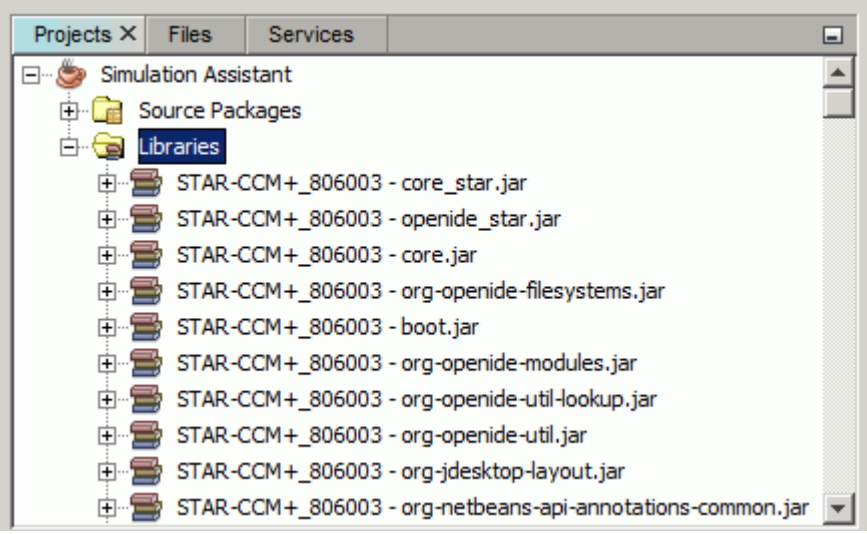
   The libraries that you add are displayed in the *Classpath* tab.



8. If you want to reference the Java documentation, add the following folder:
   - `[STAR-CCM+_install_dir]\doc\client\html`
9. Click **OK** to close the *Customize Library* dialog.
10. Select the library package that you just created and click **Add Library**.

    The *Add Library* dialog closes.
11. Click **OK** to close the *Project Properties* dialog.

The STAR-CCM+ libraries are added to the project.


Projects
Files
Services
Simulation Assistant
Source Packages
Libraries
STAR-CCM+_806003 - core_star.jar
STAR-CCM+_806003 - openide_star.jar
STAR-CCM+_806003 - core.jar
STAR-CCM+_806003 - org-openide-filesystems.jar
STAR-CCM+_806003 - boot.jar
STAR-CCM+_806003 - org-openide-modules.jar
STAR-CCM+_806003 - org-openide-util-lookup.jar
STAR-CCM+_806003 - org-openide-util.jar
STAR-CCM+_806003 - org-jdesktop-layout.jar
STAR-CCM+_806003 - org-netbeans-api-annotations-common.jar

Developing a Simulation Assistant

A Simulation Assistant is a user-defined panel within the STAR-CCM+ user interface. The panel is intended to provide a series of tasks that a user completes in sequence.

This section of the documentation provides documentation on the Java classes and coding that you use to create a Simulation Assistant. This material assumes that you are familiar with Java coding as well as STAR-CCM+. It also assumes that you are using an Integrated Development Environment (IDE) such as NetBeans 7.3.1 or later.

Contents:

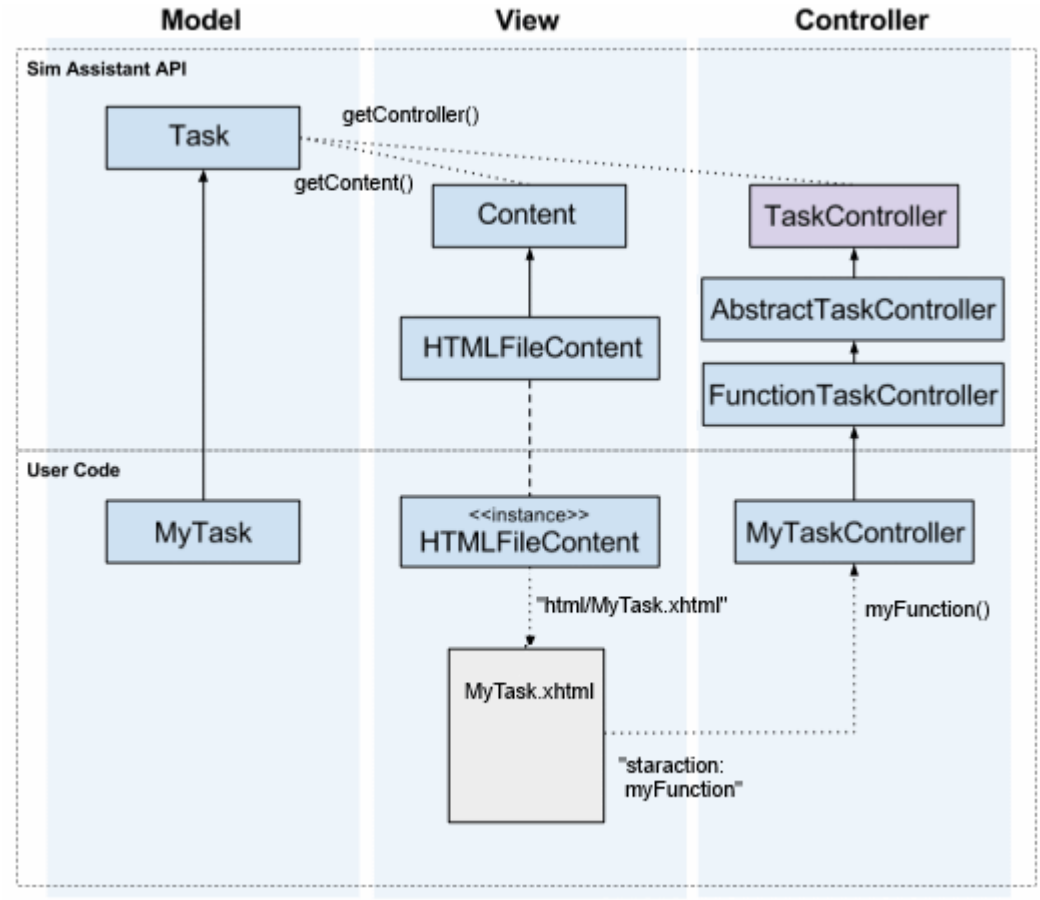
Overview
Files
Tasks and Classes
Star Actions
Annotations
Client Server Objects and Lookups
Tasks
Conditions, Triggers, and Predicates
XHTML
Supplied Classes
Using the Simulation Assistant Wizard in NetBeans

Overview

The Simulation Assistant uses a simple version of the model-view-controller (MVC) design approach to separate the underlying data (model), how the data is represented (view), and how actions are performed (controller).

At the highest level, the Simulation Assistant, you only create a model with simple data (a derivative of the Java class SimulationAssistant). The view and the controller are handled internally.

For an individual task, the model (the Task class), specifies the type of view (the Content class). The model also specifies the controller (a derivative of the TaskController class) to represent and control itself at runtime. The view and controller are instantiated when the Task is created and displayed. For a task, you almost always create a model (a derivative of Task) and a controller (typically a derivative of FunctionTaskController). The type of view is specified using one of the pre-existing Content classes containing or pointing to some XHTML.



As the API was developed to maximize coding flexibility, the model, view, and controller can be extended in a number of ways. This development section emphasizes the expected approach.

For a rudimentary Simulation Assistant, see the Hello World example at:

`[STAR-CCM+_Installation] > star > examples > assistant > HelloWorld`

See also the tutorial Simulation Assistant: Internal Flow Assistant.

Contents:

Model
View
Controller

Model

Task objects are the model objects for a task.

They contain the name, description, state (hidden, disabled, enabled, or complete), a list of subtasks (if any), and pointers to the view and controller to use with the Task (the `getContent()` and `getControllerClass()` functions).

Excluding subtasks, the initial values for all of these items is most easily specified using a `@StarAssistantTask` annotation on your Task subclass. When the data in the model changes (for example, the state transitions from disabled to enabled), the model fires a property change, which the view observes.

View

Content objects link a Task to a view of the Task.

The Simulation Assistant API uses subclasses of Content to determine the correct `TaskContentView` for rendering the Task. Typical Content-derivative choices include `HTMLFileContent`, which points to a complete XHTML file to be displayed, and `HTMLFragmentContent`, which contains a snippet of XHTML to display for the task.

In more advanced situations, you can use a Java UI component such as a JPanel to represent the Task. Write a custom Content-derivative and a custom TaskContentView-derivative, and register the new combination in `TaskContentLoader`.

Controller

The `TaskController` handles user-invoked actions.

The actions that the controller receives from the view are string commands in their purest form. Sometimes these have name/value pair arguments, but the most commonly used TaskController-derivative, `FunctionTaskController`, interprets these commands to be no-argument functions of the `FunctionTaskController` subclass that the author has written.

To use `FunctionTaskController`, your subclass must be public, and the function to execute must be public and take no arguments. For example, a command like "`myFunction`" is represented in the HTML as `<a href="staraction:myFunction">`.When clicked, it executes a function in your `FunctionTaskController` subclass with the signature `public void myFunction()`.

Files

Developing a Simulation Assistant uses the following file types:

- Java files, to contain the code for the assistant.
- JAR files, to wrap and export the Simulation Assistant project as a pre-compiled file.

  You can use **File** > **Load Simulation Assistant** to load assistants directly from either Java or JAR files.
- XHTML files, to contain text and links for the user interface. You can also write this text as inline XHTML, within the Java files of the assistant. To make sure that your XHTML code is well-formed, go to http://validator.w3.org/.
- Image and media files, to augment XHTML text. Use image files for mathematical notation. Supported formats include GIF, JPG, PNG, and SVG.

Tasks and Classes

The Simulation Assistant consists of a list of tasks to be done in order.

Example:

1. Import the geometry
2. Create the regions from the parts
3. Create and set up the physics of the simulation
4. Set up and generating the mesh
5. Visualize the solution
6. Run the simulation

In turn, these tasks often consist of subtasks. Example:

3a. Create physics continuum

3b. Set material properties

3c. Set initial conditions and boundaries

Set up the structure of classes in the Java code of the assistant to parallel the task structure, so that each high-level class performs one high-level class. Classes for subtasks are called by the top level tasks. See Supplied Classes for the classes that STAR-CCM+ provides.

Use the following rules:

- Create a Java file containing the class for your Simulation Assistant, extended from `SimulationAssistant`. See SimulationAssistant Class.
- Within your Simulation Assistant class, create a list of tasks for the top-level outline of the Simulation Assistant.
- Create a series of Java files, one for each task, extended from `Task`. See Task Class.
- Create XHTML code for each task, supplying the text and links for the user interface.
  - For extended text, use a separate XHTML file. Link to it with the `contentPath` parameter of the annotation for the task, or pass an HTMLFileContent object to the `setContent()` method of the task.
  - For small sections of XHTML code, specify them with the `contentPath` parameter of the annotation for the task, or pass an HTMLFragmentContent object to the `setContent()` method of the task.
- Define the actions for the tasks in extensions of AbstractTaskController.
- Where necessary, define the pre-conditions and post-conditions for the tasks, using Condition, ConditionTrigger, and their extentions.

Star Actions

A Star Action is an action that is performed in the STAR-CCM+ interface.

The Star Action links a word or phrase in XHTML text to a function named in the task. The assistant performs the Star Action when the user clicks the text.

Star Action XHTML code has the format:

```
<a href="staraction:function_name">action_text</a>
```

Examples:

```
<a href="staraction:createPhysicsContinuum">Create</a>
```

```
<a href="staraction:materialProperties">Define</a>
```

The `action_text` strings (`Create` and `Define` in the examples) appear as the link text presented by the Simulation Assistant:

1. Create a physics continuum
2. Define the material properties

The `function_name` strings (`createPhysicsContinuum` and `materialProperties` in the examples) are the names of Java functions that are used in the task.

Annotations

The Simulation Assistant provides two annotations for generic set methods:

- `@StarAssistant`, for use in extensions of the SimulationAssistant Class class, sets the values of `desc`, `display`, and `outline`.

  See: **Help** > **Java API** > **star.assistant.annotation** > **StarAssistant**
- `@StarAssistantTask`, for use in extensions of the Task Class class, sets the values of `contentFragment`, `contentPath`, `controller`, `display`, `id`, and `state`.

  See: **Help** > **Java API** > **star.assistant.annotation** > **StarAssistantTask**

Client Server Objects and Lookups

Client Server Objects (CSOs) are any of the items that are represented in the STAR-CCM+ simulation tree.

A model, an interface, a mesh, an injector, a property, a field function, are all examples of CSOs. It is a very broad category.

A lookup is a search tool that is assigned to a list of CSOs associated with a task. If you set a task lookup to search for a CSO and it fails to find the CSO in its own list, the lookup searches the lists of related tasks. It checks, in order, preceding siblings, parent, preceding siblings of the parent, grandparent, and so on, up to the lookup list for the whole Simulation Assistant.

Put CSOs on the lookup list when you expect to use them again in the task, or in a parent or child of the task. A CSO that satisfies a Condition is automatically added to the lookup. See Conditions, Triggers, and Predicates.

The lookup-related functions, provided in AbstractTaskController, are:

Function	Description
`addToTaskLookup(cso)`	Puts a CSO on the lookup list, where cso is the object in question.
`lookupObject(class)`	Searches for a CSO on the lookup list hierarchy, where class is the class of the object you are looking for. The lookup returns the first object of the class.
`lookupObjects(class)`	Searches for all CSOs of a class on the lookup list hierarchy, where class is the class of the object you are looking for. The lookup returns all objects in the class.
`getLookup()`	Returns the entire lookup object.
`printLookup()`	Prints the lookup contents to the output window.
`removeFromTaskLookup(cso)`	Removes a CSO from the lookup list, where cso is the object in question.
`clearTaskLookup()`	Clears the lookup list of all CSOs.

For coding specifics, see the STAR-CCM+ **Help** menu:

Help > **Java API** > **star.assistant.ui** > **AbstractTaskController**

Contents:

Guidelines

Guidelines

Add a CSO to the lookup when you expect to use it with a different action, particularly when that action is in the same task or in a parent or child task. The lookup is also critical for many pre- and post-condition tests, so if a CSO being tested in a condition has been successfully set up, add it to the lookup.

A CSO that satisfies a Condition is automatically added to the lookup.

Tasks

The Simulation Assistant is composed of Tasks, each in a separate Java file.

Each Task controls a separate panel in the Simulation Assistant window. Task panels display text, including links that let users perform Star Actions.

Tasks share data by putting it in their lookups, where other tasks find it when they use the hierarchical search in `lookupObject()`.

A Task can have a list of Pre-conditions, a list of Post-conditions, both, or neither.

- If there is a list of Pre-conditions, all the conditions on the list must be satisfied before the Task is enabled.

- If there is a list of Post-conditions, all the conditions on the list must be satisfied before the Task is marked as complete.

Contents:

Branching and Looping

See Also:

Task Class
Conditions, Triggers, and Predicates

Branching and Looping

Tasks can branch and loop.

To use branching, make a Task with two or more subtasks. These subtasks remain hidden until a choice is made in the parent Task, usually by clicking a link. The choice activates one of the branches, making the chosen subtask visible and leaving all of the other subtasks hidden.

See `\star\examples\assistant\SimpleBranch`

To use looping, use LoopTask, LoopingCondition, and LoopIterationTrigger.

See `\star\examples\assistant\SimpleLoop`

Conditions, Triggers, and Predicates

Use Conditions to control what tasks are enabled or disabled in the Simulation Assistant, and to make sure that the user creates a valid simulation.

Conditions are classes that track objects meeting certain criteria. When a Condition finds such an object, it *holds* the object (records a reference to it) and sets its *Satisfied* property to `true`. If the object is deleted, or if its properties or the properties of its children change to no longer meet the criteria, the Condition stops holding and sets its *Satisfied* property to `false`.

A Condition hunts for objects using a set of Triggers. A Trigger monitors a set of objects, such as those in the lookup hierarchy. When it finds the type of object that is specified for it, it passes the object to the Condition for evaluation.

A Condition makes the evaluation using a Predicate. If the object meets the criteria for the Predicate, the Predicate returns `true` and the Condition sets its *Satisfied* property to `true`.

A Condition has one Predicate, and the Predicate evaluates one object at a time. However, the object can contain other objects or point to other objects (for example, if the passed object is a manager node). The Predicate can evaluate the object with combinations of logical operations using the CompoundPredicate extension.

See:

- Condition
  - subclass CSOCondition

- ConditionTrigger (abstract class)
  - subclass CSOConditionTrigger
  - subclass LookupConditionTrigger
    - sub-subclass CSOLookupConditionTrigger
  - subclass LoopIterationTrigger
  - subclass TaskConditionTrigger
- Predicate
  - subclass CompoundPredicate

XHTML

The text appearing in a task panel is coded in XHTML.

You can write the text in either of two ways: by using XHTML formatted text strings (XHTML fragments), or by referencing XHTML files in the Java files.

Use formatted text strings for brief messages and referenced XHTML files for extended text.

This example uses the XHTML fragment format:

```
@StarAssistantTask(display="Hello Task",
contentFragment="<ul><li>
<a href=\"staraction:hello\">Hello</a>!</li></ul>",
controller=HelloWorld.HelloWorldTask.
HelloWorldTaskController.class)
```

This example uses an XHTML file:

```
@StarAssistantTask(display="Create A Continuum With
Pre-defined Models"
contentPath="html/CreateContinuaTask.xhtml",
controller=CreateContinuaTask.
CreateContinuaTaskController.class)
```

XHTML text can be used only once in a task panel. However, a task panel can display any one of several alternate passages, depending on conditions. For situations like this one, you can use the HTMLFileContent and HTMLFragmentContent classes to specify the alternate text.

Supplied Classes

This section describes the classes that are supplied by STAR-CCM+.

In most cases, most or all of the classes in your Simulation Assistant are extensions of these classes. The supplied classes are:

High-level classes:

- SimulationAssistant Class

- Task Class
- LoopTask

Task controller classes:

- AbstractTaskController
- TaskController
- FunctionTaskController

Content classes:

- Content
- HTMLFileContent
- HTMLFragmentContent

Condition classes:

- Condition
- CSOCondition
- LoopingCondition

Condition trigger classes:

- ConditionTrigger
- CSOConditionTrigger
- LookupConditionTrigger
- CSOLookupConditionTrigger
- LoopIterationTrigger
- TaskConditionTrigger

Predicate classes:

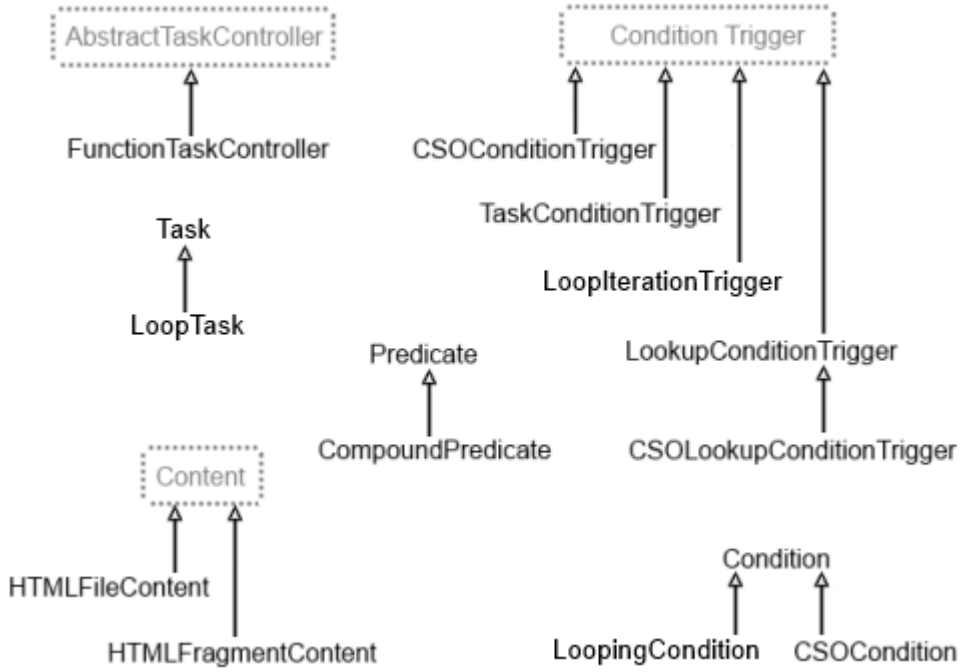
- Predicate
- CompoundPredicate

Contents:

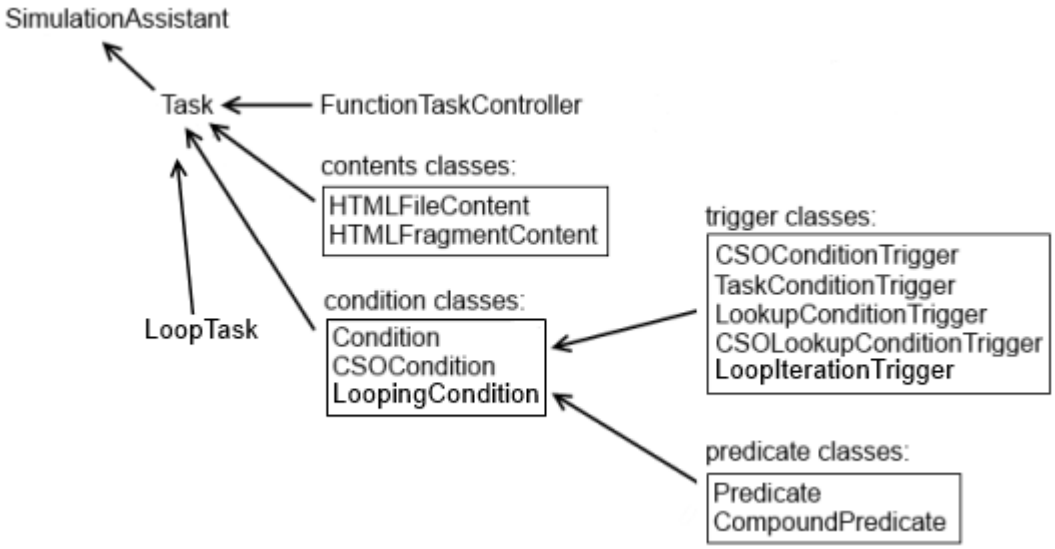
Class Relationships
SimulationAssistant Class
Task Class
Task Controller Classes
Content Classes
Condition Classes
Condition Trigger Classes
Predicate Classes

Class Relationships

The following diagram shows inheritance relationships between the provided classes:



The following diagram shows usage relationships between the provided classes:



SimulationAssistant Class

Extend this class to create the top-level class for your Simulation Assistant.

The top-level class contains the title of the Simulation Assistant and a list of the constituent tasks, which are given in order of execution. See Task Class.

A SimulationAssistant must contain the following items:

- Imports:
  - star.assistant.SimulationAssistant
  - star.assistant.Task
- A constructor with the name of the assistant.
- A list of tasks

A SimulationAssistant often contains these items:

- The annotation `@StarAssistant`
- Imports:
  - java.util.ArrayList
  - java.util.List
  - star.assistant.annotation.StarAssistant
  - star.assistant.annotation.StarAssistantTask
  - star.assistant.ui.FunctionTaskController

Example:

```
import java.util.ArrayList;
 import java.util.List;
 import star.assistant.SimulationAssistant;
 import star.assistant.Task;
 import star.assistant.annotation.StarAssistant;

@StarAssistant(display = "Internal Flow Assistant")
 public class InternalFlowAssistant extends
 SimulationAssistant {

public InternalFlowAssistant() {
...
    }
}
```

Online API:

For coding specifics, see the STAR-CCM+ **Help** menu:

Help > **Java API** > **star.assistant** > **SimulationAssistant**

Task Class

Use extensions of this class to create the classes for individual tasks in your Simulation Assistant, one class for each task pane in the assistant. The Task displays text and allows the user to perform specified Star Actions.

The Task can have a list of pre-conditions that must be satisfied before the user can perform the task. The `setPreconditions()` function receives this list.

Independently, the Task can have a list of post-conditions that must be satisfied before the user can complete the function and continue. The `setPostconditions()` function receives this list.

A Task must use import star.assistant.Task

A Task often uses these items:

- @StarAssistantTask
- import star.assistant.annotation.StarAssistantTask
- import star.assistant.ui.FunctionTaskController
- setPreconditions()
- setPostconditions()

Example:

```
package Assistant;

import star.assistant.Task;
import star.assistant.annotation.StarAssistantTask;
import star.assistant.ui.FunctionTaskController;
...

@StarAssistantTask(display = "Run the Simulation",
  contentPath = "HTML/06_RunSimulation.xhtml",
    controller = Task06RunTheSimulation
      .RunTaskController.class)

public class Task06RunTheSimulation extends Task {

  public class RunTaskController extends
    FunctionTaskController {

...

    }

}
```

Online API:

For coding specifics, see the STAR-CCM+ **Help** menu:

Help > **Java API** > **star.assistant** > **Task**

Contents:

LoopTask

LoopTask

LoopTask is an extension of the Task Class.

Use extensions of this class to create a basic task that keeps a counter on a defined action and notifies the user when a predefined condition has been met. Use `LoopingCondition` in `LoopTasks`, to control the number of repetitions. You can also use any other Conditions.

Example:

```
import star.assistant.LoopIterationTrigger;
 import star.assistant.LoopTask;
 import star.assistant.LoopingCondition;
 import star.assistant.annotation.StarAssistantTask;
 import star.assistant.ui.FunctionTaskController;

...

@StarAssistantTask(display="Create Boundaries for a Region",
                   contentPath="html/CreateBoundariesTask.xhtml",

controller=CreateBoundariesTask.CreateBoundariesTaskController.class)
public class CreateBoundariesTask extends LoopTask<Boundary> {


    public CreateBoundariesTask() {

...

        //loop condition
         LoopingCondition<Boundary> loopCondition = new
LoopingCondition<Boundary>();
         loopCondition.setDesc("At least 3 boundaries.");
         LoopIterationTrigger<Boundary> loopTrigger = new
LoopIterationTrigger<Boundary>(3);
         loopCondition.setLoopTrigger(loopTrigger);
         loopCondition.setPredicate(new Predicate<List<Boundary>>() {


            @Override
             public boolean evaluate(List<Boundary> boundaries) {

...

                return false;
                 }

...
```

```
                }

                return true;
            }
        });

        setLoopCondition(loopCondition);

...
    }
...
}
```

Online API:

For coding specifics, see the STAR-CCM+ **Help** menu:

Help > **Java API** > **star.assistant** > **LoopTask**

Task Controller Classes

These classes handle user input.

Contents:

AbstractTaskController
TaskController
FunctionTaskController

AbstractTaskController

FunctionTaskController inherits functions from this class. `AbstractTaskController` is an abstract class implementing TaskController.

Online API:

For coding specifics, see the STAR-CCM+ **Help** menu:

Help > **Java API** > **star.assistant.ui** > **AbstractTaskController**

TaskController

FunctionTaskController inherits functions from this class. `TaskController` is an interface that is implemented by AbstractTaskController.

Online API:

For coding specifics, see the STAR-CCM+ **Help** menu:

Help > **Java API** > **star.assistant** > **TaskController**

FunctionTaskController

`FunctionTaskController` is an extension of AbstractTaskController. Extend it to create a controller that defines functions that are called when the user clicks a hyperlink. The functions execute code sequentially (including macro code); that is, operations are synchronous.

A FunctionTaskController must use the following items:

- import star.assistant.Task
- import star.assistant.ui.FunctionTaskController
- A function name matching a Star Action name in the associated XHTML file.

A FunctionTaskController often uses these items:

- @StarAssistantTask
- import star.assistant.annotation.StarAssistantTask

Example:

```
public class CreateBoundaryTaskController extends
FunctionTaskController {

public void createBoundary() {
Region region = lookupObject(Region.class);
Boundary boundary = region.getBoundaryManager().createEmptyBoundary();
boundary.setBoundaryType(InletBoundary.class);
addToTaskLookup(boundary);
}
}
```

Online API:

For coding specifics, see the STAR-CCM+ **Help** menu:

Help > **Java API** > **star.assistant.ui** > **FunctionTaskController**

Content Classes

These classes control the XHTML text that the assistant displays.

Contents:

Content
HTMLFileContent
HTMLFragmentContent

Content

HTMLFileContent and HTMLFragmentContent inherit functions from this abstract class.

Online API:

For coding specifics, see the STAR-CCM+ **Help** menu:

Help > **Java API** > **star.assistant** > **Content**

HTMLFileContent

Use this class to reference an XHTML file when you cannot use an annotation.

For instance, a task sometimes must display different text depending on varying conditions. In such a case, you can use this class to reference the alternate text. If the amount of text is small, use HTMLFragmentContent.

Example:

```
Content content = new HTMLFileContent();
content.setValue("CCWorkflow1_MeshSetup.xml");
setContent(content);
```

Online API:

For coding specifics, see the STAR-CCM+ **Help** menu:

Help > **Java API** > **star.assistant** > **HTMLFileContent**

HTMLFragmentContent

Use this class to display text in a task panel when you cannot use the text string in an annotation.

If the amount of text is large, use HTMLFileContent.

Example:

```
Content c = new HTMLFragmentContent();
c.setValue("<ul><li>
<a href=\"staraction:hello\">Hello</a>!</li></ul>");
this.setContent(c);
```

Online API:

For coding specifics, see the STAR-CCM+ **Help** menu:

Help > **Java API** > **star.assistant** > **HTMLFragmentContent**

Condition Classes

These classes control what tasks are enabled or disabled in the assistant.

Contents:

Condition
CSOCondition
LoopingCondition

Condition

Use extensions of this class to monitor a set of objects for potentially relevant objects.

A Condition uses one or more ConditionTrigger extentions and a Predicate extension. When the type of object that is assigned to a `ConditionTrigger` appears, the Condition evaluates the object using the Predicate. If the Predicate returns the value `true`, the `ConditionTrigger` stops monitoring other objects, the Condition holds the object, and sets its *Satisfied* property to `true`.

If the Predicate returns the value `false`, the Condition sets its *Satisfied* property to `false` and continues monitoring the set of objects.

This class has CSOCondition as a subclass.

Example:

```
Condition<String> cnd = new Condition<String>();
cnd.setDesc("A string containing \"Cat\"");
LookupConditionTrigger<String> cndTrigger =
new LookupConditionTrigger<String>(String.class);
cnd.setTriggers(Collections.singleton(cndTrigger));
cnd.setPredicate(new Predicate<String>() {
@Override
public boolean evaluate(String string) {
if (string.contains("Cat")) {
return true;
}

return false;
}
});
```

Online API:

For coding specifics, see the STAR-CCM+ **Help** menu:

Help > **Java API** > **star.assistant** > **Condition**

See Also:

Conditions, Triggers, and Predicates

CSOCondition

`CSOCondition` is an extension of Condition. Use extensions of this class to monitor conditions that are associated with CSOs. `CSOCondition` watches for the addition and deletion of the CSO or its children, in addition to property changes.

Example:

```
CSOCondition<PhysicsContinuum> cnd = new
CSOCondition<PhysicsContinuum>();
cnd.setDesc("The physics continuum must have the Radiation model
enabled.");
CSOLookupConditionTrigger<PhysicsContinuum> cndTrigger =
new
CSOLookupConditionTrigger<PhysicsContinuum>(PhysicsContinuum.class);
cnd.setTriggers(Collections.singleton(cndTrigger));
cnd.setPredicate(new Predicate<PhysicsContinuum>() {
@Override
public boolean evaluate(PhysicsContinuum continuum) {

if (continuum.getModelManager().hasModel(
RadiationModel.class) != null) {
return true;
}

return false;
}
});
```

Online API:

For coding specifics, see the STAR-CCM+ **Help** menu:

Help > **Java API** > **star.assistant** > **CSOCondition**

See Also:

Conditions, Triggers, and Predicates

LoopingCondition

`LoopingCondition` is an extension of Condition. Use extensions of this class in a LoopTask to monitor the state of a LoopIterationTrigger.

Example:

```
 //loop condition
LoopingCondition<Boundary> loopCondition = new
LoopingCondition<Boundary>();
loopCondition.setDesc("At least 3 boundaries.");
LoopIterationTrigger<Boundary> loopTrigger = new
LoopIterationTrigger<Boundary>(3);
loopCondition.setLoopTrigger(loopTrigger);
loopCondition.setPredicate(new Predicate<List<Boundary>>() {

    @Override
    public boolean evaluate(List<Boundary> boundaries) {
...
        }

        return true;
    }
});
setLoopCondition(loopCondition);
```

Online API:

For coding specifics, see the STAR-CCM+ **Help** menu:

Help > **Java API** > **star.assistant** > **LoopingCondition**

See Also:

Conditions, Triggers, and Predicates

Condition Trigger Classes

These classes work within Conditions and monitor objects.

Contents:

ConditionTrigger
CSOConditionTrigger
LookupConditionTrigger
CSOLookupConditionTrigger
LoopIterationTrigger
TaskConditionTrigger

ConditionTrigger

Use extensions of this class in Conditions to monitor system events and detect when potentially relevant objects become available. When the type of object for the `ConditionTrigger` appears, the trigger passes

the object to the Condition for evaluation. If the Condition is satisfied and holds the object, it deactivates the trigger.

`ConditionTrigger` is an abstract class. It has CSOConditionTrigger, CSOLookupConditionTrigger, and TaskConditionTrigger as subclasses.

Online API:

For coding specifics, see the STAR-CCM+ **Help** menu:

Help > **Java API** > **star.assistant** > **ConditionTrigger**

See Also:

Conditions, Triggers, and Predicates

CSOConditionTrigger

`CSOConditionTrigger` is an extension of ConditionTrigger. Use extensions of this class in Conditions to monitor conditions that are associated with a particular CSO and its children.

Example:

```
public CSOCondition<PhysicsContinuum> createCondition(PhysicsContinuum
pc) {
CSOCondition<PhysicsContinuum> cnd = new
CSOCondition<PhysicsContinuum>();
cnd.setDesc("The physics continuum must have the Radiation model
enabled.");
CSOConditionTrigger<PhysicsContinuum> cndTrigger =
new CSOConditionTrigger<PhysicsContinuum>(pc);
cnd.setTriggers(Collections.singleton(cndTrigger));
cnd.setPredicate(new Predicate<PhysicsContinuum>() {
@Override
public boolean evaluate(PhysicsContinuum continuum) {
if (continuum.getModelManager().hasModel(
RadiationModel.class) != null) {
return true;
}
return false;
}
});
return cnd;
}
```

Online API:

For coding specifics, see the STAR-CCM+ **Help** menu:

Help > **Java API** > **star.assistant** > **CSOConditionTrigger**

See Also:

Conditions, Triggers, and Predicates

LookupConditionTrigger

`LookupConditionTrigger` is an extension of ConditionTrigger. Use extensions of this class in Conditions to monitor the task Lookup for the addition and removal of objects of a particular class.

Example:

```
CSOCondition<PhysicsContinuum> cnd = new
CSOCondition<PhysicsContinuum>();
cnd.setDesc("The physics continuum must have the Radiation model
enabled.");
CSOLookupConditionTrigger<PhysicsContinuum> cndTrigger =
new
CSOLookupConditionTrigger<PhysicsContinuum>(PhysicsContinuum.class);
cnd.setTriggers(Collections.singleton(cndTrigger));
cnd.setPredicate(new Predicate<PhysicsContinuum>() {
@Override
public boolean evaluate(PhysicsContinuum continuum) {
if (continuum.getModelManager().hasModel(
RadiationModel.class) != null) {
return true;
}
return false;
}
});
```

Online API:

For coding specifics, see the STAR-CCM+ **Help** menu:

Help > **Java API** > **star.assistant** > **LookupConditionTrigger**

See Also:

Conditions, Triggers, and Predicates

CSOLookupConditionTrigger

`CSOLookupConditionTrigger` is an extension of LookupConditionTrigger. Use extensions of this class in Conditions to monitor CSOs in the task Lookup. The trigger watches for the addition and deletion of the CSO or its children, in addition to property changes. The trigger passes the object to the Condition only when these changes occur.

Example:

```
import star.assistant.CSOLookupConditionTrigger;
...
public class ModifyPolynomialDensityTask extends Task {
    public ModifyPolynomialDensityTask() {

    //add a precondition
    //precondition: unavailable unless Polynomial in T exists.
    CSOCondition<Continuum> continuumCondition = new
CSOCondition<Continuum>();
    continuumCondition.setDesc("A liquid physics continuum with a
\"Polynomial in T\" method assigned for the Polynomial Density
property.");
    CSOLookupConditionTrigger<Continuum> continuumConditionTrigger =
new CSOLookupConditionTrigger<Continuum>(Continuum.class);
```

Online API:

For coding specifics, see the STAR-CCM+ **Help** menu:

Help > **Java API** > **star.assistant** > **CSOLookupConditionTrigger**

See Also:

Conditions, Triggers, and Predicates

LoopIterationTrigger

`LoopIterationTrigger` is an extension of ConditionTrigger. Use extensions of this class in LoopingConditions to monitor a list of objects whenever an object is added to the list, to determine if a LoopTask repeats.

Example:

```
 //loop condition
LoopingCondition<Boundary> loopCondition = new
LoopingCondition<Boundary>();
loopCondition.setDesc("At least 3 boundaries.");
LoopIterationTrigger<Boundary> loopTrigger = new
LoopIterationTrigger<Boundary>(3);
loopCondition.setLoopTrigger(loopTrigger);
loopCondition.setPredicate(new Predicate<List<Boundary>>() {
    @Override
    public boolean evaluate(List<Boundary> boundaries) {
...
        }

        return true;
```

```
    }
});
setLoopCondition(loopCondition);
```

Online API:

For coding specifics, see the STAR-CCM+ **Help** menu:

Help > **Java API** > **star.assistant** > **LoopIterationTrigger**

TaskConditionTrigger

Use extensions of this class in Conditions to monitor a task for property changes. This way, tasks can monitor each other.

Example:

```
public Condition<Task> createCondition(Task task) {
Condition<Task> cnd = new Condition<Task>();
cnd.setDesc("The task must be complete.");
TaskConditionTrigger cndTrigger = new
TaskConditionTrigger(task);
cnd.setTriggers(Collections.singleton(cndTrigger));
cnd.setPredicate(new Predicate<Task>() {
@Override
public boolean evaluate(Task task) {
if (task.getState() == Task.TaskState.COMPLETE) {
return true;
}
return false;
}
});
return cnd;
}
```

Online API:

For coding specifics, see the STAR-CCM+ **Help** menu:

Help > **Java API** > **star.assistant** > **TaskConditionTrigger**

See Also:

Conditions, Triggers, and Predicates

Predicate Classes

These classes work within Conditions and evaluate objects.

Contents:

Predicate
CompoundPredicate

Predicate

Use extensions of this class in Conditions to evaluate objects. The Predicate evaluation function returns `true` or `false`, and all classes that implement Predicate must define this function. Each Condition has one Predicate, which evaluates a single object, but the object itself can contain other objects or point to other objects, and the Predicate can base its evaluation on these objects. The Predicate does not carry over any information from one evaluation to the next. Do not use it to alter any objects or properties in its evaluation code.

Example:

```
physicsContinuumCondition.setPredicate(new
Predicate<PhysicsContinuum>() {

@Override
public boolean evaluate(PhysicsContinuum continuum) {
if (continuum != null &&
continuum.getReferenceValues()
.has(ReferencePressure.class) != null) {
return true;
}
return false;
}
});
```

Online API:

For coding specifics, see the STAR-CCM+ **Help** menu:

Help > **Java API** > **star.common.filters** > **Predicate** and its subclasses.

See Also:

Conditions, Triggers, and Predicates

CompoundPredicate

`CompoundPredicate` is an extension of Predicate. Use extensions of this class to make evaluations from logical combinations of sub-Predicates. Extensions for `And`, `Or`, and `Not` relations are provided.

Example:

```
CSOCondition<Boundary> cnd = new CSOCondition<Boundary>();
cnd.setDesc("An inlet or outlet boundary.");
```

```
CSOLookupConditionTrigger<Boundary> cndTrigger =
new CSOLookupConditionTrigger<Boundary>(Boundary.class);
cnd.setTriggers(Collections.singleton(cndTrigger));
Predicate<Boundary> inletPredicate = new Predicate<Boundary>() {
@Override
public boolean evaluate(Boundary boundary) {
if (boundary.getBoundaryType() instanceof InletBoundary) {
return true;
}
return false;
}
};
Predicate<Boundary> outletPredicate = new Predicate<Boundary>() {
@Override
public boolean evaluate(Boundary boundary) {
if (boundary.getBoundaryType() instanceof OutletBoundary) {
return true;
}
return false;
}
};
List<Predicate<Boundary>> predicates = new
ArrayList<Predicate<Boundary>>();
predicates.add(inletPredicate);
predicates.add(outletPredicate);
cnd.setPredicate(new OrPredicate<Boundary>(predicates));
```

Online API:

For coding specifics, see the STAR-CCM+ **Help** menu:

Help > **Java API** > **star.common.filters** > **CompoundPredicate**

Help > **Java API** > **star.common.filters** > **AndPredicate**

Help > **Java API** > **star.common.filters** > **NonePredicate**

Help > **Java API** > **star.common.filters** > **OrPredicate**

See Also:

Conditions, Triggers, and Predicates

Using the Simulation Assistant Wizard in NetBeans

The Simulation Assistant Wizard is a NetBeans plugin that automatically generates a Simulation Assistant project in the NetBeans IDE.

This plugin is installed into the NetBeans IDE. It cannot be installed directly within STAR-CCM+. It is not necessary to use the Simulation Assistant Wizard in order to create a Simulation Assistant, however, the

Wizard is provided as a helpful tool that automates some of the process common to creating a new Simulation Assistant.

This section describes how to install and run the Simulation Assistant wizard. The wizard creates a skeleton Simulation Assistant project with a single Task and a single Star Action on that task. You can then customize this skeleton in order to create a useful Simulation Assistant. Running the wizard again will create additional Tasks and Star Actions on a project.

Contents:

Installing The Wizard
Running the Wizard

Installing The Wizard

1. Select **Help** > **Steve Portal** and login.
2. On the Steve Portal, go to the *Downloads* tab, locate **NetBeansWizard.nbm**, and download it.
3. Open NetBeans.
4. From the NetBeans **Tools** menu, select **Plugins**.
5. In the *Plugins* dialog, click the *Downloaded* tab.
6. Click **Add Plugins...** , and locate `NetBeansWizard.nbm`.
7. Select `NetBeansWizard.nbm` and click **Open**.
8. Select **Simulation Assistant Helper Module** and click **Install**.
9. Work through the installation dialog, taking the defaults.

Running the Wizard

1. From the Netbeans **File** menu, select **New Project**.
2. In the *New Project* dialog, select STAR-CCM+ from **Categories** and **Simulation Assistant Project** from **Projects**.
3. Click **Next** to activate the wizard.
4. Enter the information in the fields:
   a) Enter a title and description, then click **Next**.
   b) Enter a project name and, if necessary, browse to a new location, then click **Next**.
   c) Enter a package name in lowercase, for example, `star`.
   d) Enter a class name or accept the default, then click **Next**.
   e) If the STAR-CCM+ library is not on the *Available Libraries* list, create it:
      1. Click **Managing Libraries...**

         The *Ant Library Manager* dialog opens.
      2. Click **New Library...**

         The New Library dialog opens.
      3. In *Library Name*, enter STAR-CCM+, then click **OK**.
      4. Click **OK** to return to the *Include Libraries* dialog.

f) Select STAR-CCM+ from the *Available Libraries* list and click **Managing Libraries...**.

The *Ant Library Manager* dialog opens.

g) Click **New Library...**.

The *New Library* dialog opens.

h) Enter STAR-CCM+ as the library name and click **OK**.

i) Click **Add JAR/Folder...** and enter the JAR files in the following libraries in `[STAR-CCM+_Installation]/star/lib/java/platform`:

- `.../core/*.jar`
- `.../core/locale/*.jar`
- `.../lib/*.jar`
- `.../modules/*.jar`
- `.../modules/ext/*.jar`
- `.../modules/locale/*.jar`

j) When done, click **OK**.

5. Under *Available Libraries*, select the STAR-CCM+ library you created and click **Next**.
6. Click **Finish**.

Wizard Guide

This part of the documentation illustrates how STAR-CCM+ is used for special application tools and projects.

The application tools are:

- The Turbo wizard
- The Fire and Smoke wizard
- The Thermal Comfort wizard
- The SPEED Motor Tool Import wizard

The other parts of the documentation cover the following:

- An introduction to using STAR-CCM+
- Procedures for using 3D-CAD to build geometry from scratch within the STAR-CCM+ environment
- How to generate or import meshes and work with them
- Modeling physics—the concepts, theory, and implementation details of the models
- Using tools in STAR-CCM+ to facilitate your work
- Detailed information on interacting with CAD/CAE products in a STAR-CCM+ simulation
- Examples of applying the models of STAR-CCM+ in sophisticated cases
- Preparation and procedures for running a STAR-CCM+ simulation
- Analyzing results through visualization and plotting
- A tutorial guide that shows you how to use STAR-CCM+ for various applications in a step-by-step format
- A guide to using STAR-View+, a viewer for three-dimensional STAR-CCM+ scenes

Contents:

Using the Turbo Wizard
Using the Fire and Smoke Wizard
Using the Thermal Comfort Wizard
Introduction to the SPEED Motor Tool Import Wizard
Using The IDF Import Wizard

Using the Turbo Wizard

The turbo wizard is used in the setup of turbomachinery simulations.

The turbo wizard provided by STAR-CCM+ is used in the setup of turbomachinery simulations to:

- generate a high-quality mesh for the fluid domain surrounding a blade, or blades
- set boundary and initial conditions for the fluid domain (optional)

This section includes information on the capabilities of the turbo wizard and describes how to use the wizard to set up a turbomachinery case.

Contents:

What is the Turbo Wizard?
Turbomachinery Related Capabilities of STAR-CCM+
Setting Up the Turbo Wizard
Saving and Loading Turbo Wizard Data

What is the Turbo Wizard?

The turbo wizard is an automatic meshing tool for creating high-quality turbomachinery meshes quickly.

Because these meshes are multi-block and comprised of hexahedral cells, they are highly accurate and efficient to solve. Its capabilities include:

- Elliptic and Algebraic grid generation system
- Moving mesh setup
- Direct blade definition from section data
- Boundary surface monitoring (forces, pressures...)

Because the mesh near the blade is so critical to the accuracy of the analysis, the resolution requirements on the mesh are demanding. In order to avoid creating meshes with unnecessarily high cell counts, the turbo wizard creates a mesh using one of the elliptic (”O”) grid types.

While most problems can be meshed in several ways, the best choice of mesh topology depends upon your particular geometry and analysis requirements.

The process for analyzing turbomachinery problems is divided into three steps:

1. Meshing the geometry
2. Solving the flow problem
3. Post-processing the flow solution

The geometry and demands put upon the resolution of the model near a typical blade/impeller setup make step 1 challenging. The turbo wizard simplifies the creation of a high-quality mesh and application of appropriate boundary conditions to it.

A typical analysis starts with either a blade design or a CAD model of the blade. This can be imported into the turbo wizard by creating an input file that defines the blade surface as a set of coordinates. By default, the flow is assumed to move in the +z direction.

The fluid mesh is then created around it, boundary conditions are applied to the fluid domain, and the model is solved in STAR-CCM+.

See Also:

Working with Turbomachinery Input Files

Turbomachinery Related Capabilities of STAR-CCM+

The turbo wizard is the primary tool for turbomachinery problems, but there are several other STAR-CCM+ features that can help with the CFD solution.

The summary of the capabilities of these tools includes :

- Modeling Options:
  - Multiple (rotating) Reference Frames
  - Chemically Reacting Flows
  - Free Surface Flows
  - Conjugate Heat Transfer
- Turbulence Models:
  - Spalart-Allmaras
  - K-Epsilon, K-Omega...
- Boundary Conditions:
  - Inlet
  - Stagnation
  - Radial Equilibrium
  - Non-reflecting (transient)
  - Pressure
- Solver:
  - Second-Order
  - Segregated
  - Algebraic Multigrid Acceleration
- Automatic Meshing Tools:
  - Tetrahedral, hexahedral, and polyhedral meshing with surface extrusion layers
  - Elliptic mesh generator for hexahedral meshes near the blade with the turbo wizard

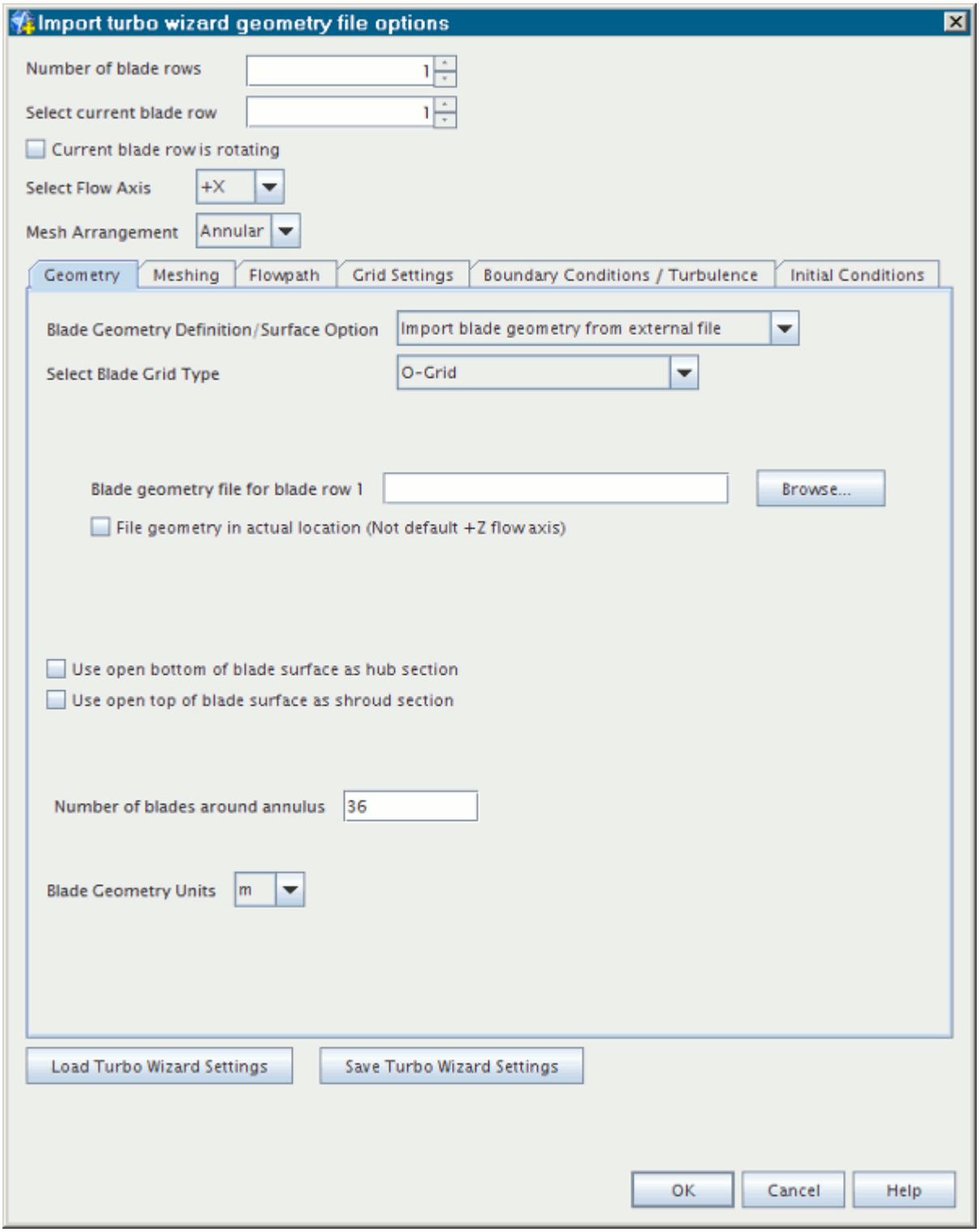
A typical example can involve generating a single-sector mesh of the near-blade region using the turbo wizard and then including this portion in a more complicated model of a piping, ducting, or another fluid handling system. This approach leverages the high-quality hexahedral meshing capabilities of the turbo wizard with the general, automatic meshing capabilities of STAR-CCM+. The result is an easy-to-create mesh that gives you control over fine details, while minimizing effort and solution time.

Setting Up the Turbo Wizard

The turbo wizard is used to generate a high-quality mesh for turbomachinery blades, and can also be used to set up boundary and initial conditions.

These mesh and physics properties are specified using the *Import turbo wizard geometry file options* dialog, which appears when the turbo wizard is activated.

Clicking **OK** in this dialog always triggers mesh generation. If the information entered in the dialog is invalid or incomplete, this may result in an error. Clicking **Cancel** exits the dialog but does not clear the settings and data that have already been entered. This is discussed in greater detail below.



There are two broad categories of data that are entered in the dialog:

- Meshing data—specifies the properties of the mesh and is defined in the fields outside of the tabbed panel, then in the first four tabs:
  - Number of blade rows
  - Rotation of the current row

- Flow axis
- Mesh arrangement
- Geometry tab
- Meshing tab
- Flowpath tab
- Grid settings tab

This data must be entered to generate a mesh.

- Physics data—sets the boundary and initial conditions and is defined remaining two tabs:
  - Boundary conditions / turbulence tab
  - Initial conditions tab

  You do not have to enter values in these tabs, but the turbo wizard uses the default values to set up the models and conditions in the simulation if no data is provided.

The turbo wizard is an options dialog, so all settings and values entered in the dialog are retained when it is closed and reopened. When the simulation is closed, the turbo wizard dialog is cleared, and the only way of retaining the information is to save the turbo wizard data separately. Turbo wizard data is related to the active simulation, and cannot be applied across several simulations unless the settings are saved in one simulation and loaded into the next.

Contents:

Activating the Turbo Wizard
Defining the Number of Blade Rows
Defining Rotation
Defining the Flow Axis
Selecting Mesh Arrangement
Setting Up the Blade Geometry
Selecting the Mesh Generation Method
Defining the Flow Path
Controlling the Grid Settings
Setting Up the Boundary Conditions
Setting Up the Initial Conditions
Working with Turbomachinery Input Files

See Also:

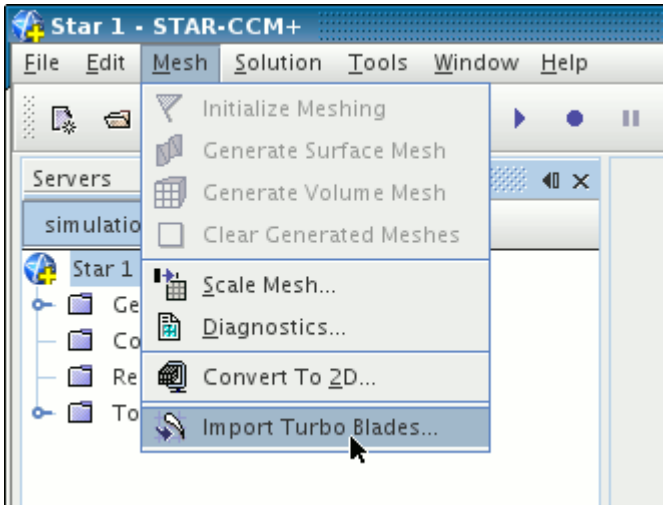
Saving and Loading Turbo Wizard Data

Activating the Turbo Wizard

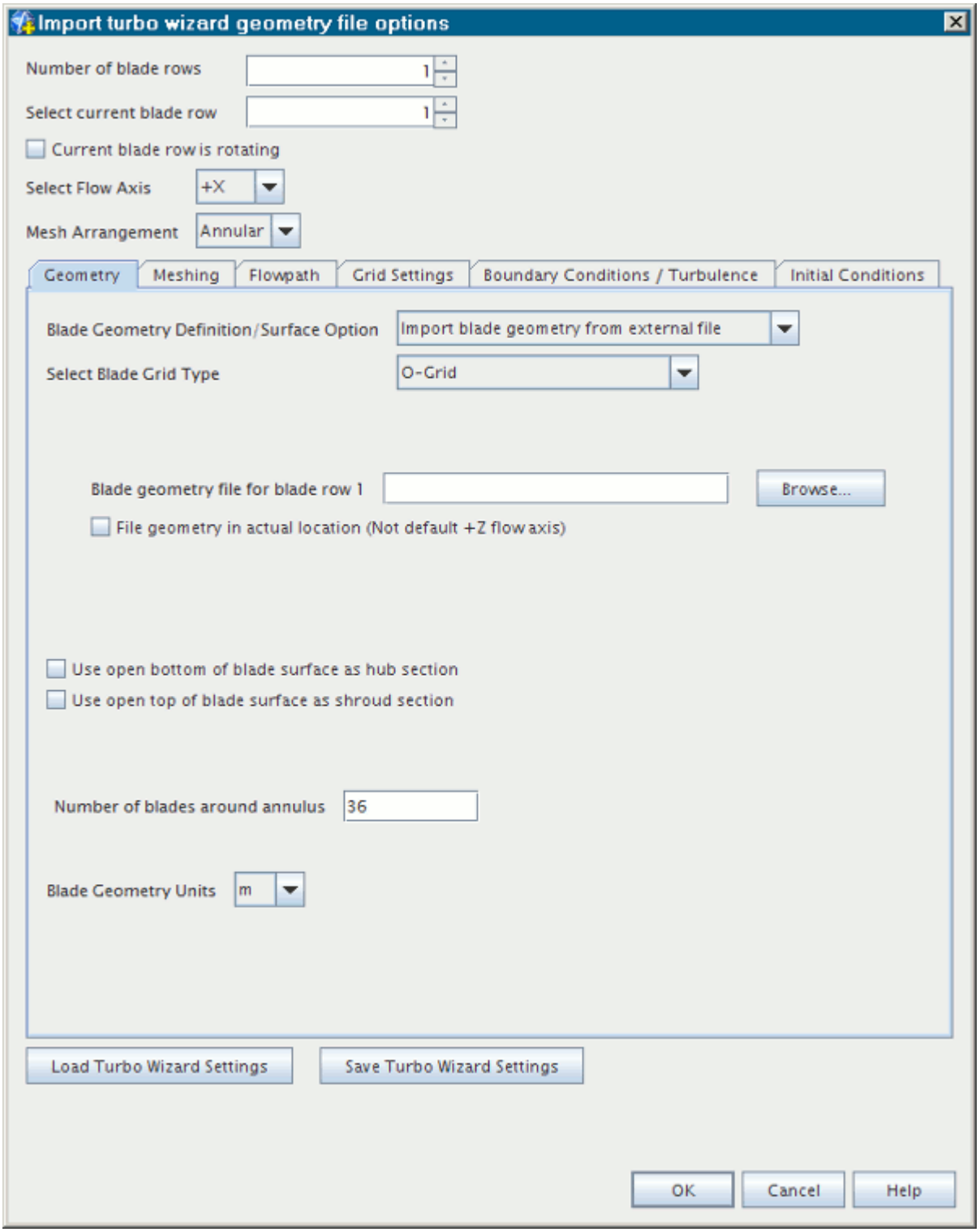
This section describes how to activate the Turbo Wizard.

To activate the Turbo Wizard:

1. In an open simulation, select **Mesh** > **Import Turbo Blades...** in the menu bar.



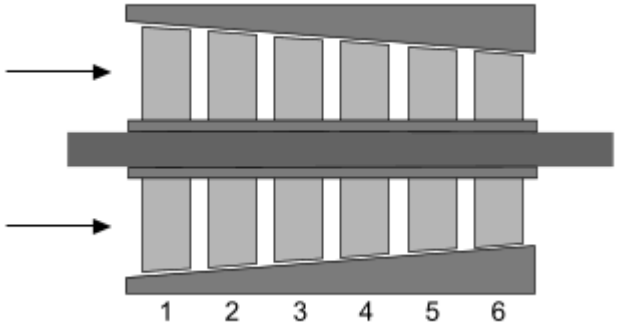
The *Import turbo wizard geometry file options* dialog opens.



Defining the Number of Blade Rows

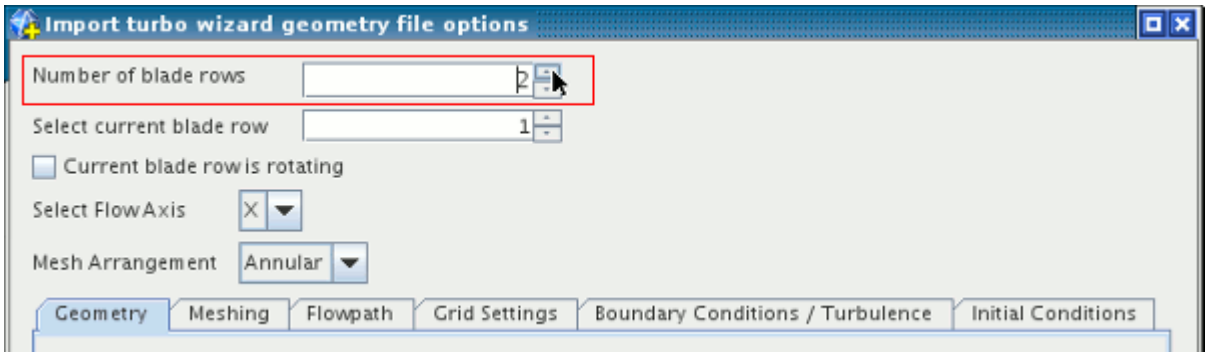
The turbo wizard can generate a mesh for multiple blade rows, and the first step in using the turbo wizard is to specify the number of blade rows in the model.

For cases involving multiple blade rows, the turbo wizard assumes the convention indicated in the diagram below, where blade row 1 is the inlet, and subsequent blade rows are numbered incrementally.



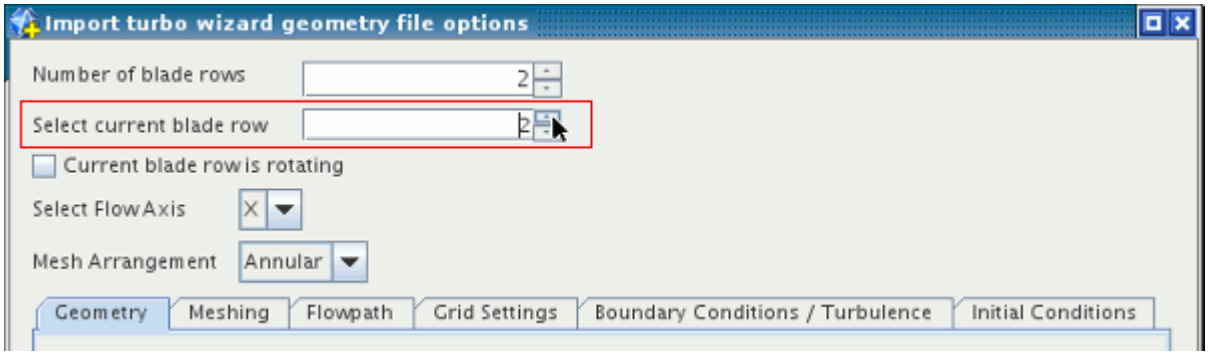
1. The number of blade rows is specified by increasing or decreasing the integer value for the *Number of blade rows* property at the top of the turbo wizard dialog.

   The default value is 1.



2. Mesh data is entered separately for each blade row, so the next step is to indicate for which blade row the first set of data is entered.
3. The blade row is selected using the *Select current blade row* property.

   For example, setting the value to 2 allows you to enter data for blade row 2.



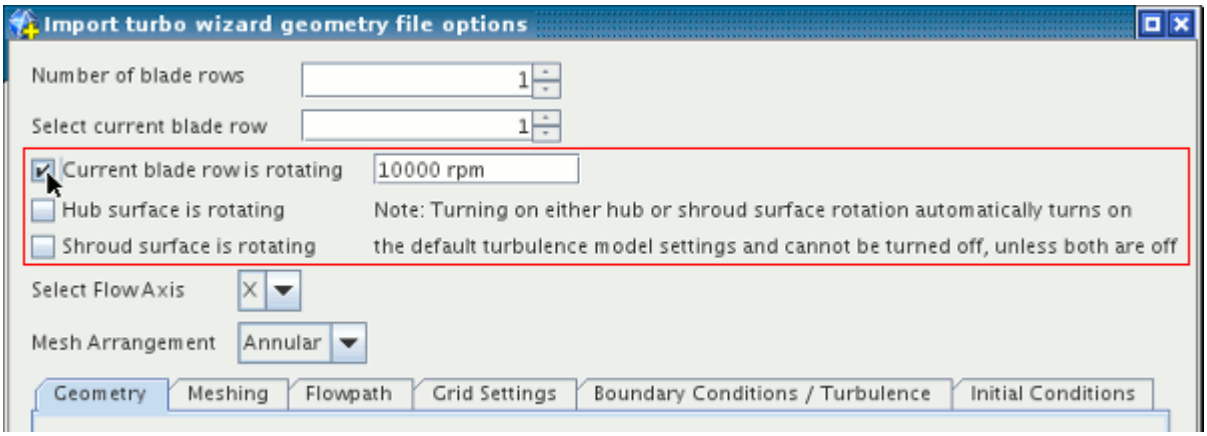
4. To switch to a different blade row, change the value of the *Select current blade row* property.

Data entered in the *Geometry*, *Meshing*, *Flowpath*, and *Grid Settings* tabs, as well as the rotation option, is specific to each blade row.

Defining Rotation

Rotation is defined per blade row.

Rotation is defined per blade row and is enabled by activating the **Current blade row is rotating** checkbox.



When the **Current blade row is rotating** checkbox has been activated, the following fields are made available:

- Rotational speed of the blade row, in `rpm`.
- **Hub surface is rotating**—Activate this checkbox to indicate that the hub is rotating with the blade.
- **Shroud surface is rotating**—Activate this checkbox to indicate that the shroud is rotating with the blade.

These values help the turbo wizard define the sliding interfaces correctly in the simulation.

Default turbulence models are activated automatically when a blade row is specified as rotating. The default is a stationary blade row.

See Also:

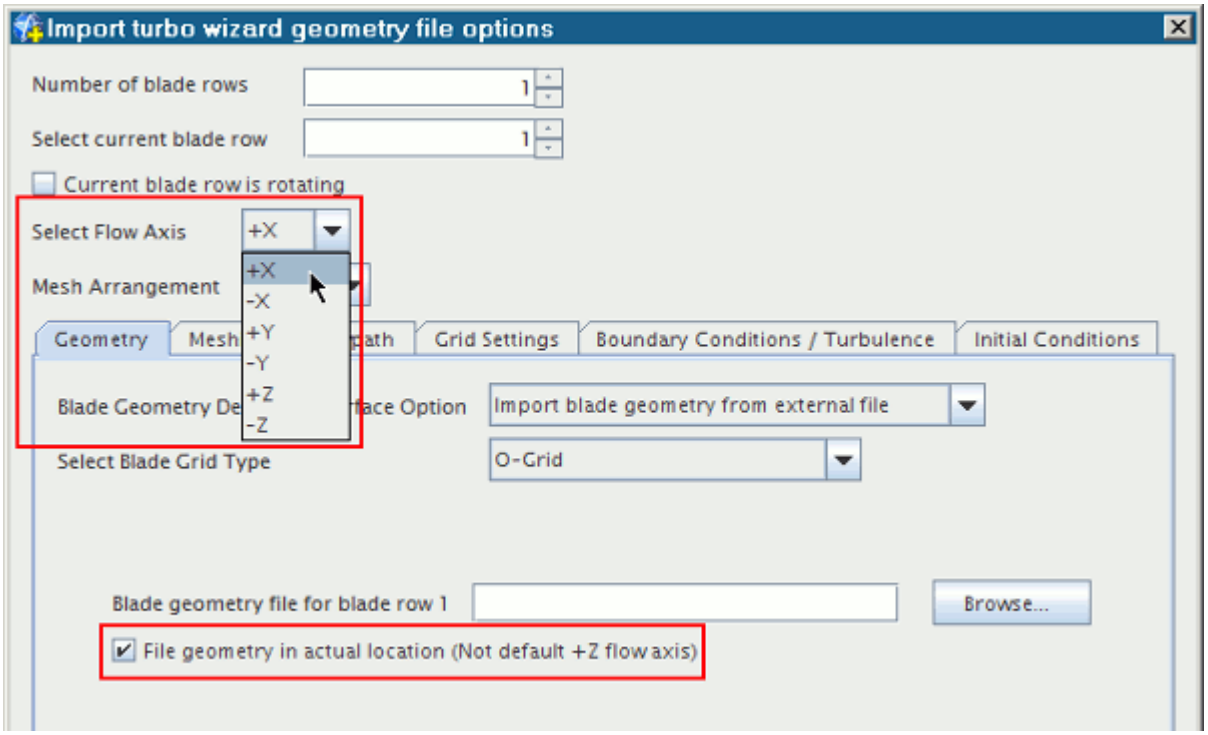
Defining the Number of Blade Rows

Setting Up the Boundary Conditions

Defining the Flow Axis

The flow axis defines the direction of flow over the blade.

For impeller blades, this direction is at the inlet.



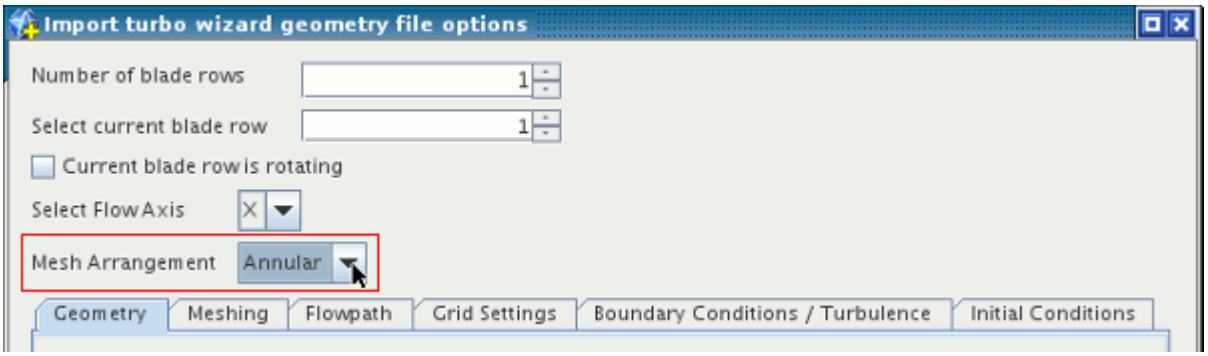
The **Select Flow Axis** menu is used to specify the **X**, **Y**, or **Z** axis as the flow axis. Use this menu only if you want to define a flow axis other than the default position (flow direction +z).

To apply your selection in this menu, make sure that you activate the checkbox **File geometry in actual location**. By default this option is deactivated.

Selecting Mesh Arrangement

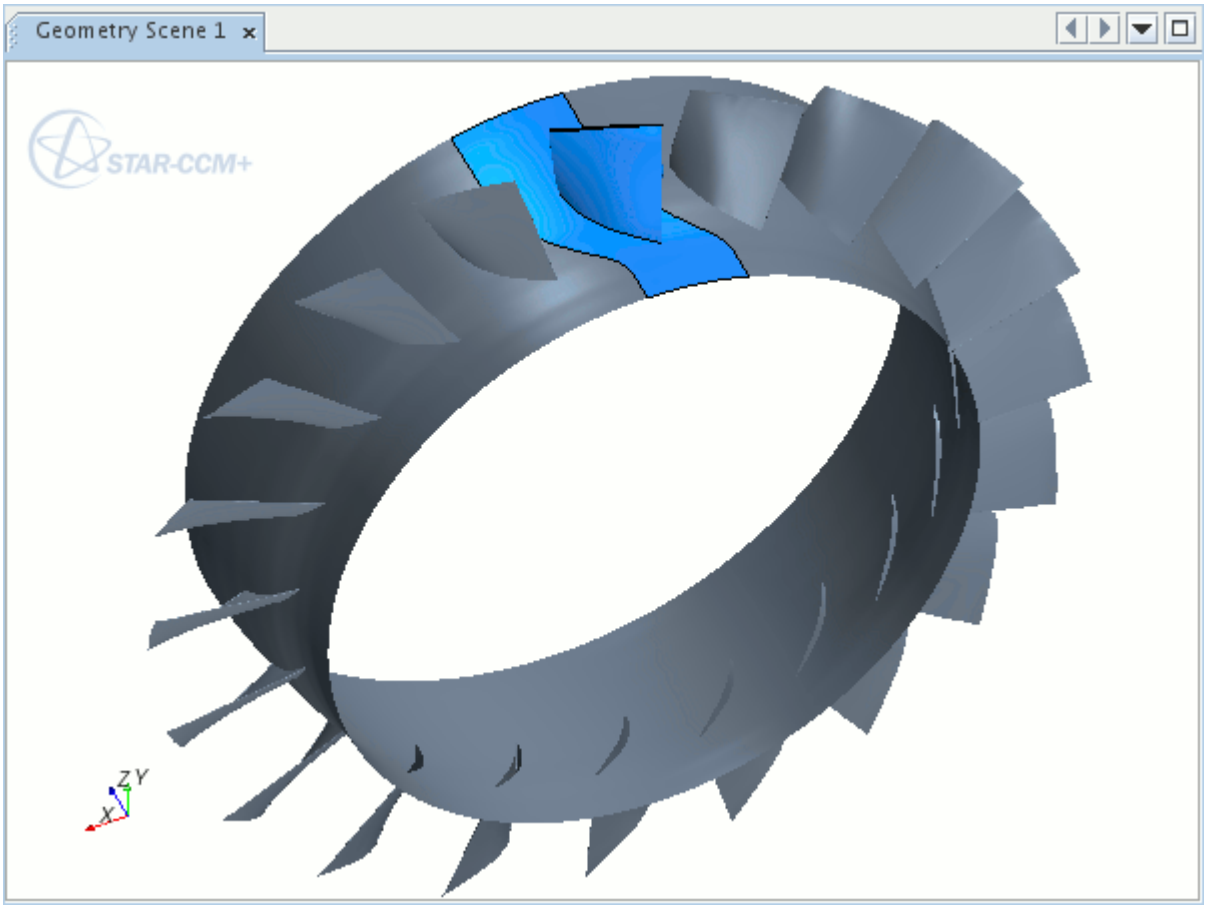
The **Mesh Arrangement** option indicates whether the blades are arranged around an annulus or in a linear pattern.

This setting influences the mesh generated by the turbo wizard and changes the options available when setting up the blade geometry.



Annular

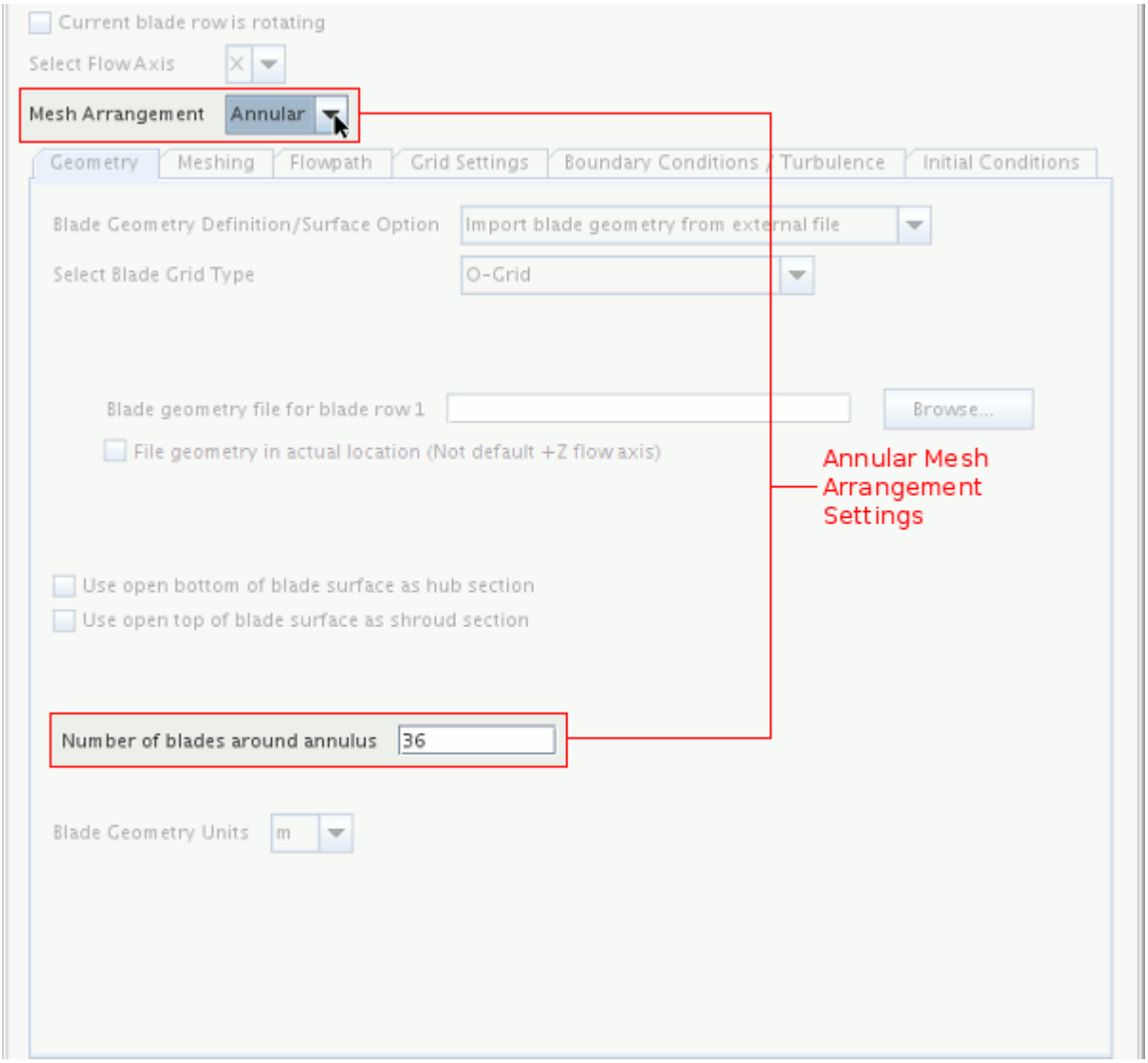
In an annular mesh arrangement, the blade geometry is rotated around an axis. This is the default option and is suitable for a typical turbo wizard analysis. In the following screenshot, a transform has been used to show the full rotation of the blade geometry (blue).



This option can be specified by selecting **Annular** from the **Mesh Arrangement** menu. This is the default option and has an additional setting:

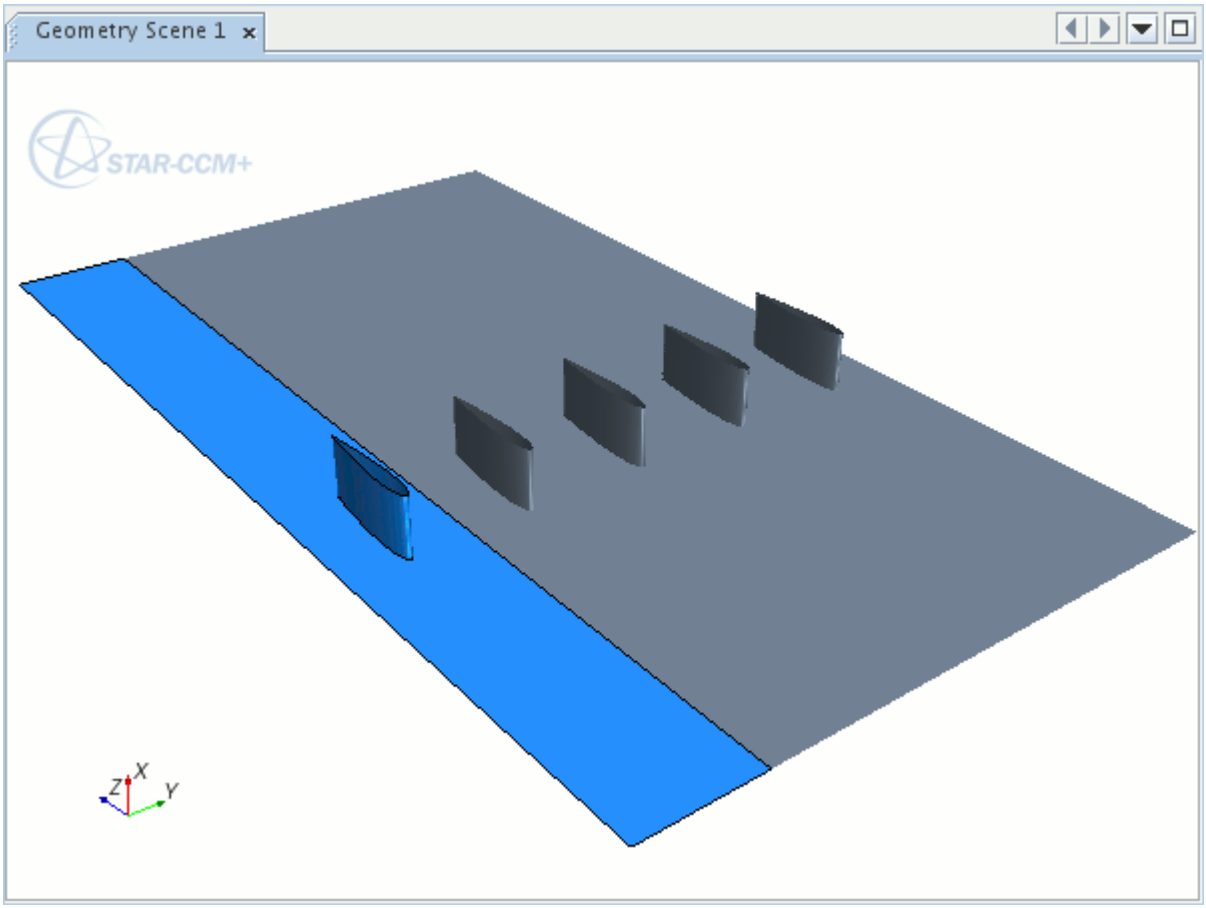
- **Number of blades around annulus**

  This option indicates the total number of blades around the annulus and different values can be applied to each row. For an impeller-splitter case, this number refers to the total number of impeller-splitter pairs around the annulus.



Linear

In a linear mesh arrangement, blades are laid out side-by-side in a linear pattern. The example below shows a single blade in the linear mesh arrangement; a transform has been used to shown several repetitions of the blade geometry (blue).



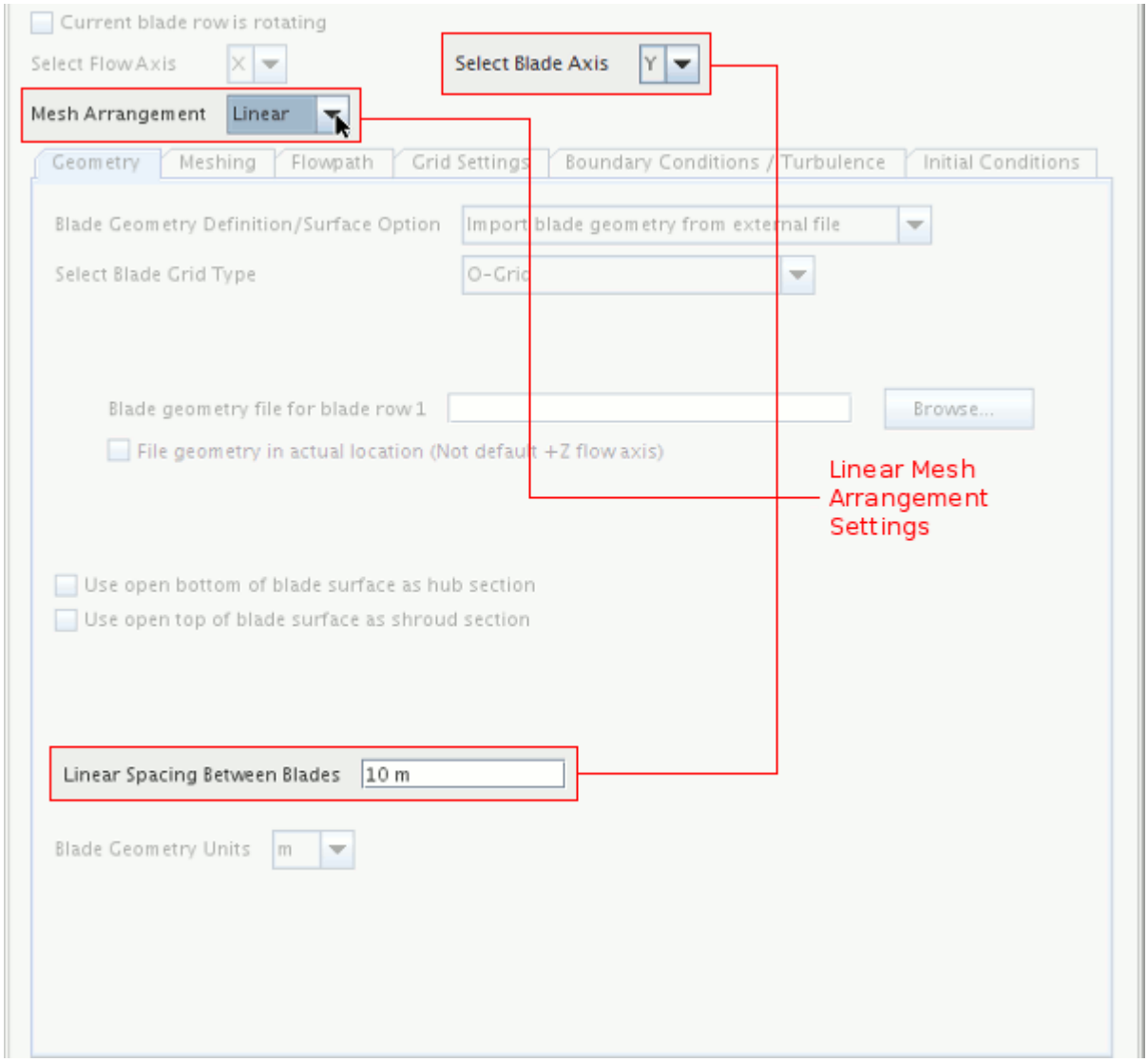
This option is specified by selecting **Linear** from the **Mesh Arrangement** menu, which activates two more settings:

- **Select Blade Axis**

  This is the direction in which the blade pattern is repeated. In the example above, the y axis is the blade axis.

- **Linear Spacing Between Blades**

  This distance is measured from a point on one blade to the same point on the next blade (either in the repeating pattern or between blade rows). It can also be thought of as the width of the flow domain extending laterally from the first blade. The same spacing distance is applied to all blade rows.

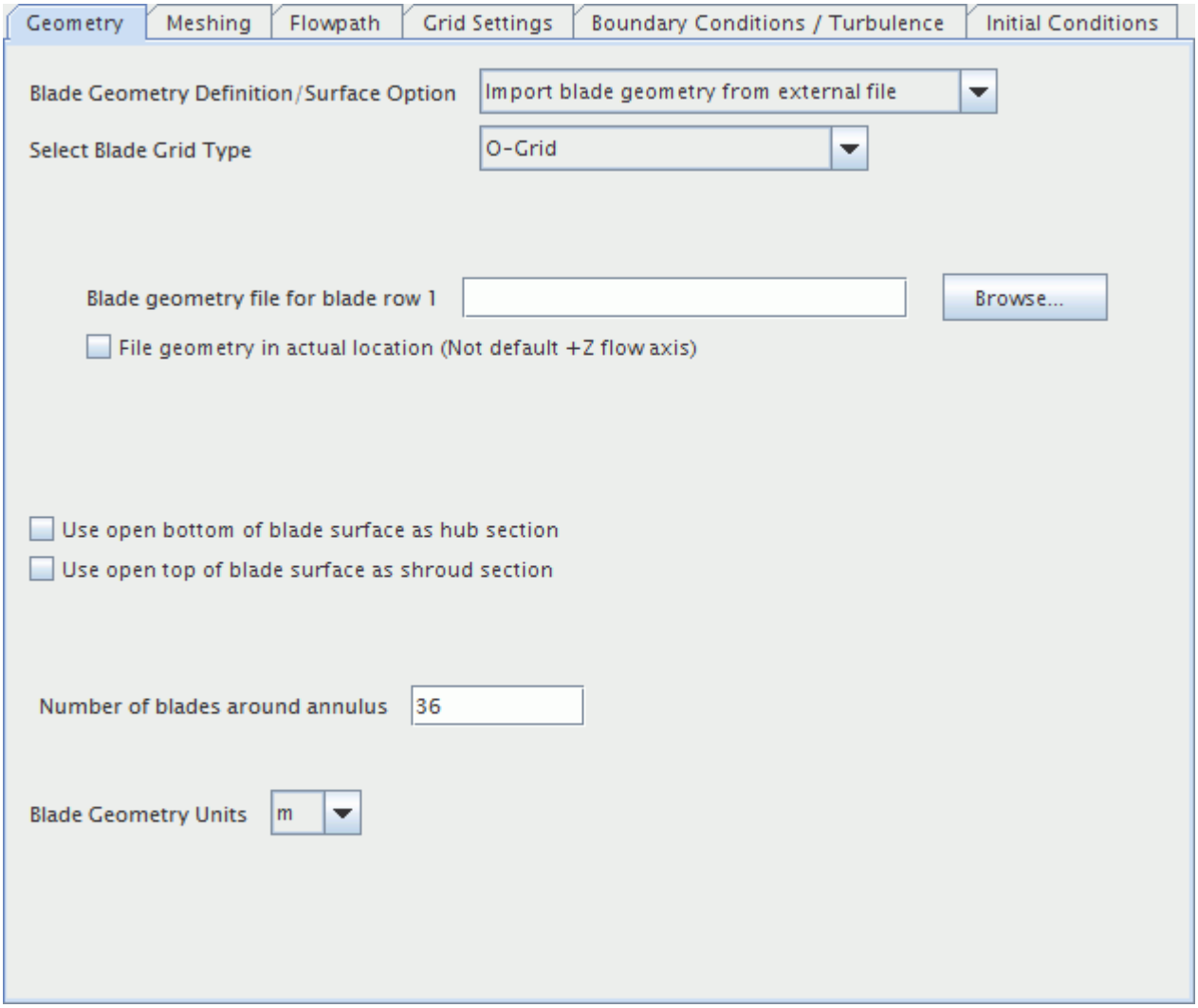


See Also:

Setting Up the Blade Geometry

Setting Up the Blade Geometry

The *Geometry* tab is used to specify the source, type, and characteristics of the blade geometry.



- The Blade Geometry Definition/Surface Option determines the source of the blade geometry, and offers the choice of importing the blade definition from an external file, or using an existing surface representation.
- The Select Blade Grid Type option identifies the blade type, and influences the shape of the grid generated around the blade.

The remaining options indicate the characteristics of the blade geometry:

- **Use open bottom of blade surface as hub section**

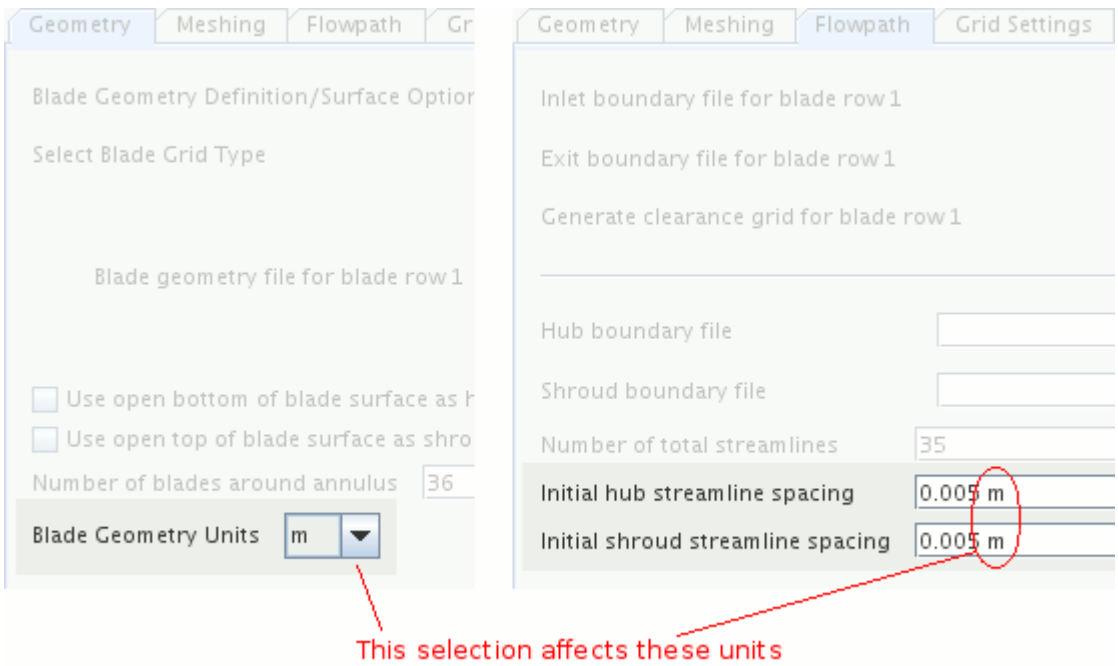
  If the blade definition does not intersect the hub, or if it just touches the hub, the turbo wizard is not able to calculate a section at that point and mesh generation fails. You should check this box unless the blade definition extends past the hub.
- **Use open top of blade surface as shroud section**

  If the blade definition does not intersect the shroud, or if it just touches the shroud, the turbo wizard is not able to calculate a section at that point and mesh generation fails. You should check this box unless the blade definition extends past the shroud.
- **Number of blades around annulus** or **Linear spacing between blades**

This option depends on whether the mesh arrangement has been specified as annular or linear.

- **Blade Geometry Units**

  This option identifies the units in which the blade geometry was defined. The units must be the same for each import file, and must match those used in the simulation if the blade geometry is defined by an existing surface representation. The choice of units affects the units for the streamline spacing options in the *Flowpath* tab.



Contents:

Defining the Blade Geometry Source
Selecting the Blade Type

See Also:

Selecting Mesh Arrangement

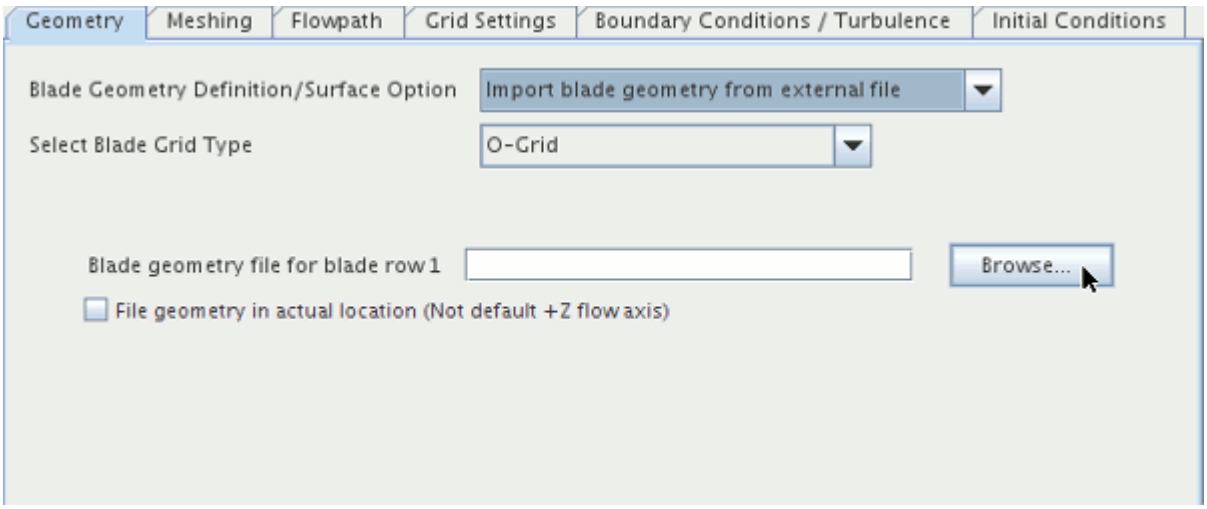
Defining the Blade Geometry Source

The source of the blade geometry for each blade row can be either an external file that you create, or an existing surface representation in STAR-CCM+.

Each option is described below.

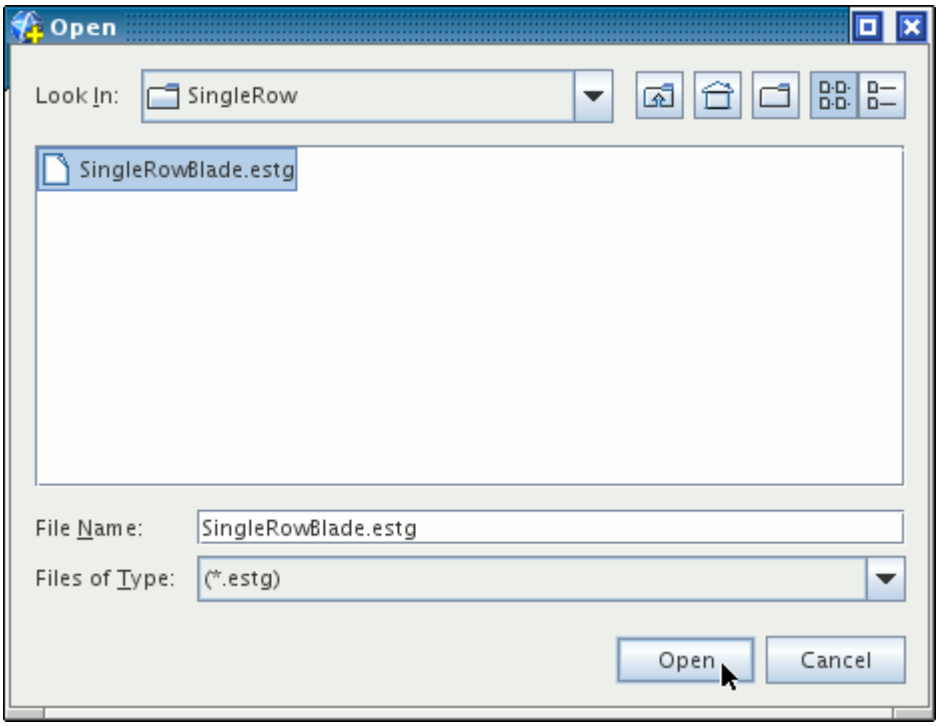
Importing Geometry from an External File

Blade geometry can be imported from an external .estg file by selecting **Import blade geometry from external file** for the **Blade Geometry Definition/Surface Option**.



When you define a flow axis other than the default position (flow direction +z), make sure that you activate the checkbox **File geometry in actual location**. By default this option is deactivated.

To select the `.estg` file, click the **Browse...** button next to the **Blade geometry file for blade row x** option. This action opens an operating system-specific standard dialog that can be used to locate and open the file.

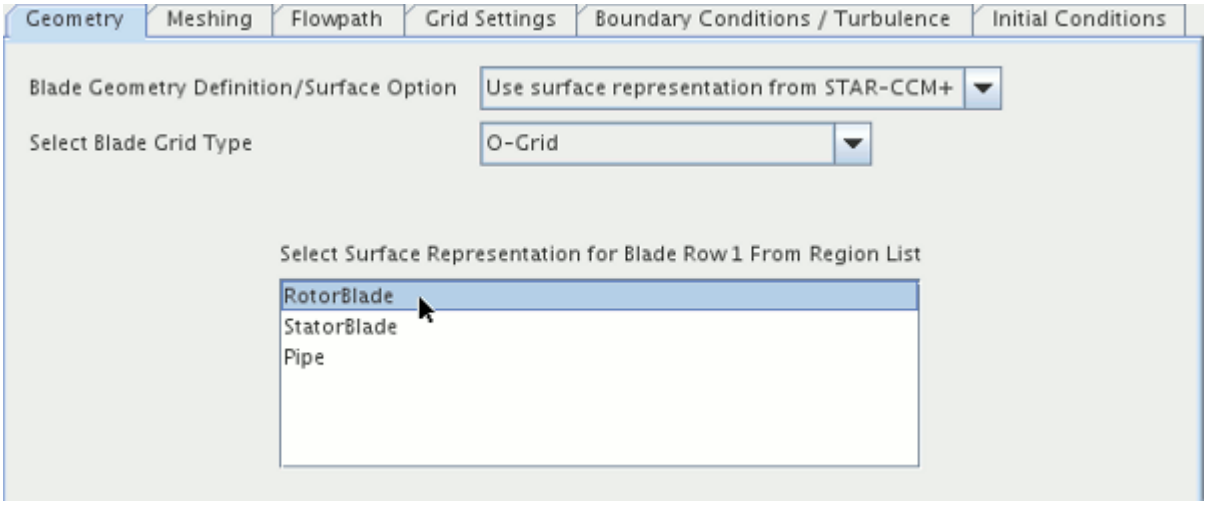


Using an Existing Surface Representation

Blade geometry defined in an existing surface representation can be used by selecting **Use surface representation from STAR-CCM+** for the **Blade Geometry Definition/Surface Option**. A suitable surface representation can be obtained by importing the blade surface data into STAR-CCM+. Each blade must be

defined in a separate region, so once the surface data has been imported some region manipulation may be required.

The region in which the blade is defined is selected from those listed in the Select Surface Representation for Blade Row x From Region List box.



Consider the following when choosing this approach:

- Selection is on a region-wise basis, rather than a boundary-wise one. So all surfaces in the region are used as the airfoil geometry.
- Only the airfoil surface itself should be in the region. If the hub/shroud section areas are closed surfaces, the surfaces should be removed in order to allow the turbo wizard to determine the sections properly at these locations. That is, there should be an open hole at the bottom (hub) and top (shroud) locations of the airfoil surface.
- If you plan on meshing a radial blade or capped Open-TE C-grid blade, these grid types require that two (2) points at the trailing edge location of the airfoil are segments (that is, corner points). This can be done when using surface representations by creating feature curves (edges) which lie along the corner edges in these blade types.
- Feature curves should only be on these segmented corner locations. If they are anywhere else on the model, the turbo wizard understands that those locations should also be segmented.

See Also:

Defining the Number of Blade Rows

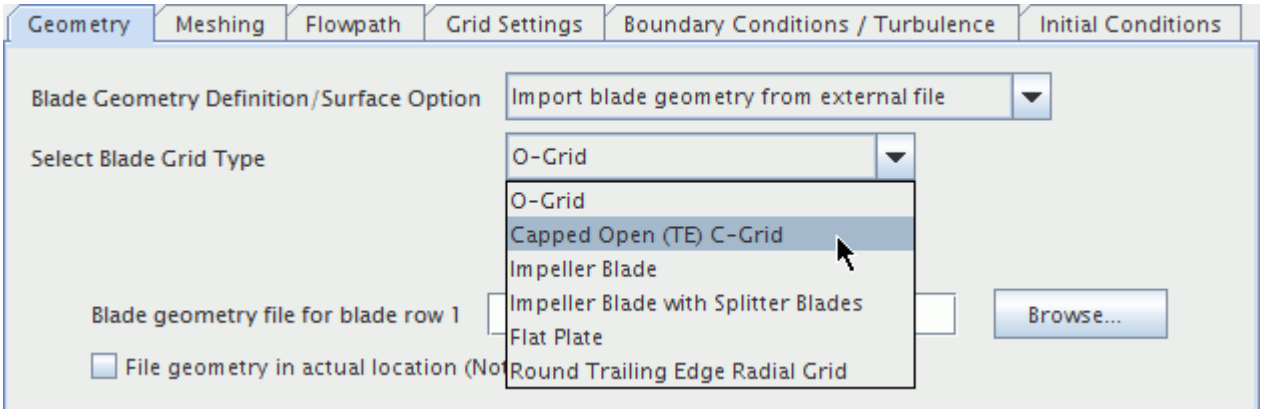
Understanding the .estg File Format

Defining the Flow Axis

Selecting the Blade Type

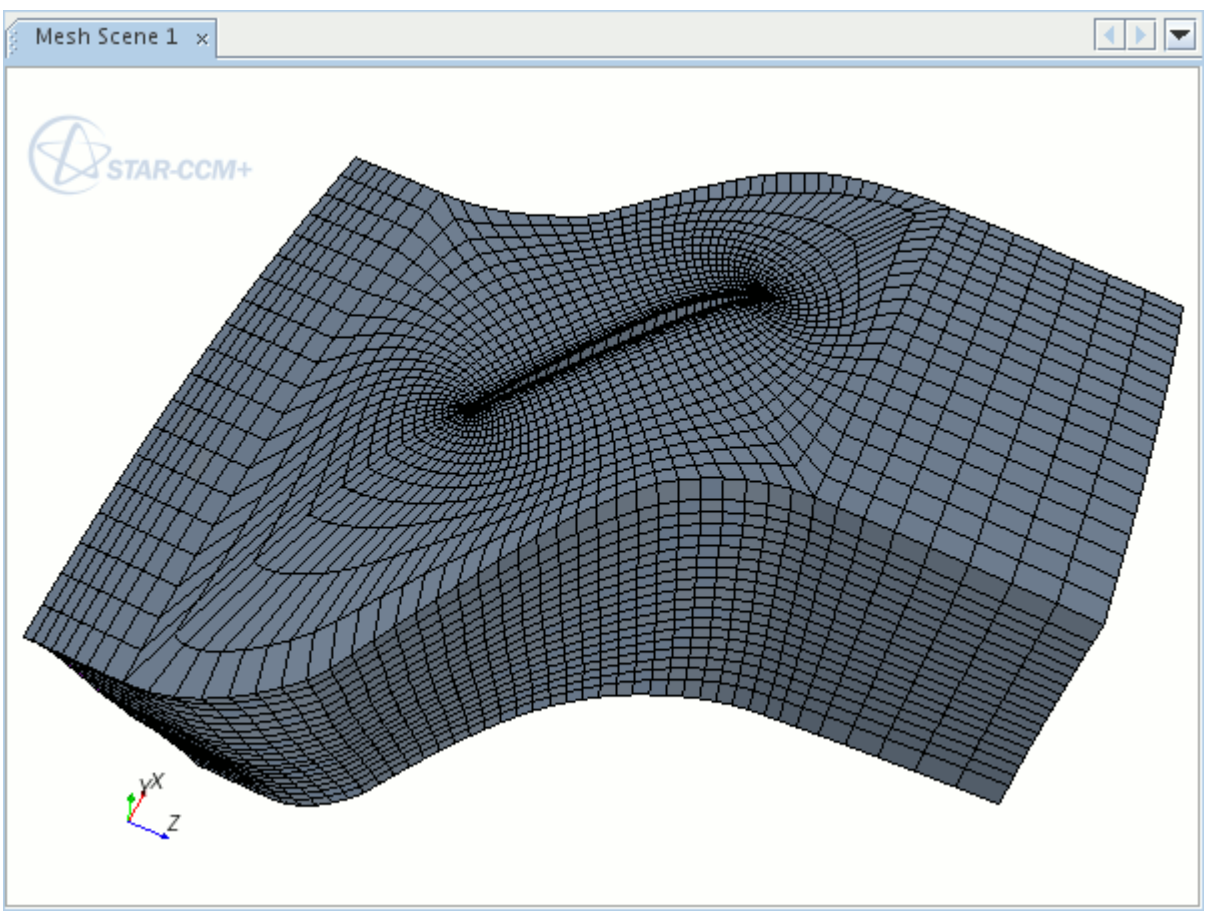
The **Select Blade Grid Type** option indicates the type of blade defined in the source file or representation, and specifies the corresponding grid that is generated by the turbo wizard.

For each blade grid type, the resulting mesh is characterized by the pattern of grid lines around the blade. The options available are described below.



O-Grid

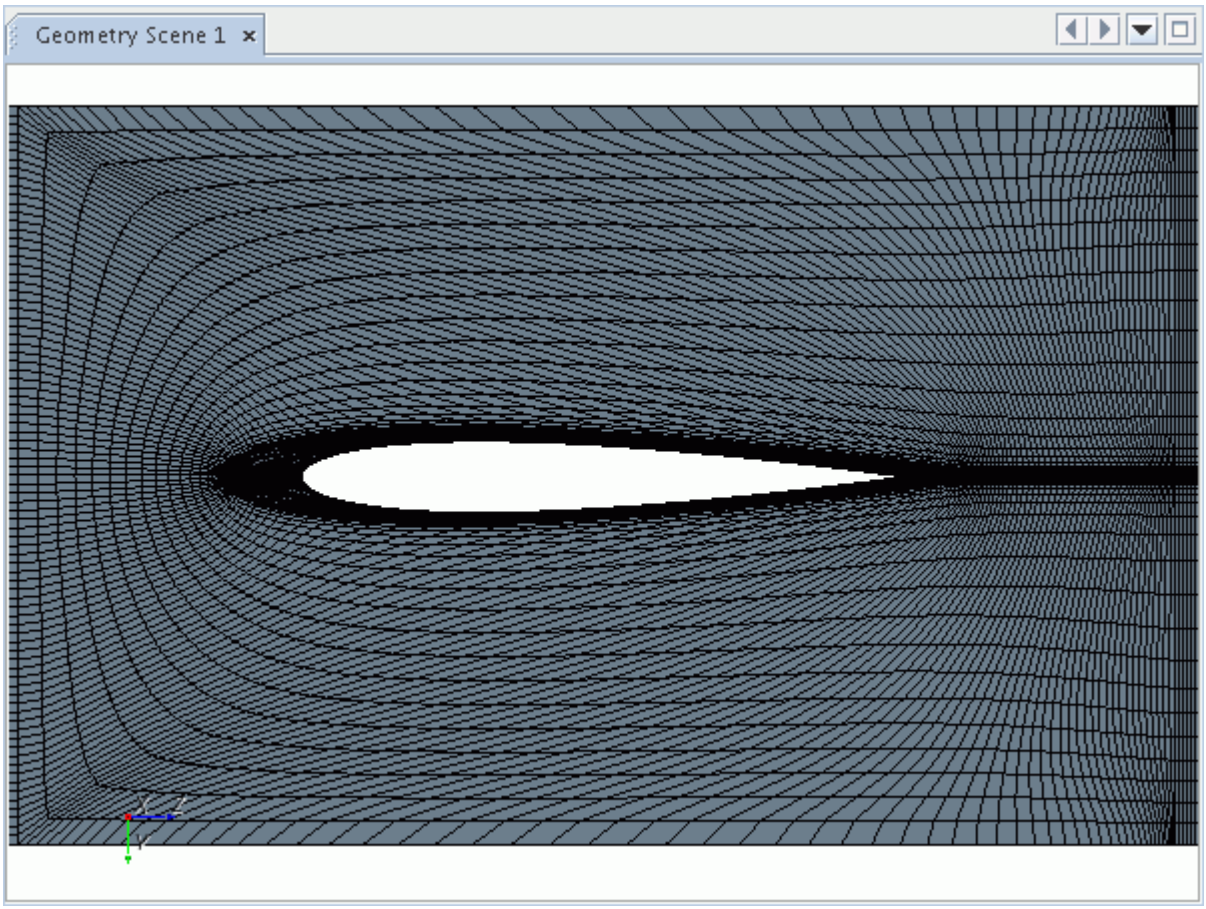
The O-Grid is suited to meshing axial blades that have tapered leading and trailing edges. It generates closed loops of grid lines around the blade, “O”, and an orthogonal grid upstream and downstream of the blade, known as an “H”-grid.



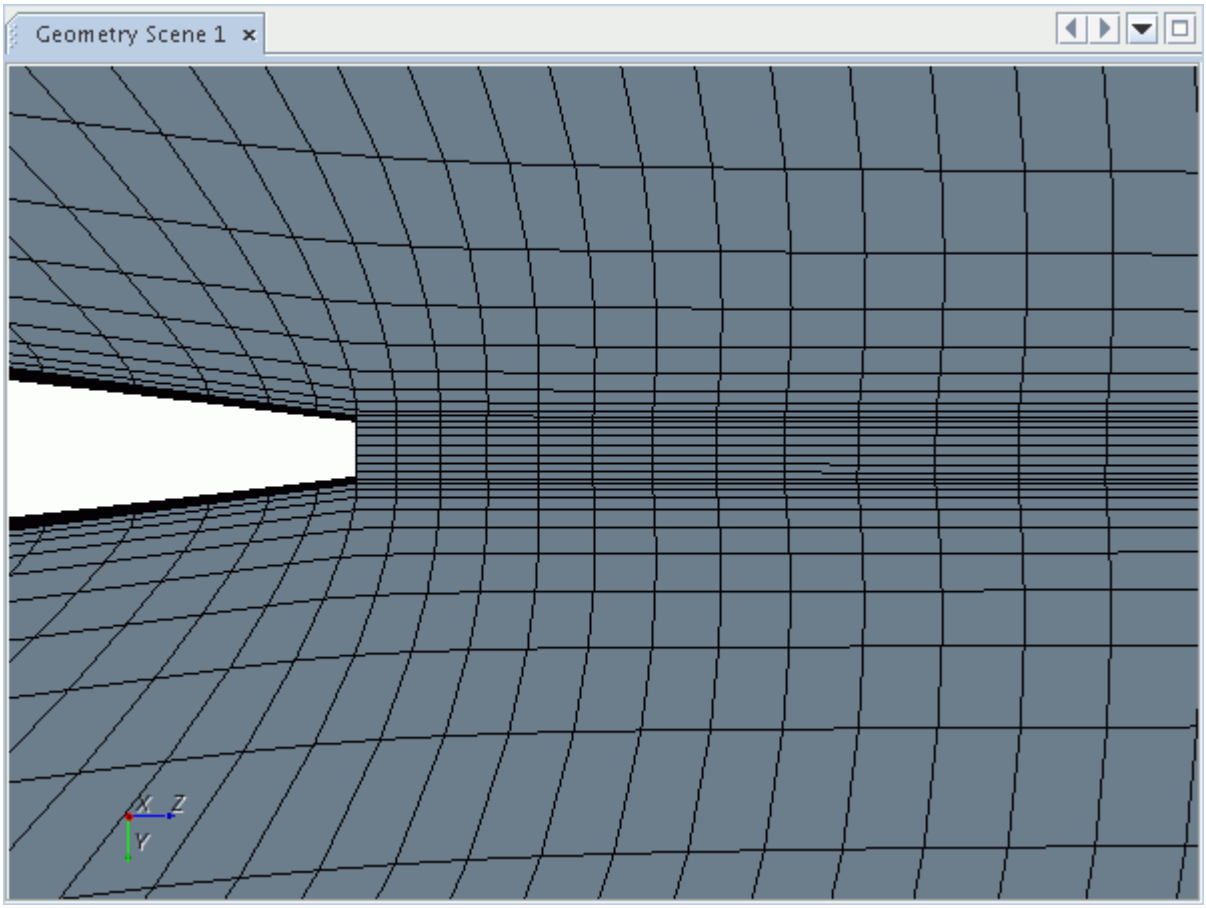
Capped Open (TE) C-Grid

The Capped Open (TE) C-Grid is useful for meshing blades that have a blunt trailing edge. For example:

It generates a C-Grid around the leading edge of the blade, and a fine orthogonal grid projecting from the trailing edge of the blade.



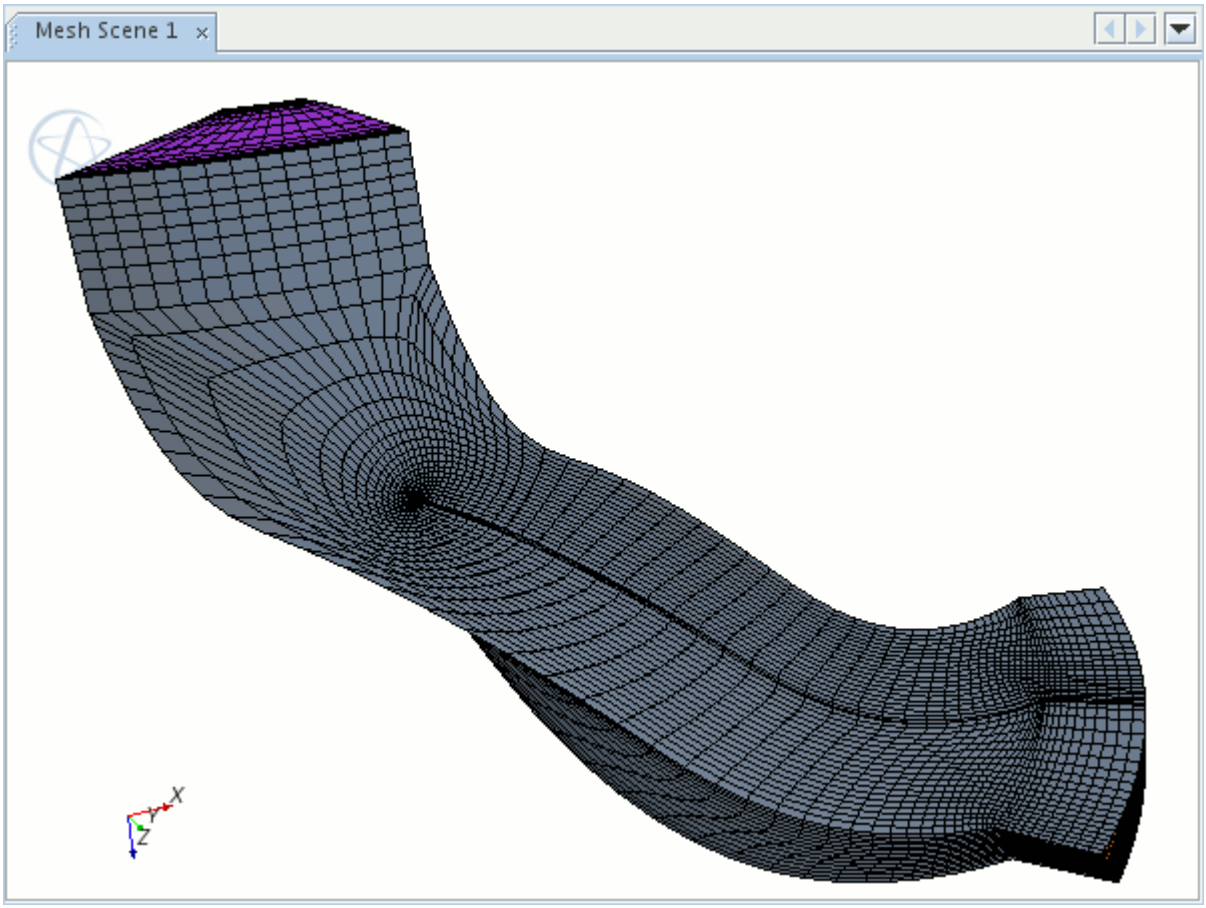
Provided below is a close-up of the orthogonal grid from the trailing edge of the blade.



To successfully create a Capped Open (TE) C-Grid blade, ensure you use feature curves to mark the corners of the blunt trailing edge.

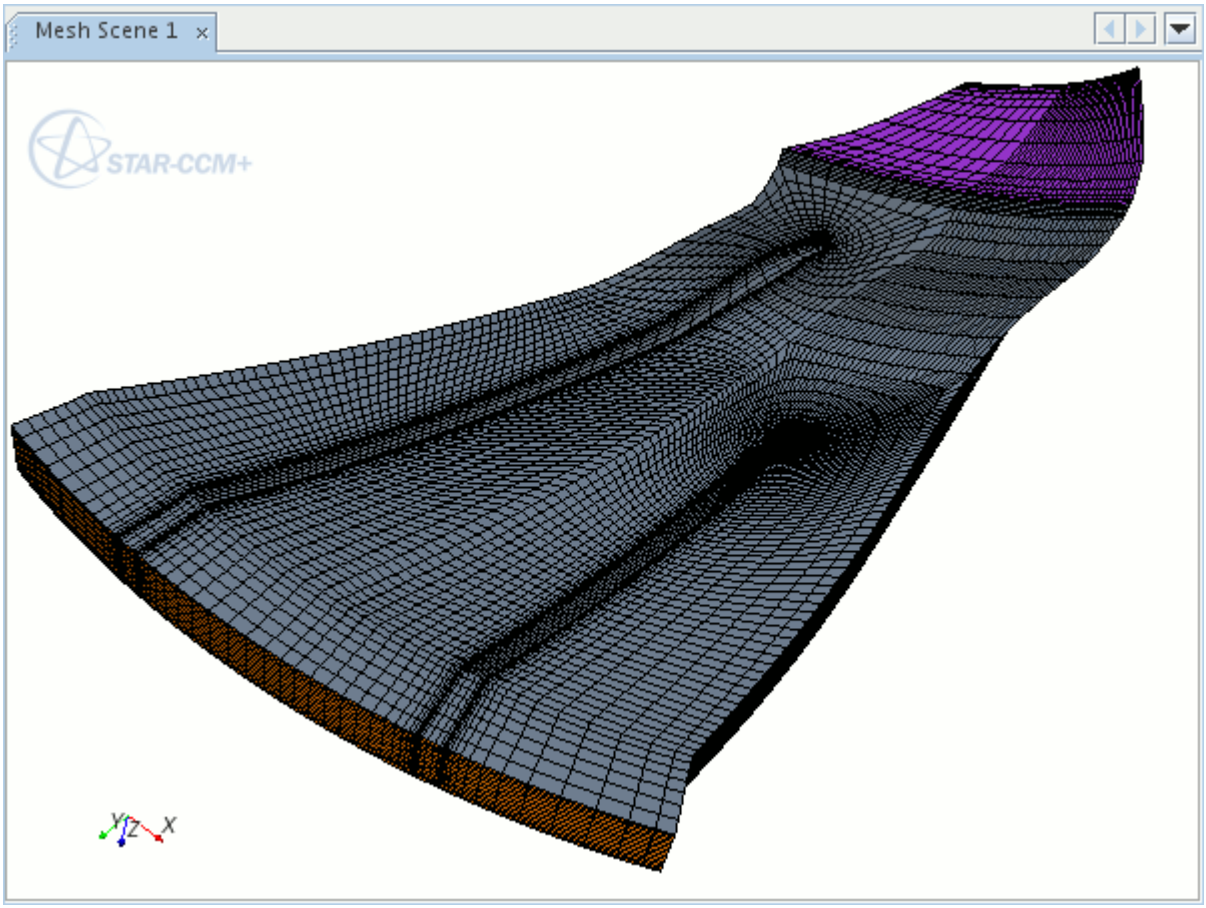
Impeller Blade

This option should be selected if the blade is an impeller. The impeller grid takes into account the blunt trailing edge, and the mesh generated around the blade has characteristic C-shape grid lines. An H-grid is generated upstream and downstream of the blade.



Impeller Blade with Splitter Blades

This option allows a grid to be generated for an impeller blade and a splitter blade. Blade geometry for the impeller and splitter must be specified in separate `.estg` files or regions, depending on the selected blade geometry source. The **Select Blade Geometry Designation** option is used to switch between the impeller blade and the splitter blade, allowing the geometry for each blade to be selected separately.

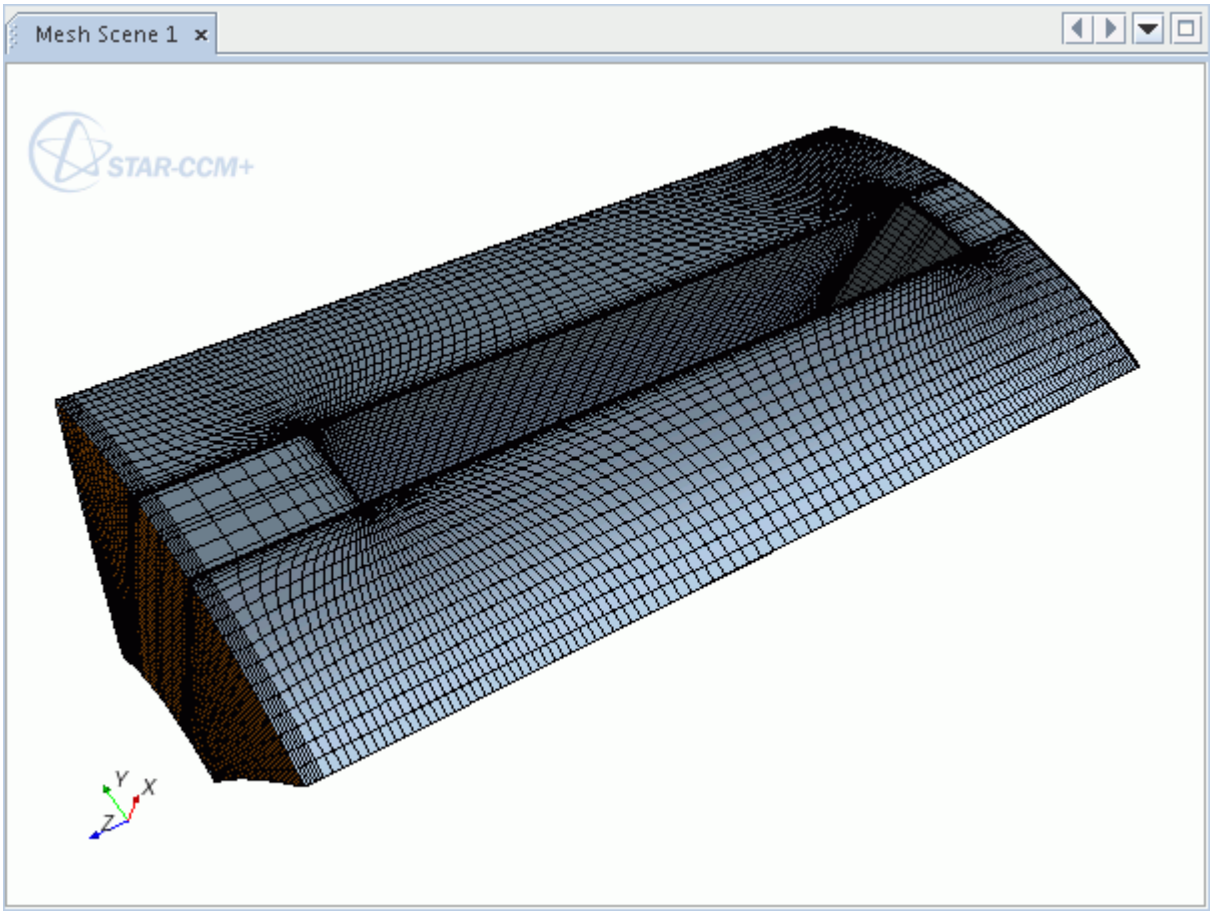


It is only possible to model a single splitter blade, so the **Number Of Splitter Blades** value is fixed at `1`.

Flat Plate

This option should be selected if the blade is a flat plate (the leading and trailing edges are blunt). An example is provided below:

An orthogonal grid is projected from the blunt edges of the blade.



Round Trailing Edge Radial Grid

This mesh works the same as the impeller grid type, except that the mesher distributes points along the curve that defines the geometric section.

See Also:

Defining the Blade Geometry Source

Selecting the Mesh Generation Method

The options in the *Meshing* tab determine which mesh generation method is used by the turbo wizard to generate the mesh.

The two options available are:

- Use the turbo wizard to automatically generate an elliptical grid.
- Use the turbo wizard to generate a closed surface mesh that can be meshed using the volume meshing models available in STAR-CCM+.

The **Mesh generation method** is applied to all blade rows.

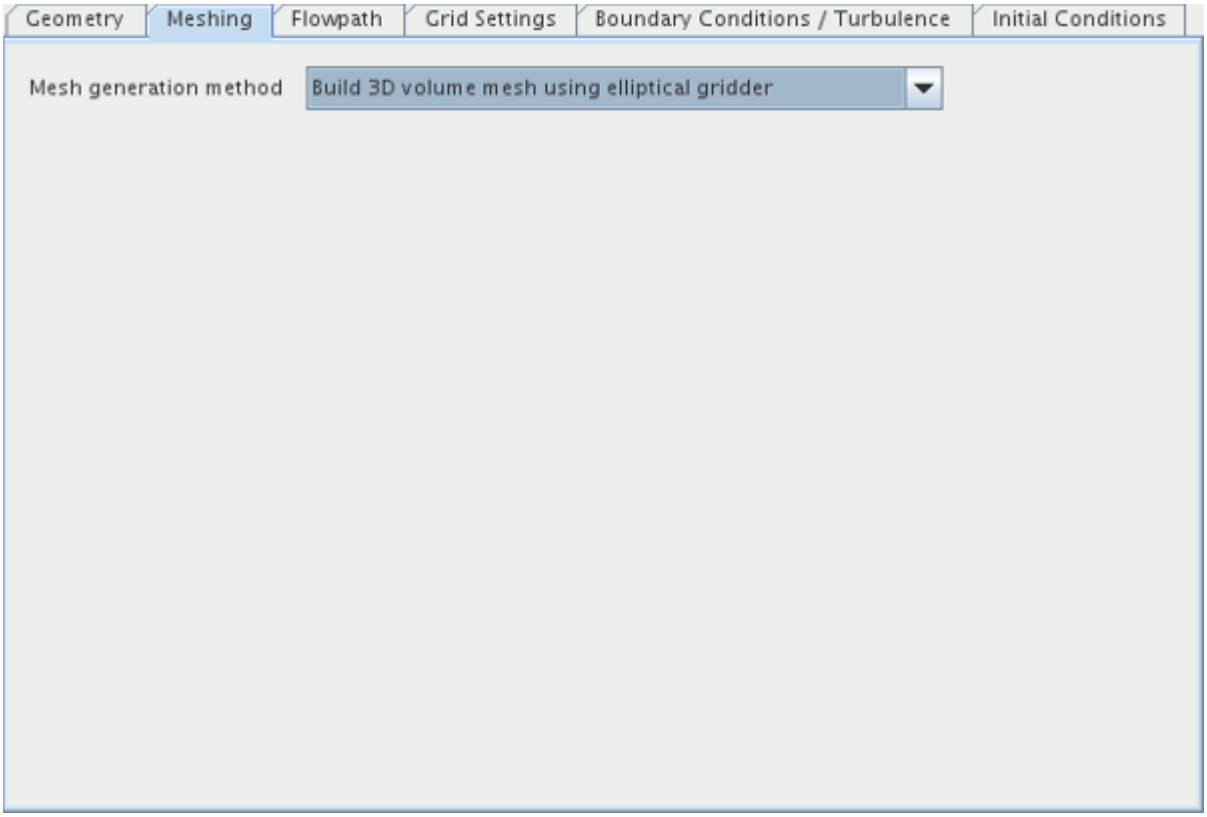
Contents:

Generating an Elliptical Grid Volume Mesh
Generating a Closed Surface Mesh

Generating an Elliptical Grid Volume Mesh

The default **Mesh generation method** in the turbo wizard is **Build 3D volume mesh using elliptical gridder**.

This option results in the generation of a volume mesh with an O-grid or C-grid around the blade, depending on the blade type, and an H-grid upstream and downstream of the blade. There are no additional settings required for this option.



See Also:

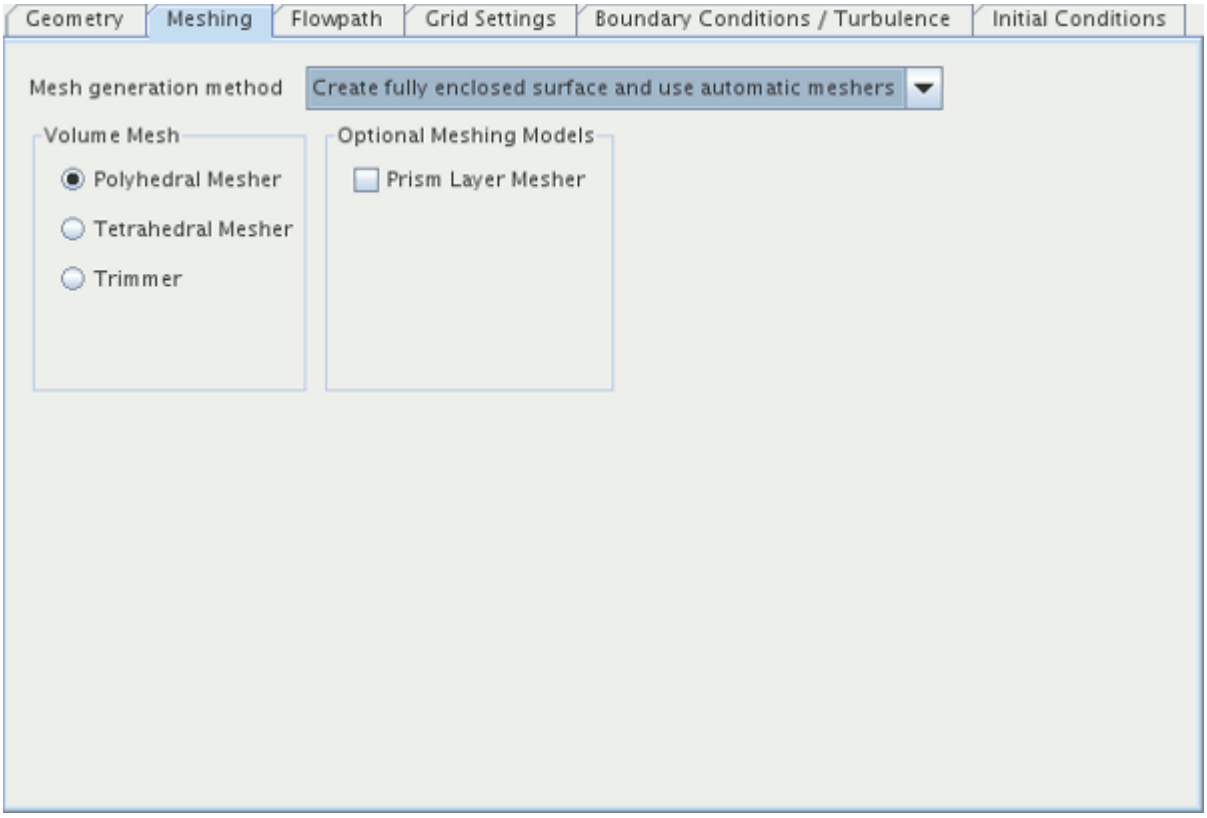
Selecting the Blade Type

Generating a Closed Surface Mesh

The turbo wizard can be used to generate a closed surface mesh for the blade geometry.

From this, a volume mesh can be generated outside of the turbo wizard, using one of the standard volume meshing models available in STAR-CCM+. This technique is useful when the blade geometry forms part of a larger model, as it allows entire flow domain to be meshed with the same shape cells, for example polyhedral.

This option is accessed by setting the **Mesh generation method** to **Create fully enclosed surface and use automatic meshers**. The volume meshing models are selected from the *Volume Mesh* and *Optional Meshing Models* group boxes, and when the surface mesh is generated the turbo wizard automatically sets up a mesh continuum with the selected models that can be used later to generate the volume mesh.



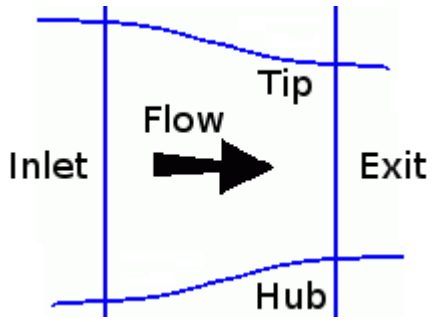
When this option is selected, the turbo wizard does not generate a volume mesh. The volume mesh must be generated outside of the turbo wizard, and the parameters of the volume mesh should be entered in the usual way.

Defining the Flow Path

The flow path geometry is a two-dimensional view of the machine showing the boundaries of the flow in the radial-axial plane.

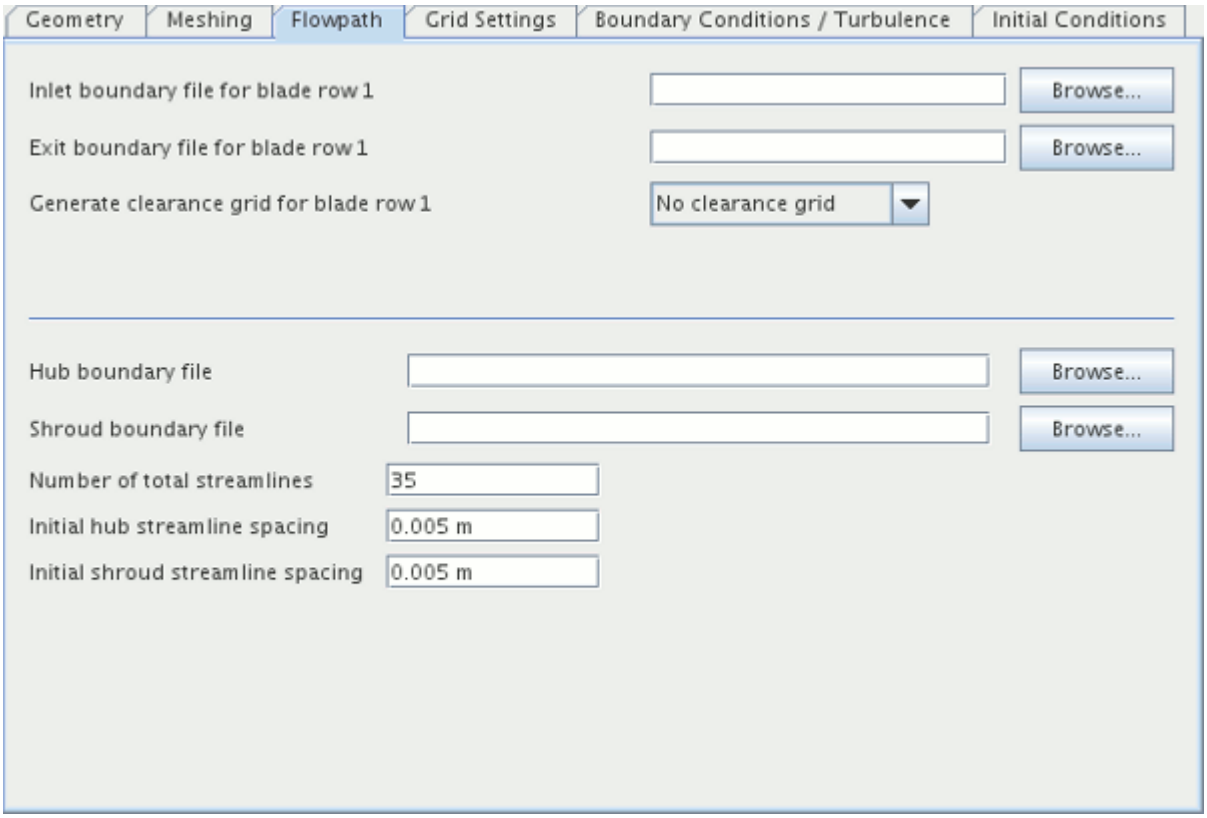
This path is defined by four cubic splines:

- Hub
- Tip
- Inlet
- Exit



These splines must be distinct (one cannot serve as both hub and inlet for example) and are defined in `.bndy` files. They need not meet at their endpoints, but must certainly cross in order to define a closed, planar area.

In the **Flowpath** tab, the full flow path is defined, and a clearance grid at the tip or hub can also be specified.



For the geometry to be valid all geometry splines must be defined:

- **Inlet/Exit boundary file for blade row X**

  The `.bndy` file for the inlet/exit of each row. Typically the spline data for the exit of one row are the same as the inlet data of the next row. You can use the same `.bndy` file for each but they must both be specified in each row.

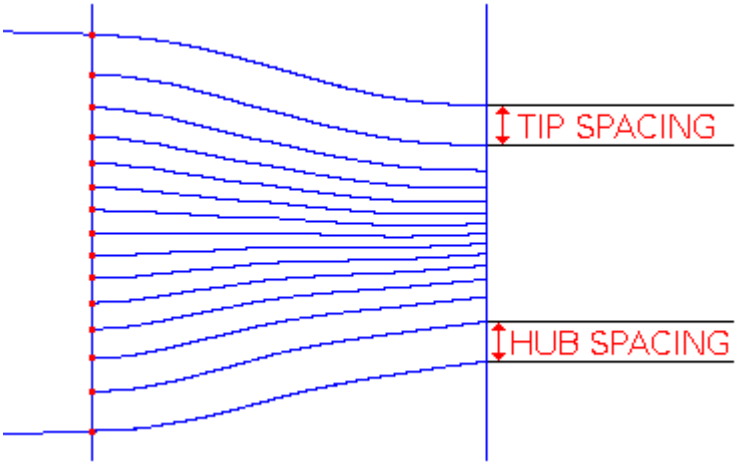
  The inlet and exit boundary definitions must go radially outward, that is from hub to shroud.

- **Hub/Shroud boundary file**

  The `.bndy` files for the hub and shroud. These files define the splines for the entire machine, and not for each blade row.

  The hub and shroud boundary definitions must go axially outward, that is from inlet to exit.

In the turbo wizard, the term streamlines refers to a series of splines used to define surfaces that subdivide the flow volume in the radial direction. The mesh is defined such that each streamline (or stream surface) lies on a cell boundary, so specifying the number of streamlines also determines the radial size of the mesh. Each streamline is marked with a red dot in the diagram below.



- **Number of streamlines**

  This specifies how many streamlines are generated from the hub to the shroud (including the hub and shroud streamlines). If a clearance grid is specified, this number also includes the streamlines in the clearance grid.
- **Initial hub/shroud streamline spacing**

  The spacing parameters, shown above, allow you to cluster the streamlines near the hub and shroud walls, resulting in larger spacing between streamlines in the middle of the flow path. The turbo wizard uses the spacing values as guidelines and modifies them based on the blade span and the number of streamlines. It is not possible to generate streamlines that are clustered around the middle of the flow path, instead the turbo wizard generates streamlines that are uniformly spaced.

Contents:

Specifying a Clearance Grid

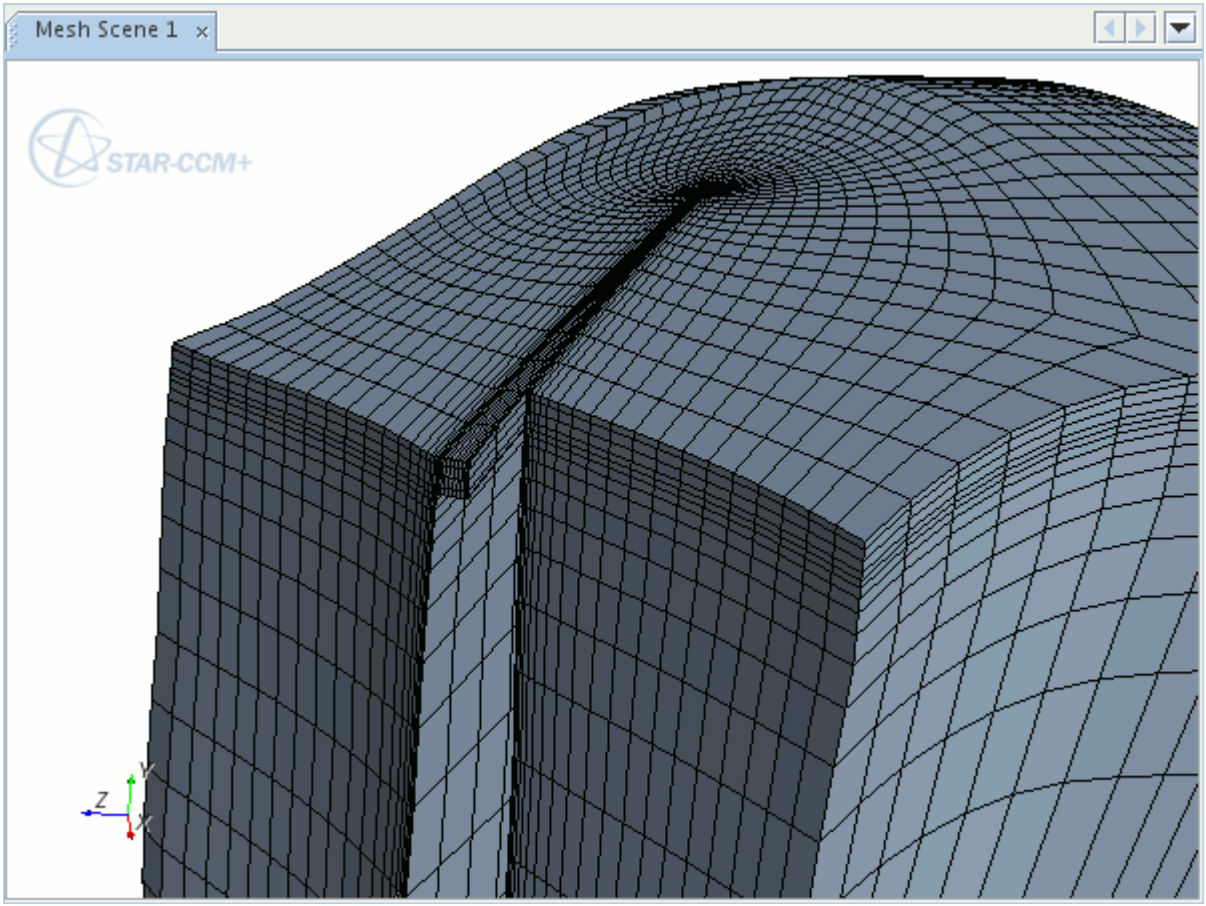
See Also:

Understanding the .bndy File Format

Specifying a Clearance Grid

For each blade row, a clearance grid can be generated in order to model the blade tip clearance at the hub or shroud.

It is generated into the flow domain from the hub or shroud, and ends at a streamline defined by either a fixed distance from the hub or shroud, or an imported `.bndy` file. The streamlines used by the clearance grid are taken from the total number of streamlines defined in the Flowpath tab, and are uniformly spaced.

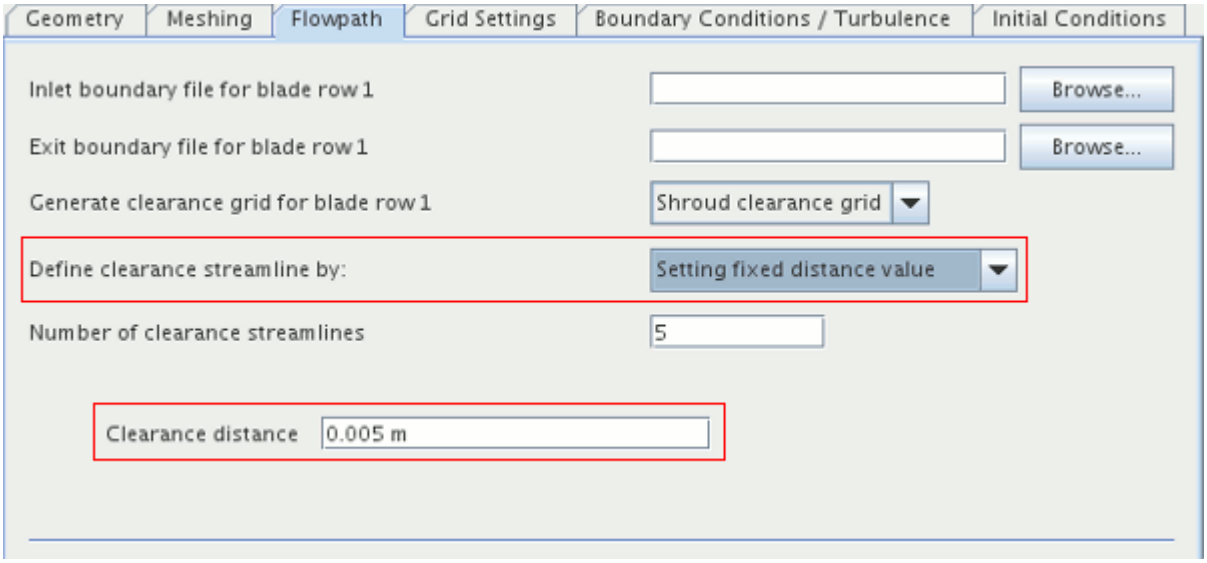


A clearance grid can be specified by using the drop-down menu for the **Generate clearance grid for blade row x** option and selecting either **Hub clearance grid** or **Shroud clearance grid**. The clearance grid can be deactivated by selecting **No clearance grid**.

Defining the Clearance Streamline

The clearance streamline limits the extent of the clearance grid. It is counted as one of the total number of streamlines in the flow path, and is defined either by setting a fixed distance from the shroud or hub, or by a `.bndy` file.

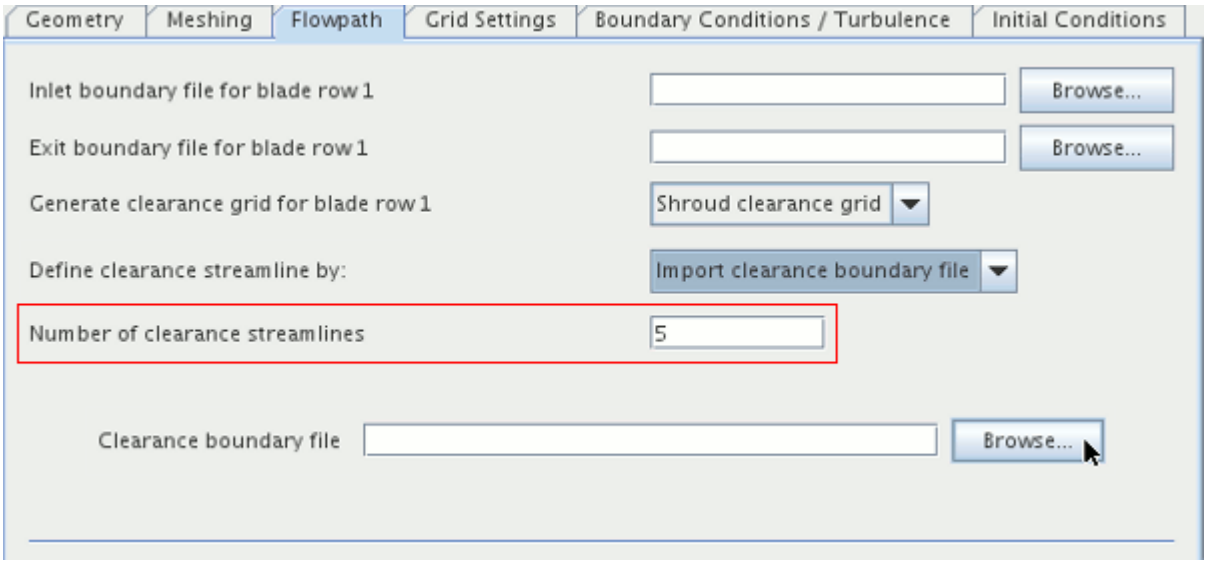
To define the clearance streamline by a specifying a fixed distance, select **Define clearance streamline by: Setting fixed distance value**. This option causes the **Clearance distance** box to appear, which can be used to set the distance of the streamline from the hub or shroud. The turbo wizard uses this value as a guideline to generate the clearance streamline.



To define the clearance streamline by importing a `.bndy` file, select **Import clearance boundary file** from the **Define clearance streamline by:** menu. This option causes the **Clearance boundary file** option to appear. The `.bndy` file can be selected by clicking the **Browse...** button and selecting the relevant file.

Setting the Number of Clearance Streamlines

The **Number of Clearance Streamlines** indicates how many of the total number of streamlines are in the clearance grid. This number includes the hub or shroud streamline and the clearance streamline. The minimum number of streamlines in the clearance grid is `5`.



See Also:

Defining the Number of Blade Rows

Understanding the .bndy File Format

Controlling the Grid Settings

The *Grid Settings* tab can be used to control the parameters that define the mesh.

They are accessed through three subtabs:

- Distribution

  These settings control the spacing and clustering of the grid around the blade.
- Number of points

  These settings control the dimensions of the mesh in terms of grid points.
- Block boundary settings

  These settings control the relative axial size of the three grid blocks.

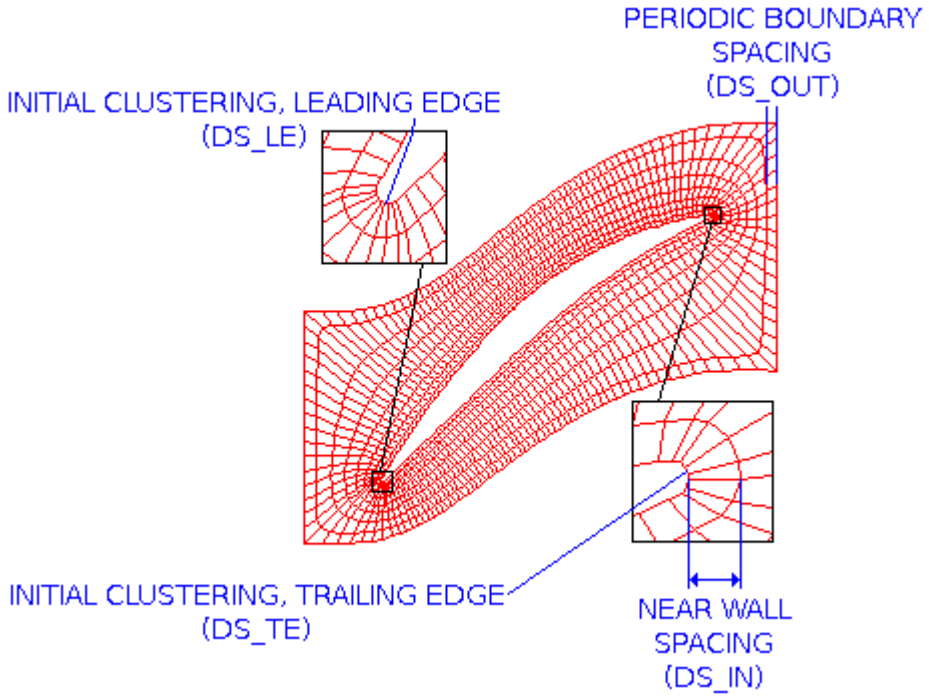
Contents:

Defining the Grid Distribution
Defining the Number of Grid Points
Defining the Grid Block Boundary Settings

Defining the Grid Distribution

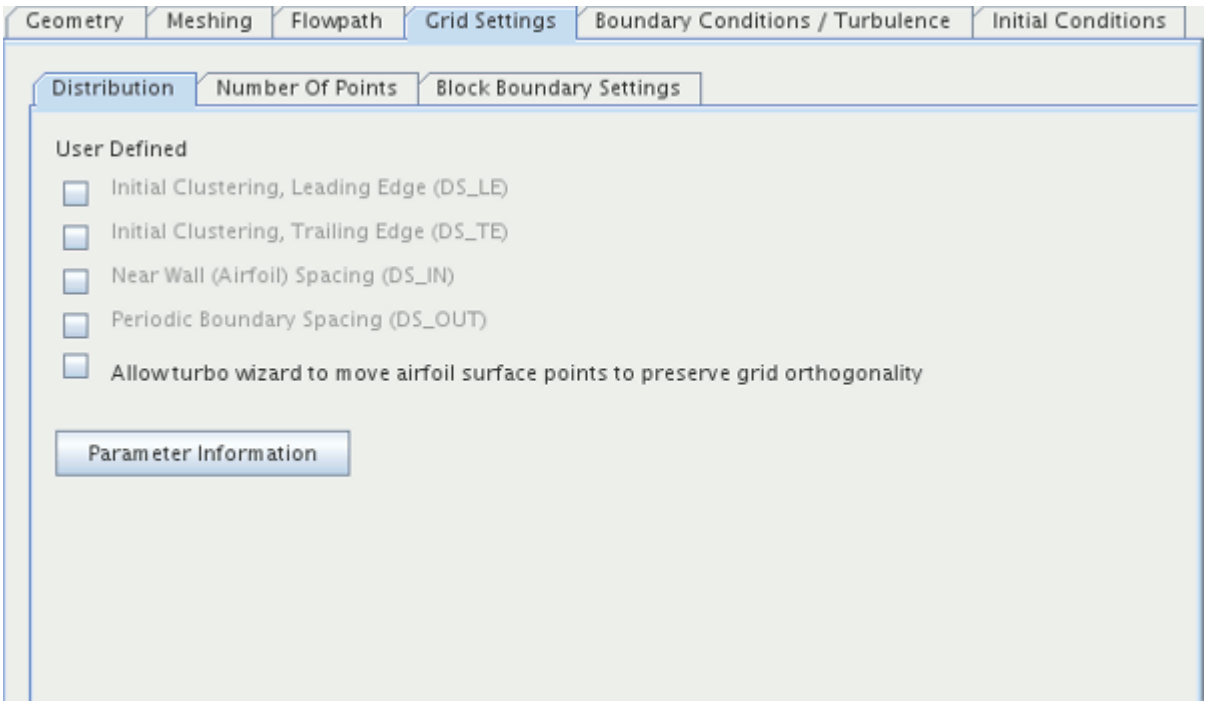
The settings in the *Distribution* subtab can be used to control several characteristic cell lengths of the mesh.

The characteristic cell lengths of the mesh are shown in the diagram below. This diagram can also be accessed from within the turbo wizard by clicking the **Parameter Information** button in the *Distribution* subtab.



- **Initial Clustering, Leading Edge (DS_LE)**—the minimum cell length parallel to the blade surface at the leading edge.
- **Initial Clustering, Trailing Edge (DS_TE)**—the minimum cell length parallel to the blade surface at the trailing edge.
- **Near Wall (Airfoil) Spacing (DS_IN)**—the minimum cell length perpendicular to the blade surface for cells adjacent to the blade.
- **Periodic Boundary Spacing (DS_OUT)**—the minimum cell length perpendicular to the blade surface for cells on the outer surface of the mesh segment surrounding the blade.

A custom value for any of these parameters can be specified by activating the **User Defined** checkbox for the relevant parameter. This causes a text box to appear in which the new value can be entered. The units used in the turbo wizard are defined in the Geometry tab.

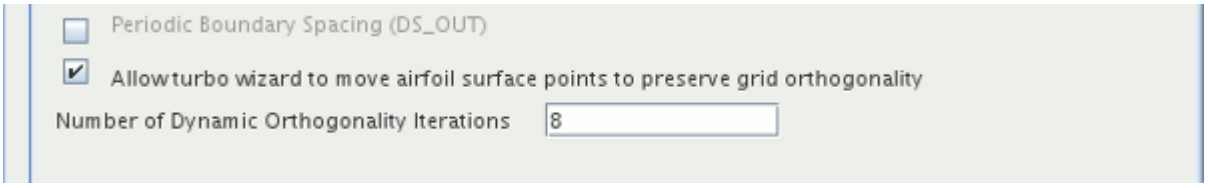


The last option in the *Distribution* tab is:

- **Allow turbo wizard to move airfoil surface points to preserve grid orthogonality**

  This option adjusts the position of points on the blade surface to ensure that grid lines meet the blade surface perpendicularly. The shape of the blade is not changed. This option is only available for O-grid blade types.

When this option is selected, you are able to specify the **Number of Dynamic Orthogonality Iterations**. The default value is `8`, and this should only be changed if you are an advanced user of the turbo wizard. By increasing this value you can improve the quality of the mesh at the blade surface, however the mesh generation process is much slower. The maximum applicable value is approximately `40`.



See Also:

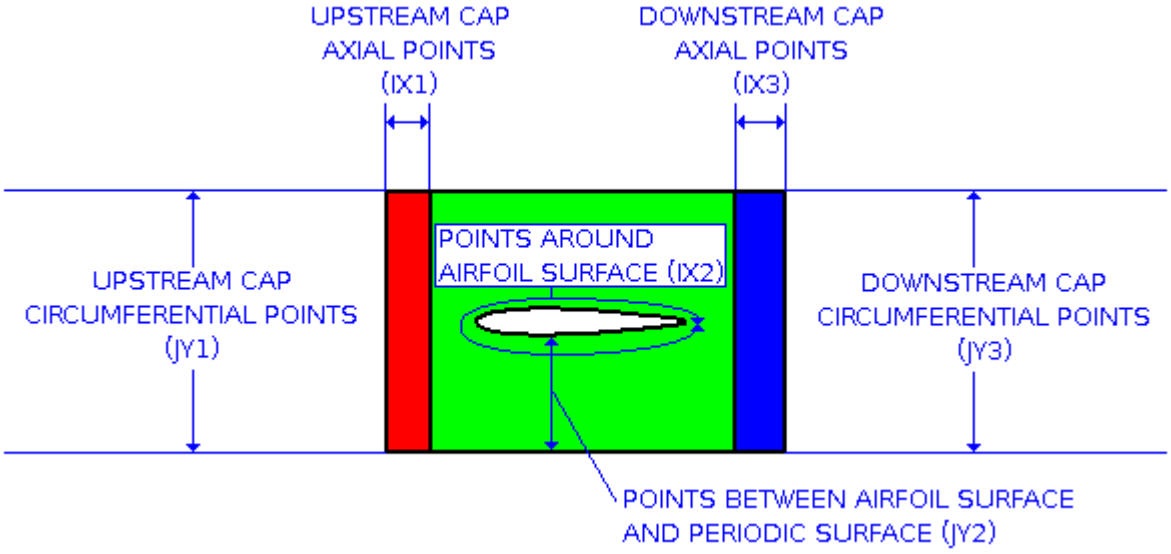
Setting Up the Blade Geometry

Selecting the Blade Type

Defining the Number of Grid Points

The settings in the *Number Of Points* subtab determine the dimensions of the mesh in terms of grid points (or number of cell boundaries).

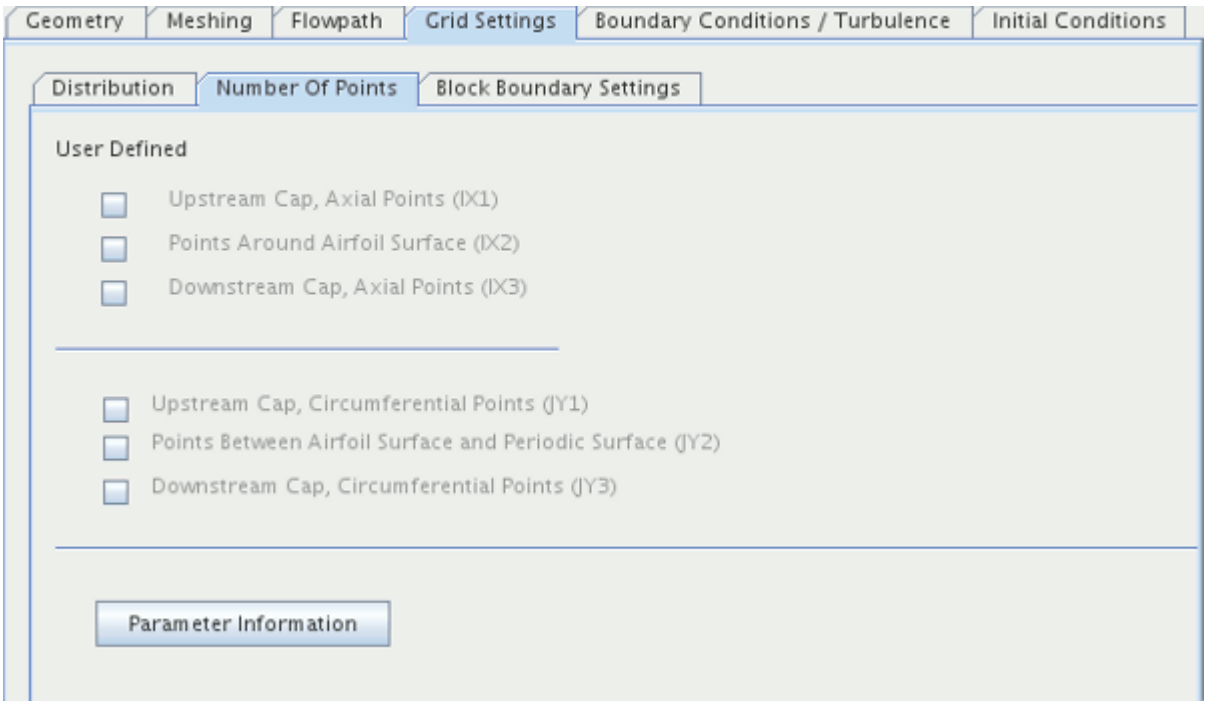
Mesh sizes can be specified for each section of the mesh, for example in the upstream H-grid. The options available are shown in the diagram below, which can also be accessed from within the turbo wizard by clicking the **Parameter Information** button in the *Number Of Points* subtab.



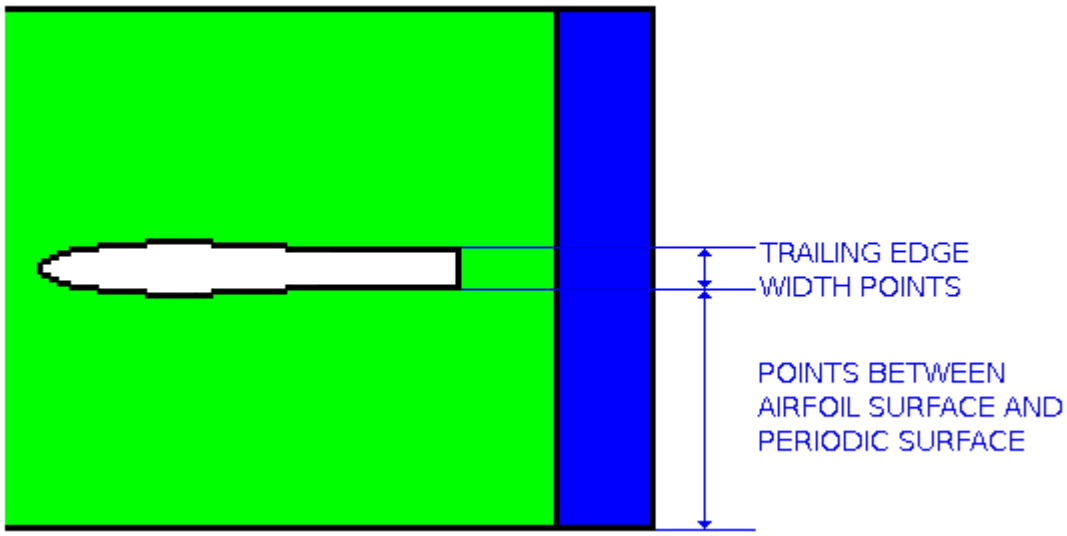
- **Upstream Cap, Axial Points (IX1)**—the mesh dimension in the flow direction for the upstream H-grid block.
- **Points Around Airfoil Surface (IX2)**—the mesh dimension around the blade for the O-grid or C-grid block.
- **DownStream Cap, Axial Points (IX3)**—the mesh dimension in the flow direction for the downstream H-grid block.
- **Upstream Cap, Circumferential Points (JY1)**—the tangential mesh dimension for the upstream H-grid block.

- **Points Between Airfoil Surface and Periodic Surface (JY2)**—the mesh-to-boundary dimension for the O-grid block.
- **Downstream Cap, Circumferential Points (JY3)**—the tangential mesh dimension for the downstream H-grid block.

A custom value for any of these parameters can be specified by activating the **User Defined** checkbox for the relevant parameter. This causes a text box to appear in which the new value can be entered.

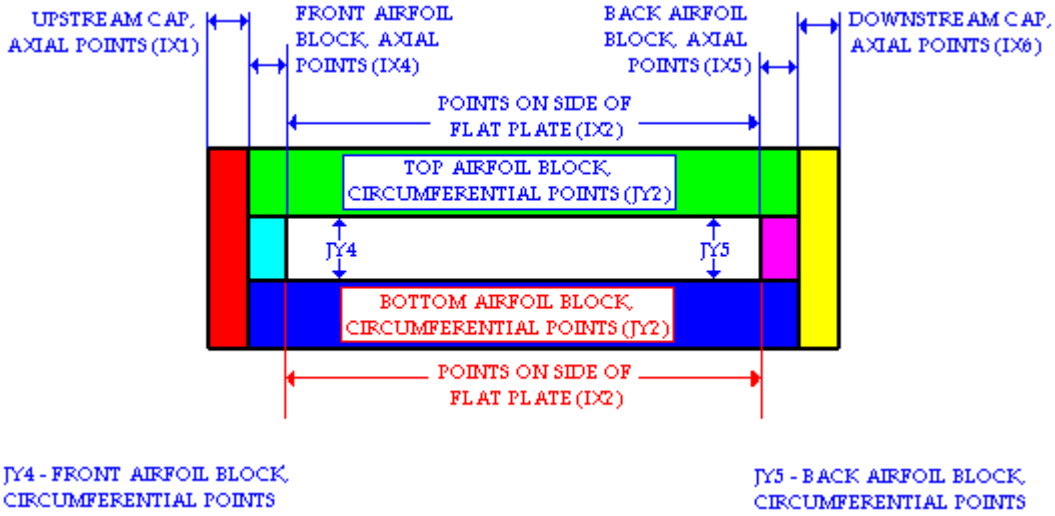


The **Trailing Edge Width Points** option replaces **Downstream Cap, Circumferential Points** when an impeller blade type is selected. In such cases, the dimensions of the mesh in the downstream H-grid are specified by this option and the **Points Between Airfoil Surface And Periodic Surface** option, as shown below.



The **Points Between Camber Line and Airfoil Surface** option becomes available in the *Number Of Points* tab when a clearance grid has been specified. This option specifies the mesh dimension across the thickness of the blade, as shown below. The turbo wizard uses this value as a guideline, and it can be altered depending on the dimensions used for the rest of the mesh.

The options available for the Flat Plate grid type are shown in the diagram below, which can also be accessed from within the turbo wizard by clicking the **Parameter Information** button in the *Number Of Points* subtab.



- **Upstream Cap, Axial Points (IX1)**—the mesh dimension in the flow direction for the upstream H-grid block.
- **Points On Side Of Flat Plate (IX2)**—the mesh dimension in the flow direction for the side of the flat plate.
- **Front Airfoil Block, Axial Points (IX4)**—the mesh dimension in the flow direction for the leading edge of the flat plate.
- **Back Airfoil Block, Axial Points (IX5)**—the mesh dimension in the flow direction for the trailing edge of the flat plate.
- **Downstream Cap, Axial Points (IX6)**—the mesh dimension in the flow direction for the downstream H-grid block.
- **Top Airfoil Block, Circumferential Points (JY2)**—the tangential mesh dimension for the top of the flat plate.
- **Bottom Airfoil Block, Circumferential Points (JY3)**—the tangential mesh dimension for the bottom of the flat plate.
- **Front Airfoil Block, Circumferential Points (JY4)**—the tangential mesh dimension for the leading edge of the flat plate.
- **Back Airfoil Block, Circumferential Points (JY5)**—the tangential mesh dimension for the trailing edge of the flat plate.

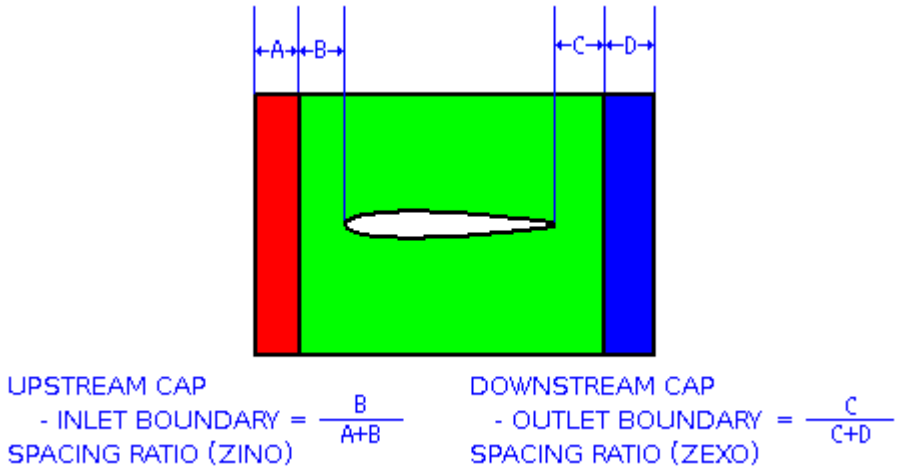
See Also:

Selecting the Blade Type
Specifying a Clearance Grid

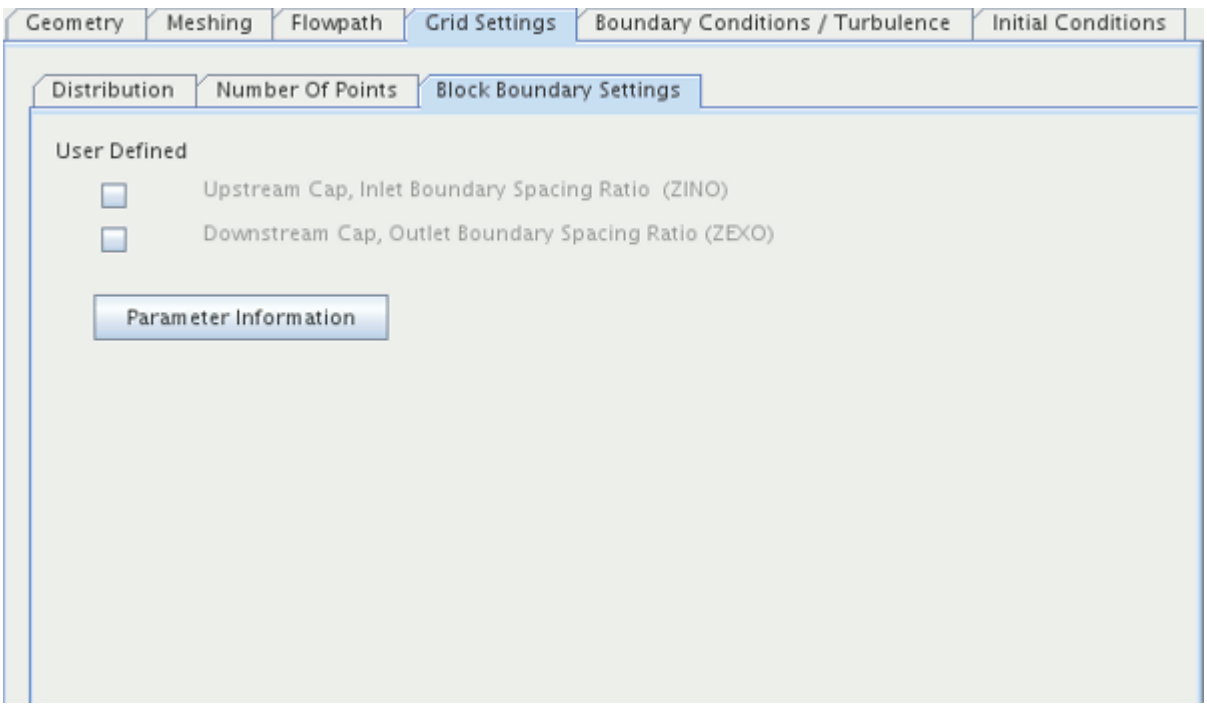
Defining the Grid Block Boundary Settings

The *Block Boundary Settings* subtab is used to control the relative axial size of the three grid blocks: upstream H-grid, O-grid, or C-grid around the blade, and the downstream H-grid.

These relative sizes are specified by two spacing ratios, shown below. This diagram can also be accessed from within the turbo wizard by clicking the **Parameter Information** button in the *Block Boundary Settings* subtab. When an impeller blade type is specified the **Downstream Cap, Outlet Boundary Spacing Ratio** option is unavailable as it cannot be modified for a C-grid.



A custom value for any of these parameters can be specified by activating the **User Defined** checkbox for the relevant parameter. This causes a text box to appear in which the new value can be entered.



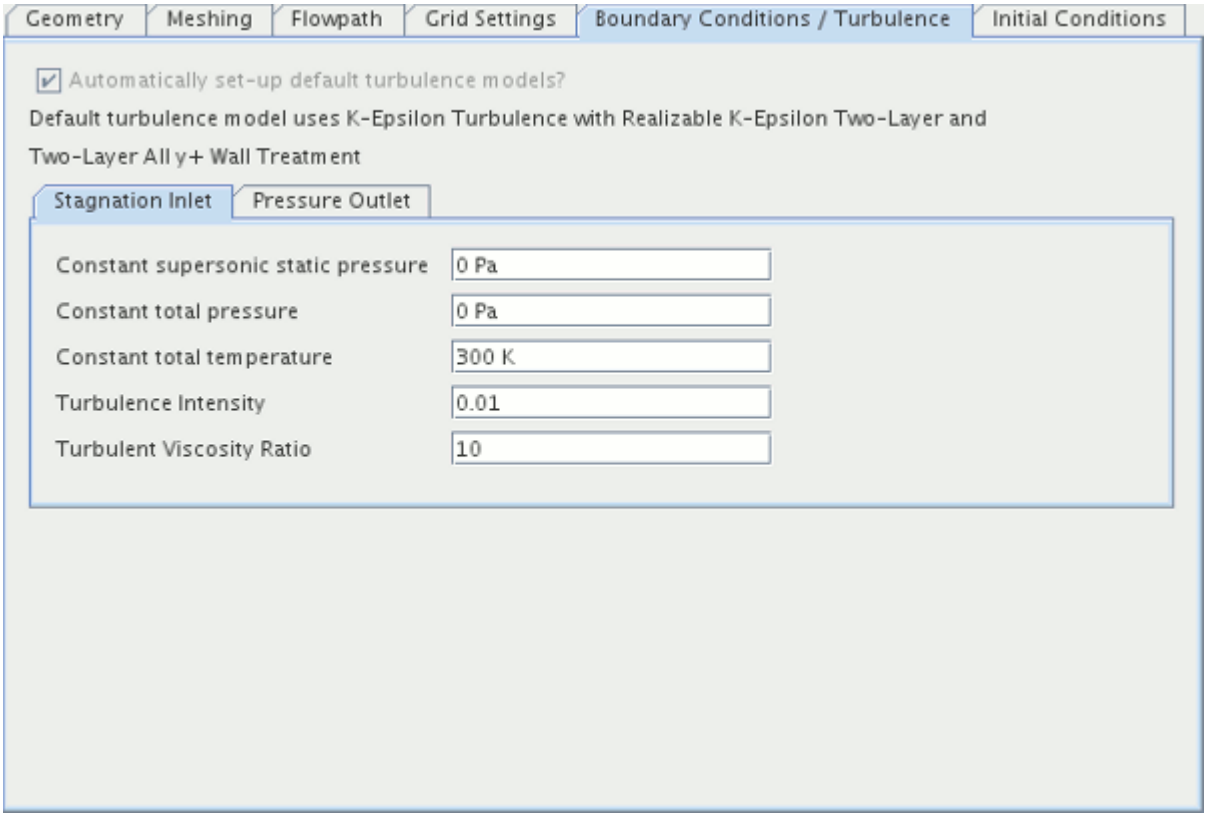
See Also:

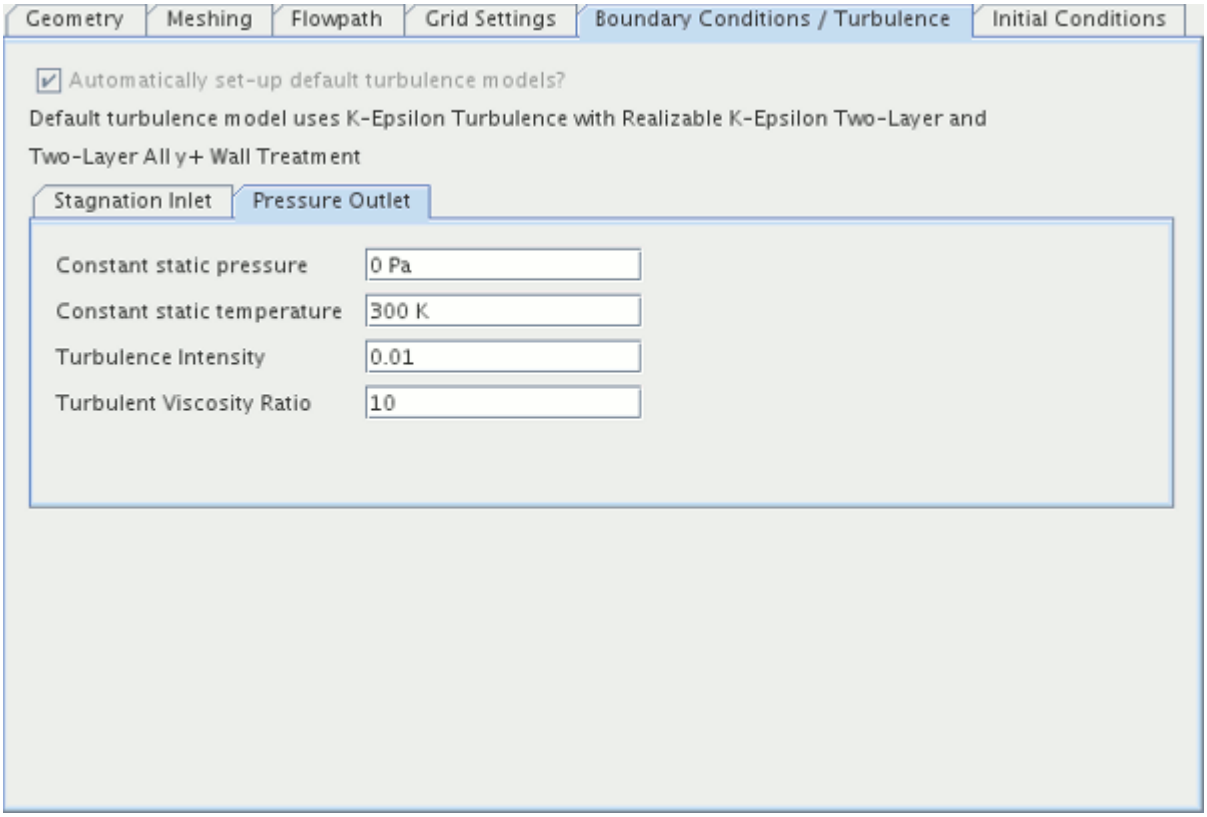
Selecting the Blade Type

Setting Up the Boundary Conditions

The turbo wizard automatically prepares the boundaries, interfaces, and physics models necessary for the simulation.

In the *Boundary Conditions / Turbulence* tab the boundary conditions for the inlet and outlet can be specified, and these are passed on to the relevant objects in the simulation tree. Activating the **Automatically set-up default turbulence models?** checkbox implements the default turbulence models, which are shown in the tab. If a rotation has been specified for the current blade row, this option is already selected and is grayed out.





If the default values in this tab are left unchanged, the turbo wizard prepares the simulation using these values.

See Also:

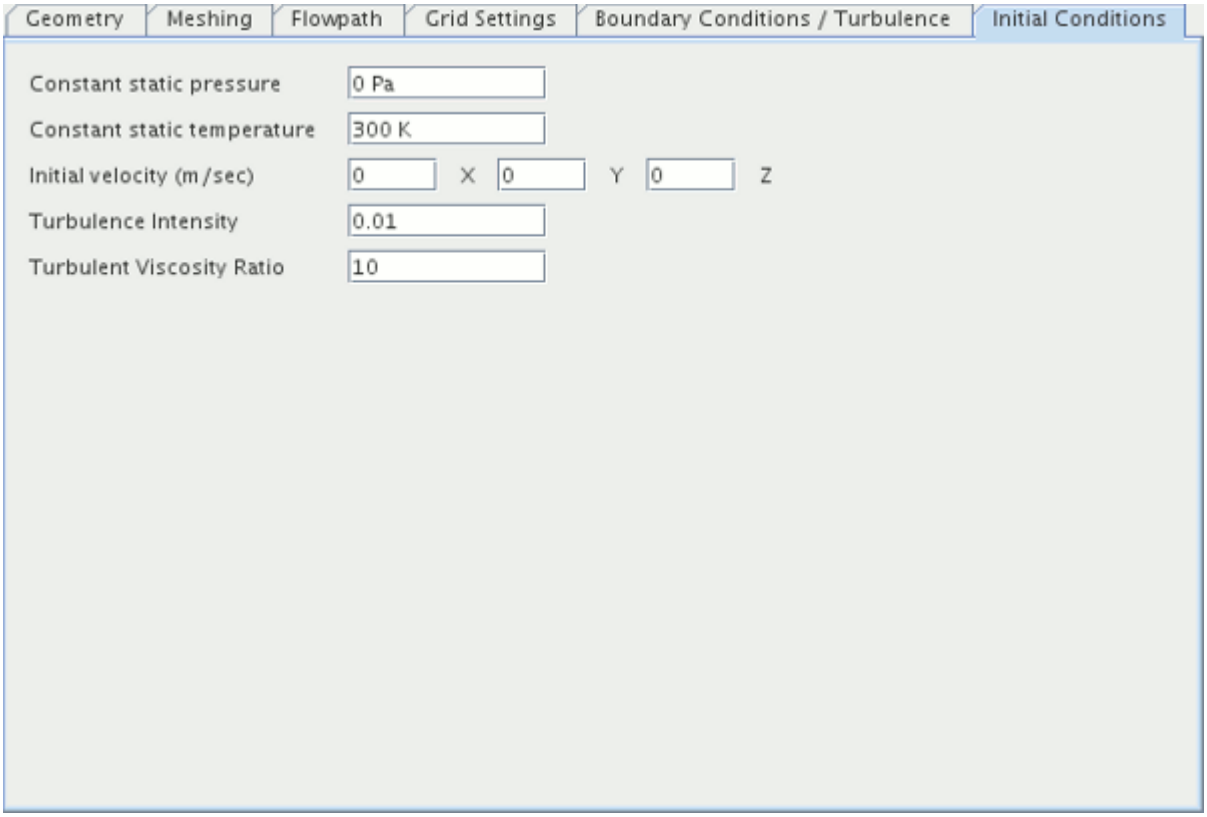
Defining Rotation

Defining the Number of Blade Rows

Setting Up the Initial Conditions

The turbo wizard automatically prepares the boundaries, interfaces, and physics models necessary for the simulation.

In the *Initial Conditions* tab, the initial conditions for the simulation can be specified. If the default turbulence models have been implemented in the boundary conditions / turbulence tab, the initial turbulence intensity and viscosity ratio can also be specified.



If the default values in this tab are left unchanged, the turbo wizard prepares the simulation using these values.

See Also:

Setting Up the Boundary Conditions

Working with Turbomachinery Input Files

The turbo wizard uses input files of the formats `.bndy`, `.estg`, and `.trbw`.

The first two you prepare. The most common means for creating these files is a typical spreadsheet program. However, it is also possible to write code in Fortran for this purpose.

Contents:

Understanding the .bndy File Format
Understanding the .estg File Format
Understanding the .trbw File Format

Understanding the .bndy File Format

This section describes the `.bndy` file format.

Each spline of the flow path geometry is defined by its (r, z) coordinates in a cylindrical coordinate system. The flow is assumed to move in the +z direction. To make viewing the model more convenient, this flow path is constructed on the q = 0 plane.

The flow path is defined by reading in vertex locations from an external `.bndy` file an example of which is shown below:

```
31  (Centimeters)
-22.860001 -17.779999 -12.700000 -7.620000
-3.070860  -2.435860 -1.800860  -1.165860
-0.264160   0.101600   0.736600  1.371600
2.006600   2.644140  3.279140   3.914140
4.521200   5.153660   5.788660  6.423660
7.058660   7.693660  8.328660   8.963660
9.598660  10.233660  10.868659 11.503659
12.138659  12.646660 15.240000
17.525999 17.525999 17.525999 17.525999
17.525999 17.541241 17.571720 17.602200
17.739361 17.805401 17.957800 18.122900
18.298161 18.440401 18.567400 18.674080
18.768059 18.867119 18.948400 19.029680
19.110960 19.177000 19.235420 19.293840
19.329399 19.364960 19.390360 19.405600
19.420839 19.433540 19.433540
```

The format of this file is:

- Line 1 - the number of points in the file and the units
- The rest of the lines are a space and carriage return separated list of numbers. In the example above, the first 31 points are the z positions of the spline and the second set of 31 numbers is the radial coordinate.

Understanding the .estg File Format

The `.estg` file is an external text file that contains the x, y, and z coordinates of the blade profile at each of two or more sections.

By default, the flow is assumed to move in the +z direction. An example follows:

```
#
# Input File: EZTurbo
#
# Original Data File: From NASA
# Modifications     : None
# Date              : 05 September 2000
#
# Case: Rotor 37
#
#       NPTS   NBLD CW  RPM
#       ====  ===== ==  ===
```

```
ROTOR    198    36   T    0
NPTS 198
AIRFOIL   1
    columns do not mean anything, first 198 points z, then x, y
   0.0281091   0.0289117   0.0298336   0.0308953
   0.0321196   0.0335267   0.0351471   0.0370063
   0.0391449   0.0415985   0.0444127   0.0476459
   0.0513517   0.0555958   0.0604470   0.0659739
   0.0722525   0.0793617   0.0873903   0.0964298
   0.1065792   0.1179478   0.1306549   0.1440579
   0.1589214   0.1753774   0.1935733   0.2136664
   0.2358321   0.2602583   0.2886035   0.3199688
   ...
AIRFOIL   13
   0.6383425   0.6388631   0.6394575   0.6401458
   0.6409383   0.6418501   0.6428990   0.6441055
   0.6454872   0.6470772   0.6489009   0.6509911
   0.6533939   0.6561421   0.6592839   0.6628652
   ...
```

Any line that does not begin with a recognized keyword (NPTS, AIRFOIL) or space is ignored by the importer. So in the preceding example all the lines beginning with `#` and the `ROTOR` lines are ignored. The different lines are:

- `NPTS` - the number of points per section. Each section must have the same number of points.
- `AIRFOIL` - the section number starting with the hub section and ending at the shroud. The importer reads the file until the end, there is no need to specify up front how many sections are in the file.
- `<SPACE>` - a data line. The data is simply a series of space or line feed separated numbers. In the above example, the first 198 points are the z values, the next 198 points are the x values and the next 198 points are the y values of the set of x, y, z coordinates defining the profile. There should be `(NPTS * 3)` values in each section.

Understanding the .trbw File Format

The `.trbw` file is a STAR-CCM+-generated non-compressed XML file which stores all of the turbo wizard user interface settings in an external file.

This file can be generated by clicking the **Save Turbo Wizard Settings** button.

To restore the saved settings in the turbo wizard, simply click the **Load Turbo Wizard Settings** button and navigate to the `.trbw` file. Opening the file loads it back into the turbo wizard interface.

Caution: In typical situations, you should not attempt to edit the XML file, as this might corrupt it. The file is fairly descriptive in what parameters are being stored and saved, and so you can modify this file at your own risk.

Saving and Loading Turbo Wizard Data

Settings and values entered in the turbo wizard dialog are referred to as turbo wizard data.

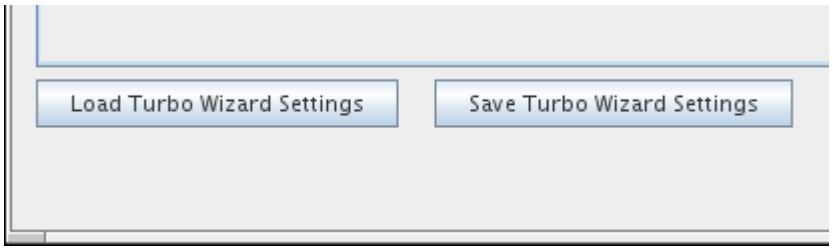
It is useful to store this data so that it can be used in later simulations, and this is possible by saving it in a `.trbw` file. This file format is specific to the turbo wizard, and files of this type can be loaded into the wizard as described below. The options to save and load turbo wizard data are only accessible if the turbo wizard has been activated.

Note: Saving and loading Turbo Wizard Data is not currently recorded by user macros in STAR-CCM+.

Saving Turbo Wizard Settings

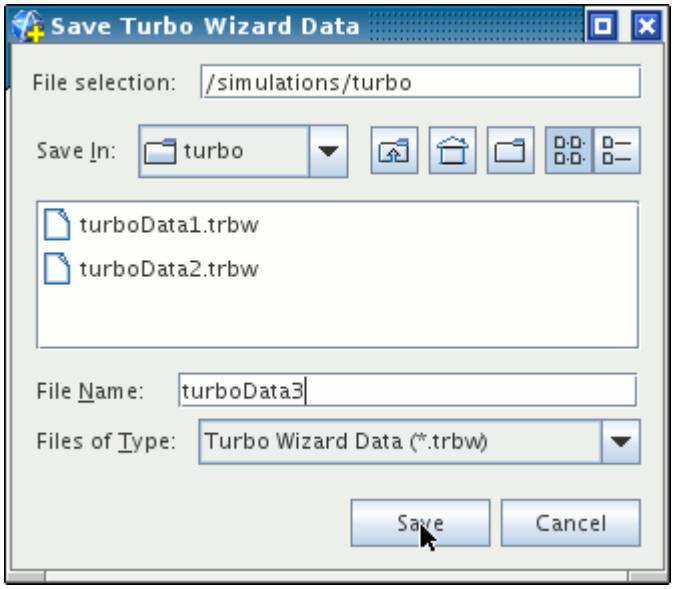
To save the turbo wizard data for use with other turbomachinery simulations:

1. Click the **Save Turbo Wizard Settings** button at the bottom of the *Import turbo wizard geometry file options* dialog.



   This opens the *Save Turbo Wizard Data* dialog with the Files of Type filter set to `*.trbw` format.

2. To save the data, navigate to the preferred location, enter an appropriate file name and click on the **Save** button.



See Also:

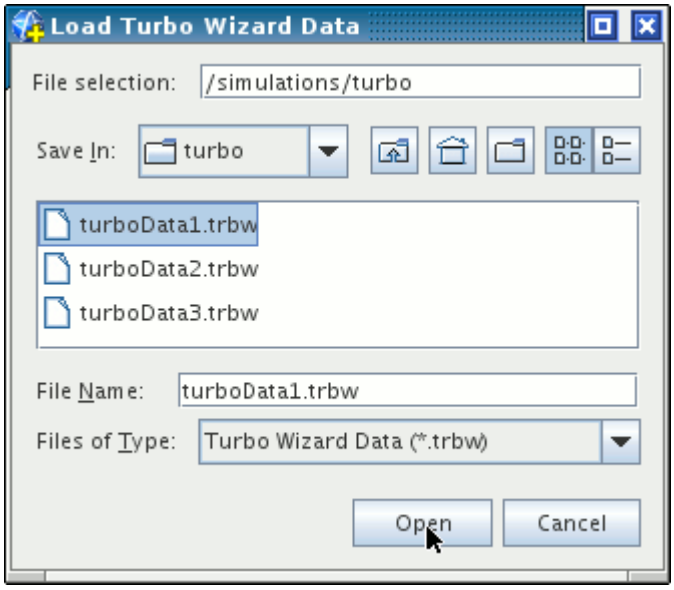
Activating the Turbo Wizard

Loading Turbo Wizard Settings

To import turbo wizard data into the current simulation:

1. Click the **Load Turbo Wizard Settings** button.

   This opens the *Load Turbo Wizard Data* dialog.
2. To load the data navigate to the appropriate location, select the turbo wizard data file and click the **Open** button.



Using the Fire and Smoke Wizard

This section describes the fire and smoke wizard feature of STAR-CCM+.

This section explains the following:

- What is the fire and smoke wizard?
- Other capabilities of STAR-CCM+ that can be used with the fire and smoke wizard for problems of this type
- Setting up the fire properties

Contents:

What is the Fire and Smoke Wizard?
STAR-CCM+ Capabilities Related to Fire and Smoke
Setting Up the Fire Properties

What is the Fire and Smoke Wizard?

The fire and smoke wizard is an automatic simulation tool for simulating a building fire.

Its capabilities for this application include :

- Fire regions and sources
- Soot sources
- Velocity inlet boundaries for ventilation
- Fire doors
- Heat and smoke detectors

STAR-CCM+ Capabilities Related to Fire and Smoke

This section lists some of the other STAR-CCM+ features that can help model fire and smoke.

The fire and smoke wizard is the primary tool for these types of problems, but there are several other STAR-CCM+ features that can help with the CFD solution:

- Modeling Options:
  - Stationary
  - Three-Dimensional
- Radiation Models:
  - Discrete Ordinates Method (DOM)
- Turbulence Models:
  - Spalart-Allmaras
  - K-Epsilon, K-Omega...

- Boundary Conditions:
  - Velocity Inlet
  - Stagnation
  - Pressure
- Solvers:
  - Second-Order
  - Segregated
  - Algebraic Multigrid Acceleration
- Reports, monitors, and plots
- Derived parts

Setting Up the Fire Properties

This section outlines the process for setting up the fire properties.

This feature would be used after you generate a mesh. The process involves the following:

- Activating the wizard
- Creating fire volumes
- Setting up fire dynamics
- Specifying fire doors
- Setting up ventilation
- Preparing the soot source
- Adding heat and smoke detectors
- Specifying optical properties
- Using post-processing planes
- Setting graph parameters

Contents:

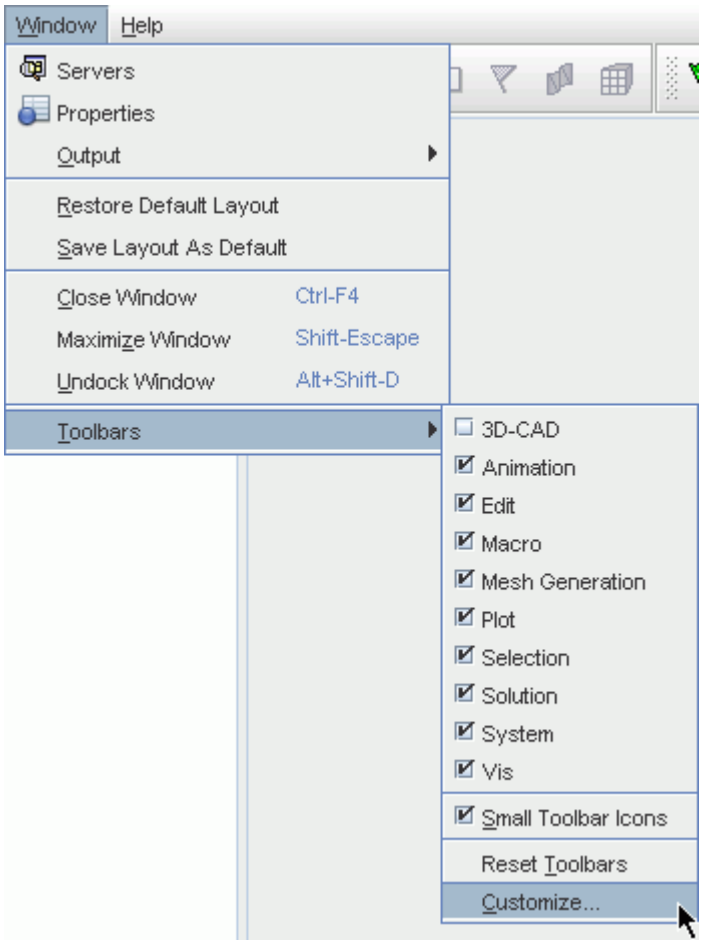
Activating the Fire and Smoke Wizard
Creating Fire Volumes
Setting Up Fire Dynamics
Specifying Fire Barriers
Setting Up Ventilation
Preparing the Soot Source
Adding Heat and Smoke Detectors
Specifying Optical Properties
Using Post Processing Planes
Setting Graph Parameters

Activating the Fire and Smoke Wizard

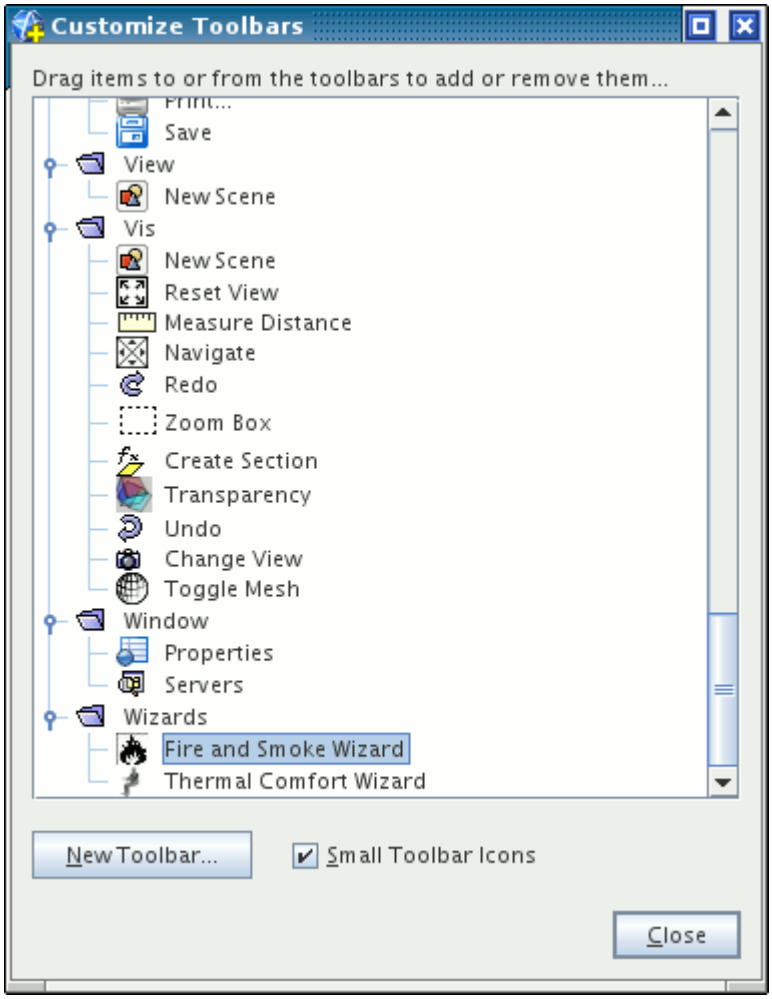
The properties of the fire are set using the **Fire and Smoke Wizard**.

To begin accessing this feature:

1. Select **Windows** > **Toolbars** > **Customize** from the menu bar.

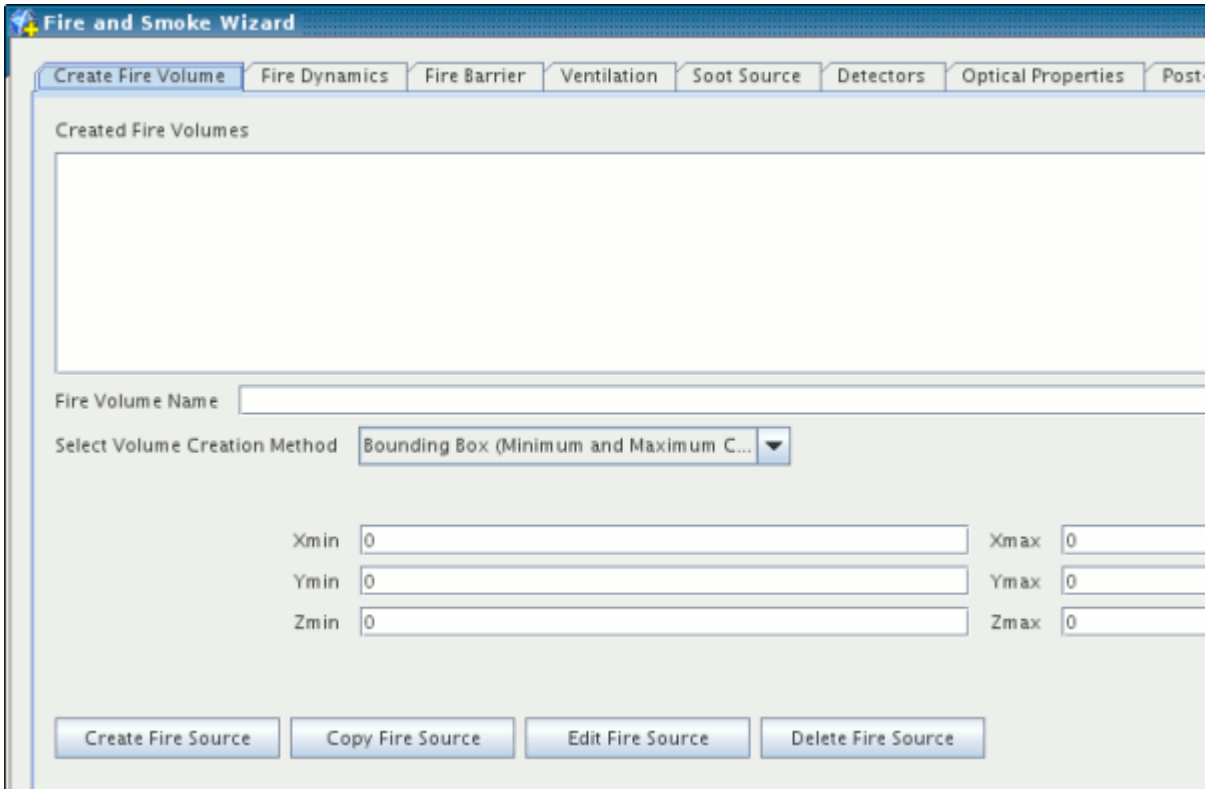


This opens the *Customize Toolbars* dialog as shown in the following screenshot.



2. Drag and drop the **Fire and Smoke Wizard** icon onto the end of one of the toolbars near the top of the workspace.
3. Click **Close** on the *Customize Toolbars* dialog.
4. Click the (**Fire and Smoke Wizard**) button.

   This opens the *Fire and Smoke Wizard* dialog.



There are two broad categories of detail that this dialog provides:

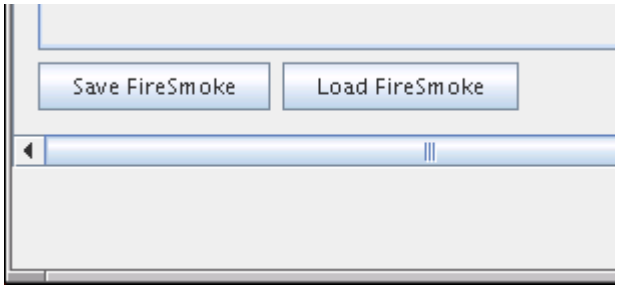
- Required fire characteristics are provided in the following tabs:
  - *Fire Dynamics*
  - *Soot Source*
  - *Optical Properties*
- Optional characteristics of the fire and building are provided in the following tabs:
  - *Create Fire Volume*
  - *Fire Barrier*
  - *Ventilation*
  - *Detectors*

Post-Processing Planes let you display data in precisely defined sections and output that data to images.

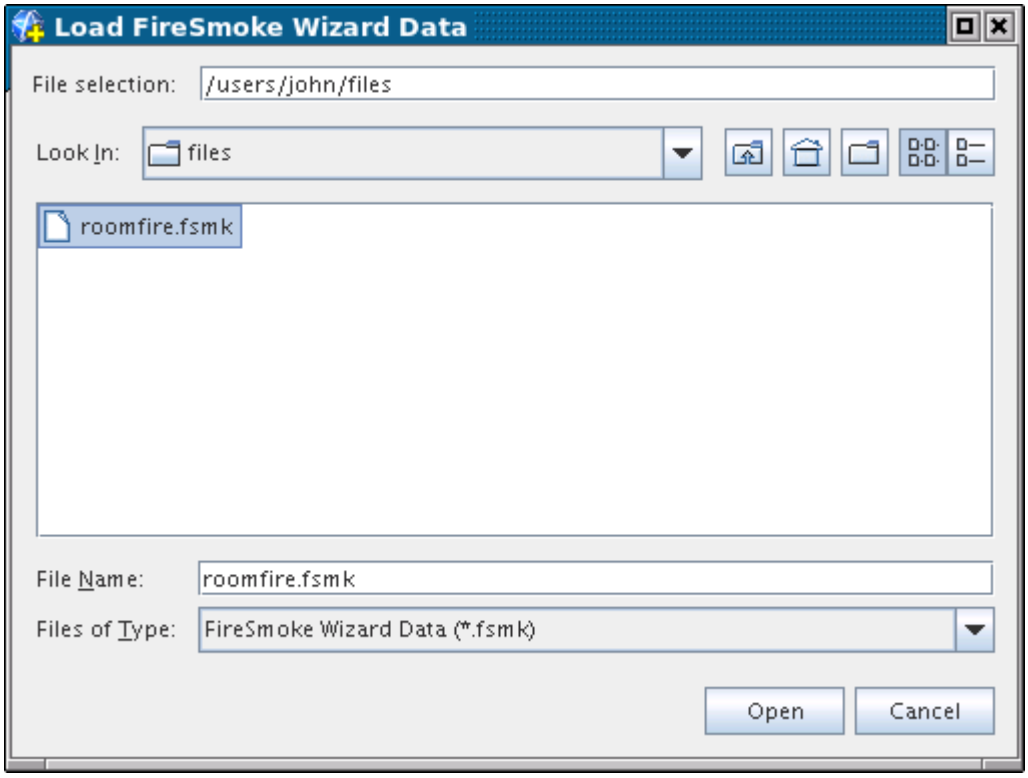
5. Finally, after you click **Apply**, the wizard automatically saves your settings to an external `.fsmk` file.

   This allows you to import those settings into another fire and smoke simulation.

6. To import the file, click the **Load FireSmoke** button near the bottom of the dialog.

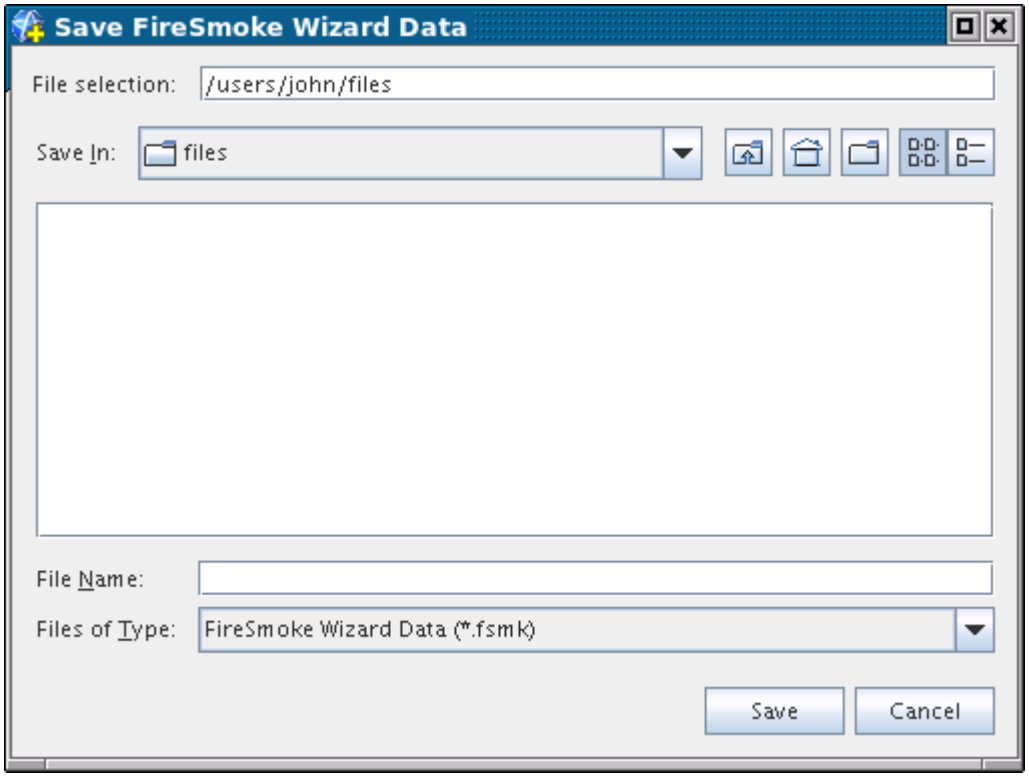


This activates a standard file-opening dialog with the Files of Type filter set to the `.fsmk` format.



7. Navigate to the file and click **Open**.
8. Additionally, before you click **Close** in the wizard, you can save the file with a name and path of your choice by clicking the **Save FireSmoke** button.

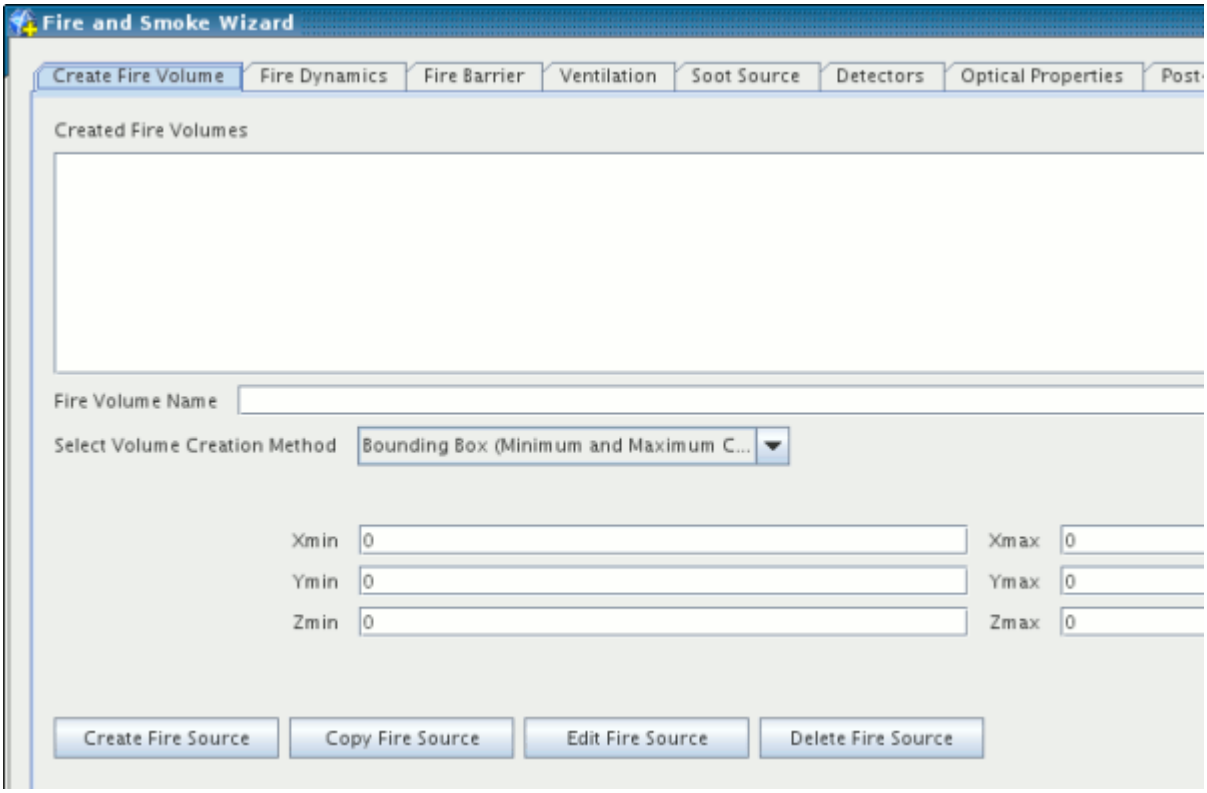
   This opens a *Save* dialog.



9. Navigate to the preferred location, enter a file name, and click **Save**.

Creating Fire Volumes

The *Create Fire Volume* tab allows you to define a space for the fire that is an alternative to the regions created in the mesh generation process.



Begin by entering a name for the fire volume in the **Fire Volume Name** text box.

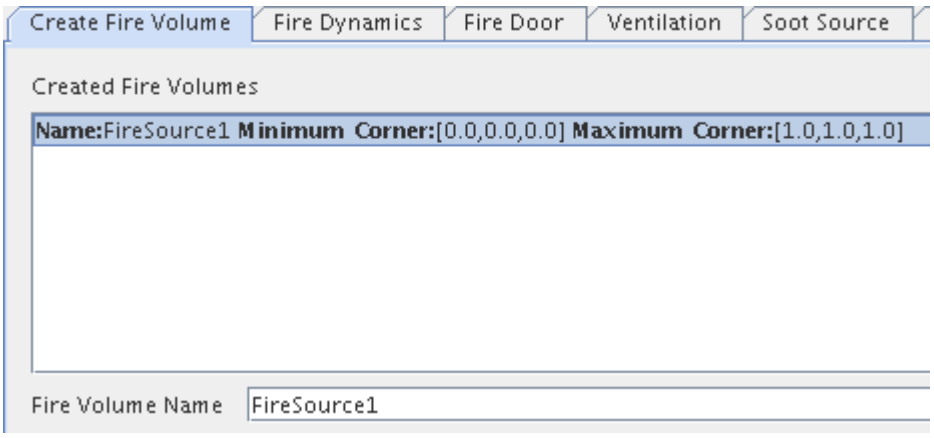
Next select the volume creation method from the drop-down list:

- A bounding box with minimum and maximum coordinates
- The absolute size with a midpoint and size bounds

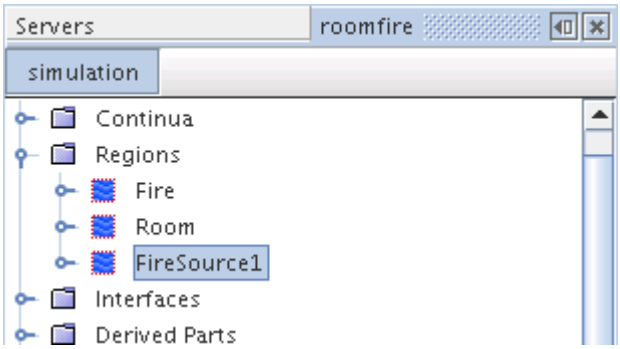
The fields below the drop-down list change with your selection of method. Enter the appropriate coordinates.

Once you enter values, you can create the volume by clicking the **Create Fire Source** button.

The fire volume that you create appears with its pertinent data in the Created Fire Volumes list.



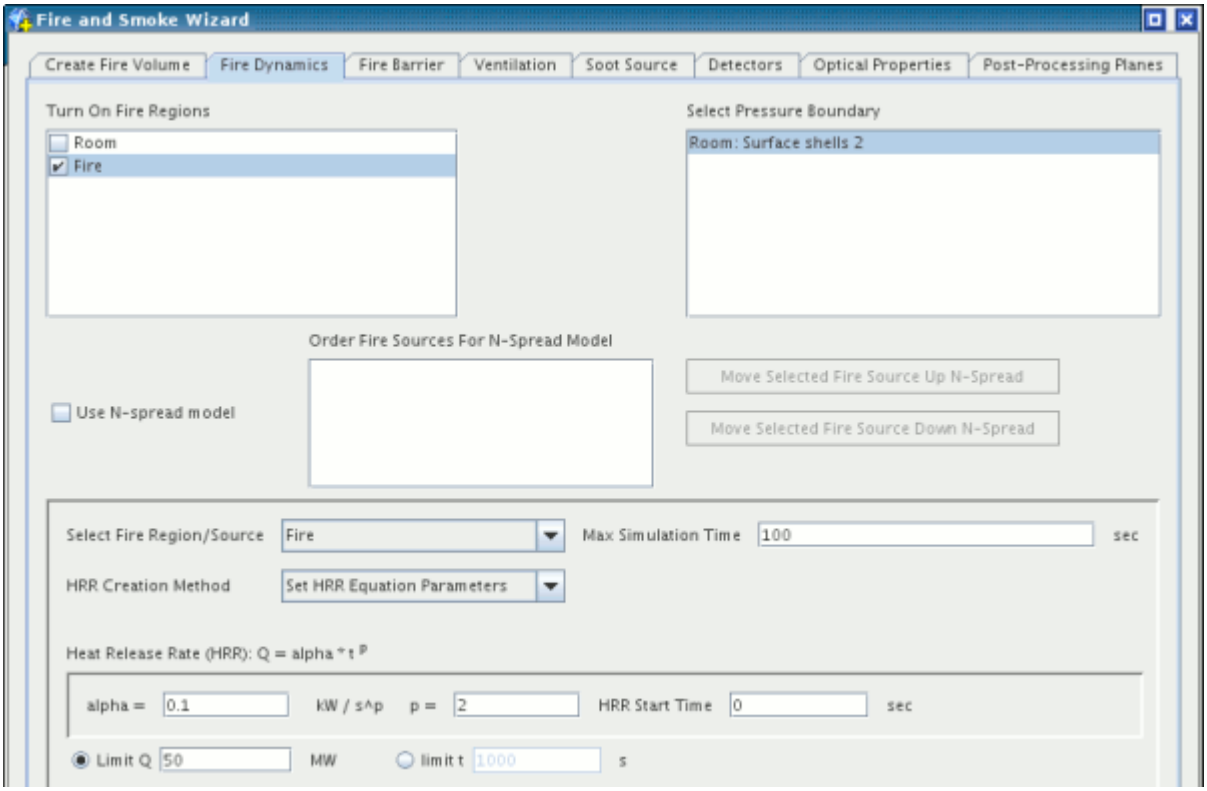
Once you finish with the wizard by clicking **Apply**, the fire volume also appears in the object tree within the **Regions** node.



It is also possible to manage the fire volumes in the list. For example, if you wish to create a second fire volume similar to the first, you need not repeat the entire creation process. Simply select the name of the fire volume in the list and click **Copy Fire Source**. An additional fire volume appears in the list. Select that copy, make any changes you wish, and then click **Edit Fire Source** to apply your changes. You can also delete fire volumes by selecting them in the list and clicking **Delete Fire Source**.

Setting Up Fire Dynamics

Set up the dynamic properties of the fire by selecting the *Fire Dynamics* tab.

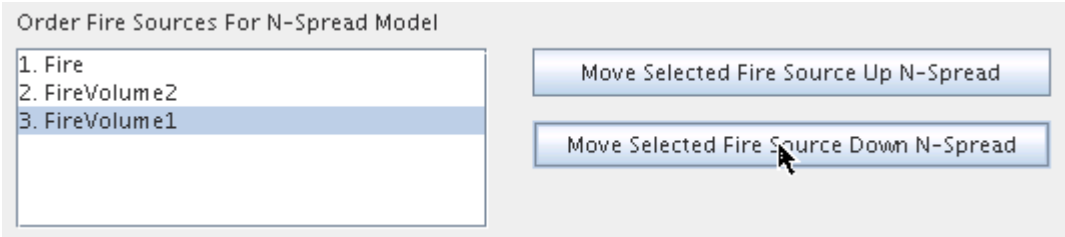


The *Turn On Fire Regions* list in the upper left lets you tick the checkboxes of regions that represent the fire, including any fire volumes that you may have created in the previous tab.

Select the pressure boundary in the list in the upper right.

If you wish to use an N-spread model, then tick the checkbox.

This makes the fire sources appear in the list, and you can use the buttons to the right of it to change the sequence. Simply select a fire source in the list and click the button to move it up or down within the list.

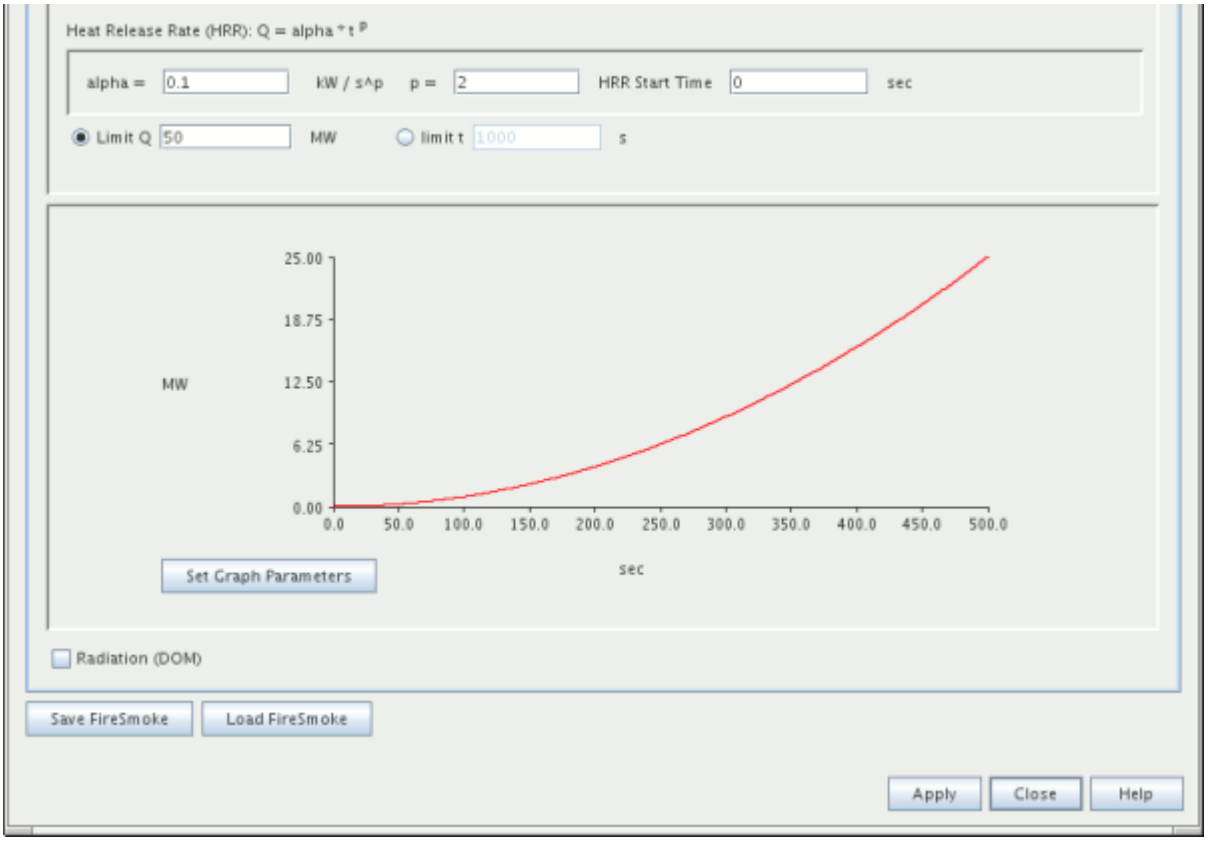


Select a fire volume from the drop-down list, and do as follows:

- Specify the maximum simulation time for it.
- Specify the values for alpha, p, and the HRR start time.
- Use the radio buttons to select whether you want to specify the limit for Q or for t, and then enter the value.
- Select another fire volume from the list, if applicable, and repeat the above steps.

The fire and smoke wizard retains your entries for each of the fire volumes.

This wizard also sets up a monitor plot of the heat release rate, which is previewed in the panel.



To control the parameters of this plot, click the **Set Graph Parameters** button.

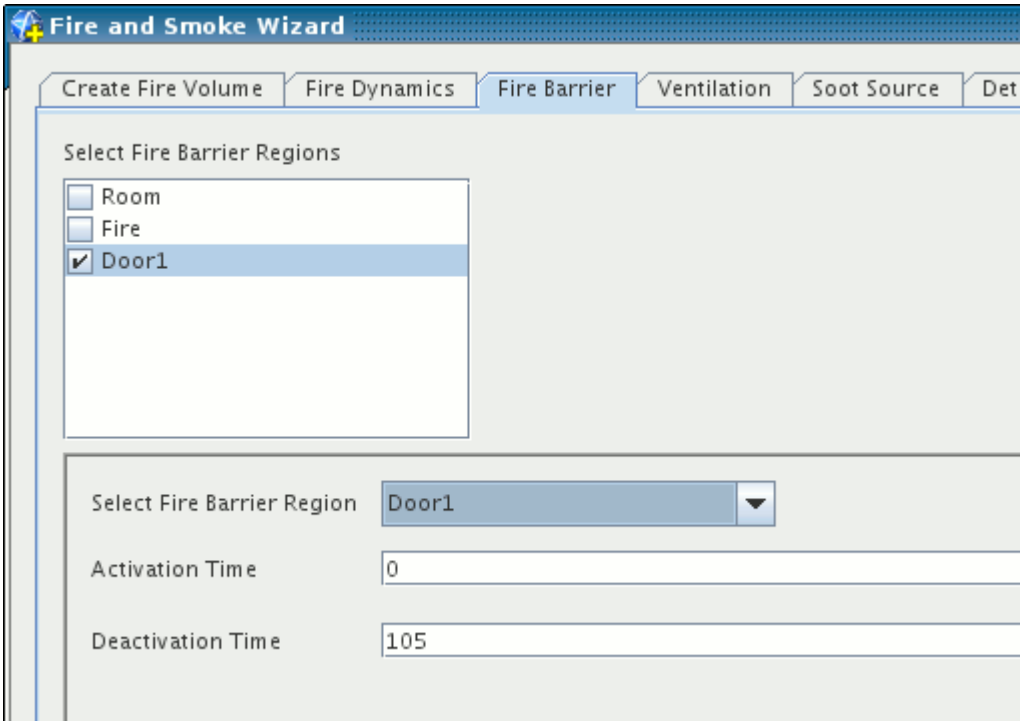
You can also tick a checkbox to use the DOM (Discrete Ordinates Method) radiation model. When you activate the radiation model, STAR-CCM+ calculates the absorption coefficient of the air and gas mixture [1] as:

$$\begin{aligned} a &= 0.01, \text{if} T < 323\text{K} \\ a &= 0.01 + (3.49/377)(T - 323), \text{if} 323K \leq T < 700\text{K} \\ a &= 3.5 + (3.5/700)(T - 700), \text{if} T > 700\text{K} \end{aligned} \tag{18}$$

[1] *A.J. Grandison, E.R. Galea, and M.K. Patel. May 2001. "Fire Modeling Standards/Benchmark—Report on SMARTFIRE Phase 2 Simulations", Fire Safety Engineering Group, University of Greenwich, May 2001.*

Specifying Fire Barriers

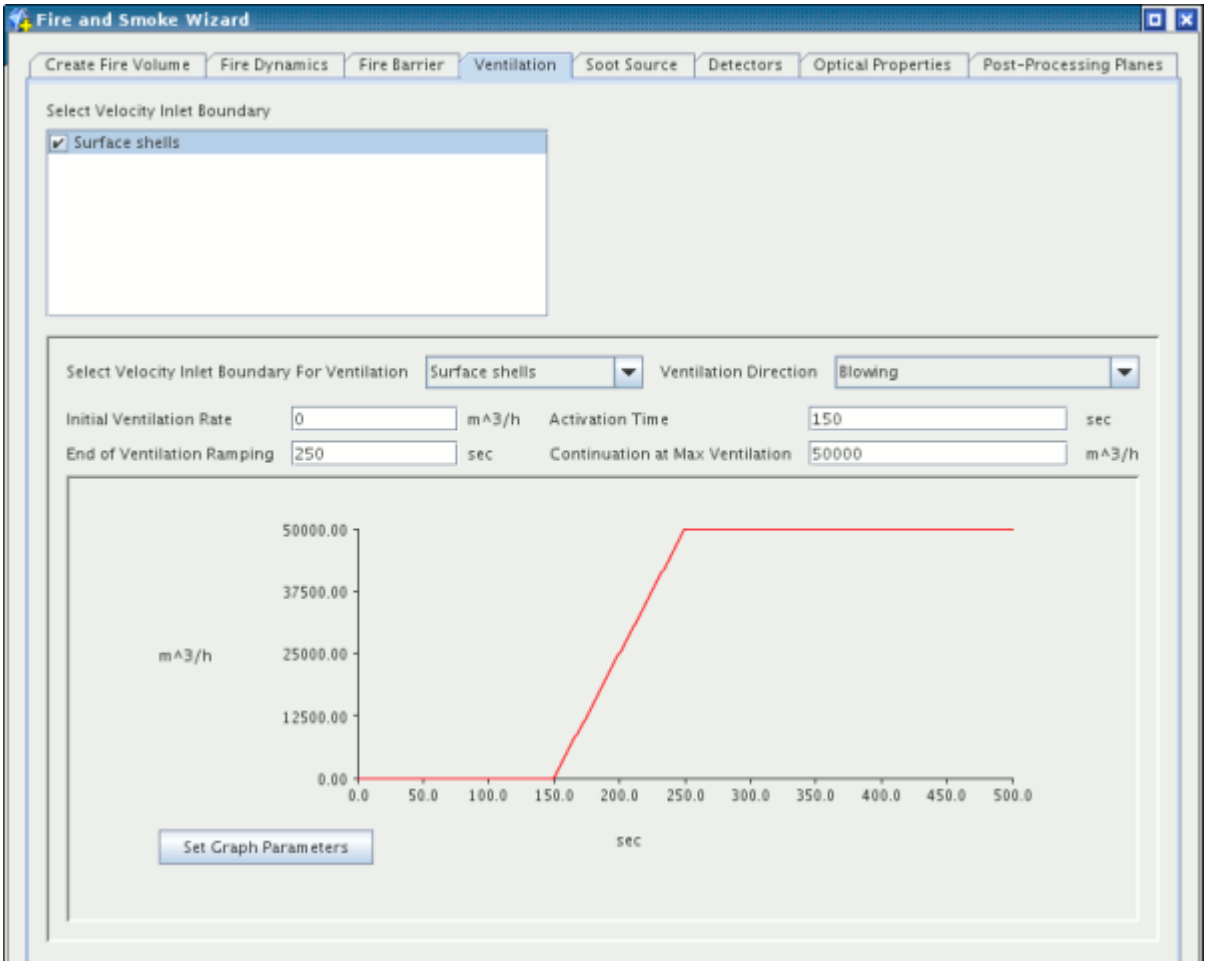
If a fire barrier, such as a fire door, is one of the regions created in your mesh, you can select it in this panel.



You can specify the activation and deactivation times of each fire barrier that you select in the drop-down list.

Setting Up Ventilation

You can select a velocity inlet boundary for ventilation, and then specify ventilation properties.



Begin by selecting a velocity inlet boundary from the list in the upper left.

If you are using multiple velocity inlet boundaries, use the drop-down list **Select Velocity Inlet Boundary For Ventilation** to select each one, and do as follows:

- Decide whether the ventilation direction is blowing or extraction, and select that option from the drop-down list.
- Enter values for initial ventilation rate, activation time, end-of-ventilation ramping, and continuation at maximum ventilation.
- Select another velocity inlet boundary from the list, if applicable, and repeat the above steps.

The fire and smoke wizard retains your entries for each of the velocity inlet boundaries.

This wizard also sets up a monitor plot of the heat release rate, which is previewed in the panel. To control the parameters of this plot, click the **Set Graph Parameters** button.

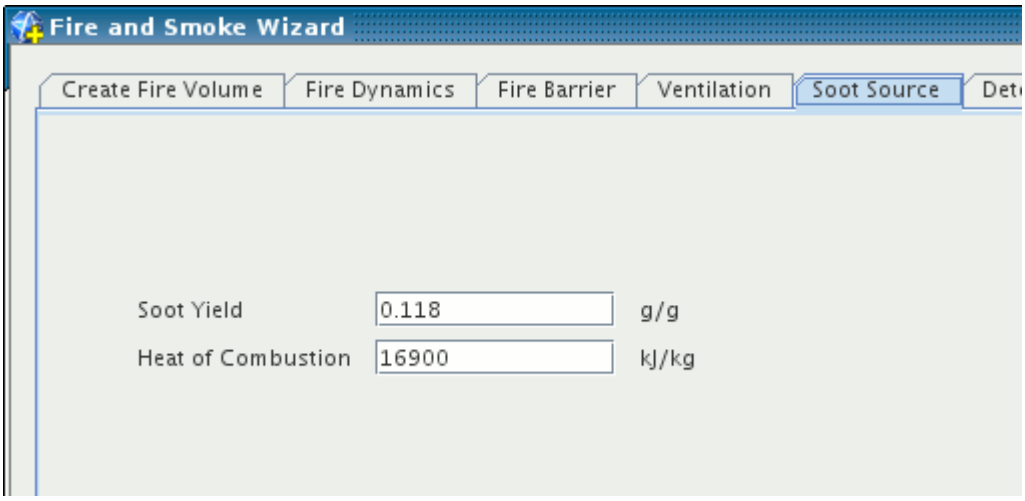
See Also:

Setting Graph Parameters

Preparing the Soot Source

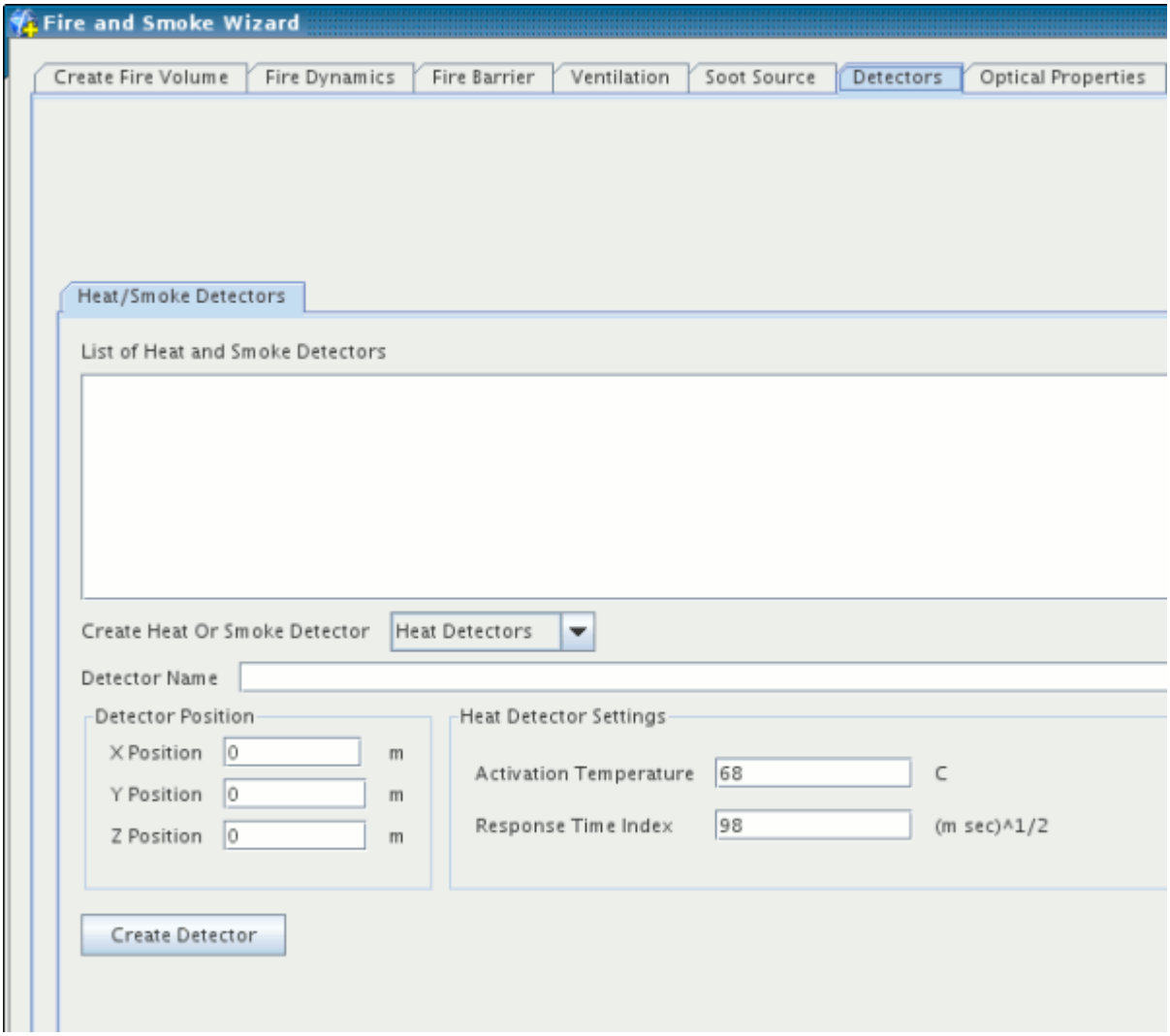
The Soot Source tab lets you specify the soot yield and heat of combustion.

Enter values, or accept the defaults, for the soot yield and heat of combustion.



Adding Heat and Smoke Detectors

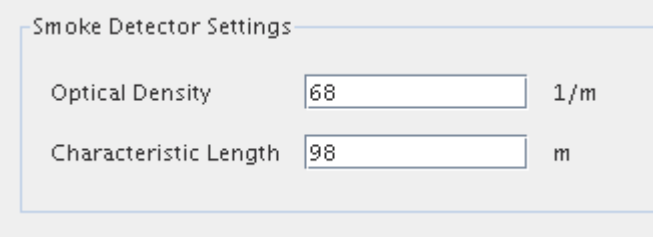
The *Detectors* tab lets you incorporate heat detectors or smoke detectors into your fire and smoke simulation.



To incorporate heat detectors or smoke detectors into your fire and smoke simulation:

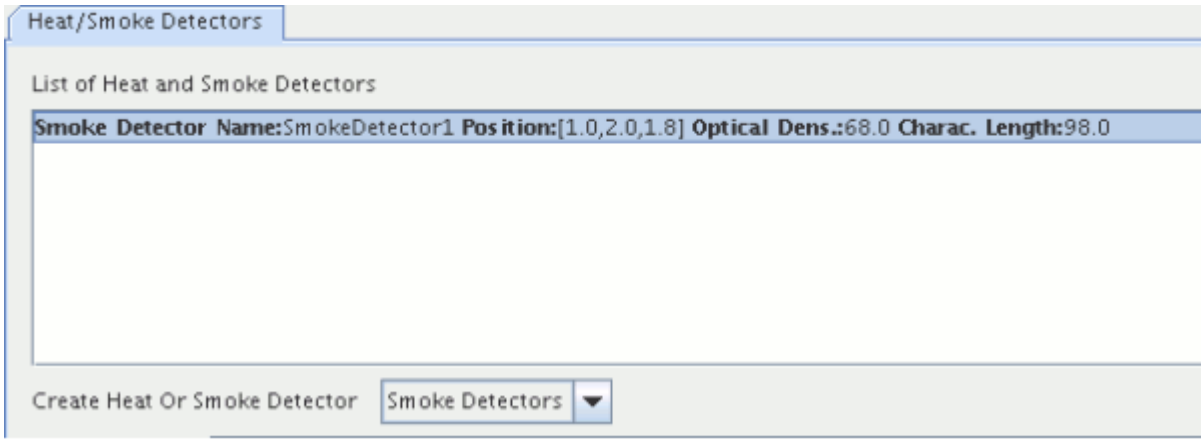
1. Begin by selecting the type of detector from the **Create Heat Or Smoke Detector** drop-down list.

   The settings for a particular type of detector change with your selection.
2. Enter a name for the detector in the **Detector Name** text box.
3. Enter the coordinates of the detector in the *Detector Position* group box.
4. Enter the settings that are specific to that type of detector:
   - For a heat detector, enter the activation temperature and response time index as shown in the preceding screenshot.
   - For a smoke detector, enter the optical density and characteristic length.

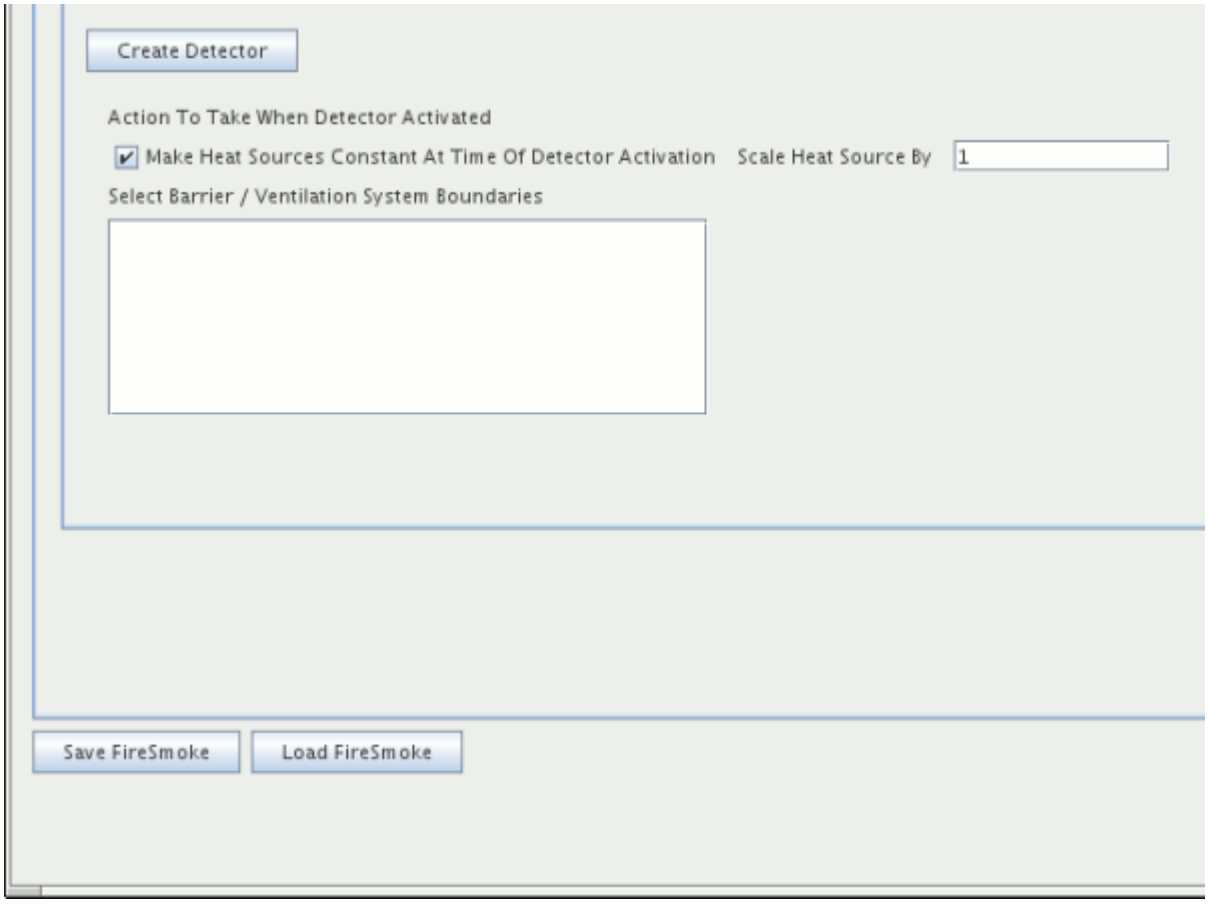


5. Once you finish entering values, you can create the detector by clicking the **Create Detector** button.

   The new detector appears with its pertinent data in the List of Heat and Smoke Detectors.

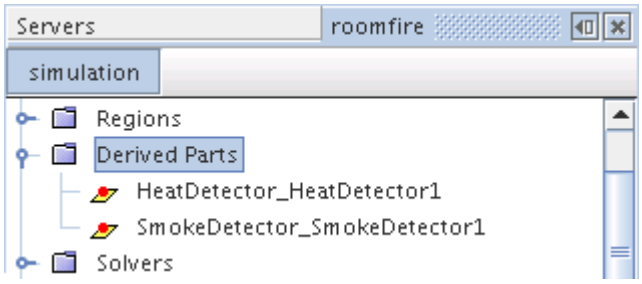


When the list contains one or more detectors, an additional set of options becomes available.

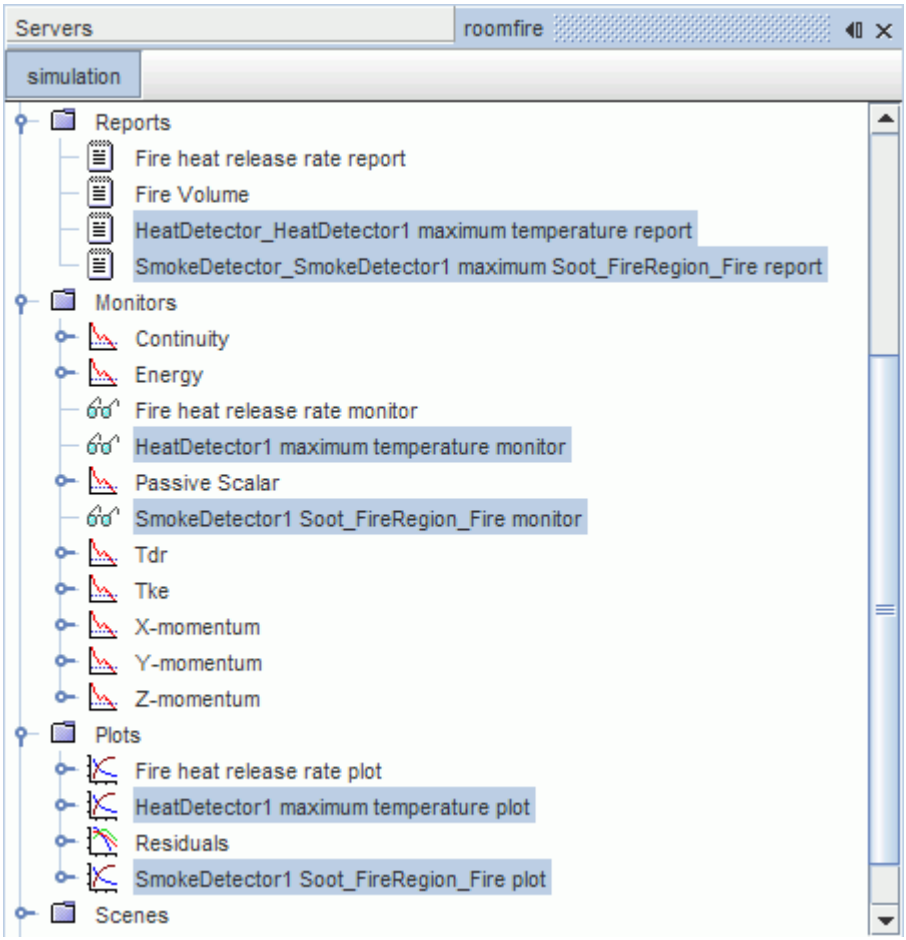


The settings are specific to each detector that you highlight in the detector list.

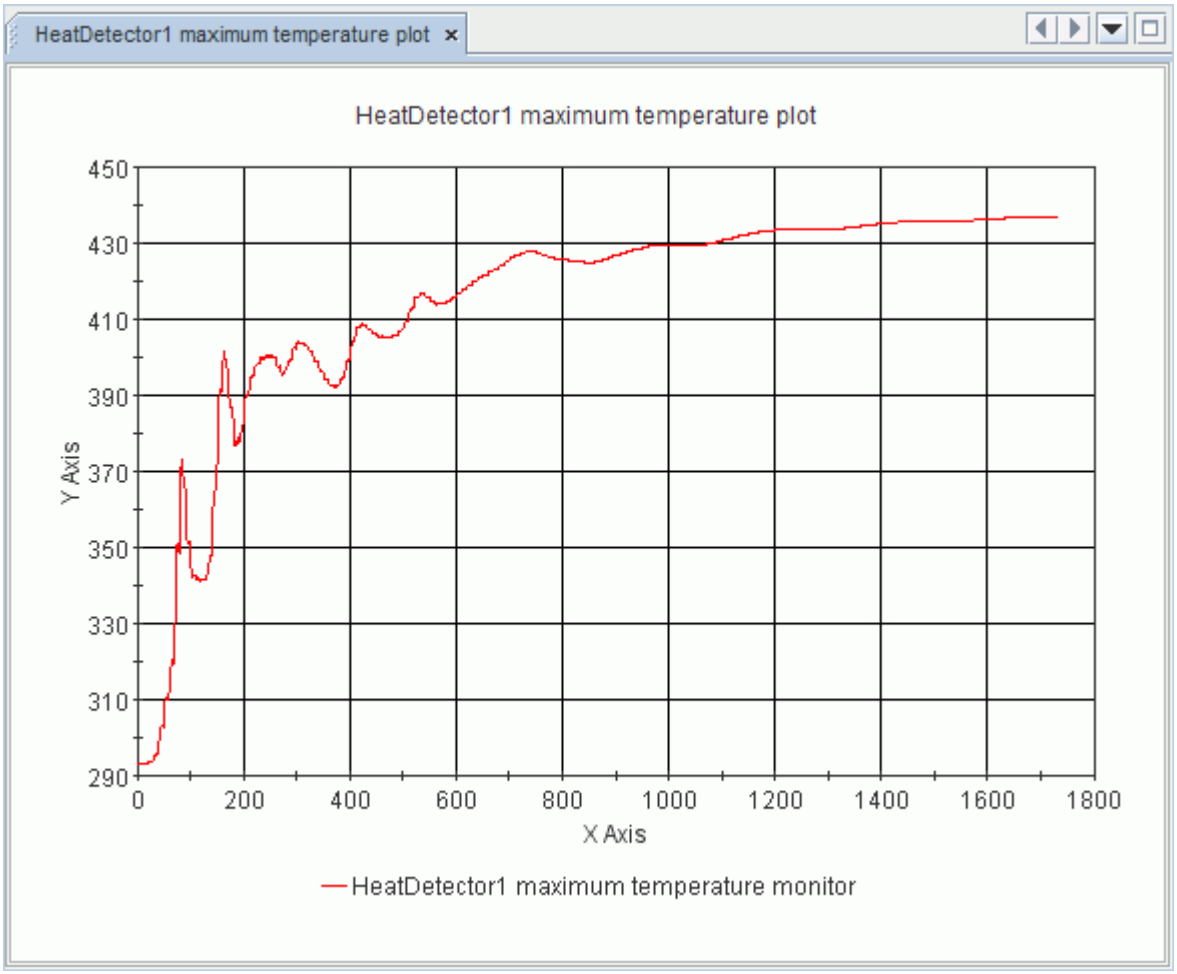
6. After you finish with the wizard by clicking **Apply**, a point derived part is created for each detector and appears in the object tree in the **Derived Parts** node.



Additionally, a report, monitor, and plot is created for each detector. These appear in the object tree in the **Reports**, **Monitors**, and **Plots** nodes respectively.



For heat detectors, the report monitors the maximum temperature at the probe location. The corresponding plot displays the temperature against iteration number.



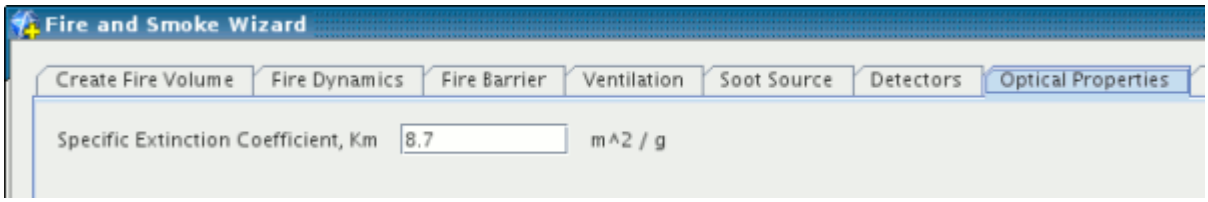
The report created for smoke detectors monitors the soot mass fraction at the probe location.

Specifying Optical Properties

The Optical Properties tab lets you set the specific extinction coefficient.

To specify optical properties:

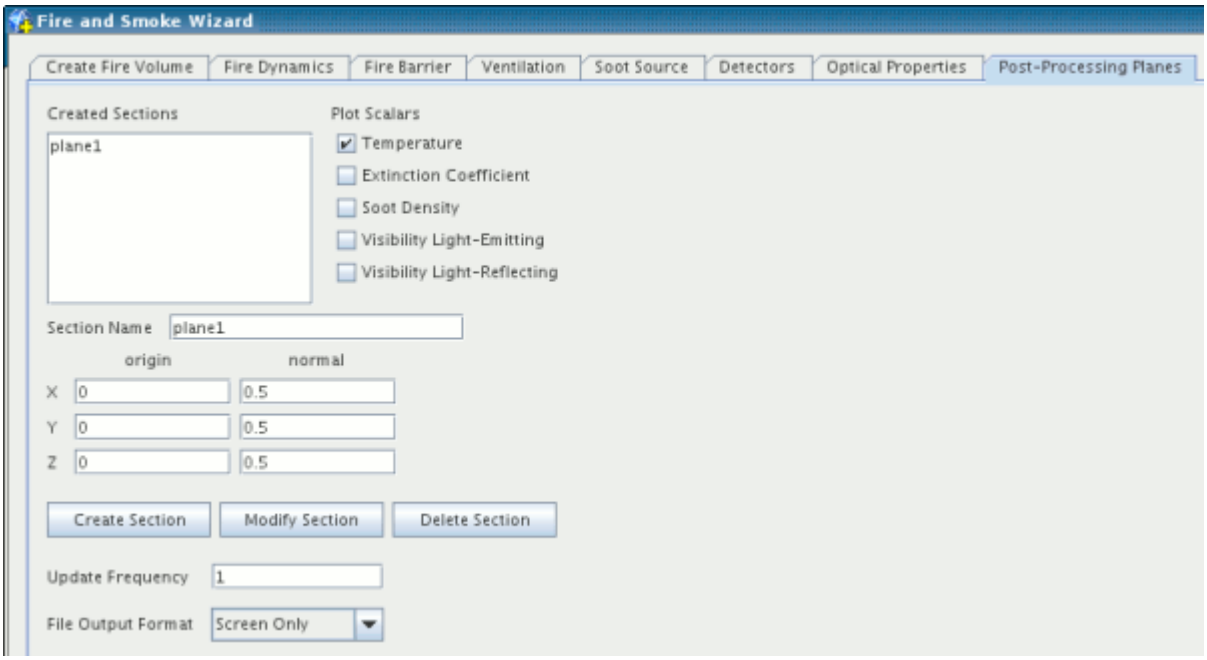
1. Enter a value, or accept the default, for the specific extinction coefficient.



Using Post Processing Planes

The Post-Processing Planes tab lets you create plane sections.

You can create plane sections with this tab of the wizard.



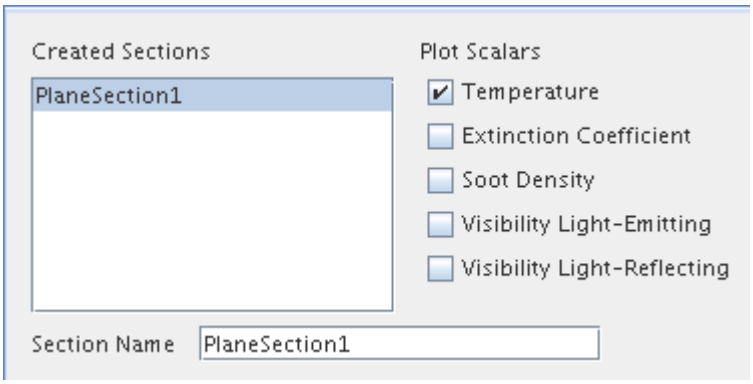
Begin by entering a name for the plane section in the **Section Name** text box.

Make your selection of scalar values to display by ticking the appropriate checkbox:

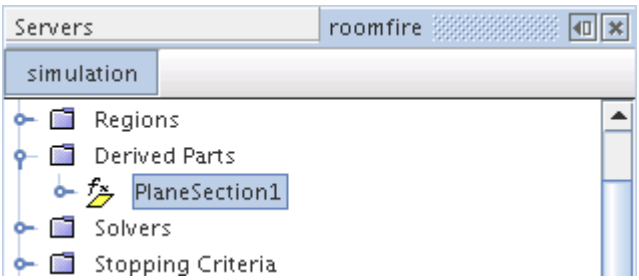
- **Temperature**
- **Extinction Coefficient**
- **Soot Density**
- **Visibility Light-Emitting**
- **Visibility Light-Reflecting**

Enter the appropriate coordinates for the origin and normal.

Once you have finished, you can create the plane section by clicking the **Create Section** button. The name of the section appears in the Created Sections list.



The plane section also appears in the object tree within the **Derived Parts** node.

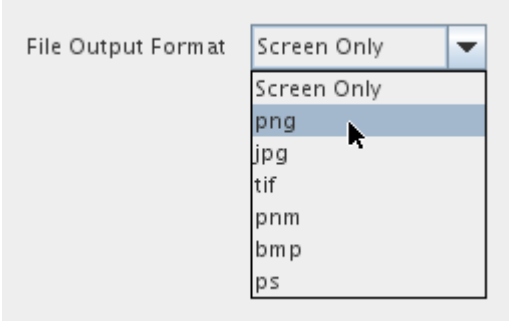


It is also possible to manage the plane sections in the list:

- To change a section, select its name in the list, make changes, and click **Modify Section**.
- To create another section similar to an existing one, select the name of the section in the list, make changes to it, give it a new name and click **Create Section**. An additional section appears in the list. If you did not name it, a number `2` is added to the original section's name by default.
- To delete a section, select it in the list and click **Delete Section**.

The **Update Frequency** setting lets you control how often the scene containing the data should be updated during the solver run. The default setting is `1`, or each iteration.

Finally, you can select a file format for output of the displayed data to an image file.



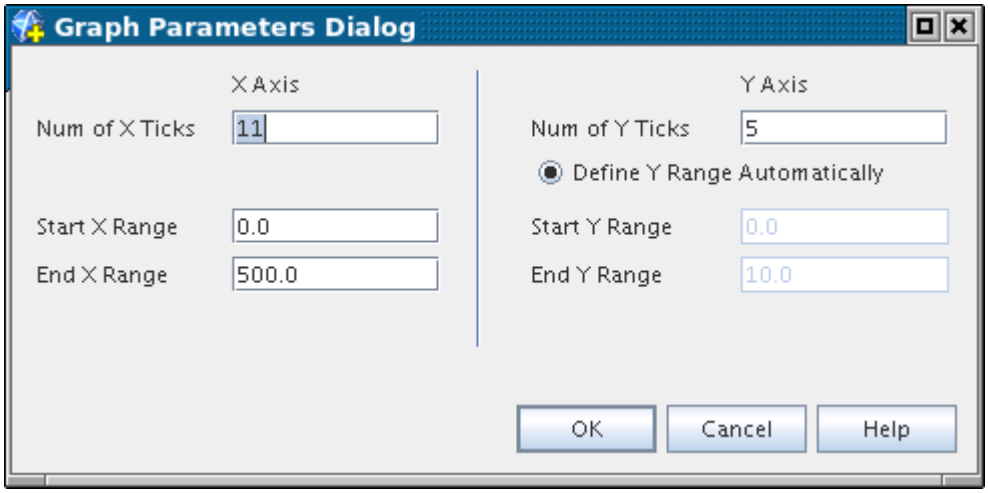
Setting Graph Parameters

This feature in the fire and smoke wizard lets you customize the monitor plots that it automatically generates.

To control the parameters of a plot:

1. Click the **Set Graph Parameters** button in the *Fire Dynamics* or *Ventilation* tabs of the wizard.

   This opens the *Graph Parameters* dialog.



2. Enter your specifications and click **OK**.

Using the Thermal Comfort Wizard

This section explains the thermal comfort feature of STAR-CCM+.

- What is the thermal comfort model?
- What is the Thermophysiological Occupant Model?
- What is the thermal comfort wizard?
- What are TCM Passengers?
- What are TCM Boundaries?
- What is the thermal comfort solver?
- Other capabilities of STAR-CCM+ that can be used with the thermal comfort wizard for problems of this type
- Setting up a thermal comfort case
- Performing a TCM simulation
- Post-processing a TCM simulation
- Information on working with TCM input files
- Information on working with TCM output files

Contents:

What is the Thermal Comfort Model?
What is the Thermophysiological Occupant Model?
What is the Thermal Comfort Wizard?
What are TCM Passengers?
What are TCM Boundaries?
What is the Thermal Comfort Solver?
STAR-CCM+ Capabilities Related to Thermal Comfort
Setting Up a Thermal Comfort Case
Performing a TCM Simulation
Post Processing a TCM Simulation
Working with TCM Input Files
Working with TCM Output Files
Bibliography

What is the Thermal Comfort Model?

The thermal comfort model enhances simulation of passenger comfort inside a vehicle (typically an automobile).

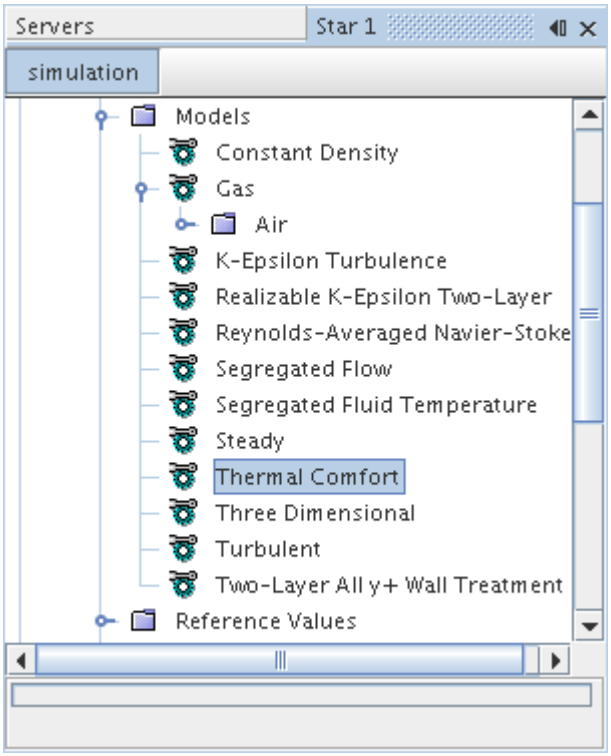
Its capabilities for this application include :

- Passenger characteristics—height, clothing resistance, and metabolic rate
- Relative humidity

- Insulating effects of the vehicle walls
- Radiation and convection for the dashboard and other surfaces exposed to the vehicle powertrain
- Air recirculation, which uses the air humidity to evaluate the energy loads on the air conditioning system. There are certain requirements and considerations that you should note when running air recirculation simulations.

The model provides these capabilities by allowing STAR-CCM+ to work with a special one-dimensional code called TIM (German acronym for “Thermophysiological Occupant Model”). TIM models and quantifies the thermophysiological state of a human passenger seated inside a car, or other enclosed space.

The thermal comfort model is represented with a node by that name among the physics continuum models. Its node is subject to common menu items, and appears when you activate the model.



Activating the thermal comfort model provides access to the thermal comfort wizard, which is used to define key simulation settings. Using the thermal comfort model also adds the following nodes to the object tree, which become placeholders for properties defined in the wizard:

- TCM Passengers
- TCM Boundaries
- Thermal Comfort Solver

Notes on Air Recirculation Simulations

The air recirculation feature requires the Multi-Component Gas model, with water vapor defined as the first component and dry air as the second component. You must select the components in this order, as this is how the Thermal Comfort wizard interprets the components.

The temperature and humidity of the recirculated air supplied to the air conditioning system is calculated with the following settings:

- *Percentage of recirculated air* from the Advanced Settings tab.
- *External Humidity* (relative or absolute) from the Advanced Settings tab.
- *External Bulk Temperature* from the External Convection and Radiation tab.

The amount of heat that needs to be removed from the air to match the temperature at the inlet is calculated as the energy load on the air conditioning system. If there are multiple inlets with different temperatures, the lowest temperature is used. In addition, if the temperature of the water vapor in the recirculated air is below the saturation temperature, the energy due to condensation is also calculated.

The total energy load on the air conditioning system is written to the `air_conditioner_load.dat` file after each iteration.

See Also:

Activating the Thermal Comfort Model
What is the Thermal Comfort Solver?
What is the Thermal Comfort Wizard?
What are TCM Passengers?
What are TCM Boundaries?
Adding External Convection and Radiation
Using Advanced Settings

What is the Thermophysiological Occupant Model?

The thermophysiological regulation of the human body can be seen as a regulatory circuit consisting of a passive and an active system.

The thermophysiological occupant model is a one-dimensional code that models the influence of ambient conditions in a passenger cabin on the thermal condition of an occupant. This model comprises the passive and the active system. The passive system describes the internal and external heat transfer mechanisms of the human body. The active system describes the thermophysiological regulation of the human body. The model accounts for the influence of clothing and contact areas--such as contact with the seats.

Due to the temperature difference, the human body transfers heat between the skin and the environment. Further, there are internal heat transfer processes resulting from the inhomogeneous temperature distribution within the human body. Under normal conditions, a human releases heat to the environment. In equilibrium, the heat release equals the heat production of the body.

Endotherms use thermophysiological regulation to keep their bodies in a state of thermal equilibrium, regardless of the surrounding environment.

Contents:

Passive System
Active System

Passive System

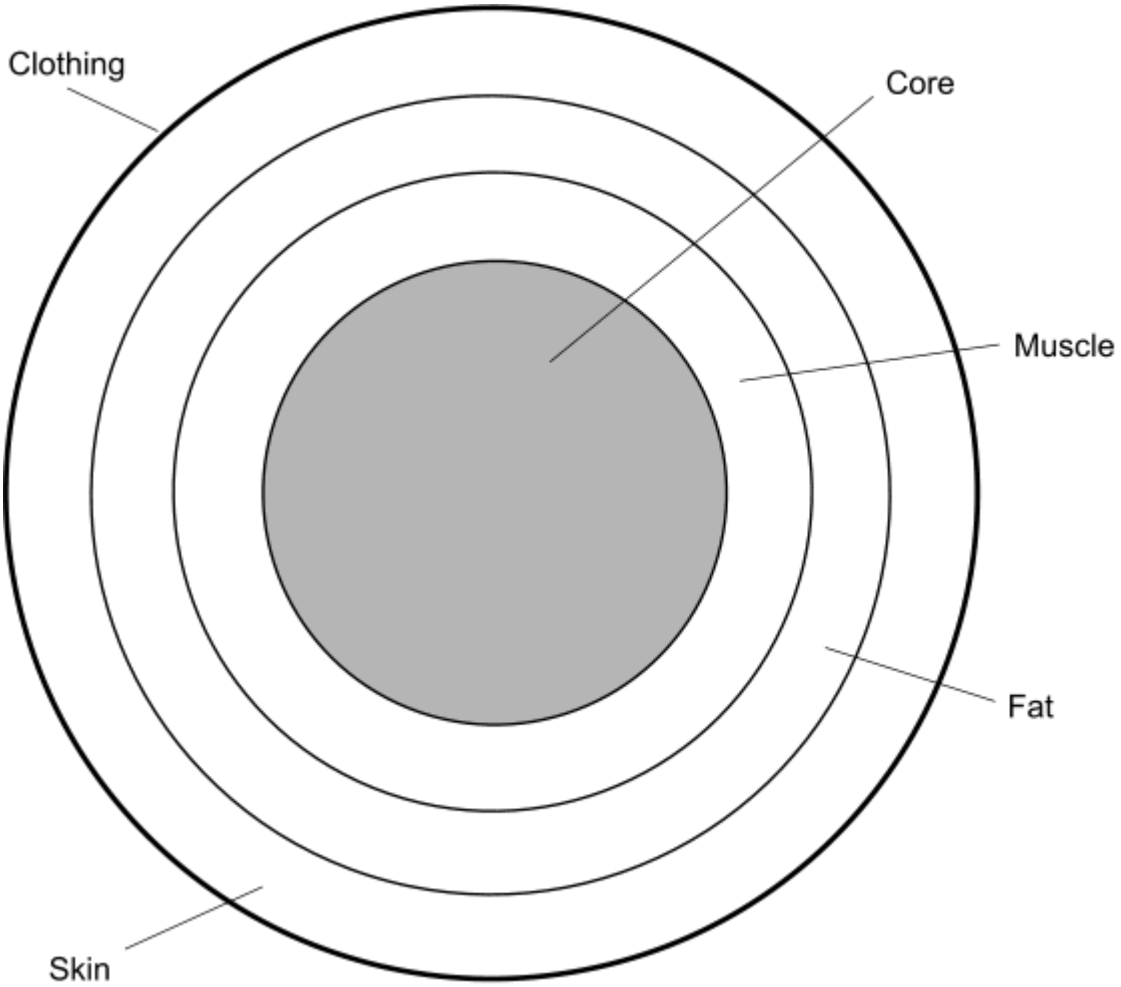
Within the thermophysiological occupant model, the passive system captures thermal effects at global and local levels.

In order to capture local effects, the human body is divided into segments and elements. This system has 14 segments and a central vascular system.

Each segment is composed of four concentric telescoping cylinders (the elements). The cylinders correspond to:

- Core

  This element consists to a large extent of skeleton.
- Muscle
- Fat
- Skin



The head is treated as a sphere with four shells. The central vascular system is a separate element. Through blood circulation, the central vascular system is responsible for the heat transport in the axial direction of the body.

The indices that are used in the formulation of this model are:

- j: Numbering of the segments (from 1 through 14)
- i: Numbering of the elements (from 1 through 4)

Contents:

Body Heat Storage
Metabolic Heat Production
Internal Heat Transfer
External Heat Transfer
Heat Balance
Clothing
Contact Surface

Body Heat Storage

This section describes the body heat storage model.

The human body stores heat in the different body parts. For a constant specific heat capacity c_p, the stored body heat flux $\dot{Q}_{ST}$ is directly proportional to the temperature gradient $\frac{\partial T}{\partial t}$:

$$\dot{Q}_{ST_{i,j}} = c_{p_{i,j}} \cdot \frac{\partial}{\partial t}\left(T_{i,j}\right) \tag{19}$$

The stored body heat flux of the central vascular system $\dot{Q}_{ST_{CVS}}$ is given by:

$$\dot{Q}_{ST_{CVS}} = c_{p_{CVS}} \cdot \frac{\partial}{\partial t}\left(T_{CVS}\right) \tag{20}$$

where c_p is the specific heat capacity.

The specific heat capacity of the body parts is calculated from the mass of the body elements and the specific heat capacities of the body elements:

$$c_{p_{1,j}} = m_{MUS_{1,j}} \cdot c_{p_{MUS}} + m_{BONE_{1,j}} \cdot c_{p_{BONE}} \tag{21}$$

$$c_{p_{2,j}} = m_{MUS_{2,j}} \cdot c_{p_{MUS}} \tag{22}$$

$$c_{p_{3,j}} = m_{3,j} \cdot c_{p_{FAT}} \tag{23}$$

$$c_{p_{4,j}} = m_{4,j} \cdot c_{p_{SKIN}} \tag{24}$$

The specific heat capacity of the different tissue types are ([2]; [3]):

Tissue Type	Specific Heat Capacity (kJ/kg K)
Bones	2093
Muscle	3768
Fat	2512
Skin	3768
Blood	3768

Metabolic Heat Production

This section describes the metabolic heat production model.

The human body produces heat due to biochemical reactions taking place in the body. This heat production is characterized by the base metabolic rate $\dot{Q}_{MET,BAS}$, which is always present to keep the human alive, and a portion of the metabolic rate due to activity. The activity portion consists of the muscular work $\dot{Q}_{MET,WORK}$ and, if applicable, the shivering due to cold $\dot{Q}_{MET,SHIV}$.

When computing the heat production through muscular work, a mechanical efficiency of zero is assumed. This assumption is valid for most types of physical activity. The core, fat, or skin elements do not influence the heat production, because the metabolic work $\dot{Q}_{MET,WORK,\,TOT}$ takes exclusively place within the muscles. The metabolic work has a local constant $L_{MUS,WORK}$ that is used to distribute the heat production to the individual muscle elements. The equations for computing the metabolic heat production are as follows:

Core:

$$\dot{Q}_{MET_{1,j}} = \dot{Q}_{MET,BAS_{1,j}} \tag{25}$$

Muscle:

$$\dot{Q}_{MET_{2,j}} = \dot{Q}_{MET,BAS_{2,j}} + \dot{Q}_{MET,SHIV_{2,j}} + \dot{Q}_{MET,WORK,TOT_{2,j}} + L_{MUS,WORK} \tag{26}$$

Fat:

$$\dot{Q}_{MET_{3,j}} = \dot{Q}_{MET,BAS_{3,j}} \tag{27}$$

Skin:

$$\dot{Q}_{MET_{4,j}} = \dot{Q}_{MET,BAS_{4,j}} \tag{28}$$

Internal Heat Transfer

This section describes the internal heat transfer model.

Generally, the heat flux is directed from inside the body to the peripheral regions. This heat transfer is called internal heat transfer. The border to the environment is the skin. The heat transfer is characterized by heat conduction and convection.

Heat conduction between two elements i and i+1 of a segment j is given by:

$$\dot{Q}_{COND_{i,j}} = R_{COND_{i,j}} \cdot \left(T_{i,j} - T_{i+1,j}\right) \tag{29}$$

The conductivities $R_{COND}(W/K)$ between the different shells are listed in the following table:

Segment	Core - Muscle	Muscle - Fat	Fat - Skin
Head	4.650	9.070	15.350
Torso	1.690	6.450	18.260
Upper arm	0.843	2.790	8.373
Forearm	0.843	2.790	8.373
Hand	2.835	5.580	5.580
Thigh	1.833	4.920	17.619
Lower leg	1.222	3.280	11.746
Foot	2.935	10.000	8.085

Heat is transferred between body parts through the transport of blood in a circulatory system. The blood extracts the heat from the warmer regions of the body and releases it to the colder regions.

Heat transfer from one element of a segment to the corresponding element in the next segment is calculated as:

$$\dot{Q}_{BLOOD_{i,j}} = \dot{V}_{BLOOD_{i,j}} \cdot c_{p,BLOOD} \cdot \rho_{BLOOD} \cdot \left(T_{BLOOD} - T_{i,j}\right) \tag{30}$$

The density of blood is assumed to be 1111 kg/m^3. The volumetric flow rate of blood varies according to the individual segments. Each body element has a base volumetric flow rate $\dot{V}_{BLOOD,BAS}\left(cm^3/s\right)$ that is listed in the following table:

Segment	Core	Muscle	Fat	Skin
Head	13.330	0.075	0.033	0.400
Torso	64.400	1.777	0.638	0.583
Upper arm	0.048	0.086	0.022	0.035
Forearm	0.048	0.086	0.022	0.035
Hand	0.014	0.007	0.007	0.277
Thigh	0.183	0.308	0.067	0.238
Lower leg	0.122	0.206	0.044	0.158
Foot	0.021	0.004	0.011	0.417

In addition to the constant base values, the blood volumetric flow rates of the skin and muscle shell can change due to physical activity or thermoregulation. The blood volumetric flow rates of the core and fat, however, always have the same base value.

External Heat Transfer

This section describes the external heat transfer model.

The human body exchanges heat with the environment through the following heat exchange mechanisms:

- Heat transfer due to convection and radiation
- Heat losses due to diffusion, transpiration, and respiration.

Each heat loss is consolidated to a single value for the global heat balance.

Convection

The heat flux of an undressed skin element of a segment j is computed as:

$$\dot{Q}_{CONV_j} = \alpha_{CONV_j} \cdot \left(T_{4,j} - T_{AIR_j}\right) \tag{31}$$

where:

$\dot{Q}_{CONV_j}$	Convective heat transfer coefficient of segment j
$T_{4,j}$	Skin temperature of segment j
T_{AIR_j}	Temperature of air

Radiation

Heat transfer due to radiation takes place between the skin surface or the clothes surface of the human body and the surrounding surfaces. By use of the Stefan-Boltzmann law, a radiation heat balance for each skin segment can be formulated as follows:

$$\dot{Q}_{RAD_j} = \epsilon_{SKIN_j} \cdot \sigma \cdot A_j \cdot \left(T_{4,j}^4 - T_{ENV_j}^4\right) \tag{32}$$

where:

ϵ_{SKIN_j}	Emissivity of skin of segment j
σ	Stefan-Boltzmann constant
A_j	Skin surface of segment j
T_{ENV_j}	Environment temperature of segment j

For temperature differences between the skin and the environment that are smaller than 20 K, the following linear approximation can be used:

$$\dot{Q}_{RAD_j} = \alpha_{RAD_j} \cdot A_j \cdot \left(T_{4,j} - T_{ENV_j}\right) \tag{33}$$

where α_{RAD_j} is a linearized heat transfer coefficient for radiation.

Using the equations for convection and radiation above, the two exterior heat fluxes are combined to an overall value — the so-called dry heat flux $\dot{Q}_{DRY}$. The heat transfer coefficients for convection and radiation are combined to a joint coefficient α_{DRY_j}:

$$\alpha_{DRY_j} = \alpha_{RAD_j} + \alpha_{CONV_j} \tag{34}$$

With this dry heat transfer coefficient, an effective temperature T_{EFF_j} is calculated for each skin element. The effective temperature is formed by using the temperature of the ambient air T_{AIR} and the temperature of the surrounding surfaces T_{ENV}:

$$T_{EFF_j} = \frac{\alpha_{CONV_j} \cdot T_{AIR_j} + \alpha_{RAD_j} T_{ENV_j}}{\alpha_{DRY_j}} \tag{35}$$

The equation for the heat loss due to convection and radiation can now be written as:

$$\dot{Q}_{DRY_j} = \alpha_{DRY_j} \cdot A_j \cdot \left(T_{4,j} - T_{EFF_j}\right) \tag{36}$$

With the above equation, all dry heat losses are described for the total heat balance.

Respiration

Heat loss due to respiration occurs because of the difference in humidity and temperature between incoming and outgoing air.

The heat loss due to the humidity difference is proportional to the water vapor pressure gradient between the lung and the ambient air. It also depends on the inhaled air volume and, as such, from the metabolic rate. The heat loss due to respiration is given by:

$$\dot{Q}_{RESP,HUM} = 1.752 \cdot 10^{-5} \cdot \dot{Q}_{MET} \cdot \left(p_{SAT,LUNG} - p_{AIR}\right) \tag{37}$$

The heat loss due to the temperature difference of the inhaled air is given by:

$$\dot{Q}_{RESP,TEMP} = 0.0014 \cdot \left(34 - T_{AIR}\right) \tag{38}$$

The total heat loss due to respiration is therefore:

$$\dot{Q}_{RESP} = \dot{Q}_{RESP,HUM} + \dot{Q}_{RESP,TEMP} \tag{39}$$

The total heat loss due to respiration is extracted one half each from the head core and the torso core.

$$\dot{Q}_{RESP_{1,1}} = \frac{1}{2}\dot{Q}_{RESP,HEAD} \tag{40}$$

$$\dot{Q}_{RESP_{1,2}} = \frac{1}{2}\dot{Q}_{RESP,TORSO} \tag{41}$$

Diffusion

Because of perspiratio insesibilis, the diffusion of water through the human skin, there is a permanent heat loss $\dot{q}_{DIFF}$ to the environment. According to [4], this heat loss can be specified as 5.6 W/m^2 for the entire body. As the heat loss is distributed to all skin elements, an equal rate of 5.6 W/m^2 can be assumed for all skin elements. The heat flux due to diffusion is thus given by:

$$\dot{Q}_{DIFF_j} = \dot{q}_{DIFF} \cdot A_j \tag{42}$$

where A_j is the area of a segment.

Transpiration

A further form of heat release is the evaporation of sweat on the surface of the skin during transpiration. During transpiration, water vapor transport takes place through the boundary layer. The driving force is the difference between the vapor pressure at the body surface and the vapor pressure of the ambient air. The heat flux due to transpiration is thus given by:

$$\dot{q}_{TRANS} = \beta \cdot \rho \cdot r \cdot \left(X_{SKIN} - X_{AIR}\right) \tag{43}$$

where:

X_{SKIN}	Moisture content of air at the skin surface
X_{AIR}	Moisture content of ambient air
β	Mass transport coefficient
r	Heat of evaporation of water

Taking into account the analogy between heat transfer and mass transport:

$$\beta = \frac{\alpha_{CONV}}{\rho \cdot c_p \cdot Le^{0.67}} \tag{44}$$

The Lewis number Le does not deviate much from a value of one for the considered range of application. Therefore, the following relation can be assumed for the surface heat flux due to transpiration evaporation:

$$\dot{q}_{TRANS} = \frac{\alpha_{CONV}}{c_p} \cdot r \cdot \left(X_{SKIN} - X_{AIR}\right) \tag{45}$$

Environmental conditions lead to a maximum evaporation rate. When the environment is saturated, sweat can no longer evaporate. Drops of sweat form on the skin that do not contribute to the heat loss. The maximum evaporation rate is given by:

$$Q_{EVAP,\max_j} = k \cdot \alpha_{CONV} \cdot A_j \cdot \left(p_{SAT,SKIN} - p_{AIR}\right) \tag{46}$$

where $k = 0.0165 K/Pa$ is the evaporation coefficient [5].

The heat losses due to diffusion and transpiration are summed as:

$$\dot{Q}_{EVAP_j} = \dot{Q}_{DIFF_j} + \dot{Q}_{TRANS_j} \tag{47}$$

Heat Balance

This section describes the Heat Balance model.

The general total heat balance for the human body is:

$$\dot{Q}_{MET} - \dot{Q}_{ST} - \dot{Q}_{DRY} - \dot{Q}_{EVAP} - \dot{Q}_{RESP} - W = 0 \tag{48}$$

The difference between $\dot{Q}_{MET}$ and W corresponds to the metabolic heat production. This heat balance can be established for all segments. The work W is equal to zero. Solving the above equation for the stored body heat flux $\dot{Q}_{ST}$ for the elements of a body segment and assuming that $T_{Cloth} > T_{Core} > T_{Muscle} > T_{Fat} > T_{Skin}$:

Core:

$$\dot{Q}_{ST_{1,j}} = \dot{Q}_{BLOOD_{1,j}} + \dot{Q}_{MET_{1,j}} - \dot{Q}_{COND_{1,j}} - \dot{Q}_{RESP_{1,j}} \tag{49}$$

Muscle:

$$\dot{Q}_{ST_{2,j}} = \dot{Q}_{BLOOD_{2,j}} + \dot{Q}_{MET_{2,j}} - \dot{Q}_{COND_{2,j}} + \dot{Q}_{COND_{1,j}} \tag{50}$$

Fat:

$$\dot{Q}_{ST_{3,j}} = \dot{Q}_{BLOOD_{3,j}} + \dot{Q}_{MET_{3,j}} - \dot{Q}_{COND_{3,j}} + \dot{Q}_{COND_{2,j}} \tag{51}$$

Skin:

$$\dot{Q}_{ST_{4,j}} = \dot{Q}_{BLOOD_{4,j}} + \dot{Q}_{MET_{4,j}} + \dot{Q}_{COND_{3,j}} - \dot{Q}_{DRY_j} - \dot{Q}_{EVAP_j} \tag{52}$$

In addition to these heat balance equations, the central cardiovascular system must be taken into account:

$$\dot{Q}_{ST,BLOOD} = \sum_{j=1}^{14}\left(\sum_{i=1}^{4}\left(-\dot{Q}_{BLOOD_{i,j}}\right)\right) \tag{53}$$

In each segment, a radial heat flux is taken into account through the term $\dot{Q}_{COND}$. All elements are coupled through the central cardiovascular system of the above equation. The central blood temperature is a function of all temperatures and blood volumetric flow rates in the body.

In total, there are 57 differential equations that must be solved. At the beginning of the calculation, the temperatures must be initialized. An equilibrium condition is reached when the element temperatures no longer change $\left(\frac{\partial T}{\partial t} = 0\right)$ and the stored body heat flux becomes zero.

Clothing

This section describes the clothing model.

Clothing is viewed as an additional layer on the skin establishing a direct contact between skin and clothing. The clothing only influences the dry and wet heat fluxes of the passive system. The clothing does not influence the active system.

The influence of clothing on the dry heat flux is taken into account through a thermal efficiency factor F_{TE_j}.

This factor quantifies the influence of the insulation through clothing in relation to the dry heat transfer coefficient:

$$F_{TE_j} = \frac{\alpha_{CLOTH_j} \cdot F_{SURF_j}}{\alpha_{CLOTH_j} + \alpha_{DRY_j} \cdot F_{SURF_j}} \tag{54}$$

The thermal efficiency factor ranges between 0 and 1, where 1 corresponds to a unclothed condition. The factor F_{SURF} takes the increase in size of the surface due to clothing into account. According to DIN ISO 7730, F_{SURF} for a skin element is computed as follows:

$$F_{SURF_j} = 1.00 + 1.29 \cdot I_{CLOTH} \tag{55}$$

$$F_{SURF_j} = 1.05 + 0.645 \cdot I_{CLOTH} \tag{56}$$

where I_{CLOTH} is the isolation factor clothing in m^2K/W.

Including F_{TE_j}, the dry heat flux becomes:

$$\dot{Q}_{DRY_j} = \alpha_{DRY_j} \cdot F_{TE_j} \cdot A_j \cdot \left(T_{4,j} - T_{eff_j}\right) \tag{57}$$

The temperature of the clothing surface T_{CLOTH_j} is:

$$T_{CLOTH_j} = T_{4,j} - \frac{\dot{Q}_{DRY_j}}{A_j \alpha_{CLOTH_j}} \tag{58}$$

A further effect of clothing is to restrain evaporation of moisture on human skin. To account for this effect, a factor F_{FD} is defined that reflects the water vapor permeability of clothing.

$$F_{FD_j} = \frac{1}{1 + 0.92 \cdot \left(\frac{\alpha_{CONV_j}}{\alpha_{CLOTH_j}}\right)} \tag{59}$$

$$Q_{EVAP,\max_j} = k \cdot \alpha_{CONV_j} \cdot F_{FD_j} \cdot A_j \cdot \left(p_{SAT,SKIN_j} - P_{AIR_j}\right) \tag{60}$$

Contact Surface

This section describes the Contact Surface model.

The passenger is permanently in contact with the vehicle seat. Therefore, it is necessary to divide the skin elements at the contact segments into two different surfaces. This leads to two skin temperatures for a segment:

- Skin temperature for the surface in contact with the seat
- Skin temperature for the free surface

The effective temperature at the contact surface is replaced by the temperature of the seat $T_{CONTACT}$. The dry heat flux is calculated only from heat conduction between the contact surface and the seat. Taking clothing into account, the thermal efficiency factor at the contact surface is given by:

$$F_{TE_j} = \frac{\alpha_{CLOTH_j} \cdot F_{SURF_j}}{\alpha_{CLOTH_j} + \alpha_{CONTACT} \cdot F_{SURF_j}} \tag{61}$$

The dry heat flux then becomes:

$$\dot{Q}_{DRY_j} = \alpha_{CONTACT} \cdot F_{TE_j} \cdot A_{CONTACT_j} \cdot \left(T_{4,j} - T_{CONTACT_j}\right) \tag{62}$$

The thermal efficiency factor F_{TE} is used to modify all base values. For all further calculations, such as regulation or heat losses, the respective temperatures of the skin elements are used and so two separate values result.

Active System

The thermophysiological regulation is a system whose functioning is still not fully understood today. A description or a model for the thermophysiological regulation applies the relations of cause and effect that are obtained through experimental results.

Such a model cannot describe the direct functions and processes of the central nervous system.

Generally speaking, the used thermophysiological regulation model is a cascaded proportional regulation. A central controller calculates four controlled variables for the entire body based upon the temperature signals. These controlled variables are passed on to all segments where they are processed in a local controller unit. The local controller provides blood flow rates and heat fluxes that influence the thermal condition of the body. This way, the influence of different body parts on the total thermal condition can be investigated.

The skin temperatures of all body segments and the head core temperature are the input quantities of the controller. The head core temperature corresponds in the model to the temperature of the hypothalamus, the center of temperature regulation. This temperature is particularly important for the thermophysiological regulation, because the controller always uses it as a reference. From these segment temperatures, the control deviations, θ, are calculated. The control deviation consists of the difference between the respective segment temperature and a target value that is called reference temperature T_{REF}. The reference temperature is known for each segment. This calculation is performed for the head core temperature and the skin temperatures.

$$\theta_{1,1} = T_{1,1} - T_{REF_{1,1}} \tag{63}$$

$$\theta_{4,j} = T_{4,j} - T_{REF_{4,j}} \tag{64}$$

These control deviations are the basis for the computation of the global controlled variables. For the skin temperatures, it is distinguished whether the control deviation is a warm or a cold signal. A positive deviation corresponds to a warm signal and a negative deviation corresponds to a cold signal:

$\theta_{WA_{4,j}} = \theta_{4,j}$ for $\theta_{4,j} \geq 0$

$\theta_{CO_{4,j}} = -\theta_{4,j}$ for $\theta_{4,j} < 0$

The element signals are combined to a total signal. When calculating this total signal, the distribution of the thermal receptors on the skin (cold and warm spots) are taken into account. The values are obtained through multiplication of the number of cold and warm spots of a skin element with the corresponding area fraction of the entire body surface [6]. The following table gives the distribution of receptors for each skin element:

Segment	Zref (cold signal)	Xref (warm signal)
Head	6.05e-02	0.201
Torso	0.4926	0.38
Upper arm	0.01714	0.02999
Forearm	0.01714	0.01625
Hand	0.09224	0.025
Thigh	0.04514	0.08999
Lower leg	3.01e-02	0.03275
Foot	1.67e-02	0.0155

The total signals of the skin elements for warm and cold impulses are written as:

$$\theta_{WA,TOT} = \sum_{j=1}^{14} \left(\theta_{WA_{4,j}} \cdot X_{REF_j} \right) \tag{65}$$

$$\theta_{CO,TOT} = \sum_{j=1}^{14} \left(\theta_{CO_{4,j}} \cdot Z_{REF_j} \right) \tag{66}$$

By using these values together with the control deviation of the head core temperature, the global controlled variables during heat load (vasodilation and transpiration) and during cold load (vasoconstriction and shivering) are calculated.

After calculating the global controlled variables, these values are converted into local controlled variables for each body part. The global conversion constants are labelled G and the local conversion constants are labelled L. The global controlled variables are denoted S. To characterize the type of controlled variable, the indices V_d for vasodilation, V_c for vasoconstriction, $Transp$ for Transpiration, and $Shiv$ for shivering are used. The local controlled variables are volumetric flow rate of blood, evaporation, and metabolic heat production.

Contents:

Target Value
Thermal Transpiration
Vasomotricity

Thermoregulatory Heat Production

Target Value

In the regulatory circuit, the target values correspond to the reference temperatures.

The target values have constant values and therefore, the command values are neglected. The head core reference temperature is 37.3 °C. The reference temperatures for each body element T_{REF} are given in the following table:

Segment	Core	Muscle	Fat	Skin
Head	37.3	36.7	36.343	36.116
Torso	37.336	36.965	35.593	34.736
Upper Arm	35.739	35.176	34.629	34.407
Forearm	35.739	35.176	34.629	34.407
Hand	35.684	35.627	35.592	35.543
Thigh	36.561	35.908	35.090	34.814
Lower leg	36.561	35.908	35.090	34.814
Foot	35.647	35.573	35.548	35.502

Under normal conditions, the body is at reference temperature. In such a case, the body has no regulatory activity.

Thermal Transpiration

This section describes the Thermal Transpiration model.

The global controlled variable, transpiration, is calculated as:

$$S_{TRANSP} = G_{TRANSP,KK} \cdot \theta_{1,1} + G_{TRANSP,HA}\left(\theta_{WA,TOT} - \theta_{CO,TOT}\right) \tag{67}$$

The basis for this global controlled variable and the used coefficients are experiments described in [7].

The following assumptions are made:

- Transpiration is proportional to the head core temperature at a constant average skin temperature.
- Transpiration is proportional to the average skin temperature at a constant head core temperature.

The global constants for calculating the transpiration signal have the following values:

$$G_{TRANSP,KK} = 372.2\,\frac{W}{K}$$

$$G_{TRANSP,HA} = 33.7\,\frac{W}{K}$$

The above equation means that the influence of the core head temperature is approximately 11 times higher than of the skin temperature.

The local effect of transpiration is accounted for by a factor that depends on the control deviation and consequently on the temperature of each skin element. The factor behaves in such a way that when a local control deviation of the skin temperature occurs, an n-fold amplification of transpiration takes place. The value of this amplification factor can vary in literature. For this model, the value from [4] is used, which is 2.

$$\dot{Q}_{TRANSP_j} = \left(L_{TRANSP} S_{TRANSP} \right) \cdot 2^{\frac{\theta_{4,j}}{10}} \tag{68}$$

The distribution of sweat formation across the skin surface is assumed to correspond to the distribution of perspiratory glands on the skin surface . The values of this local controlled variable are listed in the following table:

Segment	L_{TRANSP}
Head	0.081
Torso	0.481
Upper arm	0.039
Forearm	0.039
Hand	0.016
Thigh	0.064
Lower leg	0.043
Foot	0.018

The local transpiration rate is limited by the maximum possible evaporation. The evaporation also includes diffusion. This maximum evaporation rate tells how large the total evaporation heat loss of the body to the environment can be at maximum. This limitation results from the vapor pressure difference between the skin surface and the environment and additionally depends on the convective heat transfer coefficient.

If the computed value for the evaporation heat loss exceeds the sum of the constant diffusion heat flux $\dot{Q}_{DIFF}$ and the evaporation through transpiration $\dot{Q}_{TRANSP}$, the maximum possible value is taken. Therefore, the heat loss due to transpiration is reduced:

$$\dot{Q}_{TRANSP_j} = \dot{Q}_{EVAP,MAX_j} - \dot{Q}_{DIFF_j} \tag{69}$$

The ratio of effective transpiration to maximum possible transpiration yields the degree of moisture w_j , that is, the fraction of moistened skin to total skin surface.

$$w_j = \frac{\dot{Q}_{TRANSP_j} + \dot{Q}_{DIFF_j}}{\dot{Q}_{EVAP,MAX_j}} \tag{70}$$

Vasomotricity

Vasomotricity enables the human body to react to heat or cold loads. The global controlled variables are vasoconstriction in case of cold load and vasodilation in case of heat load.

These two global controlled variables are calculated separately by the controller and are used for the calculation of the local controlled variables of the elements:

$$S_{VD} = G_{VD,KK} \cdot \theta_{1,1} + G_{VD,HA}\left(\theta_{WA,TOT} - \theta_{CO,TOT}\right) \tag{71}$$

$$S_{VC} = -G_{VC}\left(\theta_{1,1} + \theta_{WA,TOT} - \theta_{CO,TOT}\right) \tag{72}$$

For the vasomotricity, the local controlled variable is given by the variation of the volumetric blood flux $\dot{V}_{BLOOD}$ that flows through the various skin parts. The local effect of the skin temperatures of the segments is taken into account when the volumetric blood fluxes in the skin parts are calculated.

A decrease in skin temperature of 10 °C leads to an increase of resistance in the blood vessels [4]. This effect is taken into account through the factor $2^{\left(\theta_{4,j}/6\right)}$. The volumetric blood flux through the skin parts is composed as follows:

$$\dot{V}_{BLOOD_{4,j}} = \left(\frac{\dot{V}_{BLOOD,BAS_{4,j}} + L_{VD_j} S_{VD}}{1 + L_{VC_j} S_{VC}}\right) \cdot 2^{\left(\frac{\theta_{4,j}}{6}\right)} \tag{73}$$

The above equation connects the local and the global controlled variables. The global controller constants are:

$$G_{VD,KK} = 32.5 \cdot 10^{-6} \frac{m^3}{sK}$$

$$G_{VD,HA} = 2.1 \cdot 10^{-6} \frac{m^3}{sK}$$

The vasoconstriction shows a high sensitivity with respect to the head core temperature and the skin temperature. For the calculation of the vasoconstriction controlled variable signal, a multiplicative approach is chosen. The controller constant is in this case:

$$G_{VC} = 5 \frac{1}{K}$$

The influence of vasodilation on the skin elements is investigated in [8]. Based on those results, the distribution for the vasodilation is chosen from the values that return the biggest dilation of the blood vessels.

The local constants for vasodilation and vasoconstriction are listed in the following table:

Segment	L_{VD}	L_{VC}
Head	0.132	5.0e-02
Torso	0.322	0.0
Upper arm	2.375e-02	1.25e-02
Forearm	2.375e-02	1.25e-02
Hand	6.049e-02	0.175
Thigh	6.9e-02	5.999e-02
Lower leg	4.6e-02	4.0e-2

Segment	L_{VD}	L_{VC}
Foot	5.0e-02	0.175

Thermoregulatory Heat Production

When the body temperature drops, the skeletal muscles are activated. This process is called shivering and serves to maintain the inner temperature.

The global controlled variable for shivering is given by:

$$S_{SHIV} = -G_{SHIV} \cdot \theta_{1,1} \cdot \theta_{CO,TOT} \tag{74}$$

Analogous to the other thermo-regulatory controlled variables, the global controlled signal for shivering is transferred to the local controller and is multiplied by a controller constant to obtain the local portion of the heat production:

$$Q_{MET,SHIV_{2,j}} = S_{SHIV} \cdot L_{MUSC,SHIV_j} \tag{75}$$

According to measurements in [9] and [10], there is a multiplicative connection between the controlled variable and the local controller constant. From experiments, the global controller constant is:

$$G_{SHIV} = 24.425 \frac{W}{K^2} \tag{76}$$

The distribution of the heat production in each muscle element $L_{MUSC,SHIV}$ is listed in the following table. Additionally, the distribution of heat production through muscular work in each muscle element $L_{MUSC,WORK}$ is listed:

Segment	$L_{MUSC,SHIV}$	$L_{MUSC,WORK}$
Head	2.0e-02	0.0
Torso	0.86	0.3
Upper arm	1.25e-02	2.0e-02
Forearm	1.25e-02	2.0e-02
Hand	0.0	5.0e-03
Thigh	2.1e-02	0.18
Lower leg	1.4e-02	0.12
Foot	0.0	5.0e-03

The volumetric flux of blood through the muscle elements is given by:

$$\dot{V}_{BLOOD_{2,j}} = \dot{V}_{BLOOD,BAS_{2,j}} + 0.2514 \cdot 10^{-6} \cdot S_{SHIV} L_{MUSC,SHIV} + 0.2514 \cdot 10^{-6} L_{MUSC,WORK} Q_{MET,WORK,TOT} \tag{77}$$

What is the Thermal Comfort Wizard?

The thermal comfort wizard is a tool that provides a clear and structured approach to setting up a thermal comfort simulation.

The wizard is used to define the key properties required by the TIM program, which are summarized below:

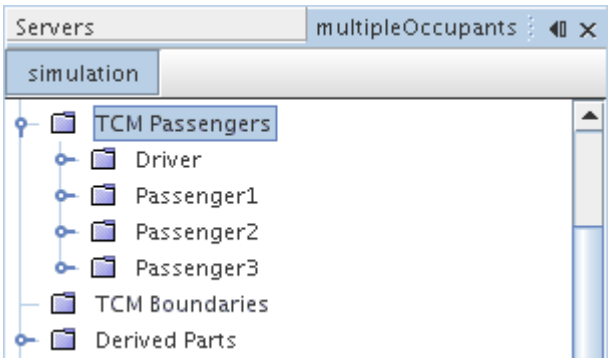
- General settings, which define the TCM passenger properties:
- Advanced settings, which define the solver properties:
- External convection and radiation, which define TCM boundary properties.
- Pre-processing, which creates scenes and reports specific to thermal comfort simulation.
- Post-processing, which creates plots and reports specific to thermal comfort simulation.

A guide to using the thermal comfort wizard is included in the Setting up a Thermal Comfort Case section.

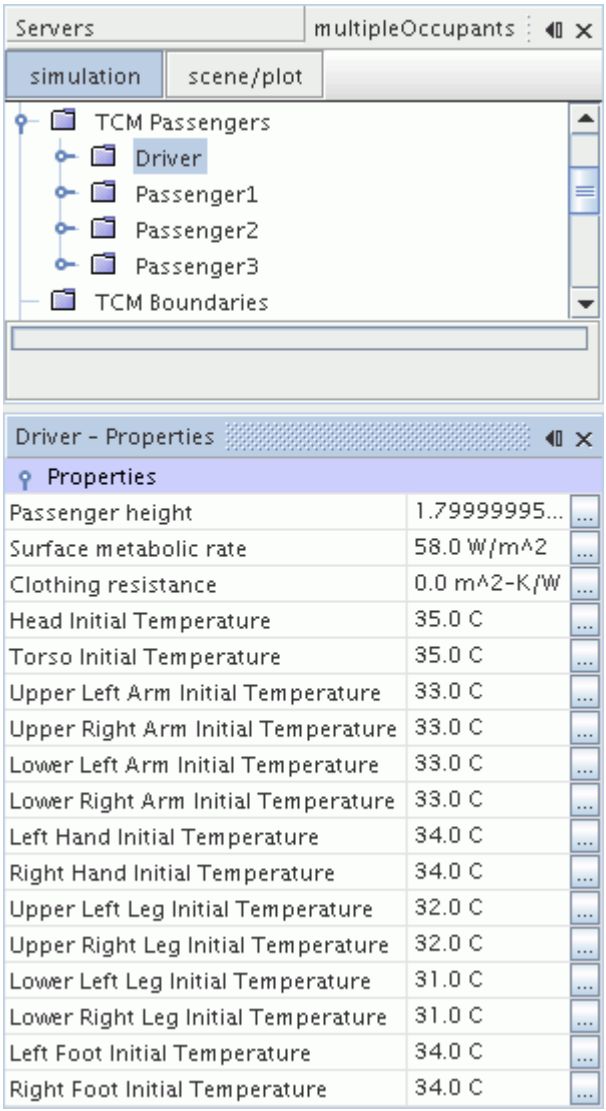
What are TCM Passengers?

Thermal comfort passengers are defined in the General Settings tab of the thermal comfort wizard.

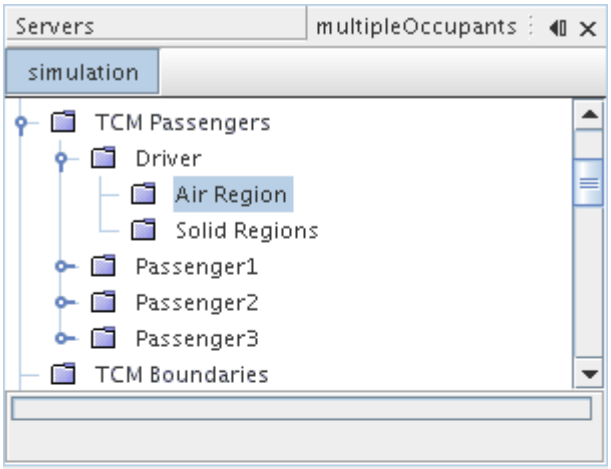
The TCM passengers are represented with a node by that name in the object tree. Its node is subject to common menu items, and appears when you activate the thermal comfort model.

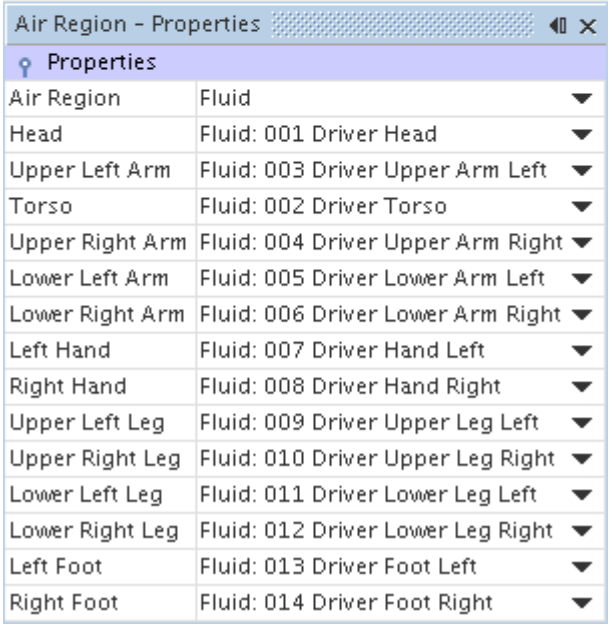


The **TCM Passengers** manager contains subnodes for each individual occupant in the vehicle, as defined by the wizard. These subnodes are placeholders for the passenger characteristics, which are displayed in the *Properties* window.



The managers for each occupant also contain subnodes, which hold information regarding the body segment boundaries, for both solid and air regions.





See Also:

Working with General Settings

What is the Thermal Comfort Wizard?

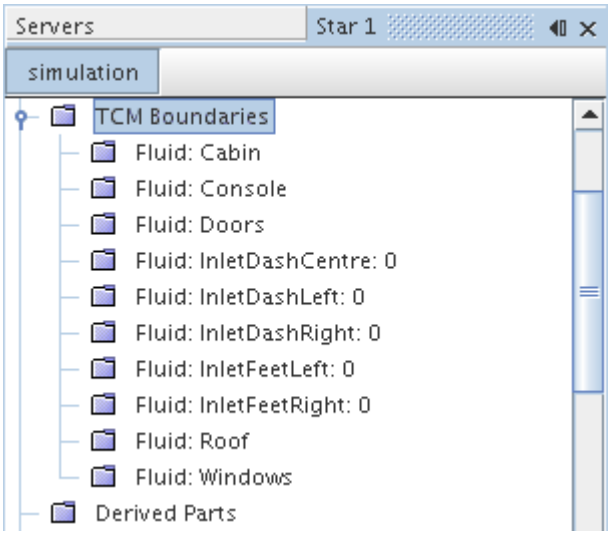
Activating the Thermal Comfort Model

What is the Thermal Comfort Model?

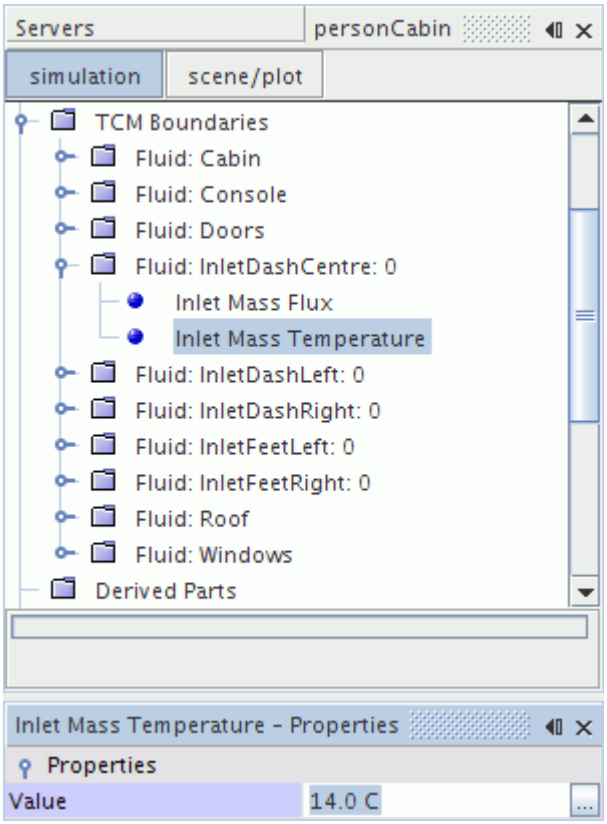
What are TCM Boundaries?

Thermal comfort boundaries are defined in the External Convection and Radiation tab of the thermal comfort wizard.

The TCM boundaries are represented with a node by that name in the object tree. The node appears when you activate the thermal comfort model.



Each TCM boundary has a subnode(s) that represents the boundary conditions. Selecting this node displays the boundary value in the *Properties* window.



See Also:

Adding External Convection and Radiation

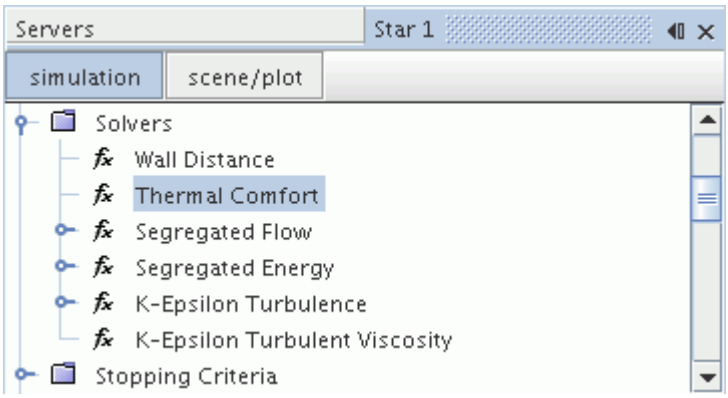
What is the Thermal Comfort Wizard?

Activating the Thermal Comfort Model
What is the Thermal Comfort Model?

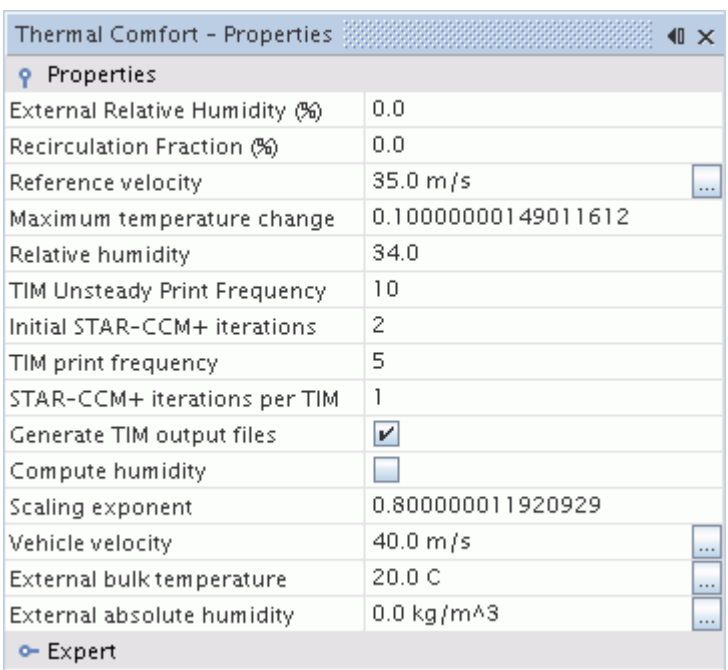
What is the Thermal Comfort Solver?

Thermal comfort solver settings are defined in the Advanced Settings tab of the thermal comfort wizard.

The thermal comfort solver is represented with a node by that name in the object tree. Its node is subject to common menu items, and appears when you activate the thermal comfort model.



The **Solver** > **Thermal Comfort** node acts as a placeholder for solver properties specific to the TIM program, which are displayed in the *Properties* window.



See Also:

Using Advanced Settings
What is the Thermal Comfort Wizard?
Activating the Thermal Comfort Model
What is the Thermal Comfort Model?

STAR-CCM+ Capabilities Related to Thermal Comfort

This section outlines the STAR-CCM+ features that are related to thermal comfort modeling.

The thermal comfort wizard is the primary tool for these types of problems, but there are several other STAR-CCM+ features that can help with the CFD solution:

- Modeling Options:
  - Stationary
  - Three-Dimensional
  - Solid Energy
- Radiation
- Turbulence Models:
  - K-Epsilon, K-Omega...
- Boundary Conditions:
  - Mass Flow Inlet
  - Pressure
- Interfaces:
  - Internal
  - Contact
- Solvers:
  - Second-Order
  - Segregated
  - Algebraic Multigrid

Setting Up a Thermal Comfort Case

This section outlines the process of setting up a passenger comfort case with the thermal comfort wizard.

The first step of a thermal comfort analysis is to define the simulation domain. In some scenarios, the symmetry of the respective vehicle allows you to reduce the size of the domain and the number of passengers to be modeled.

Commercial aircraft, for example, accommodate large numbers of passengers. To make the simulation efficient, take advantage of the symmetry and periodicity of the seating layout, the windows, the cooling ducts, and so on, and model just a section of the whole domain. For passenger vehicles, it seems that there is symmetry for the passengers, however the ducting and other geometric constraints of the passenger cabin are rarely symmetric. For passenger vehicles, you typically model all passengers of interest.

You use the wizard after having generated a mesh, selected models, and set up initial conditions and boundary conditions. The process involves the following:

- Activating the thermal comfort model
- Activating the thermal comfort wizard
- Working with general settings
- Using advanced settings
- Adding external convection and radiation
- Pre-processing
- Saving and reloading TCM settings

Contents:

Activating the Thermal Comfort Model
Activating the Thermal Comfort Wizard
Working with General Settings
Using Advanced Settings
Adding External Convection and Radiation
Pre Processing
Saving and Reloading TCM Settings

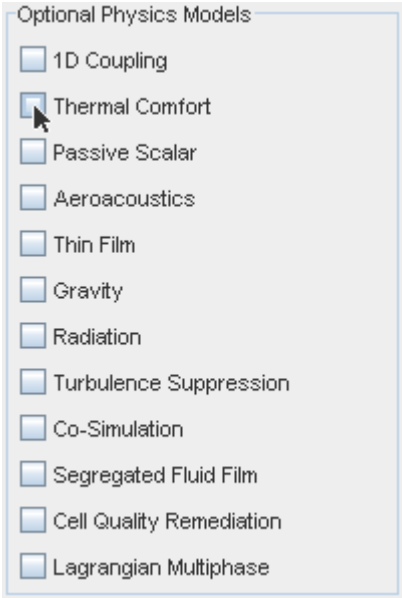
Activating the Thermal Comfort Model

The Thermal Comfort model is selected from within the *Physics Model Selection* dialog on the relevant continuum.

To select the thermal comfort model, the following models are required:

- Any model from the *Material* group, depending on simulation requirements.
  - However, if you wish to model air recirculation, you must also select Multi-Component Gas from the *Material* group box to model the water vapor and dry air gas components.
- Either the Segregated or Coupled solver.
- An energy model from the *Optional Physics Models* group, which may be one of the following:
  - Segregated solver:
    - Segregated Fluid Temperature
    - Segregated Fluid Isothermal
    - Segregated Fluid Enthalpy
    - Segregated Solid Energy
  - Coupled solver:
    - Coupled Energy
    - Coupled Solid Energy

The Thermal Comfort model is then available in the *Optional Physics Models* group.



When the Thermal Comfort model is activated, the TCM Passenger and TCM Boundaries managers become available in the object tree.

See Also:

What are TCM Passengers?

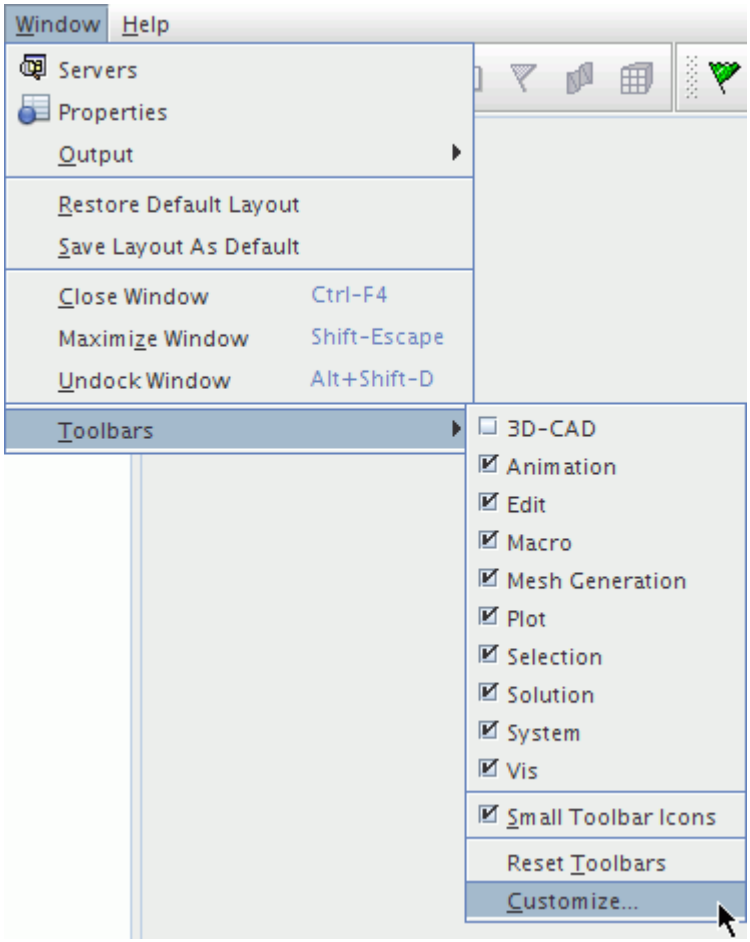
What are TCM Boundaries?

Activating the Thermal Comfort Wizard

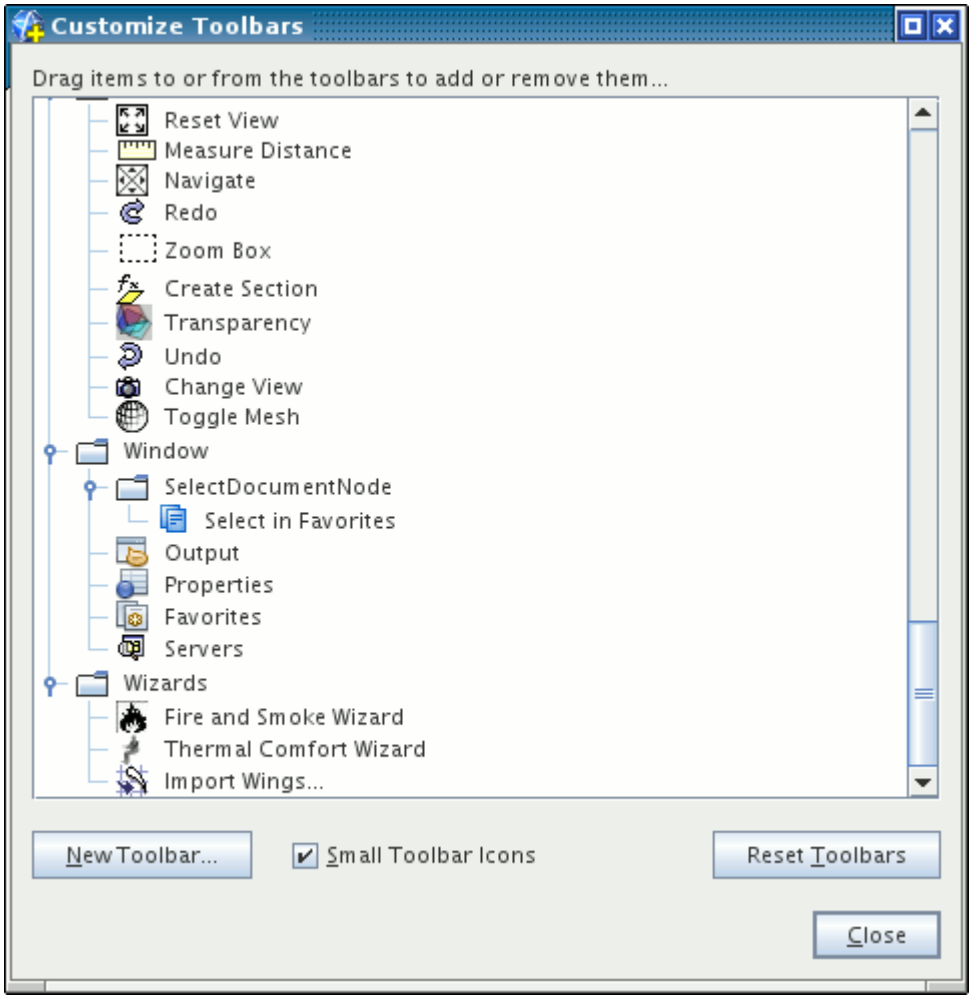
The properties are set using the Thermal Comfort Wizard.

To begin accessing this feature:

1. From the menu bar, select **Window** > **Toolbars** > **Customize**.

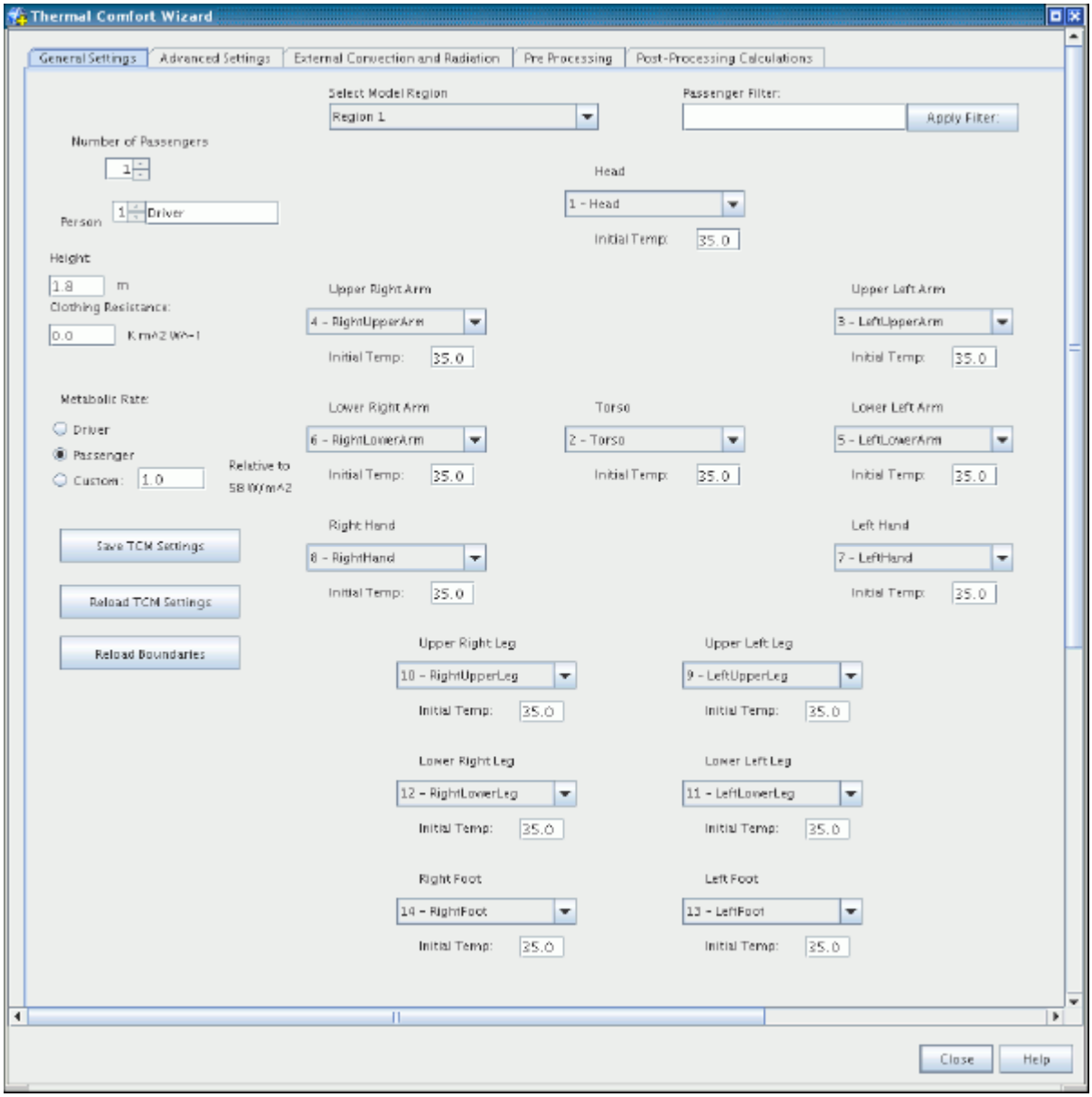


This opens the *Customize Toolbars* dialog as shown in the following screenshot.



2. Scroll down to the **Thermal Comfort Wizard** icon.
3. Drag and drop this icon onto the end of one of the toolbars near the top of the workspace.
4. Click **Close** on the *Customize Toolbars* dialog.
5. Click the (**Thermal Comfort Wizard**) button.

   This opens the *Thermal Comfort Wizard* dialog.



The thermal comfort wizard assumes that the model has been built, and boundaries defined. Initial temperature values for the passenger body segments must also be specified before using the wizard. Data is then entered into the wizard to correctly define the characteristics of the thermal comfort simulation. Information entered in the *Thermal Comfort Wizard* dialog is retained automatically, so you can move between each of the tabs and even close and reopen the wizard without losing any information. The data is only applied to the simulation and added to the relevant managers in the object tree when the Save TCM Settings button is clicked. It is also possible to reload information into the wizard, to match the current TCM settings in the STAR-CCM+ simulation.

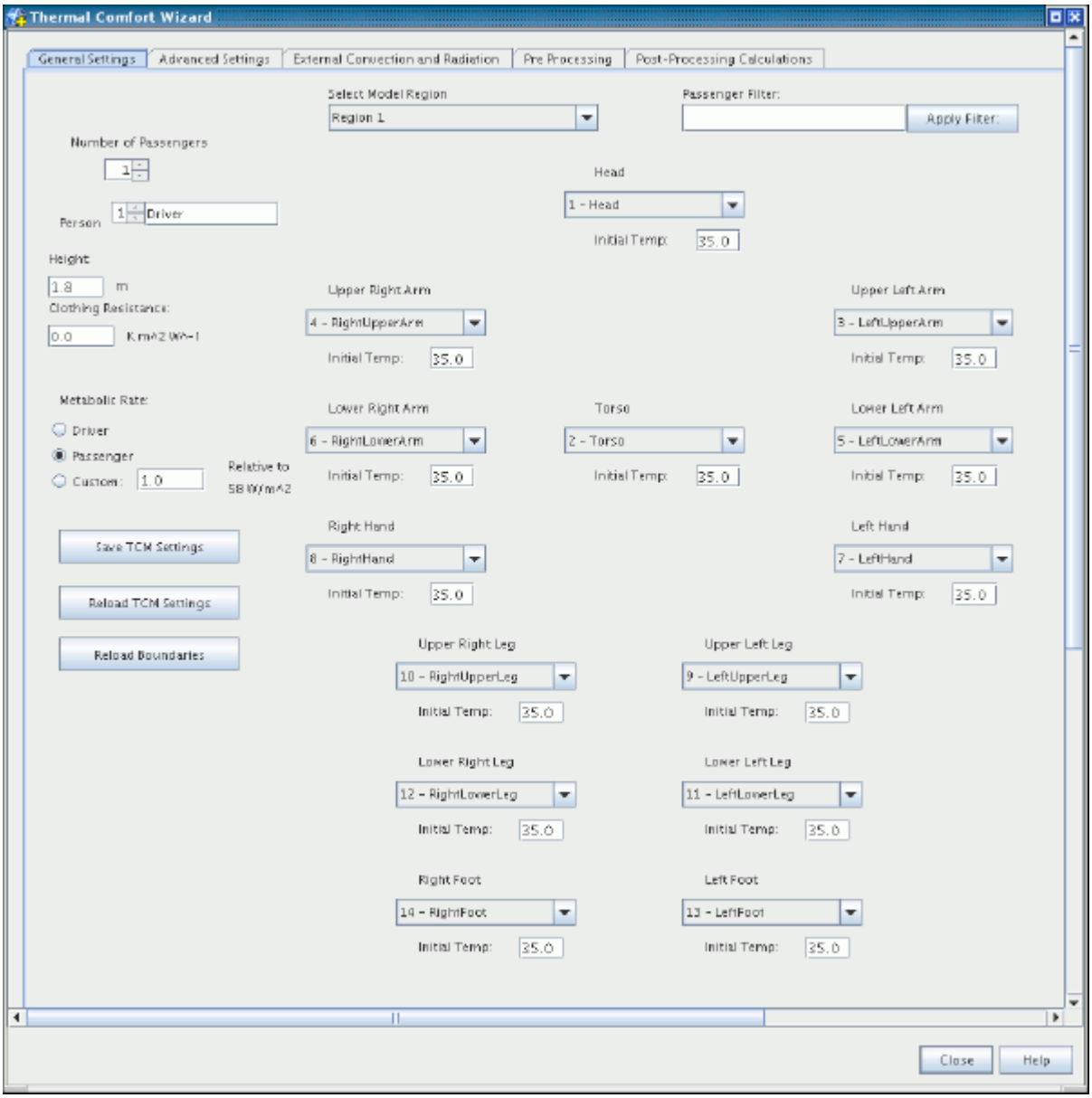
The dialog includes a **Close** button that closes the wizard without writing to the file. Data entered in the wizard are cleared only if the STAR-CCM+ client is shut down.

See Also:

Saving and Reloading TCM Settings

Working with General Settings

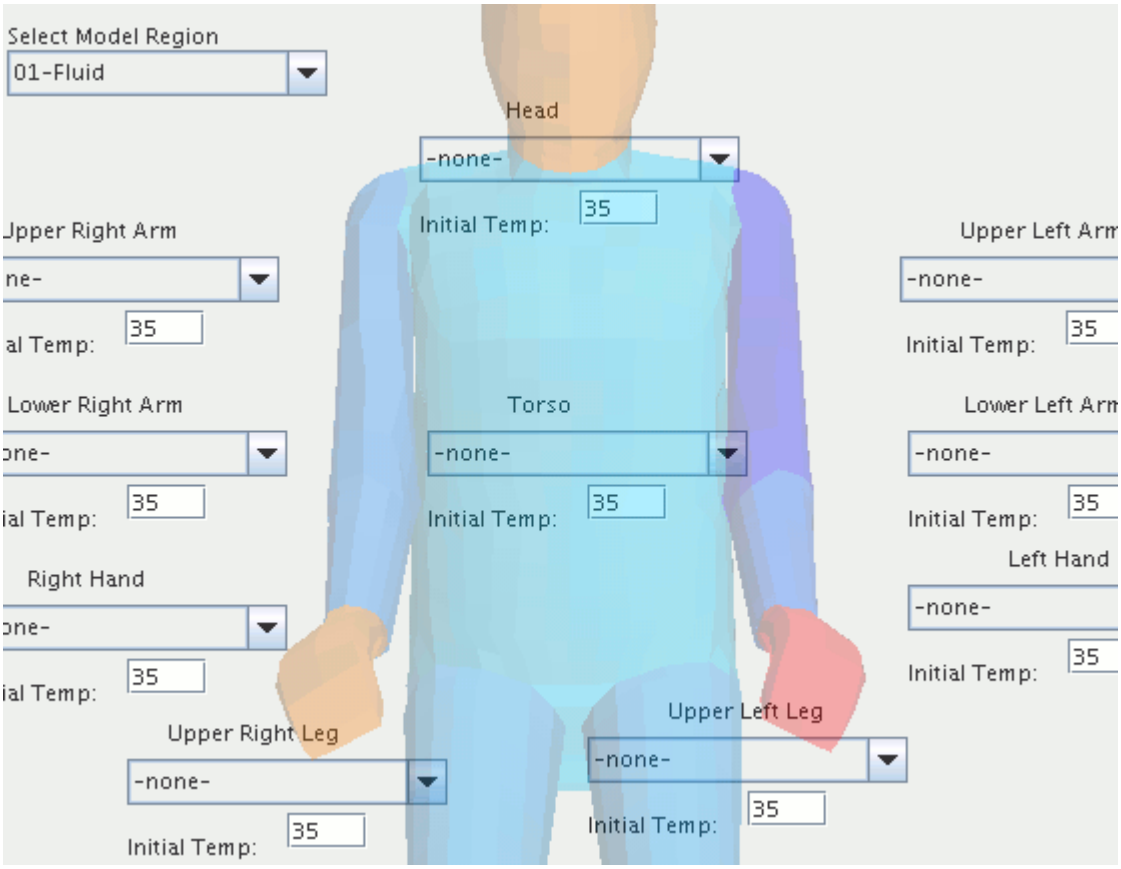
The *General Settings* tab allows you to specify properties for regions and boundaries.



Conventions for Preparing Body Segment Boundaries

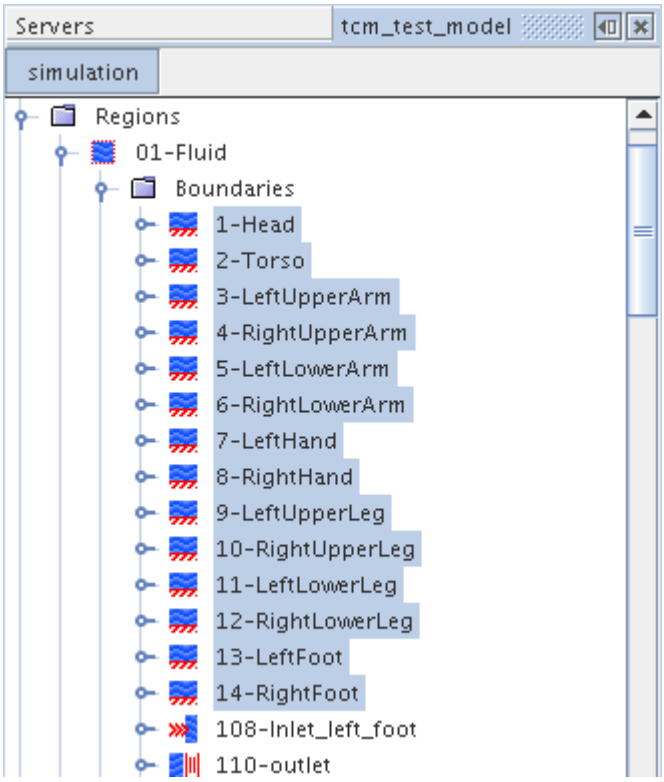
Each passenger should be composed of fourteen body segments, which must be defined as separate boundaries. These boundaries correspond to the fourteen drop-down lists in the General Settings tab.

Provided below is a diagram illustrating how the body segment menus have been arranged to approximate a human body.



In cases where the passenger touches solid regions, such as the torso and legs touching a seat, there must be a separate boundary for each solid segment as well.

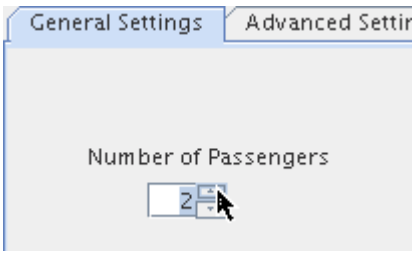
It is important to note that boundary labels need to include the names of their respective body segments, as shown in the following screenshot. The thermal comfort wizard recognizes only the boundaries that are named in accordance with this convention.



If there are multiple people in the simulation, the boundary names should reflect this, for example `Passenger2 LeftUpperArm`.

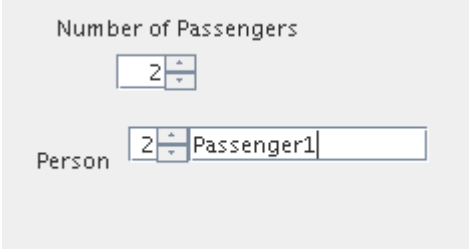
Entering Passenger Properties

Begin using the wizard by selecting a value in the Number of Passengers spin box. This value should match the total number of people represented in the STAR-CCM+ simulation.



Based on the number you specify, the Person setting contains a numbered list representing passengers. Select a passenger by the number, and complete your settings for that passenger throughout the tab, as described in detail in the rest of this section. Once all the settings have been provided for that passenger, you can change the Person number and enter the next set of data. The thermal comfort wizard retains your entries for each passenger.

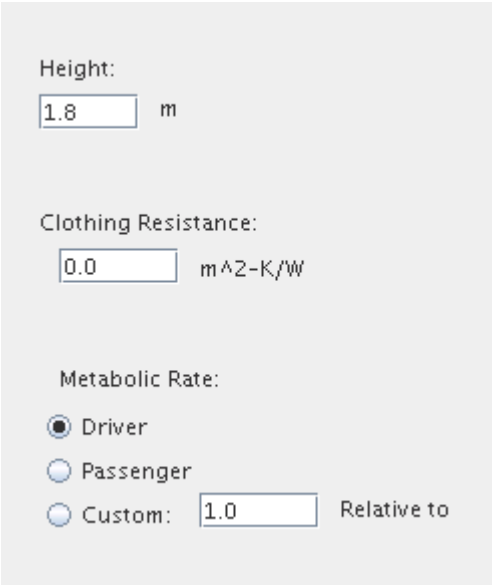
For each passenger, begin by assigning the passenger a generic name, such as `Passenger1`. By default the first passenger has the name `Driver`, which can be changed.



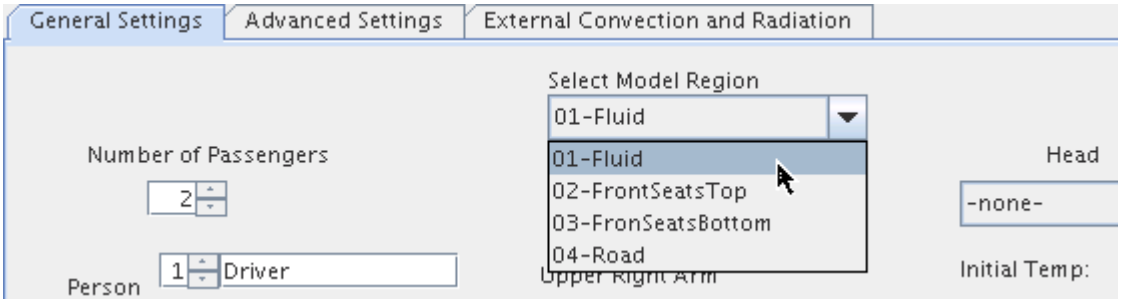
Specify height, clothing resistance, and metabolic rate.

- The passenger **Height** is used by the TIM program to calculate the body mass distributions for the vascular system and the body surface area values.
- The **Clothing Resistance** defines the thermal conductivity of the passenger's clothing in m^2K/W.
- The Metabolic Rate is the rate at which the person releases energy to the surroundings as heat, which depends on their level of activity.

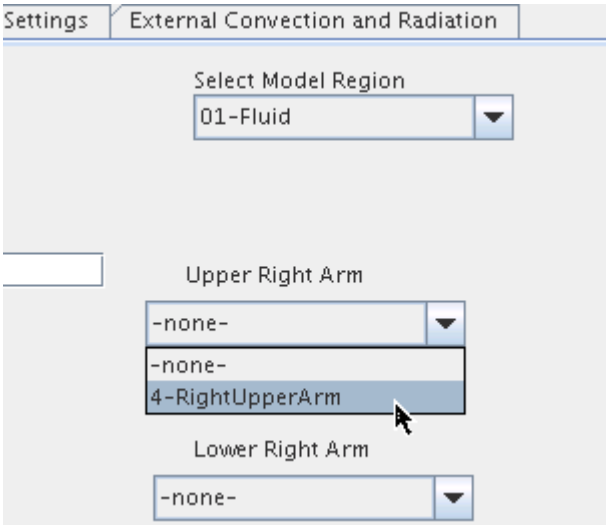
  It is set by selecting one of the **Driver** or **Passenger** radio buttons, or by entering a **Custom** value. When the **Passenger** option is selected the metabolic rate is set to `58 W/m^2`, which is equivalent to a person who is seated and relaxed. When the **Driver** option is chosen the metabolic rate is set to `75.4 W/m^2` (passenger value `x1.3`), indicating that the driver is more active. The **Custom** option allows you to specify an alternative value by entering a scaling factor relative to 58 W/m^2, and is usually in the range of `0.8` to `2` (where `0.8` is equivalent to reclining, and `2` corresponds to medium activity.)



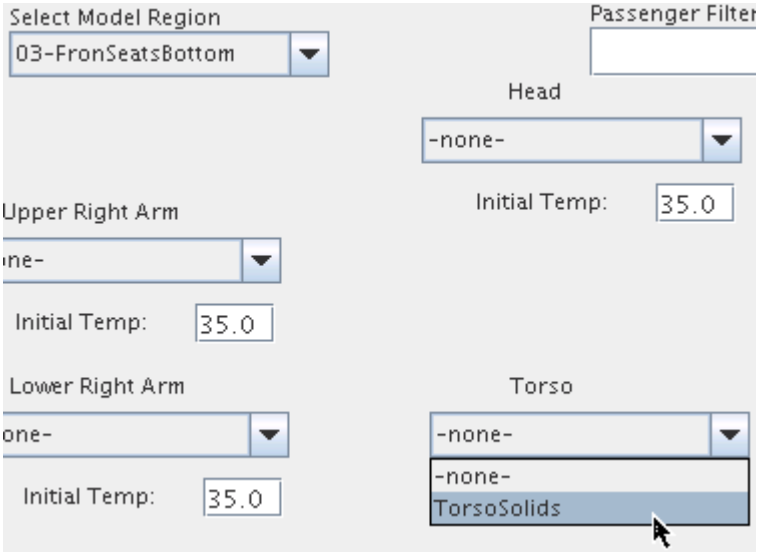
Make sure that the correct fluid region is selected under Select Model Region. The options in this drop-down list, and in similar ones for boundaries, correspond to the regions and boundaries that you created for this simulation file.



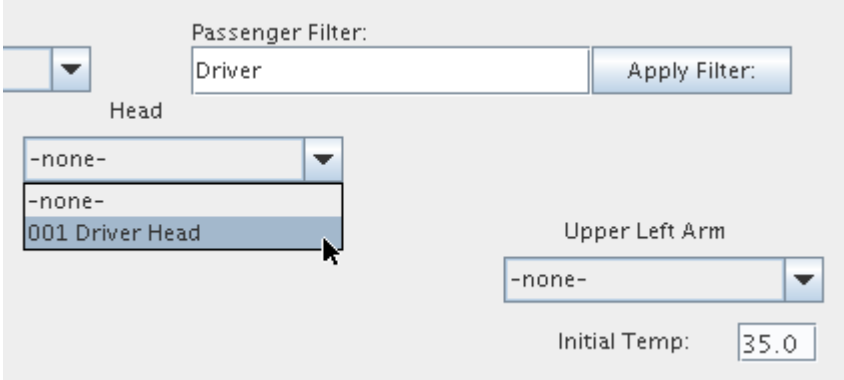
Select the appropriate boundary for each body segment.



Repeat this for any solid regions and boundaries that are in contact with the passenger.

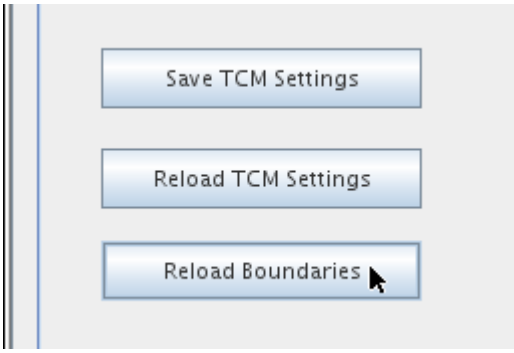


For a multiple occupant model, the *Passenger Filter* can be used as a search function to display the relevant boundaries for a particular passenger. For example, before opening the thermal comfort wizard, the driver boundaries can be renamed to contain the word driver. `Driver` can then be applied in the filter to display the required boundaries in the drop-down menu for each body segment.



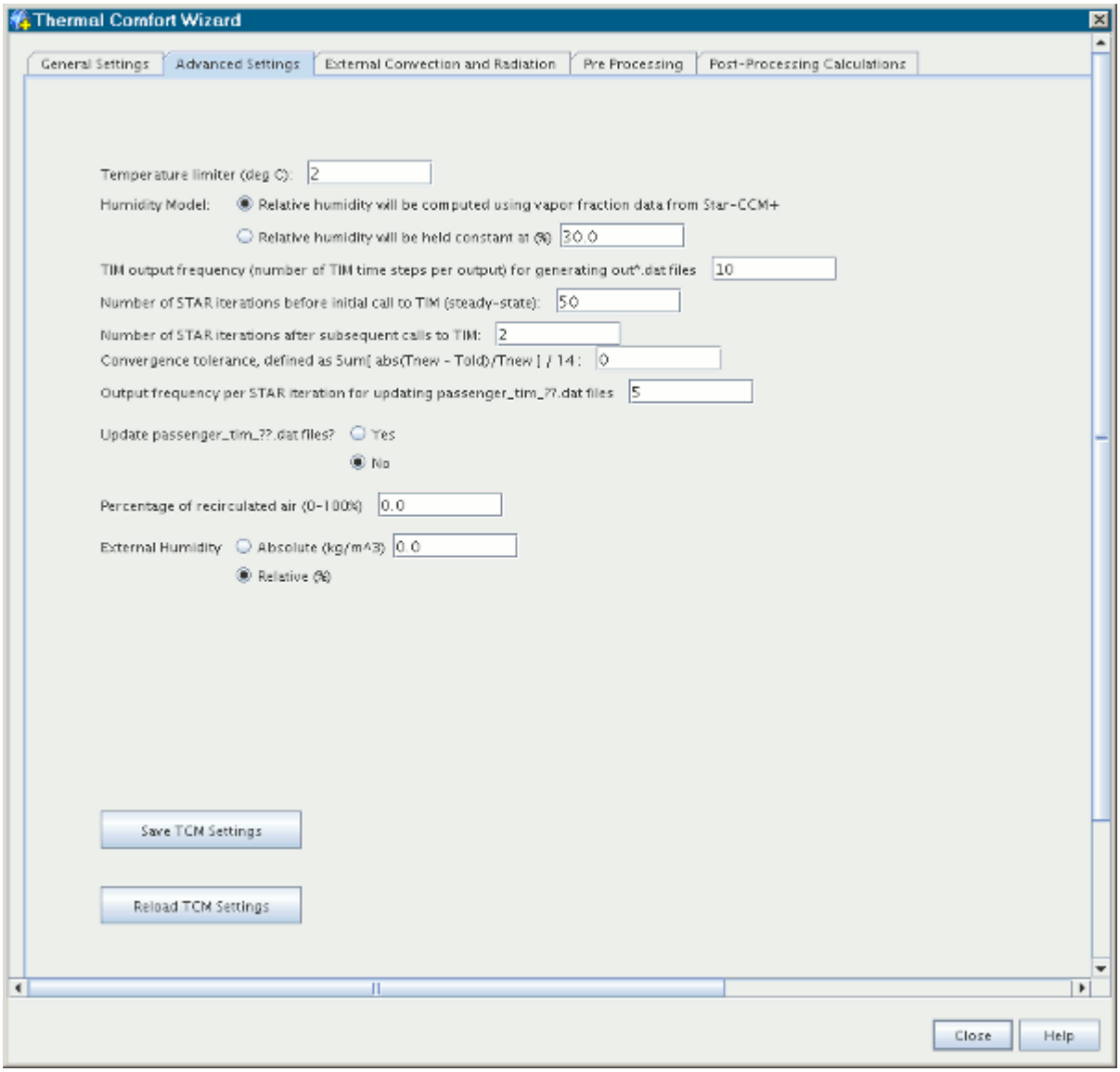
You can specify an initial body segment temperature in the **Initial Temp** text box. This is the value passed to the TIM program the first time it is called. Thereafter, the body segment temperature updates each time TIM is run. This body segment temperature differs from the boundary temperature specification set in the simulation tree. The boundary temperature specification is used during the initial iterations before calling TIM, and is then overwritten by the body segment temperature. All temperatures in the wizard are specified in degrees Celsius.

The **Reload Boundaries** button causes the *Thermal Comfort Wizard* to re-read the boundary information in the simulation and update options available in the Boundary Selection menus. This is used when boundary names have been changed in the simulation, and resets all boundary menus to `-none-`.

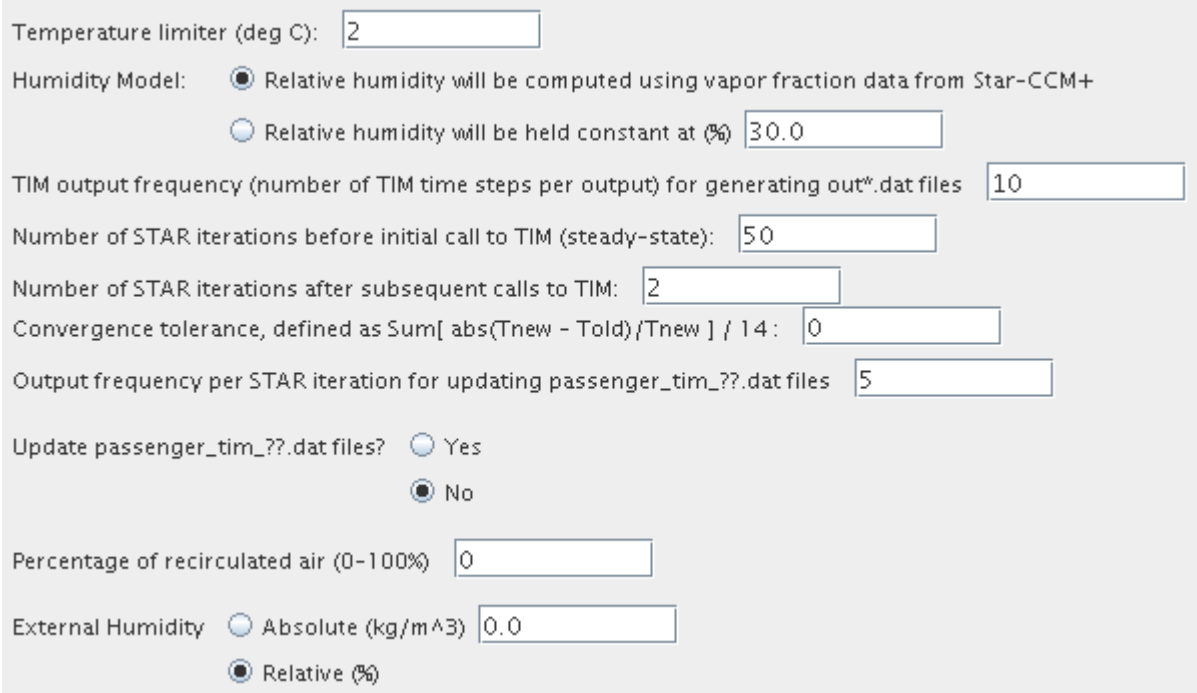


Using Advanced Settings

If you are an experienced user of TCM, you may want to adjust parameters in the *Advanced Settings* tab. Otherwise, leave the settings at their defaults.



This tab includes a host of controls with which you can refine your TCM settings.



- **Temperature Limiter**—the maximum amount TIM is allowed to change the skin temperatures it reports to STAR-CCM+ each time it runs. Larger values lead to faster convergence, but smaller values result in a tighter convergence.
- **Humidity Model**—This radio button determines whether the evaporation off of the passengers' skin is based on humidity values calculated in the CFD model, or uses a single fixed value for relative humidity. The default value for a constant humidity level is 30%.
- **TIM Output Frequency**—specifies how many TIM time steps to run each time it is called.
- **Number of STAR iterations before initial call to TIM**—sets how many iterations to run the CFD-only portion of the model. This gives the flow field time to develop before performing the passenger thermal calculations as well.
- **Number of STAR iterations after subsequent calls to TIM**—sets how frequently the passenger thermal calculations are performed.
- **Convergence tolerance**—stops the simulation if the rate of change of passenger skin temperatures falls below this threshold.
- **Output frequency**—specifies how often to update the files with detailed results of the passengers' thermal calculations. There is a separate file for each passenger, numbered `01`, `02`, ...
- **Update passenger_tim_??.dat files**—This radio button specifies whether to write to the files containing results of the passengers' thermal calculations.
- **Percentage of recirculated air (0-100%)**—Specifies how much of the cabin air is recirculated and cooled by the air conditioning system. For example, a value of 0 means that the air cooled by the air conditioning system has the same temperature and humidity as the external air. Similarly, a value of 100 means that the air being cooled has the same temperature and humidity as the air at the outlet boundary.
- **External Humidity**—This radio button specifies whether the external humidity used in the air recirculation calculations is relative or absolute. The value of external humidity (relative or absolute) is entered in the text field next to the radio button.

You can use the Maximum Steps and Stop File stopping criteria to stop the simulation running.

See Also:

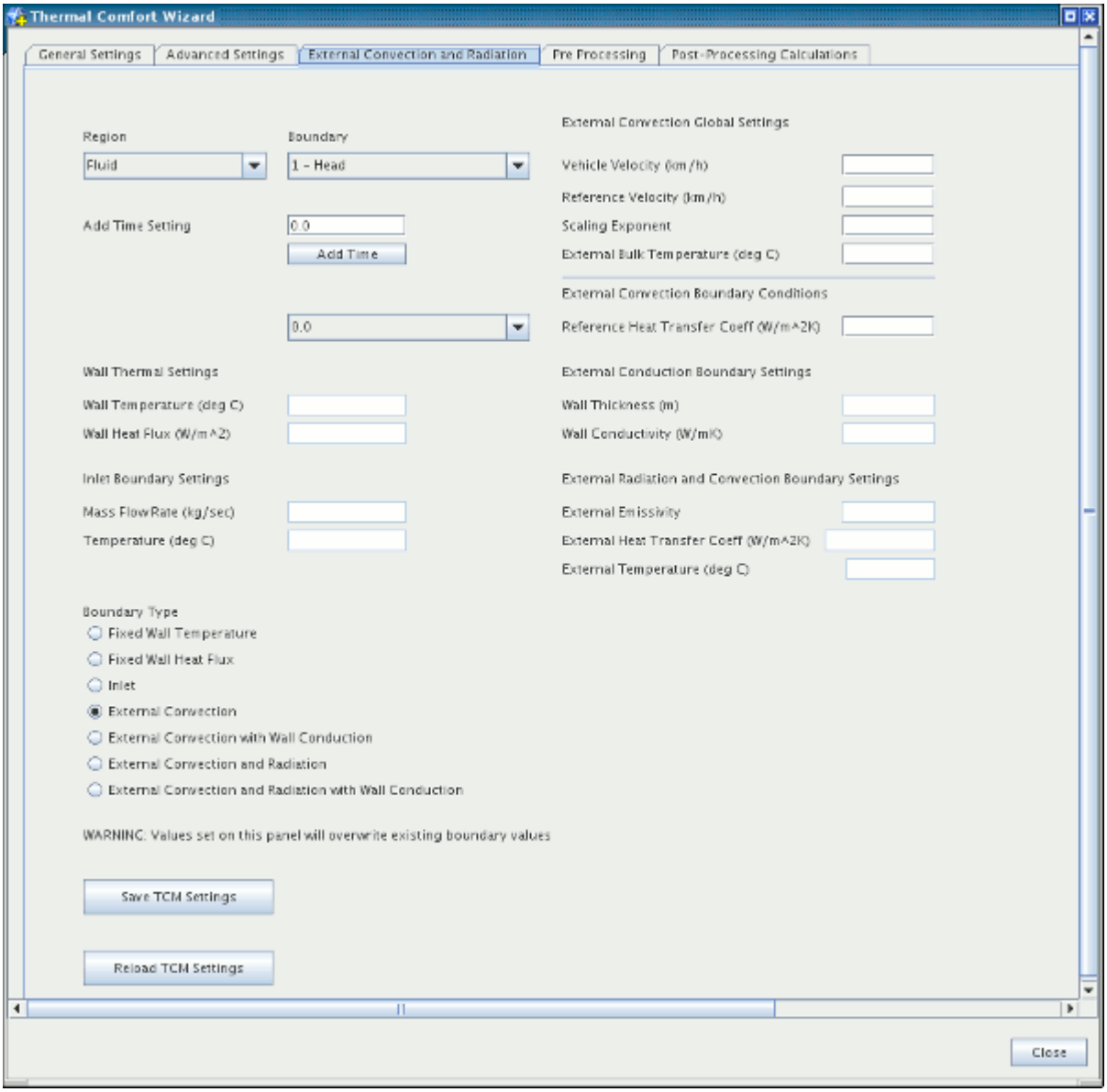
What is the Thermal Comfort Model?

Adding External Convection and Radiation

In the *External Convection and Radiation* tab you can define the properties of any thermal boundaries in the model, as well as specify the mass flow rate and temperature at the inlets.

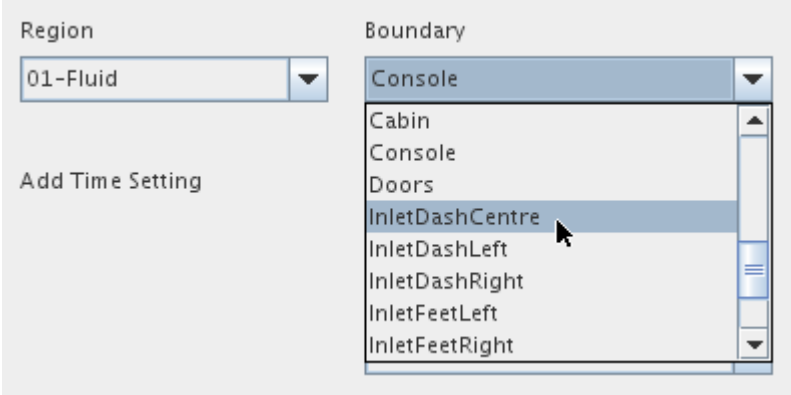
It lets you apply external convection and non-solar radiation heat fluxes easily, as well as specify changes to thermal properties with time (transient cases only). In order to do this, you must set up the appropriate boundary conditions in advance, for example setting the Thermal Specification to *Convection*. The values set in this panel are used by TIM when the TCM model is run, and overwrite any existing boundary values.

The functionality and different sections of the panel are described in detail below.



Boundary Selection

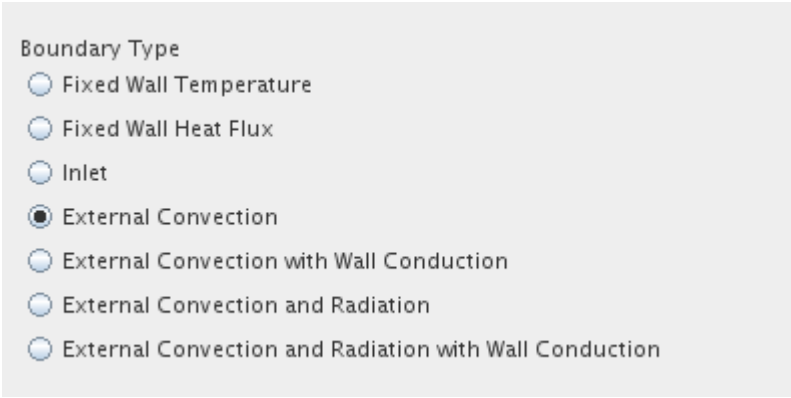
To specify the thermal properties or flow conditions for a boundary, it must first be selected using the drop-down menus labeled **Region** and **Boundary**.



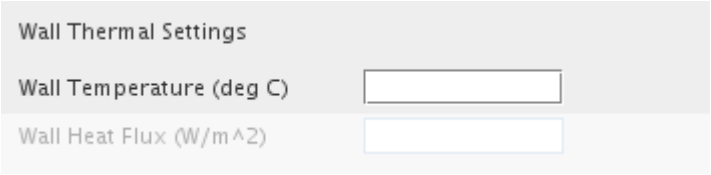
When you have finished entering data for a particular boundary, the next boundary can be selected. Data is retained for each boundary.

Boundary Type

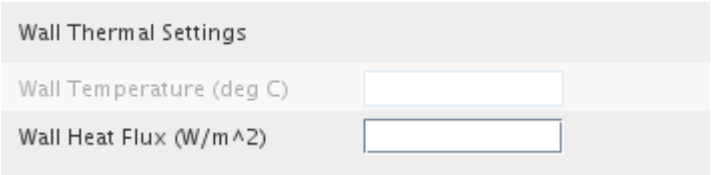
When the boundary has been selected, as described above, the next step is specifying the boundary type. This is done by selecting the appropriate radio button under Boundary Type, which activates the corresponding text boxes so that values can be entered. The different options are described below.



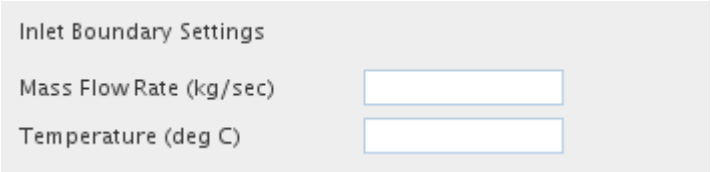
- **Fixed Wall Temperature**—boundary thermal specification is set to **Temperature**. Activates the **Wall Temperature** text box (shown below) under Wall Thermal Settings in which a constant temperature value can be entered.



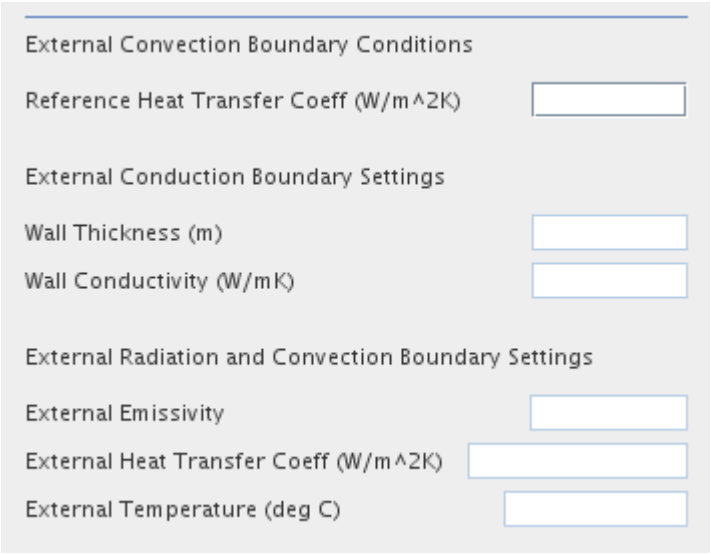
- **Fixed Wall Heat Flux**—boundary thermal specification is set to **Heat Flux**. Activates the **Wall Heat Flux** text box under Wall Thermal Settings in which a constant heat flux value can be entered.



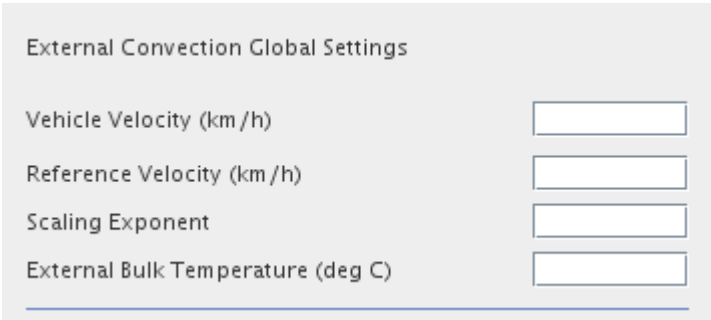
- **Inlet**—boundary type is set to **Mass Flow Inlet**. Activates the **Inlet Boundary Settings** text boxes where the mass flow rate and temperature at the inlet can be entered.



- **External Convection**—boundary thermal specification is set to *Convection*. Activates the **External Convection Boundary Condition** text box, allowing a reference heat transfer coefficient to be entered.
- **External Convection with Wall Conduction**—same as the previous setting but has more options for specifying a wall thickness and corresponding conductivity value in the **External Conduction Boundary Settings** text boxes. This allows you to simulate the insulating effects of walls without explicitly modeling them as solid cells.
- **External Convection and Radiation**—boundary thermal specification is set to *Convection*, and Radiation physics models have been enabled. Activates the **External Radiation and Convection Boundary Settings** text boxes, allowing external thermal properties to be entered. When this option is selected STAR-CCM+ performs a linearized approximation of the radiation heat flux in parallel to convective heat transfer.
- **External Convection and Radiation with Wall Conduction**—same as the previous setting but has more options for specifying a wall thickness and corresponding conductivity value in the **External Conduction Boundary Settings** text boxes. This allows you to simulate the insulating effects of walls without explicitly modeling them as solid cells.



The External Convection Global Settings are common to all boundaries, and allow heat transfer coefficients for one velocity (reference) to be applied to a model operating at a different velocity (vehicle). The external bulk temperature for the model is also specified here, which is used when calculating external convection and the temperature of recirculated air.



Because external heat transfer coefficient values may not be available for the speed at which the vehicle is traveling, TCM provides this method to scale the values that are available:

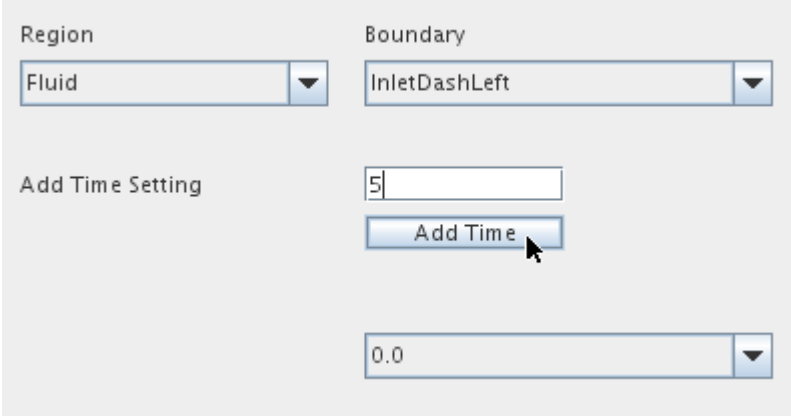
- **Vehicle Velocity** is the actual velocity at which the vehicle is traveling.
- **Reference Velocity** is the velocity at which the external heat transfer coefficients are valid.
- For each external boundary, a **Reference Heat Transfer Coefficient** can be entered. These heat transfer coefficients are scaled by the ratio of the two velocities raised to the **Scaling Exponent** before being applied to the boundaries:

  $$h_{app} = h_{ref} \cdot \left(\mathbf{v}_{vel}/\mathbf{v}_{ref}\right)^n$$

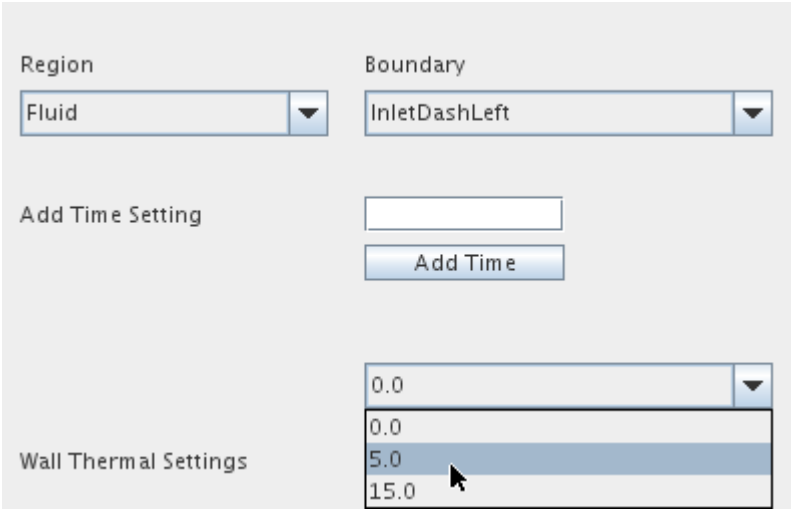
  where h_{app} is the applied heat transfer coefficient, h_{ref} is the reference heat transfer coefficient, $\mathbf{v}_{vel}$ is the vehicle velocity, $\mathbf{v}_{ref}$ is the reference velocity, and n is the exponent.

Adding Time Settings

In transient cases you can specify changes to the thermal properties of a boundary with time, for example at time `0` the wall temperature is `10` deg C, then at `30` seconds the temperature increases to `15` deg C. To do this, you must first specify after how many seconds a change occurs. This is done by entering a value (in seconds) in the **Add Time Setting** text box and clicking **Add Time**.



This value is added as an option in the drop-down menu as shown below. To use the time setting, first ensure that the appropriate boundary has been selected, then use the drop-down menu to select a time setting. When you do this the text boxes containing the boundary's values are cleared, allowing you to enter the new values. Values are scaled linearly between time settings and multiple time settings can be used in a TCM simulation.



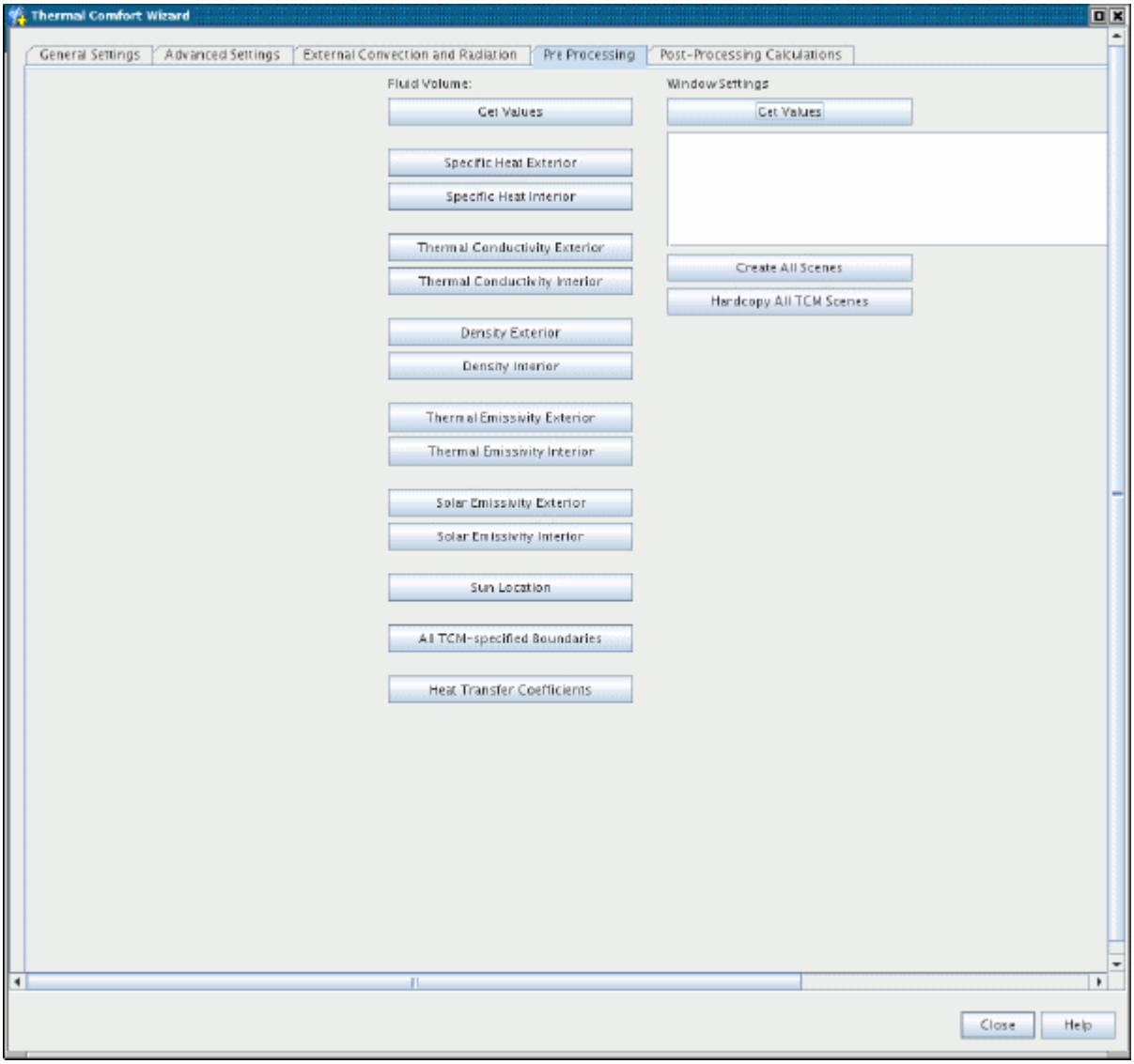
See Also:

What is the Thermal Comfort Model?

Pre Processing

The thermal comfort wizard allows you to perform more pre-processing steps in order to get useful data from the model and prepare scenes.

Details on preparing scenes for analysis are included in the subsection called Preparing Scenes. This section covers all other pre-processing capabilities available in the *Pre Processing* tab.



Fluid Volume > Get Values

When you click the **Fluid Volume > Get Values** button, the thermal comfort wizard evaluates the fluid volume in the model and report the size of the volume in m^3.

Window Settings > Get Values

Clicking the **Window Settings > Get Values** button reports the area of each window in the simulation. The boundaries are selected automatically based upon the standard naming conventions. A text file called `window_area.dat` is also written to the current working directory, which contains the window area information.

Creating All Scenes

Clicking the **Create All Scenes** button creates all the scenes described in the Preparing Scenes section. Each scene is created with an isometric view (-X, -Y, +Z) and a reverse isometric view (+X, +Y, -Z).

Hardcopy All TCM Scenes

Clicking the **Hardcopy All TCM Scenes** button creates a PNG image file of all scenes that you have created with the thermal comfort wizard. For example, if you have clicked **Specific Heat Exterior** and **Density Exterior**, then only those two scenes are exported. The image file is located in the current working directory.

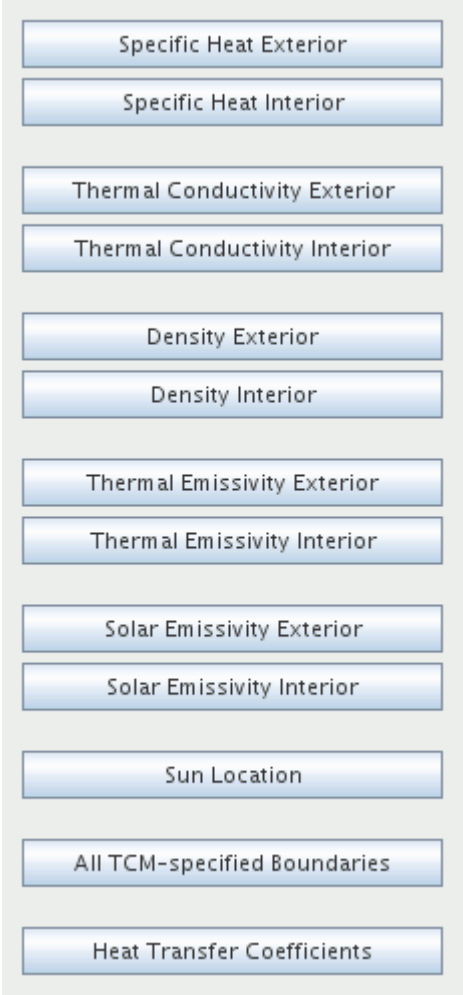
Contents:

Preparing Scenes

Preparing Scenes

This section describes how to prepare scenes.

The set of buttons shown below can be used to create scenes displaying contours of various field functions on certain boundaries in the model.



Specific Heat, Thermal Conductivity, and Density

These buttons generate scenes showing contours of specific heat, thermal conductivity, or density for the vehicle exterior or interior solids. The exterior scene includes all solid regions that do not have "`road`" in the name and are on the exterior of the model; it is intended for models in which the windows, doors, and other panels are modeled as solids. The interior scene is used to display contours on solid regions on the interior of the model, which are identified as those with "`seat`" in the name. If there are no solid regions in the model, these buttons are not activated.

Thermal Emissivity and Solar Emissivity

These buttons generate scenes showing contours of thermal emissivity or solar thermal emissivity for the exterior and interior surfaces of the fluid regions (not solid regions). The exterior scene displays contours on all fluid boundaries in the model, whereas the interior scene only includes boundaries that have "`seat`" in their name. If the model does not include a radiation model, these buttons are not activated.

It is assumed that two separate radiation spectral bands have been defined in the model, and that the first band has lower wavelengths (that is, is solar radiation), while the second band has higher wavelengths (that is, is non-solar radiation). For the Solar Emissivity scenes, the emissivity values for the first band are displayed. For the Thermal Emissivity scenes, the emissivity values for the second band are displayed.

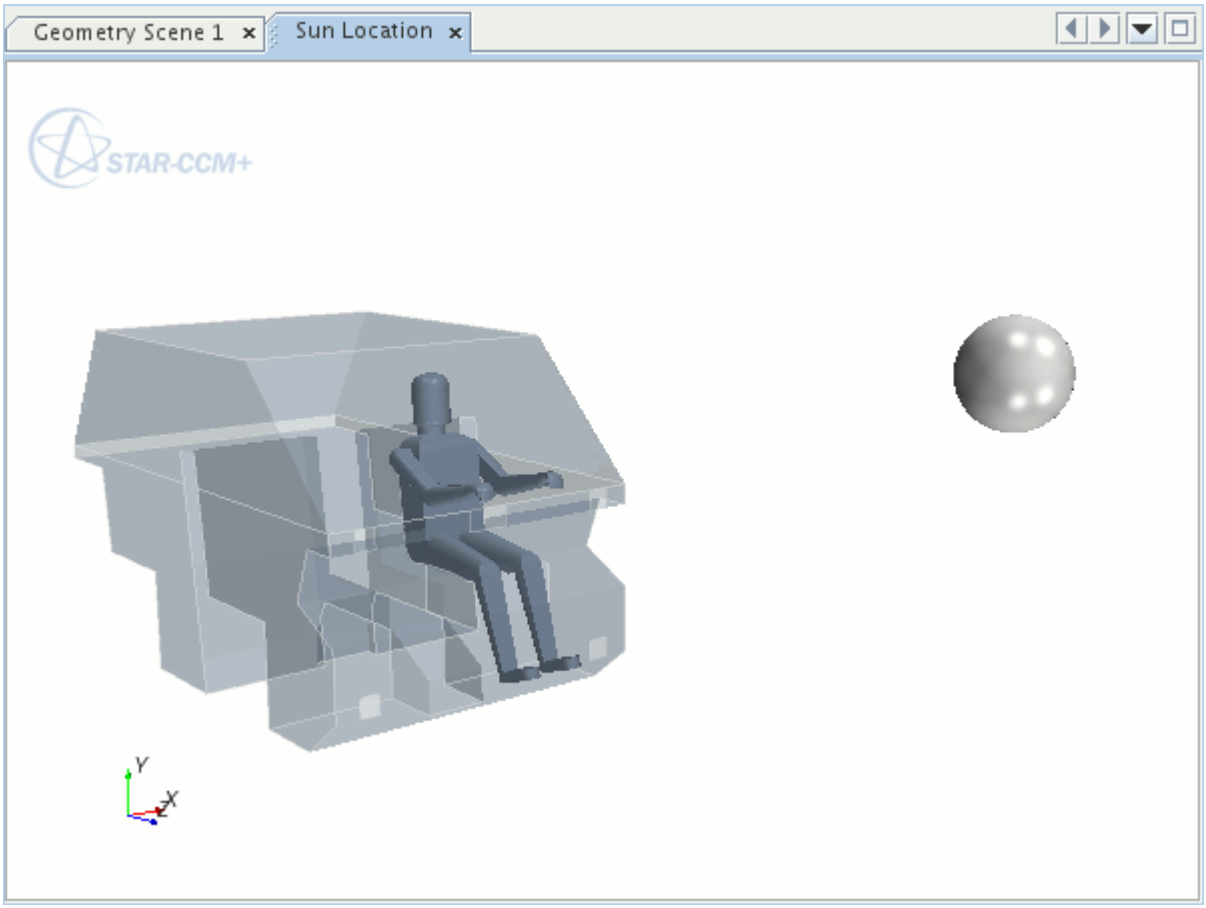
Displaying Sun Location

For a case where the sun has been included in the simulation by using the Solar Loads model, the **Sun Location** button can be used to give a visual indication of the location of the sun. The sun is represented in the model by a sphere, and its position is calculated automatically based on the *Azimuth* and *Altitude Angles* you have specified in the solar load properties.

When you click the **Sun Location** button:

- A new sphere part is created;
- A new *Sun Location* scene is created;
- The fluid region and the sun part are displayed in the scene.

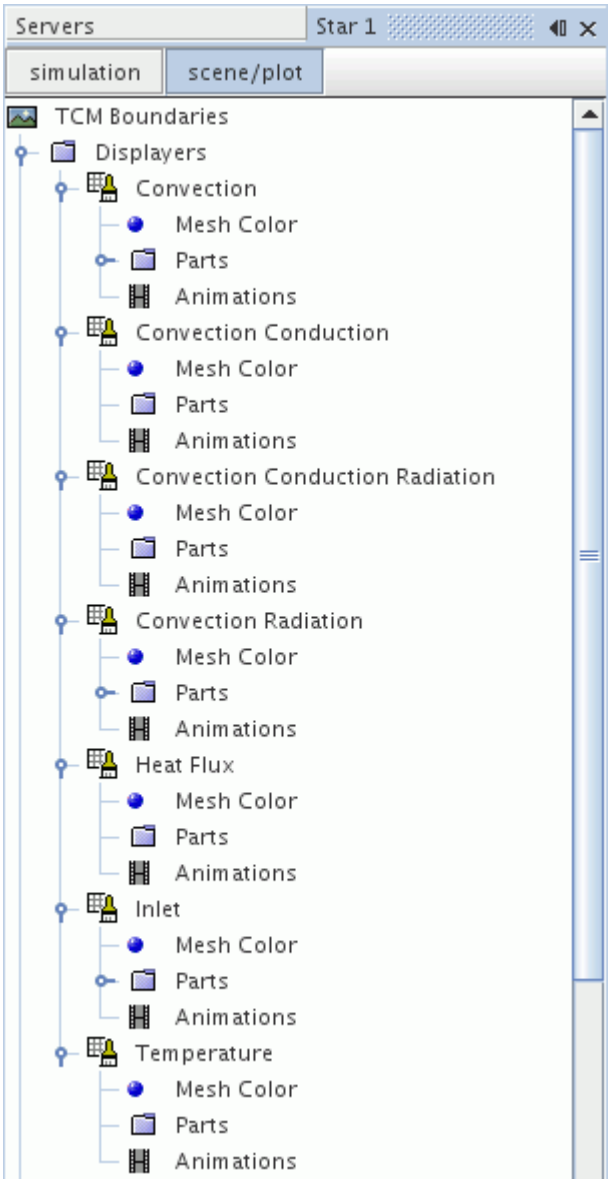
You are able to view this scene once you have closed the thermal comfort wizard.



Displaying All TCM-Specified Boundaries

Clicking the **All TCM-specified Boundaries** button creates a geometry scene displaying all boundaries specified within the thermal comfort wizard's *External Convection and Radiation* tab. This is useful for identifying the boundary types assigned in the thermal comfort wizard, and quickly creating scalar or vector scenes with the TCM boundaries by adding scalar or vector displayers.

The TCM boundaries are categorized with a different geometry displayer for each boundary type.



A mesh color is applied to each scene displayer to visually distinguish between TCM boundary types. The scene displayers, equivalent TCM boundary type, and mesh color are summarized below:

Scene Displayer	TCM Boundary Type	Mesh Color
Convection	External Convection	Red
Convection Conduction	External Convection with Wall Conduction	Red
Convection Conduction Radiation	External Convection and Radiation with Wall Conduction	White
Convection Radiation	External Convection and Radiation	Cyan
Heat Flux	Fixed Wall Heat Flux	Blue
Inlet	Inlet	Magenta

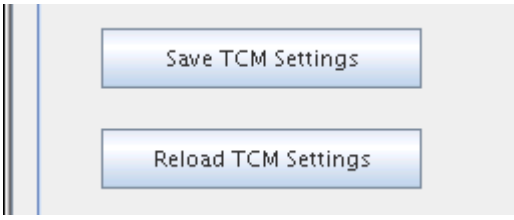
Scene Displayer	TCM Boundary Type	Mesh Color
Temperature	Fixed Wall Temperature	Green

Heat Transfer Coefficients

Clicking the **Heat Transfer Coefficients** button creates a scalar scene displaying the heat transfer coefficients for all solid regions and boundaries. This speeds up the solution monitoring and post-processing requirements by automatically creating a heat transfer coefficient scene with typical settings and boundaries.

Saving and Reloading TCM Settings

The **Save TCM Settings** button takes the data currently entered in the thermal comfort wizard, and adds it to the relevant placeholders in the object tree.



You can also load the existing TCM settings into the wizard by clicking the **Reload TCM Settings** button. This replaces the current settings in the wizard with the TCM settings stored in the STAR-CCM+ object tree. Any changes made in the dialog since the last time the TCM settings were saved are overwritten when the data is loaded.

Performing a TCM Simulation

Once the TCM settings have been saved, the simulation can be run.

You can run the TCM simulation as a steady-state or a transient simulation. The TIM program itself is a transient code.

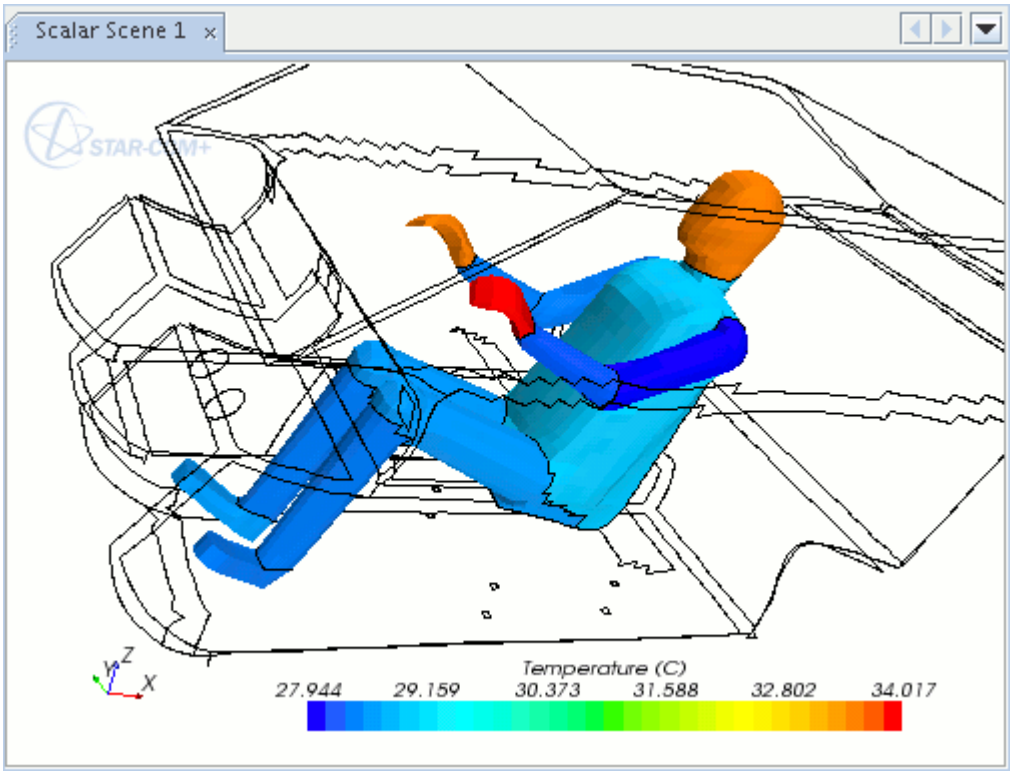
For steady-state simulations, STAR-CCM+ instructs TIM to solve for a very long period of time (18000 seconds) each time that it is called.

For transient simulations, STAR-CCM+ automatically instructs TIM to advance its solution by amount that is equal to the specified time-step size of STAR-CCM+. To achieve this, TIM performs one or more internal time-steps. To synchronize the two simulations, do one of the following:

- In the Thermal Comfort Wizard, select the *Advanced Settings* tab and set **Number of STAR iterations after subsequent calls to TIM** to the same value as the *Maximum Inner Iterations* under **Stopping Criteria** > **Maximum Inner Iterations** in the STAR-CCM+ simulation tree.
- In the STAR-CCM+ simulation tree, select the **Solvers** > **Thermal Comfort** node and set *STAR-CCM+ Iterations per TIM* to the same value as the *Maximum Inner Iterations* under **Stopping Criteria** > **Maximum Inner Iterations**.

The solver runs the STAR-CCM+ simulation for the specified number of iterations, then passes the heat flux values for each body segment to the TIM program. TIM uses the heat fluxes to calculate skin and clothing temperatures, which are passed back to STAR-CCM+. The two programs continue this iterative process until the change in body temperature falls below the convergence criterion, until you stop the solver, or until the stopping criteria have been reached.

The results can be displayed in a typical scalar scene.



Using Auto Save in a TCM Simulation

The auto save capability within STAR-CCM+ can be used in a TCM simulation to save the `passenger_state.dat` file at given intervals. This file is used by TIM to store data relating to the thermal state of the passenger, and is updated after each TIM iteration. When the `passenger_state.dat` file exists in the working directory, TIM automatically uses this as the starting point for subsequent iterations. By using the autosave facility, not only can you save backup copies of the `.sim` file, but you can also save the corresponding `passenger_state.dat` file, allowing you to restore the simulation following a system failure, for example.

When restarting from a solution other than the current one, it is usually a good idea to restore the `passenger_state.dat` file as well as the CFD solution. This is because the internal body temperatures are stored in this file and not in the `.sim` file. To restore a previous solution, replace the `passenger_state.dat` file with the desired `passenger_state.dat.xx` file, by removing the `.xx` from the end of the file name. This has no effect on the TCM settings in the `passenger.dat` file.

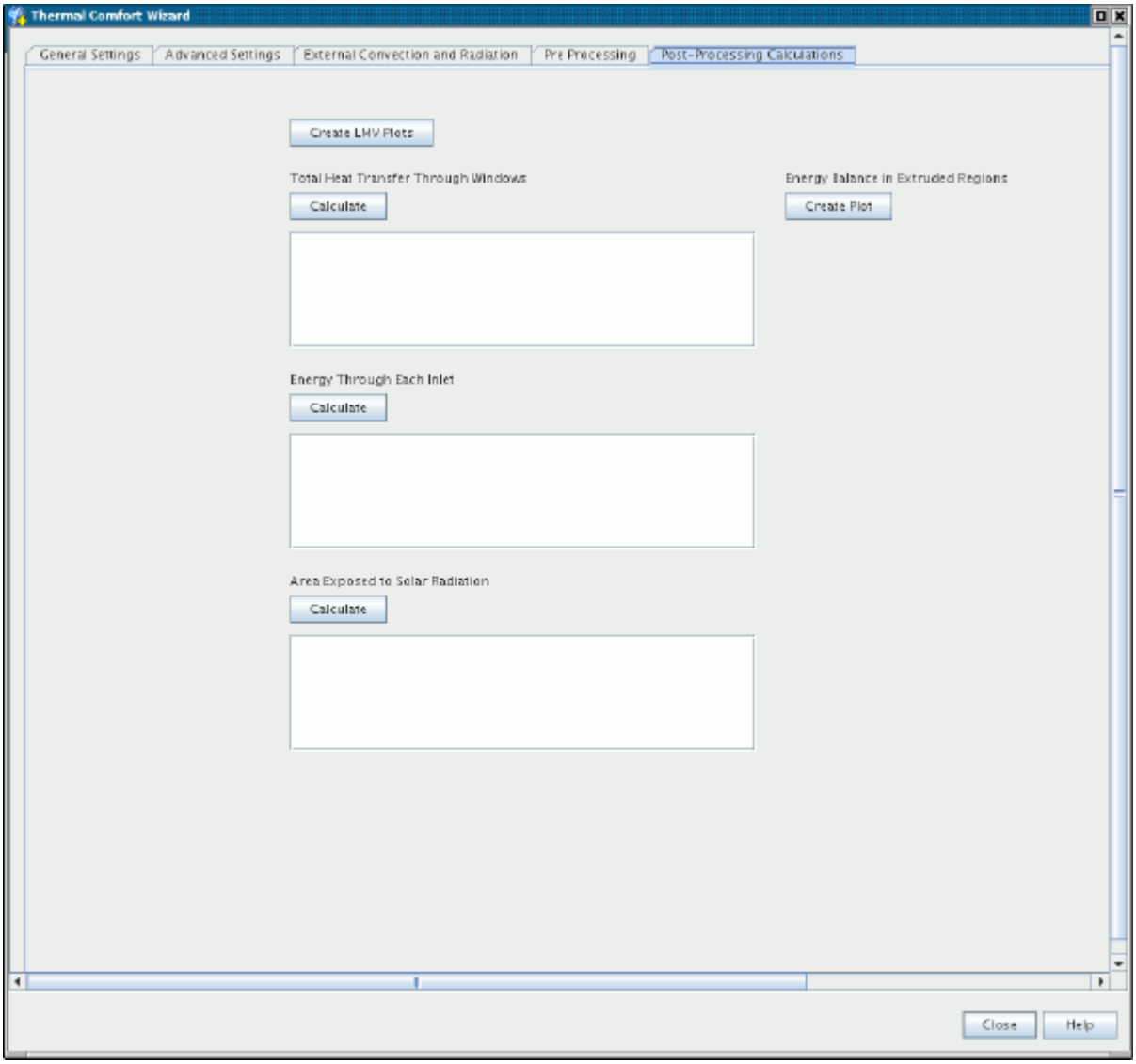
See Also:

Saving and Reloading TCM Settings

Post Processing a TCM Simulation

The thermal comfort wizard has some additional post-processing capabilities that are specific to thermal comfort simulations.

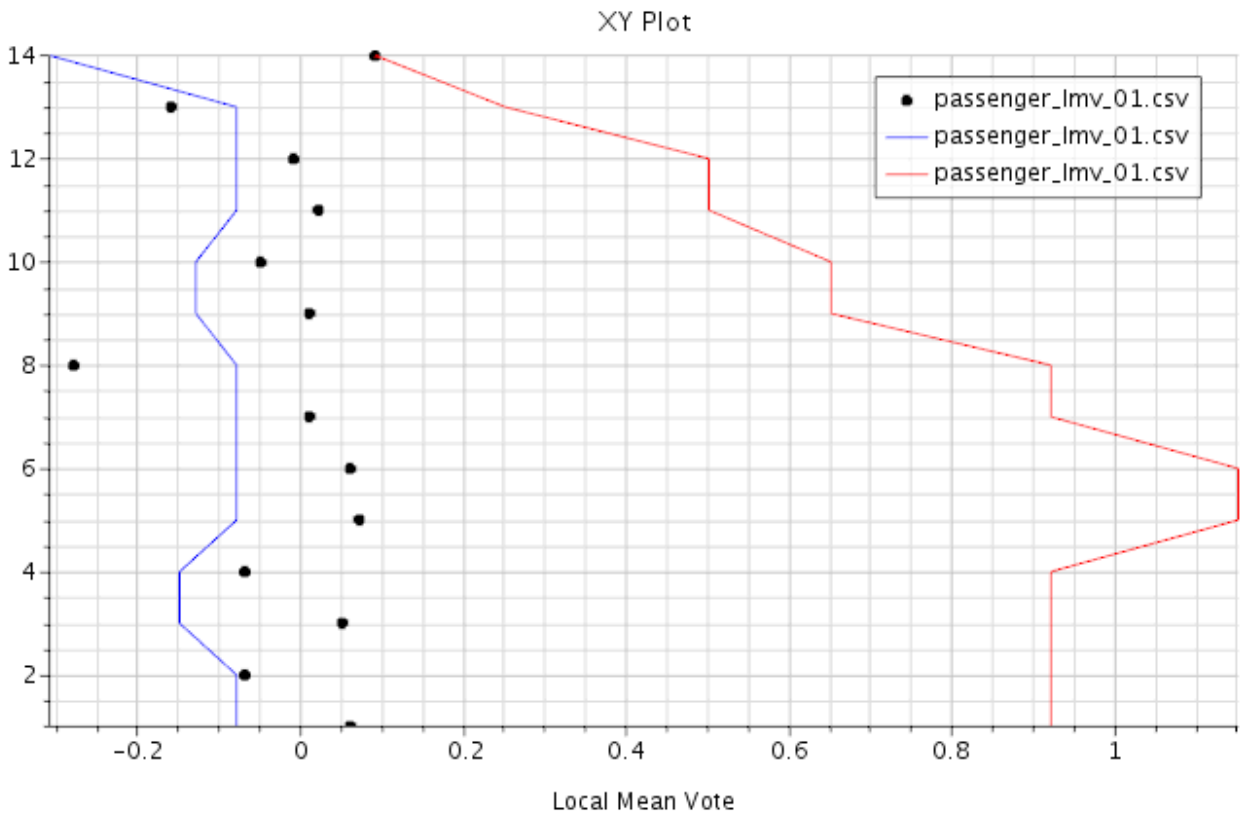
These are available in the *Post-Processing Calculations* tab and are described below.



Creating LMV Plots

The **Create LMV Plots** button allows you to create plots displaying the LMV (local mean vote) values for occupants in the model. When you click this button, the plot(s) is created and a hardcopy of each plot, such as

the one shown below, is saved as a `.png` file. As well as this, a `.csv` file containing the numerical data is generated. LMV calculations require that radiation be active in the model.



In the LMV plot, the upper and lower limits for thermal comfort are indicated by red and blue lines. The corresponding local mean vote for each part is shown as a black dot on the plot. If a black dot is on the left of the blue line, the corresponding body part may be considered to be too cold. If the black dot is on the right of the red line, the body part may be considered to be too hot. In either of these cases, this would indicate an inadequate level of thermal comfort for the passenger.

Energy Balance in Extruded Regions

Clicking the **Create Plot** button under *Energy Balance in Extruded Regions* creates a plot displaying the heat transfer (measured in Watts) on the interior and exterior surface of extruded regions. Before creating this plot, you must follow the naming conventions below so that the wizard selects the appropriate regions and boundaries:

- The name of the extrusion region must start with `Extrusion` (for example, Extrusion roof).
- The name of the interior boundary must end with `base` (for example, Interior base).
- The name of the exterior boundary must end with `top` (for example, Exterior top).

Calculating Values

There are three boxes in the *Post-Processing* tab that allow you to calculate the following values:

- Total heat transfer through windows

  This is the total heat transfer through each window, in W. A file called `window_heat_transfer.dat` is created in the current working directory, which contains the heat transfer data.
- Energy through each inlet

  This is the amount of energy transferred into the cabin through each inlet, in W.
- Area exposed to solar radiation

  This is the total area of faces where direct solar irradiation is >0 for each window and passenger boundary, in m^2.

Clicking the **Calculate** button for each value causes the thermal comfort wizard to perform the calculations and report the values in the corresponding box.

Working with TCM Input Files

This section describes the format of the external `passenger.dat` file, which is used with the thermal comfort wizard.

This file contains values for each passenger, and is used with the information in the TCM Passenger and TCM Boundaries managers for the thermal comfort calculations. The file is composed of sections, each of which starts with `SECTION <SECTION NAME>` and finishes with `END`. Any lines in the file which begin with `%` are considered comments. Additionally, everything following `!` is considered a comment.

An example of the file format has been provided.

PASSENGER

`SECTION PASSENGER` contains one line, with the number of passengers in the simulation (including the driver, if applicable).

PASSDATA

`SECTION PASSDATA` loops through data of all the passengers. For each passenger, the following items are required (one item per line):

- Passenger name
- Passenger height (in meters)
- Passenger surface metabolic rate (relative to 58 W / m^2)
- Clothing resistance (in m^2K/W)—This is applied to all body segments except the head and hands.
- Initial surface temperatures (in degrees C) for each body segment -- Each passenger is divided into 14 segments, and data for the segments is always presented in the sequence shown in the example.

After all of the data is given for Passenger #1, the sequence is repeated for Passenger #2, #3, and so on, as needed.

Contents:

File Format Example

See Also:

What is the Thermal Comfort Wizard?

What are TCM Passengers?

What are TCM Boundaries?

File Format Example

This section contains an example of a typical `passenger.dat` file.

```
% All lines that begin with a % are considered as comments
% ======================================================
SECTION PASSENGER
% ======================================================
1
END
%
SECTION PASSDATA
% ======================================================
% Data for passenger 1
% ======================================================
Driver ! Passenger name
1.85 ! Passenger height
1.3 ! Passenger surface metabolic rate (relative to 58 W/m^2)
0.035 ! Clothing resistance (m^2K/W) all segments except head & hands
%
%
% Initial body segment surface temperatures (deg C) (must be in this
order)
35.0 ! Head
35.0 ! Torso
35.0 ! Upper Left Arm
35.0 ! Upper Right Arm
35.0 ! Lower Left Arm
35.0 ! Lower Right Arm
35.0 ! Left Hand
35.0 ! Right Hand
35.0 ! Upper Left Leg
35.0 ! Upper Right Leg
35.0 ! Lower Left Leg
35.0 ! Lower Right Leg
35.0 ! Left Foot
35.0 ! Right Foot
% ============ END OF PASSENGER 1 DATA ==================
% To add more passenger data, simply repeat the above lines for each
passenger
```

```
END
% ======================================================
```

Working with TCM Output Files

This section details the contents of the thermal comfort output file, `passenger_tim_xx.txt`, where `xx` refers to the passenger number in the simulation.

This file contains information regarding the passenger characteristics used in the simulation, and a summary of results. The sections of the output files are listed below:

- Passenger information:
  - Anthropometric Basis Data
  - Initial Thermo-Physiological Quantities
  - Isolation Values: Clothing / Car Seat
  - Basis Values for Body Segments
  - Boundary Conditions at Beginning of Computation
- Results:
  - Final Thermo-Physiological Quantities
  - Values for Body Segments
  - Boundary Conditions at End of Computation

Contents:

Anthropometric Basis Data
Initial Thermo Physiological Quantities
Isolation Values Clothing / Car Seat
Basis Values for Body Segments
Boundary Conditions at Beginning of Computation
Final Thermo-Physiological Quantities
Values for Body Segments
Boundary Conditions at End of Computation

Anthropometric Basis Data

This section of the `passenger_tim_xx.txt` file contains details regarding the passenger anthropometry.

```
ANTHROPOMETRIC BASIS DATA :
============================

          SIZE (M)                    =
          WEIGHT (KG)                 =
          SKIN UPPER SURFACE (M2)     =
```

The passenger weight and skin surface area are defined based upon the passenger height entered in the wizard.

Initial Thermo Physiological Quantities

This section of the `passenger_tim_xx.txt` file contains information relating to the passenger vascular system, heat production, and heat loss, at the start of the run.

```
THERMO-PHYSIOLOGICAL QUANTITIES :
=================================

     CENTRAL VASCULAR SYSTEM :
          MASS (KG)                    =
          HEAT CAPACITY (J/K)          =
         REF.-TEMPERATURE (C)          =

     HEAT PRODUCTION (W) :
          BREATHING                    =
          ADDITIONAL ACTIVITY          =
          ENTIRE CONVERSION            =
          IN MET:                      =

     HEAT LOSS (W) :
         BREATHING
         SKIN      :
           SWEATING AND EVAPORATION    =
           DRY HEAT LOSS               =
           OVERALL                     =
         BODY    :
          SKIN AND RESPIRATION         =

     HEAT BALANCE (W) :
          PRODUCTION - LOSS            =


     BLOOD VOLUME FLOW  (CM3/S) :
          BREATHING                    =
```

The `IN MET` property shown above relates to the metabolic rate entered in the wizard. This value, along with the passenger skin surface area is used to determine the `ENTIRE CONVERSION` property, which is the total passenger heat production.

Isolation Values Clothing / Car Seat

This section of the `passenger_tim_xx.txt` file contains information relating to the passenger clothing and car seats.

```
ISOLATION VALUES: CLOTHING / CAR SEAT :
=========================================

     CLOTHING:
     ----------
     UNCLOTHED:

     TAKING HUMIDITY TRANSFER RESISTANCE
     CLOTHING ACCORDING TO NISHI/GAGGE


     CAR SEAT:
     ------------
           PADS:  HEAT CAPACITY
              HUMIDITY TRANSFER RESISTANC

      THE BACK  : HEAT CAPACITY
              HUMIDITY TRANSFER RESISTANT
```

Basis Values for Body Segments

This section of the `passenger_tim_xx.txt` file contains tabulated data regarding body segments and segment layers.

This section of the `passenger_tim_xx.txt` file summarizes the following properties:

- Mass (kg)
- Heat capacity (J/K)
- Skin surface area (m^2)
- Heat production from breathing (MW)
- Evaporation from skin (W)
- Blood flow for breathing (cm^3/s)
- Heat transfer (W/K)
- Temperature (C)

Provided below is an example of the information found in this section:

```
 MASS PORTION OF BODY SEGMENT (KG) :

                CENTER      MUSCLE      FAT         SKIN
-------------------------------------------------------
HEAD             4.047       0.744       0.372       0.268
TRUNK           11.113      17.549       7.067       1.346
UP-ARM LE        0.562       0.842       0.242       0.121
UP-ARM RI        0.562       0.842       0.242       0.121
LO-ARM LE        0.562       0.842       0.242       0.121
LO-ARM RI        0.562       0.842       0.242       0.121
HAND LE          0.130       0.037       0.074       0.093
HAND RI          0.130       0.037       0.074       0.093
UP-LEG LE        2.078       3.053       0.714       0.359
UP-LEG RI        2.078       3.053       0.714       0.359
LO-LEG LE        1.385       2.035       0.476       0.240
LO-LEG RI        1.385       2.035       0.476       0.240
FOOT LE          0.216       0.037       0.112       0.119
FOOT RI          0.216       0.037       0.112       0.119
-------------------------------------------------------
TOTAL  (KG) :   71.890
```

Boundary Conditions at Beginning of Computation

This section of the `passenger_tim_xx.txt` file contains information about the boundary conditions on each of the passenger body segments at the start of the run.

The following conditions are detailed:

- Heat flux coefficients:
  - Convection
  - Radiation
  - Clothing
  - Heat transfer coefficient
  - Effective temperature (C)
- Temperatures and coefficients:
  - Air temperature
  - Confinement surface temperature
  - Thermal efficiency factor
  - Permeation efficiency factor

Final Thermo-Physiological Quantities

This section of the `passenger_tim_xx.txt` file contains results relating to the passenger vascular system, heat production, and heat loss.

```
THERMO-PHYSIOLOGICAL QUANTITIES  :
================================

     CENTRAL BLOOD  :
         TEMPERATURE (C)              =
           HEAT TRANSPORT             =


     HEAT PRODUCTION (W) :
          BREATHING                   =
          ADDITIONAL ACTIVTIY         =
          ENTIRE CONVERSION           =


     HEAT LOSS (W) :
         BREATHING            :
         SKIN      :
           SWEATING AND EVAPORATION  =
           DRY HEAT LOSS             =
           OVERALL                   =
          BODY   :
          SKIN AND RESPIRATION       =


     HEAT BALANCE (W) :
          PRODUCTION -  LOSS         =


     BLOOD VOLUME FLOW  (CM3/S) :
          BREATHING                  =
          CONTROL PORTION            =
          ENTIRE CONVERSION          =
```

Values for Body Segments

This section of the `passenger_tim_xx.txt` file contains tabulated data regarding body segments and segment layers:

This section of the `passenger_tim_xx.txt` file summarizes the following properties:

- Heat Production (MW)
- Heat flux from skin (W)
- Blood flow (cm^3/s)
- Heat transfer between layers (W/K)
- Temperature difference (K)
- Temperature allocation (C)
  - Mean skin temperature (C)
- Surface temperature (C)
- Heat balance (W)
- Heat flux from skin (W)
- Evaporation heat balance (W)

- Water vapor pressure (Pa)

Provided below is an example of the information found in this section:

```
 SEGMENT HEAT PRODUCTION  (MW) :

                CENTER        MUSCLE       FAT        SKIN
-------------------------------------------------------------
HEAD         14718.80        435.11      130.80      62.90
TRUNK        47916.20      28491.39     2560.80     325.00
UP-ARM LE      184.80       1479.28       97.60      21.80
UP-ARM RI      184.80       1479.28       97.60      21.80
LO-ARM LE      184.80       1479.28       97.60      21.80
LO-ARM RI      184.80       1479.28       97.60      21.80
HAND LE         48.30        279.51       31.60      19.90
HAND RI         48.30        279.51       31.60      19.90
UP-LEG LE      721.98      10895.77      255.60      92.88
UP-LEG RI      721.98      10895.77      255.60      92.88
LO-LEG LE      481.32       7263.85      170.40      61.92
LO-LEG RI      481.32       7263.85      170.40      61.92
FOOT LE         82.40        271.61       46.80      33.50
FOOT RI         82.40        271.61       46.80      33.50
-------------------------------------------------------------
TOTAL  (W)  :  143.29
```

Boundary Conditions at End of Computation

This section of the `passenger_tim_xx.txt` file contains information about the boundary conditions on each of the passenger body segments at the end of the run.

The following conditions are detailed:

- Heat flux coefficients:
  - Convection
  - Radiation
  - Clothing
  - Heat transfer coefficient
  - Effective temperature (C)
- Temperatures and coefficients:
  - Air temperature
  - Confinement surface temperature
  - Thermal efficiency factor
  - Permeation efficiency factor

Bibliography

This section lists the bibliographic references for the Thermal Comfort wizard.

[2] *Meeh, K. 1879. “Oberflächenmessung des menschlichen Körpers”, J. Biol., 15, pp. 428-458*

[3] *DuBois, D. and DuBois, E.F. 1915. “Clinical Calorimetry: Tenth paper, A formula to estimate the approximate surface area if height and weight be known”, Arch. Intern. Med., 17, pp. 863-871*

[4] *Stolwijk, J.A.J. 1971. “A mathematical model of physiological temperature regulation in man”, NASA Contractor Report, NASA CR-1855*

[5] *Berbner, D.F., Kerslake, D.McK., and Waddel, J.L. 1958. The relation between the coefficients for the heat exchange by convection and by evaporation in man”, J. Physiol., 141, pp. 164-168*

[6] *Aschoff, J. and Wever, R. 1958. “Kern und Schale im Wärmehaushalt des Menschen”, Die Naturwissenschaften, 45, Nr. 20, pp. 477-485*

[7] *Stolwijk, J.A.J. and Hardy, J.D. 1966. “Temperature regulation in man - A theoretical study”, Pflüger Arch., 307, pp. 129-162*

[8] *Hertzman, A.B. and Randall, W.C. 1948. “Regional differences i the basal and maximal rates of blood flow in the skin”, J. Appl. Physiol., pp. 234-241*

[9] *Benzinger, T.H., Kitzinger, C., and Pratt, A.W. 1963. “The Human Thermostat”, In: Temperature. Its Measurement and Control in Science and Industry, 3, Part 3, Biology and Medicine, Hardy, J.D. (Editor)*

[10] *Nadel, E.R. 1969. “The role of peripheral thermoreceptors in the integrated thermoregulatory response in resting and excercising man”, Dissertation, Univ. Calif., Santa Barbara*

Introduction to the SPEED Motor Tool Import Wizard

This section describes the STAR-CCM+ SPEED Motor Tool Import Wizard.

SPEED is a software package that Siemens PLM Software provides for use in the design and analysis of electrical machines. It provides a wide array of machine design options, including a material database and analysis programs to evaluate machine performance. The STAR-CCM+ SPEED Motor Tool Import Wizard lets you import SPEED electrical machine data into STAR-CCM+ for further analysis, such as calculating 3D magnetic fluxes within the electrical machine.

Contents:

Importing SPEED Data Into STAR-CCM+
Accessing the SPEED Motor Tool Import Wizard
Using the SPEED Motor Tool Import Wizard
Using SPEED Data in STAR-CCM+

Importing SPEED Data Into STAR-CCM+

You can import SPEED data into STAR-CCM+ using a geometry definition file (`.xGDF`).

In the general SPEED finite element analysis workflow, after you have defined the machine geometry using the SPEED design module, you can use the GoFER tool to set up a finite element analysis. The GoFER tool creates an `.xGDF` file that contains the machine design and the selected material properties.

The `.xGDF` file contains the following information:

- Geometry of the machine
- Material properties and geometry of each subdomain
- Eccentricity details
- Skew details
- Material property tables that describe the magnetic flux density $\mathbf{B}$, magnetic field strength $\mathbf{H}$, and permeability $\mathbf{B/H}$ of the motor components.
- Motor type details that are required for modeling the endwindings, and winding information from the SPEED winding editor.

  STAR-CCM+ currently supports importing the following for the stator:

  - Simple windings

    Simple windings are defined in the STAR-CCM+ SPEED Motor Tool Import Wizard. These windings are single-layer (one coil per slot), where the span of each winding is one. Therefore, for a twelve-slot stator, six coils are created. If you choose not to run an electromagnetic simulation, these windings are further simplified by merging the individual windings into one winding body. These merged windings are referred to as "simple-bulk" or "bulk-type" windings. For simple windings, the Import Wizard is primarily used to specify the shape of the endwindings.
  - Complex windings

Complex windings are defined in the `.xGDF` file. These windings can be single-layer or two-layer (two coils per slot). A slot can also be empty. The span of each winding can be greater than one (the winding does not return to its neighboring slot) and spans can also be negative. For this reason, complex windings are also known as distributed windings.

STAR-CCM+ currently supports single-layer, complex windings, for all spans. STAR-CCM+ also supports two-layer, complex windings but only if the winding spans are all one. STAR-CCM+ currently does not support many realistic two-layer, complex windings for spans greater than one.

The `.xGDF` file contains no information on the exterior flow domain. You can specify this information manually when you import the `.xGDF` file.

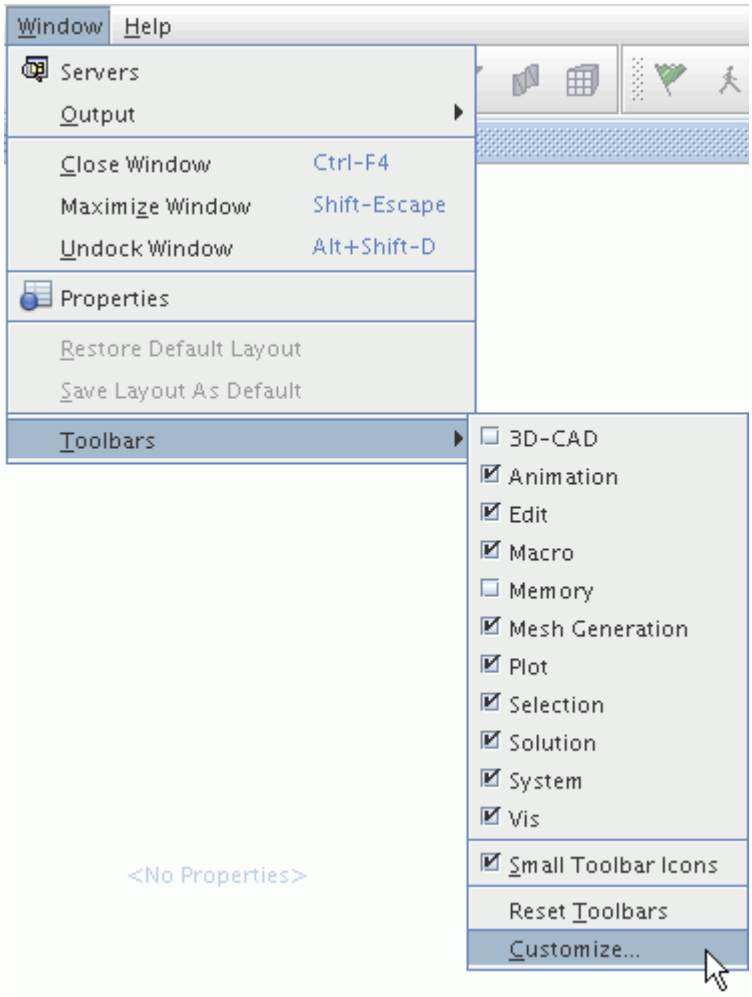
You can import a `.xGDF` file that contains a symmetrical sector of the machine design, rather than the complete geometry. The sector can be any appropriate section that utilizes the motor symmetries, such as a half sector, quarter sector, sixth sector, or eighth sector. For more information, see Importing Symmetry-Reduced Models.

Accessing the SPEED Motor Tool Import Wizard

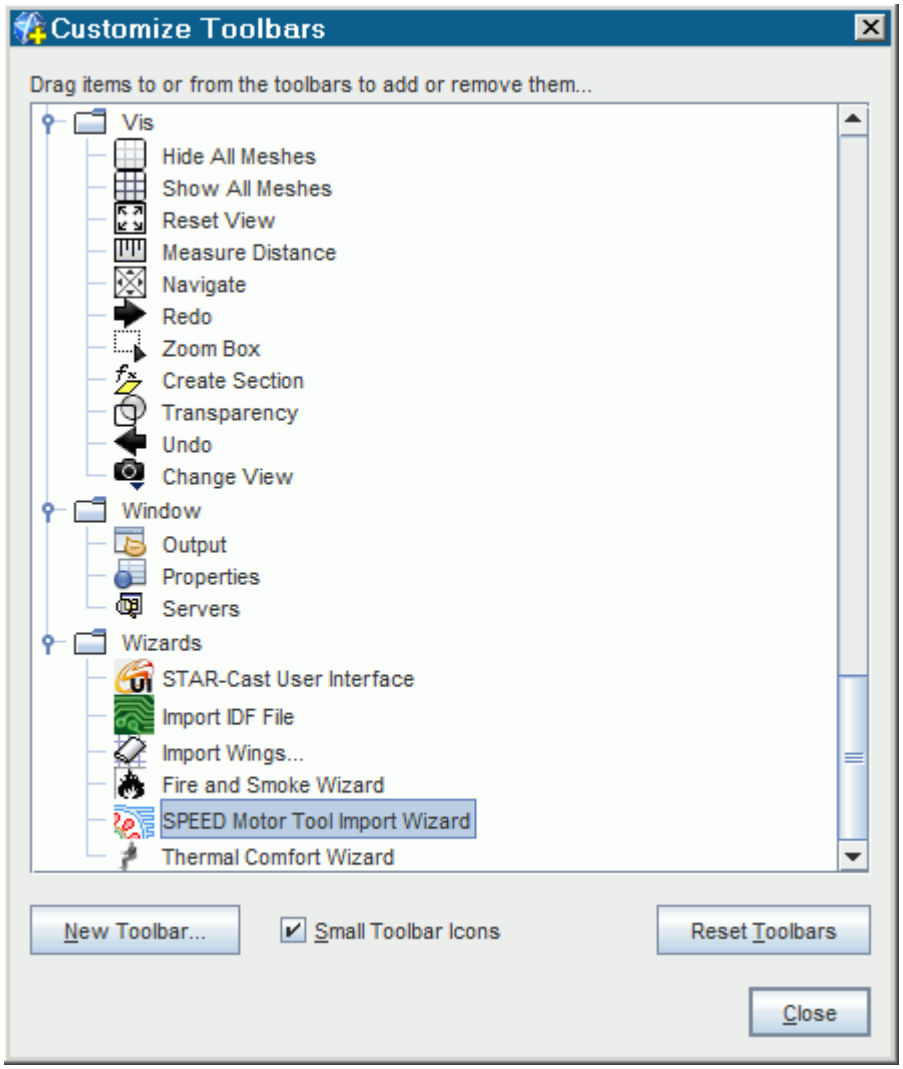
This section describes how to make the SPEED Motor Tool Import Wizard accessible in STAR-CCM+.

To make the SPEED Motor Tool Import Wizard accessible in STAR-CCM+, add the SPEED Motor Tool Import Wizard icon to the toolbar:

1. In the STAR-CCM+ menu, select **Window** > **Toolbars** > **Customize**.



2. In the *Customize Toolbars* dialog, under the **Wizards** node, select SPEED Motor Tool Import Wizard.



3. Drag the SPEED Motor Tool Import Wizard icon onto one of the toolbars at the top of the STAR-CCM+ workspace and drop it at the appropriate position.

The icon is added to the toolbar at the specified location.

4. In the *Customize Toolbars* dialog, click **Close**.

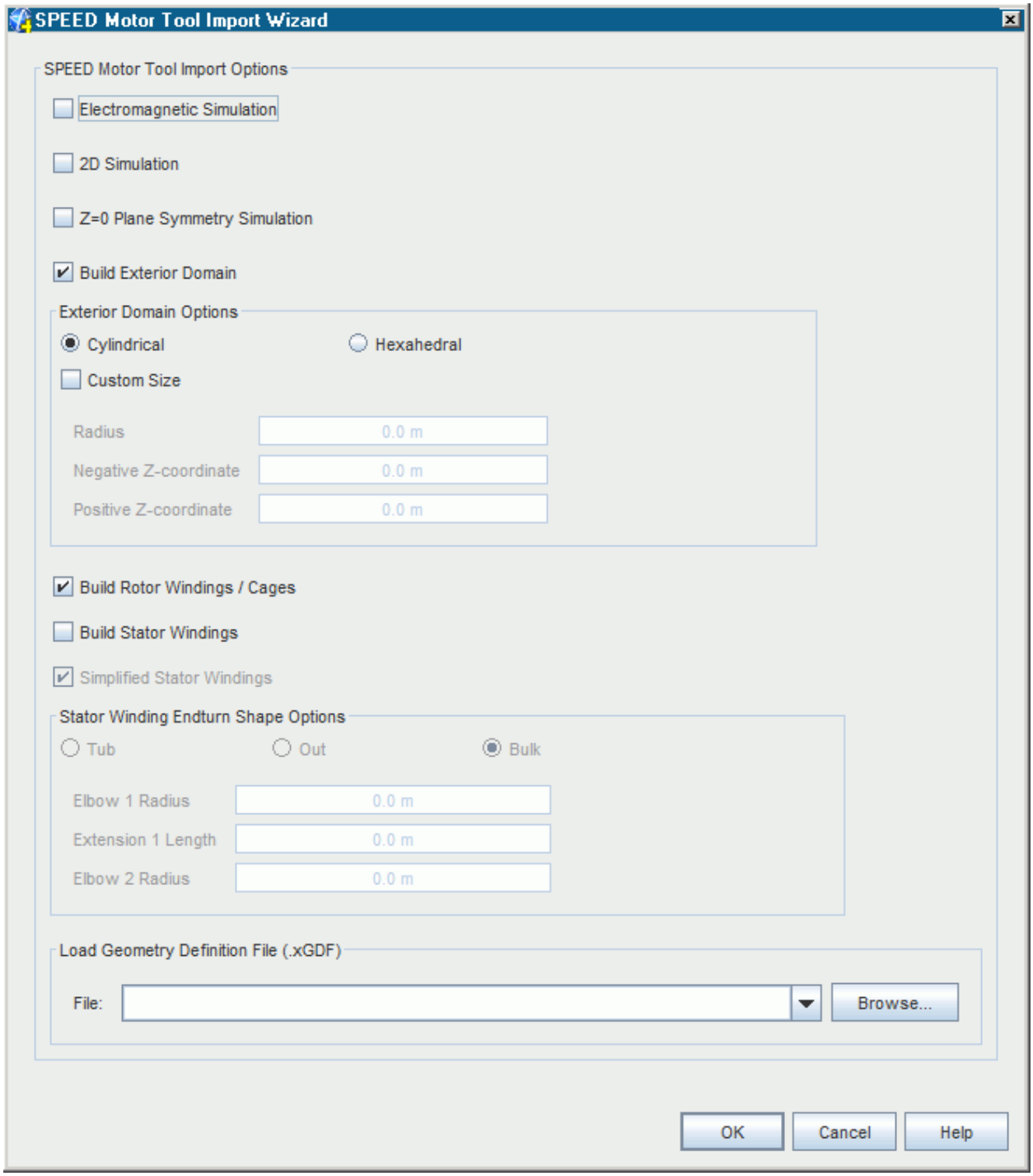
Using the SPEED Motor Tool Import Wizard

This section describes how to use the SPEED Motor Tool Import Wizard.

To use the SPEED Motor Tool Import Wizard:

1. In an open simulation, click the (**SPEED Motor Tool Import Wizard**) button in the STAR-CCM+ toolbar.

   The **SPEED Motor Tool Import Wizard** dialog appears.



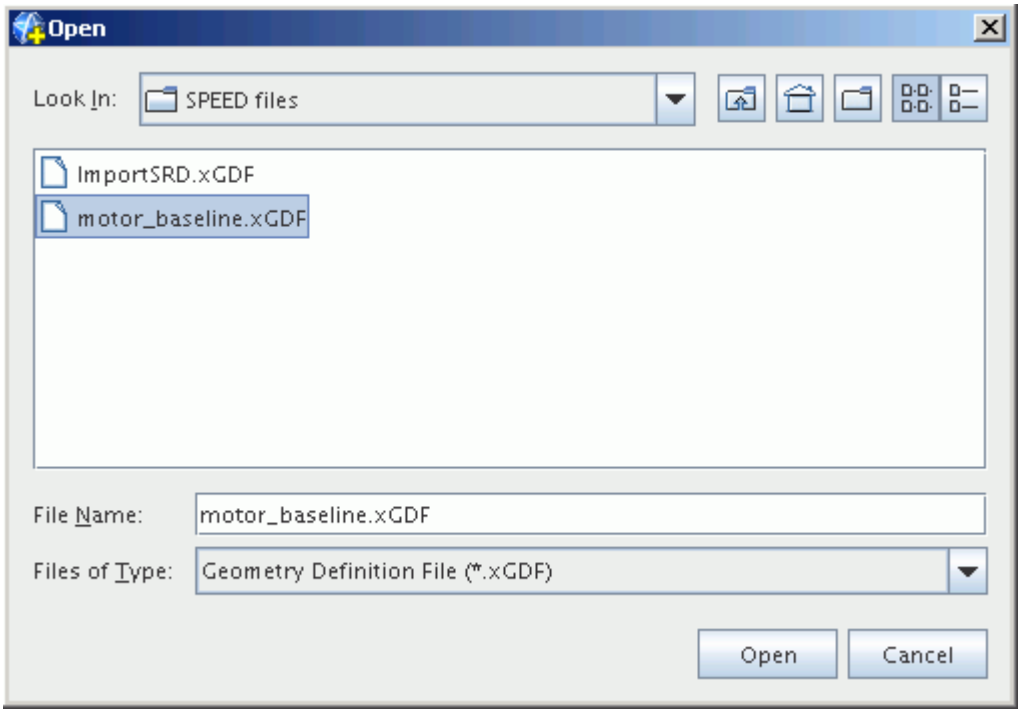
2. In the **SPEED Motor Tool Import Wizard** dialog, select the appropriate *SPEED Motor Tool Import Options*:

Electromagnetic Simulation	Specifies the simulation type. This option is deactivated by default. If you activate this option, the three-dimensional electromagnetic solver is activated. For more information, see Setting the Simulation Type.

2D Simulation	Imports the model for a 2D simulation. When this option is activated, the geometry is created as a thin slice suitable for 2D meshing. The 3D geometrical features that are specified in the `.xGDF` file, such as stator windings, rotor field windings, tilted axis eccentricity, and skew, are not imported, and no end-windings are generated. Activating this option automatically activates the Electromagnetic Simulation option. Here, the two-dimensional electromagnetic solver is used. For more information on 2D meshing, see Creating a True Two-Dimensional Mesh.
Z=0 Plane Symmetry Simulation	Creates a model that is symmetric in the axial direction. When you import an `.xGDF` file with this option activated, the full model is created in CAD. The imported models are always centered about the Z=0 plane. At the end of the CAD generation process, the model is sliced at Z= 0 and the portion on the Z-positive side is discarded. During analysis setup, the symmetry plane condition is applied at the Z=0 faces. This option applies to 3D modeling only. If selected, the **2D Simulation** option is automatically deactivated.
Build Exterior Domain	Builds an exterior domain according to the specified parameters. This option is activated by default and is required for 2D simulations. If you do not want to create an exterior domain, deactivate this option. For more information, see Setting up an Exterior Domain.
Build Rotor Windings / Cages	When this option is activated (the default setting), STAR-CCM+ creates either simple rotor field windings or a complete rotor cage, including bars and end-rings, according to the type of motor that is imported. Geometry for end-rings with fins is supported. End-ring geometry parameters are easily identified in the `.xGDF` file. For more information, see Setting up End-rings.
Build Stator Windings	When this option is deactivated (the default setting), the stator windings are not modeled in the motor geometry. If this option is activated, by default, STAR-CCM+ builds the stator windings according to the **Simplified Stator Windings** option setting. For electromagnetic simulations, windings are required. For cooling simulations, windings are optional: if you want to simplify the geometry, you can ignore the windings. However, if you include the windings, the imported windings can be modeled as simple or complex. For more information, see Setting up Windings.

Simplified Stator Windings	When this option is activated (the default setting), STAR-CCM+ builds the motor windings according to the parameters that are specified in the *Stator Winding Endturn Shape Options* group box. If this option is deactivated, the windings information is imported from the `.xGDF` file. Deactivate this option when the winding configuration is complex, such as for a mixed or chain type, single-layer scheme or for any two-layer winding scheme, including those winding schemes that use diamond windings. For more information, see Setting up Windings.

3. Under *Load Geometry Definition File*, specify the `.xGDF` file that you want to import into STAR-CCM+:
   1. Click **Browse** and then, in the *Open* dialog, select the appropriate `.xGDF` file.



   2. Click **Open** to load the selected file.
4. In the **SPEED Motor Tool Import Wizard** dialog, click **OK** to confirm the settings.

The SPEED Motor Tool Import Wizard imports the specified `.xGDF` file into STAR-CCM+.

For more information, see Using SPEED Data in STAR-CCM+.

Contents:

Setting the Simulation Type
Setting Up an Exterior Domain
Setting up Windings
Setting Up Eccentric Motors
Setting up Skewed Motors

Setting Up End-rings
Importing Symmetry Reduced Models
Magnet Coordinate Systems

Setting the Simulation Type

By default, the SPEED Motor Tool Import Wizard sets up an electromagnetic simulation. However, you can choose to set up a cooling simulation instead.

An electromagnetic simulation is a transient simulation. STAR-CCM+ turns on the relevant electromagnetic solvers in physics continua, invokes sliding mesh for rotor motion, and creates electromagnetic-specific reports and field functions.

A cooling simulation is a steady-state simulation. STAR-CCM+ invokes multiple reference frames for rotor motion.

You can specify that the simulation is a 2D simulation. When this option is enabled, the geometry is created as a thin slice suitable for 2D meshing. The 3D geometrical features that are specified in the `.xGDF` file, such as stator windings, rotor field windings, tilted axis eccentricity, and skew, are not imported, and no end-windings are generated.

Choosing a 2D simulation implies an electromagnetic simulation, so the **Electromagnetic Simulation** option is activated automatically when you select the **2D Simulation** option. You can deactivate **Electromagnetic Simulation** if desired.

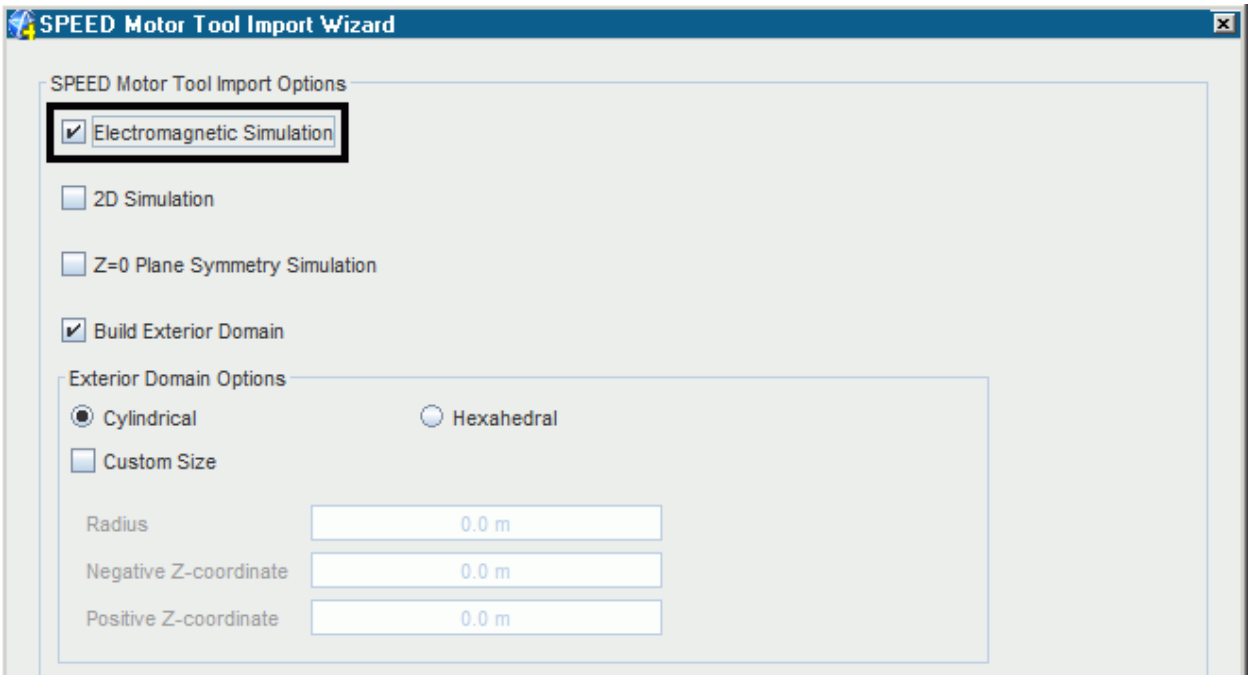
The external domain is required for 2D simulations, so the **Build Exterior Domain** option is activated automatically. You can choose **Cylindrical** or **Hexahedral** to specify the domain shape, but you cannot modify the domain size: the **Custom Size** option is disabled for 2D simulations.

Windings cannot be included in 2D simulations. When you select the **2D Simulation** option, the options for **Build Rotor Windings / Cages**, **Build Stator Windings**, and **Simplified Stator Windings** are disabled.

Note: If the stator windings are complex and of the Full, Side-by-Side (two-layer) type, deactivate the **Simplified Stator Windings** option before you select the **2D Simulation** option. This setting is required to split the “Slot” bodies to accommodate the winding configuration.

To set the simulation type:

1. Under *SPEED Motor Tool Import Options*, activate or deactivate **Electromagnetic Simulation**, whichever is appropriate.



2. If you want to specify a 2D simulation, activate **2D Simulation**.

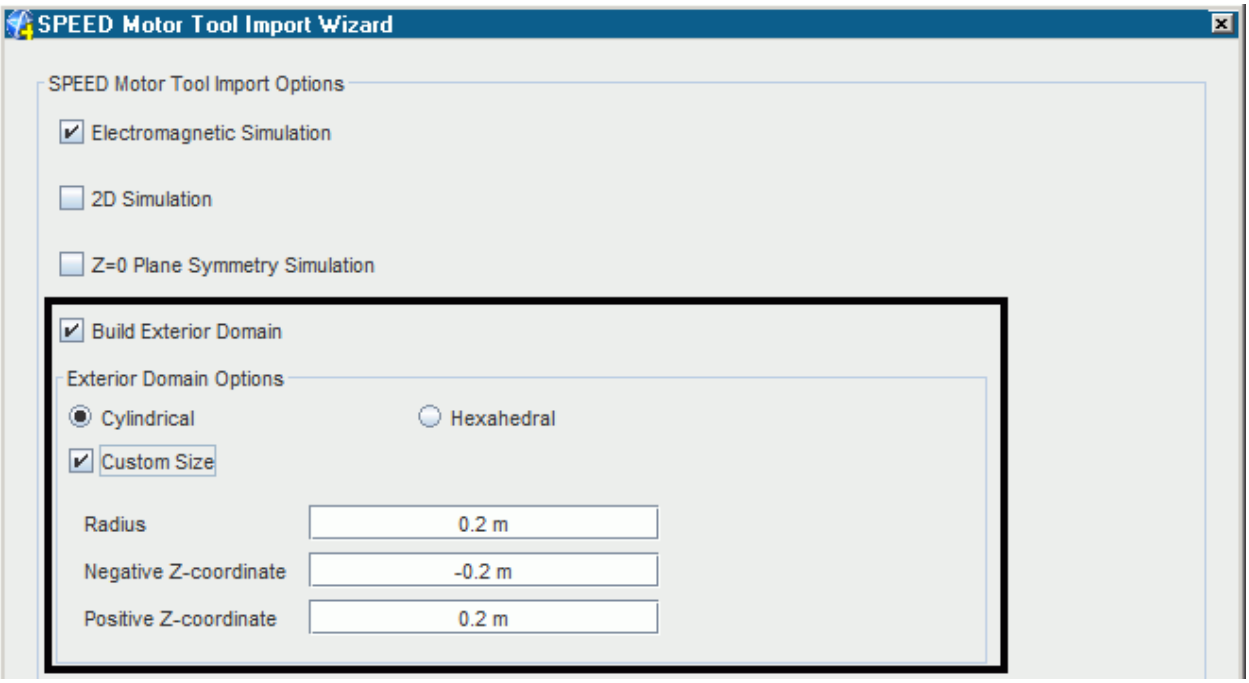
Setting Up an Exterior Domain

The .xGDF file specifies the motor geometry only: it does not include any exterior domain information. You can let the SPEED Motor Tool Import Wizard build an exterior (air flow) domain as part of the import process.

Alternatively, you can build an exterior domain manually in STAR-CCM+ as a separate action.

To set up an exterior domain as part of the import process:

1. Under *SPEED Motor Tool Import Options*, select **Build Exterior Domain**.



2. Specify the shape of the domain by selecting the appropriate option: **Cylindrical** or **Hexahedral**.

   The cross-section of the exterior domain can be circular or square. If circular, the resulting 3D shape is cylindrical. If square, the resulting 3D shape is planar, hexahedral.

   The default dimensions are automatically calculated to enclose the entire motor, including any endwindings. If necessary, you can modify the created domain in STAR-CCM+.

3. If you want to make the exterior domain larger than the default size, select **Custom Size** and then set the appropriate dimensions:

Radius	The radius of the cylinder, or the X and Y dimensions of the hexahedron.
Negative Z-coordinate	The Z dimension in the negative direction. The specified value must include the negative sign.
Positive Z-coordinate	The Z dimension in the positive direction.

   The dimension values must include the units.

The SPEED Motor Tool Import Wizard ensures that the size of the exterior domain is large enough to enclose the motor, including the endwindings. If you specify a dimension that is too small, it is corrected automatically when the .xGDF file is imported into STAR-CCM+. The *Output* window displays details of any corrections that are made during the import process.

Setting up Windings

The `.xGDF` file can include both rotor and stator winding information from the SPEED winding editor. You can choose to import this information or ignore it. For stator windings, you can create simplified windings instead.

There are three classes of windings: Bulk, Simple, and Complex. Bulk and Simple windings are simplified representations of the windings. STAR-CCM+ imports these windings as follows:

- Bulk windings are imported into a single region named **Windings**.

  The appropriate thermal conductivity direction field is created automatically and is set as the axis for the winding anisotropic thermal conductivity. For more information, see Bulk Windings Thermal Conductivity.
- Simple windings are imported into a separate region for each winding. These regions have simple names such as **Winding 1** and **Winding 2**.
- Complex windings are imported into a separate region for each winding. These regions have detailed names that include the phase number and the coil number that are taken from the `.xGDF` file winding data.

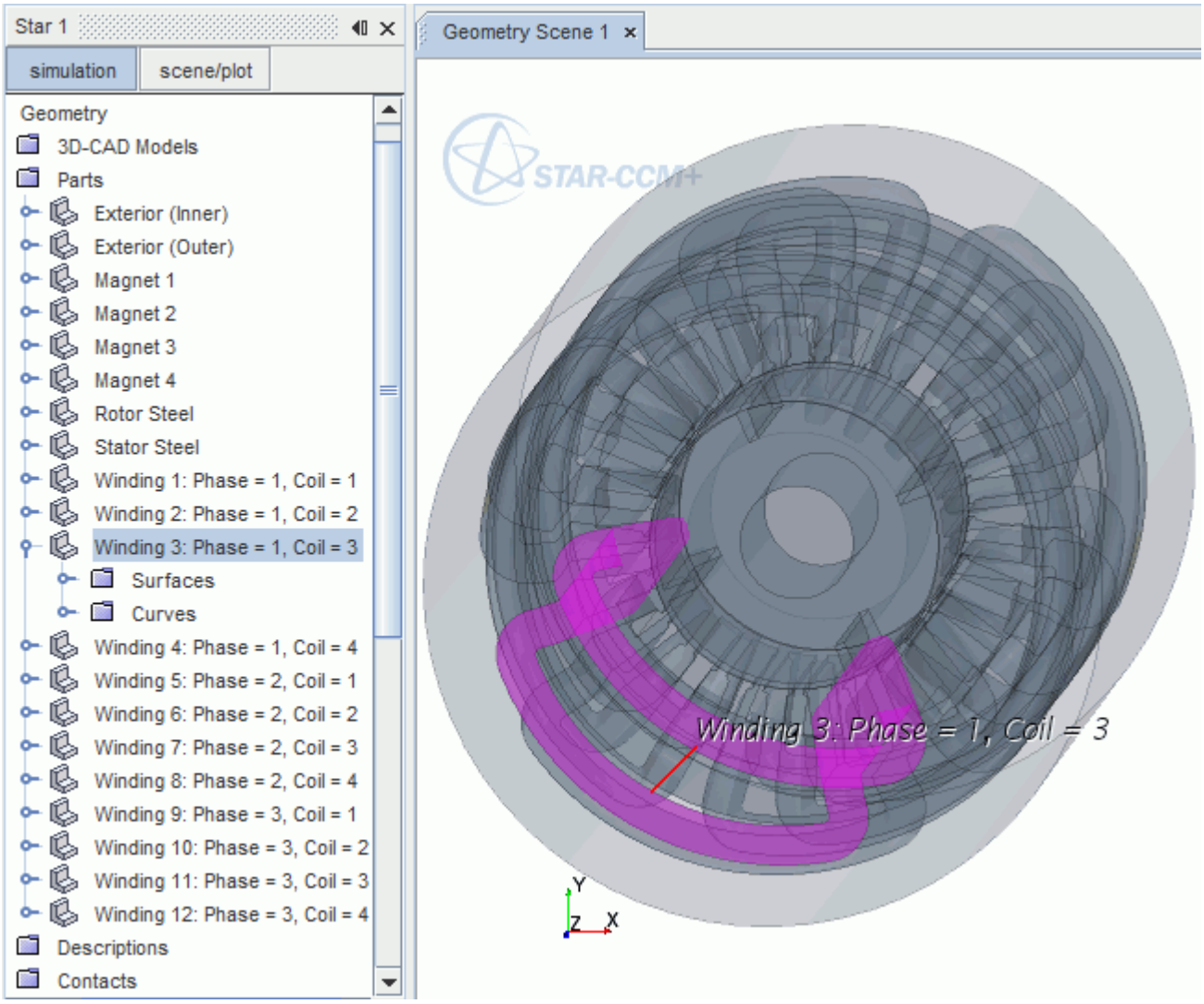
STAR-CCM+ can import the following types of windings from the `.xGDF` file:

Stator	Simple windings Single coil side type Complex windings Any two-layer coil side type Complex windings, including those windings that use diamond endwindings.
Rotor	For Induction motors, the rotor field windings are imported as Bulk Tub-type windings. For all other motors, the rotor field windings are imported as Simple Tub-type windings (separated windings).

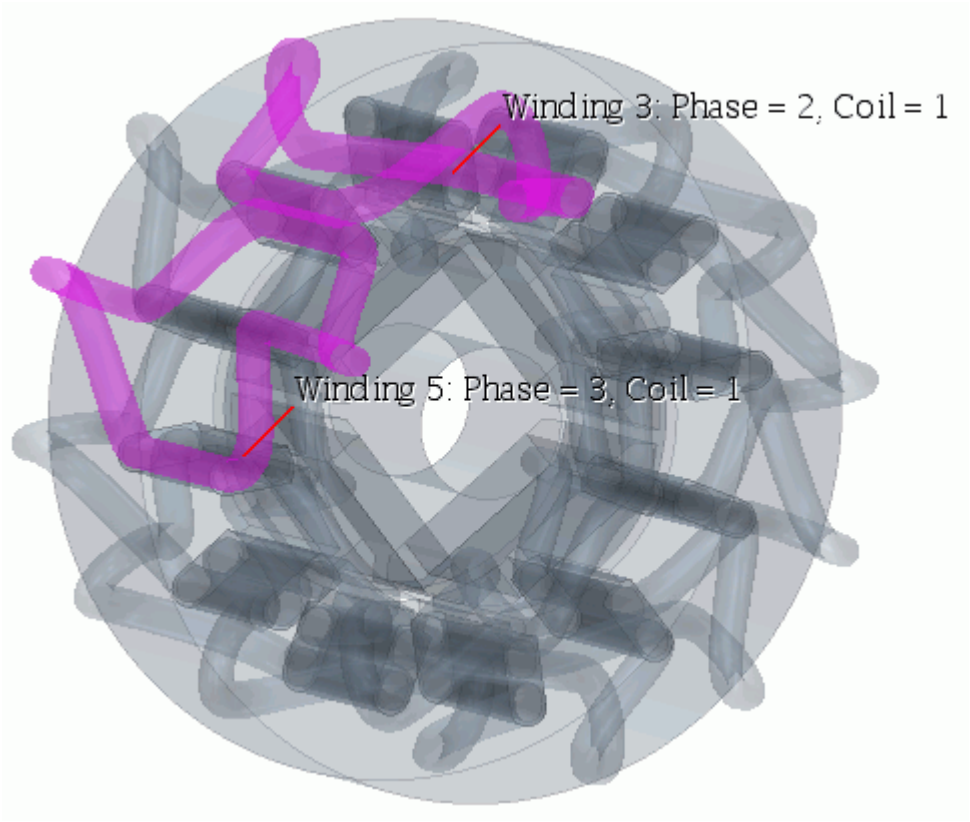
Any other types of complex windings that are specified in the `.xGDF` file cannot be imported into STAR-CCM+.

To include windings in the imported model, activate the appropriate setting in the *SPEED Motor Tool Import Wizard* dialog.

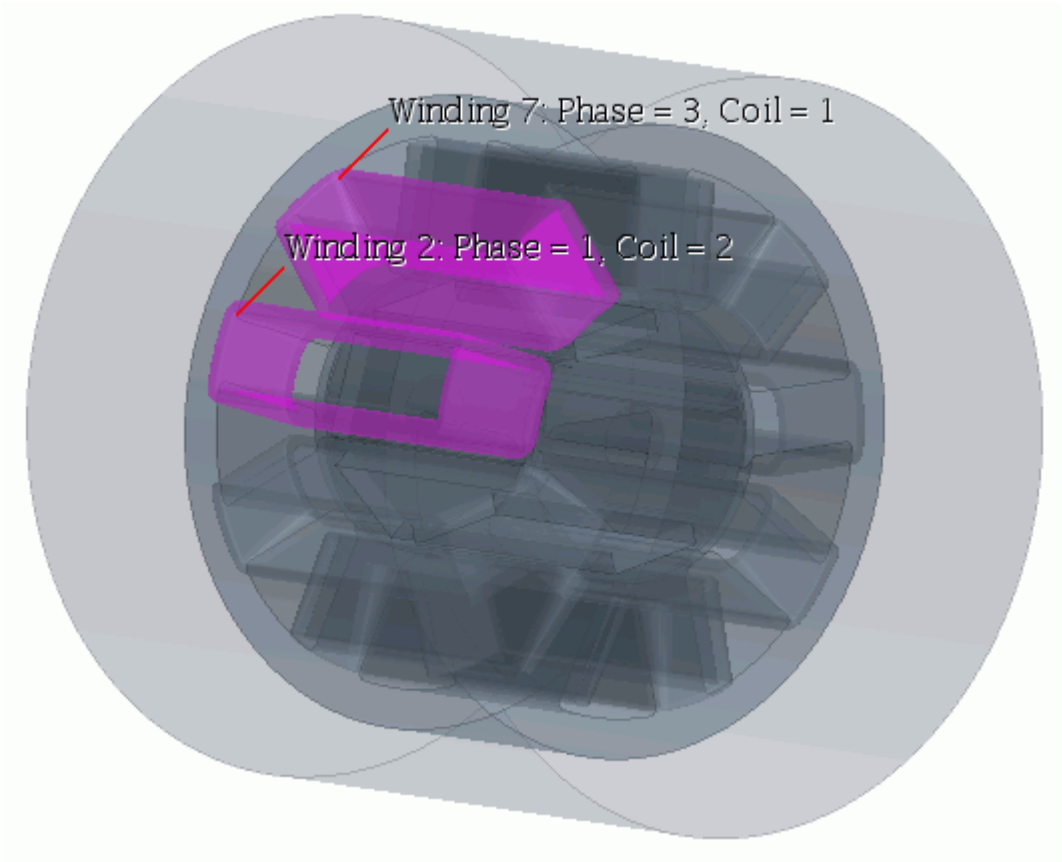
An example of a motor with single coil side type complex windings imported from a `.xGDF` file is shown below:



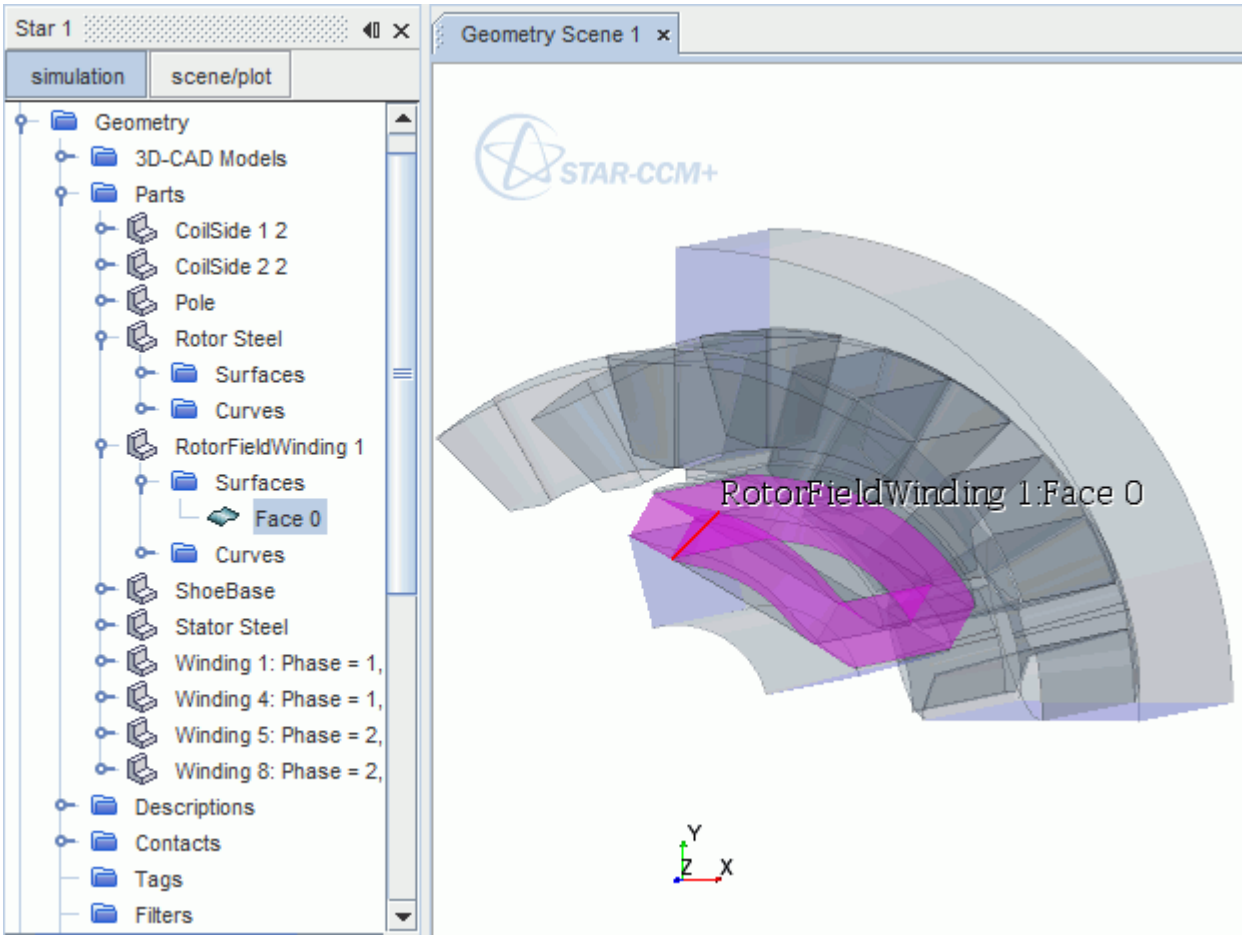
An example of a motor with two-layer top-bottom coil side complex windings imported from a `.xGDF` file is shown below:



An example of a motor with two-layer side-side coil side complex windings imported from a `.xGDF` file is shown below:



An example of a non-induction type motor sector model with rotor windings imported from a `.xGDF` file is shown below:



If you want to create simplified windings, specify the appropriate parameters. The SPEED Motor Tool Import Wizard lets you specify simple stator windings only: the endturn span is always one, that is, the winding connects adjacent slots. An endturn span of one is typical for PC-SRD motors. Rotor windings and complex stator windings, where the endturn span is greater than one (that is, the winding spans multiple slots), cannot be specified in the SPEED Motor Tool Import Wizard - they must be imported.

If you specify simplified windings, the Slot (PC-BDC) or Coil (PC-SRD) regions are removed from the imported geometry. Winding regions are added in the appropriate location.

For electromagnetic simulations, windings are required. For cooling simulations, windings are optional. However, if you include the windings, the imported windings can be modeled as "simple" or "complex". Under the **Simplified Stator Windings** setting, you can specify either **Tub**, **Out**, or **Bulk** windings. For cooling simulations, simple windings are recommended, using the **Bulk** option.

To set up windings:

1. Under *SPEED Motor Tool Import Options*, select the appropriate options:
   - **Build Rotor Windings / Cages**

     When this option is activated (the default setting), STAR-CCM+ creates either rotor field windings or a rotor cage according to the type of motor that is imported.

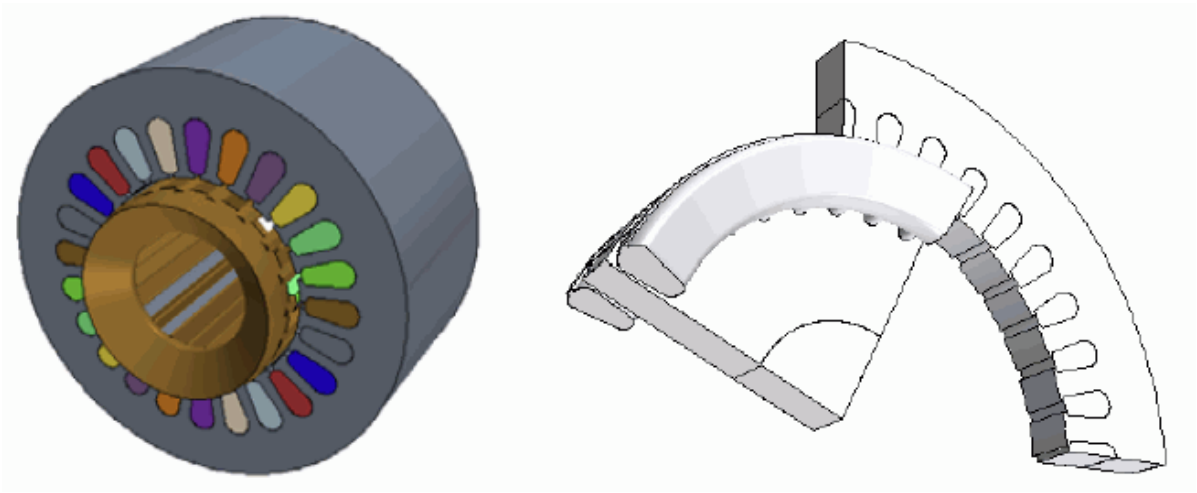
For induction motors imported from PC-IMD which have solid bodies that are designated as “Cage…”, the end-ring geometry for the cage is created. STAR-CCM+ creates the geometry according to the dimensions of the “Cage” bodies, and you cannot modify the shape of the end-rings.

For other motor types, such as those imported from PC-BDC, when motors have solid bodies that are designated at “Field Coil”, the endturn geometry for the field coils is created. STAR-CCM+ creates the geometry according to the dimensions of the “Field Coil” bodies, and you cannot modify the shape of the rotor field winding endturns.

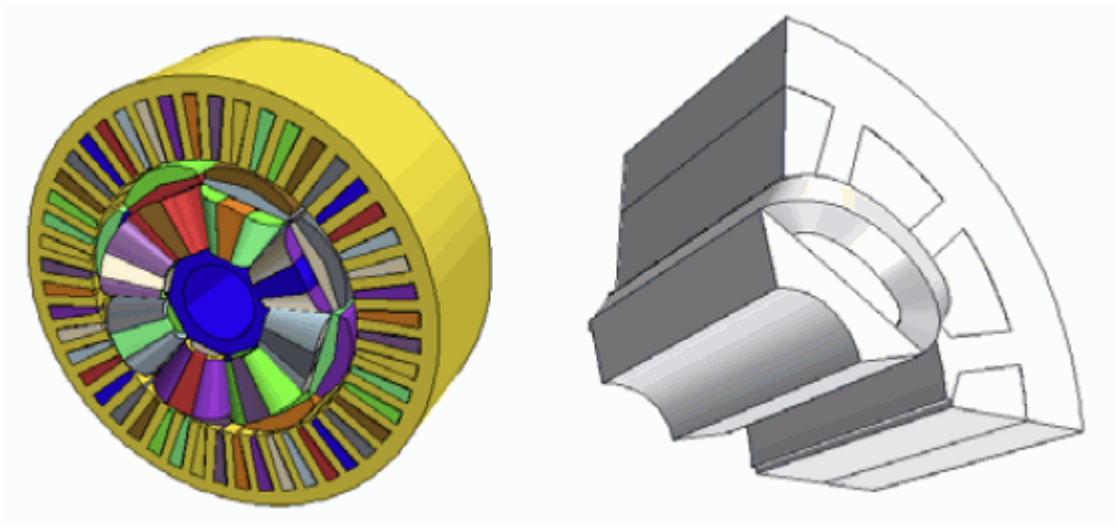
In both cases, the only information that is imported from the `.xGDF` file is the 2D cross-section starting from the rotor. The `.xGDF` file does not supply any information for the geometry of either the cage end-rings or rotor field winding endturns.

Examples of each case are shown below.

Cage end-rings created from “Cage 1” solid bodies for induction motors (PC-IMD):



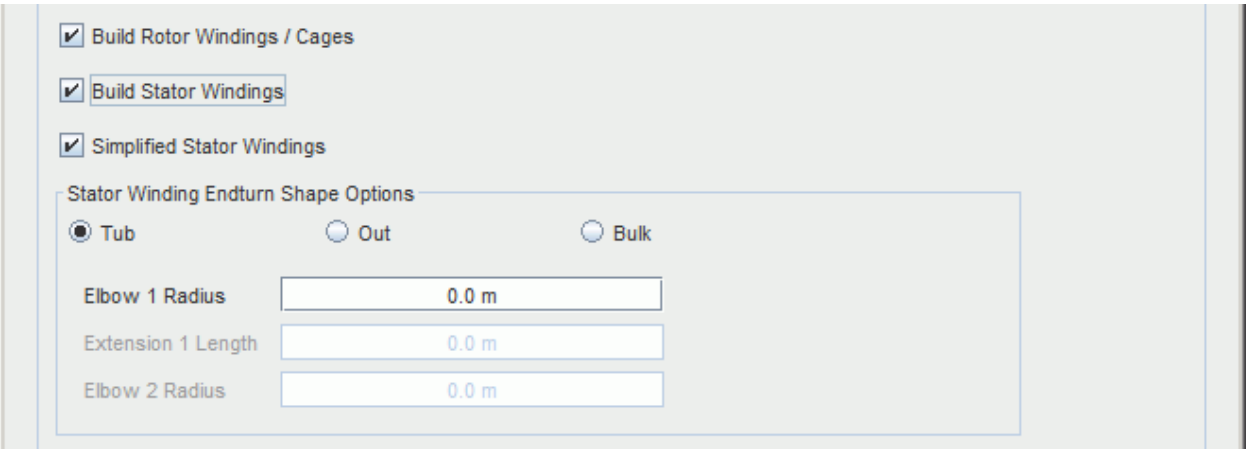
Rotor field windings created from “Field Coil” solid bodies for brushless, DC motors (PC-BDC):



- **Build Stator Windings**

  When this option is deactivated (the default setting), the stator windings are not modeled in the motor geometry.

  If this option is activated, STAR-CCM+ builds the stator windings according to the **Simplified Stator Windings** option setting.



If this option is deactivated, the windings are not modeled in the motor geometry.

2. Set the **Simplified Stator Windings** option as appropriate:
   - If this option is deactivated (the default and recommended setting for electromagnetic simulations), the windings information is imported from the `.xGDF` file.

     STAR-CCM+ automatically calculates the axial extensions (the straight part of the coil between the stator and elbow) so that the windings do not interfere.
   - When this option is activated, STAR-CCM+ builds the motor windings according to the parameters that are specified in the *Stator Winding Endturn Shape Options* group box.

     For an electromagnetic simulation, any windings information in the `.xGDF` file is ignored.

     For a cooling simulation, the **Simplified Stator Windings** setting specifies the general shape of the endwindings.
3. Under *Stator Winding Endturn Shape Options*, specify the winding endturn shape by selecting the appropriate option: **Tub**, **Out**, or **Bulk**.

   If **Electromagnetic Simulation** is deactivated, the default and recommended option is **Bulk**, although all other winding options are available. Using the **Bulk** option, the individual coils in the stator are melded into one body for the endwindings, using the tub-type endturn shape. In general, bulk-tub shaped windings provide a better description of the true winding geometry. However, to better model the correct volume covered by true winding geometry, the winding data from the `.xGDF` file is used to modify the thickness of the bulk-type winding body (that is, the toroidal thickness).

   For more information, see Cooling Simulation Endwindings.

   If **Electromagnetic Simulation** is activated, the **Tub** type endturn shape becomes the default option. The **Bulk** type option is not allowed.
4. Set the appropriate dimensions:

Elbow 1 Radius	This parameter applies to both winding shapes.
Extension 1 Length	The straight extension from elbow 1 to elbow 2. This parameter applies to **Out** windings only.
Elbow 2 Radius	This parameter applies to **Out** windings only.

The dimension values must include the units.

The SPEED Motor Tool Import Wizard automatically calculates a minimum and maximum set of values, which are based on the motor geometry.

The minimum dimensions require minimum physical space while ensuring, in most cases, that the endwinding does not interfere with itself. The maximum dimensions ensure that the endturn does not go beyond the adjacent slot, as the span must equal one.

You can obtain the minimum dimensions by entering `0` for each parameter. You can obtain the maximum dimensions by entering a large number, such as `1000000`.

The maximum value for some parameters can be unrealistic. The *Output* window displays details of any corrections that are made during the import process.

Contents:

Setting up Diamond Endwindings
Cooling Simulation Endwindings

Setting up Diamond Endwindings

The SPEED Motor Tool Import Wizard lets you import diamond endwinding information into STAR-CCM+.

When you specify diamond endwindings in SPEED, the winding pattern and the parameters that define the endwinding geometry are included in the `.xGDF` file. For diamond endwindings, the endwinding type (`<EWtype>`) is `3`.

Four geometric parameters are used to create a diamond endwinding:

ext	Specifies the length of the straight extension beyond the stator.
height	Specifies the total length from the stator to the knuckle end.
knuckleradius	Specifies the radius of the knuckle.
alpha	Specifies the angular incline of the endwinding in degrees. This value depends on the "Go" coil.

These parameters are illustrated below:

The diamond winding parameters that you specify in SPEED are imported directly from SPEED into STAR-CCM+ as is. The spatial domain of the SPEED program is limited to 2D. Therefore, SPEED cannot determine or guarantee, before the file is imported, whether the imported winding dimensions cause

construction problems in 3D. When you create diamond windings, you require a clear understanding of the geometry that is involved as well as the parameters that are used to define it.

Note the following:

- The straight extension can be zero (that is, `<ext>` ≥ 0).
- The knuckle radius can be zero (that is, `<knuckleradius>` ≥ 0). Setting this parameter to zero results in the smallest allowable radius that is based on the coil cross-sectional area.
- Choose a reasonable height. Too small a value results in either construction failure when creating the initial winding or winding interference between two or more windings. For compact windings, a different winding type can be more suitable. There is no limit on how large the height can be.
- The incline angle, `<alpha>`, can have a negative, zero, or positive value. Typically, the angle is zero or greater to avoid interference with rotor components.

Diamond endwindings are specified in the `.xGDF` file as illustrated in the following example:

```
        <coil number="4">
          <Go> 4 </Go>
          <Return> 2 </Return>
          <Span> -2 </Span>
          <Turns> 12 </Turns>
          <GoLoc> 1 </GoLoc>
          <RetLoc> 2 </RetLoc>
          <EWtype> 3 </EWtype>
          <EWset> 1 </EWset>
        </coil>
      </phase>
    </coillist>
  <ewparams>
    <ewparam set="1" >
      <ext> 5 </ext>
      <height> 30 </height>
      <knuckleradius> 5 </knuckleradius>
      <alpha> 10 </alpha>
    </ewparam>
  </ewparams>
</winding>
```

If you want to modify the endwinding parameters, edit the `.xGDF` file before you import it.

Note: Any changes that you make to the `.xGDF` file are at your own risk. The parameter values are imported into STAR-CCM+ as is, and are not checked or corrected.

Cooling Simulation Endwindings

For a cooling simulation, the imported windings are modeled as Bulk-type windings in the motor geometry.

The **Simplified Stator Windings** setting specifies the general shape of the endwindings: either Bulk-Tub or Bulk-Out shaped windings.

The thickness of the endwindings toroid in the imported model is defined as follows:

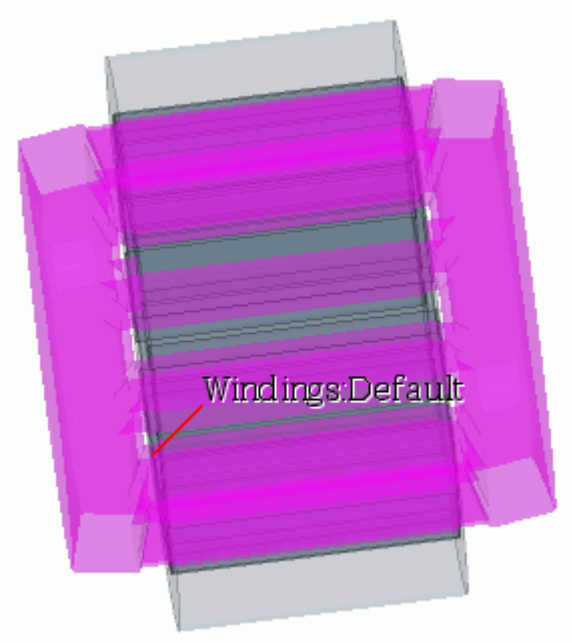
- If no windings information is available in the `.xGDF` file, the endwinding thickness is set equal to the thickness of the slot face sketch.
- If windings information is available in the `.xGDF` file, the thickness of the endwindings toroid is calculated as:

  $$\frac{1}{2} \times \text{maximum span} \times \text{coil thickness}$$

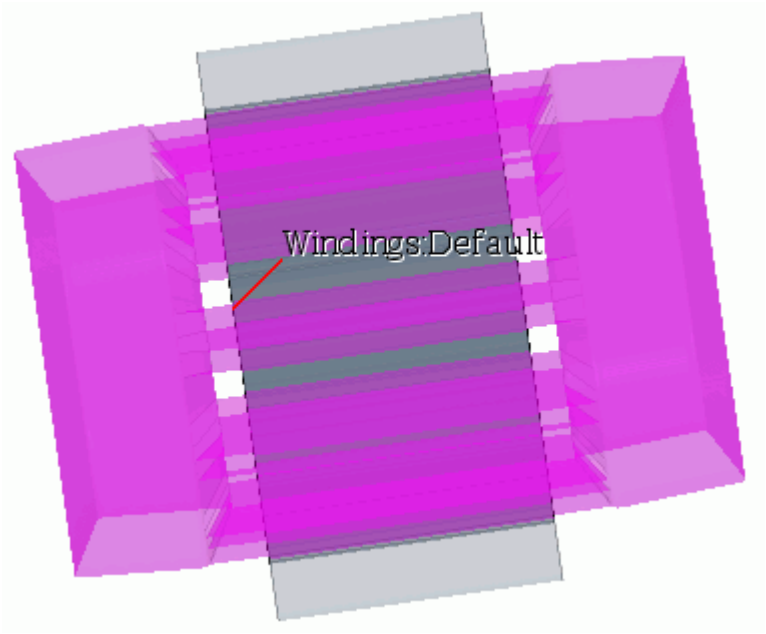
  For example, if the maximum span is 10, the endwindings toroid thickness is 5 times the thickness of the slot or coil face sketch from which the winding originated. If no explicit coil face is provided, the slot is assumed to be single layered, meaning one coil per slot.

  The span value is a non-zero integer and can be negative, so the absolute value is used. If the maximum span is 1 or 2, the thickness of the endwindings toroid is set to the coil thickness.

The model that is shown below has the endwindings toroid that is calculated for a maximum span of 3.



The example below shows the same model with the endwindings toroid calculated for a maximum span of 6.



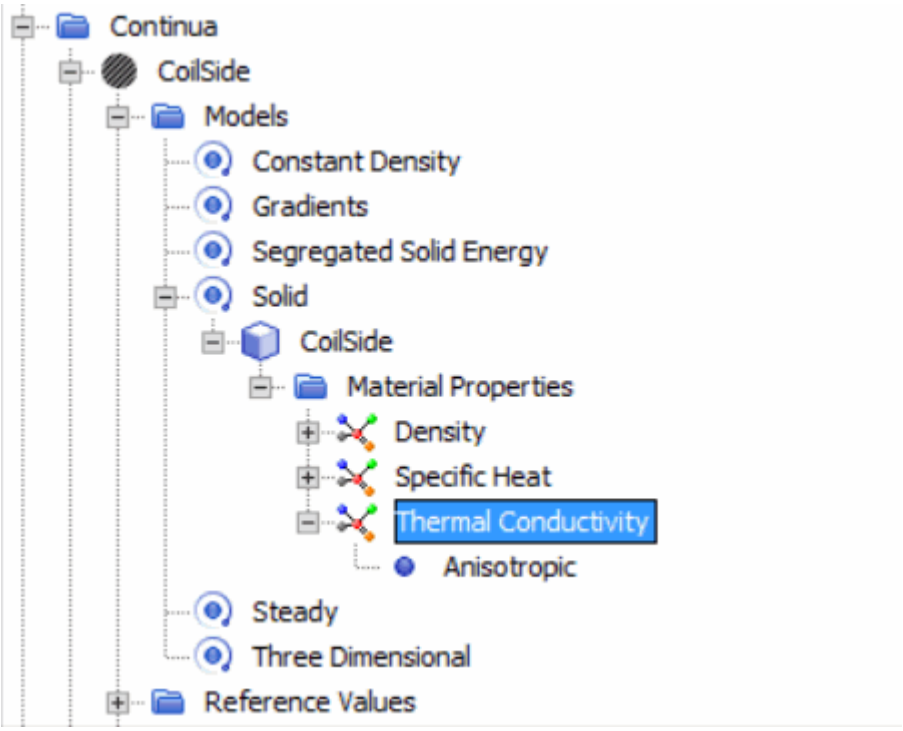
Bulk Windings Thermal Conductivity

For thermal modeling of electric machines, STAR-CCM+ has the ability to define complex anisotropic thermal conductivity that realistically models the thermal conductivity of electric machine windings.

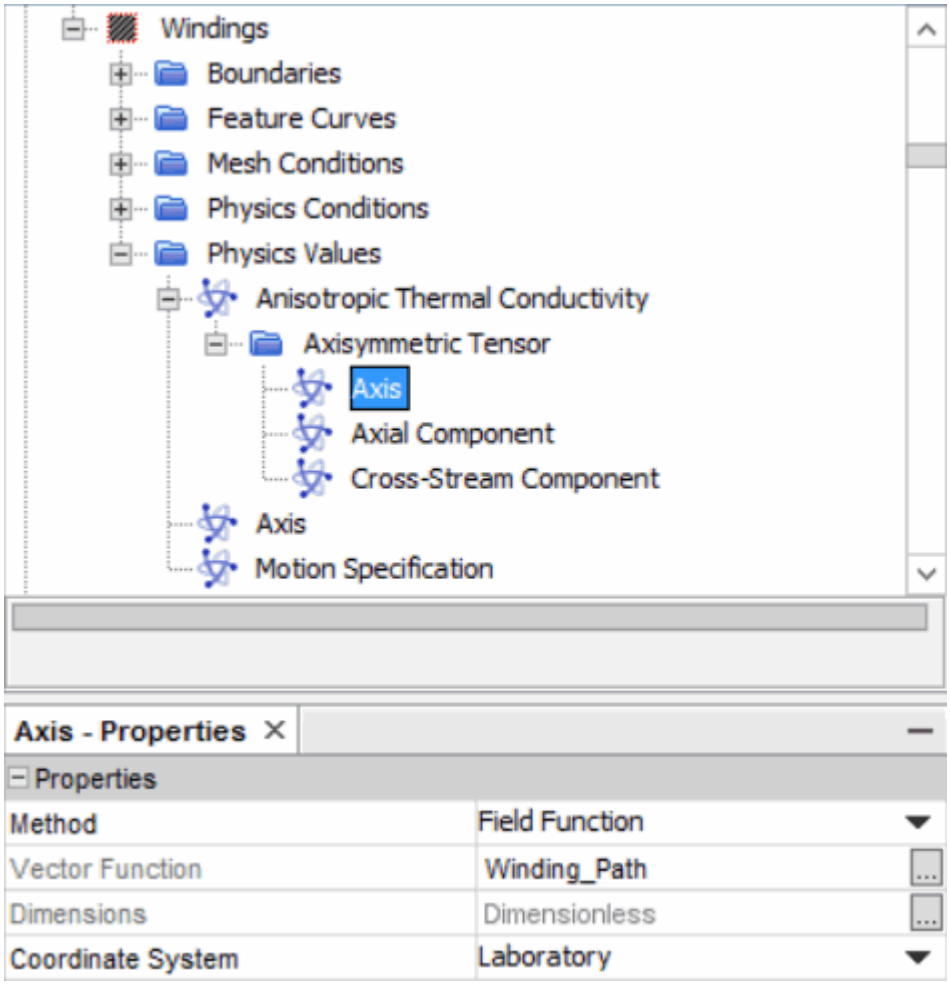
When you import Bulk-type windings, the appropriate thermal conductivity direction field is created automatically and is set as the axis for the winding anisotropic thermal conductivity. This feature applies to the PC-BDC type of motor.

An example is illustrated below:

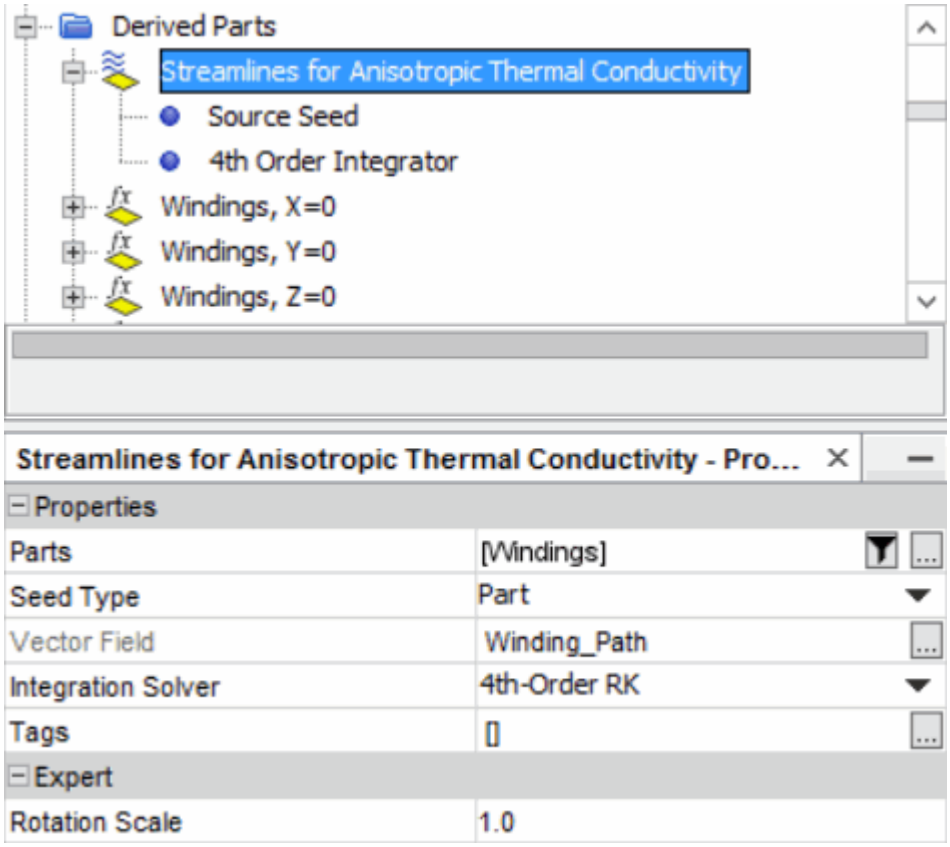
- In the physics continua that corresponds to the windings region, the *Thermal Conductivity* solid material property is set to anisotropic.



- In the windings region, the **Anisotropic Thermal Conductivity** is specified as an axisymmetric tensor with the *Axis* set to the `Winding_Path` field function. STAR-CCM+ creates this vector field function automatically when the `.xGDF` file is imported.

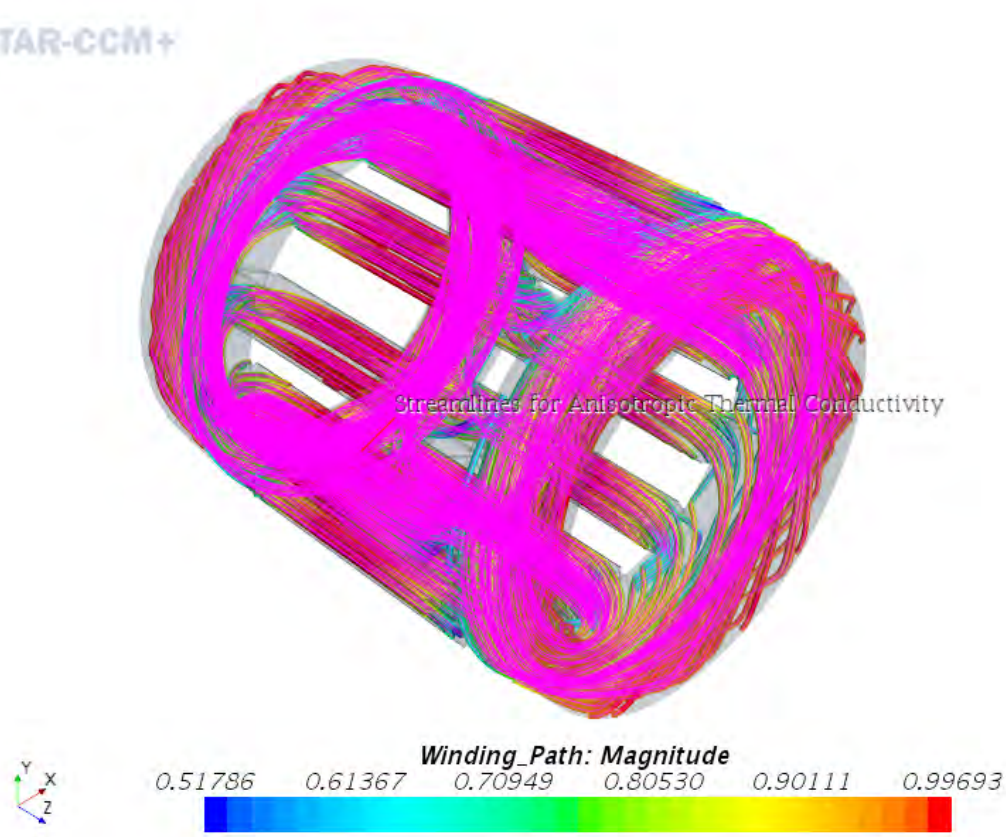


- The **Streamlines for Anisotropic Thermal Conductivity** derived part is created automatically.



- The **Anisotropic Thermal Conductivity** scene is also created.

  The example below shows the scene with streamlines created.



Setting Up Eccentric Motors

The SPEED Motor Tool Import Wizard lets you import eccentricity information into STAR-CCM+.

If you design an eccentric (rotor off-center) motor in SPEED, the eccentricity information is included in the `.xGDF` file. The SPEED Motor Tool Import Wizard lets you import the eccentricity information into STAR-CCM+. However, if you want to modify the eccentricity parameters, you can edit the `.xGDF` file before importing it.

Note: Any changes that you make to the `.xGDF` file are at your own risk. The eccentricity values are imported into STAR-CCM+ as is, and are not checked or corrected.

Eccentricity is specified in the `.xGDF` file as follows:

```
<eccentricity>
     <mode> 0 </mode>
     <dx> 0 </dx>
     <dy> 0 </dy>
     <dz> 0 </dz>
     <dthx> 0 </dthx>
```

```
    <dthy> 0 </dthy>
  </eccentricity>
```

The parameters are as follows:

mode	Specifies the eccentricity mode. Valid values are `0` (static) or `1` (dynamic). Static eccentricity is a simple lateral displacement of both the rotor and the axis of rotation of the rotor away from the global Z-axis. Dynamic eccentricity is when the centroid of the rotor is displaced from its axis of rotation. The rotor wobbles around its axis of rotation, which is the global Z-axis.
dx, dy, dz	Specifies the lateral displacement of the rotor about the global origin (0, 0, 0). SPEED motors are centered about the global origin. Non-eccentric motors rotate about the global Z-axis. There are no restrictions on lateral displacements, and they can be used with a tilt displacement.
dthx, dthy	Specifies the X-tilt displacement and Y-tilt displacement of the rotor. The axis of rotation of the rotor is not parallel to the axis of the stator. In this configuration, the air gap profile at one end of the machine is different to the profile at the other end and the model includes axial forces as well as radial forces. X-tilt displacement rotates the rotor about the X-axis of the global origin (up or down), using the right-hand rule. Y-tilt displacement rotates the rotor about the Y-axis of the global origin (left or right), using the right-hand rule.

Setting up Skewed Motors

SPEED lets you design skewed motors, where either or both the rotor and the stator are twisted along the rotation axis. Skew information is included in the `.xGDF` file.

The SPEED Motor Tool Import Wizard supports importing continuous skew settings and stepped skew settings into STAR-CCM+. The imported parts are separated into the appropriate regions, according to the skew mode:

- Skew mode 1 (continuous): One region per part.
- Skew mode 2 (stepped): One unique region per slice (step).
- Skew mode 3 (V-continuous): Two regions per part, split symmetrically.
- Skew mode 4 (V-stepped): One unique region per slice (step).

If you want to modify the skew parameters, edit the `.xGDF` file before you import it.

Note: Any changes that you make to the `.xGDF` file are at your own risk. The skew values are imported into STAR-CCM+ as is, and are not checked or corrected.

Skew is specified in the `.xGDF` file as illustrated in the following example:

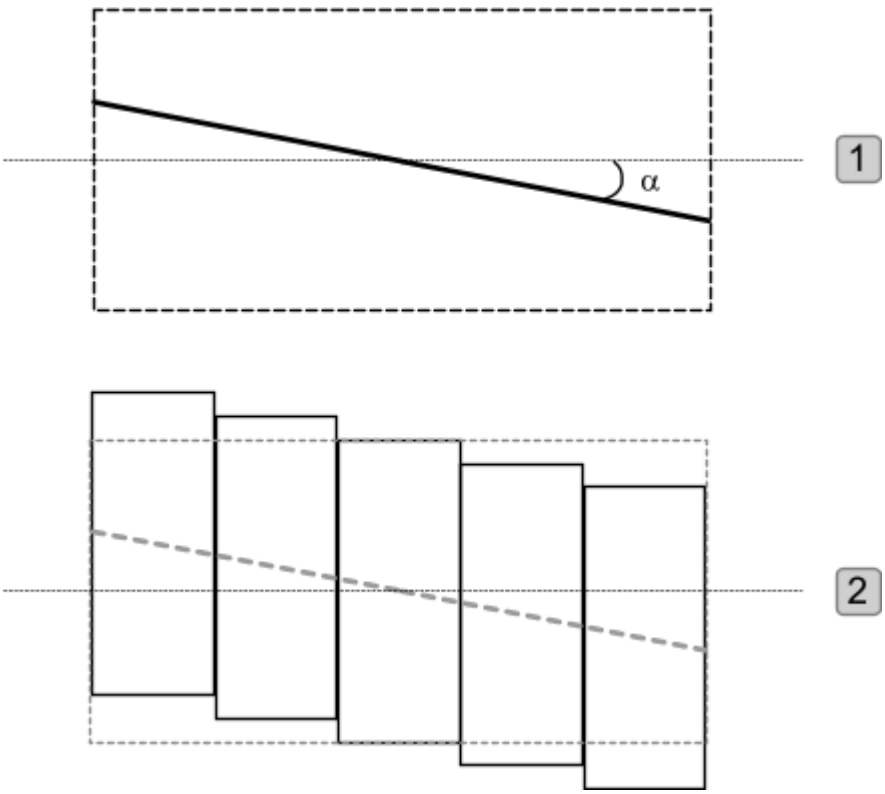
```
<skew>
  <rotor>
    <mode> 1 </mode>
    <angle> 30 </angle>
    <slices> 2 </slices>
    <sections> 1 </sections>
    <symmetry> 0 </symmetry>
  </rotor>
  <stator>
    <mode> 0 </mode>
    <angle> 0 </angle>
    <slices> 1 </slices>
    <sections> 1 </sections>
    <symmetry> 0 </symmetry>
  </stator>
</skew>
```

The parameters are as follows:

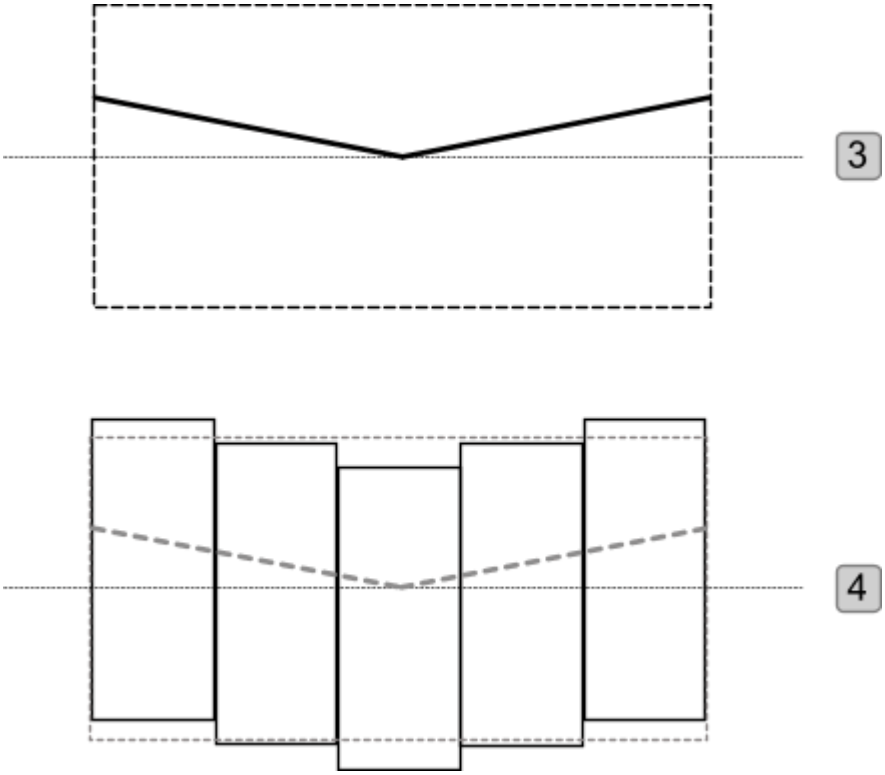
rotor	Specifies the skew settings for the rotor.
stator	Specifies the skew settings for the stator.
mode	Specifies the skew mode. Valid values are: • `0` - none • `1` - linear • `2` - stepped • `3` - V-linear • `4` - V-stepped
angle	Specifies the skew angle about the Z-axis of the global origin, using the right-hand rule. The angle can be negative or positive. Note the following: • if mode = `0`: the angle = `0` and is ignored • if mode = `1` or `2`: the angle is across the full length of the section. • if mode = `3` or `4`: the angle is across half the length of the section.
slices	Specifies the number of steps in a stepped skew model: • if mode = `0`, `1`, or `3`: the number of slices = `1` and is ignored. • if mode = `2` or `4`: the number of slices = the number of steps across the length of the section.
sections	Specifies the number of repeatable units of the model, as described by the mode, angle, and slices settings.

symmetry	Specifies the symmetry of the sections when the model contains two or more sections. The valid values are: • `0` - each section is a repeat of the previous section. • `1` - each section is a mirror image of the previous section. If sections = `1`, symmetry = `0`.

The following diagrams illustrate the settings that you can make for each skew mode. Each example shows a single section, and the skew angle α is the same for each.



1	Skew mode is Linear; skew angle is α.
2	Skew mode is Stepped; skew angle is α; number of slices is 5.



3	Skew mode is V-Linear; skew angle is α.
4	Skew mode is V-Stepped; skew angle is α; number of slices is 5.

Setting Up End-rings

The SPEED Motor Tool Import Wizard lets you import end-ring information into STAR-CCM+.

When you design an induction motor in SPEED, the end-ring information is included in the `.xGDF` file. However, if you want to modify the end-ring parameters, you can edit the `.xGDF` file before importing it.

If you edit the end-ring parameters, use caution and note the following:

- No dimension parameters can be negative.
- A minimum tolerance distance of 0.1 mm is used for some parameters where a zero value would cause the geometry creation to fail.
- If end-ring fins are not desired, set <numrfins> to zero, rather than alter the fin dimensions.
- A cage must have two end-rings but can use two different end-ring types. End-ring fins can also differ from one end-ring to the other.

Note: Any changes that you make to the `.xGDF` file are at your own risk. The end-ring values are imported into STAR-CCM+ as is, and are not checked or corrected.

End-rings are specified in the `.xGDF` file as follows.

Type A, B, and C end-ring parameters:

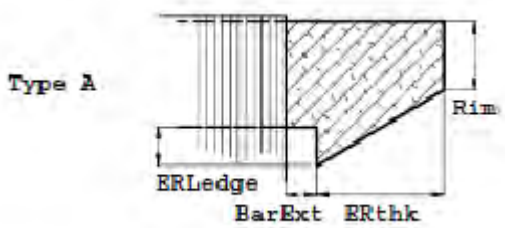
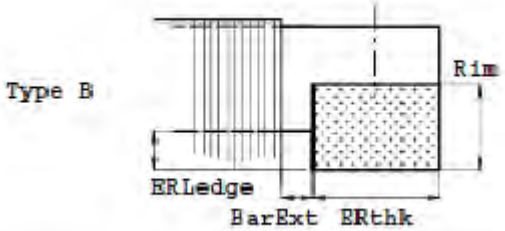
```
<endrings>
    <endring1>
      <barext> 4 </barext>
      <ertype> 0 </ertype>
      <erthk> 8 </erthk>
      <erledge> 3 </erledge>
      <rim> 6 </rim>
      <numrfins> 20 </numrfins>
      <rfinthk> 2 </rfinthk>
      <rfinl> 10 </rfinl>
      <rfinledge> 2 </rfinledge>
      <rfinrim> 4 </rfinrim>
    </endring1>
    <endring2>
      <barext> 4 </barext>
      <ertype> 0 </ertype>
      <erthk> 8 </erthk>
      <erledge> 3 </erledge>
      <rim> 6 </rim>
      <numrfins> 20 </numrfins>
      <rfinthk> 2 </rfinthk>
      <rfinl> 10 </rfinl>
      <rfinledge> 2 </rfinledge>
      <rfinrim> 4 </rfinrim>
    </endring2>
  </endrings>
```

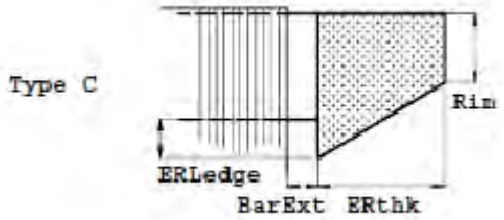
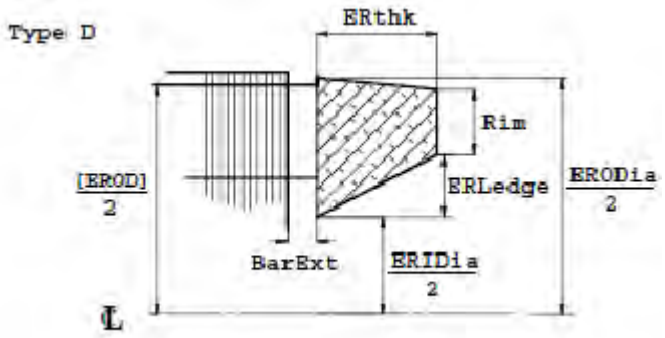
Type D end-ring parameters:

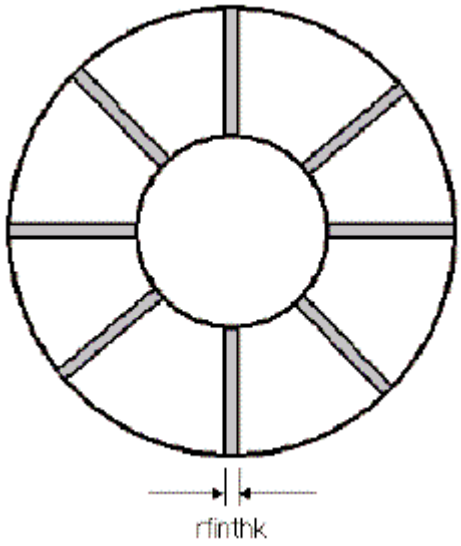
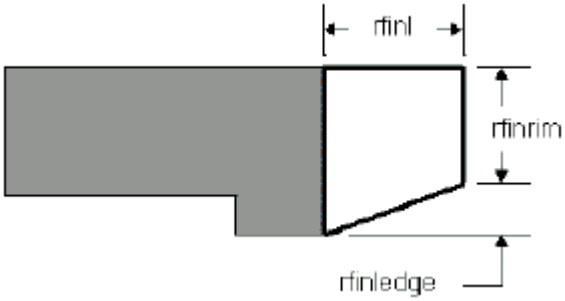
```
<endrings>
    <endring1>
      <barext> 2 </barext>
      <ertype> 3 </ertype>
      <erthk> 10 </erthk>
      <erledge> 2.5 </erledge>
      <rim> 5 </rim>
      <eridia> 30.0 </eridia>
      <erodia> 50.0 </erodia>
      <numrfins> 36 </numrfins>
      <rfinthk> 0.5 </rfinthk>
      <rfinl> 10 </rfinl>
      <rfinledge> 2 </rfinledge>
      <rfinrim> 1 </rfinrim>
    </endring1>
    <endring2>
      <barext> 2 </barext>
      <ertype> 3 </ertype>
```

```
      <erthk> 10 </erthk>
      <erledge> 3 </erledge>
      <rim> 4 </rim>
      <eridia> 30.0 </eridia>
      <erodia> 50.0 </erodia>
      <numrfins> 36 </numrfins>
      <rfinthk> 0.5 </rfinthk>
      <rfinl> 10 </rfinl>
      <rfinledge> 2 </rfinledge>
      <rfinrim> 1 </rfinrim>
    </endring2>
  </endrings>
```

The parameters are as follows:

endring1 / endring2	Distinguishes between the -Z side end-ring (endring1) and the +Z side end-ring (endring2).	
barext	Distance that the cage bar extends beyond the rotor stack.	
ertype	Defines the end-ring type. The available values are 0, 1, 2, 3.	
	0	
	1	
	2	

		
	3	
erthk	End-ring thickness in the Z-direction.	
erledge	Distance that the end-ring extends radially inward from the cage bars.	
rim	End-ring thickness in the R-direction (leading or trailing edge).	
eridia	Inner radial dimension parameter for type D end-rings only (see ertype 3)	
erodia	Outer radial dimension parameter for type D end-rings only (see ertype 3)	
numrfins	Number of end-ring fins.	
rfinthk	End-ring fin thickness.	

	
rfinl	End-ring fin length in Z-direction.
rfinledge	Radial distance to define fin shape.
rfinrim	End-ring fin thickness in R-direction (leading or trailing edge). 

Note the following:

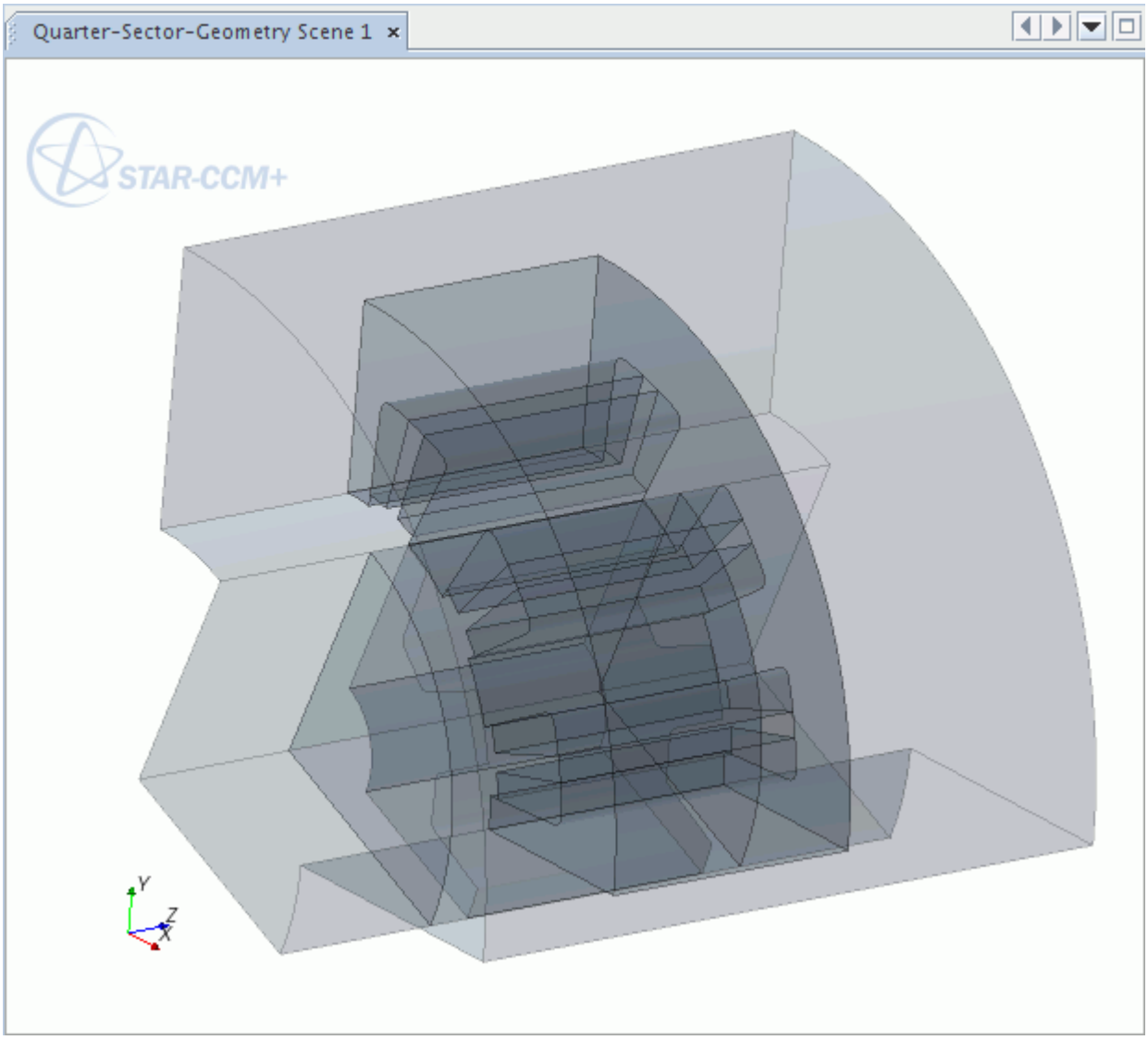
- If ertype is zero (Type A), the end-rings and cage bars are united into one component. The material of this single cage component is that of the cage bars created in SPEED.
- If ertype if not zero (Types B, C, and D), the end-rings are of a different material than the cage bars. The new material set for the end-rings is Al.
- Cage end-rings fins are always of the same material as the cage end-rings.

For more information on the induction motor design parameters that you can specify in SPEED, refer to the SPEED documentation.

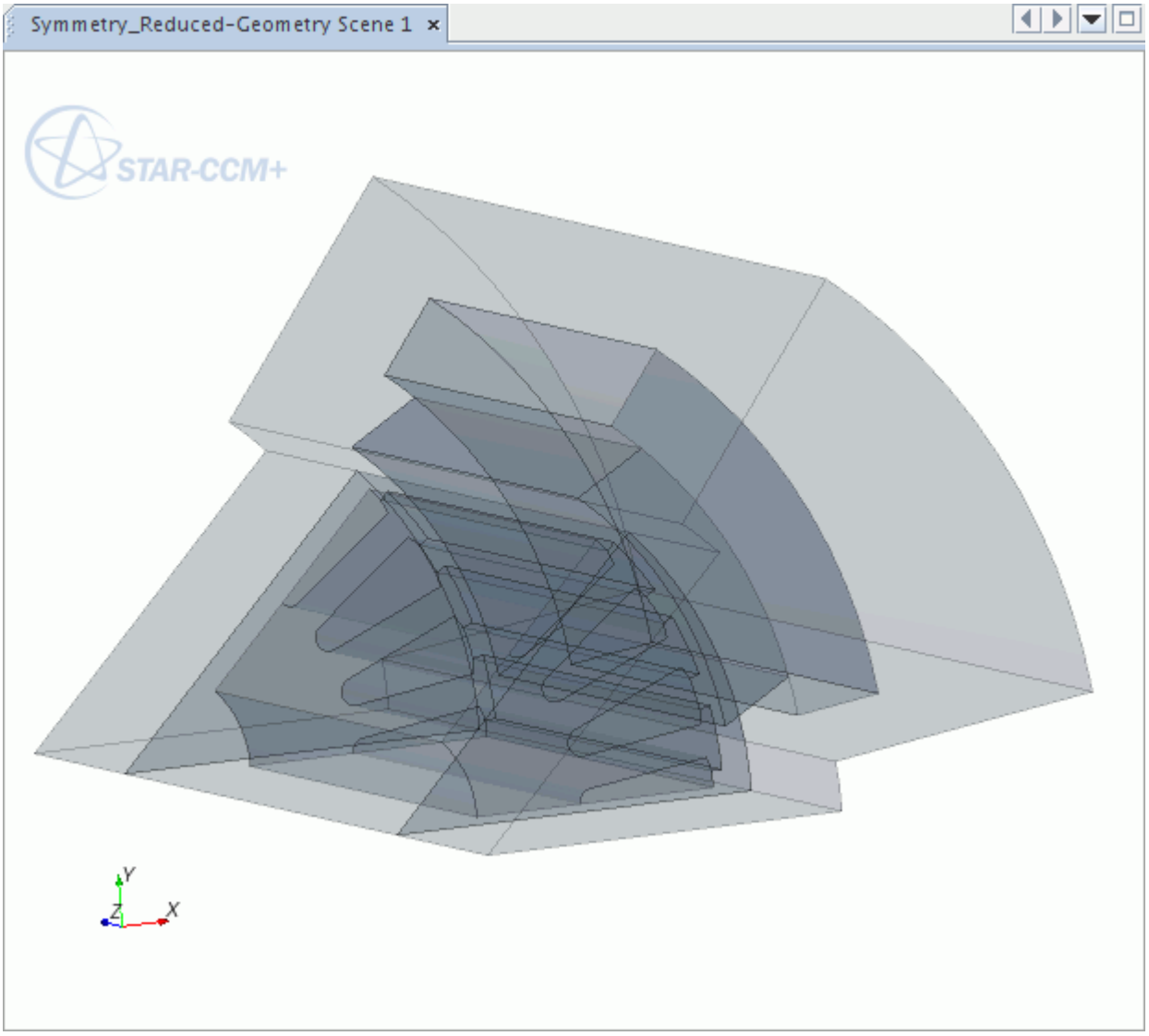
Importing Symmetry Reduced Models

This section describes the symmetry-reduced models that you can import.

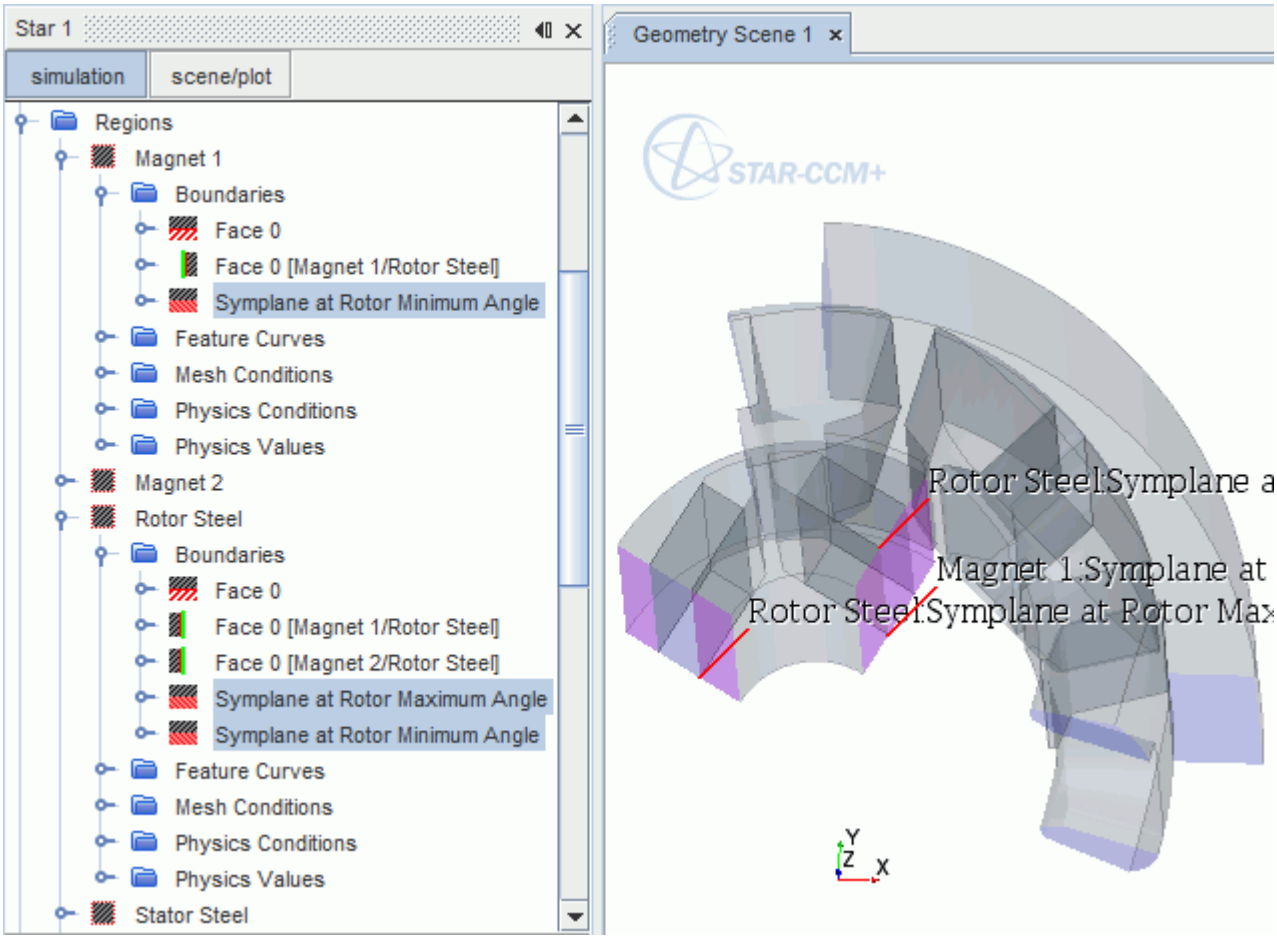
STAR-CCM+ lets you import a `.xGDF` file that contains a symmetrical sector of the machine design, rather than the complete geometry.



The sector can be any appropriate section that utilizes the motor symmetries, such as a half sector, quarter sector, sixth sector, or eighth sector.



As part of the import process, all body faces on the two periodic planes are given a unique face name. An example is shown below.



Importing a sector model lets you exploit axisymmetric periodic geometry to reduce the cell count and the simulation time. STAR-CCM+ automatically creates the appropriate periodic conditions between the two symmetry boundaries.

However, depending on the requirements of the simulation, you may need to use the full geometry. If necessary, you can use the 3D-CAD modeler to create the full model, or use SPEED to create a Whole Motor model that you can import.

If the reduced sector domain cuts through a slot, the windings can be incorrect in the imported model. To resolve this issue, you can expand the domain to include a half span to the adjacent slot.

See Also:

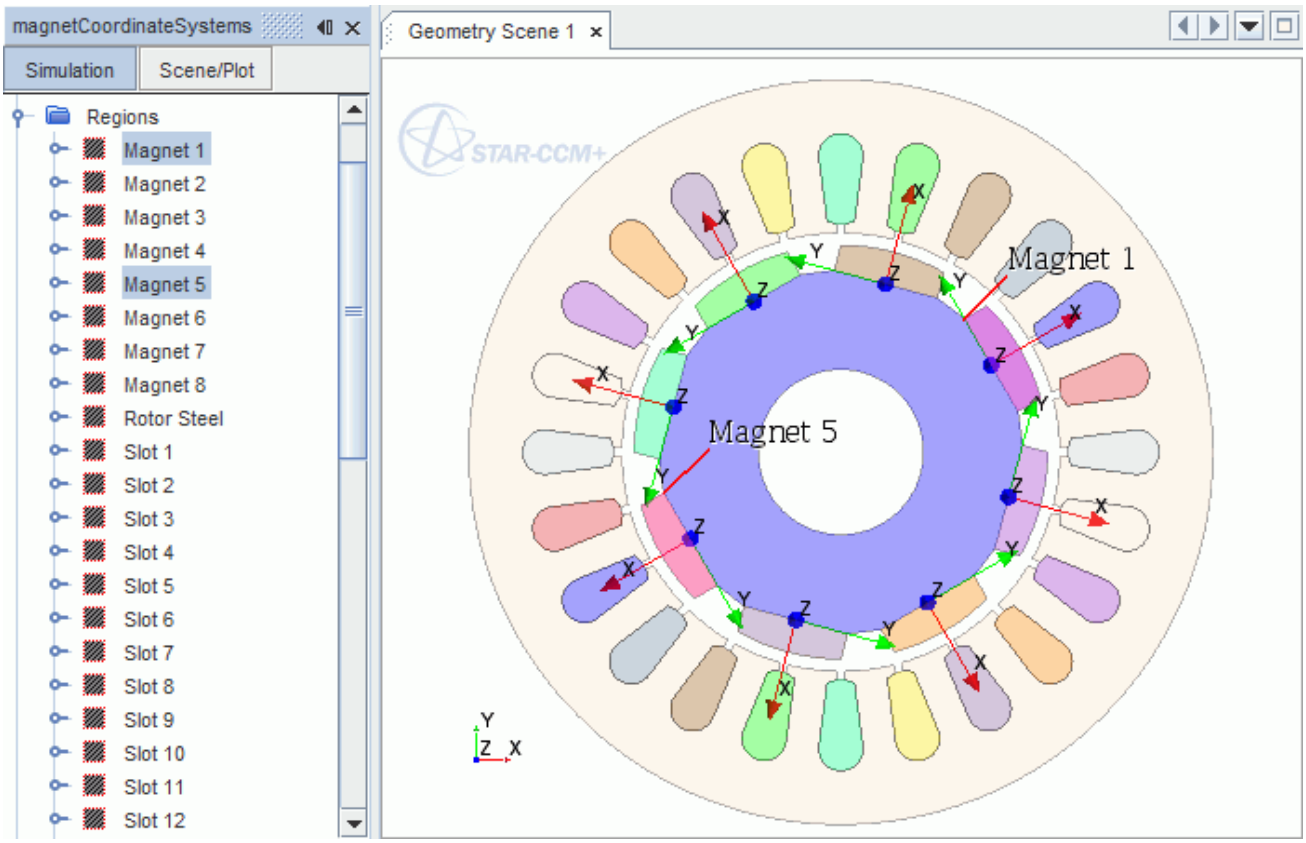
Working with 3D-CAD Models

Magnet Coordinate Systems

STAR-CCM+ creates magnet coordinate systems for motors that have permanent magnets of the Cartesian type.

Radially magnetized magnets use the global cylindrical coordinate system that STAR-CCM+ creates automatically.

A region is created for each magnet that is defined in the `.xGDF` file, as shown in the example below.



STAR-CCM+ creates a unique coordinate system for each magnet region. In each magnet coordinate system, the Z axis points in the same direction as the global Z axis. The X axis is typically perpendicular to the base of the magnet (the face nearest to the axis of rotation). An example is shown below:

Skewed Motors

For linearly skewed motors (mode 1), STAR-CCM+ creates one region for each magnet. For V-linear skewed motors (mode 3), STAR-CCM+ creates two regions per magnet. For stepped skewed or V-step skewed motors (modes 2 and 4 respectively), STAR-CCM+ creates one region for each skew slice, per magnet. This convention is repeated for each skew section, regardless of the skew mode.

Illustrated below is an 8-magnet, PC-BDC motor with a stepped skewed rotor. The rotor is composed of three slices and only one section.

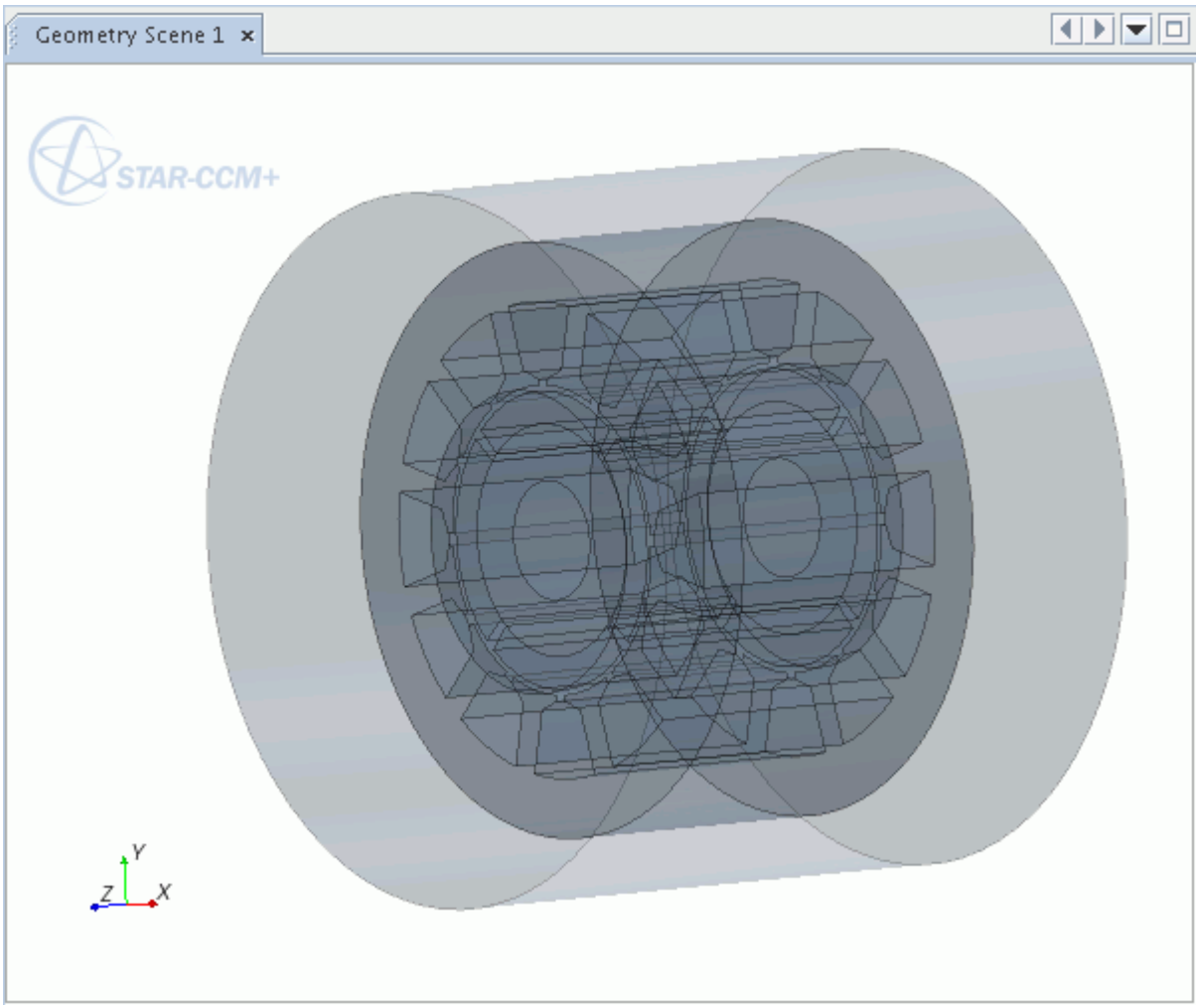
Therefore the number of magnet regions, and therefore the number of magnet coordinate systems, is *magnets x slices x sections*: 8 x 3 x 1 = 24.

Using SPEED Data in STAR-CCM+

This section summarizes how SPEED data is used in STAR-CCM+.

The SPEED Motor Tool Import Wizard can take a few seconds to load the data from the `.xGDF` file into STAR-CCM+. A 3D representation of the SPEED geometry appears in the *Geometry Scene*.

An example is shown below with the transparency activated.

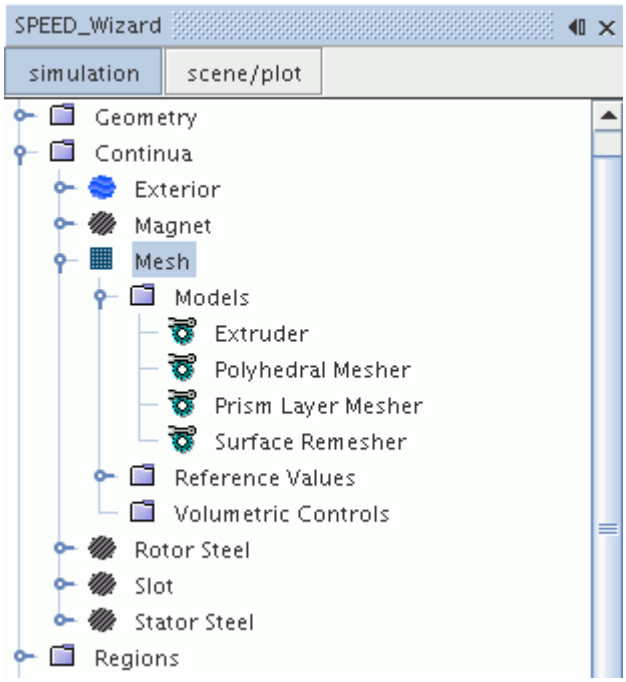


Once the SPEED model has been imported into STAR-CCM+, you can treat it as any CAD geometry and make any necessary modifications.

In addition to importing the geometry, STAR-CCM+ does the following:

1. Generates parts, regions, and appropriate interfaces for the machine, and populates the Parts, Regions, and Interfaces nodes in the Simulation tree.
2. Creates a mesh continuum with the following models:
   - **Extruder**
   - **Polyhedral Mesher**
   - **Prism Layer Mesher**

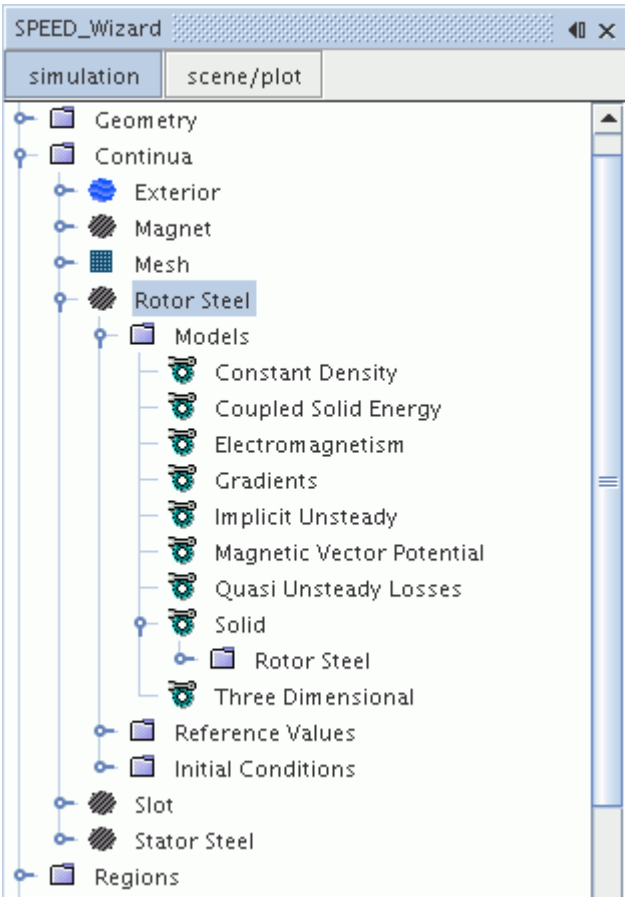
- **Surface Remesher**



You can modify these settings if necessary, but it is recommended that you use the default models.

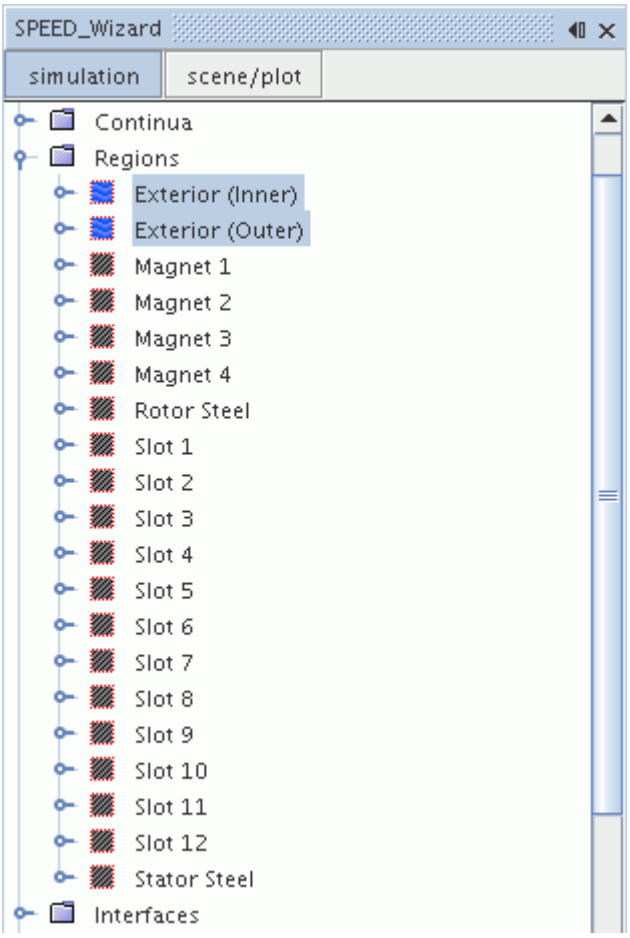
3. Creates the appropriate physics continua, such as Magnet, Rotor Steel, Slot, Stator Steel, and assigns each to the corresponding region. Each of these physics continua includes the Magnetic Vector Potential model.

   The example that is shown below is the Rotor Steel continuum.

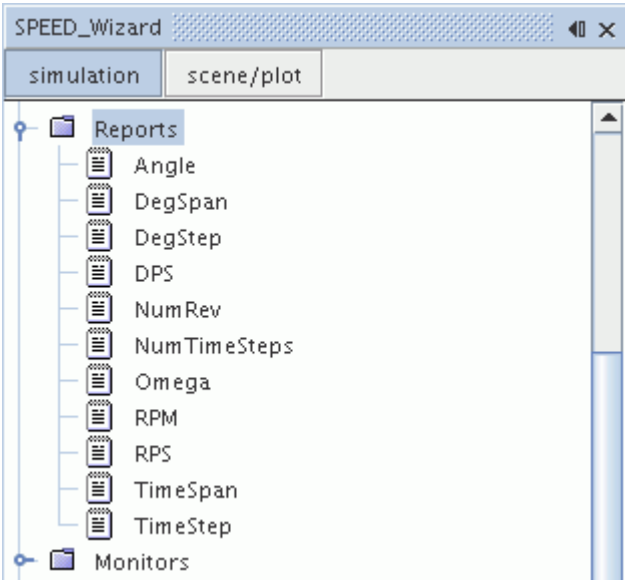


4. If you selected the Build Exterior Domain option in the SPEED Motor Tool Import Wizard, STAR-CCM+ creates the appropriate bounding box to represent the solution domain.

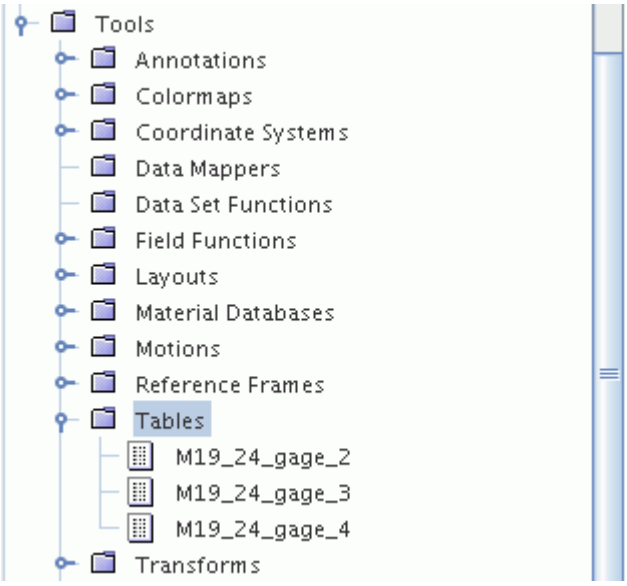
   By default, the domain is sized automatically based on the dimensions of the machine. It is made up of Exterior (Outer) and Exterior (Inner) regions to represent the stationary and rotating domains. The machine is subtracted from both the outer and inner domains.



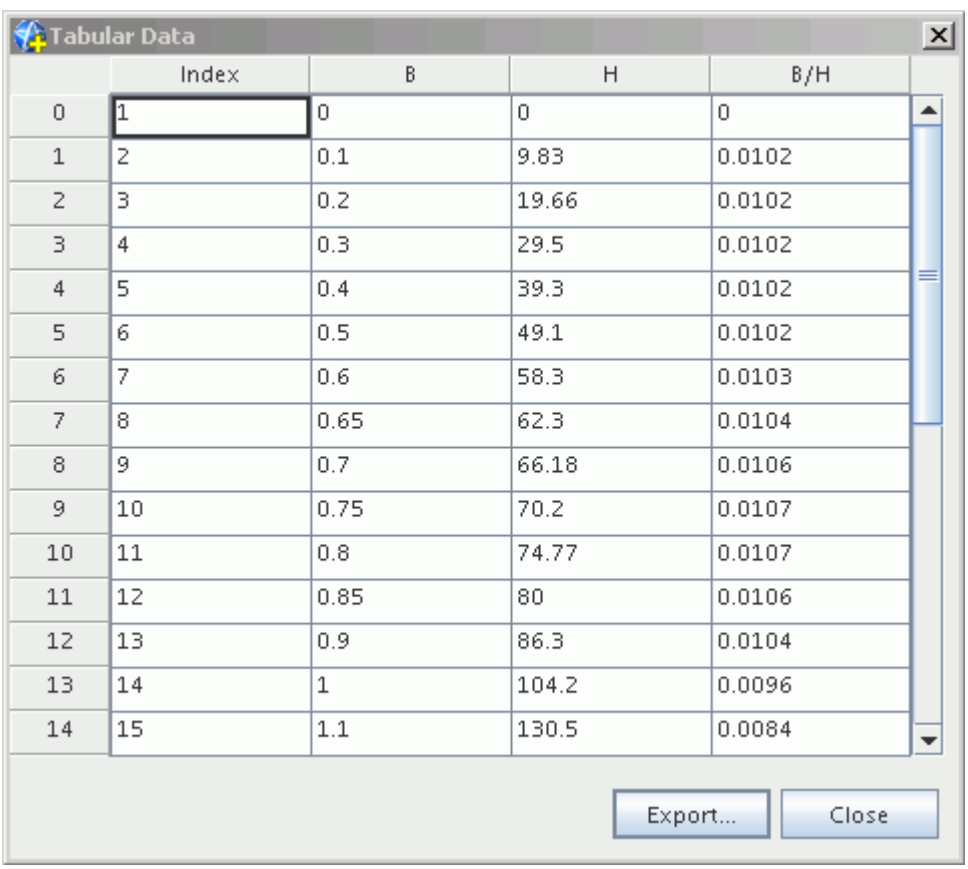
5. Creates plane sections at X = Y = Z = 0 and adds them under the **Derived Parts** node.
6. Generates the appropriate field functions and reports for the electromagnetic analyses.



7. Populates the **Tools** > **Tables** node with property tables based on the materials that are selected in SPEED.



These material property tables can be used as tabular data for the magnetic permeability material property of the Magnetic Vector Potential model. The tables contain columns for the magnetic flux density, magnetic field strength, and permeability. An example table is shown below:



	Index	B	H	B/H
0	1	0	0	0
1	2	0.1	9.83	0.0102
2	3	0.2	19.66	0.0102
3	4	0.3	29.5	0.0102
4	5	0.4	39.3	0.0102
5	6	0.5	49.1	0.0102
6	7	0.6	58.3	0.0103
7	8	0.65	62.3	0.0104
8	9	0.7	66.18	0.0106
9	10	0.75	70.2	0.0107
10	11	0.8	74.77	0.0107
11	12	0.85	80	0.0106
12	13	0.9	86.3	0.0104
13	14	1	104.2	0.0096
14	15	1.1	130.5	0.0084

See Also:

Setting Up an Exterior Domain

Using The IDF Import Wizard

The IDF Import Wizard imports IDF (Intermediate Data Format) files into STAR-CCM+.

IDF files contain design information about circuit boards and electronic components, and are widely used in the electronics cooling industry. The import process reads these files and builds the circuit board in 3D-CAD.

IDF files are neutral text files that typically contain board and library information. Board files contain information about the shape and size of a circuit board, and define the location and size of holes, and the placing of components. Library files contain information about the shape and size of the components. STAR-CCM+ supports the following board file and library file formats:

Board File	Library File
`.emn`	`.emp`
`.brd`	`.lib` `.pro`
`.brf`	`.ldf`

When using the Import IDF Tool, you select the board file. However, the library file must also be in the same directory to load the complete circuit board. If no library file is found, you are asked to locate it manually.

In the electronics cooling industry, the board and its components are all built from the XY plane. The board outline and the components are created from 2D profiles, which are then extruded outwards.

Contents:

Importing IDF Files

Importing IDF Files

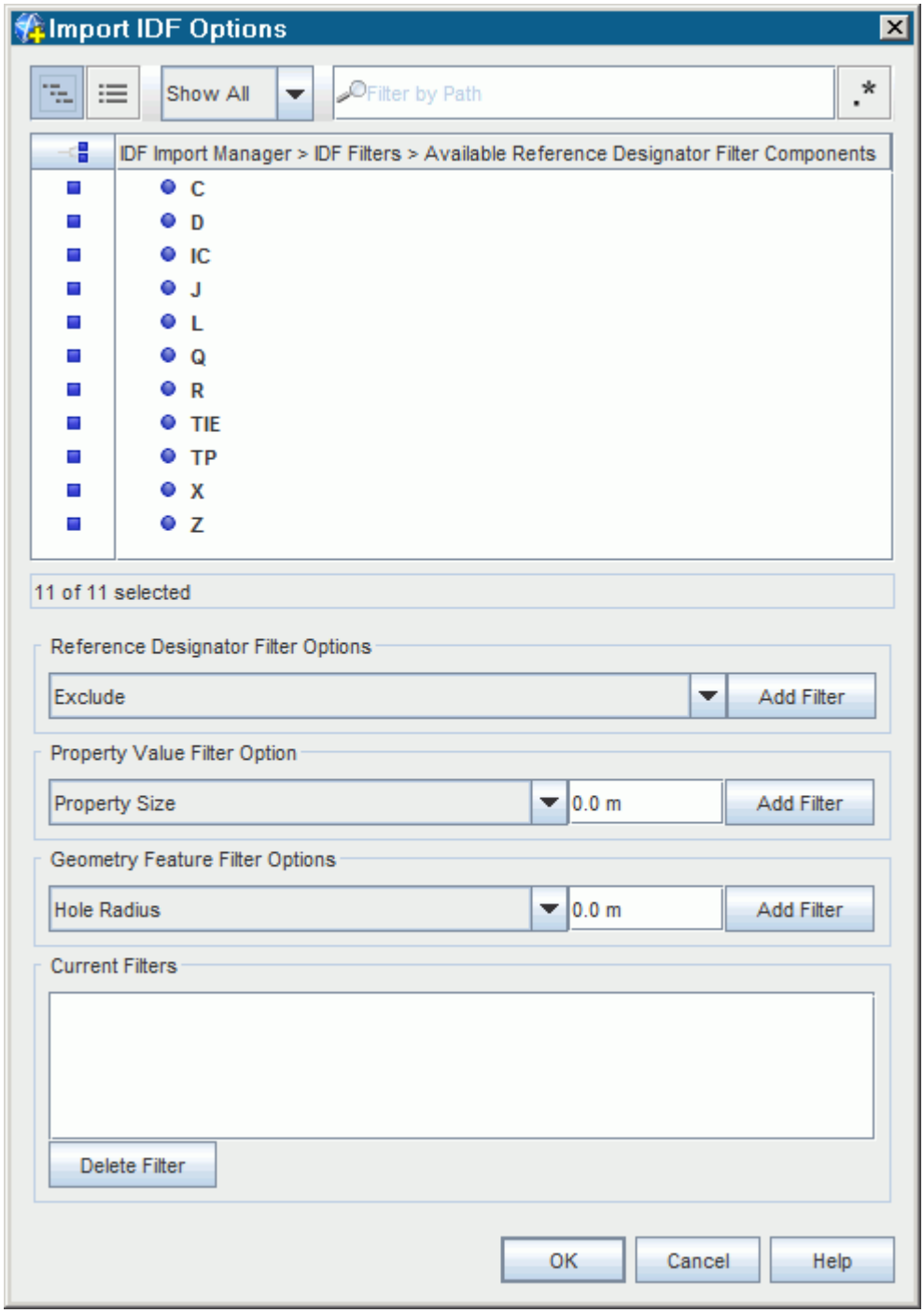
This section describes how to use the IDF Import Wizard.

The IDF Import Wizard is hidden by default from the STAR-CCM+ user interface.

To import IDF files:

1. Add the Import IDF File tool to the STAR-CCM+ user interface. See Customizing a Toolbar.
2. Click the **Import IDF File** icon.

   The *Import IDF Options* dialog appears.



3. Set the desired import options. See Import IDF Options Panel.
   1. Filter out specific components:

      To omit specific components, use **Exclude** filters. Any selected components in the *Available Reference Designator Filter Components* list are excluded from the import. By default, all components are selected.
   2. Filter out components based on size:

To omit all components that are smaller than a specified size, add a Property Size filter. The size that you specify corresponds to the maximum size of a bounding box that fully encloses the 2D profile of the component.

3. Filter out holes on the circuit board:

   To omit holes from the circuit board, add a Hole Radius filter. Any holes which are smaller than the specified value are filtered out.

4. Click **OK**.

The specified components are read into STAR-CCM+ and are subsequently built in 3D-CAD. Each component is built using CAD operations. The benefit of building all components is that you can then modify the circuit board design in 3D-CAD in response to analysis results and re-run the simulation.

When the import completes, a 3D-CAD model named **IDF Import 3D-CAD 1** is added to the **Geometry** > **3D-CAD Models** node. Edit this 3D-CAD model to enter the 3D-CAD environment.

Contents:

Import IDF Options Panel

Import IDF Options Panel

This section describes the *IDF Options* panel.

Use the following options to specify what components to import into STAR-CCM+.

Available Reference Designator Filter Components	Object selector panel that contains a list of components that are identified in the selected file. Select the components that you want to use with the *Reference Designator Filter Options*. By default, all components are selected.	
Reference Designator Filter Options	**Exclude**	Excludes the components that are selected in the *Available Reference Designator Filter Components* list.
Property Value Filter Option	**Property Size**	Specifies the size of the components to use in the filter. The size is defined as the maximum edge in the bounding box that encloses the 2D profile of the component. When added to the filter, components that are smaller than the specified size are not imported.
Geometry Feature Filter Option	**Hole Radius**	Specifies the radius of the hole to use in the filter. When added to the filter, holes which are smaller than the specified value are not imported.

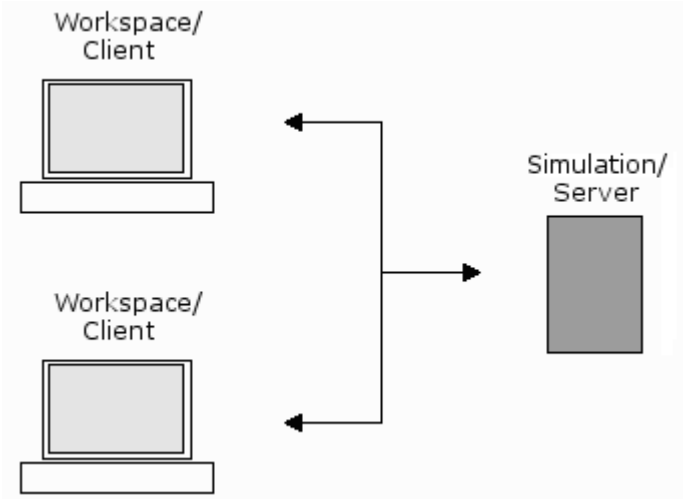
Current Filter	Contains a list of active filters. To delete a filter (or groups of filters), select the filter and click **Delete Filter**.

Client-Server Setup

STAR-CCM+ uses a client-server architecture in which the simulation objects are created on the server and the client provides access to these objects for setup and display.

STAR-CCM+ provides two types of client:

- Interactive client: this client provides the graphical user interface (workspace) that you work with to set up your simulation and to analyze results after obtaining a solution. Every action that you make in the interactive client is recorded in a default Java macro. You can also record separate Java macros that only contain commands for specific sets of actions. You can replay these Java macros in both the interactive client and the batch client.
- Batch client: this client runs from the command line, and can interpret commands contained in a Java macro file that you provide. You typically use the batch client when you want to run the STAR-CCM+ solver on a remote machine or cluster, or as a background job on your own machine.



Communication between a client and a server goes through a port connection that the operating system of the server machine provides. This network port is identified by a five-digit port number on the machine that hosts the server. When you launch a STAR-CCM+ server, the server hostname and port number are output to the command console. Although knowing the port number is essential for connecting a client to a server, STAR-CCM+ can scan host machines for running servers and find the port numbers for you.

Multiple clients can connect to a server at the same time, allowing you to check the progress of a batch job, and also to permit collaboration. When anything changes on the server, that change is reflected on all clients.

Some actions, such as selecting entities during mesh repair, only take place on the client and will not be seen on the other clients.

The client and the server processes can run on the same machine, or they can run on remote machines, and even different architectures. For example, the client can be run on your Windows workstation while the server is run on a Linux cluster. Ideally, the client process should be run on your local machine, which should have hardware graphics acceleration available, while the server should run on a machine or machines with the computing resources suitable for your simulation.

This architecture also allows you to break and make a connection between the client and the server processes, without disrupting the server calculations. So you could start a job from a Windows client, disconnect, and then reconnect the next morning from a Linux client to check on progress.

Note: When connecting to a server, the client must have the same STAR-CCM+ version number as the server. You cannot mix version numbers across a client-server connection.

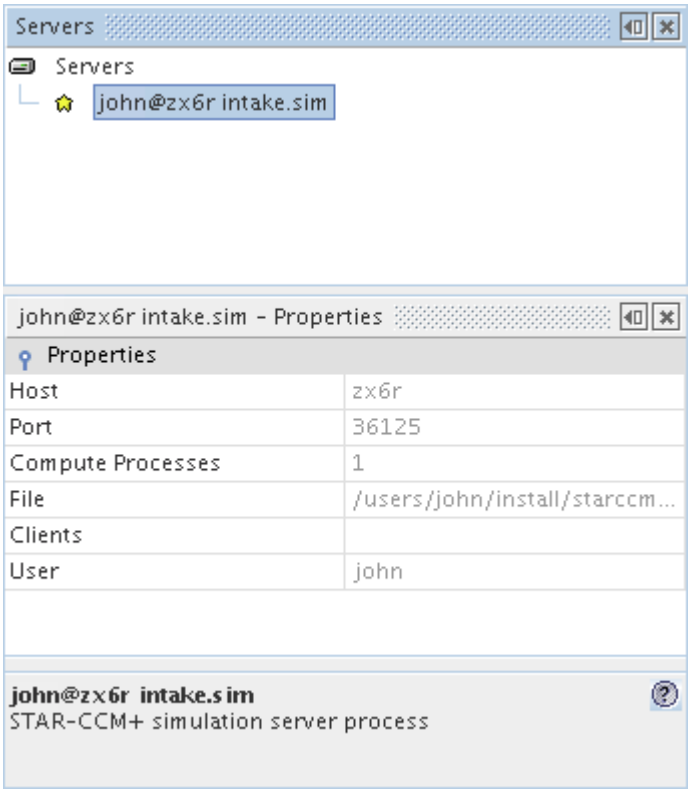
Server Processes

Within the STAR-CCM+ workspace, STAR-CCM+ provides a window that displays a list of server processes that you have launched within the local network. STAR-CCM+ populates this window by sending a broadcast to the local network and listing servers that respond.

The *Servers* window is closed by default. To open it:

- From the top menubar, select **Window** > **Servers**.

One server process node, labeled in the form user@machine path, is displayed for each serial server process or parallel master process. STAR-CCM+ defines a local network as the group of machines that it can access without crossing a router, subject to any firewalls blocking the range of possible server locator ports (47827-47926).



To view server and client processes registered by the operating system of a machine that you are logged into:

- On Linux, issue the `ps` command and view the list of threads. As a general rule, the server threads are listed as `star-ccm+`, and the client threads appear as `java`.
- On Windows, launch the task manager and choose to view all processes. The server threads are listed as `star-ccm+`, and the interactive client appears as `starccmw.exe`.

Contents:

Client-Server Connections
Mixed or Double Precision
Launching STAR-CCM+ and Working With Simulations
Working With Parallel Servers
Command-Line Reference

Client-Server Connections

STAR-CCM+ allows you to connect a client to any running server for which you have access, and also to disconnect clients when you choose. The client version must match the server version for a successful connection. Secure connections require additional inputs.

As several related actions are possible for client-server connections, these actions are described in the following table.

Action	Steps
Disconnect a client from its server	• Within the STAR-CCM+ workspace, make sure that the correct simulation window is active. • Select **File** > **Disconnect**. This action disconnects the client from the server. Both processes are still running, although you no longer see the simulation window in the client. If all clients are disconnected from a running server, no graphics scenes will be saved to files. STAR-CCM+ warns you when disconnecting the last client. You can close the client workspace after disconnecting from a server. The server continues to run. **Note:** When a simulation is started with the **Server Connection Mode** set to **No Connect**, the **File** > **Disconnect** menu item is grayed out.
Connect to a server that you started	You can choose from multiple methods: • From within the STAR-CCM+ workspace, select **Window** > **Servers**, right-click on a server entry, and select **Connect**. • Alternatively, from the top menubar, select **File** > **Connect to Server...** and enter the *Host* and *Port* number of the server you want to connect with.
Connect to a server through a secure tunnel	You can connect to a running server through an SSH tunnel. This approach is necessary when the client machine cannot see the server host, but you have SSH permissions to another machine which can see the server host. These SSH permissions must allow you to connect without requiring a password. The most common example of this scenario is when the server is running on the nodes of a cluster, you want to run the client on your local workstation, and your workstation can see the head-node of the

Action	Steps
	cluster but not the nodes used to run jobs. Therefore you must connect using an ssh tunnel through the head-node. 1. If you always use the same SSH tunnel settings, first define these settings as defaults for all your STAR-CCM+ sessions: a. In the STAR-CCM+ workspace, select **Tools** > **Options**. b. Within *Options*, select **SSH Tunnel** and complete the property fields required. c. Close the *Options* dialog. 2. From the top menubar, select **File** > **Connect to Server...**. 3. Activate *Connect Through SSH Tunnel*. Modify the subsequent properties if necessary. 4. Enter the *Host* and *Port* for the target server.

Contents:

Providing Remote Server Access
Server Processes Reference
Connect to Server Dialog Reference
Connecting a Client from the Command Line
Running the Client in Batch
Configuring the Server to Shutdown When Idle
Closing the Server Automatically Upon Solver Completion
Working with Remote Servers
Starting a Proxy Server on a Host
Starting a Server without a Client
Displaying Diagnostic Messages from the Server and Client
Setting Network Binding Order for Failed Visualization Connections

Providing Remote Server Access

Before STAR-CCM+ can launch server processes on remote machines, you must provide details of a remote shell environment that STAR-CCM+ can use. This environment must not require any password when making connections, although you can use a different username from your own.

Make sure that the system on which the master STAR-CCM+ process runs has a remote shell command installed and available to the command line.

- On Linux, the default option is `ssh`. Note that prior to STAR-CCM+ v11.02, the default option was `rsh`.
- On Windows, you can use `PuTTY`, or an alternative that you provide.
- For either system, you can specify an alternative command.

To configure STAR-CCM+:

1. Within the interactive STAR-CCM+ workspace, select **Tools** > **Options**
2. In the *Options* dialog, select *Environment*, then set the following properties:
   - *Generic Remote Process Launch Command*
   - *Server Remote Process Launch Command*

   The Generic command provides default inputs for processes launched from the client. The Server command provides inputs for processes launched from the server, such as parallel server processes and co-simulation. For details, see the Environment section of the default properties reference.
3. Make sure that passwordless connections are possible. For example, issue an ssh command from a command prompt, such as `ssh <host> hostname`.

Server Processes Reference

Server processes are listed beneath the **Servers** node in the *Servers* window. Additionally, this node has its own pop-up menu that provides server-specific actions.

Servers Manager Properties

Refresh Time	This is the upper limit on time (in milliseconds) that may elapse before STAR-CCM+ refreshes the list of processes. The list also gets updated in response to events such as the opening or closing of a simulation. However, the window must be in the foreground to be updated.
Time to Live	Essentially, this is the number of routers that the broadcast message will be allowed to jump when searching for servers on the network. However, the accuracy of this number depends on how routers on the internal network are configured.

Server Process Node Pop-Up Menu

Connect	Executes the connect to server action to connect this client to the selected server process. More than one client can be connected to a given server process.
Properties	Displays the properties of the object in a separate stay-on-top modeless dialog.

Server Process Properties

These properties apply to each entry listed beneath the **Servers** node.

Host	The name of the computer where the STAR-CCM+ server process is running (read-only).
Port	The network port on which the STAR-CCM+ server process is listening for clients (read-only).
Compute Processes	Indicates the number of processes running for the selected STAR-CCM + server (read-only).
File	The path name of the simulation file (read-only). This value may be blank if the server process is for a new, unsaved simulation.
Clients	The user name and computer IP address of each connected client computer. This read-only value may be blank if no client is currently connected to the server.
User	The name of the user who started the STAR-CCM+ server process (read-only).

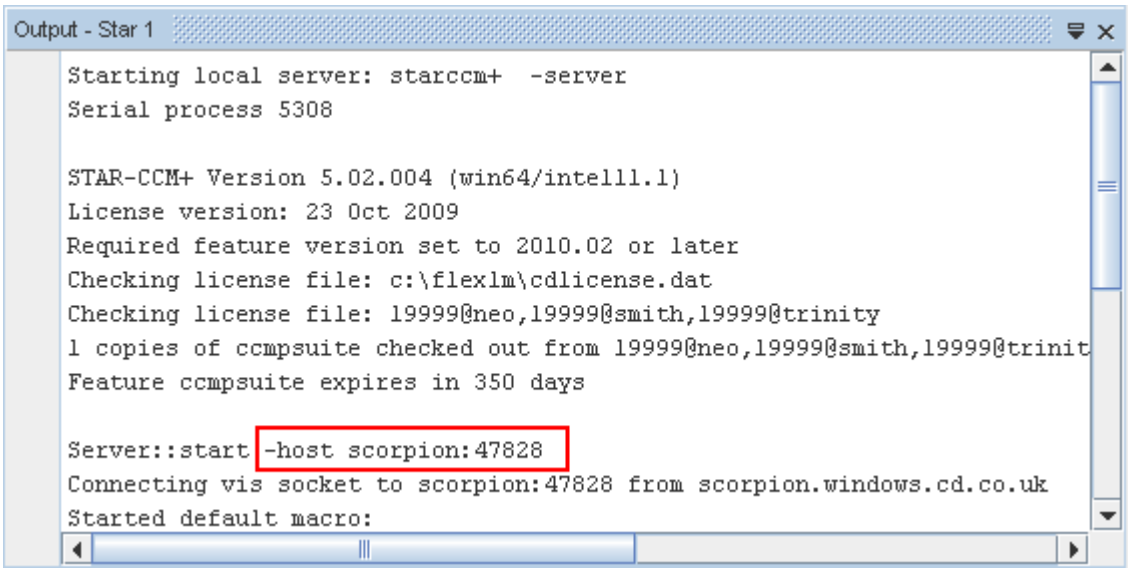
Connect to Server Dialog Reference

STAR-CCM+ allows you to manually enter details of a server to which you wish to connect.

Host	The host name of the machine where the server process is running. The dialog includes a scroll-down list of past client-server connections, allowing you to access a host multiple times while entering its information only once.
Port	The port number that the server process is listening on. When a STAR-CCM+ server process is started, it reports the port number that it is listening on. In the following example, the server is started on the host cindy and is listening on the port 47828: `Server::start -host cindy:47828`
File	Read-only entry that shows the name and location of the simulation file associated with the server.
Scan Host	When clicked, this button checks for STAR-CCM+ server processes on the specified host, and indicates how many it finds.
Connect Through SSH Tunnel	Activates the secure shell tunnel options, and applies these in order to access the chosen *Host*.
Secure Shell Tunnel Host	The name of the host machine through which you wish to tunnel. In a cluster situation this would be the cluster head-node.
Secure Shell Tunnel Host Username	Your user name on the ssh host. You must be able to access this host without a password.
Secure Shell	Choose the type of shell that you want to use for the tunnel. The default type is **ssh**.
Secure Shell Options	Include any extra options.

Connecting a Client from the Command Line

When STAR-CCM+ starts a new server, it prints the hostname and port number in the *Output* window in a form that matches the syntax required by the STAR-CCM+ launch command.



Using the output shown above, the command to launch a client that connects to this server would be:

```
starccm+ -host scorpion:47828
```

The same output will be provided when a server is launched remotely from the command line; it is then convenient to copy and paste the relevant portion in order to launch a local client.

Running the Client in Batch

Large simulations generally require that you prepare the case on your local machine, then submit the case to a remote machine, or remote cluster, for running. In these cases, you can use the STAR-CCM+ batch client that executes a series of commands from a macro file without user interaction.

Running in batch mode typically requires that you execute STAR-CCM+ from the command line, pass it the name of a macro script that contains all the commands necessary to load a mesh, set properties, run a number of iterations and render images of scenes and/or plots. In this mode, the batch client compiles and runs the script, passing on all the information and instructions to the server.

The instructions for recording, editing and running a macro file are found under section, Scripting the Application. Similar macro recording techniques can be used to set up a large job on a small machine that would not normally handle the large job. These batch jobs can also be submitted to batch management systems using additional command line parameters.

A batch job can also be accessed while in progress, as multiple clients can connect to the same server. By connecting through an additional client, you can perform interactive tasks such as opening scenes and adjusting solver parameters. You can use the auto save feature to save automatically at the end of a run.

Preparing Large Simulation Cases

You can prepare the batch script for a large problem that would not necessarily fit onto your computer. This approach would typically happen if your machine did not have enough memory for a large mesh.

In this scenario, first set up a smaller, similar mesh and record the macro using the smaller mesh. Then edit the script and change the name of the mesh file that is imported to represent the original problem. For example, the code in question may look something like:

```
getSimulation().getImportManager().importFile(resolvePath("smallMesh.cc
m"));
```

Change the file name in quotes in addition to any other changes you were needing to make on that file.

Batch Summary Report

A summary report of a batch simulation can be obtained by providing the following command line option:

```
-batch-report
```

The report in `.html` format provides details of the system the simulation was run on, as well as system information about the simulation itself including elapsed times. It also contains a copy of the STAR-CCM+ console output that was written during the simulation.

When you enter the line command, the report is generated in your default browser.

Error Codes

When a batch job completes, STAR-CCM+ returns known error codes. These codes can be accessed using techniques appropriate to the Operating System. On Linux using the Bash shell, the error code can be

accessed from the variable `$?`. On Windows, the error code is accessed using the environment variable `%errorlevel%`.

Exit Code	Description
`0`	No Error
`1`	Generic Error
`2`	Usage Error
`4`	File Error
`8`	Licensing Error
`16`	Floating Point Error
`32`	Non-Recoverable Error

If more than one error condition is encountered the error codes are accumulated. For example, if a file error and a non-recoverable error is encountered the exit code is "4 + 32 = 36".

On Linux, STAR-CCM+ also returns exit codes that are associated with known signals. For example, if STAR-CCM+ is terminated by sending SIGTERM—which typically has the value 15—to a server process, the batch STAR-CCM+ simulation returns exit code 143 (which is 128 + signal number).

When using parallel simulations the Message Passing Interface (MPI), which is responsible for spawning the server processes, may return a non-zero unrecognized error code if the server processes are terminated. The exit code depends upon the specific MPI being used and also on the operating system.

If a subset of parallel processes exit, MPI may explicitly terminate the remaining processes using a platform-dependent method. For example, on Linux, MPI may send SIGTERM to any remaining processes and therefore their exit codes are 143 (the exit code associated with SIGTERM).

Configuring the Server to Shutdown When Idle

You can configure STAR-CCM+ to release a license automatically (by shutting down the server) when it detects no activity for a set amount of time.

You control the automatic server shutdown through a configuration file, `sitesettings.props`, that was added to your STAR-CCM+ installation directory during the installation process. By default, the STAR-CCM+ server does not shutdown automatically. If no configuration file is found, the default behavior stands and no timeout occurs.

When the automatic server shutdown feature is active, the server issues a message in the output window upon startup. If you set the timeout value to two hours, the message reads:

```
Server idle time limit is 2 hours.
```

STAR-CCM+ checks for the following activity on the server every five minutes:

- reading/saving files
- visualization requests
- editing properties
- meshing
- solving

If STAR-CCM+ detects no activity, and the timeout time is approaching, the server issues a warning message in the client output window. For any timeout value less than two hours, the idle warning is issued at 90% of the timeout value. For timeout values over two hours, the idle warning is issued one hour before the timeout.

If no activity is detected after the warning message, the server shuts down at the set timeout value, releasing the license. The client, however, remains open. A copy of the simulation file, named `[filename]@idle.sim`, is saved in your working directory.

Note: If you set the timeout value to 5 minutes or less, the actual server shutdown takes place 10 minutes after the warning message appears. This behavior is designed to give you enough time to react to the warning message before the server shuts down.

If multiple clients are connected to the same server, the warning message is sent to all clients, and only one client needs to respond to reset the timeout. Performing any action that invokes the server resets the timeout counter.

Automatic Server Shutdown in Batch Mode

If no client is connected to the server, the batch job stops by itself, closes the server, and automatically releases the license. If, however, one or more clients are connected to a running batch job, the server does not close down, and the license remains checked out. To avoid this issue, set up the automatic server shutdown feature. See Setting Up the Server to Shutdown Automatically.

Contents:

Setting Up the Server to Shutdown Automatically

Setting Up the Server to Shutdown Automatically

In order to set up the automatic server shutdown, you need to be able to access the STAR-CCM+ installation directory. If STAR-CCM+ is installed on a network location, contact your system administrator and ask them to perform the changes to the configuration file.

To set up the automatic server shutdown:

1. Navigate to your STAR-CCM+ installation directory:
   - For Windows: [starccmp_installation]\star\config
   - For Linux: [starccmp_installation]/star/config
2. Open the `sitesettings.props` file in a text editor of your choice.

An example `sitesettings.props` file is shown below:

```
# STAR-CCM+ Site Settings
#Mon Jul  2 15:45:59 EDT 2012
ServerIdleTimeout=0
```

3. Enter the timeout value (in hours) next to the `ServerIdleTimeout=` argument. If you leave this field as `0`, the automatic server shutdown feature is disabled.
4. Save the configuration file.

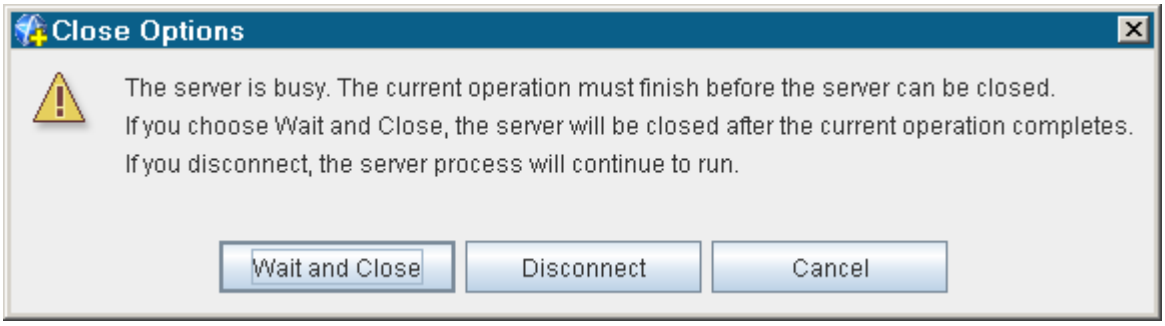
Closing the Server Automatically Upon Solver Completion

Often you may use the interactive client to monitor the mesh generation, the early stages of the solution, and to ensure that everything is set up correctly with no sign of problems. At this point, you may decide to leave the case running overnight, or over the weekend. You can instruct STAR-CCM+ to close the server when the solver is finished. This feature is particularly useful if you are using Power-On-Demand (PoD) licensing, as you are being charged for the duration of time the server is active, even though you may not be actively using it.

To instruct STAR-CCM+ to automatically shut down the server when the solver is finished:

1. Click the **X** button next to the simulation in the *Explorer* pane.

The *Close Options* dialog appears.



The following options are available:

2. **Wait and Close** - this option leaves the server running, with the interactive client open and connected, until all convergence criteria are met. When the convergence criteria are met, the server will save the simulation file and close automatically. The *Output* window in the client will remain visible. This option is ideally suited to running an analysis using PoD licensing.
3. **Disconnect** - this option disconnects the interactive client from the server. The solver continues running and the server will remain active indefinitely until you manually close it. You can reconnect to the server and close it.

Note that the above dialog and options are only available when you close the simulation while the solver is running.

Working with Remote Servers

The client-server model employed in STAR-CCM+ allows you to start a server on a different machine to the client workspace.

This approach is particularly useful if you wish to set up and monitor a simulation from your workstation but let another machine compute the solution. Starting remote servers requires you to have remote shell access to the remote server machine without password prompting (for example, passwordless `ssh`). The `starccm+` command must be available on each of the remote machines.

By default STAR-CCM+ uses `ssh` as the remote shell program used to launch remote servers. You can use alternative remote shells such as `rsh`, `PuTTY`, or a custom shell command that you provide. For a custom shell command, the remote shell specified must conform to the syntax:

```
"remote shell command" <HOST> -l <USER>
```

where `HOST` is the remote machine to which the remote shell will connect and the username is specified using the `-l` syntax.

Before STAR-CCM+ is launched, you must ensure that the remote shell program can connect to the server without requiring further action by you (such as entering a password or username).

Windows systems do not provide native server-side support for remote shell access. However, you start remote servers from a client running on a Windows machine if an appropriate command, such as `PuTTY`, is available for STAR-CCM+ to use. Before STAR-CCM+ is launched, you must ensure that the remote shell program on Windows will connect to the server without requiring further action by you.

Separate sections cover the specifics of starting the server remotely in:

- Serial with the workspace
- Serial with the command line
- Parallel with the workspace
- Parallel with the command line

You can also log in to the remote machine directly, launch a server from the command line, and then connect to that server from your own machine using the host and port information that the server prints on startup.

Clients may be launched from the command line that connect to specified remote servers.

Dealing With a Protocol Failure

On some Unix based systems there is a limit to the number of open remote shell sessions. If this limit is exceeded, the system may report a protocol failure when starting a remote server or a parallel server. This problem is usually the result of defunct rsh processes and you should first ensure all such processes are killed.

If the problem persists the administrator should ensure the limit on the number of remote processes is not set too low. On Linux systems the file `/etc/inetd.conf` can be modified to allow more rsh processes per minute. For example,

```
shell stream tcp nowait root /etc/tcpd2 in.rshd
```

to

```
shell stream tcp nowait.200 root /etc/tcpd2 in.rshd
```

Starting a Proxy Server on a Host

You can specify a host on which to start a proxy server using the `-proxy <host>` option.

You can use the `-proxy` option with both serial and parallel simulations, and all methods of specifying the hosts where the simulation runs, including the `-batchsystem` command-line argument. To expose simulations that are running on clusters to workstations outside the cluster's private network, use the `-proxy` option.

A typical command to enter in the command line is:

```
starccm+ -batch -np 64 -machinefile hostfile -rsh ssh -proxy cluster-
head mycase.sim
```

where:

- `-batch` is the argument that is used to run the case in batch mode
- `-np 64` specifies the number of cores to use (64 in this example)
- `-machinefile hostfile` is the configuration file for the host you want to spawn the proxy server on
- `-rsh ssh` is the shell that you want to use to spawn a proxy server on the specified host. By default, the secure shell (`ssh`) is used. You can specify to use the remote shell (`rsh`) by adding -rsh rsh to the command line.
- `-proxy cluster-head` is the head node of the cluster you want to connect to.
- `mycase.sim` is the name of the simulation file you want to connect to

The proxy server appears in the *Servers* tab in STAR-CCM+. You can connect a client to this server. The proxy server continues running until the connection with the parent server is lost.

On startup, the *Output* window shows:

```
Server::start -host cluster-node-01:47827
```

This message identifies which host the server is running on and the port number on which the server is listening for client connections. In the example above, the host is `cluster-node-01` and the port number is `47827` (which is the default STAR-CCM+ port number).

Additionally, the output also displays:

```
ProxyServer:start -host cluster-head:47827
```

This message identifies the host and port number of the proxy server. In the example above, the cluster's head node is `cluster-head` and the port number is `47827`. The port numbers of the server and proxy server do not have to be the same.

You can now connect to either the server or proxy server using the respective host and port combinations. If connecting to the server from outside the cluster, it is likely that you cannot see the cluster nodes (such as `cluster-node-01`) but are able to see the cluster's head node (`cluster-head`). In this case, connect through the proxy server.

Connecting to the proxy server grants you the same access to the simulation as connecting directly to the server.

Starting a Proxy Server from a Windows Machine

The proxy server requires that rsh/ssh is able to start a remote process from the server's host to the proxy host. Windows machines do not support rsh/ssh by default—you can obtain an additional package (Windows Services for Unix) that adds this support from Microsoft: http://www.microsoft.com/download/en/details.aspx?id=274. Alternatively, you can use a separate program such as PuTTY.

Starting a Server without a Client

You can open an existing simulation in a server without explicitly starting the client.

In this scenario, the server will start and the simulation file will be opened and the server will simply wait for a client to connect to it. This may be useful if you want to start a remote server from a Windows machine for example.

To open a simulation on a server without launching the client, issue the following command:

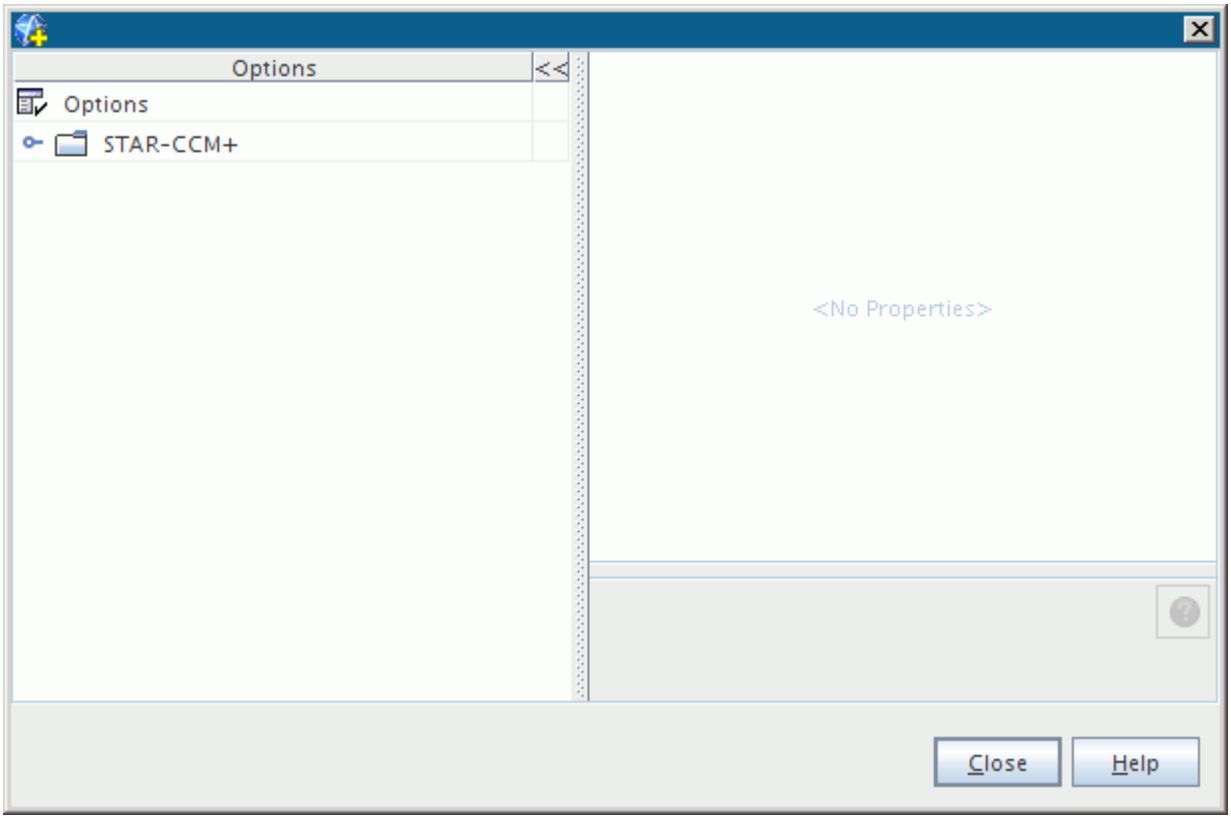
```
% starccm+ -server -port [portnumber] file.sim
```

The server will use the port specified via the `-port` option (default `47827`) or the first open port after the specified port, until it has tried the number of ports specified via `-portrange` (default 100). If it cannot find an open port in that range (default `47827` - `47927`), the server will print a message and exit.

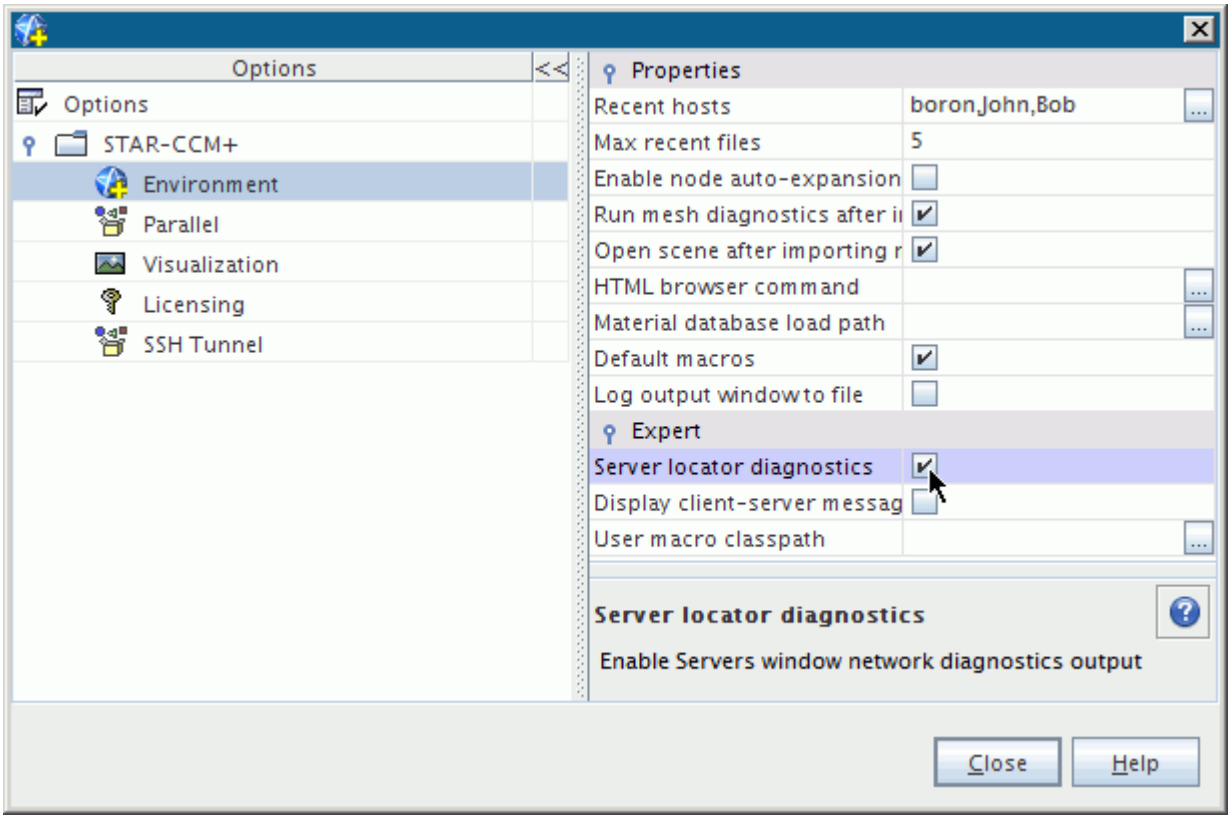
Displaying Diagnostic Messages from the Server and Client

To troubleshoot client-server connections, the server locator allows you to gather diagnostic information from the client and server in one place.

To make the server locator available, select **Tools** > **Options**. This activates the *Options* dialog.



Open the **STAR-CCM+** node, then select the **Environment** node to access its properties, which include the *Server locator diagnostics* expert property.



Activate the checkbox of this property to have messages generated by the client and server. The client and any connected server will print messages related to the server locator in the *Output* window.

For complete information about client-server activity, use this feature by following these steps:

- Start a server with a line command:

```
% starccm+ -server -cout file.sim
```

- Launch the workspace and connect to the server.
- Activate the server locator option. After a few seconds diagnostic information from the client and server will be generated.

Setting Network Binding Order for Failed Visualization Connections

When sockets between client and server are opened Java and C++ use slightly different rules for picking up the network interface, so on some machine with multiple interfaces the Visualization socket fails to open. This problem can be overcome by ensuring that your default LAN interface is at the top of the network adapter binding order on your machine. To do this on Windows:

1. Open the Network Connections folder. To do this: click **Start** on the desktop, select **Run...**, enter `ncpa.cpl` in the *Run* dialog and click **OK**.
2. Press the `<ALT>` key to make the menu show up on the *Network Connections* folder. Select **Advanced** > **Advanced Settings...** Enter the Administrator password if prompted.
3. Click the *Adapters and Bindings* tab.
4. Use the arrow buttons to move your Local Area Connection to the top of the Connections list.
5. Click **OK** to close the dialog.

This should resolve issues with the Visualization connection error.

Mixed or Double Precision

STAR-CCM+ comes in two versions: a mixed precision version and a double precision version. Both versions are provided on the download area of the Steve Portal.

The double precision version of STAR-CCM+ is generally intended for cases that involve physical phenomena of vastly differing scales. Examples include: sound wave propagation in water, three-dimensional microstructure battery modeling, electrodynamics simulations, and multiphase flows.

The advantage of running these simulations in double precision is increased accuracy due to less round-off error. This increase in accuracy leads to robust convergences of problems that would otherwise not converge to an adequate result in single precision.

When running simulations with double precision:

- The memory requirements are 20% greater than with mixed precision.
- The overall simulation time increases by 15-20%.
- The size of `.sim` files increases by not more than 100% (depending on the proportion of floating point fields for the case).

Refer to the release notes for a list of the platforms for which a double precision version is available.

To run simulations with double precision, install a double precision version of STAR-CCM+. The interface and workflow for both versions are identical.

Precision

The term precision refers to the size of (and hence, number of digits of precision that are stored for) floating point data. A single precision float occupies 4 bytes of memory and has 7 digits of precision. A double precision float occupies 8 bytes of memory and has 15 digits of precision.

Double Precision Variables

In the mixed precision version of STAR-CCM+, the fields of double precision floating point data include:

- Coordinates (that is, vertices locations in space)
- Pressure
- Displacements (for example, solid-stress displacements).

In the double precision version of STAR-CCM+, all fields of floating point data are double precision.

Fields of other types (for example, integers and booleans) are the same in either version.

Launching STAR-CCM+ and Working With Simulations

A STAR-CCM+ simulation exists on a server, and is accessed by connecting a client to the server on which the simulation exists.

New simulations (with their corresponding servers) can be created interactively from the client workspace, or can be created using the command line. Similarly, existing simulations can be loaded interactively or from the command line.

The server that is used to host new or existing simulations can be launched as a serial or parallel process.

Contents:

Launching STAR-CCM+
Interactive Startup Reference
Creating and Opening Simulations from the Command Line in Serial Mode
Starting Remote Servers from the Command Line
Restrictions on Names of Files and Paths in STAR-CCM+
Using the Simulation Pop-Up Menu
Saving Simulations to File
Reverting Unsaved Changes to the Simulation
Closing the Simulation
Closing the Client Workspace
Viewing and Editing the Log File
Using Summaries

Launching STAR-CCM+

Launching STAR-CCM+ is generally a two-step process. For an interactive session, you first launch the client, and then you launch a server process from the client. You can also launch servers directly and connect to them later.

STAR-CCM+ operates using a client-server architecture. The client software provides an interactive Graphical User Interface (the workspace) that connects to a simulation running on one or more server processes. Two steps are required to launch an interactive session of STAR-CCM+ for a new simulation:

1. Launch the STAR-CCM+ user interface (also called, the interactive client):
   - On Windows:
     - Start the client from the STAR-CCM+ entry in **Start** menu, assuming you did not change any of the defaults during the installation
   - On Linux:
     - First make sure that you know the full path to the STAR-CCM+ installation that you wish to use.
     - From the command prompt, enter the STAR-CCM+ command as shown in the following example:

       ```
       % [INSTALL_DIR]/starccm+10.02.003/star/bin/starccm+ &
       ```

       where [INSTALL_DIR] is the path to the installation directory.

   If everything is set up correctly, STAR-CCM+ displays a splash screen for several seconds, and then opens the user interface. At this point, the user interface, or client, is not connected to any server.
2. Launch a STAR-CCM+ server from the user interface:
   - When you want to create a new simulation:
     - Choose **File** > **New...**

     Alternatively, you can:
     - Click the **Create a File** button on the toolbar
     - Enter the keyboard shortcut, `<Ctrl><N>`.
   - When you want to load an existing simulation:
     - Choose **File** > **Load...**.

     Alternatively, you can:
     - Click the **Load a File** button on the toolbar
     - Enter the keyboard shortcut, `<Ctrl><L>`
     - Choose **File** > **Recent Files** followed by an entry in the list that appears
   - For both options, complete the dialog according to the properties described in Interactive Startup Reference.

Separate sections of the user guide provide detailed information on the client-server architecture that is used in STAR-CCM+, and also on running parallel servers for large simulations.

If you are using STAR-CCM+ for the first time, work through the Introductory Tutorial before going any further.

Interactive Startup Reference

STAR-CCM+ presents you with a dialog containing startup options when you create a simulation or load an existing simulation. These options control how STAR-CCM+ launches the server. Some options are specific to the *Load Simulation* dialog.

The launch dialog is divided into several groups:

- Process Options
- Saved Configurations
- Microsoft HPC Server
- License
- Server Connection Mode
- Simulation File Browse and Query (Load Simulation only)
- Command-line entry

Process Options

Serial

Launches a server on a single process on the local machine. This means that both the client and the server are running on your local machine. This setting is the default setting.

Remote Server

To launch a server in serial on a remote machine, activate the **Remote Server** checkbox. Complete the *Remote Host* and *Remote User* properties, and set the type of *Remote Shell* that STAR-CCM+ must use for the connection. If you previously set default values for the *Generic Remote Process Launch Command* in the *Options* dialog, these values are shown here.

Note: If you are using a remote server with a different user name, the **Connection Mode** of the simulation must be set to **Collaboration**.

To view files on the remote machine, activate **Use Remote File Browser**.

Parallel on Local Host

Launches a server running across the number of *Compute Processes* that you choose. STAR-CCM+ applies the default choice of Message Passing Interface (MPI) unless you specify an alternative within the *Command* line entry at the bottom of the dialog. See Supported MPI Implementations.

Parallel on Named Hosts

As for **Parallel on Local Host**, except that you can enter a list of Host names and the number of compute processes that STAR-CCM+ allocates on each.

Specify the machines by typing entries into the list. Click in the Host column of the first row of this list and type the first machine name. Press `<Tab>` or `<Enter>` to move to the next row.

In addition to the table format, this dialog contains buttons for modifying the list of hosts. Simply enter the name of the host in the **Add Host** text box and click the **Add** button. The **Delete** button deletes from the list a host whose name you select in the list. STAR-CCM+ stops using this host for parallel processing.

Verify all your hostnames before attempting to launch the simulation. If any of the hostnames is incorrect when you click the **OK** button, STAR-CCM+ generates an error message and no simulation is started.

If you frequently use the same set of machines for your parallel jobs, you can create a machine file for convenience.

Parallel Specified by Machine File

Launches a server running on machines that STAR-CCM+ selects, in order, from the chosen *Machine File*. For details on how you define this file, see Defining a Machine File. Set the number of *Compute Processes* for the server.

Saved Configurations

To save all the options on the current dialog:

- Enter a name within the text entry and click **Save**.

To reload an existing configuration:

- Select the configuration from the drop-down list.

Microsoft HPC Server

When this option is activated, a server command is submitted to the cluster using the Microsoft job scheduler. An argument, `-ccs`, is added to the *Command* property in the dialog to indicate this behavior. This functionality is available only on Windows Server 2012 machines.

License

Choose the correct license option for your organization:

- **Lite**: launches STAR-CCM+ with a limited set of capabilities. Not compatible with Power Session or Power-On-Demand. See Capabilities of STAR-CCM+ Lite
- **Power Session**: allows you to launch a STAR-CCM+ server with an unlimited number of compute processes.
- **Power-On-Demand**: allows you to use a STAR-CCM+ Power Session from a public license server rather than a license server on your own network. See Power-On-Demand Licensing.

Server Connection Mode

The following settings control whether and how other users may connect to the server of this simulation:

- **Default**: A user on the local area network (LAN) can connect using the user name of the person who started the server.
- **No Connections**: The server session is exclusive to the STAR-CCM+ client (or console) session that launched it. No other users can connect to that server. The **File** > **Disconnect** menu item is grayed out.
- **Collaboration**: Any STAR-CCM+ user on the LAN can connect to the server regardless of the user name.

Simulation File Browse and Query

The **Use Remote File Browser** checkbox, when activated, starts the file browser on a remote machine when you click the **Browse** button.

You would use this option if a remote server is specified and the remote machine does not share a file system with the local machine. However, sometimes it is faster to use the local file browser when the local and remote hosts do share a common file system (for example, NFS or Windows Shared Folders). The path to the simulation file must always be valid on the server host (controller host for parallel).

Command

This entry box shows the command line options that the client applies when you launch the server. Some options are defined by choices that you make in the launch dialog; other options, such as the choice of remote shell, are set within the *Options* dialog. For a list of command-line options, see the Command-Line Reference.

- If you edit the command line directly, STAR-CCM+ parses the options and updates its launch settings. For example, if you change "-np 2" to "-np 4", STAR-CCM+ updates the **Compute Processes** entry in the launch panel.
- Invalid edits to the command line are ignored.
- STAR-CCM+ retains changes that you make to launch settings for the rest of the client session.

Creating and Opening Simulations from the Command Line in Serial Mode

Using the `starccm+` command on its own, without any other options, launches the STAR-CCM+ client workspace.

Options can be added to the `starccm+` command to achieve more than just launching the client workspace.

- Creating a simulation connected to an interactive client
- Creating a simulation without an interactive client
- Loading an existing simulation in an interactive client
- Loading an existing simulation without an interactive client

Creating a New Simulation Connected to an Interactive Client

A single command-line expression can be used to:

- Launch an interactive client on the local machine
- Start a serial server with a new simulation on the local machine
- Connect the interactive client to the serial server

The command-line expression for these tasks is:

```
% starccm+ -new
```

Creating a New Simulation Without an Interactive Client

To create a simulation on a serial server, without launching an interactive client, use the `-server` option:

```
% starccm+ -server
```

The output of this command includes a line that provides the port number of the newly launched server. An example is:

```
% Server::start -host allegro.ourdomain.com:47830
```

To connect a client to this server, you would use the command:

```
% starccm+ -host allegro.ourdomain.com:47830
```

This action would launch a client on the local machine and connect it to the server running on `allegro`.

Loading an Existing Simulation In An Interactive Client

To launch an interactive client on the local machine:

- start a serial server on the local machine
- load an existing simulation into this server
- connect the client to the server

you can use the following command:

```
% starccm+ <file.sim>
```

where `<file.sim>` is the name of the simulation file you wish to load into the serial server.

Note: On Windows platforms due to a Windows Command Prompt limitation in character representation, ensure that the locale is set to the same language as the .sim filename that is being opened.

Loading an Existing Simulation Without an Interactive Client

To launch a serial server on the local machine and load an existing simulation into it, you can use the following command:

```
% starccm+ -server -load <file.sim>
```

The output from this command ends with a line containing the host and port numbers that are needed when connecting a client.

Starting Remote Servers from the Command Line

The functionality for starting a remote STAR-CCM+ server involves a simple command, around which you add options depending on what needs specified.

- Starting a server without an interactive client
- Starting a server with an interactive client

Starting a Server without a Client

The server can be started on a remote machine using a remote shell such as `rsh` or `ssh` in the same way as any other command. For example, using `ssh`:

```
ssh user@myremotemachine starccm+ -server
```

Make sure that remote shell access is permitted and configured.

To specify that a particular simulation file, in this example `file.sim`, be opened:

```
ssh user@myremotemachine starccm+ -server <absolute_pathname>/file.sim
```

Make sure that `file.sim` is in a path accessible from `myremotemachine` (the sim file must be readable by the server), and that the appropriate version of STAR-CCM+ is being started on `myremotemachine`.

If you wish to use a version of STAR-CCM+ other than the one to which your `$PATH` environment variable points, provide the absolute pathname:

```
ssh user@myremotemachine <absolute_pathname>/starccm+ -server
<absolute_pathname>/file.sim
```

Using a Client When Starting a Server

To start a local client (in this example Windows) along with a remote server:

```
starccm+.exe -rsh ssh -on user@myremotemachine <absolute_pathname>/
file.sim
```

- The client starts locally, then attempts to start the server remotely. The *Output* window includes something like the following:

  ```
  Starting remote server: ssh user@myremotemachine -n -o
  StrictHostKeyChecking=yes
  ```

  ```
  starccm+ -on user@myremotemachine -server <absolute_pathname>/
  file.sim
  ```

- The `-rsh` command is instructing the STAR-CCM+ client in the above example to use `ssh`.
- The `-on` is only relevant for the following purposes:
  - In parallel—it is used to specify the hosts to run workers on.

- In serial—it tells the client where to start a serial server.

You can also specify the path where STAR-CCM+ is installed:

```
"c:\Program Files\CD-adapco\STAR-CCM+ <version_number>\star\bin\starccm
+.exe" -rsh ssh -on user@myremotemachine <absolute_pathname>/file.sim
```

Restrictions on Names of Files and Paths in STAR-CCM+

The characters in the following table have restrictions on their use in the names of files or paths in STAR-CCM+.

Character	Restriction
\	Usable as a path separator character in Windows; not usable as a file name character.
/	Usable as a path separator character in Linux (and generally usable as a path separator in the Windows 32-bit API functions); not usable as a file name character.
:	Usable as a drive letter separator in Windows; not usable as a file name character.
!	Can cause problems if used in a file name in a shell, since this character recalls a previous shell command.
"	Not usable for file name characters.
'	
*	Not usable for file name or path name characters.
?	
|	
<	
>	
$	
&	

Also, on Windows, the strings listed below will not be recognized by STAR-CCM+ if you use them as entire file names. This is true for upper or lower case. For example:

- `AUX` is not recognized.
- `AUX.txt` is not recognized.

But:

- `AUXILIARY.txt` is recognized.
- `AUXILIARY` is recognized.

The strings are:

```
AUX
```

```
CLOCK$
```

```
COM1
```

```
COM2
```

```
COM3
```

```
COM4
```

```
COM5
```

```
COM6
```

```
COM7
```

```
COM8
```

```
COM9
```

```
CON
```

```
LPT1
```

```
LPT2
```

```
LPT3
```

```
LPT4
```

```
LPT5
```

```
LPT6
```

```
LPT7
```

```
LPT8
```

```
LPT9
```

```
NUL
```

```
PRN
```

Using the Simulation Pop-Up Menu

Save	Executes the save action to save the latest changes to disk.	
View Log	Executes the view log action and opens a dialog allowing you to view and edit the simulation log file.	
Summary Report	Makes a summary report of the simulation objects available through one of the following techniques.	
	Browser	Executes the view summary report action and displays the report in a web browser.
	To File...	Executes the save summary report action and generates a copy of the report in `.html` or `.xml` format.
Collapse Tree	Closes all open nodes in the object tree so that only the top-level manager nodes are visible.	

See also: common menu items.

Saving Simulations to File

A simulation file contains the complete set of simulation objects, state, and data for the problem.

It is the restart file for STAR-CCM+ and is stored with the `.sim` extension. Since it is a binary file, applications other than STAR-CCM+ cannot read it. The file can safely be moved, copied, and renamed using operating system commands as it is the only file that is associated with a simulation.

This section covers the topics that are relevant to saving simulations in a simulation file:

Contents:

Saving Over an Existing Simulation File
Saving a Copy of a Simulation
Saving Multiple Simulations
Renaming the Simulation
Using Backup Files
What Happens If a Disk Is Full?
Creating a Template Simulation File

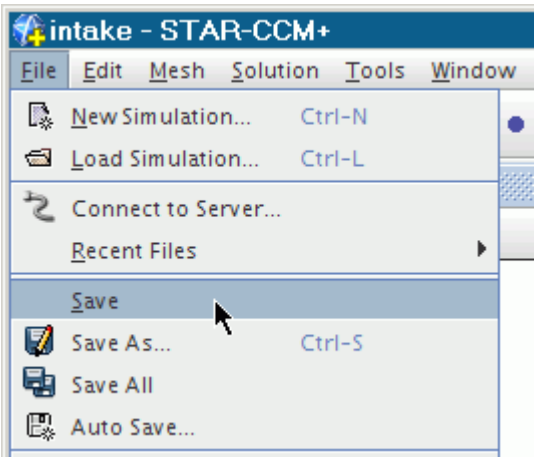
Saving Over an Existing Simulation File

Once you have changed a simulation, you can save it to a simulation file.

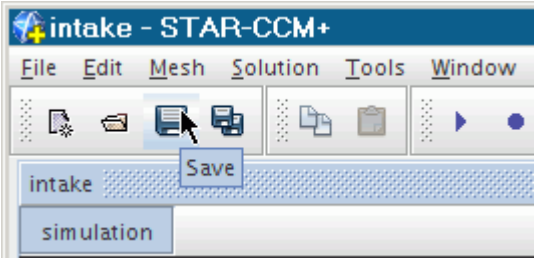
This action writes changes to the simulation file on the hard drive where the simulation resides. If the simulation being saved is new (has not been saved yet), then the save as action is executed. This action is only available when a simulation tree is active.

There are a number of ways to save a file:

- Selecting the **File** > **Save** menu option



- Clicking the **Save** toolbar icon

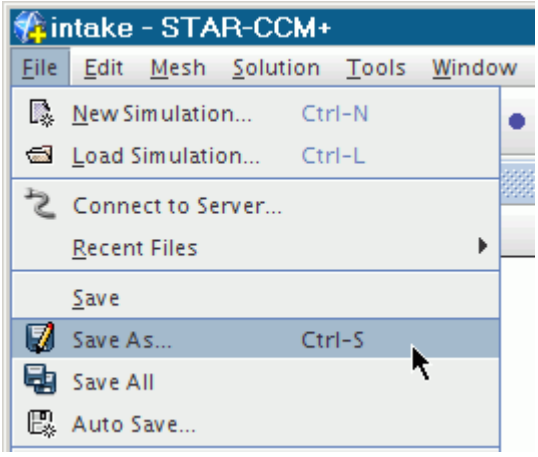


- Using the <Ctrl><S> keyboard shortcut

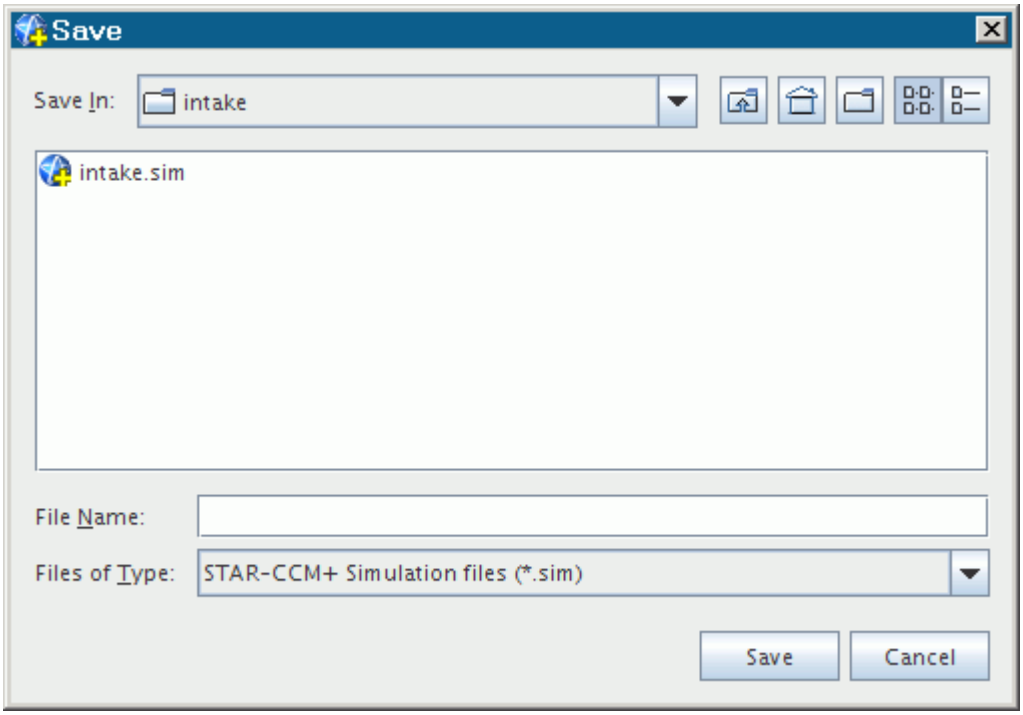
These methods are only available if you have changed the simulation since it was opened. If no changes were made, the menu option and the toolbar are grayed and the keyboard shortcut has no effect.

Saving a Copy of a Simulation

The **File** > **Save As** menu option allows you to save the simulation into a new simulation file.

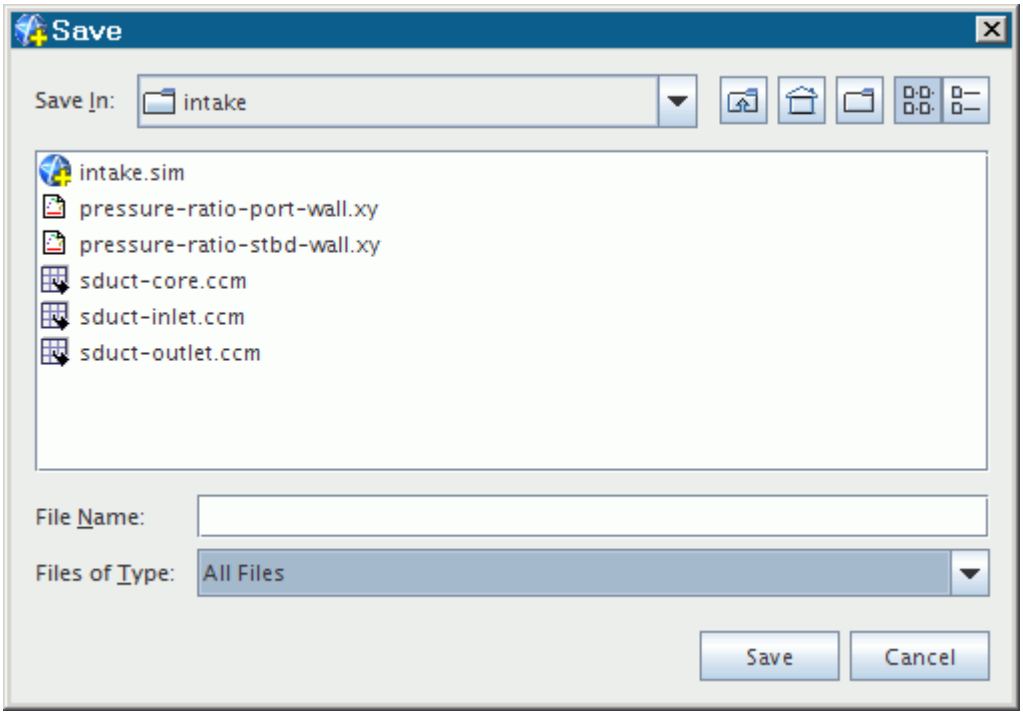


Selecting this option opens an operating system standard dialog to locate and name files to save.



The default directory is the last directory that was accessed. If none have been accessed, STAR-CCM+ uses the directory where the current simulation file is located, if applicable. Otherwise, STAR-CCM+ points to your `home` directory on Linux, or a comparable default directory on Windows.

This file save dialog only lists `*.sim` files by default. If you enter a new FileName, you do not have to enter the `.sim` FileName extension since it is automatically appended for you. It is still automatically appended even if you have already added a file extension to the name. Even if you select `All Files` in the Files of Type drop-down list, it is still appended.



- Entering `fred` saves the file as `fred.sim`
- Entering `fred.dat` saves the file as `fred.dat.sim`
- Entering `fred.sim` saves the file as `fred.sim`

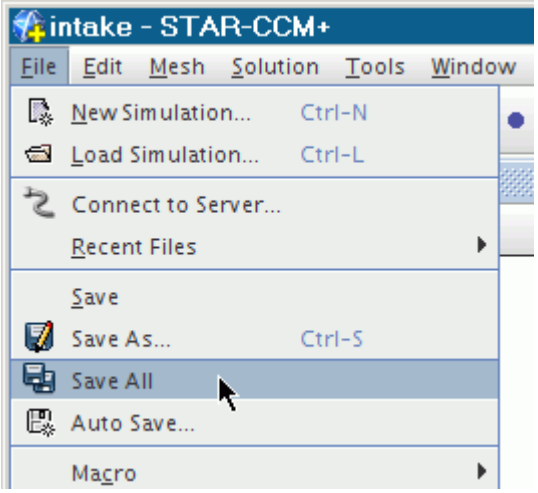
Unless you are saving a newly created simulation, the original simulation file remains in the original location, so that you are effectively making a copy of the original file.

Saving Multiple Simulations

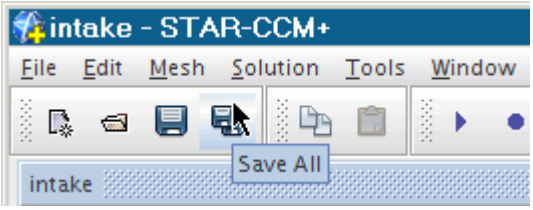
Once you have changed one or more open simulation files, you can save changes to them with a single action.

To save multiple simulations:

- Selecting the **File** > **Save All** menu option



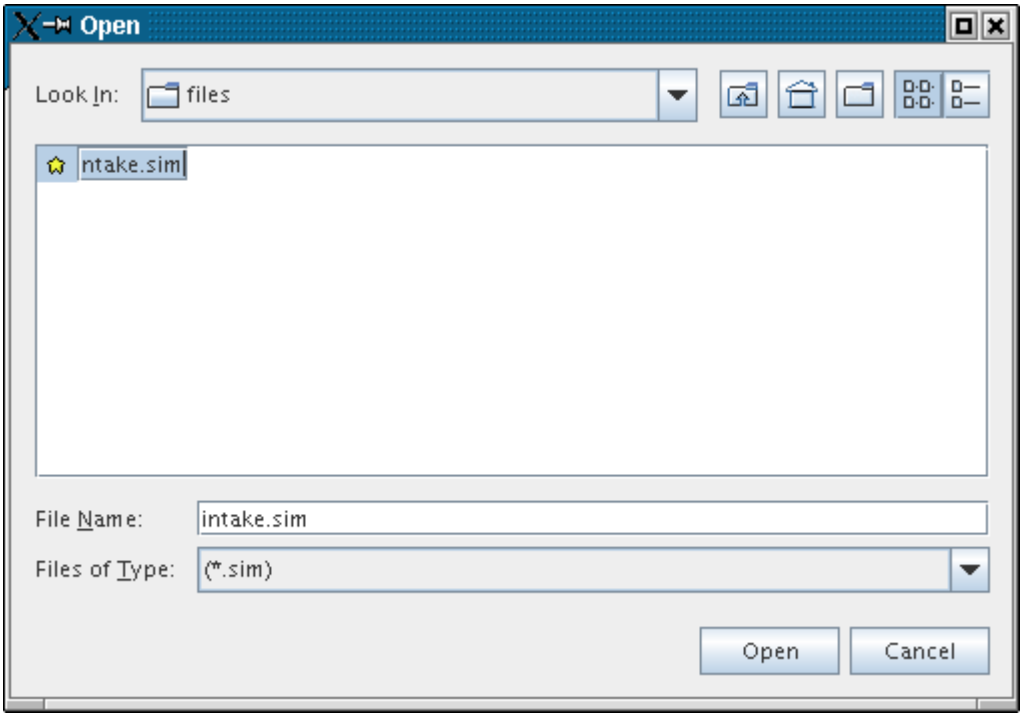
- Clicking the **Save All** toolbar button



These methods are only available if you have changed any simulation since it was opened. If no changes were made, the menu option and the toolbar button are grayed.

Renaming the Simulation

A simulation file can be renamed using the inline editor, except within the STAR-CCM+ client this technique works for simulation files only through slow clicking, and only within the *Open* dialog.



Do not, however, change the `.sim` extension as STAR-CCM+ has an association with files of that extension:

- The file open dialog only lists `*.sim` files by default.
- Opening a file that does not have a `.sim` extension does not open the file and display the simulation objects, even if the file is a simulation file.
- There is a Windows desktop association with the `.sim` extension that allows you to DoubleClick the file to open it in the STAR-CCM+ client.

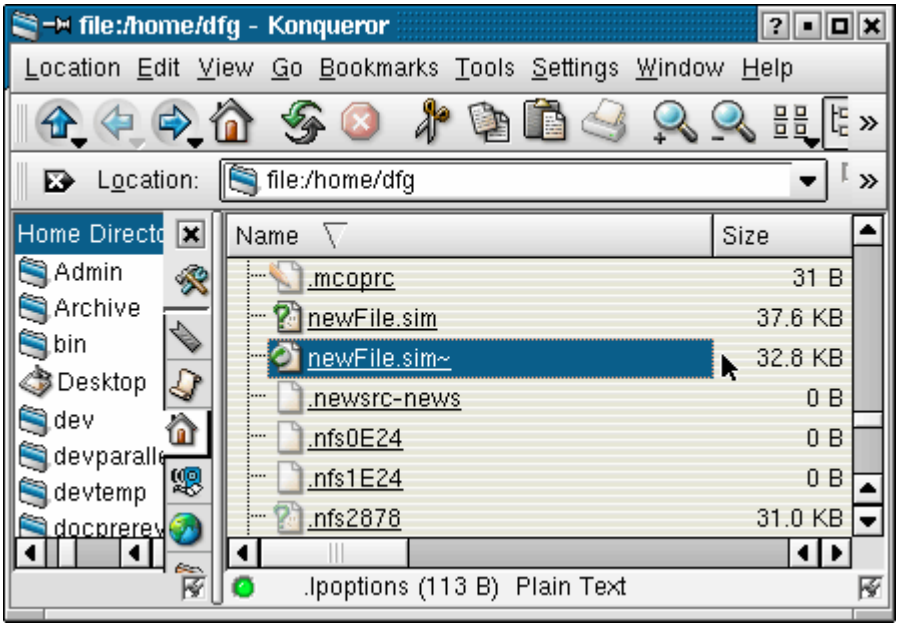
Using Backup Files

STAR-CCM+ generates a set of files that can come in handy as backups of your work.

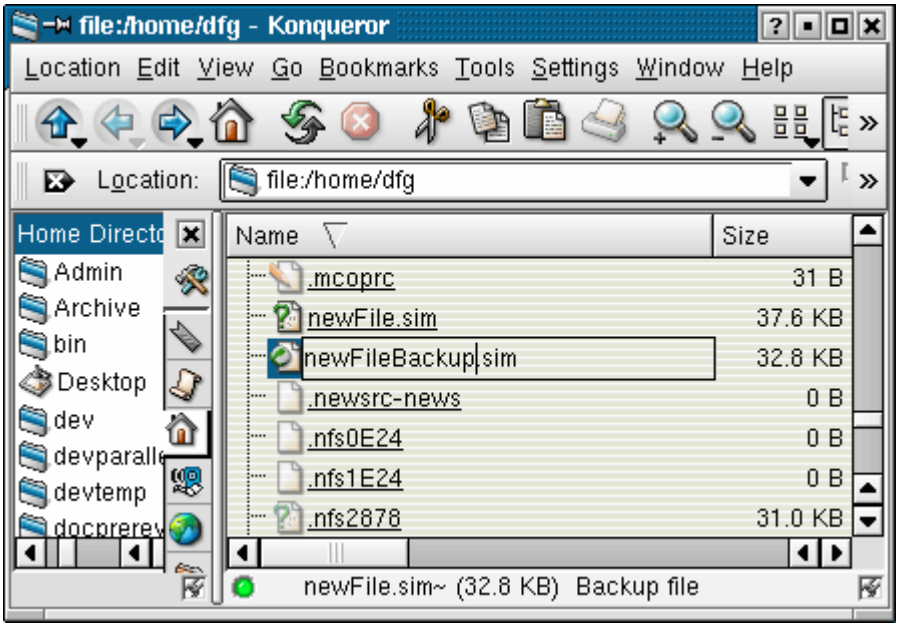
In your standard file-management software, along with your simulation files, sometimes there are more files in the same location with similar file names. Some file names only differ by a tilde character (~) added to each extension. For example, the files `intake.sim` and `intake.sim~` present in the same folder.

STAR-CCM+ automatically creates such a file once an existing simulation file is modified and saved; it is always the previous version of that simulation. Therefore if you wish to undo some changes after saving them to a simulation file, you can do so by opening the copy and saving it over the original simulation file.

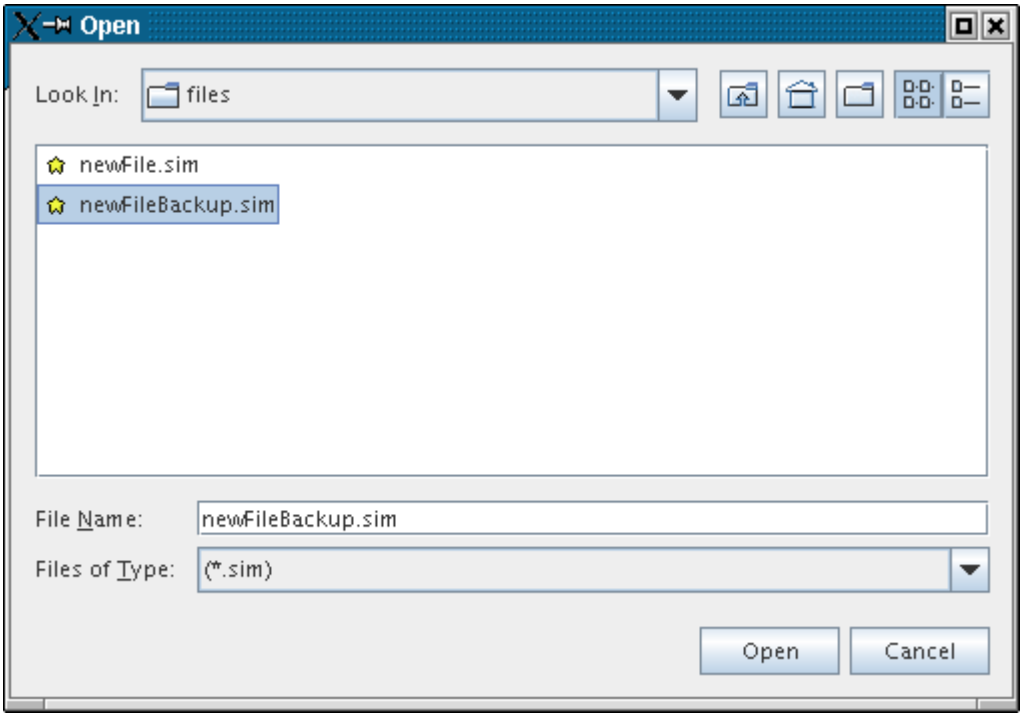
The STAR-CCM+ client does not always recognize this type of file, which prevents accidental opening and modification. However, it is easy to make the file accessible. Begin by selecting it in your file-management software.



Rename it in order to remove the tilde and give it a name different from the original simulation file.

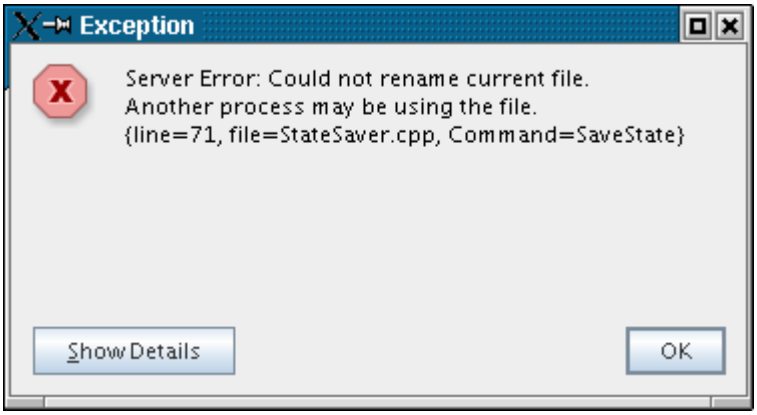


With the tilde removed, the node of the file appears in the *Open* dialog, which is activated by the **File** > **Load Simulation...** menu item.



This means that the file is now readily accessible to STAR-CCM+.

If you do not have write-access privileges for a simulation file, STAR-CCM+ does not only fail to save your changes, but also fails to create or modify the copy. That fact is reflected in an exception message like this one.



What Happens If a Disk Is Full?

If the disk becomes full while you save a simulation file, STAR-CCM+ notifies you that the file could not be saved due to insufficient disk space.

To save the simulation file, do either of the following:

- Save to a different volume.

- Remove data from the target volume and try to save again.

Creating a Template Simulation File

You can create a simulation file as a template for multiple cases. STAR-CCM+ lets you set up many elements of the simulation before generating a mesh, importing a mesh, or running the simulation.

These elements include:

- Setting up a mesh continuum
- Setting up a physics continuum
- Adding reports
- Adding scenes, monitor plots, and XY plots
- Creating regions and boundaries (when appropriate)

A template can be helpful when, for example, you want to use the same set of continuum models to compare meshes.

Regions and boundaries are appropriate for a template only when you are working with multiple cases that have the same region and boundary names. For example, the optimization of a pipe, in which the geometry stays the same while changes are made in the dimensions or the locations of bends.

Reverting Unsaved Changes to the Simulation

Use the Reload Simulation feature to return to the latest saved version of your simulation file.

To revert your unsaved changes to a simulation:

1. Select **File** > **Reload Simulation**.

   Note: When no simulation is open, or when a simulation is new or unchanged, this menu item is unavailable (grayed out).

2. To confirm your choice, click **Yes** in the *Reload Last Save* dialog.

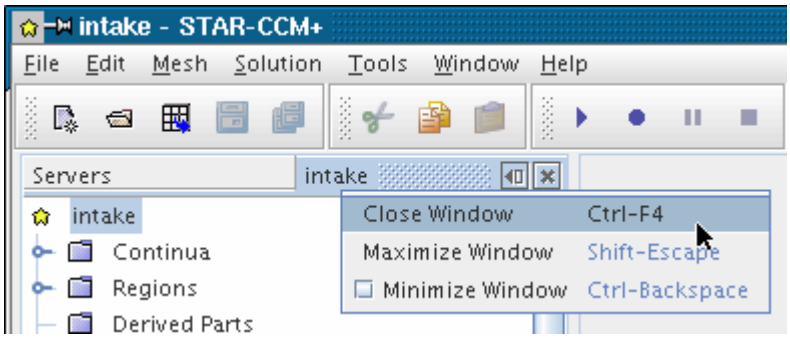
   The simulation file is reloaded in its last saved state.

Closing the Simulation

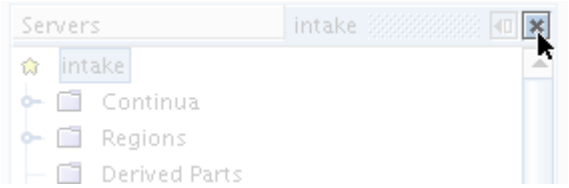
When you are done with a simulation, you can close it to free up the machine resources it has been using.

You can close the simulation with any of the following methods:

- Right-clicking the simulation window title and selecting **Close** from the pop-up menu;

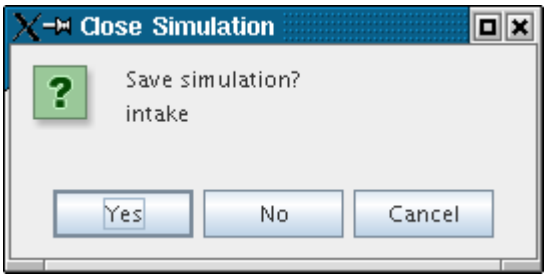


- Clicking the close icon on the simulation window title



- Pressing the `<Ctrl><F4>` keyboard shortcut

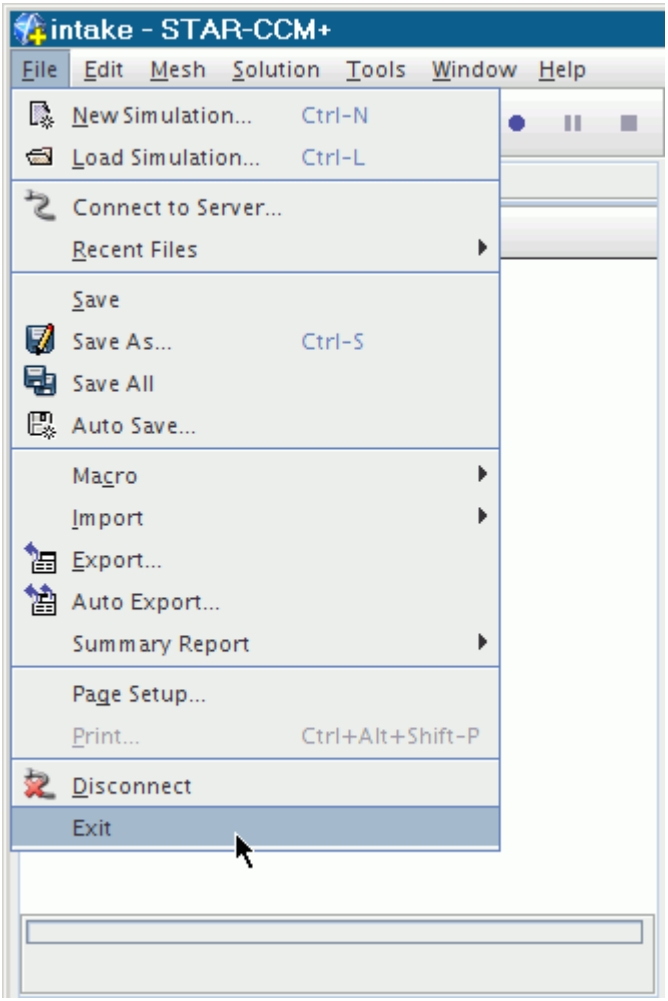
Each of these options executes the close window action. If anything in the simulation has been changed since opening or creating it, you are prompted to save the simulation:



Closing the Client Workspace

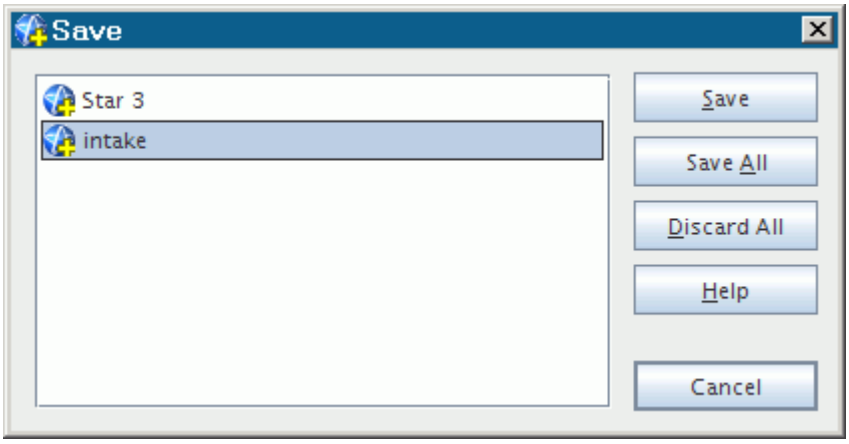
When you are done with simulations, close the client workspace to regain resources that it was using:

- Select **File** > **Exit** from the menus



- Press the `<Alt><F4>` keyboard shortcut

The exit application action is executed, which warns you with the following dialog when there are simulations still open and require saving.



Select an entry in the list and click **Save** to save a particular file and remove it from the list. To discard some files, save the files that are needed one at a time, then click **Discard All** when the list has been reduced to the files that you can discard.

Viewing and Editing the Log File

This section defines what is meant by a log file, and describes how to view a log file and add comments to one.

What Is a Log File?

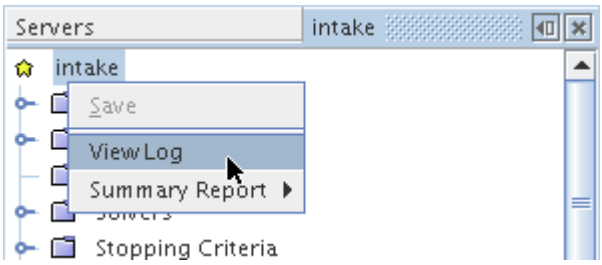
Whenever you save a simulation, an entry is added to a simulation log. The simulation log is a record of each save operation through the life of the simulation, which specifically records the following for every save operation:

- Date
- Time
- Version number
- Architecture that was used

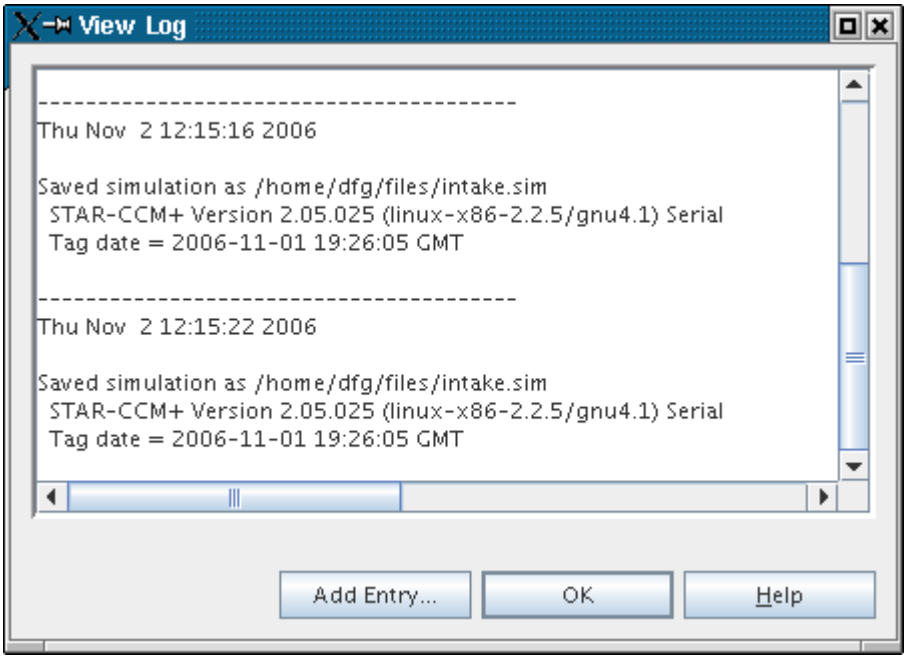
The information is stored in the simulation file itself, and not in a separate file. The log could be useful if you are checking compatibility issues between architectures, or between greatly disparate version numbers, for example.

Viewing the Log

You can view a log file by selecting the **View Log** menu item from the pop-up menu on the root node of the simulation tree:

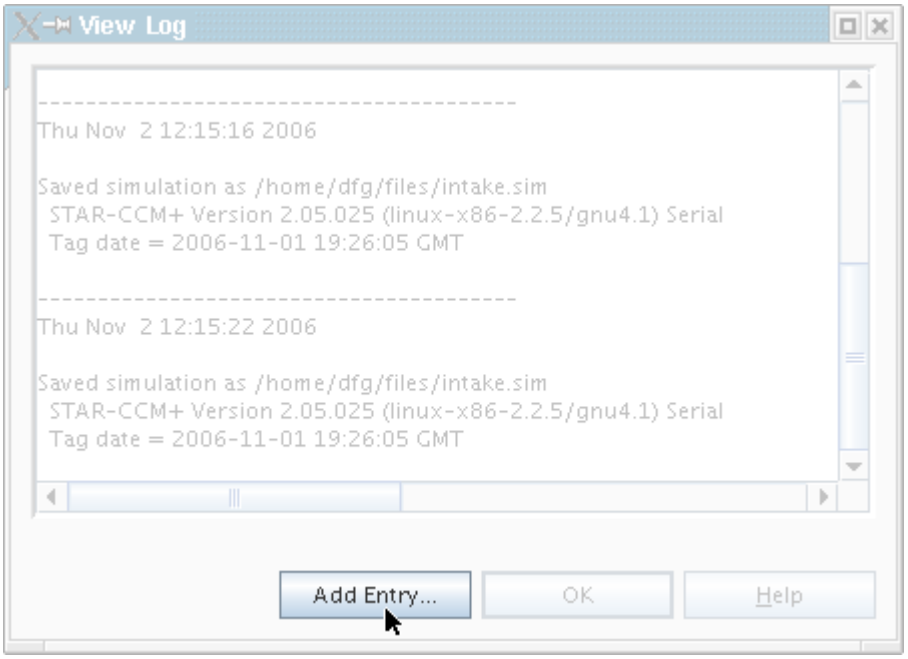


The view log action is executed, and displays the contents of the log in a separate window.

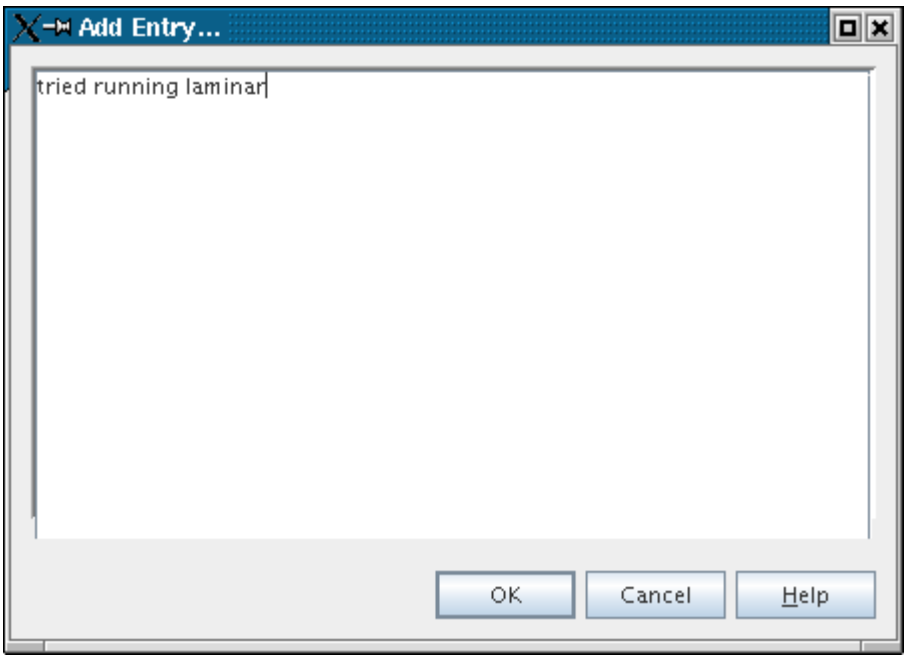


Adding Comments

You can also intersperse a log with comments of your own. If you want to indicate, for example, that you tried a particular CFL number, switched models, or imported a certain file type. Clearly, you do not get a record of the results at each of the log points (although you could add key results in the log yourself). However, these entries could be useful in the future when trying to remember the options that you have already tried. To add an entry to the log file, view the log and then click the **Add Entry...** button on the *View Log* dialog.



A simple text editor opens, allowing you to enter some text that is appended to the log.



Once you have accepted the editor dialog by clicking **OK**, you cannot edit or delete an added entry.

Using Summaries

You can create one or more summaries of a simulation, specify the objects to include in the summary, and generate HTML or XML output.

Custom summaries cover simulation objects that you select, such as a physics continuum, one or more regions, or any combination of them. A full summary contains basic information for the entire simulation.

Custom summaries are saved and restored as part of the simulation. Also, if you change any of the simulation objects, the next output from the summary reflects those changes.

A custom summary can contain the following types of information:

- Images of selected scenes and plots
- Properties of selected simulation objects, including:
  - Boundaries
  - Continua (physics and mesh) and their models
  - Coordinate systems
  - Geometry parts
  - Mesh operations
  - Motions
  - Part surfaces
  - Reference frames
  - Regions
  - Reports
  - Representations
  - User-defined field functions

Advantages of a customizable summary report include:

- It uses less time than traversing the object tree to compare settings among many objects, especially in a large simulation.
- Using identical summaries in multiple simulations allows easy comparison of the same type of object, such as a physics continuum or a region.
- You can include in a custom summary only the objects on which you want a report.
- You can define any number of summaries in a simulation.

Contents:

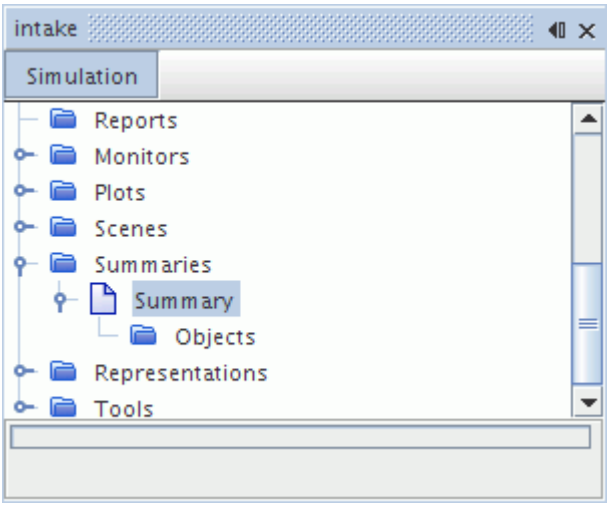
Generating a Custom Summary
Generating a Full Simulation Summary

Generating a Custom Summary

You can generate a summary of the parts of the simulation that are of interest.

1. Right-click the **Summaries** node and select **New**.

   A node with the default name **Summary** appears in the object tree with an **Objects** sub-node.



2. To specify objects for the summary, select objects using one of the following techniques:
   - Drag and drop elements from elsewhere in the object tree onto the **Objects** node.
   - Use the object selector dialog:
     1. Right-click the **Objects** node and select **Choose Elements**.
     2. In the object selector dialog, select the elements that you want included in the report. The object selector always shows all objects that are currently in the summary, even if they were added with the drag-and-drop technique.
3. To review your selections, expand the **Objects** node.
4. To remove some of your selections from the summary, do either of the following:
   - Right-click the object within the **Objects** node and select **Delete**.
   - Use the object selector dialog.
5. For easy reference to multiple summaries, rename each **Summary** node according to its content.

   Note: You cannot rename a summary sub-node that represents a simulation object, such as a continuum. However, if you rename the actual simulation object, the summary sub-node that represents it has the new name immediately.

6. If you have multiple summaries, you can copy or move sub-nodes between them.
7. To generate a summary report, right-click the summary node and select **Create Output File**.
8. In the standard *Save* dialog, navigate to the preferred location, enter a name for the HTML file in the **File Name** text box, and click **Save**.
9. To view the summary report, open the HTML file in the browser of your choice.

   In addition to the HTML file itself, a folder is generated in the same location, and with the same base filename, as the HTML file. This folder contains associated files, including any images.

   Note: To avoid breaking links, when you move the HTML file, move the associated folder along with it.

Generating a Full Simulation Summary

This type of summary contains information about a comprehensive range of objects that are related to the simulation.

You have a choice of two techniques for generating a full simulation summary:

- When using the custom summary feature, select all available objects. For details, see Generating a Custom Summary.
- Use a single step to create a simulation summary. The instructions for this option follow.

Generate a simulation summary directly in your web browser:

1. To generate a full simulation summary in one step, do either of the following:
   - Right-click the root node of the simulation tree and select **Summary Report** > **Browser**.
   - Select **File** > **Summary Report** > **Browser**.
2. View the summary in your default web browser.
3. If your browser fails to respond, change the browser path that is used by STAR-CCM+. See Changing the Default Browser.

Generate a simulation summary as an `.html` or `.xml` file on your hard drive:

4. To save the summary as a file, do either of the following:
   - Right-click the root node of the simulation tree and select **Summary Report** > **To File**.
   - Select **File** > **Summary Report** > **To File**.
5. In the standard *Save* dialog:
   a) Navigate to the preferred location.
   b) In the **File Name** text box, enter a name for the file.
   c) In the **Files of Type** drop-down list, select the file format.
6. Click **Save**.

You can also send the simulation summary to Siemens PLM Software support. See Sending a Simulation Summary Report.

Working With Parallel Servers

A parallel computation is one where the work is computed by more than one process.

In distributing the parallel processes across multiple cores, the time that is taken to complete the work can be reduced. It is highly desirable to compute CFD solutions in parallel because of the computationally intensive work that is required to acquire converged solutions. STAR-CCM+ can be run in parallel mode providing a scalable solution to solving problems faster and to attempting larger problems.

The core of this functionality is based upon a data parallelism paradigm in which the mesh is distributed across a number of processes. Each process applies numerical methods identical to those used in a serial solution. Communication between the processes ensures consistency with a serial solution.

This section describes the following:

Contents:

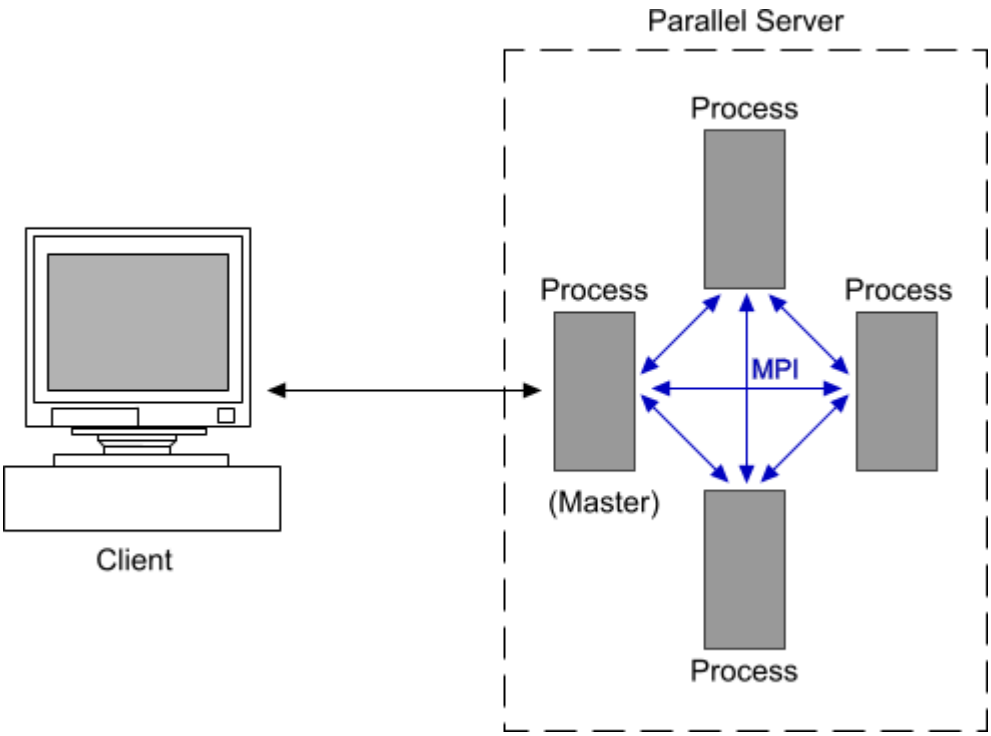
Parallel Servers and Requirements
Supported MPI Implementations
Using a Local Multi-Core Workstation
Using a Remote Multi-Core Workstation
Using a Group of Machines
Troubleshooting
Balancing
Getting Parallel Help through the Command Line
Viewing the Partitioning
Using Batch Management Systems
Cluster Node Communication
Expert MPI Options
Running Distributed Parallel Simulations on Multiple Windows Machines

Parallel Servers and Requirements

A parallel server is a STAR-CCM+ server that is set to run on multiple cores or a cluster of machines when it is first launched.

Launching a server in parallel is a transparent process in STAR-CCM+. You can launch STAR-CCM+ in parallel either through the command line or through the client GUI.

The parallel server is composed of N processes. Each process computes an equal share of the computational workload. The client only ever communicates with the first process in the parallel server which is referred to as the master process. This master process participates equally in the computational workload.



Processes communicate with each other by passing messages. These messages conform to a programming standard known as the Message Passing Interface (MPI). STAR-CCM+ is designed to work with different implementations of MPI, such as IBM Platform MPI, and Microsoft MPI. One or more of these implementations have been installed on your system along with STAR-CCM+. You may also have access to vendor-specific implementations, depending on your system. A command line option, `-mpi`, allows you to select a particular implementation when you launch the parallel server.

Requirements for Running in Parallel

In order to run in parallel, STAR-CCM+ must be installed on each of the machines you wish to run on. On Windows machines, make sure that MPI is installed on each machine, and that the STAR-CCM+ installation path is the same on all the machines. On Unix machines, make sure that you have remote shell access to each machine.

Currently, STAR-CCM+ only supports homogeneous parallel servers so, for example, you cannot have some processes running on a Windows machine, and other processes running on Linux machines. The client process does not need to be on the same architecture as the server processes.

Also make sure that you have the correct number and type of licenses that are required for a parallel server.

Hyper-Threading

Generally, for best performance, it is recommended that you turn off any features that artificially increase the number of cores on a processor, such as hyper-threading. In situations where hyper-threading is turned on to benefit other applications, you should generally avoid loading more processes than there are physical cores.

In addition to turning off hyper-threading, it is recommended that you set your BIOS settings to favor performance rather than power saving. Almost all heavily parallelized applications suffer performance problems if the CPU frequencies are spun up and down. In most parallel applications, and STAR-CCM+ in particular, when one core is operating at a reduced frequency, all other cores are running at lower frequencies as well.

See Also:

Supported MPI Implementations

Supported MPI Implementations

Regardless of whether you use a local multi-core workstation, remote machines, or a remote cluster to run your parallel server, all approaches require an implementation of the Message Passing Interface (MPI).

The default MPI for Windows and Linux is IBM Platform MPI and is supplied with STAR-CCM+. For Windows Server 2012, you are advised to use Microsoft MPI, which is suppled with the operating system.

You can select an alternative implementation of MPI using the `-mpi` argument on the command line. For example:

```
-mpi msmpi
```

selects the Microsoft MPI distribution on Windows. Command line options for supported distributions are given in the tables below.

Windows

MPI	Version	Supplied with STAR-CCM+	Command line option
IBM Platform MPI (default)	9.1.4.4	Yes	`platform`
Intel MPI	2017.2	Yes	`intel`
Microsoft MPI	7	No	`msmpi`

Linux

MPI	Version	Supplied with STAR-CCM+	Command line option
IBM Platform MPI (default)	9.1.4.3	Yes	`platform`
Open MPI	1.8.0 to 1.8.4	No	`openmpi`
Intel MPI	2017.2	Yes	`intel`
SGI MPI	2.11 and newer	No	`sgi`
Cray MPI	7	No	`crayxt`

Some notes on these distributions are as follows:

- IBM Platform MPI is a commercial product and is considered one of the best MPI distributions on the market. It supports many different Interconnects such as Infiniband (using different drivers OFED, QLogic, PSM) and Myrinet. IBM Platform MPI uses the fastest network channel that it detects without any use specification.

- Open MPI is an actively developed and widely used open source MPI distribution. Open MPI supports a range of interconnects. However, due to limited access to the full range, STAR-CCM+ provides a mechanism to use locally installed Open MPI installations that are built to support specific hardware.
- Intel MPI is a mature MPI that has become more competitive in recent years. Resident memory usage with Intel MPI is 40-50% less compared to other MPIs.
- SGI MPI is an MPI distribution that is optimized for SGI architectures.
- Cray MPI is required when running on Cray clusters.

Contents:

Using Open MPI
Using Cray MPI
Selecting the Fabric
CPU Binding
Registering Your Credentials for Distributed Simulations on Windows
Using Alternative Versions of IBM Platform MPI

Using Open MPI

Open MPI is an actively developed and widely used open source MPI distribution. Although STAR-CCM+ does not include a distribution within its installer, STAR-CCM+ does support several methods by which you can specify a locally installed version for it to use.

Open MPI has support for shared memory, TCP, and OpenIB interfaces.

To find the MPI libraries, STAR-CCM+ checks locations that are defined by:

- `OPENMPI_DIR` environment variable
- `OMPI_HOME` environment variable
- Default installation directories under `/opt`. In this case, STAR-CCM+ attempts to use the latest version available.

To use Open MPI libraries at runtime, include either of the following options with the `starccm+` command:

- `-mpidriver openmpi`
- `-mpi openmpi`

Open MPI command line options can be passed into STAR-CCM+ using any of the following options:

- `-mpiflags <mpirun options>`
- `-mpidriver openmpi:<mpirun options>`
- `-mpi openmpi:<mpirun options>`

Open MPI Runtime Dependencies

Some Open MPI distributions have a runtime dependency on the rdmacm communication manager. If this library is not installed on your system, warning messages appear in the console window at runtime.

Often this runtime library is installed by your network fabric. If this library is not installed, your system administrator can configure your system using the `yum` command:

```
yum install librdmacm.x86_64
```

Using Cray MPI

STAR-CCM+ supports the Cray Extreme Scaling Mode through Cray's optimized MPI libraries on Cray's XT/XE/XC machines.

The following prerequisites must be satisfied in order to run STAR-CCM+ on Cray XT/XE machines:

- STAR-CCM+ must be installed on the Lustre file system.
- The Data Virtualization Service (DVS) must be running on the cluster so that the compute nodes can access both the STAR-CCM+ installation and a fully functional Linux OS environment. (Without DVS running the compute node's Cray Linux Environment only provides a subset of Linux functions which do not satisfy the STAR-CCM+ runtime requirements).

The native Cray MPI libraries are specified at runtime with the command line options:

```
-mpidriver crayxt
```

or

```
-mpi crayxt
```

By default STAR-CCM+ attempts to locate the native MPI libraries in default locations. If the environment variable `CRAY_MPICH2_DIR` is set it will be used as the default location of the installed MPI (this variable will typically be set when the `cray-mpich` or `cray-mpich2` Linux module has been loaded, either explicitly by the user or by default by the system). Cray MPI supports applications built with different compilers. STAR-CCM+ must use the Cray MPI libraries built using the GNU compilers. When specifying the location of the installed MPI using the Linux module functionality, make sure that the GNU compiler build of Cray MPI is selected. Alternatively, STAR-CCM+ will attempt to locate the Cray MPI library under the directory `/opt/cray/mpt/default/*/mpich2-gnu`.

You can specify the location of the native MPI libraries using an environment variable, for example:

```
setenv CRAYXT_HOME /opt/cray/mpt/default/xt/seastar/mpich2-gnu
```

Ensure that the version of Cray MPI specified is compatible with STAR-CCM+.

Because Cray MPI is tightly coupled with the job scheduler, when launching STAR-CCM+ you must specify the number of processes using the `-np` command line option. You do not have to provide a list of hosts using either the `-machinefile`, `-on`, or `-batchsystem` command line options.

The `-mppflags` command line option passes additional flags directly onto the `aprun` command. `aprun` is similar to the mpirun command on other systems.

An example PBS submission script used on a Cray XE system is shown below:

```
# Change to the directory that the job was submitted from
cd $PBS_O_WORKDIR
```

```
# specify the location of MPI
export CRAYXT_HOME=/opt/cray/mpt/default/gni/mpich2-gnu/47
# query the scheduler to get the total number of processes
np=$(qstat -f $PBS_JOBID | grep Resource_List.mppwidth | awk '{print
$3}')

$HOME/myworkspace/STAR-CCM+9.06.011/star/bin/starccm+ -mppflags "-N 16
-d 2" -np $np -collab -mpidriver crayxt -batch $HOME/myworkspace/data/
mycase/run.java $HOME/myworkspace/data/mycase/mysim.sim
```

In this example, two additional flags were passed to the `aprun` command: `"-N 16 -d 2"`. The first option `-N 16` specifies to allocate 16 STAR-CCM+ server processes per node on the cluster. The second option `-d 2` specifies the number of cores reserved for each STAR-CCM+ server process. The two options combined request a full node that will be semi-subscribed by STAR-CCM+ processes. This has been found to be optimal when using systems based upon the AMD Interlagos chip where each integer unit on the chip is considered as a core. Because each two integer units on Interlagos share one floating point unit, the `-d 2` option is a more efficient ratio to use for STAR-CCM+.

On Intel-based systems, no semi-subscription is required -- either omit `-d`, or use `-d 1`. Some Intel-based systems also have a feature called hyper-threading, which gives two virtual cores per physical core. STAR-CCM+ typically will run more efficiently without hyper-threading. On the `aprun` command flags, either omit `-j`, or use `-j 1`, to disable hyper-threading.

To submit the STAR-CCM+ job to the PBS scheduler, enter the following in the command line:

```
qsub -l mppwidth=128 mysubmitscript
```

where in this example 128 STAR-CCM+ processes are requested.

Cray systems are often installed as part of a large HPC facility. The rules, settings, and procedures may differ for each site. Users should always refer to any local documentation when installing STAR-CCM+.

Selecting the Fabric

The *fabric* describes the interface between the MPI layer and the network hardware. For instance, TCP is the fabric interface to Ethernet. Infiniband hardware can be interfaced using different software layers, each described using a different fabric selection.

Each MPI implementation defaults to the most suitable fabric. In most cases, you do not need to specify the fabric. You need to specify the fabric only if the MPI distribution fails to automatically pick the correct fabric or if you wish to use a non-default fabric. The fabrics available for selection depend upon their availability on the system, and the MPI implementation being used. For example, if the system has Myrinet MX hardware installed, it is unlikely that the QLogic PSM interface is available. In this case, the default MPI should detect the Myrinet MX hardware and automatically select the correct fabric at runtime.

On Cray systems, Cray MPI must be used and you have no control over fabric selection. On these systems, the locally installed MPI distribution is tuned with the hardware to provide optimal performance.

Fabric selection is not supported when using SGI MPI at this time. SGI MPI is tuned for optimal performance on SGI architectures.

The fabric can be specified using the command line option:

```
-fabric <fabric>
```

If you enter the fabric option in uppercase (e.g. `-fabric TCP`), this specifies no fall-back. If you specify the fabric option in lowercase (e.g. `-fabric tcp`), STAR-CCM+ falls back to an alternative fabric if the specified fabric is not available, and if an alternative fabric is available.

Fabric Option	Fabric
`-fabric TCP`	Ethernet
`-fabric IBV`	Infiniband OpenFabrics Alliance Verbs API (IBV)
`-fabric UDAPL`	Infiniband User Direct Access Programming Library (UDAPL)
`-fabric PSM`	Infiniband QLogic Performance Scaled Messaging (PSM)
`-fabric GM`	Myrinet GM
`-fabric MX`	Myrinet MX

The `-fabric <fabric>` option is equivalent to the `-mpp "-<fabric>"` option used in earlier scripts for HPC clusters. Thus `-fabric TCP` is equivalent to `-mpp "-TCP"`.

The`-fabric <fabric>` option is equivalent to the IBM Platform MPI options passed using the `-mpp "-<fabric>"` option. Thus `-fabric TCP` is equivalent to `-mppflags "-TCP"` in IBM Platform MPI.

To show the fabric selected by MPI, where applicable, use:

```
-fabricverbose
```

CPU Binding

CPU binding constrains a simulation to run only on a particular set of CPUs or cores, and not to migrate to others during the life of the simulation.

In the STAR-CCM+ command line, you use the `-cpubind` option to specify the default CPU binding. CPU binding is turned on by default as binding processes can improve the performance of STAR-CCM+.

This option applies to both serial and parallel cases on Windows and Linux. In serial, the option pins the serial server process to the core that is assigned to the process by the scheduler. In parallel, the server process on each host is bound to the cores alternating between available sockets on multi-socket chips.

For parallel simulations, there are two binding policies that you can specify using the following argument:

```
-cpubind <policy>
```

Where `<policy>` is:

- `bandwidth` - default option that allocates processes to cores alternating between sockets
- `latency` - allocates processes to all cores on the same socket before allocating to additional sockets

If you do not specify a policy in the argument, the default policy is used.

This option is equivalent to these options in earlier scripts:

- IBM Platform MPI: `-mppflags "-aff=automatic:bandwdith"`
- Intel MPI: `-mppflags "-binding pin=1;cell=core;map=scatter"`
- Open MPI: `-mppflags "-bind-to-core -bysocket"`

To request a report of the CPU binding in parallel, type:

```
-cpubind v
```

or

```
-cpubind <policy>, v
```

When STAR-CCM+ identifies another STAR-CCM+ server running on any of the hosts that are being used, it disables any requested CPU binding for all processes on all hosts that are being used. This avoids processes being bound to identical cores, which can result in significant performance degradation.

You can manually deactivate CPU binding by typing the following argument in the command line while launching STAR-CCM+:

```
-cpubind off
```

Registering Your Credentials for Distributed Simulations on Windows

You must register your credentials before running distributed simulations on multiple nodes.

This section describes how to:

- register your credentials on non-HPC Server versions of Windows
- register your credentials on Windows HPC Server 2012 R2

Registering Your Credentials on Non-HPC Server versions of Windows

To register your credentials on non-HPC Server versions of Windows:

For IBM Platform MPI:

1. Start a **Command Prompt** from the **Start** menu.
2. Enter the command:

   ```
   <path>\mpidiag -s 127.0.0.1 -cache -at
   ```

   Where `<path>` is the full path to `mpidiag`, for example, `C:\Program Files\CD-adapco\11.06.###\STAR-CCM+11.06.###\mpi\platform\9.1.4.2\windows\bin`
3. Enter your password when requested.

For Intel MPI:

1. Start a **Command Prompt** from the **Start** menu.

2. Enter the command:

```
<path>\mpiexec.exe -register
```

Where `<path>` is the full path to the `mpiexec.exe` executable file in your STAR-CCM+ installation directory.

The following is an example of a typical path for a 64-bit installation using STAR-CCM+ version 12.06.###:

```
C:\Program Files\CD-adapco\STAR-CCM+12.06.###\mpi\intel
\5.0.3.048\win64\bin
```

3. Enter your username and password when requested.

Registering Your Credentials on Windows HPC Server 2012 R2

To register your credentials on a Windows HPC Server 2012 R2 machine:

1. Launch STAR-CCM+ with the desired number of cores. See Launching STAR-CCM+.

   If your credentials are not already registered, the *Connect to [host_name]* dialog appears.

2. Enter your user name and password.
3. To store these credentials, activate the **Remember my password** option.
4. Click **OK**.

Some additional points must be noted when using IBM Platform MPI on Windows Server 2012 machines for distributed simulations on multiple nodes.

- You must launch STAR-CCM+ from within a PowerShell command window.
- You must define the current working directory as a UNC path that includes the name of the head node on which you are launching STAR-CCM+.

An example of a UNC path is:

```
\\hpcmachine\WORKAREA\fluidsims\case1
```

- The working directory must be shared and accessible from all compute nodes.
- If you used the `-NOADMIN` command line option to install STAR-CCM+ without having administration privileges, you may receive the following error message when you attempt to register your credentials:

```
connect() to IP 127.0.0.1/port 8636, failed:
GetLastError(10061)Socket Connect failed: 10061
```

Your credentials are still registered even though this message appears.

Using Alternative Versions of IBM Platform MPI

On Linux platforms, STAR-CCM+ can be configured to use an alternative version of IBM Platform MPI to the one that is included in the STAR-CCM+ installer. This facility is useful if your cluster requires features that are not supported by the version that is provided in the installer. This facility also allows older versions of STAR-CCM+ to use more recent versions of IBM Platform MPI.

Provide the location of the alternative installation of IBM Platform MPI using the environment variable, `PCMPI_DIR`.

Only use this feature when the bundled version of IBM Platform MPI prohibits the use of the machine.

To create the environment variable, enter the following in the command prompt:

- For BASH/KSH shells:

```
% export PCMPI_DIR=<path to IBM Platform MPI installation root
directory>
```

- For BASH CSH shells:

```
% setenv PCMPI_DIR <path to IBM Platform MPI installation root
directory>
```

Using a Local Multi-Core Workstation

If a workstation is a multi-core machine it is convenient to exploit these resources with STAR-CCM+, either from the workspace or from the command line. Typically, you can experience a significant decrease in the time that is required to reach a solution, although limits on the parallel efficiency can be experienced due to the sharing of system resources under this architecture.

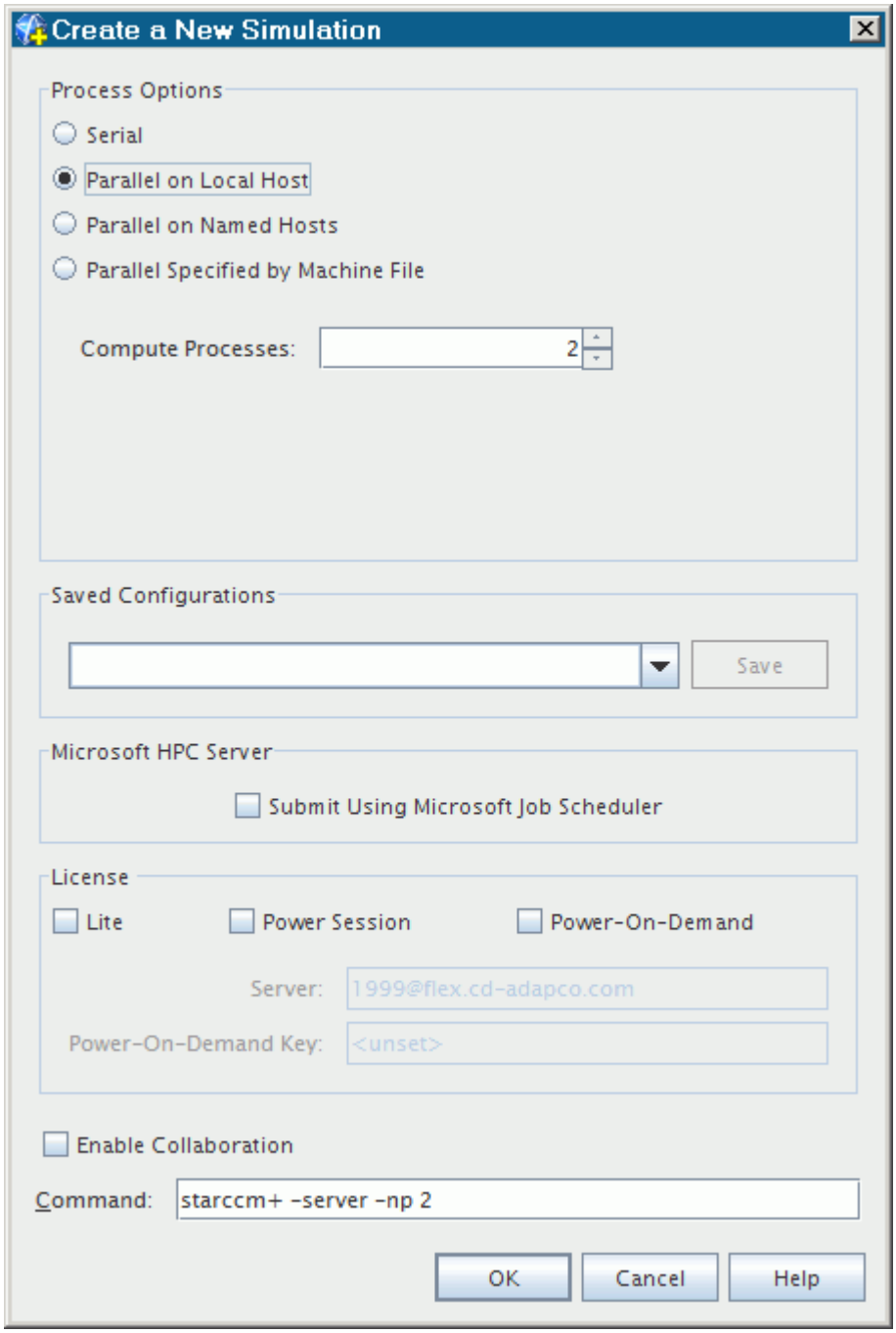
See Also:

Launching STAR-CCM+

Starting from the Workspace

To start a simulation on a cluster using the workspace:

1. Launch the workspace.
   - If you are starting a new simulation, select **File** > **New Simulation...** to activate the *Create a New Simulation* dialog.
   - If you are opening an existing simulation, select **File** > **Load Simulation...** to activate the *Load Simulation* dialog.
2. By default serial mode is selected, so click the **Parallel on Local Host** radio button to change to parallel mode.



3. Enter the number of processes in the **Compute Processes** box. You can also increment this value using the arrows. The equivalent command line which is used to start the server is displayed in the **Command** text box.
4. Click **OK** to accept the settings. Now, any simulation that you open runs the server in parallel mode with the specified number of processes on the local host.

Starting from the Command Line

If your machine has 2 cores, two processes can be started using the `-np` option. The full command line would be:

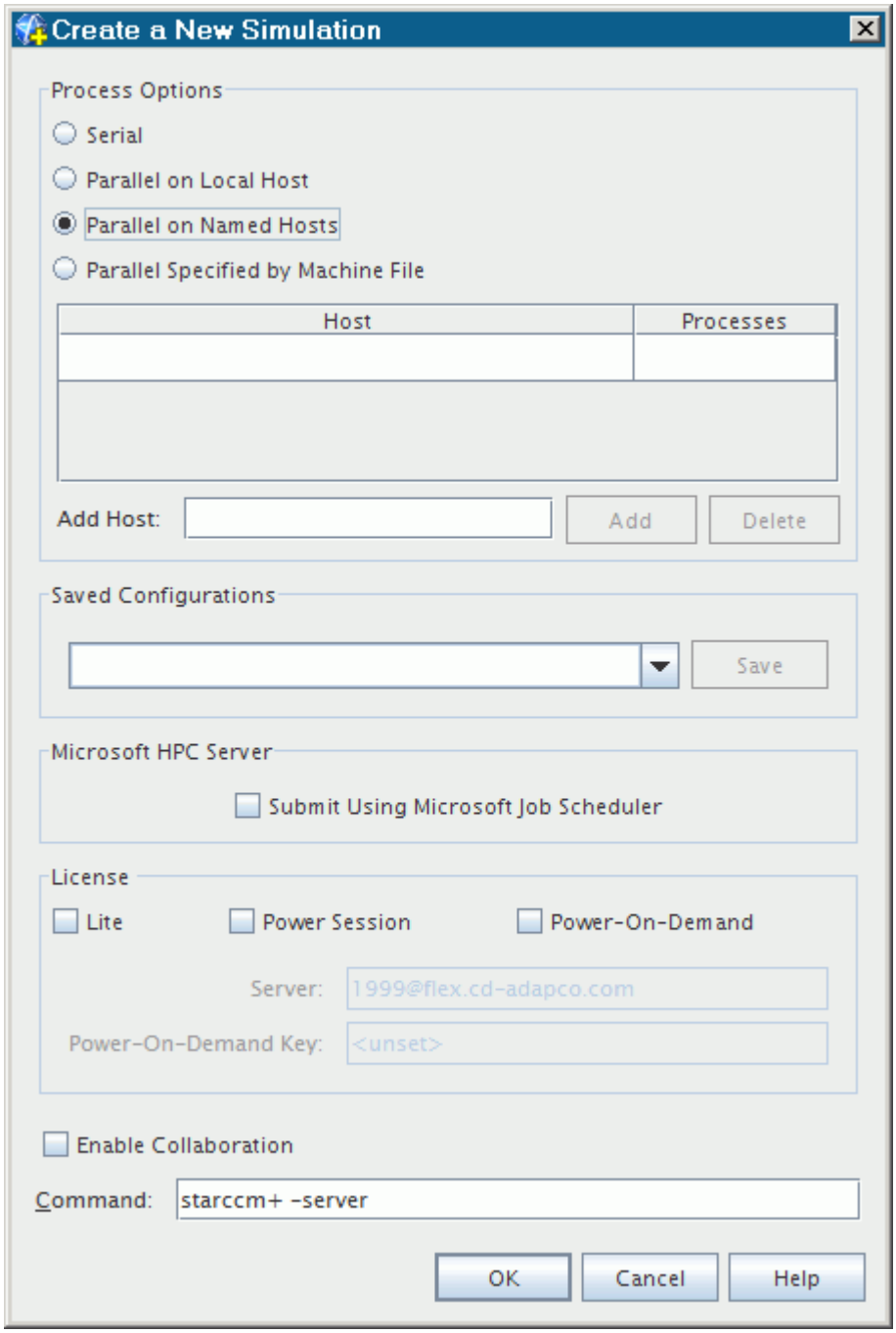
```
starccm+ -np 2
```

In this simple case, you do not need to specify the `-on` option or the `-machinefile` option. STAR-CCM+ presumes that you are referring to the local host when you omit these options.

Using a Remote Multi-Core Workstation

To start a simulation on remote machines using the workspace:

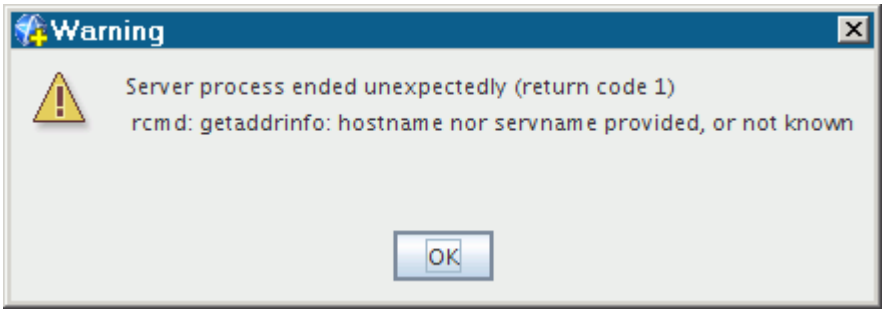
1. Open the workspace.
   - If you are starting a new simulation, select **File** > **New Simulation...** to activate the *Create a New Simulation* dialog.
   - If you are opening an existing simulation, select **File** > **Load Simulation...** to activate the *Load Simulation* dialog.
2. By default serial mode is selected, so click the **Parallel on Named Hosts** radio button to specify the host machines for parallel mode. This option displays a small spreadsheet where you can enter the list of hosts.



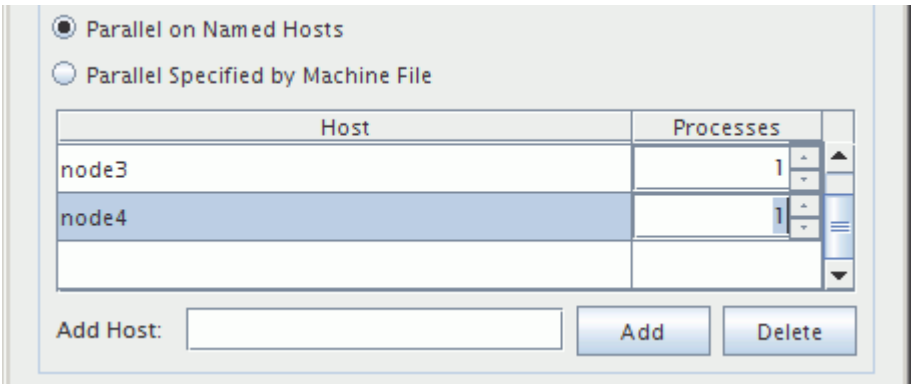
3. Specify the machines by typing entries into the list. Click in the Host column of the first row of this list and type the first machine name. Press `<Tab>` or `<Enter>` to move to the next row.

In addition to the table format, this dialog contains buttons for modifying the list of hosts. Simply enter the name of the host in the **Add Host** text box and click the **Add** button. The **Delete** button deletes from the list a host whose name you select in the list. STAR-CCM+ stops using this host for parallel processing.

Verify all your hostnames before attempting to launch the simulation. If any of the hostnames is incorrect when you click the **OK** button, STAR-CCM+ generates an error message and no simulation is started.



4. Specify the number of processes in the Processes column. Add the other machine names in the same fashion until you are done.



5. Accept the completed dialog by clicking the **OK** button. Now, any simulation that you open runs the server in parallel mode and distributes the processes on the machines specified. The parallel controller process is started on one of the hosts by default.

See Also:

Launching STAR-CCM+

Using a Group of Machines

A group of machines can also be utilized in parallel mode, using the workspace or the command line. It is also possible to start a server allocated across a group of machines manually. This facility provides a much more scalable solution if you wish to decrease significantly the time that is required to reach a solution while also increasing the total memory available for a simulation. In addition, you can edit hostnames of machines based on your network requirements. With the Power Session license feature, you can run one job on an unlimited number of machines. Finally, for machines that support it, a parallel I/O option is available to improve the save and restore performance.

Information is provided separately about how STAR-CCM+ works with specific network devices. You can start a proxy server on the cluster to expose running simulations to workstations outside the cluster's private network.

Contents:

How Many Cluster Nodes Should I Use for a Simulation?
Defining a Machine File for Parallel Hosts
Starting from the Command Line
Starting a Cluster of Machines Manually
Editing Hostnames of Machines
Using the Power Session License Feature
Using Parallel I/O

See Also:

How Many Cluster Nodes Should I Use for a Simulation?
Starting from the Command Line
Starting a Cluster of Machines Manually
Using the Power Session License Feature
Editing Hostnames of Machines
Using Parallel I/O
Cluster Node Communication
Starting a Proxy Server on a Host

How Many Cluster Nodes Should I Use for a Simulation?

It is important to scale the simulation size with the number of cluster nodes being used. The optimal scaling is determined from a ratio of the time to compute and the time that is taken to exchange data between cluster nodes.

If the simulation is distributed between too many cluster nodes, the time that is taken to perform the necessary data exchange dominates the total time that is taken to compute each simulation iteration. Consequently, no benefit is achieved from using more cluster nodes.

The time that it takes to exchange data between cluster nodes is determined by the size of the simulation, the number of cluster nodes that are used, and the speed at which the network hardware can exchange data.

CFD calculations require pulling large amounts of data through each core continuously. Since the typical calculations performed on the data are relatively simple, the key limitation on overall performance comes from the ability to access the data. Multiple cores have to share memory bandwidth, which may not be able to keep up with the data flow requirements.

Therefore, the optimal number of computational processes is dependent on the specification of the nodes and the interconnects used. It is recommended that when using 100/10BaseT Ethernet cards the minimum number of mesh cells per process is 100,000, whereas on higher-performance network hardware this limit can be decreased.

Defining a Machine File for Parallel Hosts

If you are not using a dedicated cluster with its own batch management system, you can define a group of networked machines on which STAR-CCM+ can launch a parallel server.

It is useful to specify the machines using an MPI file when multiple machines are being used. The format of the file is simply a list of machines on which you wish to place processes. The name of the machine can be concatenated with a colon and an integer to represent the number of processes you wish to place on the machine. An example of such a file could be:

```
node1
node2:3
node3:2
```

This file dictates that processes are allocated to machines `node1`, `node2`, and `node3`. 3 processes are allocated to `node2`, and 2 processes are allocated to `node3`.

If the machine file specifies more resources than are requested using the `-np` option the machine file is parsed allocating processes to the machines until the requested number have been allocated. In the above example, if you requested 3 processes using `-np 3` on the command line the processes would be placed on `node1` and `node2`.

If the machine file specifies fewer resources than are requested using the `-np` option the allocation of processes is continued from the start of the machine file once the end is encountered. In the above example, if you requested 8 processes the first would be allocated to `node1`, the second, third, and fourth allocated to `node2`, and the fifth and sixth allocated to `node3`. Then the seventh would be allocated to `node1`, and the eight allocated to `node2`.

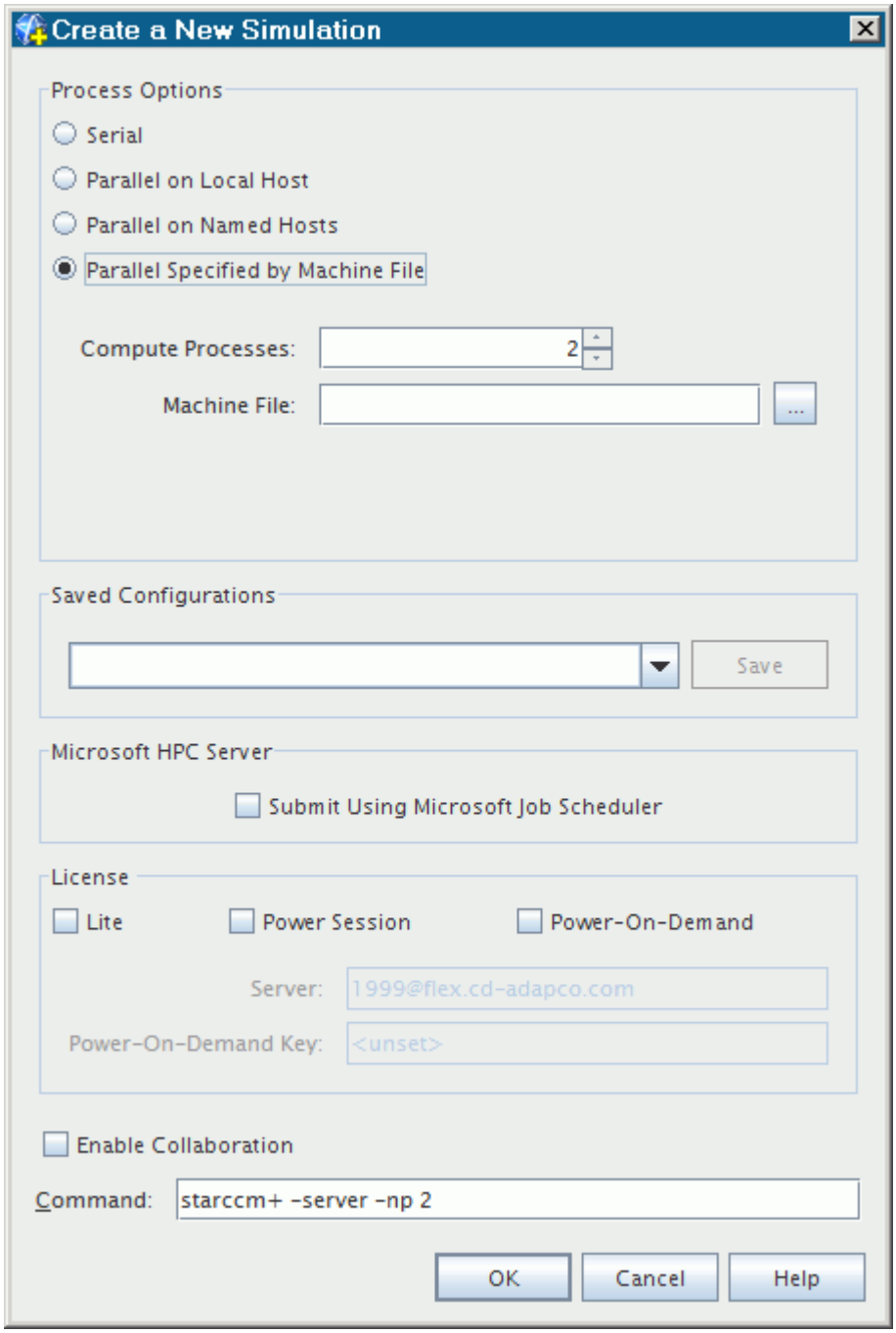
To use the machine file option from the command line, specify the file using the machine file option `-machinefile <file name>`. A typical command line looks like:

```
% starccm+ -machinefile mymachines -np 4 ~/airfoil.sim
```

The machine file can also be specified from the workspace. In the *Create a New Simulation* dialog or the *Load Simulation* dialog, click the **Parallel Specified by Machine File**radio button. Then do the following:

- Enter the number of required processes in the **Compute Processes** spin box.

- Either enter the name of the machine file in the **Machine File** text box or click the ellipsis button to open the file browser.



The **Command** text box displays the command line that is used for launching the server. You can edit this command line directly to override the command line parameters that the GUI choices generate.

Starting from the Command Line

Assume that you have a cluster of four machines that are called `node1`, `node2`, `node3`, and `node4`, and you want to run `airfoil.sim` in your home directory using one process per machine. If you are starting this job from another machine that shares a common view of the filesystem where your simulation file is located (for example, your home directory on your Linux machine is shared through NFS with the cluster machines), enter the command:

```
% starccm+ -on node1,node2,node3,node4 ~/airfoil.sim
```

This action starts the STAR-CCM+ client on your local machine, which in turn starts the STAR-CCM+ server on the cluster machines with one parallel process per machine. The named simulation file is read in to the server.

There is no mechanism for specifying a different user name in parallel mode, and you cannot start the master on a remote machine using the `-machinefile` option.

STAR-CCM+ assumes that the install path is the same on all the cluster machines as on the machine on which the server is started. If NFS is being used, the install path must be the same.

Starting a Cluster of Machines Manually

If the filesystem associated with the cluster is not visible from your local machine, if you are starting a parallel job from a Windows machine for example, you can start the remote server manually. To start the remote server manually, remotely log in to the cluster machine (through telnet or rlogin), and then connect to the server from the client. (To facilitate this process, read about how STAR-CCM+ works with specific network devices).

Telnet to the master by typing the command:

```
telnet node1
```

into the *Run* dialog (use **Start** > **Run...** to get to that dialog) or in a command shell window and login to that machine as you normally would.

From your telnet window, issue the command:

```
% starccm+ -server -on node1,node2,node3,node4 ~/airfoil.sim
```

The `-server` option instructs STAR-CCM+ to start the server rather than the client process. The `-on` option prescribes the hosts where processes are started. When starting a server, the `-on` argument must specify the local host. The workspace is not started.

Back on your Windows machine or workstation open the workspace of STAR-CCM+ and connect to the server.

Editing Hostnames of Machines

A command line option permits you to edit the hostnames of the machines that are prescribed to the server. This facility is primarily for the use with batch systems.

Some clusters are set up so that each cluster node (machine) has two network cards. The first network card can be a default ethernet card while the second one can be a higher-performance network card. In these circumstances, the hostnames that are associated with the second card are not the same as those allocated to the first card. Typically the hostnames are appended by a string, but it depends on how the machine is configured. The batch systems do not see second cards, so when the batch system allocates nodes on which jobs can run, they refer to the default cards.

However, when you run STAR-CCM+ you may want to take advantage of the higher-performance card. To do this, you would need to translate the hostname that is provided by the batch system to the hostname you actually want to use.

Suppose that you have a cluster of machines with names:

```
green-node1
green-node2
green-node3
...
green-nodeN
```

The batch system can allocate four nodes:

```
green-nodeN1
green-nodeN2
green-nodeN3
green-nodeN4
```

However, the application wants to use the higher performance network cards which could be named:

```
green-nodeN1-eth1
green-nodeN2-eth1
green-nodeN3-eth1
green-nodeN4-eth1
```

To use the higher performance network cards, enter the following in the command line:

```
-hostregexpr /<pattern>/<substitution>/
```

where the “regular expression” `pattern` is replaced by the regular expression `substitution`. Regular expressions are a general way of pattern matching strings.

Also, `/<pattern>/<substitution>/` is passed directly into the command:

```
echo $machinename | sed -s 's/<pattern>/<substitution>/g'
```

Examples of how you could use this technique are:

- Appending a string to the end of every hostname:

  ```
  -hostregexpr /\$/<string>/
  ```

  So to append `-eth1` you would enter:

  ```
  -hostregexpr /\$/-eth1/
  ```

The dollar sign represents the end of line (or end of hostname) and replaces it with the string. At the same time, it is necessary to escape the dollar sign to prevent the command-line shell from interpreting it. Hence, the syntax is `/\$/-eth1/` instead of `/$/-eth1/`.

- Adding a string to the start of every hostname:

```
-hostregexpr /^/eth1-/
```

This option gives entries such as `eth1-green-nodeN1`. (The ^ is the start of a line or hostname in a regular expression. There is no need to escape this character as it has no special meaning to the shell.)

- Substituting a common string with another:

```
-hostregexpr /green/green-eth1/
```

This option results in `green-eth1-nodeN1`.

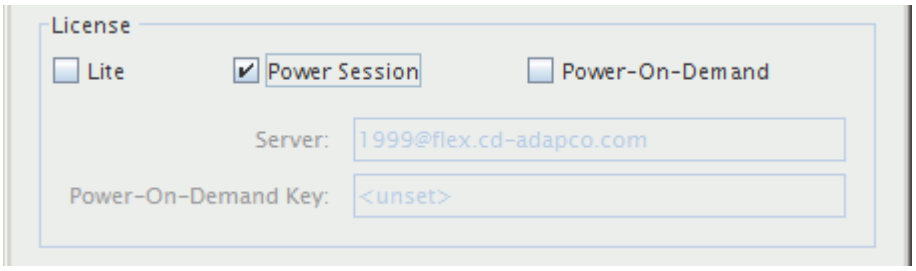
Using the Power Session License Feature

The Power Session license feature lets you run one job on an unlimited number of cores. A list of the different license features is provided in the STAR-CCM+ Licensing Guide section. When this type of license is used, other licenses such as `ccmpsuite` remain inactive.

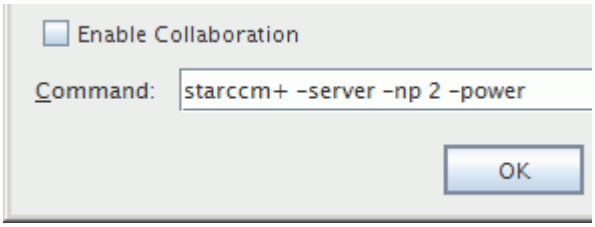
If a job is started using the Power Session, only one copy of the `ccmppower` license is checked out. Contact Siemens PLM Software to obtain this license.

Starting from the Workspace

You can activate the Power Session feature using a checkbox in the *Create a New Simulation* and *Load Simulation* dialogs.



When you tick this checkbox, the option `-power` appears in the command line of the dialog automatically.

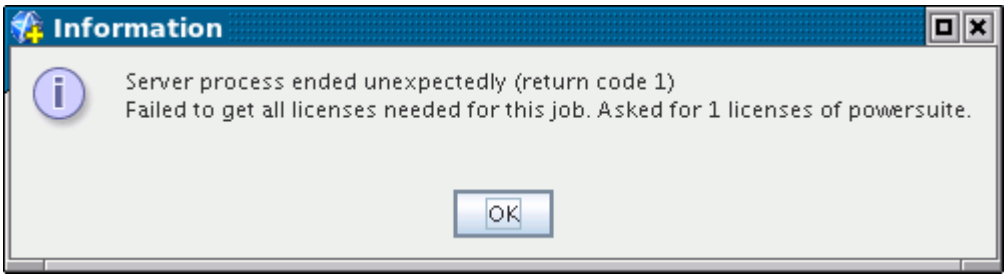


Starting from the Command Line

The command line includes the option `-power` as follows:

```
starccm+ -power -np 2
```

If the license is not available when you attempt to activate it, STAR-CCM+ generates a message like this one:



Using Parallel I/O

For systems that support it, simulation files can be saved and restored using parallel input/output (I/O).

With serial I/O (the default behavior), a single STAR-CCM+ server process performs all the data input and output on behalf of all server processes, scattering and gathering data as needed. This data flow creates a serial I/O bottleneck that can limit save and restore performance. With parallel I/O activated, multiple STAR-CCM+ server processes perform the I/O operations concurrently using the parallel file system. This option mitigates the serial I/O bottleneck, and can potentially reduce the time it takes to save and restore data.

To activate parallel I/O, you require a parallel file system. Using the feature on a locally mounted file system or network file system (NFS) usually provides no performance benefit, and can actually reduce performance. NFS is not a parallel file system.

The format of simulation (`.sim`) files that are saved with parallel I/O activated is the same as the format of simulation files that are saved using default serial I/O. A simulation file that is saved using parallel I/O can be read using serial I/O. Similarly, a simulation file that is saved using serial I/O can be read using parallel I/O.

Backward Compatibility

You can restore all old simulation files using parallel I/O. To obtain the full performance benefit of parallel I/O on restore, the simulation file must have been saved using STAR-CCM+ version 6.06 or higher. Although the simulation file format for serial and parallel I/O are the same, the format was revised for version 6.06 to improve its application to parallel I/O.

Limitations

Current limitations are:

- Linux only. Currently, parallel I/O is only available on Linux platforms.
- Simulation files only. Currently, the capability is only available for saving and restoring `.sim` files.

Contents:

What are the Requirements for Parallel I/O?
Activating Parallel I/O

What are the Requirements for Parallel I/O?

The main requirement for this feature is a parallel file system. A command line option is required to activate the feature with STAR-CCM+. Further requirements are listed in the following sections.

Parallel File System

A parallel file system is required to get benefits from this feature. Examples of parallel file systems include PanFS from Panasas, Lustre, and the General Parallel File System (GPFS) from IBM. Of parallel file systems, the current capabilities have been tested and tuned for PanFS. The feature should work on other parallel file systems that are supported by the I/O framework of the Message Passing Interface (MPI) standard.

Common Visible Path

A common visible file path on all compute nodes is required for this feature to work correctly. The volume on the parallel file system (where the simulation file is accessed) must:

1. be mounted on all compute nodes where the associated STAR-CCM+ server processes are being run; and,
2. have a common mount point. For example, if a volume is mounted as `/foo` on some nodes and `/fu` on other nodes, the simulation file cannot be saved as `/foo/bar.sim` with parallel I/O since `/foo` is not visible on some of the nodes.

STAR-CCM+ attempts to revert to serial I/O in cases where these requirements are not met, but this is not guaranteed to be successful.

Sufficient Hardware Resources

Even with a parallel file system, sufficient hardware resources must be available to obtain significant performance benefits. For example, with Panasas storage systems, multiple shelves (or blade racks) are required to obtain significant performance improvements relative to serial I/O (especially on a fast network).

Sufficiently large simulations

The benefits of parallel I/O can only be realized for sufficiently large simulation files (typically containing more than 1 million cells). This is because sufficiently large storage blocks are required to effectively stripe across the available file system servers.

Activating Parallel I/O

You can activate parallel I/O for simulation file saves and restores globally from the command line using the `-pio` option. For example:

```
$ starccm+ -pio -np 8 foo.sim
```

When this flag is included, the given simulation file is restored using parallel I/O, and all subsequent simulation files are saved using parallel I/O.

Troubleshooting

This section contains some troubleshooting tips.

Specifying Parallel Hosts Yields Invalid Entries

- Make sure that the specified host names are spelled and configured correctly.

Command-Line Option Fails

- Ensure that the command-line shown in the GUI conforms to the options described in Command-Line Reference. See Command-Line Reference.

A Windows Cluster Does Not Work

- Install the most recent release version of STAR-CCM+ available using the same installation path on all machines, and ensuring that the MPI component is chosen and set up correctly.
- Verify on each machine that you can start serial and local parallel servers.
- Start the multi-host parallel server, trying one host at a time. Starting remote servers is not supported under Windows, so when starting a multi-host parallel server the master process must be run on the local host.
- If it does not work, turn off the firewalls on the machines and try again. If you need help, start the server from the command line with the `-v` option to get debug output, as follows:

  ```
  "c:\Program Files\CD-adapco\STAR-CCM+ <version_number>\star\bin
  \starccm+.exe" -server -on host1,host2 -v
  ```

  where `host1` and `host2` are the two machines that you want to run on, and one of them is the current local machine.

Shared Memory Limits Are Too Low

Linux workstations are often configured with low limits on the amount of allowable shared memory. This restricts how much memory can be pinned by the libraries that MPI uses. These libraries can print warning messages even when only using a single host and they usually indicate the limits are set too low—even if the library isn't being used. Set workstations to have high or preferably unlimited limits.

The error messages are:

- From IBM Platform MPI:

  ```
  libibverbs: Warning: RLIMIT_MEMLOCK is 32768 bytes
  ```

- From Open MPI:

  ```
  The OpenIB BTL failed to initialize while trying to allocate some
  locked memory.
  This typically can indicate that the memlock limits are set too
  low. For most HPC installations, the memlock limits should be set
  to "unlimited".
  ```

```
The failure occurred here:

Host: compute_node.example.com
OMPI source: btl_opebib.c:114
Function: ibv_create_cq()
Device: Out of memory
Memlock limit: 32767
```

The Open MPI Frequently Asked Questions describe how the limits are changed (see answer to How can a system administrator (or user) change locked memory limits?)

In the latest release of IBM Platform MPI, the mpid, which is part of the MPI environment, also requests shared memory. When it fails, it returns a message similar to the one shown below:

```
Cannot create shared memory segment of <size> bytes.
```

The Platform MPI Release Notes describe how to increase the maximum allowable shared memory size. See page 32 of the Linux Platform MPI 8.03.00.00 Release Notes.

Infiniband Spawns Error Messages for High Process Counts

When STAR-CCM+ runs on non-optimal Infiniband networks using the default Platform MPI, the MPI settings can result in robustness issues when using high process counts. The issue can manifest as error messages from the Infiniband drivers. Examples include:

```
ibv_poll_cq(): bad status 12
```

and:

```
error message: transport retry exceeded error
```

You can improve the robustness of STAR-CCM+ by using the following command line options:

```
-mppflags "-e MPI_IBV_QPPARAMS=21,7,31,7,128"
```

The additional robustness incurs extra memory overhead, especially with higher process counts.

Balancing

Care must be taken to balance a parallel configuration to achieve maximum performance. Generally, allocate each process to a single core where no other jobs are being executed. Any additional load on the core to which a process is allocated results in an imbalance in the parallel configuration and paging, resulting to a degradation in performance. In general, the slowest core in the machine limits the parallel simulation: so if one core has a greater load than the remaining cores, all cores are limited by the excessively loaded core.

In the case of a distributed parallel simulation, performance is limited by the slowest machine in the cluster. If one node has a greater load than the remaining nodes, all nodes are all limited by the excessively loaded node.

Getting Parallel Help through the Command Line

You can get information on the parallel options through the command line.

To get detailed help about parallel in the console, enter the following in the command line:

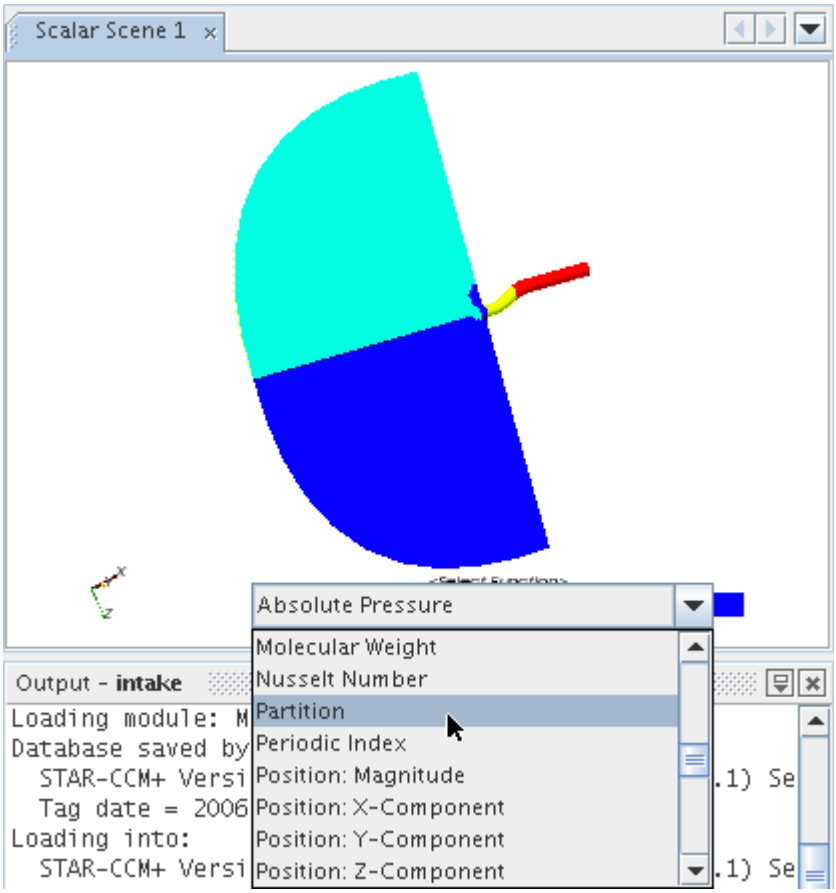
```
starccm+ -phelp
```

This command can also be used with specific parallel options by adding the syntax `:<parallel option>`. For example, to get specific help on the `-np` option, enter:

```
starccm+ -phelp:-np
```

Viewing the Partitioning

The distribution of the simulation data can be visualized in the same way as any other scalar quantity: adding a scalar scene or adding a scalar displayer to an existing scene, and then changing the scalar quantity selection (the field function) to `Partition`. The cells are colored according to the partition that they have been allocated to by STAR-CCM+.



Using Batch Management Systems

Batch systems manage the allocation of computational jobs to a single large computer or a cluster of smaller computers.

You typically submit a job to a batch system requesting computational resources through the attributes of that particular batch system. The batch system then manages the job, allocating it specific machines when they become available based on the requested resources. In this way, resources are delegated to computational jobs in a fair and efficient manner that is based on rules that are configured by your system administrator.

STAR-CCM+ can be run in batch mode under the control of a batch system. If you wish to run STAR-CCM+ in parallel under the control of a batch system, certain resource attributes must be obtained from the batch system and used to construct the correct command-line arguments for STAR-CCM+. A common way of doing this is to examine attributes of the batch system in the batch system's submission script and pass these attributes to the STAR-CCM+ command line. For example, if you requested two machines from the batch system when you submitted the job, the batch system should provide a way of finding out which machines have been allocated. These machine names can then be used to construct a machine file for the parallel execution of STAR-CCM+.

Automatic transfer from the batch system to STAR-CCM+ of a limited number of attributes is supported for the following batch systems:

Contents:

Working in Unix-Based Batch Systems
Using PBS
Using LSF
Using Sun Grid Engine
Using Load Leveler
Using Windows Server 2012

Working in Unix-Based Batch Systems

STAR-CCM+ helps you interact with certain batch systems by simplifying the submission process. You can either write a submission script manually or permit STAR-CCM+ to submit itself automatically to the system.

Using Manual Submission

STAR-CCM+ simplifies manual submission by providing the `-batchsystem` argument, which automatically extracts batch system properties and translates them into STAR-CCM+ command-line arguments. This method requires you to construct a small submission script manually, and is the most popular submission technique today. This technique is most easily explained through an example of a simple submission script, which is in the section on using the PBS Batch System.

Due to the multitude of batch systems and possible configurations for them, the details of the particular submission script that you may wish to construct are beyond the scope of this section. Refer to the

documentation of your own specific batch system and, if necessary, consult with your system administrator to identify the attributes to forward to the STAR-CCM+ command line.

If you want to use a different parallel configuration, perhaps one more complex, write a custom script. The standard transfer of information from the batch system to STAR-CCM+ will not be adequate. This method involves bypassing the `-batchsystem` option, extracting the batch system properties manually, and forwarding them to your STAR-CCM+ script.

An example of a custom script is included in the section on using the PBS Batch System.

Using Automatic Submission

STAR-CCM+ can submit itself automatically to a selection of popular batch systems, so you do not need to write a submission script. In this setup, each process is placed on its own batch system node by default.

Note: You cannot use automatic submission if you have a custom parallel configuration.

The general syntax is as follows:

```
starccm+ <options> -batchsystem <batchsystem>:"-submit <batchoptions
submitoptions>" <sim file>
```

where the batch system is: `pbs`, `lsf`, `sge`, or `ll`.

Among the submit options are those that can be forwarded to the `submit_starccm+` script, and those that can be forwarded to the batch system submission command. Options that are forwarded to the `submit_starccm+` script include:

- `-logfile <file>` Specifies the log file to receive the output.
- `-submitfile <file>` Specifies the filename of the submission script generated.
- `-writescript` Writes the submission script and exits (that is, it does not actually submit the job).

  This option is useful when you want to generate a template submission script, rather than launch an actual submission.
- `-tmpdir <dir>` Specifies the temporary directory.
- `-v` Increases verbosity.
- `-cleanup` Specifies running the `kill_procfile` script after completing the job.

An example of automatic submission is included in the section on using the PBS Batch System.

Using PBS

When you specify PBS manually on the command line, STAR-CCM+ will automatically distribute a process to each PBS node allocated to the PBS job.

An alternative is automatic submission.

To specify PBS manually, create a simple script file (for example, `pbsbatch`) that contains the full command line that you would use if you were not using a batch management system, but include the `-batchsystem` option:

```
#!/bin/sh
cd $PBS_O_WORKDIR
/install_path/star/bin/starccm+ -batchsystem pbs -batch \
 <macro file>
```

Next, submit the job to the batch management system, specifying the number of resource nodes to use:

```
% qsub pbsbatch -q <queue name> -l nodes=<# processes>
```

To see how the job is progressing, you can use the following command:

```
% qstat -a
```

The `-batchsystem` argument only supports the case in which one process is distributed to each of the PBS nodes that are allocated for the job. If processes are to be distributed to the PBS nodes in a different manner, you cannot use the `-batchsystem` argument and must write a custom submission script. An example of such a script is provided below.

```
#!/bin/sh
# This script requests 4 OpenPBS nodes each with
# two cores. It then allocates two starccm+
# processes to each OpenPBS node.
# Request 4 nodes each with two cores
#PBS -l nodes=4:ppn=2
##################################################
# Additional PBS directives could be placed
# here. For example the queue could be specified
# or the maximum cpu or wall time specified.
##################################################
# specify 8 processes
NCPU=8
echo -------------------------------------------------------
echo 'This job is allocated on '${NCPU}' cpu(s)'
echo 'Job is running on node(s): '
cat $PBS_NODEFILE
echo -------------------------------------------------------
cd $PBS_O_WORKDIR
starccm+ -batch <macro file> -np $NCPU -machinefile $PBS_NODEFILE
exit
```

Automatic Submission to PBS

For PBS, an example automatic submission command is displayed below:

```
starccm+ -rsh ssh -mpidriver platform:"-UDAPL -prot" -batch
profile.java -batchsystem pbs:"-submit -logfile logfile -cleanup -
```

```
lwalltime=1:00:00 -lselect=14:ncpus=8:mpiprocs=8" /
path_to_simulation_file/setup.sim
```

An example of the submission script that is automatically generated by STAR-CCM+ is provided below.

```
#!/bin/sh
# This file has been automatically generated by the submit_starccm+
script.
# It is used to submit a STAR-CCM+ command to the pbs batch management
# system and should be removed automatically once the job has finished.
# The existence of this script implies the job is still being executed.
# run exit_hook on exit
trap 'eval $exit_hook' 0
trap 'eval $exit_hook; echo "`basename $0` Interrupted"; exit_hook=;
exit' 2
trap 'eval $exit_hook; echo "`basename $0` Terminated"; exit_hook=;
exit' 15
trap 'eval $exit_hook; echo "`basename $0` Aborted"; exit_hook=; exit'
6
exit_hook="rm -f /home/cdadapco/noel/data/f1/47million/pids.19762;
$exit_hook"
exit_hook="kill_procfile /home/cdadapco/noel/data/f1/47million/pids.
19762; $exit_hook"
### PBS ###
### PBS Options added automatically
### Name of job
#PBS -N setup
## Set the PBS shell to sh
#PBS -S /bin/sh
## Request wall clock time of 24 hours
#PBS -lwalltime=24:00:00
### Join stdout and stderr
#PBS -j eo
### Set output file
#PBS -o /home/cdadapco/noel/data/f1/47million/setup.19762
### Print some useful PBS output
echo "-------------------------------------------------------" >>
logfile.19762
echo "Job is running on node" >> logfile.19762
cat $PBS_NODEFILE >> logfile.19762
echo "-------------------------------------------------------" >>
logfile.19762
echo "PBS: qsub is running on $PBS_O_HOST" >> logfile.19762
echo "PBS: originating queue is $PBS_O_QUEUE" >> logfile.19762
echo "PBS: executing queue is $PBS_QUEUE" >> logfile.19762
echo "PBS: working directory is $PBS_O_WORKDIR" >> logfile.19762
echo "PBS: execution mode is $PBS_ENVIRONMENT" >> logfile.19762
echo "PBS: job identifier is $PBS_JOBID" >> logfile.19762
```

```
echo "PBS: job name is $PBS_JOBNAME" >> logfile.19762
echo "PBS: node file is $PBS_NODEFILE" >> logfile.19762
echo "PBS: current home directory is $PBS_O_HOME" >> logfile.19762
echo "PBS: PATH = $PBS_O_PATH" >> logfile.19762
echo "------------------------------------------------------" >>
logfile.19762
### Set working directory
WORK_DIR="/home/cdadapco/noel/data/f1/47million"
cd $WORK_DIR
### Simulation File
SIM_FILE="setup.sim"
### Macro
MACRO="profile.java"
### Run Command
starccm+ -rsh ssh -mpidriver platform -mppflags "-UDAPL -prot" -batch
$MACRO -batchsystem pbs $SIM_FILE -batchsystemoptions " -nosubmit" -
procfile /home/cdadapco/noel/data/f1/47million/pids.19762 >> logfile.
19762
### Sleep for 2 minutes. This helps ensure the STAR-CCM+ server
### has shut down before any prologue scripts kill remaining processes
sleep 120
```

The submission command that automatically gets executed looks like this:

```
qsub -lwalltime=1:00:00 -lselect=14:ncpus=8:mpiprocs=8 /tmp/pbs-
cdadapco20273/pbs-submit.20273
```

Using LSF

By specifying LSF on the command line, STAR-CCM+ will automatically distribute a process to each LSF node allocated to the job.

An alternative is automatic submission.

Submit the job to the batch management system, specifying the number of resource nodes to use (\ indicates a line continuation):

```
% bsub -q <LSF queue> -n <number of processes> \
/install_path/star/bin/starccm+ -batchsystem lsf \
-batch <macro file> /path_to_simulation_file/simfile.sim
```

To see how the job is progressing:

```
% bjobs
```

Using Sun Grid Engine

By specifying Sun Grid Engine on the command line, STAR-CCM+ will automatically distribute a process to each Sun Grid Engine node allocated to the job.

An alternative is automatic submission.

Submit the job to the batch management system, specifying the number of resource nodes to use:

```
% qsub -pe <SGE Parallel Environment> <number of SGE nodes> <Sun Grid
Engine Script>
```

where the minimal Sun Grid Engine script would be

```
#!/bin/sh
/install_path/star/bin/starccm+ -batchsystem sge -batch <macro file> /
path_to_simulation_file/simfile.sim
```

To see how the job is progressing:

```
% qstat
```

Using Load Leveler

By specifying Load Leveler on the command line, STAR-CCM+ will automatically distribute a process to each Load Leveler node allocated to the job.

An alternative is automatic submission.

Submit the job to the batch management system, specifying the number of resource nodes to use:

```
% llsubmit <Load Leveler Script>
```

where the minimal Load Leveler script would be:

```
#!/bin/sh
# @ job_type = parallel
# @ node = <number of processes>
/install_path/star/bin/starccm+ -batchsystem ll -batch <macro file> /
path_to_simulation_file/simfile.sim
```

To see how the job is progressing:

```
% llq
```

Using Windows Server 2012

STAR-CCM+ batch jobs can be submitted to the Windows Server 2012 job scheduler interactively or using the -msjob option. Parallel jobs that are submitted to the job scheduler use MS-MPI by default. Before you can

submit a batch job to the Windows job scheduler, register your credentials. See Registering Your Credentials on Windows Server 2012.

To submit the job:

```
[INSTALL PATH]\star\bin\starccm+.exe -msjob [STAR-CCM+ options] -batch
[PATH TO SIMULATION FILE]\simfile.sim
```

The working directory, standard output, and standard error are set to the directory that the simulation file is in.

If the job is successfully submitted, the job scheduler prints the job ID:

```
Job has been submitted. ID: 230.
```

Cluster Node Communication

This section explains the following:

Contents:

What is a Network Device?
What Different Network Devices Does STAR-CCM+ Support?
On Which Platforms Are Different Network Devices Used?

What is a Network Device?

A cluster of machines is connected using a network which provides the means for communication between the cluster nodes. The hardware that is used for the network determines the maximum speed that data can be exchanged between cluster nodes.

When running STAR-CCM+ in parallel, data is exchanged between the different cluster nodes several times during each simulation iteration, and so the performance of the network affects the performance of STAR-CCM+. Typically, clusters are not accessible from outside the head node. The nodes often have both ethernet and another flavor of network card. The network configuration prohibits direct access to the cluster nodes.

Support for each different network option is provided by using different MPI libraries. The MPI library provides the software to accomplish data transfer between different cluster nodes. Different MPI libraries interface with, and take optimal advantage of, the underlying network hardware, and the dependent communication protocol for inter-node communication. All MPI's use shared memory for intra-node communication.

By default STAR-CCM+ uses the included IBM Platform MPI or SGI MPI if it is available on the hosts. SGI machines are often configured as huge shared memory systems even though they are distributed memory systems. SGI MPI is optimal for SGI shared memory systems.

What Different Network Devices Does STAR-CCM+ Support?

STAR-CCM+ can take advantage of different network devices that are installed between cluster nodes. A list of supported devices includes:

- Shared Memory
- Ethernet
- Myrinet GM and Myrinet MX
- Infiniband:
  - OpenFabrics Enterprise Distribution using Verbs API Infiniband Interface (VAPI)
  - SilverStorm using User-Level Direct Access Programming Infiniband Interface (uDAPL)
  - QLogic Infiniband Interface
- SGI Shared Memory

Different network hardware can be selected with the command line by specifying the respective MPI driver and options:

```
-mpidriver <MPI driver><:driver options>
```

The `<driver options>` portion of the command line permits you to incorporate options that you want to add, depending on your use of the MPI application and the machines on which you are running STAR-CCM+. However, these options must only be added by a support person or someone with expert knowledge of the MPI application.

Using Shared Memory

STAR-CCM+ always uses shared memory between processes running on the same host, regardless of the network hardware that is chosen to pass information between processes on different hosts.

Using Ethernet

If IBM Platform MPI cannot detect a faster network card, it reverts to using Ethernet. Ethernet can be explicitly chosen using:

```
-mpidriver platform:-tcp
```

In some cluster configurations, each node can be equipped with more than one network device. For example, a simple 100/10BaseT Ethernet device might be accompanied by a higher-speed Gigabit Ethernet device. Since both devices use the same underlying communication protocol (TCP/IP), the same MPI library can be used, but when two network devices are present using the same communication protocol, it is often necessary to tell STAR-CCM+ explicitly which one to use. This is normally achieved by specifying the correct hostnames that are associated with each network device.

Using Myrinet

Myrinet network devices provide support by two software interfaces: the GM Interface and the more recent MX (Myrinet Express) Interface. STAR-CCM+ with Myrinet can be faster than Ethernet, especially when large numbers of processes are involved.

You need a Myrinet card (in addition to the Ethernet network card) in each machine you want to use, as well as the appropriate drivers.

Using the Myrinet GM Interface

If the device supports the GM Interface, you can choose it by using:

```
-mpidriver myrinet
```

By default when you choose Myrinet GM using the above command, or let IBM Platform MPI choose Myrinet GM automatically using:

```
-mpidriver platform
```

IBM Platform MPI will try and locate the Myrinet GM lib `libgm.so`. It will first check whether the file is in the default system path, then it will look in the default installation path `/opt/gm/lib/`.

If the file is not located in the default installation path or in the system path, you may specify the location by including the environment variable in your `.cshrc` file:

```
setenv MPI_ICLIB_GM <installation path>/libgm.so
```

Using the Myrinet MX Interface

If the device supports the MX Interface, it may be chosen using:

```
-mpidriver myrinet-mx
```

By default when you choose Myrinet MX using the above command, or let IBM Platform MPI choose Myrinet MX automatically using:

```
-mpidriver platform
```

IBM Platform MPI will try and locate the Myrinet MX lib `libmyriexpress.so`. It will first check whether the file is in the default system path, then it will look in the default installation path `/opt/mx/lib/`.

If the file is not located in the default installation path or in the system path, you may specify the location by including the environment variable in your `.cshrc` file:

```
setenv MPI_ICLIB_MX <installation path>/libmyriexpress.so
```

The Myrinet MX driver has not been tested.

Using Infiniband

Infiniband is a specification of a high-performance network device. Different Infiniband network solutions exist, and supported devices include:

- Open Fabrics Enterprise Distribution (OFED) Infiniband

  ```
  -mpidriver infiniband:-ibv
  ```

- SilverStorm via the User Level Direct Access Programming Interface (uDAPL)

  `-mpidriver infiniband:-udapl`

- QLogic Infiniband Interface

  `-mpidriver infiniband:-psm`

However, rather than specify the Infiniband interface, you can simply use the generic command to activate it:

```
-mpidriver infiniband
```

STAR-CCM+ translates this action as `-mpidriver platform` and thus lets IBM Platform MPI decide which Infiniband interface is most suitable. If no Infiniband interface exists, an alternative transport layer is chosen by IBM Platform MPI, but no warning is given to you.

Using SGI Shared Memory

SGI machines are often configured such that a distributed memory system appears as a shared memory system. Such SGI machines are often equipped with an MPI library that can take advantage of this specific memory architecture. STAR-CCM+ can take advantage of this memory architecture on single-host, multi-core

machines where the SGI MPI library has been installed. STAR-CCM+ probes for SGI MPI and uses it automatically, provided you have not used the `-machinefile` or `-on` options. You do not need to enter a separate command.

SGI MPI must be installed separately on your machine; it is not distributed with STAR-CCM+.

If SGI MPI is not installed under the default directory (`/usr`), the environment variable `SGI_MPI_HOME` must be set such that `libmpi.so` and `libmpi++.so` exist in `$SGI_MPI_HOME/lib/`. Also, the executable `mpirun` must exist in `$SGI_MPI_HOME/bin/`. The `starccm+` script can show how this works. As only single-host machines are supported, prescribe the number of processes using the `-np` rather than the `-on` or `-machinefile` options.

SGI MPI uses a concept of cpusets to control which processes are run on which CPUs. To use SGI MPI, either specify a particular cpuset, otherwise STAR-CCM+ runs on the default cpuset. For more information about cpusets, refer to the SGI Technical Documentation.

Processes are placed on the CPUs within the cpuset in a round-robin fashion unless the `-oncpu` option is invoked to specify which CPUs relative to the cpuset are used. The SGI command `dplace` must be available to use `-oncpu`. The `starccm+` script checks for this option.

On some SGI systems, you can prescribe the CPUs where the STAR-CCM+ processes are run using the following command line option:

```
-oncpu <CPU1,CPU2,...,CPUN>
```

The series that is shown in brackets can either be a comma-separated list or a range that is given by a hyphen. The exact format is the same as the SGI `dplace -c` command. This option is helpful when you wish to run on one shared-memory machine while restricting which CPUs of that machine STAR-CCM+ uses. For example:

```
starccm+ -np 10 -oncpu 0-12
```

To achieve good performance, make sure that the cpuset has enough CPUs allocated to it to permit each process to be placed on their own CPUs.

Polling is off by default. This driver is able to support polling to enhance performance by adding the option:

```
-mpidriver sgi:-polling
```

To deactivate it again, use the following line command:

```
-mpidriver sgi
```

Note: Do not over-subscribe a node with processes as it can result in performance degradation.

When starting SGI MPI, you may encounter the following warning:

```
"MPI: This executable was not linked with MPT version 1.6.1 or later.
Memory mapping features such as single copy transfers, MPI one-sided
communication and shmem functions are not available unless you relink
your executable with the current MPT library."
```

It is a known problem with SGI and can be ignored.

On Which Platforms Are Different Network Devices Used?

The table below represents the support for the different interconnects on different release platforms.

Platform	Interconnect
Linux	ethernet (IBM Platform MPI)
	ethernet (SGI MPI)
	Myrinet-GM (IBM Platform MPI)*
	Myrinet-GM (SGI MPI)
	Myrinet-MX (IBM Platform MPI)**
	Myrinet-MX (SGI MPI)
	Infiniband (IBM Platform MPI)*
	Infiniband (SGI MPI)
Windows	ethernet (MS-MPI)
	ethernet (IBM Platform MPI)
	Infiniband (MS-MPI)

* Tested but not quality-assured support.

** No testing done at this time.

Expert MPI Options

IBM Platform MPI is the default MPI used on Linux and Windows. Typically, you should not need to work with IBM Platform MPI expert options.

However, if you wish to do so, refer to the IBM Platform MPI User's Guide (`pcmpi.XX.XX.XX.ug.pdf`) and Release Notes (`pcmpi.XX.XX.XX.rn.pdf`) that are located in the STAR-CCM+ installation directory:

- Linux: `[INSTALL DIR]/mpi/platform/X.X.X.X/[ARCHITECTURE]/doc/`
- Windows: `[INSTALL_DIR]\mpi\platform\X.X.X.X\[ARCHITECTURE]\help\doc\`

Running Distributed Parallel Simulations on Multiple Windows Machines

On Windows platforms you can run distributed parallel simulations using cores from multiple machines, in conjunction with IBM Platform MPI. To do this, use Windows PowerShell to launch STAR-CCM+ from within a shared directory, which is accessible to all machines that you want to use. You must have read/write access to the shared directory.

To run distributed parallel simulations on multiple machines:

1. Make sure that your credentials are registered on all machines that you want to use. See Registering Your Credentials for Distributed Simulations on Windows.
2. Click <**Start**> and type `powershell` in the search field.
3. Right-click on **Windows PowerShell** from the list of search results, and select **Run as Administrator**. You must have administrator privileges.

The *Administrator: Windows PowerShell* command prompt appears.

4. Change to the shared working directory:

```
> cd \\<MACHINE_NAME>\<SHARED_DIR>
```

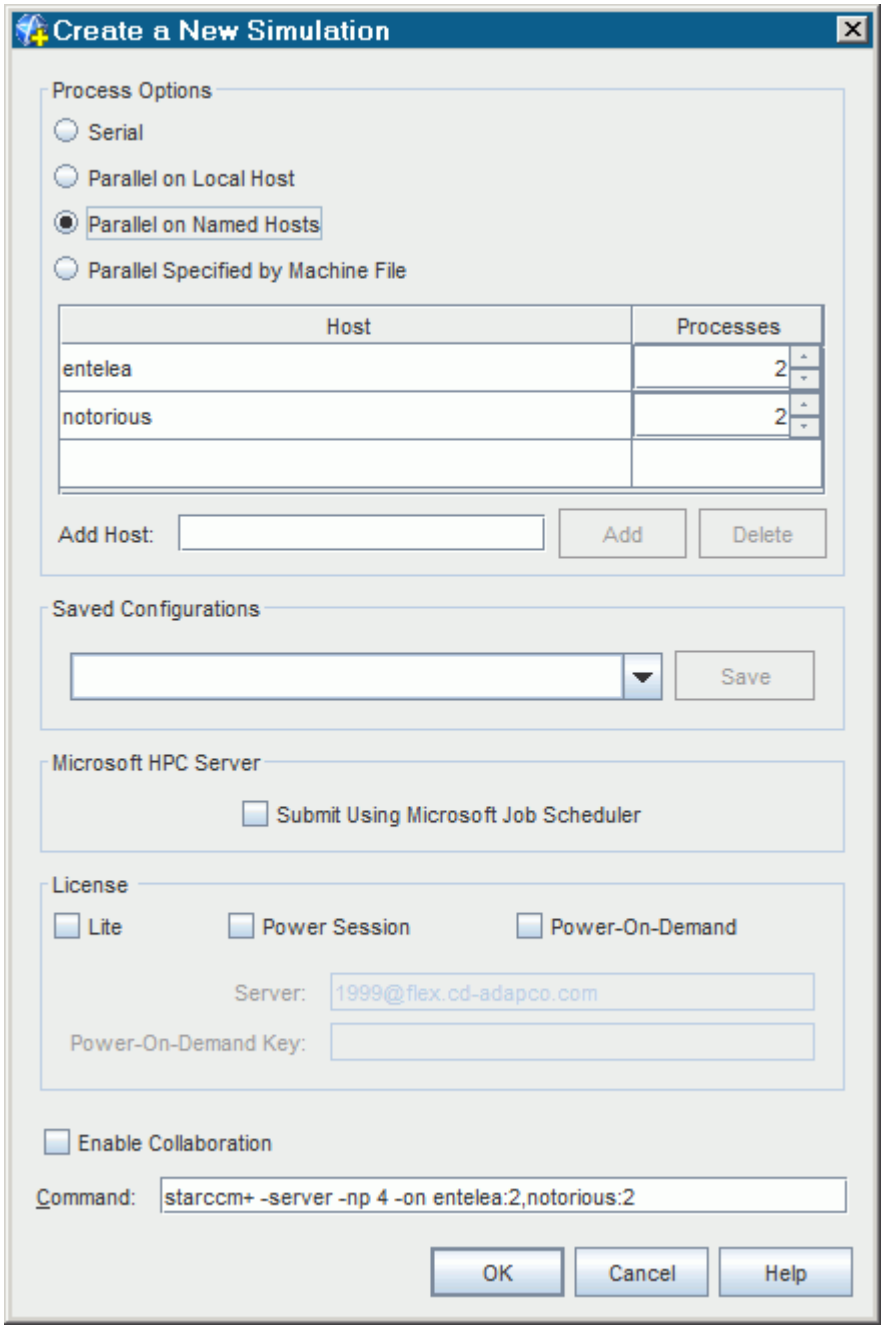
5. Launch STAR-CCM+:
   - If STAR-CCM+ has the same installation path on each machine, enter:

     ```
     > starccm+
     ```

   - If STAR-CCM+ has a different installation path on each machine, use the UNC path to the STAR-CCM+ executable:

     ```
     > \\<MACHINE_NAME>\<STAR-CCM+_INSTALL_DIR>\starccm+
     ```

   The STAR-CCM+ GUI appears.
6. Create or load a simulation file as you normally would. See Creating a New Simulation Interactively and Loading an Existing Simulation Interactively.
7. In the *Create a New Simulation* or *Load Simulation* dialog, select **Parallel on Named Hosts** from the *Process Options*. See Using Parallel on Multiple Host Machines.
8. Enter the host names and the corresponding number of cores to use from each host.



9. If you are opening an existing file, select the file that you want to load.
10. Click **OK**.

STAR-CCM+ creates or opens the file, and uses the allocated cores from each machine. The relevant information is displayed in the *Output* window.

Command-Line Reference

The `starccm+` command can start either the STAR-CCM+ client:

```
starccm+ [options] [simulation]
```

or server process:

```
starccm+ -server [options] [simulation]
```

Without any arguments, the STAR-CCM+ GUI client is started. No server process is started, as that is only required when a simulation is created or loaded.

If a `simulation` file is named without any options, both the STAR-CCM+ client and a server are started, and the named file is loaded into the server.

If the `-server` argument is specified, only the STAR-CCM+ server process is started. Any server option can also be specified on the command-line to start the client; the client passes those options on when it starts the server process.

General Options

Argument	Description
`-info`	Prints information about the simulation file: the version of STAR-CCM+ with which that file was saved (in this example 12.06.###), number of partitions, file size, and number of iterations in the file: `STAR-CCM+ 12.06.### (linux-x86_64-2.5/gnu4.6) Serial 680.34 KB  Iter 66`
`-ini <file>`	Activates an `.ini` file that contains general and client options.
`-loc`	For the client, this option turns off broadcasting of queries to locate servers on the network; it also prevents the server from responding to any locator queries. The broadcast from the client is a UDP multicast, along with TCP pings, to known hosts. This multicast search is started by default when you launch the STAR-CCM+ client without a simulation file. In Windows this option can cause network access warnings from your personal firewall.
`-new`	Launches the STAR-CCM+ workspace and starts a new simulation without accessing a simulation file.

Argument	Description

-printpids	Prints process IDs. An example of the output format is as follows:
	`Host ID 0 Rank 0 on marlin32 with process ID 7270`
	`Host ID 0 Rank 1 on marlin32 with process ID 7422`
	`Host ID 0 Rank 2 on marlin32 with process ID 7591`

-v, -vv, -vvv	Activates Verbose mode, prints environment changes and sub-commands.
-version	Prints the STAR-CCM+ version information (in this example 12.06.###): `12.06.###`

License Options

Argument	Description
-lite	Requests a Lite Session license as opposed to a full license. Only a reduced functionality set is available.
-power	Starts STAR-CCM+ with the Power Session license feature.
-doepower	Uses a Power Session license for the optimization session license.
-nosuite	Prevents main licenses from being used to enable cores for parallel computation. Only parallel licenses are used to enable cores for parallel computation.
-powerpre	Uses 10 DOEtoken licenses to launch a STAR-CCM+ pre- and post-processing session.
-tokensonly	Specifies to only use DOEtoken licenses for optimization studies. With this option, you can also use DOEtoken licenses for regular parallel runs in STAR-CCM+.
-notokens	Specifies to not use DOEtoken licenses for optimization studies.
-doe-prefer-hpcdomains	Changes the order in which the DOEtoken license feature is checked out. Using this option checks out the hpcdomains license feature before the DOEtoken license feature for optimization studies.
-podkey <value>	Specifies a Power on Demand license key.
-doeuuid <value>	Specifies a DOE UUID (universal unique identifier).
-licpath <path:...>	Specifies a license path that overrides the default license path. When this option is used, STAR-CCM+ uses the path to define a new CDLMD_LICENSE_FILE environment variable for the spawned server session.
-reserve <lic1[,lic2,...]>	Allows you to specify add-on licenses that can be automatically checked out.
-noreserve	Prevents add-on licenses from being checked out unless a model operation requires them.

Argument	Description
`-noretry`	Prevents add-on licenses from being retried when a significant operation is carried out.
`-norelease`	Prevents add-on licenses from being released after importing geometry.

Client Options

Argument	Description
`-batch [command]`	Starts STAR-CCM+ in batch mode (without a GUI). The following additional commands are available: • `mesh` — runs the meshing pipeline • `run` — runs the solver until the stopping criteria are satisfied • `step` — steps the solution • `<macro>.java` — runs the specified Java macro file If no command is specified, the default action is to run the solver. In each case, the simulation is auto-saved through the auto save feature (unless that property is turned off). You can specify multiple commands within the `-batch` argument. In the following example, three macros are run in order before running the mesh pipeline and the solver: `-batch macro1.java,macro2.java,macro3.java,mesh,run`
`-batch-report`	Generates a summary report of a batch simulation. The report appears in `.html` format. This report includes details about the computer system where the simulation was run, as well as details about the simulation itself.
`-classpath <classpath>`	Overrides the *User macro classpath* setting from the *Options* dialog. This option only affects the macro compilation in the current session of STAR-CCM+.
`-clientldlibpath <path>:<path>:<path>`	This option permits you to add directory paths explicitly to the STAR-CCM+ client library search path (`LD_LIBRARY_PATH`) for the client runtime environment.
`-config <id>`	Accesses a saved configuration of server options. (These options are also listed in the Saved Configurations list of the dialog that loads a simulation or creates a new one.)
`-save-config <id>`	Saves a configuration of server options and adds it to the Saved Configurations list of the dialog that loads a simulation or creates a new one.
`-fontsize <size>`	Adjusts the size of the font that is used in the STAR-CCM+ client workspace, such as dialogs and the object tree, for that session. Add the desired font size with this command. For example, a command of `starccm+ -fontsize 16` sets the font size to 16 points.

Argument	Description
-host <host> -port <port>	Specifies the host and port of a running STAR-CCM+ server process to which the client connects. If the -host and -port options are used, a simulation file cannot be specified.

-laf <look-and-feel class name>	Changes the look and feel of the STAR-CCM+ client. This options accepts any values that are currently supported by NetBeans on the version of JDK that is used by STAR-CCM+ on your operating system.
	For example, the default look and feel on Linux can be changed using:
	`-laf Nimbus`
	or on Windows:
	`-laf Metal`
	If the look and feel class is not recognized or is not available, a warning message is shown and STAR-CCM+ reverts to the default look and feel.

-locale <locale>	Overrides the default locale of your operating system. You can run STAR-CCM+ in the following languages: • English US: en:US • Japanese: ja • Chinese (simplified): zh • Russian: ru:RU
-macro <file>	Runs the macro file upon client startup. This option is similar to -batch in that a macro file is read, but in this case a GUI is opened. This option requires specification of a simulation file. Since it is impossible to record a macro that includes the opening of a simulation file, there is never a conflict.
-nbuserdir <dir>	Lets you specify a location, other than $HOME, in which the .star-*-*-* directory is written. Enter this command after client startup. This option changes the location of key STAR-CCM+ information, including the user settings directory and client log messages. Consider using this option when you do not have write access to your $HOME directory, or when you want to share modules.
-noexit	Prevents the server process from shutting down after batch file execution has completed. By default, the client and server processes both shut down after the macro completes.
-on <host>	Specifies the hostmachine on which to start the server process. By default, when a new simulation is created or an existing simulation file is opened, the STAR-CCM+ client starts a server process on the local machine. If the -on option is specified however, the client attempts to start the server process on the named host machine using the appropriate remote shell.

Argument	Description
	Additionally, this option can be used directly as a parallel option to specify the hosts that the parallel server uses.
`-reset`	Restores the default client layout (including window positions) and settings (options and mounted directories).

`-time`	Prints server-elapsed and CPU times after a batch run. Specifically, it prints a summary of computational resource use of the server on close-down.
	In serial:
	`Saving: /home/ncr/sims/suprBump@00077.sim`
	`Server Elapsed Time :  5.362  (s)`
	`Server CPU Time  :  0.830  (s)`
	`Closing connection to server and exiting`
	In parallel:
	`Auto save: /home/ncr/sims/suprBump@00077.sim`
	`Saving: /home/ncr/sims/suprBump@00077.sim`
	`|Global Rank|Server Elapsed Time (s)|Server CPU Time (s)|MPI Overhead (s)|MPI Overhead %Elapsed Time (s)|`
	`|  0|  7.120|  1.250|  0.461|  6.476|`
	`|  1|  7.120|  5.150|  2.371|  33.302|`
	`|  2|  7.120|  5.120|  2.425|  34.056|`
	`|  3|  7.120|  5.160|  2.451|  34.421|`
	`Maximum worker communication time on rank  : 3`
	`Minimum worker communication time on rank  : 0`
	`Average worker communication time  : 1.927`
	`Standard deviation worker communication times : 0.847`
	`Closing connection to server and exiting`

`-userdir <name>`	On Linux, adds a user settings directory to your `home` directory with `name` optionally added to the folder name.
`-nbuserdir <name>`	On Linux, specifies the NetBeans directory.
`-clientcore`	On Linux, reserves one core for running the client in a batch system. Use this flag if you perform a lot of post-processing while running in parallel.

Argument	Description
-clientldpreload <file:...>	On Linux, sets the client LD_PRELOAD environment variable.

Server Options

Argument	Description
-arch <architecture>	This option allows you to override the automatic determination of the architecture of the machine. The specified architecture must exist within the installation and must be compatible with the machine architecture. If these conditions are not met, STAR-CCM+ exits, or runtime errors occur. For example, this argument can work when a 32-bit Linux installation is executed on a 64-bit Opteron-based architecture: `-arch linux-x86-2.2.5`
-batchsystem <system>	This option can be used to aid the startup of parallel STAR-CCM+ under the control of a supported Batch Management System (for usage see Using Batch Management Systems). Currently the supported systems are: • PBS (<system>=pbs) • LSF (<system>=lsf) • Sun Grid Engine (<system>=sge) • Load Leveler (<system>=ll)
-collab	In the startup of a server, allows others to connect to it. (By default only the user who starts a STAR-CCM+ server is allowed to connect to it.)
-cout	Messages are directed to the console rather than to the STAR-CCM+ client.
-jobmanager <URL>	Specifies the URL of the Resource Manager that controls the resources for the STAR-CCM+ server. The Resource Manager automatically adds this option to the server.
-ldlibpath <path>:<path>:<path>	This option permits you to add directory paths explicitly to the STAR-CCM+ server library search path (LD_LIBRARY_PATH) for the server runtime environment. The -ldlibpath option is required when STAR-CCM+ relies upon third-party software that is installed in custom locations. In addition, consider using it for user code in the following instances: • Linking against a math library like IMSL or NAG. (Getting access to the functions in them is one reason to write user code.) • Most compilers that can build a .so file dynamically link it against their own runtime libraries. Add paths to these libraries, unless they happen to be the system libraries (as they are with GNU compilers on Linux).

Argument	Description
-load	Loads the specified simulation file as soon as the server starts. Without this option, the simulation database is loaded on demand when the client connects to the server.
-msjob[:<opt>]	Submits a cluster job to the Windows Server job scheduler. This functionality is available only on Windows Server 2012 machines.
-noconnect	In the startup of a server, specifies a session that is exclusive to the STAR-CCM+ console session that launched it. No other users can connect to that server.
-port <port>	Specifies server port for STAR-CCM+ to open.
-portrange <range>	Specifies the number of ports the server can attempt to bind to.
-server	Starts the STAR-CCM+ server with an automatically generated port number.
-rsh <option> -svrrsh <option>	The -rsh <option> option can be used to specify: • the remote shell used to start remote servers from the client • the remote shell used in the server to launch parallel processes, co-simulations, and so on. The -svrrsh <option> specifies the server's remote shell only. When starting a server using -server, the -rsh or -svrrsh options default to ssh. In a new client session the remote shell settings default to those set by the environment options, **Generic Remote Process Launch Command** and the **Server Remote Process Launch Command**.
-taskid <pid>	Specifies the task ID of the Resource Manager that controls the resources for the server. The Resource Manager automatically adds this option to the server.

Visualization

Argument	Description
-hardwarebatch	Uses hardware acceleration for rendering graphics scenes when running in batch. This option can provide faster performance when post-processing large numbers of scenes. Hardware drivers must be installed properly and be in use on the machine where the client process of STAR-CCM+ is running. For Linux/Unix machines, run an X server on the machine where the client process of STAR-CCM+ is running. This X server could be your interactive X session, or another X session that you have permission to run.

-mesa	Uses the alternative mesa graphics libraries in the following situations:
	• There is a problem with the graphics driver
	• You use software rendering for:
	  ◦ Running in batch
	  ◦ Displaying a client remotely with an X-server

Parallel Options

Argument	Description
-machinefile <file>	MPI machine resource file name. Only used with parallel operation. The processes are allocated to the machines in the resource list starting with the first entry. If the number of requested processes (defined by -np) exceeds the number of machines that are listed, the remaining processes continue to be allocated starting with the first entry in the resource list. Cannot be used with -on.
-mpi <driver> -mpidriver <driver>	Specifies the MPI driver to use <ms, platform>. If omitted, STAR-CCM+ uses a default driver. In addition, this command can help if you do not know the specific inter-connect hardware available on a machine.
-mpiflags <options>	MPI driver expert options.
-mpitest [:<options>]	Performs MPI tests and exits.
-np <np>	Number of processes to use. Only used for parallel operation. If the -np option is omitted and -machinefile or -on is specified, the number of processes is assumed to be the number that the resource list prescribes. If neither -on or -machinefile is specified with this option, then -on localhost:<np> is assumed.
-np <np[,npcosim]>	Number of processors to use for co-simulation, where np is the number of processes for STAR-CCM+ and npcosim is the number of processes for the other tool.
-on <host[:np][,host[:np]]>	List of hosts to run parallel processes on. If :np is specified, then np processes are put on that machine. There must be no spaces in the list of hosts. The processes are allocated to the machines in the resource list starting with the first entry adding np processes before moving on to the next machine in the list. If the number of requested processes (defined by -np) exceeds the number of machines that are listed, the remaining processes continue to be allocated starting with the first entry in the resource list. Cannot be used with -machinefile.
-pio	Activates parallel input/output on RAID-5 file systems. When this flag is included, the given simulation file is restored using parallel I/O, and all subsequent simulation files are saved using parallel I/O.
-cpubind <policy>	Specifies what binding policy <off, bandwidth, latency> to use when binding CPUs. You can request a report to display the binding layout by entering: -cpubind v

Contents:

Working with .ini Files

See Also:

Working with .ini Files
Using the Power Session License Feature
Using Power Tokens for Pre- and Post-Processing Sessions
Controlling the Check Out Order of DOE Parallel Licensing
Power-on-Demand Licensing
STAR-CCM+ Add-on Modules
Using Parallel I/O
CPU Binding

Working with .ini Files

You can use an `.ini` file as a shortcut for general and client command line options when starting local STAR-CCM+ simulations.

To use this feature, add the `-ini` option to your line command:

```
% starccm+ -ini myinifile.ini <simfile>
```

The file contains simple listed commands, with any line beginning with `#` considered a comment. For instance, if you usually prefer to use `rsh` rather than `ssh` on your system, the `.ini` file could look like the following:

```
# set default rsh command
-rsh rsh
```

A more complex `.ini` file could be constructed as follows:

```
# set default rsh command
-rsh rsh
# set batchsystem and always allocate the controller process to a cpu/
core shared with a worker process
-batchsystem pbs:
# MPI options
#   - HP MPI 2.3.0
#   - SilverStorm Infiniband Driver using UDAPL layer
#   - Bind processes to cores
-mpidriver hp2:"-prot -UDAPL -cpu_bind=MAP_CPU:0,1,2,3,4,5,6,7"
```

Arguments that are passed on the command line take precedence over the ones that are listed in the `.ini` file. For example, suppose that you issue the following command line:

```
% starccm+ -ini myinifile.ini <simfile> -rsh ssh
```

The `-rsh` argument in the command line takes precedence over the one in the `.ini` file. Currently there is no way to merge these two arguments.

Your system administrator can create a global `.ini` file that applies these options to all servers.

Contents:

Creating a Global .ini File

See Also:

Creating a Global .ini File

Creating a Global .ini File

System administrators can create a global `.ini` file that contains command line options that affect all users. This ability is particularly useful if there are system-wide options that all users should be using.

For a list of command line options, refer to Command Line Reference.

Create the global `.ini` file as described in Working with .ini Files and place it in the following directory of your STAR-CCM+ installation:

```
[INSTALL_DIR]/star/config/ini
```

Name the `.ini` file according to the architecture of the system where it is installed, for example:

```
[INSTALL_DIR]/star/config/ini/starccm+-<arch>.ini
```

Where `<arch>` is the architecture of your system.

Naming the Linux Global .ini File

On Linux, you can find the architecture by using `maparch`, which is located in `[INSTALL_DIR]/star/bin/maparch`. An example command to run `maparch` is:

```
% ./star/bin/maparch
linux-x86_64-2.5
```

The `.ini` file is then named:

```
[INSTALL_DIR]/star/config/ini/starccm+-linux-x86_64-2.5.ini
```

Naming the Windows Global .ini File

On Windows, the `.ini` file is named:

```
[INSTALL_DIR]\star\config\ini\starccm+-windows.ini
```

See Also:

Command-Line Reference

Activating Parallel I/O

Pre-Processing

STAR-CCM+ solves physics problems on a computational mesh. To provide this mesh, you first define geometry using internal or external tools, and then create a mesh pipeline whose operations generate the mesh in repeatable steps.

Contents:

Modeling Geometry
Defining the Regions Layout
Meshing

Modeling Geometry

This part of the User Guide describes how to use 3D-CAD, the parametric solid modeler available within STAR-CCM+. It also describes the features and tools available for further preparation of the geometry model and its surfaces.

For many simulations in STAR-CCM+, the first step involves preparing the geometry of the case under investigation. This geometry could come from a model that was prepared previously in a CAD package, or could be created directly using the 3D-CAD module within STAR-CCM+.

3D-CAD

This fully integrated CAD tool allows you to build geometry from scratch within the STAR-CCM+ environment, as well as import external CAD models and modify them before running a simulation. 3D-CAD is tailored to suit the needs of a CFD analysis and allows you to change the geometry quickly and easily, before rerunning the simulation.

The chapters in this section covering 3D-CAD are as follows:

- Creating geometry with 3D-CAD
- Importing geometry into 3D-CAD
- Exporting geometry from 3D-CAD
- Using 3D-CAD models in a simulation

Preparation of the Geometry Model

The concepts that are associated with manipulating and preparing geometry have been distinguished from the ones that relate to regions and boundaries through the introduction of geometry parts. Geometry parts are contained within the Geometry Parts node of the object tree. They are fully described in the section Using Geometry Parts.

The other parts of the documentation cover the following:

- An introduction to using STAR-CCM+
- How to generate or import meshes and work with them
- Modeling physics -- the concepts, theory, and implementation details of the models
- Using tools in STAR-CCM+ to facilitate your work
- Detailed information on interacting with CAD/CAE products in a STAR-CCM+ simulation
- Examples of applying the models of STAR-CCM+ in sophisticated cases
- Preparation and procedures for running a STAR-CCM+ simulation
- Analyzing results through visualization and plotting

- A wizard guide that shows how STAR-CCM+ features are combined in customized interfaces for special applications
- A tutorial guide that shows you how to use STAR-CCM+ for various applications in a step-by-step format
- A guide to using STAR-View+ , a viewer for three-dimensional STAR-CCM+ scenes

Contents:

Preparing the Geometry
Using 3D-CAD
Importing Geometry
Understanding Geometry Parts
Manipulating Parts
Part Contacts
Repairing CAD Geometry
The Surface Repair Tool
Displaying the Geometry

Preparing the Geometry

In general, a STAR-CCM+ analysis requires one or more geometry parts from which you define the spatial construction of the regions. STAR-CCM+ allows you to create geometry parts directly using the built-in 3D-CAD modeler, or import geometry from a wide range of source formats.

Complete the layout planning stage before starting this workflow.

1. Provide initial geometry using one of the following methods:
   - Create geometry using the integrated 3D-CAD module.
   - Create a simple shape part directly (block, cone, cylinder, or sphere).
   - Import an existing CAD model in its native or neutral format.
   - Import an existing surface mesh for which the underlying CAD is no longer available.

   When importing geometry, on the import options dialog, retain **Create Part Contacts from Coincident Entities** as activated unless you have a specific reason not to.
2. If you created geometry using 3D-CAD, convert the 3D-CAD model to equivalent geometry parts.

   See Creating geometry parts from 3D-CAD.
3. Check the validity of parts and repair them when necessary:
   a) For imported CAD parts, run the CAD repair tool.
   b) For non-CAD parts, or if the CAD repair tool could not fix all issues for imported CAD parts, run the surface repair tool. See Repairing the surface.
4. If necessary, position parts so that they align correctly. See Transforming parts.
5. If the geometry parts represent the solid components only, and you require the complementary fluid volume, extract the fluid volume. See Extracting the fluid volume.
6. In some cases, adjacent part surfaces can overlap without having any common edges around the overlapping area. To create common edges, which is essential for a quality mesh, imprint the overlapping parts. See Imprinting adjacent parts.
7. Expand the **Geometry** > **Contacts** node and review the part contact objects.

   A part contact references two parts that are adjacent to each other. When both geometry parts are transferred to the regions domain, STAR-CCM+ uses the part contact information in creating interfaces (for boundary-based interfaces). You can also use the same information in creating parts-based interfaces.

When you prepare your geometry for a STAR-CCM+ simulation, consider the physics models that you are going to be using in the simulation.

For example, if you are modeling fluid flow through a valve:

- assign an inlet boundary to the surface where fluid enters
- assign an outlet or pressure boundary to the surface where fluid leaves.

To assign boundaries to surfaces, there must be separately selectable surfaces for both boundaries. To split surfaces at the geometry stage, use the following tools:

- 3D-CAD models: set face names
- Geometry parts: split surfaces by patch, angle, part curves, or non-contiguous

Using 3D-CAD

The first part of this chapter is an introduction to 3D-CAD modeling that covers the fundamental concepts behind 3D-CAD. If you are new to solid modeling or 3D-CAD, read this section first.

Subsequent chapters describe how to create geometry using the tools available in 3D-CAD. These tools are:

- Working with sketches
- Converting sketches to bodies
- Working with bodies
- Working with fillets and chamfers
- Working with design parameters
- Working with fluid volume extraction

To create geometry that is based on an imported CAD file, refer to the following section:

- Importing geometry into 3D-CAD

Contents:

Introducing 3D-CAD Modeling
Sketches
Constraining a Sketch
Dimensioning Sketches
Creating Bodies From Sketches
Working with Bodies
Working with Faces, Edges, and Vertices
Filtering Selections
Working with Patterns
Working with Fillets and Chamfers
Working with Design Parameters
Working with Volume Extraction
Removing Unwanted Features
Viewing the Properties of Geometric Objects
Working with Custom Name Attributes
Working with Reference Geometry in 3D-CAD
Viewing Dependency Information of Features and Bodies
Importing Geometry into 3D-CAD
Exporting Geometry from 3D-CAD
Working with 3D-CAD Models
The Turbo Slicing Tool

Introducing 3D-CAD Modeling

3D-CAD is a feature-based parametric modeler within STAR-CCM+ that allows you to build geometries from scratch. The geometries that you create with 3D-CAD are stored as 3D-CAD models, which you can then convert to geometry parts for integration with the meshing and simulation process.

The first step in building a geometry is to create a 3D-CAD model. This action activates 3D-CAD and you can start to create your geometry. Models are created by carrying out a series of operations such as sketching, extruding, and cutting. Each operation is a feature of the model, and the chronological overview of these features is displayed in the feature tree.

In general, the process of building a model in 3D-CAD begins with the creation of a sketch. You can convert this sketch to a body by extruding or revolving. You can add or remove material from this body by creating more sketches and performing extrusions, revolutions, sweeps, lofts, and cuts. Body operations allow you to modify the geometry further. In addition to solid modeling, 3D-CAD also supports surface modeling.

You can modify geometry that you created in 3D-CAD at any time by reactivating 3D-CAD and modifying the model. When the model is updated the changes are applied throughout the simulation, requiring you only to remesh and rerun the case.

A major feature of 3D-CAD is design parameters, which allow you to modify the 3D-CAD model from outside of 3D-CAD. These design parameters allow you to solve for a particular geometry, change the size of one or

more components, and rerun the case quickly. If a solution has already been computed, it is mapped across to the modified geometry, decreasing the overall run time.

You can check the validity of geometries that you create as well as geometries that you import in 3D-CAD.

Contents:

What Is a 3D-CAD Model
Visualizing a 3D-CAD Model
Checking Geometry Validity
3D-CAD Feature Previews
Working with the 3D-CAD View Scene
Working with Features and the Feature Tree
Working with the 3D-CAD Toolbar
Working with the Visibility Toolbar in 3D-CAD
Surface Modeling in 3D-CAD

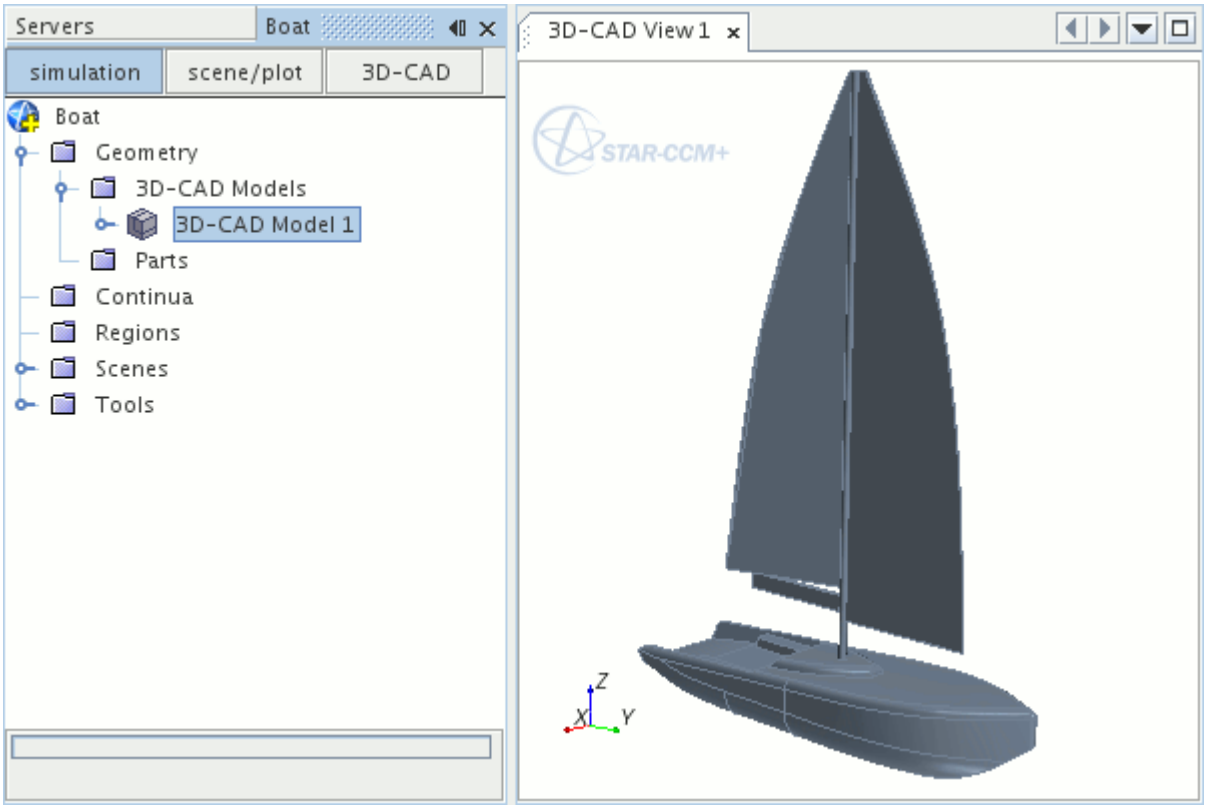
See Also:

What Is a 3D-CAD Model
Creating Geometry Parts from 3D-CAD
Working with Features and the Feature Tree
Sketches
Surface Modeling in 3D-CAD
What Is a Design Parameter?
Understanding Geometry Parts

What Is a 3D-CAD Model

A 3D-CAD model is a solid model that is created in STAR-CCM+ using the feature-based parametric solid modeler, 3D-CAD.

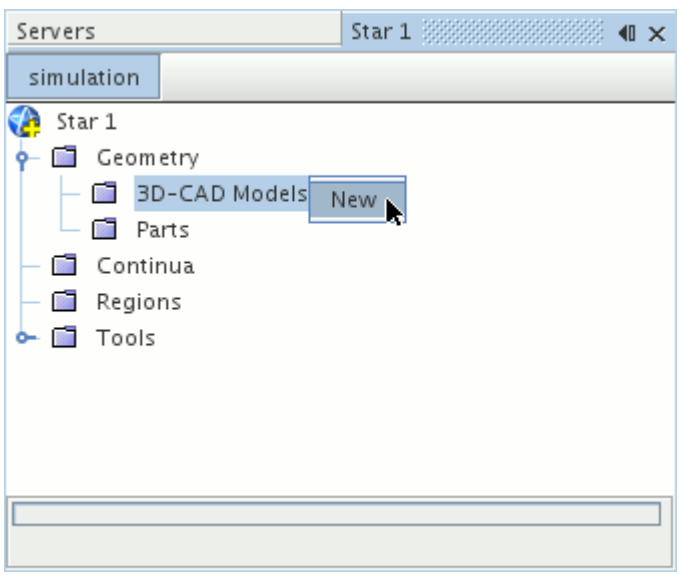
You create models within the familiar environment of STAR-CCM+. These models are stored under the **Geometry** > **3D-CAD Models** node in the simulation tree.



Creating a 3D-CAD Model

To create a 3D-CAD model:

- Right-click on the **Geometry** > **3D-CAD Models** node and select **New**.



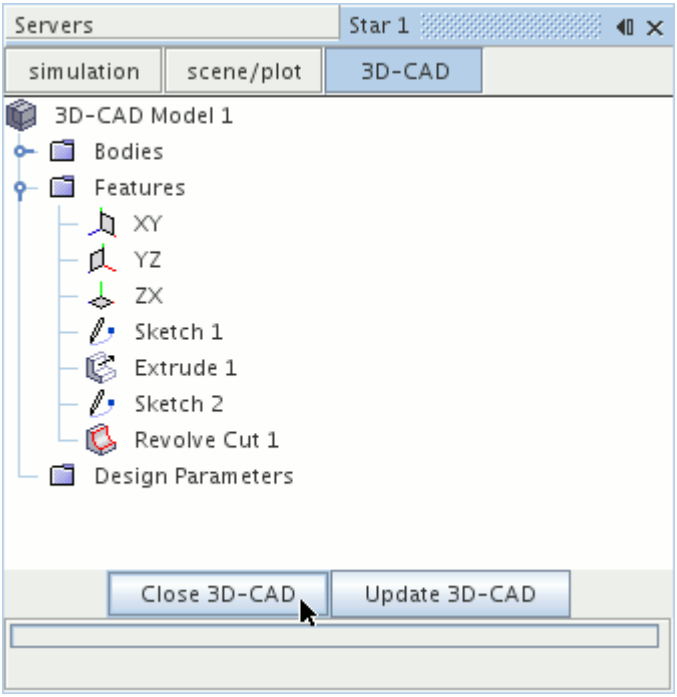
This action creates a 3D-CAD model node under the **Geometry** > **3D-CAD Models** node and activates 3D-CAD. When 3D-CAD is open, the 3D-CAD model tree is shown in the explorer pane and a *3D-CAD View*

scene is displayed in the *Graphics* window. You can begin adding features to the model by creating a sketch or sketch plane, or by importing geometry from an external CAD file. To finish working on the model, close 3D-CAD.

Exiting 3D-CAD

To exit 3D-CAD and return to the main simulation tree:

- Click the **Close 3D-CAD** button.

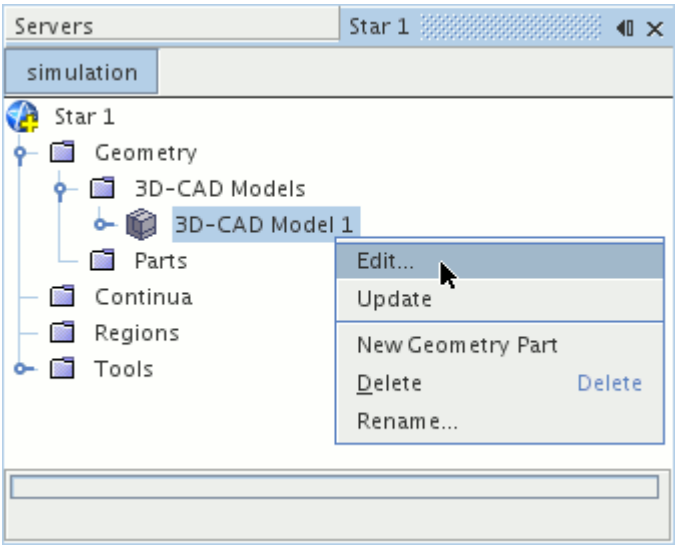


The *3D-CAD View* scene automatically closes, and view settings in the scene are discarded. You can save these settings before exiting 3D-CAD by storing the current view. You can then restore the view in another scene later on. To reactivate 3D-CAD, you can edit an existing 3D-CAD model or create a model.

Editing an Existing 3D-CAD Model

You can edit existing 3D-CAD models and resume from where you left off. To edit a 3D-CAD model:

- Right-clicking on the model node under the **Geometry** > **3D-CAD Models** manager node and select **Edit...**



3D-CAD launches and opens the *3D-CAD View* scene. You can resume working on your 3D-CAD model. When you have finished editing the model, you can close 3D-CAD and return to the simulation tree. If you already used the model to create geometry parts, update these parts for the changes to take effect.

Note: Further steps are required if you used the geometry to generate a mesh or run a simulation. See Modifying a 3D-CAD Model.

See Also:

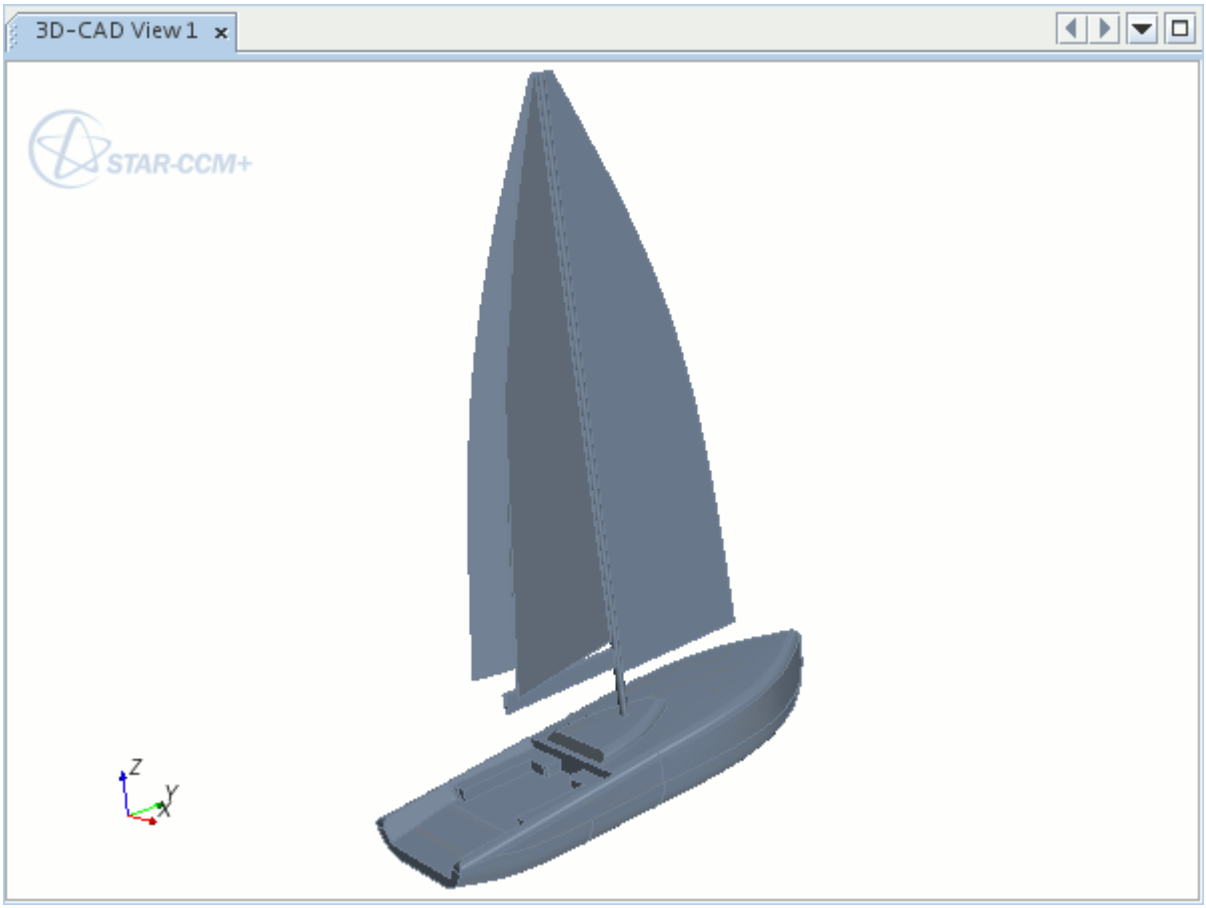
Sketches

Sketch Planes

Creating Geometry Parts from 3D-CAD

Visualizing a 3D-CAD Model

3D-CAD models are visualized in the *3D-CAD View* scene. This scene opens automatically when you activate 3D-CAD by creating or editing a model.



In this scene, you can rotate and zoom in to view the geometry model, as well as hide and show select faces and bodies. To display all hidden restore the view, right-click a blank space and select **Restore All** followed by the appropriate item that you want to restore. You can control the how geometry is displayed in the *3D-CAD View* scene as well as what is displayed using the Visibility toolbar.

When you close 3D-CAD, the *3D-CAD View* scene also closes. You can restore the scene by reactivating 3D-CAD as described above. Closing 3D-CAD also discards the view settings—when a new *3D-CAD View* scene is opened, the default view is shown. You can save the view settings before exiting 3D-CAD by storing the current view. You can restore these view settings in a scene later on.

To visualize a 3D-CAD model outside of 3D-CAD:

1. Convert the model to geometry parts. See Creating Geometry Parts from 3D-CAD.
2. Display the parts in a geometry scene. To create a geometry scene:
   - Click the (**Create/Open Scenes**) button in the **Vis** toolbar and select **Geometry**.
   - Alternatively, right-click on the **Scenes** node and select **New Scene** > **Geometry**.

See Also:

What Is a 3D-CAD Model

Hiding Bodies

Hiding Faces

Working with the Visibility Toolbar in 3D-CAD

Checking Geometry Validity

STAR-CCM+ can check the validity of geometries within the 3D-CAD module.

Checking for valid geometries is important before setting up a simulation—particularly when using imported geometries. Invalid geometries can cause problems during the meshing process. You can check the validity of an entire model, specified bodies, and specified faces in 3D-CAD. You can check the validity of sheet bodies as well as solid bodies.

Use one of the following actions to run a validity check:

- In the feature tree, right-click a body that you want to check and select **Check Validity**.
- In the feature tree, right-click the model node and select **Check Validity**.
- In the 3D-CAD View scene, right-click on a face and select **Check Validity** from the *Body* group box.
- In the 3D-CAD View scene, right-click on a face and select **Check Validity** from the *Face* group box.

  A message with the results of the validity check is printed in the *Output* window. For example, if a body is not valid, the following message is displayed:

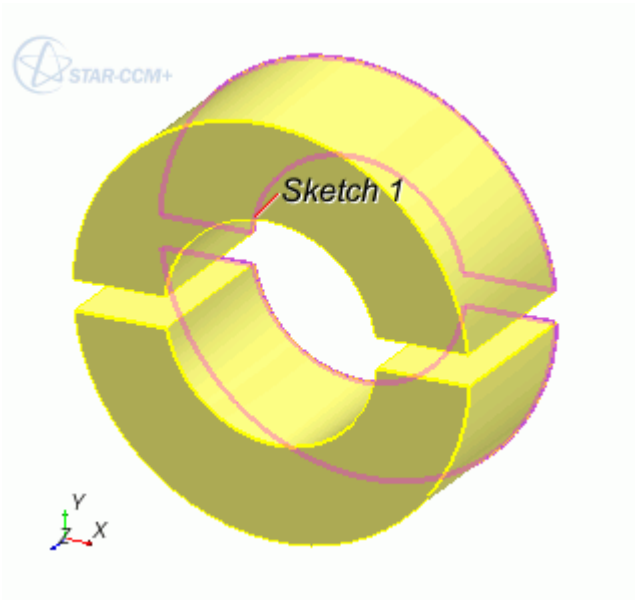
```
Body Body 2 is not valid.
```

Checking Validity Upon Import

You can automatically check the validity of a geometry when you import it into 3D-CAD. See Importing Geometry into 3D-CAD.

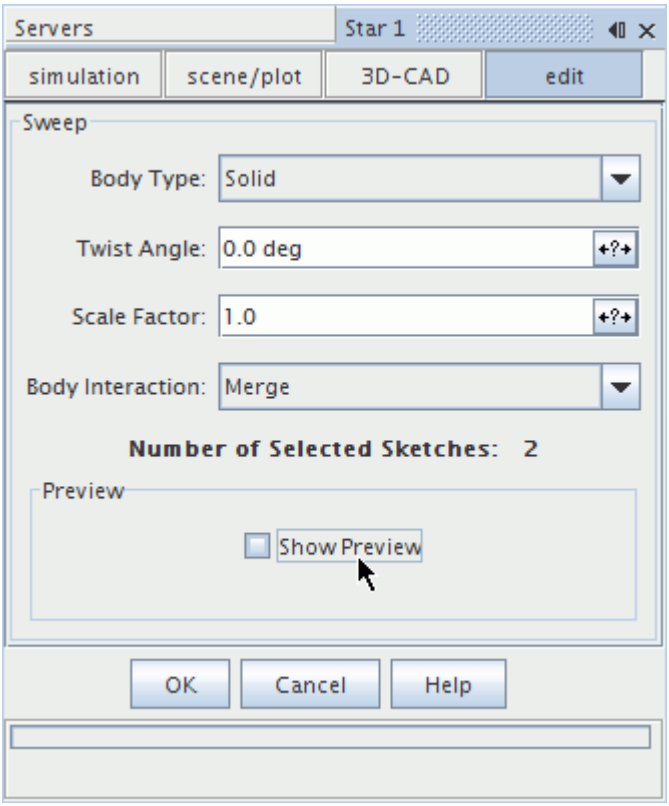
3D-CAD Feature Previews

For most 3D-CAD features, STAR-CCM+ displays a preview of the expected result in the *3D-CAD View* scene before you complete the action to create the feature. If you have a complex model, and find that the preview function takes too long, you can deactivate the function in the feature panel.



To deactivate the preview display:

1. Deactivate the *Show Preview* option in the relevant feature panel.

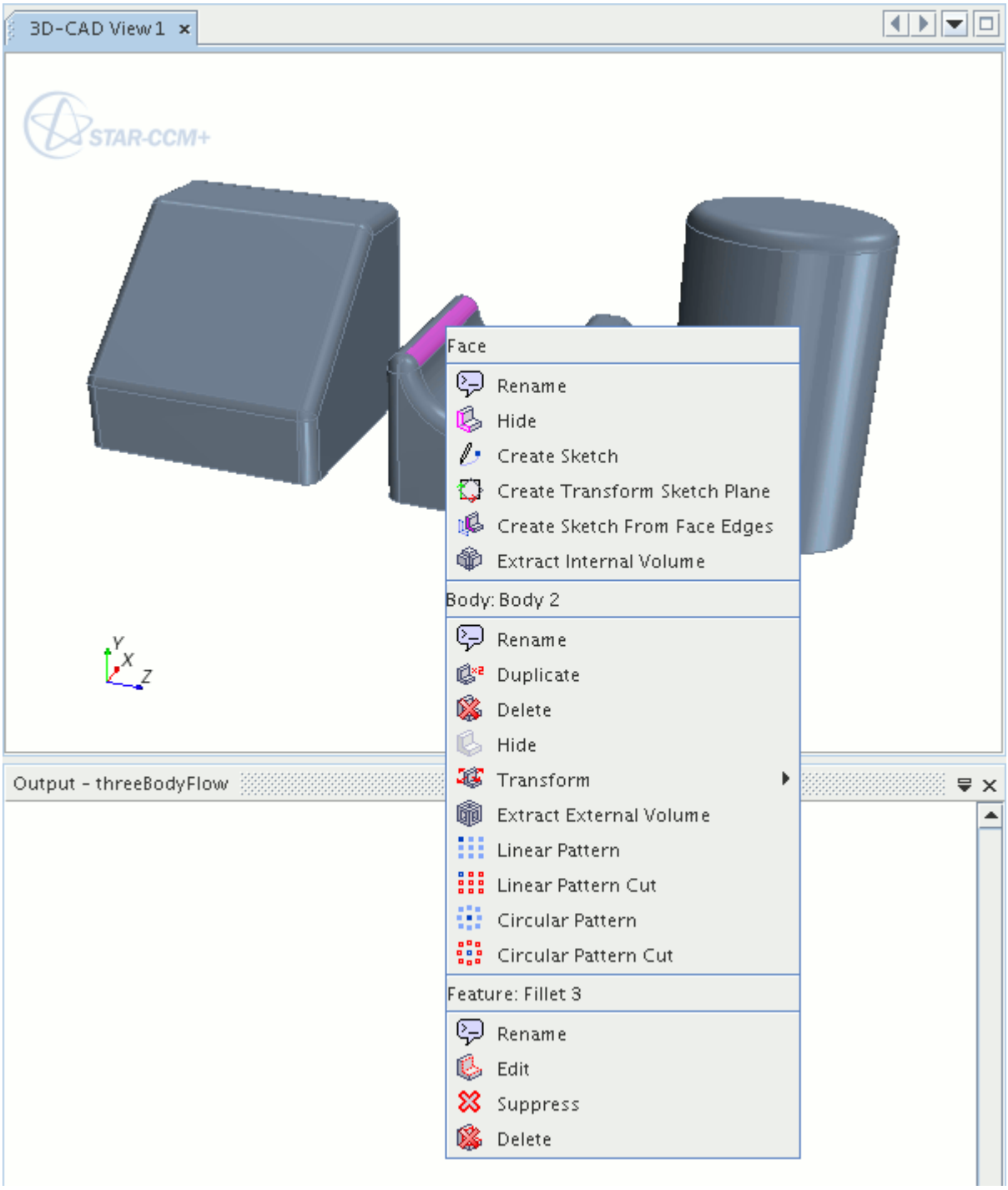


Working with the 3D-CAD View Scene

You can interact with the 3D-CAD model geometry in the *3D-CAD View* scene.

Right-clicking on a face or edge in the *3D-CAD View* scene brings up a menu with options separated into three categories: `Face`, `Body`, and `Feature`. This menu also informs you of the name of the selection, what body it belongs to, and what feature it belongs to. Options that are specific to either a face, feature, or body are listed in the relevant category.

In the following example, the right-click menu informs you that the highlighted selection is a **Face** that belongs to **Fillet 3** and is found on **Body 2**.



Right-clicking on features in the *3D-CAD View* scene is a quick and easy way of identifying them without needing to search the 3D-CAD feature tree. You can get more information about each option that is available in the right-click menu can be found by hovering the mouse pointer over the desired menu item and pressing the `<F1>` key.

You can select either a face or an edge in the *3D-CAD View* scene. To facilitate edge selection on a geometry:

- Set the **Selection Filter** in the Selection Toolbar to **Select Curves**. See Selecting Parts by Type in the Display.

This action makes right-clicking on a specific edge of a geometry model easier, especially on larger, and more complex models.

Working with Features and the Feature Tree

Creating geometry in 3D-CAD involves working with features and the feature tree. This section describes how to roll the feature tree back, forward, and to the end.

This section also explains how to suppress and unsuppress features. Manipulating the feature tree using these tools changes the state of the features and allows you to control where new features are inserted in the feature tree. Different feature state icons appear next to the feature nodes when their state changes.

Note: Right-clicking on a feature node presents a pop-up menu relevant to that particular feature. Some of the actions are common to most of the features, while others are more specific.

What are Features?

STAR-CCM+ uses a feature-based modeling approach in its tools for building 3D-CAD models. In this approach, you construct the 3D-CAD model using a series of operations, or features, each of which represents a particular action.

For example, a sketch feature is added to the model when you construct points, lines and curves on a sketch plane. An extrude feature is added when you select an existing sketch feature and extrude it in a particular direction. The entire set of features that is used to construct a model is stored in the feature tree in the order that you created them. For a full list of the features available in 3D-CAD, see Feature Tree Node Reference.

What is the Feature Tree?

The feature tree of a 3D-CAD model is an ordered sequence of all the actions (or features) that are required to build the final model. The **Features** manager node acts as a parent to all feature nodes within the tree. A full list of these nodes is given in the feature tree node reference section.

The feature tree is dynamic—that is, editing an existing feature in the tree impacts the outcome of subsequent features. The result from one feature acts as the input for the next feature, so that the resulting model is the cumulative effect of applying all the features to the original canonical sketch planes.

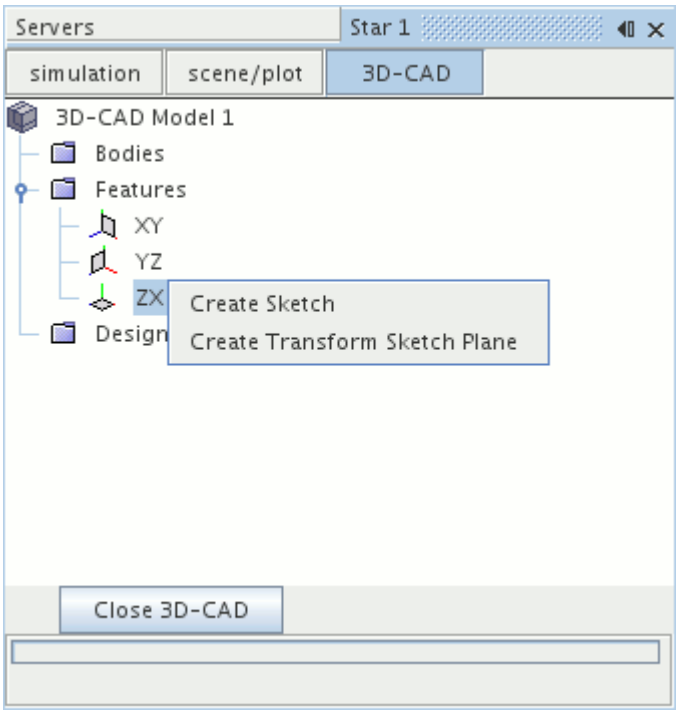
The dynamic nature of the feature tree means that you manipulate it to achieve different forms of the final model. If you roll back the feature tree to a particular point, then only those features that appear before the roll point are executed in the CAD engine. When you add a feature to a rolled-back tree, it is inserted at the roll point, rather than at the end of the tree. Rolling the tree forward (either to a specific feature or to the end of the tree) results in all intermediate features being reactivated within the CAD engine. These items are discussed in more detail in the relevant sections.

You can also suppress features to deactivate them. Suppressed features are not removed from the tree, and can be unsuppressed to reinstate their contribution.

You can switch between the 3D-CAD feature tree and the simulation object tree without having to close 3D-CAD.

Initial State

In its initial state, the feature tree contains nodes for each of the three canonical sketch planes. Right-clicking on any of these three nodes provides actions to create a sketch on the sketch plane, or to create a further sketch plane that is based on the initial canonical sketch plane.



Contents:

Common Feature Actions Reference
Rolling Back
Understanding Where New Features Are Inserted
Reordering Features
Suppressing Features
Understanding Feature States
Creating Basic Geometric Shapes in 3D-CAD
Feature Tree Node Reference
Changing Feature Properties From the 3D-CAD Feature Tree

See Also:

Rolling Back
Understanding Feature States
Understanding Where New Features Are Inserted
Suppressing Features

Common Feature Actions Reference

The following actions are common to nodes across 3D-CAD:

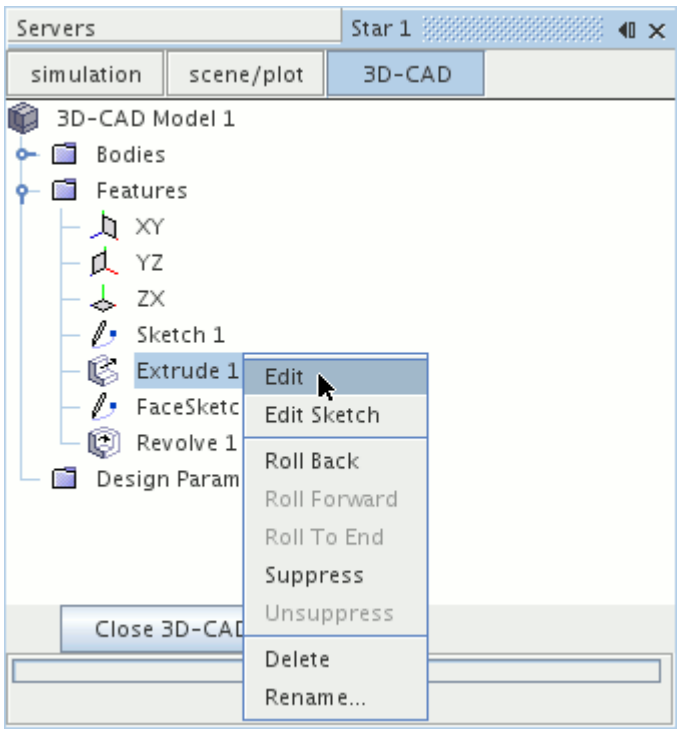
- Edit
- Delete
- Rename

Note: You cannot edit, delete, or rename the three default sketch planes.

Editing a Feature

By editing a feature, the underlying properties of the feature can be altered, such as angle of revolution, sketch dimension, or selected bodies. To edit a feature:

1. Right-click on the feature node in the feature tree and select **Edit**.



   This action automatically rolls back to the selected feature and opens the corresponding panel.

2. Perform the changes that you want.
3. Click **OK** to confirm the changes.

Additional information is available for editing:

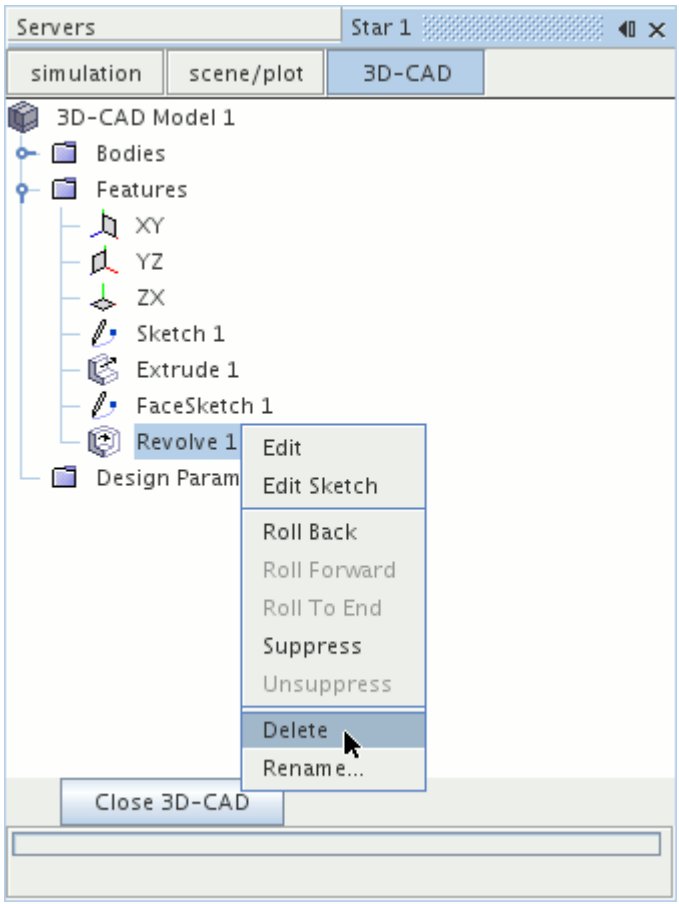
- Sketches
- Lofts
- Sweeps
- Extrusions
- Revolves

Deleting a Feature

You can permanently remove a feature from your 3D-CAD model by deleting it from the feature tree.

To delete a feature:

- Right-click on the feature node and select **Delete**.



The feature is removed from the feature tree and the 3D-CAD model. If the selected feature is a parent feature— that is, a feature upon which other features depend, it cannot be deleted until its child features have been deleted.

Caution: This action is irreversible.

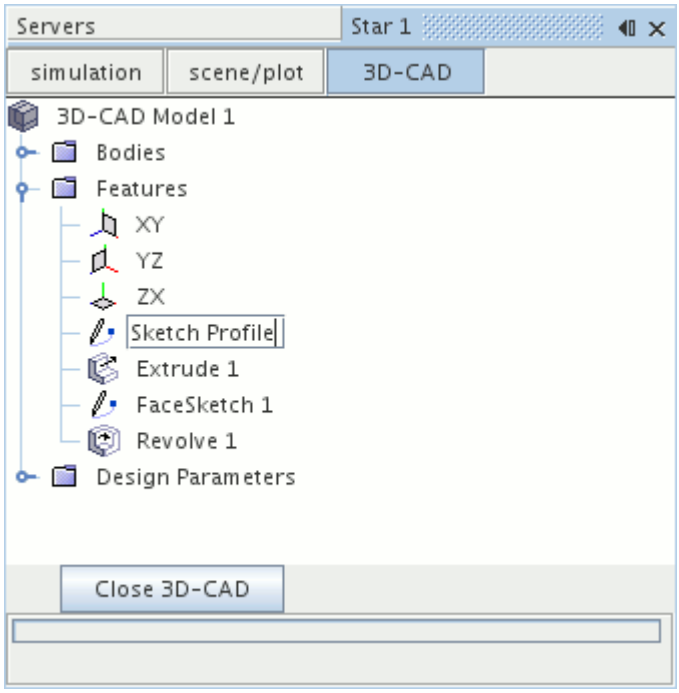
Additional information is available for deleting design parameters. See Deleting a Design Parameter.

Renaming a Feature

You can rename most feature nodes in the 3D-CAD feature tree. This capability is useful for complex models containing many features.

To rename a feature:

- Use the inline editor



- Alternatively, use the right-click menu.

Similarly, you can rename body and design parameter nodes.

See Also:

Renaming Objects
What Is a Design Parameter?
Overview of Bodies in 3D CAD

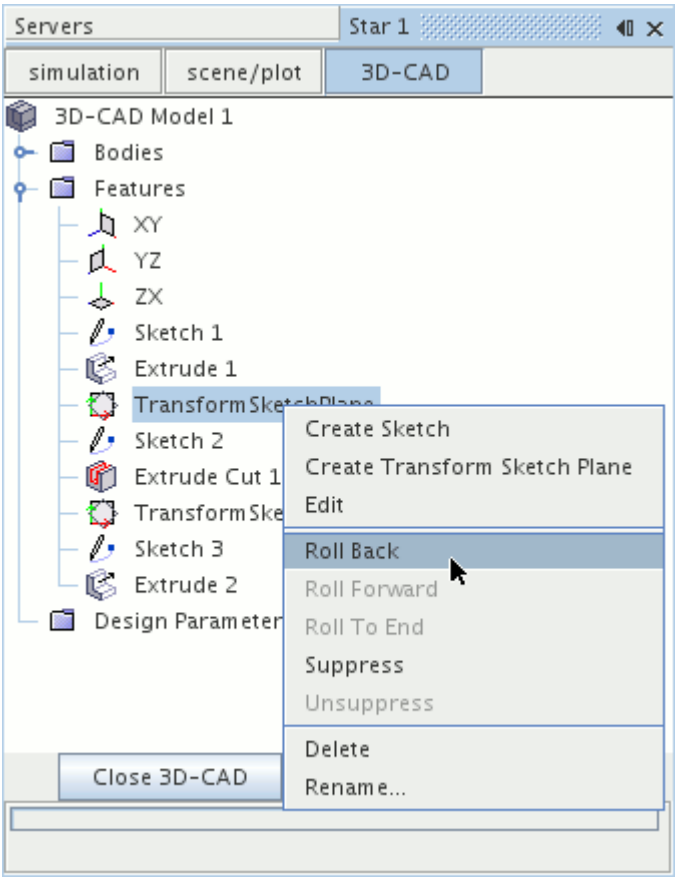
Rolling Back

When you roll back to a particular feature, you temporarily deactivate that feature and all subsequent features.

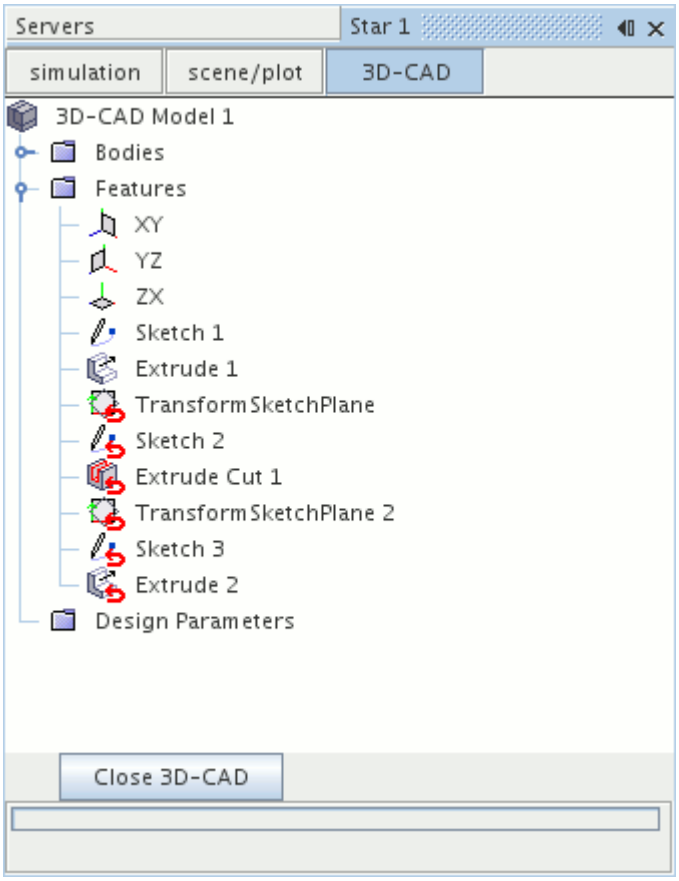
While in the rolled back state, you can insert new features in the feature tree. These features are placed directly above the rolled-back features, and will remain there after you roll forward. The 3D-CAD model is only constructed from active features. Rolled-back features are overlaid with the roll back symbol .

To roll back to a particular point:

- Right-click the feature node that you want to roll back to and select **Roll Back**.



The selected node, and those following to the end of the feature tree, are deactivated.

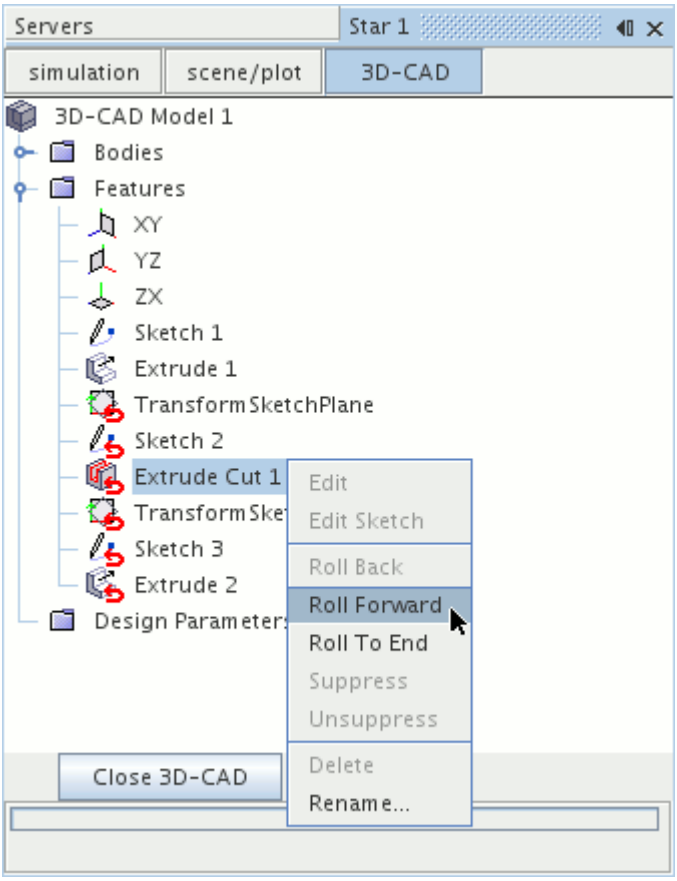


Rolling Forward

When the 3D-CAD feature tree contains rolled back features, you can roll forward to one of those features. Rolling forward to a particular feature reactivates that feature and all features that come before it. The CAD engine includes these reactivated features in the construction of the 3D-CAD model.

To roll forward to a feature:

- Right-click the feature node that you want to roll forward to and select **Roll Forward**.



This node, and all rolled-back nodes that come before it, are reactivated in the feature tree.

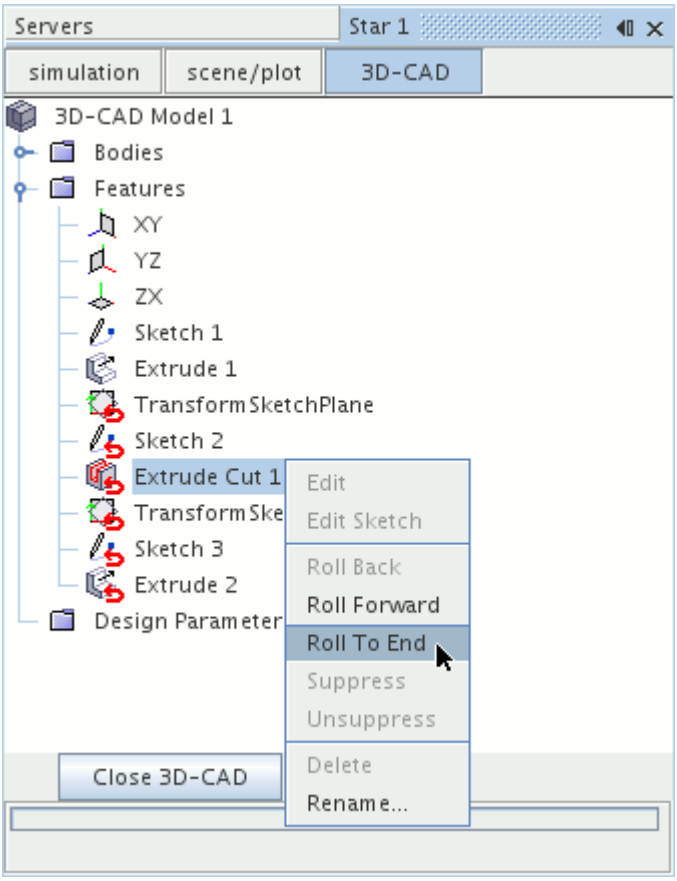
Rolling to the End

When the 3D-CAD feature tree contains rolled-back features, rolling to the end reactivates all features in the feature tree. The CAD engine includes all active features in the construction of the 3D-CAD model.

Note: Features that were suppressed before applying the roll back will remain suppressed after rolling to the end.

To roll to the end:

- Right-click a rolled-back node in the feature tree and choose **Roll To End**.

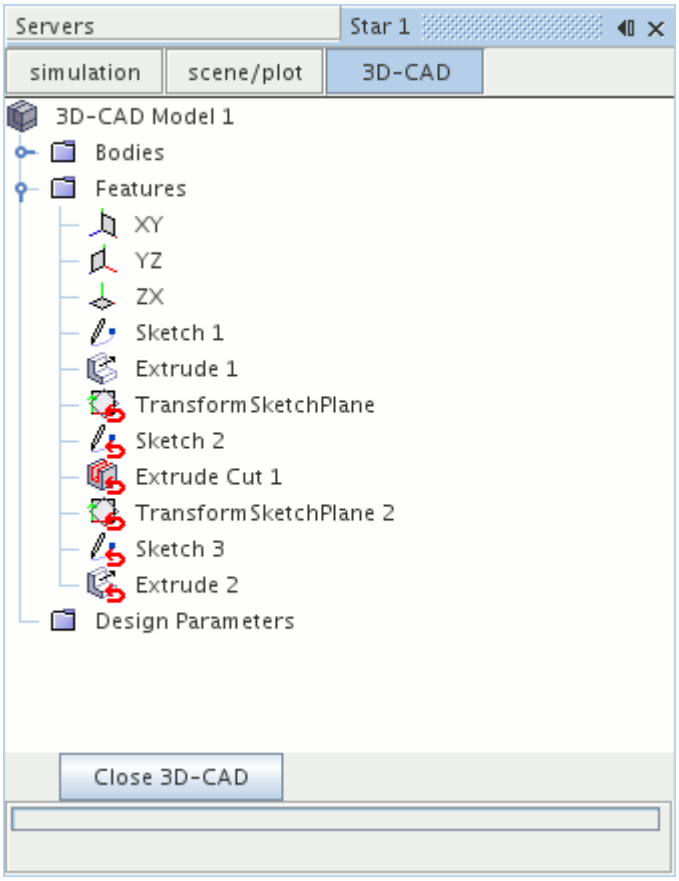


All rolled back features are reactivated.

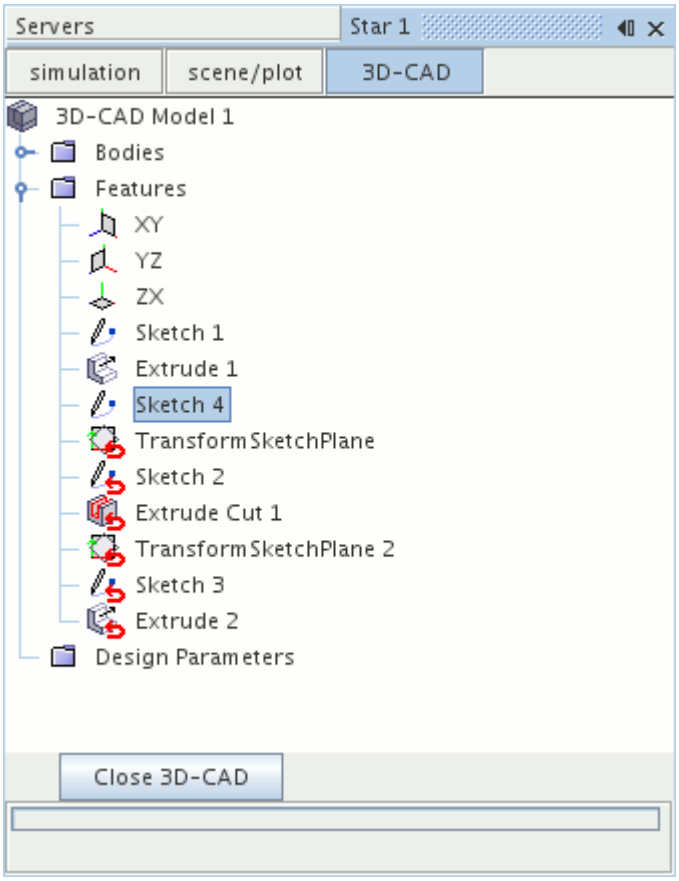
Understanding Where New Features Are Inserted

The point at which new features are added depends on whether you have rolled back the feature tree. If you rolled back the feature tree, the new features are added after the last active feature. If you did not roll back the feature tree, new features are added at the end.

In the following example, the feature tree is rolled back to the **TransformSketchPlane** feature.



If you create a new feature at this point (for example, a sketch), it is inserted before the first rolled-back feature, **TransformSketchPlane**.



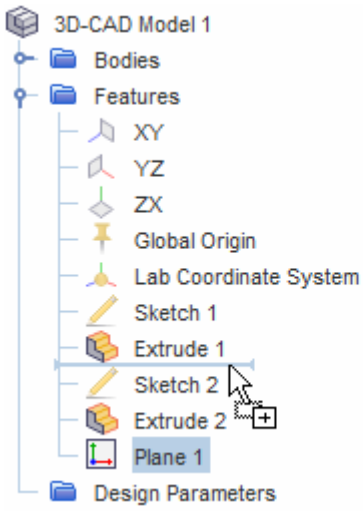
Reordering Features

You can reorder the position of features in the feature tree.

Features that are dependant on another feature cannot be moved to appear before the feature that they depend on. You can only reorder a single feature at a time.

To reorder a feature:

1. In the feature tree, click and drag the desired feature up or down the feature tree.



A line appears in the feature tree at the location where the mouse pointer is to indicate where the selected feature will be placed.

The feature is placed in the selected position. All features that come below the insertion point become out of date and display the (**Out-of-Date**) status.

2. Click **Update 3D-CAD** to rebuild the feature list.

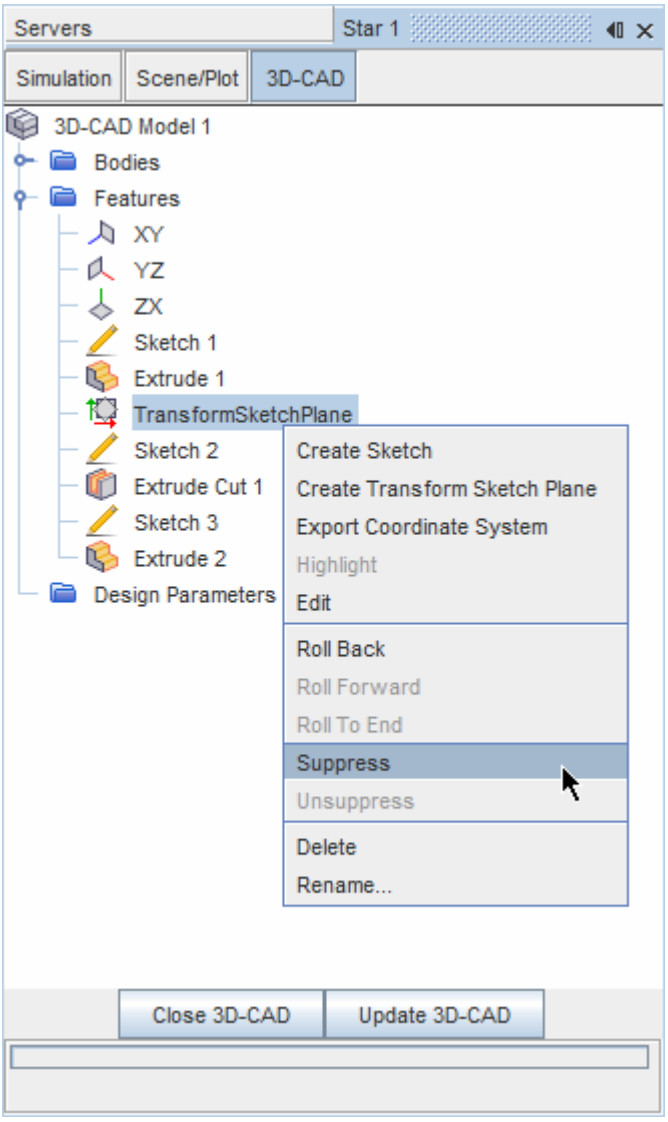
Suppressing Features

Suppressing a feature deactivates it and any features that depend on it.

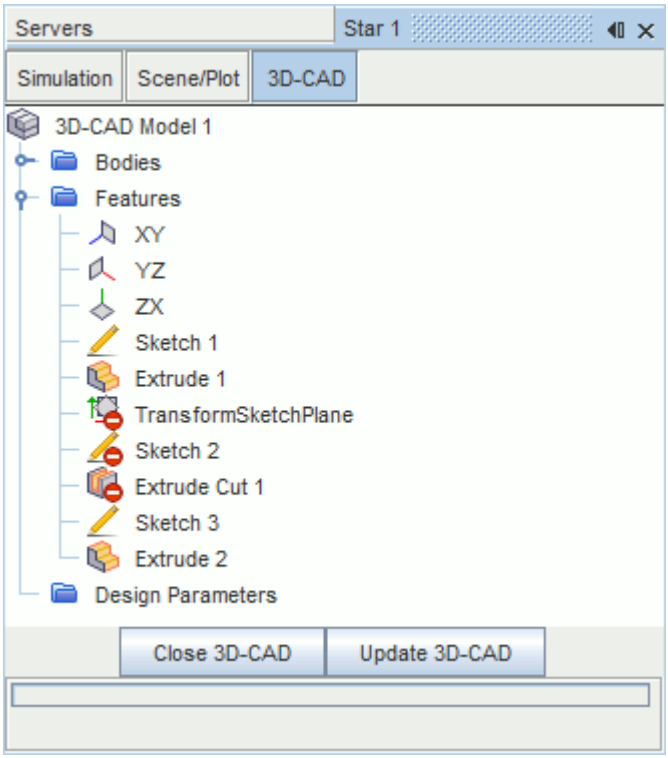
Suppressed features are indicated by the symbol. You can suppress multiple features at once by selecting multiple feature nodes in the feature tree using the <`Ctrl`> and <`Shift`> keys.

To suppress a feature:

- Right-click the feature that you want to suppress and choose **Suppress**.



The selected feature is suppressed. The child features of the sketch plane (**Sketch 2** and **Extrude Cut 1**) are also suppressed.



Note: Suppressing a feature can invalidate subsequent features in the feature tree, and cause warning dialogs to appear if the CAD engine fails to reconstruct the model.

Unsuppressing Features

You can unsuppress a previously suppressed feature to reinstate its operation in the construction of the model.

To unsuppress a feature:

- Right-click the suppressed features in the feature tree and select **Unsuppress**.

The selected features are unsuppressed and the suppressed icon overlays are removed. If the suppressed features have dependents (which would also have been suppressed), it is possible to unsuppress the dependent features first. This action also unsuppresses the parent feature. The precise behavior depends on the nature of the feature tree. You must unsuppress child features if you first unsuppressed the parent feature. You can unsuppress multiple features at once by selecting them in the feature tree using the <`Ctrl`> and <`Shift`> keys.

Understanding Feature States

Each feature node has a unique icon to indicate what the feature does.

If the feature is in its normal state, and it is correctly constructed by the CAD engine, the icon appears as normal. If the feature is not in its normal state, an additional symbol appears on the feature icon to indicate the alternate state. The possible states are shown in the table below:

Status	Description
Rolled Back	The feature is further down the tree than the roll back point and so is not executed in the CAD engine.
Suppressed	The feature, or its parent feature, is suppressed and is not executed in the CAD engine.
Invalid	The CAD engine has failed to update or execute the feature during model construction. Editing an invalid feature can allow you to work out the cause of the problem, and correct it accordingly.
Out-of-Date	Indicates that a feature is out-of-date and requires updating.

See Also:

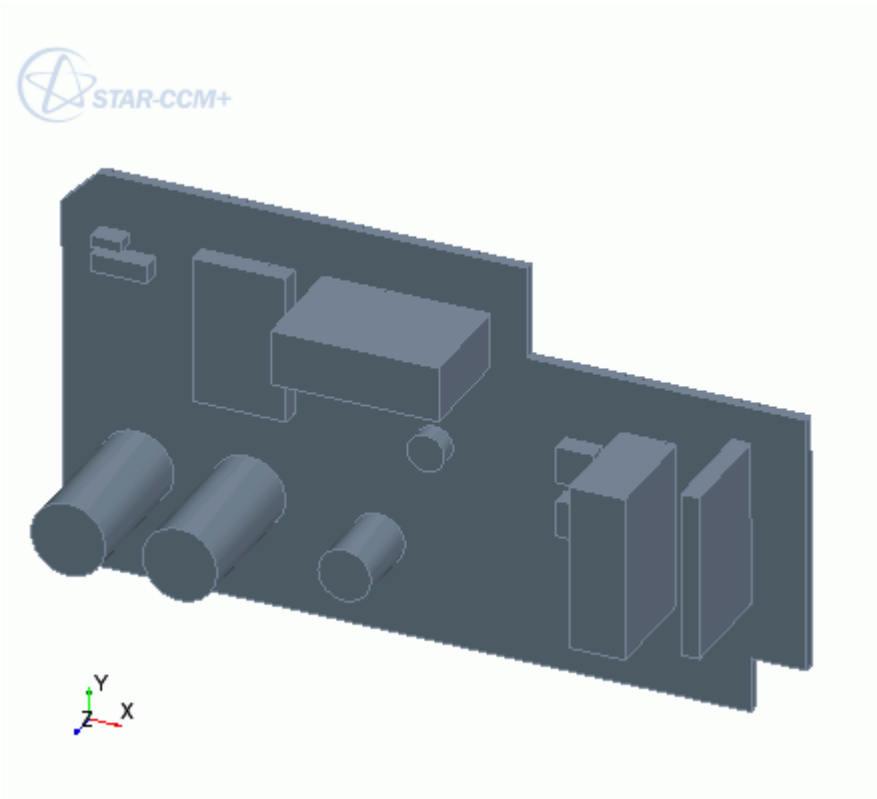
Feature Tree Node Reference

Rolling Back

Suppressing Features

Creating Basic Geometric Shapes in 3D-CAD

In electronics cooling applications, most electronic components are represented by basic geometric shapes such as blocks and cylinders. 3D-CAD allows you to create these simple shapes in a single operation without having to create a sketch first.

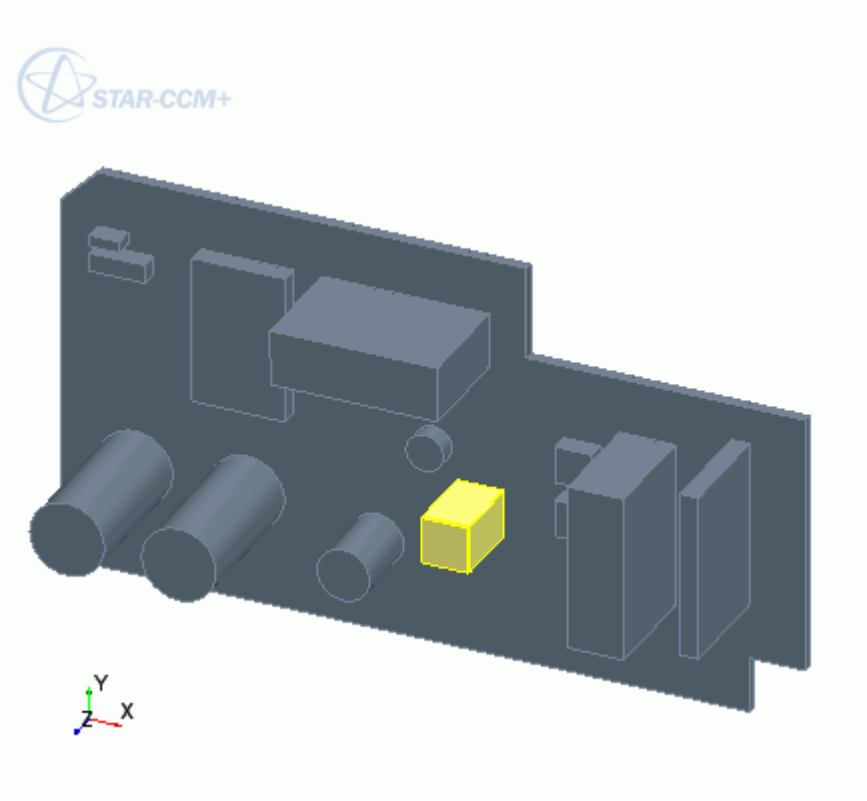


Note: In the electronics cooling industry, the circuit board and its various components are defined in the XY plane. 3D-CAD uses this convention when you create simple shapes.

To create a basic shape:

1. Right-click the **Features** node and select:
   - **Solid Primitive** > **Block** to create a block.
   - **Solid Primitive** > **Cylinder** to create a cylinder.
2. In the subsequent dialog, set the shape parameters.
   - For blocks, see Create Block Panel.
   - For cylinders, see Create Cylinder Panel.
3. Expose any properties that you want as design parameters by clicking (**Expose Parameter)** next to the property.

   The *3D-CAD view* scene updates to show a preview of the body.



4. Click **OK**.

Create Block Panel

Use the following properties to define a simple block shape in 3D-CAD. The block extends symmetrically from the center point.

Length (X Distance)	Specifies the total size of the block in the X direction.	
Width (Y Distance)	Specifies the total size of the block in the Y direction.	
Height (Z Distance)	Specifies the total size of the block in the Z direction.	
Center	Specifies the center point of the base of the shape.	
	X	X coordinate.
	Y	Y coordinate.
	Z	Z coordinate.
Coordinate System	Specifies the coordinate system in which you define the center point.	

Create Cylinder Panel

Use the following properties to define a simple block shape in 3D-CAD.

Radius	Specifies the radius of the cylinder.

Height	Specifies the amount that the cylinder extends outwards from the center point.	
Center	Specifies the center point of the base of the shape.	
	X	X coordinate.
	Y	Y coordinate.
	Z	Z coordinate.
Coordinate System	Specifies the coordinate system in which you define the center point.	

See Also:

What Is a Design Parameter?

Feature Tree Node Reference

The following features can be used to create a 3D-CAD model:

	XY	The XY canonical sketch plane. It cannot be edited or deleted.
	YZ	The YZ canonical sketch plane. It cannot be edited or deleted.
	ZX	The ZX canonical sketch plane. It cannot be edited or deleted.
	Chamfer	Added when a chamfer is applied to edges of a body.
	Circular Pattern	Added when a circular pattern is created from a body.
	Circular Pattern Cut	Added when a circular pattern cut is created from a body.
	Create Sheet Bodies	Added when a sheet body is created from a solid body, or another sheet body.
	Delete Body	Added when a body is deleted.

	Delete Faces	Added when a face is deleted.
	Duplicate Body	Added when a body is duplicated.
	Extract External	Added when the external volume of space around a model is extracted and used to create a new solid body.
	Extract Internal	Added when the internal volume of space within a model is extracted and used to create a new solid body.
	Extrude	Added when a sketch is extruded.
	Extrude Cut	Added when an extrude cut feature is used to remove material from the model by sweeping a sketch in a specific direction.
	Face Sketch	Added when a sketch is created by extracting the edges of a face.
	Filled Surface	Added when a filled surface is created.
	Fillet	Added when a fillet is applied to the edges of a body.
	Import Cad	Added when geometry is imported into 3D-CAD from an external CAD file.
	Imprint	Added when two or more bodies are imprinted onto each other.
	Intersect Bodies	Added when two or more bodies are intersected to find the common volume that is shared by the bodies.
	Linear Pattern	Added when a linear pattern is created from a body.

	Linear Pattern Cut	Added when a linear pattern cut is created from a body.
	Loft	Added when two or more sketches are lofted to create a body.
	Loft Cut	Added when a lofted cut, created from two or more sketches, is used to remove material from the model.
	Mirror Body	Added when one or more bodies is mirrored.
	Move Body	Added when one or more bodies is translated.
	Revolve	Added when a sketch is revolved.
	Revolve Cut	Added when a revolve cut feature is used to remove material from the model by revolving a sketch around an axis.
	Rotate Body	Added when one or more bodies is rotated.
	Scale Body	Added when one or more bodies is scaled.
	Sew Sheet Bodies	Added when one or more sheet bodies are sewn together.
	Sketch	Added when a sketch is created.
	Subtract Bodies	Added when one or more bodies is used to subtract material from a target body.
	Sweep	Added when a sketch is swept along a profile to create a body.

	Sweep Cut	Added when a swept cut, which is created by sweeping a sketch along a profile, is used to remove material from the model.
	Unite Bodies	Added when two or more bodies are united.

See Also:

Sketches
Creating Bodies From Sketches
Working with Patterns
Working with Volume Extraction
Working with Fillets and Chamfers
Working with Bodies

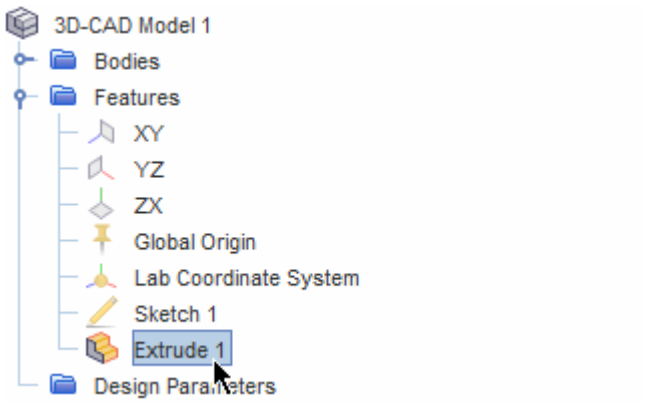
Changing Feature Properties From the 3D-CAD Feature Tree

You can edit the properties of a feature at the feature tree level, without having to enter the feature.

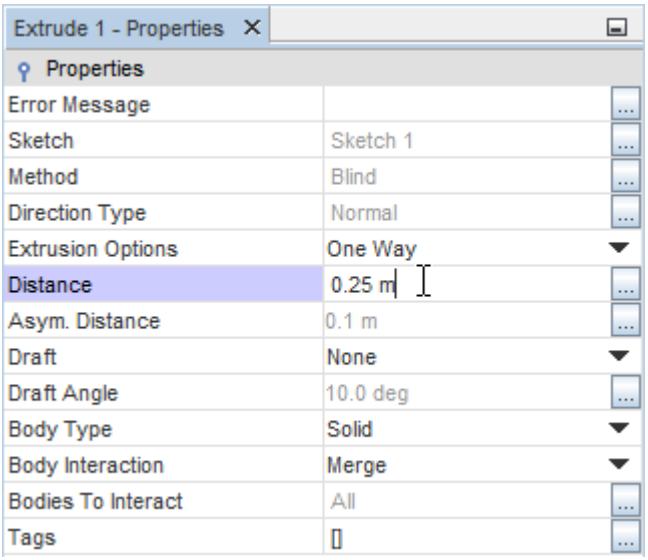
This functionality allows you to modify feature properties more quickly.

To edit feature properties in the feature tree:

1. In the 3D-CAD feature tree, select the feature that you want to modify.



2. In the *Properties* window, modify the desired property of the feature and press <`Enter`>.



The feature is marked as out of date and displays the ⚠ (**Out-of-Date**) status.

3. Click **Update 3D-CAD**.

   The CAD geometry is updated with the changes to the feature.

Working with the 3D-CAD Toolbar

You can create features in 3D-CAD using multiple methods. The default method is to right-click on objects in the 3D-CAD feature tree and in the *3D-CAD View* scene. An alternate method is to use the 3D-CAD toolbar.

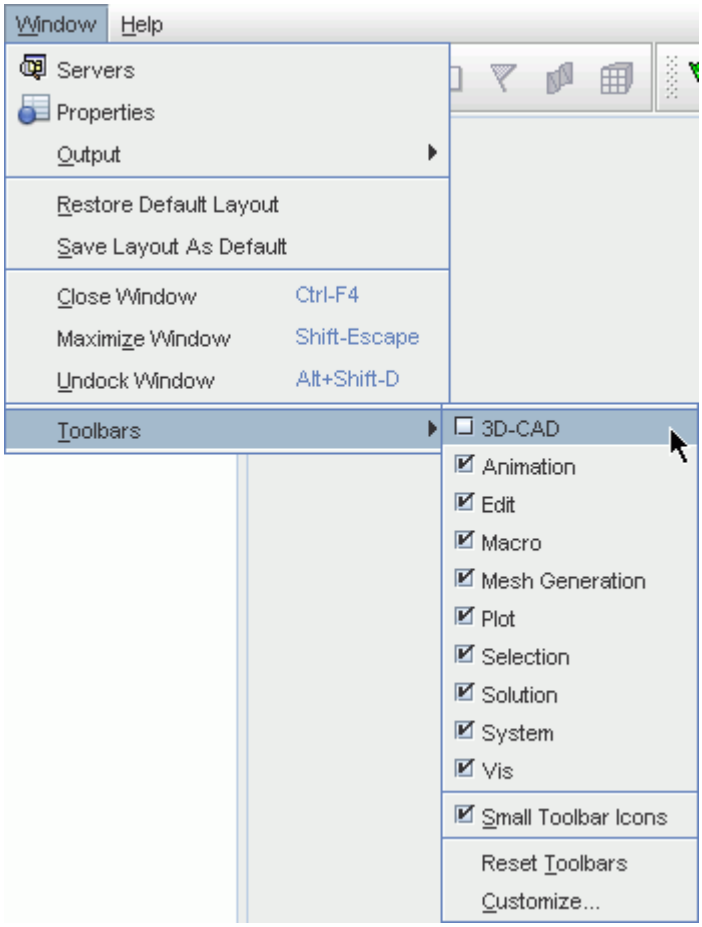
This section contains the following topics:

- Activating the 3D-CAD Toolbar
- Using the 3D-CAD Toolbar

Activating the 3D-CAD Toolbar

By default, the 3D-CAD toolbar is deactivated. To activate the 3D-CAD toolbar:

- Select **Window** > **Toolbars** > **3D-CAD**.



The 3D-CAD toolbar is activated and appears next to the Selection toolbar. See Selecting Parts by Type in the Display.

Using the 3D-CAD Toolbar

There are two ways to use the 3D-CAD toolbar:

- Pre-selecting
- Post-selecting

The first method involves selecting an item before using the toolbar. For example, selecting a sketch profile before using the extrude function. The second method does not have any item active when using the toolbar, instead the desired item is selected after using the toolbar, when the feature panel is open.

The 3D-CAD toolbar is shown below.

The operations that are provided in the toolbar include:

- Create Sketch

- Create Transform Sketch Plane
- Create Extrude
- Create Extrude Cut
- Create Revolve
- Create Revolve Cut
- Create Sweep
- Create Loft
- Create Fillet
- Create Chamfer

Contents:

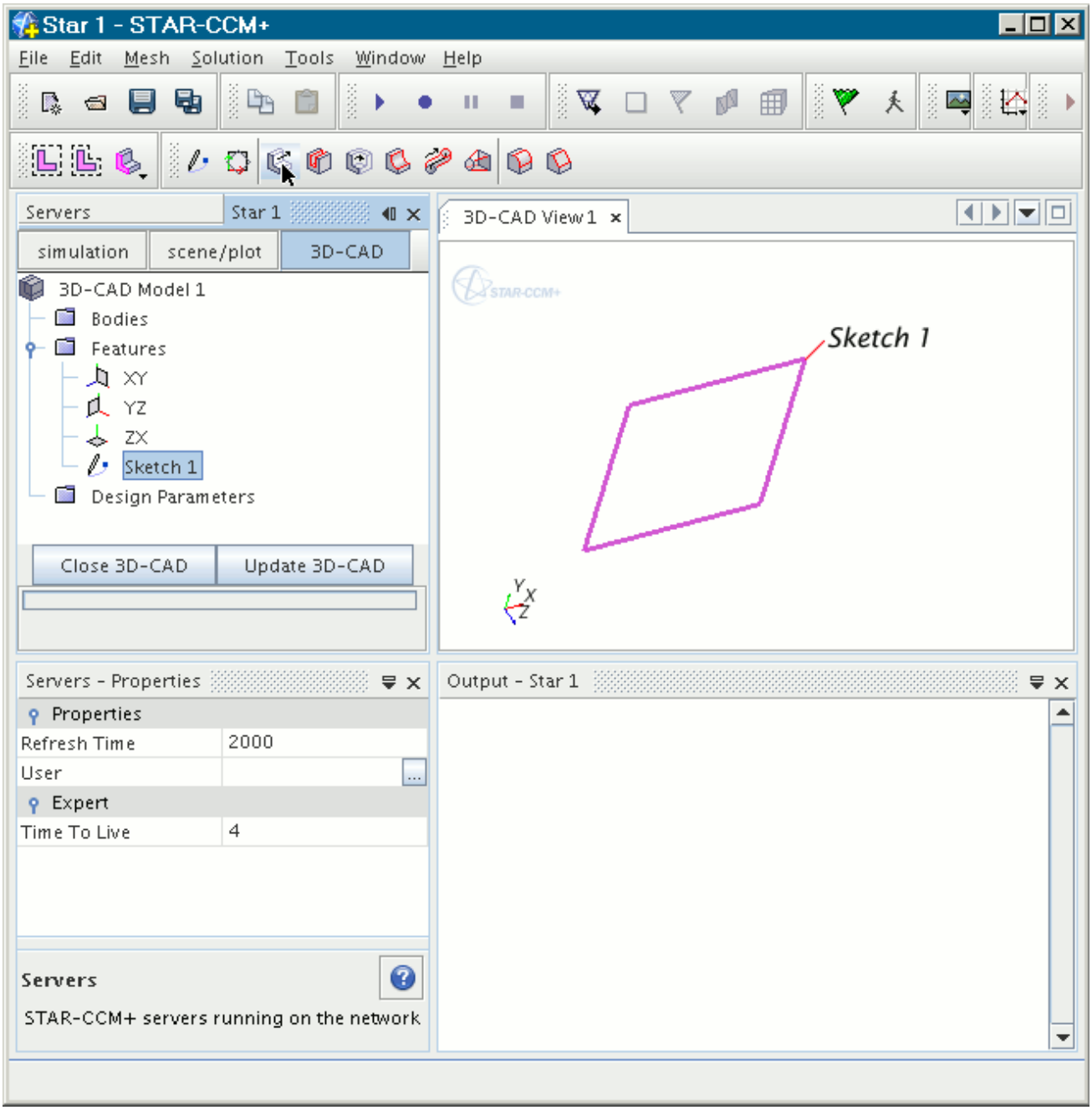
Pre-Selecting
Post-Selecting

Pre-Selecting

Pre-selecting means that you select the face, edge, or sketch to use with a feature before you click the relevant button in the 3D-CAD toolbar.

Using this method takes you directly to the relevant feature panel.

For example, if you select a sketch in the feature tree and then select the **Create Extrude** button in the 3D-CAD toolbar, you are taken to the *Extrude* panel.



Post-Selecting

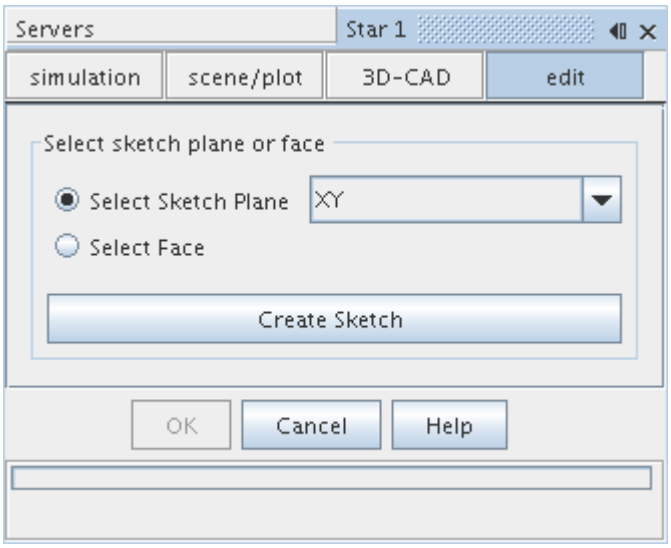
Post-selecting means that you use the 3D-CAD toolbar with nothing pre-selected.

If no selection is active, the *Select Sketch Plane or Face* panel appears. Select a suitable face, edge, or sketch to continue to the relevant feature panel.

This panel differs slightly depending on what feature you are creating.

Creating a Sketch and Transform Sketch Plane

If you click **Create Sketch** on the 3D-CAD toolbar, and no selections are active in 3D-CAD, the *Select Sketch Plane or Face* panel appears.



1. Select the desired sketch plane from the drop-down list or a face from an existing body.
2. Click **Create Sketch** to continue to the *Sketch* panel.

The same procedure applies when you create a transform sketch plane. See Creating a Transform Sketch Plane.

Creating Any Other CAD Features

When you create any other CAD using the 3D-CAD toolbar, the normal feature panel appears.

1. Select the sketch or edge that you want to use from the *3D-CAD View* scene.
2. Set up the feature as described in the relevant feature section.

Working with the Visibility Toolbar in 3D-CAD

The Visibility toolbar is designed to make working with complex models easier. With this toolbar, you can control how geometry is displayed in the Graphics window as well as what is displayed.

The Visibility toolbar, , appears at the top of the *3D-CAD View* scene and remains there throughout the 3D-CAD session. You cannot close or hide the Visibility toolbar.

The toolbar has three main functions:

- Color Mode
- Model Visibility
- Reference Geometry Visibility

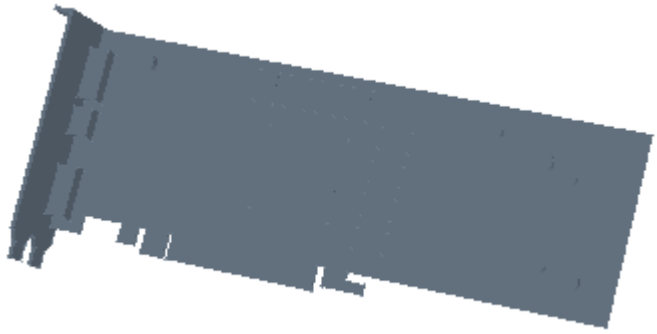
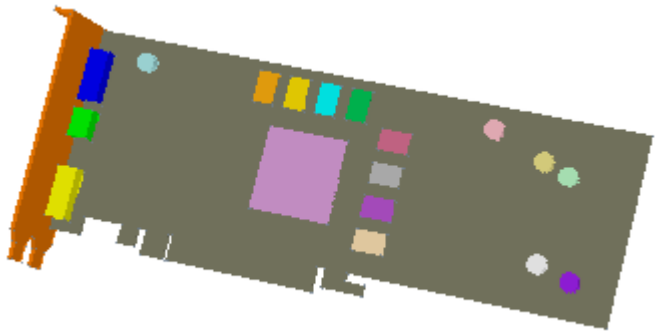
To use the toolbar:

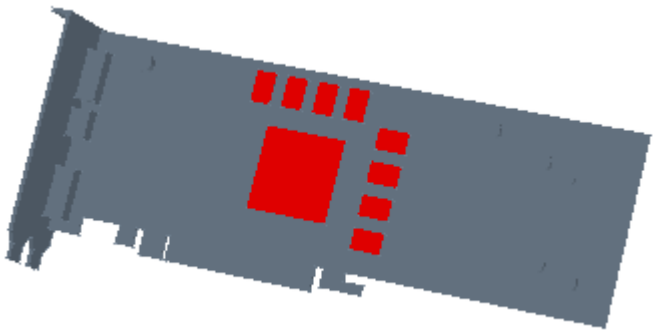
- Click the desired function and select an appropriate option from the drop-down menu.

All the toolbar options are stored in the simulation file and restored next time that you load the file.

Color Mode Options

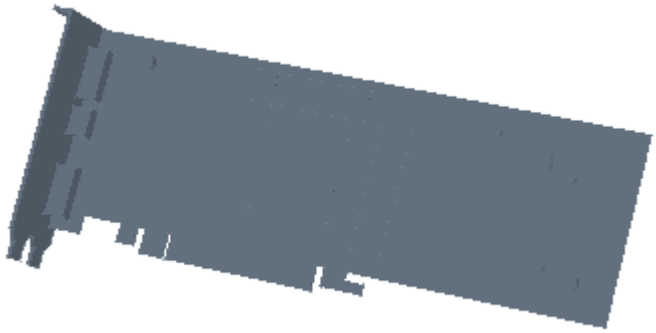
The color mode controls how the geometry is displayed in the Graphics window. You can activate the Solid Bodies, Sheet Bodies, and Edges options individually or in combination.

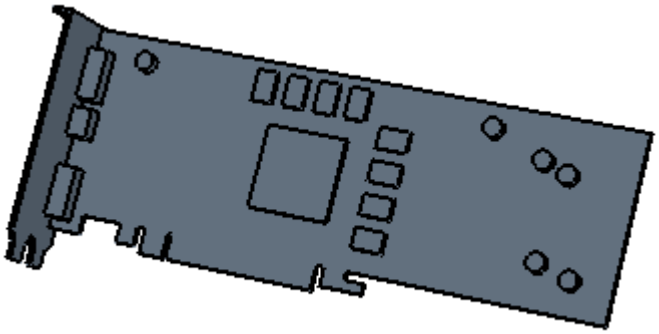
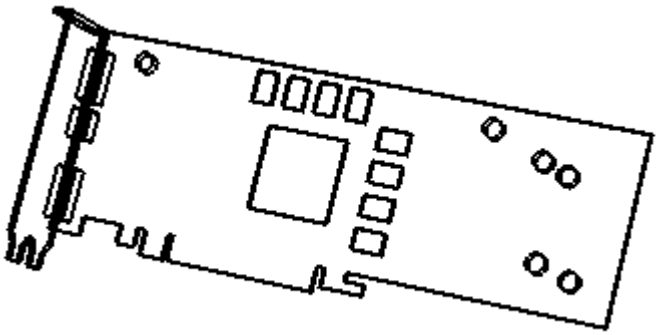
Option	Description
(**Default Color**)	Colors all solid bodies in the CAD model the default gray color, and all sheet bodies a gold color. This is the default option. 
(**Distinguish Bodies**)	Colors each body in the CAD model a different color. Bodies are assigned distinct colors until they are all used, at which point the colors are cycled. Neighboring bodies can have the same color. Colors that are assigned to a particular body are random. Once a color is assigned to a particular body, the color can change when you load the model in a different session. In some cases, you may need to revert to the default color scheme as the assigned color can conflict with feature-specific colors, such the previews. 
(**Distinguish Named Faces**)	Colors named faces in the CAD model a different color. Faces that have the **Default** name are assigned the default gray color, even if the **Default** node is renamed. All faces that correspond to an assigned name are given the same

Option	Description
	color. Faces with different names are assigned distinct colors until they are all used, at which point the colors are cycled. In the following example, the CPU and memory chips have the same name. In some cases, you may need to revert to the default color scheme as the assigned color can conflict with feature-specific colors, such the previews. 

Model Visibility Options

The model visibility controls what is displayed in the Graphics window.

Option	Description
All	Displays solid bodies, sheet bodies, and edges in the Graphics window.
Solid Bodies	Displays solid bodies in the Graphics window. By default, this option is activated.
Sheet Bodies	Displays sheet bodies in the Graphics window. By default, this option is activated.
Edges	Displays edges in the Graphics window. By default, this option is deactivated. When there are multiple bodies or levels in a geometry, many details are hard to see without the edges being visible. 

Option	Description
	Turning on edge visibility makes it easier to see the details.  Turning off all visibility options except the edges provides a wireframe view of the model. 

Reference Geometry Visibility Options

The reference geometry visibility controls the display of reference geometry. You can choose to display or hide the following reference geometries:

- Reference Points
- Axes
- Planes
- Coordinate Systems

You can activate any combination of options to choose what is displayed in the Graphics window. For more information about reference geometry, see Working with Reference Geometry in 3D-CAD.

Surface Modeling in 3D-CAD

Surface modeling is the representation of three-dimensional geometry using zero-thickness surfaces.

In 3D-CAD, zero-thickness surfaces are called sheet bodies. 3D-CAD provides several functions that allow you to create and modify sheet bodies. You can also convert a sheet body to a solid body, or create a sheet body from a solid body.

Two important functions of surface modelling in 3D-CAD are that it allows you to repair imported CAD geometry, and also create zero-thickness baffles.

In 3D- CAD, sheet bodies are created in the same way as solid bodies—there are no separate features or toolbars. When it is possible to create a sheet body or a solid body, the *Body Type* option appears in the relevant feature panel, allowing you to choose what type of body you want to create. Currently, you cannot trim or remove areas from sheet bodies. The following tools are available for sheet bodies:

- Extrude
- Revolve
- Loft
- Sweep
- Chamfers
- Fillets
- Boolean operations
- Transform operations

3D-CAD enables you to perform the following useful functions with your existing geometries:

- Filling holes with sheet bodies
- Sewing sheet bodies together to form a solid body
- Creating a sheet body from a solid body

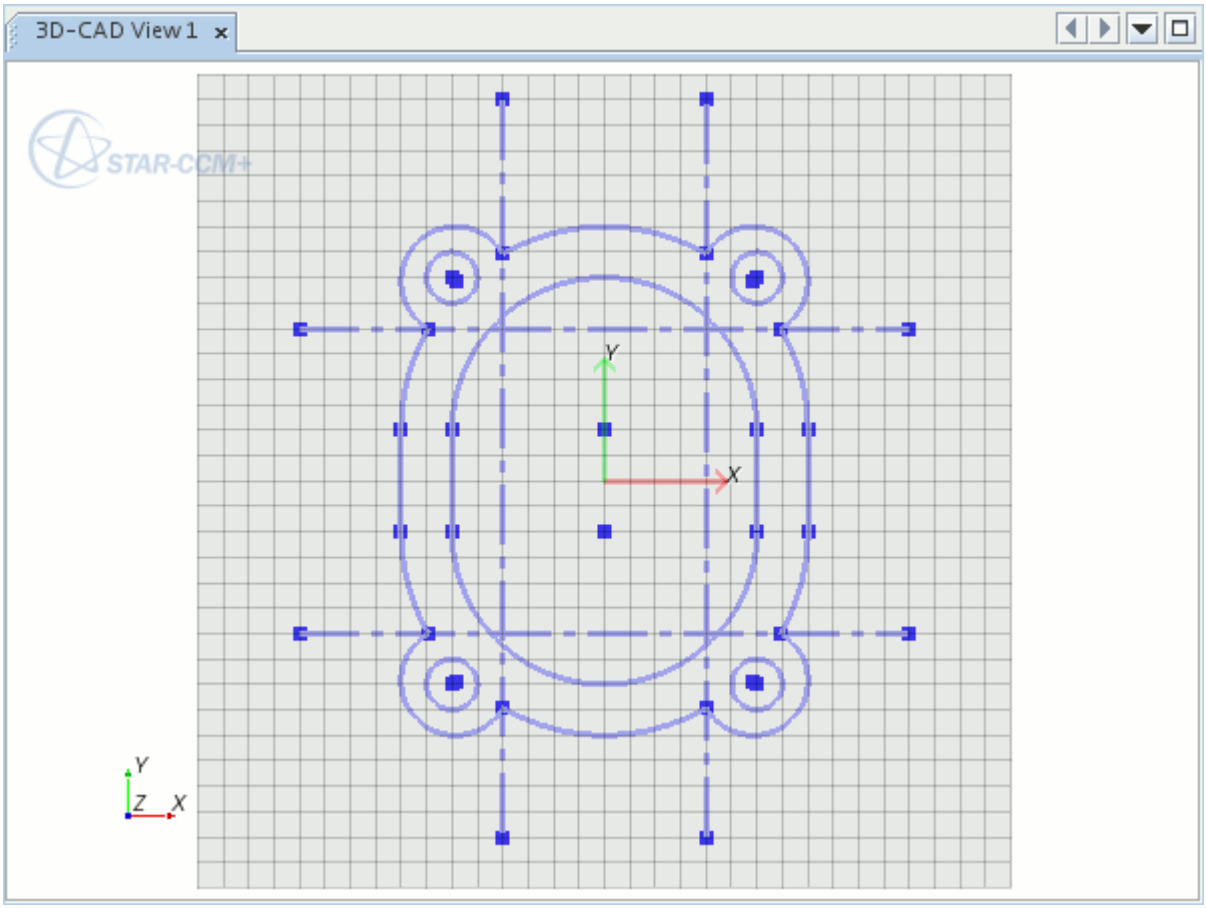
Sketches

Sketches are two-dimensional profiles that are created on a planar surface. They are composed from edges and points.

Many of the operations for creating solid bodes take sketches as their inputs. For example, the extrude operation requires a sketch as an input, along with the direction vector in which to extrude the sketch.

Every sketch is created on a sketch plane, which defines the position and orientation of the sketch in the 3D-CAD model.

You construct a sketch from sketch entities: lines, circles, arcs, splines, and more. Sketch entities are defined using the local coordinate system on the sketch plane.



Face Sketches

A Face Sketch is a sketch that you create by extracting the edges of a face on a solid body. You cannot modify these converted entities. The link between these sketch entities and the face is retained so that changes you make to the body face are propagated to the Face Sketch. The sketch entities that are linked to the face cannot be edited through splitting or deleting operations, but you can convert them to construction entities.

Contents:

Creating a Sketch
Creating a Sketch from Existing Edges
Creating a Sketch at the Center of a Face
Creating a Sketch in 3D
Trimming Sketches
Sketch Planes
Sketch Entity Reference
Sketch Display Options
Viewing the Status of Sketch Primitives
Splines
Sketch Entity Operations

Creating Sketch Fillets
Creating Sketch Chamfers
Projecting Entities to the Active Sketch
Showing/Hiding Relations
Importing and Replacing Sketch Curves

See Also:

Splitting and Trimming Sketch Entities
Deleting and Duplicating Entities
Working with Construction Entities

Creating a Sketch

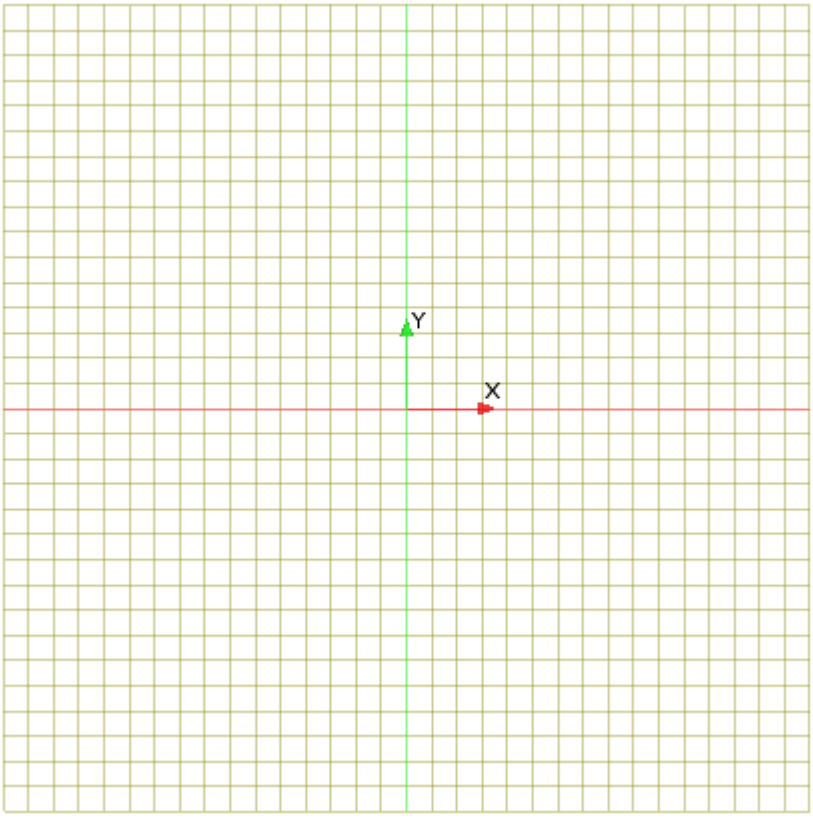
Creating a sketch is often the first step in creating a solid body. Sketches are associated with planar surfaces, which are known as sketch planes.

To create a sketch:

1. Identify the planar surface on which to create the sketch. This planar surface can be one of the default sketch planes, a previously created transform sketch plane, or the surface of an existing body.

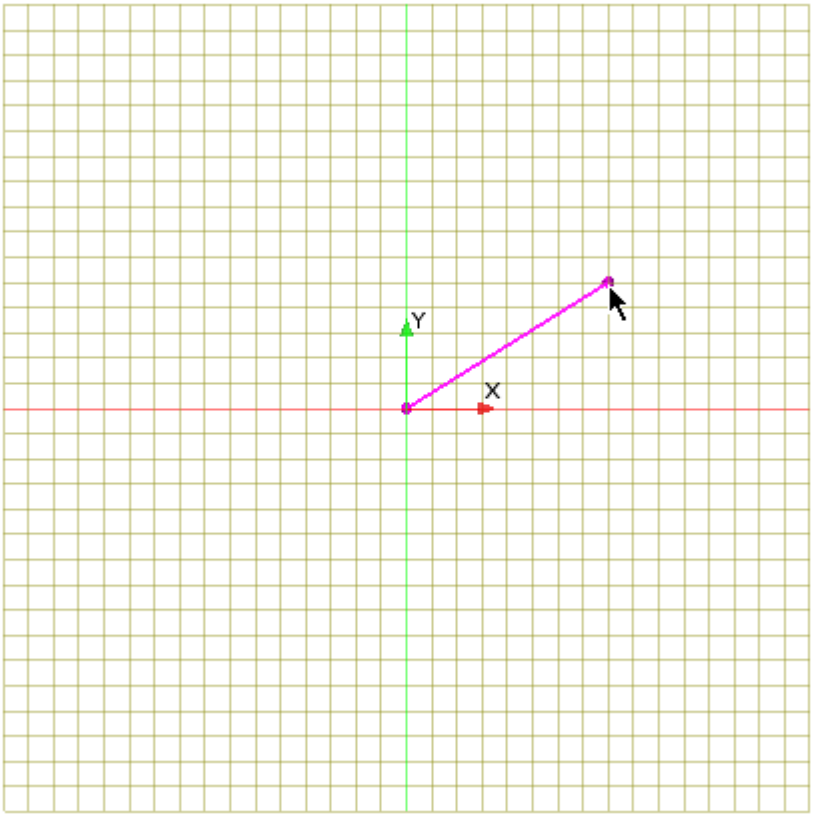
   See Sketch Planes.
2. Activate the sketch tool on the chosen surface. The method that you use depends on the type of surface:
   - For a default sketch plane, or a transform sketch plane, right-click the corresponding feature node in the feature tree, or the actual feature in *Graphics* window and choose **Create Sketch**.
   - For the face of an existing body, right-click the face in the *Graphics* window and choose **Create Sketch** > **On Face**.
   - To place the sketch at the face center on an existing body, right-click the face in the *Graphics* window and choose **Create Sketch** > **At Face Center**.

   The *Sketch* panel appears in the left-hand side of the window. The *Sketch* panel contains the *Create Sketch Entities* and *Display Options* boxes. A sketch grid with X and Y axes appears within the *Graphics* window.

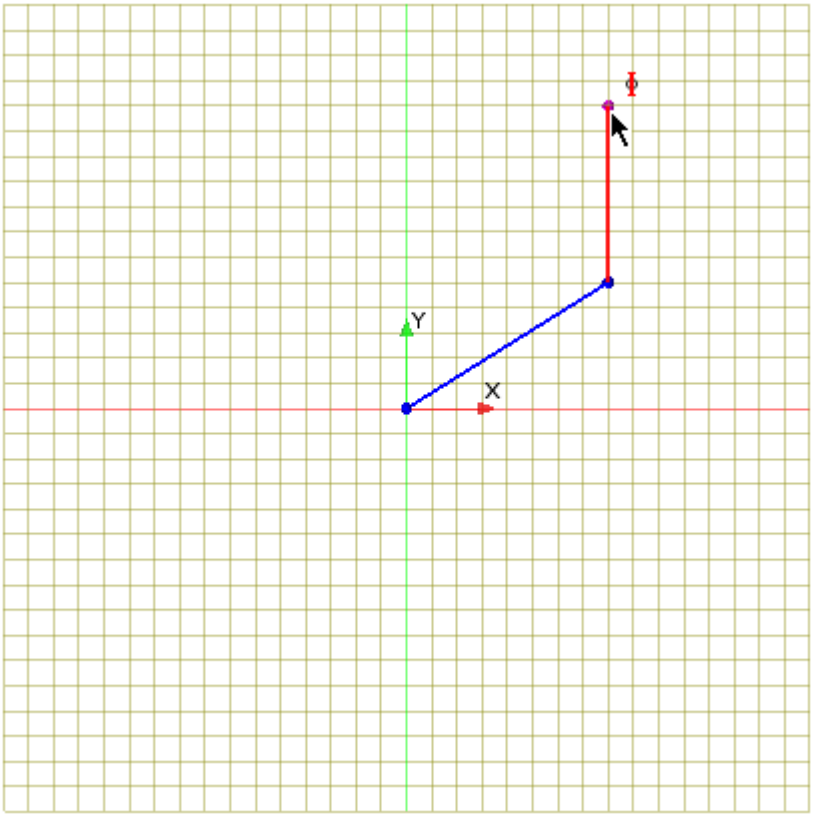


3. Use the tools in the *Display Options* box to control the appearance of the sketch environment in the *Graphics* window. See Sketch Display Options.
4. Use the tools in the *Create Sketch Entities* box to create the sketch. See Sketch Entity Reference.

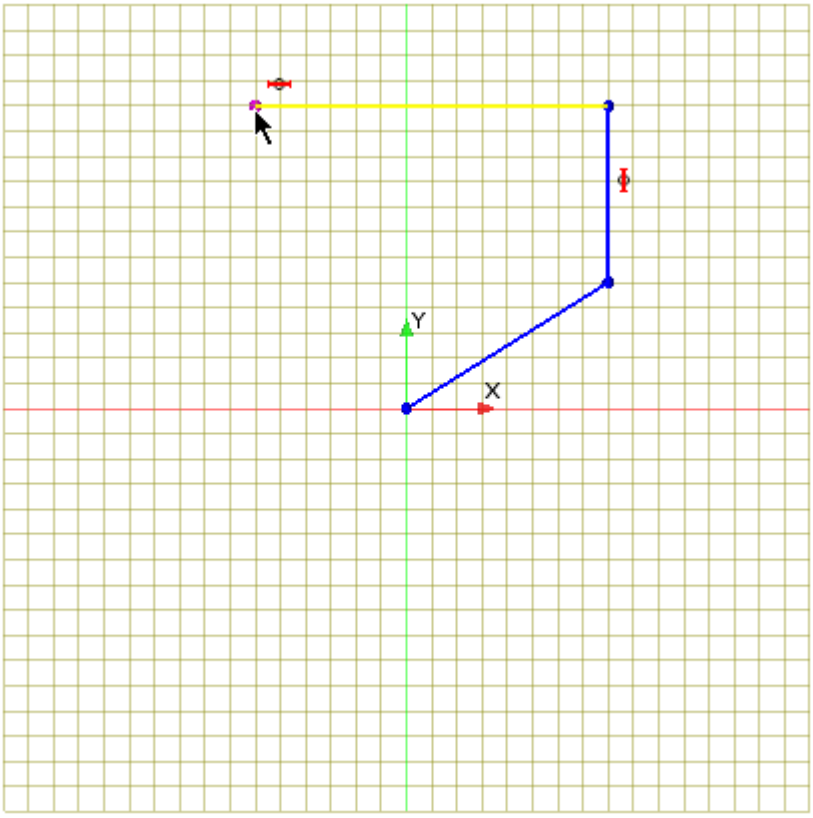
   As you draw sketch in 3D-CAD, the color of the entities change depending on their state. Active entities are colored in purple.



When you click to finalize an entity, the color changes to blue. 3D-CAD applies some constraints interactively as you sketch. For example, lines that you draw in the vertical direction appear in red and snap to this direction. When you click to draw the line in this position, a vertical constrain is added automatically.



Similarly, when you draw in the horizontal direction, the entity is displayed in yellow and a horizontal constraint is added.



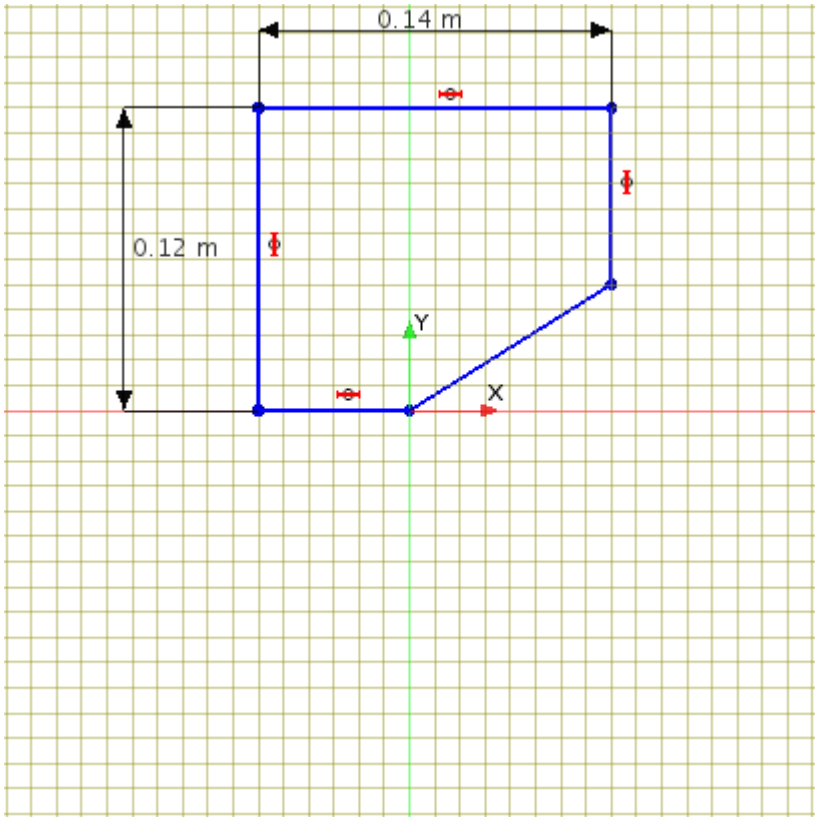
Other interactive constraints include tangent, parallel, coincident, and perpendicular.

5. Add further constraints that fix the position of sketch entities relative to each other, and to specific locations. See Constraining a Sketch.

   Some constraints require that you select multiple sketch entities before they appear in the context menu.

   For a full list of constrain types, see Constraint Types.

6. Dimension the sketch using constant sizes or parametric expressions. See Dimensioning Sketches.



7. When the sketch is complete, click **OK** in the *Sketch* panel.

A **Sketch** node is added to the feature tree.

Contents:

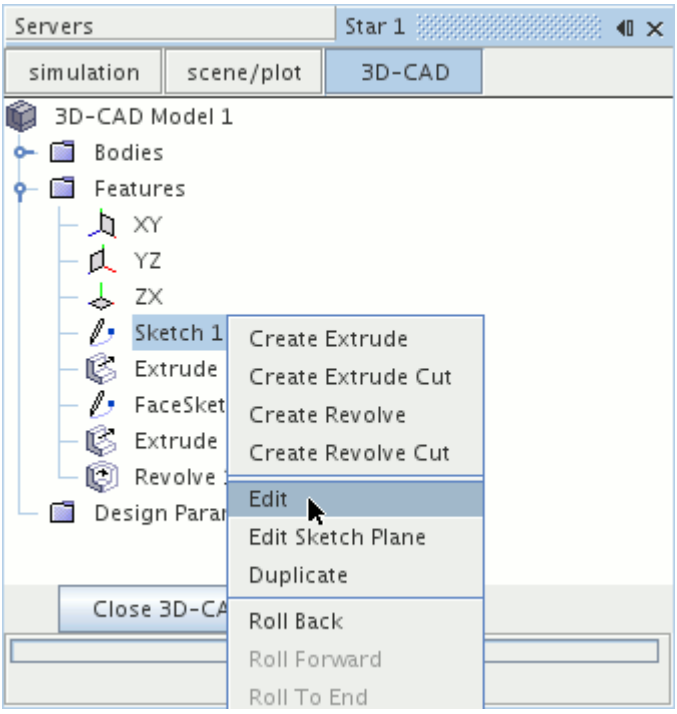
Editing a Sketch
Duplicating Sketches

Editing a Sketch

You can modify existing sketches by editing them. Changes that are made to a sketch are propagated to dependent features further down the feature tree.

To edit a sketch:

1. Right-click the corresponding sketch node and select **Edit**.



This action automatically rolls back the 3D-CAD model to the selected sketch and activates Sketch Mode.

2. Modify your sketch using the tools available in the *Sketch* panel. You can create, edit, and delete sketch entities.
3. Click **Apply** to confirm the changes and exit the sketch.

Note: If the changes cause any feature in the model to fail to rebuild properly, the feature state of affected features changes. To resolve the issue, update the model, or edit the affected features.

See Also:

Rolling Back

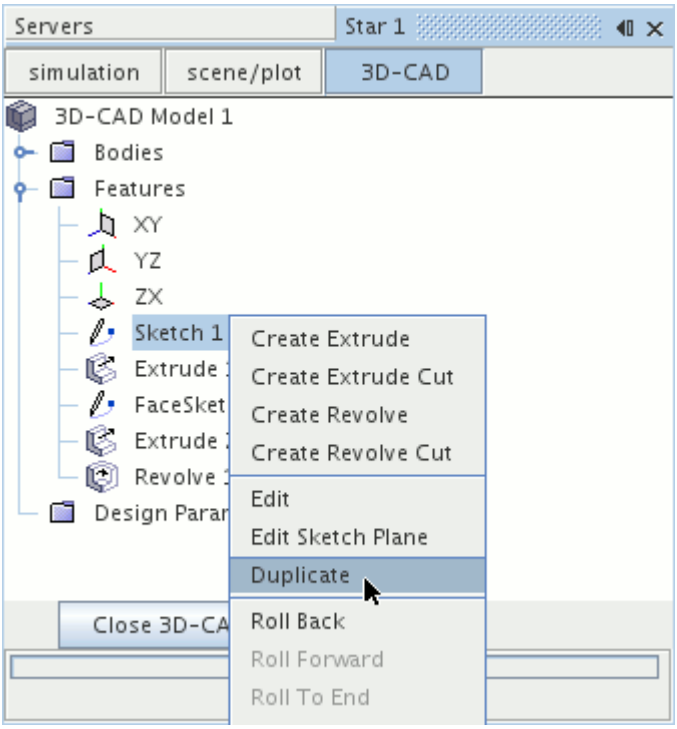
Understanding Feature States

Updating the 3D-CAD Model

Duplicating Sketches

You can make an identical copy of a sketch. To duplicate a sketch:

1. Right-click a **Sketch** node and select **Duplicate**.



An identical sketch feature is added to the feature tree. When you carry out this operation on a FaceSketch, an ordinary sketch feature is created.

Note: Duplicating a sketch also retains any design parameters that are associated with that sketch. Editing the design parameter affects both the original sketch and the duplicate sketch.

See Also:

Sketches

Working with Design Parameters

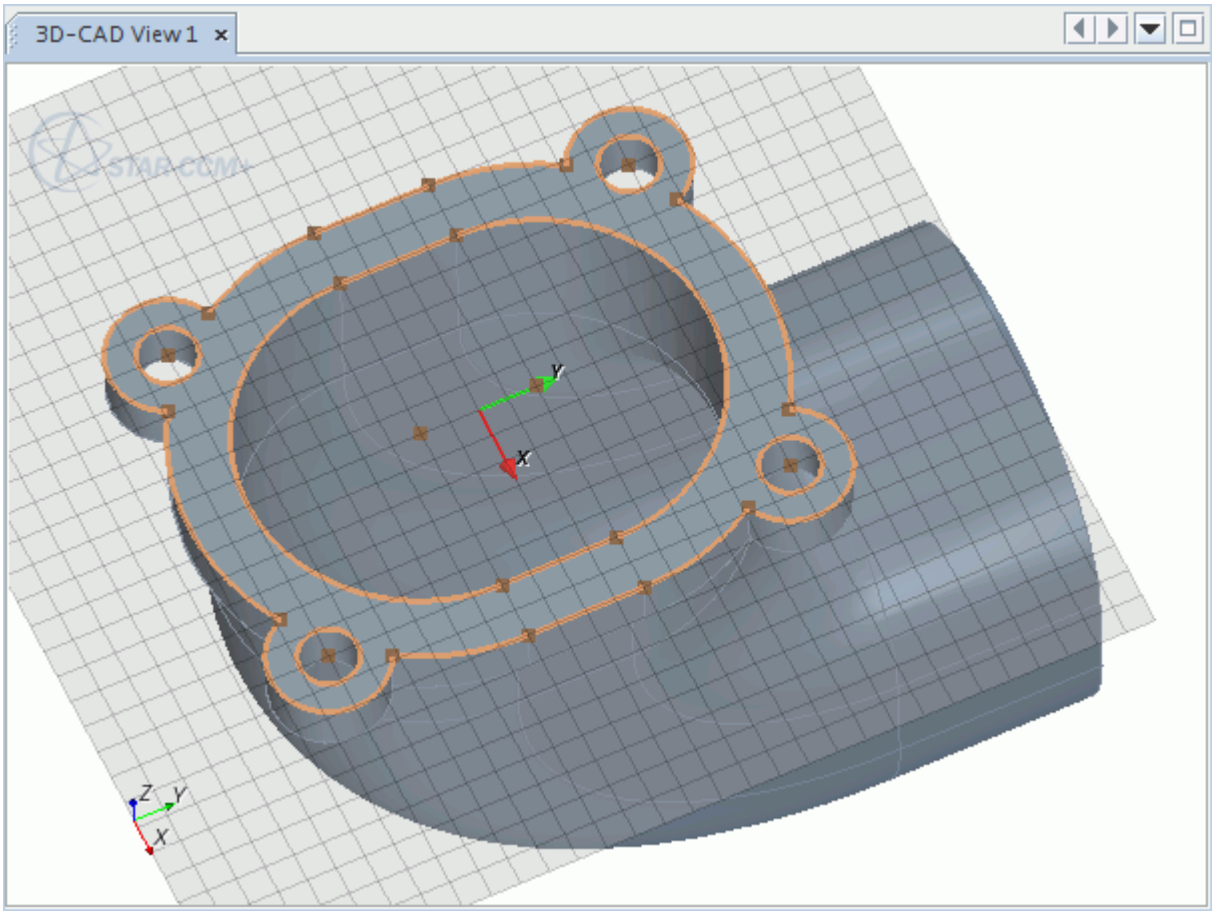
Creating a Sketch from Existing Edges

A face sketch is a sketch that comes from the edges of a planar face on existing bodies.

To create a face sketch:

1. Right-click on the target face in the *3D-CAD View* scene and select **Create Sketch** > **From Face Edges**.

STAR-CCM+ activates sketch mode, and adds the edges that define the face to the sketch.



2. If you intend to use the face edges for reference only, convert them to construction entities.
   - Multi-select the edges on the sketch, right-click, and choose **Set As Construction**.
3. Add other sketch entities as required.
4. When complete, click **OK** on the *Sketch* panel.

A **Sketch** node is added to the feature tree.

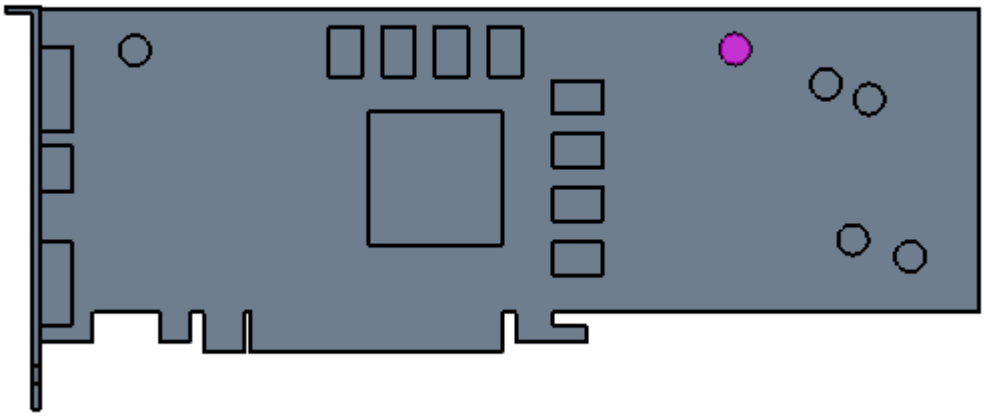
Creating a Sketch at the Center of a Face

You can choose to place a sketch at the center of a face on an existing body.

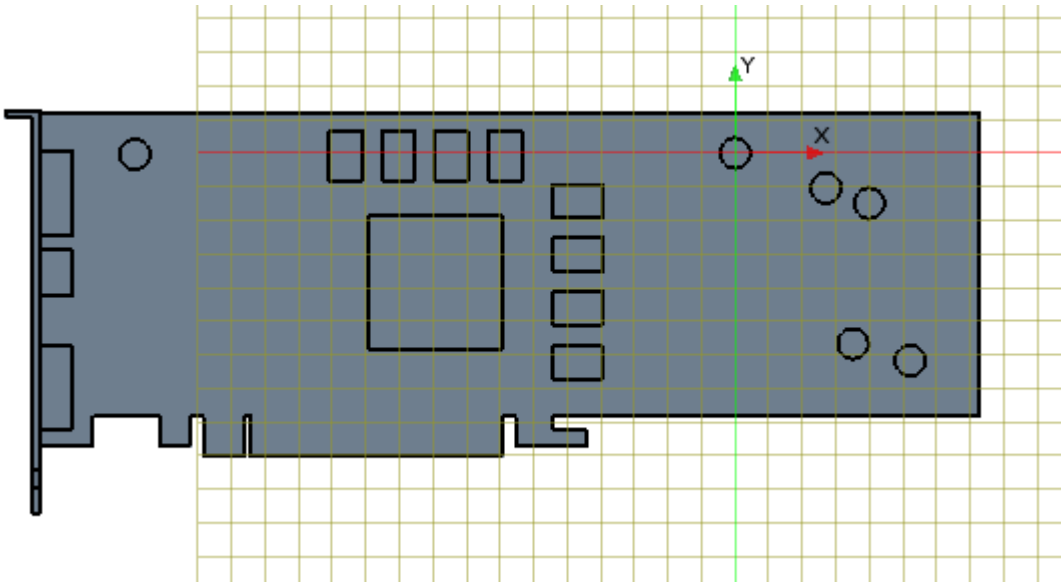
The center of the face is calculated and the origin is placed at this location.

To create a sketch at the center of a face:

1. In the *Graphics* window, right-click the desired face and select **Create Sketch** > **At Face Center**.



The sketch plane is placed on the selected face with the origin set to the center of the face.



2. Draw the sketch entities.
3. When complete, click **OK**.

A **Sketch** node is added to the feature tree.

Creating a Sketch in 3D

Use 3D sketching to create and modify more complex shapes in three dimensions.

In 2D sketching you are restricted to two dimensions on a plane; 3D sketching is not restricted to any plane. With 3D sketching you can create more advanced lofts with guide wires, more complex surfaces to fill gaps or

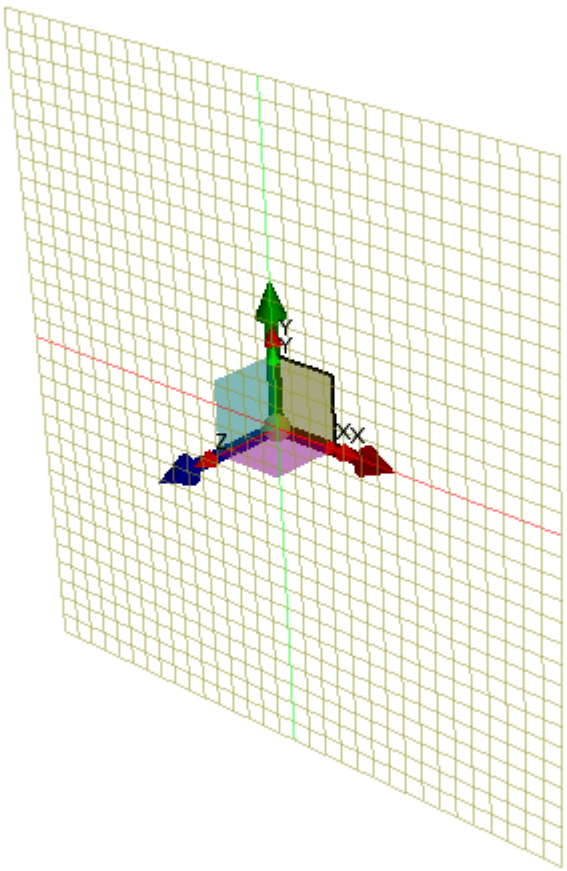
holes with more control, and allow for a broader range of parametric surfaces for design exploration. You can also use 3D sketching to edit imported 3D curves.

When sketching in 3D, you can interact with existing bodies, where you can convert any point or edge on a body to a 3D sketch entity simply by right-clicking and selecting the appropriate option.

To create a 3D sketch:

1. Enter the 3D sketching module:
   - Right-click the **Features** node and select **Create 3D Sketch**.
   - Right-click a CAD edge in the *3D-CAD View* scene and select **3D Sketch From Edges**.

   A grid is displayed in the *3D-CAD View* scene. At the center of the grid is a triad. The triad indicates the X, Y, and Z directions and allows you to manipulate sketch entities.

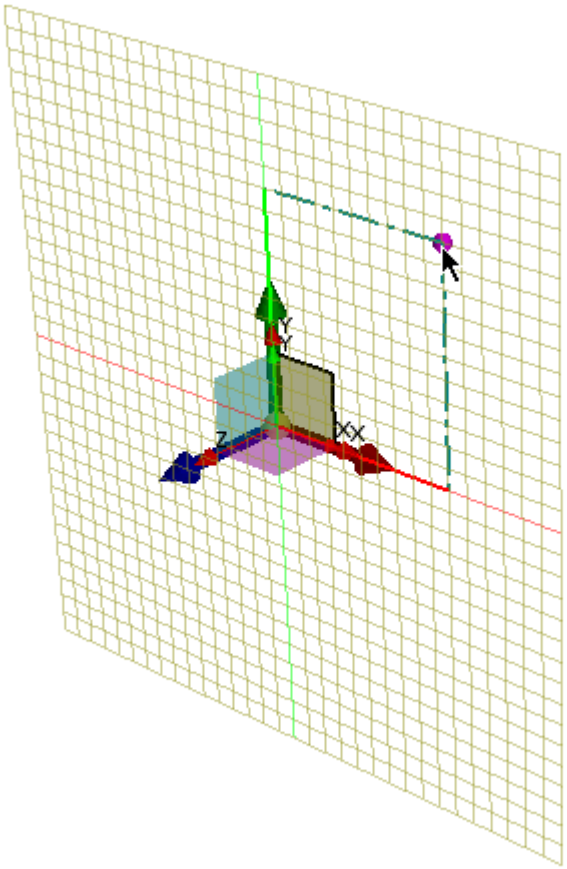


   The 3D sketch panel appears on the left hand side of the window. This panel contains the sketch tools and display options. For more information about the display options, see Sketch Display Options.

2. Select a 3D sketch tool and start drawing a sketch.

   The tools in the 3D sketch module behave in the same way as their 2D counterparts, with the exception being that in 3D sketching you can draw in all three dimensions. See 3D Sketch Entities.

When sketching in 3D, entities that you draw in the graphics window are constrained in the active plane, where the grid is displayed. As you move the mouse pointer around the scene, the two dimensions that define the active plane are projected to the mouse pointer.

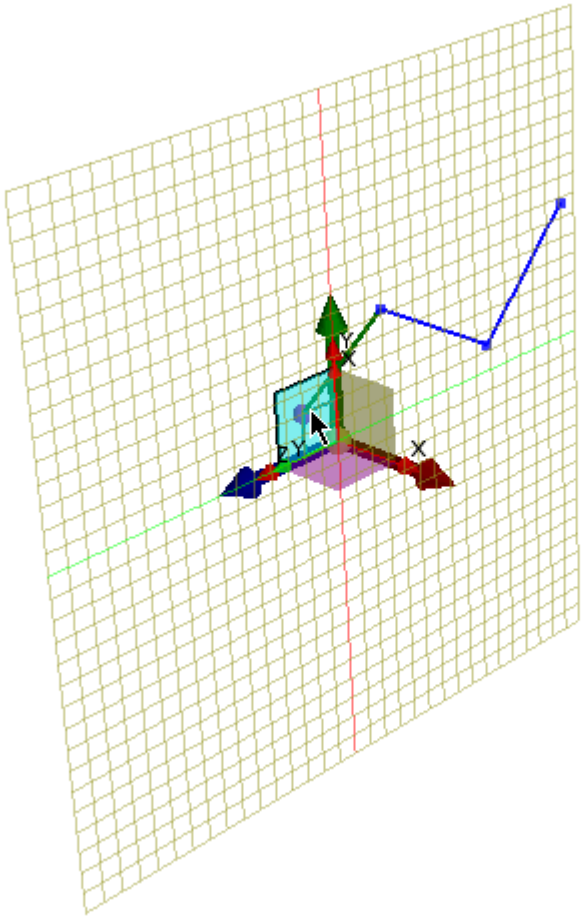


With the (**Line on Surface**) and (**Spline on Surface**) tools you can draw on the surfaces of a geometric object. When sketching on surfaces, deactivate the (**Pick on Plane**) option to limit the placement of points and sketches to geometry surfaces only.

You can switch planes to start drawing in a different direction. To switch plane:

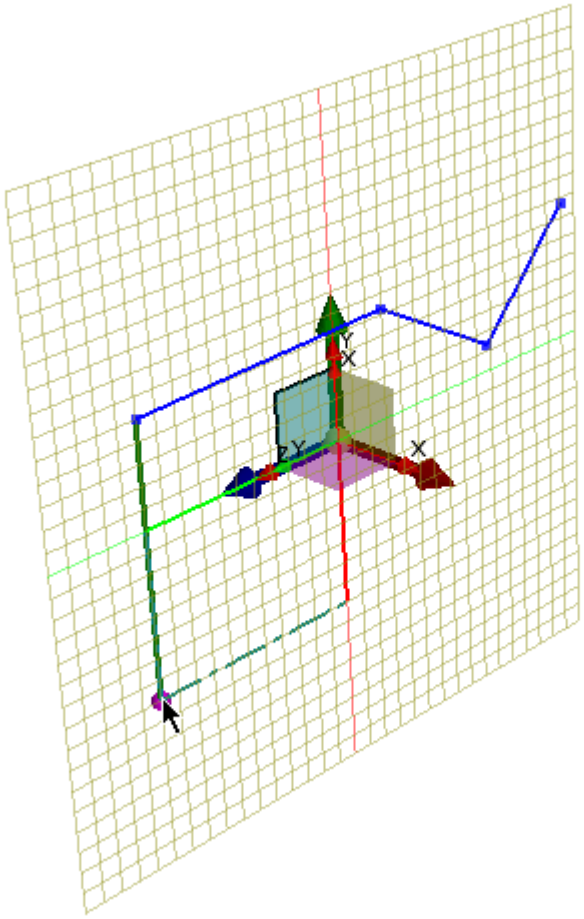
3. Click the desired plane on the triad to switch to this plane.

   The sketch grid switches to the active plane.



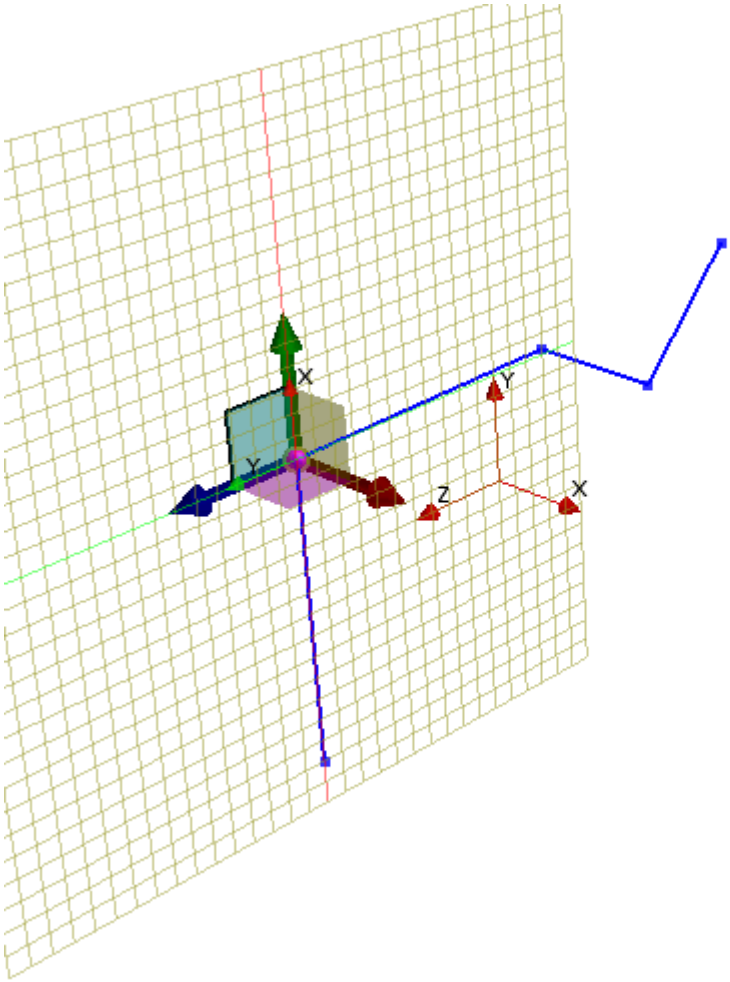
4. Continue drawing in the active plane.

   You can switch planes/direction as many times as needed.



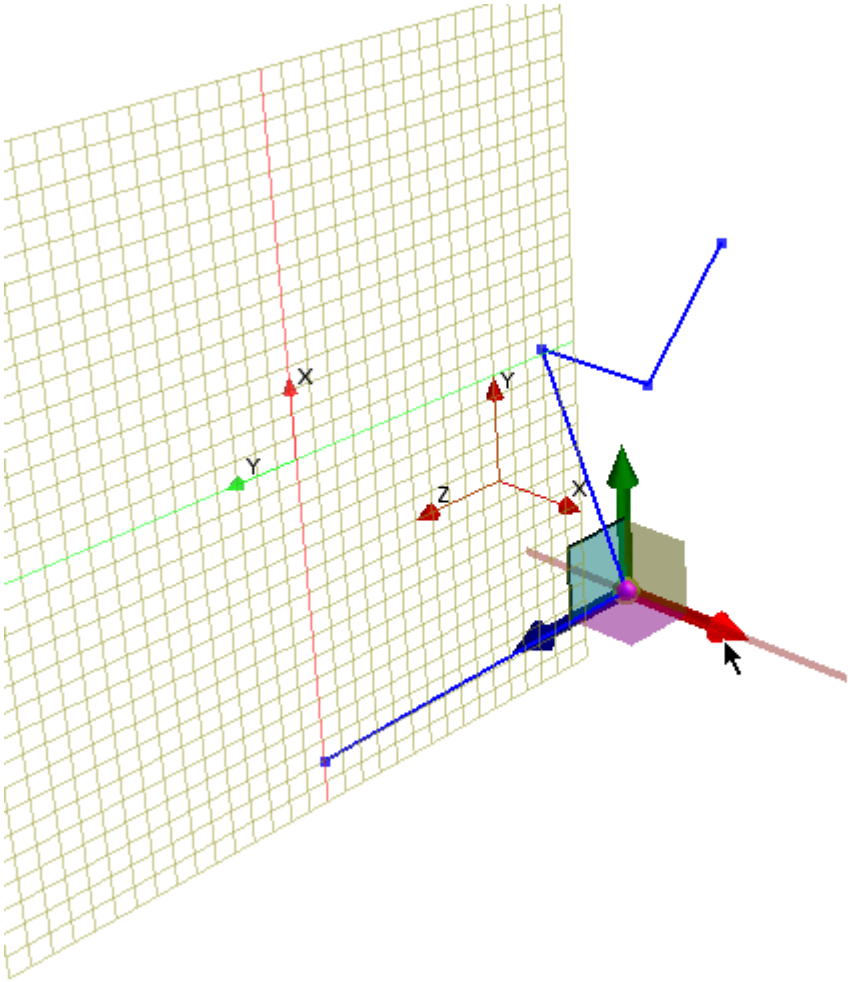
5. To stop drawing, press the <**Esc**> key.
6. To edit the position of a point in 3D space, click the point.

   The triad snaps to the selected point.

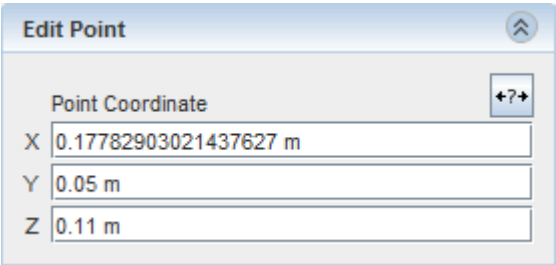


7. Move the mouse pointer over the desired triad direction arrow and click and drag the arrow.

   As you move the mouse pointer over each direction arrow on the triad, these become highlighted.



8. To move points more precisely, click the point in the graphics window and modify the point coordinates in the sketch panel.



You can interact with existing bodies while sketching in 3D. While drawing a 3D sketch, you can snap to points on existing bodies. You can also convert points and edges from the existing bodies into 3D sketch entities:

9. Right-click on a point or an edge on a body and select **Convert to 3D Entities**.

   The selections are converted to 3D sketch entities.

10. To complete the 3D sketch, click **OK**.

    A **3DCurve** feature is added to the 3D-CAD feature tree.

Trimming Sketches

The trim tool enables you to quickly trim segments of overlapping sketch entities.

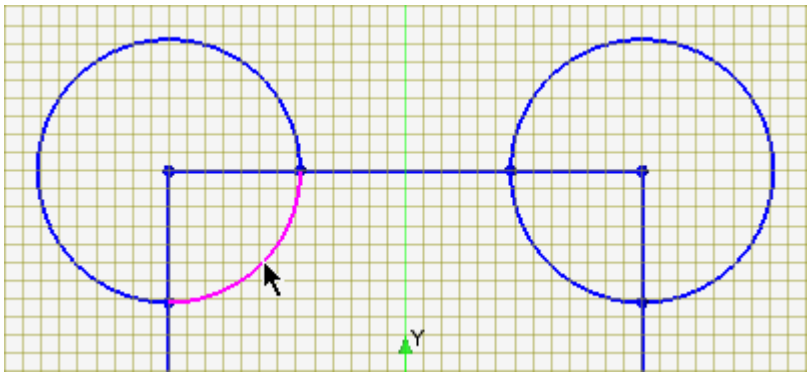
When you activate the trim tool, you enter quick trim mode. Actions that you perform while in this mode are separate to the actions you perform in sketch mode. While in quick trim mode, you can undo and redo actions.

To use the quick trim tool:

1. While in sketch mode, click (**Trim and Delete Primitives**).

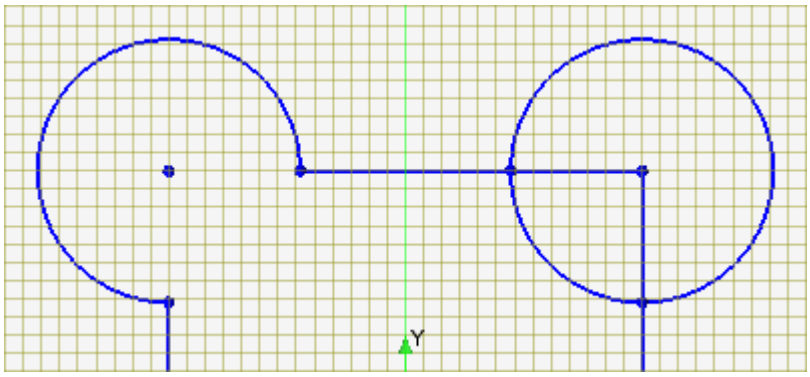
   The *Quick Trim* group box appears in the sketch tools panel and quick trim mode is activated. The quick trim tool identifies points where two sketch entities intersect and splits them at this point. All intersecting sketch entities in the sketch are split to form segments. You can select each segment independently.
2. Move the mouse pointer over the sketch in the *3D-CAD View* scene.

   As you move over the segments, these are highlighted in the *3D-CAD View* scene.



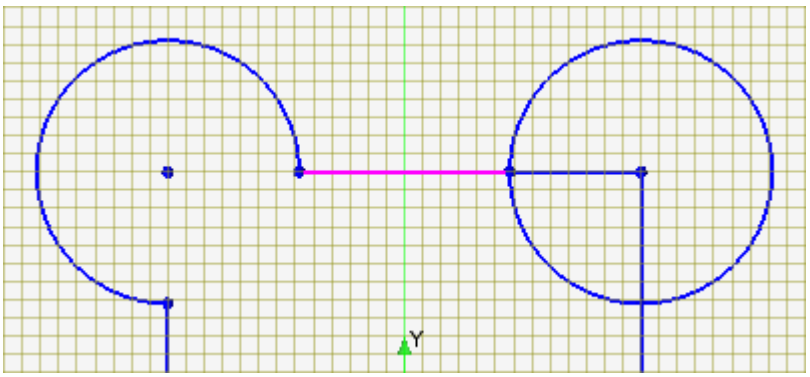
3. Click a segment to delete it.

   You can click and delete multiple segments within the active session. To undo an action, click the undo and redo buttons.



4. Click **OK** to complete the trim operation.

Quick trim mode closes and you are returned to sketch mode. On exiting quick trim, the sketch entities that are not involved with the trimming process revert back to their original state. However, sketch entities that you have trimmed maintain the segments/splits even after you exit the quick trim tool, as shown below.



Sketch Planes

A sketch plane is a flat surface on which you can draw a sketch.

The sketch plane definition comes from the coordinates of its origin and two direction vectors. The two direction vectors provide the local x and y axes on the plane.

There are three types of sketch plane:

Default Sketch Planes

Default sketch planes are labeled **XY**, **YZ**, and **XZ**. These three sketch planes are the starting point for every 3D-CAD model. The default sketch planes are predefined—you cannot create, edit, or delete them. Typically, the first step when creating a 3D-CAD model is to create a sketch on one of the default sketch planes.

Transform Sketch Planes

Transform sketch planes are sketch planes defined relative to an existing sketch plane or body face. Transformation parameters define the relation of the sketch plane to the reference sketch plane or face. These parameters are a translation vector and three angles of rotation.

See Creating a Transform Sketch Plane.

Reference Planes

Reference planes are custom sketch planes that you create. You can use any available local coordinate system to define this type of sketch plane.

See Working With Reference Geometry in 3D-CAD.

A 3D-CAD model can contain any number of transform sketch planes and reference planes. You can also export the coordinate system of a sketch plane in 3D-CAD for use in STAR-CCM+.

Contents:

Creating a Transform Sketch Plane
Changing the Sketch Plane of a Sketch
Exporting the Coordinate System from a Sketch Plane

See Also:

Sketches

Creating a Sketch

Creating a Transform Sketch Plane

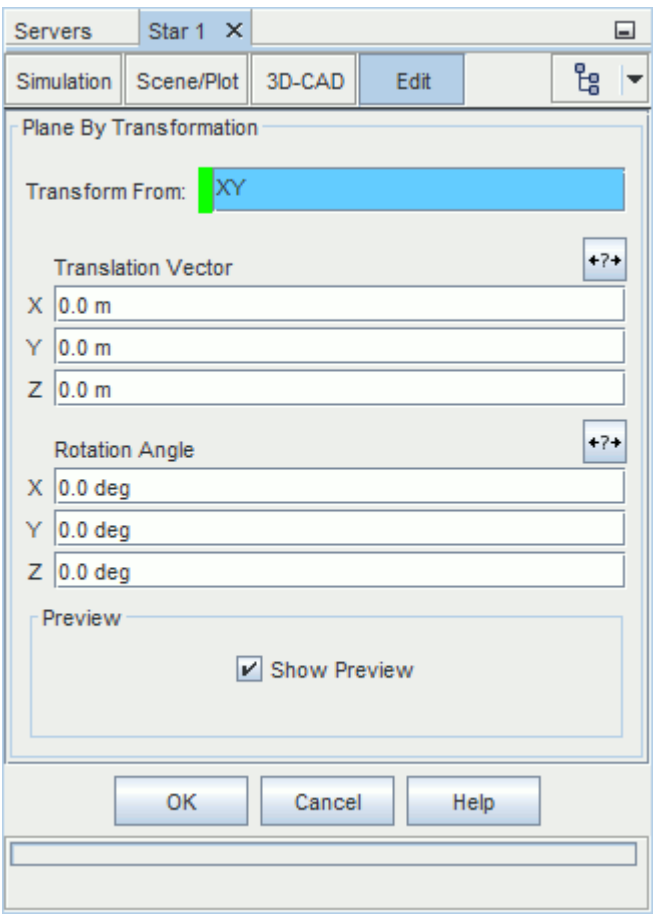
Exporting the Coordinate System from a Sketch Plane

Creating a Transform Sketch Plane

To create a transform sketch plane:

1. Select the base sketch plane using one of the following methods:
   - Right-click on one of the three default sketch plane nodes in the feature tree and select **Reference Geometry** > **Plane by Transformation**.
   - Right-click on an existing face of a geometry and select **Reference Geometry** > **Plane by Transformation**. A transform sketch plane that you create on a face with zero translation or rotation lies directly on that face.
   - Click **Create Plane By Transformation** in the 3D-CAD toolbar. If you do not pre-select a sketch plane or face when using the toolbar, select the desired sketch plane or face from the Select Sketch Plane or Face panel.

   The *Plane By Transformation* panel appears.



2. Enter the transform parameters. See Plane By Transformation Panel.
3. If required, create design parameters.

   A preview of the transform sketch plane appears in the *3D-CAD View* scene.
4. Click **OK**.

A transform sketch plane is created and a **Plane** node is added to the 3D-CAD feature tree.

Contents:

Plane By Transformation Panel

See Also:

Creating a New Design Parameter

Plane By Transformation Panel

You can create a transform sketch plane from another sketch plane or from a face on a geometry. Define the transform sketch plane using the following parameters:

Transform From	Specifies the sketch plane or planar surface to act as the source for the transformation.
Origin and X, Y Axes	Specifies the position of the origin. **Use Global Origin and Axes** sets the origin of the transform sketch plane to match the global origin in 3D-CAD. **Use Face Center and Face Orientation** sets the origin of the transform sketch at the center of the selected face. The orientation of the transform sketch plane also matches that of the selected face. **Use Face Center as Origin** sets the origin of the transform sketch at the center of the selected face. This option is only available when you select create a transform sketch plane from a face.
Translation Vector	Specifies the direction vector for the translation of the sketch plane.
Rotation Angle	Specifies the angle of rotation for the sketch plane. Note that if you define a Translation Vector and a Rotation Angle, the sketch plane is translated according to the translation vector before the angle of rotation is applied.

Changing the Sketch Plane of a Sketch

You can move sketches onto different sketch planes or faces to modify the position of its child features. For example, this facility permits you to relocate the position of an extruded body by associating its starting sketch with a different sketch plane.

To change the sketch plane on which a sketch is located:

1. Right-click on the relevant **Sketch** node in the feature tree and select **Edit Sketch Plane**.
2. In the *Edit Sketch Plane* panel:, select one of the following options:
   - To relocate the sketch onto a particular sketch plane, click **3D-CAD** and select the desired sketch plane from the feature tree. You can only select sketch planes that are higher up in the feature tree than the selected sketch. Sketch planes that are below the selected sketch are in a rolled-back state

and cannot be selected. If necessary, you can insert alternative sketch planes further up in the feature tree.

- To relocate the sketch onto a face, select the desired planar face in the *3D-CAD View* scene.

3. Click **OK**.

The sketch and any dependent features are relocated according to the settings that you applied in the panel.

See Also:

Sketch Planes

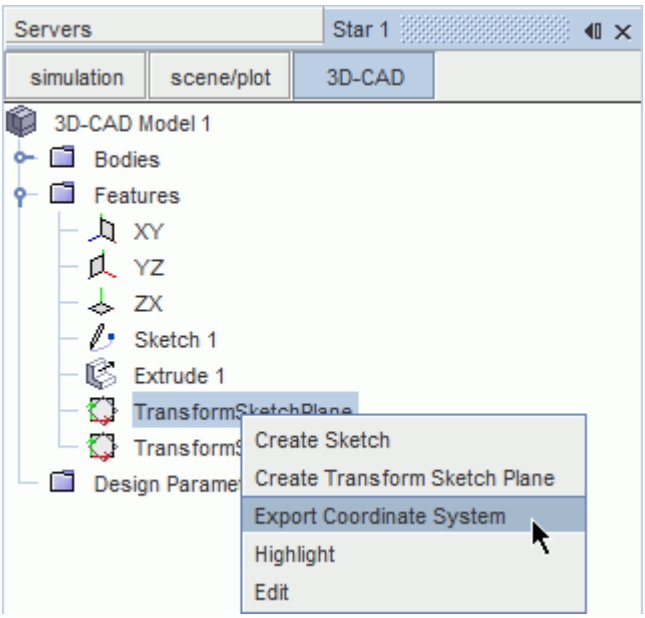
Understanding Where New Features Are Inserted

Exporting the Coordinate System from a Sketch Plane

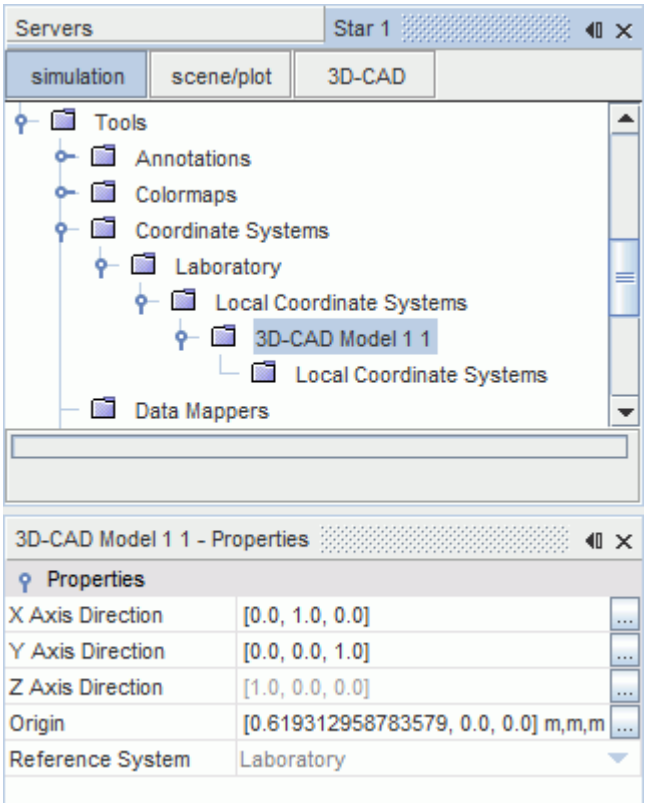
You can export the coordinate system of a sketch plane in 3D-CAD for use in STAR-CCM+ so that it becomes available as a local coordinate system under **Tools** > **Coordinate Systems**.

To export the coordinate system:

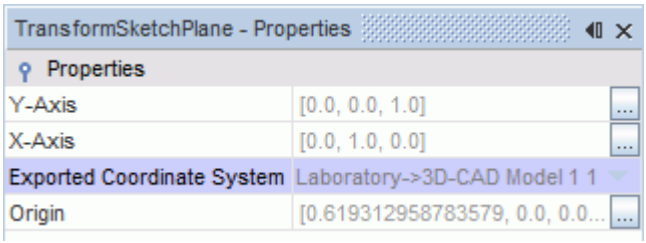
1. Right-click on a sketch plane in the 3D-CAD object tree and select **Export Coordinate System**.



A new local Cartesian coordinate system is created and added to the **Local Coordinate Systems** folder of the **Laboratory** coordinate system. This local coordinate system is linked to the coordinate system of the original sketch plane in 3D-CAD.



In the *Properties* window of the sketch plane, the *Exported Coordinate System* property confirms the successful export of the coordinate system.



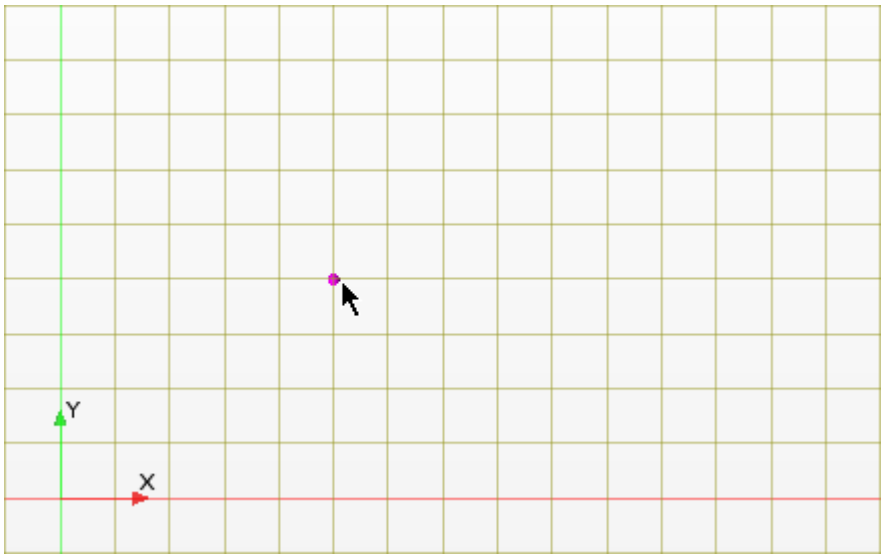
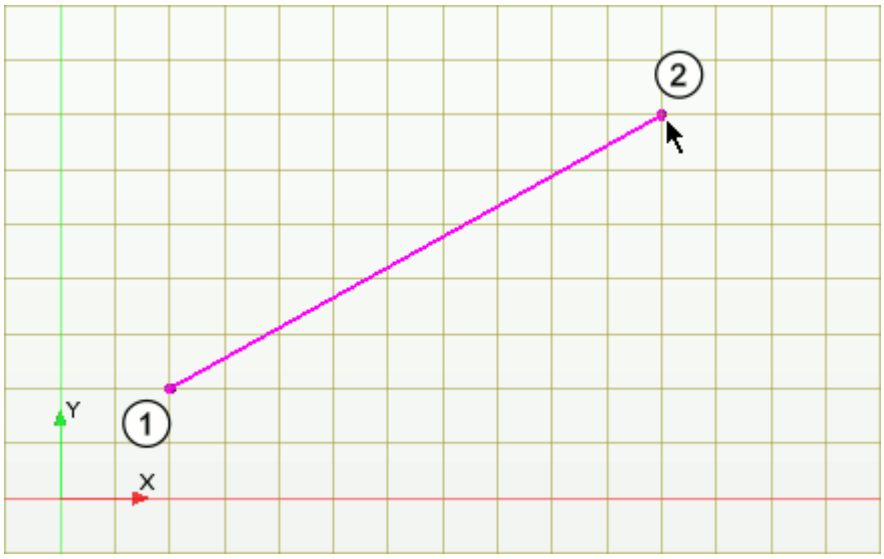
Caution: If you change the settings of the coordinate system in **Tools** > **Local Coordinate Systems** > **3D-CAD [model_name]** > **1**, these changes will be lost the next time that you update the 3D-CAD model. If you modify the sketch plane in 3D-CAD, however, the exported coordinate system is automatically updated with the changes.

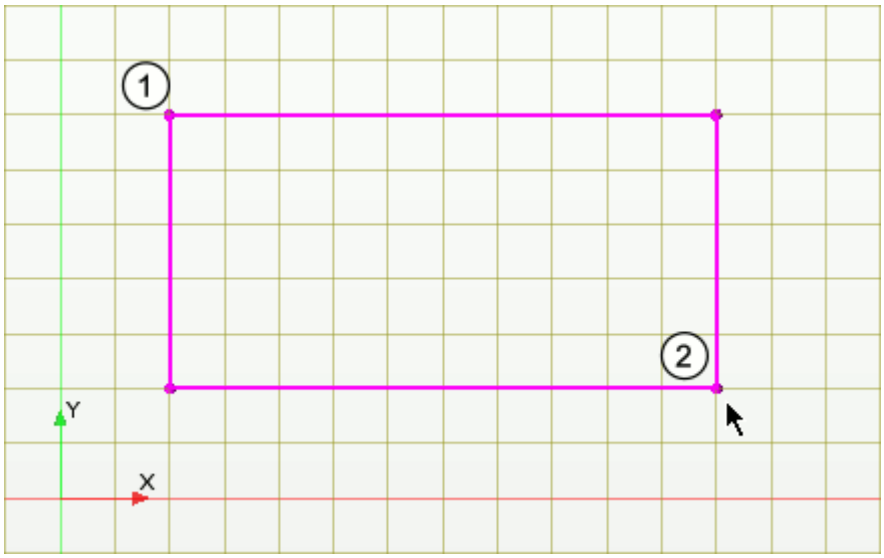
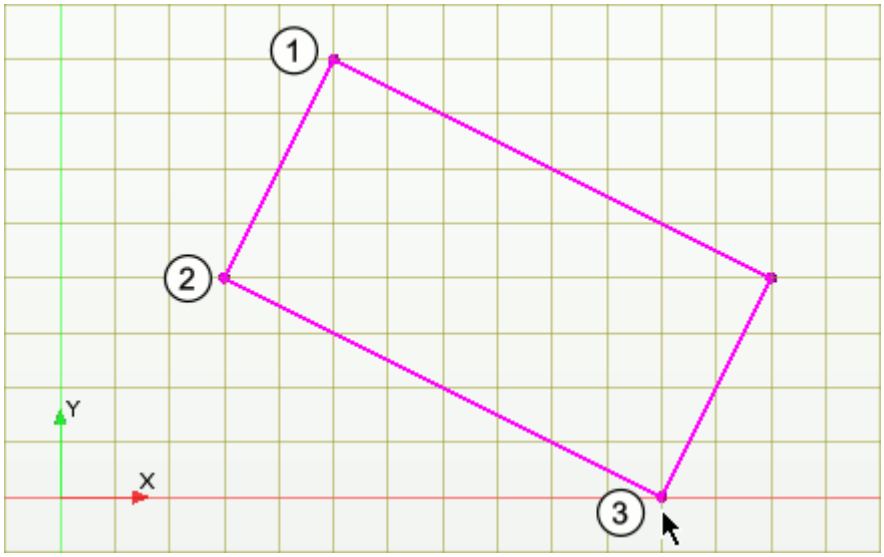
You cannot create Cylindrical or Spherical coordinate systems directly from 3D-CAD. Instead, create these coordinate systems starting from the exported local coordinate system in the **Tools** > **Coordinate Systems** objects tree.

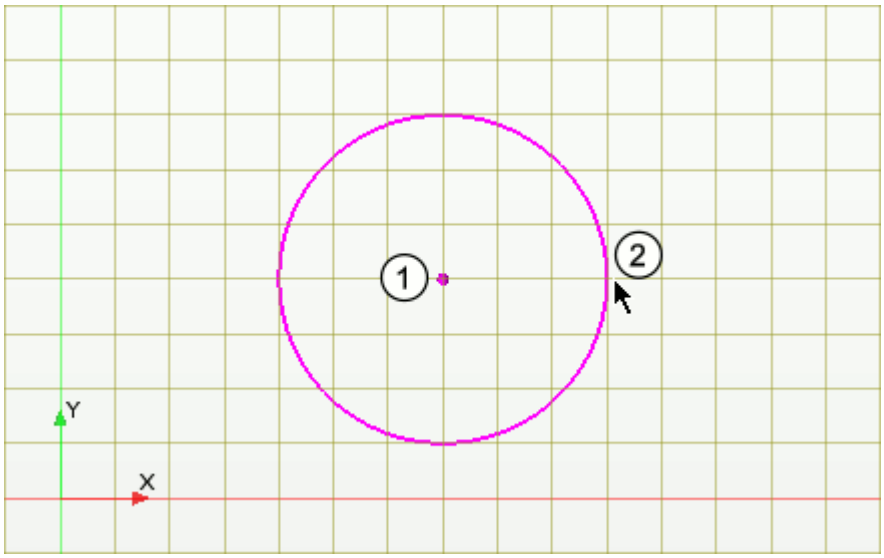
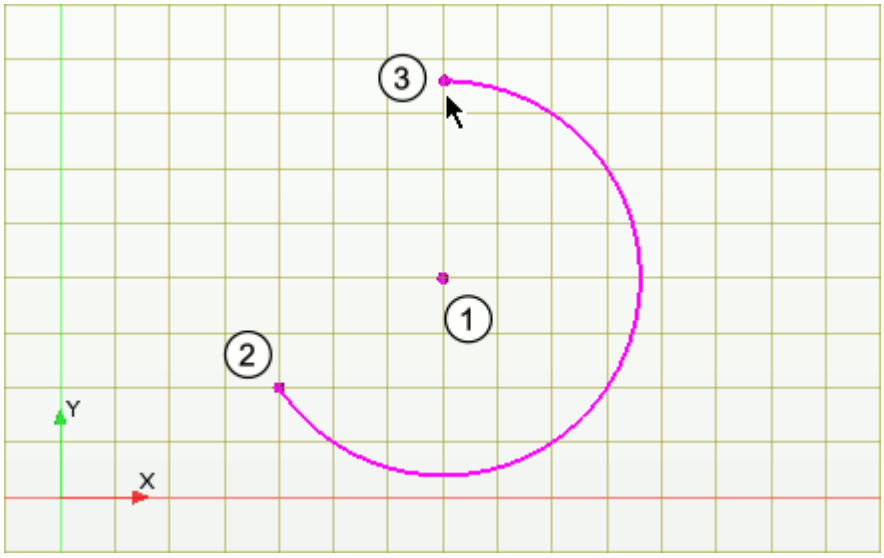
Sketch Entity Reference

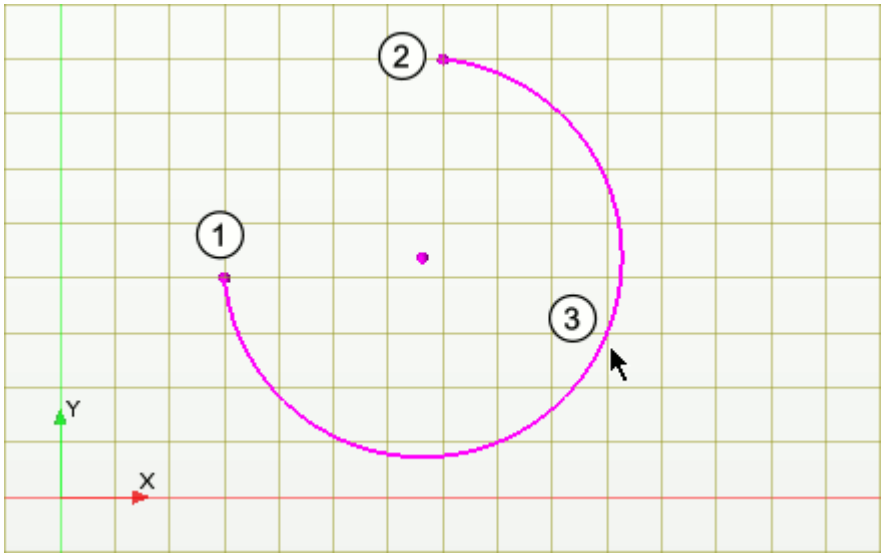
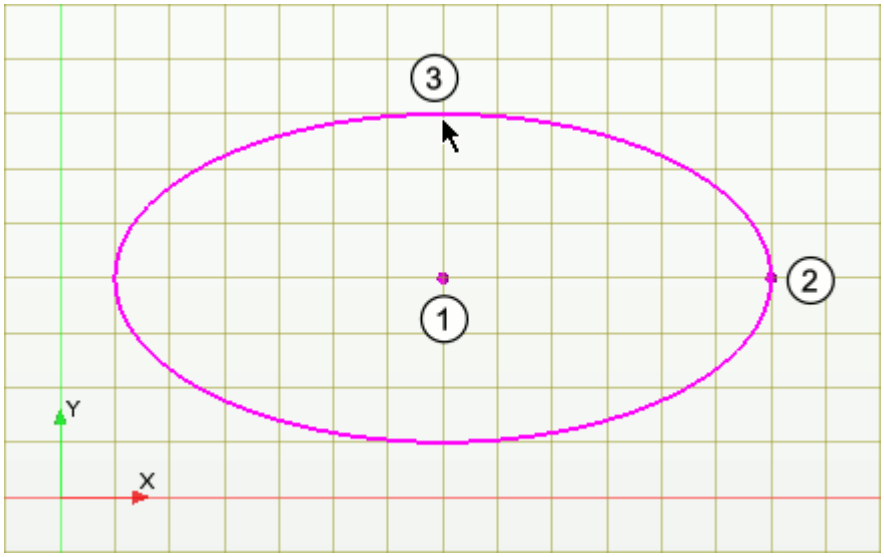
Sketches are comprised of sketch entities. The following table describes the sketch entities that are available in 3D-CAD.

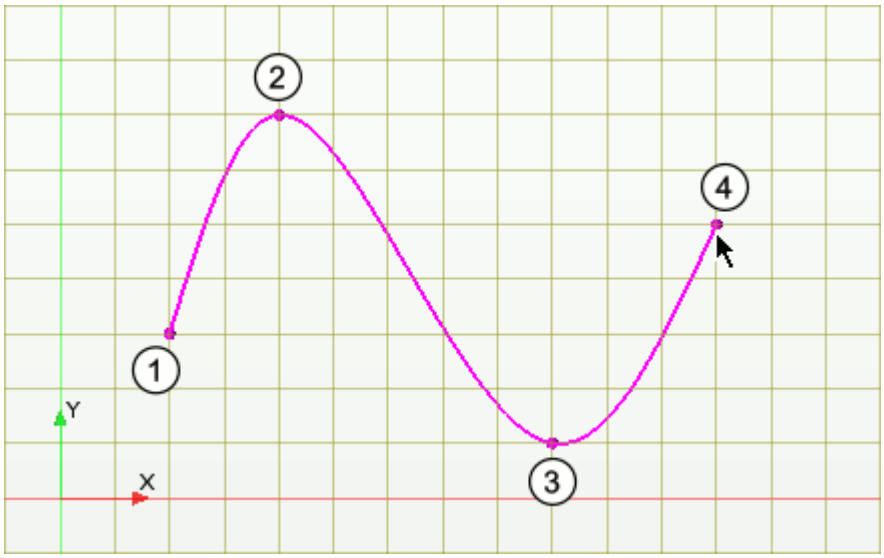
2D Sketch Entities

Sketch Entity	Description and Procedure
 (**Point**)	A point is a single vertex. To create a point, click the location where you want to place it. 
 (**Line**)	A line is a straight edge that is constructed from a start point and an end point. To create a line, you require two mouse clicks: the first defines the start point, the second defines the end point. To create multiple lines, continue to define points after defining the end point of the first line. Click <`Esc`> to exit the line tool 
	A rectangle is constrained to be aligned with the X and Y-axis directions on the sketch plane. To create a rectangle, you require two mouse clicks: one to define the location of the first corner and the second to define the opposite corner. Rectangles are

Sketch Entity	Description and Procedure
(**Rectangle**)	automatically constrained so that one set of opposite lines is horizontal, and the other is vertical. 
 (**Three-Point Rectangle**)	A three-point rectangle can be oriented in any direction. To create this type of rectangle, you require three clicks: the first two clicks set the length and orientation of the rectangle; the third click sets the width. The rectangle is automatically constrained with perpendicular and parallel constraints. 
 (**Circle**)	To define a circle, you require two mouse clicks: one to set the center of the circle, and the other to set the radius of the circle.

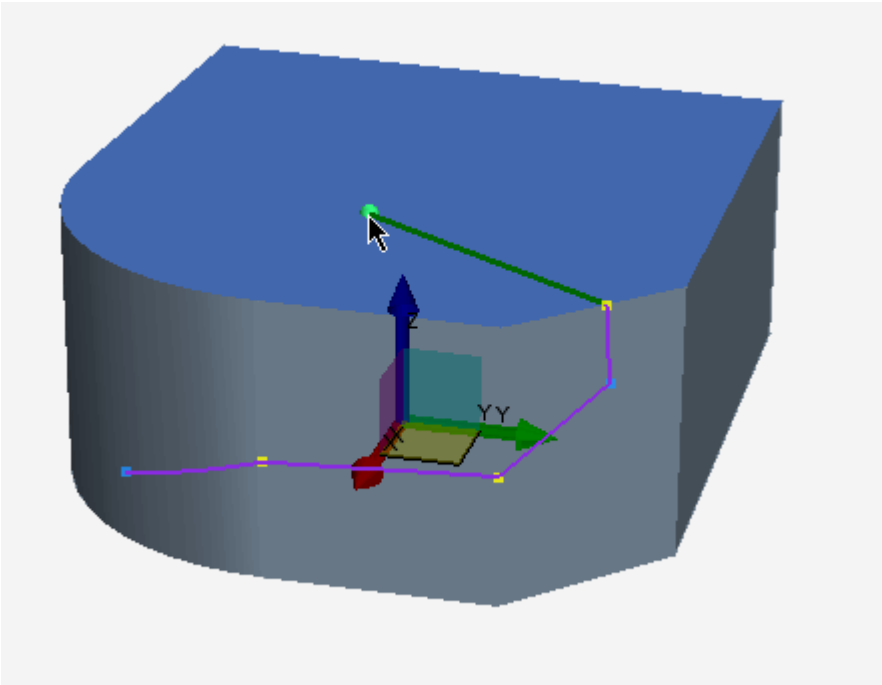
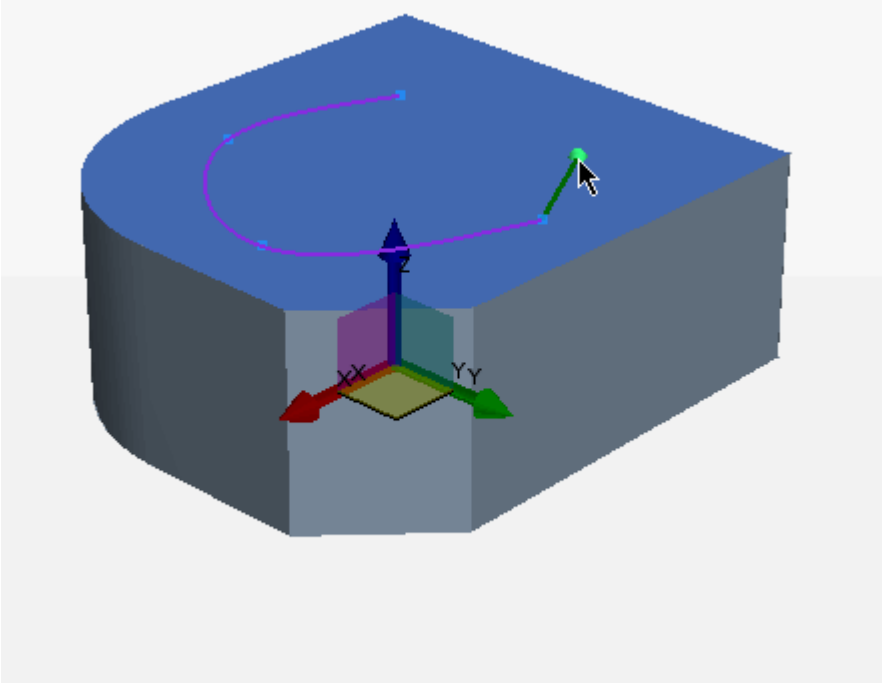
Sketch Entity	Description and Procedure
	
 (**Center-Point Circular Arc**)	A circular arc is a segment of a circle. To create a center-point circular arc, you require three mouse clicks: one to define the center of the arc, and two further clicks to set the start point and the end point. The arc is drawn anti-clockwise from the start point to the end point. STAR-CCM+ provides a function that converts and arc into its complement. This function is useful when the original arc is drawn in the opposite direction to the one that you require. See Converting an Arc Into its Complement. 
	A circular arc is a segment of a circle. To create a three-point circular arc, you require three mouse clicks: the first two clicks set the start and end points of the arc, and the third click sets the radius of the arc. STAR-CCM+ provides a function that converts and arc into its complement. This function is useful when the original arc is drawn in the opposite direction to the one that you require. See Converting an Arc Into its Complement.

Sketch Entity	Description and Procedure
(**Three-Point Circular Arc**)	
 (**Ellipse**)	An ellipse is a closed loop in the shape of an oval or flattened circle. To create an ellipse, you require three mouse clicks: one to set the center of the ellipse, followed by two further clicks to define the minor and major radii. You can create an elliptical arc by splitting and trimming an ellipse. It is possible to apply minor and major radius dimensions by right-clicking on the sketch and selecting Apply Minor Radius or Apply Major Radius. 
 (**Spline**)	A spline is a smooth curve that is composed from a series of points (called spline points). To create a spline, you require a minimum of three mouse clicks, which set the position of each spline point. You can add or remove points from the spline, as well as edit the position of each spline point individually. See Splines.

Sketch Entity	Description and Procedure
	

3D Sketch Entities

 (**Create Point**)	Draws a point in 3D space. See Point.
 (**Create Line**)	Draws a line in 3D space. See Line.
 (**Create Spline**)	Draws a spline in 3D space. See Spline.
 (**Create Line on Face**)	Draws a line that is constrained to a CAD surface. This tool is only available when sketching in 3D. To draw a line on a face, you require a minimum of two mouse clicks. The line can span multiple faces as long as there is a point on the adjoining edge. When you move the mouse pointer over a surface, this surface is highlighted in blue to indicate that the line will be constrained to it. When you move the mouse pointer over edges, these are highlighted in yellow.

	
 (**Create Spline on Face**)	Draws a spline that is constrained to a CAD surface. This tool is only available when sketching in 3D. To draw a spline on a face, you require a minimum of three mouse clicks. The spline is limited to a single face at a time. When you move the mouse pointer over a surface, this surface is highlighted in blue to indicate that the spline will be constrained to it. When you move the mouse pointer over edges, these are highlighted in yellow. 

See Also:

Constraint Types

Dimension Types

Sketch Display Options

This section describes how you can control the view in the *3D-CAD View* scene while you are working on a sketch.

(**Show Grid**)	Controls how the grid is displayed in the sketch plane. Click this icon to choose the behavior of the sketch grid in the *Grid Display Options* dialog. **Show bounded grid** displays the grid in a stationary box. **Show un-bounded grid** causes the grid to snap to the edges of the *3D-CAD View* scene when you zoom into the scene. **Hide grid** hides the grid from the scene. These options are stored in each sketch. When you edit or resume a sketch, these settings are maintained. When the grid is too dense (due to the zoom level or grid spacing settings), the grid becomes hidden.
(**Snap to Grid**)	Controls whether the mouse pointer snaps to intersections in the grid lines. This option is activated by default. When the grid is turned off, the mouse pointer still snaps to it. This option is stored in each sketch. When you edit or resume a sketch, these settings are maintained.
(**Show/Hide Relations**)	Controls whether the dimension markers, constraint glyphs, or the smart interaction features are displayed in the 2D sketch. See Showing/Hiding Relations. For details about smart interaction while sketching, see Creating a Sketch.
(**View Normal to Sketch Plane**)	Aligns the view so that it is normal to the sketch plane. The sketch plane appears automatically when in sketch mode. It displays the local axes of the sketch plane and the sketch plane origin.
(**Set Sketch Grid Spacing**)	Controls the distance between gridlines on the sketch plane. In the *Grid Spacing* dialog, enter a value (with units) for *Grid Spacing* to control the spacing of the major grid lines. *Number of Fine Grid Divisions* controls the number of divisions between each major grid line. Model dimensions in 3D-CAD must not exceed 10,000 m. Setting large values for grid spacing (>100 m) can result in errors when creating sketch entities.
(**Show Status of Sketch Primitives**)	Controls whether the status of the sketch primitives is shown in the sketch. See Viewing the Status of Sketch Primitives.
(**Show or Hide Triad**)	Controls whether the triad is displayed in the 3D sketch. The triad only applies when sketching in 3D and allows you to move selected sketch entities along the X, Y, and Z directions. See Creating Sketches in 3D.
(**Pick on Plane**)	Controls the placement of points and sketch entities on sketch planes. When deactivated, you cannot place sketch entities on sketch planes. This option only applies when sketching in 3D and is designed to help you drawn on surfaces

	using the **Line on Surface** and **Spline on Surface** 3D sketch tools. See 3D Sketch Entities.

Viewing the Status of Sketch Primitives

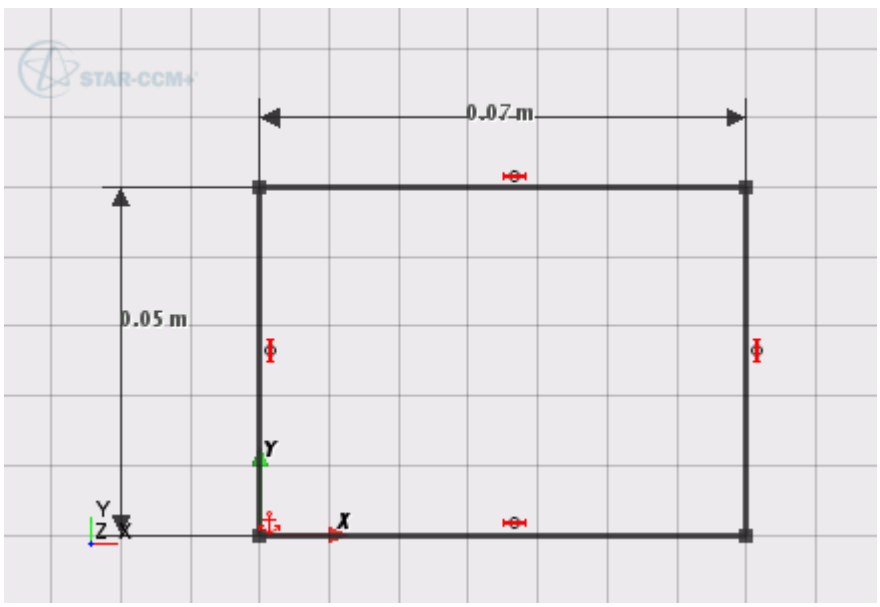
The status of sketch primitives indicate the validity of the sketch. This feature is useful for finding issues with your sketch, such as small gaps in sketches or overlapping primitives, which otherwise can be hard to see and would return errors further down the pipeline.

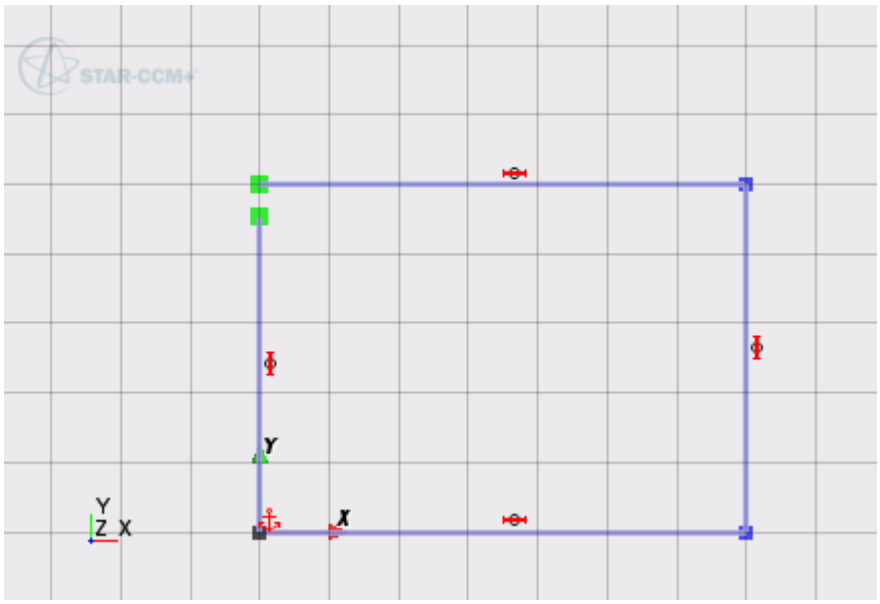
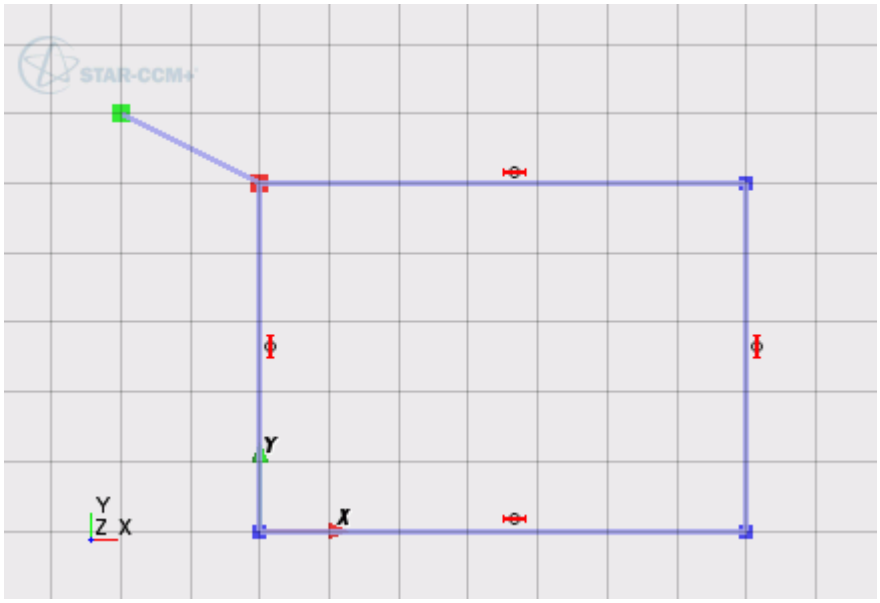
To display the status of sketch primitives:

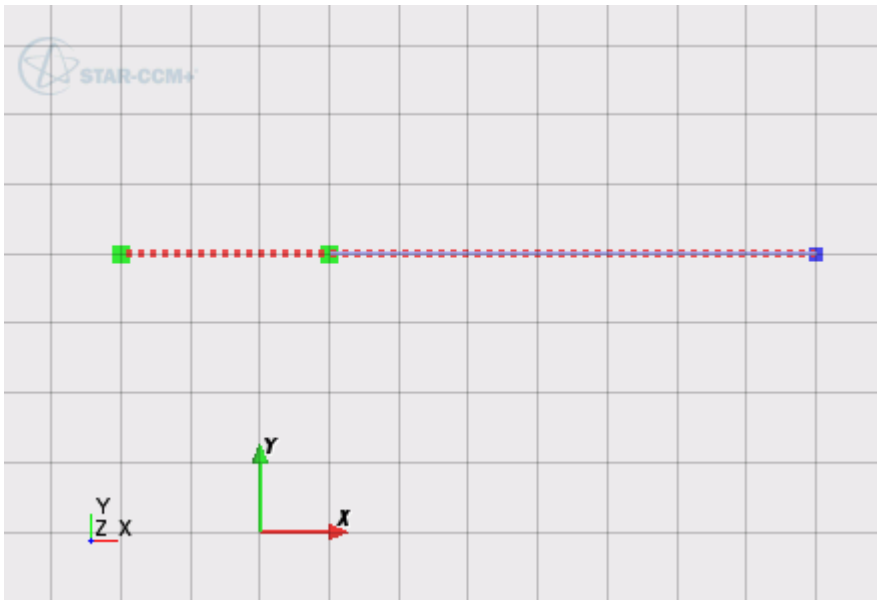
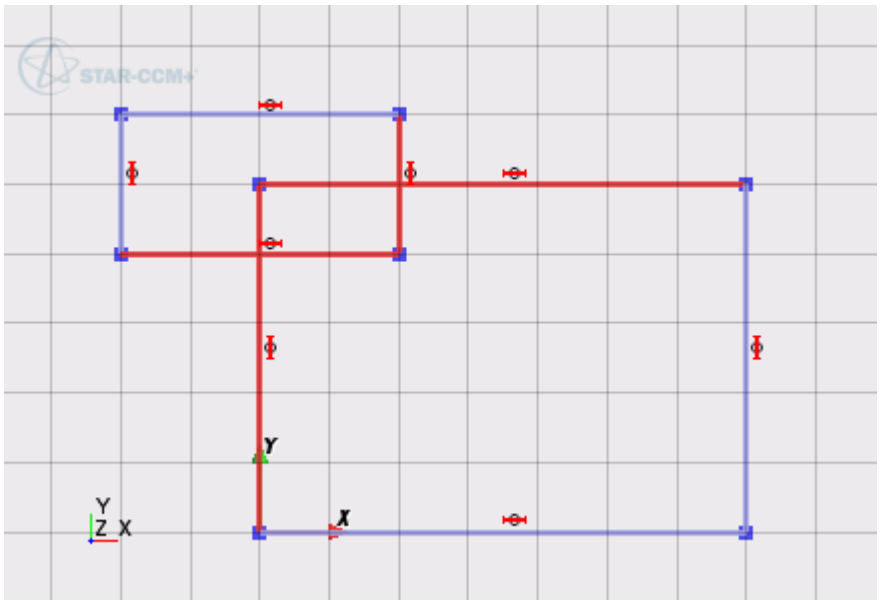
- In the *Sketch* panel, click (**Show Status of Sketch Primities**).

  The status of each primitive is displayed in the sketch scene. The status of the primitives is updated dynamically.

A list of colors/statuses and their corresponding meaning is displayed below:

Status	Description and Example
Black edges and vertices	Indicates that the sketch primitives are fully defined. 
Green vertices	Indicates the point at which the sketch profile is open.

Status	Description and Example
	
Red vertices	Indicates the point at which the vertices are non-manifold (shared by two or more sketch entities). 
Red dashed edges	Indicates that there are two or more overlapping sketch primitives. Only one of the sketch primitives is shown in a dashed red line.

Status	Description and Example
	
Red edges	Indicates intersecting primitives. 

Splines

A spline is a smooth curve that is composed from a series of points (called spline points).

In 3D-CAD, splines must be composed of at least two points. Typically most splines contain three or more points.

Each point on a spline has additional properties that define the angle and curvature of the spline passing through the point. You can modify these properties using splines handles that appear when you select a spline.

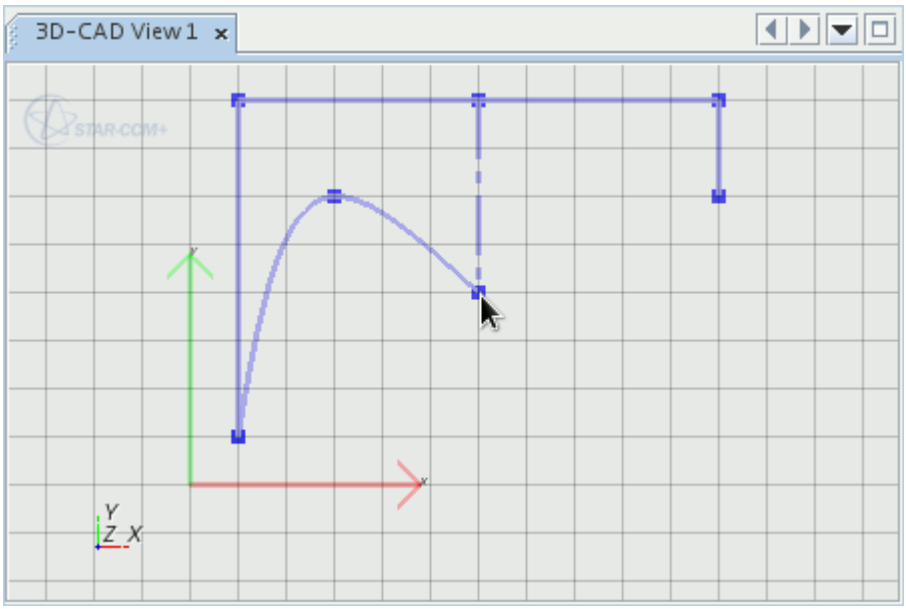
A list of topics that cover more advanced features of splines is provided below:

Contents:

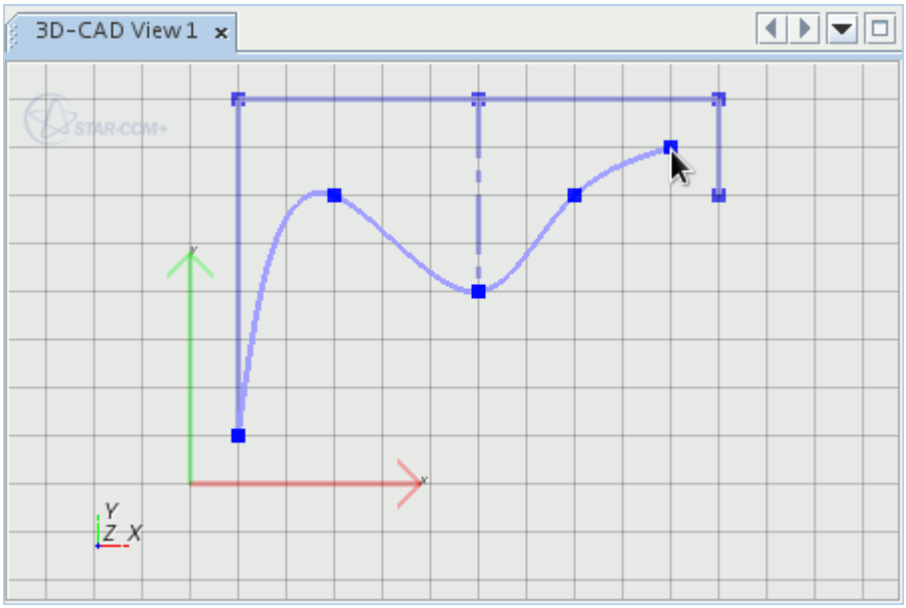
Placing Splines on Existing Points
Adding and Removing Points on a Spline
Controlling Spline Curvature

Placing Splines on Existing Points

When you place a spline point on an existing point in the sketch, 3D-CAD assumes that it is the end of the spline, and exits the spline creation tool.



If you want to carry on adding spline points after passing through an existing point, press and hold the <`Ctrl`> key before clicking the existing point. This action allows you to create splines that pass through existing points in the sketch.



The spline point is placed on top of the existing point. To connect the two points, apply a coincidence constraint.

Note: Do not split splines using the **Split for Trim** operation. Doing so modifies the spline curvature.

See Also:

Constraint Types

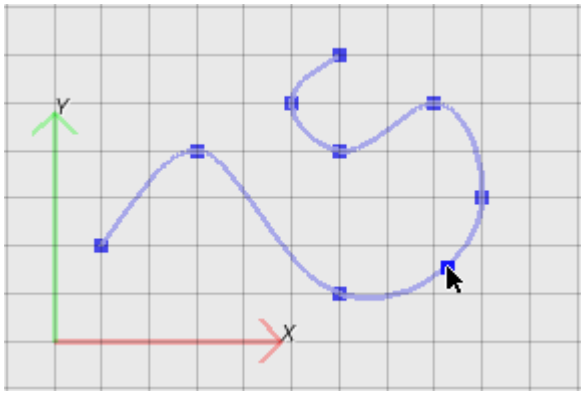
Splitting and Trimming Sketch Entities

Adding and Removing Points on a Spline

You can edit splines by selecting individual spline points and modifying them using the same technique as for points. You can also add and remove points from a spline.

To add a point to a spline:

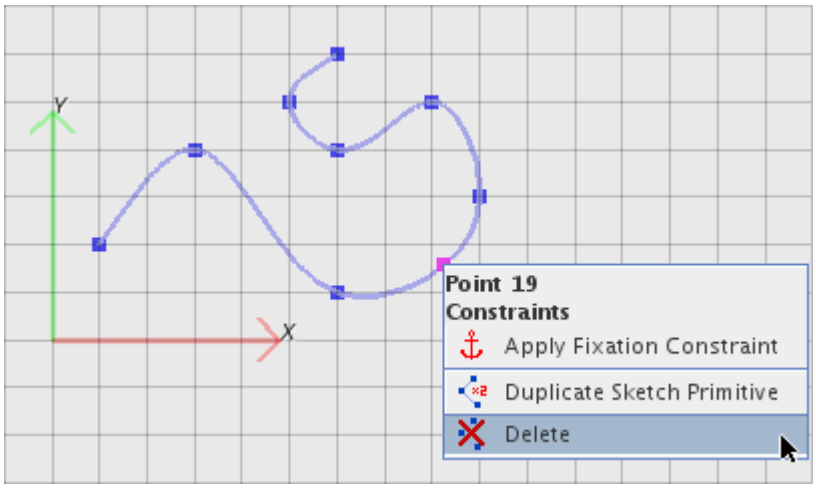
- Click the (**Create point**) button, and click on the desired location on the spline.



You can modify this new point in the same manner as any other point on the spline.

To remove a spline point:

- Right-click the point and select **Delete**, or select the point and press the <`Delete`> key.



The spline redraws based on the remaining spline points.

See Also:

Sketch Entity Reference

Controlling Spline Curvature

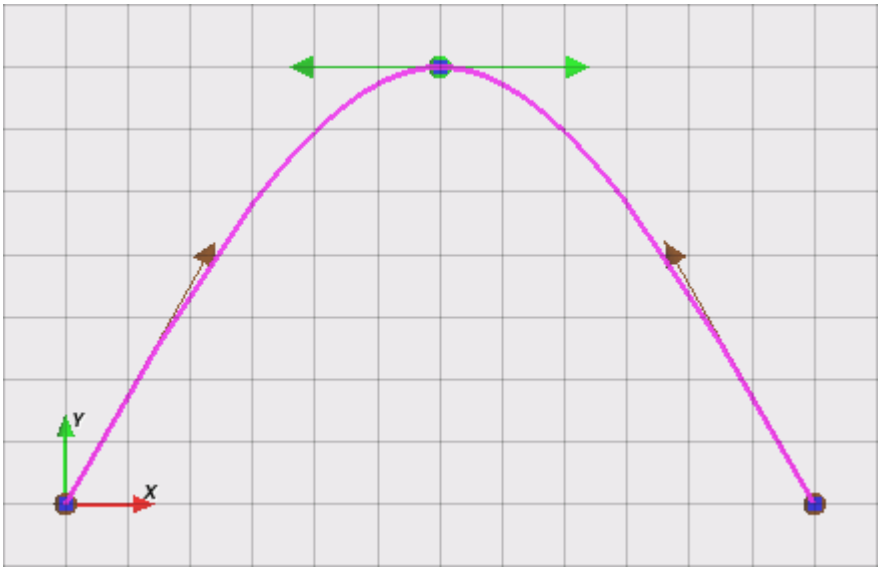
STAR-CCM+ provides more controls for the curvature of a spline passing through a point. These controls are:

- Angle: the angle of the spline, measured anti-clockwise from the x axis of the sketch plane.
- Magnitudes: parameters that control the curvature of a spline on either side of a point. The radius of the curve increases as the magnitude increases.

To modify spline curvature interactively:

1. Click the spline whose curvature you wish to modify.

Spline handles appear for each point on the spline. The curvature of the spline at a point is always tangential to the spline handle.



2. Click and drag the arrowheads of each spline handle as follows:
   - To change the amount of curvature in the direction of the arrow, extend or reduce the length of the arrow. Press `<Ctrl>` while moving the arrow to keep the current angle.
   - To change the angle of the spline, rotate the arrow around the spline point. Press `<Shift>` while rotating the arrow to keep the current length.
3. To complete the operation, click the sketch away from the spline.

You can also modify spline curvature using the *Spline Properties* panel that appears when you select a spline.

Note: Adding or deleting points to a spline with a modified curvature reverts the spline back to its original curvature. It is recommended that you add or remove points on a spline before editing its curvature.

Relaxing the Spline Handles

If the spline becomes twisted or contains sharp angles, you can smooth it by relaxing the spline handles. This feature is also useful for cases where the spline changes due to constraints.

To relax the spline handles:

1. Right-click the spline you want to smooth and select **Relax All Spline Tangent Handles**.

The spline is recalculated to fit smoothly through the existing spline points.

Sketch Entity Operations

There are two general methods that you can use to edit a sketch entity:

- Editing sketch entity properties in the relevant properties panel.

  See Editing Sketch Entity Properties.
- Moving sketch entities interactively in the *Graphics* window.

  See Editing Sketch Entities Interactively.

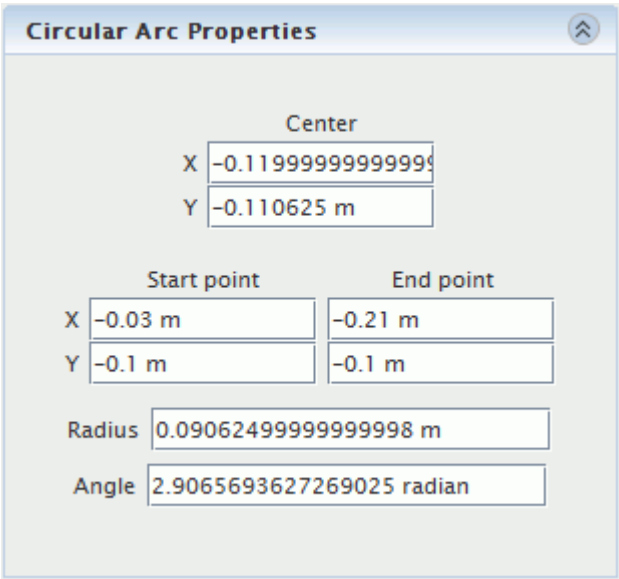
Further operations are available, as described in subsequent topics within this section.

Editing Sketch Entity Properties

To have more control over sketch entities, you can edit the sketch entity settings through the *Properties* panel.

To edit a sketch entity:

1. In the *3D-CAD view* scene, click the sketch entity that you want to modify.
2. In the *Properties* panel, enter the values that you want to use for the sketch entity. The properties differ for each sketch entity



The sketch entity updates with the new settings. All coordinates are defined in the local coordinate system on the sketch plane.

Editing Sketch Entities Interactively

You can modify sketch entities in the 3D-CAD view by interacting with the lines and points that make up the entity.

To edit an entity:

- In the *3D-CAD View* scene, click and drag an entity. Clicking different parts of an entity affects it in different ways:
  - Dragging a point, stretches the entities that are connected to the point.
  - Dragging a line, moves the entire entity and maintains its size and orientation. Neighboring entities stretch to maintain the connection to the moved entity.
  - Dragging the edge of a circle, arc, or ellipse, modifies its radius.
  - Dragging the center point of a circle, arc, or ellipse, moves the entire entity, maintaining the radius.

Contents:

Working with Construction Entities

Splitting and Trimming Sketch Entities
Creating An Elliptical Arc
Converting An Arc Into Its Complement
Offsetting Sketch Entities
Deleting and Duplicating Entities
Moving Sketch Entities

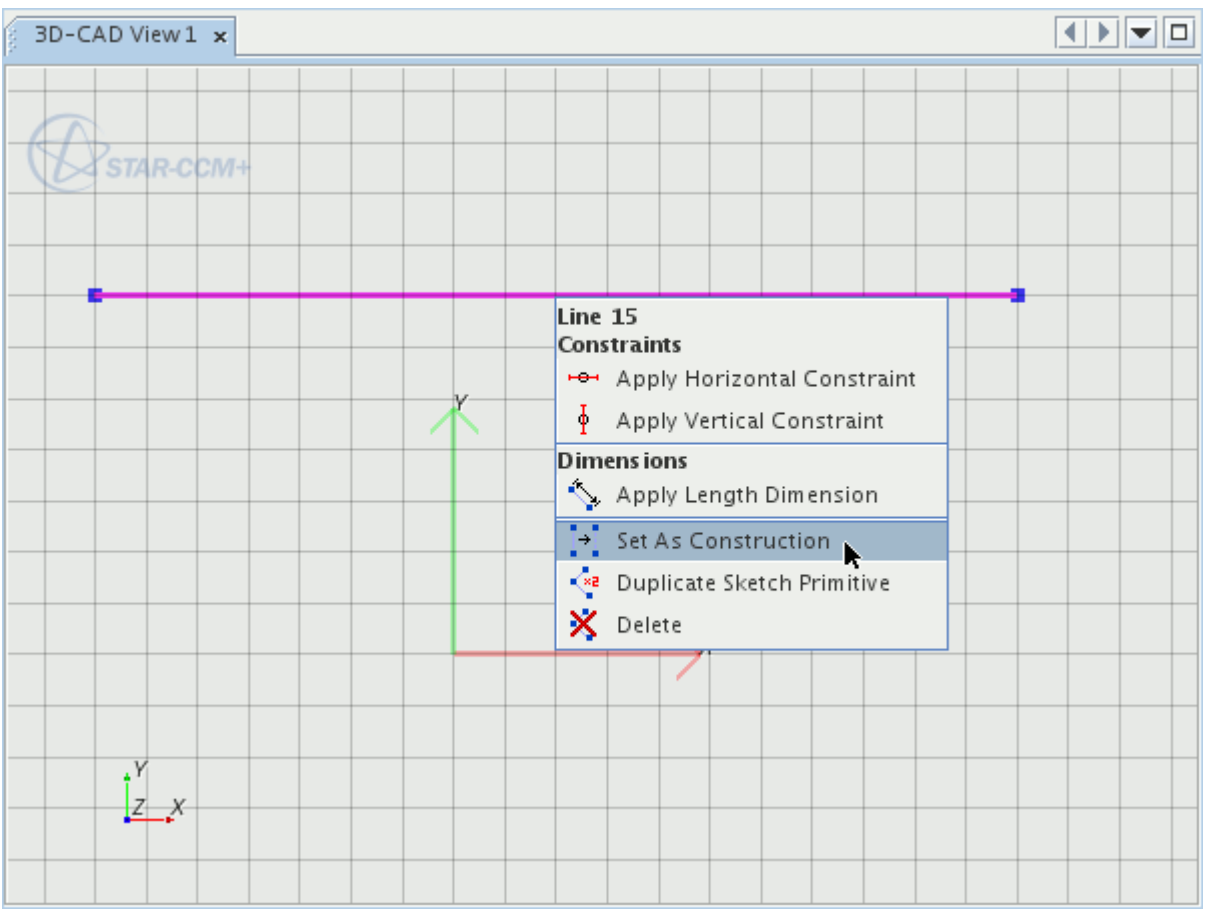
Working with Construction Entities

Construction entities are used within a sketch to aid the sketching process. 3D-CAD ignores construction entities unless the sketch is used to create a revolved feature.

When working with a sketch, the cursor snaps to a construction entity when it gets near to it. They are created by converting an existing sketch entity into a construction entity, and are only visible when in sketch mode. Construction entities are displayed as dashed lines within the sketch.

To create a construction entity:

1. Create a normal sketch entity.
2. Right-click on the sketch entity in the *3D-CAD View* scene and select **Set As Construction**.

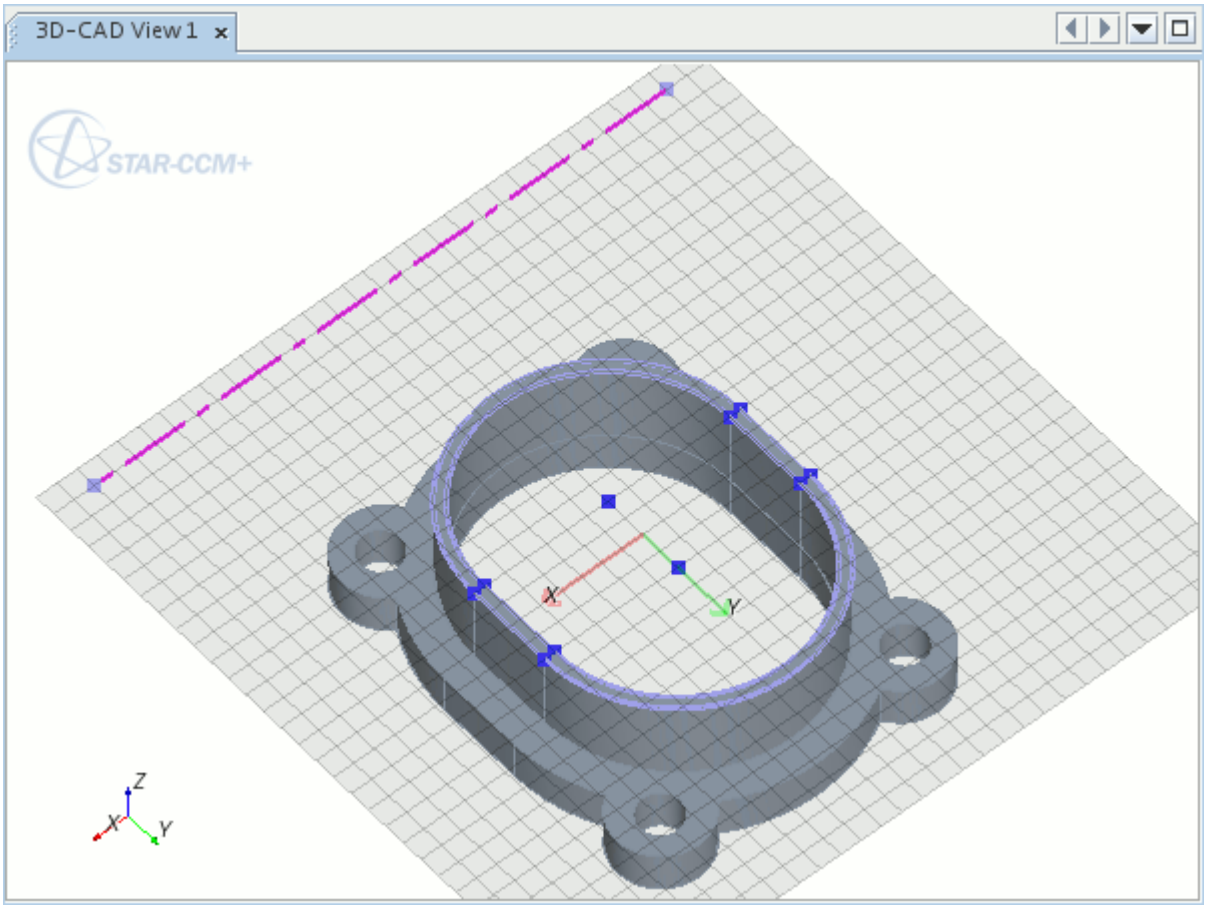


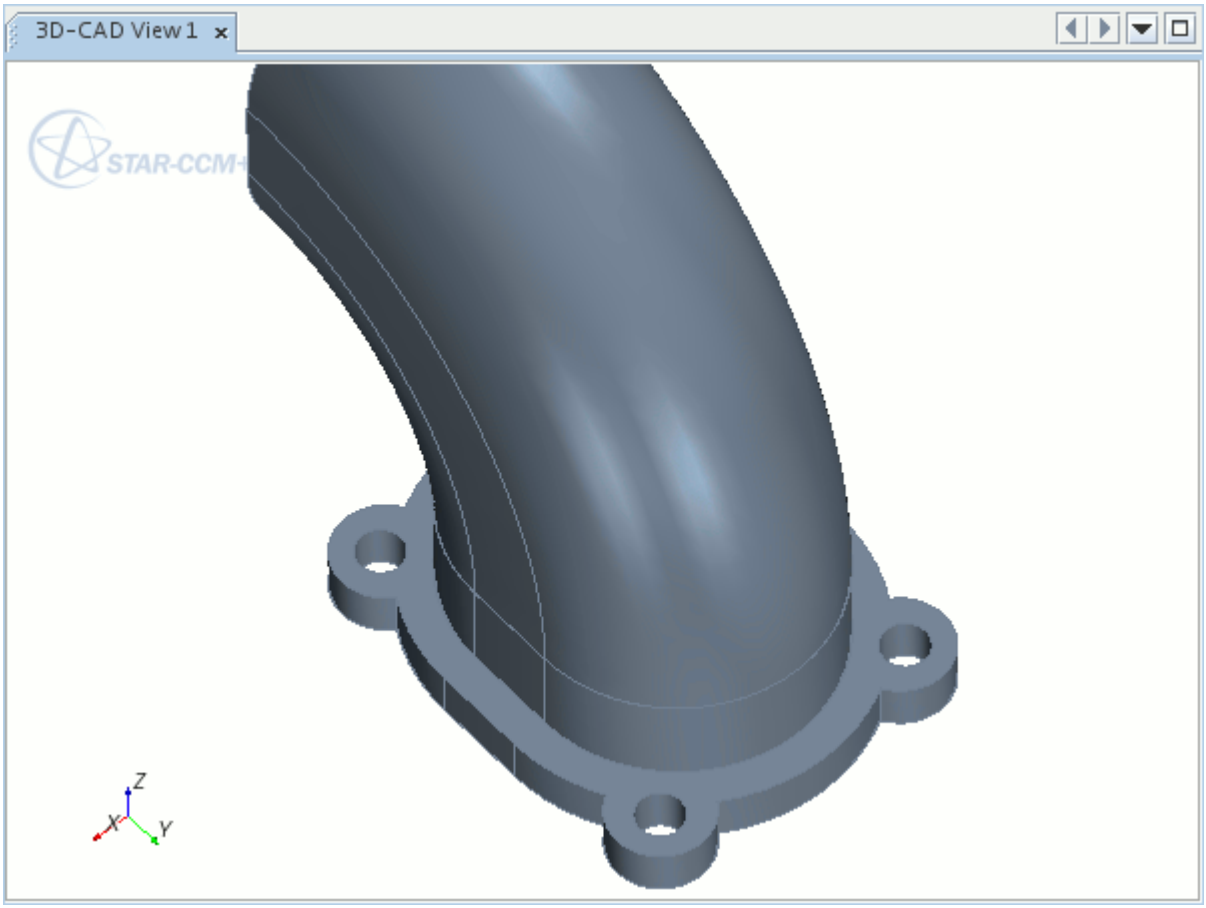
Note: You can select multiple sketch entities by holding down the <`Ctrl`> key.

The solid entity changes from a solid line to a dashed line, to show that it is now a construction entity.

Using Construction Entities for Revolved Features

When you use a sketch to create either a revolved or a revolved cut feature, it must contain at least one construction line to define the axis about which the sketch revolves.


3D-CAD View 1
STAR-CCM+
X
Y
Z
X
Y



See Also:

Sketches

Sketch Entity Reference

Revolved Features

Splitting and Trimming Sketch Entities

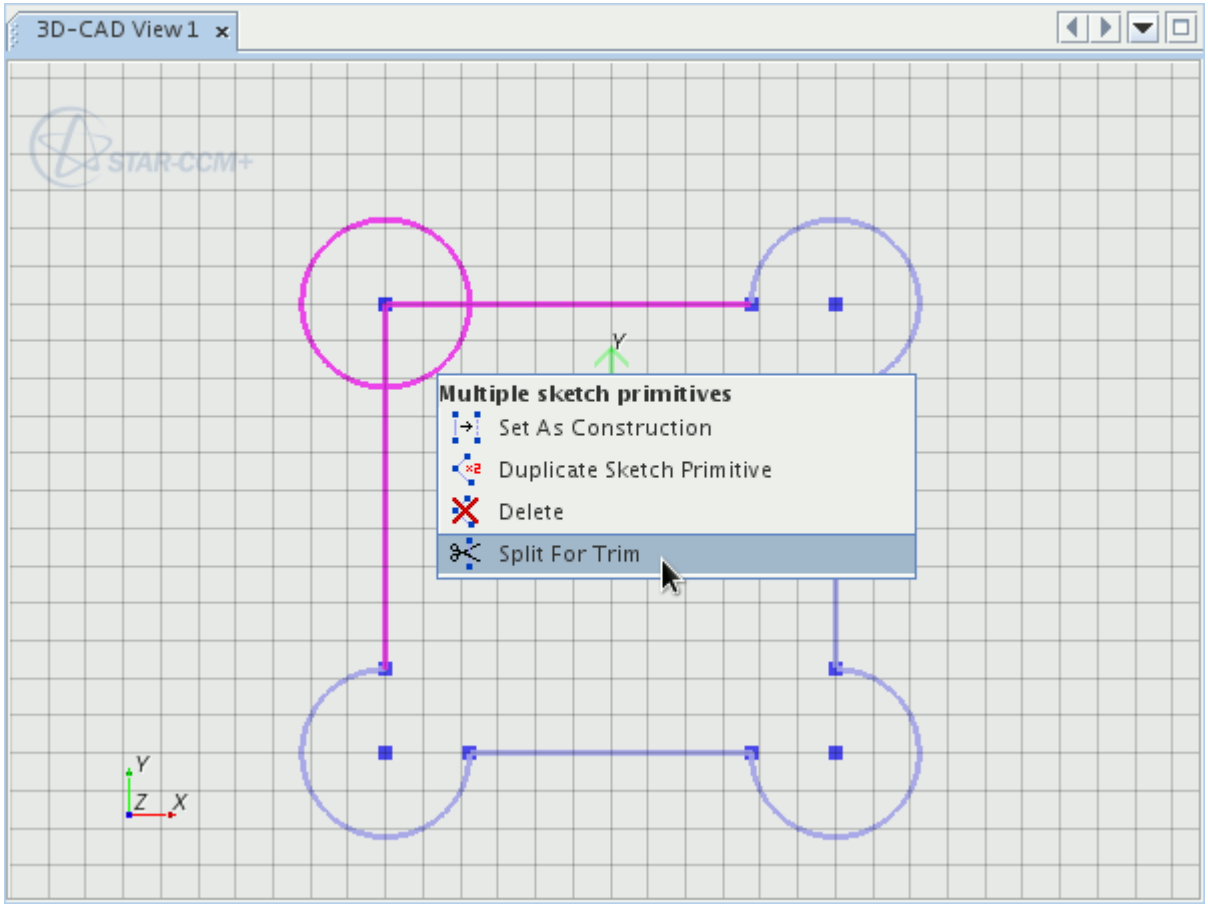
You can trim overlapping sketch entities by splitting them and deleting the unwanted segments. When using the **Split for Trim** operation, the entity that you select first is split where it intersects the sketch entities that you select next. The sketch entities that are used for splitting can either fully overlap the target, or have a point lying on the target.

Note: When a circle is split, the result is two circular arcs, therefore it must be intersected at a minimum of two points by the splitting entities. The same must be done when splitting an ellipse.

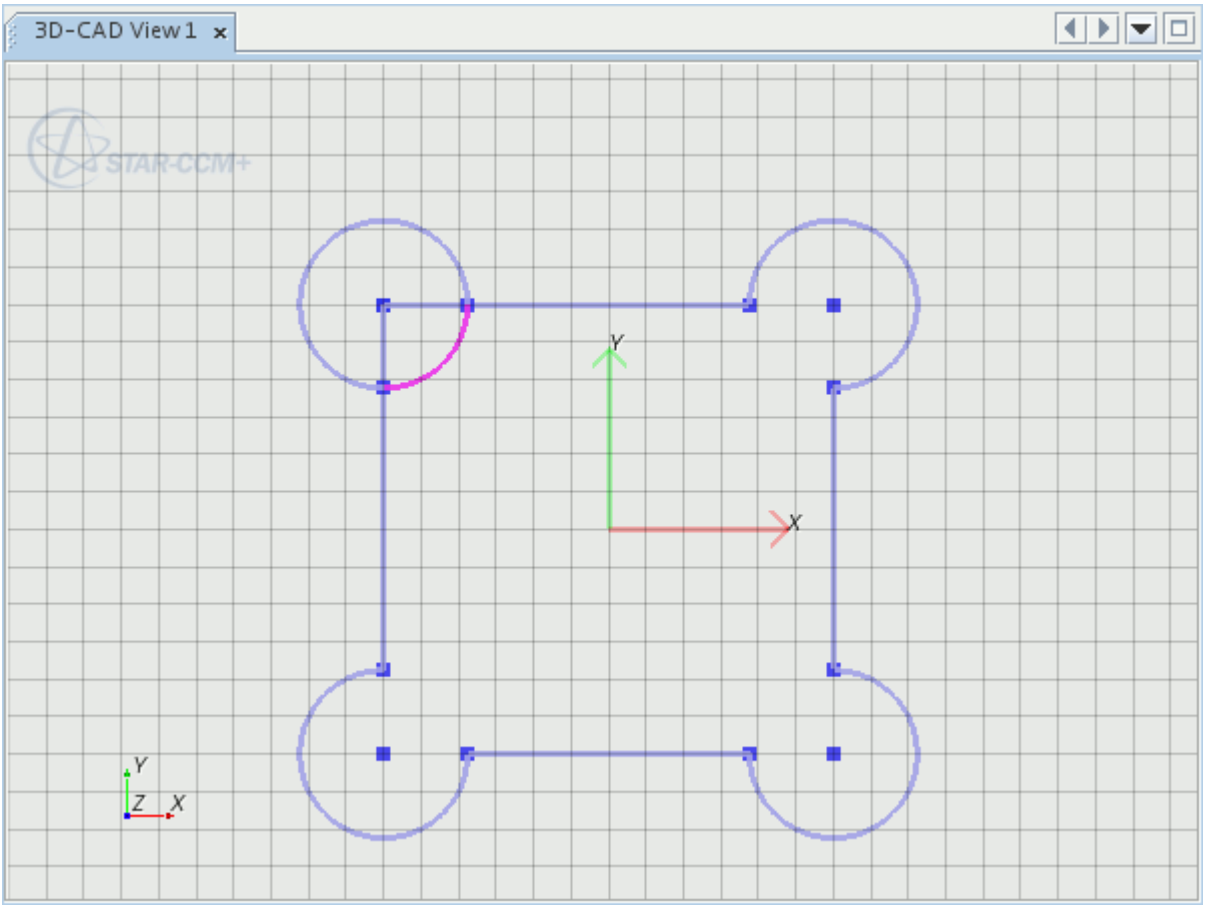
To split a sketch entity:

1. Select the target sketch entity in the *3D-CAD View* scene (the circle that is highlighted in the screenshot below).
2. Hold down the <`Ctrl`> key and select the sketch entities that you want to use to split the target (the two perpendicular lines that are highlighted below).

3. Right-click on any of the selected entities and select **Split For Trim**.



The target sketch entity splits at the points of intersection. The entities that intersected the target sketch entity remain unchanged. You can use the delete operation to trim any unwanted entities, such as the arc highlighted below. To complete the sketch below, you could use **Split for Trim** to split each of the perpendicular lines, using the circular arc as the splitting entity in each operation.



It is good practice to check the connections between points after using the **Split for Trim** operation. To check whether the points are connected following a split operation:

4. Click the intersection point and drag it to one side.
5. If the two sketch entities come apart when the point is moved, apply a coincidence constraint between the two points to form a closed sketch. You can drag the point back onto the existing point on the sketch plane; a coincidence constraint is automatically applied.

Dimensions that are attached to a sketch entity are lost when the **Split for Trim** action is carried out.

See Also:

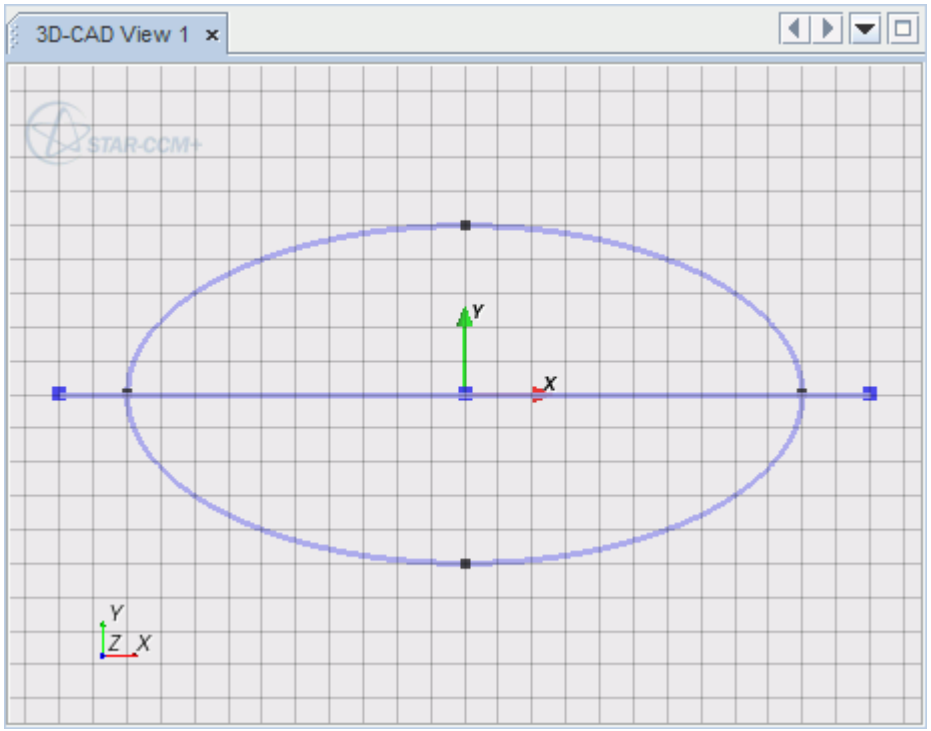
Deleting and Duplicating Entities
Sketch Entity Reference
Constraint Types
Dimensioning Sketches

Creating An Elliptical Arc

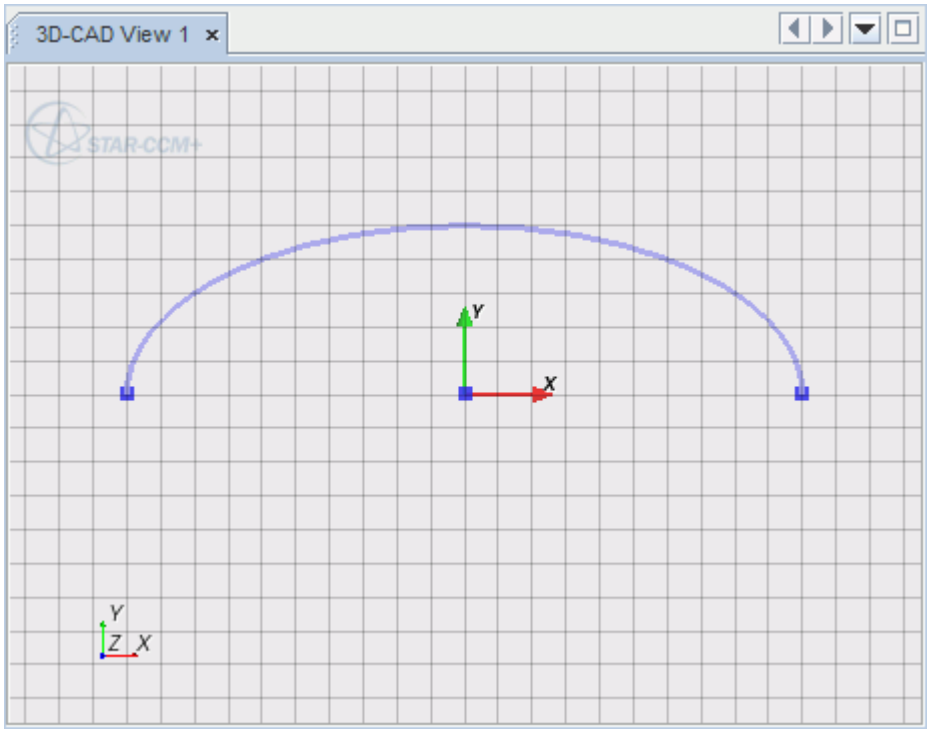
An elliptical arc cannot be created using the center-point or three-point arc tools. Create elliptical arcs by splitting an ellipse.

To construct an elliptical arc:

1. Create an ellipse.
2. Create a sketch entity that intersects the ellipse at two points.



3. Use the sketch entity to split the ellipse by the Split for Trim operation. The result, after deleting the unwanted entities remaining, is an elliptical arc.



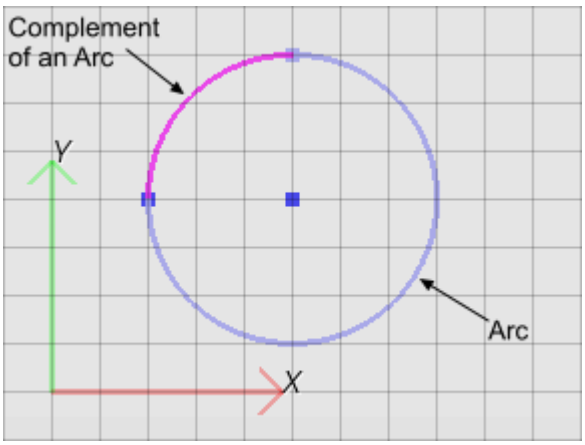
You can apply minor and major radius dimensions to elliptical arcs by right-clicking on the sketch and selecting Apply Minor Radius or Apply Major Radius. If you applied the minor or major radius dimension before splitting the ellipse, the dimension is retained, although the dimension annotation (and constraint) is removed from the scene. Reapply the dimension to see the annotation and add the constraint.

See Also:

Sketch Entity Reference
Splitting and Trimming Sketch Entities
Dimension Types

Converting An Arc Into Its Complement

The complement of an arc is the arc that you get when you swap the start and end points of the original arc. This feature is useful when the original arc is drawn opposite to the arc that you require.

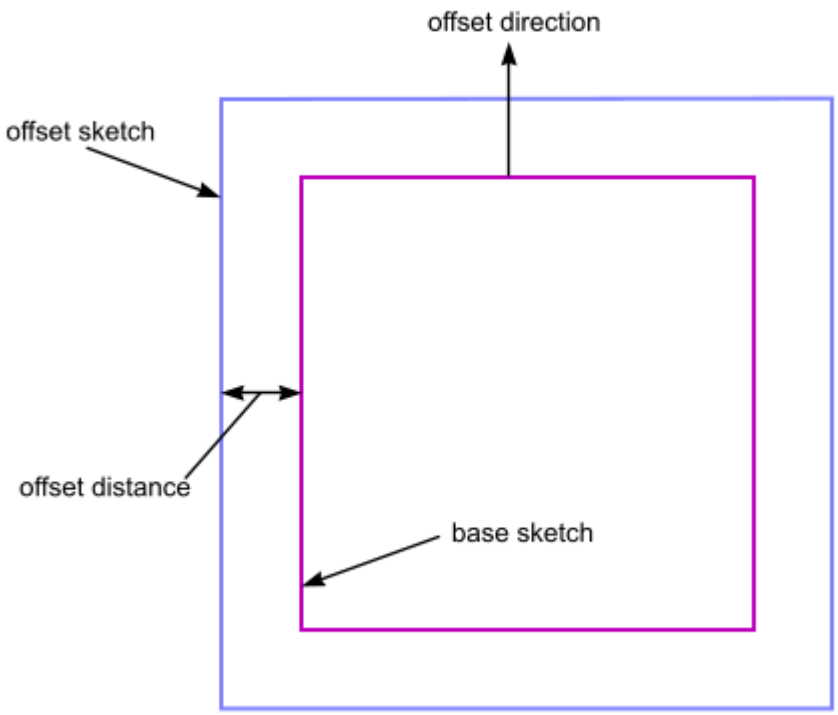


To convert an arc into its complement:

1. Right-click an arc in the *3D-CAD View* scene and select **Convert Into Complement**.

Offsetting Sketch Entities

You can uniformly displace sketch entities within the sketch plane. This operation is called an offset.

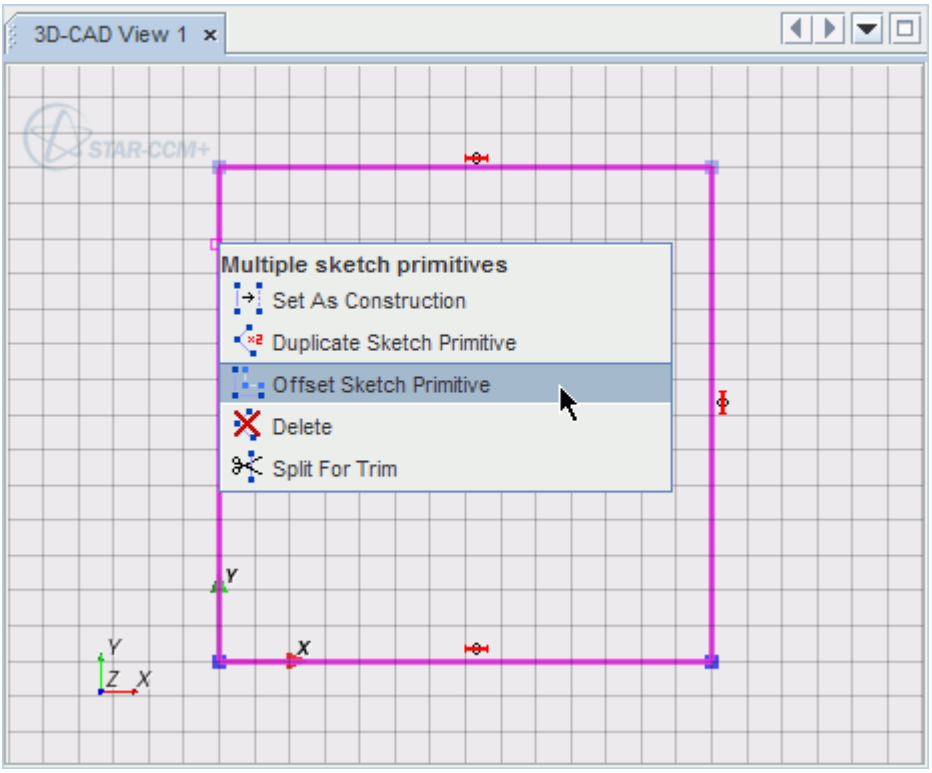


You can control the offset direction and the offset distance in each operation. This operation works with all sketch entities: lines, circles, ellipses, arcs, and splines. The offset operation is compatible with both open and closed sketches, and projected entities from outside the active sketch. The original sketch entities can be converted to construction lines or left as they are.

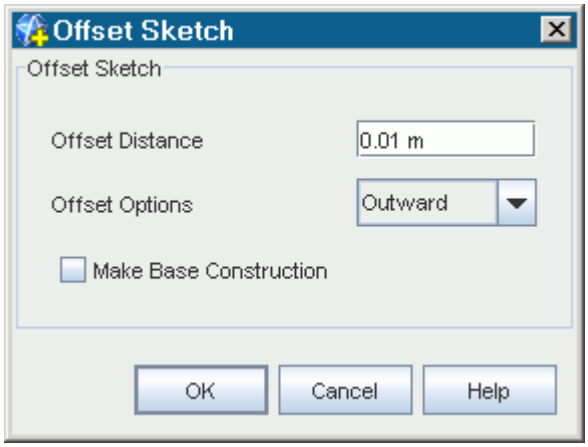
Note: The offset operation does not work with faces.

To offset a sketch entity:

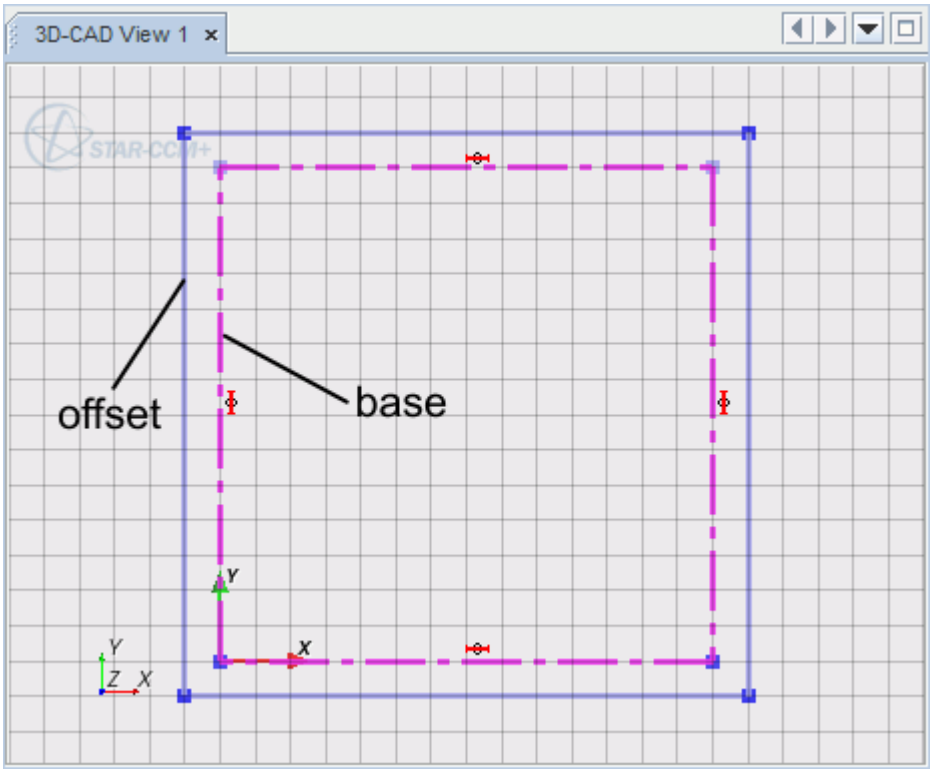
1. Select the desired entity (or entities) that you wish to offset.
2. Right-click on one of the selections and select **Offset Sketch Primitive**.



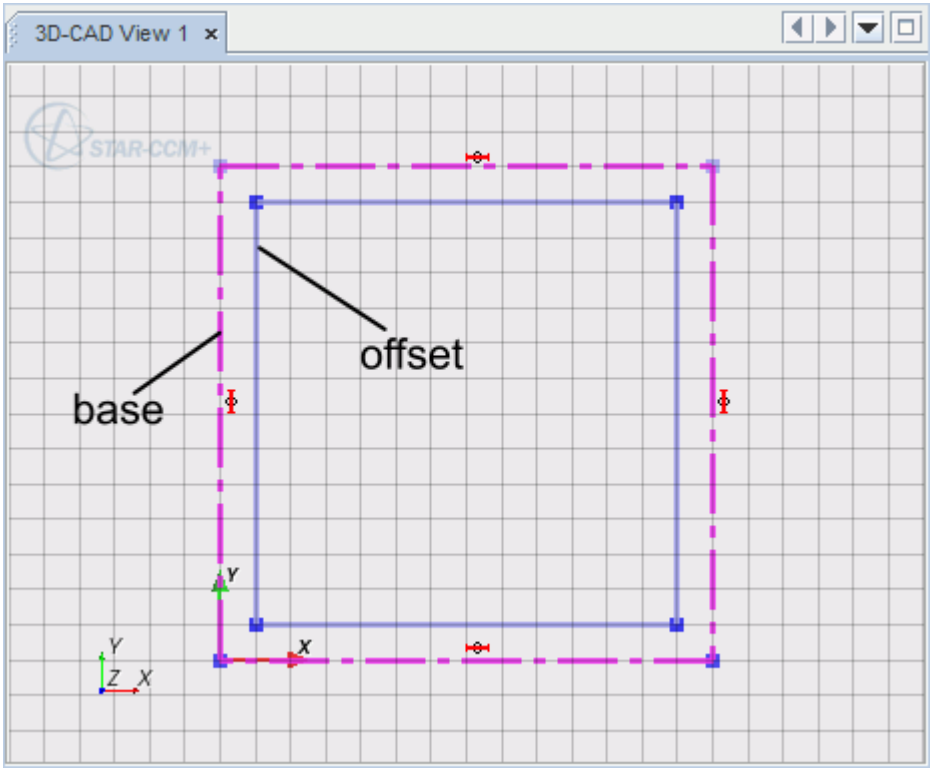
The *Offset Sketch* dialog appears.



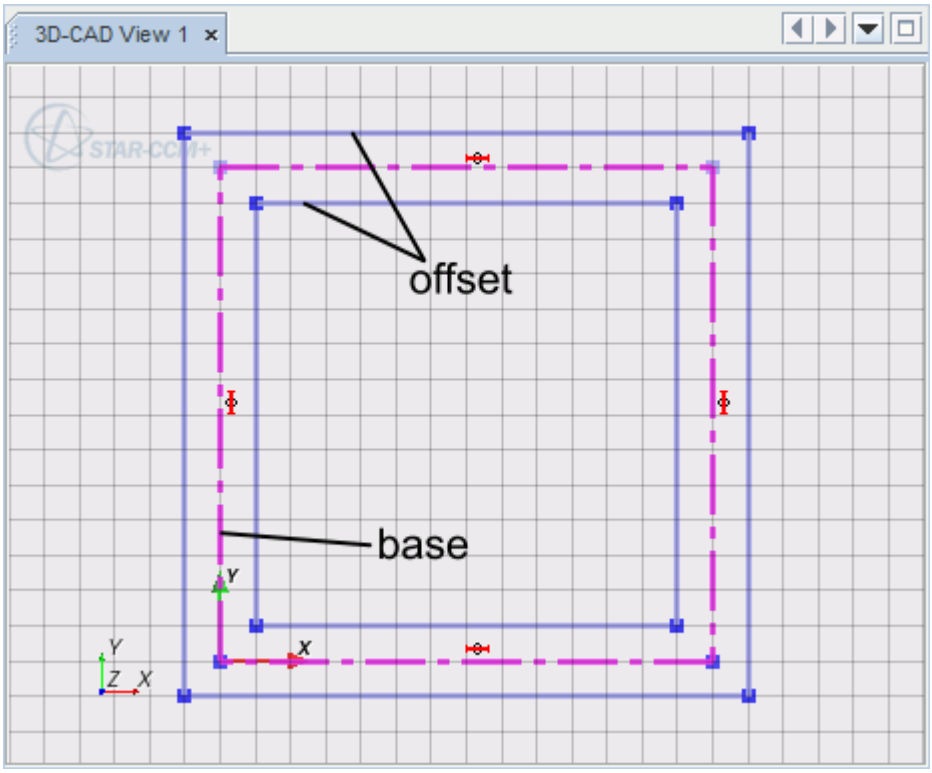
3. In *Offset Distance*, set the size of the offset. This value controls the distance between the base sketch entity and the offset sketch entity.
4. In *Offset Options*, select the type of offset that you want to perform. Three options are available. These options control the direction of the offset. In the examples that are shown below, the base entities are shown as highlighted and have been converted to construction lines:
   - **Outward—**in a closed sketch, the offset is on the outside of the enclosed sketch area.



- **Inward**—in a closed sketch, the offset is on the inside of the enclosed sketch area.



- **Bi-directional—**this causes the offset to take place on both sides of the selected sketch entities.

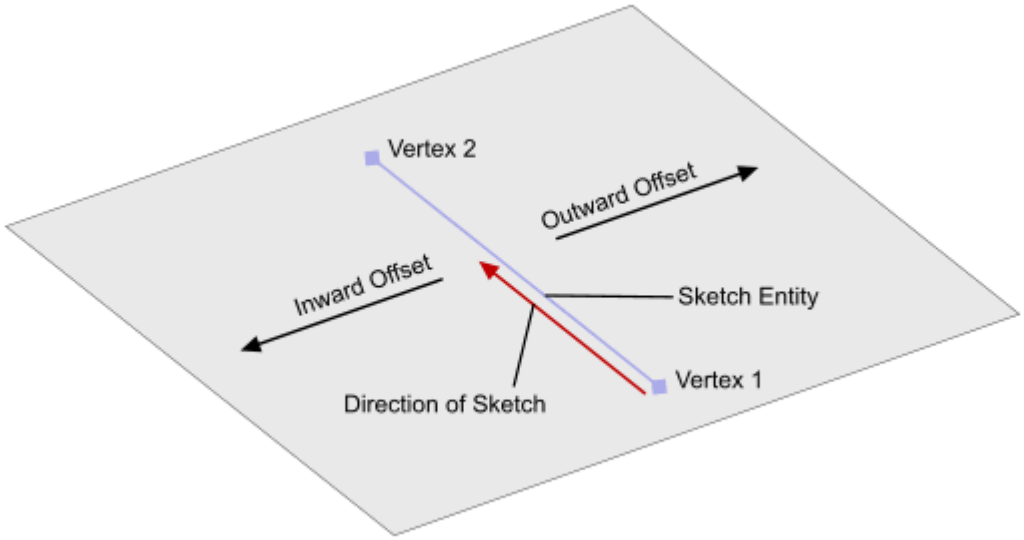


5. If you want to convert the base sketch to a construction line during the operation, activate *Make Base Construction*.
6. Click **OK**.

The offset operation is performed. You can undo/redo the operation using the appropriate buttons in the *Sketch* in-place dialog.

Offsetting Open Sketches

In an open sketch, the offset direction is set according to the direction of the sketch (that is, the order in which you created the vertices). For example, consider the case that is shown below:



Creating an offset with the *Offset Options* property set to **Outward** results in the offset being placed on the right-hand side of the sketch entity. If the *Offset Options* property is set to **Inward**, the offset appears on the left side of the original, as you look along the direction of the line.

See Also:

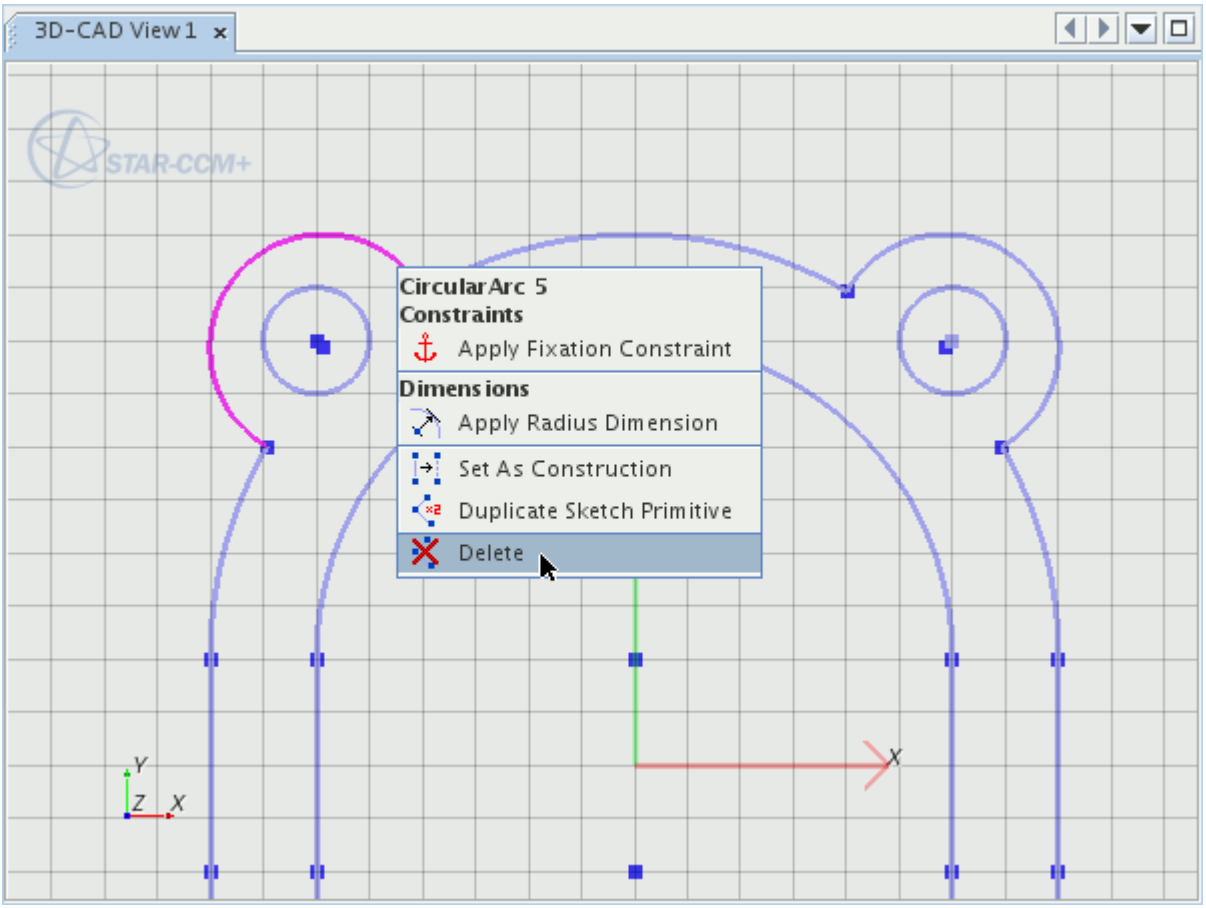
Sketch Entity Reference

Projecting Entities to the Active Sketch

Deleting and Duplicating Entities

To duplicate a sketch entity:

- Right-click on the desired entity and choose **Duplicate sketch primitive**.



The sketch entity is duplicated. The duplicated sketch entity occupies the same location as the original. Move and edit it as required. You can duplicate multiple sketch entities at once by holding the <`Ctrl`> key while selecting them.

Deleting Sketch Entities

To delete a sketch entity:

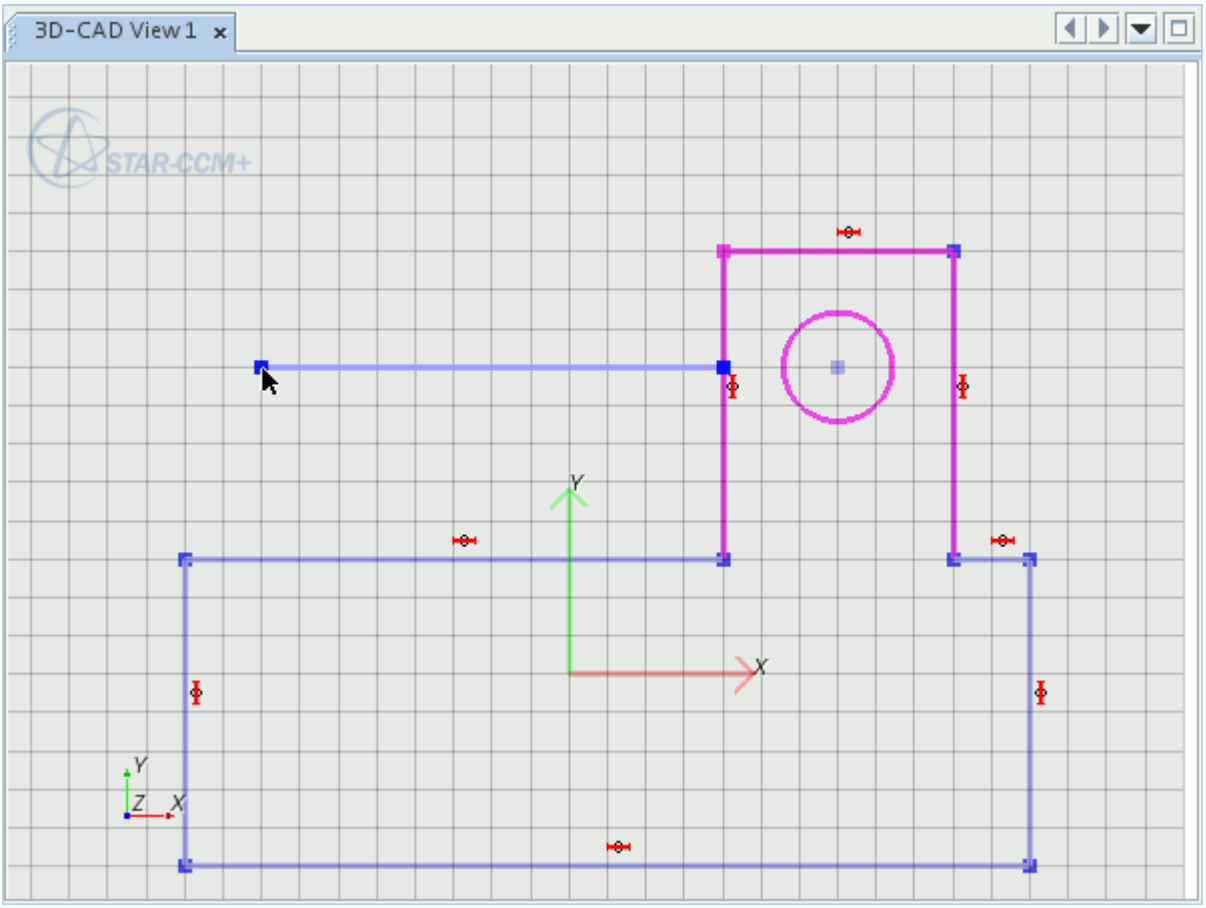
- Right-click on the desired entity and choose **Delete**.
- Alternatively, select the sketch entity and press the <`Delete`> key.

You can delete multiple sketch entities at once by holding the `<Ctrl>` key while selecting them.

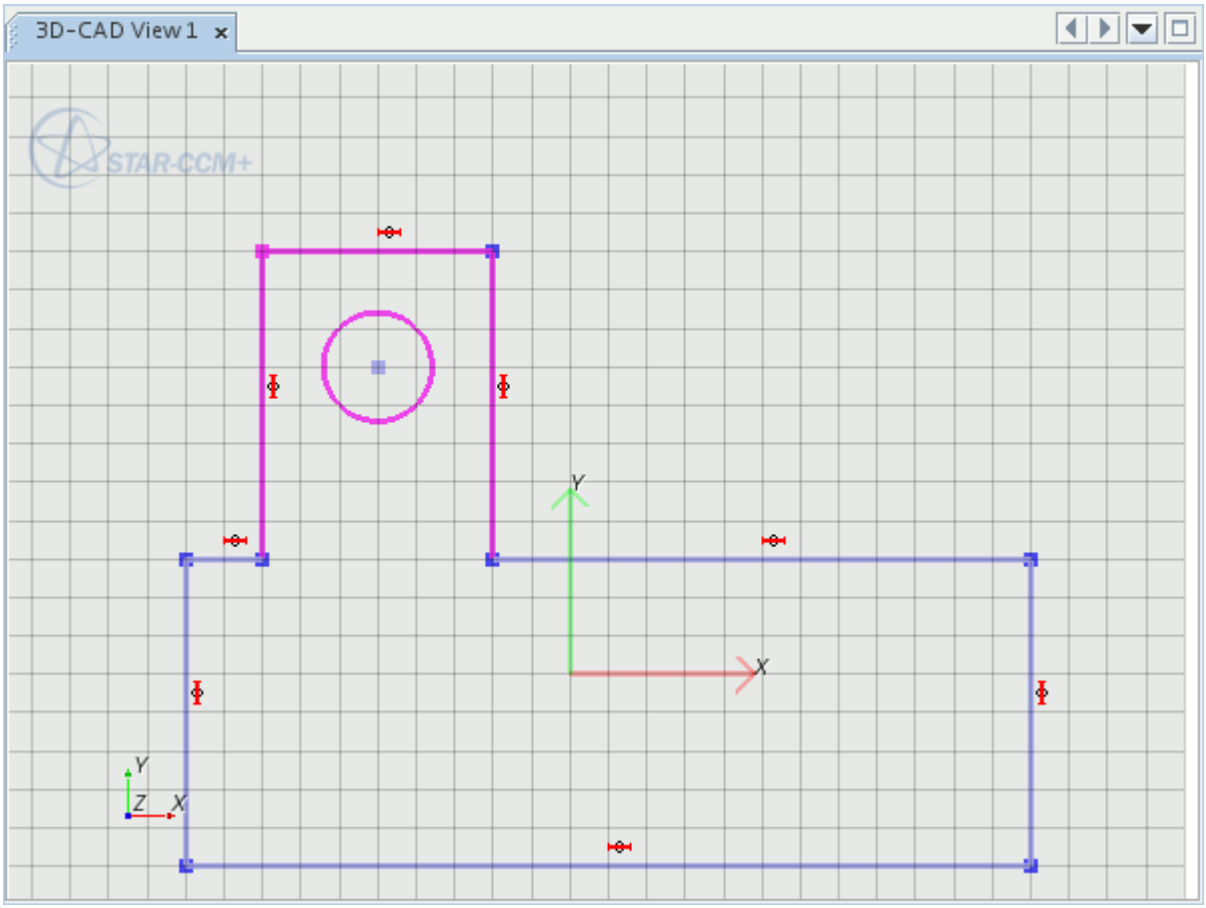
Moving Sketch Entities

To move a sketch entity:

1. Click and drag the sketch entity to its new position on the sketch plane. You can move several sketch entities at the same time by first holding down the <`Ctrl`> key and selecting the sketch entities that you want to move. Then, keeping the <`Ctrl`> key down, click on one of the selected entities, and drag it to a new position.



All of the selected sketch entities are moved according to the translation vector defined by the drag-and-drop operation.



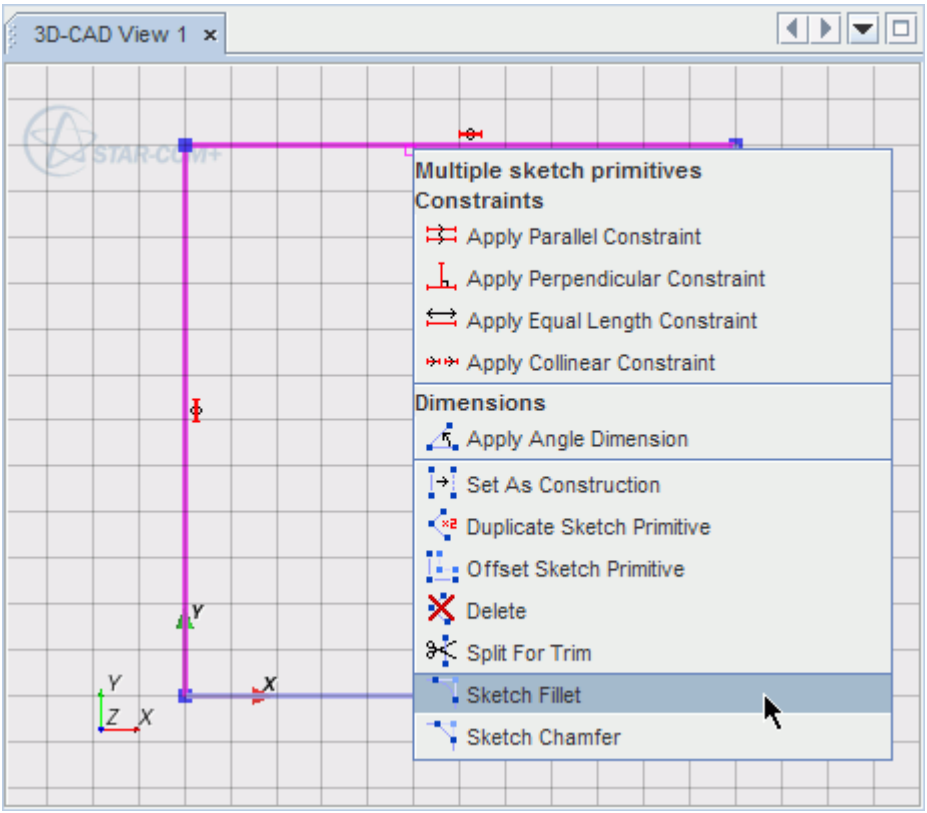
Creating Sketch Fillets

The **Sketch Fillet** option rounds off the corner that is formed between two lines on a sketch.

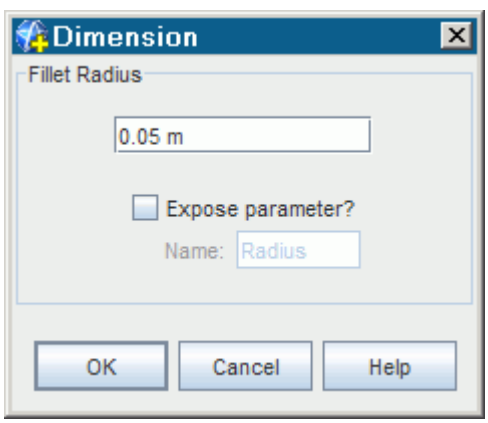
You can only create a fillet between two lines at a time: these lines can be two straight lines, a straight line and a curve, or two curved lines. The two lines can be disjointed, intersecting or have a common end point.

To create a sketch fillet:

1. Select the two sketch lines that you want to fillet.
2. Right-click on one of these lines and select **Sketch Fillet**.



The *Dimensions* dialog appears.



3. Enter the radius of the fillet in the *Fillet Radius* textbox. You can define the fillet radius as a numerical value, or as an expression using design parameters.
4. If you set up a design parameter and want to expose it, activate the *Expose Parameter?*.
5. Click **OK**.

The fillet is applied to the selected entities.

Two sets of tangent and coincidence constraints are added to the sketch when the fillet is created. A radius dimension is also created. You can edit the radius dimension to control the fillet radius.

Note: The fillet radius must be smaller than the length of the shortest sketch line.

You can undo/redo the sketch fillet feature using the appropriate buttons in the *Sketch* in-place dialog.

Intersecting Lines

You can create fillets between intersecting lines. STAR-CCM+ retains the longest portions of the selected lines, and creates a fillet at the point where they cross. This operation is shown in the following diagrams:

There are two fillet locations (the two intersections), STAR-CCM+ chooses the lower intersection as this leaves the longer line segments. The remaining parts of the intersecting lines are discarded:

The sketch entities remain selected after the operation so that you can create a fillet again without the need to reselect the entities. In the example, creating a second fillet would connect the lines at the top:

Disjointed Lines

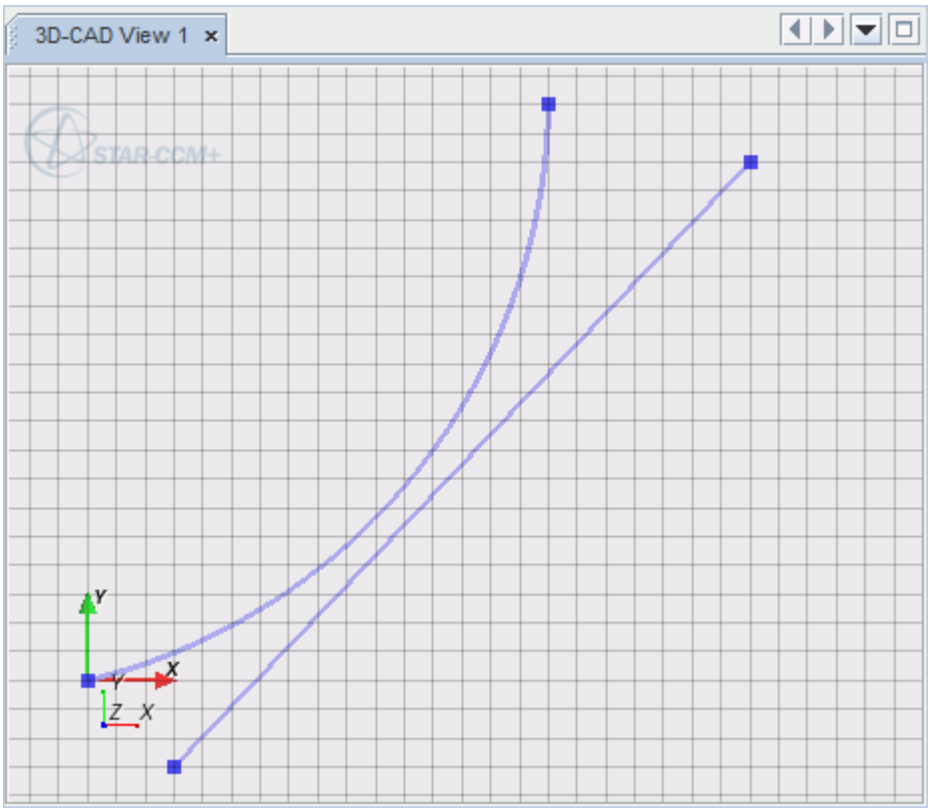
You can create fillets between disjointed lines. STAR-CCM+ extends each line to a point where it meets with the other line, and creates the fillet there. This operation is shown in the following diagrams:

3D-CAD View 1
STAR-CCM+
Y
Z
X
Y
X

3D-CAD View 1
STAR-CCM+
0.02 m
Y
Z
X
Y
X

Non-Intersecting Lines

You can create fillets between two non-intersecting sketch features. STAR-CCM+ places a fillet at the point that retains the two longest segments of the features.



In the following example, there are two possible places where a fillet could be placed (based on the specified fillet radius). The longest segments are retained when the fillet is placed on the top right, as shown highlighted in red:

3D-CAD therefore chooses this location, and the remaining segments are discarded:

See Also:

What Is a Design Parameter?

Constraint Types

Dimension Types

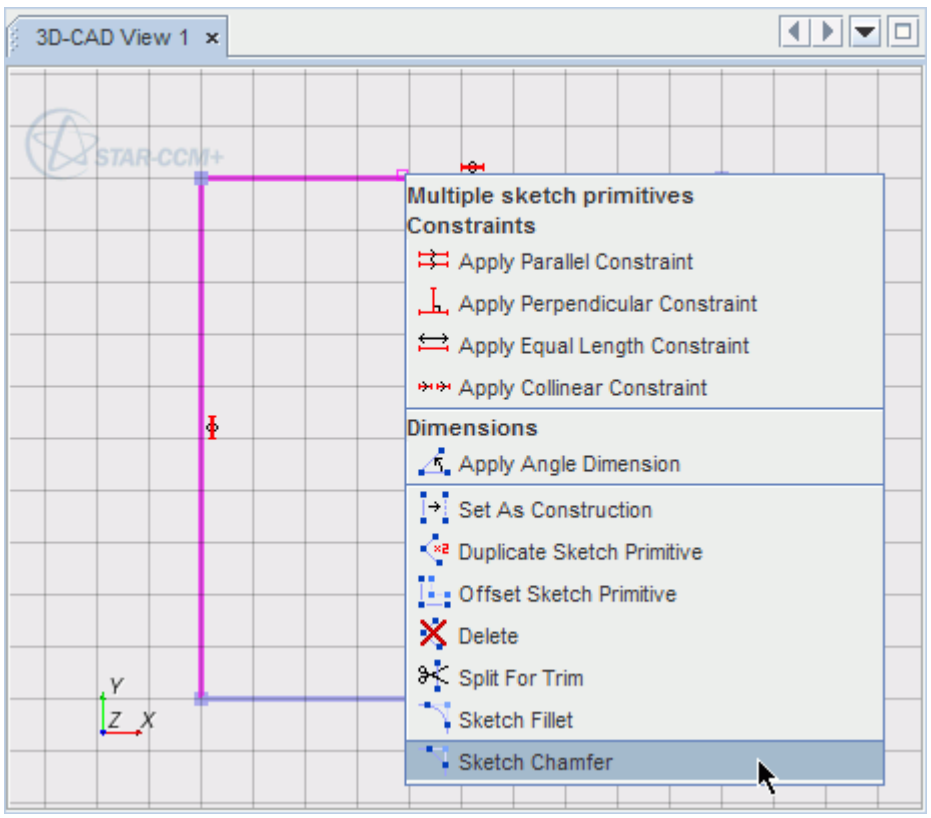
Creating Sketch Chamfers

A sketch chamfer is a feature that makes a straight cut across the corner that is formed between two lines on a sketch.

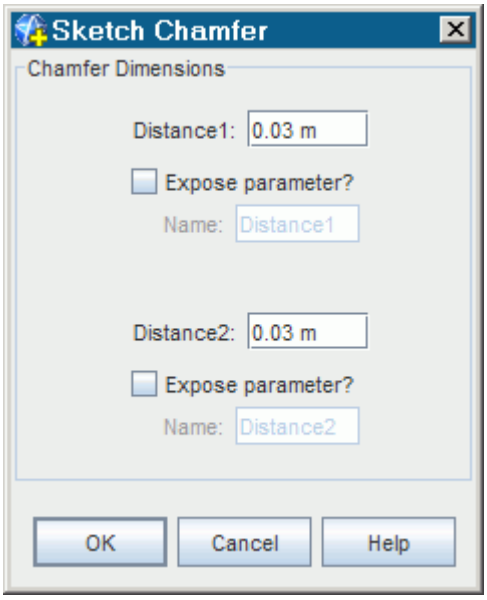
You can only create chamfers between two lines at a time; these lines can be disjointed, intersecting or have a common end point. When the lines intersect each other, the chamfer is created between the two longest segments of the lines.

To create a sketch chamfer:

1. Select the two sketch lines that you want to chamfer.
2. Right-click on one of these lines and select **Sketch Chamfer**.



The *Sketch Chamfer* dialog appears.



3. Enter values for *Distance 1* and *Distance 2*. These values define the size of the chamfer and are the distances that the chamfer will be from the point where the two lines meet.

   Note: You can define the chamfer using a numerical value, or by an expression using design parameters. Activate *Expose Parameter ?* for each distance property to expose it as a design parameter.

4. Click **OK**.

The chamfer is applied to the selected entities.

Two sets of coincidence constraints are added to the sketch when the chamfer is created. Two distance dimensions are also created; you can edit these dimensions to control the chamfer size.

Note: The dimensions controlling the chamfer must be smaller than the length of the shortest original sketch line. Once created, the chamfer dimensions can be edited. The sketch chamfer feature can be undone/redone using the appropriate buttons in the *Sketch* in-place dialog.

See Also:

What Is a Design Parameter?

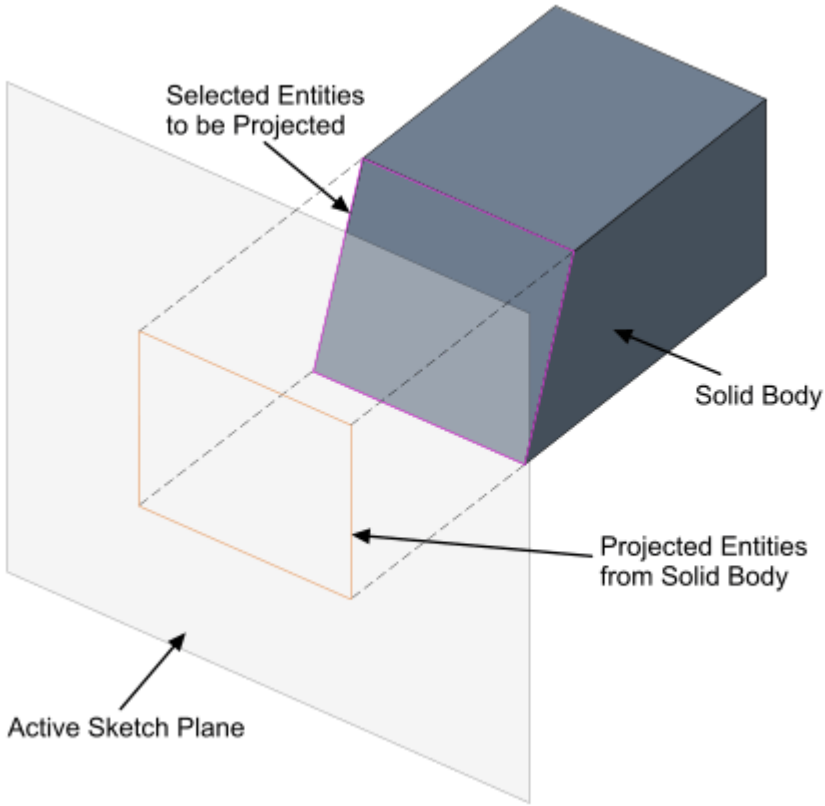
Constraint Types

Dimension Types

Projecting Entities to the Active Sketch

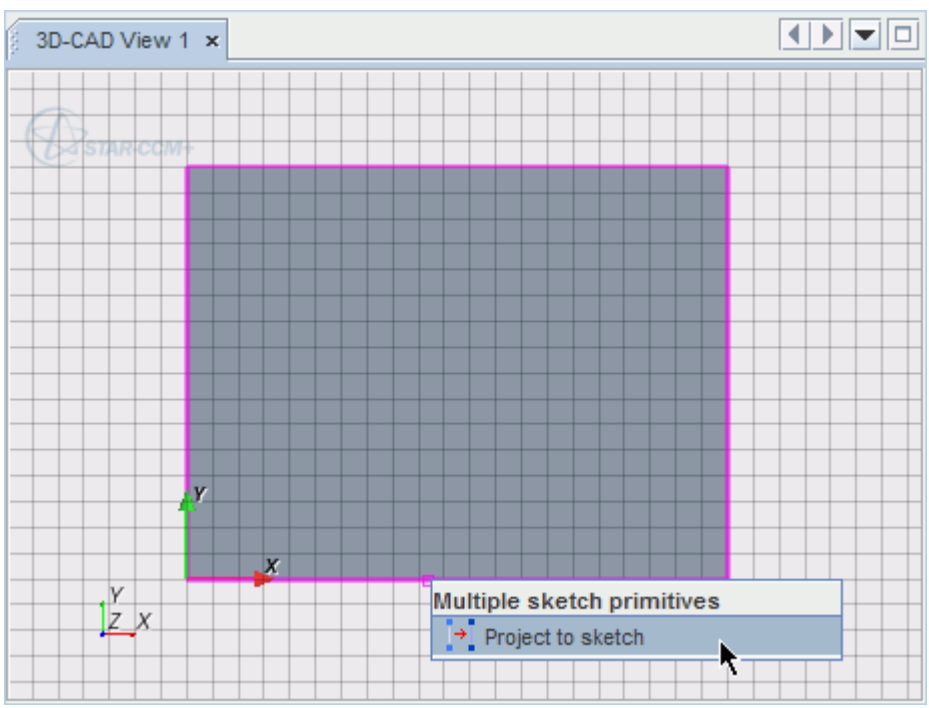
You can project sketch entities from other sketches onto the active sketch.

Additionally, you can project edges and faces from other bodies to the active sketch.

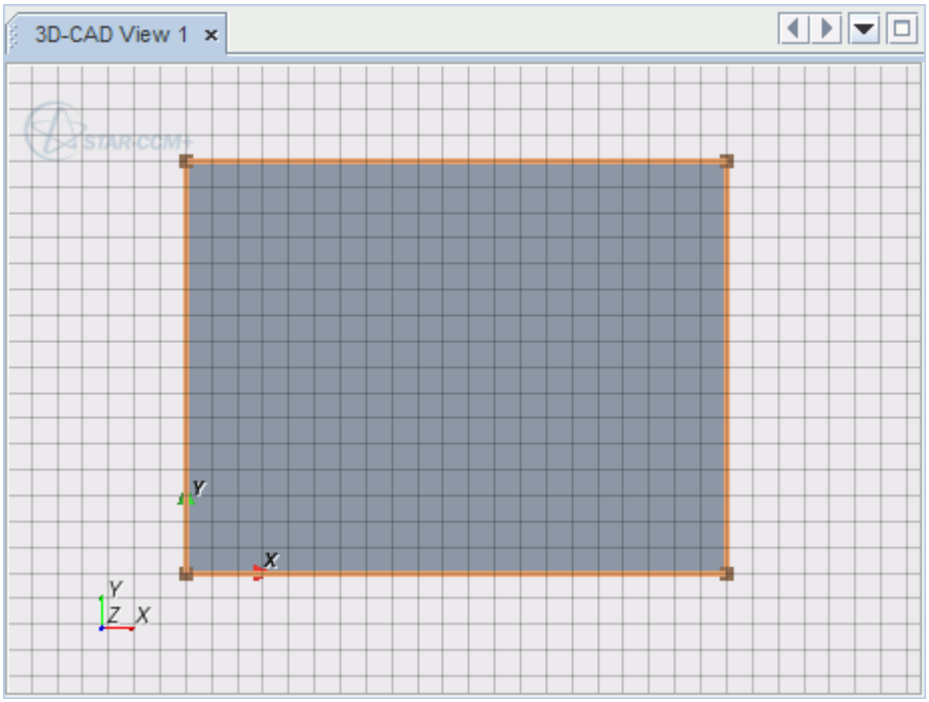


To project entities:

1. Select the desired edges or faces that you wish to project.
2. Right-click on one of the selections and select **Project to sketch**.



The selected entities (or in the case of a face, the edges of the face) are projected onto the active sketch. Projected entities are displayed in orange for easier identification.



Note: The projected entities are associative, meaning that if you modify the original entity, the projection is automatically updated to match the modification.

You can use projected entities with other sketch entities in the current sketch. You can undo or redo the project feature operation using the appropriate buttons in the *Sketch* in-place dialog.

You cannot move or modify projected entities. You can, however, create a duplicate of the projected entities and modify those.

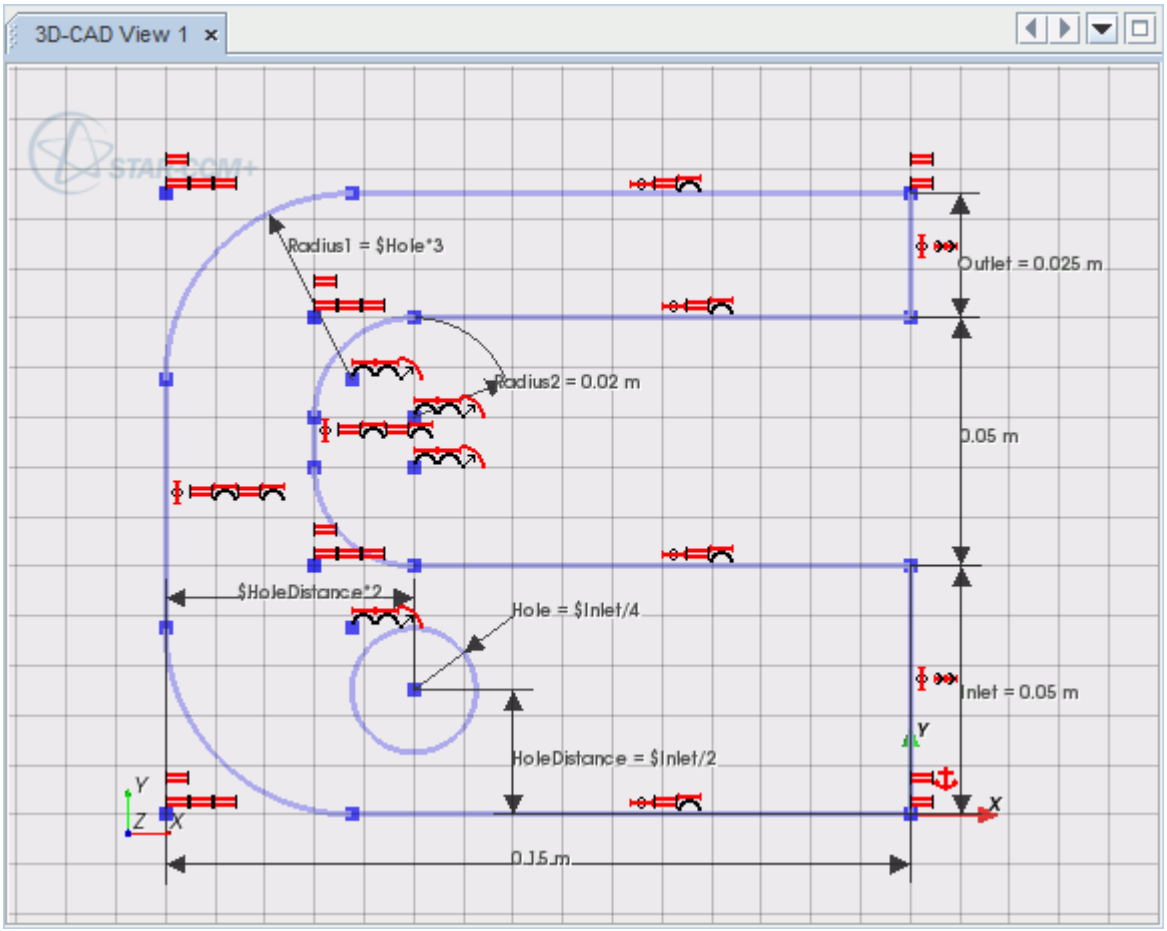
Note: The duplicated entities are not associated with the original source.

See Also:

Sketch Entity Reference

Showing/Hiding Relations

You can turn off relations, such as constraints and dimensions when the sketch scene becomes too cluttered, as shown in the following screenshot.

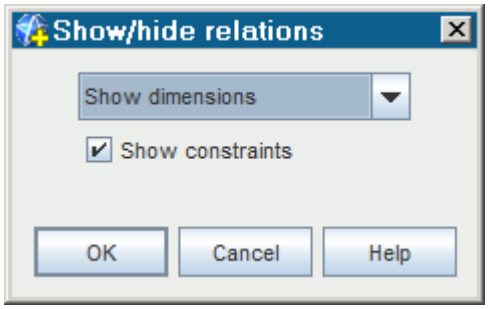


You can visualize dimensions and constraints independently of each other.

To control the visualization of the relations:

1. Click the (**Show/hide relations**) button on the *Sketch* panel.

   The *Show/hide relations* dialog appears.

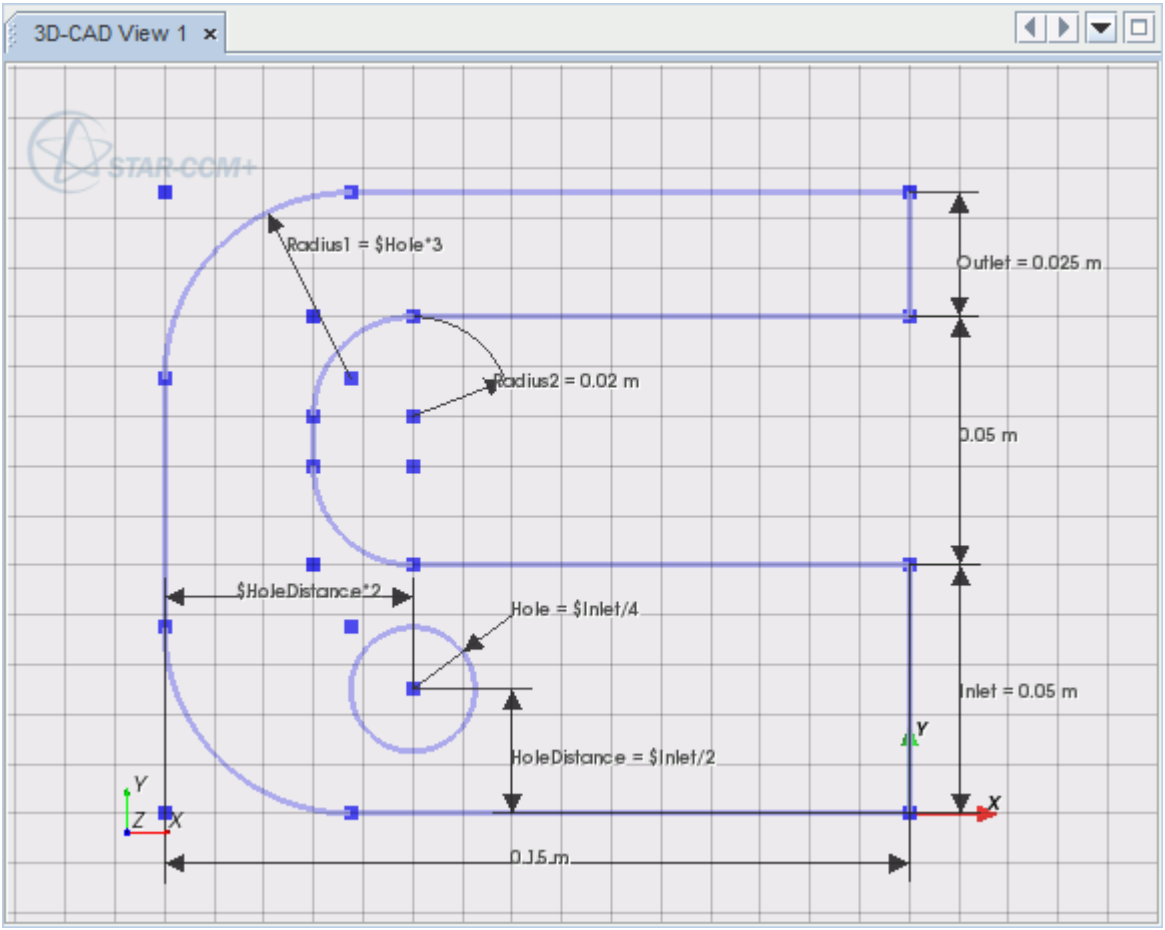


2. Here, you can control what is displayed in the sketch scene:
   - Controlling the display of dimensions
   - Controlling the display of constraints
3. Click **OK** to close the dialog and apply the changes.

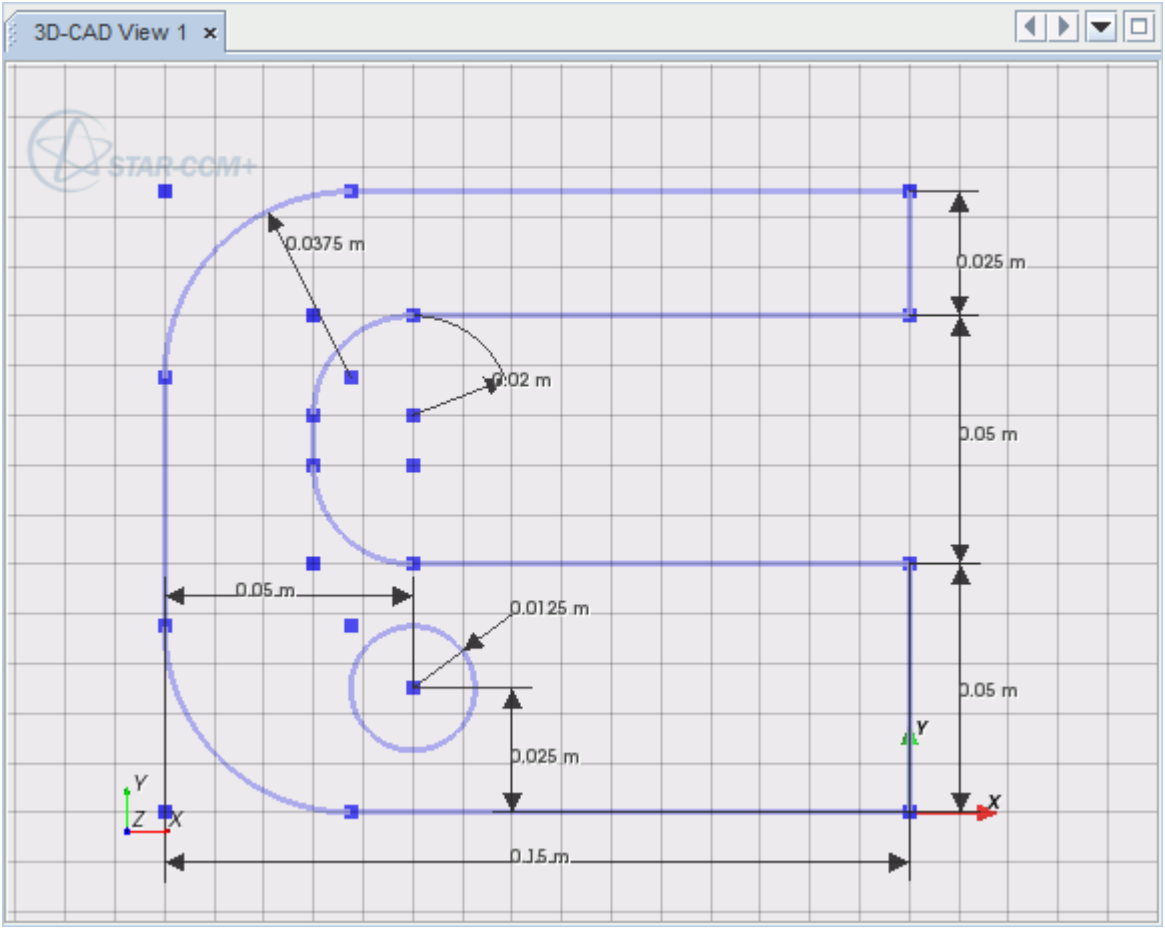
Controlling the Display of Dimensions

To control the display of the dimensions in the sketch scene, select one of the options from the drop-down menu:

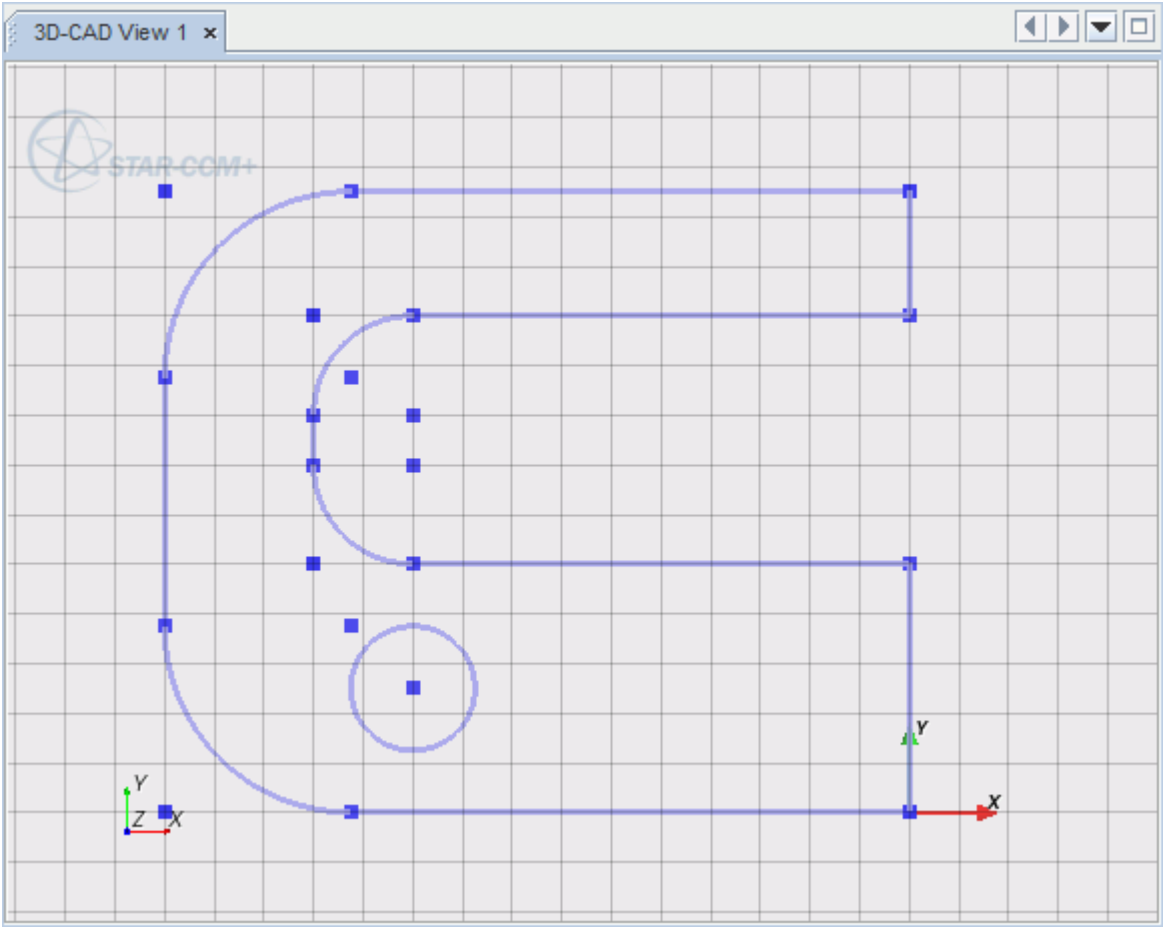
- **Show dimensions**—shows the dimensions in the sketch scene, including expressions and design parameter names.



- **Show dimension values only**—shows only the dimension values in the sketch scene. Expressions and design parameters are converted to dimension values. This option reduces the amount of text that appears in the sketch scene.



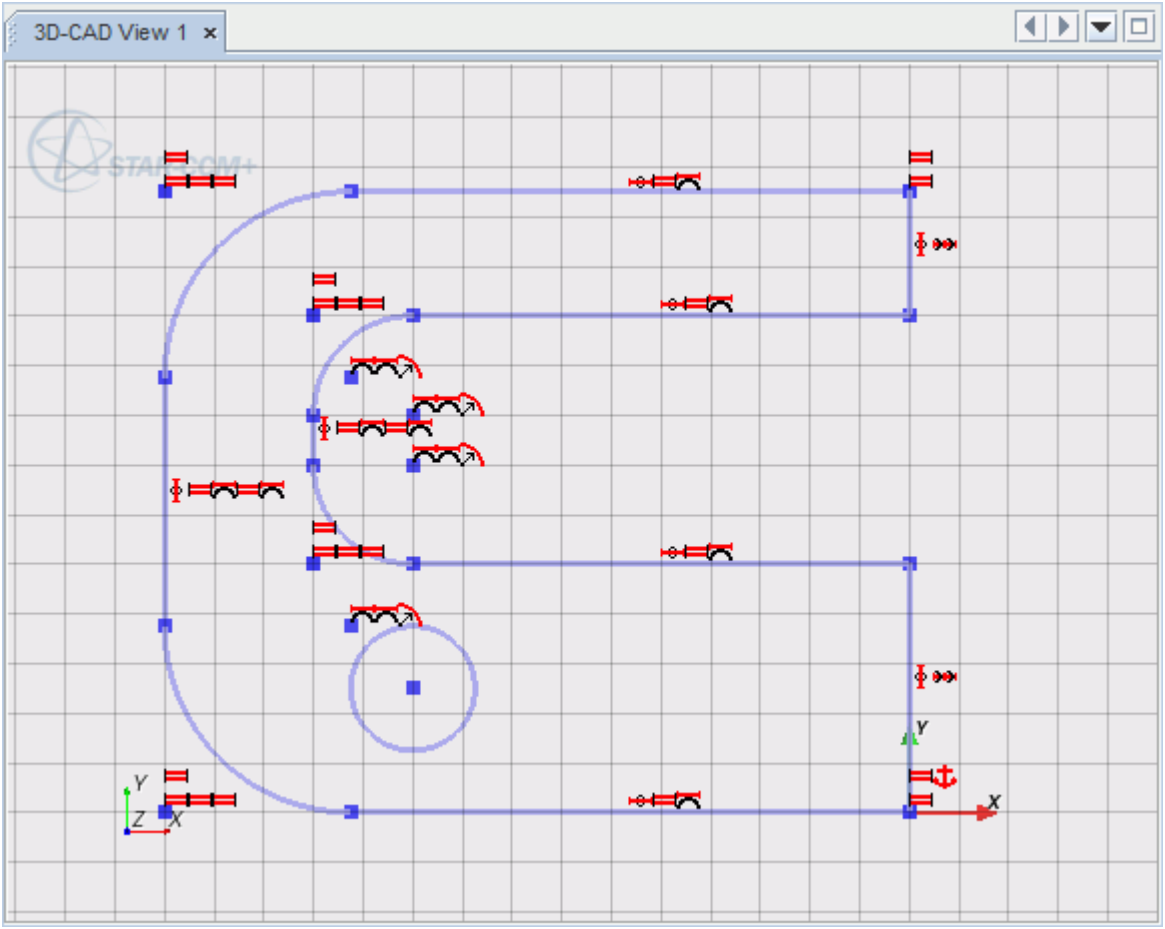
- **Hide dimensions**—hides all dimensions in the sketch scene.



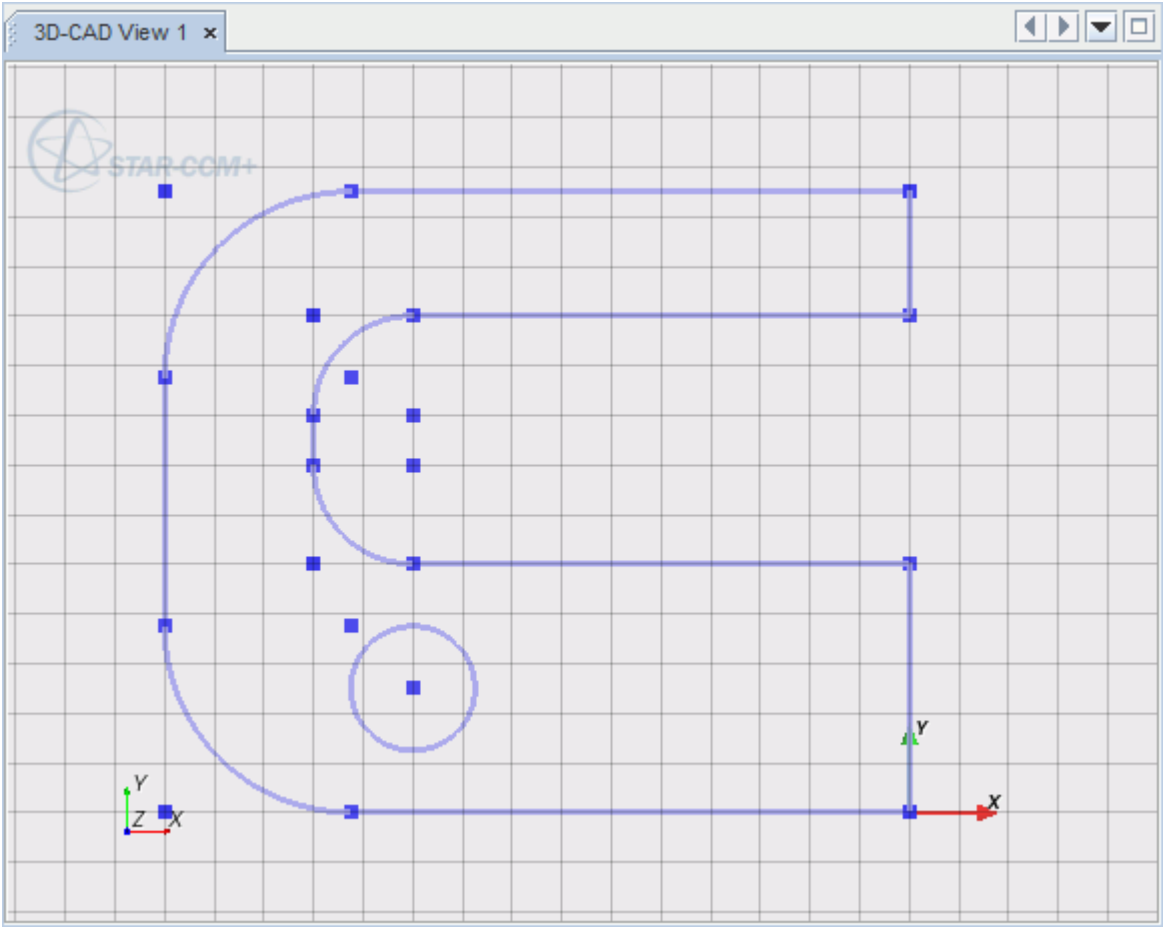
Controlling the Display of Constraints

To control the display of the constraints, activate or deactivate the *Show constraints* checkbox:

- *Activated*—the constraints are displayed in the sketch scene.



- *Deactivated*—all constraints are hidden in the sketch scene.



See Also:

Constraining a Sketch

Dimensioning Sketches

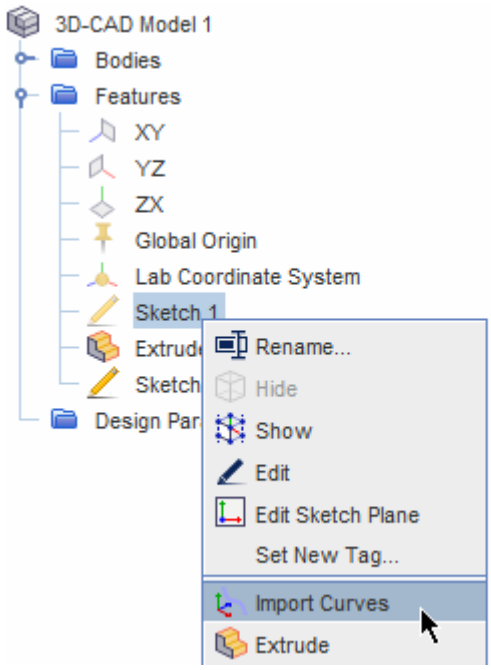
Importing and Replacing Sketch Curves

You can combine tabular and user-created sketches by importing a curve into an existing 2D or 3D sketch. You can also replace a specific spline within a sketch with a new spline from a .csv file to enable tabular-based design changes.

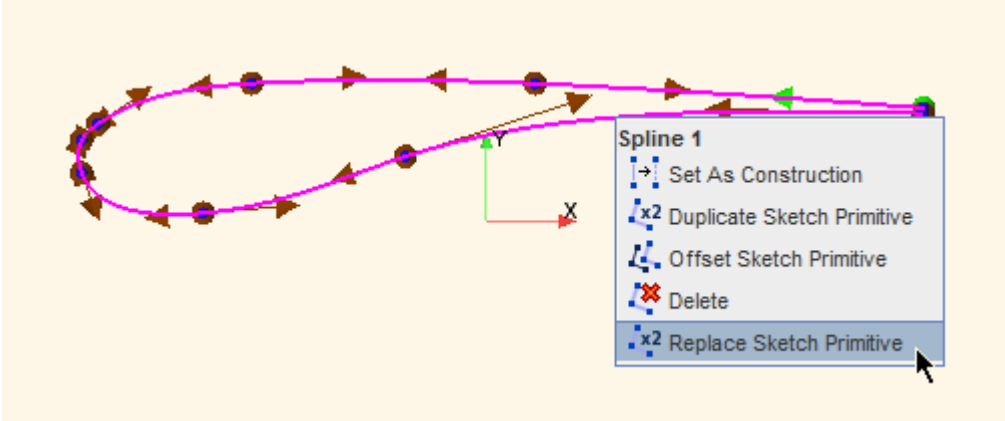
All curves that you import must be in .csv format. See File Format for Imported Curves.

To import and replace sketch curves:

1. Choose where to import the sketch curves, or which spline to replace:
   - To import sketch curves into an existing sketch, right-click the target sketch in the 3D-CAD feature tree and select **Import Curves**.



- To replace an existing spline within a sketch, while in sketch mode, right-click the target spline and select **Replace Sketch Primitive**.



2. In the *Open* dialog, select the `.csv` file that contains the curve data and click **OK**.
3. In the *Import Curves to Sketch* dialog, set the import options and click **OK**.

   You can import the curve as a spline or a polyline. You can also choose to close the curve if it is open. See Curve Import Options.

The curve is imported into the sketch, and is added in addition to any sketch entities that are already within it. When replacing a specific spline, use the **Undo** action to revert back to the original spline.

Constraining a Sketch

Constraints are used to impose geometric limitations on sketch entities as well as to control their position on the sketch plane.

You can apply a constraint to a single sketch entity to define how it relates to the sketch plane, or to multiple sketch entities to define the relationship between them. It is possible to constrain the active sketch using sketch entities from another existing sketch or geometry. The constraints that are shown in the right-click menu differ depending on the selection of sketch entities.

To apply a constraint:

1. In the *3D-CAD view* scene, select the sketch entities that you want to constrain.
2. Right-click one of the selections and select the type of constraint you want to apply.

   Only constraints that are relevant to your selection appear in the right-click menu. See Constraint Types.

Each constraint that you apply is depicted by a symbol, such as ⊗ for a concentric constraint, and is added to the sketch. Move the mouse pointer over these symbols in the *3D-CAD View* scene to highlight the pair of sketch entities that are associated with the constraint.

To delete a constraint, select the constraint symbol it in the *3D-CAD View* scene and press the `<Delete>` key.

When you apply a constraint to a pair of sketch entities, 3D-CAD can move one of the sketch entities in order to satisfy the constraint. You cannot influence which sketch entity moves—3D-CAD determines which entity to move. If, however, you want to ensure that a particular sketch entity does not move when the constraint is applied, apply a fixation constraint to one of the sketch entities.

When you apply constraints to multiple sketch entities at the same time, 3D-CAD applies the constraints to each pair of sketch entities. Consequently, you end up with multiple constraint symbols added to your sketch. Move the mouse pointer over each constraint symbol to highlight the pair of sketch entities that are associated with each constraint.

To use the sketch plane origin for constraining a sketch, create a point at `[0, 0]` and fix it in position. You can then use this point to constrain sketch entities in relation to the origin.

If the new constraint cannot be satisfied, due to any existing constraints or dimensions on the sketch, an error message is shown and the constraint is not applied. To resolve this issue, delete any existing constraints or dimensions that are counteracting the new constraint.

3D-CAD considers lines to be infinitely long and circular arcs to be complete circles when constraints are applied. Therefore, a point that is constrained to coincide with a line does not always appear to lie on the line. Similarly, a point that coincides with a circular arc does not appear to lie on the arc. In each case, the point is coincident with some point on the line segment that defines the infinite line, or the arc that defines the full circle.

Contents:

Constraint Types
Applying Constraints that Reference Other Sketches or Existing Geometries
Reviewing Applied Constraints
Removing Constraints

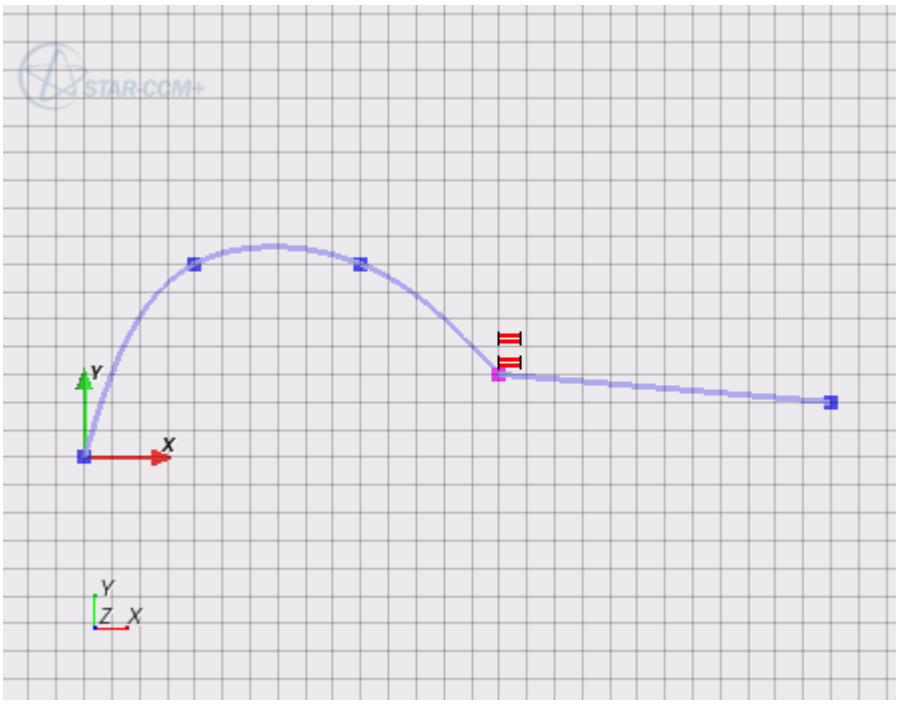
See Also:

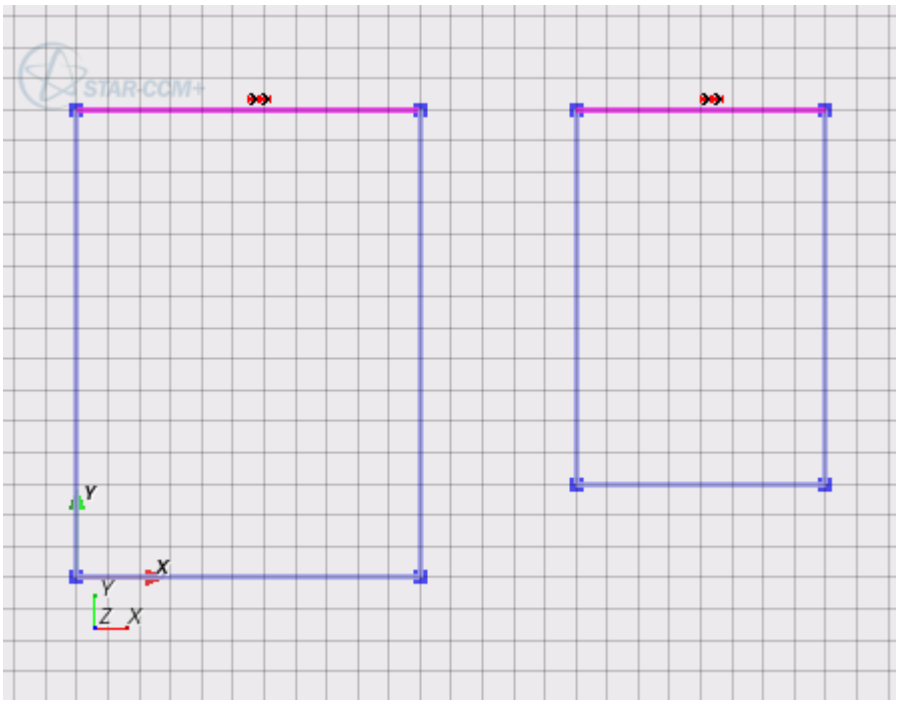
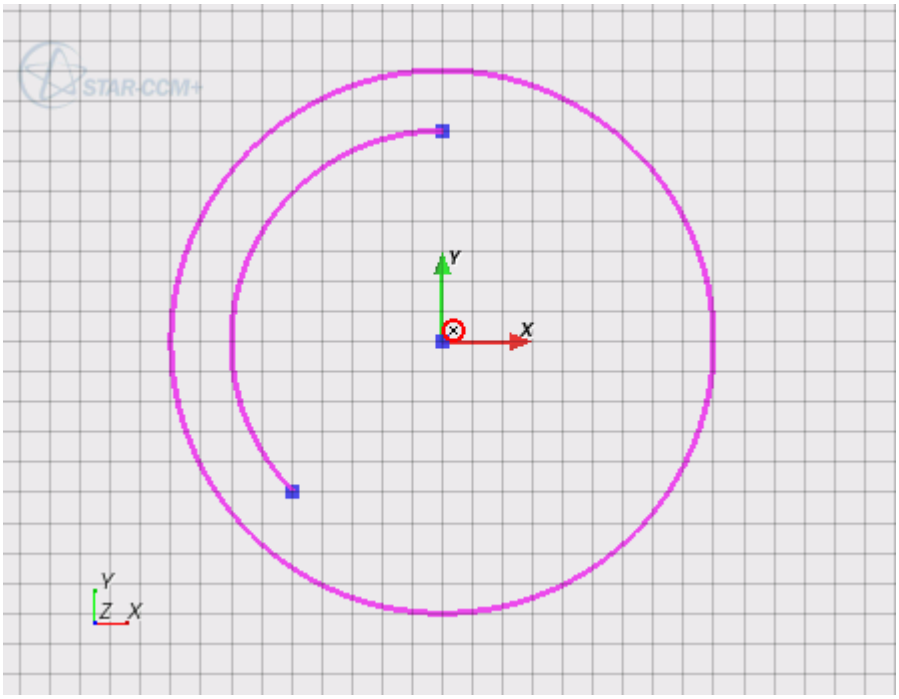
Sketches
Sketch Planes

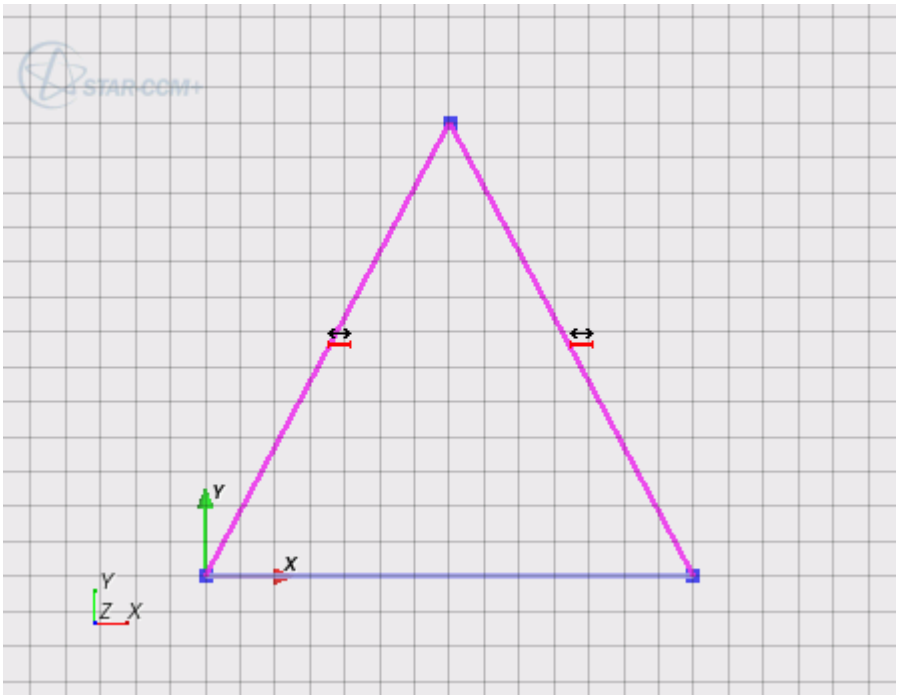
Sketch Entity Reference
Constraint Types
Removing Constraints
Dimensioning Sketches

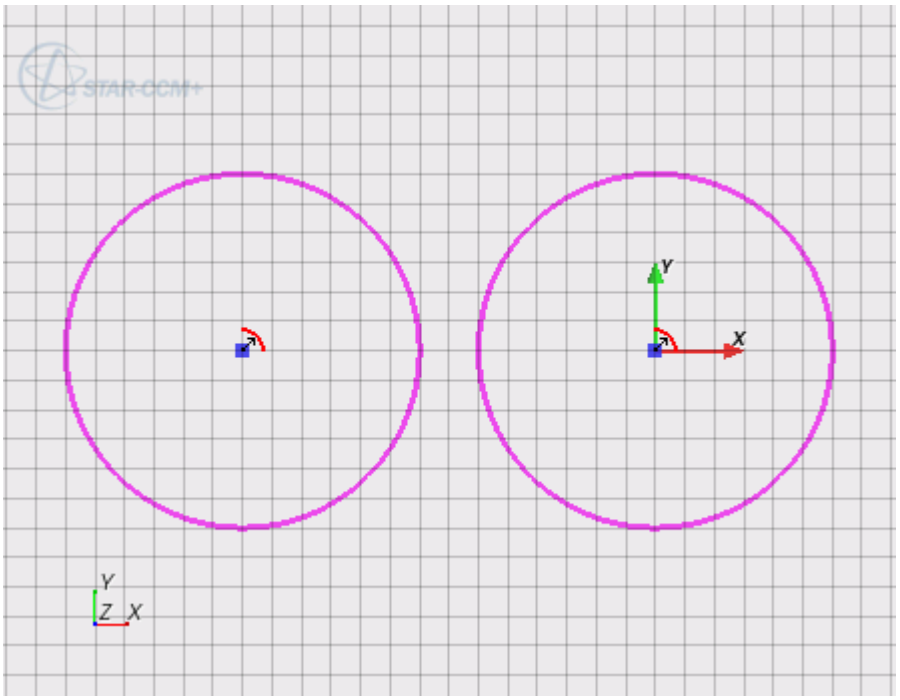
Constraint Types

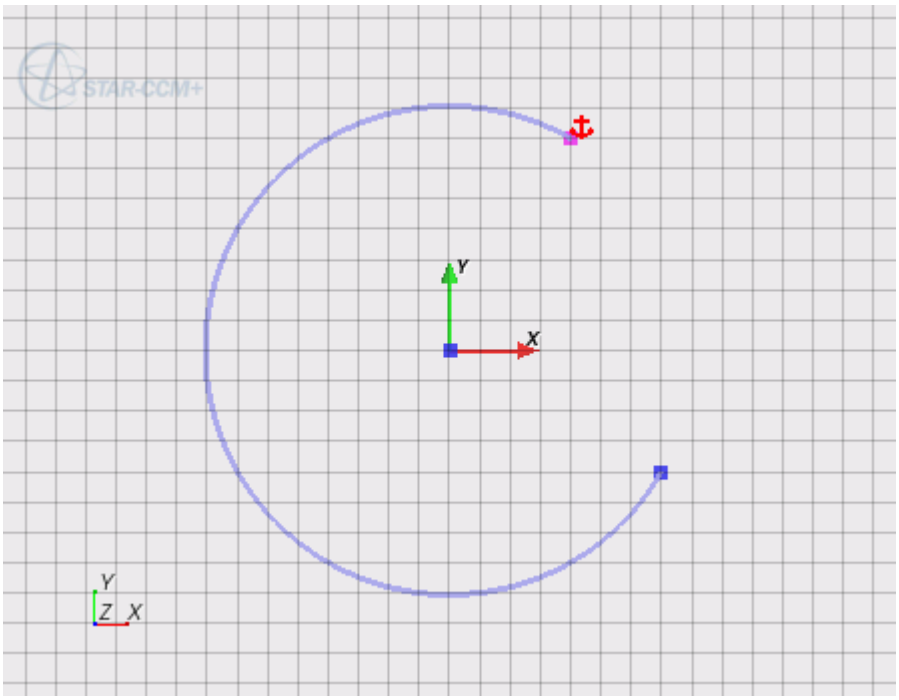
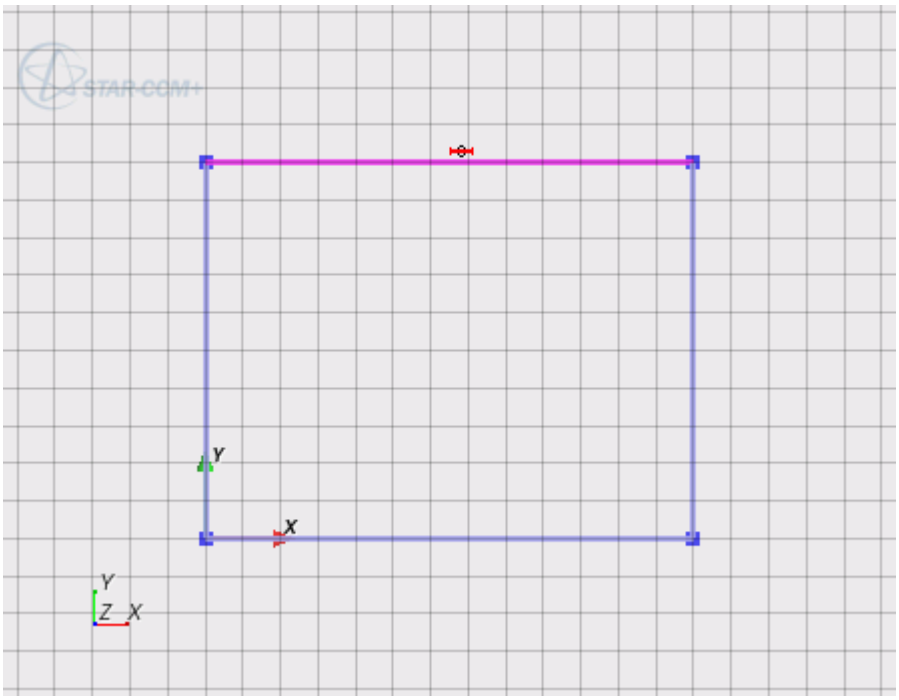
A list of available coincidence constraints is provided in the following table.

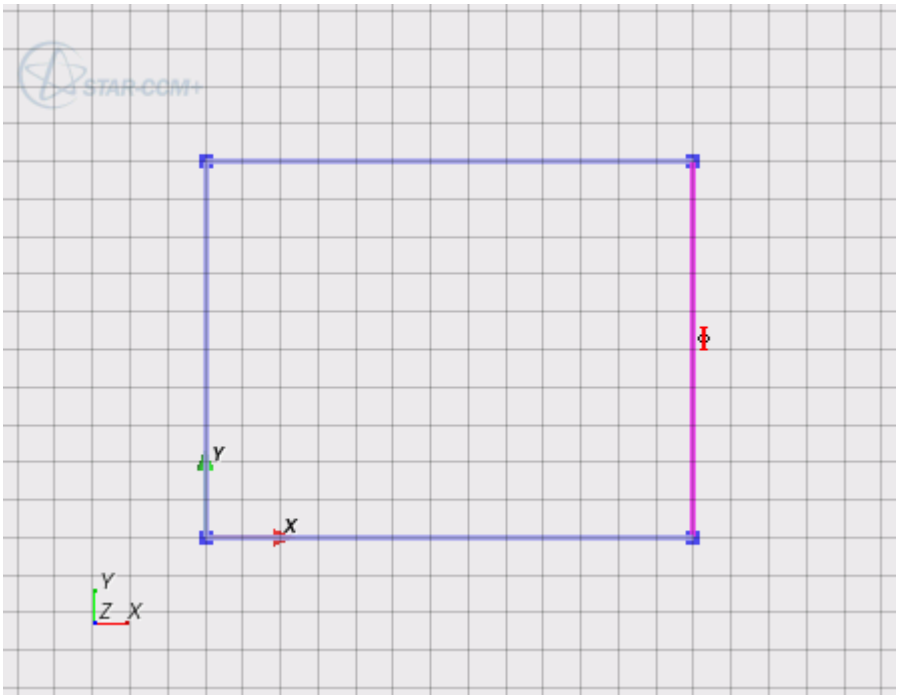
Constraint	Description
Coincidence	A coincidence constraint forces a point on a sketch entity to coincide with another sketch entity on the sketch plane. The second sketch entity can be a line, circle, circular arc, or another point. You can apply a coincidence constraint between two circular arcs, causing the arcs to share center point and radius. 
Collinear	A collinear constraint forces the selected sketch entities to lie on the same infinite line. You can only apply collinear constraints to lines. Upon applying the constraint, one line maintains its position while the second line moves to meet the requirements of the constraint. You can apply this constraint to any number of sketch entities in one go.

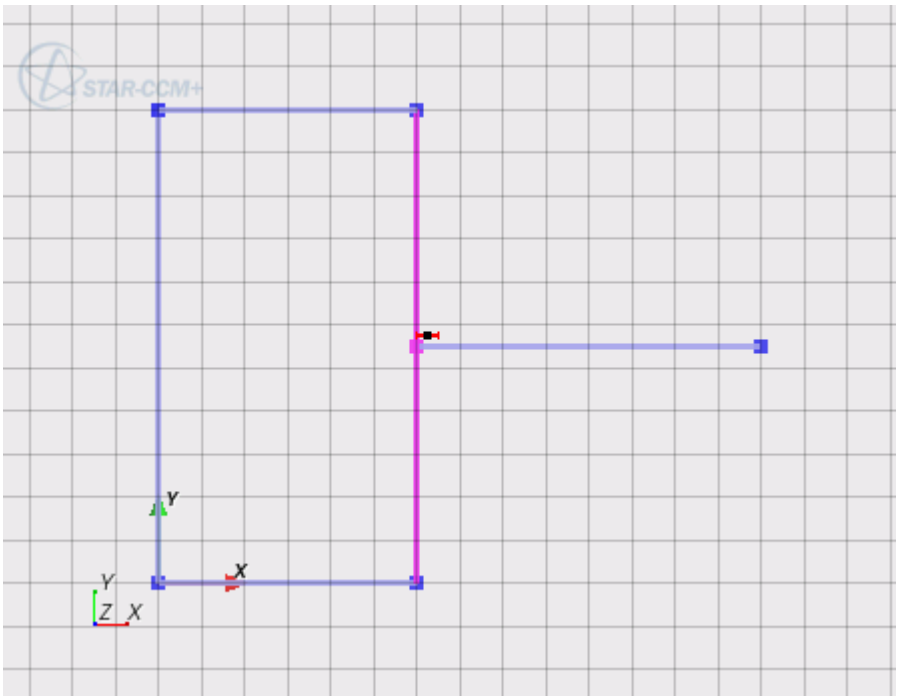
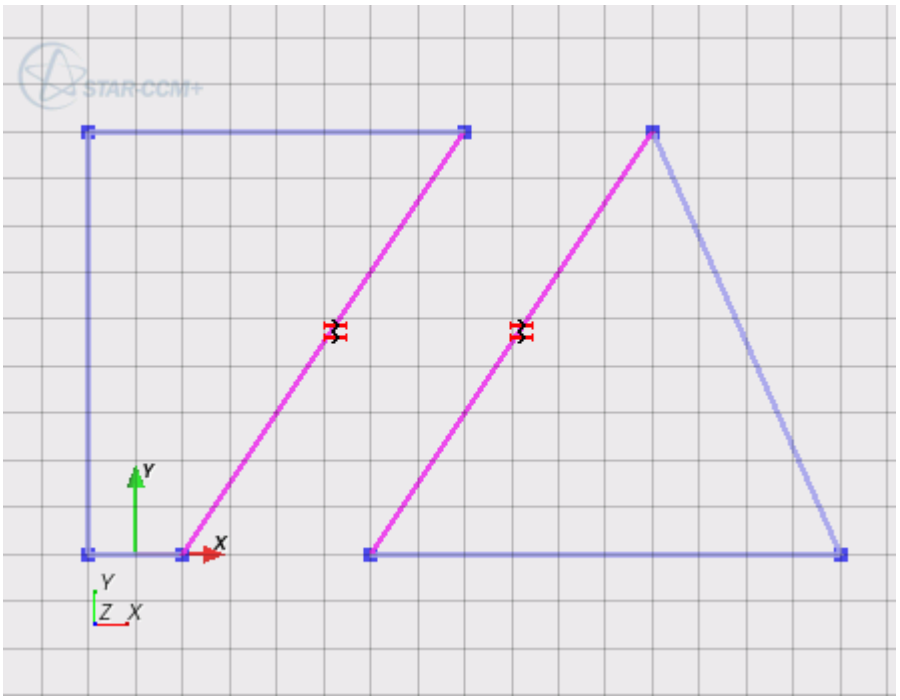
Constraint	Description
	
Concentric	A concentric constraint forces selected sketch entities to share center points. You can apply concentric constraints to circles, circular arcs, and ellipses. You can also apply concentric constraints between these sketch entities and a point, which forces the center point of the circle/arc/ellipse to coincide with the point. You can apply this constraint to any number of sketch entities in one go. 

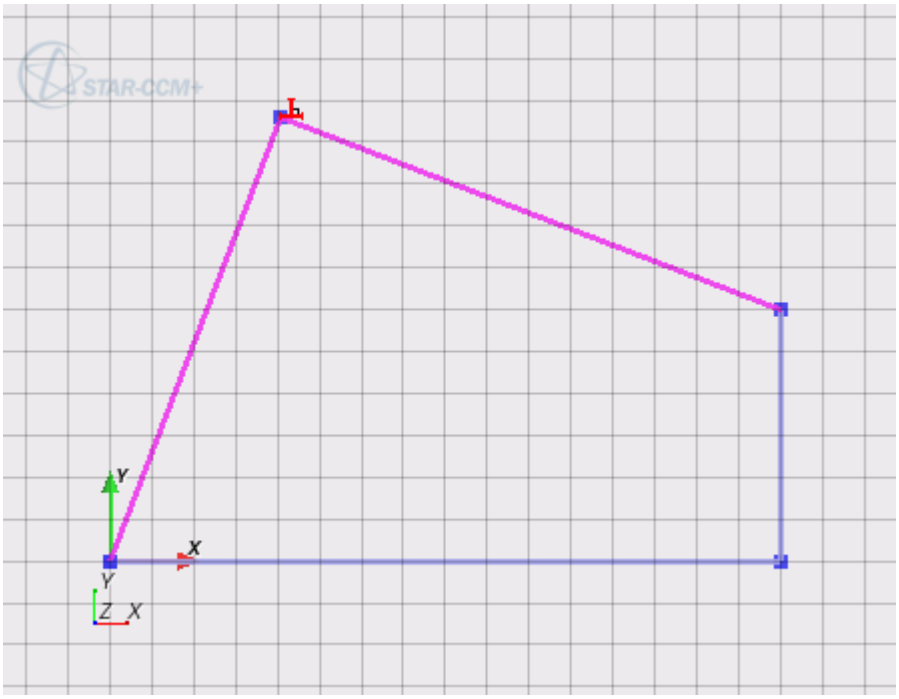
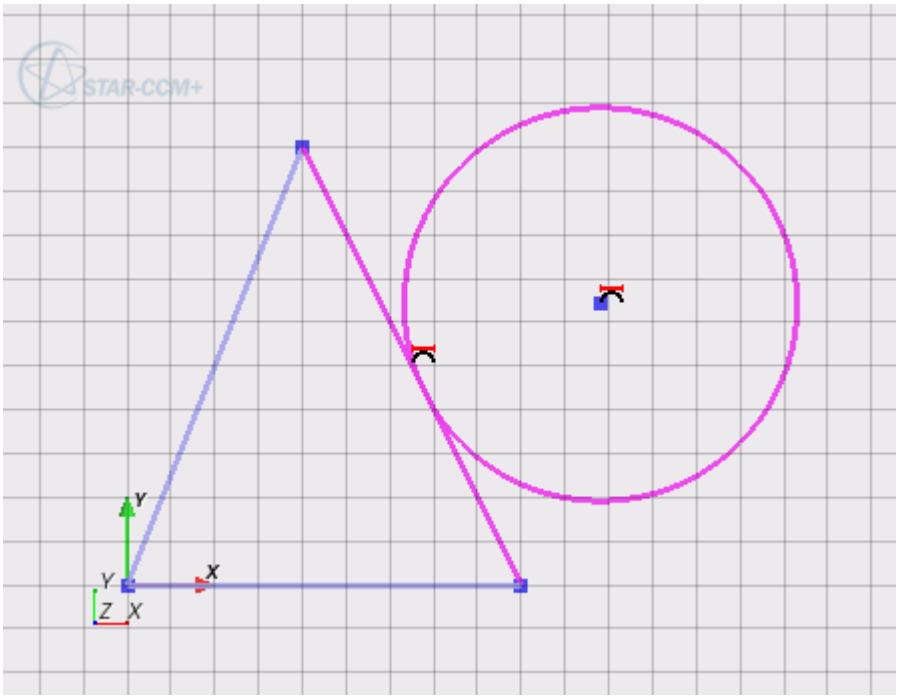
Constraint	Description
Equal Length	An equal length constraint forces the selected line sketch entities to be the same length, and can only be applied to lines. You can use equal length constraints with a length dimension, which fixes the length of both lines. You can apply this constraint to any number of sketch entities in one go. 
Equal Radius	An equal radius constraint sets the radius of selected circular sketch entities to be equal. You can apply this constraint to circle and circular arc sketch entities. To specify which circle governs the size of the radius, apply a radius dimension to that particular circle before applying the equal radius constraint. You can apply this constraint to any number of sketch entities in one go.

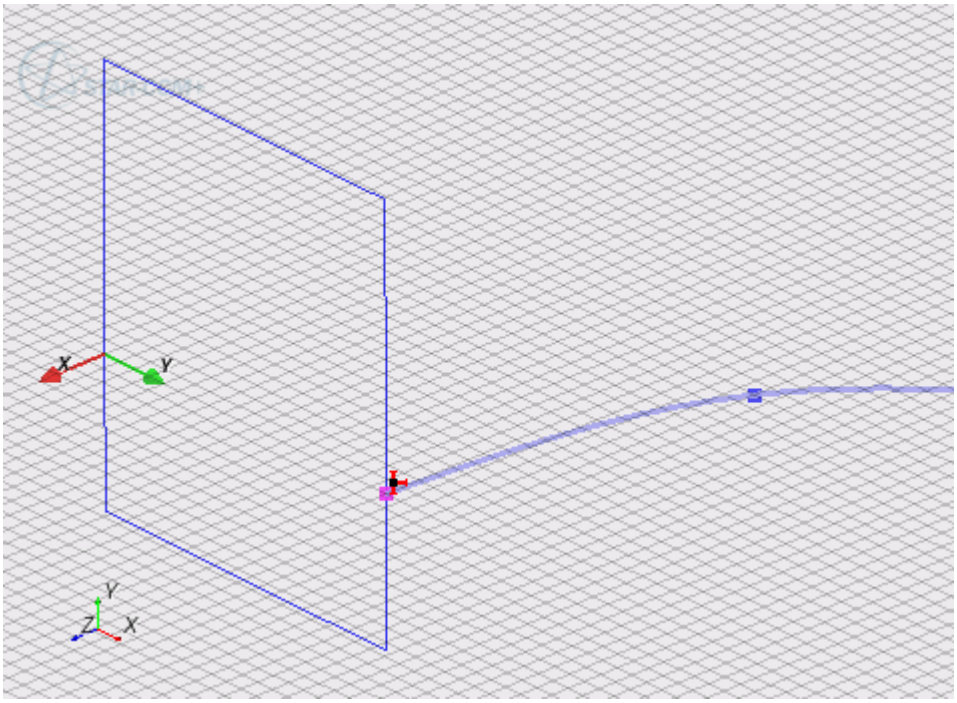
Constraint	Description
	
Fixation	A fixation constraint anchors the position of selected sketch entities on the sketch plane. You can apply fixation constraints to points, circles, circular arcs, ellipses, and spline points. When this constraint is used with a circular sketch entity, it is automatically applied to the center point. To fix the size of the circular entity, apply a radius dimension. To fix the position of a line sketch entity, apply a fixation constraint at the start and end points of the line. You can apply this constraint to any number of sketch entities in one go.

Constraint	Description
	
Horizontal	A horizontal constraint aligns a line sketch entity with the local X axis on the sketch plane. In most cases, the start point of the line is moved to satisfy the constraint, but this behavior depends on which constraints and dimensions have already been applied to the sketch. The length of the line can change when you apply the constraint (unless you apply a length dimension). You can apply this constraint to any number of sketch entities in one go. 

Constraint	Description
Vertical	A vertical constraint aligns a line sketch entity with the local Y axis on the sketch plane. In most cases, the start point of the line moves to satisfy the constraint, but this behavior depends on which constraints and dimensions are already applied to the sketch. The length of the line can change when you apply the constraint (unless you apply a length dimension). You can apply this constraint to any number of sketch entities in one go. 
Midpoint	A midpoint constraint forces a point sketch entity to lie on the midpoint of a line. You can apply midpoint constraints between a point and a line sketch entity.

Constraint	Description
	
Parallel	A parallel constraint forces selected line sketch entities to be parallel. You cannot apply a parallel constraint to two lines that are at right angles to each other. You can apply this constraint to any number of sketch entities in one go. 

Constraint	Description
Perpendicular	A perpendicular constraint forces two line sketch entities to be aligned at right angles to each other. You cannot apply a perpendicular constraint to two parallel lines. 
Tangent	A tangent constraint forces an edge (of a circular, elliptical, or line sketch entity) to be tangential to a selected circle, circular arc, or ellipse. You can apply this constraint to circle, circular arc, ellipse, or line sketch entities. 

Constraint	Description
Pierce	When two sketch planes intersect, you can apply a pierce constraint between a point sketch entity (from the active sketch) and an edge sketch entity (from the reference sketch). This type of constraint forces the point sketch entity to pierce the edge sketch entity. This feature is useful when creating sweeps and guided lofts, as it ensures coincidence between the guide sketch and profile sketch.  In the above example, the two sketch entities are on different planes that are perpendicular to each other. This method would also work for any two sketches that have intersecting planes. If two locations exist where the point can pierce the reference sketch, the edge closest to the point is pierced.

See Also:

Dimension Types

Sketch Entity Reference

Swept Features

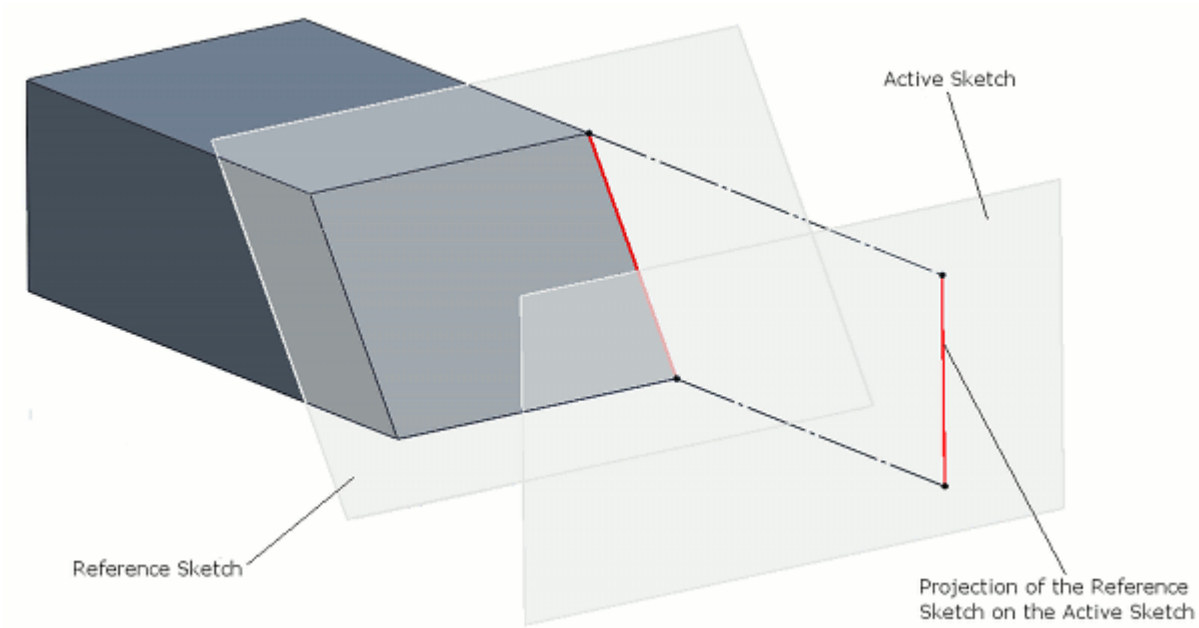
Using Guide Wires in Loft Operations

Applying Constraints that Reference Other Sketches or Existing Geometries

You can select an edge from another sketch or geometry when creating a constraint in the active sketch.

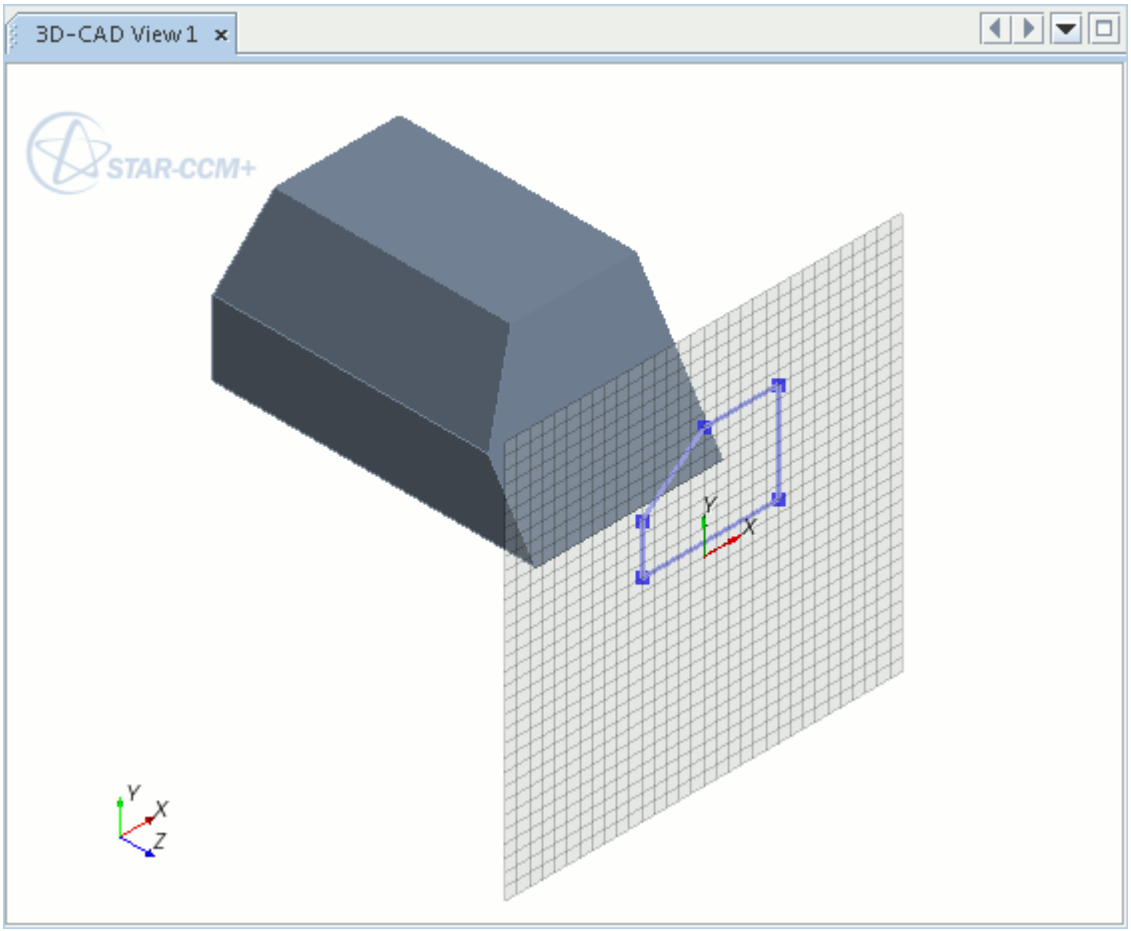
The selected edge is projected onto the active sketch and is used to constrain the target entity in the active sketch. If the plane of the active sketch intersects the plane of the reference sketch, you can use the pierce constraint. See Pierce Constraint. This constraint is only available when selecting a point from the active sketch and an edge from the reference sketch, and ensures that the two sketch entities are coincident.

An example demonstrating how the projection works is shown below.



This feature gives you the ability to align sketch entities with one or more existing sketches. The same is also true for reference geometries, where the selected edge on the reference geometry is projected onto the active sketch. You can only use edges from a reference sketch/geometry to constrain an active sketch. You cannot use vertices from the reference sketch/geometry. When you modify a reference entity, the projection of that entity on the constrained sketch is also modified. This action can change the constrained sketch. If you delete a reference entity, the constraint that is based on that projection is also deleted. Using a similar method, you can pierce another sketch on a different plane, ensuring that the active sketch and the reference sketch are coincident.

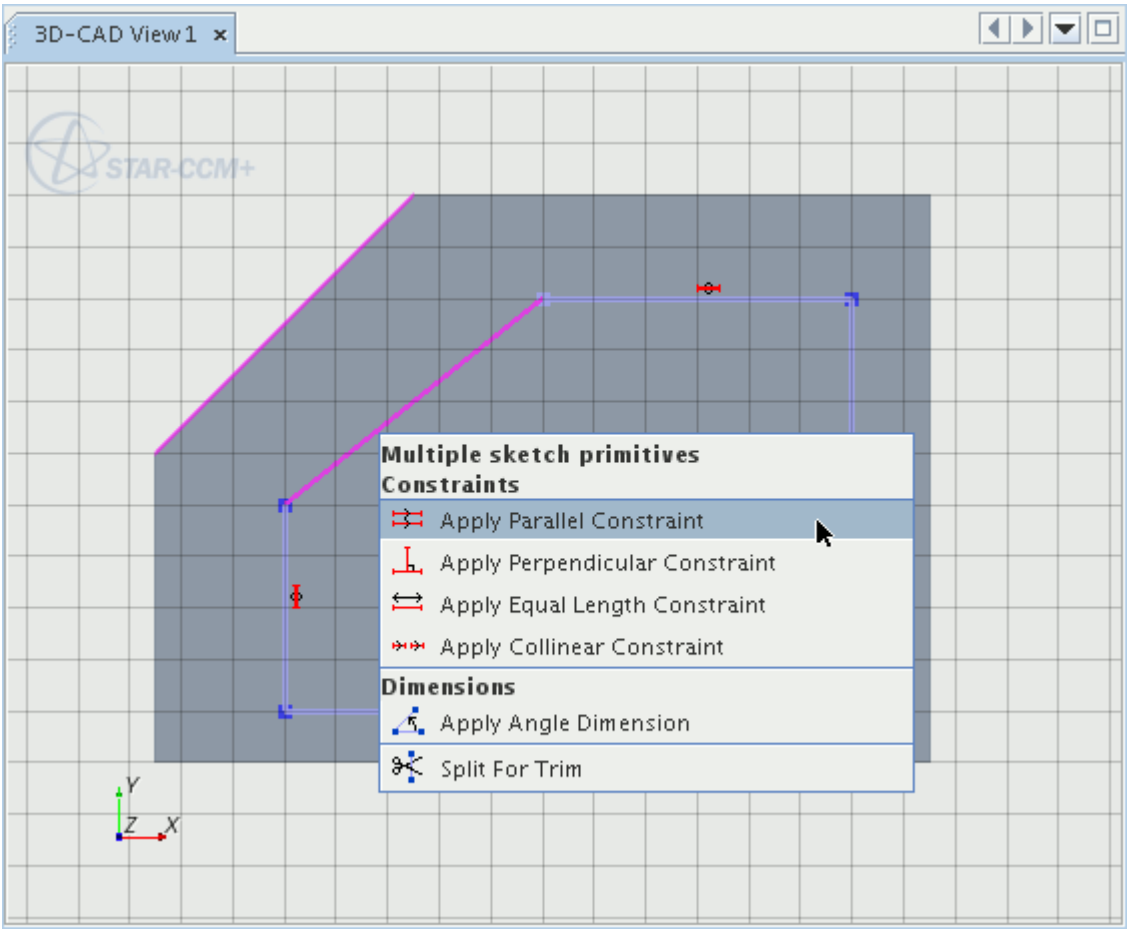
In the following example, the gray geometry is used to constrain the active sketch (shown in blue).



To constrain the active sketch using a reference geometry:

1. Select the edge that you want to use as a reference from the reference geometry. This edge highlights in purple.

2. While holding down the `<Ctrl>` key, right-click the desired entity on the active sketch and select the required constraint from the right-click menu. In this case, the line that is on a diagonal must be parallel to the selected edge of the reference geometry, so **Apply Parallel Constraint**is selected.



The diagonal edge from the reference geometry is projected to the active sketch and the constraint operation is carried out.

Note: The projection does not appear on the active sketch. This operation is internal.

A symbol denoting the type of constraint is placed on the constrained sketch entity.

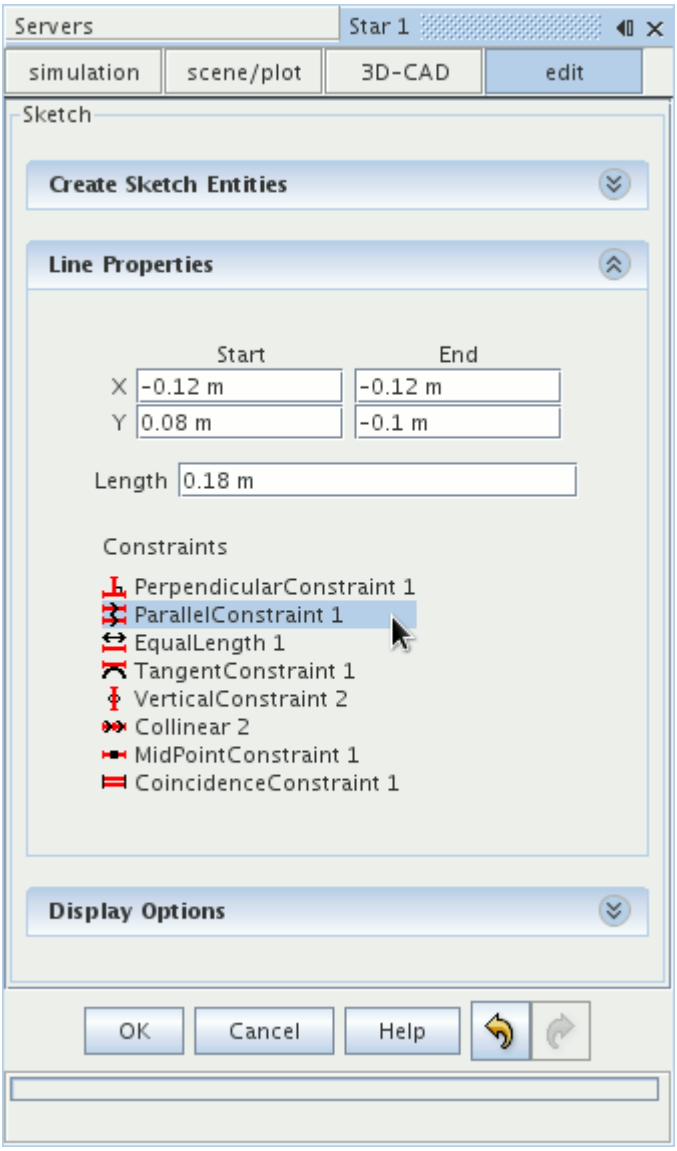
The two diagonal lines are now parallel to each other.

Reviewing Applied Constraints

When constraints have been used in a sketch, glyphs are displayed next to the constrained sketch entities. By selecting a constraint glyph in the *3D-CAD View* scene, you can see which entities the constraint applies to, and also delete the constraint. Constraints can be hidden from the sketch scene.

To review constraints that are applied to a particular sketch entity:

1. Select the sketch entity in the *3D-CAD View* scene. A list applied constraints are displayed in the *Properties* box in the *Sketch* panel.
2. Select a constraint from this list. All sketch entities that the constraint applies to become highlighted in the *3D-CAD View* scene.



You can use this list to remove a constraint from the sketch.

See Also:

Removing Constraints

Showing/Hiding Relations

Removing Constraints

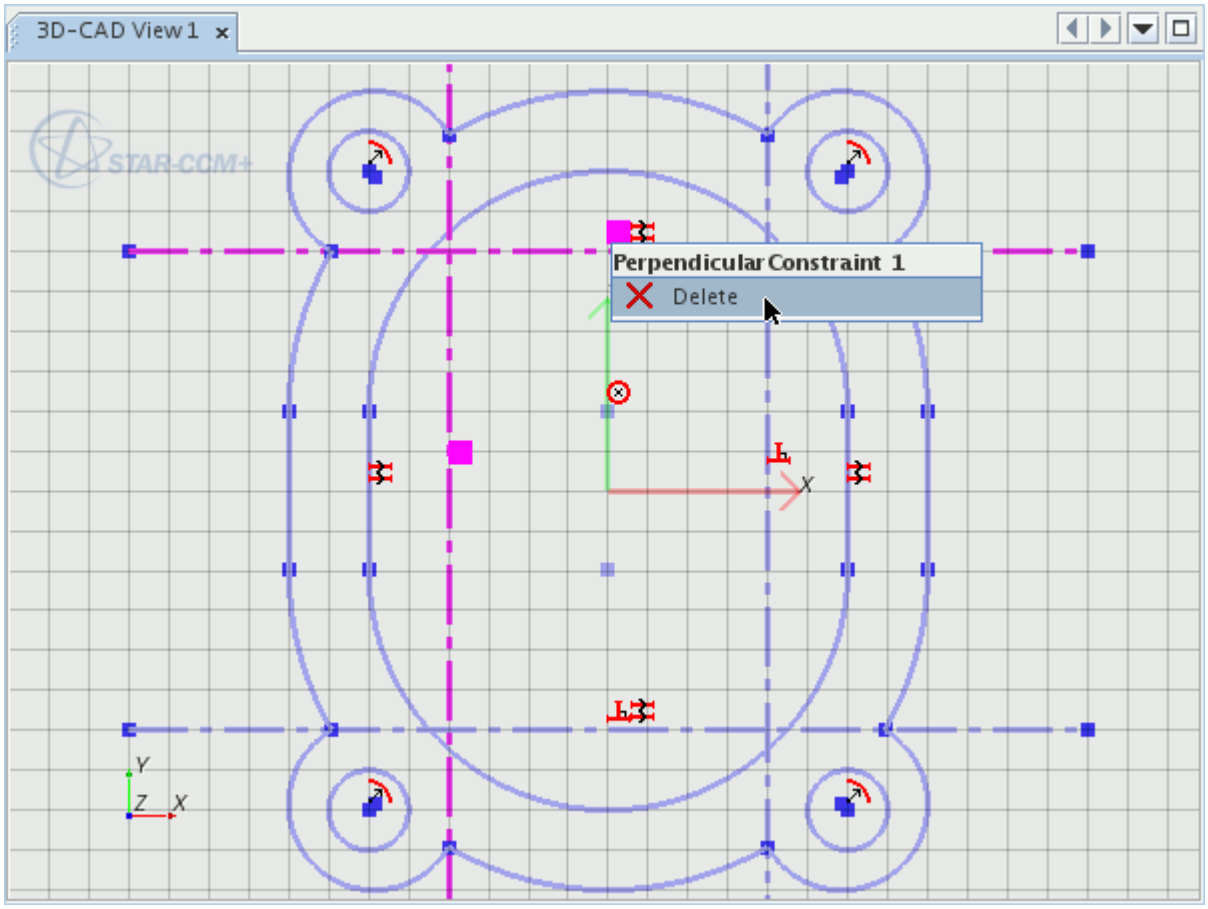
You can remove constraints from a sketch in two ways:

- By selecting the constraint glyph in the *3D-CAD View* scene.
- By using the constraints list in the *Sketch* panel.

Removing Constraints From the 3D-CAD View Scene

To remove a constraint directly in the *3D-CAD View* scene:

- Right-click on the constraint glyph in the *3D-CAD View* scene, and select **Delete**.

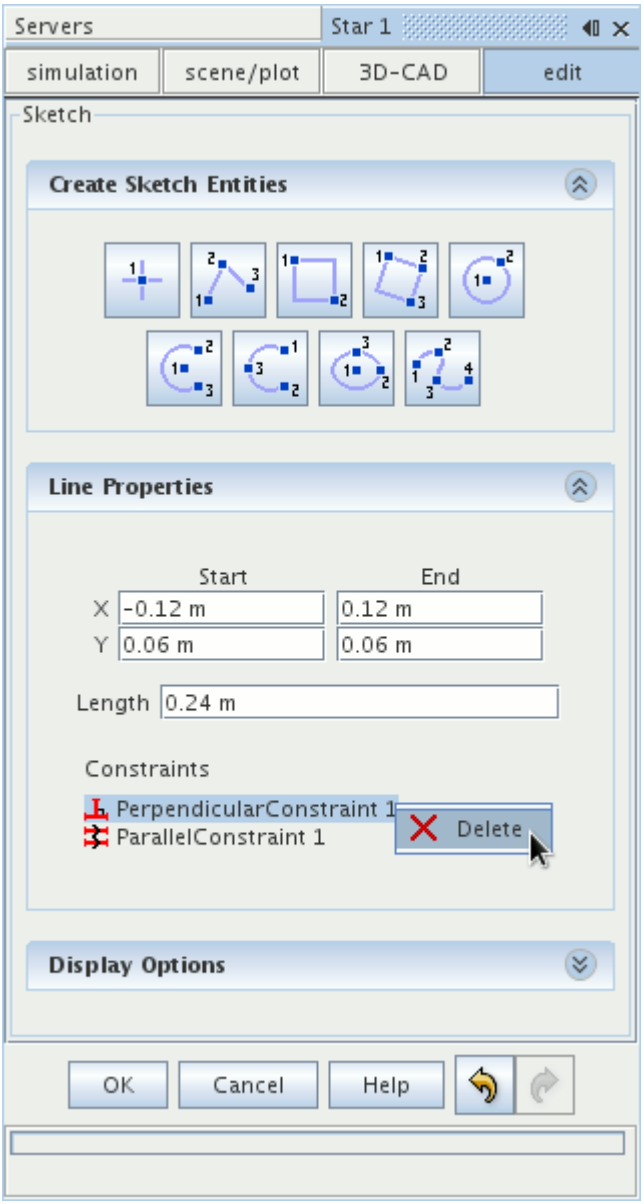


- Alternatively, select the constraint glyph and press the `<Delete>` key. Only select one of the glyphs where a constraint is applied to a pair of sketch entities.

Removing Constraints From the List of Constraints

To remove a constraint using the list of applied constraints in the *Sketch* panel:

1. Select the sketch entity in the *3D-CAD View* scene, from which you wish to remove a constraint. In the *Sketch* panel, a list of constraints that are applied to the selected sketch entity is displayed.
2. Right-click on the constraint you wish to remove and select **Delete**.



See Also:

Constraining a Sketch

Reviewing Applied Constraints

Dimensioning Sketches

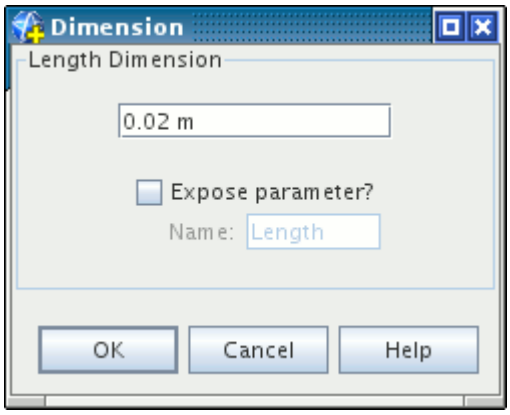
Dimensions are used to control sizes, distances, and angles in a sketch.

When you apply a dimension to one or more sketch entities, it fixes the size, distance, or angle to a specific value. The value can be a numerical quantity, or specified using design parameters, and can only be changed by editing the dimension.

Caution: There is a numerical limit to the geometrical precision of 3D-CAD models. Do not apply dimensions less that 1 micron (1.0E-6 m).

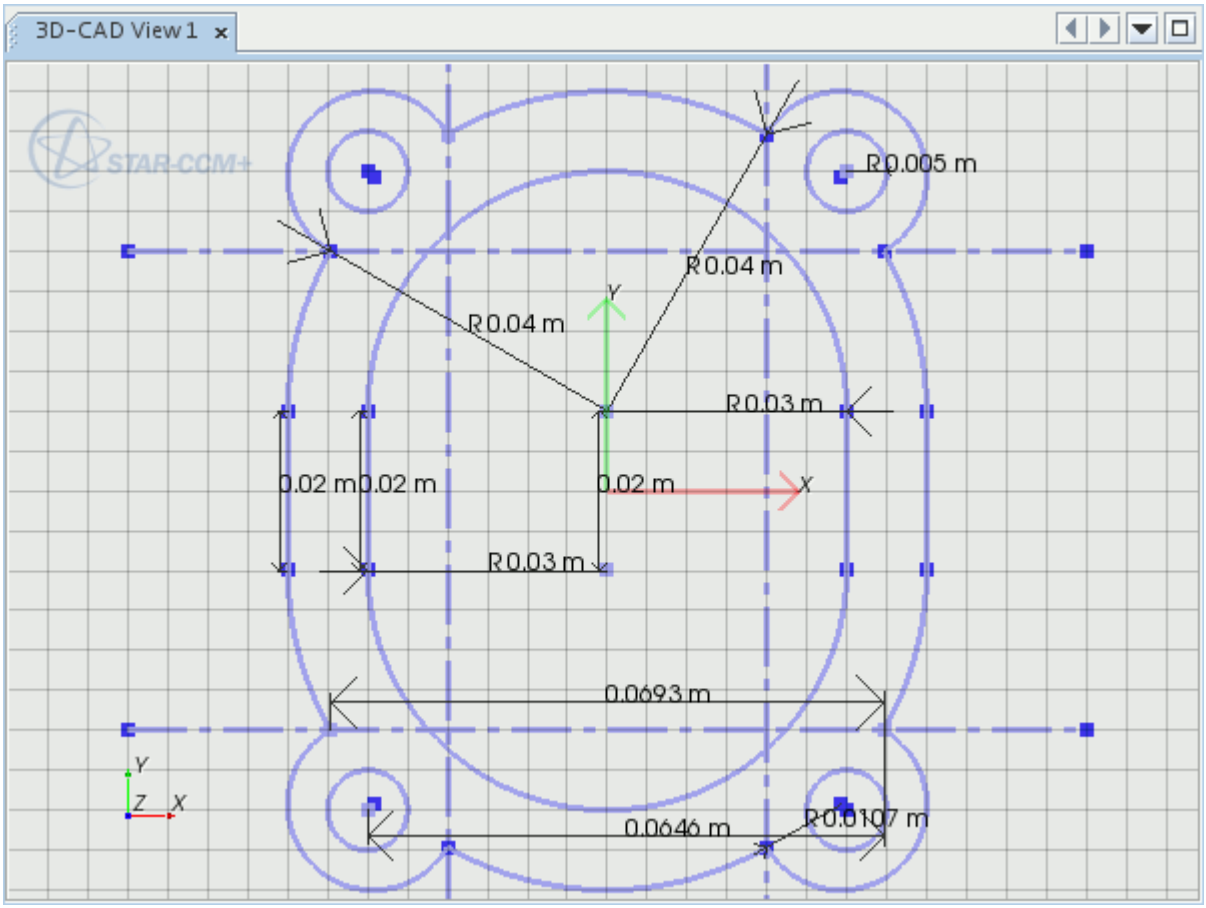
To dimension a sketch:

1. In the 3D-CAD view scene, right-click on the desired sketch entities and select the type of dimension that you want to apply. See Dimension Types.
2. Enter the dimension values in the *Dimension* dialog. You can specify the units by typing them directly in the textbox.



3. To expose the dimension as a design parameter, activate *Expose parameter?*
4. Click **OK**.

If necessary, you can hide dimensions in the sketch to give you a clearer view of the sketch geometry. Alternatively, you can reposition the dimension.



Contents:

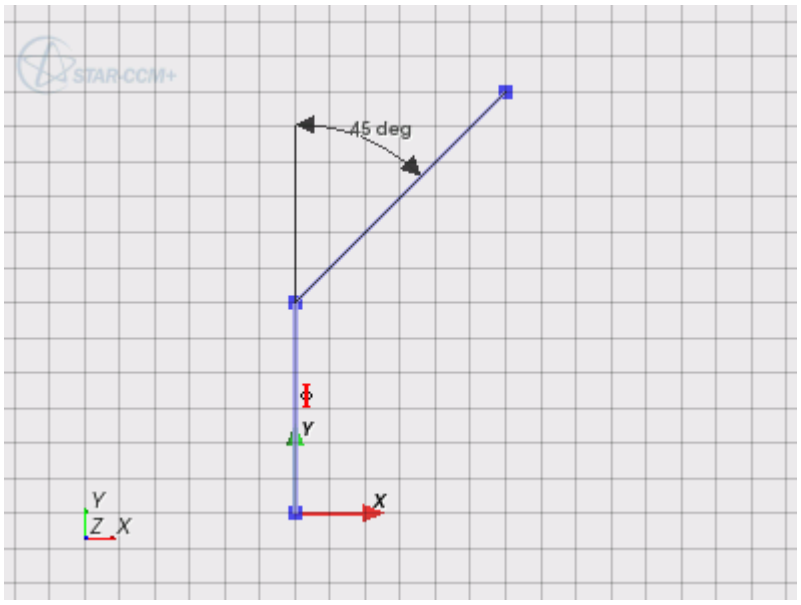
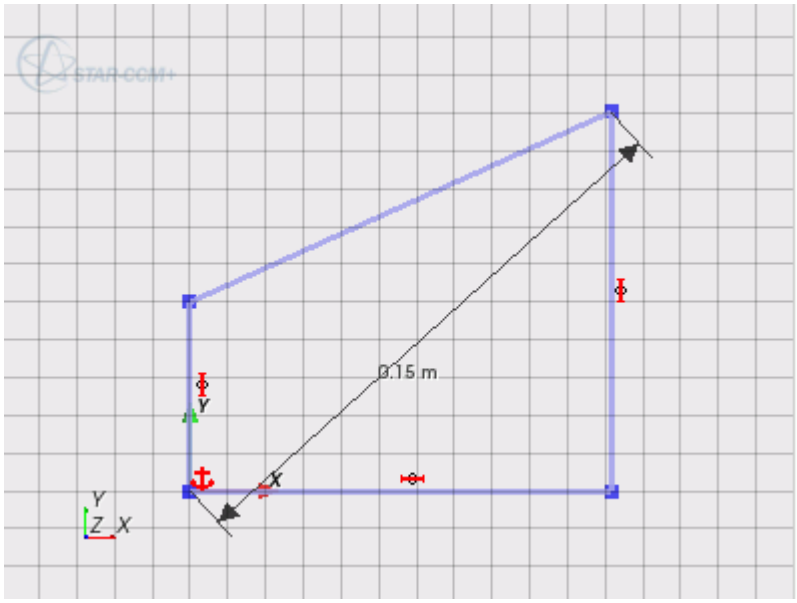
Dimension Types
Reviewing Applied Dimensions
Editing the Value of Dimensions
Repositioning Dimensions
Removing Dimensions

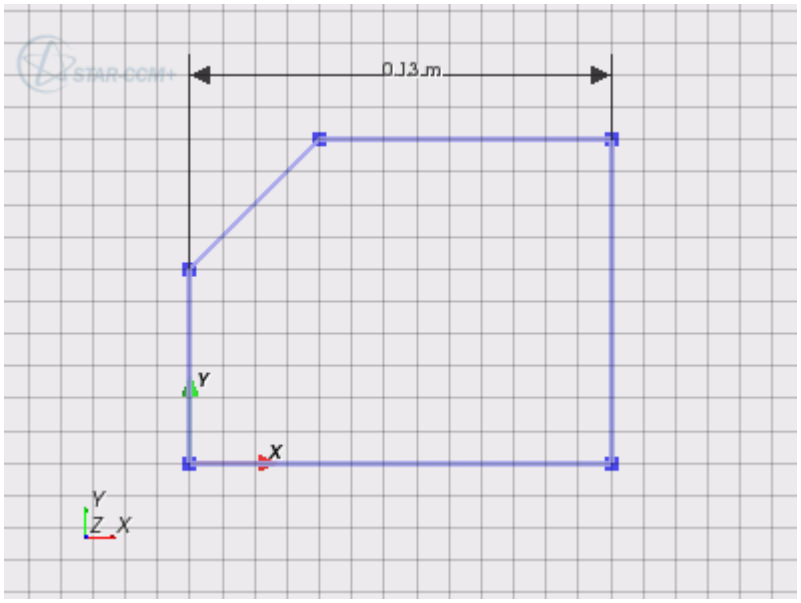
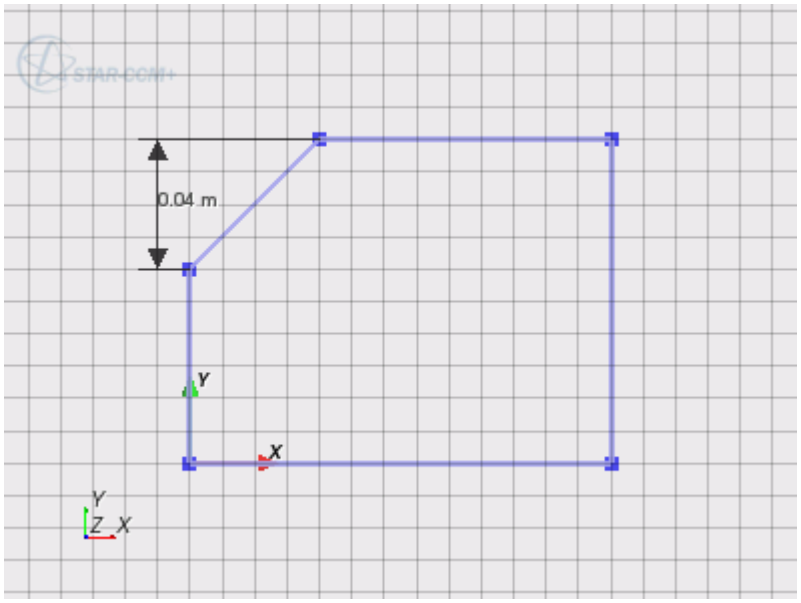
See Also:

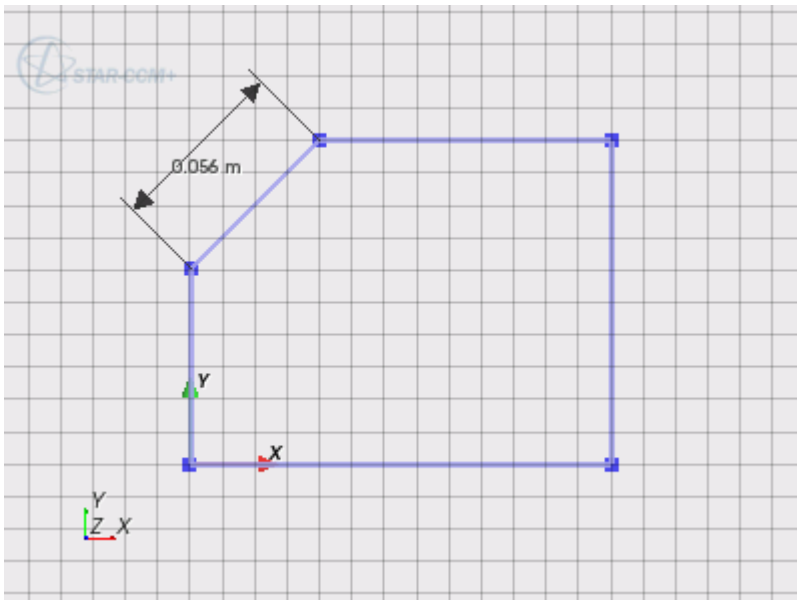
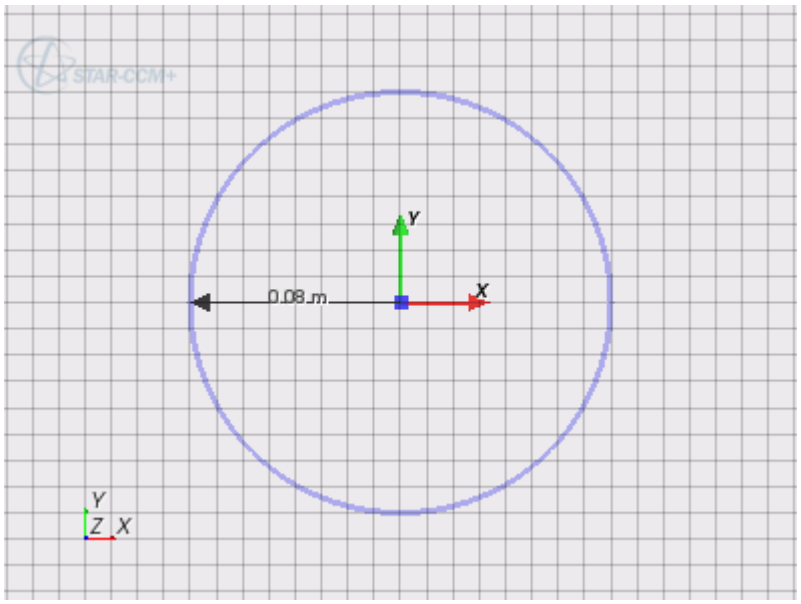
Sketches
Working with Design Parameters
Showing/Hiding Relations

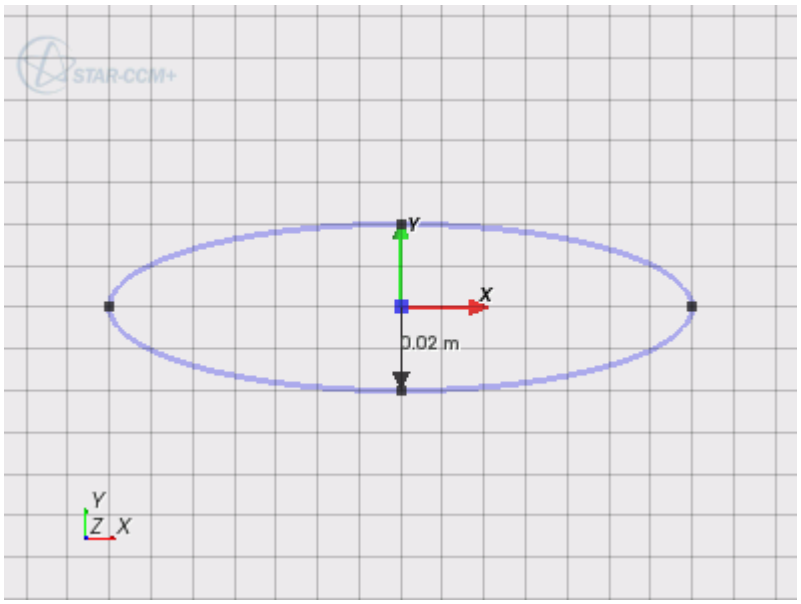
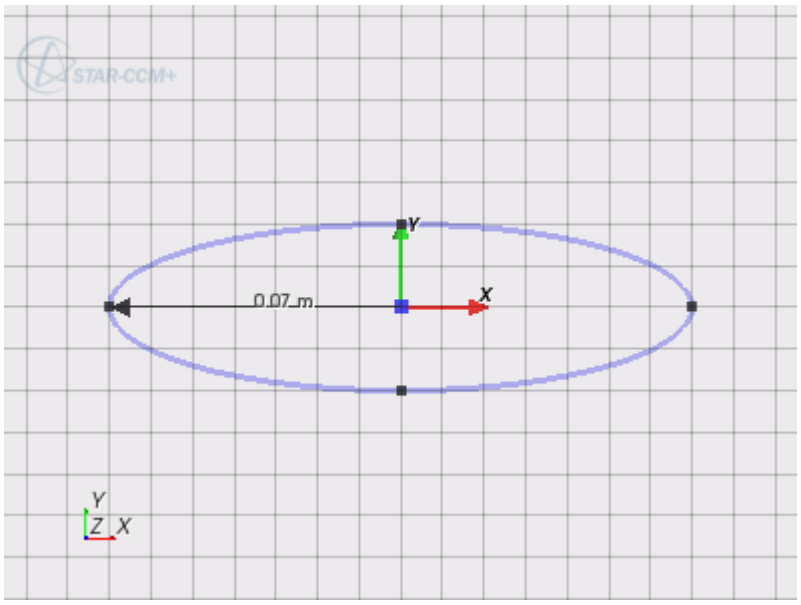
Dimension Types

A list of available dimensions, with a short description, is provided in the following table. For information on how to apply dimensions, see Dimensioning Sketches.

Dimension	Description
Angle	An angle dimension specifies the angle between two line sketch entities. The acute angle is always displayed. If the angle between the lines is obtuse, STAR-CCM+ displays the acute angle between one of the lines and an extension of the other. The tolerance for an angle dimension is `0.001` radians, which is approximately `0.057` degrees. 
Distance	A distance dimension sets the straight-line distance between two selected sketch entities. You can apply this dimension between a point and a line, between two points, or between two lines. 

Dimension	Description
Horizontal Distance	A horizontal dimension sets the distance in the X-direction between two points. 
Vertical Distance	A vertical dimension sets the distance in the Y-direction between two points. 
Length	A length dimension sets the length of a line sketch entity.

Dimension	Description
	
Radius	A radius dimension sets the radial size of a circle or circular arc sketch entity. 
Minor Radius	The minor radius dimension sets the small diameter of an ellipse sketch entity.

Dimension	Description
	
Major Radius	The major radius dimension sets large diameter of an ellipse sketch entity. 

See Also:

Sketch Entity Reference

Reviewing Applied Dimensions

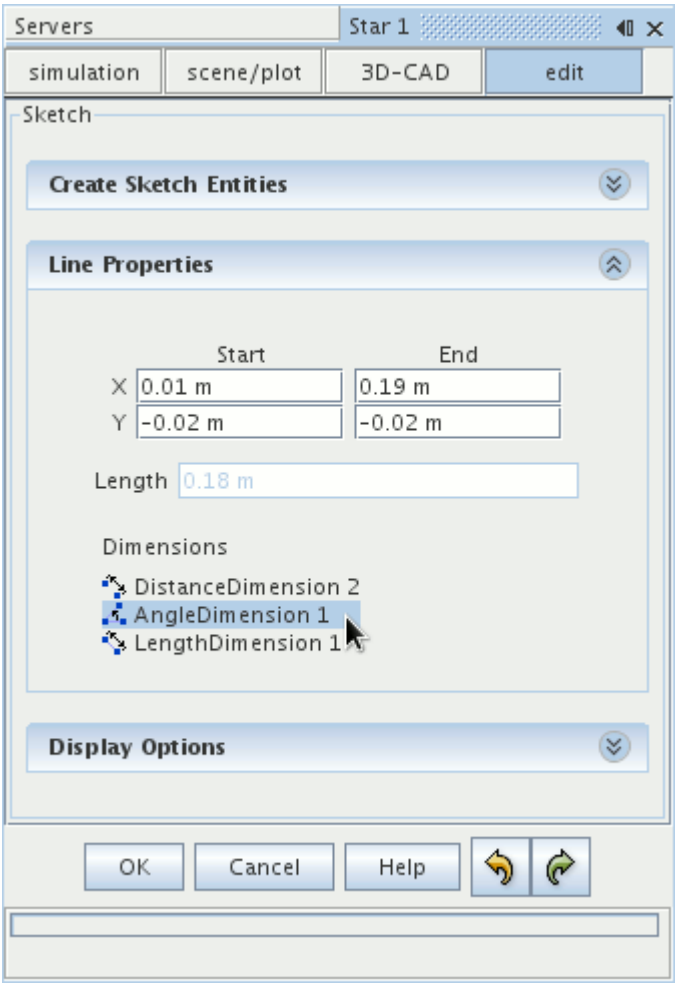
You can review sketch dimensions by displaying them in the *3D-CAD View* scene.

If you exposed dimensions as a design parameter, an expression, such as `Length = 0.01 m`, is shown next to the dimension arrow. If you defined a dimension by an expression, such as `2*$Length`, the expression is displayed next to the dimension arrow. You can select dimensions that are displayed in the *3D-CAD View* scene and edit or delete them as necessary.

To view a list of dimensions that are applied to a particular sketch entity:

1. Select a sketch entity in the *3D-CAD View* scene.

   The list appears in the *Properties* box in the *Sketch* panel.



2. Select a dimension from this list.

   The relevant sketch entities become highlighted in the *3D-CAD view* scene. You can also use this list to edit or delete a dimension, as necessary.

See Also:

Showing/Hiding Relations

What Is a Design Parameter?

Editing the Value of Dimensions

You can edit the value of a dimension in two ways:

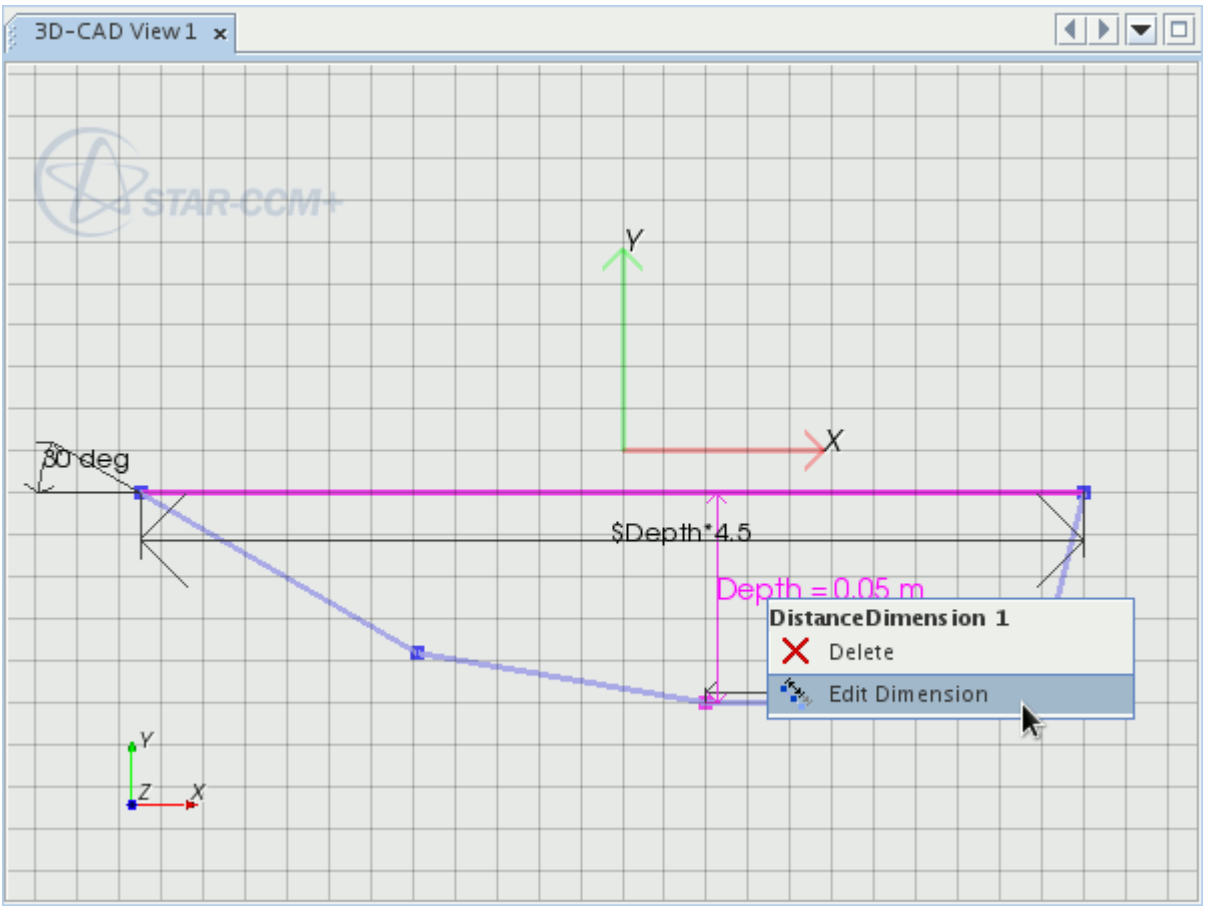
- By selecting the dimension in the *3D-CAD View* scene.
- From the Dimensions list in the *Sketch* panel.

Either of these methods opens the *Dimension* dialog, where you can enter a new numerical value or expression in the textbox. Editing the dimension also allows you to expose the dimension as a design parameter, or conceal a previously exposed design parameter.

Editing Dimensions Using the 3D-CAD View Scene

To edit a dimension directly in the *3D-CAD View* scene:

1. Right-click on the dimension marker in the sketch and select **Edit Dimension**.



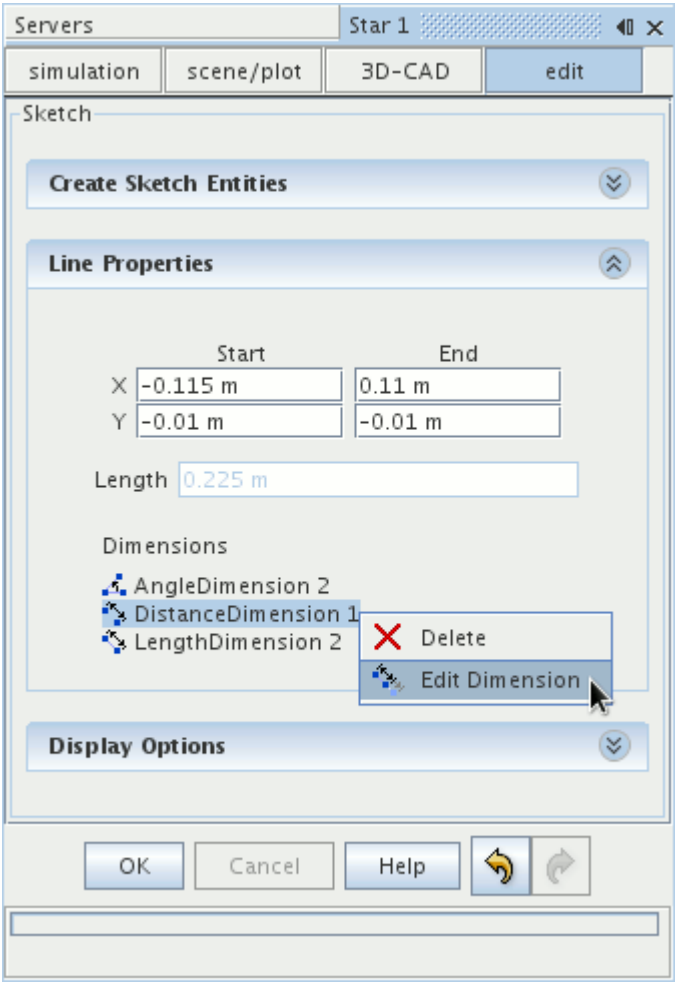
The *Dimension* dialog appears.

1. Enter the new value.
2. Click **OK**.

Editing Dimensions Using the Sketch Panel

To edit a dimension using the list of applied dimensions in the *Sketch* panel:

1. Select the sketch entity to which you applied the dimension. In the *Sketch* panel, a list of dimensions that are applied to the selected sketch entity is displayed in the *Properties* box.
2. Right-click on the dimension that you want to edit and select **Edit Dimension**.



The *Dimension* dialog appears.

1. Enter the new value.
2. Click **OK**.

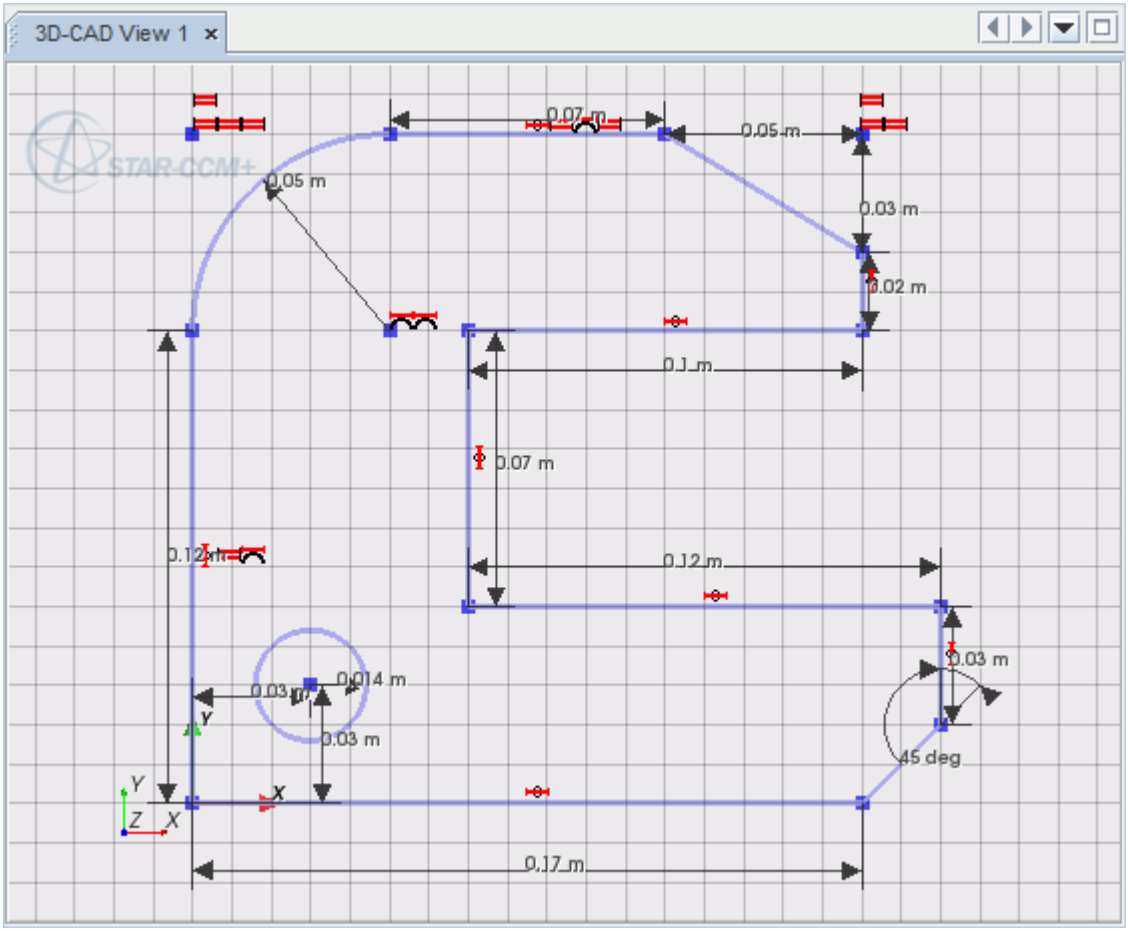
See Also:

What Is a Design Parameter?

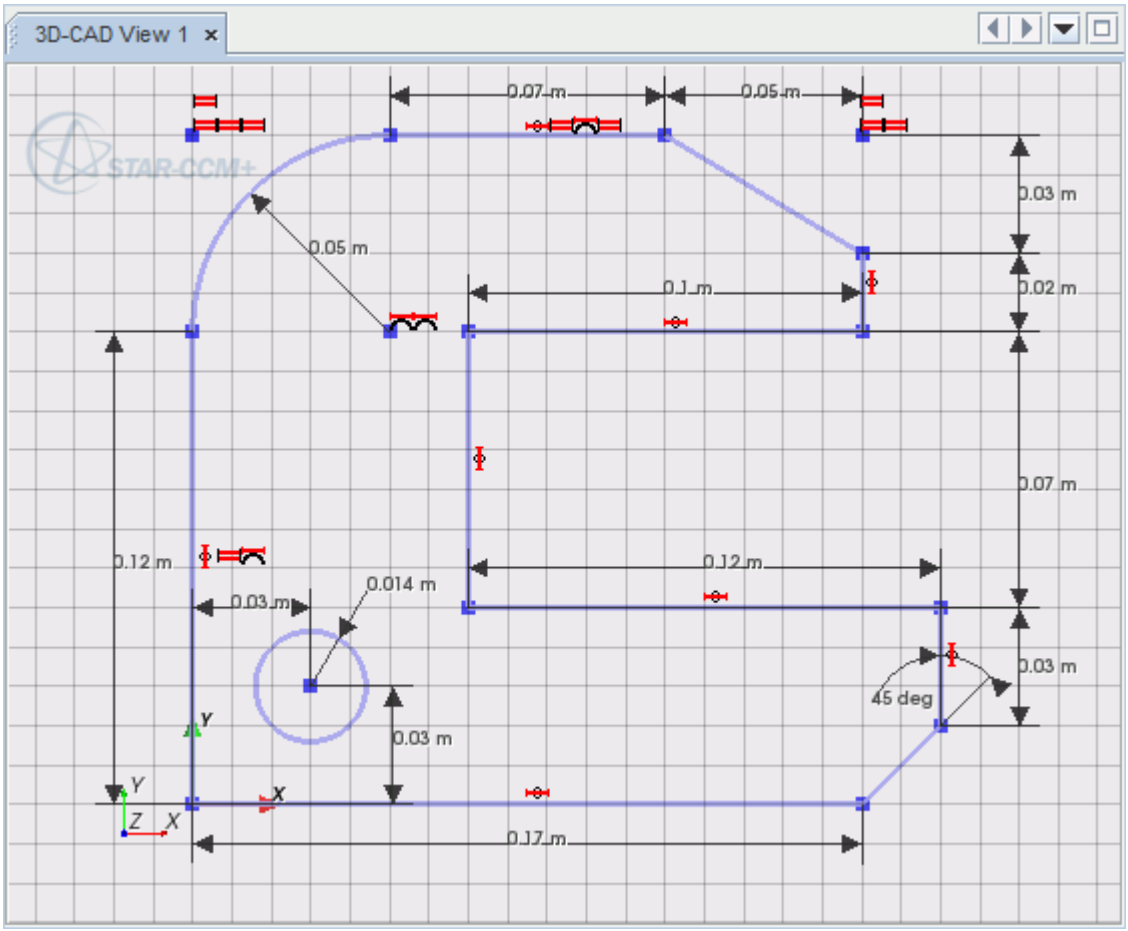
Repositioning Dimensions

You can reposition sketch dimensions in the sketch scene.

This feature is useful for moving dimensions that are blocking the view of the sketch entities and sketch constraints, or for improving the presentation of the sketch. The following screenshot shows a cluttered sketch.



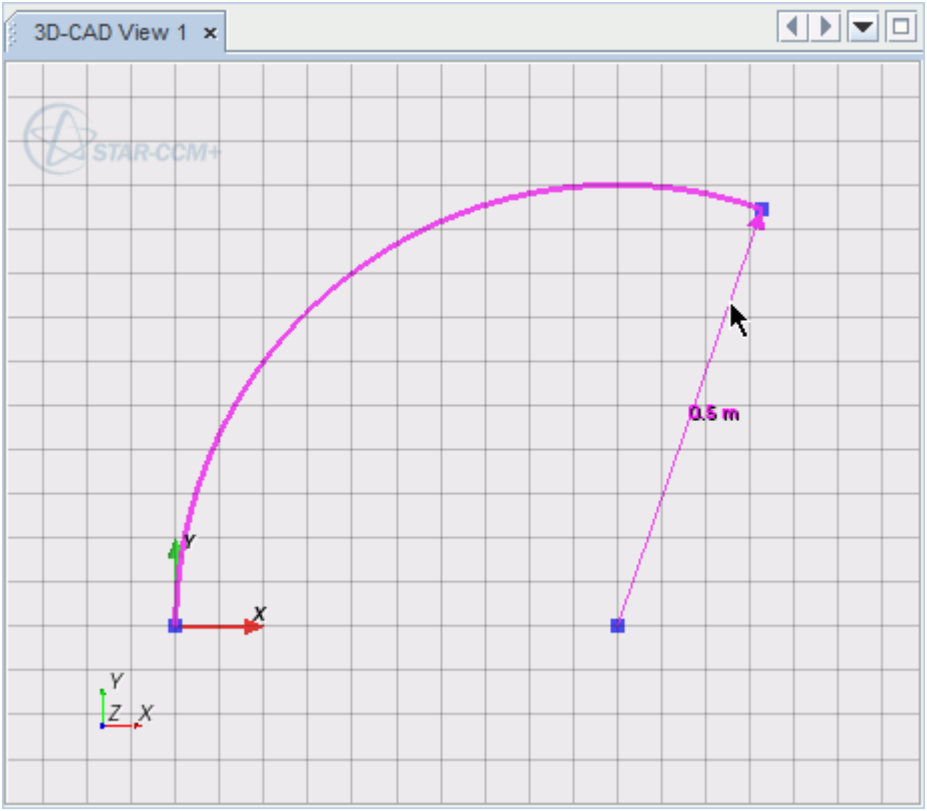
Rearranging the dimensions makes it easier for you to work with the sketch.



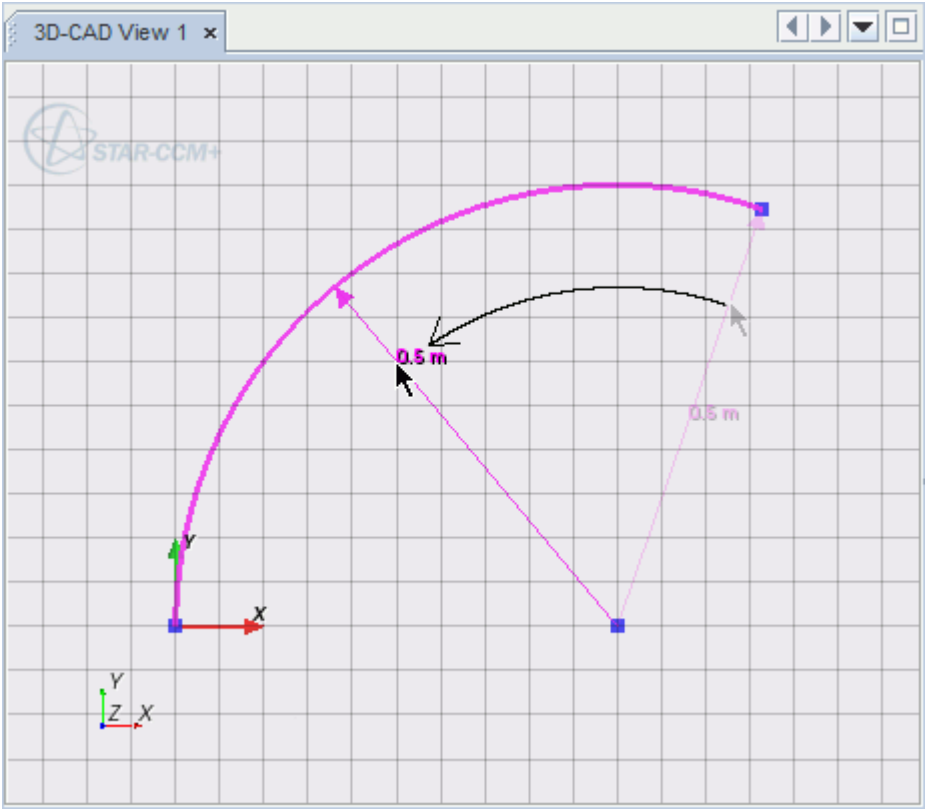
You can move all dimension types in 3D-CAD.

To move a dimension:

1. Click any part of the dimension in the `3D-CAD View` window. The selected dimension, and the sketch entities that correspond to it, are highlighted in the sketch scene.

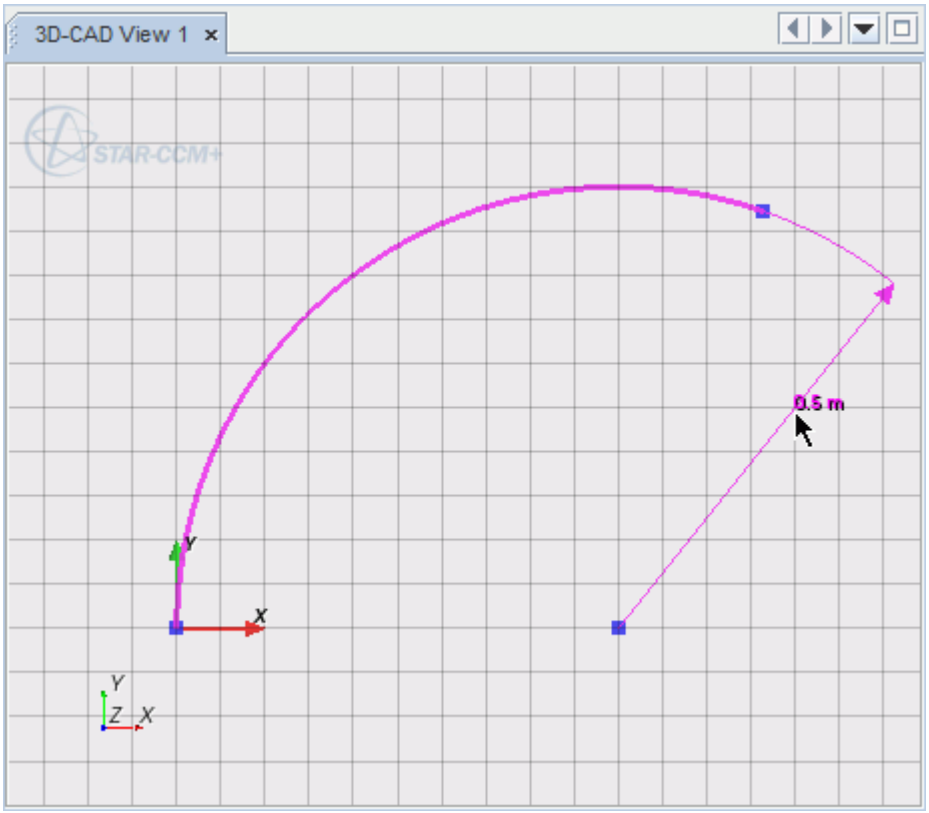


2. Drag the dimension to its new position. The dimension label snaps to your mouse pointer and follows it until you release the mouse button.



While moving the mouse pointer, the dimension label also snaps to each grid intersection.

If the dimension label is moved beyond the extent of the sketch entity, a projection appears.



Angle dimensions are projected in the same way as radius dimensions.

3D-CAD View 1
STAR-CCM+
45 deg
Y
Z
X
Y
X

3D-CAD View 1
STAR-CCM+
65 deg
Y
Z
X
Y
X

Removing Dimensions

You can remove dimensions from a sketch in two ways:

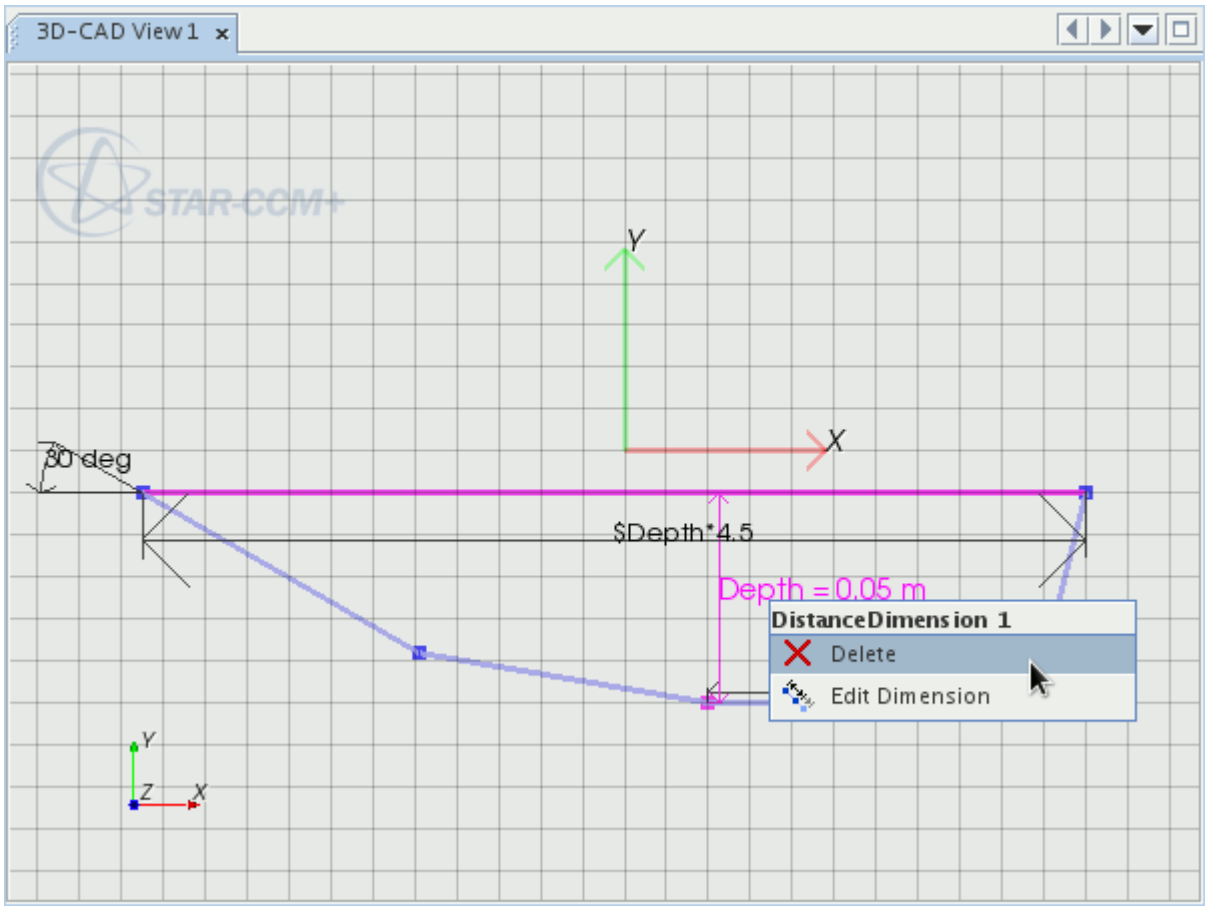
- By selecting the dimension in the *3D-CAD View* scene.
- From the Dimensions list in the *Sketch* panel.

Note: Deleting a dimension that is exposed as a design parameter does not delete the associated design parameter.

Removing Dimensions Using the 3D-CAD View Scene

To remove a dimension directly in the *3D-CAD View* scene:

- Right-click on the dimension marker in the sketch and select **Delete**.



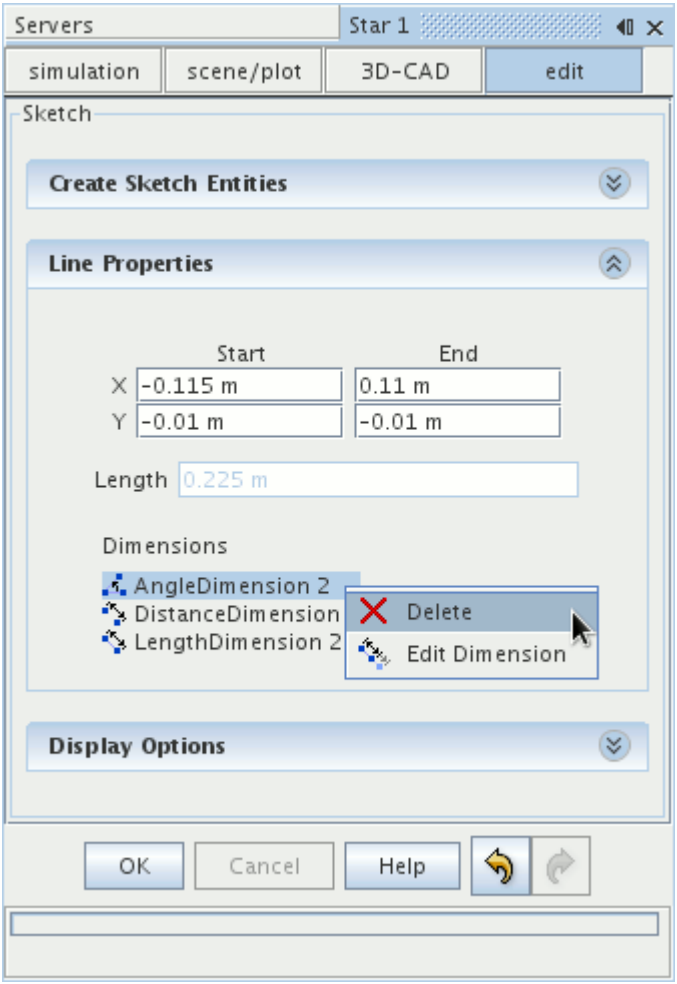
- Alternatively, select the dimension and press the `<Delete>` key.

Removing Dimensions Using the Sketch Panel

To remove a dimension using the list of applied dimensions in the *Sketch* panel:

1. Select the sketch entity in the *3D-CAD View* scene from which you want to remove a dimension. In the *Sketch* panel, a list of dimensions that are applied to the selected sketch entity are displayed in the *Properties* box.

2. Right-click on the dimension that you want to remove and select **Delete**.



See Also:

What Is a Design Parameter?

Creating Bodies From Sketches

In 3D-CAD, you create bodies by extruding, revolving, lofting, or sweeping sketches. Once a body is created, you add or remove material from it.

Use the following features to create bodies from sketches:

- Extrusion Features—An extrusion feature projects a chosen sketch in a specified direction to create a body or a cut.
- Revolved Features—A revolve feature revolves a sketch around a chosen axis to create a body or cut. The axis of revolution is defined by a construction line in the same sketch.
- Lofted Features—A loft fills the space between two sketches to form a solid or sheet body. At least two sketches containing the loft profiles are required. You can also use guide wires to guide the loft from one

profile to the next. A loft cut uses the same principle as a loft to remove material from a solid body. You require at least two sketches containing the profiles of the loft. You can also use guide wires to guide the loft.

- Swept Features—A sweep requires two sketches, one to define the sweep profile, and the other to define the sweep path. The profile is swept along the path. A sweep cut requires two sketches, one to define the sweep profile, and the other to define the sweep path.

You can also create a sheet body from a closed sketch. See Creating Sheet Bodies from Sketches.

Contents:

Extrusion Features
Revolved Features
Lofted Features
Swept Features
Creating Sheet Bodies from Sketches

See Also:

Sketches
Overview of Bodies in 3D CAD

Extrusion Features

An extrusion projects a chosen sketch in a specified direction to create a body or a cut.

If the sketch is a closed profile, the extrusion forms a solid body. If the sketch is an open profile, the extrusion forms a sheet body. Open profiles do not form cuts; they only form new sheet bodies. In an extruded cut, the solid body that is formed by projecting the closed sketch is subtracted from any other solid bodies in its path.

When creating an extruded body or cut, you can set a draft angle that expands or contracts the profile along the extrusion distance.

Requirements

For solid bodies:

- The sketch must contain only closed loops, with no free edges.
- Sketch entities must not overlap or lie on top of each other.
- For sketch entities containing two closed profiles, one profile must be entirely enclosed by the other. You cannot define more than two closed profiles for an extrusion.

For sheet bodies:

- Open sketches are supported.
- Sketch entities can overlap or lie on top of each other.

Construction lines are ignored in the extrusion process.

Contents:

Creating an Extrusion Feature
Extrude and Extrude Cut Panels
Draft Angle Limitation

See Also:

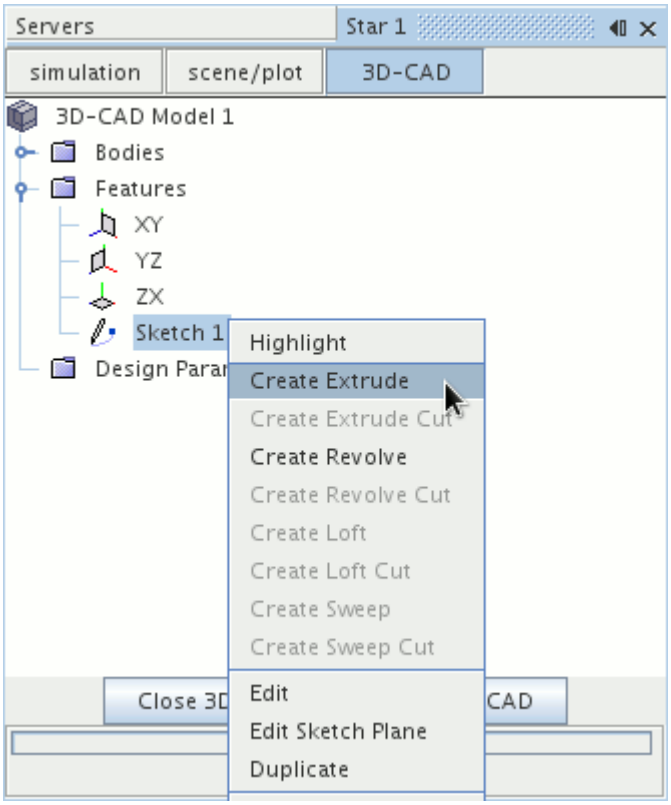
Sketches
Working with Construction Entities

Creating an Extrusion Feature

Use the same procedure to form an extrusion body or cut.

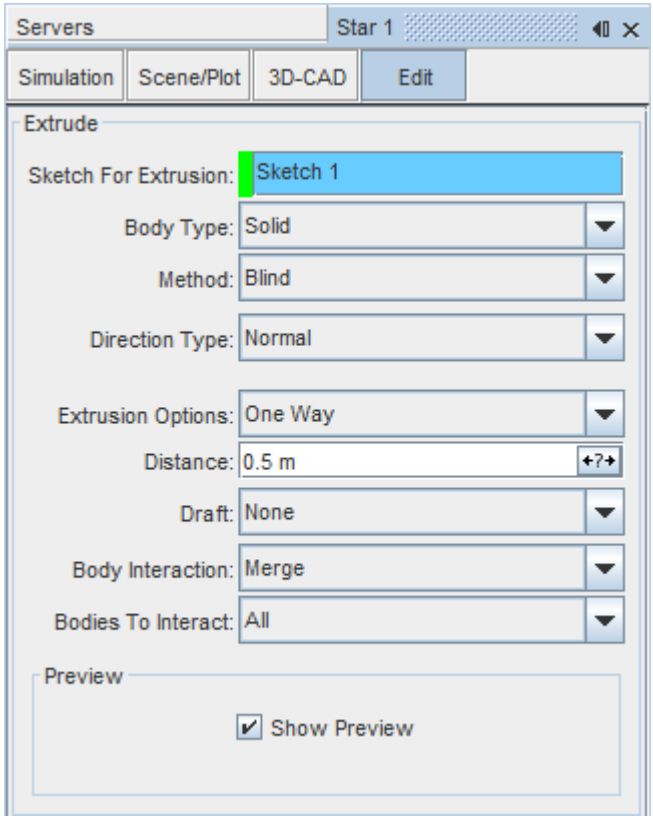
To create an extrusion feature:

1. Either:
   - Right-click the sketch feature that you wish to extrude and choose **Create Extrude** or **Create Extrude Cut**.



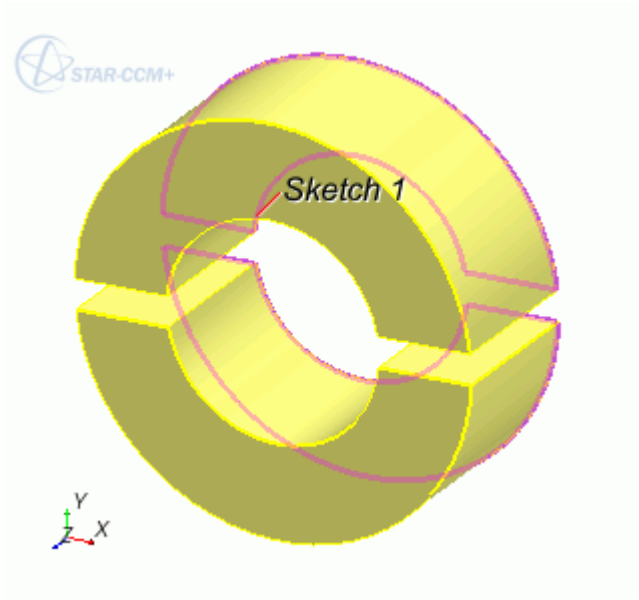
   - Click (**Create Extrude**) or (**Create Extrude Cut**) in the 3D-CAD toolbar and choose the source sketch for the extrusion.

The *Extrude* panel appears, with the identity of the chosen sketch shown in the bottom of the panel.



2. Enter the extrusion parameters. See Extrude Panel.
3. Expose any properties that you want as design parameters by clicking the (**Expose Parameter)** button next to the property.

   The *3D-CAD view* scene updates to show a preview of the body.



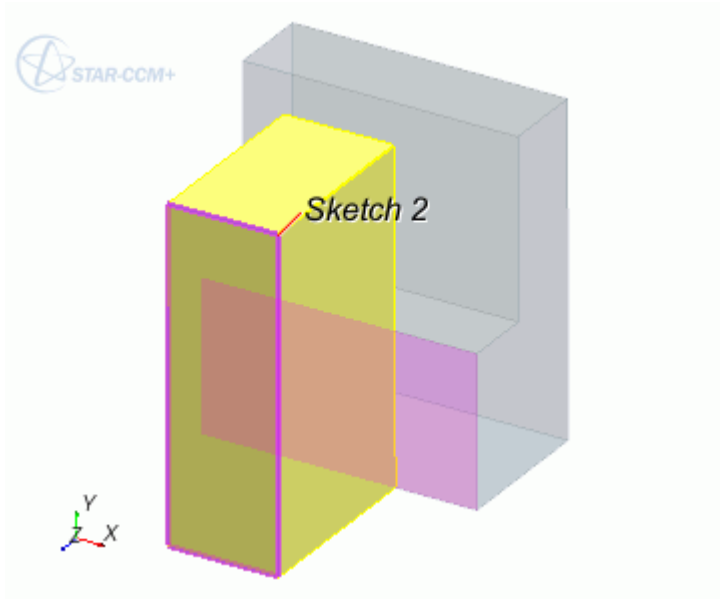
4. Click **OK**.

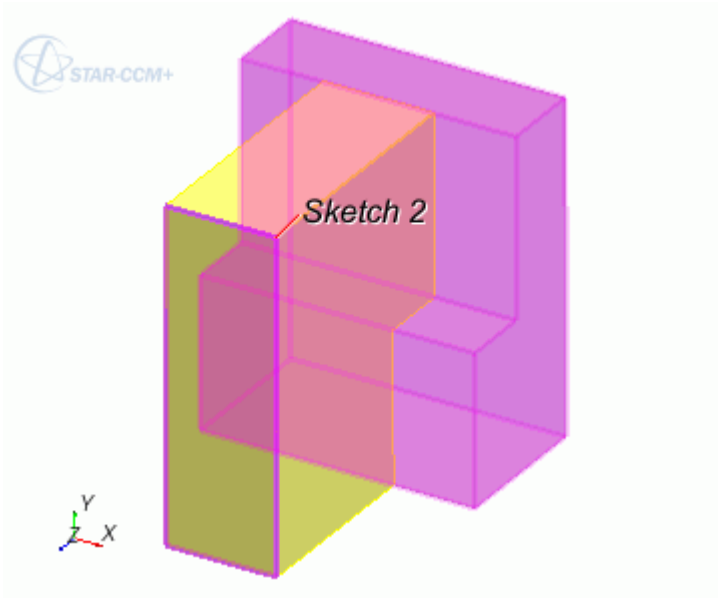
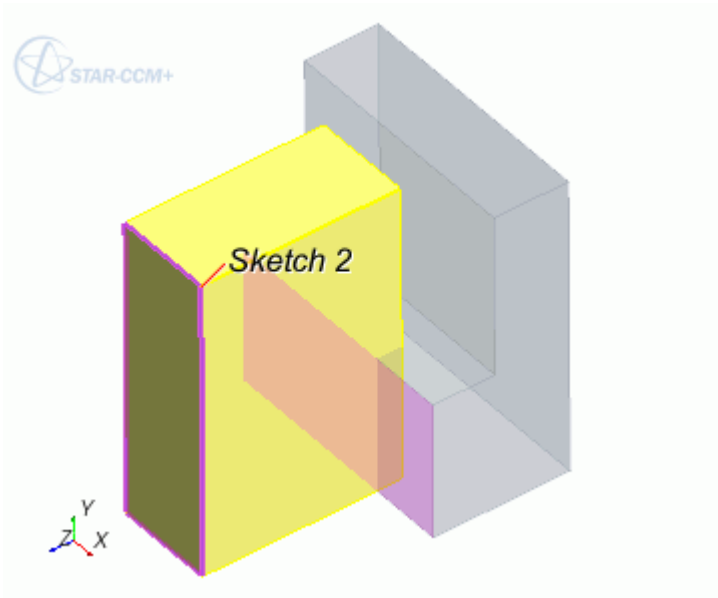
See Also:

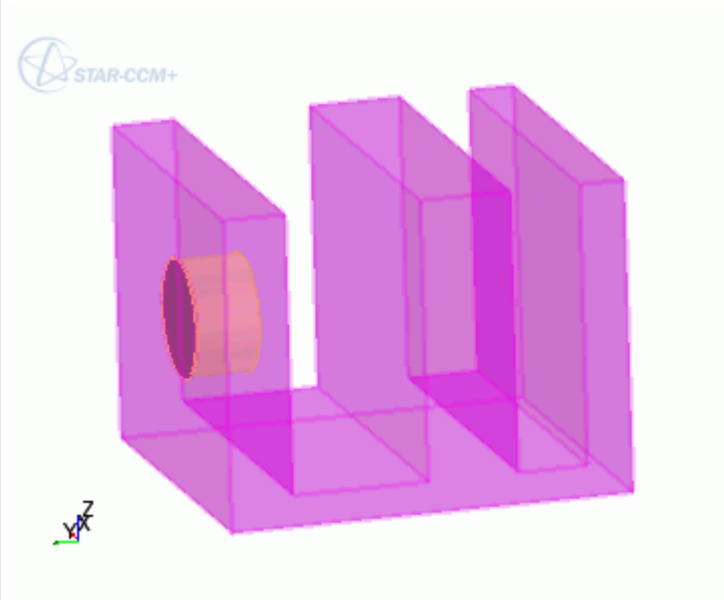
What Is a Design Parameter?

Extrude and Extrude Cut Panels

Use the following properties to define the behavior of extrusion features.

Sketch For Extrusion	Specifies the sketch to extrude.	
Sketch to Extrude Cut	Specifies the sketch to extrude cut.	
Body Type	Defines the type of body that results from the operation.	
	Solid	The resulting body is a closed solid.
	Sheet	The resulting body is an open sheet body.
Method	Defines how to set the end-point of the extrusion.	
	Blind	Extrudes the sketch to the distance that you set in the **Distance** property.
	Through All	Extends the sketch through all bodies, cutting everything in its path.
	Up To Face	Extends the sketch to a selected face or its underlying surface. Click the desired face to select it. The face that you select is displayed in the *Selected Face* property. 

	Up to Body	Extends the sketch up to a selected body. Click the desired body to select it. The body that you select is displayed in the *Selected Body* property. 
	Beyond Face	Extends the sketch beyond the selected face by the amount that you set in the **Offset Distance** property. 
	Through Body First Segment	Extends the sketch to cut through the first segment in the selected body. Click the desired body to select it.

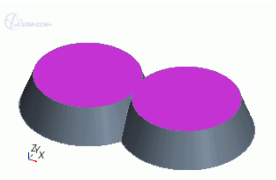
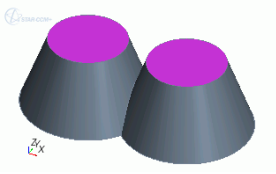
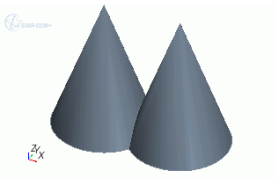
		
Direction Type	Specifies the direction of the extrusion.	
	Normal	Extrudes the sketch in the normal direction.
	Reverse	Extrudes the sketch opposite to the normal direction.
	Specified	Extrudes the sketch in a custom direction, whose vector components are specified in the *Direction Vector* property.
	Reference Axis	Extrudes the sketch in the direction of a reference axis. Select an existing reference axis from the *3D-CAD View* scene. See Working with Reference Geometry in 3D-CAD.
Direction Vector	Specifies the custom extrusion direction. This property is activated when you set the *Direction* property to **Specified**.	
	X	X component.
	Y	Y component.
	Z	Z component.
Coordinate System	Specifies the coordinate system that is used in the extrusion operation.	
Extrusion Options	Specifies whether the sketch extends in a single direction or both directions.	
	One Way	Extrudes the sketch in one direction (specified in the *Direction* parameter).
	Two Way Symmetric	Extrudes the sketch the same distance in both directions.
	Two Way Asymmetric	Extrudes the sketch in both directions. The extrusion on either side of the sketch can be a different size. This option activates the **Asymmetric Distance** text box.
Distance	Specifies the length of the extrusion.	

Asymmetric Distance	Specifies the asymmetric extrusion distance. This option is activated when you set the *Extrusion Options* property to **Two Way Asymmetric**.	
Offset Distance	Specifies the offset distance of the extrusion from the selected surface. The offset distance is measured from the surface that you select.	
Draft	Specifies whether the extrusion has a constant cross-sectional profile along its length, or whether it tapers inwards or expands outwards.	
	None	Extrusion has a constant cross-section along the extrusion.
	Inward	Extrusion tapers inwards. When applying an inward draft to a concave profile, the profile must not contract to the point where its edges overlap and intersect. Similarly, if the draft splits the original profile into two profiles, the operation fails.
	Outward	Extrusion expands outwards.
Angle	Specifies the draft angle (in degrees or radians). This option is activated when you set the *Draft* property to **Inward** or **Outward**. Enter values in the range of -90 to 90 degrees, or $-\frac{\pi}{2}$ to $\frac{\pi}{2}$ radians.	
Body Interaction	Specifies how the solid body that is created by the extrusion interacts with other bodies adjacent to it, as defined by **Interacting Bodies**.	
	None	The new body is added to the **Bodies** manager on its own. No change is made if the body is adjacent to, or overlaps, surrounding bodies.
	Merge	The new body is united with any adjacent or overlapping bodies; otherwise it is added on its own to the **Bodies** manager.
	Imprint	The edges of the extruded body are imprinted on any adjacent or overlapping bodies.
Bodies to Interact	Specifies the bodies which interact with the feature.	
	All	The feature interacts with all contacting bodies.
	Selected	The feature interacts with selected bodies only. Click inside the **Interacting Bodies** group box and select the bodies to interact with from the from the *3D-CAD View* scene. These bodies are highlighted in blue.
Bodies to Cut	Specifies the bodies that the feature cuts.	
	All	The feature cuts all bodies.
	Selected	The feature cuts only selected bodies. Click inside the **Cut Bodies** group box and select the bodies to cut from the *3D-CAD View* scene. These bodies are highlighted in blue.

Draft Angle Limitation

When applying a draft angle to an inward curving profile, make sure that the projected surface does not split along the extruded distance.

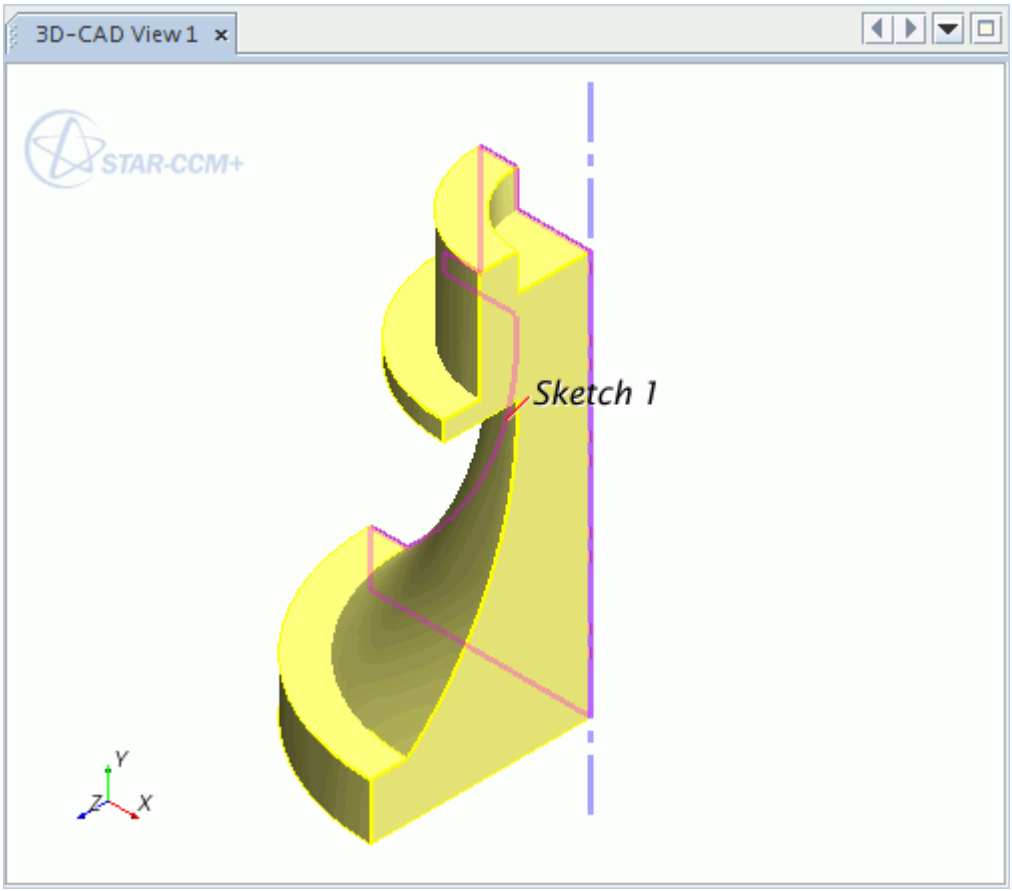
The source profile that is shown in the following example splits when a draft angle is applied along the extrusion.

	Valid: the extrusion stops before the profile is split.
	Invalid: the profile splits due to the combination of draft angle and extrusion length.
	Valid: there is no profile.

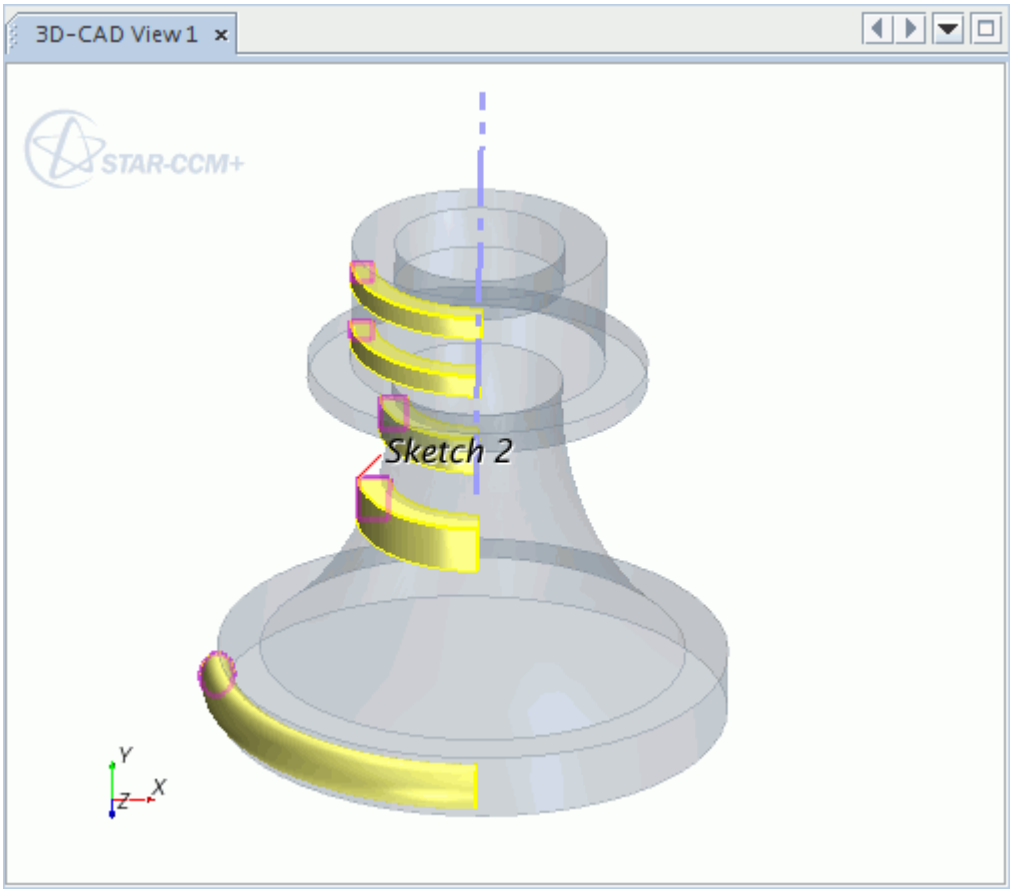
Revolved Features

A revolved feature is an object whose cross-sectional profile is constant around a specified axis.

It is created by sweeping a two-dimensional profile, defined in a sketch, around a specified axis by a given angle.



A revolved cut is defined in the same way; the volume that is defined by the revolved profile is subtracted from any bodies through which it cuts.



Requirements

For solid bodies:

- The sketch must contain only closed loops, with no free edges.
- Sketch entities must not overlap or cross each other.
- For sketch entities containing two closed profiles, one profile must be entirely enclosed by the other. You cannot define more than two closed profiles for a revolve feature.

For sheet bodies:

- Open sketches are supported.
- Sketch entities can overlap or lie on top of each other.

The sketch must contain at least one construction line, which defines the axis about which the sketch revolves.

Contents:

Creating a Revolved Feature
Revolve and Revolve Cut Panels

See Also:

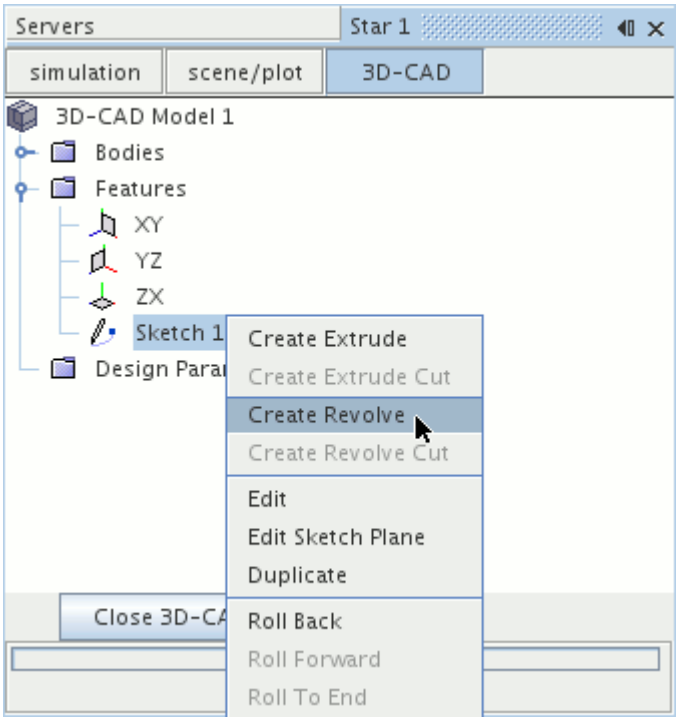
Sketches

Working with Construction Entities

Creating a Revolved Feature

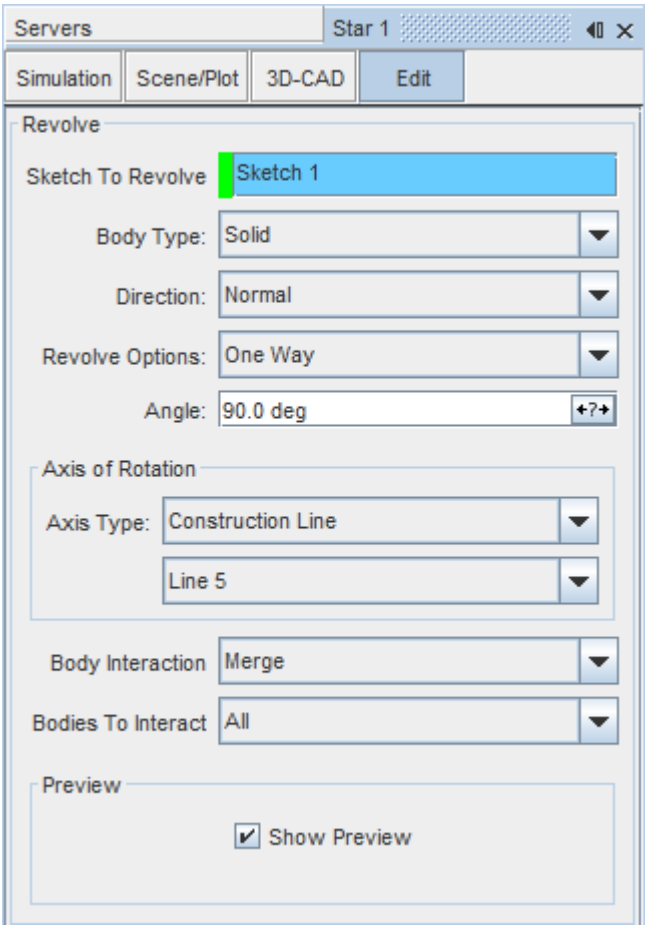
The Revolve and Revolve Cut features share the same procedure. To create a revolved feature:

1. Either:
   - Right-click the sketch feature that you wish to revolve and choose **Create Revolve** or **Create Revolve Cut**.



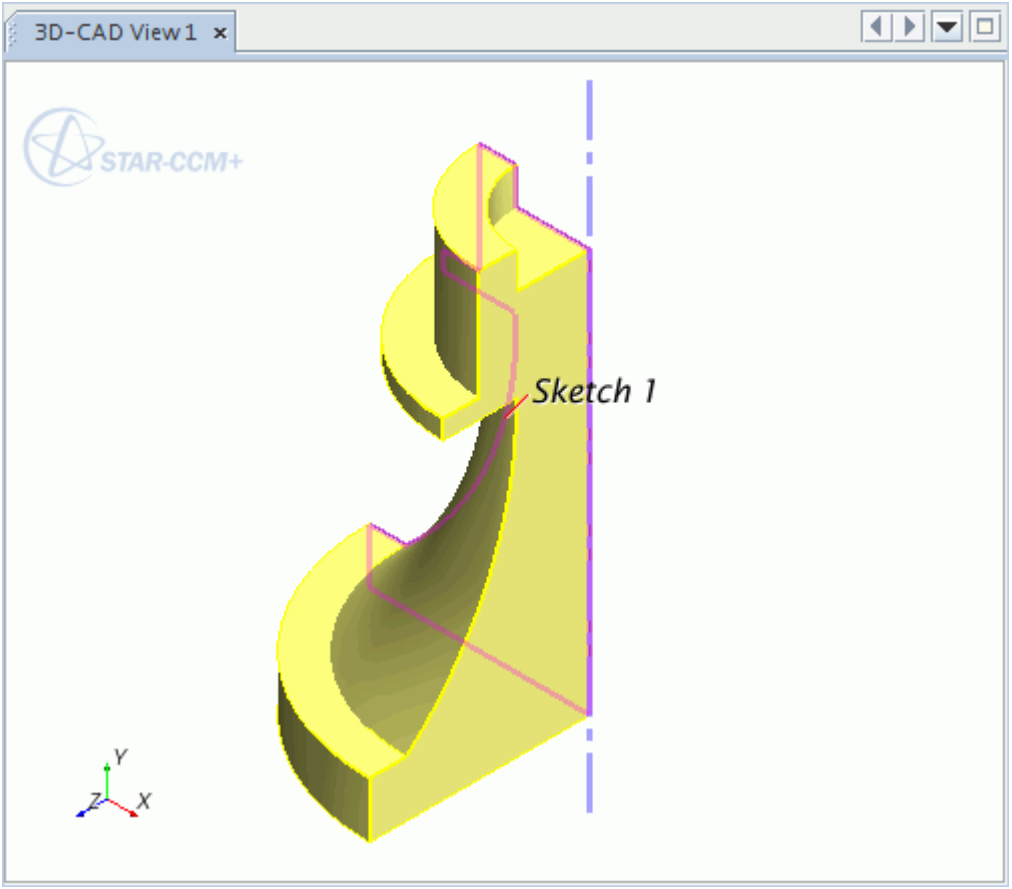
   - Click (**Create Revolve**) or (**Create Revolve Cut**) in the 3D-CAD toolbar and choose the source sketch for the revolve feature.

   The *Revolve* panel appears.



2. Enter the revolve parameters. See Revolve Panel.
3. Expose any properties that you want to use as design parameters by clicking the (**Expose Parameter)** button next to the property.

   A preview of the revolution is shown in the *3D-CAD View* scene.



4. Click **OK** to perform the revolve.

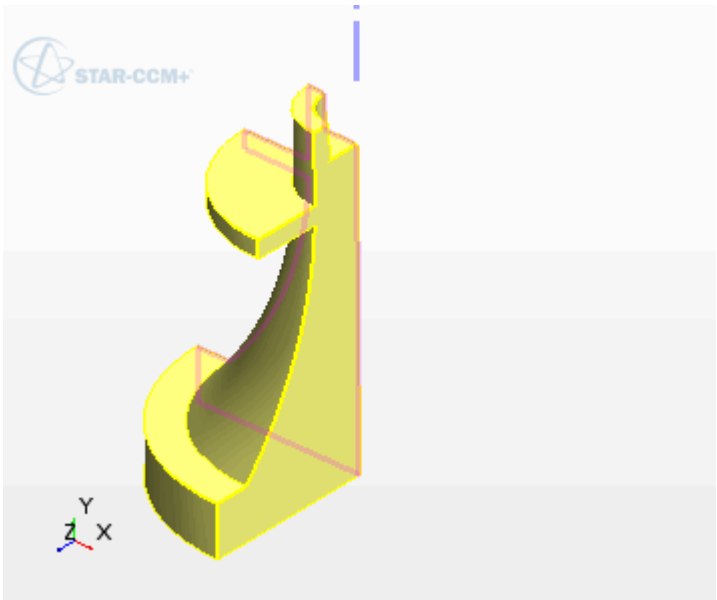
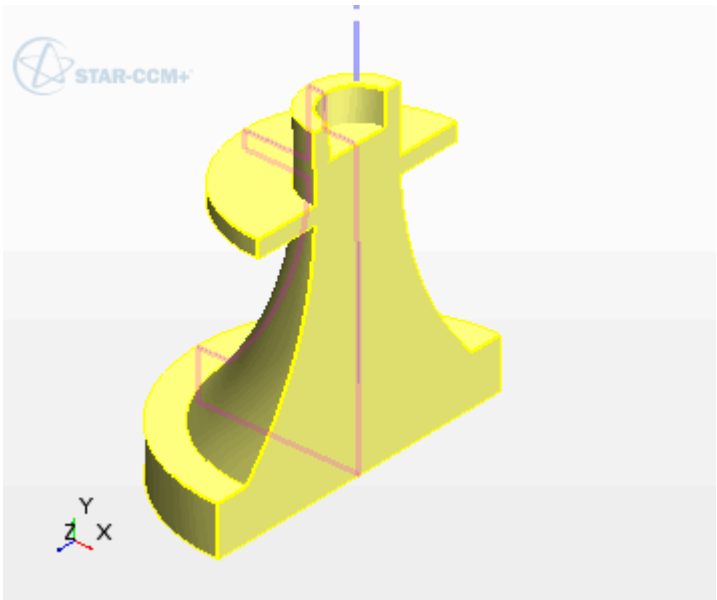
See Also:

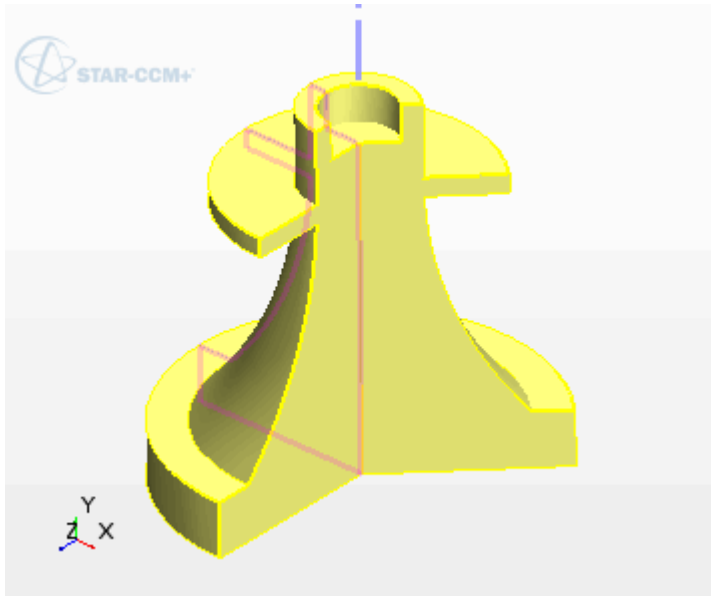
What Is a Design Parameter?

Revolve and Revolve Cut Panels

Use the following properties to define the behavior of the revolve or revolve cut feature.

Sketch to Revolve	Specifies the sketch to revolve.	
Body Type	Defines the type of body that results from the operation.	
	Solid	The resulting body is a closed solid.
	Sheet	The resulting body is an open sheet body.
Direction	Specifies the direction of the revolution.	
	Normal	Revolves the sketch in the normal direction.
	Reverse	Revolves the sketch opposite to the normal direction.
Revolve Options	Specifies whether the revolve extends in a single direction or both directions.	

	One Way	Revolves the sketch in one direction (specified in the **Direction** parameter). 
	Two Way Symmetric	Revolves the sketch the same amount in both directions. 
	Two Way Asymmetric	Revolves the sketch in both directions. The revolve angle on either side of the sketch can be different. This option activates the *Asymmetric Angle* parameter.

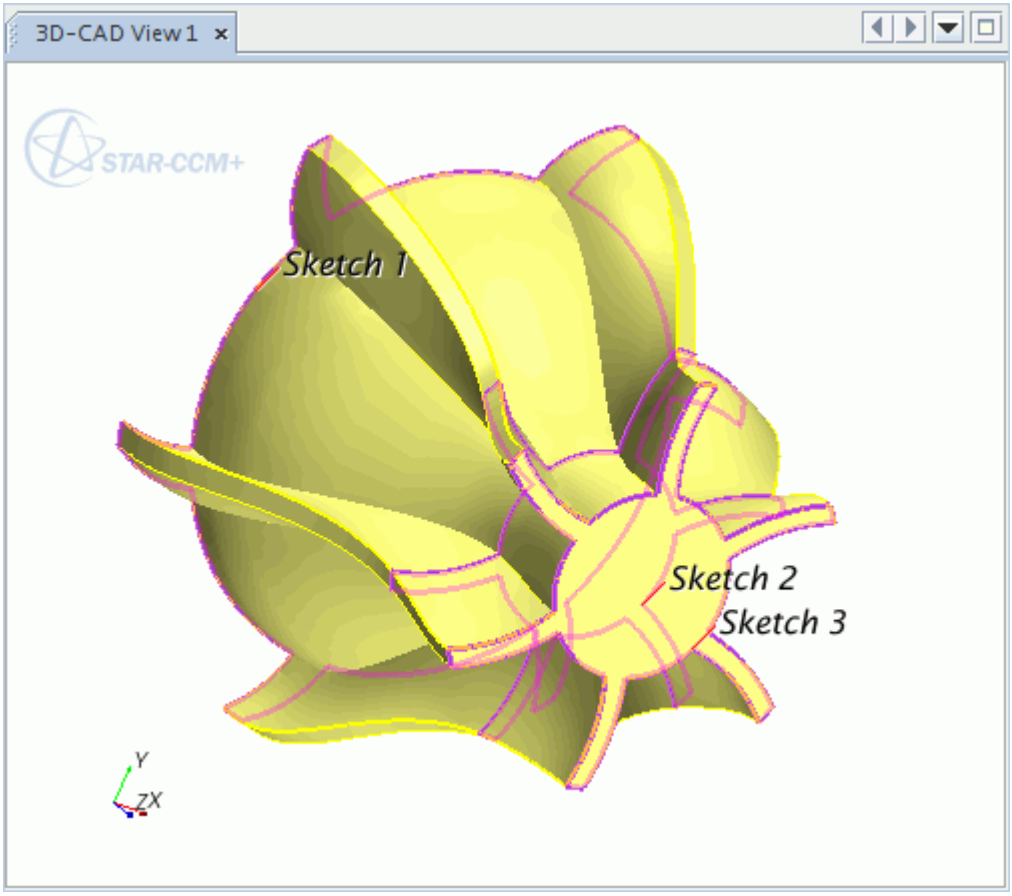
			
Angle	Specifies the angle through which the sketch is revolved.		
Asymmetric Angle	Specifies the asymmetric angle through which the sketch is revolved. This option is activated when you set *Revolve Options* to **Two Way Asymmetric**.		
Axis Type	Specifies the axis of about which the sketch revolves.		
	Specified	Allows you to manually specify the axis of rotation direction and position.	
		Axis Direction	Specifies the direction of the axis of revolution. Enter the [X, Y, Z] components of the direction vector.
		Axis Position	Specifies the position of the axis of revolution. Enter the [X, Y, Z] coordinates for the axis.
		Coordinate System	Defines the coordinate system for the axis of revolution. If you have specified an alternative coordinate system within the simulation, you can define the axis with reference to this coordinate system instead of the default, **Laboratory**.
	Construction Line	Specifies a construction line as the axis of rotation. Select the construction line from the drop-down menu. If the selected sketch contains multiple construction lines, these are listed in the drop-down menu.	
	Reference Axis	Specifies a reference axis as the axis of rotation. Select an existing reference axis from the drop-down menu. See Working with Reference Geometry in 3D-CAD.	
Body Interaction	Specifies how the solid body that is created by the revolve feature interacts with other bodies adjacent to it, as defined by **Interacting Bodies**.		
	None	The new body is added to the **Bodies** manager on its own. No change is made if the body is adjacent to, or overlaps, surrounding bodies.	
	Merge	The new body is united with any adjacent or overlapping bodies; otherwise it is added on its own to the **Bodies** manager.	

	Imprint	The edges of the revolved body are imprinted on any adjacent or overlapping bodies.
Bodies to Interact	Specifies the bodies which interact with the feature.	
	All	The feature interacts with all contacting bodies.
	Selected	The feature interacts with selected bodies only. Click inside the **Interacting Bodies** group box and select the bodies to interact with from the from the *3D-CAD View* scene. These bodies are highlighted in blue.
Bodies to Cut	Specifies the bodies that the feature cuts.	
	All	The feature cuts all bodies.
	Selected	The feature cuts only selected bodies. Click inside the **Cut Bodies** group box and select the bodies to cut from the *3D-CAD View* scene. These bodies are highlighted in blue.

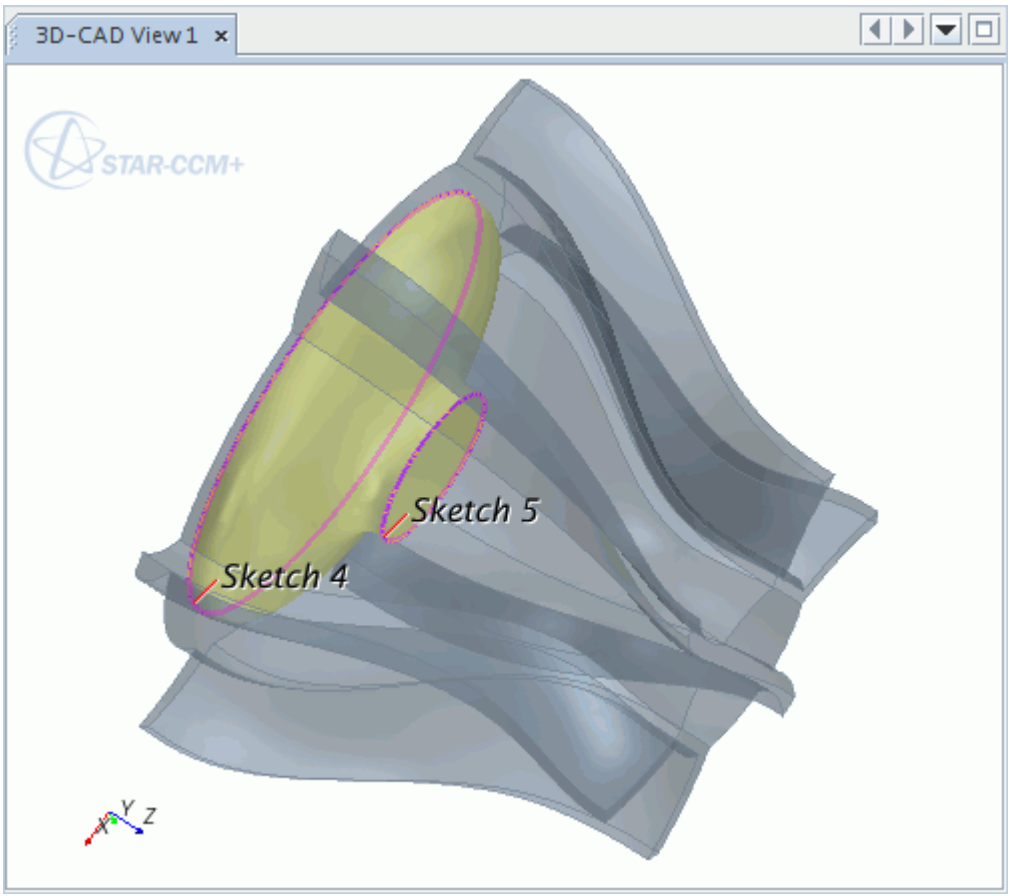
Lofted Features

A lofted feature creates a body by fitting surfaces between a series of profile sketches.

The cross-section of this body varies along the length of the feature according to the shape of each profile sketch. The start and end sketches define the faces that cap either end of the solid body. You require a minimum of two sketches to create a single loft.



A lofted cut is defined in the same way; the volume that is defined by the lofted profile is subtracted from any bodies through which it cuts. You can only use lofted cuts on solid bodies.



When 3D-CAD creates the lofted feature, it matches each vertex on one sketch with a vertex on the next sketch, and fits curves through the pairs for the fitted surfaces. You can manually control which vertices make up these pairs using a feature called Vertex Matching. This technique is useful when you are creating a loft between several sketches, between sketches that have complex geometry, or to define an alternative loft path.

You can further control the shape of the loft by using guide wires. This feature enables you to create complex shapes with twists along the length of the solid.

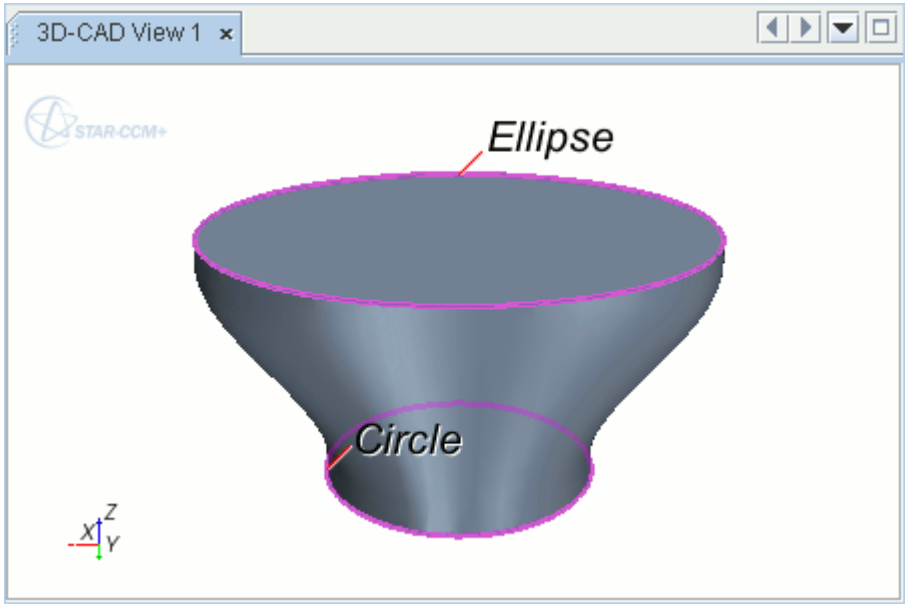
Requirements

The following criteria must be satisfied for the loft to complete successfully:

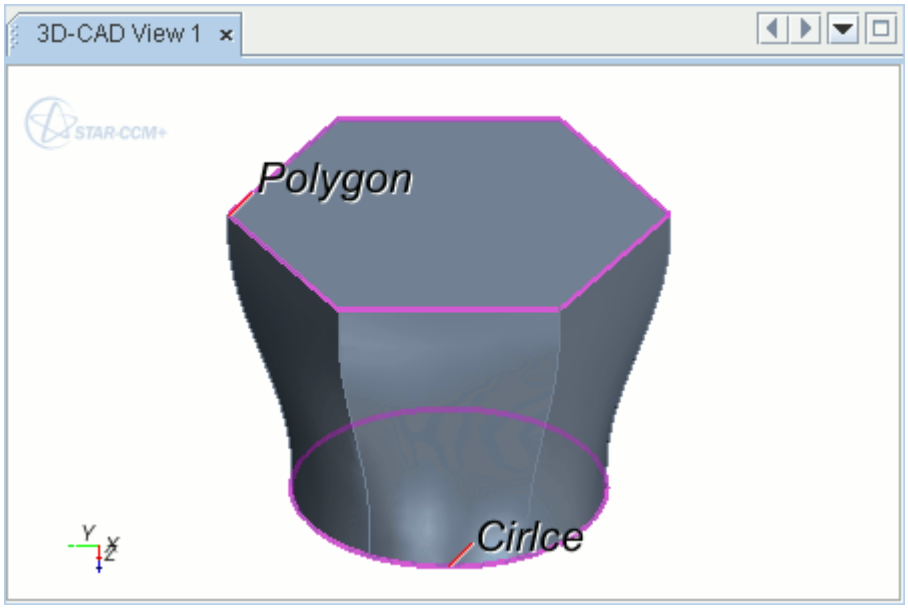
- Each profile sketch must contain a single, closed loop to define the shape of the loft.
- The number of points that lie on the closed loop must be the same for each sketch, unless, for two consecutive sketches, one loop has no points, such as a circle or ellipse. For lofting, splines are considered to have no points. In this case, a loop that defines a single, closed spline is treated in the same way as a circle or ellipse, regardless of the number of points on the spline.

Some examples of the closed loops that can be used to created lofts are shown in the following images:

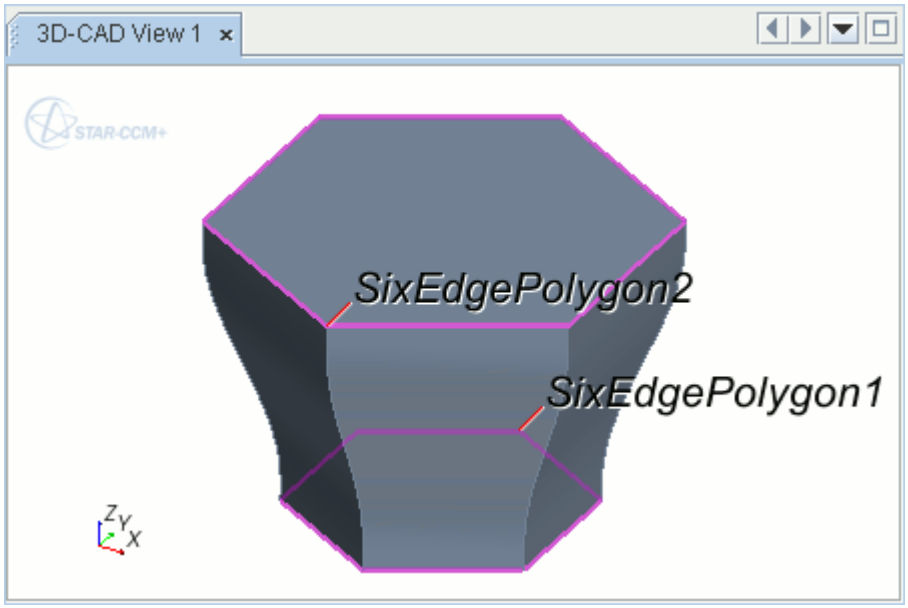
- A circle and an ellipse (two closed loops that contain no points):



- A circle and a polygon (a closed loop with no points and a closed loop with any number of points):



- Two polygons (two closed loops that have the same number of points):



When designing a lofted feature, observe the following points:

- Sketches can lie on sketch planes that are oriented in any direction, but no two consecutive sketches can lie on the same plane.
- You cannot create a lofted feature that passes through itself at any point.
- The shape and position of sketches must avoid any radical changes in the cross-sectional profile of the body.
- Ensure that sketches do not result in any self-intersections in the solid body.
- Construction lines have no impact on the loft.

Contents:

Creating a Lofted Feature
Loft and Loft Cut Panels
Matching Vertices Manually
Using Guide Wires in Loft Operations

See Also:

Using Guide Wires in Loft Operations
Overview of Bodies in 3D CAD

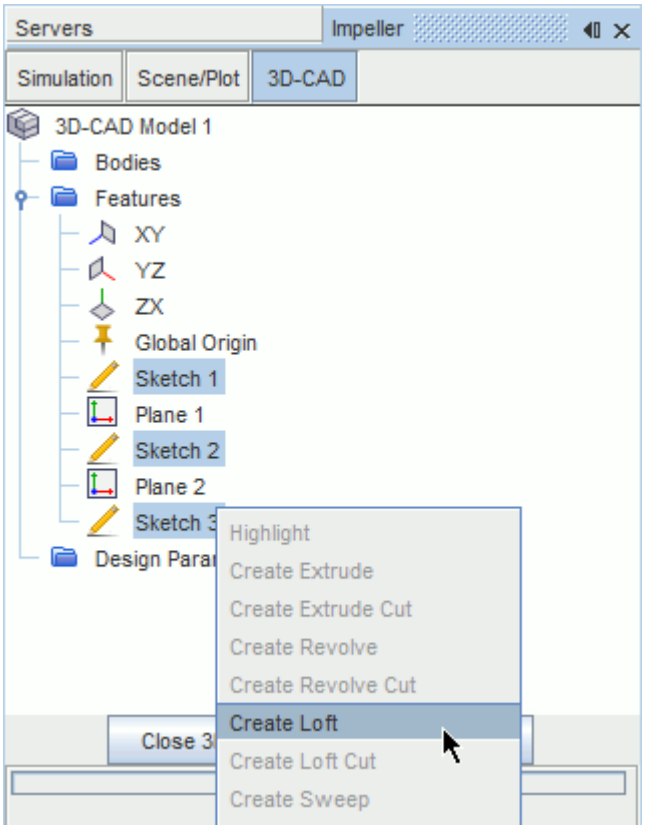
Creating a Lofted Feature

The Loft and Loft Cut features share the same procedure.

Note: In STAR-CCM+ v11.02, a new method is introduced to generate the loft shape. Any existing lofts that you edit in v11.02 are regenerated using the new method. In these cases, a message appears in the *Loft* panel. Once you edit and accept the changes, you will not be able to return to the previous loft shape.

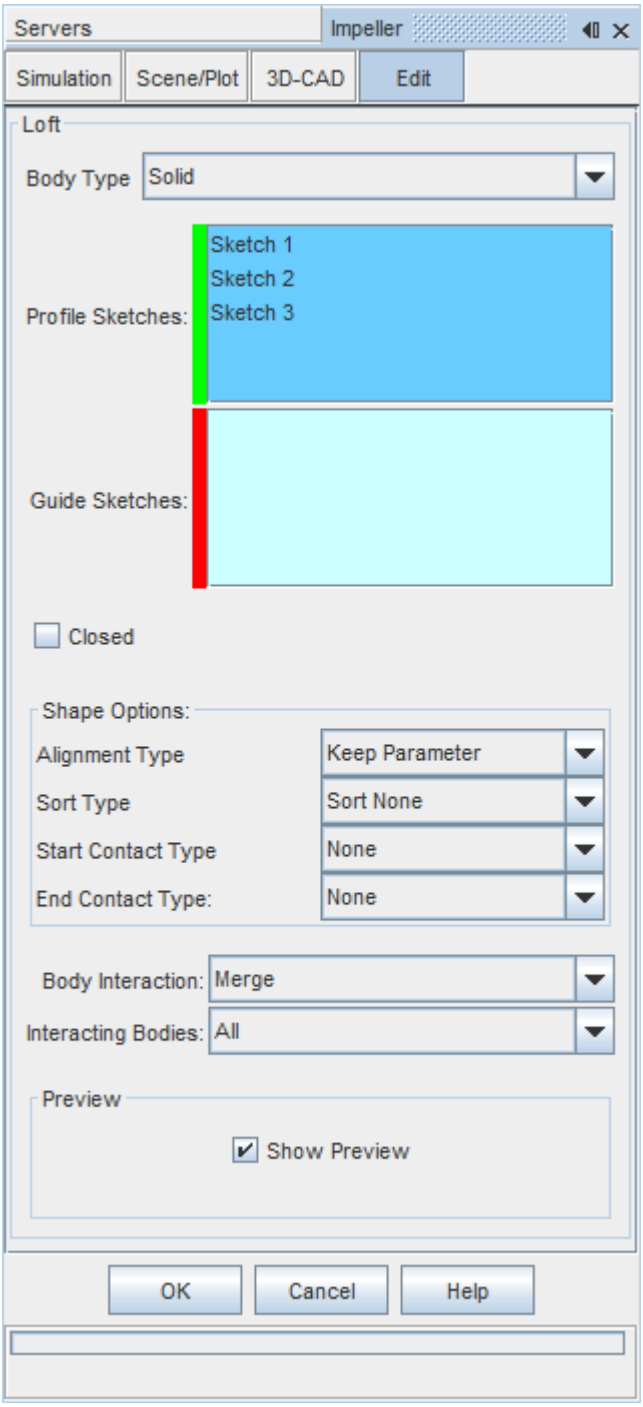
To create a lofted feature:

1. Press `<Ctrl>` key and select the sketch profiles in the order in which the loft passes through them.
2. Either:
   - Right-click on one of the profiles and select **Create Loft** or **Create Loft Cut**.



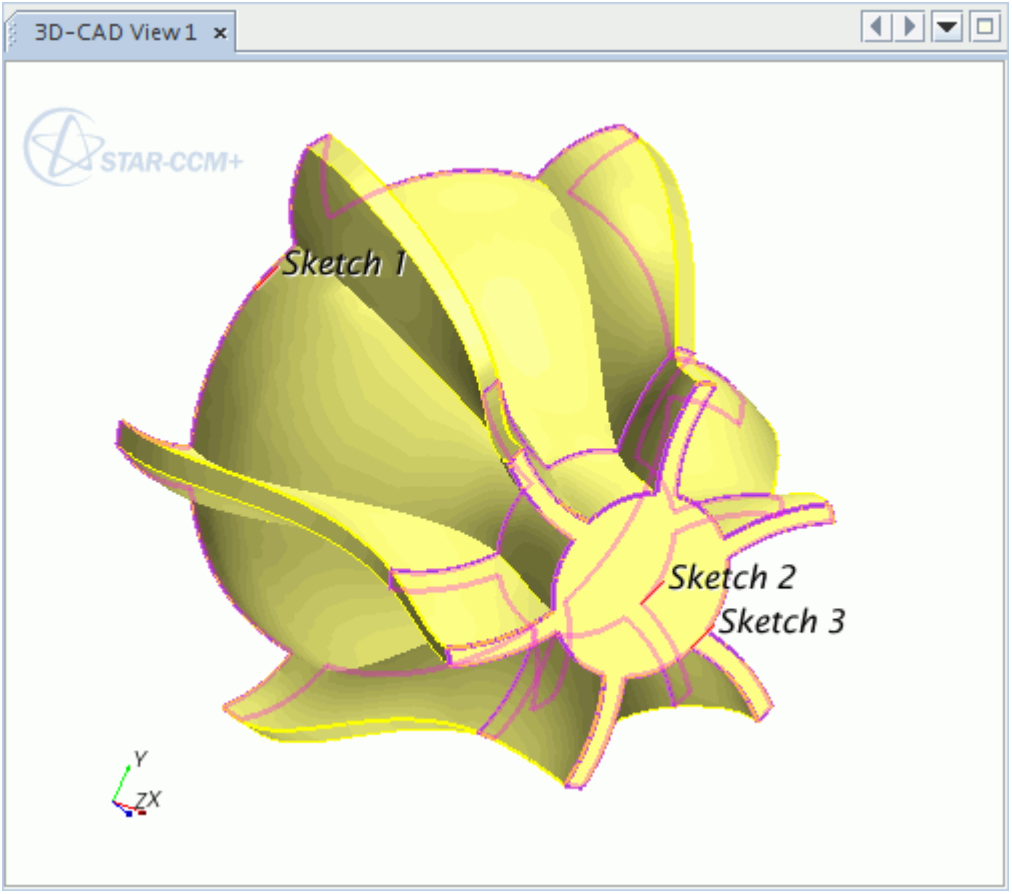
   - Click (**Create Loft**) in the 3D-CAD toolbar.

   The *Loft* panel appears.



3. If the loft requires guide wires to correctly define the loft, select these from the *3D-CAD View* scene and follow the procedure in Using Guide Wires in Loft Operations.
4. If vertices on successive profiles must be matched manually to get the correct orientation, set *Alignment Type* to **User Defined** and follow the procedure in Matching Vertices Manually.
5. Set the remaining loft parameters as you require. See Loft and Loft Cut Panels.

   A preview of the lofted feature appears in the *3D-CAD View* scene.



6. Click **OK** to create the loft.

See Also:

What Is a Design Parameter?

Loft and Loft Cut Panels

Use the following properties to control the behavior of the loft feature or loft cut.

Body Type	Defines the type of body that results from the operation.	
	Solid	The resulting body is a closed solid.
	Sheet	The resulting body is an open sheet body.
Profile Sketches	Specifies the profile sketches for the loft operation. Profile sketches appear in a green color in the *3D-CAD View* scene.	
Guide Sketches	Specifies the guide curves for the loft operation. Guide curves appear in a red color in the *3D-CAD View* scene.	
Closed	Activate to close an a open sheet body loft. Applies to lofts that form a loop.	
Alignment Type	Defines how vertices are aligned and matched in the loft direction.	

	Keep Parameter	Default alignment. Curve parameter remains unchanged.
	Arc Length	Modifies the curve parameter to use the arc length parameterization.
	Maximum Curvature	Modifies the curve parameter to match the maximum curvature in the profile sketches.
	Airfoil	Designed for airfoil shapes with smooth three-dimensional curves, for example blades and wing sections.
	User Defined	Provides the option to manually match vertices across successive sketch profiles. See Matching Vertices Manually. The path of the vertex matching appears in a blue color in the *3D-CAD View* scene.
Start/End Contact Type	Specifies the start/end contact parameters.	
	None	Loft body is not influenced by the start/end profile.
	Contact Normal	Loft body is perpendicular to the start/end sketch.
	Clamped	Loft body is tangent to the start/end profile.
	Natural	Loft body has a minimum bending in the loft direction at the start/end profile.
Body Interaction	Specifies how the solid body that is created by the loft interacts with other bodies adjacent to it, as defined by **Interacting Bodies**.	
	None	The new body is added to the **Bodies** manager on its own. No change is made if the body is adjacent to, or overlaps, surrounding bodies.
	Merge	The new body is united with any adjacent or overlapping bodies; otherwise it is added on its own to the **Bodies** manager.
	Imprint	The edges of the lofted body are imprinted on any adjacent or overlapping bodies.
Bodies to Interact	Specifies the bodies which interact with the feature.	
	All	The feature interacts with all contacting bodies.
	Selected	The feature interacts with selected bodies only. Click inside the **Interacting Bodies** group box and select the bodies to interact with from the from the *3D-CAD View* scene. These bodies are highlighted in blue.
Bodies to Cut	Specifies the bodies that the feature cuts.	
	All	The feature cuts all bodies.

	Selected	The feature cuts only selected bodies. Click inside the **Cut Bodies** group box and select the bodies to cut from the *3D-CAD View* scene. These bodies are highlighted in blue.

Matching Vertices Manually

By default, 3D-CAD predicts the pairs of matching vertices between successive profiles in lofts. Alternatively, you can use the Vertex Matching feature to manually define the matching pairs.

You can also use vertex matching to define an alternative loft path, which enables you to twist the body between the profile sketches. During the process of matching vertices, the path between sketch profiles is determined by matching a vertex on one sketch profile with a vertex on the next profile.

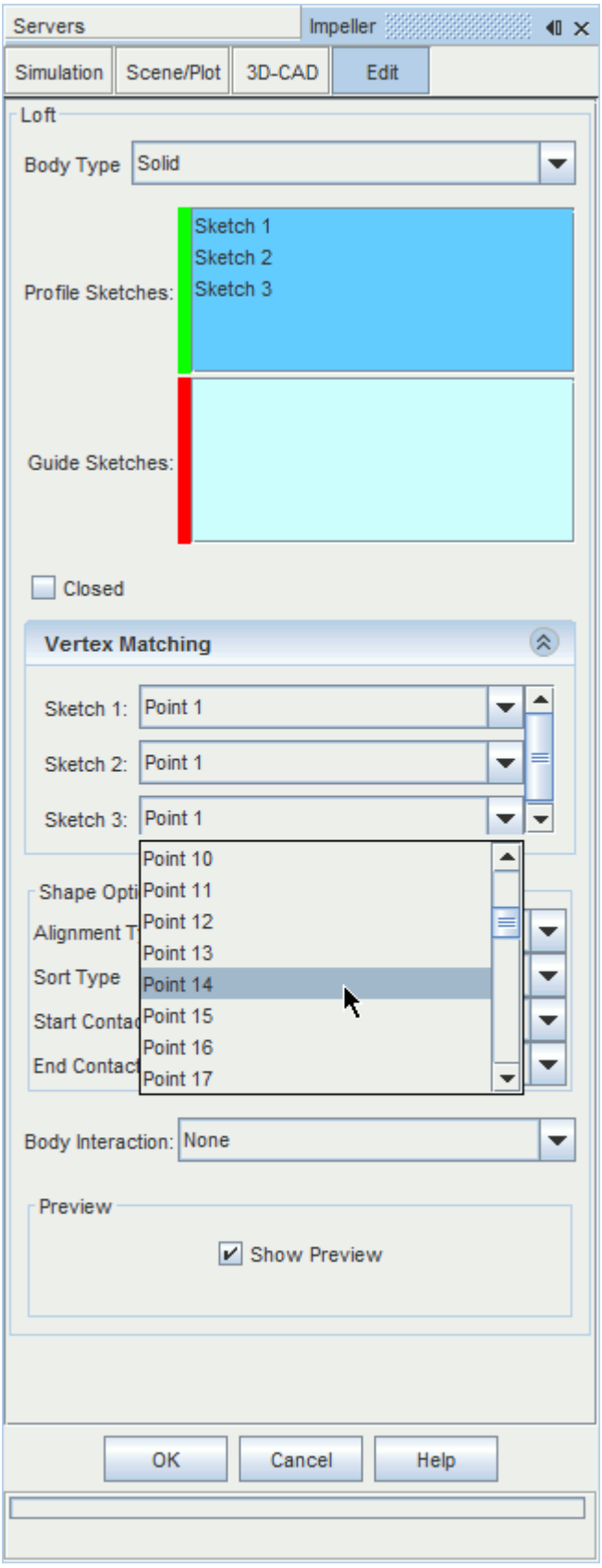
Note: To use vertex matching, the loft profiles must have the same number of vertices.

To carry out vertex matching:

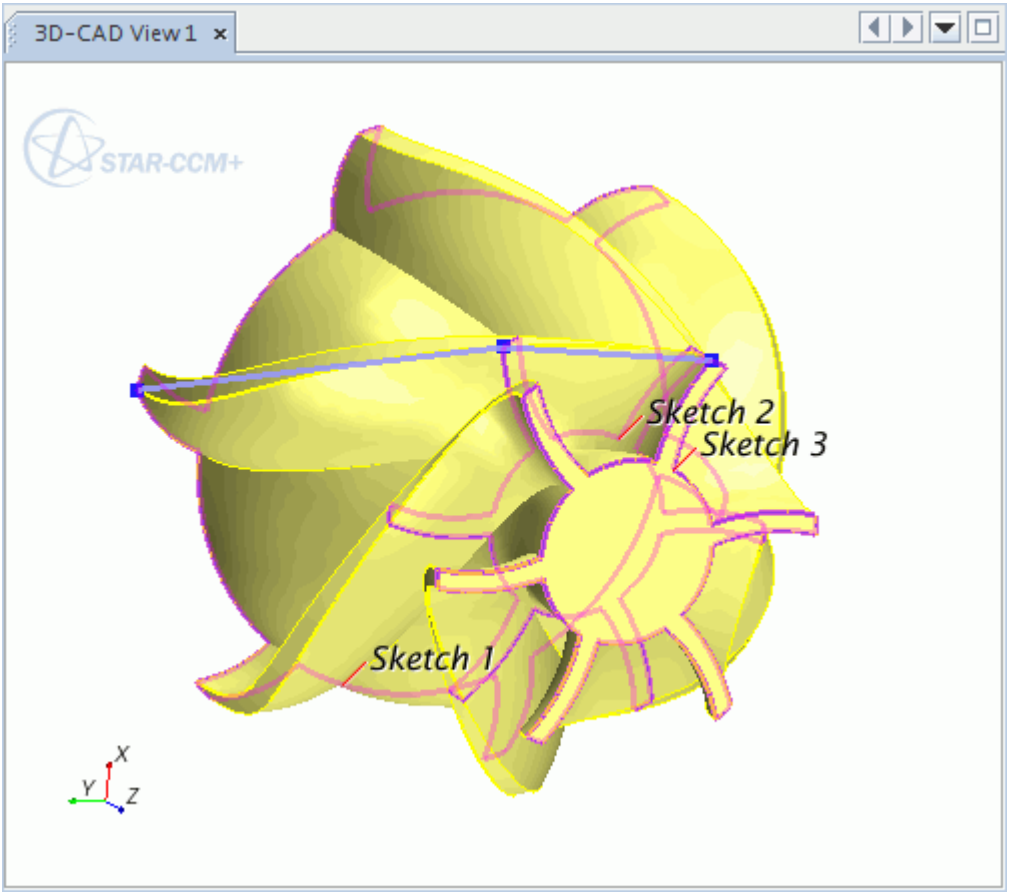
1. In the *Loft* panel, set *Alignment Type* to **User Defined**.

   The *Vertex Matching* box appears and displays a list of sketches that are available in the loft, along with the vertices that belong to each sketch.
2. For each sketch, select a vertex from the corresponding list in the *Vertex Matching* box.

   You can find out the name of a particular point by editing the corresponding sketch before creating the loft, and right-clicking on the point in the *3D-CAD View* scene; the point name is displayed at the top of the right-click menu.

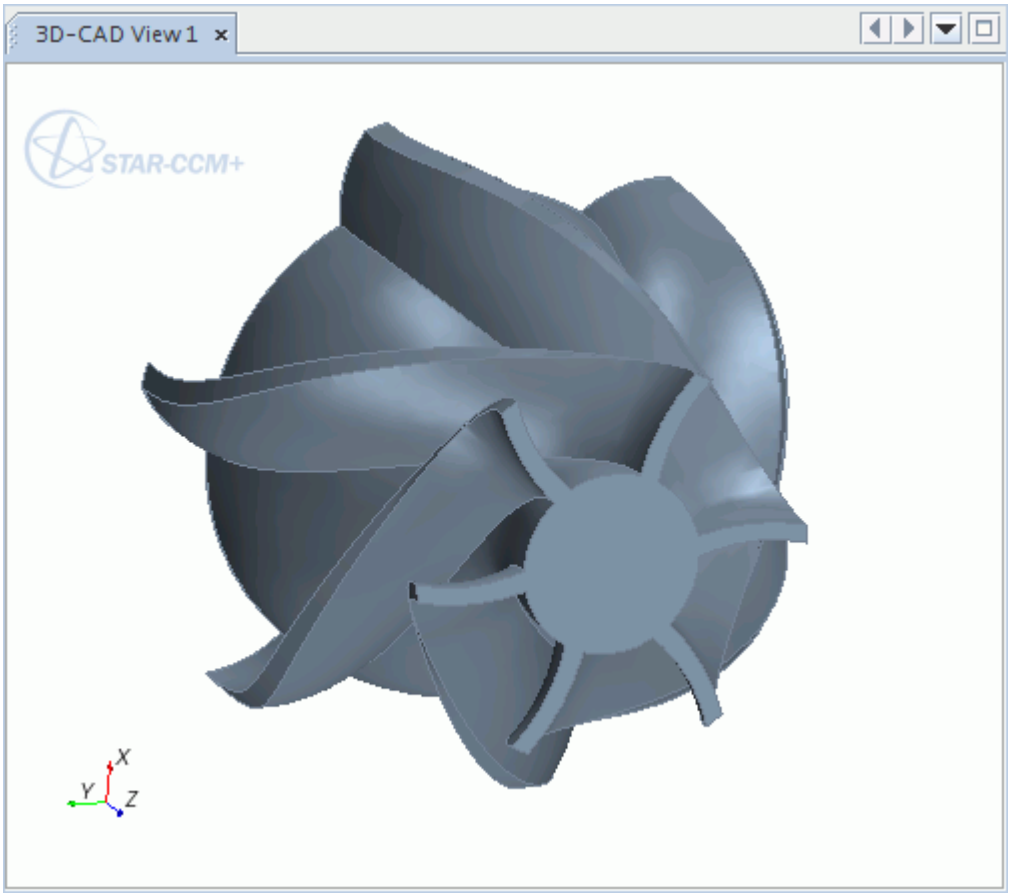

Servers
Impeller
Simulation
Scene/Plot
3D-CAD
Edit
Loft
Body Type
Solid
Profile Sketches:
Sketch 1
Sketch 2
Sketch 3
Guide Sketches:
Closed
Vertex Matching
Sketch 1:
Point 1
Sketch 2:
Point 1
Sketch 3:
Point 1
Point 10
Point 11
Point 12
Point 13
Point 14
Point 15
Point 16
Point 17
Shape Opti
Alignment T
Sort Type
Start Contac
End Contact
Body Interaction:
None
Preview
Show Preview
OK
Cancel
Help

The points that you select from each sketch define a loft path. This path is displayed in the *3D-CAD View* scene. You can check that you have selected the correct vertices by rotating and zooming in on the model. A preview of the loft is displayed in the *3D-CAD View* scene.



3. Click **OK** to create the loft.

The final loft is shown below.



Using Guide Wires in Loft Operations

You can create more complex lofts by using guide wires.

The guide wires constrain the loft along its length and give you greater control on the shape of the loft. You provide guide wires as open sketch profiles along the length of the loft.

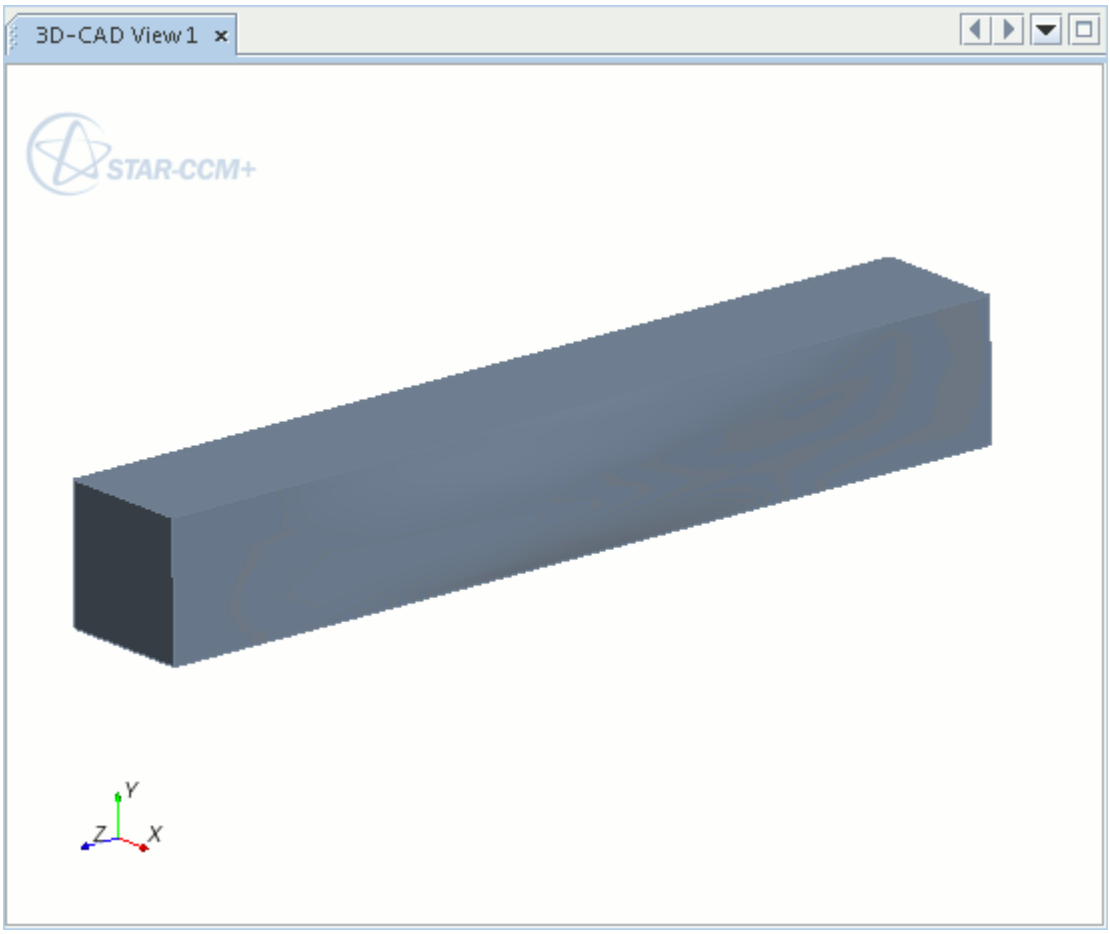
Requirements

The following rules apply when using guide wires:

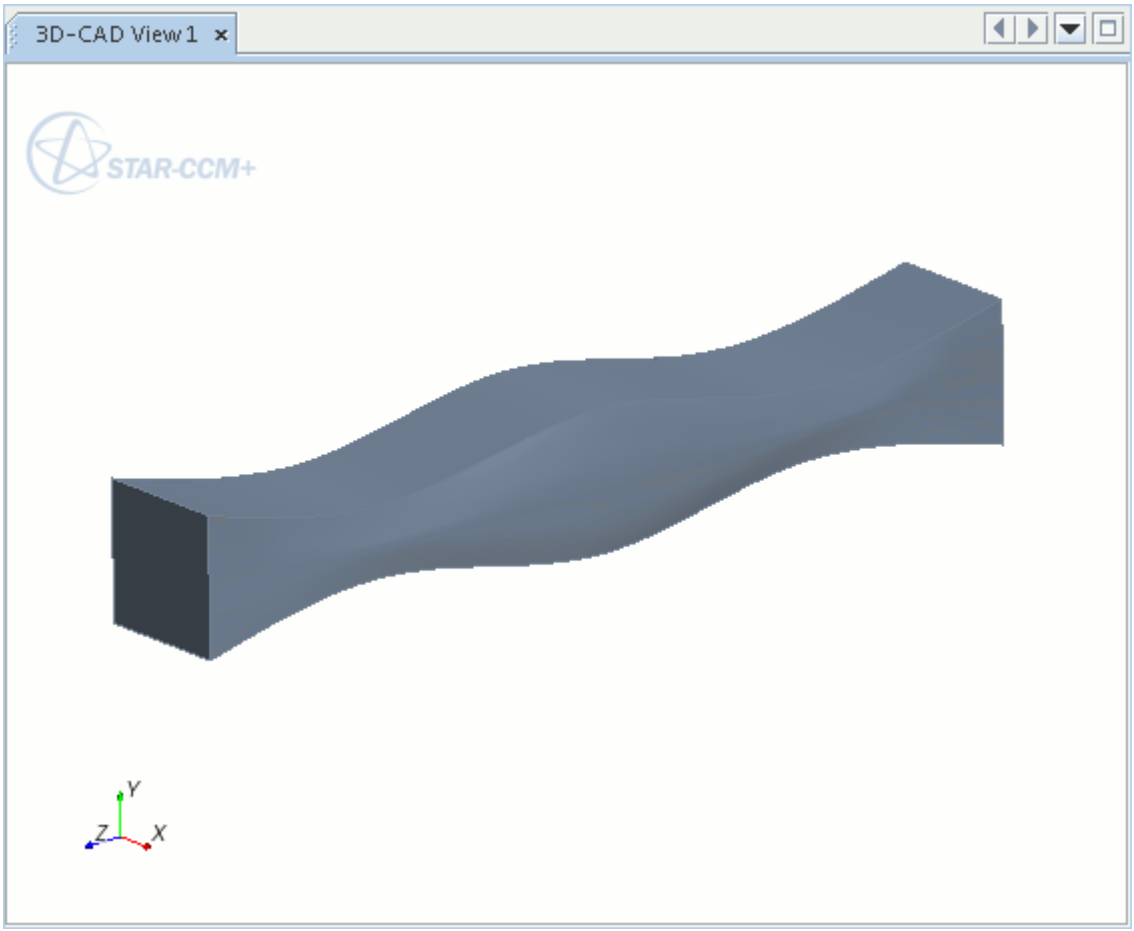
- All loft profiles must have the same number of vertices. For example, a square profile cannot be linked to an elliptical profile using guide wires. However, a square profile can be connected to another profile that contains 4 vertices, as both profiles contain the same number of points.
- The end points of the guide wires must be coincident with the vertices on the start and end sketch profiles for the loft. Guide wires must also pass through the relevant vertices of intermediate profile sketches. Use the pierce constraint to force the points on one sketch to coincide with the edge of another sketch. See Pierce Constraint.
- Only open sketches can be used as guide wires. Closed sketches are automatically assumed to be loft profiles.
- Guide wires must be formed from single, continuous, sketch entities.
- A single sketch can contain more than one guide wire.

Example

The following example demonstrates how the guide wires constrain the loft. 3D-CAD automatically calculates the shape of the loft based on the shape of the profiles and the distance that the profiles are apart.



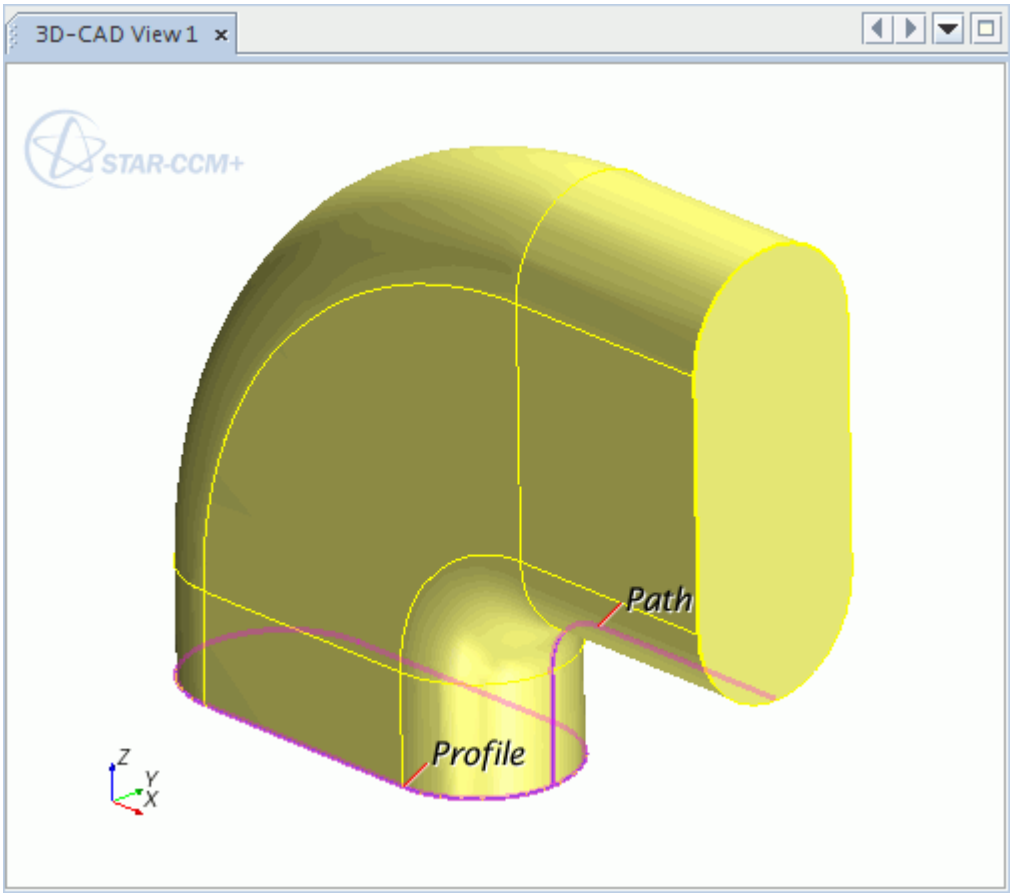
By adding guide wires to the loft, you can control its final shape.



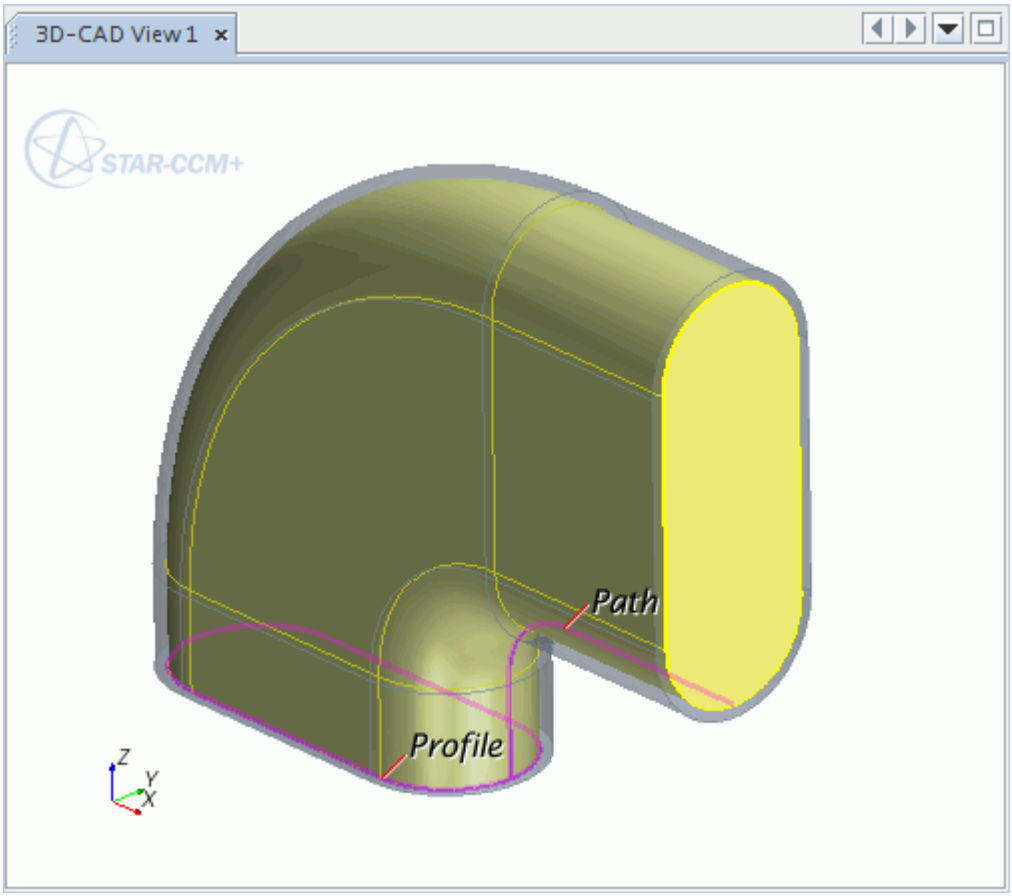
Swept Features

A swept feature creates a body by sweeping a profile along a specified path.

The profile and path are defined in separate sketches.



A swept cut is defined in the same way as a sweep, however the feature determines where material is removed from a body. You can only use swept cuts on solid bodies.



The order in which you select the sketches affects how the feature is created. The sketch that you select first defines the sweep profile. The second sketch that you select defines the sweep path. Renaming the sketches according to their purpose makes it easier to choose them correctly.

Requirements

The following criteria must be satisfied for the sweep to complete successfully:

- The profile for sweeping must contain only closed loops.
- For sketch entities containing two closed profiles, one profile must be entirely enclosed by the other. You cannot define more than two closed profiles for a sweep feature.
- Sketch entities must not intersect.
- The path can be either open or closed, but must be defined by a single loop.
- The path may be located anywhere in relation to the profile, however the position of the path affects the creation of the body. For predictable results, it is advised that the starting point of the path intersects the profile.
- The path must not allow the created feature to pass through itself at any point.
- The path must avoid curves with radical changes in direction.

The sketches may contain any number of construction entities.

Contents:

Creating a Swept Feature
Sweep and Sweep Cut Panel

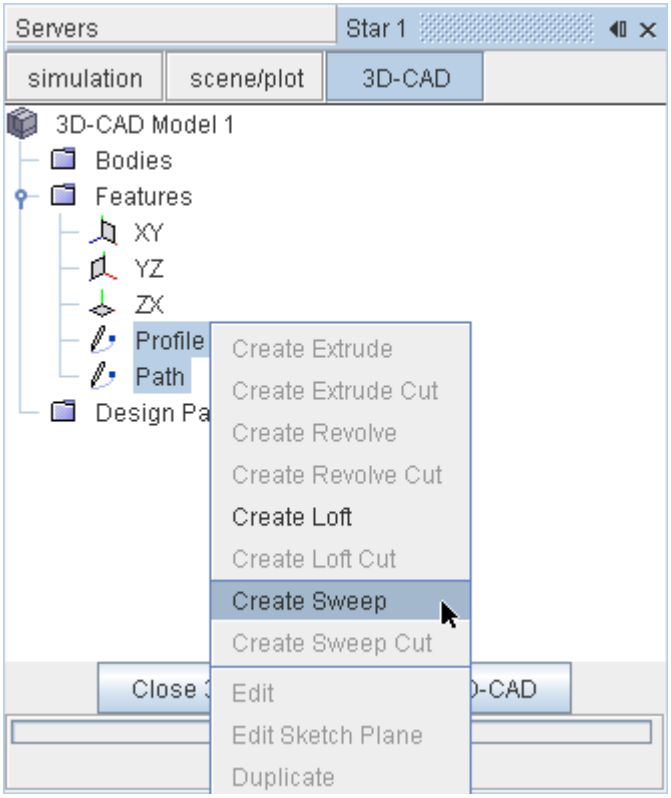
See Also:

Sketches
Working with Construction Entities

Creating a Swept Feature

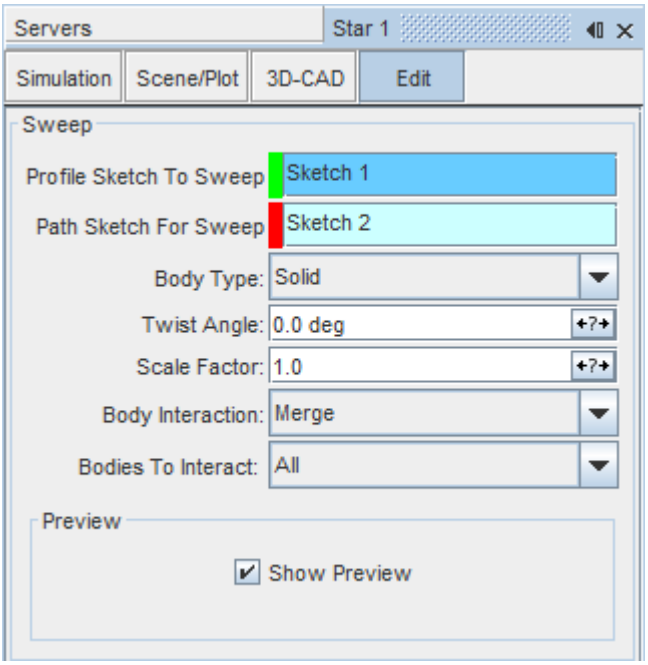
The Sweep and Sweep Cut features share the same procedure. To create a swept feature:

1. Select the sketch that contains the sweep profile from the **Features** manager.
2. Hold `<Ctrl>` and select the sketch that contains the sweep path.
3. Either:
   - Right-click on either sketch in the feature tree and select **Create Sweep**.



   - Click (**Create Sweep**) in the 3D-CAD toolbar.

   The *Sweep* panel appears.

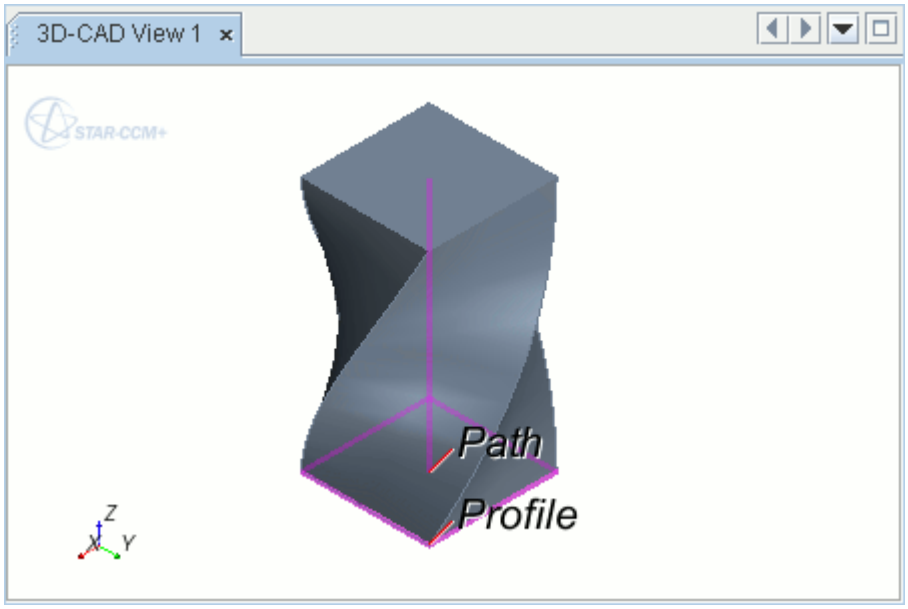
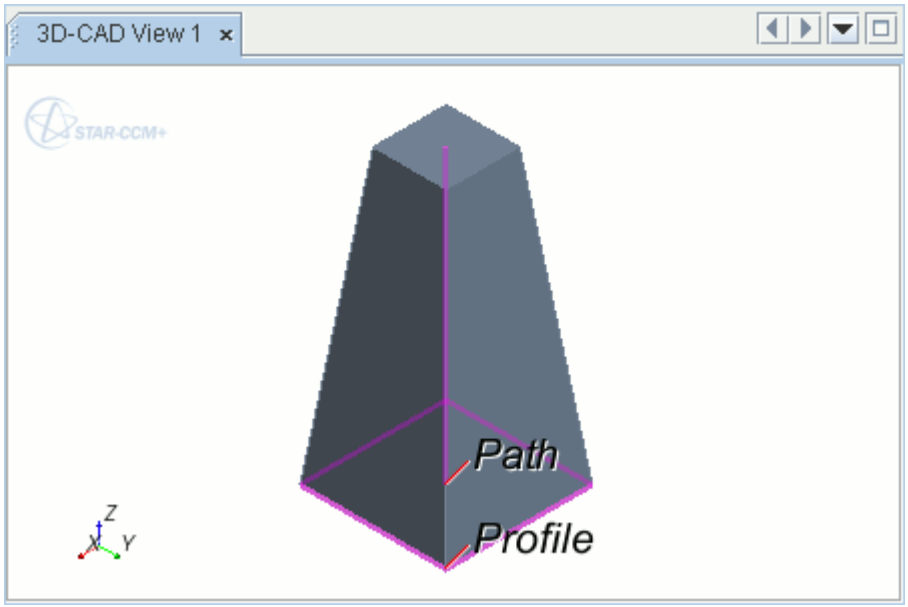


4. Enter the sweep parameters. See Sweep Panel.
5. Expose any properties that you want as design parameters by clicking the (**Expose Parameter)** button next the desired feature.
6. Click **OK**.

Sweep and Sweep Cut Panel

Use the following properties to control the behavior of the sweep or sweep cut feature.

Profile Sketch to Sweep	Specifies the profile for the sweep.	
Path Sketch For Sweep	Specifies the path for the profile to follow.	
Body Type	Defines the type of body that results from the operation.	
	Solid	The resulting body is a closed solid.
	Sheet	The resulting body is an open sheet body.
Twist Angle	Defines the angle that the profile is rotated through as it sweeps along the path. The sweep path defines the axis of twist, which allows the twist to be offset from the center-line of the body.	

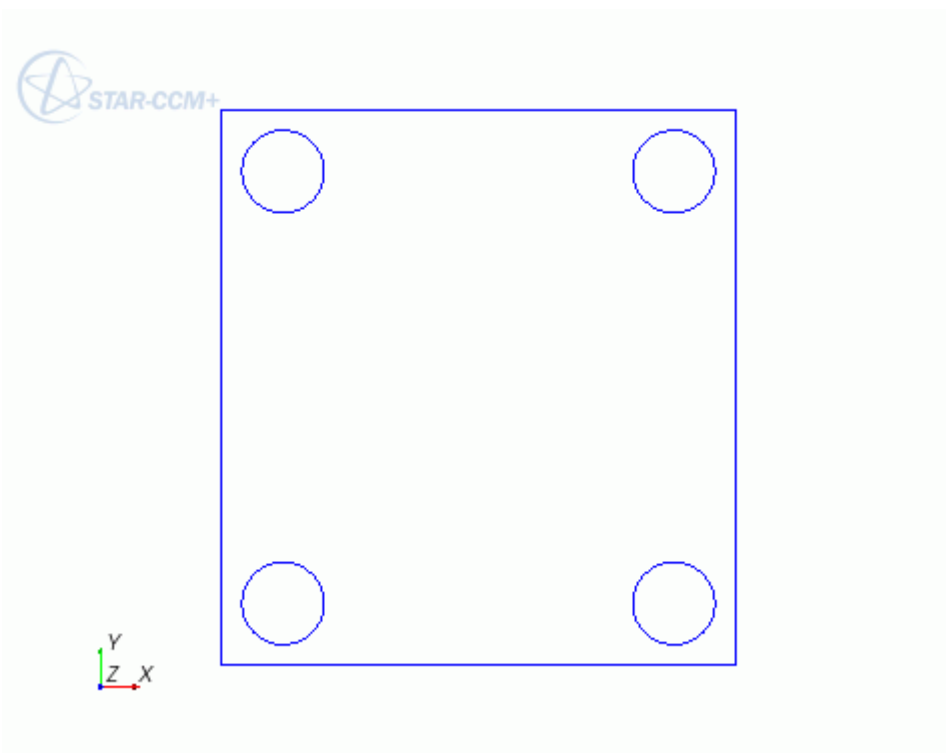
	
Scale Factor	Defines the enlargement or shrinkage of the profile as the feature sweeps along the path. The position of the sweep path in relation to the profile affects the outcome of the scaling by defining the projection of the body. A scale factor of 1.0 maintains the original scale of the profile through the sweep. 
Body Interaction	Specifies how the solid body that is created by the sweep feature interacts with other bodies adjacent to it, as defined by **Interacting Bodies**.
	None — The new body is added to the **Bodies** manager on its own. No change is made if the body is adjacent to, or overlaps, surrounding bodies.
	Merge — The new body is united with any adjacent or overlapping bodies; otherwise it is added on its own to the **Bodies** manager.

	Imprint	The edges of the extruded body are imprinted on any adjacent or overlapping bodies.
Bodies to Interact	Specifies the bodies which interact with the feature.	
	All	The feature interacts with all contacting bodies.
	Selected	The feature interacts with selected bodies only. Click inside the **Interacting Bodies** group box and select the bodies to interact with from the from the *3D-CAD View* scene. These bodies are highlighted in blue.
Bodies to Cut	Specifies the bodies that the feature cuts.	
	All	The feature cuts all bodies.
	Selected	The feature cuts only selected bodies. Click inside the **Cut Bodies** group box and select the bodies to cut from the *3D-CAD View* scene. These bodies are highlighted in blue.

Creating Sheet Bodies from Sketches

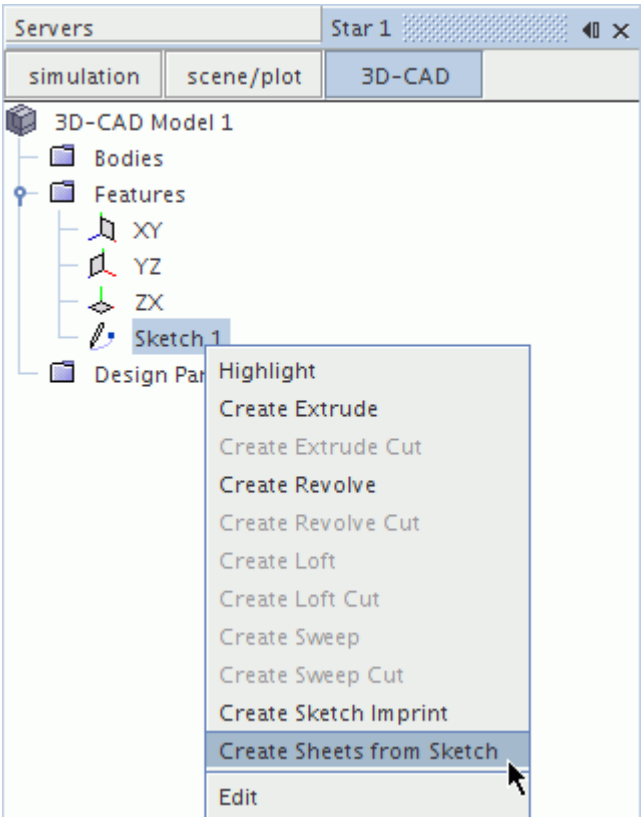
You can create sheet bodies from closed sketches.

The sketch can contain multiple closed loops as shown below. This feature is useful for creating baffles.



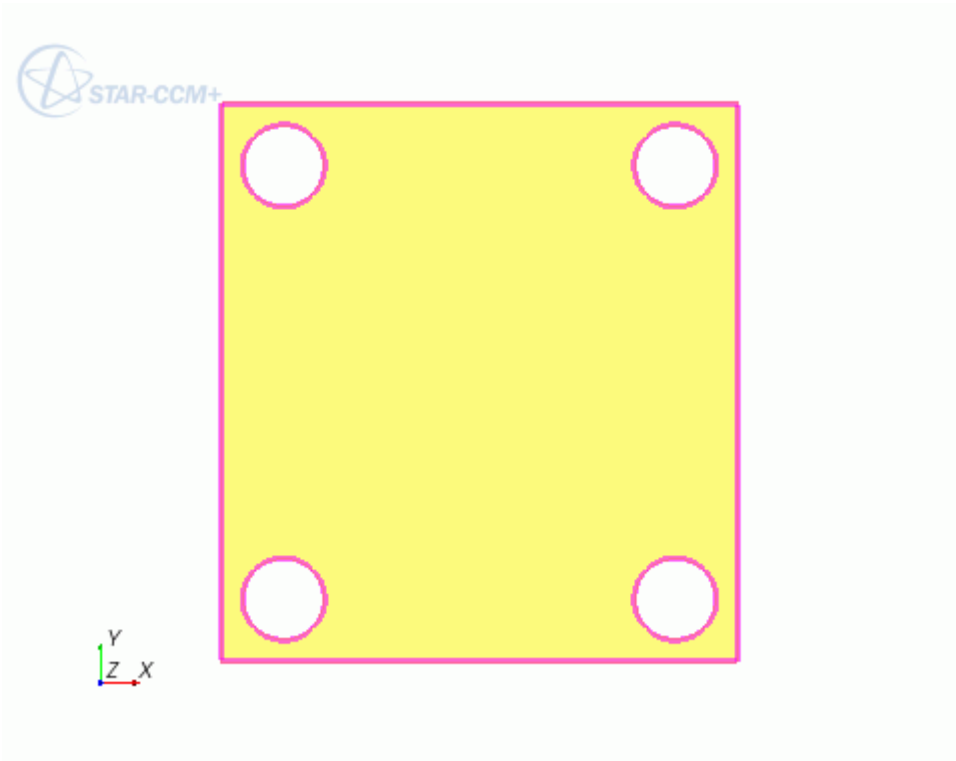
To create a sheet body from a sketch:

1. Right-click the desired sketch in the 3D-CAD feature tree and select **Create Sheets from Sketch**.



The *Convert Sketch to Sheet* panel appears. The selected sketch is listed in the panel.

A preview of the operation appears in the *3D-CAD View* scene.



2. Click **OK**.

Working with Bodies

Bodies represent the individual objects that make up the 3D-CAD model.

There are two types of bodies in 3D-CAD: Sheet Bodies and Solid Bodies. All bodies are stored under the **Bodies** manager node for the model, and can be operated on by executing the commands that are shown below. The faces of a body can also be operated on, and are accessed by selecting them in the *3D-CAD View* scene.

You can perform the following actions with a 3D-CAD body:

- Rename
- Delete
- Show/Hide
- Duplicate
- Creating Sheet Bodies from a Solid Body
- Sewing Sheet Bodies Together
- Shelling a Body
- Thickening a Body
- Transform (including Move, Scale, Rotate, and Mirror)
- Boolean (including Unite, Intersect, Subtract, Imprint, and Slice)
- External volume extraction

Some features require you to select multiple bodies. The selection group boxes, located in the relevant feature panel, are designed for this purpose.

You can import individual, or groups of bodies into the main simulation. You can also export bodies from 3D-CAD as a neutral CAD format. The following options are available from the main simulation tree as well as 3D-CAD:

- Exporting Geometry from 3D-CAD
- Creating Geometry Parts from 3D-CAD

Contents:

Overview of Bodies in 3D CAD
Naming a Body
Deleting a Body
Hiding Bodies
Duplicating a Body
Creating Sheet Bodies from Solid Bodies
Sewing Sheet Bodies Together
Shelling a Body
Thickening a Body

Transforming a Body
Performing Boolean Operations
Imprinting Sketches onto Bodies

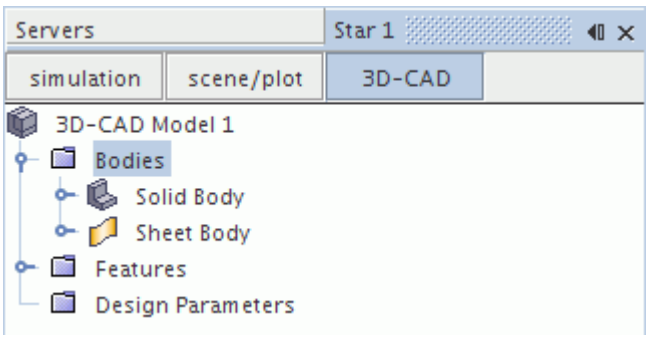
See Also:

Faces, Edges, and Vertices in 3D-CAD
Working with Faces, Edges, and Vertices

Overview of Bodies in 3D CAD

In 3D-CAD, a body is a single object, formed from a series of operations and features.

You can have two types of bodies: Solid or Sheet (zero-thickness). Bodies that you create and import in 3D-CAD are stored under the **Bodies** manager node in the 3D-CAD model tree.

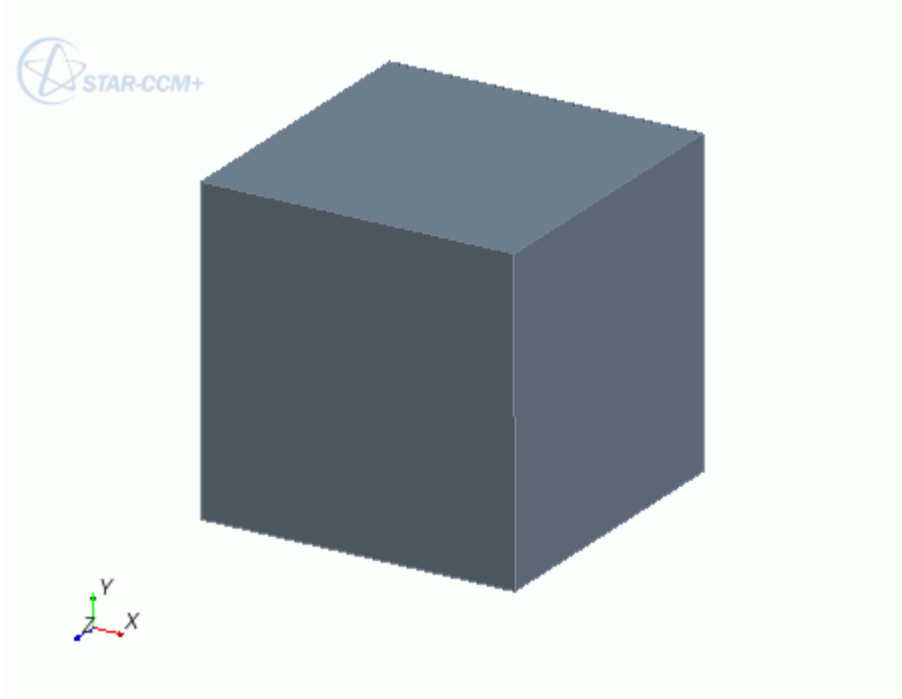


When you begin modeling in 3D-CAD, you create bodies from sketches, or by importing geometry.

Solid Bodies in 3D CAD

Solid bodies are closed bodies that have a finite volume.

Solid bodies appear in a gray color in the *3D-CAD View* scene, and identified by a icon in the **Bodies** manager node.

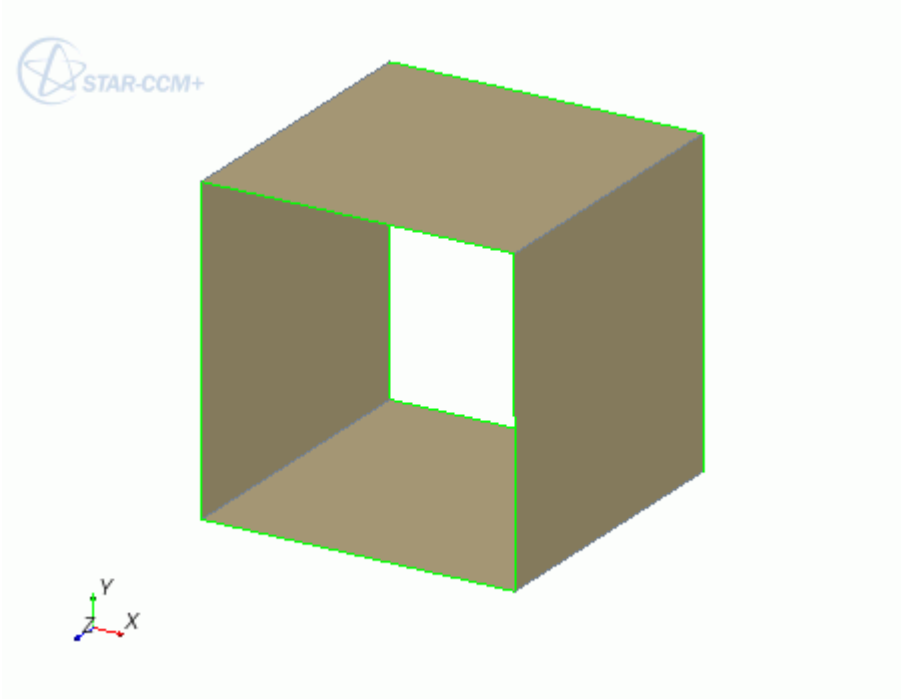


The following actions are available for solid bodies:

Sheet Bodies in 3D-CAD

Sheet bodies are composed from surfaces that have zero-thickness.

Sheet bodies appear in a brown color in the *3D-CAD View* scene, and are identified by a icon in the **Bodies** manager node.



The green edges on a sheet body correspond to free edges, and indicate the outer perimeter of the sheet body. See Free Edge. The following actions are available for sheet bodies:

See Also:

Sketches

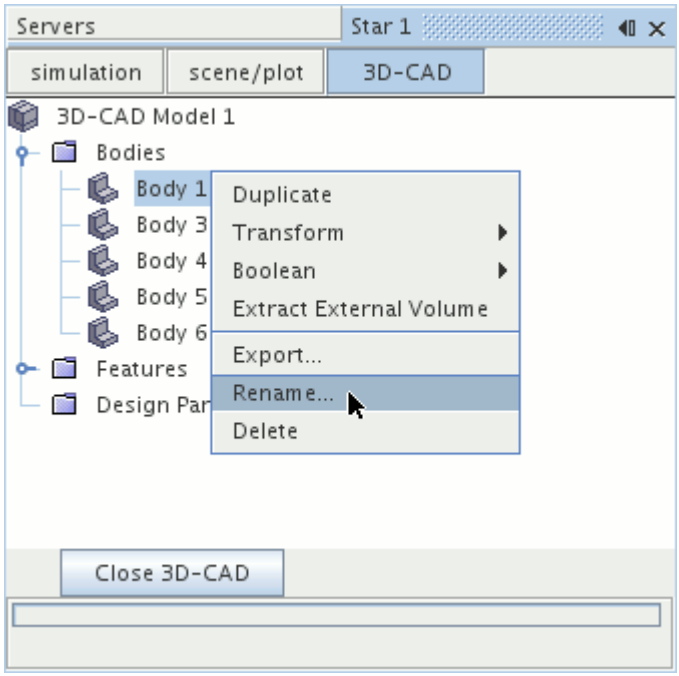
Importing Geometry into 3D-CAD

Naming a Body

The names that are given to bodies in 3D-CAD are preserved when you use them to create geometry parts. Bodies that have no given name are labelled as **Default**.

To set the name of a body:

1. In the **Bodies** manager node, right-click on the relevant body node and select **Rename**.



2. In the inline editor, enter the new name for the body.
3. To confirm the change, press `<Enter>`.

See Also:

Creating and Updating 3D-CAD Parts
Working with Custom Name Attributes

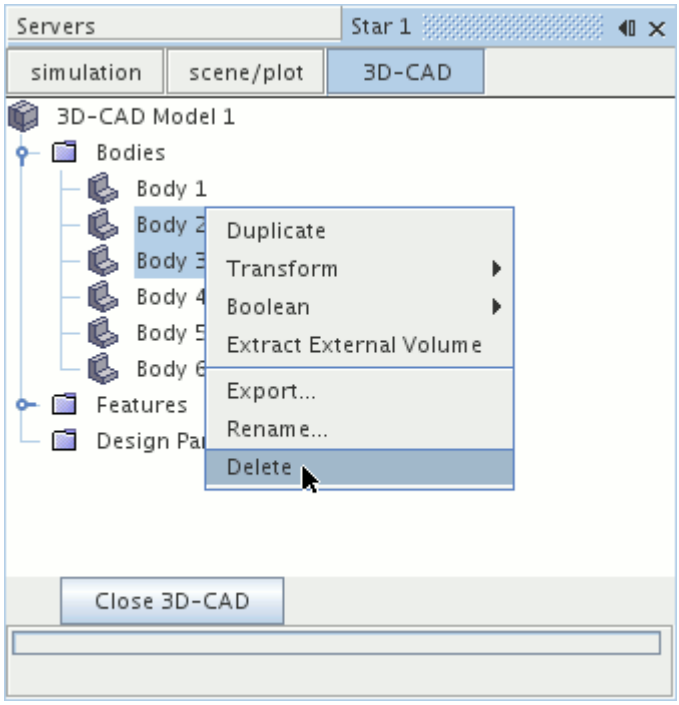
Deleting a Body

When you delete a body, it is removed from the 3D-CAD model. In 3D-CAD, there are two methods that you can use to delete an object:

To delete a body:

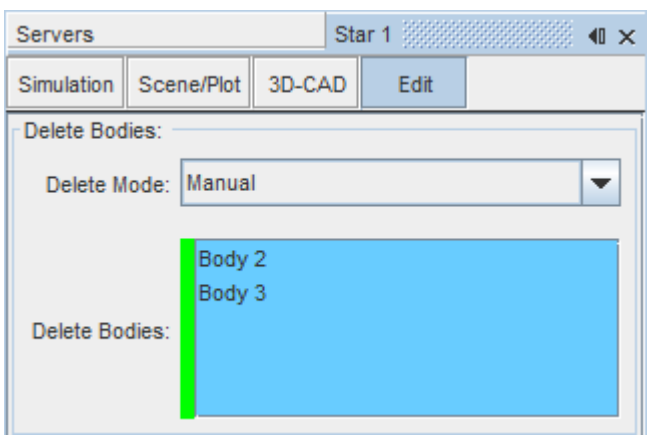
1. In the **Bodies** manager node, right-click on a body node and select **Delete**.

   To select multiple bodies, hold down the `<Ctrl>` key.



Alternatively, right-click the body in the *3D-CAD View* scene and select **Body** > **Delete**.

The *Delete Bodies* panel appears.



2. Choose the *Delete Mode* you want to use:
   - **Manual** — Choose this method to manually select the bodies to delete.
   - **Automatic** — Choose this method to automatically delete bodies using filters. For example, you can remove all solid bodies that have a volume smaller than 1 m^3 but keep all other bodies in the geometry model.
3. Select the bodies to delete:
   - For manual deletion, select bodies interactively from the *3D-CAD View* scene or from the **Bodies** manager node.

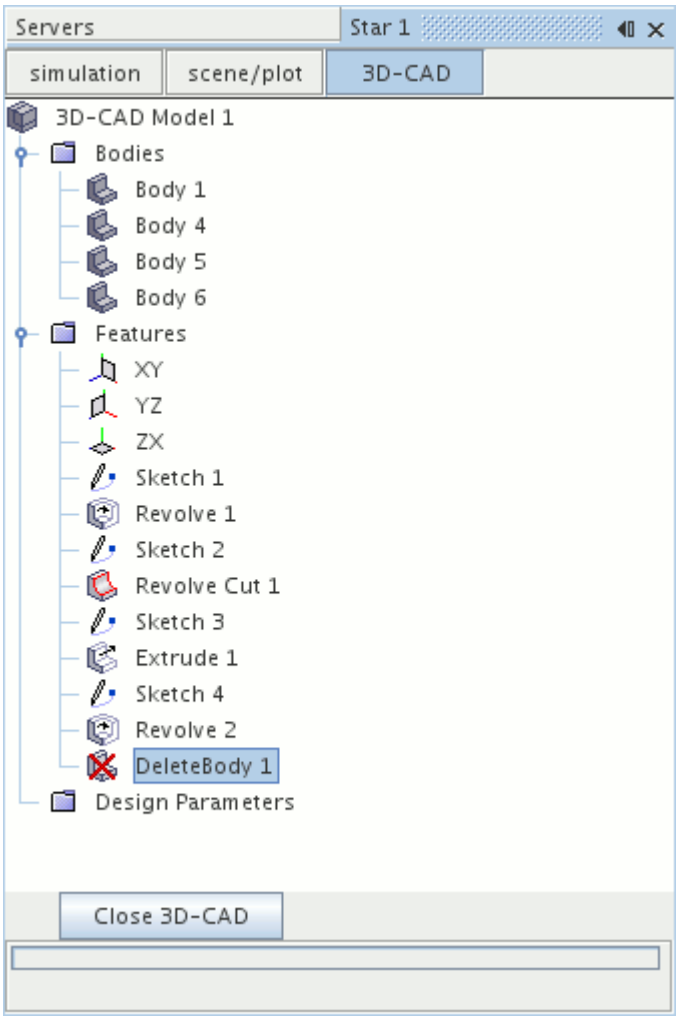
     Hold down the <`Ctrl`> key and select any additional bodies that you want to delete.

To select concealed bodies, hide faces on the obstructive parts. See Hiding Faces.

- For automatic deletion, see Delete Bodies Panel.

4. Click **OK**.

The selected bodies are deleted from the model, and a single **DeleteBody** node is added to the feature tree.



- Alternatively, use the <Delete> key.

Contents:

Delete Bodies Panel

Delete Bodies Panel

Use the automatic delete mode of the Delete Bodies operation to delete multiple bodies in a single operation.

Delete Mode	Specifies the mode for the Delete operation. Select **Manual** to manually select what bodies to delete. Select **Automatic** for the operation to search for and remove multiple bodies in a single operation.

Solids	When active, searches for solid bodies to delete based on the **Volume** and search options. Bodies that match the search criteria are deleted.	
Sheets	When active, searches for sheet bodies to delete based on the **Area** and search options. Bodies that match the search criteria are deleted.	
Search Options	Specifies what range of values to search for when deleting bodies.	
	Below	Deletes bodies with values below the specified amount.
	Between	Deletes bodies with values between the specified range.
	Above	Deletes bodies with values above the specified amount.
	Not Between	Deletes bodies with values outside the specified range.
Preview Mode	Specifies what to display in the *3D-CAD View* scene.	
	Result	Displays the result of the operation. This is the default preview behavior, where the result of the operation is displayed in yellow.
	Candidate Bodies	Displays the bodies that are identified for deletion by the Delete operation. Candidate bodies are displayed in blue. The corresponding bodies are listed in the group box below.

Hiding Bodies

The hide feature allows you to hide bodies temporarily from the *3D-CAD View* scene.

You can also hide bodies while performing a CAD operation. This function is useful for gaining a clear view of other CAD components. You can restore hidden bodies again by using the show feature. If you want to hide a particular face on a body, use the hide face action. See Hiding Faces.

Hiding a Body

To hide a body:

- Right-click the body from within the **Bodies** manager node and select **Hide**. You can select multiple bodies by holding down the `<Ctrl>` key.
- Alternatively, right-click the body in the *3D-CAD View* scene and select **Body** > **Hide Body**.

In the feature tree, you can identify hidden bodies by the grayed out icon .

To hide all bodies other than the selected ones:

- Right-click the body from the **Bodies** manager node and select **Show Only**.
- Alternatively, right-click the body in the *3D-CAD View* scene and select **Body** > **Show Only**.

All bodies, other than those selected, are hidden. If you choose to show only a body that is already hidden, the operation shows this body and hides all other bodies.

Restoring a Body

To restore a body:

- Right-click the body from within the **Bodies** manager node and select **Show**. You can select multiple bodies by holding the <`Ctrl`> key.
- Alternatively, while performing a CAD operation in *Edit* mode, right-click a blank area in the *3D-CAD View* scene and select **Restore Hidden Bodies**.

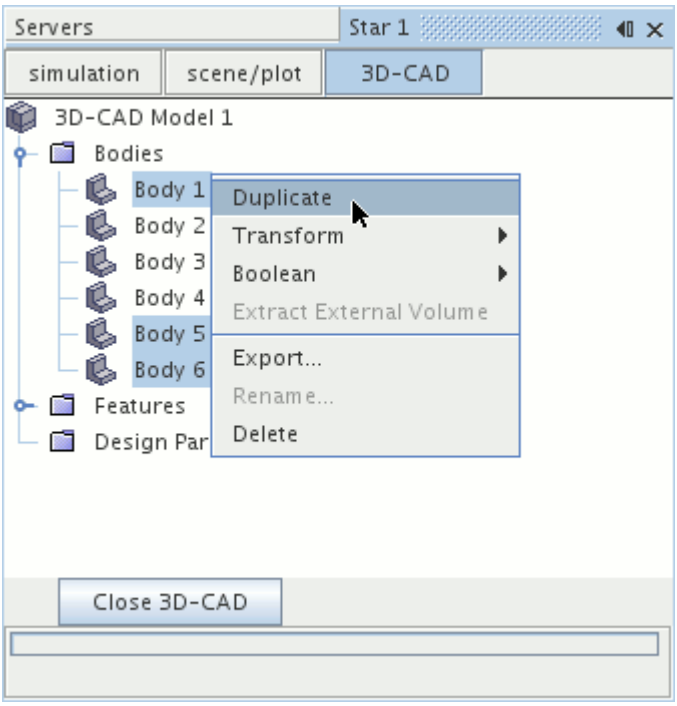
When you restore a hidden body in the *3D-CAD View* scene, the body icon returns to normal .

Duplicating a Body

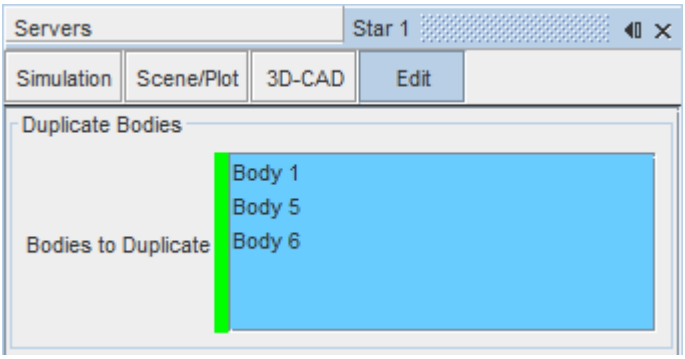
When you duplicate a body, an identical copy of the body is created and added to the **Bodies** manager node. The new body is given a default name, for example `Body 5`, and can be operated on. It is not necessary to duplicate a body before transforming it. In this case, use the copy option in the desired transform panel. See Creating a Copy During the Transform Operation. It is possible to duplicate several bodies at once, as described below.

To duplicate a body:

1. In the **Bodies** manager node, right-click on a body node and select **Duplicate**. You can select multiple bodies by holding down the <`Ctrl`> key.

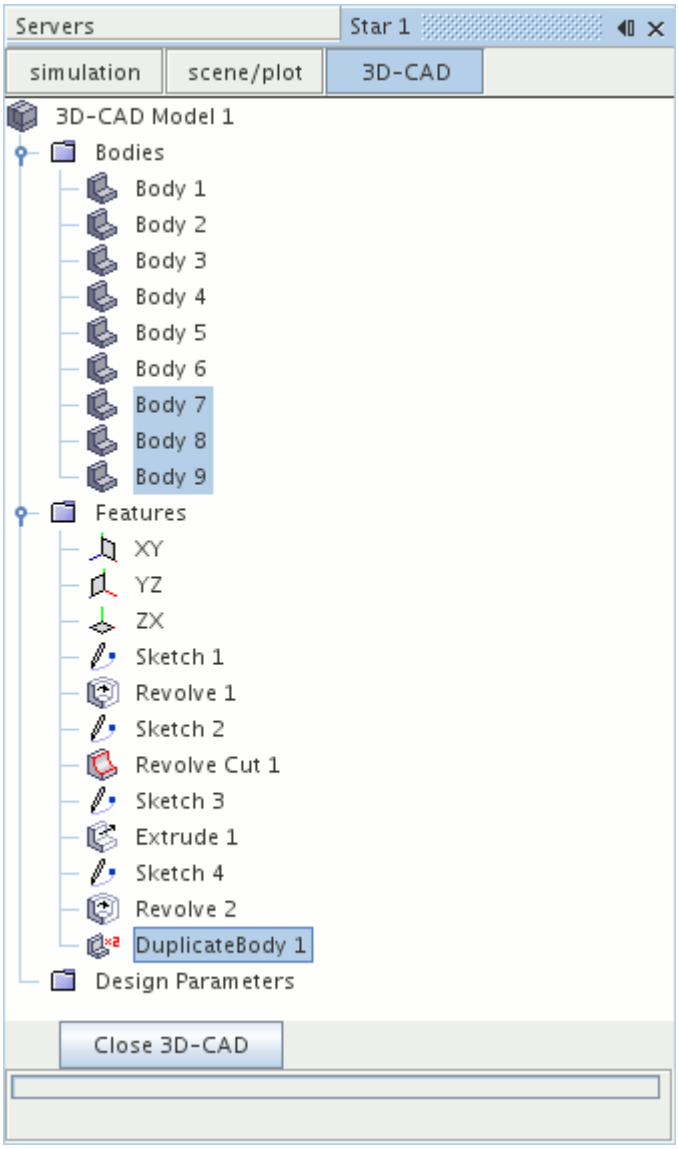


The *Duplicate Bodies* panel appears. The selected bodies are shown in the *Bodies to Duplicate* group box.



2. Click **OK**.

The selected body is duplicated and a body node is added to the object tree for each of the selected bodies. A single **DuplicateBody** feature node is also added to the feature tree.



See Also:

Transforming a Body

Creating Sheet Bodies from Solid Bodies

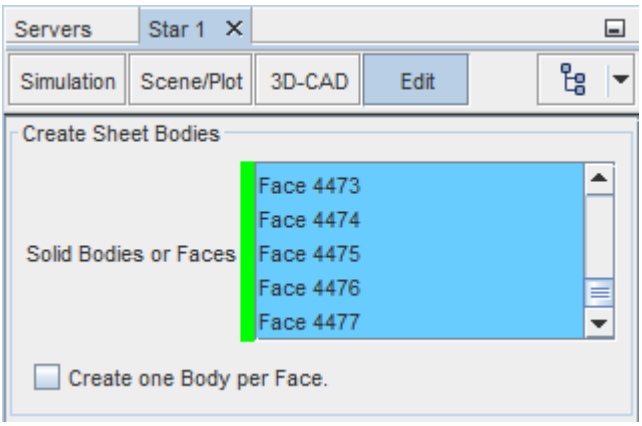
You can copy the faces off of a solid body to create sheet bodies. You can copy the entire solid body, or only specific faces. You can choose to create a single sheet body from all faces, or a minimum number of separate sheet bodies. The number of separate sheet bodies that are generated depends on the topological connectivity of the geometry. The original solid body remains unchanged.

To create sheet bodies from solid bodies (or faces):

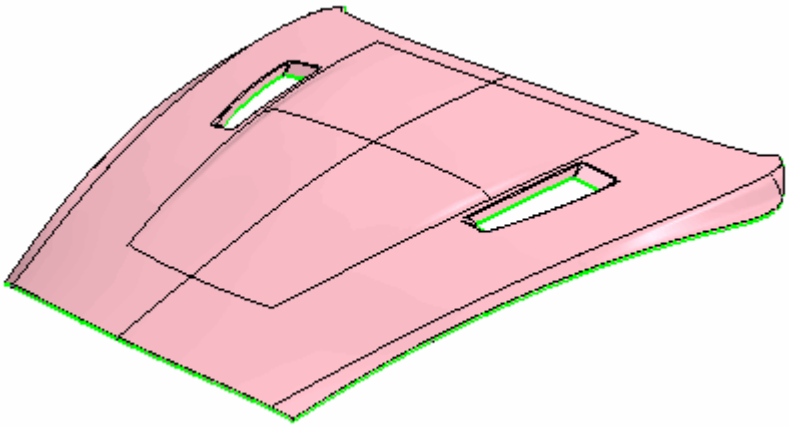
1. Select the bodies (or faces) that you want to convert, using one of the following methods:
   - Select one or more body nodes in the **3D-CAD Bodies** node.
   - Select one or more faces in the *3D-CAD View* scene.
2. Right-click one of the selected items and choose **Create Sheet Bodies** from the context menu.

   In the *3D-CAD View* scene, the context menu provides the Create Sheet Bodies action in both the Faces and the Bodies sections. If you choose the option from the Faces section, the selected faces are converted to sheet bodies. If you choose the option from the Bodies section, all faces on all selected bodies are converted to sheet bodies.

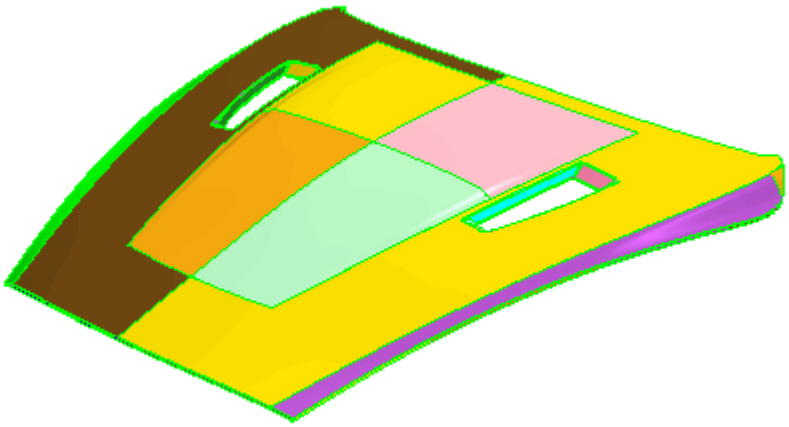
   The *Create Sheet Bodies* dialog appears.



3. Choose whether you want to create a single body from all faces, or separate bodies for each face.
   - To create a single sheet body for all faces, deactivate the **Create one Body per Face** option. This option is deactivated by default.

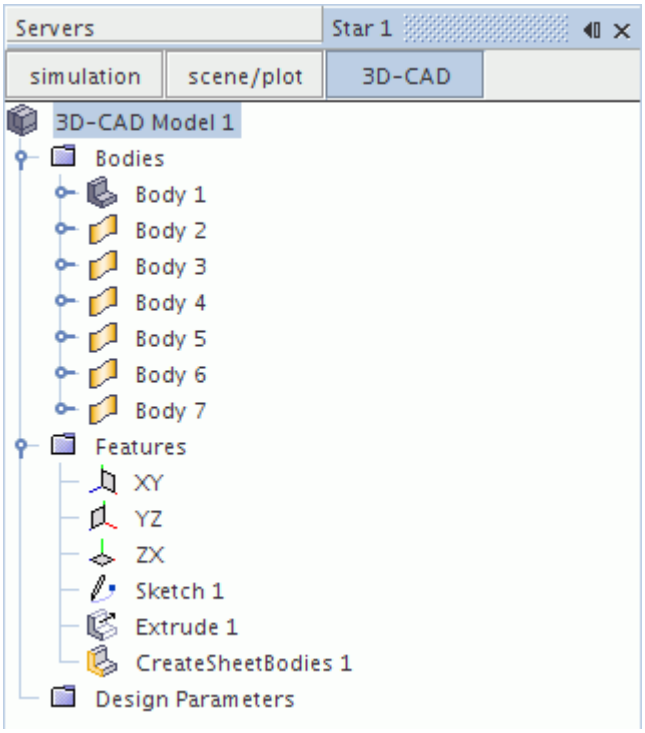


- To create the minimum number of separate sheet bodies, activate the **Create One Body per Face** option.



4. Click **OK** to complete the feature.

Depending on the option you choose, a single sheet body or multiple sheet bodies are created and placed in the **Bodies** manager node. A **CreateSheetBodies** feature node is added to the feature tree.



Sewing Sheet Bodies Together

When the free edge of a sheet body coincides with the free edge of an adjacent sheet body, STAR-CCM+ can sew the edges together to form a single sheet body. Similarly, if two free edges on the same sheet body overlap, STAR-CCM+ can merge them with the Sew function.

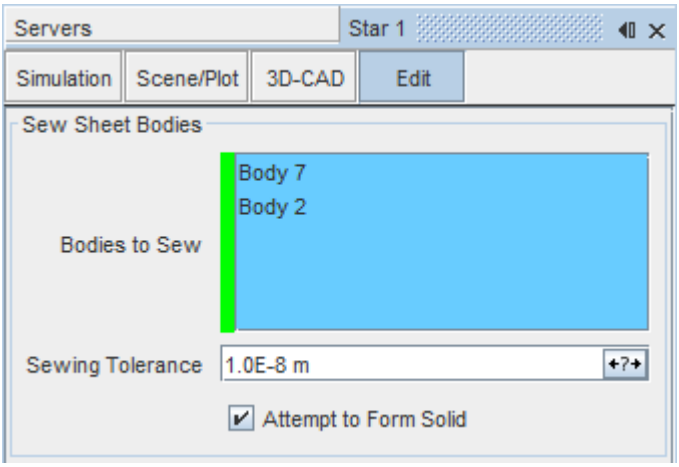
If the sheet bodies form a closed geometry, you have the option to convert the sheet bodies into a solid body.

Only bodies whose edges are within the specified tolerance can be sewn together. The maximum recommended tolerance for sewing is 1.0E-5 m. To fill gaps larger than this maximum tolerance, use the Fill Surface feature instead. See Filling Holes.

To sew sheet bodies:

1. Select the sheet bodies to sew using one of the following methods:
   - Select one or more sheet bodies in the *3D-CAD View* scene.
   - Select one or more sheet body nodes in the 3D-CAD **Bodies** node.
2. Right-click on a selection and choose **Sew Sheet Bodies**.

   The *Sew Sheet Bodies* panel appears.



3. Define the sew parameters. See Sew Sheet Bodies Panel.
4. Click **OK**.

Contents:

Sew Sheet Bodies Panel

See Also:

Faces, Edges, and Vertices in 3D-CAD

Sew Sheet Bodies Panel

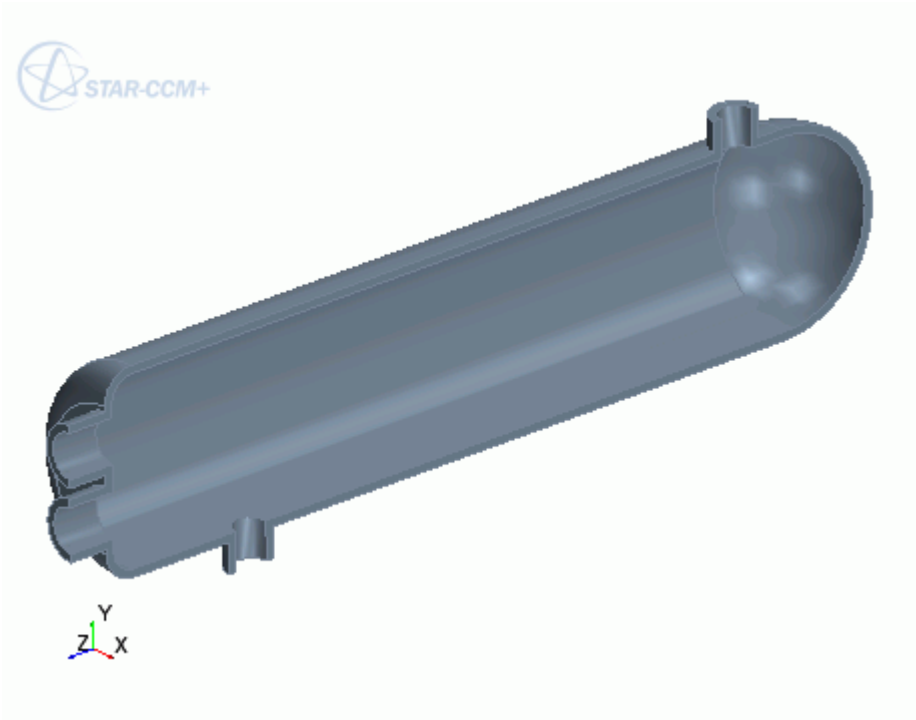
The Sew Sheet Bodies action merges sheet bodies whose edges coincide. For a set of sheet bodies that form a closed volume, you can choose to convert them to a solid body.

Bodies to Sew	Specifies the sheet bodies to sew.
Sewing Tolerance	Defines the maximum distance that can exist between the edges of adjacent sheet bodies. If the sheet bodies have larger gaps between them, increase the tolerance accordingly.
Attempt To Form Solid	When activated, the Sew Sheet Bodies action attempts to form a solid body from the volume enclosed by the selected sheet bodies.

Shelling a Body

The Shell tool converts a solid body into a thin-walled, hollow, solid body.

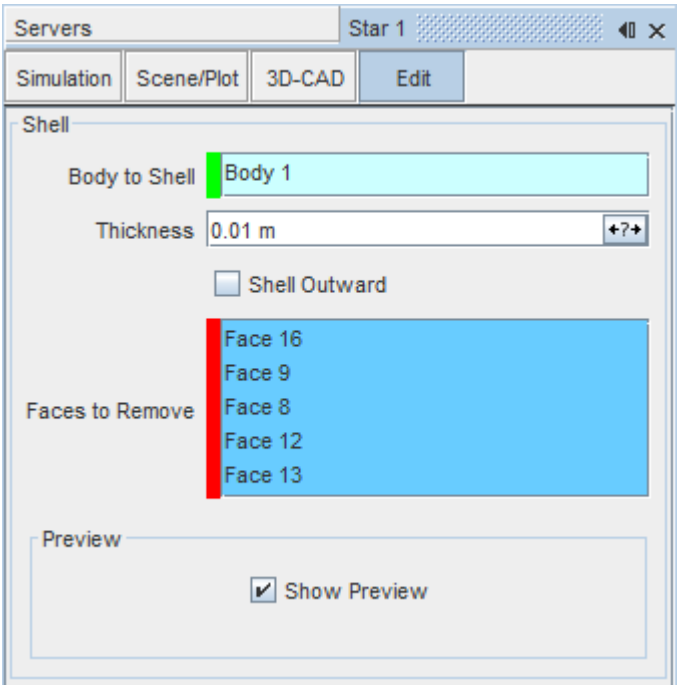
You can choose to remove faces and expose the hollow volume of the solid. The shell tool can extrude the original walls inwards or outwards according to your choice.



To shell a solid body:

1. Select the body to shell using one of the following methods:
   - In the 3D-CAD feature tree, right-click the body and select **Shell**.
   - In the graphics window, right-click the body to shell and select **Shell**.

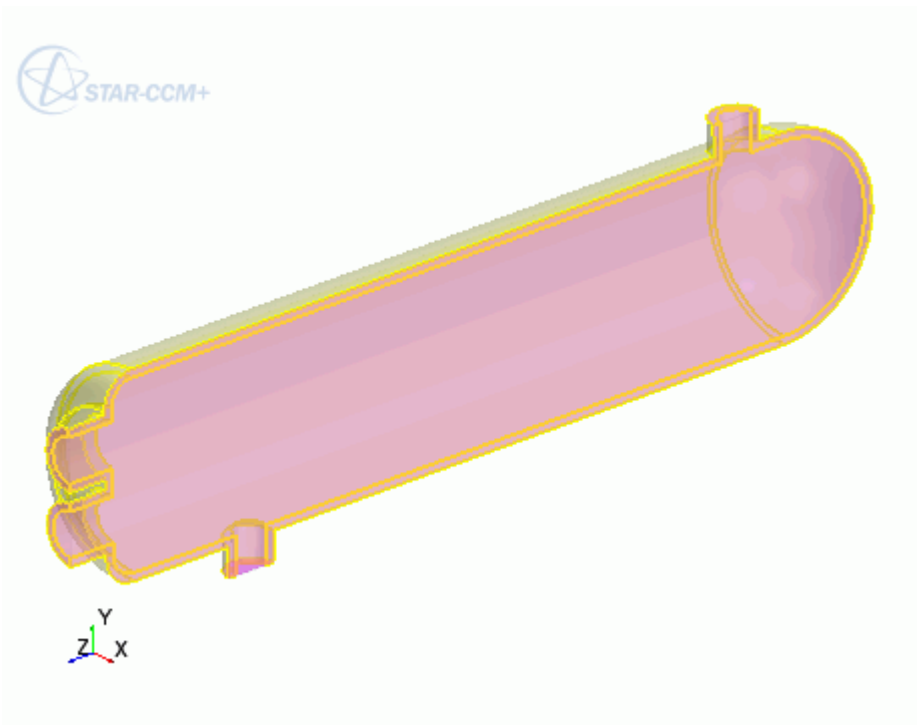
   The *Shell* panel appears.



2. Define the shell parameters. See Shell Panel.

3. If you want to remove any faces, use the **Select From Scene** tool.

   A preview of the operation is displayed in the graphics window.



4. Click **OK**.

The solid body is converted to a thin-walled structure and a **Shell** feature is added to the feature tree.

Contents:

Shell Panel

Shell Panel

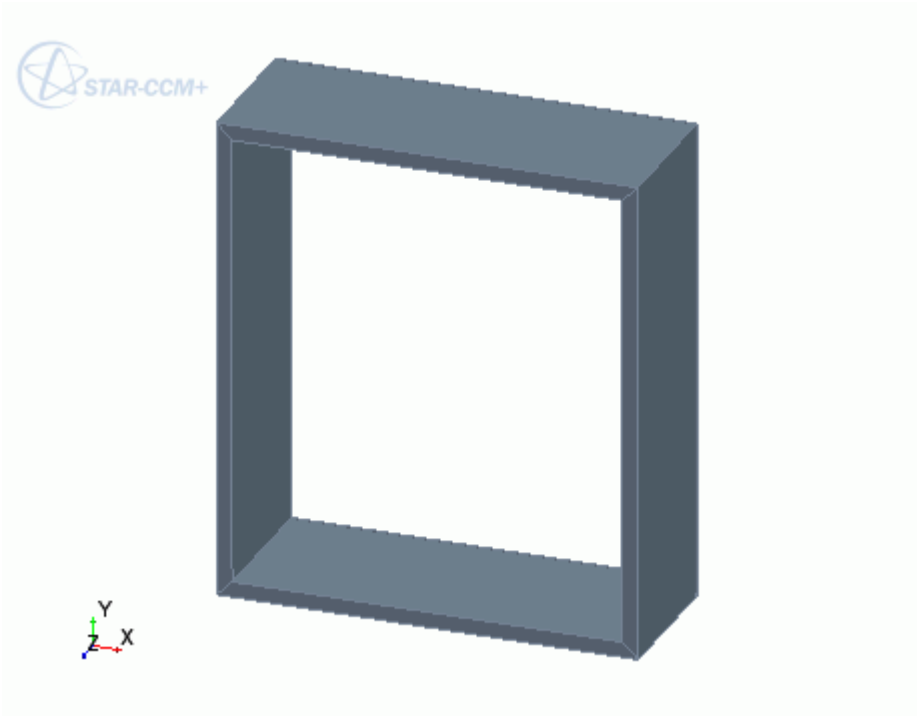
The Shell action hollows out a solid body. You specify the wall thickness of the shell feature.

Body to Shell	Specifies the solid body to shell. You can shell only one solid body per operation.
Thickness	Defines the thickness of the solid walls.
Shell Outward	When activated, the Shell action uses the existing surface of the solid as the internal surface of the shell. Otherwise, the tool creates a new internal surface that is offset inwards from the existing surface of the solid.
Faces to Remove	Removes the selected faces from the solid body and exposes the hollow volume.

Thickening a Body

You can thicken a sheet body by a specified amount.

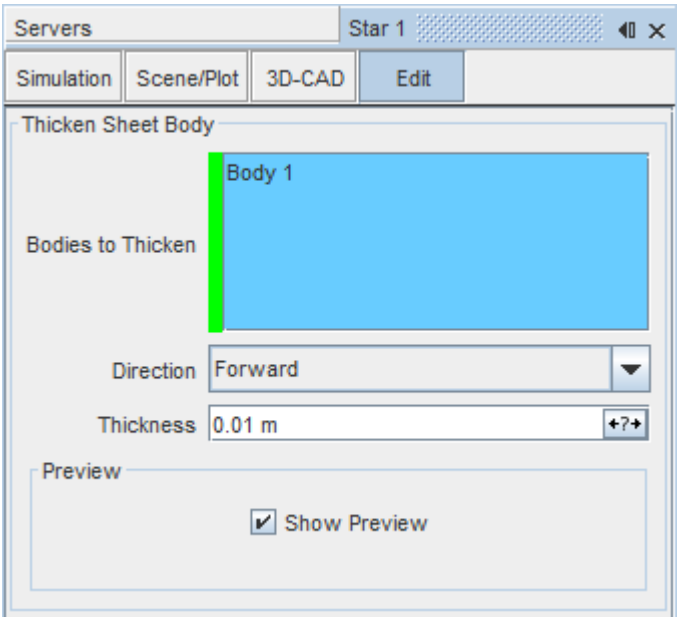
This operation converts the sheet body into a solid body. You can thicken multiple bodies in a single operation.



To thicken a sheet body:

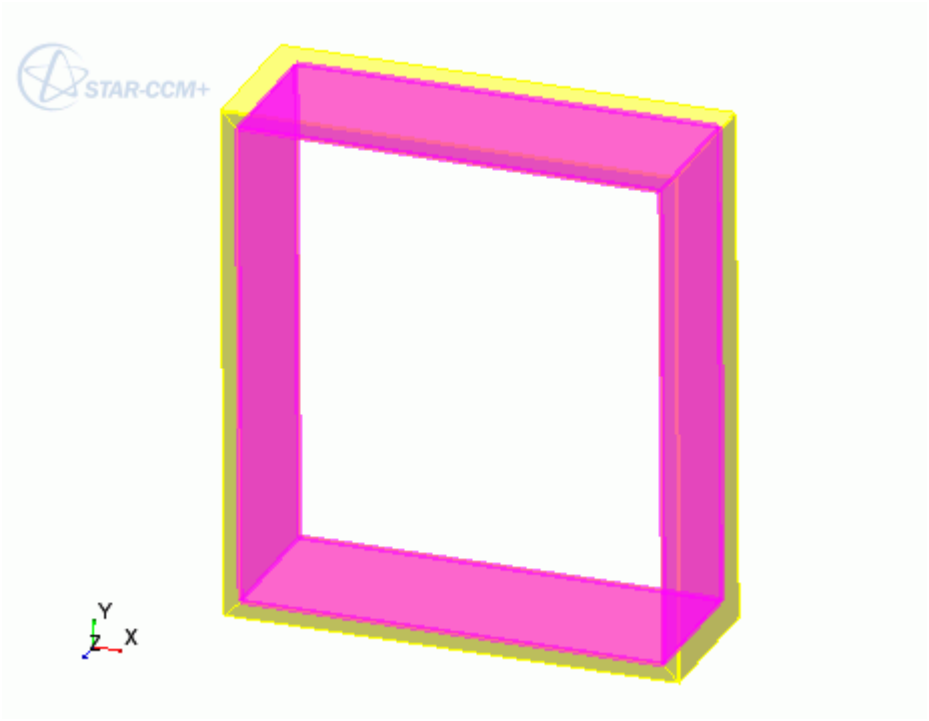
1. Right-click a sheet body in the *3D-CAD View* scene and select **Thicken**.

   The *Thicken Sheet Body* panel appears.



2. Set the parameters for the thicken operation. See Thicken Sheet Body Panel.

   A preview of the operation is displayed in the graphics window.



3. Click **OK**.

The sheet body is thickened and converted to a solid body. A **ThickenSheetBodies** feature is also added to the feature tree.

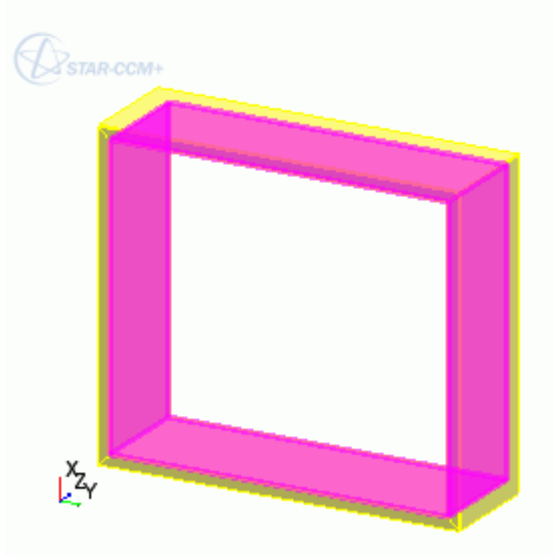
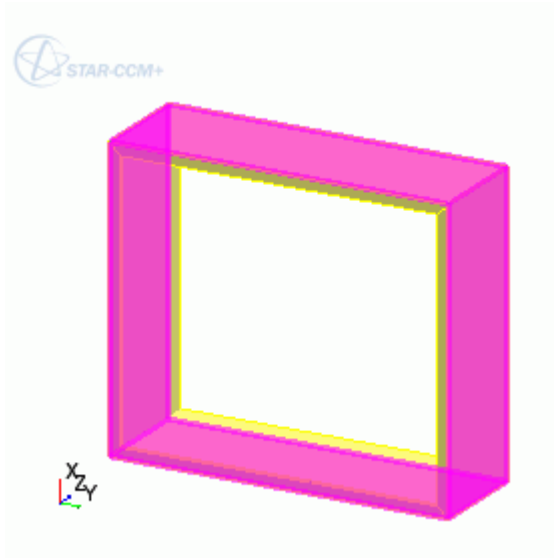
Contents:

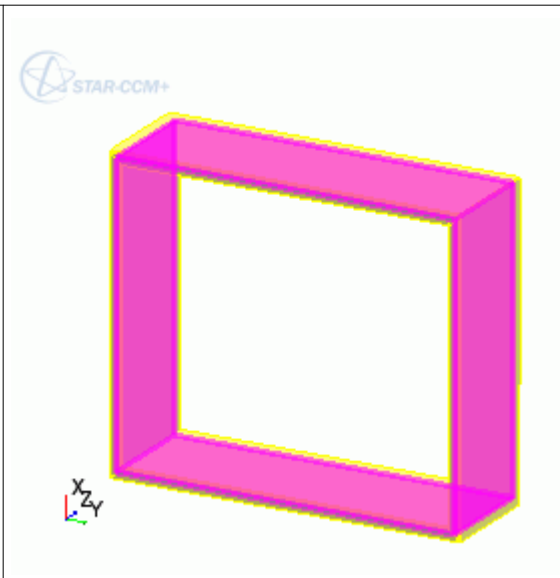
Thicken Sheet Body Panel

Thicken Sheet Body Panel

The Thicken action adds thickness to a sheet body, converting it into a solid body.

Bodies to Thicken	Specifies the bodies to thicken.	
Direction	Specifies the direction to thicken the sheet body.	
	Forward	Thickens the sheet body along the face normal.

		
	Backward	Thickens the sheet body against the face normal. 
	Symmetric	Thickens the sheet body by an equal amount on both sides. The total thickness of the body is the specified thickness.

		
Thickness	Specifies the amount to thicken the sheet body.	

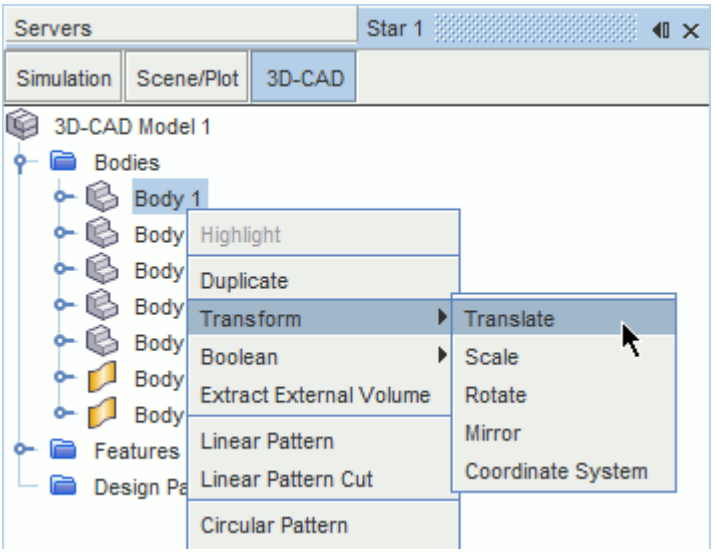
Transforming a Body

You can transform bodies by performing the following operations. It is possible to create a copy of the original body during a transform operation.

- Translate
- Scale
- Rotate
- Mirror
- Coordinate System (Transform)

A general workflow to follow when transforming a body is provided below. In the following example, a translation transformation is applied.

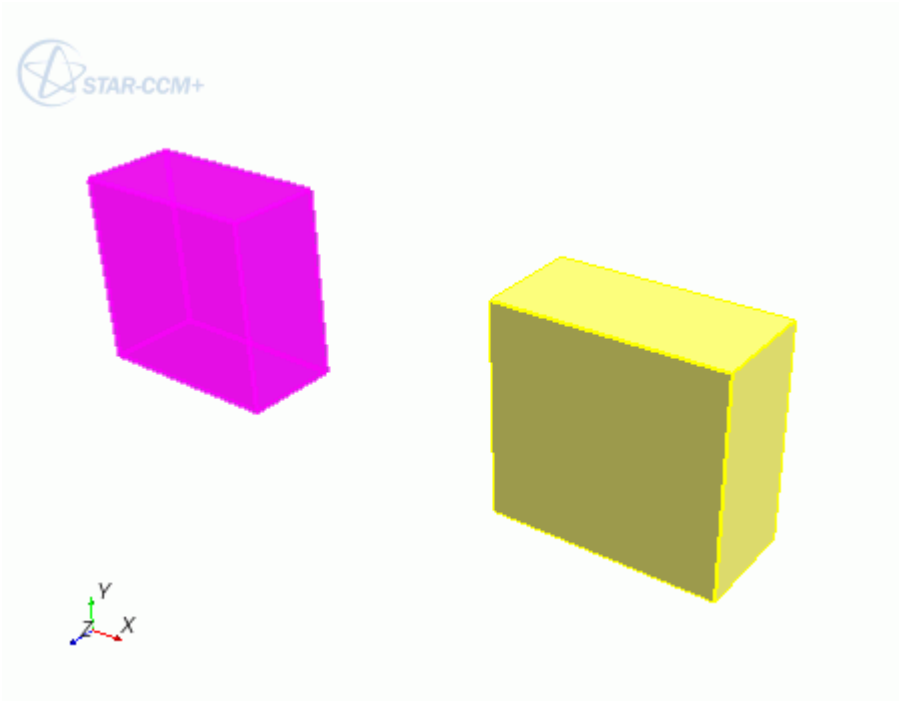
1. In the **Bodies** manager node, right-click on the desired body and select **Transform** > **[Operation Name]**. To select multiple bodies, hold down the <`Ctrl`> key.



The relevant panel appears for the transformation.

2. Define the properties for the transformation. See the relevant sections:
   - Translate Bodies
   - Scale Bodies
   - Rotate Bodies
   - Mirror Bodies
   - Coordinate System (Transform)
3. Expose any properties as design parameters by clicking the (**Expose Parameter**) button.
4. Choose whether to apply the transformation to the original body, or to a copy of the original. See Creating a Copy During the Transform Operation.

   A preview of the transformation is shown in the *3D-CAD View* scene.



5. Click **OK**.

The transformation is applied to the selected bodies. Each operation creates a feature node in the 3D-CAD feature tree.

Contents:

Translating a Body
Scaling a Body
Rotating a Body
Mirroring a Body
Transforming a Body Using Coordinate Systems
Creating a Copy During the Transform Operation

Translating a Body

Translating a body moves it from one position to another.

You can translate multiple bodies simultaneously. The *Translate Bodies* panel options are listed below.

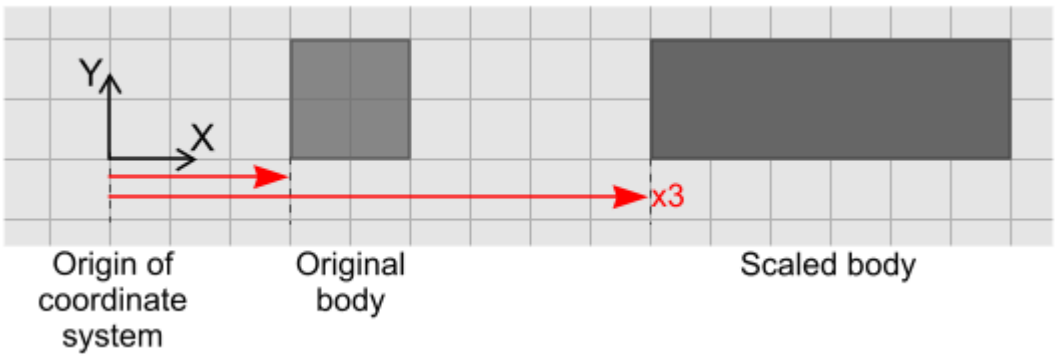
Translation Vector	Defines the direction of translation. Enter the [X, Y, Z] components of the translation vector
Coordinate System	Defines the coordinate system for the transformation. If you have specified an alternative coordinate system within the simulation, you can define the transformation with reference to this coordinate system instead of the default, **Laboratory**.

Action	Specifies whether to carry out the transformation on the original body or a copy of the original body. See Creating a Copy During the Transform Operation.

Scaling a Body

Scaling a body proportionally increases or decreases its size with respect to the selected coordinate system.

You can scale multiple bodies at once. When a body is scaled, the scale factor is applied from the origin of the chosen coordinate system, as shown in the following diagram.



The origin of the coordinate system may not always be the same as the origin of the sketch plane. If you are using a coordinate system other than the default, you can check the coordinates of the origin by selecting the relevant node under **Tools** > **Coordinate Systems** in the main simulation tree. The origin is displayed in the *Properties* window.

Scale Body Panel

The *Scale Bodies* panel options are listed below.

Scale all dimensions uniformly	Specifies whether the selected bodies are scaled uniformly in all directions. If you want to scale the body uniformly in all directions, activate this option. The value that you enter in the first *Scale Factor* text box applies to the remaining text boxes (which are grayed out)
Scale Factor	Defines the amount that the bodies are scaled. Enter the scale factor for each direction: **X**, **Y**, and **Z.**
Coordinate System	Defines the coordinate system for the transformation. If you have specified an alternative coordinate system within the simulation, you can define the transformation with reference to this coordinate system instead of the default, **Laboratory**.

Action	Specifies whether to carry out the transformation on the original body or a copy of the original body. See Creating a Copy During the Transform Operation.

Rotating a Body

Rotating a body turns it around a specified axis.

You can rotate several bodies around the same axis simultaneously. The *Rotate Bodies* panel options are listed below.

Rotate Axis	Defines the axis about which the selected bodies rotate.	
	Axis Direction	Defines the direction vector of the axis of rotation. Enter the `[X, Y, Z]` components for direction vector.
	Axis Position	Defines the origin of the axis of rotation. Enter the `[X, Y, Z]` coordinates of a point that lies on the axis of rotation.
Coordinate System	Defines the coordinate system for the transformation. If you have specified an alternative coordinate system within the simulation, you can define the transformation with reference to this coordinate system instead of the default, **Laboratory**.	
Angle	Specifies the angle of rotation.	
Action	Specifies whether to carry out the transformation on the original body or a copy of the original body. See Creating a Copy During the Transform Operation.	

Mirroring a Body

Mirroring a body reflects it about a specified plane.

You can mirror several bodies simultaneously.

Mirror Plane	Defines the plane about which the selected bodies are mirrored.	
	Normal	Defines the direction vector that is normal to the mirror plane. Enter the `[X, Y, Z]` components of this direction vector.
	Position	Defines a point that lies on the mirror plane. Enter the `[X, Y, Z]` coordinates of this point.
Coordinate System	Defines the coordinate system for the transformation. If you have specified an alternative coordinate system within the simulation, you can define the transformation with reference to this coordinate system instead of the default, **Laboratory**.	
Action	Specifies whether to carry out the transformation on the original body or a copy of the original body. See Creating a Copy During the Transform Operation.	

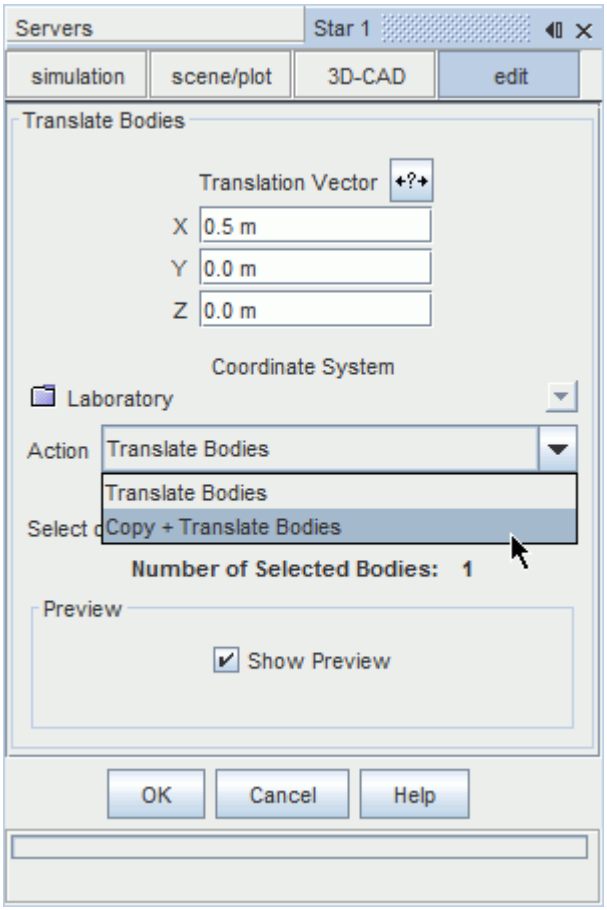
Transforming a Body Using Coordinate Systems

Use this operation to transform bodies from one coordinate system to another.

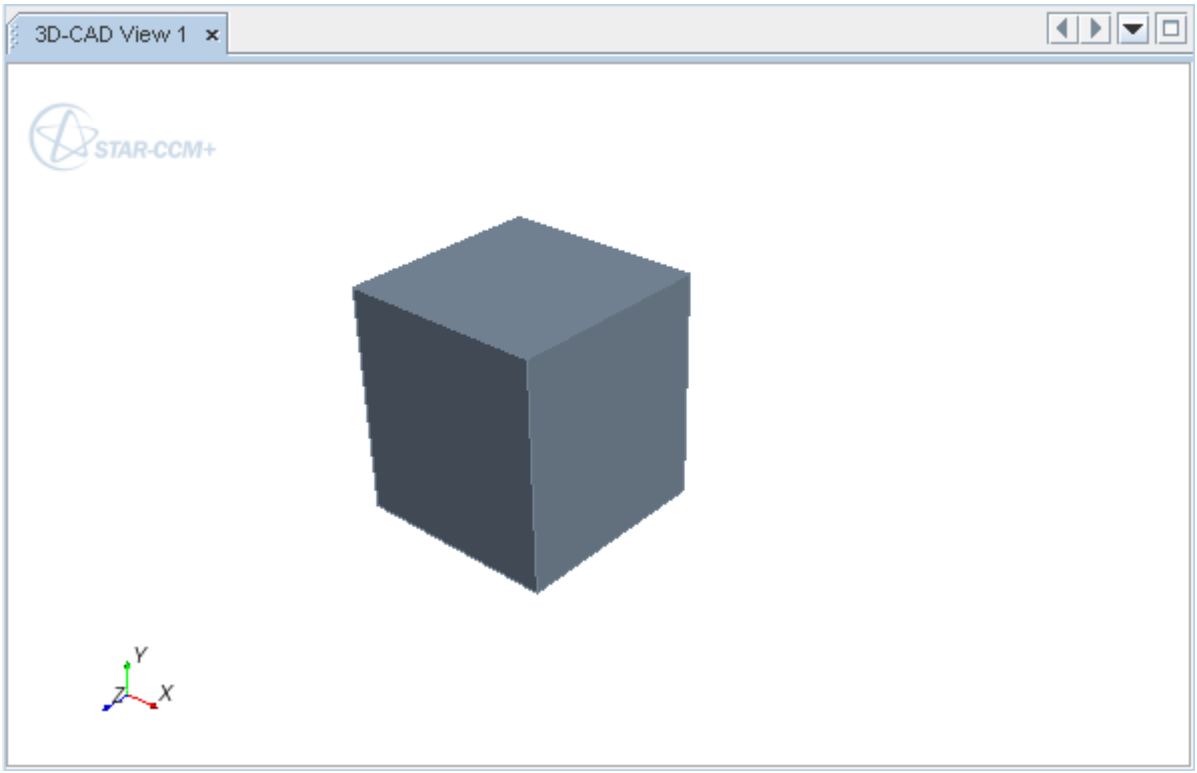
Bodies to Transform	Specifies the bodies to transform in the operation. Select bodies from the *3D-CAD View* scene or from the **Bodies** node in the feature tree.
From	Specifies the source coordinate system. Set **Coordinate System Source** to the coordinate system that you want to transform from. The default option is the imported coordinate system. If you choose a reference coordinate system, select this from the *3D-CAD View* scene or from the 3D-CAD feature tree.
To	Specifies the target coordinate system. Set **Coordinate System Source** to the coordinate system that you want to transform to. The default option is the imported coordinate system. If you choose a reference coordinate system, select this from the *3D-CAD View* scene or from the 3D-CAD feature tree.
Action	Specifies whether to carry out the transformation on the original body or a copy of the original body. See Creating a Copy During the Transform Operation.

Creating a Copy During the Transform Operation

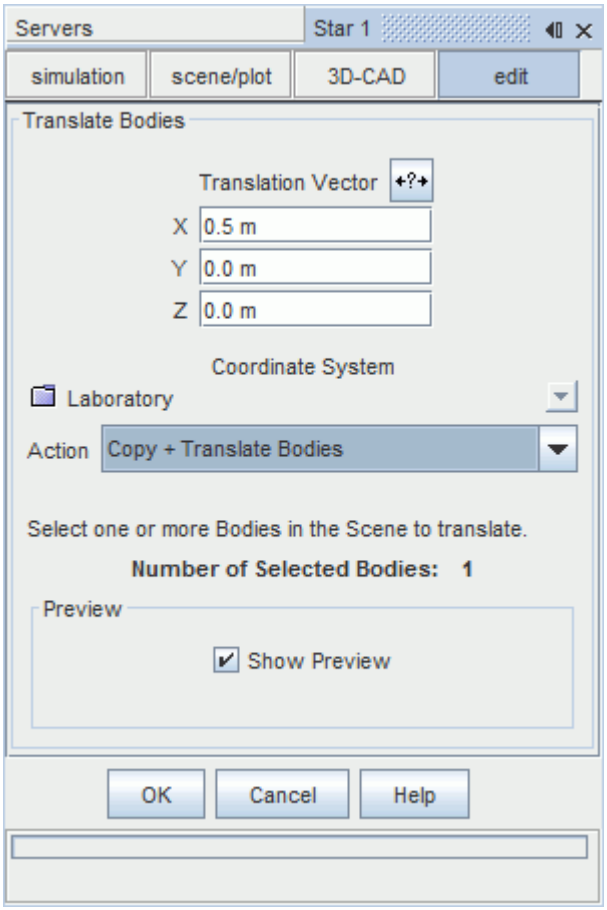
The **Copy** option eliminates the need for you to duplicate a body before transforming it. Each transform feature panel contains an **Action** drop-down list that allows you to make a copy of the original body and transform the copy, maintaining the original body.



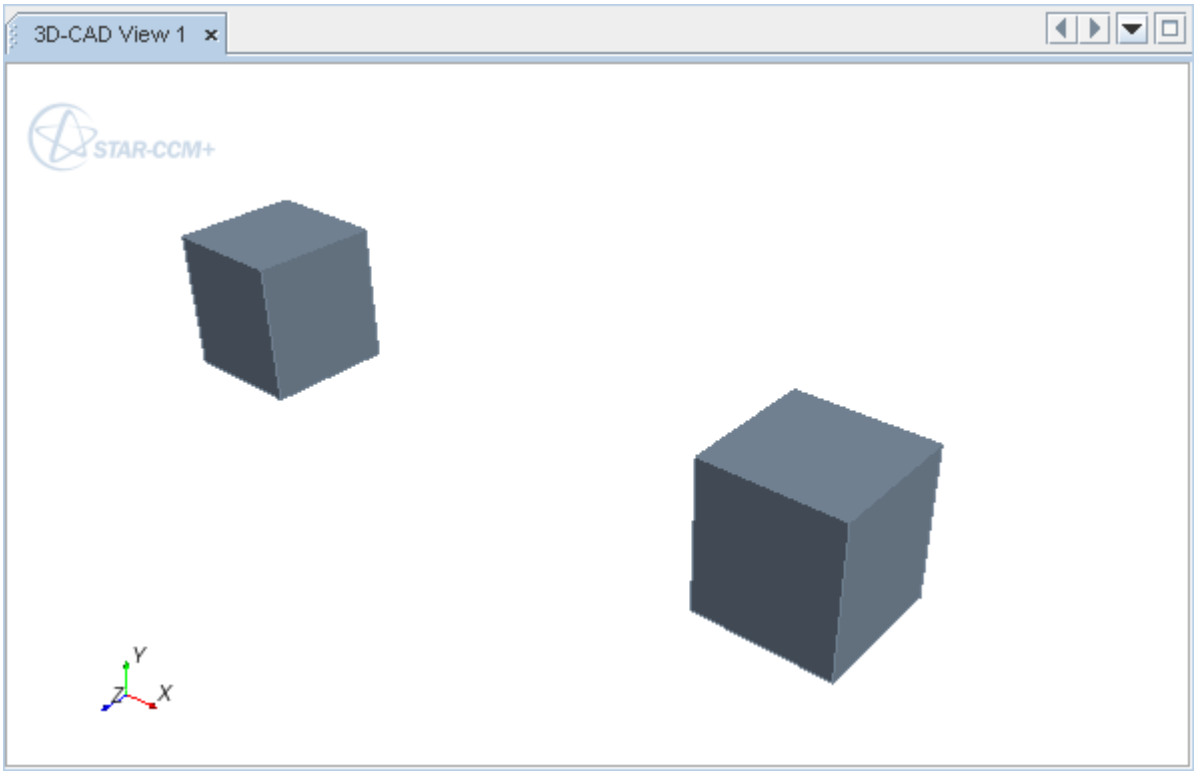
By default the original body is transformed. An example that demonstrates how the **Copy** option works is shown below.



A copy of the body is being translated `0.5 m` in the X direction.



The result of this operation is shown below. The original body has not moved; the transform operation translated the copy of the original body.



See Also:

Duplicating a Body

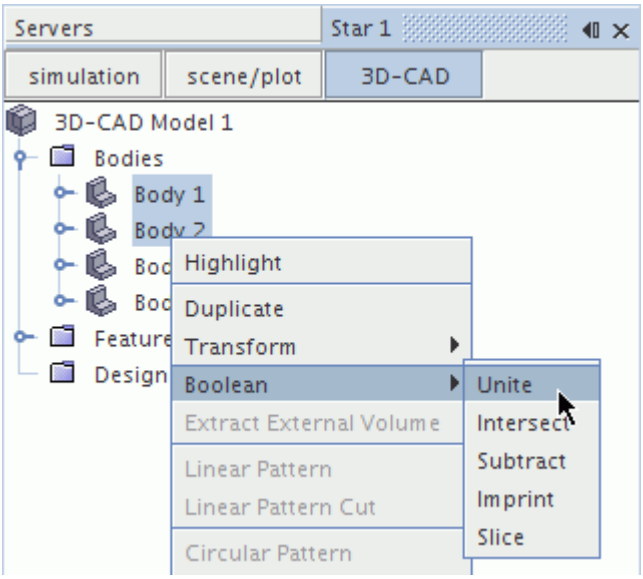
Performing Boolean Operations

You can perform the following Boolean operations in 3D-CAD:

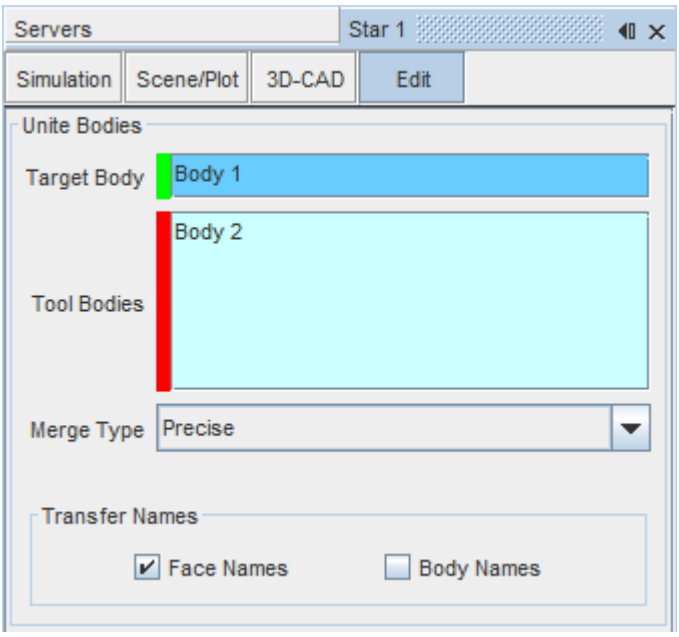
- Unite
- Intersect
- Subtract
- Imprint
- Slice

A general workflow to follow is provided below. In the following example, two bodies are united.

1. Hold down the <`Ctrl`> key and select the relevant bodies from the **Bodies** manager node.
2. Right-click on one of the selected bodies and select **Boolean** > **[Operation Name]**.



The relevant panel appears for the Boolean operation. The number of selected bodies is displayed here.

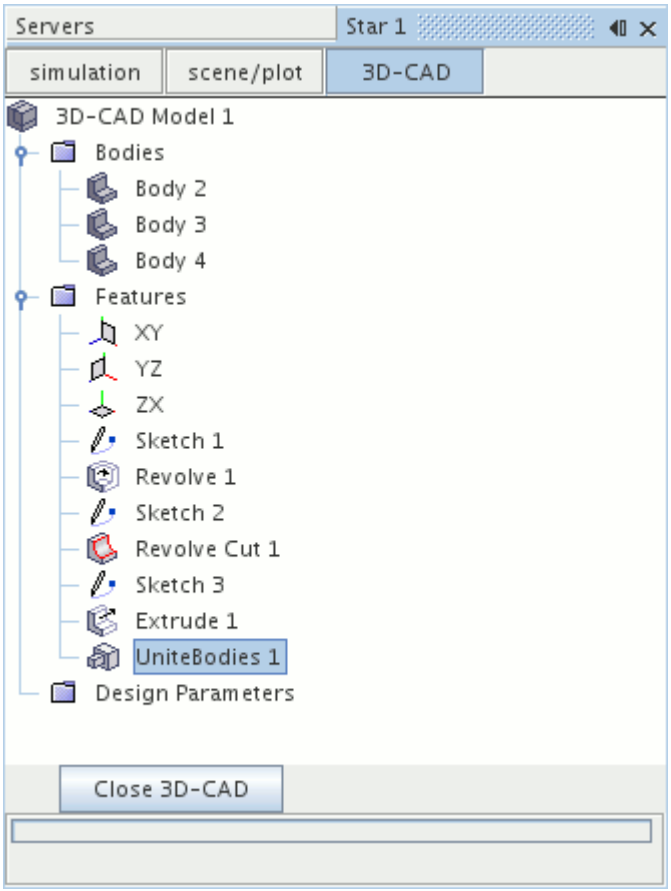


3. Define the properties for the Boolean operation.

   If a Boolean operation fails to complete successfully with the precise option, for example precise imprinting, try using the tolerant option.

4. Click **OK**.

A feature node is added to the feature tree for each Boolean operation.



Contents:

Uniting Bodies
Intersecting Bodies
Subtracting Bodies
Imprinting Bodies
Slicing a Body

Uniting Bodies

The Unite action merges two or more contacting bodies into a single body.

The *Unite Bodies* panel options are listed below.

Target Body	Specifies the body to which material is added. The common volume of the selected tool bodies is added to the target body. To change the selection, click in this group box and select the desired body from the *3D-CAD View* scene or from the **Bodies** node in the 3D-CAD feature tree.
Tool Bodies	Specifies the bodies that add material to the target body. To select target bodies, click in this group box and select the desired bodies from the *3D-CAD View* scene or from the **Bodies** node in the 3D-CAD feature tree.

Merge Type	Defines the type of merge parameter to use in the operation.	
	Precise	Uses the default internal model precision of `1.0E-8 m` as the tolerance. This tolerance setting requires that faces of separate bodies be within 1.0E-8 m of each other. This option takes a minimum amount of computational time and works well for all coincident faces and surfaces.
	Tolerant Edges	Reduces the model precision by using a higher tolerance value that you provide on all the edges in the model. This option causes any further geometry operations to use the lower precision model.
	Tolerant Faces	Finds the coincident faces within the specified tolerance. This option is well-suited for skewed faces without sacrificing the precision on the rest of the model.
	Tolerant Faces and Edges	Provides a combination of the **Tolerant Edges** and the **Tolerant Faces** options. Use this option only if the other options fail.
Tolerance	Specifies the tolerance for the operation when you set **Merge Type** to a tolerance. You are advised to increase the tolerance incrementally from `1.0E-8 m` (which is the precise tolerance) to `1.0E-7`, `1.0E-6`, `1.0E-5` until successful. Only increase the tolerance further if the neighboring feature dimensions are much larger. A very large tolerance value may cause small features to collapse internally and make the model invalid.	
Transfer Names	Transfers names from the tool bodies to the target body. You can transfer face names and body names. Activate the **Face Names** option to transfer custom face names from the tool bodies to the target body. If both the target body and the tool body contain a custom name, the name on the target body has precedence. Activate the **Body Names** option to transfer the name of the tool body to faces on the same tool body that do not have a custom name.	

Intersecting Bodies

Bodies that are intersecting share a common volume.

An Intersect operation combines these bodies and leaves only this common volume.

If the bodies do not all share a common volume, the operation deletes all selected bodies. If bodies do not intersect, use the transform operation to move the bodies until they share a common volume. Insert the transformation feature before the **IntersectBodies** feature.

You can specify the precision for this operation in the panel.

Bodies to Intersect	Specifies the bodies to intersect. To change the selections, click in this group box and select the desired bodies from the *3D-CAD View* scene or from the **Bodies** node in the 3D-CAD feature tree.	
Precision Type	Specifies the precision for the intersect operation.	
	Precise	Uses the default internal model precision of `1.0E-8 m` as the tolerance. This tolerance setting requires that faces of separate bodies be within 1.0E-8 m of each other. This option takes a minimum amount of computational time and works well for all coincident faces and surfaces.

	Tolerant Edges	Reduces the model precision by using a higher tolerance value that you provide on all the edges in the model. This option causes any further geometry operations to use the lower precision model.
Tolerance	Specifies the tolerance for the operation when you set **Precision Type** to **Tolerant Edges**. You are advised to increase the tolerance incrementally from `1.0E-8 m` (which is the precise tolerance) to `1.0E-7`, `1.0E-6`, `1.0E-5` until successful. Only increase the tolerance further if the neighboring feature dimensions are much larger. A very large tolerance value may cause small features to collapse internally and make the model invalid.	

See Also:

Transforming a Body
Understanding Where New Features Are Inserted

Subtracting Bodies

In a subtract operation, the volume of one or more bodies is removed from another body.

The bodies from which material is removed are known as the targets, and the bodies that are used to remove the material are known as tools. You can use any number of tool bodies to subtract material from any number of target bodies. You can choose whether to delete or whether to keep the tool bodies in this operation.

Each tool body must share some of its volume with the target. If a tool body does not overlap the target, it is deleted from the model during the operation and the target body is unchanged—a **SubtractBodies** feature node is added to the feature tree. If the bodies do not overlap, use the transform operation to move them so that they do overlap.

You can specify the designation of target and tool bodies in the *Subtract Bodies* panel.

Target Bodies	Specifies the bodies from which material is removed. The common volume of the selected tool bodies is removed from the target body. To change the selections, click in this group box and select the desired bodies from the *3D-CAD View* scene or from the **Bodies** node in the 3D-CAD feature tree.	
Tool Bodies	Specifies the bodies that remove material from the target body. To change the selections, click in this group box and select the desired bodies from the *3D-CAD View* scene or from the **Bodies** node in the 3D-CAD feature tree.	
Keep Tool Bodies	When activated, retains the tool bodies after the operation. By default, this option is deactivated and the tool bodies are consumed in the operation.	
Imprint	Imprints the tool bodies onto the target bodies after performing the operation. This option is only available when you activate the **Keep Tool Bodies** option.	
Precision Type	Specifies the precision for the subtract operation.	
	Precise	Uses the default internal model precision of `1.0E-8 m` as the tolerance. This tolerance setting requires that faces of separate bodies be within 1.0E-8 m of each other. This option takes a

		minimum amount of computational time and works well for all coincident faces and surfaces.
	Tolerant Edges	Reduces the model precision by using a higher tolerance value that you provide on all the edges in the model. This option causes any further geometry operations to use the lower precision model.
Tolerance	Specifies the tolerance for the operation when you set **Precision Type** to **Tolerant Edges**. You are advised to increase the tolerance incrementally from `1.0E-8 m` (which is the precise tolerance) to `1.0E-7`, `1.0E-6`, `1.0E-5` until successful. Only increase the tolerance further if the neighboring feature dimensions are much larger. A very large tolerance value may cause small features to collapse internally and make the model invalid.	
Transfer Names	Transfers names from the tool bodies to the target body. You can transfer face names and body names. Activate the **Face Names** option to transfer custom face names from the tool bodies to the target body If both the target body and the tool body contain a custom name, the name on the target body has precedence. Activate the **Body Names** option to transfer the name of the tool body to faces on the same tool body that do not have a custom name.	

See Also:

Transforming a Body

Imprinting Bodies

The imprint operation is carried out between two or more bodies that are overlapping or have coincident faces.

This operation pastes the lines and curves from each body onto the faces where the two bodies meet or intersect. When you perform an imprint operation, the affected faces are not deleted or merged, but the face is split where the new lines and curves are added to each face.

This operation allows you to manipulate a face for further sketching operations, or to automate the creation of contact data before importing the model into a simulation. If the faces are coincident, imprinting also generates part contact data. STAR-CCM+ uses part contact data to create interfaces between regions.

Requirements

There are two types of imprinting in 3D-CAD: precise imprinting, and tolerant imprinting.

For precise imprinting, the following criteria must be satisfied:

- The bodies must overlap by less than `1.0E-8 m`, or have coincident faces.
- A minimum of two bodies are required for imprinting.

If the selected bodies do not meet the criteria that are mentioned above, the surface of each body remains unchanged after the operation, but an **Imprint** feature node is still added to the feature tree. If bodies are not in contact, use the transform operation to move the bodies so that they overlap or make contact between faces.

Imprint Bodies Panel

The *Imprint Bodies* panel options are listed below.

Imprint Option	Specifies whether to imprint all the selected bodies together or whether to imprint two groups of bodies together.	
	Within group of bodies	Imprints all selected bodies together.
	Between two groups of bodies	Imprints the bodies in the first group with the bodies in the second group.
First Group	Specifies the bodies for the first group. To change the selections, click in this group box and select the desired bodies from the *3D-CAD View* scene or from the **Bodies** node in the 3D-CAD feature tree.	
Second Group	Specifies the bodies for the second group. To change the selections, click in this group box and select the desired bodies from the *3D-CAD View* scene or from the **Bodies** node in the 3D-CAD feature tree.	
Imprint Type	Defines the tolerance method to use in the operation.	
	Precise	Uses the default internal model precision of `1.0E-8 m` as the tolerance. This tolerance setting requires that faces of separate bodies be within 1.0E-8 m of each other. This option takes a minimum amount of computational time and works well for all coincident faces and surfaces.
	Tolerant Edges	Reduces the model precision by using a higher tolerance value that you provide on all the edges in the model. This option causes any further geometry operations to use the lower precision model.
	Tolerant Faces	Finds the coincident faces within the specified tolerance. This option is well-suited for skewed faces without sacrificing the precision on the rest of the model.
	Tolerant Faces and Edges	Provides a combination of the **Tolerant Edges** and the **Tolerant Faces** options. Use this option only if the other options fail.
Tolerance	Specifies the tolerance for the operation when you set **Imprint Type** to a tolerance. You are advised to increase the tolerance incrementally from `1.0E-8 m` (which is the precise tolerance) to `1.0E-7`, `1.0E-6`, `1.0E-5` until successful. Only increase the tolerance further if the neighboring feature dimensions are much larger. A very large tolerance value may cause small features to collapse internally and make the model invalid.	

Contents:

Imprinting Groups of Bodies

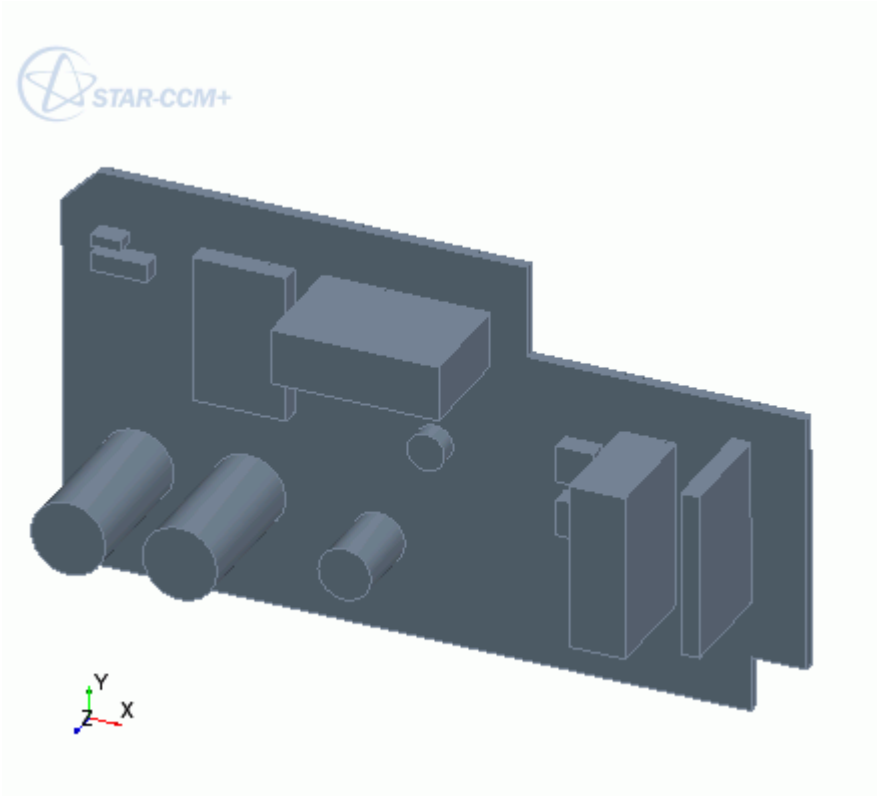
See Also:

Transforming a Body

Using Part Contact Data to Create Interfaces

Imprinting Groups of Bodies

In 3D-CAD, you can define and imprint two groups of bodies together rather than imprinting all bodies together. Imprinting all bodies together is computationally expensive, especially if you have hundreds of bodies, and in some cases is unnecessary. For example, if you want to imprint multiple bodies onto a single body, such as in electronics cooling, it is more efficient to perform only the imprint that is required.



An example workflow is provided below:

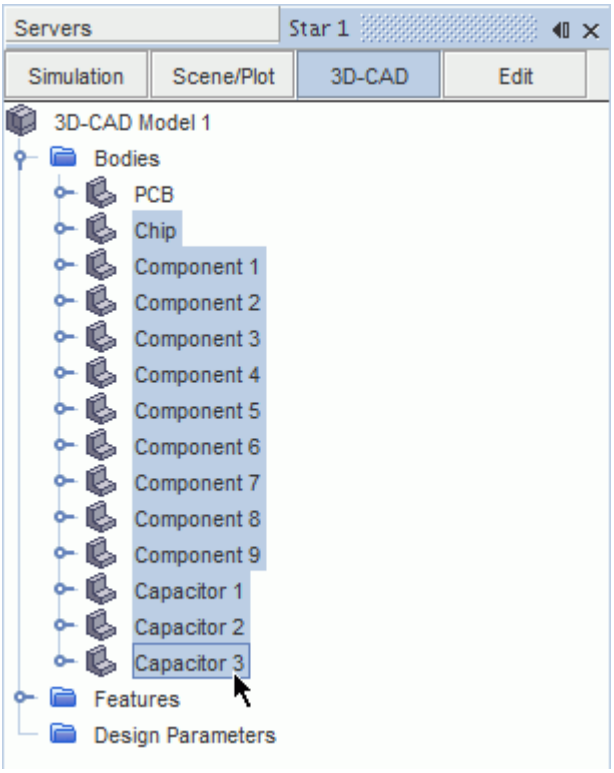
1. In the *Imprint Bodies* panel, set *Imprint Option* to **Between two groups of bodies**.

   Two group boxes appear.
2. To define the first group of bodies, click inside the group box labelled *First Group* and select the select the printed circuit board from the *3D-CAD View* scene.

   The selection is highlighted in green in the *3D-CAD View* scene.
3. To define the second group of bodies, click inside the group box labelled *Second Group* and select the select the desired components.

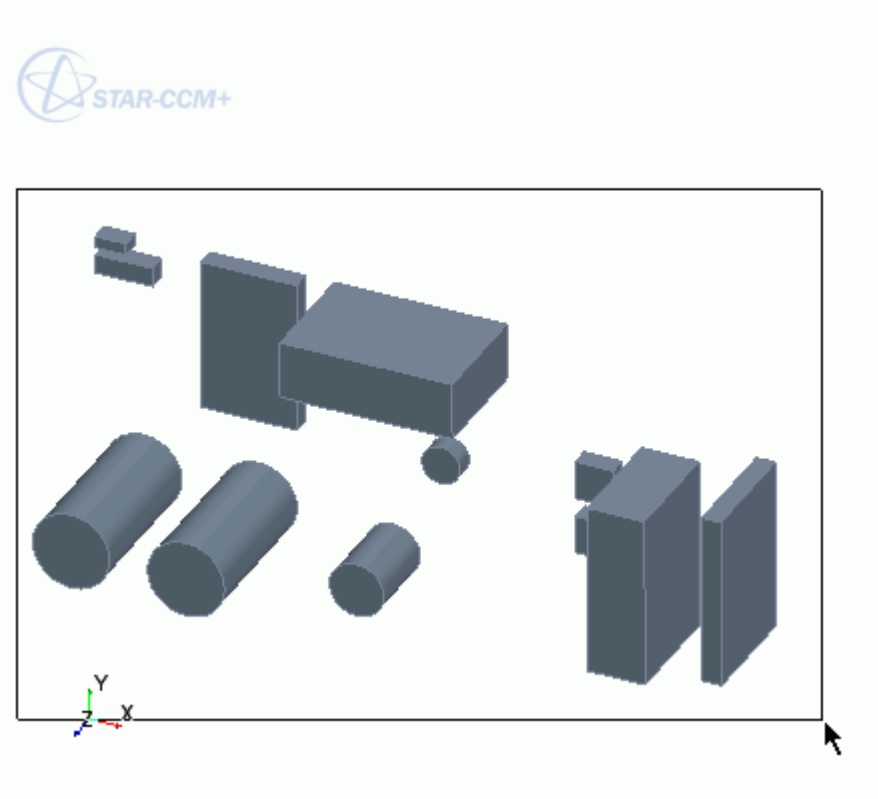
   There are two recommended methods to select a group of bodies:

   - Using the 3D-CAD feature tree:
     1. Click the **3D-CAD** tab to switch to the 3D-CAD feature tree.
     2. Expand the **Bodies** node and multi-select the bodies that you want to include in the group.



3. Click the **Edit** tab to return to the *Imprint Bodies* panel.

- Using the Rubberband Select tool:
  1. In the 3D-CAD View scene, hide the printed circuit board. See Hiding Bodies.
  2. Set the Selection Filter to only select Volumes. See Filtering the Selections.
  3. Use the Rubberband Select tool and drag a box around the bodies that you want to include in the group. See Selecting in a Rubberband.



The selections are highlighted in red in the *3D-CAD View* scene.

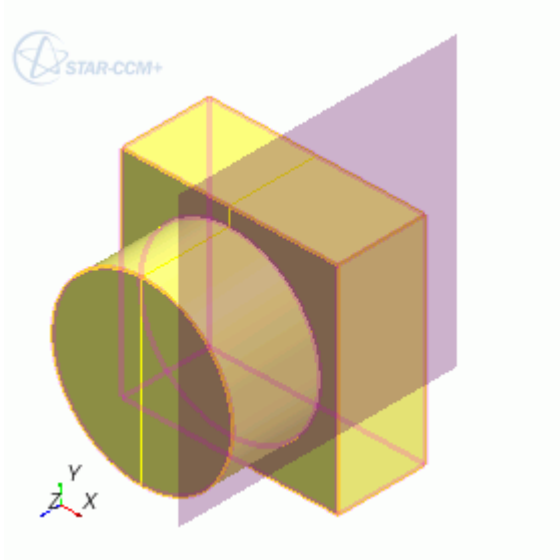
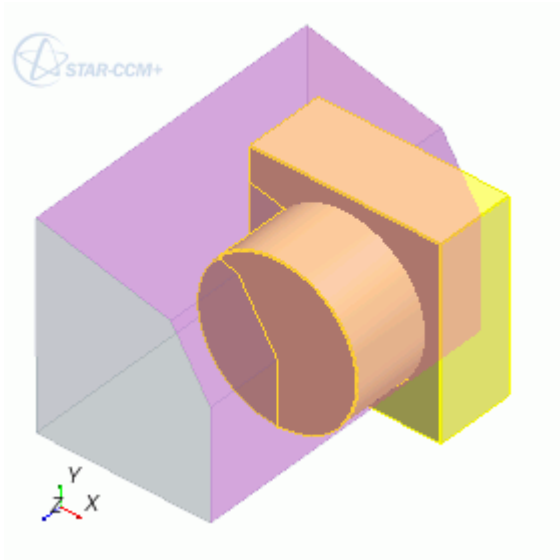
4. Set the imprint options. See Imprint Bodies Panel.
5. Click **OK** to perform the imprint.

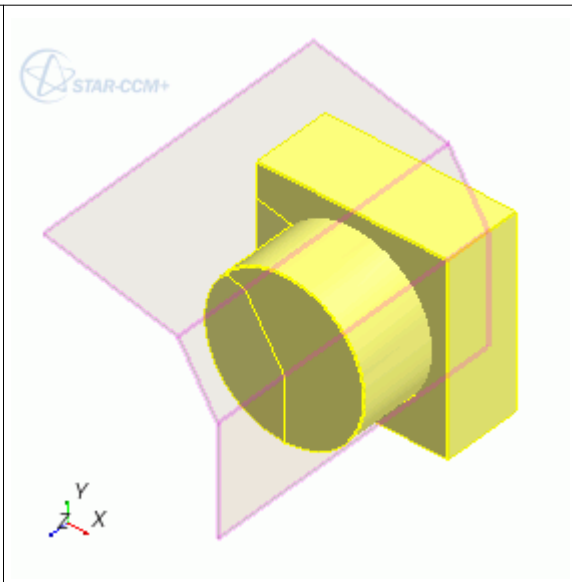
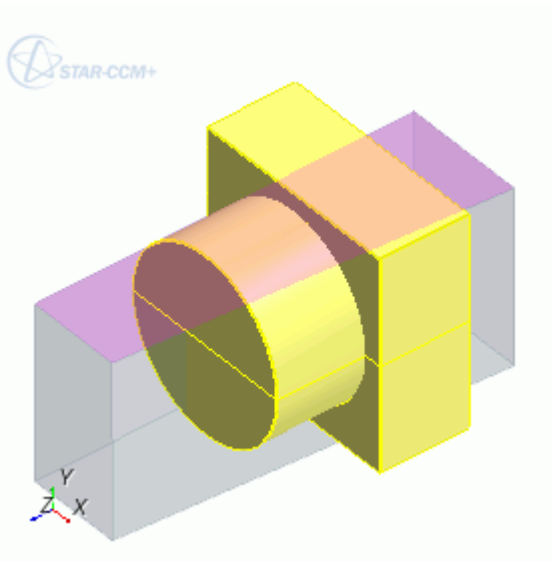
Slicing a Body

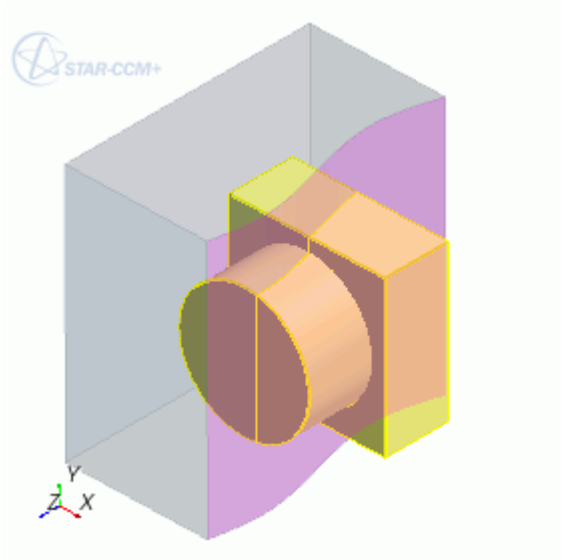
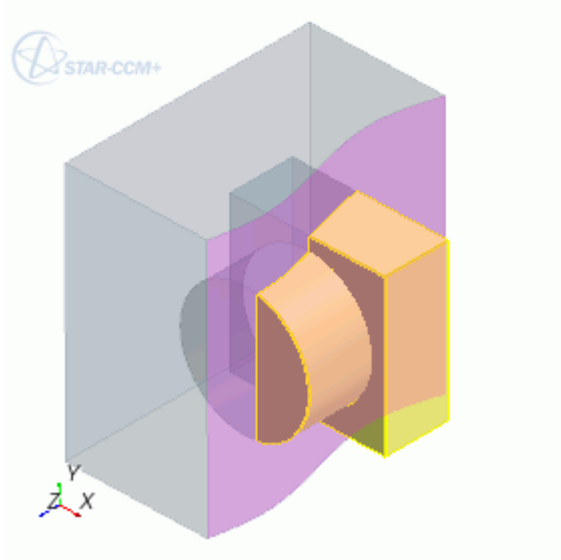
Use the slicing operation to split an existing body into two or more bodies.

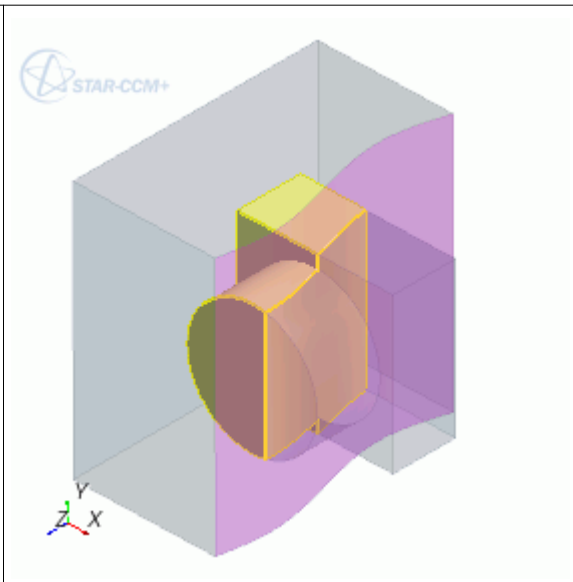
You can slice a body using a sketch plane, a face of another body, or a sheet body. You can slice a body using a single slicing tool, or multiple slicing tools. When using a single tool for the slice operation, you can choose whether to retain both of the resulting bodies, or only one on a particular side of the slice. When using multiple slicing tools, all resulting bodies are retained. The *Slice Bodies* panel options are listed below.

Bodies to Slice	Specifies the bodies to slice.	
Slice by	Defines the method to use for slicing the selected body.	
	Sketch Plane	Slices the body using an existing sketch plane.

		
	Face	Slices using the selected faces. Select the faces to slice the body using the (**Select From Scene**) tool. 
	Sheet Body	Slices the body using the select sheet bodies. Select the sheet bodies from the scene using the (**Select From Scene**) tool.

		
Tool Faces	Specifies the faces to use as the slice tool. This group box appears when you set **Slice by** to **Face**. To select the faces, click inside this group box and select the desired faces from the *3D-CAD View* scene.	
Tool Sheet Bodies	Specifies the sheet bodies to use as the slice tool. This group box appears when you set **Slice by** to **Sheet Body**. To select the sheet bodies, click inside this group box and select the desired sheet bodies from the *3D-CAD View* scene or from the **Bodies** node in the 3D-CAD feature tree.	
Extend Face	Extends the selected planar or cylindrical face to slice through the body. This is useful for when the slicing face either does not intersect the target body, or only partially intersects the target body. 	
Select Side	Specifies which side of the slice is kept after performing the slice operation. This option is greyed out if you select multiple faces or sheet bodies.	

	Both Sides	Retains both sides of the slice. The body is split into two separate bodies. 
	Front Side	Retains the front side of the slice. 
	Back Side	Retains the back side of the slice.

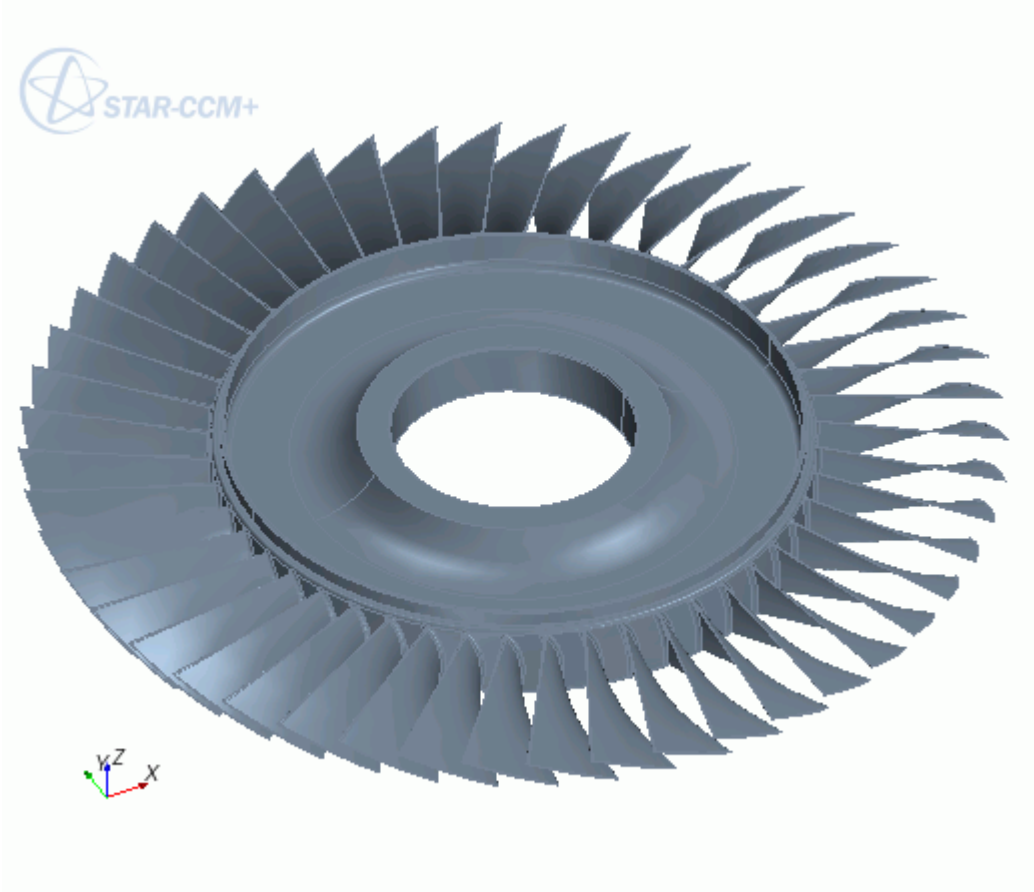
		

Contents:

Slicing a Body Using Multiple Faces or Sheet Bodies

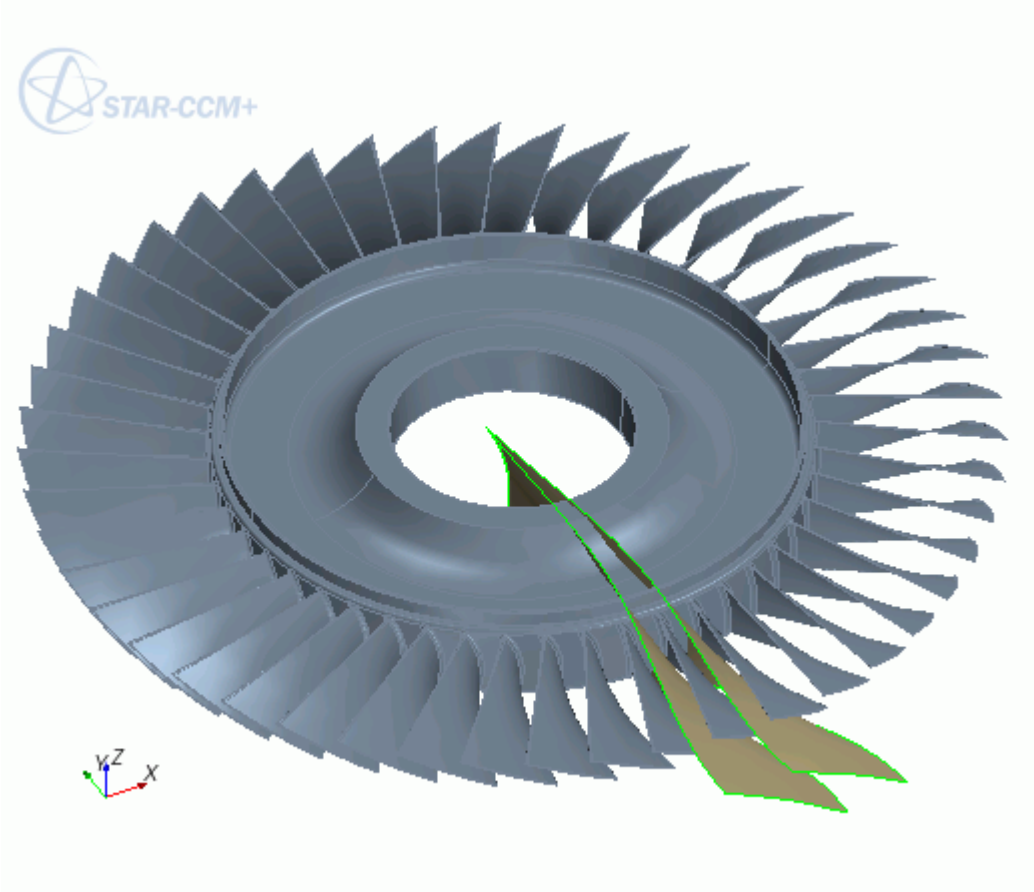
Slicing a Body Using Multiple Faces or Sheet Bodies

You can slice a body with multiple faces or sheet bodies. This functionality is useful for creating a section from a periodic geometry, such as is commonly found in turbomachinery.



An example workflow is provided below:

1. Create the sheet bodies to act as the slicing tools.



2. In the 3D-CAD feature tree, right-click the body you want to slice and select **Boolean** > **Slice**.

   The *Slice Bodies* panel appears.

3. Set *Slice by* to **Sheet Body**.

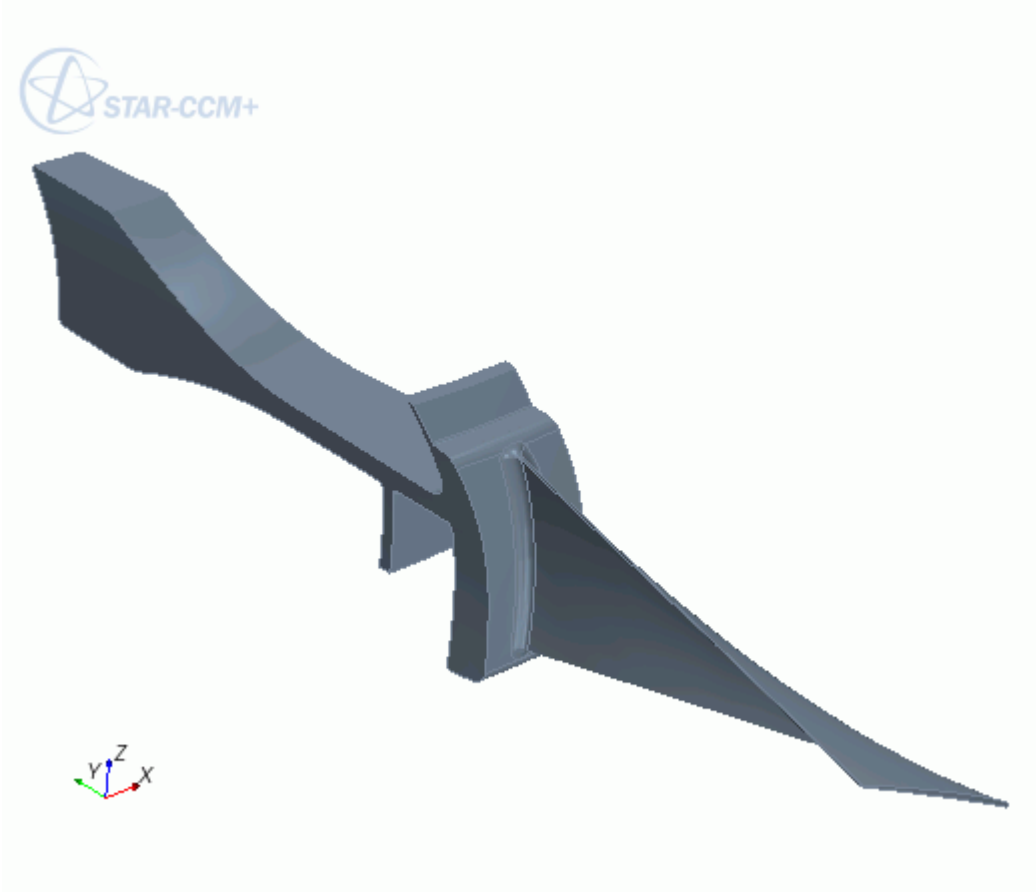
   The *Tool Sheet Bodies* group box appears. This group box allows you to use sheet bodies as the slicing tool.

   If you want to select specific faces from a body, set this option to **Face**. See Slicing a Body.

4. Click inside *Tool Sheet Bodies* group and select the slicing tools (the sheet body faces) from the *3D-CAD View* scene or from the **Bodies** node in the 3D-CAD feature tree. You can select any number of slicing tools in the same operation.

5. Perform the slice.

The section is created. Hide the other bodies in the 3D-CAD tree to view the section.



Imprinting Sketches onto Bodies

You can split a surface of a solid or sheet body by imprinting a 2D or 3D sketch onto it.

You can then use the split surface in order to create a boundary condition within regions, for example.

Requirements

To successfully imprint a sketch onto a surface, the following criteria must be satisfied:

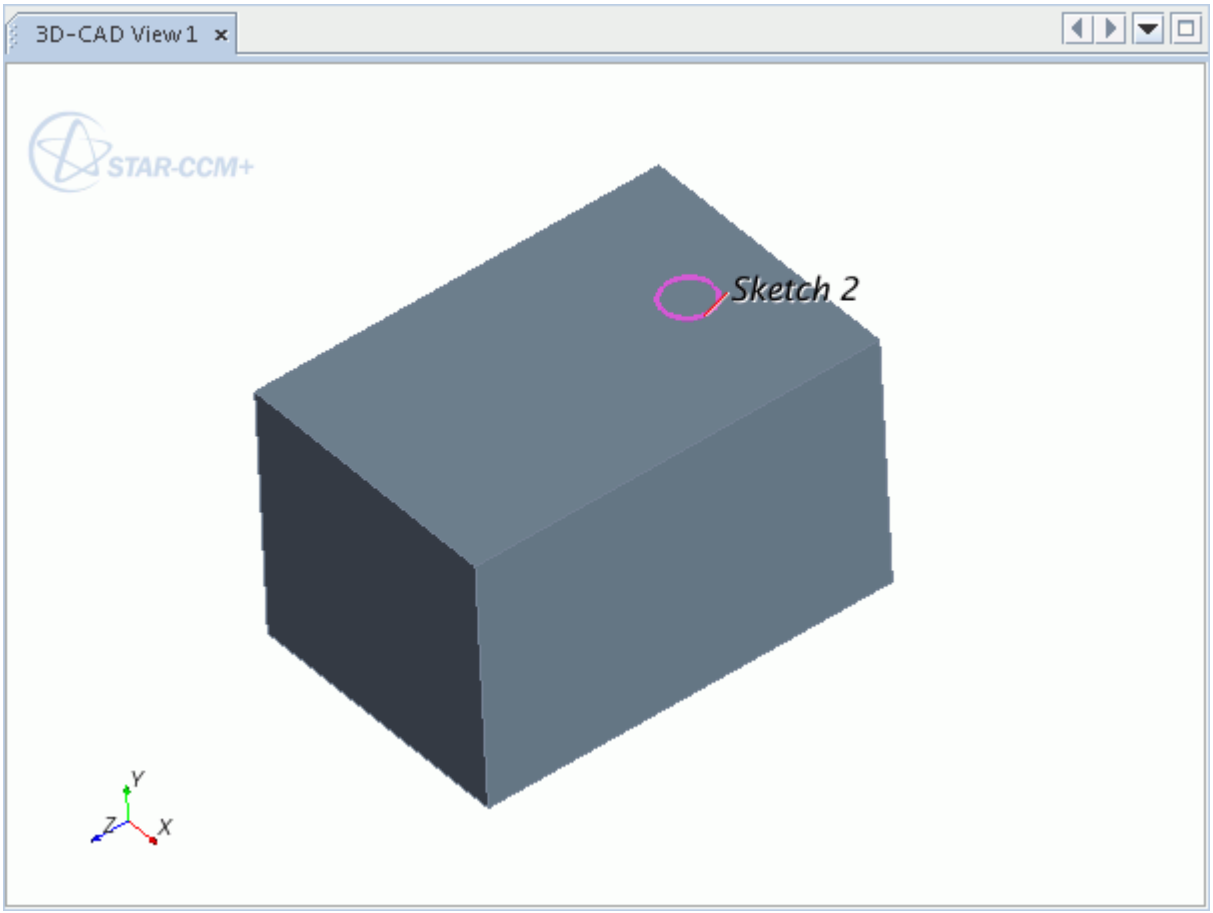
- The sketch can lie on the same plane as the target surface, or it can be projected onto the target surface from a distance.
- The sketch can be 2D or 3D.
- The target surface can be planar or curved.
- The sketch can contain closed loops or open profiles. Multiple profiles in one sketch are permitted.
- An open profile can terminate within the bounds of the target surface or it can cross the target surface entirely.
- If the sketch lies partially on the target surface, this part of the sketch is used to split the surface.

Contents:

Creating The Imprint
Sketch Imprint Properties

Creating The Imprint

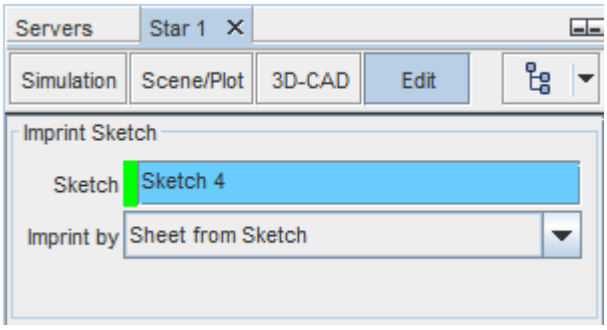
In the following example, the sketch of a circle is imprinted onto the top surface of a solid block. The sketch lies on the plane of the top surface.



To create a sketch imprint:

1. Right-click on the desired sketch in the 3D-CAD object tree and select **Sketch Imprint**.

The *Imprint Sketch* panel appears and the selected sketch is highlighted in the *3D-CAD View* scene, as shown above.

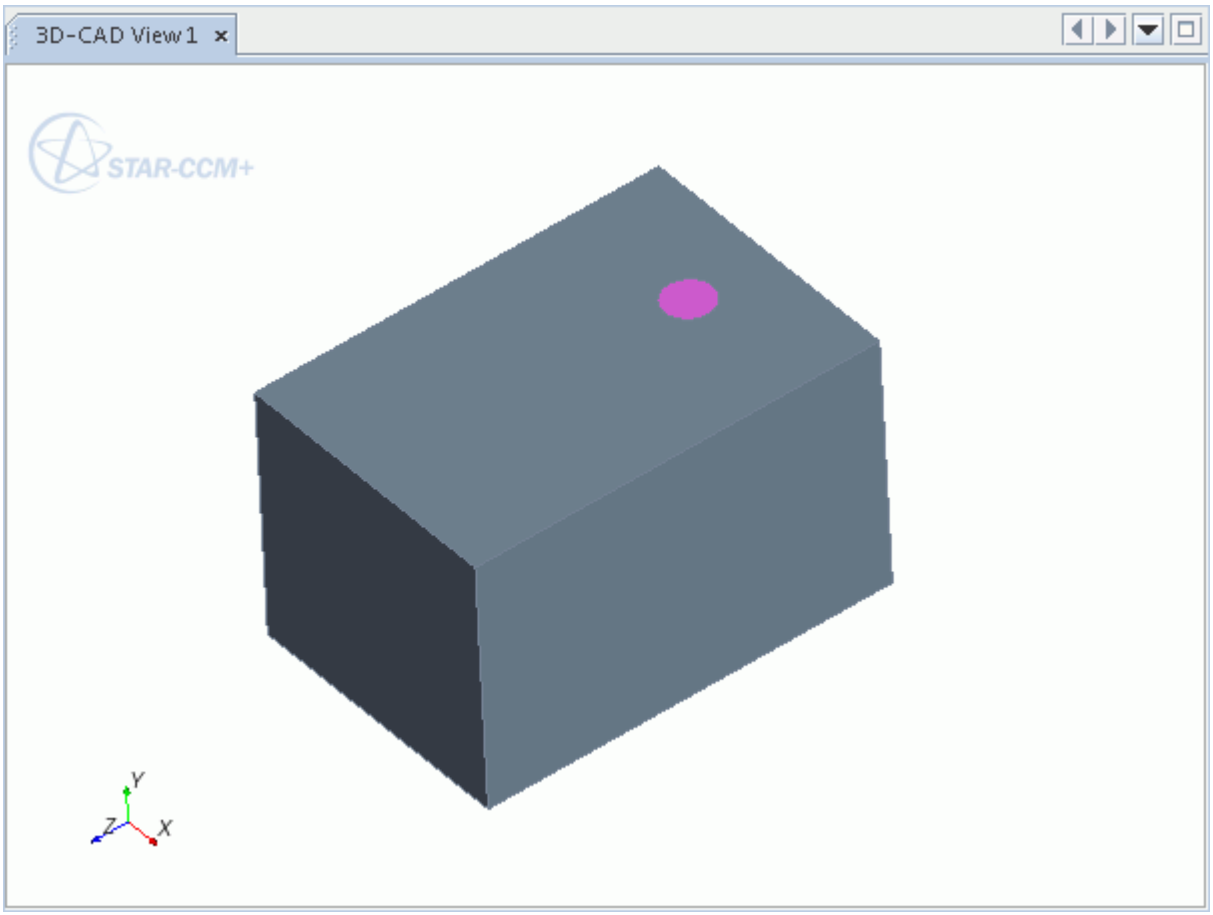


2. Choose the imprint options. For this case, where the sketch lies on the same plane as the body surface, the default options are appropriate.

   For cases where the sketch lies on a different plane, or in the case of a 3D sketch, adjust the imprint options accordingly. See Sketch Imprint Properties.

3. Click **OK** to imprint the sketch.

The imprinted sketch splits the original face of the block. The new face that is created by the imprint feature is shown highlighted below.



Sketch Imprint Properties

The sketch imprint properties control the behaviour of the imprint for a sketch. You can imprint a sketch that lies on the same plane as the body/face that you want to imprint, or you can project the sketch onto a body/face that lies some distance away from it.

Sketch	Specifies the sketch that is imprinted onto the body surface.	
Imprint by	Specifies the method that is used to imprint onto body surfaces.	
	Sheet from Sketch	Imprints the sketch onto the surface. The sketch and the surface must lie on the same plane.
	Project Curves on Bodies	Projects the sketch onto the selected bodies/faces. The direction of the projection is controlled using the **Direction Type** and the subsequent options that appear for that particular direction type.
	Project Curves on Faces	
Direction Type	Specifies the direction of the projection.	
	Proximal	Specifies the direction to match the face normal. The length of the projection is specified using a maximum distance, **Max Projection Distance**.
	Normal	Specifies the direction to be normal to the sketch plane. The sketch must be 2D.
	Reverse	Specifies the direction to be the reverse of the normal direction. The sketch must be 2D.
	Specified	Allows you to specify a custom direction using the **Coordinate System Source** options.
	Reference Axis	Specifies the direction using a reference axis that you created. See Working With Reference Geometry in 3D-CAD: Reference Axes.
Coordinate System Source	Defines the custom direction for the projection using coordinate systems. This option appears when you set **Direction Type** to **Specified**.	
	Sketch Plane	Uses the coordinate system from the sketch plane.
	Imported	Uses an imported coordinate system. Select a coordinate system from the drop-down list that appears below this option.
	Reference Coordinate System	Uses a reference coordinate system which you create. Click in the selection field that appears below this option and select the reference coordinate system in the *3D-CAD View* scene. See Working With Reference Geometry in 3D-CAD: Reference Coordinate Systems.
Direction Vector	Defines the direction of the projection using **X**, **Y**, and **Z** direction vectors.	

Projection Options	Specifies whether the sketch is projected forward or backwards from its initial position. **One Way** projects the sketch forward using the parameters that you set for the direction of the projection. **Two Way** projects the sketch forwards and backwards using the parameters that you specify.
Bodies/ Faces to Imprint	Specifies the bodies/faces that are imprinted by the sketch. You can imprint onto sheet bodies and solid bodies.
Max Projection Distance	Specifies the maximum distance for the sketch to project. This option is only valid when you set **Direction Type** to **Proximal**. If a body or face falls within this distance, the sketch is imprinted on this body/face. Bodies/faces that lie outside this distance are not imprinted.

Working with Faces, Edges, and Vertices

In 3D-CAD, vertices define edges, edges, define faces, and faces define bodies. These entities define the surfaces and curves in a geometry part, and the boundaries within a region.

You cannot access vertices, edges, or faces in the 3D-CAD object tree. You can only select faces, edges, and vertices in the *3D-CAD View* window. You can create new features from faces and edges, such as sketches and sketch planes, chamfers and fillets, as well as internal volume extractions. You can also label a face, edge, or vertex with a specific name. A named face becomes its own surface when you convert the model to a geometry part. You can use named vertices in FEA application as a constraint or as a point load.

This section covers the following topics:

Contents:

Faces, Edges, and Vertices in 3D-CAD
Hiding Faces
Deleting Faces
Extending Surfaces
Filling Holes
Bridge Surfaces
Setting Face, Edge, and Vertex Names
Visualizing Named Faces, Edges, and Vertices
Splitting a Face by Offsetting an Edge

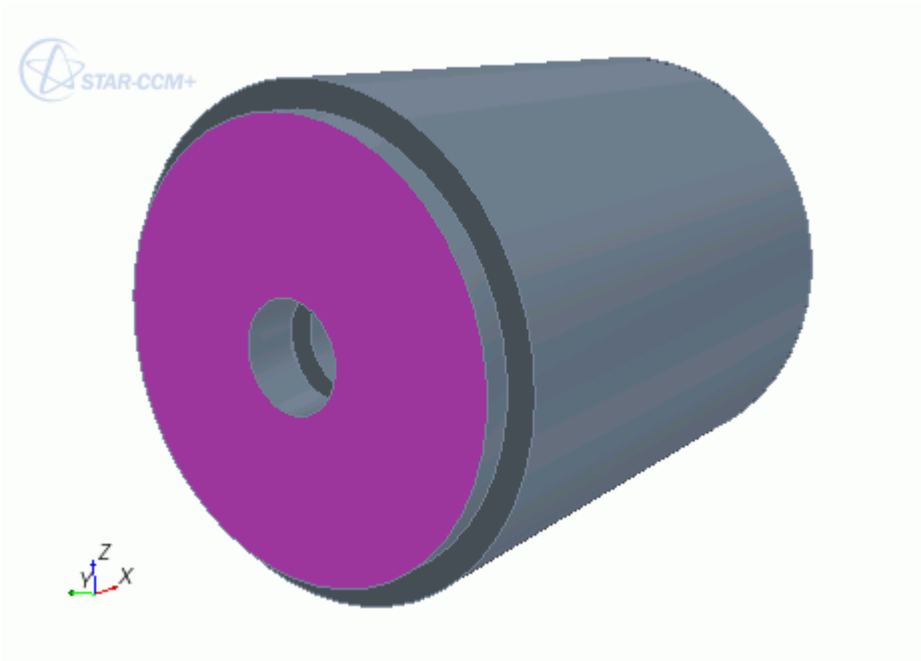
See Also:

Overview of Bodies in 3D CAD

Faces, Edges, and Vertices in 3D-CAD

Faces

A face is a bounding surface of a body. Faces are created and updated automatically when you generate or modify a body. You can operate on faces through the *3D-CAD View* window.

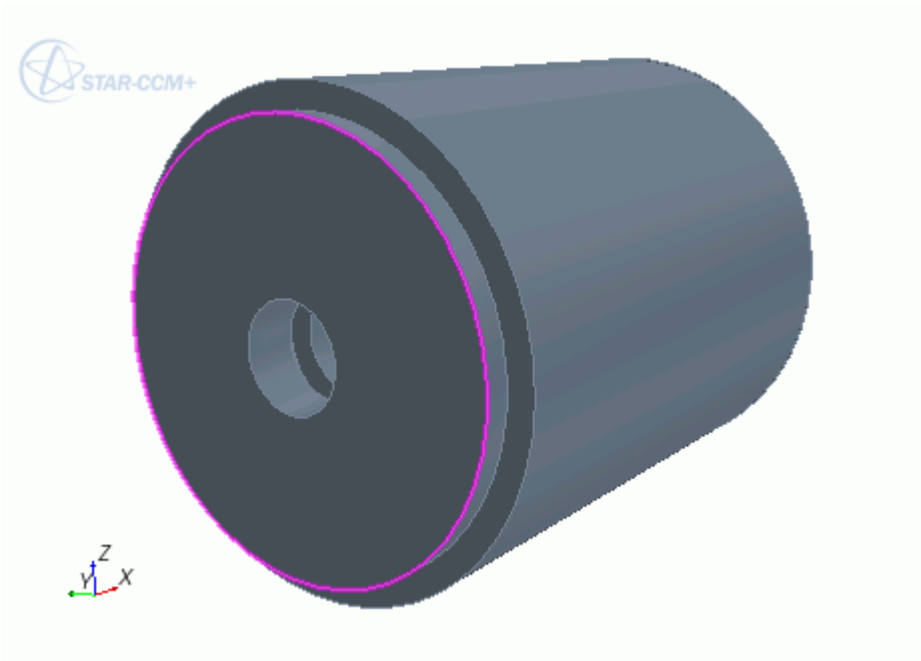


Note: Double-clicking a face selects other faces that are connected and tangent to it.

You can give names to faces in 3D-CAD (these then appear in the **Named Faces** folder within the body node). These surfaces then become boundaries when you assign the part to a region. When you convert a 3D-CAD model to geometry parts, the faces become surfaces.

Edges

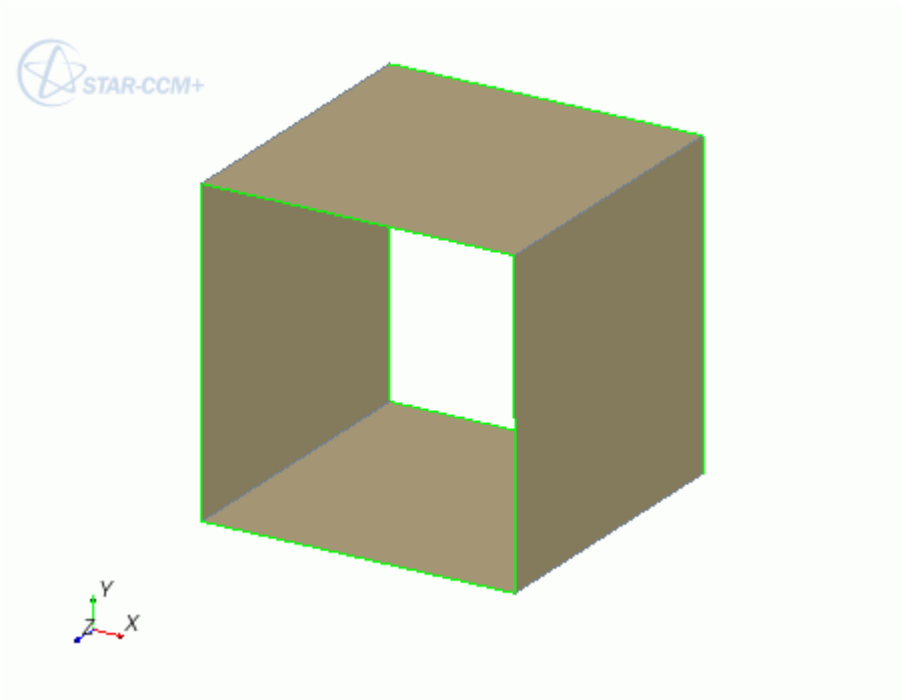
An edge is the line joining two adjacent faces of a body. An edge is created and updated automatically when you generate or modify a body. You can operate on edges through the *3D-CAD View* window. When you convert a 3D-CAD model to geometry parts, the edges become curves.



Note: Double-clicking an edge selects other edges that are connected and tangent to it.

Free Edges

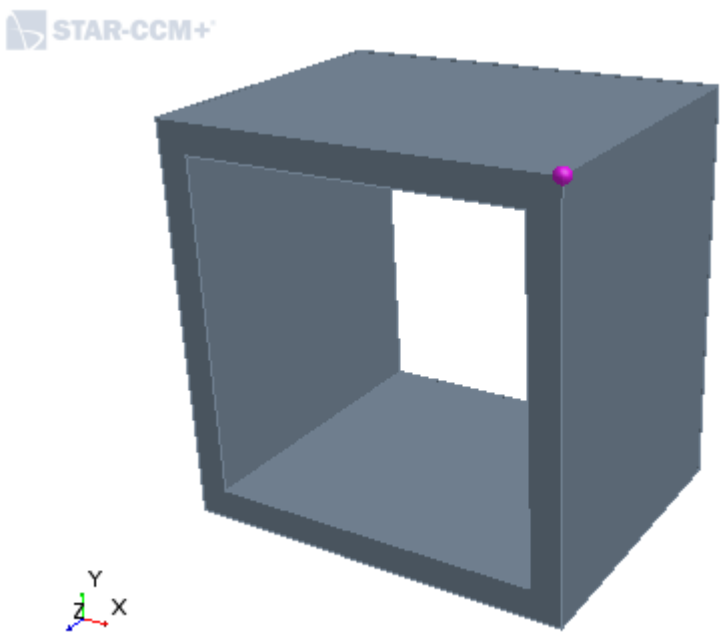
A free edge is an edge that lies on only one face. Free edges are indicated by a green color in the *3D-CAD View* scene. If a free edge appears between two sheet bodies, this indicates that the sheet bodies are not connected. They can be sewn together if their edges coincide. If free edges form a closed loop, you can created a filled surface. You cannot use free edges with a Fillet or Chamfer operation.



Note: Double-clicking a free edge selects all neighboring free edges that form a closed loop with that edge.

Vertices

A vertex is a point that lies at the end of an edge. Typically, edges have two vertices, one on either end of the edge. Closed edges, such as circles, have no vertices. You can interact with vertices on bodies in the *3D-CAD View* window. When you convert a 3D-CAD model to geometry parts, the vertices become available as points for use in FEA applications.



See Also:

Working with Bodies
Creating Geometry Parts from 3D-CAD
Working with Fillets and Chamfers
Setting Face, Edge, and Vertex Names

Hiding Faces

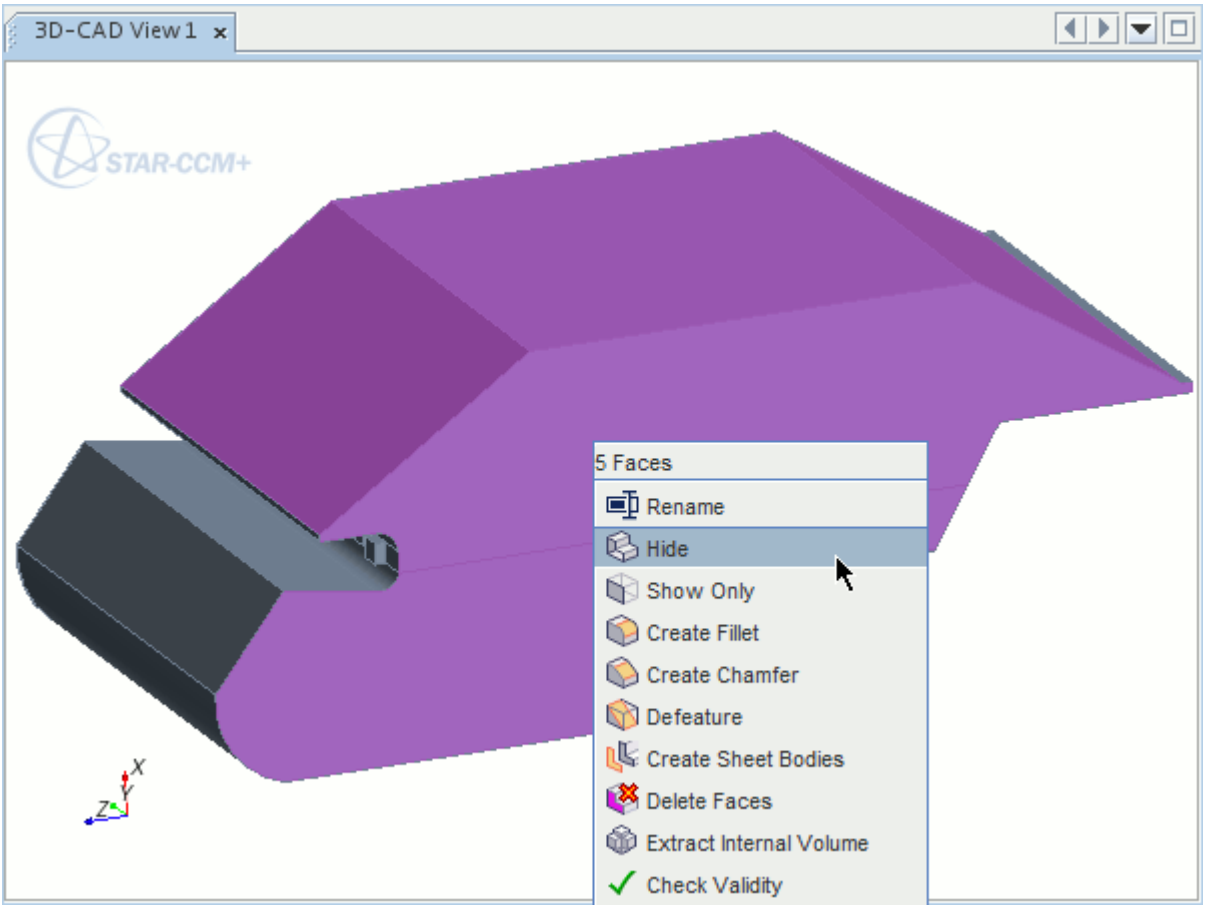
You can gain access to concealed parts by hiding obstructive parts.

You can use this technique to rename faces that were previously obscured, or to create features from them. You can also hide faces on bodies when you are performing body operations, allowing you to select bodies that are positioned within other bodies. If you want to hide an entire body from the scene, use the hide body action. See Hiding Bodies.

Hiding a Face

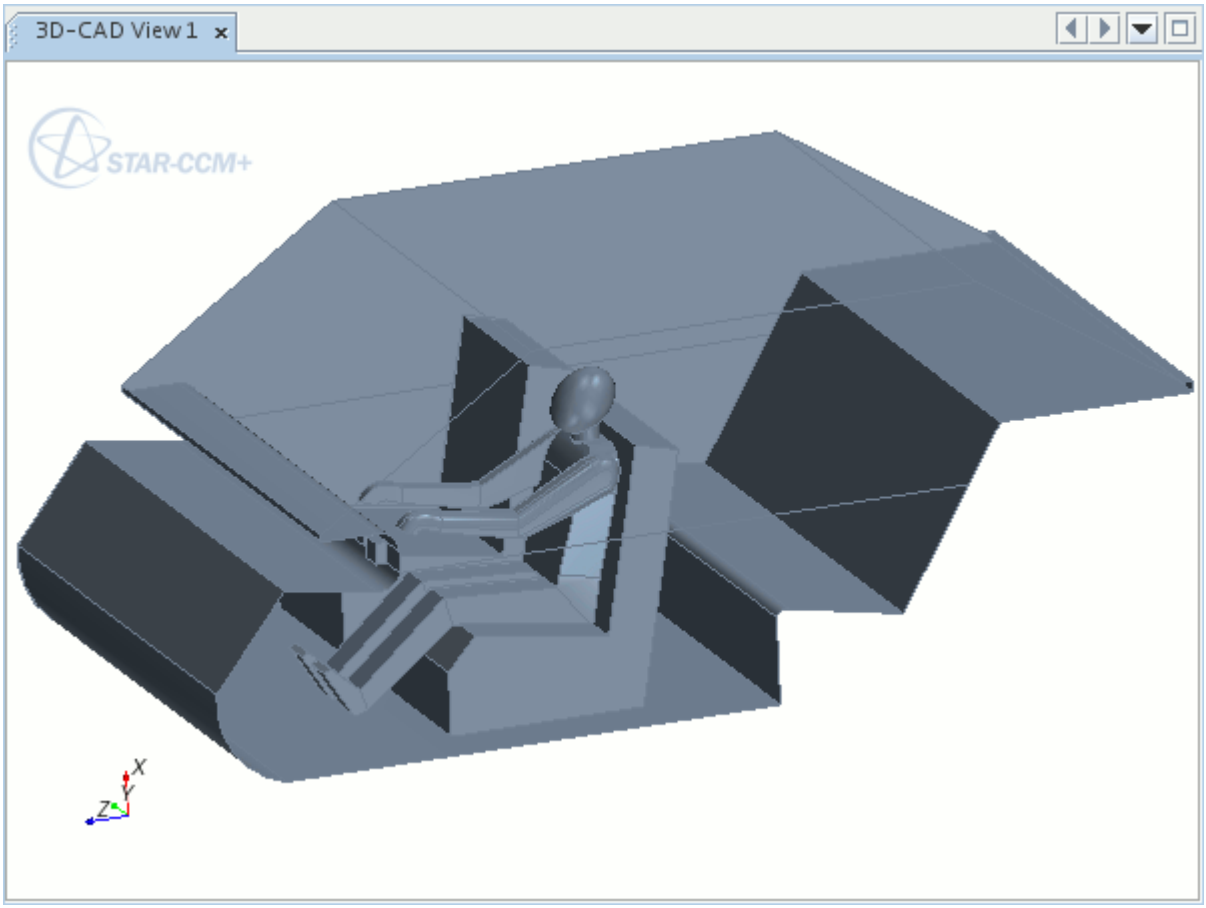
To hide a face:

- Right-click the face in the *3D-CAD View* scene and select **Hide**. To select multiple faces, hold down the `<Ctrl>` key.



- Alternatively, while performing a CAD operation in *Edit* mode, right-click the face in the *3D-CAD View* scene and select **Hide Face**. You can only hide one face at a time using this method.

The selected faces are hidden in the scene, exposing the geometry behind.



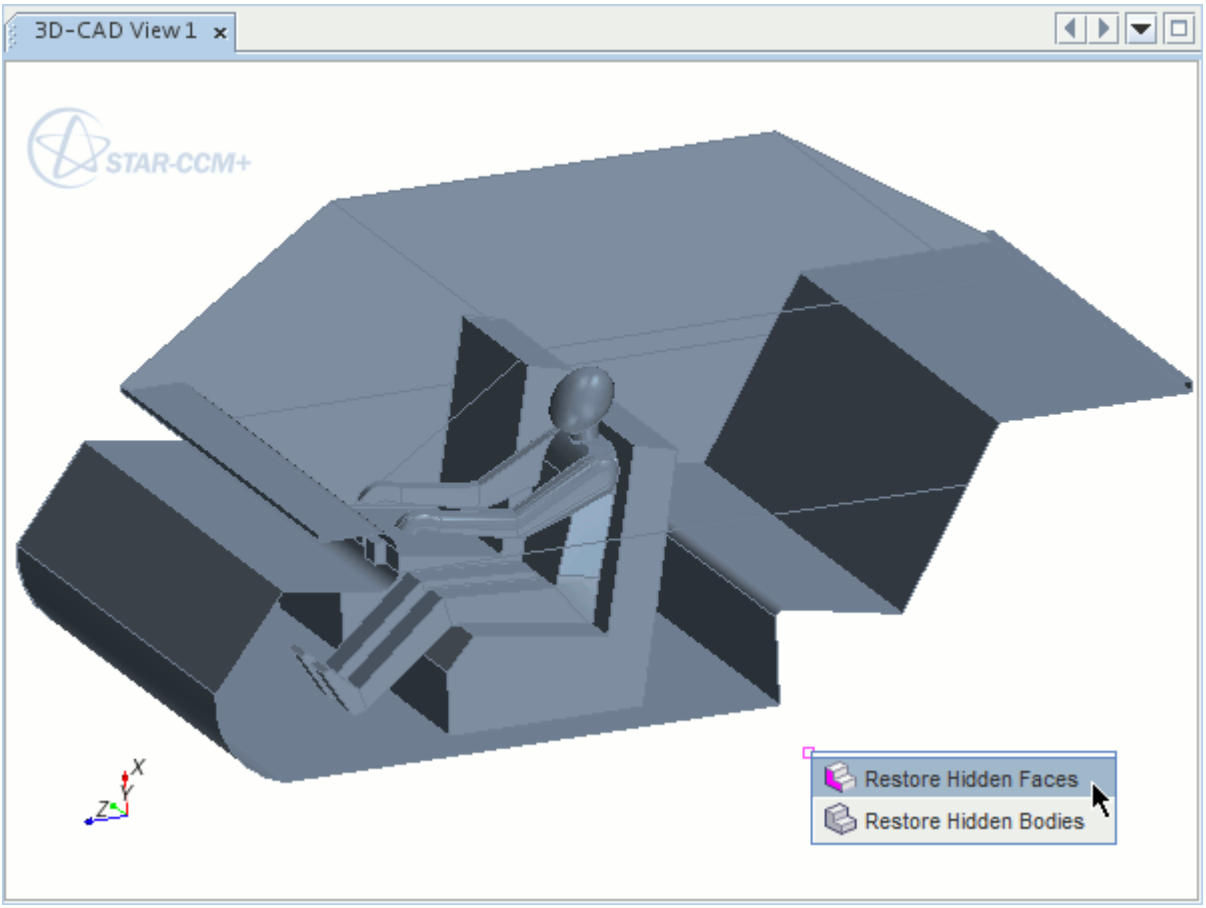
To hide all faces other than the selected ones:

- Right-click the body in the *3D-CAD View* scene and select **Body** > **Show Only**.

Restoring a Face

To restore hidden faces in the *3D-CAD View* scene:

- Right-click in the white space around the model and select **Restore Hidden Faces**. It is not possible to restore individual faces; this action restores all hidden faces in the model.

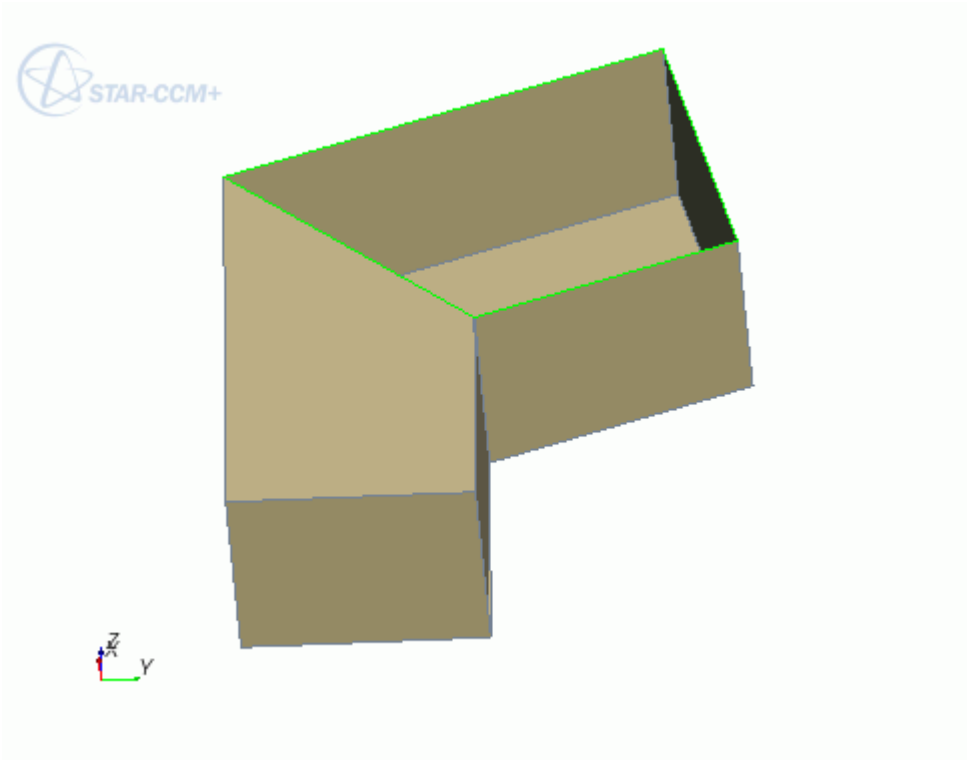


Deleting Faces

The Delete Faces action removes faces that you select from the 3D-CAD model.

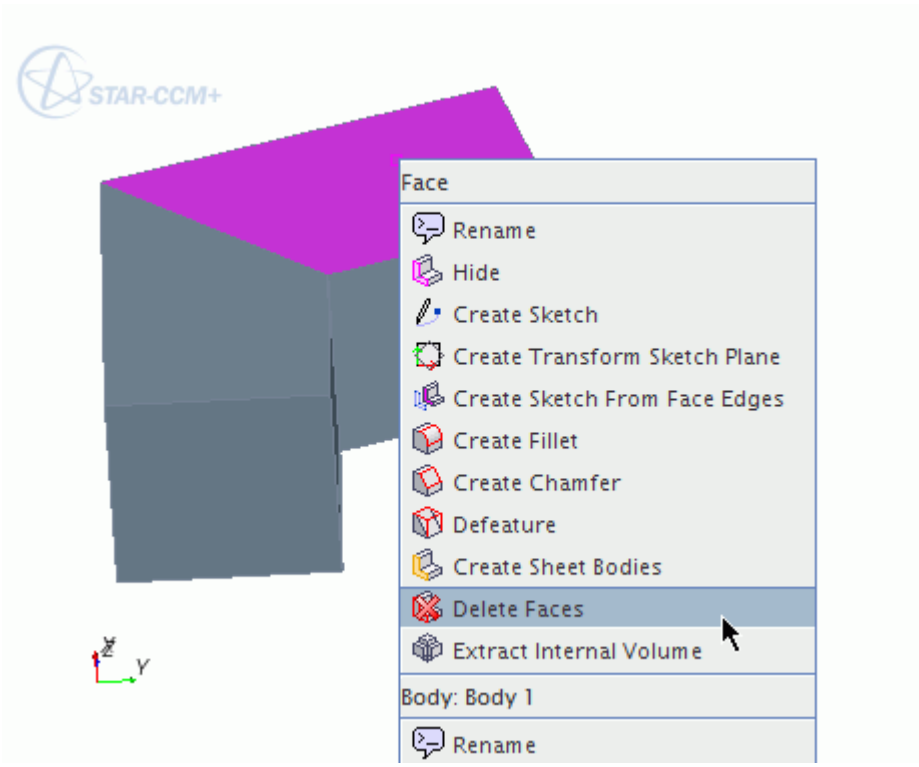
You can delete faces from solid bodies or sheet bodies. A **DeleteFaces** feature is also added to the 3D-CAD feature tree. If you want to temporarily hide a face, and not delete it, use the Hide operation instead. See Hiding Faces.

After deleting a face from a solid body, the solid body is converted to a sheet body. If the deletion action results in a disconnected topology, multiple bodies are created. If the deletion action results in a topology with face groups that are connected only by vertices, then the feature creates separate bodies for each such face group. If you select all the faces on a body, the entire body is deleted. The edges that bound the deleted surface become free edges. You can also delete faces from sheet bodies.



To delete a face from a body:

1. In the *3D-CAD View* scene, right-click on the face that you want to delete and select **Delete Faces**.



The *Delete Faces* panel appears. The selected faces are shown in the group box.

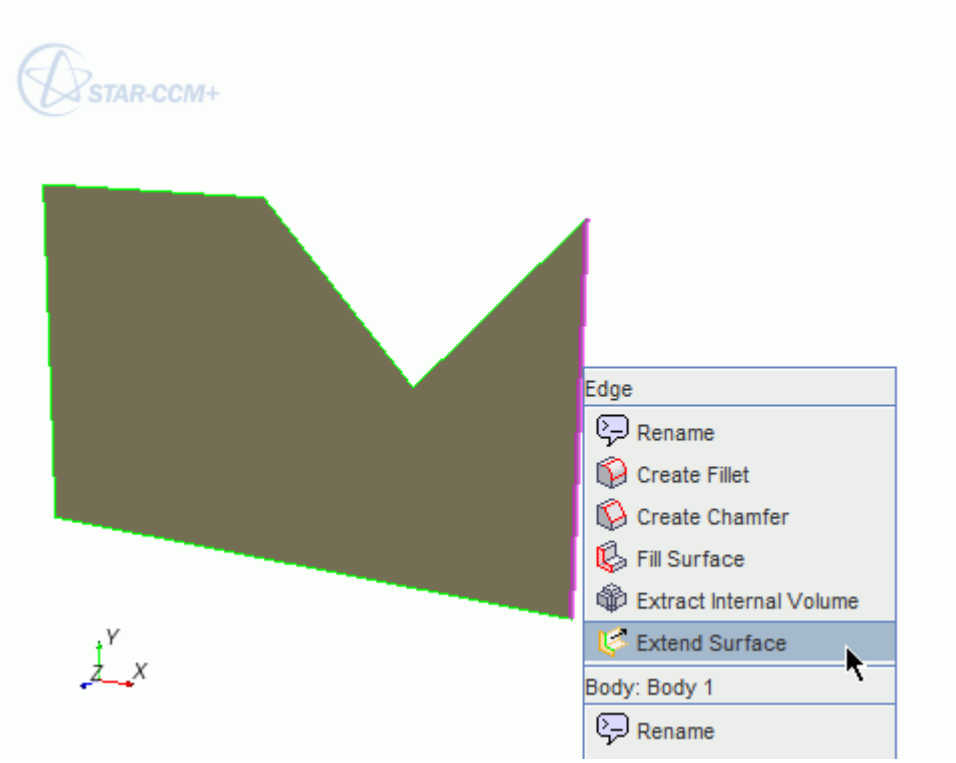
2. Click **OK**.

Extending Surfaces

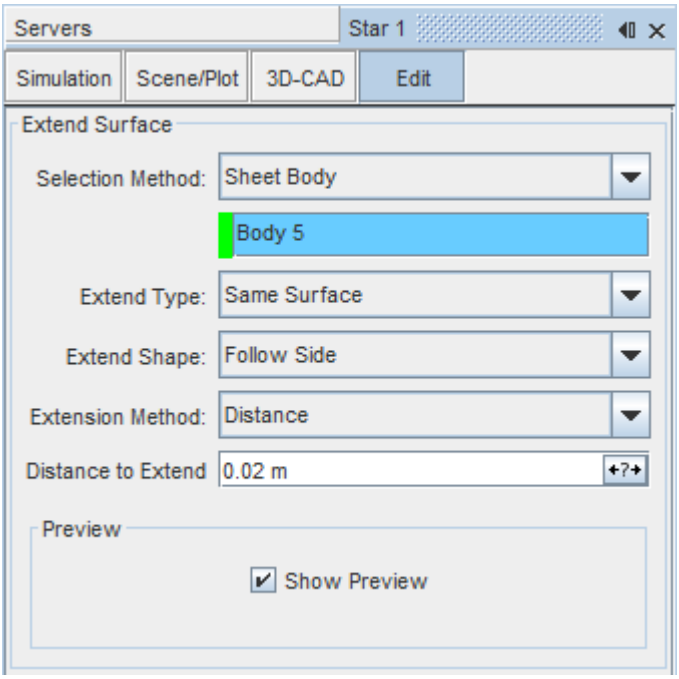
The Extend Surface feature extends the specified face or free edge while maintaining the overall shape and tangency of the source surface. You can only extend faces and free edges on sheet bodies.

To extend a surface:

1. Right-click a surface or free edge and select **Extend Surface**.

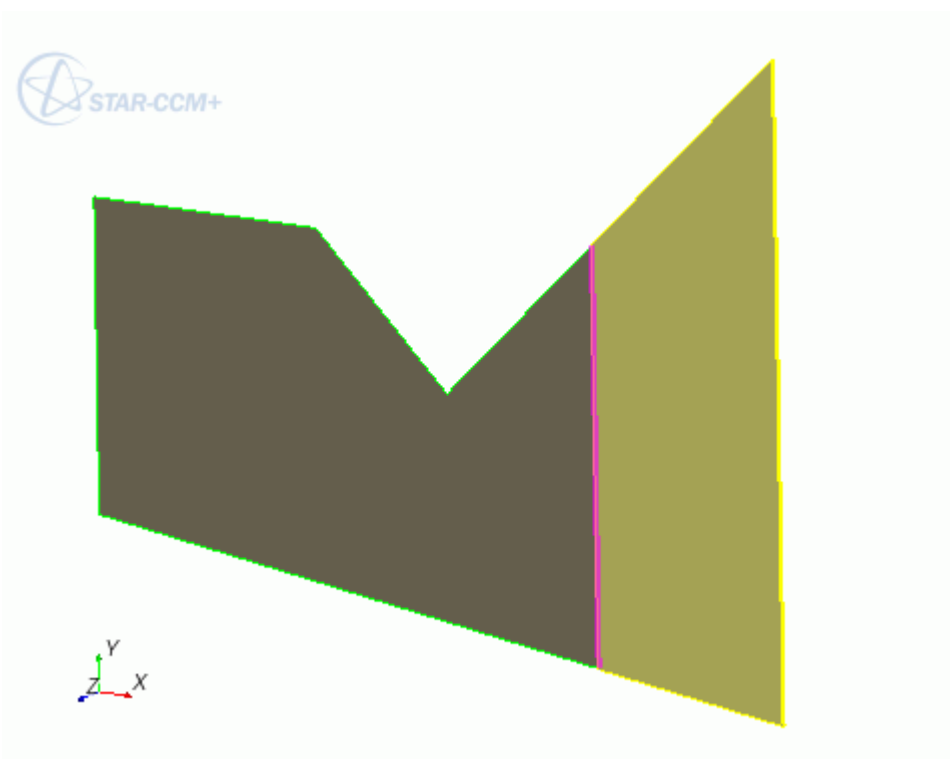


The *Extend Surface* panel appears.



2. Enter the parameters for the extended surface. See Extend Surface Panel.
3. If extending to a surface or body, click inside the *Target Face* or *Target Body* group box and select the desired target surface from the *3D-CAD View* scene.

   A preview feature is displayed in the *3D-CAD View* scene.



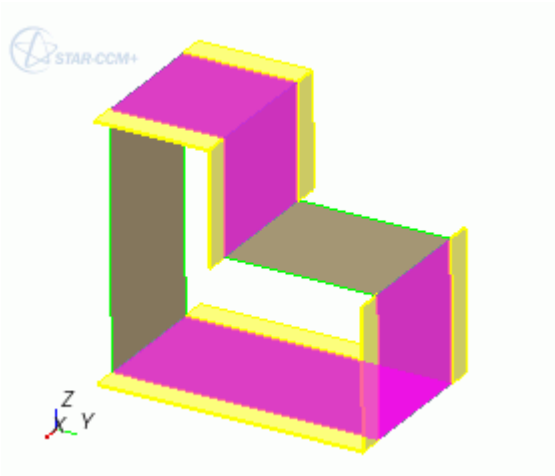
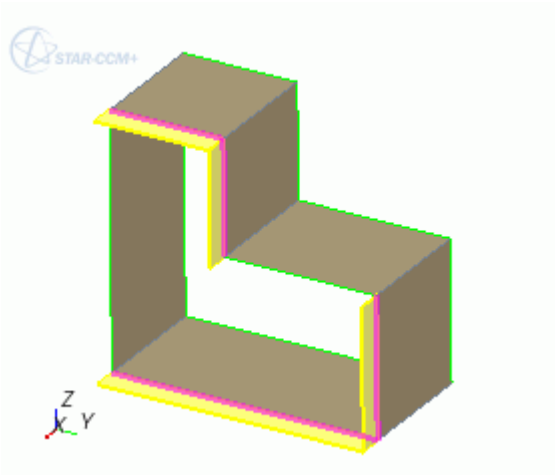
4. Click **OK**.

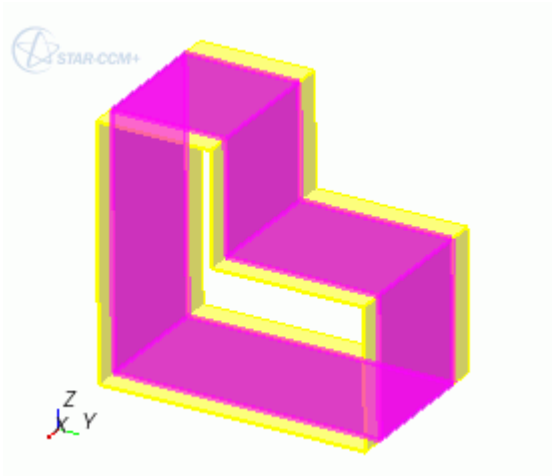
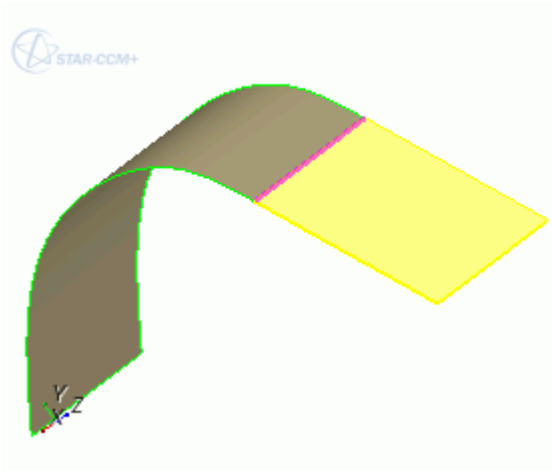
Contents:

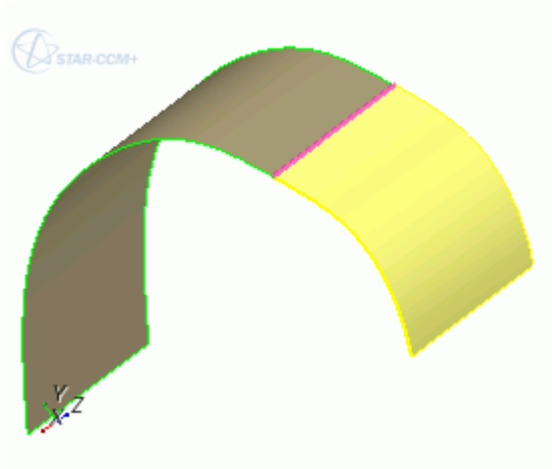
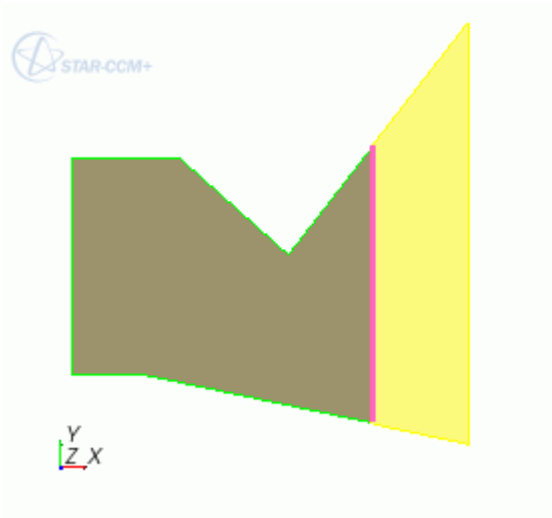
Extend Surface Panel

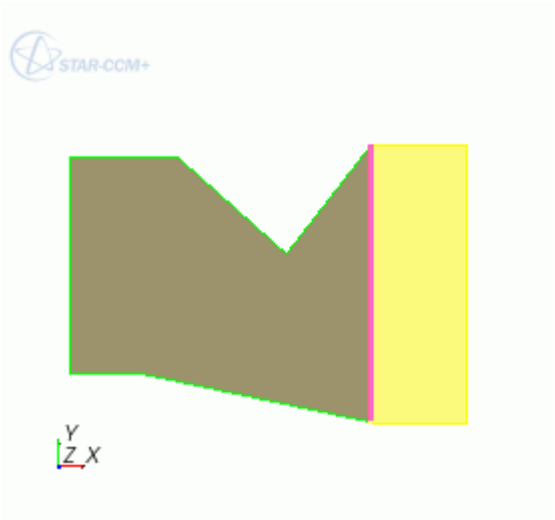
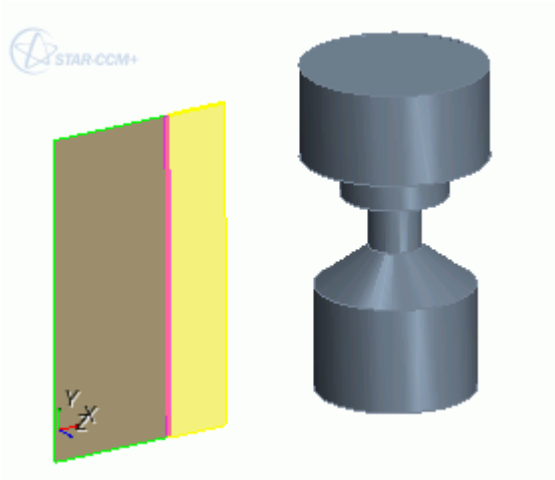
Extend Surface Panel

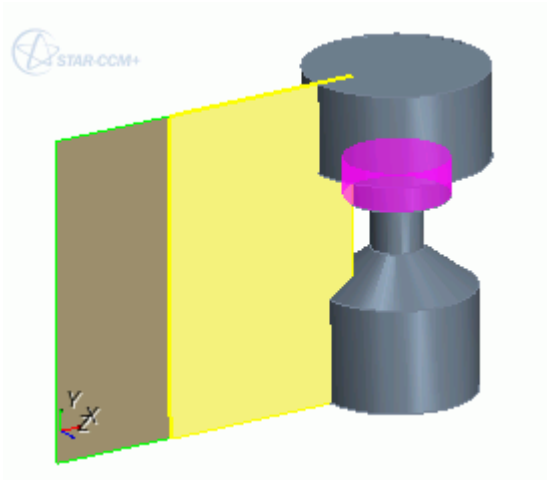
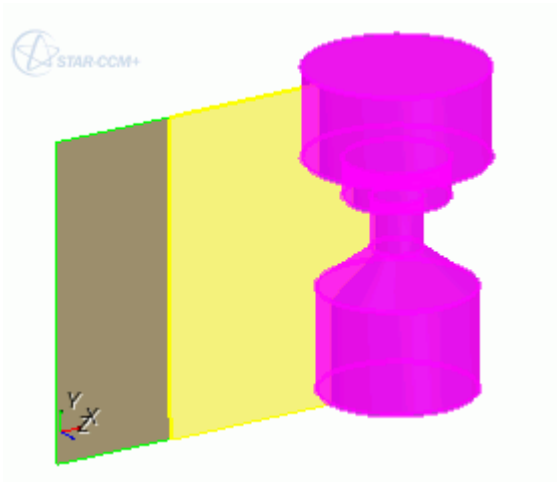
The extend surface operation extends existing surfaces on sheet bodies.

Selection Method	Specifies which part of the sheet body is used for this operation.	
	Faces or Free Edges	Faces: all free edges on the selected faces are extended.  Free Edges: the selected free edges are extended. 
	Sheet Body	All free edges of the sheet body are extended.

		
Extend Type	Specifies whether the extension follows the curvature of the sheet body from which the free edge belongs or not.	
	Linear	The extended surface is planar and tangent to the existing sheet body.  If the free edge lies on an arc, the extended surface maintains the curvature of the sheet body.
	Same Surface	The extended surface maintains the curvature of the sheet body.

		
Extend Shape	Specifies whether the extension follows the free edges of the face or whether it extends the surface normal to the edge.	
	Follow Side	Extension maintains shape of source surface. 
	Normal to Base	Extension is normal to the selected edges.

		
Extension Method	Species how to define the extension.	
	Distance	Extends blindly by the specified amount. 
	Up to Face	Extends up to a selected face. To select a face, click inside the *Target Face* group box and select the desired face in the *3D-CAD View* scene.

		
	Up to Body	Extends up to a selected body. To select a body, click inside the *Target Body* group box and select the desired body in the *3D-CAD View* scene or from the **Bodies** node in the 3D-CAD feature tree. 
Distance to Extend	Specifies how far to extend the surface.	

Filling Holes

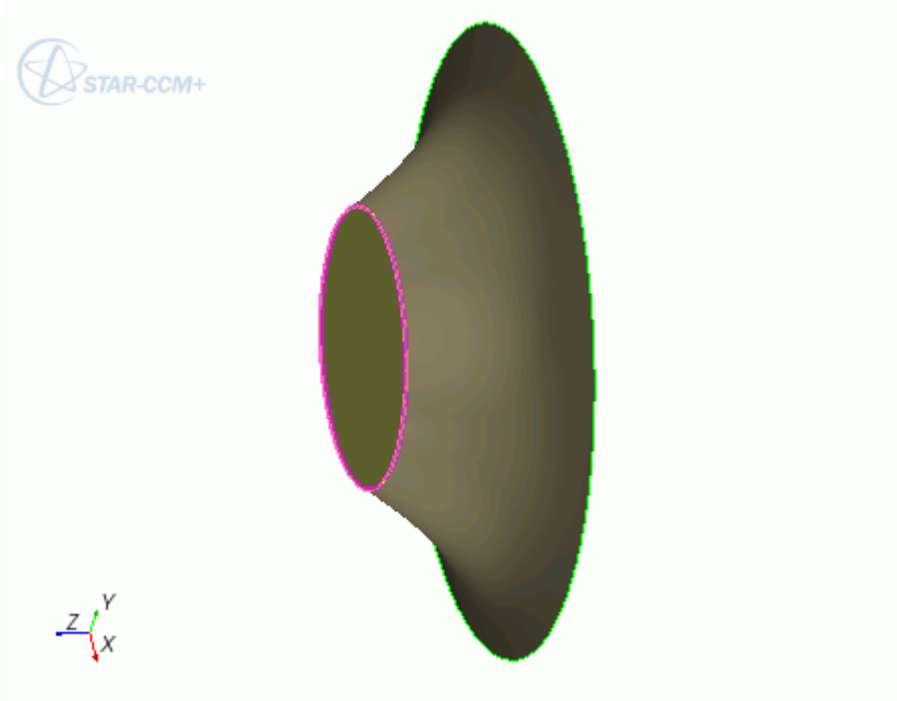
The Fill Surface feature allows you to create a surface from free edges on the perimeter of a hole in a sheet body.

The resulting surface is automatically sewn into the sheet body containing the hole. If the final sheet body forms a closed volume, STAR-CCM+ converts it to a solid body.

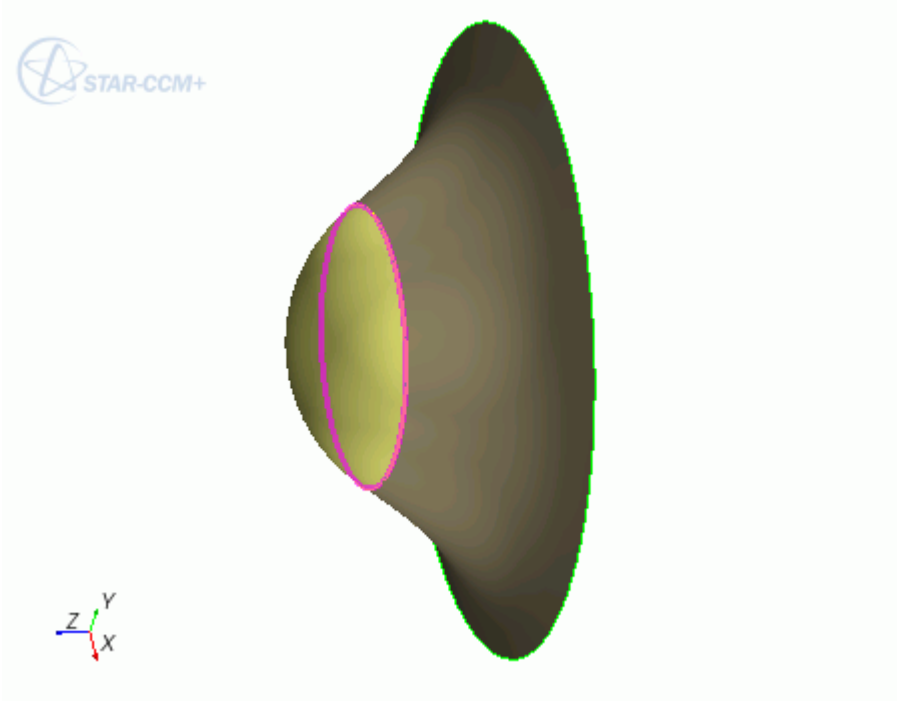
Free edges must:

- belong to a single sheet body. Edges on solid bodies are not supported.
- form a single closed loop. Filling between two concentric circles, for example, is not supported.

You can control the tangency of the filled surface. By default the filled surface is planar.



Alternatively, you can choose for the filled surface to be tangent to adjacent faces.



Contents:

Creating a Filled Surface

Fill Surface Panel

Creating a Filled Surface

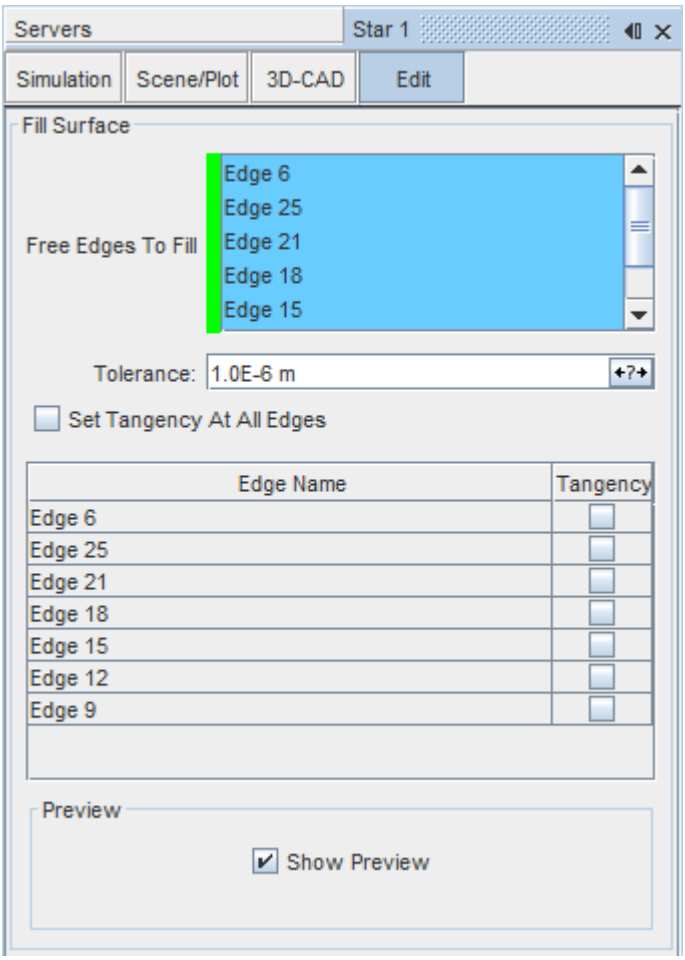
To create a filled surface:

1. In the *3D-CAD View* scene, select all the free edges surrounding the hole.

   These edges must form a closed loop. You can double-click a free edge to select all other free edges in the same loop.

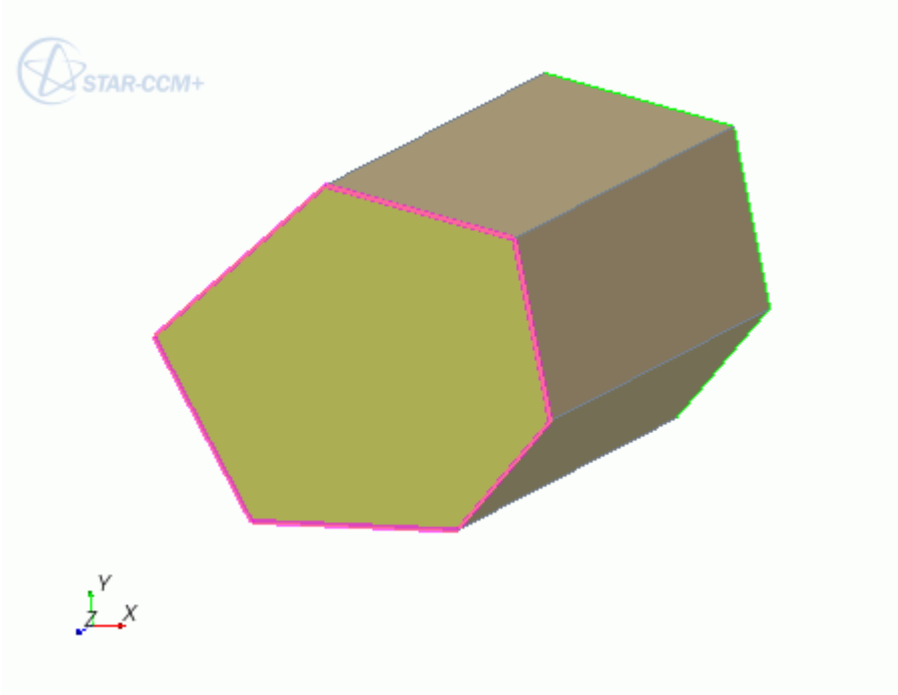
2. Right-click one of the edges and select **Fill Surface**.

   The *Fill Surface* panel appears.



3. Enter the parameters for the filled surface. See Fill Surface Panel.

A preview of the filled surface is displayed in the *3D-CAD View* scene.



4. Click **OK** to complete the feature.

A **Fill Surface** node is added to the 3D-CAD feature tree.

Fill Surface Panel

The fill surface operation creates a surface within free edges that form a closed loop.

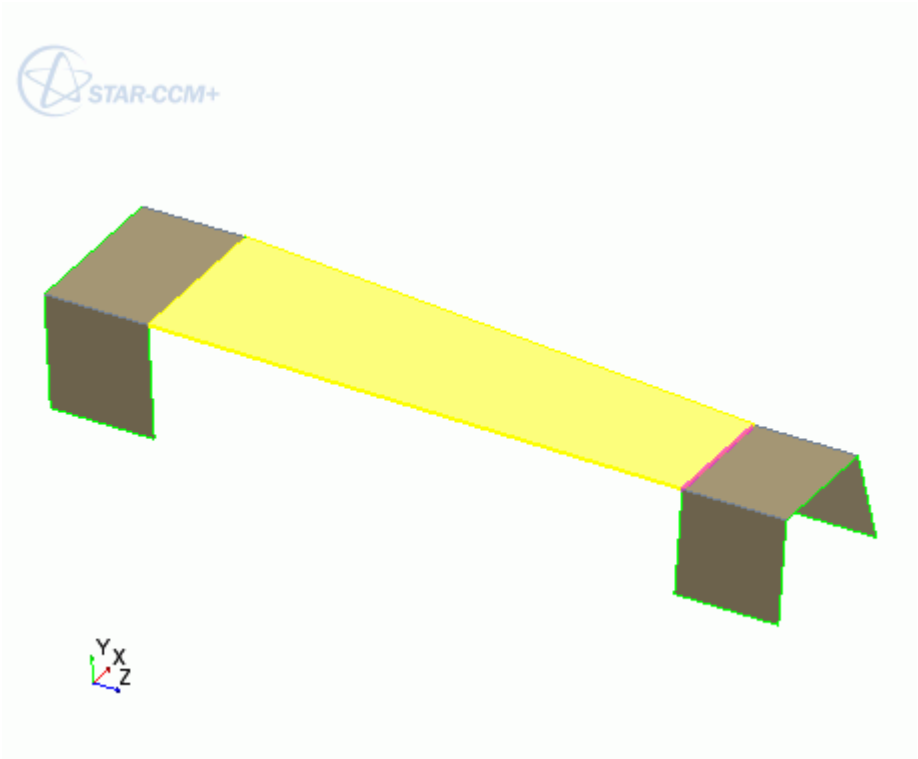
The *Fill Surface* panel options are listed below.

Free Edges To Fill	Specifies the edges to fill.	
Set Tangency at All Edges	Specifies whether to make the resulting filled surface tangential to all adjacent faces.	
Tolerance Value	Controls the overall accuracy of the result. Larger tolerance values can result in simpler geometries.	
Selected Edges	This list contains all the selected free edges. Clicking on each item in the list highlights the relevant edge in the *3D-CAD View* scene.	
	Edge Name	Lists the names of the selected free edges.
	Continuity	When activated for an edge, STAR-CCM+ maintains tangency between the fill surface and the face adjacent to the edge.

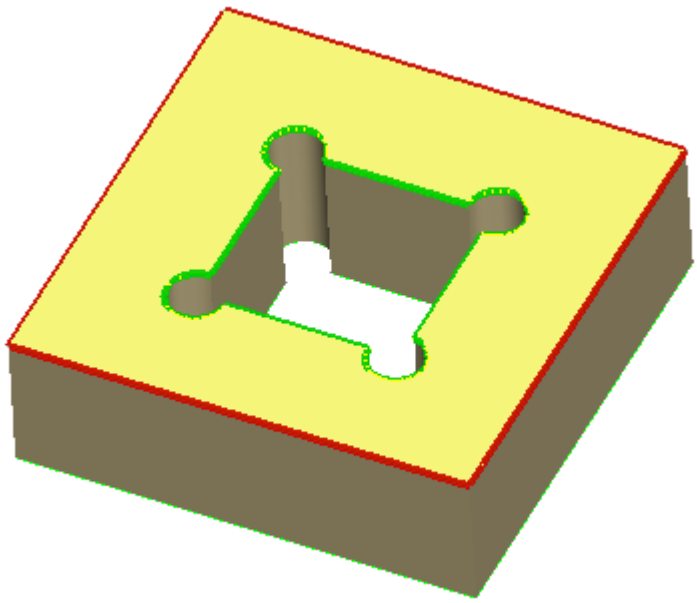
Bridge Surfaces

The Bridge Surface action creates a sheet body between two sets of edges on existing sheet bodies.

You can use this feature to repair geometries that have disjointed sheet bodies.



You can also use this feature to fill gaps between two sets of closed edges.



Prerequisites

The following conditions must be satisfied to create a bridge surface:

- Each set of edges must belong to a single body.
- Edges on each set must be connected such that they form a manifold topology.
- Both sets of edges must be either open or closed. You cannot use a combination of one open set and one closed set of edges.
- If both sets of edges lie on the same plane, for closed profiles you can have a different number of edges in either set. For sets of edges that are not on the same plane, you must have the same number of edges in each set.

Contents:

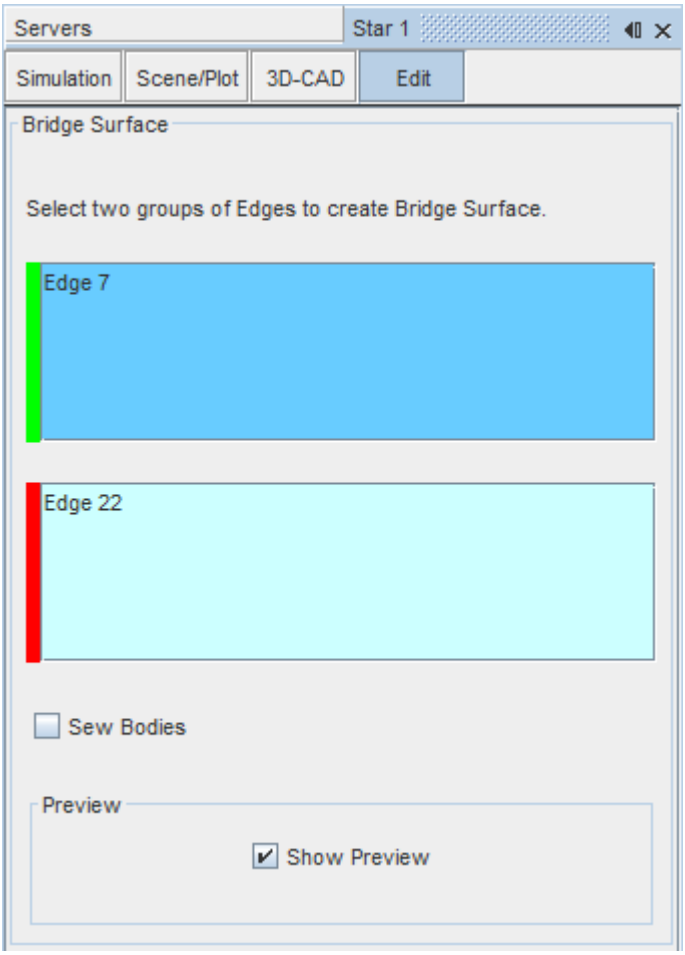
Creating a Bridge Surface
Bridge Surface Panel

Creating a Bridge Surface

To create a bridged surface:

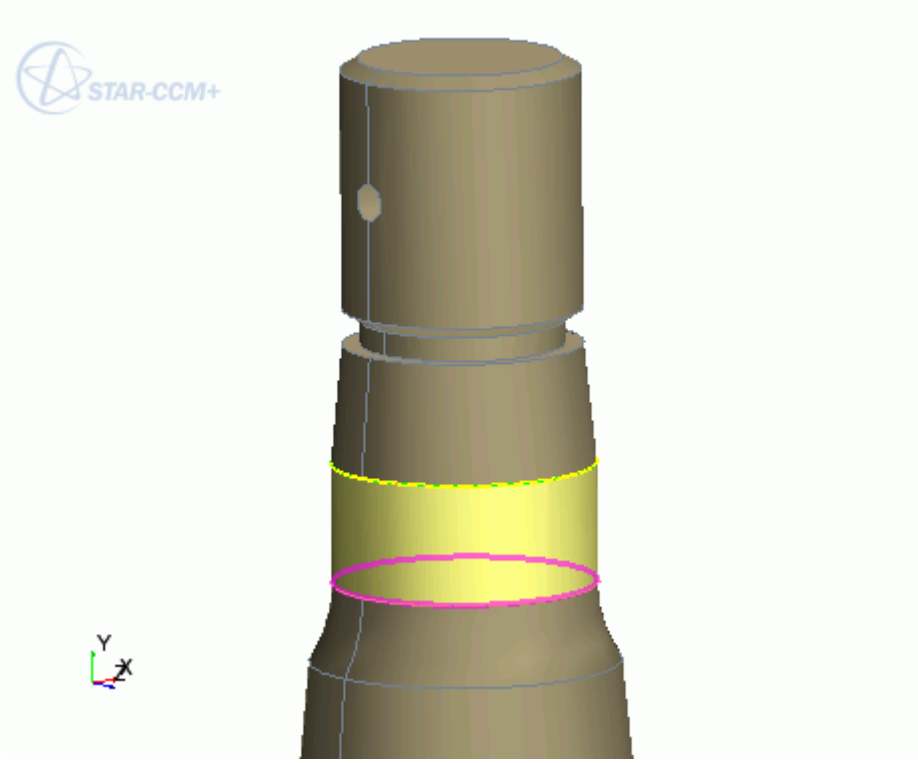
1. In the 3D-CAD View scene, select all the edges (from both bodies) that you want to use to create the bridge surface.
2. Right-click on a selection and select **Bridge Surface**.

   The *Bridge Surface* panel appears.



3. Modify the selection of source and target edges using the two group boxes.
4. Set any additional options in the panel. See Bridge Surface Panel.

   A preview of the operation is displayed in the *3D-CAD View* scene.



5. Click **OK**.

The bridge surface is created and a **Bridge Surface** feature is added to the feature tree.

Bridge Surface Panel

The Bridge Surface action creates a surface between two sets of edges.

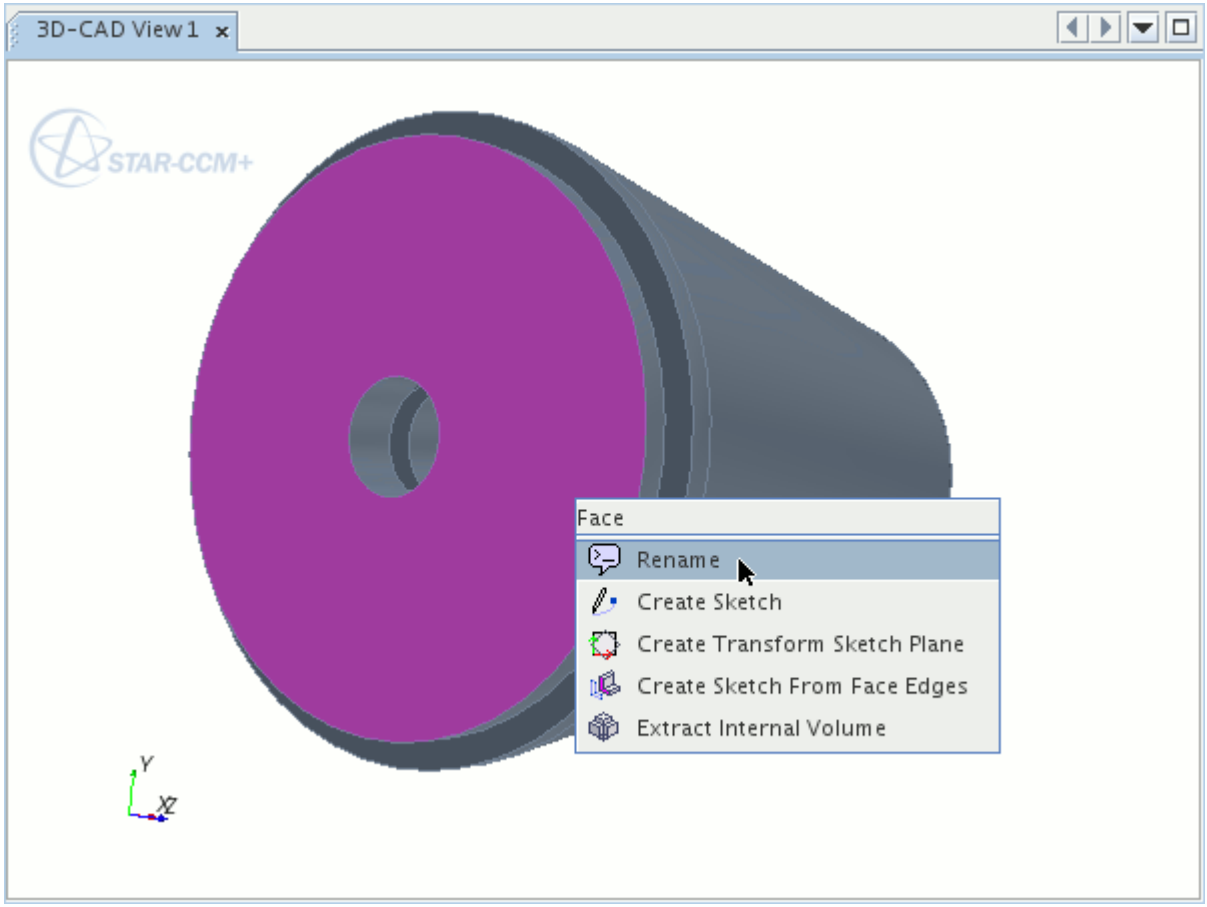
Sew	Specifies whether to sew sheet bodies together. If the resulting geometry is closed, a solid geometry is created.
Tolerance Value	Controls the tolerance for sewing sheet bodies together.

Setting Face, Edge, and Vertex Names

When you convert a 3D-CAD model to geometry parts, the faces of the model become part surfaces and the edges become curves. By naming a face on the 3D-CAD model, it is defined as a separate surface within the part definition, which makes it easier to specify boundaries later on. Named edges on the 3D-CAD model are defined as separate curves within the part definition. Named vertices are defined as part points in the part definition and can be used in FEA applications. All named faces, edges, and vertices are listed in the 3D-CAD feature tree, where you can edit and delete them. Faces and edges that have no given name are labelled as **Default**.

To set the name of a face, edge, or vertex:

1. Select the face, edge, or vertex in the *3D-CAD View* window. To select multiple items, hold down the **<Ctrl>** key.
2. Right-click the selection and select **Rename**.



3. In the inline editor, enter the new name for the selection.
4. Press `<Enter>`.

Assigning the same name to multiple faces/edges/vertices adds them to the same surface/curve/point in the part definition. You can give selected entities the same name in one operation (as described above), or give entities the same name individually.

You are advised to rename entities after you complete the 3D-CAD model. Changes that you make to the model while working on it can affect the name settings. In general, a face name is retained when you modify a face, providing that some of the original face still exists. If the face itself is split across multiple bodies, only one segment of the face retains the name, and the other segments of the face have no name.

See Also:

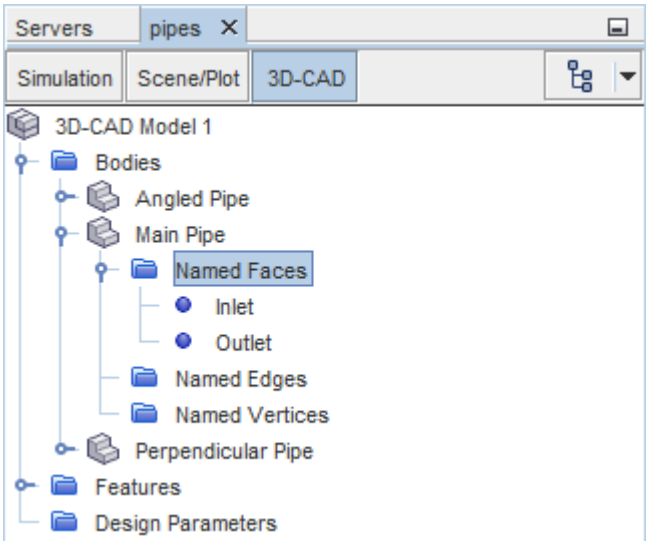
Creating Geometry Parts from 3D-CAD

Working with Custom Name Attributes

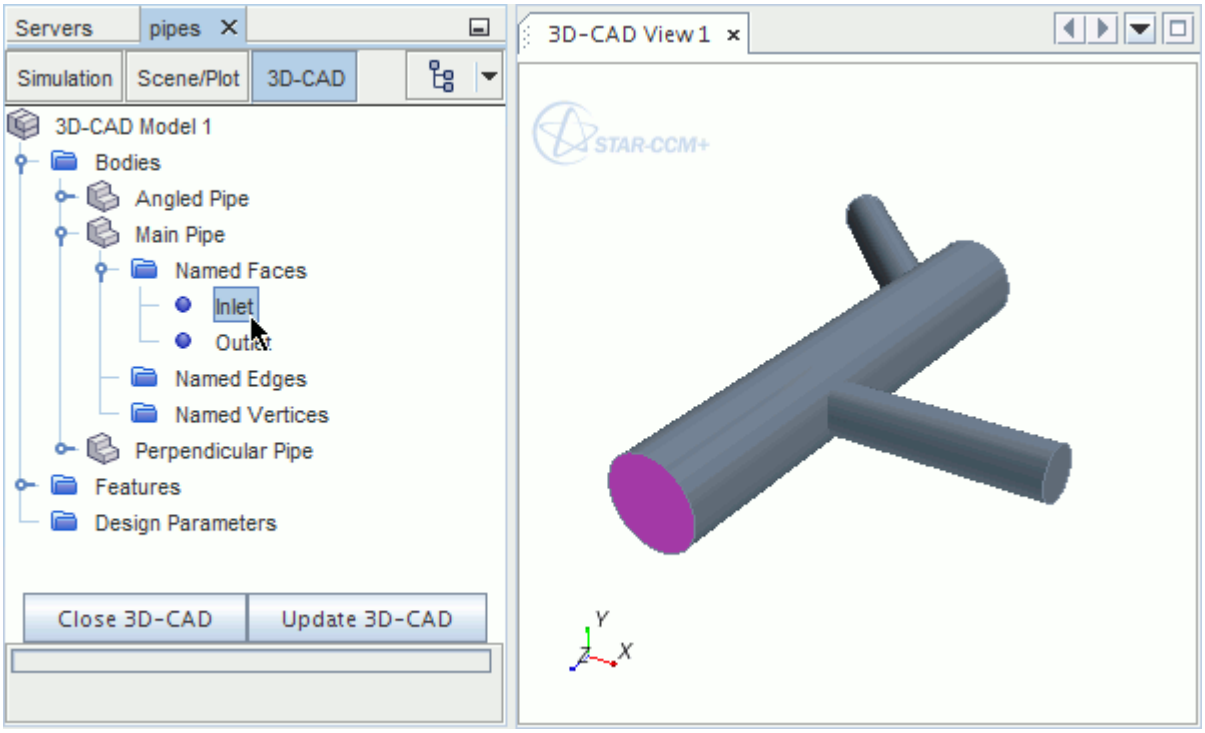
Visualizing Named Faces, Edges, and Vertices

Named entities (faces, edges, and vertices) are grouped under the **Bodies** > **Body** > **Named [Entity]** node in the 3D-CAD object tree. Named faces in imported CAD geometries (for supported formats only) are maintained and added to the **Named [Entity]** node at the end of the import process.

In the example below, the named faces of **Main Pipe** are shown.



Clicking the nodes highlights the relevant entities in the *3D-CAD View* scene.

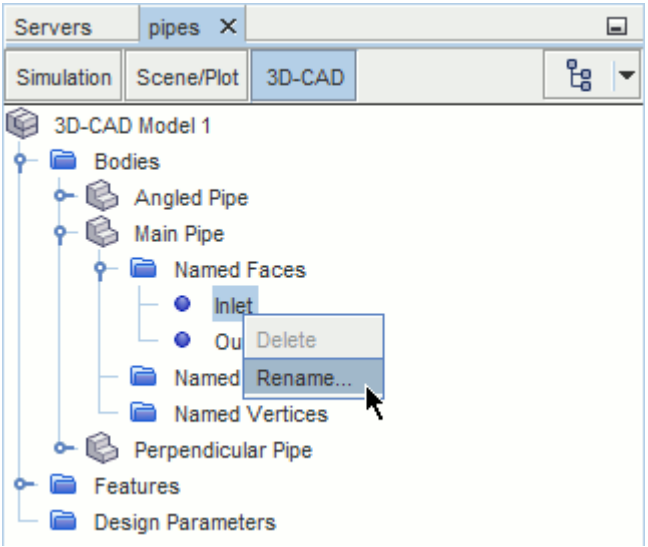


Multiple entities of the same type (such as faces) that share a name are stored under one node in the object tree and appear in the *3D-CAD View* scene when you select the node.

Editing Names From the 3D-CAD Feature Tree

To rename an entity that is listed:

1. In the 3D-CAD feature tree, right-clicking the desired face/edge/vertex name node and select **Rename...**



2. Enter the new name in the inline editor.
3. Press `<Enter>`.

Names that you edit this way are not sorted in alphabetical order. To sort the face names by alphabetical order:

- Right-click on the **Named Faces** manager node and select **Refresh**.

Removing Names From Entities

To remove a name from an entity:

- In the 3D-CAD feature tree, right-click on a face/edge/vertex name node and select **Delete**.

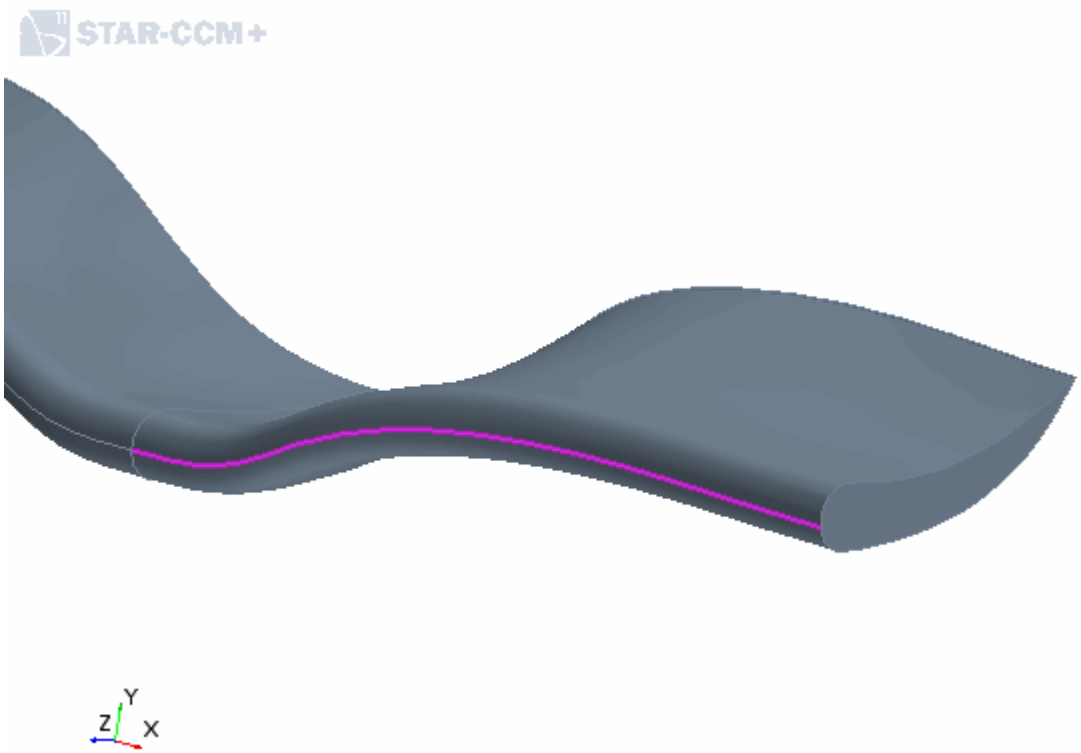
Splitting a Face by Offsetting an Edge

The Offset Edges tool splits a specified face by offsetting selected edges.

This feature is useful for splitting fillets to provide greater control when meshing the geometry.

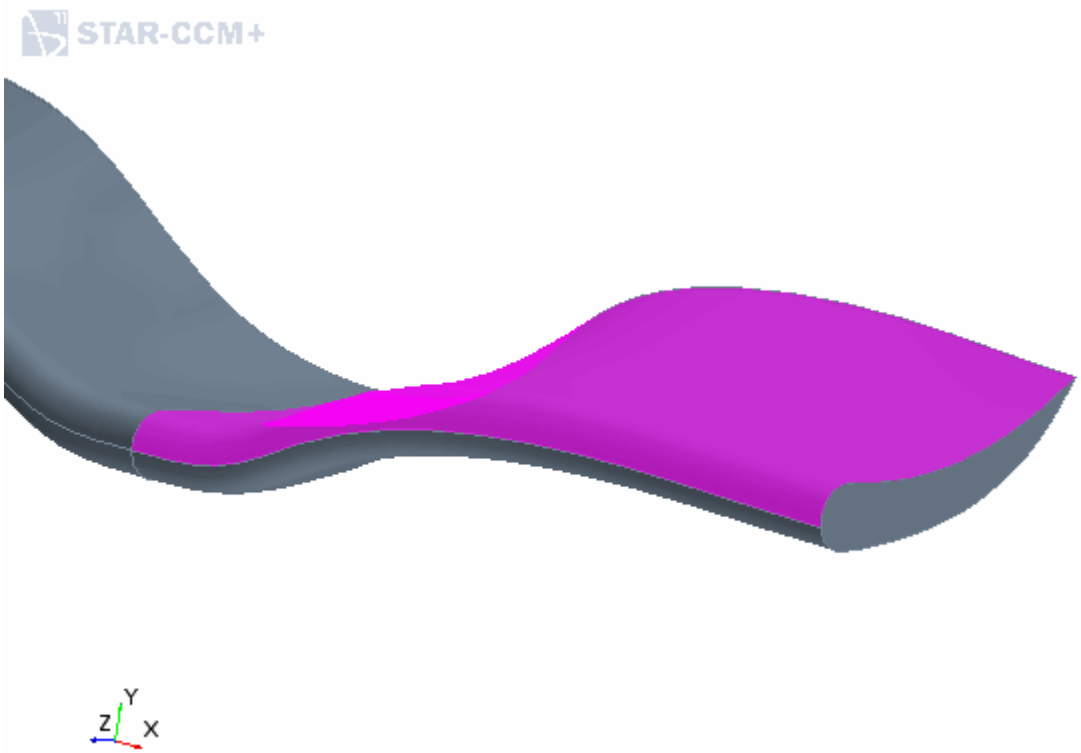
To split a face by offsetting an edge:

1. In the *3D-CAD View* scene, select the edge that you want to offset.
   You can select multiple connected edges.



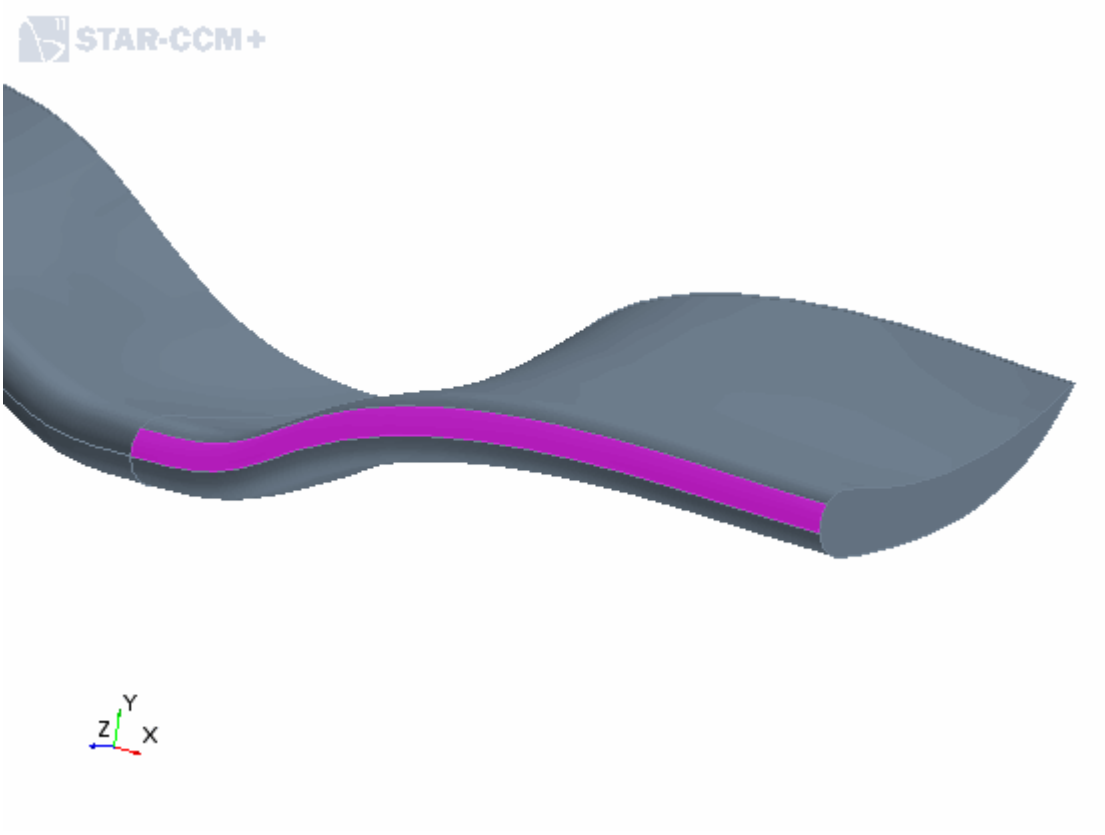
2. Select the face that you want split.

   You can select only one face to split at a time.



3. Right-click a selected edge and choose **Offset Edges**.
4. In the *Offset Edges* panel, set the split parameters. See Offset Edges Panel.

The face is split using the edges that you selected. You can rename and use the new patches in downstream applications. If the edge extends beyond the face, for example when you select a patch, the entire surface is split.



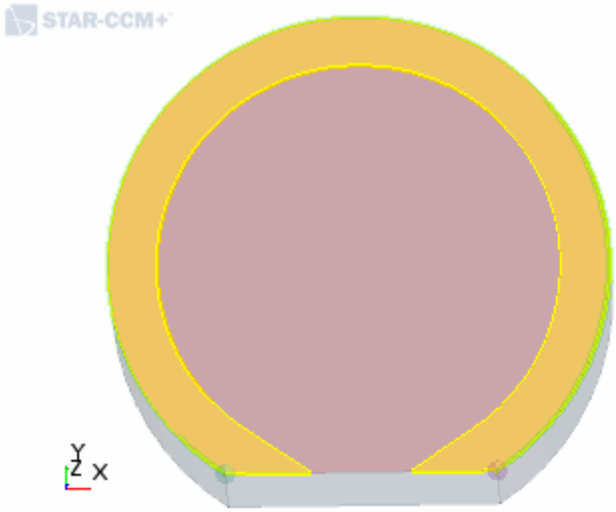
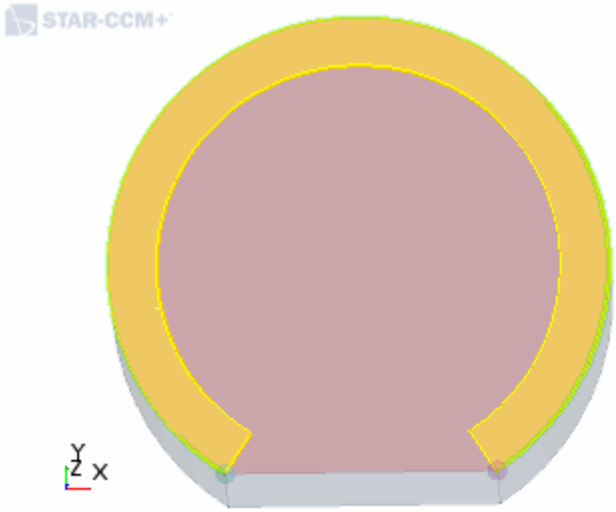
Contents:

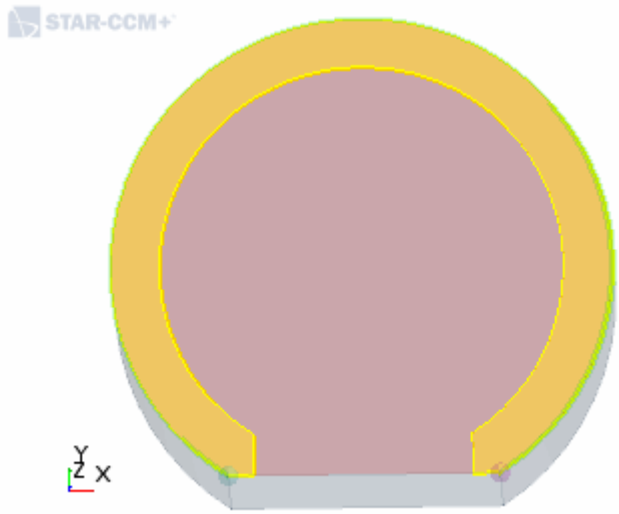
Offset Edges Panel

Offset Edges Panel

Offset edges to split a face.

Edges to Offset	Specifies the edges to offset on the selected face. You can select multiple edges that are connected.	
Reference Face	Specifies the face to split using the offset edges. You can select only one face at a time, however, faces that are tangent to the selected face are also split in the operation. Faces can be curved.	
Offset Distance	Specifies the distance to offset the selected edges.	
End Conditions	Controls the shape of the offset at the end points. You control the shape of the offset near each end point individually through the **Extend Type** parameter.	
	Tangent	Maintains the tangency of the offset edge.

		
	Normal	Makes the offset edge normal to the source edge. 
	Closest	Uses a combination of tangent and normal edges at the end points.

Filtering Selections

While working in the 3D-CAD environment, you can filter selections to make it easier to select particular bodies or features in the geometry model.

The selection filter in the 3D-CAD environment is different to that in the main STAR-CCM+ environment. The 3D-CAD selection filter is more tailored towards CAD geometry models and provides more filters.

To filter selections in 3D-CAD:

1. Click (**Selection Filter**) in the *Selection* toolbar and activate the desired filters.

   You can select any combination of filters:

   - Bodies
   - Faces
   - Edges
   - Vertices
   - Reference Geometry
2. Select the entities from the geometry model in the *3D-CAD View* scene.

   Entities outside the filter parameters are ignored.

Working with Patterns

The linear pattern and circular pattern features are used to reproduce seed (original) bodies multiple times within a 3D-CAD model.

The arrangement of the bodies follows a regular pattern, which is defined in the pattern dialog.

Contents:

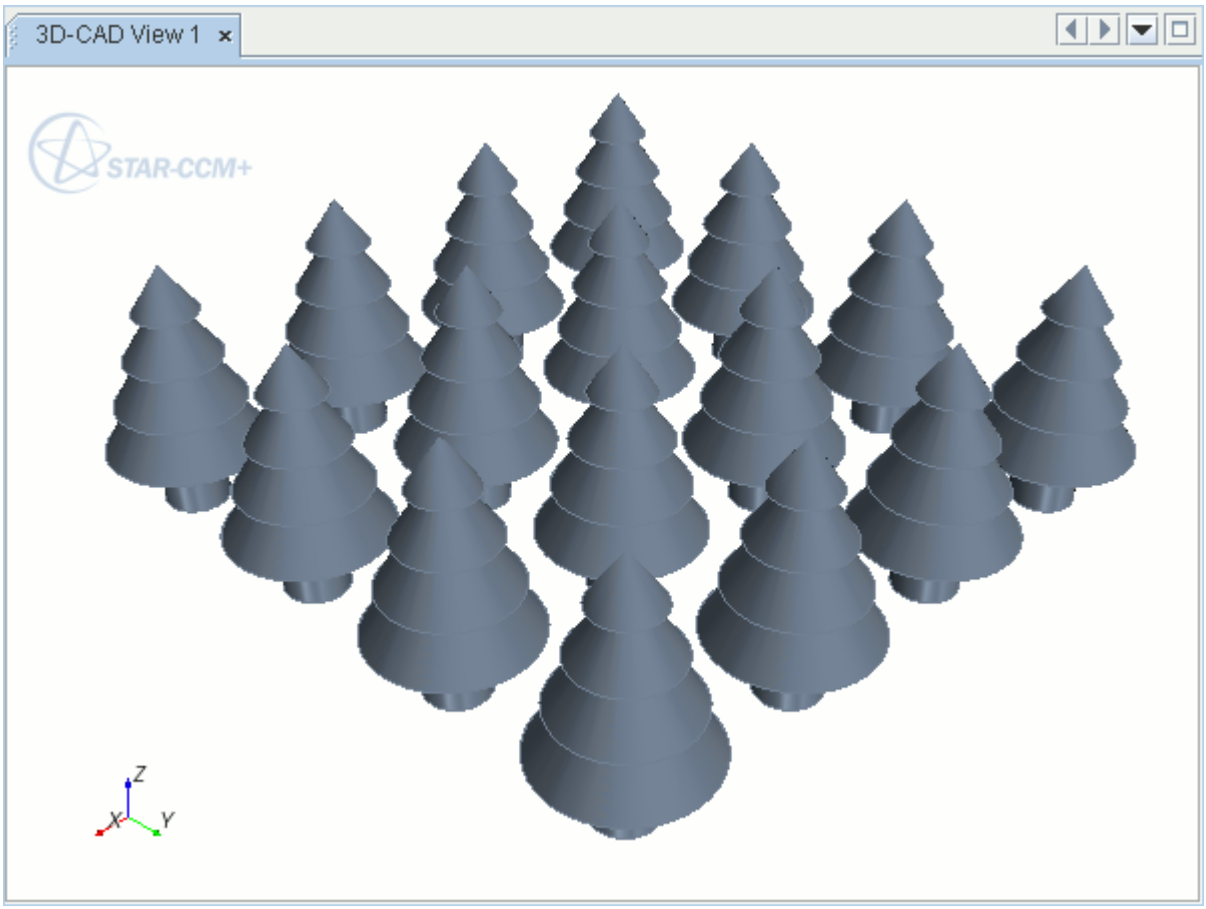
Working with Linear Patterns
Working with Circular Patterns

Working with Linear Patterns

The linear pattern tool is used to reproduce a series of identical bodies, from seed bodies, which follow a regular formation along straight lines.

You can create a pattern feature from any number of bodies with the properties defined in the linear pattern dialog. A linear pattern cut is created in a similar way, however the seed bodies and pattern determine where material is removed.

You define the pattern by one or two linear directions extending from the original body. The number of bodies and the distance between them determine the extent of the pattern.



Contents:

Creating a Linear Pattern
Body Linear Pattern Panel
Creating a Linear Pattern Cut

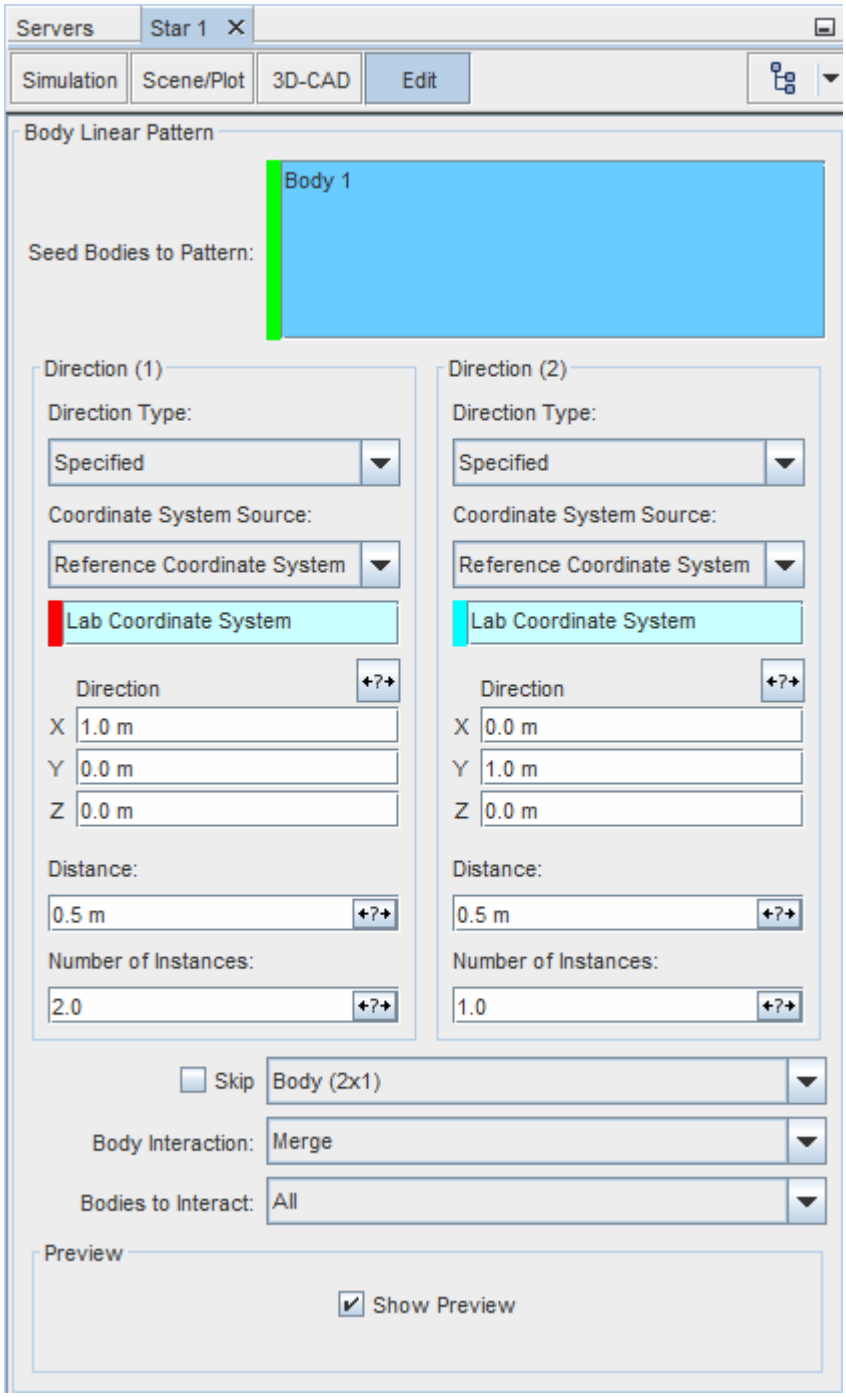
Creating a Linear Pattern

When creating a linear pattern, you require at least one seed body.

To create a linear pattern:

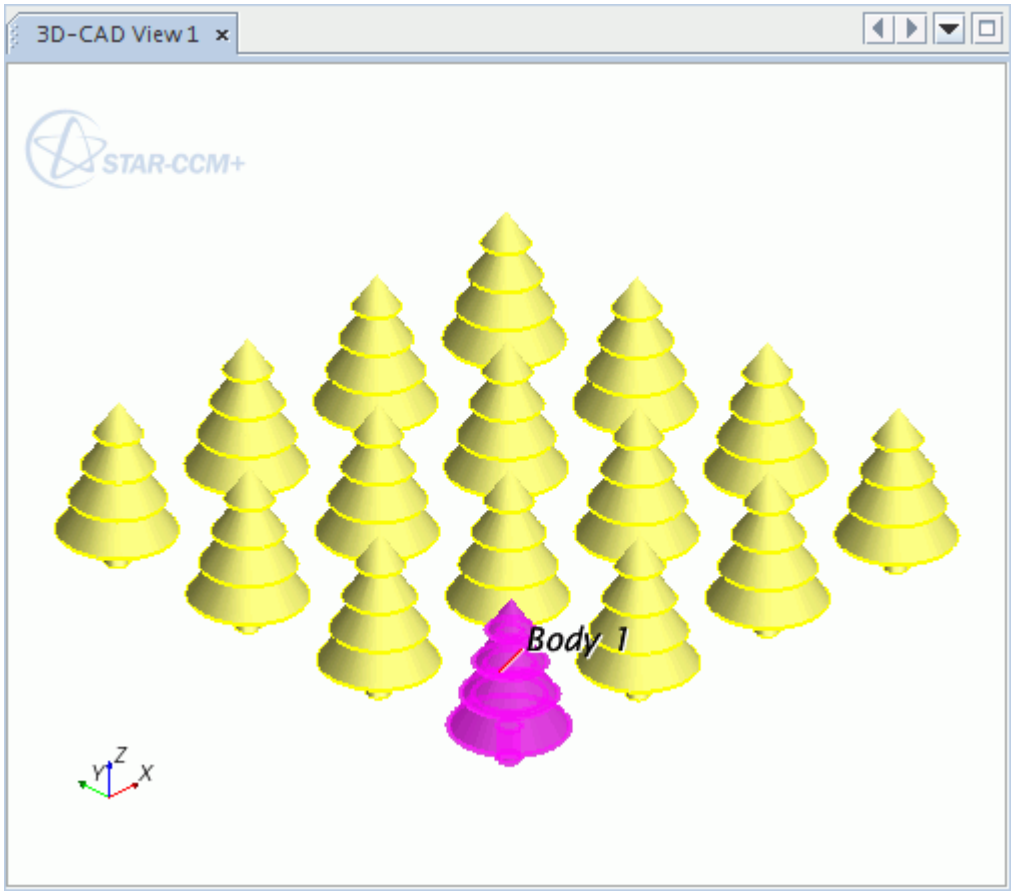
1. In the **Bodies** manager node, right-click on a seed **Body** and select **Pattern** > **Linear Pattern**.

   The *Body Linear Pattern* panel appears.



2. To modify the seed body selection, click inside the *Seed Bodies to Pattern* group box and select the bodies from the *3D-CAD View* scene or from the 3D-CAD feature tree.
3. Use the *Body Interaction* option to control how patterned bodies to interact with existing bodies.
4. Set up the remaining linear pattern options. See Body Linear Pattern Panel.
5. Expose any options as design parameters by clicking the corresponding (**Expose Parameter**) button.

   A preview of the pattern is displayed in the *3D-CAD view* scene.



6. Click **OK**.

The pattern is created. Additional bodies are added to the **Bodies** node of the 3D-CAD object tree. Each new body has the name of the seed body plus its position index. For example, if you use a body (named `Tree`) to create a 2 x 2 matrix pattern, the new bodies are called `Tree (1 x 2)`, `Tree (2 x 1)`, and `Tree (2 x 2)`. The generalized naming convention is `Tree (i x j)`, where `i` is the row number and `j` is the column number in the pattern. If you rename the seed body and click the **Update** button, all child bodies adopt the new name plus the position index, as previously described.

You can select each new body independently from the rest of the pattern. Therefore you can apply body operations, such as transform, unite, intersect, subtract, and imprint, to individual bodies.

See Also:

Transforming a Body

Body Linear Pattern Panel

A list of parameters that are found in the *Body Linear Pattern* panel is provided in the following table.

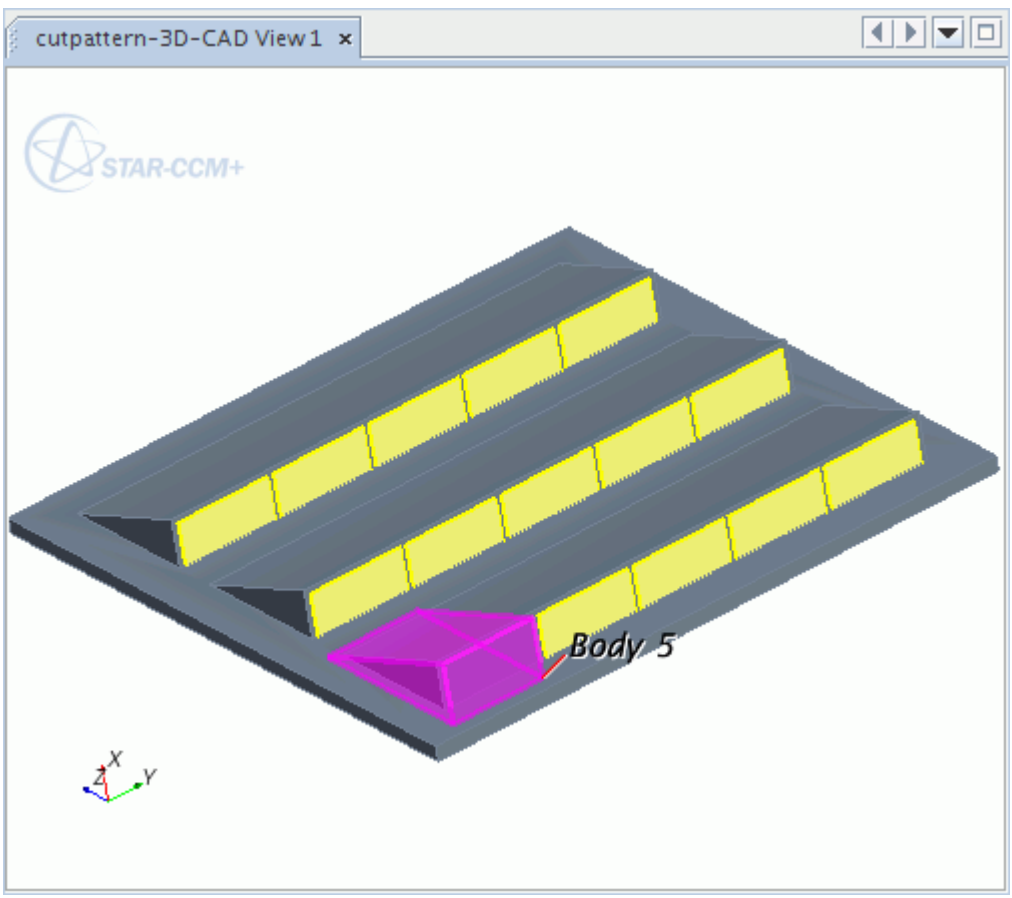
Seed Bodies to Pattern	Specifies the seed bodies for the pattern. To modify the seed body selection, click inside the *Seed Bodies to Pattern* group box and select the bodies from the *3D-CAD View* scene or from the 3D-CAD feature tree.		
Direction (n)	Specifies the parameters for one of the two directions in the linear pattern.		
	Direction Type	Specifies whether to define the direction of the pattern using a specified vector or using a reference axis.	
		Specified	Extends the pattern in a custom direction, whose vector components are specified in the *Direction* property.
		Reference Axis	Extends the pattern in the direction of a reference axis. Select an existing reference axis from the *3D-CAD View* scene. See Working with Reference Geometry in 3D-CAD.
	Coordinate System Source	Specifies the coordinate system that is used in the operation. Custom coordinate systems that you specified in the simulation are listed here.	
		Imported	Uses the imported coordinate system for the operation.
		Reference Coordinate System	Uses a reference coordinate system for the operation. Select an existing reference coordinate system from the *3D-CAD View* scene. See Working with Reference Geometry in 3D-CAD.
	Direction	Specifies the direction that pattern extends towards. Enter the [X, Y, Z] components of the direction vector using the global coordinate system.	
	Distance	Specifies the gap between each body instance, in meters. To use different units, enter the value with the new units.	
	Number of Instances	Specifies how many times the body is reproduced in each direction. The value that you enter must be a whole number. If you enter a decimal value, it is rounded down to a whole number.	
Coordinate System	Specifies the coordinate system that is used in the operation. Custom coordinate systems that you specified in the simulation are listed here.		
Skip	Allows you to ignore specified bodies in the feature creation. To skip a body, select the body position from the drop-down menu and activate this option. To skip multiple bodies, select another body from the drop-down menu and activate the option for each body.		
Body Interaction	Specifies how the pattern interacts with other bodies adjacent to it, as defined by **Interacting Bodies**. The bodies must coincide or overlap for these options to work.		
	None	Creates new bodies from the linear pattern only. This option does not affect any existing bodies.	
	Merge	Merges all linear pattern bodies into the existing body. No new bodies are created from the pattern.	
	Imprint	Creates new bodies from the pattern. The edges and vertices are imprinted on the affected faces of each body.	

Bodies to Interact	Specifies the bodies which interact with the pattern.	
	All	The pattern interacts with all contacting bodies.
	Selected	The pattern interacts with selected bodies only. Click inside the *Interacting Bodies* group box and select the bodies to interact with from the *3D-CAD View* scene or from the **Bodies** node in the 3D-CAD feature tree.
Bodies to Cut	Specifies the bodies that the pattern cuts.	
	All	The pattern cuts all bodies.
	Selected	The pattern cuts only selected bodies. Click inside the *Interacting Bodies* group box and select the bodies to cut from the *3D-CAD View* scene or from the **Bodies** node in the 3D-CAD feature tree.

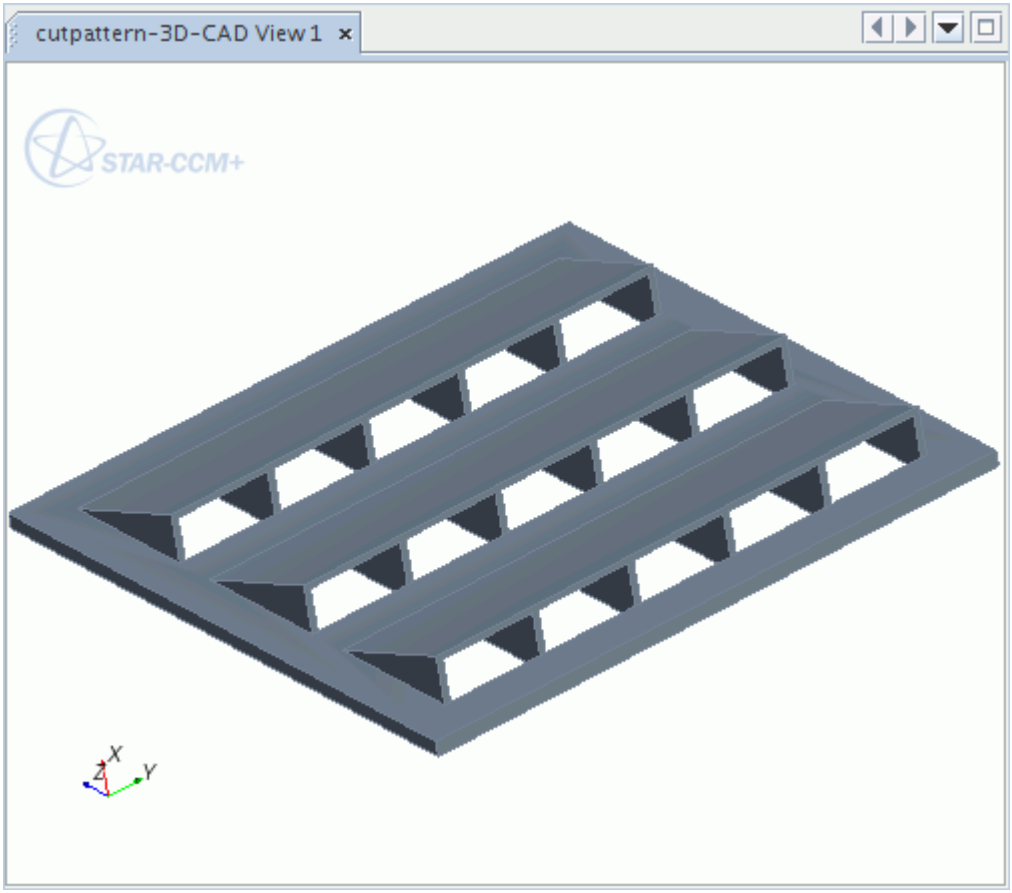
Creating a Linear Pattern Cut

Linear pattern cuts behave in the same way as linear patterns, except that the seed body defines a pattern that removes material from existing bodies.

You require at least two bodies to use a linear pattern cut.



To create a linear pattern cut, see Creating a Linear Pattern. An example of a linear pattern cut is shown below.

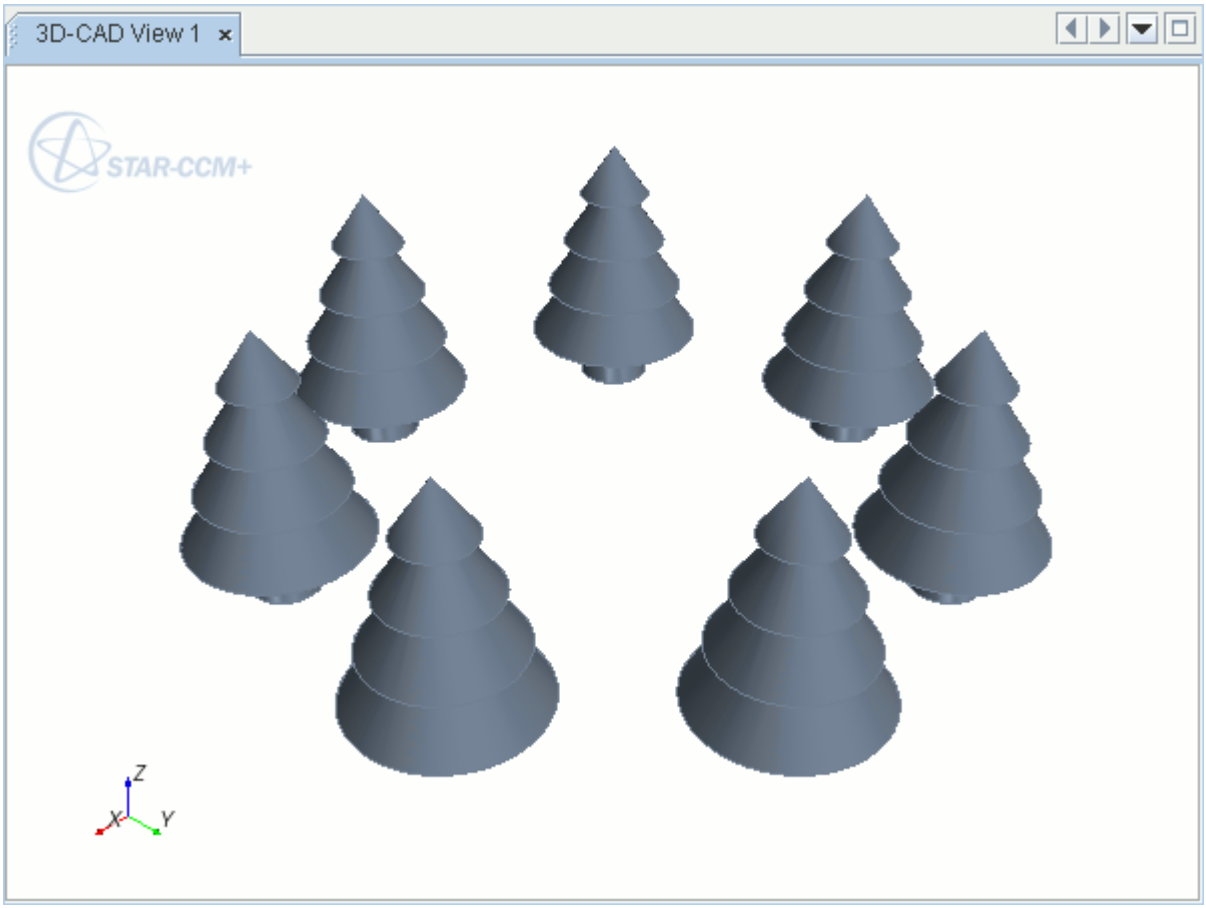


Working with Circular Patterns

The circular pattern tool is used to reproduce a series of identical bodies, from seed bodies which follow a regular formation around a circle.

You can create a pattern feature from any number of bodies with the properties defined in the circular pattern dialog. A circular pattern cut is created in a similar way, however the seed body and pattern determine where material is removed from the model.

The pattern is created around a circle, which is defined by the position and direction of a central axis. The extent of the pattern is determined by the number of bodies and the angular increment between them, or the full circumference of the circle.



Contents:

Creating a Circular Pattern
Body Circular Pattern Panel
Creating a Circular Pattern Cut

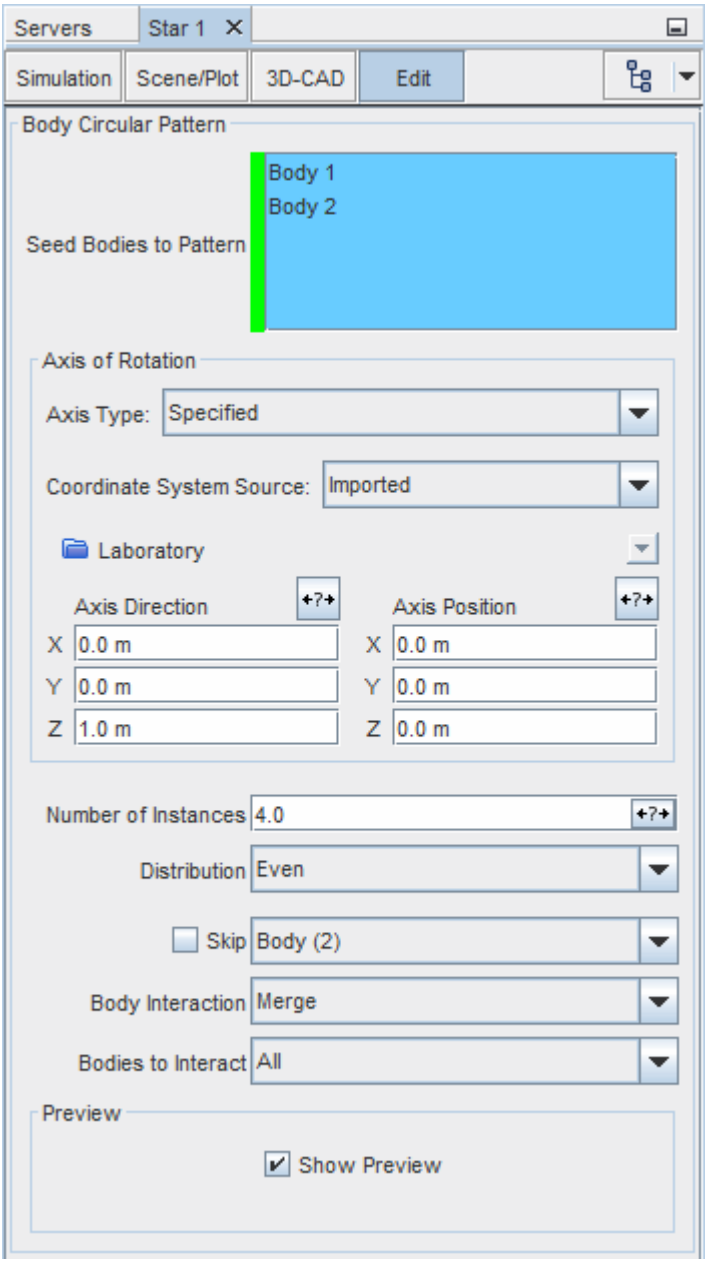
Creating a Circular Pattern

When creating a circular pattern, you require at least one seed body.

To create a circular pattern:

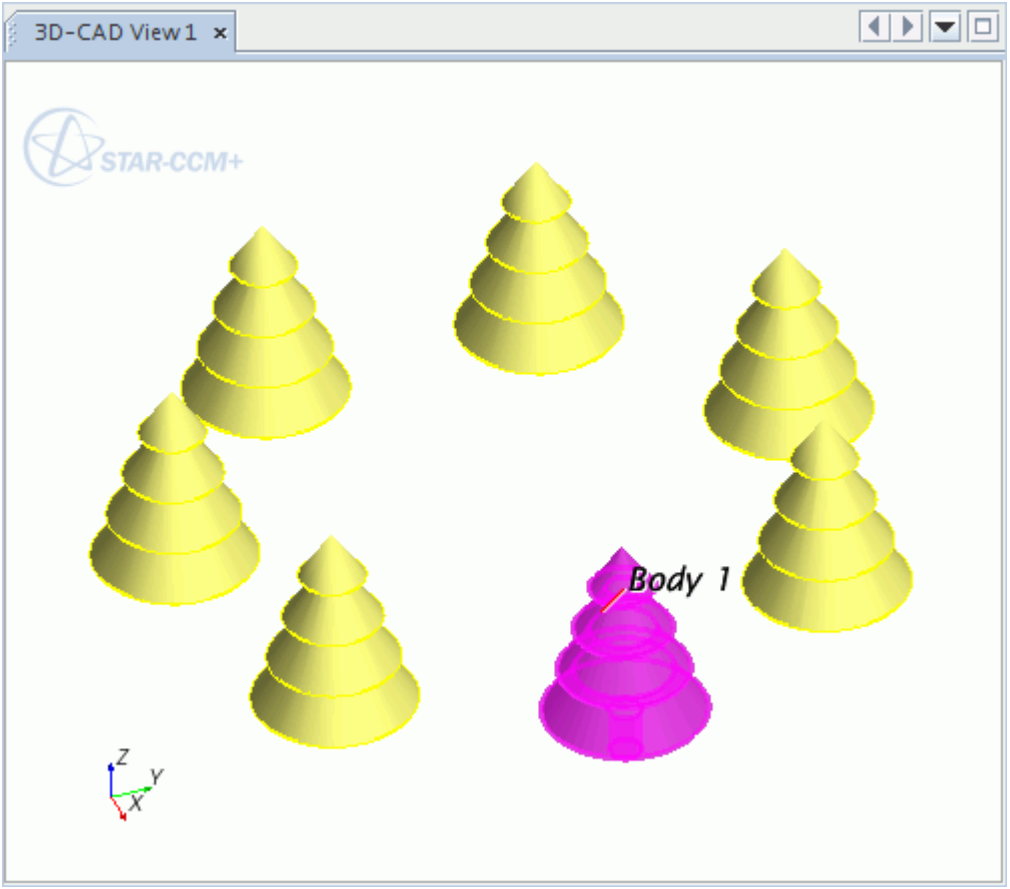
1. In the **Bodies** manager node, right-click on a seed **Body** and select **Pattern** > **Circular Pattern**.

   The *Body Circular Pattern* panel appears.



2. To modify the seed body selection, click inside the *Seed Bodies to Pattern* group box and select the bodies from the *3D-CAD View* scene or from the 3D-CAD feature tree.
3. Use the *Body Interaction* option to control how patterned bodies to interact with existing bodies.
4. Set up the remaining circular pattern options. See Body Circular Pattern Panel.
5. Expose any options as design parameters by clicking the corresponding (**Expose Parameter**) button.

   A preview of the operation is displayed in the 3D-CAD View scene.



6. Click **OK**.

The pattern is created. Additional bodies are added to the **Bodies** node of the 3D-CAD object tree. Each new body has the name of the seed body plus the instance number. For example, if you use a body (named `Tree`) to create a circular pattern with 4 instances, the new bodies are called `Tree (n)`, where `n` is the instance number of the body. If you rename the seed body and click the **Update** button, all child bodies adopt the new name plus the instance number, as previously described.

You can select each new body independently from the rest of the pattern. Therefore you can apply body operations, such as transform, unite, intersect, subtract, and imprint, to individual bodies.

See Also:

Transforming a Body

Body Circular Pattern Panel

A list of parameters that are found in the *Body Circular Pattern* panel is provided in the following table.

Seed Bodies to Pattern	Specifies the seed bodies for the pattern. To modify the seed body selection, click inside the *Seed Bodies to Pattern* group box and select the bodies from the *3D-CAD View* scene or from the 3D-CAD feature tree.

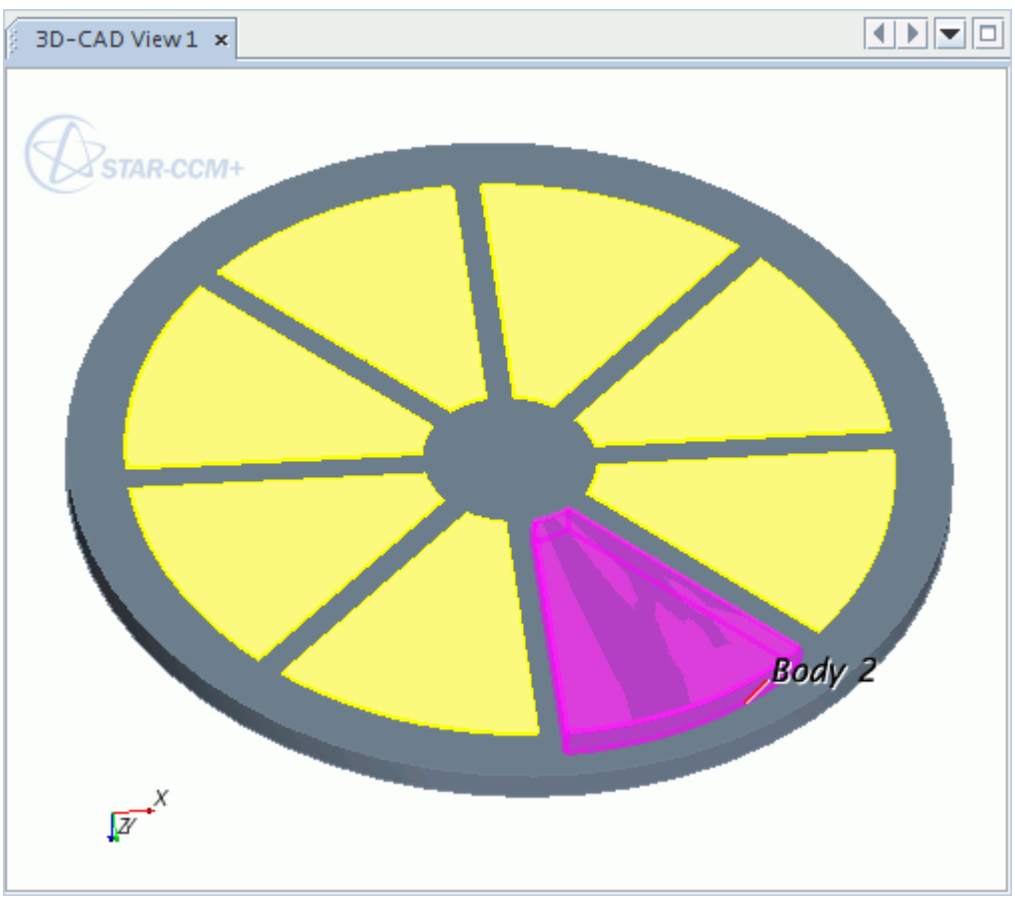
Axis of Rotation	Determines the axis about which the circular pattern is created.		
	Axis Type	Specifies whether to define the direction of the pattern using a specified vector or using a reference axis.	
		Specified	Creates the pattern about a custom axis, whose vector components are specified in the *Axis Direction* property.
		Reference Axis	Creates the pattern about a reference axis. Select an existing reference axis from the *3D-CAD View* scene. See Working with Reference Geometry in 3D-CAD.
	Coordinate System Source	Specifies the coordinate system that is used in the operation. Custom coordinate systems that you specified in the simulation are listed here.	
		Imported	Uses the imported coordinate system for the operation.
		Reference Coordinate System	Uses a reference coordinate system for the operation. Select an existing reference coordinate system from the *3D-CAD View* scene. See Working with Reference Geometry in 3D-CAD.
	Axis Direction	Specifies the direction of the axis of revolution. Enter the `[X, Y, Z]` components of the direction vector using the global coordinate system.	
	Axis Position	Specifies the origin of the axis of revolution. Enter the `[X, Y, Z]` coordinates of the origin.	
	Coordinate System	Specifies the coordinate system that is used in the operation. Custom coordinate systems that you specified in the simulation are listed here.	
Number of Instances	Specifies how many the seed body is reproduced around the circle. The value that you enter must be a whole number. If you enter a decimal value, it is rounded down to a whole number.		
Evenly Distributed	Activate this option to space the bodies in the pattern apart equally.		
Angular Increment	Specifies the angle difference between each body in the feature, relative to the axis of revolution and the seed body.		
Skip	Allows you to ignore specified bodies in the feature creation. To skip a body, select the body position from the drop-down menu and activate this option. To skip multiple bodies, select another body from the drop-down menu and activate the option for each body.		
Body Interaction	Specifies how the pattern interacts with other bodies adjacent to it, as defined by **Interacting Bodies**. The bodies must coincide or overlap for these options to work.		
	None	Creates new bodies from the circular pattern only. This option does not affect any existing bodies.	
	Merge	Merges all circular pattern bodies into the existing body. No new bodies are created from the pattern.	

	Imprint	Creates new bodies from the pattern. The edges and vertices are imprinted on the affected faces of each body.
Bodies to Interact	Specifies the bodies which interact with the pattern.	
	All	The pattern interacts with all contacting bodies.
	Selected	The pattern interacts with selected bodies only. Click inside the *Interacting Bodies* group box and select the bodies to interact with from the *3D-CAD View* scene or from the **Bodies** node in the 3D-CAD feature tree.
Bodies to Cut	Specifies the bodies that the pattern cuts.	
	All	The pattern cuts all bodies.
	Selected	The pattern cuts only selected bodies. Click inside the *Interacting Bodies* group box and select the bodies to cut from the *3D-CAD View* scene or from the **Bodies** node in the 3D-CAD feature tree.

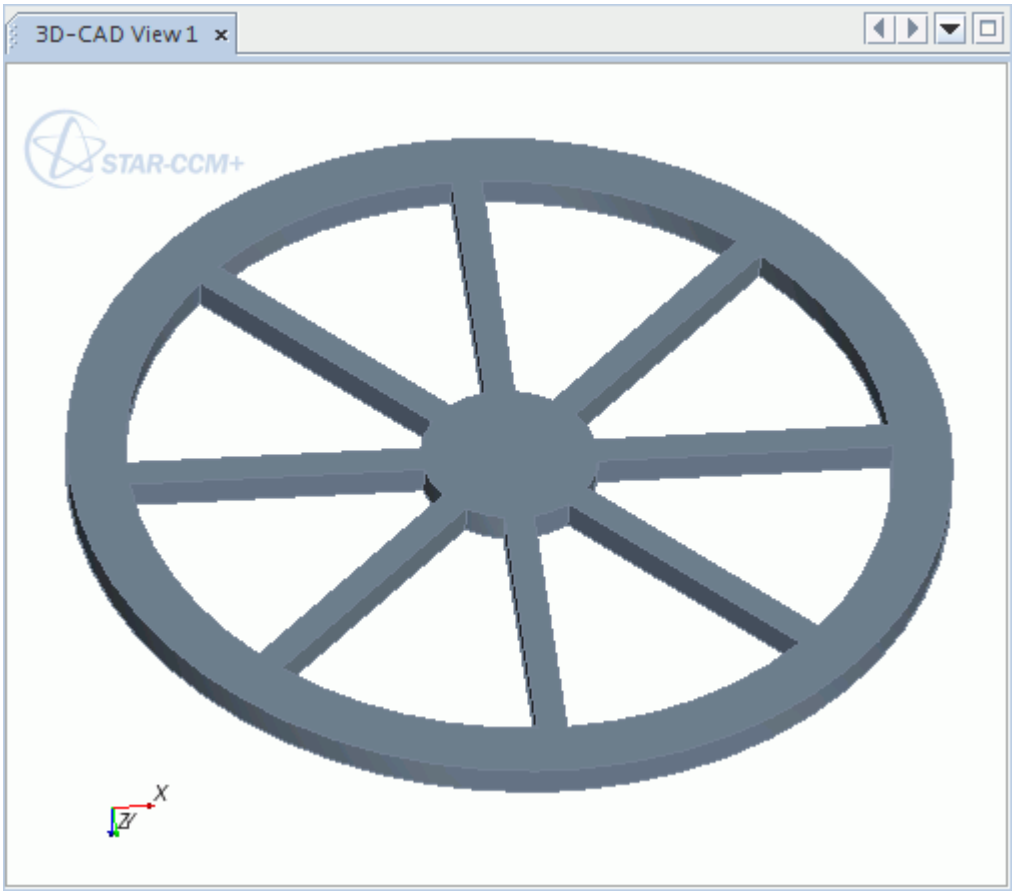
Creating a Circular Pattern Cut

Circular pattern cuts behave in the same way as circular patterns, except that the seed body defines a pattern that removes material from existing bodies.

You require at least two bodies to use a circular pattern cut.



To create a circular pattern cut, see Creating a Circular Pattern. An example of a circular pattern cut is shown below.



Working with Fillets and Chamfers

Fillets and chamfers are used to blend two faces together, either by creating a smooth curve between them, or by cutting off the corner to leave a flat surface.

These features are applied to edges and faces, which can only be selected through the *3D-CAD View* window. These features are described in the following sections:

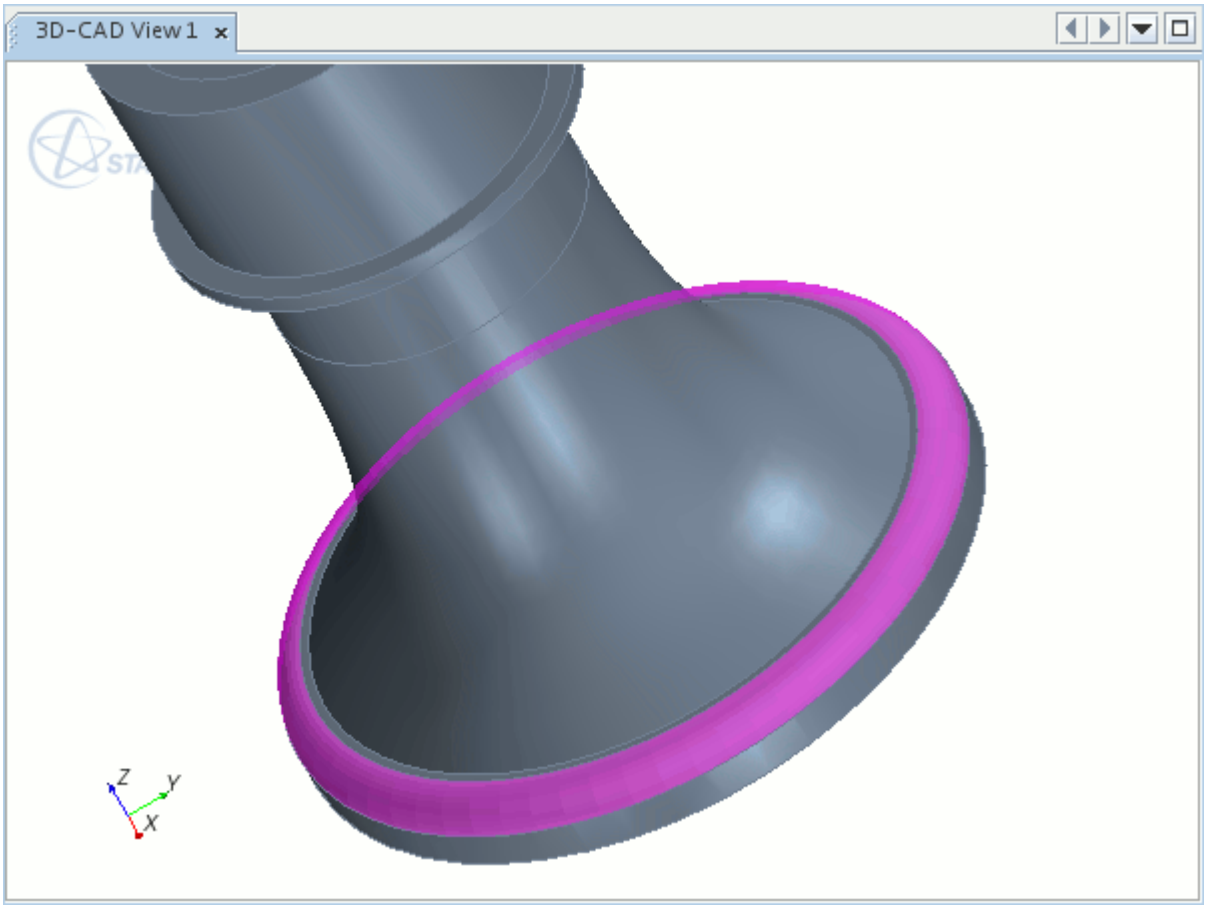
Contents:

What Is a Fillet?
Applying a Fillet Feature
What Is a Chamfer?
Applying a Chamfer Feature

What Is a Fillet?

A fillet is a feature that rounds-off the edge, or corner, of a body by replacing it with a smooth curve of constant radius.

Fillets can be applied to interior and exterior corners.

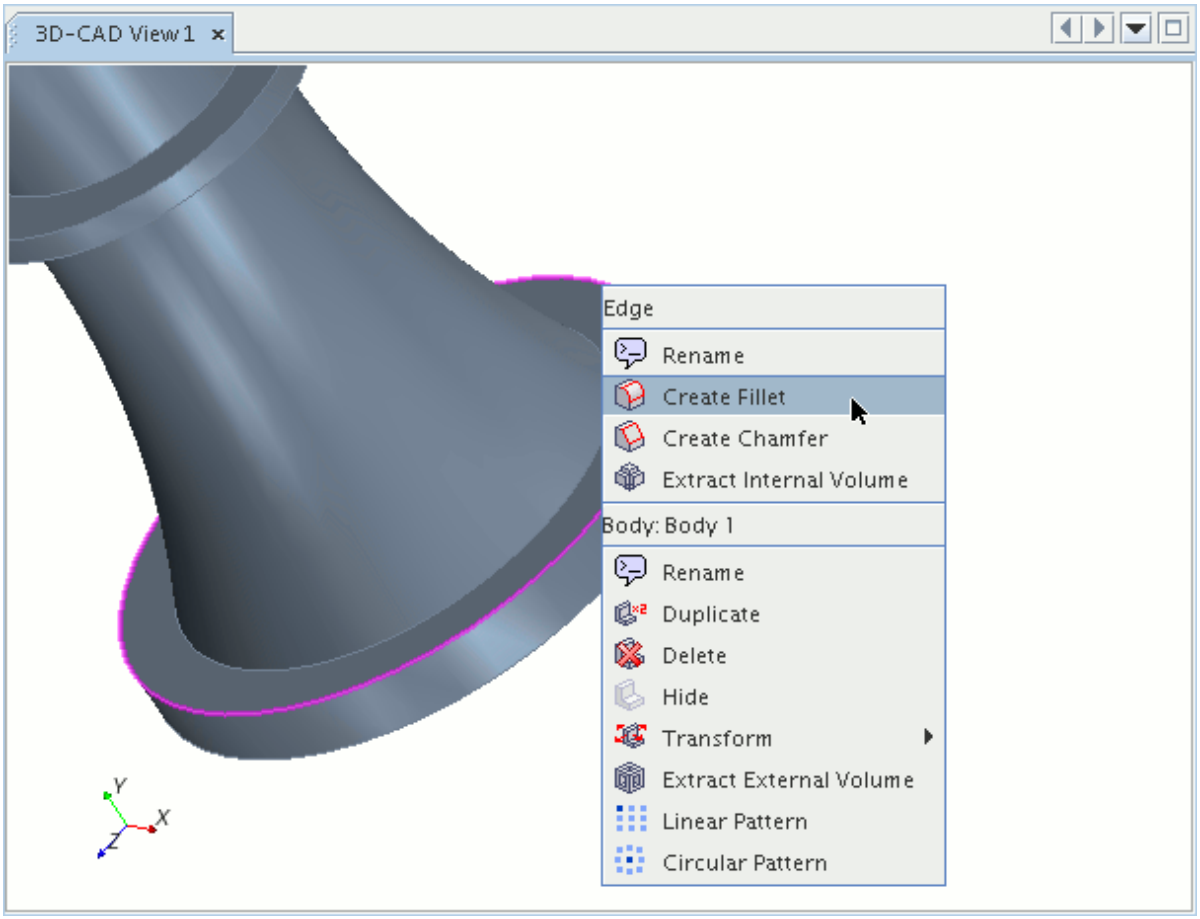


Applying a fillet successfully depends on what edges or faces you select and the size of the fillet in relation to the model. If you are having difficulty in applying a fillet, try decreasing the size of the radius or selecting fewer edges/faces. In some cases, a fillet that is applied to a particular edge can extend automatically to include adjacent edges.

Applying a Fillet Feature

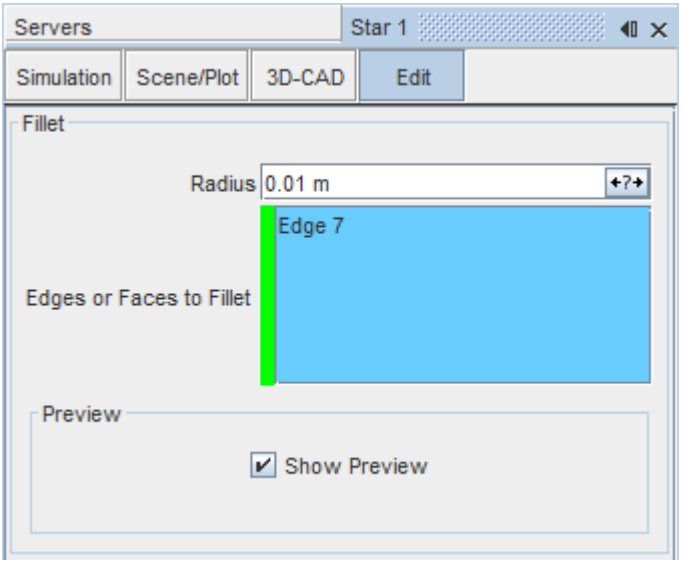
To apply a fillet:

1. Either:
    - Right-click on an edge or face in the *3D-CAD View* window and select **Create Fillet**.



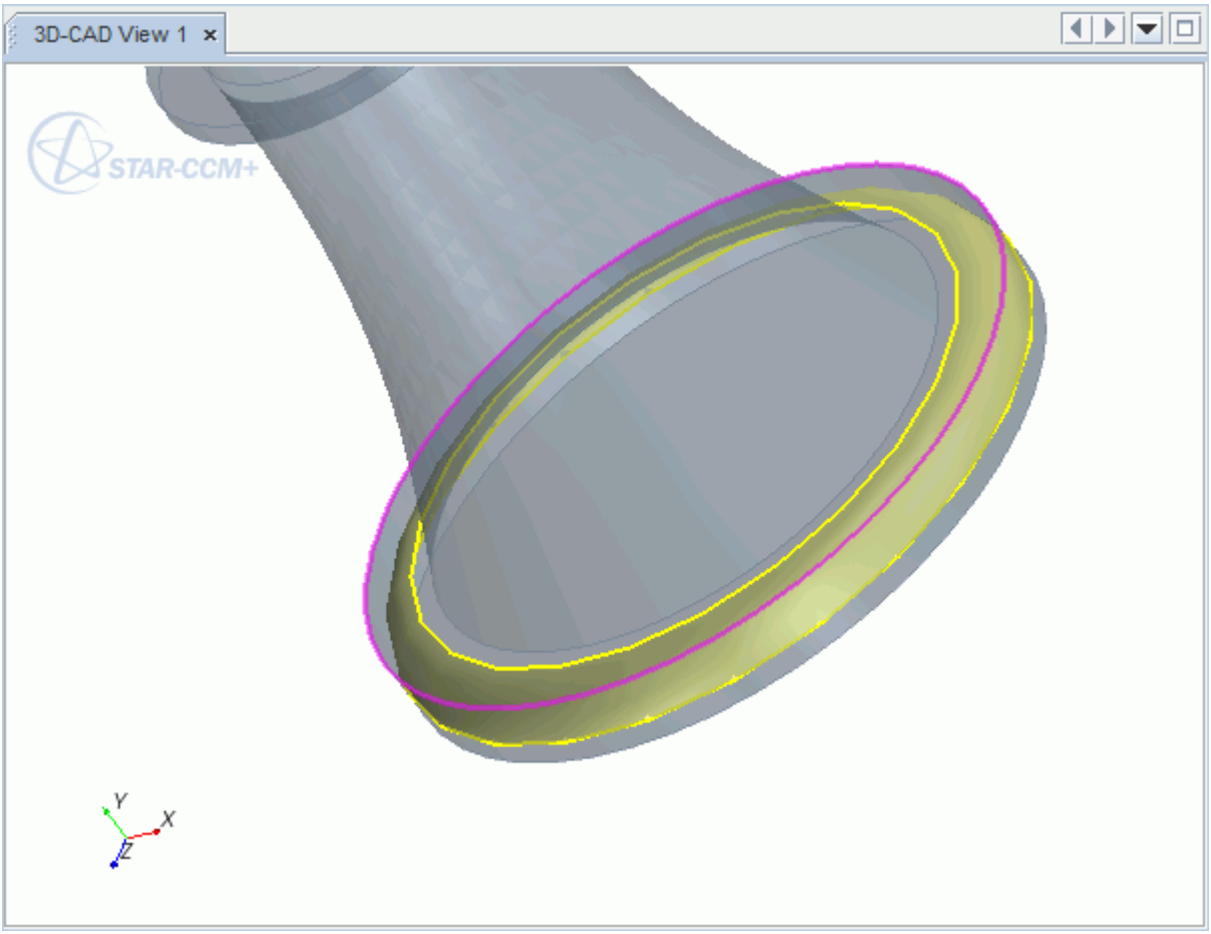
- Click (**Create Fillet**) in the 3D-CAD toolbar.

The *Fillet* panel appears.



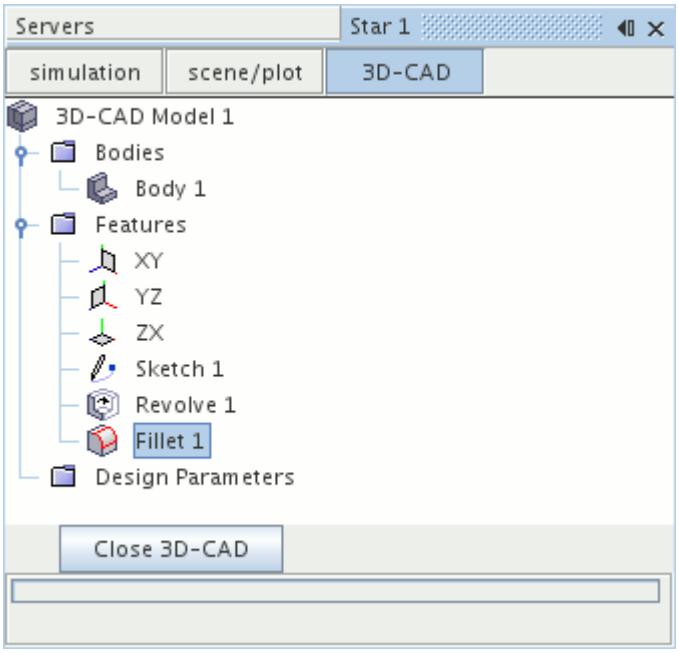
2. Select any additional faces and edges for the fillet. To make the edge or face selection process easier, use the Selection Filter. (See Filtering the Selections.) You can select multiple edges or faces at once by holding down the <`Ctrl`> key.
3. In the **Radius** textbox, enter the radius of the fillet. If you selected a face, the fillet is applied to all edges from the face.
4. If necessary, expose the radius as a design parameter by clicking the (**Expose Parameter**) button.

   The number of edges and faces that are selected is shown next to the Number of Selected Edges and Number of Selected Faces fields respectively. A preview of the fillet is shown in the *3D-CAD View* scene. The fillet is shown in yellow, while the edges and faces that are used to create the fillet are shown in purple.



5. Click **OK** to apply the fillet to the selected edges.

A **Fillet** node is added to the feature tree.



To change which edges or faces the feature is applied to, or to update the size of the fillet, right-click on the **Fillet** node and select **Edit...**

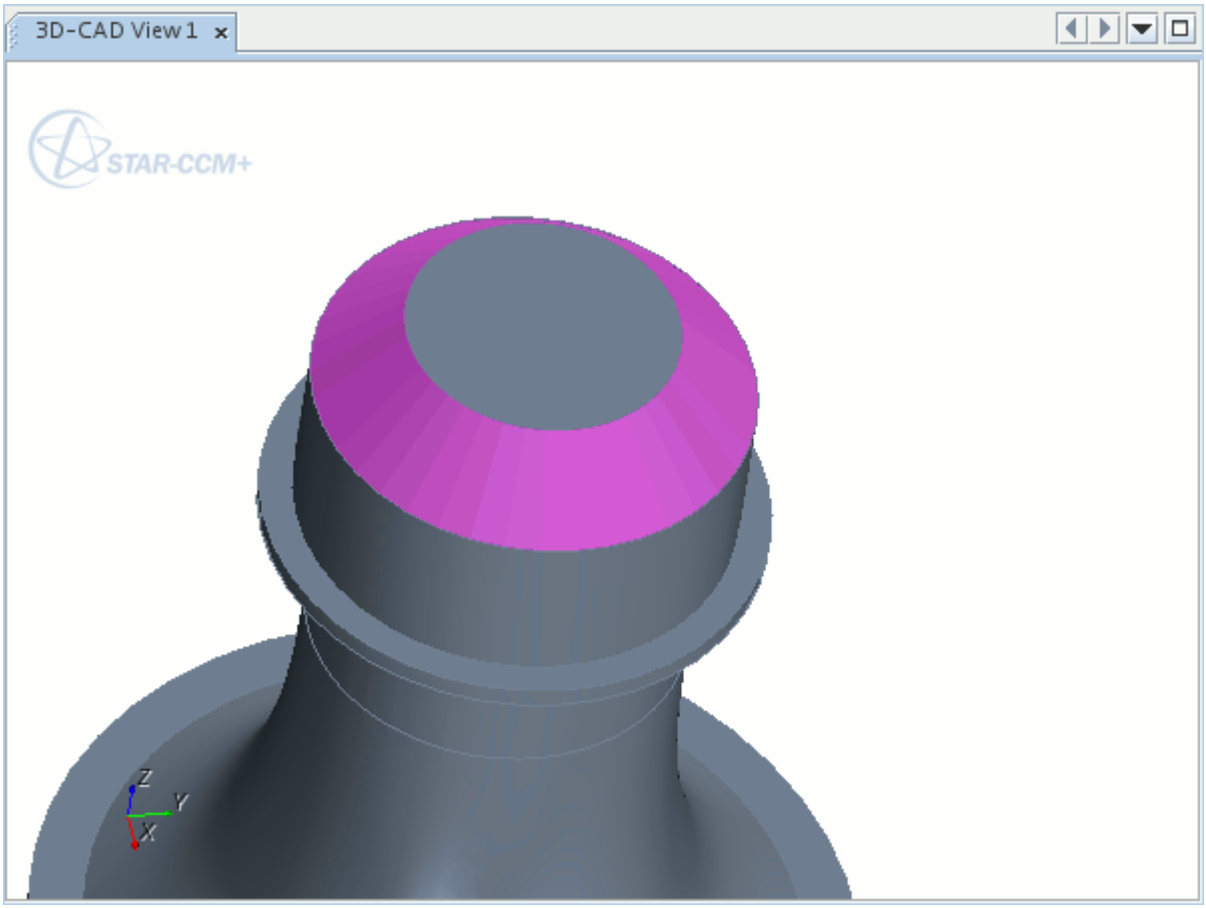
See Also:

What Is a Design Parameter?

What Is a Chamfer?

A chamfer is a feature that makes a straight cut across the corner of a body, resulting in a flat surface.

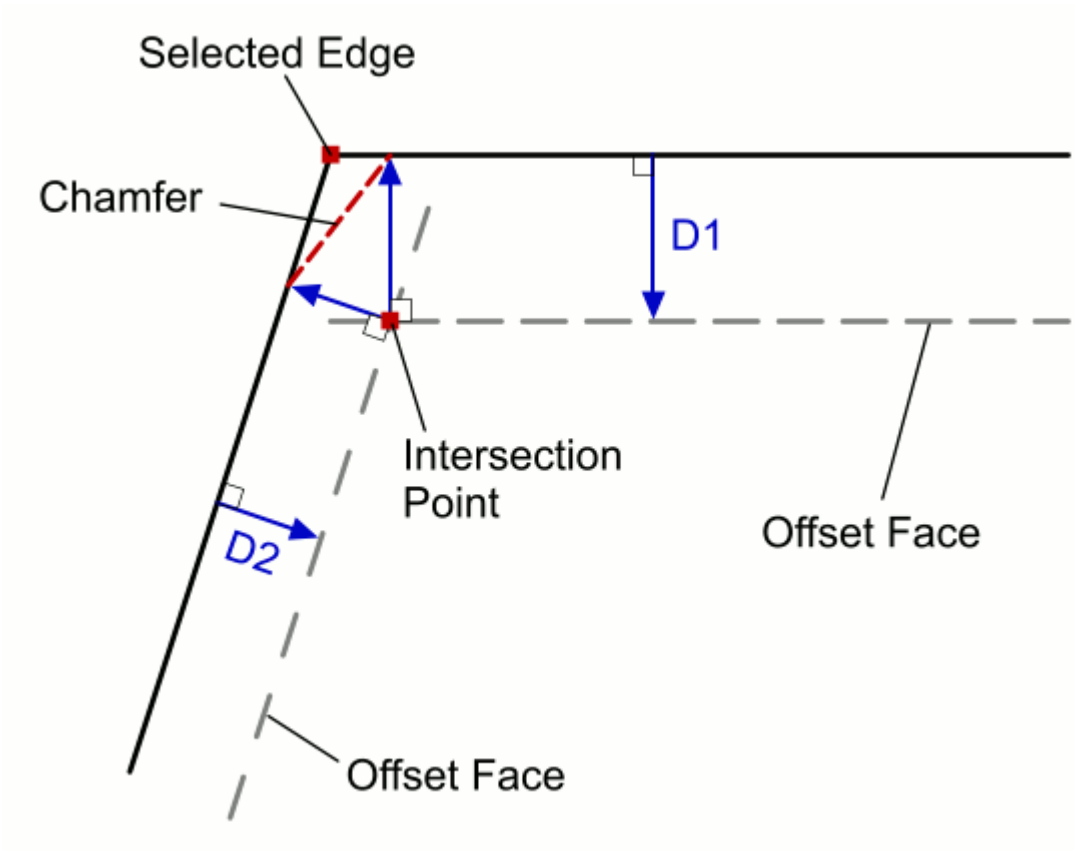
The size of the chamfer is defined as two values which are the distances from the selected edge into each of the faces. Chamfers can be applied to interior and exterior corners. When applied to interior corners, chamfers add material to the body.



When you select an edge, the chamfer tool:

1. Offsets the faces that are adjacent to the selected edge with the specified distances, *Distance 1* and *Distance 2*.
2. Finds the point where the two offset faces intersect.
3. Projects this point back to the original faces, following the path that is normal to the original faces.
4. Creates the chamfer through these projected points.

An illustration of how the chamfer tool works is shown below.



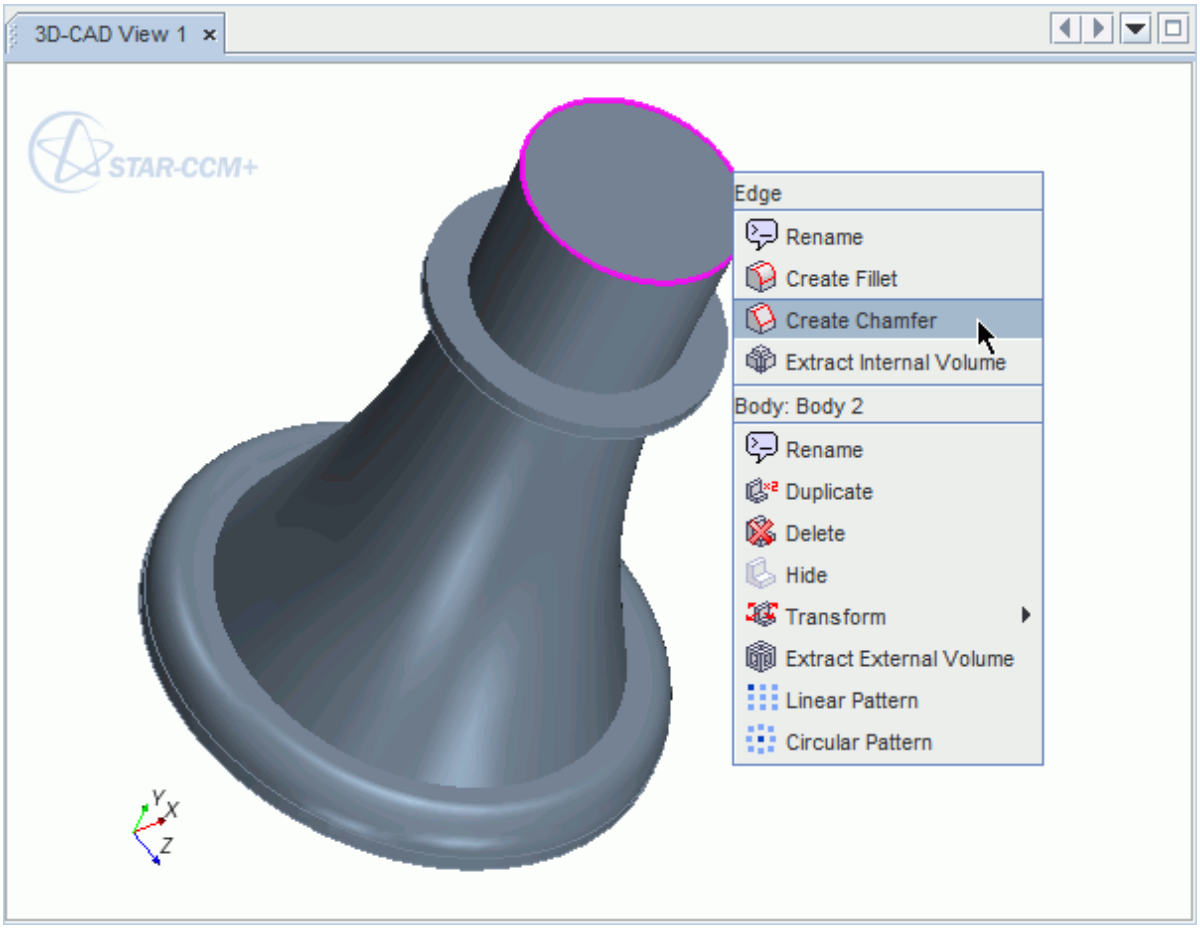
You can also apply a chamfer to the edges of a selected face. The chamfer tool creates the chamfer for each edge following the same principles as described above.

Applying a chamfer successfully depends on which edges or faces you select, and the size of the chamfer in relation to the model. If you are having difficulty in applying a chamfer, try decreasing the dimensions of the chamfer or selecting fewer edges/faces. In order to solve the model geometry, a chamfer that is applied to a particular entity can extend automatically to include adjacent edges.

Applying a Chamfer Feature

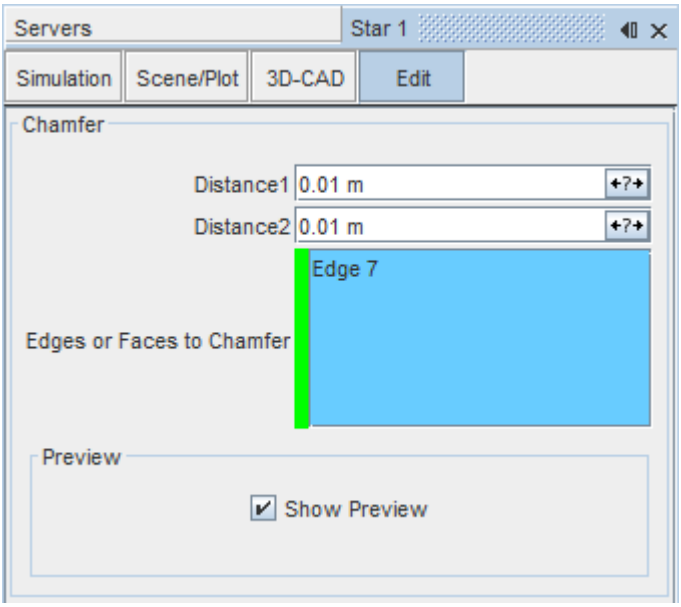
To apply a chamfer:

1. Either:
   - Right-click on an edge or face in the *3D-CAD View* scene and select **Create Chamfer**.



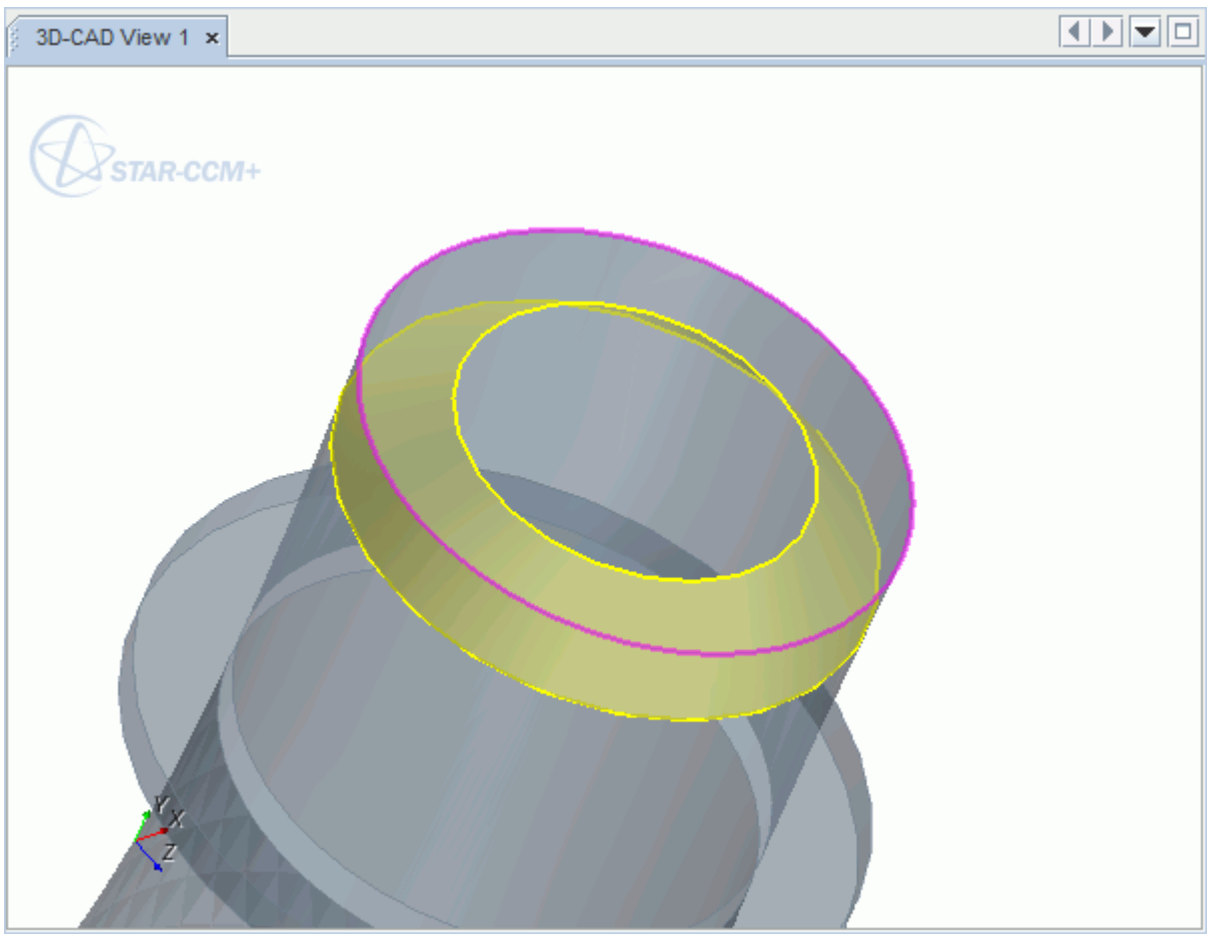
- Click (**Create Chamfer**) in the 3D-CAD toolbar.

The *Chamfer* panel appears.



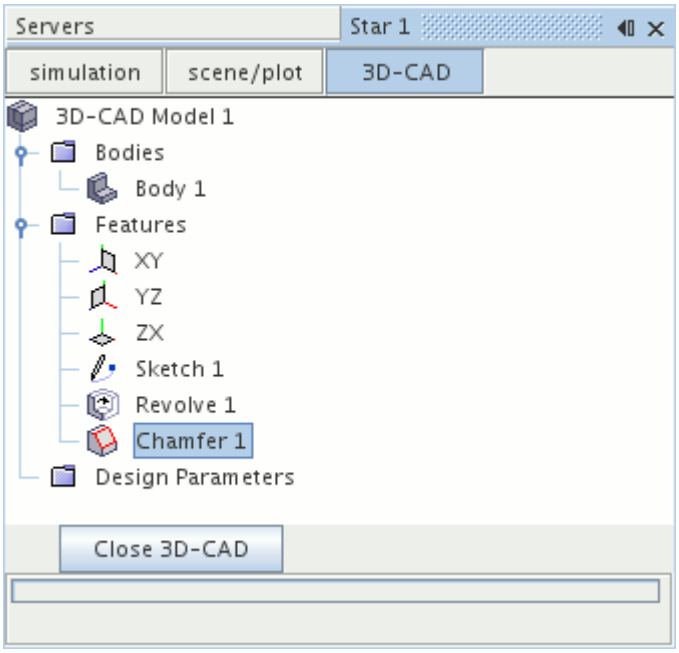
2. Select any additional edges and faces in the *3D-CAD View* window. To make the selection of faces or edges more easily, use the Selection Filter. (See Filtering the Selections.) You can select multiple edges/faces at once by holding down the <`Ctrl`> key.
3. In the **Distance** textboxes, enter the chamfer size. These values determine the distance that the chamfer extends from the selected edge, or edges. It is not possible to determine which face each distance is applied to.
4. If necessary, you can expose each distance as design parameter by clicking the (**Expose Parameter**) button.

   The number of edges and faces that have been selected is shown next to the Number of Selected Edges and Number of Selected Faces fields respectively. A preview of the chamfer feature is displayed in the *3D-CAD View* scene, and allows you to visualize the chamfer feature and switch the two distances. The chamfer is shown in yellow, while the edges and faces that are used to create the chamfer are shown in purple.



5. Click **OK** to apply the chamfer to the selected edges.

A **Chamfer** node is added to the feature tree.



To modify the dimensions of the chamfer, right-click on the **Chamfer** node and select **Edit...**

See Also:

What Is a Design Parameter?

Working with Design Parameters

Design parameters are used in 3D-CAD to specify characteristic dimensions of a model.

They can be based on a number of different features including a dimension in a sketch, the depth of an extrusion or the angle of a revolved feature. They are stored under the **Design Parameters** manager node for the model, which can also be accessed from the main simulation tree. Since design parameters can be edited from outside 3D-CAD mode, they can be used to control the size of various features in the model.

This section covers the following:

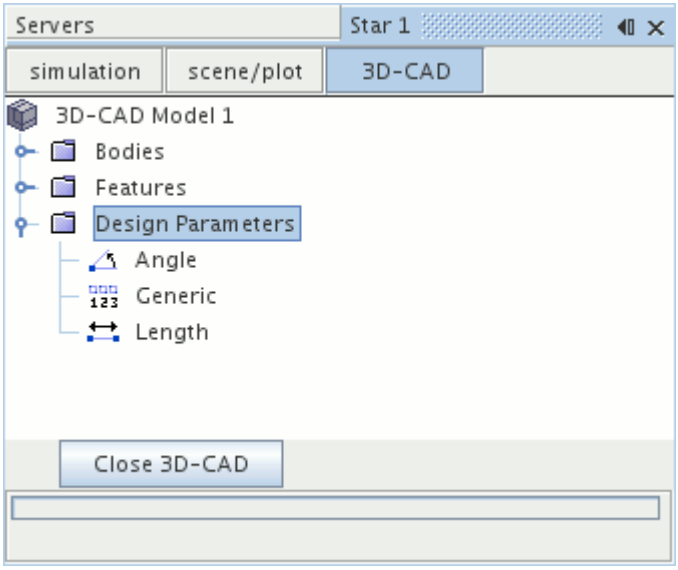
Contents:

What Is a Design Parameter?
Creating a New Design Parameter
Exposing a Parameter
Using Design Parameters in an Expression
Changing the Value of a Design Parameter
Deleting a Design Parameter

What Is a Design Parameter?

A design parameter is a reference variable that you can use to define model dimensions and feature sizes.

Each design parameter has a name and a value. When you enter this name into a dimension dialog, for example, the dimension is set to the corresponding value of the design parameter. Design parameters are versatile, and you can use them to form larger expressions. Another benefit of using design parameters is that they allow you to change feature values from outside of 3D-CAD.



There are three types of design parameter:

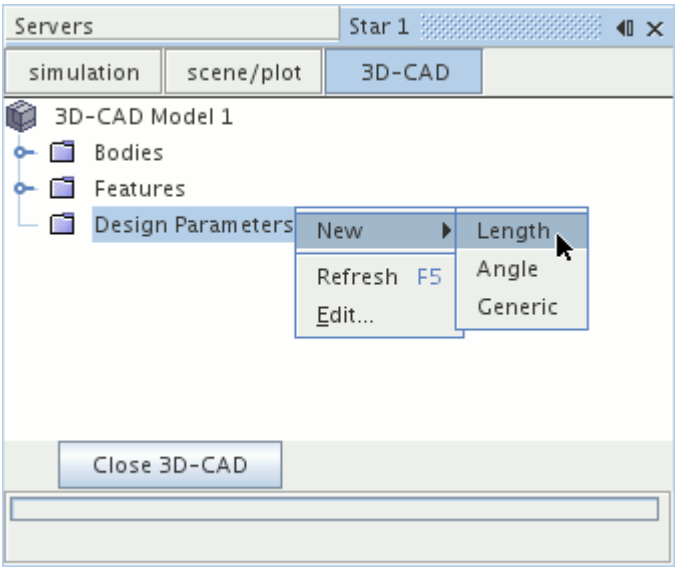
- Angle
- Length
- Generic

You can explicitly create design parameter, or you can expose an existing dimension (Angle and Length design parameters only). Angle design parameters are defined in either degrees (deg) or radians (radian). Length design parameters are defined in one of the standard units of length.

Creating a New Design Parameter

To manually create a design parameter:

- Right-click on the **Design Parameters** node and select **New** followed by one of the available options:



- **Angle**
- **Length**
- **Generic**

Note: Selecting either **Angle** or **Length** allows you to choose which units define the design parameter. Selecting **Generic** lets you enter a dimensionless quantity.

A new design parameter node, of the selected type, is added to the object tree under the **Design Parameters** manager node. Design parameters that you create in this way have the default value of `0.0`. To set the new value:

1. Select the design parameter node.
2. In the *Properties* window, enter the parameter value in the *Value* property.

Note: You can set the units of an **Angle** or **Length** design parameter by typing them after the quantity, for example `10 cm`. Alternatively, click the ... (**Custom Editor**) button next to the *Value* property, and select an option from the Units drop-down menu.

Design parameters that are created using this method are referred to as User-defined. You can also create design parameters by exposing a parameter. These types of parameter are referred to as Feature-exported.

Exposing a Parameter

You can expose an existing dimension as a design parameter and rename this parameter.

A corresponding node is placed in the **Design Parameters** manager node in the 3D-CAD feature tree. This node allows you to use the dimension as a variable when defining other model attributes. You can also edit the value of the dimension by changing the value of the associated design parameter. Use this technique if you want to control a model dimension without editing the corresponding feature directly, or if you want to define other variables in terms of the selected dimension.

There are two methods of exposing a design parameter:

- Exposing a dimension that defines a feature
- Exposing a dimension that defines a sketch entity

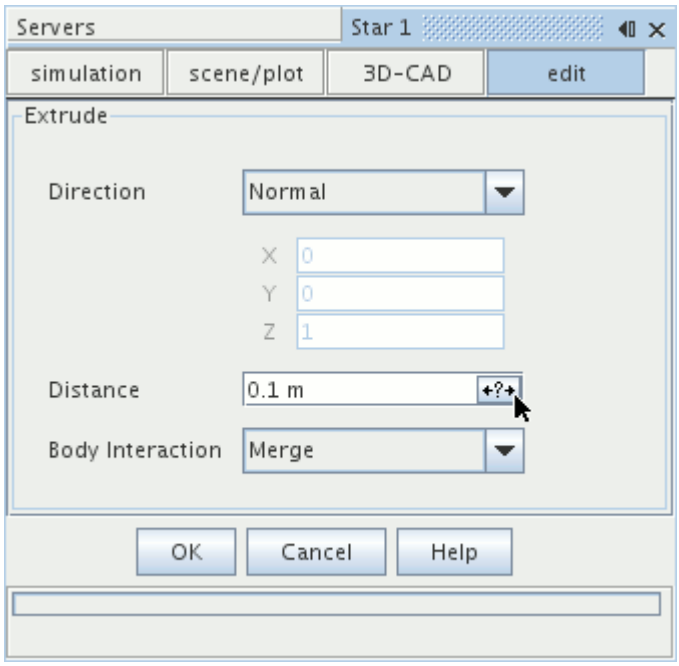
A design parameter that is created using this method is referred to as Feature-exported. You can also create design parameters explicitly using the **Design Parameters** manager node. These types of design parameters are referred to as User-defined.

When you delete a feature that contains an exposed dimension, the associated design parameter is not deleted. Instead, it is reclassified as User-defined.

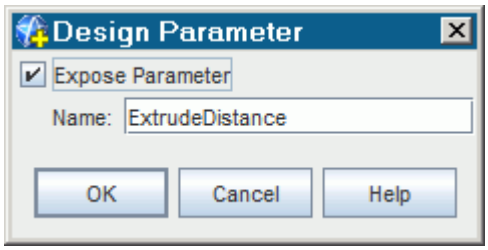
Exposing a Parameter from a Feature

To expose a feature dimension as a design parameter:

1. In the corresponding feature dialog, click the (**Expose Parameter**) button.

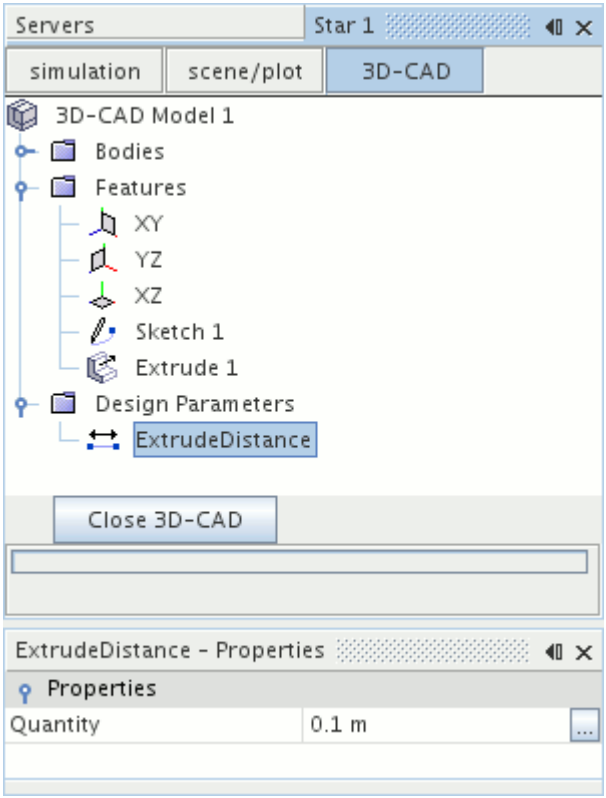


The *Design Parameter* dialog appears.



2. Activate **Expose Parameter** to confirm that you want to expose this dimension.
3. In the *Name* field, enter the name for the design parameter.
4. Click **OK**.

A design parameter node is added to the 3D-CAD model tree under the **Design Parameters** manager node.



Note: If you cancel the creation of the feature, the design parameter node still exists in the model tree. To reverse this process, conceal the parameter.

Examples of feature dimensions that you can expose as design parameters are:

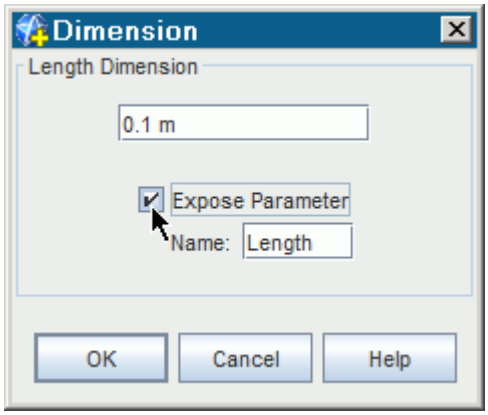
- The **Distance** of an extrusion or extruded cut feature
- The **Angle** of a revolved or revolved cut feature
- The **Radius** of a fillet feature
- The **Distances** of a chamfer feature

Exposing a Parameter from a Sketch Entity

You can expose a dimension that defines the size of a sketch entity as a design parameter. To expose this dimension:

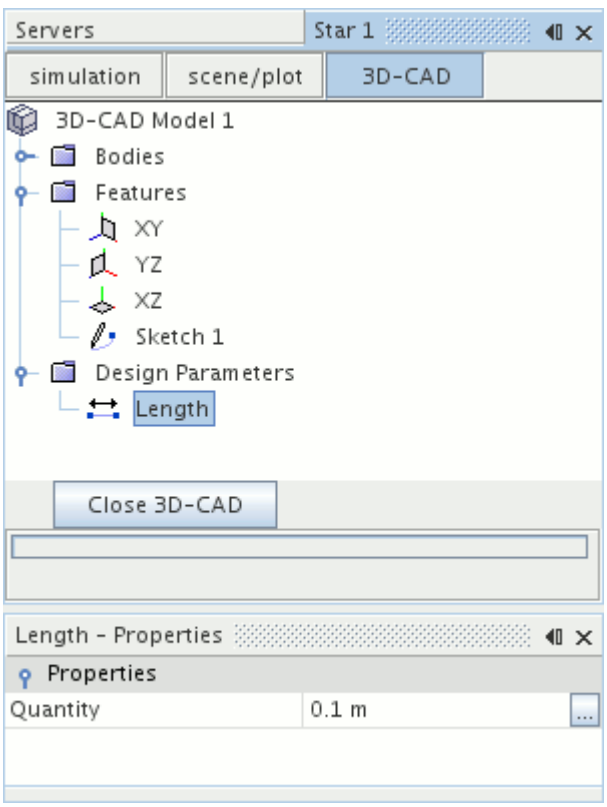
1. Applying a dimension to a sketch entity, or edit an existing dimension.

   The *Dimension* dialog appears.



2. Activate **Expose Parameter**.
3. In the **Name** field, enter the name of the design parameter.
4. Click **OK** to expose the parameter.

A design parameter node is added to the 3D-CAD model tree under the **Design Parameters** manager node.



Note: If you cancel the creation of the feature, the design parameter node still exists in the model tree. To reverse this process, conceal the parameter.

Contents:

Concealing a Design Parameter

See Also:

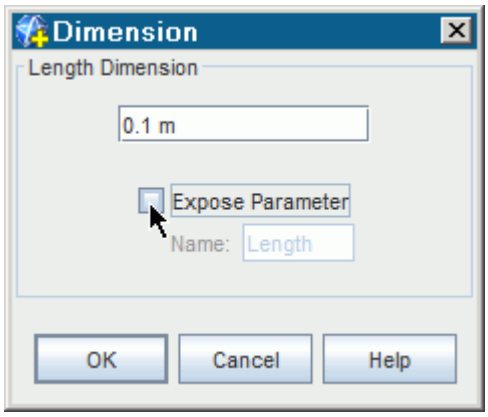
Creating Bodies From Sketches

Sketch Entity Reference

Concealing a Design Parameter

To remove design parameters from the 3D-CAD feature tree, conceal them. To conceal a design parameter:

1. Access the *Dimension* dialog or the *Parameter Name* dialog. See Exposing a Parameter.
2. Deactivate **Expose Parameter**.

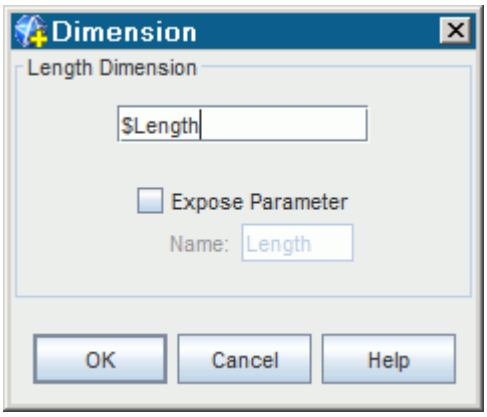


This action hides the design parameter and removes the design parameter node from the 3D-CAD feature tree. Any model dimensions that you specified using this design parameter are no longer satisfied, and errors may be reported for the affected features. To resolve the related errors, check the value of existing design parameters and dimensions.

You can also delete design parameters.

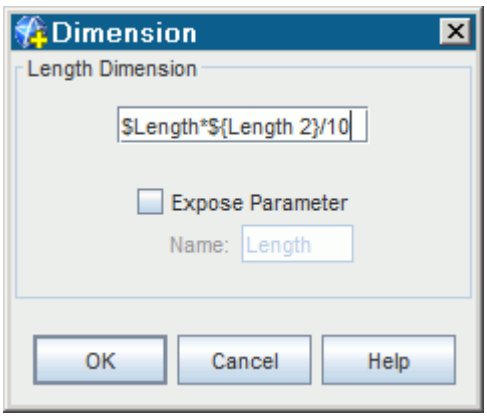
Using Design Parameters in an Expression

You can use design parameters to specify the value of model dimensions by typing $ followed by the parameter name into the dimension textbox. This format is the simplest form of expression.



Note: Design parameters are case-sensitive.

You can also use design parameters as variables in longer expressions by using operators `* / - +` and parentheses `( )`. Spaces are acceptable between terms and operators in the expression. If the name of the design parameter, however, contains any spaces, enclose it in `{ }`, for example `${Length 2}`.



3D-CAD assumes that design parameters are in SI units. Dimensions that are defined by expressions are displayed in SI units only. When conflicting types of design parameter are used in the same expression, for example Angle and Length, 3D-CAD interprets the values in the SI units that correspond to the type of dimension.

Changing the Value of a Design Parameter

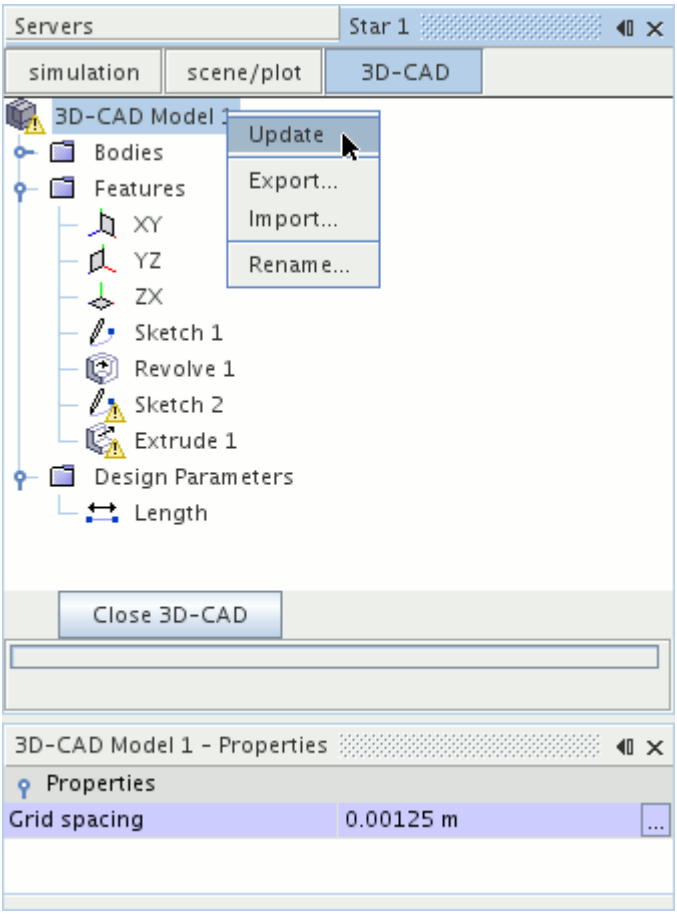
Changing the value of a design parameter modifies the size of any dependent features. This ability allows you to control the size of model attributes without having to edit specific features. By using this technique, you can also modify the 3D-CAD model from outside of 3D-CAD.

To change the value of a design parameter:

1. Under the **Design Parameters** node, select the design parameter that you want to modify.
2. In the *Properties* window, either:
   - Enter the new value for the parameter in the *Value* field.
   - Click ... next to the value to bring up the *Value* dialog and enter the new value for the parameter.

     If the design parameter contains an expression, click ... in the *Value* dialog to bring up the expression editor and modify the expression using the editor. See Inserting Expressions Interactively.

   Modified features require an update for the changes to take effect. These features are indicated as an altered feature state.
3. Right-click the 3D-CAD model node and select **Update**.



The affected features are updated with the new parameter value.

See Also:

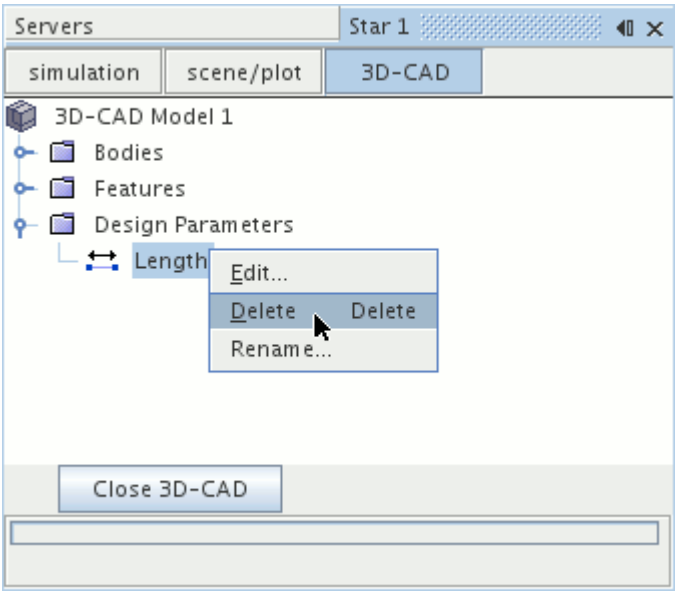
Modifying a 3D-CAD Model

Understanding Feature States

Deleting a Design Parameter

There are two ways that you can delete a design parameter:

- Select the relevant node under the **Design Parameters** manager node and press the <`Delete`> key.
- Alternatively, right-click the relevant node under the **Design Parameters** manager node and select **Delete**.



You can also remove an exposed parameter from the feature tree by concealing it.

If you used the deleted design parameter to define any dimensions in the model, 3D-CAD is unable to solve the model geometry for the affected features. This situation is indicated in the feature tree by a change in the feature states, and the 3D-CAD model node displays a change of state. To remedy this situation, update any dimensions that are defined using the deleted design parameter. Alternatively, create another design parameter and specify the same parameter name as the deleted design parameter.

See Also:

Understanding Feature States

Working with Volume Extraction

3D-CAD can automatically extract the volume of space surrounding a model or volume of space inside a model, and use it to form a new, separate solid body.

This process is often used for creating geometry that represents a fluid domain, which you can then import into a simulation to analyze internal or external flows. This chapter describes the methods that are used to extract the internal or external volume of a model:

Contents:

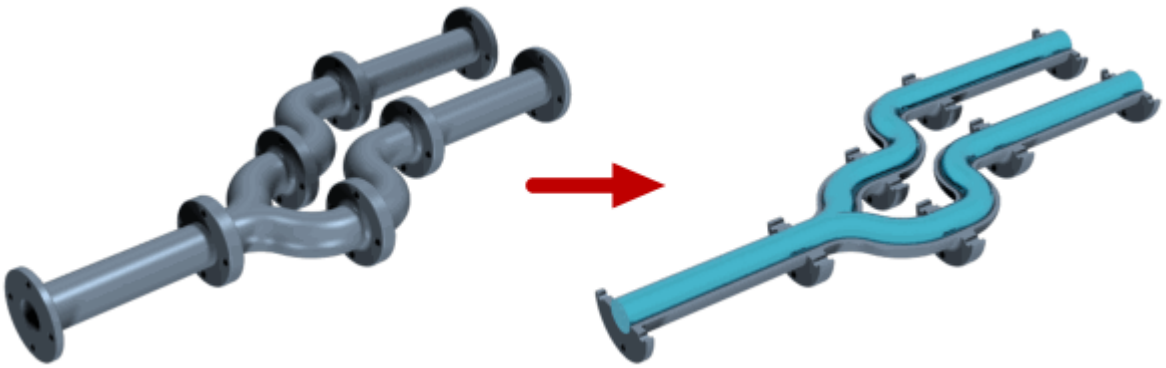
What Is Volume Extraction?
Internal Volume Extraction
External Volume Extraction

What Is Volume Extraction?

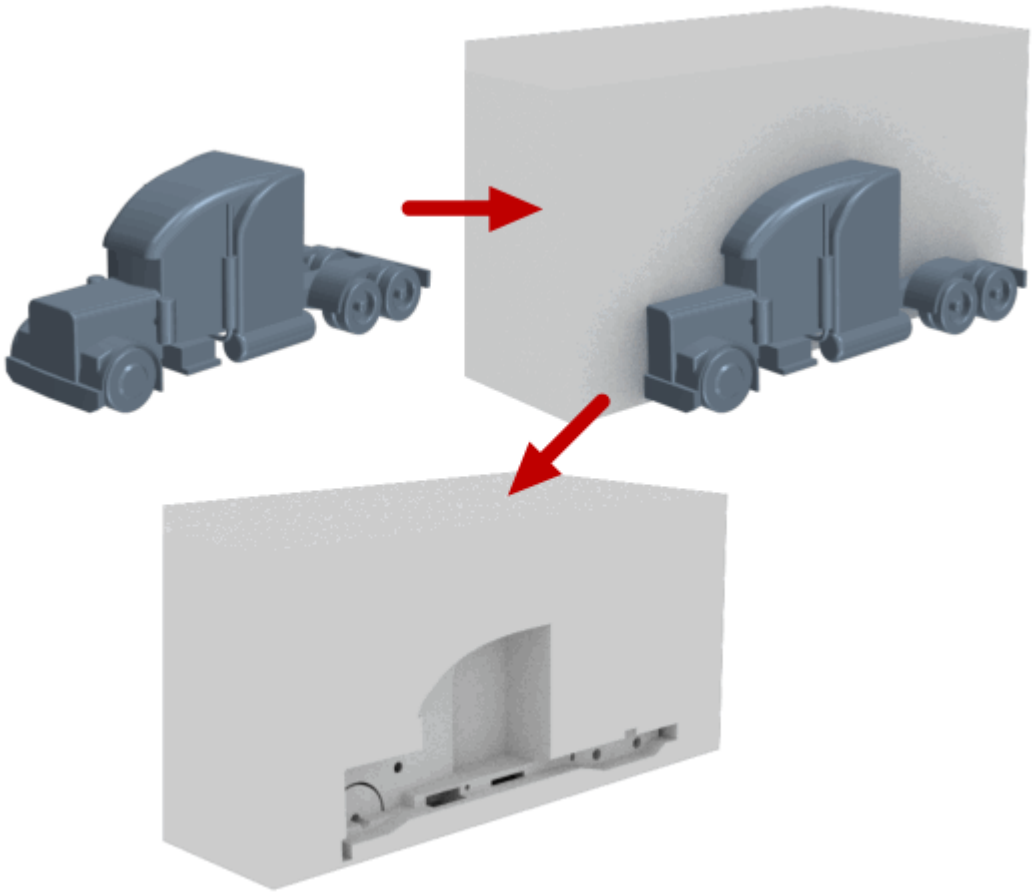
When you extract a volume, 3D-CAD finds a continuous surface within, or around, a solid model and uses it to create a new body.

The body is extended up to arbitrary limits, which are determined by whether the extraction is internal or external.

In internal volume extraction, the process can be thought of as capping all inflow and outflow openings in the model, and filling the internal void with a new solid body. If there are any other holes in the geometry, material can leak out, so the geometry must be closed, apart from any inlets and outlets. You identify the location of these holes during the feature creation process, and 3D-CAD caps them automatically. The model must meet certain criteria for the volume extraction process to be successful. See Internal Volume Extraction: Requirements.



In external volume extraction, you create a bounding body around the model and extract this model from the bounding body. In this process, any holes, hollows, and crevices in the geometry model are filled with the new solid body, in the same way that they are filled by a surrounding volume of gas or liquid. The bounding body can be created around a number of separate bodies, can intersect bodies, and can be any shape. This process is useful for creating a fluid domain for the analysis of external flows (for example, flows around a vehicle).



Internal Volume Extraction

In internal volume extraction, 3D-CAD caps holes in the model geometry, and fills the void inside the model with a new solid body.

The holes are often entry and exit points for a fluid, such as the start and end of a pipe. You identify these openings by selecting faces and/or edges. When you select a face, 3D-CAD analyzes it for the presence of any holes. Selecting edges specifies the exact location of a hole. If a solid body can be created by capping some or all of the holes that are found, the extraction process is successful. The extraction process fails if material can leak out of the model through any other holes.

The internal volume can span multiple bodies. During the operation, all bodies are united. When you create a solid body using internal volume extraction, an imprint operation is carried out automatically so that the edges of the new body are imprinted on existing bodies, and vice versa. The internal volume extraction tool recognizes any bodies that intersect or are located within the internal volume, and molds the new solid body around them. If these bodies do not contribute to the flow, you can choose to exclude them from the extraction.

Requirements

The 3D-CAD model must meet the following criteria in order to carry out internal volume extraction:

- The edge loop of each hole must lie on a single face, or cross a maximum of two connected faces. If the loop crosses three or more faces, 3D-CAD does not recognize the hole.
- The edges of each hole must lie on the same plane. Objects inside a hole are allowed and can extend beyond the hole opening.
- The faces or edges that define the holes in the model are selected manually, and there can be any number of holes in the geometry.
- If the internal volume spans multiple bodies, these bodies must be able to unite successfully.
- Annular regions are supported.

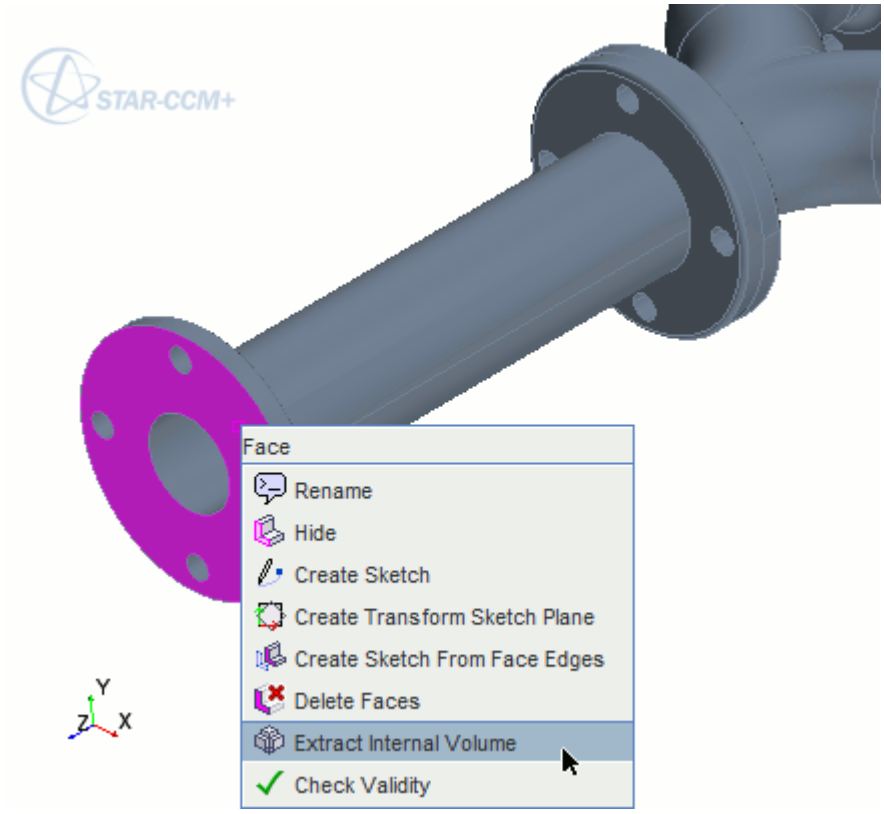
Contents:

Extracting the Internal Volume

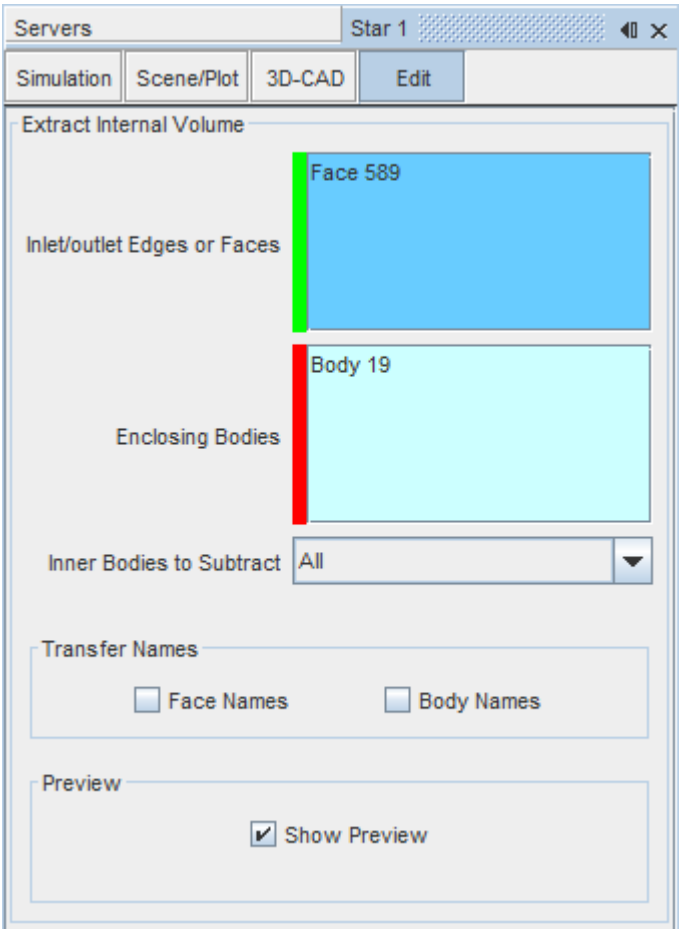
Extracting the Internal Volume

To carry out internal volume extraction:

1. Rotate the model in the *3D-CAD View* scene so that a face containing a hole is visible.
2. Right-click on the face, or the edge of the hole, and select **Extract Internal Volume**.



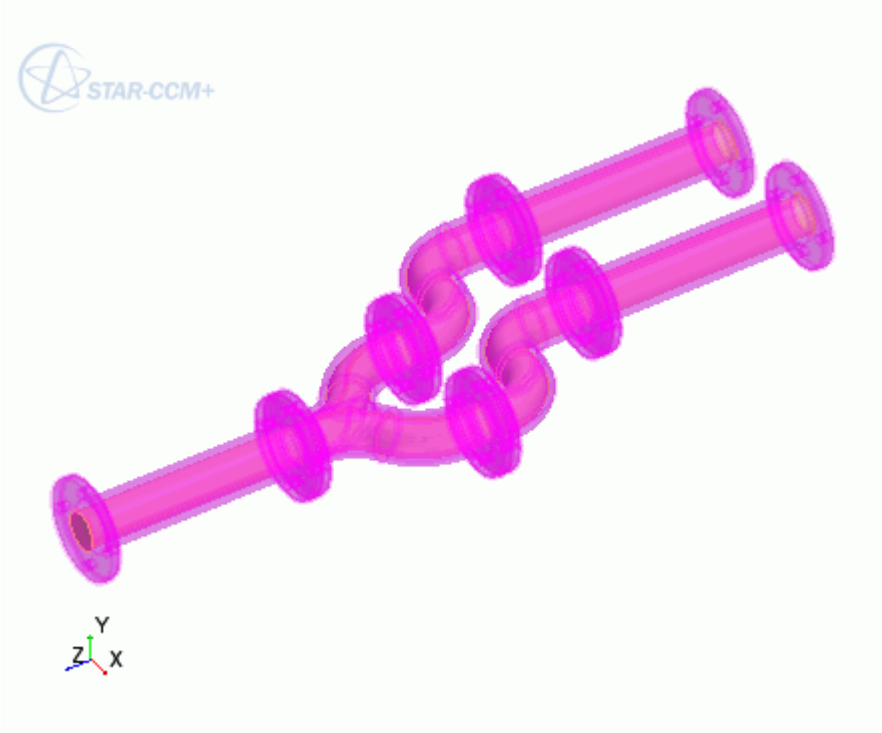
The *Extract Internal Volume* panel appears.



Here, you can see how many edges, faces, and bodies are currently selected.

3. In the *3D-CAD View* scene, hold down the <`Ctrl`> key and continue to select faces and edges to indicate the position of each hole. You can select any combination of edges and faces. To specify the location of a hole that crosses two connected faces, select the edges.
4. If the internal volume spans multiple bodies, click inside the *Enclosing Bodies* group box and select the bodies from the *3D-CAD View* scene or from the **Bodies** node in the 3D-CAD feature tree to select all bodies that are located between the outermost bodies of the volume.

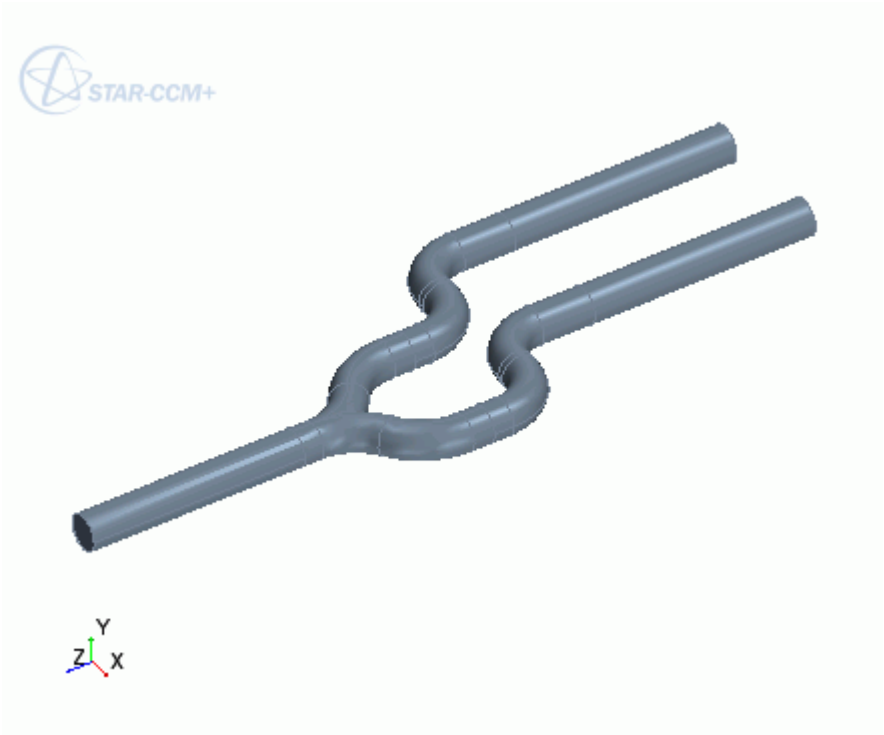
   3D-CAD attempts to fill the internal space with a new solid body, placing caps over any holes that are identified during selection. This action results in the creation of a new body, which is previewed in yellow in the *3D-CAD view* scene.



5. If the empty space enclosed by the original geometry contains additional bodies, such as external bodies that partially submerge into the internal volume, these bodies are subtracted by default. If you want to subtract only specific bodies from the internal volume, set *Inner Bodies to Subtract* to **Selected**, click inside the *Inner Bodies* group box, and select those bodies from the from the *3D-CAD View* scene or from the **Bodies** node in the 3D-CAD feature tree.

   If you do not want to subtract any inner bodies from the internal fluid volume, set *Inner Bodies to Subtract* to **Selected** and select no bodies. This feature allows you to exclude small bodies that do not contribute to the flow.
6. Choose whether to transfer custom names from the external body to the extracted volume. Activate the **Face Names** option to transfer custom face names from the external bodies to the extracted volume. Activate the **Body Names** option to transfer the name of the external body to faces on the extracted volume.
7. Click **OK** to proceed with the volume extraction.

An **Extract Internal** feature node is added to the feature tree, and the new body is added to the **Bodies** manager node. To visualize the new internal body, hide the original bodies so that you are only left with the new body.



To edit the edges and faces that define the feature:

- Right-click on the **Extract Internal** node and select **Edit**.

Troubleshooting

If you experience difficulty in extracting the internal volume of a model, try the following steps:

- Check that you selected all holes in the model geometry, as described above.
- Modify the geometry so that holes are on planar faces.
- Alter the model so that each hole is on a single face.
- If the internal volume spans multiple bodies, make sure that all bodies are selected and that it is possible to unite all the bodies together.
- If the geometry contains multiple holes on the same face and you cannot extract the desired hole, select the edge of the hole that you want to extract, rather than the face.

External Volume Extraction

External aerodynamics simulations require a fluid domain surrounding the object of interest. You can create this fluid domain as a solid body in 3D-CAD and extract the required external volume, leaving a cavity of the geometry that you want to analyze.

When the model itself contains a fully enclosed internal void, the external volume extraction feature also creates a solid body within this void.

Contents:

Extracting the External Volume
Extract External Volume Panel

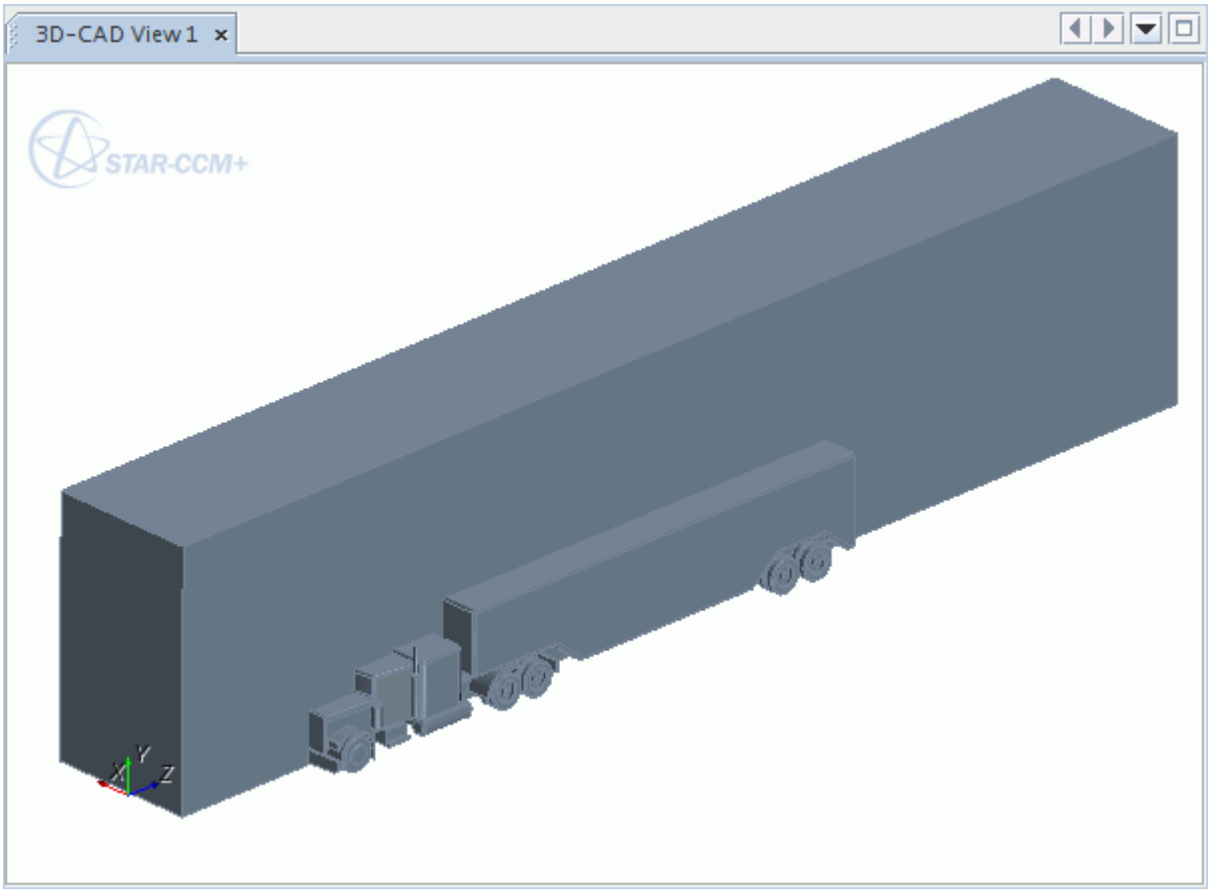
Extracting the External Volume

There are two steps to extracting the external volume. The first step is to create a separate body around the model which defines the extents of the fluid domain. This body can be any size and shape, and can intersect the model geometry. Typically you define this body by creating a sketch, then extruding or revolving this sketch. Other methods can also be used such as duplicating an existing body, or importing a body.

To create the fluid domain body in 3D-CAD:

1. Create a sketch. See Creating a Sketch.
2. Right-click on the **Sketch** feature node and select **Create Extrude** or **Create Revolve**, depending on the type of fluid domain you want to create.
3. In the feature creation panel for the extruded or revolved feature, make sure that **Body Interaction** is set to **None**. You do not want the new body to merge with the existing bodies.

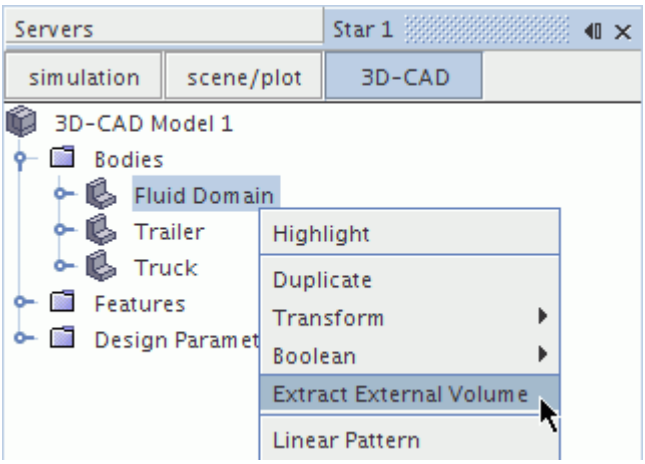
In the example below, the fluid domain surrounding a truck and trailer is extracted. The size of the fluid domain is purposely small for this example. As the case is symmetric, only half of the truck (and fluid domain) are required to model the case accurately .



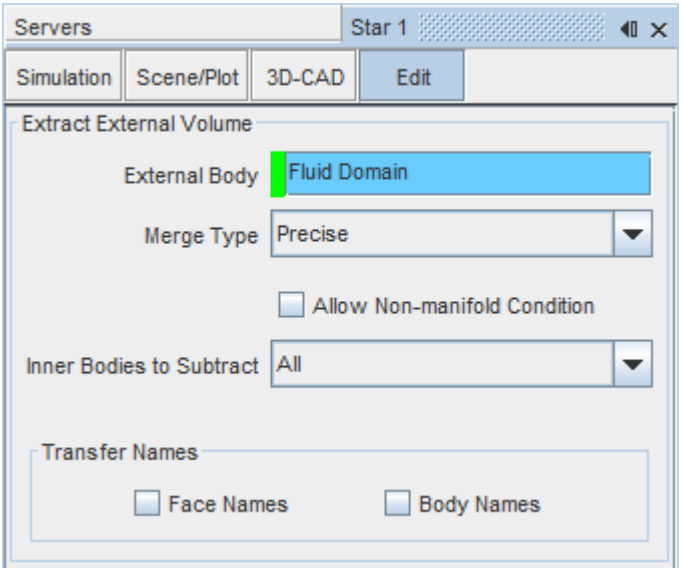
This new fluid domain body is added to the feature tree. It is recommended you rename bodies to keep track of them.

To extract the external volume:

1. Right-click on the fluid domain body under the **Bodies** manager node and select **Extract External Volume**.



The *Extract External Volume* panel appears.



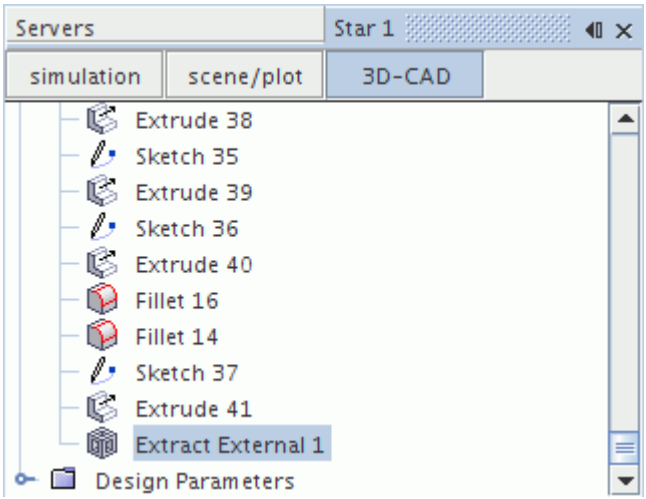
2. Set the extraction properties. See Extract External Volume Panel.

Note: You can modify the external body selection by clicking inside the *External Body* field and then selecting the desired body from the *3D-CAD View* scene or from the **Bodies** node in the 3D-CAD feature tree.

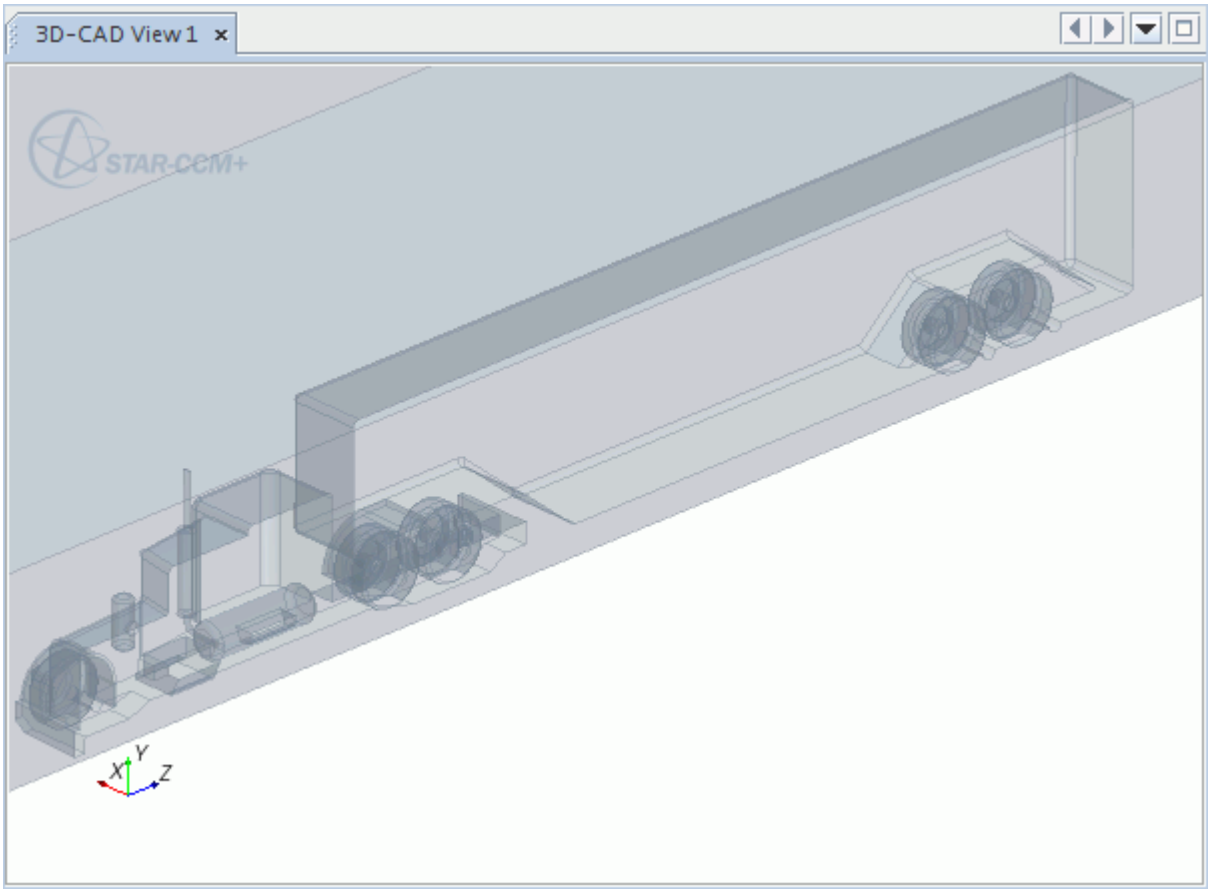
3. Click **OK** to perform the volume extraction operation.

3D-CAD creates a duplicate of the fluid domain body, unites the other bodies (the truck and trailer), and subtracts these bodies from the duplicate of the fluid domain. The duplicate body ends up with a cavity where the truck and trailer are positioned.

A new **Extract External** feature node is added to the feature tree. If the body that was used in the extraction process is no longer required, you can delete it.



In the following *3D-CAD View* scene, the original fluid domain, truck, and trailer are hidden. Scene transparency is also activated.



See Also:

Duplicating a Body

Importing Geometry into 3D-CAD

Extract External Volume Panel

Use the following properties to define the external volume extraction.

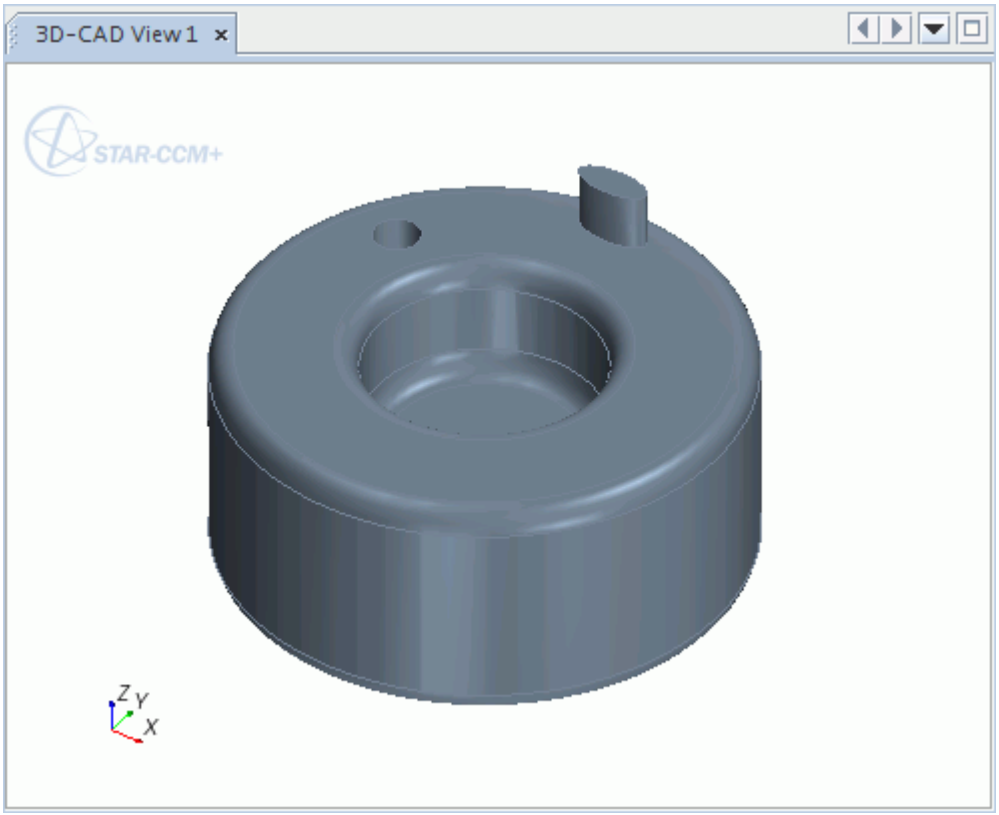
Select External Body	Click to select the external body with the **Select From Scene** tool.	
Merge Type	Defines the tolerance method to use in the operation.	
	Precise	Uses the default internal model precision of `1.0E-8 m` as the tolerance. This tolerance setting requires that faces of separate bodies be within 1.0E-8 m of each other. This option takes a minimum amount of computational time and works well for all coincident faces and surfaces.
	Tolerant Edges	Reduces the model precision by using a higher tolerance value that you provide on all the edges in the model. This option causes any further geometry operations to use the lower precision model.

	Tolerant Faces	Finds the coincident faces within the specified tolerance. This option is well-suited for skewed faces without sacrificing the precision on the rest of the model.
	Tolerant Faces and Edges	Provides a combination of the **Tolerant Edges** and the **Tolerant Faces** options. Only use this option if the other options fail.
Tolerance	Specifies the tolerance value for the operation. This property only becomes active when you choose a tolerant merge type.	
Allow non-manifold condition	Temporarily allows non-manifold conditions while performing volume extraction.	
Inner Bodies to Subtract	Specifies the extraction method when extracting the external volume.	
	All Inner Bodies	Extracts all inner bodies from the outer body.
	Selected Inner Bodies	Extracts only the selected bodies from the outer body. Select the inner bodies using the appropriate **Select From Scene** tool.
Select Inner Bodies	Click to select inner bodies to extract from the external body with the **Select From Scene** tool.	
Transfer Names	Transfers names from the inner bodies to the extracted volume. You can transfer face names and body names. Activate the **Face Names** option to transfer custom face names from the inner bodies to the extracted volume. If both the external body and inner body contain a custom name, the name on the external body has precedence. Activate the **Body Names** option to transfer the name of the inner body to faces on the same inner body that do not have a custom name.	

Removing Unwanted Features

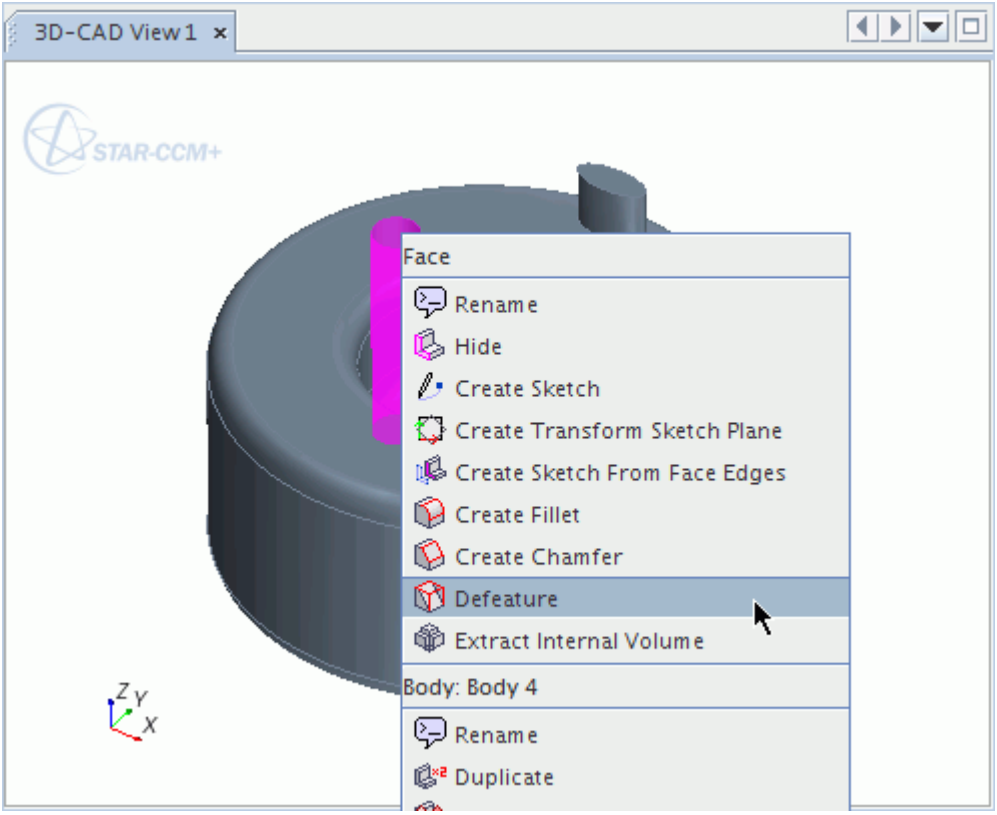
The defeature function in 3D-CAD helps you to remove unwanted features from a geometry.

Unwanted features can include holes, protrusions, chamfers, and fillets. The defeature function can also merge divided surfaces into a single surface.



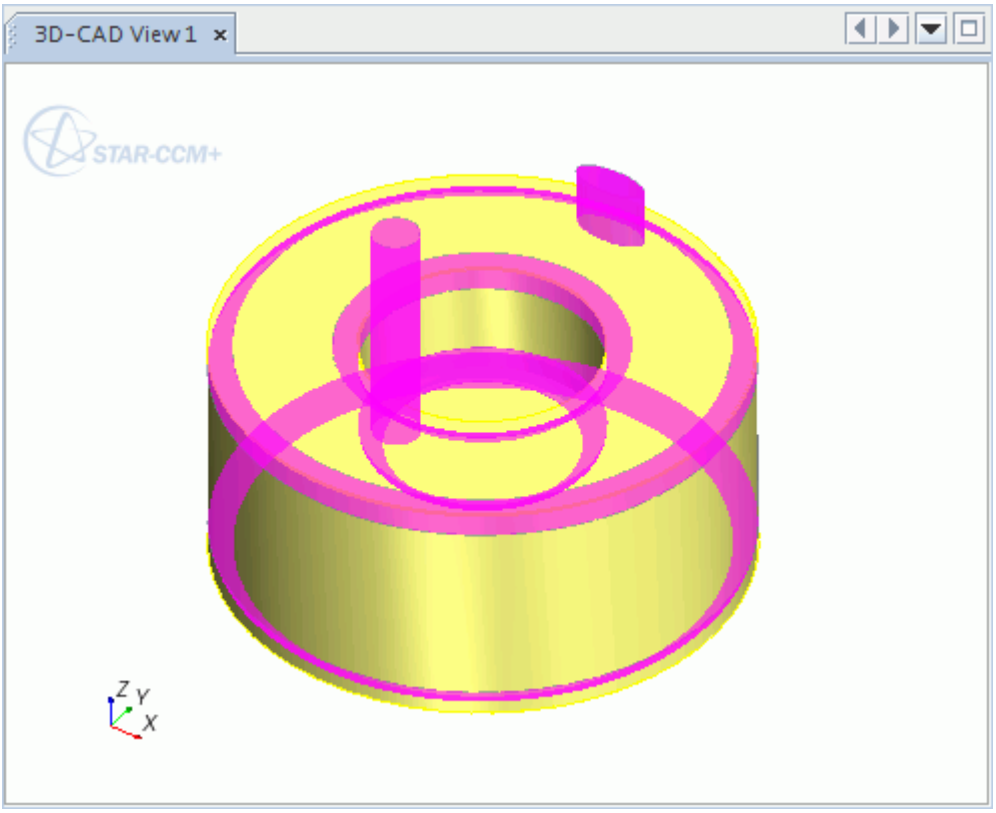
To defeature a geometry:

1. Right-click anywhere on a body in the *3D-CAD view* scene and select **Defeature**.



The *Defeature* panel appears.

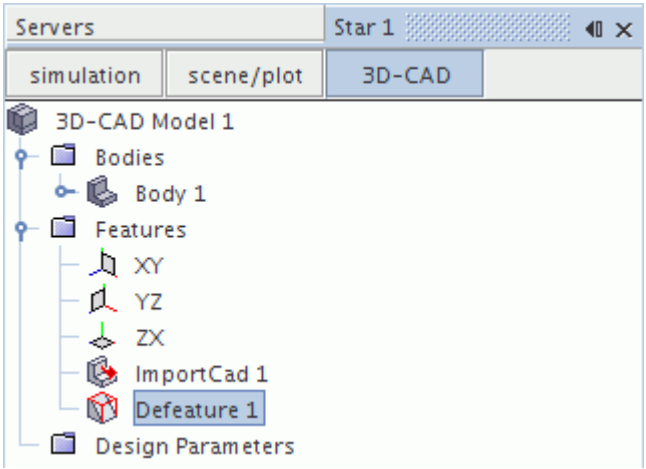
2. Choose the method you want to use:
   - **Automatic** — Choose this option to automatically defeature the geometry using filters. You can remove holes, chamfers, and fillets in this way. For example, you can remove all holes that are smaller than a specified diameter but keep all other holes on the geometry.
   - **Manual** — Choose this option to manually defeature the geometry.
3. Select the features that you want to remove:
   - For automatic defeaturing, see Defeature Panel.
   - For manual defeaturing, select the faces of the features that you want to remove on the geometry. To select multiple features, hold down the <`Ctrl`> key.



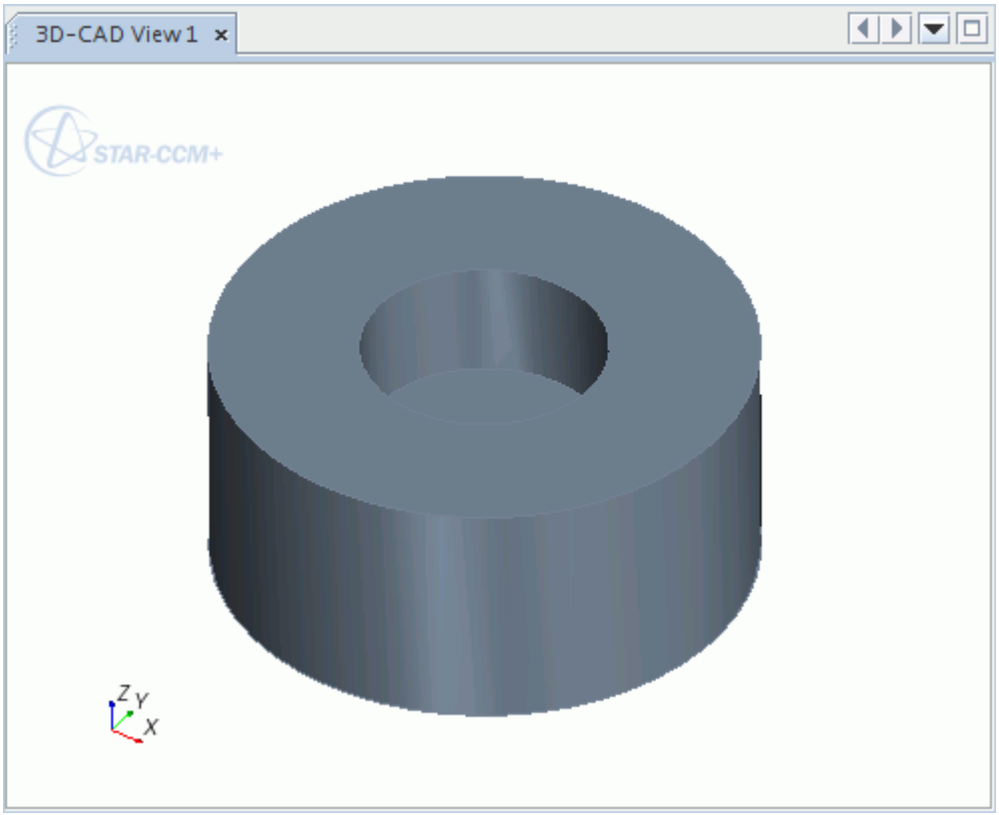
As you make selections, STAR-CCM+ provides a yellow preview of the defeatured geometry.

4. Click **OK** to perform the defeature.

When the defeature operation completes, STAR-CCM+ adds a **Defeature** node to the feature tree.



The defeatured geometry is shown below.



If the defeature operation encounters an error during the operation, the operation completes with a **Partially Successful** status. For example, if you are defeaturing holes, and some of the holes have cutouts in them, the operation removes all holes without the cutouts and leaves all remaining holes with the cutouts. A symbol appears next to the partially successful operation in the 3D-CAD feature tree.

Contents:

Defeature Panel

See Also:

Viewing the Properties of Geometric Objects

Defeature Panel

Use the Defeature operation to remove large numbers of features in a single operation.

Defeature Mode	Specifies the mode for the Defeature operation. Select **Manual** to manually select what features to remove. Select **Automatic** for the operation to search for and remove multiple features in a single operation.
Bodies	Specifies what bodies to include in the Defeature operation. Select **All** to include all bodies, and **Selected** to include specified bodies only. Click the bodies you want to select in the *3D-CAD View* scene.

Fillets	When active, searches for fillets to remove based on the **Radius** and search options.	
Chamfers	When active, searches for chamfers to remove based on the **Width** and search options.	
Holes	When active, searches for holes to remove based on the hole **Diameter** and search options.	
Search Options	Specifies what range of values to search for when removing features.	
	Below	Removes features with values below the specified amount.
	Between	Removes features with values between the specified range.
	Above	Removes features with values above the specified amount.
	Not Between	Removes features with values outside the specified range.
Preview Mode	Specifies what to display in the *3D-CAD View* scene.	
	Result	Displays the result of the operation. This is the default preview behavior, where the result of the operation is displayed in yellow.
	Candidate Defeature Faces	Displays the features that are identified for removal by the Defeature operation. Candidate features are displayed in blue. The corresponding faces are listed in the group box below.
	Successfully Defeatures faces	Displays features that the operation can remove successfully. Successful features are displayed in green. The corresponding faces are listed in the group box below.
	Unsuccessfully Defeatured Faces	Displays features that the operation cannot remove. Unsuccessful features are displayed in red.

Viewing the Properties of Geometric Objects

You can view the properties of features in 3D-CAD.

Geometric object properties can include information such as the length of an edge, the radius of a hole, the area of a face, or the volume of a body. You can view the properties of bodies, faces, and edges. Information about geometric objects is especially useful when using the Defeature operation, where for example, you want to know the size of objects that you want to remove using automatic defeaturing.

To view the properties of a feature:

1. In the *Graphics* window, right-click the geometric object that you want to evaluate and select **Properties** under the appropriate heading.

   For example, to determine the moment of inertia of an edge, right-click the body and select **Properties** under the *Body* heading.

   The property information is displayed in the *Output* window.

```
Body Properties:
-----------------------------------------------------------------------------------------
Body Name:                Body 3
-----------------------------------------------------------------------------------------
Volume:                   0.011538 m^3
Center of Gravity:        [-0.00090914, -0.0068250, 0.25430] m,m,m
Moment of Inertia:        Ix:[0.00040325, -7.4097e-06, -4.8014e-06] kg-m^2,kg-m^2,kg-m^2
                          Iy:[-7.4097e-06, 0.00040288, -7.4262e-06] kg-m^2,kg-m^2,kg-m^2
                          Iz:[-4.8014e-06, -7.4262e-06, 0.00016032] kg-m^2,kg-m^2,kg-m^2
-----------------------------------------------------------------------------------------
Values in Coordinate System: Laboratory
Extents:
Min:                      [-0.13000, -0.14000, 0.0000] m,m,m
Max:                      [0.15000, 0.14000, 0.61000] m,m,m
-----------------------------------------------------------------------------------------
```

You can also evaluate the properties of multiple objects.

Calculating Moment of Inertia in 3D-CAD

As with other properties in 3D-CAD, the moment of inertia is calculated by the Parasolid modeling kernel. Moment of inertia is calculated about the center of gravity. The mass of the object is calculated as $mass = \frac{volume}{density}$. The default density is unity.

See Also:

Removing Unwanted Features

Working with Custom Name Attributes

In 3D-CAD, custom names for bodies, faces, and edges are time-stamped to improve robustness when modifying the geometry.

Renaming these objects in 3D-CAD creates a hidden rename attribute feature in the 3D-CAD feature tree. Deleting a custom name also adds a delete name attribute feature in the feature tree. By default, these name attribute features are hidden.

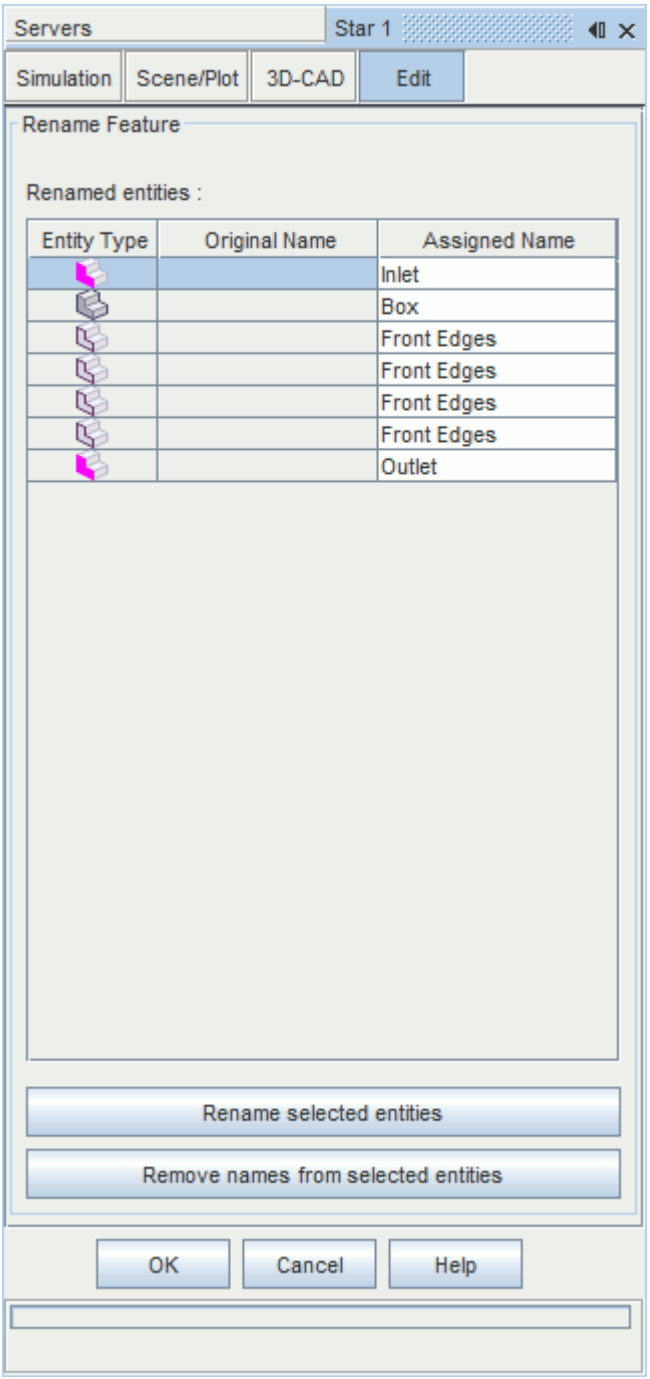
To display the hidden name attributes and use the advanced features:

1. Right-click the **Features** node in the 3D-CAD feature tree and select **Toggle Visibility of Attributes Features**.

   All hidden feature attributes are displayed as features in the feature tree.

2. Right-click the attribute features and select **Edit**.

   The *Rename Feature* panel appears.



3. To modify a custom name, select faces from the *Renamed Entities* list.

   The selected faces are highlighted in the *3D-CAD View* scene.
4. To rename a custom name:
   - Click **Rename selected entities**, enter a new name in the *Rename* dialog, and click **OK**.
   - Edit the name that appears in the *Assigned Name* column and press <`Enter`>.
5. To remove a custom name from the selected faces, click **Remove names from selected entities**.
6. To rename entities while the *Rename Feature* panel is active, right-click the entity in the *3D-CAD View* window and rename it.

The custom name is added to the *Rename Feature* panel.

7. Click the title of a column to sort the content in the *Rename Feature* panel.

Delete the custom name attribute from the feature tree to permanently remove the custom name. Suppress the custom name attribute in the feature tree to temporarily remove the custom name.

Working with Reference Geometry in 3D-CAD

Reference geometry helps you construct models in 3D-CAD. You can also use reference geometry in downstream applications, such as FEA.

Points, Planes, Axes, and Coordinate Systems are all types of reference geometry. You can define reference geometry relative to other geometry, such as imported models, 3D-CAD models, or other reference entities. All reference geometries are created as features in the 3D-CAD feature tree.

To create a reference geometry:

1. Right-click the **Features** node, select **Reference Geometry** > **[Reference Geometry Type]**, and select the desired method.
2. In the panel that appears, set the appropriate options for the reference geometry. See Reference Geometry Options.

   A preview of the reference geometry is displayed in the *3D-CAD View* scene.
3. Click **OK** to create the reference geometry.

   A feature that corresponds to the reference geometry is added to the 3D-CAD feature tree.
4. You can control the visibility of individual reference geometries using one of the following techniques:
   - In the feature tree, right-click the feature that corresponds to the desired reference geometry and select **Hide**/**Show**.
   - In the *3D-CAD View* scene, right-click the reference geometry and select **Hide**/**Show**.

Reference Points

Reference points can refer to a fixed or parametric position in space. Every 3D-CAD model contains a global origin, which is created by default and is located at position [0, 0, 0].

In addition to the global origin, you can create reference points using the following methods:

- By Location
- By Offset
- From Entity Center
- By Projection
- By Import

Reference Axes

Reference axes are useful for controlling extrude and pattern directions. You can create reference axes using the following methods:

- By Two Points
- From Linear Entity

- From Revolved Entity
- From Point and Plane
- By Intersecting Planes

Reference Planes

Reference planes are useful for positioning sketches with reference to an existing point. You can create reference planes using the following methods:

- By Definition
- By Transformation
- By Three Points
- From Point and Axis

Reference Coordinate Systems

Coordinate systems are useful for defining centers of rotation. You can create reference coordinate systems using the following methods:

- By Definition
- By Three Points
- By Import
- From Point and Axis
- From Point and Plane

Contents:

Reference Geometry Options

Reference Geometry Options

Adjust the reference geometry options.

Reference Point Options

Define the reference point using the following options.

Point by Location	
Location	Specifies the absolute [X, Y, Z] coordinates for the reference point.
Point by Offset	
Offset From	Specifies an existing point from which to offset the reference point. The existing point can be that on a body or that of an existing reference point.
Offset	Specifies the offset distance for the reference point. Leaving the offset distance as zero creates a reference point on the existing point.
Point From Entity Center	

Entity	Specifies the point at the center of the selected edge for cylindrical or elliptical entities, or at the center of the toroidal or spherical geometry.
Center Type	Specifies how the center for the selected entity is calculated. Some of the options are deactivated depending on the entity that you select. **Centroid** uses the center of mass of the geometric object. **Geometry Center** uses the equidistant point from all the points that are lying on the geometric object, for example a circular edge or a spherical face. For elliptical edges, the geometry center is the intersection of its major and minor axes. For toroidal faces, the geometric center is the center of revolution. **Midpoint** is the midway between the two extremes of a geometric object, for example, the halfway-point along an edge, or the center of a face.
Point by Projection	
Source Point	Specifies an existing point from which to project to the target entity. The existing point can be that on a body or that of an existing reference point.
Target Entity	Specifies the target plane, surface, or edge to project to. The target plane or surface can be planar or non-planar. The target edge can be linear or non-linear.
Point by Import	
File Name	Specifies the file to import. The file must be in CSV (Comma Separated Value) format. The rows in the file define the points to create, while the columns define the [X, Y, Z] coordinates for the reference point.

Reference Axis Options

Define the reference axis using the following options.

Axis By Two Points	
Start Point	Specifies the start point and end point for the reference axis.
End Point	
Axis From Linear Entity	
Linear Entity	Specifies the linear edge for the axis. The reference axis is collinear with the selected edge.
Axis From Revolved Entity	
Revolved Entity	Specifies the face of the revolved entity. The reference axis is created on the axis of revolution.
Axis From Point and Plane	
Through Point	Specifies the origin of the axis. Select an existing point from the *3D-CAD View* scene.
Normal Plane	Specifies the axis direction. Select a plane or planar surface from the *3D-CAD View* scene. The direction is normal to the selected plane.
Axis By Intersecting Planes	
Planar Entity 1	Specifies the planes and/or planar surfaces, whose intersection defines the axis.

Planer Entity 2	

Reference Plane Options

Define the reference plane using the following options.

Plane By Definition	
Coordinate System	Specifies the coordinate system to use for the plane.
Origin	Specifies coordinates of the origin of the plane.
X Direction	Specifies the direction vector for the X direction of the plane.
Y Direction	Specifies the direction vector for the Y direction of the plane.
Renormalize	Makes sure that the X and Y directions of the sketch plane are normal to each other. Clicking this button can result to changes in some of the specified values.
Plane By Transformation	
Translation Vector	Specifies the direction and magnitude of the translation for the plane.
Rotation Angle	Specifies the rotation of the plane.
Plane By Three Points	
Origin	Specifies the origin of the plane. Select a point from the *3D-CAD View* scene.
Point on X-axis	Specifies the X-axis for the plane. Select a point from the *3D-CAD View* scene.
Point on X-Y plane	Specifies the XY plane. Select a point from the *3D-CAD View* scene.
Plane From Point and Axis	
Origin	Specifies the origin for the plane. Select an existing point from *3D-CAD View* scene.
Normal	Specifies the normal of the plane. Select a linear edge or reference axis from the *3D-CAD View* scene. This edge must not coincide with the point chosen as the origin. The positive X-axis of the plane is the implicit projection of the specified origin onto the specified linear edge or reference axis.

Reference Coordinate System Options

Define the reference coordinate system using three points.

Coordinate System by Definition	
Origin	Specifies the origin of the coordinate system. Define the [X, Y, Z] coordinates of the origin.
X-Axis Direction	Specifies the X-axis for the coordinate system. Define the direction of the X-axis using [X, Y, Z] coordinates.
Vector on X-Y plane	Specifies a vector on the XY plane which is formed by a point on the XY plane and the origin. The Y-axis is a special case of this vector. Define a direction vector that lies on the XY plane using [X, Y, Z] coordinates.
Coordinate System By Three Points	

Origin	Specifies the origin of the coordinate system. Select a point from the *3D-CAD View* scene.
Point on X-axis	Specifies the X-axis for the coordinate system. Select a point from the *3D-CAD View* scene.
Point on X-Y plane	Specifies the XY plane of the coordinate system. Select a point from the *3D-CAD View* scene.
Coordinate System by Import	
Imported Coordinate System	Imports an existing local coordinate system that you created in STAR-CCM+. See Creating a Local Coordinate System.
Coordinate System From Point and Axis	
Origin	Specifies the origin of the coordinate system. Select a point from the *3D-CAD View* scene.
Z-Axis	Specifies the Z-direction of the coordinate system. The positive Y-axis is defined using the right-hand rule for the cross product between the Z-axis and X-axis. Select a linear edge or axis from the *3D-CAD View* scene that does not pass through the origin.
Coordinate System From Point and Plane	
Point	Specifies the point that the X-axis and Y-axis pass through. Select a point from the *3D-CAD View* scene.
Plane	Specifies the Z-axis of the coordinate system. The X and Y directions of the coordinate system are the same as the X and Y directions of the plane or planar surface. Select a plane or planar surface from the *3D-CAD View* scene. The positive Z-axis is normal to the selected plane or planar surface.
Placement	Specifies placement of the origin. **On Point** sets the origin for the coordinate system as the selected point. **On Plane** sets the origin on the selected plane or planar surface.

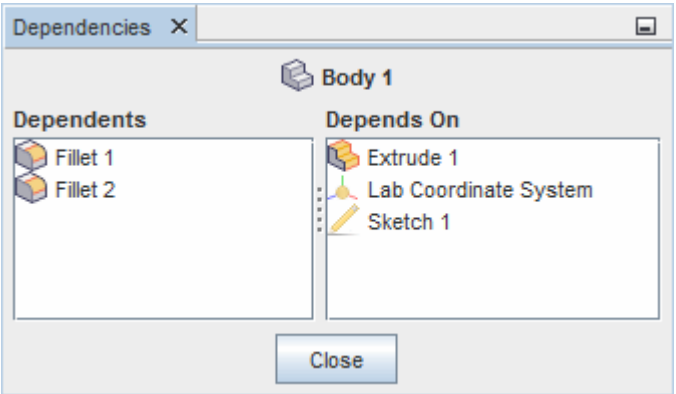
Viewing Dependency Information of Features and Bodies

Use the dependency viewer to see what features a particular feature, body, or design parameter depends on, and what impact these have on the rest of the geometry. The dependency viewer is especially useful when you want to delete a body or feature.

To view the dependencies of a feature or body:

1. Choose the feature or body that you want to view the dependencies on:
   - In the 3D-CAD feature tree, right-click the body or feature and select **Show Dependencies**.
   - In the *3D-CAD View* scene, right-click the body or feature and select **Body/Feature** > **Show Dependencies**.

   The *Dependencies* panel appears.



In this panel, the features that depend on the feature/body that you selected are listed under the **Dependents** group box. The features that the selected feature/body depends on are listed under the **Depends On** group box.

You can position the *Dependencies* panel anywhere in the STAR-CCM+ workspace.

The information displayed in the *Dependencies* panel does not change or update when you select other features in the 3D-CAD feature tree or graphics window. To display the dependency information for a different feature or body, repeat Step 1 for the desired feature.

Note: While the *Dependencies* panel is open, you cannot create or edit features in 3D-CAD.

2. In the *Dependencies* panel:
   a) Select a feature from either the **Dependents** or **Depends On** lists.

      The relevant feature is highlighted in the *3D-CAD View* scene as well as in the 3D-CAD feature tree.

   b) Double-click a feature to view features and bodies that are dependent on that feature, as well as to see what that feature depends on.
3. To continue working on the 3D-CAD model, click **Close** in the *Dependencies* panel.

Importing Geometry into 3D-CAD

3D-CAD allows you to import geometry from many different formats into STAR-CCM+ and modify it ahead of running a simulation. This ability facilitates the simulation process by allowing you to defeature a complicated geometry, extract an internal or external volume, and set up boundary names on the CAD model. This chapter includes information on the following topics:

Contents:

Importing Geometry into a 3D-CAD Model
Modifying Imported Geometry
Using 2D and 3D Curves in 3D-CAD

See Also:

Removing Unwanted Features
Working with Volume Extraction
Setting Face, Edge, and Vertex Names

Importing Geometry into a 3D-CAD Model

3D-CAD supports the following CAD file formats:

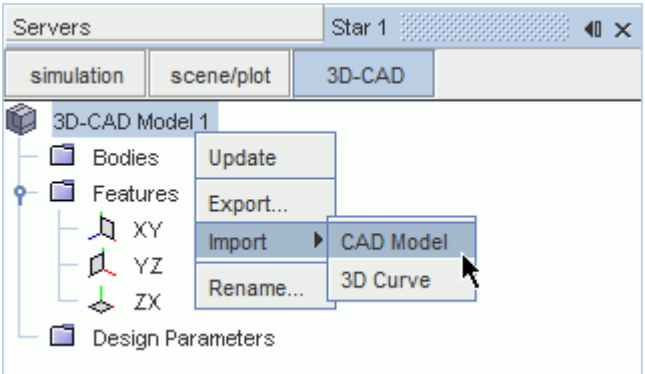
- .x_t/.x_b - Parasolid Transmit files
- .iges/.igs - IGES files
- .step/.stp - STEP files
- .vda - VDA files
- .prt/.asm - Pro/ENGINEER files
- .model/.exp/.session - CATIA V4 files
- .catpart/.catproduct - CATIA V5 files
- .sldprt/.sldasm - SolidWorks files
- .prt - NX files
- .par/.asm - Solid Edge files
- .ipt/.iam - Inventor files
- .3dm - Rhinoceros 3D Model files
- .jt - JT Open B-rep files

You can import solid and sheet (zero thickness) bodies into 3D-CAD. During the import process, you are presented with the option to import sheet bodies.

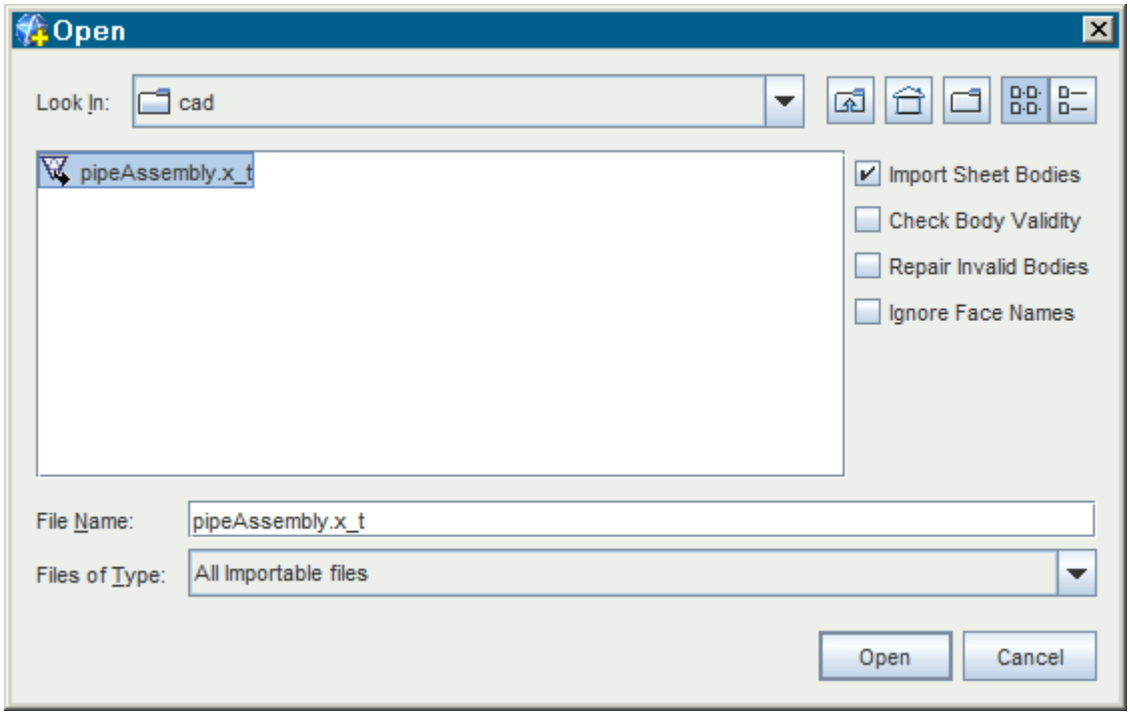
Importing Geometry into 3D-CAD

An active 3D-CAD session is required to import geometry into 3D-CAD. If no session is active, you can create one, or edit an existing 3D-CAD model. You can import multiple CAD geometry files at the same time.

1. Right-click on the **3D-CAD Model** node at the top of the model tree and select **Import** > **CAD Model**

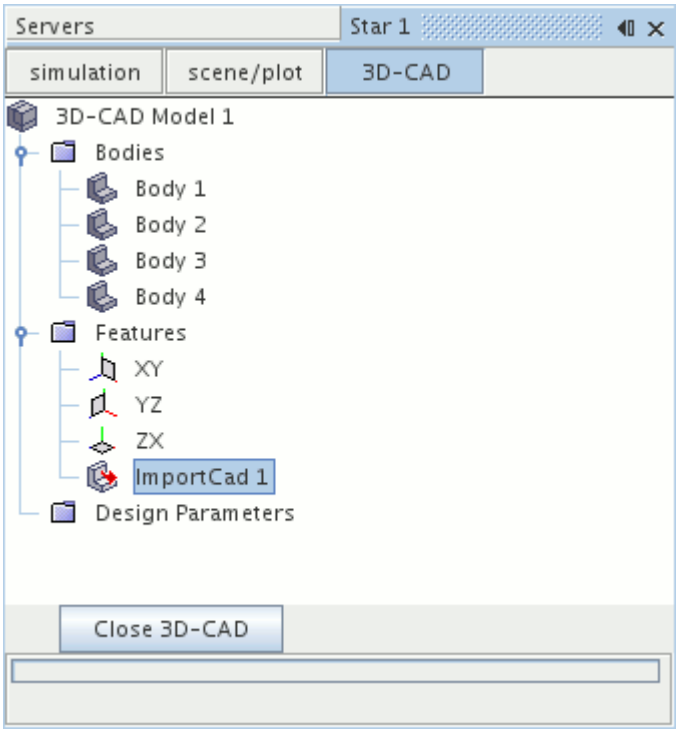


The *Open* dialog appears.



2. Select the CAD files to import.
3. Choose the import options. See 3D-CAD Import Options.
4. To begin the import process, click **Open**.

The geometry is imported into the 3D-CAD model and one or more body nodes are created under the **Bodies** manager node. The import process also adds an **ImportCad** node to the 3D-CAD feature tree.



If any errors are encountered during the import, the operation completes with a **Partially Successful** status. For example, if a valid CAD file does not import bodies into 3D-CAD, the import operation is partially successful. A symbol appears next to the partially successful operation in the 3D-CAD feature tree.

To undo the import:

- Delete the **ImportCad** feature.

Once the geometry is imported, you can modify it using the tools available in 3D-CAD.

3D-CAD Import Options

Select the appropriate import options for your CAD geometry.

Option	Description
Import Sheet Bodies	When activated, imports sheet bodies into 3D-CAD. Deactivate this option if you want to ignore sheet bodies and only import solid bodies.
Check Body Validity	When activated, checks the validity of the CAD geometry during the import process. See Checking Geometry Validity.
Repair Invalid Bodies	When activated, attempts to repair invalid bodies during the import process. This option invokes both the Parasolid healing option from the CAD translator, and the automatic healing from the CAD Repair tool. Parasolid healing only takes place when you activate this option and import the CAD model into STAR-CCM+. If you import the CAD model without activating this option, and subsequently run CAD repair manually, only the CAD repair healing is executed. See Repair Invalid Bodies.
Ignore Face Names	When activated, ignores the names of faces, edges, and vertices in the imported CAD geometry. Some CAD formats can contain unique names for every face, edge, and vertex. Large or complex CAD geometries can have a significant number of named faces, edges, and vertices. Importing such files into 3D-CAD can affect the performance of 3D-CAD.

See Also:

What Is a 3D-CAD Model

Modifying Imported Geometry

You can modify geometries that you import into 3D-CAD in the following ways:

- Adding features to the model by creating more geometries.
- Extracting an internal or external volume in order to analyze the internal or external flow.
- Defeaturing the geometry by deleting bodies, and using extruded and revolved cuts.
- Imprinting a profile on the surface using the imprint operation.
- Renaming faces so that they are defined as separate surfaces when the model is used in the simulation.

See Also:

What Is Volume Extraction?

Removing Unwanted Features
Imprinting Bodies
Setting Face, Edge, and Vertex Names

Using 2D and 3D Curves in 3D-CAD

You can import and use 2D and 3D curves in 3D-CAD.

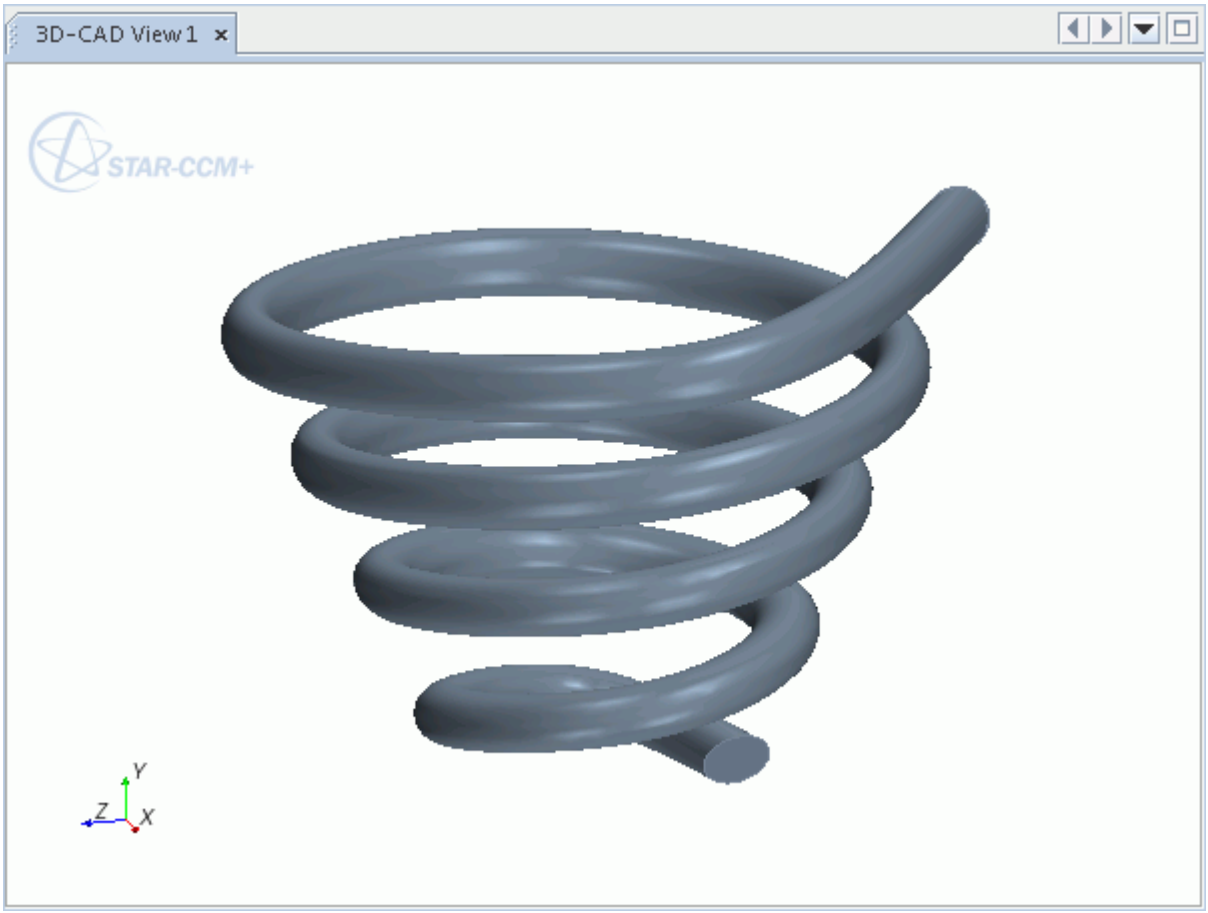
A 2D curve is a sketch entity constrained to a 2D plane (sketch plane), while a 3D curve is not constrained to any 2D plane. A simple example of a 3D spiral is shown below.

You can import curves in 3D-CAD to perform the following operations:

- Extrusions: You can use a closed curve to form a solid body, or an open curve to form a sheet body.
- Sweeps: You can use curves both as profiles or as path curves while creating a Sweep feature. For a curve to be used as a profile, it can be closed (to form a solid body), or open (to form a sheet body). For it to be used as a path, it must be an open curve.
- Lofts: You can use two or more curves (or a combination of curves and planar sketches) to create a Loft feature. For a curve to be used as a profile, it must be closed (to form a solid body), or open (to form a sheet body). For it to be used as a guide wire, it must be an open curve. A Pierce constraint can be used to connect profiles to guide wires—profiles and guide curves must intersect. See Pierce Constraint.

You cannot use 3D curves with the Revolve feature. Composite curves that contain sharp edges are not supported. 3D curves that you import are fully editable. 2D curves are imported as sketches on the XY sketch plane and are fully editable. You can reposition the sketch onto a transform sketch plane if required.

An example of a sweep using a 3D curve is shown below, whereby it is used as the path for the sweep feature.



Contents:

Importing 2D and 3D Curves from a CSV File
Importing Multiple 3D Curves from a Single CSV File
Importing and Replacing Sketch Curves
Curve Import Options
File Format for Imported Curves

Importing 2D and 3D Curves from a CSV File

You require an active 3D-CAD session to import a 2D or 3D curve into 3D-CAD. If no session is active, you can create one, or edit an existing 3D-CAD model. Files containing curves must be tabulated in the correct format, with the units of the coordinate points in meters. See Format for Imported Curves.

To import curves:

1. Right-click the **3D-CAD Model** node at the top of the model tree and select the appropriate option:
   - For 2D curves, select **Import** > **2D Curve**.
   - For 3D curves, select **Import** > **3D Curve**.
2. In the *Open* dialog, select the `.csv` file containing the coordinates of the 2D or 3D curve.

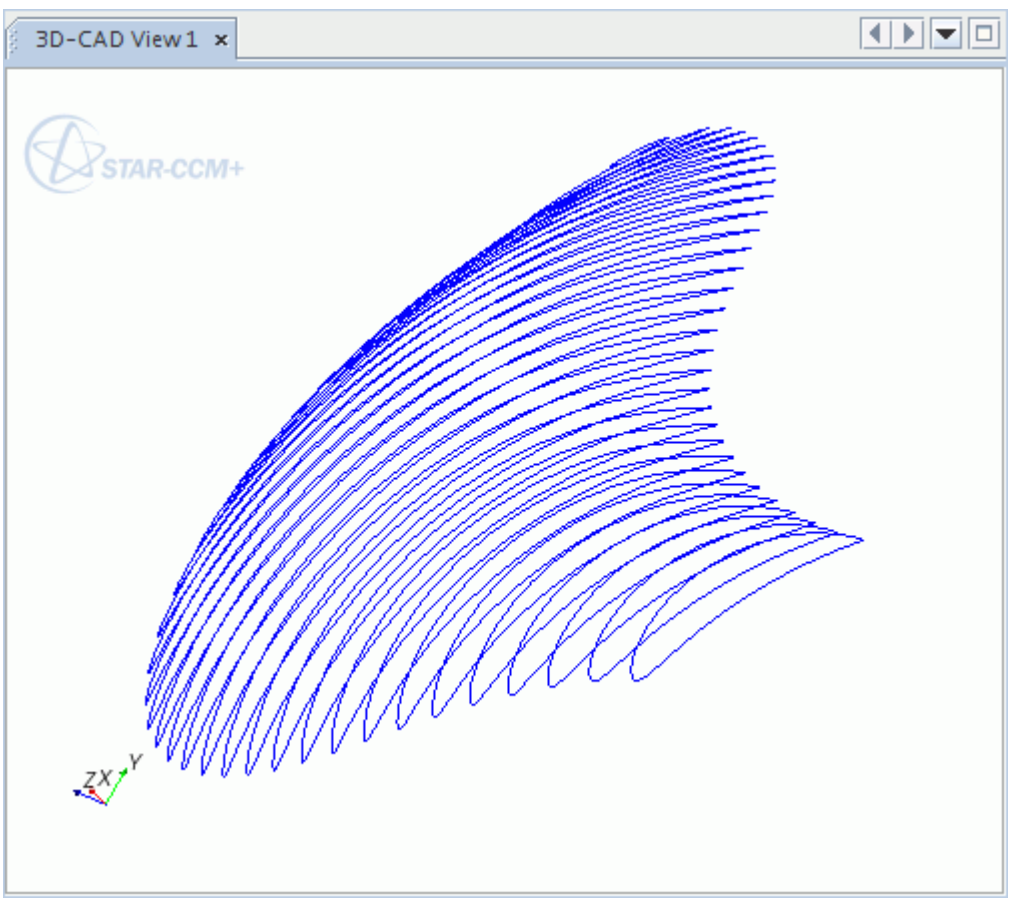
   You can import multiple `.csv` files at the same time. The `.csv` file can contain multiple 3D curves.
3. Click **OK**.
4. In the *Import Curve Options* dialog, select the desired import options. See Curve Import Options.
5. Click **OK** to import the curve.

2D curves are imported as sketch features and are constrained to the default **XY** sketch plane. 3D curves are imported as **3DCurve** features and are not constrained.

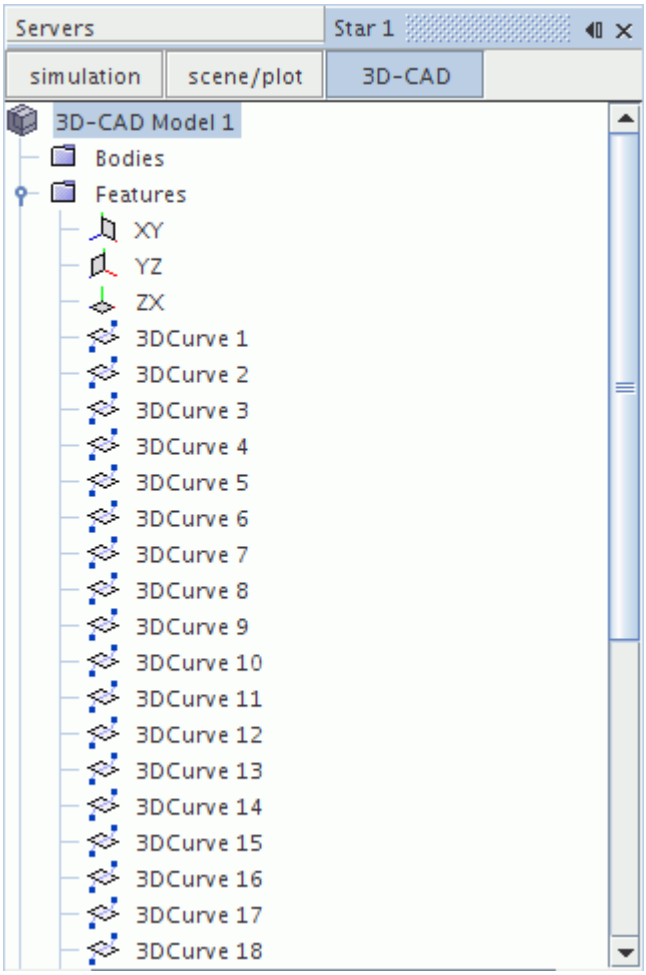
Importing Multiple 3D Curves from a Single CSV File

You can import multiple closed 3D curves that are stored in a single file. Importing multiple open 3D curves is not supported. 3D-CAD looks for identical vertices in the import files and classifies everything in between these two points as a single 3D curve. As a result, each 3D curve is closed, regardless of whether they are open or closed in the .csv file. You can use a blank line to separate each 3D curve in the `.csv` file.

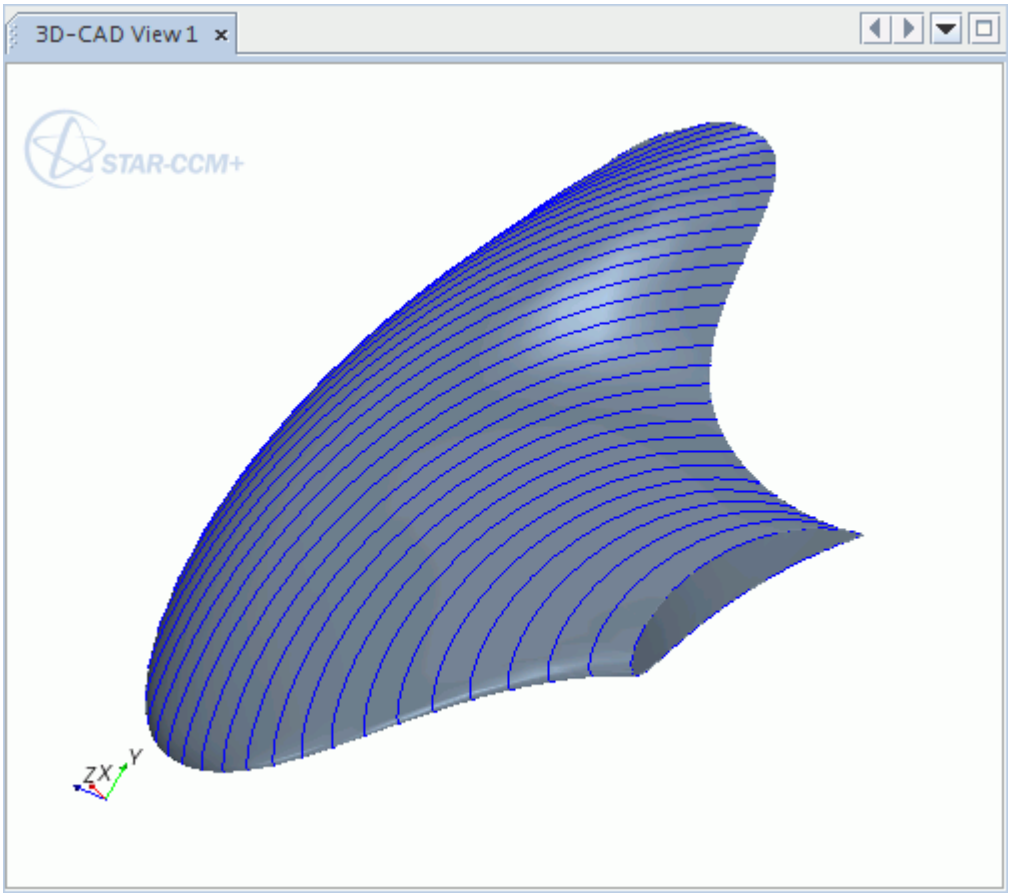
In the following example, the 3D curves that make up the sections through a propeller blade are imported into 3D-CAD from a single `.csv` file.



The 3D curves are imported as separate features.



You can then use these imported profiles to create a loft of the propeller blade.



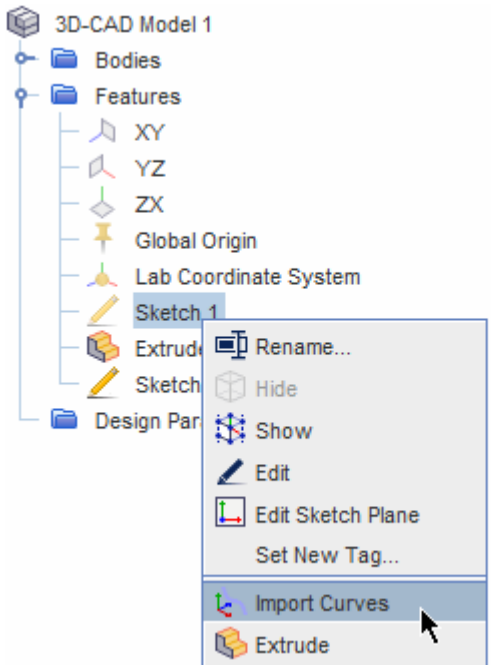
Importing and Replacing Sketch Curves

You can combine tabular and user-created sketches by importing a curve into an existing 2D or 3D sketch. You can also replace a specific spline within a sketch with a new spline from a .csv file to enable tabular-based design changes.

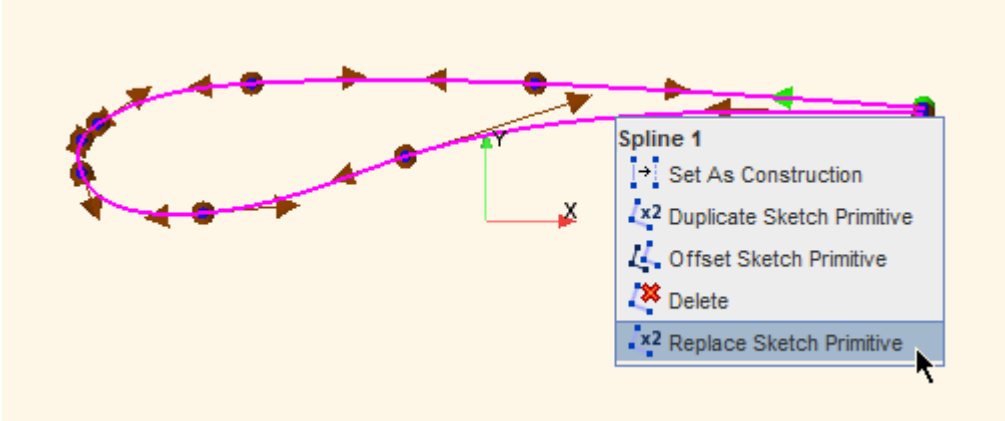
All curves that you import must be in .csv format. See File Format for Imported Curves.

To import and replace sketch curves:

1. Choose where to import the sketch curves, or which spline to replace:
   - To import sketch curves into an existing sketch, right-click the target sketch in the 3D-CAD feature tree and select **Import Curves**.



- To replace an existing spline within a sketch, while in sketch mode, right-click the target spline and select **Replace Sketch Primitive**.



2. In the *Open* dialog, select the `.csv` file that contains the curve data and click **OK**.
3. In the *Import Curves to Sketch* dialog, set the import options and click **OK**.

   You can import the curve as a spline or a polyline. You can also choose to close the curve if it is open. See Curve Import Options.

The curve is imported into the sketch, and is added in addition to any sketch entities that are already within it. When replacing a specific spline, use the **Undo** action to revert back to the original spline.

Curve Import Options

Set the import options to control how 2D or 3D curves are imported from `.csv` files.

Import as spline	Imports the 2D or 3D curve as a spline. If you intend to use a 3D curve in a 3D-CAD operation (such as **Extrude**, **Loft**, and **Sweep**), select this option.
Import as polyline	Import the 2D or 3D curve as a succession of straight lines, using the coordinates as end-points. Note that the resulting polyline does not define a smooth curve.
Close the curve	Closes the imported curve. Useful when the spreadsheet contains an open profile, such as an aerofoil profile, and you want to close this.
Coordinate System	Specifies the coordinate system in which the points are defined. This option only applies to 3D curves. Currently only Cartesian and Cylindrical coordinates are supported.

File Format for Imported Curves

The `.csv` file containing the 2D or 3D curve coordinate points must be in comma-separated values (`.csv`) format.

Text is not supported in this file—meaning that headers, descriptions, or units must not be included in these files. A maximum of 8 decimal places per coordinate is supported. There is no limit to the number of rows (coordinate lines) per file. The coordinate points must be in meters. If your geometry uses a different system of units, convert the coordinate points into meters.

An example file for a 3D curve, in Cartesian coordinates, is shown below:

```
0.01817128,0.01377324,-0.01098872
0.04074841,0.02954630,-0.01947049
0.07237303,0.04867008,-0.02909696
0.12597960,0.07179400,-0.03726481
0.16208829,0.08217564,-0.04081605
0.22814133,0.09360534,-0.04507754
0.28460602,0.09557598,-0.04720828
```

Each row represents a point on the curve, and must contain three comma-separated values, for both 2D and 3D curves.

Requirements for importing 3D curves:

- When using Cartesian coordinates, the format must be [X, Y, Z]
- When using Cylindrical coordinates, the format must be [r, ϑ, z]
- It is possible to import 2D profiles and sketches as a 3D curve by setting one of the axes to a constant for all points

Requirements for importing 2D curves:

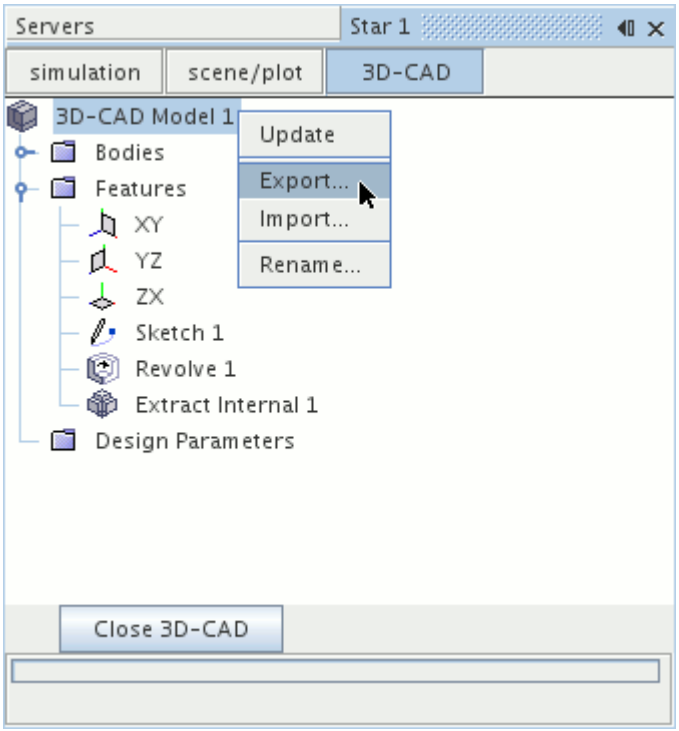
- The format must be [X, Y, Z]
- The Z axis must be constant
- The constant value of Z can be non-zero, however these values are ignored, and the 2D curve is placed on a sketch on the XY plane

Exporting Geometry from 3D-CAD

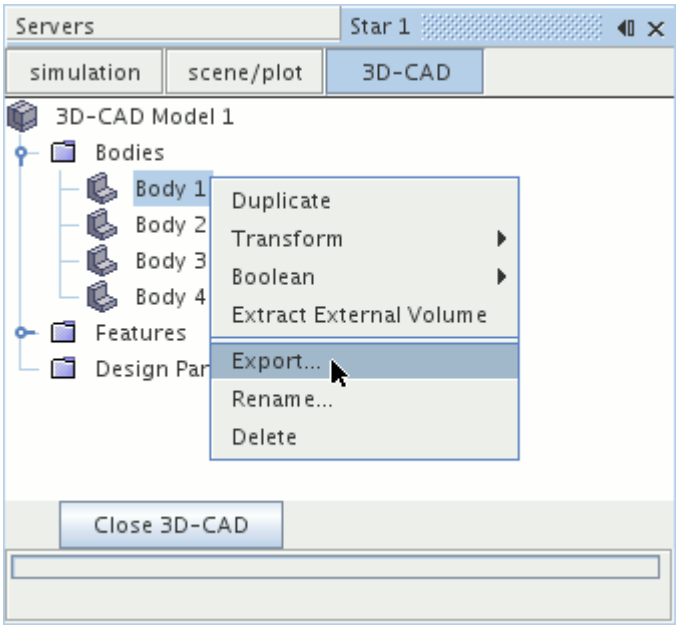
3D-CAD allows you to export solid and sheet bodies to a selection of neutral CAD formats. You can choose to export the whole model, or only selected bodies.

To export a 3D-CAD geometry:

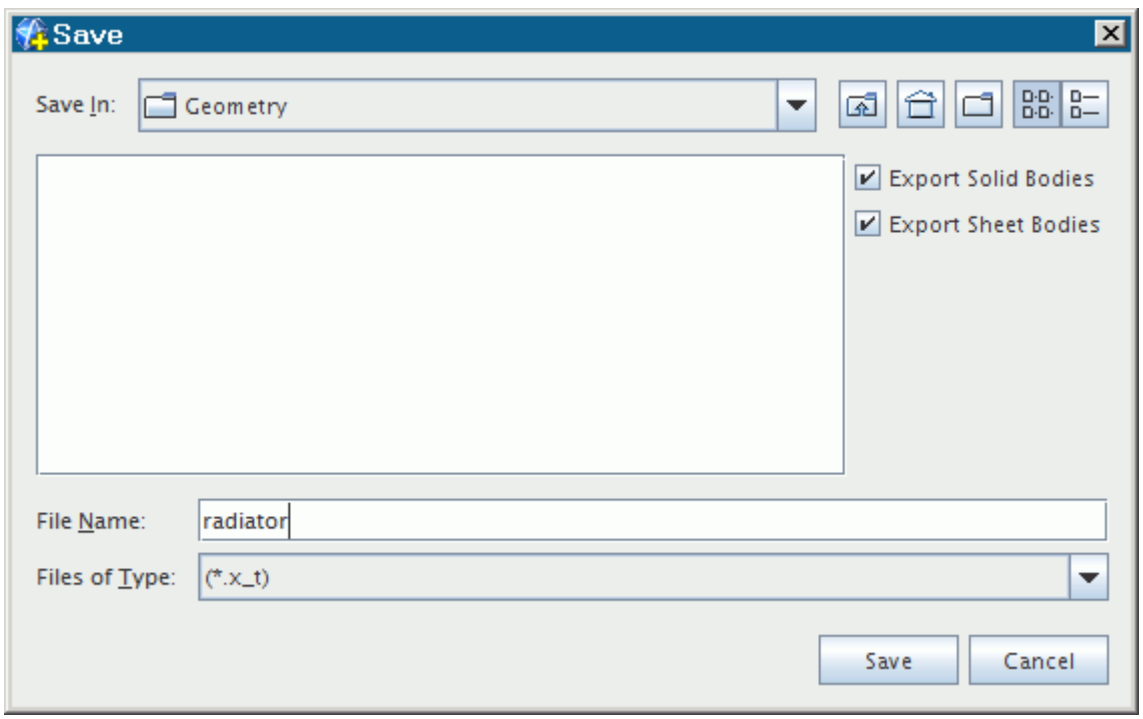
1. Choose the bodies to export:
   - If you want to export the entire 3D-CAD model, right-click the root node in the 3D-CAD feature tree and select **Export...**



   - If you want to export selected bodies only, multi-select the bodies within the **Bodies** node, then right-click on a selection and select **Export...**



The *Save* dialog appears.



2. Navigate to the directory where you want to save the file.
3. In the *File Name* field, enter a name for the file.
4. In the *Files of Type* field, select an export file format:
   - Parasolid Text (`.x_t`)
   - Parasolid Binary (`.x_b`)
   - Initial Graphics Exchange Specification (`.iges`)[1]

- Standardized Exchange of Product (`.step`)[1]

5. Choose the export options. See 3D-CAD Export Options.
6. Click **Save**.

The file is exported to the chosen location. If you renamed a body during the 3D-CAD session, its name is preserved in the exported file.

1. Exporting to STEP and IGES formats requires a CAD Exchange licence.

Contents:

3D-CAD Export Options

3D-CAD Export Options

When exporting a model or body in 3D-CAD, you have a variety of options to choose from, depending on the format you choose.

These options are listed below:

Export Solid Bodies	Specifies whether to export solid bodies.	
	Activated	Includes solid bodies in the export.
	Deactivated	Excludes solid bodies in the export.
Export Sheet Bodies	Specifies whether to export sheet bodies.	
	Activated	Includes sheet bodies in the export.
	Deactivated	Excludes sheet bodies in the export.
Save Solids as Faces (IGES)[1]	Specifies how the solid geometry is saved.	
	Activated	Saves all faces of solid bodies as Trimmed Surface Entities (Type 144). The solid is represented by an Associativity Instance Entity (Type 402).
	Deactivated	Saves the geometry as a Manifold B-Rep Solid (Type 186).
Convert Analytics to NURBS (IGES/STEP)[1]	Specifies whether to convert analytic surfaces (such as cylinders, cones, and planes) to NURBS or not.	
	Activated	Analytic surfaces are converted to NURBS surfaces.
	Deactivated	Analytic surfaces remain the same.
STEP Format[1]	Specifies which version of STEP to export to.	
	AP 203	Saves using Application Protocol 203.

	AP 214	Saves using Application Protocol 214.

1. Exporting to STEP and IGES formats requires a CAD Exchange licence.

Working with 3D-CAD Models

With 3D-CAD, you can create models and perform a full CFD simulation all within the user-friendly environment of STAR-CCM+. The fully integrated modeler makes using 3D-CAD models in a simulation quick and easy, and enhances the process of performing design and optimization studies.

To use a 3D-CAD model in a simulation, the first step is to convert it to geometry parts. This action brings the CAD data into the simulation, where it can be assigned to regions and boundaries.

Following this action, it is possible to modify the 3D-CAD model and run the same simulation for the altered geometry. These processes can be automated using the scripting functionality of STAR-CCM+. You can also synchronize the geometry parts with the 3D-CAD model so that additional bodies that you create in 3D-CAD are automatically created when you update the geometry parts list. If the changes you make in 3D-CAD result in bodies being removed, the corresponding parts are removed from the parts list.

Contents:

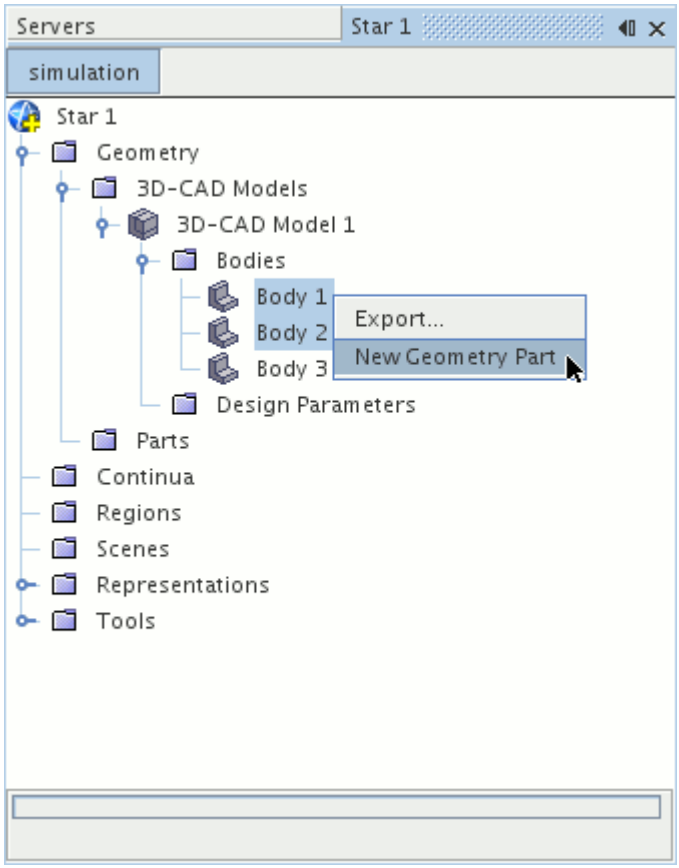
Creating Geometry Parts from 3D-CAD
Part Creation Options Dialog
Modifying a 3D-CAD Model
Updating the 3D-CAD Model
Synchronizing Geometry Parts with the 3D-CAD Model

Creating Geometry Parts from 3D-CAD

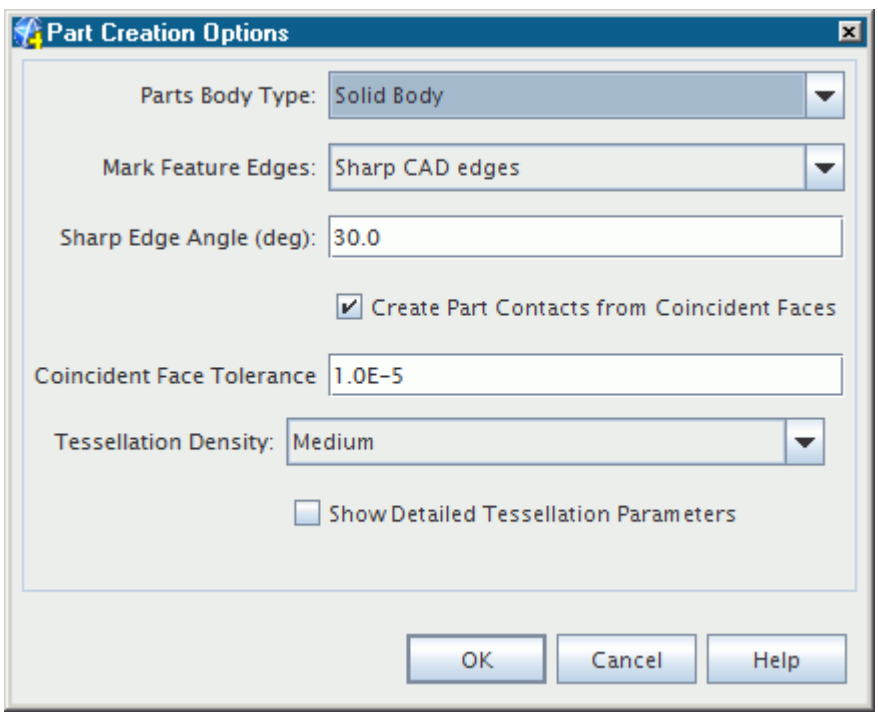
To use a 3D-CAD model in a simulation, it must first be used to create geometry parts. This process is described below and can be performed on either the whole 3D-CAD model, or on one or more of the model bodies.

To create geometry parts:

1. Exit 3D-CAD.
2. Select the **3D-CAD Model** node or one or more **Body** nodes, right-click, and select **New Geometry Part**.



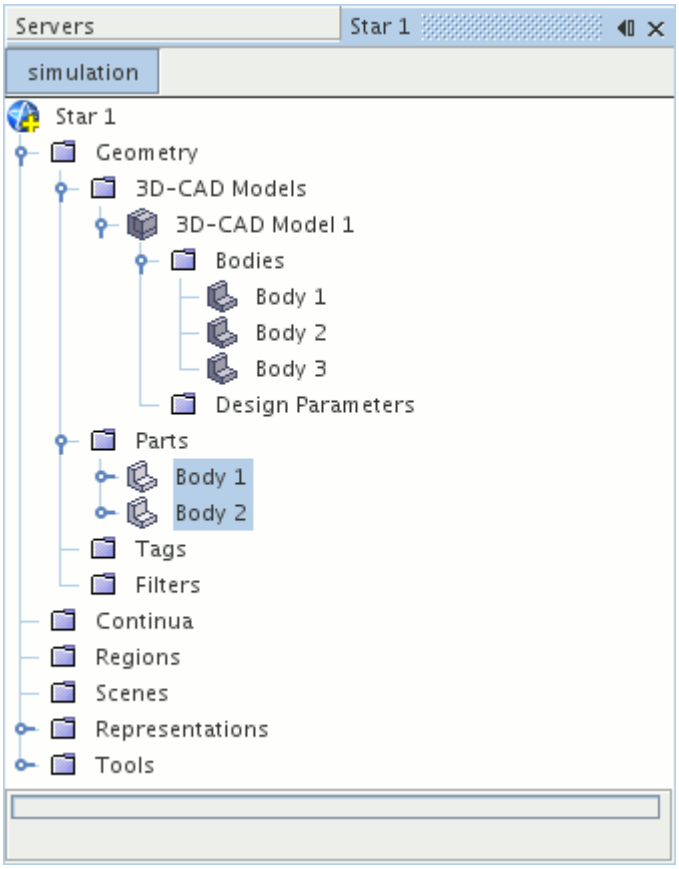
The *Part Creation Options* dialog appears.



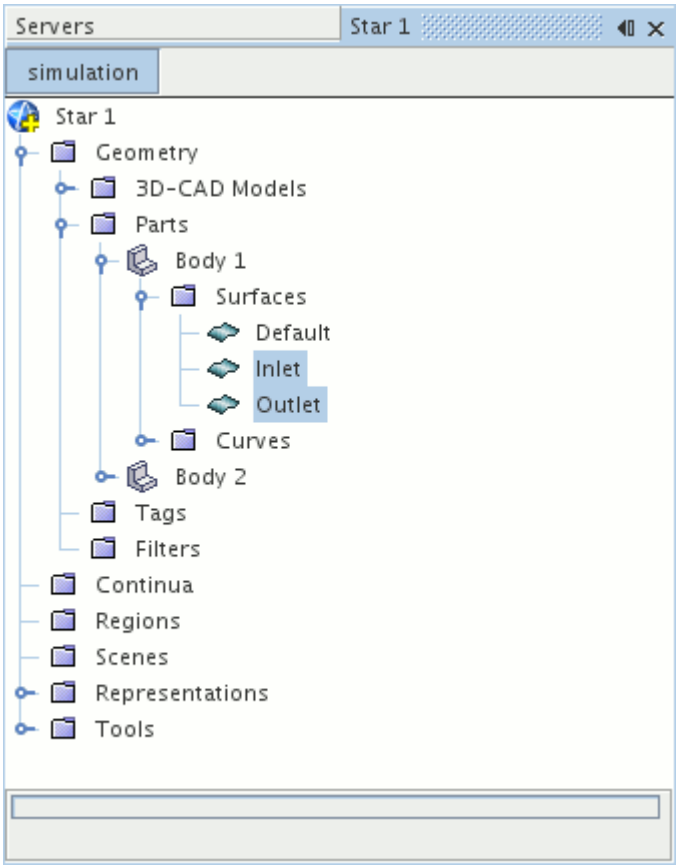
3. Specify the parameters for the new part. See Part Creation Options Dialog and Tessellation Density.

4. Click **OK**.

When geometry parts are created from the whole 3D-CAD model, all the bodies that are contained within the model are shown under the **Geometry** > **Parts** manager node. When particular bodies are selected, only these bodies are shown under the **Parts** node.



If you renamed any of the model faces during the 3D-CAD creation process, these faces are defined as separate surfaces under the **Surfaces** node for the respective body part. In the example below, two faces on **Body 1** were renamed `Inlet` and `Outlet`, respectively. When the body was converted, these faces were used to create separate surfaces. All other faces are included in the **Default** surface definition. To rename faces in the 3D-CAD model, edit the model. When done, update the geometry parts that are affected.



See Also:

What Is a Geometry Part?

What Is a 3D-CAD Model

Updating 3D-CAD Parts

Part Creation Options Dialog

When creating parts, the following options are available:

Parts Body Type	This option allows you to specify what type of body is used to create the parts.	
	Solid and Sheet Body	Both solid and sheet bodies are used to create parts.
	Solid Body	Only solid bodies are used to create parts.
	Sheet Body	Only sheet bodies are used to create parts.
Mark Feature Edges	This option allows you to specify which of the mesh edges that lie along CAD edges are marked as feature edges. CAD edges are the edges or curves that belong to the underlying CAD model; mesh edges are the line segments that bound the mesh faces.	

	No CAD edges	Does not mark any mesh edges as feature edges.
	Sharp CAD edges	Marks any mesh edge that lies along a sufficiently sharp CAD edge (based on the *Sharp Edge Angle* value) as a feature edge. For any given CAD edge, either all or none of the mesh edges that lie along the CAD edge are marked as feature edges.
	All CAD edges	Marks all mesh edges that lie along any CAD edges as features edges.
Sharp Edge Angle (deg)	Allows feature curves to be created based on a supplied angle. The default angle is 30 degrees.	
Create Part Contacts from Coincident Faces	Activating this option causes STAR-CCM+ to search for coincident faces in the 3D-CAD geometry. For each pair found, part contact data is created and stored in the *Contacts* property of each part. See Creating Part Contacts From Coincident Faces.	
Coincident Face Tolerance	When the *Create Part Contacts from Coincident Faces* option is activated, part contact data is generated for parts that share coincident faces that are within the *Coincident Face Tolerance* distance from their neighbors. This option is intended to close small, erroneous gaps in imported geometries. Do not change this value specifically to create contacts or interfaces between parts that have large gaps between them. If parts have large gaps between them, move one part closer to the other using a translation operation.	
Tessellation Density	This option determines the level of surface refinement that STAR-CCM+ uses when creating the geometry part, and affects the degree to which surface curvature is preserved in the created part. In general, it is recommended that you start by using one of the coarse options, then review the surface triangulation to see if the geometry part needs to be retessellated with a higher density. See Retessellating a Part. For geometries with high amounts of curvature, and for which a higher surface fidelity is desired, one of the finer options is recommended. The tessellation density option has no effect on planar surfaces—the same triangulation is obtained for these surfaces regardless of the settings used.	
	Very Coarse	Uses the least number of triangles to discretize a curved surface.
	Coarse	Uses fewer triangles to discretize a curved surface.
	Medium	Uses the default number of triangles to discretize a curved surface.
	Fine	Uses more triangles to discretize a curved surface.
	Very Fine	Uses the most number of triangles to discretize a curved surface.
Show Detailed Tessellation Parameters	Activating this option displays detailed tessellation Parameters within the *Part Creation Options* dialog. You can then edit these Detailed Tessellation Parameters as required.	

Modifying a 3D-CAD Model

As the 3D-CAD modeler is fully integrated into the STAR-CCM+ environment, it is quick and easy to modify the 3D-CAD model before and after the simulation is run.

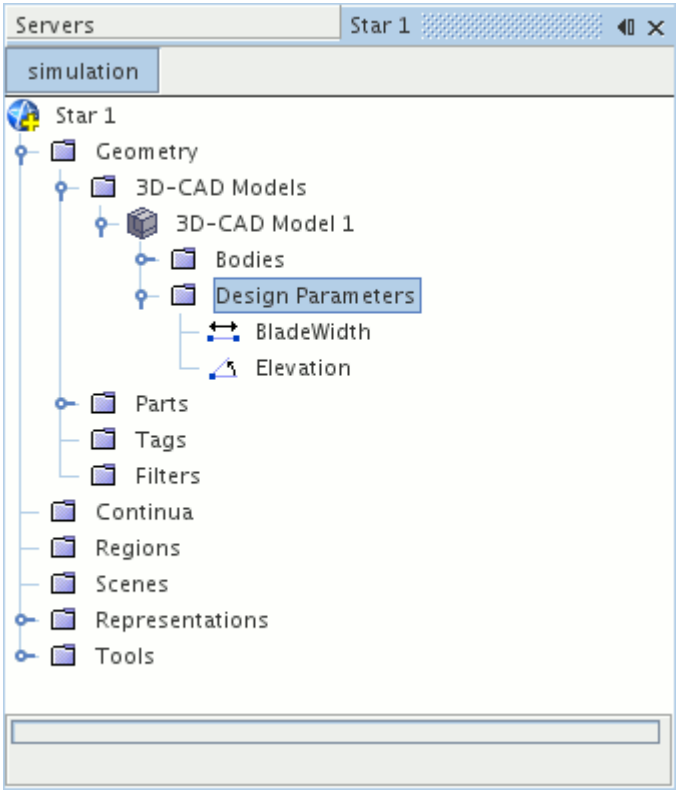
If a solution has been found and you wish to rerun the simulation for a slightly different geometry, you can modify the 3D-CAD model and rerun the simulation. In this case, STAR-CCM+ automatically maps the previous solution onto the new geometry, decreasing the simulation run-time.

You can modify a 3D-CAD model by:

- Editing it explicitly
- Changing the value of a design parameter

The first method activates 3D-CAD and gives you access to the full range of CAD tools to edit the geometry.

The second method allows you to modify the model geometry without activating 3D-CAD. To use this method, you must prepare design parameters in advance. Design parameters appear under the **Design Parameter** node for the 3D-CAD model in the main simulation tree once you close 3D-CAD. By changing the value of a design parameter and updating the model, you can change the associated dimension in the 3D-CAD model.



For the changes to take effect in the rest of the simulation, perform the following steps:

- Update the affected geometry parts
- Regenerate the surface and volume mesh

The region and boundary data is mapped onto the updated geometry and you can proceed with the analysis. It is good practice to check that the parts, regions, and boundaries are all associated with the correct sections of the geometry following a modification to the 3D-CAD model.

When the original 3D-CAD model is used to create geometry parts, a single part surface, named **Default,** represents all unnamed faces.

If you:

1. rename or split the **Default** part surface into new part surfaces,
2. then modify the geometry in 3D-CAD,
3. and update the geometry parts,

the CAD surfaces and boundaries that are associated with each part surface are maintained as much as possible.

When you name surfaces on a 3D-CAD model, then use that 3D-CAD model to create geometry parts, individual part surfaces are created for each of the named 3D-CAD surfaces. If you edit the part in 3D-CAD, then update the geometry parts, the part surfaces, and associated boundaries, are maintained.

See Also:

What Is a 3D-CAD Model
What Is a Design Parameter?
Updating the 3D-CAD Model
Updating 3D-CAD Parts
What Are Leaf-Level Parts?

Updating the 3D-CAD Model

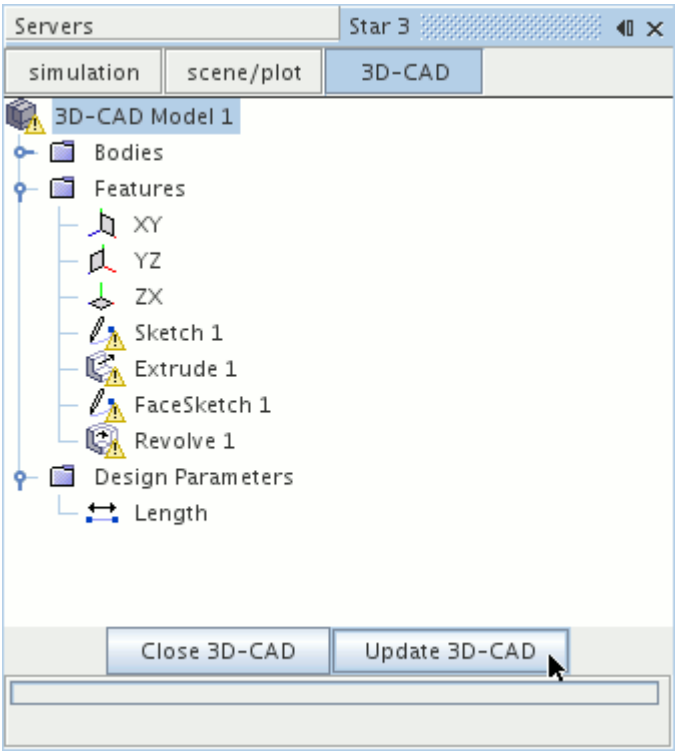
The 3D-CAD model contains all features that define the 3D-CAD geometry. When you make a change to a feature in the 3D-CAD model, you must update the model for the changes to take effect.

For example, increasing the number of blades on a fan by changing the value of a design parameter, requires you to update the model. Updating the model causes 3D-CAD to rebuild the model based on active features. Features that you have rolled back or suppressed are not affected.

When a model needs updating, the **3D-CAD Model** node and affected feature nodes display the ⚠ (**Out-of-Date**) feature state icon.

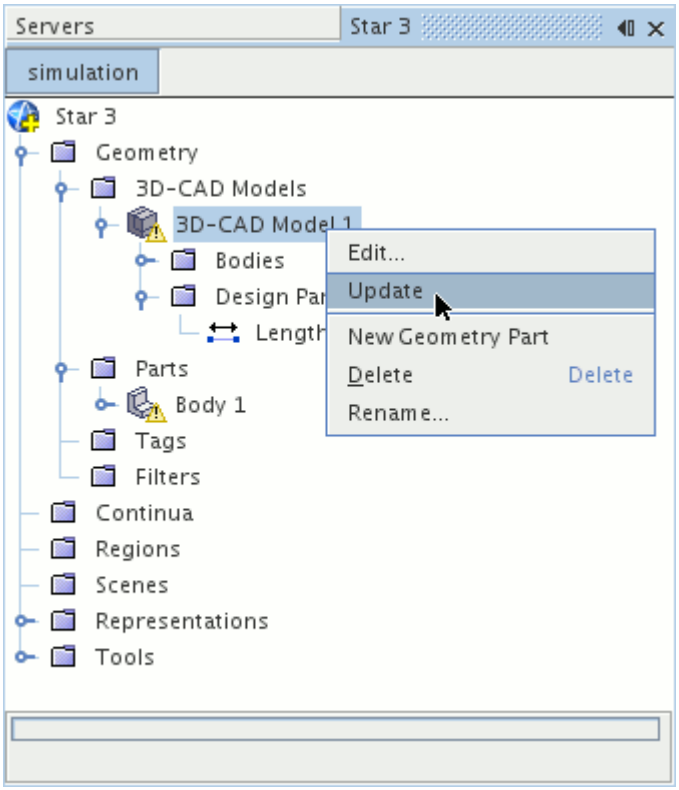
To update the model:

- Click the **Update 3D-CAD** button.

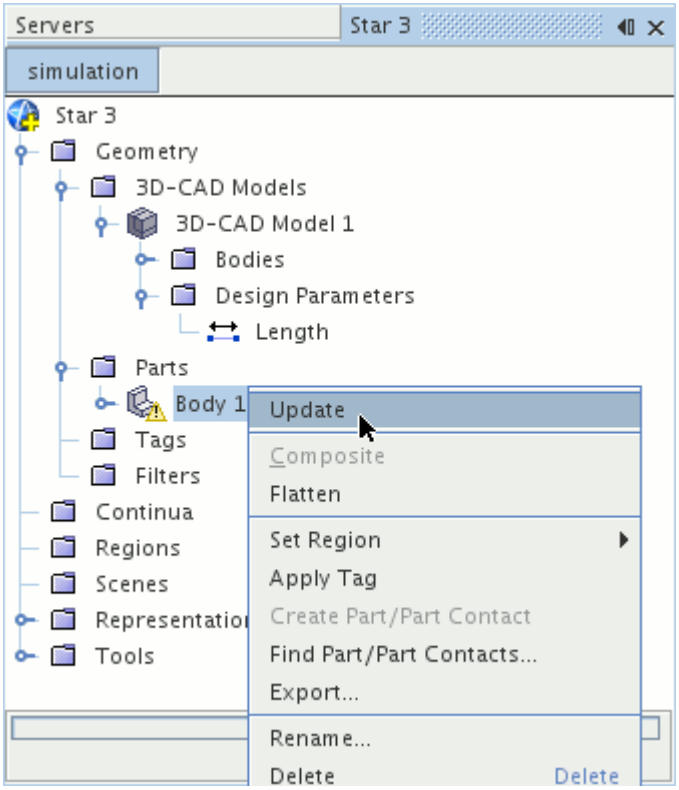


- Alternatively, right-click on the 3D-CAD Model node and select **Update**.

You can use the same process to update a model that is showing the ⚠ (**Out-of-Date**) icon under the **Geometry** > **3D-CAD Models** node in the main simulation tree. This action is necessary if you modified the model by changing the value of a design parameter outside of 3D-CAD.



If you created geometry parts from the 3D-CAD model, update them by right-clicking and selecting **Update**.



See Also:

Rolling Back
Suppressing Features
Understanding Feature States
Modifying a 3D-CAD Model
What Is a Design Parameter?
Modifying a 3D-CAD Model

Synchronizing Geometry Parts with the 3D-CAD Model

Synchronize the geometry parts with the 3D-CAD model to maintain an up-to-date list of geometry parts every time you update the 3D-CAD model and geometry parts list.

When you modify the 3D-CAD model, either directly within 3D-CAD, or through a design parameter, you must manually update the model. You must also manually update the geometry parts that are created from the 3D-CAD model.

If you make changes to the 3D-CAD model that result in additional bodies, you must manually create the corresponding geometry parts from the 3D-CAD model, in addition to the existing geometry parts. If the changes to the 3D-CAD model result in the removal of bodies, you must manually delete the corresponding obsolete geometry parts. Keeping track of these changes in a large assembly can be challenging.

Synchronizing the geometry parts with the 3D-CAD model automatically resolves the changes that you make in the 3D-CAD model. If the changes to the 3D-CAD model result in additional parts, the corresponding geometry parts are created. If the changes to the 3D-CAD model result in the removal of parts, the corresponding geometry parts are removed. When synchronized, named entities in the 3D-CAD model (faces/edges/vertices) are linked to the equivalent geometry part objects (surfaces/curves/points) and are also updated to match the changes made in the 3D-CAD model.

Note: Synchronizing a 3D-CAD model with geometry parts still requires you to manually assign the new parts to regions. Similarly, if a geometry part is removed as a result of the synchronization, you must manually delete or update the region that this part was associated with.

Note: Any changes that you make to the geometry parts at the part level, such as splitting surfaces by patches or combining patches, will be lost when you next update the geometry parts.

To synchronize the 3D-CAD model with the geometry parts:

1. Select the **3D-CAD Models** > **[3D-CAD Model]** that you want to synchronize.
2. In the *Properties* window set *Part Update Method* to **Synchronize**.
3. Modify the 3D-CAD model. See Modifying a 3D-CAD Model.

   The 3D-CAD Model and geometry parts are flagged as outdated with the ⚠ (**Out-of-Date**) icon. No changes are made to the geometry parts at this stage.
4. Update the 3D-CAD model. See Updating the 3D-CAD Model.
5. Update the geometry parts. See Updating the 3D-CAD Model.

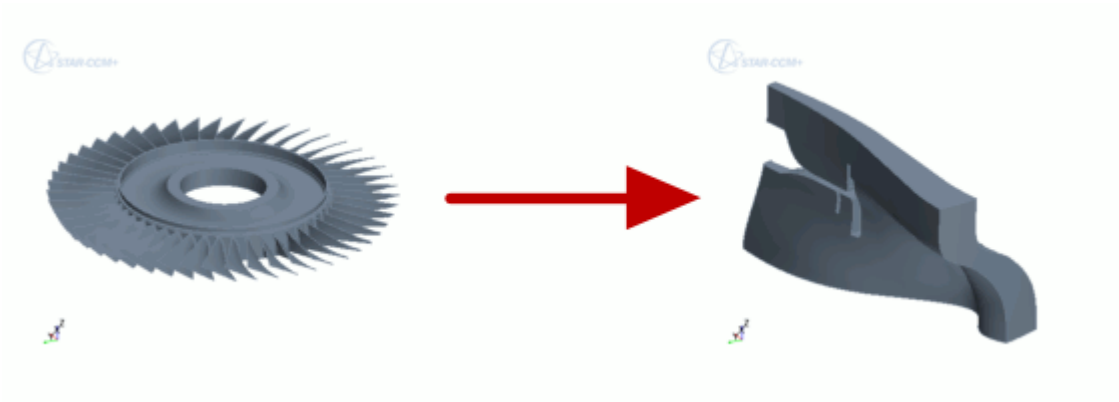
The geometries parts are synchronized with the 3D-CAD model. If new bodies were added in the 3D-CAD model, the corresponding geometry parts are added to the geometry parts list. Similarly, if bodies were

removed from the 3D-CAD model, the corresponding geometry parts are removed from the geometry parts list.

The Turbo Slicing Tool

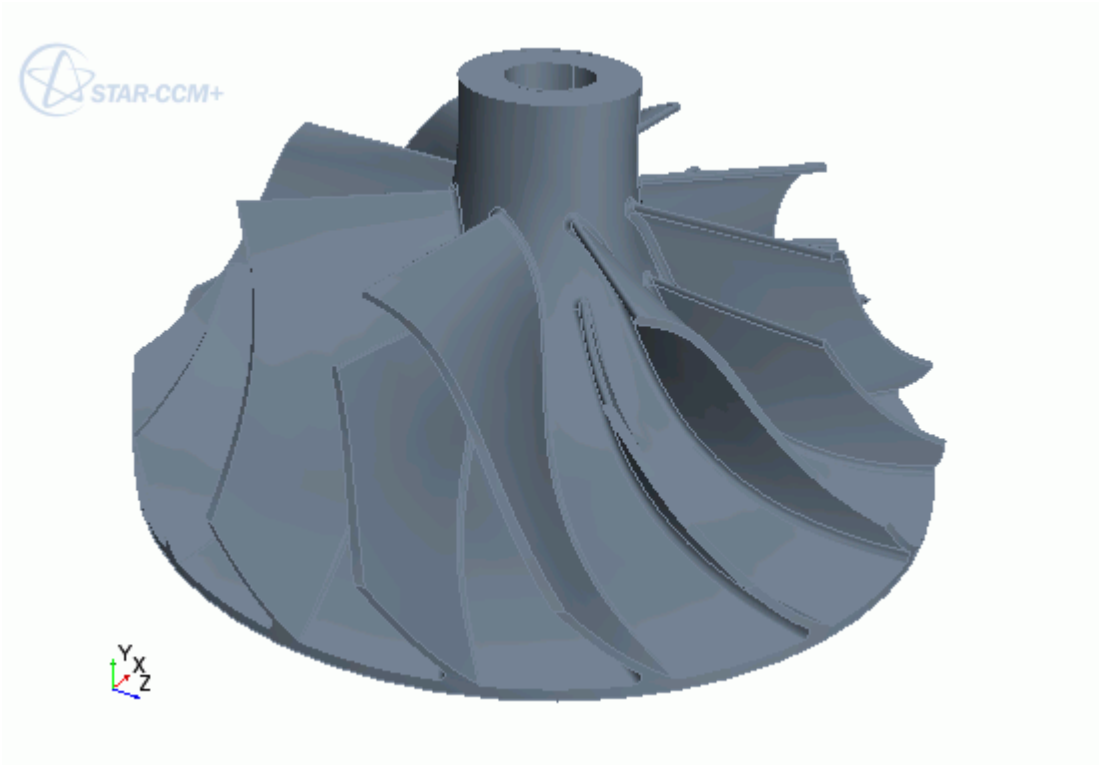
The turbo slicing tool extracts the primary gas path from the extracted fluid volume of a turbomachinery blade row.

With the inputs that you provide, this tool generates a neutral surface (a three-dimensional sheet body) through a single blade that extends from the blade edges to the inlet, outlet, hub, and shroud. You can then use this generated surface to slice the fluid volume and isolate the selected blade for analysis.



When simulating a single blade, you can use a much finer mesh which affords you a more detailed look at the flow near the blade.

Typically, you start with the raw CAD geometry of the turbomachinery. The Turbo Slicing tool is suited for axial geometries (axial-flow turbomachinery, propellers, fans), as well as radial geometries (radial compressors, radial diffusers).



Your aim is to extract the fluid domain around a single blade. For axial geometry, to achieve this, you can create a fluid domain for the entire geometry, then proceed to focusing on a single blade. Although the Turbo

Slicing tool is designed to operate on one blade row at a time, you can use the tool multiple times to create slice features for multi-blade row machines.

For radial geometry that have different types of blades, such as full blades and splitter blades, you can create a periodic fluid domain by including these two types of blades in the fluid domain. If the radial geometry only contains one type of blade, then you can create a periodic fluid domain that contains a single blade.

Contents:

Extracting the Overall Fluid Volume
Generating the Neutral Surface
The Turbo Slicing Surface Panel
Extracting the Final Fluid Volume

Extracting the Overall Fluid Volume

The Turbo Slicing Surface tool operates on a fluid volume that you extract from the geometry of the turbomachinery.

In 3D-CAD, there are multiple methods you can use to create the fluid volume, including the Extract Volume feature and Boolean operations. The following example uses the Subtract feature in 3D-CAD.

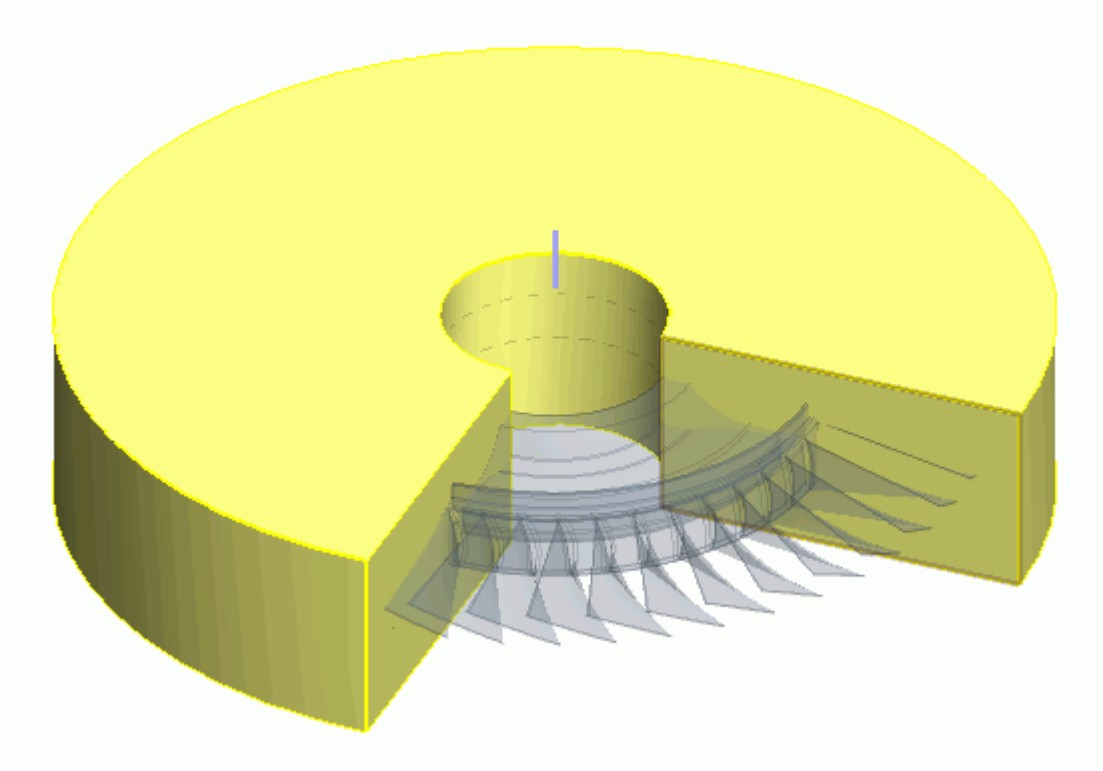
To create a fluid volume for the turbomachinery geometry:

1. Launch 3D-CAD and import the turbomachinery geometry.
2. Make sure that the geometry is aligned with the default axes.

   If necessary, use the **Transform** operations to reposition the geometry. See Transforming a Body.
3. Create a sketch on one of the primary sketch planes, ideally on the plane that runs through the center of the geometry.
4. Draw the profile of the overall fluid volume, making sure to align it with the geometry.

   Create the volume so that the boundaries coincide with the extents to which you to extend the neutral surface.

   Note: If your axial geometry does not have a hub, you can mimic one by positioning the fluid volume wall in the place that the hub would occupy. The neutral surface will extend to the axis of rotation.
5. Revolve the sketch 360 degrees about the centerline of the geometry.



6. Set the *Body Interaction* parameter in the *Revolve* panel to **None**.

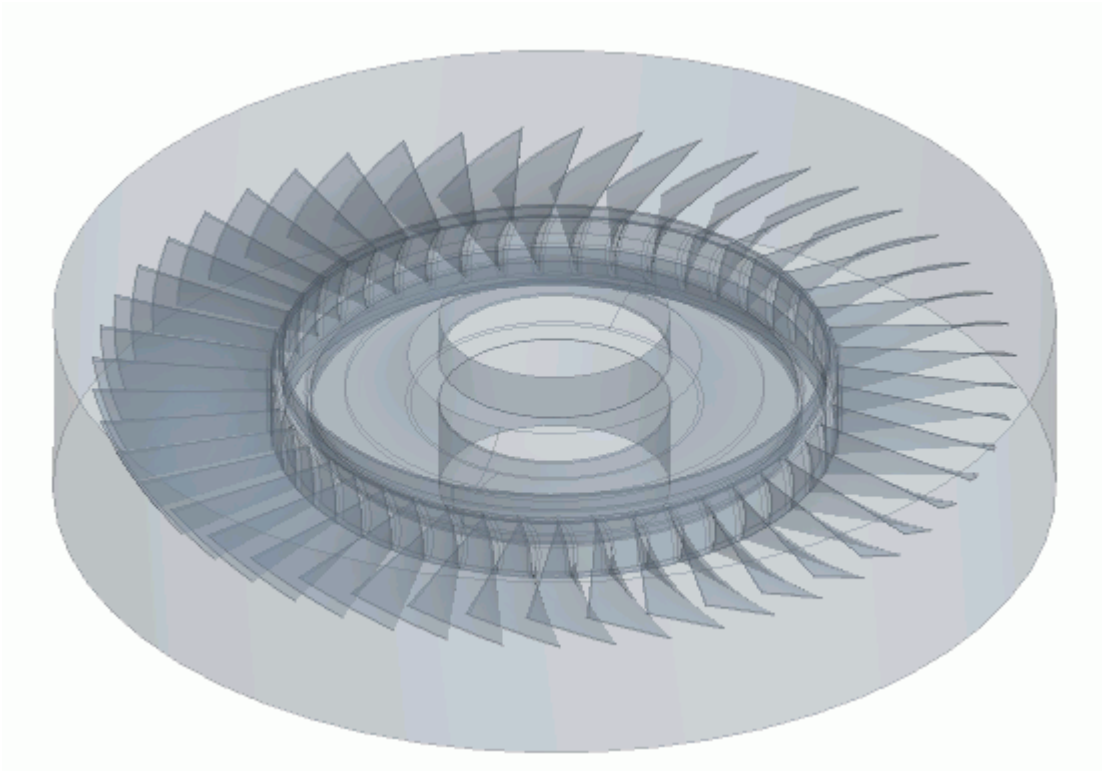
   By default this option is set to **Merge**.

7. Subtract the original geometry from the fluid volume. See Performing Boolean Operations.

   By default, this operation consumes the original geometry. If you want to retain the original geometry, activate the **Keep Tool Bodies** option in the *Subtract Bodies* panel.

8. If you retained the original geometry, hide it.

The fluid volume is complete.



Generating the Neutral Surface

The Turbo Slicing Surface tool generates a neutral surface through a chosen blade in your turbomachinery geometry which represents the primary gas path.

Before using the Turbo Slicing Surface tool, make sure that your geometry has a suitable fluid domain. See Extracting the Overall Fluid Volume.

Note: The Turbo Slicing Tool focuses on generating the neutral surface and does not take into consideration the direction of flow. Radial geometry can consist of compressors or turbines, depending on the direction of the flow through the geometry. When defining the inlet and outlet faces for a radial geometry, define these faces so that they are consistent with a radial compressor, as shown in the example screenshots.

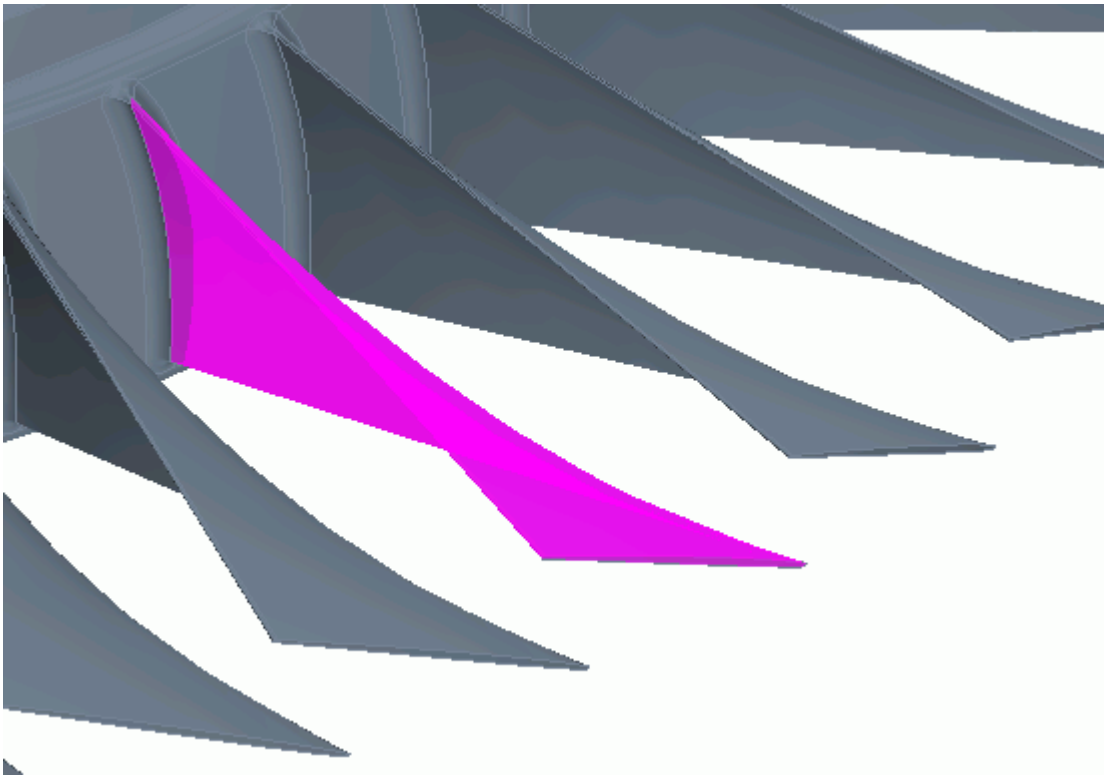
To extract the blade passage fluid volume:

1. Place the (**Create Turbo Slicing Surface**) icon on the toolbar. See Customizing a Toolbar.
2. Click (**Create Turbo Slicing Surface**).
   The *Turbo Slicing Surface* panel appears.
3. Specify the type of geometry:
   - For axial geometries, set *Blade Type* to **Axial**.
   - For radial geometries, set *Blade Type* to **Radial** or **Radial Diffuser** depending on the type of geometry you have.
4. In the *3D-CAD View* scene, select the faces that make up a single blade.

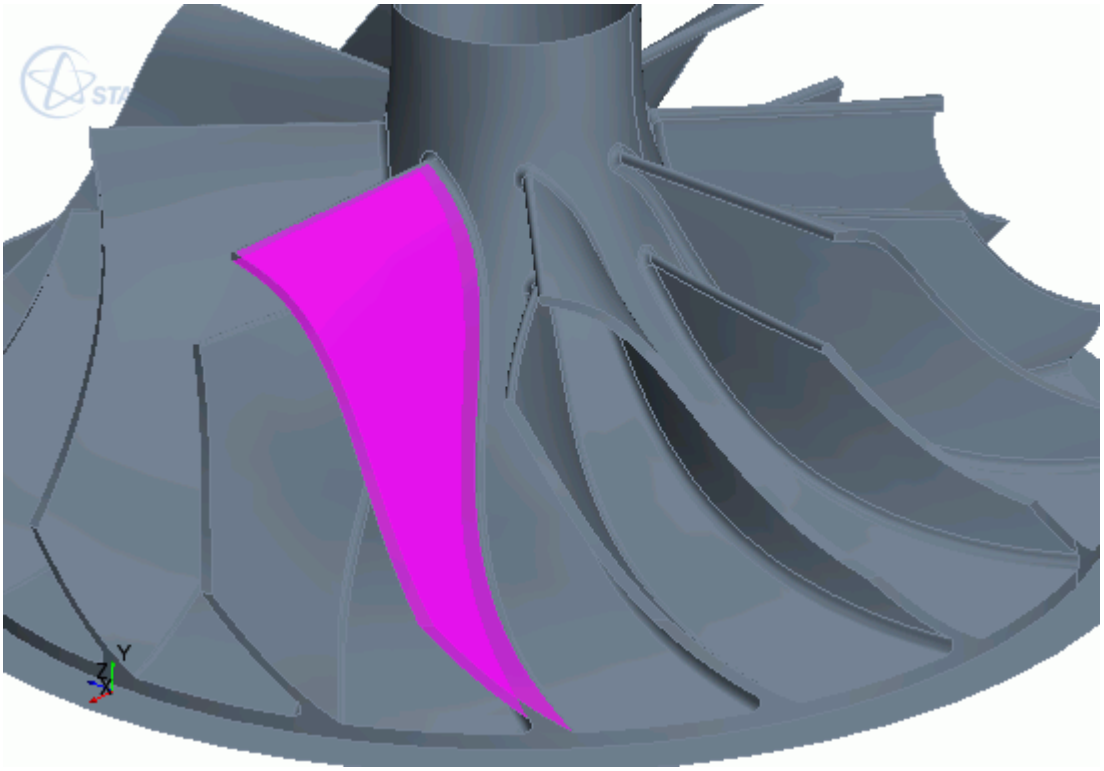
To help with the selection, you can hide any obstructing faces. See Hiding Faces.

In some cases, small patches can cause problems. In these cases, omit the small patches from the blade selection. If patches exist, it is not necessary to fully capture the blade leading and trailing edge faces as the neutral surface is an approximate mid-surface.

- For axial geometries:



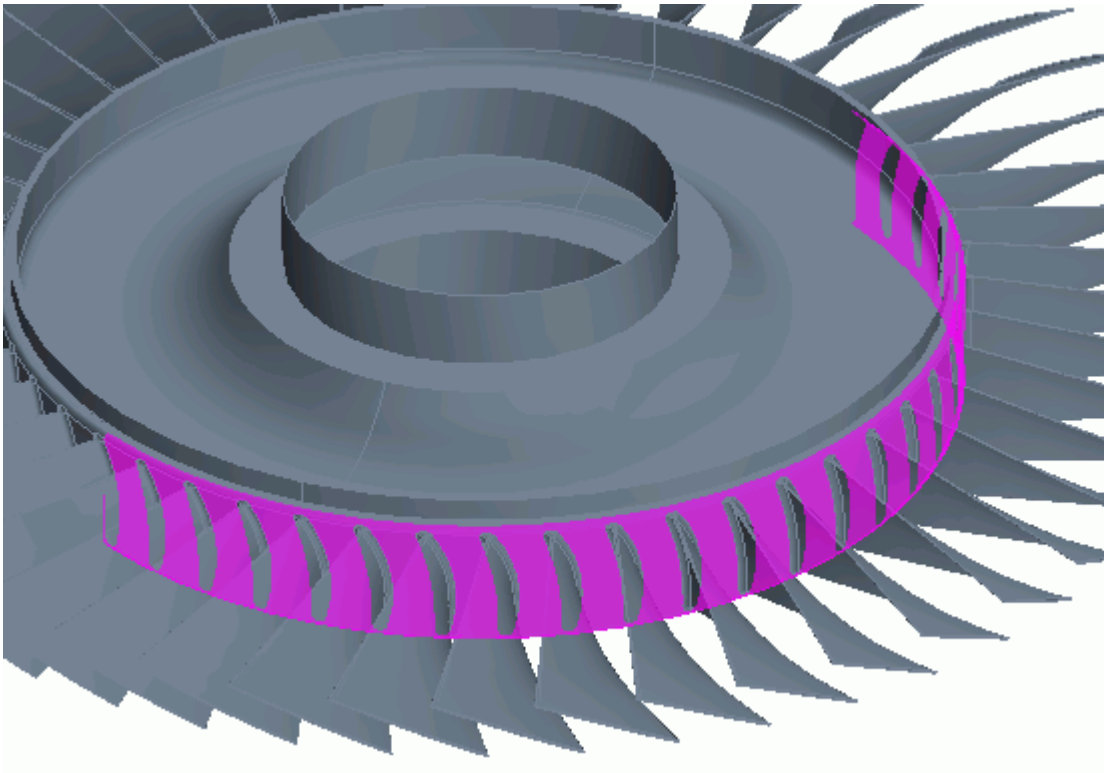
- For radial geometries:



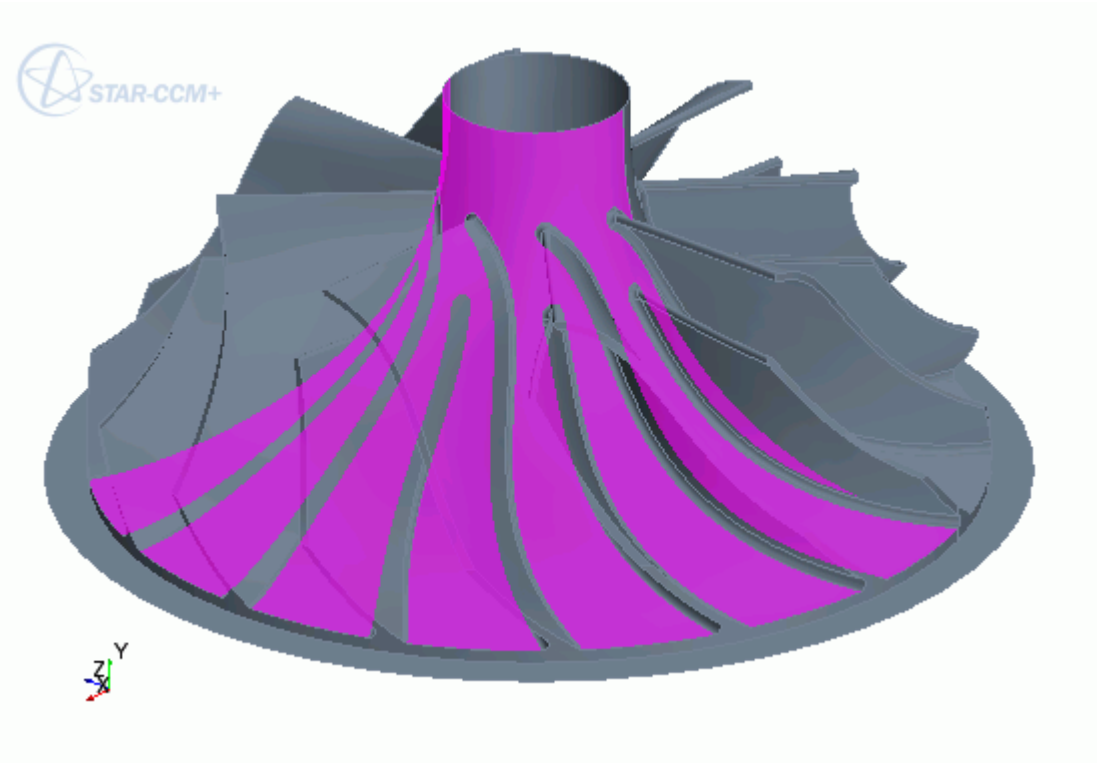
5. Click inside the *Hub Faces* group box to activate it and in the *3D-CAD View* scene, select the faces that make up the hub.

   You are advised to only select the hub faces that envelop the selected blade, and that define the hub surface of revolution. It is not necessary to select all the faces that make up the hub. Do not select faces that are coincident with upstream or downstream boundaries.

   - For axial geometries:



- For radial geometries:

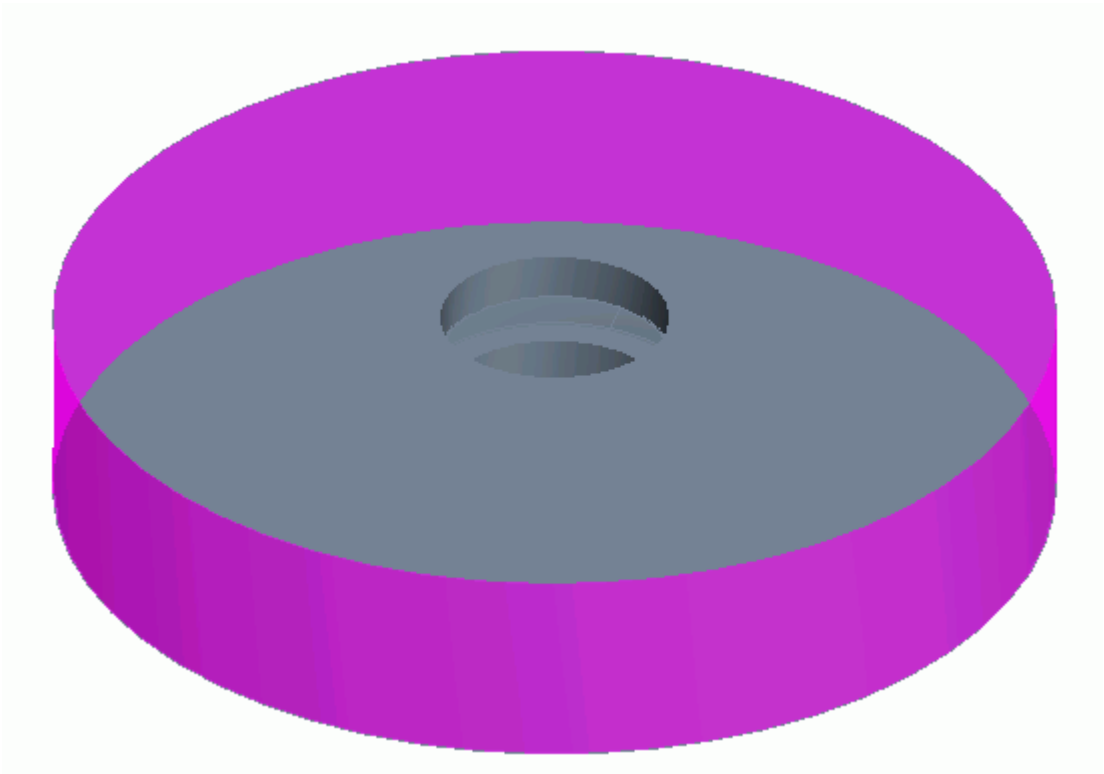


6. Click inside the *Shroud Faces* group box to activate it and in the *3D-CAD View* scene, select the faces that make up the shroud.

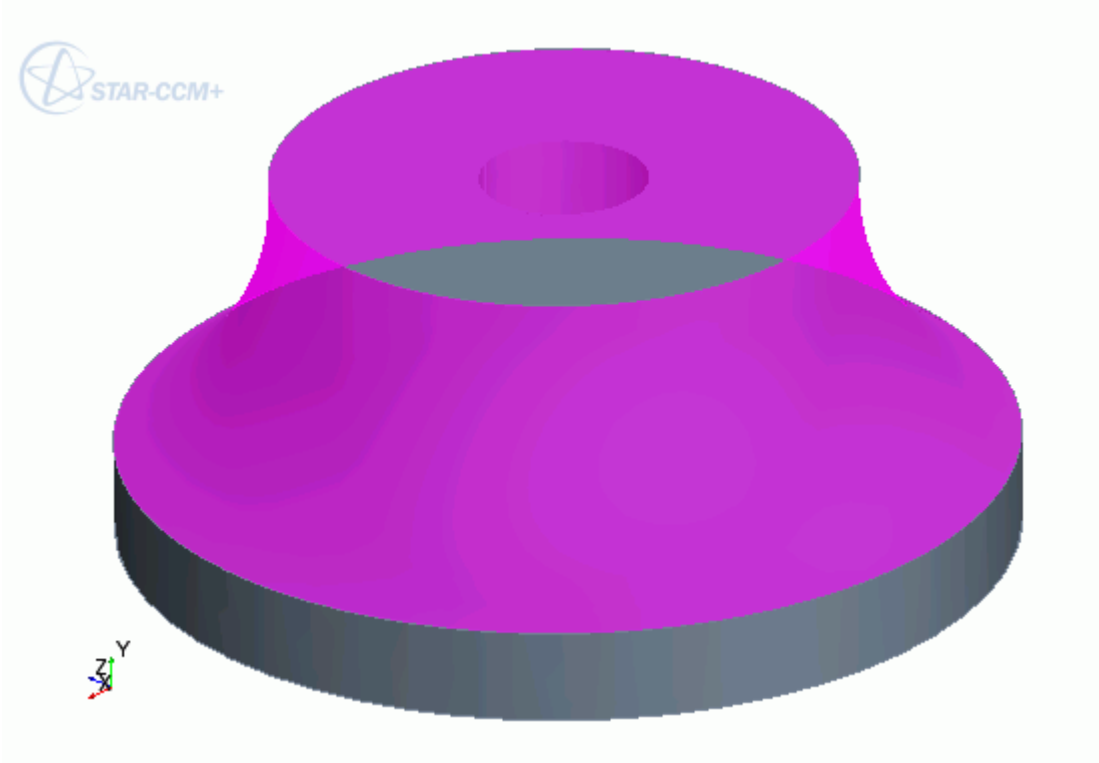
In cases without a shroud, for example open propellers, the shroud faces are the outer boundary of the extracted fluid domain.

If you hid any faces on the domain, restore them by right-clicking on a blank area in the scene and selecting **Restore Hidden Faces**. In many cases, the shroud is the domain wall.

- For axial geometries:



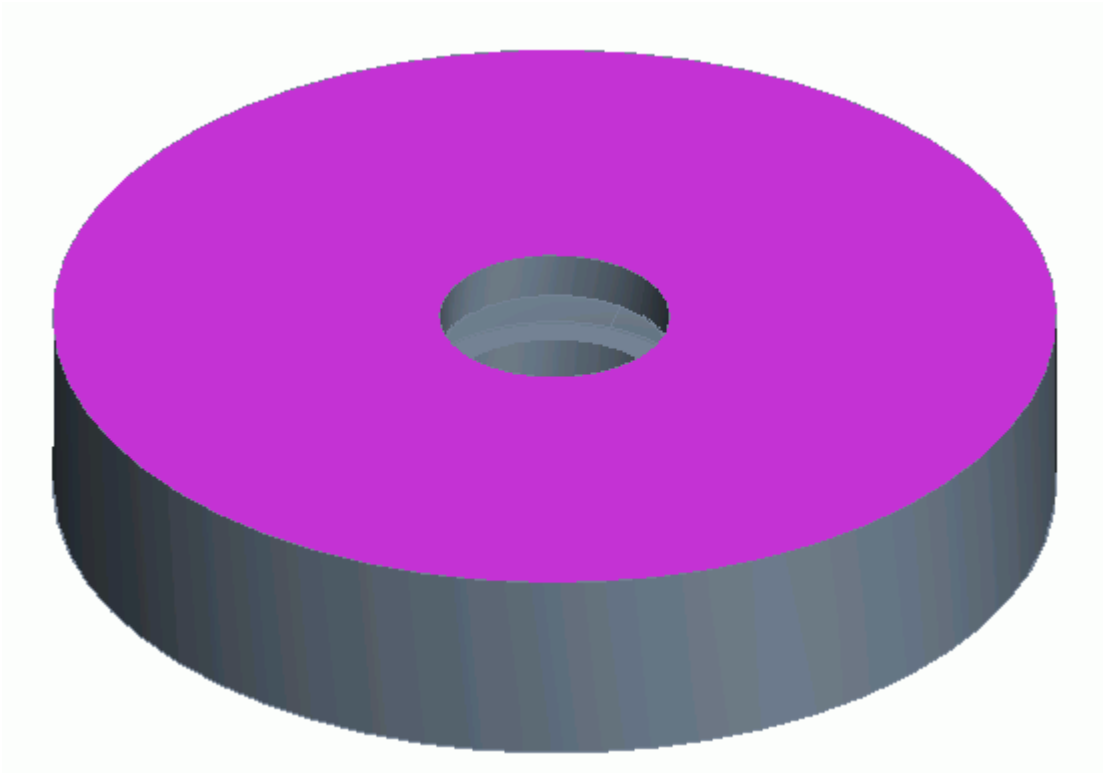
- For radial geometries:



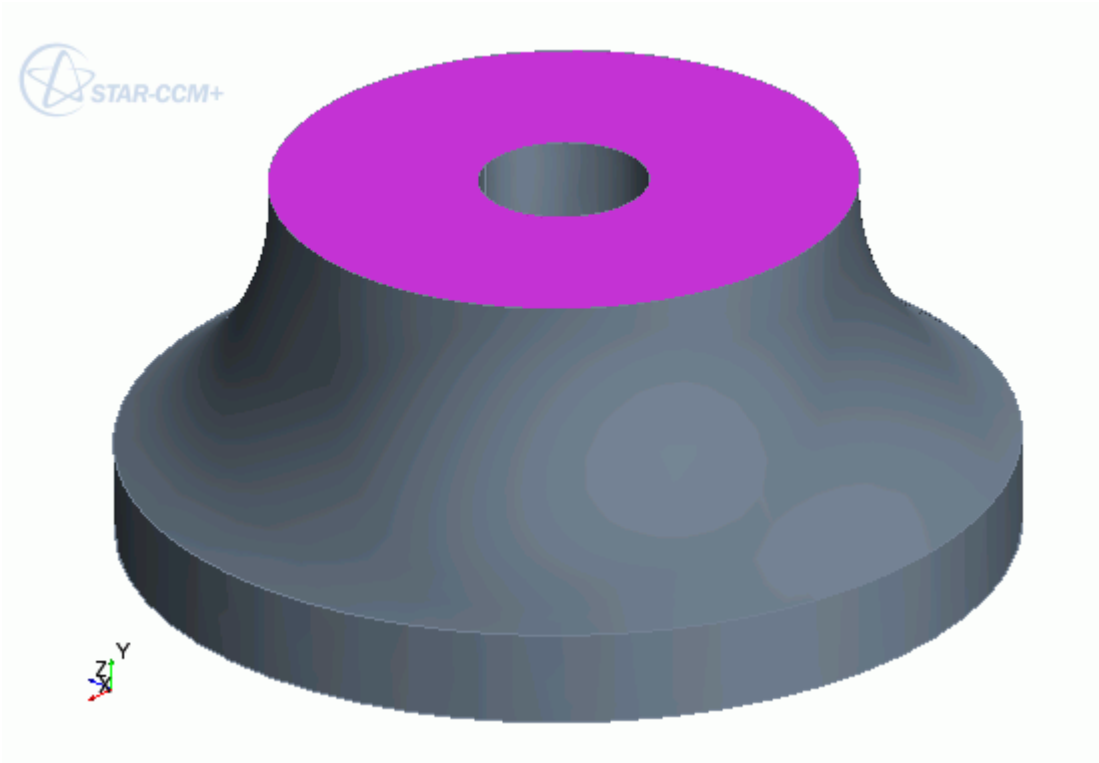
7. Click inside the *Inlet Faces* group box to activate it and in the *3D-CAD View* scene, select the faces that correspond to the inlet.

   The inlet must be planar.

   - For axial geometries:

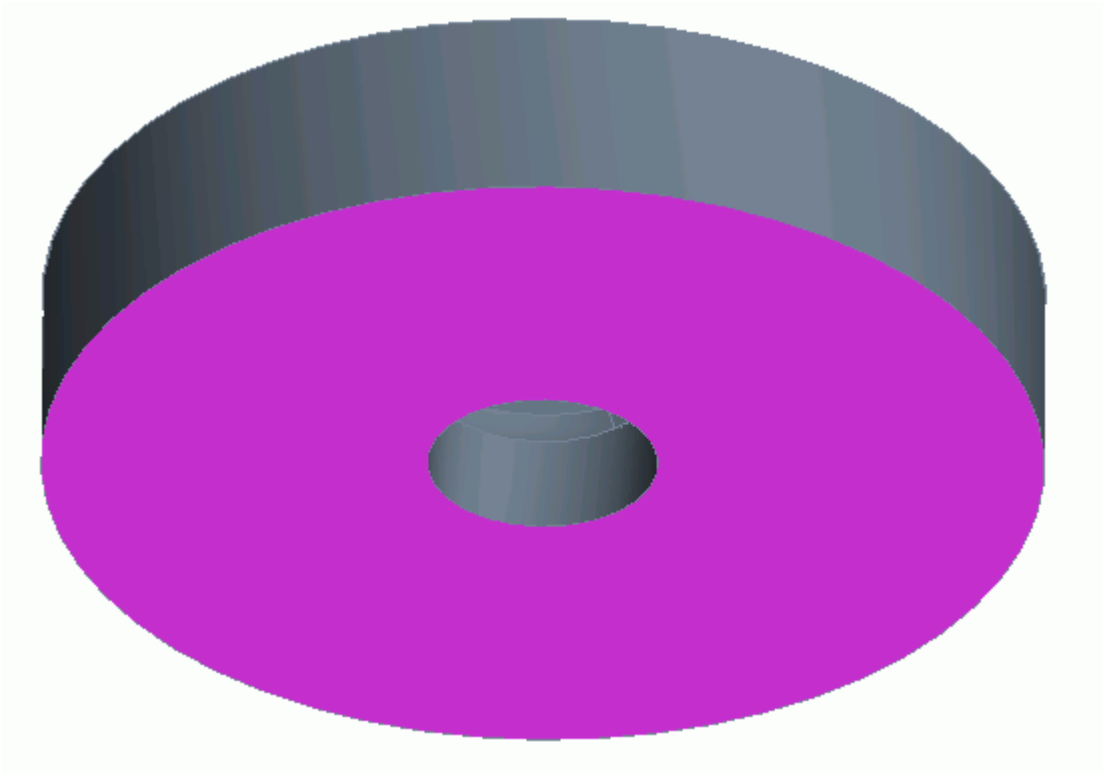


- For radial geometries:

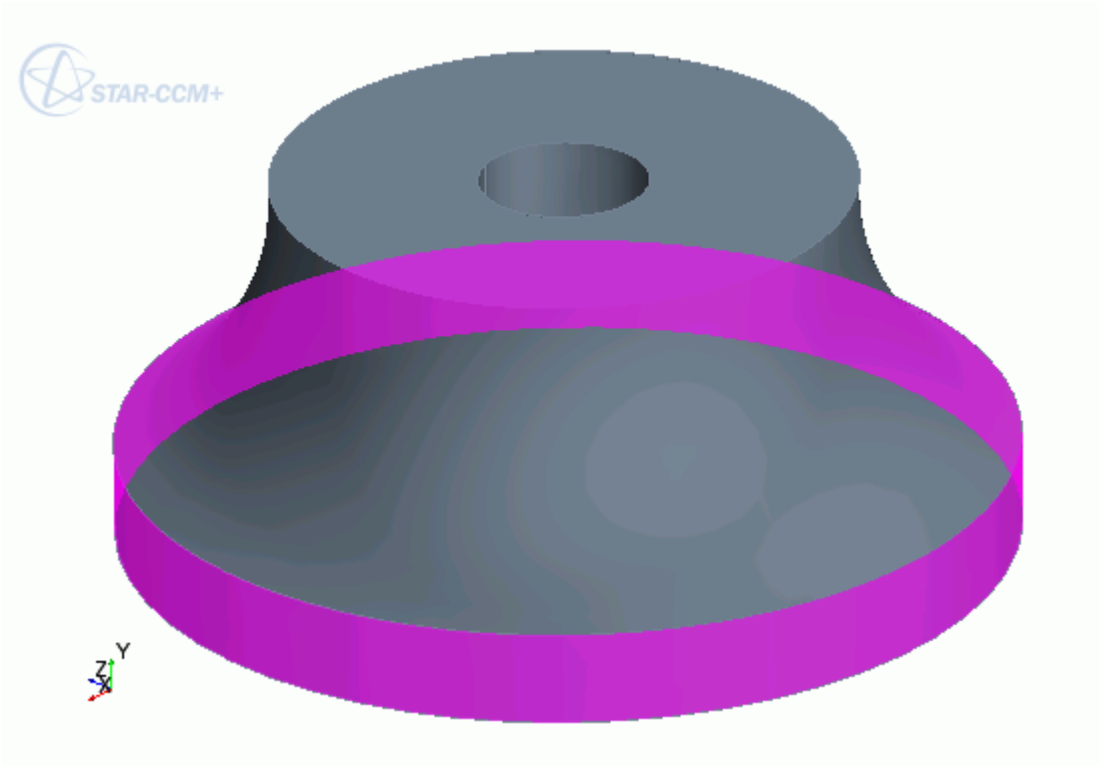


8. Click inside the *Outlet Faces* group box to activate it and in the *3D-CAD View* scene, select the faces that correspond to the outlet.

- For axial geometries, the outlet must be planar. It is also good practice for the outlet to be parallel to te inlet.

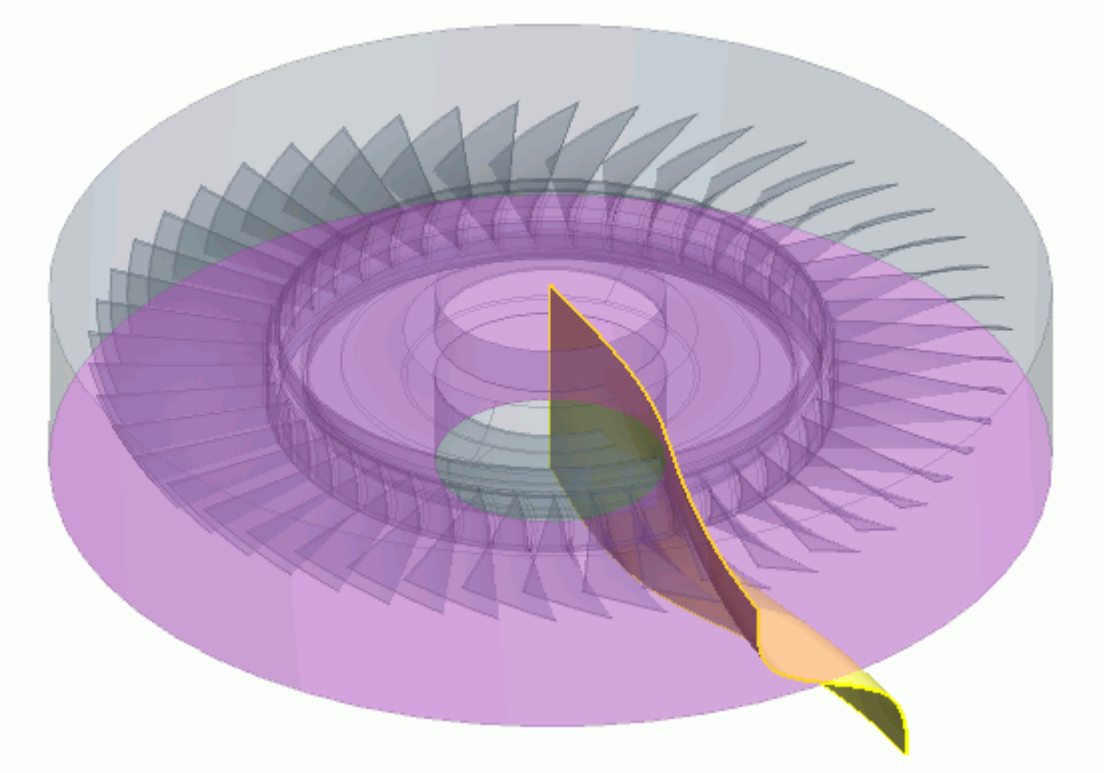


- For radial geometries, the outlet is expected to be perpendicular to the inlet.

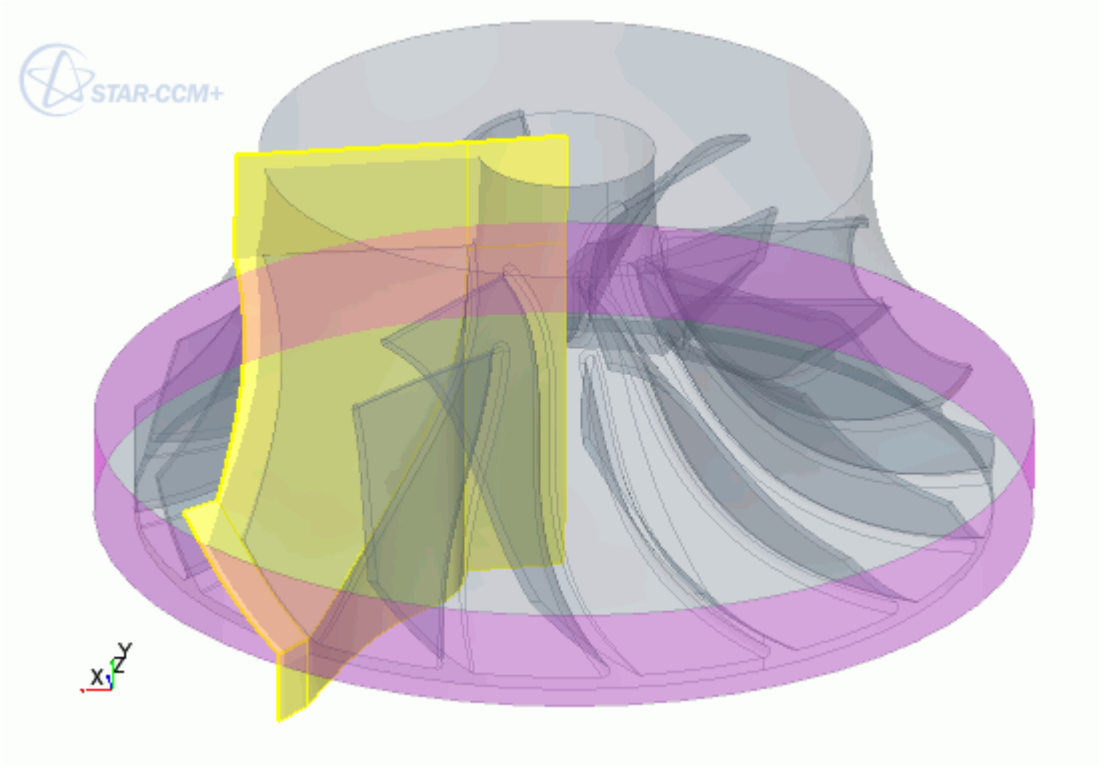


If the neutral surface can be calculated, a preview appears.

- For axial geometries:



- For radial geometries:

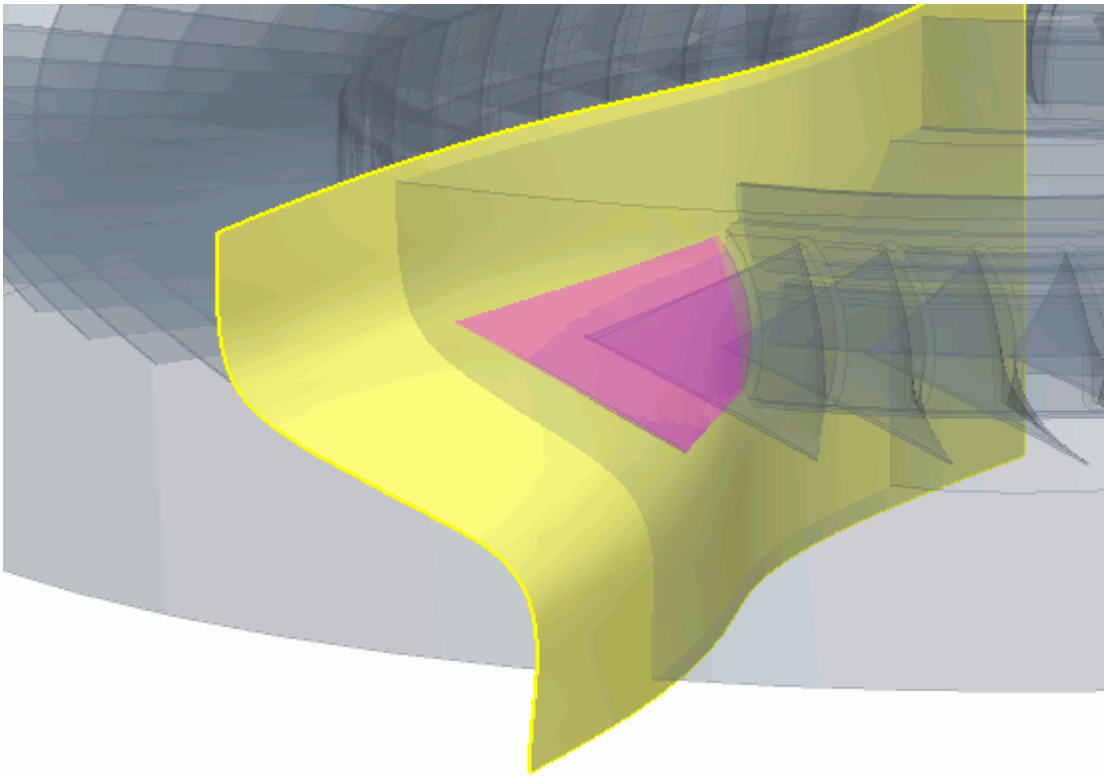


9. Set the turbo slicing surface parameters so that the neutral surface fits through the center of the blade. Also make sure that the areas where the neutral surface meet the inlet and outlet are normal to these boundaries.

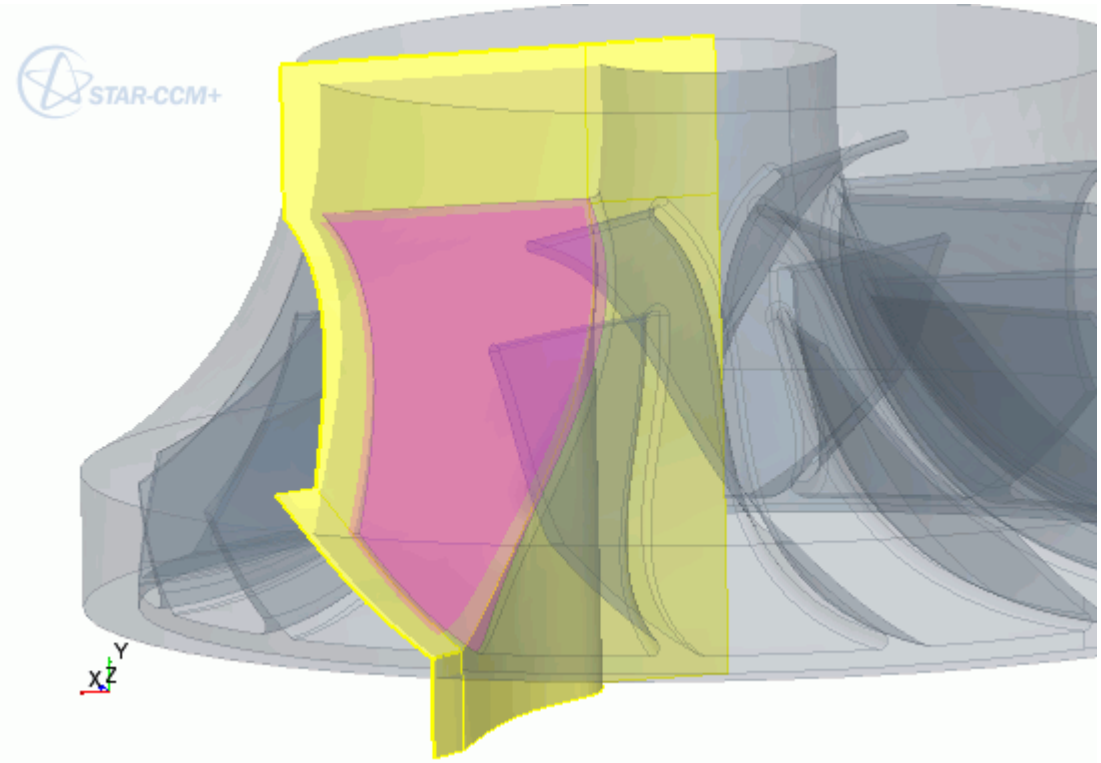
   The surface must fully intersect all domain walls so that you can use it to slice the single blade passage from the overall fluid domain.

   See Turbo Slicing Surface Panel for a definition of the parameters.

   - For axial geometries:



- For radial geometries:

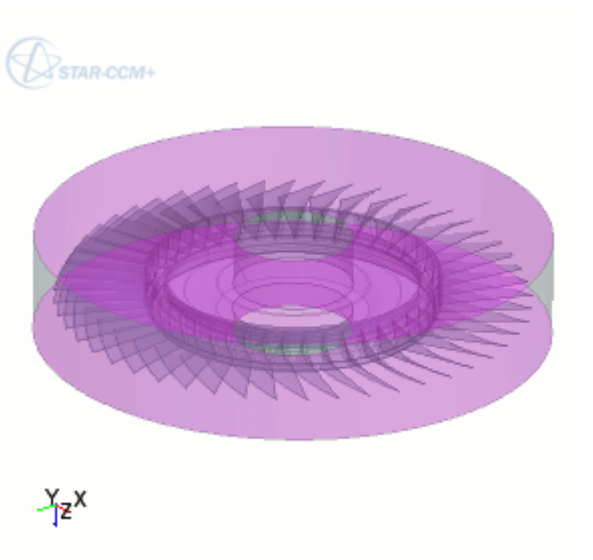
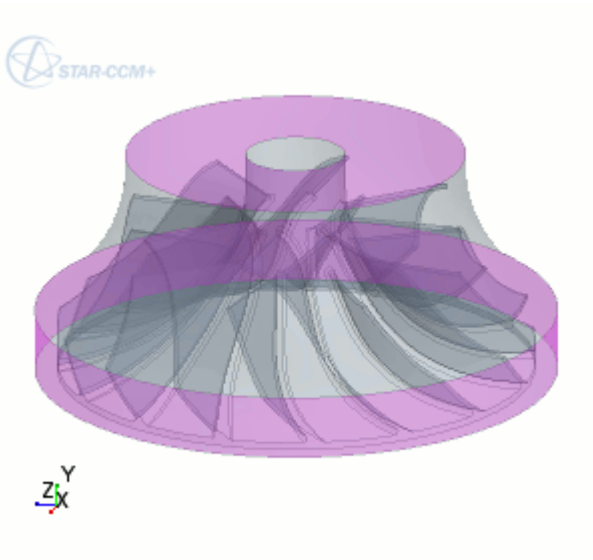


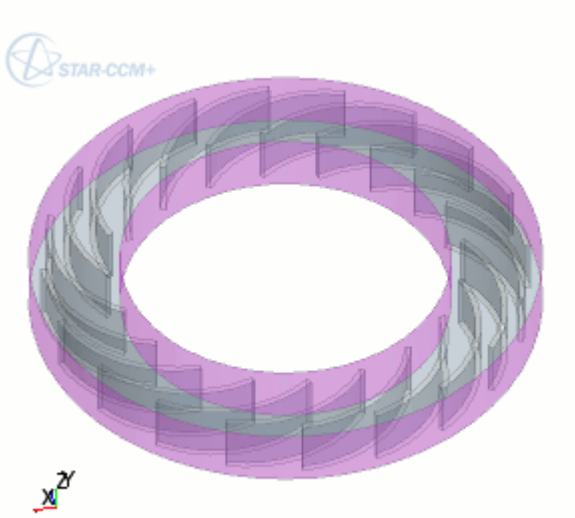
10. Click **OK**.

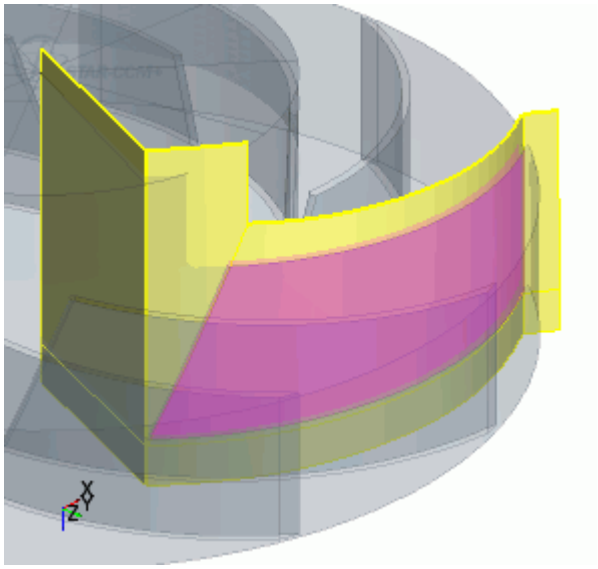
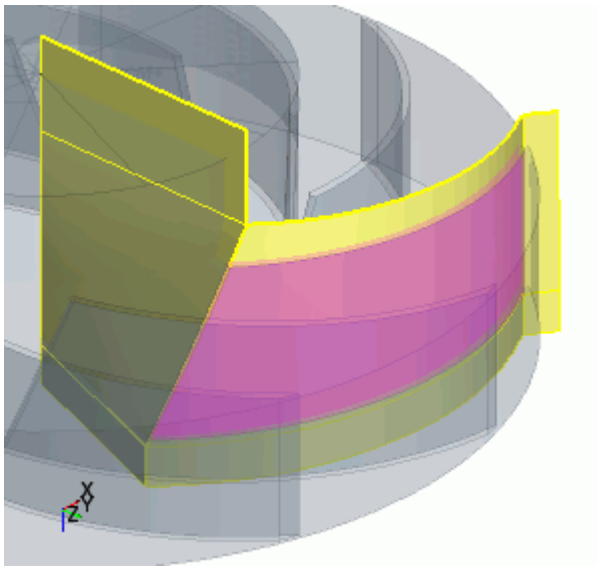
The neutral surface is created and appears as a sheet body under the **Bodies** node of the feature tree.

The Turbo Slicing Surface Panel

The options to control the neutral surface parameters are found in this panel.

Turbo Type: Blade Type	Specifies the type of geometry you want to generate the neutral surface on.	
	Axial	The inlet and outlet faces on the geometry are parallel to each other and perpendicular to the axis of rotation. The inlet and outlet faces must be planar. 
	Radial	The inlet and outlet faces on the geometry are perpendicular to each other. The inlet face is typically planar and normal to the axis of revolution. 

	Radial Diffuser	The inlet and outlet faces on the geometry are concentric or almost concentric. The inlet and outlet faces must have the same, or almost the same, axis of revolution. 
Blade Faces	Specifies the faces that make up the blade through which the neutral surface must pass. The neutral surface is created based on the selected blade surfaces. As a guideline, select all the faces that form a loop around the blade surfaces. If the blades contain cooling features, you can ignore these. If the blade contains small patches, you are advised to omit these from the blade selection, as they can cause problems.	
Hub Faces	Specifies the faces that make up the hub. Select the faces that surround the blade of interest.	
Shroud Faces	Specifies the faces that make up the shroud. Select the faces that surround the blade of interest.	
Inlet faces	Specifies the faces that make up the inlet.	
Outlet Faces	Specifies the faces that make up the outlet.	
Slicing Surface Parameter: Surface Resolution	Specifies the spanwise (hub to shroud) resolution of the grid that is used to define the neutral surface.	
Leading Edge/ Inlet Parameter: Transition Shift	Specifies the transition of the neutral surface from the inlet to the leading edge of the selected blade. The neutral surface is constrained to be normal to the inlet plane. This parameter defines the non-dimensional location between the inlet and the blade leading edge where the sheet transitions to satisfy the normal constraint. The valid range is from 0.0 to 1.0, from the leading edge to the inlet. The smaller the value, the sharper the transition.	
Leading Edge/ Inlet Parameters:	Specifies how the neutral surface is generated on the leading edge of the selected blade. Select the option that returns the best neutral surface.	

Leading Edge Type	**Leading Edge to Inlet**	The neutral surface extends from the leading edge of the selected blade to the inlet surface. 
	Leading Edge to Axis	The neutral surface extends from the leading edge of the selected blade to the axis of revolution. 
Trailing Edge/ Outlet Parameter: Transition Shift	Specifies the transition of the neutral surface from the outlet to the trailing edge of the selected blade. The neutral surface is constrained to be normal to the outlet plane. This parameter defines the non-dimensional location between the outlet and the trailing edge of the blade where the sheet transitions to satisfy the normal constraint. The valid range is from 0.0 to 1.0, from the trailing edge to the outlet. The smaller the value, the sharper the transition.	

Extracting the Final Fluid Volume

After generating the neutral surface, you can use this to extract the final fluid volume for your analysis.

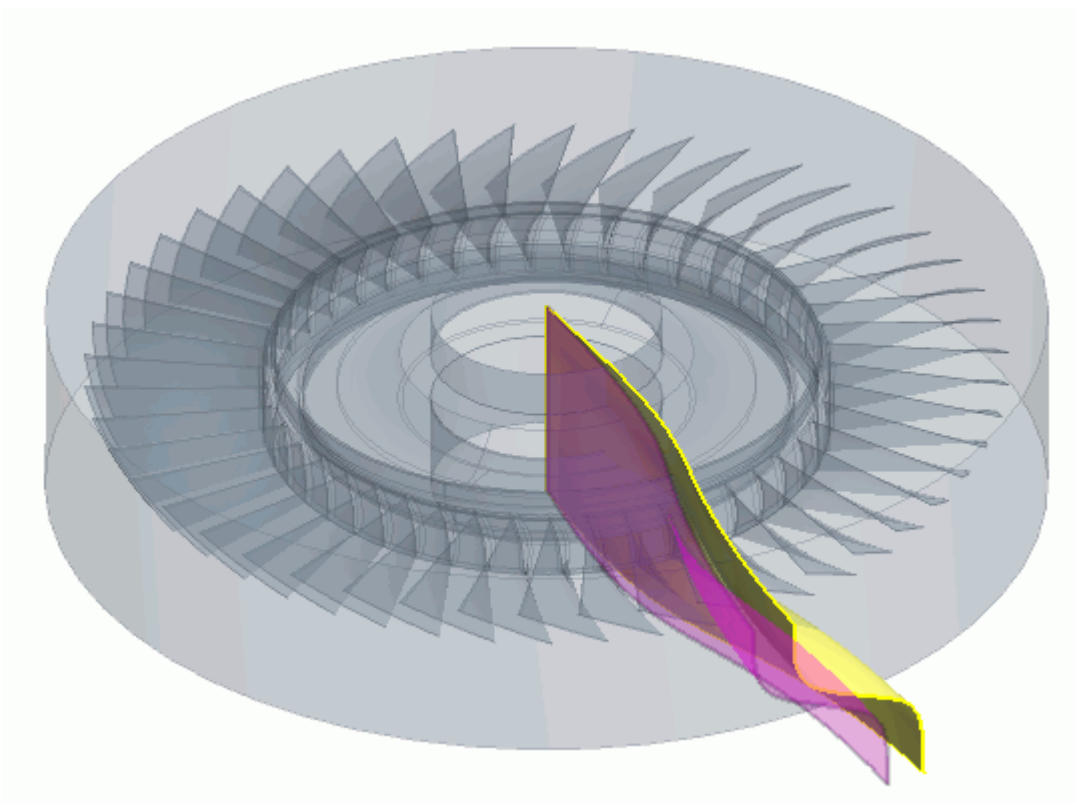
Use the generated neutral surface to slice the fluid domain:

1. Rotate the neutral surface in one direction: See Transforming a Body.
   - For axial geometries, rotate the neutral surface so that it is equally spaced between two blades.

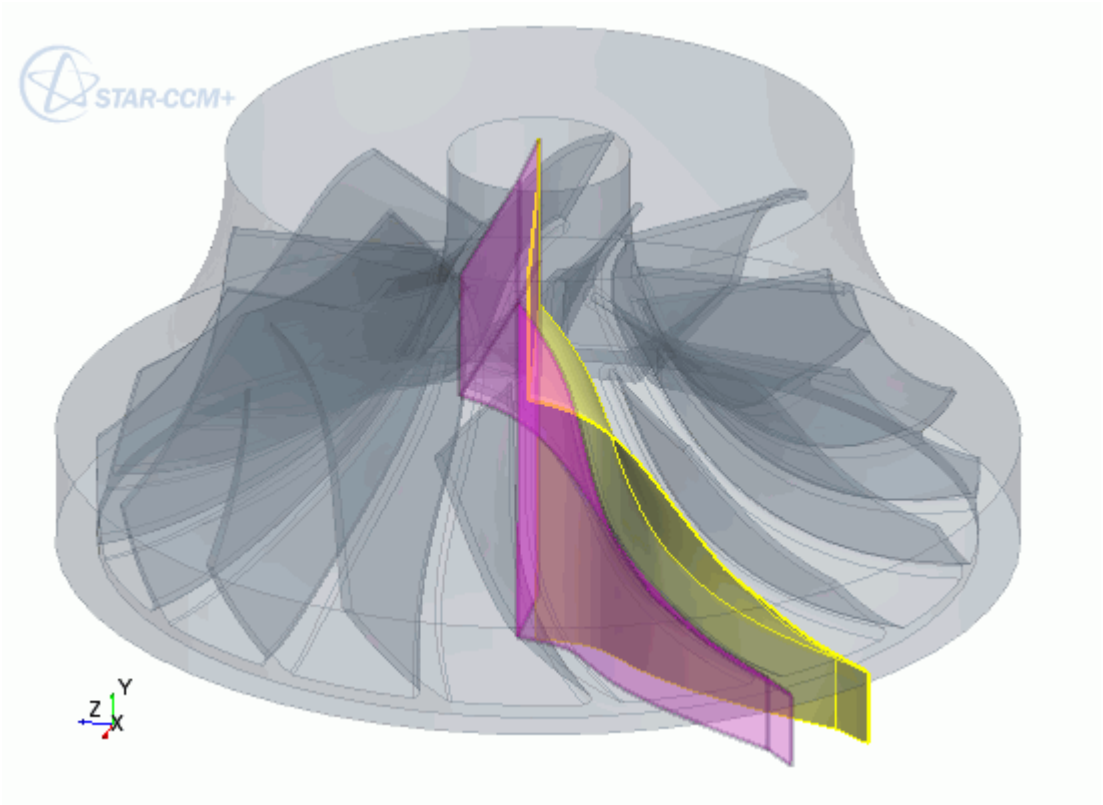
     A formula to calculate the exact amount to rotate the neutral surface is provided below:

     $$\frac{(360/N)}{2}$$

     where N is the total number of blades in your geometry.

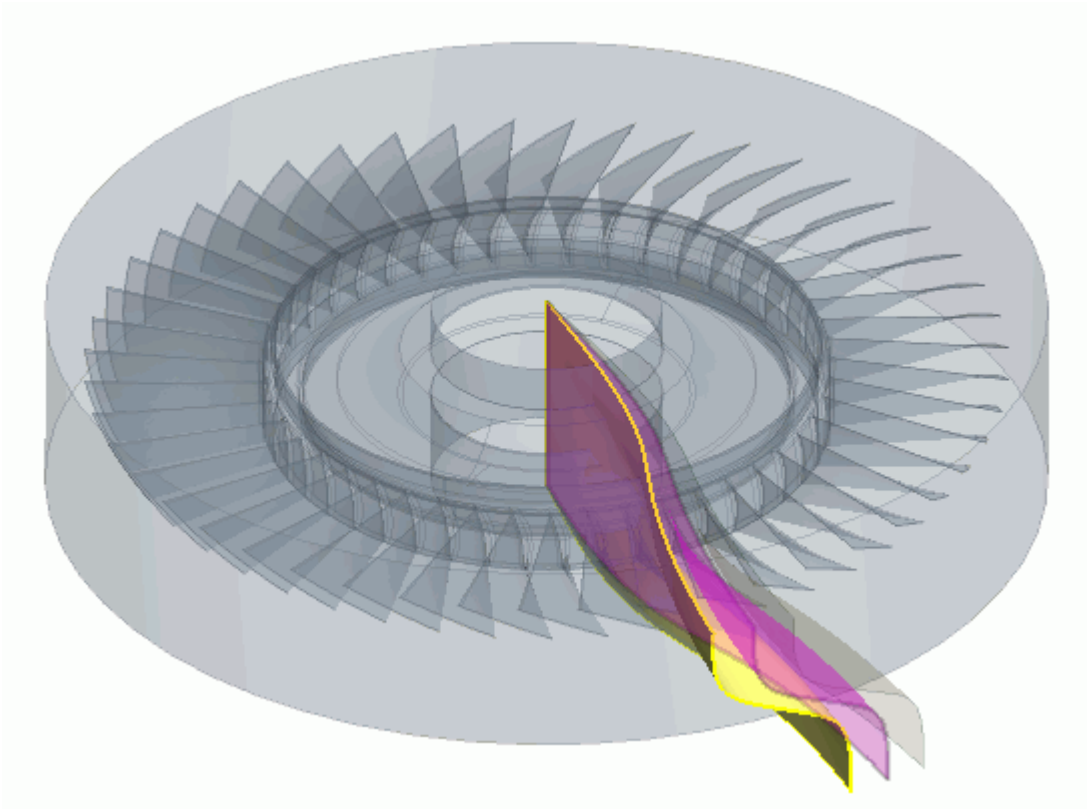


   - For radial geometries with different types of blade, rotate the neutral surface so that it is equally spaced between the full blade and the splitter blade.

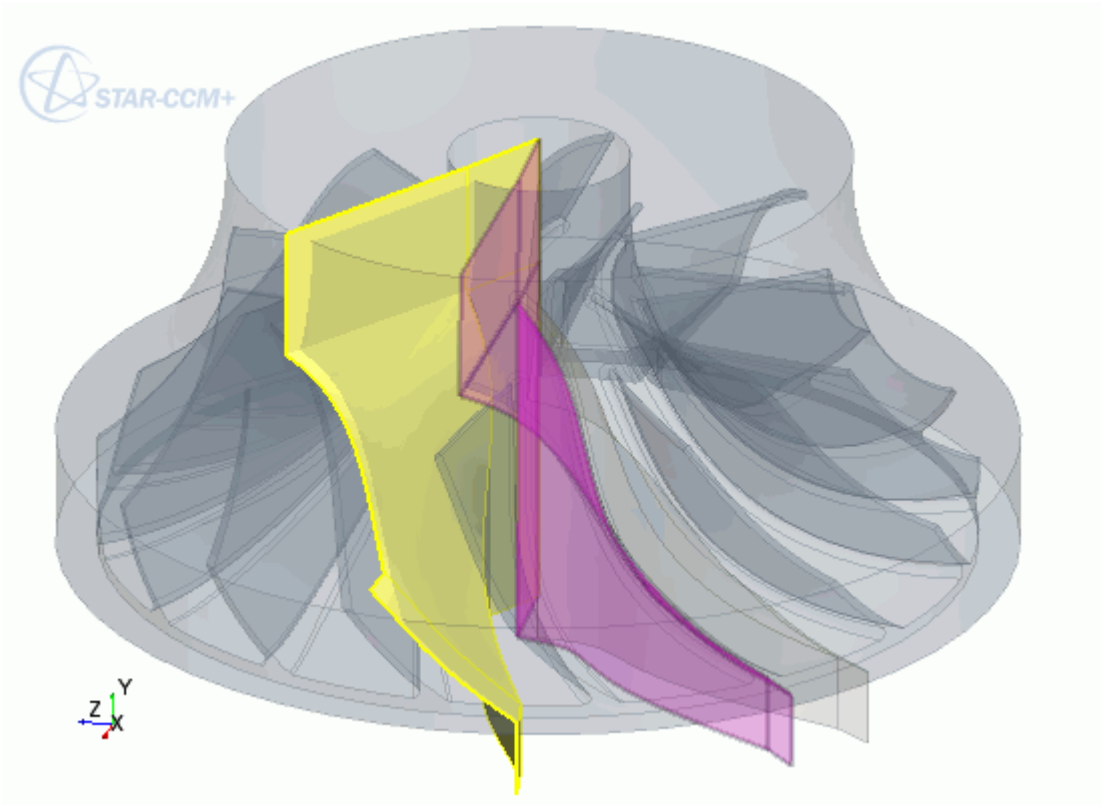


Set *Action* to **Copy + Rotate Bodies**. The default option rotates the body, however you want to keep the original position for when you rotate the neutral surface in the opposite direction.

2. Rotate the original neutral surface in the opposite direction:
   - For axial geometries, repeat the previous step and rotate the original neutral surface in the opposite direction by the same amount.



- For radial geometries, rotate the original neutral surface in the opposite direction so that it is equally spaced in between the next full blade and the splitter blade.



Leave *Action* set to **Rotate Bodies**.

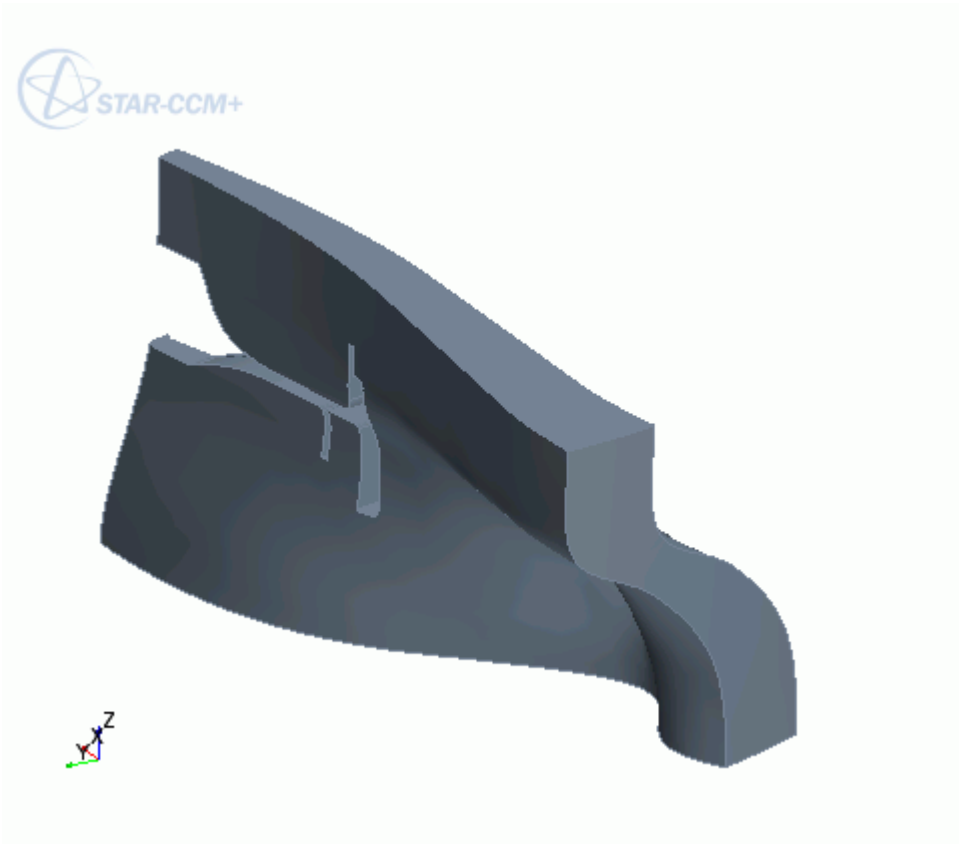
3. Slice the fluid domain using the two neutral surfaces. See Performing Boolean Operations.

   Set *Slice by:* to **Sheet Body**.

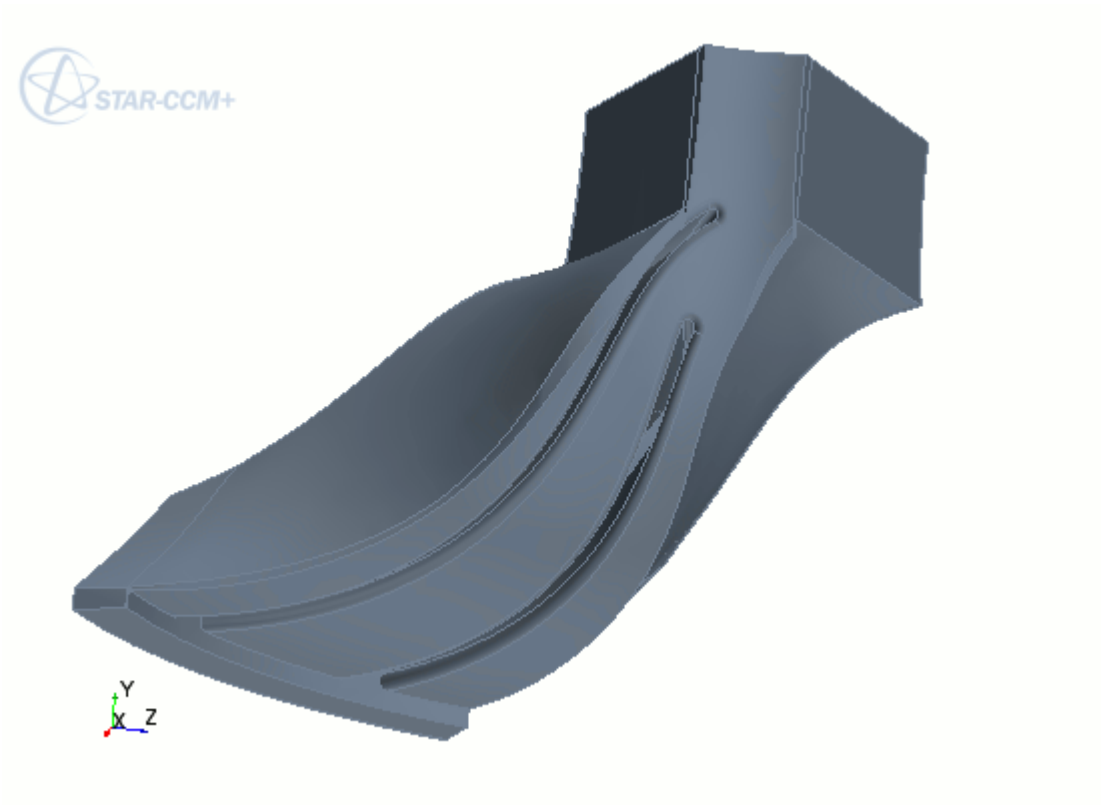
4. Use the **Select From Scene** tool to select the two neutral surfaces from the scene.
5. Delete the two neutral surfaces as well as the original fluid domain - these items are no longer required.

The extracted fluid domain is complete.

The final fluid domain for the axial geometry is shown below.



The final fluid domain for the radial geometry is shown below.


STAR-CCM+
Y
X
Z

Importing Geometry

One of the possible starting points for the meshing process is to import surface data.

Surface data exists in many different formats. To import geometry into STAR-CCM+, this data must be translated into a format that STAR-CCM+ understands using a CAD translator.

There are three different types of surface data that you can import into STAR-CCM+:

- Surface meshes
- Neutral format data
- Native CAD data

Additionally, you can import XML data files that contain assembly data and references to component files based on the above formats.

Note: To import native CAD files, you require an additional CAD Exchange Reader license. To import JTOpen files, you require an additional JTOpen Reader license.

For each of the above three file types, the option to import the data as regions or to import the data as parts is provided during the import process. Progress that is made during the translation/import process can be reviewed and the resulting imported surface can be checked for errors if necessary. If problems are found, they can generally be fixed using the surface repair tools.

If the surface data resulted in regions and boundaries being created then they can be scaled, rotated, translated, or reflected in order to position them properly relative to one another. Surface data that is imported as geometry parts can similarly be manipulated before being assigned to regions.

Additional region-based manipulations are possible in order to make them conform with the requirements for multi-region meshing.

Surface data can be imported to complement existing volume meshes so that only part of the simulation domain requires preparation for volume meshing.

Importing Geometry Using the CAD Client Translators

The CAD client translators can offer a more robust method of importing geometry into STAR-CCM+. There are two methods of importing geometry into STAR-CCM+ using the CAD client translators:

- Import geometry directly from the CAD package using the CAD clients
- Import geometry using the CAD client translators from within STAR-CCM+.

Importing geometry directly from the CAD package maintains a link between the CAD package and STAR-CCM+, and allows you to transfer design data such as design parameters, coordinate systems, and reference planes, in addition to the geometry model.

The following CAD packages support this functionality:

- STAR-CAT5
- STAR-NX
- STAR-Inventor
- STAR-Creo

Note: To use the CAD client translators, you require an additional CAD client license.

Due to the nature of the different translators, geometry that you import using the CAD Client translators may differ slightly to geometry that you import using the CAD Exchange translator. For example, for native CATIA v5 files, there is a chance that the surface name or surface topology may be different depending on what translator you use to import into STAR-CCM+. If this occurs, use the **Define in Work Object** option on the main element inside CATIA and save the geometry. Import this saved geometry into STAR-CCM+ using the CAD Exchange translator.

Contents:

What Is a Surface Mesh?
What Is Neutral Format Data?
What Is Native CAD Data?
What Is XML Data?
CAD Packages Support
Importing Surface Data into STAR-CCM+
Importing Surfaces Using an XML File
Importing Multiple Surface Data Files
Reviewing the Surface Data Import Report
Using Surface Data Import Representations
Checking the Surface Validity

What Is a Surface Mesh?

A surface mesh is a discrete representation of the geometry of the individual regions that are used for the volume mesh generation.

It is made up of faces (triangles) and vertices. Additionally, feature curves are included in surfaces that are created in some pre-processing packages. Internal feature surfaces such as baffles or interfaces are allowed.

The ideal surface mesh has the following properties:

- Closed - Contains no free edges or mismatches
- Manifold - Edges are shared between no more than two triangles
- Non-intersecting - The surface does not self-intersect

The quality of triangle faces is also important. The ideal surface contains:

- Equilateral sized triangles
- Gradual variation in triangle size from one to the next
- No sharp angles, surface folds, or other such characteristics within the triangle proximity sphere

Surface meshes traditionally originate from CAD packages directly or through third-party pre-processing/surfacing software, and are saved in specific formats. Consider carefully the method that you use to generate the surface mesh, as it directly influences the quality of the resulting volume mesh and the effort it takes to get to this step. In cases where a bad quality starting surface is the only option available, the surface wrapper and/or surface remesher tools can be used to improve further the surface as defined above. In cases where

the import surface contains quads or other elements with more than three sides, the face is automatically triangulated.

Contents:

Supported Surface Mesh Input Formats

Supported Surface Mesh Input Formats

The currently allowable import formats that support surfaces are:

- .dbs - pro-STAR surface database mesh file
- .inp - pro-STAR cell/vertex shell input file
- .nas - NASTRAN shell file
- .pat - PATRAN shell file
- .stl - Stereolithography file
- .fro - FELISA front surface file
- .jt - JT Open surface file
- .bif - Medina binary input file
- .3dm - Rhino 3D Model

All of these formats support single or multiple region definitions but always result in a single region definition when geometries are imported from a single file. When importing multiple files, you can choose to generate either a single region or multiple regions. For more information on the various import options, see Importing Surface Data into STAR-CCM+.

If feature curves exist in the original surfacing package, they are useful to maintain. Refer to the table below to see whether feature curves are maintained on import.

A summary of the surface import formats is provided below:

Format Name	File Extension	Binary ASCII	Feature Curves	Multi-Region	Boundary Names
Database	`.dbs`	Binary	Yes	Yes	Yes
Cell/Vert/Inp	`.cell/.vrt/.inp`	Both	Yes	Yes	Yes
NASTRAN	`.nas`, `.bdf`, `.dat`	ASCII	No	Yes	Yes
PATRAN	`.pat`, `.ntl`, `.neu`	ASCII	No	Yes	Yes
STL	`.stl`, `.STL`	Both	No	Yes	No
FELISA	`.fro`	Both	No	Yes	Yes
JT Open	`.jt`	Binary	No	Yes	Yes
MEDINA Binary Input	`.bif`	Binary	No	Yes	Yes

Format Name	File Extension	Binary ASCII	Feature Curves	Multi-Region	Boundary Names
Rhinoceros 3D Model	`.3dm`	Binary	No	Yes	No

pro-STAR Database Format

The pro-STAR database format is native to the pro-STAR and pro-surf pre-processing software by Siemens PLM Software and can store either surface type or volume type meshes (or in combination if desired). STAR-CCM+, however, can only read surface meshes from this file. More than one geometry can be stored in the file, with each being identified from a numerical database ID value and an optional descriptive name. Feature curves that are supported and boundary names that are used in the pre-processing software are preserved when read into STAR-CCM+. Cell table indexes (in pro-STAR) and surface groups (in pro-surf) both produce the equivalent patch definitions when read into STAR-CCM+. The file is binary to preserve file size and is the recommended file format to use when transferring surface data from other Siemens PLM Software pre-processing software programs to STAR-CCM+.

Surfaces that are imported from a database file are always represented as surface mesh parts.

pro-STAR Cell and Vertex Shell Input Format

The pro-STAR cell/vertex shell input format is a combination of three files which together reconstruct the surface geometry details. The three files are:

- `.cel`- Cell file
- `.vrt`- Vertex file
- `.inp`- pro-STAR command input file

The input (`.inp`) file is always ASCII and contains pro-STAR commands that define the cell table entries for the mesh, as well as commands to read in the other two files with the appropriate ASCII or binary format option. Feature curves are supported and boundary names are read from the cell table entries when available. Different cell table indexes in the cell file result in the equivalent patch definitions when read into STAR-CCM+.

Surfaces that are imported from pro-STAR cell/vertex/input files are always represented as surface mesh parts. Additionally, when exporting a surface mesh to the `.inp` format, any spaces within boundary names are replaced with underscores.

NASTRAN Format

The NASTRAN format is an ASCII-based file that is commonly used by a number of CAD packages and pre-processing software. This format supports patch definitions but does not support feature curves. The current importer only recognizes shell (two-dimensional face) type elements, so volume meshes cannot be imported using this format. When preparing the NASTRAN file, it is recommended that the Free Field Format is used to export bulk data.

Surfaces that are imported from Nastran files are always represented as surface mesh parts. Surface names are preserved on import.

PATRAN Format

The PATRAN format is an ASCII-based file that is commonly used by a number of CAD packages and pre-processing software. It does not support feature curves. The current importer only recognizes shell (two-dimensional face) type elements so volume meshes cannot be imported using this format. Patch definitions are supported.

Surfaces that are imported from PATRAN files are always represented as surface mesh parts.

Stereolithography (STL) Format

STL files can be binary or ASCII and are a universal method of transferring surface data using the least possible number of triangles. As such, the quality of the surface data is usually not good and some additional work is required before the geometry is usable. Feature curves are also not supported and all surface patches are always placed into one boundary definition.

Surfaces that are imported from STL files are always represented as surface mesh parts. There is no limit to the file size when importing an STL file.

FELISA Front Surface Format

The Front Surface `.fro` file contains 3D surface triangulation data that is produced by the SURFACE module from the FELISA collection of programs. The SURFACE module accepts geometric curve and surface components as its input (generated by hand or from CAD software). The surface triangulation data can be used as a starting point for surface and volume meshing within STAR-CCM+.

Surfaces that are imported from Front Surface files are always represented as surface mesh parts.

JT Open Surface Format

JT Open is a 3D data format provided by Siemens PLM Software. It is commonly used for product visualization and data exchange. It can contain a combination of tessellated and B-rep data to represent surfaces. STAR-CCM+ supports both JT B-rep and Parasolid (XT) B-rep formats. The STAR-CCM+ import process recognizes units specifications in the JT Open file.

If the JT Open file contains both a tessellation representation and B-rep representation, the *JT Import Options* box appears. From here, you can choose what representation you want to import into STAR-CCM+. The import dialog presents options that are relevant to the type of data that is contained in the file.

To import tessellated data from a JT Open file, you require a `JTOpen` feature license. To import B-rep data from a JT Open file, you require a CAD Exchange license.

When importing in Part Mode, if the JT Open file does not contain units, you can select the desired units from the drop-down menu.

When importing in Regions Mode, you can select the units from the drop down menu. The selected units are applied to all the JT Open files, even if they contain different units. If a hierarchy of parts of parts is being imported in the JT format, and one or more of the parts have units specified, it is essential for all units to be the same.

Tessellated surfaces that are imported from JT Open files are represented as surface mesh parts. B-rep surfaces are represented as CAD parts.

MEDINA Binary Input File

The MEDINA binary input file format is a binary file that is used in the automotive and aerospace industries. Patch definitions, surface, and volume meshes are all supported by this format. XML (MEDINA format) files are not supported with version of STAR-CCM+ running on Windows operating systems.

To import or export a MEDINA file, you require a `medina-import` feature license.

The table below summarizes the element types that can be imported to, and exported from, STAR-CCM+ using the MEDINA format.

Element Type	Import Support	Export Support
Element ID	Yes	No
NODE 1 (Vertex)	Yes	Yes
PROP 464 (Property Data)	Yes	Yes
TRIA 31	Yes	Yes
QUAD 41	Yes	Yes
TETRA 61	Yes	No
PENTA 63	Yes	No
HEXA 65	Yes	No
PYRA 71	Yes	No

Surfaces that are imported from MEDINA files are always represented as surface mesh parts. Surface names are preserved on import.

Rhinoceros 3D Model File

Rhino 3D Model file format was originally designed to accurately transmit NURBS geometries between applications and is commonly used by the marine industry.

Rhino files can contain a combination of tessellated and B-rep data. If the geometry contains both types of data, you must import the geometry twice, changing the Import Type each time.

Surfaces that are imported from Rhino 3D Model files are always represented as surface mesh parts.

What Is Neutral Format Data?

Neutral format data files allow CAD model data to be exchanged between different programs while still maintaining the model integrity.

They contain an exact description of the geometry of the individual regions that are used for the volume mesh generation. Surface meshes, on the other hand, contain a discrete representation. Depending on the format, the neutral data usually contains some kind of boundary representation (B-REP) model which provides a continuous mathematical description of the component, from which a discrete representation can be created. Provided the correct preparation is made, this continuous mathematical description can be used for CAD projection of surface vertices.

Contents:

Supported Neutral Format Input Formats

Supported Neutral Format Input Formats

The currently supported neutral format data files are:

- .igs/.iges - Initial Graphics Exchange Specification file
- .stp/.step - Standardized Exchange of Product file
- .vda - Verband der Automobilindustrie surface data exchange file
- .x_t/.x_b - Parasolid Transmit file

All of these formats support single or multiple region definitions in a single file. When importing multiple files as regions, the choice of generating either a single region per file or multiple regions is provided. For more information on the various import options, see Importing Surface Data into STAR-CCM+.

None of the neutral file formats support feature curves (line cells). Feature curves can be automatically created based on an angle (sharp edges) or all feature edges during the import process if desired.

A summary of the neutral format import types is given below:

Format Name	File Extension	Feature Curves	Multi-Region	Boundary Names
IGES	`.igs/.iges`	No	Yes	No
STEP	`.stp/.step`	No	Yes	Yes
VDA	`.vda`	No	Yes	Yes
Parasolid Transmit	`.x_t/.x_b`	No	Yes	Yes

IGES Format

Initial Graphics Exchange Specification (IGES) is an ASCII-based neutral data format that allows the exchange of digital information among CAD systems. It can contain models in the form of solid models as well as other types of product data. Although it has been around for almost 30 years, and has been superseded by other formats such as STEP, it is still the most widely used standard for translating CAD data.

The demarcation of parts and surfaces is determined automatically based on the bodies and patches that are contained in the file. Surfaces that are imported from IGES files are always represented as CAD parts initially.

No units specification option is provided as this information is automatically preserved by the file.

STEP Format

Standard for the Exchange of Product model data (STEP) is an international standard for the exchange of industrial product data. It was developed to provide a method for describing product data through its life cycle but is typically used to exchange data between various CAD, CAM, and CAE applications. STEP files are saved in ASCII. STAR-CCM+ currently allows the import of AP203 and AP214 (geometry only) STEP file formats only.

Surfaces that are imported from STEP files are always represented as CAD parts initially. No units specification option is provided as this information is automatically preserved by the file.

VDA Format

VDA is a CAD data exchange format that allows surface data to be transferred from one CAD program to another. The VDA name stands for Verband der Automobilindustrie and originated from the German automobile industry. It is also called VDA-FS in some CAD programs.

No units specification option is provided as this information is automatically preserved by the file. Surfaces that are imported from VDA files are always represented as CAD parts initially.

Unclosed surfaces in the original `.vda` file remain unclosed when imported into STAR-CCM+. Use the surface wrapper to obtain a closed surface suitable for volume meshing.

Parasolid Transmit Format

Parasolid is a geometric modeling kernel that is owned by Siemens PLM. It is used as the basis of many CAD, CAM, and CAE packages for modeling three-dimensional solid objects. Parasolid Transmit files can be binary (`.x_b`) or ASCII (`.x_t`) - the selection of the file format type is made during the export process.

No units specification option is provided as this information is automatically preserved by the file. Surfaces that are imported from Parasolid Transmit files are always represented as CAD parts initially.

To allow precision to be handled correctly, all parts of the Parasolid body must fit in a box no larger than 1000 x 1000 x 1000 units and is centered at the origin. For example, if the default units in the CAD application are set to meters, the maximum distance in any one direction that can be modeled is 1 km.

What Is Native CAD Data?

Native CAD data files are created with parametric based CAD programs and contain all the relevant solid model information for that program.

The model information can be read for each of the supported file types and converted into individual regions that can be used for surface and volume meshing. An additional reader license, specific to each CAD program format, is required in order to read the native files. Contact your Siemens PLM Software representative for more information on how to obtain the appropriate license.

Provided the correct preparation is made, the underlying CAD model can be used for CAD projection of surface vertices.

Contents:

Supported Native CAD Import Formats

Supported Native CAD Import Formats

The type of native CAD file you can import is dependent on the hardware that you are running the STAR-CCM+ client on.

The following options are supported:

- Creo Elements/Pro
- CATIA v4
- CATIA v5
- CATIA v5-6 2012
- NX
- Solid Edge
- Inventor
- SolidWorks

The CAD Exchange Reader add-on license is required to import native CAD formats. All the above formats support single or multiple region definitions in a single file. When importing multiple files as regions, the choice of generating either a single region per file or multiple regions is provided. None of the native CAD formats support feature curves (line cells). You can automatically create feature curves that are based on an angle or all feature edges during the import process if desired. For more information on the various import options, see Importing Surface Data into STAR-CCM+.

Surfaces that are imported from native CAD files are always represented as CAD parts initially.

Creo Elements/Pro 5.0

Creo Elements/Pro 5.0 (formerly Pro/ENGINEER) is a parametric and fully integrated 3D CAD/CAM/CAE software package from Parametric Technology Corporation (PTC). Solid models that are produced using the CAD modeler are saved using files with the extensions `.prt` (for parts) and `.asm` (for assemblies). These files can be read into STAR-CCM+ while still retaining individual part and assembly names, as well as the association between each. If your geometry models contain accelerator (`.xpr`/`.xas`) files, then these files are required to import the geometry into STAR-CCM+.

CATIA V4 Format

CATIA (Computer Aided Three Dimensional Interactive Application) is a multi-platform 3D CAD/CAM/CAE software from Dassault Systemes. CATIA V4 is principally based on piece wise polynomial surfaces. CAD models that are saved using files with extensions `.model`, `.exp` and `.session` can be read into STAR-CCM+.

CATIA V5 Format

CATIA (Computer Aided Three Dimensional Interactive Application) is a multi-platform 3D CAD/CAM/CAE PLM software from Dassault Systemes. CATIA V5 features a parametric NURBS-based modeler. CAD models that are saved using files with extensions `.CATPart` and `.CATProduct` can be read into STAR-CCM+.

CATIA V5-6 2012

CATIA (Computer Aided Three Dimensional Interactive Application) is a multi-platform 3D CAD/CAM/CAE PLM software from Dassault Systemes. CATIA V5-6 2012 features a parametric NURBS-based modeler and

offers enhanced compatibility between CATIA V5 and CATIA V6 files. CAD models that are saved using files with extensions `.CATPart` and `.CATProduct` can be read into STAR-CCM+.

Siemens NX Format

NX is a 3D CAD/CAM/CAE PLM software from Siemens PLM Software. NX was originally called UniGraphics. NX is a parametric solid/surface feature based software that utilizes the Parasolid modeling kernel. CAD models that are saved using files with extensions `.prt` can be read into STAR-CCM+.

Solid Edge Format

Solid Edge is a 3D CAD software from Siemens PLM Software. Solid Edge is a parametric solid/surface feature based software that utilizes the Parasolid modeling kernel. CAD models that are saved using files with extensions `.par` and `.asm` can be read into STAR-CCM+.

Autodesk Inventor Format

Inventor is a 3D CAD software from Autodesk Inc. Inventor is a parametric solid/surface feature based software that utilizes the proprietary ShapeManager modeling kernel. CAD models that are saved using files with extensions `.ipt` and `.iam` can be read into STAR-CCM+.

SolidWorks Format

SolidWorks is a 3D mechanical CAD program from the SolidWorks Corporation, a subsidiary of Dassault Systemes. It is available only for the Windows OS. SolidWorks is a parametric feature-based software that utilizes the Parasolid modeling kernel. CAD models that are saved using files with extensions `.sldprt` and `.sldasm` can be read into STAR-CCM+.

What Is XML Data?

XML files are top-level data files that typically describe the assembly structure and components of a product.

It contains references to other files that contain the actual surface data for parts. Currently, two different implementations are supported - a basic version for Medina files and a more advanced PLM-based implementation suitable for other file formats such as JTOpen.

XML data provides a convenient method of importing hundreds of files into STAR-CCM+ using a common set of options for each, while still maintaining the structural hierarchy of components within the original design. This method allows you to import assemblies into STAR-CCM+ and retain the assembly structure in the geometry parts tree. Similarly, you can export data from the parts tree to a PLM-based XML file. This capability is useful for exporting many parts, or for preserving the organization of the parts for import into another simulation. Meta data information, such as materials, are currently ignored.

CAD Packages Support

This section contains a list of supported CAD Clients, the CAD import versions, and the CAD export versions.

CAD Packages for CAD Clients

When installing CAD Clients, the target CAD package must be present on the installation machine. Without this, you can force the installation of a particular sub-component, but there is no guarantee that this will work correctly with your CAD package. The following CAD packages are required to run the CAD Clients:

CAD Client	CAD Package Version	Comments
STAR-CAT5	CATIA V5-6R2013 (R23)	An additional `ME2` or `MD2 + GPS` configuration license is required from Dassault Systemes to run STAR-CAT5.
	CATIA V5-6R2014 (R24)	
	CATIA V5-6R2015 (R25)	
	CATIA V5-6R2016 (R26)	
STAR-NX	NX 9.0 (Maintenance Package 1 or later)	
	NX 10.0	
	NX 11.0	
STAR-Creo	Creo Parametric 2.0 (Datecode M200 or later)	
	Creo Parametric 3.0 (Datecode M080 or later)	
STAR-Inventor	Inventor 2016	
	Inventor 2017	

CAD Import Versions

CAD import is available on all platforms. The supported file formats and corresponding version numbers are given below.

File Format	File Extensions	Versions Supported	Add-on Required	Type of Import
CATIA V4	.model, .exp, .session	Up to 4.2.5	CAD Exchange	Triangulated Data, B-rep
CATIA V5	.CATPart, .CATProduct, .cgr	Up to V5-6 R2016 (R26)	CAD Exchange	Triangulated Data, B-rep
CATIA V5 (3D XML)	.3dxml	Up to V5-6 R2016 (R26)	CAD Exchange	Triangulated Data
CATIA V6 (3D XML)	.3dxml	Up to 2013	CAD Exchange	Triangulated Data
SolidWorks	.sldprt, .sldasm	Up to 2016	CAD Exchange	Triangulated Data, B-rep
JT Open	.jt	Up to v10.0	JTOpen Reader & CAD Exchange	Triangulated Data, B-rep

File Format	File Extensions	Versions Supported	Add-on Required	Type of Import
NX	.prt	UG v11.0 to NX 11.0	CAD Exchange	Triangulated Data, B-rep
Solid Edge	.par, .asm	v19 to v20, ST to ST9	CAD Exchange	Triangulated Data, B-rep
Parasolid	.x_t, .x_b,	Up to v30.00.167	None	B-rep
Pro/E - Creo	.asm, .prt	Pro/E 19.0 to Creo 4.0	CAD Exchange	Triangulated Data, B-rep
Inventor	.ipt, .iam	Up to 2017	CAD Exchange	Triangulated Data, B-rep
IGES	.igs, .iges	5.1, 5.2, 5.3	None	B-rep
STEP	.stp, .step	AP 203 E1/E2, AP 214, AP 242	None	B-rep
Stereo Lithography	.stl	All versions	None	Triangulated Data
VDA-FS	.vda	1.0, 2.0	CAD Exchange	B-rep
Rhino 3D	.3dm	4, 5	CAD Exchange	Triangulated Data, B-rep
ACIS	.sat, .sab	Up to V26.0	CAD Exchange	Triangulated Data, B-rep

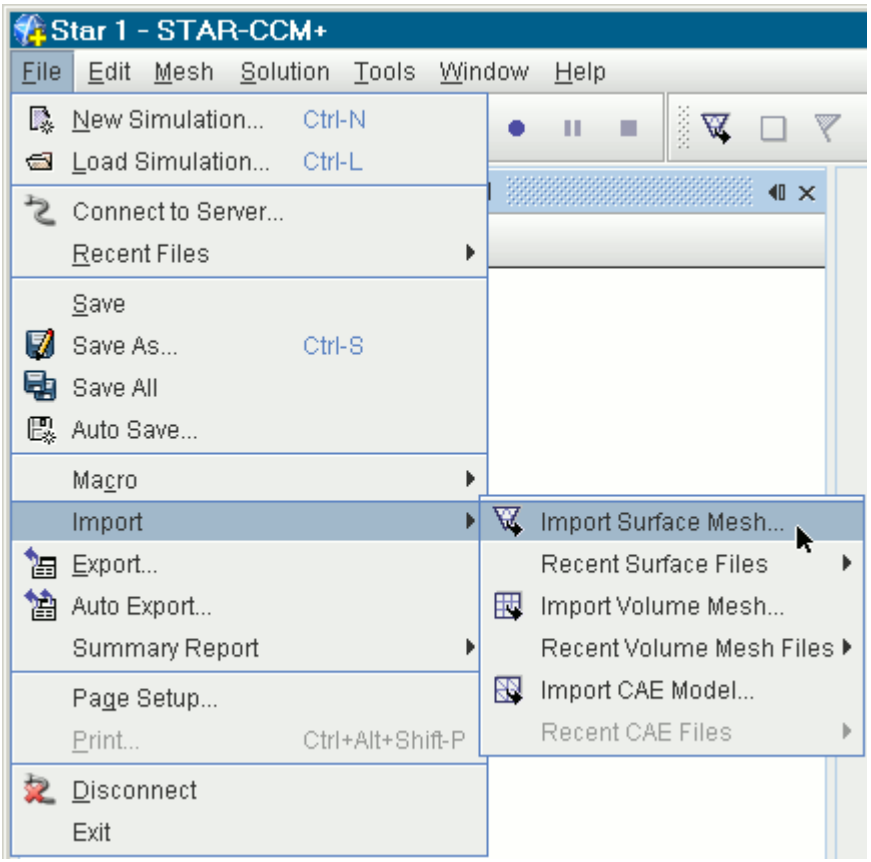
CAD Export Versions

CAD export is available on all platforms. 3D-CAD supports exporting to the following file formats.

File Format	File Extensions	Versions Supported	Add-on Required	Type of Export
Parasolid	.x_t, .x_b	v30.00.167	None	B-rep
IGES	.igs, .iges	5.3	CAD Exchange	B-rep
STEP	.stp, .step	AP 203 E1/E2, AP 214, AP 242	CAD Exchange	B-rep

Importing Surface Data into STAR-CCM+

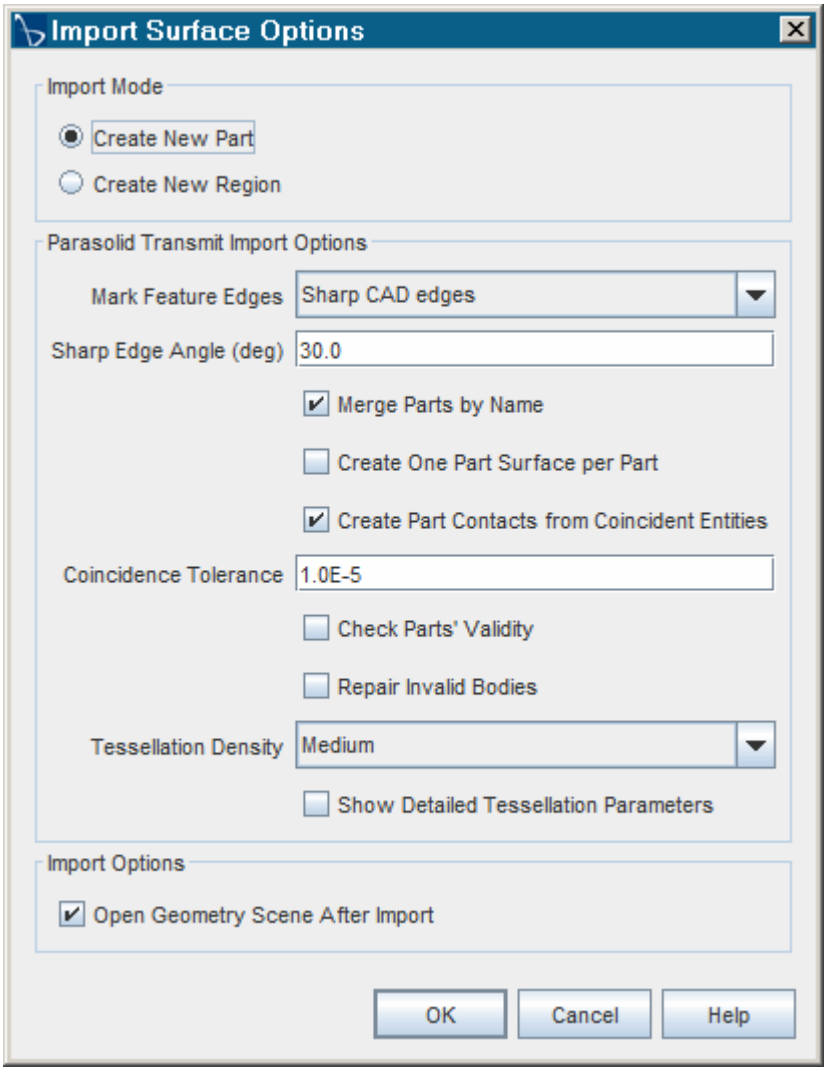
All surface data (except for XML-based imports) for use in the meshing pipeline is imported into STAR-CCM+ using the **File** > **Import** > **Import Surface Mesh...** menu item:



Alternatively, the (**Import Surface**) toolbar button can be used instead.

The standard file import *Open* dialog panel appears.

To import the file or files, select them and click **Open**. The *Import Surface Options* dialog appears. An example is shown below:



Depending on whether a surface file, neutral file or native CAD file format is selected, STAR-CCM+ prompts for various information to assist with the import process and better prepare the starting surface.

Note: If you are unsure about the quality of the surface mesh that you are importing, use the manual surface repair tools to identify, isolate, and fix errors on a surface. See Preparing Surfaces. For CAD geometry, you can activate the **Repair Invalid Bodies** option to instruct STAR-CCM+ to attempt to automatically heal the CAD data in the geometry. This option invokes both the Parasolid healing option from the CAD translator, and the automatic healing from the CAD Repair tool. Parasolid healing only takes place when you activate this option and import the CAD model in STAR-CCM+. If you import the CAD model without activating this option, and subsequently run CAD repair manually, only the CAD repair healing is executed. See Repairing Geometry. You can also use the Check Parts Validity option to make sure the parts that you import are valid.

By default, all surfaces are imported as parts, but the *Import Mode* option can be toggled to allow surfaces to be imported as regions instead if desired.

Additional information pertaining to each file format is provided in the format and import reference. Once the options have been selected, the **OK** button can be pressed to import the indicated file.

When importing native CAD or neutral format data, whenever STAR-CCM+ marks a mesh edge as a feature edge, it records which CAD edge the feature edge lies along. This association of feature edges to CAD edges can be used later. For example, when the surface wrapper or surface remesher splits a feature edge that is known to lie along a CAD edge, it projects the split point to the CAD edge. (This projection requires you to activate the CAD Projection option.)

The mark feature edges option that is listed above controls this marking of edges. If no mesh edges are marked as feature edges during the import stage, it is not possible to recover the associativity with underlying CAD edges.

Contents:

Import Mode
Boundary Mode
Region Mode
Part Mode
Surface Mode
Units
Merge Vertices
Database ID
Import Type
Using the CAD Client Translators to Import Geometry
Mark Feature Edges
Sharp Edge Angle
Merge Parts by Name
Create One Part Surface Per Part
Create Part Contacts from Coincident Entities
Coincident Face Tolerance
Check Parts' Validity
Repair Invalid Bodies
Tessellation Density
Show Detailed Tessellation Parameters
Sewing Tolerance
Multi-Pass Sewing
Open Geometry Scene After Import
Run Diagnostics After Import
Elements to Import

Import Mode

The *Import Mode* option determines whether the surface data file or files are imported as regions or as geometry parts.

In general, choose a mode that is based on the nature of the import data and what you want to do with it. For example, if you only have a few bodies and do not want to maintain hierarchal assembly of parts then

importing the data as regions is probably more efficient. However, if you have dozens of bodies and want to organize them then in assemblies for future reference then using the parts option would be better. Also, to use closed bodies as volumetric sources, import them using the parts option.

Boundary Mode

When you select the **Create new Region** option for *Import Mode*, the boundary mode determines:

- How different surface patches that have been identified in the surface file are imported into STAR-CCM+
- How these patches are converted into boundaries

A patch is a part of the surface that has been tagged in some way to be different from the rest. For example, it has been assigned a different cell type or family name. Up to three different boundary mode options are presented:

- **One boundary per cell type/property ID/solid** (default) - This option results in a different boundary being created for each cell type, property ID or solid depending on the input file type. For the database and cell/vertex/input formats, the **Combine boundaries by name** option is on by default. This option is also available to allow cell tables with the same name to be automatically combined during import if desired.
- **One boundary per group** - This option results in a different boundary being created for each group. A group specifically refers to cell table grouping that was defined previously in the pro-STAR or pro-surf pre-processing software.
- **One boundary for all cell types/property ID/solid** - This option results in only one boundary being created regardless of the initial number of cell types/property ID/solid definitions.

In all the above situations, the original patch definition in the import surface can be used later to edit the boundary definitions further with the split boundary by patch option.

Additionally, when importing pro-STAR database or cell/vertex/input files, if the **One boundary per cell type** option is selected for **Boundary Mode** then the **Combine Boundaries by name** toggle is available. Activating this option combines any boundaries that have the same name. Boundaries can have the same name when two or more cell types in pro-STAR have the same name.

STAR-CCM+ automatically assigns boundary types when a mesh is imported.

Region Mode

The region mode determines how the regions are created based on the input data.

The option is displayed when importing surfaces as regions and under the following conditions:

- Two or more surface mesh files are selected.
- One or more neutral format or native CAD data files are selected.

When importing multiple surface mesh files, two different options are provided for the region mode:

- **One region per file** - This option results in one region being created for each separate input file. The region name that is used matches the file from which it came.
- **One region for all files** (default) - This option only creates one region for all the files. In this instance, an additional option to **Combine patches across files** is provided to group any duplicate patch definitions in each file into the same boundary. (This grouping does not involve deleting one of the patches.) The

default setting does not combine the patches. The patch combine option is not provided in the case of STL files.

When importing neutral format or native CAD data files, the options change to be as follows:

- **One region per body** - This option results in one region being created for each body that is contained in the input file.
- **One region for all bodies** - This option creates one region for all bodies that are contained in the file.

In addition to the above, if multiple files are selected then the **One region per file** option is also available.

Part Mode

The **Part Mode** option is available when importing surface mesh files and determines the way geometry parts are created based on the underlying surface description.

The option is displayed when importing surfaces as parts and under the following conditions:

- A single STL (`.stl`), database (`.dbs`), or JTOpen (`.jt`) file type is selected.
- Two or more surface mesh files are selected.

Three different options are provided for the part mode depending the above selections:

- **One part** - This option results in only one part being created from the surface file (applicable when a single STL, database, or JTOpen file has been selected).
- **One part per file** - This option results in a new part being created for each input file (applicable when two or more surface mesh files have been selected).
- **One part per body** - This option results in a new part for each underlying body that is contained in surface mesh file or files. The exact definition of “body” is given below for each surface mesh file type.

The definition of the word “body” varies according to the surface mesh file type. The following is a summary of the terms that are used in the import dialog:

Body Name	File Format Applicable To
Cell Type	pro-STAR (`.inp`), database (`.dbs`)
Solids	STL (`.stl`)
Patches	Felisa Front Surface (`.fro`)
Shapes	JTOpen (`.jt`)
Property ID	Nastran (`.nas`), Patran (`.pat`)

For example, importing a surface from a database file includes the **One part per Cell Type** option for **Part Mode**. When you select this option, each cell type in the surface mesh results in a separate part being generated.

The selection that is made for **Part Mode** also influences the state of the Surface Mode option. Typically, the surface mode options are only available when the **One part** or **One part per file** option has been selected for **Part Mode**.

Surface Mode

The **Surface Mode** option is available when importing surface mesh files and determines the way individual part surface nodes are created for a geometry part.

The option is displayed when importing surfaces as geometry parts. Where the **Part Mode** option is also available then the part mode choice influences the state of the surface mode selection. Generally, the surface mode choice is only available when the import results in one part being created for each file.

Depending the surface mesh file type being imported, up to four different options are available:

- **One surface per body** - This option results in a surface node for each body that is contained in the file.
- **One surface per group** - This option results in a surface node for each group. The definition of a group depends on the file type.
- **One surface for all bodies** - This option results in only one surface node being created for the file.
- **One Surface per color** - This option results in a surface node for each color. This option is only available for STL binary files.

The definition of a *body* varies according to the file type and is described in the part mode section.

Additionally, for some surface mesh files, if the **One surface per cell type** option is selected for **Surface Mode** then the **Merge cell types by name** toggle is available. Activating this option combines any surfaces that have the same name. Surfaces can have the same name when two or more cell types in pro-STAR have the same name.

Units

A choice of centimeters (cm), feet, (ft), inches, (in), kilometers, (km), meters (m) (default), miles (mi), millimeters (mm), micrometers (um), and yards (yd) is allowed.

The selection determines the scale factor that is applied to the import surface to make it conform to SI (meters). For example, selecting millimeters automatically scales the vertices by 0.001. If a unit other than one listed was used for the surface file, select the meters option (equivalent to no scaling). After the import process is complete, you can apply a user-defined scaling factor to scale the mesh.

If a unit other than meters (m) was selected, then the option to **Set preferred units for length** is available (off by default). This option can be used to set the selected unit as the default for all mesh reference and input values that involve length. Leaving the option off maintains meters as the default unit for length.

Merge Vertices

When importing either STL or JTOpen files, this option allows the vertices for each surface element to be automatically merged or not.

The default is to merge the vertices. In addition, the vertex merge tolerance value (default 1.0e-06 m) can be set to accommodate different scale specifications in the imported file.

This option only operates on individual parts. It does not attempt to merge coincident vertices between different parts.

Database ID

When a pro-STAR database file is imported, the database ID option allows specific surface geometries to be selected, when the file contains more than one.

The database ID contains a reference ID number and an optional text summary describing the contents.

Import Type

This option appears when importing geometry from file types that support both CAD and tessellated imports--such as CATIA V5 and SolidWorks formats.

Two options are available:

- **CAD Geometry**- Imports the CAD geometry.
- **Tessellation** - Imports the original surface mesh data from the CAD package. The tessellation density cannot be modified after import. Also note that if this option is selected, any options available on the CAD geometry (such as CAD-based boolean operations and aligned meshing) do not apply.

Using the CAD Client Translators to Import Geometry

You can import geometry into STAR-CCM+ using the CAD client translators. These translators can provide a better result when importing your geometry compared to other translators.

To use this feature, you must have a valid CAD Client license and CAD client installation on your machine. If you are importing a native CAD format, the appropriate CAD client is selected by default.

To use the CAD client translators when importing geometry in STAR-CCM+, in the *Import Surface Options* dialog, activate the **Use CAD Client** option.

If a geometry file contains design parameter data from a CAD client, importing this file into STAR-CCM+ also imports the design parameters. A link between the imported geometry and the original geometry file is also established. See **[CAD Client]** > **[CAD Client User Guide]** > **Transferring Geometry to STAR-CCM+** > **Transferring Geometry with Design Parameters**.

Mark Feature Edges

This option allows you to specify which of the mesh edges that lie along CAD edges are marked as feature edges.

CAD edges are the edges or curves that belong to the underlying CAD model; mesh edges are the line segments that bound the mesh faces.

This option applies only to those mesh edges that lie along edges of the CAD model - it does not mark as feature edges any mesh edges that do not lie along CAD edges. The available settings are:

- No CAD edges - Do not mark any mesh edges as feature edges.
- Sharp CAD edges - Mark as a feature edge any mesh edge that lies along a sufficiently sharp CAD edge (based on the **Sharp Edge Angle** value, default 30 degrees). For any given CAD edge, either all or none of the mesh edges that lie along it are marked as feature edges.

- All CAD edges - Mark as features edges all mesh edges that lie along any CAD edges.

Marking feature edges in this way activates CAD projection back to the underlying CAD edge when changes are made to the mesh discretization of those edges.

Sharp Edge Angle

Creates feature edges from edges of attached faces that make an angle sharper than the *Sharp Edge Angle* value.

The default *Sharp Edge Angle* is 30 degrees.

Merge Parts by Name

This option is available when the **Import Mode** is set to create new geometry parts.

Any bodies in the imported assembly with the same name are merged into one part automatically when this option is activated. If this option is deactivated, each body becomes its own part in STAR-CCM+ regardless of name.

Create One Part Surface Per Part

When activated, ignores the different surface names in each part and instead adopts the name of the parent part.

Use this option to merge all part surfaces of a part into a single part surface. This option is available when the **Import Mode** is set to create new geometry parts.

Create Part Contacts from Coincident Entities

This option is available when the **Import Mode** is set to create new geometry parts.

When activated, the importer searches all of the imported CAD faces for pairs of coincident faces that are within the specified *Coincident Face Tolerance*. For each pair found, part contact data is created and stored in the *Contacts* property of each part. Further information is provided in the section, Creating Part Contacts From Coincident Entities.

Coincident Face Tolerance

When the *Create Part Contacts from Coincident Entities* option is activated, the option to specify the *Coincident Face Tolerance* becomes available.

The *Coincident Face Tolerance* specifies the maximum distance at which coincident faces are imprinted. The default value is 1.0E-5. This option is intended to close small, erroneous gaps in imported geometries. Do not use this option to create contacts or interfaces between parts that have large gaps between them. If parts have large gaps between them, move one part closer to the other using a translation operation.

Check Parts' Validity

The Check Parts' Validity option is available when importing any surface mesh file type. When this option is activated, the validity of the surface mesh import is examined.

The main purpose of this option is to provide you with information on whether the selected bodies are geometrically and topologically valid, from the Parasolid geometry kernel point of view. A simple message is displayed in the output window which states if the surface mesh is valid. For example: `Body 2 is not valid`. If the surface mesh is not valid, you can check the quality thresholds and fix the validity issues using the surface repair tool. See the Surface Repair Tool.

Repair Invalid Bodies

Imported CAD models can contain geometry that is not suitable for use in a STAR-CCM+ analysis. This type of geometry is considered invalid, and typically contains errors in the Parasolid geometry. When this happens, you can instruct STAR-CCM+ to attempt repairs on the geometry.

Typically, you activate this option after determining that your geometry contains invalid bodies. To check your geometry for invalid bodies, see Checking Geometry Validity and Check Parts Validity.

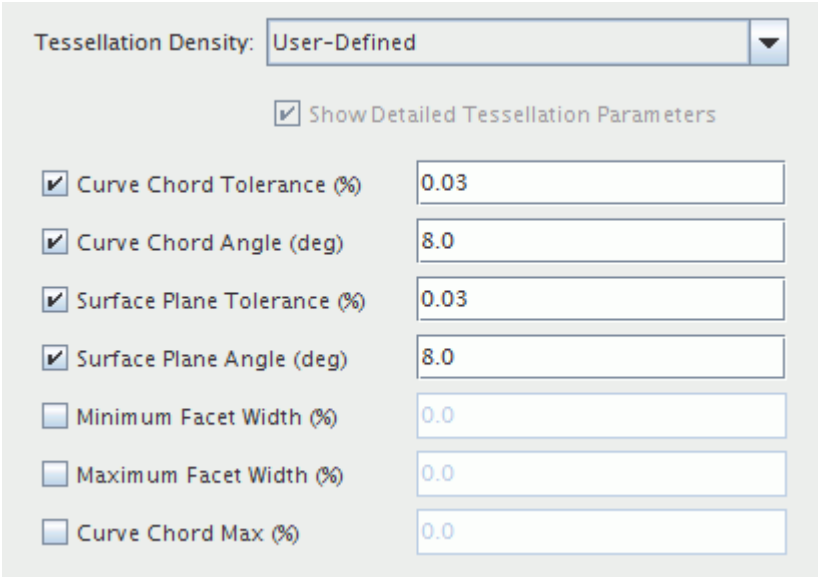
When active, the Repair Invalid Bodies option attempts to heal errors with the underlying Parasolid geometry and reports on the number of invalid bodies that are successfully repaired.

Tessellation Density

The **Tessellation Density** determines the level of surface refinement that STAR-CCM+ uses when importing geometry.

This option affects the degree to which surface curvature is preserved in the imported model, that is, a fine option uses more triangles to discretize a curved surface, while a coarse option uses less. In general, it is recommended you start by importing using one of the coarse options. After importing, review the surface triangulation and, if necessary, retessellate the geometry using a higher density. For geometries with high amounts of curvature, it is recommended to use one of the finer tessellation options if a higher surface remeshing fidelity is desired. The tessellation density option has no effect on planar surfaces - the same triangulation is obtained for these surfaces regardless of the setting used.

Selecting the **User-Defined Tessellation Density** option automatically activates the **Show Detailed Tessellation Parameters** option. The Detailed Tessellation Parameters are then displayed within the *Import Surface Options* dialog. You can edit these parameters as required.



Activating the **Show Detailed Tessellation Parameters** option is also available when each of the other **Tessellation Density** options are selected.

The **Distance-biased** option contains a pre-configured set of tessellation parameters that have shown to produce good results for a wide variety of geometry models.

When selecting the **User-Defined** option, the tessellation parameter values are copied from the default **Medium Tessellation Density** setting. If you select any **Tessellation Density** option other than user-defined before selecting user-defined, the tessellation parameter values displayed are those of the previously selected **Tessellation Density**. This allows you to make adjustments to the tessellation parameter values for each of the **Tessellation Density** options.

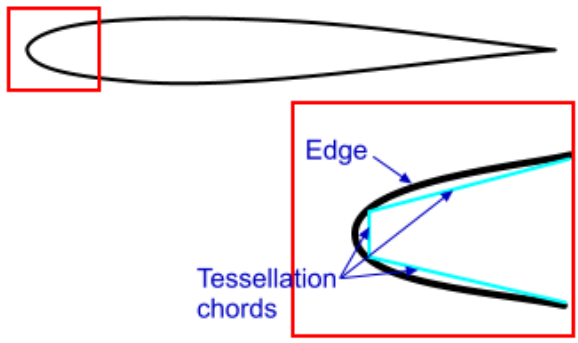
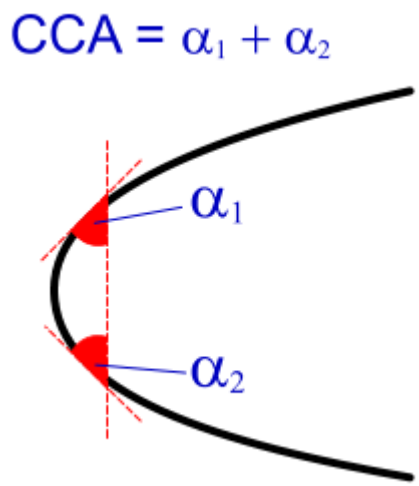
For geometry parts of 3D-CAD, CAD and shape types, it is possible to retessellate the parts using a different tessellation density before they are used in the meshing pipeline.

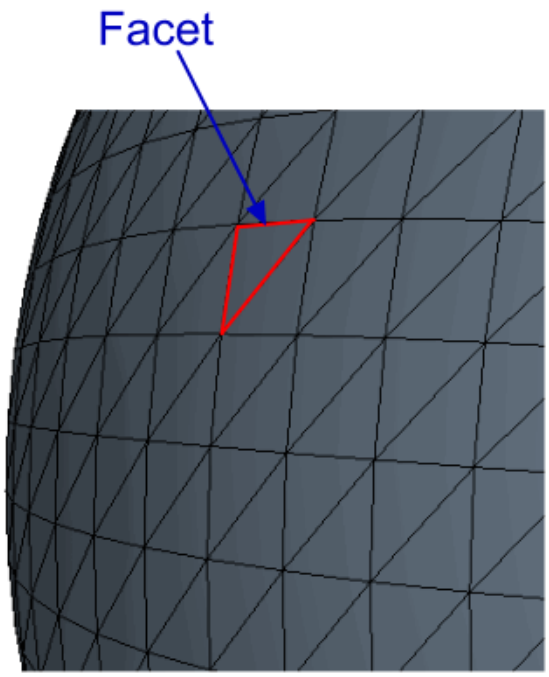
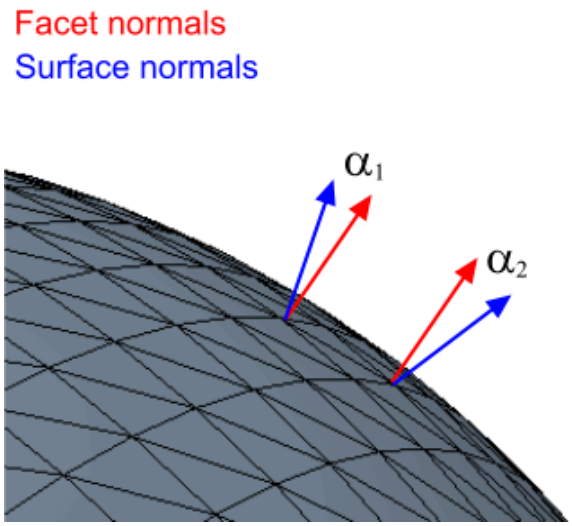
Show Detailed Tessellation Parameters

Activating the **Show Detailed Tessellation Parameters** option displays detailed tessellation parameters within the *Import Surface Options* dialog. You can then edit these parameters as required.

All distance and length tessellation parameters are specified as a percentage of the length of the diagonal of the bounding box in which the parts or bodies are contained.

Curve Chord Tolerance	The maximum allowed distance between an edge and the chords that are used to approximate it.

	
Curve Chord Angle	The maximum angle (in degrees) that is allowed between a chord and its originating curve. This angle is calculated as the sum of the two angles that are formed between a chord and the curve tangents. 
Surface Plane Tolerance	The maximum allowed distance between a tessellation facet and the surface. All points of the tessellation facet lie on the surface being tessellated.

	
Surface Plane Angle	The maximum allowed sum of the two angles (in degrees) between the normal of a facet and two face normals, which are measured at any two positions on the boundary of the facet. 
Minimum Facet Width	Minimum length of any of the sides of the facet. The value specifies a width below which Parasolid disregards supplied curve, surface, and planar tolerance values. You can produce facets with dimensions smaller than this width.

Maximum Facet Width	Maximum length of any of the sides of the facet.
Curve Chord Max	The maximum allowed length of a chord that is used to approximate a portion of an edge.

Sewing Tolerance

For IGES files, it is possible for gaps to exist in the sheet bodies during translation. The sewing tolerance is used to create solid bodies after the translation process.

Gaps that exist between each sheet are sewn together based on the supplied sewing tolerance value (1e-5 m by default). If the sheet bodies are not sewn together properly, then each sheet is treated as a separate body. If the **Region mode** is set to **One region per body**, multiple regions are produced. Additionally, if the translated surface is used as the starting point for the surface wrapper, then the surface quality at edges and corners where sheets meet is not as good as when the surfaces are closed to form a solid body.

See Also:

Multi-Pass Sewing

Multi-Pass Sewing

For IGES files, it is possible for gaps to exist in the sheet bodies during translation.

The Multi-Pass Sewing option, if enabled, sews together gaps that exist between each sheet, based on the sewing tolerance, before and after importing the geometry into STAR-CCM+. If this function is disabled, sewing only takes place once after the geometry is imported. This procedure aims to create solid bodies with improved surface quality at edges and corners where the sheets meet.

See Also:

Sewing Tolerance

Open Geometry Scene After Import

A new geometry scene can be created automatically containing the imported surfaces.

This option is on by default.

Run Diagnostics After Import

The surface quality mesh diagnostics report for the imported surface is produced in the *Output* window when this option is turned on for the import of surface meshes.

Elements to Import

For Medina files, this option allows you to select what mesh (surface, volume, or both) to import into STAR-CCM+.

Importing Surfaces Using an XML File

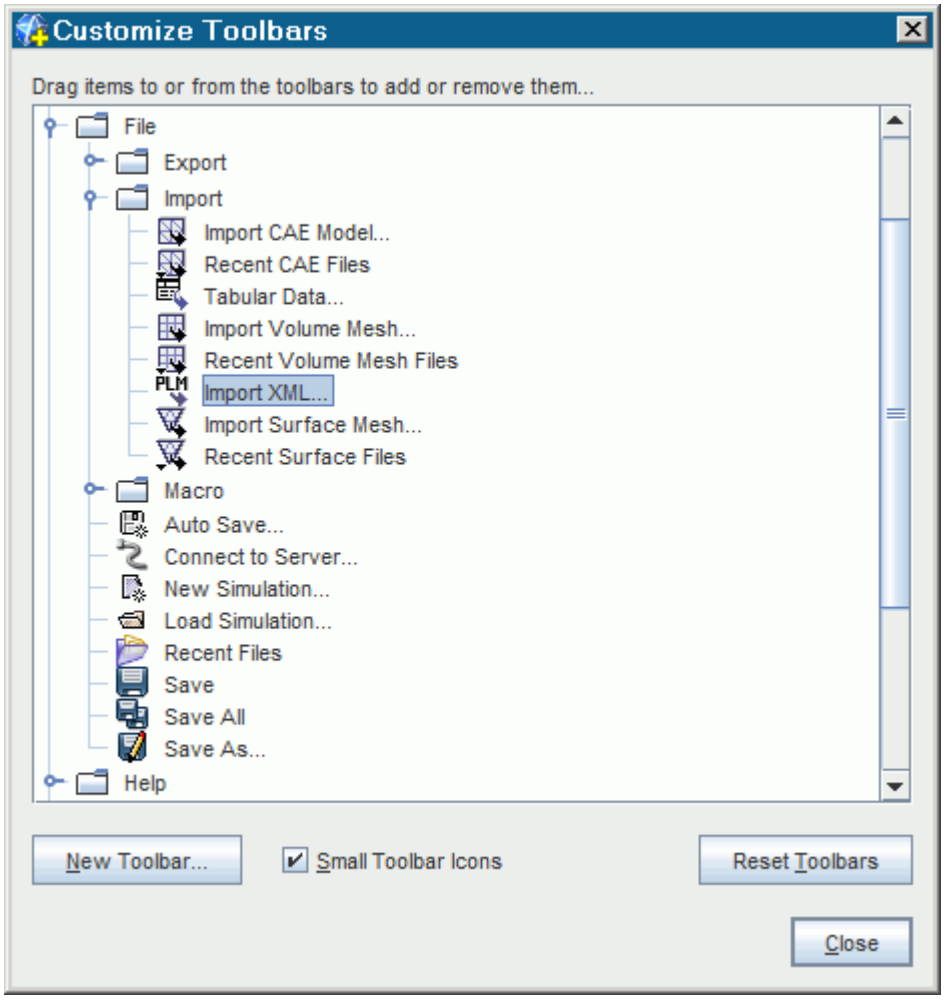
XML-based data files can be used to import many parts with a given hierarchy, such as a CAD assembly.

When this method is used, STAR-CCM+ automatically imports the individual parts that are specified in the XML file and arranges them according to the data in the file. The procedure for XML import is similar to that described above but it uses a different starting point, namely the **Import XML** toolbar button. By default, the button is not available on the interface. You can add it to any suitable toolbar or a new toolbar (as desired).

To add the **Import XML** button to a toolbar:

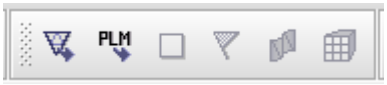
1. Open the *Customize Toolbar*s dialog by selecting **Window** > **Toolbars** > **Customize...** from the main STAR-CCM+ display.

   The *Customize Toolbars* dialog appears.
2. In the dialog, locate the **File** > **Import XML...** node.



3. Drag and drop the **Import XML...** node to an existing toolbar (such as the Mesh Generation one) or use the **New Toolbar...** button to create a new empty toolbar beforehand.

The button is now available on the main display.



4. Click **Close**.
5. Press the **Import XML...** button.
6. In the *Open* dialog, set the **Files of Type** option to the desired file type (either `.xml` for Medina files or `.plmxml` for other types) and locate the import file.
7. To start the import process, click **Open**.

   The *Import Surface Options* dialog appears. This dialog provides options to Merge Vertices (with a vertex merge tolerance), and to set the units for each imported reference file. The merge vertices option is applied to STL and JTOpen format files as they are imported. When importing a JT Open file, you can also specify the JT Open Files Import Type as tessellated data or B-rep (Boundary Representation) data. You can also specify the Tessellation Density of the imported surface.
8. Make the necessary changes and click **OK**.

Each part that is listed in the XML file is imported one at a time using default options depending on the format. If multiple instances of the same part are listed, then a separate part is created for each. Any part files that cannot be located are listed at the end of the process.

XML (MEDINA format) files are only supported in versions of STAR-CCM+ running on Linux. PLMXML files are supported in both Windows and Linux versions of STAR-CCM+.

Contents:

JT Open Files Import Type

JT Open Files Import Type

This property becomes available in the *Import Surface Options* dialog when importing a JT Open file which contains tessellation data and B-rep data.

The following options are available:

Tessellation Only	Only Tessellation data is imported.
B-rep Only	Only B-rep (Boundary Representation) data is imported. Parts that are imported with B-rep data maintain underlying CAD data that is associated with them. CAD data allows better fidelity of the surface during remeshing and when performing CAD-Based boolean operations. Mesh data is also created for B-rep parts during import.
Prefer Tessellation	If the JT file contains Tessellation data, the Tessellation data is imported preferably to any B-rep data. However, if there is no Tessellation data, B-rep data is imported.
Prefer B-rep	If the JT file contains B-rep data, the B-rep data is imported preferably to any Tessellation data. However, if there is no B-rep data, Tessellation data is imported.

For more information, see JT Open Surface Format.

Importing Multiple Surface Data Files

STAR-CCM+ allows multiple surface input files of the same format to be imported simultaneously using the **File** > **Import Surface** option.

Multiple files are selected by pressing the **Ctrl** or **Shift** key at the same time as selecting the file. In addition to the import options provided for single file input, the **Region Mode** option can further influence how regions are created when importing multiple surface data files as regions.

When importing surface mesh files, regardless of the **Region Mode** option that is selected, the imported surface contains two surfaces where one part of the geometry region joins the next. Additional advice on what to do about duplicate surfaces at mesh interfaces is given in the section on multi-region meshing.

Importing a series of surface mesh files one at a time produces the same result as importing the files simultaneously using the **One region for all files** option without merging the patches (or vertices in the case of STL files). This option is effectively the safest import option as no assumptions are made about how the surfaces are joined.

Note the following:

- When you import multiple pro-STAR database files, it is assumed that the surface in each file is saved with the same database ID value.
- It is assumed that the same units have been used for all surface input files.
- Two or more surface data files of differing formats cannot be imported at the same time.

When you import multiple neutral format or native CAD files, regardless of the **Region Mode** option that is selected, the imported data contains two boundary surfaces at common interfaces where one geometry region joins the next. Additional advice on what to do about duplicate surfaces at mesh interfaces is given in the section on multi-region meshing.

Note the following:

- Importing a series of neutral format or native CAD data files one at a time produces the same result as importing the files simultaneously using either the **One region per file** or **One region per body** option for **Region Mode**. This option is the safest import option as no assumptions are made about how the faces are joined.
- Two or more neutral data files of differing formats or two or more native CAD data files of differing formats cannot be imported at the same time.

Reviewing the Surface Data Import Report

A summary of the surface statistics for the imported file is reported in the *Output* window.

A typical example for when surface mesh files are imported is shown below:

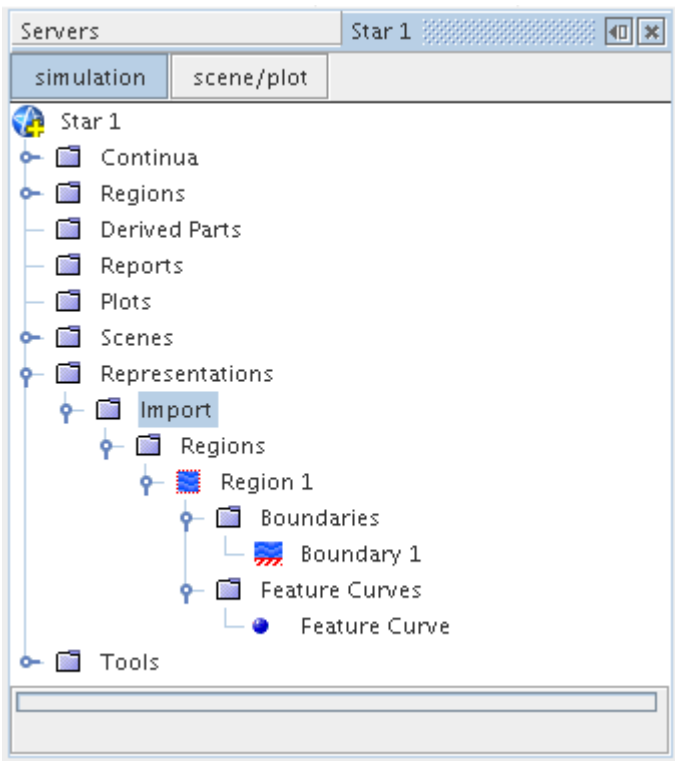
```
Output - Star 1
Reading surface from .dbs file: star.dbs, id: 2
Scaling from model units: meter
Done importing surface.  Imported surface statistics:
  # Faces:         11400
  # Vertices:      5702
  # Special Edges: 182
```

When you import surface mesh files, review the total number of faces, vertices, and edges for the surface to make sure that they are what you expect. `Special Edges` refer to line cells, which are used to represent feature curves.

A more extensive report is obtained when running the mesh diagnostics option and reviewing the report for the surface mesh. When you import translated neutral format or native CAD data, the face quality for the translated neutral format surface can be low since only a basic triangulation of the surface is performed.

Using Surface Data Import Representations

When importing surfaces as regions, an **Import** representation is created under the **Representations** node in the simulation tree.



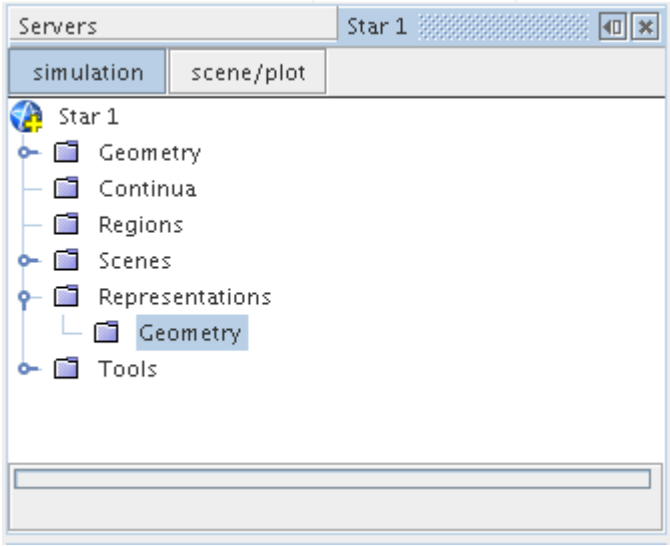
The import node can be used to:

- Export the surface.
- Replace the existing surface with a new one.

- Repair the surface manually.

If you are replacing the existing surface, all existing parameter settings are maintained.

When a surface is imported as a part, a **Geometry** representation is created instead:



The geometry node is simply a place holder in this instance to indicate that a surface has been imported as a part. As such it has no properties.

Checking the Surface Validity

The validity of the surface can be checked using one of two different methods:

- Using the surface preparation options dialog.
- Using thresholds.

Three different derived part threshold scalars can be used to check the quality. The checks are:

- Free edges
- Non-manifold edges
- Pierced Faces

Free edges refer to cell faces that are not connected to a neighboring face. A non-manifold edge refers to a cell face edge that is connected to two (or more) other edges. A pierced face is one that has one or more edges of another surface cell piercing it.

To create the above threshold checks, do the following:

- In the **Input Parts** list, specify the boundary surfaces that you want to check. It is not sufficient to include only the region part.
- Select the appropriate *Scalar* for the desired surface check (either **Free Edges**, **Non-manifold Edges, or Pierced Faces**).
- Set the **Extraction Mode** to **All Above**.
- Set the **threshold value** to 0 and press the **Create** button.

A new derived part is created containing the cell faces next to the problem areas which allows you to investigate the problems further.

If you have any doubts over the validity of the starting surface, then run these checks as a precaution. Fix any problems that are found: the volume meshers cannot proceed until all problems are fixed. Currently, free edges can be fixed by using the hole filler, the edge zipper, or the manual fixing tools, depending the nature of the problem. A non-manifold edge is sometimes valid depending on whether the surface cells causing the problem are supposed to be a part of the geometry or not. For example, consider a geometry that contains a baffle surface that is joined to the exterior surface. This geometry is not a problem as long as the surface is converted to an interface before proceeding with the volume meshing. If the non-manifold edge is not expected, then you can delete the faces that are causing the problem. Hand fixing tools to remove pierced faces are now supported in STAR-CCM+ and as such can be fixed directly within the software itself.

Understanding Geometry Parts

You can use the geometry parts manager node to setup meshing procedures.

Using the parts-based meshing approach allows a more flexible, re-playable, pipeline of operations, which is particularly useful for any or all of the following scenarios:

- Where you want to mesh multiple parts in the same region with different types of volume mesh.
- Where a hierarchical assembly of CAD assemblies and parts exist that must be maintained within the STAR-CCM+ environment.
- Where you want the control over the ability to repair or swap out individual parts without having to work with the entire surface.
- Where you want to define a template `.sim` file, containing meshing and modeling settings so that geometries can be imported later.
- Where you want to use a non-standard volumetric control shape that a closed volume part defines.
- Where you want to use the 3D-CAD modeler within STAR-CCM+ to design and parameterize a geometry so that an optimal flow solution is obtained.

This section covers how geometry parts are created or imported, how to organize them and how to apply tags and filters to them. Geometry parts can be repaired and manipulated in a similar way to surface meshes. Once parts have been organized and prepared, they are assigned to regions and an initial surface mesh is created by initializing the meshing models.

Although the parts-based meshing approach is recommended, you do not have to use parts to generate a mesh. You can import parts, assign parts to regions, then generate mesh at the regions level. However, if you are working with assemblies that contain dozens or hundreds of parts, using the parts-based meshing approach is beneficial. Also, if you intend to use the 3D-CAD modeler to design the geometry from scratch or modify an existing CAD model, using geometry parts is a natural part of the process of going from CAD to flow solution.

Contents:

What Is a Geometry Part?
What Is Metadata?
Creating and Importing Parts
Organizing Assemblies Using Composite Parts
Using Part Contact Data to Create Interfaces
Updating Part Interfaces

See Also:

Introducing 3D-CAD Modeling
Repairing Part Surfaces and Curves
Manipulating Parts
The Surface Repair Tool

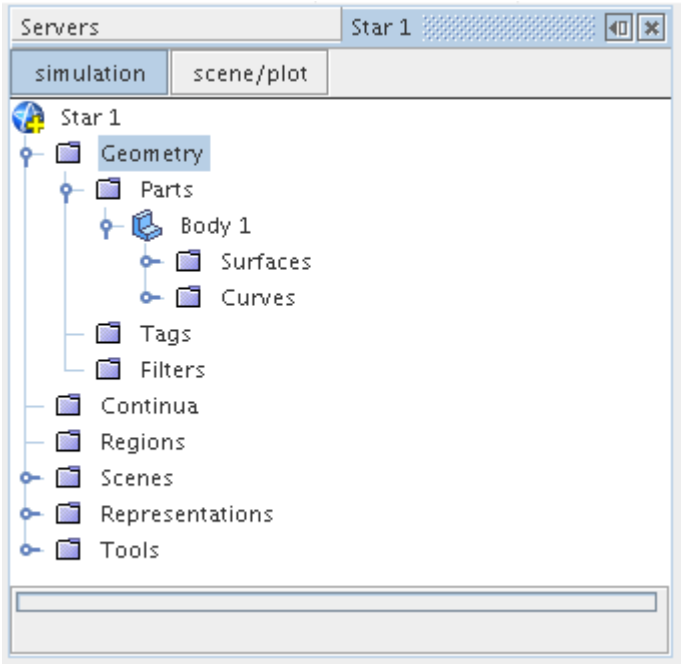
What Is a Geometry Part?

A geometry part represents an object (leaf level part) or collection of objects (composite part) that can be used as the input geometry for the meshing tools.

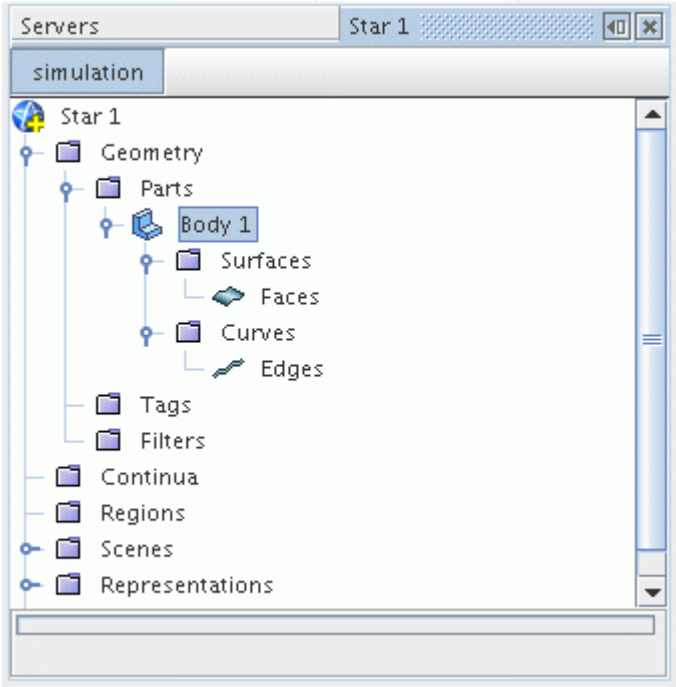
These geometry parts can originate from several different sources:

- Leaf Level parts
  - conversion of 3D-CAD bodies
  - imported surfaces
  - creation of shape geometries within STAR-CCM+
  - results of mesh operations
- Composite Parts
  - new empty composite parts
  - from two or more existing parts

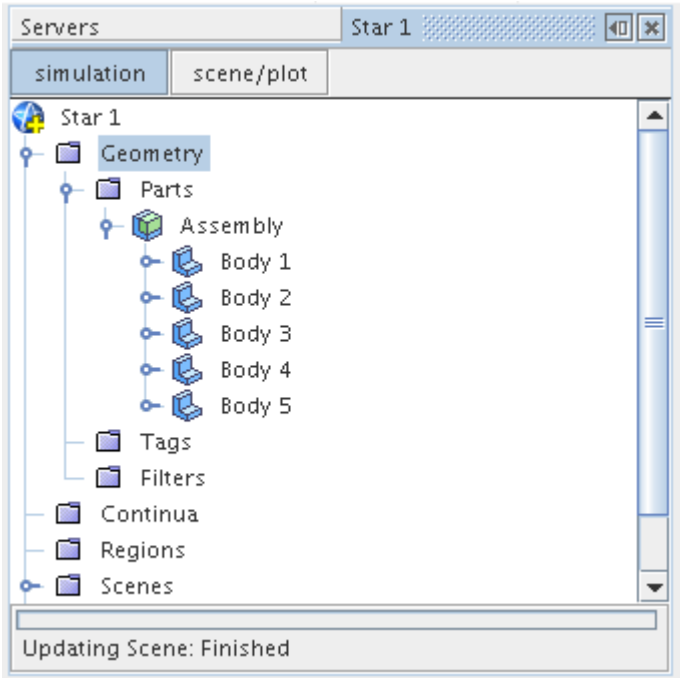
The representation of a single part (generated during import and called **Body 1**) in the explorer pane is shown below:



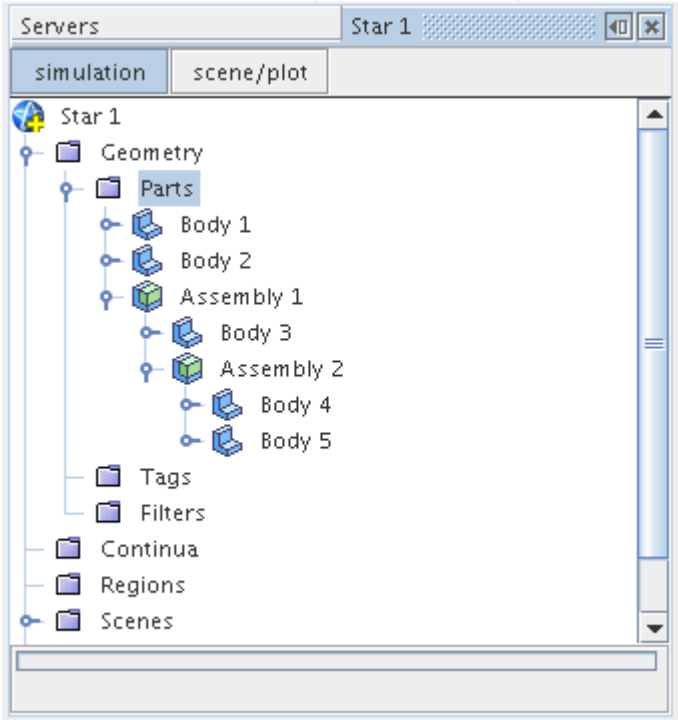
Leaf-level parts are the most fundamental kind of part represents a surface and curve set, which themselves contain faces and edges respectively. A surface can contain one or more sets of faces and a curve can contain one or more sets of edges. These features are represented below for the **Body 1** part:



A geometry part can also refer to an assembly (composite) of parts or any combination of parts and assemblies. Assemblies can also contain subassemblies, which themselves can contain further subassemblies, and so on, for which the final level is one or more leaf-level parts. An example composite part, with five leaf-level parts, is shown below:



A more complex example, containing leaf-level parts, assemblies, and subassemblies is shown here:



There is no limit to the number of leaf-level and composite parts that are allowed in STAR-CCM+. Within a given composite part, the leaf-level parts must have unique names although these names can be reused in other composite parts.

In order to help identify and sort parts, they can be tagged such that a filter can be applied to them when they must be selected for a given meshing operation. Leaf-level parts can also be rearranged so that they can belong to composite parts and composite parts can be exploded so that they no longer form part of an assembly. This feature makes the organization and managing of parts a lot easier than if they are represented only as regions.

When you examine geometry parts in a scene, you can select them by type using a filter in the Selection toolbar.

Geometry parts provide a more efficient way of organizing data that are eventually used for the meshing tools. Not all parts are required for a given meshing process: in order to represent the domain of interest, specific geometry combinations can be built up instead, by selectively assigning parts to regions.

Geometry parts, as well as part surfaces, part curves, and part contacts, all have unique indices that you can use in field functions.

Contents:

What Are Leaf-Level Parts?
What Are 3D CAD Parts?
What Are CAD Parts?
What Are Shape Parts?
What Are Mesh Parts?
What Are Mesh Operation Parts?
Common Properties of Leaf Level Parts

What Are Composite Parts?
What Is a Part Surface?
What Is a Part Curve?

What Are Leaf-Level Parts?

Leaf-level parts are parts that represent an object, such as a three-dimensional shape, which is formed from a collection of surfaces and curves.

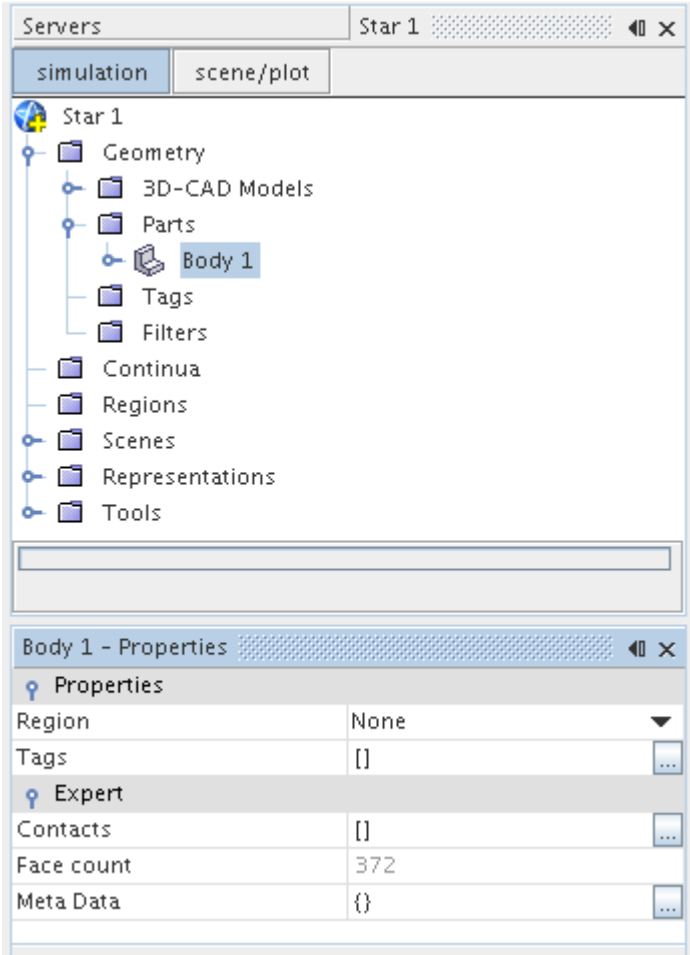
Different operations can be carried out on parts to modify them, for example combining two or more parts. The following types of leaf-level parts are supported in STAR-CCM+:

- 3D-CAD parts that originated from the 3D-CAD modeler
- CAD parts that originate from a CAD import
- Surface mesh parts that originate from non-CAD import formats
- Shape parts that can be created from within the **Parts** node itself.
- Mesh Operation parts that are the result of mesh operations.

What Are 3D CAD Parts?

3D-CAD parts originate from being directly converted from the 3D-CAD modeler.

Light gray part icons, such as this one , represent these parts. They still retain a connection to the originating 3D-CAD body such that if a design parameter is changed then the 3D-CAD part can be updated to match the design change that was made. These parts have properties and an extended right-click menu.



3D-CAD Parts Properties

See Common Properties of Leaf-Level Parts

What Are CAD Parts?

CAD parts originate from native CAD and neutral format CAD data.

Blue part icons, such as this one , represent these parts. Once imported into the geometry part node, they are considered static topologies although they can be transformed and duplicated if necessary. CAD parts share the common properties for leaf-level parts and are subject to the same set of operations on their right-click menu.

CAD Parts Properties

See Common Properties of Leaf-Level Parts

What Are Shape Parts?

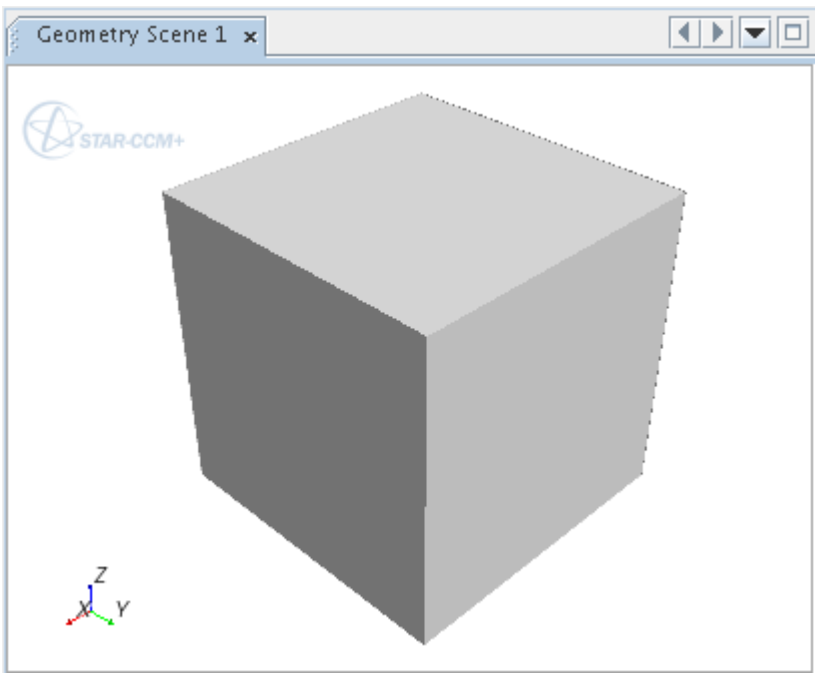
Shape parts are simple shape geometries (block, cone, cylinder, or sphere) that can be created directly within the **Parts** node without requiring use of the 3D-CAD modeler.

These simple parts can be used to provide exterior boundary domains for other parts, or to provide the input shapes for volumetric controls. They could also be used to generate a basic mesh directly.

The blue icons , , and , represent shape parts. Shape parts dimensions can be edited at any time. They expose the common properties for leaf-level parts along with other properties specific to their geometrical definition. Details on creating these parts are provided in the section, Creating Simple Shape Parts.

Block Part

This part is a regular six-sided block that can be created in a chosen coordinate system. Geometric properties are provided on the part node in addition to the common set for leaf-level parts.

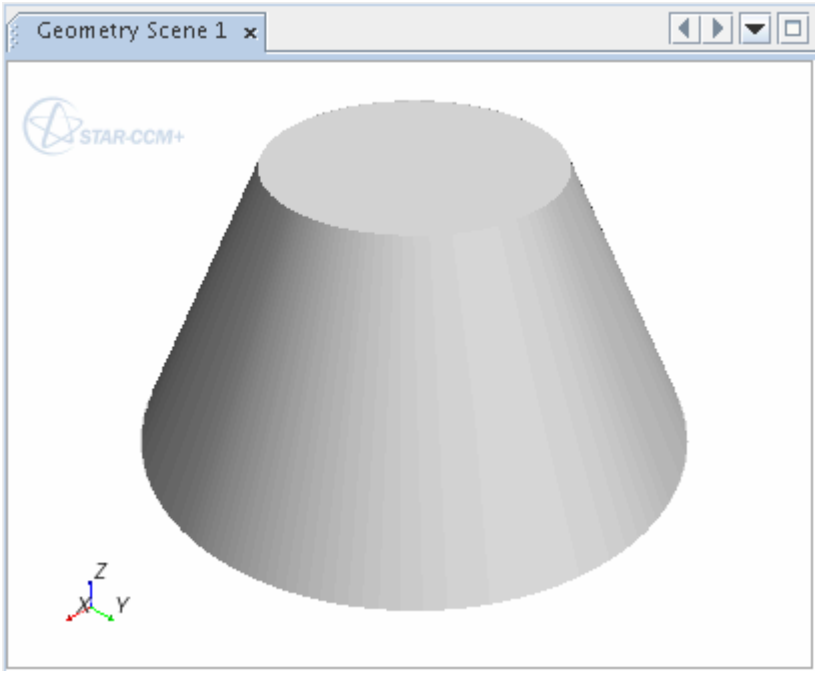


Additional Geometric Properties

Corner 1	Coordinates of the first corner of the block.
Corner 2	Coordinates of the diagonally opposite corner of the block.
Coordinate System	Coordinate system in which the corner coordinates are defined.

Cone Part

This part is a cone that is defined by the center coordinates of the start and end faces and their respective radii. It can be created in a chosen coordinate system. Geometric properties are provided on the part node in addition to the common set for leaf-level parts.

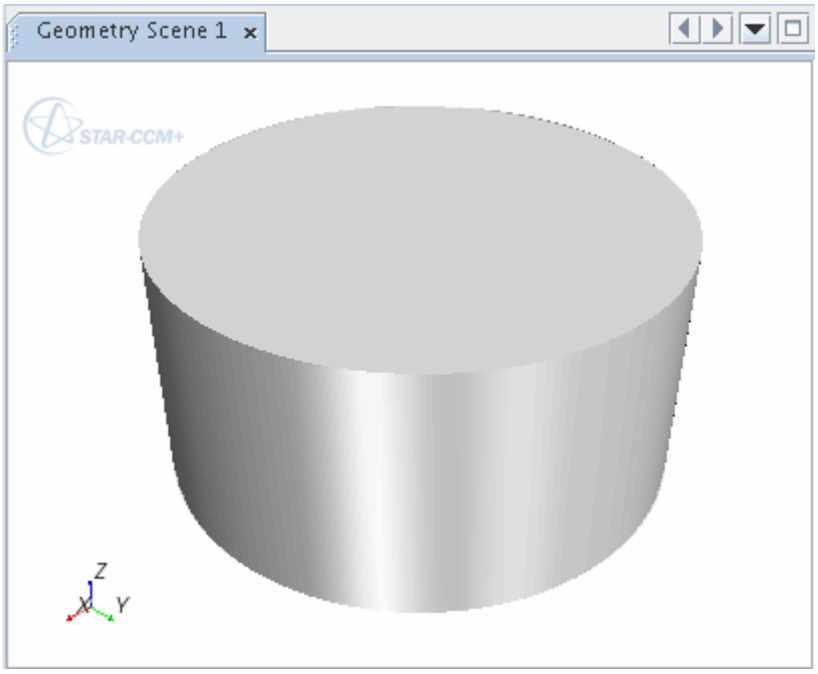


Additional Geometric Properties

Start Radius	Radius of the face whose center is the start coordinate.
End Radius	Radius of the face whose center is the end coordinate.
Start Coordinate	Face center coordinate of the start face.
End Coordinate	Face center coordinate of the end face.
Coordinate System	Coordinate system in which the start and end coordinates are defined.

Cylinder Part

This part is a cylinder that is defined by the center coordinates of the start and end faces and a constant radius. It can be created in a chosen coordinate system. Geometric properties are provided on the part node in addition to the common set for leaf-level parts.

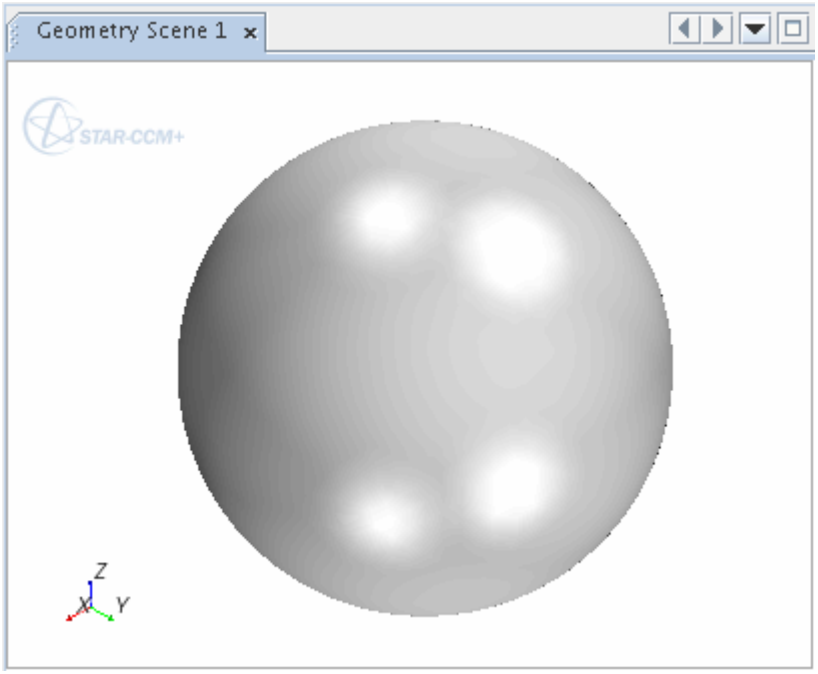


Additional Geometric Properties

Radius	The constant radius of the cylinder.
Start Coordinate	Face center coordinate of the start face.
End Coordinate	Face center coordinate of the end face.
Coordinate System	Coordinate system in which the start and end coordinates are defined.

Sphere Part

This part is a sphere that is defined by an origin and radius, and can be created in a chosen coordinate system. Geometric properties are provided on the part node in addition to the common set for leaf-level parts.



Additional Geometric Properties

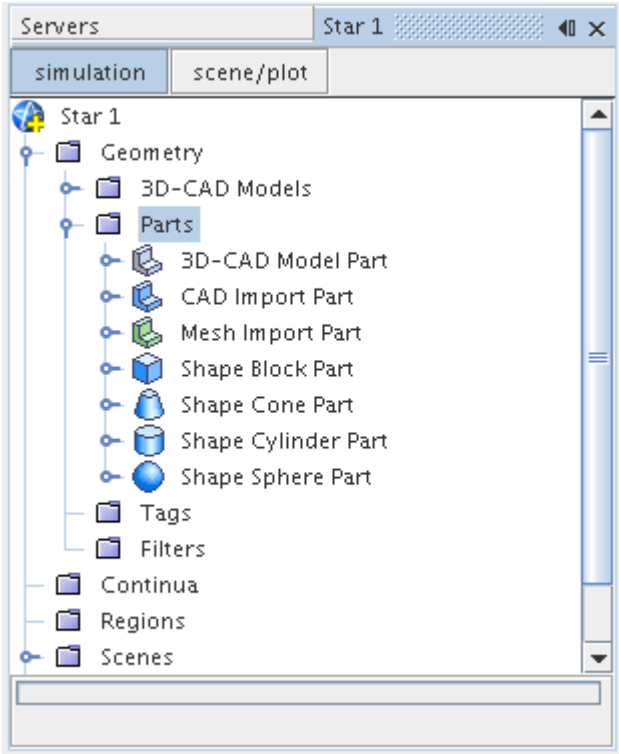
Origin	Origin of the sphere.
Radius	Radius of the sphere.
Coordinate System	Coordinate system in which the origin is defined.

What Are Mesh Parts?

Mesh parts are parts that originate from surface mesh data such as STL, NASTRAN, and PATRAN.

Green part icons, such as this one , represents these parts. The tessellation density of these parts cannot be modified after import, although these parts can still be modified manually using the surface repair tool.

Any combination of part types is allowed in STAR-CCM+ and each can be assigned to the same or different regions for meshing purposes. A summary example of all the different kinds of leaf-level parts is shown below:



Mesh Parts Properties

See Common Properties of Leaf-Level Parts

What Are Mesh Operation Parts?

Mesh Operation parts are parts that result from executing mesh operations.

The Mesh Operation Part icons are identical to the specific mesh operation icon that they are created from. If you modify the Mesh Operation, re-execute the Mesh Operation to update the Mesh Operation Part.

Mesh Parts Properties

Operation	Specifies the type of mesh operation that is used to create this part.

See Common Properties of Leaf-Level Parts

Common Properties of Leaf Level Parts

When selected, leaf-level parts expose the following properties and expert properties in the *Properties* window.

Common Part Properties

Region	The region to which this part has been assigned (if any). By default, the *Region* property is **None** meaning that the part has not yet been assigned to a region in readiness for meshing.

Expert Properties

Tags	List of tags (if any) that have been assigned to this part. These tags are used with filters in part selection operations. Click **[]** to edit the list directly.
Metadata	Metadata is user-defined data that is stored as key-value pairs. This field can be populated automatically for some import formats and can be used with filters for part selection operations. Click **{}** to edit the list directly.
Index	Geometry parts, part surfaces, part curves, and part contacts, all have a unique read-only index that you can use to help you identify specific objects. This feature is useful when creating field functions.
Contacts	This property is used to specify which other parts it is in contact with. Contacts are always reciprocal and can be converted into interfaces automatically when generating a new region during region assignment.
Descriptions	Provides a list of descriptions that are available for the part. For more information, see Mesh Descriptions.
Face count	The number of faces present on the part surfaces.

What Are Composite Parts?

Geometry parts, once converted, imported, or created within STAR-CCM+, can be reorganized if necessary to provide a more coherent tree structure.

Composite parts allow other parts to be grouped into assemblies and subassemblies for easier organization. Exploding a composite part allows the children of the part to become parent nodes instead.

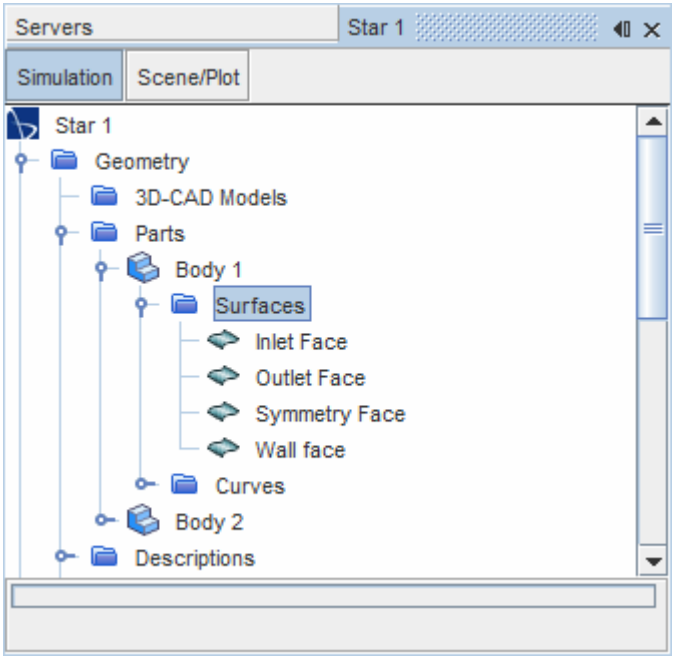
What Is a Part Surface?

A part surface is a collection of patches on a geometry part.

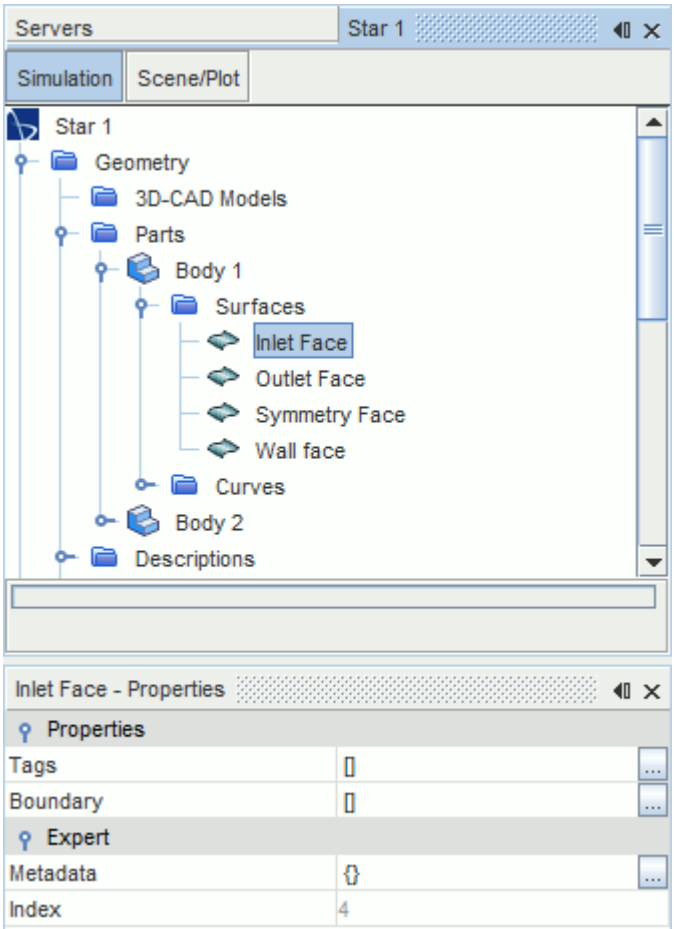
Patches on a part surface are composed of a collection of mesh faces. A patch can only belong to one part surface, although a part surface can contain more than one patch. However, it is possible to manipulate which part surface a patch belongs to using the split by patch feature.

The **Parts** > **[Part]** > **Surfaces** node is composed of one or more **[Part Surface]** nodes which make up the shape of the leaf-level part geometry. In order to be valid for use with the meshing tools, each part must be composed of at least one part surface.

To see the part surfaces that belong to a part, open the **[Part]** > **Surfaces** node, as shown below:



Each part surface node is represented using the icon. To view the properties of a part surface in the *Properties* window, select a **[Part Surface]** node:



Part Surface Properties

All part surface nodes have the following properties:

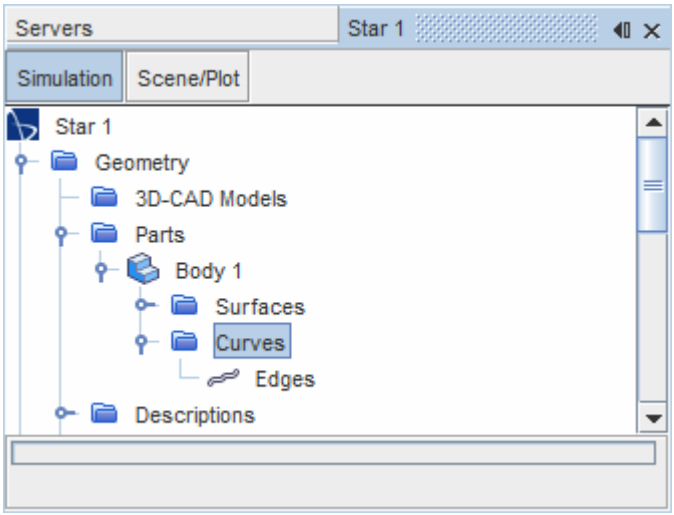
- tags specifier
- boundary specifier
- metadata specifier
- index specifier

What Is a Part Curve?

A part curve is a collection of feature edges on a geometry part.

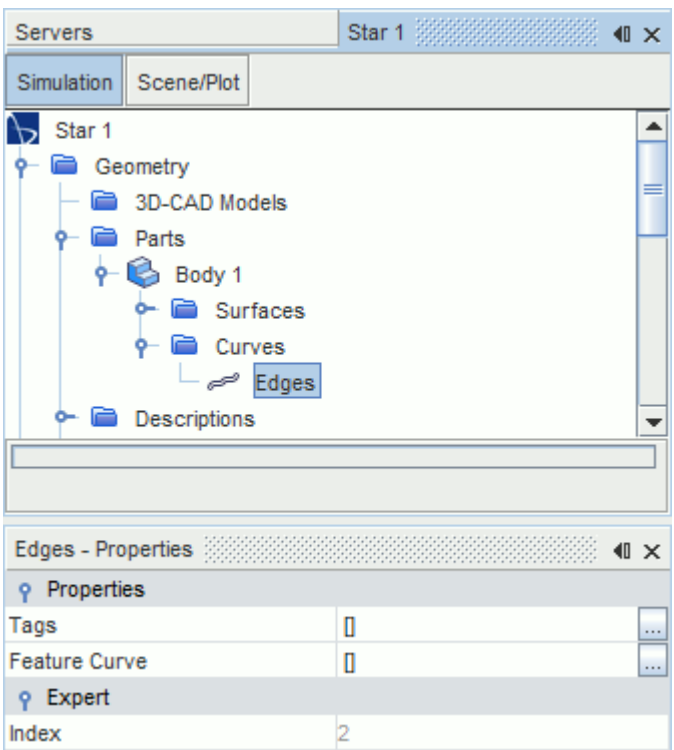
The **Parts** > **[Part]** > **Curves** node is composed of one or more **[Part Curve]** nodes which make up the sharp edges and features of the leaf-level part geometry. The choice of whether the part curve contains feature edges is made during the import process and is arbitrary depending on the requirements of the geometry.

To see the part curves that belong to a part, open the **[Part]** > **Curves** node, as shown below:



Each part curve node is represented using the icon. The part curves can be manipulated in different ways so that the assignment of feature edges to part curves can be made easier if necessary.

The properties of a part curve can be seen by selecting the **Curves** > **[Part Curve]** node.



Part Curve Properties

All part curve nodes have the following properties:

- Tags specifier
- Feature curve specifier

- Index specifier

Mark Part Curves Options Dialog

Mark Sharp Edges	When activated, edges that occur between two faces that meet at an angle equal to or less than the specified Sharp Edge Angle are marked as part curves.
Sharp Edge Angle	The angle above which all edges are marked as part curves. If you set the angle to 50, then any angle above 50 is marked as a part curve and any angle below 50 is not.
Mark Free Edges	When activated, all free edges are marked as part curves. See What Are Free Edges?
Mark Non-manifold Edges	When activated, all non-manifold edges are marked as part curves. See What Are Non-Manifold Edges?
Mark Patch Perimeters	Marks all edges that surround a patch (a collection of mesh faces on a part or region) as part curves.
Mark Part Surface Perimeters	Marks all edges that surround a part surface (a collection of patches on a geometry part) as part curves.
Mark Boundary Perimeters	Marks all edges that surround a boundary as part curves.
Automatically Fix Errors	Automatically fixes minor errors related to face quality, face proximity, and self-intersection.
Maintain Existing Edges	Edges that are already defined as part curves are maintained.

What Is Metadata?

The metadata specifier can be used to associate a key and value with the part.

A key is typically a property of the part, such as “material”, “color” or “grade”. The value is the respective quantity or identifier, such as “steel”, “blue” or “2”. The value can be either a string (word) or an integer. In order to create a valid metadata input, both a key and a value must be specified. Each geometry part can have one or more metadata that are specified for it.

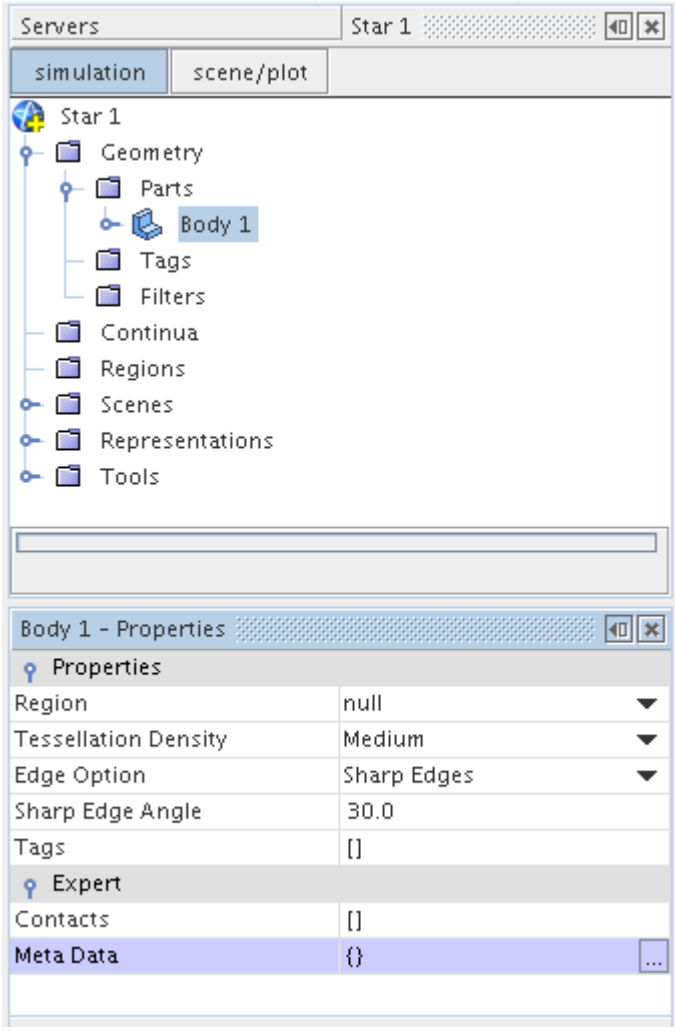
Metadata must be entered using the format `{key:value}`. For example:

```
{material:steel}
```

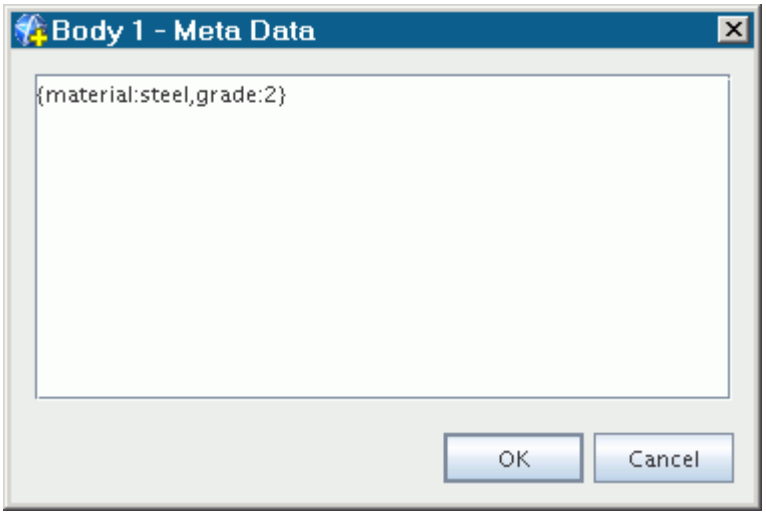
If more than one metadata specifier is required for the part or part surface, then a comma can be used to separate them. For example:

```
{material:steel,color:blue}
```

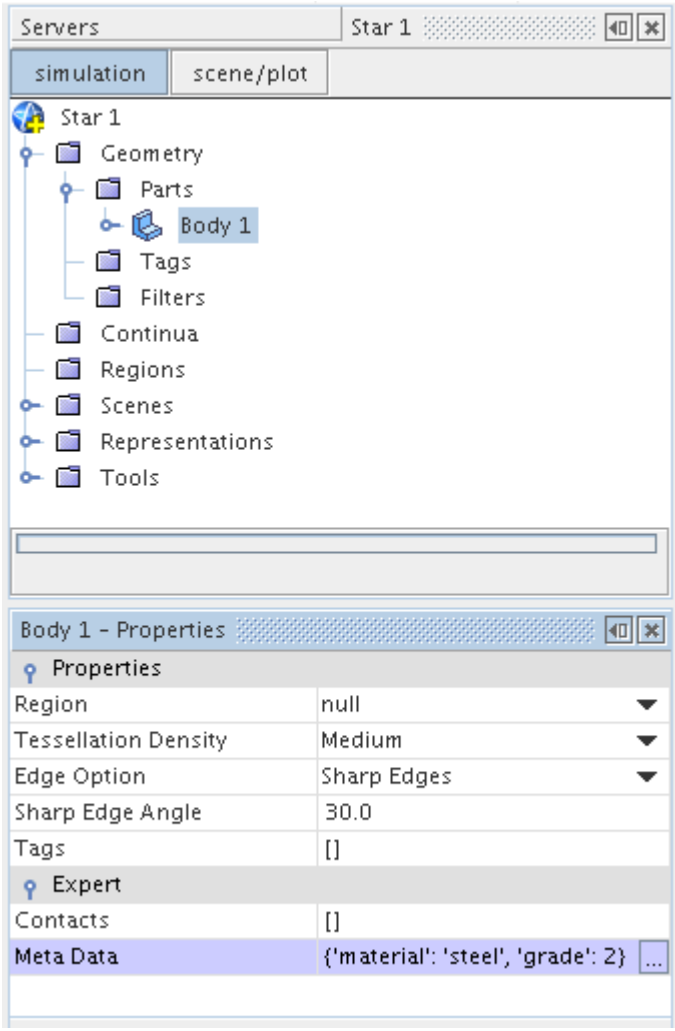
The *Properties* window lets you enter the metadata key and value for the part:



The specifier can be entered directly in the window (within the {} braces) or the ... button can be pressed to open the property customizer window. The metadata specifier can then be entered. For example:



Clicking **OK** applies the metadata text and reorganizes it if necessary to conform to the requirements of STAR-CCM+. The metadata can be reviewed afterwards in the part *Properties* window:



Notice that STAR-CCM+ automatically adds quotation marks `''` to `material`, `steel`, and `grade`, but not to the `2`. When quotation marks are not applied, this means that the `2` is an integer value. When quotation marks are applied to the value, for example `{'grade':'2'}`, this means that the `2` is a string. The same holds true for decimal values `{'grade':2.0}` and boolean values `{'conductive':true}`, where the quotation marks around the value are omitted.

Once the metadata information is entered, it can be used as part of a filter or series of filters in order to help identify and select parts at a later stage. The metadata specifier has no influence on the simulation and is provided as an organizational feature only.

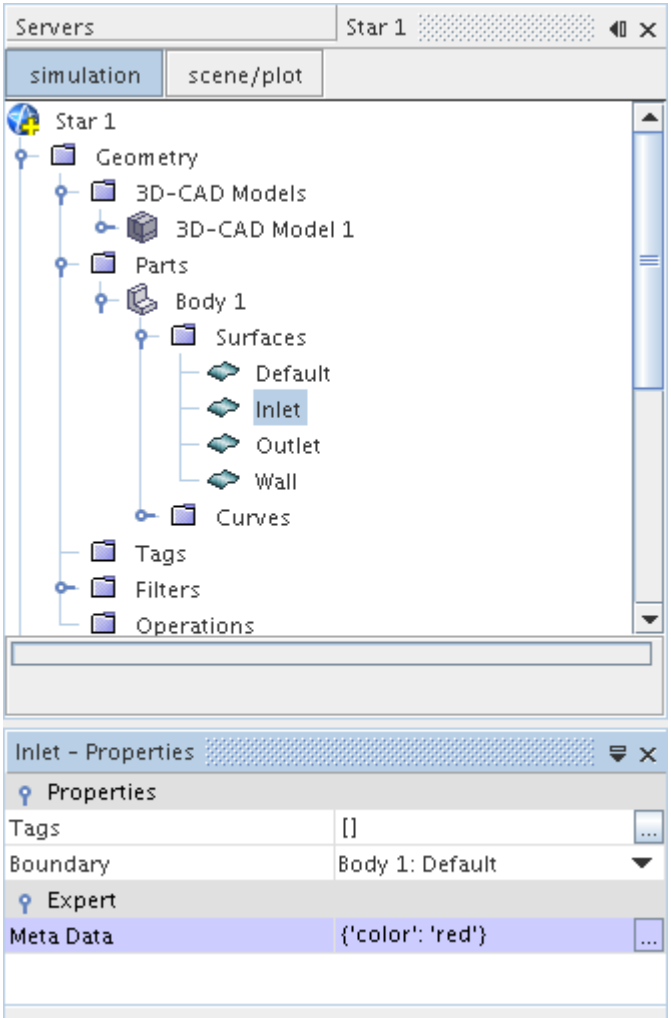
Contents:

Metadata on Part Surfaces

Metadata on Part Surfaces

Part surfaces can also have metadata.

The procedure for setting up metadata is the same as for parts.



Creating and Importing Parts

Geometry parts can be created in the following ways

Geometry parts can be created in the following ways:

- Converting 3D-CAD models to parts
- Importing surfaces as parts
- Importing multiple parts using an XML file
- Creating simple shapes within STAR-CCM+

Contents:

Creating and Updating 3D-CAD Parts
Importing Surfaces as Geometry Parts
Creating Simple Shape Parts

Creating and Updating 3D-CAD Parts

3D-CAD parts are created when a 3D-CAD model is converted to a geometry part.

3D-CAD parts are created when a 3D-CAD model is converted to a geometry part. The 3D-CAD model must first be created using the tools or import options that are provided in the 3D-CAD module, for which full documentation is provided in the section, Using 3D-CAD. If the 3D-CAD model is modified after its corresponding 3D-CAD part has been created, it will be necessary to update the 3D-CAD part to reflect the changes.

Contents:

Updating 3D-CAD Parts

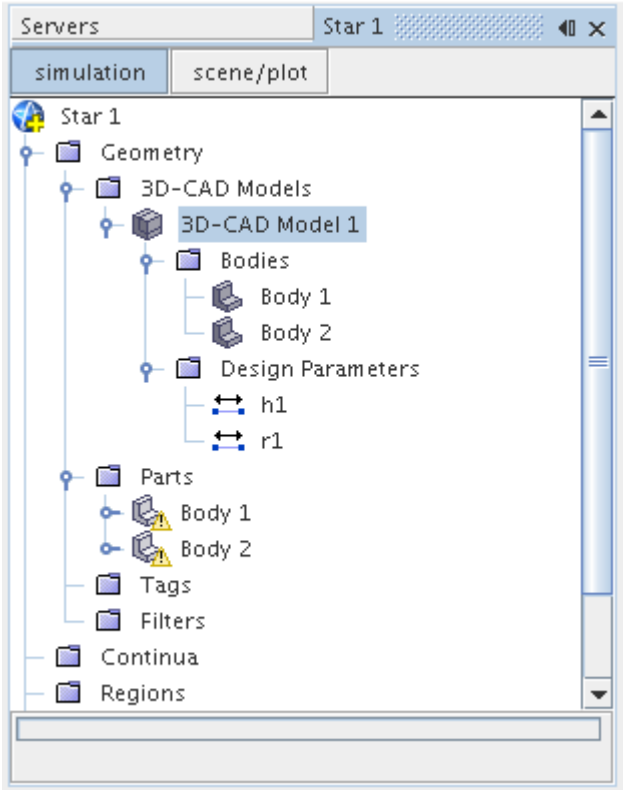
Updating 3D-CAD Parts

After making changes in a 3D-CAD model, update any existing geometry parts that are based on the 3D-CAD model.

Update leaf-level parts that are created from a 3D-CAD model when either of the following occurs:

- the originating 3D-CAD model was edited
- the design parameters on which the original 3D-CAD model bodies are based on change for any reason.

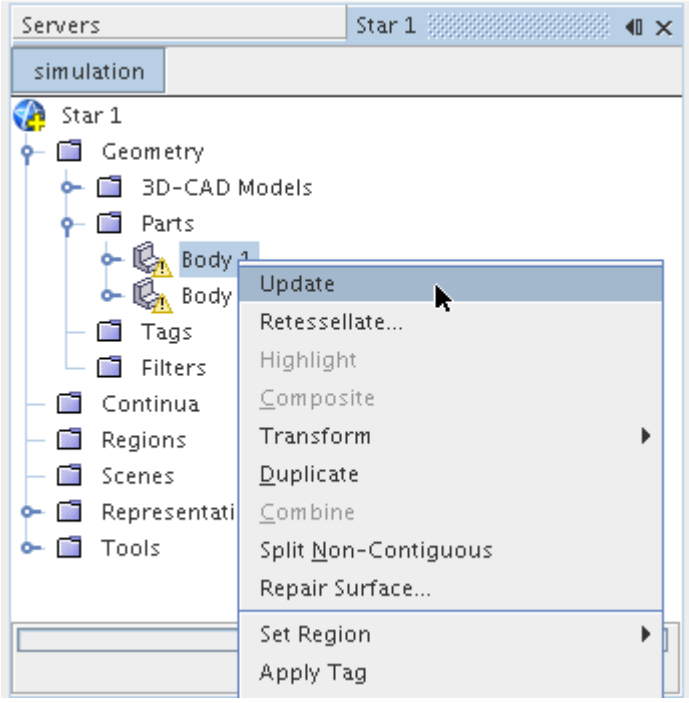
In the latter instance, 3D-CAD model parts are flagged with the “out-of-date” ⚠ badge once the 3D-CAD model bodies have been updated themselves after the design parameter change has been made. Examples of two out-of-date flagged parts are shown below:



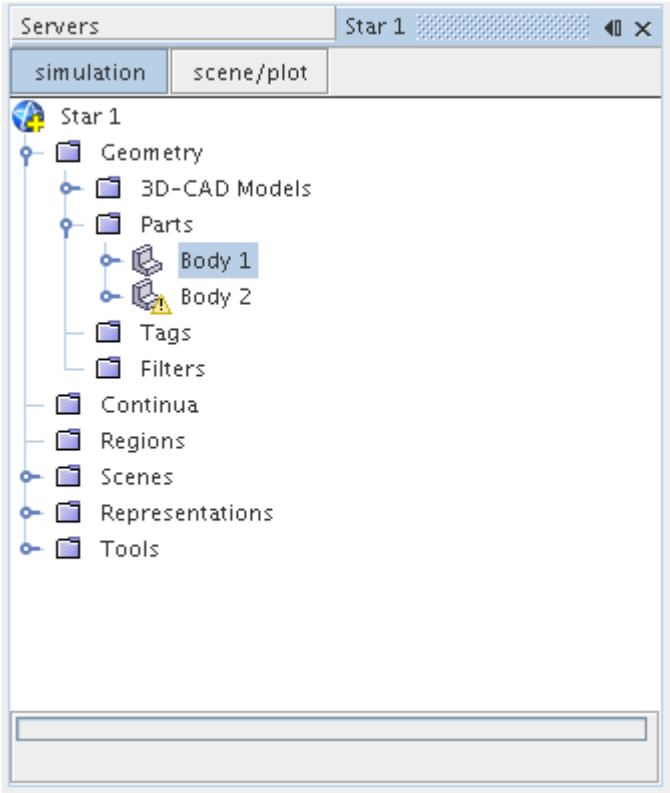
In the above example, the two bodies (**Body 1** and **Body 2**) belonging to the **3D-CAD Model 1** node were updated after the design parameters **h1** and **r1** were changed.

In order to obtain the correct geometry, update **Body 1** and **Body 2** in the **Parts** node.

1. This update can be done by selecting the parts node of interest, right-clicking, and selecting **Update** from the pop-up menu:



The geometry part is now updated with the correct dimensions and retessellated. The “out-of-date” warning badge is removed, indicating that the part is now up-to-date.



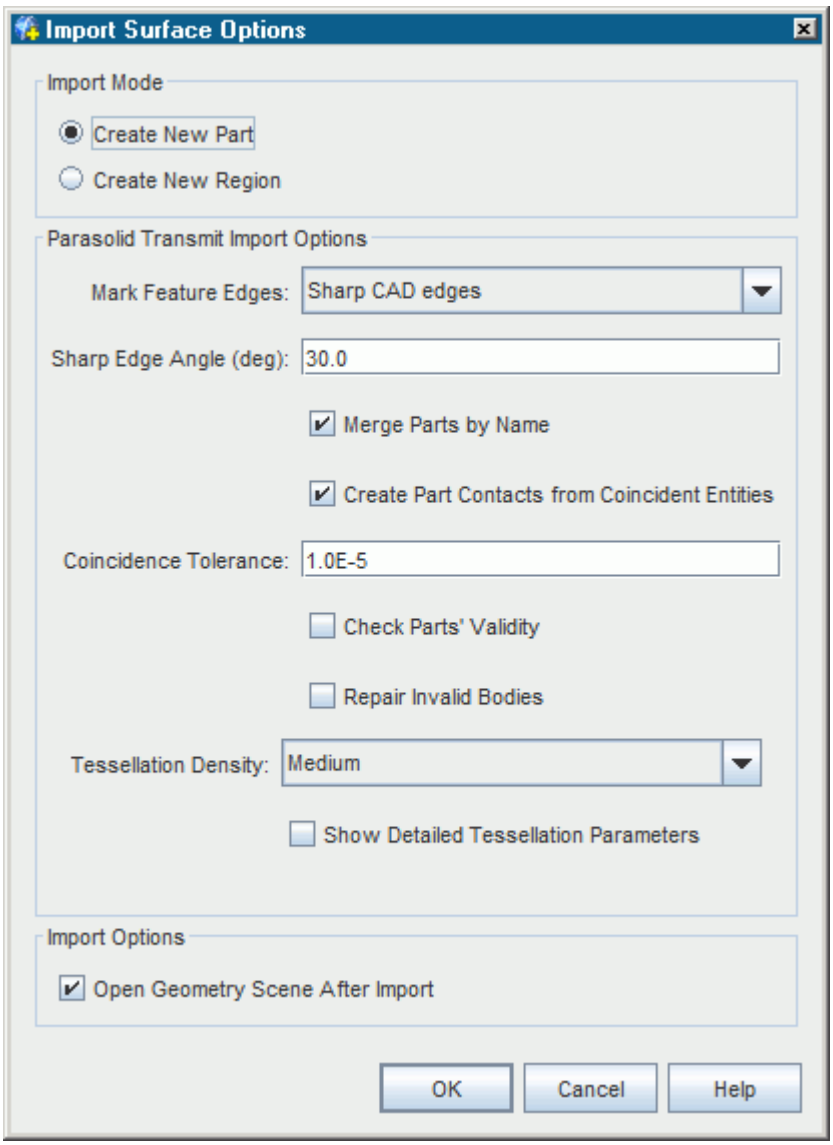
Multiple part selections can be updated using this method.

The initial surface representation for meshing (if it exists) must be reinitialized after updating 3D-CAD parts. Stale 3D-CAD parts and the initial surface representation (if present) are automatically updated on invoking either of the surface or volume meshing pipelines.

Importing Surfaces as Geometry Parts

The option to create a geometry part is selected by default when importing surface data.

Selecting the **Create New Part** option within the *Import Mode* group box presents the import options relevant to the file type being imported. An example for importing CAD data is shown below:



Depending on the type of surface file, the import process results in one or more leaf-level parts being created. Composite parts (assemblies) are also created when the original file supported and used this option. CAD

parts are created when the original files contain neutral format data or native CAD data. Mesh parts are created when the original files contain surface mesh data.

The import parts process reads all CAD parts before tessellation begins. If a surface on a part fails to tessellate (due to invalidities in the CAD model), the tessellation process stops for that part and continues for the next part. All parts are imported into STAR-CCM+, but those parts which failed to tessellate completely do not display correctly.

The Mark Feature Edges and Tessellation Density options are described within the section, Importing Surface Data into STAR-CCM+. When the **Create Part Contacts from Coincident Faces** option is activated, part contact data is generated for parts that share coincident faces that are within the *Coincident Face Tolerance* distance from their neighbors. Do not use this option to create contacts or interfaces between parts that have large gaps between them. If parts have large gaps between them, move one part closer to the other using a translation operation. If the **Merge Parts by Name** option is selected, any two bodies in the imported assembly with the same name are merged into one part automatically. If the option is deactivated, each body in the assembly becomes its own part in STAR-CCM+ regardless of its name.

To manage the hierarchy of various components to better suit the analysis in hand, geometry parts can be reorganized once they have been imported. Once organized, the meshing models can be initialized to provide a starting initial surface representation that is based on the currently selected geometry parts. In general, all the native CAD file formats plus the Parasolid Transmit format can maintain the same assembly hierarchy as the original CAD program that produced the geometry in the first place. Additionally, the part names are also carried over for these formats and, under certain circumstances, the names that are attached or tagged to faces are also translated. IGES data does not carry part names but STEP data will under certain circumstances. In this respect, the Parasolid Transmit file is the better export option out of all of the neutral file formats in terms of maintaining the organizational data structure.

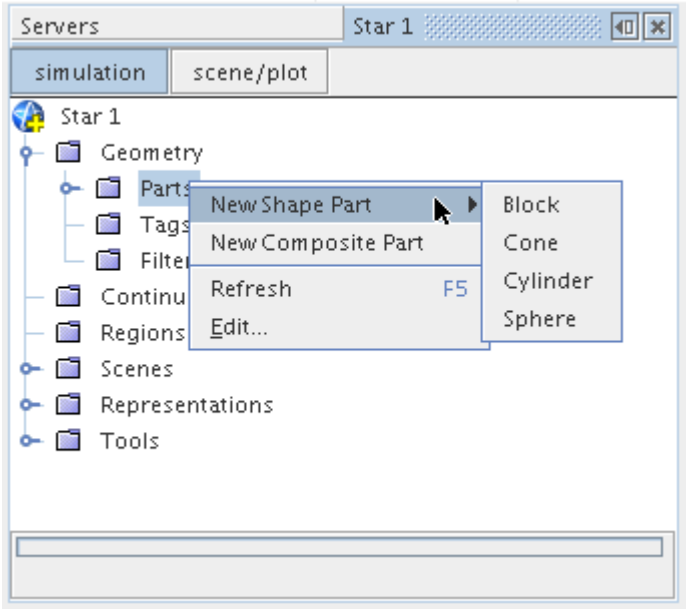
The import of a geometry part also creates a **Geometry** representation in the **Representation** node. This node has no properties and is simply a place holder for indicating that geometry parts exist.

Creating Simple Shape Parts

Shape parts are normally created at any point after a geometry part has been imported so that the exterior domain can be sized using the imported part as a reference.

They can be added at any time before meshing initialization is performed.

To create a shape part, right-click the **Geometry** > **Parts** node and select **New Shape Part** from the pop-up menu:



Four different shape parts are listed:

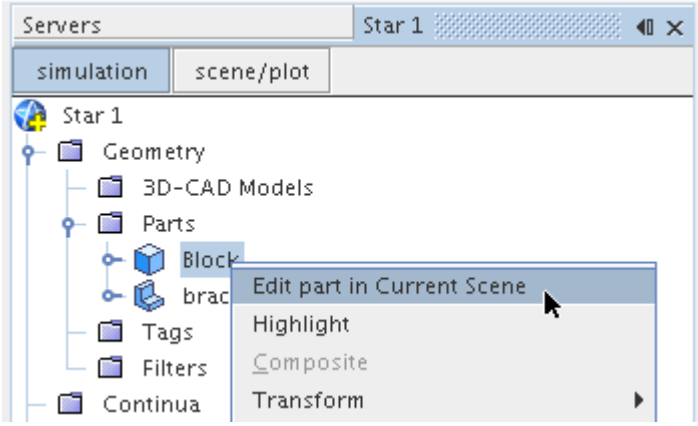
- Block
- Cone
- Cylinder
- Sphere

Select the option that corresponds with the shape that you want to create. If a valid scene is open, then the *edit* tab appears in the explorer pane allowing the properties of the shape to be entered, along with other applicable options. The graphical point tool is also displayed allowing you to set the coordinate values graphically if preferred.

Once all the properties are specified, pressing the **Create** button generates the part. You can create additional parts of the same shape by changing the input values and pressing **Create** again. A new leaf-level part with CAD association is added to the **Parts** node for each new shape that is created.

Note: Only simple shape parts that are created using STAR-CCM+ v9.06 or later have CAD association. However, if you use STAR-CCM+ v9.06 to change the parameters of a simple shape part created in STAR-CCM+ v9.04 or earlier, the simple shape part is given CAD association.

The shape part can be re-edited at any time by selecting the shape part node, right-clicking, and selecting **Edit Part in Current Scene**:



Alternatively, the values and options that are used for each shape can be edited directly using the *Properties* window for each node, as described in the section on each part shape. You can enter strings with mixed units (see Entering Mixed Units). If a simple shape part is modified using surface repair, the CAD association is updated.

If a valid scene is not open when selecting the shape part to create, then a normal dialog appears instead of the *edit* tab. The appropriate properties and selections as described above are prompted for and the shape can be created as normal.

Contents:

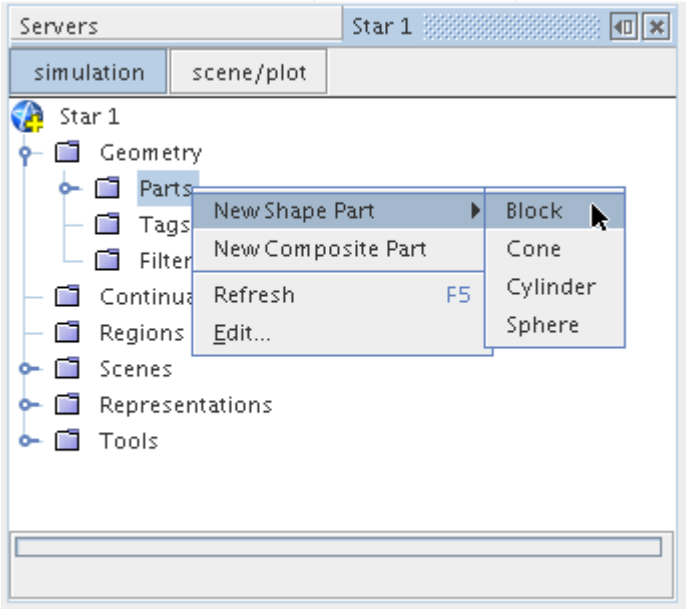
Creating a Block
Creating a Cone
Creating a Cylinder
Creating a Sphere

Creating a Block

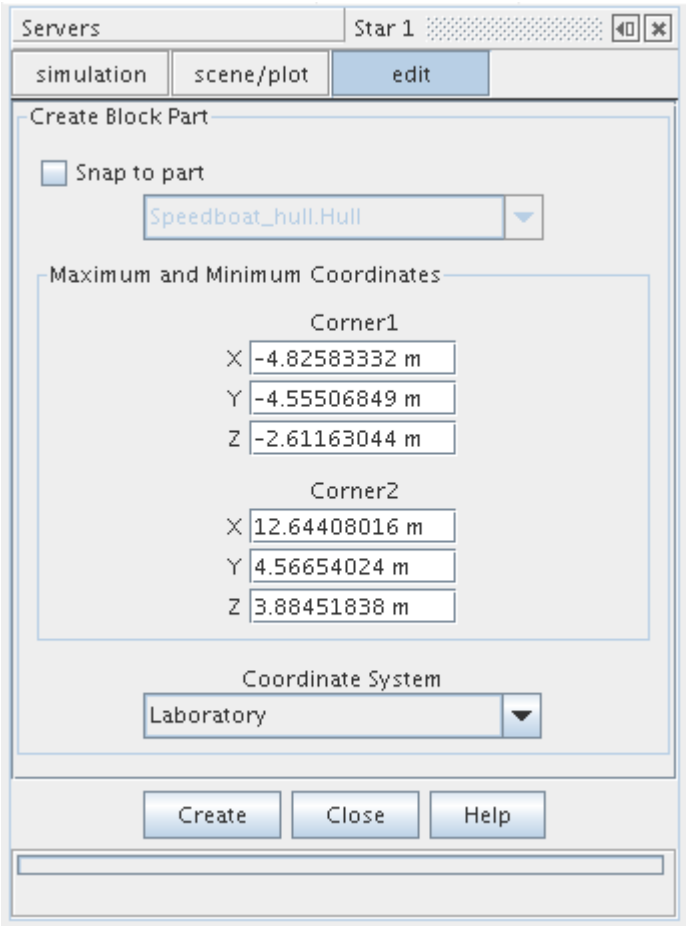
This section describes how to create a block part.

To create a block part:

1. Right-click the **Parts** node and select **New Shape Part** > **Block** from the pop-up menu:

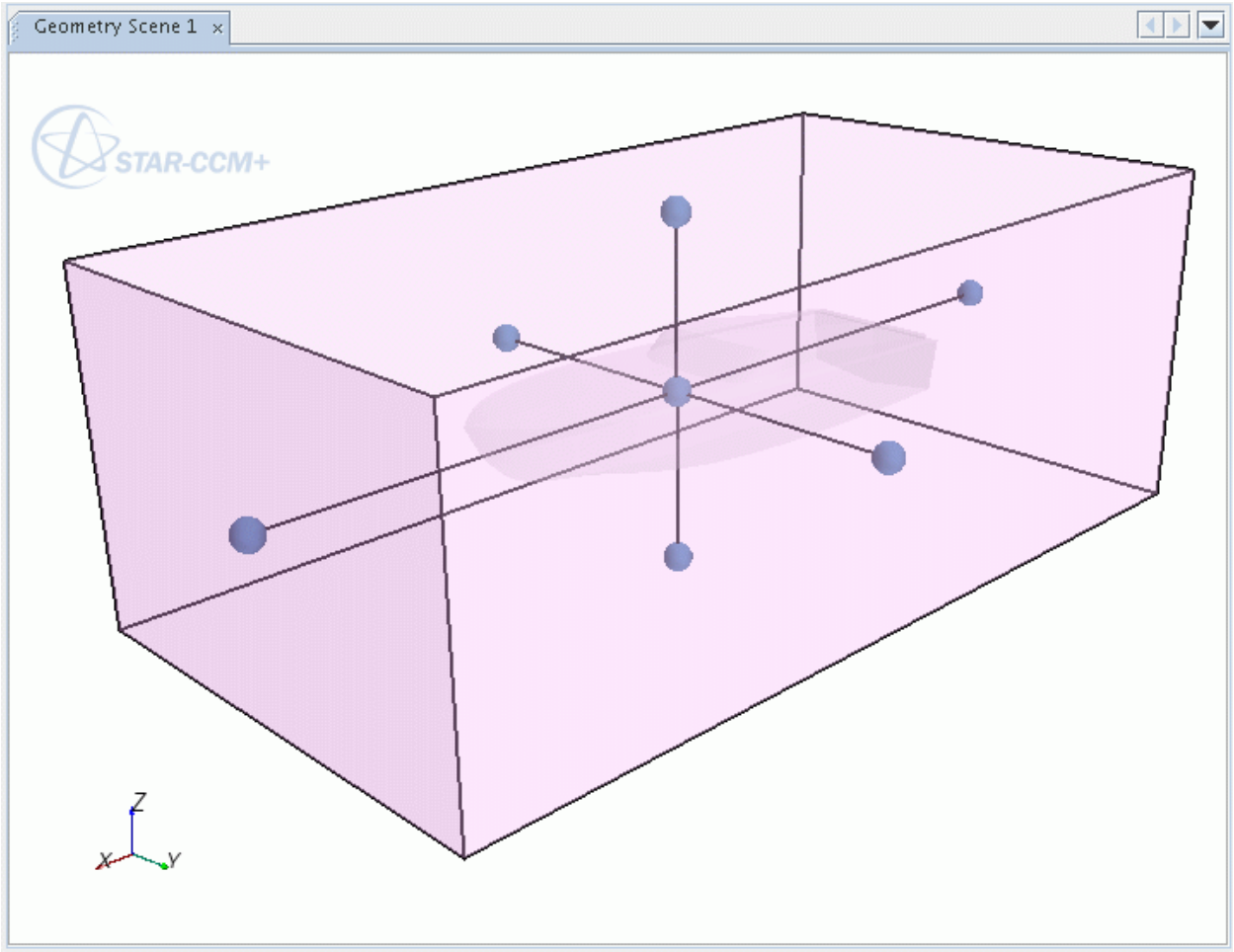


The *Create Block Part* window appears in the *edit* tab in the explorer pane:



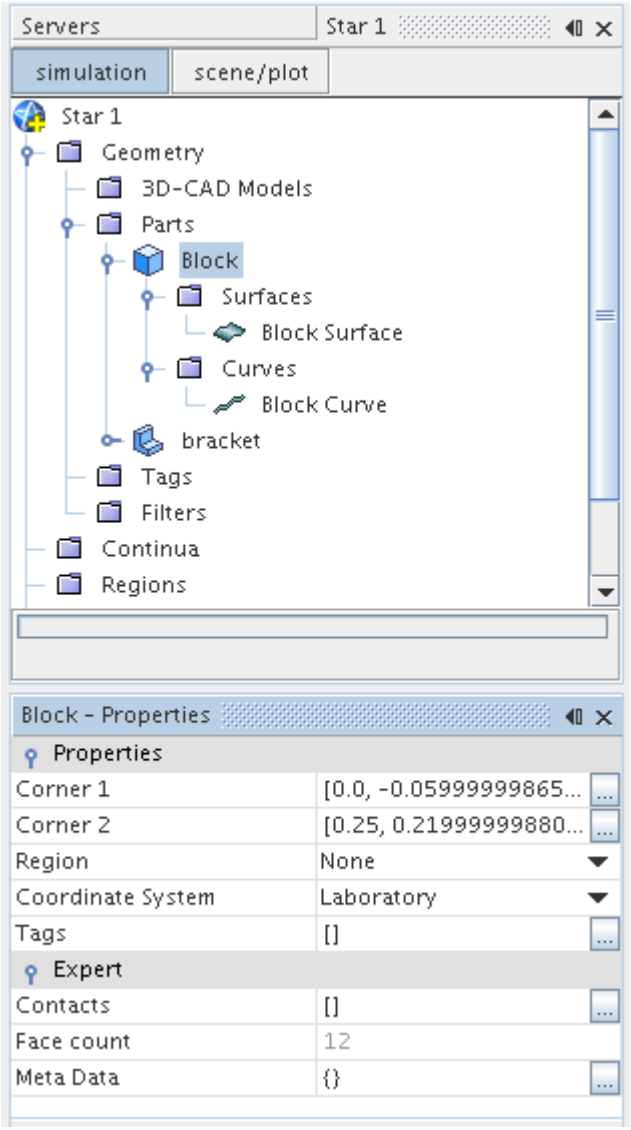
2. To define the block opposite corners, supply the maximum and minimum X, Y and Z (or equivalent) coordinates in the current **Coordinate System**.

   These coordinates can be entered manually into the window or by using the graphical based point tool, which is automatically displayed when the *edit* tab is opened. The spherical balls in the center of each block face can be dragged using the mouse to extend the block to the appropriate dimensions, as shown below:



3. Right-clicking anywhere on the magenta colored box and moving vertically resizes all the block dimensions.
4. Clicking the box using the middle mouse button translates the box in any direction while maintaining its current size.
5. To generate the part shape, click **Create** once the coordinates have been set. Additional block parts can be created by changing the coordinate values and clicking **Create** again.
6. Click **Close** when all operations are complete.

   For each block created, a new part node appears in the **Parts** folder:



7. Selecting the block part node exposes its properties in the *Properties* window.

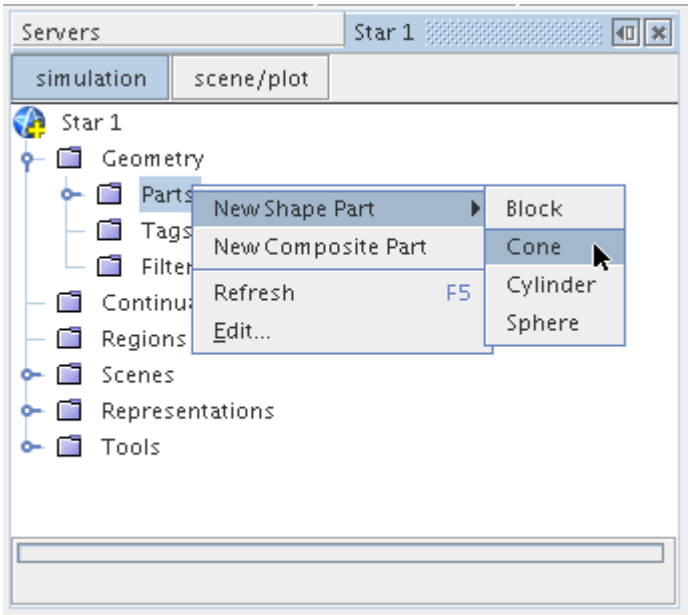
By default, block parts are composed of one surface node and one curve node, both of which can be manipulated if necessary.

Creating a Cone

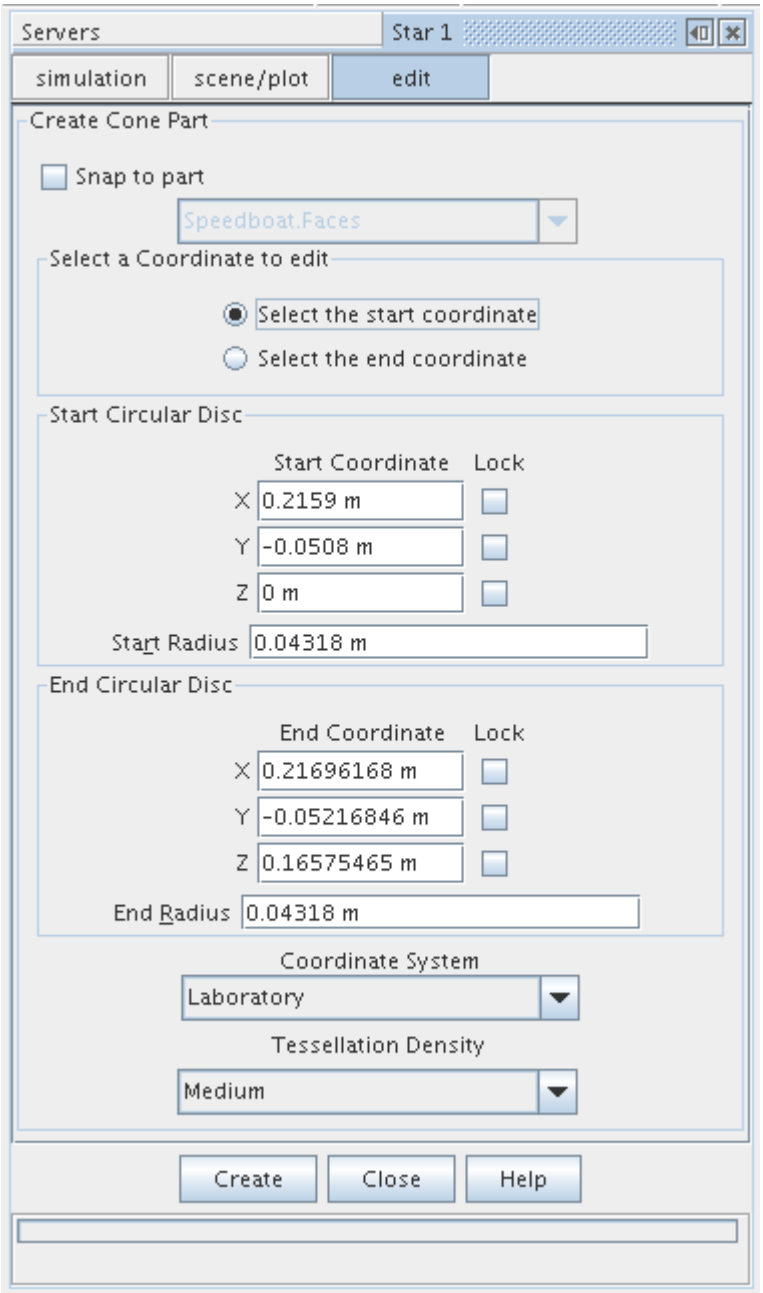
This section describes how to create a cone part.

To create a cone part:

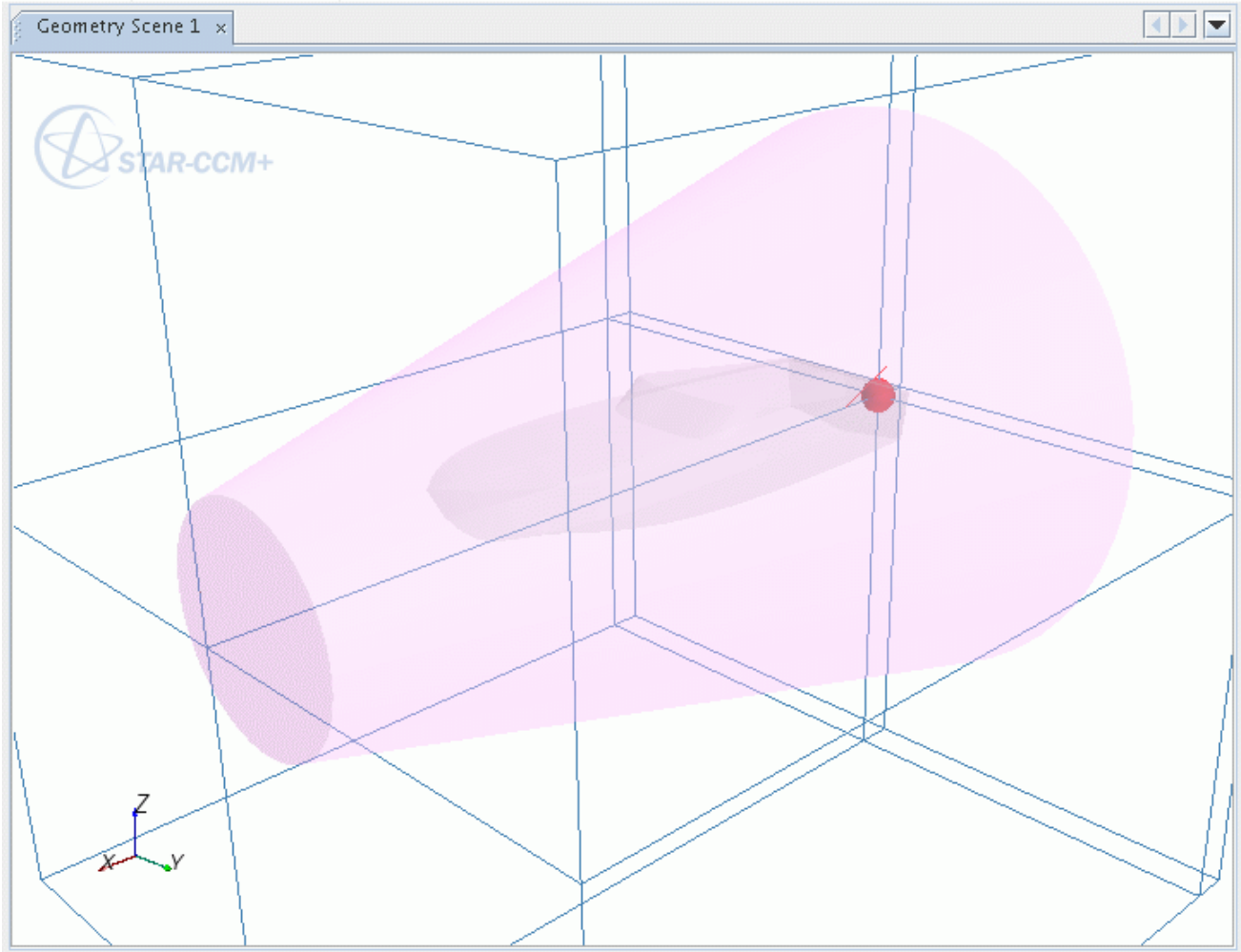
1. Rright-click the **Parts** node and select **New Shape Part** > **Cone** from the pop-up menu:



The *Create Cone Part* window appears in the *edit* tab in the explorer pane:

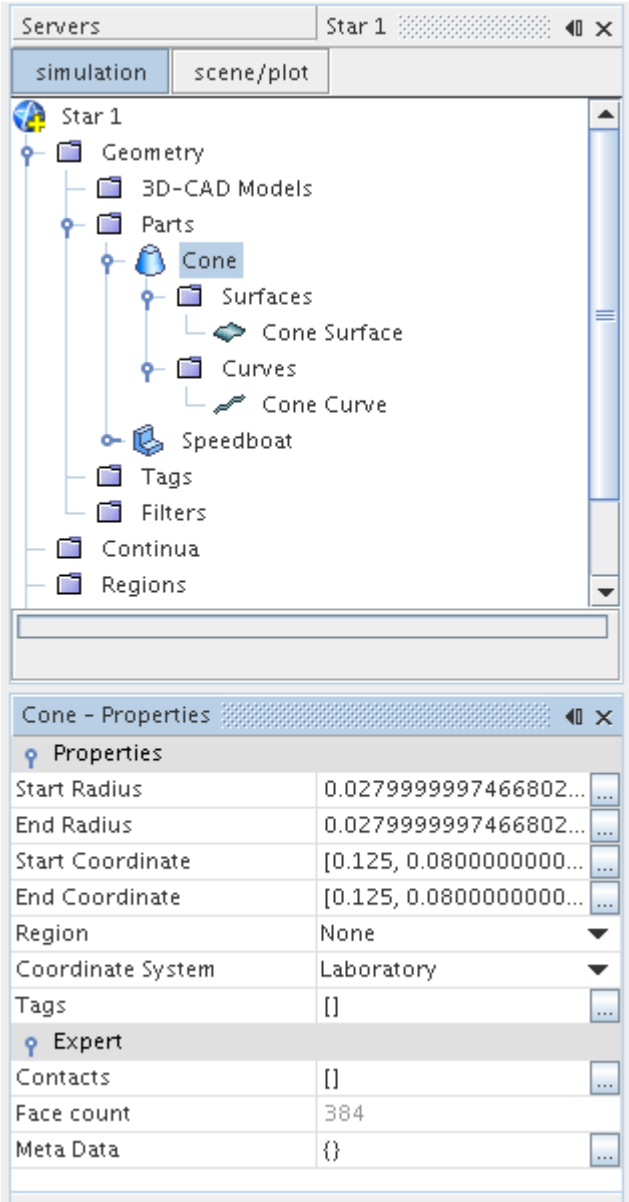


Two separate locations (Start Circular Disc and End Circular Disc) and two radii (**Start Radius** and **End Radius**) based on either the global cartesian or a local coordinate system must be defined. These values can be entered manually into the window or by using the graphical based point tool, which is automatically displayed when the *edit* tab is opened. The spherical balls at the center of each disc face can be dragged using the mouse to extend the cone to the appropriate dimensions, as shown below:



2. Once the coordinates and radii have been set, press the **Create** button to generate the part shape at the specified tessellation density. Additional cone parts can be created by changing the coordinate values and pressing **Create** again.
3. Press the **Close** button when all operations are complete.

   For each cone created, a new part node appears in the **Parts** folder:



4. Selecting the cone part node exposes its properties in the *Properties* window.

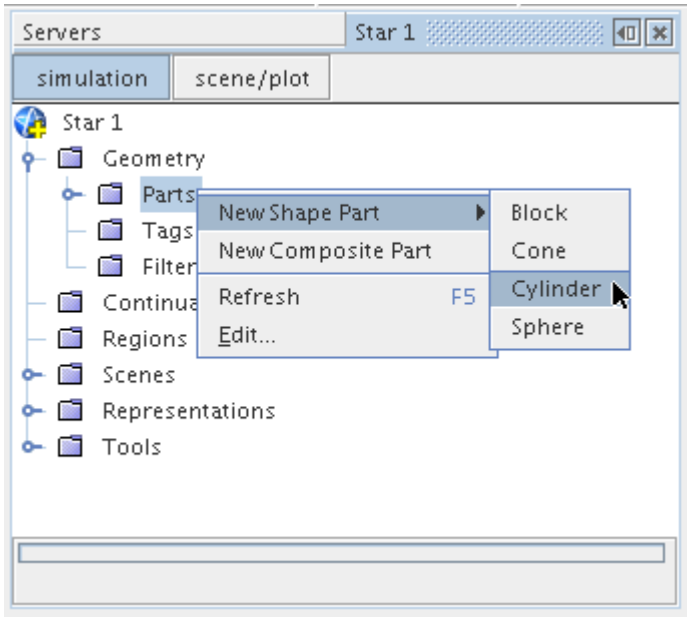
By default, cone parts are composed of one surface node and one curve node, both of which can be manipulated if necessary.

Creating a Cylinder

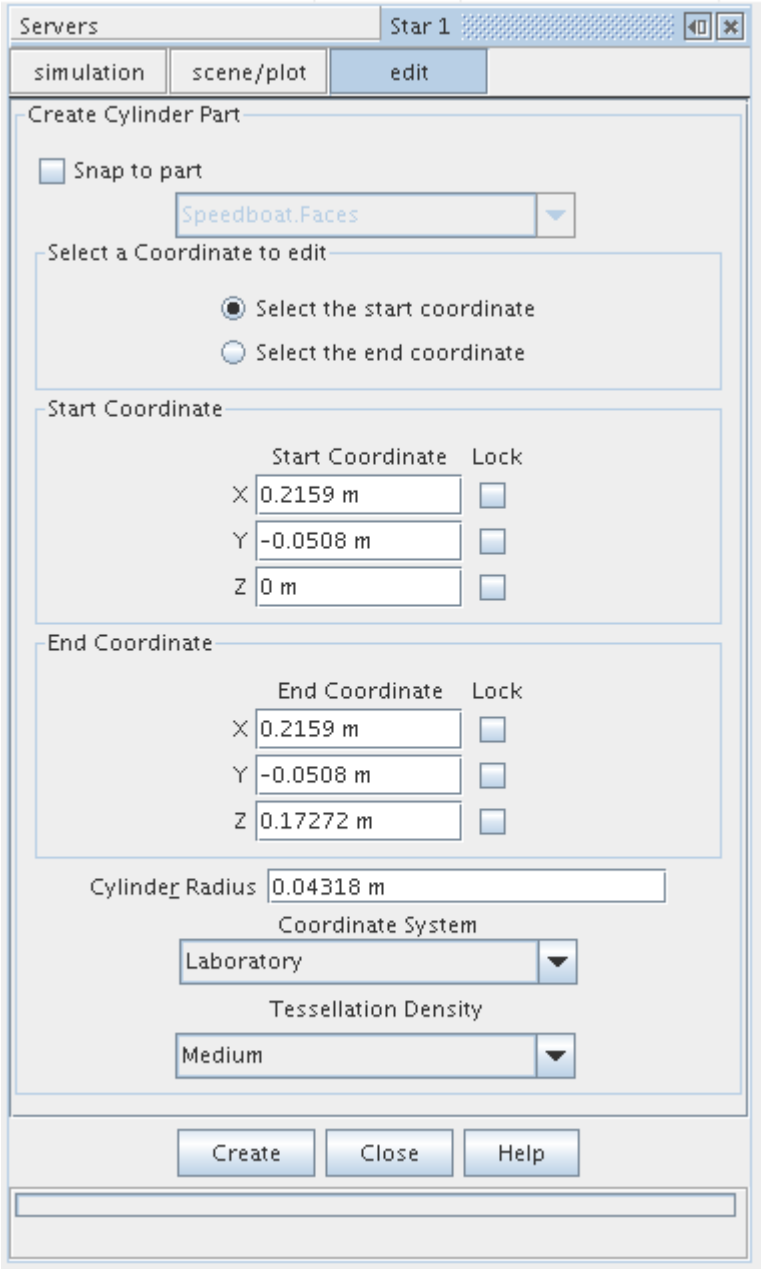
This section describes how to create a cylinder part.

To create a cylinder part:

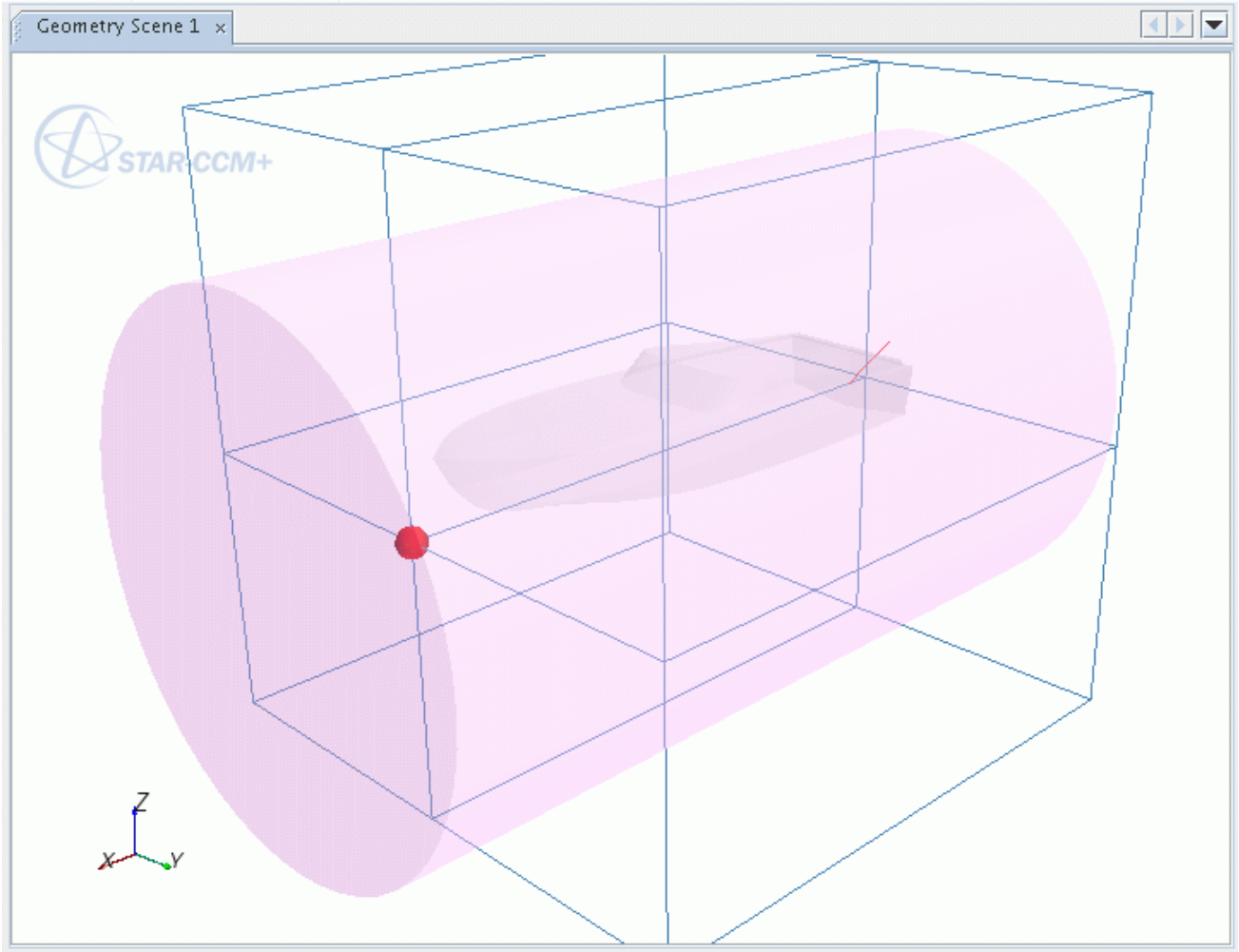
1. Right-click the **Parts** node and select **New Shape Part** > **Cylinder** from the pop-up menu:



The *Create Cylinder Part* window appears in the *edit* tab in the explorer pane:



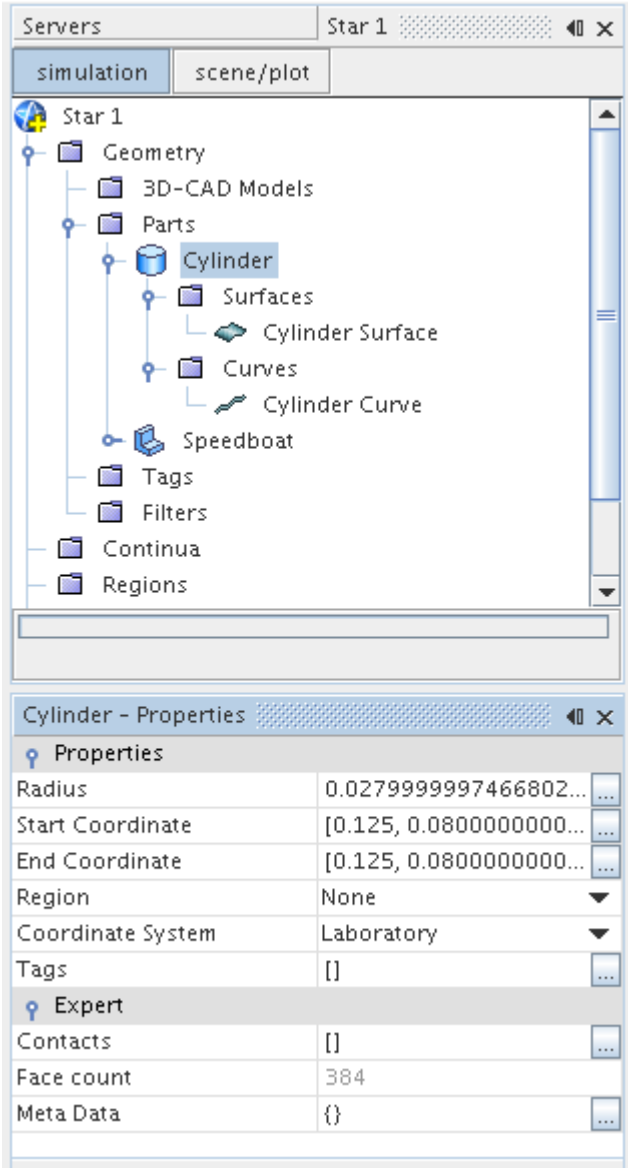
Two separate locations (Start Coordinate and End Coordinate) and the **Cylinder Radius** must be supplied based on either the global cartesian or a local coordinate system. These values can be entered manually into the window or using the graphical based point tool, which is automatically displayed when the *edit* tab is opened. The spherical balls at the center of each disc face can be dragged using the mouse to extend the cylinder to the appropriate dimensions, as shown below:



To generate the part shape at the specified tessellation density:

2. Click **Create** once the coordinates and radius have been set. Additional cylinder parts can be created by changing the coordinate values and clicking **Create** again.
3. Click **Close** button when all operations are complete.

For each cylinder created, a new part node appears in the **Parts** folder:



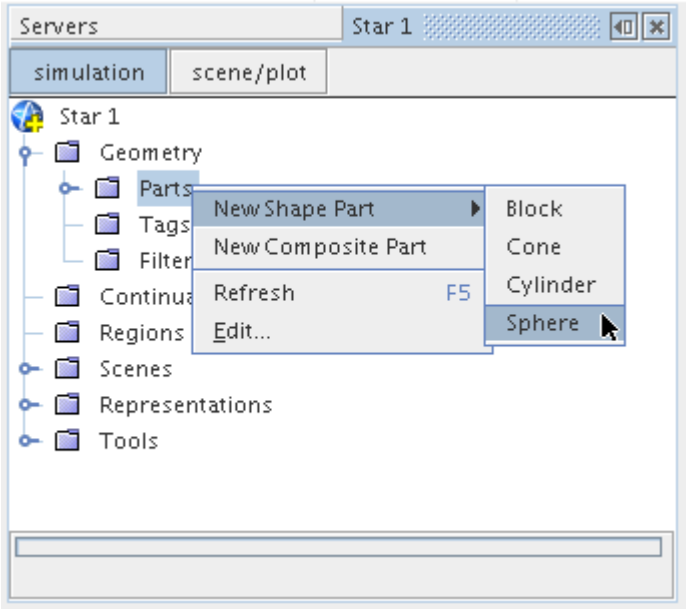
Selecting the cylinder part node exposes its properties in the *Properties* window. By default, cylinder parts are composed of one surface node and one curve node, both of which can be manipulated if necessary.

Creating a Sphere

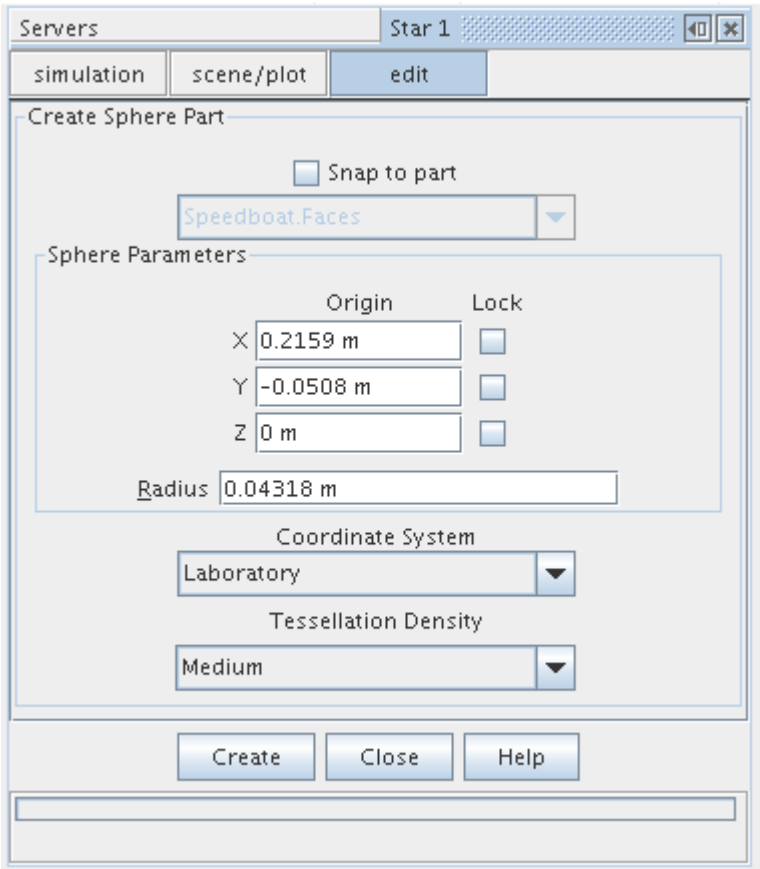
This section describes how to create a sphere part.

To create a sphere part:

1. Right-click the **Parts** node and select **New Shape Part** > **Sphere** from the pop-up menu:

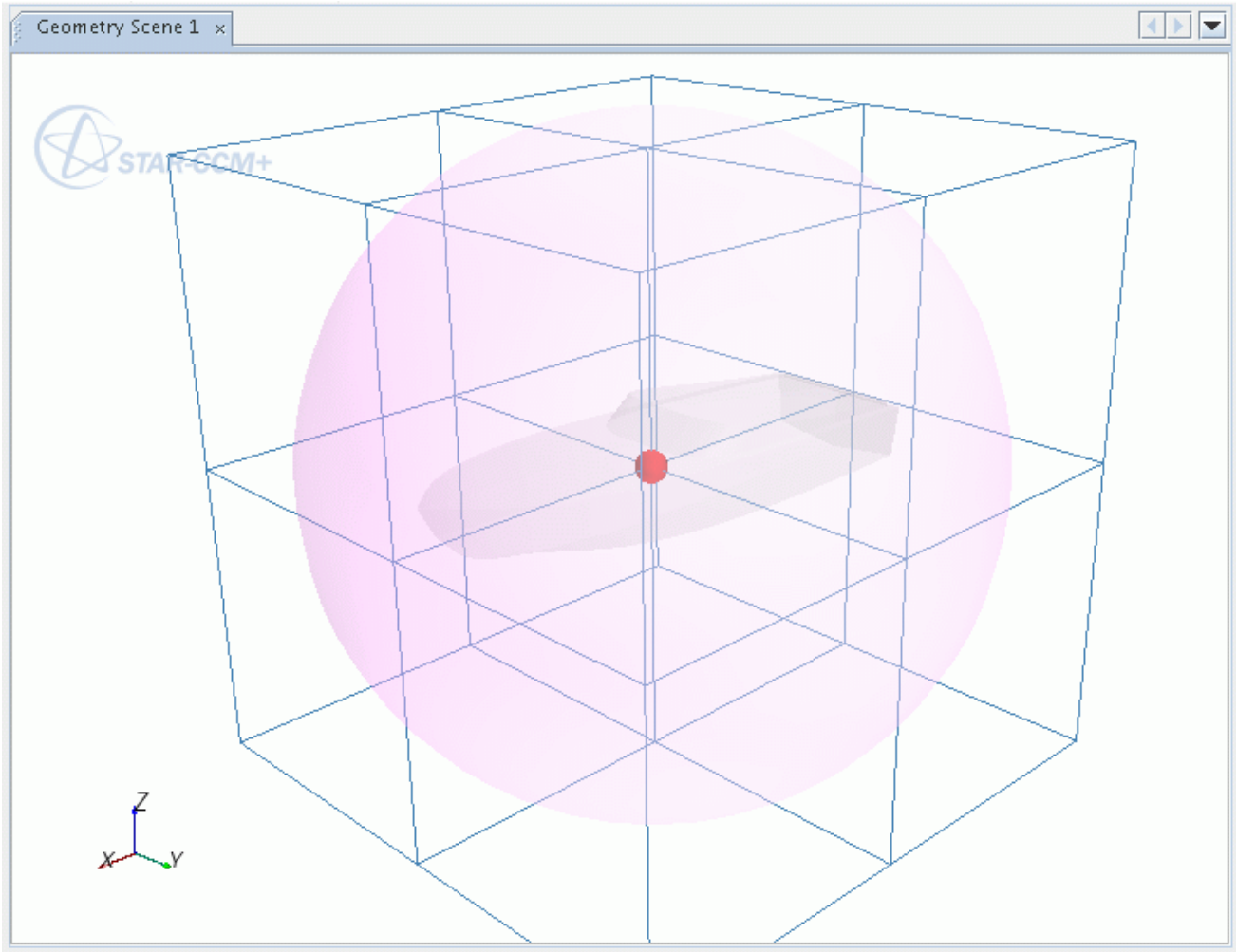


The *Create Sphere Part* window appears in the *edit* tab in the explorer pane:



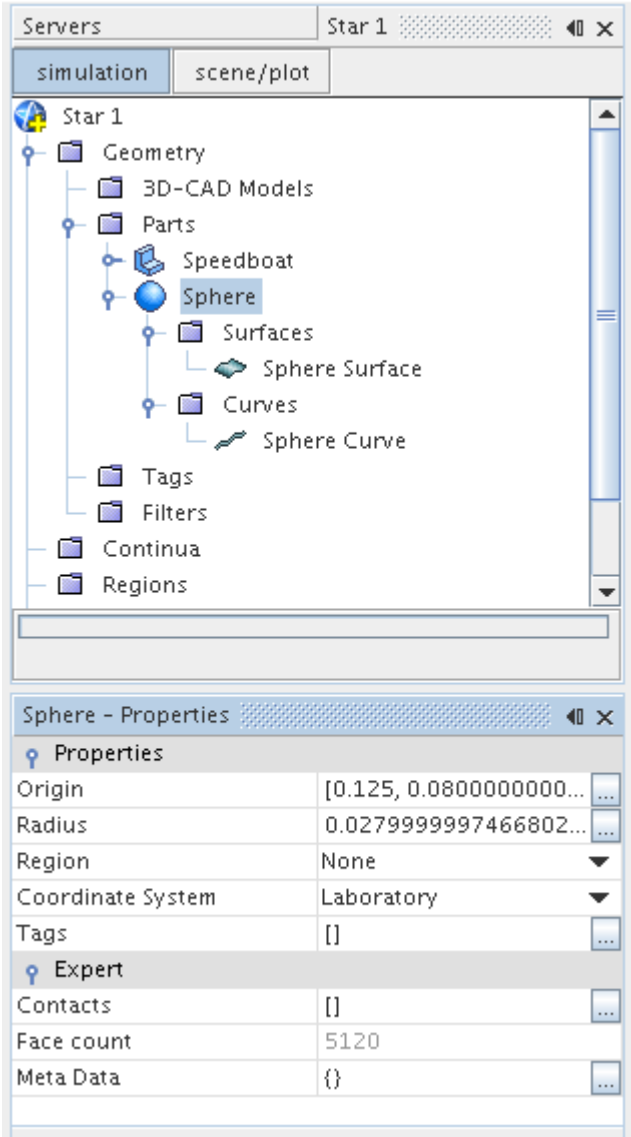
The **Origin** of the sphere and the **Radius** must be defined, based on either the global cartesian or a local coordinate system. The point tool allows the origin to be selected graphically by dragging the ball to the

appropriate location, or alternatively, the coordinates can be entered directly into the window. An example of a sphere shape part during the creation process is shown below:



2. Once the coordinates and radius have been set, press the **Create** button to generate the part shape at the specified tessellation density. Additional spherical parts can be created by changing the coordinate values and pressing **Create** again.
3. Press the **Close** button when all operations are complete.

For each sphere created, a new part node appears in the **Parts** folder:



Selecting the sphere part node exposes its properties in the *Properties* window. By default, sphere parts are composed of one surface node and one curve node, both of which can be manipulated if necessary.

Organizing Assemblies Using Composite Parts

Composite parts allow other parts to be grouped into assemblies and subassemblies for easier organization.

Exploding a composite part reverses this process. These operations are described as follows:

- Creating composite parts (assemblies)
- Exploding composite parts

Contents:

Creating Composite Parts (Assemblies)
Exploding Composite Parts

Creating Composite Parts (Assemblies)

In order to help organize leaf-level parts or other composite (subassembly) parts, you can create a composite part.

Two general methods to create composite parts are:

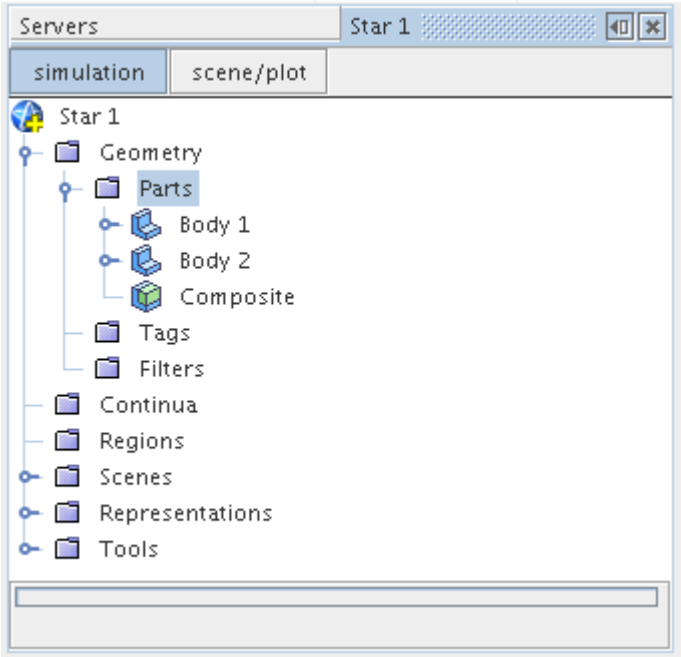
- create a new empty composite part and move existing leaf-level parts or other composite parts into it afterwards
- create a composite part from two or more existing parts

Creating a New Empty Composite Part

To create a new empty composite part (that does not contain any other parts):

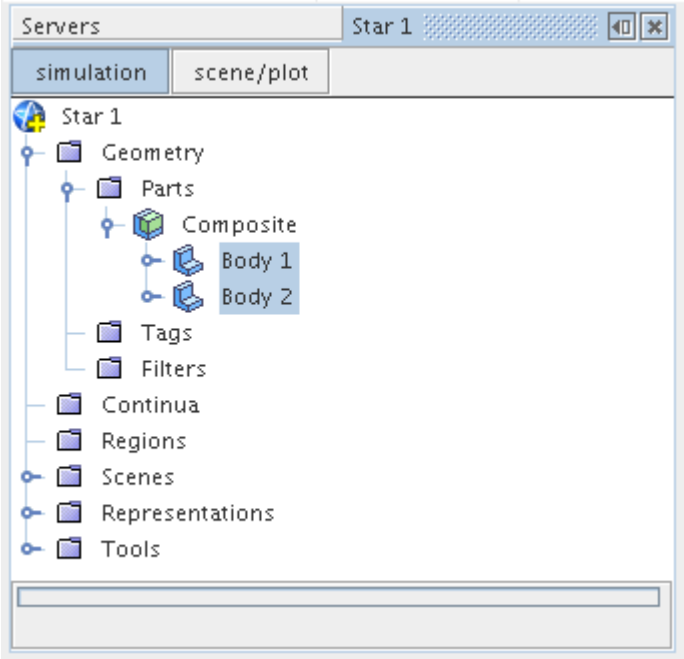
1. Right-click the **Parts** node and select **New Composite Part** from the pop-up menu.

   A new composite part node appears, with the default name **Composite**:



2. Leaf-level or other composite parts can be moved into the new composite part by selecting them and dragging and dropping them into the composite part node.

   This results in the parts becoming children of the new composite part, as shown in the arrangement below:



Any number of new composite parts can be created using this method.

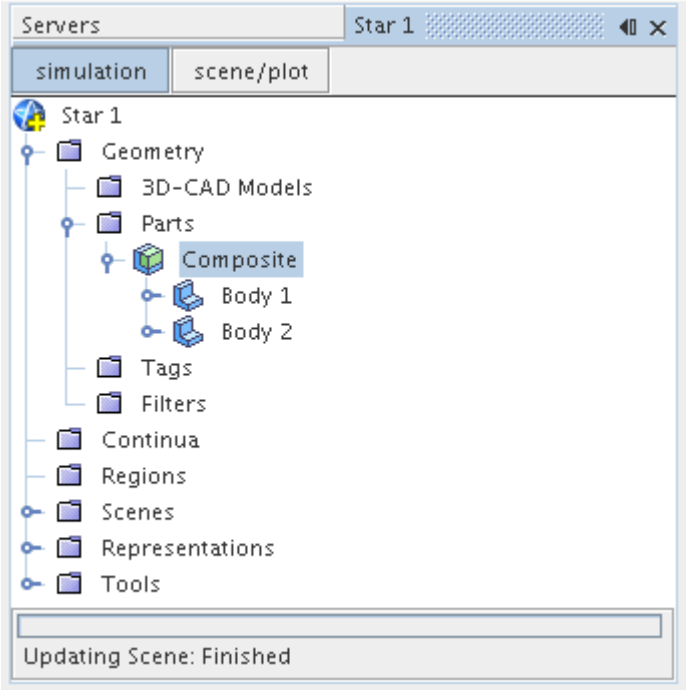
Creating Composite Parts from Other Parts

A new composite part can be created directly from two or more existing parts so that the selected parts automatically become children of the new composite part.

For example, to make two leaf-level parts the children of a new composite part:

3. Select the leaf-level parts, right-click, and select **Composite** from the pop-up menu.

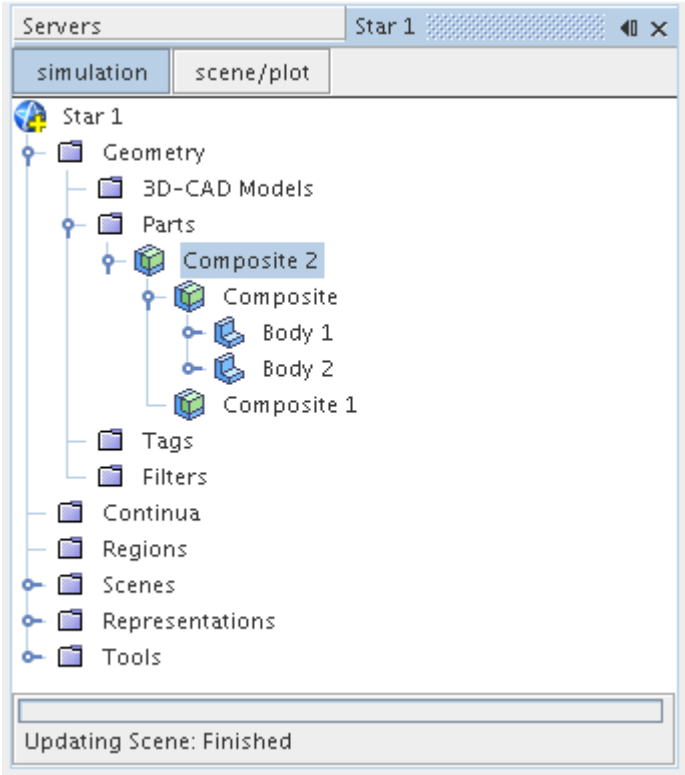
   A new node with the default name **Composite** is created containing the selected leaf-level parts as children:



Similarly, a new composite part can be created from two existing composite parts, such that they automatically become children or subassemblies of the new composite part.

4. As for leaf-level parts, select the two composite part nodes, right-click, and select **Composite** in the pop-up menu.

In this instance, composite parts **Composite** and **Composite 1** are rearranged to become subassemblies of new part **Composite 2**:



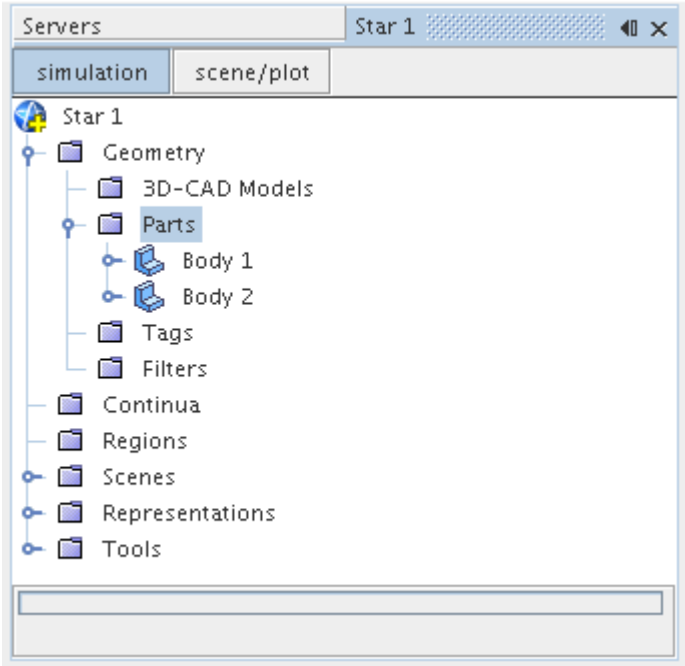
Exploding Composite Parts

A composite part can be exploded such that it is removed from the tree and any child parts of the node are moved up one level.

To explode a composite part:

1. Select the composite node, right-click, and select **Explode** from the pop-up menu.

The selected composite part node is removed and the child nodes are positioned at the next level up:



If the part contains no child nodes, the composite part node is removed.

Using Part Contact Data to Create Interfaces

A geometry part has a *Contacts* property that provides a list of all other parts with which the part is in contact. This part contact data can be used to generate interfaces for meshing and physics coupling automatically without having to create and assign coincident faces manually.

The two key steps in successfully obtaining interfaces are to:

- Generate part contact data using an appropriate method.
- Assign parts to regions in the correct manner.

Using this procedure is useful when you have lots of parts that are in contact with each other and want to produce a conformal mesh at the interface of each part. This procedure requires you to use the surface remesher with the tetrahedral or polyhedral volume meshers.

Generating Part Contact Data

Part contact data can be obtained through the following routes:

- For CAD parts, if the original parts are imprinted on their adjacent neighbors before being imported, select the **Create Part Contacts from Coincident Entities** option on the *Surface Import Dialog*.

  This option causes STAR-CCM+ to detect the coincident faces that are within the specified *Coincident Face Tolerance* automatically and create corresponding part contacts. For 3D-CAD models, this option is used internally when converting them to 3D-CAD parts. See the section, Creating Part Contacts From Coincident Entities for more information.
- The imprint mode of the Surface Preparation Tool can be used to generate coincident part surfaces and associated part contacts.

This facility is available for all part types, including surface mesh parts and inherited part surfaces. An example of the procedure is given below in section, Imprinting Parts on their Neighbors.

- The **Find Part/Part Contact...** menu action can be used to run a proximity analysis that detects adjacent parts.

  This action does not imprint neighboring parts: imprinting happens during the surface and volume meshing process, which can only take place after the parts have been assigned to regions. Further information is given in section, Finding Part Contacts Automatically.
- It is also possible to create part contact data manually, either by using the **Create Part/Part Contact** menu item or by populating node properties directly.

A part contact browser is provided to give easy visualization of the parts that are in contact.

Assigning Parts to Regions

It is crucial that all parts are assigned to new regions in a single operation.

To assign parts to regions:

1. Select all parts in the object tree using `<Ctrl><Click>` or `<Shift><Click>`.
2. Right-click and choose **Assign Parts to Regions...**

   The *Assign Parts to Regions* dialog appears.
3. To identify which options to choose for **Region Mode** and **Boundary Mode** on this dialog, use the information in the table below. The **Create Interfaces from Contacts** option must always be activated.

Table of Valid Methods and Options

The table below summarizes the routes for generating part contact data, and gives the valid options that can then be used when assigning parts to regions using the *Assign Parts to Regions* dialog.

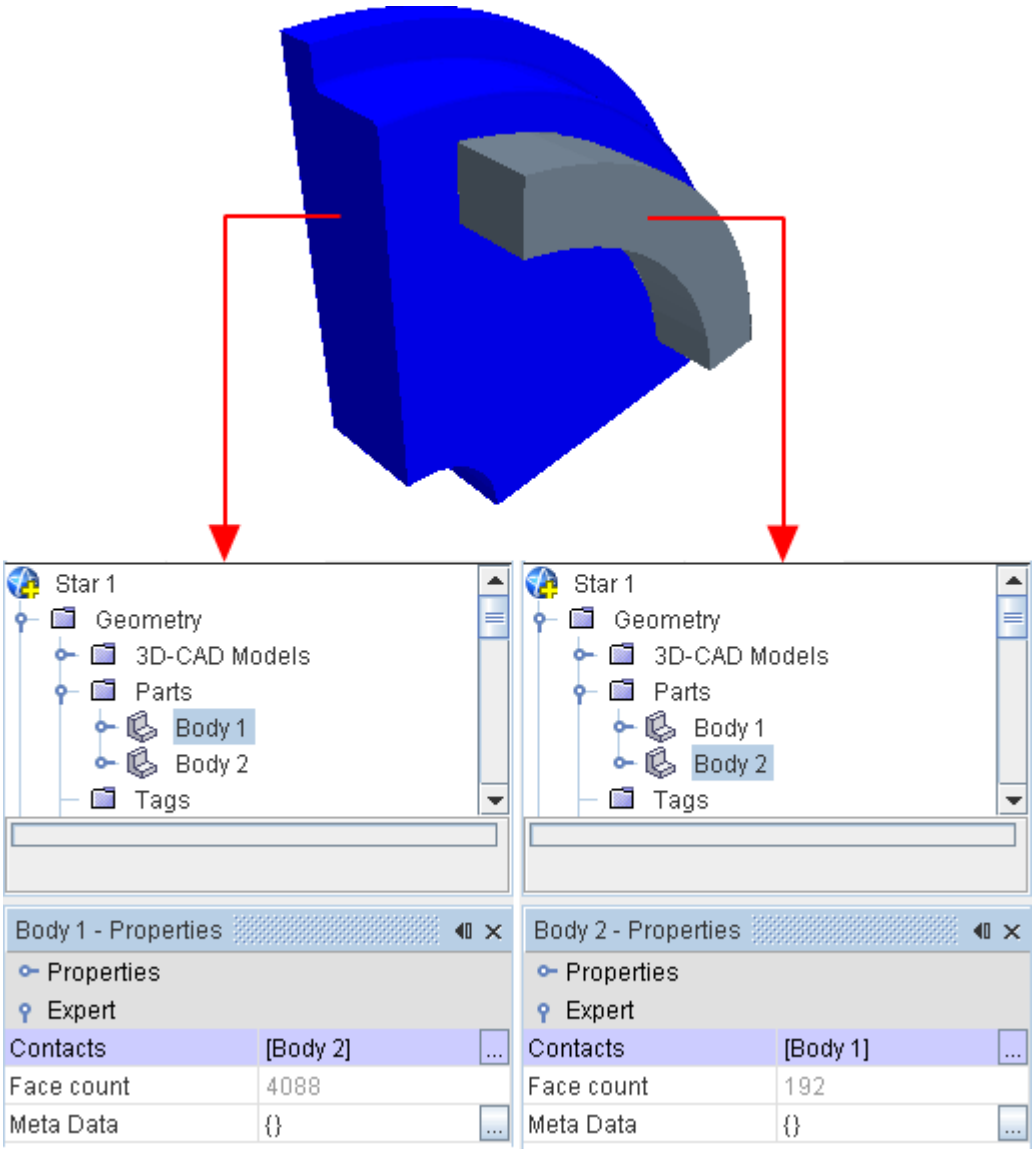
Status of original parts	Relevant Routes for Creating Part Contact Data	Valid Part Types for Route	Valid Options for Region Assignment	
			Region Mode	Boundary Mode
Neighboring parts that are already imprinted	Activate the **Create Part Contacts from Coincident Entities** option when importing. For 3D-CAD parts, this option is always activated internally when converting models to parts	CAD and 3D-CAD only	One region for all parts	One boundary per part surface
			One region per part	One boundary for all part surfaces
			One region per part	One boundary per part surface
Neighboring parts that are not imprinted	Manually imprint parts using the Imprint mode of the Surface Preparation Tool	All part types	One region for all parts	One boundary per part surface
			One region per part	One boundary for all part surfaces

Status of original parts	Relevant Routes for Creating Part Contact Data	Valid Part Types for Route	Valid Options for Region Assignment	
			Region Mode	Boundary Mode
			One region per part	One boundary per part surface
	Select the **Find Part/Part Contacts...** menu action	All part types	One region per part	One boundary for all part surfaces
	Manually define part contact data	All part types	One region per part	One boundary for all part surfaces

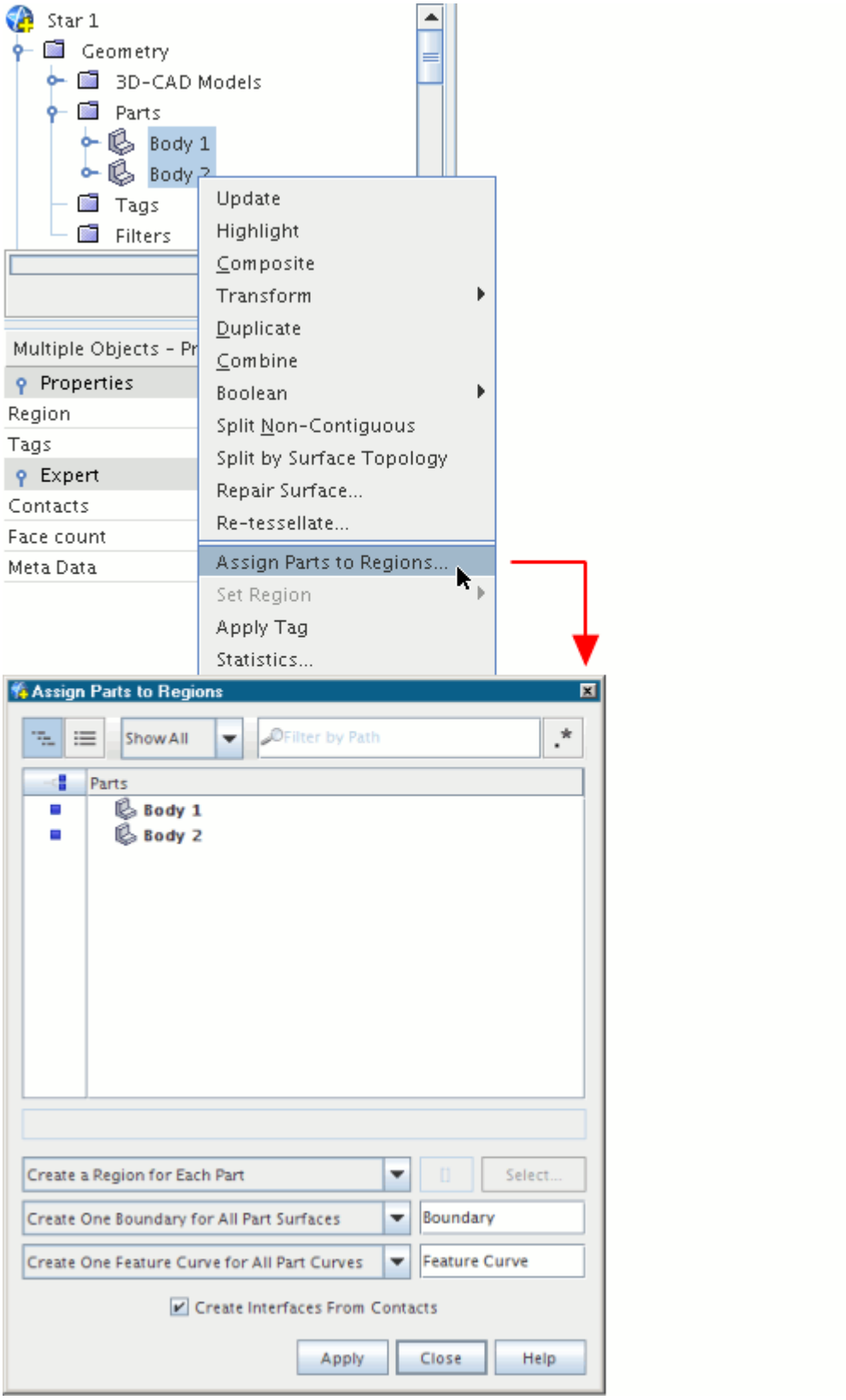
Example Process

The steps below illustrate the process of how interfaces are generated from part contact data.

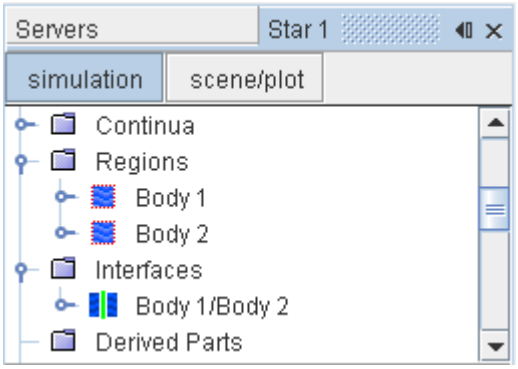
(1) Part contact information is stored on each body



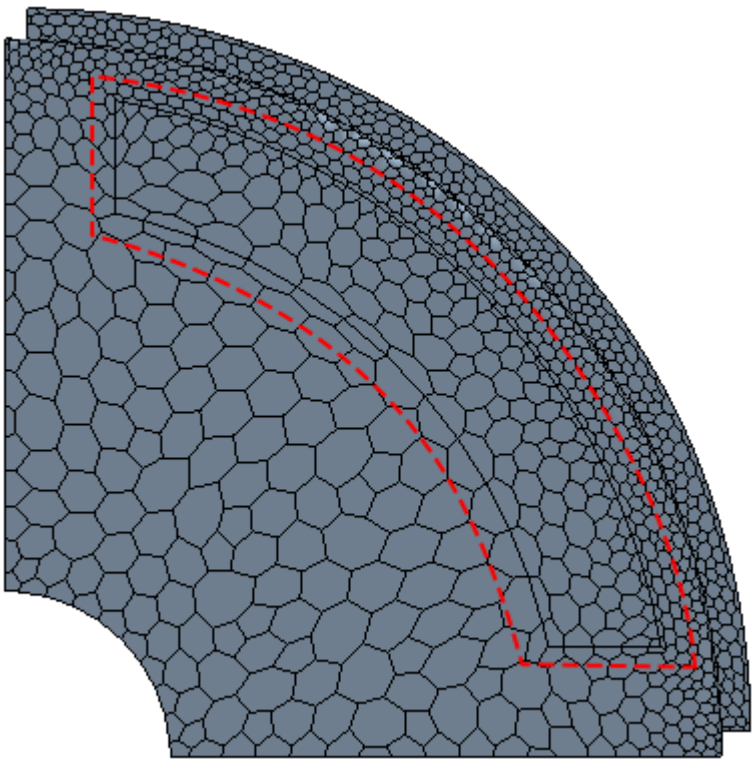
(2) Parts are assigned to new regions (in a single operation)



The result of this operation is the creation of two new regions and an interface for the contact between the regions.



(3) The final mesh contains a conformal interface where regions are in contact



Contents:

Creating Part Contacts from Coincident Entities
Imprinting Parts on Their Neighbors
Finding Part Contacts Automatically
Creating Part Contacts Manually
Browsing Part Contact Data

See Also:

Part Controls

Creating Part Contacts from Coincident Entities

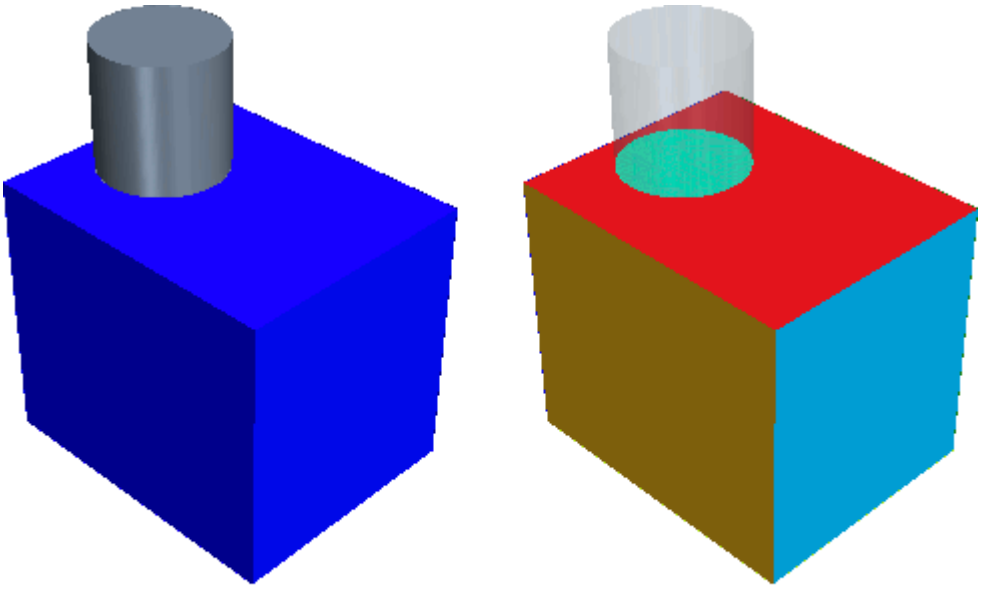
STAR-CCM+ can automatically create part contacts from pairs of coincident faces when importing CAD models, provided the **Create Part Contacts from Coincident Entities** option is selected on the *Import Surface Options* dialog.

This option is exposed when the import mode is set to create geometry parts.

The operation searches all of the imported CAD bodies for pairs of coincident faces that are within the specified *Coincident Face Tolerance*. For each pair found, part contact data is created and stored in the *Contacts* property of each part. A strict definition of coincidence is adhered to, in that the faces belonging to each pair must have an identical surface definition and identical trimming loops. They must overlap exactly without any offset unless they are imprinted.

This same operation is applied when converting 3D-CAD models to geometry parts, and when faces are imprinted using the single or multi-part imprint tool.

The example below shows a 3D-CAD model where the cylinder has been imprinted on the surface of the block. As the imprint operation results in a pair of coincident faces, STAR-CCM+ creates part contact data when the 3D-CAD model is converted to a geometry part.



Geometry Parts: standard view

Split By Patch view on larger block shows the imprinted face

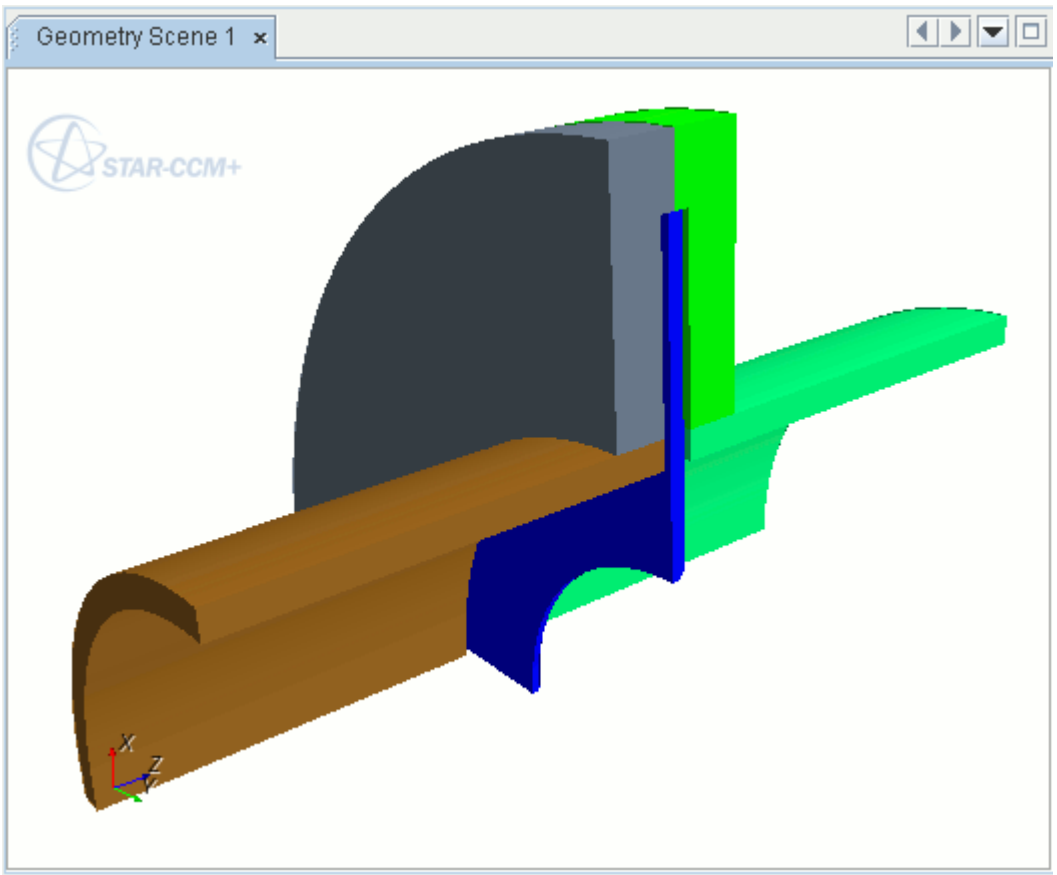
The advantage in using this method to generate the part contact data is that each part surface can be mapped to its own boundary during region assignment. A separate boundary is created for the interface. The **One boundary per part surface** option gives this result.

Imprinting Parts on Their Neighbors

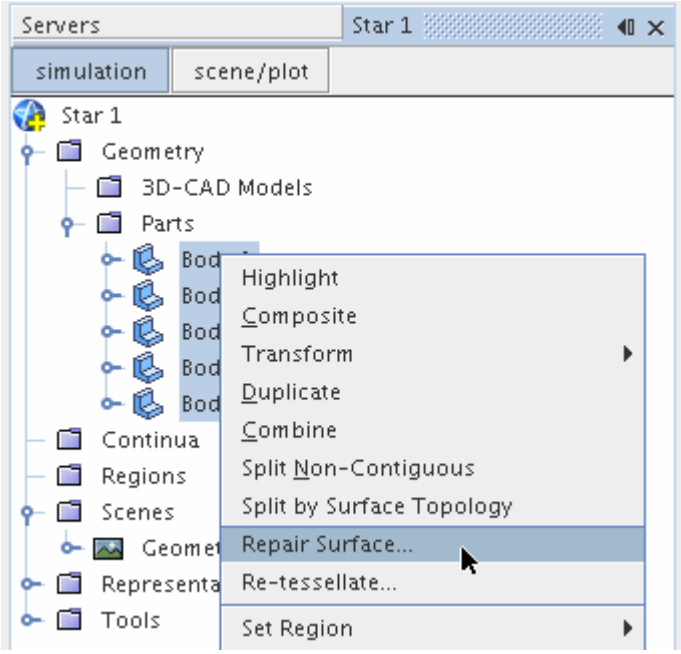
In some cases, the imported data results in multiple geometry parts that have multiple part surface per part, but the parts have not been imprinted on their neighbors. For these cases, it is possible to use the single or multi-part imprint mode of the surface preparation tool to generate separate part surfaces at the point of contact.

Part contact data is created automatically during the imprint process: you can review the contact data using the part contact browser. Following this procedure means that the **One boundary per part surface** option can be used during region assignment.

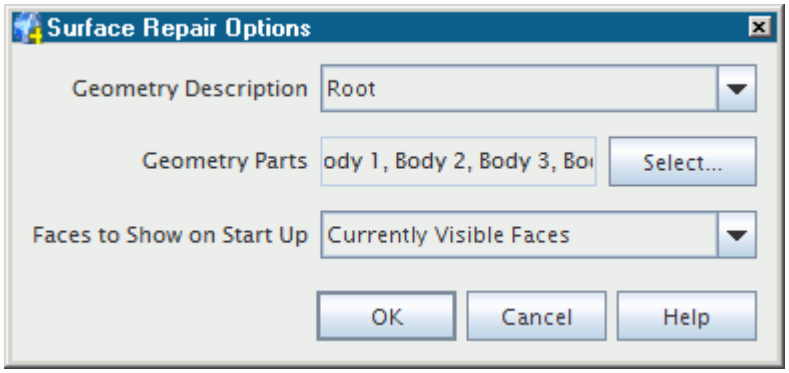
As an example of this procedure, consider the five parts that are shown below which contact at various points. On import, no coincident face pairs were present in the model.



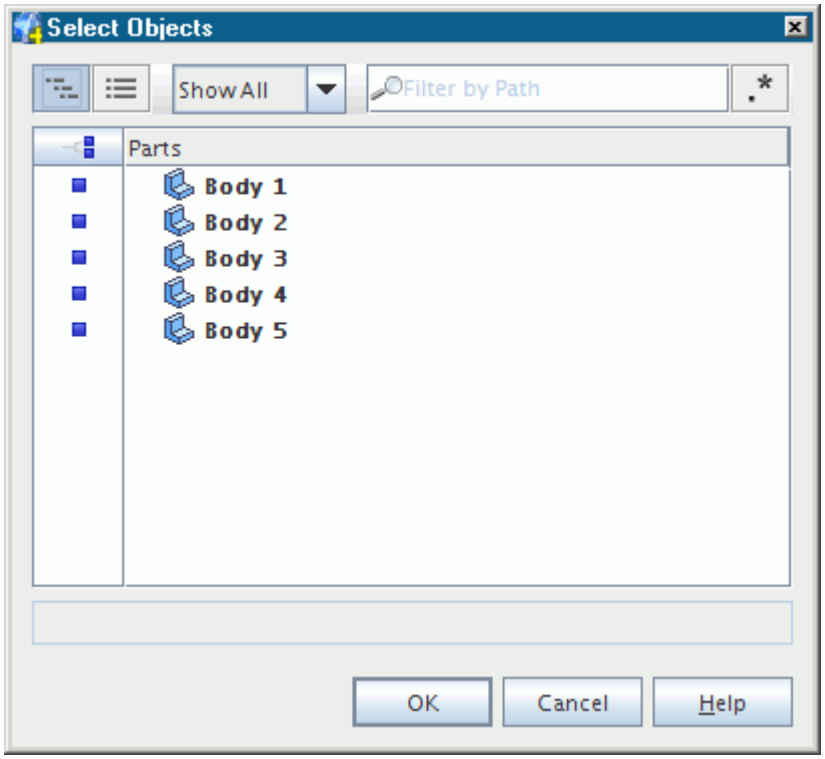
1. It is good practice to check that any surface repairs are complete before imprinting.
2. All parts are selected, and the surface preparation tool is launched using the right-click menu option, **Repair Surface...**



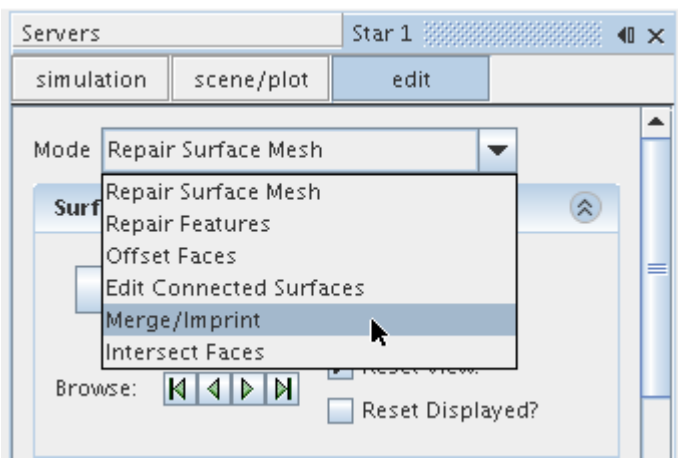
3. In the *Surface Repair Options* dialog, to include all **Geometry Parts** that are selected, click **OK**.
4. To change the selection of parts, click **Select**.



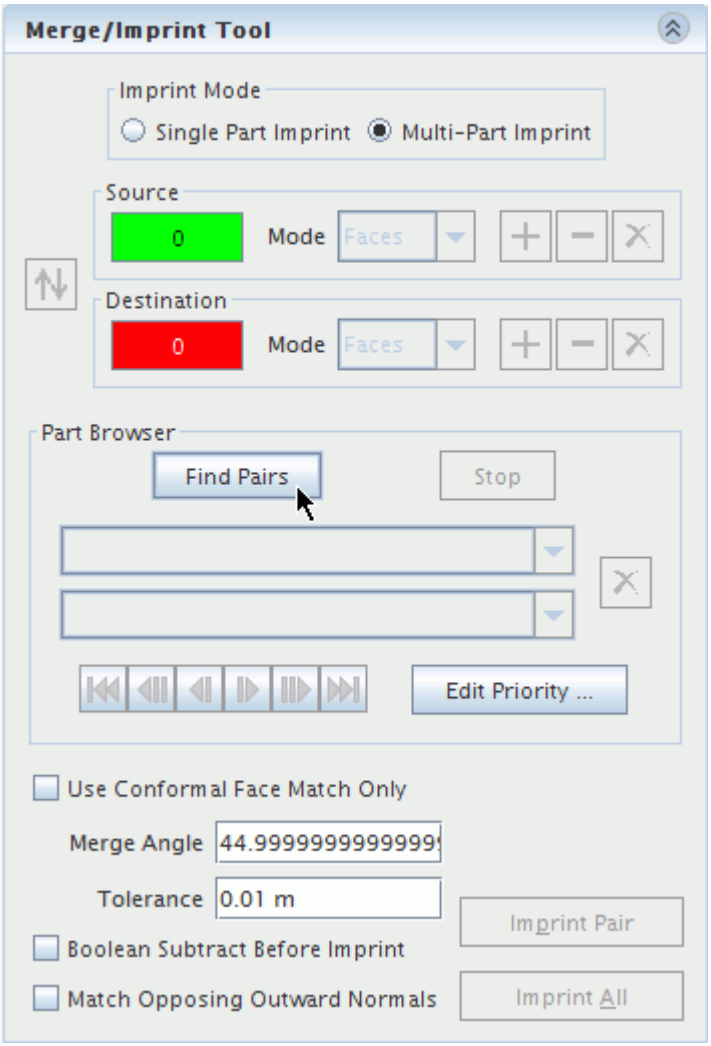
5. In the *Select Objects* dialog, click the button to the left of each part to select or deselect that part, then click **OK**.



6. In the *edit* panel that appears for the surface preparation tool, the **Mode** must be changed to **Merge/ Imprint**.



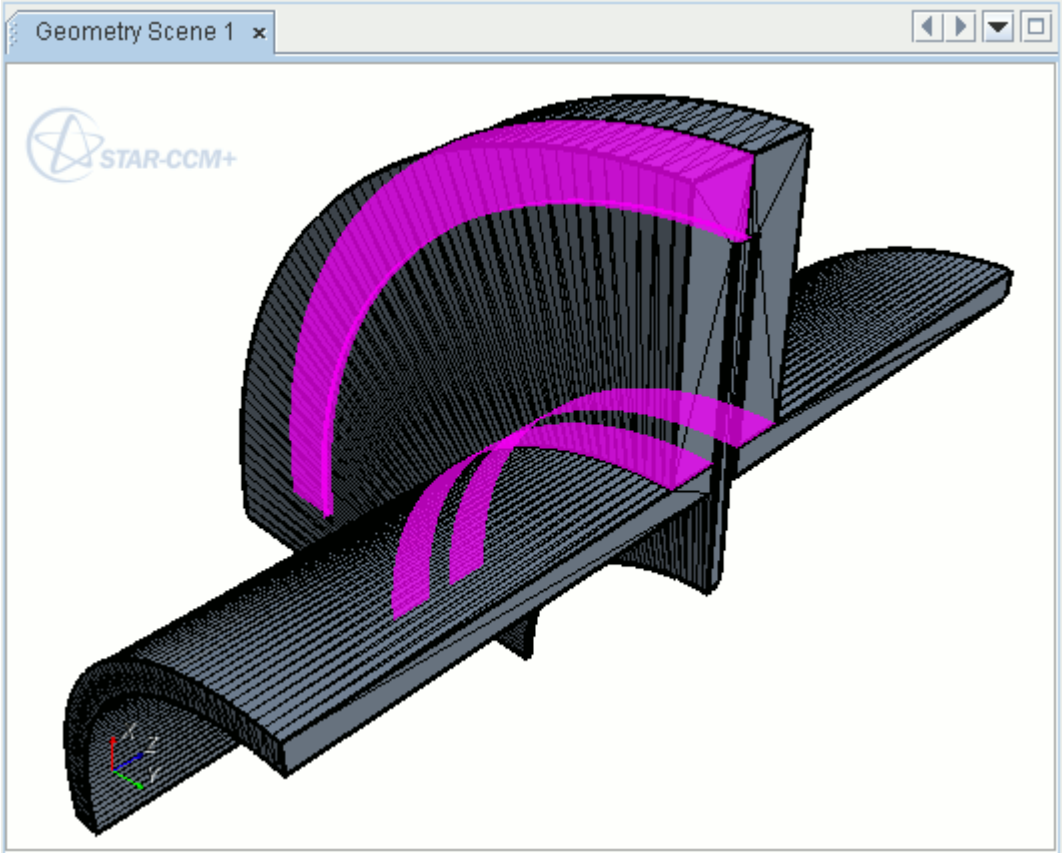
7. As multiple parts are present, the *Imprint Mode* is set to **Multi-Part Imprint**. In this mode, an automatic option is provided to find pairs of contacting parts (within the *Part Browser* group box).
8. The **Tolerance** value must be set to an appropriate value for the model, that is, a value less than the minimum expected gap thickness between non-contacting parts.
   For this example, the value is reduced to `0.0001 m`.
9. The **Find Pairs** button is clicked to execute the automatic search.



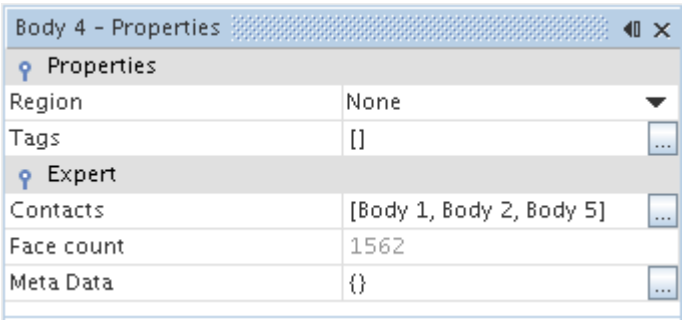
10. The part pairs that are found can be reviewed using the **Browse Pairs** navigation buttons, and if necessary, the button can be used to delete the pair.
11. Assuming all pairs have been correctly identified, the **Imprint All** button is pressed to begin the imprint operation.

    Information on the number of new interfaces that are created is shown in the *Output* window, and a final update is made to the *Graphics* window showing the newly created interfaces.

    For this example, the final view in the *Graphics* window after the imprint operation shows the creation of five new interfaces.

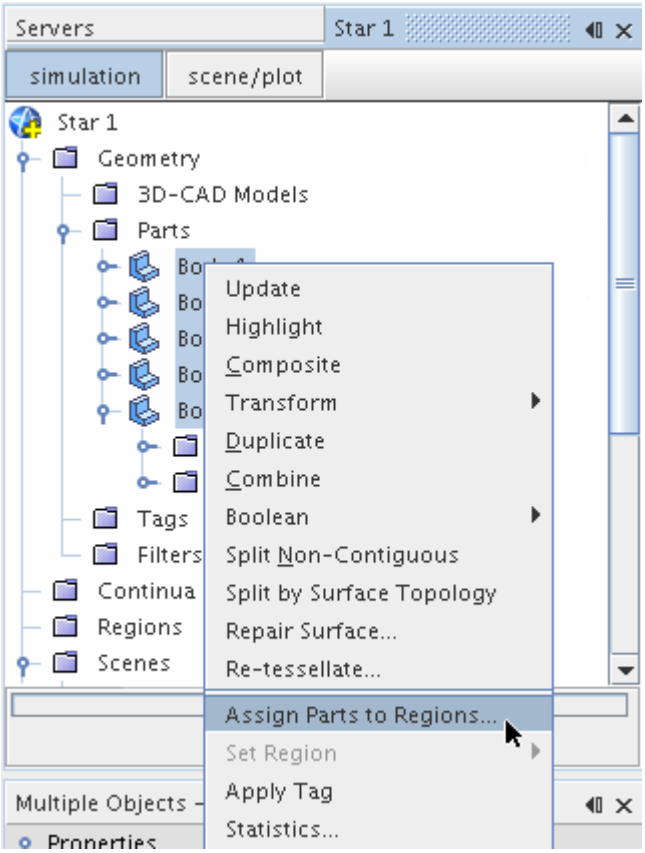


12. When you click **Close** to exit the Surface Preparation Tool, the *Contacts* property of each body is populated with the results of the imprint operation. **Body 4**, for example, is in contact with three other bodies.

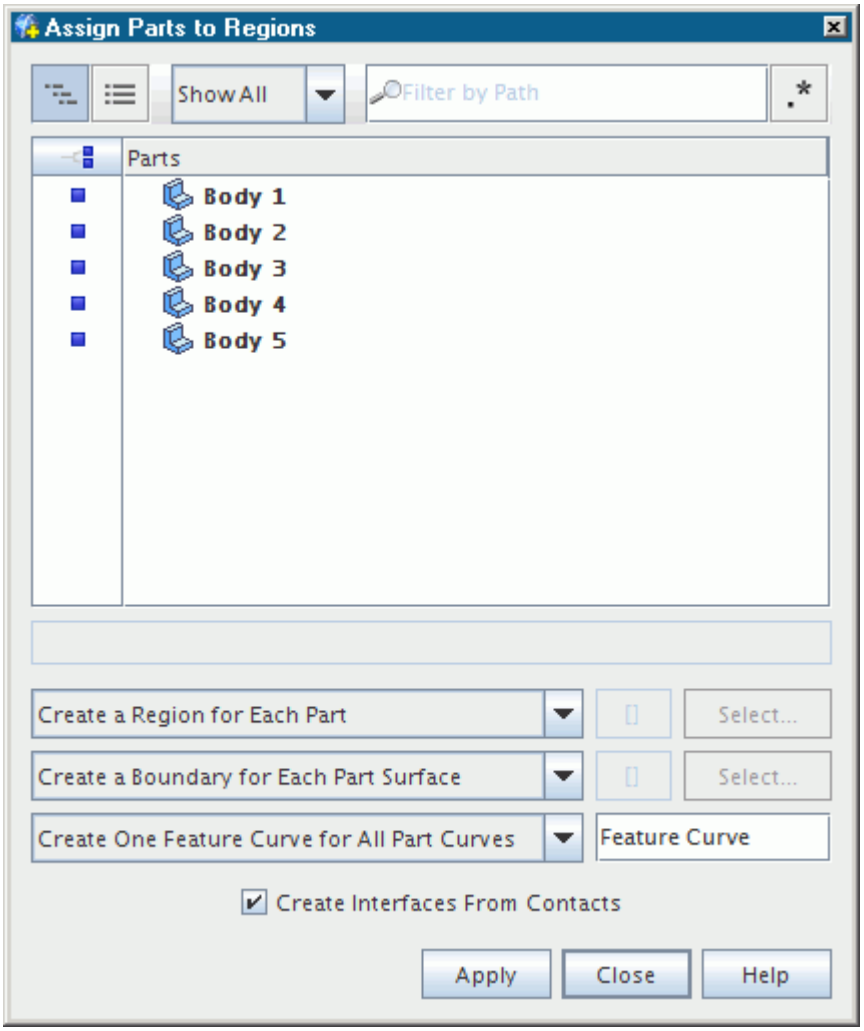


The final stage is to assign all the bodies to regions in a single operation.

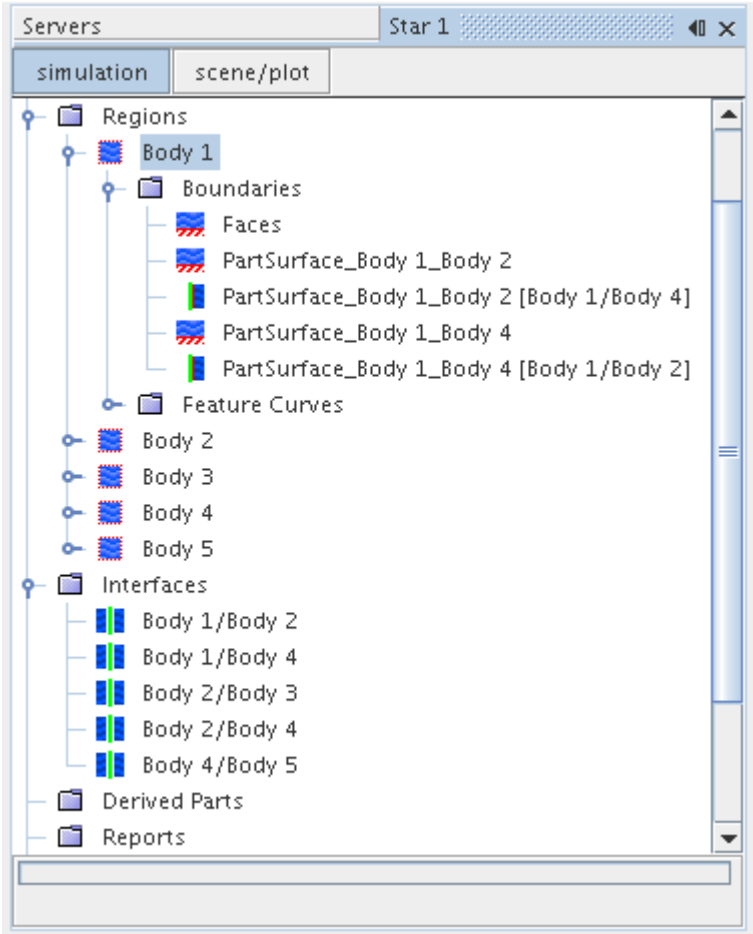
13. All parts are selected, and the **Assign Parts to Regions** option is chosen from the right-click menu.



14. In the *Assign Parts to Regions* dialog that appears, the following options are selected:
    - **Create a Region for Each Part**
    - **Create a Boundary for Each Part Surface**
    - The **Create Interfaces from Contacts** option is activated.



On clicking **Apply**, parts are assigned to new regions and faces to new boundaries. Interfaces are created from their own boundaries and shown in the **Interfaces** manager node.



You can see that separate boundaries have been created for the portions of the original surface that were separated out during the imprint process. Any other separate part surfaces in the original imported surface would also have received their own boundary.

Finding Part Contacts Automatically

Part contacts can be found automatically between two or more parts that are based on a user supplied tolerance value. This feature is useful in circumstances where several parts exist, some of which can be touching two or more other parts.

To find part contacts automatically:

1. Right-click the **Geometry** > **Contacts** node and select **Find Part/Part Contacts**.
2. In the *Part/Part Contact Search Options* dialog, select two or more parts on which to search for contacts.
3. You can specify the **Tolerance** value in any allowable units.
4. Click **OK**. Part contacts are generated for each part combination where the part surfaces contact each other within the specified tolerance value.

   The *Output* window indicates the total number of part contacts that are produced. The **Contacts** node is populated with a node for each part contact that is created and the *Contact* property for each [Part] is updated.

The part contacts list can be edited if necessary.

Creating Part Contacts Manually

This section describes how to create part contact data manually.

There are two ways to create part contact data manually:

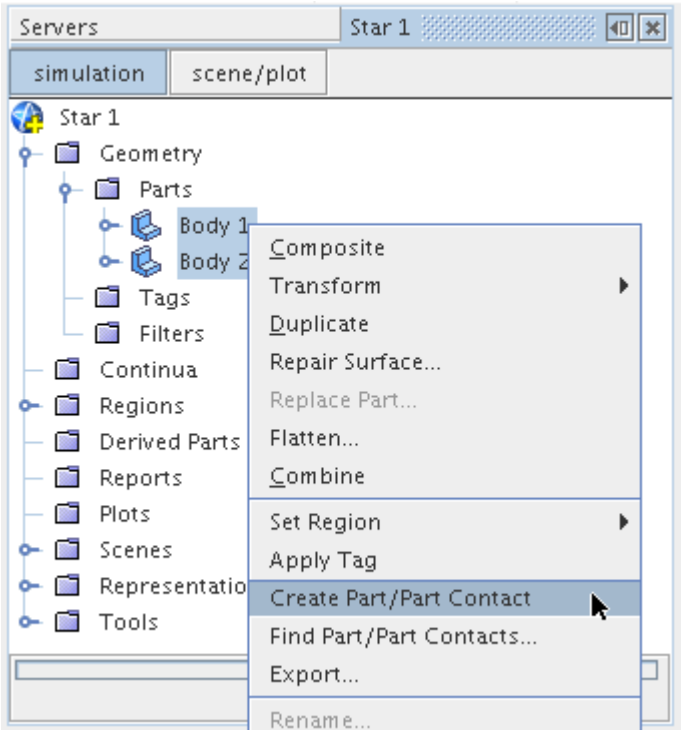
- Using the action **Create Part/Part Contact** on the right-click menu for parts.
- Editing the *Contacts* property directly.

Using the Menu Action

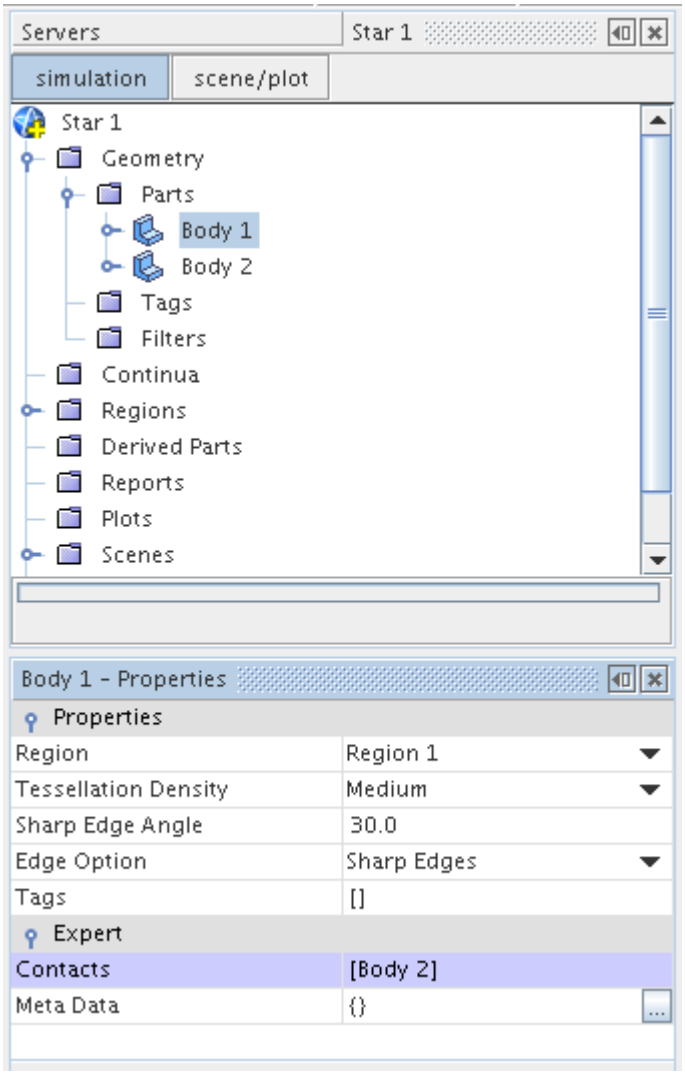
Part contacts can be created manually between two or more parts. This operation is useful when the parts have surfaces that are in contact with one another and require an interface for the physical coupling once the mesh has been built.

To create the part contact manually:

1. Select the part nodes of interest, right-click, and select **Create Part/Part Contact** from the pop-up menu:



The part contact is created and updated in the *Properties* window for each part. For example:



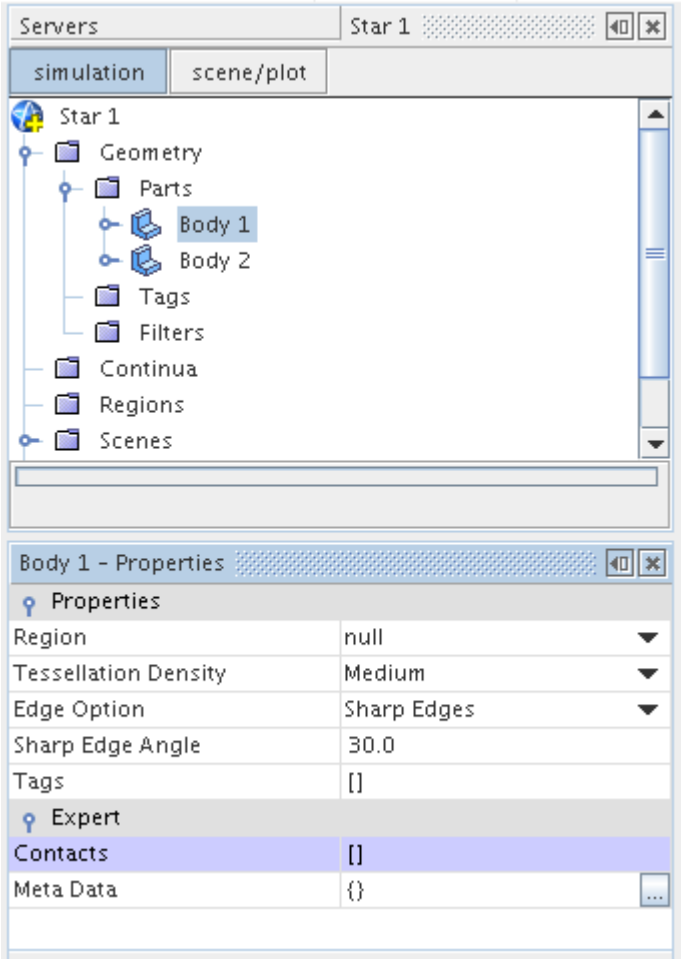
In the above example, the **Body 2** part is added as a contact for the **Body 1** part. Selecting the **Body 2** part node would show **Body 1** as a contact. The part contacts list can be edited if necessary.

This method does not check whether the parts are really in contact with one another. Each part is added to the *Contacts* list of all the other parts that are initially selected.

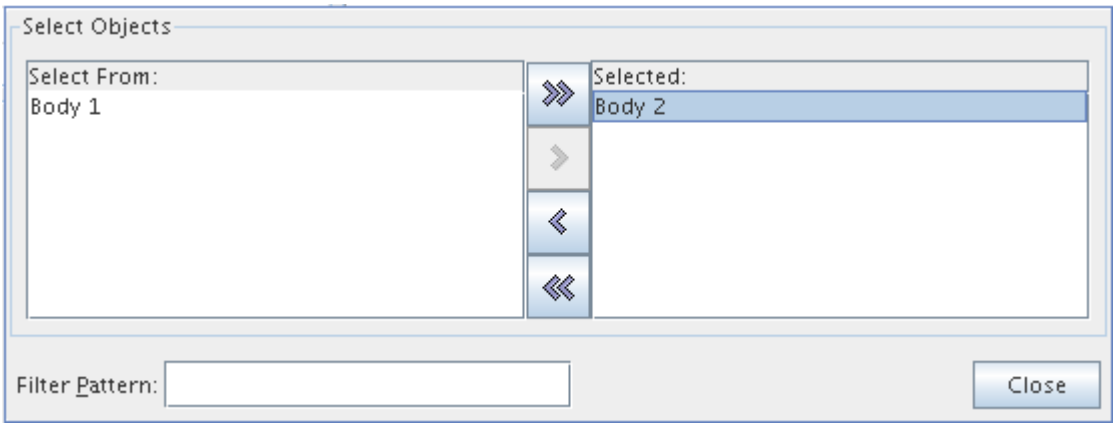
Editing the Part Contacts List

The *Contacts* property for each part node can be used to either define new part contacts or edit existing ones for the part.

2. Selecting the part node shows the current contacts list:

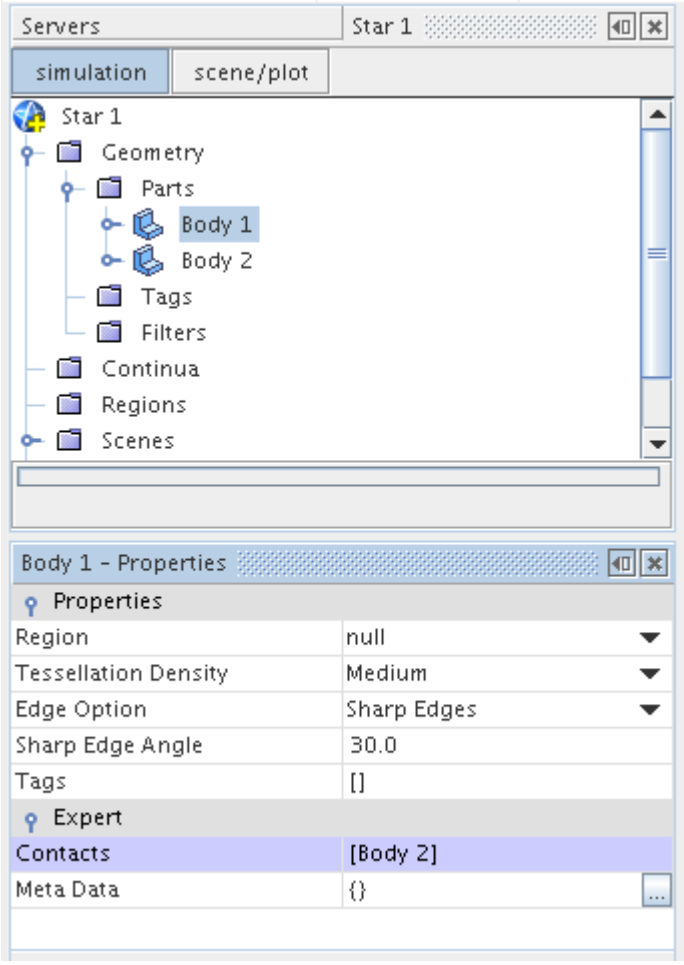


3. Clicking the *Contacts* property opens the *Select Objects* dialog:



4. The parts that are in contact with the originating part node can be selected from the list on the left-hand side and the **Close** button pressed.

   The *Contacts* list for the part node is updated accordingly:



Always define part contacts in pairs such that each part has a reciprocal member. If the part is removed from the list, also edit the reciprocal part to remove the partner node.

Browsing Part Contact Data

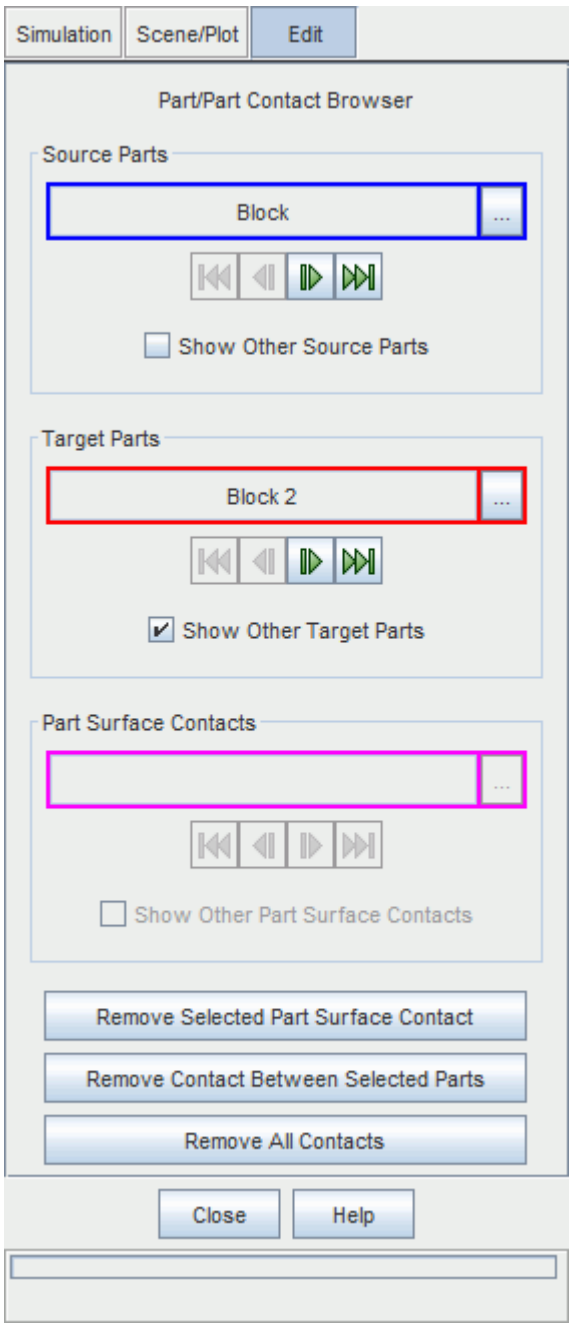
The *Part/Part Contact Browser* allows you to scan through the part contact data and see the contacting parts in the *Graphics* window.

This operation automatically updates the view to hide parts that are not part of a contact list.

To start the browser:

1. Right-click the **Geometry** > **Contacts** node and select **Browse Part/Part Contacts...**
2. In the *Part/Part Contact Browser Options* dialog that appears:
   a) Click **Select**.
   b) In the *Select Objects* dialog, select all of the parts for which you want to visualize part contact data and click **OK**.
   c) Click **OK**.

   The **Edit** panel appears showing the interface for the contacts browser.



There are three group boxes in the Part/Part Contact Browser. In each group box, you can click ... (**Custom Editor**) and select a specific part or part surface. You can also use the navigation buttons to step through the list in sequence or jump to the start or end of the list.

- *Source Parts*

  The drop-down menu lists all parts that are selected in the *Part/Part Contact Browser Options* dialog. In the *Graphics* window, the selected source part is colored blue and when *Show Other Source Parts* is activated, other source parts are displayed as transparent gray objects.

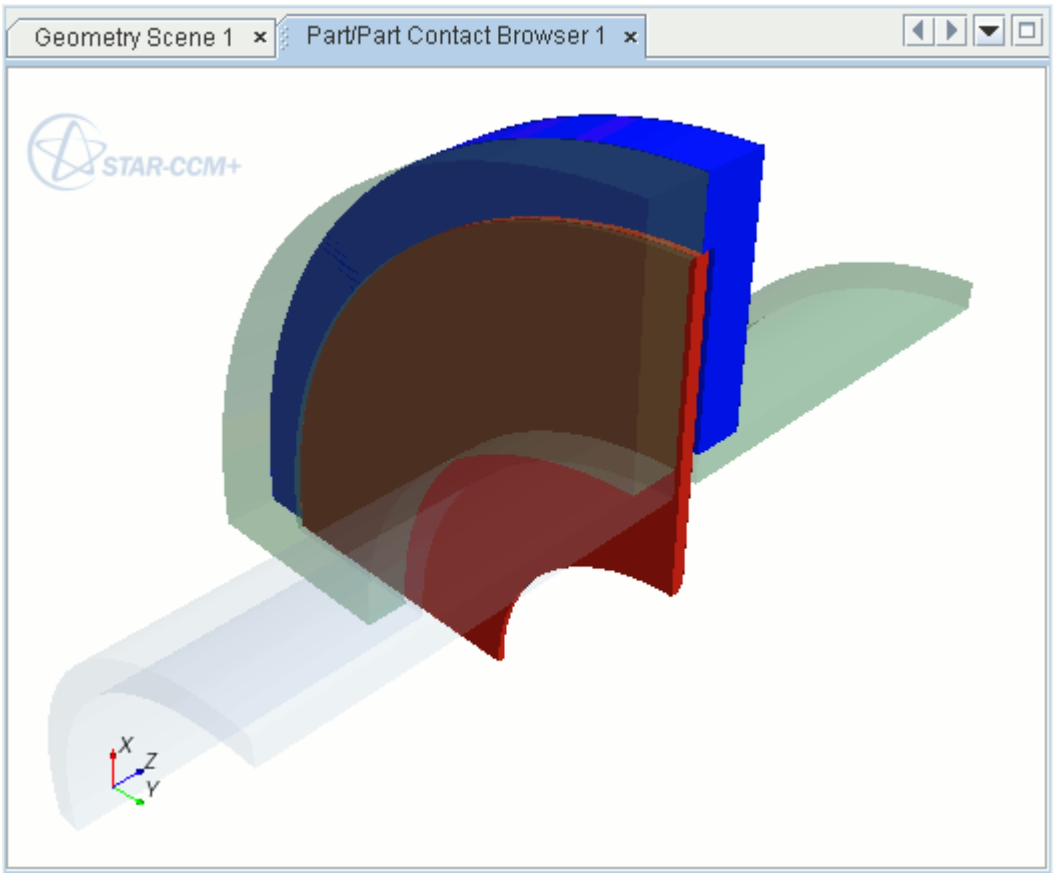
- *Target Parts*

The drop-down menu lists all parts that are in contact with the source part that is shown in the *Source Parts* group box. In the *Graphics* window, the selected target part is colored red and when *Show Other Target Parts* is activated, other target parts are displayed as transparent green objects.

- *Part Surface Contacts*

  The drop-down menu lists all part surface contacts that exist between the selected source part and target part. In the *Graphics* window, the selected part surface contact is colored magenta and when *Show Other Part Surface Contacts* is activated, other part surface contacts are displayed as transparent pink (duller than magenta).

In the example scene below, the blue part is the source part, and the red part is the target part. The transparent green parts are alternative target parts which are in contact with the source part, and the transparent gray part is an alternative source part that is available to select.



The Part/Part Contact Browser also contains buttons which allow you to remove any or all of the contacts between parts or part surfaces.

- **Remove Selected Part Surface Contact**

  Removes the part surface contact that is selected.

- **Remove Contact Between Selected Parts**

  Removes the contact that is displayed between the selected source and target parts.

- **Remove All Contacts**

Removes the contacts between all parts and part surfaces.

3. To exit the *Part/Part Contact Browser*, click **Close**.

Updating Part Interfaces

Updating part interfaces between two regions becomes necessary when one or more interfacing parts have been modified or have changed region.

In these cases, the original interface that existed between the two regions is now obsolete. If a part (or part surface) has been modified, perform an imprint before updating the interfaces.

There are two methods you can use to update part interfaces: using the **Parts** manager node, or right-clicking the desired part. In both cases, STAR-CCM+ checks part contacts, ensuring that appropriate interfaces exist where needed, and that interfaces that are no longer valid are removed.

To update interfaces:

1. Right-click the **Parts** manager node, or the desired part, and select the appropriate option:
   - For boundary mode interfaces, select **Update Interfaces** > **Update Boundary-mode Interfaces**.
   - For contact mode interfaces, select **Update Interfaces** > **Update Contact-mode Interfaces**.

Updating interfaces using the **Parts** manager node ensures that all existing parts have up-to-date interfaces. This method is the recommended way to ensure that all interfaces in the case are up-to-date. Updating interfaces by right-clicking individual parts checks and updates only the interfaces affecting the selected parts.

Manipulating Parts

STAR-CCM+ allows leaf-level parts to be manipulated in several different ways once they have been imported or created.

Note: The manipulation options available are dependent on the part type. For example, the splitting non-contiguous parts options are only available for mesh parts.

For a reference page that contains a short description of each right-click part operation, visit the Part Operations Reference Page.

Contents:

Retessellating a Part
Transforming a Part
Duplicating a Part
Combining Parts
Performing Boolean Operations
Splitting Non-Contiguous Parts
Splitting a Part by Surface Topology
Replacing a Part
Manipulating Part Surfaces
Deleting Empty Entities
Manipulating Part Curves
Repairing Part Surfaces and Curves
Exporting Parts
Getting the Face Count of a Part
Editing Imported Parts in 3D CAD
Part Operations Reference Page

See Also:

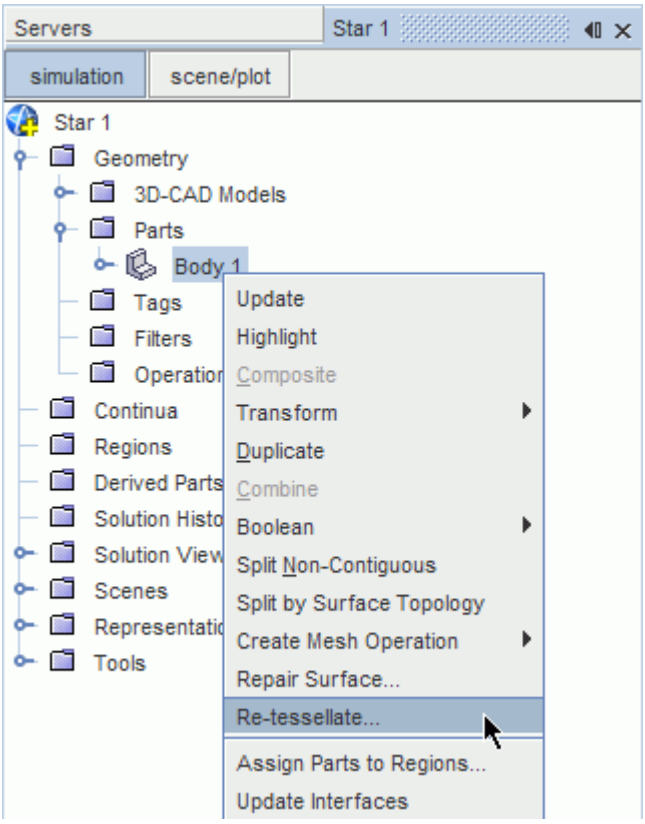
Understanding Geometry Parts

Retessellating a Part

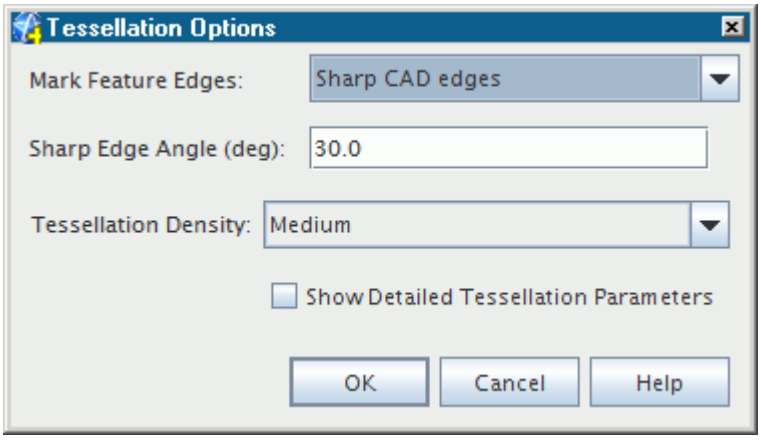
The surfaces of a geometry part are represented using triangles which come from a process that is known as tessellation.

For 3D-CAD, imported CAD, and shape part types:

1. The triangle density that is used to tessellate a surface can be modified using the **Re-tessellate...** option on the right-click menu for the part.



Selecting this option opens the *Tessellation options* dialog. In addition to providing control over the tessellation density, the dialog also provides control over the marking of CAD edges as features edges. These options behave in an identical manner to the options that are provided on the *Surface Import Options* dialog. See the sections on Tessellation Density and Mark Feature Edges for details.



It is possible to see the triangles that are used to represent the surfaces by choosing to display the mesh in a geometry scene for which the Geometry representation is active.

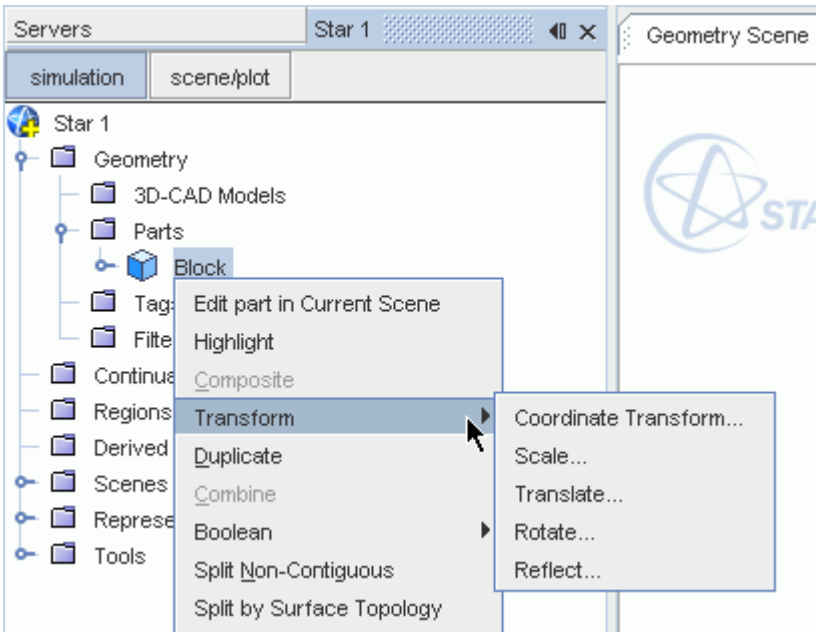
Transforming a Part

Five different types of transformations are provided to transform parts.

The five different transformations that are allowed are:

- Coordinate Transform
- Scaling
- Translation
- Rotation
- Reflection

These options appear when right-clicking a part node and selecting **Transform** from the pop-up menu.



The transform options are useful in situations where parts are not aligned correctly or have been saved at a different scale to other parts. Transform operations can be performed on one or more parts simultaneously. When transforming a composite part, the selected operation affects all leaf-level parts that are contained within the composite.

For 3D-CAD parts, transforms can be performed on the original 3D-CAD bodies using the 3D-CAD solid modeler and then updated accordingly. Any transforms made to 3D-CAD geometry parts must be repeated if their underlying 3D-CAD model is modified.

Contents:

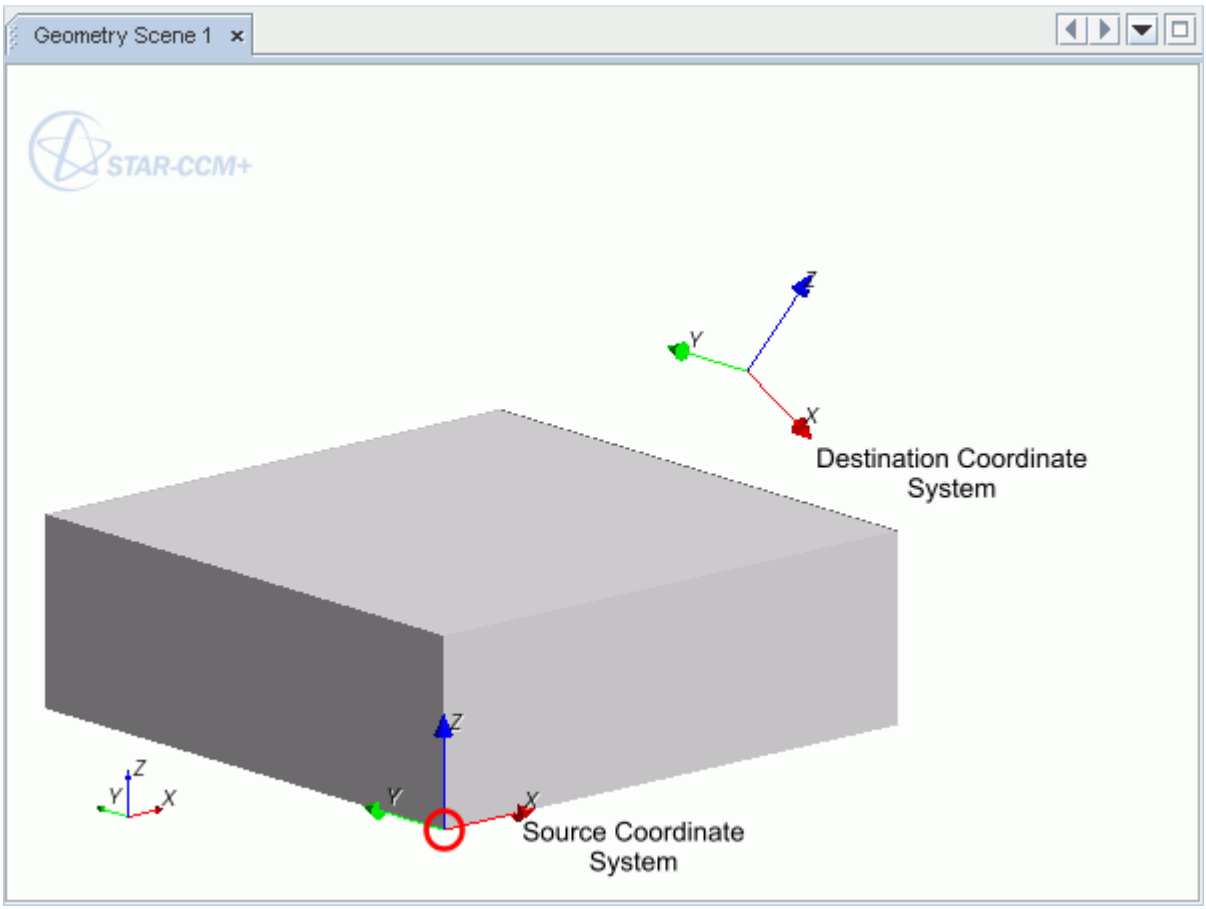
Transforming Parts to a Coordinate System
Scaling a Part
Translating a Part
Rotating a Part
Reflecting a Part

Transforming Parts to a Coordinate System

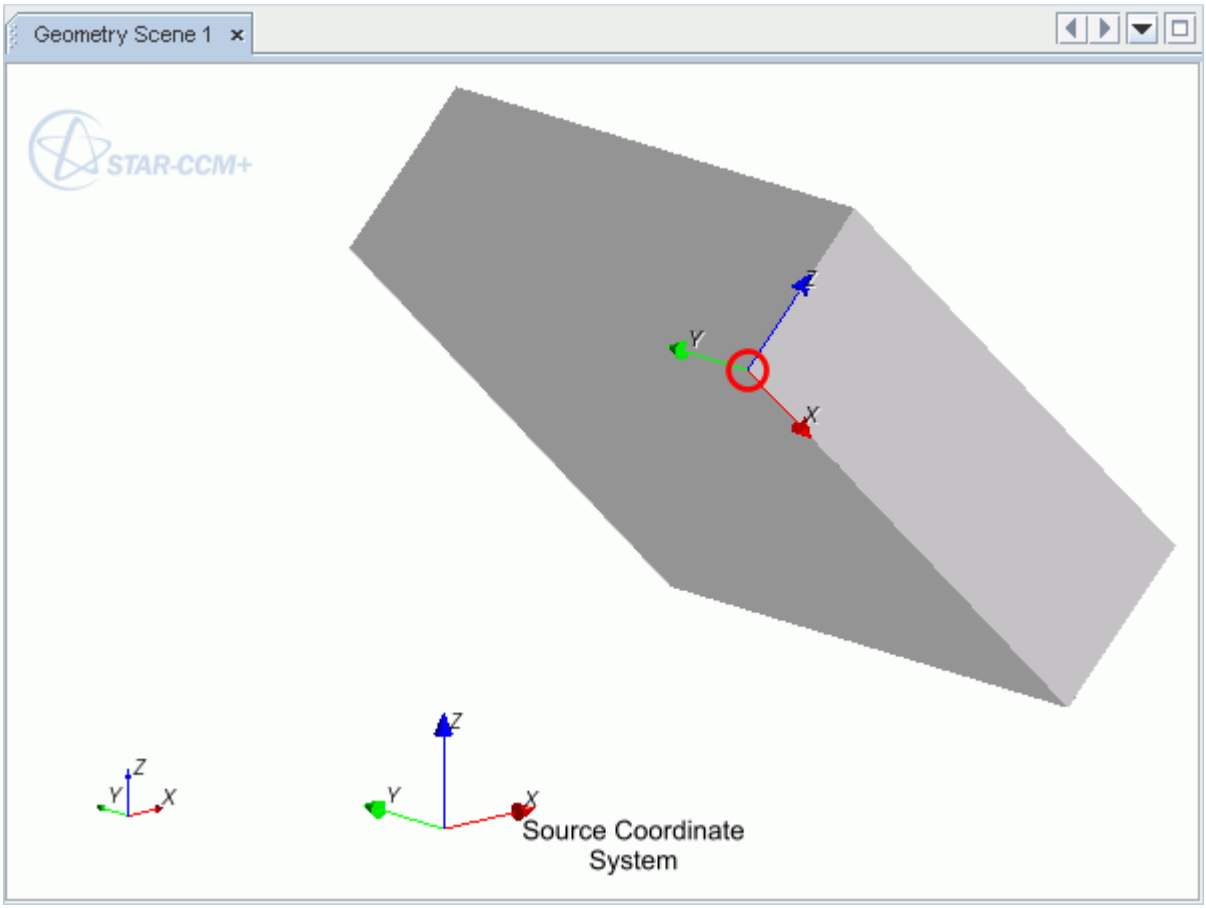
Transforming a part to a coordinate system moves a part from a position relative to a source coordinate system, to a position relative to a destination coordinate system.

This transform is useful for both translating and rotating a part in the same operation, and maintains the exact shape of the part during the transformation.

For example, the highlighted corner in the following image is at the origin of the source coordinate system.



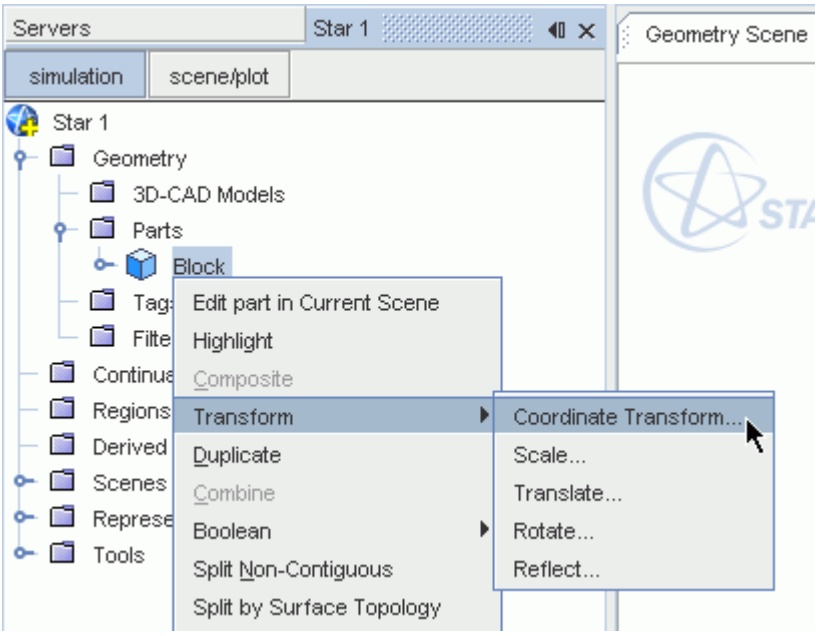
Transforming to the destination coordinate system has the following effect.



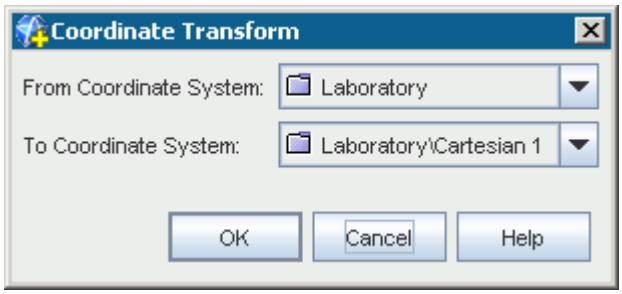
The highlighted corner is now at the origin of the destination coordinate system, and the shape of the part is maintained.

To transform a part to a coordinate system:

1. Right-click the part node and select **Transform** > **Coordinate Transform...**



The *Coordinate Transform* dialog opens.



2. Select the source coordinate system from the **From Coordinate System** menu and the destination coordinate system from the **To Coordinate System** menu.
3. Click **OK** to perform the transform.

If you select a spherical or cylindrical coordinate system, the transform uses a Cartesian coordinate system with the same origin as the selected system. This correspondence prevents distortion of the part when transforming to a non-Cartesian coordinate system.

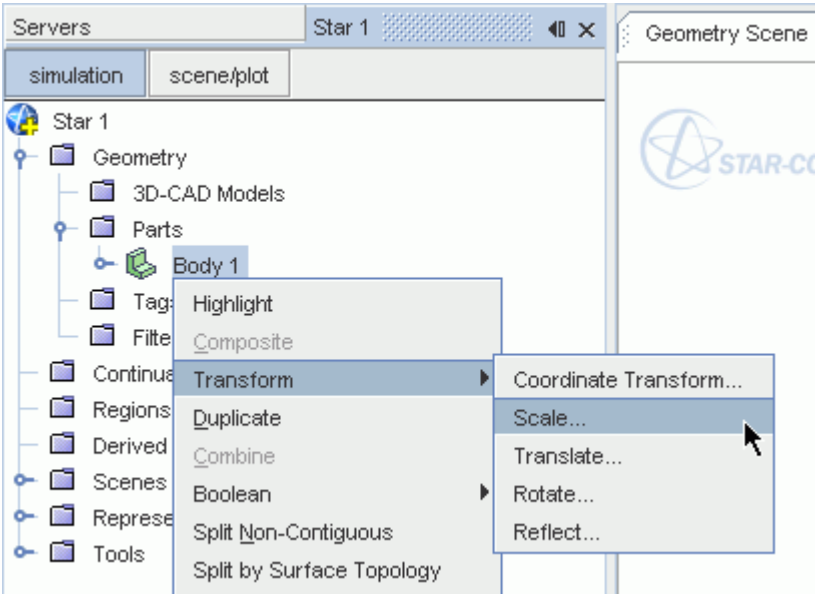
Scaling a Part

You can scale a part if it uses units that are different to other parts you have already imported.

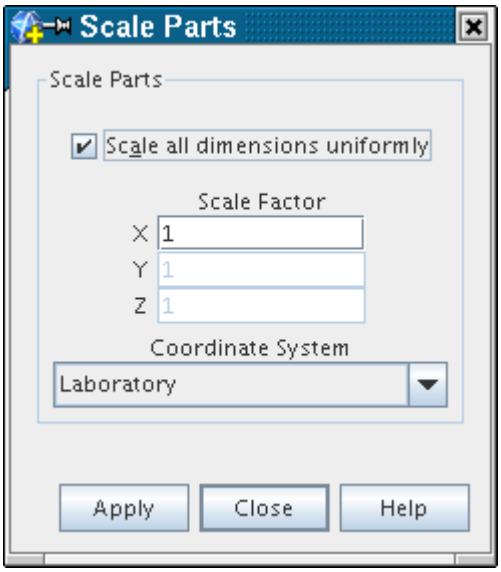
Scaling can take place in any or all of the three coordinate directions and can be used to increase or decrease the size of a part.

To scale a part:

1. Right-click the part node and select **Transform** > **Scale...** from the pop-up menu:



The *Scale Parts* dialog appears:



By default, the **Scale all dimensions uniformly** checkbox is ticked, which means that all dimensions are scaled equally. In this mode, the scale factor can be entered into the **X** text box.

2. To scale each coordinate direction independently, deselect the **Scale all dimensions uniformly** checkbox.
3. The coordinate system in which the scaling takes place can be selected from the **Coordinate System** menu.
4. Click **Apply** to carry out the operation, which is shown on the display if a valid scene is open.
5. Additional scaling operations on the part can be performed as described above and clicking **Close** when complete.

If a valid scene is open, the display is updated accordingly.

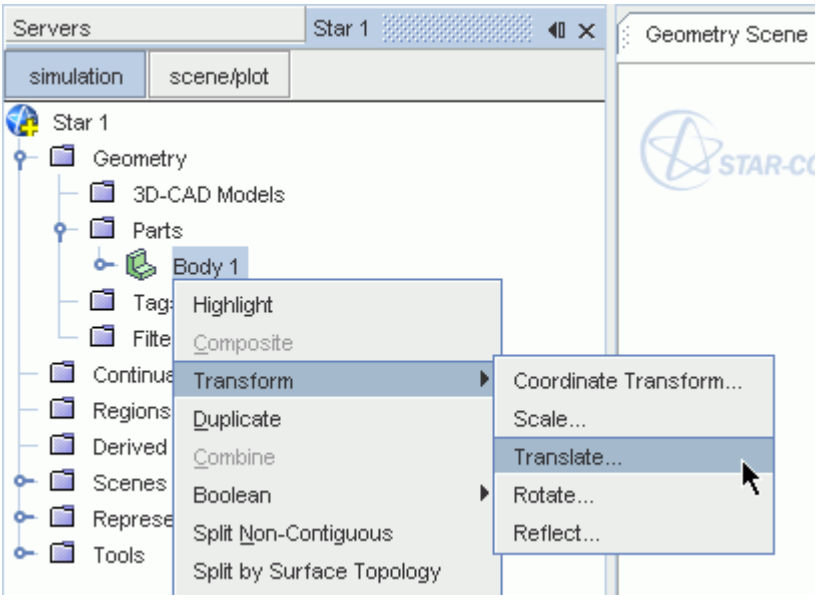
Translating a Part

Translating a part can be used to rearrange the geometry so that it fits together as intended.

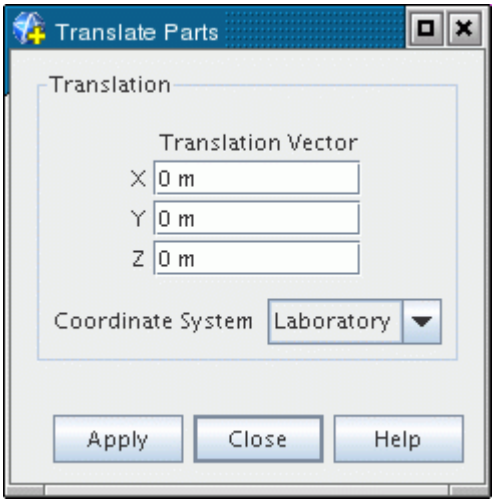
This problem can arise when different parts are built in different coordinate systems. Translation can take place in any of the three coordinate directions.

To translate a part:

1. Right-click the part node and select **Transform** > **Translate...** from the pop-up menu:



The *Translate Parts* dialog appears:



2. The *Translation Vector* component values can be entered using either positive or negative values, along with the desired units.
3. The choice of **Coordinate System** is also allowed.
4. Click **Apply** to carry out the operation, which is shown on the display if a valid scene is open.

The operation can be repeated as many times as desired.

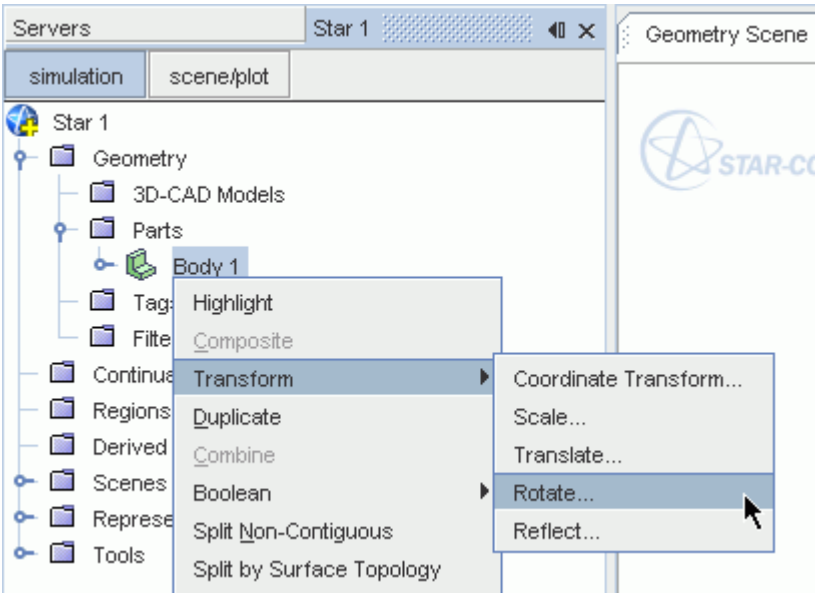
Rotating a Part

You can rotate a part to reorient it with other static components.

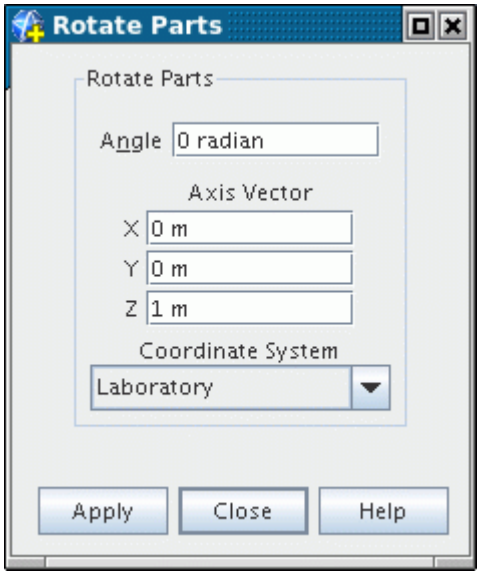
Rotation is allowed in any axis direction.

To rotate a part:

1. Right-click the part node and select **Transform** > **Rotate...** from the pop-up menu:



The *Rotate Parts* dialog appears:



2. The **Angle** of rotation can be entered (in radians by default) along with the *Axis Vector* for the rotation.
3. The choice of **Coordinate System** can also be made.
4. Click **Apply** to carry out the operation, which is shown on the display if a valid scene is open.

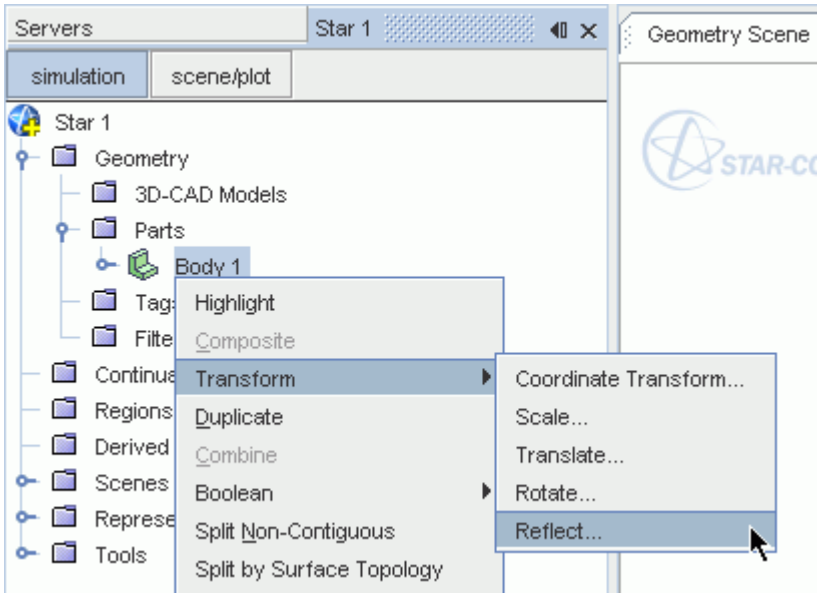
Reflecting a Part

Reflecting a part can be used to reposition it according to its mirror reflection in a given axis.

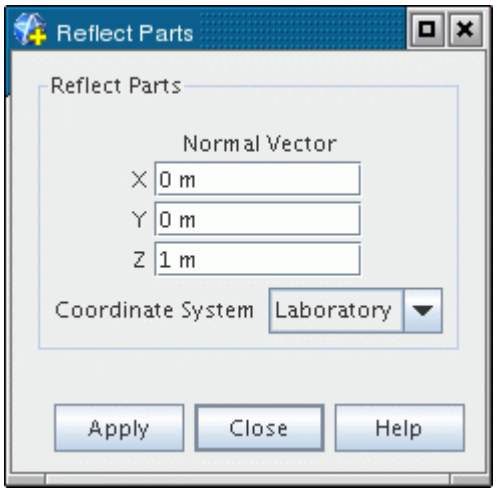
This transformation can be useful when the original part is a half model of the geometry. You can create the full model by first duplicating the part and then reflecting the copy (or original).

To reflect a part:

1. Right-click the part node and select **Transform** > **Reflect...** from the pop-up menu:



The *Reflect Parts* dialog appears:



2. Enter the **Normal Vector** values and select the **Coordinate System** before clicking **Apply**.

   If a valid scene is open, the part display is updated.

3. Clicking **Apply** a second time without changing the vector component values returns the part to its original position.

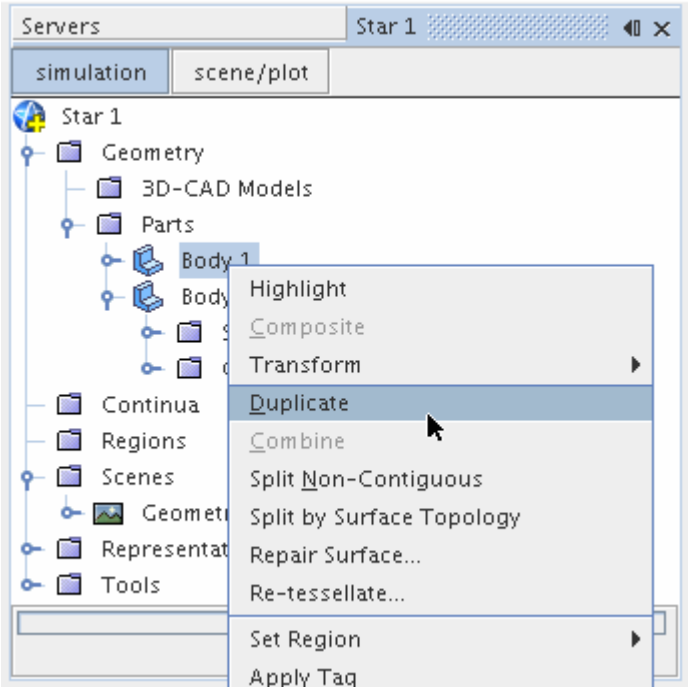
Duplicating a Part

You can duplicate a part in instances where only a half, quarter, or some fraction of a symmetrical or axis-symmetric model has been imported and the full geometry is desired instead.

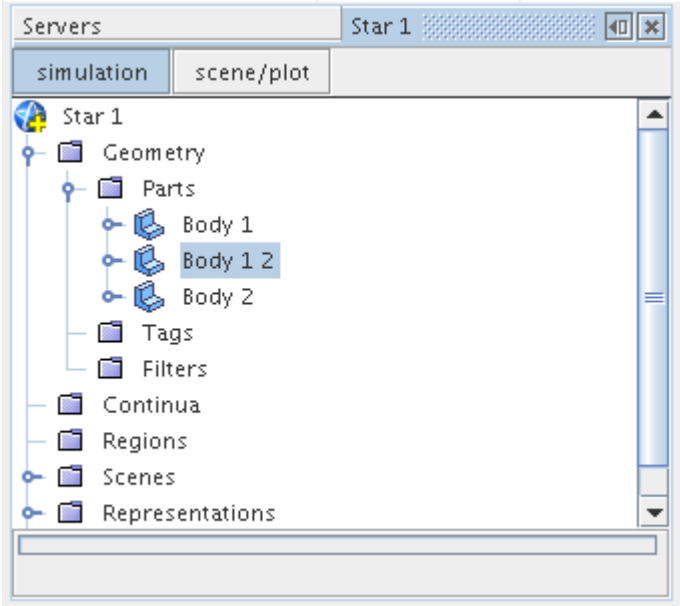
Duplicating a part produces an identical copy of it with the same properties as the original, other than region and contacts specification, which are left null. The duplicate also occupies the same location as the original until it can be transformed. If a 3D-CAD part is duplicated, the duplicate copy is a leaf-level CAD part rather than a 3D-CAD part. That is, the link to the originating 3D-CAD model is lost.

To duplicate a part:

1. Right-click the part node and select **Duplicate** in the pop-up menu:



If the source part includes any repairs, a copy of the part is made and retessellated (although the level of tessellation is retained). The new part node appears next to the original:

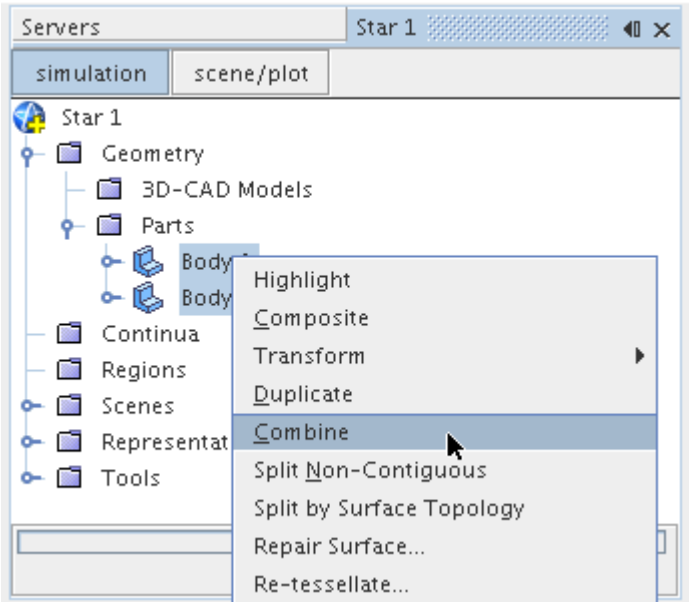


Combining Parts

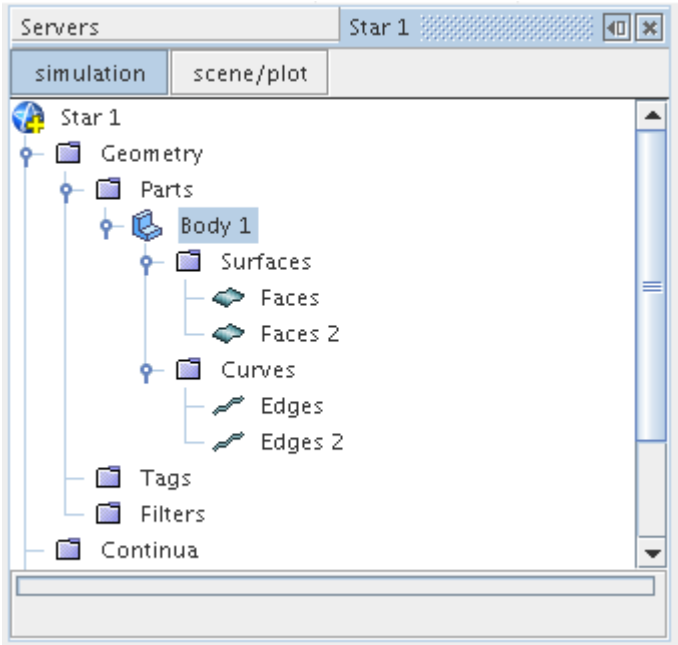
Two or more parts can be combined to create a single mesh part.

To combine parts:

1. Select the part nodes of interest, right-click and select **Combine** from the pop-up menu:

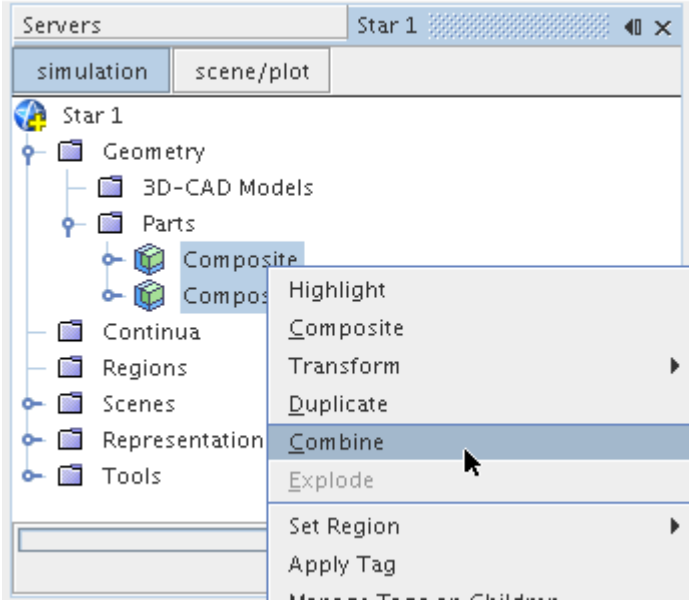


The first selected part is used as the base for all the other parts. All part surfaces and curves belonging to each part are grouped in the base part, as shown below:

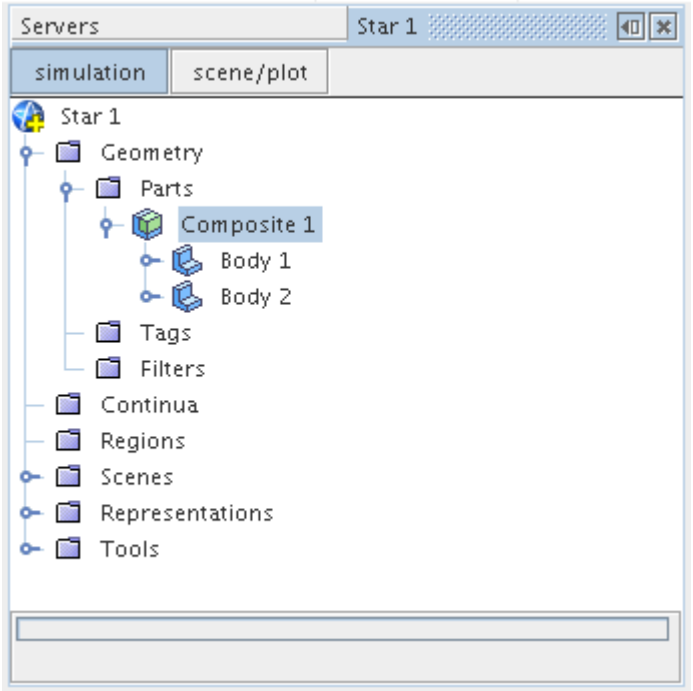


As this operation results in a single mesh part, it is not possible to reconstruct CAD parts from the single part. It is possible, however, to construct new mesh parts from one or more of the part surfaces contained within the combined part.

Two or more composite parts can also be combined as described above. For example:



All part bodies belonging to each composite are grouped into the first composite part selected:



Combining two composite parts in this way does not result in child CAD or 3D-CAD parts being converted to mesh parts. They retain their original type. You can undo the above process by first making a new empty composite part and then dragging and dropping the original part bodies back into the composite node.

Performing Boolean Operations

This section describes the basic Boolean operations that can be performed on geometry parts.

You can perform the following basic Boolean operations on geometry parts:

- Unite
- Intersect
- Subtract
- Imprint

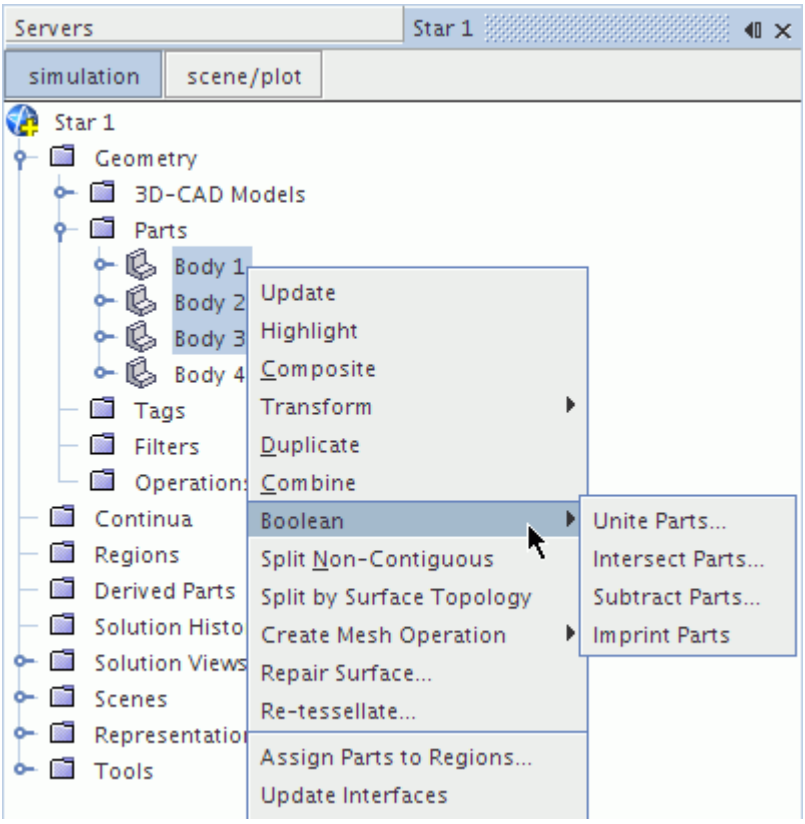
These Boolean operations are similar to the Boolean mesh operations. See Mesh Operations. However, performing a Boolean mesh operation also creates a **Geometry** > **Operations** > **[Mesh Operation]** sub-node which provides greater control when using the Parts Based Meshing (PBM) pipeline. See Advantages of Parts Based Meshing.

You can perform a Boolean operation on a combination of leaf-level and/or composite **Geometry** > **Parts**, or on a single composite part that contains at least two other parts.

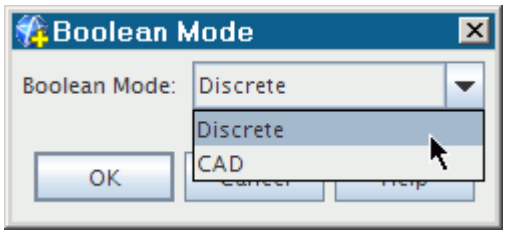
Following a Unite, Intersect, or Subtract Boolean operation, the original parts remain unchanged, and the resulting geometry is added to the tree under a new leaf-level node that is called **Unite**, **Intersect**, or **Subtract**. When performing a Boolean Imprint operation, the meshes on existing parts are modified, and matching faces are moved into a new part surface. Boolean Imprint operations do not create new parts – they only modify existing parts. The type of leaf-level part (CAD or mesh) that is created following a Boolean operation depends on the nature of the input parts and, where appropriate, which Boolean mode is selected. When a Boolean operation is carried out using parts from different levels in the tree, the resulting part is added to the tree at the same level as the highest selected part.

To create a Boolean operation:

1. Multi-select the CAD parts to include in the Boolean operation from the simulation tree.
2. Right-click one of the selected nodes and select one of the Boolean options, **Unite Parts**, **Intersect Parts**, **Subtract Parts**, or **Imprint Parts**.

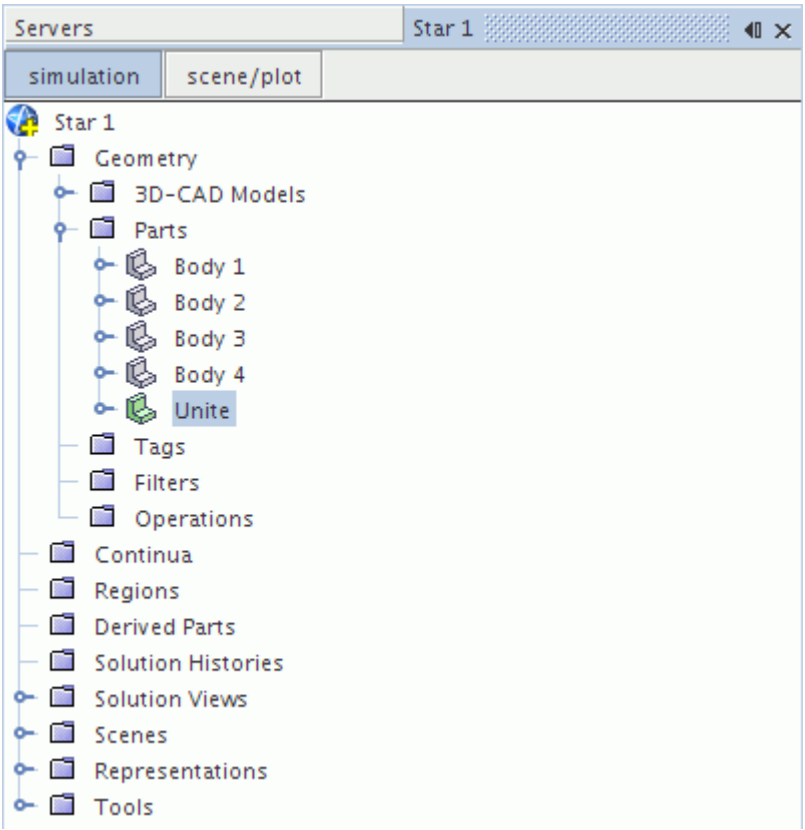


3. In the dialog that appears, set the **Boolean Mode** to either **CAD** or **Discrete**.



4. Click **OK**.

   A **[Boolean]** node is added to the tree for the part that the Boolean operation creates (except for Imprint operations).



The Boolean Mode

Boolean operations on geometry parts can be carried out using either the CAD data, if available, or the discretized surfaces of the parts.

The method that is used is known as the Boolean mode. If at least one of the selected parts is a mesh or shape part, including those parts within any selected composites, STAR-CCM+ uses the discretized surface (tessellated surface mesh) for all selected parts to carry out the operation. This mode is referred to as the **Discrete** Boolean mode, and in this mode the operation is carried out based on the current tessellation properties of each part. Any surface preparation that has been carried out on the input parts is incorporated into the final part, which is of Mesh type.

If the selected parts are all CAD-based parts (including 3D-CAD), when the operation is executed, you can choose between **CAD** or **Discrete** Boolean mode. When the **CAD** option is selected, the operation is carried out based on the original CAD data and the resulting part is of CAD type. If you have carried out any surface preparation on the CAD parts, such as surface repair or altering the tessellation density, these changes only affect the discrete surfaces of the part, not the CAD data. Therefore, to preserve the effects of the surface preparation in the resulting part, select the **Discrete** Boolean mode.

Unite Parts

This Boolean operation joins all the selected parts into one single part, the surface of which is formed from segments of the constituent part surfaces.

This operation differs from the Combine operation, which combines the selected parts into one part, but retains the part surfaces as separate, whole surfaces. To carry out a unite operation successfully, the selected

parts must be in contact with each other, or overlap. If the parts do not meet this criterion, the part that the Boolean operation creates contains no surface data.

Intersect Parts

This Boolean operation finds the common volume that the selected parts share and creates a single part that is based on this volume.

The surface of the part is formed from segments of the constituent part surfaces. To carry out an intersect operation successfully, all the selected parts must share a common volume. If the parts do not share a common volume, the part that the Boolean operation creates contains no surface data.

Subtract Parts

This Boolean operation subtracts a defined shape from the volume of a target part.

Any number of parts can be used to define the shape to subtract. The surface of the resulting part is formed from segments of the constituent part surfaces.

Imprint Parts

This Boolean operation modifies the meshes on existing parts and merges coincident surfaces from the selected parts together into a new part surface.

When coincident surfaces are not identical (in shape or size), the edges of the smaller surface are used to split the larger surface. By imprinting parts, you can successfully generate interfaces between multiple connected parts. The interfaces are essential for conformal meshing.

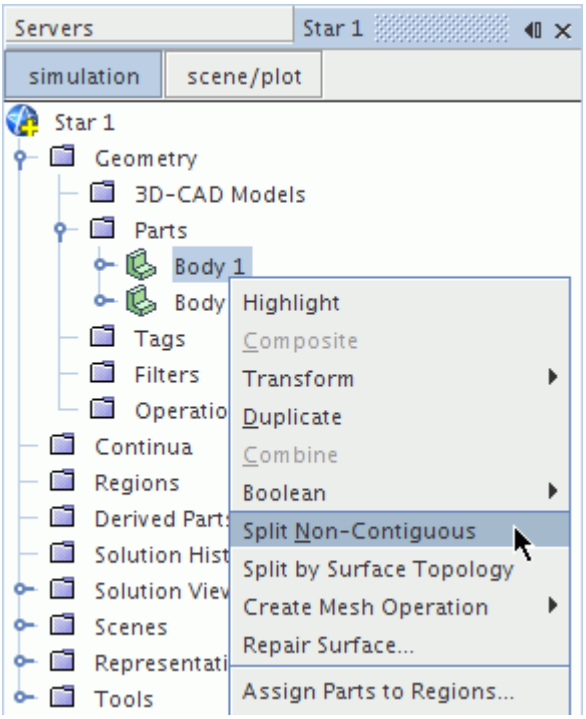
Splitting Non-Contiguous Parts

A non-contiguous mesh part (that is, a mesh part that has faces that are not connected to other faces in the part) can be split into its individual component parts if necessary.

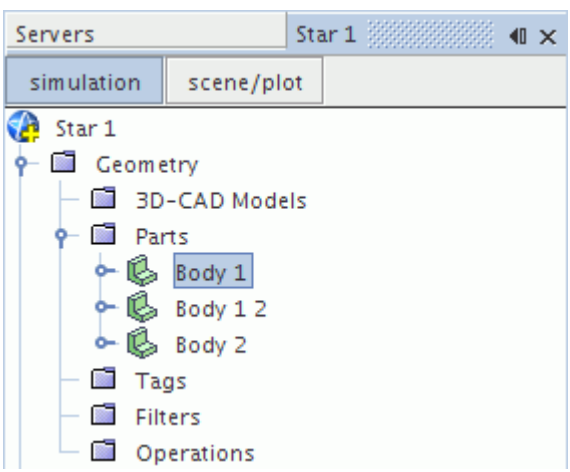
If the part contains a baffle, any baffles that remain inside the original part after splitting are preserved. Other contacts and baffles that move into the other split parts are not preserved.

To split non-contiguous parts:

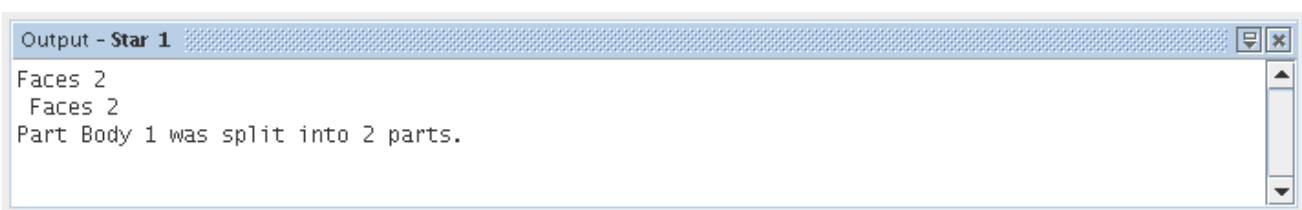
1. Right-click the part node and select **Split Non-Contiguous** from the pop-up menu:



The splitting operation proceeds and produces an additional part node for each contiguous component:



The output from a successful split operation is shown below:



Attempting to split a part that is already contiguous has no effect.

Splitting a Part by Surface Topology

The split by surface topology method can be used to split and combine parts that can be defined as distinct geometrical regions.

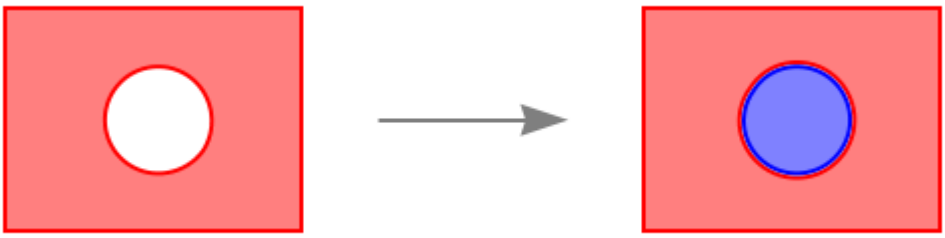
It is similar to the method used to split regions. The operation is based on the topology of closed, manifold, non-intersecting surfaces that the parts define. When assessing the surfaces, STAR-CCM+ considers the surfaces that are contained within one part to be a single surface. You do not need to combine or split any surfaces before carrying out the operation.

The technique allows you to:

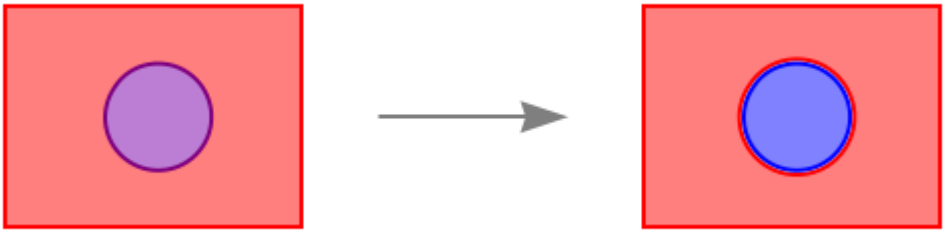
- Separate two or more closed surfaces that are not in contact into two or more parts.

- Separate two or more closed surfaces that share a common interface.

- Create new parts from internal voids.



- Redefine the surfaces of parts that contain closed surfaces or other whole parts.



- Combine parts with shared part surfaces.

When this technique is used for multiple parts, STAR-CCM+ internally combines all the parts and treats them as a single part. If a part is not modified in the operation, it is returned as a CAD part with the same name as it had previously. If a part is modified, for example it gained extra surfaces from other selected parts, it is returned as a Mesh part.

To split a part by surface topology:

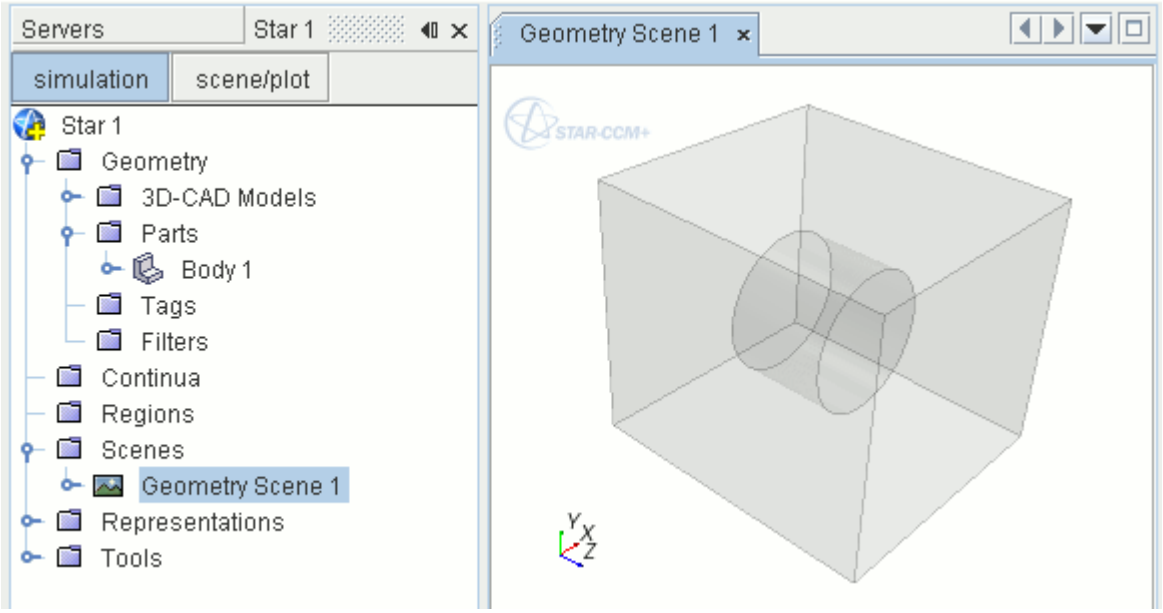
1. Right-click the part and select **Split by Surface Topology**.
2. If you wish to include several parts in the operation, select the parts using **<Ctrl><Click>**.
3. If you want to split parts by surface topology but combine parts that share a part surface, in the *Split Parts Options* dialog that appears, activate *Combine Parts With Shared Part Surfaces*.

Using Split by Surface Topology

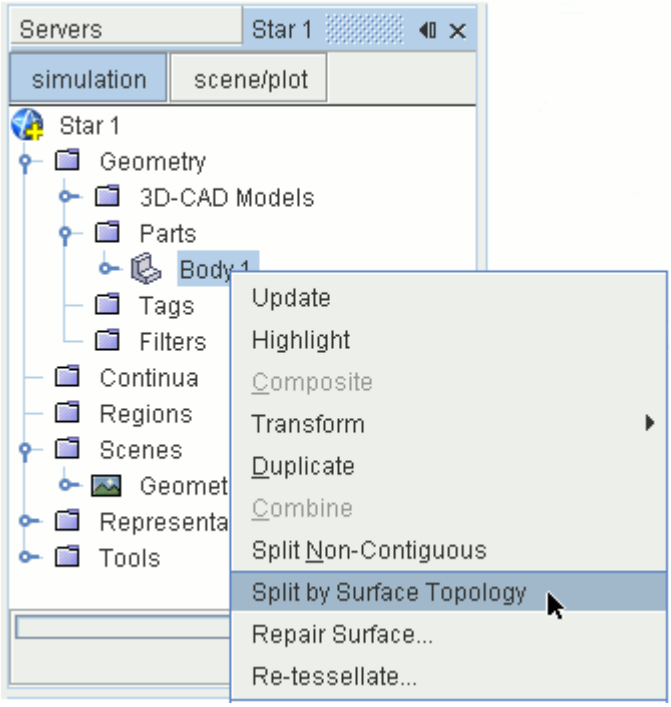
An example where a part contains an inner void and one wishes to create another part from the inner surface is illustrated below.

4. Identify parts that can be split into one or more separate parts.

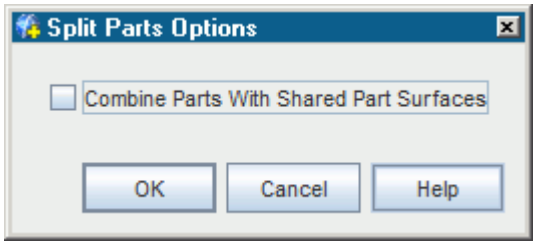
   In the figure below, the single part **Body 1** contains a cylindrical void.



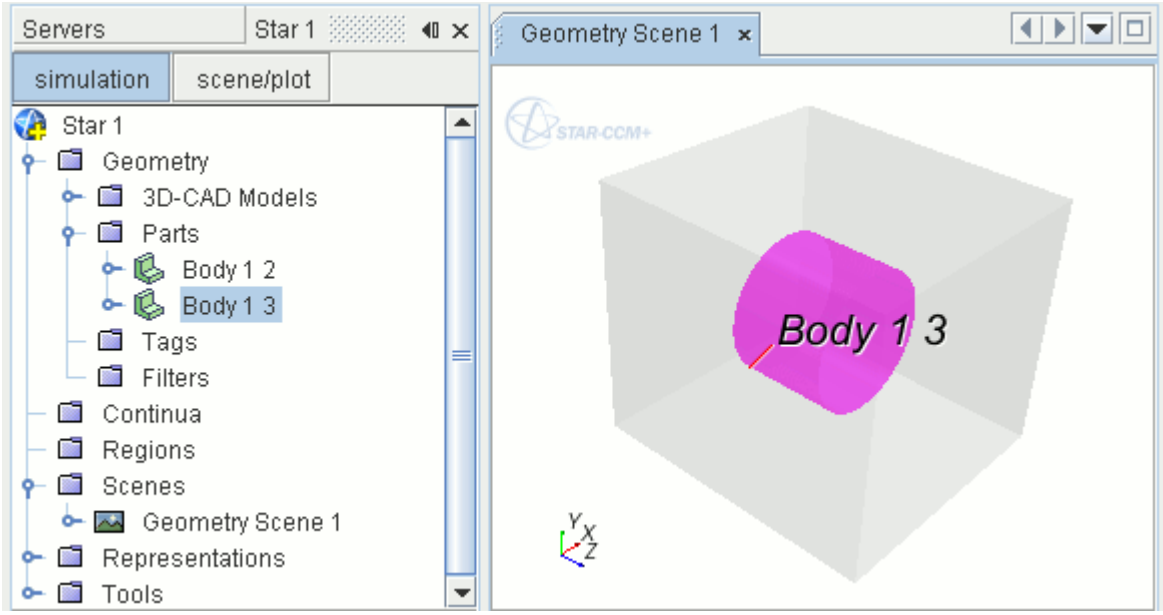
5. Right-click the part that you want to split and choose **Split by Surface Topology**:



6. In the *Split Parts Options* dialog, click **OK**.



The result can be viewed in a geometry scene as shown below:



If separate part surfaces surrounded the inner void in the original part, these surfaces would be replicated in the splitting process. The source part (**Body 1** in the example above) is removed in the splitting process.

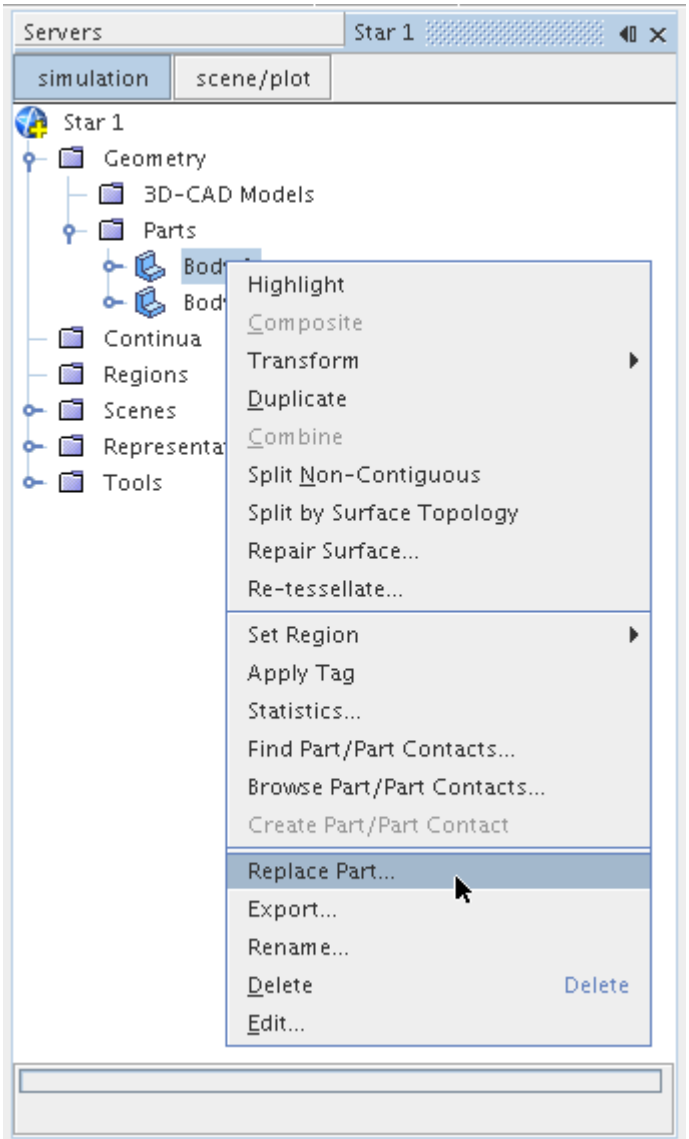
Replacing a Part

You can replace a single CAD or mesh part at any time. Replacing a part lets you provide new geometrical input for the meshing pipeline without affecting any of the other data input required to generate a mesh.

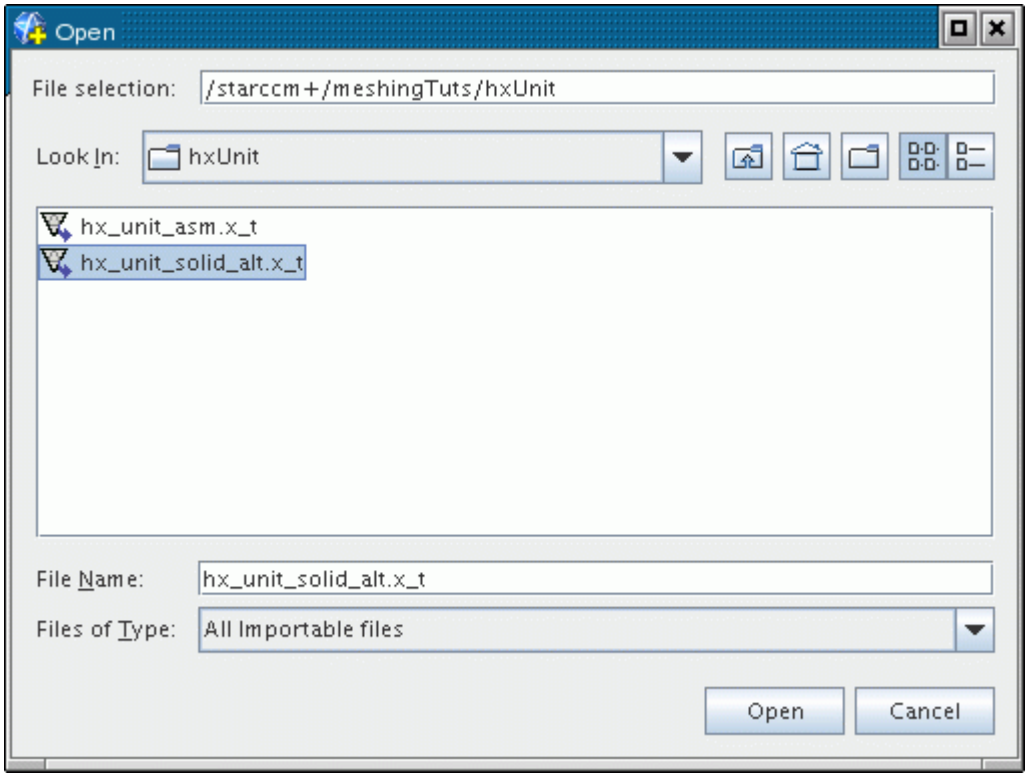
Only one part node can be replaced at a time. The operation does not apply to shape parts.

To replace a part:

1. Right-click the part node and select **Replace Part...** from the pop-up menu:



The *Open* dialog appears:



2. Select the replacement part file and click **Open** to import the data.

   When importing IGES files, to account for possible gaps that can arise in the sheet bodies during the translation, the following additional options are available:

   - Sewing Tolerance
   - Multi-Pass Sewing

The replacement part must be of the same type as the leaf-level node part that it is replacing. That is, a CAD part can only be replaced with a part originating from native CAD or neutral format data. A mesh part can only be replaced with a part originating from a surface mesh data file. The file types available in the *Open* dialog reflect this rule.

Also note that any manipulations or manual repairs that have been made to the original part are lost when the part is replaced.

Manipulating Part Surfaces

STAR-CCM+ allows leaf-level part surfaces to be manipulated in several different ways.

The following operations are possible depending on the part type:

- creating a part from a part surface
- creating an empty part surface
- combining part surfaces
- splitting non-contiguous part surfaces
- splitting a part surface by part curves

- splitting a part surface by angle
- splitting a part surface by patches
- splitting a part surface by contacts

In order to make part surfaces easier to search for, it is also possible to apply user tags to them.

Contents:

Creating a New Part from a Part Surface
Creating an Empty Part Surface
Combining Part Surfaces
Splitting Non Contiguous Part Surfaces
Splitting a Part Surface by Part Curves
Splitting a Part Surface by Angle
Splitting a Part Surface by Patch
Splitting a Part Surface by Contacts
Selecting Patches for Part Surfaces

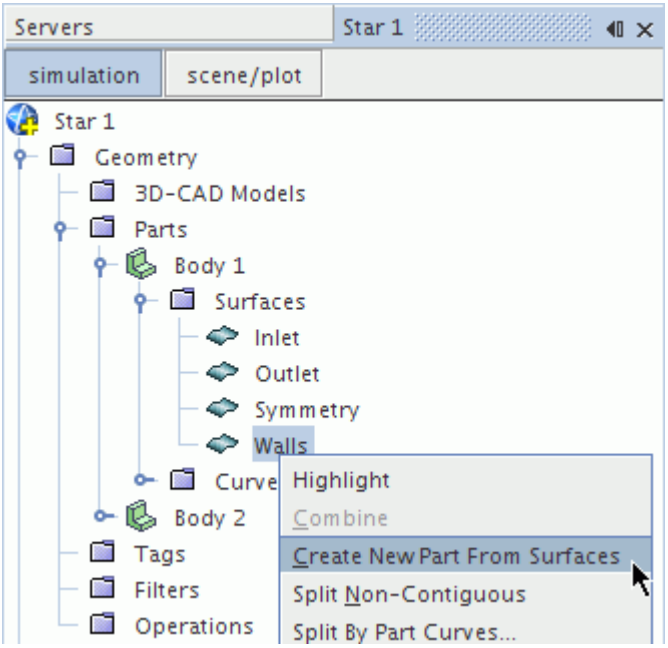
Creating a New Part from a Part Surface

Leaf-level parts can have one or more of the surfaces describing the part geometry be moved into a new part if necessary.

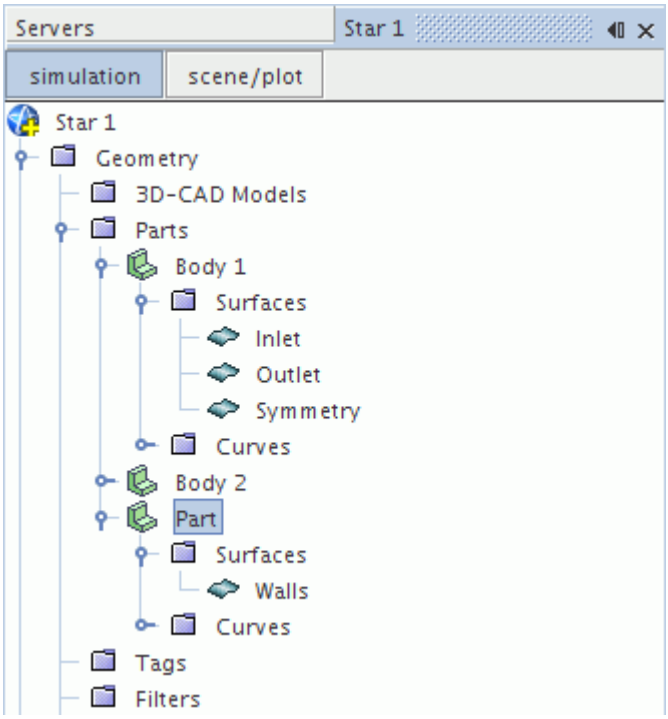
This operation would be useful for extracting a fluid volume from an enclosing solid part, for example.

To create a part from a surface node:

1. Right-click the surface node or nodes belonging to the same part and select **Create New Part From Surfaces** in the pop-up menu:



A new part with the default name **Part** is created using the surface node geometry. The surfaces retain the same names:



The original surface nodes that are selected are removed from the originating part. This operation only applies to surfaces belonging to a single part.

Creating an Empty Part Surface

In STAR-CCM+, you can create empty part surfaces for an existing part. When using surface repair features, you can then assign faces to the empty part surfaces.

To create an empty part surface for any part:

1. Expand the **Geometry** > **Parts** > **[Part]** node.
2. Right-click the **[Part]** > **Surfaces** node and select **New Part Surface**.

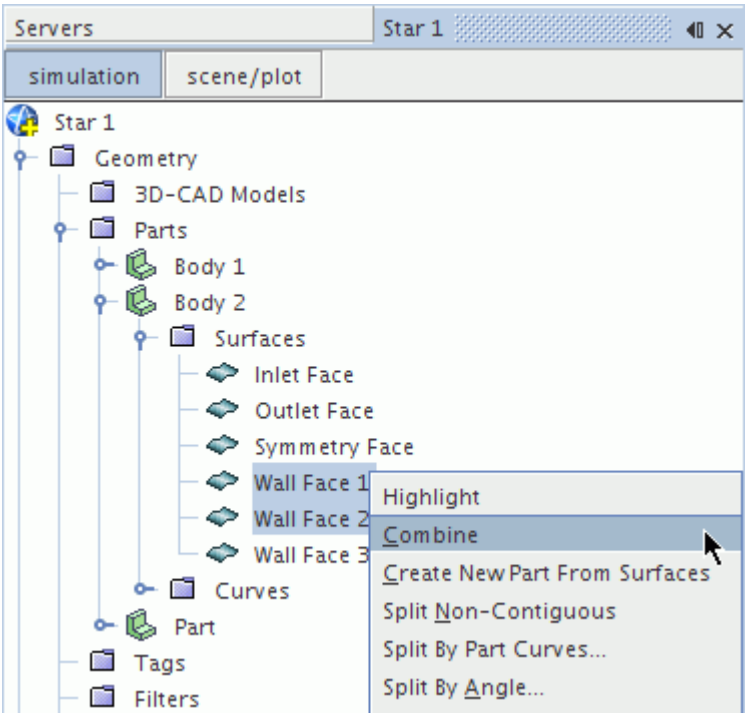
   An empty **Surfaces** > **[Part Surface]** node appears.

Combining Part Surfaces

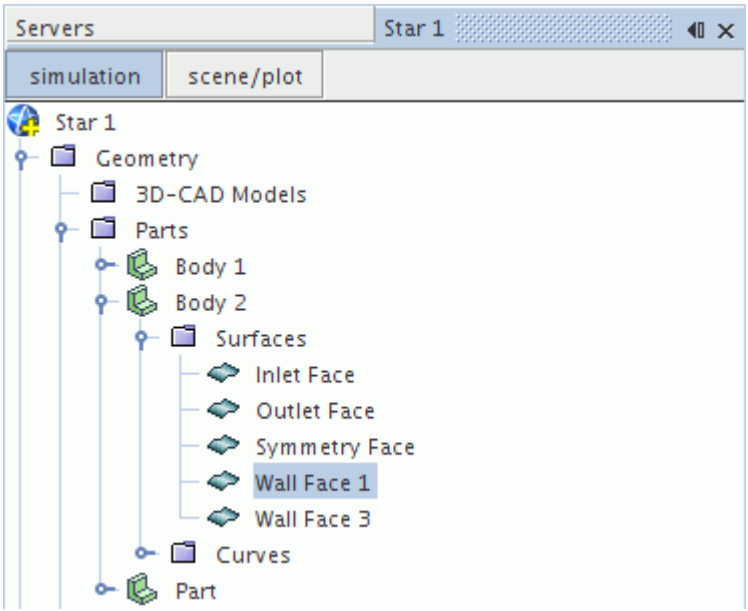
Two or more leaf-level part surfaces can be combined into a single part surface if necessary.

To combine the surfaces:

1. Select two or more surface nodes belonging to the same part, right-click and select **Combine** from the pop-up menu:



The surface nodes are combined into a single node (the first selected):



Note the following:

Boundary assignments on the secondary surface nodes are lost when they are different from the target (first selected) surface node.

Tag information relating to the secondary surface nodes are lost when they are different from the target (first selected) surface node.

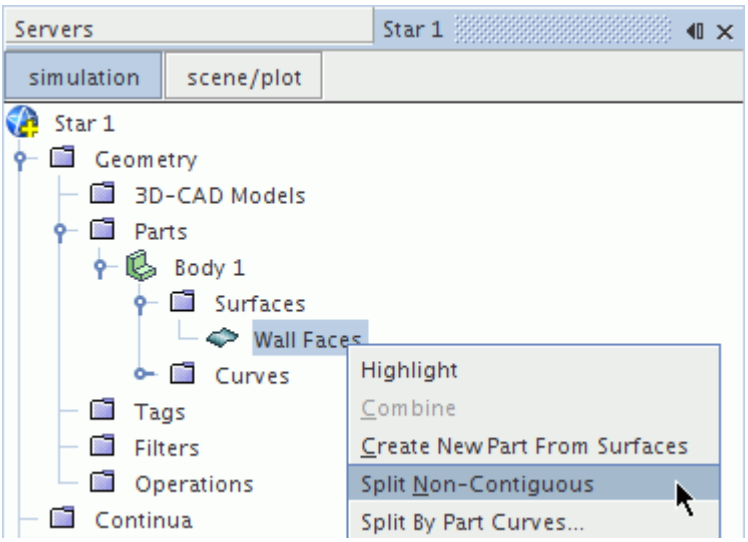
Splitting Non Contiguous Part Surfaces

A part can have one or more of its faces split based on the non-contiguous nature of the surface.

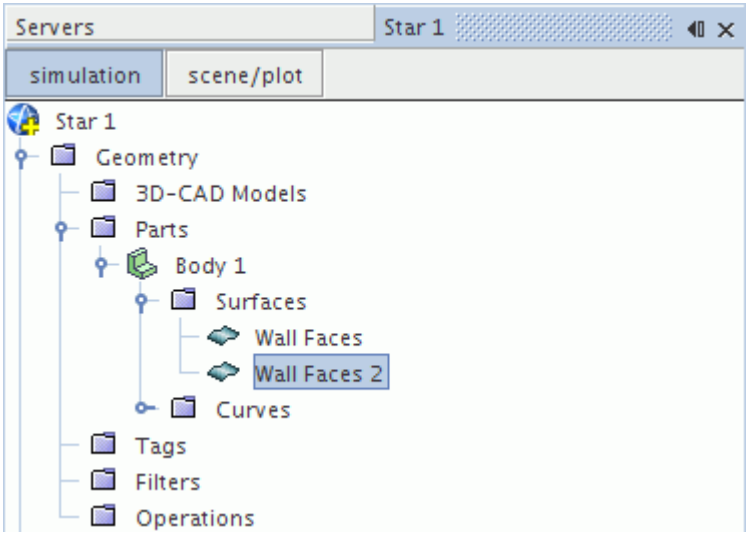
This operation splits the selected surfaces according to whether they are connected to their neighbors or not.

To split a surface:

1. Right-click the surface node and select **Split Non-Contiguous** from the pop-up menu:



Any non-contiguous surfaces are split to form new surface nodes, as shown below:



Any tags belonging to the original surface nodes are not copied to the split nodes. The new surfaces are also not assigned to any boundary.

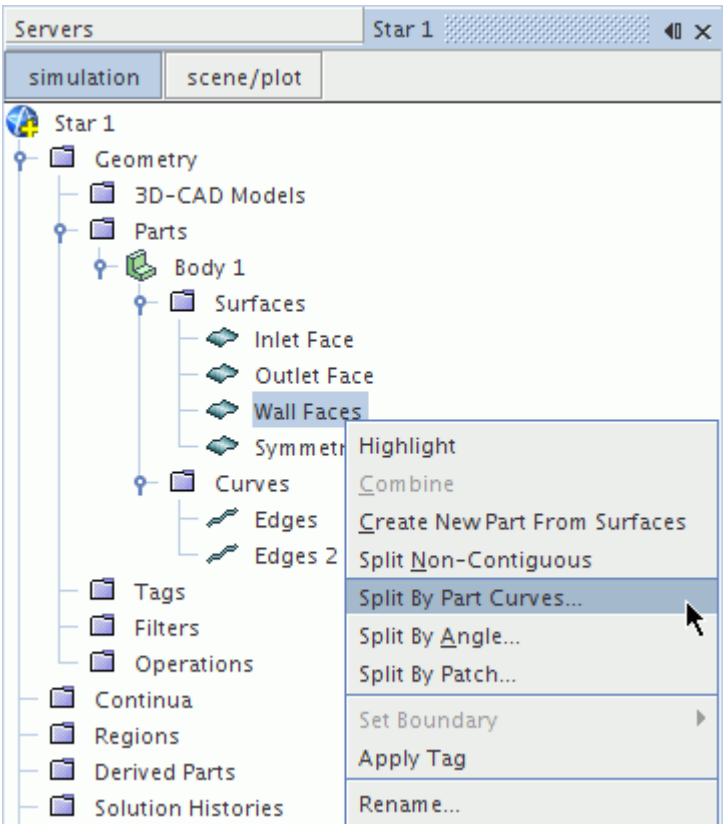
If the selected surfaces are contiguous, no action is taken.

Splitting a Part Surface by Part Curves

A part can have one or more of its surfaces split based on the part curves corresponding to the parent part.

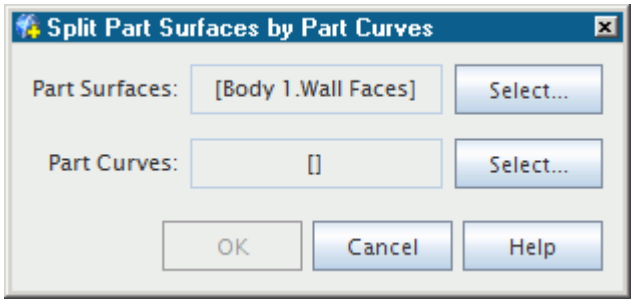
To split a surface using curves:

1. Right-click the surface node and select **Split By Part Curves...** from the pop-up menu:

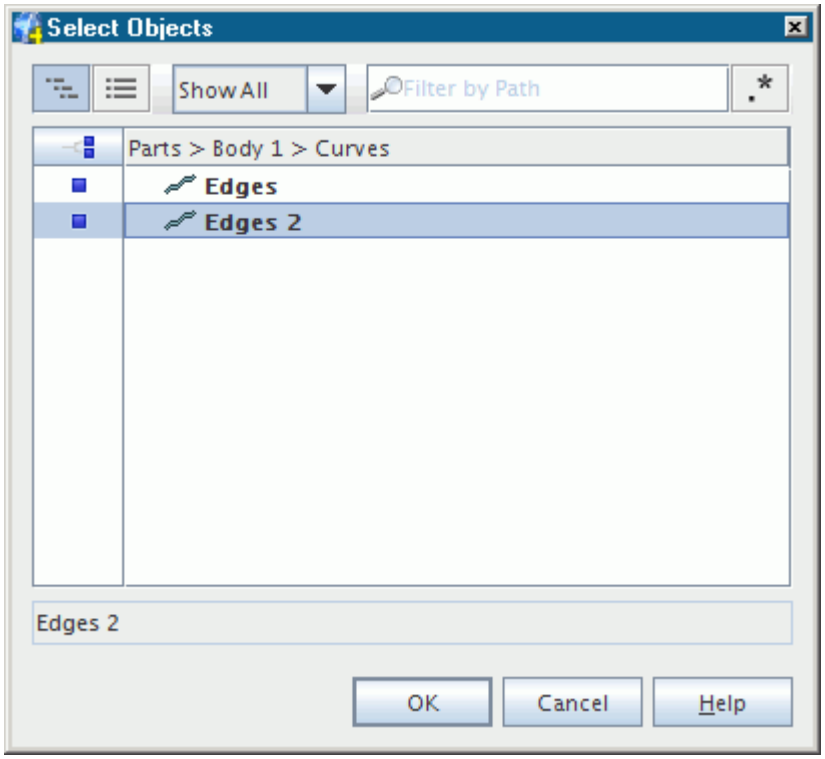


   In the *Split Part Surfaces by Part Curves* dialog, the originally selected surface nodes are already selected.

2. To select the appropriate part curves for the surface click the **Select** button to the right of **Part Curves**:

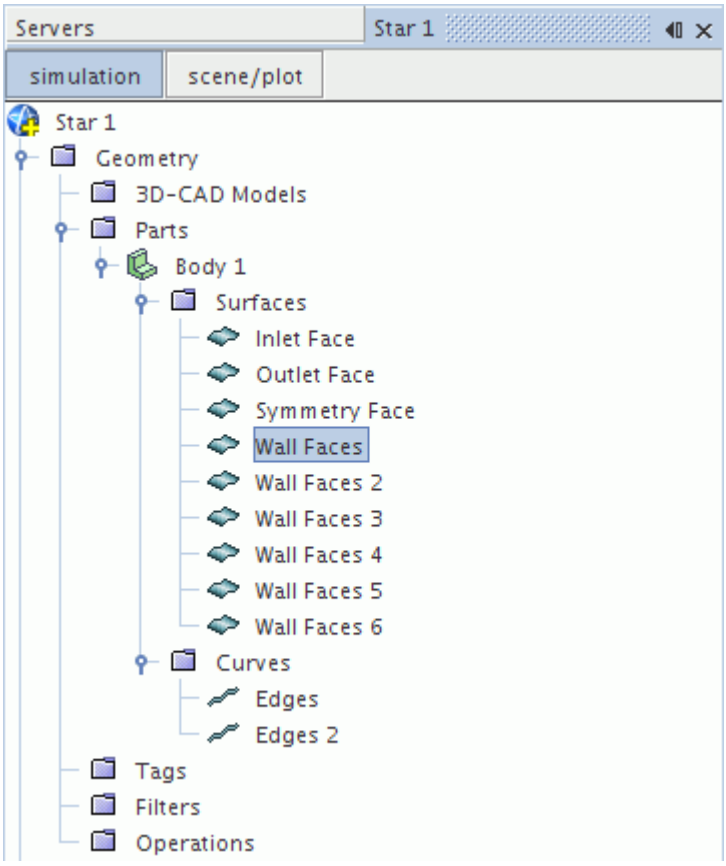


3. In the *Select Objects* dialog, select the part curves that you require.



4. Click **Apply** to carry out the operation, and **Close** to close the *Split Part Surfaces by Part Curves* dialog.

Any new surfaces that the process creates are updated in the simulation tree:



Any tags belonging to the original surface nodes are not copied to the split nodes. The new surfaces are also not assigned to any boundary.

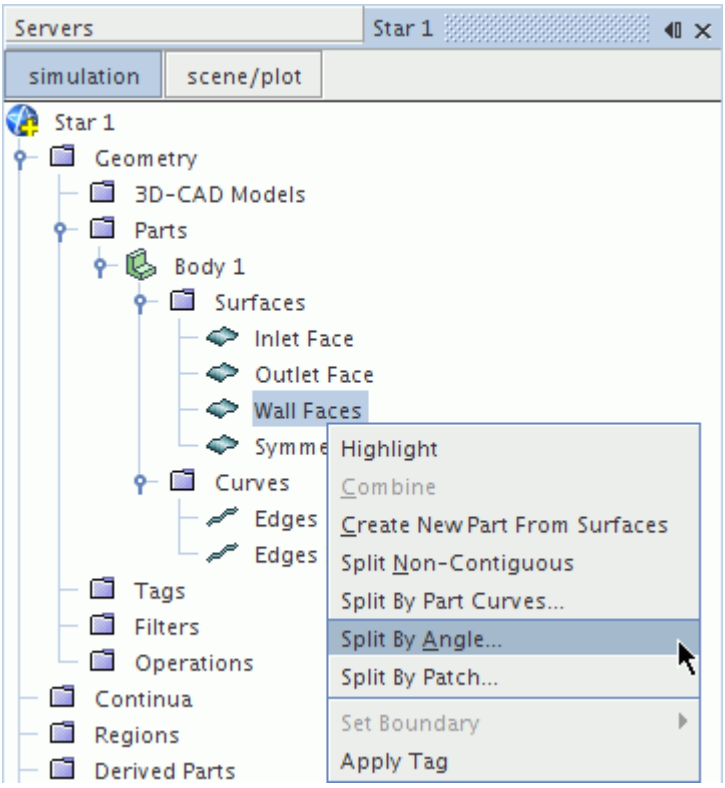
If no curves correspond with the selected surfaces, no action is taken.

Splitting a Part Surface by Angle

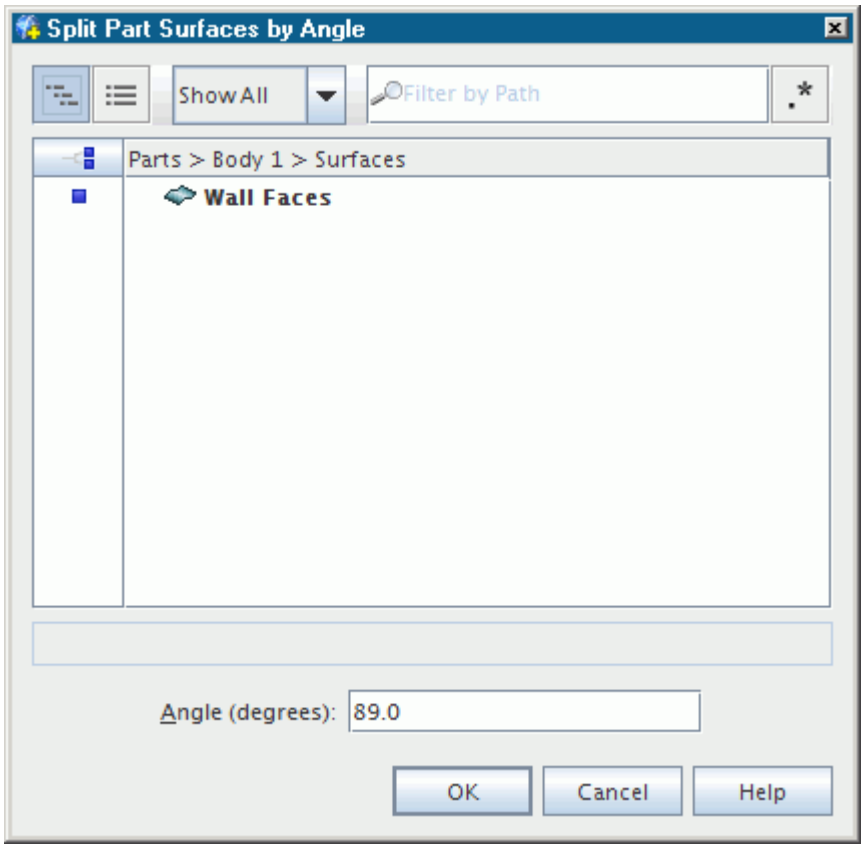
A part can have one or more of its surfaces split based on an angle.

To split a surface using an angle:

1. Right-click the surface nodes and select **Split By Angle...** from the pop-up menu:

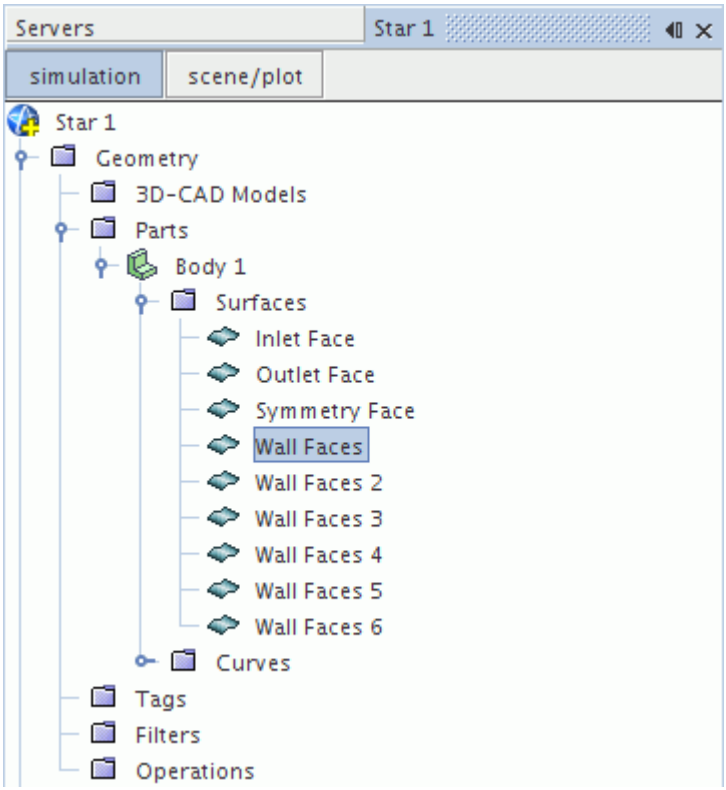


The *Split Part Surfaces by Angle* dialog appears:



2. You can select more surfaces to split by clicking **Select** button to the right of **Part Surfaces**.
3. Enter the **Angle** value (in degrees) and click **Apply** to carry out the operation.

The new surface nodes are shown in the simulation tree:



Any tags belonging to the original surface nodes are not copied to the split nodes. The new surfaces are also not assigned to any boundary.

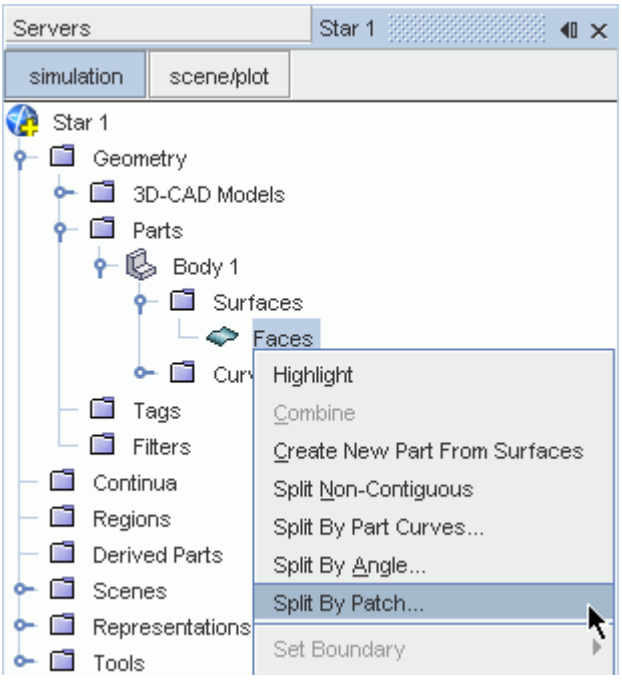
If the supplied angle value is too large, the split operation results in no additional surface nodes being created.

Splitting a Part Surface by Patch

You can split one or more surface nodes of a leaf-level part using the underlying patch information that is stored as part of the geometry.

To split a surface using patches:

1. Right-click the surface node and select **Split By Patch...**



Note: You must have a scene open to use this feature.

The *edit* tab opens in the explorer pane. The *Patch Selection Control* tools are not available until at least one *Part Surface Patch* is selected.

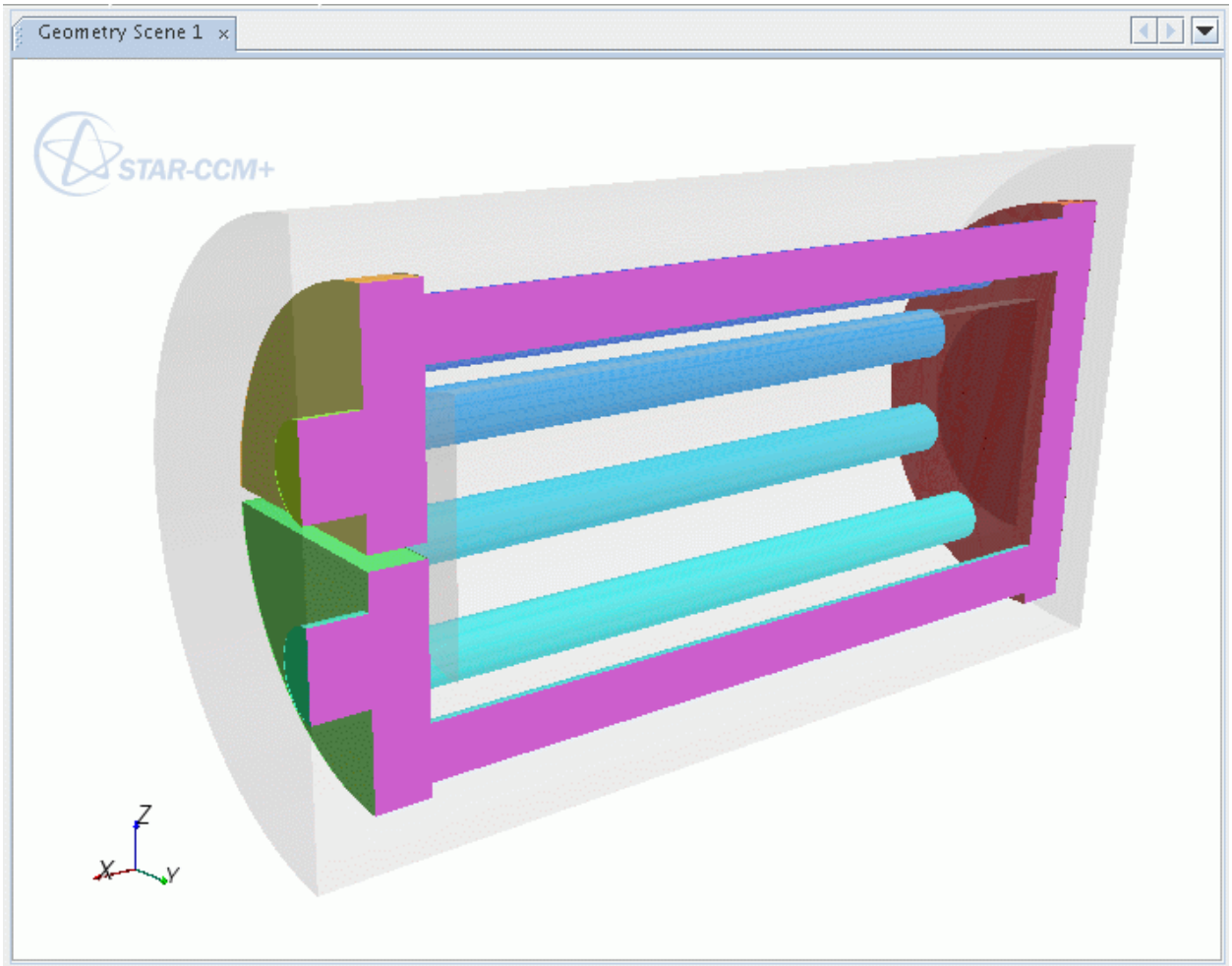
Here, you can edit the patches belonging to the selected surface node:

2. Select the part surface patches from the list box or from the geometry using the mouse in the current scene display.

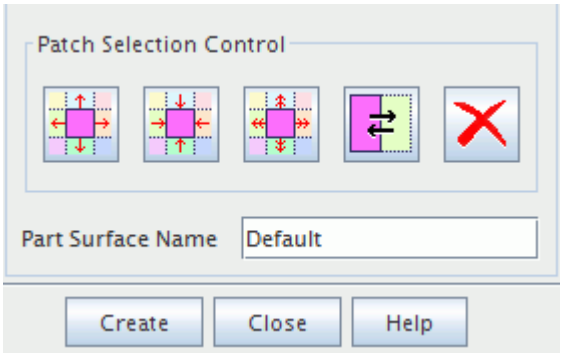
   To select multiple patches:

   - Press the `<Ctrl>` key and click on multiple part surface patches, or
   - use the Patch Selection Control tools, or
   - use the zone selection tool from the Selection toolbar

   Selecting a patch results in it being highlighted in the scene window.

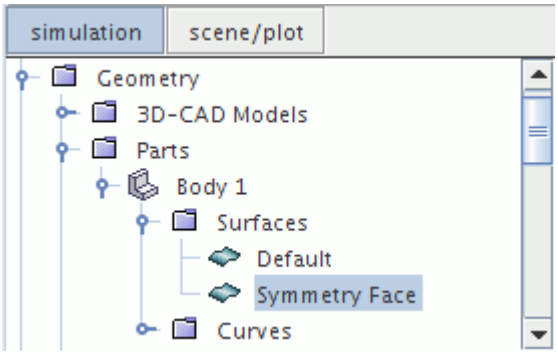


3. Enter the new **Part Surface Name**.



4. Click **Create** to create a surface node that is based on the selected patch information.
5. Repeat these steps for all required patches.
6. Click **Close** when done.

The new surface node for the part appears in the simulation tree.



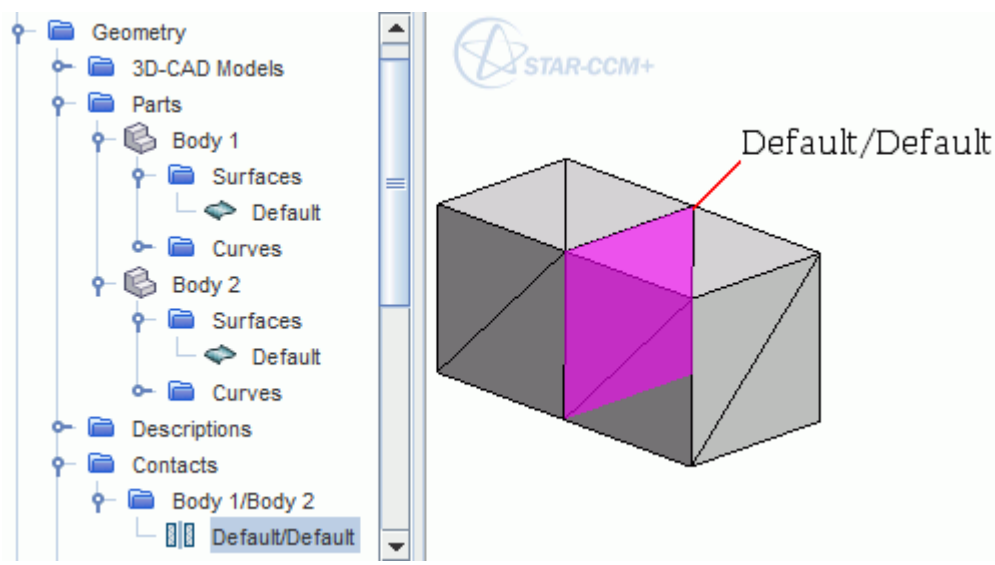
Note: Any tags that belong to the original surface nodes are not copied to the split nodes. The new surfaces are also not assigned to any boundary.

Splitting a Part Surface by Contacts

You can split one or more part surfaces by the contacts that the part surfaces make.

The **Split by Contact** function only works with strong contacts that are created in the **Root Description**. For example, parts that are imprinted in 3D-CAD have strong contacts. If contacts are created in any other downstream description, **Split by Contact** will not work.

When parts are imprinted, the contacts that are created exist between the whole part surfaces that contain the contacting faces. However, when the part surface contact is selected, only the contacting faces are highlighted in the Graphics window:



Splitting part surfaces by contacts recognizes the contacting faces on coincident surfaces and specifies these collections of faces as independent part surfaces.

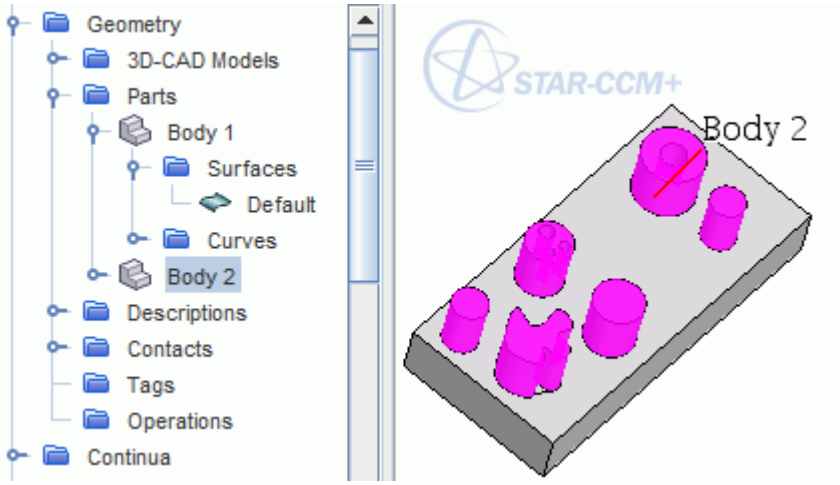
To split part surfaces using contacts:

1. Select at least one part surface node to split by contact.

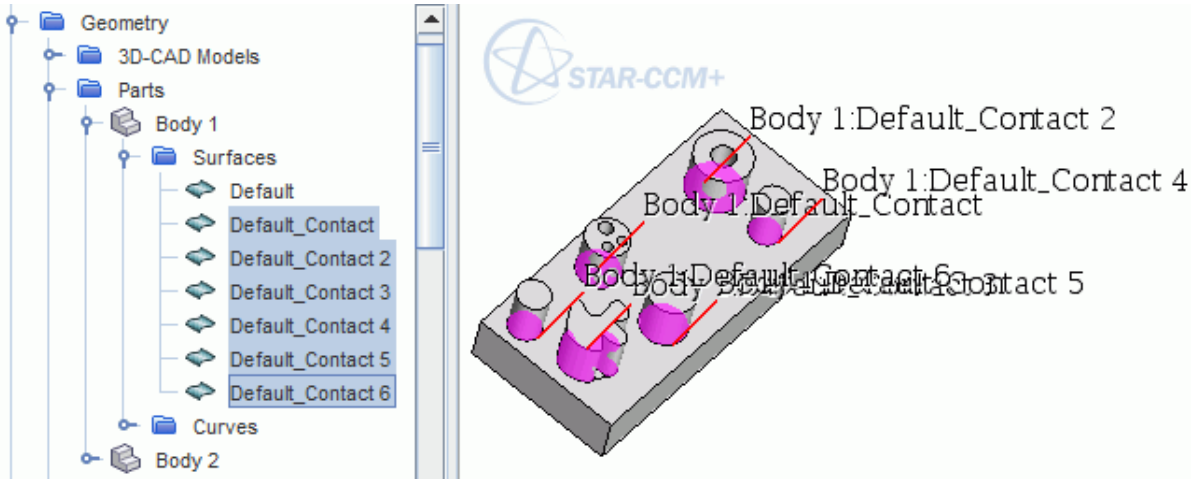
2. Right-click the part surface node and select **Split By Contact**.
3. In the **Split By Contact Options** dialog, select an option for *Create One Part Surface Per Contacting:* as follows:
   - **Patch**

     A separate part surface is created from faces that belong to each patch within the contacting part surfaces.

     For example, if there are six patches contacting a part surface:



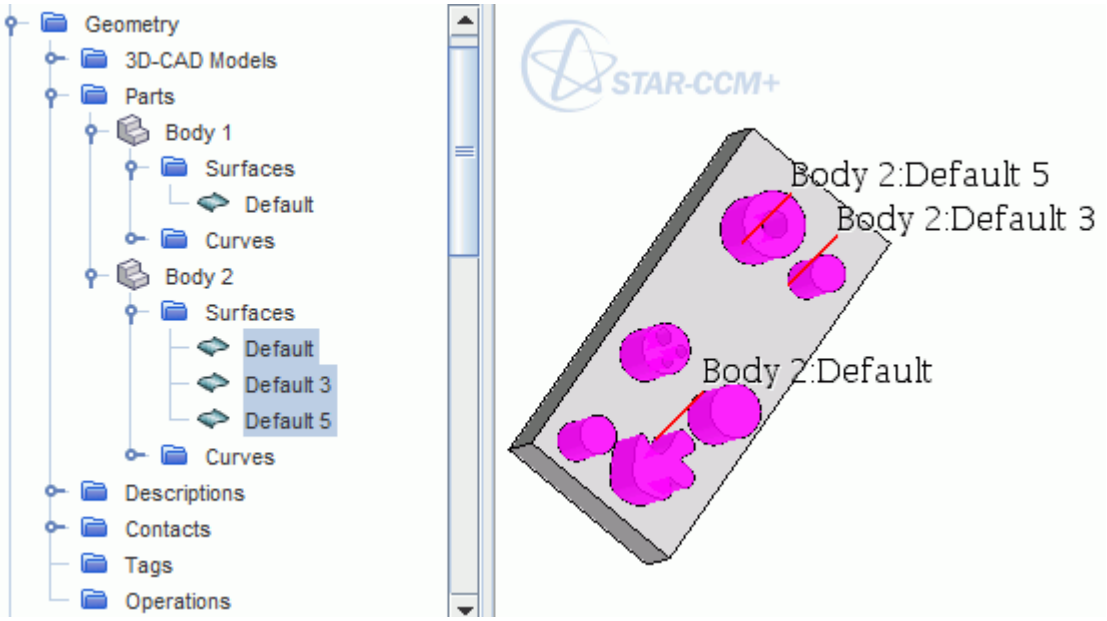
     the part surface is split to create six corresponding part surfaces:



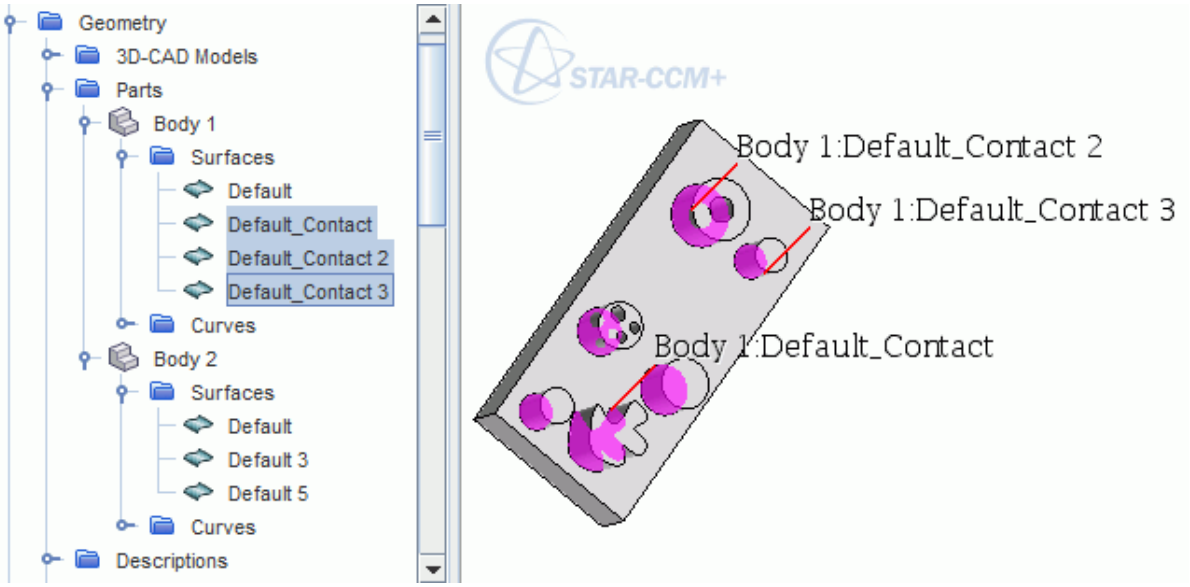
   - **Part Surface**

     A separate part surface is created from faces that belong to each contacting part surface.

     For example, if a part contains three part surfaces:



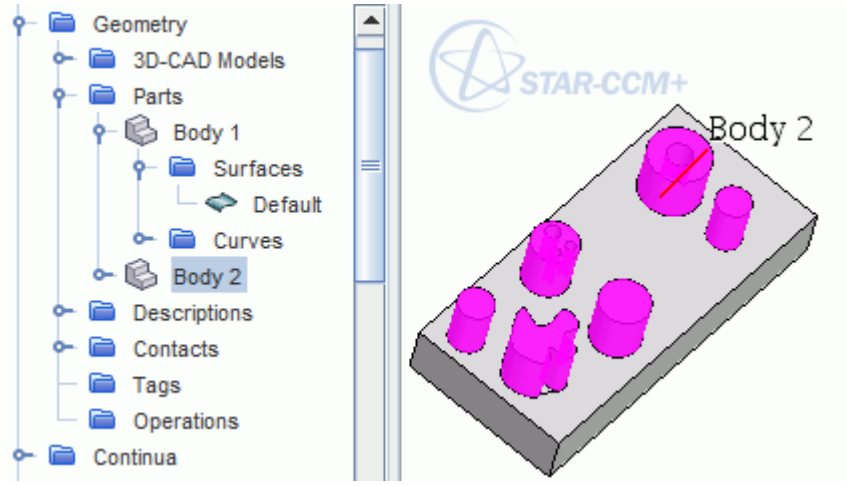
the contacting part surface is split to create three corresponding part surfaces:



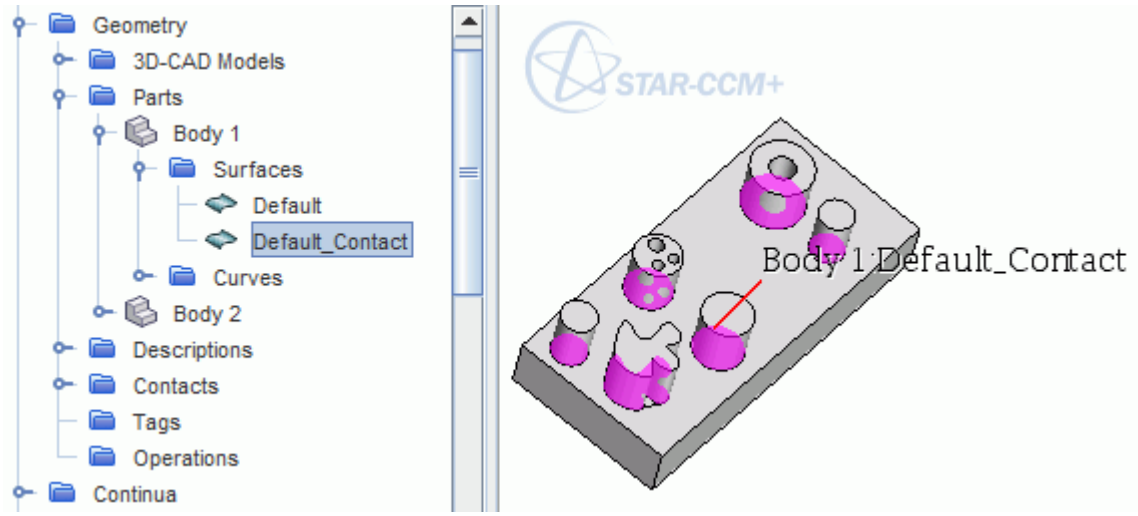
- **Part**

  A separate part surface is created from faces that belong to each part within the contacting part surfaces.

  For example, if one part surface is contacted by one part:



the part surface is split to create one corresponding part surface:



In each example, specific contacting part surfaces are created that contain the exact faces that are in contact from each original patch, part surface, or part. The **Geometry** > **Contacts** update to show the specific part surfaces that form the contacts.

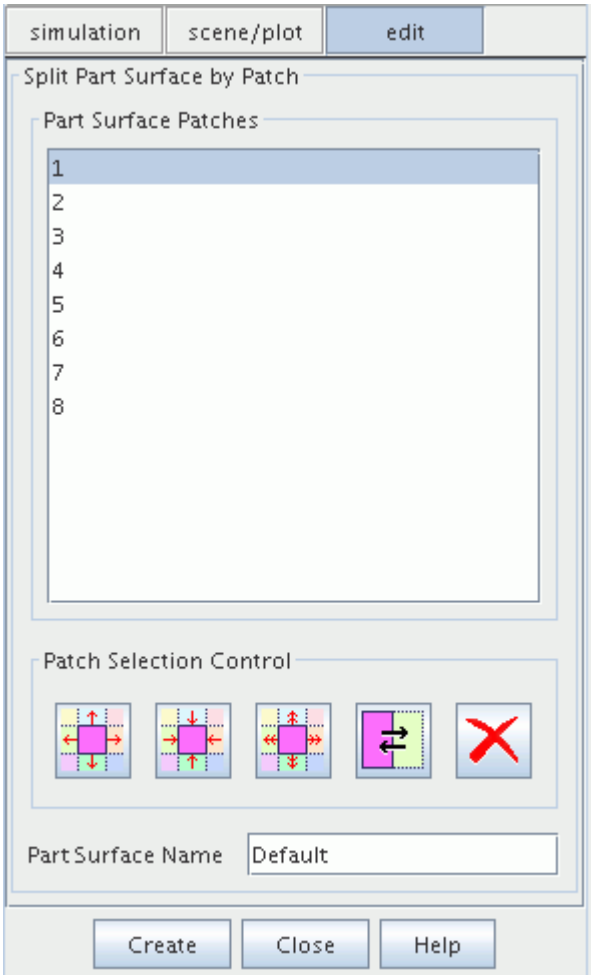
Selecting Patches for Part Surfaces

The *Patch Selection Control* tools include grow and shrink tools which facilitate the selection of patches for splitting.

These *Patch Selection Control* tools are only available for parts. There are no tools included for region boundary based patch splitting.

There are five *Patch Selection Control* tool buttons which are accessible once you select the Split by Patch option. These buttons become active once a *Part Surface Patch* has been selected from the *Part Surface Patches* group box or from the Geometry in the *Graphics* window. Use the <`up`> and <`down`> arrow keys on

the keyboard to cycle through the patches. If all patches for the part surface are selected, then only the **Patch Clear** and **Patch Shrink** buttons are available.



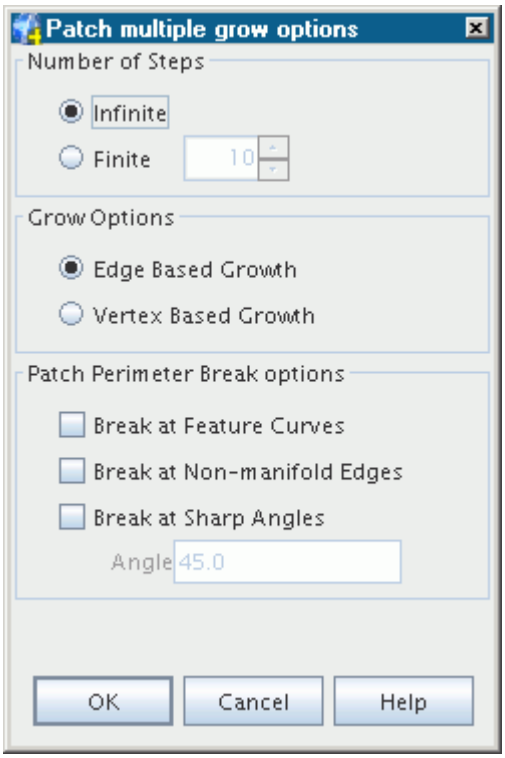
The button descriptions are as follows:

Tool	Hotkey	Description
(**Grow Selected Patches**)	`SHIFT + G`	Grows the selection one patch in each direction based on the perimeter vertices surrounding the initial patch selection.
(**Shrink Selected Patches**)	–	Shrinks the selection one patch in each direction based on the perimeter vertices of the selected patches. If all patches are selected then a shrink may not be possible. At least one free patch perimeter edge is required to start shrinking from. The result of growing a patch and then shrinking it does not always result in the original patch selection.

Tool	Hotkey	Description
(**Multi-Grow Selected Patches**)	SHIFT + M	Opens a dialog which gives you access to a variety of patch multiple grow options.
(**Invert Selected Patches**)	SHIFT + I	Selects the inverse of the **Part Surface Patches** that are selected.
(**Clear Selected Patches**)	SHIFT + N	Clears any selection of **Part Surface Patches**.

Patch multiple grow options dialog

The default patch multiple grow options are shown in the dialog below:



Descriptions of each of the patch multiple grow options are specified below:

Number of Steps	Specifies the number of steps the patch selection is grown for.	
	Infinite	Grows the patch selection infinitely. Performing an infinite grow without including some reasonable Break Options can potentially take a long time.
	Finite	Allows you to specify a finite number of patch growth steps.

Grow Options	Allows you the option to grow your selection based on growth from edges or vertex.	
	Edge Based Growth	Grows the selected patches from the outer edges of each selected patch.
	Vertex Based Growth	Grows the selected patches from the outer vertex of each selected patch.
Patch Perimeter Break Options	Allows you to specify the type of feature which the patch perimeter grows to before breaking the selection. You can set the *Patch Perimeter Break options* to any or all of the following options:	
	Break at Feature Curves	Breaks the patch selection when the patch perimeter reaches a feature curve.
	Break at Non-manifold edges	Breaks the patch selection when the patch perimeter reaches a non-manifold edge.
	Break at Sharp Angles	Breaks the patch selection when the patch perimeter reaches edges between faces that meet at this angle or sharper.
	Angle	Becomes active when the *Break at Sharp Angles* option is activated and allows you to specify the sharp angle. The units of the sharp angle are specified in degrees only.

The break options provided are similar to those in surface repair. However, the break options criteria is only assessed on patch perimeters and not on patch interiors. For example, if a patch contains an interior sharp angle (as opposed to a sharp angle at the patch perimeter) then this sharp angle is ignored and the patch will always be completely grown into. Similarly, feature curves and non-manifold edges that are contained within a patch are also ignored. It is not possible to grow into only part of a patch - only whole patches are grown into, regardless of any break option features within that patch.

Smart Selection of Patches

Smart selection of patches lets you quickly grow a patch until it reaches a natural break point in the geometry. You can use smart selection to grow a patch until it reaches a natural break by double-clicking on a patch.

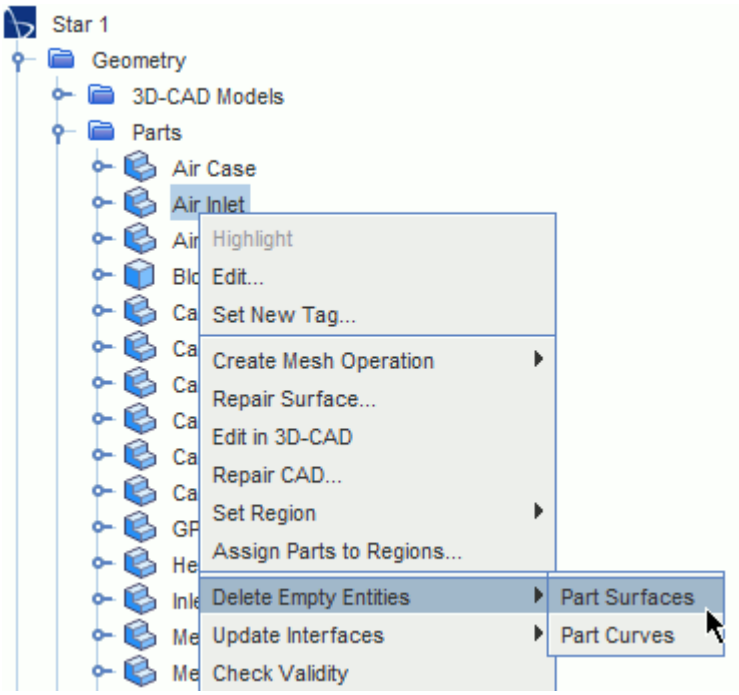
To combine a smart selection with any patches you have already selected, press and hold **<Ctrl>** before you perform the smart selection.

Deleting Empty Entities

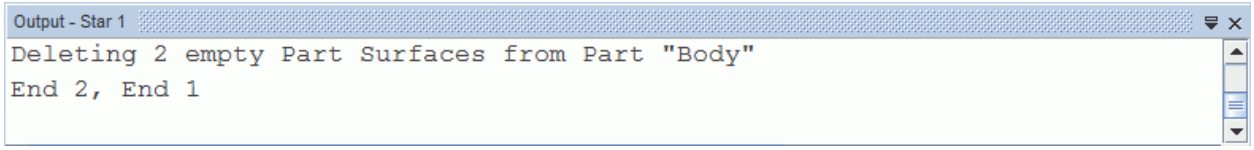
This option allows you to delete empty part surfaces and part curves. Empty part surfaces do not have face definitions in any of the descriptions in the simulation. Similarly, empty part curves do not have any feature edges in any of the descriptions in the simulation.

To delete empty entities from the simulation:

1. Expand the **Geometry** > **Parts** node.
2. Choose the entities you want to delete:
   - For empty part surfaces, right-click the relevant part node and select **Delete Empty Entities** > **Part Surfaces**.
   - For empty part curves, right-click the relevant part node and select **Delete Empty Entities** > **Part Curves**.



Information on the deletion process is displayed in the *Output* window.



Manipulating Part Curves

STAR-CCM+ allows leaf-level part curves to be manipulated in several different ways.

The following operations are possible:

- creating new part curves
- computing part curves
- combining part curves
- splitting non-contiguous part curves

In order to make part curves easier to search for, it is also possible to apply user tags to them.

Contents:

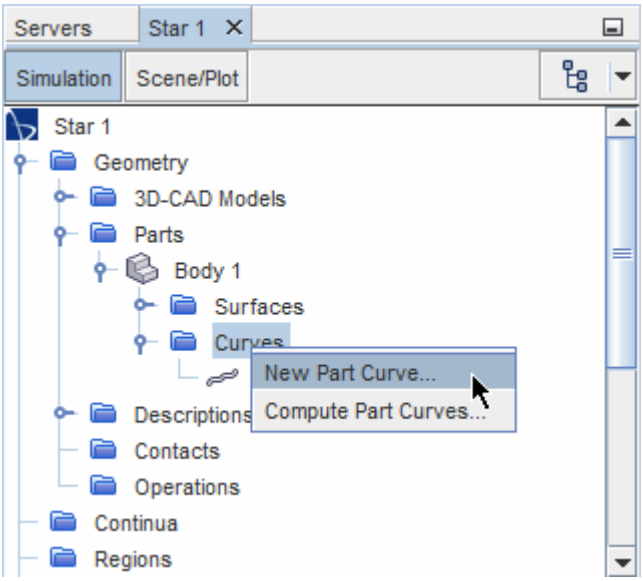
Creating Additional Part Curves
Computing Part Curves
Combining Part Curves
Splitting Non Contiguous Part Curves

Creating Additional Part Curves

You can create new, empty part curves to organise the feature curves. For example, if you split an existing part curve, you can assign it to an empty part curve.

To create a new part curve:

1. Right-click the **Curves** manager node and select **New Part Curve**.



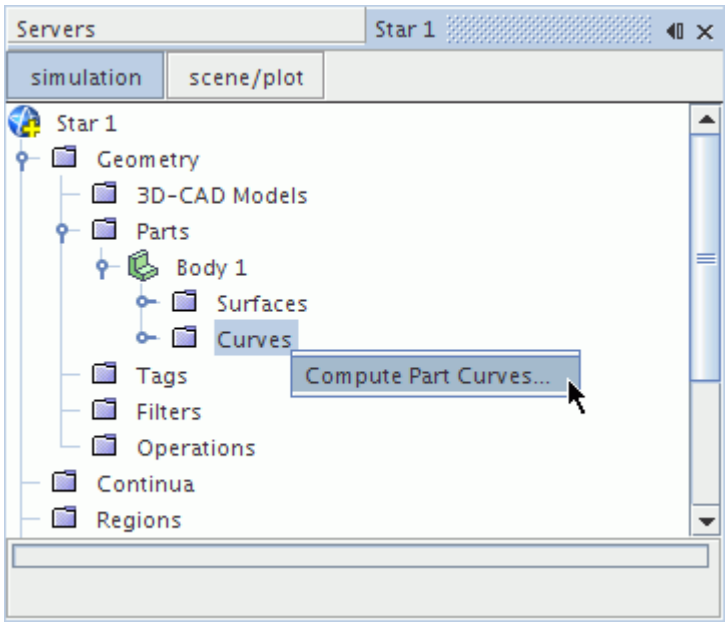
An empty part curve, **Part Curve**, is added to the **Curves** manager node. You can modify the part curve properties. See What is a Part Curve?

Computing Part Curves

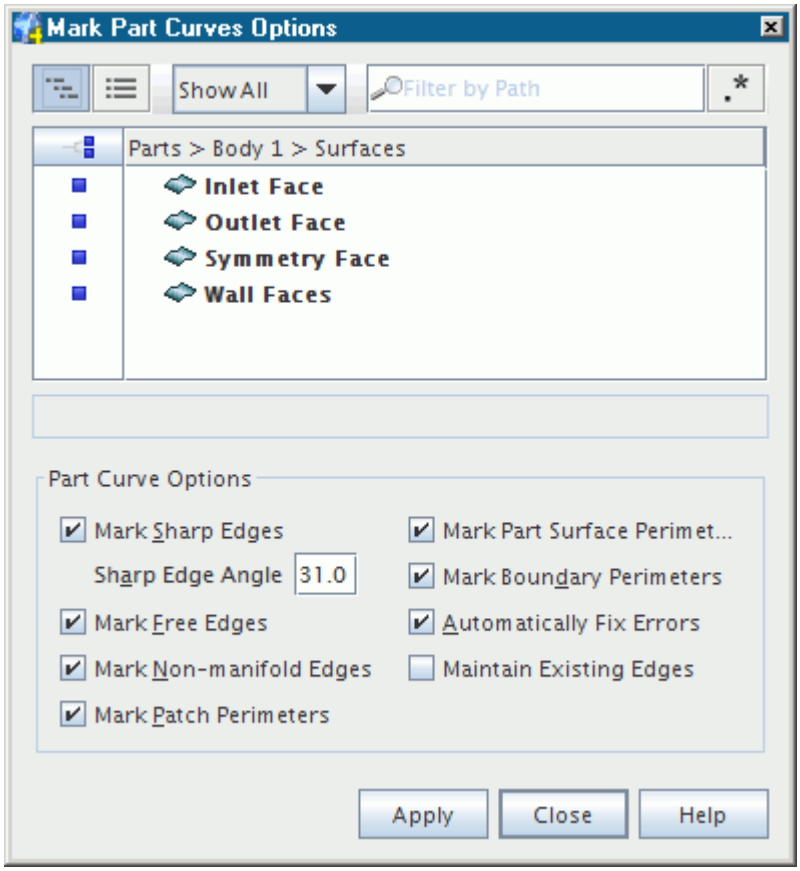
You can recompute part curves at any time. This feature lets you modify the curve definition using a number of different options, such as when more curves are needed.

To compute part curves:

1. Right-click the **Curves** manager node for the leaf-level part and select **Compute Part Curves...** from the pop-up menu:

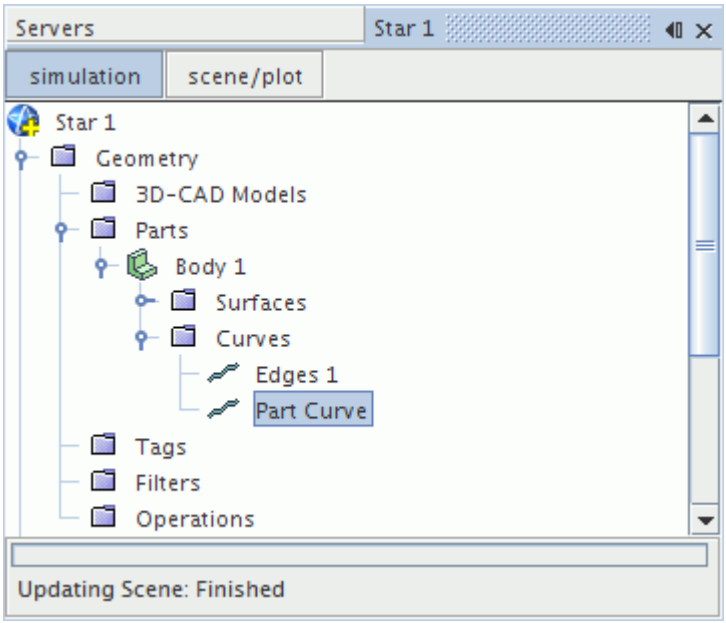


The *Mark Part Curves Options* dialog appears:



2. Select the part surfaces for which the curves are required.
3. Select the required Part Curve Options which are used for the curve computation.
4. Clicking **Apply** proceeds with the operation.
5. Once it is complete, click **Close**.

The new part curves are stored in a node that is called **Part Curve**:



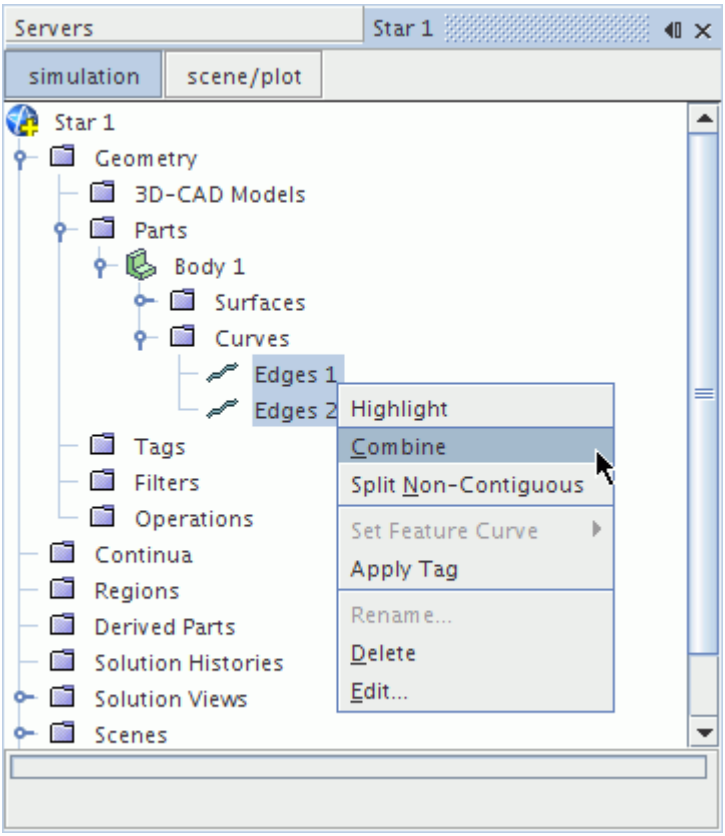
When computing new part curves, existing part curves in other nodes corresponding to the selected options are moved into the new part curve node.

Combining Part Curves

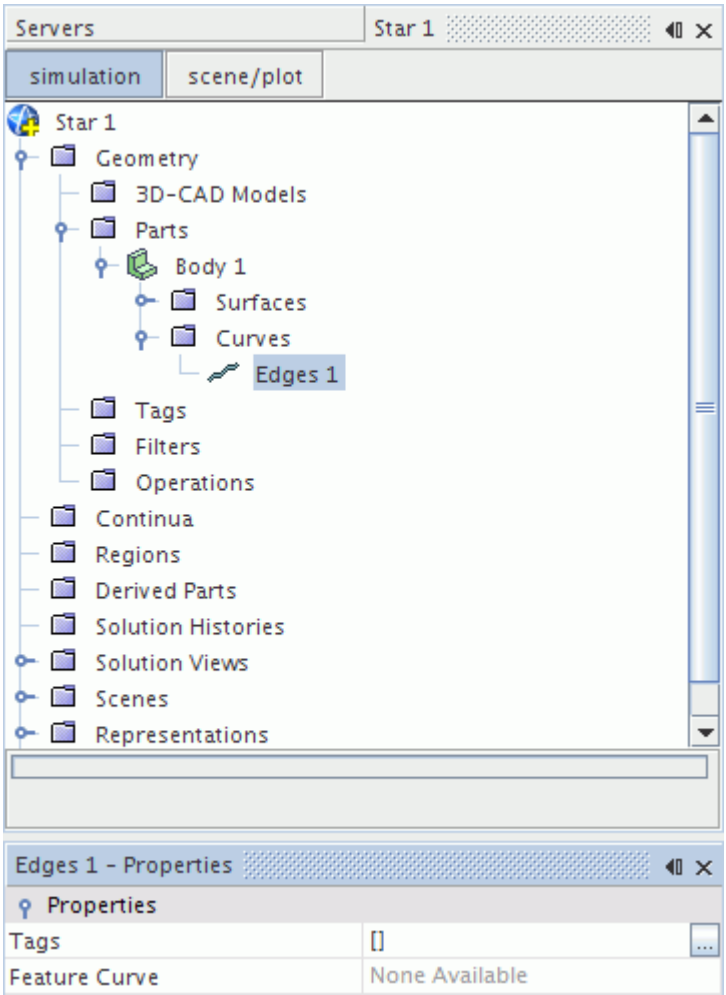
Two or more part curves can be combined into a single part curve if necessary.

To combine the curves:

1. Select two or more edge nodes belonging to the same part, right-click and select **Combine** from the pop-up menu:



The curve nodes are combined into the single node (the first selected):



Note the following:

- Tag properties relating to the secondary curve nodes are lost when they are different from the target (first selected) curve node.
- Feature curve assignments on the secondary curve nodes are lost when they are different from the target curve node.

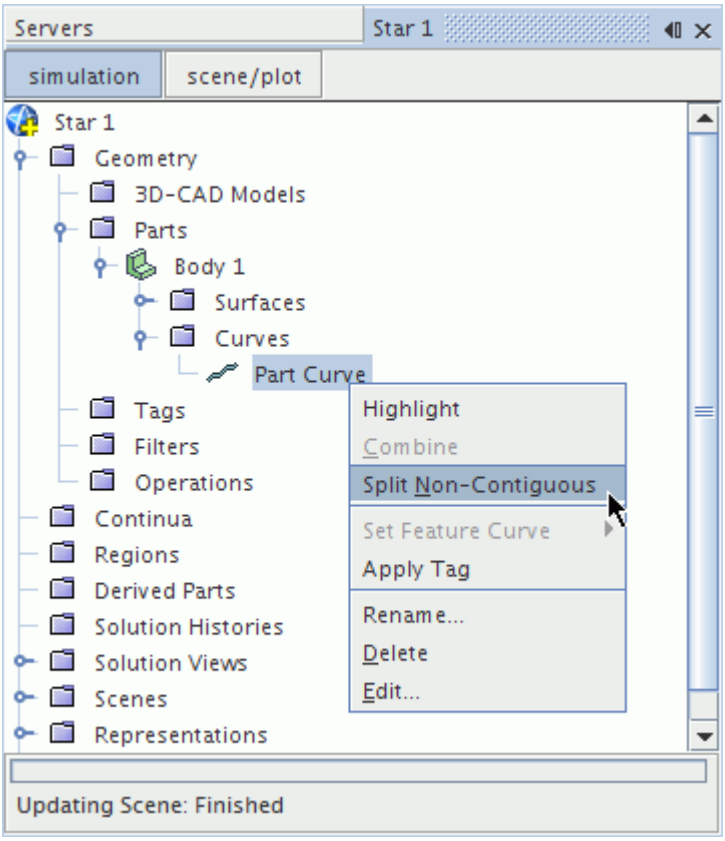
Splitting Non Contiguous Part Curves

A part can have one or more of its curves split based on the non-contiguous nature of the curves.

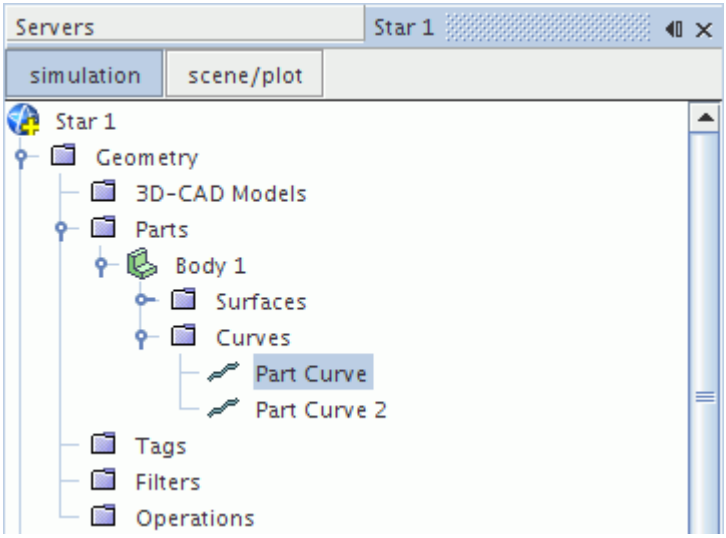
This feature splits up the selected curves depending on whether they are connected to their neighbors or not.

To split a curve:

1. Right-click the curve node and select **Split Non-Contiguous** from the pop-up menu:



Any non-contiguous curves are split to form new curve nodes, as shown below:



Any tags belonging to the original curve nodes are not copied to the split nodes. The new curves are also not assigned to any feature curves.

If the selected curves are contiguous, no action is taken.

Repairing Part Surfaces and Curves

Repairs can be made to leaf-level parts at any time before surface initialization. They can be required to close a surface or fix some other non-valid entity that the translation process created.

The tools available to do these repairs are ostensibly the same as the tools provided for surfaces within regions and boundaries under the Representations node. All of these tools are contained within the panels of the Surface Preparation Tool.

The main differences are:

- Surfaces in the display are colored as part-specific items.
- The Part Surface/Part Curve Edit tool replaces the Boundary/Feature Curve Edit tool.

If the Surface Preparation Tool is activated on a composite part, all of its child nodes are automatically selected for inclusion in the diagnostic process that precedes entry to repair mode.

Contents:

Activating the Surface Preparation Tool for Parts
Using Alternative Coloring for Part Surfaces
Moving Part Surfaces and Part Curves

See Also:

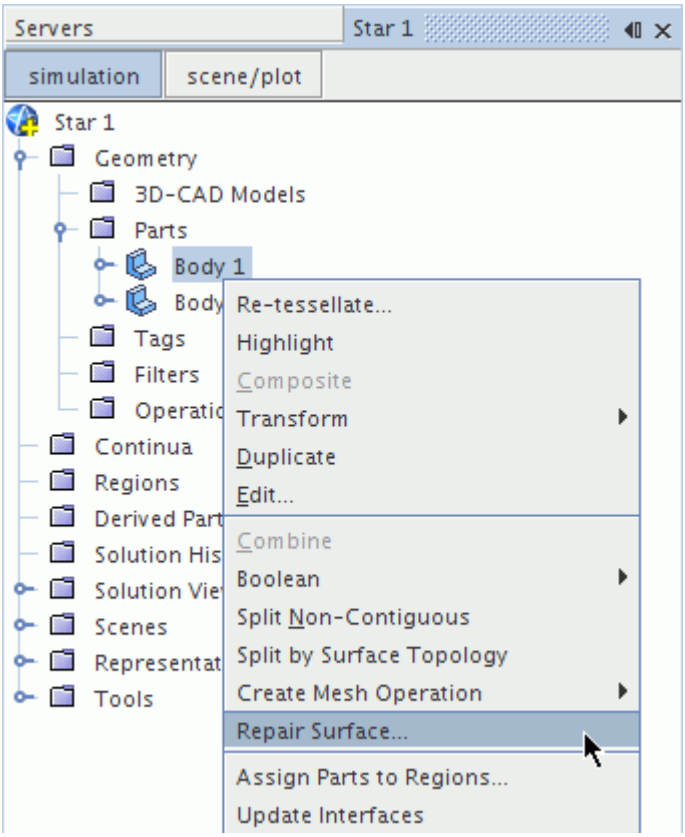
Understanding Geometry Parts

Activating the Surface Preparation Tool for Parts

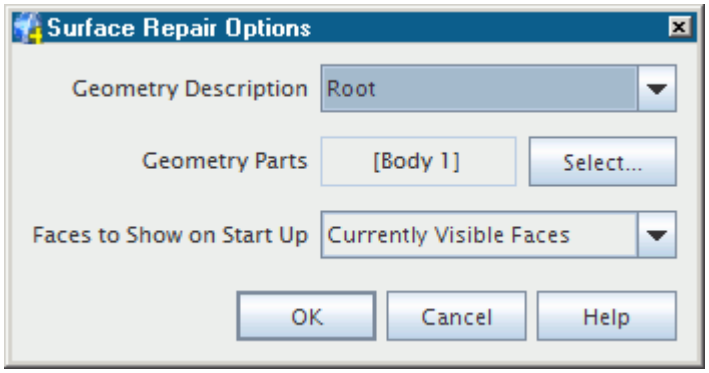
This section describes how to activate the Surface Preparation Tool.

To activate the Surface Preparation Tool:

1. Select the appropriate part nodes (which can include composite parts).
2. Right-click the selection and select **Repair Surface...** from the pop-up menu:



The *Surface Preparation Options* dialog appears:



3. Optional: To select specific parts using their name or other attributes such as tags or metadata, click the **Select** button to the right of **Geometry Parts**, then use the filter options within the *Select Objects* dialog.
4. Click **OK**.

The *edit* tab appears containing the surface preparation tool and its several modes of operation, identified as follows:

- surface diagnostics
- surface repair
- feature curve repair

- face offset tool
- connected surfaces edit
- merge/imprint tool
- face intersection tool

If the part is retessellated, any repairs that are made to part surfaces are lost. The lost repairs include the outcome of any imprint operations where more faces were created at the interface between two parts. Therefore, complete retessellation before making any manual changes with the surface repair tools.

Using Alternative Coloring for Part Surfaces

This section describes the options that are available for coloring part surfaces.

In addition to coloring surfaces by Default Color, Patch and CAD Association, the Display Control panel for surface repairs on parts provides the following alternative options:

- **Part Surface**: faces are colored according to the part surface to which they belong.
- **Part Boundary**: faces are colored according to the boundary to which they belong. This option is valid only if the part has been assigned to a region.
- **Part**: faces are colored according to the part to which they belong.

Moving Part Surfaces and Part Curves

This section describes how to move part surfaces and part curves.

You can use the *Part Surface/Part Curve Edit* panel to move faces into an existing or new part surface, and curves into an existing or new part curve. Details are given in the section, Surface and Curve Edit Controls.

Exporting Parts

This section describes how to export the data that is contained in the geometry parts tree.

Data that is contained in the geometry parts tree can be exported in two ways:

- By exporting individual leaf-level parts
- Using PLMXML export.

The PLMXML method can be used to export all the parts that are contained within a composite part, while retaining the groupings and hierarchy defined in the object tree. This method is useful if you have many parts to export and/or you wish to retain the organizational structure of the parts.

Contents:

Exporting a Part
Exporting Parts Using PLMXML

Exporting a Part

This section describes how to export leaf-level parts.

Leaf-level parts can be exported as surface mesh files using any of the following formats:

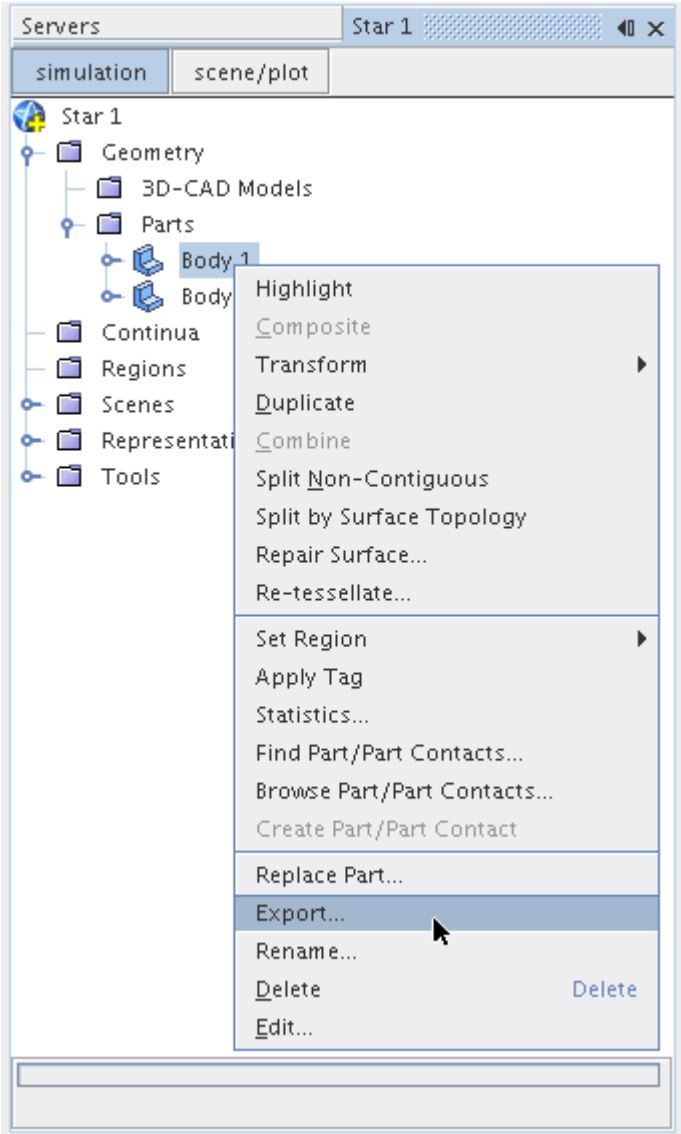
- pro-STAR database file (.dbs)
- Stereolithography (.stl)
- NASTRAN shells (.nas)
- PATRAN shells (.pat)
- pro-STAR shell input (.cel/.vrt/.inp)
- FELISA Front Surface (.fro)
- Medina Binary Input File (.bif)
- JT Open (.jt)

When exporting 3D-CAD, CAD, and shape parts, the current tessellation that is specified for the surface is used.

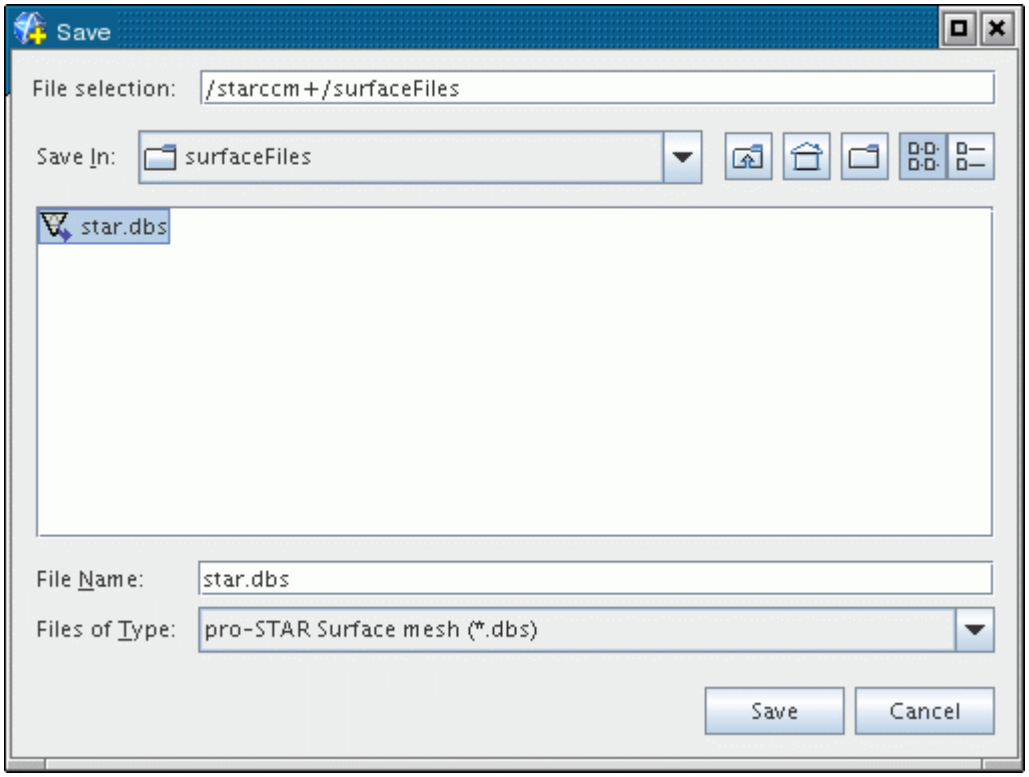
3D-CAD parts can also be exported as Parasolid Transmit files from within the 3D-CAD modeler.

To export a part:

1. Right-click the part node and select **Export...** from the pop-up menu:



The *Save* dialog appears. You can only save one part at a time using this method.



2. Specify a directory for the file.
3. In the *File Name* filed, enter a suitable name.
4. From the *Files of Type* field, select the file format.
5. Click **Save**.

Exporting Parts Using PLMXML

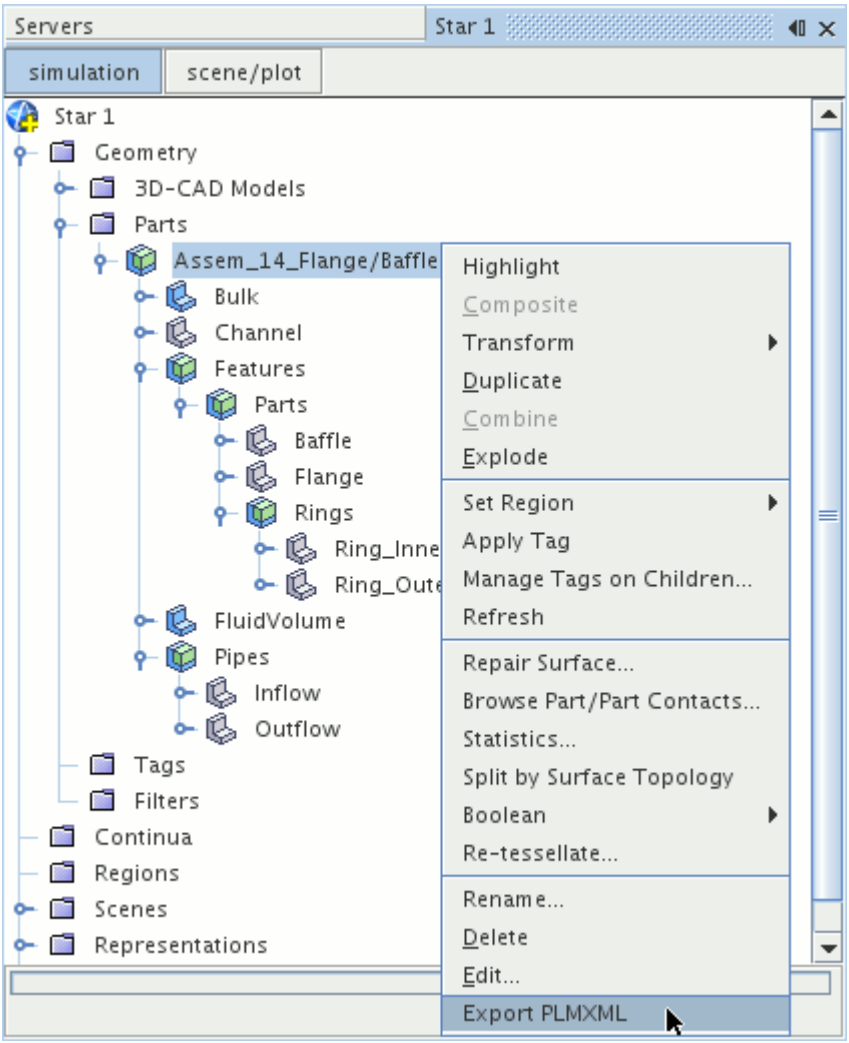
The PLMXML export option allows you to export large numbers of parts in a single operation.

The PLMXML export option creates a `.plmxml` file to retain information on how the parts are organized in the tree. When the PLMXML file is imported into another simulation, the parts are automatically imported and organized according to the structure in the PLMXML file.

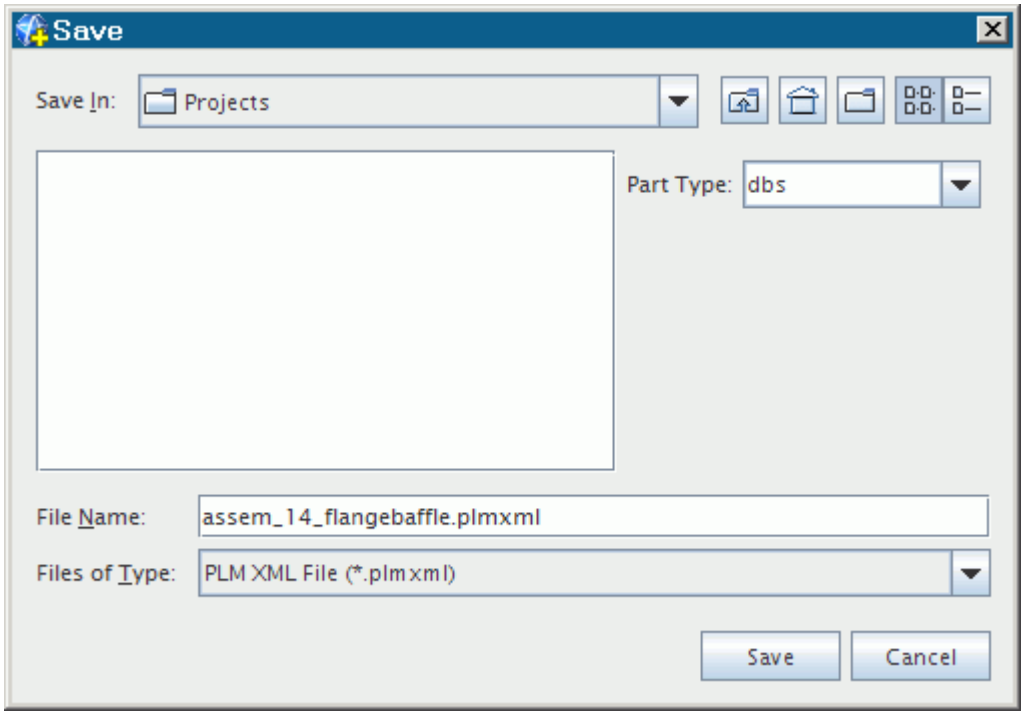
Before performing PLMXML export, ensure that the geometry parts contained in the tree are organized as necessary. The PLMXML export is only available for composite files. You can export a single composite part or multiple composite parts to a single PLMXML file.

To export the parts:

1. Right-click the composite node and select **Export PLMXML**.



The *Save* dialog appears.



2. From the **Save In** drop-down menu, specify the save directory.
3. From the **Part Type** drop-down menu, select a desired file format for the export:
   - Database (.dbs)
   - Stereolithography (.stl)
   - Nastran (.nas)
   - Patran (.pat)
   - Shell input file (.inp)
   - FELISA Front Surface (.fro)
   - Medina Binary Input (.bif)
   - JT Open (.jt)
4. In the *File Name* field, enter a suitable name.
5. Click **Save**.

All leaf-level parts are exported into a single file of the selected format. You can monitor the export progress in the *Output* window. The `.plmxml` file is also created containing the relevant structure, names, and file references for the exported parts.

To import the parts into another simulation, follow the procedure for importing XML files.

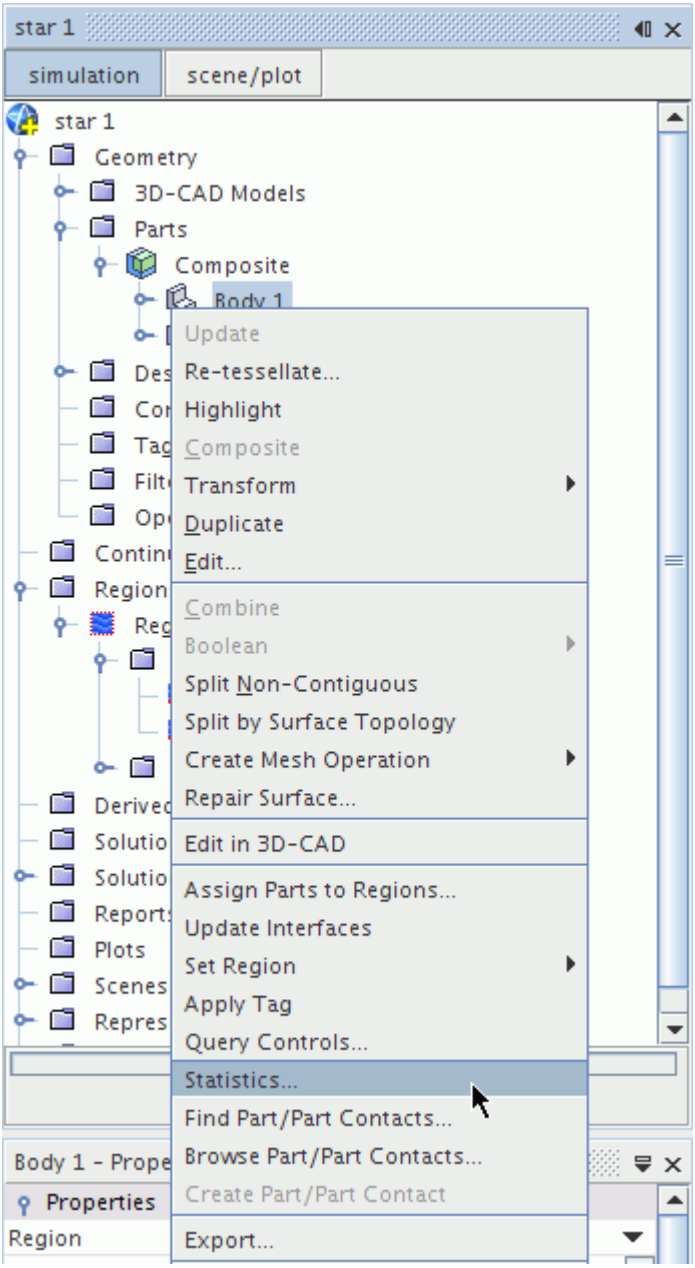
Note: You cannot import `.xml` and `.bif` files in versions of STAR-CCM+ that are running on Windows operating systems.

Getting the Face Count of a Part

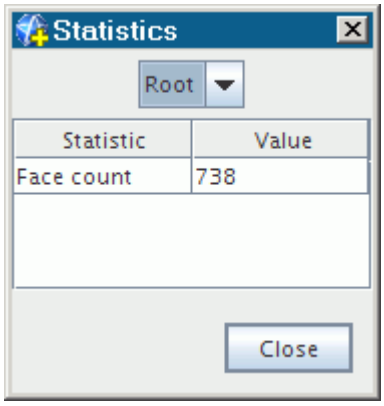
This section describes how to obtain the number of faces of a geometry part.

To obtain the number of faces of a geometry part:

1. Right-click on that part and select **Statistics**.



The *Statistics* dialog appears with a *Face count* property.



2. To check the face count for a particular description, use the drop-down list at the top of the dialog.

If the selected description does not exist for that part, the face count is zero.

This same feature works for part surfaces, part curves, and composite parts.

You can also determine the surface area of a part using the Area field function. See Volume Meshing Field Functions Reference.

Editing Imported Parts in 3D CAD

This section describes how to modify your imported geometry parts using 3D-CAD.

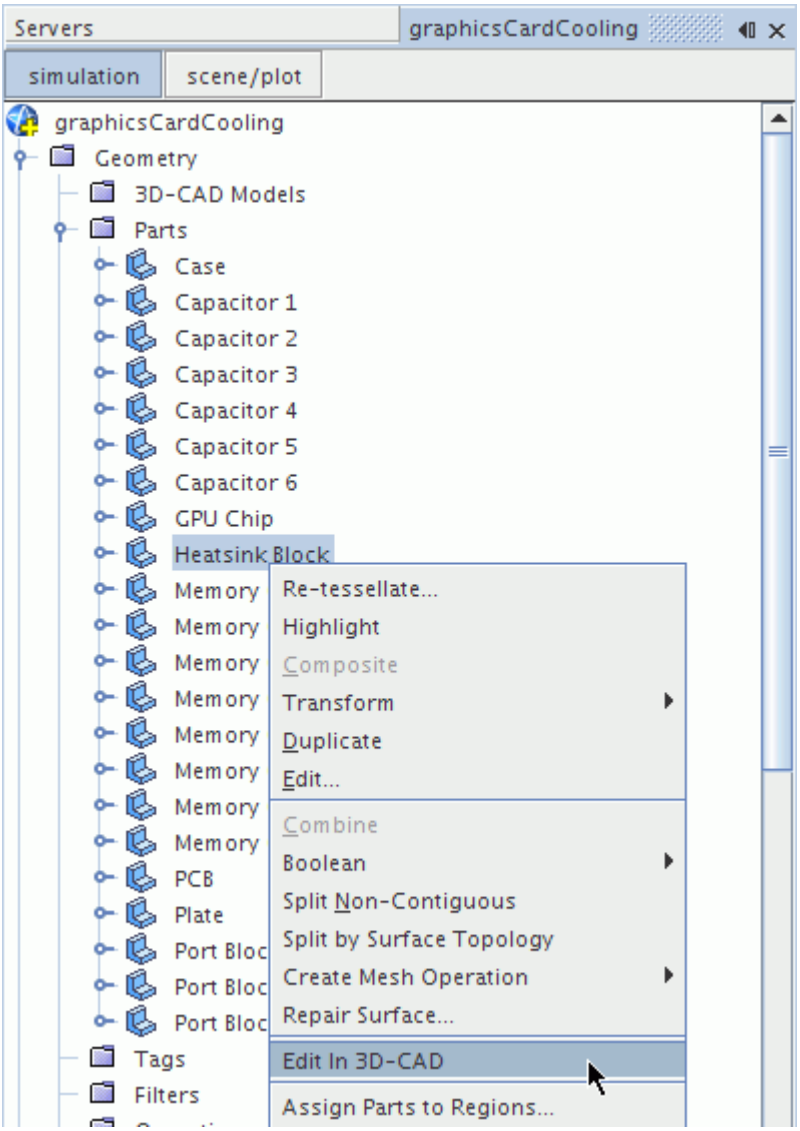
You can modify your imported geometry parts in STAR-CCM+ using 3D-CAD. This feature is useful if you do not have access to a commercial CAD package and you want to modify the CAD parts. You can carry out any 3D-CAD operation in context with your geometry. You can also rename faces on your geometry as an alternative to splitting surfaces by patch.

Note: This feature only supports parts that contain CAD data (such as CAD parts and 3D-CAD parts). Shape parts (such as Boxes, Spheres, and Cylinders) and mesh parts are not supported as they contain no CAD data.

Caution: CAD parts that you import into 3D-CAD are converted to 3D-CAD parts. This process is irreversible. This operation does not affect your original CAD geometry files.

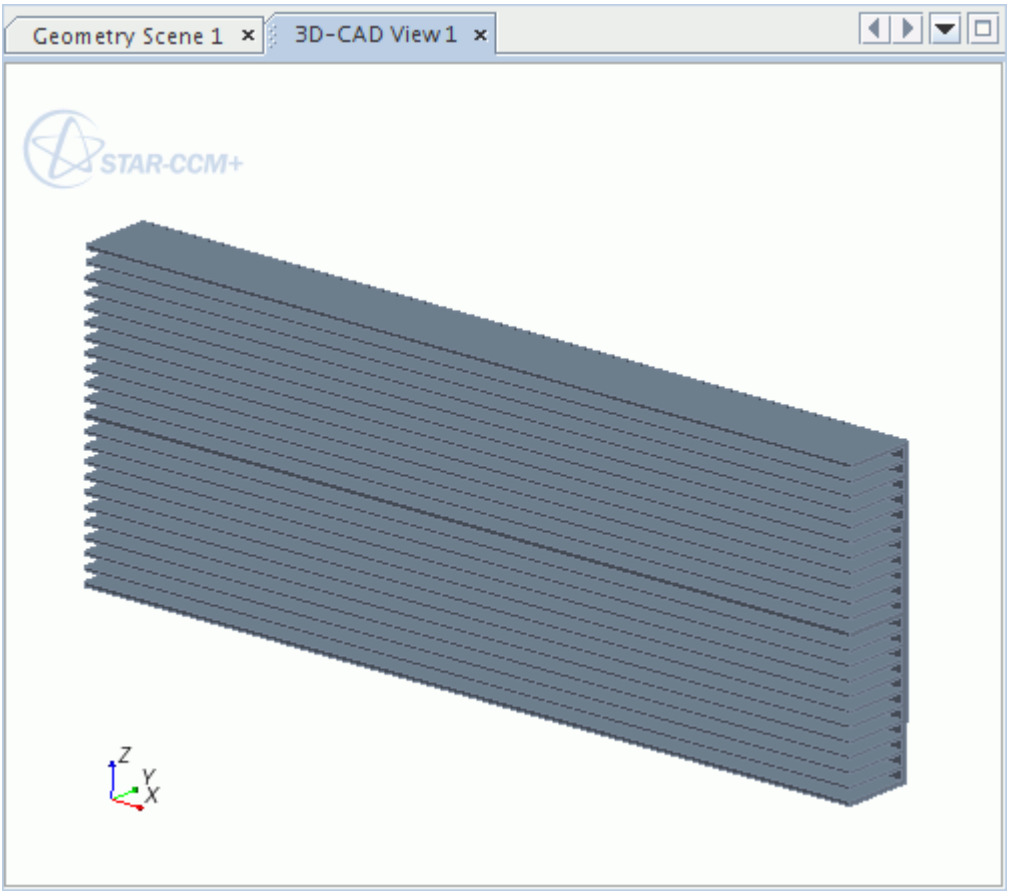
To edit a geometry part in 3D-CAD:

1. Right-click the desired part in the **Geometry** > **Parts** node and select **Edit in 3D-CAD**.

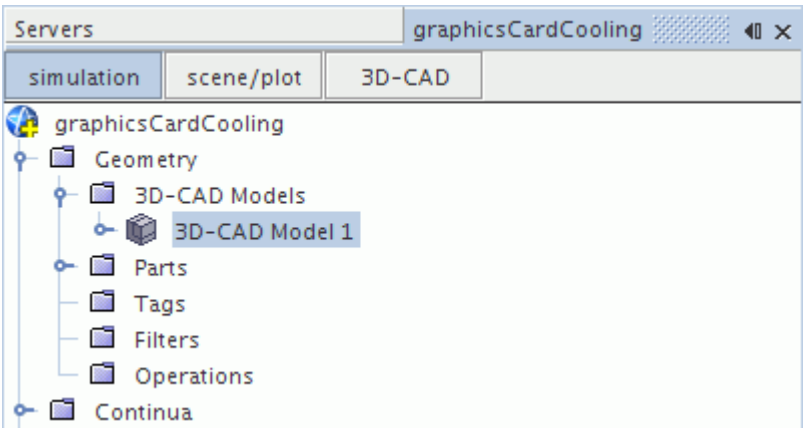


Note: You can only edit leaf-level parts in 3D-CAD.

STAR-CCM+ launches 3D-CAD and transfers your geometry over.

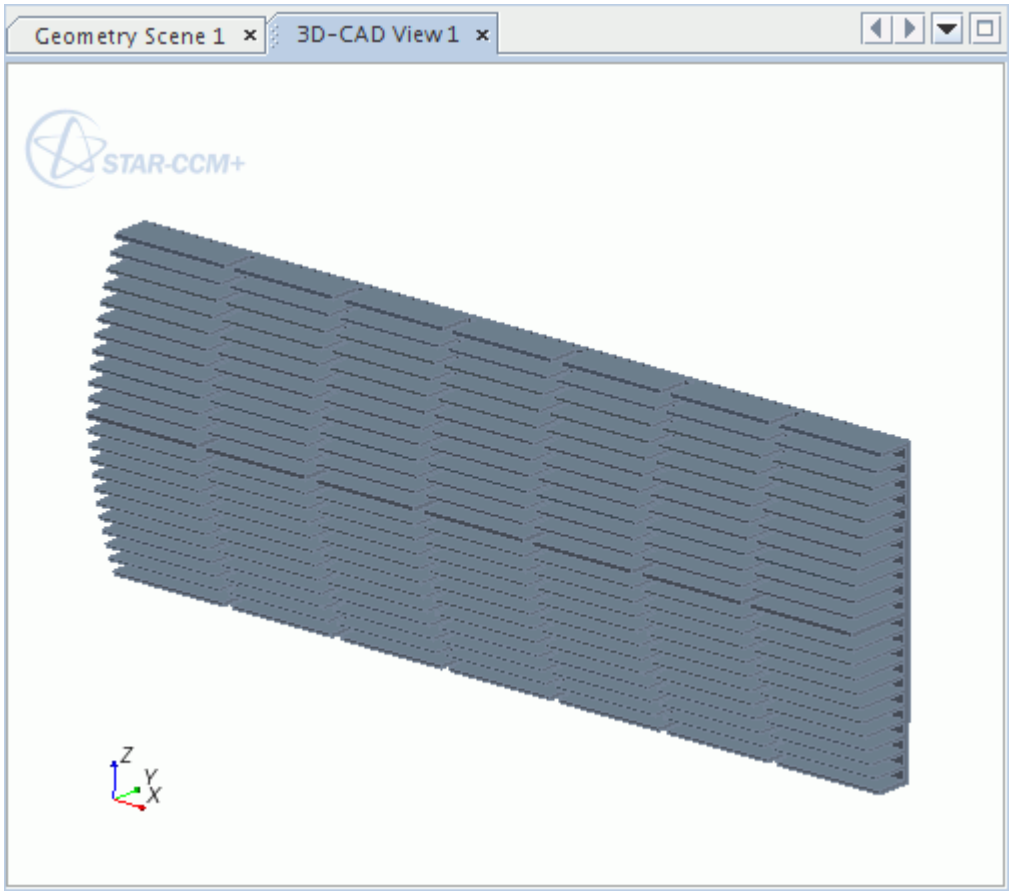


A 3D-CAD model is created for parts that you edit in 3D-CAD.



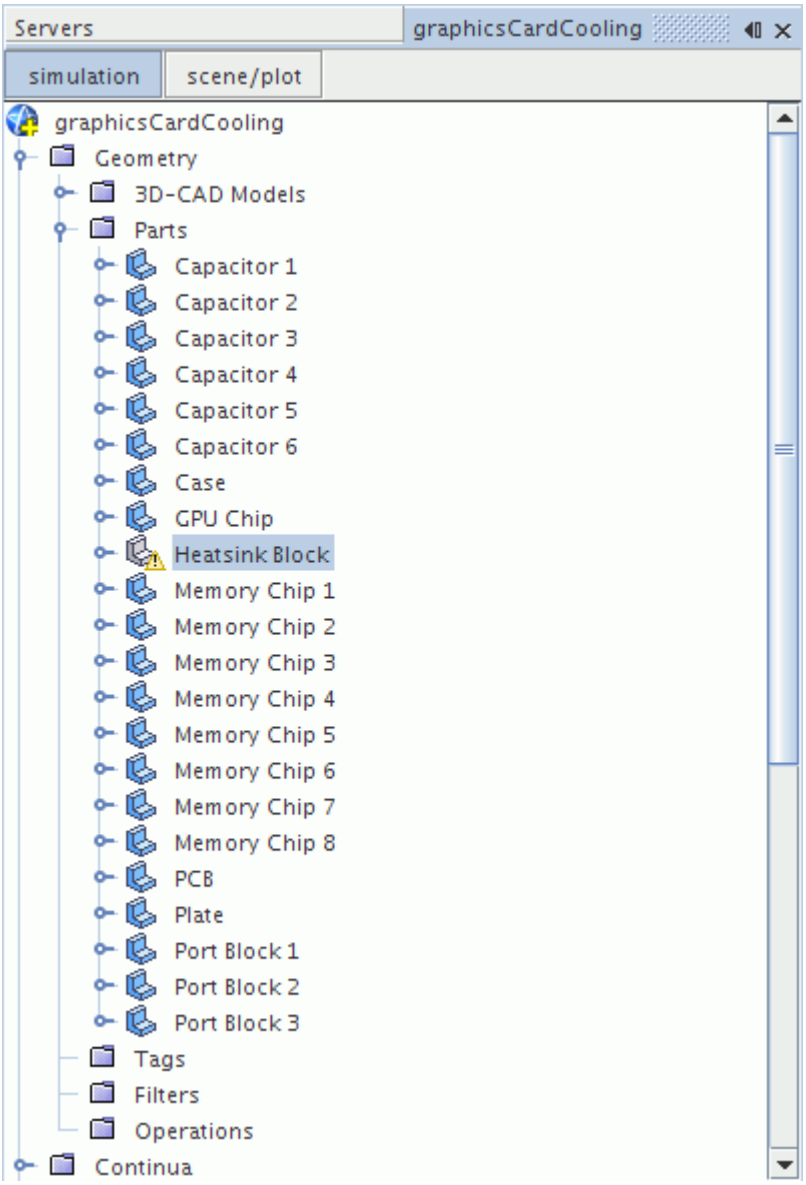
2. In 3D-CAD, perform the modifications that you want.

   In the example, to increase the surface area in contact with the cooling, the height of the cooling fins is modified and more vertical cooling channels are added.



3. Click **Close 3D-CAD** when done.

   Modified parts have new icons to show that they have been modified and require an update.



4. Right-click the modified parts and select **Update**.

Note: Once you edit a part, you cannot select to edit it with any other parts. You can only edit one 3D-CAD part at a time.

Part Operations Reference Page

This section contains a list of available operations that appear when right-clicking a part.

Each operation contains a short description and a link to the relevant section in the User Guide.

Update	If the selected part has been modified in 3D-CAD after it has been created, it will be necessary to update it to reflect the changes. This operation is only available for parts that are created in 3D-CAD.

Highlight	The selected part becomes highlighted in the *Graphics* window.	
Edit...	Opens the table editor, which combines the functionality of the simulation tree and *Properties* window. It displays the selected part, all of its child nodes, and their properties in a single dialog.	
Create Mesh Operation	Mesh operations provide a way to define certain actions on geometry parts once, and to repeat those actions easily whenever their input parts or properties are changed. The result of a mesh operation is a new geometry part, but the input parts are not removed in the process; they remain intact as before. The following operations are available:	
	Automated Mesh	Generates surface and volume meshes for selected input parts. This operation manages all of the meshers, default controls, and custom controls.
	Surface Wrapper	The surface wrapper can be used to provide a closed, manifold, non-intersecting surface when starting from poor quality CAD data.
	Badge for 2D Meshing	This operation is used to identify part surfaces from the input part that lie on the Z=0 plane and on which a 2D mesh is required.
	Automated Mesh (2D)	This operation generates a 2D mesh using the mesher that you choose. Only parts that are badged for 2D are available for selection as input parts for this operation.
	Directed Mesh	Allows you to generate a high-quality structured mesh on CAD geometries in STAR-CCM+.
	Extract Volume	This operation finds enclosed volumes within the selected part and create a part to fill the volume. This operation is similar to the split by surface topology action. The difference is that the extract volume mesh operation does not remove the input parts.
	Fill Holes	This operation can be used to close any arbitrary shaped hole using a planar surface. This option is similar to the hole filling tool available for feature curves.

	Boolean Unite Parts	This operation performs a boolean unite on all selected parts that are in contact with each other. Bodies that overlap can also be united. This operation is similar to the boolean unite action that can be performed on geometry parts. The difference is that this option creates a **Geometry** > **Operations** > **Unite** node which provides greater control when using the Parts Based Meshing (PBM) pipeline.
	Boolean Intersect Parts	This operation performs a boolean intersect on the selected parts. This operation is similar to the intersect parts action that can be performed manually on geometry parts. The difference is that this option creates a **Geometry** > **Operations** > **Intersect** node which provides greater control when using the Parts Based Meshing (PBM) pipeline.
	Boolean Subtract Parts	In a subtract operation, the volume of one or more geometry parts is removed from the target geometry part. This operation is similar to the boolean subtract action that can be performed manually on geometry parts. The difference is that this option creates a **Geometry** > **Operations** > **Subtract** node which provides greater control when using the Parts Based Meshing (PBM) pipeline.
	Boolean Imprint Parts	Merges coincident surfaces from the selected parts together. This operation is similar to the boolean imprint action that can be performed manually on geometry parts. The difference is that this option creates a **Geometry** > **Operations** > **Imprint** node which provides greater control when

		using the Parts Based Meshing (PBM) pipeline.
Repair Surface...	Activates the Surface Preparation tool where the part can be cleaned up in preparation for CFD analysis.	
Set Region	Allows you to add the selected part to an existing region. The available regions appear in the drop-down list.	
Assign Parts to Regions...	Launches the *Assign Parts to Regions* dialog, where the selected part can be added to a new or existing region. The region, boundaries, and feature curves that result from this operation can all be controlled in the dialog.	
Delete Empty Entities	Deletes empty part surfaces or part curves. Empty part surfaces do not have face definitions in any of the descriptions in the simulation. Similarly, empty part curves do not have any feature edges in any of the descriptions in the simulation.	
Update Interfaces	Checks for any changes to the part interfaces on the selected part and update them if necessary. Invalid or obsolete interfaces are removed.	
Apply Tag	Applies a created tag to the selected part. You can apply more than one tag to each part, whether a leaf-level or composite, if necessary.	
Statistics...	Displays the number of cell faces, in the initial mesh representation, that make up the selected part.	
Query Controls	Displays as a list in the Output window, all of the meshing custom control settings that are applied to the selected part – and the name of the mesh operation to which the custom control belongs.	
Composite	The selected parts are added to a composite (assembly) part. Useful for grouping parts together and for organizing the **Parts** manager node.	
Duplicate	Produces an identical copy of the selected part with the same properties as the original, other than region and contacts specification, which are left null.	
Combine	Combines the selected parts to form a single mesh part. The first selected part is the one that is used as the base for all the other parts. All part surfaces and curves belonging to each part are grouped in the base part. This operation is irreversible.	
Explode	Explodes the composite part, returning the child parts to their original level in the **Parts** manager node. This operation is only available for composite parts.	
Transform	The following Transformation operations are available; each operation launches its respective dialog:	
	Coordinate Transform...	Maps the selected part from the current coordinate system, to a coordinate system of your choice. By using this operation, you can

		translate and rotate a part at the same time while maintaining the exact shape of the part.
	Scale...	Allows you to modify the size of the selected part. Scaling can take place in any or all of the three coordinate directions and can be used to increase or decrease the size of a part. Scaling a part is useful if it uses units different to other parts you have already imported.
	Translate...	Allows you to move the selected part in any of the three coordinate directions. Translating is useful if different parts are built in different coordinate systems.
	Rotate...	Allows you to rotate the selected part) about a specified axis vector.
	Reflect...	Allows you to mirror the selected part in a specified axis. This option can be useful when the original part is a half model of the geometry. You can create the full model by first duplicating the part and then reflecting the copy (or original).
Boolean	The following Boolean operations are available:	
	Unite Parts	Joins all the selected parts into one single part, the surface of which is formed from segments of the constituent part surfaces. This operation differs from the Combine operation, which combines the selected parts into one part, but retains the part surfaces as separate, whole surfaces. To carry out a unite operation successfully, the selected parts must be in contact with each other, or overlap. If the parts do not meet this criterion, the part that the Boolean operation creates contains no surface data.

	Intersect Parts	Finds the common volume that the selected parts share and creates a single part that is based on this volume. The surface of the part is formed from segments of the constituent part surfaces. To carry out an intersect operation successfully, all the selected parts must share a common volume. If the parts do not share a common volume, the part that the Boolean operation creates contains no surface data.
	Subtract Parts	Subtracts the defined shape from the volume of a target part. Any number of parts can be used to define the shape to subtract. The surface of the resulting part is formed from segments of the constituent part surfaces.
	Imprint Parts	Merges coincident surfaces from the selected parts together. When coincident surfaces are not identical (in shape or size), the edges of the smaller surface are used to split the larger surface. By imprinting parts, you can successfully generate interfaces between multiple connected parts. The interfaces are essential for conformal meshing.
Split by Surface Topology	Can be used to split and combine parts that can be defined as distinct geometrical regions. When this technique is used for multiple parts, STAR-CCM+ internally combines all the parts and treats them as a single part. The option is available to combine parts with shared part surfaces.	
Split Non-Contiguous	Splits non-contiguous mesh parts (that is, parts that have faces that are not connected to other faces in the part). Each part can be split into its individual component parts.	
Replace Part...	This operation replaces a 3D-CAD, CAD, or mesh part in order to provide new geometrical input for the meshing pipeline (without affecting any of the other data input required to generate a mesh). Only one part node can be replaced at a time. The operation does not apply to shape parts.	

Repair CAD...	Activates the Geometry Repair tool where the CAD geometry can be diagnosed for Parasolid invalidities. The tool can attempt to repair any issues it finds automatically. This tool is only compatible with parts that contain CAD data.
Edit in 3D-CAD	Transfers the CAD part to 3D-CAD where you can work with it. Only parts with CAD data can be transferred to 3D-CAD.
Check Validity	Assesses the geometric and topological validity of the selected parts.
Re-tessellate...	Modifies the tessellation of the selected part. The surfaces of a geometry part are represented using triangles which come from a process that is known as tessellation. Selecting this operation launches the *Tessellation options* dialog.
Export...	Exports the selected part as a surface mesh file. The supported formats are listed below: • pro-STAR database file (.dbs) • Stereolithography (.stl) • NASTRAN shells (.nas) • PATRAN shells (.pat) • pro-STAR shell input (.cel/.vrt/.inp) • FELISA Front Surface (.fro)
Copy	Copies the selected part into the clipboard.
Paste	Pastes the part from the clipboard into the selected node in the simulation.
Delete	Deletes the selected part from the **Parts** manager node.
Rename...	Allows you to rename the selected part.

Part Contacts

Part contacts define the connection between two part surfaces. The meshers aim to generate a conformal mesh at the contact, which then becomes an interface in the physics region.

When you use contacts with the surface remesher it aims to generate a conformal match between the surface meshes. If the mesher cannot generate a conformal surface mesh, it terminates the meshing process and produces an error message that describes the problem.

A conformal volume mesh between contacts produces a high-quality discretization for the analysis. However, you can produce valid results with a non-conformal mesh, as STAR-CCM+ can still accurately transfer data between the cells.

For the trimmed mesher, a conformal volume mesh is not always possible, as the surface mesh does not directly define the volume mesh cells.

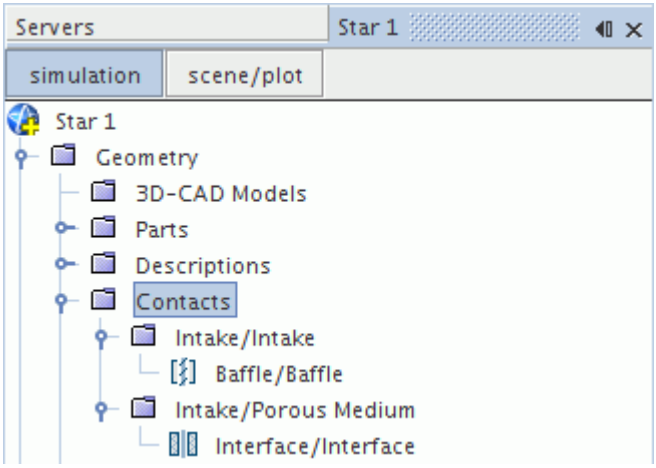
For all other meshers, a conformal, remeshed surface produces a conformal volume mesh. This conformality is inherent as the surface mesh directly defines the shape of the volume mesh cells.

However, a contact between a polyhedral and trimmed mesh is never conformal due to the difference in shape between the two types of cells.

The following types of mesh contacts are available:

- In-Place Contact
- Periodic Contact
- Baffle Contact

The **Contacts** node holds all mesh operations that are defined in the simulation.



Contents:

Part Contacts Properties and Context Menus
Creating Part Contacts
In-Place Contact

Periodic Contact
Baffle Contact

Part Contacts Properties and Context Menus

The **Contacts** manager node and the part contact subnodes provide several properties and right-click actions that allow you to check and edit the connections between parts.

Contacts Manager

The following actions are available when you right-click the **Contacts** manager node:

Context Menu	
Edit...	Opens the table editor, which combines the functionality of the simulation tree and *Properties* window. It displays the selected part, all of its child nodes, and their properties in a single dialog. See Using the Table Editor.
Find Part/Part Contacts...	Automatically finds part contacts between the selected parts (two or more parts) based on a specified tolerance value. This operation is useful in circumstances where several parts exist, some of which can be touching two or more other parts. See Finding Part Contacts Automatically.
Browse Part/Part Contacts...	Allows you to scan through the part contact data using the *Part/Part Contact Browser* dialog, and view the contacting parts in the *Graphics* window. This operation automatically updates the view to hide parts that are not part of a contact list. See Browsing Part Contact Data.
Create Part/Part Contact	Allows you to create part contacts manually between the selected parts (two or more parts). This operation is useful when the parts have surfaces that are in contact with one another and require an interface for the physical coupling once the mesh has been built. See Creating Part Contacts Manually.
Refresh	Updates the contacts node so that it reflects the latest changes to the simulation.

Part Contact Nodes

The following tables summarize the properties and right-click actions that are available for each part contact:

Properties	
Part 1	The read-only names of the parts whose surfaces are in contact.
Part 2	

Context Menu	
Highlight	Highlights all part surface contacts in a scene. This option is only available if the scene is open. See Highlighting Objects.
Delete	Removes the selected part contact. See Deleting Objects.

Creating Part Contacts

Use mesh contacts to define a connection between two part surfaces.

The meshers aim to generate a conformal mesh at the contact, which then becomes an interface in the physics region.

You can manually create the following types of part contacts:

Contents:

Creating Weak In-Place Contacts
Creating Baffle Contacts
Creating Periodic Contacts

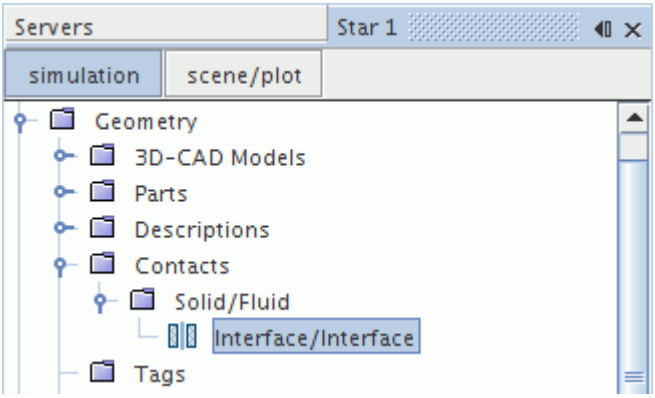
Creating Weak In-Place Contacts

Weak in-place contacts model part surfaces that are in contact with each other where the part contacts are created manually. Weak in-place contacts do not create a conformal mesh, but become an in-place, internal interface when assigning parts to regions.

To create a non-conformal in-place contact:

1. Multi-select the contacting part surfaces – which can be from one or more parts.
2. Right-click one of the selected part surfaces and select **Create weak in-place contact**.

   This action adds a part contact under the **Contacts** node.



   The resulting weak contact creates an internal interface which is used by the solver and not the mesh – therefore the mesh is non-conformal.

See Also:

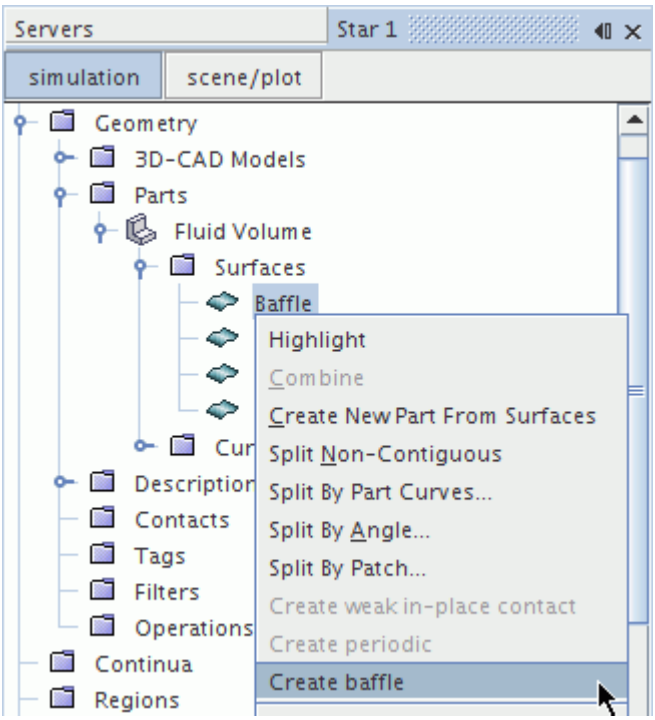
In-Place Contact

Creating Baffle Contacts

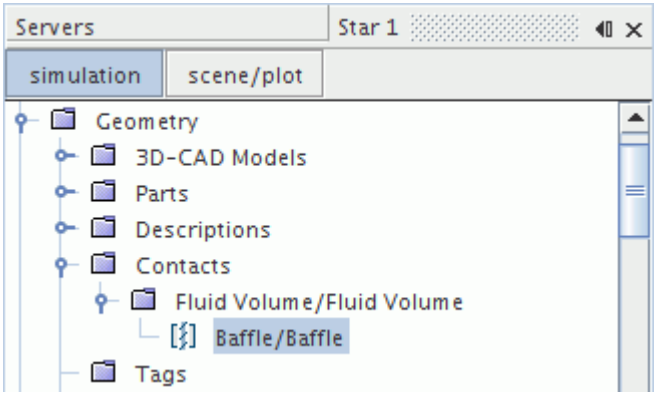
Baffle contacts model a zero-thickness surface that the meshers maintain in the surface and volume mesh. When assigning parts to regions, the baffle contact becomes an in-place, baffle interface.

To create a baffle contact:

1. Right-click a part surface and select **Create baffle**.



This action adds a part contact under the **Parts** manager, which is where you can change the baffle contact properties.



See Also:

Part Contacts

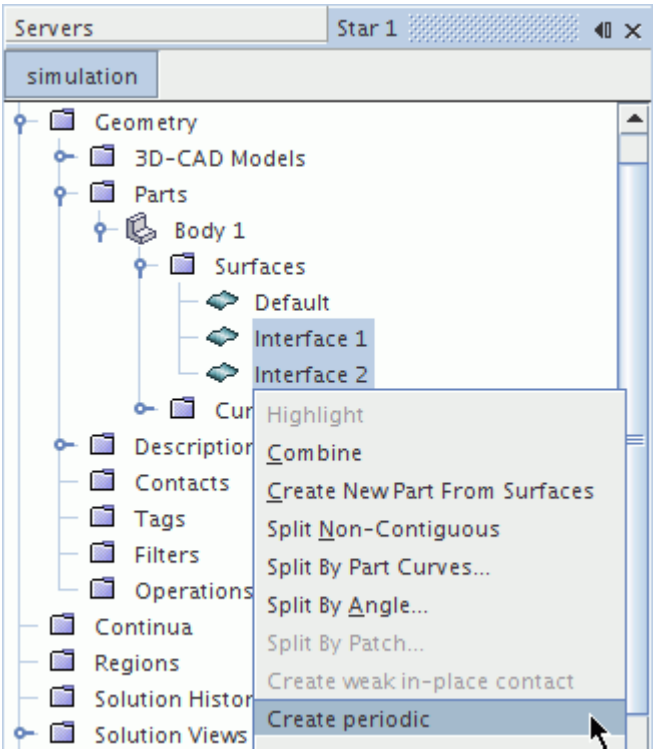
Baffle Contact
Baffle Contact Properties
Baffle Interface

Creating Periodic Contacts

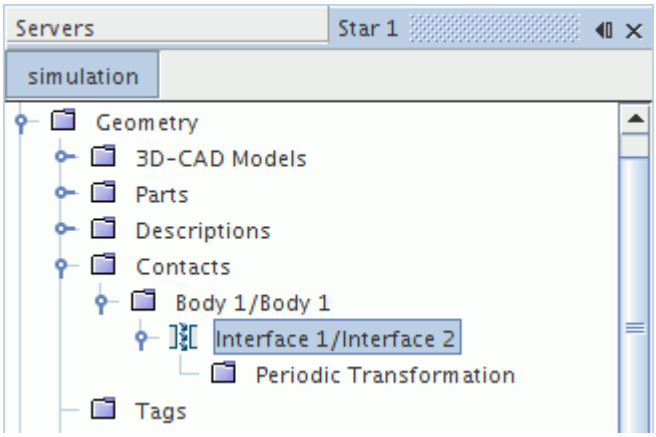
Periodic contacts model two part surfaces that are spatially detached, but a rotation and/or translation can make them coincident. When assigning parts to regions, the baffle contact becomes a periodic, internal interface.

To create a periodic contact:

1. Multi-select the periodic part surface.
2. Right-click one of the selected part surfaces and select **Create Periodic**.



This action adds a part contact under the **Parts** manager, which is where you can change the periodic contact properties.



See Also:

Part Contacts
Periodic Contact
Periodic Contact Properties
Periodic Interface Topology

In-Place Contact

An in-place contact is a direct connection between two part surfaces.

There are two types of in-place contacts: weak and strong. A strong in-place contact allows you to generate a conformal mesh between the two surfaces, whereas a weak in-place contact does not. Both types of in-place contact become interfaces when you assign the parts to regions. You can see whether an in-place contact is strong or weak by checking the *Conformality* property. See In-Place Contact Properties.

The following actions create strong in-place contacts:

- Import a surface mesh.
  - Make sure you activate the **Create Part Contact from Coincident Entities** checkbox.
- Create a geometry part from a 3D-CAD model.
  - Make sure you activate the **Create Part Contact from Coincident Entities** checkbox.
- Executing the extract volume or imprint mesh operations.

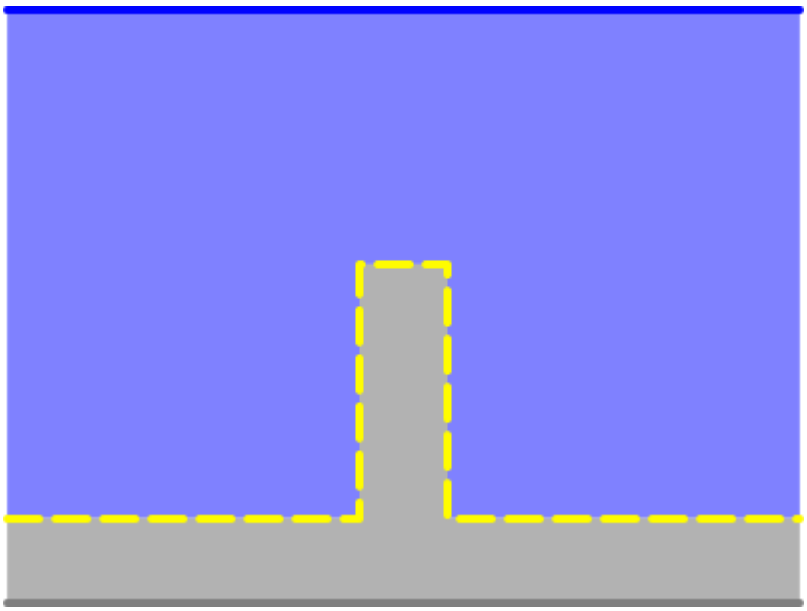
The following actions create weak in-place contacts:

- Manually creating or finding part/part contacts between parts.
- Manually creating a part contact between two surfaces.
- Executing the surface wrapper operation.

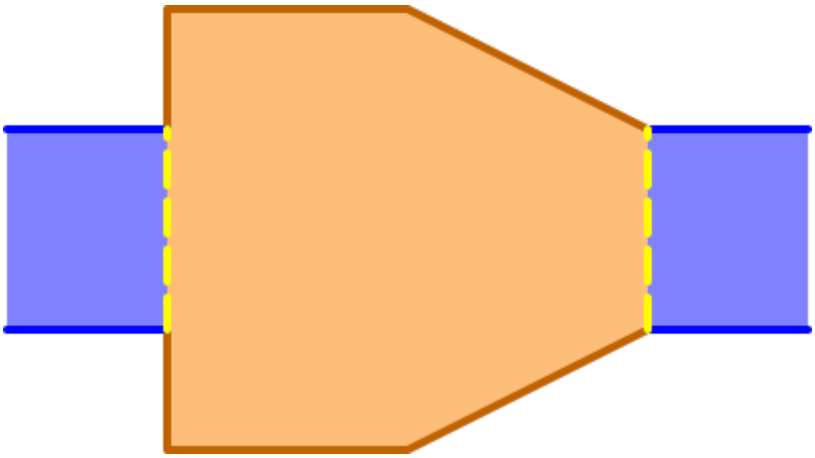
Examples of In-Place Contacts

The following examples show two types of in-place contacts. The yellow, dashed line highlights surfaces where the mesher aims to generate a conformal mesh.

This first diagram illustrates a contact between two part surfaces that match exactly.



This second diagram illustrates an in-place contact between two part surfaces that have different areas. In this case, the contact applies to both surfaces, but the mesher only aims to produce a conformal mesh between the areas of the part surfaces that are coincident.



Contents:

In-Place Contact Properties

In-Place Contact Properties

The following table summarizes the properties that are available for in-place contacts.

Part Surface 1	The read-only names of the part surfaces that are in contact.

Part Surface 2	
Conformality	Read-only property that indicates whether a contact is conformal or non-conformal. If a strong contact exists in any description, the contact is considered conformal. If a weak contact exists in all descriptions, the contact is considered non-conformal.
Interface	Displays the interface associated with the part surface contact when the parts are assigned to regions.
Tags	Applies a user-defined tag so that you can organize mesh contacts.
Metadata	Specifies the information that you can use to filter the mesh contacts, which are based on properties that are unrelated to the STAR-CCM+ simulation.
Index	Index reference for the part surface contact (read only). This index is available through the **Part Surface Contact Index** field function.

See Also:

Part Contacts

What Is Metadata?

Periodic Contact

A periodic contact defines two part surfaces that are spatially detached, but a rotation and/or translation can make them coincident.

When you create a periodic contact, the surface that you select first is the master surface and the other surface is the slave. STAR-CCM+ calculates the rotation and/or translation as relative to the master surface.

Some geometry configurations require a conformal surface and volume mesh between regions. The periodic contact helps achieve a conformal mesh between the desired regions. However, using this type of contact can lead to unwanted interfaces being generated when you assign parts to regions as by default, periodic contacts generate periodic interfaces. In some cases, periodic contacts are only required to achieve a conformal mesh between regions and a periodic interface is not required. For such cases, you can activate the **Mesh Only Contact** option on individual contacts to stop the contact from generating an interface.

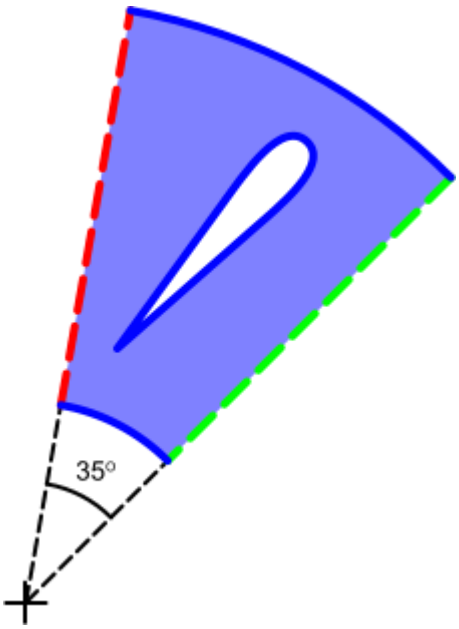
To make sure that the mesher can generate a conformal mesh, the size and shape of the periodic surfaces must be identical. If you are using a rotation and translation transformation, make sure that the rotation and translation vectors are parallel to each other. If these transformation vectors are not parallel, the resulting mesh may not be conformal. If the mesher cannot generate a conformal mesh, the meshers continue, but the *Output* window displays a warning message.

To make sure that the periodic surfaces maintain their original shape, they are excluded from the automatic surface repair operations. Similarly, any manual repairs that you make to the periodic surface after remeshing are not maintained during volume meshing.

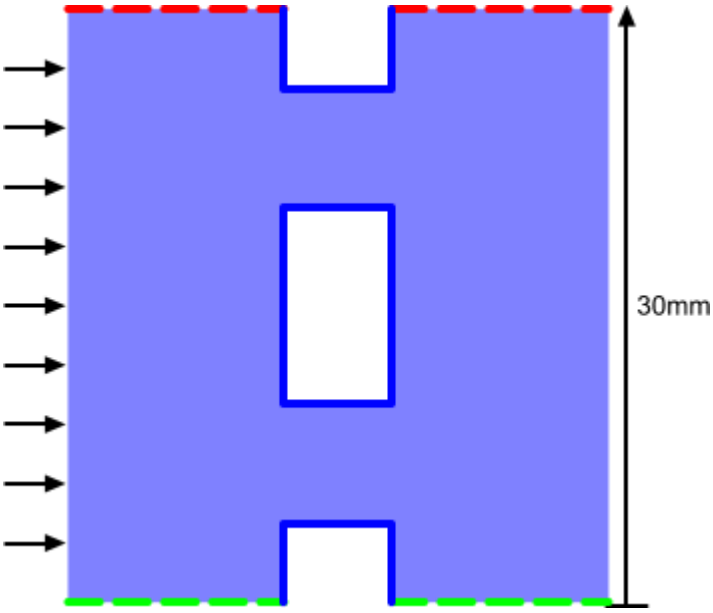
Examples of Periodic Contacts

The following examples show two types of periodic contacts. The green and red dashed lines represent the master and slave surfaces respectively.

The first diagram illustrates a rotational periodic contact.



The second diagram illustrates a translational periodic contact.



Contents:

Periodic Contact Properties

Periodic Contact Properties

The following table summarizes the properties that are available for periodic contacts.

Part Surface 1	The read-only names of the part surfaces that are in contact.
Part Surface 2	
Conformality	Read-only property that indicates whether a contact is conformal or non-conformal.
Tags	Applies a user-defined tag so that you can organize mesh contacts.
Interface	Displays the interface associated with the part surface contact when the parts are assigned to regions.
Metadata	Specifies the information that you can use to filter the mesh contacts, which are based on properties that are unrelated to the STAR-CCM+ simulation.
Index	Index reference for the part surface contact (read only). This index is available through the **Part Surface Contact Index** field function.
Surface 1	Specifies the master surface in the periodic contact, that is, the periodic transform is relative to this surface (read-only). This surface is the one that you selected first when creating the mesh contact.
Surface 2	Specifies the slave surface in the periodic contact (read-only). This surface is the one that you selected second when creating the mesh contact.
Mesh Only Contact	Controls whether the contact becomes associated with an interface or not, when you assign parts to a region. When activated, STAR-CCM+ does not create a corresponding interface in the regions definition. If you originally had this option deactivated and created an interface from this contact and later activate this option, the contact will be removed on a subsequent update.

Periodic Transformation Properties

The following table summarizes the properties that are available for the **Periodic Transformation** node.

Periodicity	**Rotational**	Specifies that the periodic transformation only requires a rotation angle to match *Surface 1* to *Surface 2*. Therefore the translation is not calculated.
	Rotational+Translational	Specifies that the periodic transformation requires a rotation angle and translation to match *Surface 1* to *Surface 2*. To achieve a conformal mesh, the rotation and

		translation vectors must be parallel to each other.
	Translation	Specifies that the periodic transformation only requires a translation to match *Surface 1* to *Surface 2*. Therefore the rotational angle is not calculated.
Axis of rotation specification	**Specify axis**	Specifies that **Axis** node defines the axis of rotation.
Rotation angle	The calculated rotation angle that matches the periodic contact between *Surface 1* to *Surface 2* (read-only).	
Translation	The calculated translation that matches the periodic contact between *Surface 1* to *Surface 2* (read-only).	
Locked	**Activated**	Does not recalculate the rotation angle and translation when you generate the mesh.
	Deactivated	Recalculates the rotation angle and translation when you generate the mesh.

Axis Properties

The following table summarizes the properties that are available for the **Axis** node.

Direction	Specifies the direction of the axis of rotation relative to the user-defined *Coordinate System*.
Coordinate System	Specifies the coordinate system to which all other properties of this node are relative.
Origin	Specifies the origin of the axis of rotation relative to the user-defined *Coordinate System*.

See Also:

Part Contacts

What Is Metadata?

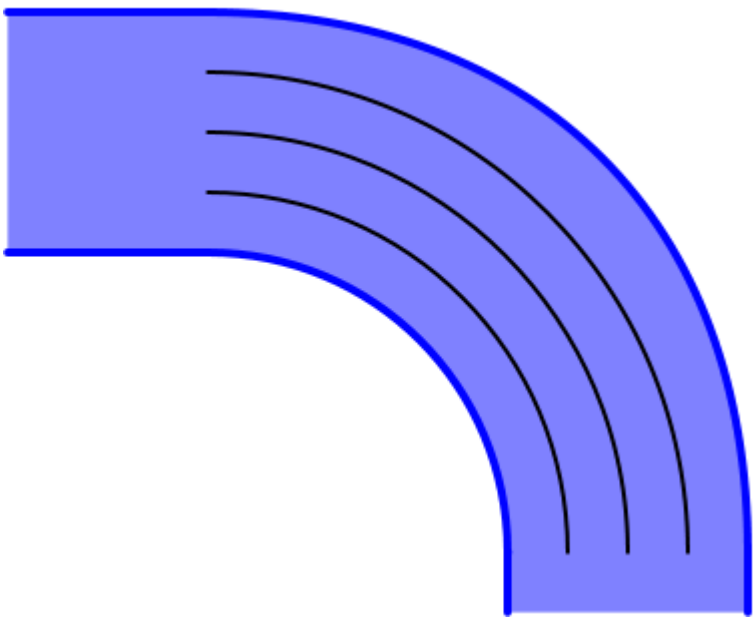
Baffle Contact

A baffle contact is a zero-thickness surface that the meshers maintain in the surface and volume meshes.

Defining a baffle contact maintains the shape of the zero-thickness surfaces in the mesh. The meshers assume that the baffle is a two-sided surface and generate a conformal mesh on either side.

Example of a Baffle Contact

The following example shows guide surfaces in a 90 degree bend. If the thickness of these guides is negligible, you can mesh them as zero-thickness baffles.



Contents:

Baffle Contact Properties

Baffle Contact Properties

The following table summarizes the properties that are available for baffle contacts.

Part Surface 1	The read-only names of the part surfaces that are in contact.
Part Surface 2	
Conformality	Read-only property that indicates whether a contact is conformal or non-conformal.
Tags	Applies a user-defined tag so that you can organize mesh contacts.
Interface	Displays the interface associated with the part surface contact when the parts are assigned to regions.
Metadata	Specifies the information that you can use to filter the mesh contacts, which are based on properties that are unrelated to the STAR-CCM+ simulation.
Show baffle proxy surface	Activating this property shows the duplicated surface that defines the other side of zero-thickness baffle. This proxy surface is not required for meshing, but is required for modeling physics.

	This property also allows you to specify the positive direction of the baffle using the **Baffle Orientation Point** node.
Index	Index reference for the part surface contact (read only). This index is available through the **Part Surface Contact Index** field function.

Baffle Orientation Point Properties

The following table summarizes the properties that are available for the **Baffle Orientation Point** node.

Orientation Point	Specifies a point to which original baffle surface faces — the other side is the proxy surface. When you assign the part to a region, the original surface becomes a boundary and the proxy surface becomes a copy of that boundary.
Coordinate System	Specifies the coordinate system to which all other properties of this node are relative.

See Also:

Part Contacts

What Is Metadata?

Repairing CAD Geometry

The Geometry Repair tool provides diagnostic checks for assessing the validity of geometry and the means to repair the geometry automatically.

This tool can work at the 3D-CAD level or the parts level, as long as there is CAD data in the geometry part.

The Geometry Repair tool searches for and fixes Parasolid invalidities. These invalidities are errors in the geometry that the Parasolid modeling kernel identifies. The capabilities of the Geometry Repair tool are identical to the capabilities of the **Repair Invalid Bodies** option available when importing geometry. If the **Repair Invalid Bodies** option does not fix a particular issue, then the Geometry Repair tool is not able to fix the issue either. While the **Repair Invalid Bodies** option has a limit to how many errors it can fix, the Geometry Repair tool can fix any number of errors.

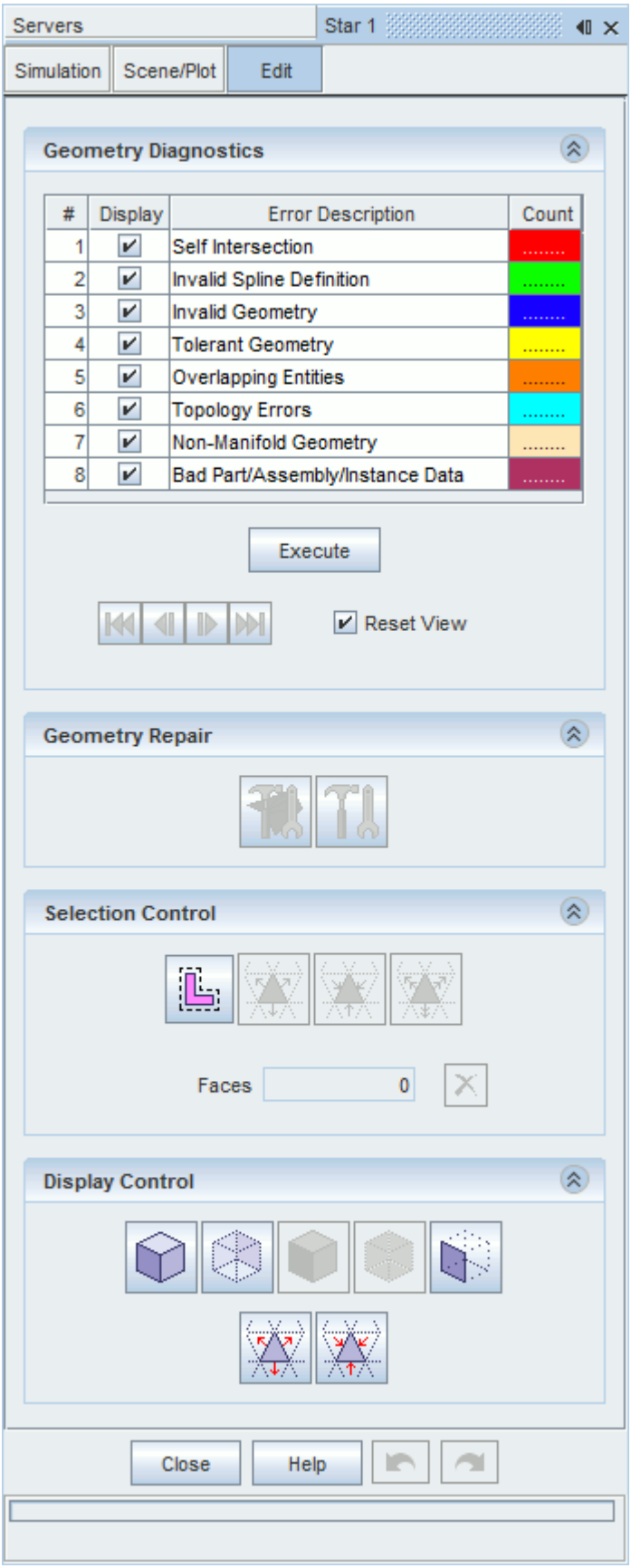
The Geometry Repair tool provides you with a user interface and a means to visualize the problem areas on the geometry. Even if this tool cannot repair a particular issue, the tool helps you to identify the problem areas on the geometry. You can then return to the native CAD package and address the identified issues there.

To repair geometry using the Geometry Repair tool:

1. Launch the tool:
   - In 3D-CAD, in the feature tree, right-click the feature that corresponds to the imported geometry and select **Repair CAD**.
   - In STAR-CCM+, from the **Geometry** > **Parts** node, right-click the geometry part and select **Repair CAD...**.

     In the *Geometry Parts Repair Options* dialog, select the appropriate options for your situation and click **OK**.

   The *Geometry Repair* panel appears.

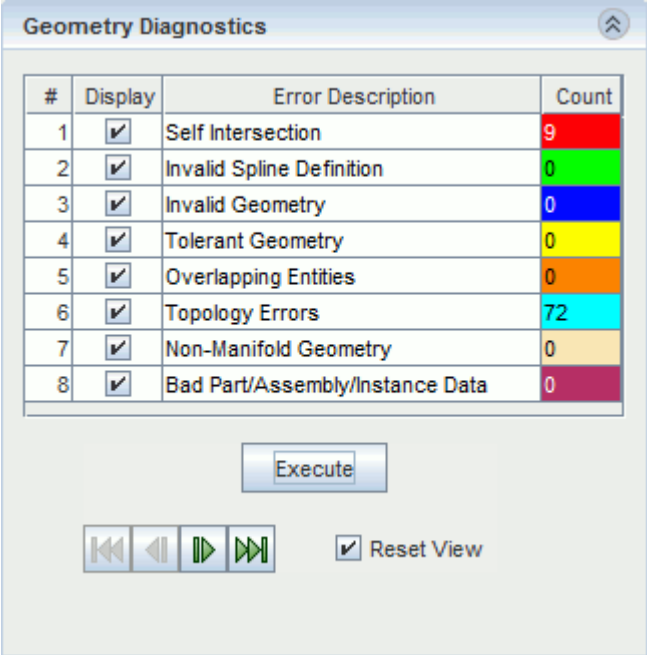

Servers
Star 1
Simulation
Scene/Plot
Edit
Geometry Diagnostics
Display Error Description Count
1 Self Intersection
2 Invalid Spline Definition
3 Invalid Geometry
4 Tolerant Geometry
5 Overlapping Entities
6 Topology Errors
7 Non-Manifold Geometry
8 Bad Part/Assembly/Instance Data
Execute
Reset View
Geometry Repair
Selection Control
Faces
0
Display Control
Close
Help

In the *Geometry Scene* window, the individual CAD faces are displayed on the geometry.

2. To run the diagnostics check, click **Execute**.

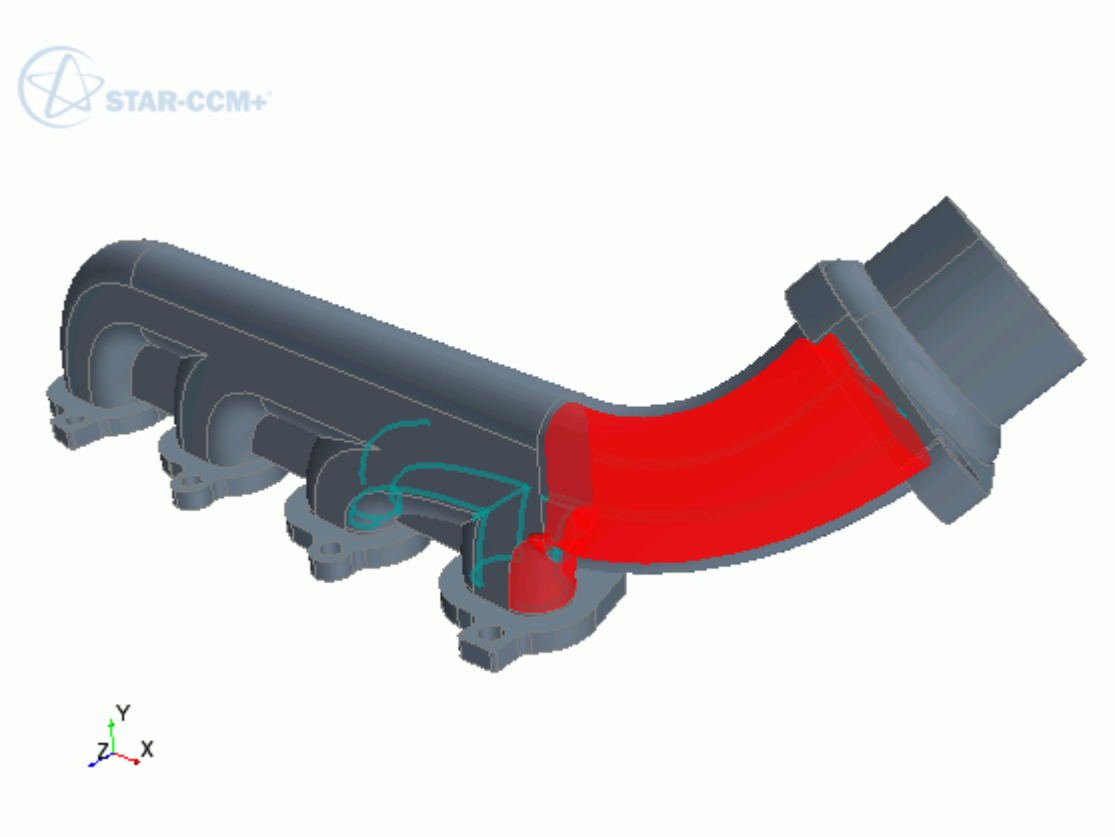
   The tool analyzes the geometry for Parasolid invalidities and displays the results in the *Geometry Diagnostics* panel.

   Errors are classified into 8 color-coded categories.



   See Geometry Error Categories.

   The affected areas are highlighted on the geometry in the color that is associated with the type of error.



Clicking on the error types in the *Geometry Diagnostics* panel selects the CAD faces that correspond to that particular error. Edges and vertices are not selected.

Note: Running diagnostics on large files with many parts is computationally expensive. If you imported the CAD geometry in the parts level, try to limit the diagnostics to smaller groups of parts where possible. You cannot select what specific parts to repair when working in the 3D-CAD level.

3. Use the keys in the *Geometry Diagnostics* panel to cycle through the identified problem areas.

   The *Geometry Scene* zooms into the problem area for each identified error. A short description of the error is provided at the bottom of the *Geometry Diagnostics* panel. Where appropriate, the location of the issue is displayed in the panel.

4. If necessary, modify the selection of the CAD faces and the visualization of these faces using the relevant functions. See Entity Selection.

5. Repair the geometry:

   - To repair the selected faces only, click (**Repair selected faces**).
   - To repair all identified errors, click (**Auto-repair geometry errors**).

   You can undo and redo actions while still in the Geometry Repair tool.

Warning: If you save the simulation while still in the Geometry Repair tool, the changes that you made so far are saved and the undo/redo functionality is reset. Once you exit the tool and accept the changes, you are no longer able to modify the repair. These actions are irreversible.

6. Click **Close**.

Contents:

Geometry Error Categories

See Also:

The Surface Repair Tool

Geometry Error Categories

The Geometry Repair tool identifies Parasolid invalidities and groups the various errors into 8 categories.

Diagnostics Category	Description
Self Intersection	Indicates errors that are associated with edges that cross each other and faces that self-intersect. In most cases, the Geometry Repair tool can fix these types of error by merging edges and vertices to remove the self-intersection.
Invalid Spline Definition	Indicates errors that are associated with poorly defined or incomplete sets of points that define a spline.
Invalid Geometry	Indicates errors in the relationships between entities in the geometry model. This type of error can also indicate that the model is corrupt. The Geometry Repair is likely not able to repair corrupted geometry. In this case, you are advised to return to the native CAD package to see if the errors can be repaired.
Tolerant Geometry	Indicates errors that are associated with loose tolerances of vertices, edges, and faces in the geometry. Loose tolerances can result in loss of data/definition of the geometry. For example, a high tolerance can prevent a gap from being recognized as a gap and ultimately stopping the gap from being closed. If the Geometry Repair tool cannot repair tolerance issues, you are advised to return to the native CAD package and look at the tolerance settings and tighten up the tolerance settings there.
Overlapping Entities	Indicates errors that are associated with overlapping faces. This type of issue could be due to tolerance issues in the geometry model.
Topology Errors	Indicates errors that are associated with fundamental errors in the entity definition. For example, faces that have edge loops that are defined in the opposite direction to other edges in the geometry.
Non-Manifold Geometry	Indicates errors where the geometry contains non-manifold vertices or non-manifold edges. A vertex is non-manifold if any of its connected faces have no other link with the rest of its connected faces apart from the vertex. An edge is non-manifold if it is shared by two or more faces.

Diagnostics Category	Description
Bad Part/ Assembly/Instance Data	Indicates errors that are associated with the assembly definition, and the parts and instances within a model. This type of error is typically caused by poorly defined shells and solids. Another cause could be issues with instancing and missing information from the parts and assemblies.

The Surface Repair Tool

The *Surface Repair* tool diagnoses errors in, and provides repair tools for, tessellated surfaces and feature curves.

It is good practice to check the validity of the surfaces on your geometry before you attempt to generate a surface mesh. Among other things, there can be faces on the surface that intersect other faces, or edges that have two or more shared faces. Other types of preparation can be required on the surfaces of your geometry. For example, if you have multiple parts that are to form multiple regions in the computational domain, you may have to imprint the surface of one part on the surface of another part at the interface between them.

You are advised to prepare and repair CAD geometry using the *Repair Geometry* tool, 3D-CAD, or the CAD operations available at the parts-level, before using the *Surface Repair* tool. The *Surface Repair* tool operates on the discretized surface, meaning that the changes only apply to this representation and do not form part of the overall meshing pipeline. All changes that you make using the *Surface Repair* tool are lost when you modify the original CAD geometry.

Changes that you make to the surface using the *Surface Repair* tool are permanent once you exit the tool. Operations that you carry out while the tool is active are reversible.

Contents:

The Surface Repair Workflow
Surface Repair User Interface
Selecting Entities
Repairing Surfaces
Repairing Features
Surface Repair Error Categories
Repair Tools Reference
Global Tools
Query Tools
Part Controls
Set Controls
Query Diagnostic Responses From Java Macros
Surface Repair Hotkeys
Pop Up Menu of the Surface Repair Tool Graphics Window

See Also:

Using the Properties Window
Repairing CAD Geometry
Changing the Parts in the Active Model
What Are Representations?
Understanding Geometry Parts

The Surface Repair Workflow

Use the Surface Repair tool to repair surface and feature curve errors manually on the discretized surface of the geometry.

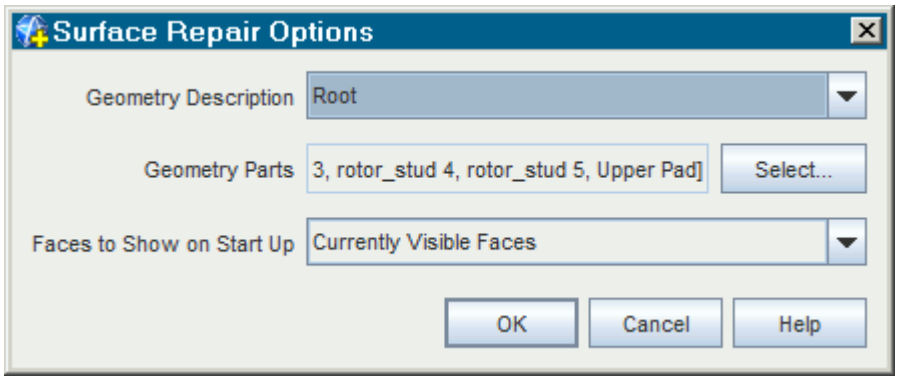
To use the surface repair tool:

1. Launch the Surface Repair tool using one of the following options:

   - From the **Mesh Generation** toolbar, click (**Launch Surface Repair**).
   - Right-click a geometry part or a surface representation and select **Repair Surface...**.

   If you have geometry parts, but no region surface representations, the tool automatically launches with the geometry parts available for repair. If you have one representation, but no geometry parts, the tool automatically launches with the representation available for repair. In all other cases, choose either **Repair Part(s)...** or **Repair Region(s)** from the pop-up menu.

2. In the *Surface Repair Options* dialog, select the parts that you want to bring into the Surface Repair tool and click **OK**. See Surface Repair Options Dialog Reference.



   The Surface Repair environment appears. See Surface Repair User Interface.

   When you launch the *Surface Repair* tool, STAR-CCM+ does a topological orientation on the faces you selected. That is, it makes sure that all the faces are consistently oriented in the same direction, whether pointing inwards or outwards. STAR-CCM+ picks the orientation belonging to the largest number of faces and orients them all in that direction.

3. Typically at this stage, you can:

   - Repair surface errors. See Repairing Surfaces.
   - Repair feature curve errors. See Repairing Features.
   - Perform a leak detection. See Leak Detection.
   - Merge or Imprint surfaces. See Merge and Imprint Mode.

     For CAD geometry, you are advised to imprint surfaces using the Boolean operations in 3D-CAD or at the parts level.

4. Identify problem areas.

   A problem area is a collection of diagnostics that are connected to each other by one or more shared vertices. Faces, edges, or vertices can be identified in more than one diagnostic check. Each face, edge,

or vertex is only displayed in the problem area collection that belongs to the last diagnostic check to identify it.

- Click on the individual colored boxes to highlight all errors of that type. Hold the <`Ctrl`> key to select multiple boxes.
- Use the (**Browse**) buttons to cycle through individual errors, one at a time.

  The scene zooms in to focus on the specific errors as you cycle through.

Note: If you modify the surface, the diagnostics and problem areas are automatically updated.

If more than 100 problem areas are identified, merges between two or more problem areas that were previously unconnected are no longer performed.

Smart selection (double-clicking a face or edge that contains a diagnostic result) of what appears to be continuous problem areas no longer produces the expected selection result. Smart selections are limited to the portions of the problem area that they belong to, even though visually they appear to be connected together.

The following message is displayed in the *Output* window:

```
Warning: Too many problem areas in surface, will not attempt to merge
neighboring areas.
```

To resolve this issue:

- Decrease the number of diagnostics that are activated. This action reduces the potential number of problem areas.
- Enter the *Manage Thresholds* panel and click **OK** to exit. The problem areas automatically recalculate. Smart selection results are then as expected, regardless of the number of problem areas.

5. Manage how the errors are displayed in the *Graphics* window:
   - Use the display controls to control how the geometry is displayed in the *Graphics* window. See Display Control Tools.
   - The **Reset View** option resets the view to zoom in on the displayed faces, edges, and vertices. Deactivate the **Reset View** option to maintain the same view.
   - The **Reset Displayed** option resets the displayed faces, edges, and vertices to the error and two neighbors. Deactivate the **Reset Displayed** option to display the same entities.
6. Use the selection controls to select the entities you want to repair. See Entity Selection.
7. Fix errors using the appropriate tools. See Repair Tools Reference.

   The tools do not allow you to create new vertices directly. If you want to fill many holes or zip many edges, use the hole filler and edge zipper tools respectively. For other cases, use the surface repair workflow to identify and repair issues.

Note: You can create coordinate systems without closing the Surface Repair tool. To create a coordinate system, switch to the **Simulation** tab to access the STAR-CCM+ object tree and create the coordinate system under the **Tools** > **Coordinate System** node. Coordinate systems that you create while the Surface Repair session is active are automatically available in tools that use a coordinate system selection. See Coordinate Systems.

8. If necessary, use the (**Undo**) and (**Redo**) functions to undo and redo actions.

   For example, if you deleted a face by mistake, click (**Undo**) to restore the deleted face and returns the part to its previous state. A complete history of all operations in the current surface preparation session is available. The history is deleted when you close the dialog.

Contents:

Surface Repair Options Dialog Reference
Changing the Parts in the Active Model

Surface Repair Options Dialog Reference

Select the parts and regions to repair using the *Surface Repair Options* dialog.

Geometry Description	Specifies the representation to bring into the *Surface Repair* tool.	
Geometry Parts	Specifies the parts to bring into the repair tool. Make selections with the *Select Objects* dialog.	
Faces to Show on Start Up	Specifies the surfaces to display in the *Surface Repair* tool when it launches.	
	No Faces	Displays no faces at startup. Select faces using the selection controls, or by clicking the categories of errors that are discovered during any diagnostics check. This option is useful for large models where plotting is slow.
	All Faces	Displays all faces in the model.
	Currently Visible Faces	Displays only the faces that are shown in the active scene.

Changing the Parts in the Active Model

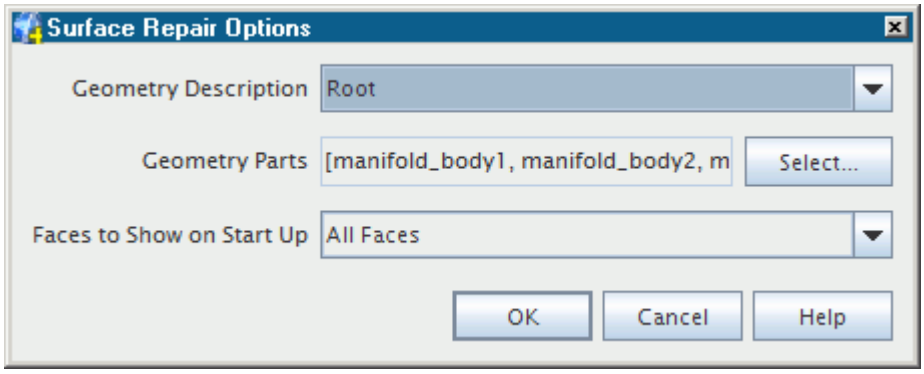
The active model is the parts or regions that are available for repair during the surface repair session.

You select the active model in the *Surface Repair Options* dialog after launching the *Surface Repair* tool. This feature allows you to simplify a complex geometry by working with smaller groups of parts or regions for diagnostics and repair. You can change the active model at any time in the *Surface Repair* tool.

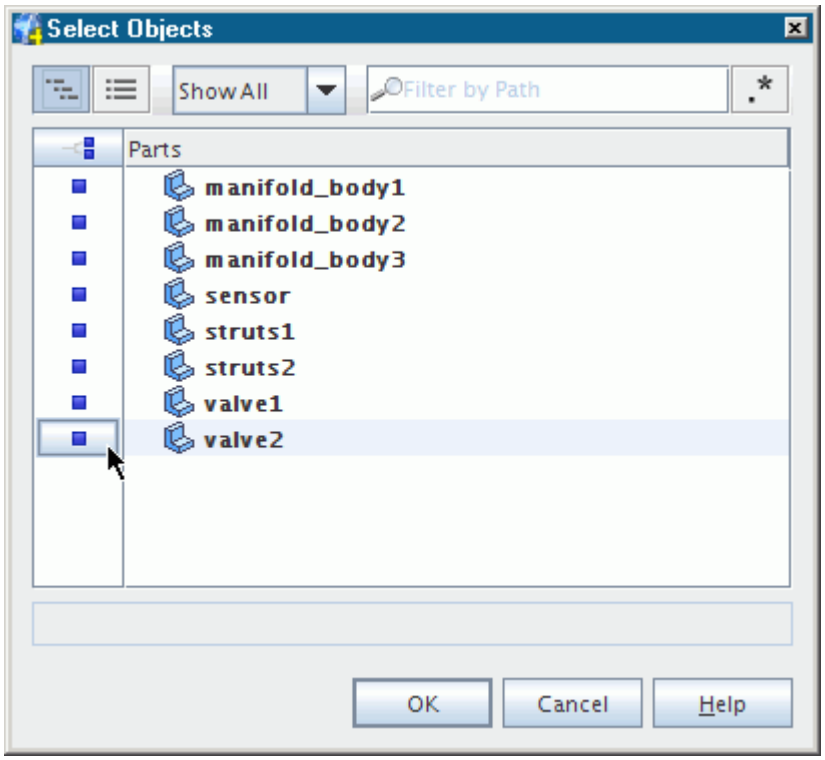
To change the parts or regions in the active model:

1. Click **Reset Parts...** or **Reset Regions...** at the bottom of the *Surface Repair* tool.

   The *Surface Repair Options* dialog, which has similar options to the dialog that appears when you launch the *Surface Preparation* tool, appears. The only exception is the **Faces to show on startup** menu, which allows you to display **All Faces** or **No Faces** only. The **Currently Visible Faces** option is not available, as changing the parts means that the currently visible faces are unlikely to be in the new active model. The default **Geometry Description** is set to the description that is currently being repaired.



2. Click the **Select** button. In the *Select Objects* dialog, select or deselect a part or region, click the button to the left of that item as shown:
3. In the *Select Objects* dialog, select or deselect a part or region and click **OK**.



Changing the parts in the active model makes the following changes to the surface repair session:

- All face, edge and/or vertex selections are deselected.
- The surface diagnostics and feature diagnostic results are deselected and the *Diagnostics* panel returns to its default state. However, the state of the diagnostic checkboxes is maintained.
- The Edit Connected Surfaces diagnostics are deselected.
- All user-defined sets are deleted and entities in the default set are deselected.
- The Undo/Redo stacks are deselected.
- The single and multi-parts Merge/Imprint pairs lists are deselected.

- Faces selected for the Intersect/Boolean Faces Tool are deselected.
- Faces selected for Find Leak Path are deselected.
- The list of part surfaces and part curves available in the Part Surface/Part Curve Edit group box is automatically updated to match the new parts in the active model. The same is true for regions, regarding boundaries and feature curves.

When projecting vertices, the *Project Vertices* tool temporarily replaces the *Surface Repair* tool. Therefore, you are unable to change the parts in the active model while projecting vertices, as the **Reset Parts...** button is unavailable.

See Also:

Using the Leak Path Tool

Intersect/Boolean Faces Tool

Set Controls

Edit Connected Surfaces Tool

Part Controls

Merge and Imprint Tool

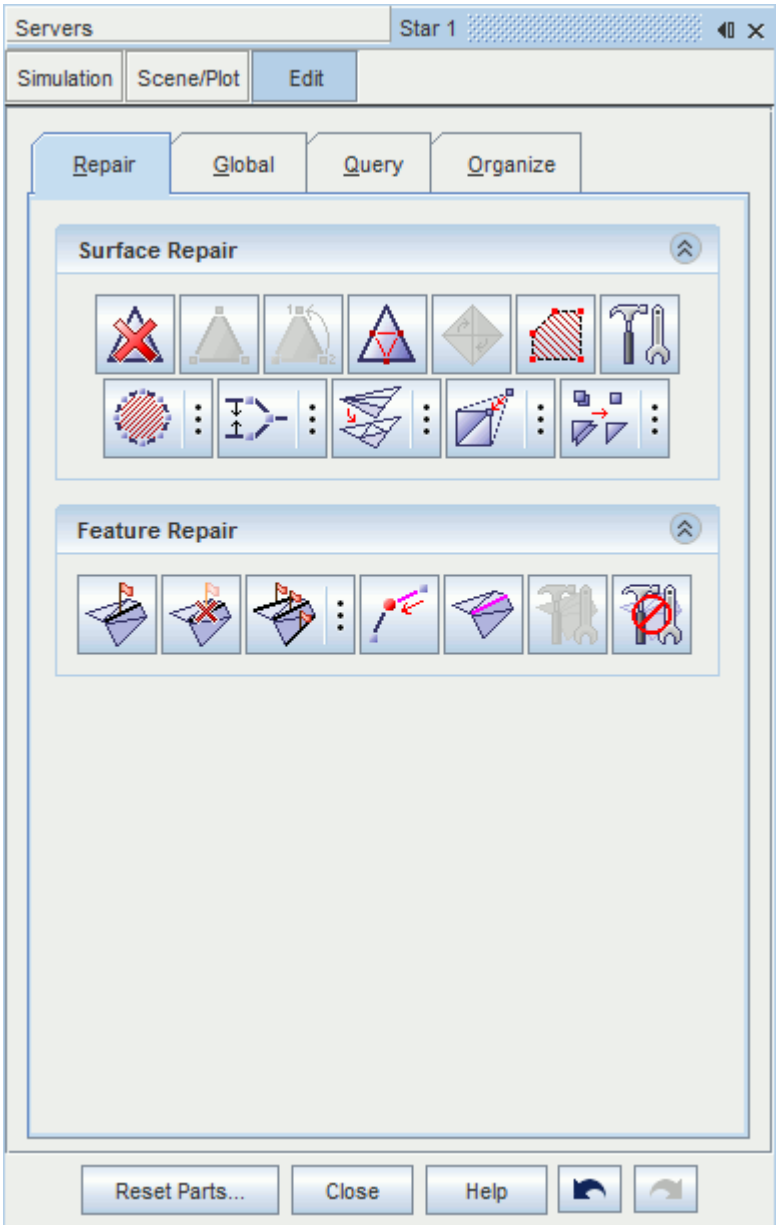
Projecting Vertices

The Surface Repair Tool

Surface Repair User Interface

The functions of the Surface Repair tool are contained within the Surface Repair environment. Repair tool modes are presented in the *Edit* panel; display, selection, and diagnostics functions are embedded within the *Graphics* window.

The different modes of the Surface Repair tool are contained within the panel on the left of the screen. Some tools contain options, and these are displayed in the bottom half of the panel.

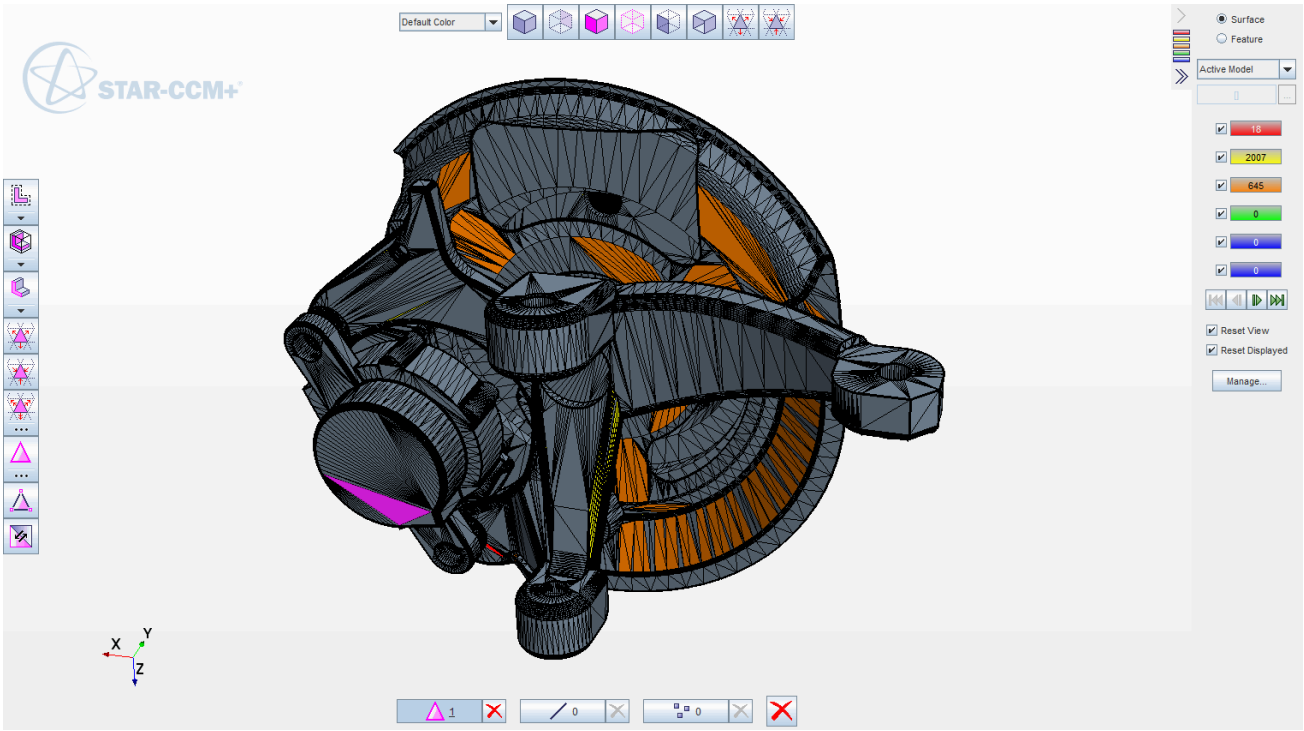


Note: To prevent accidental closure of the *Surface Repair* tool when repair operations are in progress, STAR-CCM+ temporarily deactivates the **Close** button. The **Close** button is available again when the repair operations are complete or after you click (**abort**).

Coordinate systems that you create in the STAR-CCM+ object tree while the repair session is active are automatically included in tools that use a coordinate system selection.

For a list of tools that are located in the panel, see Global Tools and Repair Tools Reference.

Features that are common to all modes of the Surface Repair tool are embedded in the *Graphics* window.



For a list of the embedded tools see Embedded Tools Reference.

Part of the embedded tools include the selection filters. See Filtering Selection Entities

Displaying Tool Options

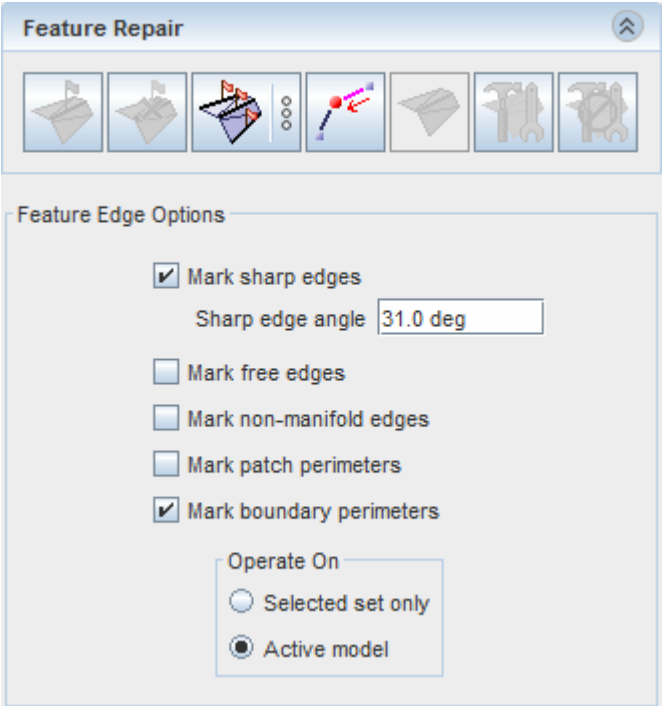
Tool options are indicated by three dots next to the tool icon . By default, the options for each tool are hidden and appear after you activate the tool.

To display the tool-specific options:

1. Click the three dots next to the tool icon.

   Note: The icon consists of two parts, the three dots and the actual tool. Clicking on the tool portion of the icon executes the tool function.

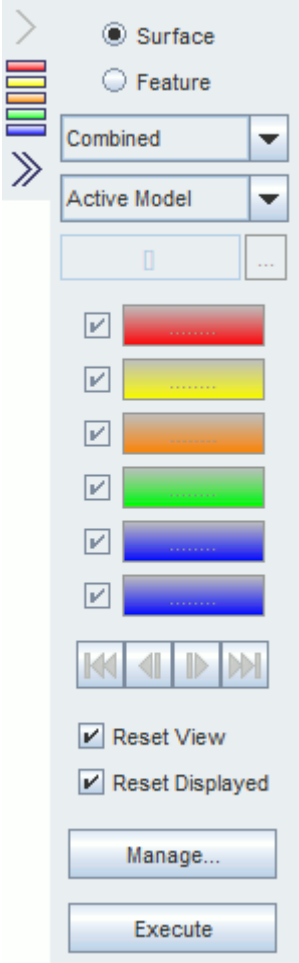
   The options appear in the bottom half of the repair panel.



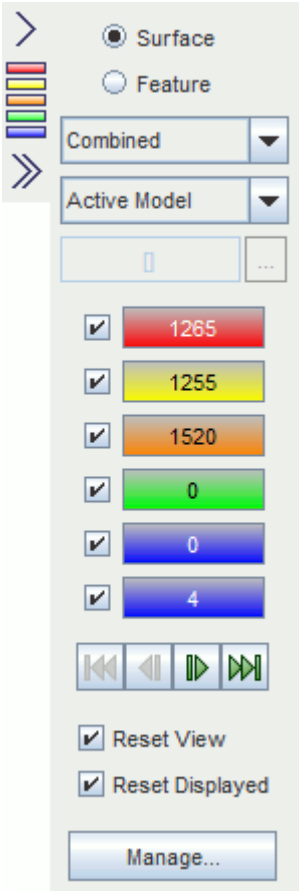
The three dots change from black to white to indicate that the options for the specified tool are displayed. You can use other tools while the options of another tool are displayed.

The Diagnostics Panel

The diagnostics slide out, located on the right side of the *Graphics* window, contains a list of thresholds and issues that the *Surface Repair* tool identifies. By default, the diagnostics are not executed. To calculate and display all enabled diagnostics, click **Execute**.

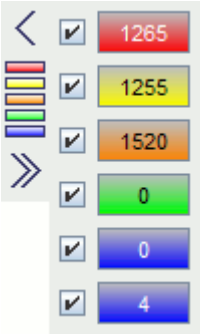


The tool analyzes the surfaces for errors and displays the results in the *Diagnostics* slide out. The **Execute** button is not longer needed at this stage because the results are always displayed and update in real-time. You can manage what thresholds are displayed in the *Geometry* window with the **Manage** feature. See Repairing Surfaces and Custom Threshold Options.



While you cannot close the diagnostics slide out, you can modify the size of the panel:

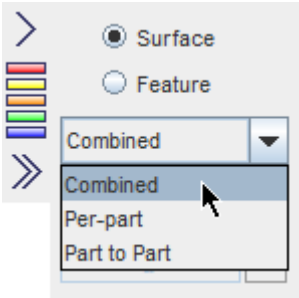
- To reduce the size of the slide out and show summary information only, click .



- To minimize the size of the slide out, click .

- To open the slide out or maximize its size, click .

The surface repair diagnostics tool can display results from each threshold diagnostic in three different interaction modes:



- **Combined** — is a mix of per-part diagnostics and part to part diagnostics, and applies to all threshold diagnostics (default and user-defined).
- **Per-part/Per-region** — only shows diagnostic results that are caused by or relate to the individual part in the absence of all other parts. All threshold diagnostics produce a per-part result. For example, in the case of pierced faces, if the part intersects itself then it will produce a per-part pierced face result. Similarly for proximity, if the part has faces in close proximity to itself, it will produce a per-part result.
- **Part to part/Region to region** — only shows diagnostic results that are due to the interaction of two or more parts. Only three diagnostics are relevant to this mode: **Pierced faces**, **Close proximity faces**, and **Duplicate faces**. All other diagnostics are hidden from the scene and the slide out in this mode.

For more specific information on how the interaction modes affect each diagnostic, see Surface Repair Error Categories.

Selecting Entities

You can select entities using a number of methods, including some specialized tools.

Most of the selection tools are embedded in the *Graphics* window. These tools are allow you to select faces, edges, and vertices by specifying a zone, threshold, or surfaces, or by growing and shrinking from an existing selection. See Selection Control Tools

Specialized tools are available in the repair panel:

- Toggle Chase Edge Selection Mode. See Using the Chase Edge Selection Mode.
- Toggle Select Curves Mode. See Using the Select Curves Mode.

Other methods of selecting entities are listed below:

- Right-click surfaces in the *Graphics* window to select associated entities.
- Click individual entities manually in the *Graphics* window to select them.
- Double-click an edge to select a continuous line of edges, whose face normals are similar to one another.
- After you have run surface diagnostics, double-click a free edge to select all connected free edges.
- Double-click a face to select a continuous surface that is constrained by feature curves or sharp edges.

Filtering Selection Entities

You can filter what is selected by each tool. Face, Edge, and Vertice filters restrict the entities available for selection. The default options activate the **Faces** filter, meaning the mouse only selects faces, with multiple selection possible by pressing `<Ctrl>` and selecting more items. Activating the **Edges** and **Vertices** filters allows selection of edges and vertices respectively.

To modify the selection filter:

1. Click on the desired filter icon to activate the filter.

   The number of selected entities for that filter are displayed in each filter icon. The state of the filters does not affect the grow and shrink options.

Clearing Selections

To deselect entities:

- To deselect individual entities, click selected entities in the *Graphics* window, while pressing the `<Ctrl>` key.
- To deselect all selections, click anywhere off the part in the *Graphics* window.
- To deselect all faces, edges and vertices independently, click next to the relevant filter icon.
- To deselect all entities, click (**Select None**).

Contents:

Embedded Tools Reference
Selection Options
Selecting With a Specified Threshold
Selecting Surfaces and Curves with the Selection Controls
Selecting Surfaces Or Curves With The Graphics Window
Using the Chase Edge Selection Mode
Using the Select Curves Mode

See Also:

Selecting With a Specified Threshold
Selecting Surfaces and Curves with the Selection Controls
Selecting Surfaces Or Curves With The Graphics Window
Using the Chase Edge Selection Mode
Using the Select Curves Mode

Storing Sets

Embedded Tools Reference

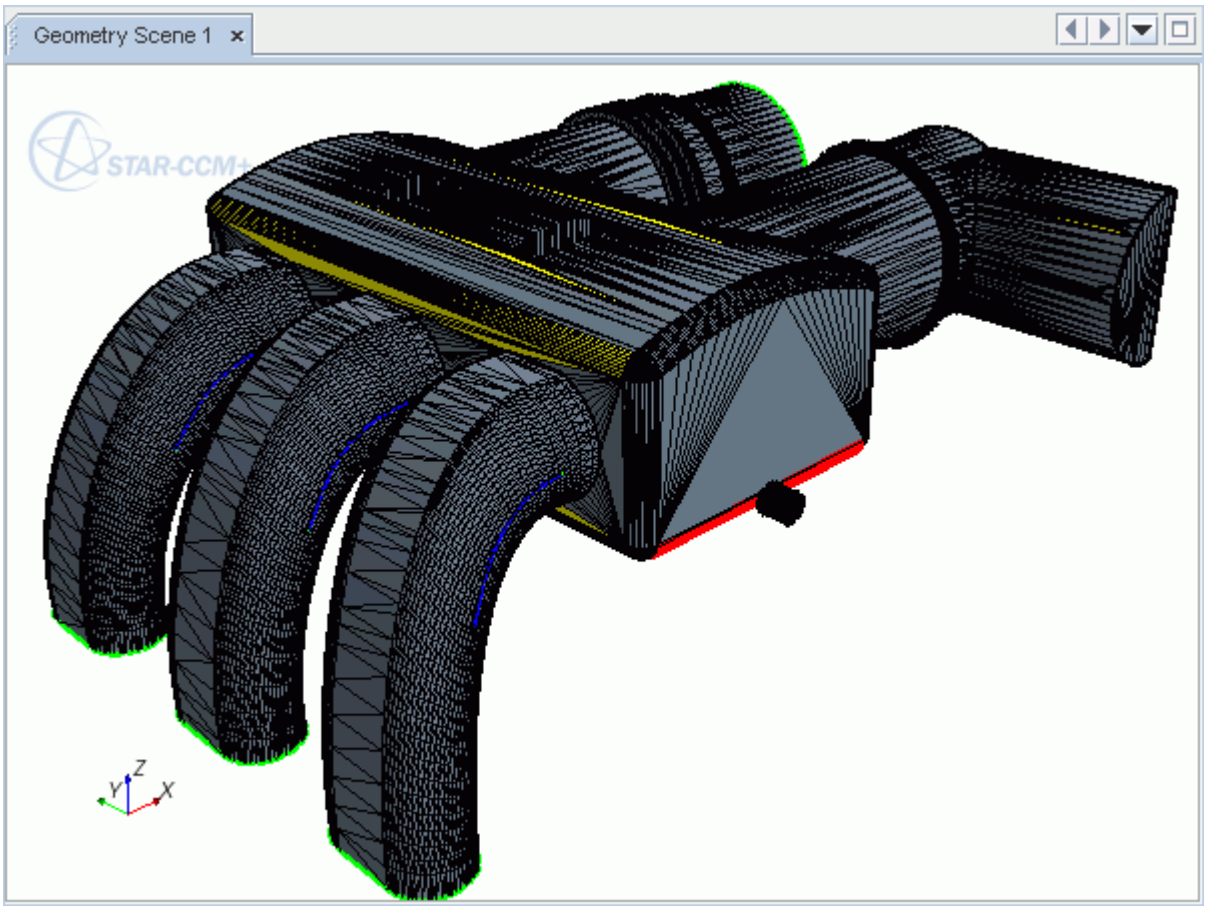
Common tools are embedded in the *Graphics* window.

Color Modes

Color modes color visible faces in the *Graphics* window. Using color modes other than the default color, disables the diagnostic colors.

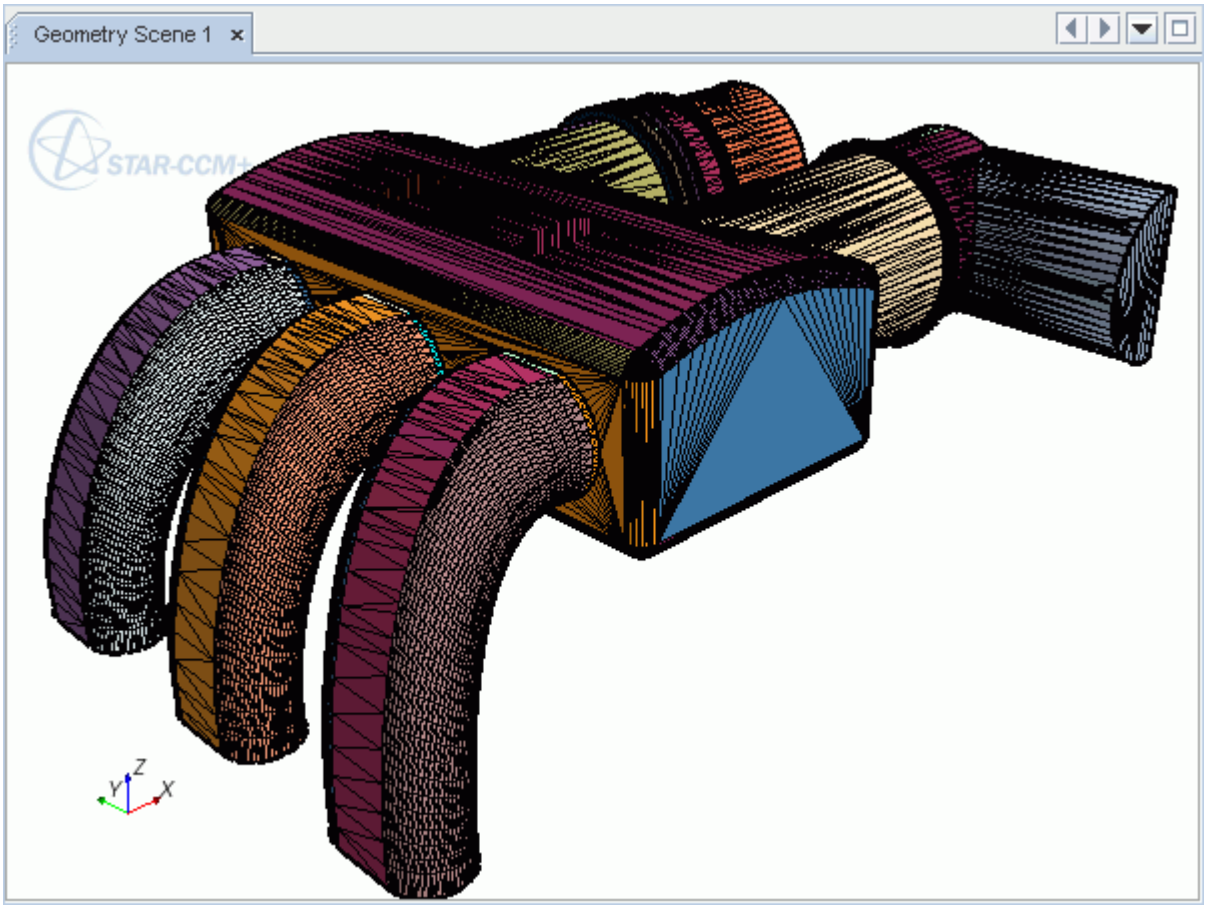
Default Color

Colors faces in the display with the default color, apart from erroneous faces that are colored by the surface diagnostics, and source and target faces in imprints and intersections.



Patch

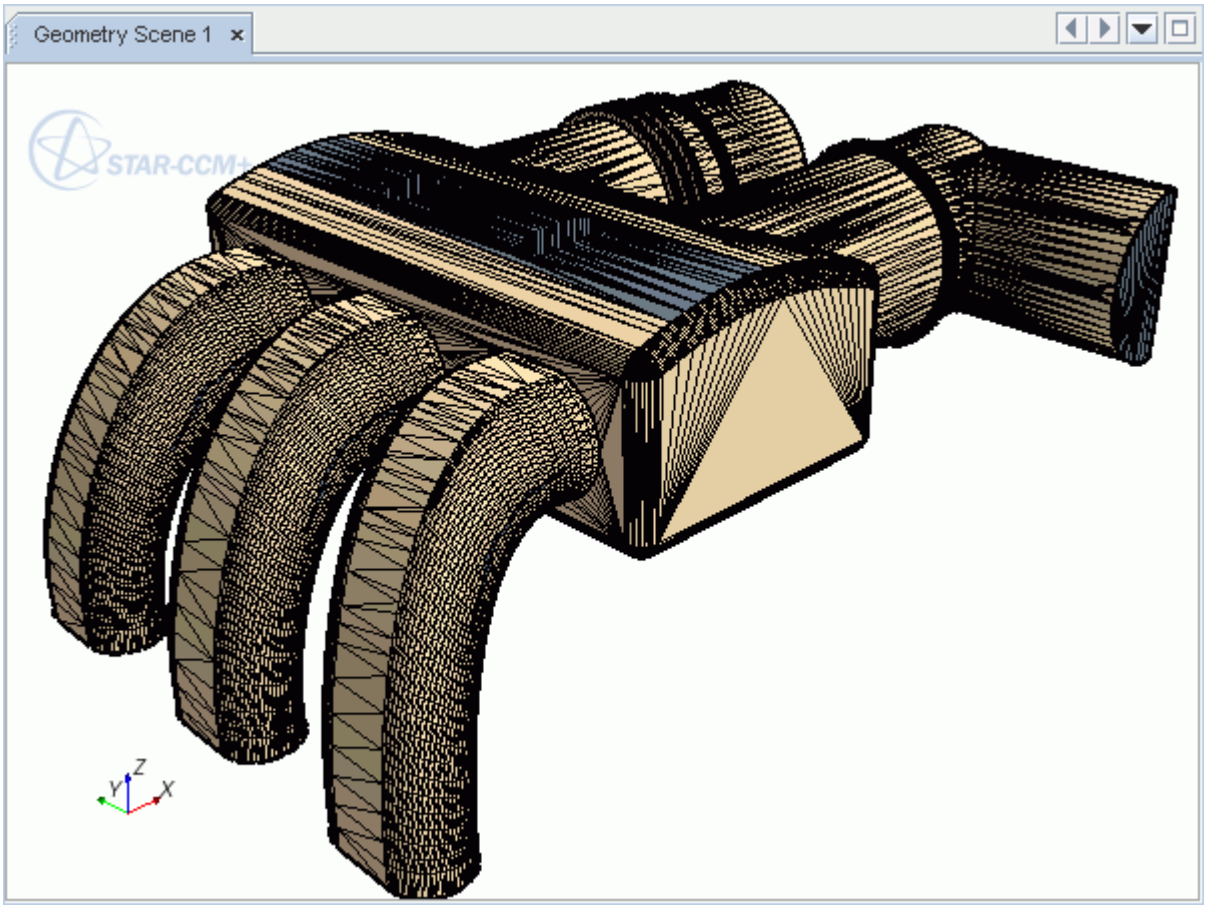
Colors faces according to their underlying patch definition. Faces that share a patch definition use the same color.



CAD Association

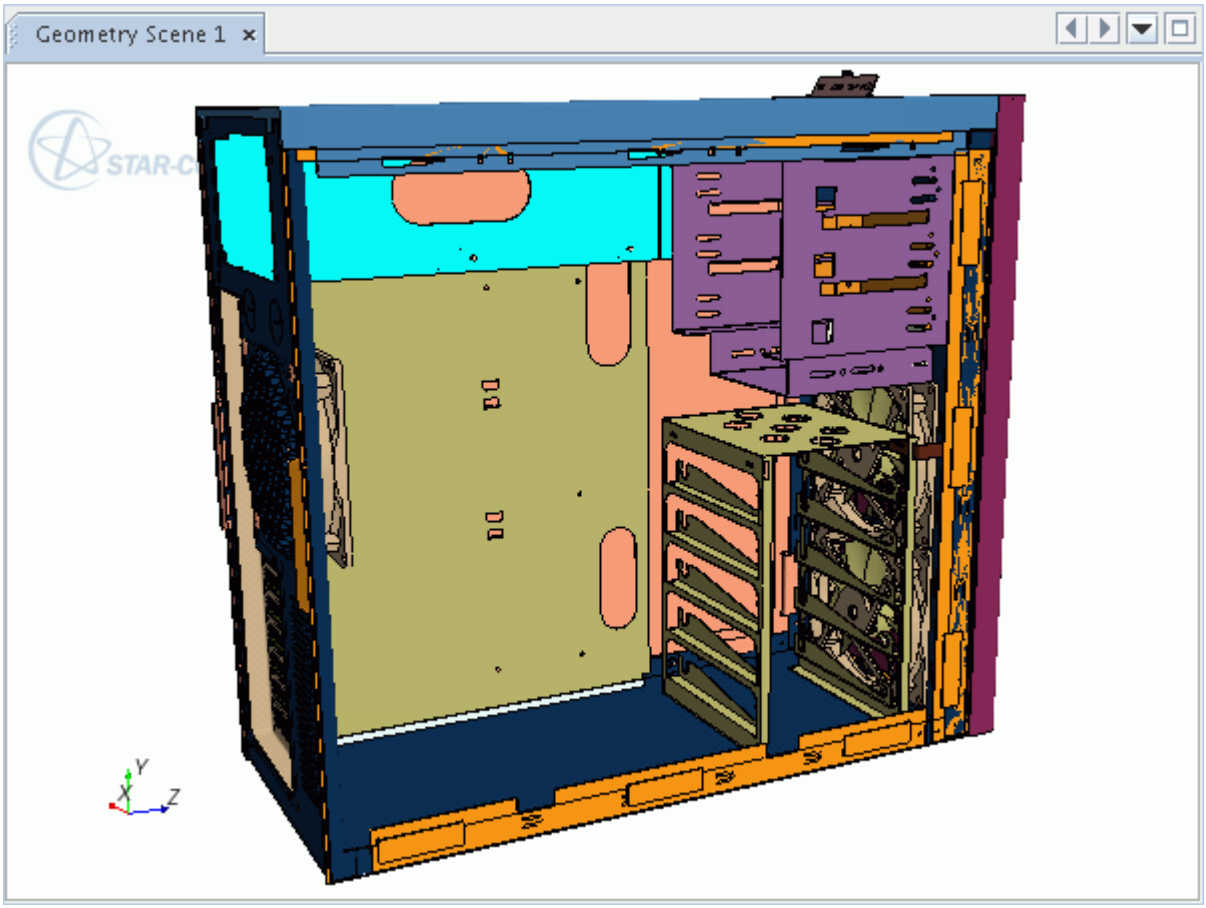
Colors faces according to whether they have CAD association or not.

Beige faces represent surfaces that have underlying CAD available for CAD projection during the meshing process. Dark gray surfaces represent surfaces with no CAD association. An example of the CAD association color mode is shown below.



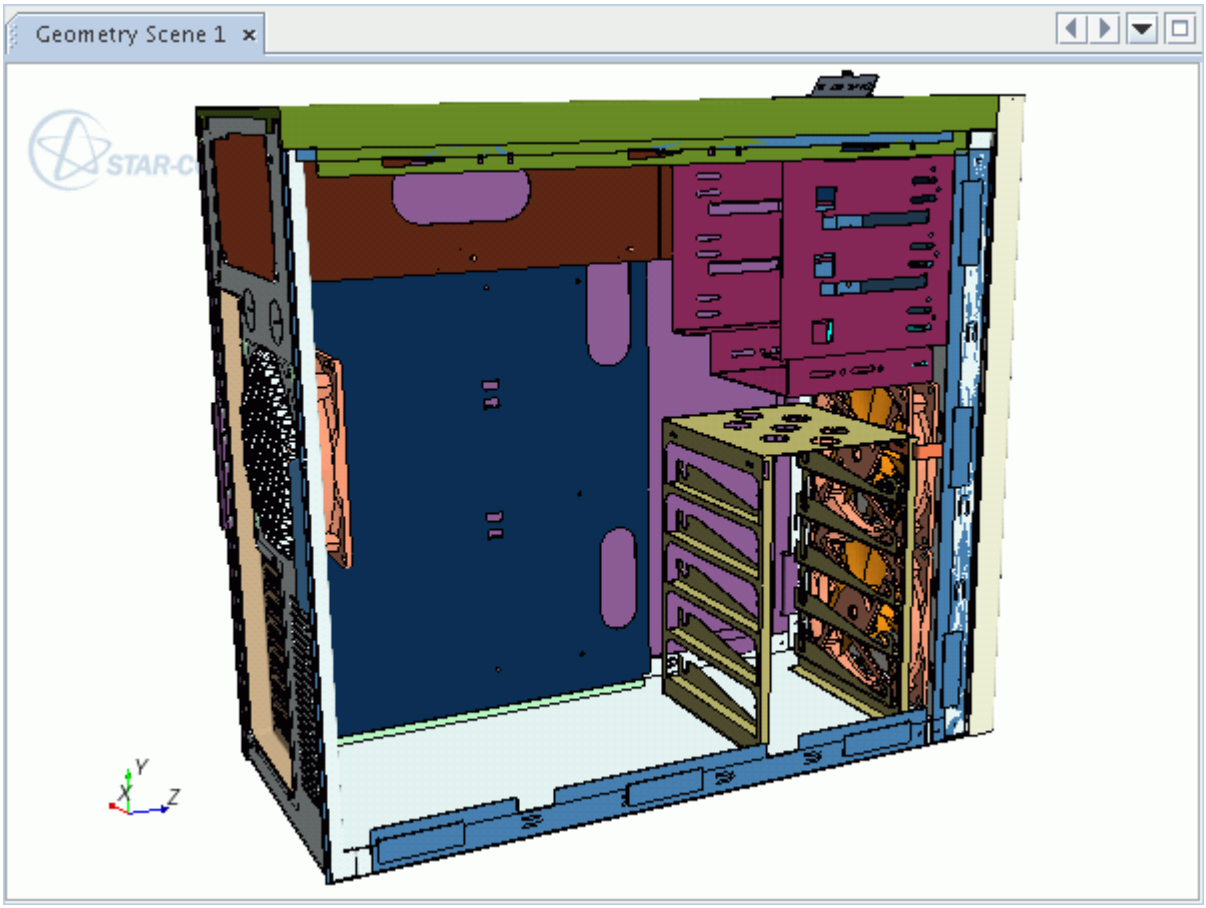
Part Surface

Colors faces according to the surface to which they belong.



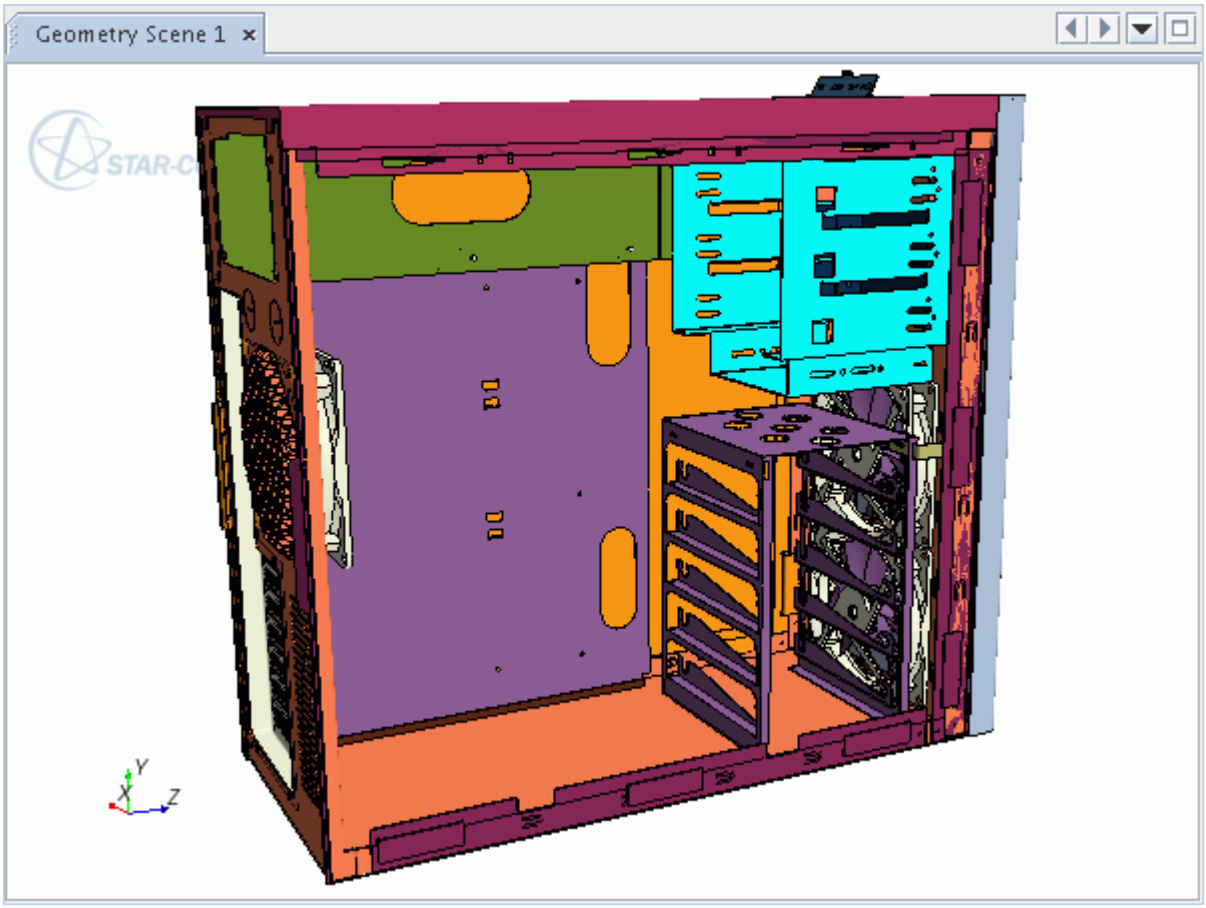
Part Boundary

Colors faces according to the boundary to which they are assigned.



Part

Colors faces according to the part to which they belong.



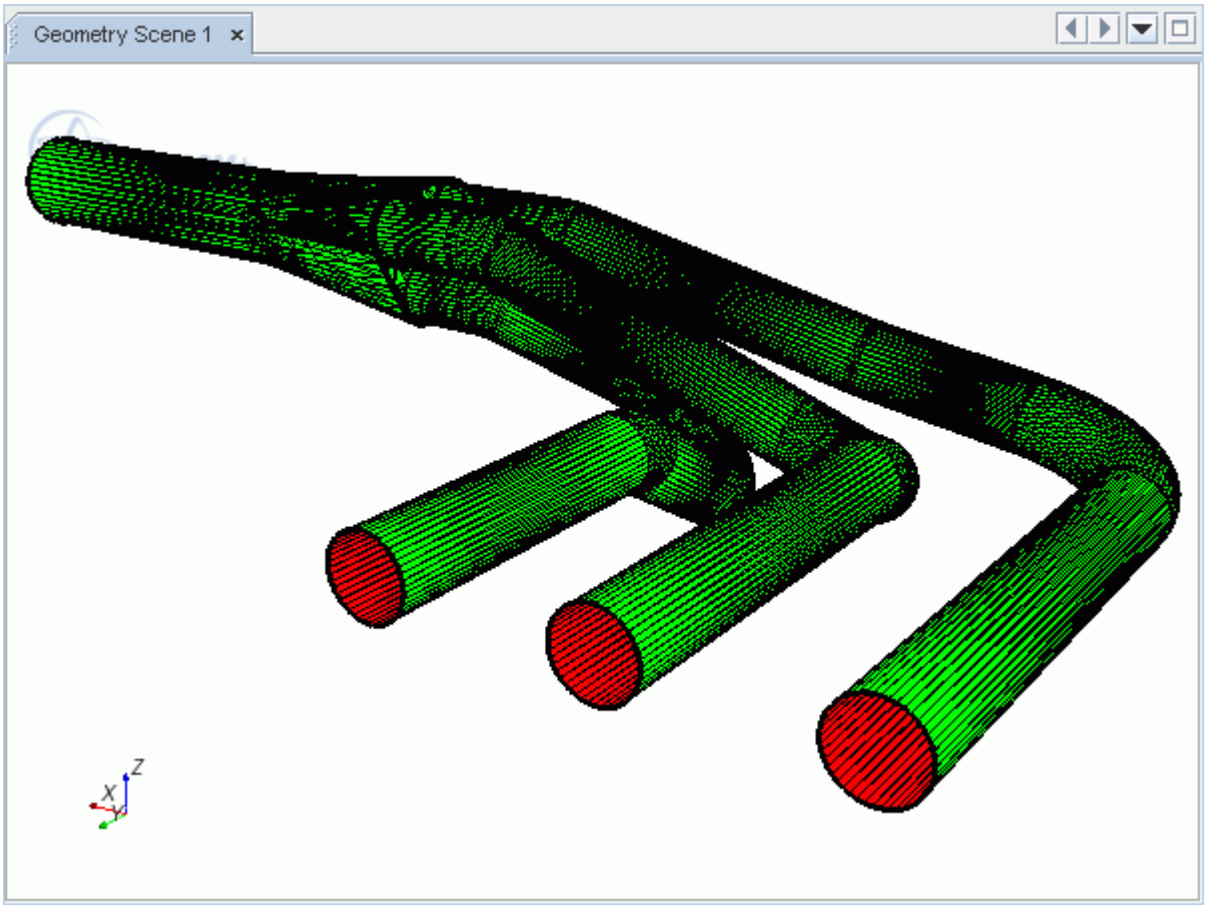
Face Normal

Colors faces according to their topological orientation, that is, the direction of the face normals.

Green faces have a positive face normal and red faces have a negative face normal.

For a closed-body, the positive face normals (green) point outwards, and negative face normals (red) point inwards. If you import a surface representing the solid component and manually extract the wetted surface using surface repair tools, the orientation is reversed as the face normals were pointing outwards from the solid component upon import.

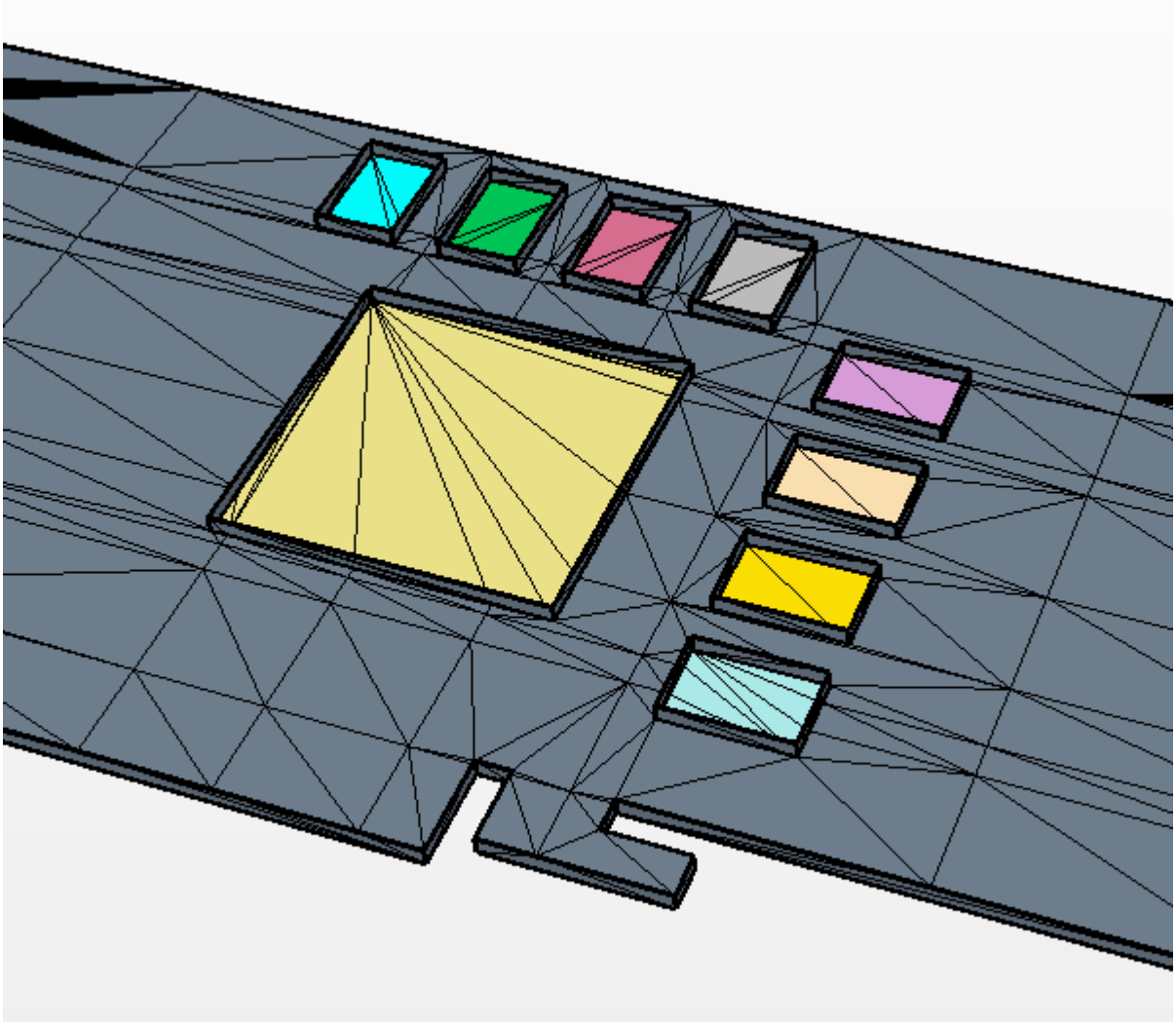
When in **Merge/Imprint** mode, this option overrides the default colors. Therefore, the green and red faces do not identify the source and target faces, but identify the positive and negative face normals instead.



Contact/Interface

Colors faces within a contact/interface according to their underlying part surfaces/boundaries.

Use the Contact/Interface color option to quickly visualize and assess the quality of any given imprint result. Any gray or different color faces seen within the context of the main contact/interface color indicates an incomplete or incorrect result and needs further attention. The colors used to represent a contact will not be the same color as used for the underlying part surface coloration. For example, if you toggle back and forth between coloring by part surface and coloring by contacts, different colors are applied to the contacting faces.



Display Control Tools

Show All Faces

Displays all part faces. Any currently selected faces remain selected.

Hide All Faces

Hides all part faces. Any currently selected faces remain selected.

Show Selected Faces

Displays the faces currently selected in the *Graphics* window.

Hide Selected Faces

Hides the faces currently selected in the *Graphics* window. You can redisplay the hidden faces by growing from neighboring displayed faces, or by showing all faces.

Show Faces on Part Surfaces

Specifies the part surfaces and contacts to display using the *Show Part Surfaces* dialog.

Hide Faces on Part Surfaces

Specifies the part surfaces and contacts to hide using the *Hide Part Surfaces* dialog.

Grow Displayed Faces

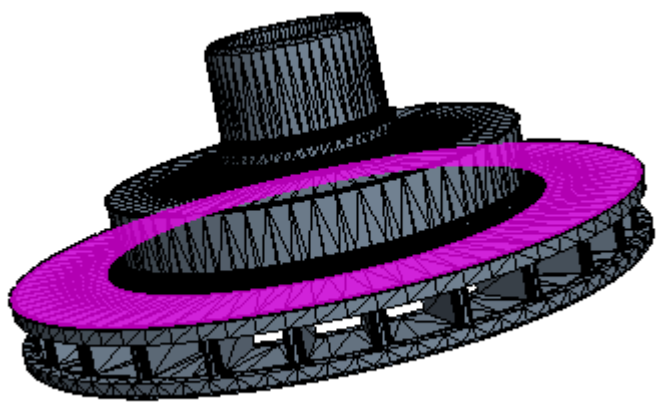
Grows display by one set of neighbor faces from the currently displayed faces.

Shrink Displayed Faces

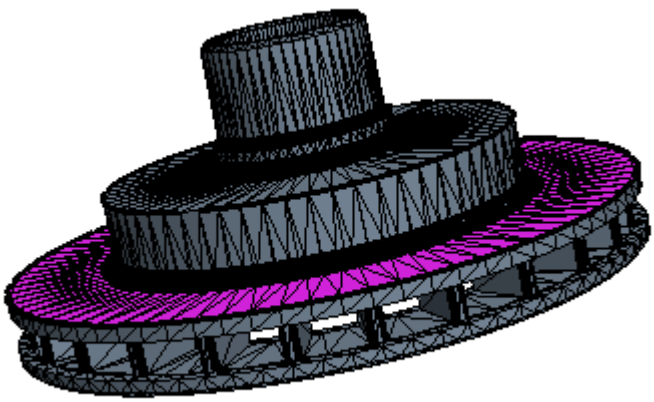
Shrinks selection by one set of neighbor faces from the currently displayed faces, which are based on connected vertices.

Toggle Face, Edge and Vertex Selection Depth Control

Switches depth control method for displaying selected faces, edges, and vertices. By default, the depth control is not activated and all selections are displayed through the model.



When depth control is active, selections that are obscured by the outside surfaces of a model are hidden from view.



Selected feature edges are not highlighted in the scene when depth control is on. Turning on screen transparency negates the effects of the depth control.

Selection Control Tools

Select Zone

This tool selects entities on the displayed surface using a polygonal zone.

The tool selects a face if all three vertices belonging to the face are within the zone, even if the zone does not completely encompass the face. The tool selects an edge if the two vertices belonging to the edge are within the zone. The tool selects a vertex when the vertex is within the zone.

To end the zone selection, right-click in the *Graphics* window or click the starting point.

The various functions of the tool are described in Selection Options.

Select Specified Threshold

This tool selects entities on the displayed surface using the *Threshold Selection* dialog.

Four selection options are available for the zone selection tool, which you can access by expanding the **Select zone** button.

Using a threshold selection option opens the *Threshold Selection* dialog, where you specify the **Threshold**, and enter **Minimum** and **Maximum** values. See Selecting With a Specified Threshold.

Select Face/Edges on Part Surfaces/Part Curves

This tool can be used to provide an object selector to select, add, remove, or subset part surfaces, curves, points, and contacts in the active model. For a part surface/curve/point, this tool selects the associated entities according to the faces, edges, and vertices filters. Baffle contacts appear in the object selectors. Periodic contacts do not appear in the object selectors, and therefore cannot be used.

The **Points** folder only appears in the object selection dialog if there is at least one part point already created in the model.

There are two approaches to selecting entities on surfaces:

- Using the Selection Controls. See Selecting Surfaces and Curves with the Selection Controls.
- Using the Graphics Window. See Selecting Surfaces and Curves with the Graphics Window.

Grow Selected

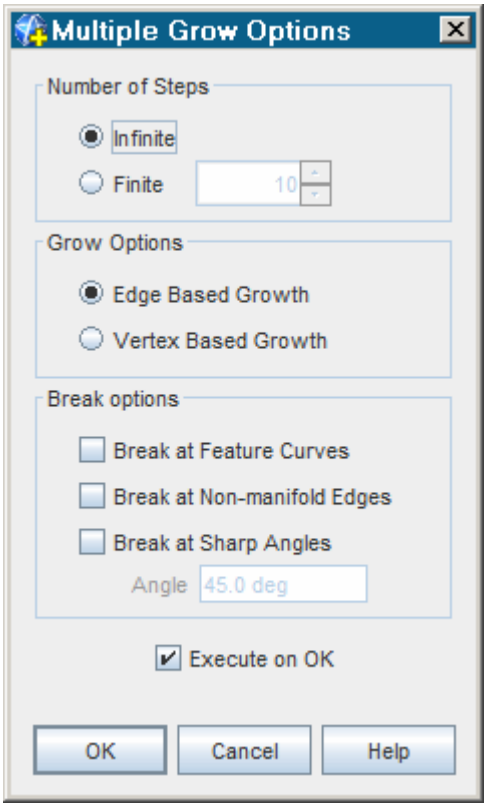
This tool grows the selection by one set of neighboring faces from the pre-selected faces, edges, or vertices. The tool grows from any connected vertices in the selection. This button becomes available when you select at least one face, edge, or vertex.

Shrink Selected

This tool shrinks the pre-selected faces by one neighbor, which is based on the connected vertices of the faces. This button becomes available when you select at least one face.

Multiple Grow Selected

This tool grows multiple sets of neighboring faces from the pre-selected faces, edges, or vertices.



Number of Steps

This group box provides two options:

- **Infinite** grows faces until the selection includes all connected part faces.
- **Finite** grows faces by the supplied finite value, with growth based on connected edges or vertices, depending on the *Grow Options*

Grow Options

The *Grow Options* determine how the selection grows, be it edge-based or vertex-based.

- **Edge based Growth** selects faces with an edge connected to the currently selected faces. The tool does not include faces connected to the surface by a vertex.
- **Vertex based Growth** selects faces with a vertex connected to the currently selected faces. This results in a faster growth of faces per step, and allows selections on another surface that is connected by a single vertex.

Break Options

This group box provides the following options for constraining the growth extent:

- **Break at Feature Curves** breaks the growth of faces at feature curves on the part surface
- **Break at Non-manifold Edges** breaks the growth of faces at non-manifold edges on the part surface
- **Break at Sharp Angles** breaks the growth of faces at sharp angles in the surface, with a user-defined **Angle**

Select Attached Edges

This tool selects the attached edges that are based on the currently displayed faces or vertices.

The *Select Attached Edges Options* dialog configures the behavior of the **Select attached edges** tool.

Options

- **Select All Edges** is the default option, which selects all edges belonging to highlighted faces and vertices.

  Deactivating this option makes the **Select Perimeter Edges** and **Select Feature Edges** options available.
- **Select Perimeter Edges** selects the edges lying on the perimeter of a face or contiguous group of faces.
- **Select Feature Edges** selects attached edges that are marked as feature curves.

  It is possible to choose **Select Perimeter Edges** and **Select Feature Edges** together.

Select Attached Vertices

This tool selects the attached vertices that are based on the selected faces or edges.

The *Select Attached Vertices Options* dialog configures the behavior of the **Select attached vertices** tool. These options are only available if part points are enabled in the simulation, or if the model contains part points.

Options

- **Select All Vertices** selects all vertices belonging to highlighted faces and vertices.
- **Select Feature Vertices** selects attached vertices that are marked as feature curves.

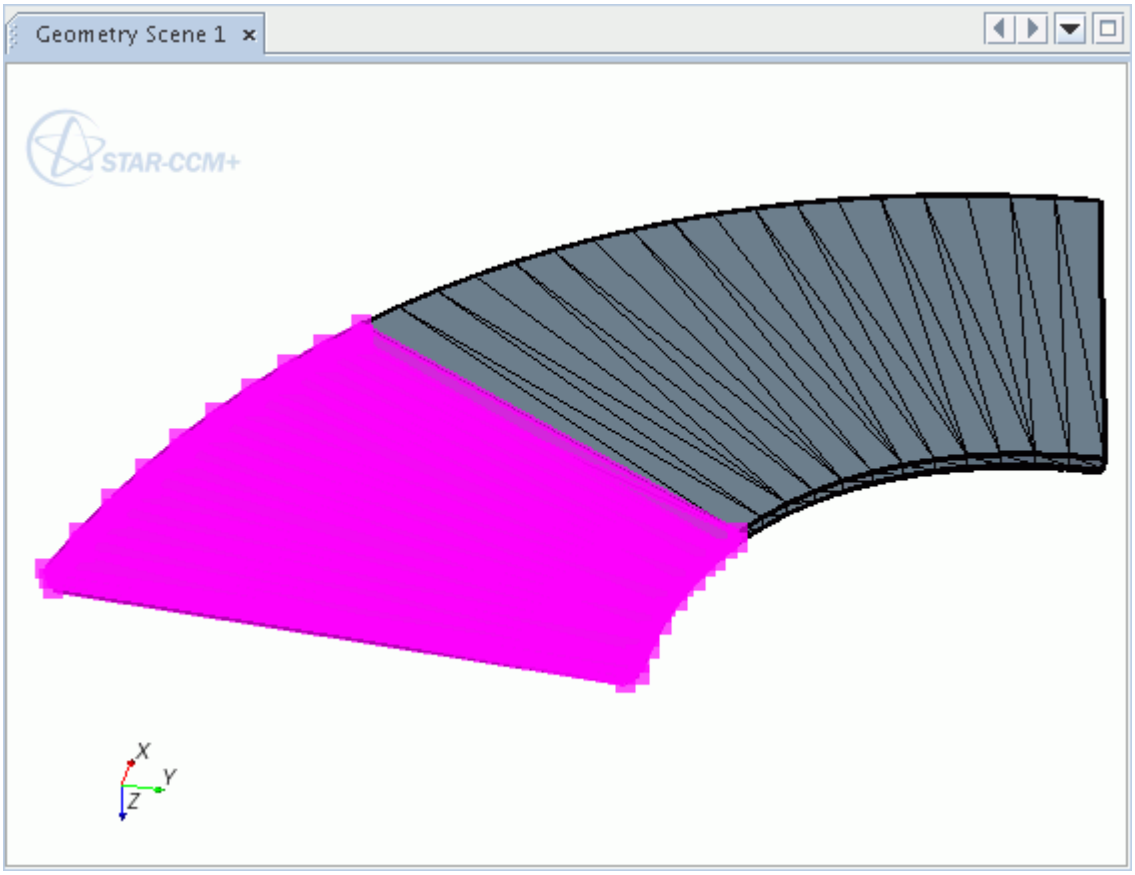
Invert Selection

This tool inverts the current selection of faces, edges, and vertices that are based on the active model (not the displayed model).

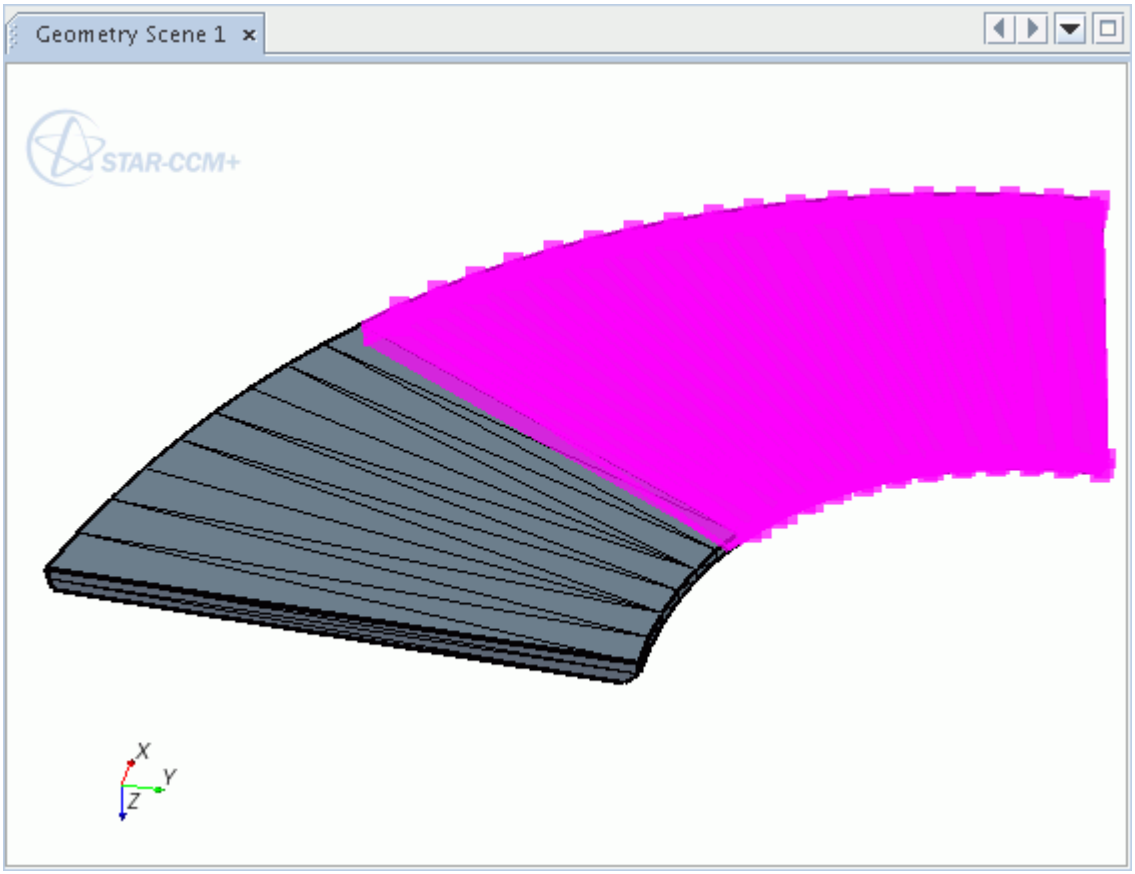
The filter options are taken into account when the inversion is performed. Suppose, for example, that you have selected a number of faces and edges, but only want to invert the selection of faces. You can deactivate the **Edges** filter checkbox, keep the **Faces** filter checkbox activated, and click the **Invert Selection** button. All faces in the active model that were not previously selected are selected, and the faces that were previously selected are deselected.

This tool is available when at least one face, edge, or vertex is selected. If nothing is selected before the inversion, the inverted set is empty.

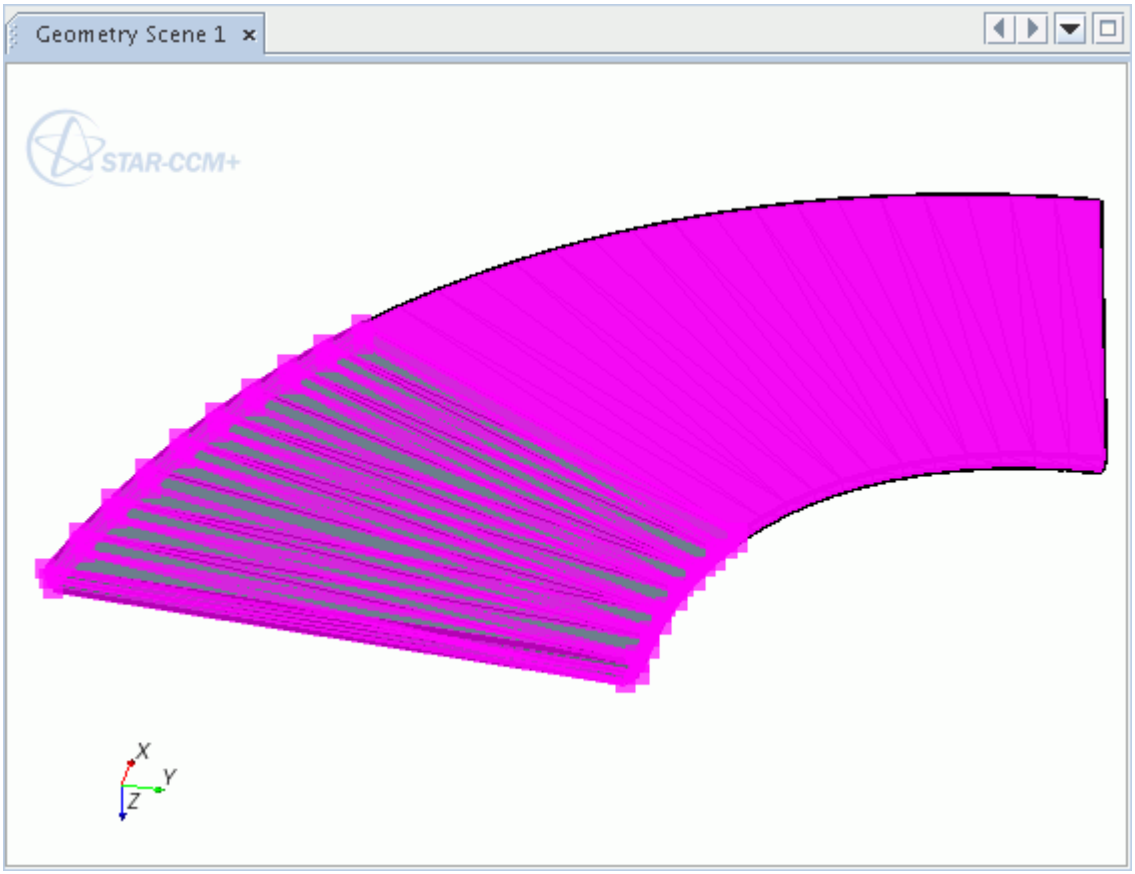
As an example of this operation, consider the following selection that contains faces, edges, and vertices.



If all three filters for **Faces**, **Edges** and **Vertices** are ticked, and the **Invert Selection** button is clicked, the corresponding inverted selection includes faces, edges, and vertices.



If, however, only the **Faces** filter had been ticked, and the filters for **Edges** and **Vertices** had been deactivated, the inverted selection would only contain faces. The rest of the current selection would remain as before.



If all three filters are deactivated, no inversion is performed.

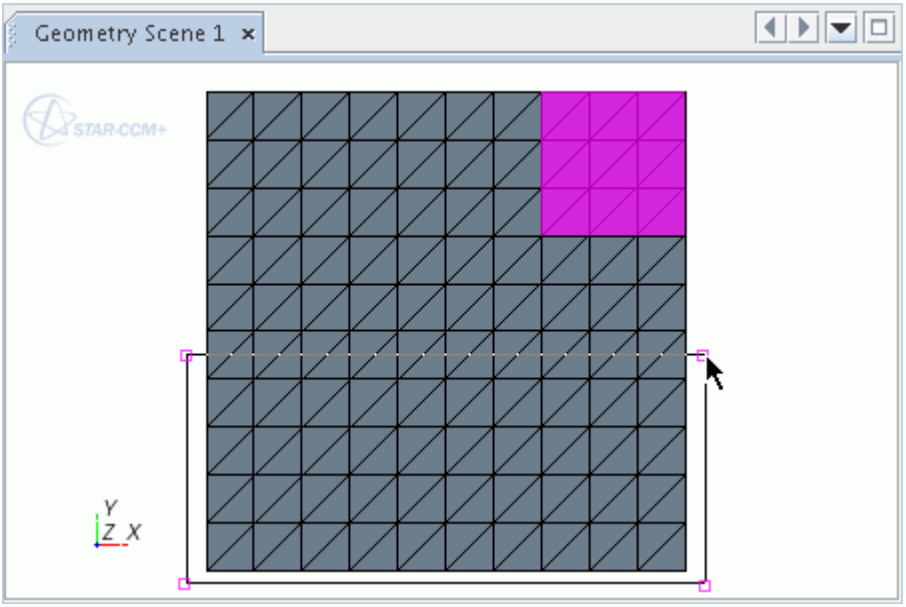
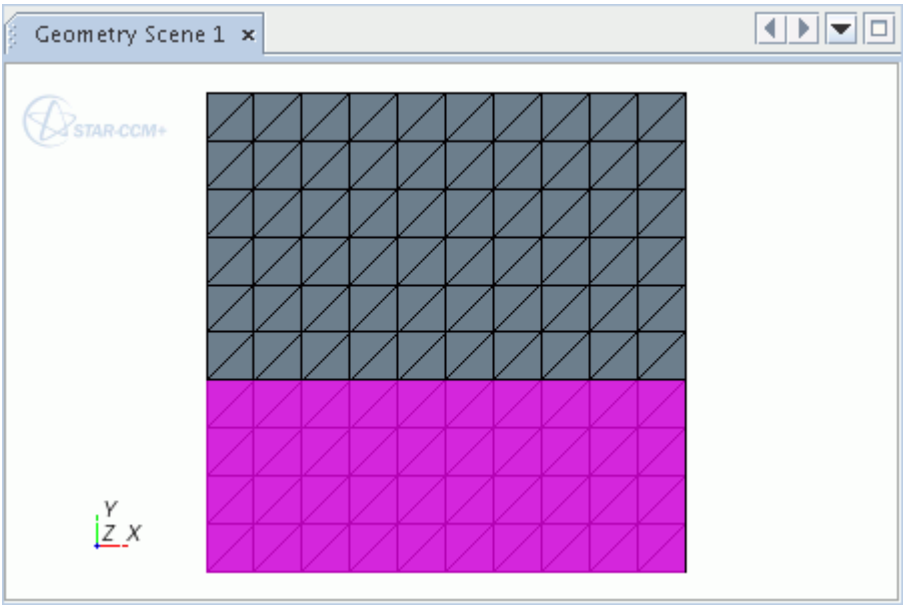
Selection Options

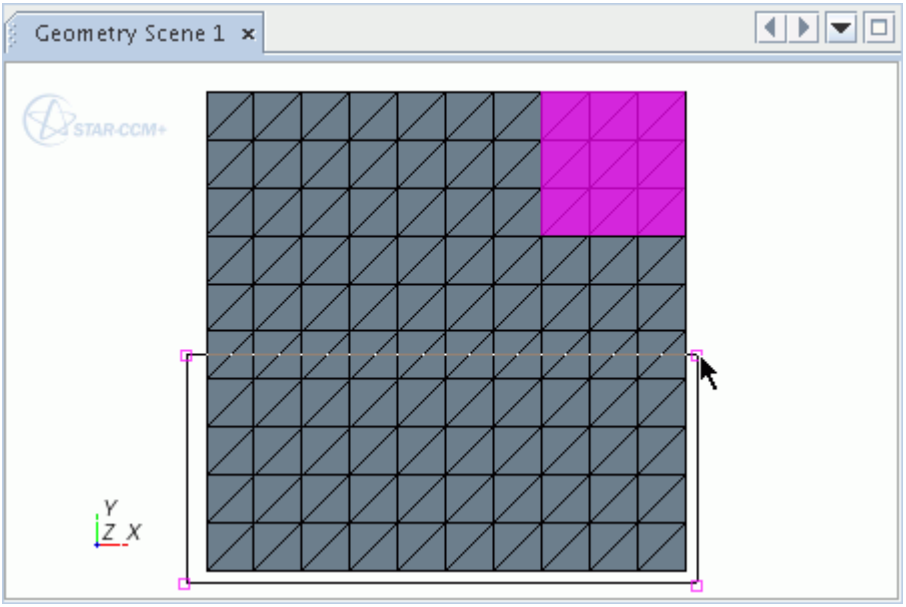
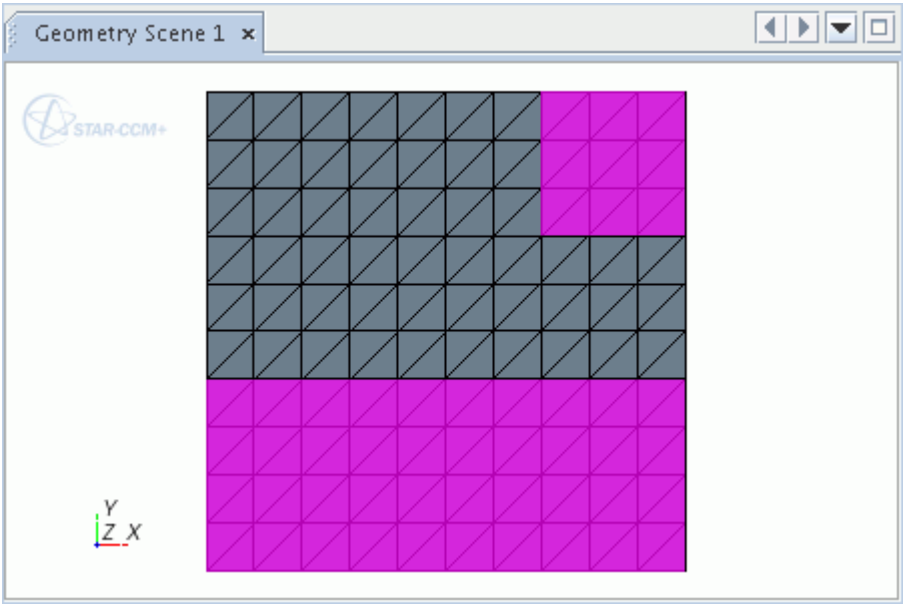
There are four selection options available for use with the zone, threshold, and surface selection tools.

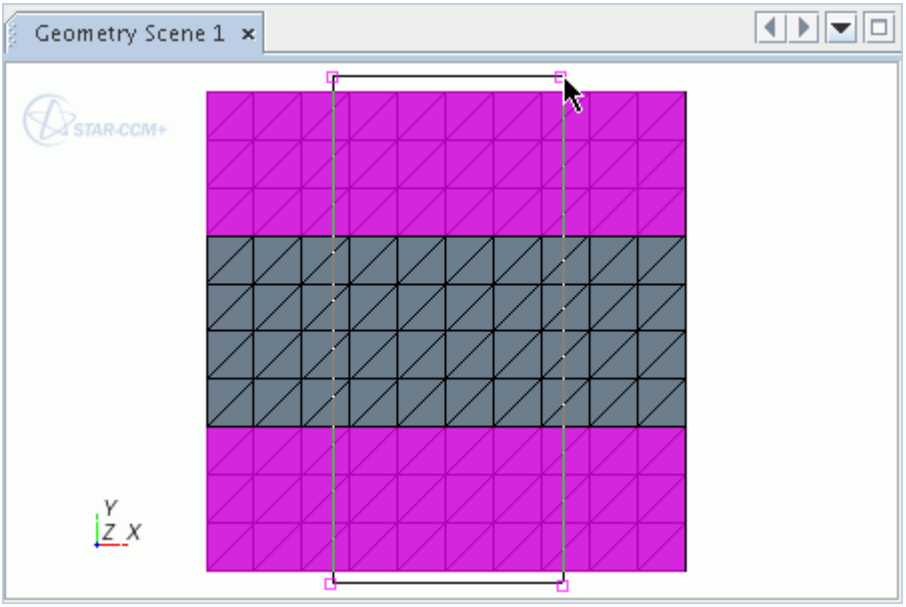
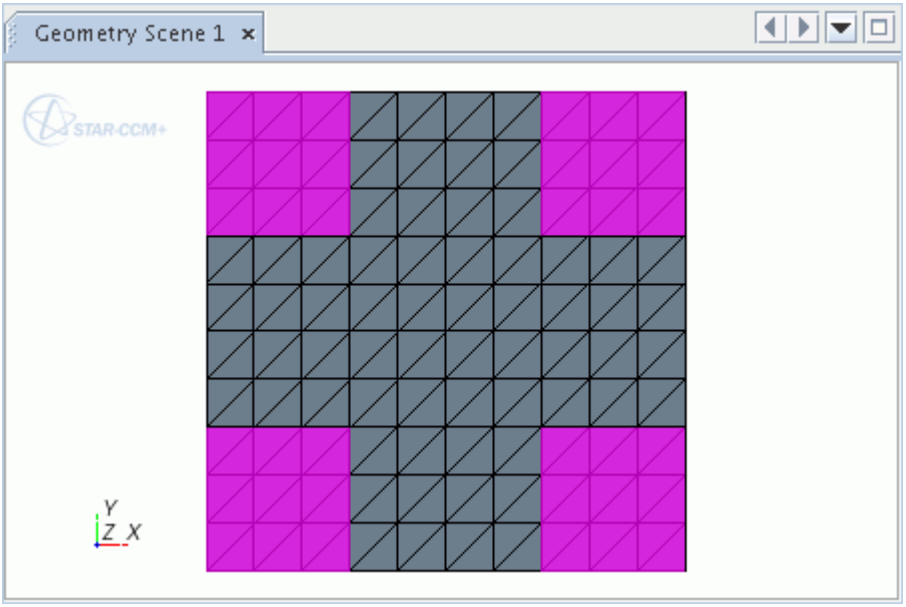
You can access different selection options by clicking the down arrow the icon of each tool.

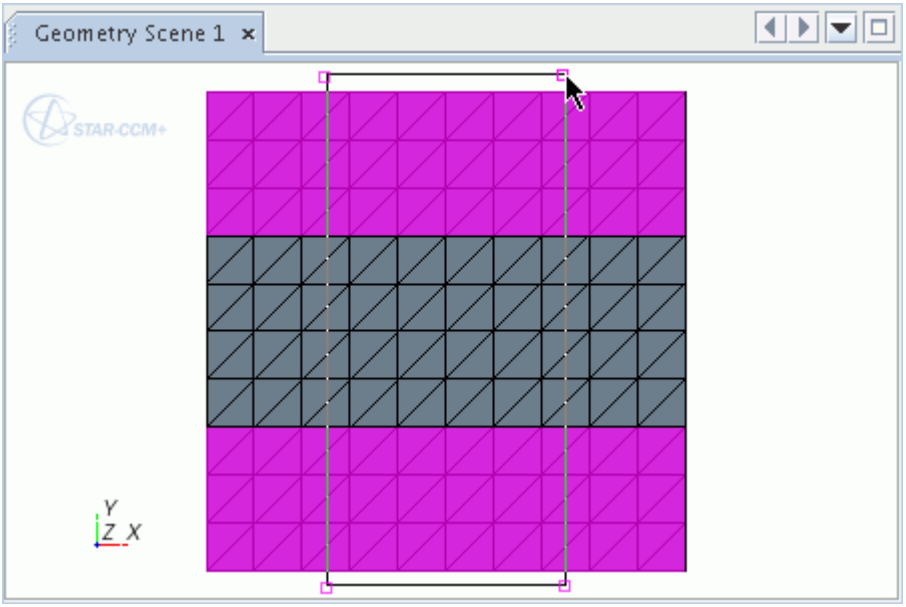
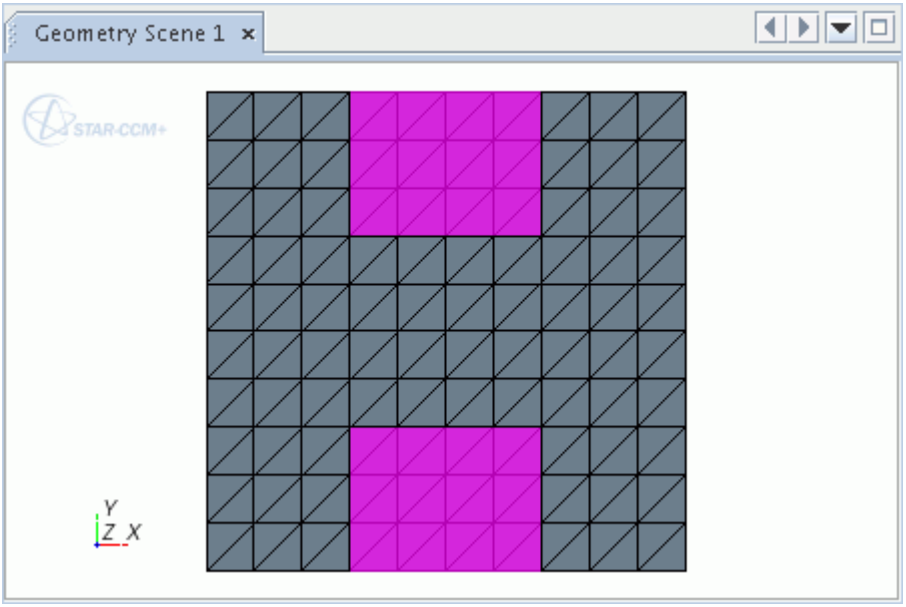
The following examples use the zone selection tool. However, the same applies for the threshold and surface selection tools. When using the *Select Part Surface/Curves* dialog, only edges that have been defined as part curves are selected.

Selection Option	Description
Select	This option is the default selection option. This option deselects any previously selected entities in preference to the new selection.

Selection Option	Description
	 Right-click to close the polygon and select all surfaces within the enclosed area. 
Add Select	This option adds entities to an existing selection.

Selection Option	Description
	 Right-click to close the polygon and add all surfaces within the enclosed area to the existing selection. 
Deselect Zone	This option deselects entities from an existing selection.

Selection Option	Description
	 Right-click to close the polygon and remove all surfaces within the enclosed area from the existing selection. 
Subset Select Zone	This option selects a new group of entities from an existing selection.

Selection Option	Description
	 Right-click to close the polygon and keep the entities within the enclosed area from the existing selection. 

See Also:

Selecting With a Specified Threshold

Selecting Surfaces and Curves with the Selection Controls

Selecting Surfaces Or Curves With The Graphics Window

Using the Select Curves Mode

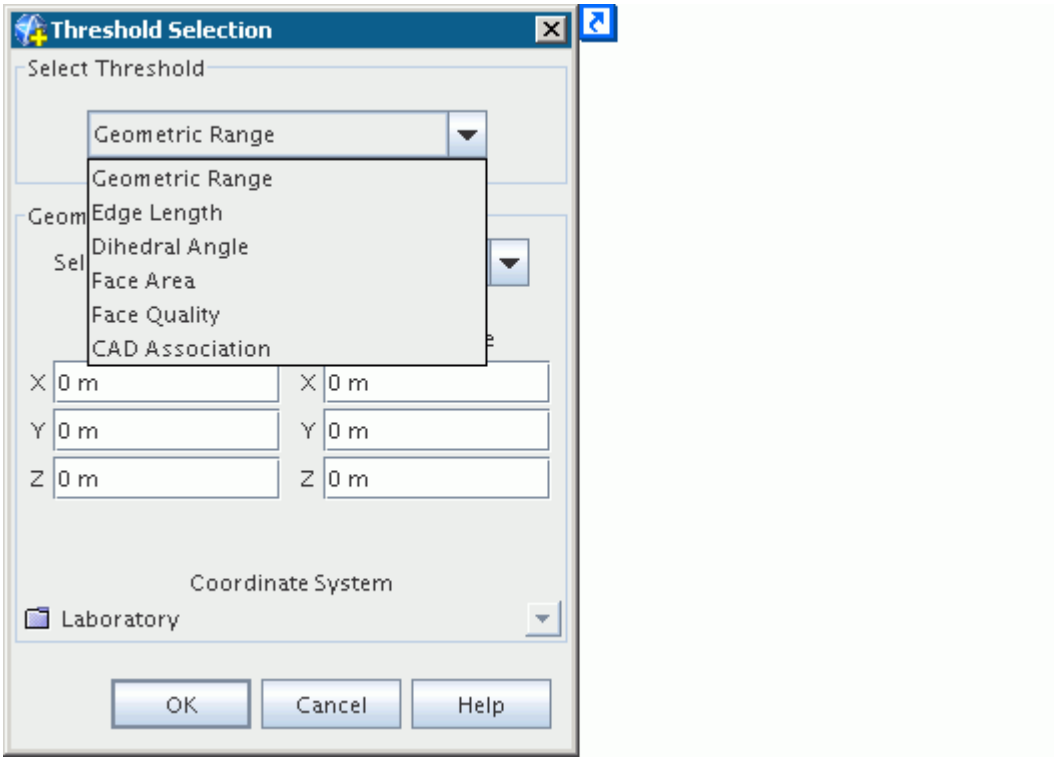
Pop Up Menu of the Surface Repair Tool Graphics Window

Selecting With a Specified Threshold

This tool selects entities on the displayed surface using the *Threshold Selection* dialog.

The filter options control which entities the tool selects.

Using a threshold selection option opens the *Threshold Selection* dialog, where you specify the **Threshold**, and enter **Minimum** and **Maximum** values.



The threshold methods are covered in the following sections.

Contents:

Using The Geometric Range Selection Mode
Using The Edge Length Selection Mode
Using The Dihedral Angle Selection Mode
Using The Face Area Selection Mode
Using The Face Quality Selection Mode
Using The CAD Association Selection Mode

See Also:

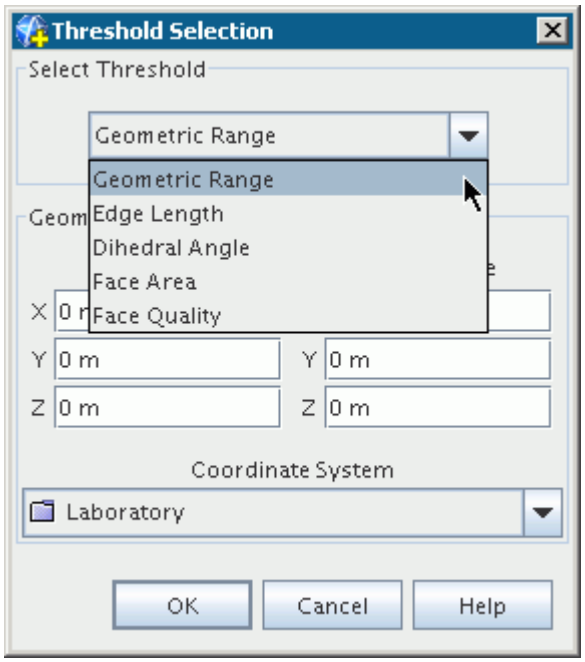
Selection Options
Selecting Entities

Using The Geometric Range Selection Mode

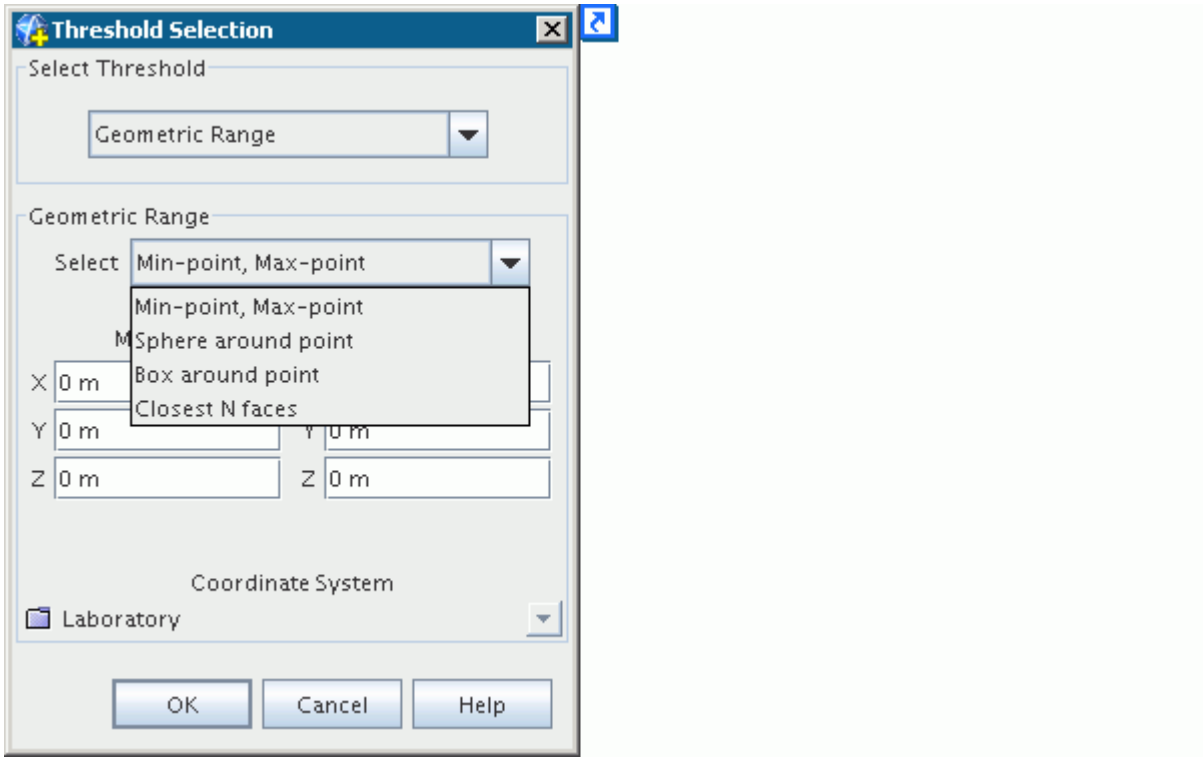
This section describes how to specify a threshold using a geometric range.

To activate the **Geometric Range** threshold:

1. Select the **Geometric Range** option from the **Select Threshold** drop-down menu.

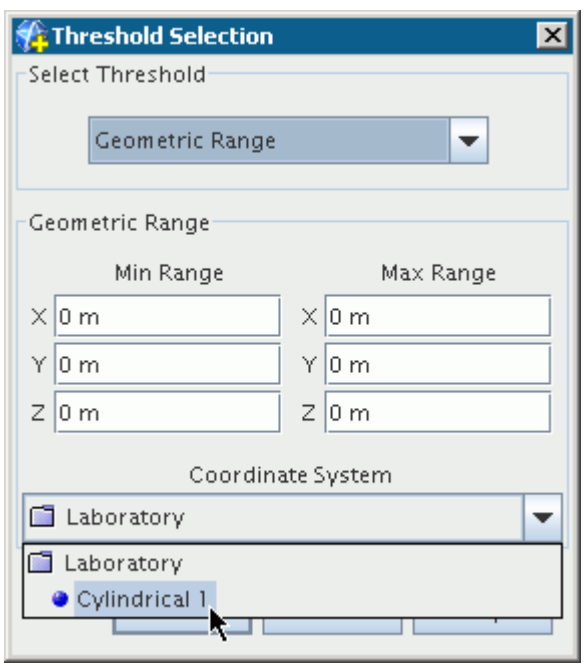


The mode has four selection options available from the **Select** drop-down menu.



The defaults for each selection option use the global laboratory coordinate system. To change the coordinate system:

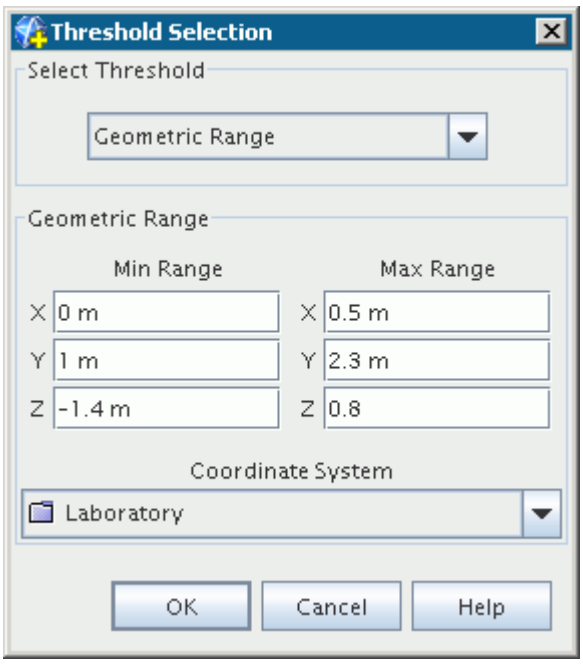
2. Use the drop-down menu that is labeled **Coordinate System**.



All properties that require specification of a point in 3-dimensional space are updated according to the chosen coordinate system. The tool accepts any available length unit, with different unit systems permitted in each field.

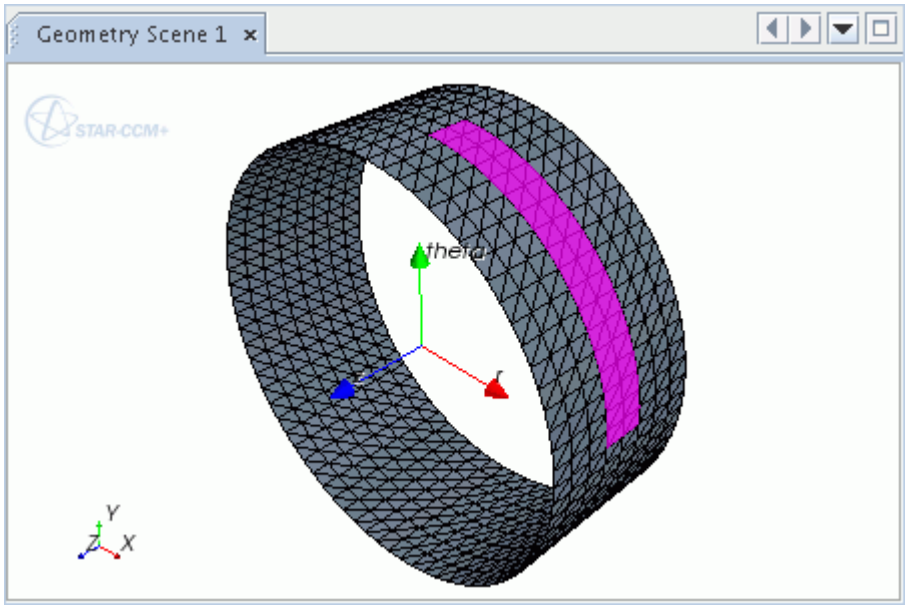
Min-Point, Max-Point

The **Min-Point, Max-Point** option specifies selection using minimum and maximum bounds, for each coordinate direction.

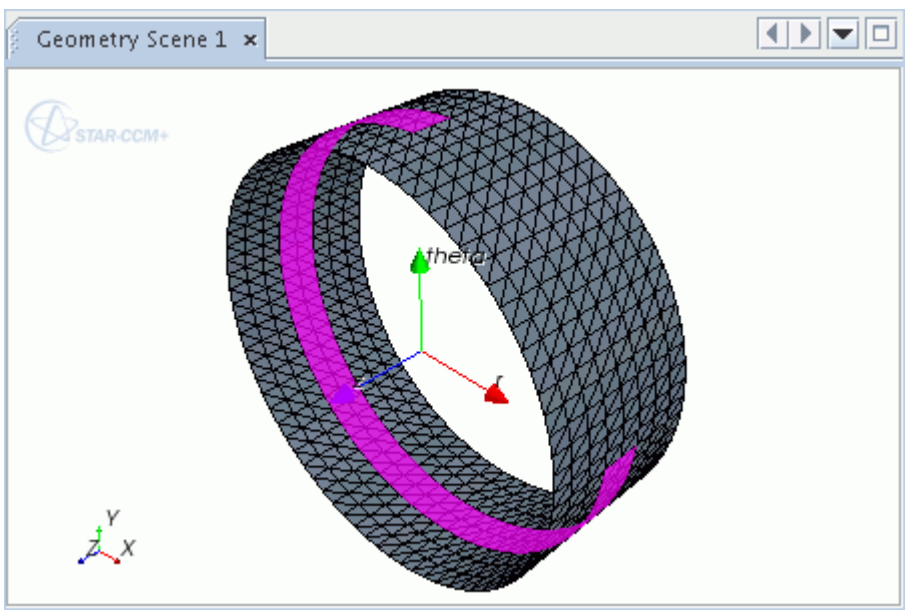


For linear values (X, Y, Z, and R), the order in which you specify the minimum and maximum bounds is not important. That is, a range from `10 mm` to `2 mm` selects the same entities as a range from `2 mm` to `10 mm`. For angular values (θ and φ) in cylindrical or spherical coordinate systems, the order of specification is important. The tool uses the right-hand rule to determine angular values in the positive rotation direction, relative to the axis of rotation. A range from `0 deg` to `90 deg` gives a different selection than a range from `90 deg` to `0 deg`. To illustrate, consider the following examples, which select faces in a cylindrical coordinate system:

- A range in θ from 0 deg (minimum) to 90 deg (maximum) results in the following selection:



- A range in θ from 90 deg (minimum) to 0 deg (maximum) results in a 270-deg selection:



The following table lists the valid range of values for angular coordinates:

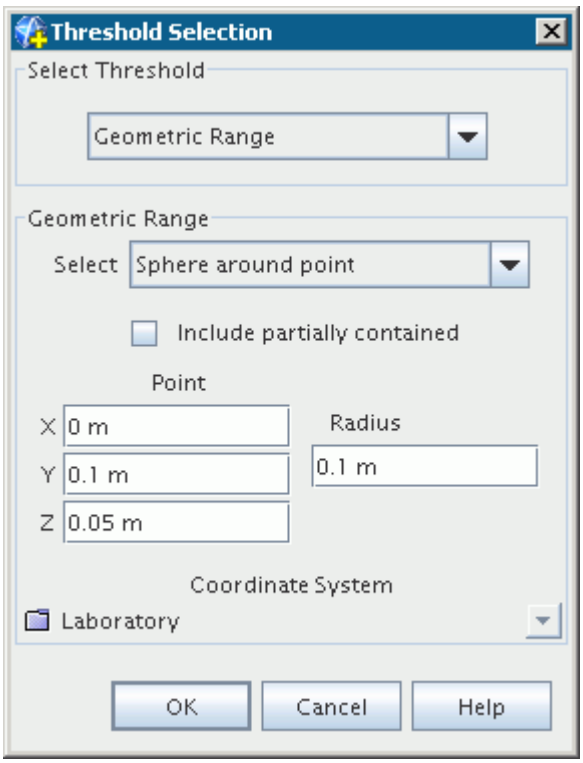
System Type	Coordinate Direction	Valid Range
Cylindrical	θ	$-2\pi \leq \theta \leq 2\pi$
Spherical	θ	$0 \leq \theta \leq \pi$
	φ	$-2\pi \leq \varphi \leq 2\pi$

The minimum and maximum bounds have precise tolerances. If a vertex is marginally out of the geometric range, the selection does not include the associated faces and edges. You can overcome this issue in one of three ways:

3. Specify a slightly larger range, for example, -5 deg to 90 deg, instead of 0 deg to 90 deg.
   This workaround is not possible for θ in a spherical coordinate system, due to the valid range limits identified above.
4. Modify the local coordinate system in a manner that allows the range to include the required entities.
5. Use the Project Vertices tool to modify the location of vertices with respect to the chosen coordinate system.

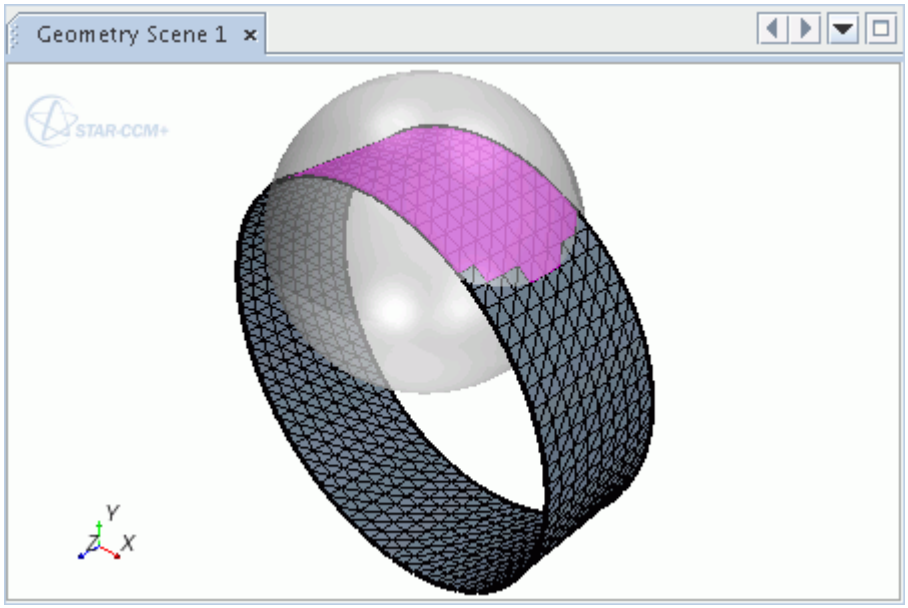
Sphere Around Point

The **Sphere around Point** option selects entities that are included within a spherical threshold, which is defined by a point and radius.



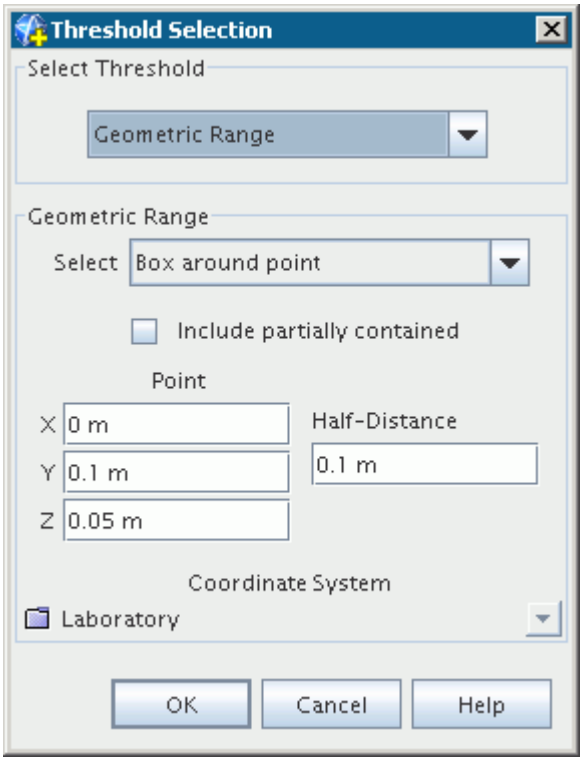
6. Activating the **Include partially contained** option selects entities if any part of the entity is within the spherical threshold. If this option is deactivated, to tool only selects entities that are contained within the threshold.
7. The *Point* property specifies the centerpoint of the spherical threshold. The *Radius* specifies the radius, and hence extent of the spherical threshold.

   Provided below is an example of the spherical threshold selection option. The transparent sphere does not appear in the *Graphics* window; it has been added for illustrative purposes.



Box Around Point

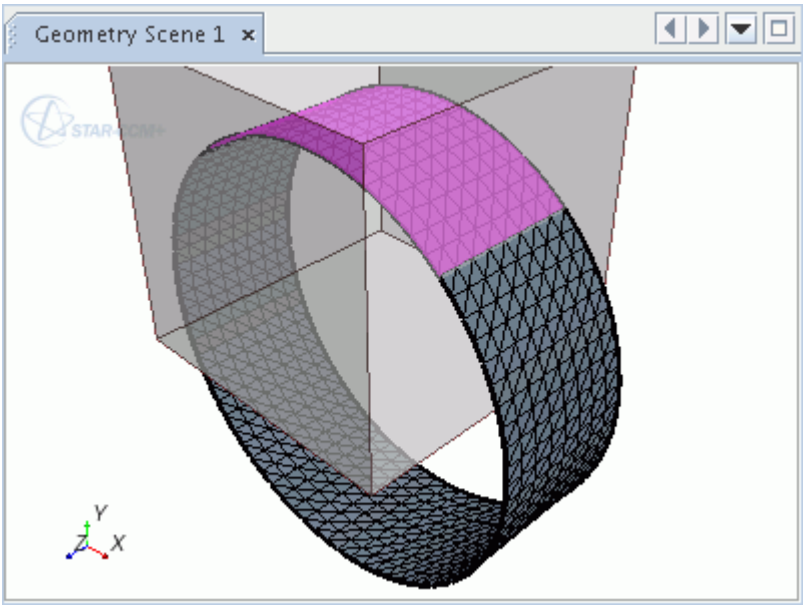
The **Box around Point** option selects entities that are included within a box threshold, which is defined by a point and half-distance.



8. Activating the **Include partially contained** option selects entities if any part of the entity is within the box threshold. If this option is deactivated, the tool only selects entities that are contained within the threshold.

The *Point* property specifies the centerpoint of the box threshold. The *Half-distance* specifies the extent of the threshold in each cartesian direction (x, y, and z), from the centerpoint. Therefore, a cube is always used as the threshold is defined by the same distance in all directions.

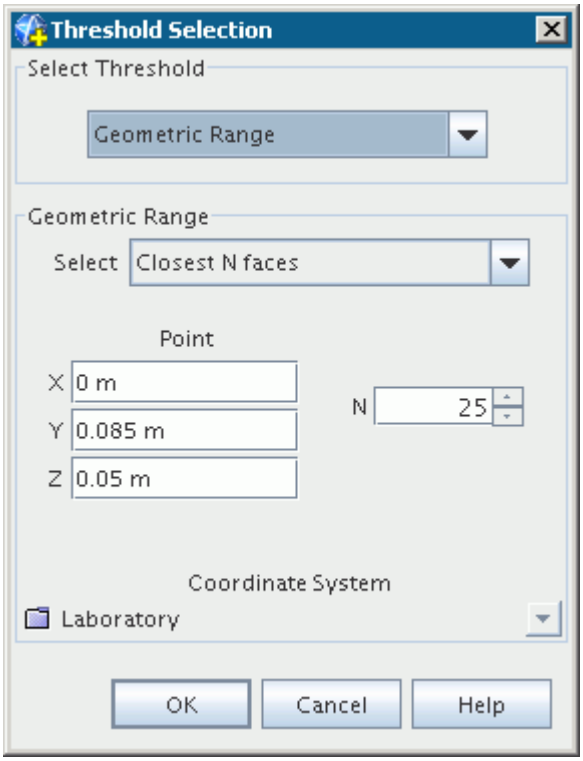
Provided below is an example of the box threshold selection option. The transparent box does not appear in the *Graphics* window; it has been added for illustrative purposes.



Closest N Faces

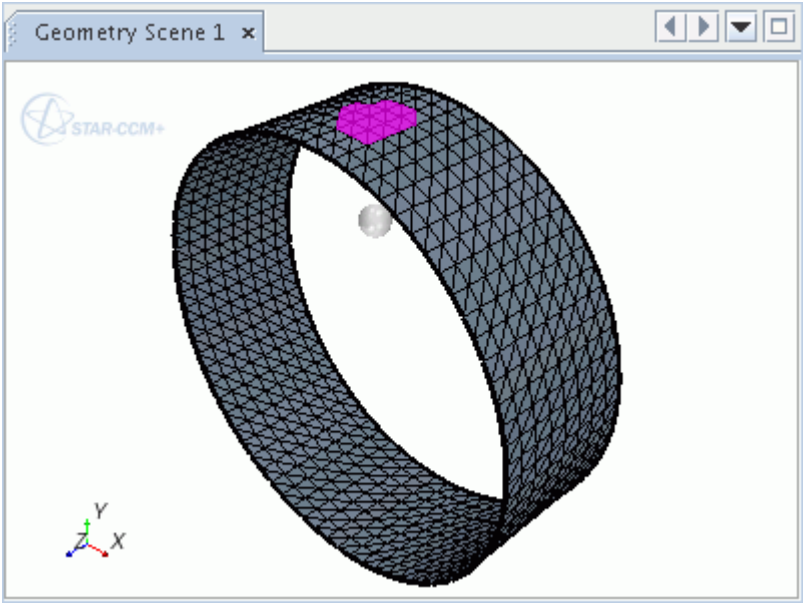
The **Closest N faces** option selects the closest user-defined amount of faces to a specified point. To determine the closest faces, the code first determines the closest point on each face to the user-defined point and finds the distance between these points.

The Closest N faces are then identified as the first N faces with the shortest distance.



9. The *Point* property specifies the reference of the threshold. The *N* property specifies the number of selected faces closest to the specified point.

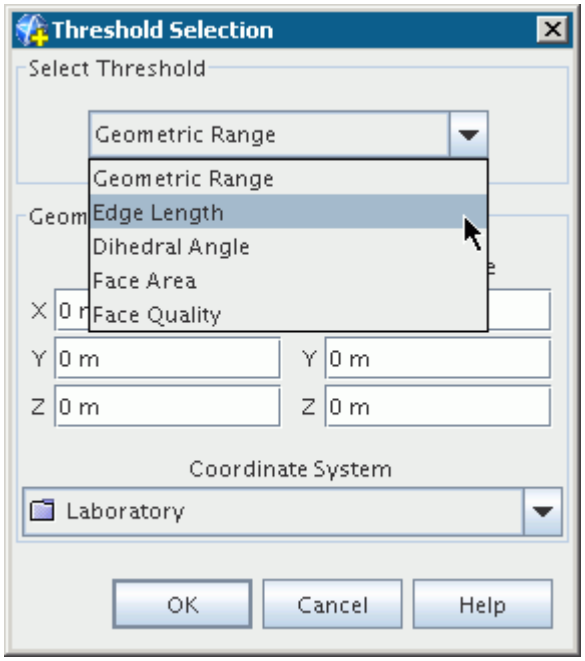
   Provided below is an example of the **Closest N faces** selection option. This example shows the closest 25 faces to the user-defined reference point. The sphere representing the point does not appear in the *Graphics* window; it has been added for illustrative purposes.



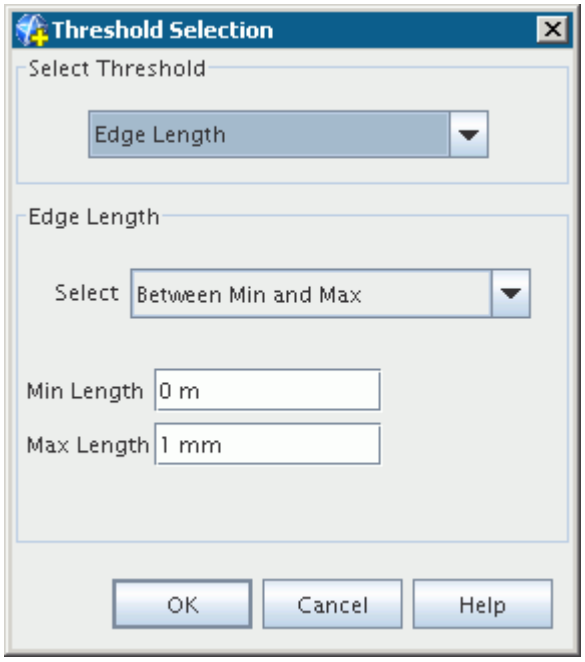
Using The Edge Length Selection Mode

The **Edge Length** threshold selects edges according to length, which is based on a user-defined range.

1. To activate the **Edge Length**selection mode, select the **Edge Length** option from the **Select Threshold** drop-down menu.



2. When using the **Edge Length** threshold, you specify the minimum and maximum edge lengths in the *Threshold Selection* dialog.



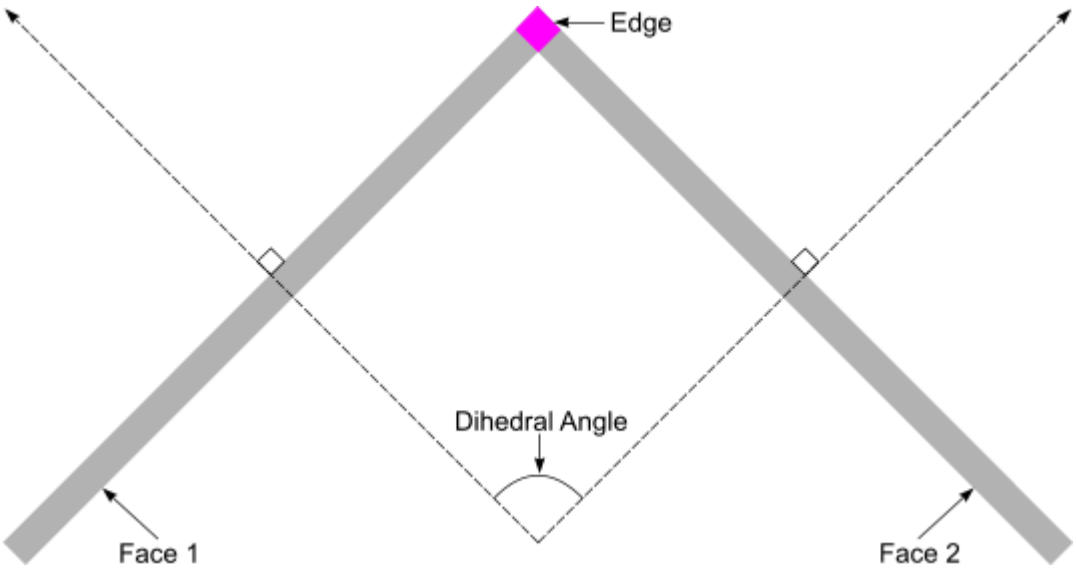
The tool accepts any available length unit, with different unit systems permitted in the minimum and maximum fields.

Four options are available from the **Select** drop-down menu, which alter the range definition.

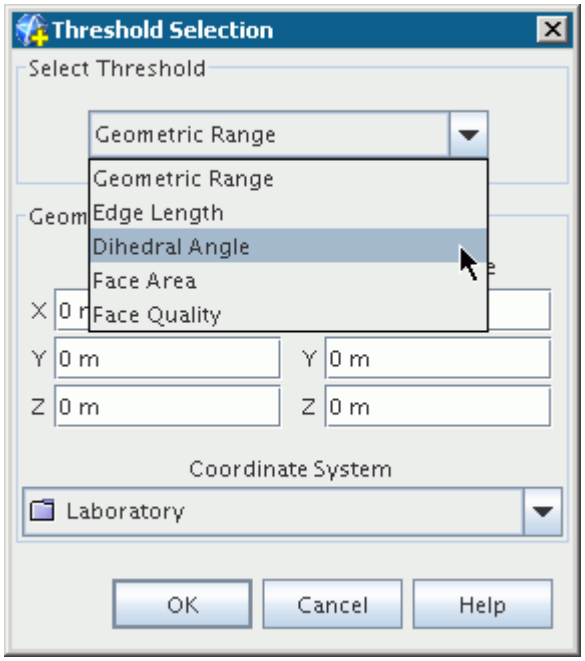
- **Between Min and Max** - Selected edges have a length between the user-defined **Min Length** and **Max Length**.
- **Above Min** - Selected edges have a length above the user-defined **Min Length**, with the **Max Length** field deactivated.
- **Below Max** - Selected edges have a length below the user-defined **Max Length**, with the **Min Length** field deactivated.
- **Below Min and Above Max** - Selected edges have a length below the user-defined **Min Length** or above the **Max Length**.

Using The Dihedral Angle Selection Mode

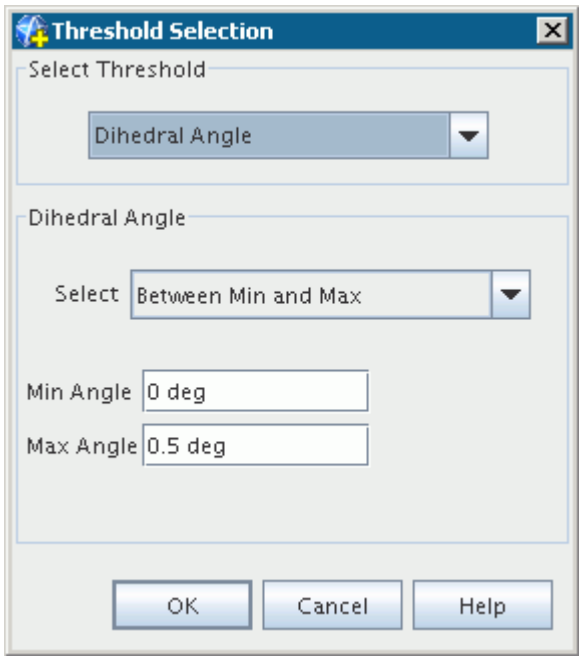
The **Dihedral Angle** threshold selects edges according to the angle between the face normals of the faces that are connected to the edge. You can select edges that have three or more faces attached.



1. To activate the **Dihedral Angle** selection mode, select the **Dihedral Angle** option from the **Select Threshold** drop-down menu.



2. When using the **Dihedral Angle** threshold, you specify the minimum and maximum angles in the *Threshold Selection* dialog.



The tool accepts any available angle unit, with different unit systems permitted in the minimum and maximum fields.

Four options are available from the **Select** drop-down menu, which alter the range definition.

- **Between Min and Max** - Selected edges have a dihedral angle between the user-defined **Min Angle** and **Max Angle**.

- **Above Min** - Selected edges have a dihedral angle above the user-defined **Min Angle**, with the **Max Angle** field deactivated.
- **Below Max** - Selected edges have a dihedral angle below the user-defined **Max Angle**, with the **Min Angle** field deactivated.
- **Below Min and Above Max** - Selected edges have a dihedral angle below the user-defined **Min Angle** or above the **Max Angle**.

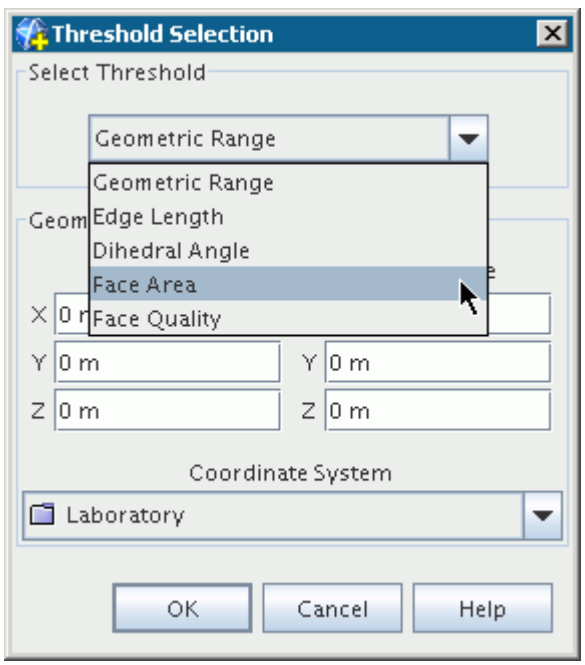
See Also:

Using the Chase Edge Selection Mode

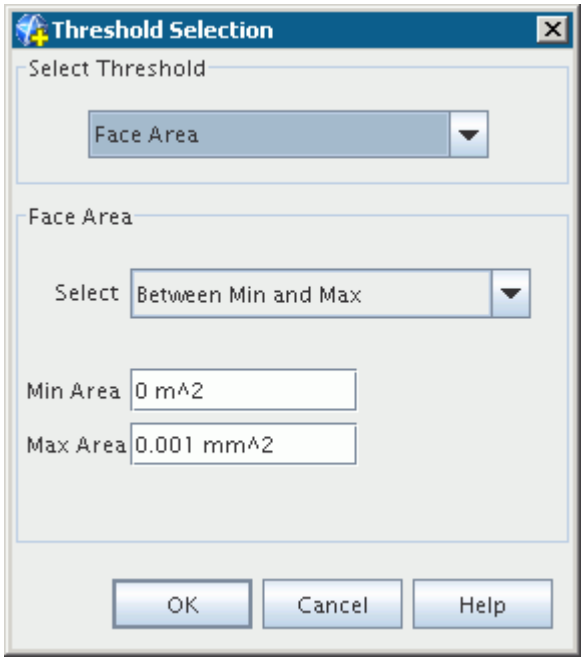
Using The Face Area Selection Mode

The **Face Area** threshold selects faces according to area, which is based on a user-defined range.

1. To activate the **Face Area** selection mode, select the **Face Area** option from the **Select Threshold** drop-down menu.



2. When using the **Face Area** threshold, you specify the minimum and maximum face areas in the *Threshold Selection* dialog.



The tool accepts any available area unit, with different unit systems permitted in the minimum and maximum fields.

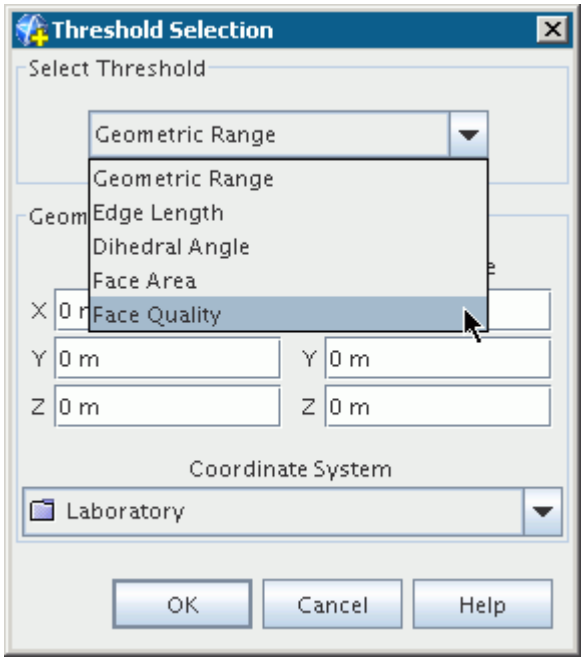
Four options are available from the **Select** drop-down menu, which alter the range definition.

- **Between Min and Max** - Selected faces have an area between the user-defined **Min Area** and **Max Area**.
- **Above Min** - Selected faces have an area above the user-defined **Min Area**, with the **Max Area** field deactivated.
- **Below Max** - Selected faces have an area below the user-defined **Max Area**, with the **Min Area** field deactivated.
- **Below Min and Above Max** - Selected faces have an area below the user-defined **Min Area** or above the **Max Area**.

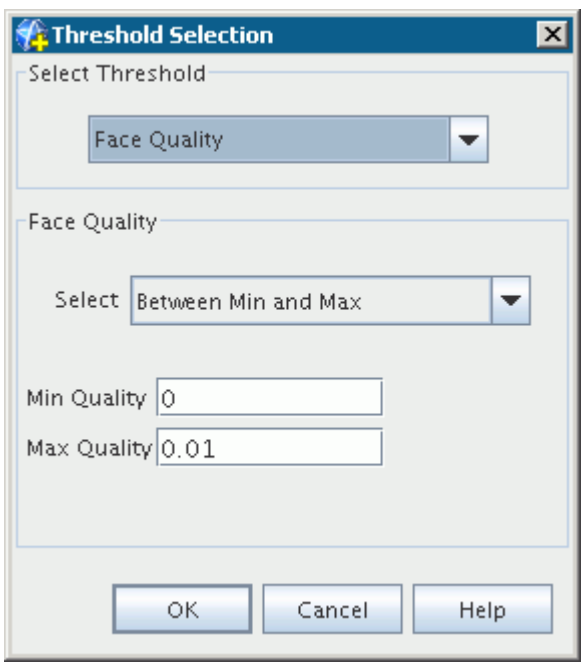
Using The Face Quality Selection Mode

The **Face Quality** threshold selects faces according to face quality, which is based on a user-defined range.

1. To activate the **Face Quality** selection mode, select the **Face Quality** option from the **Select Threshold** drop-down menu.



2. When using the **Face Quality** threshold, you specify the minimum and maximum face quality in the *Threshold Selection* dialog.



Four options are available from the **Select** drop-down menu, which alter the range definition.

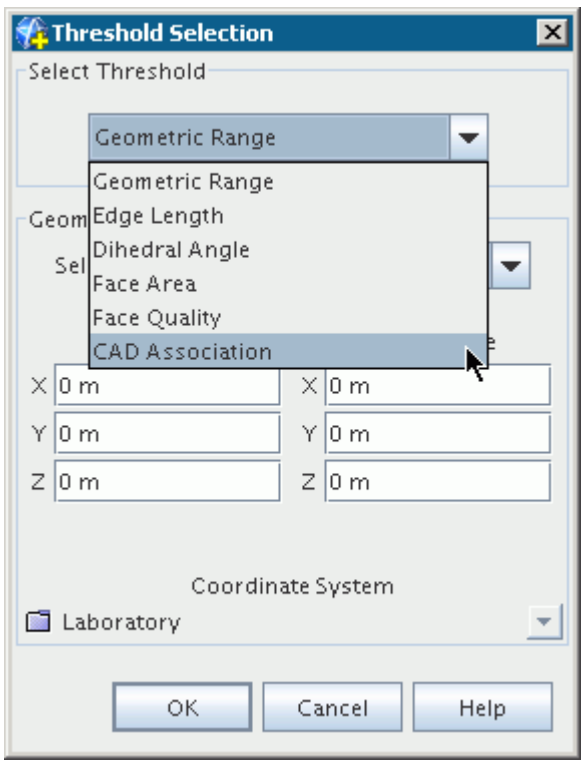
- **Between Min and Max** - Selected faces have a face quality between the user-defined **Min Quality** and **Max Quality**.
- **Above Min** - Selected faces have a face quality above the user-defined **Min Quality**, with the **Max Quality** field deactivated.

- **Below Max** - Selected faces have a face quality below the user-defined **Max Quality**, with the **Min Quality** field deactivated.
- **Below Min and Above Max** - Selected faces have face quality below the user-defined **Min Quality** or above the **Max Quality**.

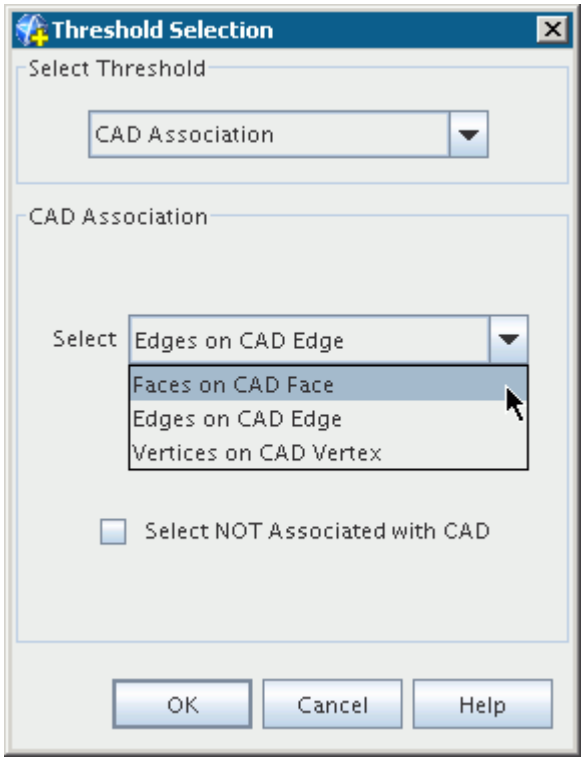
Using The CAD Association Selection Mode

The **CAD Association** threshold selects faces, edges, or vertices that are based upon CAD association.

1. To activate the **CAD Association** selection mode, select the **CAD Association** option from the **Select Threshold** drop-down menu.

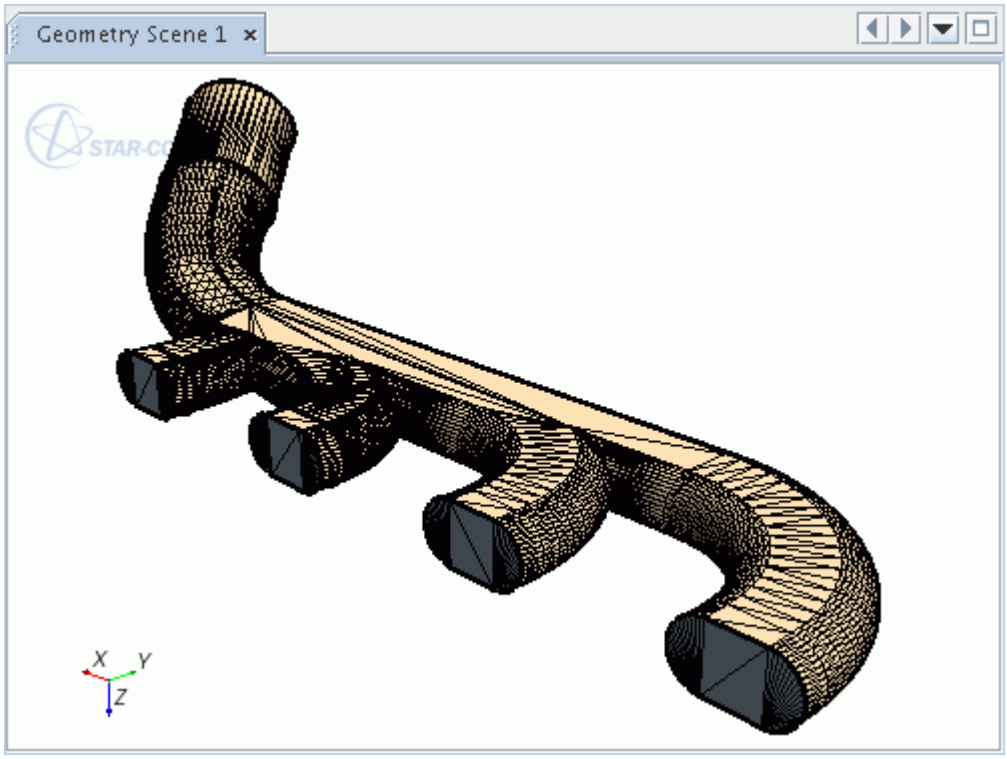


2. Three specification options are available from the **Select** drop-down menu:
   - **Faces on CAD Face** selects all faces that have association with CAD faces. A CAD face is a surface on the part that is defined in the CAD model.
   - **Edges on CAD Edge** selects all edges that have association with CAD edges. A CAD edge is the perimeter of a CAD face. This option is useful for identifying which edges are used with the aligned meshing and geometric features when using the surface remesher.
   - **Vertices on CAD Vertex** selects all vertices that have association CAD vertices. A CAD vertex is the point where two CAD edges intersect.

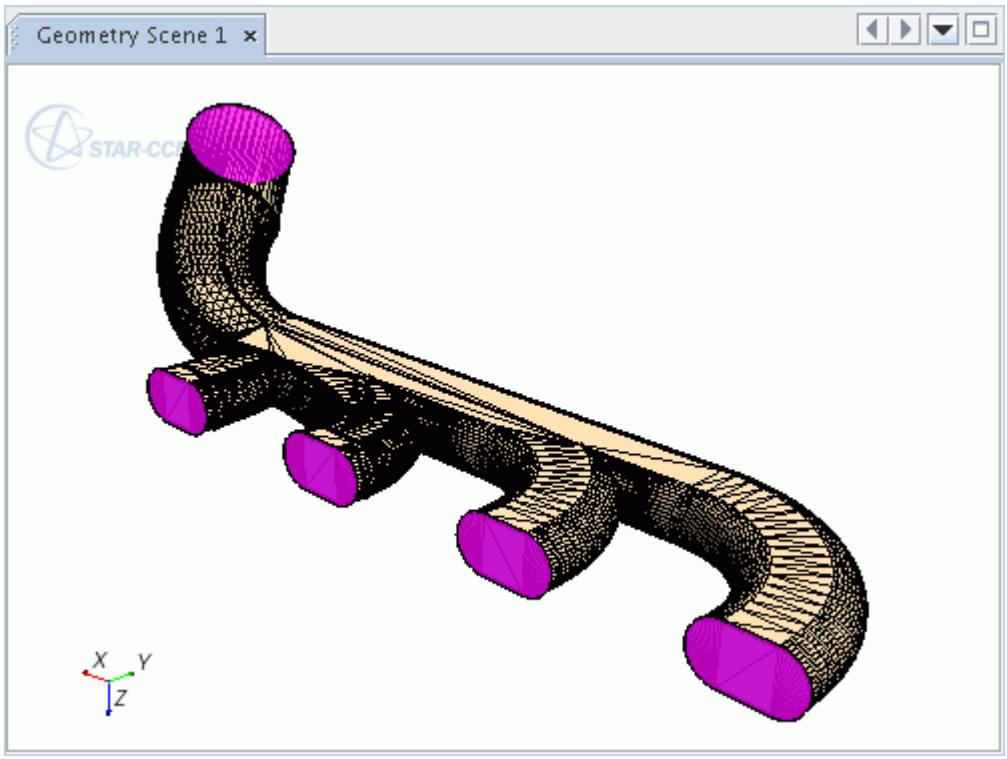


Activating the **Select NOT Associated with CAD** option has the opposite effect and selects face, edges, or vertices that do not have CAD association.

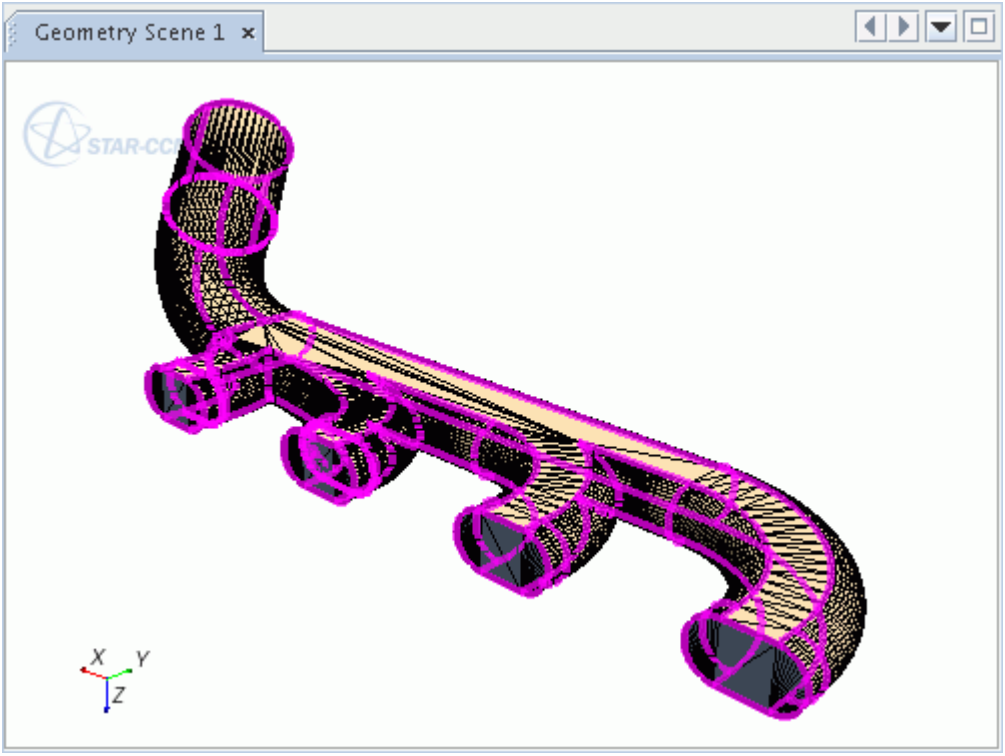
As an example, consider the following surface, which has faces with CAD association highlighted in beige using the CAD association display option:



Using the **Faces on CAD Face** option and ticking the checkbox to select faces that do not have CAD association has the following effect:



The following image shows edges that have CAD association, which is used with the aligned meshing and geometric features:

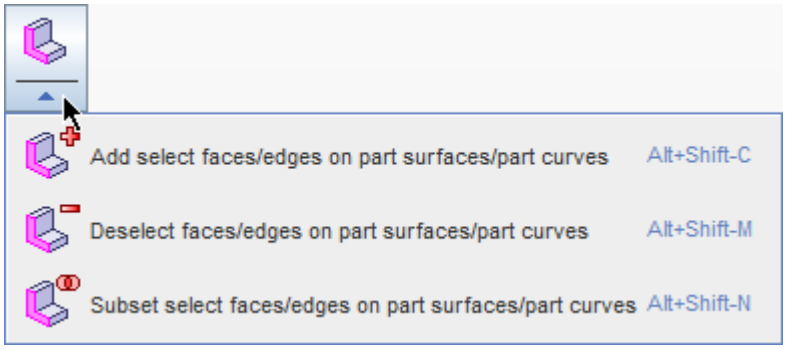


Selecting Surfaces and Curves with the Selection Controls

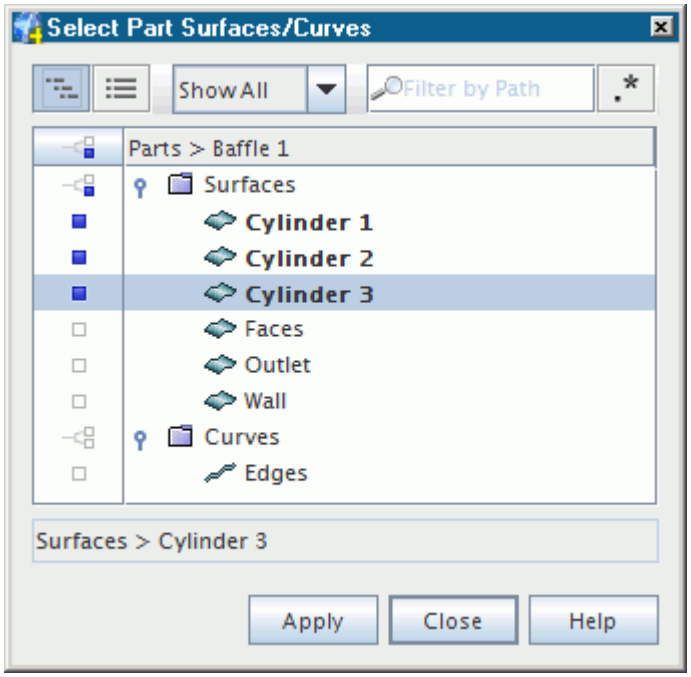
Use this method of selecting entities to pick surfaces from the object selection dialog.

To select surfaces and curves with the selection controls:

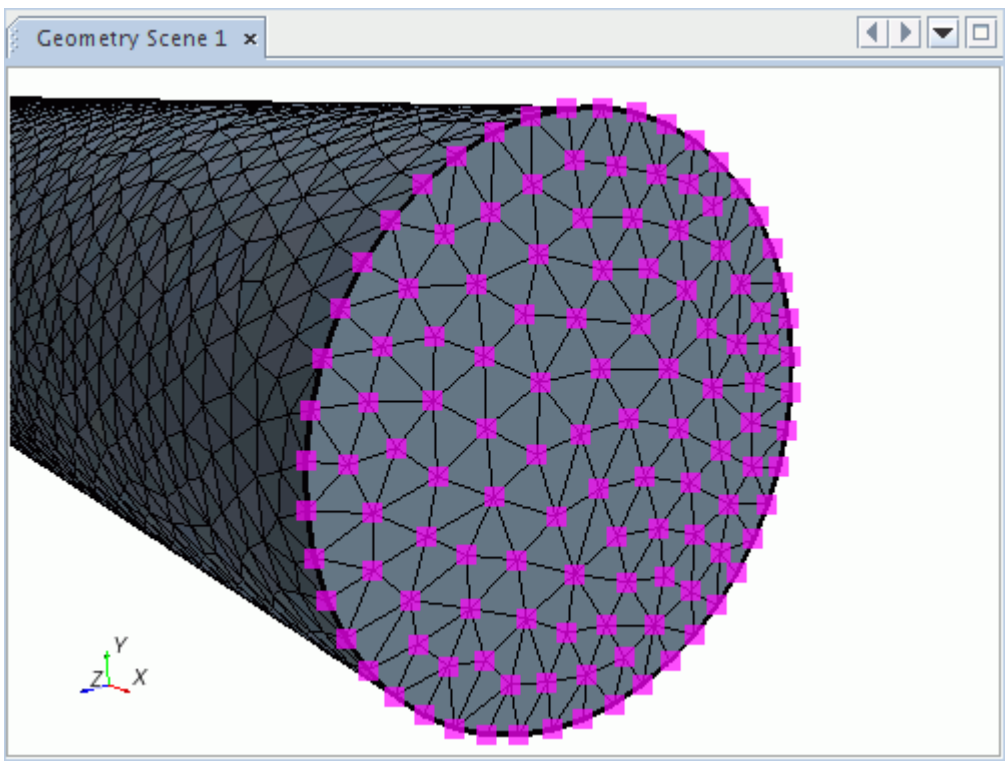
1. Select the appropriate entity filters which define the selection of associated faces, edges, and/or vertices. This action is not necessary if you are selecting part curves, as only edges are available.
2.
3. Click (**Select Face/Edges on Part Surfaces/Part Curves**) or the click the down arrow below this tool and select one of the options.



4. In the *Select Part Surface/Curves* dialog, select the required items and click **Apply**.



For example, selecting this outlet boundary with only the Vertices filter activated has the following result.



See Also:

Selection Options

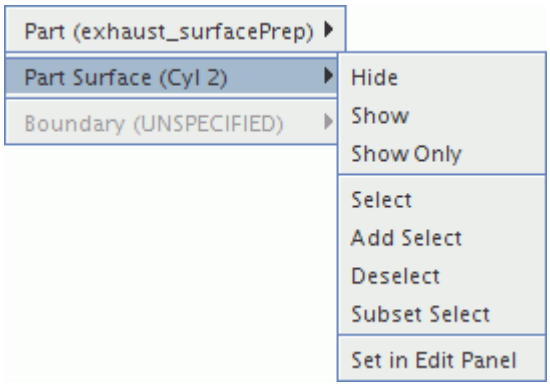
Selecting Entities

Selecting Surfaces Or Curves With The Graphics Window

This method of selecting entities allows you to pick surfaces and curves by right-clicking in the *Graphics* window.

To select surfaces or curves with the *Graphics* window:

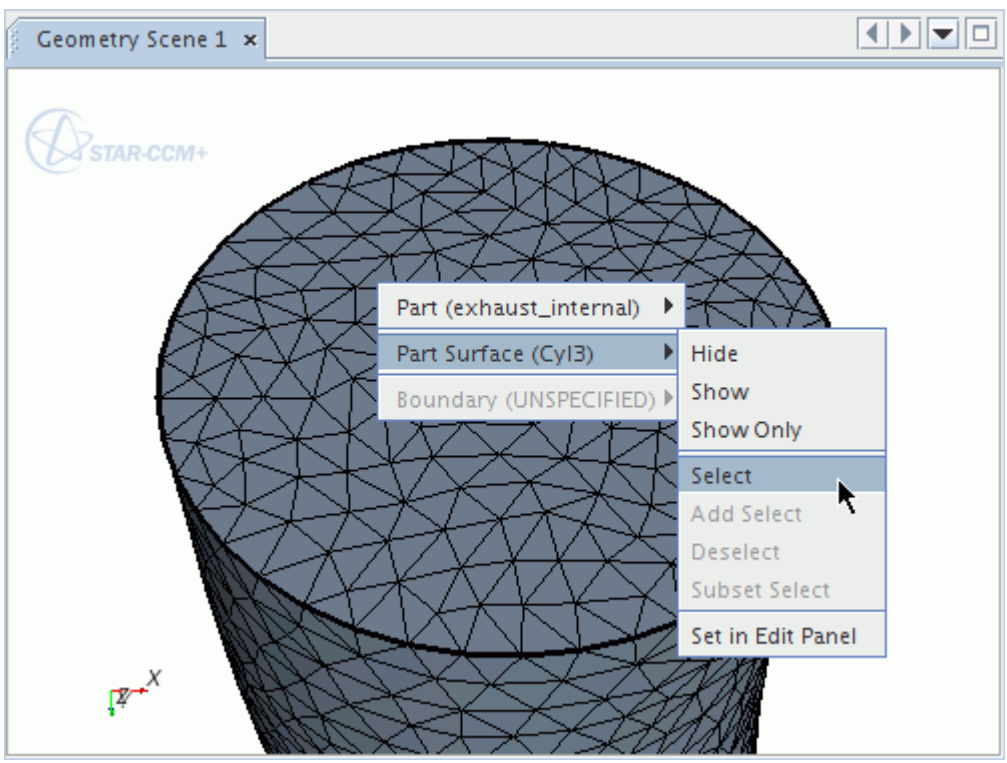
1. Select the appropriate entity filters which define the selection of associated faces, edges, and/or vertices.
2. Right-click the required surface in the *Graphics* window and select one of the selection options.
   If no entities are currently selected, only the **Select** option is available.



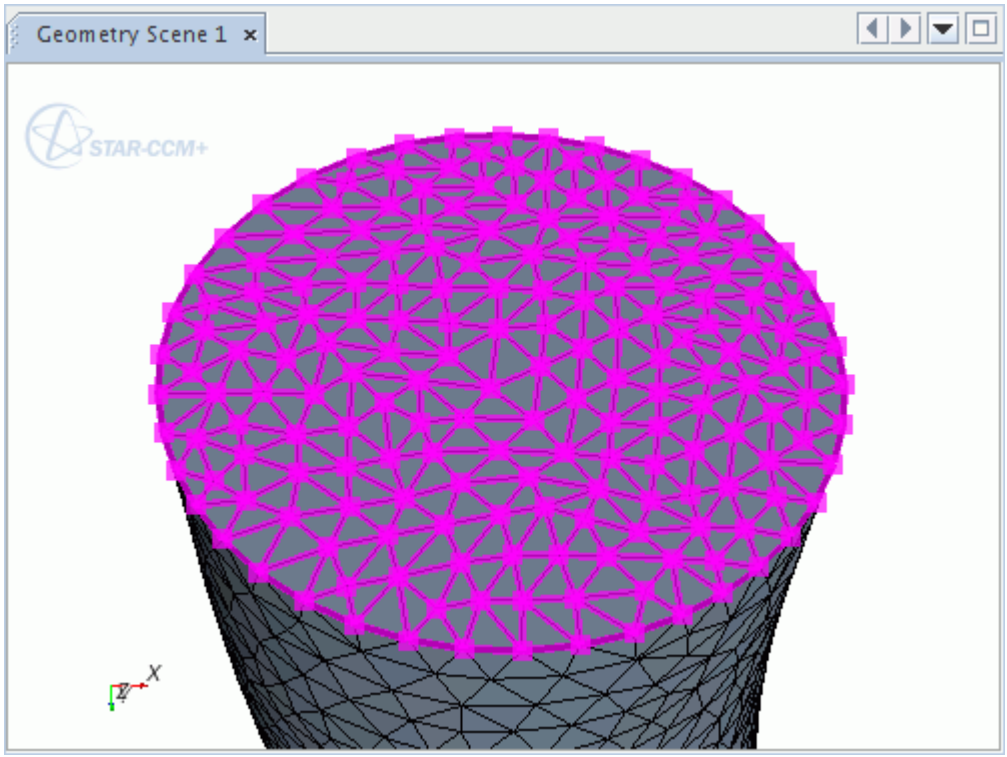
Similarly, you can use the same selection options on feature curves.

Example of Selecting Entities on a Surface

The following image shows a surface for which the tool selects associated entities.

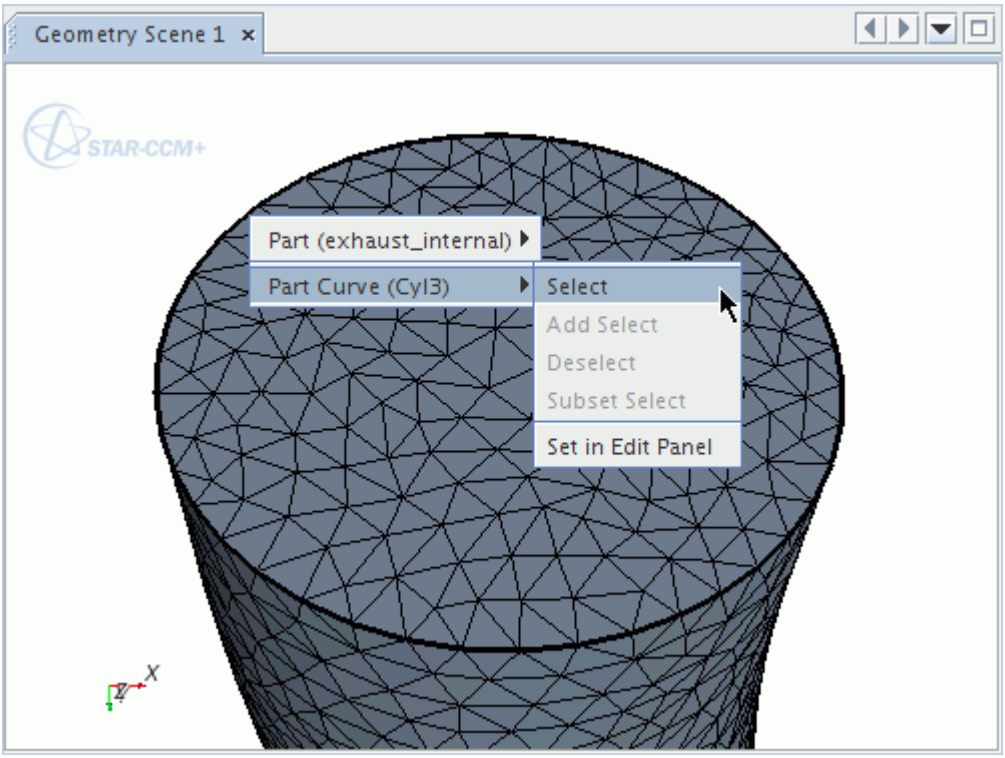


With the **Edges** and **Vertices** filters activated, the tool selects the following entities.

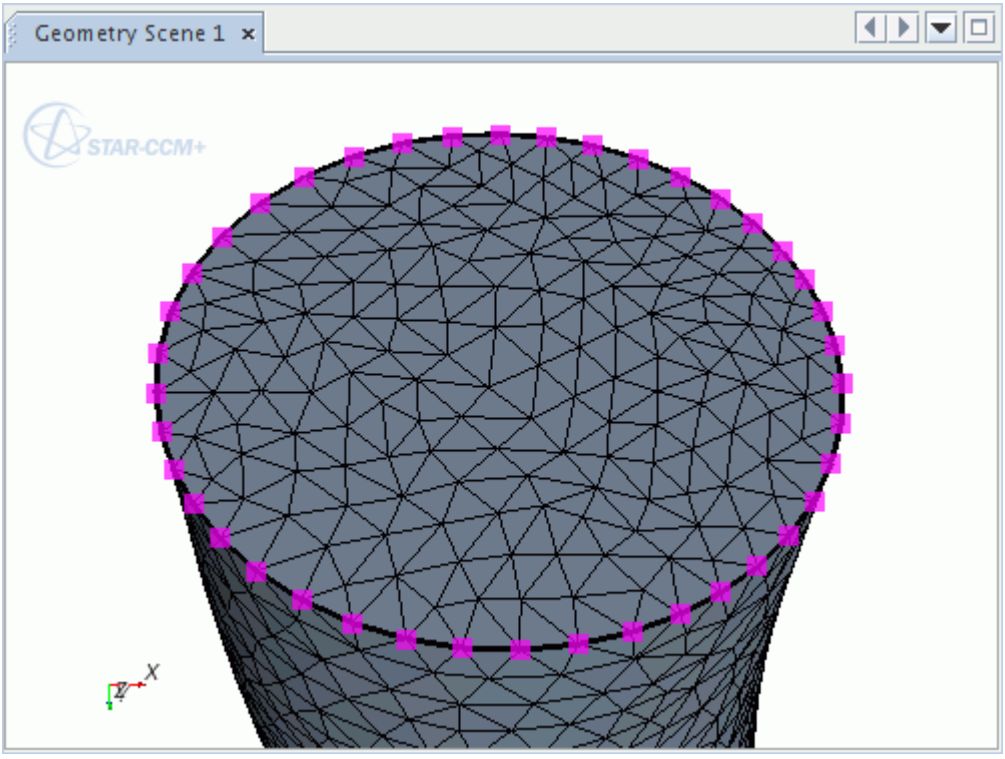


Example of Selecting Entities on a Curve

The following image shows a curve for which the tool selects associated entities.



With the **Vertices** filter activated, the tool selects the following entities.



See Also:

Selection Options
Selecting Entities

Using the Chase Edge Selection Mode

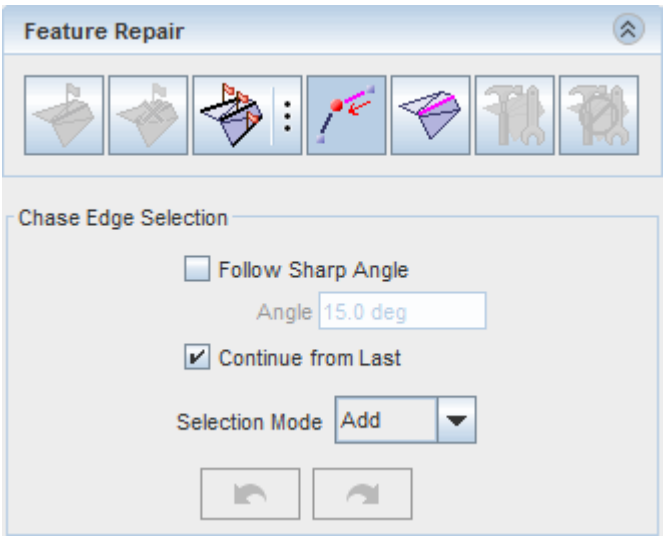
You can select edges manually using the shortest distance between two vertices.

This tool is available when you select the **Edges** filter.

To select edges using the chase edge selection mode:

1. Activate the edge selection filter.
2. Click (**Toggle Chase Edge Selection Mode**).

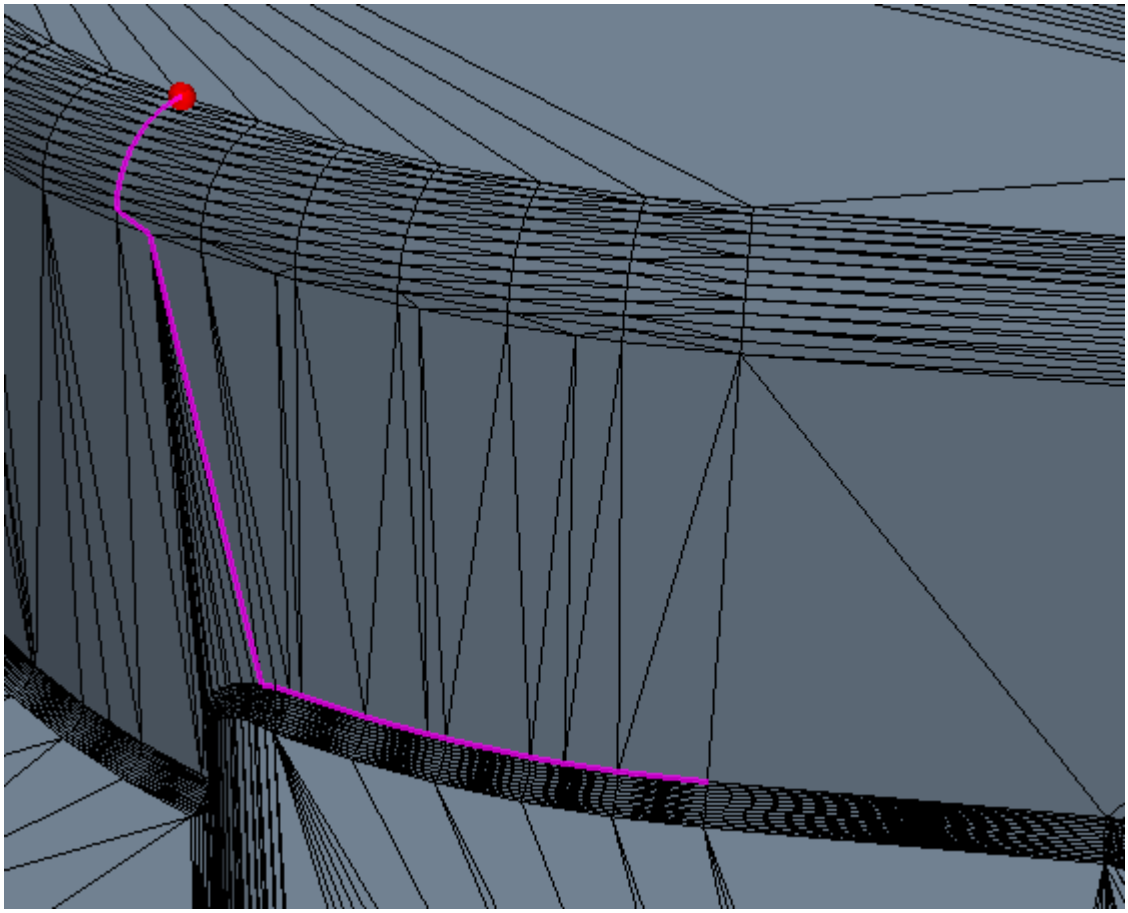
   The *Chase Edge Selection* box appears in the repair panel.



3. Enter the parameters for chase edge mode. See Chase Edge Selection Panel.
4. In the *Graphics* window, click a vertex to select it.

   A red sphere marks this vertex as the starting point.

5. Click another vertex.



The red sphere relocates to indicate the current vertex selection. The edges along the shortest path between the previous and current vertex are selected.

Leaving *Chase Edge Selection* mode maintains selections so you can perform further operations on the edges.

Use the **Undo** and **Redo** buttons to cycle backwards and forwards through the steps that are made in the selection process.

Chase Edge Selection Panel

Enter the parameters to control the edge selection using the Chase Edge Selection mode.

Parameter	Description
Follow Sharp Angle	When activated, selects a path that is defined by edges whose attached faces have a dihedral angle, in degrees, that is greater than the user-defined *Angle*.
Continue From Last	Continues selecting edges from the last selected vertex. Deactivating **Continue from Last** only allows selections between the first and second selected vertices (that is, a non-continuous selection).

Parameter	Description
Selection Mode	Specifies whether the next set of edges you select are included or excluded from the current selection. Set *Selection Mode* to **Add** or **Remove** to control the behavior.

See Also:

Using The Dihedral Angle Selection Mode

Selecting Entities

Using the Select Curves Mode

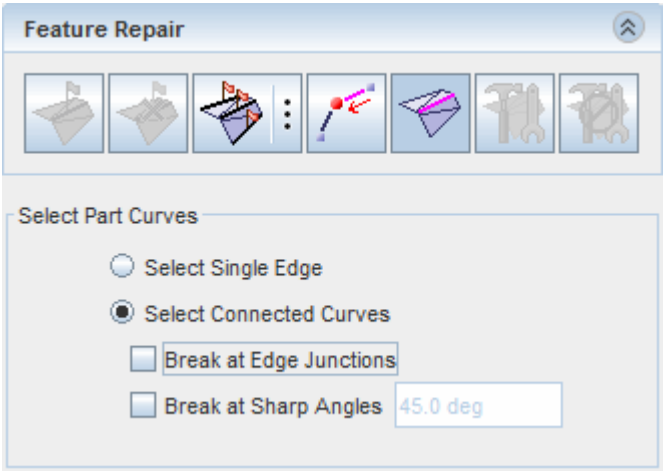
Use the Select Curves Mode to quickly select edges that are marked as feature curves by the **Mark Features by Angle** tool or related tools.

Typical uses for the tool include unflagging feature edges and assigning feature edges to a new or existing part curve.

1. Activate the **Edges** filter.
2. Click (**Toggle Select Curves Mode**).

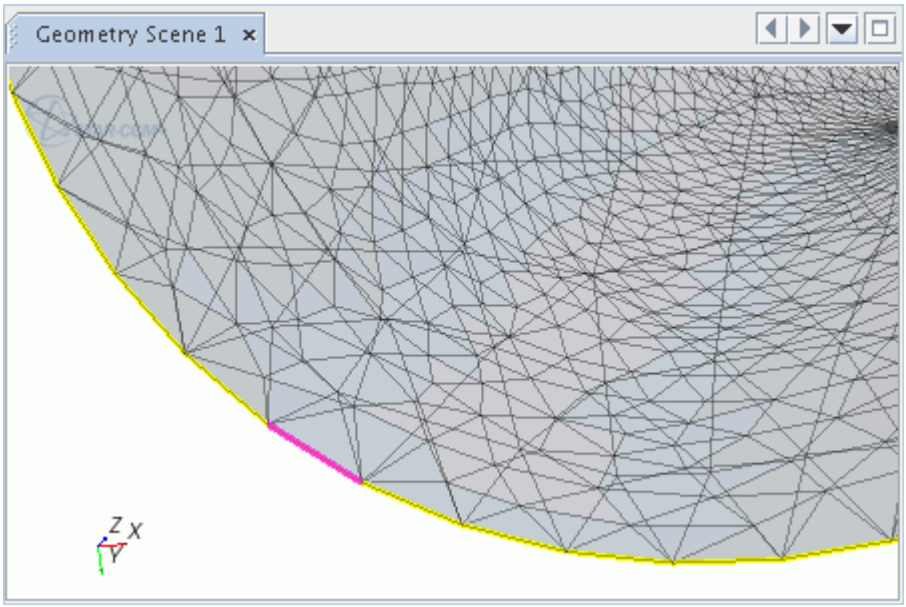
   Note: This action deselects all previous face and vertex selections, and only maintains existing edge selections if they are feature curves.

   The *Select Part Curves* box appears in the repair panel.

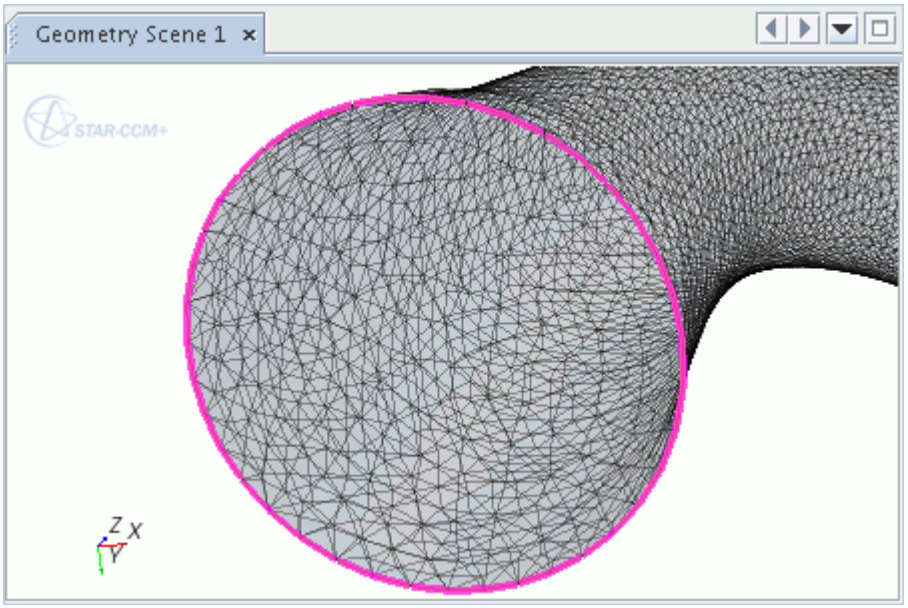


   The part surfaces become translucent and feature curves are highlighted in yellow.

3. Select the desired edges:
   - To select one feature edge at a time, select **Select Single Edge** and make the selections in the *Graphics* window.



- To select all feature edges that are attached to the selected edge, select **Select Connected Curves** and make the selection in the *Graphics* window.



  Activating the **Select Connected Curves** option allows the specification of two further options. These options determine where the feature edge selection stops:

  - **Break at Edge Junctions** - Stops the selection where three or more edges meet.
  - **Break at Sharp Angles** - Stops the selection at a user-defined angle.

- Use the Select Zone tools .

  Clicking multiple feature edges automatically adds to the current selection (that is, you do not need to press `<Ctrl>` for multiple selections).

To deselect a selection, there are two available options:

4. Click a selected feature edge in the *Graphics* window to deselect the edge or connected edges.

   The settings in the *Edit Feature Curves* tool determine the edge or the connected edges.

5. Click (**Select none**) to deselect all selected edges.

See Also:

Selection Options

Selecting Entities

Repairing Surfaces

Diagnose and fix surfaces before running the surface remesher.

It is important to repair the tessellated surface before moving onto the next stage in the meshing process.

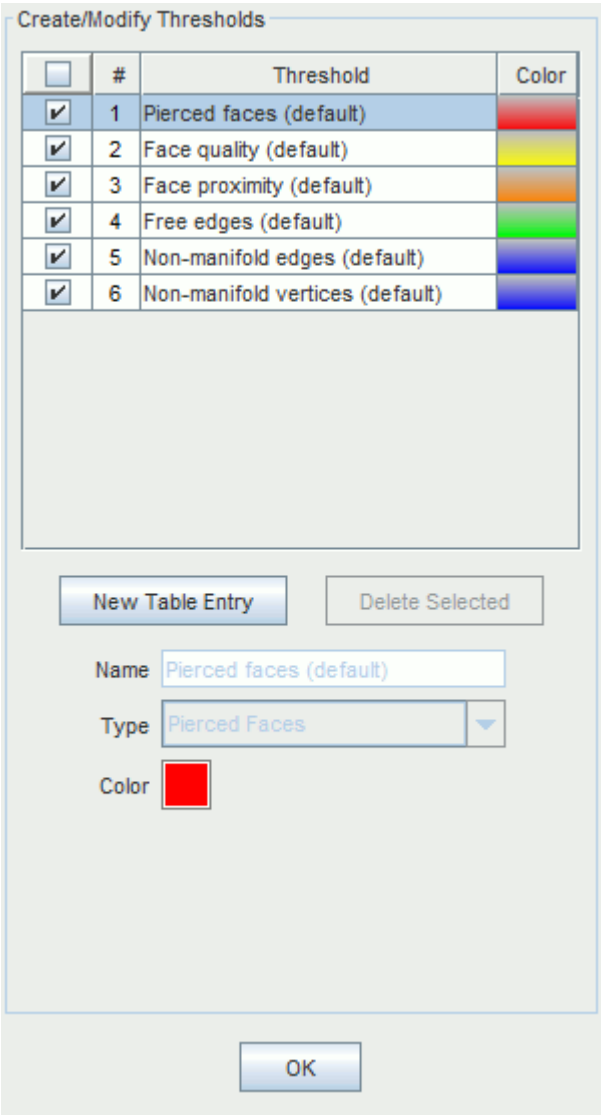
Note: It would not be feasible to document an exhaustive list of surface repair examples that use the tools. Only experience can lead to effective and efficient use of the tools.

The surface repair operations can affect association with underlying CAD parts. In general, any surface repair operation that creates new faces causes a loss of CAD association for the new faces, and possibly the face neighbors. If you activate CAD projection, no projection takes place for these disassociated faces. Therefore, if CAD association is important, keep surface repair to a minimum. You can view the current CAD association for a given surface using the Color Faces display control.

A typical workflow to follow to fix surfaces errors is presented below:

1. In the *Surface Repair* tool, in the *Diagnostics* slide out, select **Surface** and click **Manage**.

   The *Create/Modify Thresholds* panel appears.



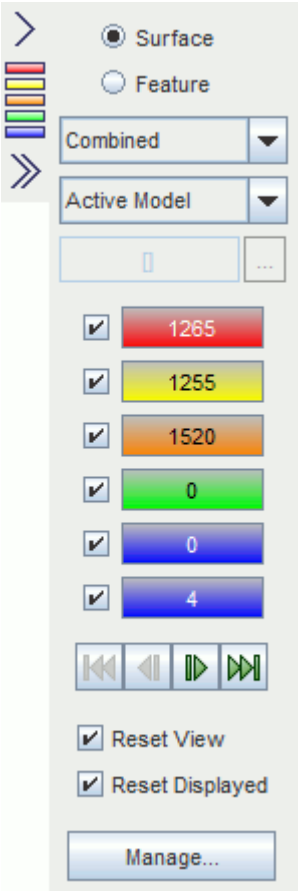
2. In the *Create/Modify Thresholds* panel:

- Deactivate and activate the desired thresholds using the corresponding checkboxes in the first column. To activate or deactivate all thresholds, use the **Enable All Thresholds** option in the table header.
- Click **New Table Entry** to create a custom threshold and define it. See Custom Threshold Options.

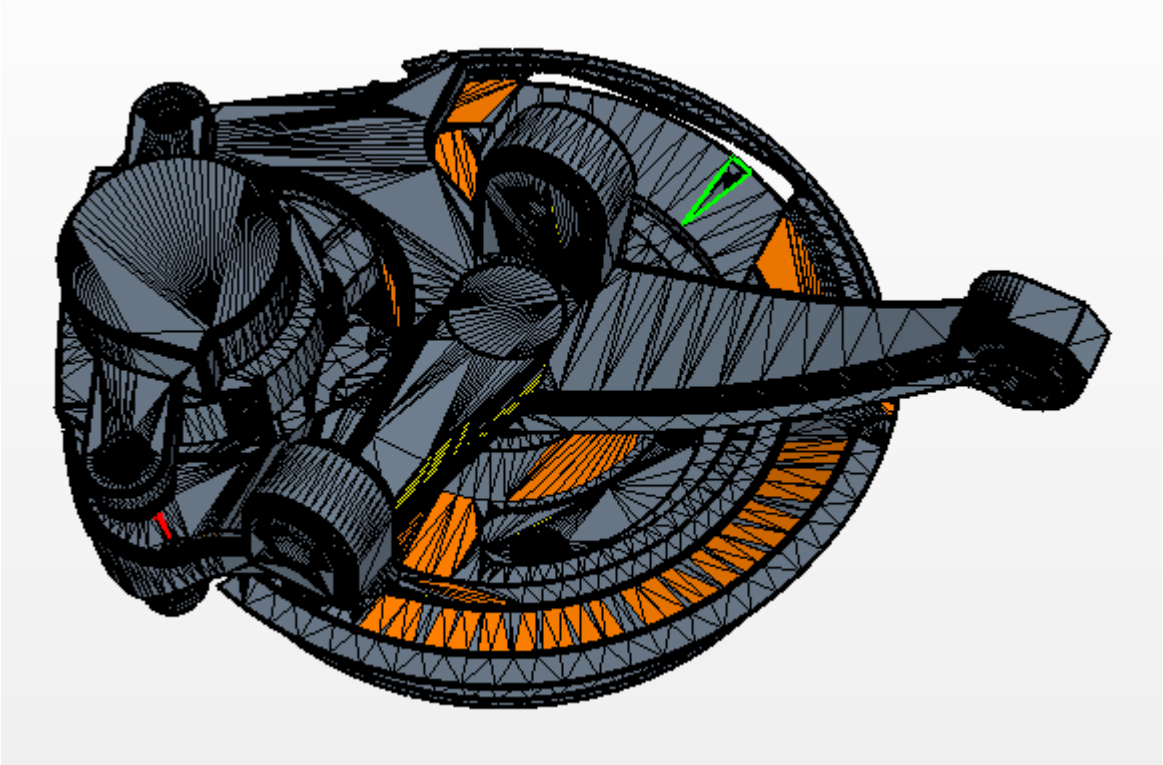
  You cannot change the diagnostic type once it has been created.
- Select a custom threshold and click **Delete Selected** to delete it. You can only delete custom thresholds.

3. Click **OK** to run the diagnostics.

   The tool analyzes the surfaces for errors and displays the results in the *Diagnostics* slide out.



Errors are classified into color-coded categories. The colored boxes in the *Diagnostics* slide out report the number of times that a category of error is found in the active model. The same colors are used in the active scene to highlight the faces and edges on which the errors were found.



See Surface Errors for a list of errors.

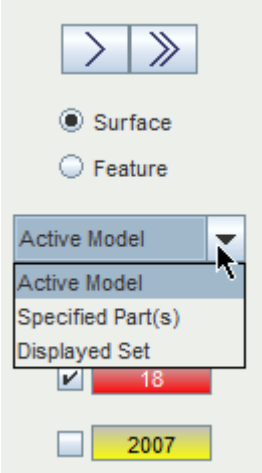
As you fix errors, the number in each category updates to indicate the remaining number of items for which that category of error is found. It is common for the number errors to increase for error types while fixing another error type. For example, deleting faces to remove a pierced surface temporarily increases the number of free edges.

Note: When you click **OK** in the *Create/Modify Thresholds* panel, STAR-CCM+ only reruns diagnostics that need to be rerun due to changes to the input.

Diagnostics are marked as out-of-date if:

- You modified its threshold properties in the panel.
- You execute other diagnostics while it was not activated. This condition is true even if the other diagnostics are up-to-date.
- Once you deactivate a diagnostic and click **OK**, it is marked out-of-date.

4. Control what displays in the Diagnostics slide out. See Diagnostics Slide Out Threshold Input Reference.



Contents:

Custom Threshold Options
Diagnostics Slide Out Threshold Input Reference
Understanding Surface Mesh Diagnostics for Multi-Part Cases
Merging Vertices
Merging Duplicate Faces
Filling Holes

Custom Threshold Options

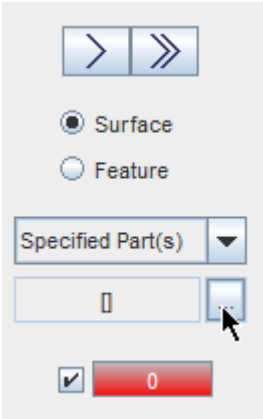
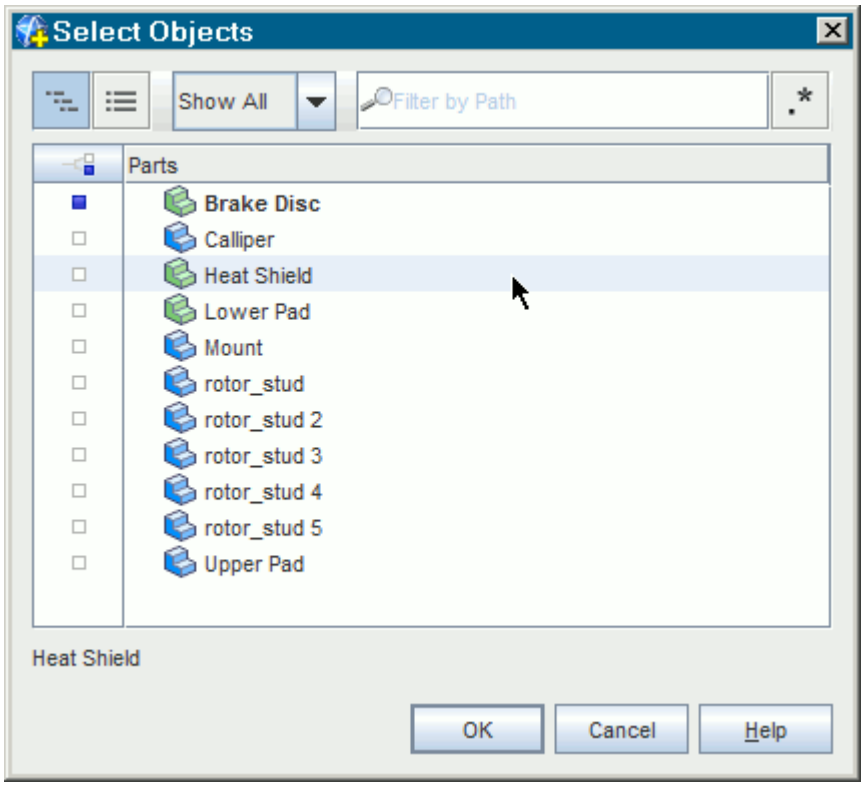
Create custom thresholds using the settings in the Create/Modify Thresholds panel.

Property	Description	
Name	Specifies the name of the custom threshold.	
Type	Specifies the type of threshold.	
	Global Free Edges	Selects edges identified as global free edges. A global free edge is one that belongs to one face only and considers edges at part-to-part perimeters to be connected. The regular free edge diagnostic, in contrast, always considers all free edges from a per-part perspective and marks part-to-part perimeters as being free. See Surface Repair Error Categories: Free Edges. Edges on a baffle surface are not flagged as global free edges.
	Face Quality	Selects faces according to the face quality measure. See Surface Error Repair Categories: Face Quality.
	Face Proximity	Selects faces according to the face proximity measure. See Surface Repair Error Categories: Face Proximity.
	Face Area	Selects faces according to face area.
	Edge Length	Selects edges according to edge length.

Property	Description	
	Edge Dihedral Angle	Selects manifold and non-manifold edges according to the angle between the face normals of the connected faces of the edge. For non-manifold edges, the nearest neighbor faces are used for each angle calculation.
	Duplicate Faces	Selects faces identified as duplicates. Duplicate faces exist when the three vertices of a given triangle are shared by another triangle, potentially in a different part. A duplicate face is a subset of the face proximity diagnostic, meaning that the same faces are flagged as close proximity faces if the diagnostic is enabled. Having duplicate faces on the geometry can lead to meshing issues downstream. See Surface Repair Error Categories: Face Proximity. A pair of duplicate faces belong to the per-part result subset if both faces belong to the same part. A pair of duplicate faces belong to the part to part result subset if each face belongs to a different part. If a duplicate face pair exists on a contact, then the duplicate face pair automatically belong to the part to part subset. If the non-contact duplicate face is of the same face type as the contact face types, then the faces will also be flagged as per-part faces. For more information about per-part and part to part interaction modes, see Surface Repair User Interface. To repair duplicate faces, use the Merge Duplicate Faces tool. See Merging Duplicate Faces.
Color	Specifies the color to apply to issues detected by the custom threshold.	
Operator	**Above**	The tool searches for all values higher than the one you specify in the **Value** entry field.
	Below	The tool searches for all values lower than the one you specify in the **Value** entry field.
	Between	The tool searches for all values higher than the one that you specify in the **Min** entry field but lower than the one you specify in the **Max** entry field.
	Not Between	The tool searches for all values lower than the one that you specify in the **Min** entry field but higher than the one you specify in the **Max** entry field.
Value/Min	Specifies the value to use with the *Operator*. Depending on the operator you specify, this field can appear as **Value** or **Min**.	
Max	Specifies the maximum value for the operator.	

Diagnostics Slide Out Threshold Input Reference

After running diagnostics, you can control the input for the thresholds in the diagnostics slide out.

Input	Description
Active Model	Reports the diagnostics for the parts that are brought into surface repair initially. This setting is the default.
Specified Part/Regions	Reports the diagnostics for the selected parts or regions. To select a part or region: 1. Click ... below the threshold input display option.  The Diagnostics are reset to 0 until you select inputs from the object selection dialog. 2. In the *Select Object* dialog. 

Input	Description
	3. Click **OK**. The diagnostics are reported and displayed for the selected parts or regions only.
Displayed Set	Reports the diagnostics only for the object that is shown in the scene. When all faces are shown, the results for **Displayed Set** and **Active Model** are identical. More computations are performed when you change faces in the display, for example when you browse errors with the (**Browse**) buttons. If you have many faces, reduce your display set.

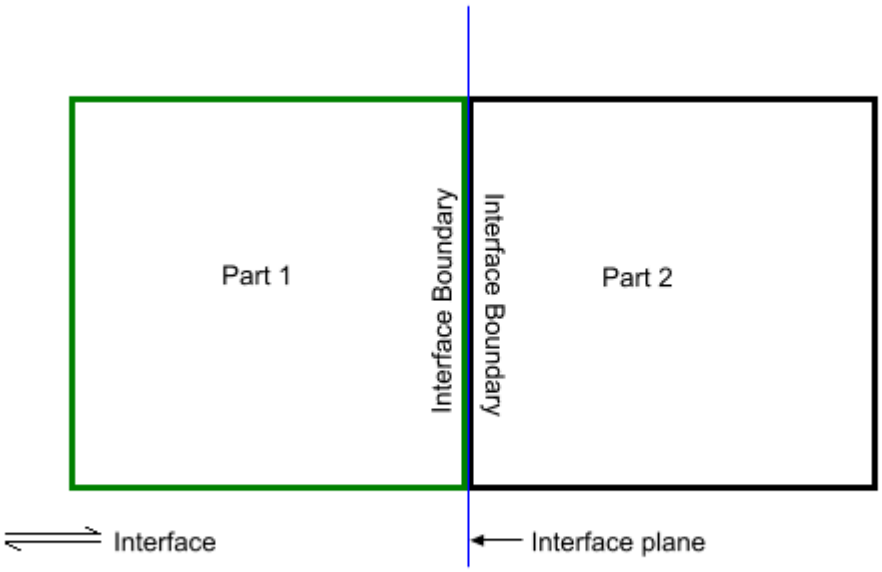
Understanding Surface Mesh Diagnostics for Multi-Part Cases

Whether the surface belongs to single or multiple parts affects the diagnosis of non-manifold edges, proximity and self-intersection.

For example, if one of these errors is on an interface between parts, diagnosing a single part does not report the error. However, the diagnostics tool calculates free edges on a per-part basis, which is independent of the number of parts that are selected for diagnostics. See Changing the Parts in the Active Model.

The main consideration in multiple parts cases is whether an interface exists between two neighboring parts. This section aims to provide an overview of what to expect when diagnosing multiple part cases, and how to recognize and fix surface errors. Some surface errors, such as free edges, do not affect the surface remesher. However, having any form of non-closed or non-manifold surface causes the volume meshers to produce an error message and does not generate a mesh.

This section uses a simple example, two cube surfaces, to highlight the different scenarios possible when dealing with multiple parts or parts. The diagram below illustrates the terminology:



The examples below focuses on whether each cube is closed (that is, has a surface at the interface plane) and whether an interface or contact exists between the two cubes. Additionally, this section covers an example where two parts or parts share edges, due to merged vertices.

It is important to note, the surface diagnostics tool identifies errors regarding the active model and not on a per-part basis, except for free edges as noted earlier. Therefore, select all required parts in the *Surface Repair Options* dialog before staring surface repair, otherwise the diagnostics tool does not identify certain errors.

Two Connected Boundaries with No Interface

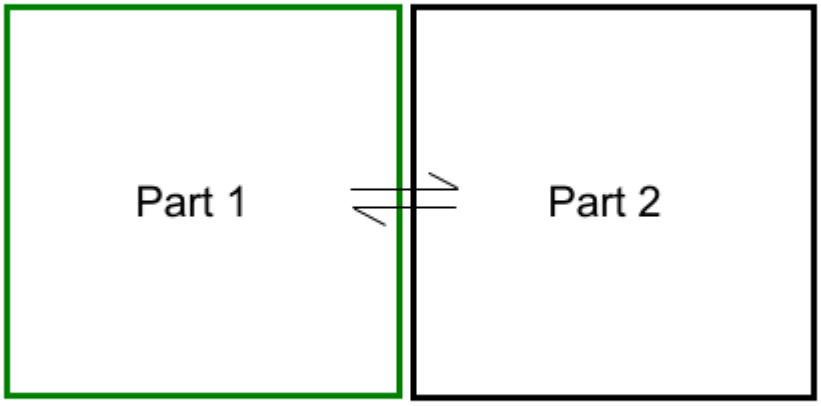
This situation is the simplest, where the two cubes are closed and no interface exists between the boundaries at the interface plane.

The surface diagnostics do not report any free edges or non-manifold edges in this instance. However, the diagnostics tool identifies interface boundaries as being close and piercing one another. Attempting to volume mesh in this situation results in an error message reporting self-intersecting surfaces (providing you are not using per-part meshing).

To fix these surface errors, create an interface between the two boundaries at the interface plane. This fix allows the volume mesher to proceed and creates a conformal mesh between the two parts, if supported by the meshing model. See Creating a Boundary Interface.

Two Connected Boundaries with an Interface

This situation is the same as the previous situation, but with an in-place interface created between the boundaries at the interface plane.



Surface diagnostics report no issues in this instance, and the meshers are able to create a volume mesh. This situation is the ideal situation for using the STAR-CCM+ meshing pipeline.

One Surface with No Interface

This section considers two situations. The first situation uses cubes without shared edges, that is, two disconnected cubes. The second situation uses cubes with shared edges, that is, two connected cubes.

In the first situation, the diagnostics tool marks the edges on the part containing the open surface as free edges. Attempting to volume mesh in this situation results in an error message reporting one part as open. To fix this error, close the open part by filling the hole, using the surface repair tools. Then create an interface between the two boundaries, which provides the ideal situation that is described above. The meshers are then able to create a volume mesh.

In the second situation, the diagnostics tool marks the edges on one part as free edges, since the surface only belongs to one part. Attempting to volume mesh in this situation produces the same error as above, reporting one part as open. To fix this error, close the open part by filling the hole, using the surface repair tools. See Repairing Surfaces.

However, this fix appears to swap one problem, free edges, for another, non-manifold edges. Therefore, create an interface between the two boundaries, which resolves the non-manifold edges, and allow volume meshing.

No Interface Boundaries, with No Interface/ Between Parts

In this situation, both cubes do not have an interface, and the surface diagnostics marks the open ends of both parts as free edges.

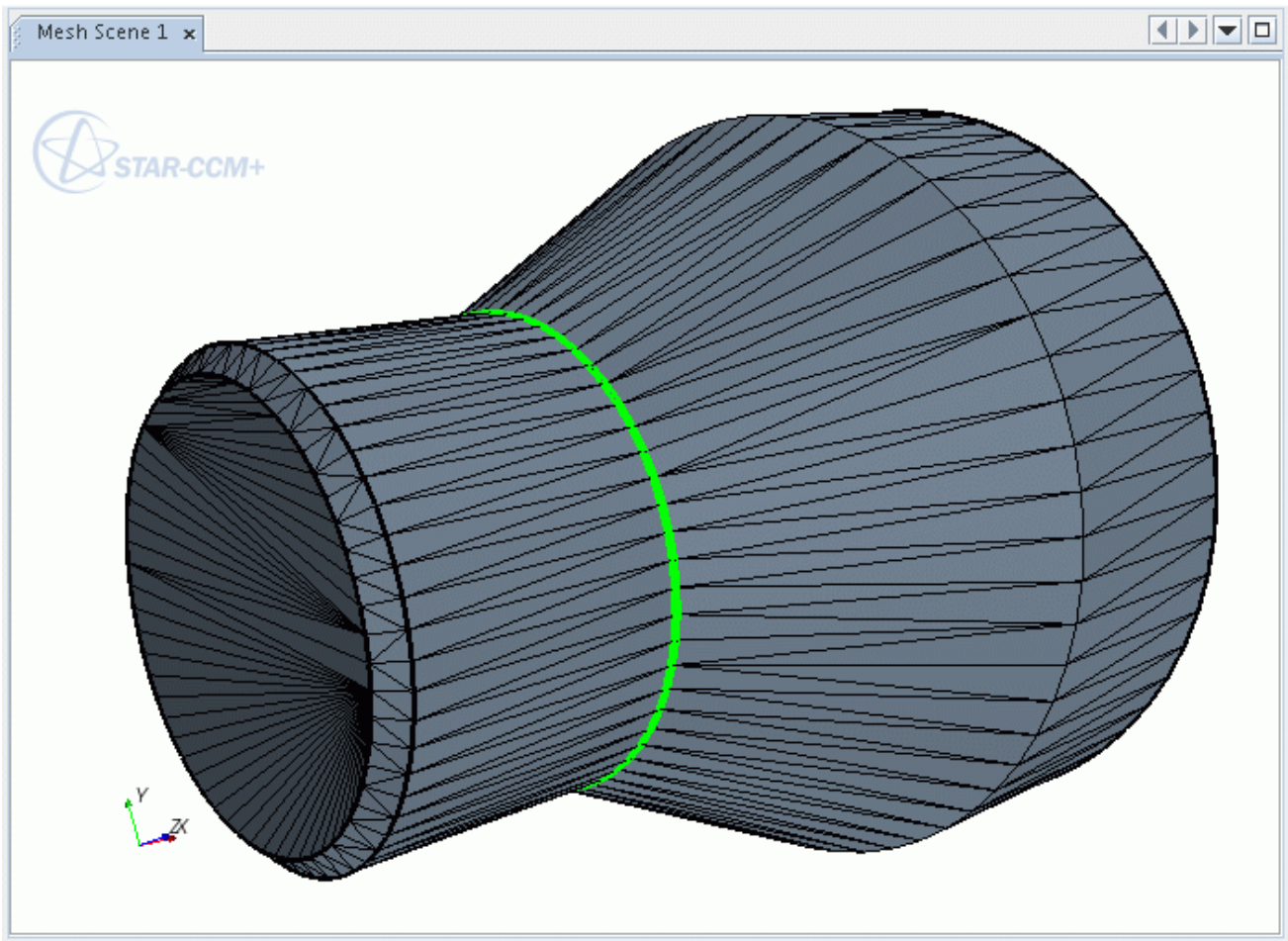
To fix this error, close each part separately by starting surface repair with only one part that is selected as the active model. Then create an interface between the two parts, which allows the volume mesher to run.

Merging Vertices

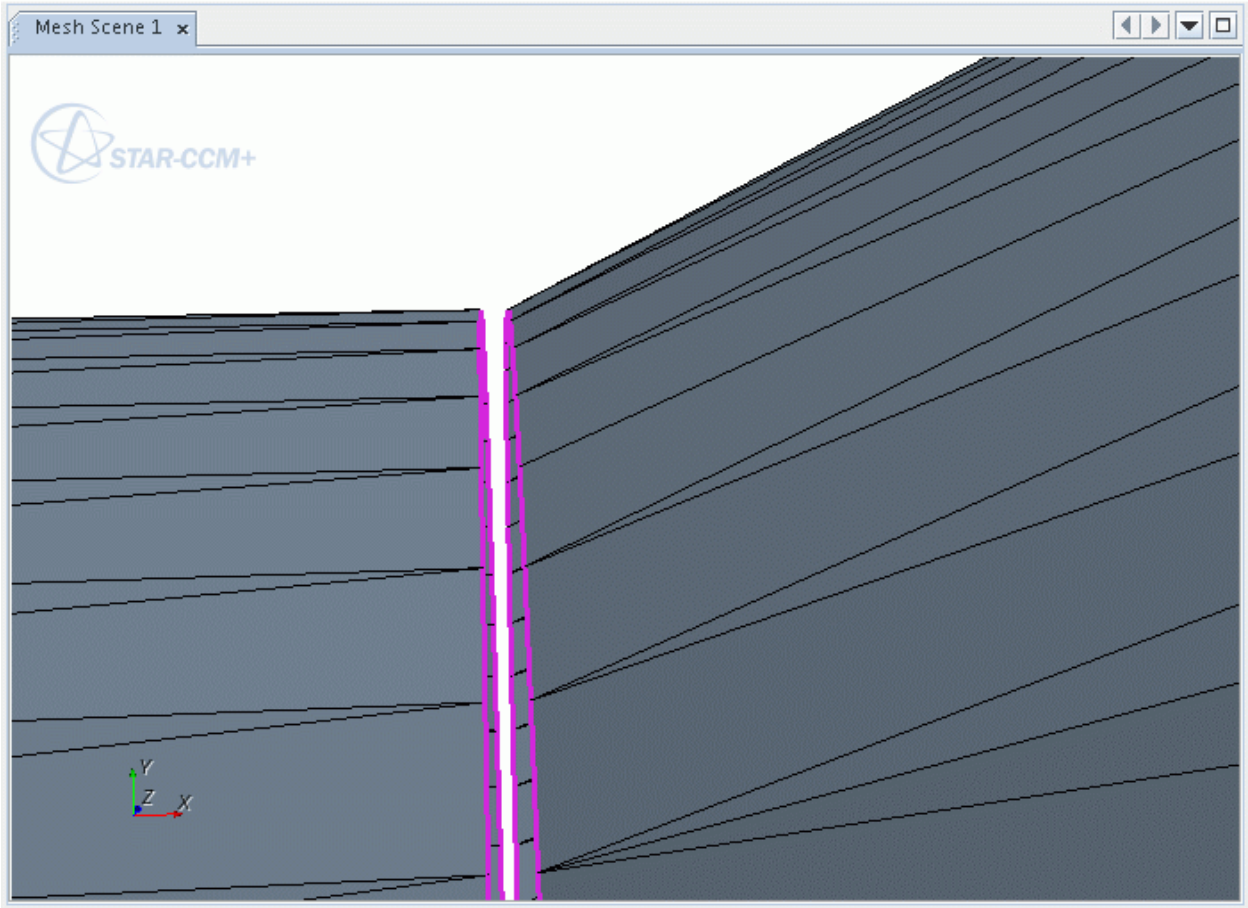
When merging vertices, review and change the tolerance value, if necessary.

Set the value to a reasonable value based on the maximum distance that separates the vertices you want to merge, plus a small tolerance (say 1/10th to 1/5th of the measured value). If the tolerance is too high, inadvertent merging of neighboring vertices can occur, especially when choosing to merge based on the active model. The safest way to merge vertices is to select the required vertices, and use the **Selected set only** option in the merge options.

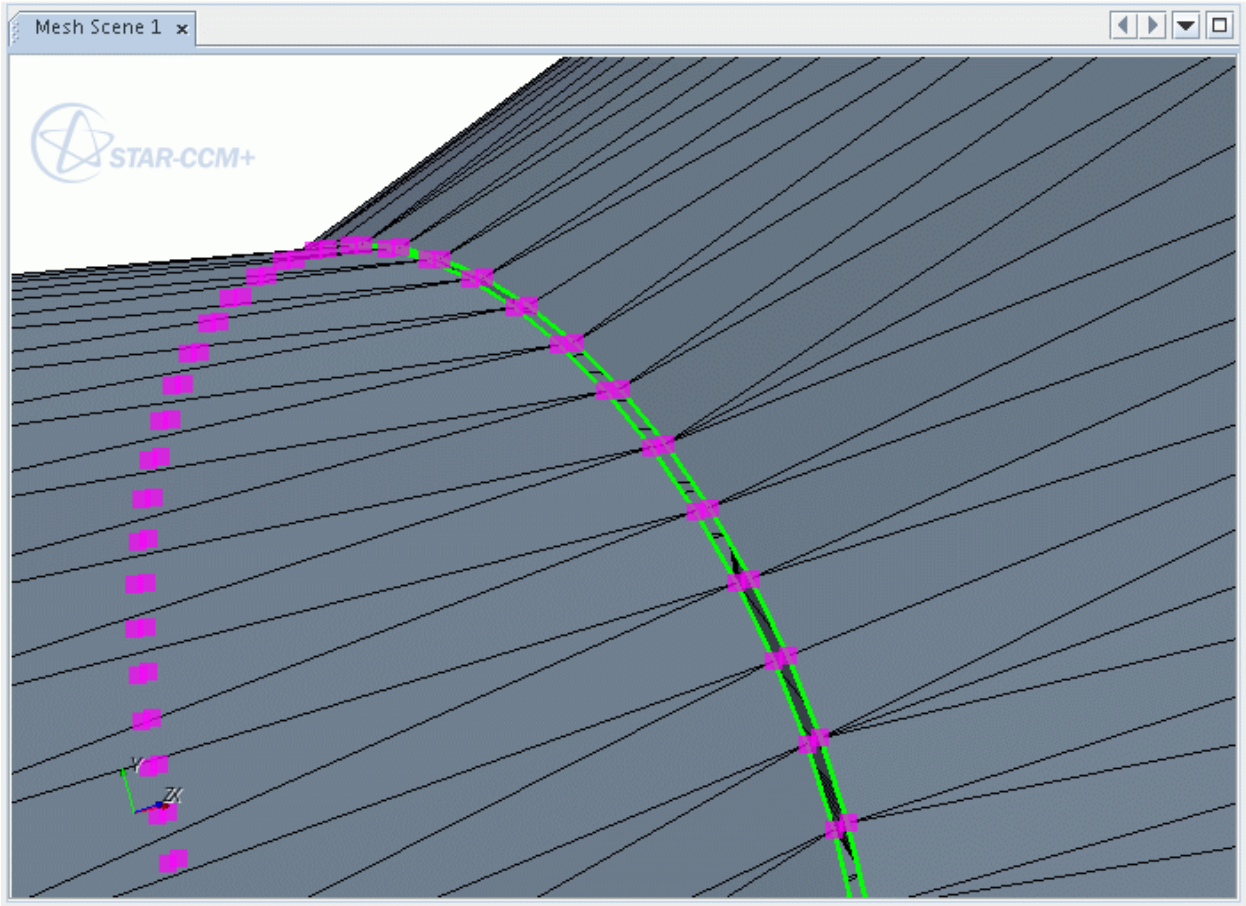
The following example is a part containing free edges, representing an unmerged portion of the surface:



1. Select the free edges by clicking the green box in the *Diagnostics* slide out.



2. Click (**Select Attached Vertices**) to select the vertices that are attached to the selected edges.



Alternatively, use the (**Select Zone**) tool to select the appropriate vertices.

3. Set the merge options:

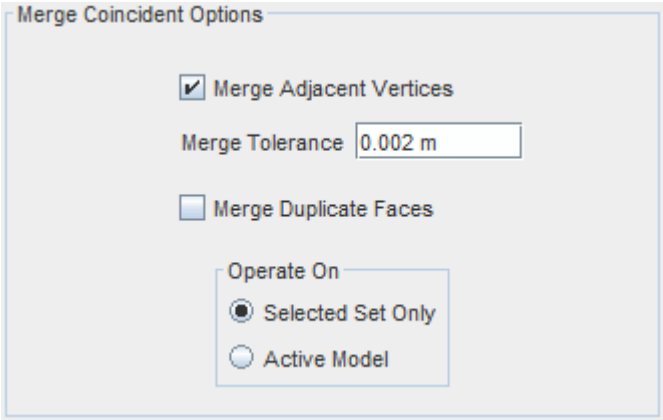
   a) Click the three dots next to the (**Merge adjacent vertices/duplicatet faces**) tool to display the options.

   b) Enter the tolerance in the *Merge Tolerance* field.

      You can estimate the required tolerance value by measuring the distance between two vertices on either side of the gap.

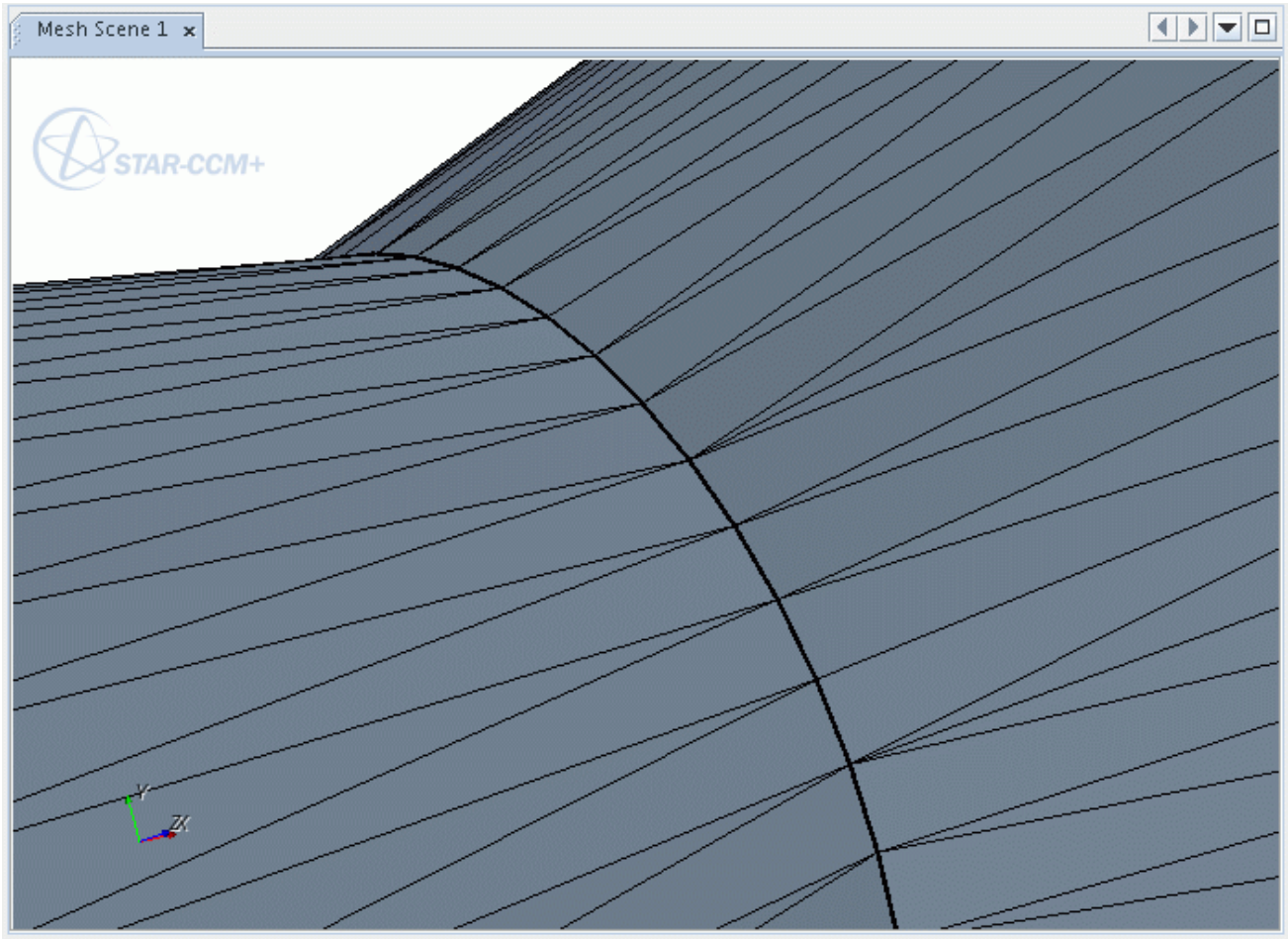
   c) Deactivate the **Merge Duplicate Faces** option.

   d) Set *Operate On* to **Selected Set Only**.

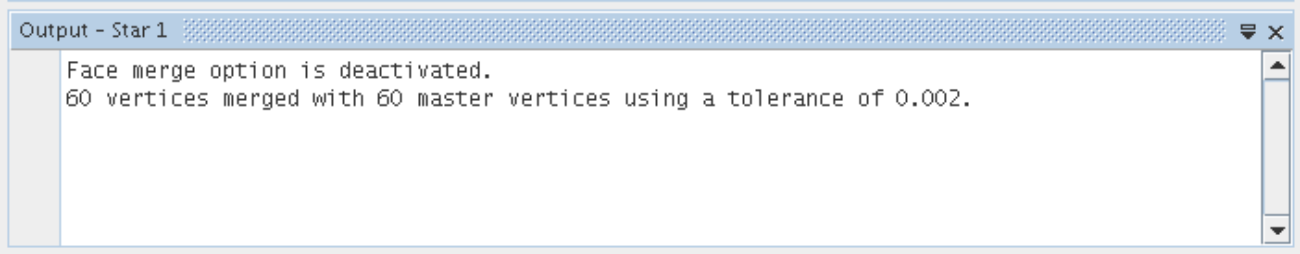


4. Click (**Merge adjacent vertices/duplicatet faces**) to merge the selected vertices.

The display updates with the merged result.



The tool has uniformly merged the vertices on one surface with the vertices on the other surface, resulting in a clean merge line. The *Output* window shows a summary of the number of vertices merged.



If you set the tolerance too low, either no or partial merging would take place. However, any unmerged vertices would remain in the selected set allowing for further merging with a revised tolerance value.

The vertex merging scheme uses a lowest vertex number approach, which means it always merges the highest vertex number (slave vertex) to the lower number (master vertex) within the specified tolerance distance.

Merging Duplicate Faces

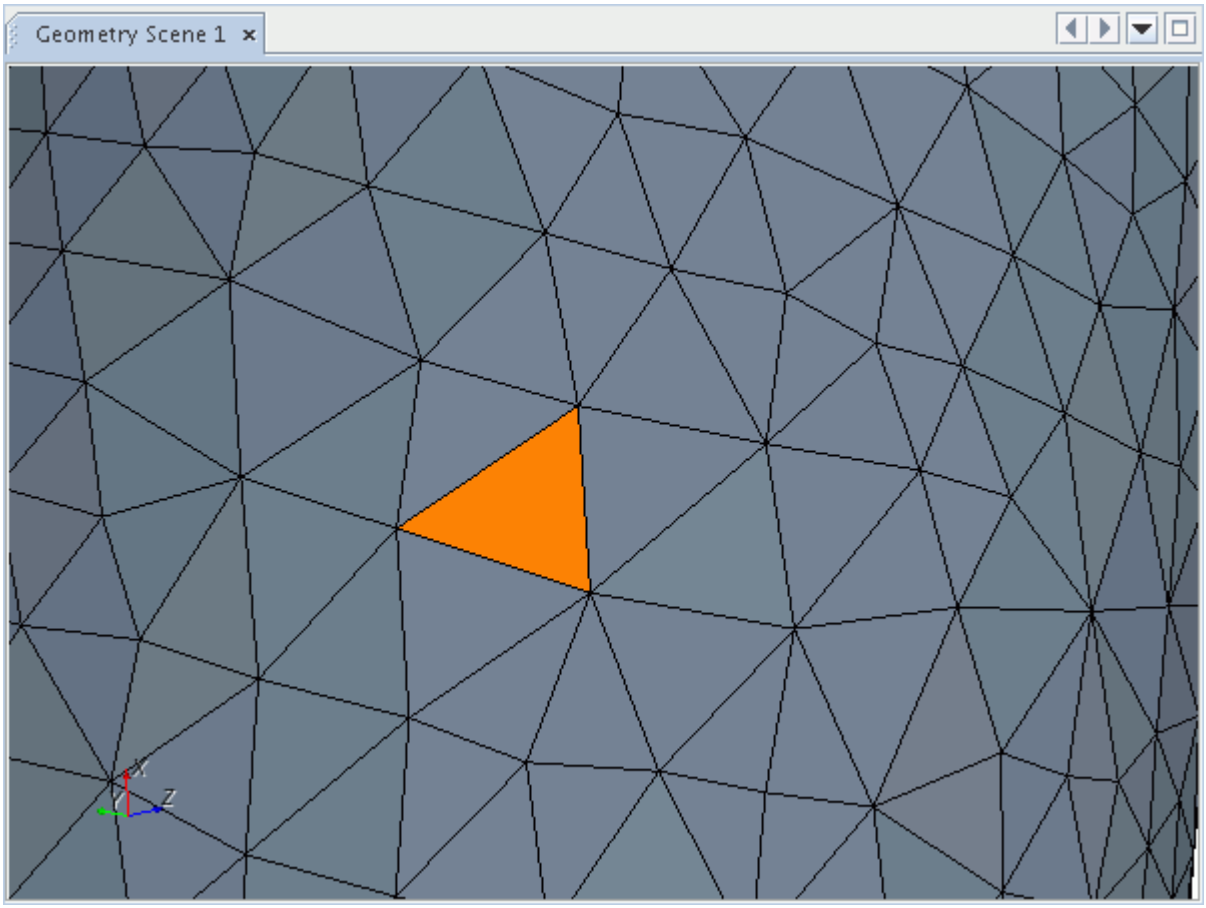
A duplicate face is a face that shares three vertices with another face.

The **Surface Repair** tool flags these as close proximity faces and colors them orange. However, not all close proximity faces are necessarily duplicate faces.

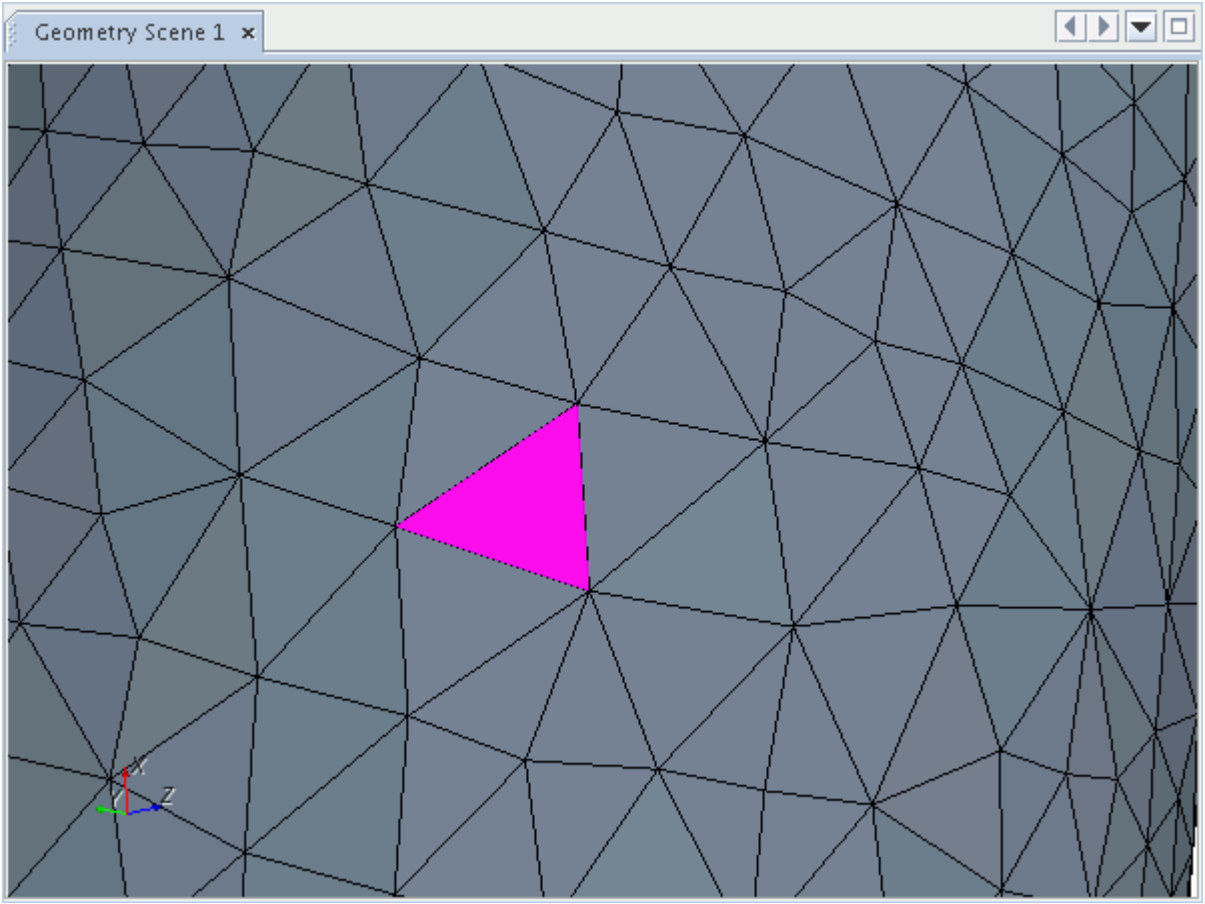
If multiple duplicate faces exist for one face, the tool removes all duplicates using the merge process.

The vertex merge tool can occasionally create duplicate faces, especially when merging vertices from close proximity faces. Using the tool on the active model merges these additional duplicate faces.

The following example shows the **Merge Duplicate Faces** tool in use.



1. Select the close proximity faces by clicking the orange box in the *Diagnostics* slide out.



2. Set the merge options:

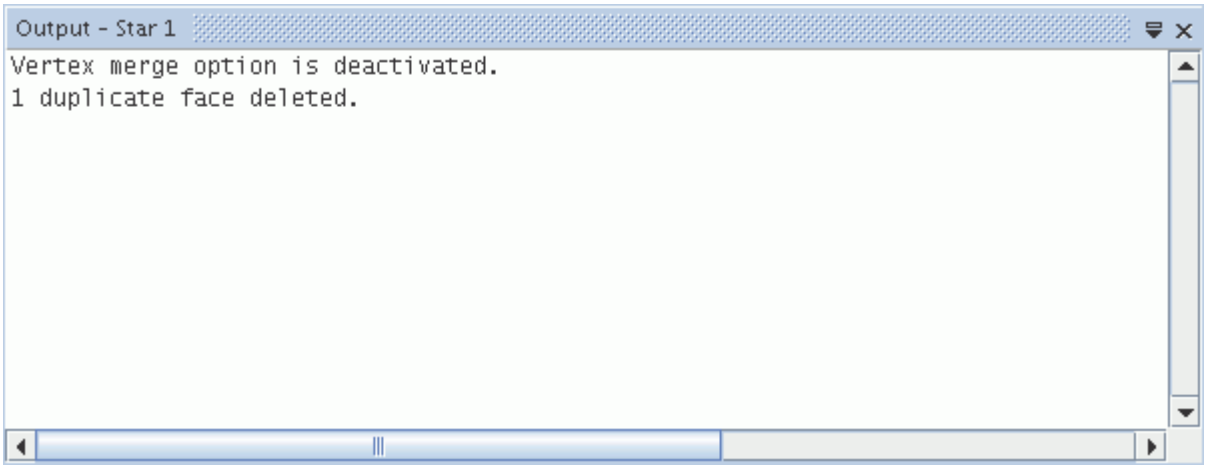
   a) Click the three dots next to the (**Merge adjacent vertices/duplicatet faces**) tool to display the options.

   b) Deactivate the **Merge Adjacent Vertices** option.

   c) Set *Operate On* to **Selected Set Only**.

3. Click (**Merge adjacent vertices/duplicatet faces**) to merge the duplicate faces.

If any of the close proximity faces are duplicates, the merge removes the duplicate face. The *Output* window reports the number of duplicate faces deleted.



Filling Holes

This tool fills a hole with new faces using the selected edge loops.

For closed loops, select at least three edges for this tool to become available. For open loops, select at least two edges for this tool to become available.

The fill holes tool has three modes:

- Mixed - Fill a combination of holes that are defined by chains of selected edges.
- Closed loop(s) only — Fill a hole that is defined by a continuous chain of selected edges.
- Open loop(s) only — Fill a hole that is defined by a discontinuous chain of selected edges.

Contents:

Filling Mixed Loop Holes
Filling Closed Loop Holes
Filling Open Loop Holes
Maintaining Local Curvature
Open Loop Closure Options
Loop Interaction Options

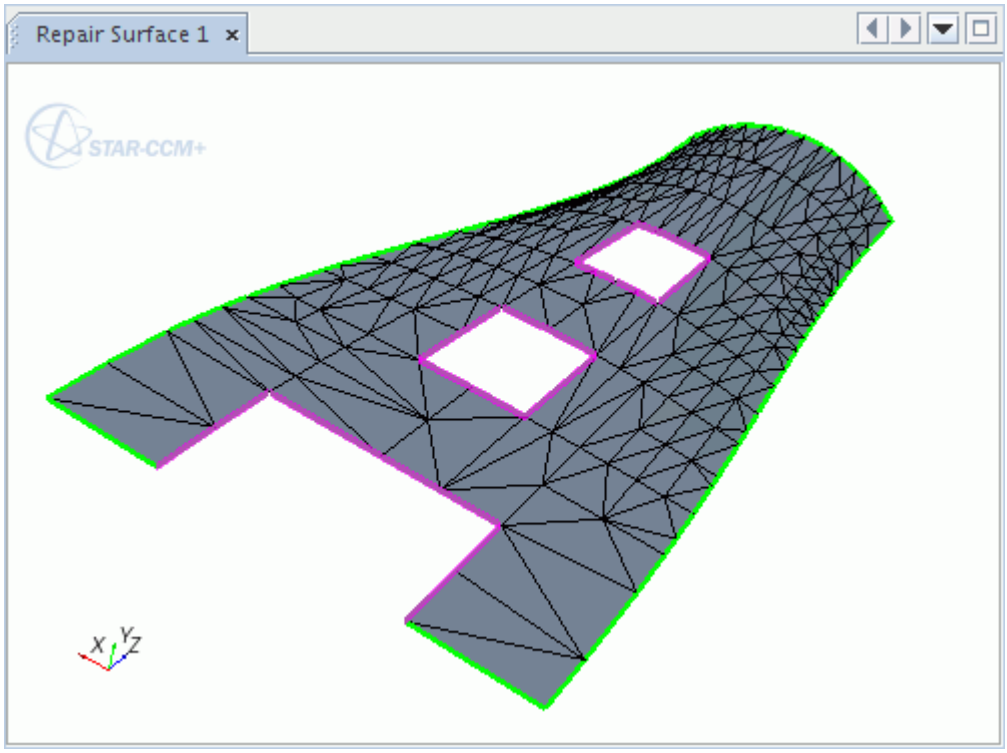
Filling Mixed Loop Holes

The mixed hole filler fills a combination of holes that are defined by continuous or discontinuous chains of selected edges.

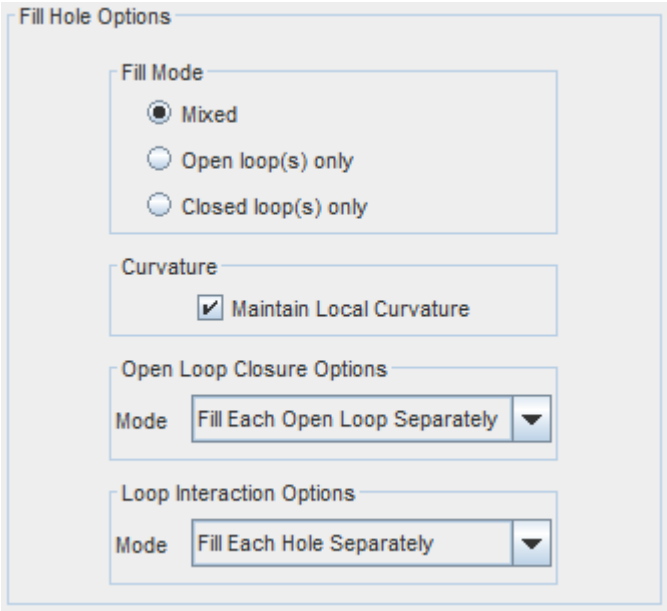
This option is useful if both open and closed loop holes exist within the same surface.

To fill mixed loop holes:

1. In the *Graphics* window, select the edges that define the open and closed loops.

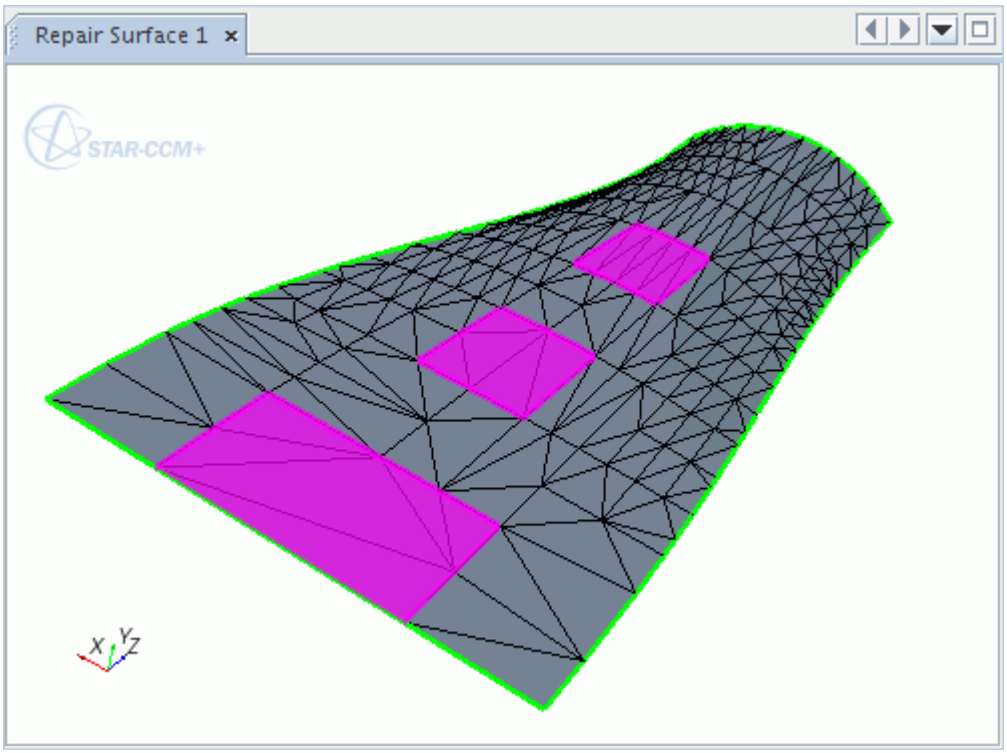


2. Click the three dots next to the (**Fill holes**) tools to display the fill hole options.
3. In the *Fill Holes Options* group box:
   a) Set *Fill Mode* to **Mixed**.
   b) Decide whether you want the fill area to maintain local curvature. See Maintaining Local Curvature.
   c) Set the *Open Loop Closure Options* mode. See Open Loop Closure Options.
   d) Set the *Loop Interaction Options* mode. See Loop Interaction Options.



4. To fill the holes, click (**Fill holes**).

The open and closed holes are filled.



See Also:

Loop Interaction Options

Filling Open Loop Holes

Filling Closed Loop Holes

The closed-loop hole filler fills holes that are defined by continuous loops of edges.

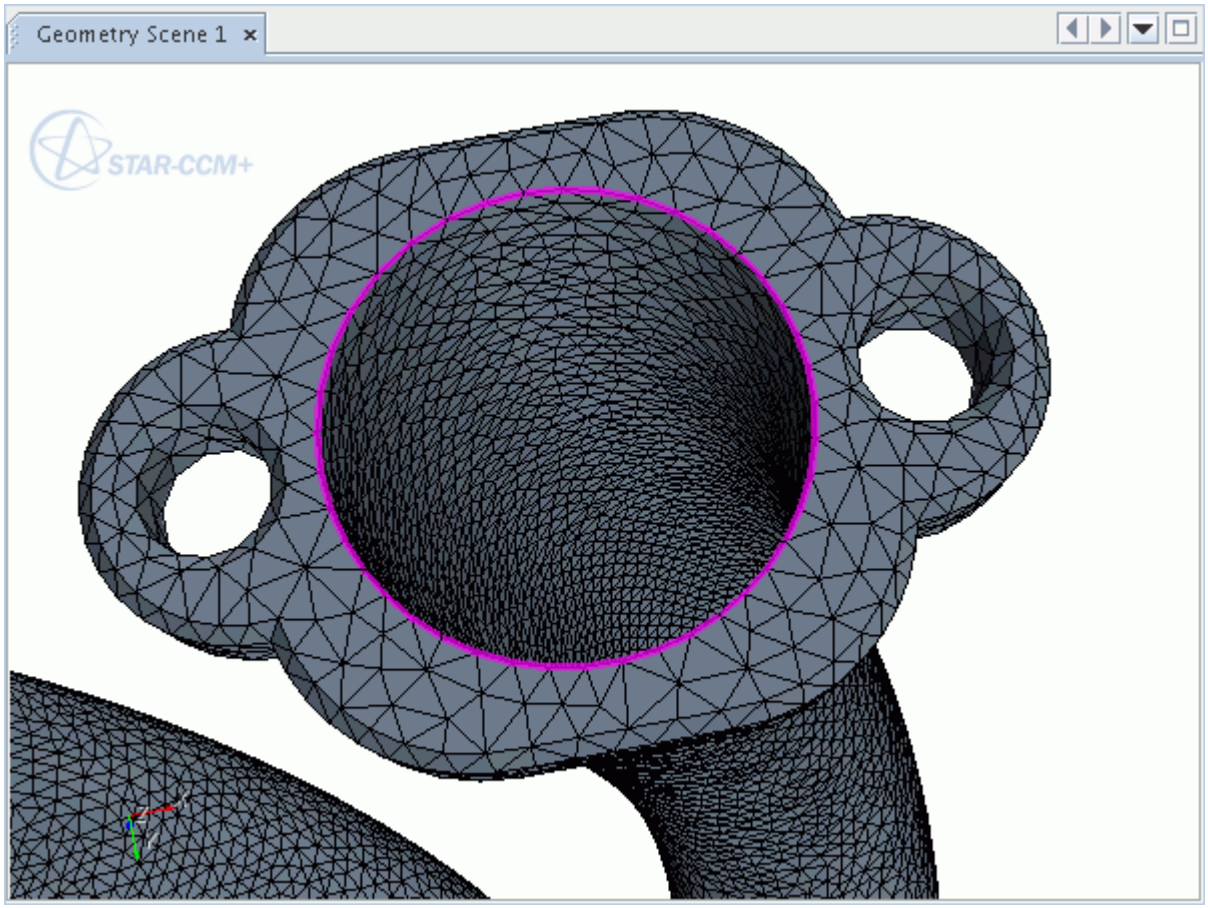
You typically use this option before extracting the internal fluid volume for a CFD analysis.

Note: When filling closed loops, the tool checks that the selected edges form a closed loop. The tool does nothing if it does not find the closed loop.

To fill a closed-loop hole:

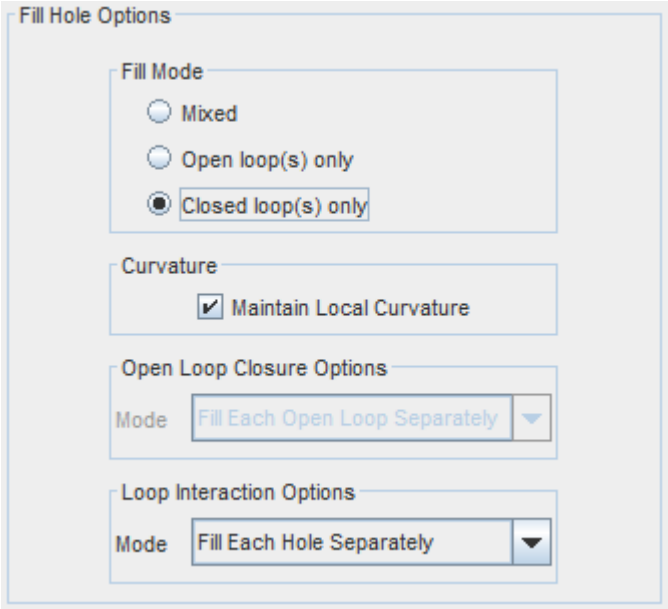
1. In the Graphics window, select the edges that define the closed loop. Double-click a single free edge to automatically selects all the free edges connected to it.

   The following example uses the fill holes tool to cap the hole in a solid component.



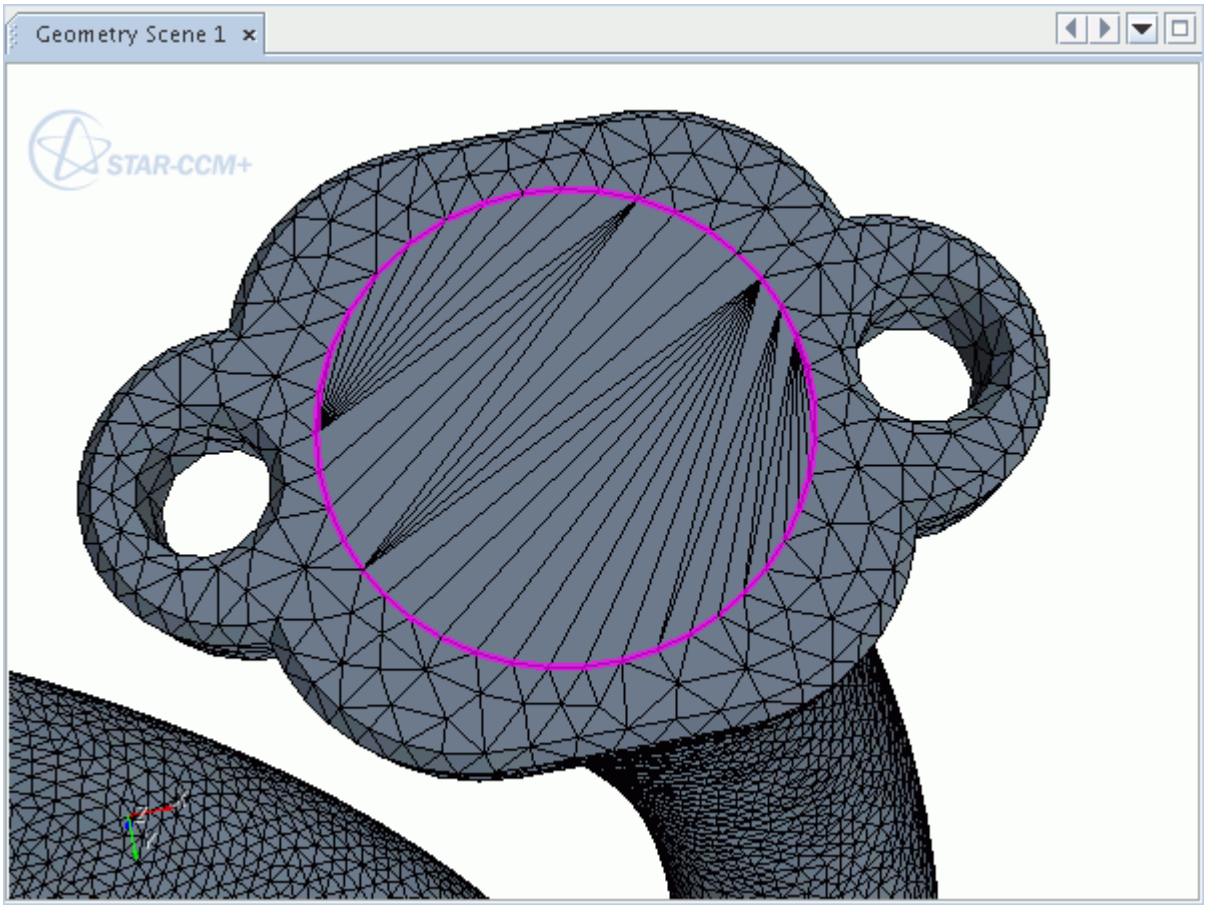
2. Click the three dots next to the (**Fill holes**) tools to display the fill hole options.
3. In the *Fill Holes Options* group box:
   a) Set *Fill Mode* to **Closed Loop(s) only**.
   b) Decide whether you want the fill area to maintain local curvature. See Maintaining Local Curvature.

c) Set the *Loop Interaction Options* mode. See Loop Interaction Options.



4. To fill the hole, click (**Fill holes**).

The hole is capped.



You can improve the poor quality faces by locally remeshing the surface. See **Remesh Selected Faces** in Surface Repair Tools.

See Also:

Loop Interaction Options

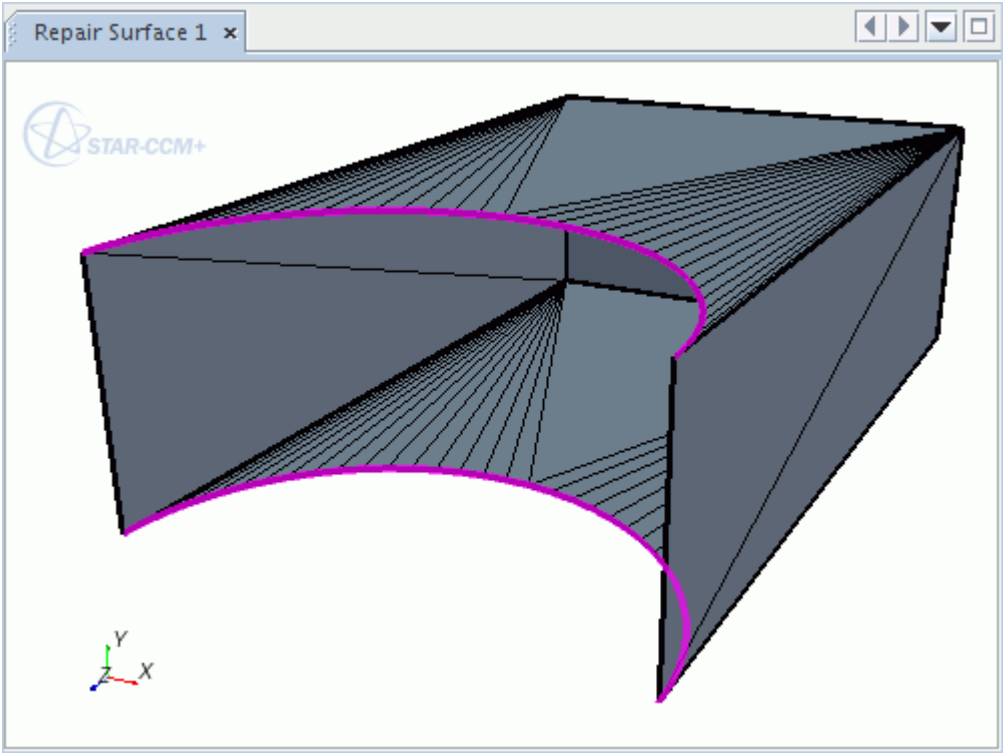
Filling Open Loop Holes

The open-loop hole filler fills holes that are defined by discontinuous loops of edges.

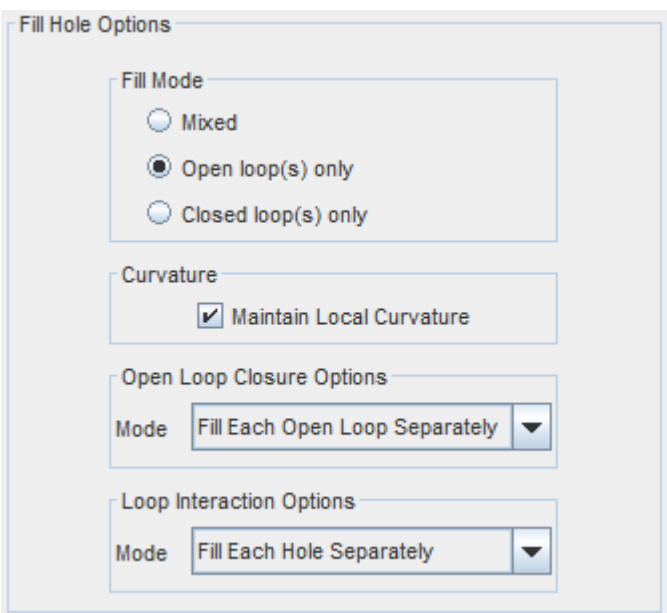
Note: When filling open loops, the tool checks for closed loops in the selection and cancels the command when it finds one.

To fill an open-loop hole:

1. In the *Graphics* window, select the edges that define the open and closed loops.



2. Click the three dots next to the (**Fill holes**) tools to display the fill hole options.
3. In the *Fill Holes Options* group box:
   a) Set *Fill Mode* to **Open loop(s) only**.
   b) Decide whether you want the fill area to maintain local curvature. See Maintaining Local Curvature.
   c) Set the *Open Loop Closure Options* mode. See Open Loop Closure Options.
   d) Set the *Loop Interaction Options* mode. See Loop Interaction Options.



4. To fill the hole(s), click (**Fill holes**).

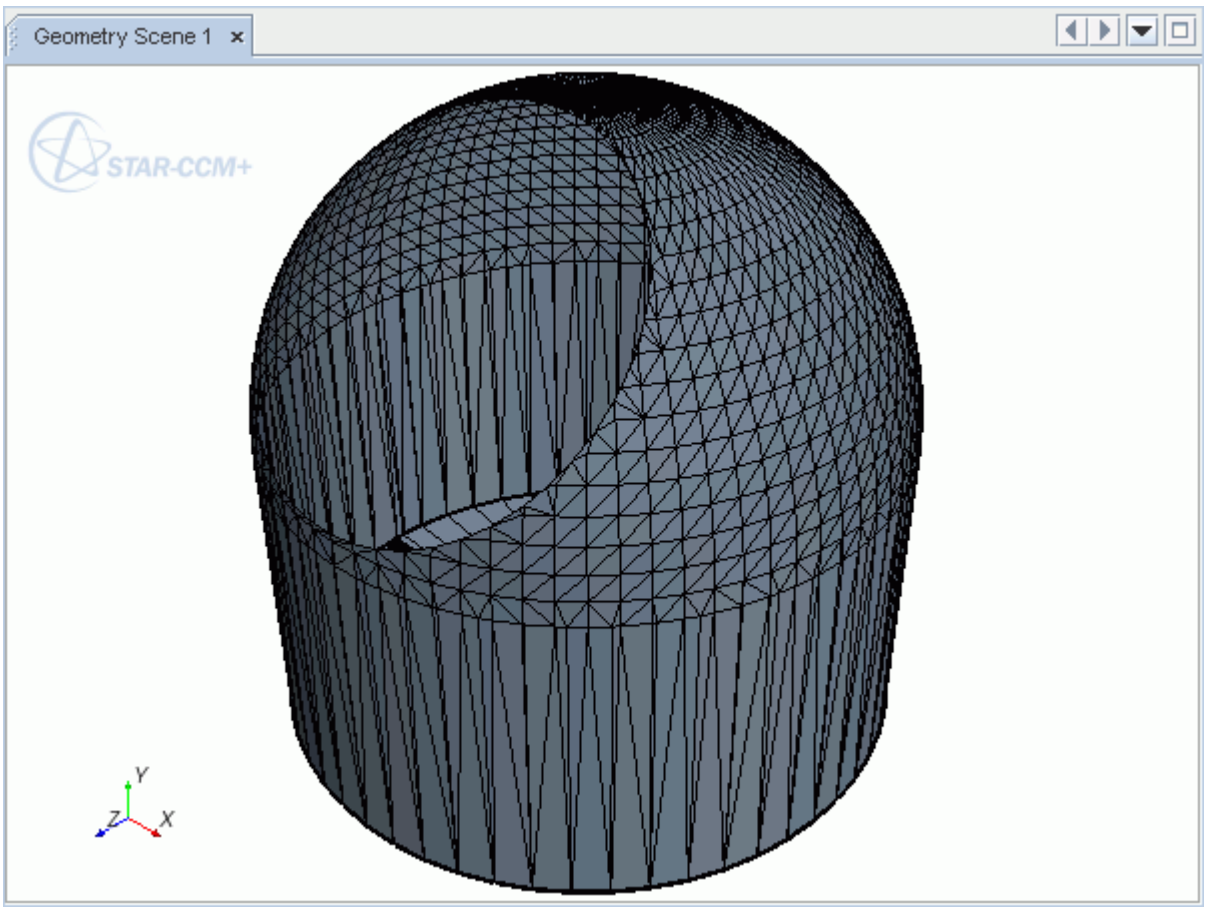
See Also:

Loop Interaction Options

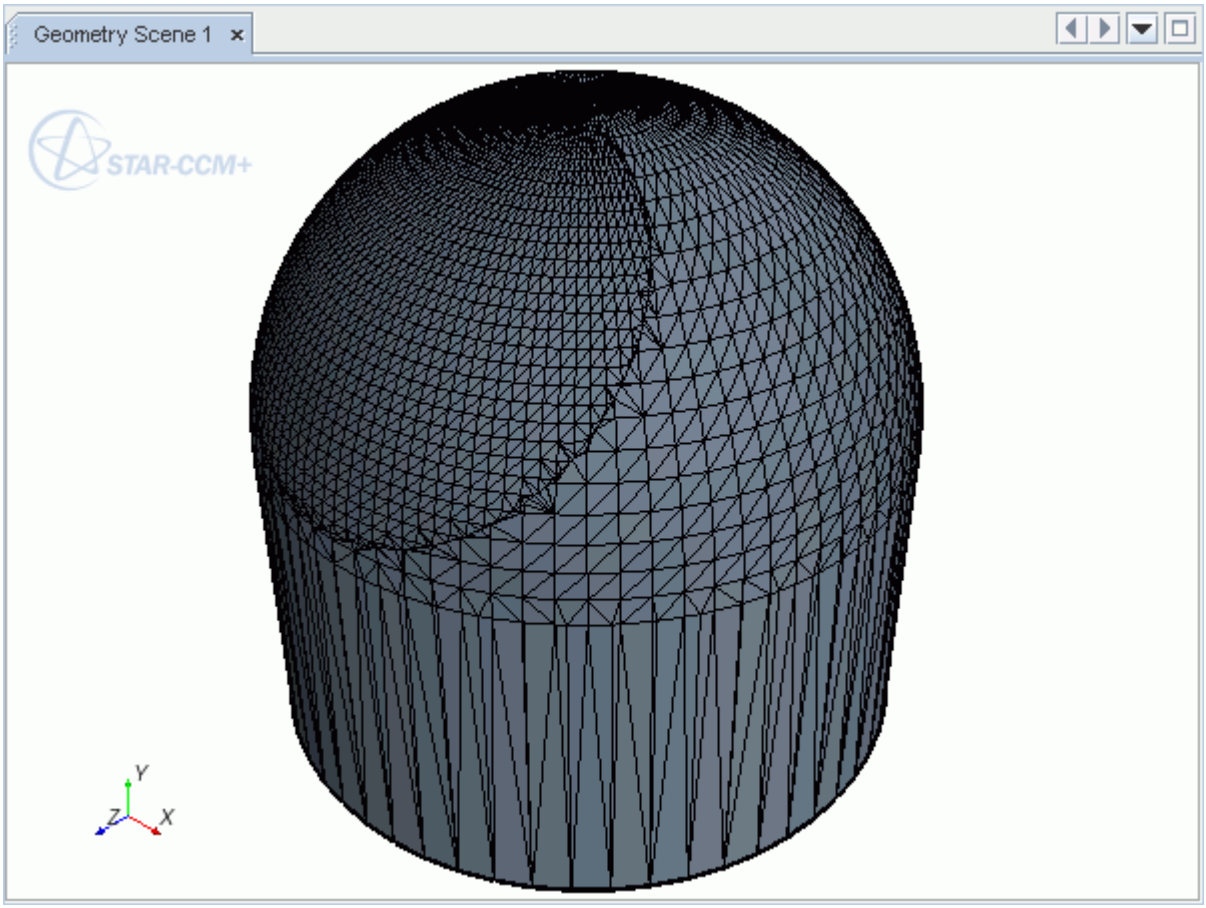
Maintaining Local Curvature

Selecting the *Maintain Local Curvature* option fills a hole with a surface that follows the surrounding curvature.

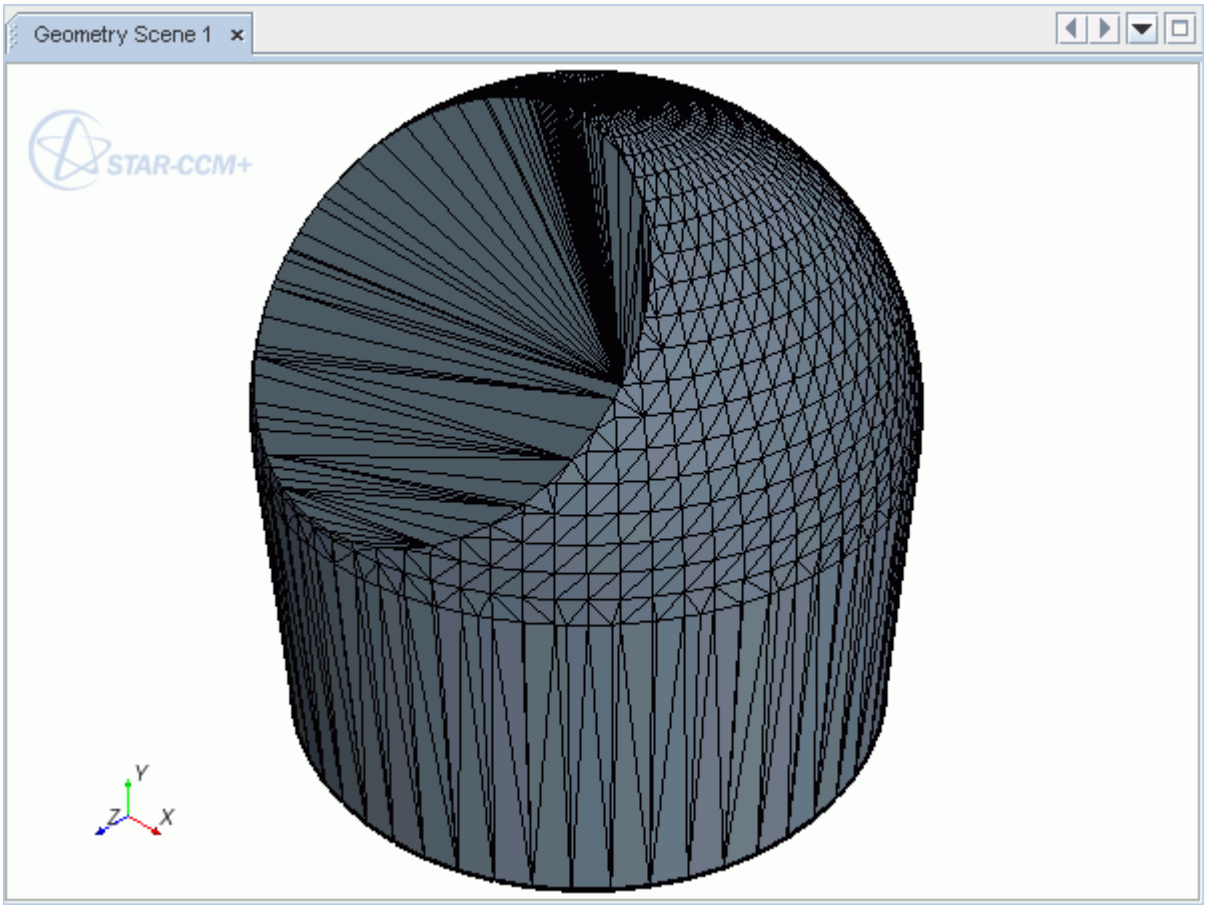
For example, the hole in the geometry below requires a filled surface that is not flat, but curves in each direction.



When the *Maintain Local Curvature* option is active, STAR-CCM+ generates the following surface.



For comparison, deactivating the *Maintain local curvature* option generates the following surface.



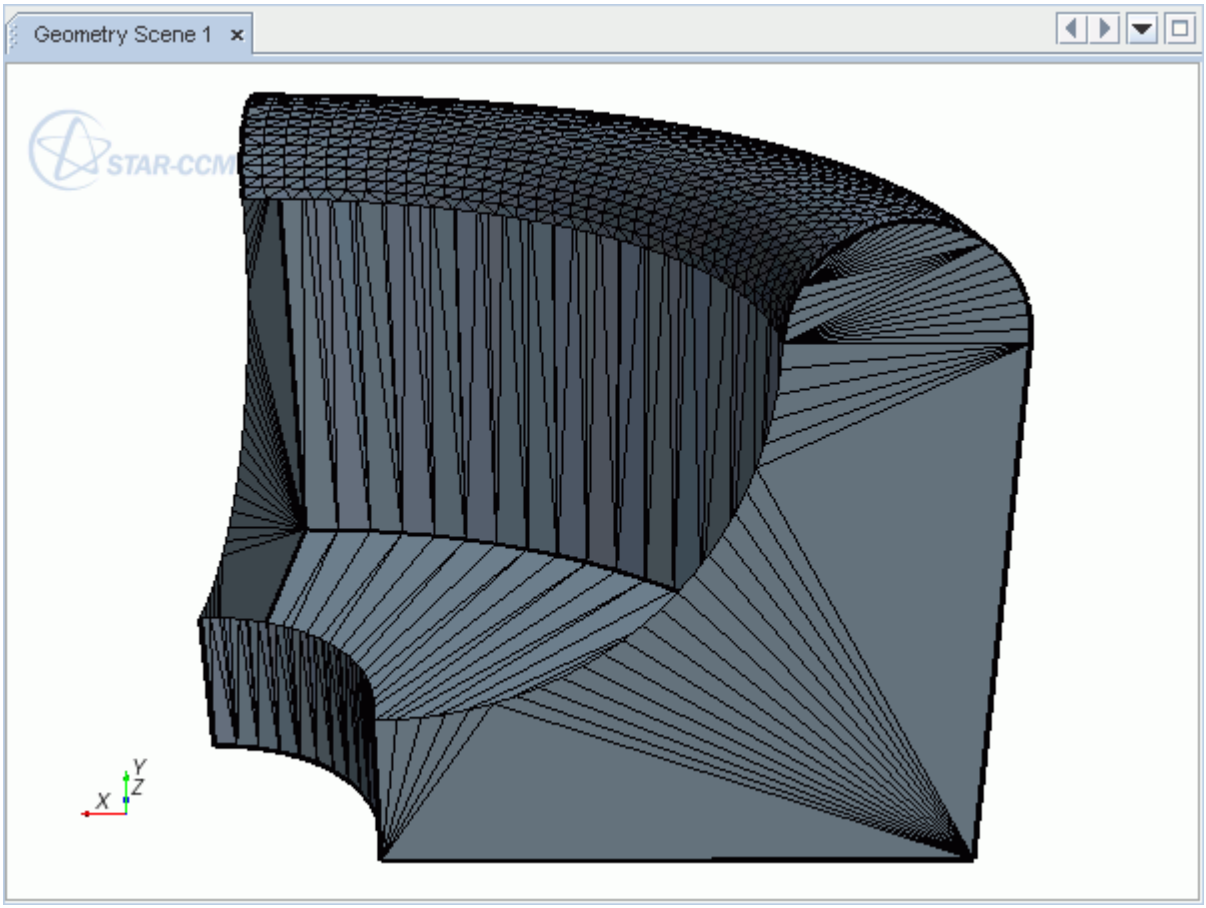
This feature produces the best results when faces surrounding the hole are associated with underlying CAD geometry. The CAD association reduces the amount of approximation that is required in generating the filled surface. If at least one of the surrounding faces has CAD association, then the filled surface also has CAD association, which allows you to project to CAD when surface meshing.

When the **Fill All Holes Together** mode is selected, the **Maintain Local Curvature** option can not be activated. Therefore to ensure curvature is maintained, set the *Loop Interaction Options* to Fill Each Hole Separately. Select each hole, then click (**Fill holes**).

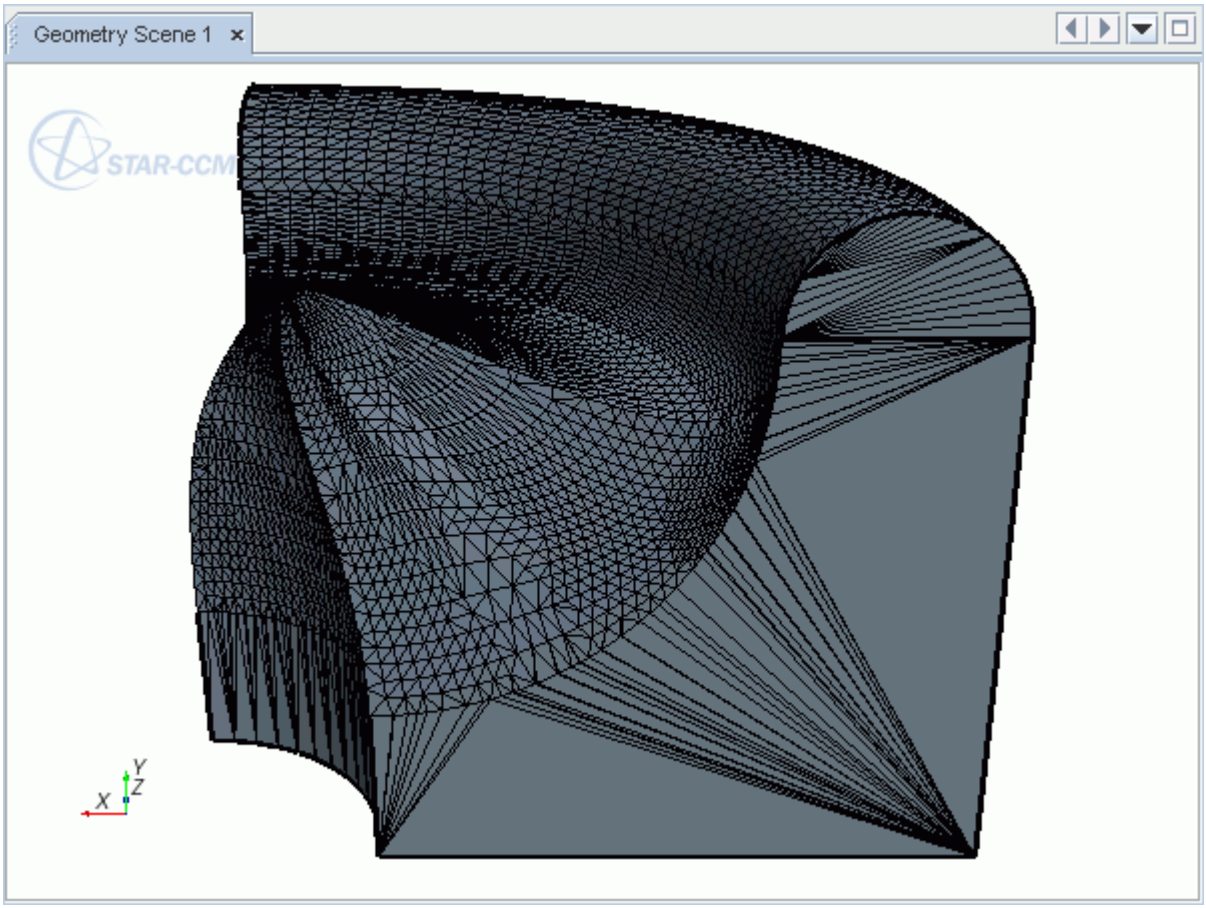
If for any reason the tool cannot maintain local curvature, it generates a filled surface without local curvature.

Using Feature Curves to Control the Filled Surface Shape

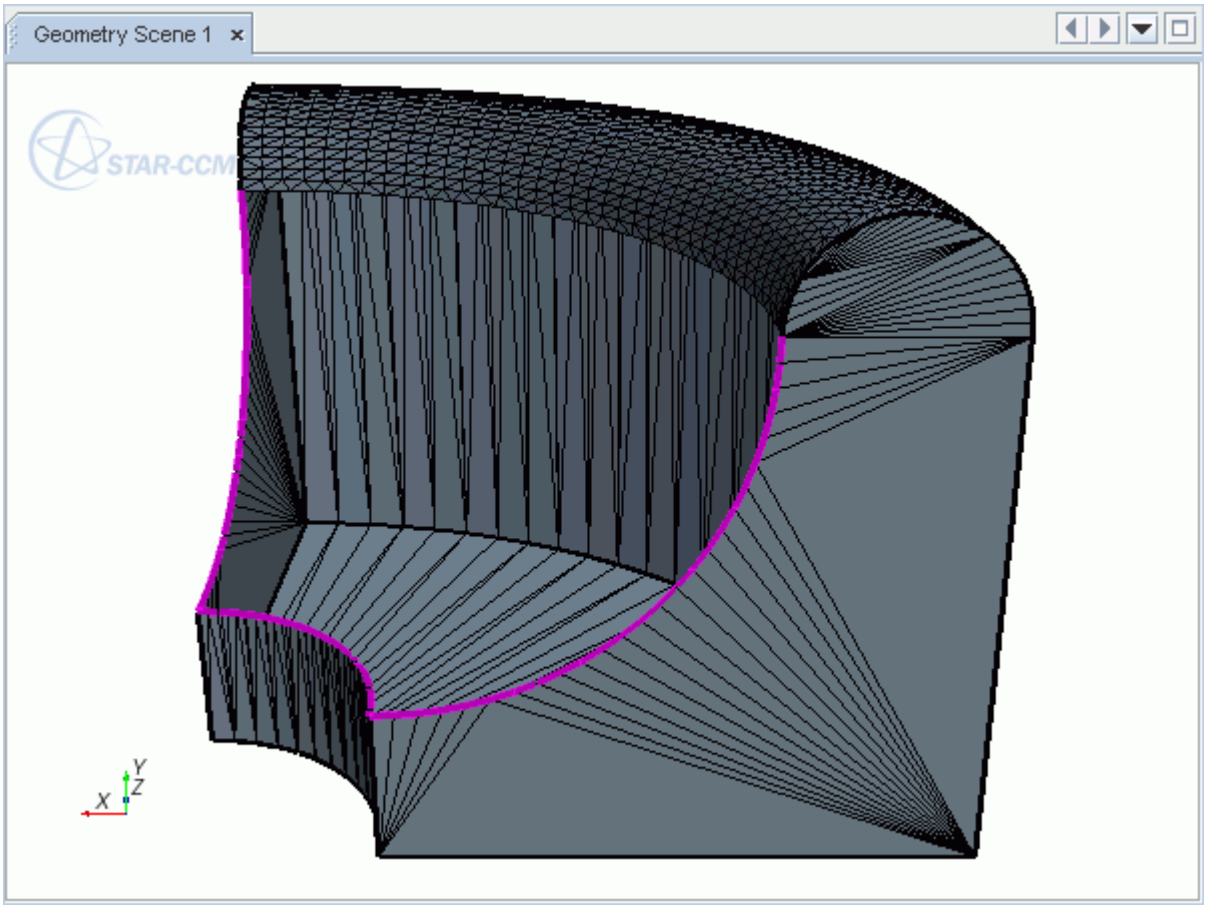
As a rule, mark sharp edges as feature curves before using the **Maintain Local Curvature** option. Applying feature curves to the edges surrounding the hole controls the construction of the filled surface. In the example below, maintaining local curvature of all surrounding edges generates an incorrect filled surface.



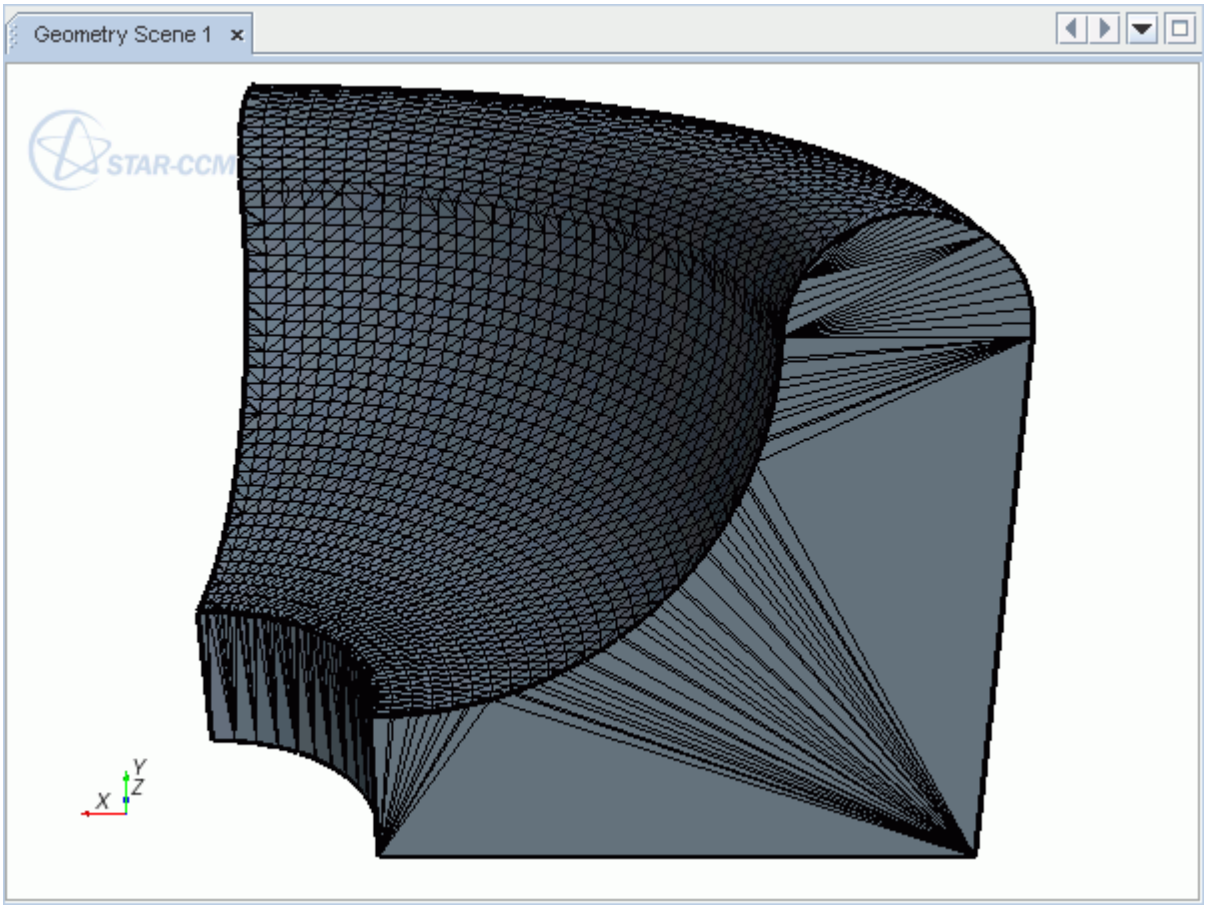
Filling the hole without marking any hole edges as feature curves generates the following filled surface. In this case, the tool maintains the curvature of the flat surfaces, which makes the filled surface initially tangential to each side. This set-up produces an incorrect shape.



Marking sharp edges as feature curves tells the hole filler to ignore local curvature around them. For this example, three sides of the hole (highlighted below) are marked as feature curves.



Running the hole filler with these edges marked as feature curves generates the correct surface.



See Also:

Loop Interaction Options

Surface Mesher Controls and Values

Open Loop Closure Options

Choose how to close open loop.

There are two methods to close open loops.

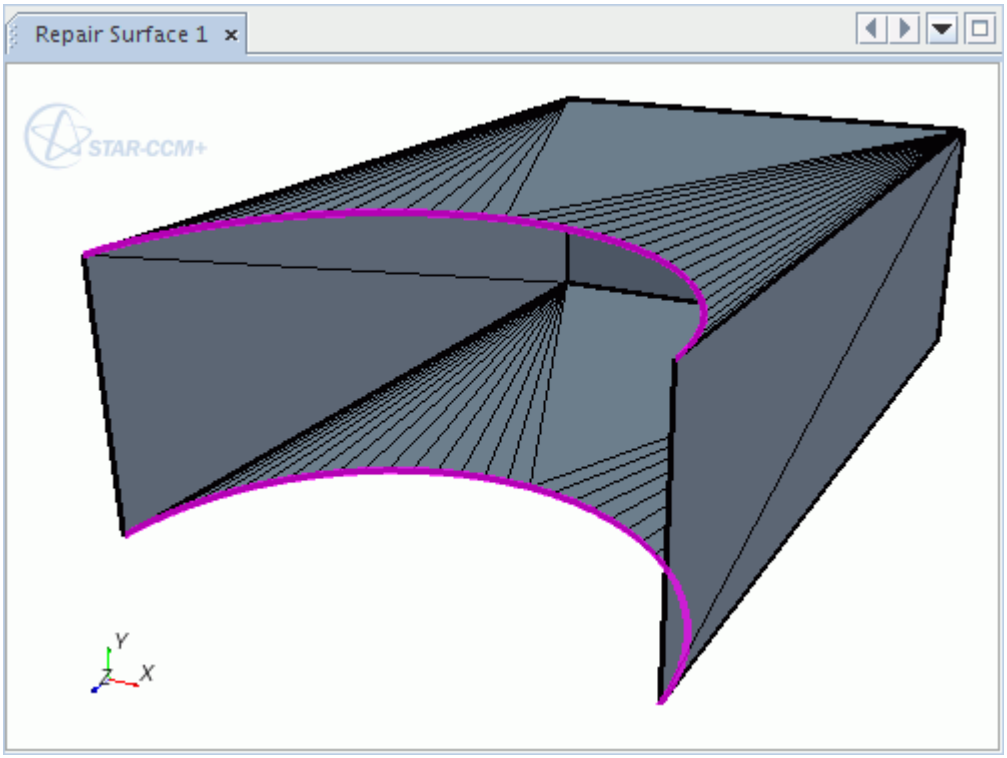
Fill Each Open Loop Separately

This option individually fills the holes defined by each open loop of edges.

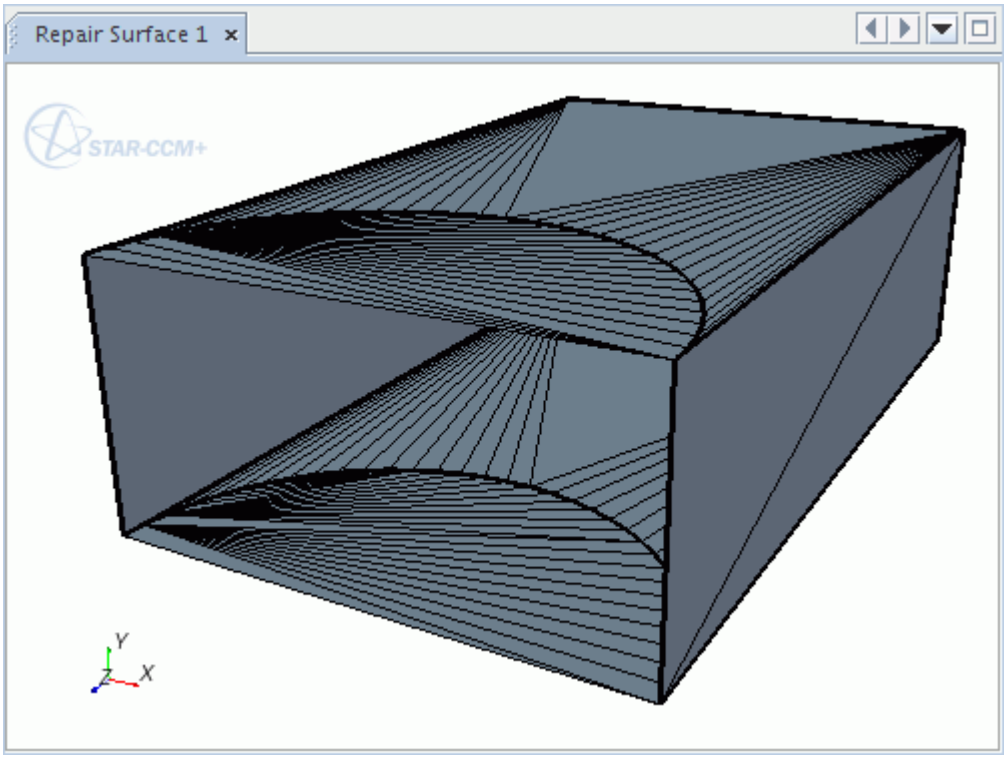
You can fill any number of open loops simultaneously.

Note: When filling separate open loops, the tool skips loops with collinear edges (straight lines) and loops with a single edge. Also, the tool checks for Y-junctions and cancels the command when it finds one.

For example, the following image shows two open loops.



Choosing to fill each open loop separately generates the following surface.

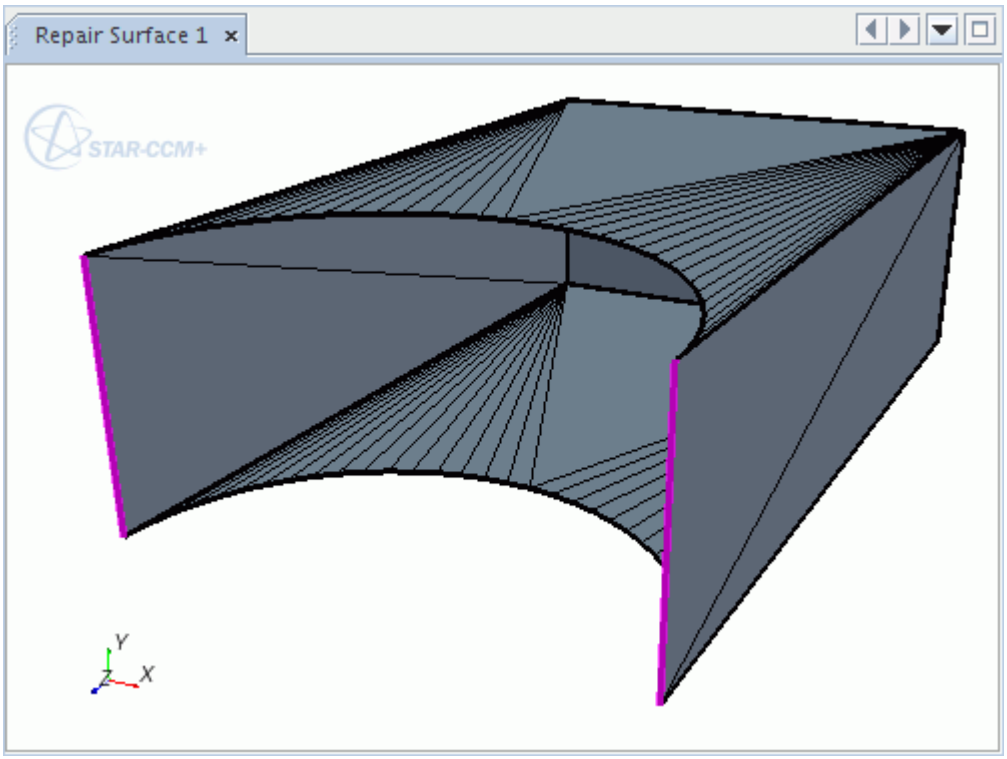


Fill Between an Open Loop Pair

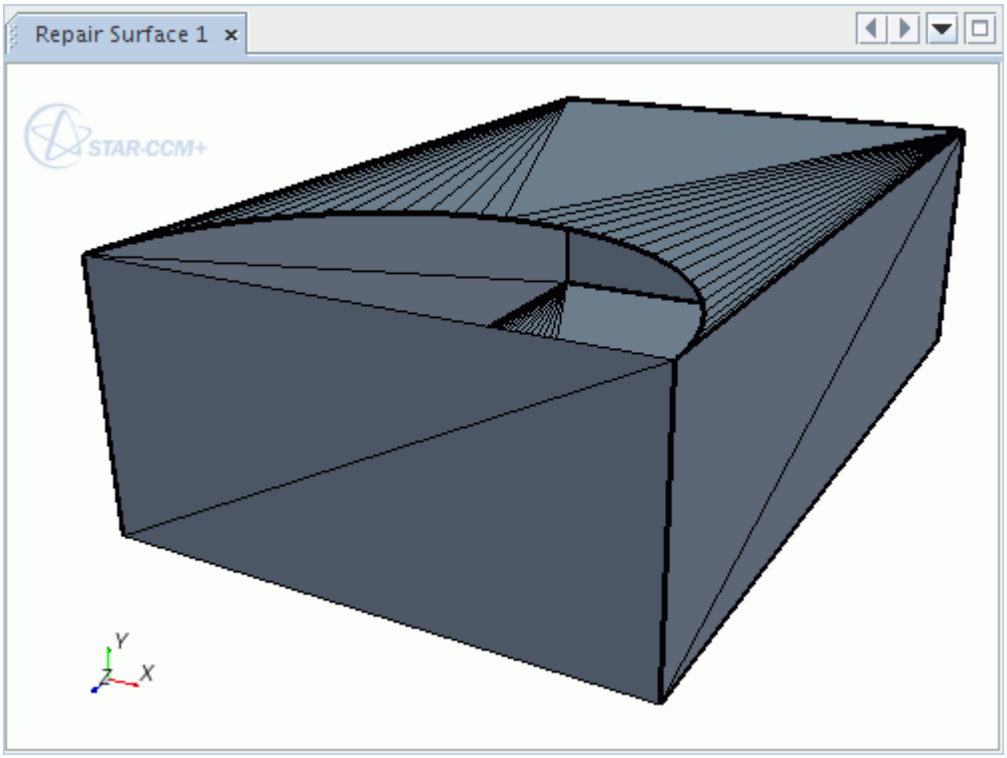
This option fills the hole between a pair of open loops.

Note: When filling a pair of open loops, you can only select two open loops; the tool cancels the command if you select more or less. Also, the tool checks for Y-junctions and cancels the command when it finds one.

For example, the following image shows two open loops.



Choosing to fill between an open loop pair generates the following surface.



Loop Interaction Options

You can fill all holes together, or separately.

- **Fill All Holes Together**

  If you have two loops, one inside the other, the **Fill All Holes Together** option will fill between the two loops. This option does not let you maintain local curvature.
- **Fill Each Hole Separately**

  If you have two loops, one inside the other, the **Fill Each Hole Separately** option will fill each loop separately and will not fill between the two loops. This option does let you maintain local curvature.

Repairing Features

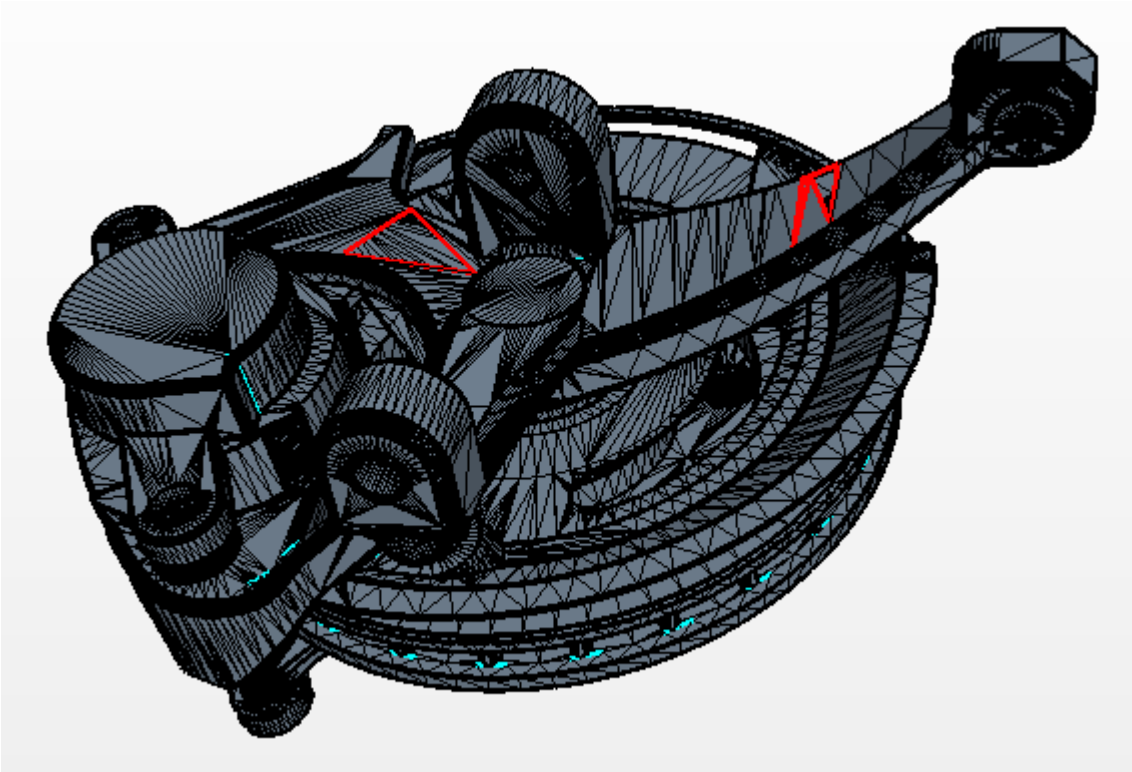
Diagnose and fix feature curves before running the surface remesher.

A typical workflow to follow to fix feature curve errors is presented below:

1. In the *Feature Repair* section of the *Repair* tab, use the (**Mark features**) tool to mark features by edge angle, free edges, non-manifold edges, patch perimeters, and boundary perimeters. See Repair Tools Reference: Mark Features Tool.
2. In the Surface Repair tool, in the *Diagnostics* slide out, select **Feature** and click **Execute** to run the diagnostics.

   The tool analyzes the geometry for surface errors and displays the results in the *Diagnostics* panel.

Errors are classified into color-coded categories. The colored boxes in the *Diagnostics* panel report the number of times that a category of error is found in the active model. The same colors are used in the active scene to highlight the faces and edges on which the errors were found.



See Feature Errors for a list of errors.

As you fix errors, the number in each category is updated to indicate the remaining number of items for which that category of error is found. It is common for the number of one error type to increase while fixing another error type. For example, deleting faces to remove a pierced surface temporarily increases the number of free edges.

3. Use the (**Not a feature error**) tool to mark features that are incorrectly identified as errors as valid features. See Repair Tools Reference: Not a Feature Error Tool.

4. Click (**Fix all feature errors**) for the Surface Repair tool to attempt to fix all feature errors.

5. Use the (**Flag edges as feature**) and (**Unflag feature edges**) to preserve or ignore geometry details respectively. See Repair Tools Reference: Flag Edge as Feature Tool and Repair Tools Reference: Unflag Edge as Feature Tool.

   You can select feature curves by using the browse buttons, or by using the mouse in the *Graphics* window. Selecting feature curves using the mouse allows greater accuracy when preserving or removing feature curves.

   As you flag and unflag feature curves, the feature diagnostics automatically update and can consider a previous error valid. Similarly, it is possible for the addition of feature curves, with the intention of fixing larger errors, to increase the error count temporarily.

   If there is a topology or geometry change while using the feature diagnostics tool, click **Execute** to recalculate the results.

Surface Repair Error Categories

The Surface Repair tool diagnoses the geometry for surface and feature errors and groups the various errors into color-coded categories.

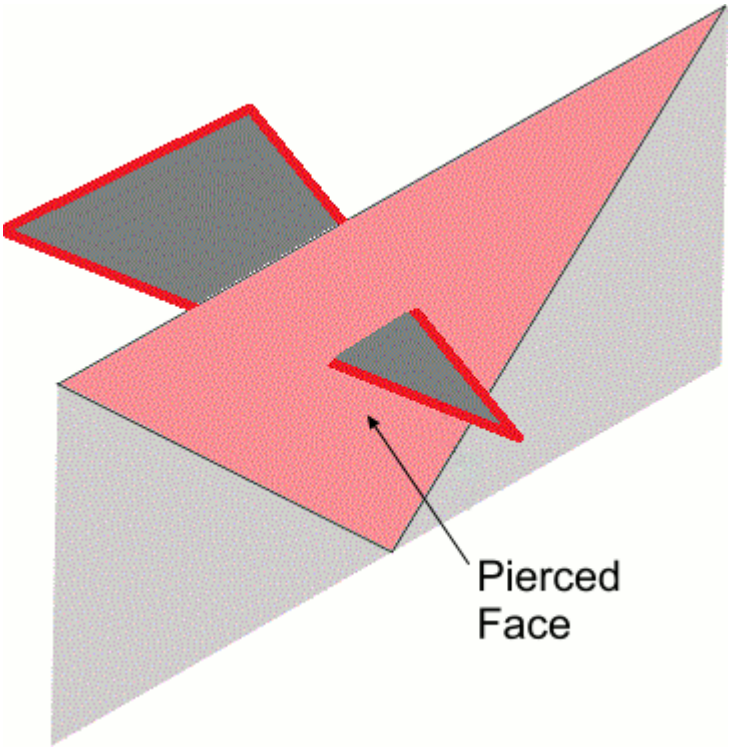
Surface Errors

The diagnostics categories for surface errors are listed below:

Pierced Faces

A pierced face is a face that is intersected by one or more edges of another face.

If an edge pierces a face, the surface diagnostics color the face red in the *Graphics* window. If an edge is piercing a face, the surface diagnostics color the edge red.



You can fix pierced faces by using the surface wrapper, or by deleting one set of erroneous faces, using the surface repair tools. Deleting faces typically leaves free edges, which you must also fix.

Interaction Mode Specifics

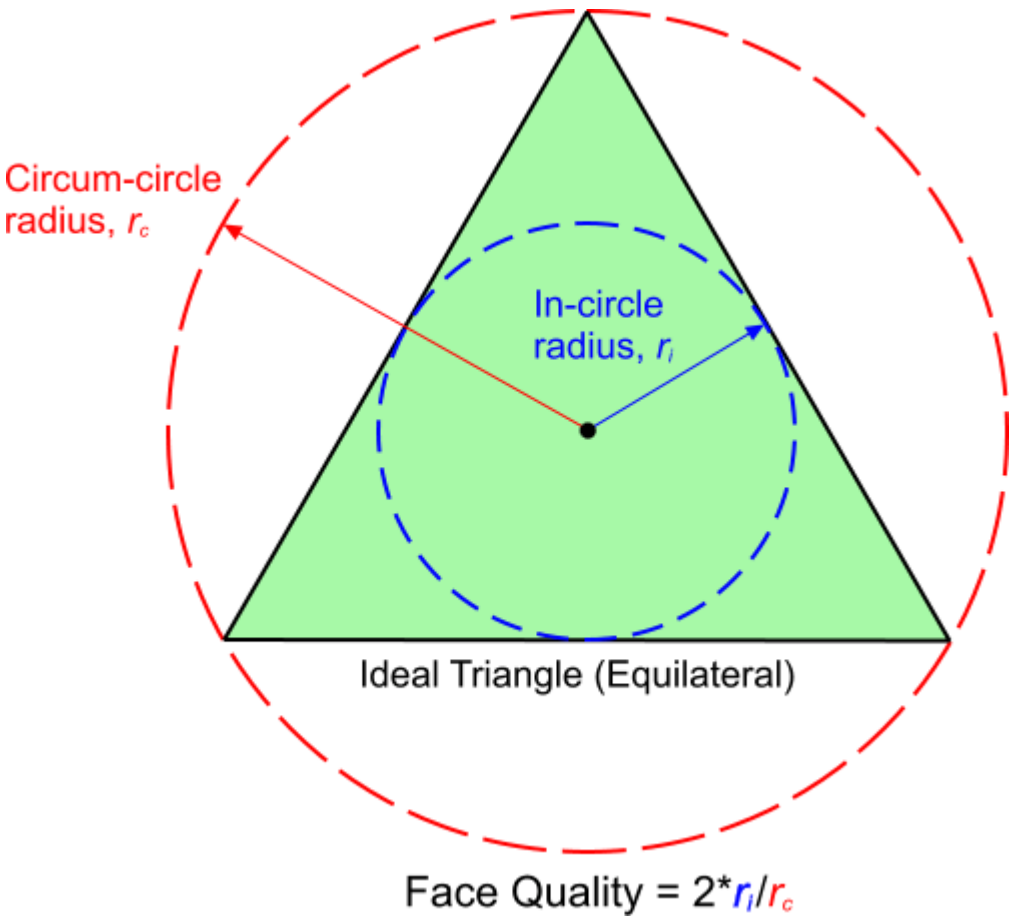
A per-part pierced face is one that has at least one edge from another face intersecting it and that face belongs to the same part. If the pierced face is in a contact, then it is considered to belong to the per-part result if faces attached to the piercing edges belong to either part making up the contact face.

A part to part pierced face is one that has at least one edge from another face intersecting it and that edge has at least one or more faces attached to it that belong to a different part. If the pierced face is in a contact (and therefore belongs to two parts), then it is considered a part to part pierced face.

Face Quality

Face quality is a measure of similarity between a face and the ideal face shape, which is an equilateral triangle.

The surface diagnostics calculate face quality as the ratio of the in-circle radius to the circum-circle radius, multiplied by a factor of 2. The in-circle of a face is the largest circle that fits inside the face, while the circum-circle is the circle that passes through the three points of the face. The diagram below illustrates these definitions for an ideal face.



Face quality ranges from 0 to 1, with 0 being a line (that is, no in-circle radius) and 1 being an equilateral triangle (that is, in-circle radius is half of circum-circle radius).

Faces that have a quality below 0.01 (by default) are considered poor quality faces. In **Surface Repair** mode, poor quality faces are colored in yellow when you activate the **Poor Quality Face** diagnostic.

There are instances when you do not need to fix poor quality faces, depending on how the surface is used. For example, if you use the surface with the surface wrapper or surface remesher, poor quality faces do not usually cause a problem.

However, if you use the surface directly for volume meshing, poor quality faces lead to poor quality or invalid volume cells. In this case, you can improve face quality by locally remeshing the poor quality faces, using the surface repair tools.

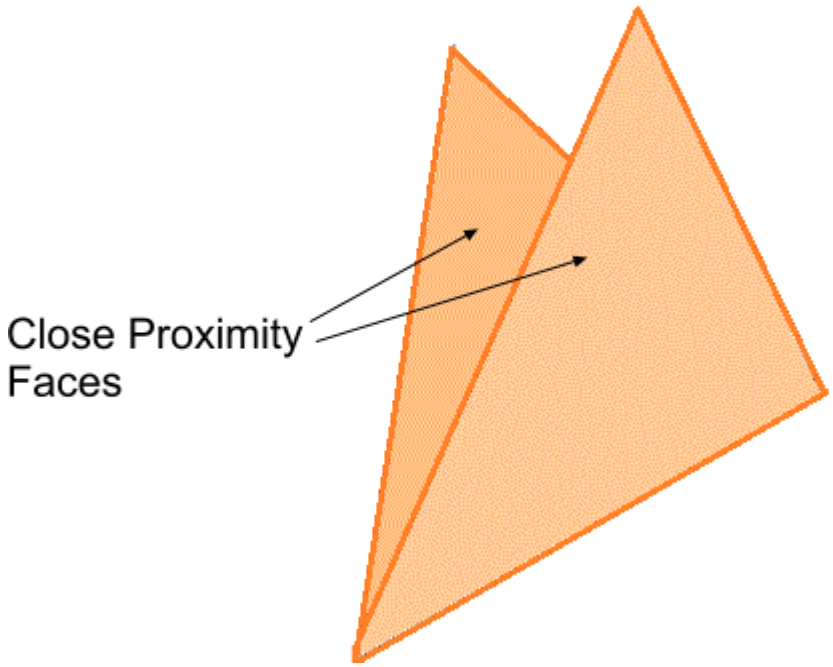
Face Proximity

Face proximity considers the distance between two faces.

Face proximity ranges from 0 to 1, with low values indicating one of the following problems:

- Folds in the surface, for example, “V” or “Z” shaped kinks in the surface causing localized overlaps
- Surfaces in close proximity, for example, two surfaces that are almost touching, such as either side of a thin baffle

Faces that have a face proximity below 0.05 (by default) are considered to be close proximity faces. In **Surface Repair** mode, close proximity faces are colored in orange when you activate the **Close Proximity Faces** diagnostic.



Depending on the nature of the problem, you can fix close proximity faces either by moving the faces further apart, or by smoothing the mesh locally, using the surface repair tools.

There are instances when you do not need to fix close proximity faces (for example, when you expect close proximity faces due to the nature of the geometry). However, pay attention to these areas during surface and volume meshing, to identify and correct any mesh quality problems.

Interaction Mode Specifics

When performing the face proximity calculation, STAR-CCM+ starts with one face and then considers all other neighbouring faces one at a time. If a face pair meets the proximity criteria, then the starting face is considered to be the "original" face and the other face in the pair is considered to be the "opposing" face.

A *per-part* close proximity face is one that has the opposing face belonging to the same part as the original face. If either the original face and/or opposing face are in a contact, then the result is considered per-part if a common part is found between the two faces.

A *part-to-part* proximity face is one that has the opposing face belonging to a different part to the original face. If either the original face and/or opposing face belong to a contact, then the face is considered to belong to the part to part result.

Bi-Directional Proximity

By default, face proximity is calculated for inward opposing normals only. In some cases, where faces are extremely close together, the checks in the volume mesher pipeline can fail. These close faces can pass the default proximity check, where the inward face normals are not opposing. Activate the **Use Bi-Directional Proximity** option, in the diagnostic slide-out, to calculate the face proximity for both directions (inward and outward facing normals). The bi-directional proximity check will then flag these close faces.

You can visualize the direction of face normals using the **Face Normal** color mode. Face normals that point outward are colored green, while those that point inward are colored red. See Embedded Tools Reference: Color Modes.

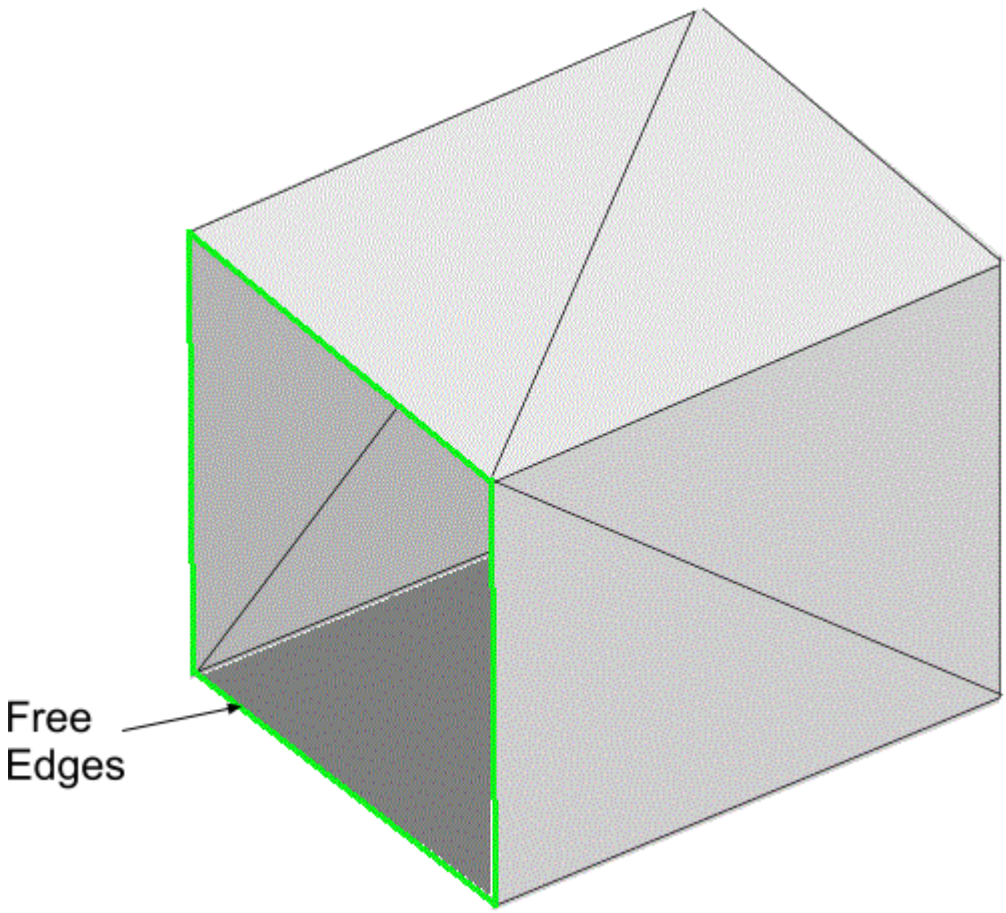
Duplicate Faces

Duplicate faces exist when the three vertices of a given triangle are shared by another triangle, potentially in a different part. All duplicate faces are close proximity faces. To view duplicate faces only, create a custom diagnostic and set the *Type to* **Duplicate Faces**. See Custom Threshold Options.

Free Edges

A free edge is one that is joined to only one face. Two joined faces share two vertices for the common edge.

For example, edges around a surface hole are marked as free edges, as is illustrated below:



The surface diagnostic colors **Free Edges** green in the *Graphics* window.

Double-clicking a free edge selects all connected free edges. This feature is useful when fixing free edges using the fill holes tool.

An understanding of how the free edges are calculated is required when you are diagnosing multiple parts.

You can fix free edges by filling or patching the hole using the surface repair tools.

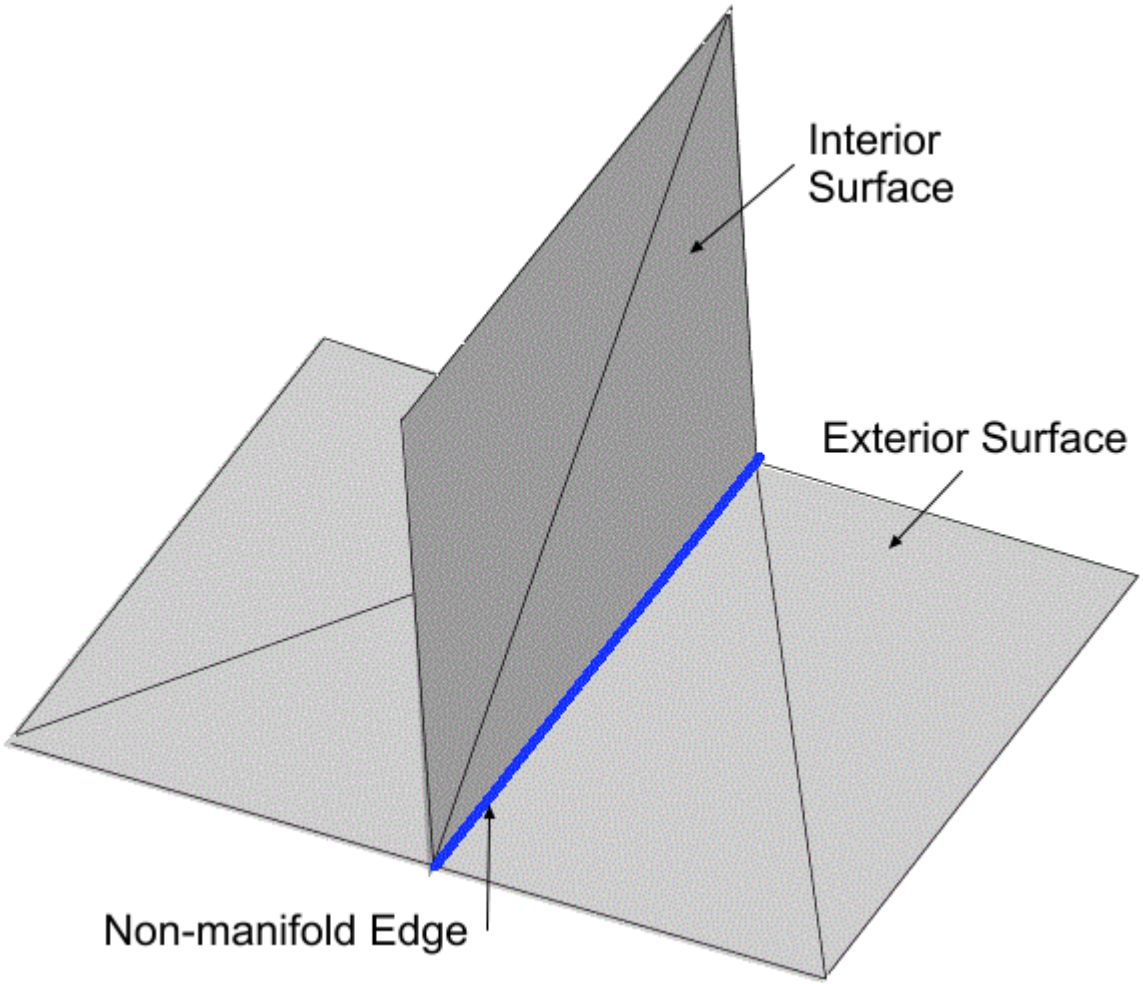
There are instances when you do not need to fix free edges. For example, a free-floating baffle has the outside edges marked as free edges, until you convert the baffle to an interface. As long as you are aware of the error, and convert the baffle to an interface before surface and volume meshing, you can ignore the free edges.

You can create a custom control to show global free edges. A global free edge is one that belongs to one face only and considers edges at part-to-part perimeters to be connected. See Custom Thresholds: Global Free Edges.

Non-Manifold Edges

A non-manifold edge is an edge that three or more other faces share.

For example, an interior surface joined to an exterior surface has non-manifold edges where the interior and exterior faces share an edge, as illustrated below:



The surface diagnostics color **Non-manifold Edges** blue in the *Graphics* window.

An understanding of how non-manifold edges are calculated is required when you are diagnosing multiple parts.

You can fix non-manifold edges by deleting the faces that are causing the additional edges, using the surface repair tools.

There are instances when you do not need to fix non-manifold edges. For example, a baffle joined to the surface of a body has the shared edges marked as non-manifold edges, until you convert the baffle to an interface. As long as you are aware of the error, and convert the baffle to an interface before surface and volume meshing, you can ignore the non-manifold edges.

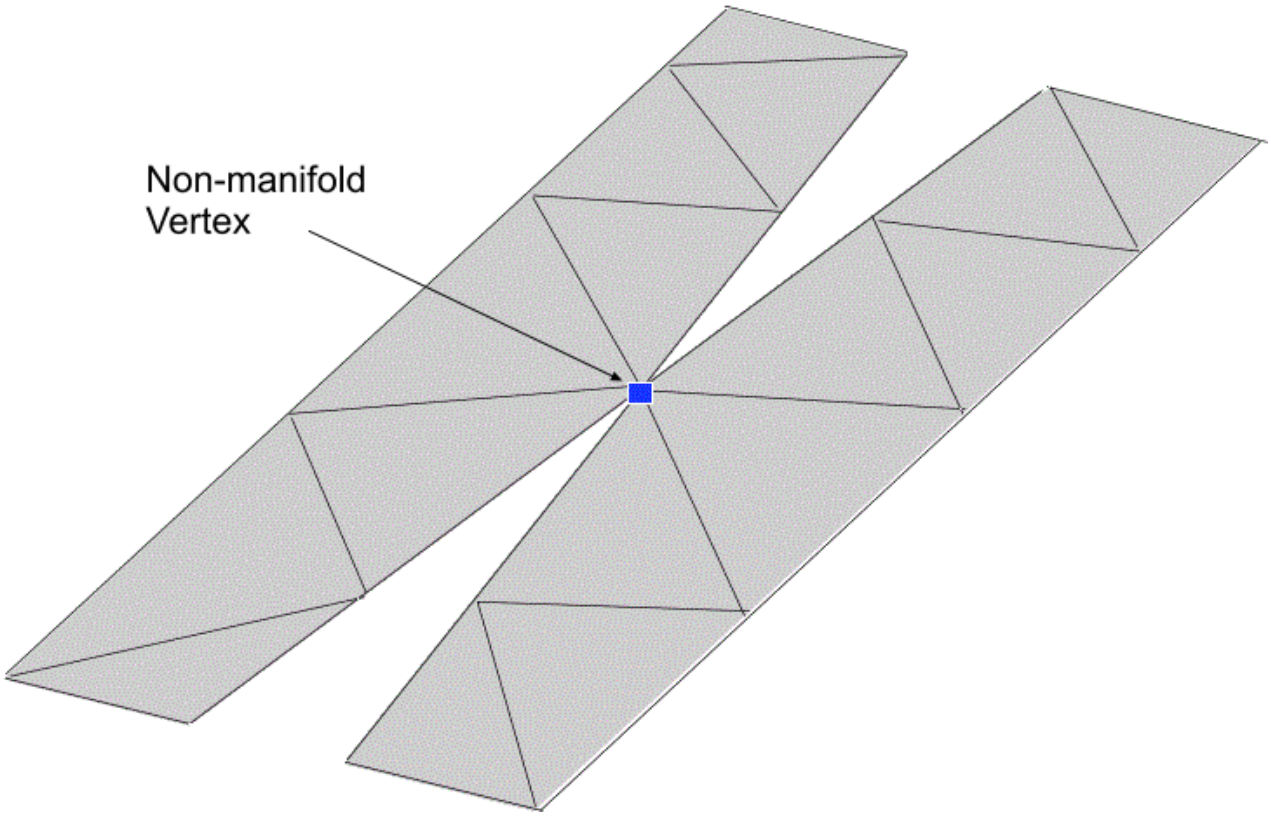
Interaction Mode Specifics

A per-part non-manifold edge is one where at least three faces attached to the edge belong to the same part. If one or more of the faces belongs to a contact, then either part that the contact face belongs to can be considered.

Non-Manifold Vertices

A vertex is non-manifold if any of its connected faces have no other link with the rest of its connected faces, apart from the vertex.

For example, the diagnostics identify a non-manifold vertex when one surface joins another at a single vertex, as illustrated below.



The surface diagnostics color **Non-manifold Vertices** blue in the *Graphics* window.

If the geometry requires connected surfaces, you can fix non-manifold vertices by filling holes or zipping edges, using the surface repair tools.

If the geometry requires disconnected surfaces, you can fix non-manifold vertices by disconnecting the faces that are causing the non-manifold vertex, using the surface repair tools.

Interaction Mode Specifics

A per-part non-manifold vertex is one where at least two faces attached to the vertex belong to the same part. If one or more of the faces belongs to a contact then, either part that the contact face belongs to can be considered.

Feature Errors

The diagnostic categories for feature errors are listed below:

Explicit Feature Error Edges

Explicit feature errors are typically undesirable part curves mixed among the explicit feature curves. Explicit feature curves are created during surface import, or during manual computation from edge angles and patches. The following types of error can occur:

- An excess of feature curves converging on a point
- Disconnected or isolated curves
- Over-constrained curves close to one another
- Missing feature curves

Implicit Feature Error Edges

STAR-CCM+ calculates implicit feature curves at the start of the meshing pipeline using boundary perimeters, free edges, and non-manifold edges that exist on the surface. The features diagnostics tool can identify such errors ahead of time, so you can fix them on the surface before starting the meshing process.

Repair Tools Reference

Use the repair tools to repair errors on the geometry surface and feature curves.

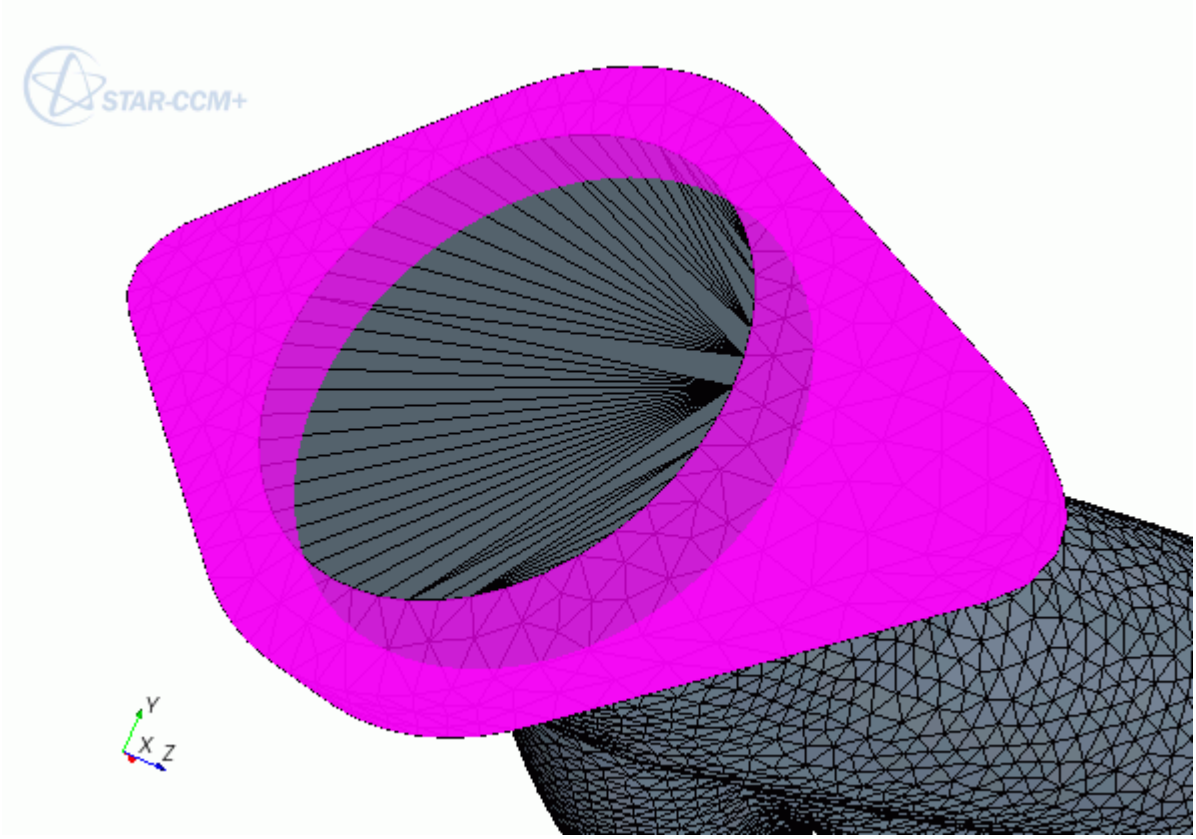
Surface Repair Tools

The Surface Repair tools repair errors on the geometry surface.

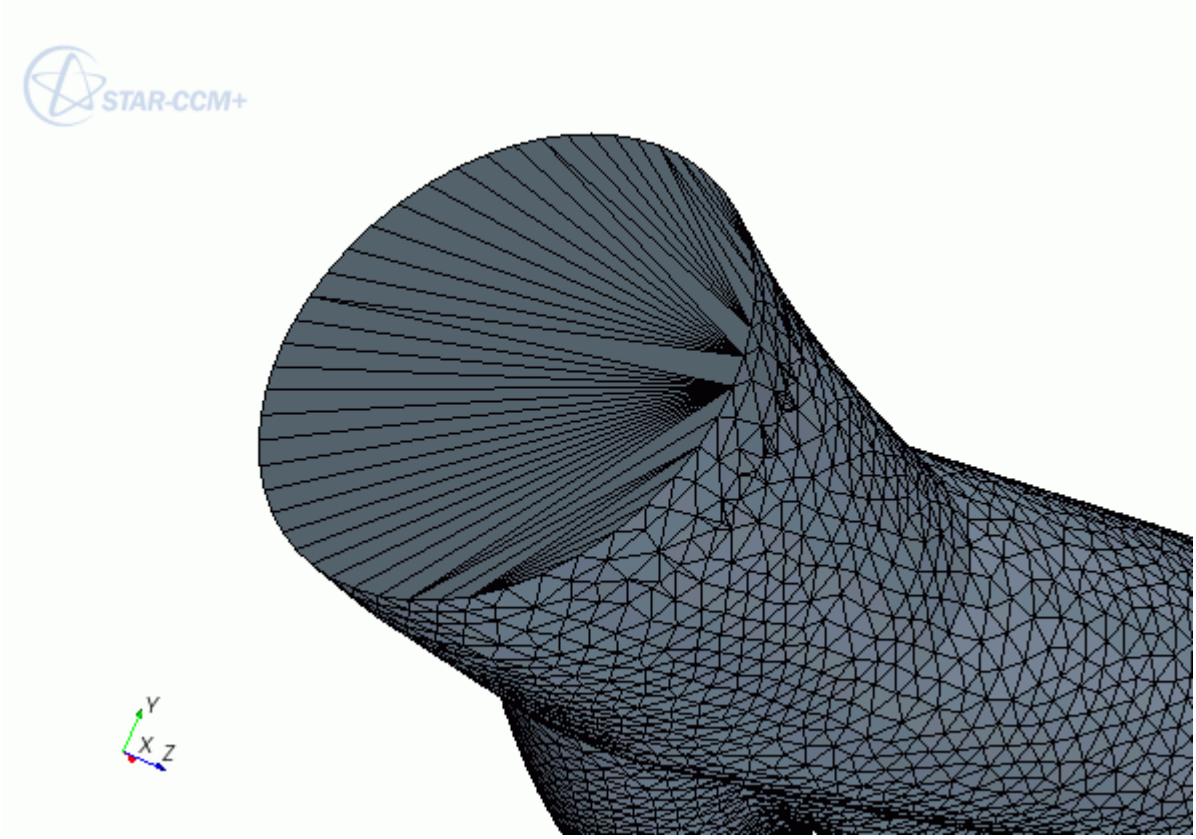
Delete selected faces

This tool deletes the selected faces from the part. The tool becomes available when you select at least one face.

The following example deletes unwanted faces from the solid part of a geometry.



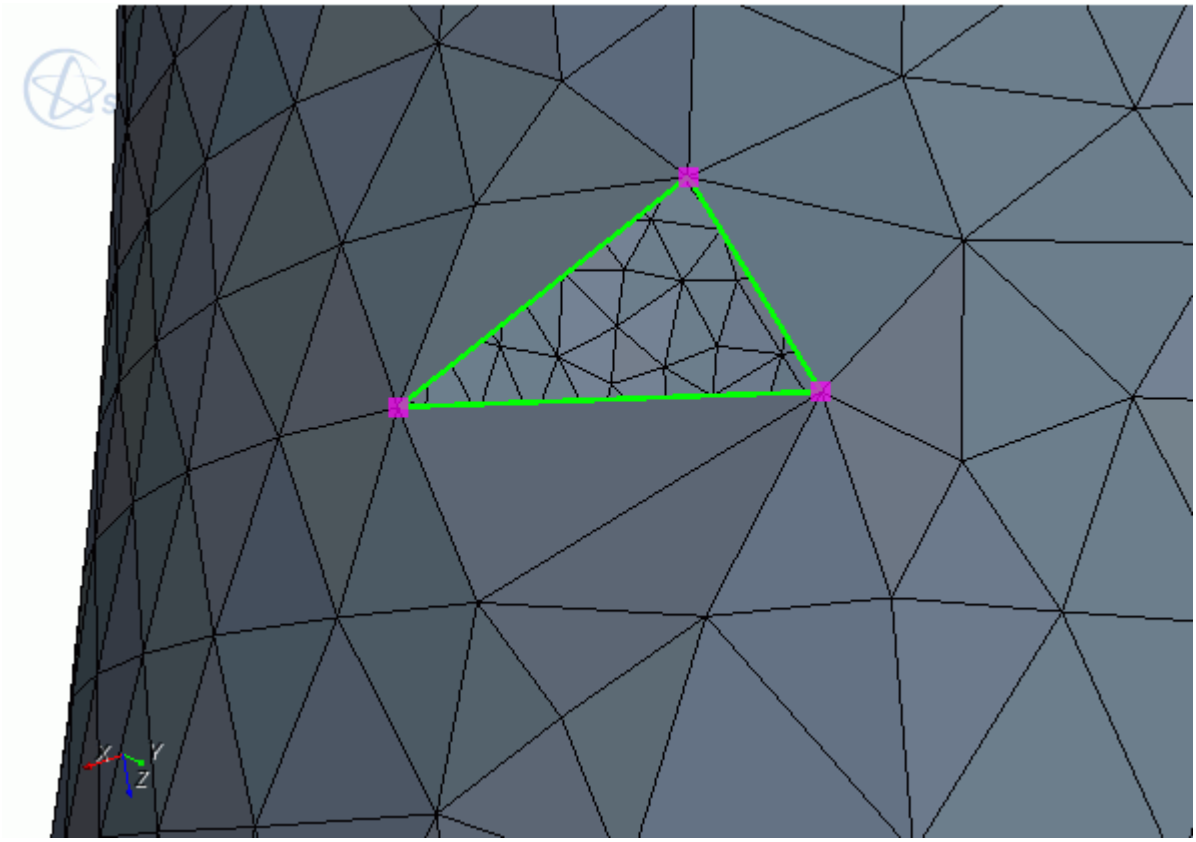
Clicking **Delete Faces** removes the extraneous faces, leaving the fluid volume.



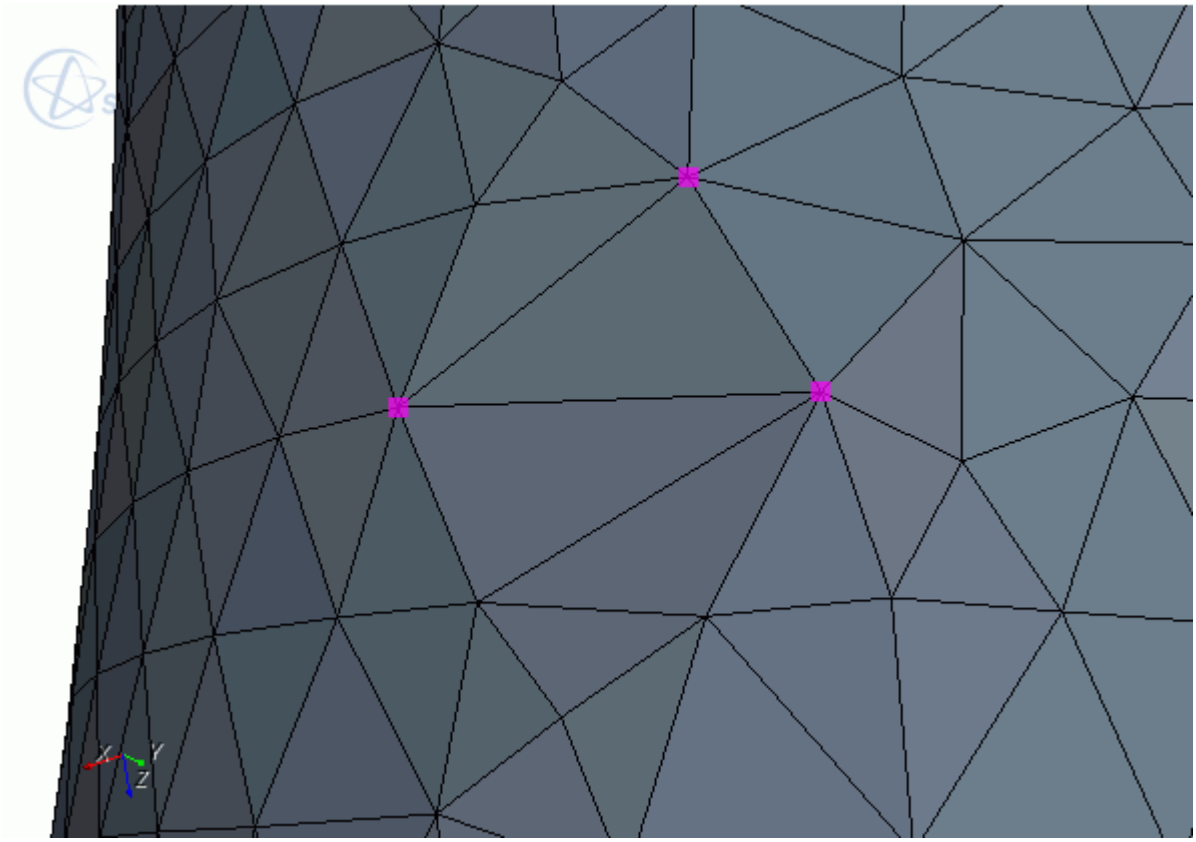
Create face from selected vertices

This tool creates a face using the selected vertices. The tool becomes available when you select three vertices.

The following example fixes free edges by creating a face. The three vertices are selected.



Clicking **Create Faces from Vertices** creates a face that is defined by the selected vertices.

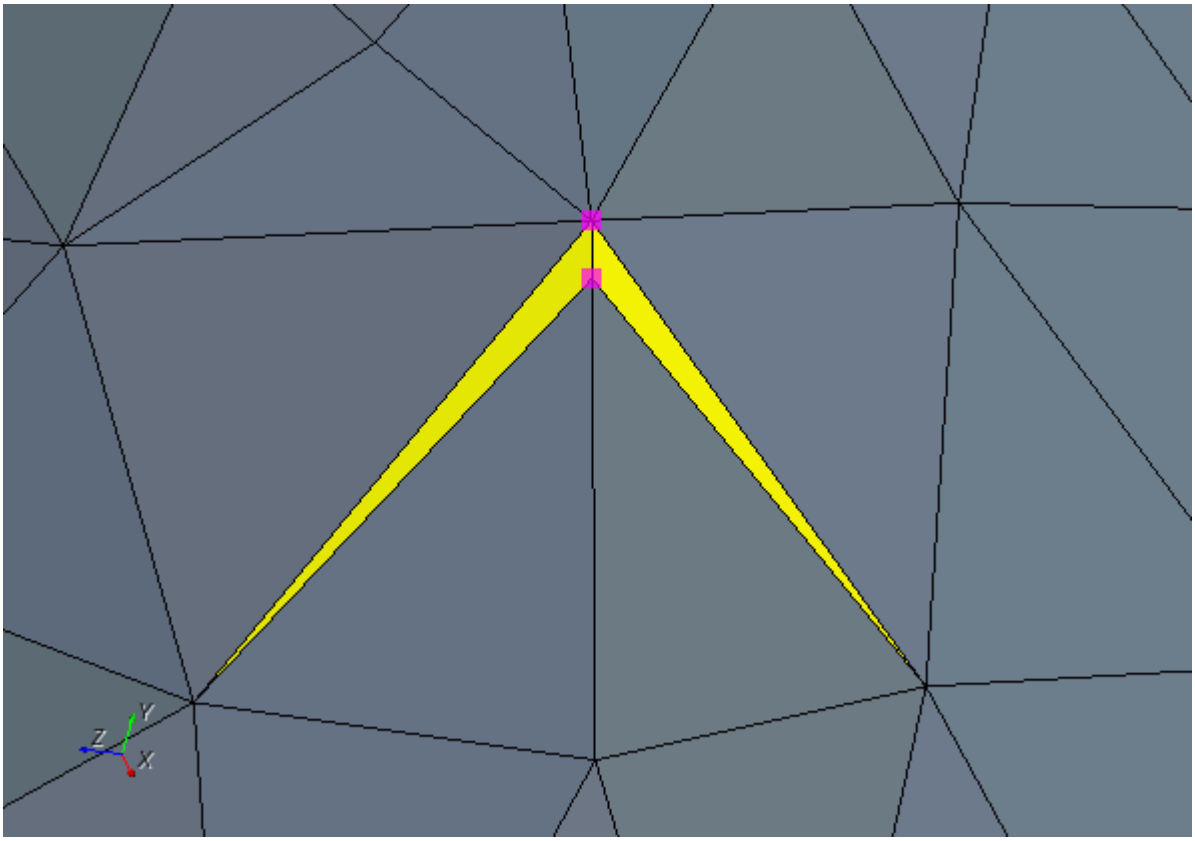


Collapse selected vertices

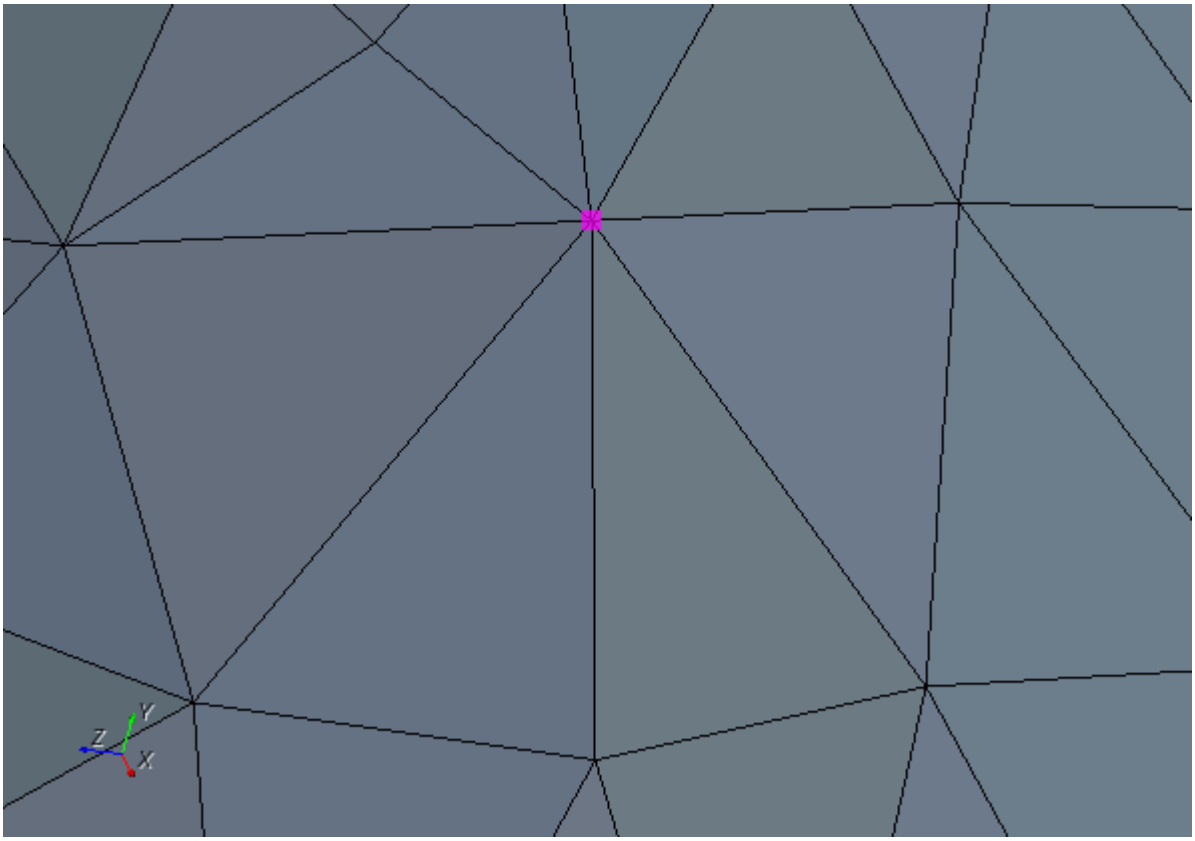
This tool collapses one vertex to the location of another.

The tool also collapses an edge or face to its mid-point. If you select an attached vertex with an edge or a face, the tool collapses the edge or face to the selected vertex. The tool becomes available when you select two vertices, an edge or a face (with an optional attached vertex for edges and faces).

The following example collapses two vertices, which fixes a poor quality face. You can encounter this scenario when fixing cracks on the surface using the zipping tool.



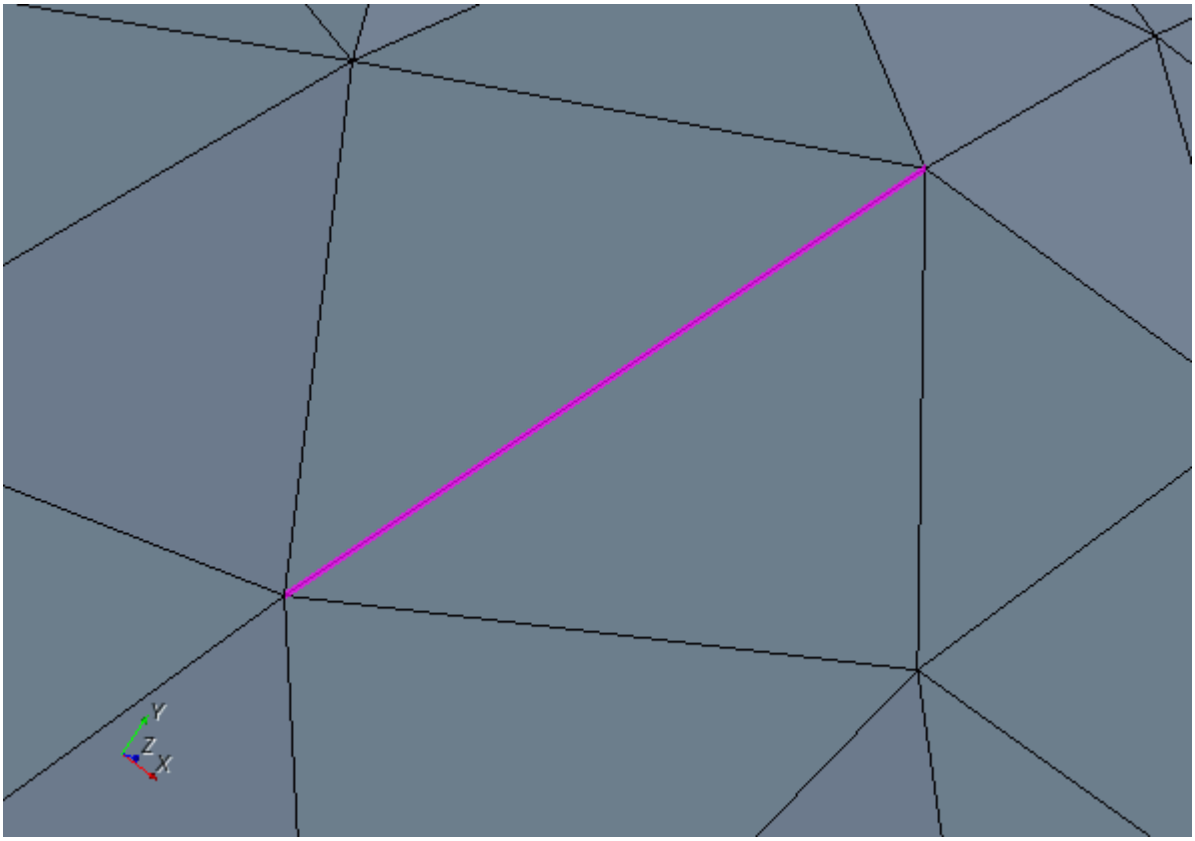
Using the **Collapse Vertices** tool removes the poor quality faces, without creating free edges.



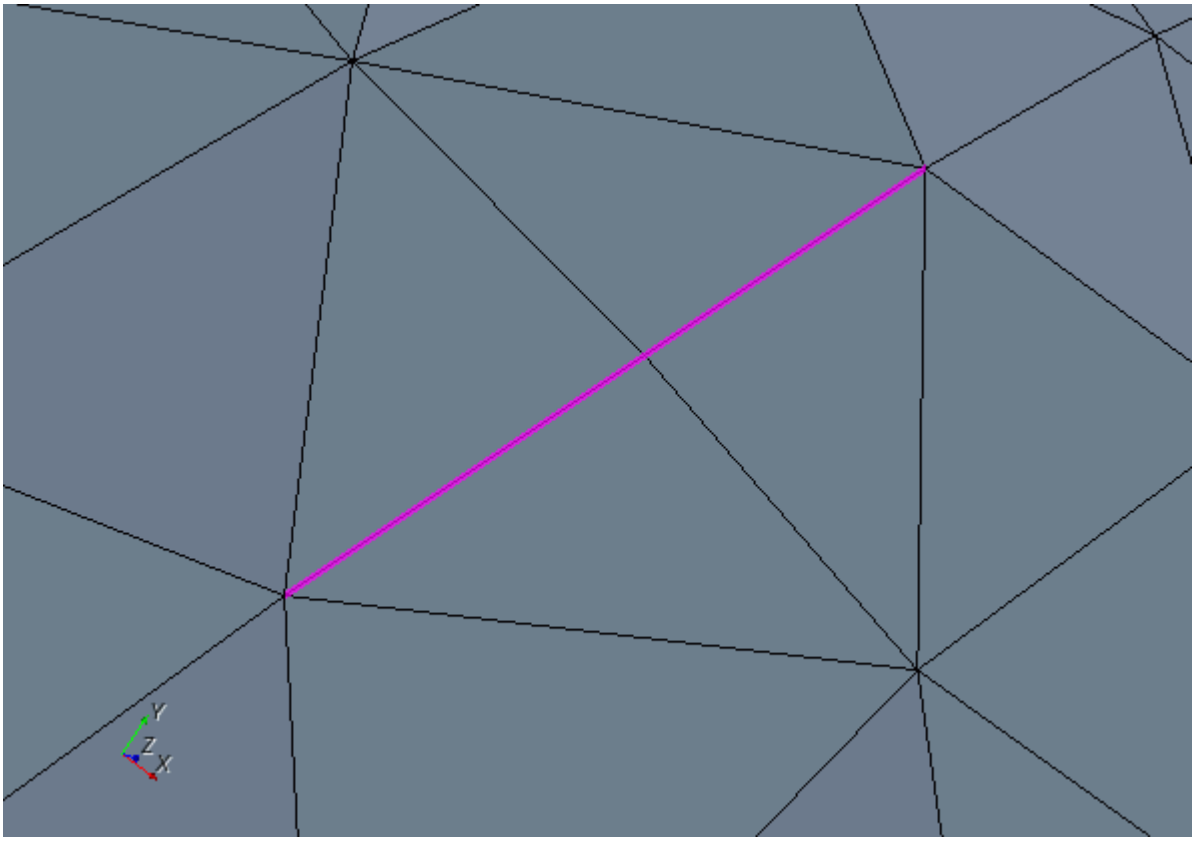
Split selected edges/faces

This tool splits a face or faces attached to an edge in half, using the mid-point of the selected edges. The tool becomes available when you select at least one edge or face.

The following example shows the splitting of two adjacent faces by splitting the connecting edge in half.



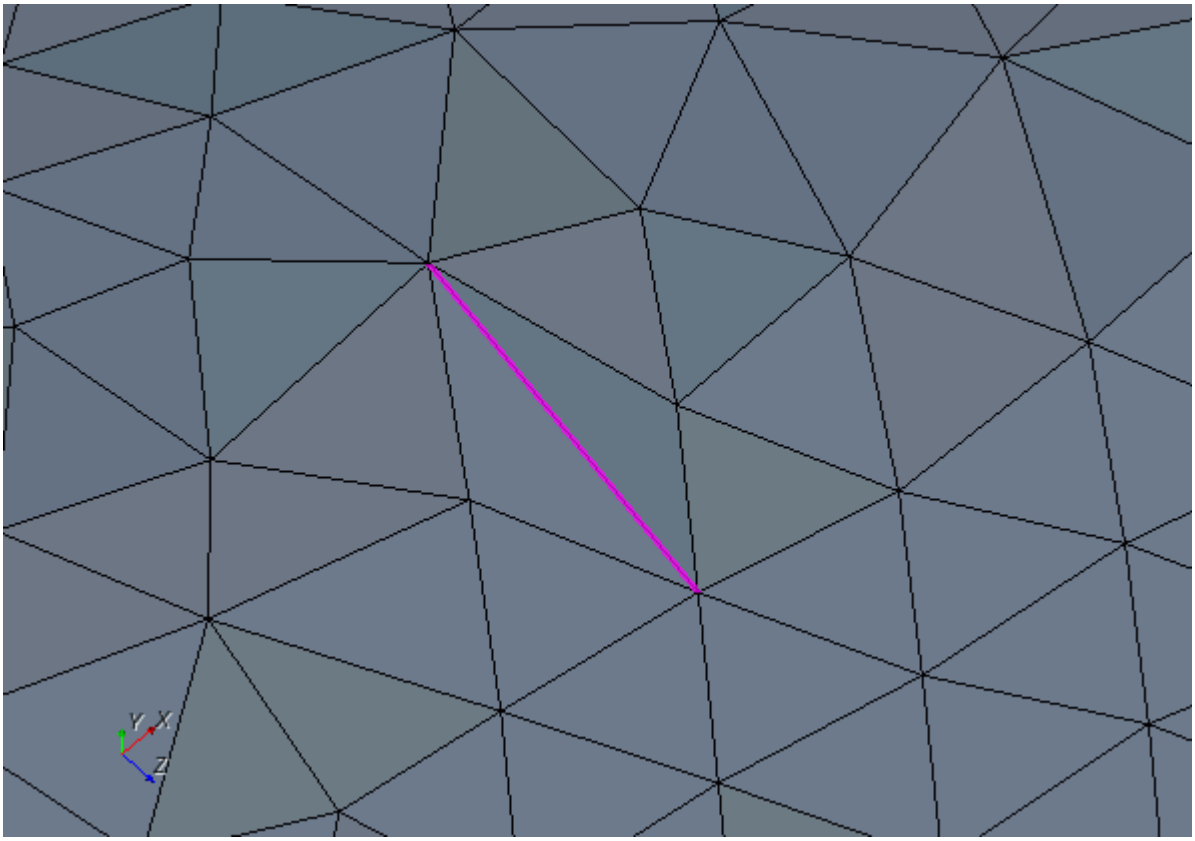
Using the **Split Edges** tool in this manner has the following results.



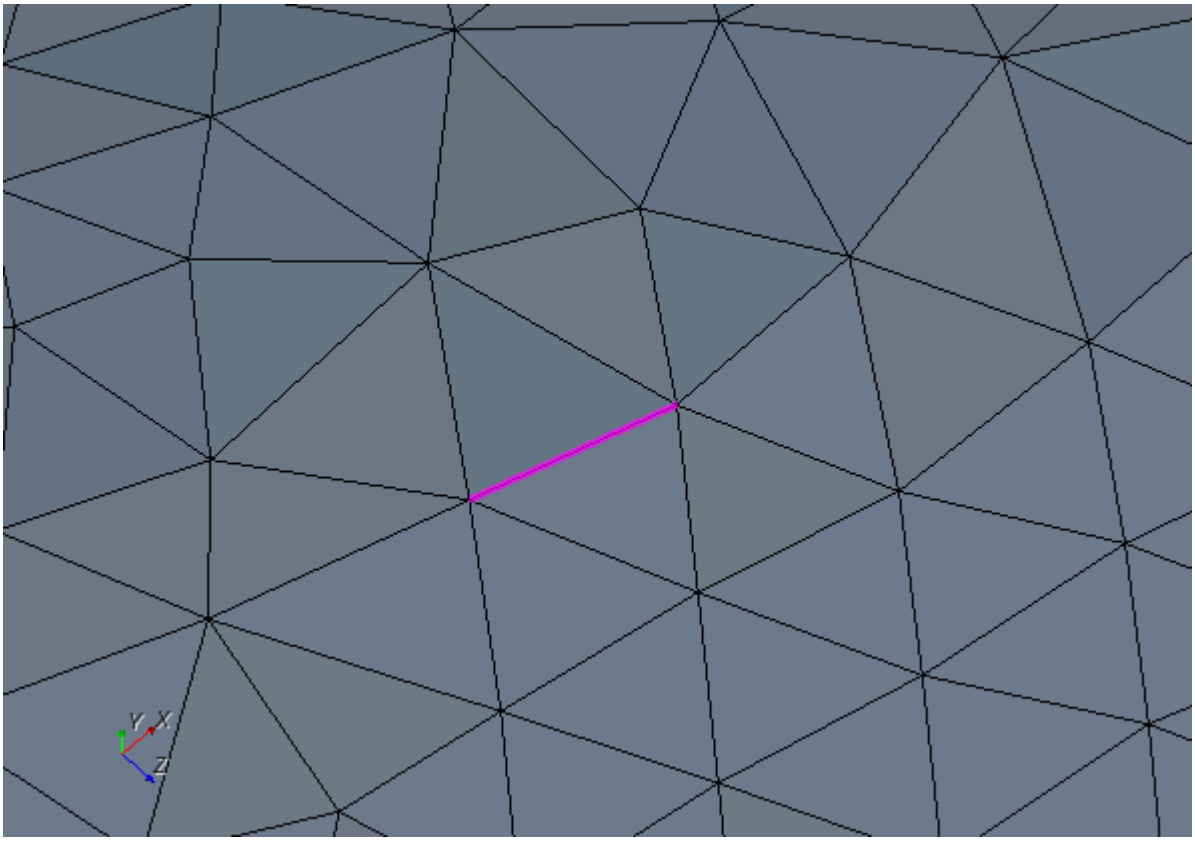
Swap selected edge

This tool swaps the connected edge between two faces. The tool is available when you have selected only one edge.

The following example swaps edges to improve face quality.



In this instance, the **Swap Edges** tool improves the quality of the connected faces, by reducing face skew.



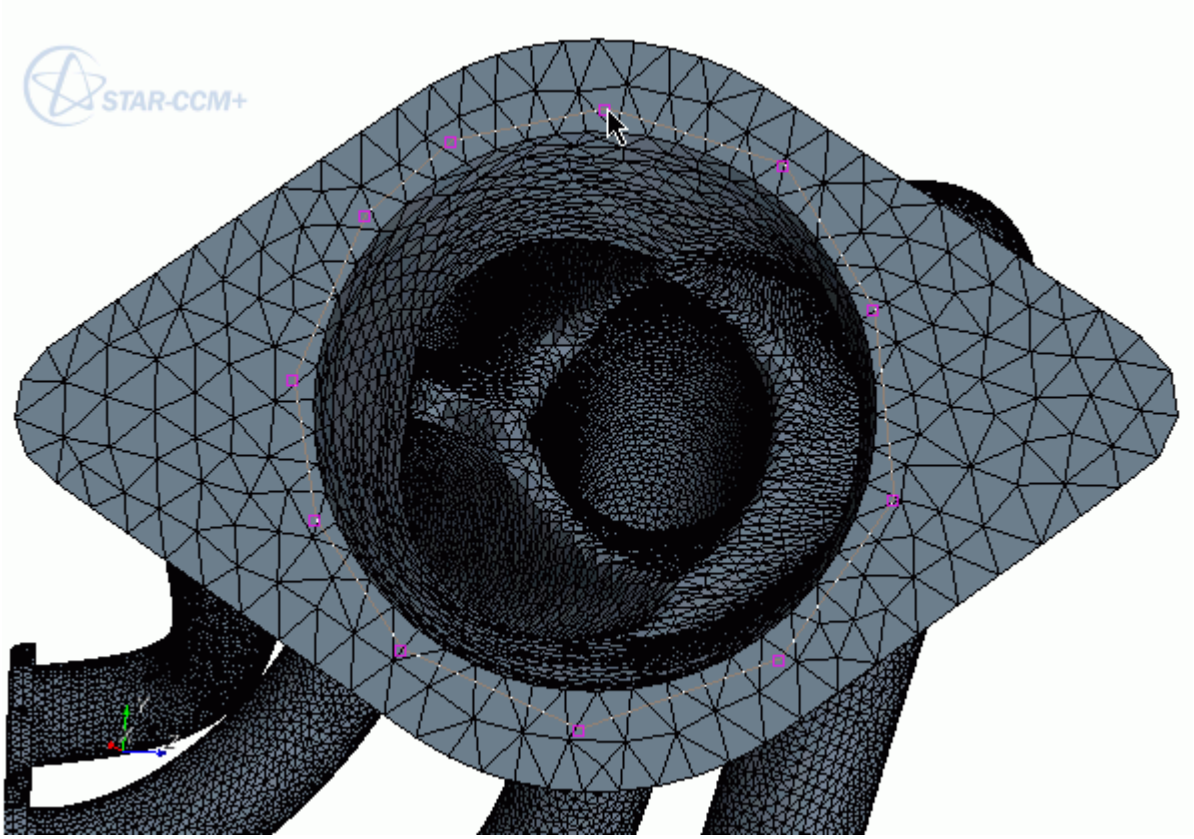
Fill polygonal patch

This tool patches an arbitrary planar hole by clicking the surface surrounding the hole.

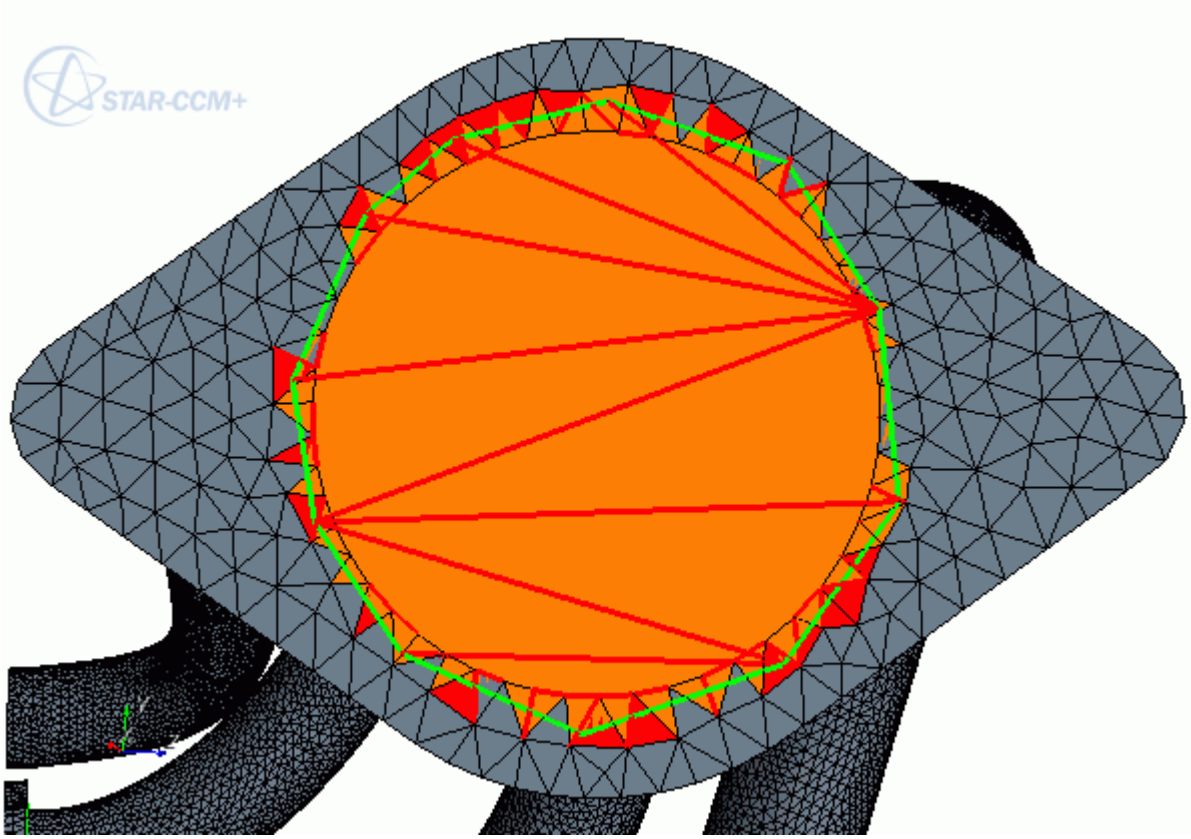
The tool is always available, as it does not require any selections.

Define the patch using the cursor in the *Graphics* window to click points around the patch perimeter. End the patch definition by right-clicking a point, or clicking the starting point.

The following example uses the **Polygonal Patch** tool to cover a hole before using the surface wrapper.



As the patch does not fill the hole exactly, the surface diagnostics report numerous errors. However, if you intend to use the surface wrapper, these errors are not an issue, as the surface wrapper only requires a closed surface.



Auto-repair surface errors

This tool automatically repairs errors reported by the following diagnostics:

- Face quality
- Face proximity
- Self-intersections

The tool uses two fixing mechanisms.

- **Remesh** - Grows the problem area and surface remeshes, removing feature curves if necessary.
- **Patch** - Grows the problem area, deletes the local surface, and then invokes the hole filler.

First, the tool applies the remesher to faces identified as having a given error. If the remeshing succeeds in eliminating the issue or improving the quality, the tool takes no further action. If remeshing does not fix the error, the tool uses the patch option to remove the original error. You typically use the auto repair tool to resolve any errors that exist after surface wrapping.

Clicking **Auto-repair surface errors** starts the fixing process according to the above criteria. The tool only applies each fixing mechanism if you have run the diagnostic check for that particular type of error. For example, if you have not checked for face quality, the auto repair tool does not fix any poor quality faces.

The auto repair tool does not fix topological errors such as free edges, non-manifold edges, and non-manifold vertices. If these errors exist on the surface, you can fix them manually before using the auto repair process for the remaining issues. The auto-repair function is primarily designed for fixing single region or per-region cases that were surface wrapped. As such, the tool does not recognize interfaces

between regions and can therefore potentially introduce errors on the faces if they are included in the active model.

Fill holes using selected edges

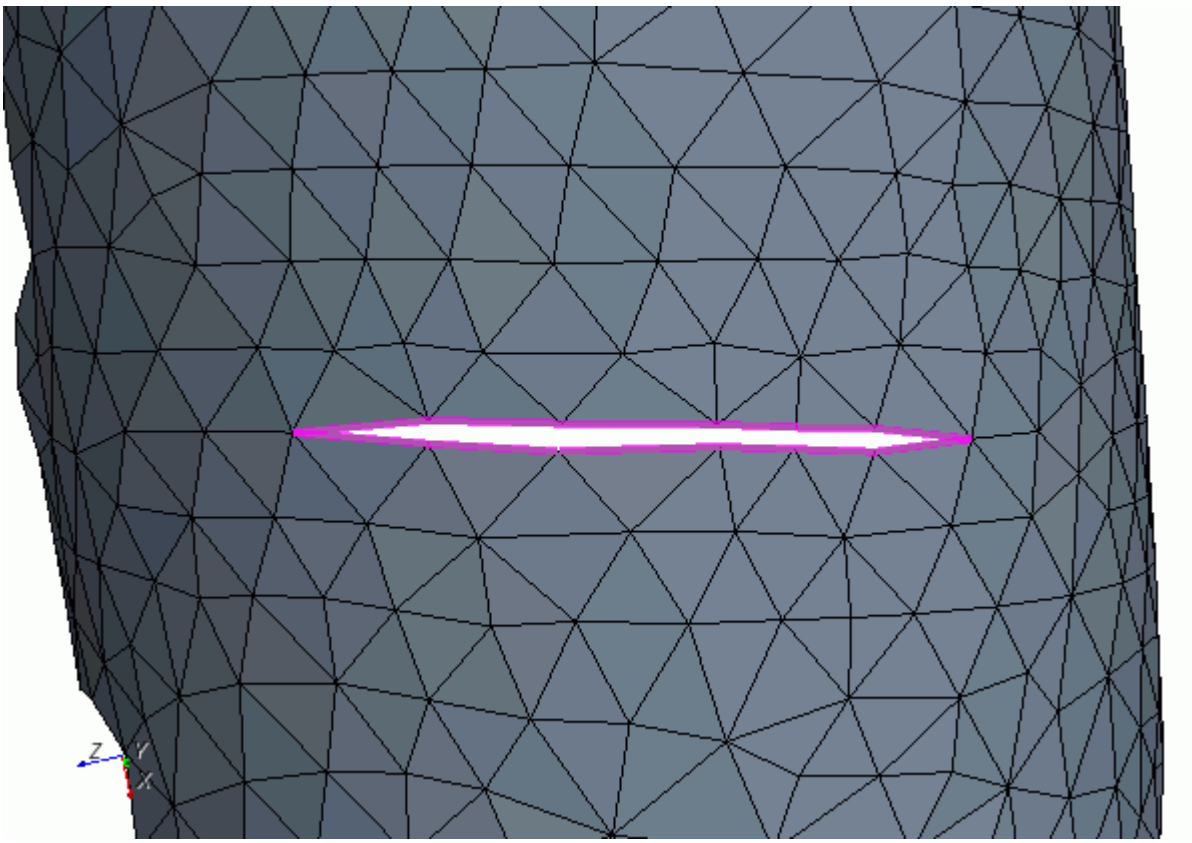
This tool fills a hole with new faces using the selected edge loops. See Filling Holes.

Zip selected edges

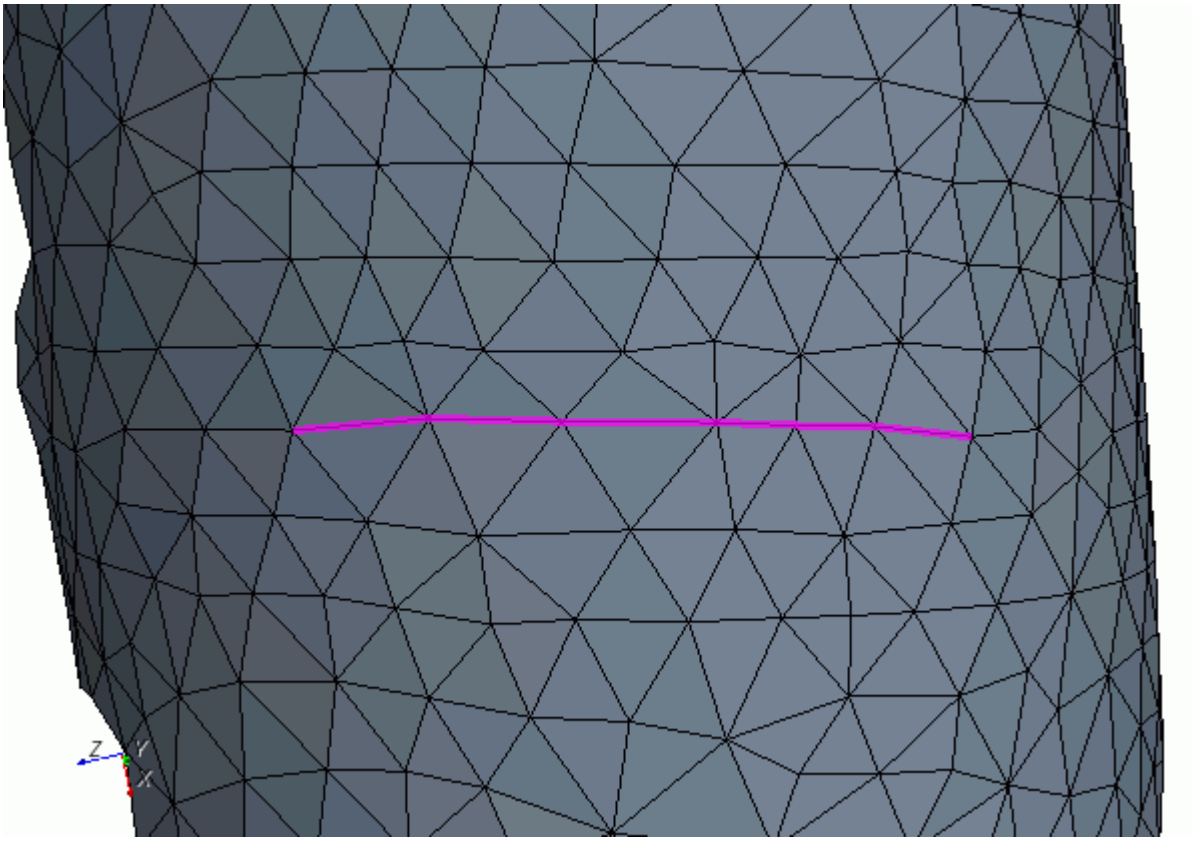
This tool locally zips faces together to remove free edges.

Select two or more edges before the tool becomes available. Right-clicking on the button provides access the same options as for zipping edges using feature curves.

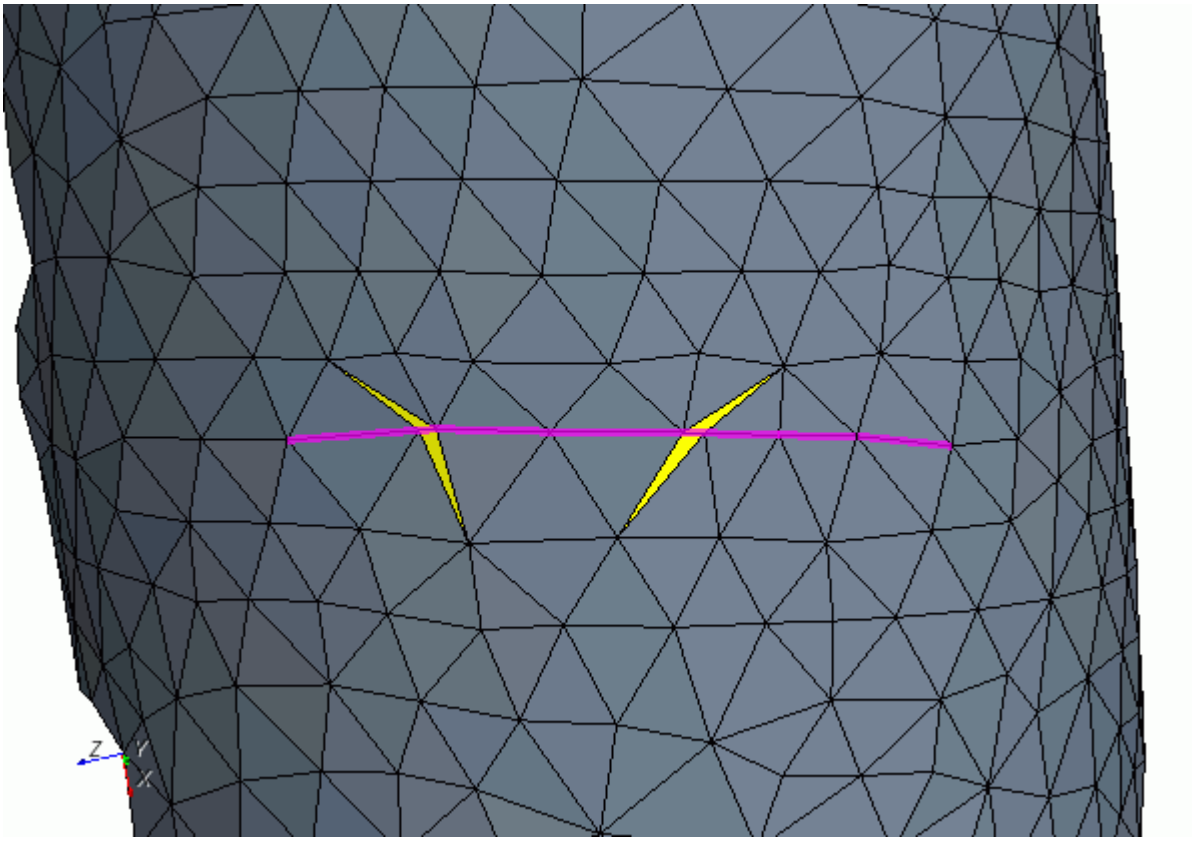
The following example uses the zip tool to close a small crack on the surface mesh.



The tool projects the vertices on one side of the crack to the edge on the other side, and merges close vertices.



In some instances, the tool can produce poor quality faces, if it fails to merge close vertices due to tolerances.



You can fix these issues in several ways:

1. Select the short edge, and use the collapse vertices tool. This collapses vertices and removes the poor quality face from the surface mesh.
2. Select the close vertices and use the merge adjacent vertices tool, with an adequate tolerance specified.
3. Select two or more free edges on the geometry, then:

   a. In the *Selection Control* group box, click (**Grow selected**) once or twice to grow the selection of neighboring faces.

   b. Click (**Zip selected edges**).

The selected faces are then locally surface remeshed. This eliminates the original faces and provides a high quality triangulation result. The advantage of this method over the other two is that it can be completely automated. There is no need to identify short edges or supply a merge tolerance.

Remesh selected faces

This tool locally remeshes the selected faces to improve the face quality.

The tool retains feature curves or edges that are flagged as feature curves on the surface. Select at least one face before this tool becomes available.

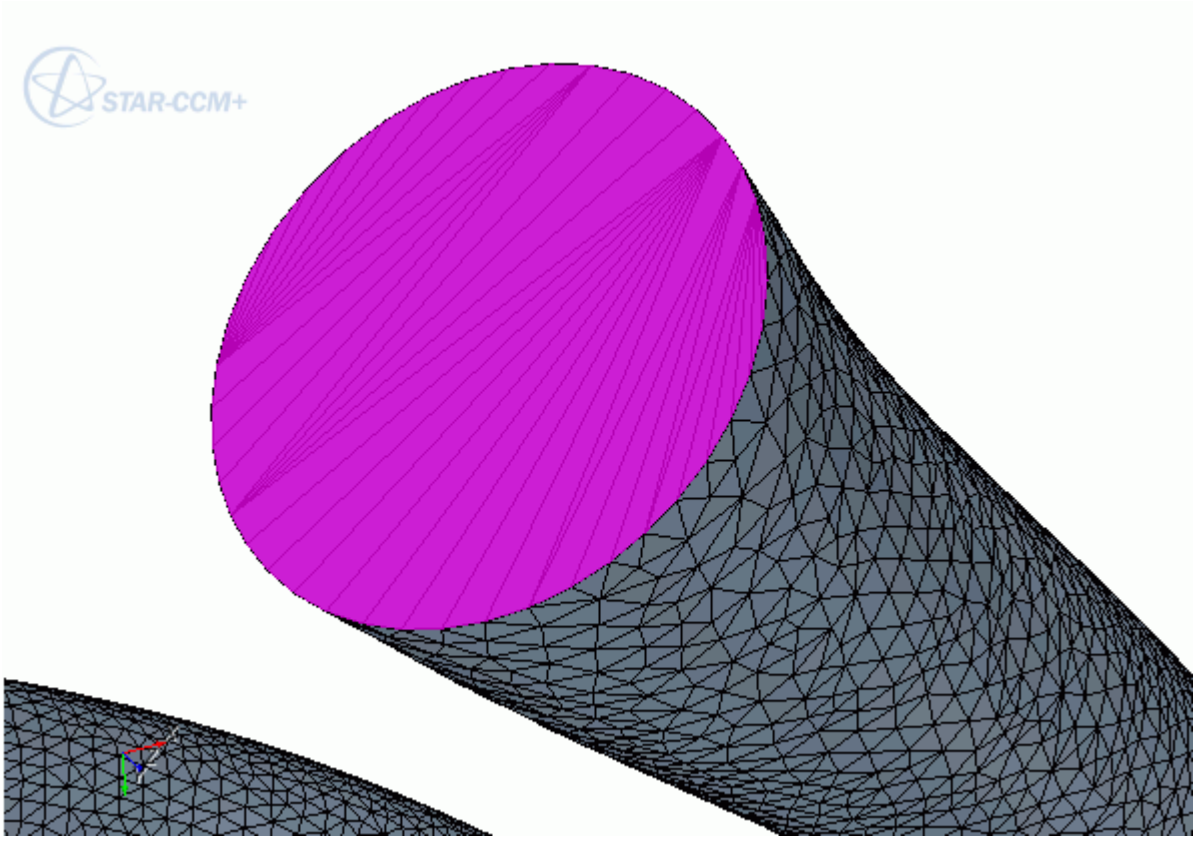
OPTIONS

Specify target size allows you to set the triangle size of the remeshed region. Deactivating the **Specify target size** option means that the surface remesher uses the local face edge lengths to determine the triangle size.

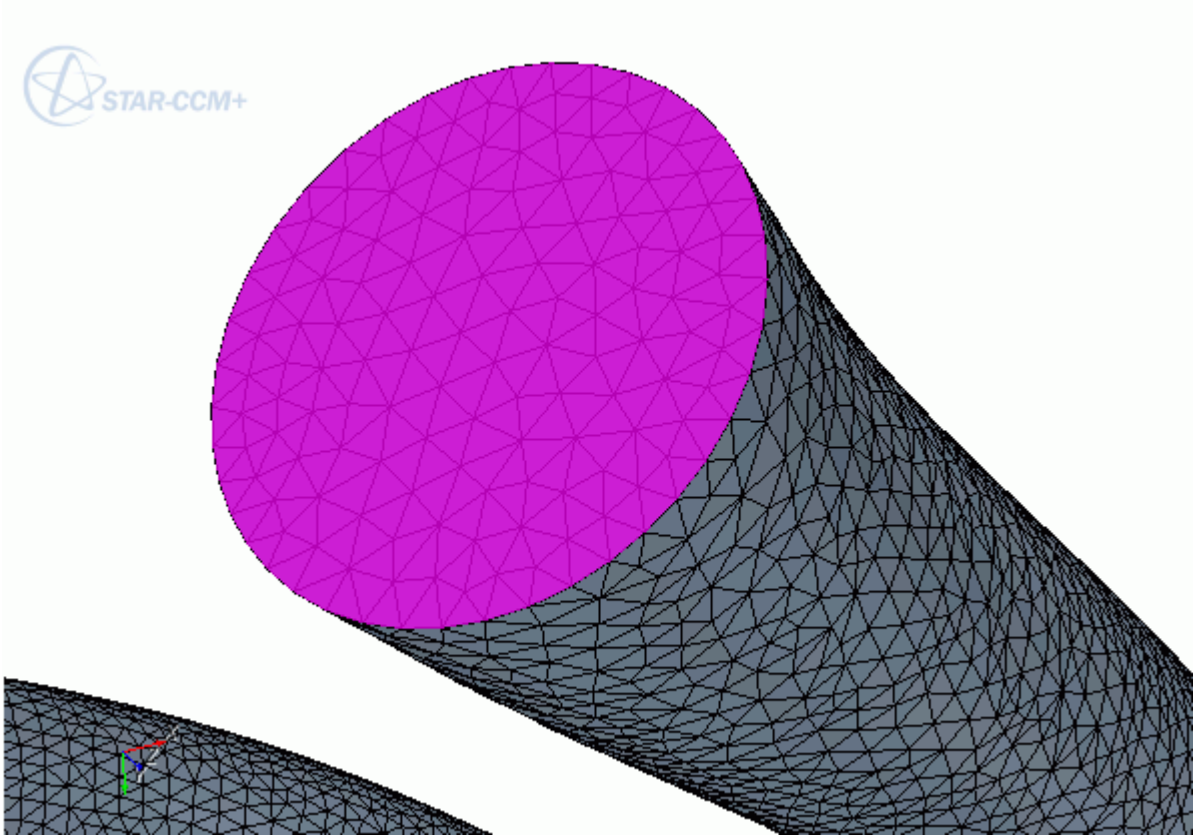
Do Curvature Refinement adds curvature-based refinement to the remeshing operation. You can specify the *#Pts/Circle* and *Minimum Size* values to improve surface definition on curved portions of the geometry.

See Surface Mesher Controls and Values.

The following example uses the local remesher to improve the quality of faces produced when filling holes.



Using the local remesher on the faces highlighted above has the following results.



Smooth selected/attached vertices

This tool locally smooths a selected vertex, or vertices to improve the quality of the surrounding faces.

Use this tool if you want to refine the quality of a surface mesh by improving the distribution of vertices on a surface or along a feature curve.

The smooth selected / attached vertices tool becomes available when you select at least one vertex, edge, or face.

There are two smoothing options:

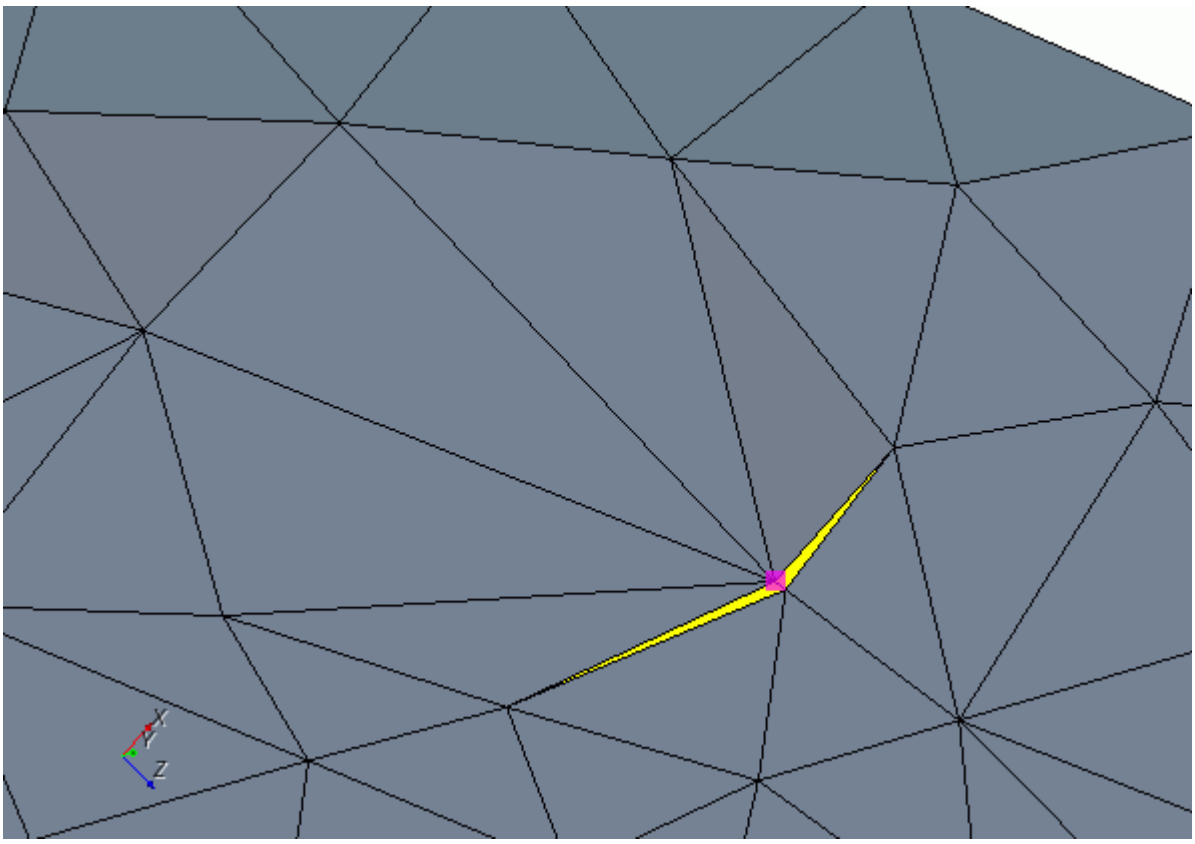
- **Unconstrained Smoothing** — Use this mode to smooth vertices arbitrarily on a surface.
- **Constrained Smoothing** — Use this mode if you want to smooth vertices in a specified coordinate direction.

Unconstrained Smoothing Options

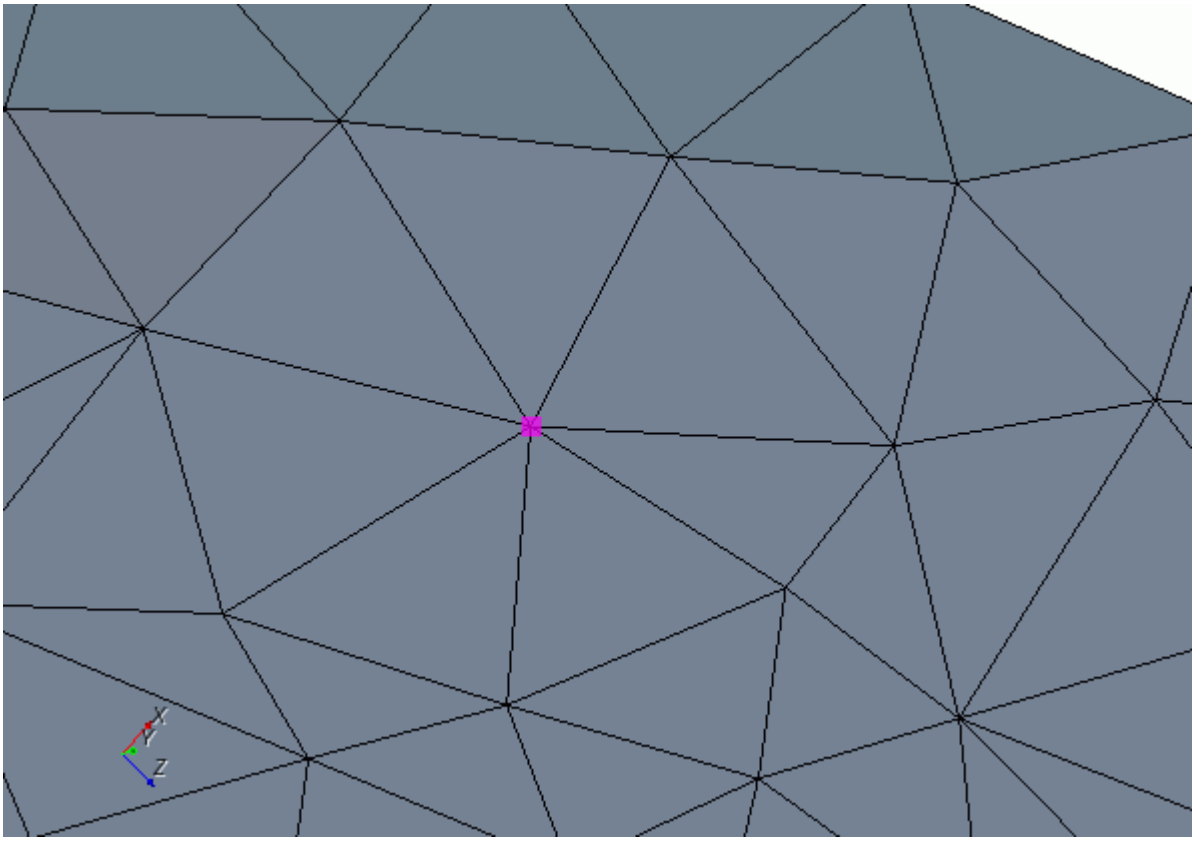
The following *Smoothing Options* are available for unconstrained smoothing:

Preserve Feature Edges	Ensures that the tool does not move vertices belonging to a feature curve. However, where long chains of feature curve vertices are selected, it is possible that these vertices are smoothed along the length of the feature curve.
Reproject to initial surface	Performs an additional step after smoothing, which attempts to project the selected vertices back to the original surface.

The following example smooths the selected vertex to increase the quality of the attached faces.



Using the **Smooth Vertices** tool with an appropriate number of iterations has the following result.



Constrained Smoothing Options

This action moves vertices along the edge to distribute the vertices more evenly. The vertices are smoothed to increase the quality of the faces surrounding the selected edge.

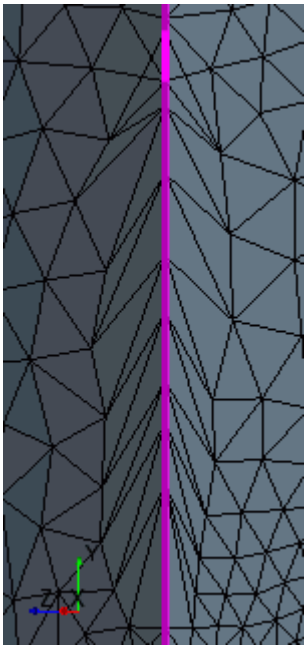
The following *Smoothing Direction* options are available for constrained smoothing:

Smooth in X direction	When active, smooths vertices in the X direction.
Smooth in Y direction	When active, smooths vertices in the Y direction.
Smooth in Z direction	When active, smooths vertices in the Z direction.
Coordinate System	Allows you to specify the coordinate system that you require. Set to **Laboratory** by default.

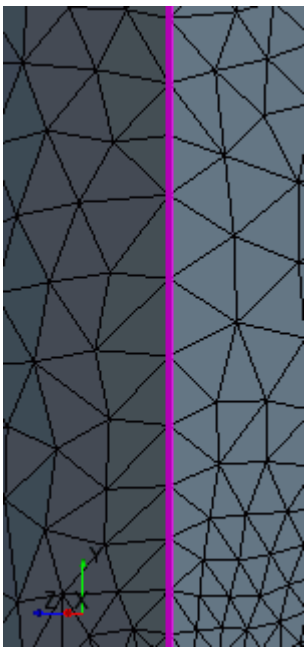
Note: Smoothing must be allowed in at least one coordinate direction. Therefore, it is not possible to deactivate all three smoothing directions.

The options to **Reproject to the initial surface** and **Preserve feature edges** are not available for constrained smoothing.

The following example shows an edge that is selected for constrained smoothing in the Y-direction.



The result after five iterations of smoothing shows that the vertices along the selected edge are smoothed in the Y-direction.



Merge adjacent vertices/duplicate faces

This tool merges adjacent vertices within a given tolerance and merges duplicate faces (that is, faces that share three vertices).

The tool is always available when you choose set *Operate On* to **Active Model**, or when you select a suitable face or vertex.

The merge options specify whether the tool merges adjacent vertices, duplicate faces, or both. The default options merge adjacent vertices and duplicate faces with a tolerance of 0.0001m.

If you activate the **Merge Adjacent Vertices** option, the tool merges vertices first. See Merging Vertices

If you activate the **Merge Duplicate Faces** option, the tool merges faces second. The tool deletes any collapsed faces or edges from the vertex merge operation in order to ensure a valid result. See Merging Duplicate Faces

The *Operate On* options determine whether the tool merges the currently selected faces and vertices, or all faces and vertices in the active model. The active model is the faces and vertices belonging to the parts originally selected in the Surface Repair Options dialog. The default option is to operate on the selected set only.

When merging vertices using the selected set only, the tool removes all merged vertices from the current selection. Therefore, if you repeat the merge with an increased tolerance, the tool only uses unmerged vertices from the original selection. However, when merging duplicate faces, the tool retains all selected faces, apart from those faces that the merging process deleted.

When merging using the active model, the tool deactivates any existing vertex or face selections. When vertex merging is complete, the tool selects the master vertex for each successful merge. Similarly, when merging faces, the tool selects the master face for each successful merge. You can use this information to judge the success of the merge, and where it took place on the part.

Repair Feature Tools

The Repair Feature tools repair errors on the feature edges on the geometry.

Flag edges as feature

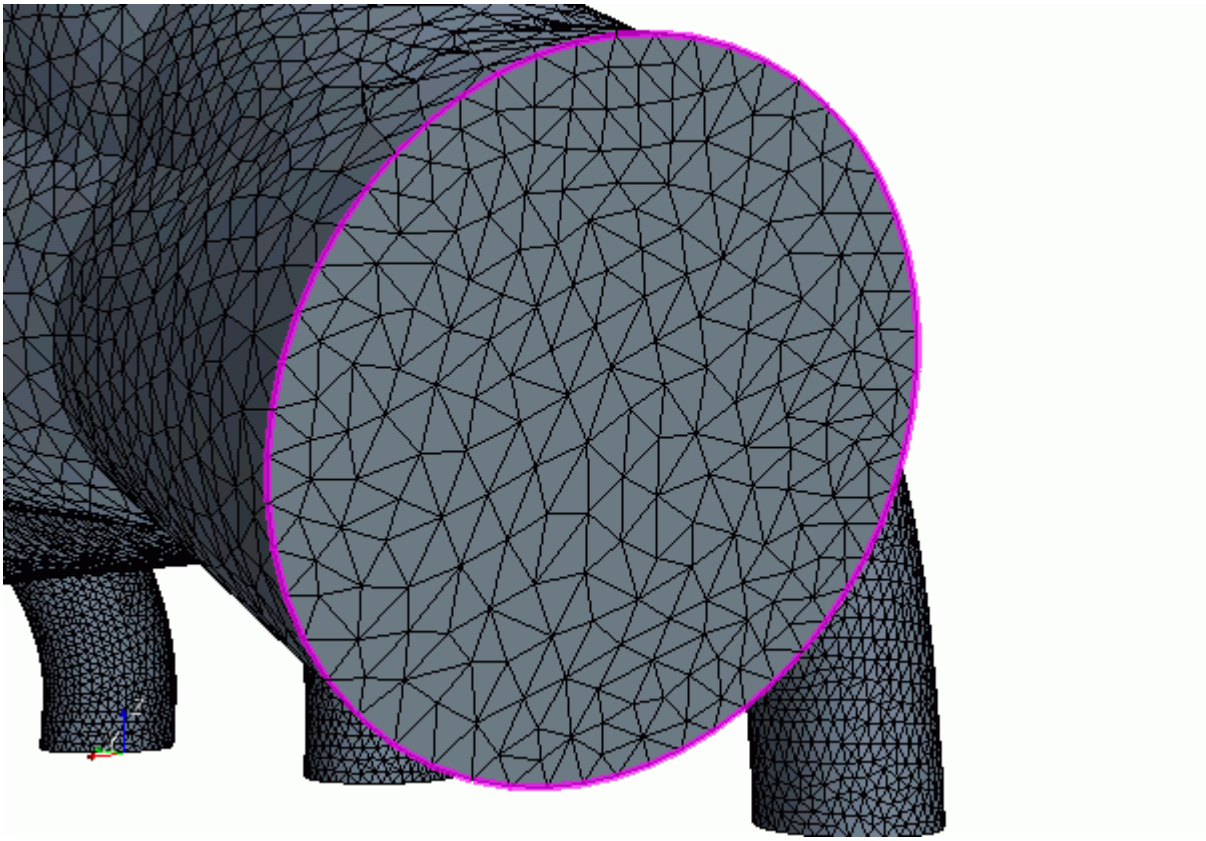
This tool creates part curves for the selected edges, in order to maintain geometry details when creating the surface mesh.

The tool becomes available when you select at least one edge.
You can assign edges to a different part curve using the *Part Surface / Part Curve Edit* tool.

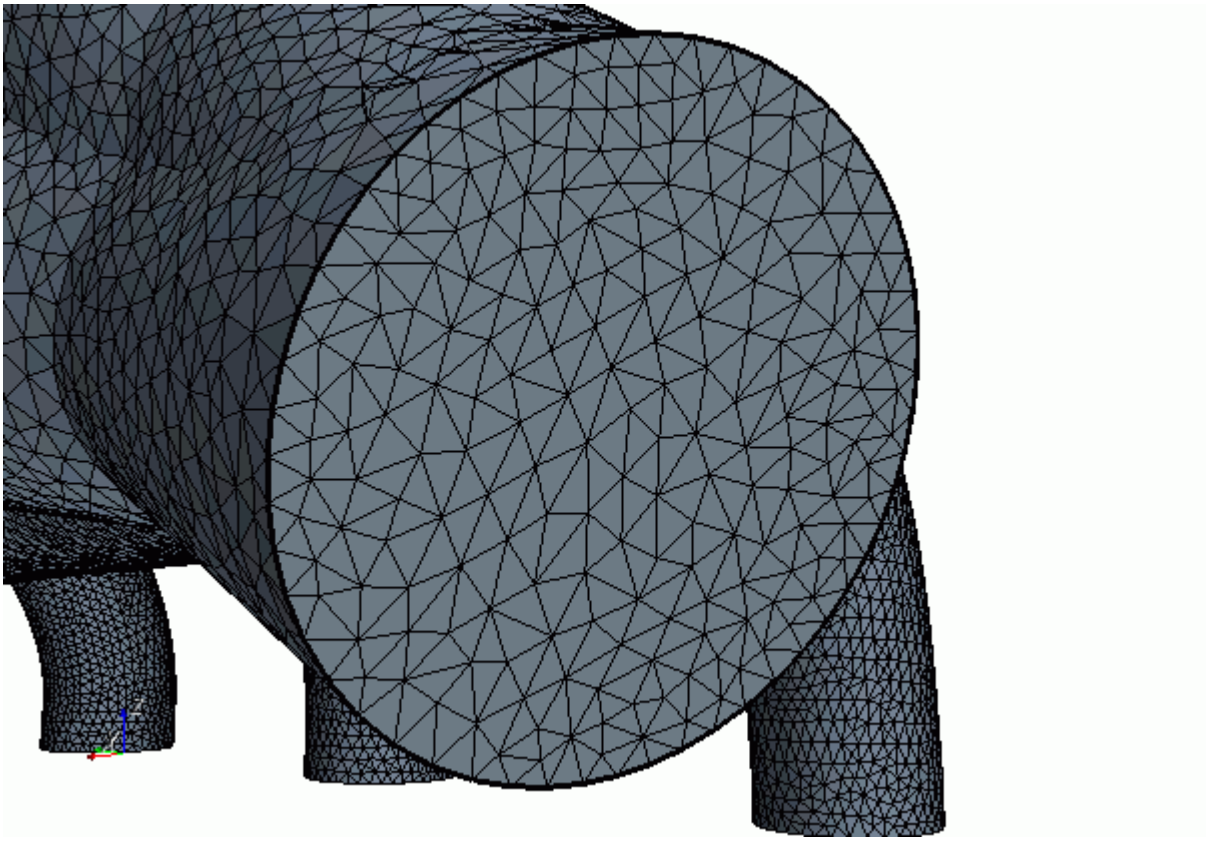
The tool adds marked feature curves to the **Surface Repair Edges** part curve, which is located in the simulation tree under the **Geometry** > **Part** > **geometry part** > **Curves** manager.

You typically use this tool when a required feature is missing, after having used the Mark Features tool to create feature curves automatically.

The following example flags the highlighted edge as a feature, in order to preserve the geometry detail during meshing.



Clicking **Flag edges as feature** creates a feature on the surface, which is marked with a thick black line.

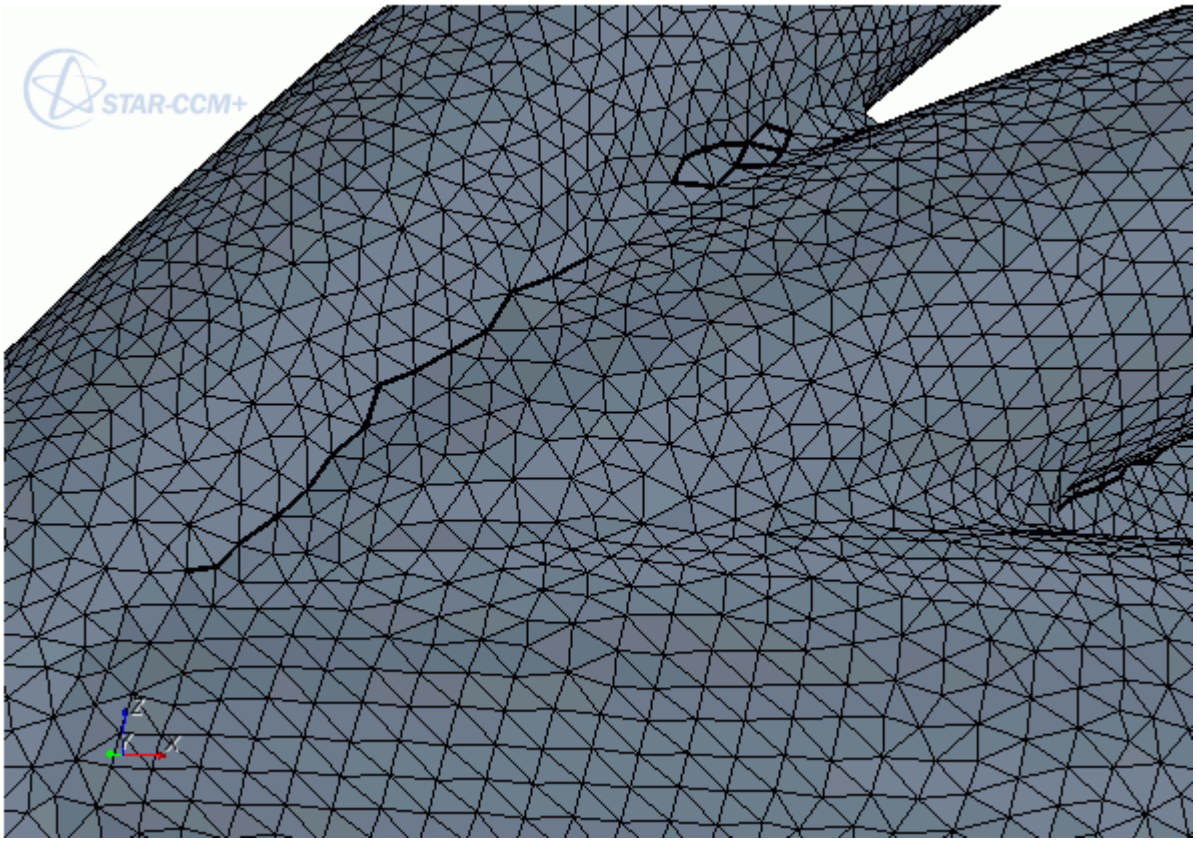


Unflag feature edges

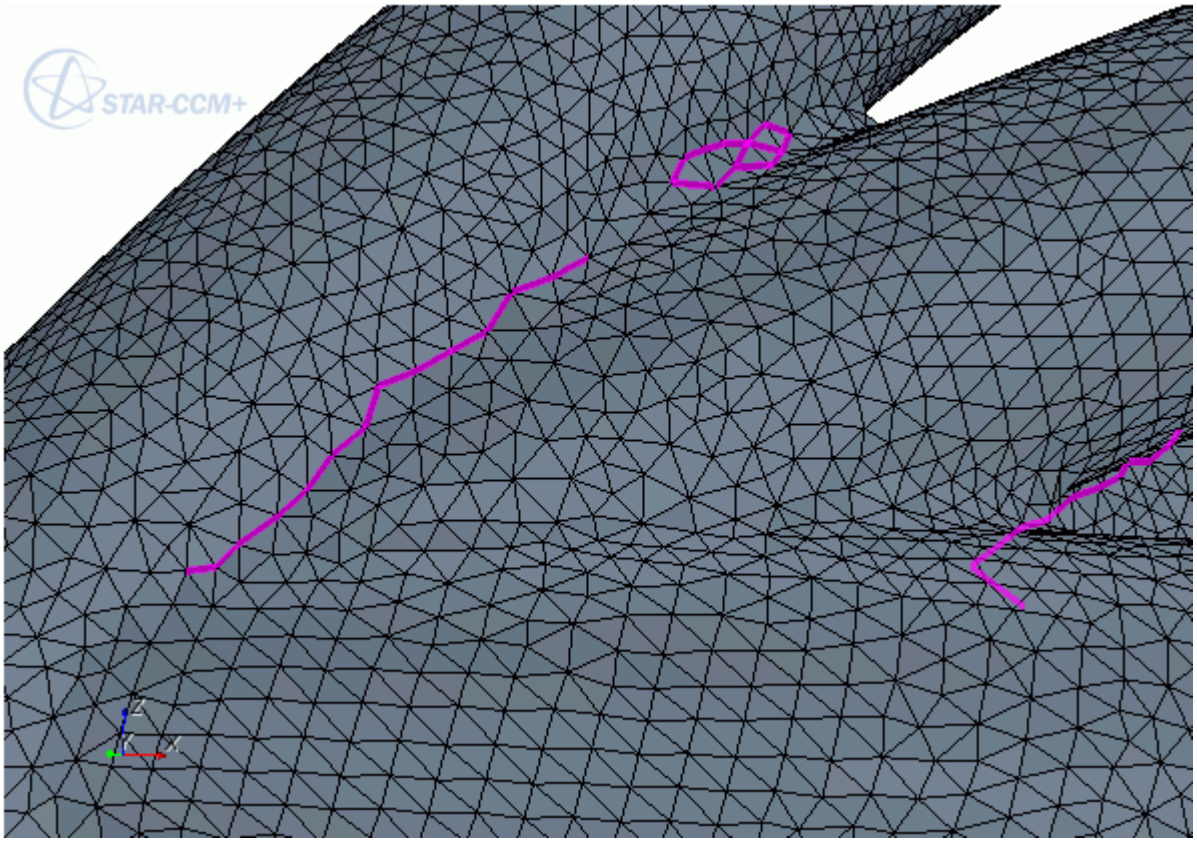
This tool deletes the selected feature curves, in order to ignore geometry detail when surface remeshing.

The tool becomes available when you select at least one edge.

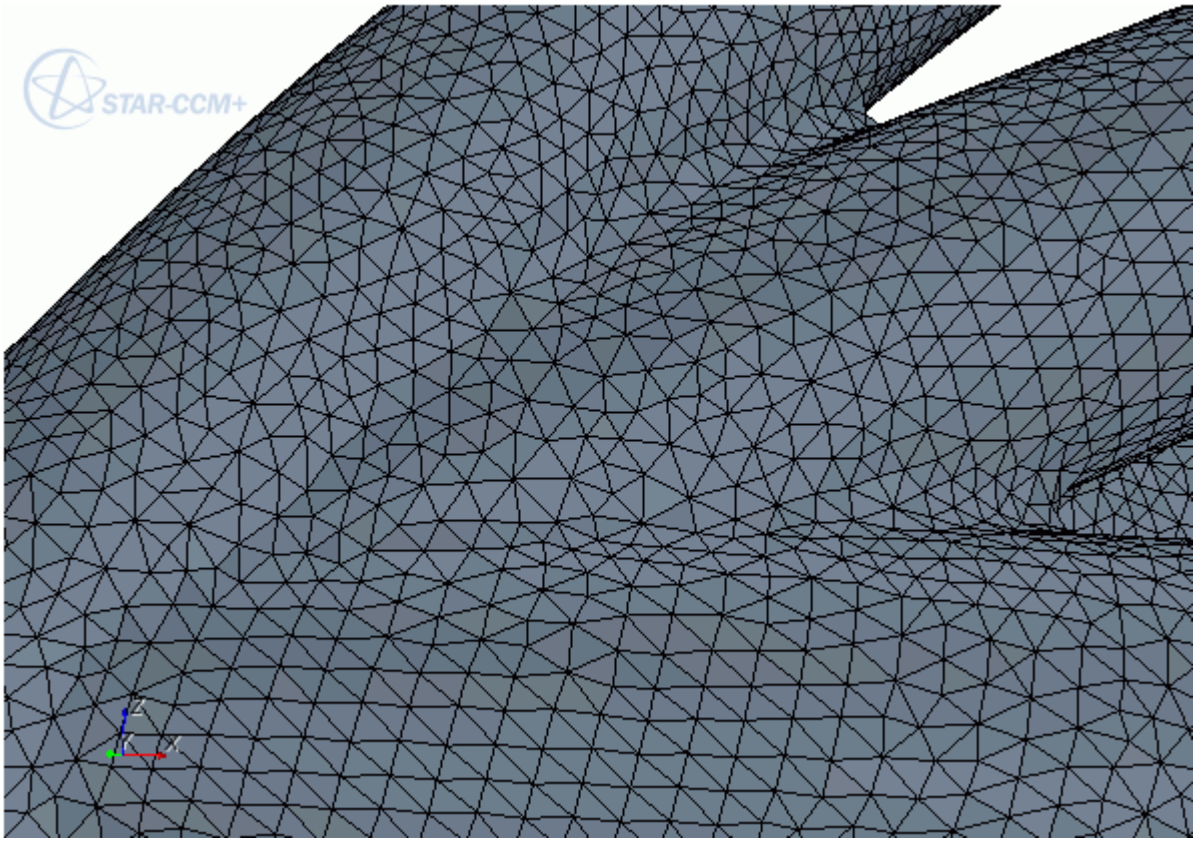
This example unflags all visible feature curves in the image below, as they do not represent important geometry details.



Select the unwanted feature curves using the mouse in the *Graphics* window.



Click the **Unflag feature edges** button to remove the selected feature curves from the surface.



Mark features

This tool automatically creates feature curves that are based on edge angle, free edges, non-manifold edges, patches, and boundaries.

The tool is always available, as no previous selections are required. The tool adds marked feature curves to the **Surface Repair Edges** part curve, which is located in the simulation tree under the **Geometry** > **Part** > **geometry part** > **Curves** manager.

In most cases, automatically marking feature curves creates some erroneous and unwanted feature curves, which you can remove by using the fix all and unflag tools. For example, in the image above, it would be advisable to remove the features that are created where the branches of the manifold join.

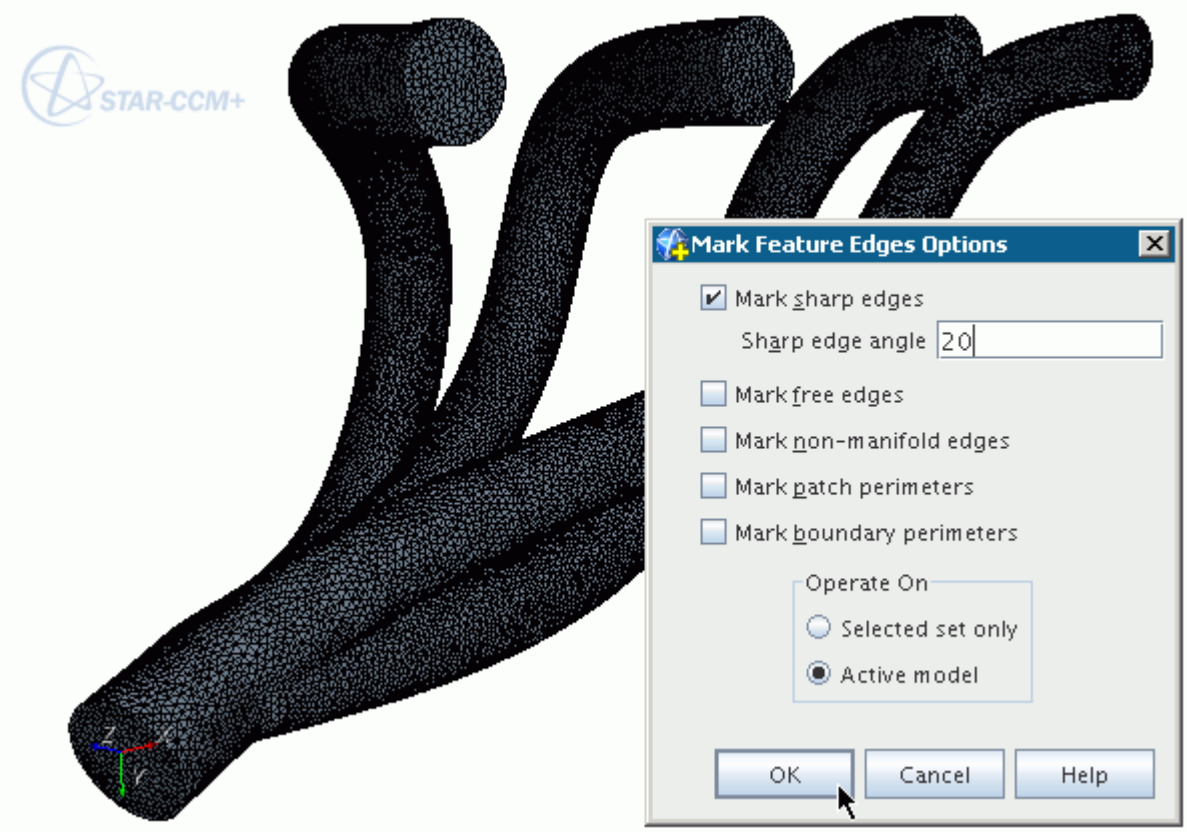
Assign edges to a different part curve using the *Feature Curve Edit* tool.

Click **Mark Features** to open the *Mark Feature Edges Options* dialog:

The default options produce feature curves that are identified with a sharp edge angle of 31 degrees, and those curves on boundary perimeters in the active model. The active model is the parts that are selected in the Surface Repair Options dialog when starting surface repair.

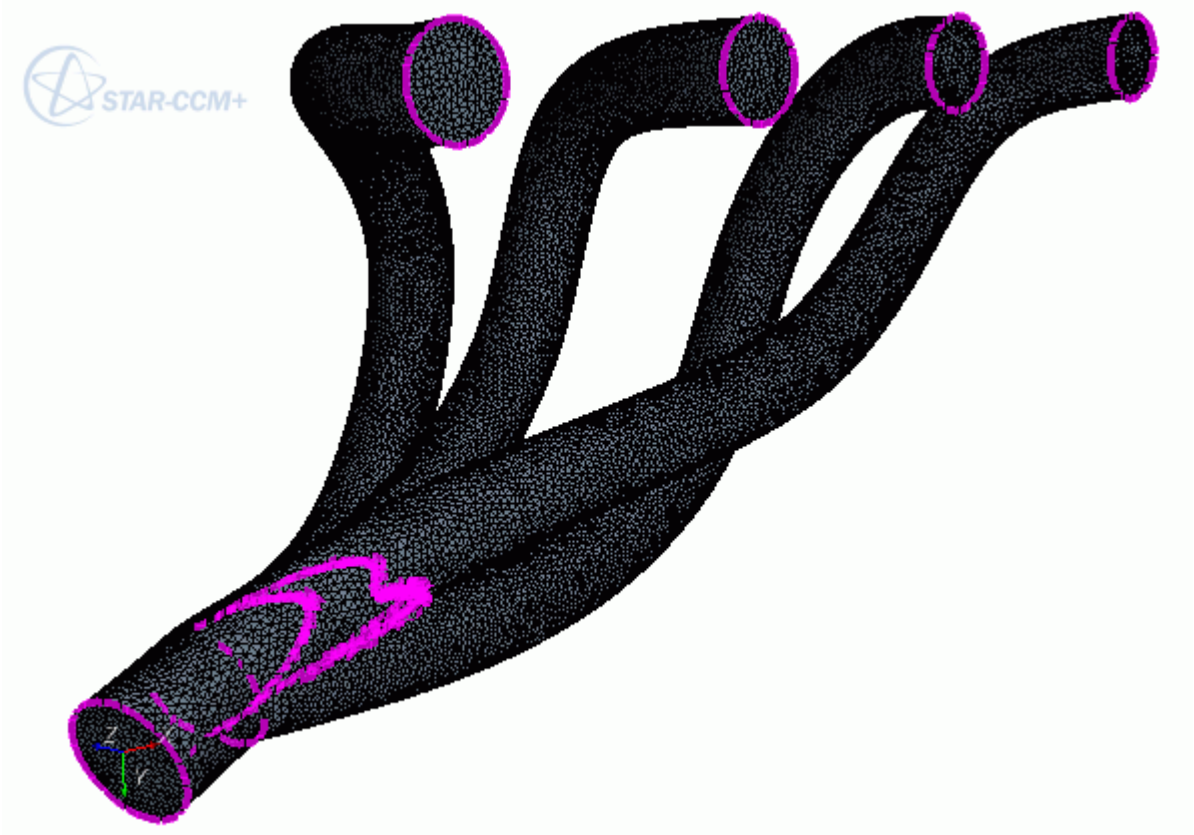
Modify the feature edge options as required.

The following example uses the Mark Features by Angle, etc tool with a sharp edge angle of 20 deg.



Click **OK** to create a part curve on all edges with an edge angle greater than the sharp edge angle.

When complete, the *Output* window reports the number of part curves marked on the surface. In addition, the tool highlights all pre-existing and new part curves in the *Graphics* window.



Toggle Chase Edge Selection Mode

This tool selects edges manually using the shortest distance between two vertices.

This tool becomes available when you select the **Edges** filter. See Using the Chase Edge Selection Mode.

Toggle Select Curve Mode

This tool lets you quickly select the edges that have been marked as feature curves by Mark Features by Angle, etc or related tools.

This tool becomes available when you select the **Edges** filter. See Using the Select Cuves Mode.

Fix all feature errors

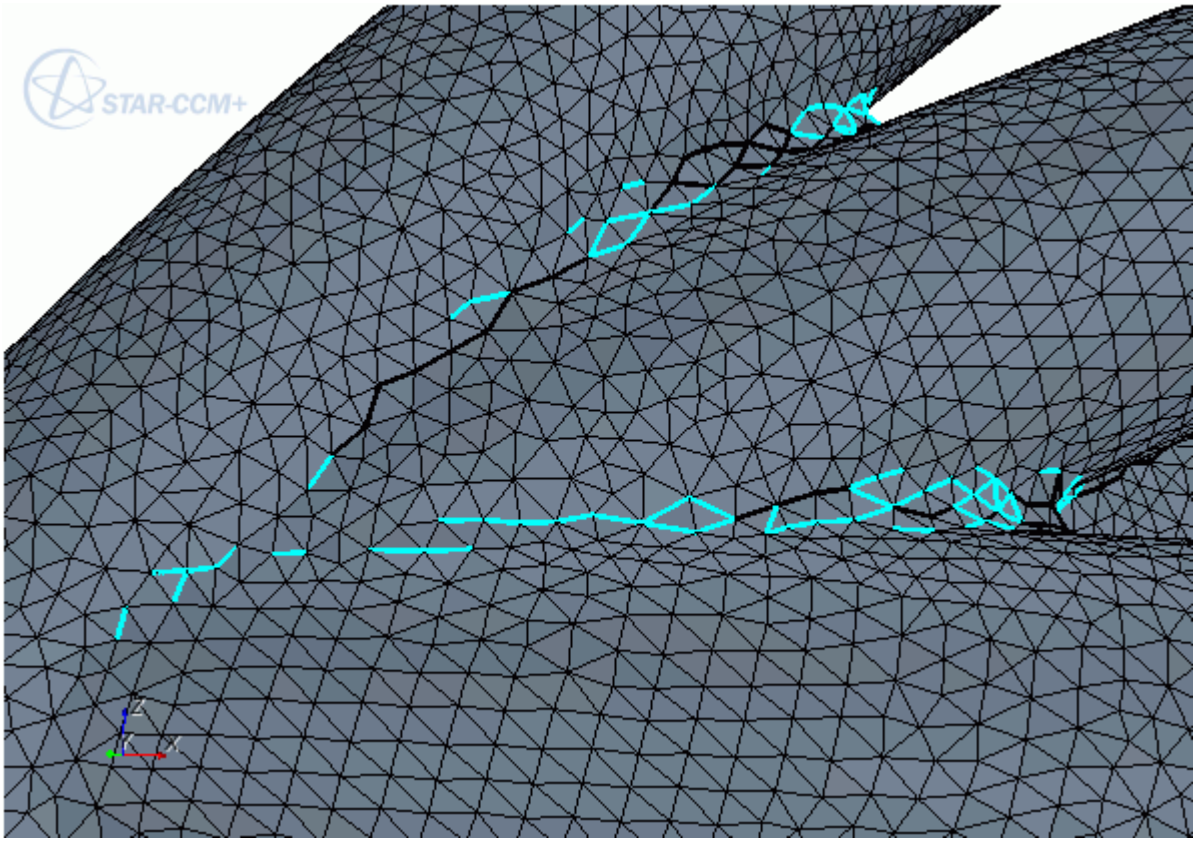
This tool fixes all feature errors in explicit feature edges that the feature diagnostics identifies.

This tool systematically browses through each feature curve error and unflags the related feature curves. To make this tool available, run the feature diagnostics.

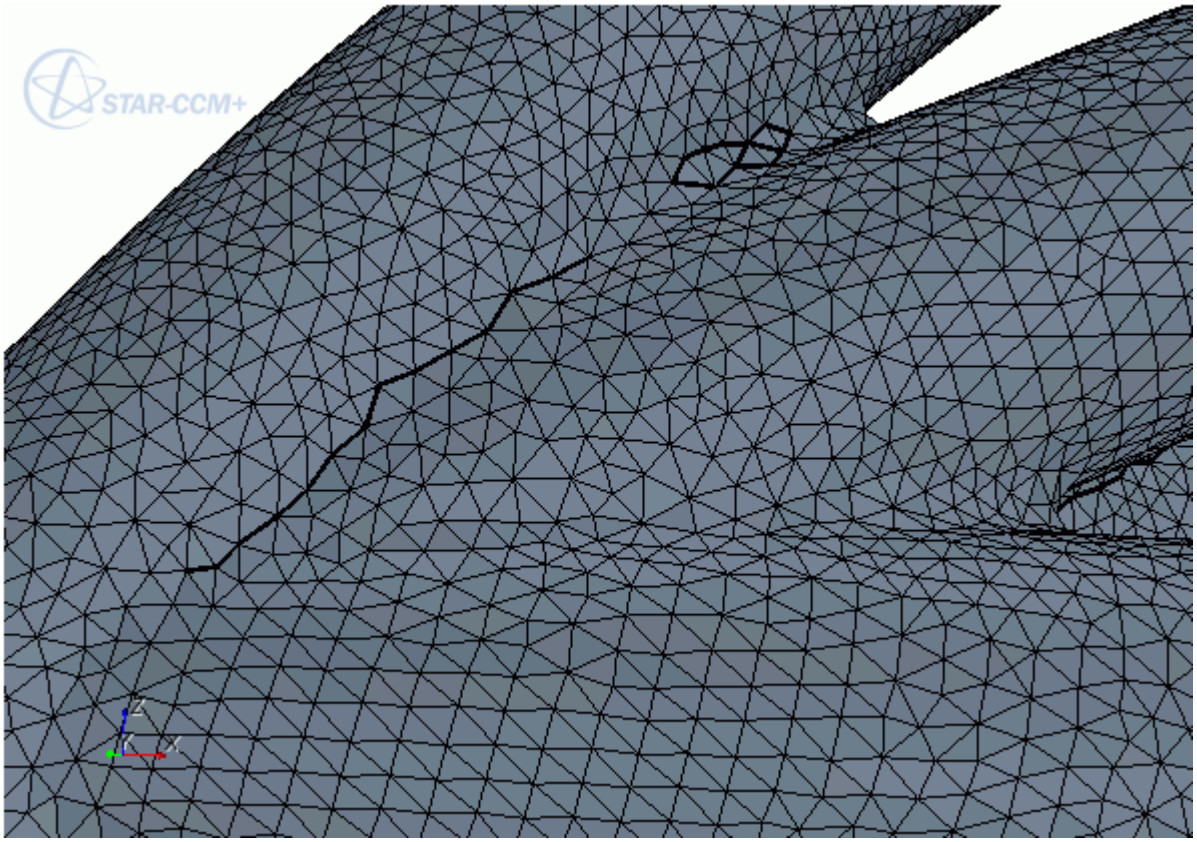
This tool only attempts to fix errors in explicit feature edges. Some errors in implicit feature edges are removed as a result of fixes to neighboring explicit feature edges, but the tool does not fix these directly.

If implicit feature errors still exist after the tool has fixed the explicit feature errors, the cause is due to the underlying surface. In this instance, you can use the surface repair tools to fix surface errors, before repairing feature errors.

In this example, the cyan lines indicate explicit feature errors, as is typical when using feature diagnostics.



Clicking **Fix all feature errors** removes these explicit feature errors from the surface.



After fixing all errors automatically, inspect the surface and unflag any feature curves that you do not want to preserve. For example, in the image above it would be advisable to unflag all the visible feature curves, even though they are not errors.

Not a feature error

This tool defines a feature error as not erroneous, and removes the error from the diagnostics summary.

The tool becomes available when you select at least one edge.

It is possible the feature diagnostics marks a part curve as an error, but you consider the part curve to be valid. In this instance, select the part curve and click **Not a Feature Error** to define the feature error as valid. The feature error summary reduces, and the feature diagnostics considers the curves valid for the current session.

Flag vertices as feature

This tool creates part points for the selected vertices. Once flagged, the vertices are represented as bold, black squares in the scene. A part point entry, **Surface Repair Points**, is added to the simulation object tree under the **Geometry** > **Parts** > **[PART]** > **Points** node.

This tool is only available when you enable part points in the simulation object tree.

Unflag feature vertices

This tool removes the selected part points.

This tool is only available when you enable part points in the simulation object tree.

Mark feature vertices

This tool marks feature points automatically based on either the active model or the current selection set, using options that you set.

To display the options for this tool, click to the right of this tool.

This tool is only available when you enable part points in the simulation object tree.

The following options are available for the mark feature vertices tool:

Sharp Vertices Along Part Curves	When active, marks vertices if the external angle between two connected feature curves is greater than the value specified in **Angle Between Edges**.
Curve Free Ends	When active, marks the open/free ends of any feature curves.
Curve Junctions	When active, marks vertices where three or more feature curves meet at a junction.
Curve Type Perimieters	When active, marks vertices where two or more curves of different curve types meet.
Operate On	Specifies what is used as the input. The **Selected set only** option uses any features curves attached to the current face and/or edge selection as the input. Note that at least one face or feature curve (not just an edge) has to be selected in this case before the **Mark feature vertices** tool is enabled. The **Active model** option uses all feature curves in the current model as the input.

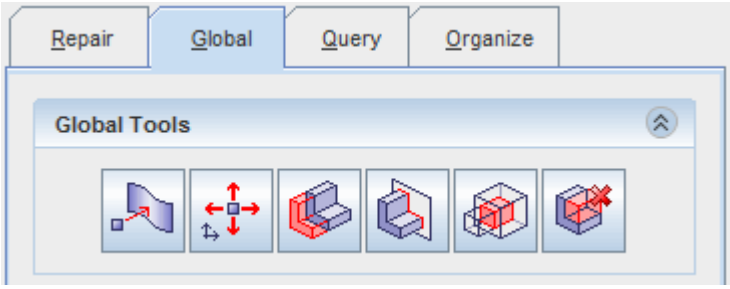
On executing the operation, all relevant feature vertices are marked based on the supplied criteria. Any previously marked feature vertices that are re-marked by the process will have their point type changed to the new type (called **Surface Repair Points**). The newly marked feature vertices are selected so that you can change their type if necessary.

Global Tools

Global tools are designed for work on larger portions of the model as opposed to individual faces, edges, and vertices.

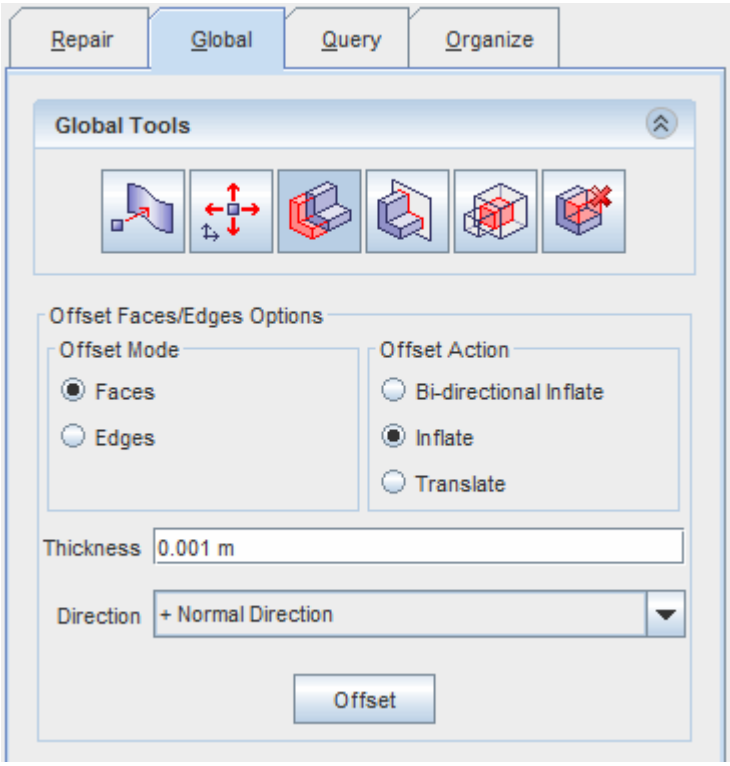
To access the Global tools:

1. In the *Surface Repair* panel, click the **Global** tab.



Note: Coordinate systems that you create in the STAR-CCM+ object tree while the repair session is active are automatically included in tools that use a coordinate system selection (for example when projecting vertices).

2. In the *Graphics* window, select the appropriate entities.
3. Click the desired tool.



The options for the selected global tool are displayed in the bottom half of the panel.

Selecting another global tool while one is already active switches to the new tool.

Creating a Local Coordinate System in Surface Repair

You can create local coordinate systems for surface repair tools that use a coordinate system without leaving the surface repair tool. The (**Create a new local coordinate system**) tool is located next to the coordinate system drop-down menu where appropriate.

To create a local coordinate system within the surface repair environment:

1. Click (**Create a new local coordinate system**).
2. In the *Coordinate System Options* dialog, choose the type of coordinate system to create. See Coordinate Systems: Local Coordinate System.
3. Create the coordinate system using the visual tools in the scene and the in-place dialog. See Creating a Local Coordinate System Interactively.
4. To add the coordinate system triad to the scene, activate the **Add coordinate system to scene on creation** option.

Coordinate system triads that are added to the scene are removed from the scene when you exit the surface repair tool.

5. Click **Create** followed by **Return**.

   You can create multiple coordinate systems in one session.

Note: As in the coordinate system tool in the simulation tree, only Cartesian coordinate systems can be used as reference systems for other local coordinate systems.

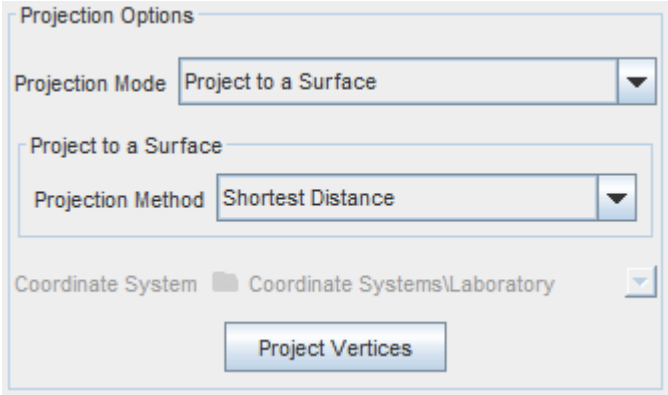
Contents:

Projecting Vertices
Transforming Surface Entities
Offset Faces/Edges Tool
Merge and Imprint Tool
Intersect/Boolean Faces Tool
Edit Connected Surfaces Tool

Projecting Vertices

This tool moves the selected vertex or vertices to a new location using one of five modes.

Click (**Project Selected Vertices**) to display the *Projection Options*.



Select the desired projection mode and enter the appropriate specification options and values, before clicking **Project Vertices** to execute the projection.

You can only specify flat planes using this tool, regardless of the coordinate system type.

Contents:

Projecting Vertices to a Plane
Projecting Vertices To A Line
Projecting Vertices In A Coordinate System
Projecting Vertices To A Surface

Projecting Vertices To An Edge
Picking Points To Define A Projection Plane Or Line

See Also:

Changing the Parts in the Active Model

Projecting Vertices to a Plane

The **Project to a Plane** mode projects selected vertices by the shortest distance to an imaginary plane.

Origin and normal components define the plane. The pick operation is available to select origin and normal points interactively in the Graphics window. You can specify origin and normal coordinates in the following ways:

- Specify the plane origin and normal using a Vector.

  Either use one or both of the following methods:

  - Define the plane origin and normal using the plane tool.

    STAR-CCM+ extracts the plane origin and normal from the orientation of a plane that you define using the Plane Tool. The vertices are then projected to the plane that passes through the specified origin.
  - Define the plane origin and normal using Fx Section.

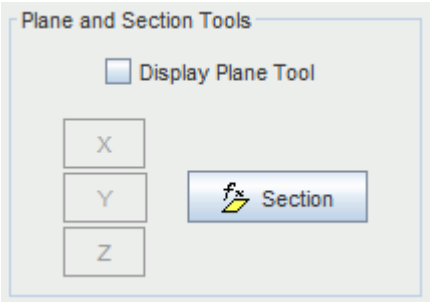
    The **Fx Section** button allows you to specify a plane section interactively in the *Graphics* window to represent the plane origin and normal.
  - Enter the plane origin and normal vector coordinates directly.
- Specify the plane origin and normal using Picked Points.

  The **Picked Points** operation allows you to select three points in a specified coordinate system to represent the plane normal. The three points all lie on the plane to which you project the vertices.

  When defining a plane with three points, all three points must be different and cannot be aligned. The plane cannot have zero area.

To Define the Plane Origin and Normal Using Fx Section:

1. Make sure that *Normal Specification* is set to **Vector**.
2. In the *Plane and Section Tools* group box, click **Fx Section.**

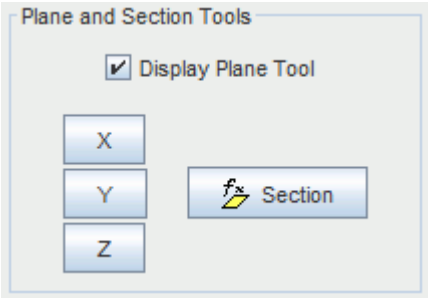


3. In the *Graphics* window, click once to begin your selection of the plane section.

4. Move the cursor until an appropriate plane section is defined, then click to confirm the selection.
5. If necessary, use the plane tool to refine the selection.

To Define the Plane Origin and Normal Using the Plane Tool:

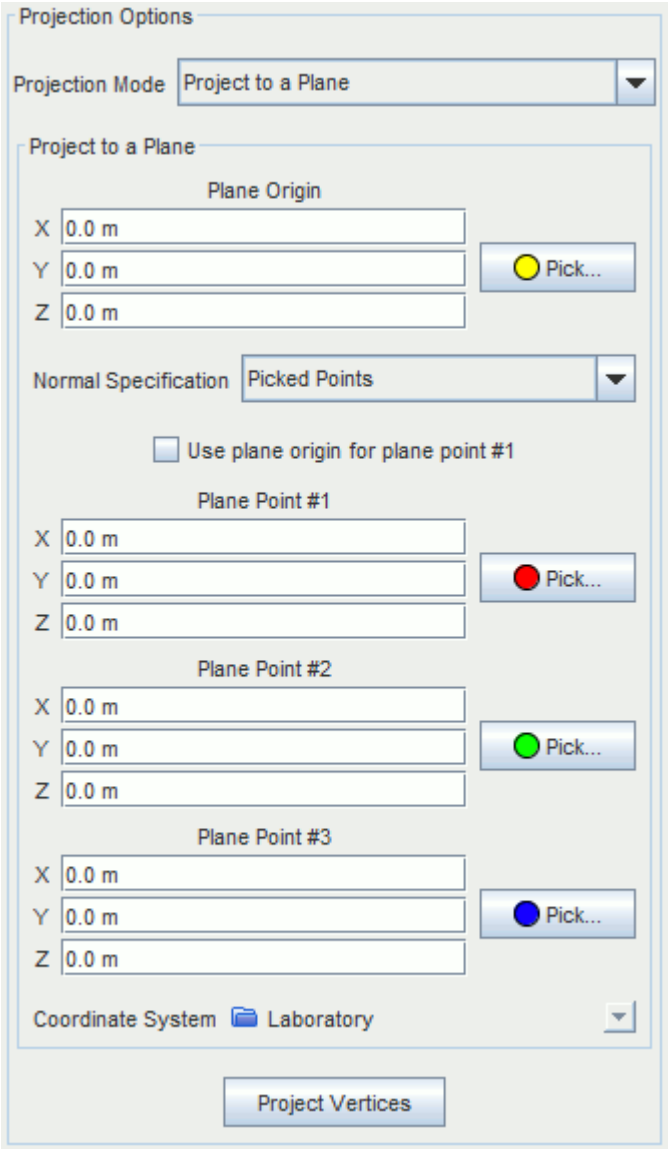
6. In the *Plane and Section Tools* group box, activate *Display Plane Tool*.



7. In the *Graphics* window, click and drag the Plane Tool to represent the plane origin and normal. You can also use the X, Y, and Z buttons to toggle the Plane Tool to lie on the X, Y, or Z axis.

To Define the Plane Origin and Normal Using Picked Points:

8. Define the Plane Origin.
9. Set *Normal Specification* to **Picked Points.**

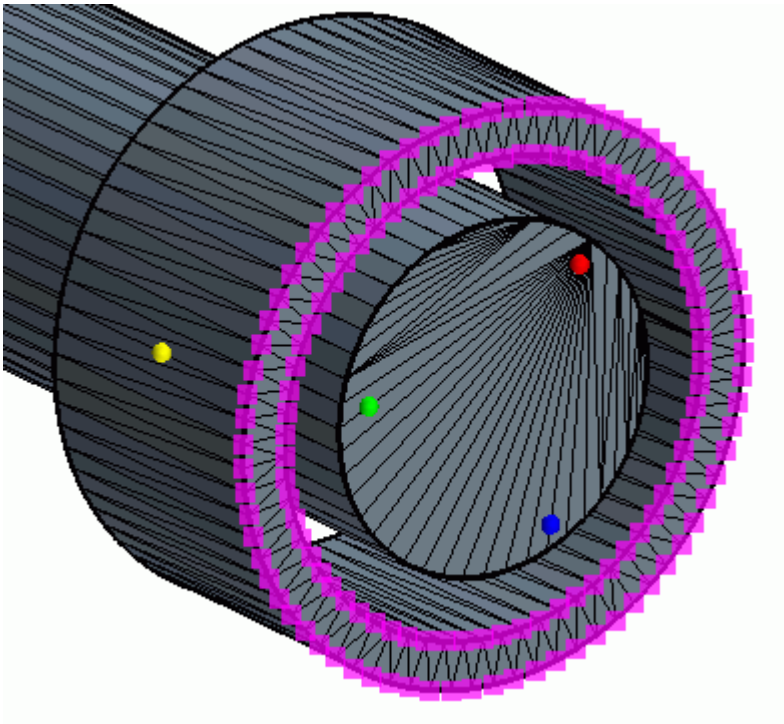


10. To specify the plane points, either:
    a) Use the **Pick** buttons to define each plane point interactively in the *Graphics* window, or
    b) Enter each set of **Plane Point** coordinates directly.

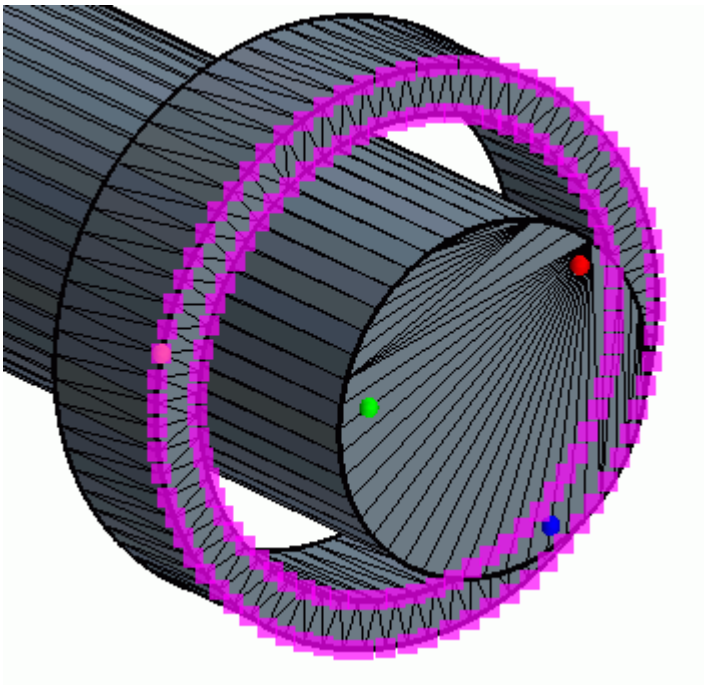
Note: To set the plane origin coordinates as one of the plane normal coordinates, activate *Use plane origin for plane point #1.*

The diagrams below demonstrate an example of this method.

After selecting the vertices to be projected and entering the project vertices mode, the origin (yellow) and normal (red, green, and blue) points are selected.



After applying the operation to the geometry, the plane on which the vertices lie is coincident with the origin. The projected plane also maintains the normal specification of the selected normal points:



Note: The **Project Vertices** button is only available if one or more vertices are selected for projection.

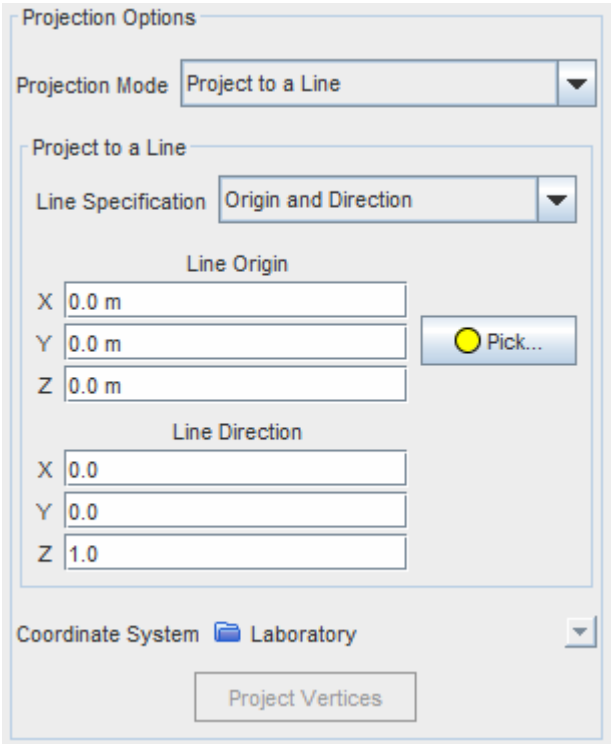
See Also:

Picking Points To Define A Projection Plane Or Line

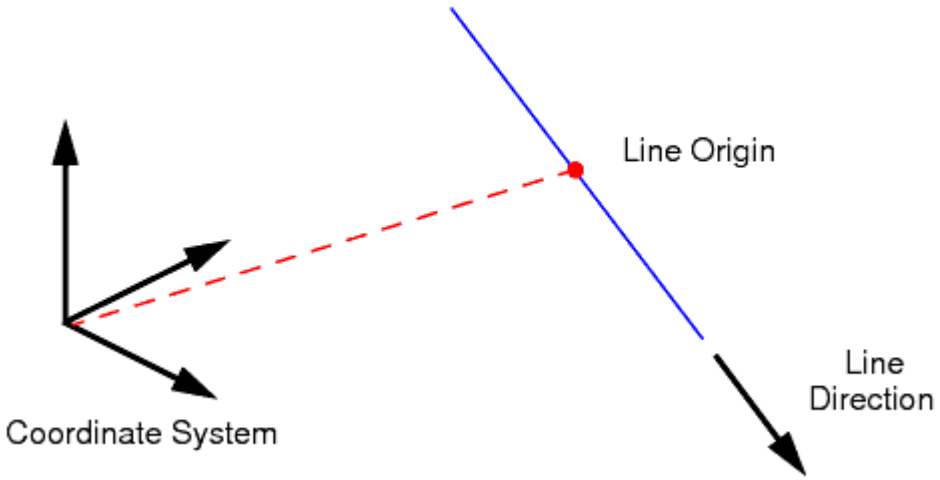
Projecting Vertices To A Line

The **Project to a Line** mode projects vertices by the shortest distance to an imaginary line.

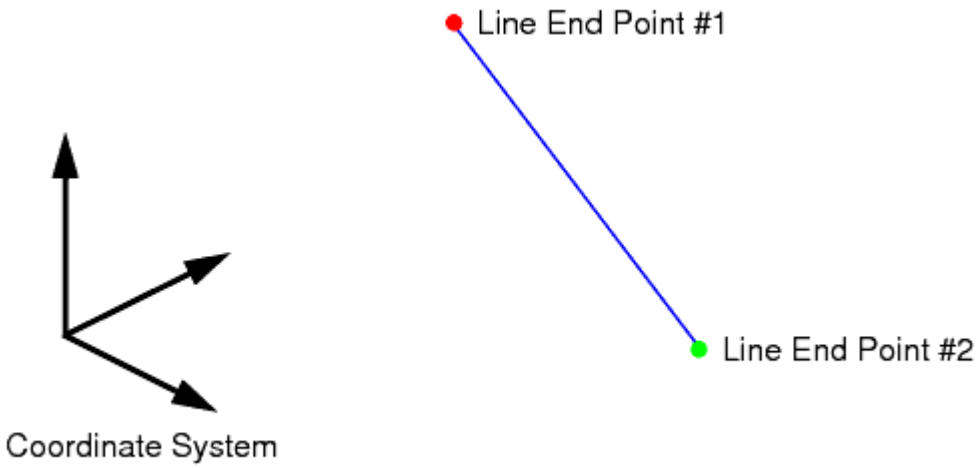
You define the line with an origin and a direction vector, or two points in a specified coordinate system. The pick operation is available to select points interactively in the Graphics window, providing a simple intuitive method of defining a line. The **Project Vertices** button is only available if one or more vertices have been selected for projection.



This diagram illustrates the **Line Origin**, **Line Direction** and **Coordinate System** specification:



This diagram illustrates the **Line End Point #** specification:



When defining a line with two points, both points must be different (that is, you cannot project to a zero length line).

You can only specify straight lines using this mode, regardless of the coordinate system type.

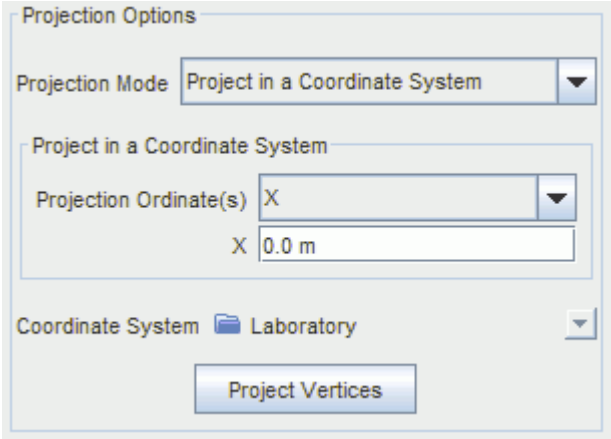
See Also:

Picking Points To Define A Projection Plane Or Line

Projecting Vertices In A Coordinate System

The **Project in a Coordinate System** mode projects vertices to a single or double ordinate location in a given coordinate system.

Any unspecified ordinates maintain their original value. For example, you can project vertices to a specified X value in a Cartesian system or to a specified R-q value in a cylindrical system. The **Project Vertices** button is only available if one or more vertices have been selected for projection.



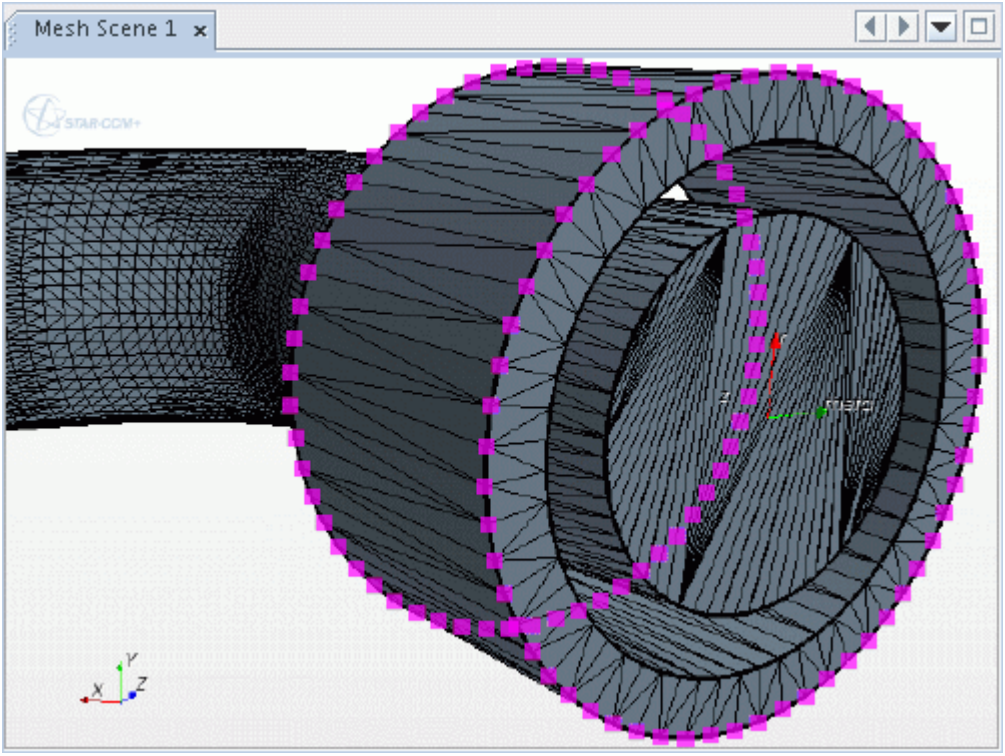
This mode differs from the project to plane mode as projections to cylindrical and spherical planes are possible.

1. First select the **Coordinate System**.

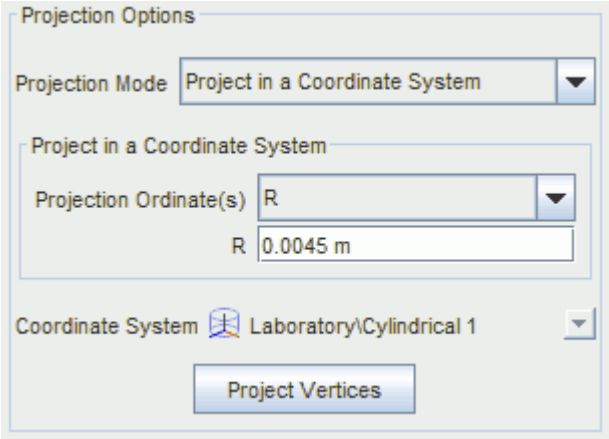
   Selecting the coordinate system (**Cartesian**, **Cylindrical**, or **Spherical**) updates the **Projection Ordinate(s)** menu with the appropriate coordinate directions.
2. You can then select the **Projection Ordinate(s)**, and enter the ordinate values in the desired units. The dialog accepts negative values for the radial direction in a cylindrical system, which projects vertices in the opposite direction.
3. The projection is performed when you click **Project Vertices**.

The following example projects vertices in the radial direction of a cylindrical system. The projection reduces the sleeve outer radius value to produce a thinner sleeve. This example uses a cylindrical coordinate system that is defined at the axis of the sleeve, where the radial axis points outwards.

4. Select the vertices on the outer surface:

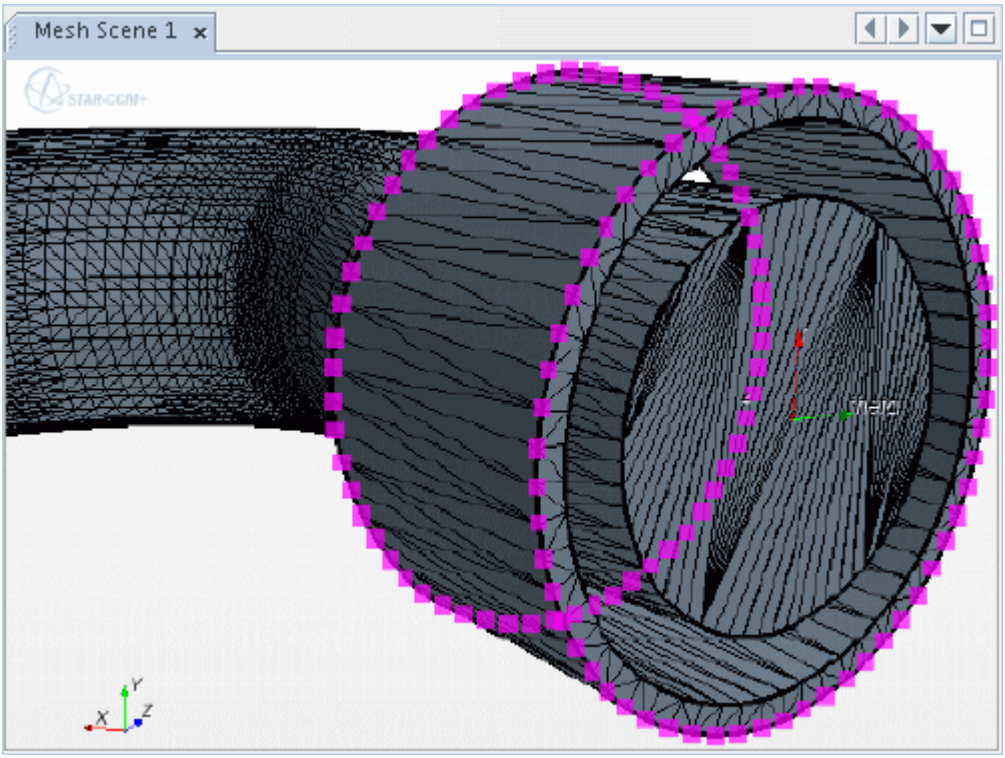


5. Open the *Project Vertices* in-place panel, and select the **Project in a Coordinate System** mode.
6. Select the cylindrical system for the **Coordinate System**, and enter the radial, R, **Projection Ordinate** value.



7. The projection is performed when you click **Project Vertices**.

   The *Graphics* window displays the new result.



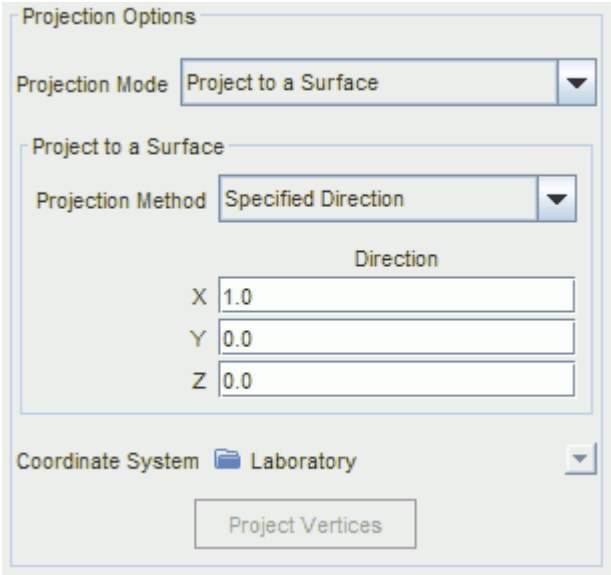
The tool maintains the original vertex selection, so you can repeat the process for different radial values.

Projecting Vertices To A Surface

The **Project to a Surface** mode projects one or more vertices to a set of faces.

The **Project Vertices** button is only available if at least one vertex and one face have been selected for projection. By default, the tool projects using the shortest distance. However, projection in a specified direction is also possible. If projecting in a specified direction with two independent sets of faces selected, the tool projects to the nearest face, regardless of whether it is in the positive or negative direction. In all instances, the tool ignores vertices that belong to the selected surface.

The image below shows the default **Project to a Surface** mode. As you can see, projection using the shortest distance does not require a coordinate system.



If you select **Specified Direction** for **Projection Method**, the **Coordinate System** menu becomes available, along with the projection **Direction**.

1. First select the **Coordinate System**.

   Selecting the coordinate system (**Cartesian**, **Cylindrical**, or **Spherical**) updates the **Direction** menu with the appropriate coordinate directions.

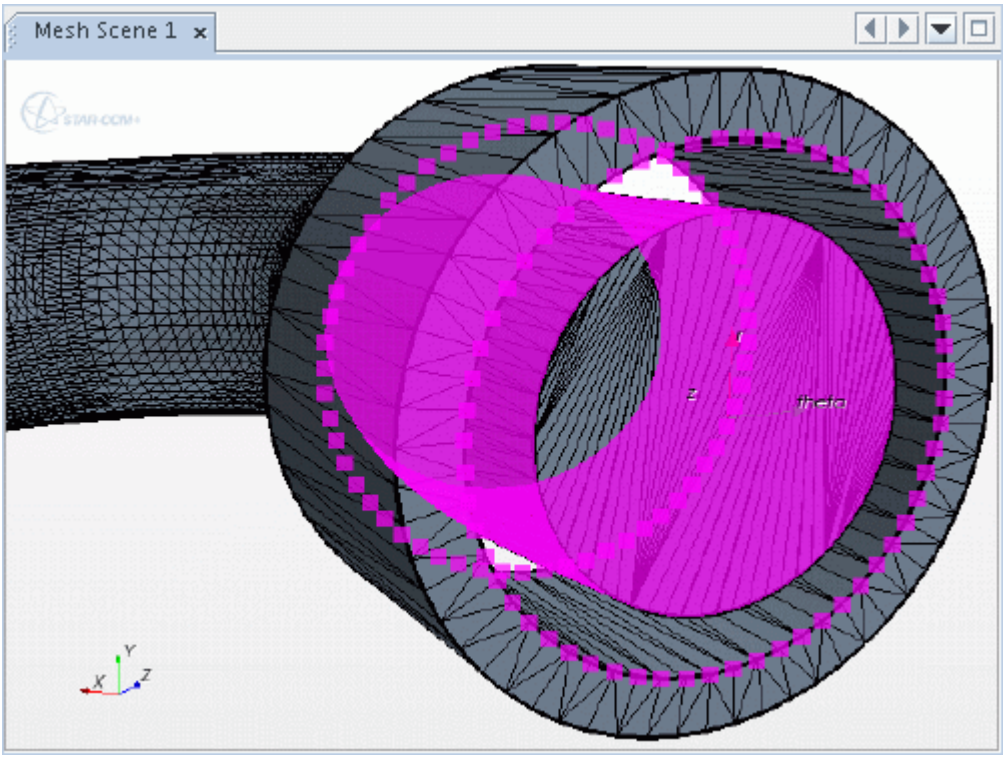
2. The projection is performed when you click **Project Vertices**.

   If projection is not possible in the specified direction, the vertex position remains unchanged and the *Output* window displays a warning message.

   Projecting vertices onto a convex surface using the shortest distance method results in “edging”, where projected vertices congregate along the edges of the selected surface. To avoid this effect, project along a specified direction instead.

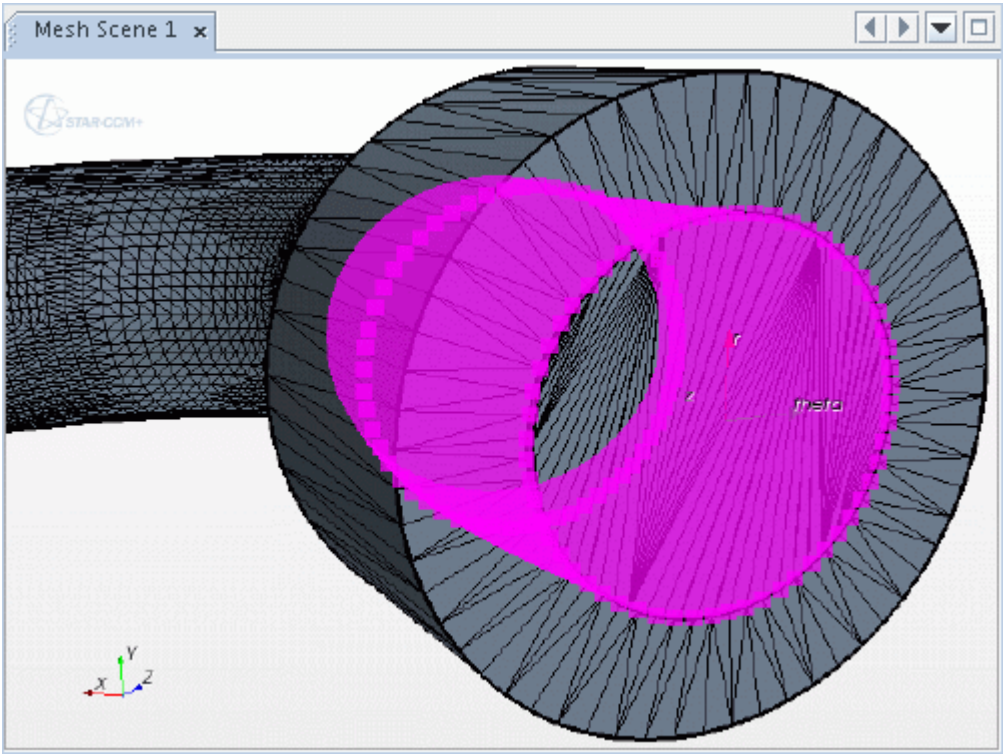
The following example projects vertices to a surface. The tool projects vertices on the inner surface of a sleeve onto the inner pipe surface, using the shortest distance method.

3. First, select the vertices on the sleeve inner surface, plus the faces on the pipe outer surface:



4. In the *Projection Options* panel, set *Projection Mode* to **Project to a Surface** mode.
5. The projection is performed when you click **Project Vertices**.

   The *Graphics* window displays the new result:

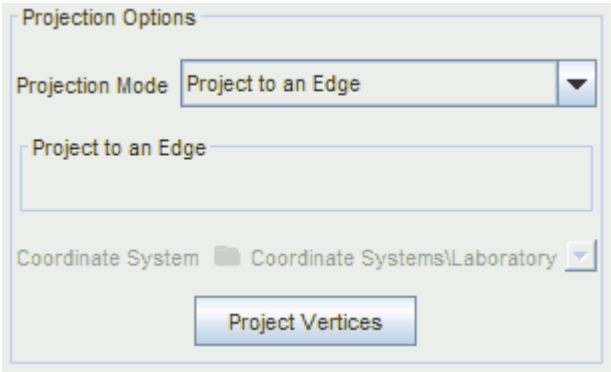


The tool maintains the original vertex and face selections, so you can repeat the projection with alternative settings.

Projecting Vertices To An Edge

The **Project to an Edge** mode allows the projection of one or more vertices to the nearest edges.

The **Project Vertices** button is only available if at least one vertex and one edge have been selected for projection. The tool ignores any vertices that belong to the selected edges.

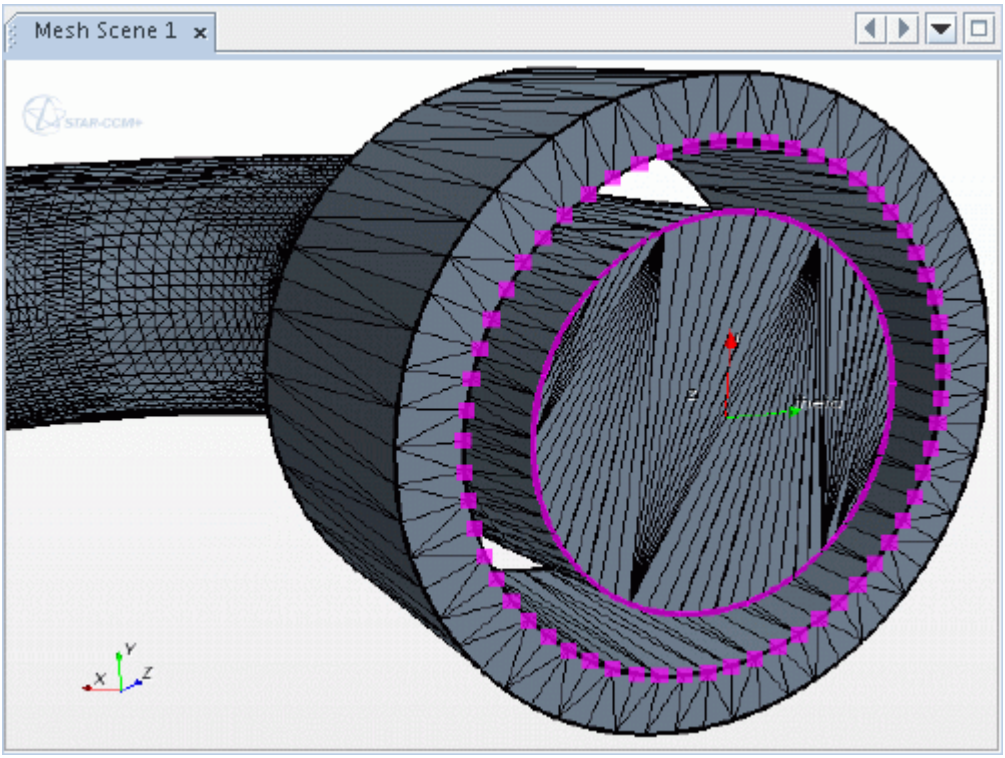


The mode requires no additional options. The projection is performed when you click **Project Vertices**.

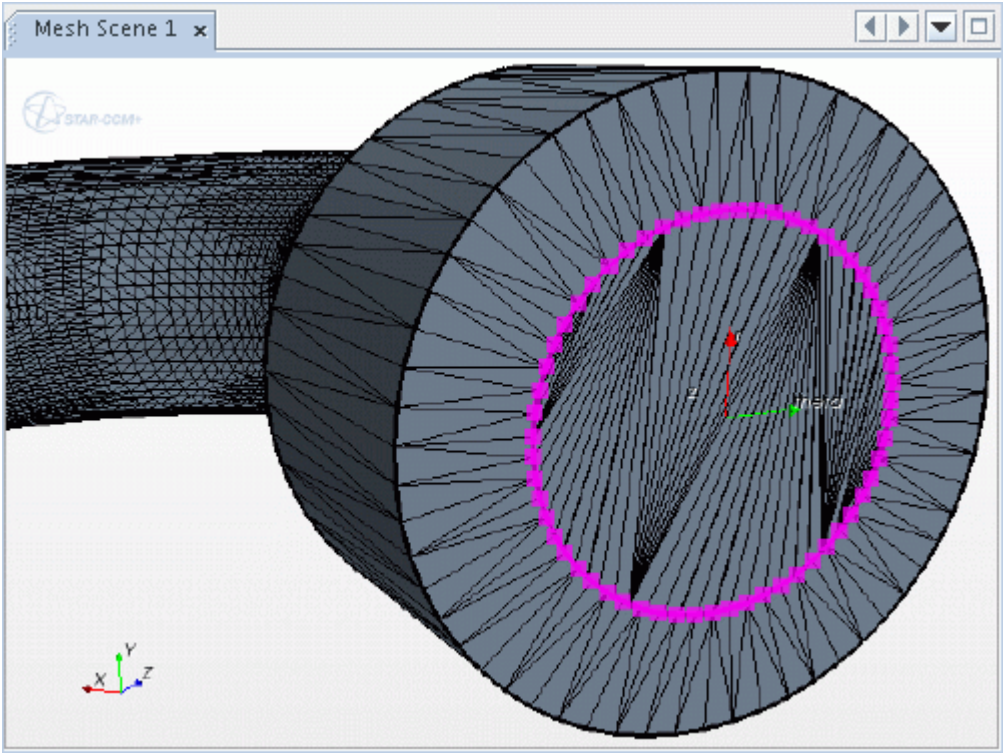
The following example projects vertices to edges. The tool projects vertices on the inner edge of a sleeve onto the inner pipe edge.

The *Graphics* window displays the result.

1. First, select the vertices on the sleeve inner surface, plus the edge on the pipe outer surface.



2. In the *Projection Options* panel, set *Projection Mode* to **Project to an Edge**.
3. Click **Project Vertices**.



Picking Points To Define A Projection Plane Or Line

When using the Project to a Plane or Project to a Line modes, you can pick positions in the *Graphics* window to define points.

This feature provides a quick and intuitive way of defining a plane or line.

To begin a pick operation, click one of the **Pick** buttons in the *Project Vertices* in-place dialog. You can then pick vertices, edges, or faces in the *Graphics* window. Picking a vertex uses the exact position of the vertex. Picking an edge uses the selected position along the direction of the edge and picking a face uses the selected position on the surface of the face. The *Output* window reports whether a face, edge, or vertex has been selected and the coordinates relative to the global coordinate system.

When a pick is made, the coordinates of the point are added to the appropriate fields in the *Project Vertices* in-place dialog. You can manually change the coordinates of the point after picking to fine-tune its position. These coordinates are relative to the current coordinate system. Therefore, changing the coordinate system relocates the point to a position relative to the new coordinate system. The point is represented in the *Graphics* window with a color-coded sphere, which automatically updates when the point position changes.

To cancel the pick operation, press `Esc` on the keyboard, or click in the white space in the *Graphics* window.

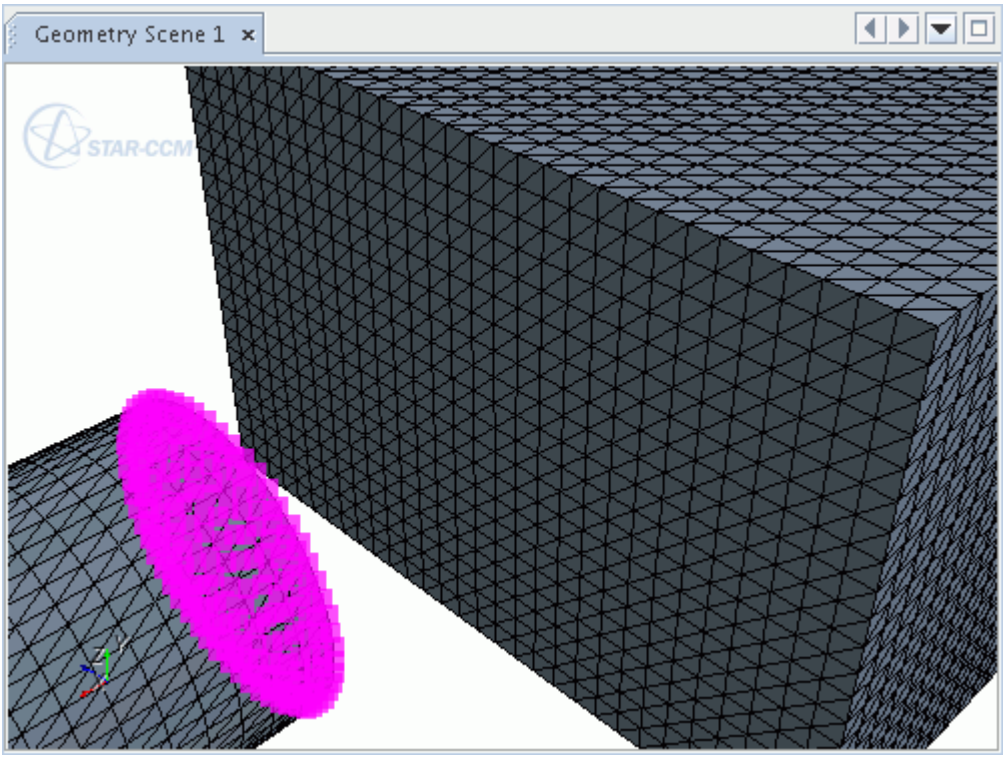
This section provides the following examples:

- Projecting to a plane defined by picking three points
- Projecting to a line defined by picking end points

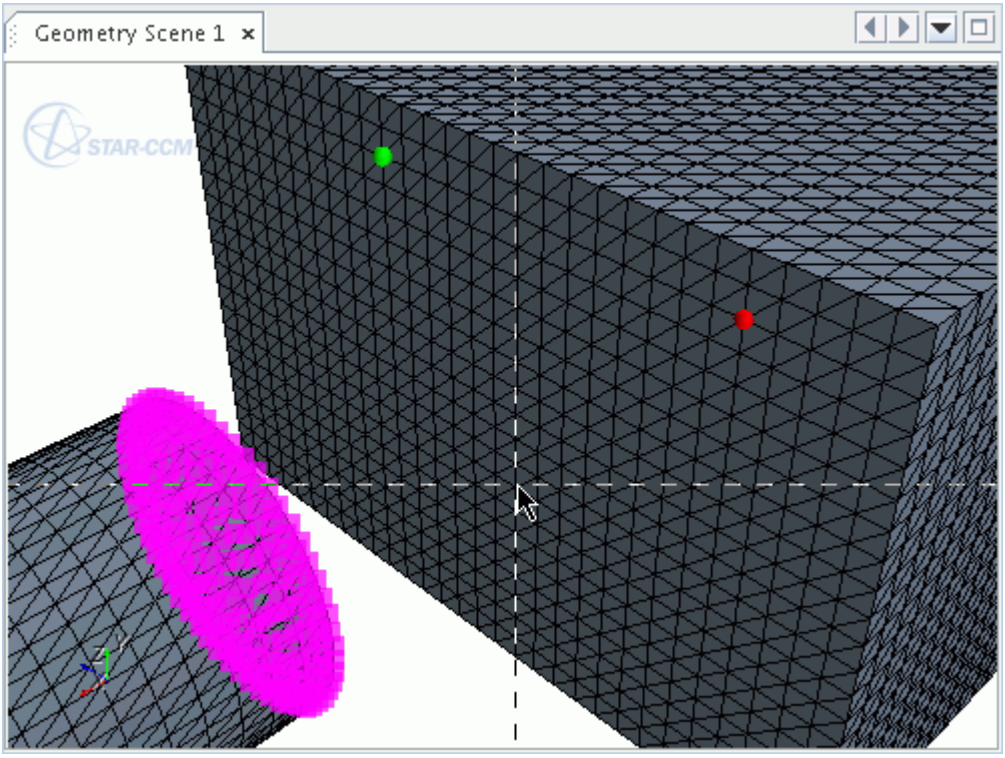
Picking Three Points to Define a Plane

This example illustrates picking three points to define a plane, in order to project a group of vertices to a surface.

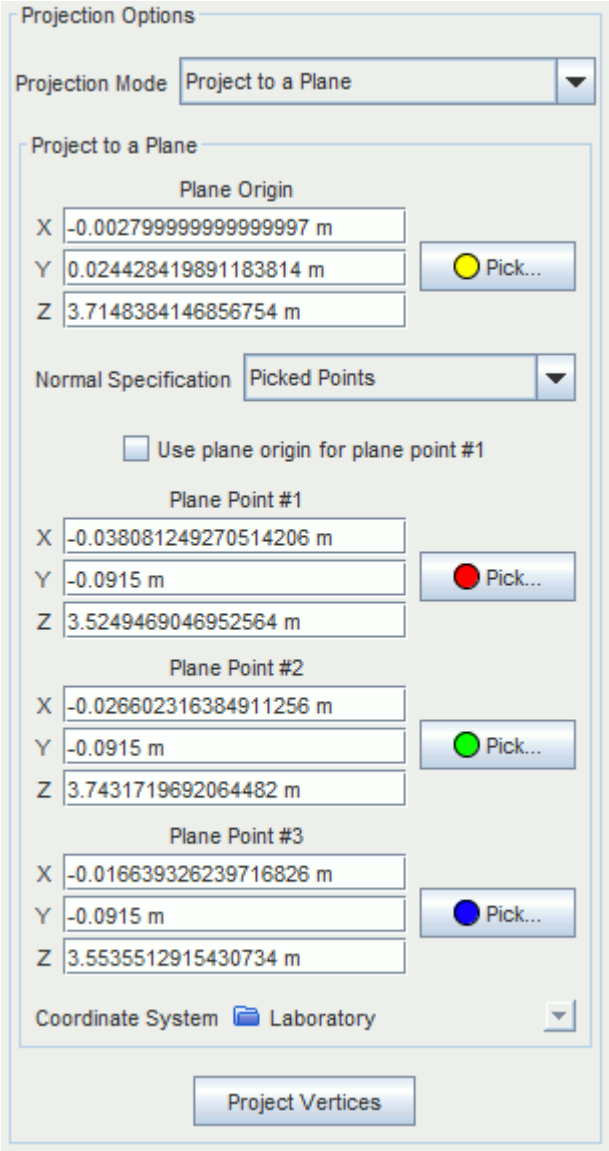
1. First, select the required vertices using the *Selection Controls*, before activating the *Project Vertices* tool.



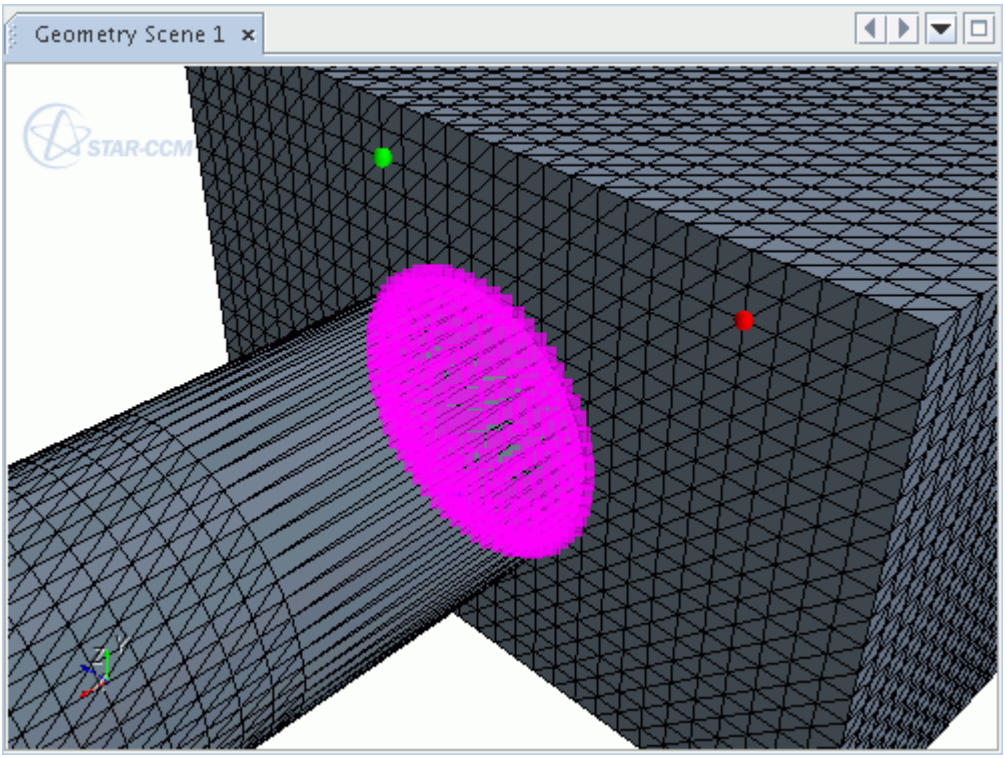
2. Define a plane by picking points on the required surface in the *Graphics* window.
   In this example, three vertices are selected, with the selection of **Plane Point #3** shown below:



The coordinates of each point are reported in the *Project Vertices* in-place dialog:



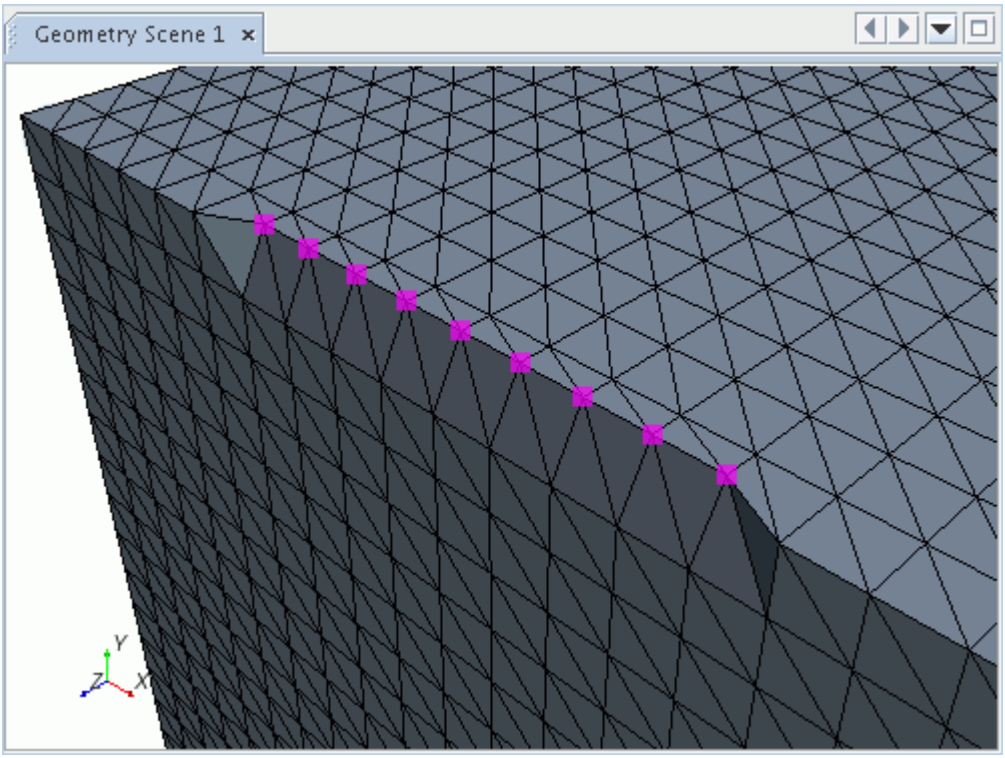
3. Clicking **Project Vertices** projects the selected vertices to the user-defined plane.



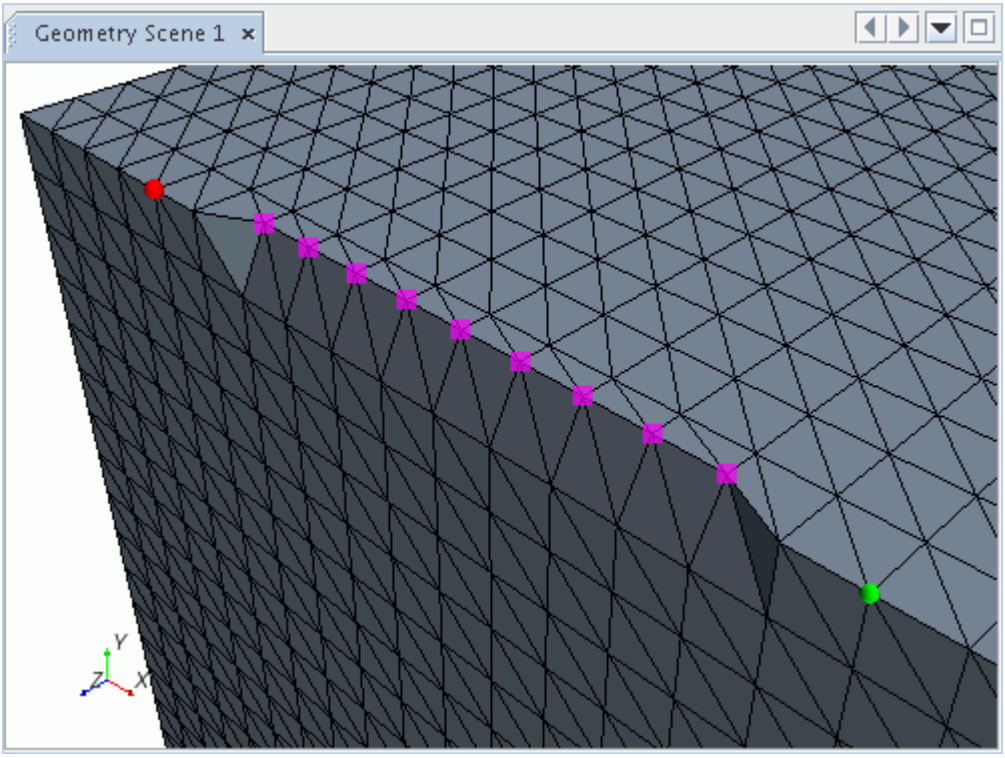
Picking End Points to Define a Line

The following example illustrates picking end points to define a line, in order to project a group of vertices the line.

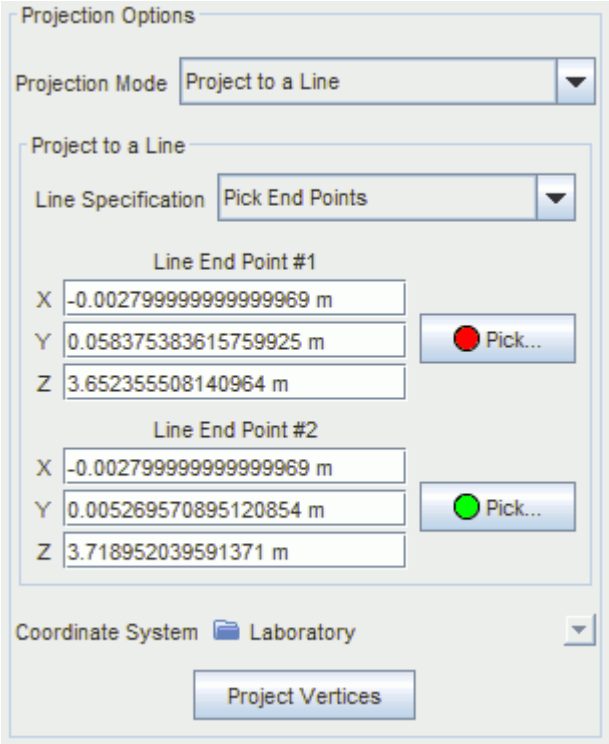
4. Select the required vertices using the *Selection Controls*, before activating the *Project Vertices* tool.



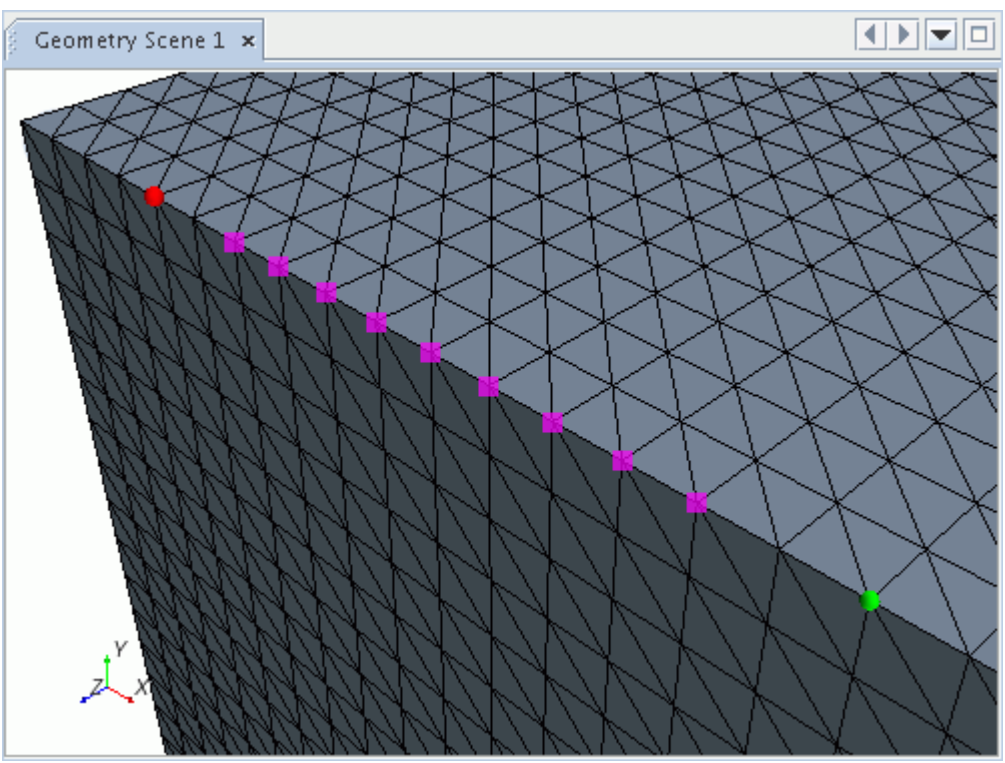
5. Define a line by picking the end points in the *Graphics* window.
   In this example, two vertices are selected:



The coordinates of each point are reported in the *Project Vertices* in-place dialog:



6. Clicking **Project Vertices** projects the selected vertices to the user-defined line.



See Also:

Projecting Vertices to a Plane
Projecting Vertices To A Line

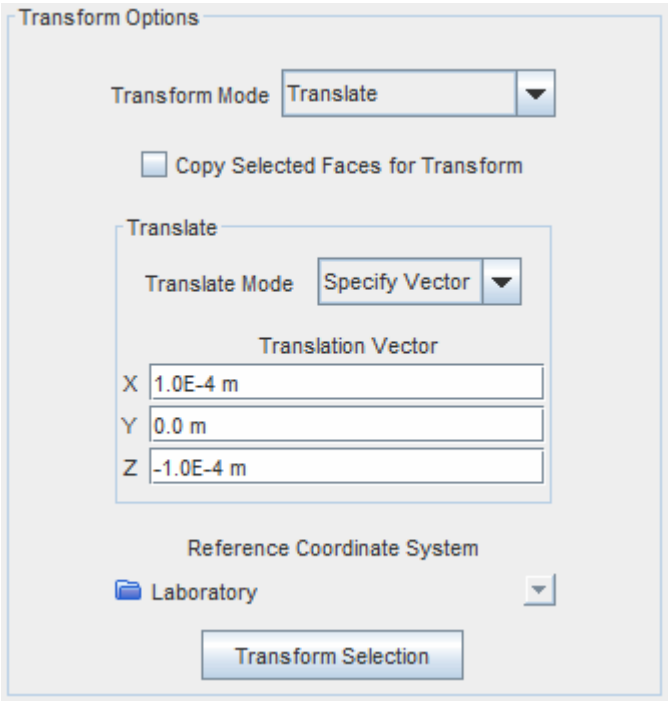
Transforming Surface Entities

This tool transforms surface entities (vertices, edges, or faces) in a local coordinate system using one of five modes.

To transform surface entities:

1. Click (**Transform vertices, edges and/or faces**).

   The *Transform Options* options panel appear.



2. Select the required transformation from the *Transform Mode* menu.
3. Select the required vertices, edges, and/or faces.

   For faces, you can choose to transform a copy of the faces and keep the original faces in place. To do this, activate *Copy Selected Faces for Transform*.

   See Using the Copy Selected Faces for Transform Option
4. Enter the required input values and make any necessary selections.

   Transformation takes place relative to the current position of the selected vertices.
5. Click **Transform Selection**.

The tool remains available after transformation so you can redo the operation with different values or specify a new transformation.

You can only use coordinate systems that were available when you opened the in-place panel. If you create a coordinate system while using the tool, it is not available until you exit and restart the **Transform vertices, edges and/or faces** tool. Also note, any values that you enter into the in-place panel are saved when you click **Return** and leave the panel.

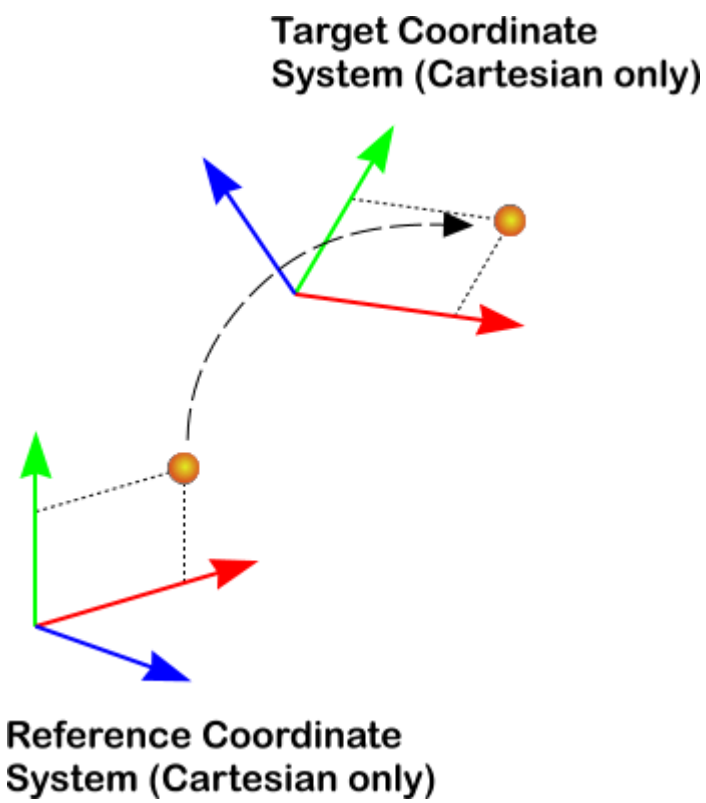
Contents:

Transforming Surface Entities to a Coordinate System
Translating Surface Entities
Rotating Surface Entities
Scaling Surface Entities
Reflecting Surface Entities
Using the Copy Selected Faces for Transform Option

Transforming Surface Entities to a Coordinate System

The **Coordinate Transform** mode transforms the selected surface entities from their position relative to a reference Cartesian coordinate system, to a new position relative to a target Cartesian coordinate system.

The diagram below illustrates the transformation for a single vertex. The ordinates of the position are the same in each Cartesian coordinate system.



- You can only use **Coordinate Transform** mode with a Cartesian coordinate system that was available when you opened the in-place panel.

The tool does not list non-Cartesian coordinate systems in the drop-down menus.

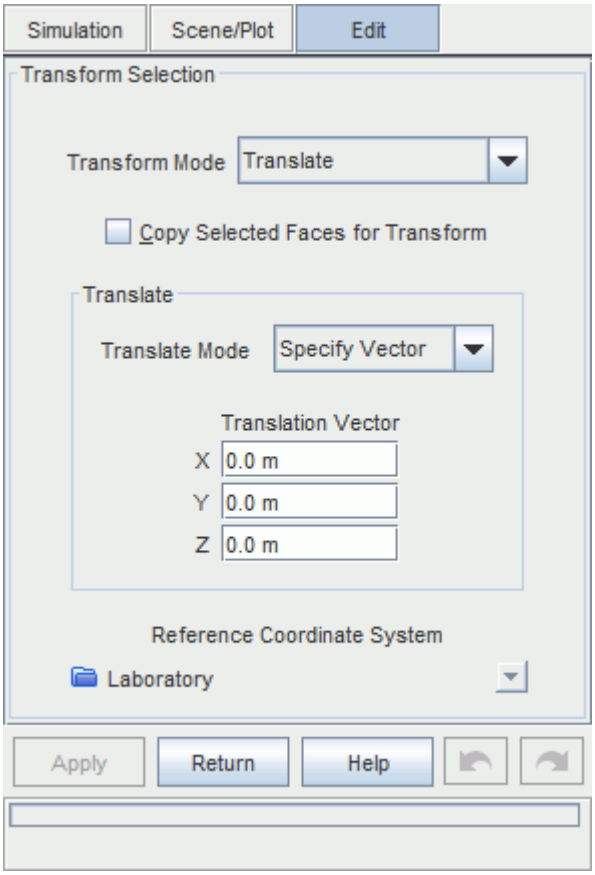
- You can use **Coordinate Transform** mode to transform an entire part from one coordinate system to another, while maintaining the relative shape of the part.

Translating Surface Entities

The **Translate** mode translates vertices, edges, and/or faces along a specified vector, relative to a selected coordinate system, or a specified distance along the vertex normal.

Translating Surface Entities Along a Specified Vector

The image below shows the default **Translate** mode, with **Specify Vector** selected from the **Translate Mode** drop-down menu.



The **Translation Vector** components specify the desired translation. These components are relative to the **Reference Coordinate System**, which you select from the menu provided.

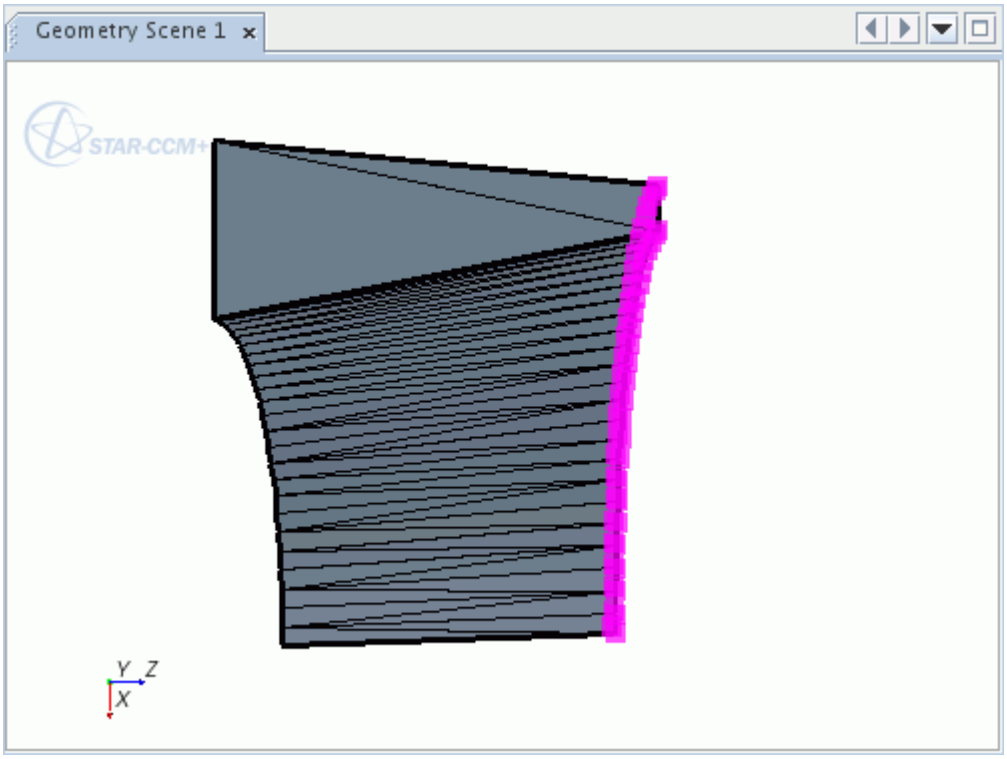
The **Reference Coordinate System** menu sets the coordinate system, with Cartesian, cylindrical, or spherical systems available. The **Translation Vector** component labels update to reflect the chosen type of coordinate system. The menu lists those systems that were available when you opened the dialog.

When translating vertices in a cylindrical or spherical coordinate system, the translation is relative to the component axes of these systems. For example, a translation in the R-direction moves the vertices in the radial direction, relative to the chosen coordinate system. Similarly, a translation in the q-direction rotates the

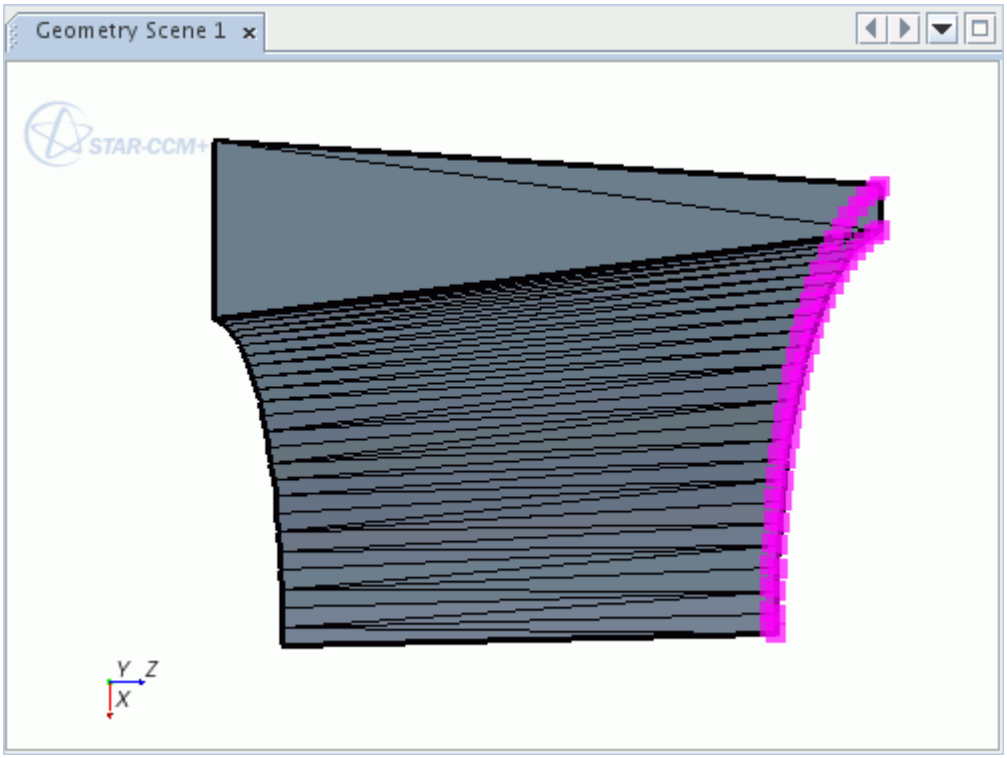
vertices about the origin. If a vertex is at R=0, translating in the radial direction does not move the vertex. Also note that the tool permits negative radial translations.

Clicking **Apply** applies the translation, and report the result in the *Output* window.

The following example translates the selected vertices in the z-direction.



In this instance, the **Translation Vector** is `[0, 0, 0.05] m,m,m` to translate the vertices 0.05 m in the z-direction. Clicking **Apply** results in the following translation.

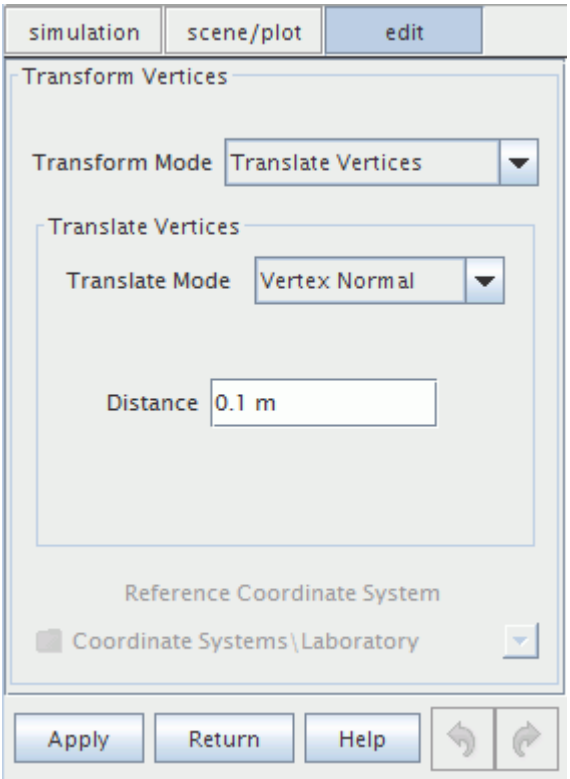


Translating Surface Entities Along a Vertex Normal

You can also translate vertices, edges, and/or faces in the normal direction by a user-defined distance. The normal is calculated as the average normal of all connected faces.

To translate vertices along a vertex normal:

- Select **Vertex Normal** from the **Translate Mode** drop-down menu.



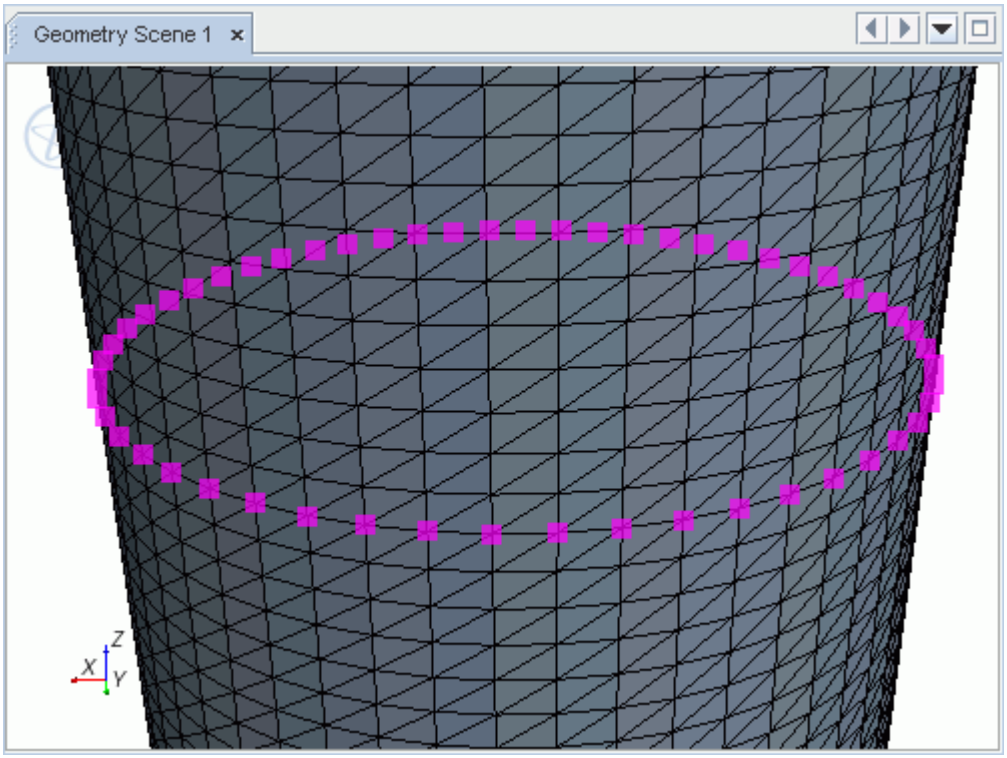
The **Distance** specifies how far the selected vertices are translated. The **Distance** is relative to the current position of the vertices that are selected, or the vertices that are attached to the edges, and/or faces that are selected. Applying the same distance twice, doubles the translation.

Clicking **Apply** applies the translation, and reports the result in the *Output* window.

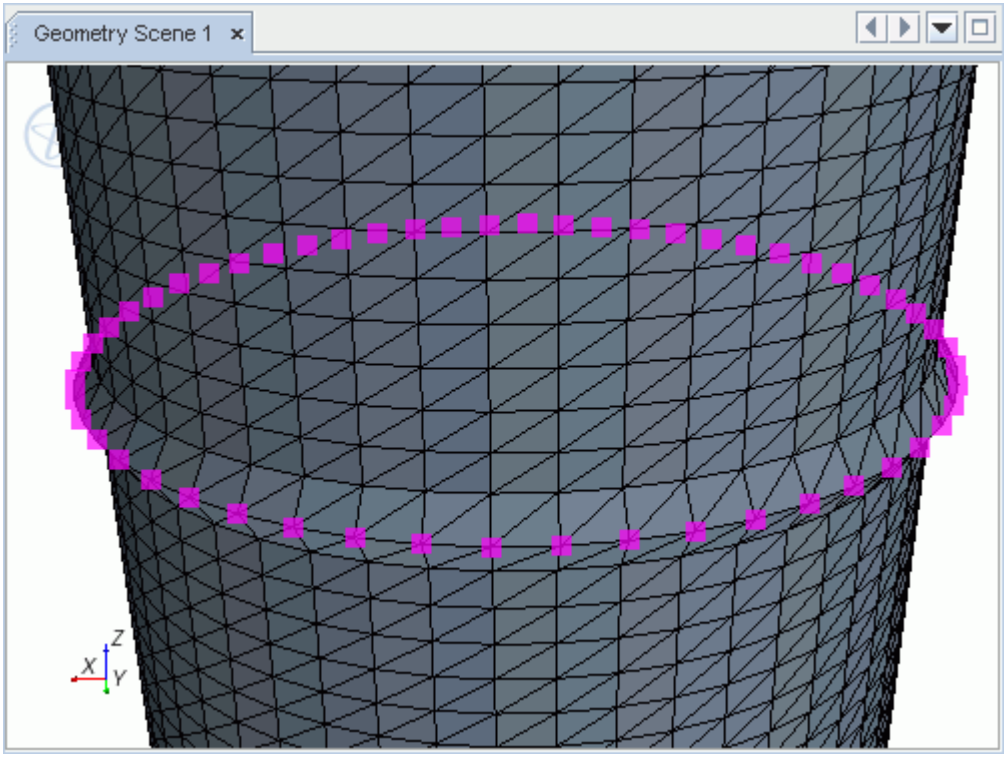
It is not always possible to calculate the vertex normal when one of the connected faces is of poor quality or if the connected faces converge to a sharp point. If the vertex normal cannot be calculated, the *Output* window reports:

```
Vertex normal calculation not possible for vertex XXXXX. This vertex
won't be translated.
```

The following example translates each selected vertex in its normal direction.



In this instance, as the surface is cylindrical, the calculated vertex normals offset the vertices, but maintain a circular shape.



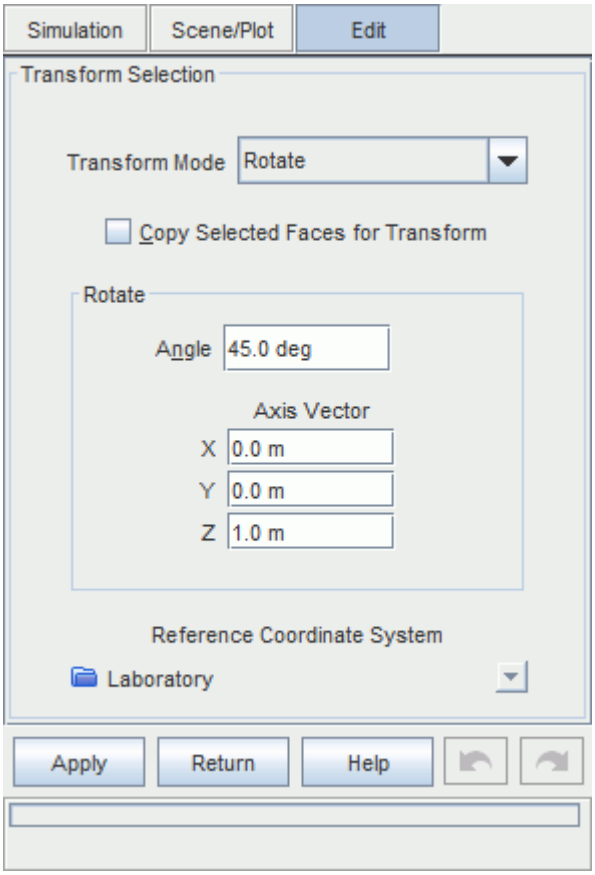
See Also:

Using the Copy Selected Faces for Transform Option

Rotating Surface Entities

The **Rotate** mode rotates the selected surface entities about an axis that is defined by vector components in the selected reference coordinate system. The rotation follows the right-hand rule.

To specify the angle in different units, enter the unit after the numerical value in the **Angle** entry box:



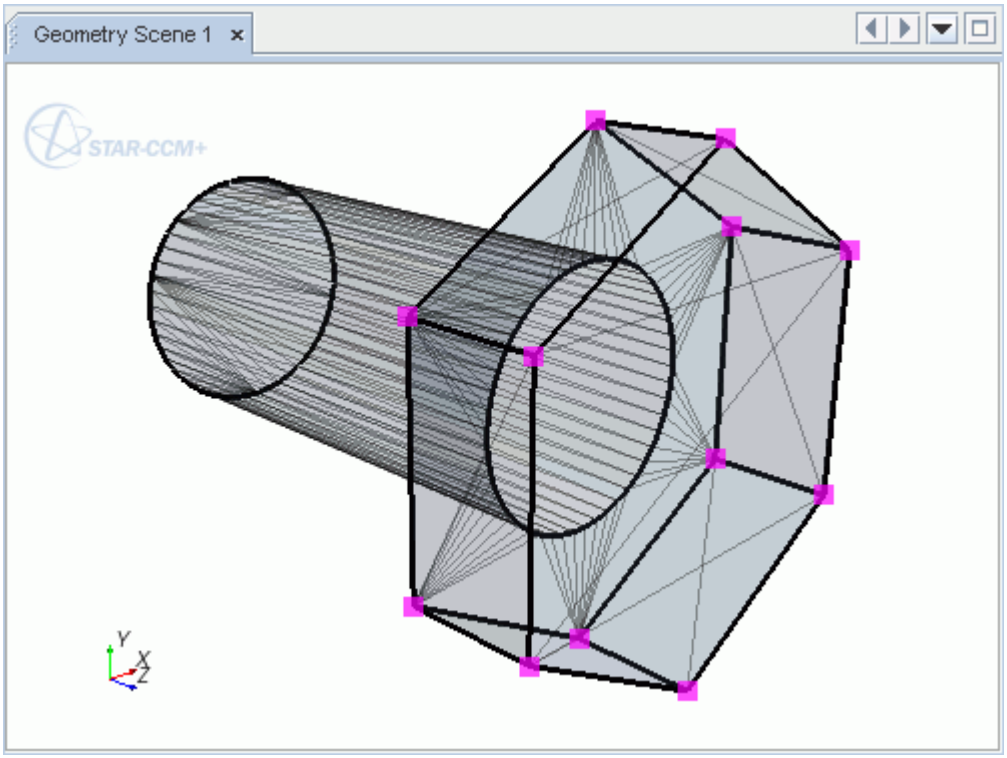
You can validate the angle specification by pressing `<Enter>` after entering the value and unit. If the unit specification is invalid, the **Angle** entry box appears red.

The **Axis Vector** component boxes define a vector about which the selected surface entities rotate. These components are relative to the **Reference Coordinate System**, which you select from the menu provided.

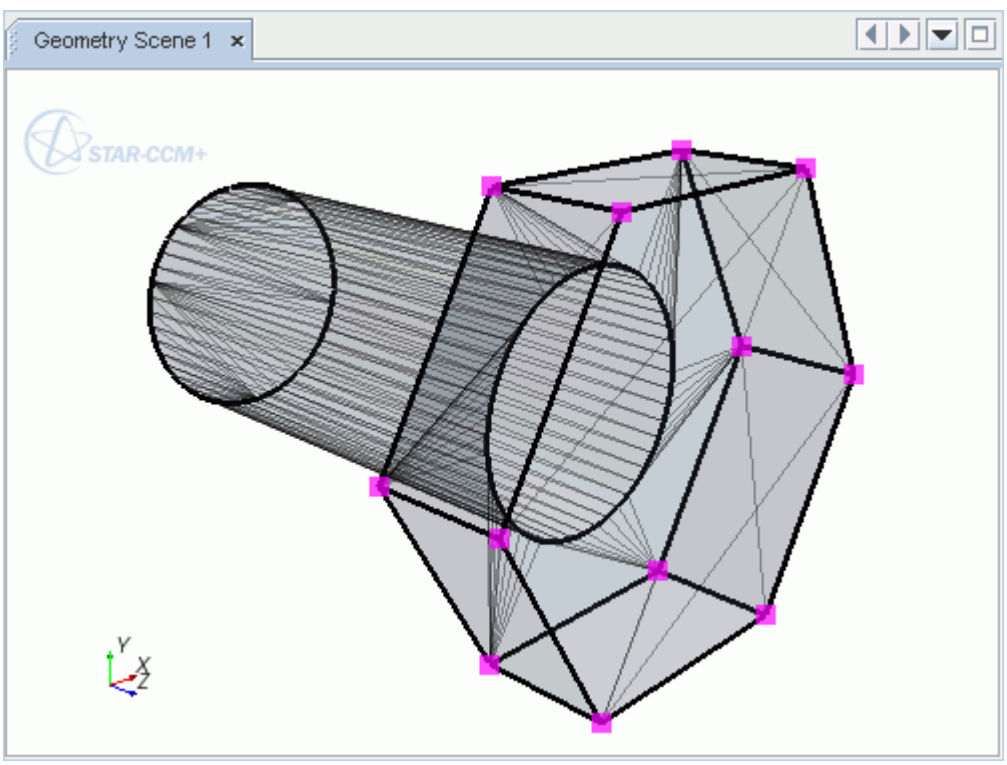
The **Reference Coordinate System** menu sets the coordinate system, with Cartesian, cylindrical, or spherical systems available. The **Axis Vector** component labels update to reflect the chosen type of coordinate system. The menu lists those systems that were available when you opened the dialog.

Clicking **Apply** applies the rotation, and records the result in the *Output* window.

The following example rotates each selected vertex about the z-axis.

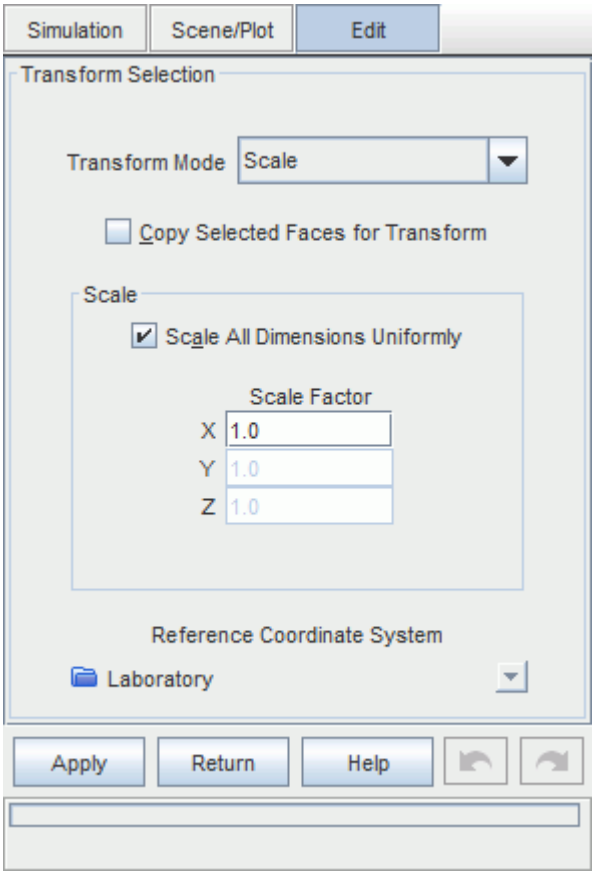


In this instance, the vertices defining the hexagonal volume rotate, but the vertices defining the cylindrical volume remain stationary. This effect can cause poor quality faces as the connectivity is maintained and faces become twisted.



Scaling Surface Entities

The **Scale** mode multiplies the position vector components of the selected entities by a user-defined scale factor.

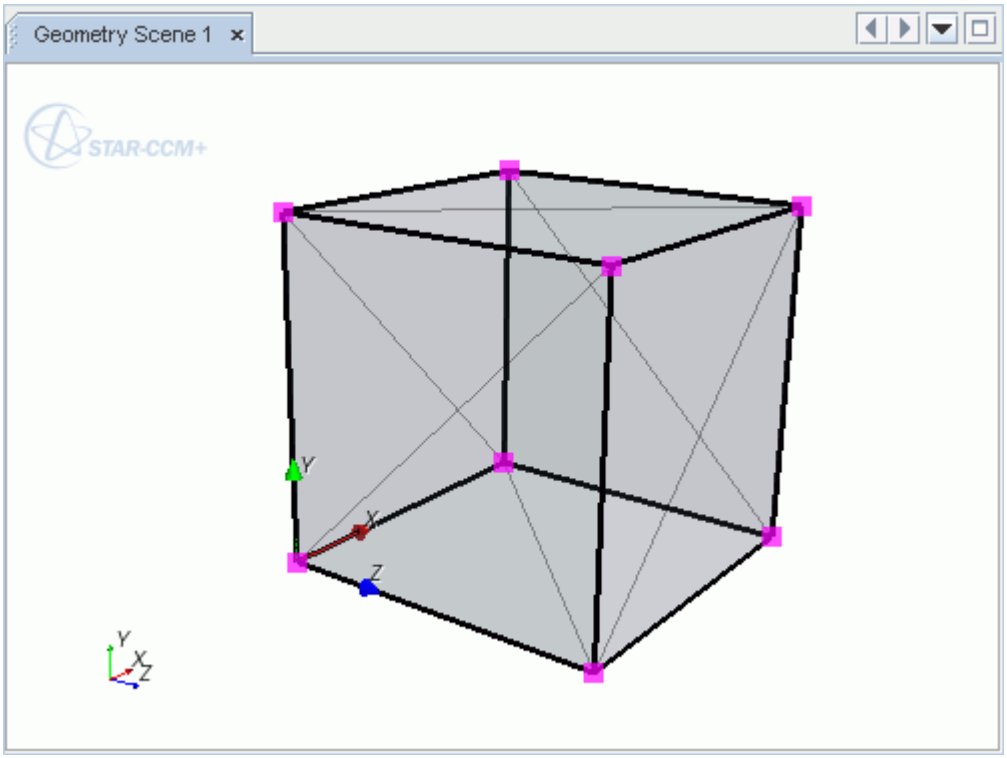


Ticking the **Scale All Dimensions Uniformly** option, which is activated by default, grays out the second and third *Scale Factor* boxes. The factor that is given in the first box then applies to all coordinate directions.

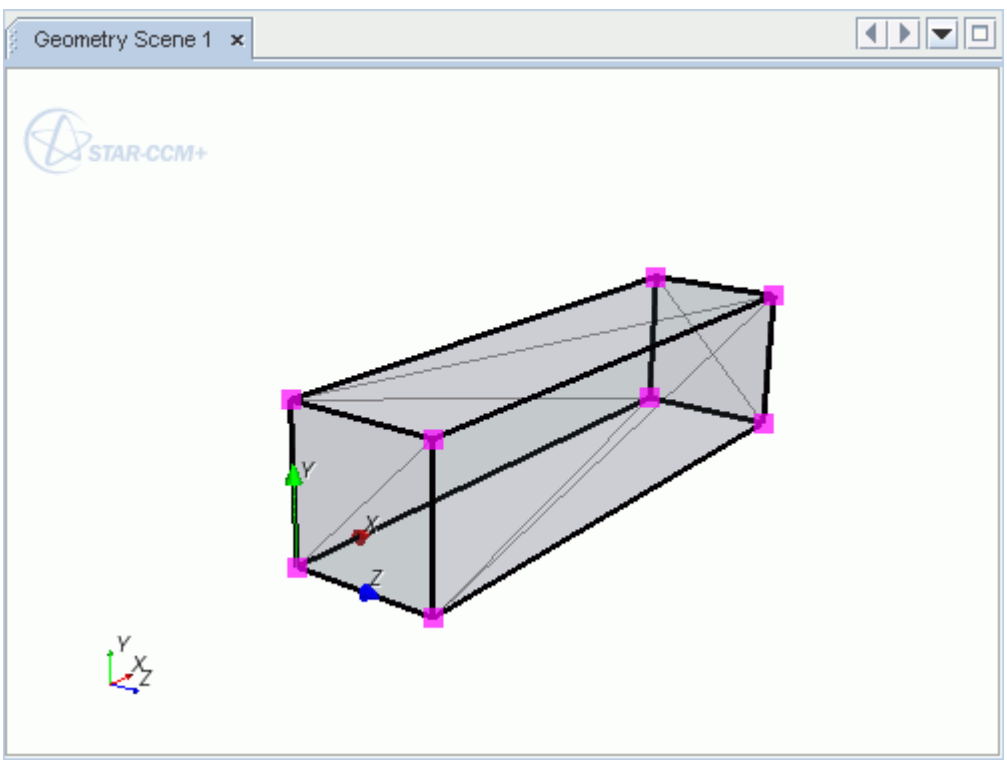
The **Reference Coordinate System** menu sets the local coordinate system, with Cartesian, cylindrical, or spherical systems available. The **Scale Factor** component labels update to reflect the chosen type of coordinate system. The menu lists those systems that were available when you opened the dialog.

Clicking **Apply** applies the scaling, and records the result in the *Output* window.

The following example scales the selected vertices:



In this instance, the **Scale Factor** in each dimension is `[2.0, 0.5, 0.5]`. As the box is located as shown by the coordinate system, the box is twice as long in the x-direction and half as long in the y- and z-direction.



Reflecting Surface Entities

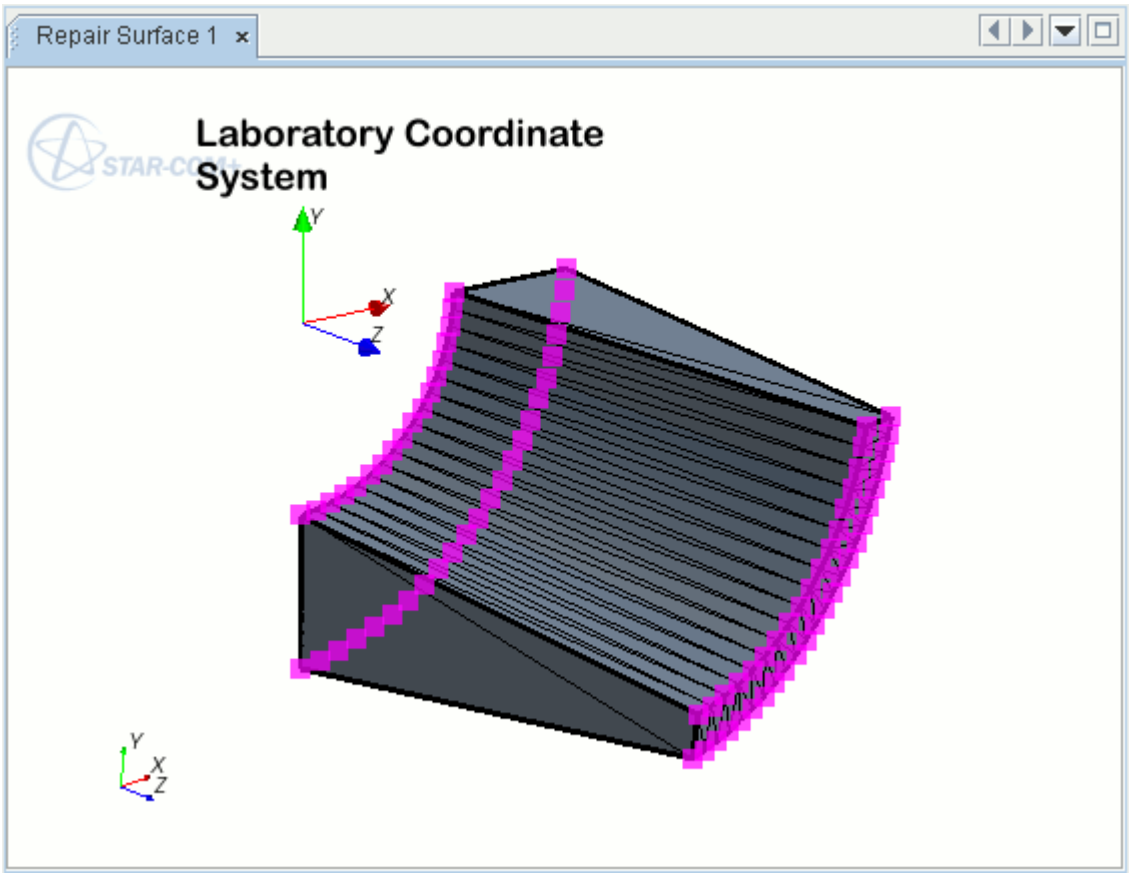
The **Reflect** mode reflects the selected entities about a plane with a user-defined normal vector.

The **Normal Vector** component boxes define a vector normal to the reflection plane. These components are relative to the **Reference Coordinate System**, which you select from the menu provided.

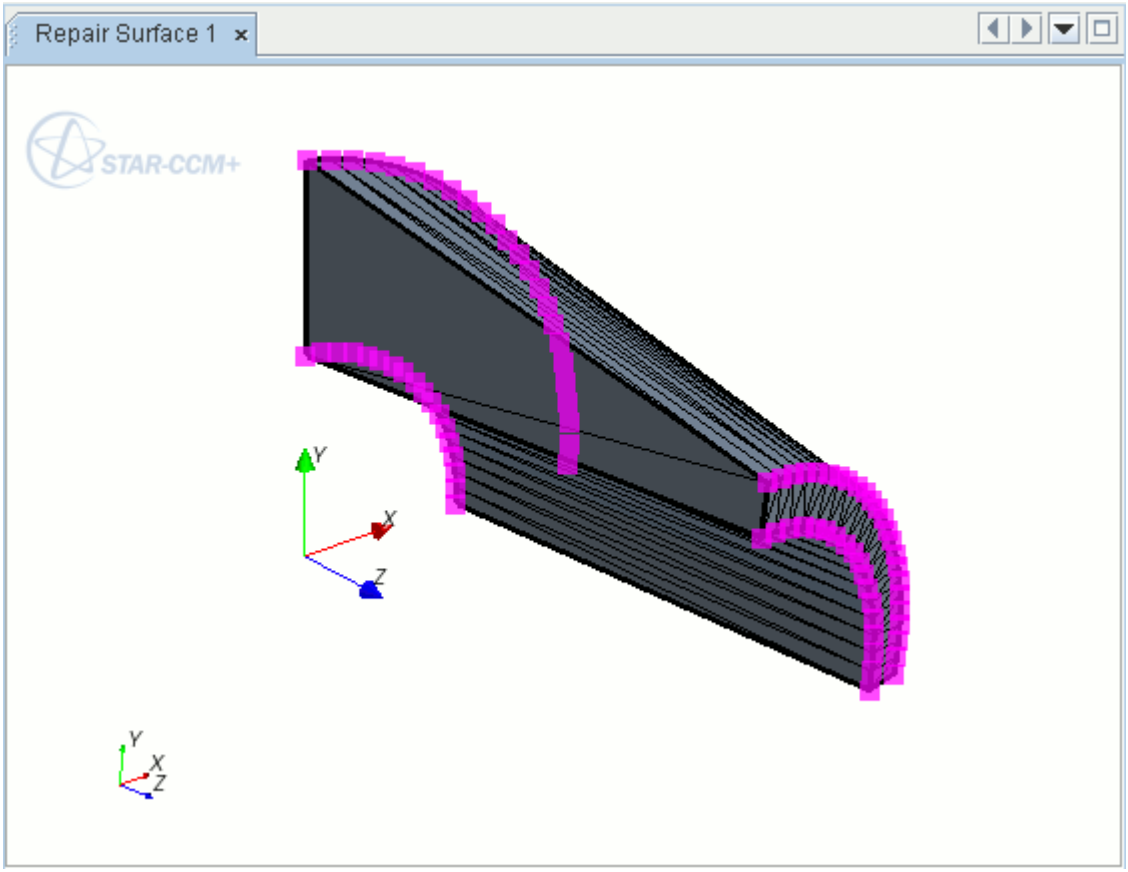
The **Reference Coordinate System** menu sets the local coordinate system, with Cartesian, cylindrical, or spherical systems available. The **Normal Vector** component labels update to reflect the chosen type of coordinate system. The menu lists those systems that were available when you opened the dialog.

Clicking **Apply** applies the reflection, and records the results in the *Output* window.

The following example reflects vertices about the X-Z plane of the laboratory coordinate system:



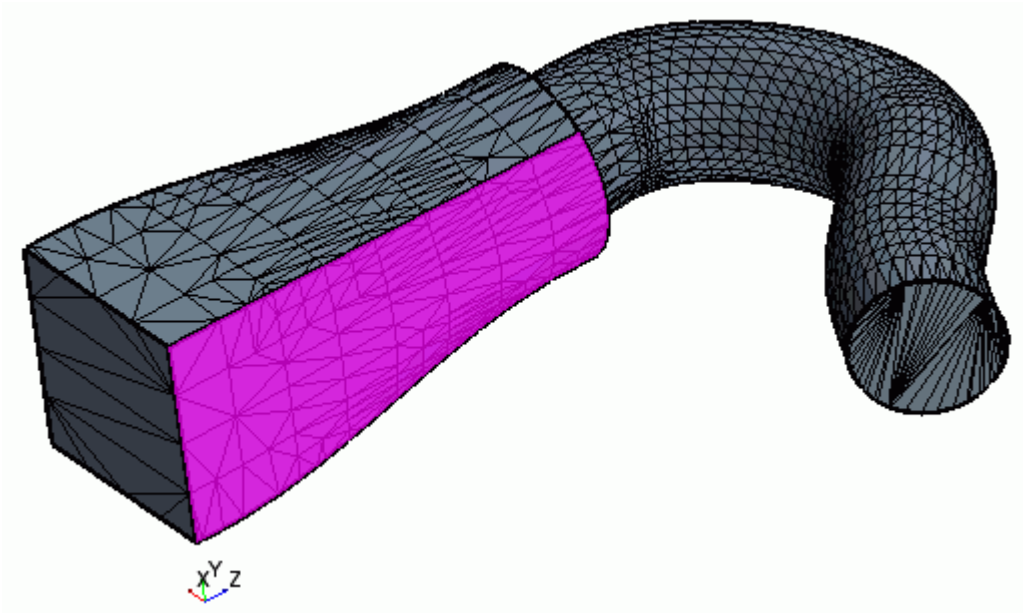
In this instance, the Y-axis is normal to the X-Z plane, so the entry for the **Normal Vector** is `[0, 1, 0]`. Clicking **Apply** results in the following reflection:



Using the Copy Selected Faces for Transform Option

Activating this option creates a copy of all faces that are selected and transforms the faces as specified.

In the example below, the highlighted faces are selected for a transform:

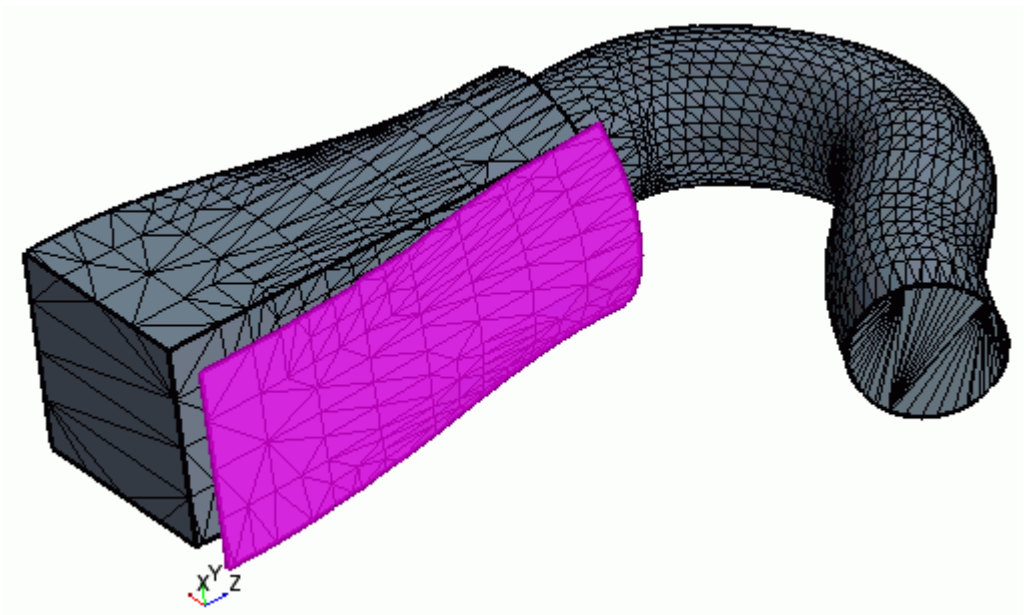


Note: At least one vertex, edge, or face must be selected to make the **Apply** button available. However, when *Copy Selected Faces for Transform* is activated, **Apply** is only available when faces are selected without edges or vertices.

When the following settings are applied:

- Transform Mode is set to Translate
- Copy Selected Faces for Transform is activated
- Translate Mode is set to Vertex Normal with a distance of 0.02 m

The result of the transform appears as follows:



Any feature curves that are attached to the selected faces are also copied. The selection set is also updated —the initial selection is lost and the latest selection contains only the copied faces and any attached feature curves.

Note: When faces or copies of faces are transformed, the transformed faces lose any CAD association.

The following example shows the result of a transform with the same settings as the example above, except *Copy Selected Faces for Transform* is deactivated:

Zero Translations

If **Transform Mode** is set to **Translate** and the translation parameters are specified as zero, the selected faces are copied and no transform takes place. Any CAD that is associated with the originally selected faces is maintained.

See Also:

Translating Surface Entities

Offset Faces/Edges Tool

The **Offset Faces/Edges** tool is used to inflate or translate faces or edges.

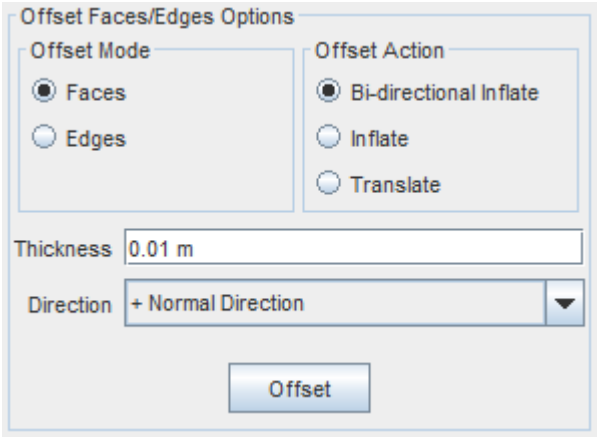
When inflating faces or edges, the selected entities are copied and offset a user-defined thickness. New faces are then created along the thickness to maintain connectivity between the original face or edge, and the new face or edge. When translating faces or edges, the selected entities are moved a user-defined distance and the connectivity with adjacent faces and edges is maintained.

Typical uses of the **Offset Faces/Edges** tool are illustrated in the following guided examples:

- Inflating a baffle surface
- Translating a surface
- Creating a zero-thickness baffle

Using **Offset Faces/Edges** tool is preferable to the wrapper inflation process, as it provides more control over the thickness and shape of baffle.

To display the *Offset Faces/Edges Options*, click (**Inflate or translate selected faces or edges**).



1. Select the surface or edges to offset using the Selection and Display controls. With the selections made, you can then set the offset properties.
2. Alternatively, you can use the surface repair tools to fix the non-manifold edges.

Offset Mode

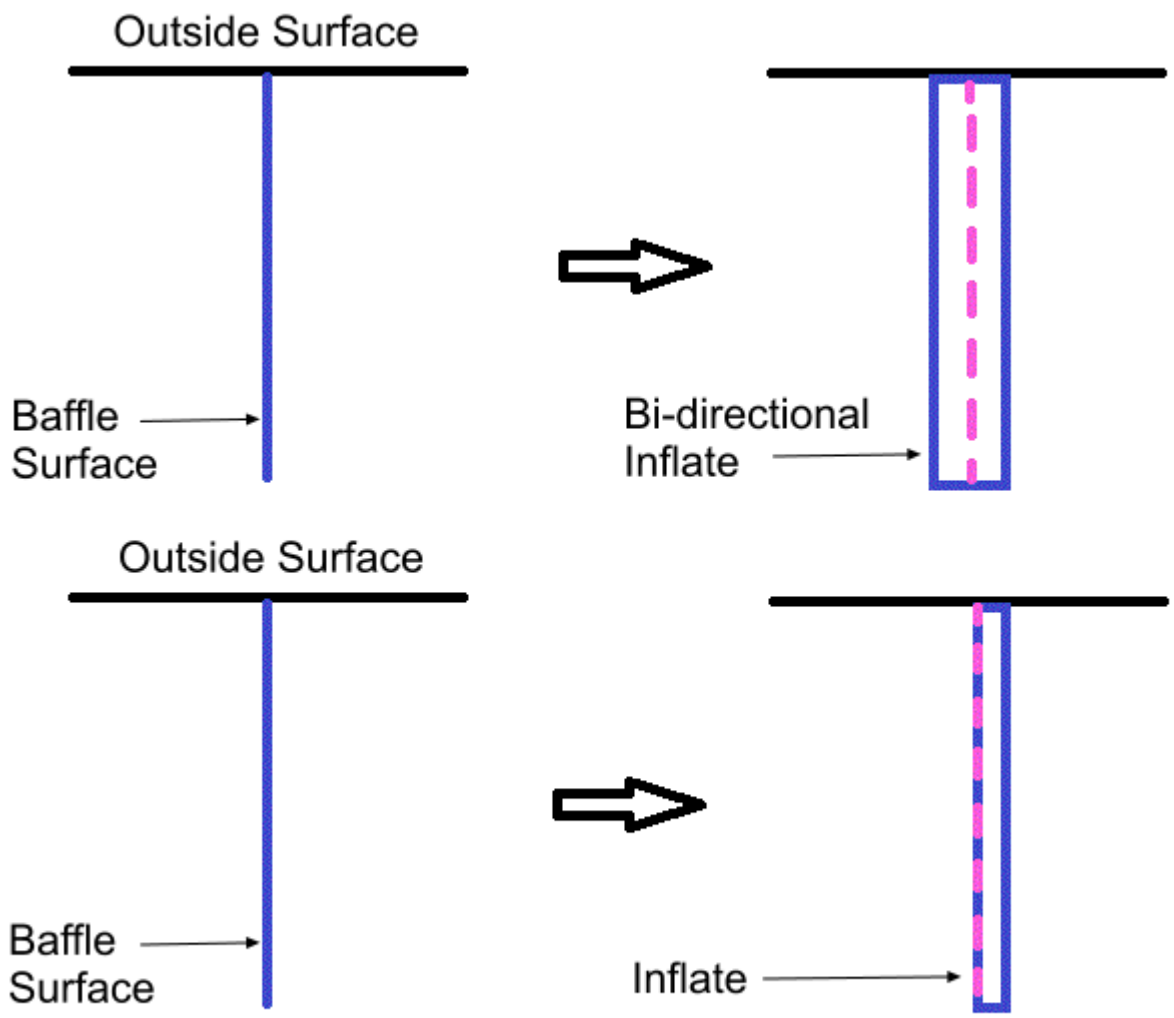
The *Offset Mode* determines whether you offset **Faces** or **Edges**.

In **Faces** mode, **Offset Faces/Edges** inflates surfaces so they have a thickness, typically surfaces representing baffles or internal walls. It also translates or extrudes a surface, moving its position, while maintaining connectivity with adjacent surfaces.

In **Edges** mode, **Offset Faces/Edges** inflates edges to create a two-dimensional surface, which can be used to create a zero-thickness baffle. It can also translate an edge, moving its position, while maintaining connectivity with adjacent edges and faces.

Inflate Offset Actions

The default options set *Offset Action* to **Bi-directional Inflate**, which inflates both sides equally, with a specified direction and thickness. Setting the *Offset Action* to **Inflate** allows you to inflate one side with a specified direction and thickness. The diagram below illustrates both of these options:

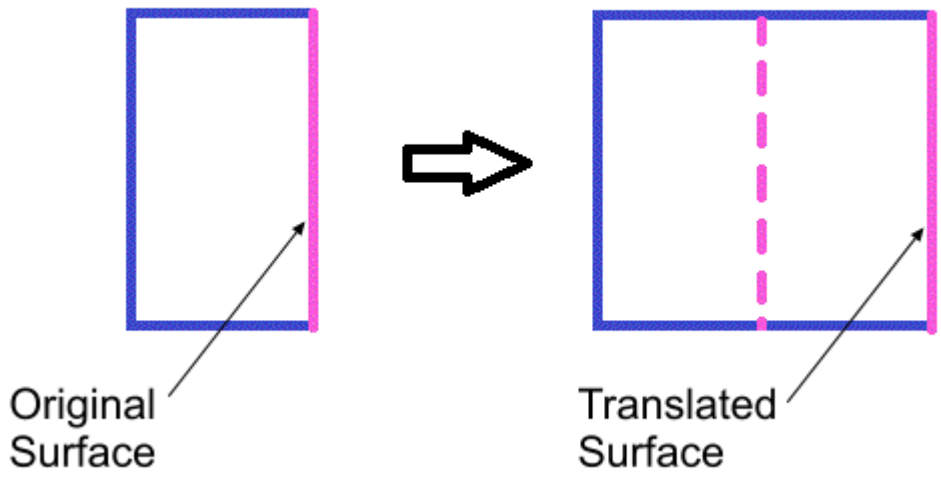


In the above examples, the baffle surface is still non-manifold. However, the surface wrapper can account for this effect, and produce a manifold surface.

Translate Offset Action

Setting the *Offset Action* to **Translate** moves the selected surface or edge, with a specified direction and thickness, and maintains the existing connectivity.

The diagram below illustrates this option:



Selecting a Direction Option

In each instance, you specify the direction of inflation using the **Direction** menu.

Selecting **Face Normal** from the **Color Face by** menu in the *Display Control*. Positive face normals are colored green and negative face normals are colored red. This feature is useful when choosing a **Direction**.

The following **Direction** options are available:

+ Normal Direction	Offsets the surface or edge in the positive normal direction. For edges, the average normal of the adjacent faces is used to determine the inflation direction. For **Bi-directional Inflate,** this option has no obvious effect, as the tool inflates in both the positive and negative direction.
- Normal Direction	Offsets the surface or edge in the negative normal direction. For **Bi-directional Inflate**, this option is not available, as it has the same effect as using the **+ Normal Direction** option.
Specified Direction	Offsets the surface or edge in a user-defined direction. This option requires **Direction Vector Components** in a given **Coordinate System** to specify the direction. This option is not available for cylindrical or spherical systems. You can only define a **Coordinate System** for cartesian systems.
Seed Point	Offsets the surface or edge in the normal direction, towards a user-defined seed point. In cases where there are multiple connected surfaces or edges, the tool uses a majority rule to determine the offset direction. This option is useful when using Bi-directional Inflation on surfaces or edges that have normals in various directions.

Setting a Thickness or Distance

With the direction option selected, you then set the **Thickness**, which defines the inflation thickness.

- Similarly, when using **Translate**, you set the **Distance**, which defines the translation distance.
- When using **Bi-directional Inflation**, the **Thickness** sets the total inflation thickness, meaning the tool inflates each side by half of this value.

Offsetting Faces

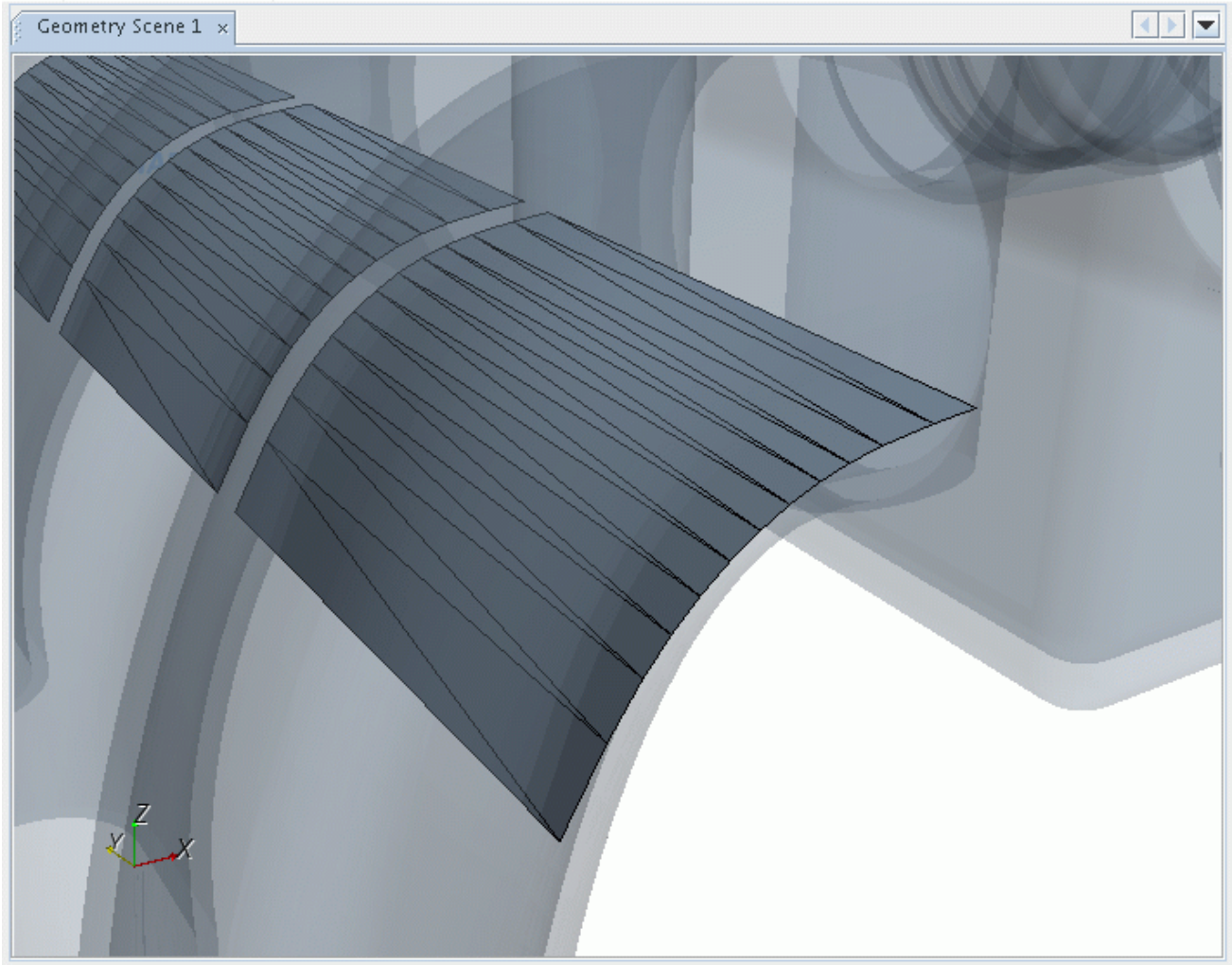
When you have provided all necessary settings, click **Offset** to carry out the operation.

Contents:

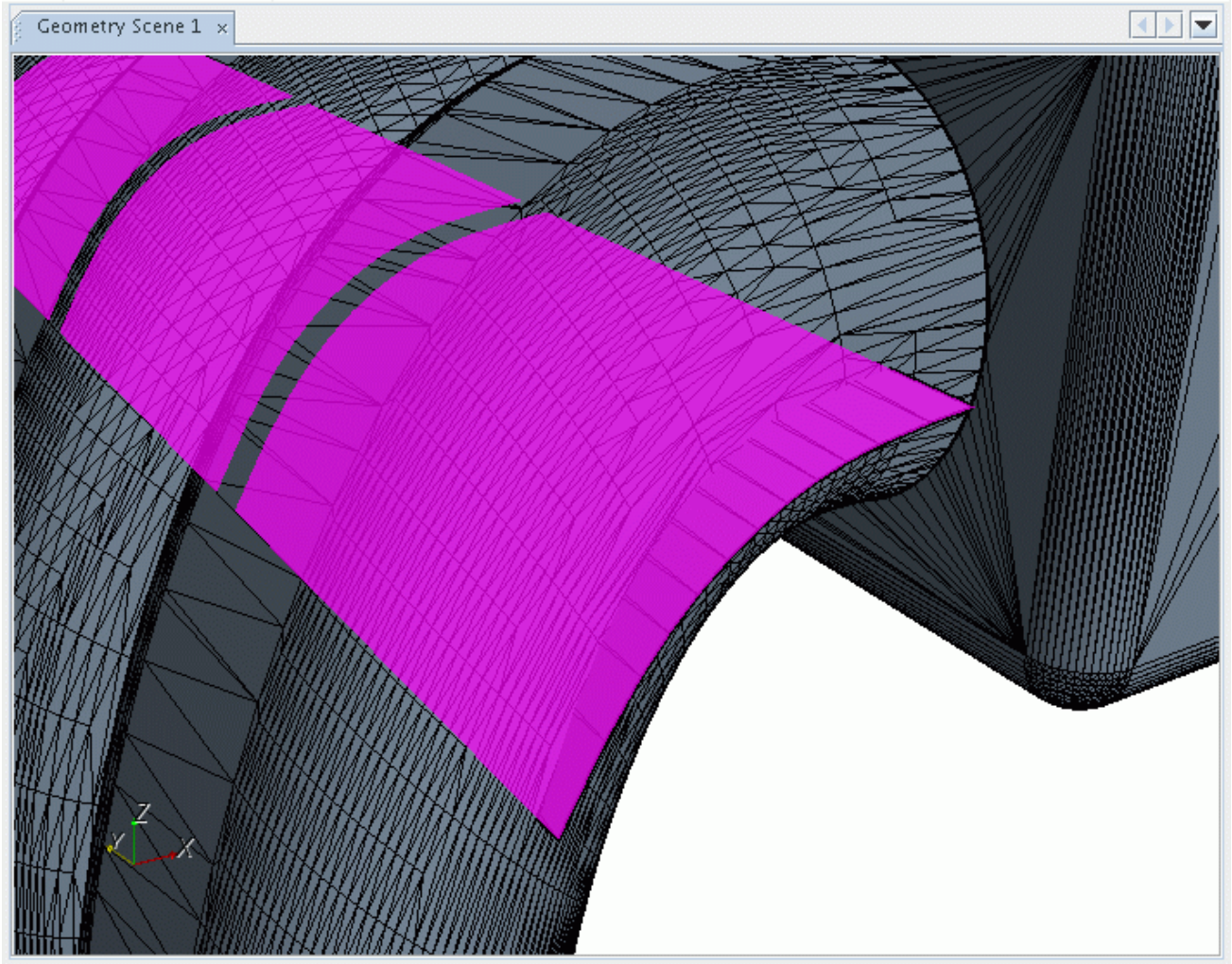
Inflating A Baffle Surface Using The Offset Tool
Translating A Surface Using The Offset Tool
Creating A Zero Thickness Baffle Using The Offset Tool

Inflating A Baffle Surface Using The Offset Tool

This example shows the inflation of a baffle surface using the **Bi-directional Inflation** option. Baffle surfaces can connect to the surrounding geometry, be free-standing, or intersect the geometry. In this case, the baffles intersect the geometry.

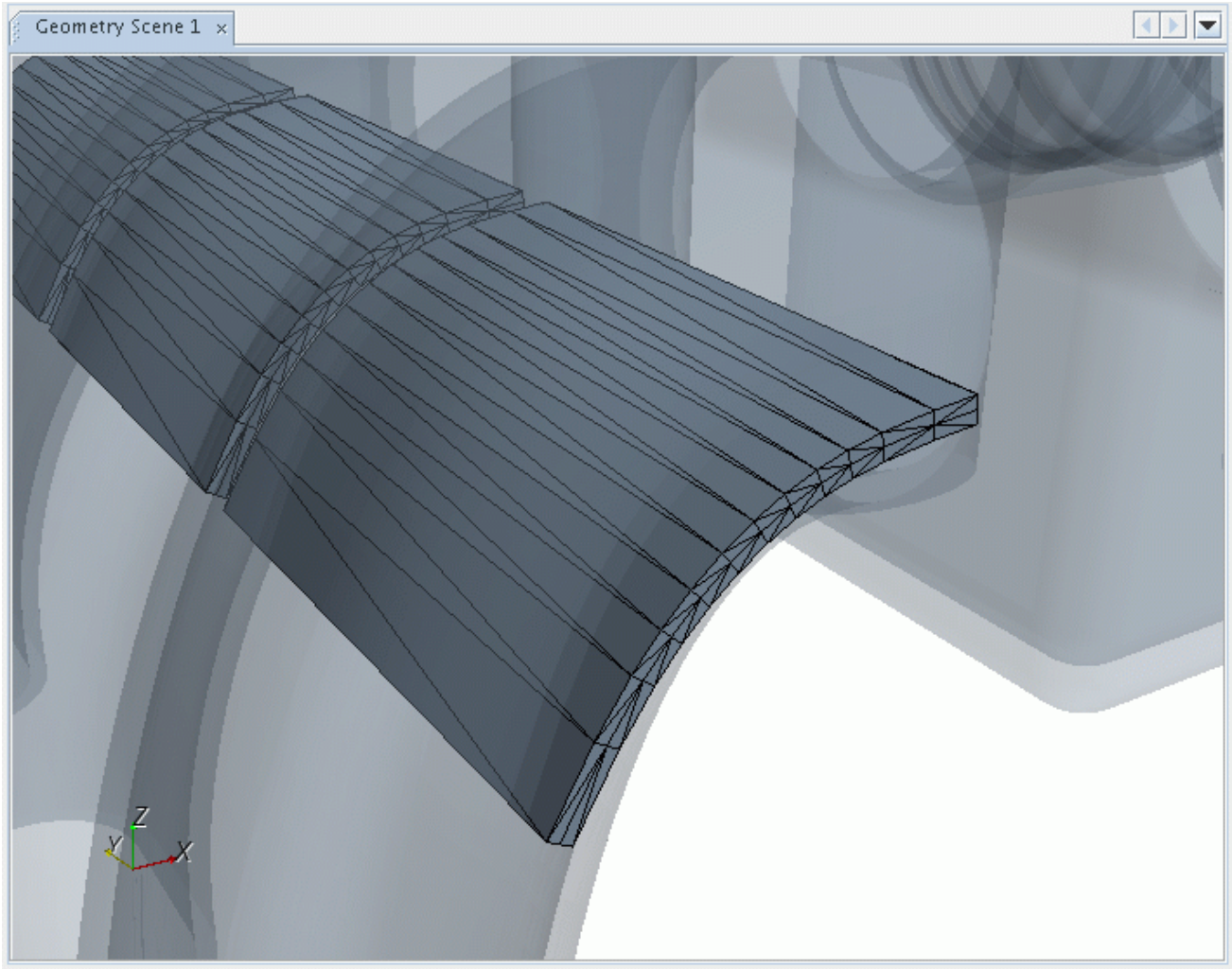


You first select the baffle surface using the Selection Controls. In this case, the **Select faces / edges on part surfaces / part curves** tool selects the three internal baffle surfaces.



You then select the required *Offset Action*. This example uses the **Bi-directional Inflation** option to inflate both sides of the baffle. The **Direction** is **+ Normal Direction**, which inflates the baffle in the normal direction of each face. The **Thickness** is an arbitrary value, with any allowable units.

To carry out the inflation, click **Offset**. The *Graphics* window updates with the modified surface.



If the starting surface contained feature curves, then these curves are also inflated and included on the new surface. Any new faces created by the process belong to the same surface as the originating face.

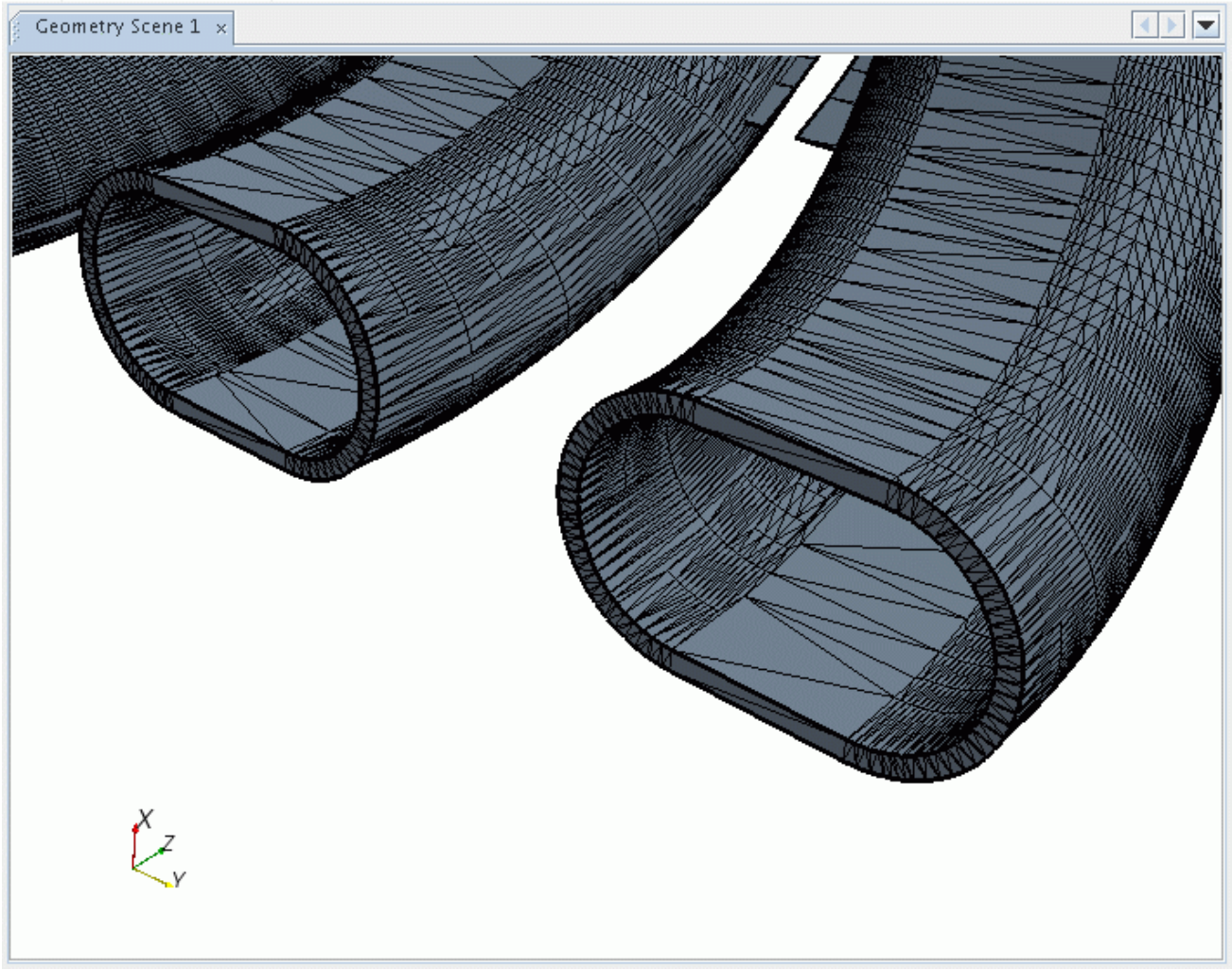
1.

See Also:

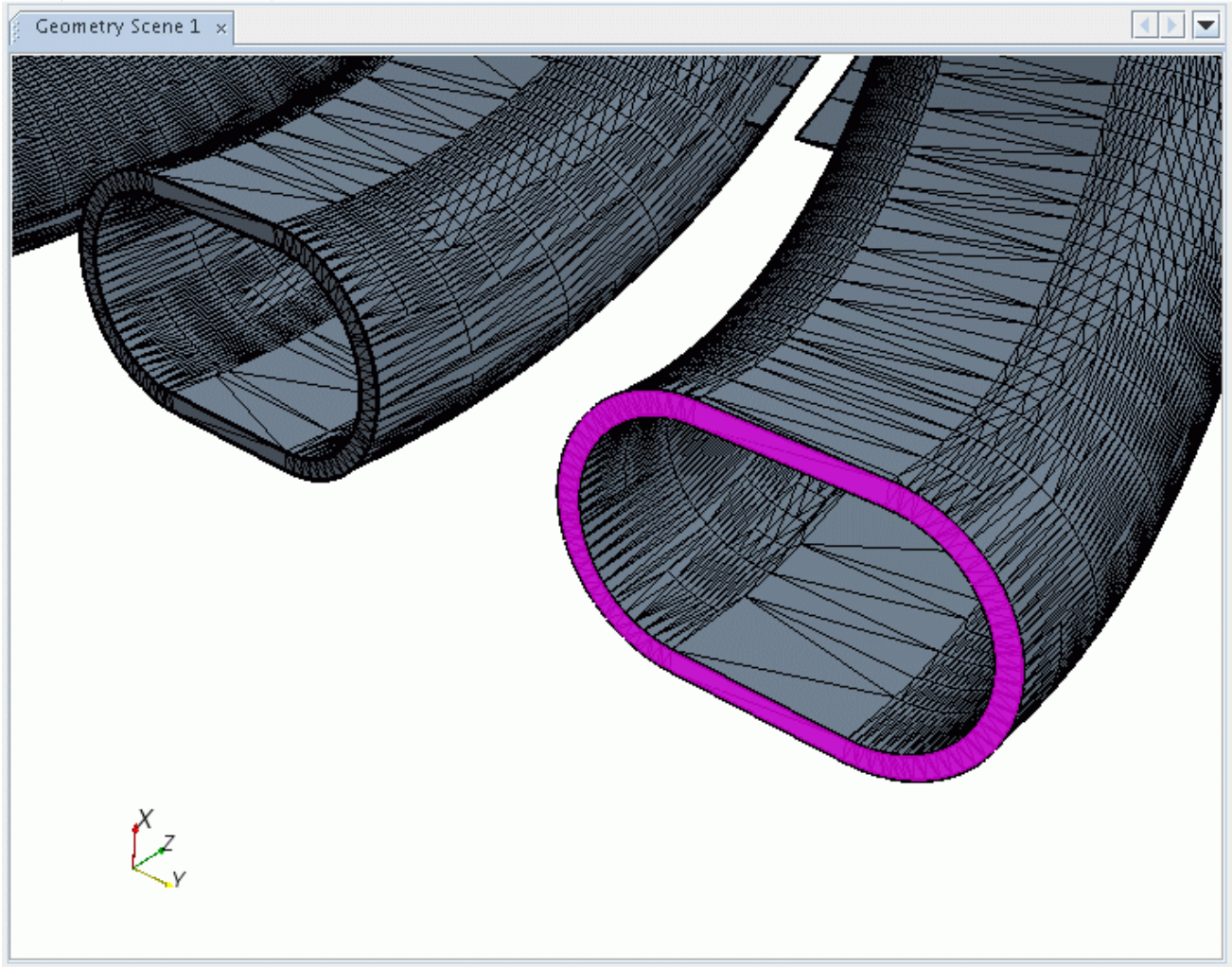
Selecting Entities

Translating A Surface Using The Offset Tool

This example shows the translation of a surface using the **Translate**option. This feature allows you to extrude or modify the surface geometry before surface meshing. In this case, the tool extrudes a manifold outlet pipe, increasing its length.

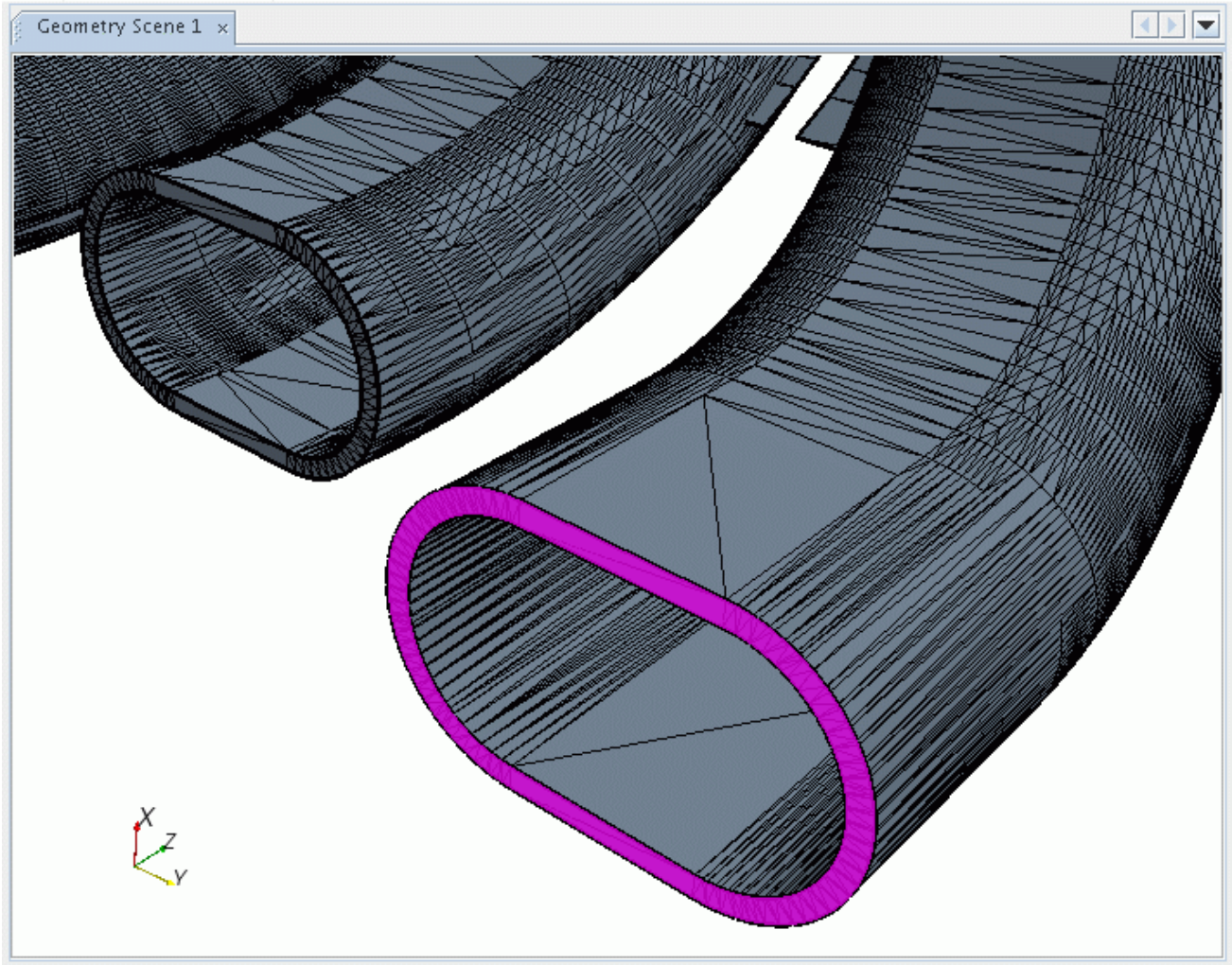


You first select the required surface using the Selection Controls. In this case, the **Select Zone** tool selects the ring of end faces:



You then select the required *Offset Action*, which is **Translate** in this case. The **Direction** is **+ Normal Direction**, although you could also use **Specified Direction** and **Seed Point**. The **Distance** is an arbitrary value, with any allowable units.

To carry out the translation, click **Offset**. The *Graphics* window updates with the modified surface.



The tool also translates any feature curves on the initial surface with the selected faces. The above translation does not create new faces.

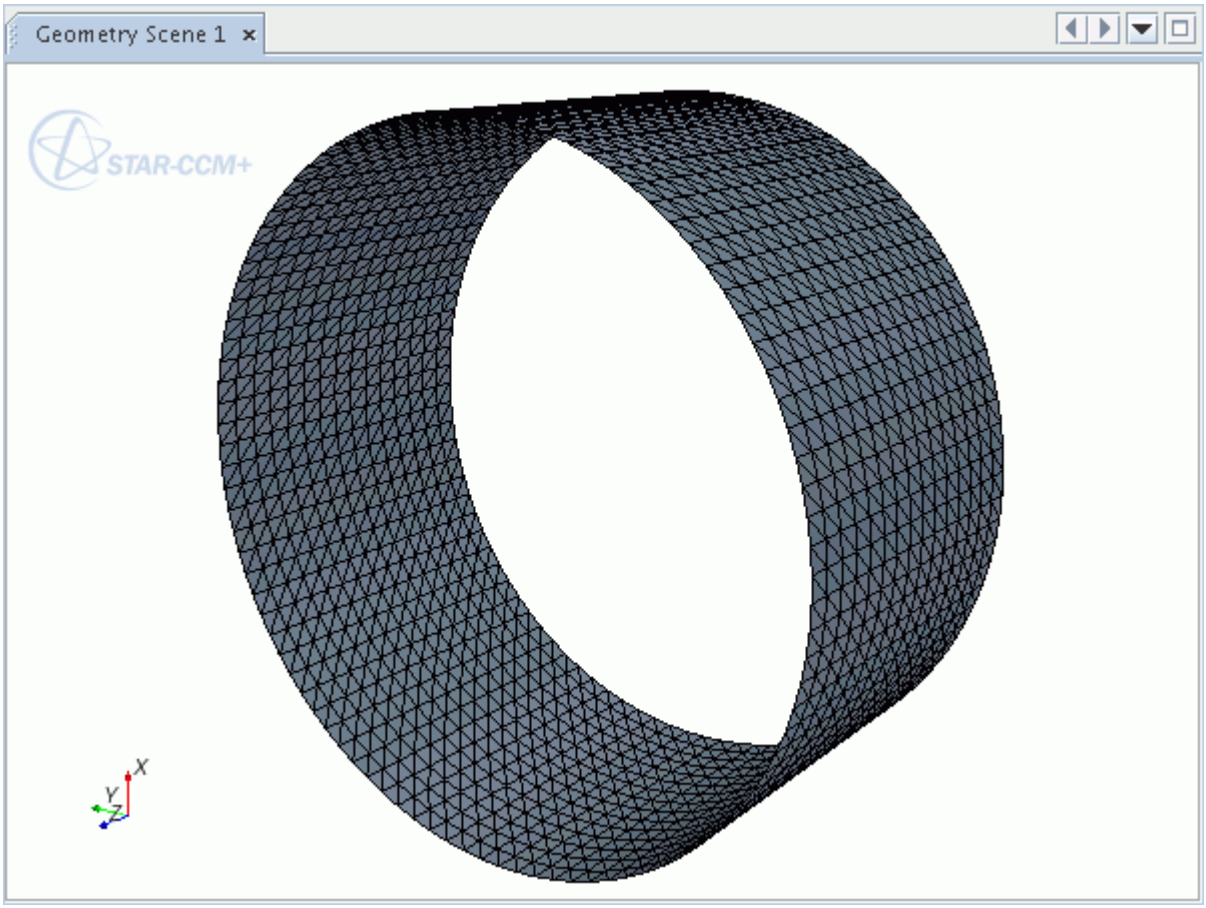
1.

See Also:

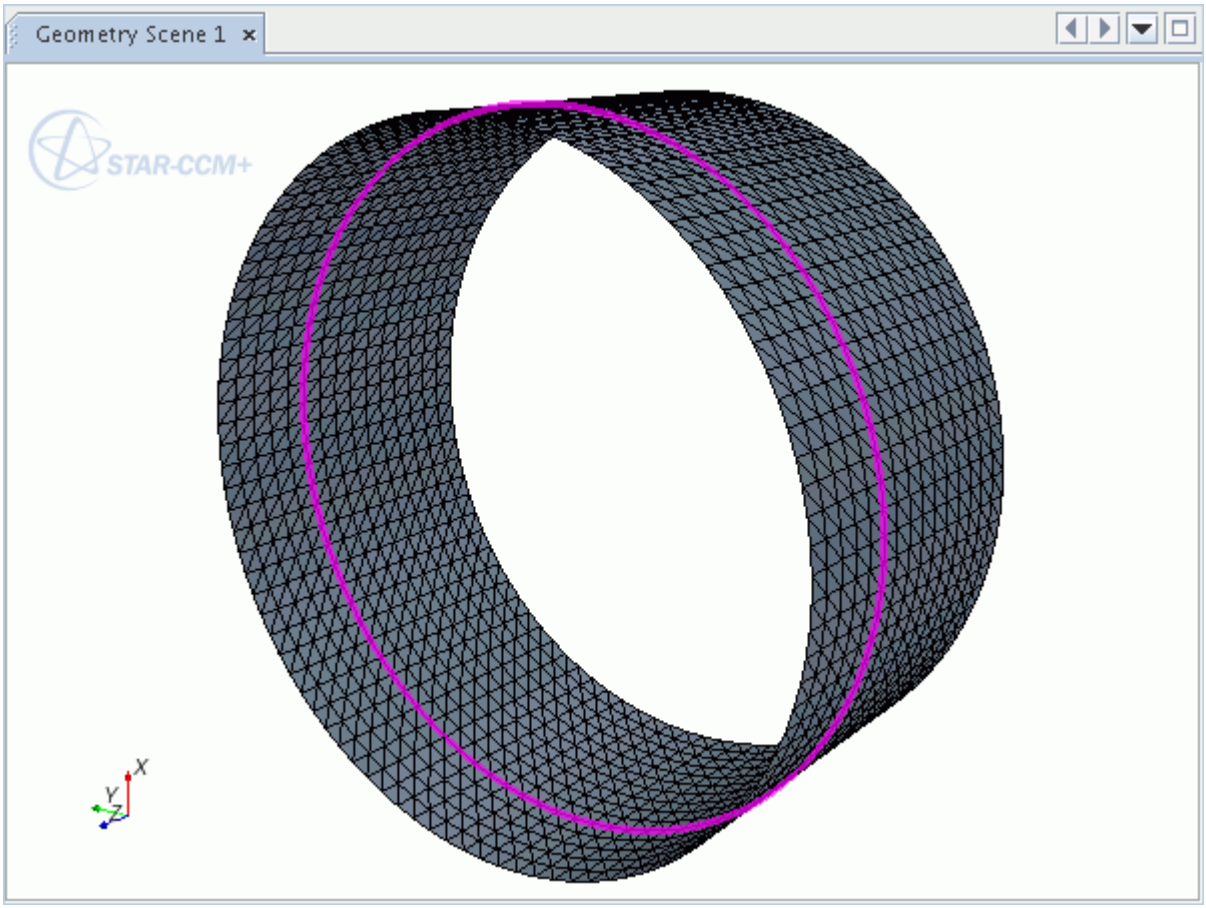
Selecting Entities

Creating A Zero Thickness Baffle Using The Offset Tool

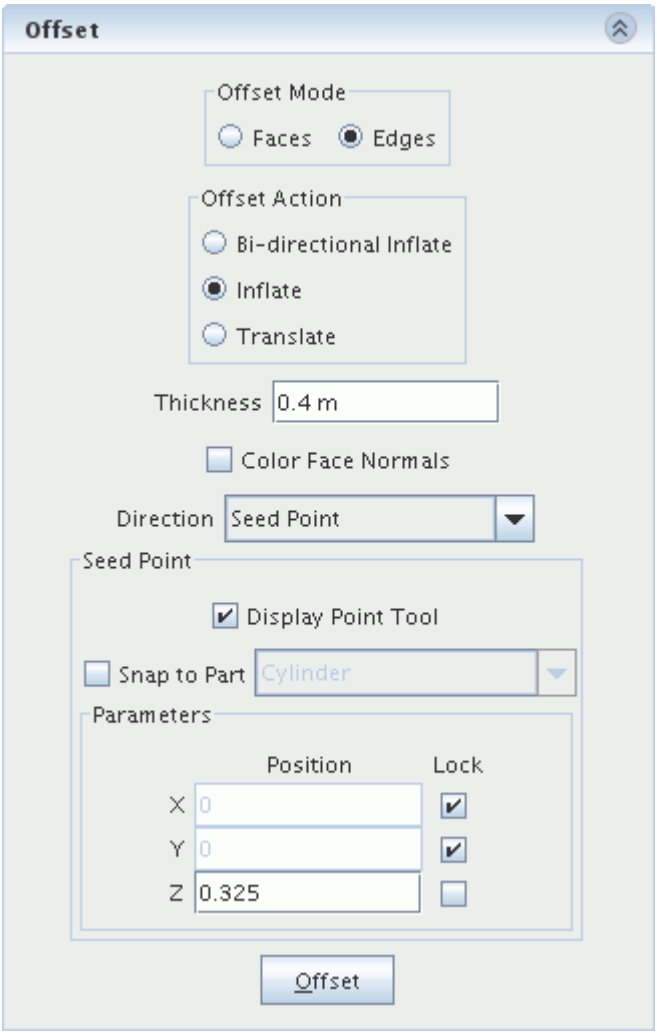
This example shows the inflation of edges using the **Offset Faces/Edges** tool in **Edges** mode. This feature can be used to create a zero-thickness baffle. In this case, a two-dimensional orifice is created in a section of cylindrical pipe.



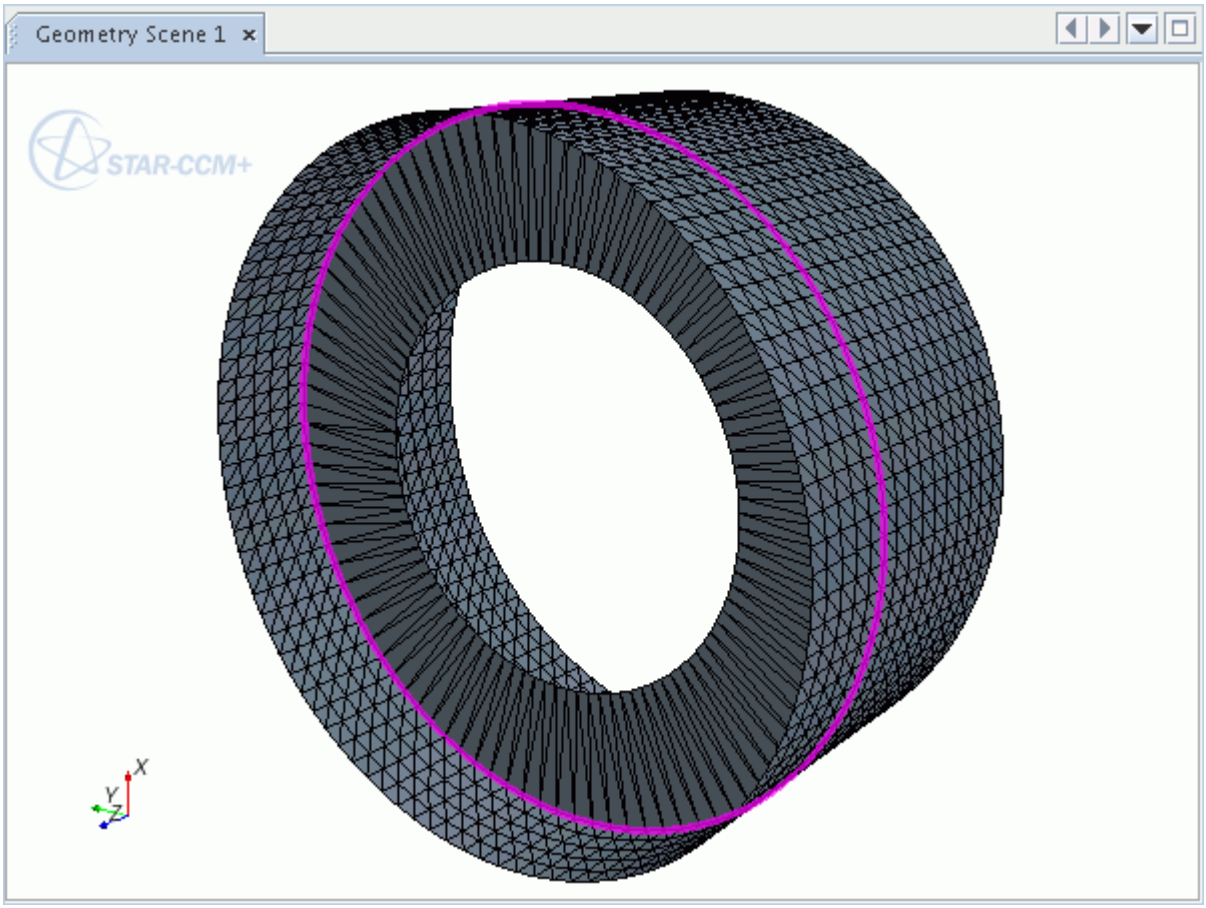
You first select the required edges using the Selection Controls.



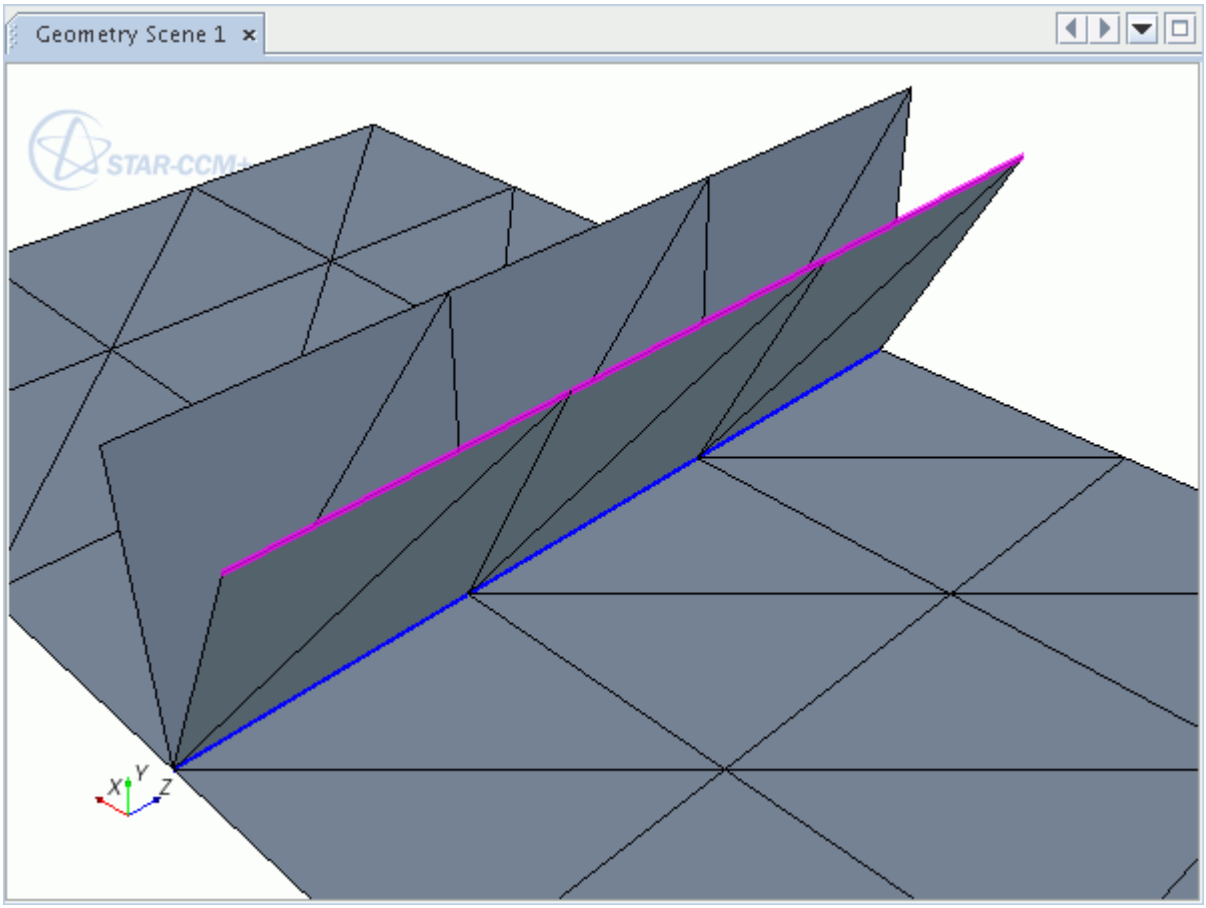
Then, select the required *Offset Mode*, which is **Edges** in this case. The *Offset Action* is set to **Inflation** so that the edge is inflated in only one direction. The **Direction** is set to **Seed Point**, with the point placed in the center of the pipe. These edges are inflated in a normal direction towards the center of the pipe.



To carry out the inflation, click **Offset**. The *Graphics* window updates displaying the zero-thickness baffle.



If a feature curve is selected, the corresponding inflated edge is also a feature curve. Also note that when inflating or translating an edge in a normal direction, the orientation of connected faces determines the direction. This orientation can cause unexpected results when inflating non-manifold edges, as the orientation of three or more edges is considered. This effect is illustrated in the example below. The blue edge is non-manifold and the pink edge is the result of an offset in the positive normal direction.



1.

See Also:

Selecting Entities

Merge and Imprint Tool

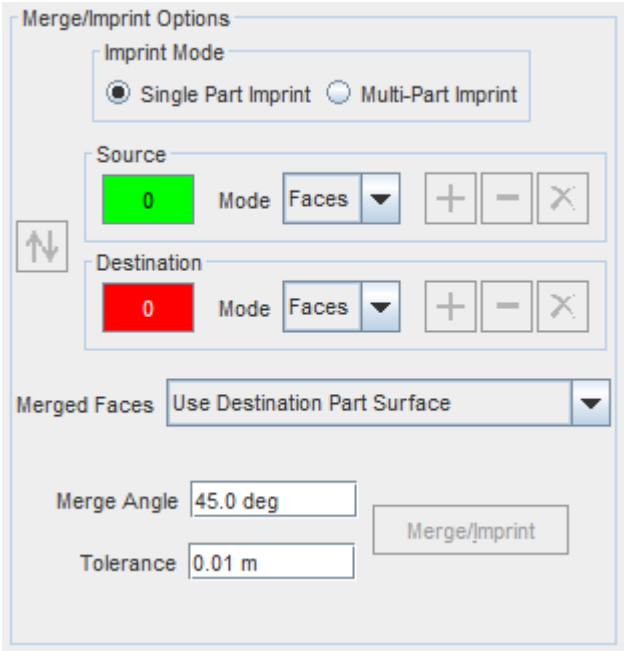
The **Merge/Imprint** tool lets you imprint sets of faces and edges onto each other.

The tool also lets you merge two or more overlapping (co-planar) surfaces or sets of faces. When using **Merge/Imprint** on a single part, the result is either a common shared surface or no surface at all (that is, the common area is removed). If there are multiple parts, the tool creates part contacts for the shared surface.

Similarly, for feature curves belonging to the same parts, the tool can imprint two or more edges onto each other, effectively zipping them together. You can also imprint edges onto faces and imprint faces onto edges, depending on the requirement. No edge options are available when imprinting across multiple parts. In all instances, the faces or edges do not need to belong to a closed surface in order to use the tool.

To display the *Merge/Imprint Options*, click (**Merge/Imprint single or multiple parts**).

Selecting **Merge/Imprint** from the **Mode** menu provides access to the *Merge/Imprint Tool* group box.



The *Merge/Imprint Tool* is applicable to two situations:

- When the surfaces belong to a single part
- When the surfaces belong to multiple parts

The details below are relevant to both situations. Review this information before using the tool for a specific example.

Merging and Imprinting Surfaces

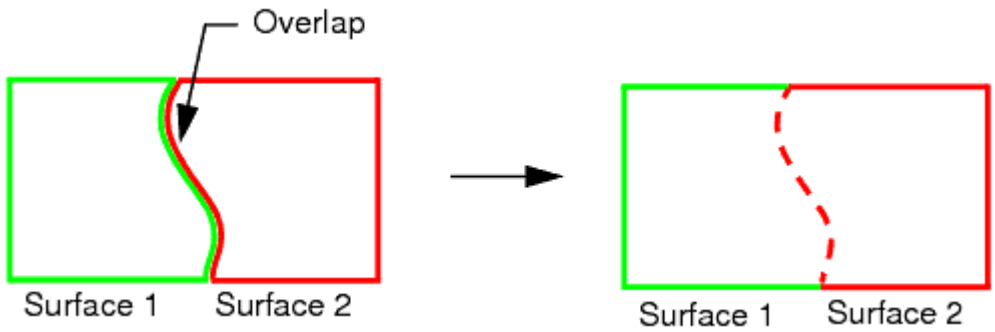
You add the required surfaces to *Source* and *Destination* groups, so that the projection scheme used for merging and imprinting knows the direction in which to travel.

Note: If there are no vertices within the imprint tolerance that can project from one surface to the other, the imprint fails. To overcome this limitation, you can add vertices to the surface with the remesh or split faces tools.

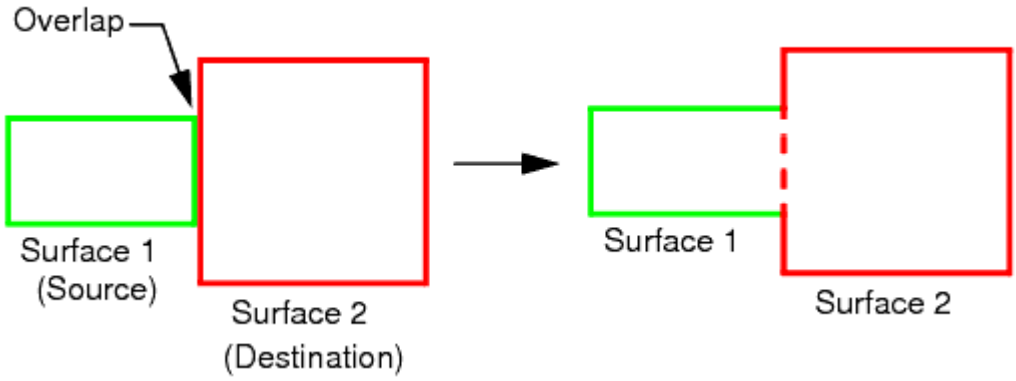
In general, two scenarios exist when the merging and imprinting of overlapping surfaces is required:

- The perimeter or area of the overlap is the same
- The perimeter or area of the overlap is different

The following diagram illustrates an example where the overlapping boundaries have the same perimeter or area:



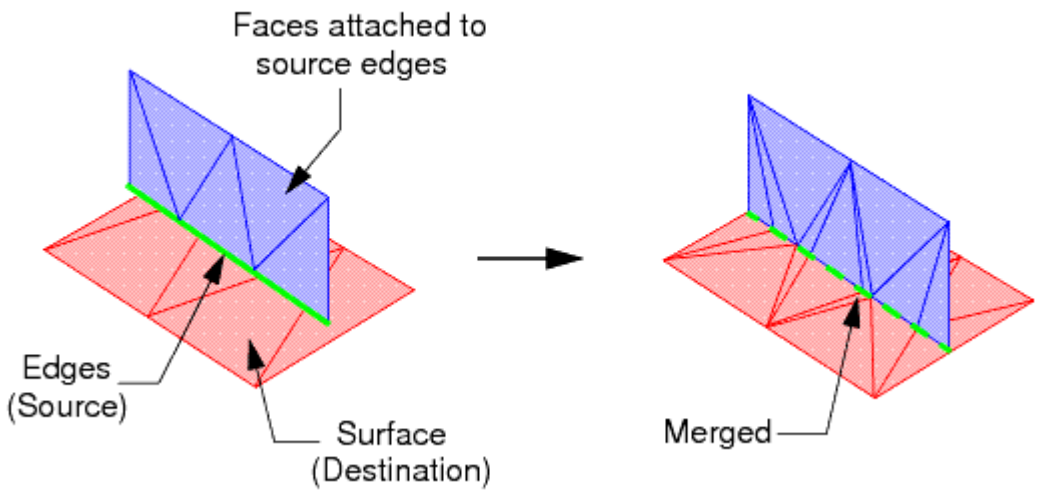
For cases where the two surfaces have a different perimeter or area, it is possible to select either surface as the source. The following diagram illustrates an example where the smaller surface is the source.



Imprinting Edges to Faces

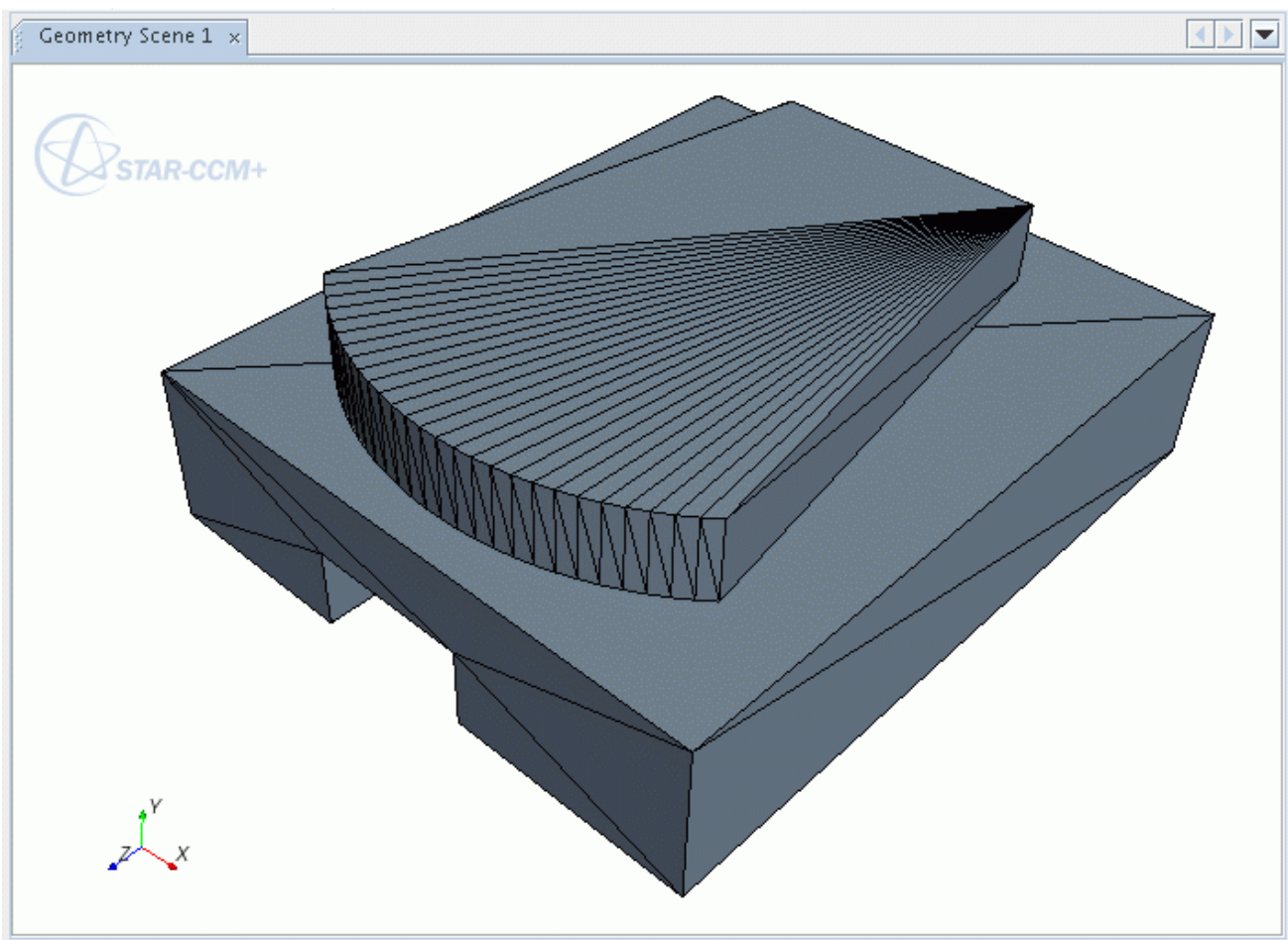
When imprinting edges onto faces for single parts, you use the same approach as described above. In this instance, the edges are typically the source, and the destination is one or more faces that are imprinted with the shape of the edge. Details of a procedure to use with the **Multi-Part Imprint** mode are given in Imprinting Baffle Edges onto Faces.

A typical example of imprinting edges onto a surface is a baffle surface perpendicular to an exterior surface. You can imprint one or more edges along the side of the baffle to the exterior surface, thus letting you convert the baffle to an interface for meshing purposes.

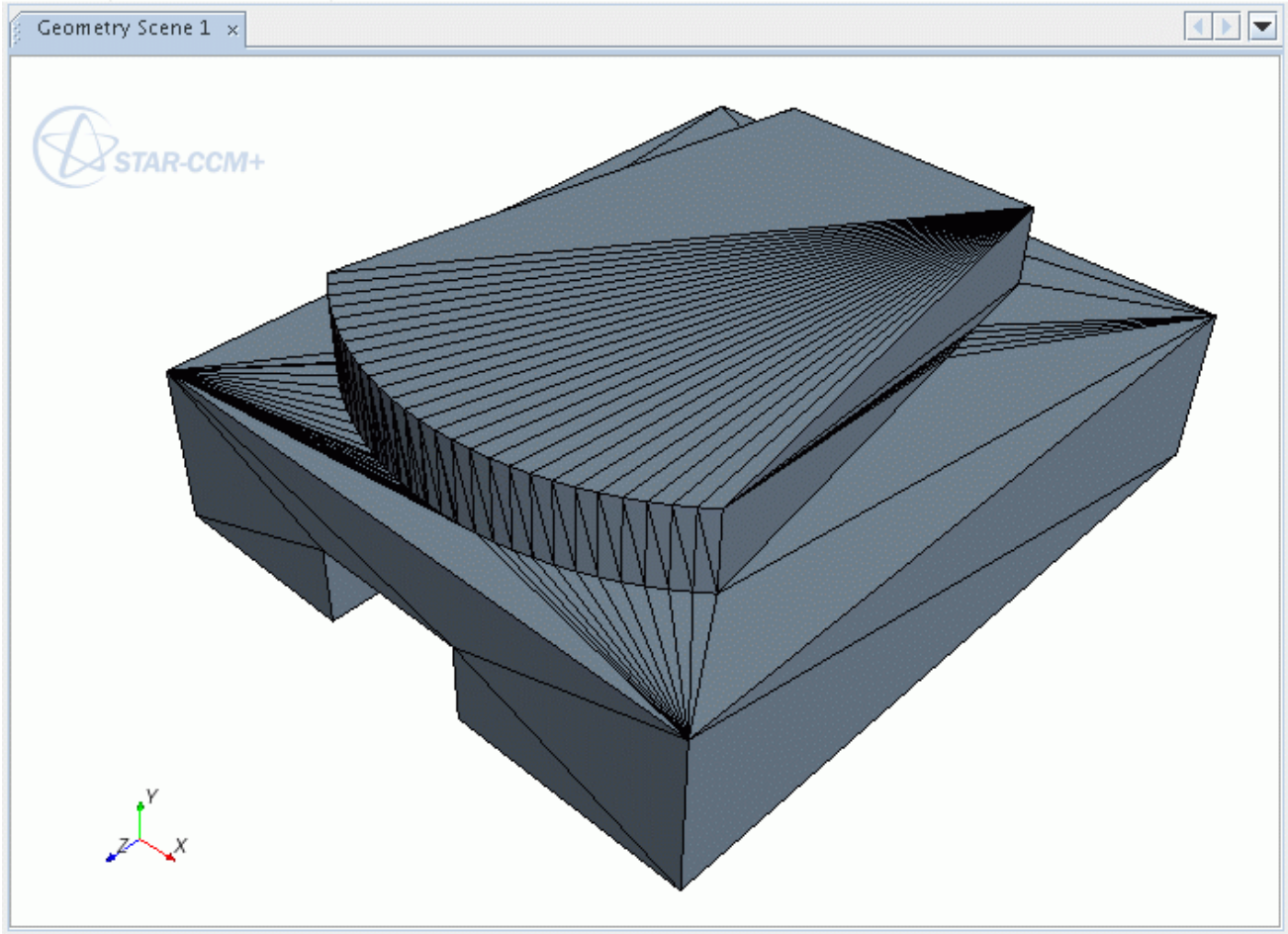


Example of Merging and Imprinting

The following example merges and imprints two closed surfaces together. Shown below are two closed bodies that are touching, but not merged.



As shown below, the result of the **Merge/Imprint** was to retriangulate both bodies, so that there is a one to one correspondence at the interface.



If both bodies belong to the same part, the tool provides the choice of keeping or deleting the shared surface. If both bodies belong to two different parts, the tool creates a part contact between the two surfaces.

Considerations When Using the Merge/Imprint Tool

The following conditions apply when using the **Merge/Imprint** tool for both single and multiple parts:

- The surfaces do not need to be unique to the overlapping area. For example, a closed body, consisting of one surface, can merge with the overlapping surface of another body.
- When merging faces with other faces, the tool does not generate feature curves. If feature curves are required, you create them manually.
- The surface faces and edges for single parts do not need to belong to a closed and manifold part. In other words, the tool allows use of planar and open surfaces.
- The quality of the original surface tessellation is not critical.
- You can use the surface remesher after merging to improve the quality of the resultant surface.

Contents:

Imprinting A Single Part
Imprinting Multiple Parts
Merge And Imprint Advanced Parameters

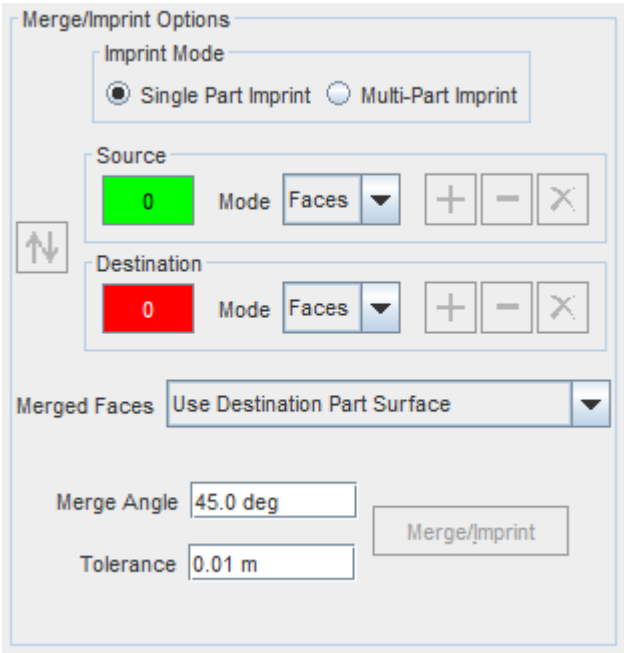
See Also:

Changing the Parts in the Active Model
Surface Remesher
Converting Boundaries to Interfaces
Imprinting Baffle Edges Onto Faces
Using Part Contact Data to Create Interfaces

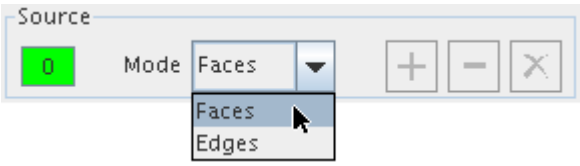
Imprinting A Single Part

This options allows you to merge two or more overlapping (co-planar) surfaces from the same part.

The following image shows the **Merge/Imprint** tool in **Single Part Imprint** mode.



1. To use the **Merge/Imprint** tool in **Single Part Imprint** mode, first select **Faces** or **Edges** from the **Mode** menu for both the *Source* and *Destination*.
   The tool allows any combination of face and edge imprinting. You can select the source and destination faces and edges using the Selection Controls.



2. With the selections made, click + for either the source or destination to add the faces or edges to the group.

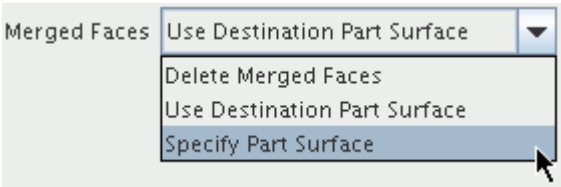
   In the *Graphics* window, faces or edges added to the **Source** appear green, and the faces or edges added to the **Destination** appear red.

3. You can remove faces or edges by selecting them in the display, and clicking − for the source or destination.

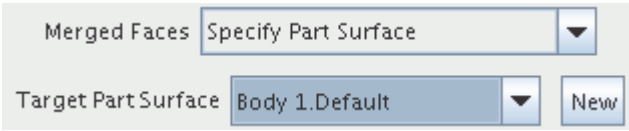
4. To remove all faces from the source or destination, click ✕.

5. If you select the opposite group of faces for source and destination, you can swap them by clicking ⇅.

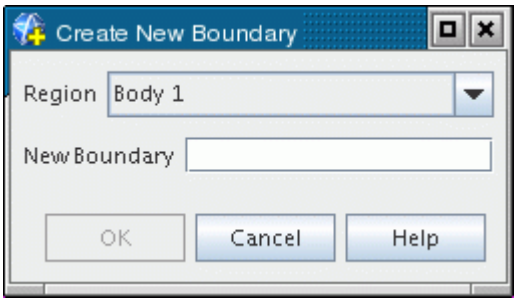
When merging/imprinting faces to other faces, the common surface belongs to the destination surface by default. Options exist to delete the common surface (that is, the two bodies merge as one), or assign the common surface to a specified part surface.



Selecting **Specify Part Surface** activates the **Target Part Surface** option, allowing selection of existing part surfaces:



Clicking **New** creates new surface using the *Create New Surface* dialog:



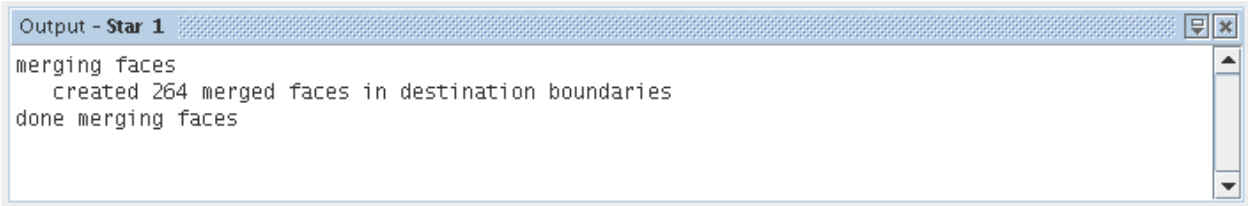
6. Enter the **New Surface** name and click **OK** to define the new surface, which appears in the simulation tree.

   Merge Angle limits the operation to faces that are at a user-defined angle to each another. For example, the default angle, `45 deg`, means that the tool merges any source and destination faces that are less than 45 degrees to each another. This feature is useful when a curved surface touches a planar surface, such as a tire contact patch on a road surface. This option is only available when merging faces to other faces.

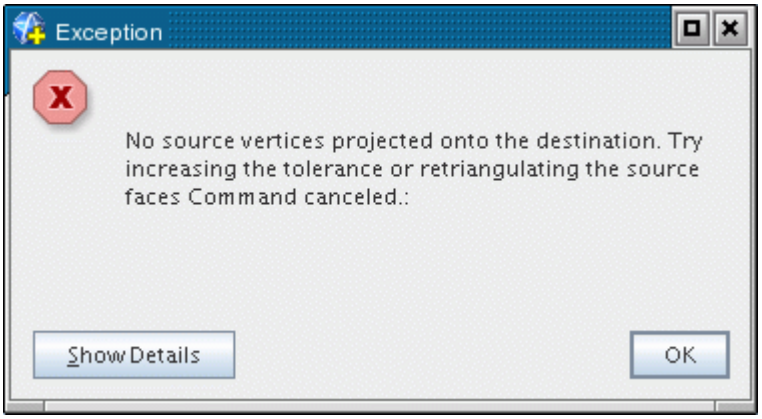
   Tolerance limits the operation to faces and edges that are a user-defined distance from one another. The tool calculates the **Tolerance** value when merging edges to other edges (hence, it is not available).

7. Click **Merge/Imprint** to carry out the operation.

   When complete, the *Output* window shows feedback from the operation.



   In situations where the merge was unsuccessful, the following error message appears.

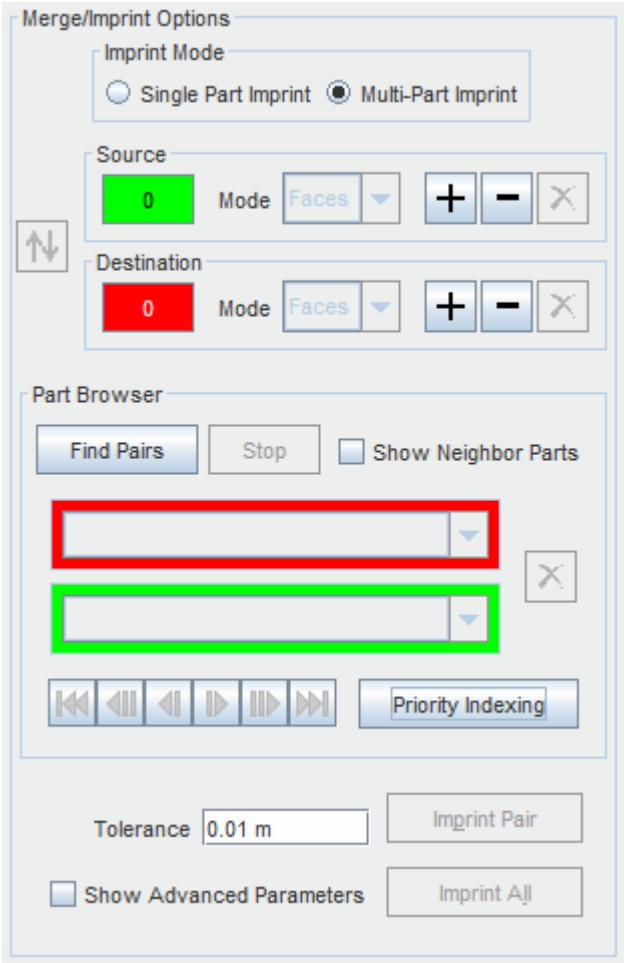


   In this instance, try reversing the selections that you made for the source and destination faces or edges. In cases where a relatively large distance separates the faces or edges, having finer triangulation for the items improves the merge, as increasing the tolerance value.

Imprinting Multiple Parts

This options allows you to merge two or more overlapping (co-planar) surfaces from multiple parts.

The following image shows the **Merge/Imprint** tool in **Multi-Part Imprint** mode.



The multi-part imprint mode differs from the single part mode in that the tool calculates source and destination pairs automatically. These calculated pairs appear in the *Part Browser*.

1. To use the **Merge/Imprint** tool in **Multi-Part Imprint** mode, first review and set the **Tolerance** value and any advanced parameters that you require.
   To access and edit advanced parameters, activate **Show Advanced Parameters**. These values affect how the tool finds pairs. For more details, see the single part imprint section.
2. Click **Find Pairs**.
   All relevant part pairs that are based on the supplied values are found.
   a) If you enter an unsuitable *Tolerance* value resulting in an excessively long search, click **Stop** to stop the process and enter a new *Tolerance* value.
   b) If the tool does not find all pairs, then change the *Merge Angle* (activate *Show Advanced Parameters*) or *Tolerance,* and click **Find Pairs** again to repeat the process.

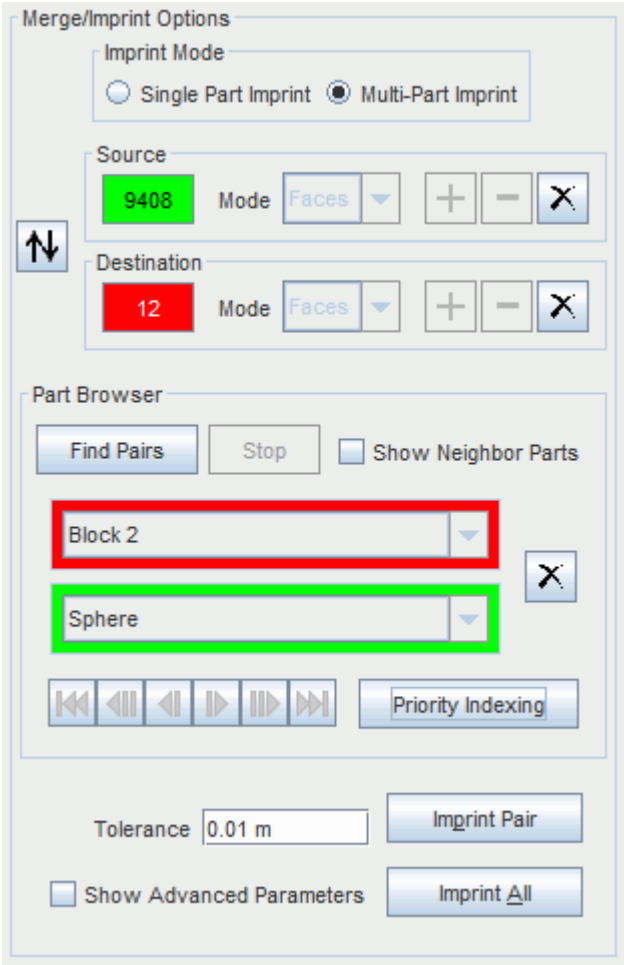
Once **Find Pairs** is complete, the *Output* window shows a summary of all pairs that are found on a per-part basis. The *Output* window displays the text `MERGED` next to a previously merged pair. You can merge and imprint pairs of parts more than once by selecting source and destination faces manually, but the automatic find does not return previously merged pairs of parts

```
Output - Star 1
region=Body 1 assigned id=1
region=Body 2 assigned id=2
region=Body 3 assigned id=3
Finding close surfaces
   tolerance = 0.01
   populating faces per part
      found 3 parts
   finding close parts
      found 2 part pairs
   done, CPU Time: 0.00, Wall Time: 0.03, Memory: 337.96 MB
 finding close pairs for region=Body 1(id=1):  Body 2(id=2), Body 3(id=3),
 finding close pairs for region=Body 2(id=2):  Body 1(id=1),
 finding close pairs for region=Body 3(id=3):  Body 1(id=1),
checking interfaces
done checking interfaces
destination region: Body 1 <->  source region:  Body 2
```

If multiple pairs are present, the **Browse Pairs** selector becomes active, allowing you to display each part pair. Clicking the forward or backward arrows cycles through each pair and displays them in the *Graphics* window.

The **Show Neighbor Parts** option displays or hides neighboring parts in the scene once part pairs are found. See Displaying Neighboring Parts.

The image below shows the **Merge/Imprint** tool after a completed search.



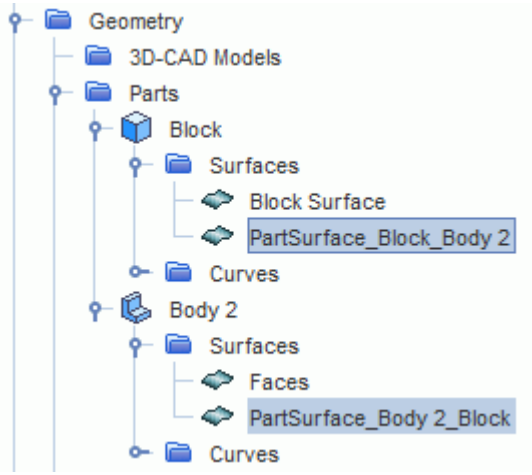
Each part pair appears green and red in the *Graphics* window, to indicate the source and destination surface respectively. Previously imprinted pairs remain in the browser, but are inactive (that is, they do not appear green or red) for the current *Surface Preparation* session.

The **Destination** and **Source** drop-down menus become available in the *Part Browser* after the search process is complete. The first drop-down menu is the **Destination** menu which contains all parts that form a pair with other parts. The second drop-down menu is the **Source** menu which contains all parts that form a pair with the selected **Destination**part.

3. If you wish to reverse the source and destination, click to swap them.
   You can also use the **Priority Indexing** button to define which part is the source and which part is the destination by comparing the **Priority Index** between the part pair. This option is useful for simulations with many parts.

4. To remove pairs from the *Part Browser*, use the menus to select the appropriate part pair, and click to remove them for the current *Surface Preparation* session.
5. When you have identified all part pairs, click **Imprint All** to imprint all part pairs at once. Alternatively, click **Imprint Pair** to imprint the part pair that is selected as the **Destination** and **Source** in the *Part Browser*.

When the default advanced parameters are maintained, the tool generates a new part contact with a conformal mesh for each imprinted part, which represents the common surface between the two parts. You can review the new surfaces in the *simulation* tab.



Contents:

Imprinting Pairs More Than Once
Editing Part Priorities
Imprinting Baffle Edges Onto Faces
Using The Exact Face Match Method
Using The Conformal Resulting Mesh Type Option
Using The Non Conformal Resulting Mesh Type
Using The Boolean Subtract Preprocess Option
Using The Opposing Outward Face Orientation Option
Displaying Neighboring Parts

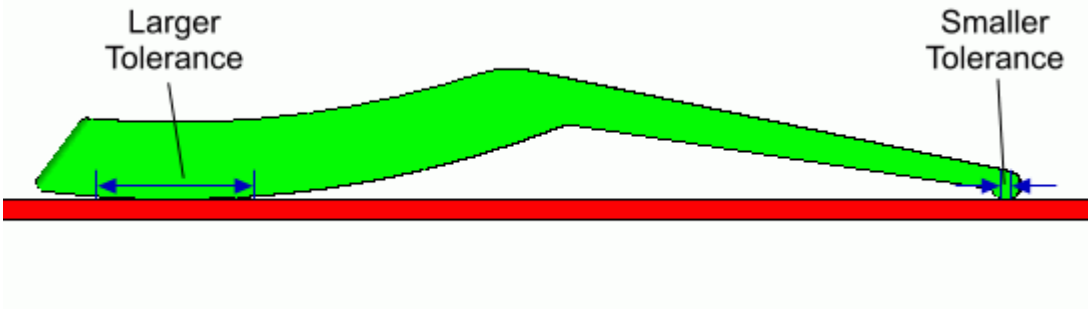
See Also:

Using Part Contact Data to Create Interfaces
Imprinting A Single Part
Merge And Imprint Advanced Parameters

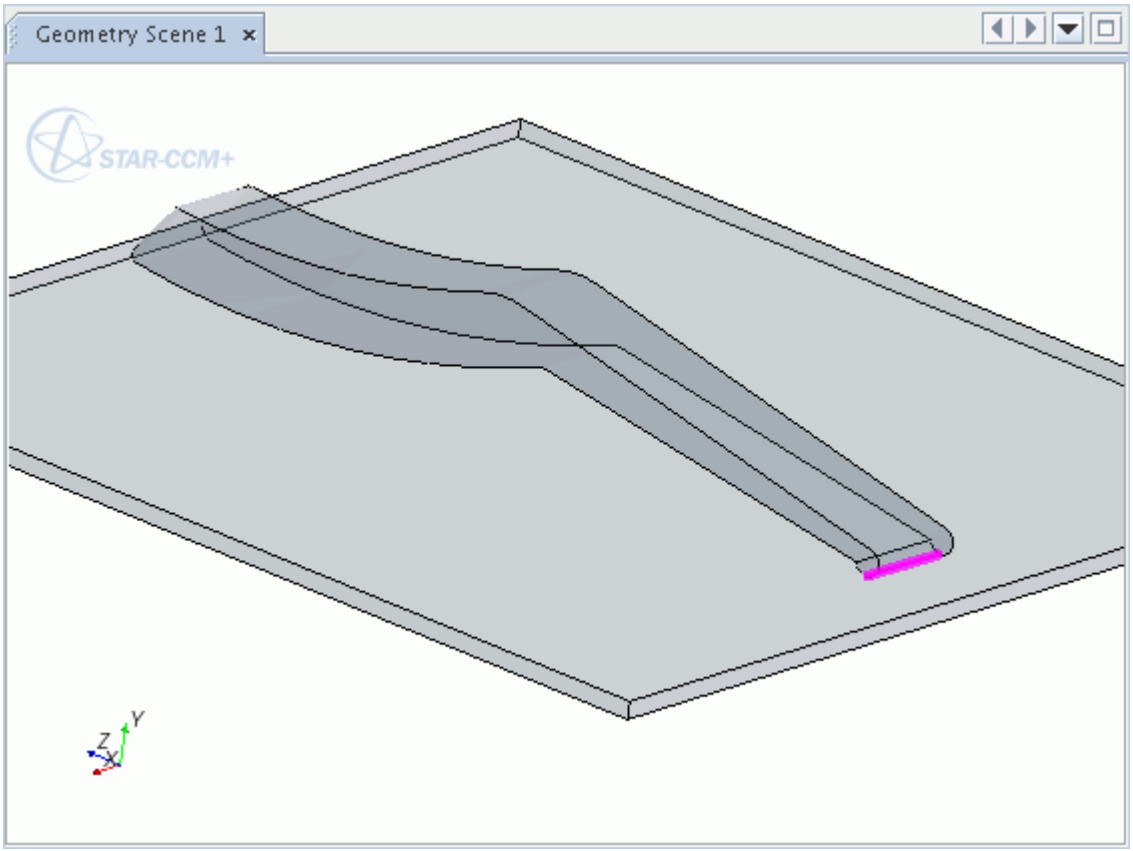
Imprinting Pairs More Than Once

In some circumstances, it is necessary to imprint a pair of parts more than once in order to apply different tolerance values to each imprint.

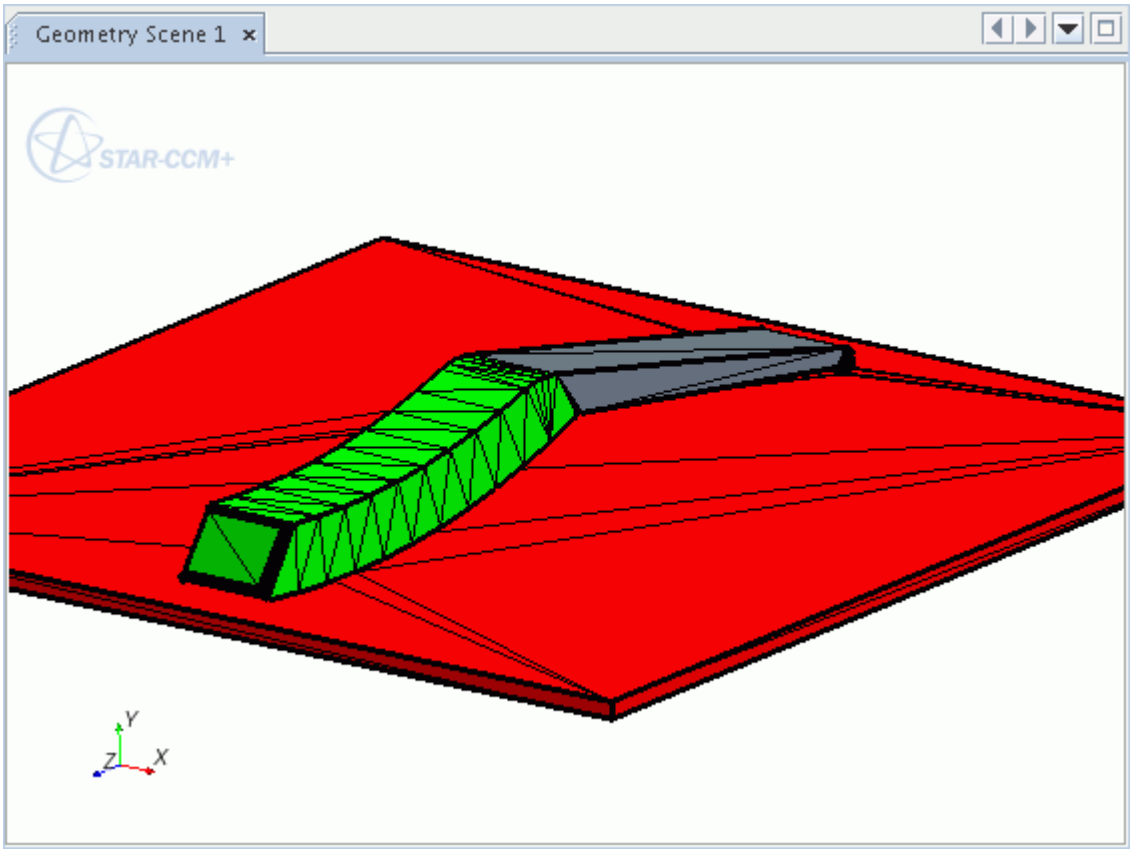
For example, in the geometry that is shown below, a large tolerance value that is applied to both ends of the top part results in the small cylinder being distorted after the imprint. Different imprint tolerances are required for each end of the top part.



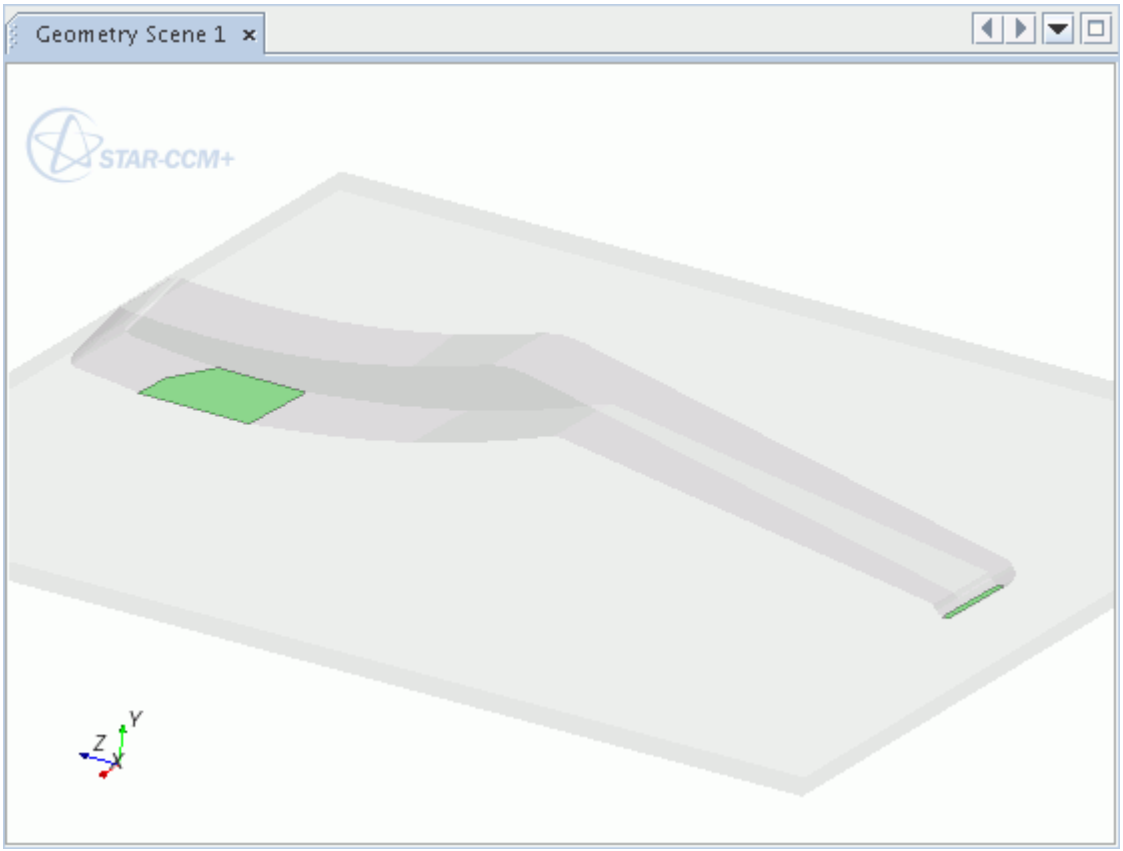
To avoid a problem with the small end of the top part, set the *Tolerance* value to a small value initially for the automatic find and the first imprint. This setting results in a match as shown below.



Perform a second imprint with a larger *Tolerance* value. This time, however, the source and destination faces must be selected manually using the *Selection Controls*. Take care to avoid selecting faces around the site of the first imprint, as the larger tolerance value could affect the selection. A selection such as shown below would be acceptable for this example.



After the imprint operation, both ends of the top part will be connected to the base with appropriate matching areas. These interfaces are shaded green in the image below.



The **Imprint All** option only works for parts that have not yet been imprinted. Successive imprints such as that demonstrated here must be determined manually.

Editing Part Priorities

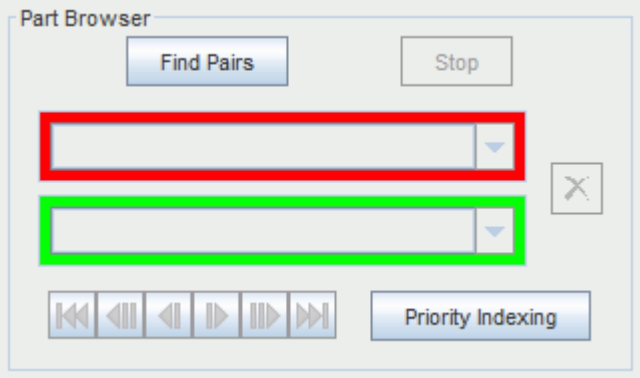
The **Edit Priorities** function defines the destination and source for each part pair by comparing the priority indices between part sets.

The part with the higher priority index becomes the destination (red), and the part with the lower priority becomes the source (green). This feature is useful when using merge/imprint on many parts, as it does not require manual assessment of each part pair.

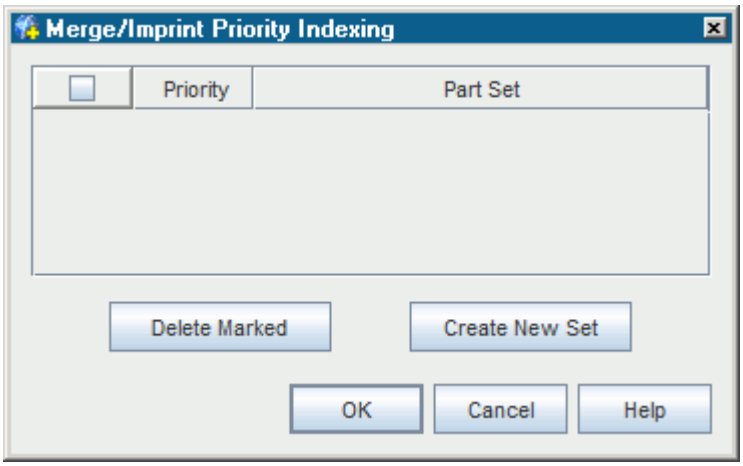
Note the following considerations when editing part priorities:

- As **Priority Indexing** requires multiple parts, the feature is only available when in **Multi-Part Imprint** mode.
- Parts cannot belong to multiple part sets. Adding a part to a new part set removes it from the original part set.
- If you make a mistake when setting priorities, you can use swap to swap the destination and source part.
- Priority values must be greater than or equal to 1.

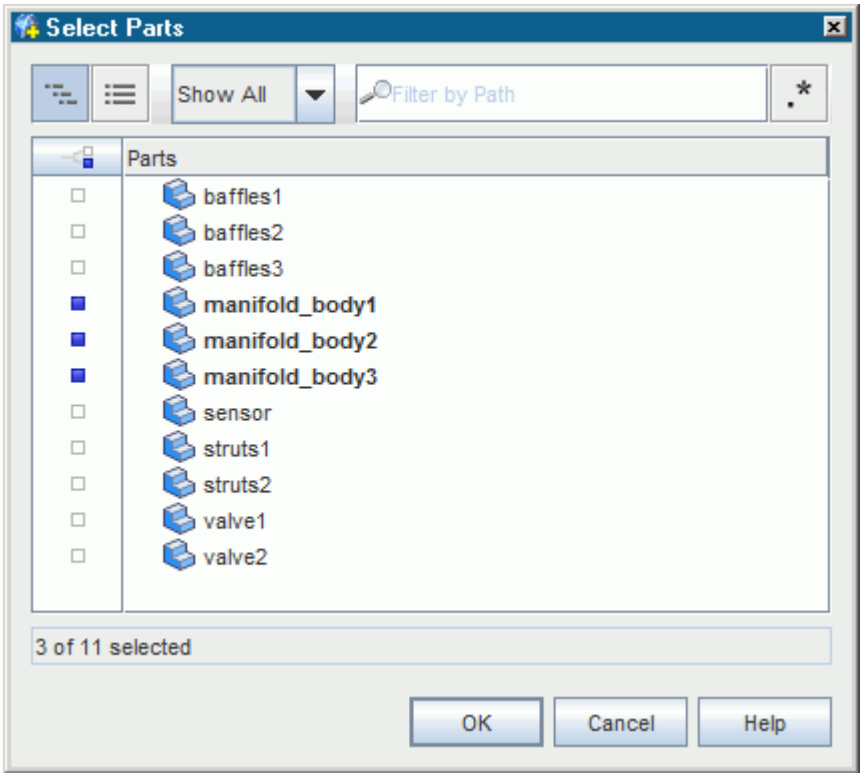
To edit part priorities, first click **Priority Indexing** within the *Part Browser* group box.



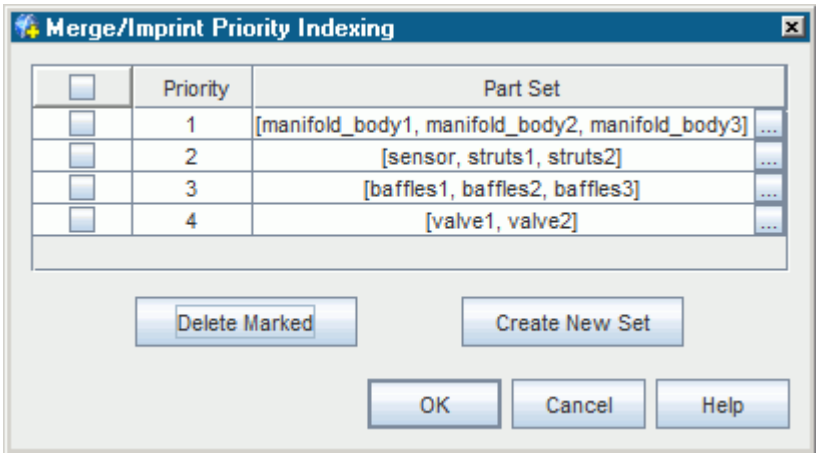
The *Merge/Imprint Priority Indexing* dialog appears. To create a **Priority** for a **Part Set**, click **Create new Set**, which adds a row to the table.



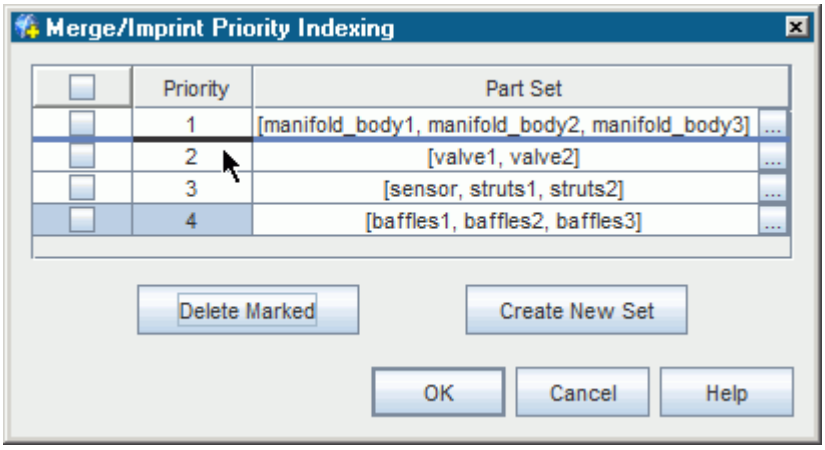
To add a **Part Set**, click the ellipsis (**...**) next to the row, which opens the Select Parts dialog. Select the parts that you want to add to the set and click **OK**.



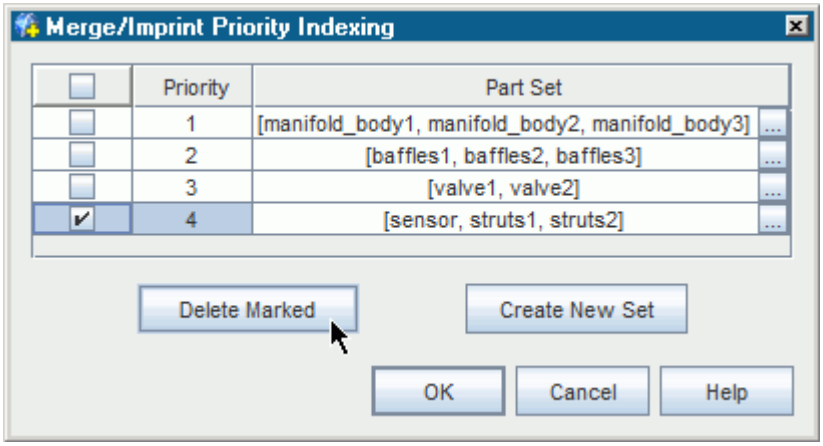
Continue creating sets and adding parts as required.



To change the priority of a **Part Set** you can drag and drop the row higher or lower in the table, or change the value in the **Priority** column.



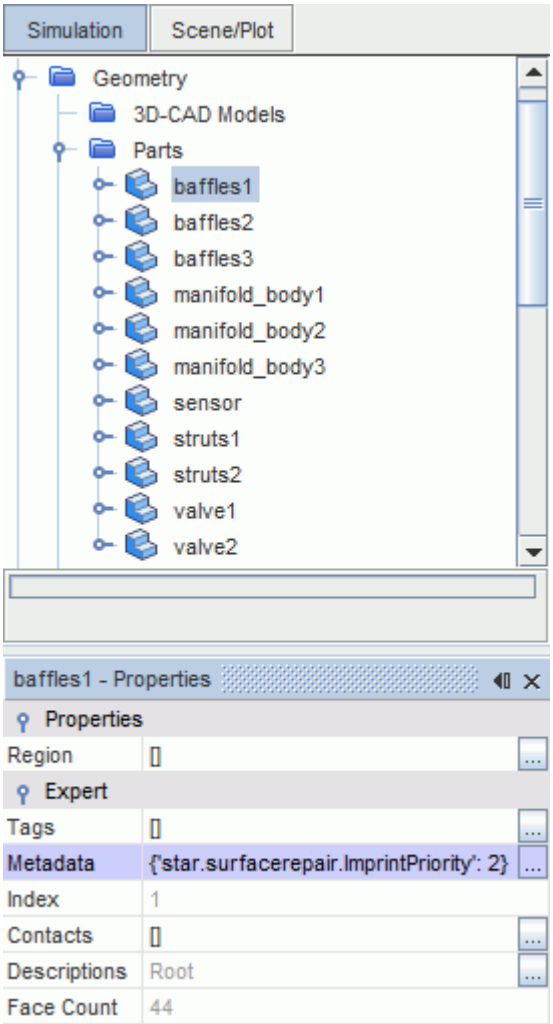
You can also delete **Part Sets** by ticking the appropriate checkbox and clicking **Delete Marked**.



Click **OK** to confirm the priority indexing. Browsing part pairs now define the part with the higher priority as the destination part.

Editing Part Priorities in the Simulation Tree

Editing part priorities using the *Merge/Imprint* tool adds a *Metadata* tag to the part properties.



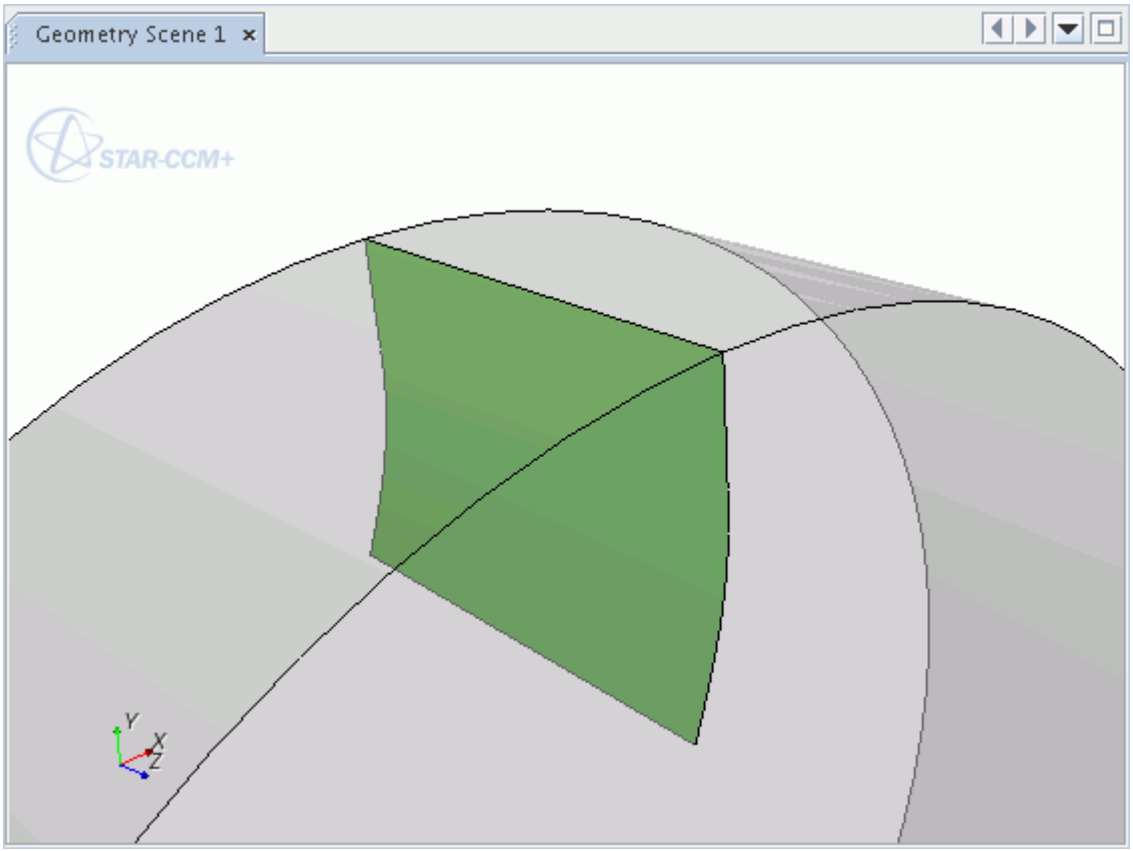
When you are not using the *Surface Preparation tool*, you can manually edit the priority index of the part by changing the integer value in the *Metadata* property.

Imprinting Baffle Edges Onto Faces

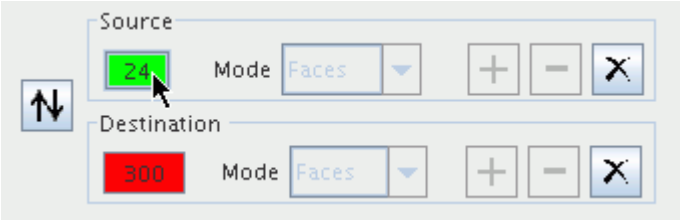
It is possible to imprint the edges of a zero-thickness baffle part onto its adjacent faces while retaining the baffle as a separate part.

The zero-thickness can then be retained during meshing. This section describes a procedure for doing this imprint.

Take the following geometry as an example. There is a single zero-thickness baffle that is contained within a solid volume. The edges of this baffle are imprinted onto the cylinder so that the baffle can then be meshed as a conformal interface.

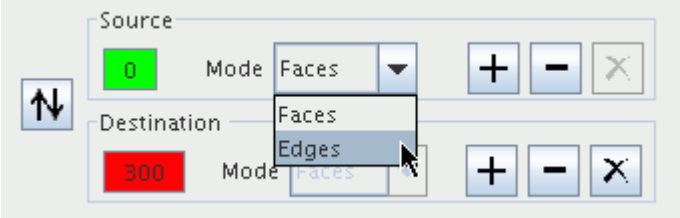


1. To begin this procedure, launch the *Surface Preparation tool* and enter the **Merge/Imprint** mode. Within the *Merge/Imprint Tool* panel, click the radio button for the **Multi-Part Imprint** mode.
2. Click the **Find Pairs** button within the *Parts Browser*. For this example, the source is initially discovered as the outer cylinder, and the destination as the baffle.
3. Click (**Swap**) to reverse this order, so that the baffle faces are set to be the source.
4. To select the faces on the baffle, click the green box within the *Source* group box.

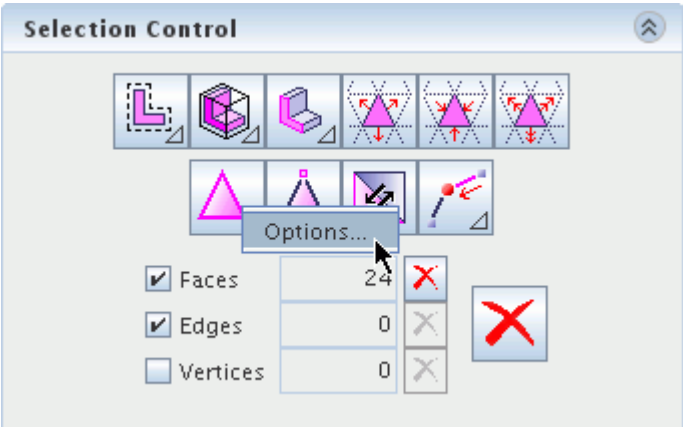


5. Now deselect the source set by clicking the button.

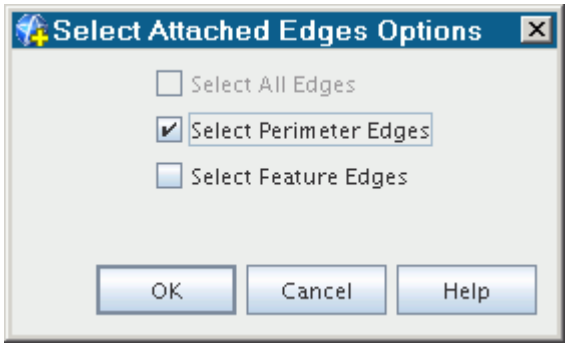
   The selection is not removed from the *Graphics* window: only the faces from the source set on the *Merge/Imprint Tool* are removed.
6. Within the *Source* group box, change the **Mode** to **Edges**.



7. Now move to the *Selection Controls* panel within the *Surface Preparation tool*, right-click (**Select Attached Edges**) and choose **Options...**



8. In the *Select Attached Edges Options* dialog, clear **Select All Edges** and tick **Select Perimeter Edges**.
9. Click **OK** to close the dialog.



10. Click (**Select Attached Edges**).

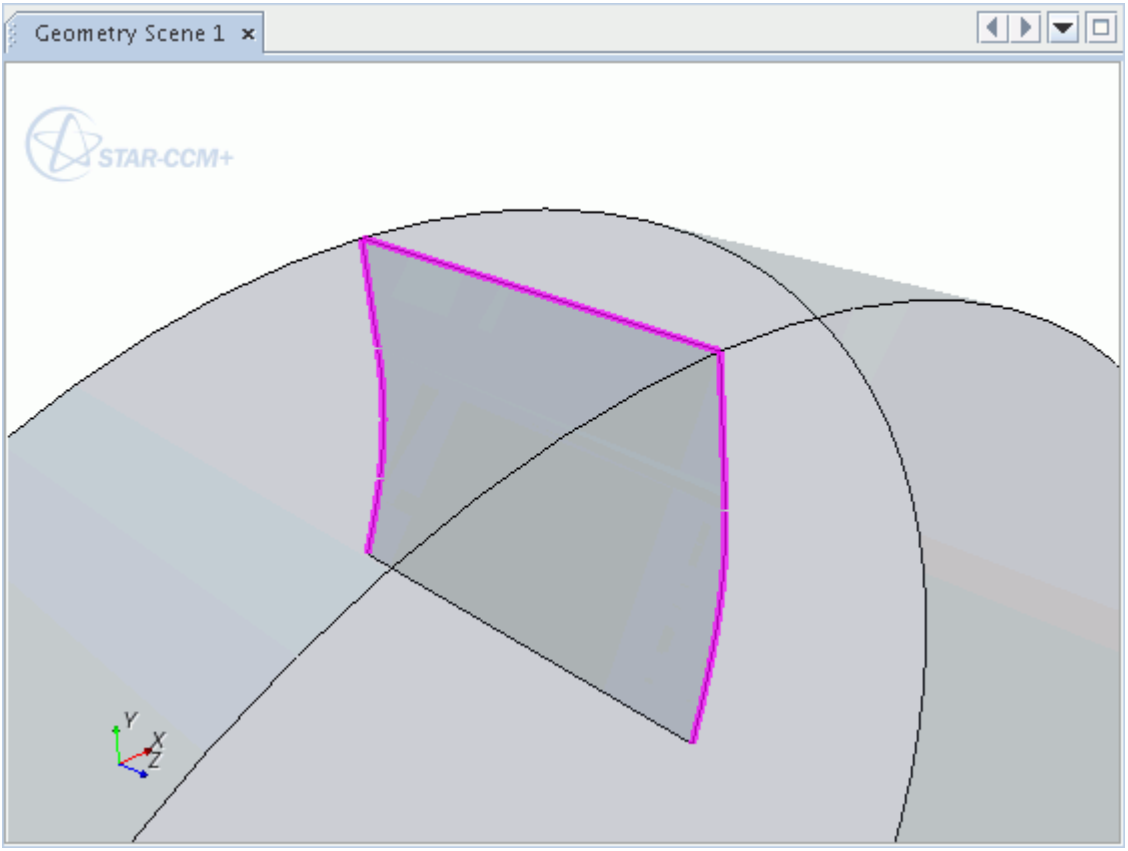
    The edges on the outside of the baffle surface are selected.

11. Click the button within the *Source* group box.

    The newly selected edges are added to the source set.

12. The source edges and destination faces are now fully set. To perform the imprint operation, click the **Imprint Pair** button.

    The imprinted edges are highlighted in the *Graphics* window on completion.



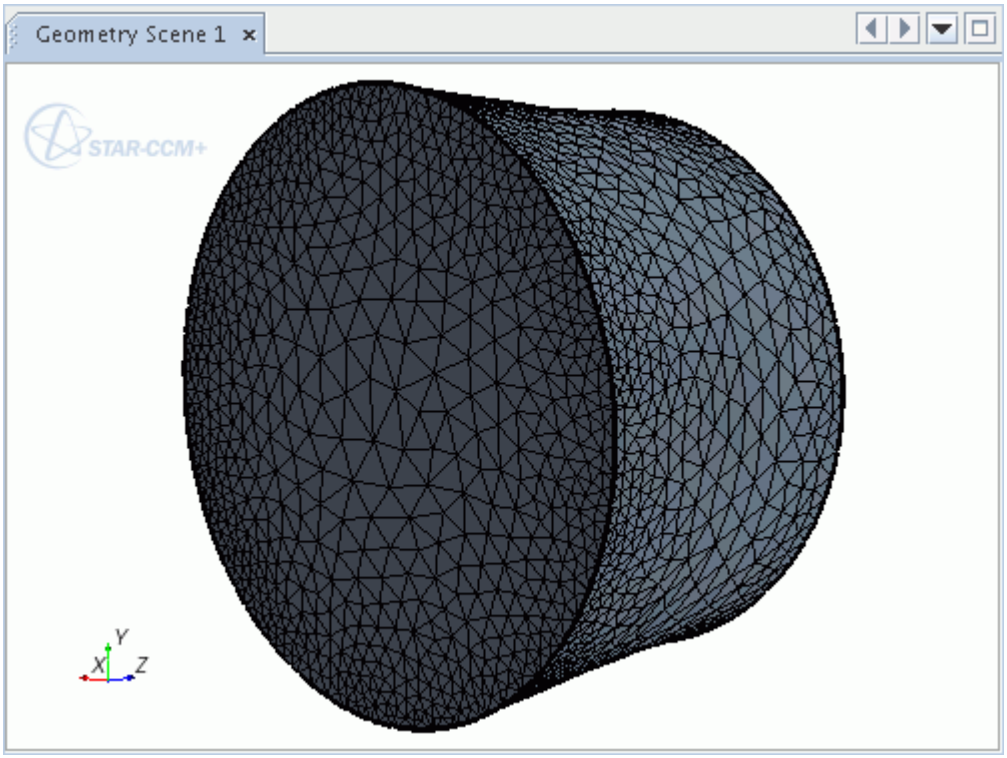
Using The Exact Face Match Method

Set the **Method** to the **Exact Face Match** option only when a part pair already has one-to-one correspondence between their coincident faces and vertices.

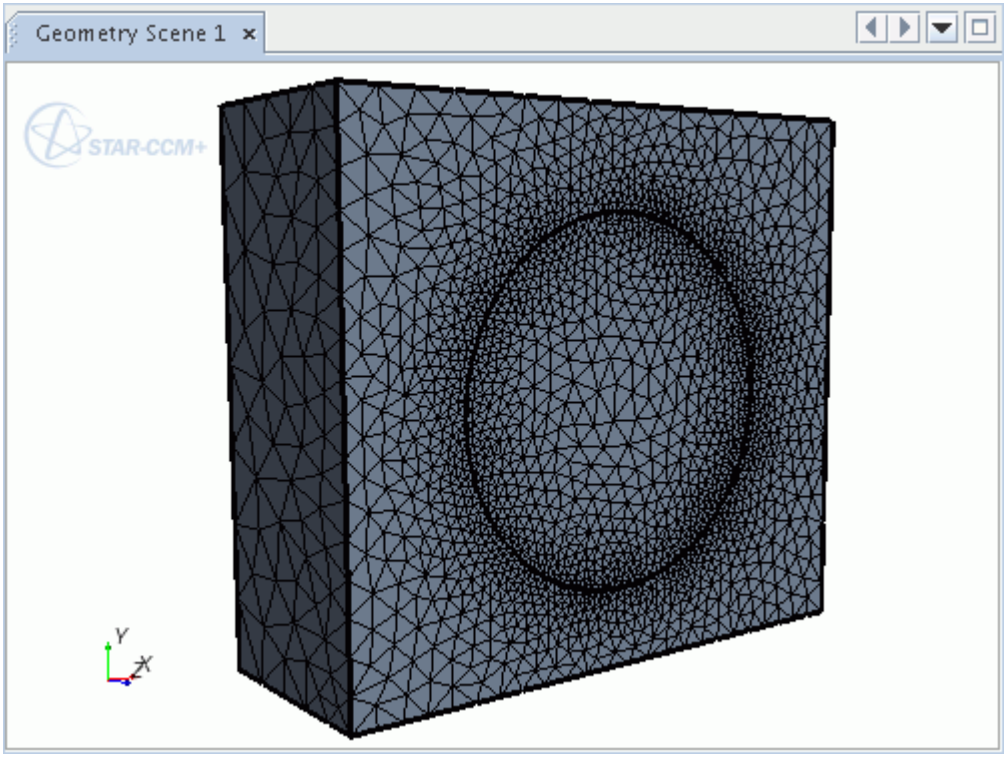
This conformal match between parts can occur in surface meshes exported from certain surface preparation software. Selecting this option speeds up the merge/imprint process, as the tool does not need to imprint and create identical faces on both surfaces. Instead, the tool simply merges the coincident vertices and faces, and creates a part contact (interface for regions) between the two parts.

When using this option, set a small **Tolerance** value, as the two surfaces are coincident. This tolerance prevents the undesired merging of faces and vertices in thin parts.

As an example of surface mesh requirements, consider the two parts that are shown below. The first image shows the contacting area of a cylinder part:



This second image shows the conformal and coincident surface mesh on the contacting block part:

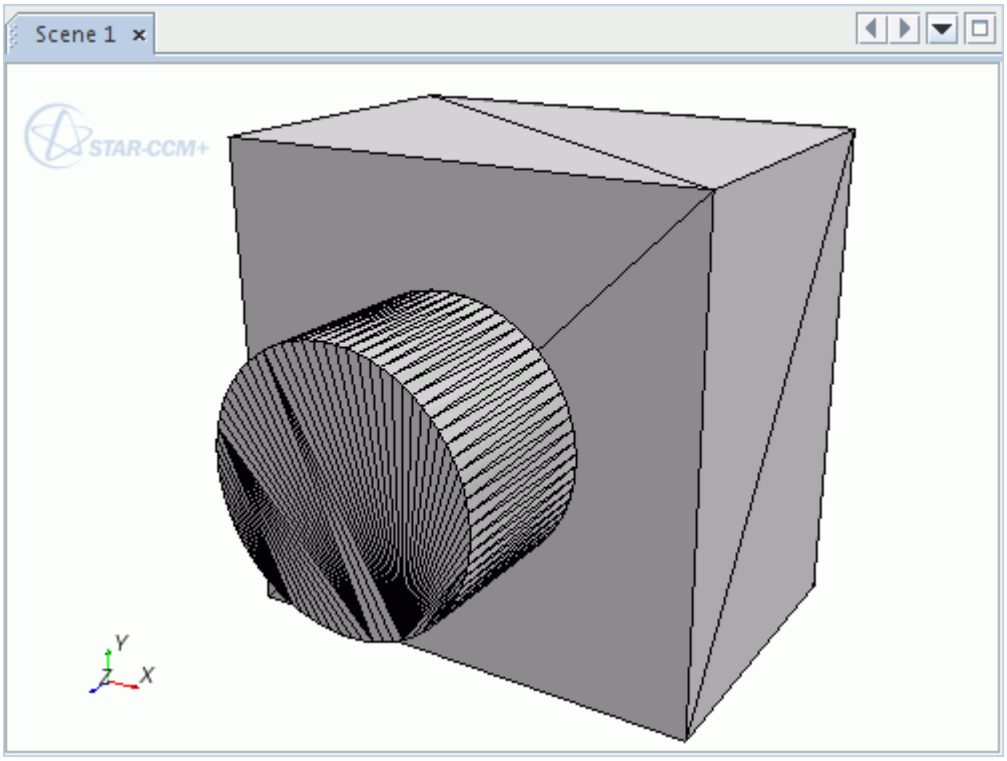


The contacting surface on the cylinder and the circular area on the block are identical.

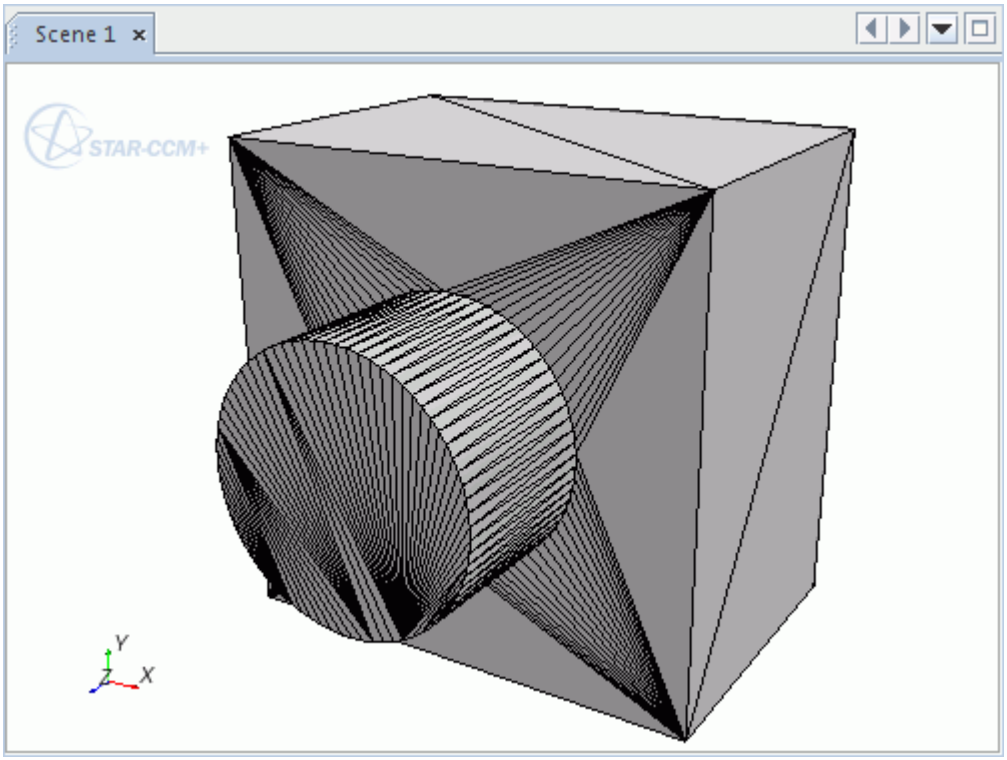
Using The Conformal Resulting Mesh Type Option

Use this option to achieve a conformal mesh between contacting surfaces.

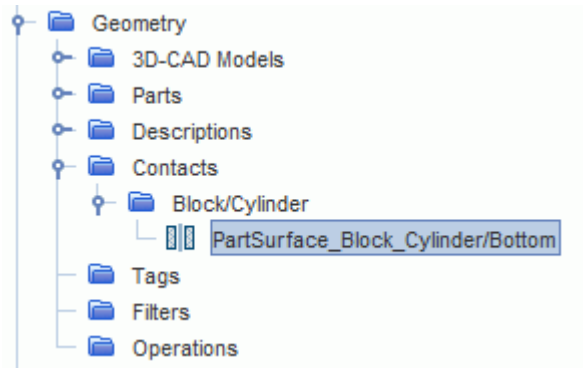
The following geometry shows two parts that have contacting surfaces between each other. To achieve a conformal mesh between the contacting surfaces, before you imprint the parts, set *Resulting Mesh Type* to **Conformal**.



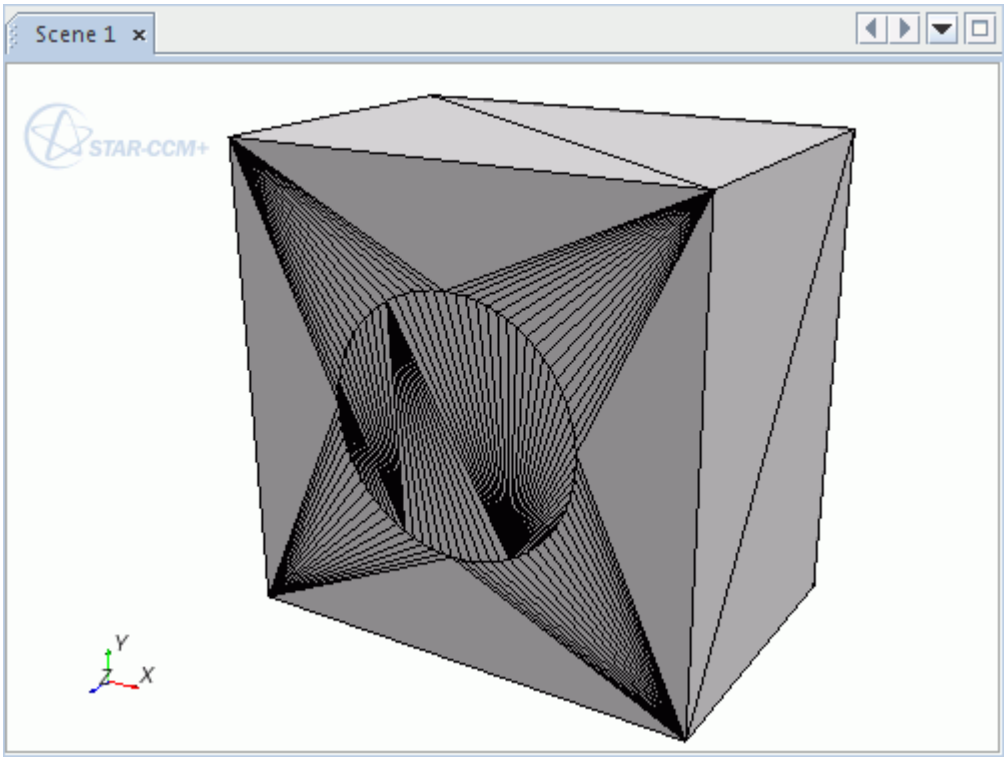
After imprinting, the resultant surfaces are conformal, which means they have the same surface discretization.



By default, in-place part contacts are created between the contacting surfaces. The part contacts appear in the simulation tree:



By hiding the cylindrical surface, you can see that the surface discretization on the contacting surfaces is identical.



Using The Non Conformal Resulting Mesh Type

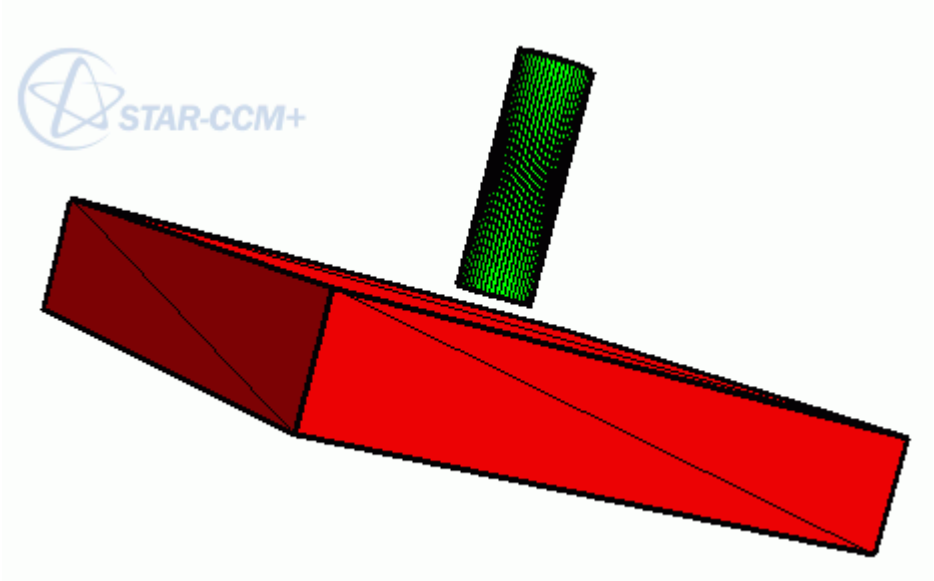
Non-conformal imprinting provides a way to imprint parts efficiently without spending time imprinting CAD parts conformally when it is unnecessary.

STAR-CCM+ allows you to imprint parts non-conformally. This process is useful when:

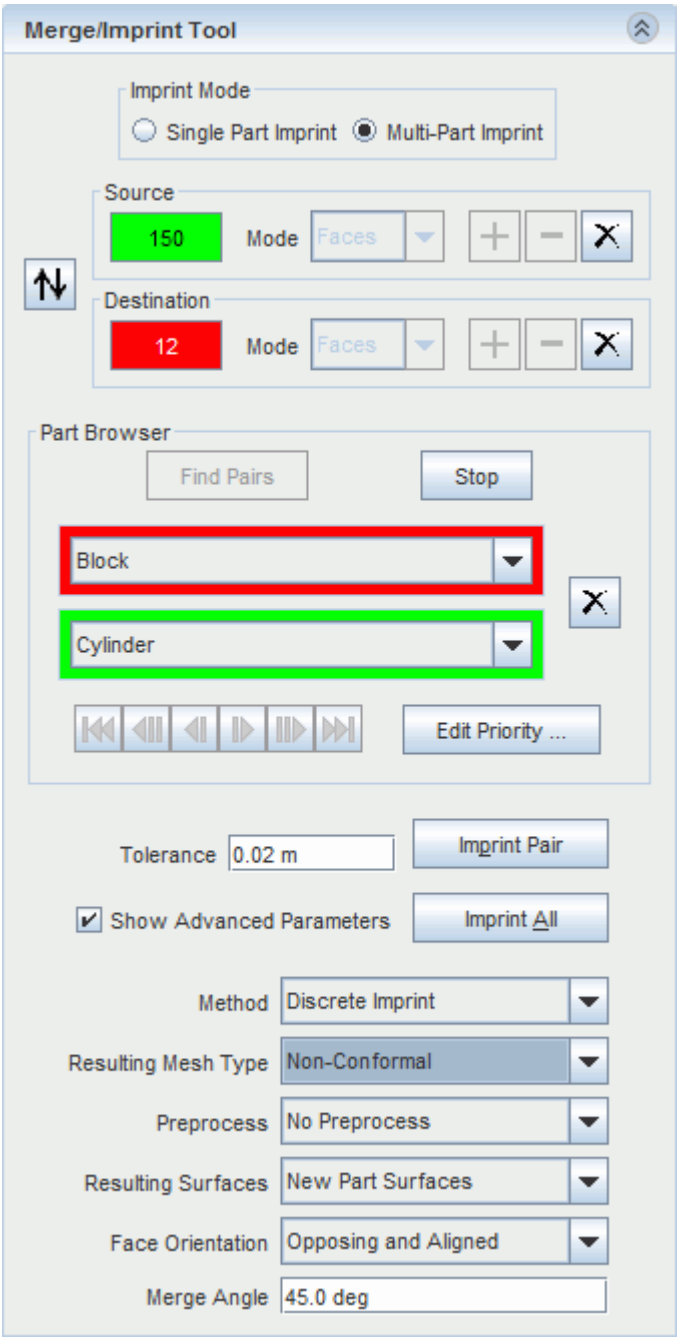
- a conformal interface is not necessary to achieve solution accuracy
- there are many contacting parts
- each part is within many pairs
- there are gaps between the CAD parts

During the process of non-conformal imprinting, pairs of split surfaces are created on contacting parts without altering the individual parts or creating shared faces.

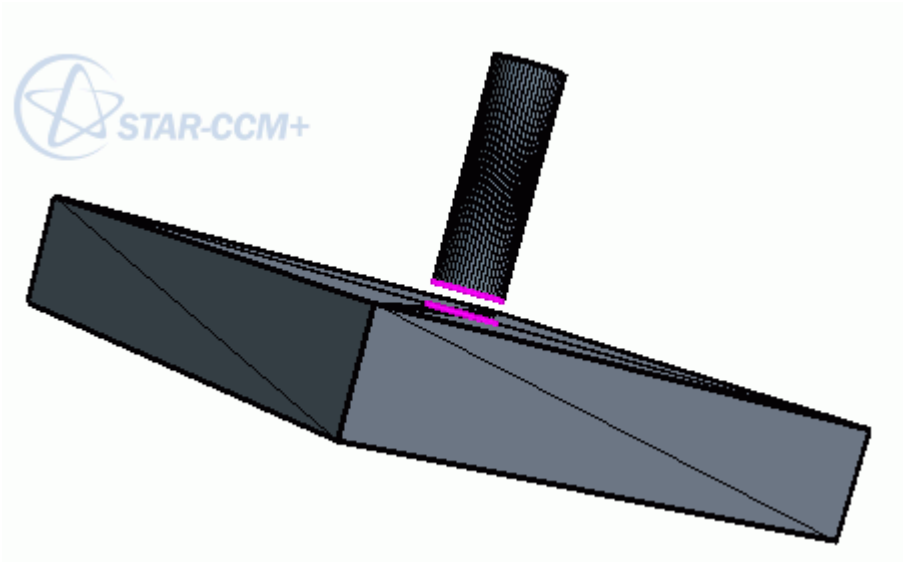
The following geometry shows two parts that are in close proximity with a small gap between the surfaces.



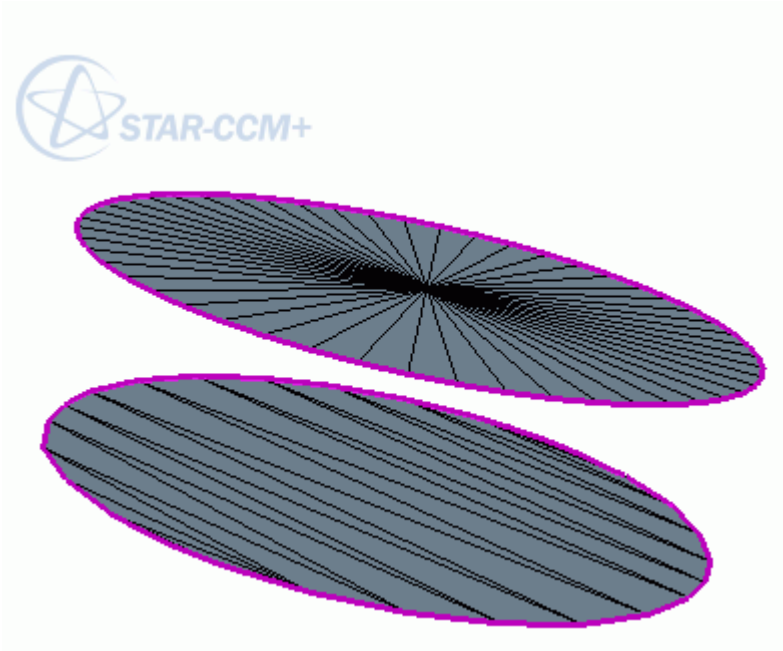
To achieve a non-conformal mesh when imprinting these surfaces, before you imprint the parts, activate *Show Advanced Parameters* and set *Resulting Mesh Type* to **Non-Conformal.**



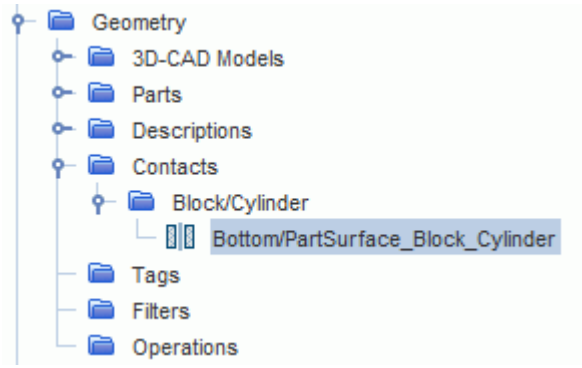
After imprinting the parts, the resultant surfaces are non-conformal, which means that the surface discretization on the imprinted surfaces does not match. Each part remains separate and there is no shared surface between each part. However, the original discretized surfaces are cut and a pair of split part surfaces are created which represent an in-place part contact between the surfaces.



Taking a closer look at the pair of split surfaces, you can see that the surface discretization on each split part surface is non-conformal:



The in-place part contact which represents the pair of split part surfaces appears in the simulation tree:



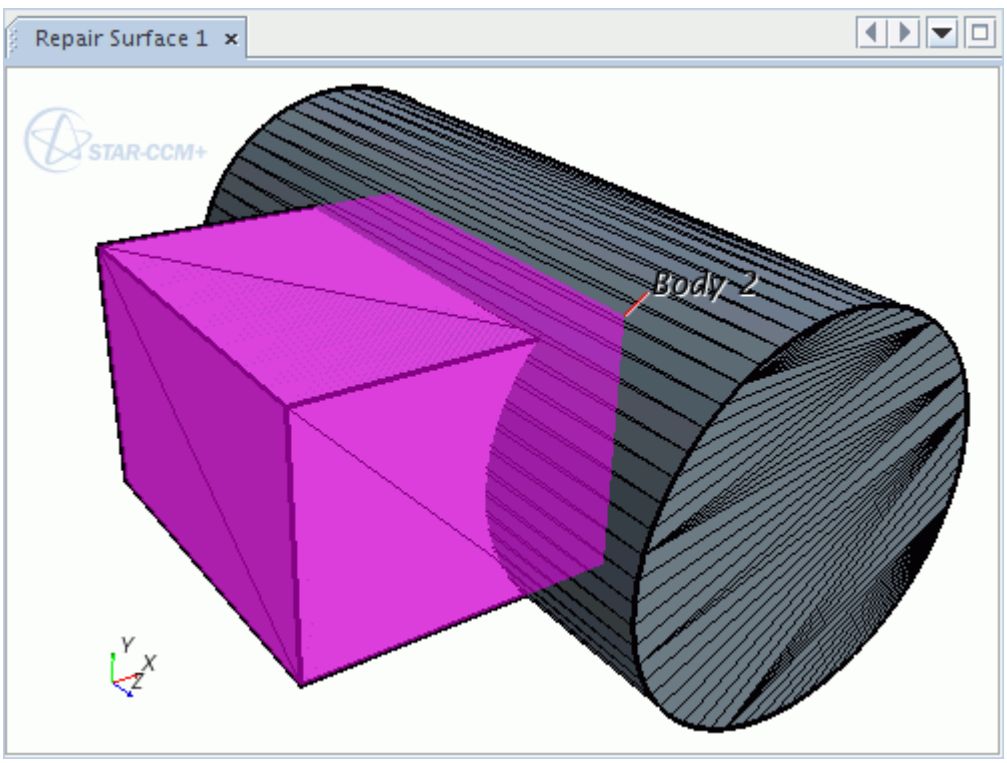
Using The Boolean Subtract Preprocess Option

This option subtracts the destination part from the source part before performing the imprint operation.

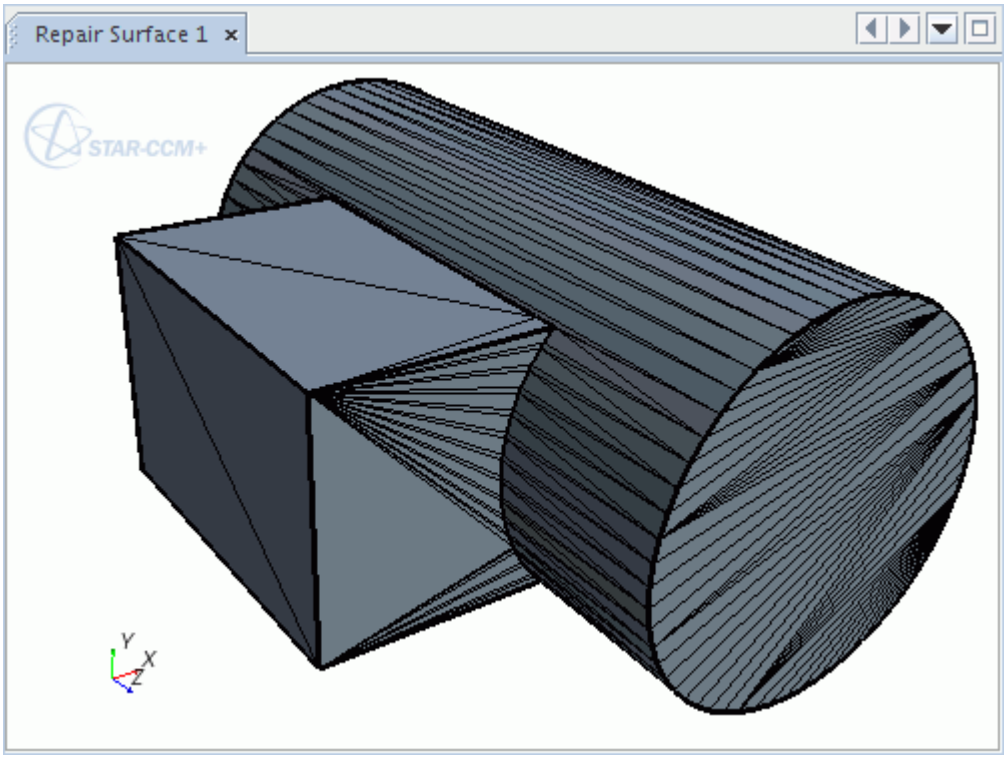
The geometry of the source part is changed, but the geometry of the destination part is maintained.

Note: To perform a boolean subtraction, the input parts must be closed and manifold. Therefore, selecting the Preprocess option, Boolean Subtract, deactivates the face selection controls, as the tool can only imprint two complete parts. If the parts are not closed and manifold, there is no subtraction, but the parts are imprinted.

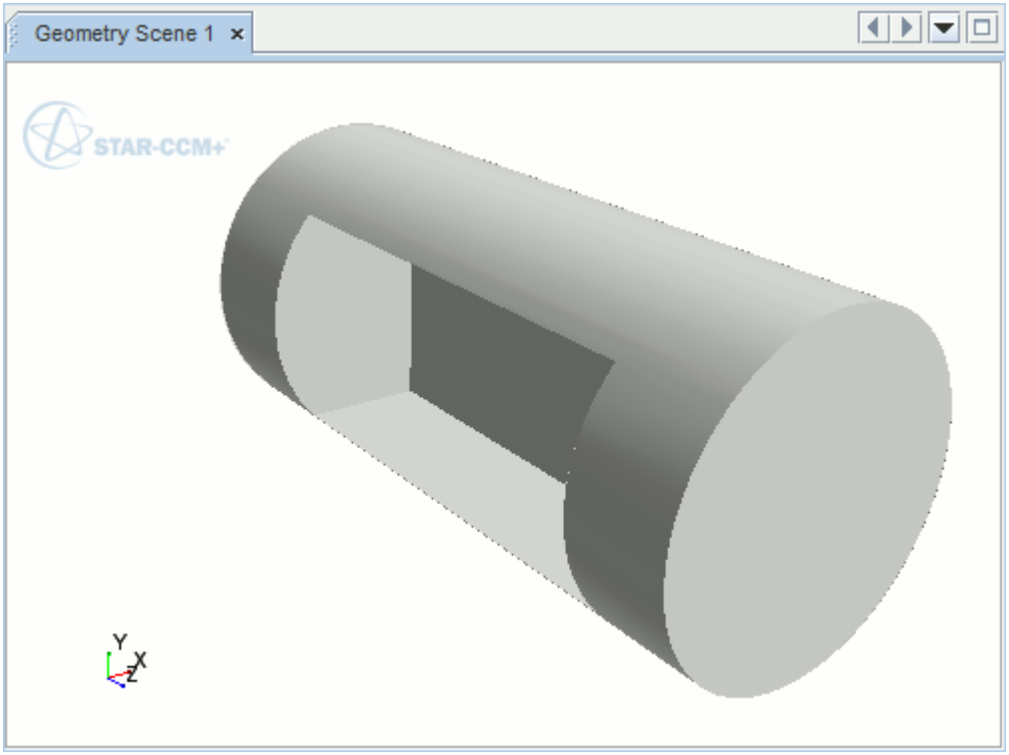
The following example shows two parts that penetrate one another.



Selecting the Boolean Subtract option and imprinting these two parts gives the following result, with the intersecting surface merged and imprinted.



By hiding the rectangular part, you can see that the rectangular part (destination) has been subtracted from the cylinder (source). The result is best seen by exiting Surface Repair.

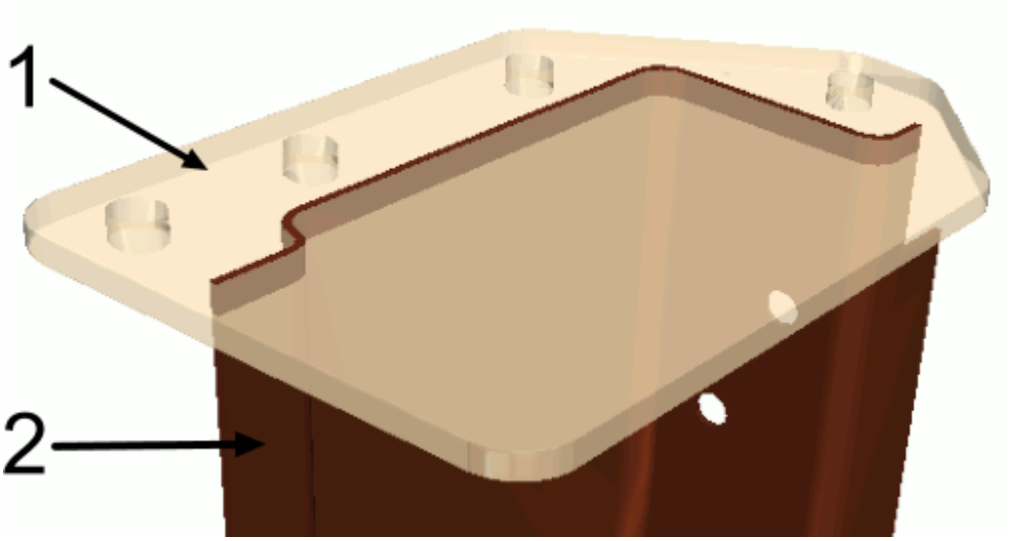


Using The Opposing Outward Face Orientation Option

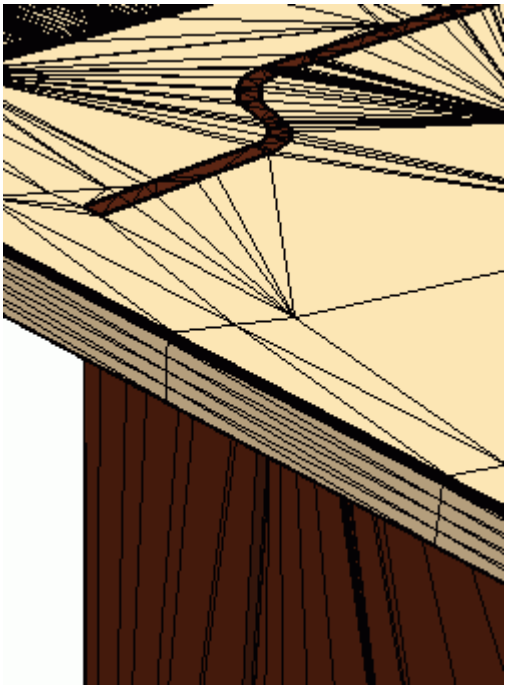
Selecting the *Face Orientation* option, *Opposing Outward*, provides an imprinting mode which ensures that only faces that are facing each other with opposing outward normal directions are imprinted or merged.

When imprinting thin, penetrating parts, the imprinter can choose the wrong side of the destination part with which to imprint or merge the source part.

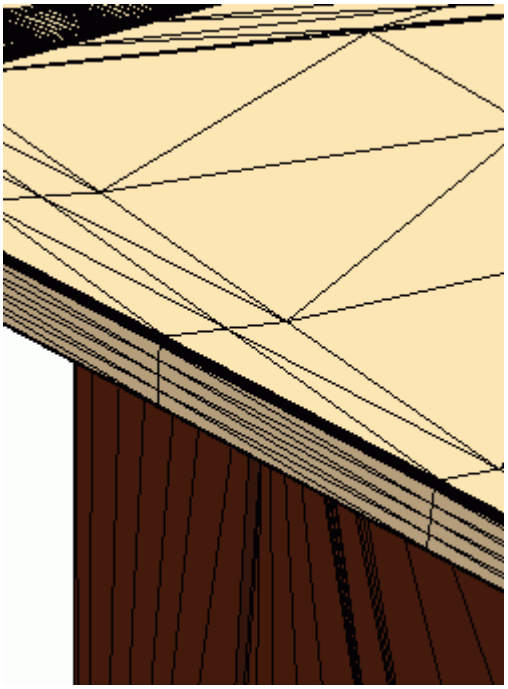
The following geometry shows two parts that intersect each other.



When these parts are imprinted with the *Opposing and Aligned* option selected, the surface of part 2 is imprinted onto the wrong side of part 1.



If these parts are imprinted with the *Opposing Outward* option selected, the surface of part 2 is imprinted onto the correct surface of part 1:

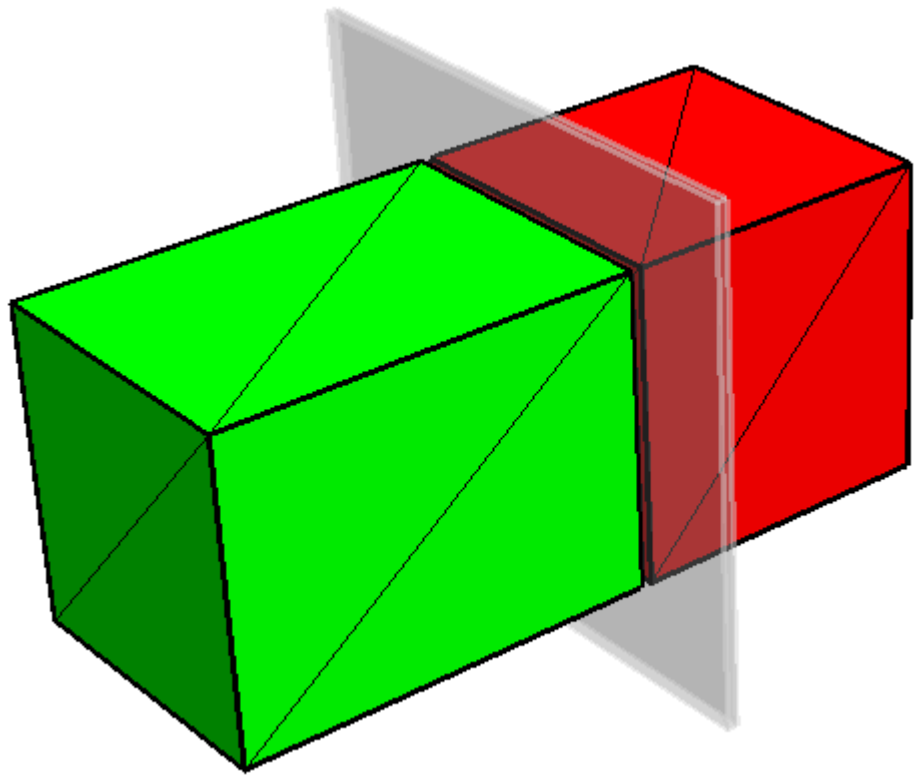


Displaying Neighboring Parts

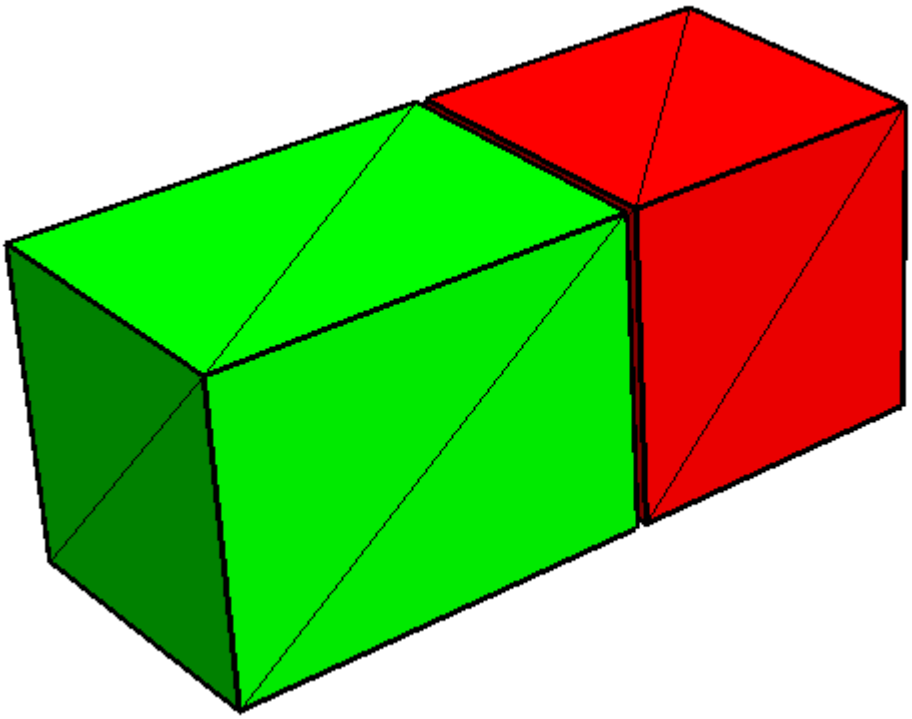
You can choose to display or hide neighboring parts in the scene once part pairs are found.

Neighbor parts are parts that are in close proximity (based on the supplied **Tolerance** input), to both the source and destination parts. If a part is only in proximity to either the source or destination, it is not displayed.

By default, the **Show Neighbor Parts** option is activated. This option is useful for judging the order to perform imprints due to the proximity of neighboring parts. For example, you can avoid performing imprints between parts that intersect others in-between them.



Deactivating this option hides the neighbor parts and only shows the destination and source parts.



The neighbor part display uses non-interactive elements to represent the neighbor display. These elements remain in the scene unless you deactivate the option, you click **Stop** in the *Part Browser* group box, or you exit the *Imprint* panel.

Note the following considerations when displaying or hiding neighbor parts:

- The neighbor parts are display-only and you cannot select or interact with them. Any selections that you make using the cursor passes through to any selectable parts underneath.
- If you use the **Show All Faces** tool to display all active repair parts while also showing neighbor parts, then the real and display parts are superimposed on each other. Similarly, **Hide All Faces** has no effect on the neighbor part display.
- You can toggle the mesh display for the neighbor parts on and off using the **Show All Meshes** toolbar toggle button. The default is for the mesh to be off.
- The **Color Displayed Faces by** option has no effect on the neighbor part display. Neighbor parts are displayed in all modes as long as the imprint panel is open and a valid current part pair is selected.
- If you use the (**Remove this source and destination pair browser**) option, then the parts will no longer appear as neighbor parts when browsing other part pairs.

Merge And Imprint Advanced Parameters

Advanced parameters are available to set when merging and imprinting multiple parts using the surface repair panel.

Method	Specifies how faces are chosen and matched for imprinting.	
	Discrete Imprint	The imprint is carried out using the tessellated surface mesh of the parts. The original tesselated faces are cut to ensure that the imprinted faces match.
	Exact Face Match	The imprint is carried out using the tessellated surface mesh of the parts. Coincident vertices and faces are merged and a part contact is created between the two parts. Specify this option for part pairs that already have exact one-to-one correspondence between the faces and vertices. See Using the Exact Face Match Method.
Resulting Mesh Type	Determines if the imprinter merges the faces of the input parts.	
	Conformal	The imprinter merges faces of the imprinted parts to produce a conformal mesh on the output surface.
	Non-Conformal	The imprinter does not merge faces of the imprinted parts and a non-conformal mesh is produced on the output surface.
Preprocess	Allows you to specify if any boolean operations are performed before parts are imprinted.	
	No Preprocess	No boolean operations are performed before parts are imprinted.
	Boolean Subtract	A boolean subtract operation is performed before parts are imprinted.
	Intersect Surfaces (available for non-conformal imprinting only)	An intersect operation is performed to cut the existing surfaces before parts are imprinted. Coincident surfaces that lie inside other surfaces are also cut and automatically included in the resulting part surfaces.
Resulting Surfaces (available for conformal imprinting only)	Allows you to specify the surface that the resulting imprinted faces are added to. This option is only available for conformal imprinting as non-	

	conformal imprinting creates new part surfaces by default. See Imprint Parts Resulting Part Surfaces.	
	Input Part Surfaces	The imprinted faces that are created are added to the part surface of the input part.
	New Part Surfaces	The imprinted faces that are created are added to a new part surface.
Face Orientation	Allows you to specify the orientation of the matching face normals.	
	Opposing and Aligned	Only target faces that are aligned and opposed outward from the normal of the matched source faces are merged.
	Opposing Outward	Only target faces that are opposed outward from the normal of the matched source faces are merged. See Using the Opposing Outward Face Orientation Option.
	Aligned Outward	Only target faces that are aligned outward from the normal of the matched source faces are merged.
Merge Angle	Specifies the maximum angle between the normals of the faces at which parts merge. Only pairs of faces which lie within the specified *Value* and have normal directions which differ by less than this *Merge Angle* are merged.	

See Also:

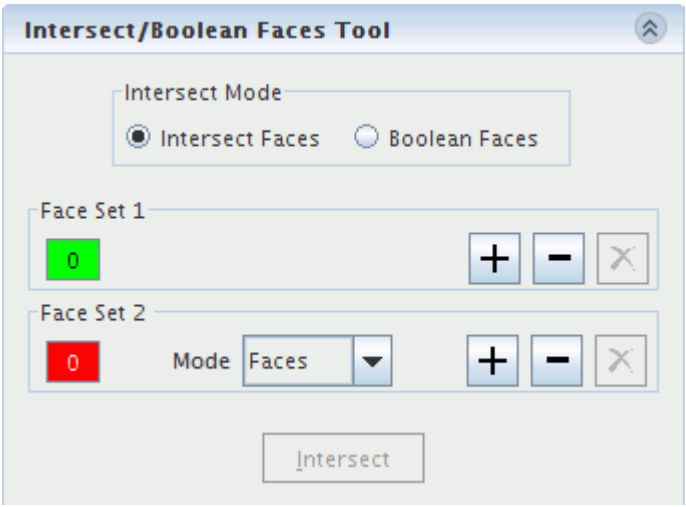
Using The Opposing Outward Face Orientation Option
Imprint Parts Resulting Part Surfaces
Using The Exact Face Match Method
Imprinting Multiple Parts

Intersect/Boolean Faces Tool

The **Intersect/Boolean Selected Faces** tool allows you to intersect or perform Boolean operations on two face sets.

A face set is a collection of faces. For face intersection, you can additionally use a user-defined plane instead of one of the face sets.

To display the *Intersect/Boolean Faces Options*, click (**Intersect or Boolean selected faces**).



This tool has the following modes:

- Intersect Faces
- Boolean Faces

Intersect Faces Mode

The **Intersect Faces** mode is similar to the boundary intersection feature, in that it intersects two or more separate surfaces to produce a merged result. Both tools produce identical results for a given face set.

The **Intersect Faces** mode provides the following functionality:

- Intersection of individual faces.
- Intersection of faces belonging to different surfaces.
- Ability to undo/redo the operation.

The following sections provide examples of using the **Intersect Faces** mode:

- Intersecting Two Face Sets
- Intersecting Faces with a Plane

Note: The tool does not create feature curves at the intersection plane of the two surfaces. If feature curves are required, create them manually.

Boolean Faces Mode

The **Boolean Faces** mode unites, intersects, and subtracts user-defined face sets.

To be a valid selection for a Boolean operation, make sure that face sets meet the following requirements:

- Each face set is composed of single or multiple closed volumes that do not contain non-manifold edges or vertices.

  If the face set does not meet this requirement, the *Output* window indicates which face set is not closed or manifold.
- Face sets cannot contain self-intersecting surfaces.

- Face sets cannot contain faces that are directly involved in a part contact. To perform a Boolean operation on such a face set, remove the part contact.

The tool aims to maintain part contacts that are indirectly involved in the Boolean operation, such as a face in a part contact that is connected to a face in a face set.

The following section provides an example of using the **Boolean Faces** mode:

- Performing a Boolean on Faces

Contents:

Intersecting Two Face Sets
Intersecting Faces With A Plane
Performing A Boolean Operation On Faces

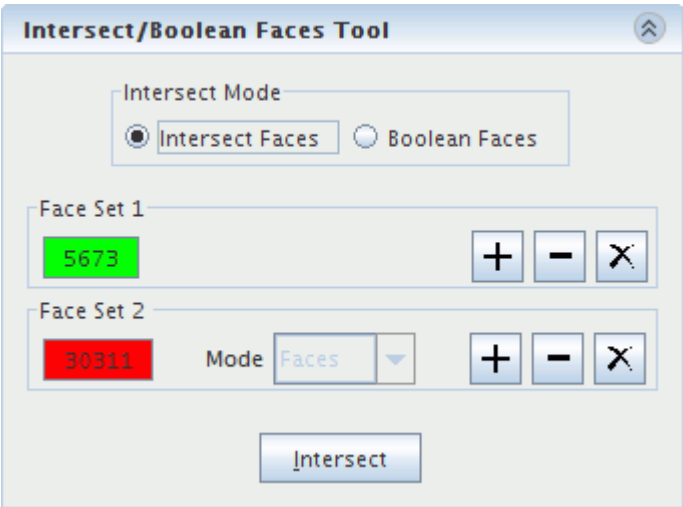
See Also:

Changing the Parts in the Active Model
Performing A Boolean Operation On Faces

Intersecting Two Face Sets

Use this procedure to intersect two face sets.

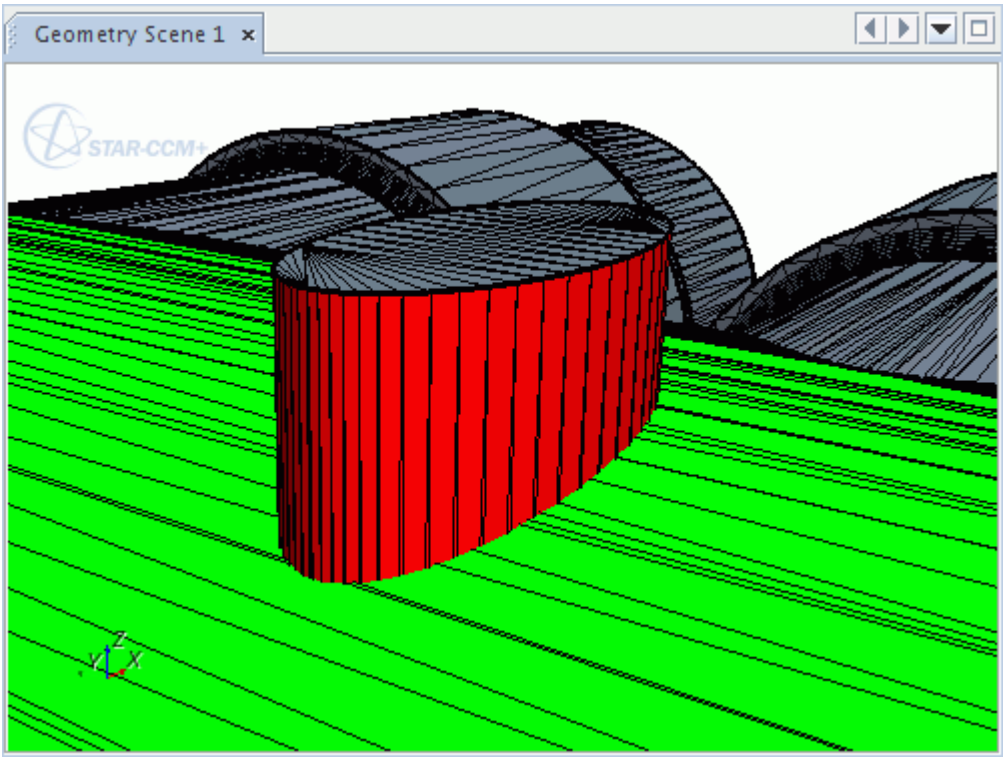
1. Set **Intersect Mode** to **Intersect Faces**.
2. Select the faces or surfaces belonging to the first *Face Set* using the Selection Controls.
3. In the *Intersect/Boolean Faces Tool* , click + for *Face Set 1* to add the selected faces to the set.
4. Ensure that **Mode** is set to **Faces** and repeat this selection and adding process for *Face Set 2*. In the *Graphics* window, *Face Set 1* appears green and *Face Set 2* appears red.



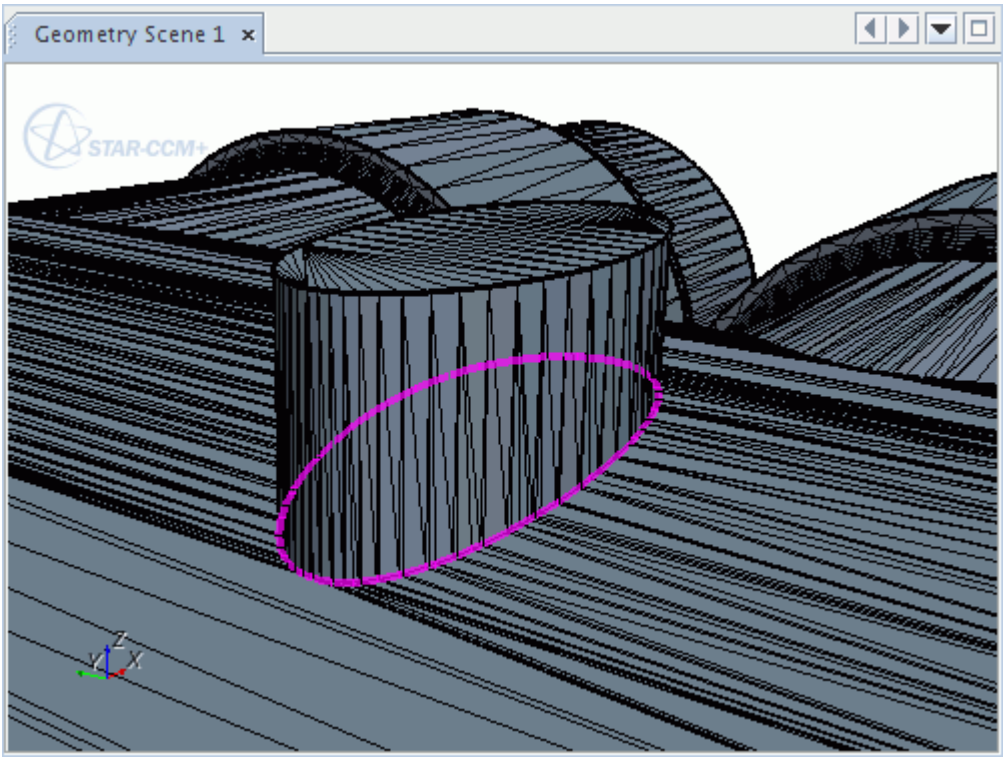
5. You can remove faces or edges by selecting them in the display, and clicking − for *Face Set 1* or *Face Set 2*.

6. To remove all faces from *Face Set 1* or *Face Set 2*, click .
7. Click **Intersect** to intersect the selected surfaces.

   The following example intersects the red and green surfaces.



After intersecting the surfaces, the tool retessellates the surfaces at the intersection plane and adds newly created faces to the nearest neighboring surface. The intersection operation also selects the edges at the join so that you can easily define them as feature curves.



See Also:

Selecting Entities

Intersecting Faces With A Plane

Use this feature to intersect geometry parts based on a plane that you specify.

To intersect a set of faces with a plane:

1. Set *Intersect Mode* to **Intersect Faces**.
2. Select the faces or surfaces that you want to intersect.
   a) Use the Selection Controls to select the appropriate faces.
   b) In the *Intersect/Boolean Faces Tool,* click + for *Face Set 1* to add the selected faces to the *Face Set*.
3. To define the intersection plane, set the **Mode** for *Face Set 2* to **Plane**.
4. Specify the Plane Origin and Plane Normal coordinates.
5. Set the **Coordinate System** to the system in which the origin and normal coordinates are defined.
6. Click **Intersect**.

Contents:

Specifying The Plane Origin
Specifying The Plane Normal

See Also:

Selecting Entities

Specifying The Plane Origin

You can specify the Plane Origin coordinates using three possible methods.

Set specific coordinates in the Plane Specification group box

1. Enter Plane Origin X Y Z values.
2. In the *Graphics* window, review the location of the Plane Origin point to ensure that it is correct.

Define coordinates interactively in the Graphics window

3. Next to the Plane Origin X Y Z coordinates, click **Pick**.
4. In the *Graphics* window, click an area on the surface of the geometry that determines the location of the Plane Origin.

Define coordinates using the Display Plane Tool

5. In the *Plane and Section Tools* group box, activate *Display Plane Tool.*
6. In the *Graphics* window, click and drag the Plane Origin **Pick** tool until the required location is achieved.

   The Plane Origin coordinates can be set as one of three Plane Normal points when specifying the Plane Normal.

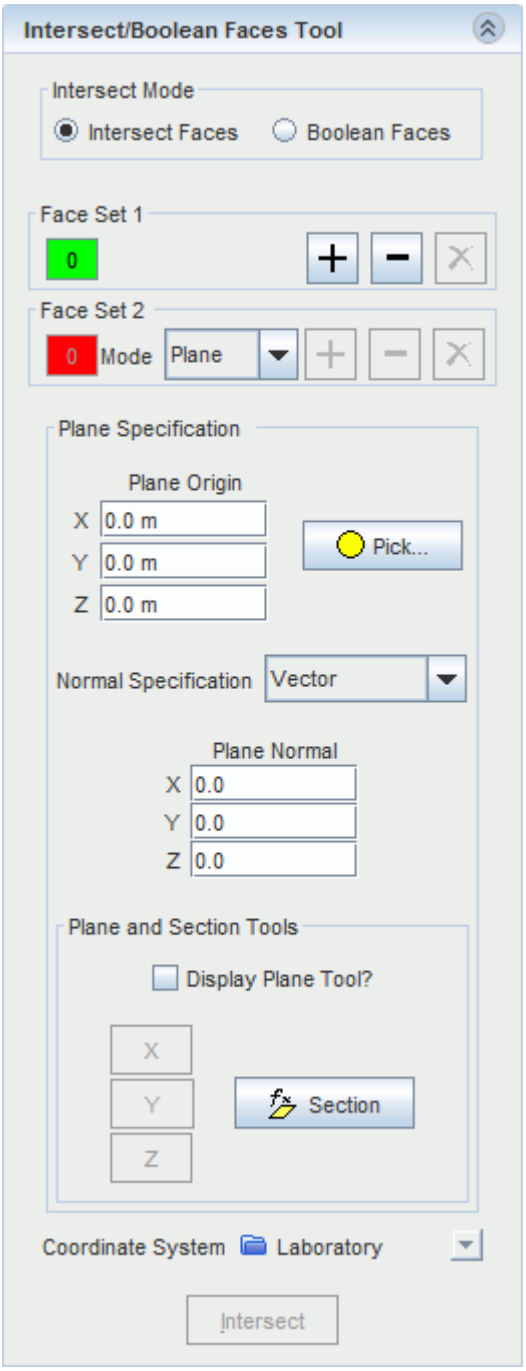
See Also:

Specifying The Plane Normal
Specifying The Plane Normal

Specifying The Plane Normal

There are two Normal Specification options available to define the Plane Normal coordinates.

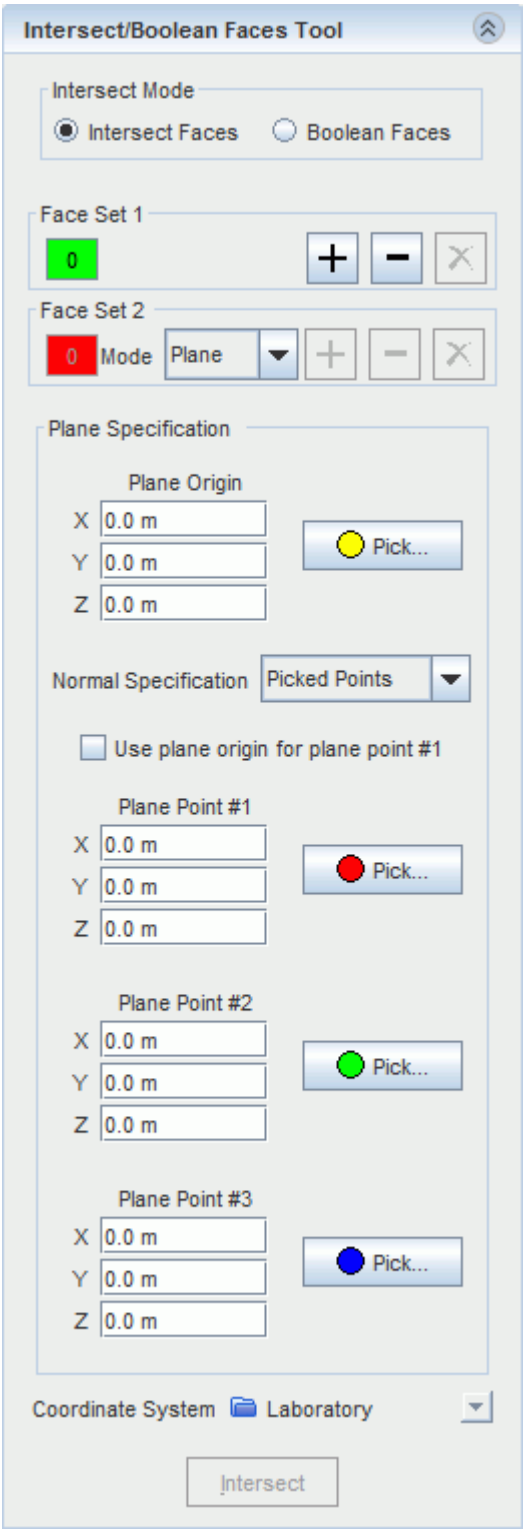
Vector



1. Set *Normal Specification* to **Vector**.
2. To specify the Plane Normal coordinates, either:
   - Enter Plane Normal X Y Z values, or
   - Define Plane Normal Vector coordinates using the Display Plane Tool.

     See Specifying the Plane Origin.

Picked points



3. Set *Normal Specification* to **Picked Points**.
4. To specify the Picked Points coordinates, either:
   - Specify three independent Plane Points coordinates that represent the Plane Normal. Using the same process as when specifying the Plane Origin, enter the Plane Points' X Y Z values directly, or define the Plane Points interactively in the Graphics window.

- Specify three Plane Points coordinates using the Plane Origin coordinates for one of the Plane Points. Activate *Use plane origin for plane point #1*, then specify the remaining Plane Points' using the same process as for specifying the plane origin.

See Also:

Specifying The Plane Origin

Specifying The Plane Origin

Performing A Boolean Operation On Faces

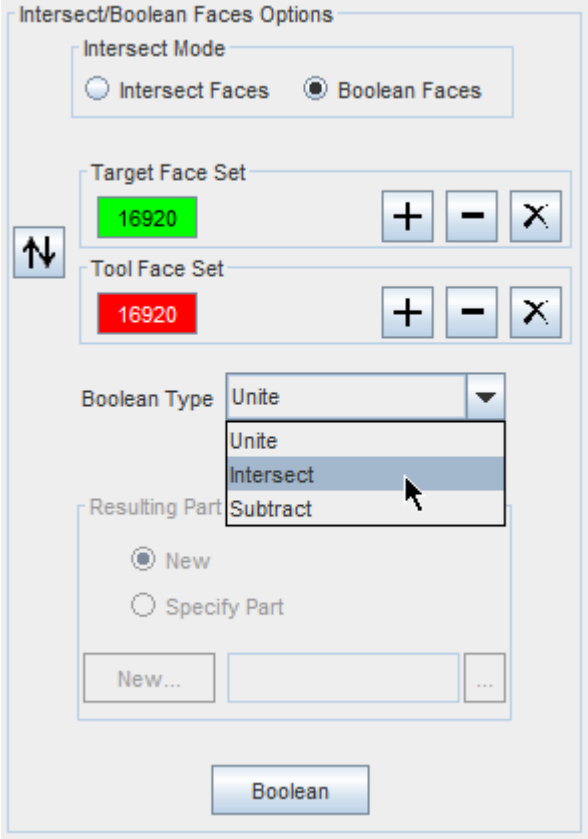
A Boolean operation allows you to unite and intersect two face sets, or to subtract the Tool face set from the Target face set.

Note: The remaining faces after a Boolean operation belong to the original part and part surface.

To perform a Boolean operation:

1. Set *Intersect Mode* to **Boolean Faces**.
2. Define the *Tool* and *Target* face sets with the selection buttons.

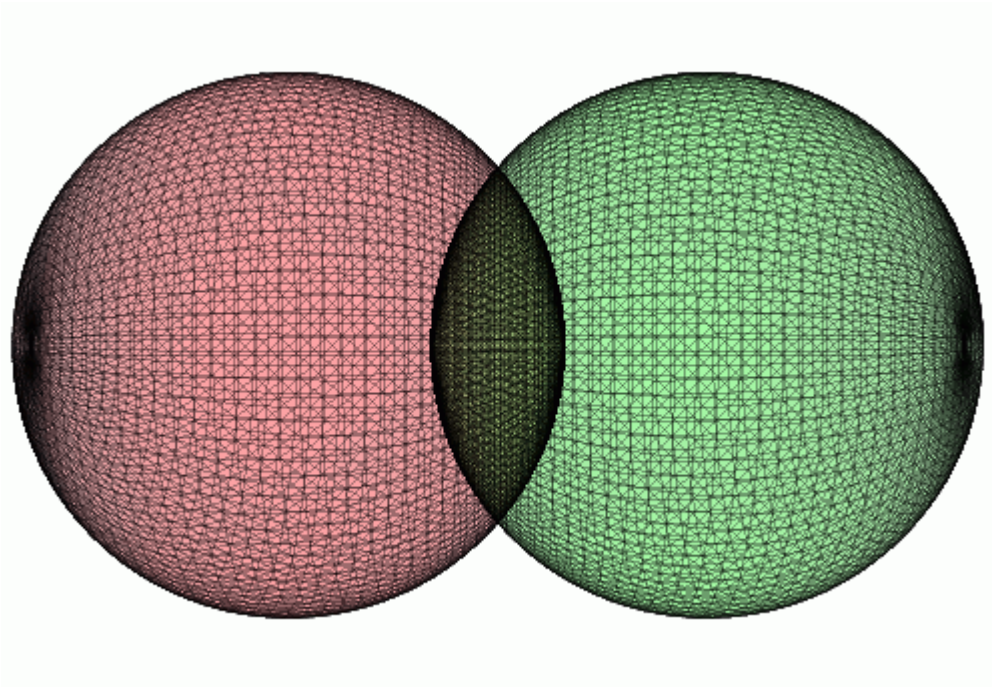
   See Intersect/Boolean Faces Mode for a list of requirements for the face sets.
3. Set **Boolean Type** to one of the following options:
   - **Unite** — Removes the faces that are enclosed by the overlapping volume between the face sets.
   - **Intersect** — Maintains only the overlapping volume between the face sets and removes all other volumes.
   - **Subtract** — Removes the Tool face set from the Target face set.
     - To achieve the desired result, you can use the swap button to change the Tool and Target selections.
     - To subtract multiple tool volumes from the target volume, first unite the tool volumes before performing the subtract operation.



4. Click **Boolean** to perform the Boolean operation.

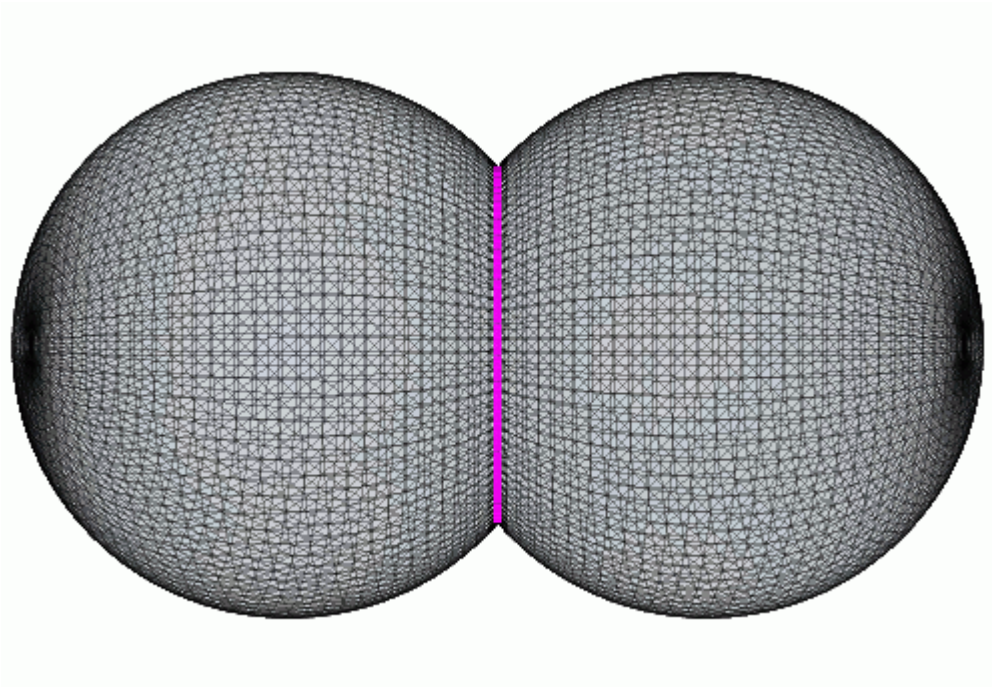
Examples of Boolean Operations

These examples use the geometry parts shown in the following image to demonstrate the different Boolean operations. These two spheres overlap so that they share a common volume. The green sphere is the Target face set and the red sphere is the Tool face set.



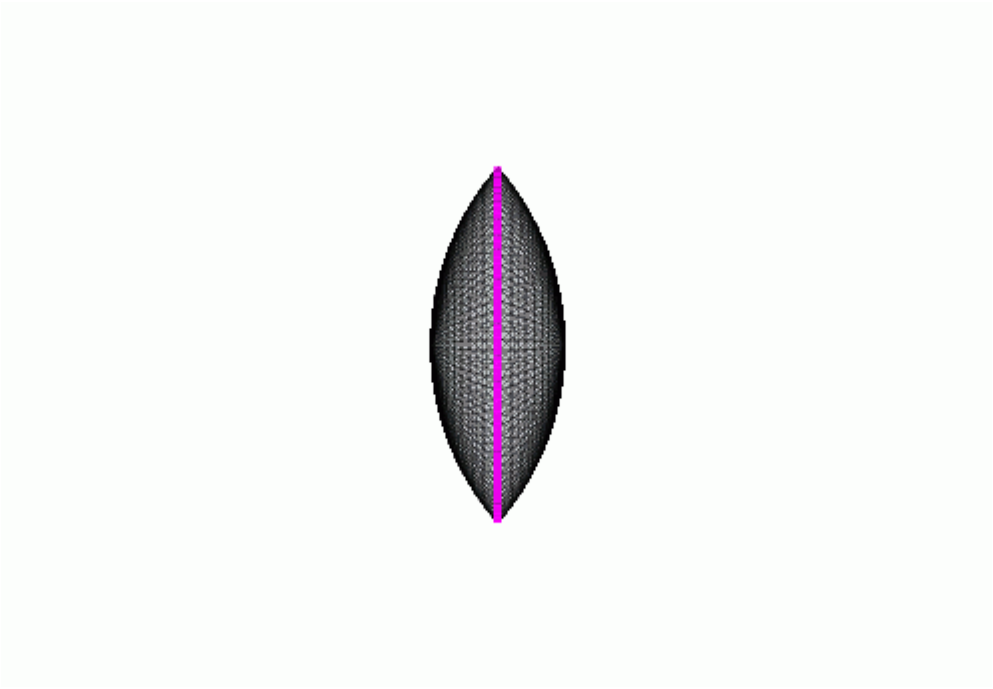
Unite

Using the unite operation removes the common volume between the two face sets.



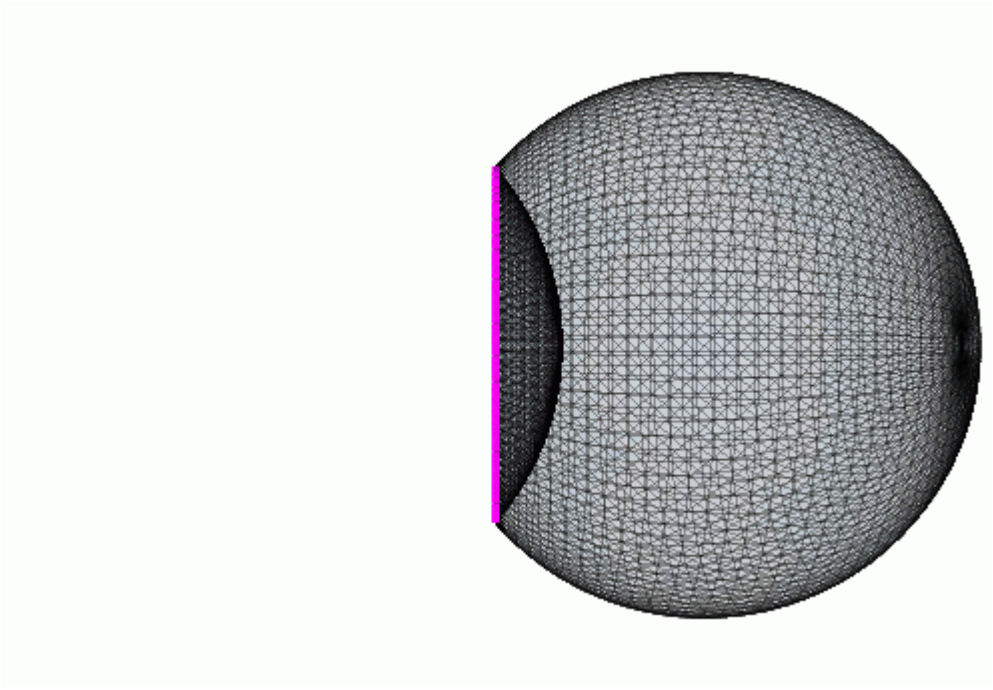
Intersect

Using the intersect operation leaves only the common volume between the two face sets.



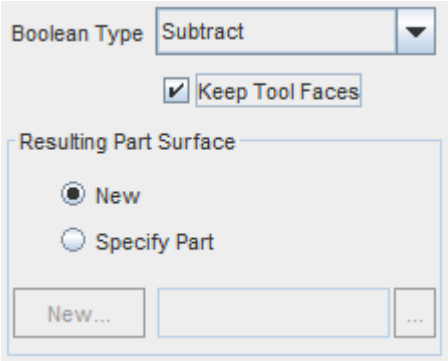
Subtract

Using the subtract operation removes the Tool face set (red) from the Target face set (green).



To keep the tool faces after performing the subtract operation, activate the **Keep Tool Faces** options. This option is only available for the **Subtract** Boolean type and is grayed out for all other Boolean types.

Two options are provided to determine the part surface types for the faces that are created at the interface between the target and tool faces during the subtract operation:



- **New** — Creates a new part surface named **Boolean Subtracted Faces** within each subtracted part.

  This option is useful in cases where multiple target parts are being subtracted and you want to maintain closed parts afterwards. The newly created faces are automatically associated with the part they are attached to.
- **Specify Part** — Assigns all created faces to a single specified part surface.

See Also:

Intersect/Boolean Faces Tool

Edit Connected Surfaces Tool

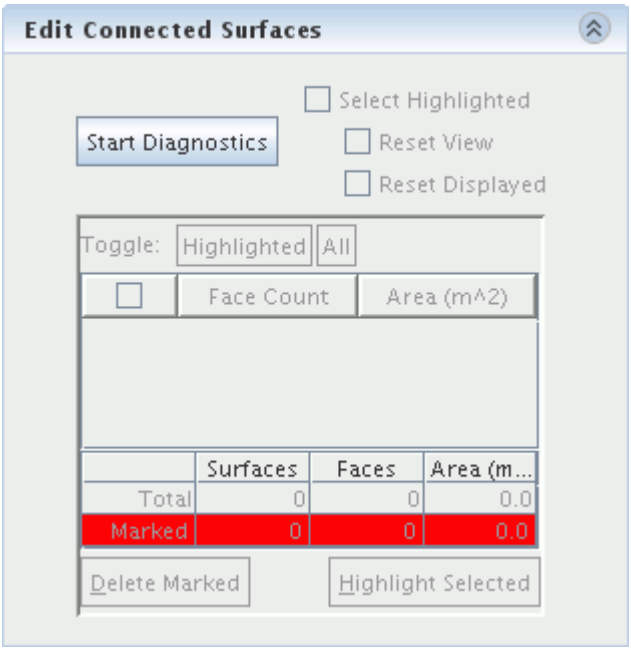
The **Edit Connected Surfaces** tool identifies the connection or disconnection between tessellated surfaces, and allows deletion of disconnected surfaces.

The tool can identify surfaces that must be retained, as they are important, while others can be unnecessary facets that you can delete.

You typically use **Edit Connected Surfaces** tool on a wrapped surface to remove geometric details, retained by the surface wrapper, which you do not require for the remaining meshing process. However, you can also use the **Edit Connected Surfaces** mode on imported, remeshed, or auto repaired surfaces.

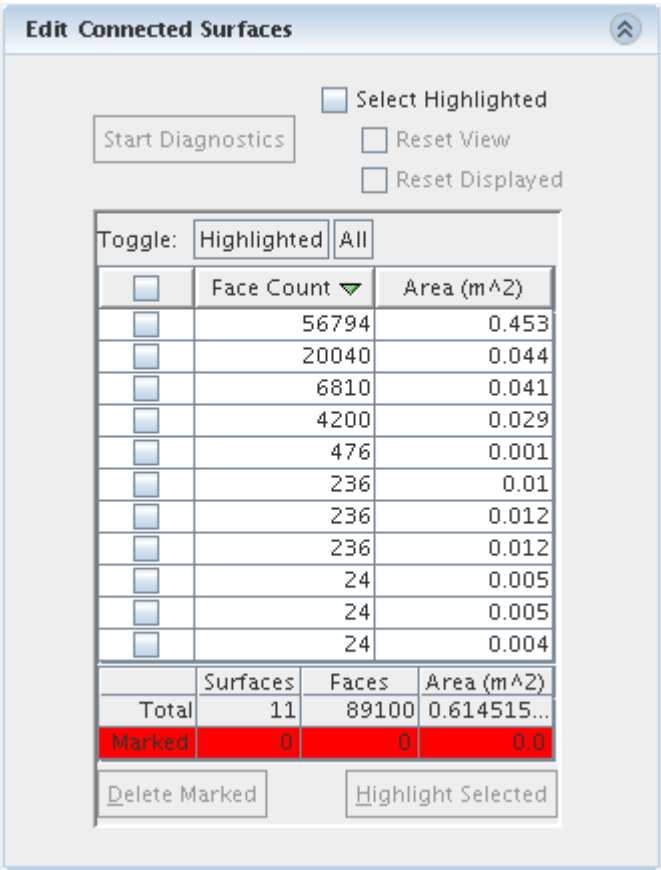
This tool deletes connected surfaces as a whole. To delete individual faces, use the Surface Repair options. See Repair Tools Reference.

To display the *Edit Connected Surfaces Options*, click (**Find, review and delete connected surfaces**).



1. To identify the connected and disconnected surfaces, click **Start Diagnostics**.

   When the diagnostics are complete, the *Edit Connected Surfaces* table populates with a summary of connected surfaces on the part:



The table contains a summary of the face count and area (m²) of each connected surface within the part. By default, the table sorts the data by descending face count.

2. You can resort the data by clicking the **Face Count** or **Area (m^2)** column headers.
   The tool also provides a summary of total **Surfaces**, **Faces, and Area** at the bottom of the group box.

   You can change the mode at any time to access one of the other tool sets. However, if you change the surface, you regenerate the connected surfaces information by clicking **Start Diagnostics**, when returning to **Edit Connected Surface** mode. Additionally, any changes that you make using **Edit Connected Surfaces** (such as deleting a surface) update the surface errors diagnostics in the surface repair tool.

   There are three different methods for using **Edit Connected Surfaces** mode:
   - Selecting surfaces in the part
   - Highlighting a selected face in the summary table
   - Deleting unwanted surfaces

Contents:

Selecting Surfaces Using The Summary Table
Highlighting A Selected Face In The Summary Table
Deleting Unwanted Surfaces Using The Summary Table

See Also:

Changing the Parts in the Active Model
Importing Surface Data into STAR-CCM+
Surface Remesher
Surface Wrapper

Selecting Surfaces Using The Summary Table

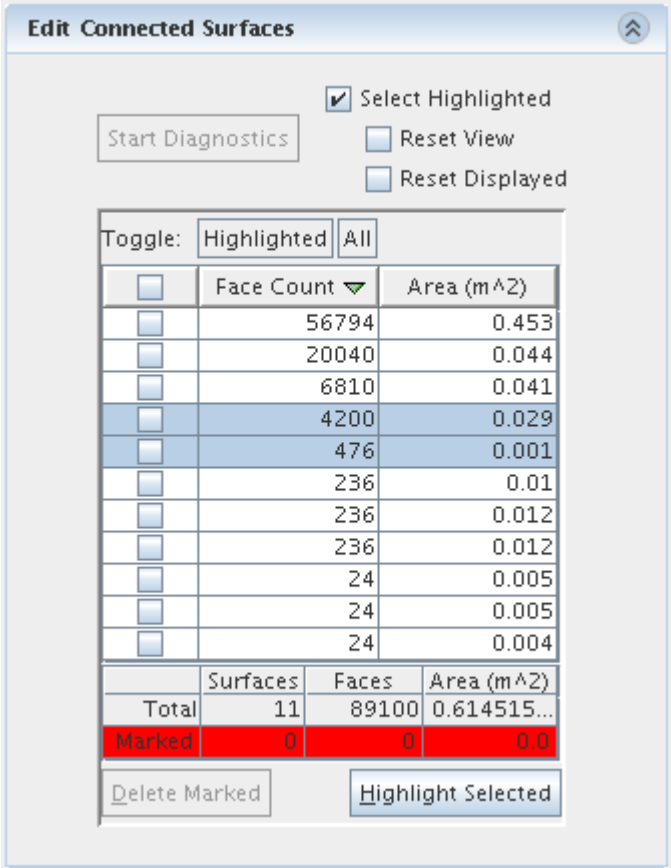
You can use the summary table to select part surfaces in the *Graphics* window, allowing you to locate and evaluate each connected surface.

To select surfaces using the summary table, you first generate the connected surface diagnostics. You can then activate the **Select Highlighted** option, which highlights selected table entries in the *Graphics* window.

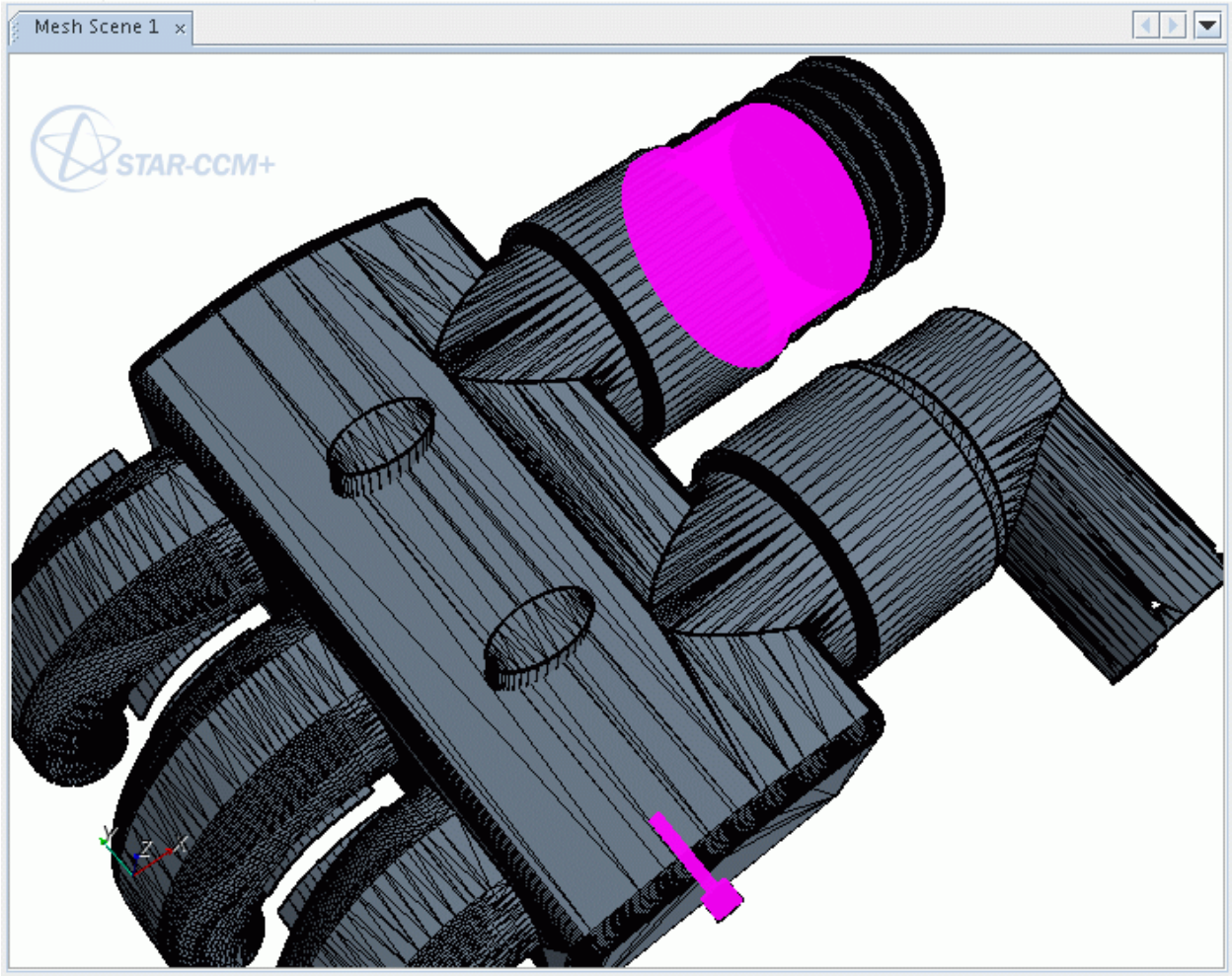
Furthermore, two more options provide further help in recognizing and identifying connected surfaces:

- **Reset View** resets the view to zoom in on the displayed faces, edges, and vertices
- **Reset Displayed** resets the displayed faces, edges, and vertices to the error and two neighbors

1. If you use **Reset Displayed**, you can redisplay the entire part by clicking **Show All Faces** within *Display Controls*.
   The following example shows two highlighted table entries with the **Select Highlighted** option activated, which selects the corresponding surfaces.



The selected surfaces appear as follows:



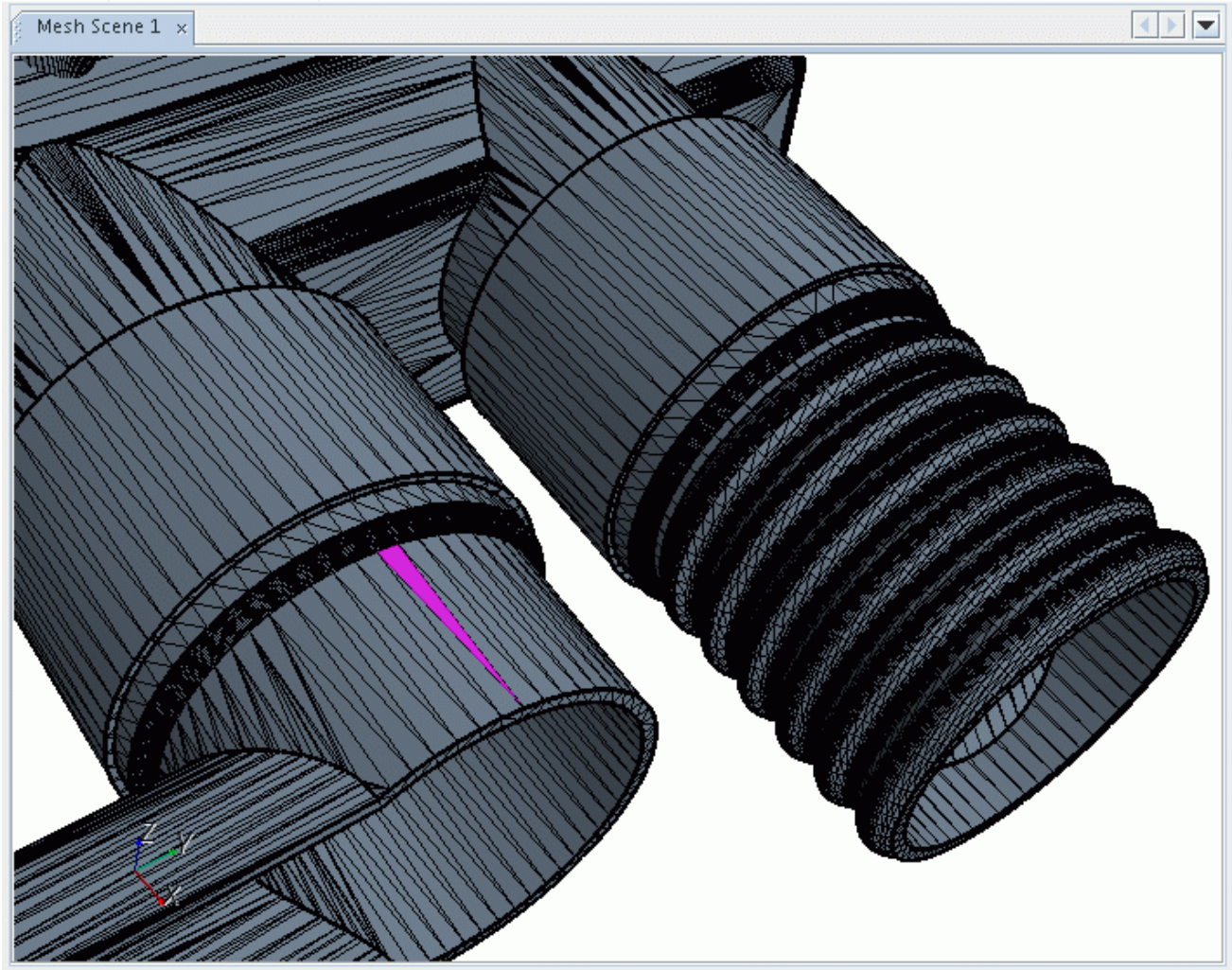
2. To deselect faces, you can click anywhere in the white space in the scene or click **Select None** within *Selection Controls*.

Highlighting A Selected Face In The Summary Table

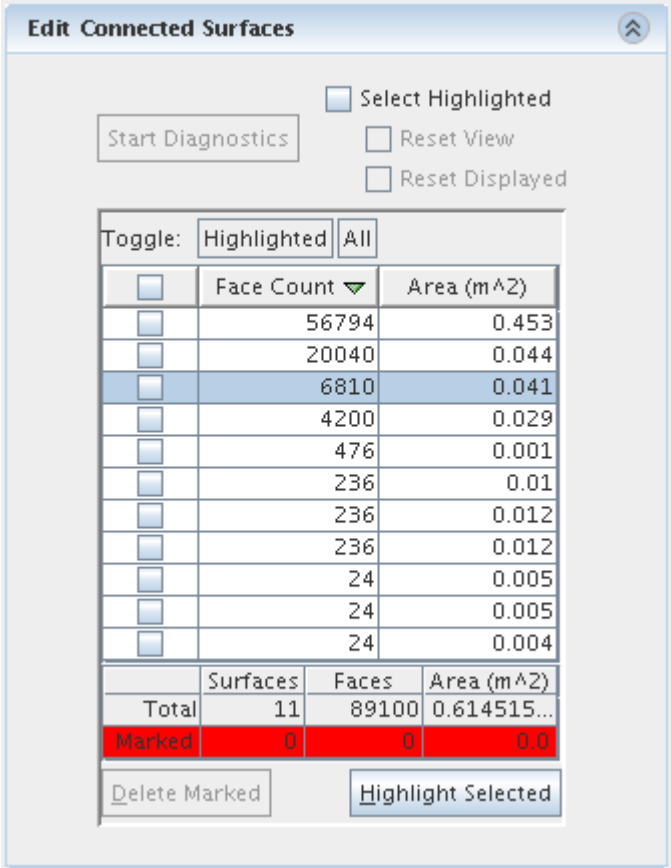
You can use the **Edit Connected Surfaces** tool to identify the table entry or entries that correspond to a selected face or faces.

To highlight a selected face in the summary table, you first generate the connected surface diagnostics. You can then select a face or faces of interest in the *Graphics* window, and click **Highlight Selected**. This option highlights the appropriate entry or entries in the summary table.

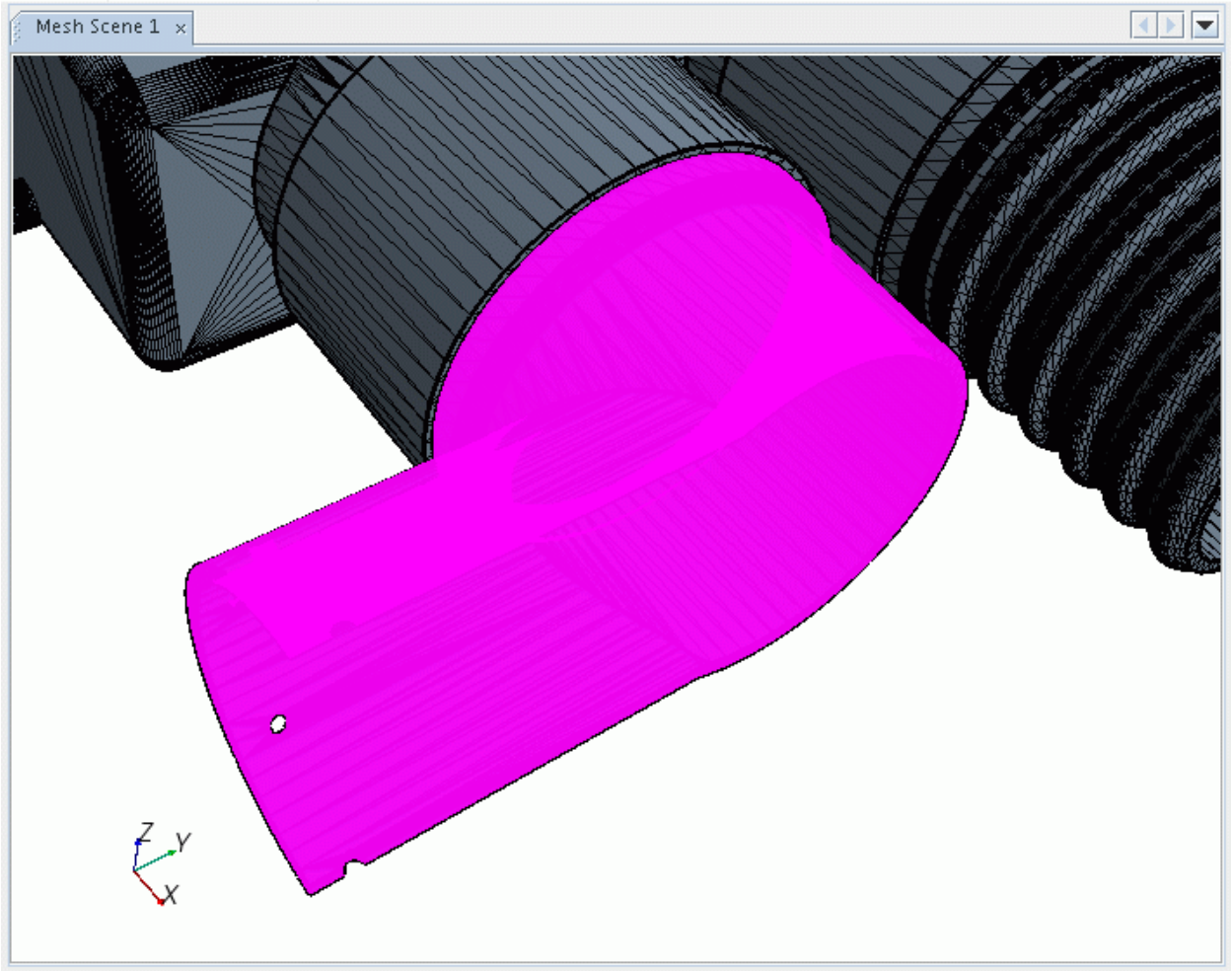
The following example shows a selected face on the part surface.



1. Clicking **Highlight Selected** highlights the table entry, indicating the connected surface group to which the face belongs.



2. If you also activated the **Select Highlighted** option, the entire connected surface would appear highlighted in the *Graphics* window.



Deleting Unwanted Surfaces Using The Summary Table

You can use the *Edit Connected Surfaces* tool to identify and delete unwanted surfaces.

To delete unwanted surface using the summary table, you first generate the connected surface diagnostics. You can then mark surfaces for deletion in the summary table using the following methods:

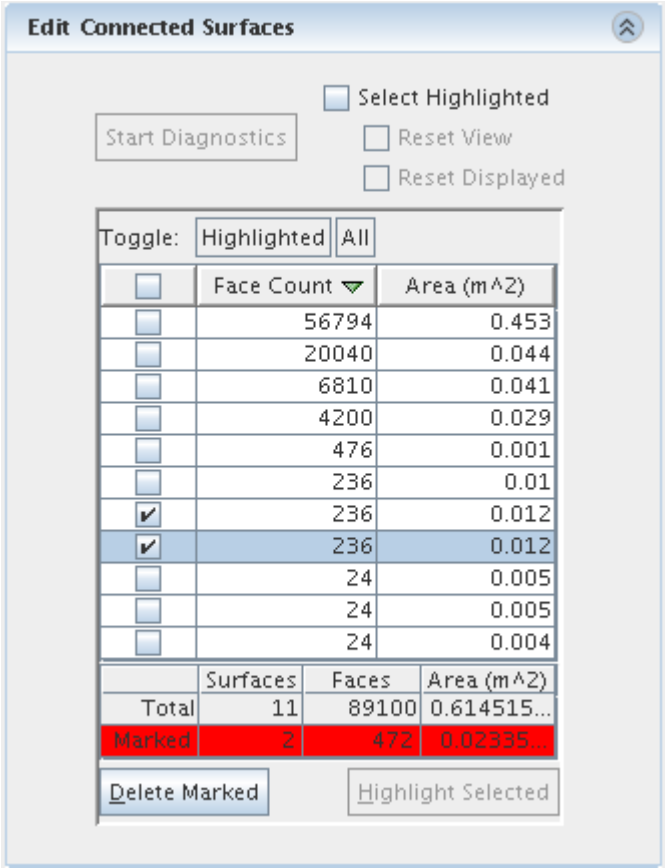
- Tick the appropriate checkbox in the left column
- Click and highlight the required table row (using `<Shift>` and `<Ctrl>` for multiple selections), and click **Highlighted**, next to **Toggle**, at the top of the table

You can also use the **All** button, next to **Toggle**, to toggle the current state of all surfaces. That is, any marked surfaces become unmarked, and any unmarked surfaces become marked. To mark all surfaces for deletion, you can tick the checkbox in the left-most column header, marking and unmarking all surfaces at once.

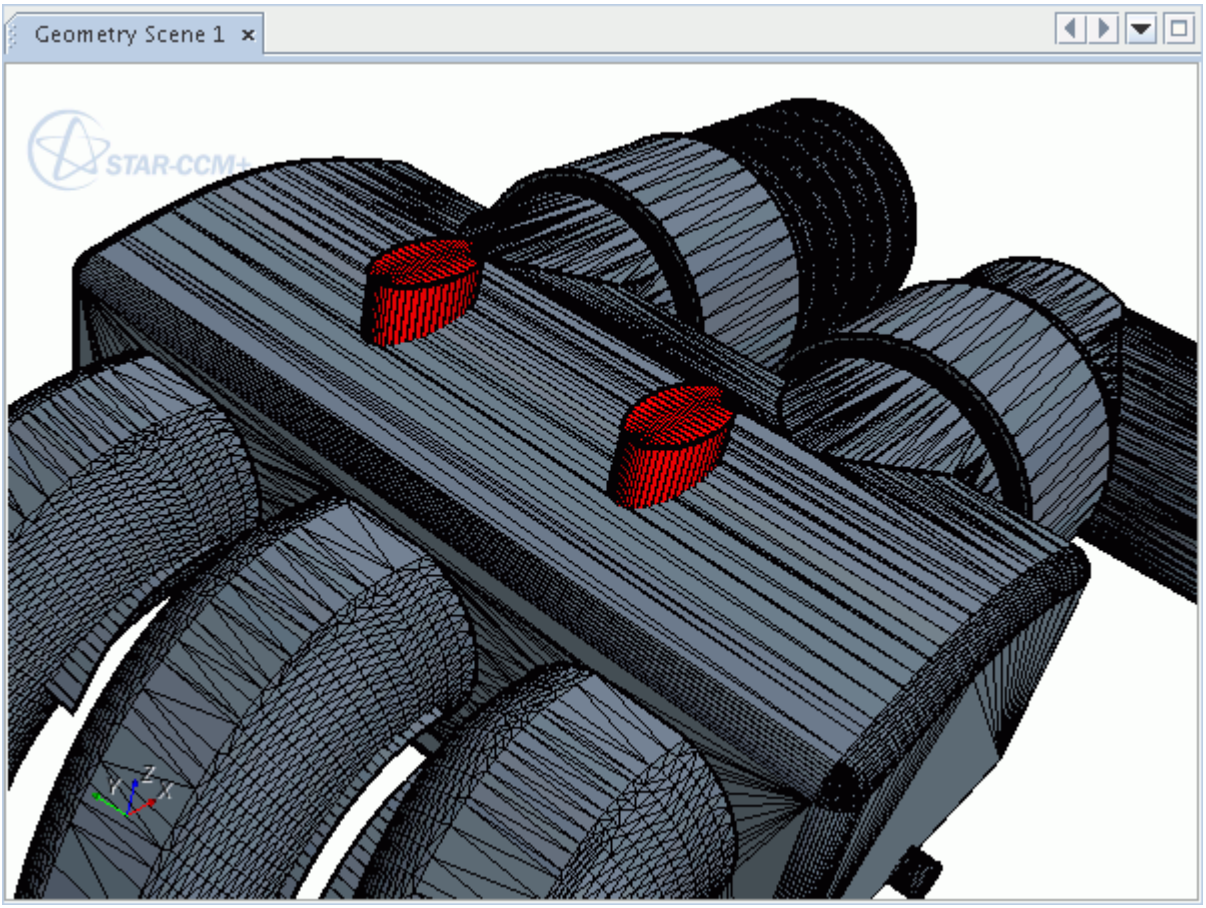
Marking a surface for deletion colors the surface red in the *Graphics* window. The tool also summarizes marked surfaces at the bottom of the group box.

1. Once you have marked all the desired surfaces, click **Delete Marked** to delete the surfaces.
   If you accidentally delete a surface, you can use undo.

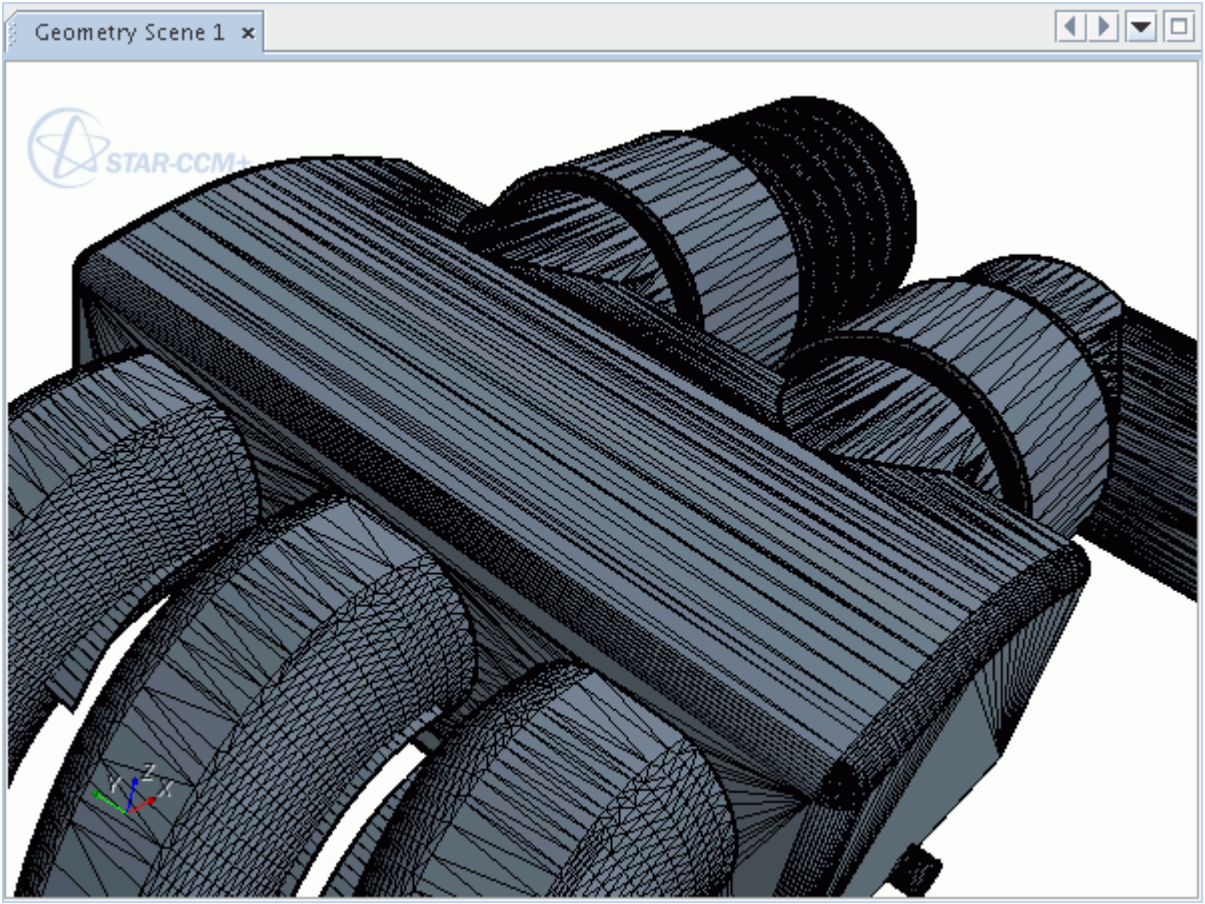
   The following example shows two marked table entries, coloring the appropriate surfaces red.



The marked surfaces appear as follows in the *Graphics* window.



2. Clicking **Delete Marked** deletes the surfaces from the part.



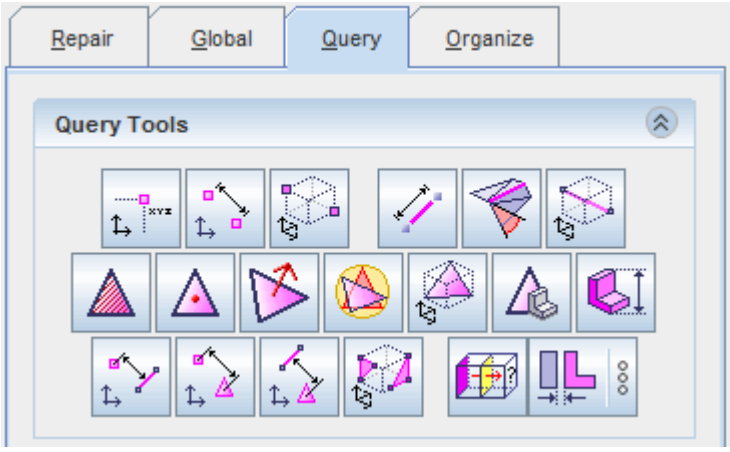
Query Tools

Query tools provide information on selected entities in the *Graphics* window.

To access the Query tools:

1. In the *Surface Repair* panel, click the **Query** tab.

   The Query tools appear.



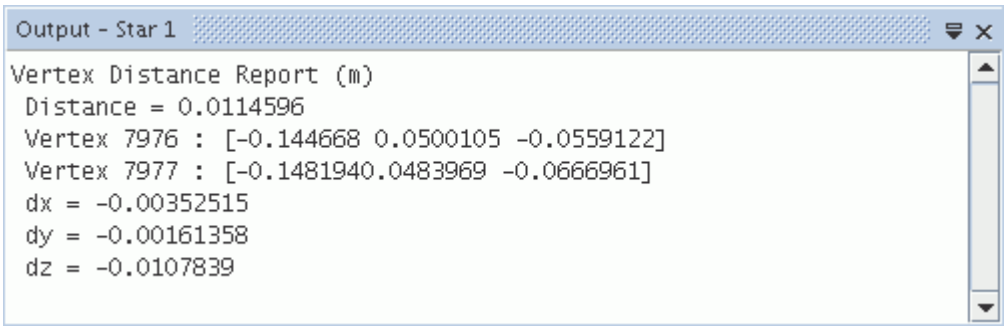
Note: Query diagnostics are also available for region repair, although the output differs. For example, the Part and Part Surface list becomes the Region and Boundary list.

2. Select the entity in the *Graphics* window and click the appropriate tool. See Query Tools Reference.

   The availability of the button depends on the type or combination of selected entities.

   If you have a mixed selection of entities (that is, a combination of faces, edges and vertices), the queries for individual types of entities are still available for selection, as well as the mixed options. In this case, any selections that do not apply to the chosen query are ignored when running the query.

Performing a query prints information to the *Output* window.



The units that are used in the output are the preferred units for the given quantity in the simulation. The preferred units for length and area can be different (for example, length in millimeters and area in square meters).

Contents:

Query Tools Reference
Using the Leak Path Tool

Query Tools Reference

Use the Query tools to get more information about entities in the geometry.

Vertex Query Tools

Vertex query diagnostics provide information specific to the vertex or vertices that are selected in the *Graphics* window.

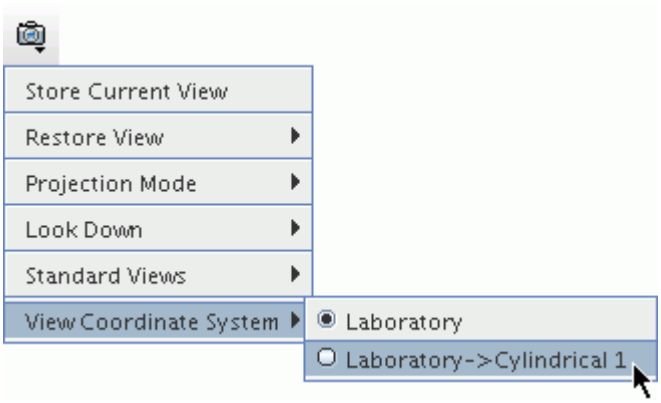
Vertex location(s) in laboratory and local systems

By default, the location is defined relative to the laboratory coordinate system. An example report for a single vertex selection is provided below:

```
Vertex Coordinate Report (vertex 7977)
 Laboratory : [-0.148194 0.0483969 -0.0666961] (m)
```

When selecting multiple vertices, the query reports the list of coordinates and ID's of the selected vertices. There is no correlation between the *Output* list order and the vertices selection order.

You can also choose to report the vertex/vertices location relative to a local coordinate system by setting the coordinate system of the view. One method of changing the view coordinate system is to click the **Save-Restore-Select views** button and select **View Coordinate System**, followed by your choice of system:



In this case, the query reports the vertex/vertices location relative to the selected local coordinate system and the laboratory coordinate system.

```
Vertex Coordinate Report (vertex 7977)
 Laboratory : [-0.148194 0.0483969 -0.0666961] (m)
 Cylindrical 1 : [0.155896 2.82593 -0.0666961] (m,radian,m)
```

Distance Between Two Selected Vertices

This query reports the distance between two vertices. It also reports the vertex IDs, locations, and the distance between the selected vertices in the x-, y-, and z-direction. By default these components are displayed relative to the laboratory coordinate system. However, if the scene view is set to a local coordinate system, the components are displayed in the output window relative to both coordinate systems. This query is available when you select only two vertices.

An example report is provided below:

```
Vertex Distance Report
 Distance = 0.339408687794 (m)
Laboratory (m):
 Vertex 3 : [-0.0563497297057 -0.03 0]
 Vertex 8 : [0.06 0.0779860997883 0.3]
 Components (dx dy dz): (0.116349729706 0.107986099788 0.3)
Caretesian 1 (m):
 Vertex 3 : [0.224019789825 0.00728642763607 0.157484804765]
 Vertex 8 : [-0.0611299627533 0.0256098352269 -0.0256875967843]
 Components (dx dy dz): (-0.285149752578 0.0183234075908
-0.183172401549)
```

Geometric Range of the Selected Vertices

This query reports the minimum and maximum coordinate values (relative to the laboratory coordinate system) of a bounding-box that contains all selected vertices. It also reports the dimensions (change in coordinate value) of the bounding-box in the x-, y-, and z-direction. Select two or more vertices before this query becomes available.

An example report is provided below:

```
Vertex Geometry Range Report (m)
 Min. Global Value : [-0.155686 0.0982237 -0.120845]
 Max. Global Value : [-0.135411 0.1397 -0.0376828]
 Components : dx = 0.0202747
              dy = 0.0414763
              dz = 0.0831617
```

You can view the geometric range using local coordinate systems.

To select a local coordinate system, click (**Save-Restore-Select Views**) and select **View Coordinate System** > **[Local Coordinate System]**. Cylindrical coordinate systems only report radial and axial ranges; angular ranges are not reported. Spherical coordinate systems only report radial ranges.

Edge Query Tools

Edge query diagnostics provide information specific to the edges selected in the *Graphics* window.

Length of the Selected Edge(s)

This query reports the length or total length of an edge or edges. For multiple edges, it also reports the minimum, maximum and average edge length. Select at least one edge before this query becomes available.

If only one edge is selected, the report appears as shown in the example below:

```
Length of edge 5334 is 0.0113075 (m)
```

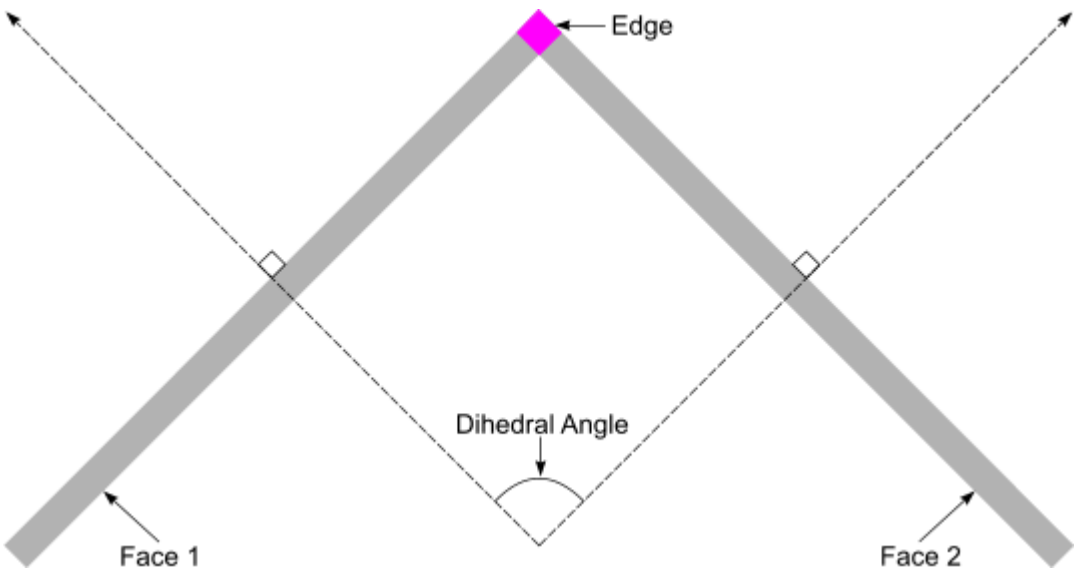
For multiple edges, the report appears as follows:

```
Edge Lenght Report (m)
 Total Edge Length = 0.130005
 Min. Edge Length = 9.72937e-005
 Max. Edge Length = 0.0278163
 Ave. Edge Length = 0.014445
```

Dihedral Angle of the Selected Edge(s)

This query reports the dihedral angle of an edge, which is the angle between its adjacent faces. This query can report the angles for edges with more than two faces connected to it, for example non-manifold edges. For multiple edges, it reports the minimum, maximum and average dihedral angle, and the number of valid edges. Edges are considered invalid if they are free, as a dihedral angle cannot be calculated for such edges. Select at least one edge before this query becomes available.

When using Edge Dihedral Angle as a Threshold Diagnostic, manifold and non-manifold edges are selected according to the angle between the face normals of the connected faces of the edge. For non-manifold edges, the nearest neighbor faces are used for each angle calculation.



If only one edge is selected, the report appears as shown in the example below:

```
Dihedral angle of edge 3175 is 0.0653701 (radian)
```

For multiple edges, the report appears as follows:

```
Edge Dihedral Angle Report (radian)
 No. of valid edges = 5
 Min. Angle = 1.49012e-008
 Max. Angle = 3.0761
 Ave. Angle = 0.637921
```

Geometric Range of the Selected Edge(s)

This query reports the minimum and maximum coordinate values (relative to the laboratory coordinate system) of a bounding-box that contains all selected edges. It also reports the dimensions (change in coordinate value) of the bounding-box in the x-, y-, and z-direction. Select at least one edge before this query becomes available.

An example report is provided below:

```
Edge Geometric Range Report (m)
 Min. Global Value : [0.0093659 -0.03175 -0.0404585]
 Max. Global Value : [0.1905 0.243205 0.0345574]
 Components : dx = 0.181134
              dy = 0.274955
              dz = 0.0750159
```

You can view the geometric range using local coordinate systems.

To select a local coordinate system, click (**Save-Restore-Select Views**) and select **View Coordinate System** > **[Local Coordinate System]**. Cylindrical coordinate systems only report radial and axial ranges; angular ranges are not reported. Spherical coordinate systems only report radial ranges.

Face Query Tools

Face query diagnostics provide information specific to the faces selected in the *Graphics* window.

Area of the Selected Faces

This query reports the area of a face or group of faces. For multiple faces, it also reports the minimum, maximum and average face area. Select at least one face before this query becomes available.

If only one face is selected, the report appears as shown in the example below:

```
Face area for face 10325 is 0.000140443 (m^2)
```

For multiple faces, the report appears as follows:

```
Face Area Report (m^2)
 Total Face Area = 0.0887778
 Min. Face Area = 3.11258e-009
 Max. Face Area = 0.00576306
 Ave. Face Area = 1.99995e-005
```

Centroid of the Selected Face

This query reports the location of the face centroid relative to the laboratory coordinate system. It also reports the face ID. This query is available when you select only one face.

An example report is provided below:

```
Face centroid location for face 19874 is (0.00137158 0.003175
-030175941) (m)
```

Normal of the Selected Faces

This query reports the face normal for a single selected face. This query is available when you select only one face.

The report appears as shown in the example below:

```
Face Normal Report (face 337)
 Face normal : [-0.000150901232119 0.000577854633815 0] (m^2)
 Normalized : [-0.252667304981 0.967553219722 0]
```

Quality of the Selected Faces

This query reports the face quality of the selected faces. For multiple faces, it also reports the minimum, maximum and average face quality. Select at least one face before this query becomes available.

If only one face is selected, the report appears as shown in the example below:

```
Face quality for face 44206 is 0.972888
```

For multiple faces, the report appears as follows:

```
Face Quality Report
 Min. Face Quality = 0.40796
 Max. Face Quality = 1
 Ave. Face Quality = 0.940224
```

Geometric Range of the Selected Faces

This query reports the minimum and maximum coordinate values (relative to the laboratory coordinate system) of a bounding-box that contains all selected faces. It also reports the dimensions (change in coordinate value) of the bounding-box in the x-, y-, and z-direction. Select at least one face before this query becomes available.

An example report is provided below:

```
Face Geometric Range Report
 Min. Global Value : [-0.060325 0.003175 -0.047625]
 Max. Global Value : [0.060325 0.250825 0.0349248]
 Components : dx = 0.12065
              dy = 0.24765
              dz = 0.0825498
```

You can view the geometric range using local coordinate systems.

To select a local coordinate system, click (**Save-Restore-Select Views**) and select **View Coordinate System** > **[Local Coordinate System]**. Cylindrical coordinate systems only report radial and axial ranges; angular ranges are not reported. Spherical coordinate systems only report radial ranges.

Part and Part Surface List for Selected Faces

This query reports the associated part and part surfaces for the selected faces. It also reports the number of selected faces on a given part surface from the total. To activate this query, select at least one face.

The following image shows an example of this report:

```
Part and Part Surface List Report
 Part: Main body -> Part Surface: "Loft" : (41 faces from 1015
selected)
 Part: Main body -> Part Surface: "Pipe" : (2496 faces from 2496
selected)
 Part: Main body -> Part Surface: "Outlet" : (9 faces from 14
```

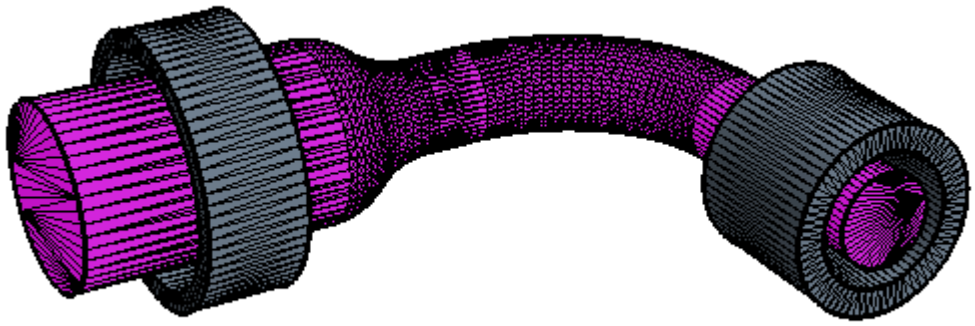
```
selected)
 Part: Sleeve -> Part Surface: "Default" : (36 faces from 376
selected)
 Part: Body 4 -> Part Surface: "Default" : (1 face from 12 selected)
 Part: Body 5 -> Part Surface: "Default" : (2 faces from 12 selected)
```

If any of selected faces belong to a part contact, the query lists the part surface contacts at the end of the report:

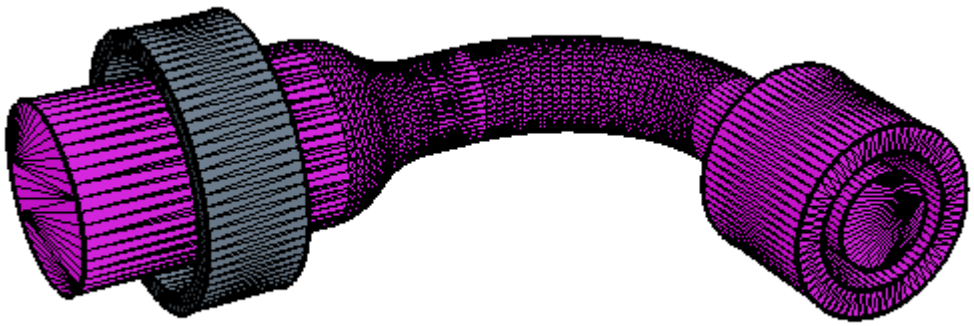
```
Part and Part Surface List Report
 Part: Body 1 -> Part Surface: "BoxSurface" : (4 faces selected from
10)
 Part: Body 1 -> Part Surface: "BoxContactFace" : (121 faces
selected from 121)
 Part: Body 2 -> Part Surface: "TubeInlet" : (63 faces selected from
63)
 Part: Body 1 -> Part Surface Contact: "Box" : (1 face selected from
115)
 Part: Body 2 -> Part Surface Contact: "Tube" : (1 face selected
from 115)
```

Volume of the Selected Faces

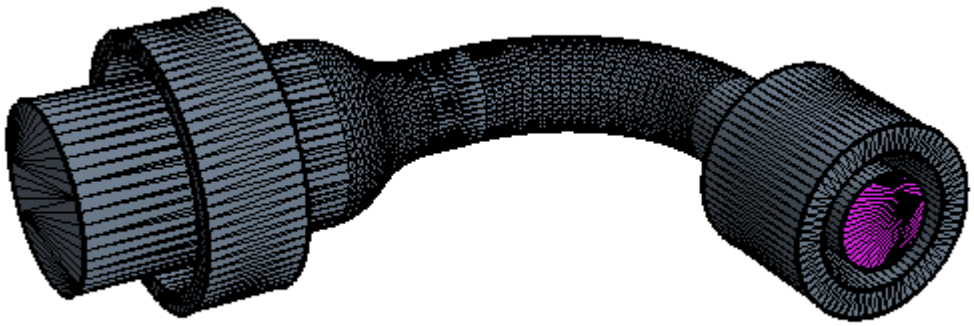
Calculates the volume that is enclosed by a set of faces. All faces must form a closed and manifold shape. Knowing the volume of a shape is useful when calculating physical properties.



If you select more than one set of manifold faces, the total volume is calculated.



If you select a set of faces that are not manifold, as shown in the following example, an error message is displayed in the *Output* window.



You can include contact faces in the selection; however, if you want to calculate the volume of two bodies that have a contact between them, do not include these faces in your selection.

You can calculate the volume where one body is encased inside another body. In this situation, the reported volume is the difference between the two bodies. If the two bodies are intersecting, subtract, intersect, or unite them before using this tool.

Mixed Query Tools

Mixed query diagnostics provide information specific to the combination of entities that are selected in the *Graphics* window.

Distance Between Selected Vertex and Edges

This query finds which of the selected edges is closest to the selected vertex and reports the shortest distance between them. It also reports the closest edge ID, the vertex ID and location, and the maximum

distance from the vertex to the edges in the x-, y-, and z-direction, relative to the laboratory coordinate system. This query becomes available when you select only one vertex and at least one edge.

An example report is provided below:

```
Vertex-Edge Distance Report (m)
 Shortest distance = 0.0143231
 Source Vertex 21672 : [-0.138287 0.0417277 -0.0405851]
 Closest Edge 3857
 dx = 0.00102346
 dy = -0.014132
 dz = 0.00209533
```

Distance Between Selected Vertex and Faces

This query finds which of the selected faces is closest to the selected vertex and reports the shortest distance between them. It also reports the closest face ID, the vertex ID and location, and the maximum distance from the vertex to the faces in the x-, y-, and z-direction, relative to the laboratory coordinate system. The shortest distance is measured between the vertex and the closest point on the closest face (that is, not necessarily the closest vertex). Select only one vertex and at least one face before this query becomes available.

An example report is provided below:

```
Vertex-Face Distance Report (m)
 Shortest distance = 0.0471872
 Source Vertex 15183 : [-0.0196444 0.0310554 0.0333477]
 Closest Face 3774
 dx = -0.0406806
 dy = 0.00704464
 dz = -0.0228494
```

Distance Between Selected Edge and Faces

This query finds which of the selected faces is closest to the selected edge and reports the shortest distance between them. It also reports the closest face ID, the closest edge vertex ID and location, and the maximum distance from the edge to the faces in the x-, y-, and z-direction, relative to the laboratory coordinate system. The shortest distance is measured between the closest vertex on the edge and the closest point on the closest face (that is, not necessarily the closest face vertex). Select only one edge and at least one face before this query becomes available.

An example report is provided below:

```
Edge-Face Distance Report (m)
 Shortes distance = 0.127133
 Source Edge Vertex 7857 : [-0.15575 0.0320246 -0.127]
 Closest Face 60911
 dx = 0.097494
 dy = 2.20414e-006
 dz = 0.0815947
```

Geometric Range of Selected Faces/Edges/Vertices

This query reports the minimum and maximum coordinate values (relative to the laboratory coordinate system) of a bounding-box that contains all selected faces, edges, and vertices. It also reports the dimensions (change in coordinate value) of the bounding-box in the x-, y-, and z-direction. To make this query available, make one of the following selections:

- At least one face and at least one vertex
- At least one face and at least one edge
- At least one vertex and at least one edge

An example report is provided below:

```
Mixed Entity Geometric Range Report (m)
 Min. Global Value : [-0.148336 -0.03175 -0.0827226]
 Max. Global Value : [0.1905 0.139436 0.0348818]
 Components : dx = 0.338836
              dy = 0.171186
              dz = 0.117604
```

You can view the geometric range using local coordinate systems.

To select a local coordinate system, click (**Save-Restore-Select Views**) and select **View Coordinate System** > **[Local Coordinate System]**. Cylindrical coordinate systems only report radial and axial ranges; angular ranges are not reported. Spherical coordinate systems only report radial ranges.

Find Leak Path

This tool selects faces using the shortest path between one source and multiple target faces. See Using the Leak Path Mode.

Tolerance from the Selected Faces to Neighboring Parts

This query identifies neighboring parts that are within a given distance of a specified part face. The neighboring parts can be a set distance away, touching, or intersecting the specified face. This tool is available when you select at least one face. The selection must not span multiple parts.

When you run the query, a report is generated and lists the neighboring parts in alphabetical order. The closest distance of the neighboring parts to the selected part are also displayed:

```
Part Tolerance Report (m)
 Input part: Differential 1
 Search distance: 0.001
Found 8 parts within search distance:
 Part: Break 20 Closest distance: 0
 Part: Break 22 Closest distance: 0
 Part: Break 30 Closest distance: 0
 Part: Differential 1 Closest distance: 0.000785971
```

```
Part: Steering Oil Tank Closest distance: 0
Part: Transmission Rod Closest distance: 4.29961e-06
```

You can also choose to select the parts and faces that fall within the search distance that you set. This feature is useful for post-processing or inspecting geometry, such as when preparing to create a multi-part contact.

The original selection set can include contacts for the part and can also be a subset of the part. You do not need to select all the faces, so a single part surface can be selected, if preferable, and the output will only reflect results that are within the same tolerance as the original selection.

Query Part Tolerance Options

Click to the right of the Query Part Tolerance tool to display the options:

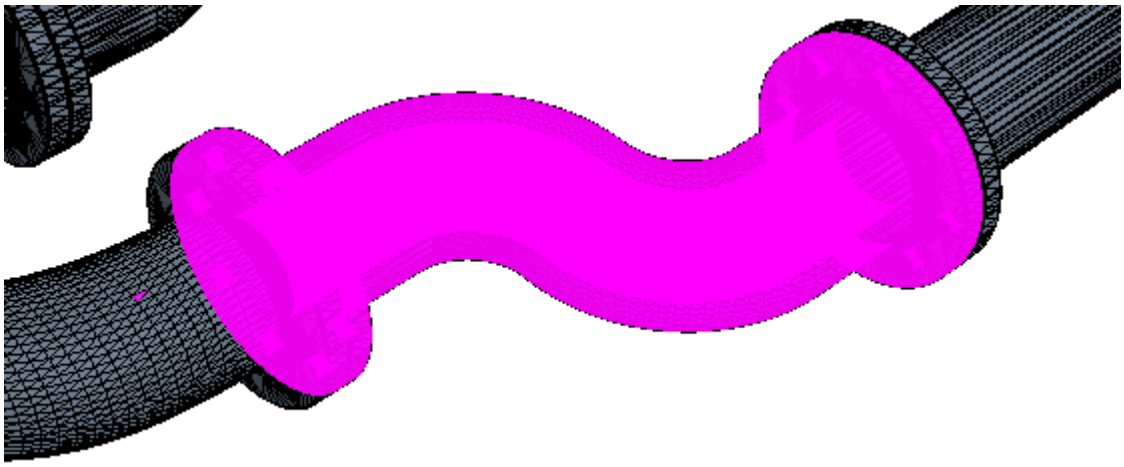
Search Distance

Specifies the maximum distance from the selected surface within which to search for neighboring parts. Parts that are within this search distance are found and reported in the *Output* window. If the **Update Selection** option is activated, the neighboring parts/surfaces are added to the selection set.

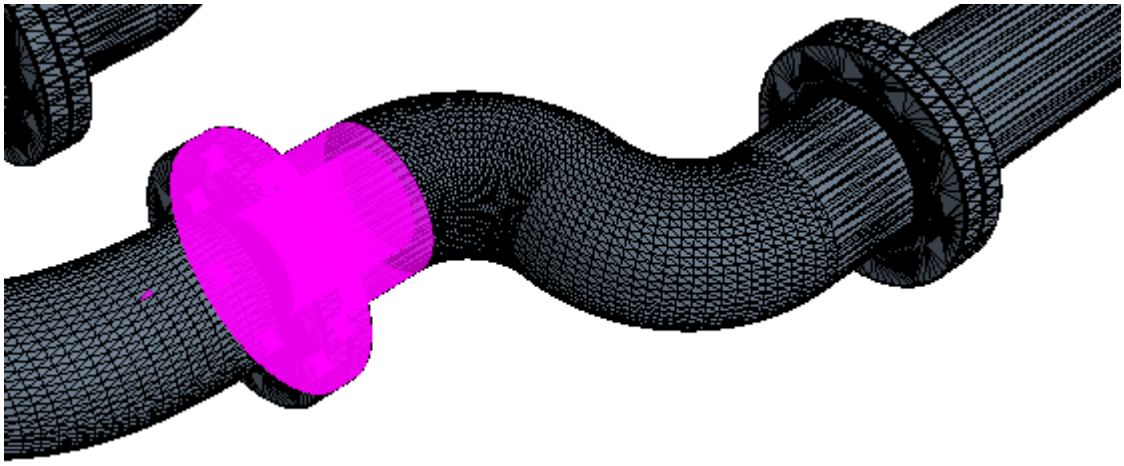
Update Selection

When activated, this option selects the parts/faces that are within the search distance. This option is activated by default. The option you choose in the *Selection Options* group box controls what is selected.

Select parts selects the entire part that contains a face that falls within the search distance.



Select in-tolerance faces selects only the faces that fall within the search distance. The parts or faces are added to the existing selection set, in addition to the original faces that you originally select.



Deactivate the **Select in-tolerance faces** option to display the results of the query in the *Output* window, without selecting any parts or faces.

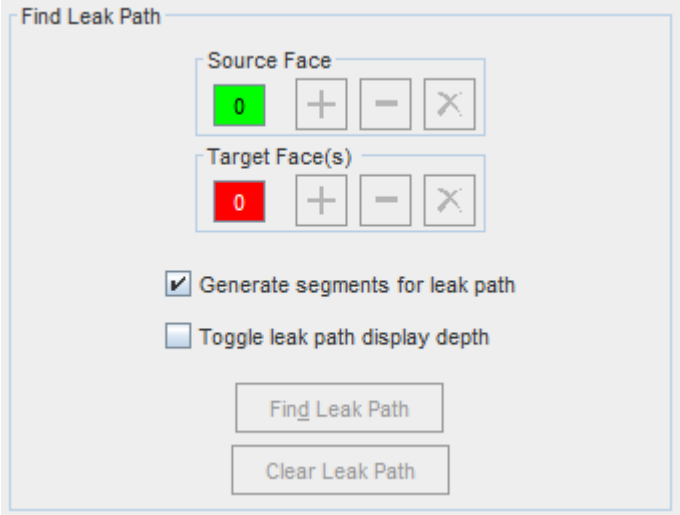
Using the Leak Path Tool

The Leak Path Tool identifies leaks in the model. A leak is a gap or a hole that would allow a path to be traced from one volume through to another volume or the outside without crossing through a surface. A fully enclosed surface has no leaks.

This tool displays the shortest path between one source and multiple target faces. The path can be displayed as a segment-based path or as selected faces. This tool is available when you select the **Faces** filter.

To identify leaks in the model:

1. Click (**Toggle find leak path mode**).
   The leak path options are displayed.



2. In the *Graphics* window, select a single source face from the displayed surfaces and click (**Add**) within the *Source Face* group box.

   Note: You cannot assign contact or interface faces as a source face.

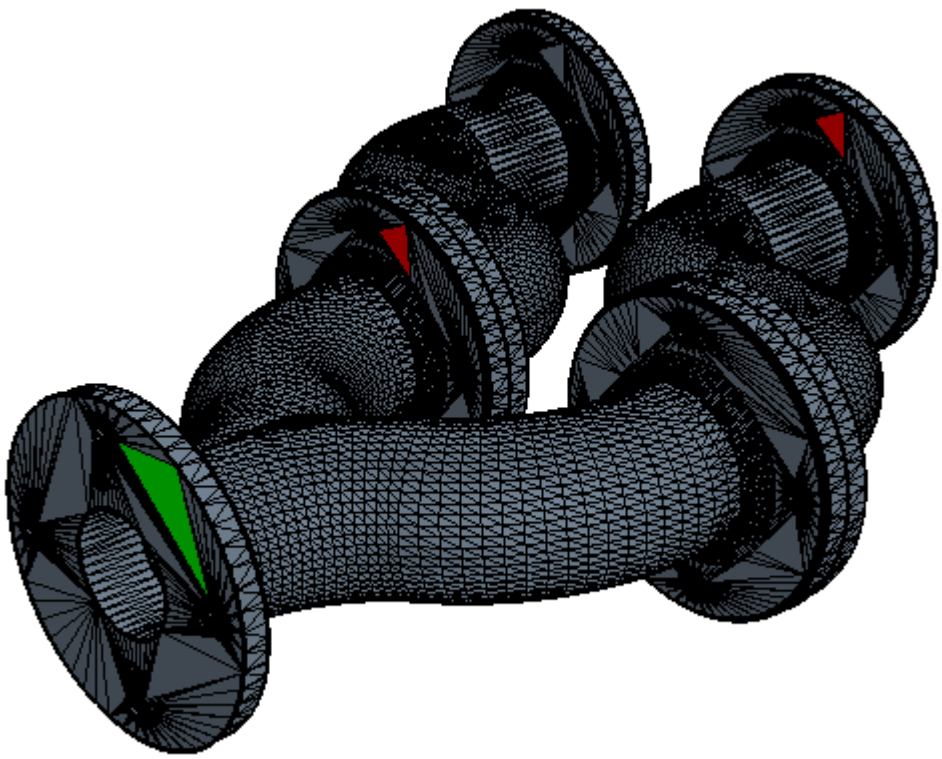
   The source face is marked in a green color.

3. Select one or more target faces from the displayed surfaces in the *Graphics* window and click (**Add**) within the *Target Face(s)* group box.

   Note: You cannot assign contact or interface faces as target faces.

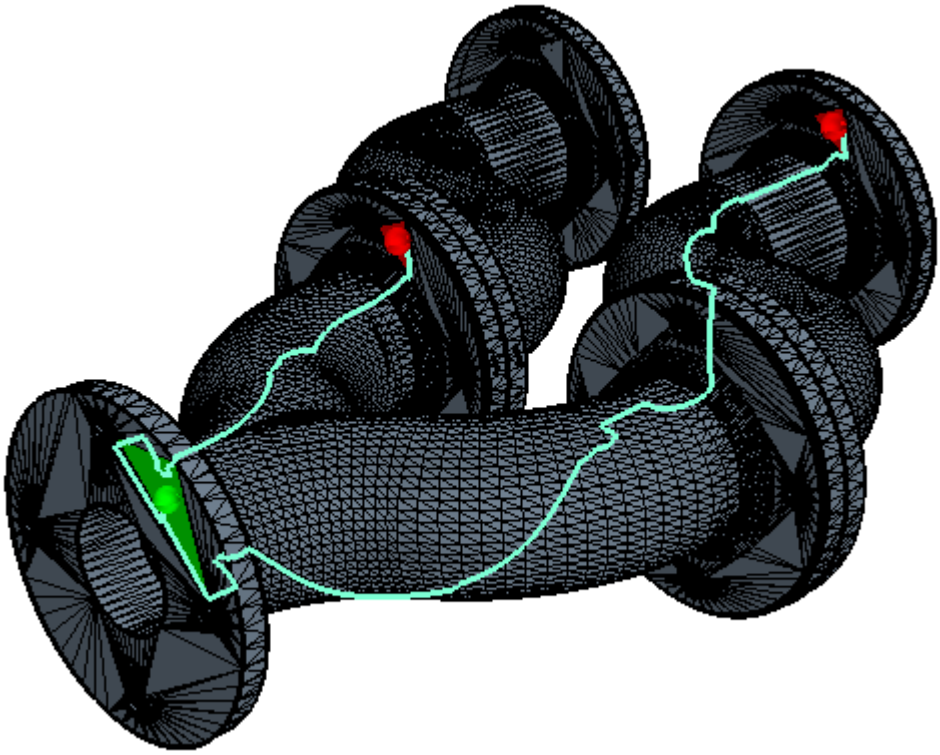
   To deselect faces from the *Source* or *Target*, select the desired faces in the *Graphics* window and click (**Remove selected faces from the merge set**) in the appropriate group box. Similarly, to deselect all faces from the *Source* or *Target*, click (**Clear all faces from the merge set**) in the appropriate group box.

   The target faces are marked in a red color.



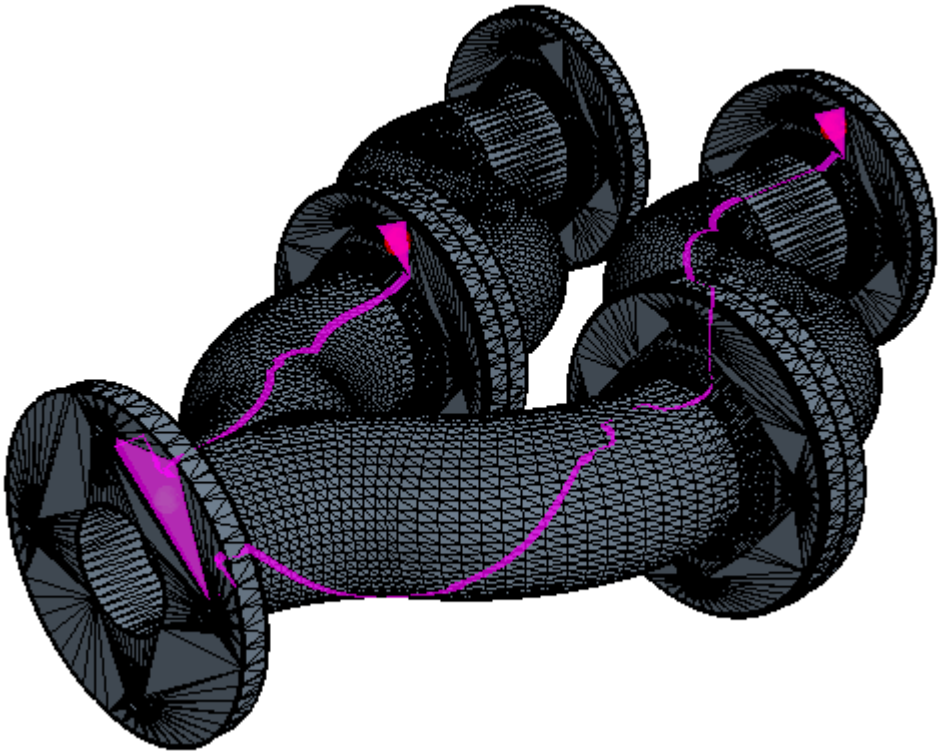
4. Click **Find Leak Path** to perform leak path detection.

The tool displays a segment-based path between the *Source Face* and the *Target Face(s)*. The path lies on the faces that form the shortest distance between the source and target faces.



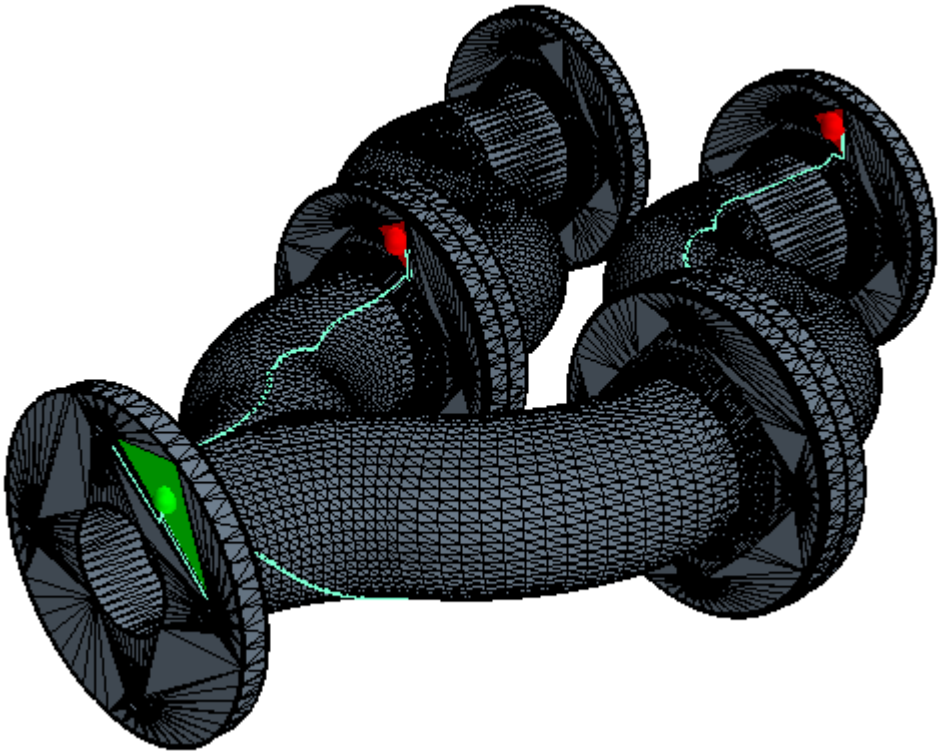
The segment path is generated based on the edge centroid of the leak path faces. A colored sphere is included at the centroid of the source and target faces so that you can still identify these faces when you close the Find Leak Path tool. The color of the leak path is different from the color map that is used for the rest of surface repair to color diagnostics and part surfaces. This different color minimizes color conflicts when viewing the segment path and other coloring options within repair at the same time.

Deactivating the **Generate segments for leak path** option and recalculating the leak path highlights the faces for the leak path. The face-based leak path is identical to the segment-based path.

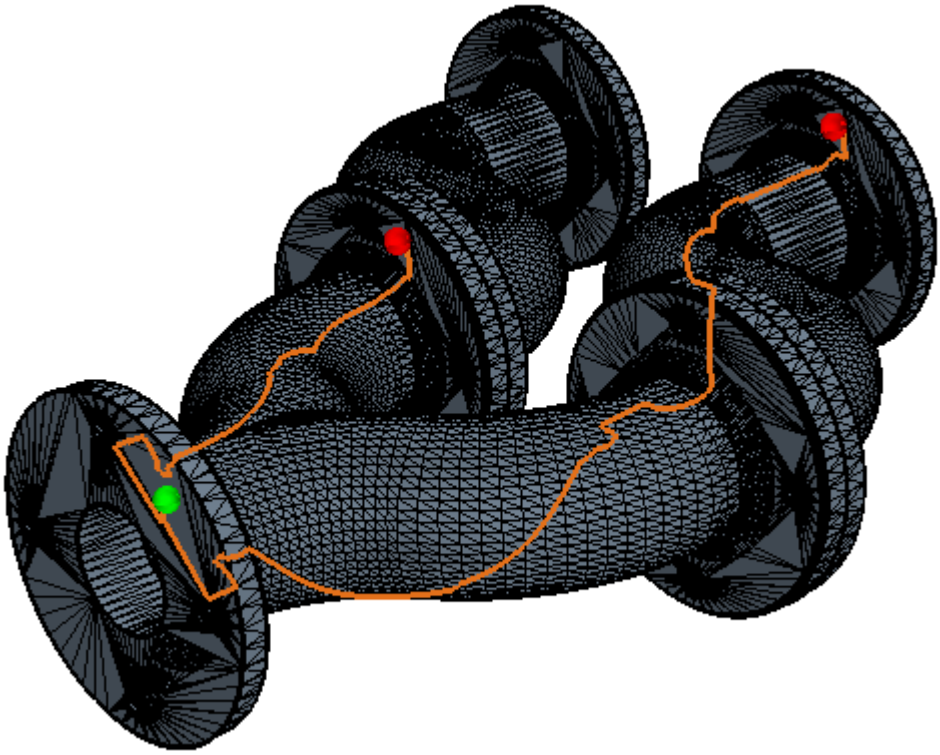


The segment-based leak path is always displayed, even if the mesh topology or geometry is changed (for example, as part of a repair operation), or if you close the **Find Leak Path** tool. To clear the path, click **Clear Leak Path**.

The **Toggle leak path display depth** option controls how the segment leak path is displayed relative to the geometry. By default, this option is deactivated, meaning that the path is superimposed on the mesh (no depth control). Activate this option to redraw the path with depth control. Some or all of the path can be hidden from view. It may be easier to visualize the exact path with depth control on, depending on the complexity of the geometry and location of the leak.



If a geometry or topology change is made to the surface mesh, the segment path color changes to orange. This change indicates that the path could potentially be out of date, depending on the change made. Click **Find Leak Path** to update the path.



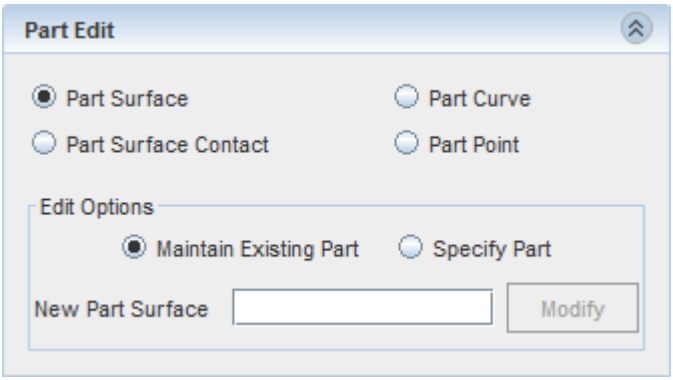
See Also:

Changing the Parts in the Active Model

Part Controls

The *Part Edit* group box, located under the *Organize* tab, assigns faces to an existing or new part surface/surface contact, feature edges to an existing or new part curve, and feature vertices to an existing or new part point.

The **Part Surface Contact Edit** tool is only available when using Parts-Based Meshing. There is no equivalent option when repairing regions.



Note: To work with vertices and points, you must enable part points at the parts level in the simulation tree before you launch the Surface Repair tool. To enable part points, in the simulation tree, right-click the **Parts** node and select **Enable Part Points**.

All the functionality of the *Part Edit* tool are available as right-click options when you make a suitable selection.

The surface and feature diagnostics update following any action that is performed with the *Part Edit* controls. Review the updated diagnostics to identify and fix any errors that the assign action causes, before closing the *Surface Preparation tool*.

You can reverse the assigning of faces or curves using the Undo button. However, this option does not apply to the creation of part surfaces, curves or points, which you must manually delete from the simulation tree. You can move faces into and out of part contacts, however, you cannot create new part contacts within this tool. To create a part contact, return to the simulation tree or use the merge/imprint tool. See Creating Part Contacts Manually and The Merge and Imprint Tool.

Note: If you break a part surface contact by moving one of the contacting faces to another surface, both contacting faces move together. This creates free edges on the remaining part.

Contents:

Creating a New Part Entity During Surface Repair
Assigning Part Entities to Target Part Entities
Assigning Faces to a Target Part Surface Contact
Modifying Part Entities Using the Right-Click Menu

See Also:

Changing the Parts in the Active Model
Using Part Contact Data to Create Interfaces

Creating a New Part Entity During Surface Repair

You can create a new part entity (surface, curve, or point) within the *Surface Repair* tool.

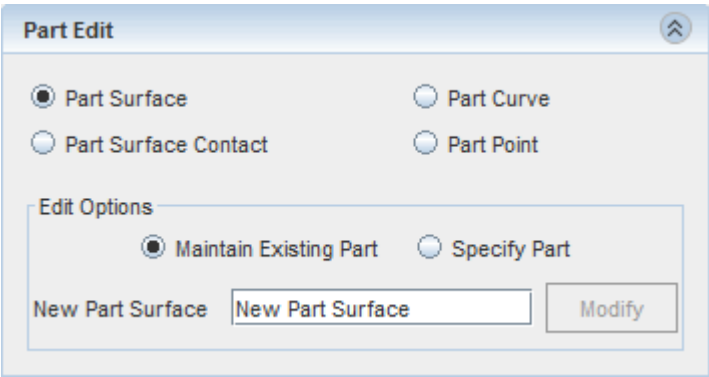
To create a new part entity:

1. In the *Graphics* window, select the part entities using the appropriate selection filter.

   Part curves and part points must be marked as features before you can modify them.
2. In the *Part Edit* group box, select the option appropriate for the part entity that you want to modify.

   For example, if you want to modify a part surface, select **Part Surface** in the *Part Edit* group box.
3. In the *Edit Options* group box:
   - To create a new part entity on the existing part:
     1. Select **Maintain Existing Part**.
     2. Enter the desired name for the new surface/curve/point in the **New Part Surface/Curve/Point** field.

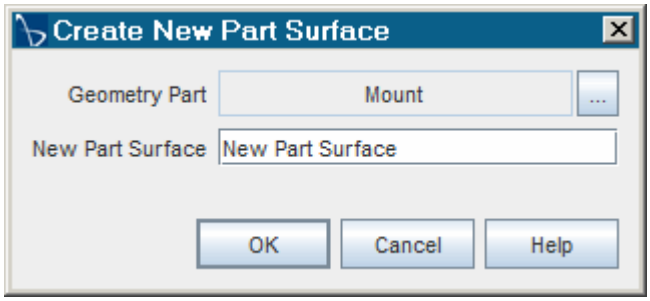
        In this example, a new part surface is named **New Part Surface**.



     This option allows you to modify part surfaces, part curves, or part points to a new part surface/part curve/part point across more than one part in one action. For example, given a selection of faces that span different parts, you can modify the selected faces to a new part surface within each original part using a single operation. This feature is useful in situations where you have multiple parts that require the same type of modification performed on them.
   - To create a new part entity on a different part:
     1. Select **Specify Part**.
     2. Click **New**.
     3. In the *Create New Part Surface* dialog:
        a. Set **Geometry Part** to the part that you want the new surface, curve, or point to belong to.

           By default, the parent part of the selection is chosen. In this example, the parent part is **Mount**.
        b. Enter the desired name for the new surface/curve/point in the **New Part Surface/Curve/Point** field.

           In this example, a new part surface is named **New Part Surface**.

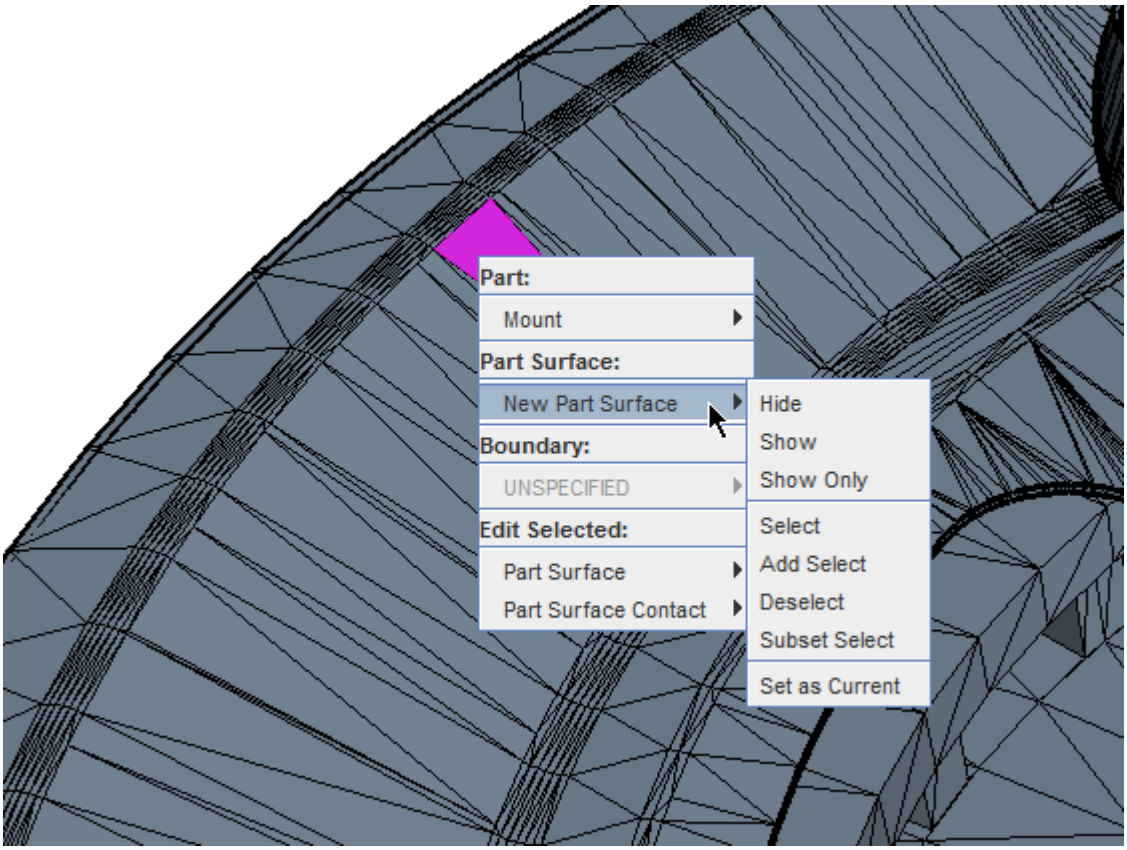


c. Click **OK**.

4. In the *Part Edit* group box, click **Modify**.

A new part surface, curve, or point is created added to the specified geometry part. A message indicating the modification is displayed in the *Output* window.

To check the modification, right-click on a modified part surface, curve, or point and check the menu entry:



Assigning Part Entities to Target Part Entities

All part entities (faces, edges, and vertices) in a part belong to a part surface/curve/point by default. This tool assigns selected part entities to an existing surface/curve/point.

To assign faces/edges/vertices to a target surface/curve/point:

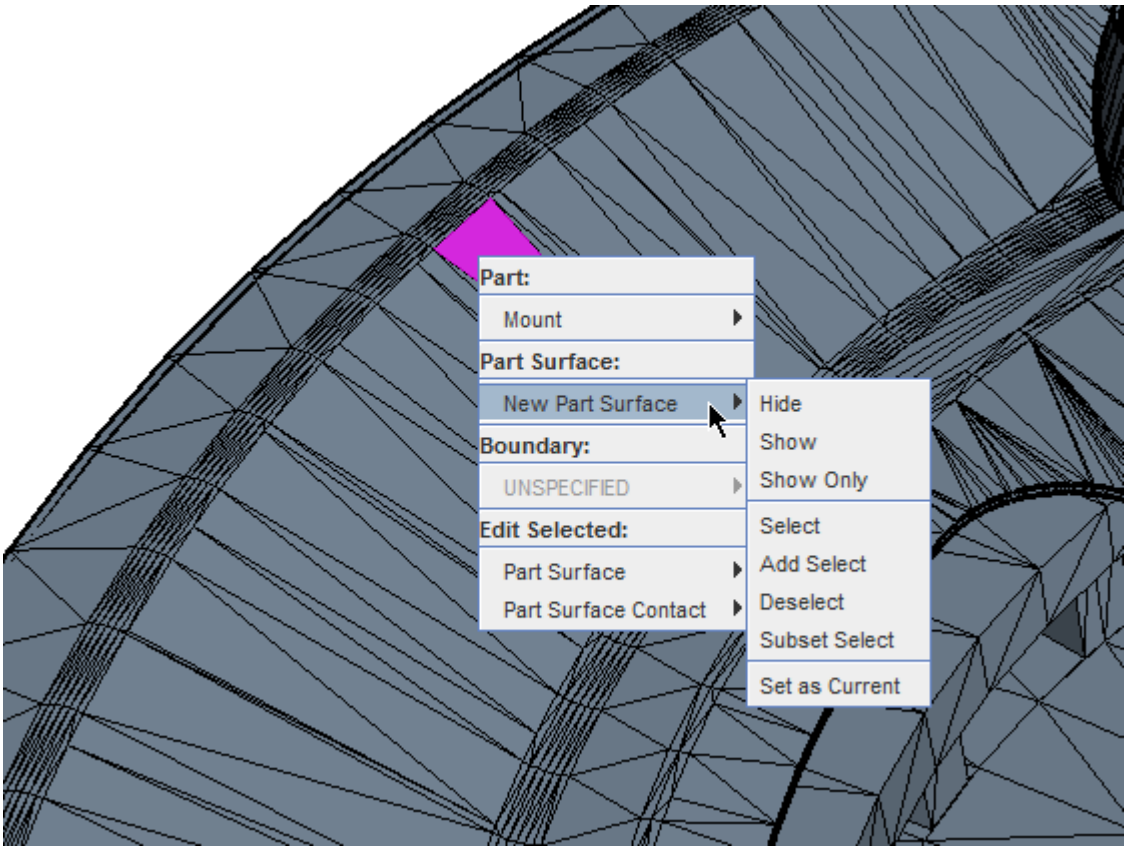
1. In the *Graphics* window, select the part entities using the appropriate selection filter.

   Part curves and part points must be marked as features before you can modify them.
2. In the *Part Edit* group box, select the option appropriate for the part entity that you want to modify.

   For example, if you want to modify a part surface, select **Part Surface** in the *Part Edit* group box.
3. In the *Edit Options* group box:
   a) Select **Specify Part**.
   b) Click ...
4. In the *Select Object* dialog:
   a) Expand the part folder and the **Surfaces/Curves/Points** folder.
   b) Select the target surface, curve, or point that you want to add the selection to.
   c) Click **OK**.
5. In the *Part Edit* group box, click **Modify**.

The selected surfaces, curves, or points are added to the target part surface, curve, or point. A message indicating the modification is displayed in the *Output* window.

To verify that modification is successful, right-click on a modified part surface, curve, or point and check the menu entry:



See Also:

Pop Up Menu of the Surface Repair Tool Graphics Window

Assigning Faces to a Target Part Surface Contact

Within the *Part Edit* group box, you can use the **Part Surface Contact** option to assign faces to existing part surface contacts. This feature is useful in situations where you use surface repair tools for creating faces that belong to a contact.

1. In the *Graphics* window, select the faces which you want to add to the target part surface contact using the appropriate selection filter.
2. In the *Part Edit* group box, select **Part Surface Contact**.

   Note: Only contacts that exist between the parts that are in the current surface repair mode are available. You cannot create a new contact using this tool. To create a new contact, return to the simulation tree. See Creating Part Contacts Manually.

3. Click ... and select the target part surface contact from the **Select Object** dialog.
4. In the *Part Edit* group box, click **Modify**.

The selected faces are added to the target part surface contact. However:

- If the selected faces belong to a contact that is different from the target contact, the following warning is displayed in the *Output* window:

  ```
  Warning -- some faces were moved out of part contacts.
  ```

- The faces that are added to the target part surface contact using this tool are assigned to a separate patch from those faces that already belong to the target part surface contact.

Modifying Part Entities Using the Right-Click Menu

In addition to using the *Part Edit* group box to modify part entities, you can modify part entities directly using the right-click menu.

To modify part entities using the right-click menu:

1. In the scene, select the part entities using the appropriate selection filter.

   Part curves and part points must be marked as features before you can modify them.
2. Right-click the selection and choose the appropriate option:
   - To create a new part entity from the selected entities:
     1. Select **Part Surface/Curve/Point** > **Modify to New**.
     2. In the *Modify Selection to New Part Surface* dialog, choose to either create a new part entity in the existing part (**Maintain Existing Part** option), or in a specified part (**Specify Part** option). When choosing to specify a part, set **Geometry Part** to the part that you want the new surface, curve, or point to belong to. By default, the parent part of the selection is chosen. In this example, the parent part is **Mount**.
     3. Set **New Part Surface/Curve/Point** to the name of the new surface/curve/point.
     4. Click **OK**.

See Creating a New Part Entity During Surface Repair for an alternative method of creating new part entities.

- To move the selected part entities to a target entity:
  1. Select **Part Surface/Curve/Point** > **Modify to Existing**.
  2. In the *Modify Selection to Existing Part Surface* dialog, expand the part folder and the **Surfaces/Curves/Points** folder.
  3. Select the target surface, curve, or point that you want to add the selection to.
  4. Click **OK**.

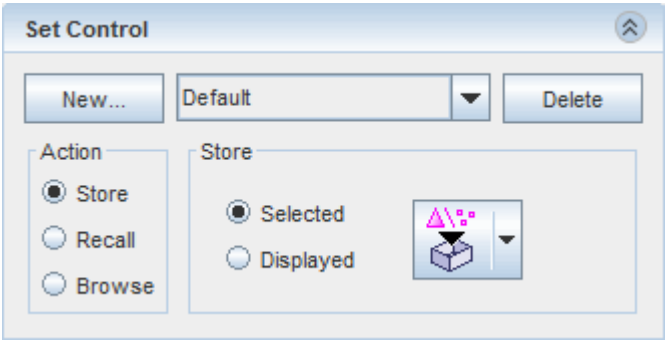
  See Assigning Part Entities to a Target Part Entity for an alternative method of moving part entities to a target entity.

Right-clicking on a baffle contact surface displays options differently from a non-baffle contact because a baffle contact is applied to a single part surface belonging to a single part.

Set Controls

A set is a collection of faces, edges, or vertices, which you can store, recall and browse, using the *Set Controls* group box.

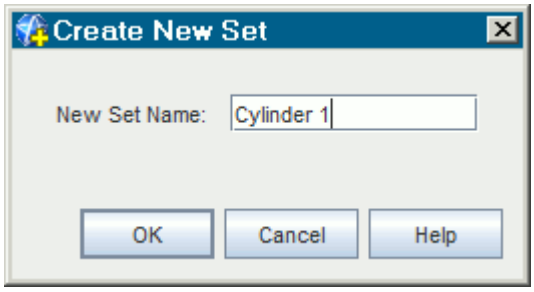
This tool is useful when working with various sections of a tessellated surface, as it allows you to retrieve a set of entities easily without the need for reselection.



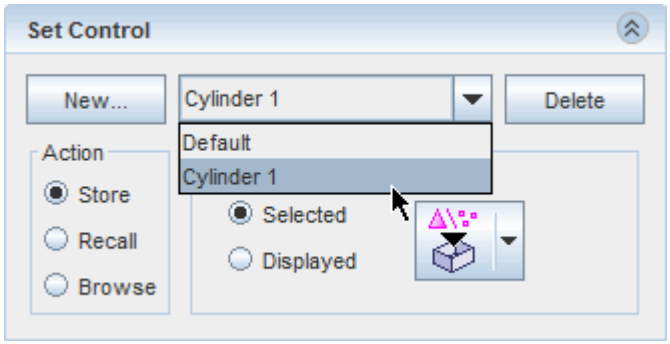
Closing the *Surface Repair* tool discards all currently stored sets (that is, sets are only available in the current session).

To create a set:

1. Click **New**.
2. In the *Create New Set* dialog, enter a unique name for the set and click **OK**.



The new set becomes available for selection in the **Set** drop-down menu.



3. To delete a set, select the set from the **Set** drop-down menu and click **Delete**.
   This action only deletes the set definition, which does not remove the entities from the surface. You cannot delete the **Default** set. The **Delete** button is unavailable for the default set.

Contents:

Storing Sets
Recalling Sets
Browsing Sets

See Also:

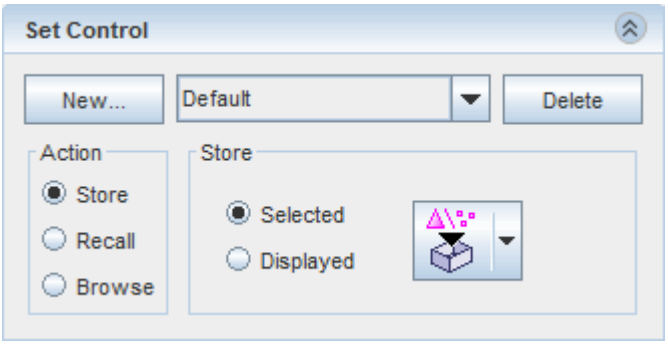
Changing the Parts in the Active Model
Storing Sets

Storing Sets

Storing allows you to save a group of faces, edges, and vertices to a previously created or default set. You can recall sets during the *Surface Repair* session.

To store a set:

1. Select **Store** from the *Action* group box.



2. Select the **Store** mode:
   - **Selected** — stores the entities that are selected with the Selection Controls.
   - **Displayed** — stores the entities that are displayed with the Display Controls
3. Store the selected or displayed entities in the current set:

   Select one of the **Store** options. Click the displayed (**Store**) icon, or click the down arrow next to this icon to access the remaining options.

   - (**Store**) — This option stores the selected or displayed entities in the current set, replacing all existing entities in the set.
   - (**Add store**) — This option adds the selected or displayed entities to the currently stored entities in the set.
   - (**Remove from set**) — This option removes the selected or displayed entities from the currently stored entities in the set.
   - (**Subset store**) — This option stores only those selected or displayed entities that already exist in the current set. Entities in the current set, that are not selected or displayed, are discarded.

   Note: The tool icon differs slightly depending on whether you choose the **Selected** or **Displayed** store option.

Both the **Store Displayed** and **Store Selected** options can store faces and edges. Vertices are never displayed in the *Graphics* window, hence the **Store Displayed** option is unable to store vertices in a set. To store vertices, select the required vertices and use the **Store Selected** option instead.

See Also:

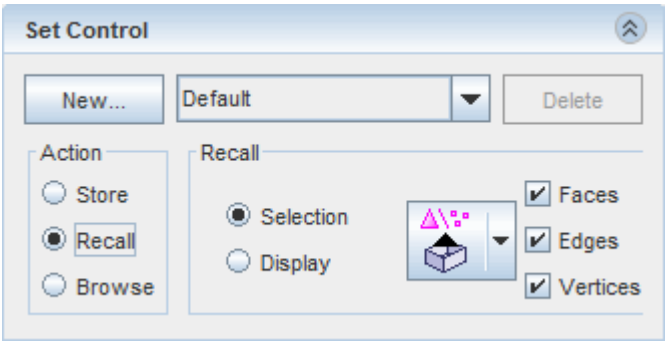
Selecting Entities
Recalling Sets
Set Controls
Recalling Sets

Recalling Sets

Recalling allows you to retrieve the collection of faces, edges, and vertices from a previously stored set.

To recall a set:

1. Select **Recall** from the *Action* group box and select the desired **Set** from the drop-down menu.



2. Select the **Recall** mode:
   - **Selection** — recalls the entities to the current selection
   - **Display** — recalls the entities to the current display

   In addition, the **Faces**, **Edges**, and **Vertices** filters are available. These filters limit the type of entities that are recalled by the tool. The default options activates all filters, which recall all entities. Deactivating a filter omits that particular entity from the recall operation.
3. Recall the selected set:

   Select one of the **Recall** options. Click the displayed (**Recall**) icon, or click the down arrow next to this icon to access the remaining options.

   - (**Recall**) — This option recalls the set, replacing all existing entities in the current selection or display.
   - (**Add Recall**) — This option adds entities in the set to the current selection or display.
   - (**Remove Recall**) — This option removes entities in the set from the current selection or display.
   - (**Subset Recall**) — This option recalls entities that are in both the set and current selection or display.

See Also:

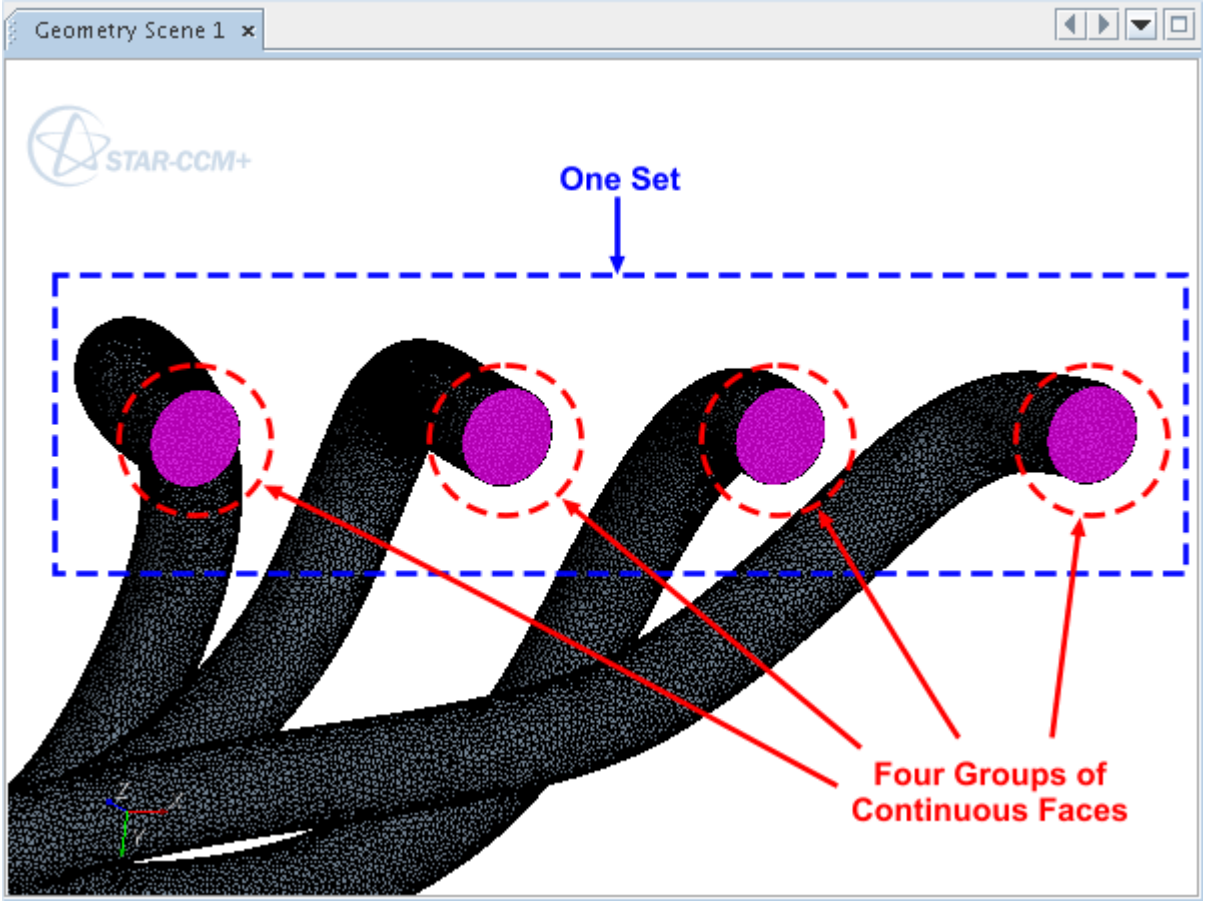
Storing Sets

Storing Sets

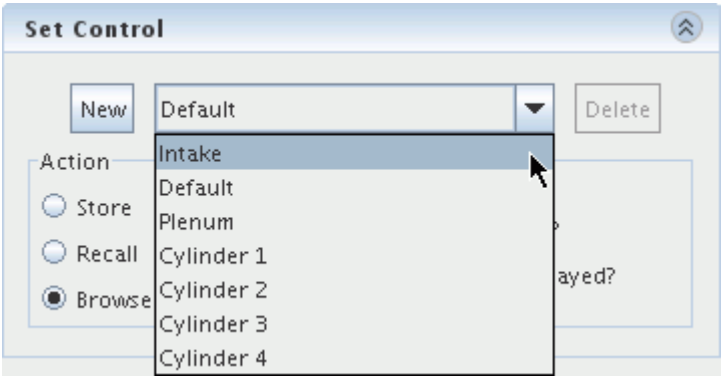
Browsing Sets

Browsing sets allows you to cycle and display groups of continuous entities within the currently selected set.

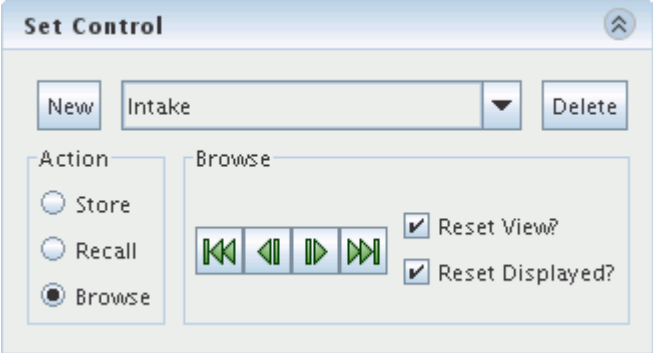
For example, in the diagram below, all the selected faces are in one set, but there are four groups of continuous faces.



1. To browse, first select a **Set** from the drop-down menu.



2. Then, select **Browse** from the *Action* group box.



The default options when browsing sets display the group of continuous entities, and two neighboring faces, in order to identify the surrounding area. Using the browse buttons also selects the group of continuous entities in the *Graphics* window.

While browsing, two options control the viewing of sets:

- **Reset View** resets the view to zoom in on the selected entities.

  If you deactivate the **Reset View** option, the view remains the same.
- **Reset Displayed** resets the displayed faces, edges, and vertices to the selected entities, and two neighbors.

  If you deactivate the **Reset Displayed** option, the displayed entities remain the same.

3. Use the browse buttons to cycle through and select continuous groups of entities within the set.

Query Diagnostic Responses From Java Macros

This section provides example Java and reference material for accessing information from surface preparation query diagnostics.

A query diagnostic response is data that comes from the server when you run a query diagnostic. Using java macros and tags, you can access individual parts of a diagnostic response to produce more specific information.

This section has the following subsections:

Contents:

Java Examples
Query Diagnostic Tag Names And Types Reference

Java Examples

The following examples show several uses of responses to obtain specific information from query diagnostics.

As prerequisites, this section contains the required import paths and basic java code.

These examples are summarized below:

- Integer — Accessing a vertex ID
- Integer Vector —Accessing the number of faces that belong to a boundary
- Double — Accessing the total length of selected edges
- Double Vector — Accessing the X, Y, and Z coordinates of a vertex
- Client-Server Objects — Access the query units, query coordinate system, and the part surface or boundaries to which a face belongs.

Import Paths

To access and use query responses, add the following import parts to your macro:

```
import star.base.neo.*;
import star.common.*;
import star.meshing.*;
```

Prerequisites

To use these examples, add the following java to your macro:

```
Simulation simulation_0 = getActiveSimulation();
PartSurfaceMeshWidget partSurfaceMeshWidget_0 =
  simulation_0.get(PartSurfaceMeshWidget.class);
SurfaceMeshWidgetQueryController
surfaceMeshWidgetQueryController_0 =
  partSurfaceMeshWidget_0.getControllers()
  .getController(
  SurfaceMeshWidgetQueryController.class
  );
```

Int Example

The following example illustrates the use of integers to access the ID of a selected vertex:

```
NeoProperty response =
  surfaceMeshWidgetQueryController_0.queryVertexXYZ();
int vertId = response.getInt("VertexId");
simulation_0.println(
  "Vertex ID: " + vertId
  );
```

IntVector Example

The following example illustrates the use of integer vectors to access the number of selected faces that belong to a boundary:

```
NeoProperty response =
  surfaceMeshWidgetQueryController_0
  .queryFaceRegionBoundaryList();
IntVector faceCount =
  response.getIntVector("SelectedBoundaryFaceCount");
simulation_0.println(
  "Face count: " + faceCount.getComponent(0)
  );
```

Double Example

The following example illustrates the use of doubles to access the total length of all selected edges:

```
NeoProperty response =
  surfaceMeshWidgetQueryController_0.queryEdgeLength();
simulation_0.println(
  "Total length: " + response.getDouble("TotalLength")
  );
```

DoubleVector Example

The following example illustrates the use of double vectors to access the X, Y, and Z coordinates of the selected vertex:

```
NeoProperty response =
  surfaceMeshWidgetQueryController_0.queryVertexXYZ();
DoubleVector labXYZ = response.getDoubleVector("LabXYZ");
simulation_0.println(
  "Vertex Coordinates: " + labXYZ
  );
simulation_0.println(
  "Vertex X-Coordinate: " + labXYZ.toDoubleArray()[0]
  );
```

ClientServerObject Examples

The following examples illustrate the use of client-server objects.

Units

This example accesses the preferred units for the query:

```
NeoProperty response =
  surfaceMeshWidgetQueryController_0.queryVertexXYZ();
Units units =
  response.<Units>getObject(
  "LengthUnits",simulation_0.getObjectRegistry()
  );
simulation_0.println(
  "Units: " + units.getPresentationName()
  );
```

Coordinate System

This example accesses the local coordinate system for the query:

```
NeoProperty response =
  surfaceMeshWidgetQueryController_0.queryVertexXYZ();
CoordinateSystem csys =
  response.<CoordinateSystem>getObject(
  "LocalCoordinateSystem",simulation_0.getObjectRegistry()
  );
simulation_0.println(
  "Local CSys: " + csys.getPresentationName()
  );
```

Part Surfaces or Boundaries

This example accesses the first part surface to which the selected faces belong:

```
NeoProperty response =
  surfaceMeshWidgetQueryController_0.
  queryFacePartSurfaceList();
Vector<PartSurface> partSurfaces =
  response.<PartSurface>getObjectVector(
  "SelectedPartSurfaces",simulation_0.getObjectRegistry()
  );
simulation_0.println(
  "First part surface: " + partSurfaces.get(0)
  );
```

This example accesses the first boundary to which the selected faces belong:

```
NeoProperty response =
```

```
  surfaceMeshWidgetQueryController_0
  .queryFaceRegionBoundaryList();
Vector<Boundary> boundaries =
  response.<Boundary>getObjectVector(
  "SelectedBoundarySurfaces",
  simulation_0.getObjectRegistry()
  );
simulation_0.println(
  "First boundary: " + boundaries.get(0)
  );
```

Query Diagnostic Tag Names And Types Reference

This section contains reference material on the available tag names and corresponding type for each query diagnostic.

The queries are summarized below:

- **Vertex Queries**
  - Vertex Location
  - Vertex Distance
  - Vertex Range
- **Edge Queries**
  - Edge Length
  - Edge Dihedral Angle
  - Edge Geometric Range
- **Face Queries**
  - Face Area
  - Face Quality
  - Face Centroid
  - Face Geometric Range
  - Part Surface List
  - Region Boundary List
  - Selection Volume
- **Mixed Entity Queries**
  - Vertex to Edge Distance
  - Vertex to Face Distance
  - Edge to Face Distance
  - Mixed Entity Geometric Range
  - Face to Part Tolerance

Contents:

Vertex Queries

Edge Queries
Face Queries
Mixed Entity Queries

Vertex Queries

This section contains reference material on the available tag names and corresponding type for all vertex-based query diagnostics.

Note: The same limitations and conditions apply to using vertex-based queries in macros as with using the GUI. See Vertex Query Tools for more information.

Vertex Location

The following response properties are available for single vertex selections.

Tag Name	Type	Description
`VertexId`	`Int`	Vertex ID of the selected vertex
`LabXYZ`	`Vector<Double>`	Coordinates of the selected vertex relative to the laboratory coordinate system
`LocalXYZ`	`Vector<Double>`	Coordinates of the selected vertex relative to the local coordinate system
`CoordinateSystem`	`ClientServerObject<CoordinateSystem>`	Coordinate system for the query
`LengthUnits`	`ClientServerObject<Units>`	Preferred length units for the query
`AngleUnits`	`ClientServerObject<Units>`	Preferred angle units for the query

The following response properties are available for single and multiple vertex selections.

Tag Name	Type	Description
`AllVertexIds`	`Int`	Vertex ID's of the selected vertices
`AllLabXYZCoords`	`Vector<Double>`	List of the coordinates of the selected vertices relative to the laboratory coordinate system
`AllLocalXYZCoords`	`Vector<Double>`	List of the coordinates of the selected vertices relative to the local coordinate system (only available if you set a local coordinate system)

Vertex Distance

Tag Name	Type	Description
StartVertexId	Int	Vertex ID of the starting vertex
EndVertexId	Int	Vertex ID of the ending vertex
VertexDistance	Double	Distance between vertices
StartVertexLabCoordinates	Vector<Double>	Coordinates of the starting vertex relative to the laboratory coordinate system
EndVertexLabCoordinates	Vector<Double>	Coordinates of the ending vertex relative to the laboratory coordinate system
XYZComponents	Vector<Double>	X, Y, and Z components of distance from minimum to maximum
LengthUnits	ClientServerObject<Units>	Preferred length units for the query
StartVertexLocalCoordinates	Vector<Double>	Starting vertex local coordinates
EndVertexLocalCoordinates	Vector<Double>	Ending vertex local coordinates
XYZLocalComponents	Vector<Double>	XYZ local coordinate system components from minimum to maximum in preferred units
AngleUnits	ClientServerObject<Units>	Preferred angle units specified for query
LocalCoordinateSystem	ClientServerObject<CoordinateSystem>	Coordinate system specified for query

Vertex Range

Tag Name	Type	Description
LabMinRange	Vector<Double>	Minimum range relative to the laboratory coordinate system
LabMaxRange	Vector<Double>	Maximum range relative to the laboratory coordinate system
XYZComponents	Vector<Double>	X, Y, and Z components of range from minimum to maximum
LengthUnits	ClientServerObject<Units>	Preferred length units for the query

Edge Queries

This section contains reference material on the available tag names and corresponding type for all edge-based query diagnostics.

Note: The same limitations and conditions apply to using edge-based queries in macros as with using the GUI. See Edge Query Tools for more information.

Edge Length

Tag Name	Type	Description
`EdgeId`	`Int`	Edge ID of the selected edge
`TotalLength`	`Double`	Total length of all selected edges
`MinLength`	`Double`	Minimum length of all selected edges
`MaxLength`	`Double`	Maximum length of all selected edges
`AveLength`	`Double`	Average length of all selected edges
`LengthUnits`	`ClientServerObject<Units>`	Preferred length units for the query

Edge Dihedral Angle

Tag Name	Type	Description
`EdgeId`	`Int`	Edge ID of the selected edge
`DihedralAngle`	`Double`	Dihedral angle of the selected edge
`ValidEdges`	`Int`	Number of edges that are valid for the query
`MinAngle`	`Double`	Minimum dihedral angle of all selected edges
`MaxAngle`	`Double`	Maximum dihedral angle of all selected edges
`AveAngle`	`Double`	Average dihedral angle of all selected edges
`AngleUnits`	`ClientServerObject<Units>`	Preferred angle units for the query
`MultiHedralAngles`	`Vector<Double>`	Dihedral angle for selected edge in preferred units

Tag Name	Type	Description
MultiHedralFaces	Vector<Double>	Face pair IDs for each dihedral angle

Edge Geometric Range

Tag Name	Type	Description
LabMinRange	Vector<Double>	Minimum range relative to the laboratory coordinate system
LabMaxRange	Vector<Double>	Maximum range relative to the laboratory coordinate system
XYZComponents	Vector<Double>	X, Y, and Z components of range from minimum to maximum
LengthUnits	ClientServerObject<Units>	Preferred length units for the query

Face Queries

This section contains reference material on the available tag names and corresponding type for all face-based query diagnostics.

Note: The same limitations and conditions apply to using face-based queries in macros as with using the GUI. See Face Query Tools for more information.

Face Area

Tag Name	Type	Description
FaceId	Int	Face ID of the selected face
TotalFaceArea	Double	Total area of all selected faces
MinArea	Double	Minimum area of all selected faces
MaxArea	Double	Maximum area of all selected faces
AveArea	Double	Average area of all selected faces
AreaUnits	ClientServerObject<Units>	Preferred area units for the query

Face Centroid

Tag Name	Type	Description
FaceId	Int	Face ID of the selected face

Tag Name	Type	Description
FaceCentroid	Vector<Double>	Face centroid coordinates
LengthUnits	ClientServerObject<Units>	Preferred length units for the query

Face Normal

Tag Name	Type	Description
FaceId	Int	Face ID of the selected face
FaceNormal	Vector<Double>	Face normal vector in preferred units
NormalizedFaceNormal	Vector<Double>	Normalized face normal vector
NormalUnits	ClientServerObject<Units>	Preferred length units specified for query

Face Quality

Tag Name	Type	Description
FaceId	Int	Face ID of the selected face
FaceQuality	Double	Face quality of the selected face
MinQuality	Double	Minimum face quality of all selected faces
MaxQuality	Double	Maximum face quality of all selected faces
AveQuality	Double	Average face quality of all selected faces

Face Geometric Range

Tag Name	Type	Description
LabMinRange	Vector<Double>	Minimum range relative to the laboratory coordinate system
LabMaxRange	Vector<Double>	Maximum range relative to the laboratory coordinate system
XYZComponents	Vector<Double>	X, Y, and Z components of range from minimum to maximum
LengthUnits	ClientServerObject<Units>	Preferred length units for the query

Part Surface List

Tag Name	Type	Description
SelectedPartSurfaces	ClientServerObject<PartSurface>	Part surfaces to which the selected faces belong
SelectedPartSurfacesFaceCount	Vector<Int>	Number of selected faces that belong to a part surface
SelectedContactSurfaces	ClientServerObject<PartSurface>	Contact part surfaces to which the selected faces belong
SelectedContactFaceCount	Vector<Int>	Number of selected faces that belong to a contact part surface

Region Boundary List

Tag Name	Type	Description
SelectedBoundarySurfaces	ClientServerObject<Boundary>	Boundaries to which the selected faces belong
SelectedBoundaryFaceCount	Vector<Int>	Number of selected faces that belong to the boundaries
SelectedInterfaceSurfaces	ClientServerObject<Boundary>	Interfaces to which the selected faces belong
SelectedInterfaceFaceCount	Vector<Int>	Number of selected faces that belong to an interface

Selection Volume

Tag Name	Type	Description
FaceVolume	Double	Volume of the face selection set
VolumeUnits	ClientServerObject<Units>	The preferred units used for the volume result
ClosedAndManifold	Boolean	True if the selection set is closed and manifold, else false

Mixed Entity Queries

This section contains reference material on the available tag names and corresponding type for all mixed-entity query diagnostics.

Note: The same limitations and conditions apply to using mixed-entity queries in macros as with using the GUI. See Mixed Query Tools for more information.

Vertex to Edge Distance

Tag Name	Type	Description
SourceVertexId	Int	Vertex ID of the selected vertex
ClosestEdgeId	Int	Edge ID of the closest edge
ShortestDistance	Double	Shortest distance from the selected vertex to the closest edge
SourceVertexLabCoordinates	Vector<Double>	Coordinates of the selected vertex relative to the laboratory coordinate system
XYZComponents	Vector<Double>	X, Y, and Z components of distance from minimum to maximum
LengthUnits	ClientServerObject<Units>	Preferred length units for the query

Vertex to Face Distance

Tag Name	Type	Description
SourceVertexId	Int	Vertex ID of the selected vertex
ClosestFaceId	Int	Face ID of the closest face
ShortestDistance	Double	Shortest distance from the selected vertex to the closest face
SourceVertexLabCoordinates	Vector<Double>	Coordinates of the selected vertex relative to the laboratory coordinate system
XYZComponents	Vector<Double>	X, Y, and Z components of distance from minimum to maximum
LengthUnits	ClientServerObject<Units>	Preferred length units for the query

Edge to Face Distance

Tag Name	Type	Description
SourceEdgeId	Int	Edge ID of the selected edge
ClosestFaceId	Int	Face ID of the closest face
ShortestDistance	Double	Shortest distance from the selected edge to the closest face

Tag Name	Type	Description
SourceVertexLabCoordinates	Vector<Double>	Coordinates of the selected edge relative to the laboratory coordinate system
XYZComponents	Vector<Double>	X, Y, and Z components of distance from minimum to maximum
LengthUnits	ClientServerObject<Units>	Preferred length units for the query

Mixed Entity Geometric Range

Tag Name	Type	Description
LabMinRange	Double<Vector>	Minimum range relative to the laboratory coordinate system
LabMaxRange	Double<Vector>	Maximum range relative to the laboratory coordinate system
XYZComponents	Vector<Double>	X, Y, and Z components of range from minimum to maximum
LengthUnits	ClientServerObject<Units>	Preferred length units for the query

Face to Part Tolerance

Tag Name	Type	Description
SelectedPart	ClientServerObject<MeshPart>	The source part used for the initial selection
CloseParts	Vector<MeshPart>	A vector of the close parts found within the specified search distance
ClosePartDistances	Vector<Double>	A vector of the closest distance for each part
SearchDistanceUnits	ClientServerObject<Units>	Units of the closest distance vector

Surface Repair Hotkeys

Hotkeys, also known as keyboard shortcuts, execute surface repair operations using the keyboard instead of the user interface.

Currently, the Surface Repair hotkeys are available in any surface preparation mode, even if the button is not visible in the *Surface Repair* in-place dialog. For example, if you are using the **Merge/Imprint** tool, you can still use `G` to flag edges as feature curves, even though the button is not visible. This shortcut saves time by removing the need to change between tools.

Surface Diagnostics

This slide-out panel appears on the right-hand-side of the *Graphics* window.

Action	Hotkey
Select First Problem Area	`UP`
Select Previous Problem Area	`LEFT`
Select Next Problem Area	`RIGHT`
Select Last Problem Area	`DOWN`
Toggle Full/Summary Size of Diagnostics Panel	`SPACE`
Open/Close Diagnostics Panel	`SHIFT + D`
Manage Diagnostics	`SHIFT + W`
Run/Execute Diagnostics	`CTRL + X`

Surface Repair

These tools appear under the *Repair* tab.

Action	Hotkey
Delete Selected Faces	`D`
Create Face from Selected Vertices	`V`
Collapse Selected Vertices	`L`
Split Selected Edges/Faces	`I`
Swap Selected Edge	`W`
Fill Polygonal Patch	`P`
Auto Repair Surface Errors	`A`
Fill Holes Using Selected Edges	`Q`
Zip Selected Edges	`E`
Remesh Selected Faces	`N`
Smooth Selected/Attached Vertices	`M`
Merge Adjacent Vertices/Duplicate Faces	`J`
Undo	`CTRL + Z`

Action	Hotkey
Redo	CTRL + Y

Feature Repair

These tools appear under the *Repair* tab.

Action	Hotkey
Flag Edges as Feature	G
Unflag Feature Edges	U
Mark Feature Edges	SHIFT + A
Toggle Chase Edge Selection Mode	SHIFT + C
Toggle Select Curve Mode	SHIFT + F
Fix All Feature Errors	SHIFT + T
Mark as Not a Feature Error	SHIFT + U
Flag Vertices as Feature	SHIFT + O
Unflag Feature Vertices	SHIFT + P
Mark Feature Vertices	SHIFT + S

Global Tools

These tools appear under the *Global* tab.

Action	Hotkey
Project Selected Vertices	CTRL + J
Transform Selected Entities	CTRL + W
Inflate or Translate Selected Faces or Edges	CTRL + O
Merge / Imprint Single or Multiple Parts	CTRL + I
Intersect or Boolean Selected Faces	CTRL + I
Find, Review, and Delete Selected Surfaces	CTRL + E

Projection Options

These options appear when the Global Tool, **Project Selected Vertices** is toggled on.

Action	Hotkey
Display the Plane Selection Tool in the Scene	ALT + 4
Perform the Projection Based on the Vertex Selection Set	ALT + J

Transform Options

These options appear when the Global Tool, **Transform Selected Entities** is toggled on.

Action	Hotkey
Perform the Transform Based on the Current Selection	`ALT + A`

Offset Faces / Edges Options

These options appear when the Global Tool, **Inflate or Translate Selected Faces or Edges** is toggled on.

Action	Hotkey
Offset Selected Faces / Edges in the Specified Direction	`ALT + C`

Merge / Imprint Options

These options appear when the Global Tool, **Merge / Imprint Single or Multiple Parts** is toggled on.

Action	Hotkey
Imprint the Selected Source Set onto the Destination Set	`ALT + I`

Intersect / Boolean Faces Options

These options appear when the Global Tool, **Intersect or Boolean Selected Faces** is toggled on.

Action	Hotkey
Intersect Selected Face Sets / Plane	`ALT + U`

Edit Connected Surfaces Options

These options appear when the Global Tool, **Find, review, and delete connected surfaces** is toggled on.

Action	Hotkey
Toggle Surfaces Highlighted in Table	`ALT + SPACE`
Toggle Mark on All Surfaces in Table	`ALT + X`
Delete Surfaces Marked in Table	`ALT + N`
Highlight Surfaces in Table That Have any Faces in Selection Set	`ALT + Y`

Query Tools

These tools appear under the *Query* tab.

Action	Hotkey
Vertex Locations in Laboratory and Local System	CTRL + SHIFT + V
Distance Between Two Selected Vertices	CTRL + SHIFT + D
Geometric Range of the Selected Vertices	CTRL + SHIFT + G
Length of the Selected Edge(s)	CTRL + SHIFT + L
Dihedral Angle of the Selected Edge(s)	CTRL + SHIFT + H
Geometric Range of the Selected Edge(s)	CTRL + SHIFT + E
Area of the Selected Face(s)	CTRL + SHIFT + R
Centroid of the Selected Face	CTRL + SHIFT + C
Normal of the Selected Face	CTRL + SHIFT + N
Quality of the Selected Face(s)	CTRL + SHIFT + Q
Geometric Range of the Selected Face(s)	CTRL + SHIFT + M
Part and Part Surface List for Selected Face(s)	CTRL + SHIFT + P
Toggle Find Leak Path Mode	SHIFT + L
Distance Between Selected Vertex and Edge(s)	CTRL + SHIFT + J
Distance Between Selected Vertex and Face(s)	CTRL + SHIFT + I
Distance Between Selected Edge and Face(s)	CTRL + SHIFT + K
Geometric Range of the Selected Faces / Edges / Vertices	CTRL + SHIFT + O

Find Leak Path

These options appear when the Query Tool, **Toggle Find Leak Path Mode** is toggled on.

Shortcuts when using **Find Leak Path** mode over-ride the shortcuts in other modes.

Action	Hotkey
Find Leak Path Between Selected Faces	ALT + D
Clear Leak Path	ALT + BACKSPACE

Set Control

These tools appear under the *Organize* tab.

Action	Hotkey
Store Selected	SHIFT + G
Add Store Selected	ALT + SHIFT + E
Remove Selected From Set	ALT + SHIFT + F

Action	Hotkey
Subset Store Selected	ALT + SHIFT + G

Part Edit

These tools appear under the *Organize* tab.

Action	Hotkey
Modify Part Surface of Selected Faces to Current	CTRL + Q

Selection Filter

These tools appear at the bottom of the *Graphics* window.

Action	Hotkey
Select None	SHIFT + N
Allow Faces to be Selected	ALT + 1
Allow Edges to be Selected	ALT + 2
Allow Vertices to be Selected	ALT + 3
Clear Selected Faces	SHIFT + 1
Clear Selected Edges	SHIFT + 2
Clear Selected Vertices	SHIFT + 3

Note: On a Windows computer, the grow and shrink displayed shortcuts do not always function when using the keys from the numeric row. In these instances, use the minus and plus keys from the numeric keypad instead.

Selection Controls

These tools appear on the left-hand-side of the *Graphics* window.

Action	Hotkey
Select Zone	SHIFT + Q
Add Select Zone	ALT + SHIFT + A
Deselect Zone	ALT + SHIFT + R
Subset Select Zone	ALT + SHIFT + S
Select Specified Threshold	SHIFT + R
Add Select Specified Threshold	ALT + SHIFT + B
Deselect Specified Threshold	ALT + SHIFT + K
Subset Select Specified Threshold	ALT + SHIFT + L

Action	Hotkey
Select Faces/Edges on Part Surfaces/Part Curves	SHIFT + B
Add Select Faces/Edges on Part Surfaces/Part Curves	ALT + SHIFT + C
Deselect Faces/Edges on Part Surfaces/Part Curves	ALT + SHIFT + M
Subset Select Faces/Edges on Part Surfaces/Part Curves	ALT + SHIFT + N
Grow selected	SHIFT + '='
Shrink selected	-
Multiple Grow selected	SHIFT + M
Select Attached Edges	SHIFT + E
Select Attached Vertices	SHIFT + V
Invert Selection	SHIFT + I

Display Controls

These tools appear at the top of the *Graphics* window.

Action	Hotkey
Show All Faces	CTRL + A
Hide All Faces	CTRL + SHIFT + A
Show Selected Faces	CTRL + F
Hide Selected Faces	CTRL + SHIFT + F
Show Faces on Part Surfaces	CTRL + B
Hide Faces on Part Surfaces	CTRL + SHIFT + B
Grow Displayed Faces	CTRL + '+'
Shrink Displayed Faces	CTRL + '-'
Toggle Face, Edge and Vertex Selection Depth Control	CTRL + SHIFT + S
Expand Color Displayed Faces by Menu	SHIFT + SPACE

Pop Up Menu of the Surface Repair Tool Graphics Window

The following pop-up menu items are available when you right-click a part or part surface in the *Graphics* window, while using the Surface Repair tool.

Hide	Hide the selected part or part surface from the display.

Show	Show the selected part or part surface in the display.
Show Only	Show only the selected part or part surface and hide all other parts or part surfaces.
Select	Select the current part, part surface, or part curve and deselect all existing selections.
Add Select	Add the current part, part surface, or part curve to the existing selection
Deselect	Remove the current part, part surface, or part curve from the selection (if it has been selected).
Subset Select	Select the current part, part surface, or part curve as a subset of the current selection. As such, the selected part, part surface, or part curve must already be included in the selection.
Set in Edit Panel	Select the current part surface or part curve in the Part Surface/Part Curve Edit group box.

See Also:

Assigning Part Entities to Target Part Entities

Selection Options

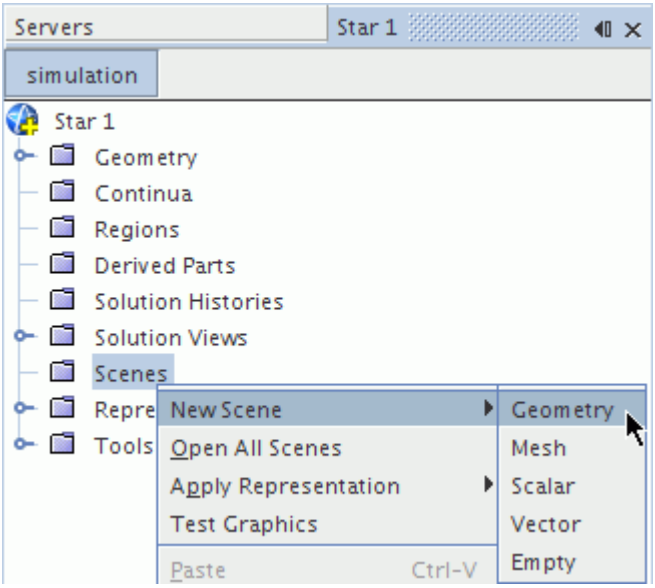
Displaying the Geometry

Displaying the geometry allows visual inspection of the geometry surface before generating a surface and volume mesh.

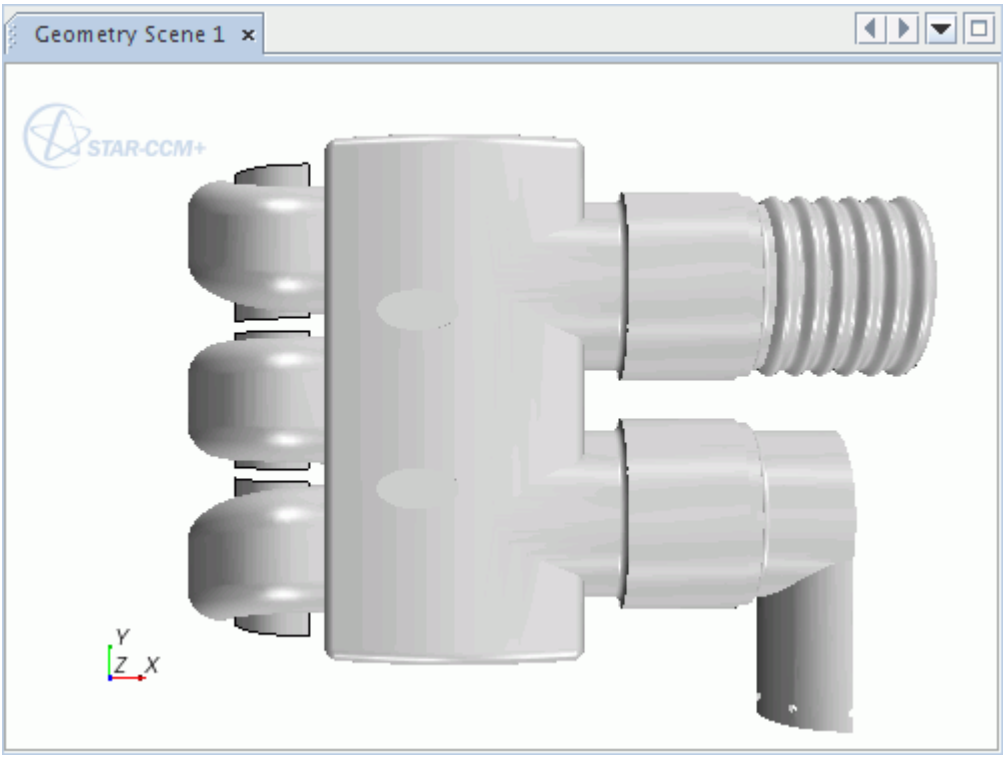
The easiest way to display the geometry is with a geometry scene. By default, this scene contains a geometry and outline displayer that have the *Geometry* and *Outline* properties that are activated respectively. You can change the properties of these displayers to control the display. See Geometry Displayer Properties.

To create a geometry scene:

1. Right-click the **Scenes** node and select **New Scene** > **Geometry**.



The geometry scene appears in the *Graphics* window. By default, the scene displays all parts with a constant color.



You can use the various view manipulations to navigate around the scene by panning, rotating, and zooming.

After creating a geometry scene, you can identify and hide regions and boundaries to simplify the display.

See Also:

Changing a Three-Dimensional Visualization Display

Managing Parts in Displayers

Defining the Regions Layout

An essential part of the process in setting up your simulation is to define the relationship between geometry parts and regions, boundaries, and interfaces.

Geometry parts are used only to define the faces, edges, and vertices that make up the surfaces of the model. The simulation domain on which mesh is generated, and for which physics is solved, is defined using regions, boundaries, and interfaces. These simulation domain objects can be created automatically using the hierarchy of geometry parts.

If you import geometry into STAR-CCM+, you can choose to interpret the geometry directly as regions, boundaries, and interfaces without going through geometry parts. This method was the default in earlier versions of STAR-CCM+. However, the recommended approach is now to use geometry parts initially and assign them to the simulation domain after making any required changes to surface or feature curve definitions.

Contents:

Regions Layout Workflow
Assigning Geometry Parts to Regions
Initial Assignment of Boundary Types
Interfaces Overview
Region Manipulation
Manipulating Boundaries
Performing Surface Boolean Operations

Regions Layout Workflow

In defining the regions layout you map between the geometrical definition of the problem and the computational (or physical) definition.

Make sure that you have all the geometry parts, and part contacts, that you require to implement the planned layout. If you have more than one region, and the regions are connected, then you must create interfaces in the region layout. STAR-CCM+ can create boundary-based interfaces or parts-based interfaces automatically.

To create the region layout:

1. If the region layout requires interfaces between adjoining regions, decide whether you are using boundary-based interfaces (one pair of boundaries per interface) or parts-based interfaces (multiple pairs of parts per interface).
2. If you want to have one region for each geometry part, with boundary-based interfaces between regions, do the following steps:
   a) Within the **Geometry** > **Parts** node, select all the parts that you intend to use in regions.
   b) Right-click on one part and choose **Assign Parts to Regions...**
   c) In the *Assign Parts to Regions* dialog, in the lower part of the dialog, choose the following options:
      - **Create a Region for Each Part**
      - **Create a Boundary for Each Part Surface**
      - **Create One Feature Curve for All Part Curves**
      - Select **Create Boundary-mode Interfaces From Contacts**
   d) Click **Apply** followed by **Close**.
3. If you want to assign multiple parts to specific regions, with parts-based interfaces between regions, do the following steps. You Either create empty regions and assign geometry parts to them afterwards, or else select multiple geometry parts and create the regions automatically. Use actions from the table below.

Action	Instructions
Create an empty region	Right-click the **Regions** node and choose **New**.
Assign a geometry part to an existing region	Within **Geometry** > **Parts**, right-click the part that you want to assign and choose **Set Region** > **Existing...**. Select the region and click **OK**.
Assign multiple geometry parts to an existing region	1. Within **Geometry** > **Parts**, right-click one of the parts that you want to assign and choose **Assign Parts to Regions...** 2. In the dialog, within the parts selection panel, select all the parts you want to assign. 3. In the lower options panel, choose **Assign All Parts to An Existing Region**, click **Select**, and choose the target region. 4. Select an appropriate option for boundaries.

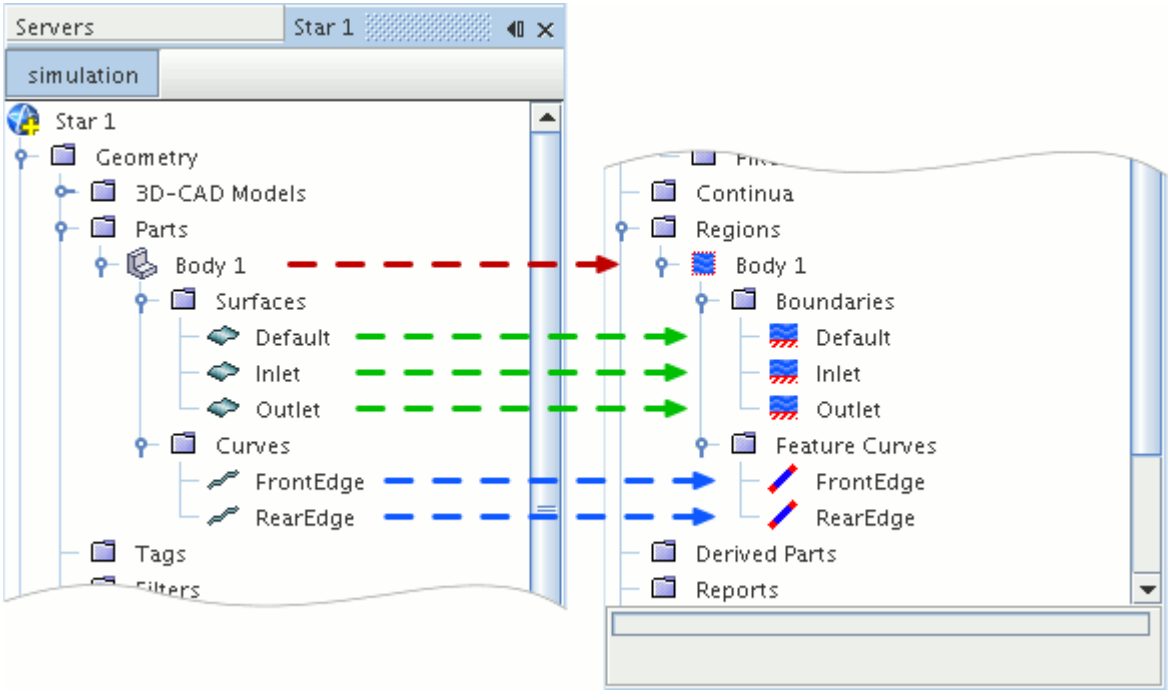
Action	Instructions
	5. Select **Create Contact-mode Interfaces From Contacts**. 6. Click **Apply** followed by **Close**.
Automatically create a single region from one or more geometry parts	1. Within **Geometry** > **Parts**, multi-select the parts that you want to assign to a new region. 2. Right-click one of the parts and choose **Set Region** > **New...** 3. Enter a name for the region and click **OK**.

4. If you want to create an interface that couples two regions by volume, for either an overset mesh calculation or a heat exchanger:
   a) Within **Regions**, multi-select the two regions that define the interface.
   b) Right-click and select **Create Interface** followed by **Heat Exchanger** or **Overset Mesh**.
5. If you want to create an interface between two boundaries, either in the same region or in different regions:
   a) Multi-select the two boundaries.
   b) Right-click and choose **Create Interface**.
6. Set the interface type. For each interface in turn:
   a) Select the interface node and set *Type* to one of the available options.
7. If, after creating the volume mesh, you wish to check the number of faces allocated to each interface:
   a) In the simulation tree, expand the **Representations** node.
   b) Navigate to the appropriate volume mesh, expand the region, and select the node that represents the side of the interface that belongs to the region.
   c) In the *Properties* window, review the number of *Faces*.

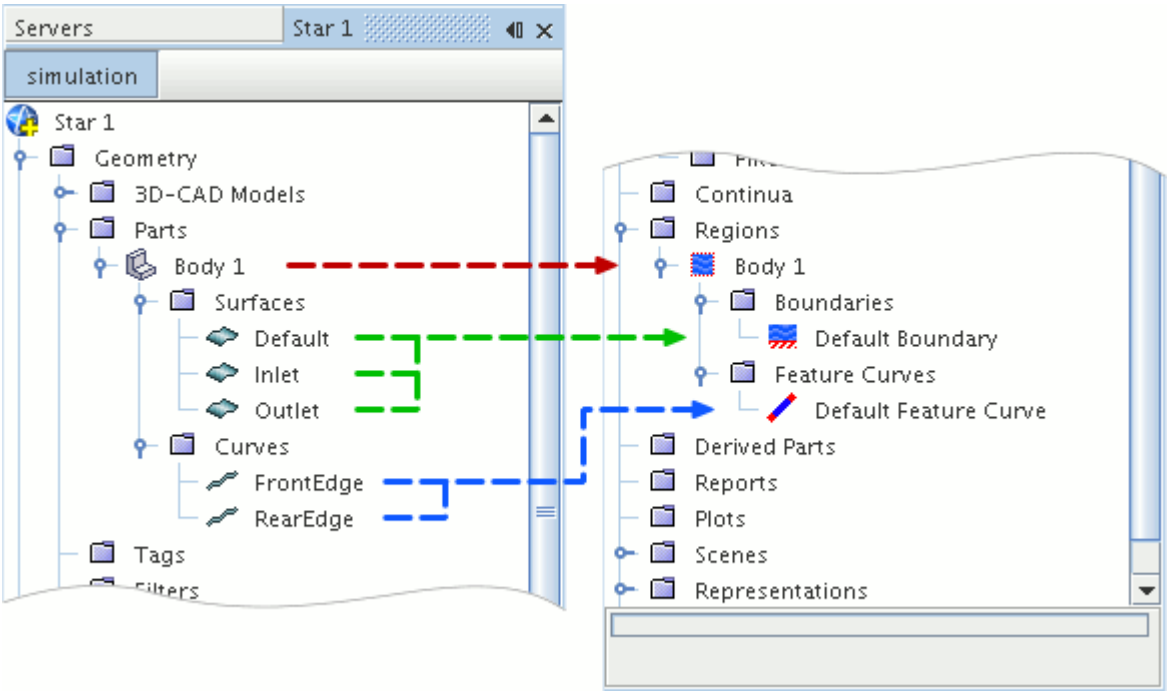
Assigning Geometry Parts to Regions

The mesh continuum models cannot be used for geometry parts. In order to mesh geometry parts, they must be assigned to one or more regions, and the mesh is then generated for the region.

The following diagram illustrates the relationship between objects in the part tree and the ones in the region tree:



A part must be assigned to a region before its part surfaces can be assigned to boundaries, and its part curves to feature curves, within the region. The relationships do not need to be one-to-one; multiple parts can be assigned to the same boundary, and multiple part curves to the same feature curve. The following mapping would also be valid, for example:



It is important to note that part contact information can only be used to generate interfaces automatically if all contacting parts are assigned to new regions in the same operation.

Contents:

Using the Assign Parts to Regions Dialog
Assign Parts to Regions Reference
Assigning Parts to Regions
Assigning Part Surfaces to Boundaries
Assigning Part Curves to Feature Curves

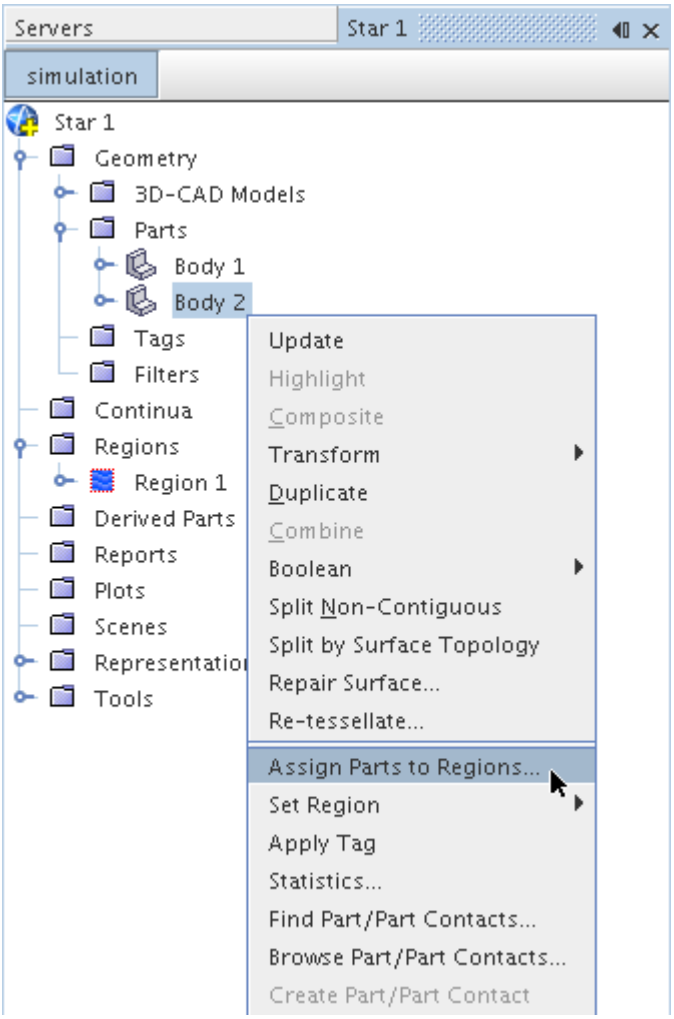
Using the Assign Parts to Regions Dialog

The Assign Parts to Regions dialog lets you assign parts to regions, boundaries, or feature curves.

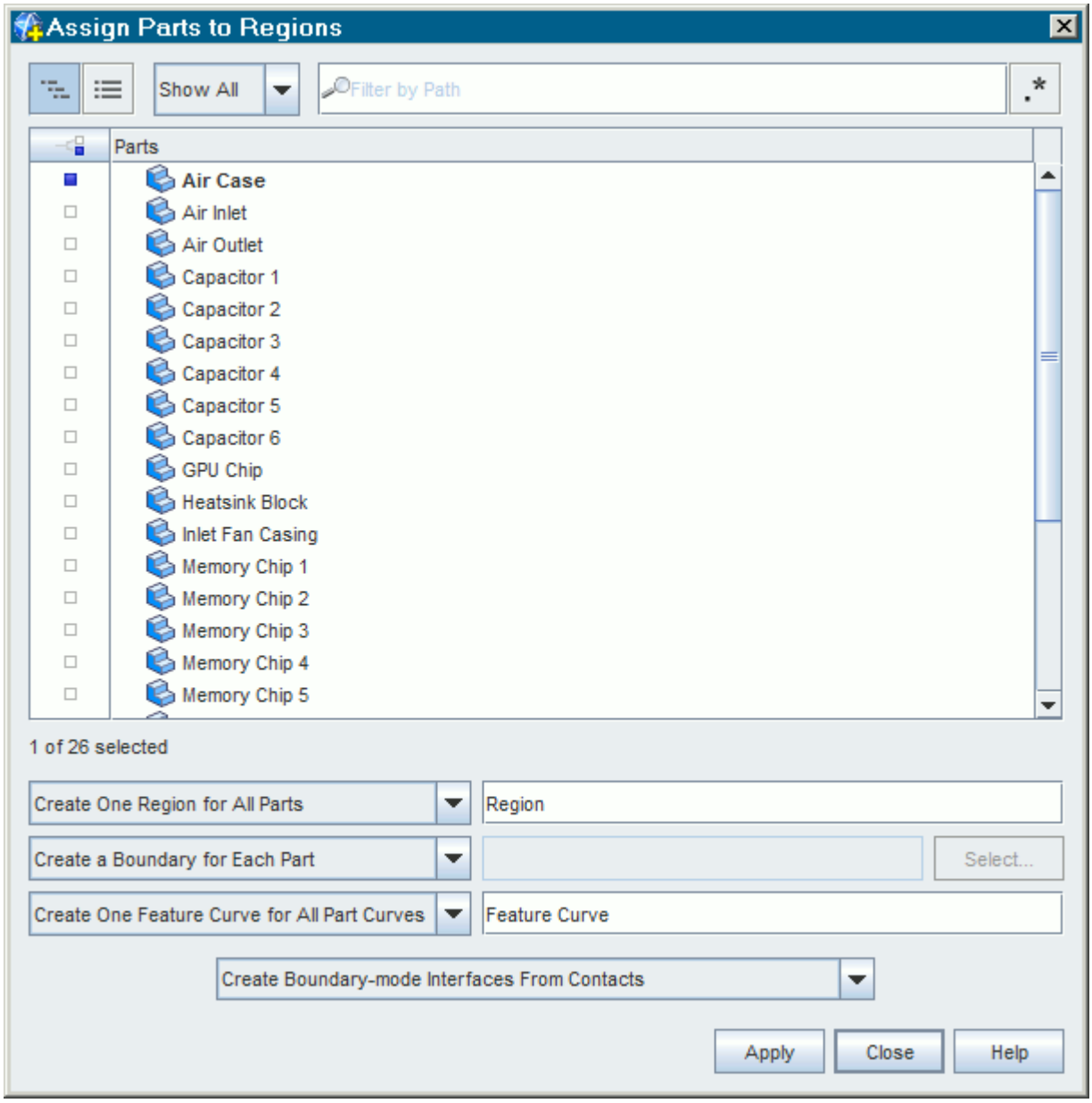
The *Assign Parts to Regions* dialog can be used to assign leaf-level parts to regions.

To bring up this dialog:

1. Right-click on the desired part node and select **Assign Parts to Regions...**.



The *Assign Parts to Regions* dialog appears.



The dialog is split into two main sections. The top section shows all the current parts in the simulation. The bottom section contains all the options for setting the regions, boundaries, and feature curves.

2. To select or deselect a part, click the selection button to the left of it in the selection column.
3. Select the options that are suitable for your workflow. See Assign Parts to Regions Reference.
4. Click **Apply** to create the region with the specified settings.

   You can assign different parts, with different settings, to regions using the same dialog.
5. Click **Close** to close the dialog.

The **Set Region** operation is independent of the **Assign Parts to Regions** operation.

Assign Parts to Regions Reference

Using the *Assign Parts to Regions* dialog, you can map geometry objects to their corresponding region objects in preparation for simulation. The dialog supports several workflows, some of which are documented elsewhere.

The dialog is split into two main sections. The top section shows all the current parts in the simulation. The bottom section contains all the options for setting the regions, boundaries, and feature curves.

Region options	**Create a Region for Each Part** Based on the chosen parts, this option creates a region for each part. **Assign All Parts to an Existing Region** Activates an object selection dialog in single-select mode. This option does not appear if no regions exist in the simulation. **Create One Region for All Parts** Default option. This option assigns the selected parts to one region. You can edit the name of the target region in the adjacent text box.
Boundary Options	**Create One Boundary for Each Part Surface** For each part surface defined in the chosen parts, this option creates a corresponding boundary in the target region. **Create One Boundary for Each Part** Creates one boundary for each part, regardless of how many part surfaces are defined on the part. **Assign All Part Surfaces to an Existing Boundary** When assigning parts to an existing region, you can similarly choose to assign part surfaces to boundaries within the region. **Create One Boundary for All Part Surfaces** Assigns all the part surfaces to one boundary. You can edit the name of the target boundary in the adjacent text box.
Feature Curve Options	**Create a Feature Curve for Each Part Curve** For the part curves defined in the chosen parts, this option creates a corresponding feature curve within the target region. **Assign All Part Curves to an Existing Feature Curve** When assigning parts to an existing region, this option assigns all part curves to an existing feature curve.

	Create One Feature Curve for All Part Curves Assigns all the part curves to one feature curve. You can edit the name of the target feature curve in the adjacent text box.
Interface Options	**Create Boundary-mode Interfaces From Contacts** Converts part contacts between the chosen parts into boundary-based interfaces between the target regions. Only one pair of boundaries can be allocated to an interface. **Create Contact-mode Interfaces From Contacts** Converts part contacts between the chosen parts into parts-based interfaces between the target regions. Any number of pairs of parts can be allocated to an interface. **Do Not Create Interfaces From Contacts** Does not generate any interfaces between the target regions.

Automatic Naming of Regions and Boundaries

In the automatic naming process, the name that is given to a region or boundary differs depending on what settings are used. In the following example, both cases have the same part and part surface name: **Body 1** and **Inlet** respectively. In both cases, the selected boundary option is **Create a Boundary for Each Part Surface**.

Region mode:	One region for each part	One region for all parts
New region name:	Body 1	Region
New boundary name:	Inlet	Body 1.Inlet

Assigning Parts to Regions

Before a leaf-level part can be used in the meshing process, it must be assigned to a region.

A bi-directional relationship exists between leaf-level parts and regions such that you can think of it in terms of “this part belongs to this region” or alternatively, “this region is composed of these parts”. Several methods exist to form these relationships:

- by using Set Region;
- by using the Assign Parts to Regions dialog; or
- by using the Parts Property in the Region Node

Any of these methods can be used to define the desired relationship between parts and regions. However, although both composite parts and leaf-level parts can be assigned to a region (using the first method that is described above), the relationship is ultimately between the leaf-level part and the region. Additionally, a leaf-level part can only belong to one region at a time. Therefore if a part is reassigned to a new region it is automatically removed from the original region part list.

Along with the part being assigned to the region, all surfaces and curves belonging to the part are also assigned to the region boundaries and features curves respectively.

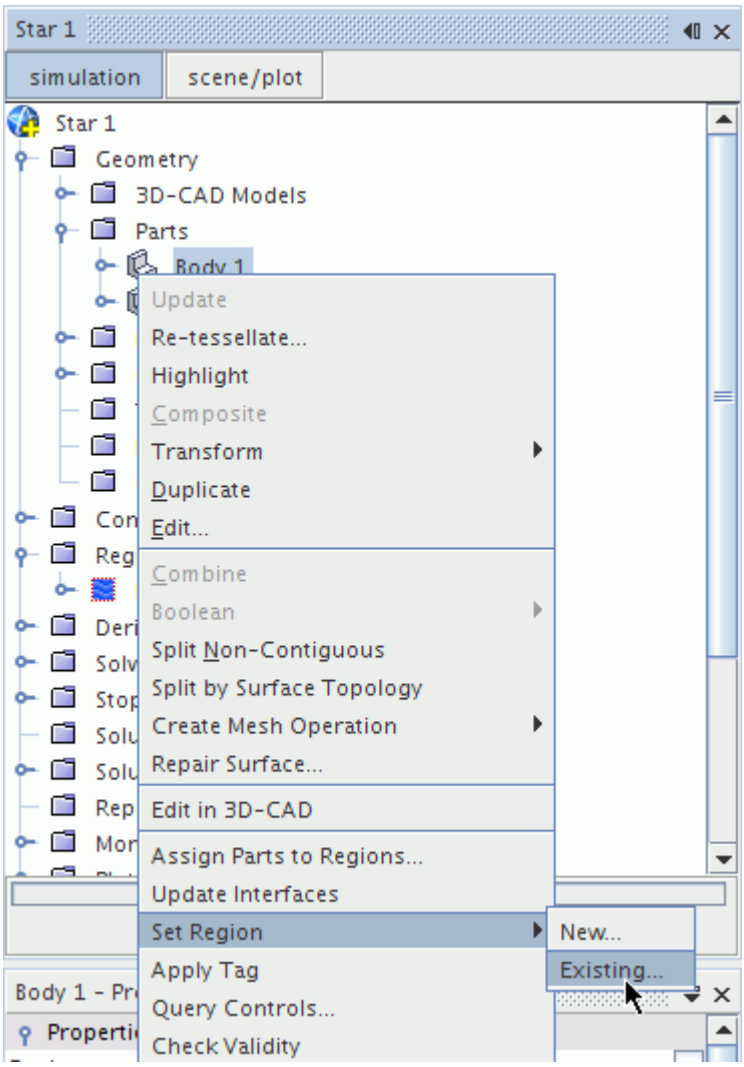
Contents:

Using Set Region
Using the Parts Property in the Region Node

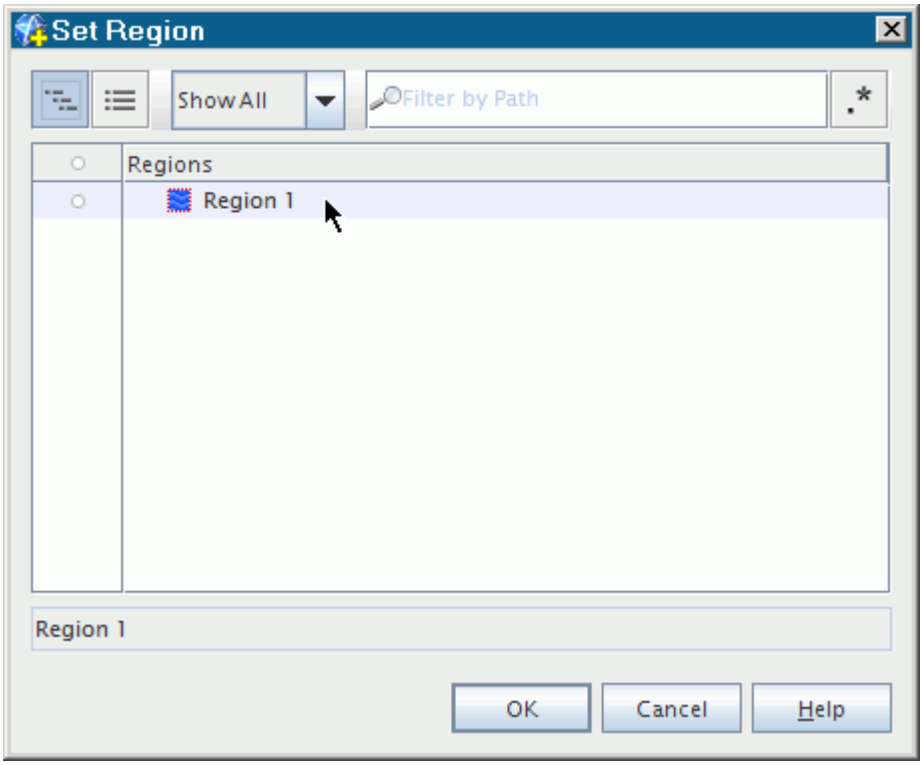
Using Set Region

The Set Region option lets you assign parts to regions.

To assign a leaf-level part directly to an existing region, right-click the part node and use the **Set Region** menu item. For a small set of regions, a list of the regions is available as a submenu. For a larger set, the **Existing** submenu item activates the object selector (see Activating the Object Selection Dialog):

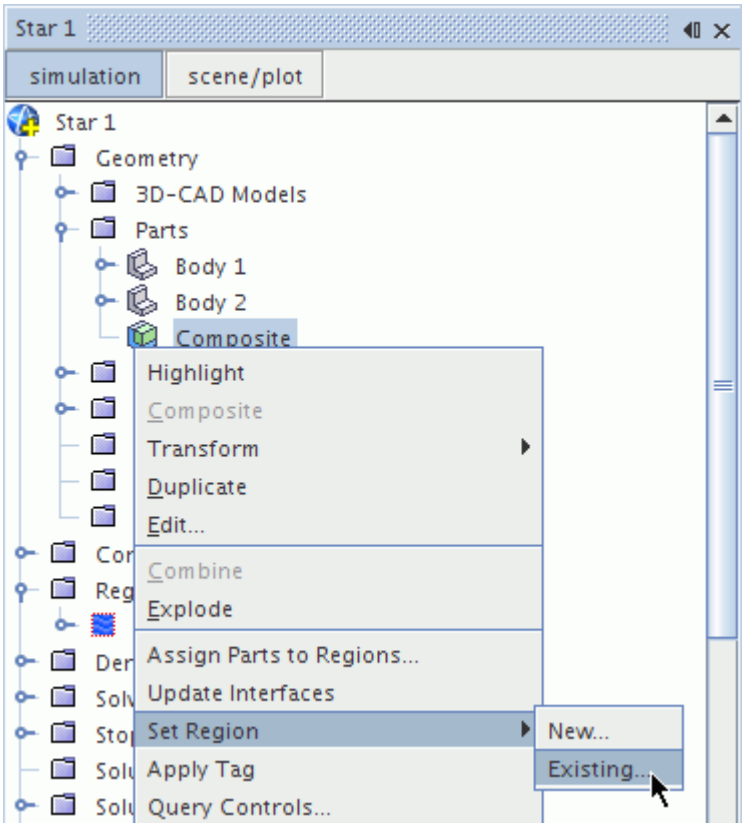


Select the name of the chosen region from the object selector.



In the above example, the **Body 1** leaf-level part is assigned to **Region 1**. More than one leaf-level part can be selected and assigned to the same region at the same time using this method.

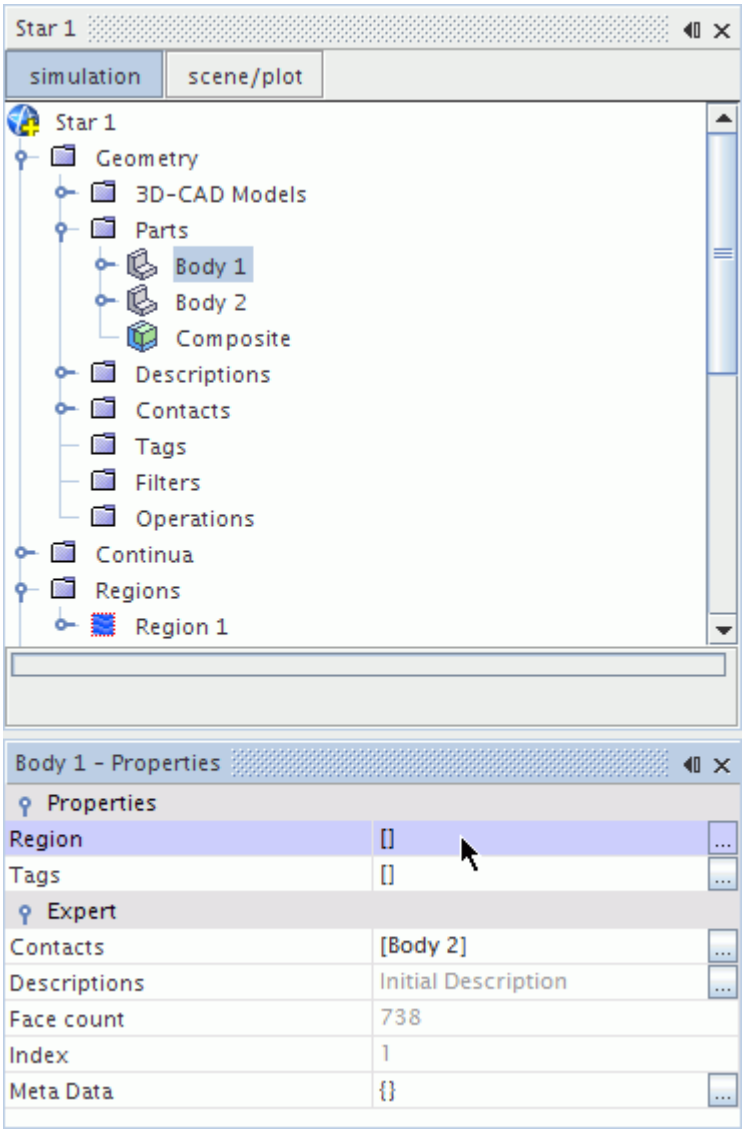
In a similar way to leaf-level parts, a composite part can be assigned to a region so that all of its child leaf-level parts are automatically linked to the region. This linking can be done using the same technique that is described previously for leaf-level parts.



This action results in all child leaf-level parts under the composite part being assigned to a region.

Setting a part to a region using this method automatically assigns all surfaces in that part to the `default` boundary in the region. If the default boundary does not exist or has been renamed, all part surfaces are assigned to the boundary with the lowest index number. The index number of each boundary is shown in the Boundary Properties window. If the name of a part surface matches the name of an existing boundary, the part surface is assigned to that boundary.

The region that a leaf-level part is assigned to can be reviewed or modified using the part node properties. Selecting the part node of interest shows the *Properties* for that node:

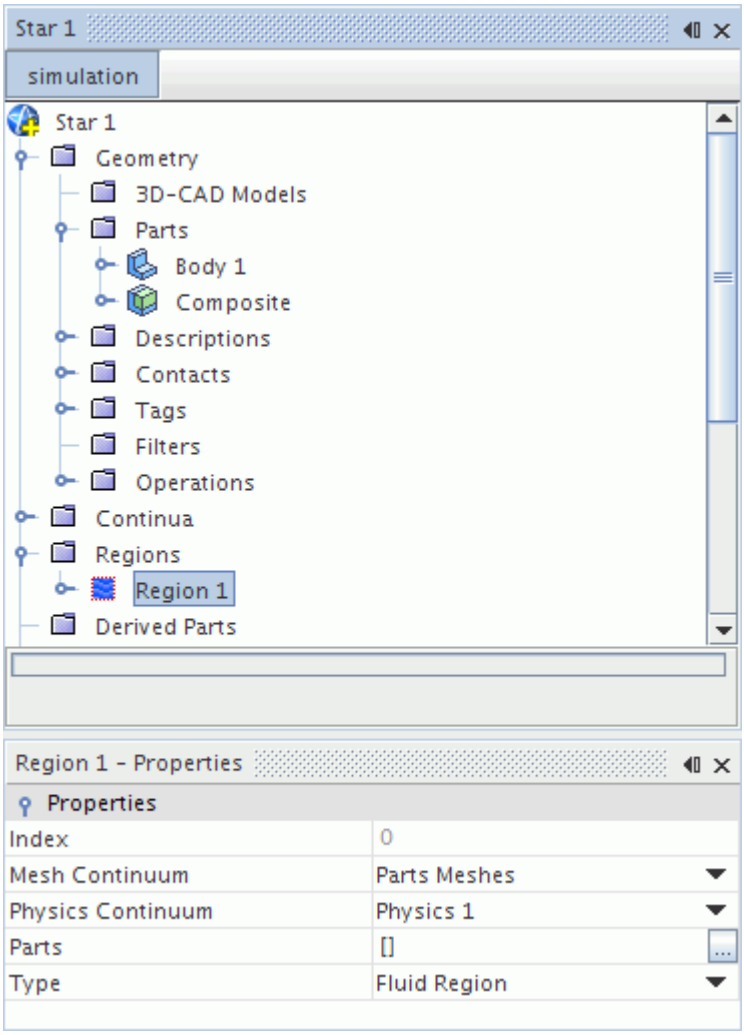


The *Region* property activates an object selector to make the region selection from the available list. New regions cannot be created using this method.

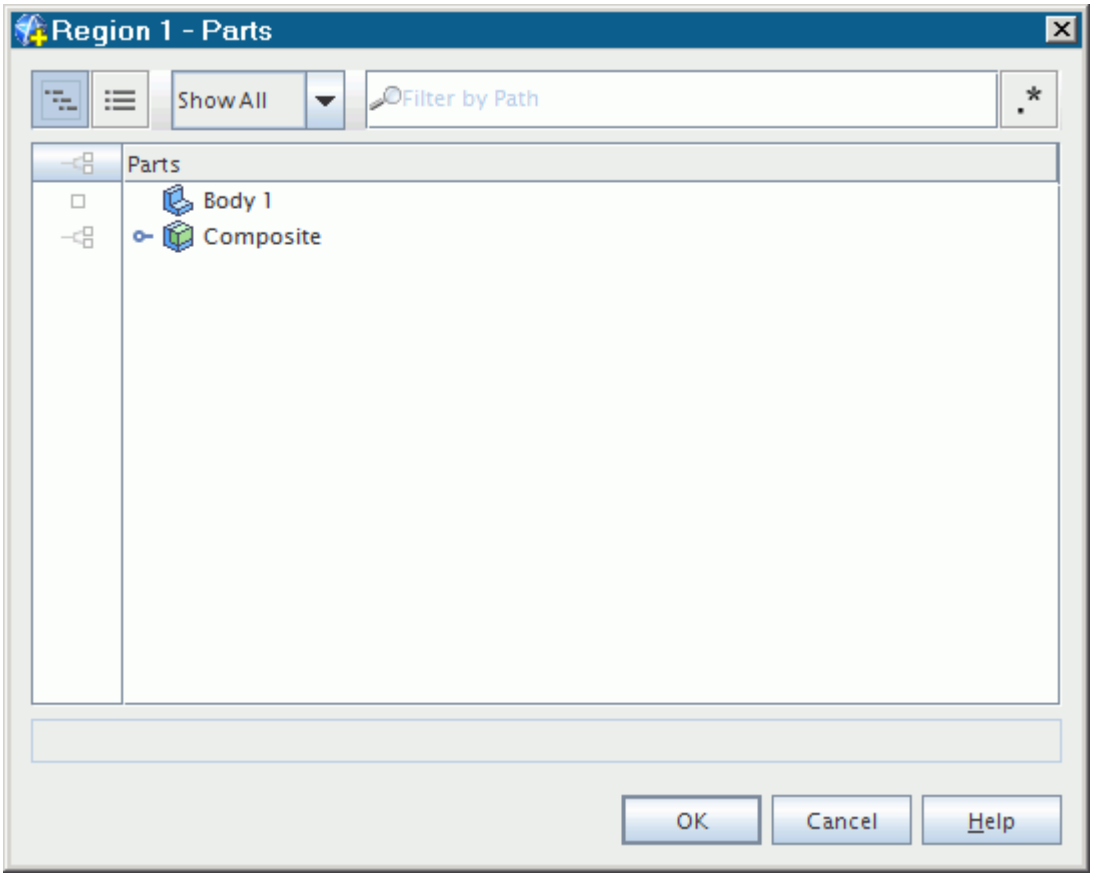
Using the Parts Property in the Region Node

The parts that are assigned to a region can be set using the region node properties.

Selecting the region of interest displays the *Parts* property. For example:



Click the right half of the property to activate the object selector. Use the features of the object selector to access and select the names of the chosen parts.



After you close the object selector, the *Parts* property of the region shows the leaf-level parts that are assigned to the region node.

When regions are created from the **Regions** manager node, initially no parts are associated with them. When parts are associated using one of the other methods that are described in this section, the *Parts* property is automatically populated.

Assigning Part Surfaces to Boundaries

As with parts and regions, a bi-directional relationship exists between part surfaces and boundaries.

Individual part surfaces cannot be assigned to a boundary until the parent part has been assigned to a region first. There are two different methods by which the surface/boundary relationship can be established:

- by assigning part surfaces to a boundary directly; or
- by selecting part surfaces for a given boundary

Either or both methods can be used. A surface however can only belong to one boundary at a time so if it is reassigned to a new boundary it is automatically removed from the original boundary list. To unassign a surface from a boundary, remove it from the *Part Surfaces* properties in the **Regions** > **Region node** > **Boundaries** > **Boundary** node that contains the surface. This step is outlined in the second method above.

After a surface face is assigned to a boundary, its color in the current scene changes from a light silver color to a darker gray color. This change indicates which surfaces have been assigned and which have not.

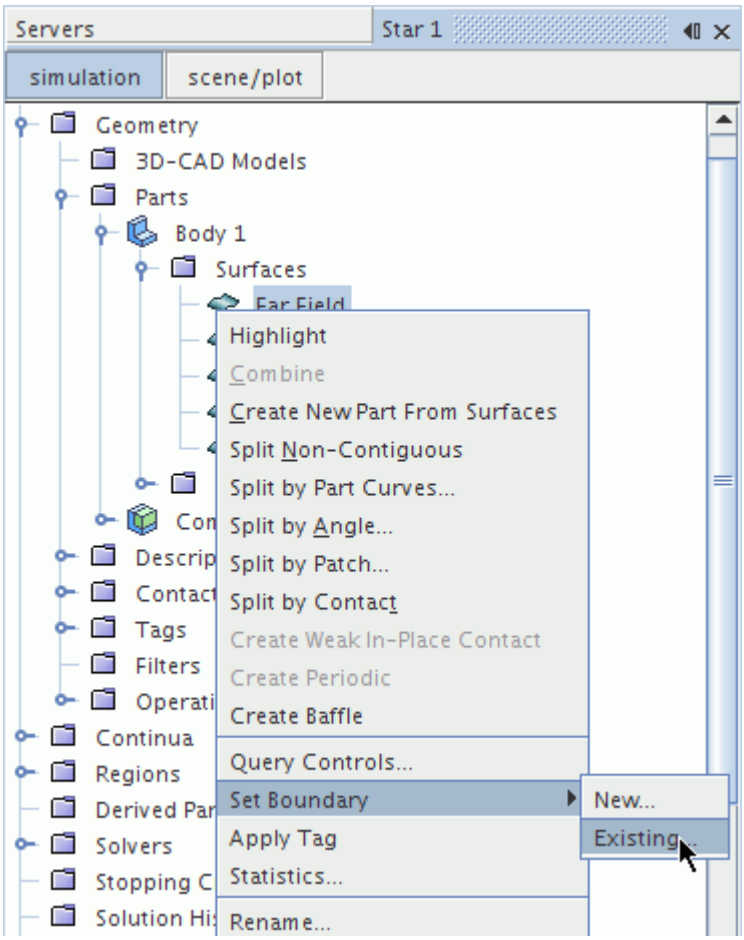
Contents:

Assigning Surfaces to Boundaries Using the Part Surface Nodes
Assigning Surfaces to Boundaries Using the Region Boundary Nodes

Assigning Surfaces to Boundaries Using the Part Surface Nodes

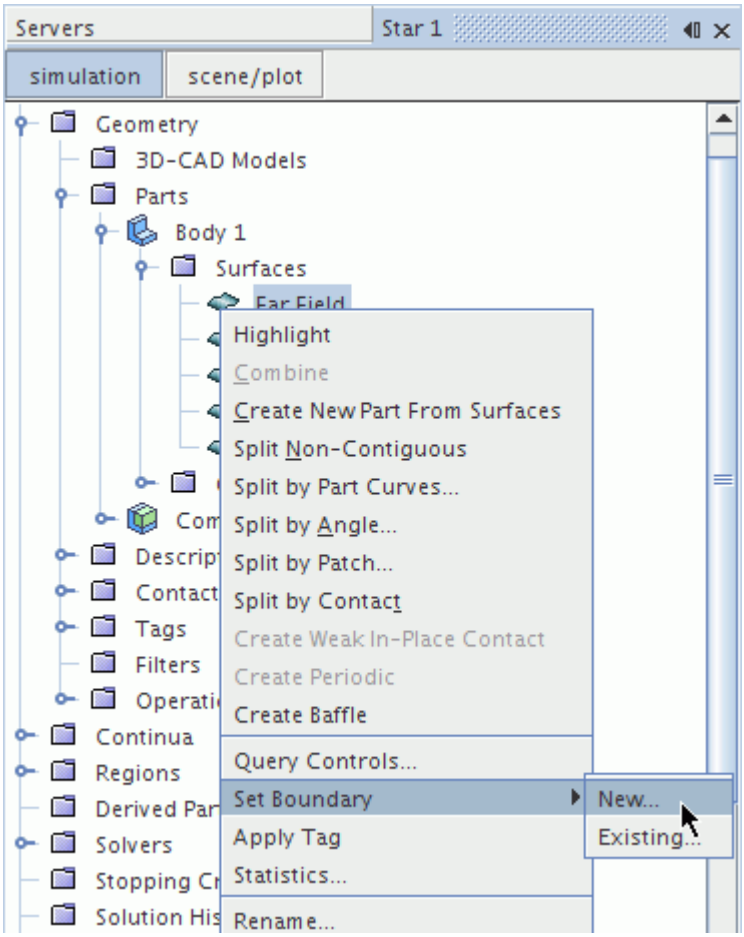
The Set Boundary option lets you assign surfaces to boundaries.

To assign a surface face directly to a boundary, right-click the face node and use the **Set Boundary** menu item. For a small set of boundaries, a list of the boundaries is available as a submenu. For a larger set, the **Existing** submenu item activates the object selector (see Activating the Object Selection Dialog):

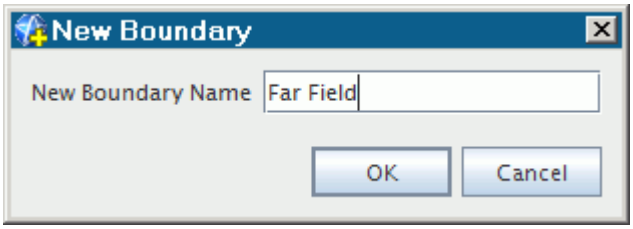


More than one face node can be selected and assigned to the same boundary using the above method.

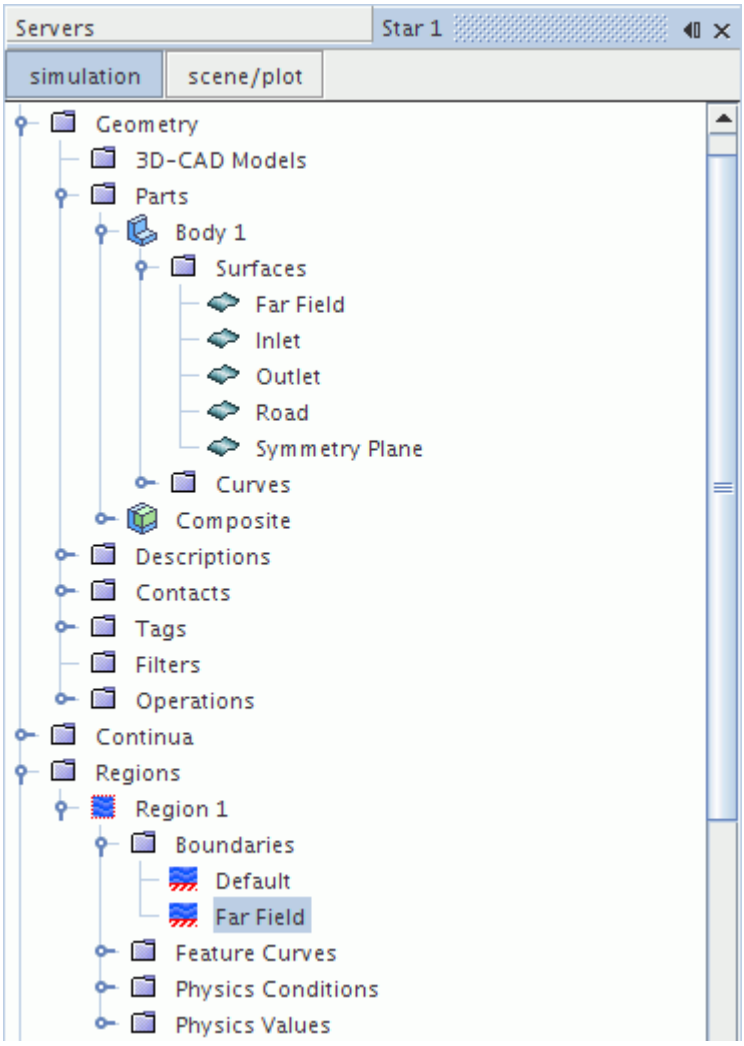
A new region always contains one boundary by default, usually with the name **Default**. If a new boundary is preferred for the selected face, then the **Set Boundary** > **New...** option can be selected instead, as shown below:



A dialog appears with a text box for you to name the new boundary. The boundary name matches the selected face node by default.



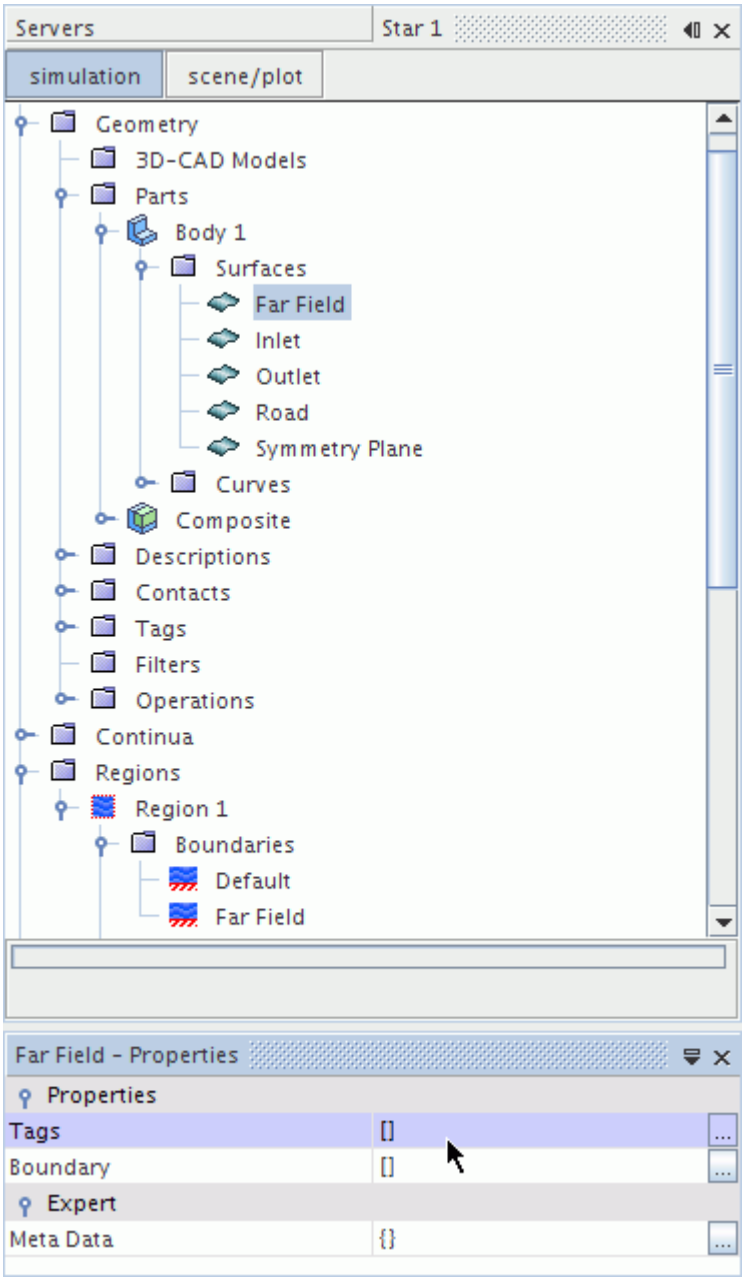
After you click **OK**, a new boundary is created and is automatically assigned the face. The new boundary is shown below:



If two or more face nodes are selected for the new boundary option, then the boundary name is based on the keyword **Boundary** instead of the face node name.

If the region to which the parent part is assigned has no boundaries existing for it, only the **New** suboption is available.

Alternatively, the boundary that a face node is assigned to can be reviewed or modified using the face node properties. Selecting the face node of interest shows the properties for that node:

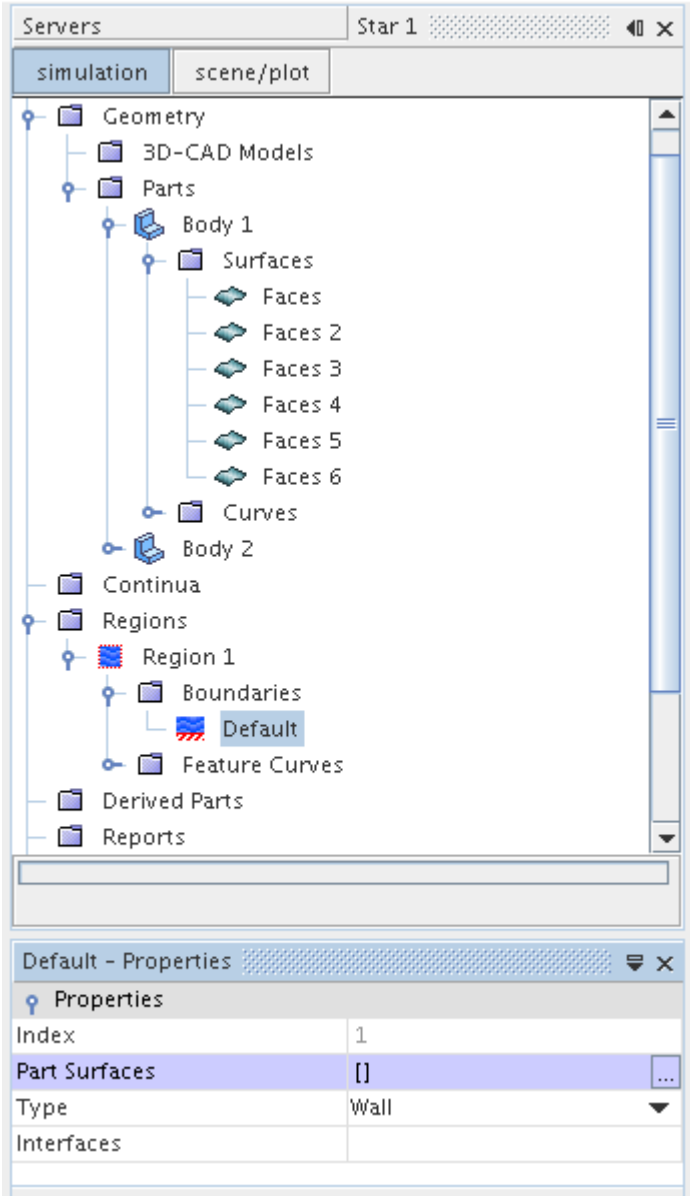


Click the right half of the property to activate the object selector, which only shows boundaries belonging to the region to which the parent leaf-level part is assigned. A new boundary cannot be created using this method.

Assigning Surfaces to Boundaries Using the Region Boundary Nodes

The part faces that are assigned to a boundary can be set using the boundary node *Properties* window.

Selecting the boundary node of interest displays the *Part Surfaces* property. For example:



Use the object selector to assign faces to the boundary.

When boundaries are created from the **Boundaries** manager node, initially no part surfaces are associated with them. When part surfaces are associated using another method, the *Part Surfaces* property is automatically updated for the boundary.

Assigning Part Curves to Feature Curves

Before you can assign individual part curves to a feature curve, you must assign the parent part to a region.

By default this action assigns all child curves to the first feature curve that belongs to the region. If necessary, modify the arrangement so that some edges belong to specific or new feature curves. As for parts and regions, a bi-directional relationship exists between part curves and feature curves. As such, there are two different methods by which the part curve/feature curve relationship can be established:

- By assigning part curves to a feature curve directly; and
- By selecting part curves for a given feature curve

Either or both methods can be used to define the desired arrangement between part curves and feature curves. However, a part curve can only belong to one feature curve at a time. Therefore if it is reassigned to a new feature curve, it is automatically removed from the original feature curve list. To unassign a part curve from a feature curve, remove it from the *Part Curves* properties in the **Regions** > **Region node** > **Feature Curve** > **Curve** node that contains the part curve. (See the second method above.)

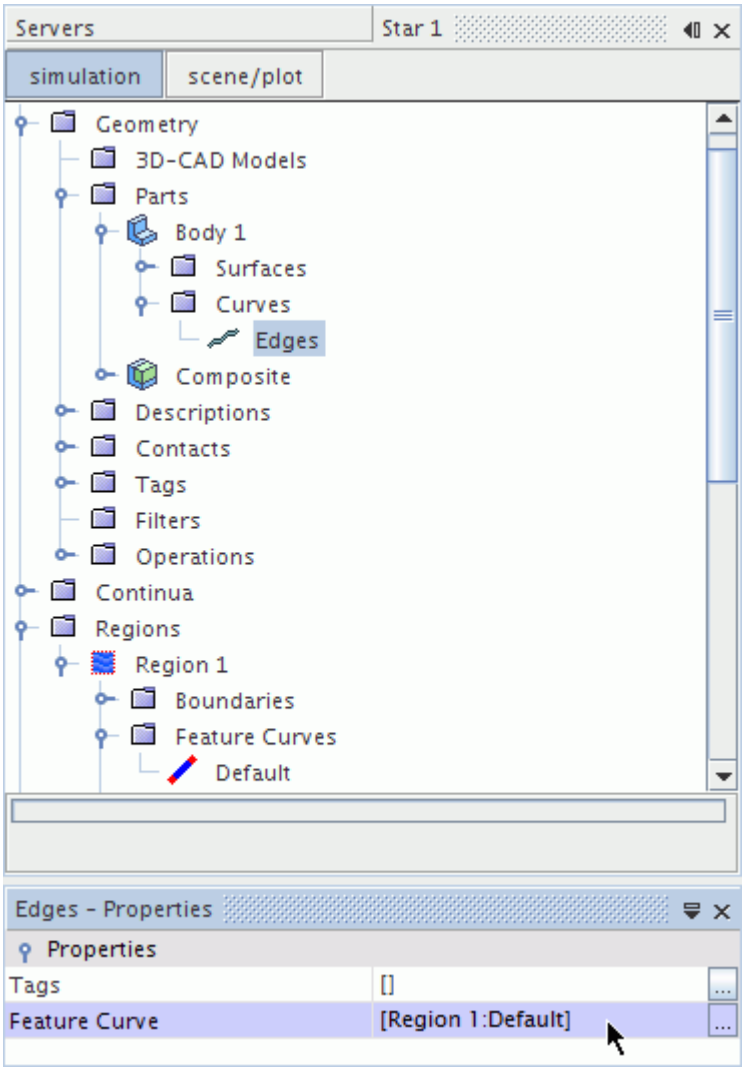
Contents:

Assigning Curves to Feature Curves Using the Part Curve Nodes
Assigning Curves to Feature Curves Using the Region Feature Curve Nodes

Assigning Curves to Feature Curves Using the Part Curve Nodes

The feature curve to which an edge node is assigned can be reviewed or modified using the edge node properties.

Selecting the edge node of interest shows the *Properties* window for that node:

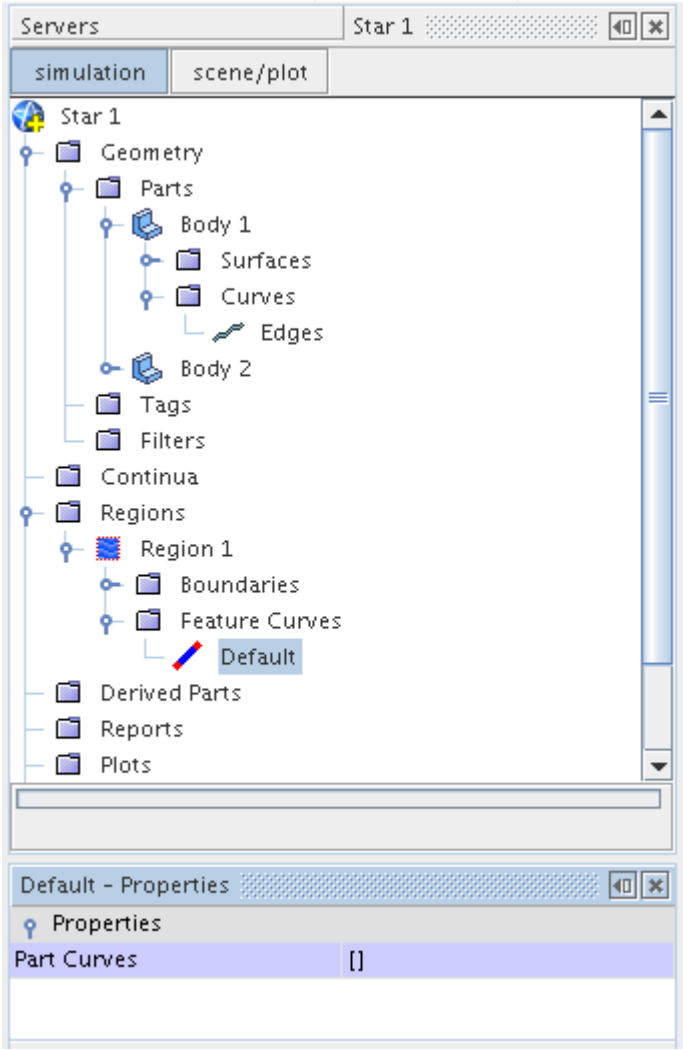


The *Feature Curve* property or the context menu item **Set Feature Curve** can be used to make the feature curve selection from the object selector (see Activating the Object Selection Dialog). The object selector only shows feature curves belonging to the region to which the parent leaf-level part is assigned. If no feature curves exist, create one before proceeding with this step.

Assigning Curves to Feature Curves Using the Region Feature Curve Nodes

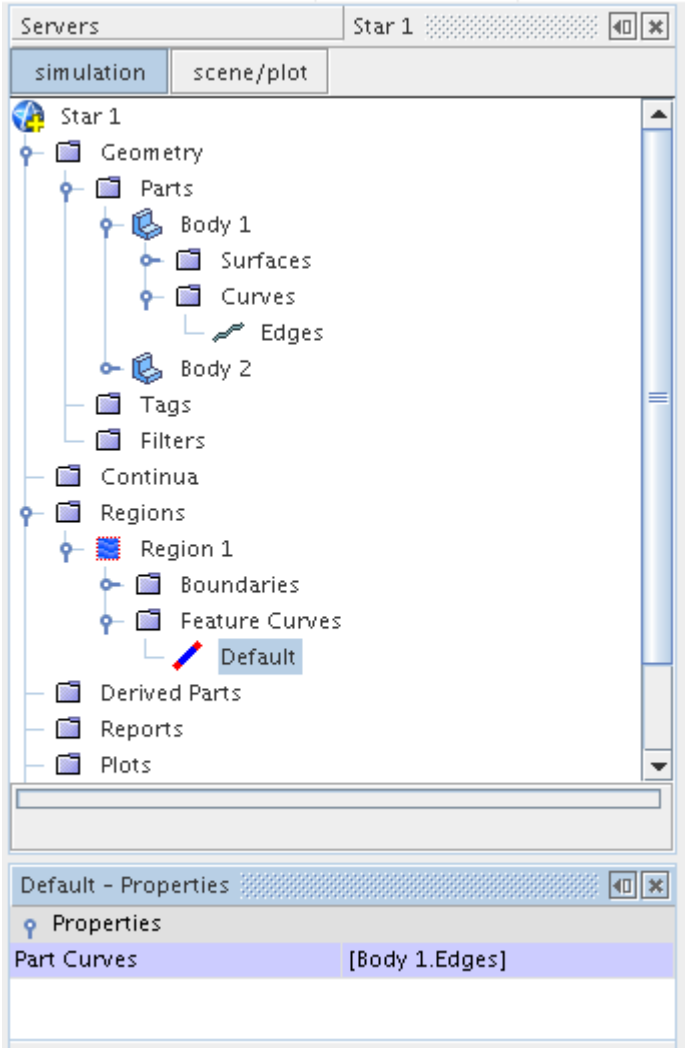
The part curves that are assigned to a region feature curve can be set using the feature curve node properties.

Selecting the feature curve node of interest display the *Part Curves* property. For example:



To assign part curves to a feature curve, click the right half of the *Part Curves* property to activate the object selector.

Select the part edges of interest and click **Close**. The *Part Curves* property now shows the part curves that are assigned to the feature curve node:



When new feature curves are created from the **Feature Curves** manager node, initially no part curves are associated with them. When part curves are associated using another method, the *Part Curves* property is automatically updated.

Initial Assignment of Boundary Types

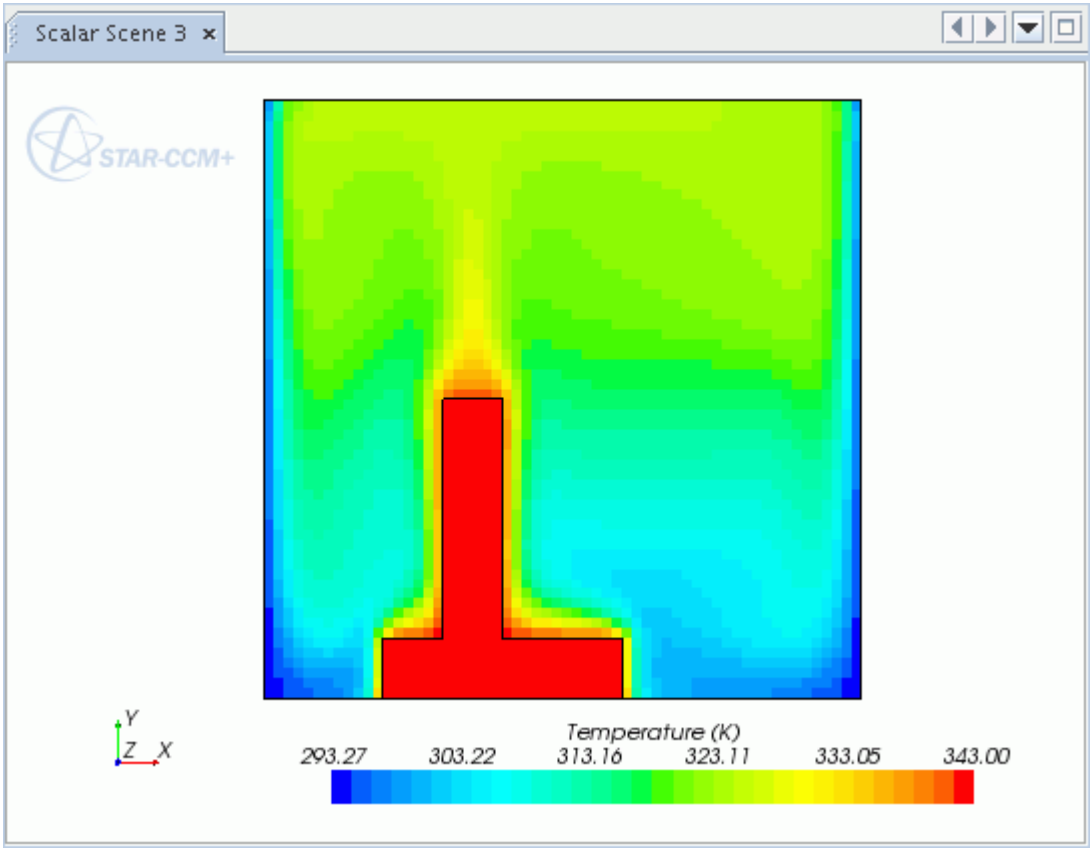
When importing a surface mesh or volume mesh as a region, STAR-CCM+ checks the names of each boundary for the text that is listed in the following table. If there is a match within the imported boundary name, the boundary is assigned the boundary type given in the second column. If there is no match, the boundary is assigned a Wall type by default.

If an imported boundary contains	It is given the following type
`massflow`	Mass Flow Boundary
`inle`	Velocity Inlet Boundary
`outl`	Outlet Boundary
`pres`	Pressure Boundary
`stag`	Stagnation Boundary
`sym`	Symmetry Boundary
`free`	Free Stream Boundary
`axis`	Axis Boundary
`(Default)`	Wall Boundary

Interfaces Overview

Interfaces are used in STAR-CCM+ to transfer solution quantities between regions in a simulation. Certain types of interfaces modify quantities passing across them, such as a porous baffle interface that presents resistance to fluid flow. Periodic and repeating interfaces allow you to reduce the scope of the solution domain.

Mass, momentum, energy, or any other simulation quantity cannot pass from one region to another unless an interface is in place between them. For example, in the simulation below, a contact interface is present between the solid and fluid regions so that the heat flux can pass between them.



STAR-CCM+ provides the following categories of interface:

Boundary-based interfaces

These interfaces are formed between a pair of boundaries that belong to either the same region or different regions. Only one pair of boundaries can be allocated to an interface. This restriction means that simulations with a high number of adjacent regions require a high number of interfaces, which can be difficult to manage.

Parts-based interfaces

For regions that are constructed from one or more geometry parts, parts-based interfaces allow you to define connections based on part contacts defined within the **Geometry** node. There is no restriction on the number of part contacts that you can allocate to a parts-based interface. For example, you can have one region that contains all the parts that represent electronic components on a motherboard. This region would have two interfaces: one for the connection between the components and the air, and another for connections between the components themselves.

Region-based interfaces

Region to region interfaces operate on all cells within the connected regions, rather than adjoining surfaces. The two types of region-based interface are:

- Overset: these interfaces facilitate overlapping meshes
- Heat exchanger: this interface type is for simulating heat transfer between cold and hot fluids in a heat exchanger

Interfaces between surfaces include contact interfaces between solid and fluid regions, internal interfaces between fluid regions, and baffle interfaces representing thin-walled separators. Interfaces that are formed between region volumes are the heat exchanger and overset mesh interfaces. A complete list of interface types is given below.

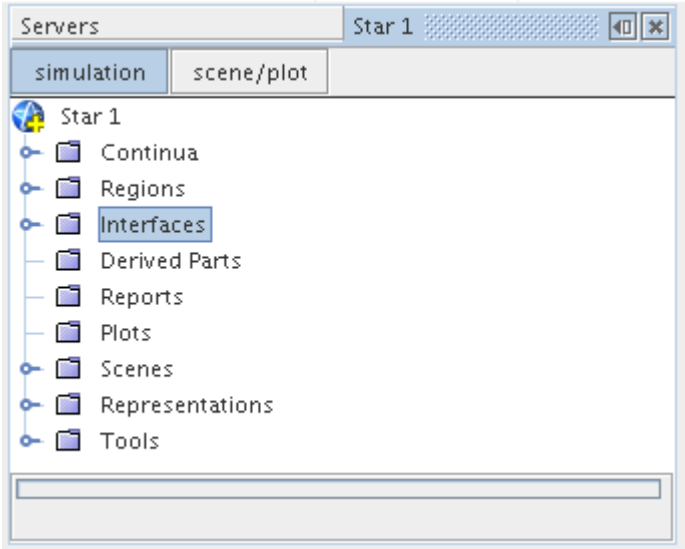
Periodic and repeating topologies can be applied to interfaces formed between boundaries. Use the interface topology property to obtain this kind of geometric transformation. Fully-developed flow and heat transfer models can be created from a fully-developed interface with a periodic topology.

Interfaces between boundaries are formed from either direct matching of faces or indirect coupling. When interfaces that use direct matching are initialized, the solver intersects faces on one side of the interface with faces on the other side. This intersection is not required if a conformal mesh is present across the interface. A conformal mesh is where a one-to-one match exists between faces and vertices on either side of the interface. This interface can be generated during the meshing process if you make appropriate selections in the meshing pipeline.

Interfaces with direct matching have a one-to-one connectivity between the intersected faces on the two associated boundaries. Hence, the interfaces with direct matching are said to have imprinted connectivity.

Indirect coupling is available for the mapped contact and mixing plane interface types. The mapped contact interface is useful for conjugate heat transfer simulations where the mesh in the solid region may be coarser than the mesh in the fluid region. Mixing plane interfaces are useful for turbomachinery calculations in steady state. Mapped contact interfaces are said to have mapped connectivity, while mixing plane interfaces are said to have connect average connectivity.

Any interfaces that are created during the simulation setup process are accessible via the **Interfaces** manager node. The **Interfaces** node has properties and a pop-up menu.



Pop-Up Menu of the Interfaces Manager

The following pop-up menu items available when you right-click the **Interfaces** manager node.

Initialize	Populates the interface boundary nodes with faces from the original boundaries.
Reset All	Removes initialization from all child interfaces. Any vertices that are removed are reported in the output window for each region.
New Group	Facilitates the grouping of objects by creating a group node within the manager node. Interface nodes can then be dragged and dropped into the new subfolder.
Group By	Facilitates the grouping of objects by moving interfaces with common properties into a property category selected in the submenu. If you then change the property for a given interface node then it does not automatically move to the appropriate subfolder.
Ungroup	Reverses the grouping process.

Interface Manager Properties

Use the properties below to adjust the specifications of the **Interface** manager.

Verbosity	Controls the level of detail output at the interface initialization. When the option is enabled, more detail is printed to the output window.
Contact Selection Priority	Lets you prioritize the assignment of contacts when assigning them with a dynamic query. See Prioritizing Assignment of Parts.
Interfaces	The number of interfaces that are created in the simulation (Read Only)

Interface Types

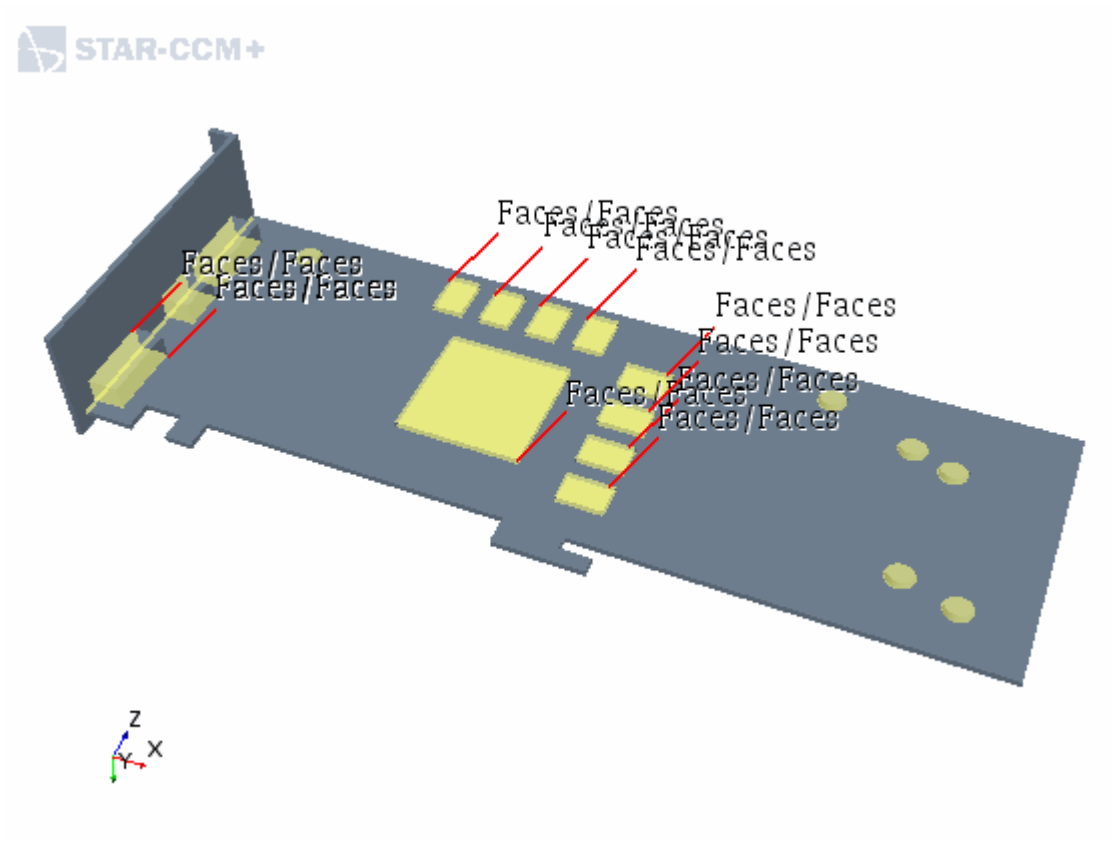
The interface type determines the behavior of the interface with respect to the modeling of flow and heat transfer. Interfaces can connect individual boundaries (faces) or entire regions (volumes). You can create the following interface types:

Type	Connects:	Connectivity
Baffle	Boundaries	Imprinted
Porous Baffle	Boundaries	Imprinted
Contact	Boundaries	Imprinted
Fan	Boundaries	Imprinted
Fluid Film	Boundaries	Imprinted
Fully Developed	Boundaries	Imprinted
Internal	Boundaries	Imprinted
Mapped Contact	Boundaries	Mapped
Mixing Plane	Boundaries	Connect Average
Blower	Boundaries	Connect Average
Heat Exchanger	Regions	Modeled
Overset Mesh	Regions	Modeled

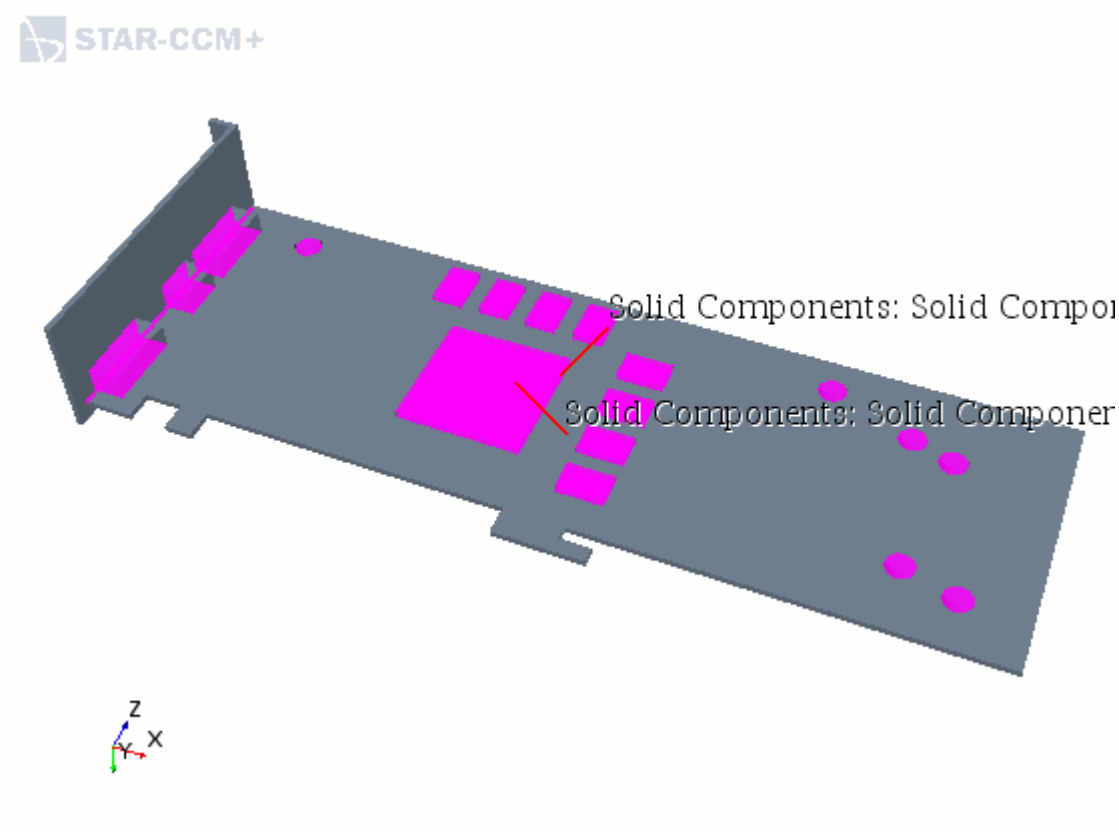
Interface Highlighting

Clicking on interfaces under the **Interfaces** node highlights the associated interfaces in the *Graphics* window.

Interfaces that are uninitialized are displayed in yellow.



Interfaces that are initialized are displayed in purple.



Contents:

Computation of Values on Interfaces
In-Place Interface Topology
Periodic Interface Topology
Repeating Interface Topology
Internal Interface
Mixing Plane Interface
Contact Interface
Mapped Contact Interface
Baffle Interface
Porous Baffle Interface
Fully Developed Interface
Orientation of Parts-Based Interfaces
Fluid Film Interface
Interface Pop Up-Menu
Deleting Interfaces
Accessing an Interface from an Interface Boundary
Converting Boundaries to Interfaces
Examining Interfaces
Inserting a Shell Region Inside an Interface

Adjusting the Interface Intersection Tolerance
Intersecting an Interface
Creating a Boundary Interface

Computation of Values on Interfaces

When you define a profile on an interface using a field function, STAR-CCM+ evaluates the field function twice (once on each interface boundary) and then averages the result.

It can be that a function that is required by the field function does not exist on both sides of the interface, in which case an error message is generated. To avoid this problem, use the *alternateValue* function in the field function definition to provide a value when one cannot otherwise be found.

For example, a scalar profile that is defined as `alternateValue($Pressure, 100000)` is evaluated as `0.5*($Pressure + 100000)` when `$Pressure` is only available on one side of the interface.

Asymmetric Values on Interface Boundaries

In some scenarios, the information that is held on the two boundaries of an interface is not the same. Depending on the type of interface, the symmetry of data that is available for input can be broken. The asymmetric interface values are as follows:

Porous Baffle Velocity Specification

For a **Resistance Based** porous baffle, the flow velocity is the same on both sides of the baffle. The flow profile is specified on the upstream side of the baffle (Boundary-0) only. The flow profile on the downstream side (Boundary-1) is computed from the values on Boundary-0 by using an appropriate transformation.

See Porous Baffle Interface.

In-Place Interface Topology

In an in-place topology, there is no physical separation in space between the two boundaries comprising the interface.

Furthermore, there is no requirement that the two boundaries have the same perimeter. After the imprinting (for surface remeshing), or intersection (for volume meshes used in an analysis) occurs the original boundaries maintain their boundary type and retain whatever surface area remains (if any).

The figure below shows different examples of in-place interfaces. The shaded red, green, and blue areas represent different mesh regions, such as porous/non-porous or fluid/solid combinations. The dotted black line in each case would represent the in-place interface that would be created between the coincident boundaries for each region. Where the areas of the original boundaries differ then remainder boundaries are produced.

When the in-place interface type is used with the surface remesher, an attempt is made at producing a conformal mesh on each side of the interface. In other words, if successful, the two boundaries when mapped from one to the other match identically in terms of the vertex locations and face topologies. The matching results in a higher-quality volume mesh definition for the analysis stage.

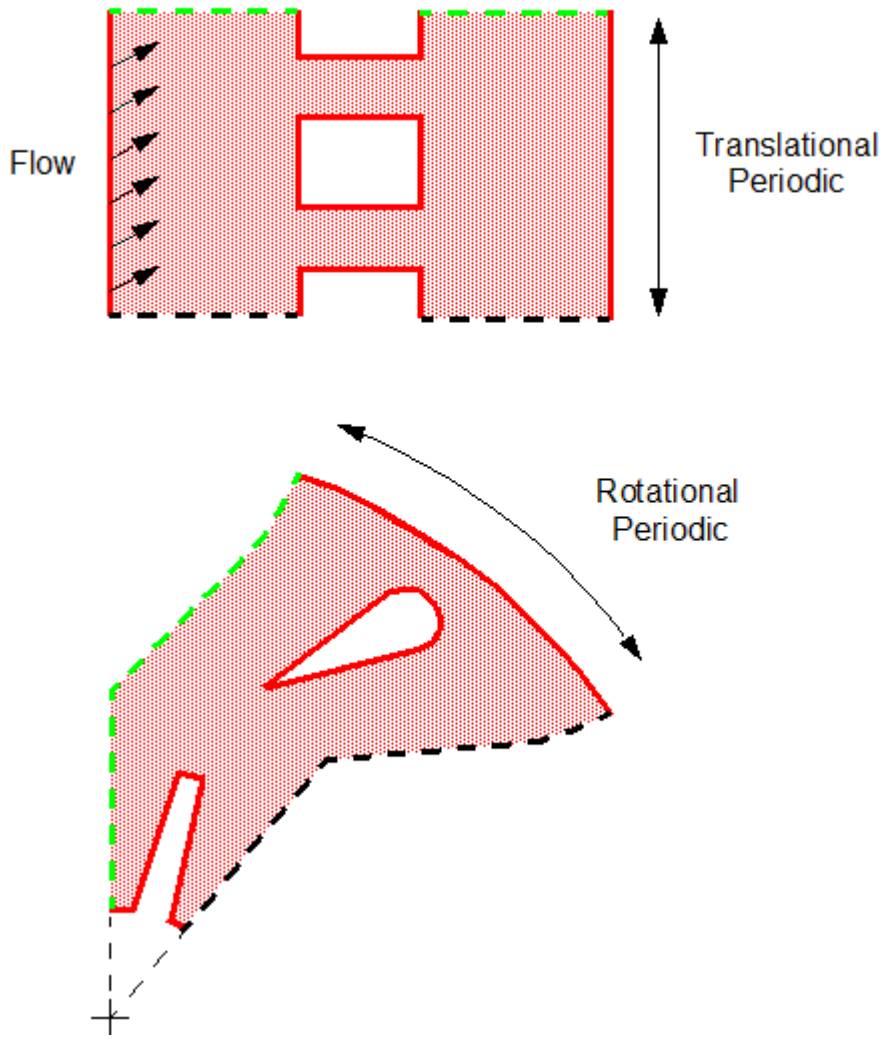
The option to produce a conformal mesh is automatically initiated when an in-place type interface has been created and the surface remesher is used afterwards. If it is not successful, then the meshing process is terminated and an error message is produced indicating the nature of the problem.

Periodic Interface Topology

The boundaries of a periodic interface are physically separated in space but can be mapped from one to the other through some constant rotation and/or translation. The periodic interface represents a cyclic repeat of information across the boundaries, so fluxes that cross one boundary are transformed and applied to the other.

This effectively approximates a repeating geometry. For example, you can model an infinitely long pipe by using a short segment of the pipe and applying a periodic interface between the two ends.

To define a periodic interface, set the perimeter of one boundary to be coincident with the transformed perimeter of the periodic matching boundary. The figures below show examples of periodic interfaces as applied to geometries that repeat in a translational and rotational manner. The green and black dashed lines in each case represent the location of the periodic interface boundary pair. For each pair, the shape and size of the boundaries match, and they are separated by a given vector or angle. The translation vector or rotation angle are calculated automatically by STAR-CCM+, and are properties of the interface **Periodic Transformation** node.



When you use a periodic interface with the surface remesher, tetrahedral, polyhedral, or thin meshers, the mesher produces a conformal mesh on each side of the interface. In other words, if successful, the two boundaries when mapped from one to the other match identically in terms of the vertex locations and face topologies. This results in a higher-quality mesh definition for the analysis stage.

Typically, the two sides of the interface are the same. However, in some cases the two sides can be different. For example, one side can consist of two patches/CAD faces, while the other side consists of just one patch/CAD face, resulting in the internal edge between them being different. An internal edge will be dropped if it fails to mate with an internal edge on the other side of the interface. In the case of CAD faces, the CAD mapping would also be dropped. The aim is to keep the faces conformal, if at all possible, by dropping less important criteria. In the above example, the less important criteria are mapping to CAD and internal feature edges.

The option to produce a conformal mesh is automatically initiated when a periodic type interface has been created and one of the above meshing tools is used thereafter. Attempts to produce a conformal mesh are made both during the surface remeshing stage and during volume mesh generation. This means that if a prepared surface mesh is imported from another source and used directly for volume mesh generation (without first resurfacing) then a conformal periodic boundary pair is still achievable within the bounds of the scheme.

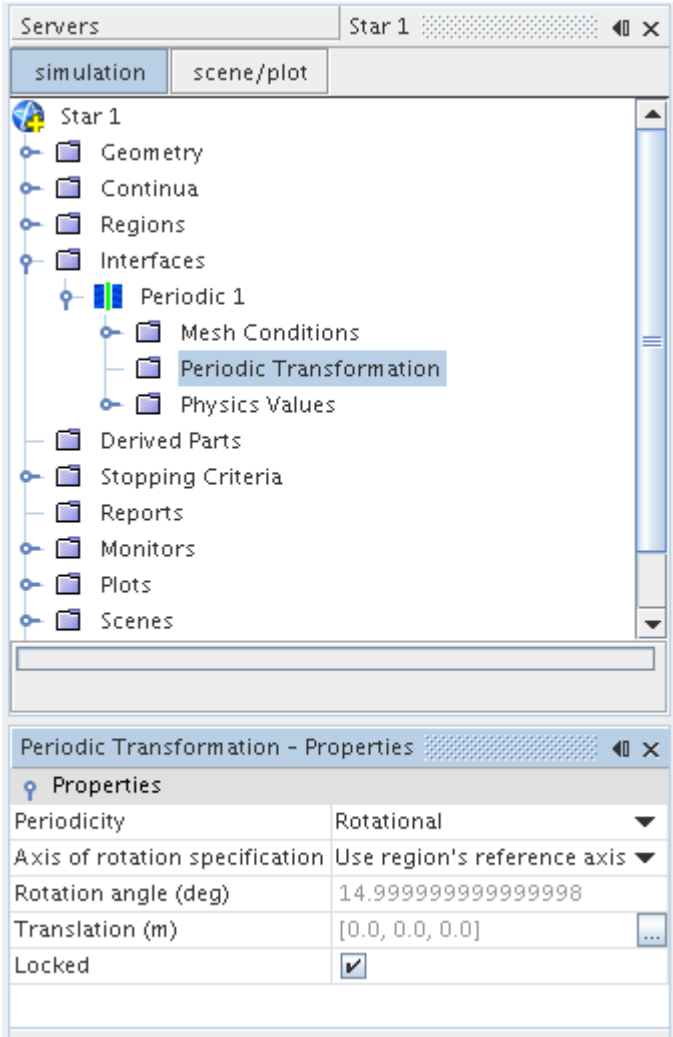
The conformal mesh mapping is limited to rotational, translational, or mixed (roto-translational) cases where a boundary pair (within the in-built tolerance) is possible. If a successful mapping is found, then the transformation matrix, number of vertex pairs and face mappings is displayed in the *Output* window.

If a transformation matrix cannot be extracted, or mesh conformity is not satisfied between the periodic boundary pair, then a warning message is printed in the *Output* window that includes the location of where these problems are.

In this instance, the meshing scheme still produces a mesh but reverts to being non-conformal, exactly as if the periodic boundary pair did not exist. The absence of a periodic match can also be confirmed by looking at the number of faces belonging to each boundary in the periodic pairing using the **Representation** node for the mesh. If the number of faces is different, then the mapping was unsuccessful.

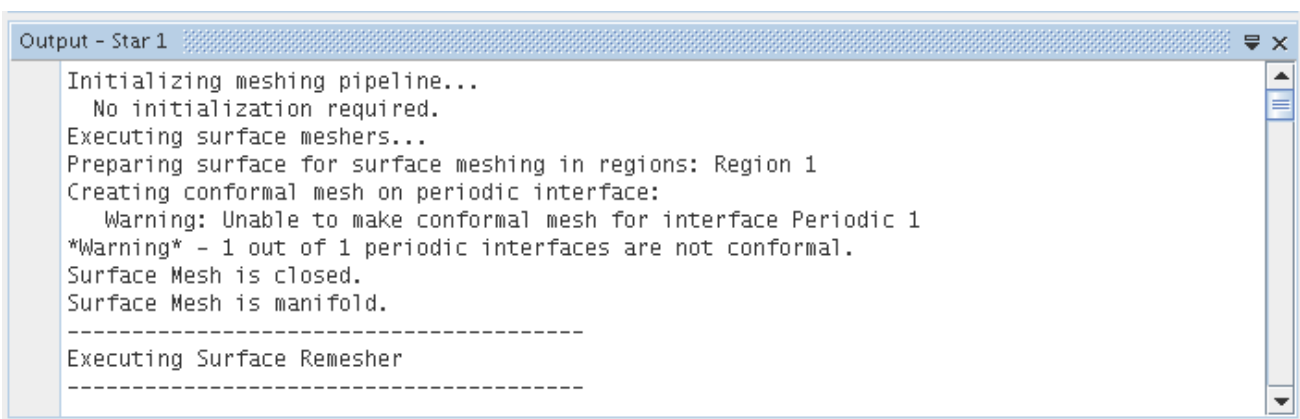
If the automatic surface repair option for the surface remesher has been activated, then periodic boundaries are excluded from this operation. Take care when modifying the surface mesh by hand once the surface remesher has been run as changes to boundaries belonging to the periodic pairing are not preserved when it comes to the volume meshing step of the process.

Once the periodic transformation properties have been supplied, the surface remesher and volume meshers can be run. These assess the information entered for the periodic interface and determine whether a conformal mesh can be produced or not. This is indicated at the start of the output log from each mesher. If successful, the calculated rotation angle and/or translational distance is displayed in the *Properties* panel:



These values are automatically “locked” for the interface node to prevent recalculation during further meshing operations. To change the values, untick the *Locked* option. The angle and translation is recalculated during meshing.

Failure to properly specify the correct periodic properties causes the conformal mesh calculation to fail as shown below:



The meshing process still continues but ultimately leads to a non-conformal mesh being produced during intersection. If a conformal volume mesh is critical then review the periodic transformation property values for errors carefully. It is also recommended that a conformal surface mesh is produced first to guarantee that the volume mesh is also conformal for the periodic interface. That is, run the surface remesher first with the interface and properties defined fully before running the volume mesher.

Periodic Transformation Properties

Periodicity

Sets the type of geometrical transformation that must be applied to vertices of the slave boundary so that they coincide with those of the master boundary. The options are:

- **Rotational+Translational**
- **Rotational**
- **Translational**

Axis of Rotation Specification

You can either choose **Specify axis** to define a unique axis for this interface using an **Axis** child node, or else **Use region's reference axis** instead. In case of the latter option, the axis properties are defined in the **Regions** > **[region]** > **Physics Values** > **Axis** node. When the periodic interface is applied to boundaries from two separate regions, the region axis from the master boundary (Boundary-0) is used.

Rotation Angle (deg)

Translation (m)

Computed transformation angle and translation distance between vertices on either side of the periodic interface.

Locked

Lock applied by the first mesh operation that calculates the transformation properties.

Repeating Interface Topology

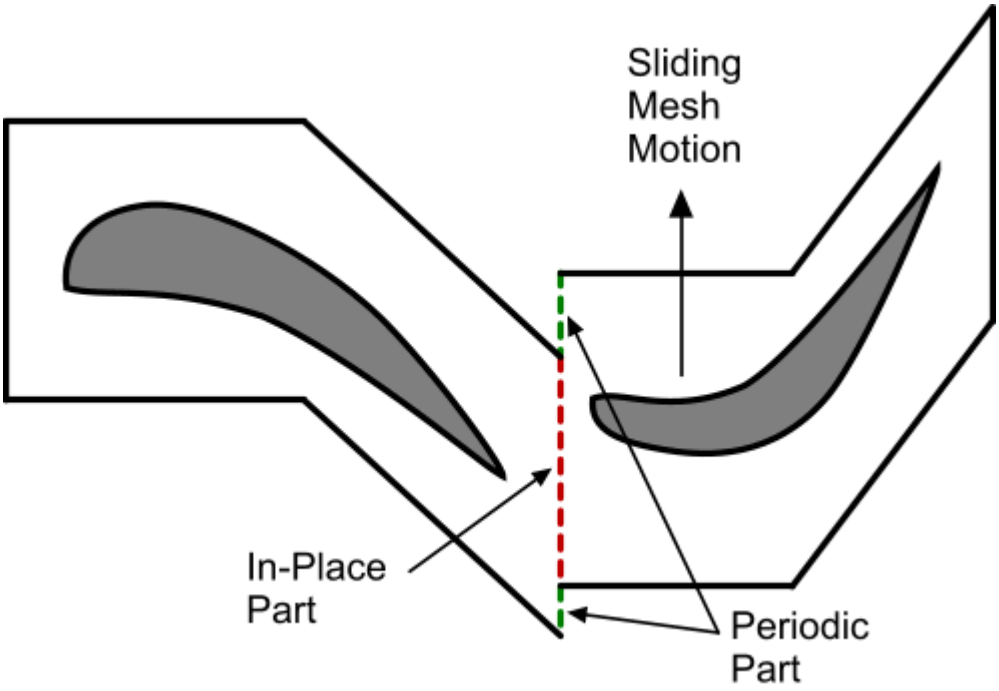
A repeating interface is a combination of in-place and periodic interfaces and is associated with periodic, sliding mesh calculations.

The two boundaries have the potential to be matched completely to form an internal interface, but the mesh motion results in one underlying boundary being indexed translationally and/or rotationally relative to the other. The parts that overlap are treated as in-place, with a periodic interface being created from the remaining boundary areas.

If the boundaries on either side have equal areas, select a repeating interface. If the areas are not equal, use a mixing plane interface instead.

The figure below shows an example of a repeating interface used to simulate rotor-stator interaction. Periodic interfaces (not labeled) cause infinite repeats of each blade row. The repeating interface between the blade row is shown in two sections: an in-place part and a periodic part. If the meshes are aligned, there is no boundary face area in the periodic part. As the mesh is moved dynamically, the mesh intersections are

recomputed. More of the face area moves from the in-place part to the periodic part, until “snap-back” occurs such that the mesh returns to its starting position.



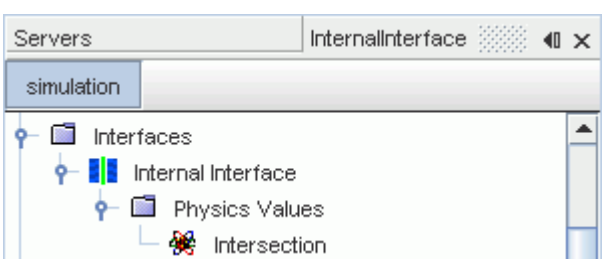
When the periodic interface type is used with the surface remesher, tetrahedral, polyhedral and/or prism mesher, then activate the per-region meshing option to mesh each region separately. A conformal mesh interface is not required in this instance since the mesh in each region is moving past one another. Alternatively, a different mesh continuum could also be used for each region.

Internal Interface

The internal interface joins two regions within the same continuum.

It can be used to combine together separate regions from the same continuum for in-place interfaces or periodic (cyclic) repeats for Periodic interfaces.

It is possible to combine regions of the same type in the same continuum using an internal interface. In certain cases, it is more efficient to use the fuse function to eliminate the boundaries altogether, resulting in a simpler topology without the interface. It has properties, values, and a pop-up menu.



Internal Interface Properties

You can use the properties that are listed below to adjust the specifications of an **Internal Interface** node.

Boundary-0	The location of the boundary on one side of the interface (Read Only)	
Boundary-1	The location of the boundary on one side of the interface (Read Only)	
Type	Defines the type of interface. Must be set to `Internal Interface`.	
Topology	Defines the connection type between the interfaces	
	In-place	Uses the in-place topology.
	Periodic	Uses the periodic topology.
	Repeating	Uses the repeating topology.
Connectivity Option	Shows how the boundaries on two sides of the interface are connected. Only **Imprinted** connectivity is possible.	

Internal Interface Values

You can use the properties that are listed below to adjust the specifications of a **Physics Values** node.

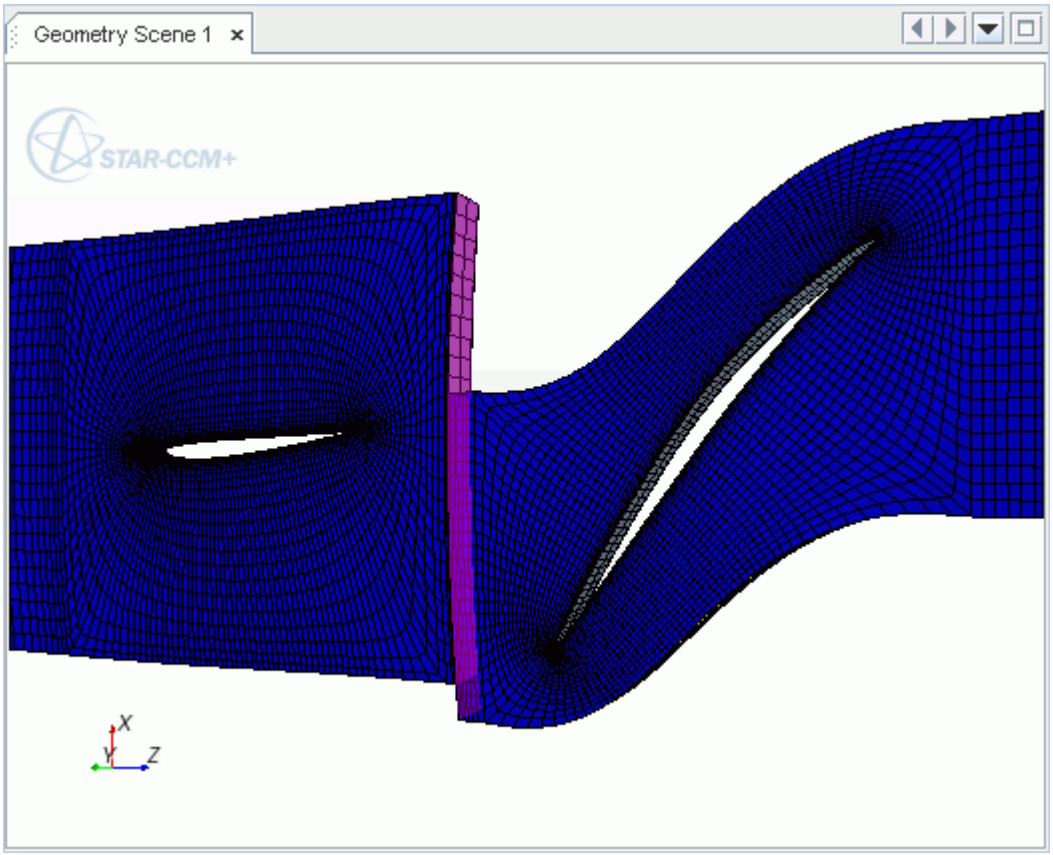
Intersection	Added for all direct interfaces to control the tolerance. The tolerance is a fraction of the smallest cell face on the secondary boundary that the secondary boundary vertices are allowed to move during an intersection. If the intersection fails, the following message is displayed in the output window - `Warning: no intersections found`. The solution cannot proceed under these conditions.

Mixing Plane Interface

The mixing-plane interface type is used to compute steady-state simulations of multi-stage turbomachinery flows.

This interface makes it possible to simulate the flow across stages with different pitches moving relative to one another. At the interface, circumferentially averaged flow field data is transferred in a conservative manner between two rotationally periodic regions. These regions must be of the same continuum, and aligned on the same axis. The mixing plane interface type is available on indirect interface topologies only.

The mixing-plane interface allows the transfer of mass, momentum, energy, and other conserved quantities across the interface using a circumferential averaging technique. The values to be transferred across the interface are averaged in concentric, circumferential rings, or bins, of uniform radial thickness, covering the entire span and pitch of the boundary. Due to the circumferential averaging, the final solution is characterized by radially varying profiles of flow field quantities that remain uniform across the pitch within any given circumferential bin.



Note: A mixing-plane interface cannot be used in combination with multi-component VOF phases. STAR-CCM+ supports the use of mixing-plane interfaces with single-component VOF phases only.

Mixing-plane interfaces can be used with the Multiphase Segregated Flow model. However, a mixing-plane interface cannot be used in combination with granular flow models.

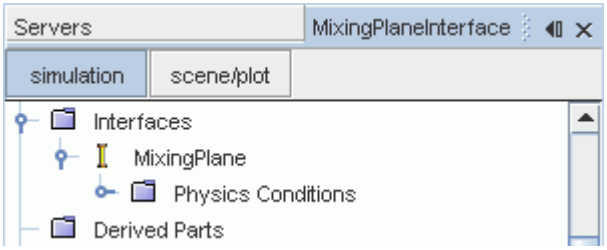
Implicit vs. Explicit Mixing-Plane Solution Approaches

The solution across the mixing-plane can be computed using either an implicit or an explicit approach.

In the implicit mixing-plane approach, the cells on either side of the interface are directly connected through each circumferential bin. The approach introduces additional degrees of freedom to the linear system – one additional degree of freedom per variable per circumferential bin. These degrees of freedom represent the conservation of fluxes across the faces composing either side of a given bin. In this way, solution of the linear system results in implicit updates to circumferentially averaged values on either side. The implicit mixing-plane approach is available with both Coupled and Segregated Flow models.

The explicit mixing-plane approach closely follows the procedure used in the Non-Reflecting Boundary Condition. Here, updated values on the interface boundaries are computed directly, based on those from the previous iteration. As in the Non-Reflecting Boundary Condition, all outgoing waves pass through the interface without reflection. Circumferential averaging of conserved quantities, combined with piecewise constant interpolation between bins in the radial direction, is used to prescribe the zero-th mode solution required by the non-reflecting boundary condition. The explicit mixing-plane approach is available in combination with the Coupled Flow model only. See [207], [208], [209] and [235].

The mixing plane interface has properties, conditions, and a pop-up menu.



Mixing Plane Interface Properties

You can use the properties that are listed below to adjust the specifications of a **Mixing Plane Interface** node.

Boundary-0

The location of the boundary on one side of the interface. (Read only.)

Boundary-1

The location of the boundary on one side of the interface. (Read only.)

Type

Defines the type of interface. Must be set to **Mixing Plane Interface**. However, when using the Regions Based Meshing (RBM) approach, with the mesh property *Per-Region Meshing* deactivated, set this *Type* property to **In-Place Interface** until the mesh is created, then set this *Type* property to **Mixing Plane Interface**. This process makes sure that the mesh that is created at the interface is non-conformal.

Topology

Defines the connection type between the interfaces. Only the **Indirect** type is possible.

Connectivity Option

Shows how the boundaries on two sides of the interface are connected. Only **Connect Average** connectivity is possible.

Mixing Plane Interface Conditions

You can use the properties that are listed below to adjust the specifications of a **Physics Condition** interface node. You can specify either explicit or implicit methods.

Connectivity Option

Option

Provides energy source options for the interface.

Implicit

Uses the implicit mixing plane approach. This is the default.

Explicit

Uses the explicit mixing plane approach.

Bin Coarsening Factor

This expert property guides the number of circumferential bins (that is, constant-thickness rings or bands) used to perform the circumferential averaging ("mixing-out") of flow field data across the implicit mixing plane interface. Increase the coarsening factor to produce fewer, coarser (wider) bin rings for use in cases with disparate mesh densities on either side of the interface. The actual number of bins is set by STAR-CCM+; the factor only has an effect if the boundary meshes do not initially yield an equal number of bins on both sides of the implicit mixing plane interface. The default value is 1.

Mixing Plane Option

Option

Activates or deactivates the non-reflecting boundary condition at mixing-plane interface boundaries. Available only with the Coupled Flow model, when **Connectivity Option** is set to **Explicit**. When the Harmonic Balance Flow and Energy model is active, the **Connectivity Option** is always set to **Explicit** and the non-reflecting boundary conditions always applies at the mixing plane interface. See Modeling Non-Reflecting Boundaries and Mixing-Plane Interfaces.

None

Deactivates the non-reflecting boundary condition at mixing-plane interface boundaries. Circumferentially averaged conserved quantities at each radial location prescribe the boundary condition at the two interface boundaries belonging to the mixing plane interface.

Non-Reflecting

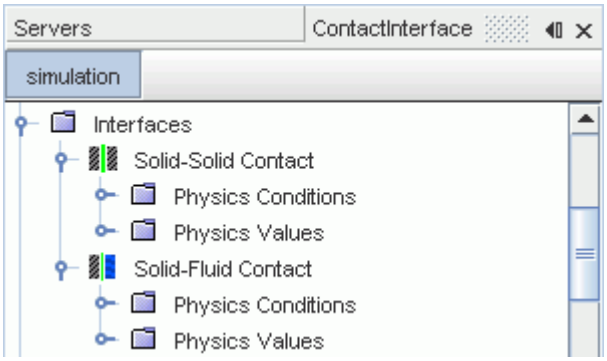
Activates the non-reflecting boundary condition at mixing-plane interface boundaries. Circumferential averaging of conserved quantities, combined with piecewise constant interpolation between bins in the radial direction, prescribe the zero-th mode solution required by the non-reflecting boundary condition. The higher modes are treated using the non-reflecting boundary condition that allows the outgoing waves to pass through the mixing plane without reflecting back into the computational domain. This setting is the default.

Contact Interface

A contact interface joins two solid regions, which can be from the same continuum or from different continua, or joins a solid region and a fluid region.

The contact interface type is used to permit conjugate heat transfer between the regions. A contact interface can also be used to specify a surface charge at the interface between a gas and a dielectric (non-conducting) material.

It has properties, conditions, values, and a pop-up menu.



Modeling

- A thermal contact resistance per unit area, R, for conjugate heat transfer can be specified. The relation between the heat flux across the contact interface, $\dot{q}''$, and the temperature difference, ΔT, is given by:

$$\dot{q}'' = \frac{\Delta T}{R} \tag{78}$$

Contact Interface Properties

You can use the properties that are listed below to adjust the specifications of a **Contact Interface** node.

Geometry	Specifies the geometry source option: • **Boundaries**: The interface is defined by the **Boundary-0** and **Boundary-1** selections. • **Contacts**: The interface is defined by the **Contacts** selections.
Boundary-0	The location of the boundary on one side of the interface (Read Only)
Boundary-1	The location of the boundary on one side of the interface (Read Only)
Contacts	Defines the part surface contacts. This property is available only when the Geometry property is set to **Contacts**.
Type	Defines the type of interface. Must be set to `Contact Interface`
Topology	Defines the connection type between the interfaces: • **In-place** • **Periodic** • **Repeating**
Connectivity	Shows how the boundaries on two sides of the interface are connected. Only **Imprinted** connectivity is possible.
Allow Per-Contact Values	Activates contact-based profiles. This property is available only when the Geometry property is set to **Contacts**.

Contact Interface Conditions

Energy Source Option	Provides energy source options for the interface.

	None	Do not specify an energy source.
	Heat Flux	Specifies a user-defined heat flux in W/m^2.
	Heat Source	Specifies a user-defined heat source in W.
Surface Electric Charge Option	Provides surface electric charge options for the interface. This option becomes available when the Electromagnetism and Electrostatic Potential models are activated in the physics continuum.	
	None	Do not specify a surface electric charge.
	Surface Charge Density	Specifies a user-defined surface electric charge in $s\text{-}A/m^2$.
External Load	Specifies the method that is used to define the load. Only available for fluid-solid contact interfaces when the Solid Stress model is activated.	
	Force	This option adds a Force node to the **Physics Values** node, which must be entered as a vector profile.
	Pressure	This option adds a Solid Stress Pressure node to the **Physics Values** node, which must be entered as a scalar profile.
	Traction	This option adds a Traction node to the **Physics Values** node, which must be entered as a vector profile.
Mechanical Interaction	Specifies the type of contact between the two solid regions.	
	Small Sliding Frictionless	The contacting surfaces are free to slide over each other, however movement is prohibited normal to the interface. The shear stress at the contact interface is equal to zero, and the normal forces are equal and opposite. No additional node is added to the tree.
	Bonded	The contacting surfaces undergo continuous displacements. If the solid regions are different materials, discontinuities in the strains are preserved across the material interface. No additional nodes are added to the tree.

Thermal Specification	Specifies the thermal conditions at the interface.	
	Conjugate Heat Transfer	This option adds a Contact Resistance node to the **Physics Values** node. The temperature at the interface is determined from the heat transferred through the interface.
	Specified Temperature	This option adds a Static Temperature node to the **`Physics Values`** node. The temperature at the interface is the same for both parent boundaries, as indicated by the selected profile.

Contact Interface Values

Contact Resistance	The value of the resistance to conduction through the interface. It is entered as a scalar profile. If a field function defines the value and the Ignore Boundary Values property on the field function is ticked, the function is computed using data from the cell next to the boundary. Contact resistance is specified in units of $m^2 \cdot K/W$ (the default) or equivalents.
Heat Flux	Specify the heat transferred per unit area. Activated when the **Energy Source Option** is set to **Heat Flux**. Heat flux is specified in units of W/m^2 (the default) or equivalents.
Heat Source	Specify the total rate of heat transfer through the interface. Activated when the **Energy Source Option** is set to **Heat Source**. Heat source is specified in units of W (the default) or equivalents.
Intersection	Added for all direct interfaces to control the tolerance. The tolerance is a fraction of the smallest cell face on the secondary boundary by which the secondary boundary vertices are allowed to move during an intersection. If the intersection fails, the following message is displayed in the output window - `Warning: no intersections found`. The solution cannot proceed under these conditions.
Static Temperature	Specify the temperature at the interface. Activated when the **Thermal Specification** option is set to **Specified Temperature**.
Surface Charge Density	Specify the surface charge density profile. Activated when the **Surface Electric Charge Option** is set to **Surface Charge Density**. The surface electric charge is specified in units of s-A/m^2 (the default) or equivalents.

Mapped Contact Interface

The Mapped Contact Interface is an indirect interface type for heat exchange at fluid/solid and solid/solid boundaries.

This interface type is a variation of contact interface type applicable to indirect interfaces. The mapped contact interface relies on an indirect association between faces of the interfacing boundaries, for data mappers, rather than an imprinted connection, for direct interfaces.

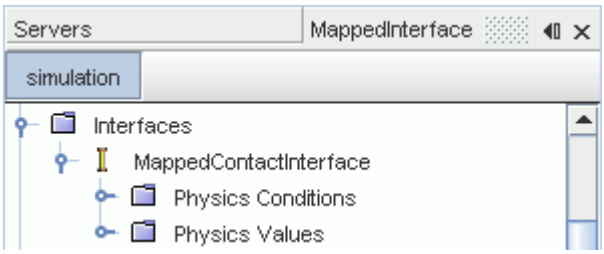
The Mapped Contact Interface is created on the volume meshes generated by the meshing process. When the interface is created from two boundaries, two new interface boundaries are automatically added. The original boundaries from which the interface is created are not destroyed.

Each face of the original boundary is linked with a group of faces on the opposing boundary using a proximity-based algorithm. Once the associativity of all the faces on the original boundary has been processed, the faces are split into an associated face-set and a disassociated face-set.

The associated face-set is transferred on to the interface boundary, while the unassociated face-set is retained on the original boundary. This is analogous to the creation of a direct interface, except that this process does not result in any new faces.

The main advantage of a Mapped Contact Interface is that it allows for a non-conformal mesh across the interface. This type of mesh is desirable in cases where the interfacing boundaries have different mesh resolutions. For instance, the heat conduction in the solid region requires a coarser mesh compared to resolving the boundary layer in the fluid domain. By precluding the need for imprinting, this approach retains the high quality of the originally created mesh and also reduces the resultant face count. A co-product of this approach is better accuracy in computing the face gradients as well as better quality mesh for visualization. This approach is also better suited for cases where the geometric features are not equally resolved on either side of the interface. While imprinting may leave more non-intersected faces, this technique can find an associated face on the opposing boundary more robustly.

The Mapped Contact Interface has properties, conditions, values, and a pop-up menu.



Mapped Contact Interface Properties

You can use the properties that are listed below to adjust the specifications of a **Mapped Contact Interface** node.

Boundary-0	The location of the boundary on one side of the interface (Read Only)
Boundary-1	The location of the boundary on one side of the interface (Read Only)
Type	Defines the type of interface. Must be set to `Mapped Contact Interface`

Topology	Defines the connection type between the interfaces. Only the **Indirect** type is possible.
Connectivity Option	Shows how the boundaries on two sides of the interface are connected. Only **Mapped** connectivity is possible.

Mapped Contact Interface Conditions

You can use the properties that are listed below to adjust the specifications of a **Physics Condition** interface node.

Energy Source Option	Provides energy source options for the interface.	
	None	Do not specify an energy source.
	Heat Flux	Specifies a user-defined heat flux in W/m^2.
	Heat Source	Specifies a user-defined heat source in W.
External Load	Specifies the method that is used to define the load. Only available for fluid-solid contact interfaces when the Solid Stress model is activated.	
	Force	This option adds a **Force** node to the **Values** node, which must be entered as a vector profile.
	Pressure	This option adds a **Solid Stress Pressure** node to the **Values** node, which must be entered as a scalar profile.
	Traction	This option adds a **Traction** node to the **Values** node, which must be entered as a vector profile.
Mechanical Interaction	Specifies the type of contact between the two solid regions.	
	Bonded	The contacting surfaces undergo continuous displacements. If the solid regions are different materials, discontinuities in the strain are preserved across the material interface.
	Small Sliding Frictionless	The contacting surfaces are free to slide over each other, however movement is prohibited normal to the interface. The shear stress at the contact interface is equal to zero, and the normal forces are equal and opposite.

Mapped Contact Interface Values

You can use the properties that are listed below to adjust the specifications of a **Physics Values** interface node.

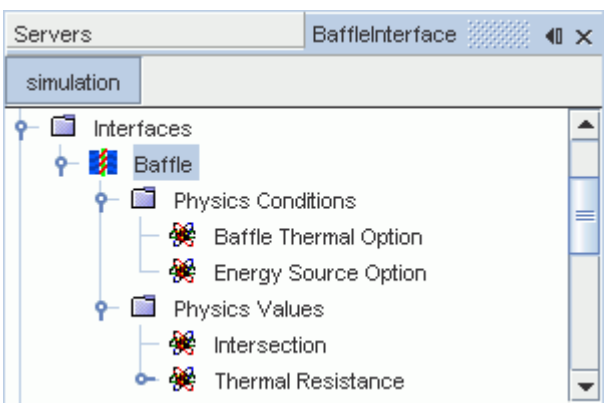
Contact Resistance	The value of the resistance to conduction through the interface. It is entered as a scalar profile. If a field function defines the value and the Ignore Boundary Values property on the field function is activated, the function is computed using data from the cell next to the boundary identified by the *Boundary-0* property of the interface.	
	Dimensions	Specify the resistance in units of $m^2 \cdot K/W$ or similar. (Read Only)
	Method	Specify the input method type from **Constant** , **Field Function** , **Table** , or **User Code** .
Heat Flux	Specifies the heat transferred per unit area. It is activated when the **Energy Source Option** is set to **Heat Flux**.	
	Dimensions	Specify the heat flux in units of W/m^2 or similar. (Read Only)
	Method	Specify the input method type from **Constant** , **Field Function** , **Table** , or **User Code** .
Heat Source	Specifies the heat transferred. It is activated when the **Energy Source Option** is set to **Heat Source**.	
	Dimensions	Specify the heat source in units of W or similar. (Read Only)
	Method	Specify the input method type from **Constant** , **Field Function** , **Table** , or **User Code** .
Interpolation Stencil	Specifies the stencil that is used to map the values.	
	Compact Stencil	This scheme builds a connectivity map by finding the nearest neighbor of a face. The scheme includes a symmetry specification step which can add extra faces to the stencil (that is, if a face on boundary 0 sees a face on boundary 1, then the face on boundary 1 will also see the face on boundary 0). After the connectivity map is created, face area ratios are used to define the interpolation weights.

Mapped Interface Tolerance	Activates the search algorithm and specifies tolerances that the algorithm uses to find the neighbors to the cells.	
	Proximity Check	When activated, the minimum proximity check occurs.
	Minimum Proximity	The relative distance (based on cell size) to the target cell. 1 is equivalent to 100%.
	Normal Directions check	When activated, the normal direction check occurs.
	Minimum Angle	The angle, in degrees, between the source and target face normals. When the minimum angle is 0, the two normals point in the same direction, when 180, the normals point toward each other.

Baffle Interface

A baffle interface physically represents one or more thin sheets of impermeable, conducting materials in a fluid.

It can be placed between regions of the same continuum or of different continua. It has properties, conditions, values, and a pop-up menu.



Modeling

- The baffle has infinitesimal thickness.
- The fluid next to a baffle treats the baffle as a wall. It applies the non-slip boundary condition in viscous flow, and the component of the velocity normal to the baffle is zero.
- A thermal resistance per unit area R for conjugate heat transfer can be specified, either as effective resistance or with multiple layers. The relation between the heat flux across the contact interface, $\ddot{q}$, and

the temperature difference, ΔT, is given by Eqn. (78). To calculate the thermal resistance of the baffle, the following formula can be used:

$$\frac{1}{R} = \sum_{i}^{n} \frac{k_i}{\Delta x_i} \tag{79}$$

where k_i and Δx_i are the conductivity and thickness, respectively, of each of the n layers comprising the baffle.

- If it is not required that heat transfer through the baffle is modeled, do not create a baffle interface. The same functionality could be achieved by assigning wall boundary types (with adiabatic thermal conditions) to the boundaries in question.

Baffle Properties

You can use the properties that are listed below to adjust the specifications of a **Baffle** interface node.

Boundary-0	The location of the boundary on one side of the interface (Read Only)	
Boundary-1	The location of the boundary on one side of the interface (Read Only)	
Type	Defines the type of interface. Must be set to `Baffle Interface`.	
Topology	Defines the connection type between the interfaces	
	In-place	Uses the in-place topology.
	Periodic	Uses the periodic topology.
	Repeating	Uses the repeating topology.
Connectivity Option	Shows how the boundaries on two sides of the interface are connected. Only **Imprinted** connectivity is possible.	

Baffle Conditions

You can use the properties that are listed below to adjust the specifications of a **Physics Condition** interface node.

Baffle Thermal Option	Specifies whether the baffle conducts energy.	
	Conducting	Thermal conduction occurs through the baffle. You would typically select this option when you use an energy model and a baffle or porous baffle interface. A **Thermal Resistance** interface value is added.
	Non-conducting	No thermal conduction occurs through the baffle.
Energy Source Option	Provides energy source options for the interface. It is activated when the **Baffle Thermal Option** is set to *Conducting*.	
	None	Do not specify an energy source.

	Heat Flux	Specifies a user-defined heat flux in W/m^2.
	Heat Source	Specifies a user-defined heat source in W.

Baffle Values

You can use the properties that are listed below to adjust the specifications of a **Physics Values** interface node.

Thermal Resistance	The value of the resistance to conduction through the baffle. It is activated when the **Baffle Thermal Option** is set to **Conducting**. A **Thermal Resistance** boundary value is added to the object tree when the Thermal Specification boundary condition is set to Environment.	
	Dimensions	Specify the resistance in units of $m^2 \cdot K/W$ or similar. (Read Only)
	Method	Specify the input method type from **Constant**, **Field Function**, **Table**, **User Code**, or Multi-Layer Resistance.
Heat Flux	Specify the heat transferred per unit area. It is activated when the **Energy Source Option** is set to **Heat Flux**.	
	Dimensions	Specify the heat flux in units of W/m^2 or similar. (Read Only)
	Method	Specify the input method type from **Constant**, **Field Function**, **Table**, or **User Code**.
Heat Source	Specify the heat transferred. It is activated when the **Energy Source Option** is set to **Heat Source**.	
	Dimensions	Specify the heat source in units of W or similar. (Read Only)
	Method	Specify the input method type from **Constant**, **Field Function**, **Table**, or **User Code**.
Intersection	Added for all direct interfaces to control the tolerance. The tolerance is a fraction of the smallest cell face on the secondary boundary that the secondary boundary vertices are allowed to move during an intersection. If the intersection fails, the following message is displayed in the output window - `Warning: no intersections found`. The solution cannot proceed under these conditions.	

Baffle Values: Multi-Layer Resistance

This **Thermal Resistance** value is useful when, for example, you are simulating conduction through multiple materials that vary in conductivity. It is calculated using Eqn. (79).

This object is also available in the Thermal Resistance boundary value, which is added to the object tree when the Thermal Specification boundary condition is set to Environment.

Multi-Layer Resistance Properties

Number of layers	Specifies the number of layers within the baffle.
Thickness of each layer	Sets a thickness for each of the layers. The number of comma-separated entries must match the number of layers. If the number of layers is decreased, the last value is dropped. If a layer is added, a zero value is appended by default.
Conductivity of each layer	Specifies the conductivity through a series of comma-separated values equal to the number of layers. STAR-CCM+ works with discrepancies between these specifications as follows: • If you make more conductivity value entries than layers, for example seven values and five layers, then STAR-CCM+ works with the first five values. • If you make fewer conductivity value entries than layers, for example three values and five layers, STAR-CCM+ interprets values for the remaining layers as zero conductivity.

Porous Baffle Interface

Physically, a porous baffle represents a porous membrane through which fluid passes and experiences a pressure drop.

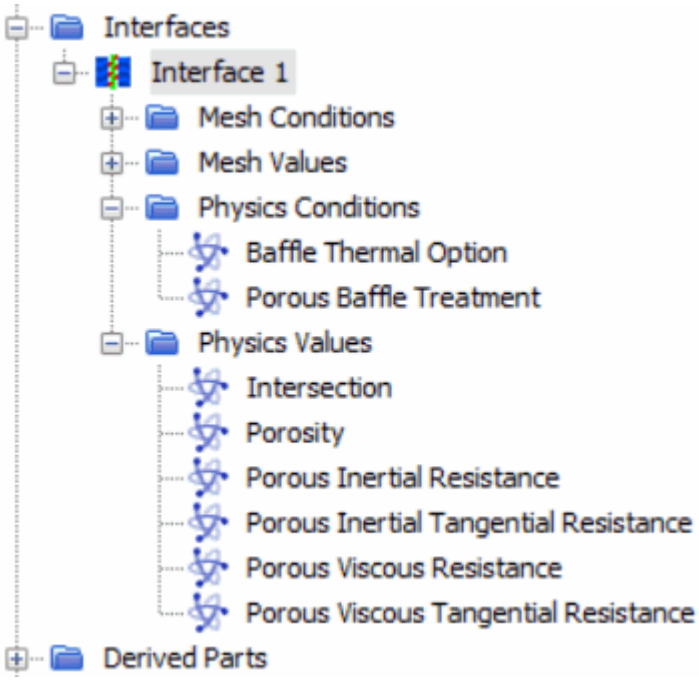
Conductive heat transfer is modeled in the impermeable portion of the baffle. Porous baffles can be used to model perforated plates, thin screens, and wire screens. The porous baffle interface node has properties and a pop-up menu.

Porous baffles are compatible with the following:

- Single-phase segregated flow and single-phase coupled flow.
- Multiphase segregated flow. The Volume of Fluid and Eulerian Multiphase Mixture models are compatible with shear-based porous baffles only.
- K-epsilon, k-omega, and RSM turbulence models.

Applications for the Porous Baffle models include flow-straightening devices and three-phase separators.

The example that is shown below is for a single-phase flow. A porous baffle that is used in a multiphase flow does not have the **Physics Values** > **Porosity** node, but is otherwise identical.



- Use this interface type between regions belonging to the same fluid continuum.
- When debugging it is appropriate to report mass flow rates and pressures on the interface boundaries. Reporting on interfaces is more representative than creating derived parts nearby.
- For multiphase flows using the VOF model, porous baffle interfaces have a conformal (that is, matching) mesh at locations where phases of different density are expected to cross the baffle. If this requirement is not observed, convergence problems could arise.
- Remember that a baffle is a “lumped” model; when modeling a baffle, you determine the bulk effects that the baffle has on the flow, for example, pressure drop, not the fine details. If you want to compute the flow coming out from the little openings next to the baffle, model the actual geometry, with these small openings, and a fine mesh. If you are interested only on the bulk effect of the baffle downstream, then you can appropriately use the porous baffle interface in your model.

Porous baffles have the following limitations and incompatibilities:

- All Eulerian Multiphase models except Multiphase segregated flow are incompatible with resistance-based porous baffles.
- The Spalart Allmaras turbulence model is incompatible with resistance-based porous baffles.
- There is no treatment for heat transfer in the baffle, but the simulation can include heat transfer elsewhere.
- Incompatible with Harmonic Balance.
- Filtration modeling is not possible because no phase-specific resistance can be set. The resistance is applied to the multiphase mixture, not to individual phases.

Modeling

STAR-CCM+ provides two forms of porous baffle.

- **Resistance Based**

This baffle treatment is suitable for modeling open baffles where the flow is predominantly normal to the surface, such as that found in grills and screens.

For turbulent flows, this option is appropriate if the holes in the baffle are such that their characteristic length is similar to the length scale of the turbulence.

- **Shear Stress Based**

  This baffle treatment, sometimes referred to as a wall-like or a wall-based porous baffle, is best suited to applications where the flow is mostly tangential to the baffle. This option is also appropriate if the velocities at the two sides of the porous baffle interface are different.

  For turbulent flows, this option is appropriate if the turbulence has a scale that reduces significantly at the baffle.

STAR-CCM+ models a porous baffle as follows:

- The baffle has infinitesimal thickness.
- If the porous baffle is thought of as a perforated surface, then the ratio of the area of the holes to the area of the surface, is called the porosity, χ. Wire screens would have a large value of χ, while something like a perforated plate would have a smaller value.
- Each interface has a direction from side 0 to side 1. When you create the interface, select side 0 first, then side 1. For a porous-baffle interface, the upstream is side 0. Therefore if you select the sides in the wrong order, the sign of the pressure jump is reversed.
- Turbulence generation at the baffle depends on the ratio of the pore size to the characteristic turbulent scales. If the baffle pores are much smaller than the turbulent scales, the baffle behaves like a wall, so the Shear Stress Based baffle is appropriate. If the baffle pores are larger, the turbulent eddies can penetrate through the baffle (which has a structure similar to a grill, rather than a porous plate), so the Resistance Based baffle is appropriate.
- For a shear-based porous baffle, the viscous shear stress is approximated to be the same as the shear stress at the interface assuming it to be a solid wall. The porosity, χ, is used only for heat and mass transfer calculations at the porous baffle surface and is not used for flow calculation.

  For a resistance-based porous baffle, the viscous stresses depend on the tangential resistance coefficients.
- For heat transfer analysis, the porous baffle offers a finite thermal resistance. Conduction is modeled as a one-dimensional process and thermal inertia is neglected. A thermal resistance per unit area, R, for heat transfer can be specified. The relation between the heat flux across the contact interface, $\dot{q}''$, and the temperature difference, ΔT, is given by Eqn. (78). Estimate the thermal resistance using Eqn. (79).
- The total heat transfer across the porous baffle is computed as a linear combination of the heat transfer due to convection and diffusion in the fluid and the conduction through the baffle. The porosity, χ, provides the linear weighting factor. This calculation is given by:

$$\dot{q}'' = \left(1 - \chi\right)\frac{\Delta T}{\left(\frac{1}{h_1} + R + \frac{1}{h_2}\right)} + \chi(\Gamma \nabla E)_f - \rho_f v_f E_f \tag{80}$$

  where h_1 and h_2 are the upstream and downstream heat transfer coefficients, respectively.
- It is assumed that the direction of the flow is unchanged as it passes through the baffle.

Setting up a Porous Baffle

To set up a porous baffle:

1. Set the interface *Type* property to **Porous Baffle Interface**.
2. In the Physics Conditions of the interface, specify the **Porous Baffle Treatment** method:
   - **Resistance Based**
   - **Shear Stress Based**

   See Porous Baffle Physics Conditions.
3. Specify the appropriate Physics Values.

   For a **Resistance Based** porous baffle, both normal resistance and tangential resistance is applied to the baffle. Set the following:

   - **Porous Inertial Resistance**
   - **Porous Inertial Tangential Resistance**
   - **Porous Viscous Resistance**
   - **Porous Viscous Tangential Resistance**

   For a **Shear Stress Based** porous baffle, normal resistance only is applied to the baffle.

   - **Porous Inertial Resistance**
   - **Porous Viscous Resistance**

   See Porous Baffle Physics Values.
4. For a **Resistance Based** porous baffle, if a turbulence model is activated in the physics continuum, you also specify the following:

   - **Turbulence Intensity**
   - **Turbulent Length Scale**

     The length scale of the turbulence is related to the characteristic geometry scale of the baffle itself, for example, the perforation diameter.

   The turbulence specification in a porous baffle is not a boundary condition, but is a source, in the same way as for Particle-Induced Turbulence. It can be considered as particle-induced turbulence, where the particles are the fixed components of the baffle.

   See Porous Baffle Physics Values.
5. Specify any tangential velocity on the porous baffle surface.

   This step is necessary if the baffle is rotating, for example.

   For a **Resistance Based** porous baffle, the flow velocity is the same on both sides of the baffle. You specify the flow velocity on the upstream side of the baffle. For a **Shear Stress Based** porous baffle, you can specify different flow velocities on either side of the porous baffle.

   Set the appropriate values in the **Regions** > **[region]** > **Boundaries** > **[interface boundary]** > **Physics Conditions** > **Tangential Velocity Specification** node for each side of the interface.

Pressure Drop Formulation

Flow through a porous medium is described by Darcy's law (Eqn. (364)), which relates the flow velocity to the pressure gradient based on a measure of permeability.

The single-phase flow pressure drop, Δp, is specified as follows:

$$\Delta p = -\rho(\alpha|v_n| + \beta)v_n \tag{81}$$

where:

- v_n is the superficial velocity normal to the surface.
- α is the user-specified constant **Porous Inertial Resistance**.
- β is the user-specified constant **Porous Viscous Resistance**.
- ρ is the fluid density at the interface.

The multiphase flow pressure drop, Δp, is specified as follows:

$$\Delta p = -\sum \gamma_i \rho_i (\alpha_i |v_i| + \beta_i) v_{n,i} \tag{82}$$

The summation index i runs over all of the phases. The multiphase resistances α_m and β_m are specified by the user. The phase resistances α_i and β_i are calculated as in the following two-phase example:

$$\rho_m = \gamma_d \rho_d + \gamma_c \rho_c \tag{83}$$

$$\alpha_d = \alpha_m \frac{\rho_m}{\rho_d} \quad \alpha_c = \alpha_m \frac{\rho_m}{\rho_c} \tag{84}$$

$$\beta_d = \beta_m \frac{\rho_m}{\rho_d} \quad \beta_c = \beta_m \frac{\rho_m}{\rho_c} \tag{85}$$

where:

- γ is the volume fraction
- v is the velocity
- the subscripts d and c stand for disperse and continuous, respectively.

Porous Baffle Properties

Use the properties that are listed below to adjust the specifications of a **Porous Baffle** interface node.

Boundary-0

The location of the boundary on the upstream side of the interface (Read Only)

Boundary-1

The location of the boundary on the downstream side of the interface (Read Only)

Type

Defines the type of interface. Must be set to `Porous Baffle Interface`.

Topology

Defines the connection type between the interfaces.

- **In-place**

- **Periodic**
- **Repeating**

Connectivity Option

Shows how the boundaries on two sides of the interface are connected. Only **Imprinted** connectivity is possible.

Porous Baffle Physics Conditions

Baffle Thermal Option

Available only when an energy model is activated in the physics continuum.

- **Non-Conducting**
- **Conducting**

See Baffle Interface.

Energy Source Option

Available only when the **Baffle Thermal Option** is set to **Conducting**.

- **None**
- **Heat Flux**
- **Heat Source**

See Baffle Interface.

Porous Baffle Treatment

- **Resistance Based**
  This baffle treatment is suitable for modeling open baffles where the flow is predominantly normal to the surface, such as that found in grills and screens.

  Both normal resistance and tangential resistance is applied to the baffle: you specify the **Porous Inertial Resistance**, **Porous Viscous Resistance**, **Porous Inertial Tangential Resistance**, and **Porous Viscous Tangential Resistance**.

  Resistance in the tangential direction is considered homogeneous. The resistance is applied to the multiphase flow, rather than applied individually for each phase, so a porous baffle is not suitable for filtration-type applications.

  If a turbulence model is activated in the physics continuum, you also specify the **Turbulence Intensity** and the **Turbulent Length Scale**.

  The flow velocity is the same on both sides of the porous baffle. You specify the flow velocity on the upstream side of the porous baffle (in the **Regions** > **[region]** > **Boundaries** > **[interface boundary]** > **Physics Conditions** > **Tangential Velocity Specification** node).

  See Asymmetric Values on Interface Boundaries.
- **Shear Stress Based**
  This baffle treatment, sometimes referred to as a wall-like or a wall-based porous baffle, is best suited to applications where the flow is mostly tangential to the baffle.

Normal resistance only is applied to the baffle: you specify the **Porous Inertial Resistance** and the **Porous Viscous Resistance**. Tangential drag is accounted for with a standard wall treatment.

You can specify different flow velocities on either side of the porous baffle (in the **Regions** > **[region]** > **Boundaries** > **[interface boundary]** > **Physics Conditions** > **Tangential Velocity Specification** node for each side of the interface).

Porous Baffle Physics Values

Heat Flux

Available only when the **Energy Source Option** is set to **Heat Flux**.

See Baffle Interface.

Heat Source

Available only when the **Energy Source Option** is set to **Heat Source**.

See Baffle Interface.

Intersection

Added for all direct interfaces to control the tolerance. The tolerance is a fraction of the smallest cell face on the secondary boundary that the secondary boundary vertices are allowed to move during an intersection. If the intersection fails, the following message is displayed in the output window - `Warning: no intersections found`. The solution cannot proceed under these conditions.

Porosity

Available for single-phase flows only.

The factor χ in Eqn. (80) that is used in calculating heat transfer and optionally viscous shear. The value ranges from 0 to 1. If the porous baffle is thought of as a perforated surface, then χ can be interpreted as the ratio of the area of the holes to the area of the surface.

Porous Inertial Resistance

For single-phase flows, the value of the coefficient α in Eqn. (81). For multiphase flows, the value of the coefficient α_m in the normal direction (see Eqn. (84)).

This value can be positive or 0. The default value is 0.

Porous Inertial Tangential Resistance

The value of the coefficient α_m in the tangential direction (see Eqn. (84)).

Available only when the **Porous Baffle Treatment** is set to **Resistance Based**.

Porous Viscous Resistance

For single-phase flows, the value of the coefficient β in Eqn. (81). For multiphase flows, the value of the coefficient β_m in the normal direction (see Eqn. (85)).

This value can be positive or 0. The default value is 0.

Porous Viscous Tangential Resistance

The value of the coefficient β_m in the tangential direction (see Eqn. (85)).

Available only when the **Porous Baffle Treatment** is set to **Resistance Based**.

Thermal Resistance

Available only when the **Baffle Thermal Option** is set to **Conducting**.

See Baffle Interface.

Turbulence Intensity

Available only when the **Porous Baffle Treatment** is set to **Resistance Based**, and a turbulence model is activated in the physics continuum.

Turbulent Length Scale

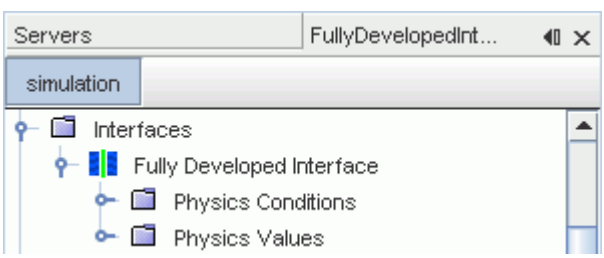
The length scale of the turbulence is related to the characteristic geometry scale of the baffle itself, for example, the perforation diameter or tube spacing.

Available only when the **Porous Baffle Treatment** is set to **Resistance Based**, and a turbulence model is activated in the physics continuum.

Fully Developed Interface

A fully developed interface is used to specify the periodic pressure drop for periodic (cyclic) repeats.

Although it makes most physical sense that this type is applied to periodic interfaces, it is not restricted to periodic interfaces. It has properties, conditions, values, and a pop-up menu.



Modeling

- For both the Coupled and Segregated Flow models, the pressure jump Δp is specified at the interface.
- For both the Coupled and Segregated Flow models, the mass flow rate $\dot{m}$ may be specified instead of the pressure jump.
- Each interface has a direction from side 0 to side 1. When you create the interface, select side 0 first, then side 1. For a fully developed interface, the upstream is side 0. Therefore if you select the sides in the wrong order, the sign of the pressure jump is reversed.
- This interface type may only be used between regions belonging to the same fluid continuum.

- If the mass flow rate is specified, there can be no inlets or outlets that are connected to the continuum containing the interface. In addition, specifying the mass flow rate typically does not make sense for an in-place interface.
- When debugging it is appropriate to report mass flow rates and pressures on the interface boundaries. This is more representative than creating derived parts nearby.

Fully Developed Interface Properties

You can use the properties that are listed below to adjust the specifications of a **Fully-developed Interface** node.

Boundary-0	The location of the boundary on one side of the interface (Read Only)	
Boundary-1	The location of the boundary on one side of the interface (Read Only)	
Type	Defines the type of interface. Must be set to `Fully-developed Interface`.	
Topology	Defines the connection type between the interfaces	
	In-place	Uses the in-place topology.
	Periodic	Uses the periodic topology.
	Repeating	Uses the repeating topology.
Connectivity Option	Shows how the boundaries on two sides of the interface are connected. Only **Imprinted** connectivity is possible.	

Fully Developed Interface Conditions

You can use the properties that are listed below to adjust the specifications of a **Physics Condition** node.

Fully Developed Flow Option	Specifies the flow when either the Coupled or Segregated Flow model is used.	
	Pressure Jump	Specifies pressure jump for the interface. A **Pressure Jump** interface value is added.
	Mass Flow Rate	Calculates the mass flow rate for the interface. When the Segregated Flow model is used, a **Mass Flow Rate** interface value is added. When the Coupled Flow model is used, this value and a **Mass Flow Rate Controls** interface value are added.
Fully Developed Energy Option	Specifies the energy option at the interface.	
	Periodic	No temperature or enthalpy discontinuity at the interface.

	Constant Temperature Walls	This option corresponds to the situation where the walls of the periodic domains are held at constant temperatures. An interface condition **Inflow Temperature Specification** is added. The interface values **Bulk Inflow Temperature** and **Temperature Reference** are added. If the wall temperature does vary, the scaling of the temperature profile is not correct, and an error is incurred.
	Constant Heat Flux Walls	This option corresponds to the situation where a constant heat flux is introduced at the boundaries of the periodic domain. An interface condition **Inflow Temperature Specification** is added. A **Bulk Inflow Temperature** interface value is added. If the heat flux does vary, the scaling of the temperature profile might be affected, introducing some additional error. Periodically varying heat fluxes might be acceptable, since the resulting temperature field is linear, allowing the principle of superposition to be applied.
Inflow Temperature Specification	Specifies the temperature option at the interface. Activated when the **Constant Temperature Walls**or **Constant Heat Flux Walls**is selected for **Fully Developed Energy Option**.	
	Bulk Mean Inflow Temperature	Specifies the bulk mean temperature at the inflow boundary. A **Bulk Inflow Temperature** interface value is added.
	Minimum Inflow Temperature	Specifies the lower limit of the temperature at the inflow boundary. A **Minimum Inflow**

		Temperature interface value is added.
	Maximum Inflow Temperature	Specifies the upper limit of the temperature at the inflow boundary. A **Maximum Inflow Temperature** interface value is added.

Fully Developed Interface Values

You can use the properties that are listed below to adjust the specifications of a **Physics Values** node.

Bulk Inflow Temperature	Specifies the bulk mean temperature at the inflow boundary.	
Temperature Reference	Activates a standard *Select Objects* dialog for selecting boundaries.	
Mass Flow Rate	The mass flow rate for the fully developed interface.	
Maximum Inflow Temperature	Specifies the upper limit of the temperature at the inflow boundary. For the **Constant Temperature Walls** condition, this value should be larger than the reference temperature.	
Mass Flow Rate Controls	Options for the *Mass Flow Rate*.	
	Computed Pressure Jump	The pressure jump across the interface (read-only).
	Verbose	Expert Property. Additional output is printed while the simulation is running when this option is activated.
	Update Frequency	Expert Property. Sets the number of iterations or time-steps that must occur before the value is updated.
	Under-Relaxation Factor	Expert Property. At each iteration, this property governs the extent to which the old solution is supplanted by the newly computed solution.
Minimum Inflow Temperature	Specifies the lower limit of the temperature at the inflow boundary. For the **Constant Temperature Walls** condition, this value should be smaller than the reference temperature.	
Pressure Jump	This interface value is only applicable when **Pressure Jump** is selected as the **Fully Developed Flow Option**.	
Intersection	Added for all direct interfaces to control the tolerance. The tolerance is a fraction of the smallest cell face on the secondary boundary that the secondary boundary vertices are allowed to move during an intersection. If the intersection fails, the following message is displayed	

	in the output window - `Warning: no intersections found.` The solution cannot proceed under these conditions.

See Also:

Orientation of Parts-Based Interfaces
Modeling Fully Developed Flow and Energy

Orientation of Parts-Based Interfaces

In cases where the correct orientation of an interface requires confirmation, STAR-CCM+ adds an **Orientation** node to parts-based interfaces. Use the right-click menu on this node to access the action for reversing the orientation. This confirmation is particularly required for interface types such as fan, blower, and fully-developed.

- The Orientation has two modes: Automatic and Manual.
  - Automatic:
    - Set direction for contacts based on regions associated with the part-surface pairs
    - Set direction for contacts based on parent boundary associated with the part-surface pairs
    - Set direction for contacts based on parts associated with the part-surface pairs
    - If mode is set to automatic, you can use the **Orient** command on the context menu to re-execute the ordering based on these rules.
  - Manual:
    - User can select contacts and reverse their direction
    - If the mode is set to Manual, you can use the Reverse Contacts... command on the context menu of the Orientation object to reverse selected contacts.
      - Invoking the command will pop-up the Reverse Contacts contact selector.
      - You can pick contacts to Reverse
      - The icon changes (small blue arrow) to indicate the "direction" of the orientation.
- Certain interfaces may have conditions that require consistent orientation, e.g. fan.

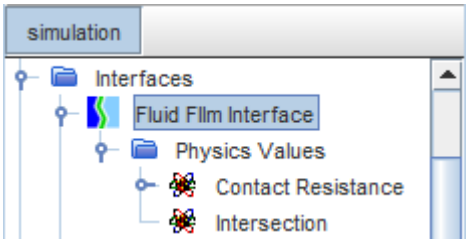
See Also:

Creating a Blower Interface
Creating a Fan Interface
Fully Developed Interface

Fluid Film Interface

The fluid film interface type is used to join a fluid and a fluid-film region.

It represents the contact surface between two fluids. It has properties, conditions, values, and a pop-up menu.



Modeling

- The equation for mass conservation is used to calculate the film thickness. The volume and surface area of the film are functions of the thickness. Contributions to mass source or sink come from droplet impingement, film stripping, mass transfer, or user-defined sources.
- The equation for momentum conservation is based on kinematic and dynamic conditions at the interface between the film and the surrounding fluid (film free surface).
- The equation for energy conservation is based on contributions to the energy source or sink from droplet impingement, film stripping, or user-defined sources.

Fluid Film Interface Properties

You can use the properties that are listed below to adjust the specifications of a **FluidFilm Interface** node.

Boundary-0	The location of the boundary on one side of the interface (Read Only)	
Boundary-1	The location of the boundary on one side of the interface (Read Only)	
Type	Defines the type of interface. Must be set to `Fluid-Film Interface`	
Topology	Defines the connection type between the interfaces	
	In-place	Uses the in-place topology.
	Periodic	Uses the periodic topology.
	Repeating	Uses the repeating topology.
Connectivity Option	Shows how the boundaries on two sides of the interface are connected. Only **Imprinted** connectivity is possible.	

Fluid Film Interface Values

You can use the properties that are listed below to adjust the specifications of a **Physics Values** interface node.

Contact Resistance	The value of the resistance to conduction through the interface. It is entered as a scalar profile. If a field function defines the value and the Ignore Boundary Values property on the field function is ticked, the function is computed using data from the cell next to the boundary.	
	Dimensions	Specify the resistance in units of $m^2 \cdot K/W$ or similar. (Read Only)

	Method	Specify the input method type from **Constant**, **Field Function**, **Table**, or **User Code**.
Intersection	Added for all direct interfaces to control the tolerance. The tolerance is a fraction of the smallest cell face on the secondary boundary that the secondary boundary vertices are allowed to move during an intersection. If the intersection fails, the following message is displayed in the output window - `Warning: no intersections found`. The solution cannot proceed under these conditions.	

Interface Pop Up-Menu

The pop-up menu items are available when you right-click an interface node.

Initialize	Populates the interface boundary nodes with faces from the original boundaries.
Reset	Reverses the process of initialization. Any vertices that the process removes are reported in the output window for each region.
Insert Shell Region Interface	Replaces the interface with a shell region and creates two new contact interfaces that connect the shell region to the parent boundaries on either side. The new shell region can be associated with a shell continuum to model both in-place and out-of-plane heat conduction. Only available for in-place interfaces. See Inserting a Shell Region at an In-Place Interface.
Swap Regions	For an overset mesh interface, this action swaps the association of the overset and background regions. See Overset Mesh Interface Properties.
Swap Boundaries	Exchanges the Boundary-0 value for Boundary-1. Only available for direct interfaces.

Deleting Interfaces

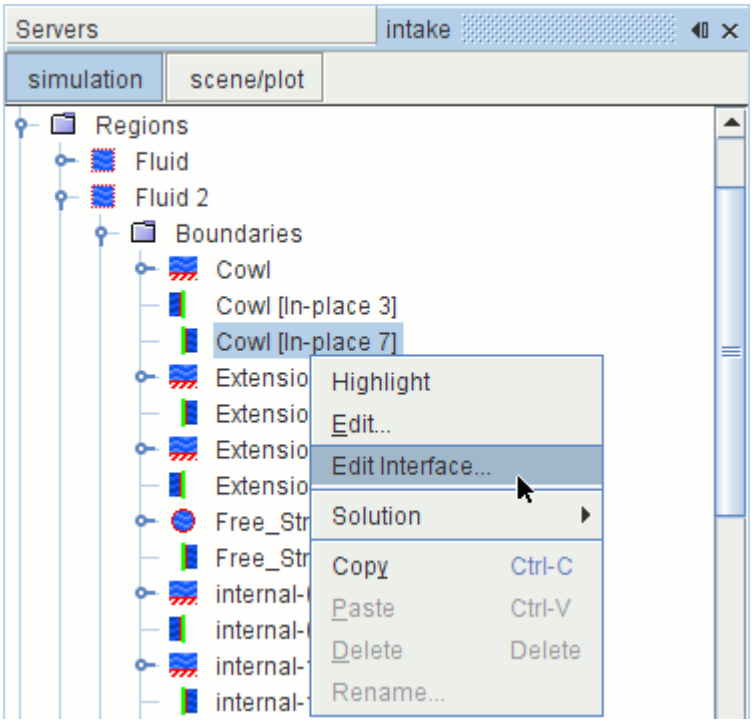
Deleting an Interface restores the boundaries to their state prior to the interface being generated.

When an interface is deleted, any new vertices or faces generated as part of the intersection process are removed. If you want to maintain the interface but delete the interface intersections then reset the interface instead. There is no undo operation, so save the simulation file performing this step if you are unsure of whether you need to delete the interface.

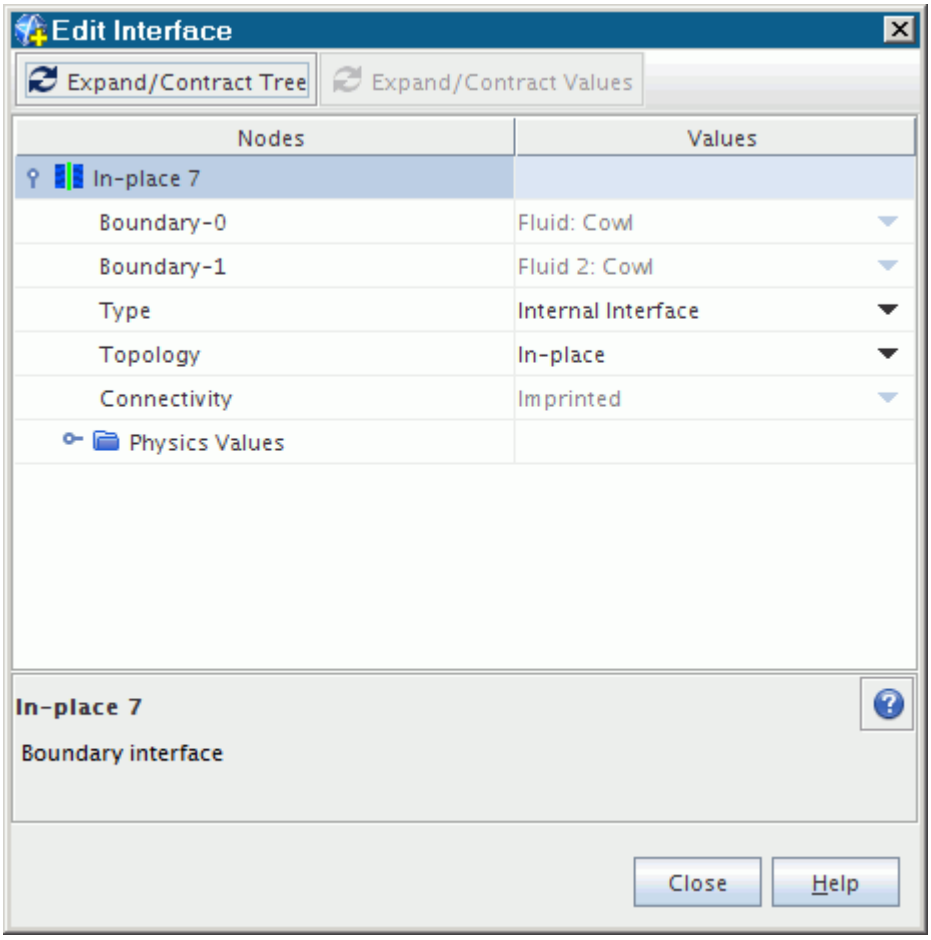
Accessing an Interface from an Interface Boundary

In some simulations, such as with conjugate heat transfer, you might have a large number of boundaries and interfaces. STAR-CCM+ lets you access the properties of an interface directly from any of its interface boundaries, rather than having to scroll through the object tree.

Right-click the node of the interface boundary and select **Edit Interface** from the pop-up menu.



The table editor for the parent interface appears.



Converting Boundaries to Interfaces

Converting boundaries to interfaces can solve two problems: insufficient demarcation of regions and surfaces that lack required thickness.

During mesh generation in STAR-CCM+, the original imported geometry may be lacking sufficient boundaries for the regions to be properly demarcated. In particular, it is a frequent occurrence that a shared boundary is used for inter-region surfaces whereas the requirements for multi-region meshing impose that two separate boundaries must exist in order for the regions to be properly defined.

Another situation is the existence of baffle surfaces within the geometry that are treated as infinitely thin surfaces that impede the flow and/or transfer heat along their length. These are typically imported as interior boundaries but is normally flagged as being non-manifold if used as part of the meshing process with STAR-CCM+. To meet the topology requirements of the meshing tools they have to be duplicated so that in a topological sense they have thickness.

Converting a boundary to an interface allows the requirements for the above situations to be met.

When a boundary is converted to an interface, two separate actions occur:

1. a copy of the original boundary or boundaries (with the same surface and perimeter triangulation) is made; and

2. an in-place interface is automatically created linking the original boundary or boundaries with the copy or copies

The copy of the original boundary can then be used as part of the definition of a new region, for example, by using the create new region from boundaries option. This process also automatically moves the interface node copy into the appropriate region. In the case of baffle surfaces, the perimeters of the two boundaries are merged so as to form an infinitely thin closed volume. If the original baffle boundary was next to an exterior surface then it is important that the baffle boundary be merged with the exterior surface converting it to an interface. The surface repair tools can be used for this purpose.

Note that converting a boundary to an interface is different from simply creating an interface between two preexisting boundaries, which corresponds with the second step described above.

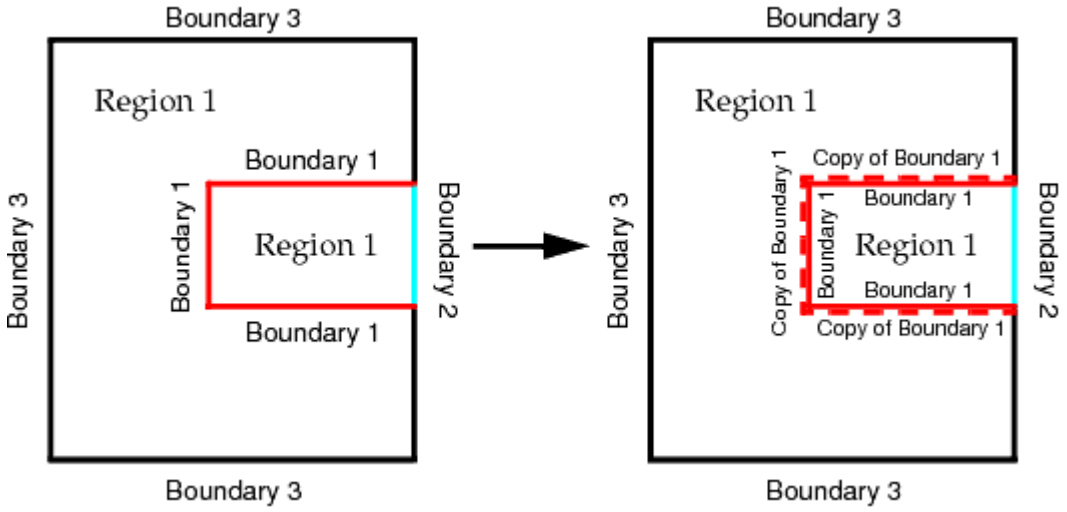
Contents:

Converting to an Interface

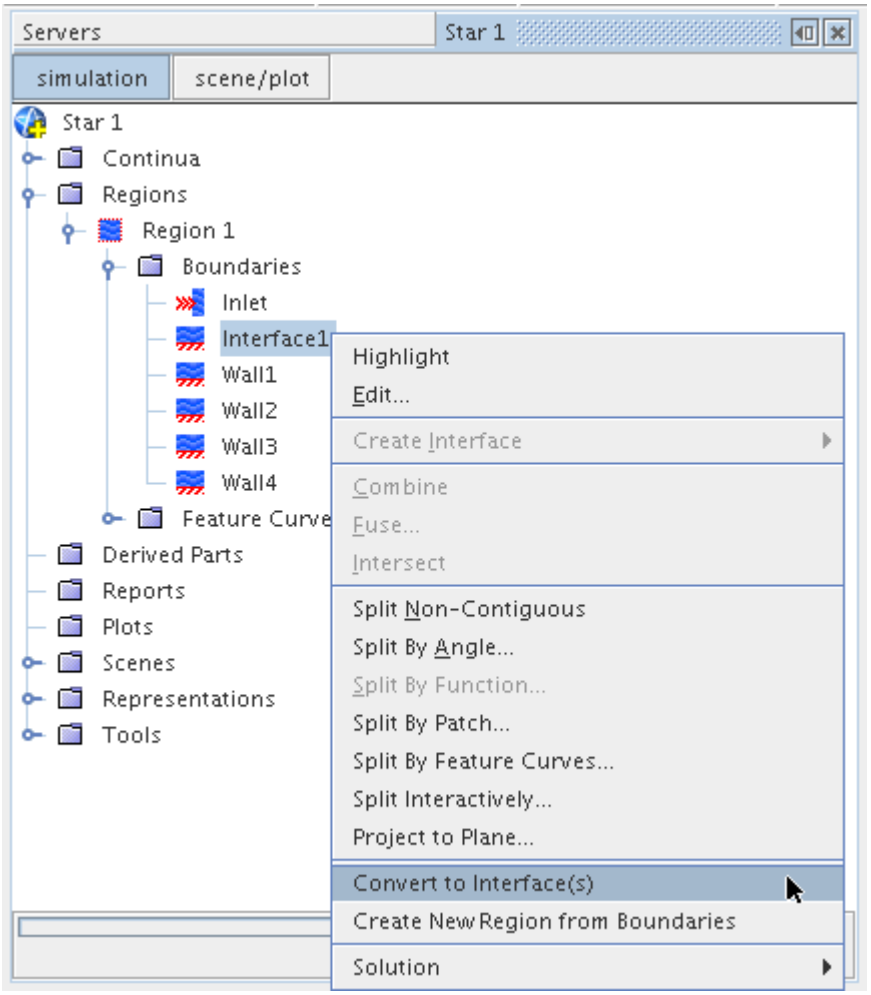
Converting to an Interface

Before proceeding with converting a boundary to an interface, it is recommended that you combine any boundaries that belong to the same interregion surface into one. This simplifies the setup process thereafter by reducing the number of interfaces created.

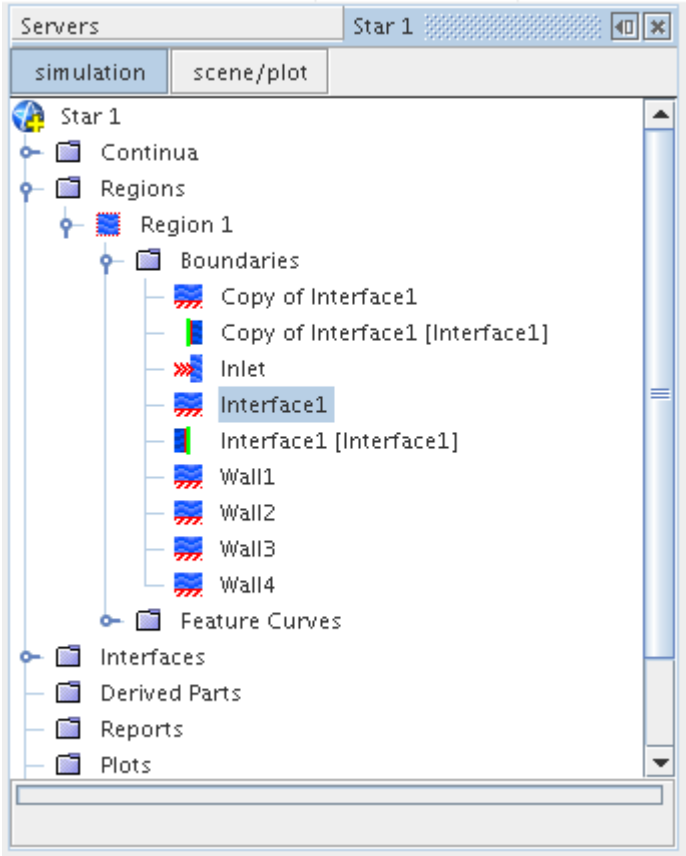
An illustration showing the before and after boundary setup when a boundary has been converted to an interface is given below:



To convert to an interface, select the boundary or boundaries to be converted and then right-click. In the popup menu that appears, select **Convert to Interface(s)**.

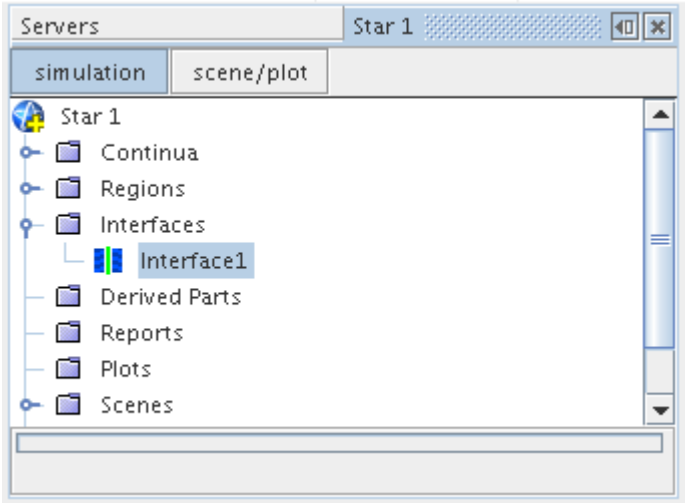


This begins the interface conversion process. Once this is complete, the simulation tree is updated with the new boundary and interface information.

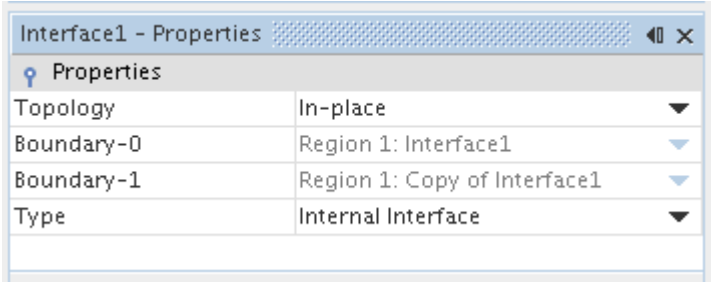


As can be seen, a copy of the original **Surface 1** boundary has been made, in addition to a pair of interface nodes linking the two boundaries. The boundary copy can now be used as part of a new region definition or simply left in the region to act as a baffle.

A new interface node with the same name as the original boundary is shown in the**Interface** node in the simulation tree. Selecting the interface node highlights the interface in the graphics window.



Examine the *Properties* window for the new interface node, for the default *Topology* and *Type* settings:



For normal meshing purposes, do not change these settings. For baffle surfaces change the *Type* property to **Baffle** or **Porous Baffle**, depending on the required flow properties.

Examining Interfaces

Interface creation normally proceeds without any problems if the underlying mesh is valid and the interfaces have been defined properly.

The most common problems associated with interfaces are:

- cells with negative volume; and
- unsuccessful intersection of interfaces

Cells with Negative Volume

One cell with a negative volume can keep the simulation from iterating. While negative-volume cells can have a variety of causes, an interface might be the cause if it was created from two incorrect or misplaced boundaries.

To detect this before running the simulation, obtain a report of the negative cell volumes after creating the interface, even if you generated such a report earlier.

Unsuccessful Intersection of Interfaces

If you attempt to solve a simulation with few or no intersections in an interface, errors are likely. Therefore, when setting up simulations with interfaces, check the interface intersection as early as possible for problems such as:

- Too few or zero intersections in an interface caused by specifying the axis of rotational periodics incorrectly.
- A boundary may unexpectedly be left without any faces after the intersection process; therefore check the boundary icons.
- The absence of any intersections in an interface usually results from the chosen boundaries not being geometrically beside each other. For details on detecting and solving , read about examining the intersection report.

To test for gaps or holes in the mesh, display the interface boundaries.

The various methods of checking the successful intersection of an interface include:

- checking the boundary icons

- examining the intersection report
- displaying the interface boundaries
- displaying remainder boundaries.

Examining an interface is important as it determines the integrity of the connection between or across the mesh from one boundary to another. Incomplete interfaces result in a loss of continuum information being transferred, resulting in a less accurate solution.

The typical causes of the intersection process failing are:

- selecting the wrong boundary pairs
- forgetting to set the rotational axis direction or setting the wrong direction for axisymmetric cases
- incorrect tolerance setting
- the boundaries are not coincident in space nor aligned correctly.

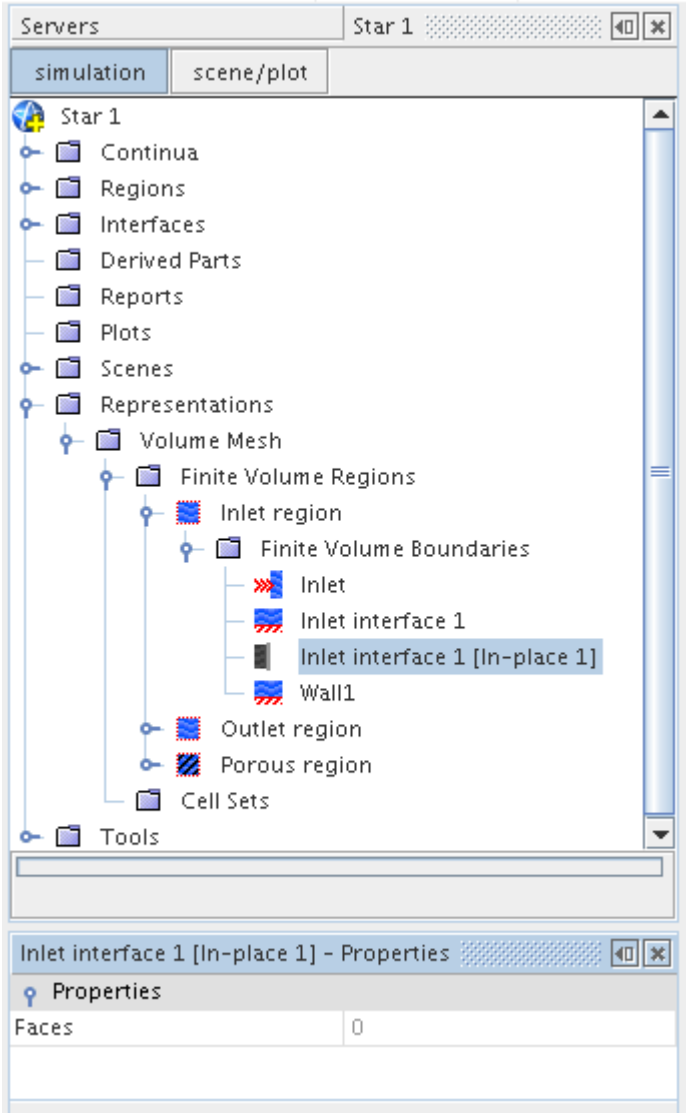
Contents:

Checking the Boundary Icons
Examining the Intersection Report
Displaying the Interface Boundaries
Displaying Remainder Boundaries

Checking the Boundary Icons

After a successful intersection process, the interface boundary icons under the **Representation** node become active.

An active interface boundary icon has an appropriate color applied to it. If an interface boundary contains zero faces after intersection then the icon under the **Representations** node remains grayed out, indicating that the intersection process was unsuccessful. The number of *Faces* in the *Properties* window also remains at zero, as shown below:



Additionally, check the intersection report for error messages and attempt to display the interface boundaries to further confirm that the interface intersection was not done.

Note: The boundary and interface icons under the original **Regions** node are not grayed out regardless of whether they contain faces or not. The intersection process affects the **Representations** node icons.

Examining the Intersection Report

The validity of the interface is only checked during the intersection process and not during the original selection of the boundaries making up the interface.

During the validity check, if no intersections can be found for the two interface boundaries then a warning message is displayed in the output window during the initialization process:

```
Output - Star 1

Updating interface In-place 1 between Inlet region:Inlet interface 1 and Porous region:Porous Interface1 tolerance 0.02
  Created 0 intersection faces

Warning: no intersections found.

Updating interface In-place 2 between Outlet region:Outlet interface 2 and Porous region:Porous Interface2 tolerance 0.02
  Created 0 intersection faces

Warning: no intersections found.
```

The likely cause in this instance is that you either selected the wrong boundary pairs or did not set the axis of rotation for an axisymmetric case. It's possible that the boundaries are not coincident in space within the specified tolerance. Changing the verbosity of the interface output may provide additional clues as to what is going on.

It is recommended that you delete or reset the interface and attempt to recreate them once you have diagnosed the original cause of the problem.

Displaying the Interface Boundaries

The quickest and easiest way to check if an interface was intersected is to display the interface faces as part of a scene.

There are three ways to show an interface in an existing scene:

- highlight the interface by selecting the appropriate boundary node in the simulation tree;
- right-click on the interface node and select highlight; and
- isolate the interface parts within the scene

If after selecting the node, no highlighted faces are seen then it means that the interface creation process was not successful. If highlighted faces appear, you can inspected them for any unexpected holes or gaps. The color of the interface can also be used to judge if the correct interface type has been selected. The following list indicates the color used by each type:

Interface Type	Color
Baffle	Brown
Contact	Green
Fan	Light brown
Fully developed	Purple
Internal	Yellow
Mixing plane	Blue
Blower	Light Blue
Porous-baffle	Orange

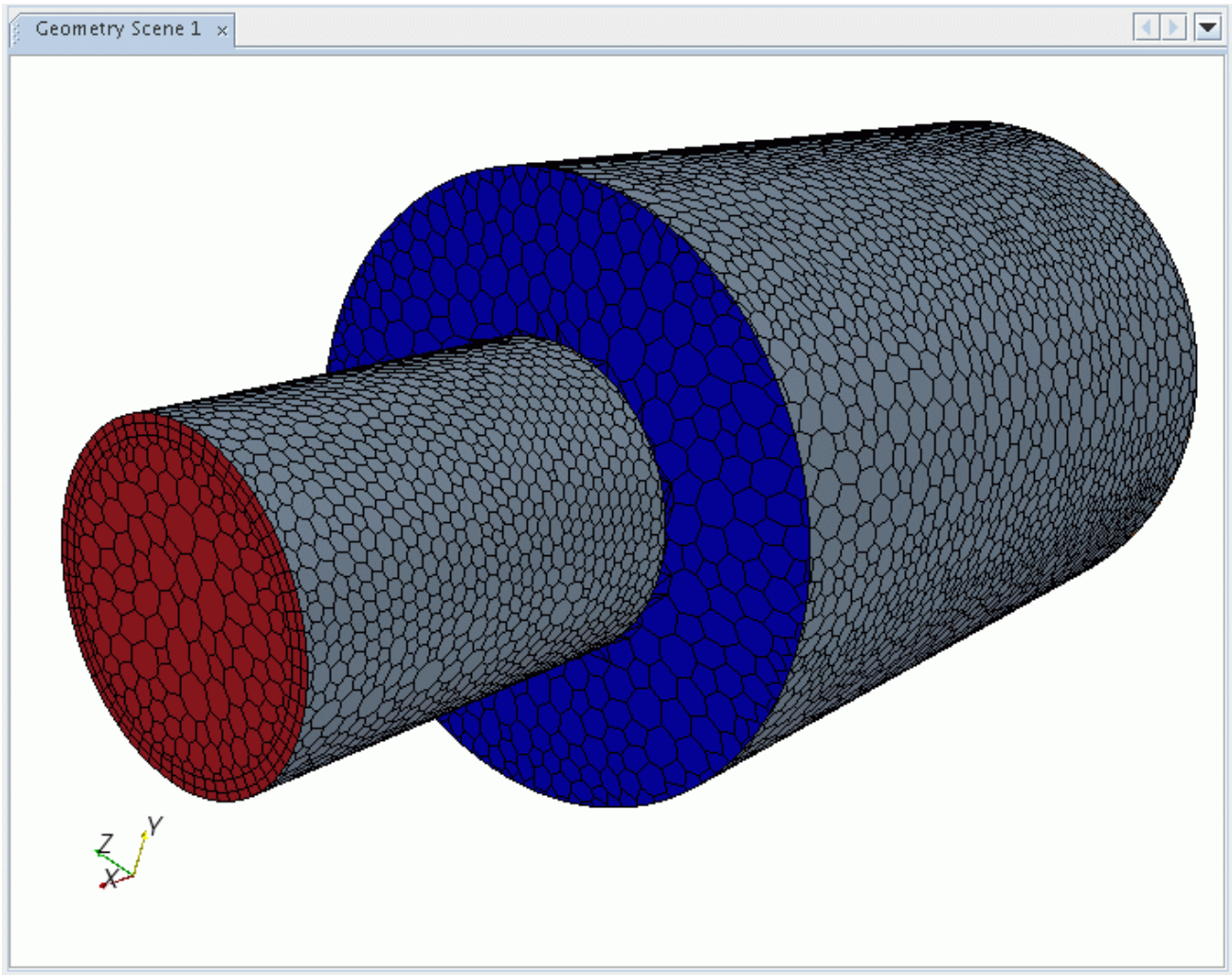
Displaying Remainder Boundaries

If the original boundaries from which the interface is created do not completely match, the intersection process leaves some boundary faces in the original boundaries. These are called remainder boundaries.

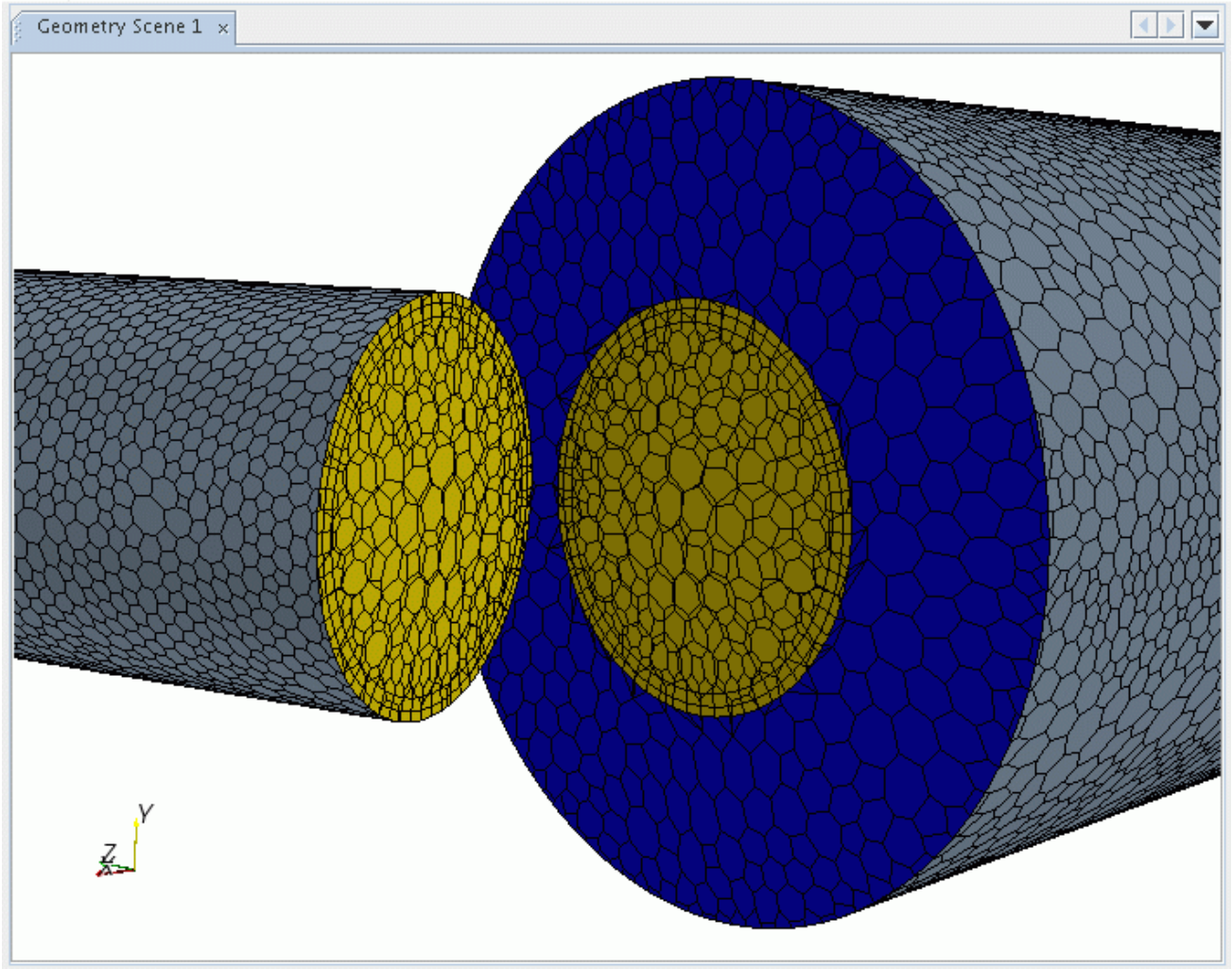
There are two instances when remainder boundaries occur:

- The two original boundaries have different perimeters; and
- The two original boundaries have the same perimeters, but are discretized to different resolutions.

In cases where the boundaries have different perimeters (typically when one has a larger area than the other) then a portion of original boundaries remain after the intersection process has taken place. An example would be when one cylinder of a smaller radius is joined to another cylinder with a larger radius:

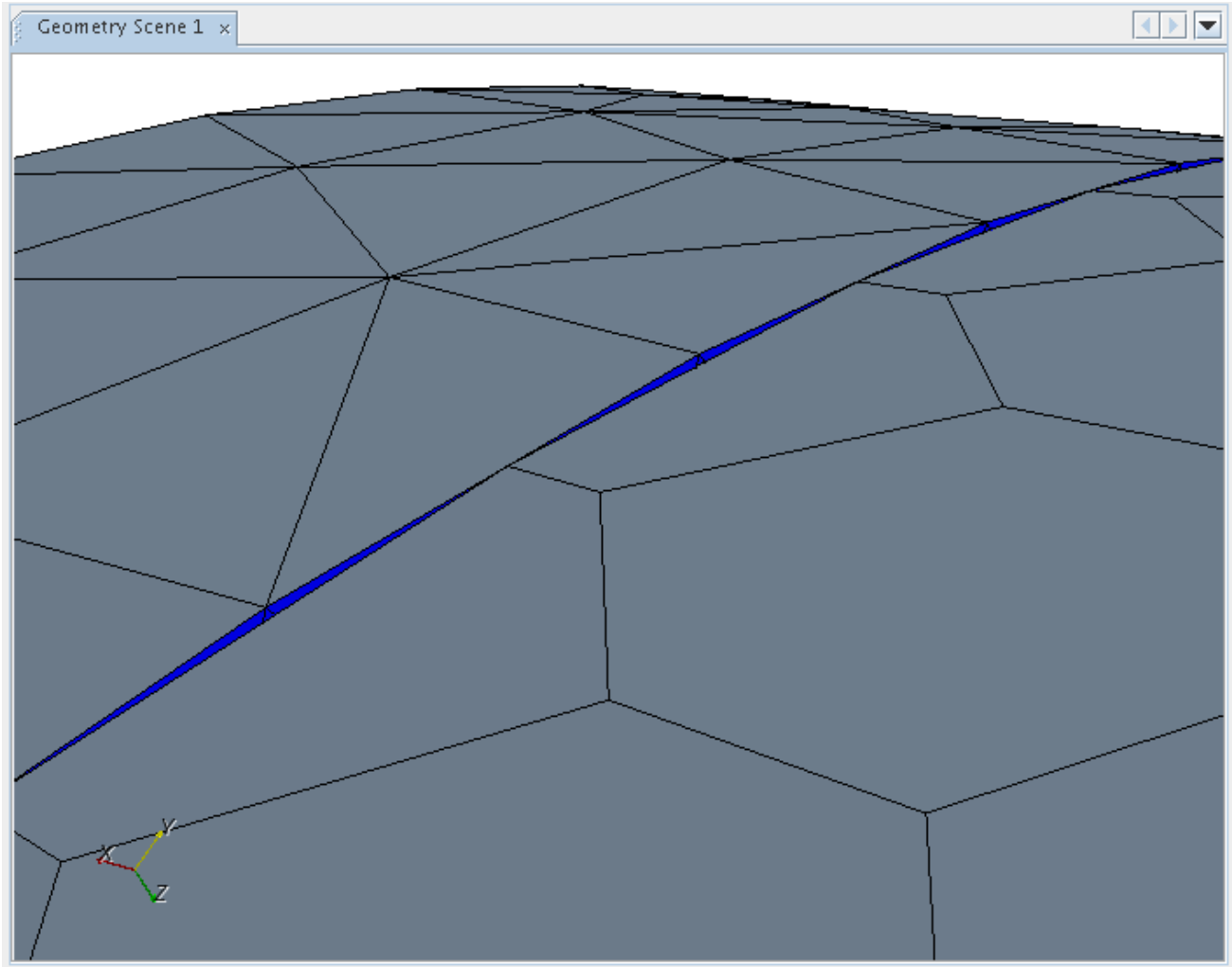


In this instance, the boundaries share the same common area in the middle but the larger cylinder has remainder boundaries, as highlighted in blue. This is further demonstrated below in an exploded view of the two regions:



The remainder boundary faces remain a part of the original boundary node. It is important that the appropriate boundary type be set for them to ensure the correct modeling characteristics.

Where the boundary perimeters are identical but have different mesh resolutions (as would happen when joining a tetrahedral type mesh to a polyhedral type one for example), then small portions of the original boundaries remain, as highlighted by the blue faces below:



In this instance, specifying the correct boundary type for the remainder boundaries is less important as their impact is usually marginal.

To visualize the boundary remainder faces, follow the same procedure as used for displaying interface boundaries, but instead of displaying the interface boundaries, select one or both of the original boundaries.

See Also:

Extrusion Parameters Properties

Inserting a Shell Region Inside an Interface

The Insert Shell Region Interface action allows you to select certain types of interface and insert a shell region between the two sides of the interface. Using this action is much quicker and easier than creating the shell region and interfaces manually.

This action is available for internal interfaces, contact interfaces, baffle interfaces, and porous-baffle interfaces. Note that shell regions are not compatible with part-based interfaces.

When you use a solid baffle to model a thin solid plate, the flow goes around the baffle. Heat can conduct through the baffle, experiencing some heat resistance, but heat cannot conduct along the baffle. If you expect a large temperature gradient along the baffle, then neglecting heat conduction along the baffle can result in a significant modeling error. The baffle can also have a large heat capacity, which is important in some unsteady cases. In order to account for conduction across the width and height of a baffle, you can replace the baffle interface with a solid shell region.

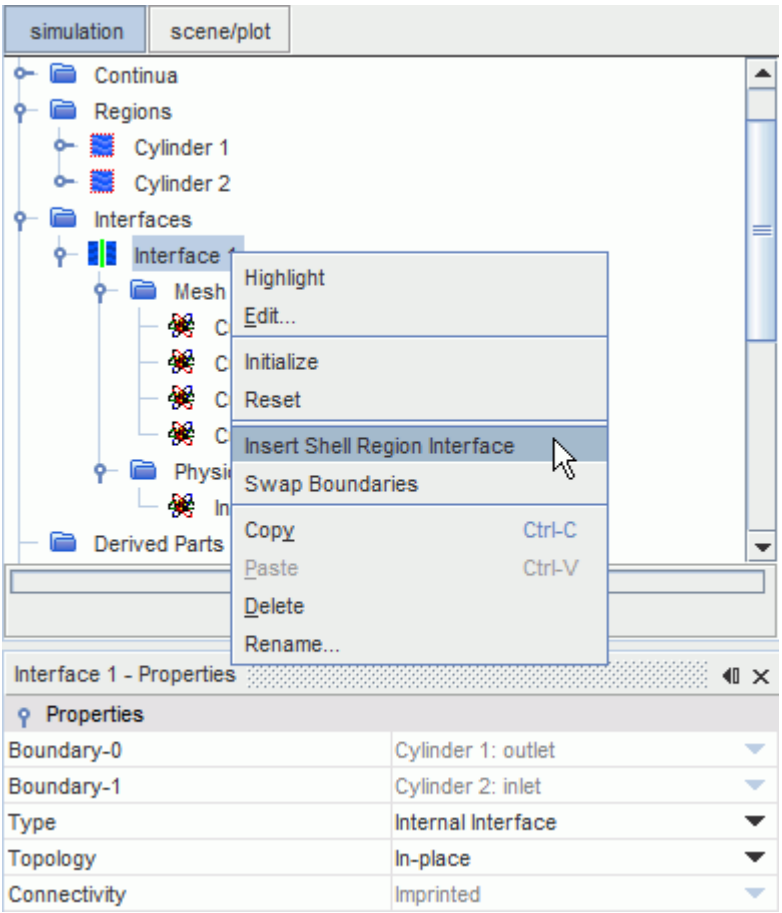
When you use this action to create a solid shell, the baffle interface is replaced with two new contact interfaces that connect the shell region to the solution domain on either side of the original baffle. A baffle interface models heat conduction through the thickness of the baffle only; a shell region models heat conduction in all directions.

A solid shell region is associated with a shell continuum. The models that are activated for that continuum define the behavior of the solid shell. The same physics continuum defines the material models for the shell.

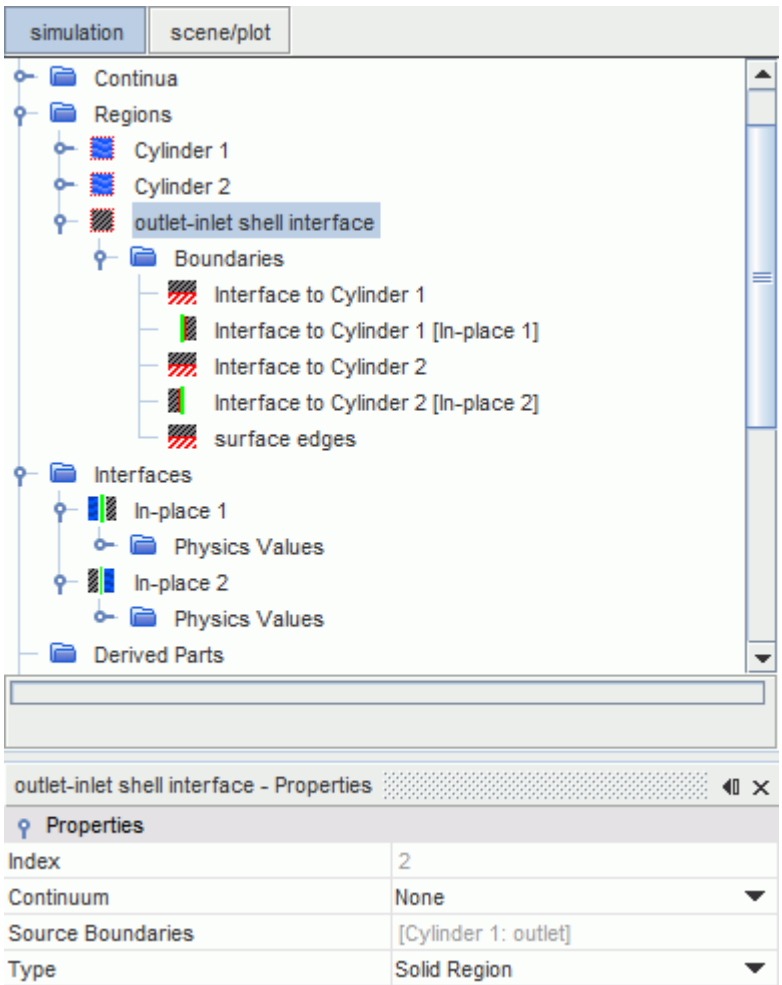
You can also use this action to create a fluid film at a Contact interface, the shell region is inserted between the fluid and solid regions. The interfaces between the fluid film shell and the fluid and solid regions are created and rearranged automatically.

To insert a shell region at a relevant interface:

1. Right-click the interface and choose **Insert Shell Region Interface**.



The shell region is created and the interfaces are rearranged automatically.



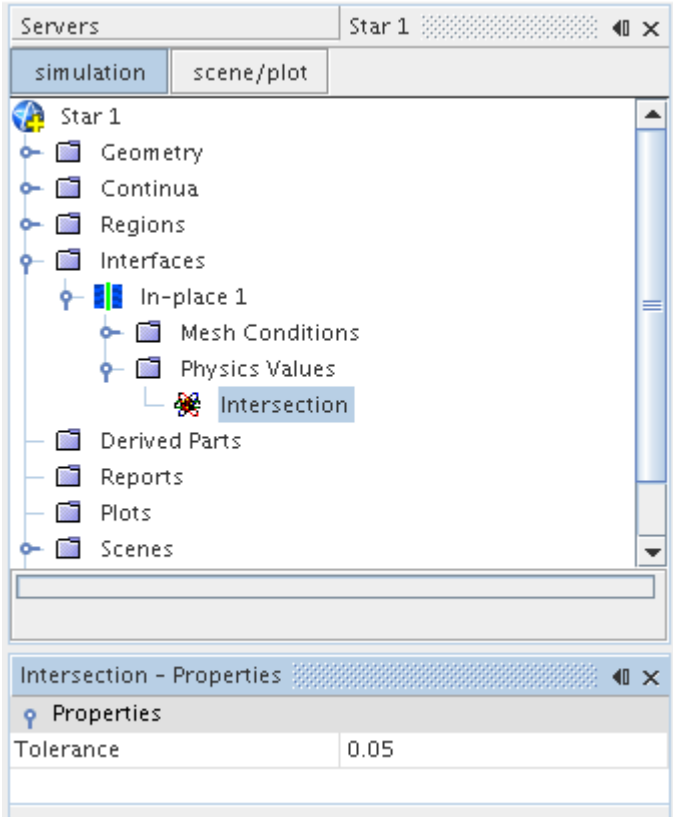
Adjusting the Interface Intersection Tolerance

The intersection tolerance limits the distance vertices on the slave boundary can be moved to match vertices, edges and faces on the master boundary for all direct type interfaces.

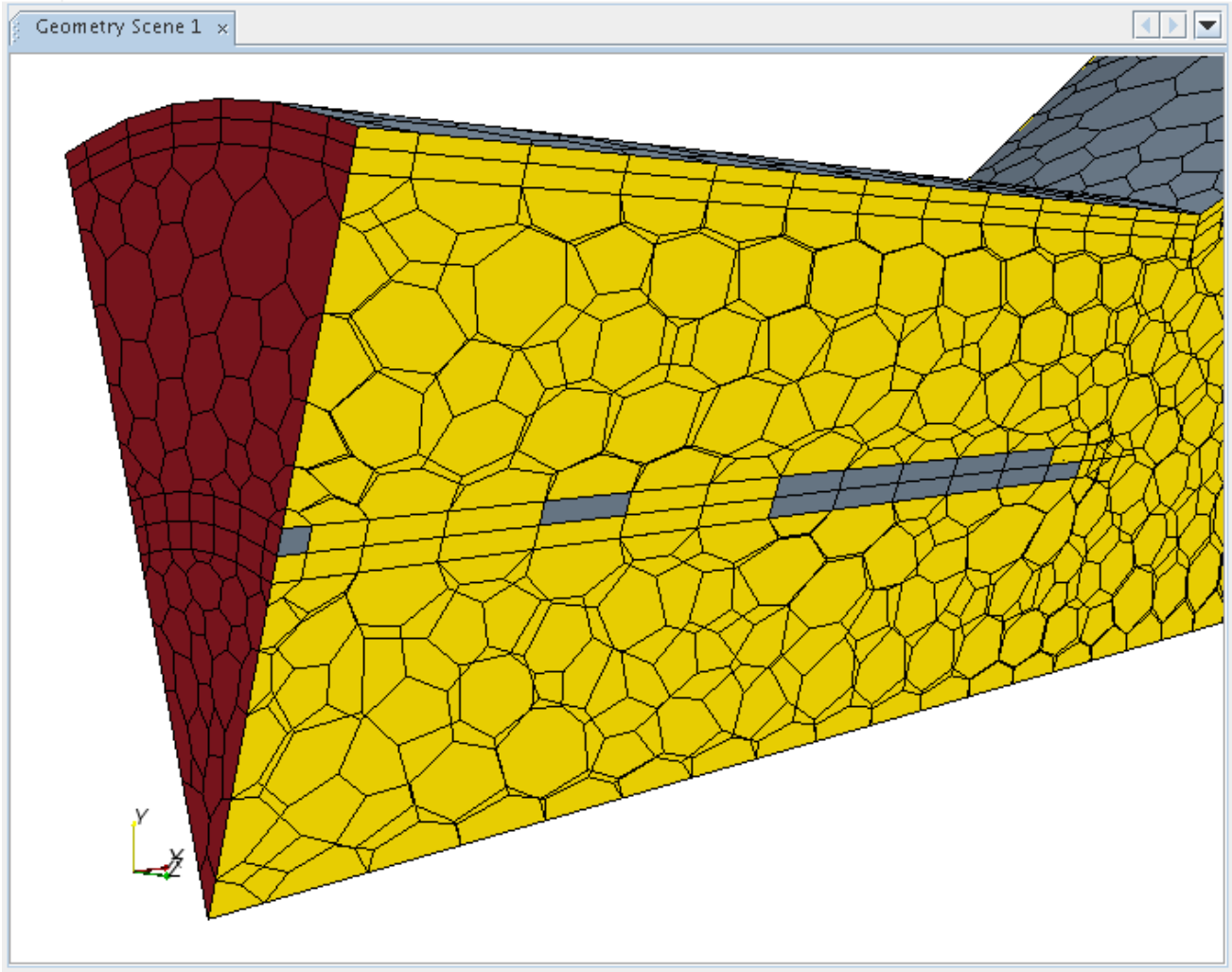
The tolerance value is specified as a fraction (from 0 to 1) of the length of the smallest edge attached to the vertex. The default value is 0.05 (which is 5% of the smallest attached edge length).

For each vertex on the slave boundary, the tolerance value is used to define the radius of a cylindrical volume within which the vertex moves during intersection. This cylindrical volume is centered on a projection vector that has the vertex as its origin, and whose direction is computed from the normals of adjacent faces. The length of the cylinder is ten times its radius.

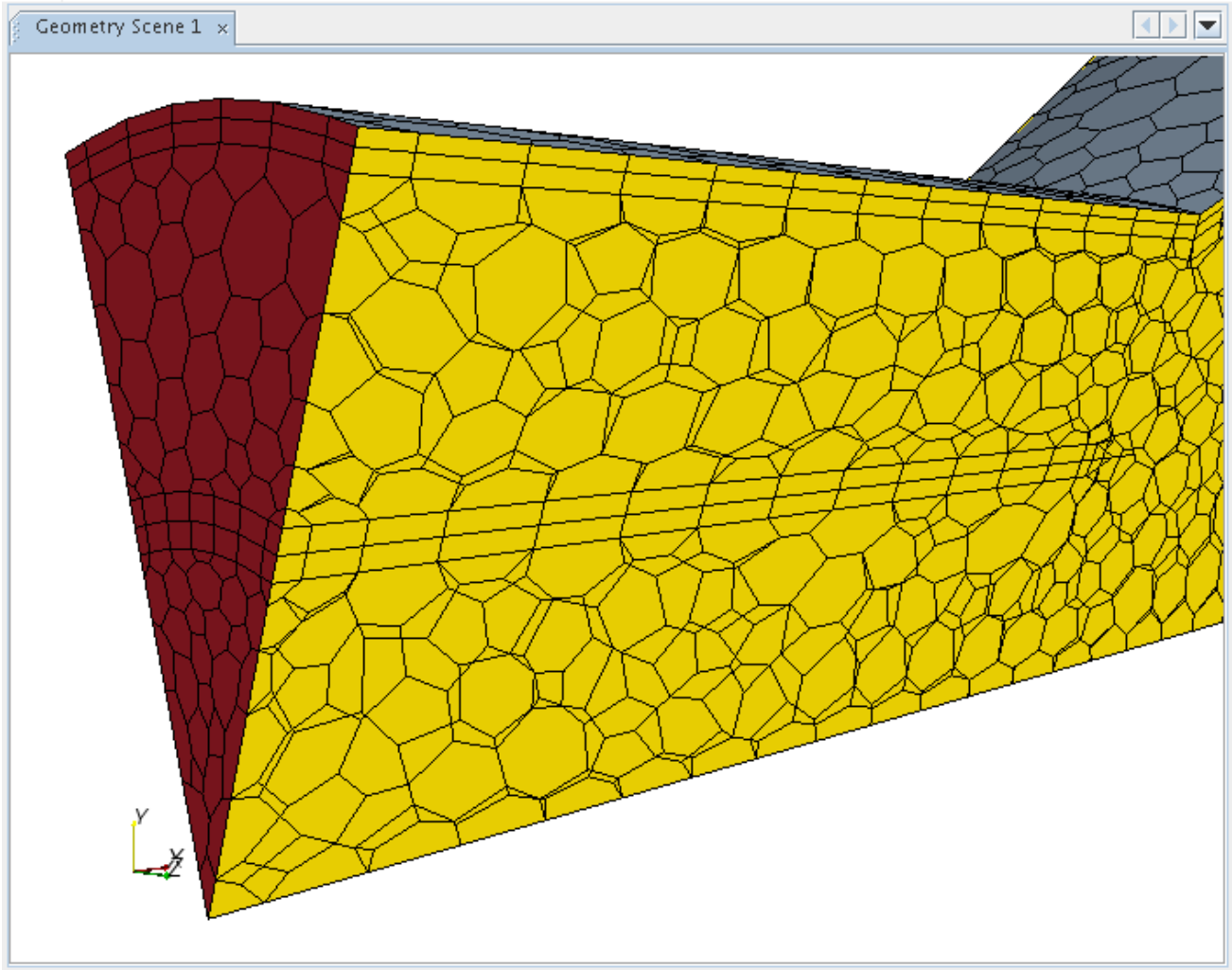
The *Tolerance* value can be accessed in the *Properties* window for the **Physical Values** > **Intersection** node for the interface, available once a physics continuum has been created:



The example below shows a case where the interface creation was not 100% successful, as it left unmatched faces (shown in gray) in the interface:



In this case increasing the tolerance value corrected the fault and allowed all of the faces to be matched:



Before changing the tolerance value, reset the interface first. You can then attempt the interface intersection again.

Intersecting an Interface

The intersection process occurs the first time the solution is initialized for the volume mesh.

This process involves the appropriate mesh calculations required to define the interface fully. For large problems with complex physics, it is generally advantageous (especially with regard to memory usage) to check the interface intersection after activating the Space model but before activating other models.

Intersection populates the interface boundary nodes with faces from the original boundaries. The faces belonging to the original boundary are removed. For a perfect intersection, where the matching areas of the original boundaries are the same, then the boundaries are left with zero faces. The boundary nodes however are not removed as the intersection process is reversible.

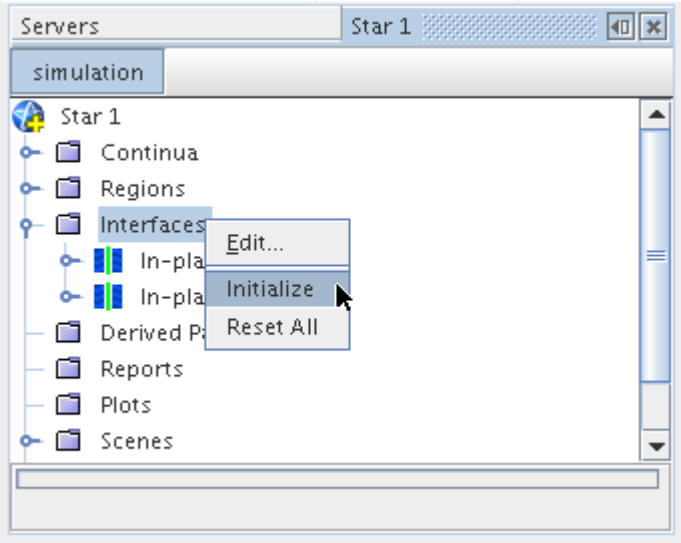
If the intersection process is not perfect or if the overlapping areas of the original boundaries are different, then there are some faces remaining in the original boundary nodes. Depending on the situation, examine these left over faces. If they are acceptable, then assign the appropriate boundary types and conditions to

them. If they are not acceptable then the intersecting tolerance can be changed and the interface reset before trying the intersecting process again.

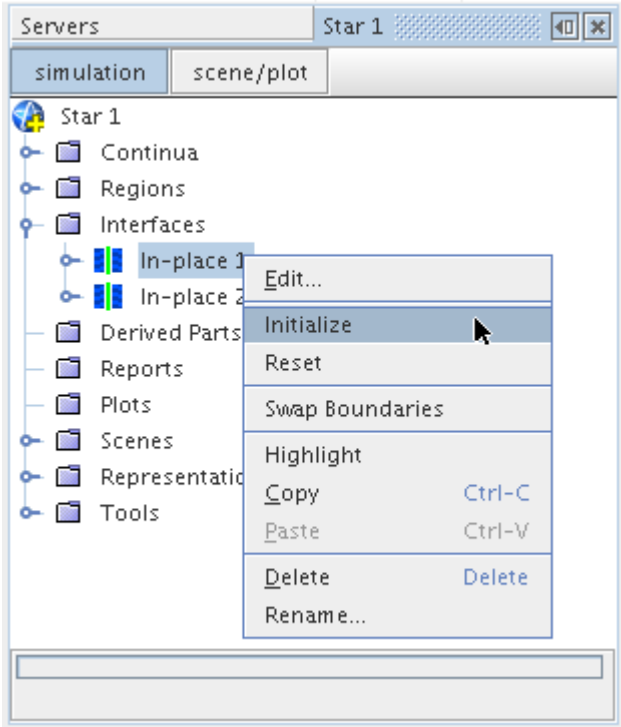
Once the interfaces have been created or reset, the intersection process can be performed in one of two ways:

- by manually initializing the solution
- by using the **Interfaces** manager node

Right-clicking on the **Interfaces** node give access to the **Initialize** option for all interfaces at the same time:

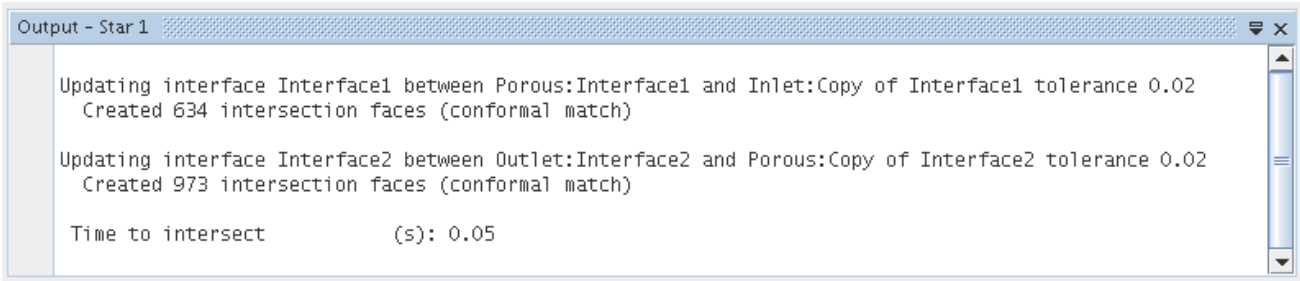


Alternatively, each interface can be initialized separately by selecting the interface node, right-clicking and selecting **Initialize**. For example:



In the case of repeating interfaces used for sliding mesh calculations, the reintersection is done dynamically.

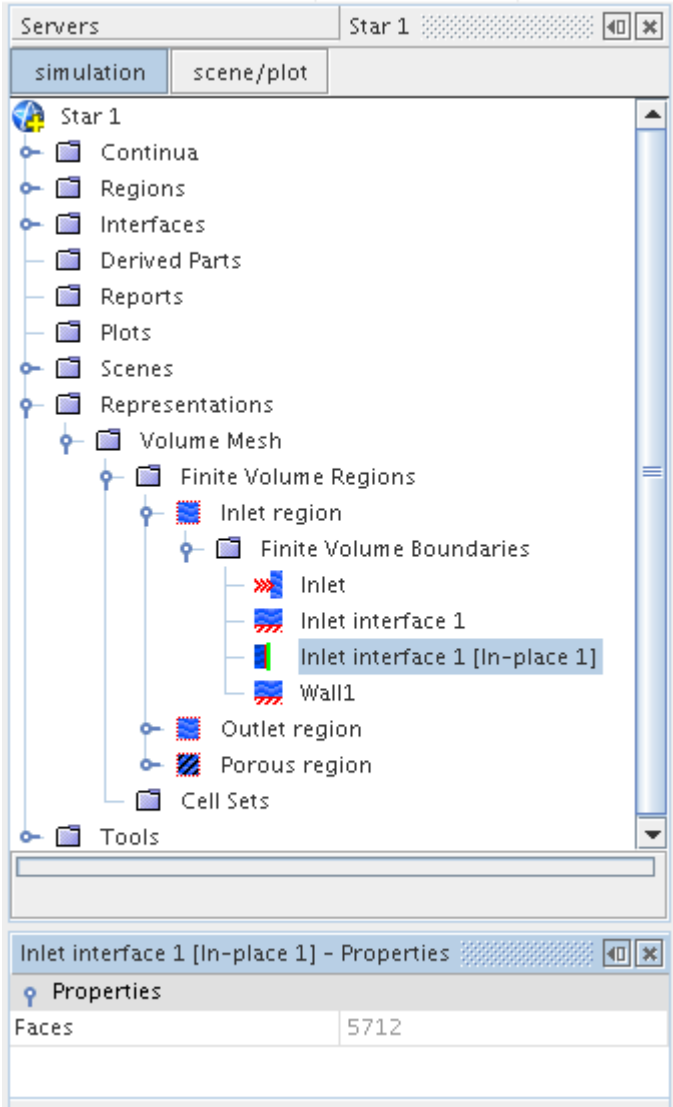
The results of the intersection process are displayed in the *Output* window along with messages related to the other initialization processes:



If the match between the two boundaries is conformal (that is, the master and slave boundary faces are all aligned and no additional intersections are required to produce a one-to-one connection) then this is indicated as such in the output, as shown above. If intersection of faces is required then the % difference in face area between the master and slave boundaries are shown instead. If the boundary perimeters are similar then only a small difference in intersection area is expected.

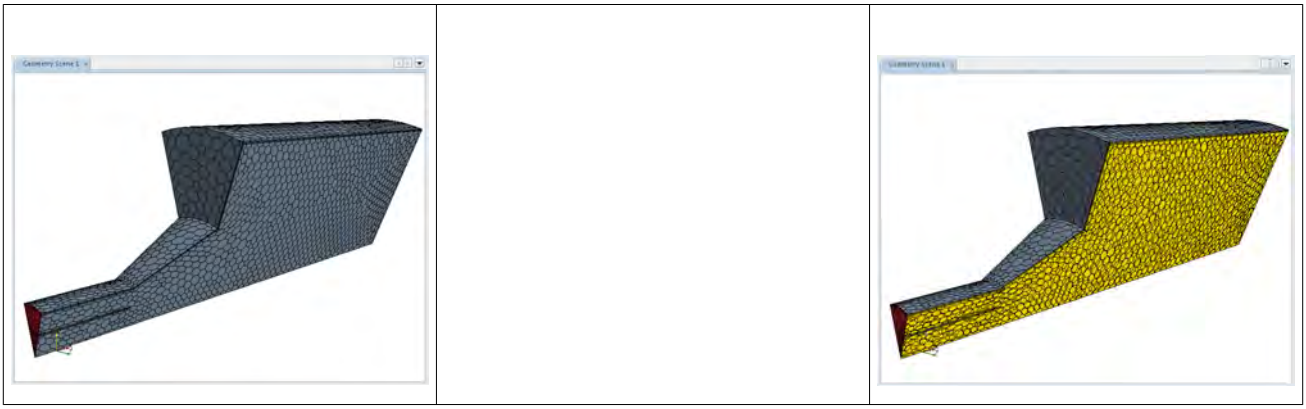
Additional information can be displayed in the output window by changing the interface verbosity.

On selecting the nodes of the interface boundaries within the **Representations** node, there is a non-zero value for the number of faces in the *Properties* window:



The interface boundary node is active.

The intersection process increases the number of faces that exist at the boundaries making up the interface. This is due to the process of subdividing the boundaries on each side so that a conformal, one-to-one connection is formed between each face making up the interface. An example before and after illustration showing the interface intersections on a periodic interface is shown below:



The intersection process may also introduce additional vertices into the model. This is not permanent and is undone when either the interface is reset or deleted. If the mesh is exported then the additional face divisions and vertices are not included. That is, the mesh is exported as if the interfaces did not exist.

If the simulation has been set up using the parallel server option, then the intersection process is done in parallel, resulting in a faster match being performed. Do not use this option if you have several interface pairs to intersect.

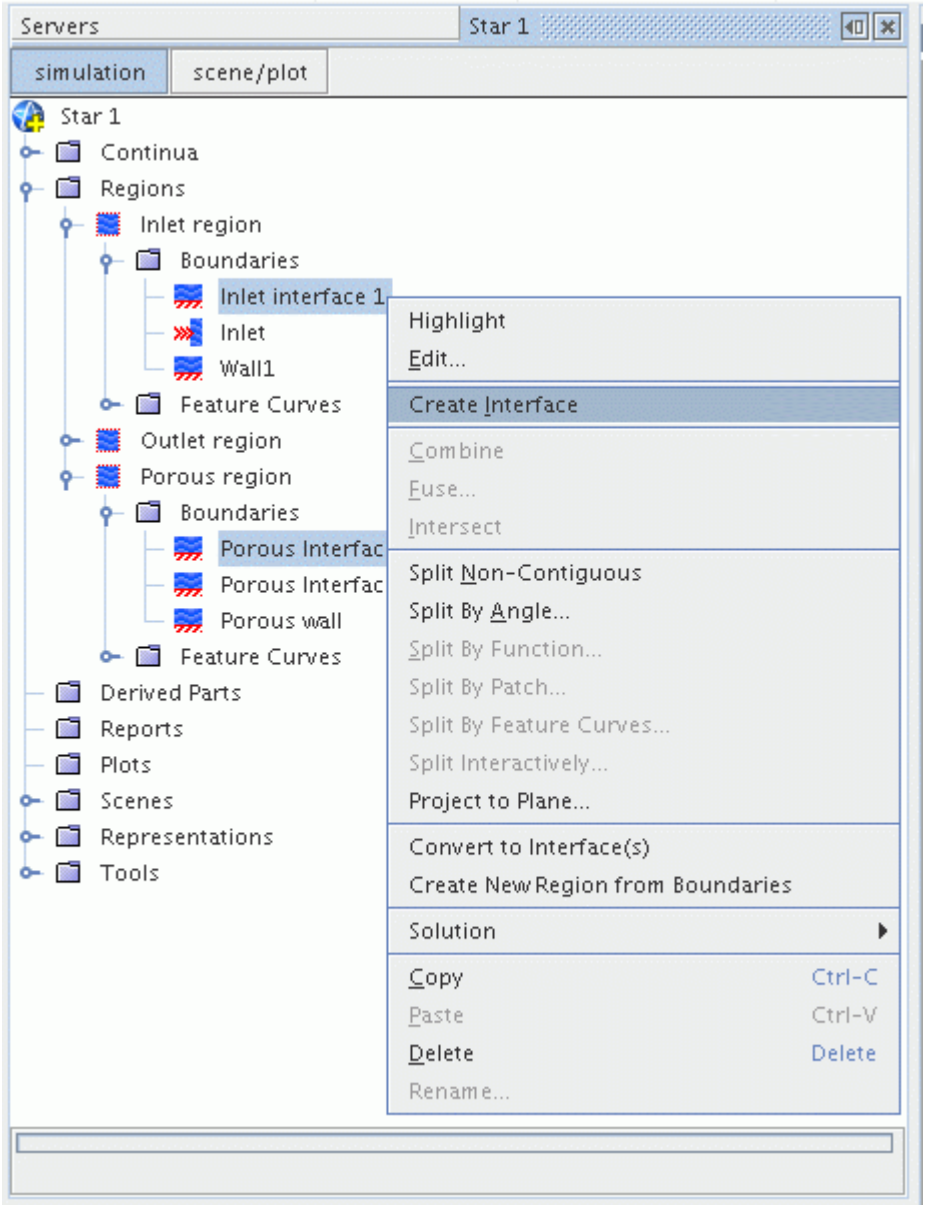
Creating a Boundary Interface

An interface can be created between boundaries to produce a conformal mesh interface (as part of the meshing pipeline), or to maintain physics continuity between regions or in the same region.

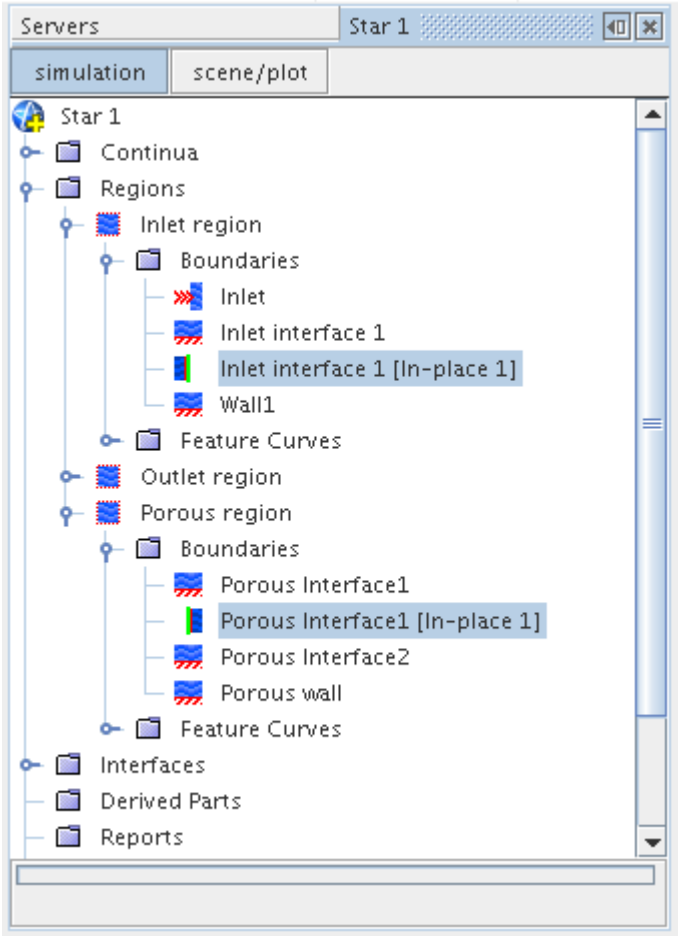
Do not confuse the act of creating an interface with converting boundaries to an interface, which is a similar but different operation used primarily for meshing purposes. In this latter instance, a single (usually common) boundary belonging to a surface mesh is duplicated and then an interface made between the original and the copy for the primary purpose of defining multiple regions for meshing.

To create an interface between boundaries, select the two boundaries that define the interface. The first boundary selected is termed the master boundary; the second is termed the slave boundary.

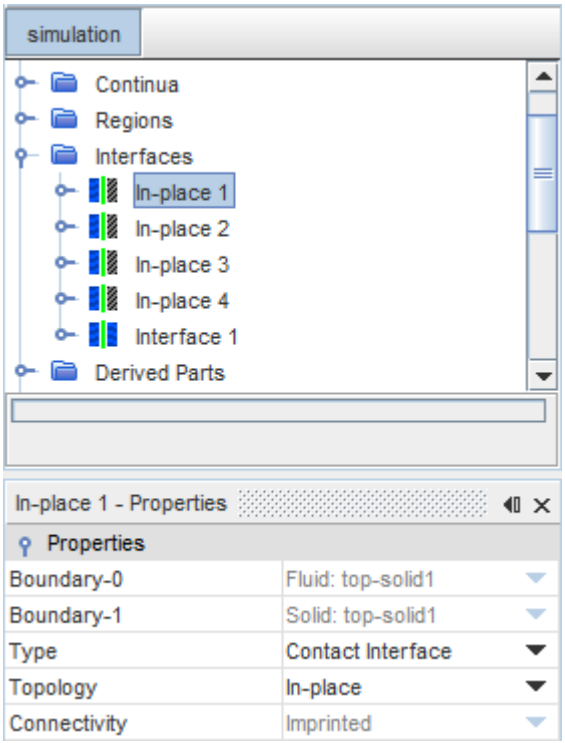
Once the boundaries have been selected (press and hold the Ctrl key to select the second node), right-click and select **Create Interface**.



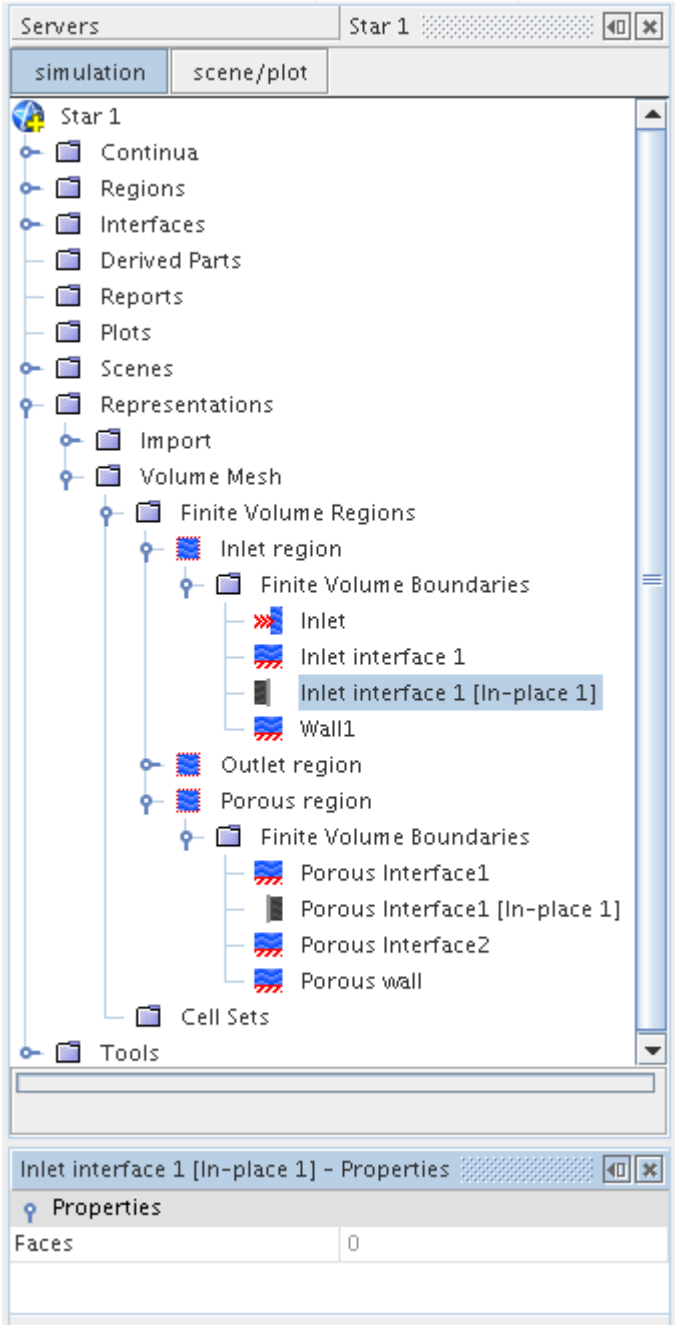
This action creates an interface between the two selected boundaries. Two new nodes appear in the simulation tree (under the **Boundaries** node for each region) indicating which boundary nodes are linked:



An interface node also appears in the **Interfaces** manager node, which has a default name, **Interface**, plus an identifying suffix. Selecting the interface node highlights the interface in the Graphics window.



To check the number of faces associated with each interface sub-node, open the subnodes of the Representations node and select the corresponding interface node:



The number of *Faces*, as shown in the *Properties* window, is zero due to the face that the intersection process has not yet been performed. The nodes appear grayed out, indicating that they are not yet active.

See Also:

What Is a Generalized Cylinder?

Region Manipulation

To fix surface or volume meshes that are not suitable for obtaining a valid solution, choose from among various techniques for manipulating regions.

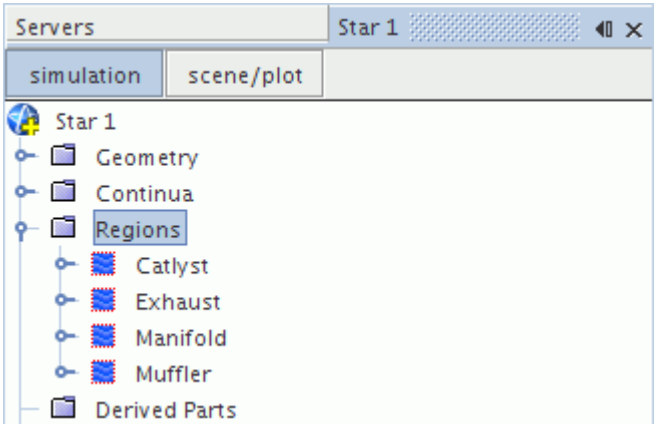
The region manipulation techniques are:

- Combining regions
- Splitting regions
- Creating new regions from boundaries (for surface meshes)

Also, the following common actions are available for region manipulation:

- Deleting regions
  - Deleting regions removes any associated Interfaces and reinstates the original boundaries of the remaining regions.
- Copying region properties
  - You only copy region type and continuum properties. Boundary conditions, values, or cells, are not copied.

Regions are represented in the STAR-CCM+ simulation under the **Regions** manager. The region properties are accessed by selecting the node. Right-clicking the node accesses the pop-up menu.



Contents:

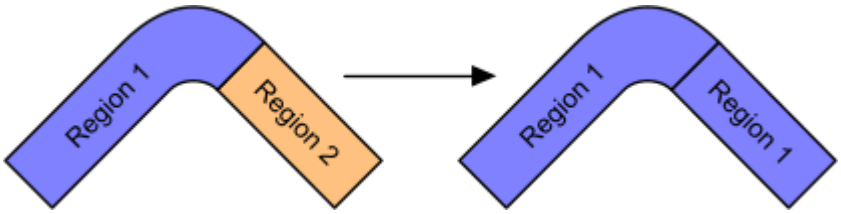
Region Combination
Region Splitting
Region Creation
Transforming Regions
Regions Manager Properties
Regions Manager Pop Up Menu

Region Combination

To have two or more computational domains share physical properties, combine regions.

There are two approaches that you can use to combine regions:

- Copy properties between regions and then create interfaces between adjacent regions.
- Combine the regions into one, as the following diagram illustrates, and then fuse or create interfaces between any adjacent boundaries.



If you do not fuse or create interfaces between adjacent boundaries, STAR-CCM+ solves the combined regions as if they are separate regions with the same properties. That is, there is no flow between the regions.

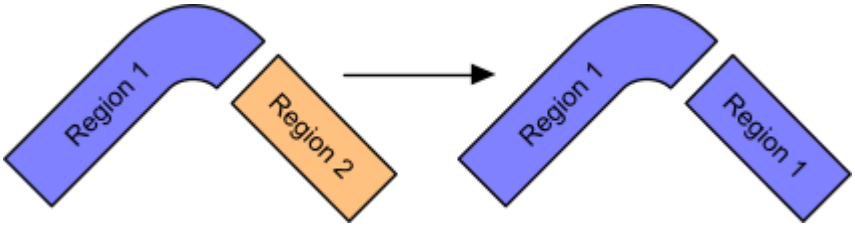
Note: It is more computationally efficient to fuse adjacent boundaries than to use interfaces.

You can combine multiple regions, providing:

- All regions share a surface or volume mesh representation.
- You associate all regions with the same mesh and physics continuum.

Initializing or running the solver reintersects interfaces. Therefore, combining regions before creating interfaces is more efficient, because the combination process resets interfaces and removes vertices from the intersection. Also, if these interfaces reference faces and vertices, the combination can remove them from other regions.

STAR-CCM+ allows combined regions to be non-contiguous. That is, there is no contact between the computational domains, but they share physical properties. The following diagram illustrates the combination of non-contiguous regions.



Contents:

Combining Surface Meshed Regions
Combining Volume-Meshed Regions

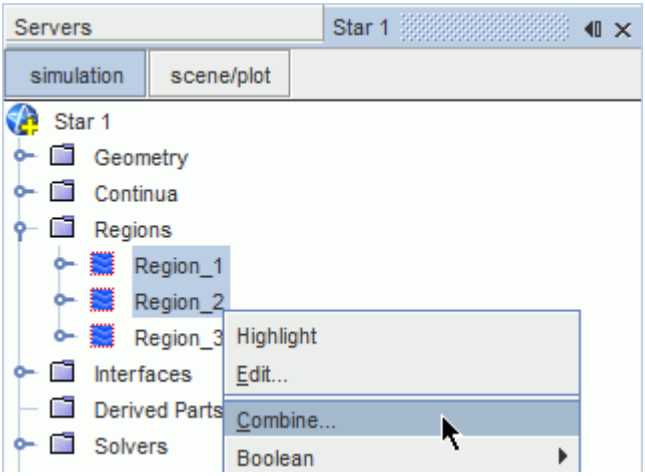
Combining Surface Meshed Regions

You can immediately combine two surface-meshed regions using the region pop-up menu.

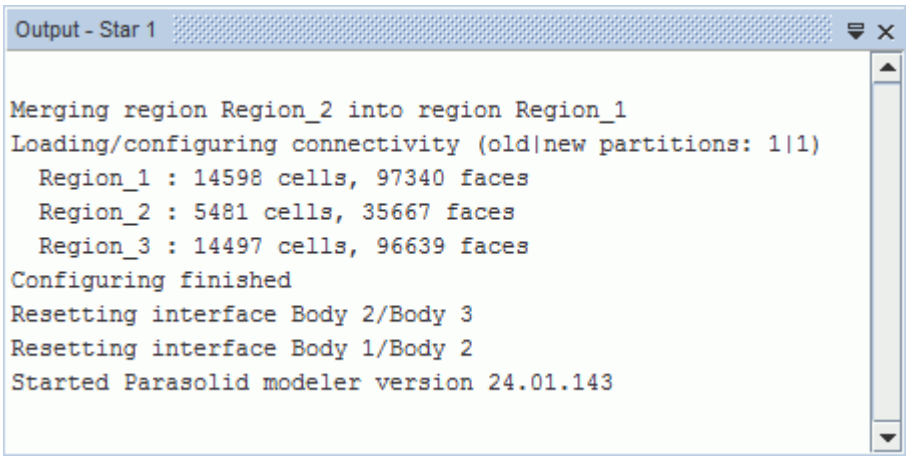
Note: The name of the combined region is the same as the region that you select first.

To combine regions that have a surface mesh:

1. Multi-select two or more regions.
2. Right-click a selected region node and select **Combine...**



Once the combine is complete, the *Output* window displays a summary of the operation.



Combining Volume-Meshed Regions

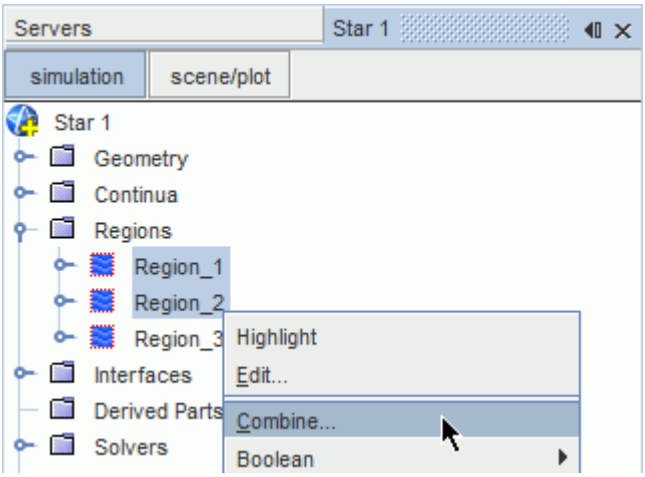
You can combine two volume-meshed regions using the region pop-up menu.

Note: The name of the combined region is the first region in the *Combine Regions* dialog (sorted in alphabetic order).

This operation can require you to fuse adjacent boundaries.

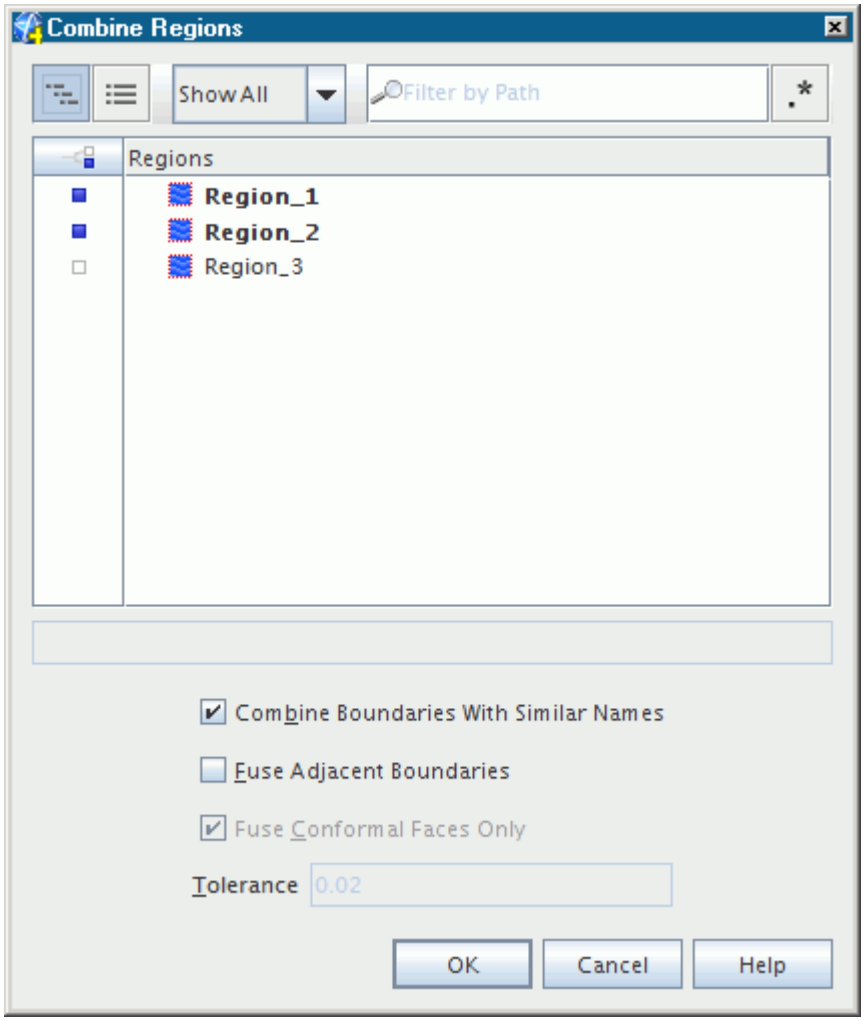
To combine regions that have a volume mesh:

1. Multi-select two or more regions.
2. Right-click a selected region node and select **Combine...**



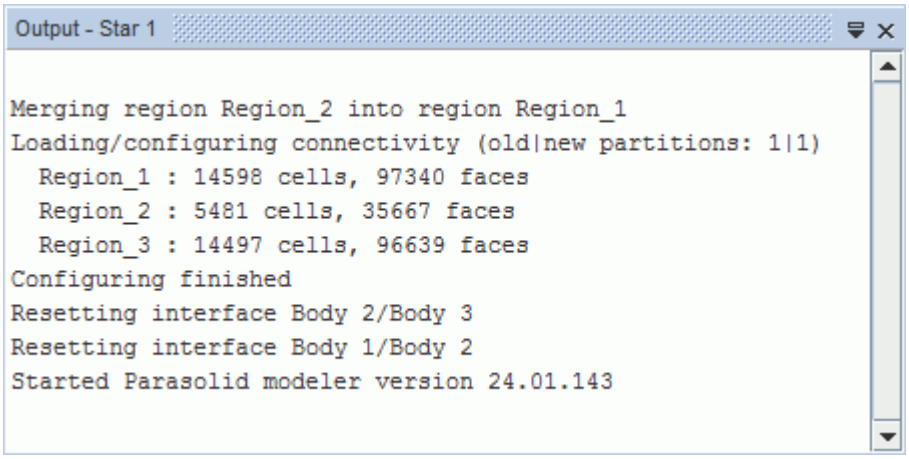
3. In the *Combine Regions* dialog, specify the region combination options.

   You can also change the selected regions within the dialog.



4. Click **OK** to combine the regions.

Once the combine is complete, the *Output* window displays a summary of the operation.



Contents:

Combine Regions Dialog

Combine Regions Dialog

The *Combine Regions* dialog provides properties for combining and fusing boundaries during the combination of volume meshes. This dialog also allows you to change the selected regions.

The following properties are available in the *Combine Regions* dialog:

Property Name	Property Description
Combine boundaries with similar names	When activated, this property combines boundaries that have the same base name. That is, boundaries that are the same up to the first space. For example, Symmetry 8 and Symmetry 9, but not Symmetry-8 and Symmetry-9. When deactivated, this property appends a space and an incremental digit to the name of one of two boundaries that share a name. For example, Symmetry and Symmetry become Symmetry and Symmetry 1.
Fuse adjacent boundaries	When activated, this property combines adjacent boundaries and replaces them with interior cell faces; thus removing the boundary.
Fuse conformal faces only	When activated, this property only combines adjacent boundaries that have cell faces with matching vertices (conformal).
Tolerance	Defines the maximum distance between fused boundaries as a percentage of the smallest edge length that is attached to each vertex. This property is only available when you select **Fuse adjacent boundaries**.

Region Splitting

Region splitting allows you to modify the simulation topology by separating regions using several characteristics. Thereafter, you can apply different continua and conditions to different parts of the model.

The following methods are available for splitting regions:

- Splitting by Surface Topology
- Splitting Non-Contiguous Regions
- Splitting Regions with a Field Function

Contents:

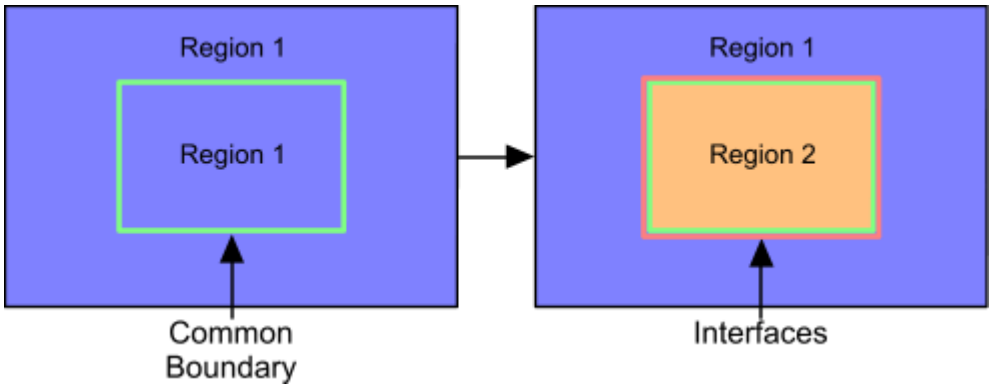
Splitting By Surface Topology
Splitting Non Contiguous Regions
Splitting Regions With A Field Function

Splitting By Surface Topology

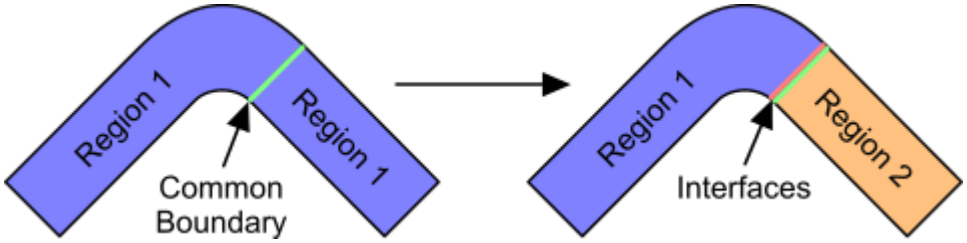
Splitting by surface topology splits closed, semi-manifold surface meshes into individual regions.

The following diagrams illustrate two scenarios where the split by surface topology option is useful:

- When one or more closed surfaces exist within another closed surface and each share a common boundary.



- When two or more closed surfaces share a common boundary.



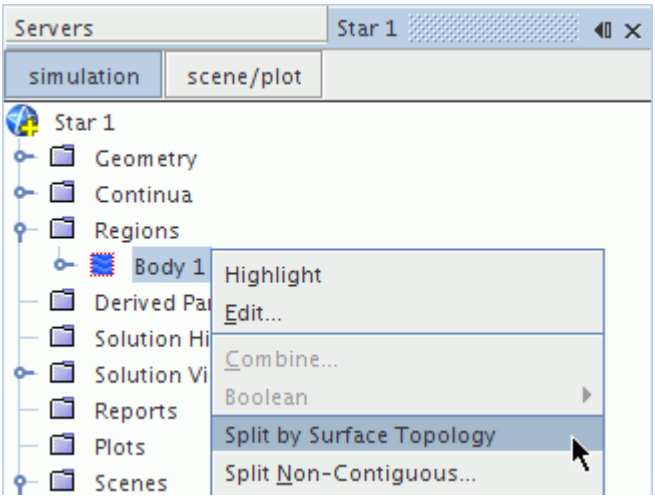
You typically find these types of geometry from CAD exported files that provide one region for all parts or bodies, with a common boundary between each.

If the regions have two adjacent boundaries, as opposed to one shared boundary, use non-contiguous splitting and manually create an in-place interface.

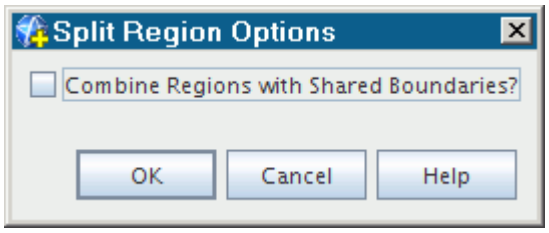
Note: Splitting by surface topology is only available when a surface mesh representation is present. Delete any volume mesh representations to access this option.

To split the geometry by surface topology:

1. Right-click on a region and select **Split by Surface Topology**.



2. In the *Split Region Options* dialog, specify the region split options



3. Click **OK** to split the regions.

   If you split invalid topologies, the resultant surface mesh requires repairing.

The split by surface topology operation copies shared boundaries to ensure that all regions are closed. The split also creates an in-place interface between copied boundaries to create a conformal mesh between the regions.

Contents:

Repairing Invalid Topology Splits
Split Region Options Dialog

Repairing Invalid Topology Splits

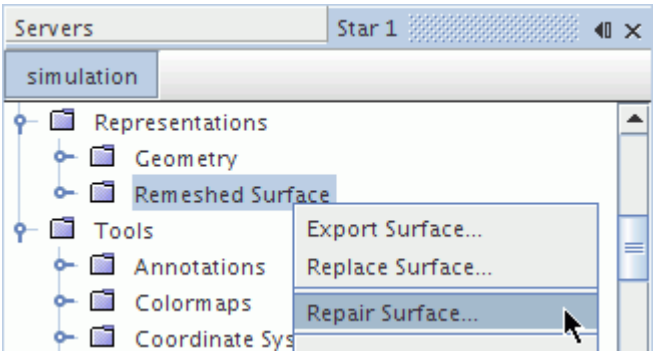
Invalid topologies are surfaces that are not closed. The split by topology operation can split these surfaces, but the resultant surface mesh requires repairing.

If you split an invalid topology, the Output window displays the following warning:

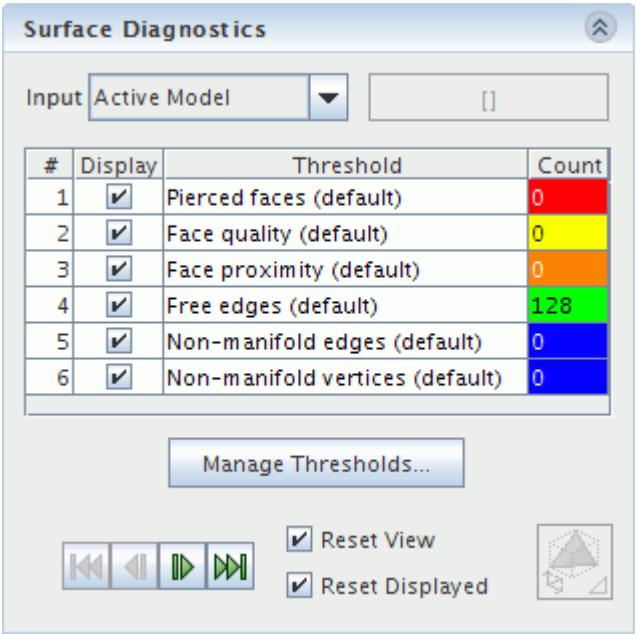
```
WARNING created invalid regions during split by topology
```

To fix the split by topology surface mesh:

1. Right-click the surface mesh representation and select **Repair Surface...**



2. In the *Surface Preparation* tool, use the Surface Diagnostics to identify surface mesh errors.



3. Use the Surface Repair tools to fix the surface mesh errors. The following tools are useful for fixing invalid topologies:
   - Zip selected edges
   - Delete faces
   - Fill holes

Split Region Options Dialog

The *Split Region Options* dialog provides a property that controls the region combination after the splitting process.

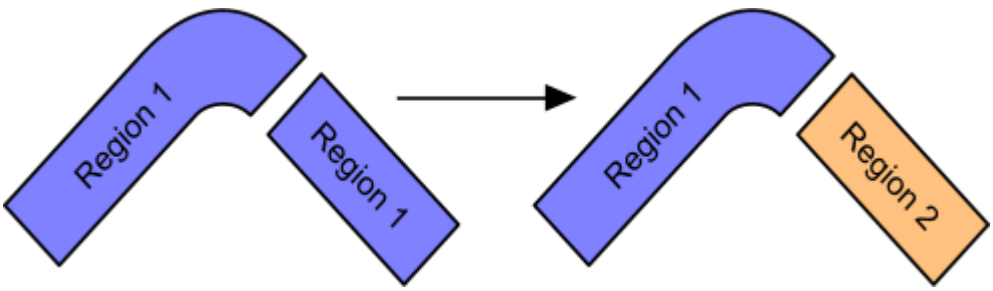
The following property is available in the *Split Region Options* dialog:

Property Name	Property Description
Combine Regions with Shared Boundaries?	When activated, this property combines regions that have the same boundary name after the split. This property is useful when many internal geometries exist that you want to belong to the same region and continuum.

Splitting Non Contiguous Regions

Non-contiguous region splitting creates regions when the original region contains surface or volume meshes that are spatially separated.

The following diagram illustrates non-contiguous region splitting:

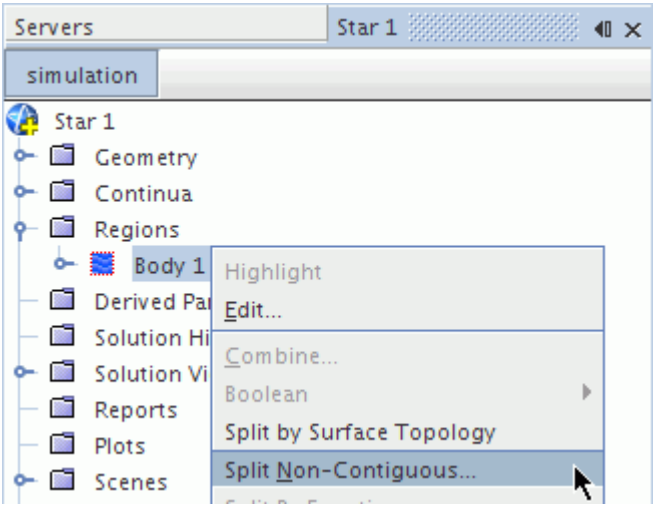


The following list defines contiguous regions:

- For surface meshes, each face shares an edge with another face in the region.
- For volume meshes, each cell shares a face with another cell in the region.

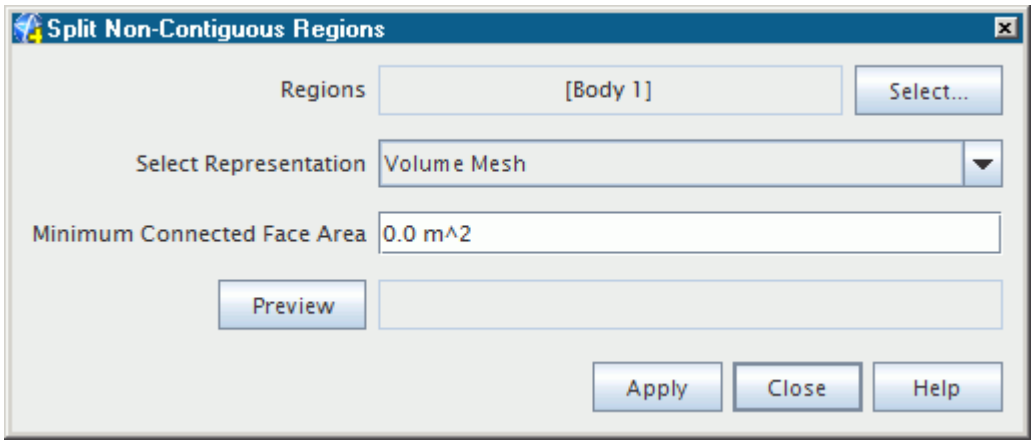
To split non-contiguous regions:

1. Right-click a region and select **Split Non-Contiguous**.



2. In the *Split Non-Contiguous Regions* dialog, set the region splitting options.

To change the selected regions, click the **Select** button.



3. Click **Apply** to split regions.

   If you attempt to split contiguous regions, the *Output* window displays a warning.

Contents:

Split Non Contiguous Regions Dialog

Split Non Contiguous Regions Dialog

The *Split Non-Contiguous Regions* dialog provides properties to control the splitting of non-contiguous regions. This dialog also allows you to change the selected regions.

The following properties are available in the *Split Non-Contiguous Regions* dialog:

Property Name	Property Description
Set Representation	Specifies the mesh representation to which the split applies. For example, if you want to split the remeshed surface, set *Set Representation* to **Remeshed Surface**.
Minimum Connected Face Area	Splits contiguous regions which have connected faces with a surface area below the user-defined tolerance. For example, if two regions share a face, you can split these regions by setting *Minimum Connected Face Area* to a larger value than the connected face surface area.

The following button is available in the *Split Non-Contiguous Regions* dialog:

Button Name	Button Description
Preview	Reports how many regions the split operation generates.

Button Name	Button Description
	This button is only available when you set *Select Representation* to a volume mesh.

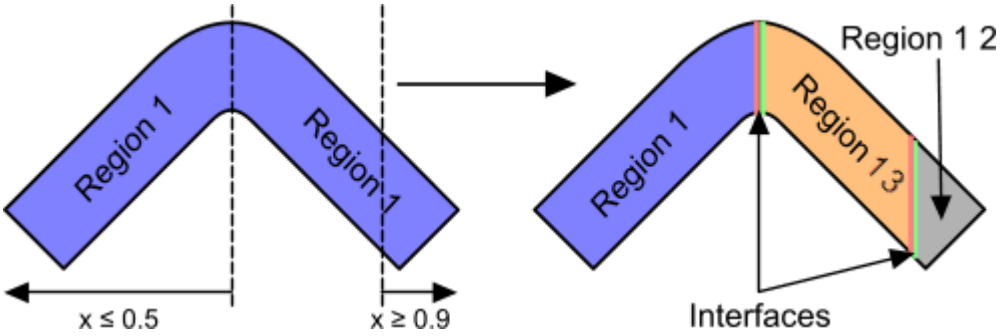
Splitting Regions With A Field Function

Splitting regions by a field function splits a surface or volume mesh region according to a user-defined expression.

This operation does not split cells. Therefore, if the cell vertices are not co-planar, the split leaves a non-uniform join between the two regions. The split generates two boundaries between the regions, from which you can create interfaces.

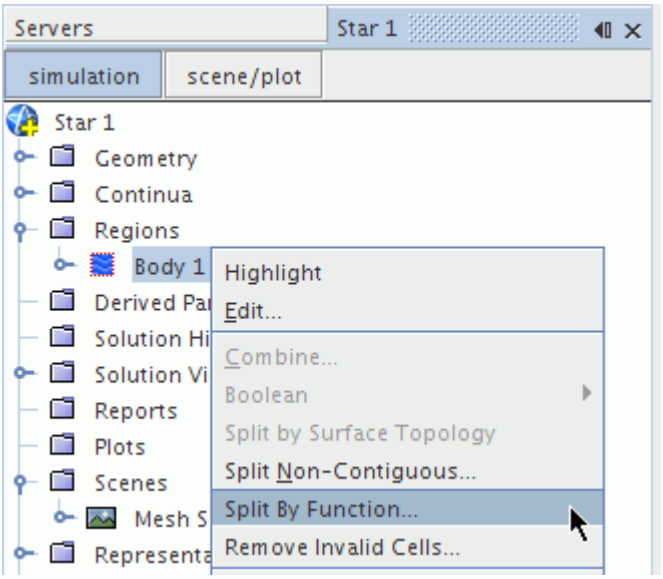
Note: The region with the most cells keeps the name of the original region. The naming conventions for new regions follow the ascending order of the number of cells in the region. For example, if the field function creates two new regions from Region 1, the new region with the least cells is Region 1 2, and the most cells is Region 1 3.

The following diagram illustrates field function region splitting:



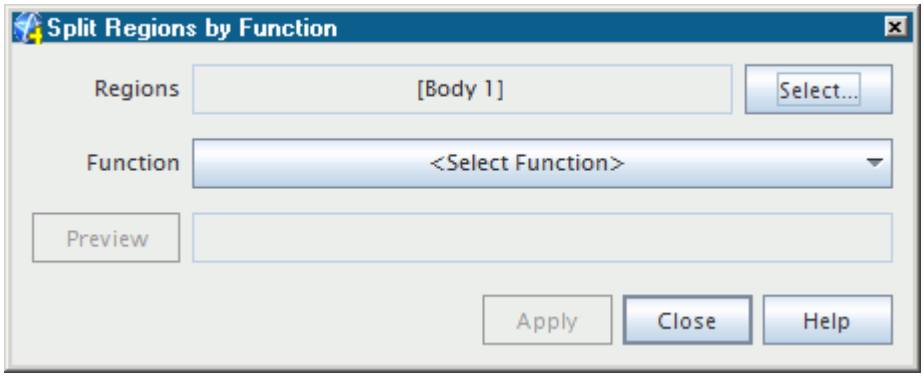
To split a region with a field function:

1. Define a field function that produces the required split
2. Right-click a region and select **Split By Function...**



3. In the *Split Regions by Function* dialog, set **Function** to the required field function.

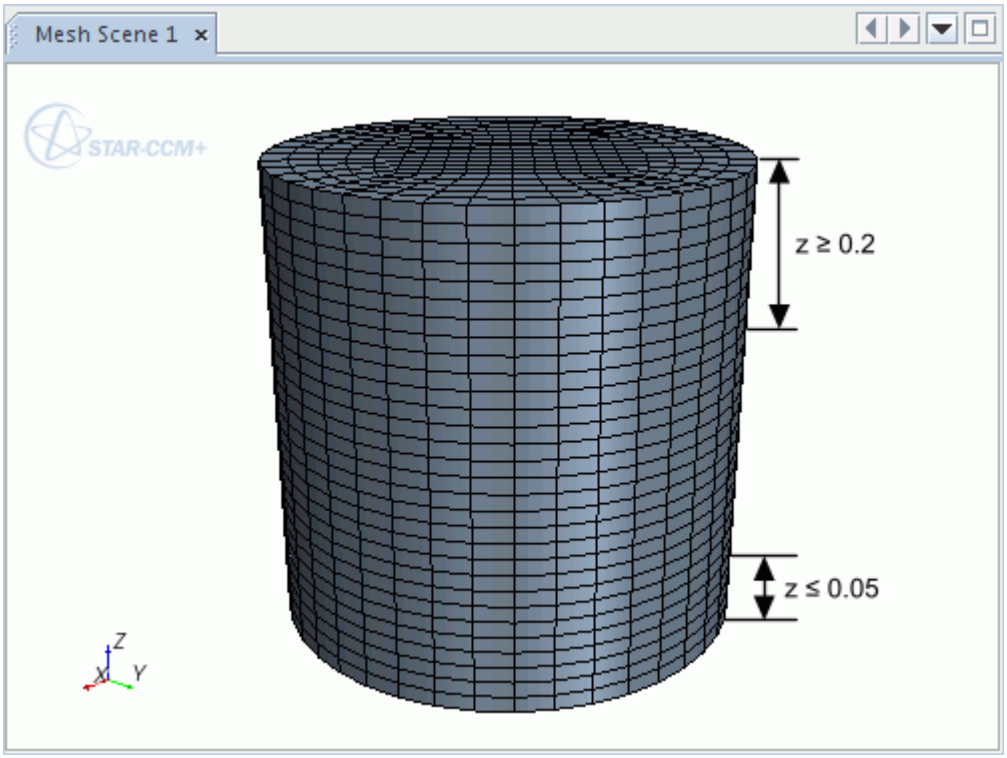
   To change the selected regions, click the **Select** button.



4. Click **Apply** to split the regions.

Example of a Field Function to Split a Region

This example splits a region into three as the following image illustrates.



You can split the region with the following field function:

```
($$Position[2] <= 0.05)? 1:(($$Position[2] >= 0.2)? 2:0)
```

In this case, after the split:

- The cells where 0.05 < z < 0.2 belong to Region 1, as it is the region with the most cells.
- The cells where z ≤ 0.05 belong to Region 1 2, as it is the new region with the least cells.
- The cells where z ≥ 0.2 belong to Region 1 3, as it is the new region with the most cells.

Contents:

Split Regions By Function Dialog

Split Regions By Function Dialog

The *Split Regions by Function* dialog allows you to select the field function for splitting. This dialog also allows you to change the selected regions.

The following property is available in the *Split Regions by Function* dialog:

Property Name	Property Description
Function	Specifies the field function that the split operation uses.

The following button is available in the *Split Regions by Function* dialog:

Button Name	Button Description
Preview	Reports how many regions the split operation generates.

Region Creation

You can create regions from existing boundaries to apply different physical properties to a portion of the fluid domain. Alternatively, you can create an empty region and assign parts to it.

Importing surfaces as regions automatically creates one or more regions from the surface topology and translation options. However, this region topology can be unsuitable for multi-region meshing and require redefinition by reallocating boundaries.

Additionally, importing surfaces as parts does not create any regions. Therefore, you can create regions so that you can allocate the appropriate parts before meshing.

When dealing with parts, you can also assign parts to regions automatically.

Contents:

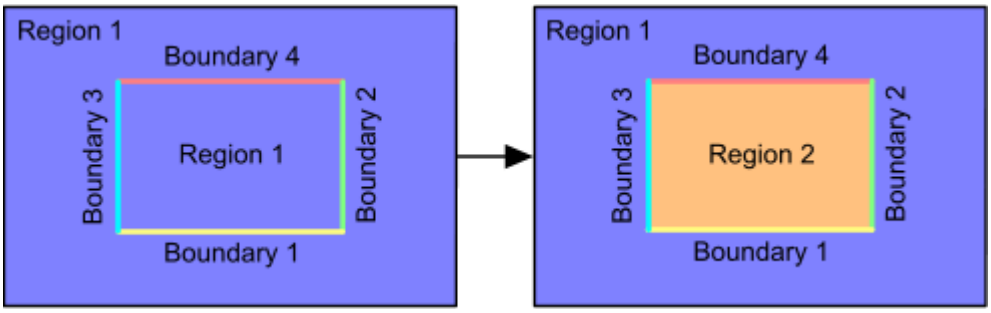
Creating Regions From Boundaries
Creating An Empty Region

Creating Regions From Boundaries

Creating regions from boundaries moves boundaries from one region to a new region. This feature allows you to modify the simulation topology and apply different physical properties to part of the fluid domain.

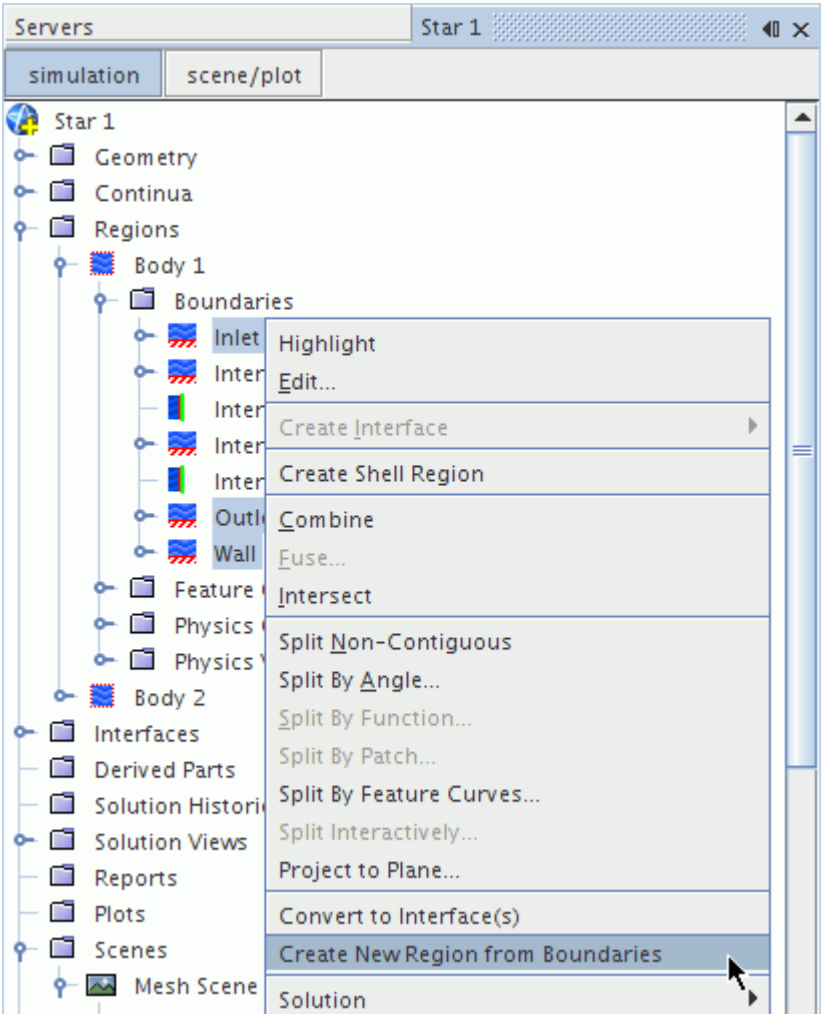
Before creating a region from boundaries, ensure that the created region will not share a boundary with any other region. In such cases, convert the shared boundary to an interface. This conversion creates an identical boundary and defines an in-place interface to connect the boundaries. You can then use the duplicate boundary to define the new region.

The following diagram illustrates the creation of regions from boundaries:



To create a region from boundaries:

1. Multi-select the required boundaries
2. Right-click one of the selected boundaries and select **Create New Region from Boundaries**.



If any region is open after this procedure, you can use the hole filler to close the surface. If you use the hole filler on a shared boundary, use an in-place interface to connect the boundaries.

Creating An Empty Region

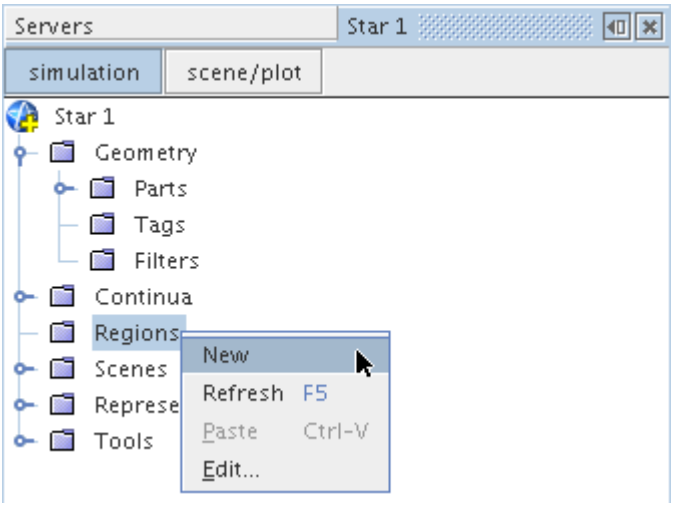
You can create an empty region and allocate parts to it for meshing.

An empty region is a region with no assigned parts. If a mesh continuum exists, creating an empty region assigns this continuum to the new region. Also the empty region has an empty wall boundary and feature curve so that you can assign part surfaces and part curves.

Note: The meshing models ignore empty regions.

To create an empty region:

1. Right-click the **Regions** manager and select **New**:



Transforming Regions

There are four types of transformations that you can apply to regions: Scale, Translate, Rotate, and Reflect.

The transformations are typically used to create separate regions when more than one mesh is imported, and these regions would not match spatially without some sort of transformation.

The transformations can be done on a single region or on multiple regions at the same time. The transformations that you apply to a region also affect the interfaces and boundaries that are attached to the region.

When regions have one of the following assigned to them–and no volume mesh, interfaces form between the surface meshes of the regions:

- geometry part
- imported part
- surface wrapper
- remeshed surface

Interfaces between surface meshes require physical contact between the parts that contain the regions, however, interfaces between volume meshes do not. Interfaces can exist between widely spaced regions of parts with volume meshes. For this reason, you can transform regions of parts with volume meshes without affecting the interfaces.

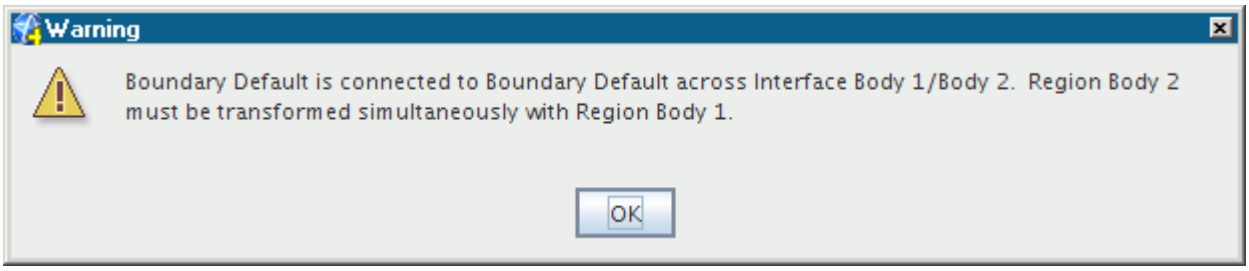
Requirements when Transforming Regions:

You can transform a region that is connected to another region through an interface without affecting that interface as long as both regions have a volume mesh.

When you transform regions that have volume meshes and are connected through an interface:

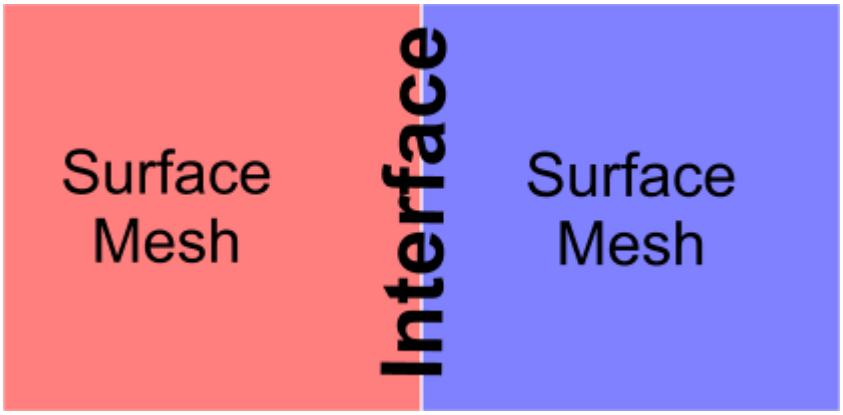
- There are no part contacts between the regions
- No errors occur
- The transformation is successful

If you attempt to transform a region that is connected to another region through an interface and one or both regions are assigned to parts without a volume mesh, an error occurs. The following error message is displayed:



When you transform regions that are assigned to parts without a volume mesh:

- Interfaces are created from part contacts between the surface meshes of the parts
- Part contacts can not be broken
- The transformation fails



When you transform regions and one or more (but not all) regions are assigned to parts without a volume mesh:

- The transformation is successful
- The interface is not maintained

To Meet Requirements when Transforming Regions:

- Create volume meshes for all regions that are connected through an interface.
- Translate all coincident parts with the region that you want to translate. Then imprint the parts and re-create the part contacts.

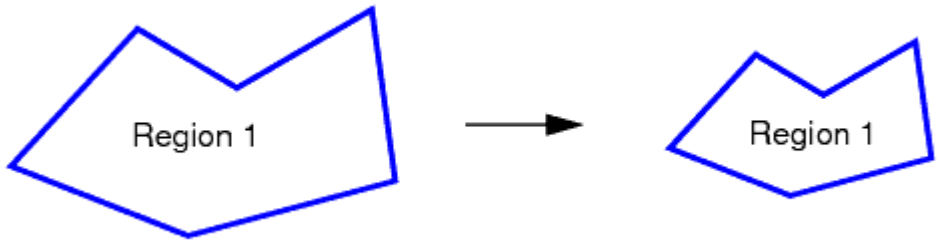
Contents:

Scaling Regions
Translating Regions
Rotating Regions
Reflect Regions

Scaling Regions

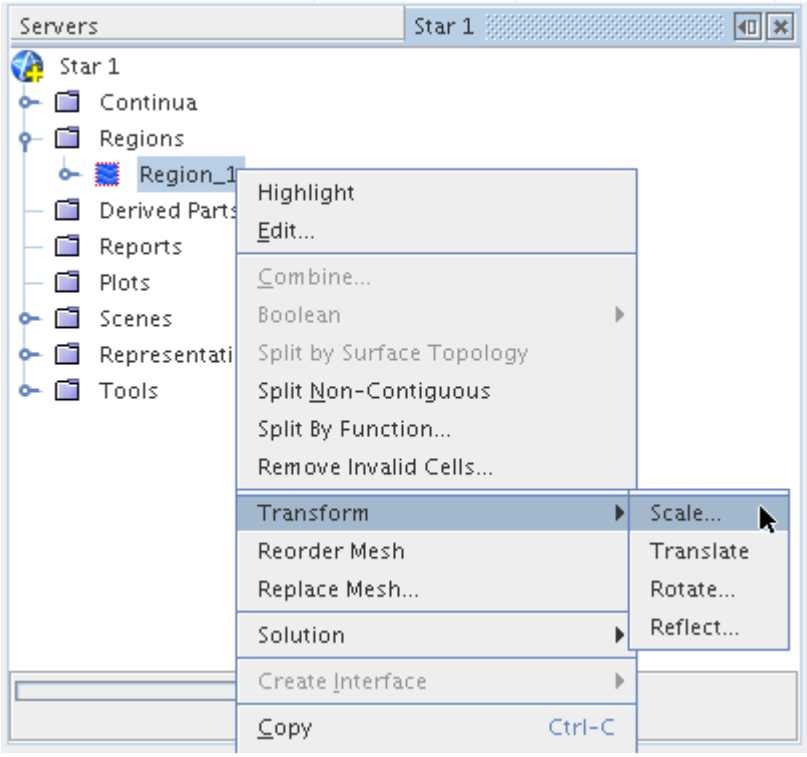
Scaling regions is especially useful when the mesh has been created in non-SI units.

If the region is displayed in an open *Graphics* window, the scaling is applied immediately to that window. The schematic below illustrates the scaling operation (with a scaling factor less than unity).

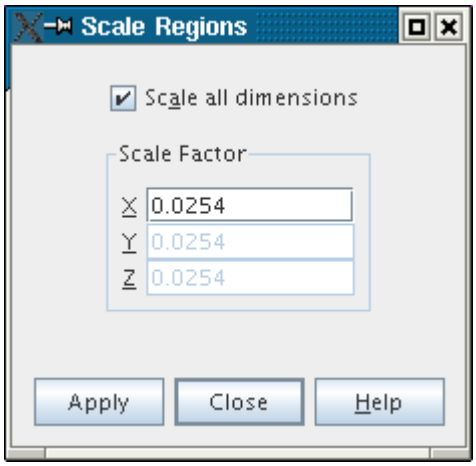


To scale a region:

1. Right-click the region of interest and select **Transform** > **Scale...** You can select multiple regions if you want to scale more than one region by the same amount.

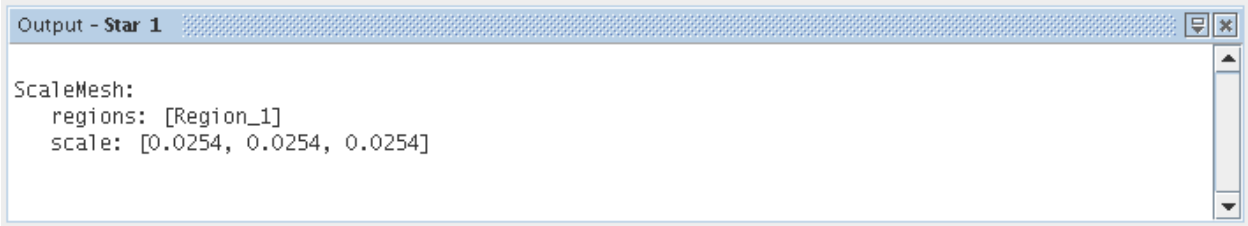


The *Scale Regions* dialog appears.



2. By default the **Scale all dimensions** option is activated, which means that all dimensions are scaled equally. In this mode, whatever you type into the **X** field is also entered into the **Y** and **Z** fields. Deactivate the **Scale all dimensions** option to scale each coordinate direction independently. In both modes, typing a number greater than 1 expands the mesh. Numbers smaller than 1 shrink the mesh.
3. Click **Apply** to begin the scaling process.

   The status bar at the bottom of the simulation tree shows the progress. After the scaling is complete, the progress bar displays `Finished`, and a summary message is displayed in the *Output* window.

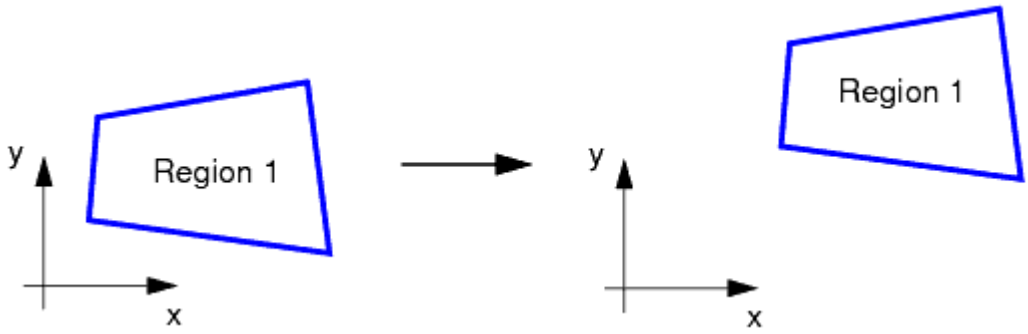


4. Click **Close** to close the dialog if no additional scaling is required.

Translating Regions

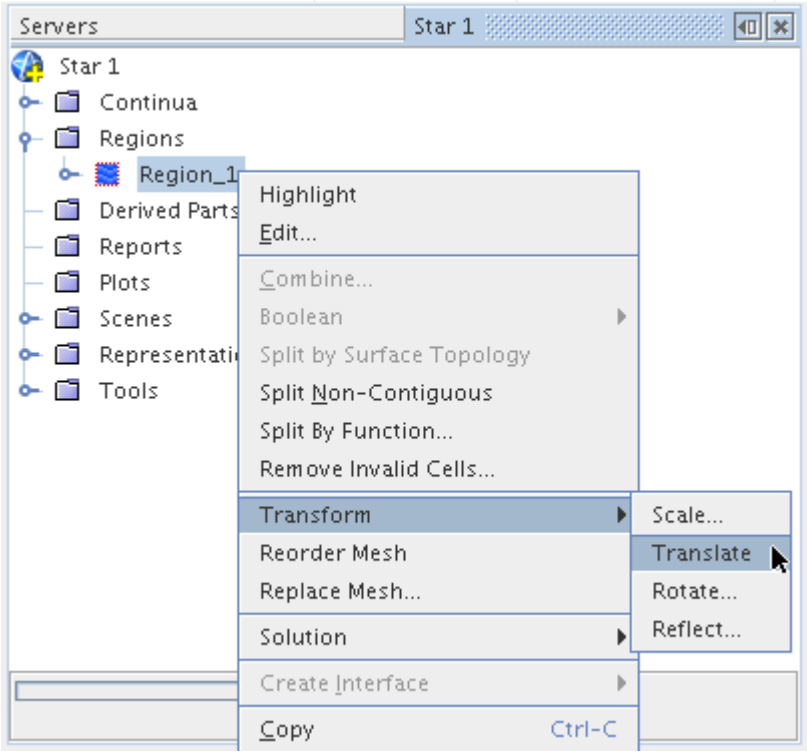
Translating regions is useful if you import multiple regions separately and you want to move them relative to each other.

In addition to changing the relative position of regions, you can use this feature to make sure that the region positions correspond to the position in space appropriate for plot data and tabular data sets. If the region is displayed in an open *Graphics* window, the translation is applied immediately to that window. The schematic below illustrates the translate operation.



To translate a region:

1. Right-click the region of interest. Select **Transform** > **Translate...** You can select multiple regions if you want to translate more than one region the same amount.



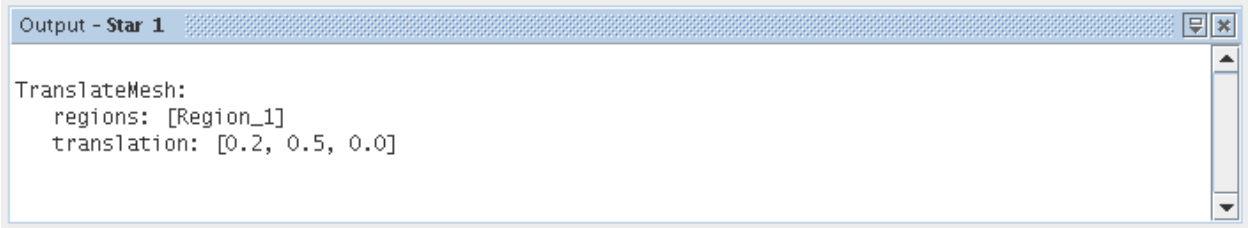
The *Translate Regions* dialog appears.

2. In the *Translation* fields, enter real numbers into the **X**, **Y**, and **Z** text boxes to define the translation vector. The units of each direction correspond to the model units. You can enter a negative value to move the region in the negative axis direction.

   Note: An invalid entry in any of the fields causes the background of the field to turn red. Clicking **Apply** with an invalid field automatically adjusts the invalid value to `0.0`.

3. Click **Apply** to translate the selections.

   The progress bar notifies you once the process is complete and a summary message is displayed in the *Output* window.

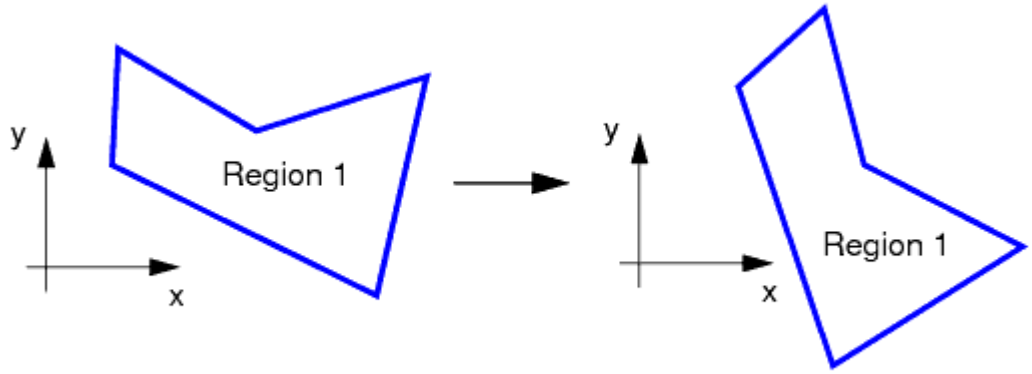


4. Click **Close**.

Rotating Regions

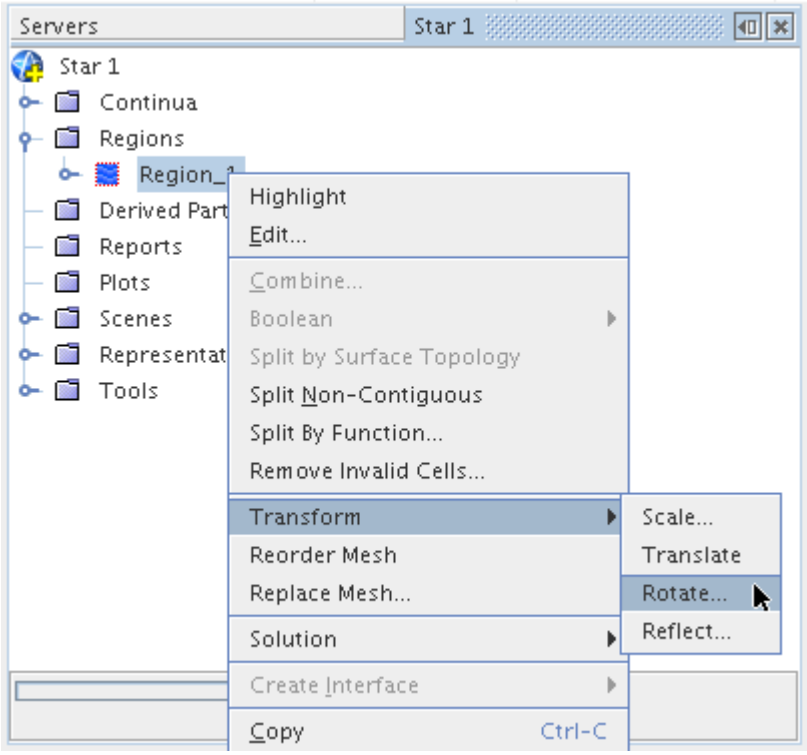
Region rotation can be used if the orientation of the imported mesh is not suitable for viewing, or if one region does not match the orientation of other regions.

When the region is rotated, the vector and tensor solution variables are rotated as well to keep them consistent. If the region is displayed in an open *Graphics* window, the rotation is applied immediately to that window. The rotation operation is illustrated below.

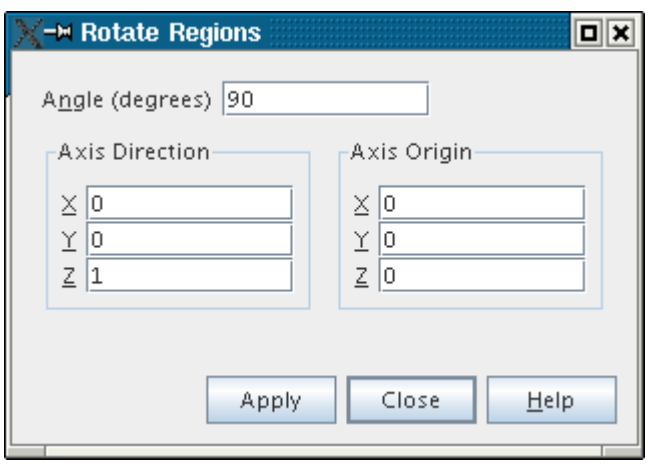


To rotate a region:

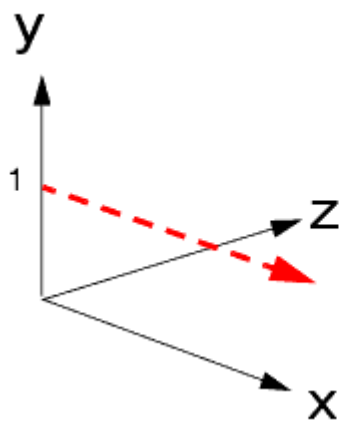
1. Right-click the region of interest and select **Transform** > **Rotate...** You can select multiple regions if you want to rotate more than one region the same amount.



The *Rotate Regions* dialog appears.



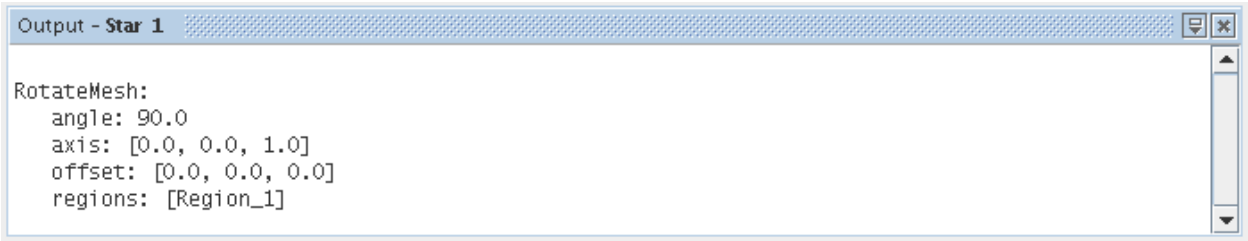
2. In the **Angle (degrees)** field, specify the amount of rotation. Rotation is specified using the right-hand rule around the axis of rotation. Negative entries are treated the same way as their positive counterparts.
3. Specify the axis of rotation using the *Axis Direction* and the *Axis Origin* components. For the example below, assuming you want to rotate around the red dashed axis, the *Axis Direction* components would be [1, 0, 0] and the *Axis Origin* components would be [0, 1, 0]:



Invalid entries cause the background of the field to turn red.

4. Click **Apply** to rotate the selections.

   The progress bar notifies you once the process is complete and a summary message is displayed in the *Output* window.



5. Click **Close**.

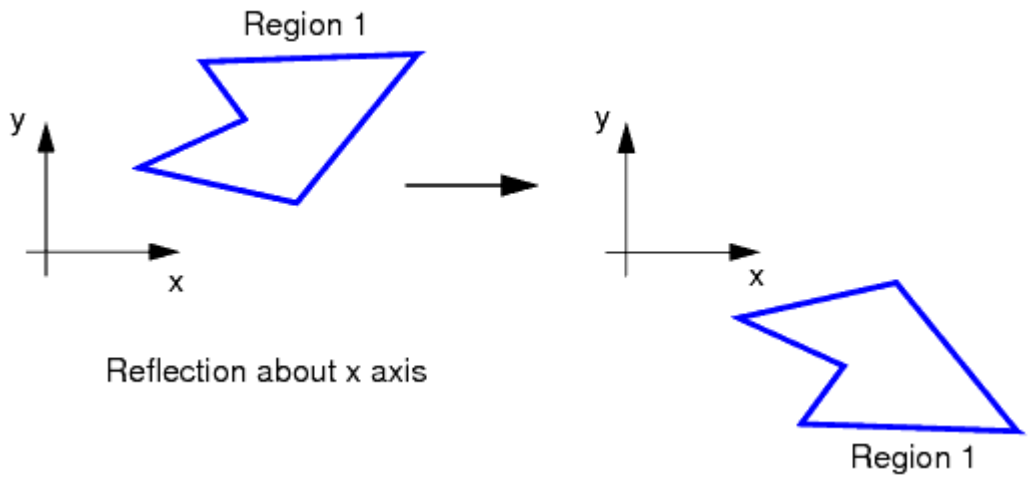
   Notes

   - Two-dimensional problems are treated in the same way as three-dimensional problems. Do not rotate the mesh off the X-Y plane.
   - When the mesh rotates, vector and tensor solution variables are rotated as well to keep them consistent. Specifically, velocity is rotated for flow models.
   - All active scenes update to reflect the new mesh configuration.

Reflect Regions

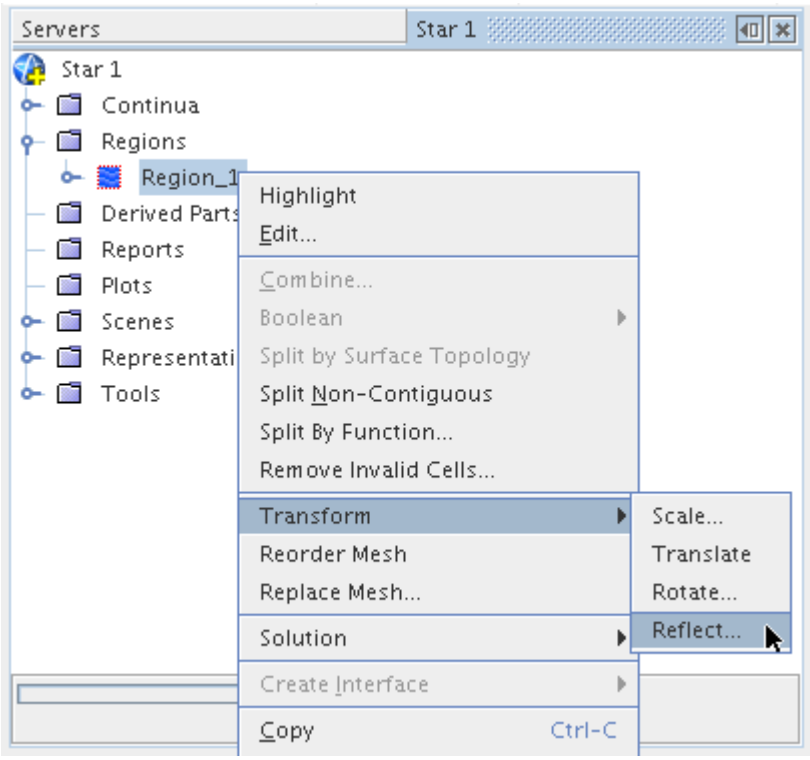
You can use the reflection option to reposition the region according to its mirror reflection about an axis or combination of axes.

If the region is displayed in an open *Graphics* window, the reflection is applied immediately to that window. The reflection operation is illustrated below:

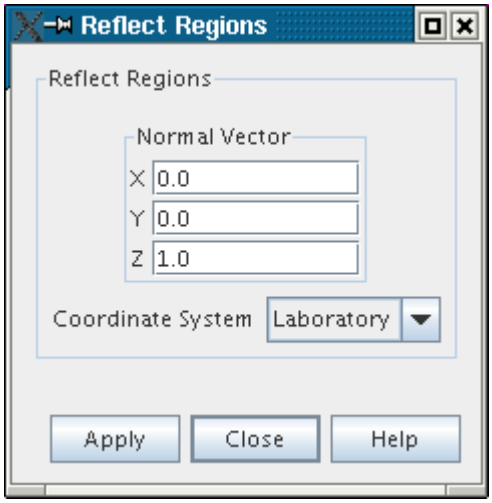


To reflect a region:

1. Right-click the region of interest and select **Transform** > **Reflect.**



The *Reflect Regions* dialog appears.



2. In the *Normal Vector* component, specify the axis about which you want the region to reflect.
3. The **Coordinate System** menu can be used to specify the coordinate system for reflection.
4. Click **Apply**.

   The progress bar notifies you once the process is complete and a summary message is displayed in the *Output* window.
5. Click **Close**.

Regions Manager Properties

The Regions manager node provides information on the number of regions in the simulation.

The following property is available from the **Regions** manager:

Part Selection Priority	Lets you prioritize the assignment of parts when assigning them with a dynamic query. See Prioritizing Assignment of Parts.
Regions	The number of regions in the simulation (read-only). This property changes when you add, combine, split, or delete regions.

Regions Manager Pop Up Menu

Right-click the Regions manager node to add regions or group them.

The following options are available when you right-click the **Regions** manager:

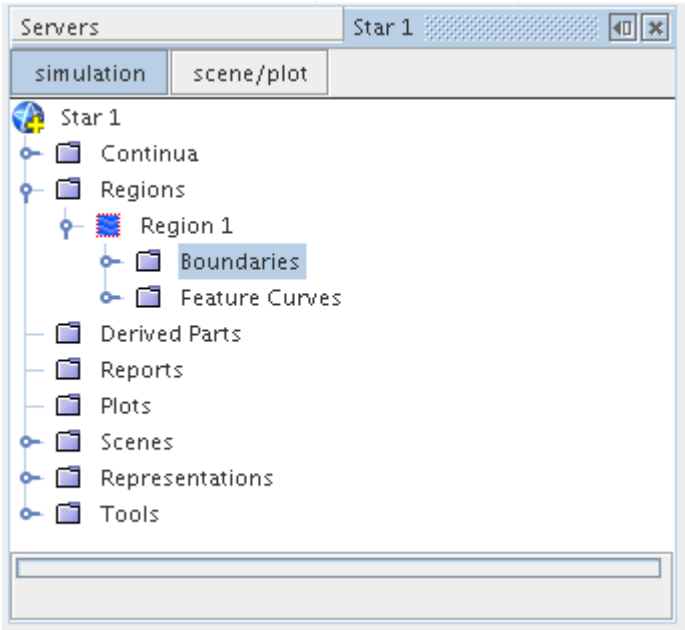
New Group	Creates a group for categorizing regions by creating a group manager in the **Regions** manager.
Group By	Group regions by: • Physics Conditions • Physics Values

	• Mesh Continuum • Physics Continuum • Parts • Type
Ungroup	Reverses the grouping process.
New	Adds a region to the simulation.

Manipulating Boundaries

It is often convenient to modify boundaries in the simulation without having to modify the mesh and re-import it.

Boundaries are easily accessible since they are included among the objects represented in the simulation tree. The **Boundaries** manager node, which is subject to common menu items, contains all the boundaries in the region.



In STAR-CCM+ it is critical that boundaries be appropriately separated and demarcated for specific conditions and values to be assigned, for both meshing and analysis setup. Surface and volume meshes originating from outside STAR-CCM+ usually have boundaries already defined in some fashion.

In most instances, boundary manipulation is only allowed when a single mesh representation is present. This is so that boundary mappings from one mesh to another can stay consistent.

Contents:

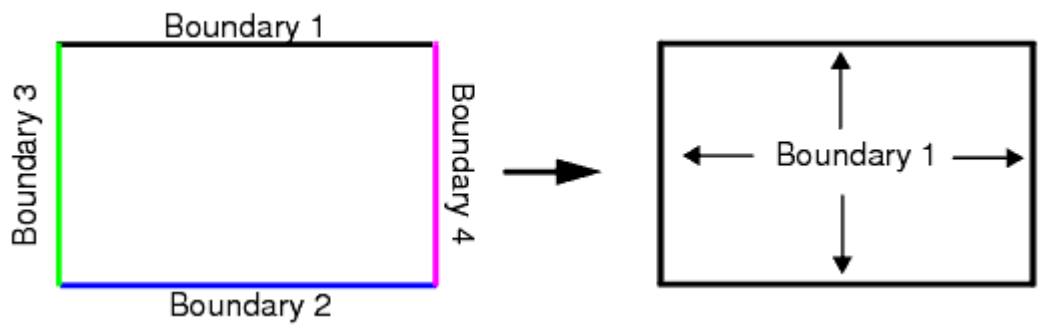
Combining Boundaries
Fusing Boundaries
Merging Boundaries
Splitting Boundaries
Creating New Boundaries
Projecting Boundaries
Selecting Boundaries
Copying Boundary Properties
Properties of the Boundaries Manager

Pop-Up Menu of the Boundaries Manager

Combining Boundaries

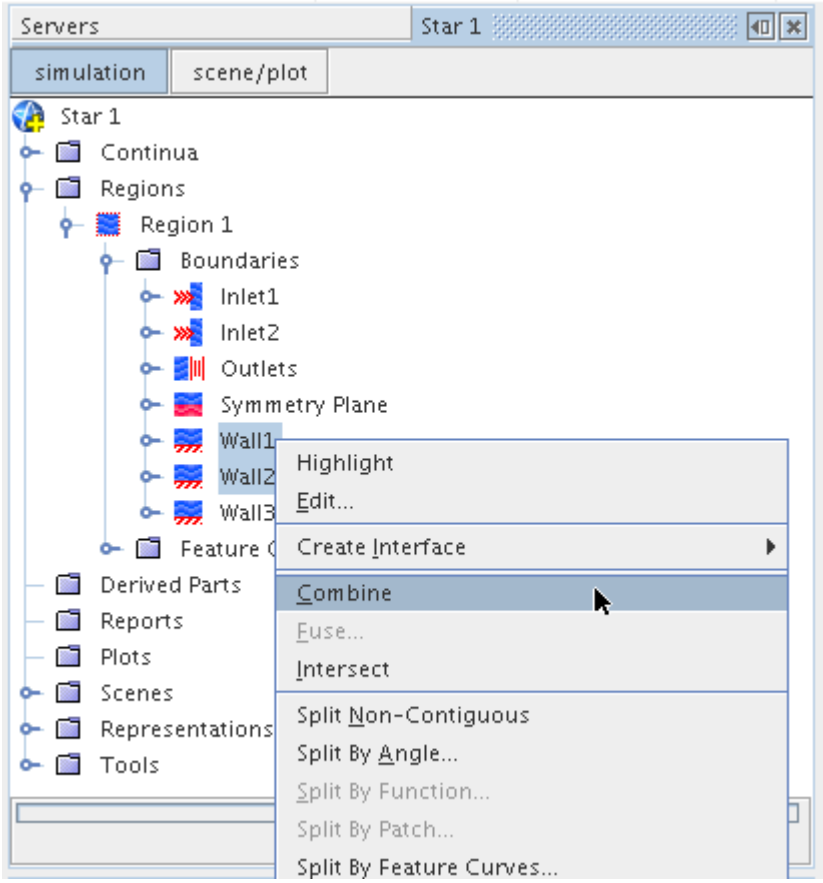
Combining two or more boundaries into a single boundary is useful when the imported surface or volume mesh contains separate boundaries which should have identical types, conditions, and values.

Combining the separate boundaries allows the boundary conditions to be set once and simplifies the topology. Combining boundaries does not modify the faces of the selected boundaries, it simply groups them together into one boundary. This operation is illustrated schematically below.



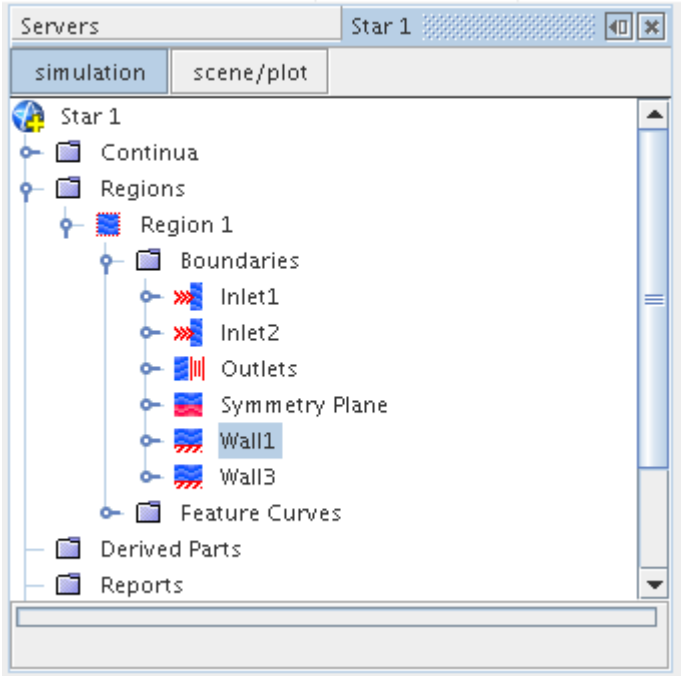
The boundaries being combined must belong to the same region and cannot include boundaries generated from computed interfaces. The boundaries being combined do not have to be contiguous. If the original boundaries were not topologically contiguous (no face from one boundary shares an edge with any faces from another boundary), the process can be reversed using the split non-contiguous boundary action.

To combine boundaries together, open the **Regions** node and the **Boundaries** node in the region of interest. Select the boundaries of interest using a multiple selection technique and right-click on the selection. Select **Combine** from the pop-up menu.



This will immediately begin the combining process and display a progress bar at the bottom of the simulation window. Once the process is complete a message will be displayed in the *Output* window indicating which boundaries were merged into which.

The boundary that is clicked first is the one that all the others are merged into. The simulation tree will be updated once the combine is complete.



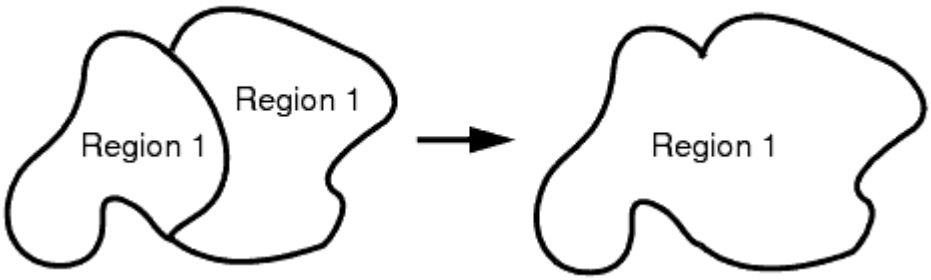
See Also:

Using Scenes to Interact with the Object Tree

Fusing Boundaries

The fuse command can be used to combine a pair of boundaries from the same volume mesh region and replace them with interior faces.

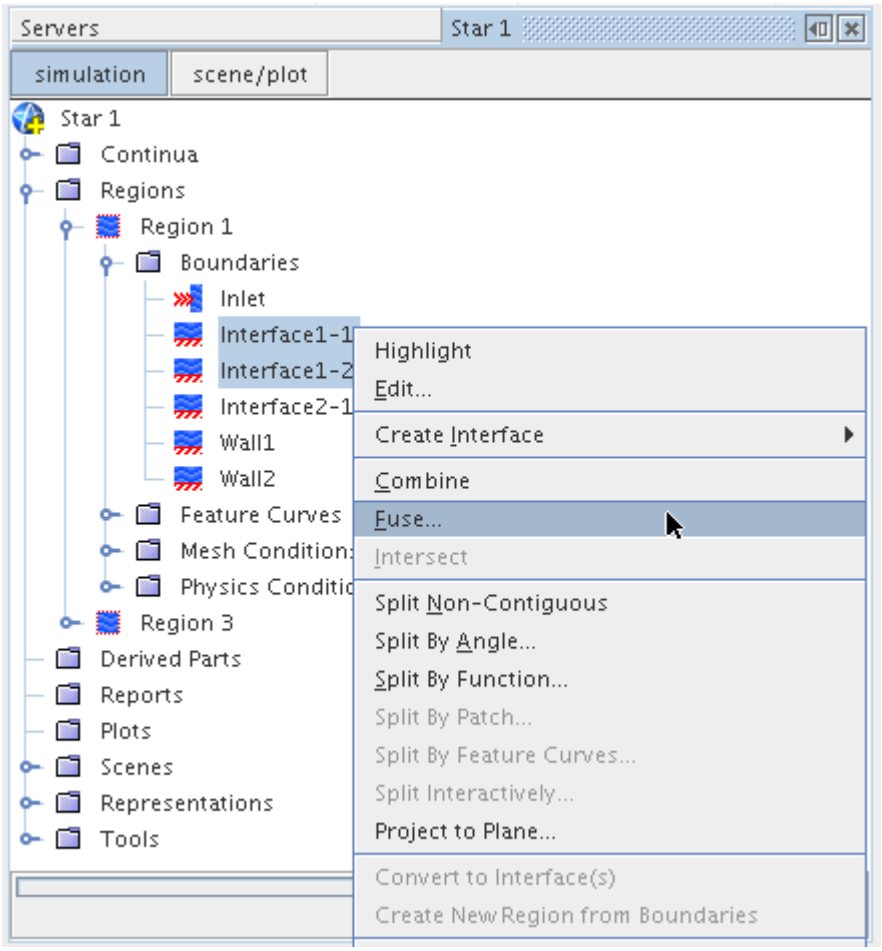
The fuse operation is illustrated schematically below.



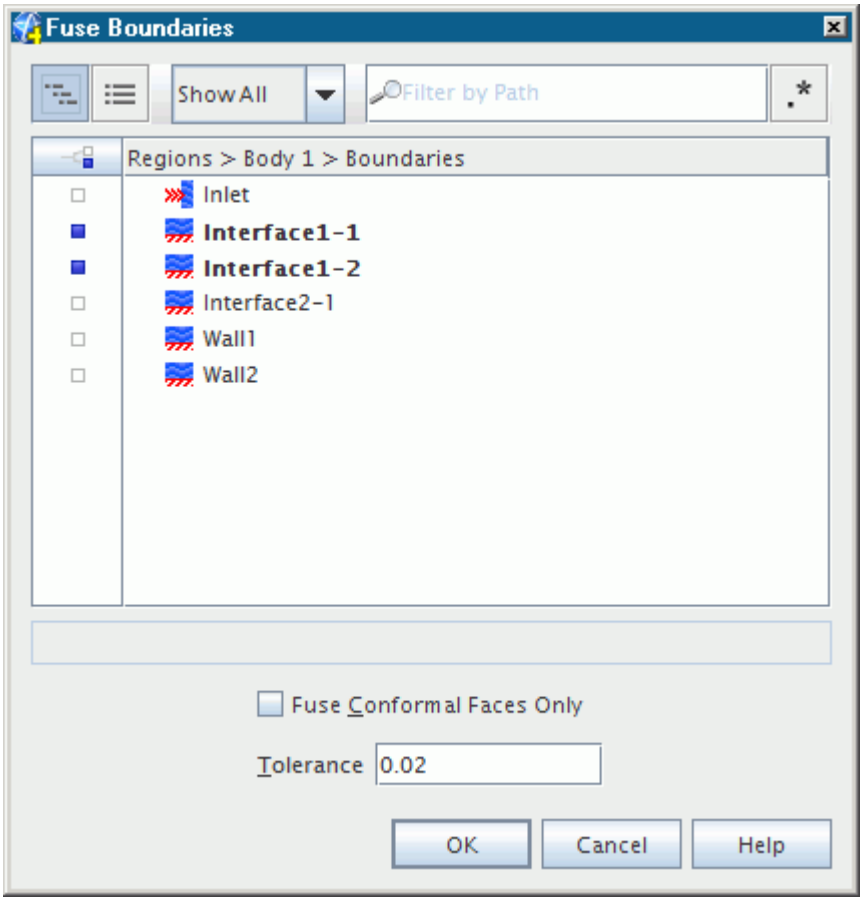
For the fuse command to be successful, the boundaries must be coincident in space, but not necessarily composed of conformal discretizations. If the two boundaries are completely conformal or can be matched exactly via the interface computation, the selected boundaries will be deleted, and the matched faces will be added to the interior of the region. If the faces cannot be completely matched, any faces that do match will be added to the interior of the region while the remaining faces will be left in the original boundaries.

The advantage of removing boundaries by fusing is that there is no additional overhead involved in the calculation (as might be incurred at an interface).

To fuse boundaries, open the **Regions** node and the **Boundaries** node in the region of interest. Select the boundaries you want to fuse using a multiple selection technique and right-click on the selection. Select **Fuse...** from the pop-up menu.



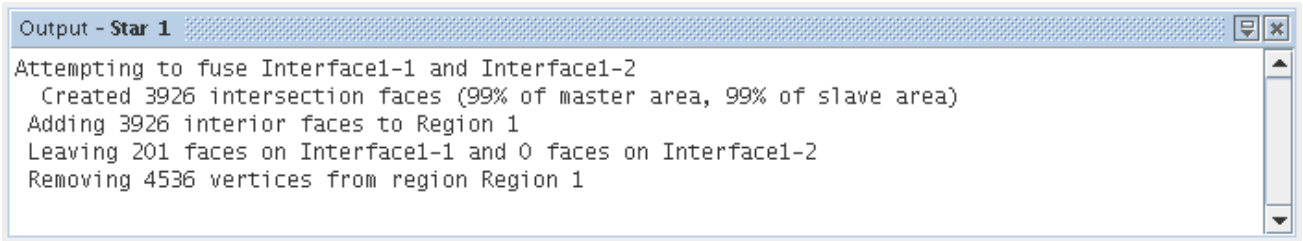
This opens the *Fuse Boundaries* dialog.



The originally highlighted boundaries will be automatically selected in the dialog with a ticked checkbox. If you wish to match only conformal faces (faces which have vertices at the same coordinates), select the **Fuse conformal faces only** option. Otherwise, the full interface intersection computation is performed on the selected boundaries.

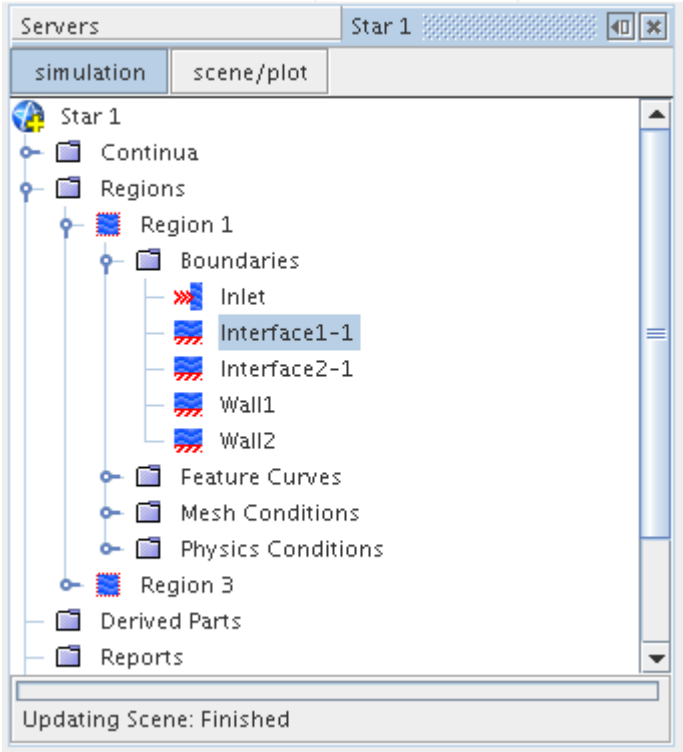
The boundaries must be geometrically adjacent and the **Tolerance** in the previous dialog is a measure of close the boundaries must be to be considered suitable candidates for fusing. The tolerance represents the amount that each vertex is allowed to move during the intersection process, and is the same as the *Intersection* tolerance specified on direct interfaces.

After entering the tolerance, click the **OK** button. This will begin the fusing process and display a progress bar at the bottom of the simulation window. Once the process is complete a message will be displayed in the *Output* window indicating the results of the fusing process.

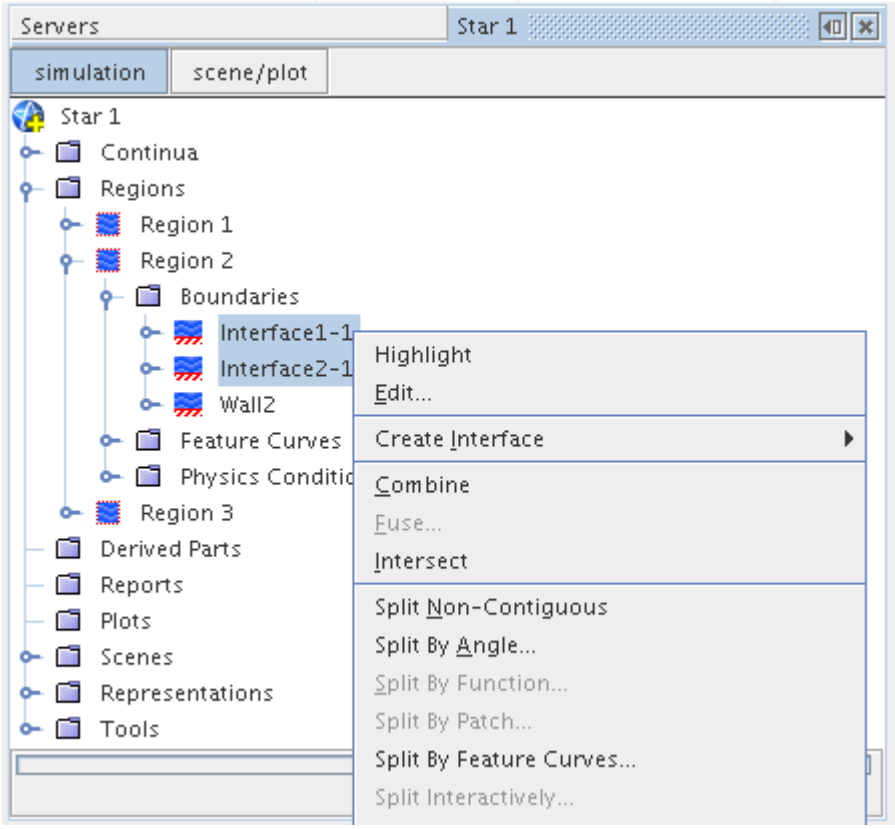


Since an interface intersection is done when fusing boundaries (to convert the boundary faces into interior faces) you will see several messages about faces and vertices changing.

The simulation tree will be updated once the fuse is complete, and if all of the faces were matched with no remainders, neither of the boundaries that were selected will exist after fusing.



In some situations the **Fuse...** menu item is grayed even though you have selected more than one boundary.



There could be four reasons for this:

- one or more of the boundaries are taking part in an interface;
- one or more of the boundaries were created by an interface;
- the selected boundaries belong to more than one region; or
- you don't have a volume mesh representation currently displayed

Merging Boundaries

Boundaries that belong to a surface mesh can be merged together if they are in the same region and have surfaces intersecting one another.

When two or more boundaries intersect then the merge operation will re-tessellate the surface so that all the triangles in the boundaries are joined together. In the case of overlapping/co-planar boundaries, a common shared boundary is created to replace the overlapping area, so as to facilitate the setup for multi-region meshing cases for example.

An alternative intersection process for surfaces, which can be in different regions, is provided in the intersect faces tool as part of surface repair. The surface repair tool can also be used to merge or imprint two overlapping boundaries.

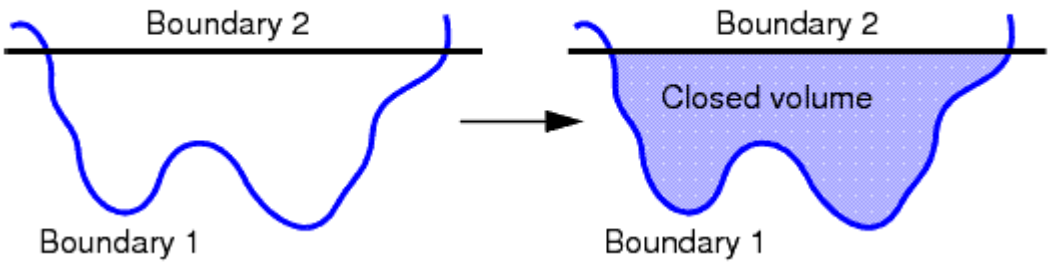
Contents:

Merging Intersecting Boundaries

Merging Intersecting Boundaries

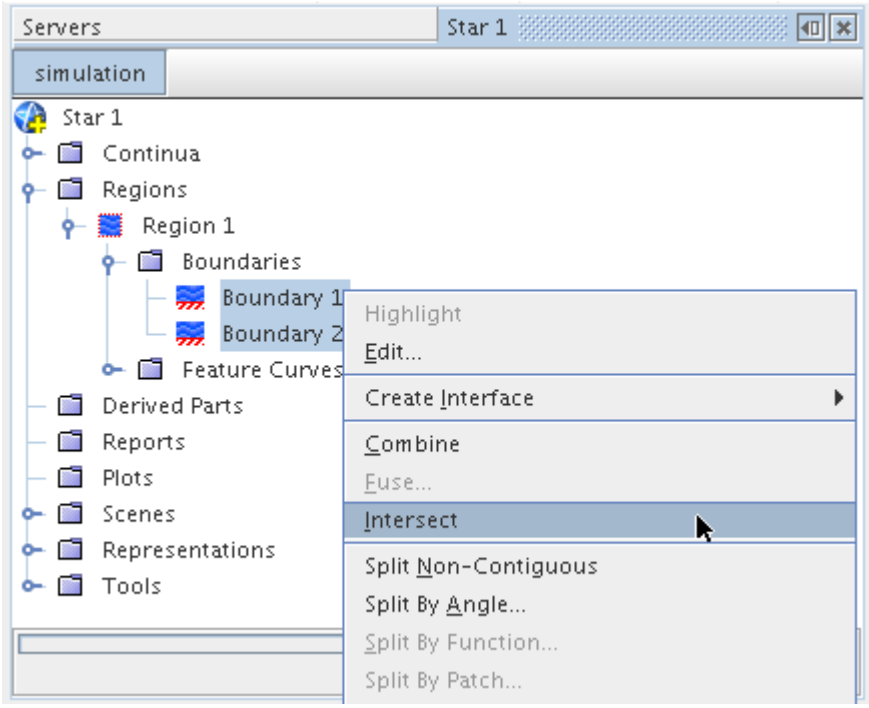
The merge intersecting boundaries feature allows two or more intersecting surface mesh boundaries to be merged together so that the triangles making up the surface are joined together.

This type of merging is achieved by re-tessellating the locally intersected surface so that new vertices and triangles are introduced at the intersection. If the input boundaries fully intersect then the result is a closed (watertight) surface volume (but not necessarily manifold), as illustrated below:

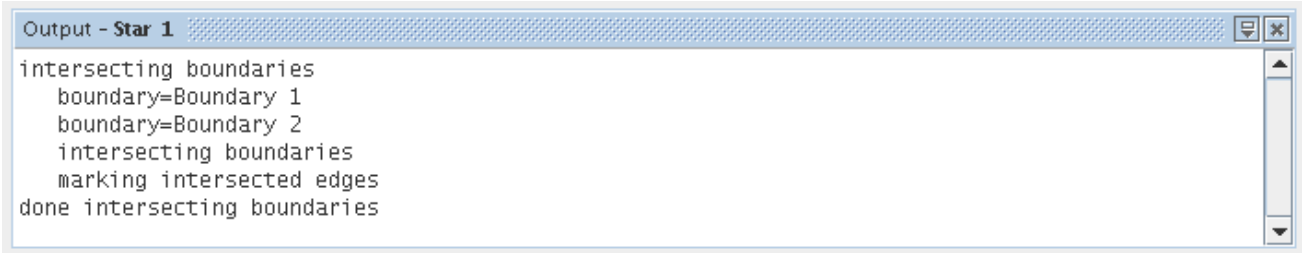


In the above example, boundaries 1 and 2 are fully intersecting but not joined in any way. After the intersection process, both boundaries have been re-tessellated at the intersection plane, forming a closed volume. You should note that no boundaries are removed or created by this process, so any surface triangles that are not required after the intersection should be removed in order to leave a manifold surface. New feature curves are however automatically created at the intersection.

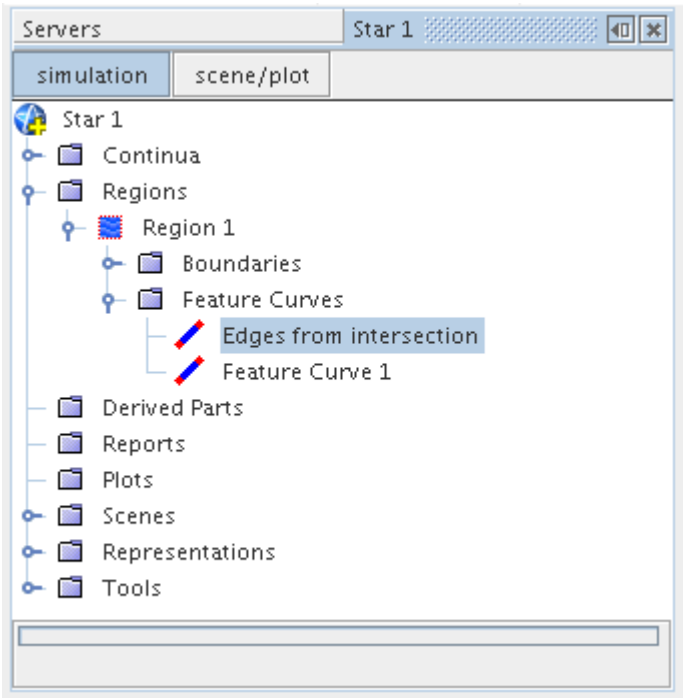
To intersect two or more boundaries, open the **Boundaries** node in the region of interest and then select the boundaries which are to be intersected. Right-click on any of the boundaries and select **Intersect** from the pop-up menu:



The intersection process will proceed and the status will be reported in the *Output* window:



Once the process is done, the current scene will be updated with the new surface triangulation at the intersection of the selected boundaries. The simulation tree will also show the new feature curves created:



The following conditions should be noted when using the boundary merge intersection tool:

- The boundaries to be intersected must belong to the same region. If they belong to different regions, then you should combine the regions first before proceeding.
- The sequence in which the boundaries are selected is not important.
- The boundaries do not need to belong to a region that is closed and manifold. In other words, planar surfaces and “open” boundaries are allowed.
- The boundaries must intersect one another, otherwise nothing will happen.
- The quality of the original boundary tessellation is not critical.
- Important: The process is not reversible, so you should save the simulation first before proceeding.
- It is recommended that the surface remesher be used after a boundary intersection to improve the quality of the surface.

Splitting Boundaries

It is useful to split boundaries if the original mesh contains too few of them for modeling or analysis purposes.

Six different ways are provided to split boundaries into smaller pieces:

- splitting non-contiguous boundaries;
- splitting boundaries by angle;
- splitting by function;
- splitting by patches;
- splitting by feature curves; and
- splitting interactively

In all cases, the properties of the new boundaries will be the same as the original boundary.

Contents:

Splitting Non-Contiguous Boundaries
Splitting by Angle
Splitting by Function
Splitting by Patch
Splitting by Feature Curves
Splitting Interactively

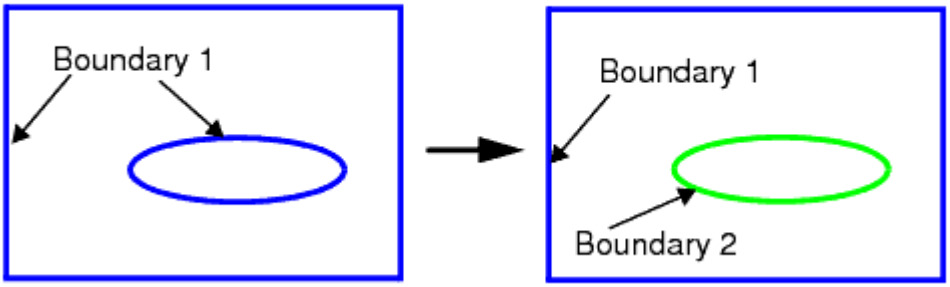
See Also:

Preparing Boundaries for the Generalized Cylinder Mesher

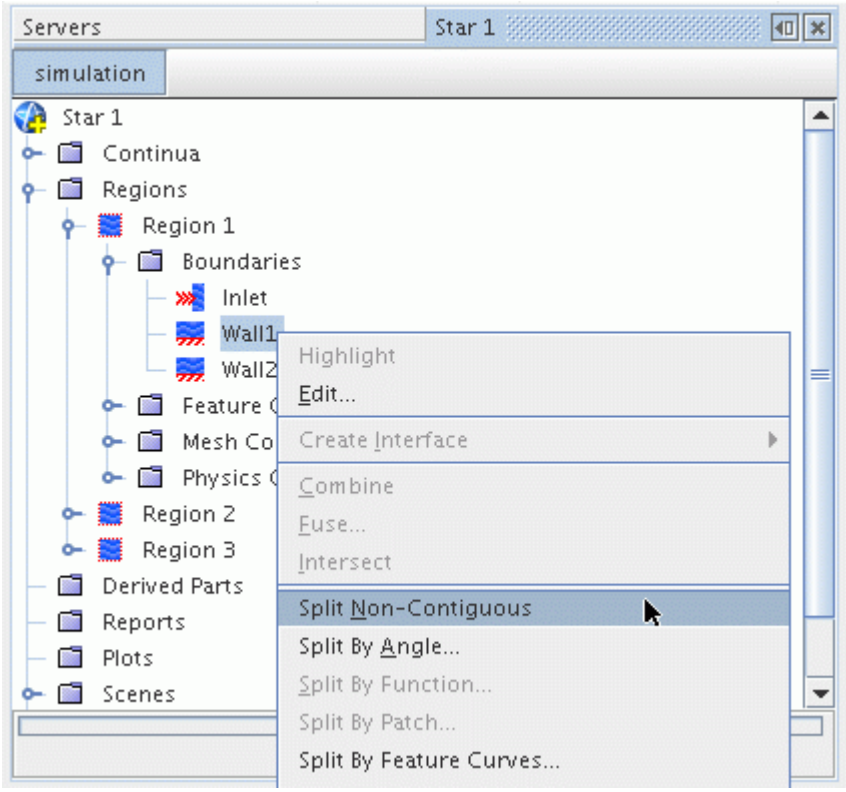
Splitting Non-Contiguous Boundaries

This function will split a surface or volume mesh boundary into its non-contiguous pieces.

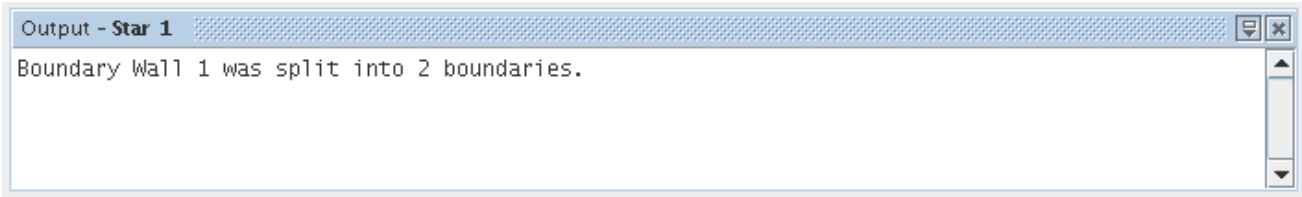
The splitting of a non-contiguous boundary is shown schematically below. If the boundary is contiguous to begin with, no split will occur.



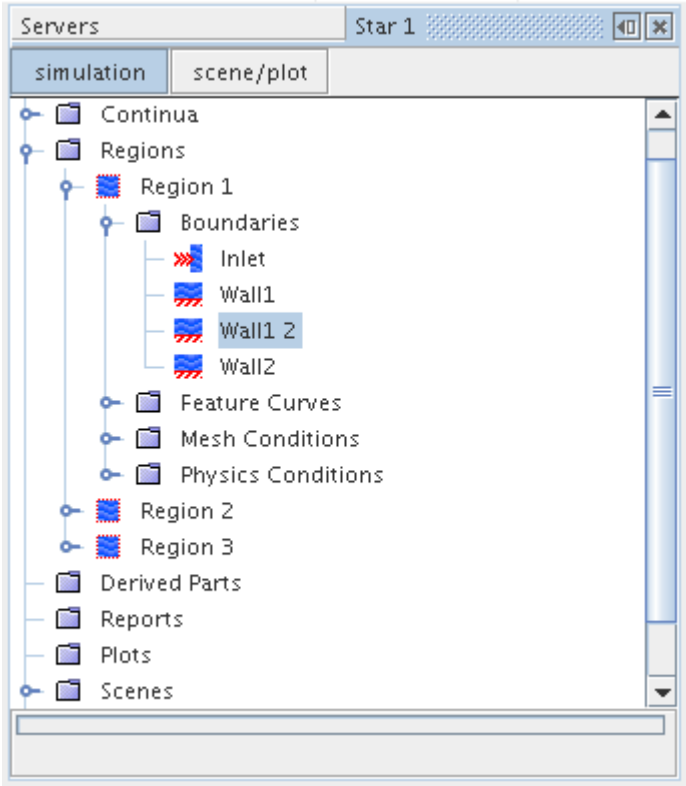
To split non-contiguous boundaries, open the **Regions** node and open the region of interest. Then right-click on the boundary or boundaries of interest from the **Boundaries** node. The selected boundaries may be from different regions. Select **Split Non-Contiguous** from the pop-up menu.



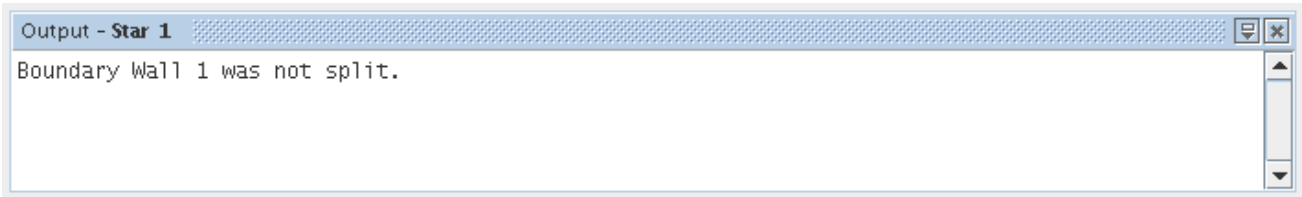
The splitting process starts and a progress bar is displayed at the bottom of the simulation tree. Once the process completes successfully, a message will displayed in the *Output* window.



The simulation will be updated to reflect the new boundaries that were created. Their names will be derived from the names of the selected boundaries with an identifying digit appended.



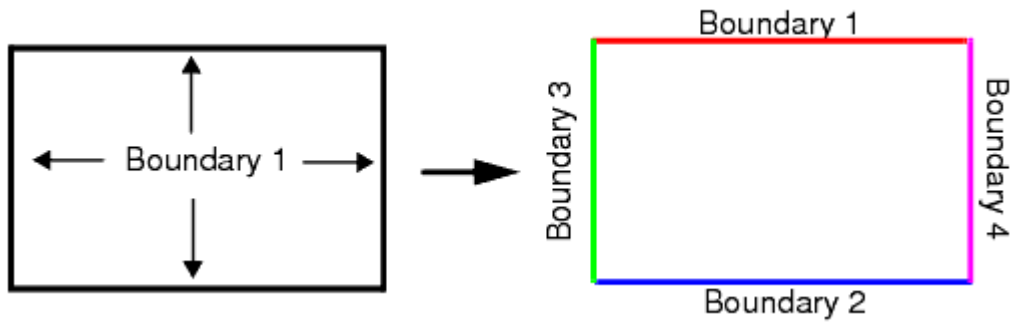
If any of the boundaries in the selection are already contiguous, no changes will be made to the boundaries and a message to that effect will be displayed in the *Output* window.



Splitting by Angle

This function enables surface or volume mesh boundaries to be split according to the included angle between faces.

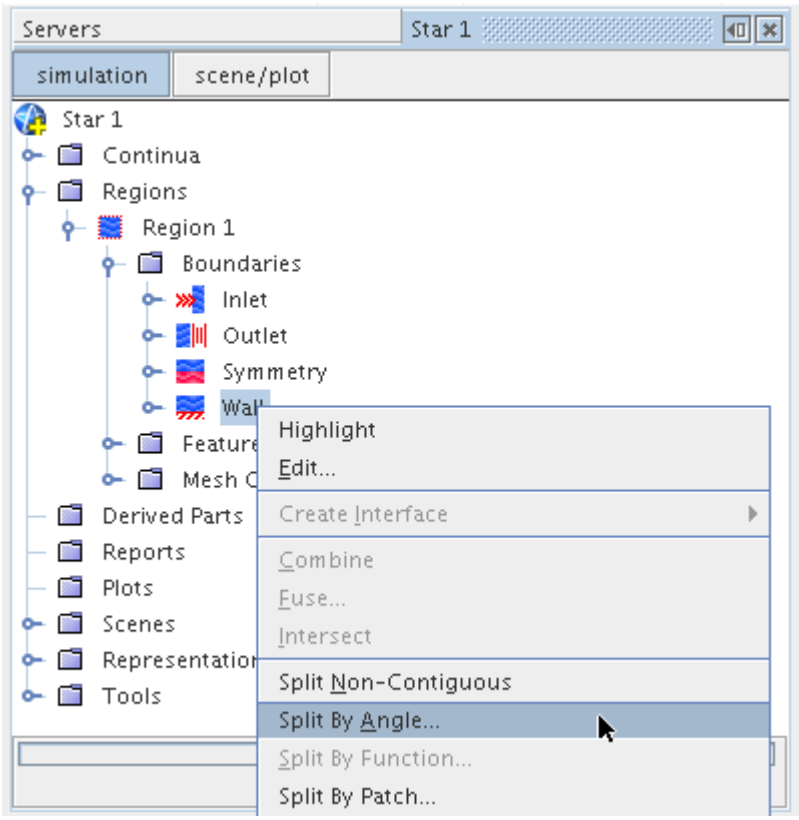
The operation of splitting by angle is illustrated schematically below.



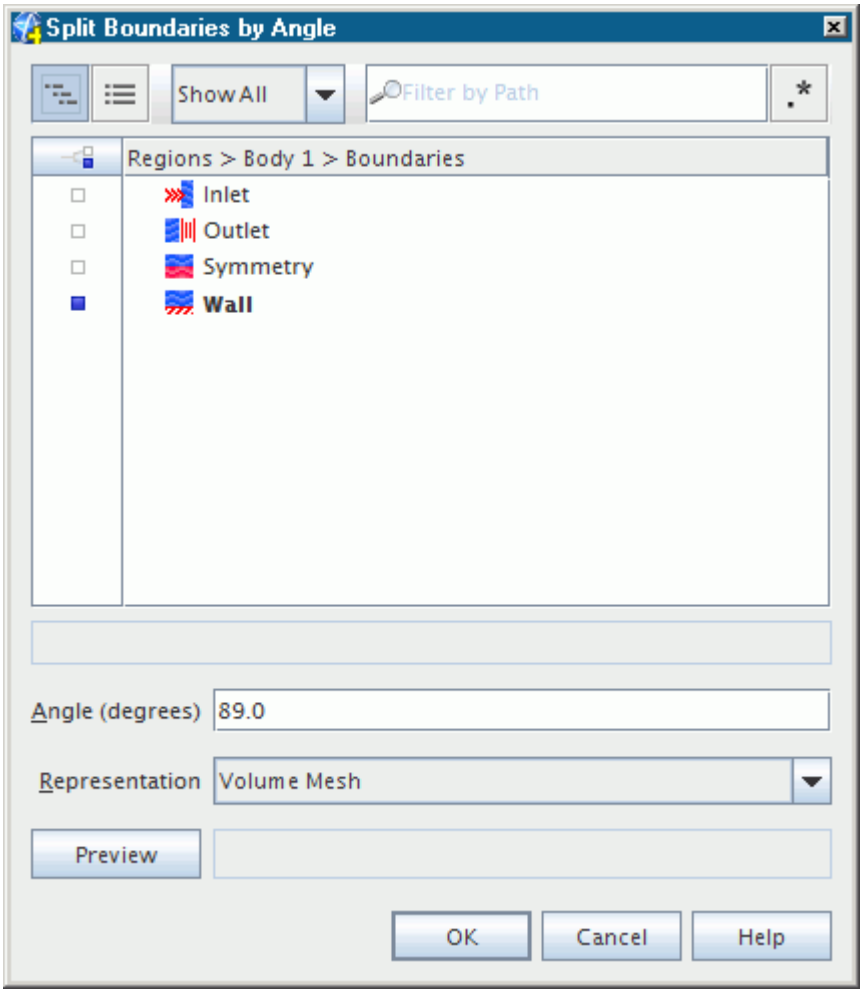
This would be useful, for example, if four walls of a box had been created as a single boundary, but you would prefer to treat them separately to support different conditions. The separation will only occur if there is a contiguous surface that can be demarcated by the specified angle criterion.

This function will also automatically execute the splitting non-contiguous boundary action, which occurs even if no splitting occurs based on the angle criterion.

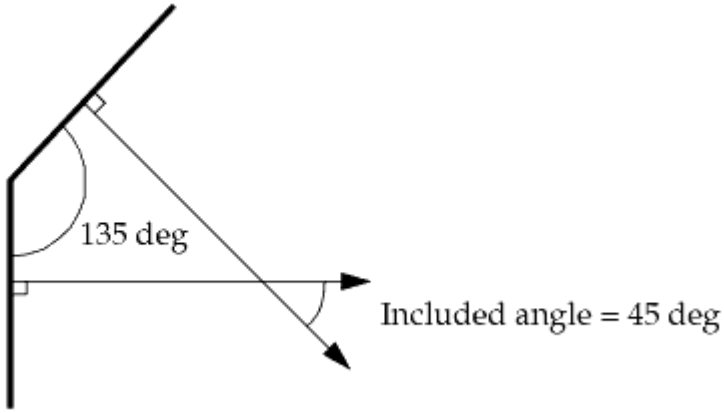
To split boundaries by angle, open the **Regions** node and open the region of interest. Then right-click on the boundary of interest from the **Boundaries** node. Select **Split By Angle...** from the pop-up menu.



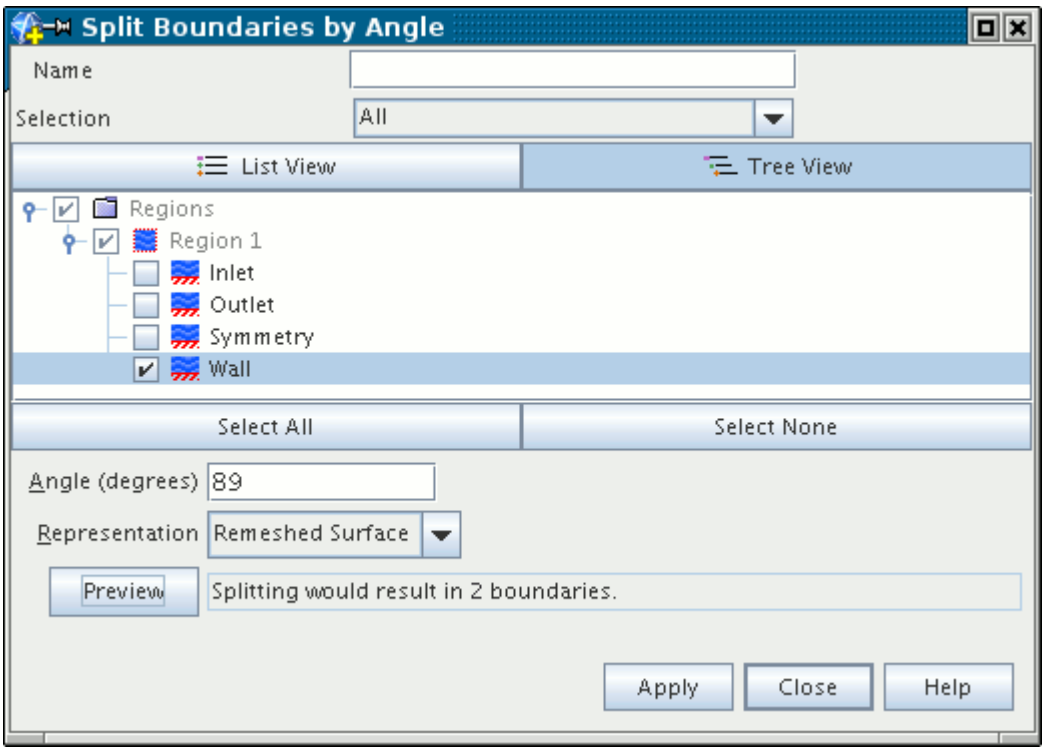
This opens the *Split Boundaries by Angle* dialog:



The boundaries that had been selected in the simulation tree are selected, and the **Angle** defaults to 89 degrees. When the angle between the faces is acute (less than ninety degrees) then an **Angle** value of less than the internal angle should be specified in order to split the boundary. When the angle between the faces is obtuse (more than ninety degrees) then the specified **Angle** value should be less than the included angle between the boundary normals. For example, to split a boundary which has faces at 135 degrees to one another, the included angle is 45 degrees, so a value of 44 degrees or less needs to be used to split the boundary. This is illustrated below:

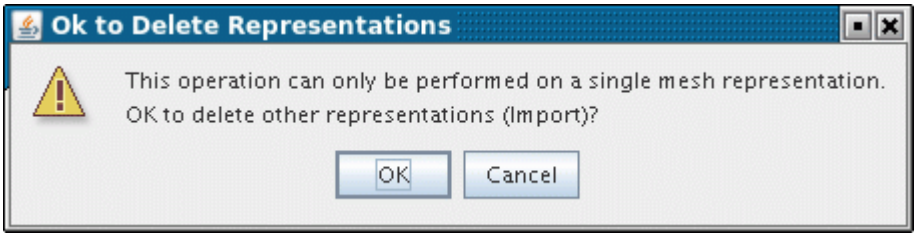


If more than one representation exists then the surface or volume to be operated upon should be selected using the **Representation** menu. Once a representation is chosen, the **Preview** button can be used to determine how many boundaries would be created based on the supplied angle. For example:

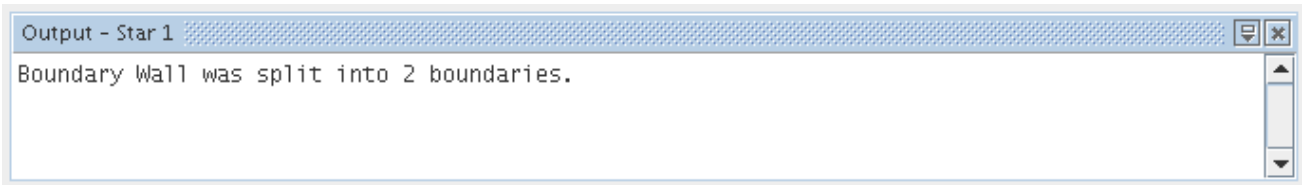


Here, the default value of 89 degrees would result in the **Wall** boundary in **Region 1** being split into 2 boundaries (the revised original plus a new one). This is a useful tool for determining the effect of the angle value in cases where there is a large variance in angles between boundaries.

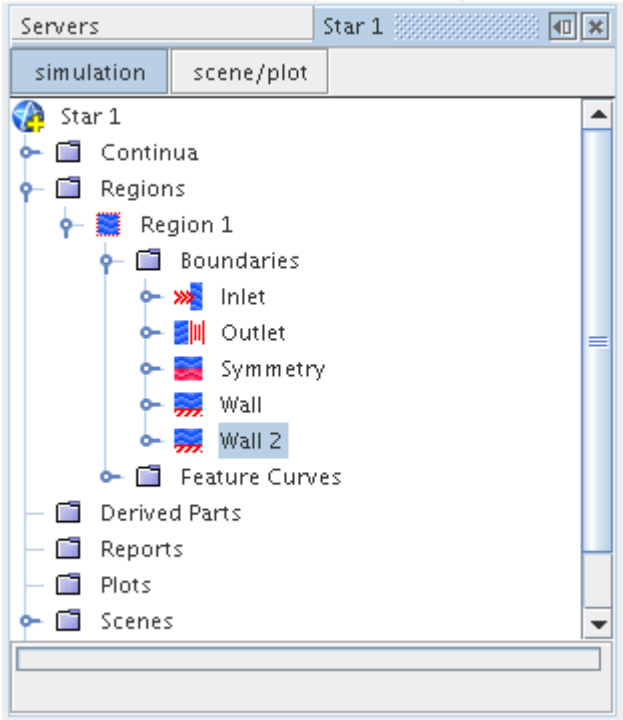
Pressing the **Apply** button will start the boundary splitting process. Should other representations exist for the simulation then they will have to be deleted first since the boundary correspondance between them will no longer be applicable. In this case, the *Ok to Delete Representations* dialog will appear:



On pressing the **OK** button, STAR-CCM+ will proceed to split the boundaries for the selected representation. A progress bar will be displayed at the bottom of the simulation tree. Once the process is complete messages in the *Output* window will indicate the new boundaries that were created:



The new boundaries will also appear on the simulation tree. The name(s) of the new boundaries are taken from the source boundary, with an identifying digit appended:



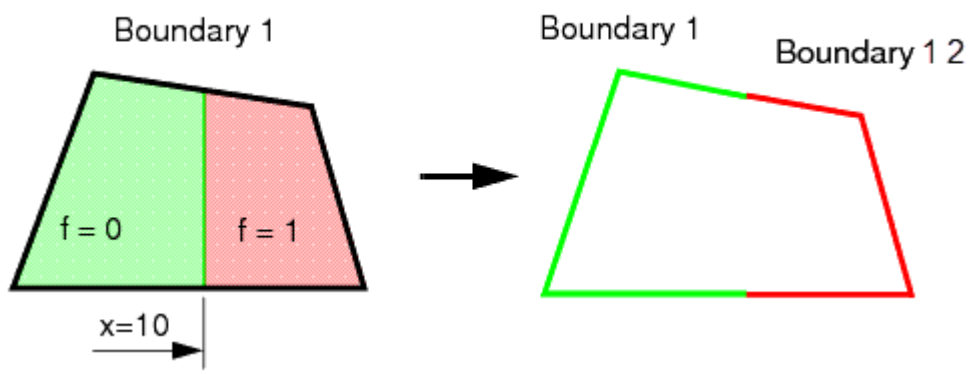
If the boundary could not be split based on the angle that you supplied, then a message to that effect will be displayed in the *Output* window indicating as such.

Splitting by Function

This operation splits volume mesh boundaries according to the rounded integral value of a specified field function.

Note: The boundary with the most faces keeps the name of the original boundary. The naming conventions for new boundaries follow the ascending order of the number of faces in the boundary. For example, if the field function creates two new boundaries from Boundary 1, the new boundary with the least faces is Boundary 1 2, and the most faces is Boundary 1 3.

The operation of splitting by function is illustrated schematically below.



In this example, the field function f has the value 0 or 1, so that two boundary are created. In this case, f might be defined as follows:

($$Centroid[0] < 10) ? 0 : 1

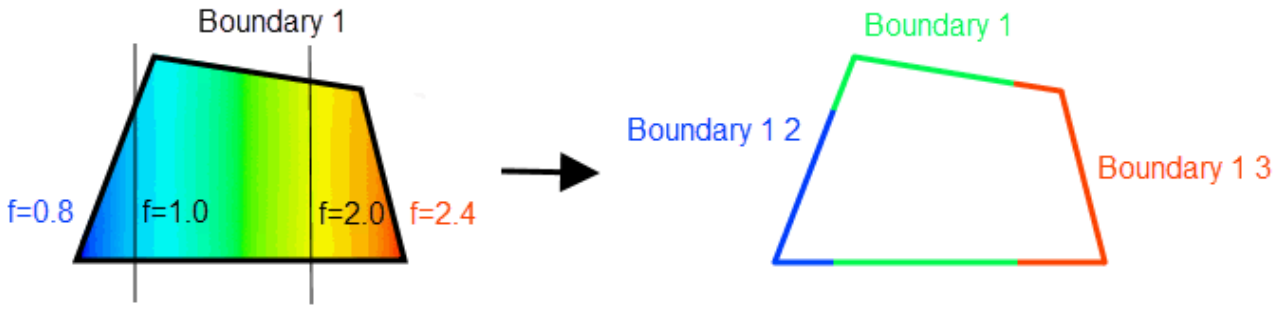
or even more simply

$$Centroid[0] < 10

both of which cause a step from $f = 0$ to $f = 1$ at the global x coordinate of 10.

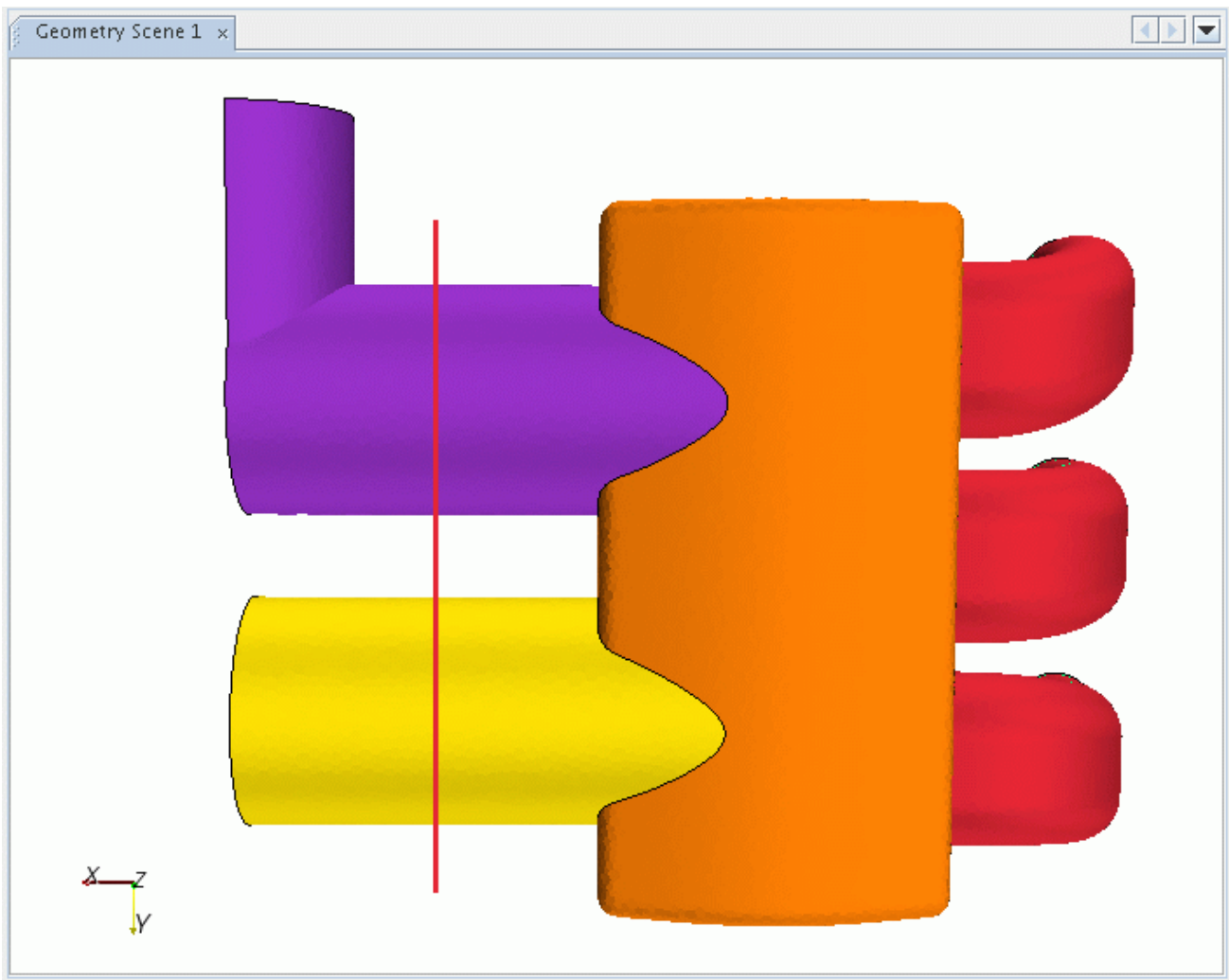
In the above example, the field function has two discrete integer values. If it consists of a range of real numbers, say $0.8 < f < 2.4$, the boundary is split into three parts containing:

- cells in which $0.8 < f < 1.0$;
- cells in which $1.0 \leq f < 2.0$;
- cells in which $2.0 \leq f < 2.4$;

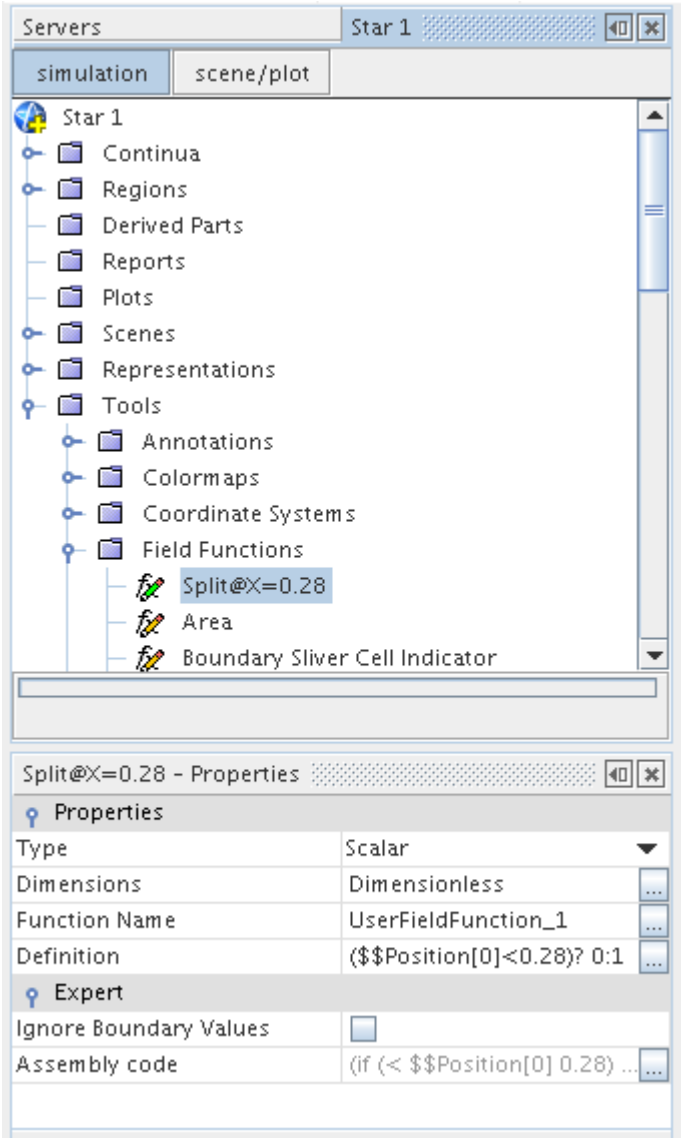


STAR-CCM+ attempts to create as many additional boundaries as there are rounded integer values given by the field function in the original boundary. In other words, if $0.8 < f < 2.4$, two additional boundaries are created, since f varies as 1 and 2 . If $0 < f < 100$, 99 additional boundaries are created. To prevent the creation of an unreasonable number of new boundaries, ensure that the field function has a restricted range.

Suppose we wish to split two purple and yellow boundaries along the X direction, as shown in the screenshot below.

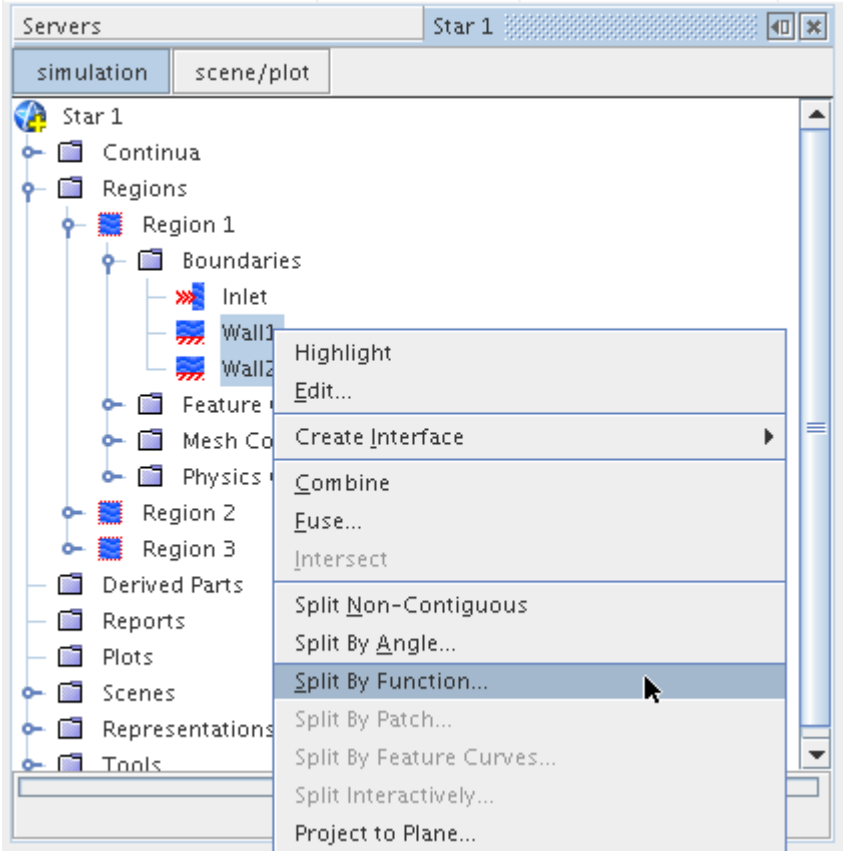


Use any technique to find the X location of the split - we will use a value of `0.28` for this case. Create a new user field function with the name **Split@X=0.28** using the **Tools** > **Field Functions** node and set the *Definition* to `($$Position[0]<0.28)? 0:1`.

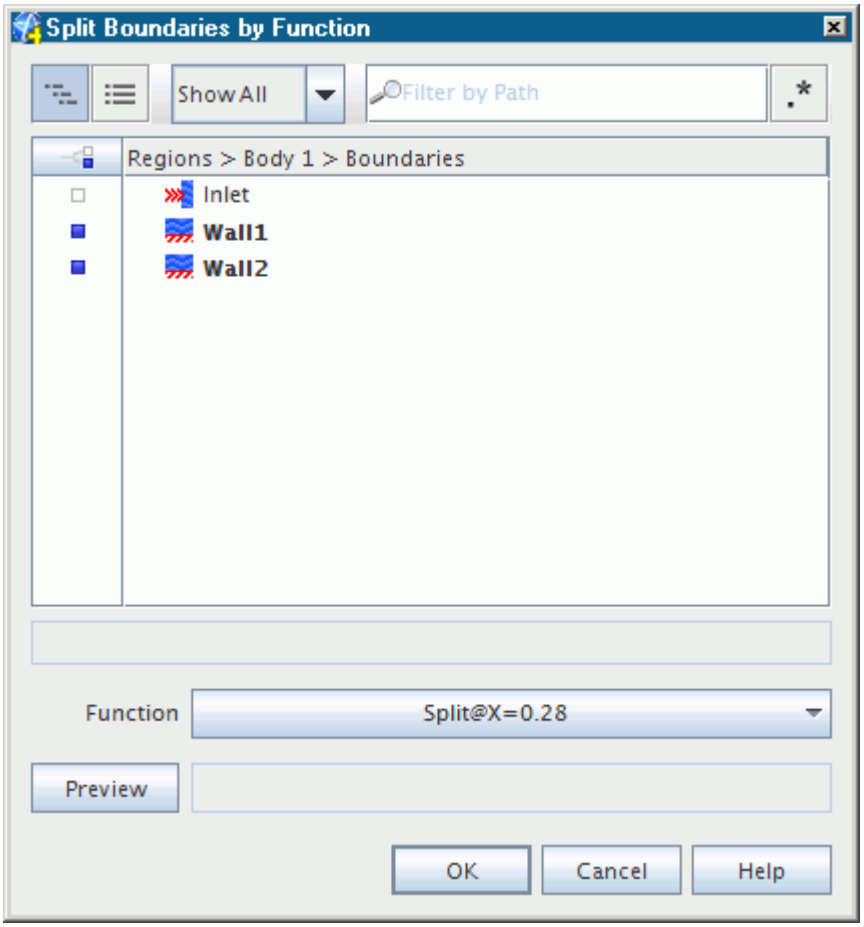


Initialize the solution to ensure that the user-defined field function will hold valid values. If errors are reported in the *Output* window then these should be corrected.

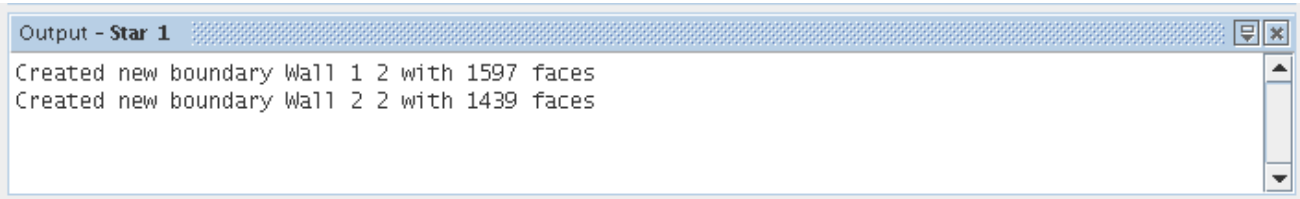
Now open the **Regions** node and open the region of interest. Right-click on the boundary of interest from the **Boundaries** node and select **Split By Function...** from the pop-up menu.



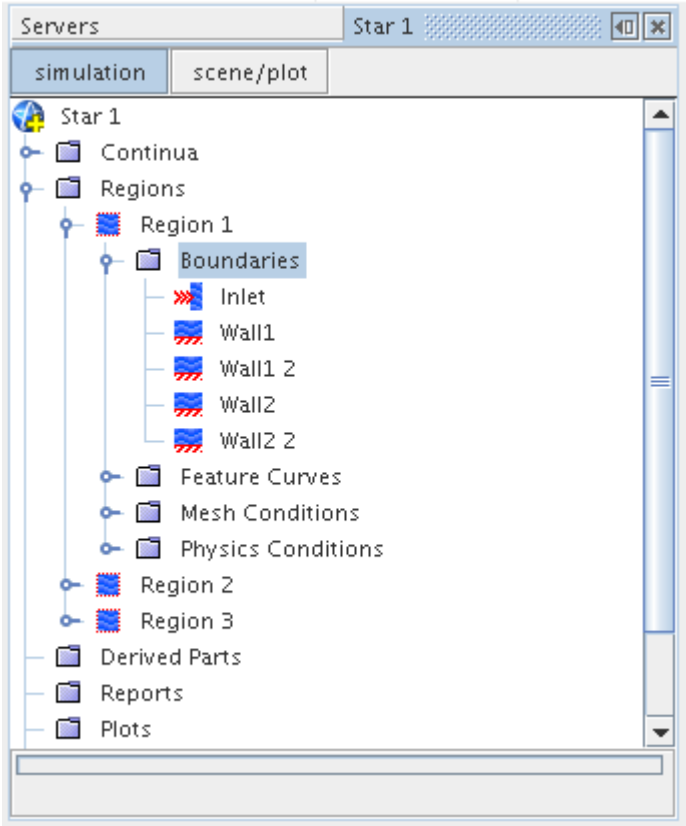
This opens the *Split Boundaries by Function* dialog with the boundaries selected in the simulation tree marked with a ticked checkbox. Pull down the Function drop-down list and select the user-defined field function that you created earlier:



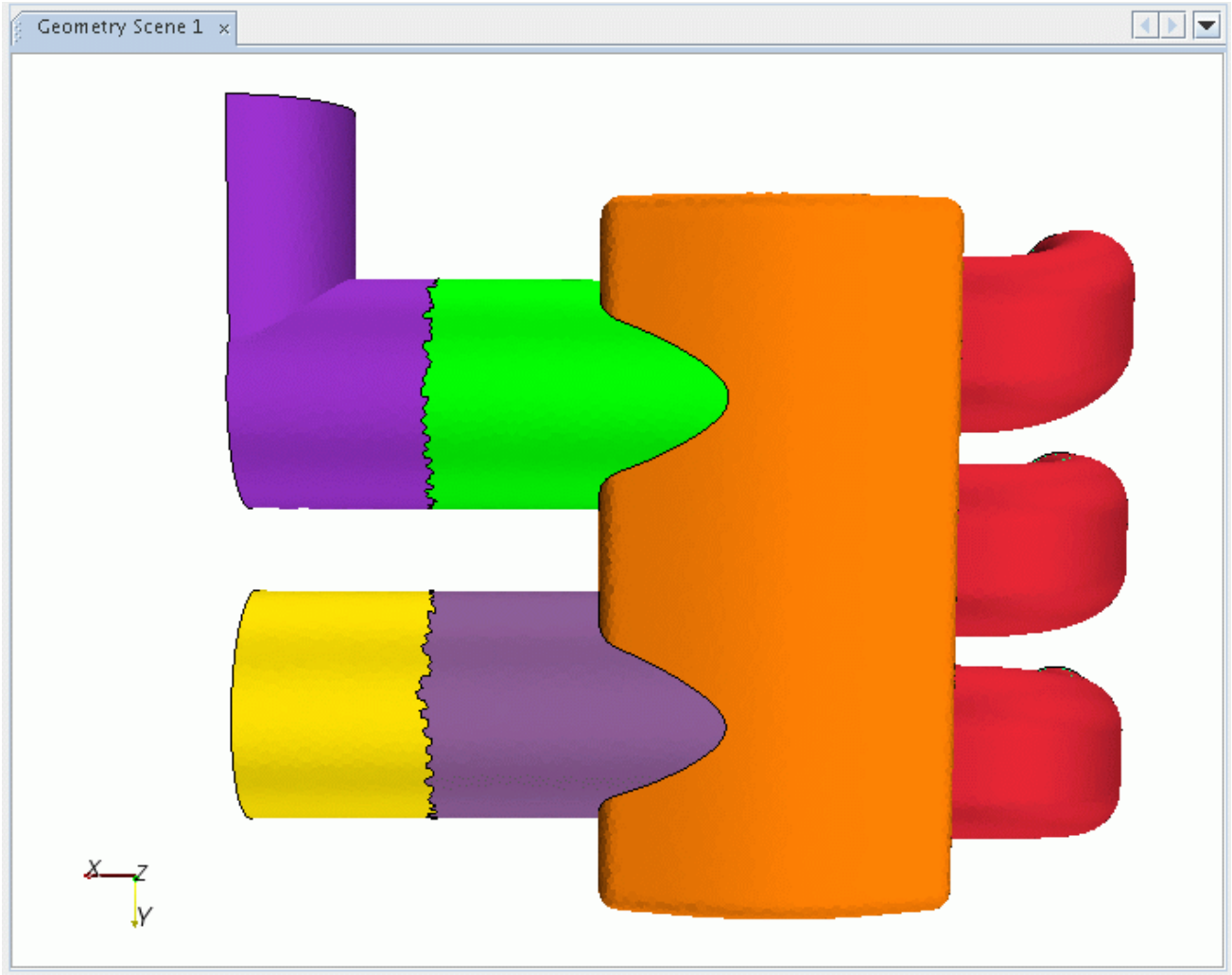
Click the **Apply** button to start the splitting function. A progress bar will be displayed at the bottom of the simulation tree. Once the process is complete a message will be displayed in the *Output* window indicating the names of the new boundaries and the number of faces in each.



The simulation tree will be updated with the new boundaries. The names of the new boundaries are taken from the original boundaries with an identifier appended.



The screenshot below shows the new split in the boundaries.



Note that the splitting process does not divide or cut boundary faces into two. The process simply divides the boundary up based on the existing face structure and retains the same total number of faces before and after.

Splitting by Patch

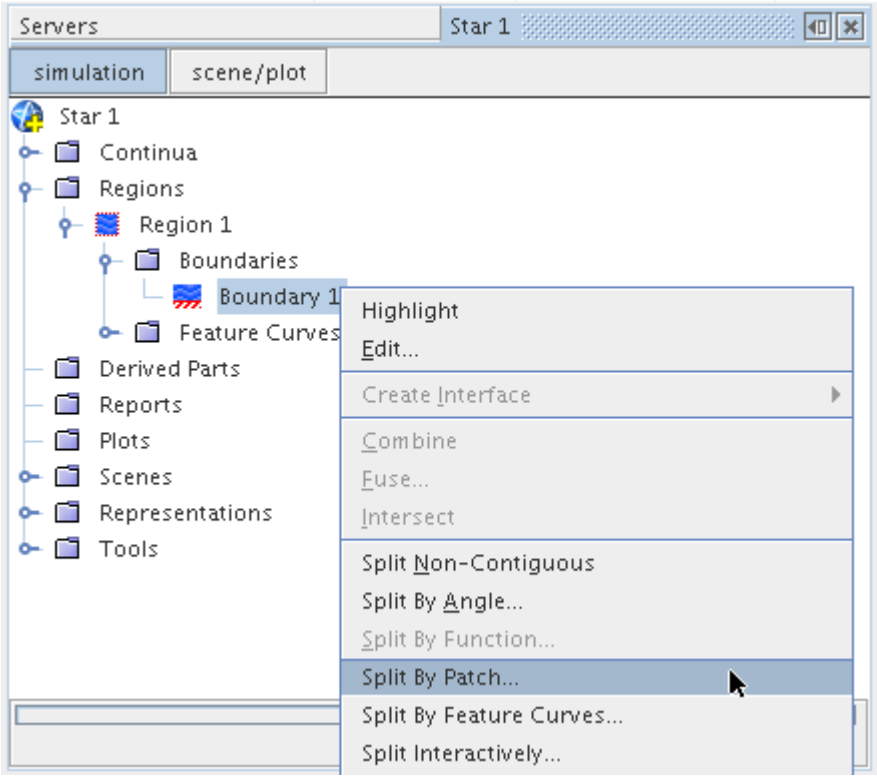
The split by patch function allows you to split a boundary based on the underlying patch definition of the imported surface geometry.

A patch represents a cell table, group, family or some other similar entity used in the original surface pre-processing software to differentiate one surface from another. This information is read into STAR-CCM+ as part of the surface import process and is maintained as part of the import representation. All surface file formats other than STL support patch definitions.

Patches are useful in STAR-CCM+ as they allow a quick and easy method of reconstructing the boundaries to match the original surface definitions should they not initially be imported this way. A boundary can contain multiple patches but a patch cannot be shared among two or more boundaries. The patch definitions can change depending on how the boundaries are manipulated. For example, if a boundary is split up (based on an angle or feature curves etc), then each new boundary will have its own patch definition.

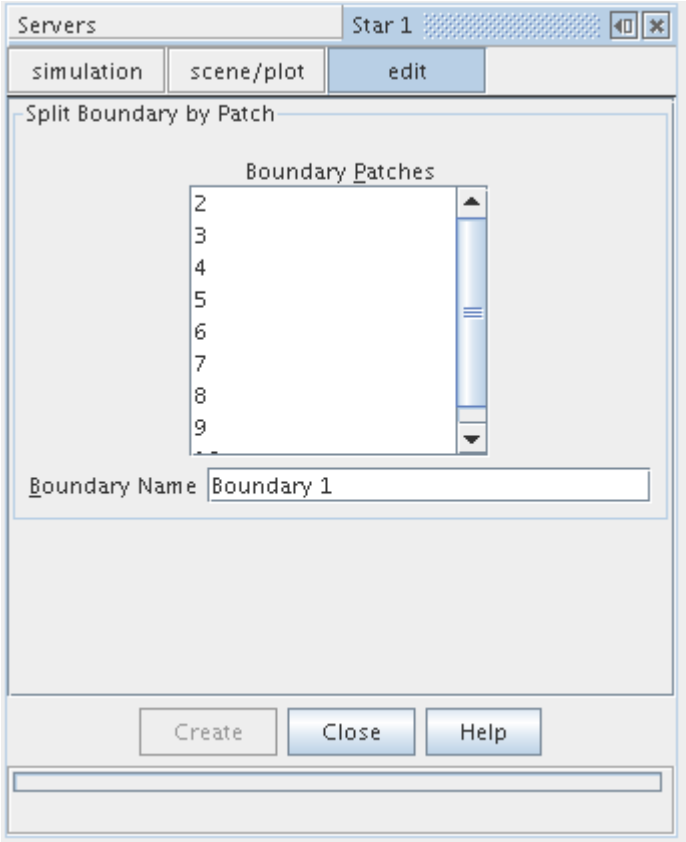
To split boundaries by patch:

1. Right-click on the boundary name or names (using multiple selection) in the **Boundaries** node and select **Split By Patch...**



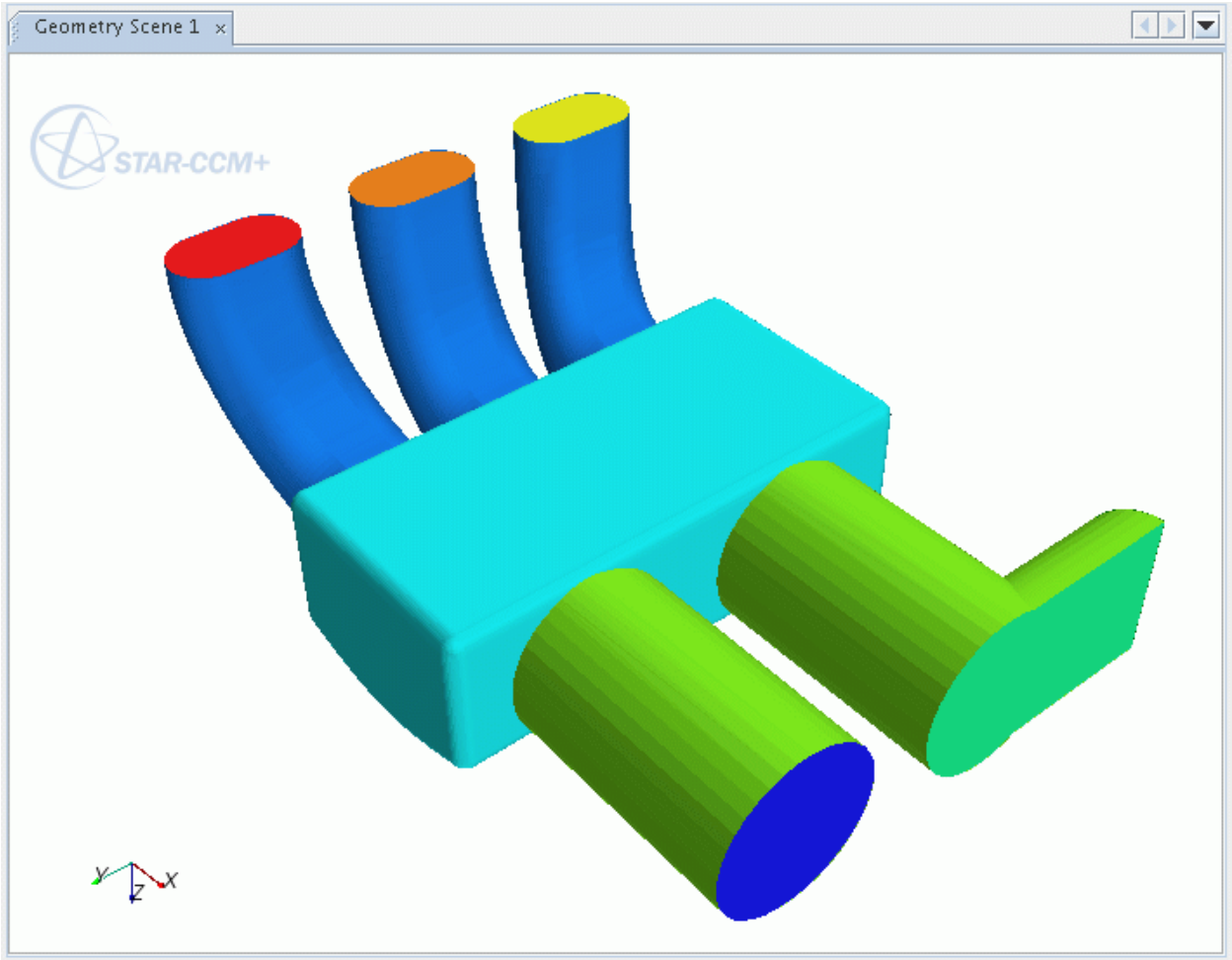
Note: You must have a scene open to use this feature.

The *Split Boundary by Patch* window appears in the *edit* tab in the explorer pane.



Here, the patches that are contained in the import surface are listed numerically. The surfaces on the geometry in the scene window will also be color-coded to represent each patch number.

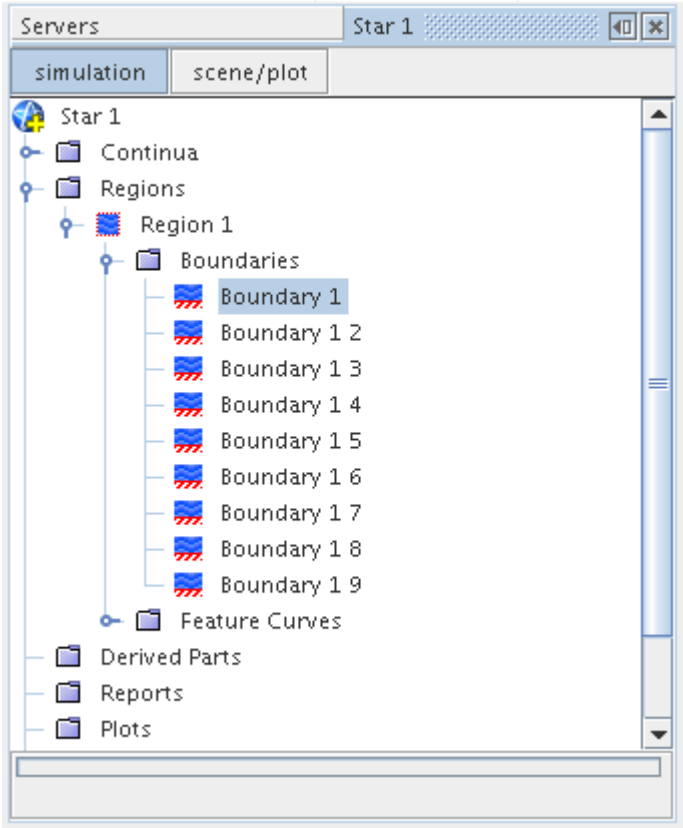
2. Select the patches from the list box or directly using the mouse in the scene display. To select multiple patches, press the `<Ctrl>` key, or use the zone selection tool from the Selection toolbar. Selecting a patch will result in it being highlighted in the scene window.



3. Enter the new **Boundary Name**.
4. Click **Create**. The relevant patch or patches will be removed from the dialog and scene so that only the remaining patches are shown.
5. Repeat the process as many times as required until only one patch is left.
6. Click **Close** when done.

Note: When only one patch remains, the **Create** button is disabled. The source name will be used for the remaining boundary.

The new boundaries appear in the simulation tree. In the following example, the boundaries were created based on 6 patches.



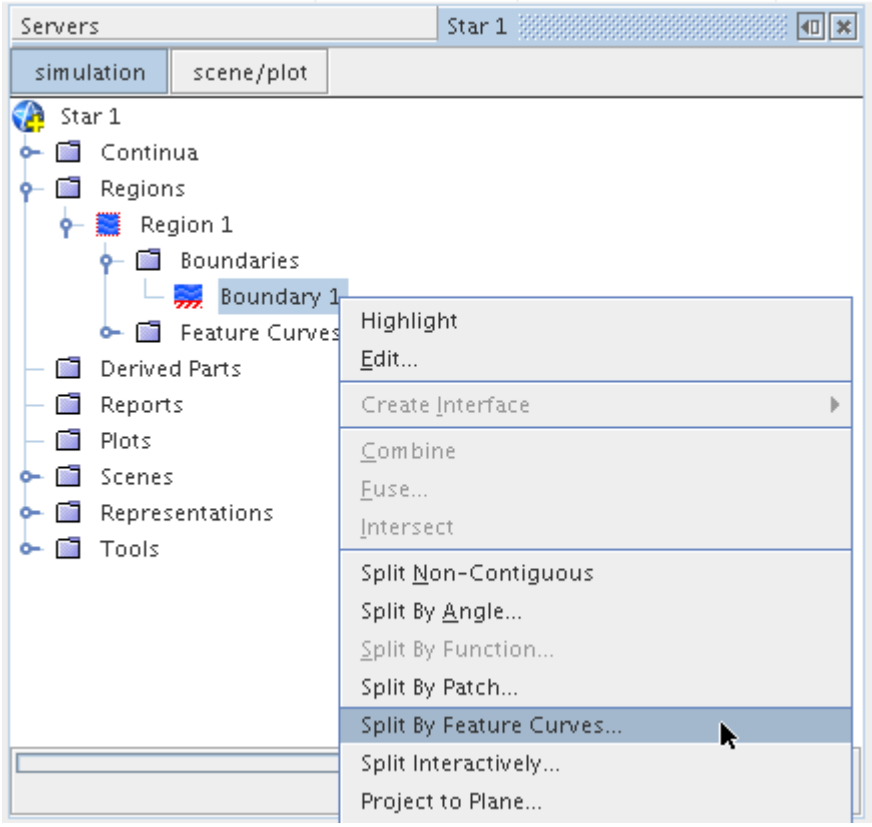
Splitting by Feature Curves

The split by feature curve function allows a boundary to be split based on feature curves that belong to the same surface mesh region.

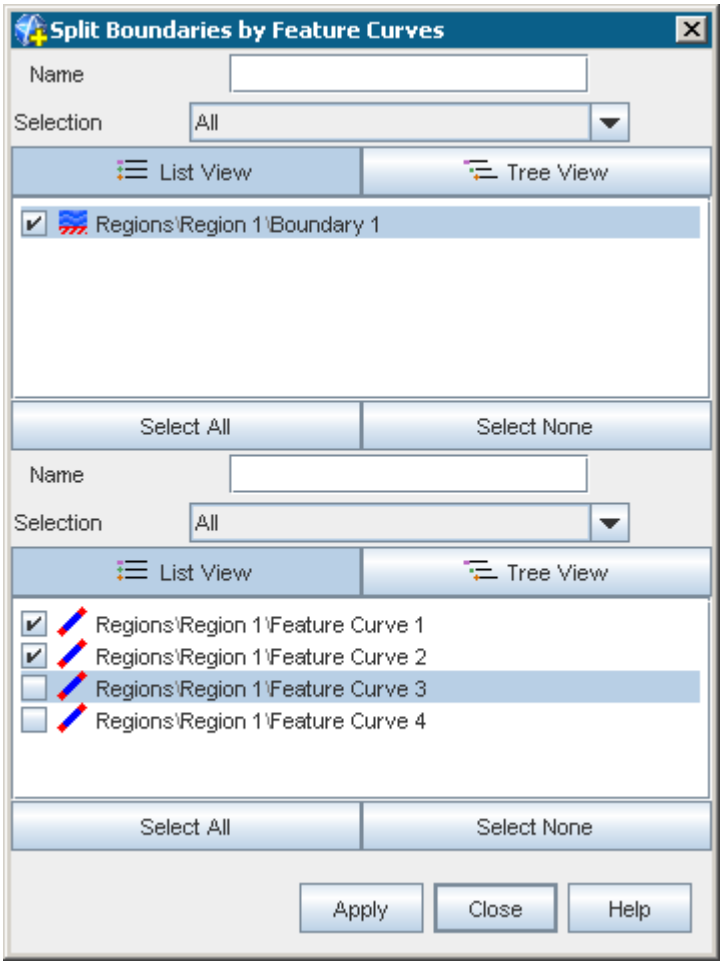
In order for a boundary to be successfully split, the feature curve must completely separate the boundary into two or more pieces.

Feature curves are usually imported as part of the surface definition using any of the allowable surface file formats other than STL and PATRAN. Being able to split a boundary using feature curves is extremely useful as feature curves can be added to any part of the surface and allow you to form any arbitrary shape that you want. Once the boundary has been split, the features curves can be deleted if desired or left in as part of the definition if they are important for maintaining a surface feature or detail.

To split boundaries by feature curves, open the **Regions** node and then the region of interest. Right-click on the boundary name or names (using multiple selection) in the **Boundaries** node and then select **Split By Feature Curves...** from the pop-up menu.

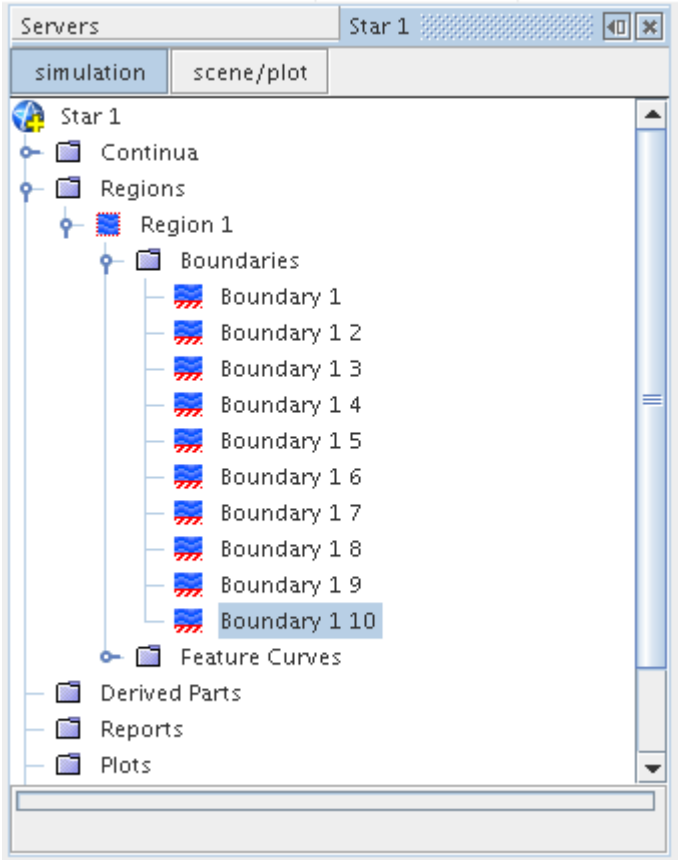


The *Split Boundary by Feature Curves* dialog will appear, as shown below:



The boundary or boundaries originally chosen in the object tree will appear selected. Use the checkboxes to select the feature curves which will be used for splitting and then press **Apply** to start the splitting process. The progress will be displayed at the bottom of the simulation tree and the final number of split boundaries will be reported back in the *Output* window.

Once the process is complete, the simulation tree will be updated with the new boundaries.



The names of the new boundaries are taken from the original boundary or boundaries with an identifier appended. Note that the feature curves are not affected in any way by this process.

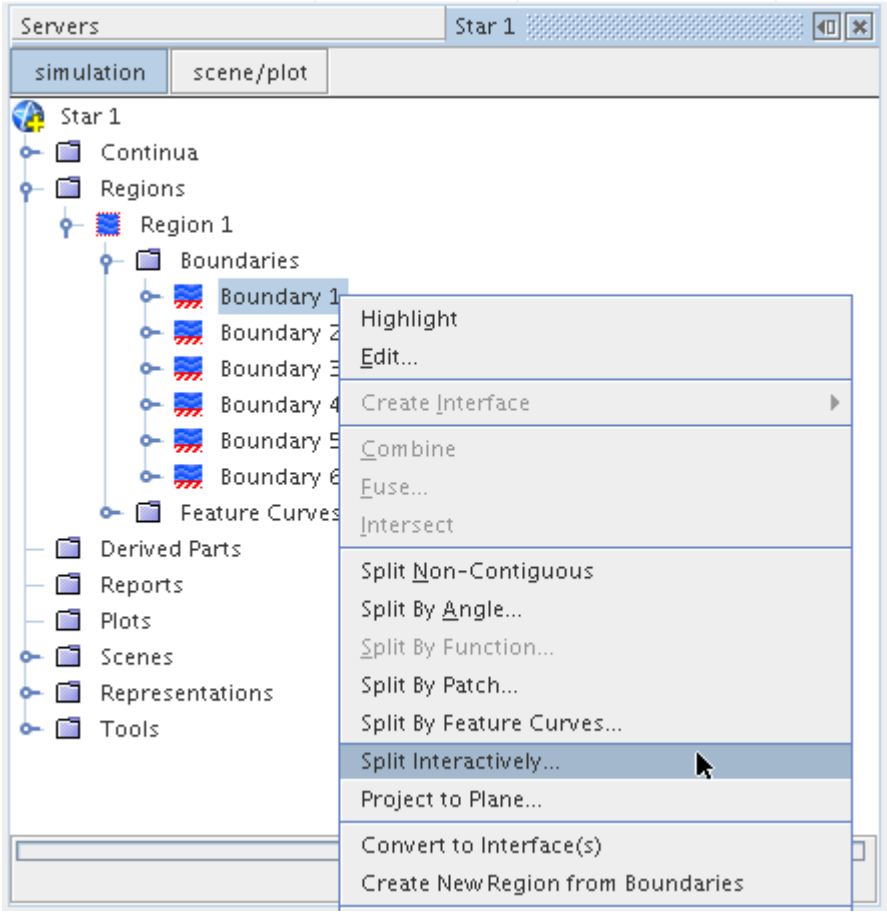
Splitting Interactively

The split interactively function allows you to split a boundary based on the triangulation of a surface mesh.

Note: This feature operates on the representation of a surface mesh. Therefore it is required that you generate the relevant surface mesh using the region based meshing approach.

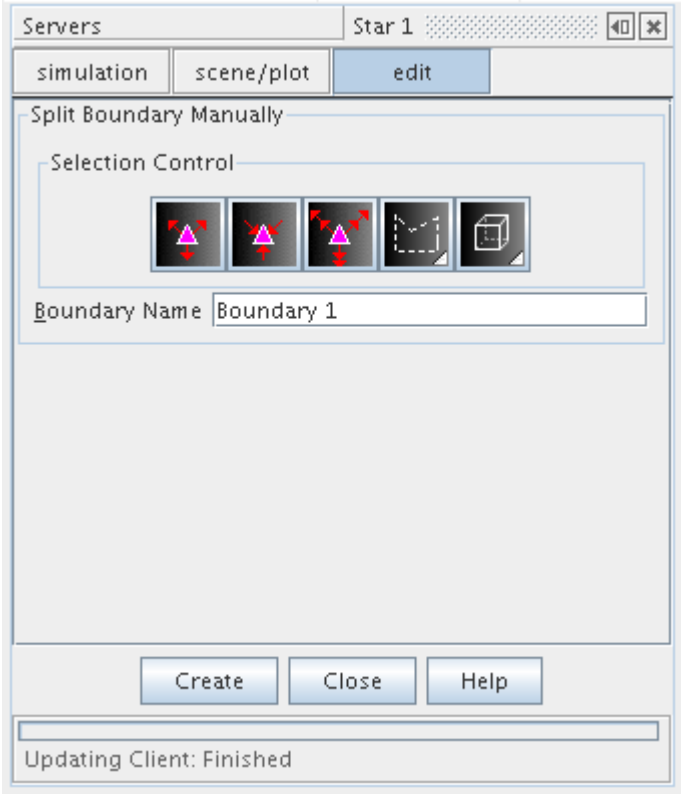
When you split a boundary interactively, anywhere from one face to all the faces belonging to the boundary can be selected to belong to the new boundary. Only one boundary can be selected to be split at a time.

To split a boundary interactively, open the **Regions** node and then the region of interest. Right-click on the boundary name in the **Boundaries** node and then select **Split Interactively...** from the pop-up menu.



If several surface representations are available, a *Select Surface Representation* window appears that allows you to select the representation you want to split. All other surface representations are cleared.

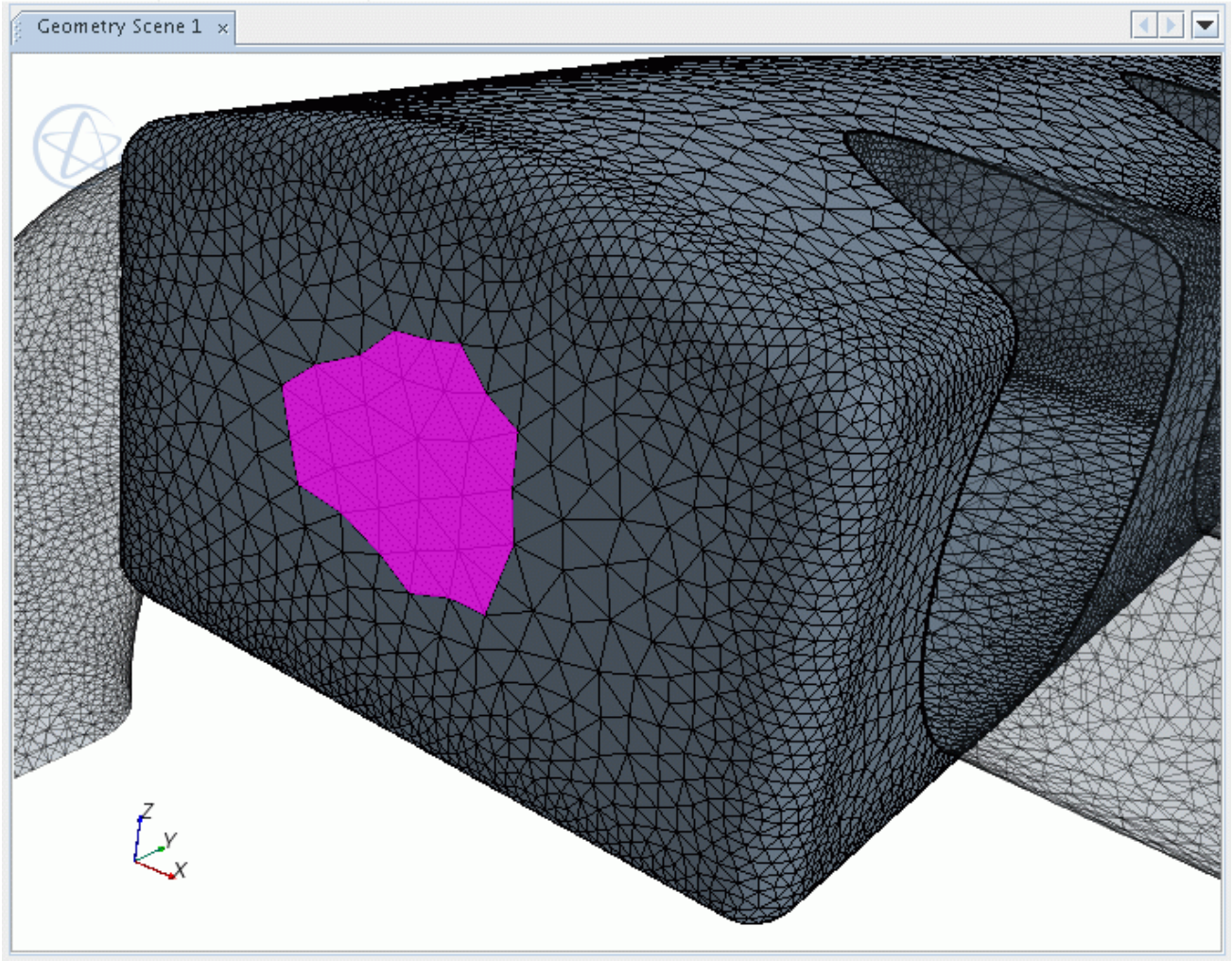
The *Split Boundary Manually* window appears in the *edit* tab:



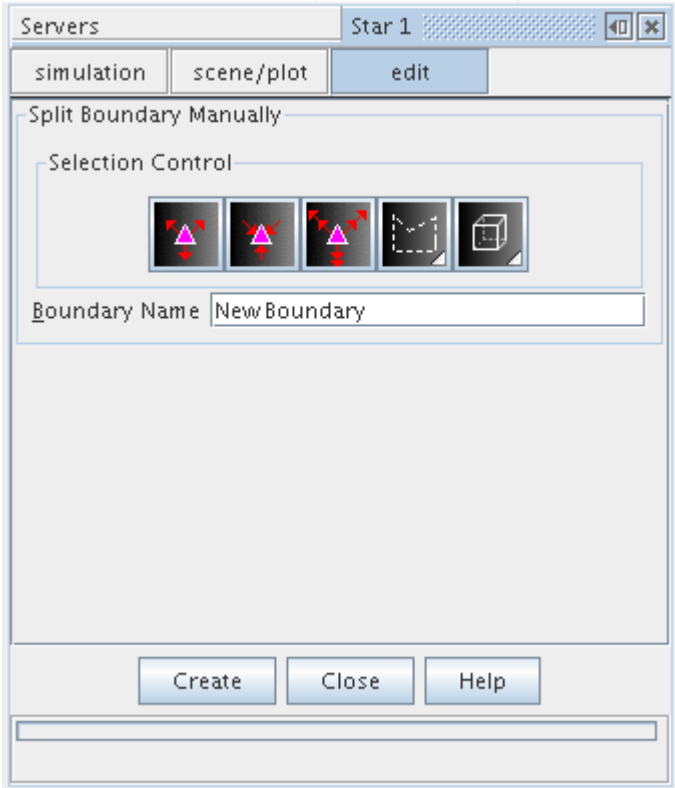
The scene also displays the model with all boundaries other than the one selected set to be transparent. By default, the mouse can be used to select faces in the displayed boundary, with multiple selections possible by pressing the Ctrl key while selecting additional faces. Double clicking on the displayed boundary selects all continuous faces constrained by feature curves or sharp edges. To unselect entities, you can click on them again (while pressing the Ctrl key) or click anywhere off the model.

The *Split Boundary Manually* tool provides access to five *Selection Control* buttons which can be used to further manipulate the face selections made by either growing from a face pre-selection or doing zone based selection/de-selections. These options are particularly useful when a large number of face selections need to be made which would otherwise be too time consuming to select individually. For more details on the buttons, refer to the selection control options for manual surface preparation, which work in a similar way.

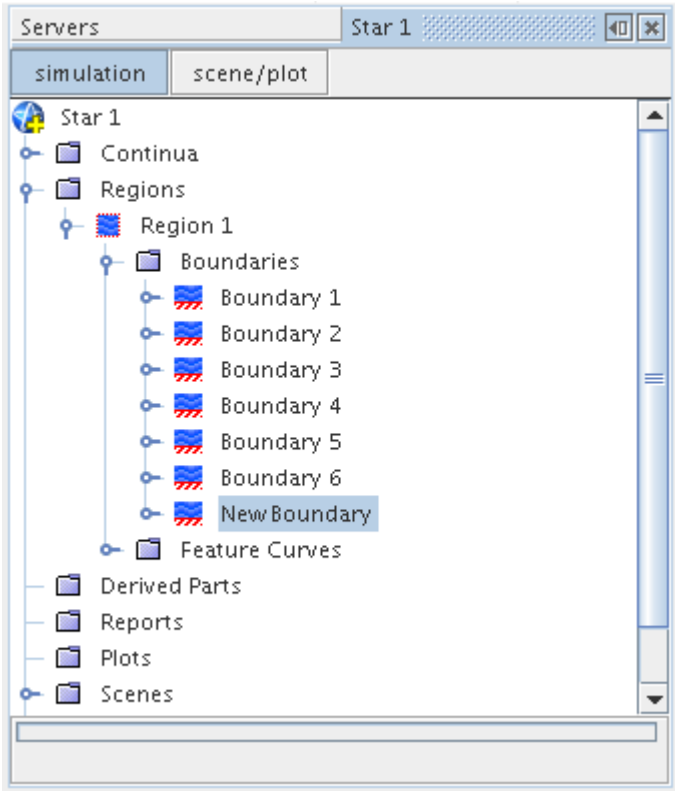
An example model with a part of a boundary selected is shown below:



When you have selected the face or faces of interest, enter in the new **Boundary Name** and then press the **Create** button to make a new boundary.



The new boundary name appears in the simulation tree.



If you do not specify a new name, then the original boundary name with a numerical increment is used instead. The newly created boundary is removed from the original displayed boundary.

The process can be repeated as many times as necessary for a given boundary and the **Close** button pressed when the operations are complete. If all faces for a given boundary are selected to become a new boundary then you will be left with an empty original boundary which should be then deleted.

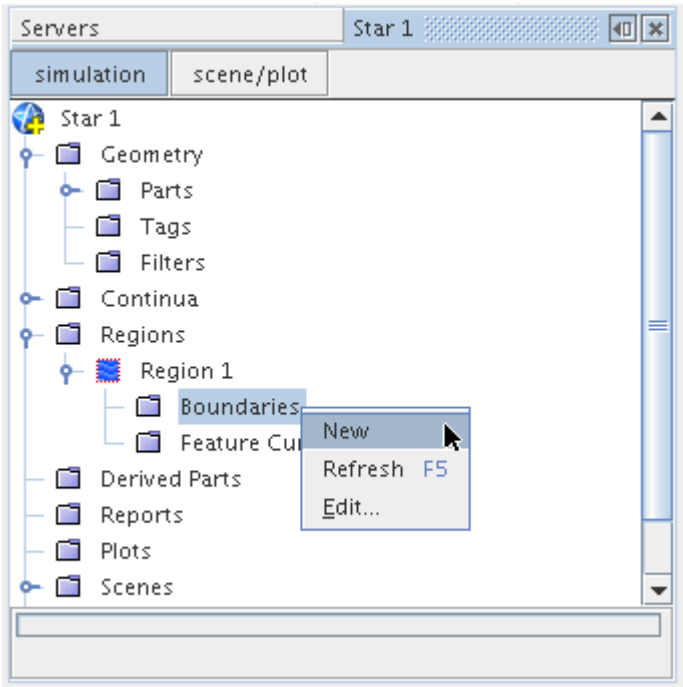
Note: As the split interactively function operates on the surface representation of a boundary, you need to rerun the volume mesher after splitting.

Creating New Boundaries

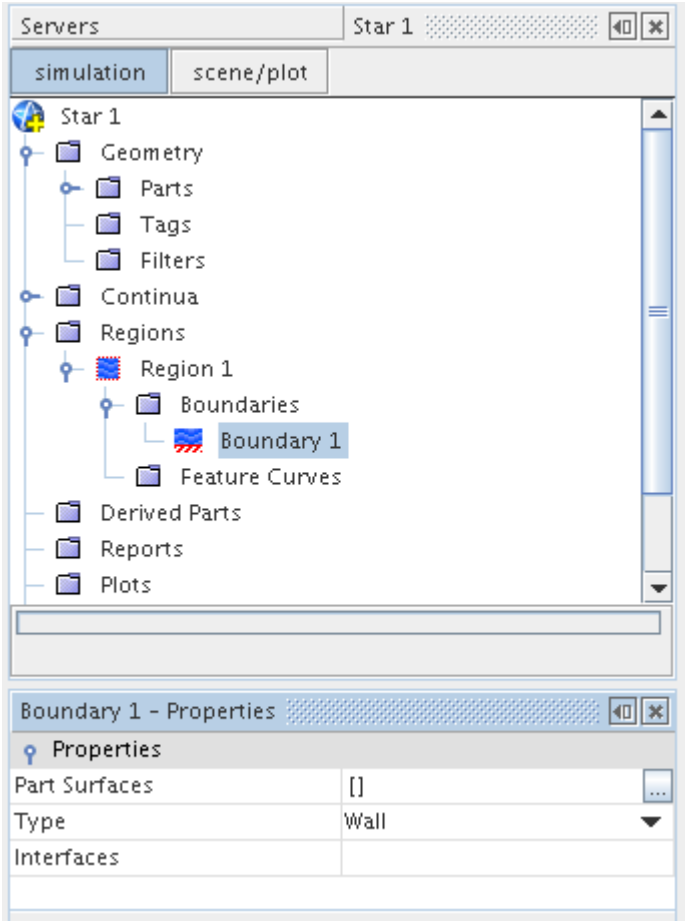
New empty boundaries can be created at any time so that part surfaces can be allocated to them as required in order to form regions for meshing purposes.

Note: Part surfaces cannot be allocated to a boundary until the part that the surface belongs to has been assigned to the parent region of the boundary.

To create a new boundary, right-click on the **Boundaries** manager node and select **New** from the pop-up menu:



The simulation tree will be updated with a new boundary node called **Boundary 1** by default:



By default, no part surfaces are assigned to this boundary but these can be manually selected. The default boundary type is also set to wall but this can be changed at anytime thereafter.

Projecting Boundaries

Boundaries belonging to a volume mesh can be projected in order to modify an existing boundary surface to conform to a given plane.

The following operations are possible:

- projecting boundaries to a plane (2D and 3D meshes); and
- projecting boundaries to an axis (2D meshes only)

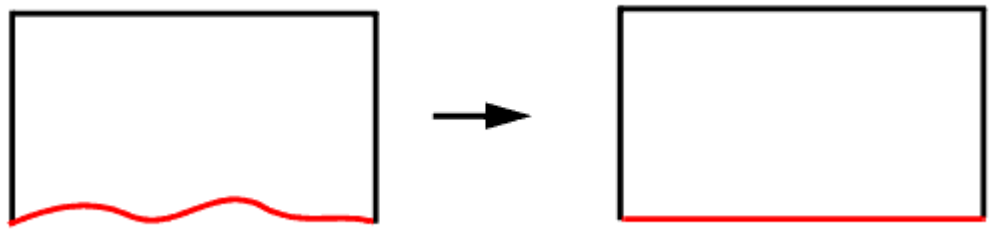
Contents:

Projecting Boundaries to a Plane
Projecting Boundaries to an Axis

Projecting Boundaries to a Plane

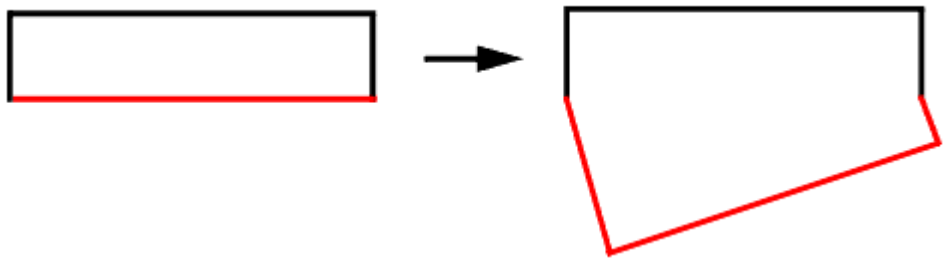
The project to plane function can be used to modify a non-planar boundary on a 3D or 2D volume mesh and make it planar.

This projection is shown schematically below:

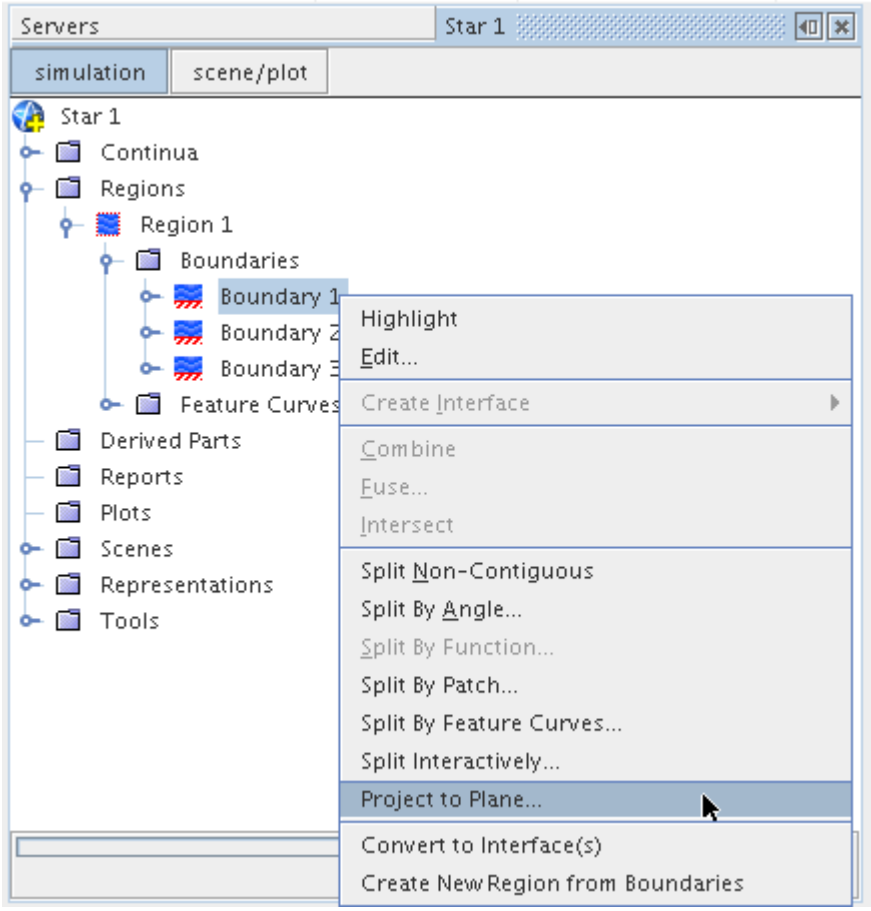


A typical application would be to make an undulating boundary more planar so that it could better be used for either the creation of an interface or for extracting a two-dimensional mesh from a three-dimensional mesh.

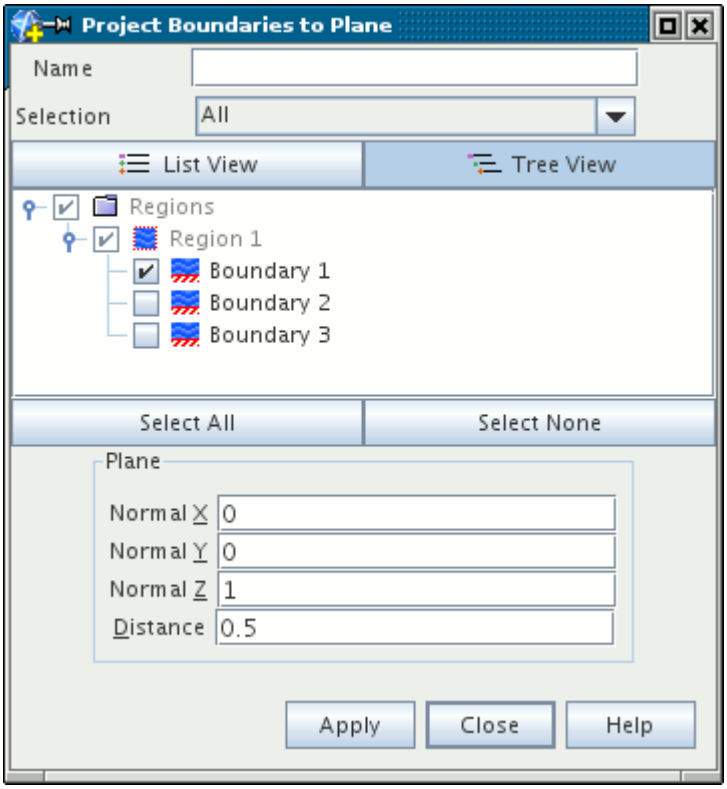
The project to plane option should be used with caution since it has the potential for adversely affecting the quality of the adjacent cells. In fact, it is possible to severely distort a boundary (and the adjacent cells) by specifying the wrong plane for projection as illustrated below:



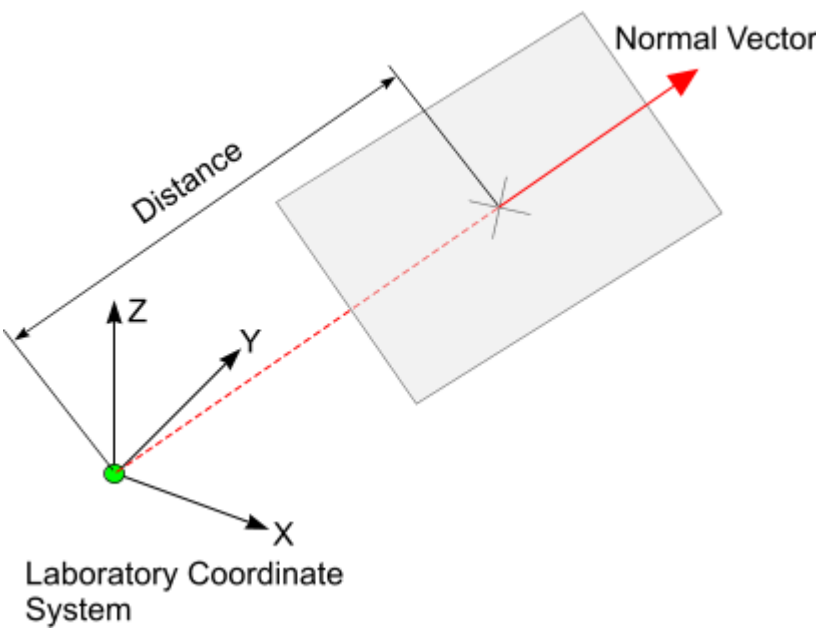
To project a boundary, open the region of interest in the **Regions** node. Then right-click on the boundary of interest from the **Boundaries** node. Select **Project to Plane...** from the pop-up menu:



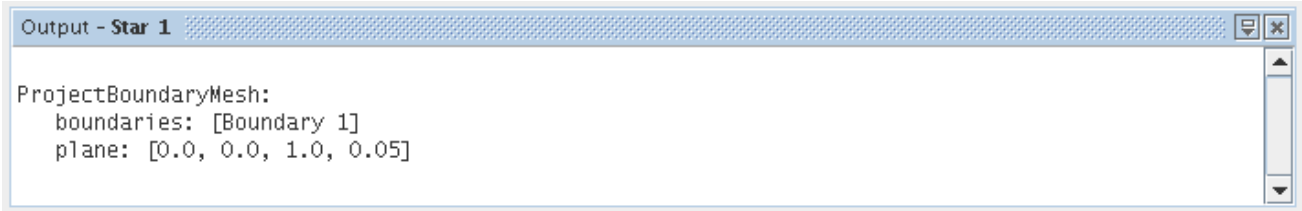
This opens the *Project Boundaries to Plane* dialog with the selected boundaries already marked with a ticked checkbox.



The plane is defined by specifying the length vector, in the *Plane* group box, that is normal to the plane onto which you want the projection to take place. The **Normal X**, **Normal Y** and **Normal Z** inputs specify the direction of the vector and the **Distance** input specifies the length of the vector. The example above specifies a plane that is parallel to the X-Z plane and that is 0.5 meters from the origin. The parameters are illustrated for a general case below.



Click the **Apply** button to start the projection process. A progress bar will be displayed at the bottom of the simulation tree. Once the process has completed successfully, a message will be displayed in the *Output* window indicating the boundaries that were projected and the plane that was used.

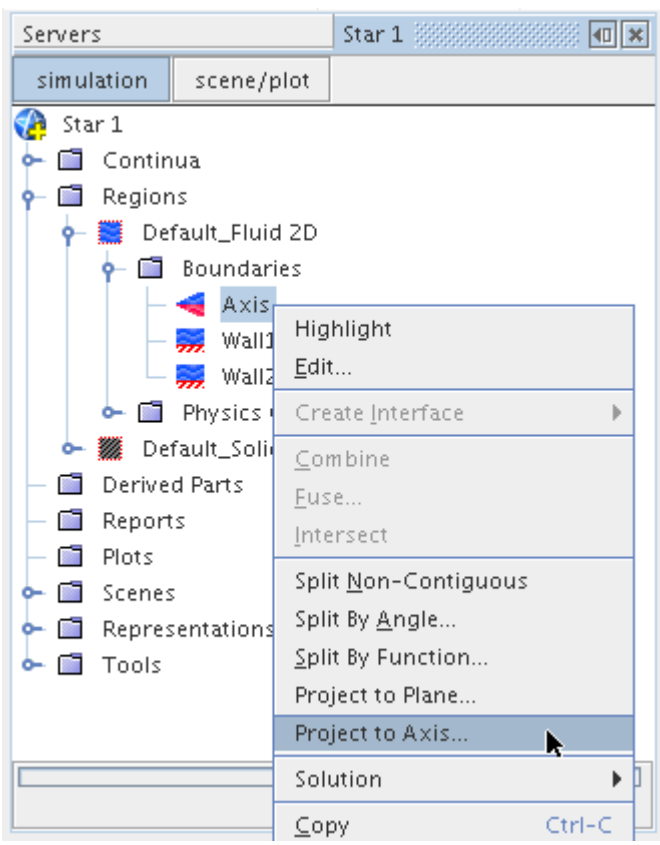


Projecting Boundaries to an Axis

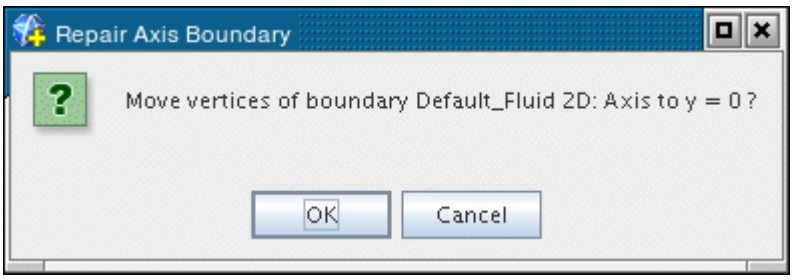
The primary application for projecting boundaries to an axis is to ensure that the correct mesh definition is prescribed for an axis-symmetric type analysis.

The project to axis option can be used on a 2D mesh containing an axis type boundary (for axis-symmetric cases). It is similar to the project to a plane option in that it can take non-planar vertices on an axis boundary and automatically project them to the y=0 axis.

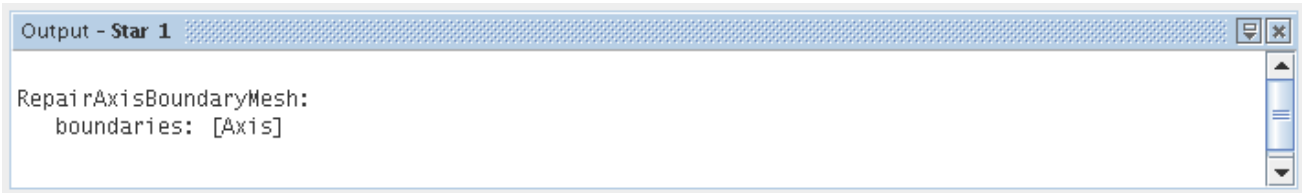
To project a boundary to the axis, the mesh must first have been converted to 2D and an axis type boundary defined. Selecting the axis boundary node and right-clicking will give access to the **Project to Axis...** option, as shown below:



The *Repair Axis Boundary* dialog will appear:



Clicking the **OK** button will proceed with the operation and project the boundary vertices to y=0 in the global coordinate system. Once the process is complete, a confirmation message will appear in the *Output* window:



Note that this operation is not reversible so it is advisable to save the simulation beforehand.

Selecting Boundaries

A dialog is available for selecting boundaries for various operations.

For most of your work with boundaries, STAR-CCM+ opens a standard object selector dialog. It allows selection of multiple boundaries from various regions.

Copying Boundary Properties

If you have two (or more) boundaries that you want to have exactly the same *Type* property, but you have chosen not to combine them, then you can copy that property from one boundary to another.

This feature works on boundaries for the *Type* property only. No other properties, such as interfaces or faces, are copied between boundaries using this operation. This operation does not work with interface boundaries.

Properties of the Boundaries Manager

The Boundaries manager node provides information on the number of boundaries in the simulation.

Part Surface Selection Priority	Lets you prioritize the assignment of part surfaces when assigning them with a dynamic query. See Prioritizing Assignment of Parts.
Boundaries	The number of boundaries in the simulation (read-only). This property changes when you add, combine, split, or delete boundaries.

Pop-Up Menu of the Boundaries Manager

Right-click the Boundaries manager node to add boundaries or group them.

These are the pop-up menu items available when you right-click the **Boundaries** manager node.

New Group	Facilitates the grouping of objects by creating a new group node within the manager node.
Group By	Allows you to group the objects according to any of the properties of the objects in the manager node, listed in a submenu of this menu item.
Ungroup	Reverses the grouping process.
New	Adds a new boundary to the simulation.

Performing Surface Boolean Operations

To combine intersecting surfaces, use boolean operations.

Surface Boolean operations can be performed between two or more closed, manifold surface type regions that intersect in order to create a single region representing the modified surface based on the Boolean type. The new surface is also closed and manifold and can be used as part of the meshing process or as the starting point for additional Boolean operations.

This feature is useful when several separate components have been imported but need to be joined together without any loss of definition occurring, as may happen if the surface wrapper were used to wrap the surfaces into one. Note however that the feature should not be used as a substitute for CAD Boolean operations, which should always be done in preference to doing them once the surface has been tessellated.

Three types of surface Boolean operations are allowed:

- unite;
- intersect; and
- subtract

Each input region can contain one or more boundaries but depending on the Boolean type selected and the shape of the geometries, existing boundaries may be deleted in the resultant surface. Feature curves will also be created automatically to represent any new features that are generated as part of the process. Any pre-existing feature curves that belonged entirely to surfaces that are removed are also removed.

The quality of the input surfaces is not critical as long as the individual regions are closed and manifold. However, non-manifold baffles are permitted, provided the baffle boundary has been converted to an interface. The tessellated surfaces generated during the import of neutral and native CAD data files are adequate starting points in this respect. The new boundary triangles that are added as part of the process are usually of relatively poor quality, so it is recommended that the surface remesher be used to improve the overall quality before performing the volume meshing step.

Note that Boolean surface operations are only possible when a single surface mesh representation is present. If additional representations exist then they must first be deleted prior to carrying out the operation.

Contents:

What are Surface Boolean Operations?
Applying a Surface Boolean

What are Surface Boolean Operations?

A surface Boolean operation allows individual closed surface volumes that are not connected but intersect one another in some fashion to be joined into one closed volume surface.

Three different types of surface Boolean operations are allowed:

- unite;
- intersect; and

- subtract

All three Boolean types work in a similar manner and produce a closed, manifold, single region surface as the output.

If the input surfaces are not intersecting (for example, as in a box within a box) then a Boolean operation should not be performed and an alternative approach is required to setup up the regions for meshing.

In situations where two closed surface volumes don't intersect but have co-planar boundaries then the surface Boolean operations should not be used. If one or more of the surfaces are not closed then the boundary merge operation should be used instead, which allows unclosed surfaces to be intersected with one another.

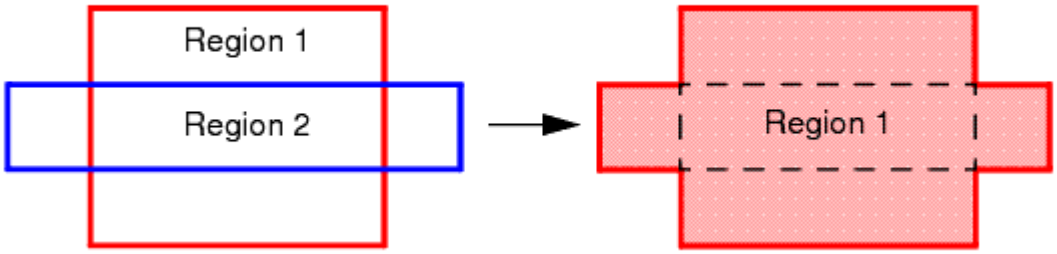
Contents:

Boolean Unite
Boolean Intersect
Boolean Subtract

Boolean Unite

This operation will join together two or more intersecting closed volume surfaces into one closed, manifold surface.

The Boolean unite operation is illustrated below:

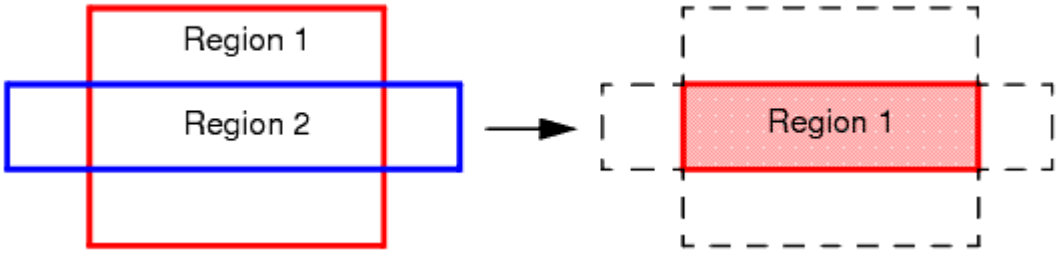


Each surface to be united should belong to a different region. After the Boolean unite, the resulting surface will be stored in the first region selected for the operation. Boundaries and feature curves not required in the output surface are automatically deleted and new curves are included for new features that are created during the process.

Boolean Intersect

This operation will extract the common volume that is shared between two or more closed volume surfaces.

The Boolean intersect operation is illustrated below:

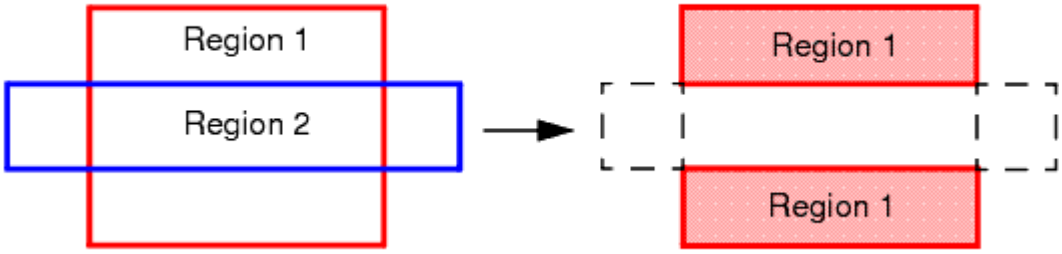


Each surface to be intersected should belong to a different region. After the Boolean intersect, the resulting surface will be stored in the first region selected for the operation. Boundaries and feature curves not required in the output surface are automatically deleted and new curves are included for new features that are created during the process.

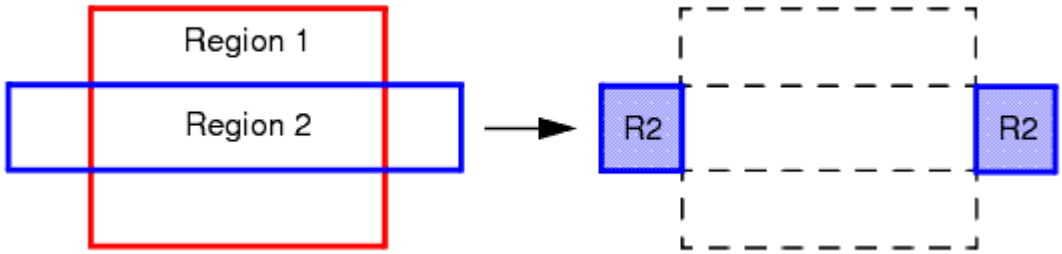
Boolean Subtract

The Boolean subtract operation will remove one or more intersecting closed volume surfaces from another closed volume surface.

For the simple situation where two bodies intersect one another, then either body 1 can be subtracted from body 2 or vice versa, which can lead to two different results. The choice of which body to subtract from is provided during the process. An illustration of both situations is provided below:



In the above example, the second body (region 2) is subtracted from the first body (region 1), leaving two large volumes in region 1. In the example below, the first body (region 1) is subtracted from the second body (region 2), resulting in a smaller set of volumes belonging to region 2.



Each surface to be subtracted should belong to a different region. Boundaries and feature curves not required in the output surface are automatically deleted and new curves are included for new features that are created during the process.

Applying a Surface Boolean

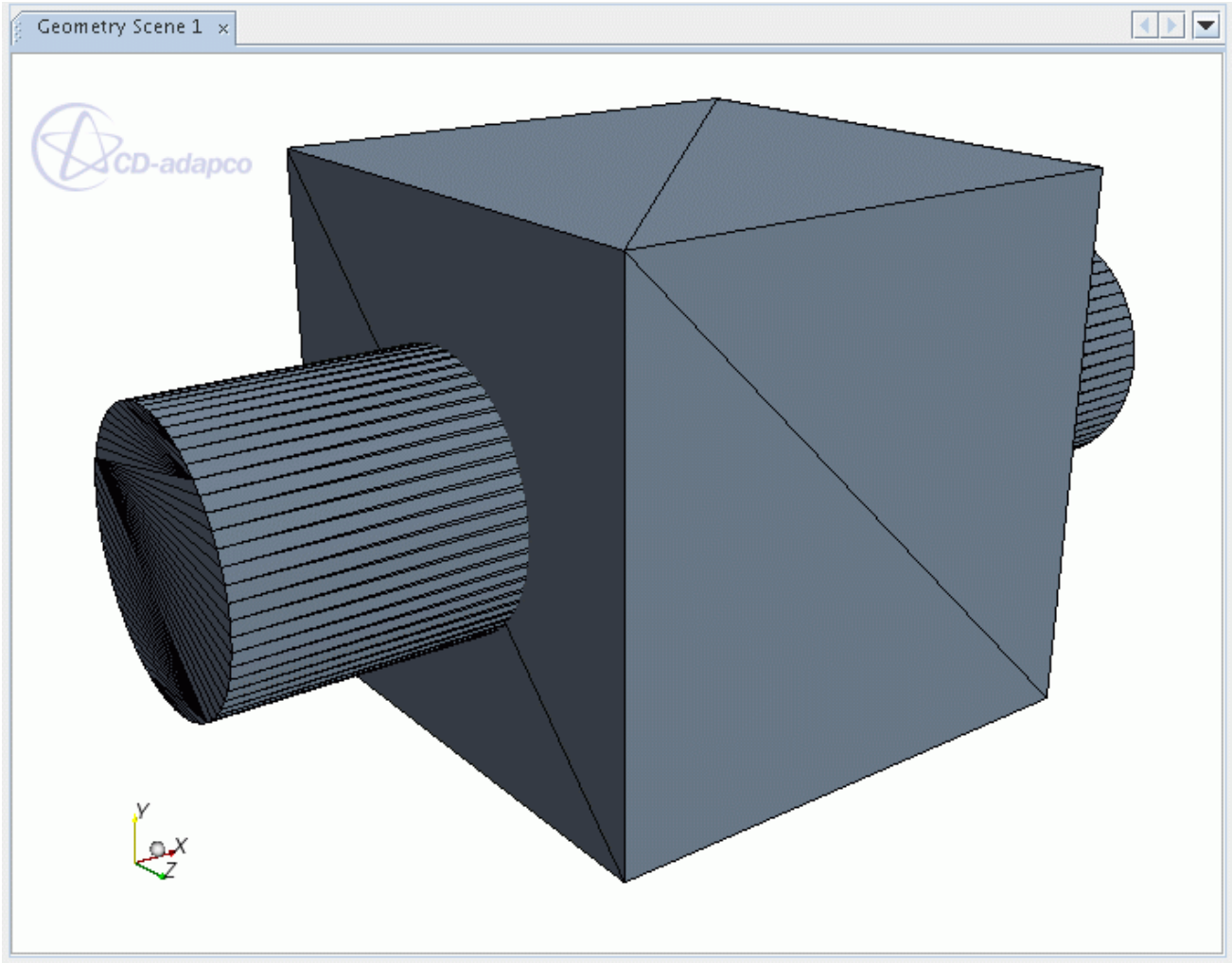
Surface Boolean operations are possible once certain prerequisites are met.

Surface Boolean operations can be carried out once a surface mesh has been imported and at least two regions exist, each containing a closed, manifold surface definition. Regions can be manipulated so that the correct arrangement can be obtained. The surfaces also need to intersect in some way so if the regions are not correctly positioned then they can be transformed so that they are properly aligned.

If any of the surfaces are not closed then the Boolean process will not proceed and the offending region or regions will be reported. You should check and repair the surface before attempting the process again.

Important: Since the Boolean operation is not reversible then its recommended that you save the simulation before proceeding with the steps outlined below.

The three different Boolean types will be demonstrated here on a simple box-cylinder geometry intersection, shown below:

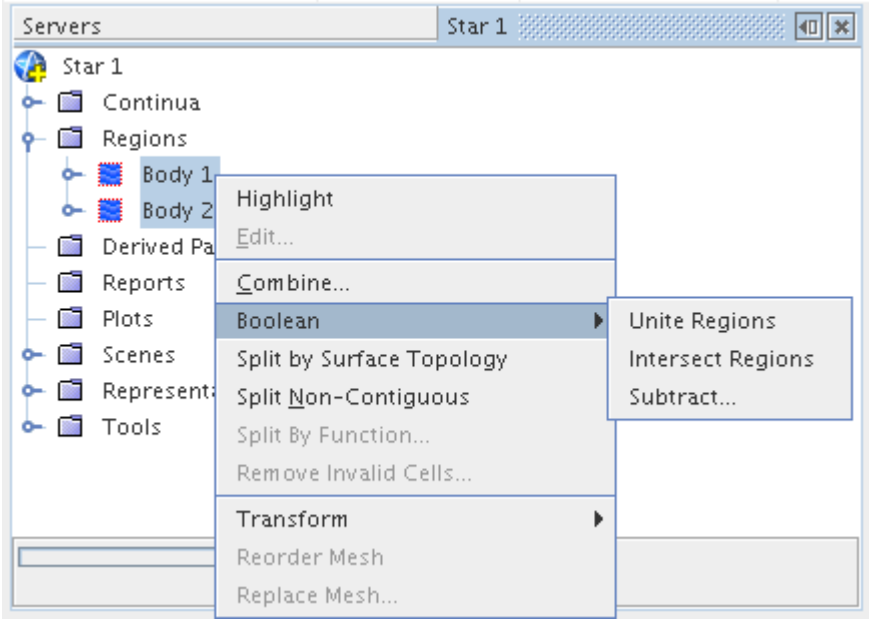


In the above example, two separate bodies were created in the STAR-Design program and exported using the Parasolid Transmit format. They were then imported into STAR-CCM+ using the default organizational and tessellation options, which resulted in two separate regions, each with one boundary.

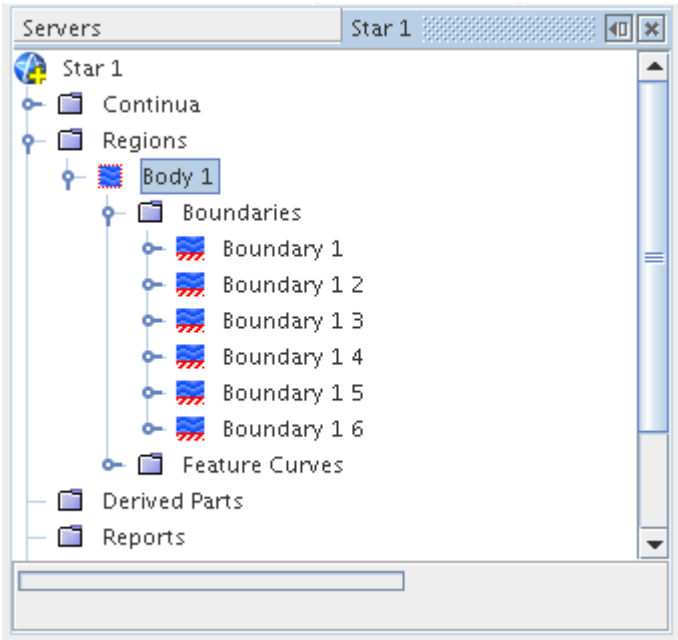
In the simulation tree, click on the first region, press and hold the CTRL button and then select the second and any additional regions to be included in the operation. Right-click on any of the selected regions and select the Boolean option. In the menu sub-options, three different selections will be listed:

- unite regions;
- intersect regions; or
- subtract...

Select the option that corresponds with the operation you wish to perform. These different selections are shown below:



Once the Boolean operation is done, the simulation tree will be updated with the new region and boundary definition. A successful Boolean operation on intersecting regions will always result in a single region definition, with additional new feature curves being created as needed. Boundaries with common names in the original regions will typically be re-named with an ascending numerical suffix:



If the original regions do not intersect, then the following behavior will be observed and reported back in the simulation tree:

- unite - combines the regions into the first one selected
- intersect - deletes the regions and boundaries selected
- subtract - leaves the region selected to subtract from

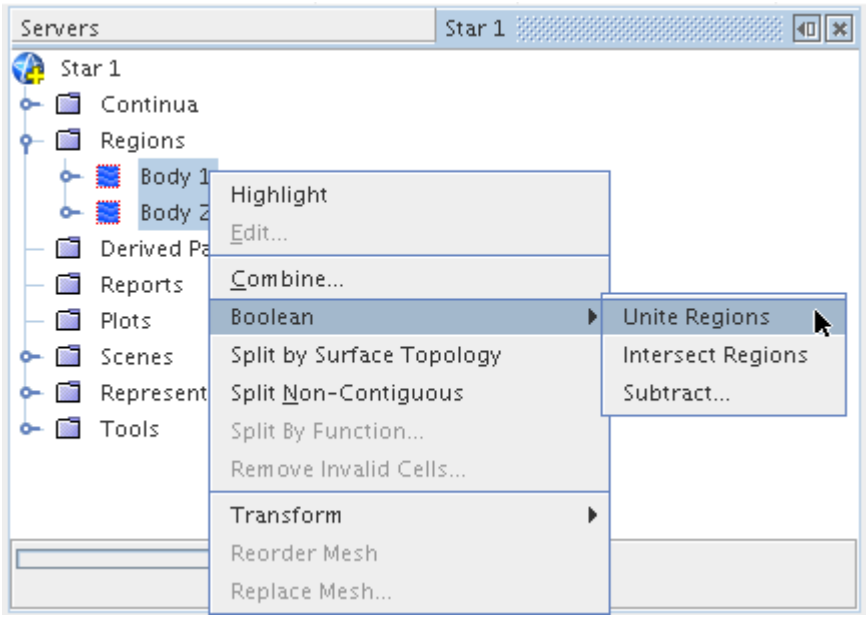
Contents:

Performing a Boolean Unite
Performing a Boolean Intersect
Performing a Boolean Subtract

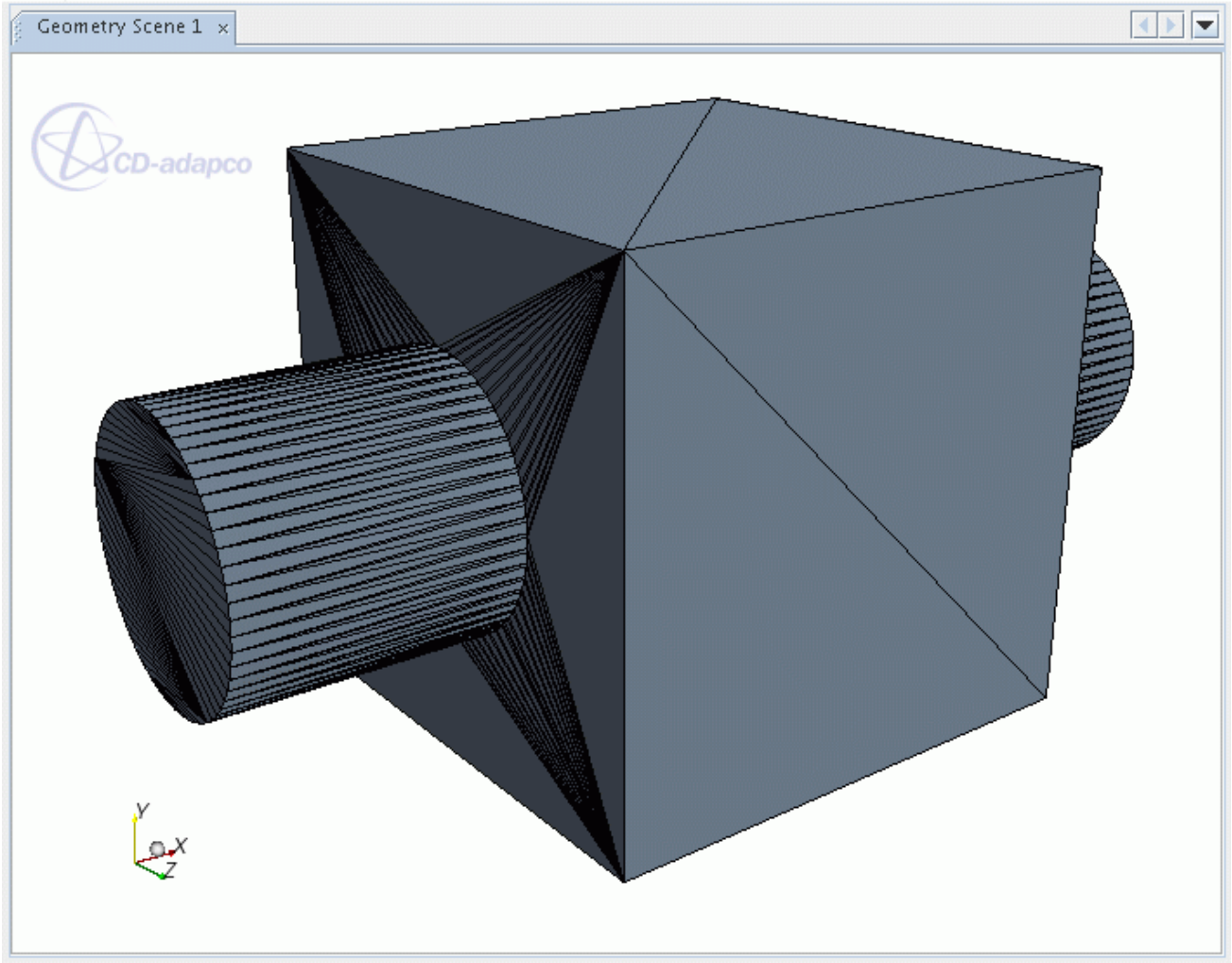
Performing a Boolean Unite

This operation lets you unite surfaces to combine regions.

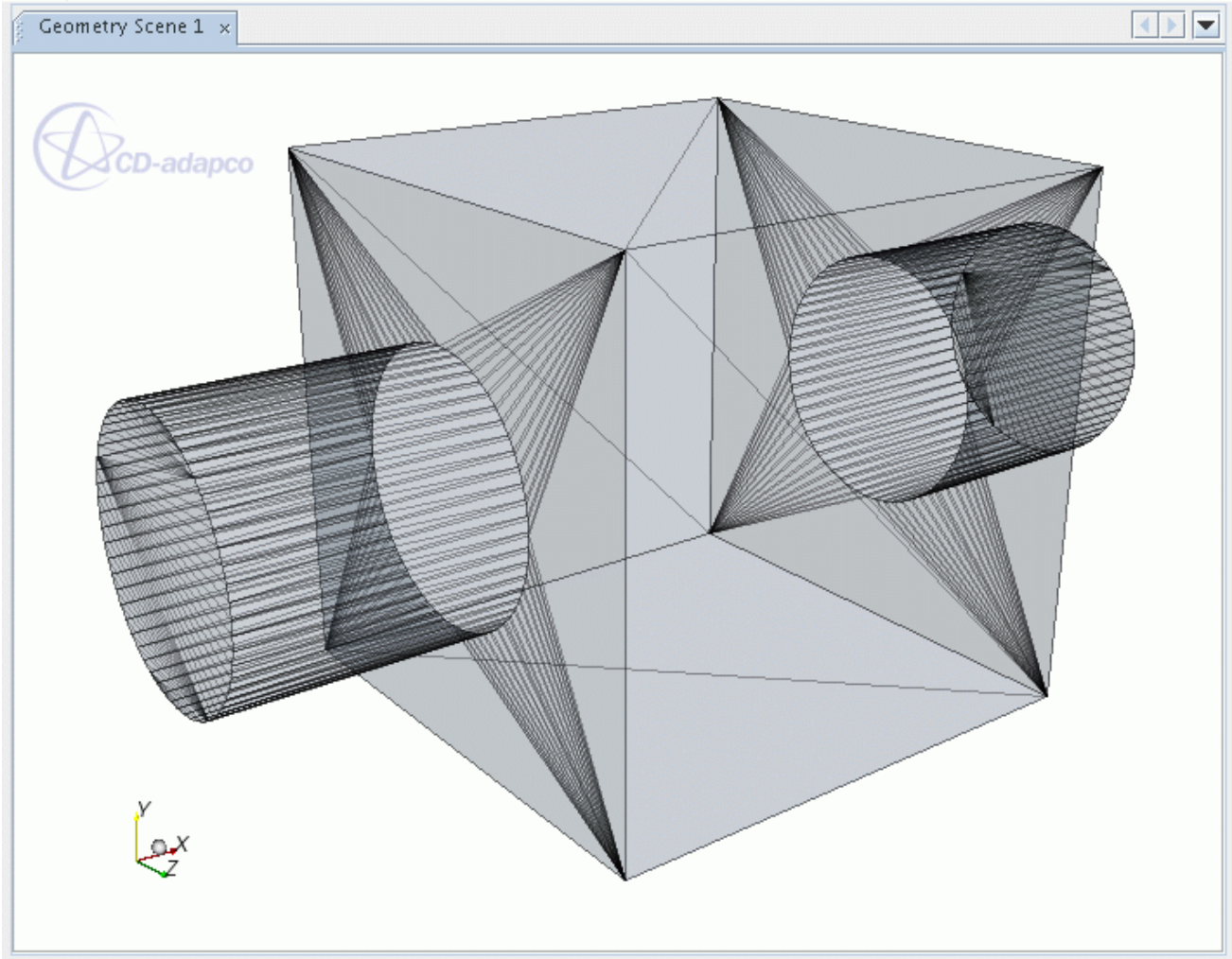
Right-click on the selected regions you want to perform the Boolean unite operation on and select **Boolean** > **Unite Regions** from the pop-up menu. This is shown below:



Based upon the starting surface geometry, the Boolean operation will proceed to unite the surfaces together and combine the regions into the first one selected. The surface representation will automatically be updated with the new surface and displayed in the current scene:



In the above example, and more clearly in the picture below, the intersecting faces of the square block have been re-tessellated to join the cylinder at the intersection. The portion of the cylinder inside the block has also been removed.



Feature curves with the node name **Edges from Boolean unite** will have been included in the geometry at the intersection of the block and cylinder.

The following feedback from the entire process will be seen in the *Output* window:

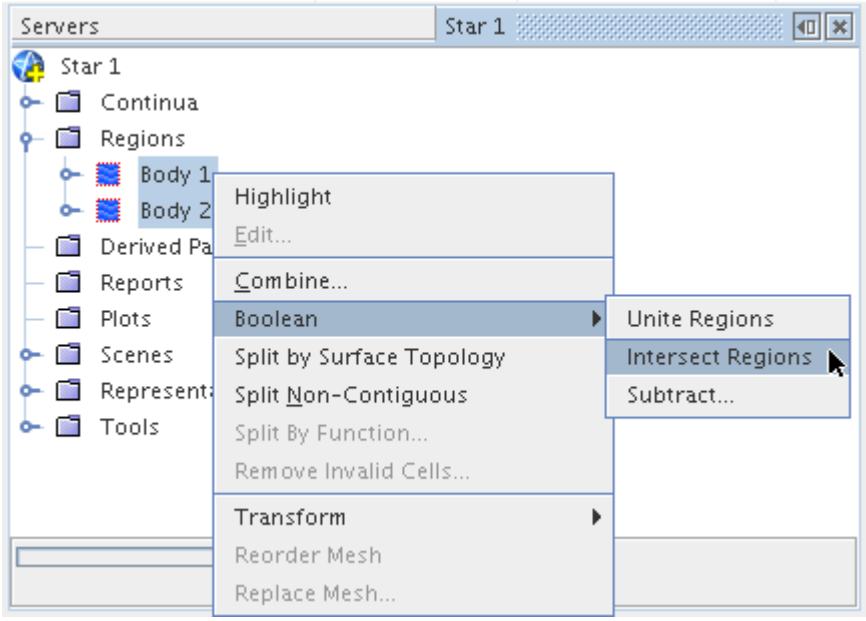
Output - **Star 1**

```
entering SurfaceMeshUniteRegions
begin uniting regions
   target region=Body 1
   peforming boolean unite with region=Body 2
   putting result into region=Body 1
   deleting region=Body 2
done uniting regions
```

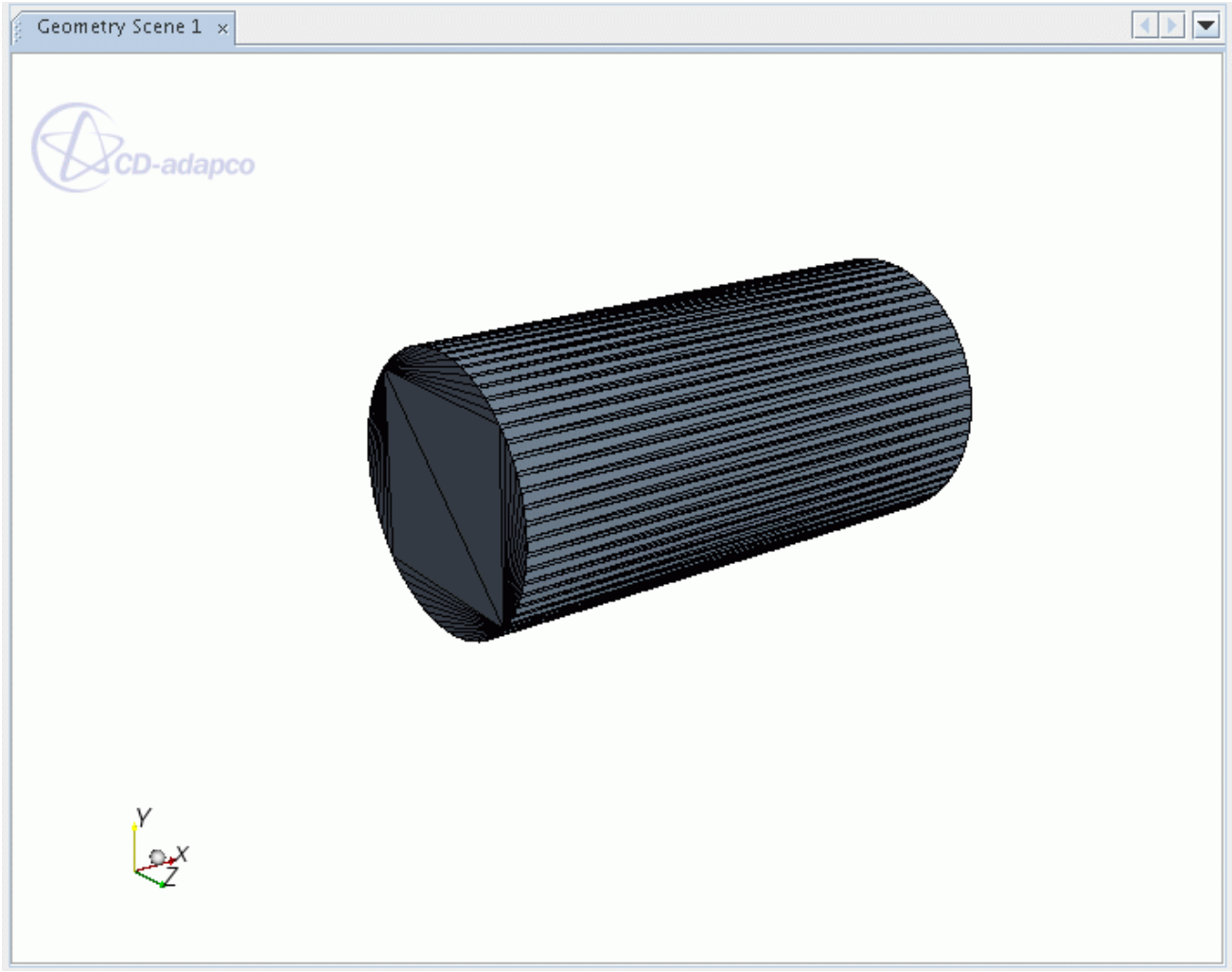
Performing a Boolean Intersect

This operation lets you intersect surfaces to combine regions.

Right-click on the selected regions you want to perform the Boolean operation on and select **Boolean** > **Intersect Regions** from the pop-up menu. This is shown below:



Based upon the starting surface geometry, the Boolean operation will proceed to intersect the surfaces together and combine the regions into the first one selected. The surface representation will automatically be updated with the new surface and displayed in the current scene:



As shown above, the intersection of a block and a tube will result in the interior tube surface being retained, with the ends being capped off to form a closed volume. Feature curves with the node name **Edges from Boolean intersect** will have automatically been included for the ends of the cylinder.

Feedback from the process will be seen in the *Output* window, as shown below:

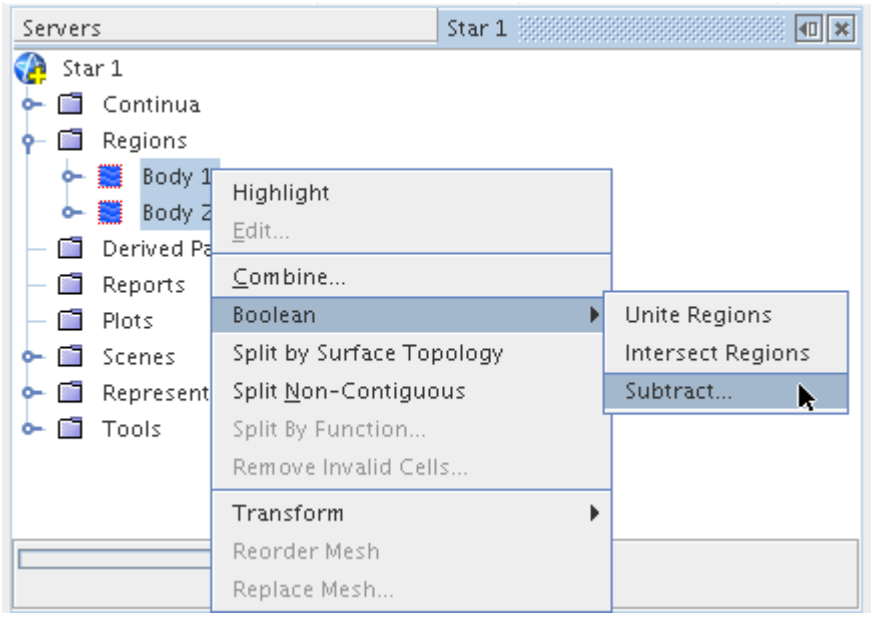
Output - **Star 1**

```
entering SurfaceMeshIntersectRegions
begin intersecting regions
   target region=Body 1
   peforming boolean intersect with region=Body 2
   putting result into region=Body 1
   deleting empty featureCurves=Feature Curve 1 from region=Body 1
   deleting empty featureCurves=Feature Curve 1 from region=Body 2
   deleting region=Body 2
done intersecting regions
```

Performing a Boolean Subtract

This operation lets you subtract one region from another.

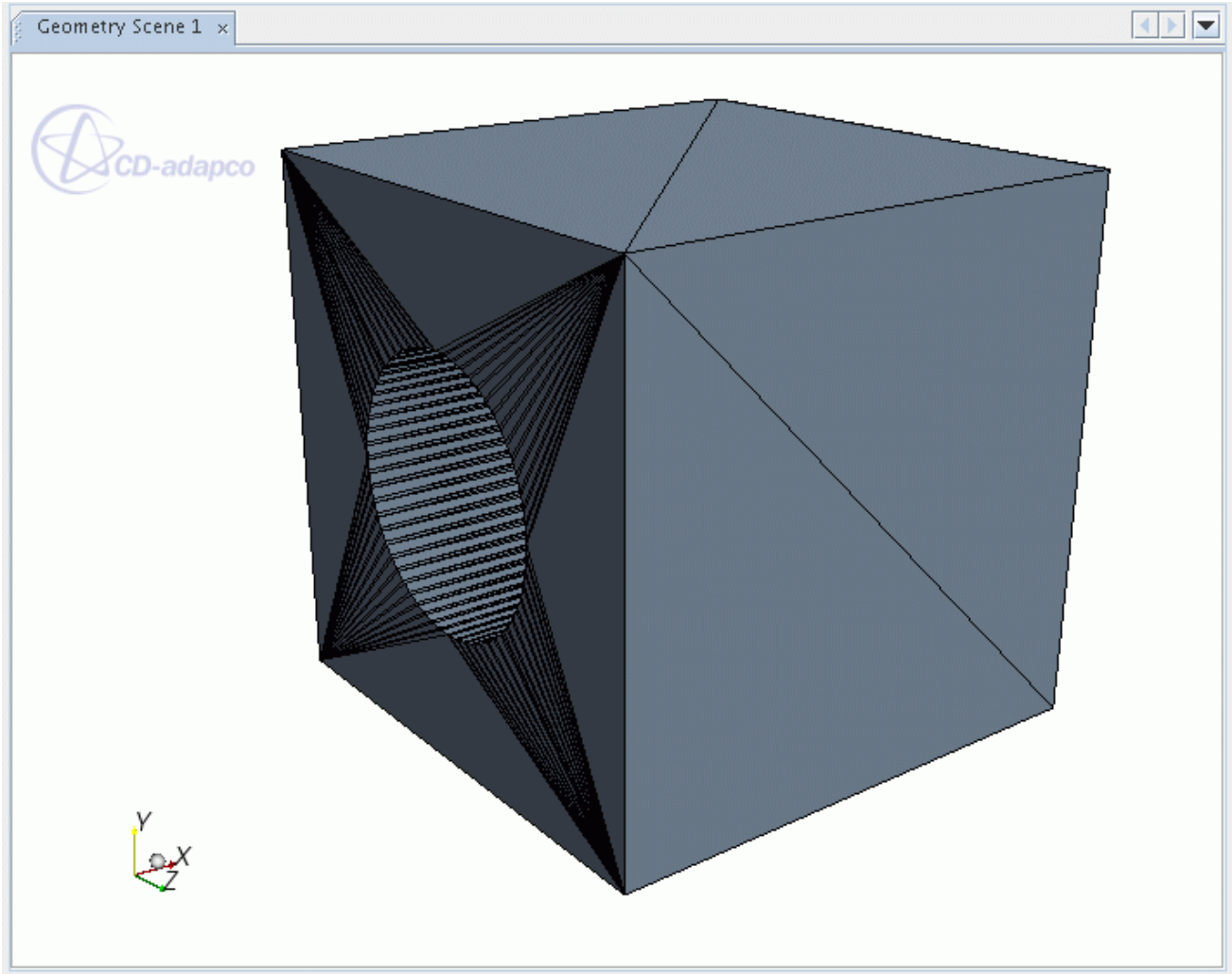
Right-click on the selected regions you want to perform the Boolean operation on and select **Boolean** > **Subtract...** from the pop-up menu. This is shown below:



The *Subtract Regions* dialog will appear as shown below:

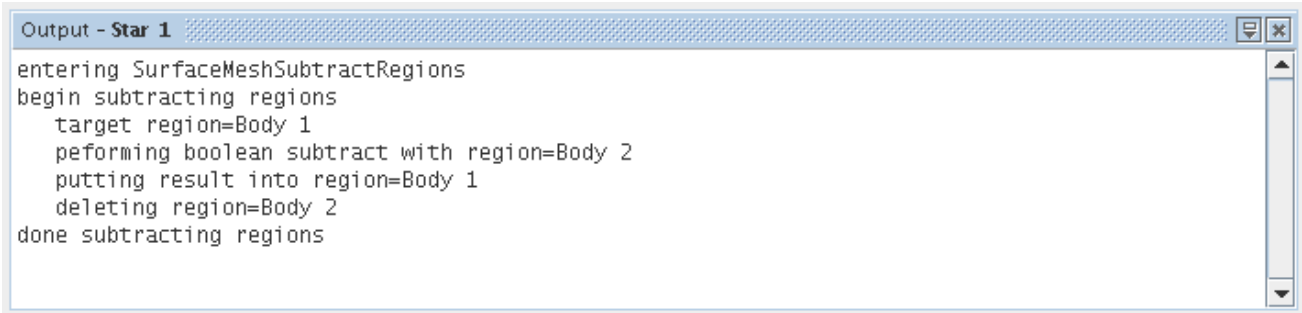
Select the appropriate region from the pull-down menu for the **Select Source Region**. This will be the region from which the other regions will be subtracted. Press **OK** to continue.

Based upon the starting surface geometry, the Boolean operation will proceed to subtract the appropriate region or regions from the source region and save the result. The surface representation will automatically be updated with the new surface and displayed in the current scene:

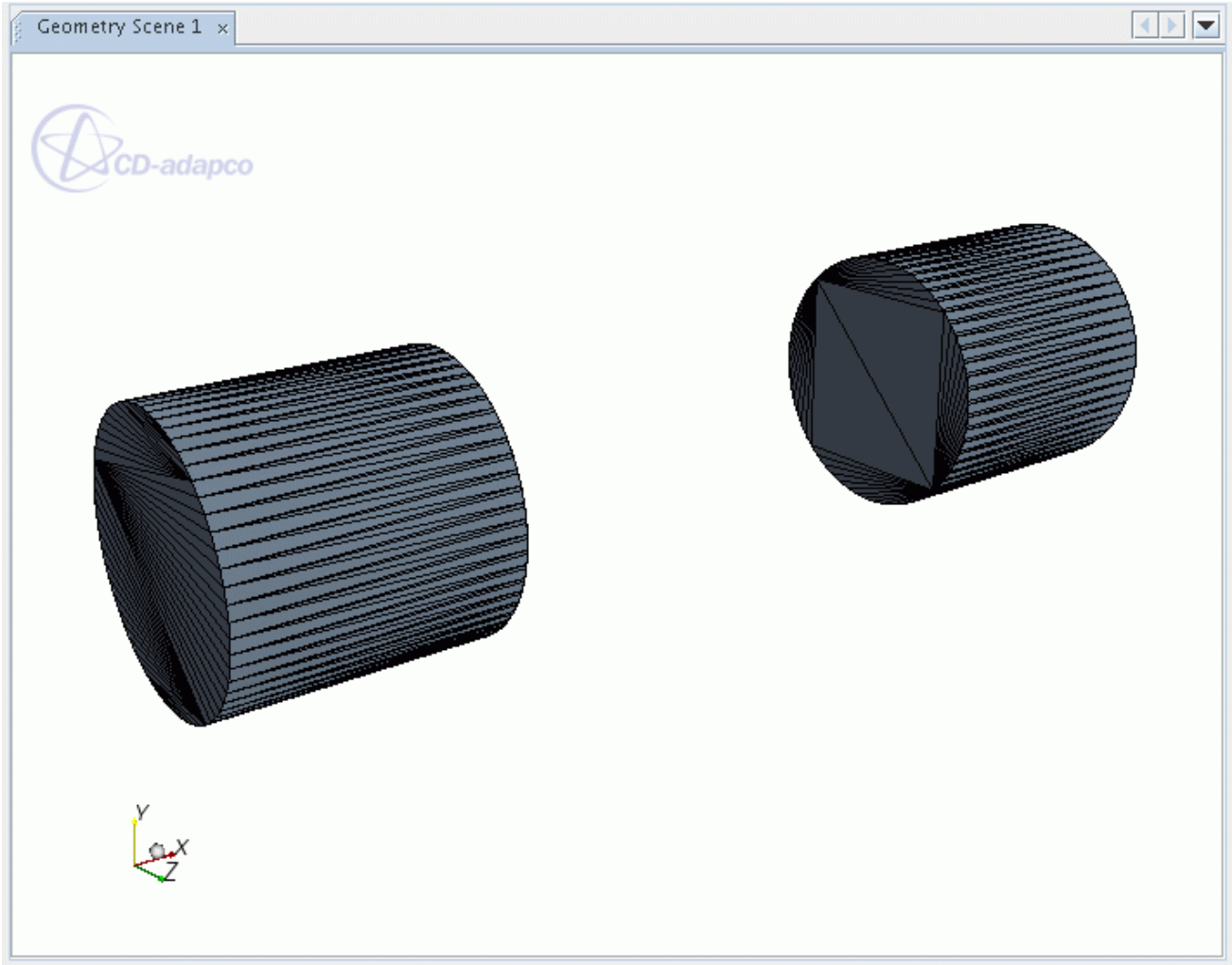


As shown above, the subtraction of the cylinder from the block region will result in a closed block surface with a hole going through it. Feature curves with the node name **Edges from Boolean subtract** will have been included in the geometry at the intersection of the block and cylinder.

The following feedback from the entire process will be seen in the *Output* window:



If **Body 2** had been selected as the source region then the result would be different, as shown here:



In this instance, the subtraction of the block from the cylinder has resulted in two closed cylinder surfaces, both of which belong to the same region.

Meshing

A mesh is a discretized representation of a geometric domain. This domain can include real-world geometry, its content, and its surrounding environment.

STAR-CCM+ solvers find solutions to physics equations at the locations defined by the mesh. For Finite Volume (FV), STAR-CCM+ computes values at cell centers; for Finite Element (FE), STAR-CCM+ computes values at element nodes (solid mechanics) or edges (electromagnetism). STAR-CCM+ can generate meshes suitable for both types of solver.

The FE approach imposes certain requirements on the mesh, for example, polyhedral and trimmed meshes are not supported. For more information on the meshing requirements for FE applications, see the following sections:

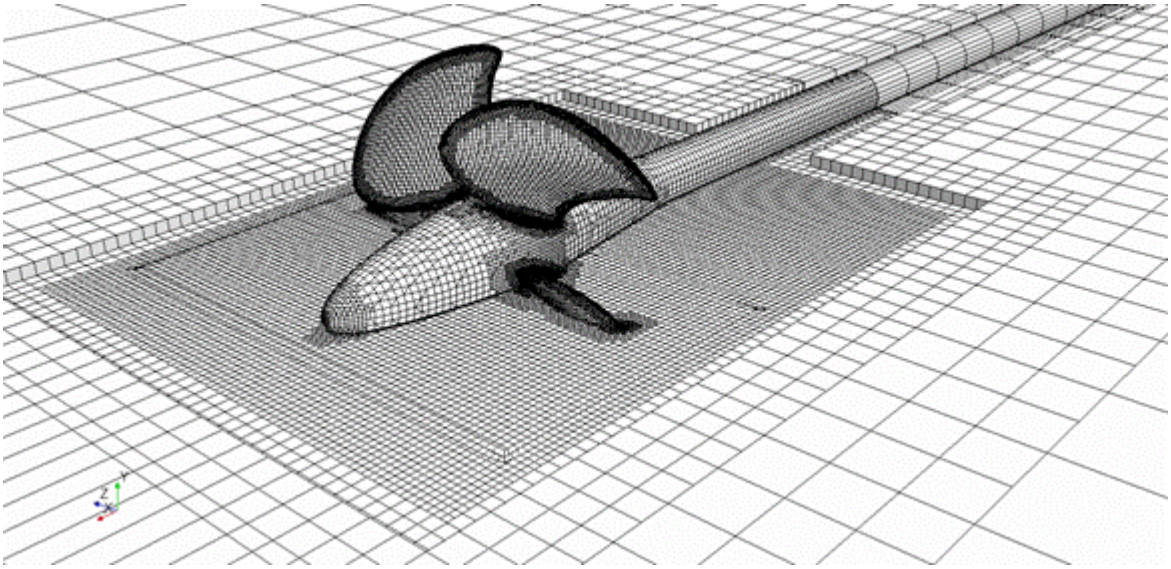
- Viscous Flow: Mesh Requirements and Guidelines
- Solid Mechanics: Mesh Requirements and Guidelines
- Finite Element Magnetic Vector Potential: Mesh Requirements

The physical space that you want to solve within is called the simulation domain. Generating a mesh typically involves creating a suitable simulation domain. There are two types of flow that require different approaches when creating the simulation domain: internal flow, such as a flow in a pipe, and external flow, such as the flow around and through a car.

In cases where you have highly-complex geometry, such as cars and aircraft, capturing this geometry can be challenging. In cases such as these, the Surface Wrapper is used to simplify and capture a watertight representation of the complex geometry. Regardless of whether you use the Surface Wrapper or not, you typically use the Surface Remesher to remesh the initial surface of a geometry. The initial surface of a geometry is usually made up of triangulated surfaces. This triangulation, also know as tessellation, typically contains highly skewed triangles and is not suitable for generating a high-quality volume mesh. Remeshing improves the overall quality of these surfaces by generating more evenly-sized triangles, which is optimal for the mesh models. The remeshed surface is then used as a foundation for the volume meshes.

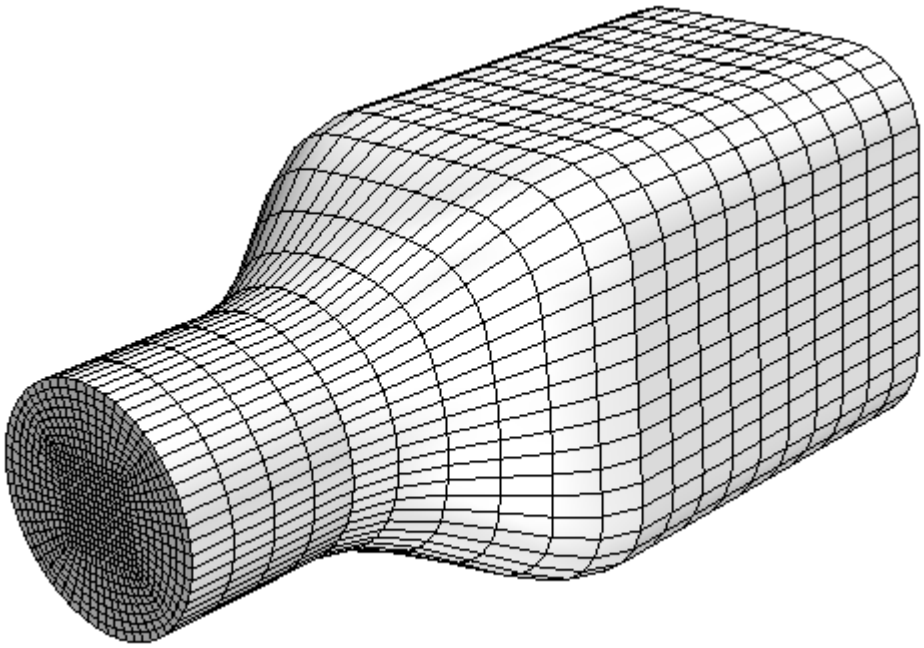
In STAR-CCM+, the surface and volume mesh can be generated automatically or they can be user-guided.

An automatic mesh typically contains irregular mesh structures and is generated using tetrahedral, hexahedral, or polyhedral cells. Refinements can be carried out where needed.



For automatic meshing, STAR-CCM+ offers Parts-Based Meshing (PBM) and Regions-Based Meshing (RBM). PBM detaches the meshing from the physics and provides a flexible and repeatable meshing pipeline. This strategy provides several advantages over region-based meshing. See Parts-Based Meshing. RBM meshes the surface and volume at the region level. You are advised to only use RBM if you intend to use a mesher that is not available in PBM. At this point in time, the only mesher not available in PBM is the Generalized Cylinder Mesher. You can achieve similar results to this mesher using DM, and you are advised to use DM over the Generalized Cylinder Mesher. See Region-Based Meshing.

A user-guided mesh typically contains a regular mesh structure. Refinements usually extend to the bounds of the domain using Directed Meshing (DM).



DM generates high-quality, meshes on CAD geometry. This strategy operates by sweeping a surface mesh along a guide surface starting from either the surface of a CAD geometry or a 2D patch mesh through the volume of a part. See Directed Meshing.

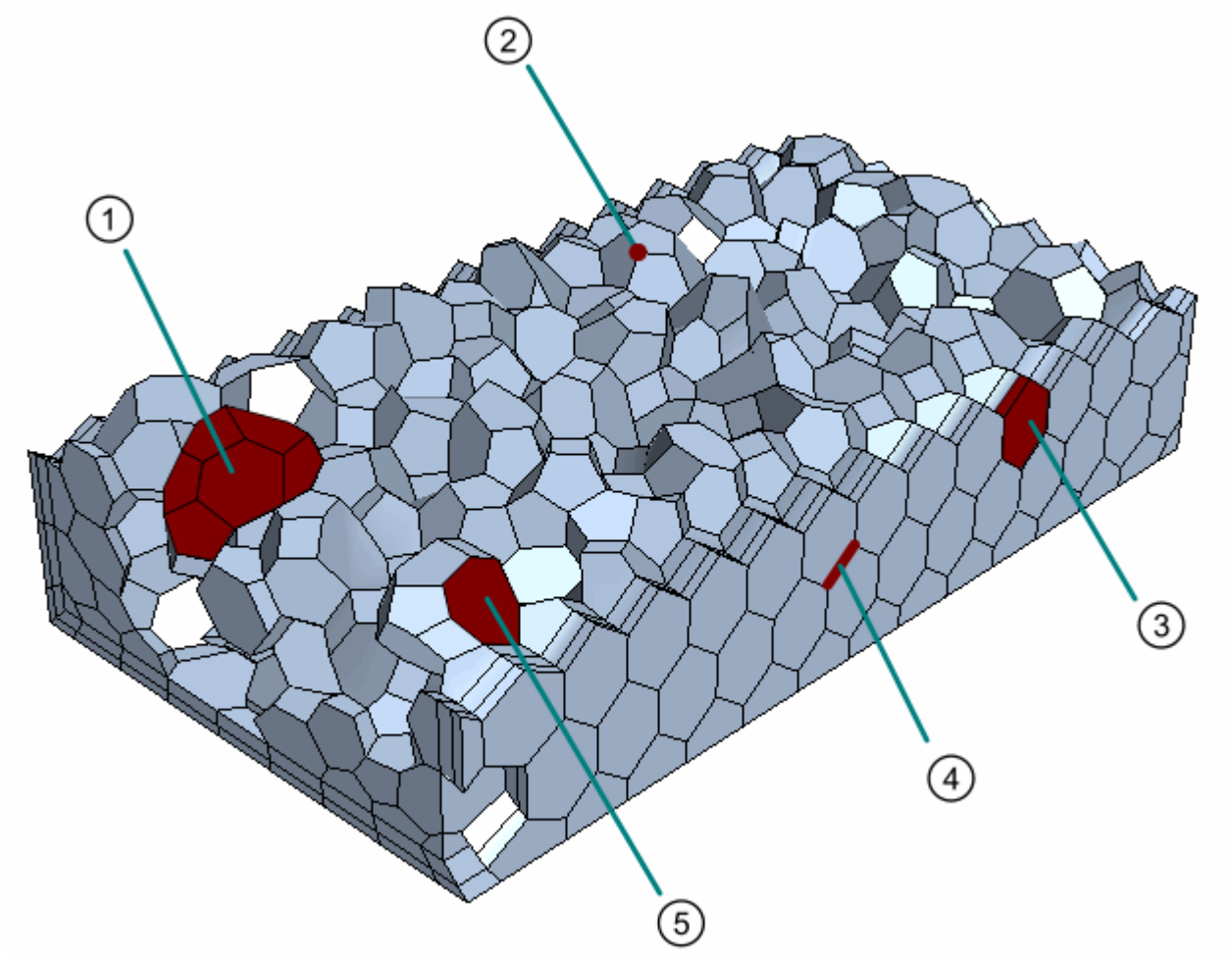
Contents:

Finite Volume Mesh Construction
Meshers in STAR-CCM+
Advantages of Parts-Based Meshing
The Meshing Workflow
Creating a Conformal Mesh
Importing Volume Meshes
Parts-Based Meshing
Region-Based Meshing
Surface Meshers
Volume Meshers
Mesh Refinement
Directed Meshing
Two Dimensional and Axisymmetric Meshes
Mesh Visualization
Checking the Surface Mesh
Checking the Volume Mesh
Meshing Field Functions Reference
Working With Representations
Scaling the Entire Mesh
Working with Baffles
Exporting Mesh and Solution Data
Multi-Region Meshing

Finite Volume Mesh Construction

The volume mesh in a simulation is the mathematical description of the space (or geometry) of the problem being solved.

The following entities are used to construct a mesh in STAR-CCM+.



1 — Cell

A cell is an ordered collection of faces that defines a closed volume in space.

In STAR-CCM+, cells can have any number of faces as long as there are enough faces to create a closed cell volume. The faces of a cell cannot intersect each other, except where they touch along the common edges.

In three dimensions, cells of arbitrary polyhedral shape are permitted. In two dimensions, cells of arbitrary polygons are permitted.

2 — Vertex

A vertex is a point in space that you can define with a position vector. A number of vertices can be used to define a feature curve or a face.

The position vector is an available field function that is composed of the vertex coordinates. This vector is quantitatively different from the centroid vector of a cell or a face.

3 — Prism Layer Cell

A prism layer cell is a specialized thin cell that form the prism layer. Prism layers are important for resolving the boundary layer. Prism layer cells are generated by the Prism Layer mesher. See Prism Layer Mesher.

4 — Feature Curve

A feature curve at its most basic consists of two vertices such that it defines a line in two-dimensional space.

Feature curves are created based on an underlying surface mesh and are used to represent sharp edges and/or surface detail that you want to maintain during the meshing process. They are not used as part of the simulation analysis and are not present in the volume mesh.

5 — Face

A face comprises an ordered collection of vertices such that they define a surface in three-dimensional space.

In STAR-CCM+, faces can have any number of vertices. Four or more faces can be used to define a 3D cell.

Make sure that faces are always simple polygons (that is, the edges do not cross each other) and non-concave.

Correct Face Orientation

You define the orientation of a face with the right-hand rule. In the above example, the face orientation is upwards (assuming you see vertex 2 as being further out of the screen than vertex 4).

When creating a mesh in your mesh generator, make sure that the mesh conforms to this rule.

Concave Face

Concavity can be calculated for any face with four or more sides and refers the amount that one part of the face “sees” another part of the same face. A non-concave cell is flat or planar. However, as the surface becomes warped or twisted, the face starts folding back on itself and produces concavity.

The concavity of a face can be determined by triangulating a face and then extend the normals of each triangle outward until they cross. The angle between the face normals is the concave angle, as illustrated below:

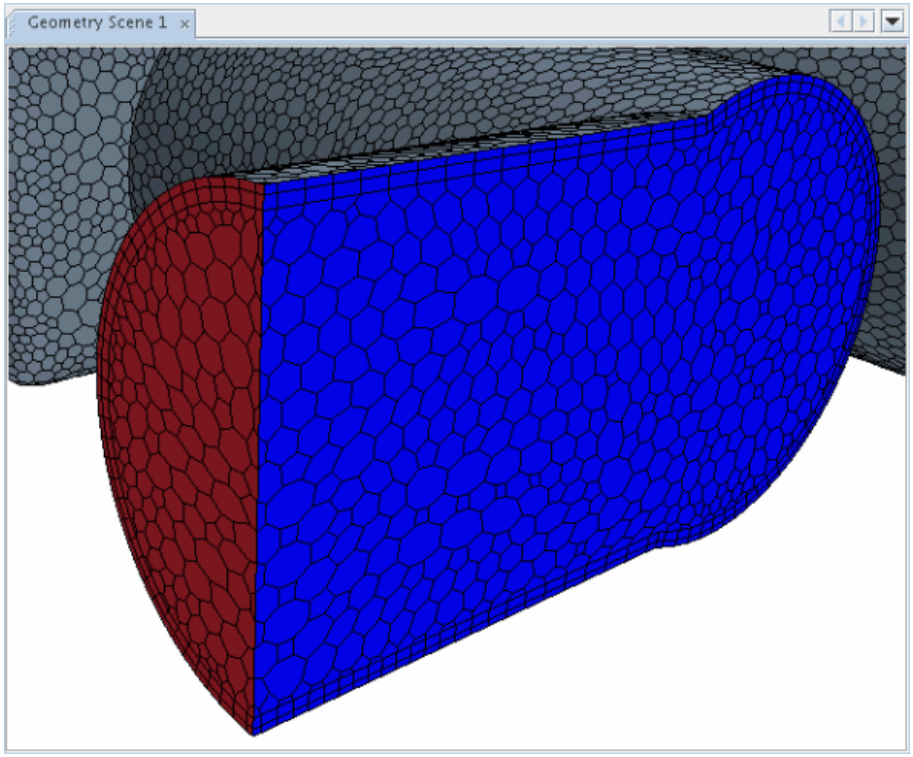
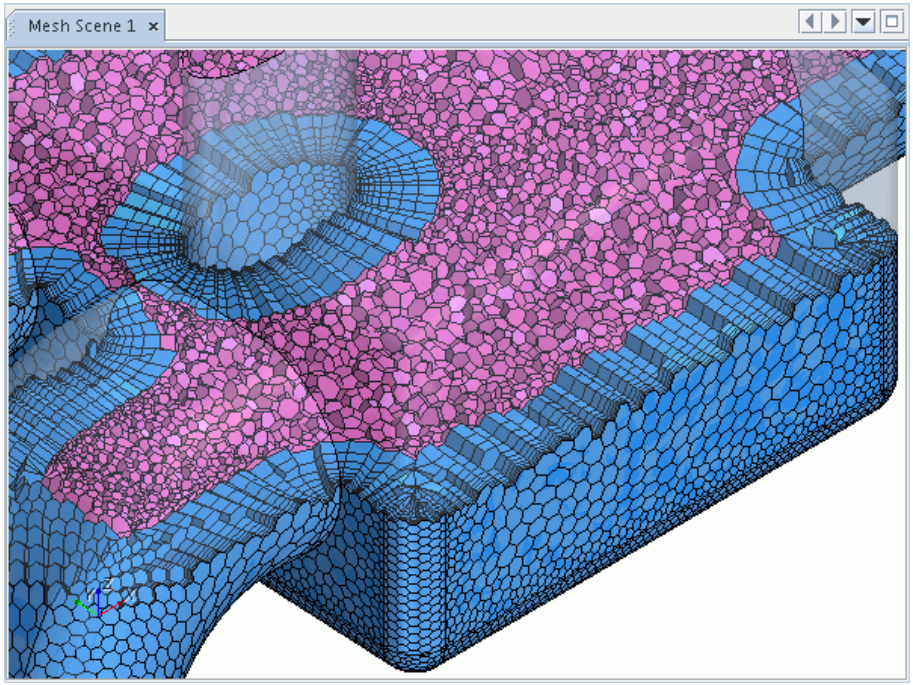
A planar face would have a concave angle of 0 degrees. Keep concave face angles as low as possible make sure that they do not exceed 89 degrees.

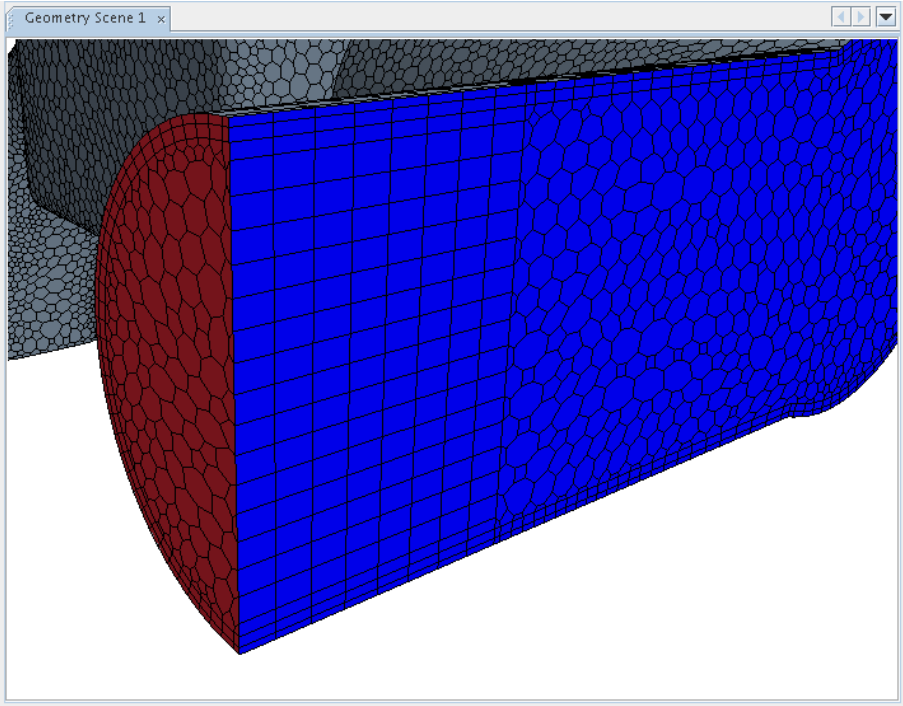
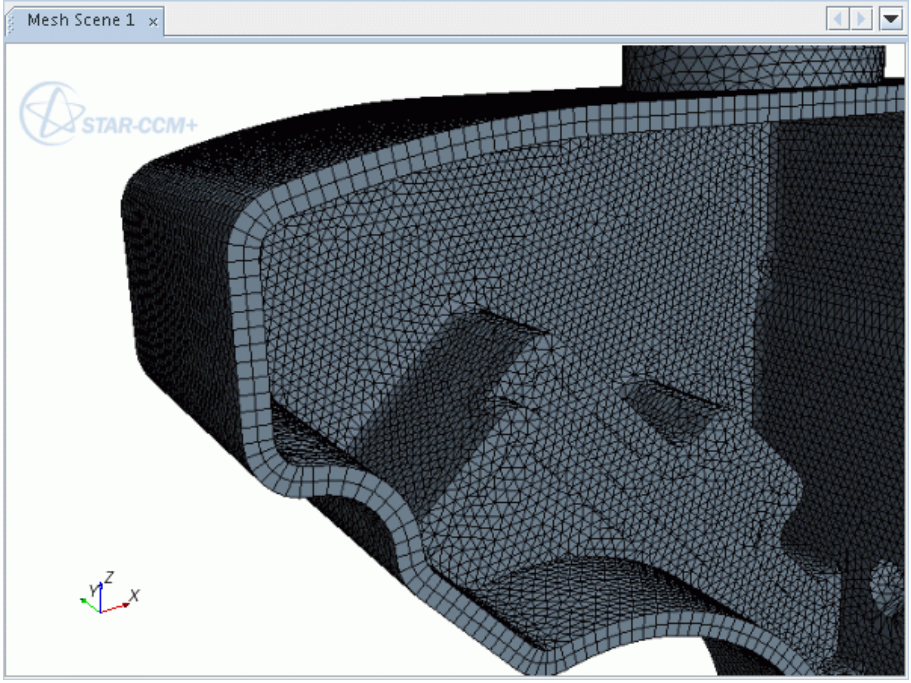
Meshers in STAR-CCM+

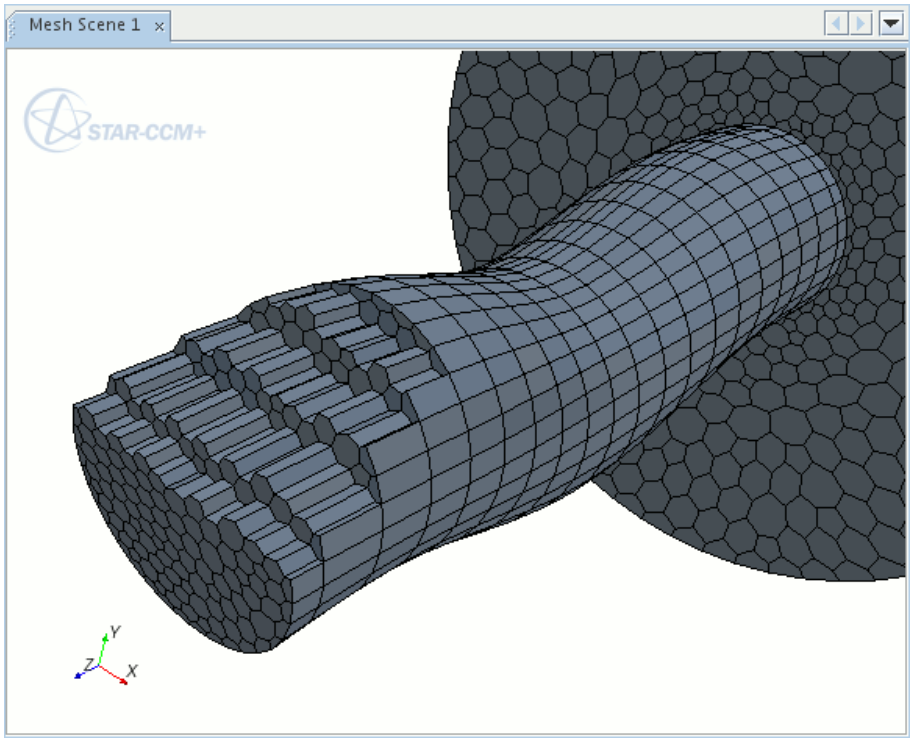
The meshers in STAR-CCM+ improve the mesh of a geometric object and makes it ready for analysis. Some meshers genrate volume cells, while other meshers attempt to simplify and repair geometry.

The following table summarizes the meshers that are currently available in parts- and region-based meshing.

Mesher	Description
Surface Remesher	Remeshes the initial surface to provide a quality discretized mesh that is suitable for CFD.
Polyhedral Mesher	Generates a volume mesh that is composed of polyhedral-shaped cells. Polyhedral meshes are suitable for: • Heat transfer • Swirling flow • Complex flow
Tetrahedral Mesher	Generates a volume mesh that is composed of tetrahedral-shaped cells.
Trimmed Mesher	Generates a volume mesh by cutting a hexahedral template mesh with the geometry surface. Trimmed meshes are suitable for: • Electronics cooling • External flow
Prism Layer Mesher	Adds prismatic cell layers next to wall boundaries. Prismatic layers capture viscous and thermal gradients at the wall.

Mesher	Description
	
Advancing Layer Mesher	Creates a volume mesh that is composed of prismatic cell layers next to wall boundaries and a polyhedral mesh elsewhere. The mesher creates a surface mesh on the wall and projects it to create the prismatic cell layers. 

Mesher	Description
Extruder Mesher	Generates an extruded mesh region from a boundary that one of the core volume meshers has meshed. 
Thin Mesher	Generates a prismatic layered volume mesh for thin geometries. 

Mesher	Description
Generalized Cylinder Mesher	Generates a volume mesh appropriate for elongated cylindrical regions. 

Contents:

Parallel Meshing
Concurrent Meshing

Parallel Meshing

You can use multiple cores to speed up volume mesh generation and create larger meshes than it is possible to create with a single core.

This feature is particularly useful when meshing large parts. Parallel meshes are conformal and of comparable quality to serial meshes. To use parallel meshing in a simulation, start the simulation in parallel mode.

The following meshing models support parallel meshing:

- Polyhedral Mesher (including the underlying tetrahedral mesher)
- Tetrahedral Mesher
- Prism Layer Mesher (including the prism layer subsurface generation for the trimmed mesher. For the tetrahedral/polyhedral mesher, the subsurface generation is carried out in serial or pseudo-parallel.)
- Trimmed Mesher

The following meshing models do not support parallel meshing:

- Thin Mesher

- Extruder Mesher
- Generalized Cylinder Mesher
- Advancing Layer Mesher

For an average mesh (ten million cells or less), it is best to use between four and eight cores with one to two million cells per core. Using a large number of cores for a relatively small mesh reduces scalability and can result in slower performance. For polyhedral and tetrahedral meshes, a minimum of 250 thousand cells are needed per core. However, for trimmed meshes, you can use a minimum of 100 thousand cells per core. If the cell count for each core in parallel falls below the minimum number of cells per core, the trimmed mesher automatically uses fewer cores than are available and re-distributes the workload between these cores. This behavior ensures that there are a suitable number of cells for each core. When the total number of cells that are meshed in parallel falls below the minimum required for each core, the trimmed mesher automatically falls back to serial.

For distributed memory machines, the mesh size is limited by the number of cores and the amount of memory that is available to each core. Ensure that you have enough cores—and memory available to each core—for your entire mesh. The following table shows general memory requirement guidelines when meshing in parallel using the different mesh types. Memory requirement for some cases can fall outside these values.

Mesh Type	Minimum Gb/million cells	Recommended Gb/million cells
Polyhedral	-	1.00-1.50
Tetrahedral	0.08	0.10-0.20
Trimmed	0.60-0.80	1.00-1.50

Note: The *Use Parallel Meshing* property affects the selected automated mesh operation only; if multiple mesh operations are present, you can run each one independently in serial or parallel.

You can perform Parallel Meshing in:

- Parts-Based Meshing
- Region-Based Meshing

Contents:

Generating a Mesh in Parallel

Generating a Mesh in Parallel

Generating a mesh in parallel meshing can speed up volume mesh generation by meshing large parts on several processors.

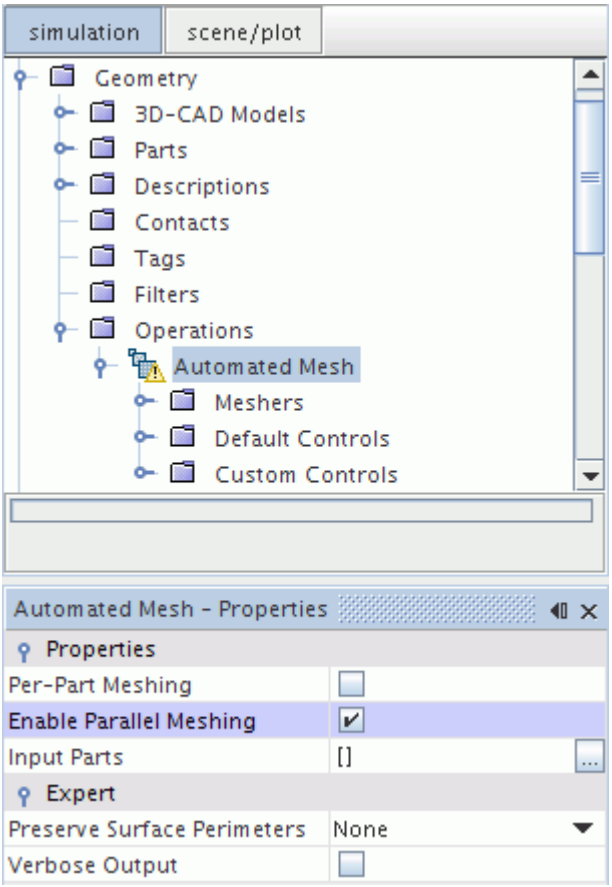
When meshing a large number of small parts using several processors, use Concurrent Meshing. See Concurrent Meshing.

Before Creating a Parallel Mesh, create a new simulation, or load an existing simulation, in parallel mode.

To Create a Parallel Mesh in Parts Based Meshing:

1. Create an Automated Mesh Operation.

2. Select the **Geometry** > **Operations** > **Automated Mesh** node and in the *Properties* window, activate *Enable Parallel Meshing*.



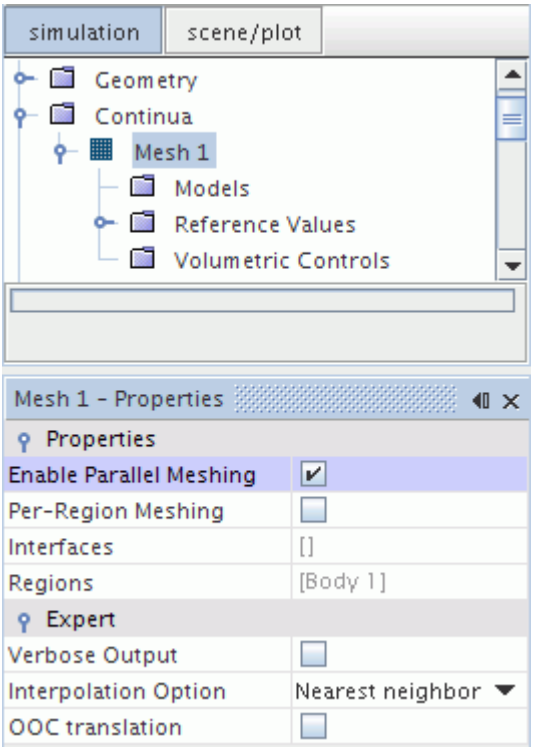
3. Set the mesh controls that you require.

See Setting Default Controls and Creating and Defining Custom Controls

4. Execute the mesh operation.

To Create a Parallel Mesh in Region Based Meshing:

5. Select the **Continua** > **Mesh** node and in the *Properties* window, activate *Enable Parallel Meshing*.



6. Set the mesh model properties that you require.

See Setting Mesh Model Properties

7. Generate the Volume Mesh.

Concurrent Meshing

You can use multiple processors to speed up volume mesh generation and mesh individual parts on separate processors.

This feature is useful when meshing large multi-part assemblies of relatively small parts in per-part mode.

Instead of meshing each individual part successively in serial, the concurrent mesher meshes separate parts simultaneously on different processors that are available.

Concurrent meshing behaves in a similar way to parallel meshing, however, the distribution of parts between processors is different:

- During concurrent meshing, each processor meshes a different part.
- During parallel meshing, parts are subdivided and meshed between different processors.

All except one of the processors that are available for concurrent meshing are assigned separate parts for meshing. This processor acts as a control node and delegates work to the other processors. For this reason, there is no improvement in speed-up when using fewer than three processors for concurrent meshing.

You can use Part Priority to define the order in which parts are meshed concurrently.

Meshes that are generated using concurrent meshing are non-conformal and are of comparable quality to meshes that are generated in serial.

The following meshing models support concurrent meshing:

- Surface Remesher
- Surface Wrapper
- Trimmed Mesher
- Polyhedral Mesher (including the underlying tetrahedral mesher)
- Tetrahedral Mesher
- Prism Layer Mesher

The following meshing models do not support concurrent meshing:

- Thin Mesher
- Extruder Mesher
- Generalized Cylinder Mesher
- Advancing Layer Mesher

Contents:

Creating a Concurrent Mesh Operation

Creating a Concurrent Mesh Operation

Concurrent meshing is useful for speeding up volume mesh generation by meshing large multi-part assemblies of small parts on several processors.

When meshing large parts with several processors, use parallel meshing.

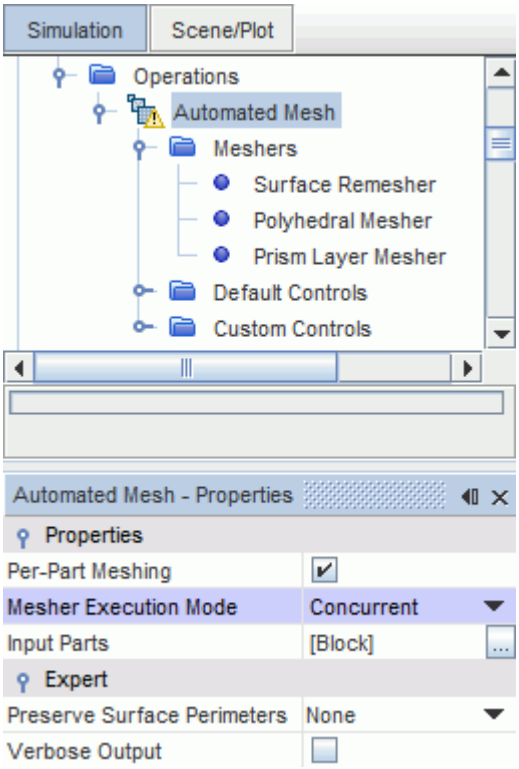
Note: Concurrent meshing is available in Parts-Based Meshing and not Region-Based Meshing.

Before Creating a Concurrent Mesh Operation:

1. Create a new simulation, or load an existing simulation, in parallel mode.
2. Make sure that there are three or more processors available.

To Create a Concurrent Mesh Operation:

3. Create an Automated Mesh operation with compatible meshing models.
4. Select the Automated Mesh node, then:
    - Activate *Per-Part Meshing*
    - Set *Mesher Execution Mode* to **Concurrent**



5. Set the mesh controls that you require.

See Setting Default Controls and Creating and Defining Custom Controls

6. Execute the mesh operation.

Note: If you attempt to abort the concurrent meshing process, the operation will only abort after all processors have finished meshing the parts that are currently assigned to them.

Advantages of Parts-Based Meshing

Parts-based meshing is the focus of all current and future meshing developments in STAR-CCM+. It uses familiar concepts—composites, parts, surfaces, and operations—that are common with CAD and CAE software.

This meshing strategy executes a pipeline of mesh operations on geometry parts. The pipeline provides the following advantages:

- Greater Flexibility: To process the CAD import to a volume mesh, you can define a complex workflow of mesh operations and execute all operations in one action.
- Better Repeatability: To make design changes, you can modify an input part and the changes are propagated through the pipeline to the volume mesh.
- Intuitive Workflow: To build complex linear workflows easily, the simulation tree displays the mesh operations in the sequential order that the pipeline executes them.

In addition to the pipeline, parts-based meshing has several changes in implementation that make it easier to use. The following list summarizes these advantages:

- Custom Mesh Controls: To refine or coarsen the mesh, you can create a single mesh control and apply it to several parts, surfaces, curves, or volumes at once. Previously, you changed the mesh values for each region, boundary, feature curve, volume source individually.
- Inherited Parts: To find objects in complex geometries and pipelines easily, you can categorize the input parts for a mesh operation separately from the part surfaces and curves that the operation generates.
- Surface Wrapper Operation: To make it easier to use the surface wrapper without the other meshers, you use the surface wrapper as a separate mesh operation (instead of a surface mesher).
- Decoupling the meshing from the physics.
- Consistent feature set and usage across all surface and volume meshers.

For unstructured meshing, parts-based meshing is the recommend strategy, unless you intend to use a mesher that is not currently available. See Meshers in STAR-CCM+.

The Meshing Workflow

This section contains the recommended workflow for setting up and generating the volume mesh at the parts level.

Parts-Based Meshing is recommended for most scenarios. Each operation is part of a repeatable pipeline. Parts-based meshing detaches the meshing from the physics and provides a flexible and repeatable meshing pipeline. This strategy provides several advantages over region-based meshing.

The procedure for generating a mesh can differ depending on what you want to simulate, however the general steps to follow to generate a mesh are typically the same.

1. Prepare the geometry for meshing using part operations.

   See Preparing Geometry for Meshing.
2. If necessary, wrap your geometry.

   See Surface Wrapping the Geometry.

   To preserve geometric fidelity, omit parts that do not require wrapping.
3. Imprint all parts together.

   See Imprint Parts.
4. Extract the simulation domain.

   Typically you extract the internal or external volume.
5. Right-click the **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
6. Select the appropriate meshers for your particular application.

   See Meshers in STAR-CCM+.
7. If you want to generate the mesh using multiple nodes, use one of the following techniques:
   - **Parallel Meshing** — this technique distributes the simulation domains across nodes. You can generate conformal meshes using this technique. See Parallel Meshing.

     To use parallel meshing, select the **Geometry** > **Operations** > **Automated Mesh** node and in the *Properties* window, activate the **Enable Parallel Meshing** option.
   - **Concurrent Meshing** — this technique distributes individual parts to individual nodes, similar to a queuing system on a cluster. You can only generate non-conformal meshes using this technique. You can change the order that parts are meshed to minimize the amount of time it takes to mesh. Concurrent meshing is only available when per-part mode is activated. See Concurrent Meshing.

     To use concurrent meshing, select the **Geometry** > **Operations** > **Automated Mesh** node and activate the **Per-Part Meshing** option and set *Mesher Execution Mode* to **Concurrent**.
8. Define the mesh settings. See Setting Default Controls.

   As a general guideline create a coarse mesh first. You can refine the mesh as needed. To help plan the mesh, use the (**Measure Distance**) tool. See Using the Distance Measuring Tool.
9. Create custom controls to refine the mesh in specific areas of the simulation domain or on specific areas of the geometry. See Custom Mesh Controls.
10. Assign the parts to regions. See Assigning Parts to Regions.
11. Create a scene to visualize the mesh. See Mesh Visualization.

12. Click (**Generate Surface Mesh**) and check the quality of the generated surface mesh. See Checking the Surface Mesh.
13. Make sure to use appropriately sized prism layers in the fluid region. See Prism Layer Mesher.
14. If you are fine-tuning the mesh around a particular area of the geometry, or if you are making changes the geometry design, set up local remeshing. See Local Remeshing.
15. Click (**Generate Volume Mesh**) and check the quality of the generated volume mesh. See Checking the Volume Mesh.

    Make sure that you check for bad cells. See Identifying and Visualizing Bad Cells.

Creating a Conformal Mesh

Wherever possible, STAR-CCM+ aims to generate a conformal mesh between parts. This section provides advice about how to make sure that a mesh is conformal.

A conformal mesh is a mesh which encompasses separate geometry parts without interrupting the continuity of the mesh between contacting parts. In a conformal mesh, the perimeter of cell faces that are on the surface of one part, match up exactly with the coincident cell faces on a contacting part.

The process of making sure that a mesh is conformal begins with the original geometry parts - long before setting up the meshing procedure. If the CAD or import tessellations of geometry parts are conformal when they are imported to STAR-CCM+, the surface remesher in STAR-CCM+ maintains this conformality. The quality of the starting geometry affects the quality of the surface mesh, which in turn affects the quality of the volume mesh, and therefore the accuracy of a solution.

The surface remesher in STAR-CCM+ aims to generate a conformal surface mesh but does not fix geometries that are non-conformal.

It is important to define the contacts between the surfaces. STAR-CCM+ creates strong contacts between part surfaces when geometry is conformal, or weak contacts when geometry is not conformal. See In-Place Contact.

- If conformal CAD parts are imported with the Import Surface Option *Create Part Contacts from Coincident Entities* activated, strong contacts are created automatically and appear under the **Geometry** > **Contacts** node. The *Output* window displays the message `Created n new Part Contacts from n pairs of coincident faces.` - where `n` is the number of strong contacts that are created.
- If no part contacts are created, you can imprint parts using an Imprint mesh operation with the option *Perform CAD Imprint* activated. See Imprint Parts. Make sure that the **Geometry** > **Operations** > **Imprint** node Property *Resulting Mesh Type* is set to **Conformal**. Upon executing the Imprint operation, the *Output* window displays the message `Imprint operation Imprint created n new contact(s).`
- You can also find part contacts by right-clicking the **Geometry** > **Contacts** node and selecting **Find Part/ Part Contacts**, however, it is not guaranteed that these contacts are strong and can create a conformal mesh.

The contacts must be defined between parts before moving on to meshing. Before creating a volume mesh, assign parts to regions with the option to *Create Interfaces From Contacts* activated. See Assigning Geometry Parts to Regions. Conformal interfaces are created from strong contacts and allow you to create a conformal mesh.

Since the meshers treat each type of interface differently, set the Interface Type before creating the mesh. See Interfaces Overview.

When a Periodic interface is set, you can determine if the mesh at the interface is conformal when running the Surface Remesher by checking messages in the *Output* window, such as:

- Before the remesher starts:

```
Before Orienting Faces outward

Creating conformal mesh on periodic interface:

Created conformal mesh for interface Block/Block -> Block Surface 2/
Block Surface 4 using translation (0 1 0)
```

- Before the volume meshing starts:

```
Surface Remesher is up to date

Creating conformal mesh on periodic interface:

Created conformal mesh for interface Block/Block -> Block Surface 2/
Block Surface 4 using translation (0 1 0)
```

When using any other type of interface, you can determine if the mesh at the interface is conformal after creating a volume mesh, either:

- Create an Automated Mesh Operation using the Surface Remesher model along with either the Tetrahedral or Polyhedral core volume mesher. The Automated Mesh Operation property *Per-Part Meshing* must be deactivated (when activated, a non-conformal mesh is created). To make sure that the mesh is conformal, all contacting parts must use the same types of mesher. The Trimmed mesher is not capable of creating a conformal mesh - regardless of using the prism mesher at the interface or not.
- Create a Directed Mesh Operation using the option to create a source mesh using an existing mesh. See Creating a Source Mesh Using an Existing Mesh.

Upon completion of volume meshing, you can check if the resulting mesh is conformal by either:

- Right-clicking the **Interfaces** > **[Interface]** node and selecting **Initialize**.

  The *Output* window displays a message such as:

```
Updating 1 boundary mode interface

Block/Block 2: 116 intersection faces (conformal match)
```

- Visualizing the mesh in the *[Mesh Scene]* window.

If the geometry is not recognized as conformal, return to the point of geometry creation and fix the geometry then re-import it. Or, after importing to STAR-CCM+, you can fix the geometry using the surface repair tools. See The Surface Repair Tool.

Importing Volume Meshes

This section provides details on the volume mesh formats supported in STAR-CCM+.

You can import a volume mesh from a range of external file formats, and check its quality using various STAR-CCM+ features, including diagnostics and reordering. The following formats are supported:

- CD-adapco CCM Format
- CD-adapco NGEOM Format
- Gridgen Meshes
- CGNS Format
- Plot3D Formats
- Fluent Formats

If you choose to create a volume mesh within STAR-CCM+, it is possible to work with surface formats, for importing into 3D-CAD and/or importing a surface mesh.

When you import a surface mesh or volume mesh, STAR-CCM+ automatically assigns boundary types.

In addition, the volume mesh importer can read model files relating to the following packages for further manipulation within STAR-CCM+:

- Abaqus (.inp)
- Ensight (.case)
- NASTRAN (.nas, .bdf)
- RadTherm (.ntl, .neu)
- STAR-CCM+ Boundary Data (.sbd)
- Visual Toolkit (.vtk)

Currently, STAR-CCM+ is limited to importing surface descriptions for NASTRAN, RadTherm, and Visual Toolkit files.

CD-adapco CCM Format

The CCM file type is native to the STAR-CD solver (STAR) and pre-processor (pro-STAR).

Since STAR-CD v4.04, the following extensions are used:

- `.ccmg` signifies a CCM file from pro-STAR containing geometry data.
- `.ccmp` signifies a CCM file from STAR (the solver) containing solution post-processing data.
- `.ccmt` signifies a CCM file from STAR containing transient post-processing data.

STAR-CCM+ can read the `.ccmg` file directly to retrieve the mesh. When reading a `.ccmp`, the corresponding `.ccmg` is also read in at the same time so that the post data can be displayed. Only the data corresponding to the first state in the `.ccmp` file is available. In the case of transient `.ccmt` files, rename them as `.ccmp` files, which allows the first time step saved on the file to be read in. Other time step values in the same file are available.

Mesh data is exported to the file by pro-STAR using the command GEOM. In the case of the `.ccmp` and/or `.ccmt` files, solution data is saved by the STAR solver directly (as part of an analysis run). pro-STAR can also write solution mapped data (command SMAP) to the `.ccmp` file as part of a restart.

A scale factor for the mesh is saved to the file during export from pro-STAR but the vertices that are contained in the file are not scaled. The scale factor is used to redimension the mesh during the import process into STAR-CCM+.

Each collection of cells in pro-STAR with the same properties (material label, material index, porosity index, spin index and group index) is considered to be one region. Different cell types do not result in different region definitions unless one of the above properties differs for the cell type.

The CCM file format is the recommended way of transferring meshes and post data from STAR-CD to STAR-CCM+. STAR-CCM+ can also export both mesh and solution data to STAR-CD using the CCM file format, although not all solution variables are transferable.

CD-adapco NGEOM Format

The `.ngeom` file type was originally designed for use with the COMET solver, but is now replaced by the CCM format.

The `.ngeom` format is exported by pro-STAR using the NGEOM command and can only contain mesh data. If the appropriate scale factor was applied during the export process, the vertices that are contained in the file conform to SI and no further action is required on importing the file into STAR-CCM+. The scale factor that is used is also saved to the file but is not used. Boundary types are inferred from boundary names, and if no inference can be made, then a boundary is assigned the type wall.

Each collection of cells with the same properties (material label, material index, porosity index, spin index and group index) in pro-STAR is considered to be one region. Cell types do not affect the region definition unless one of the above properties differs.

Gridgen Meshes

Gridgen can export Fluent format files with the `.cas` extension and STAR-CCM+ files with the `.inp` extension.

Files with the `.cas` extension have the same format as a Gambit `.msh` file and as such can be renamed to have this extension. These formats are the best route from Gridgen to STAR-CCM+. It is important to ensure that the mesh is specified as right-handed in Gridgen before it is exported.

Although Gridgen can export STAR-CD cell/vertex/boundary/input files also, they are not compatible with volume mesh import in STAR-CCM+. If you have access to pro-STAR, the files can be read in using the IFILE command and then resaved as a CCM file.

Plot3D Format

Plot3D files have various formats and extensions. STAR-CCM+ currently recognizes this file type only with the following extensions:

- `.grd`
- `.p3d`
- `.xyz`

- `.x`

and with the following characteristics:

- three-dimensional
- single-block or multi-block
- FORTRAN-unformatted or C-binary

There is no need to specify the characteristics of the file; the importer recognizes the file type and also determines whether byte-swapping is necessary. STAR-CCM+ currently does not support two-dimensional, unstructured (tetrahedral), or formatted (ASCII) Plot3D files. Double precision files are now supported.

When you import a Plot3D file, STAR-CCM+ can read boundary definitions from a `.fvbnd` file at the same time, if the `.fvbnd` file has the same base file name and is in the same directory as the Plot3D file.

Fluent Format

Fluent format files can have the following extensions:

- `.cas` - Fluent case file
- `.msh` - Gambit
- `.grd` - TGrid

These files have the same format internally, but can be written in text or binary form. STAR-CCM+ automatically recognizes a file as being binary or text and imports it without requiring further action.

Also see the guidelines for Migrating from Fluent.

CGNS Format

STAR-CCM+ can read CFD General Notation System files with the following extensions:

- `.cgns`
- `.cga`
- `.cgh`

STAR-CCM+ can import CGNS meshes containing tetrahedra, prisms, hexahedra, polyhedra, and pyramids. The import of CGNS files is intended for mesh only. STAR-CCM+ does not import any solution data that is available on the imported mesh. The import of CGNS files in STAR-CCM+ is based on the CGNS library version 3.2 (64bit).

Parts-Based Meshing

Parts-based meshing is a meshing strategy in which you create one or more mesh operations to define the steps that STAR-CCM+ performs to generate the volume mesh.

Mesh operations are actions that you use to:

- prepare the initial geometry
- surface wrap geometry
- generate surface and volume meshes
- Parts-based meshing detaches the meshing process from physics modeling and provides a flexible and repeatable sequence of mesh operations. The sequence of mesh operations is called the meshing pipeline.

Contents:

Preparing Geometry for Meshing
Surface Wrapping the Geometry
Generating Surface and Volume Mesh
Local Meshing
Creating a Parts-Based Mesh with Baffles
Mesh Operations
Extruding Surfaces from a Parts-Based Volume Mesh
Replacing a Part in a Parts-Based Volume Mesh
Mesh Descriptions
Custom Mesh Controls
Parts Meshes Continua

Preparing Geometry for Meshing

Before auto-meshing or wrapping geometry parts, perform mesh operations to prepare the geometry.

Before Preparing Geometry for Meshing:

1. Provide the geometry to use in the simulation. There are two options:
   - Import CAD data or surface mesh

     See Importing Geometry.
   - Create a model using 3D-CAD

     See Creating Geometry Parts from 3D-CAD.
2. If you create a model in 3D-CAD, convert the 3D-CAD model to geometry parts.
   a. Right-click the **Geometry** > **3D-CAD Models** > **[3D-CAD Model]** node and select **New Geometry Part**.

      The *Parts Creation Options* dialog appears.

b. Click **OK**.

To Prepare Geometry for Meshing:

1. Apply any necessary boolean operations.
   a. Right-click **Geometry** > **Operations** > **New** and select one of the following Boolean operations:
      - **Unite Parts**

        Merges multiple overlapping/coincident parts into a single part.
      - **Intersect Parts**

        Creates a part from the common volume of the selected parts.
      - **Subtract Parts**

        Removes the volume of one part from another part.

      See Performing Boolean Operations.
   b. Complete the properties of the [*Boolean operation*] dialog as required and click **OK**.

      Unexecuted Boolean operation nodes are now displayed under the **Geometry** > **Parts** and **Geometry** > **Operations** managers.
   c. Right-click the **Operations** > **[Boolean operation]** node and select **Execute**.
2. If there are holes or openings in the geometry, perform the **Fill Holes** operation before extracting the volume.
   a. Right-click the **Geometry** > **Operations** node and select **New** > **Fill holes**.
   b. Complete the properties of the **Fill Holes** operation dialog as required and click **OK**.

      Unexecuted **Fill Holes** nodes are now displayed under the **Geometry** > **Parts** and **Geometry** > **Operations** managers.
   c. Right-click the **Operations** > **Fill Holes** node and select **Execute**.

      See Fill Holes Operations for details of the operation properties.
3. If the imported geometry part represents the solid surrounding a fluid, and the fluid is the part being modeled, use the **Extract Volume** operation to create the Fluid part.

   See Extract Volume.
4. If faces of adjacent parts are in contact, use the **Imprint** operation to ensure that common surfaces of each part retain their own faces. A conformal mesh can still be formed between them.

   See the Imprint Parts Operation.
5. Assign parts to regions.

   Note: When you assign parts to regions and part surfaces to boundaries, the perimeters of the boundaries are marked internally as features for the remesher to preserve. Assigning parts to regions after an automated mesh operation means that a complete remesh is required, since the additional edges that are marked as features change the starting surface.

   See Assigning Parts to Regions.
6. Set boundary types for each boundary.

   See Setting Types.

At the end of this task, you have a geometry that is ready for surface and volume meshing. However, if the geometry has multiple defects or unintentional gaps, you can use the Surface Wrapper before generating the surface and volume mesh.

Contents:

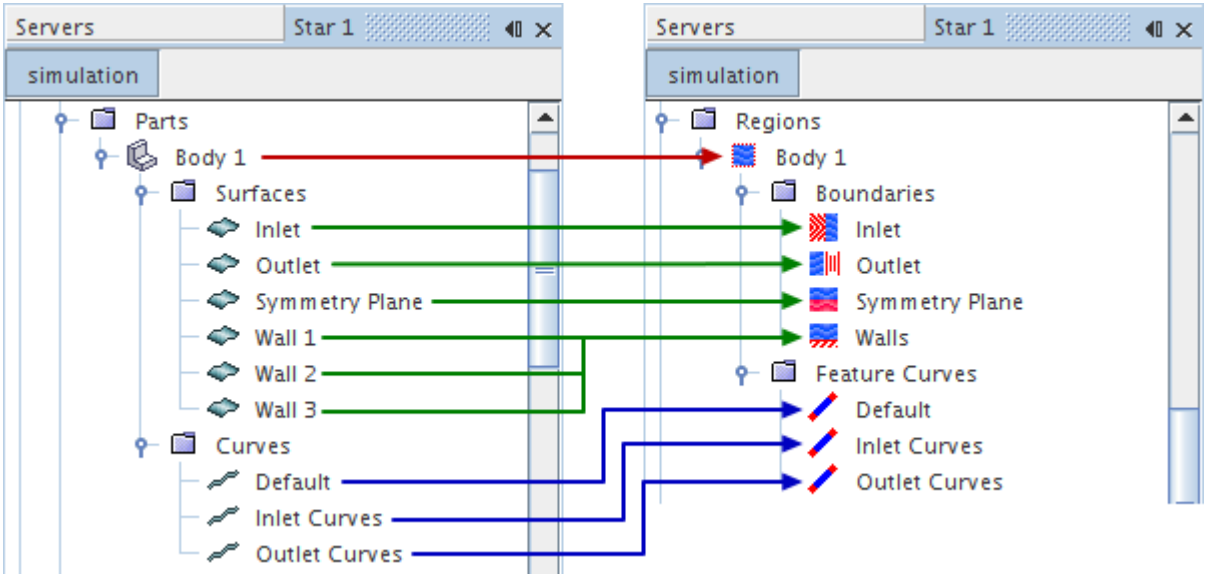
Assigning Parts To Regions

Assigning Parts To Regions

Assigning parts to regions creates a relationship between the meshed part and the physics region.

This action allows STAR-CCM+ to convert the volume mesh to a finite volume representation that the solver can use to obtain a solution.

You can assign part-level objects (parts, part surfaces, and part curves) to their equivalent region-level objects (regions, boundaries, and feature curves). From parts to regions, these assignments do not need to be one-to-one. That is, you can assign multiple part-level objects to the same region-level objects. The following diagram illustrates the connection between parts and regions.



However, you cannot assign multiple region-level objects to a single part-level object. If you reassign any of these objects to a new destination, it is automatically removed from its original assignment.

Note: Part contacts only become interfaces when you assign all contacting parts in the same operation.

The following methods are available for assigning parts to regions:

Contents:

Using the Assign Parts to Regions Dialog
Assign Parts to Regions Dialog
Using the Parts Context Menu

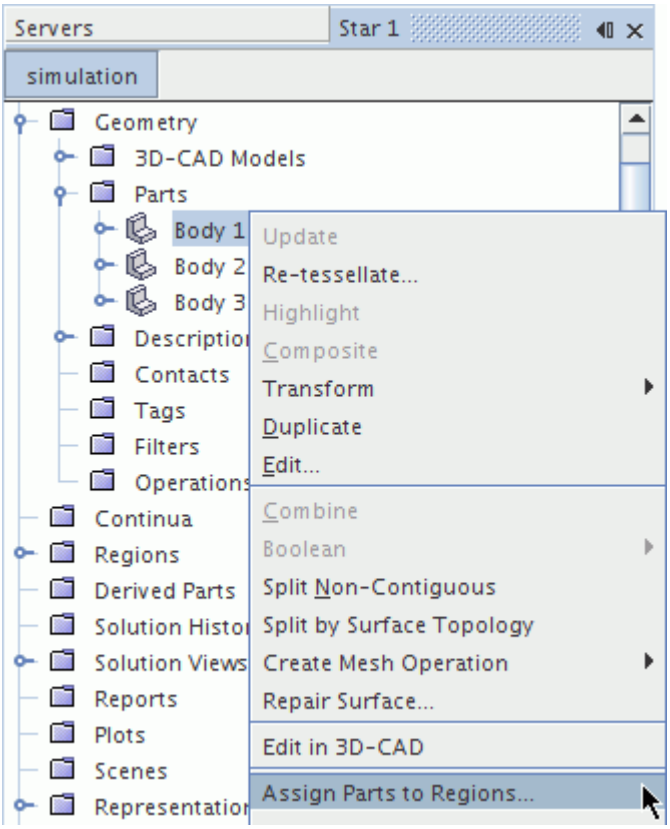
Using the Part Level Properties
Using the Region Level Properties

Using the Assign Parts to Regions Dialog

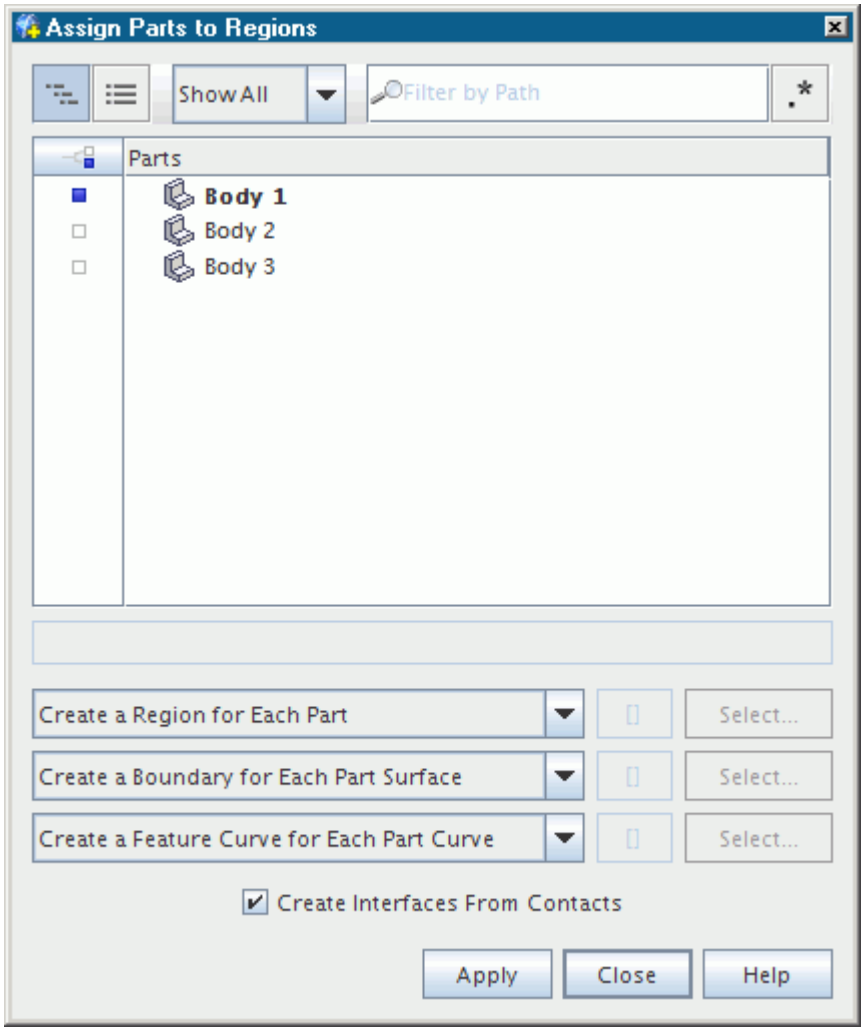
This dialog assigns several part-level objects to region-level objects in one action. This dialog is the recommended method of assigning parts to regions.

To assign parts to regions using the *Assign Parts to Regions* dialog:

1. Right-click a **Geometry** > **Parts** > **Part** node and select **Assign Parts to Regions**.



2. In upper-half of the *Assign Parts to Regions* dialog, select the parts that you want to assign to regions.
3. In the lower part of the *Assign Parts to Regions* dialog, set the properties to define how the parts are assigned to regions. See Assign Parts to Regions Dialog.



4. Click **Apply** and then **Close** to exit the dialog.

Assign Parts to Regions Dialog

First drop-down menu	**Create a region for Each Part**	Creates and assigns different regions to each selected part.
	Assign All Parts to an Existing Region	Assigns all the selected parts to an existing region. Selecting this option activates the option to select the existing region.
	Create One Region for All Parts	Creates and assigns one region to all the selected parts. Selecting this option allows you to name the region. The default name is `Region`.
Second drop-down menu	**Create a Boundary for Each Part Surface**	Creates and assigns different boundaries to each part surface.

	Create a Boundary for Each Part	Creates and assigns one boundary to all parts surfaces within each part.
	Create One Boundary for All Part Surfaces	Assigns all the part surfaces to one boundary. Selecting this option allows you to name the boundary. The default name is `Boundary`.
Third drop-down menu	**Create a Feature Curve for Each Part Curve**	Creates and assigns different feature curves to each part curve.
	Create One Feature Curve for All Part Curves	Assigns all selected part curves to one feature curve. Selecting this option allows you to name the feature curve. The default name is `Feature Curve`.
Create Interfaces from Contacts	**Activated**	Automatically generates interfaces from part contacts. When using this option, the valid settings for **Region Mode** and **Boundary Mode** depend on how you created the part contacts (see Table of Valid Methods and Options).
	Deactivated	Does not create any interfaces from part contacts.

Using the Parts Context Menu

The parts context menu allows you to assign a single part-level object to a region-level object.

The following capabilities are available with the context menu:

- Assigning Part-Level Objects to Existing Region-Level Objects
- Creating and Assigning Part-Level Objects to New Region-Level Objects

Assigning Part-Level Objects to Existing Region-Level Objects

You can use the context menu to assign a part-level object to a previously created region-level object.

The following section assigns a part to an existing region. You can use a similar procedure for assigning part surfaces to boundaries and part curves to feature curves.

Note: To assign multiple parts to an existing region, you must first create a composite part. To assign multiple part surfaces to boundaries or part curves to feature curves, you can multi-select the objects.

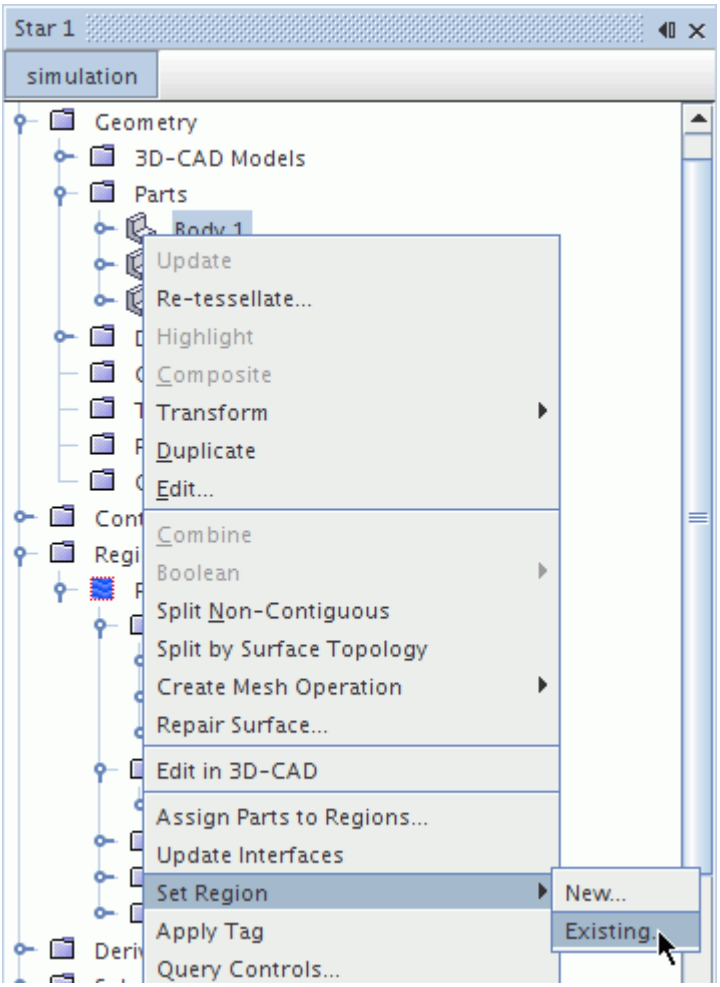
Setting a part to a region with this method assigns all part surfaces with the following conventions:

- If the name of a part surface matches the name of an existing boundary, this method assigns the part surface that boundary.
- If a boundary has the name, `Default`, this method assigns all other part surface to this boundary.
- If the `Default` boundary does not exist, this method assigns all other parts surfaces to the boundary with the lowest index. You can identify the index number of each boundary in the *Properties* window.

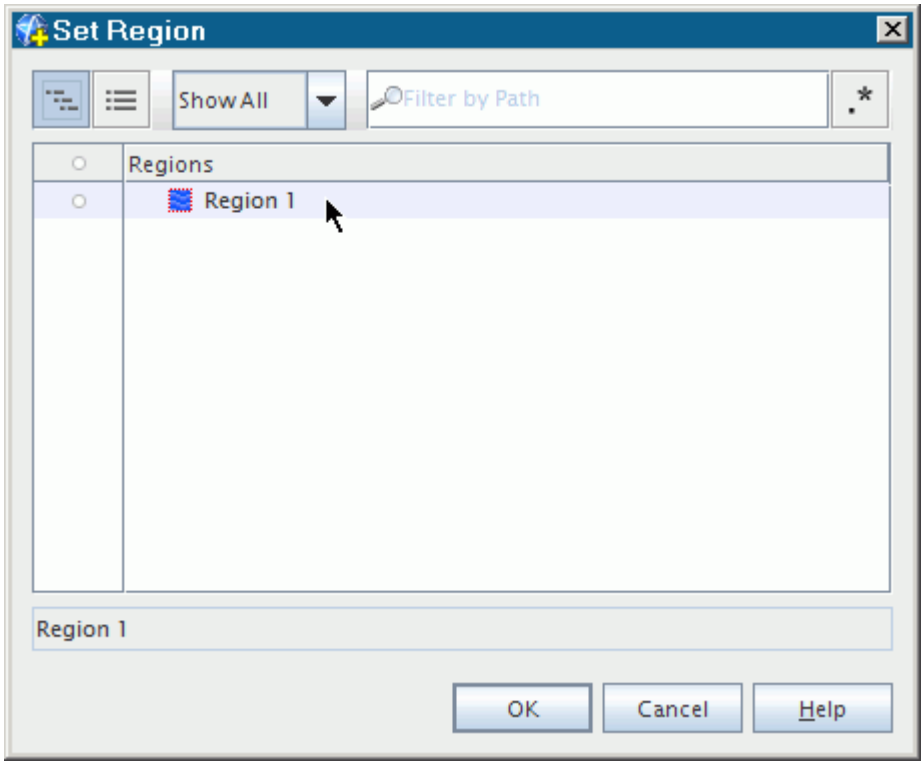
Note: A part-level object cannot belong to multiple region-level objects. Therefore, if you reassign a part to a different region, it is automatically removed from the original assignment.

To assign a part to a region with the context menu:

1. Make sure that a region exists.
2. Right-click a part and select **Set Region** > **Existing**.



3. Make a selection in the object selector. See [The Object Selection Dialog](#).



Creating and Assigning Part-Level Objects to New Region-Level Objects

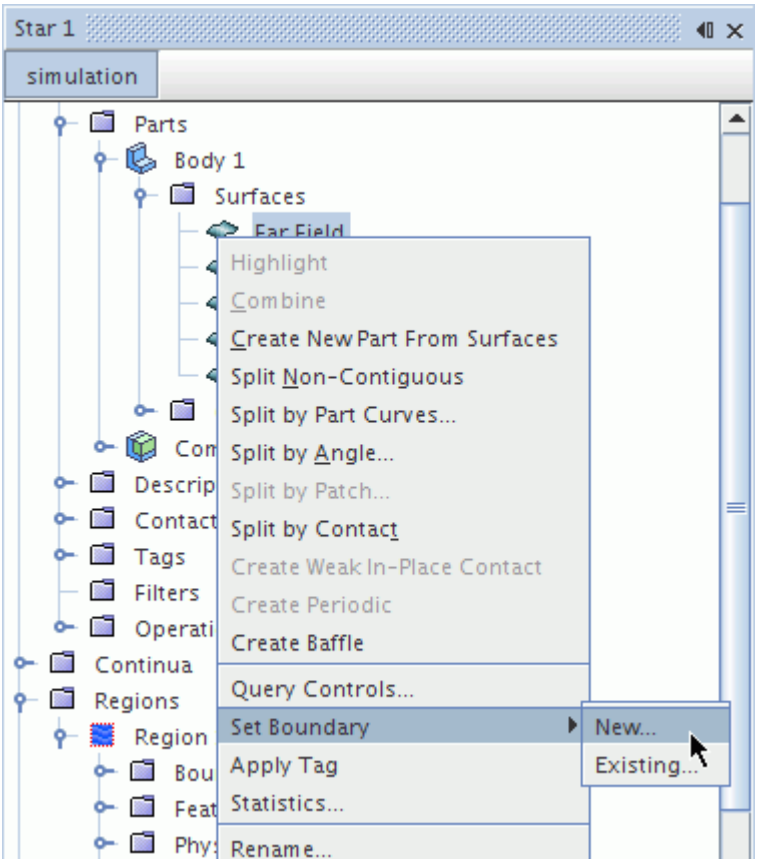
You can use the context menu to create and assign part-level objects to region-level objects. This facility is only available for part surfaces and part curves.

This action creates and assigns a boundary with the same name as the part surface. If you select multiple part surfaces, the new boundary name is based on the keyword **Boundary** (for example, Boundary 1, Boundary 2).

The following section creates and assigns a part surface to a new boundary. You can use a similar procedure for assigning part curves to feature curves.

To assign a part surface to a new boundary with the context menu:

- Right-click the required part surface and select **Set Boundary** > **New**.



You can also multi-select multiple boundaries.

Using the Part Level Properties

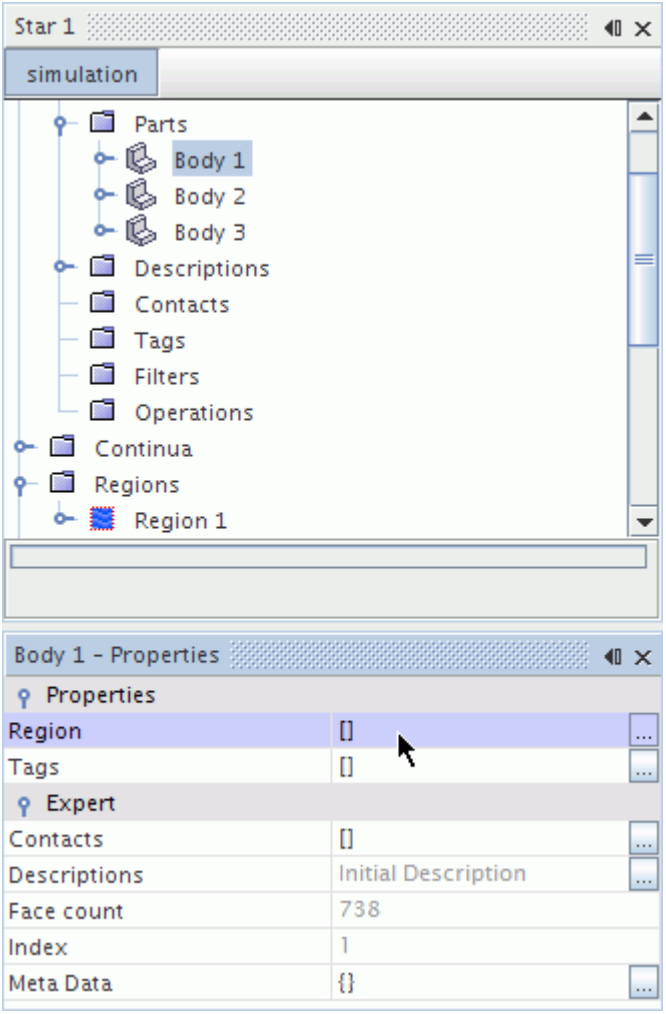
The part-level properties can assign the selected part-level object to an existing region-level object only.

The following section assigns a part to an existing region. You can use a similar procedure for assigning part surfaces to boundaries and part curves to feature curves. However, you can only assign part surfaces and part curves once you have assigned their parent part to a region.

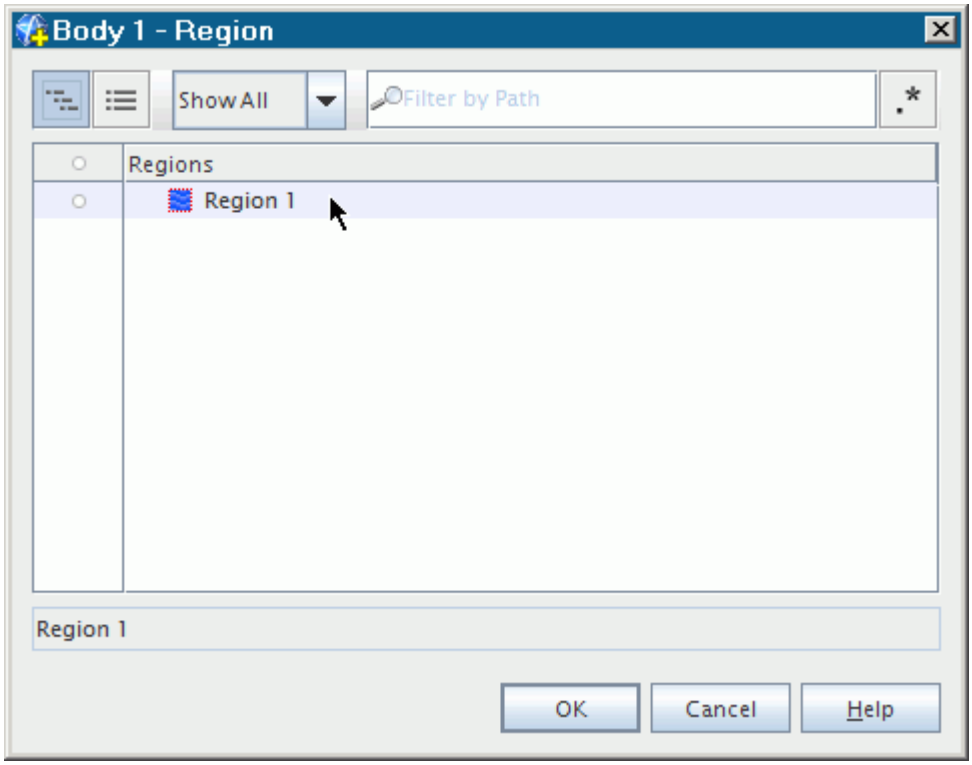
Note: A part-level object cannot belong to multiple region-level objects. Therefore, if you reassign a part to a different region, it is automatically removed from the original assignment.

To assign a part to an existing region with the part-level properties:

1. Select the required part node and click the right half of the *Region* property.



2. Make a selection in the object selector.



Using the Region Level Properties

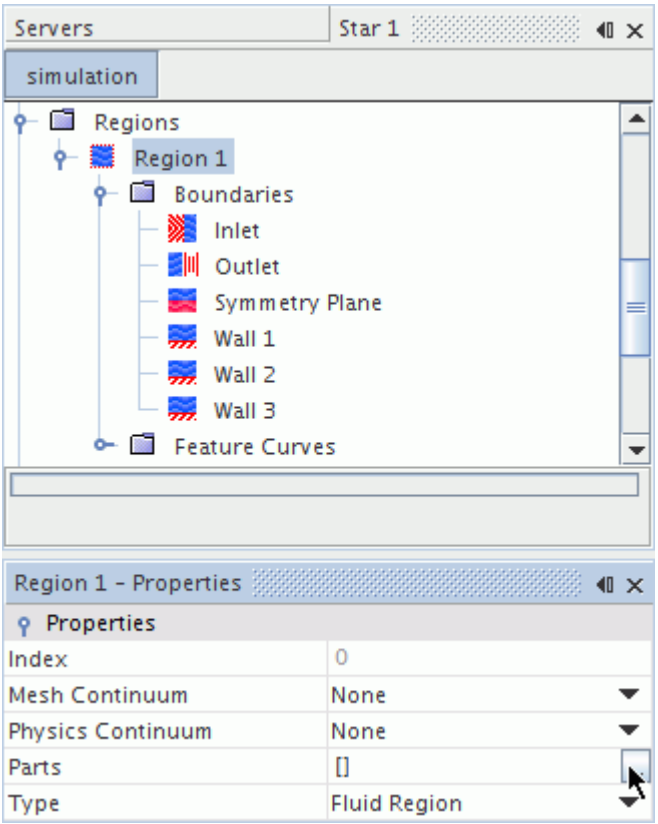
The region-level properties can assign the selected region-level object to a part-level object.

The following section assigns a region to a part. You can use a similar procedure for assigning boundaries to part surfaces and feature curves to part curves. However, you can only assign boundaries and feature curves once you have assigned their parent region to a part.

Note: A part-level object cannot belong to multiple region-level objects. Therefore, if you reassign a part to a different region, it is automatically removed from the original assignment.

To assign a region to a part with the region-level properties:

1. Select the required region and click the ellipsis **(...)** next to the *Parts* property.

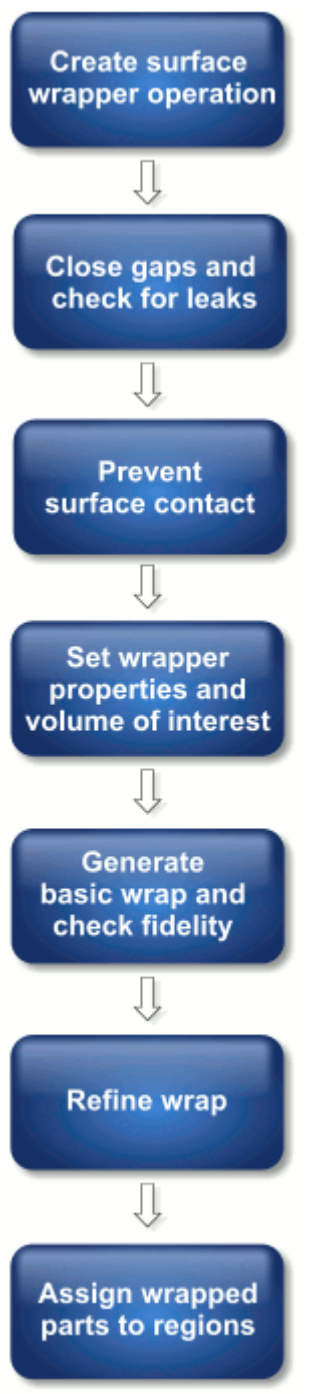


2. In the object selector, select the assigned part and click **OK**.

Surface Wrapping the Geometry

The Surface Wrapper generates a closed and manifold surface mesh from poor quality or complex CAD geometry.

The workflow to follow when surface wrapping geometry is displayed below. Click a step in the workflow to access more details about this step.



1. Creating the Surface Wrap Operation and Selecting the Parts to Wrap
2. Closing Gaps and Checking for Leaks
3. Preventing Surface Contact
4. Setting the Volume of Interest and Wrap Properties
5. Generating a Basic Wrap and Checking the Fidelity
6. Refining the Wrap
7. Assigning Wrapped Parts to Regions

At the end of this workflow, you have a water-tight geometry that is ready for surface and volume meshing.

Contents:

Creating the Surface Wrap Operation and Selecting the Parts to Wrap
Closing Gaps and Checking for Leaks
Preventing Surface Contact
Setting the Volume of Interest and Wrap Properties
Generating a Basic Wrap and Checking the Fidelity
Refining the Wrap
Assigning Wrapped Parts to Regions

See Also:

Surface Wrapper
Mesh Operations

Creating the Surface Wrap Operation and Selecting the Parts to Wrap

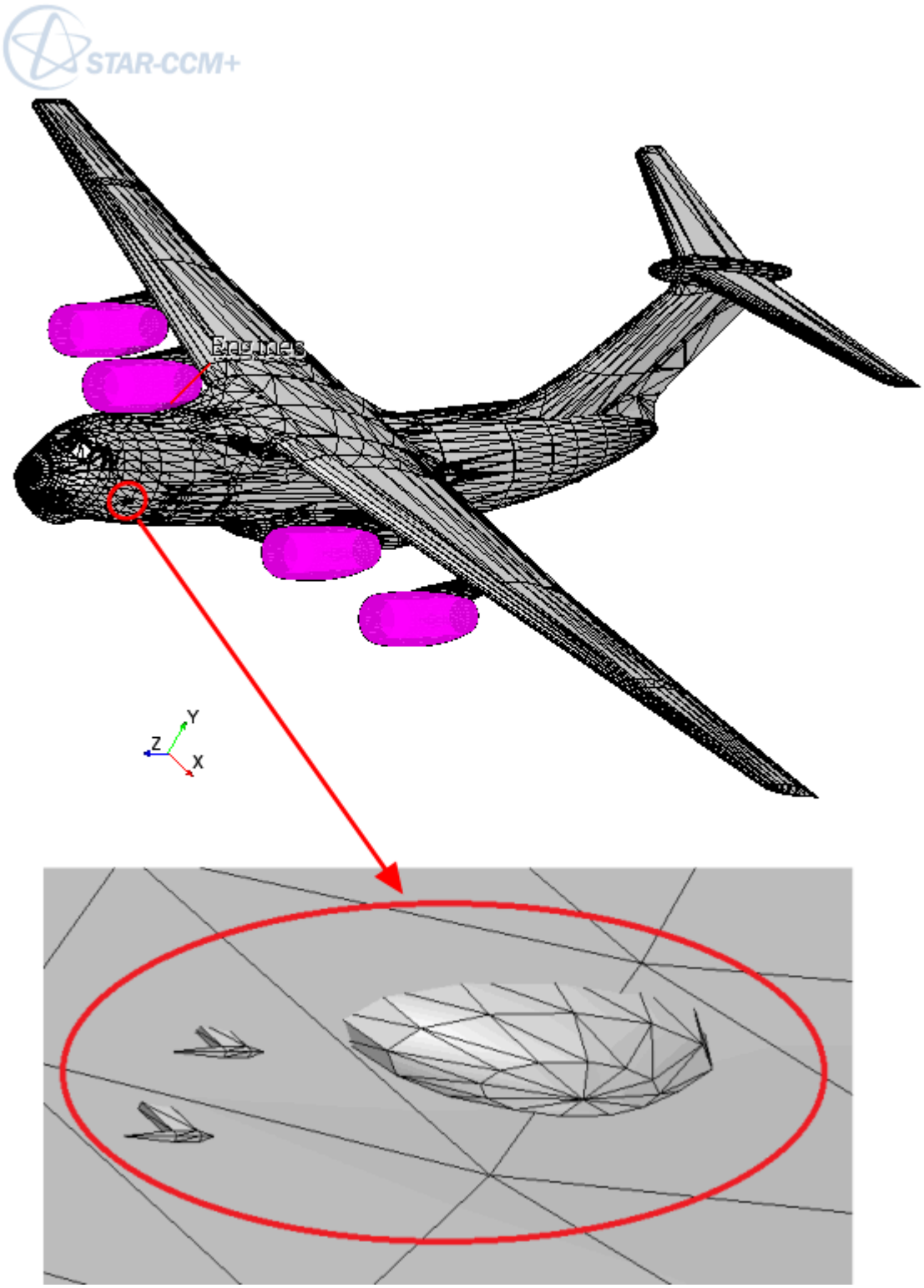
Create a Surface Wrapper operation and select the parts to wrap.

To create a **Surface Wrapper** mesh operation:

1. Right-click **Geometry** > **Operations** and select **New** > **Surface Preparation** > **Surface Wrapper**.
2. In the *Create Surface Wrapper Automated Mesh Operation* dialog, select the parts to wrap and click **OK**.

   Unexecuted **Surface Wrapper** nodes are now displayed under the **Geometry** > **Parts** and **Geometry** > **Operations** managers.
3. Select the **Default Controls** > **Base Size** and set an appropriate size for the wrap.

Some geometries only require surface wrapping for a portion of their surface. The surface wrapper allows you to exclude part surfaces that you consider sufficiently well tessellated for subsequent mesh operations. For example, the geometry below shows the original tessellated surface of an aircraft. The main aircraft body is closed and manifold, does not require surface wrapping, and contains many intricate details. However, the engines that are part of the same geometry, as highlighted below, require surface wrapping.



In cases such as these, wrapping the whole geometry is unnecessary and can defeature highly detailed part surfaces–unless the surface wrapper is excessively refined using custom controls and contact preventions. Instead, you can preserve specified input surfaces and perform partial wrapping.

4. To exclude specific part surfaces on a geometry from a Surface Wrapper operation and perform partial wrapping:
   a) Select the **Geometry** > **Operations** > **Surface Wrapper** node and activate the *Perform Partial Wrapping* option.

      A **Default Controls** > **Preserved Input Surfaces** node appears.
   b) Select the **Default Controls** > **Preserved Input Surfaces** node and set *Excluded Surfaces* to all part surfaces that you want to exclude from the wrap.

Closing Gaps and Checking for Leaks

Identify and close any obvious gaps in the geometry. Run the leak detection tool to identify less obvious gaps and holes.

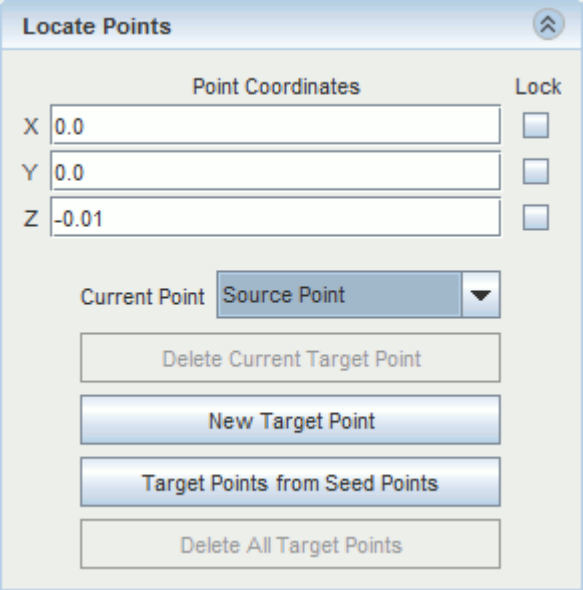
Two different tools are available to help close holes, namely the polygonal patch filler and the hole filler. The polygonal patch filler is a quick and easy way of closing arbitrary shaped holes which do not have a closed loop definition or are not planar. The process patches over the surface by creating faces that cover the hole area completely. The hole filler is a more exact method of filling well-defined holes by using a closed loop feature edge definition around the hole.

1. Open a scene. If you do not have a scene, create one.
2. Right-click **Geometry** > **Operations** > **Surface Geometry** and select **Run Leak Detection**.
3. In the *Graphics* scene, click (**Allow edges to be selected**) at the bottom of the screen to allow edge selection in the scene.
4. Patch obvious holes or gaps:
   a) If the hole is planar and has edges surrounding it, use the (**Fill holes using selected edges**) tool. See The Surface Repair Tool: Filling Holes.

      You can double-click an edge to select all other connected edges.
   b) If the hole is arbitrary, or is not planar, use the (**Fill Polygonal Patch**) tool. See Repair Tools Reference: Fill Polygona Patchl Tool.

      Click on a surface to start creating a patch. Continue clicking to form a closed loop. To finish, right-click while placing the final point.

   Note: A Tag appears on any new part surface that is created during gap closure. The tag differentiates the surfaces that gap closure creates from the pre-existing surfaces. See Tagging Simulation Objects.
5. Check the geometry for leaks:
   a) Click the *Locate Points* tab.
   b) Set *Current Point* to **Source Point**.

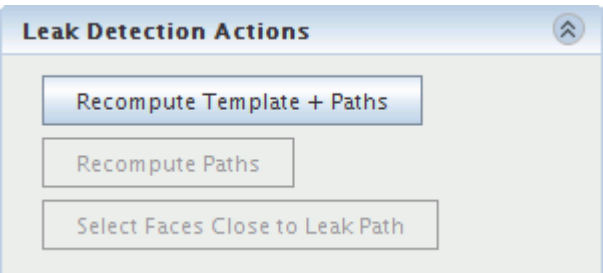


c) Set the source point location:
   - Define the point manually using the *X*, *Y*, and *Z* positions.
   - Define the point interactively using the point tool.

d) Set *Current Point* to **Target Point** and set the location as in the previous step.

e) If you require several target points, click **New Target Point** and define them.

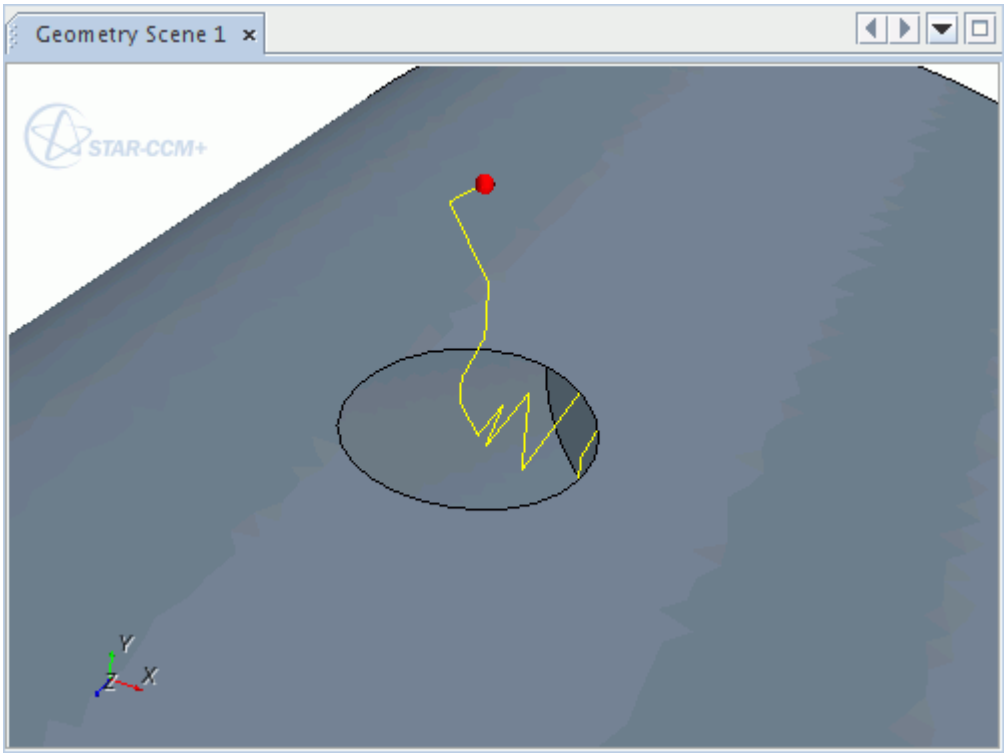
f) When you have defined all source and target points, click **Recompute Template and Paths**.



If there are no leaks, the *Output* window reports the following:

```
No path found between source and target points.
These points will be on different sides of the final wrapped
surface.
```

If a leak is identified, a path from the source to the target point appears in the *Graphics* window.

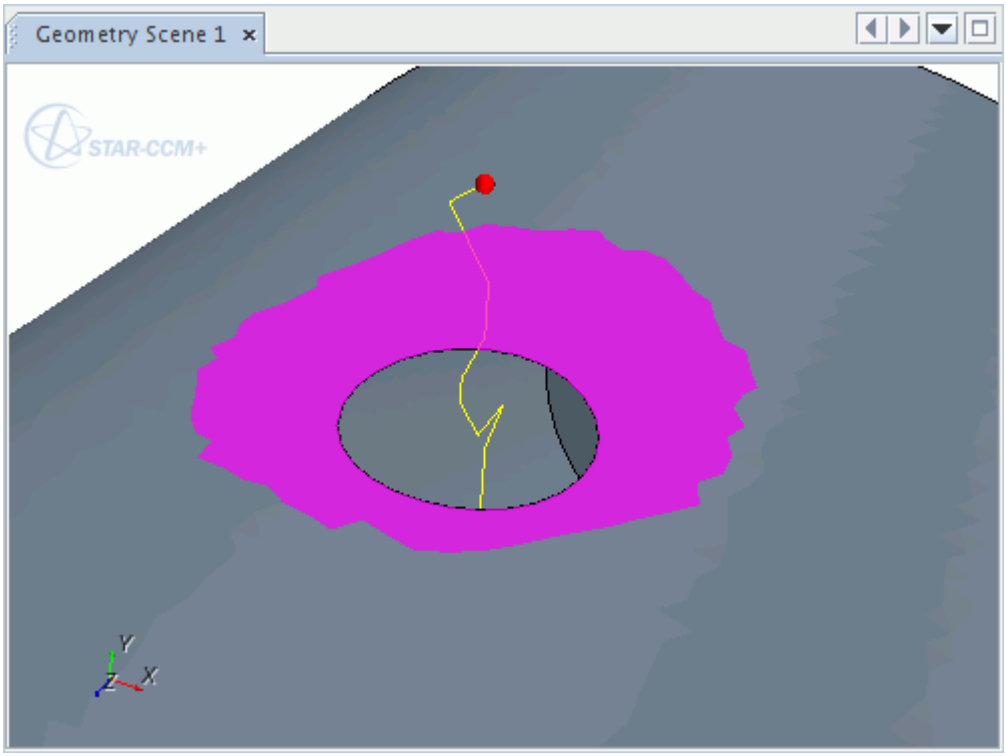


If multiple holes exist in the input surface, then a single target point only detects one of them. The addition of extra target points in different locations around the geometry or in different volumes helps detect holes that the wrapper did not fill. However, it is not guaranteed that adding a target point results in a new hole being found since the point location affects the computed path.

Click **New Target Point** to include extra target points. Each leak path to a different target point is shown in a different color, even if it passes through the same hole. The **Delete Current Target Point** button can be used to remove a pre-existing target point from the calculation.

Changing the location of any of source or target points, or adding or deleting target points requires the recalculation of the leak path. To recalculate the path, press the **Recompute Paths** button, which uses the previously defined template as the basis of the calculation.

g) Click **Select Faces Close to Leak Path** to highlight the faces that surround the hole.



h) Use the tools in the *Repair Surface* or *Organize* tab to fix any leaks that the tool detects.

6. Check for any remaining leaks:

   a) Click **Recompute Paths**.

   Depending on the number of holes that are filled and their size, it can be necessary to recalculate the template by clicking **Recompute Template + Paths**. If there is any doubt whether leaks still exist, follow this approach as the safest route for leak detection. At any time during the surface repair operation, you can return to the **Locate Points** mode dialog and reconfigure the source and/or target points to assist leak detection to find holes. Once all leak holes are closed, you can proceed to wrap the surface and then use the surface remesher to improve the resultant quality.

Preventing Surface Contact

Contact prevention ensures that selected close surfaces do not touch each other in the wrapped surface.

The following modes of contact prevention are available:

- One Group Contact Prevention
- Two Group Contact Prevention

Preventing Contact between One Group of Surfaces

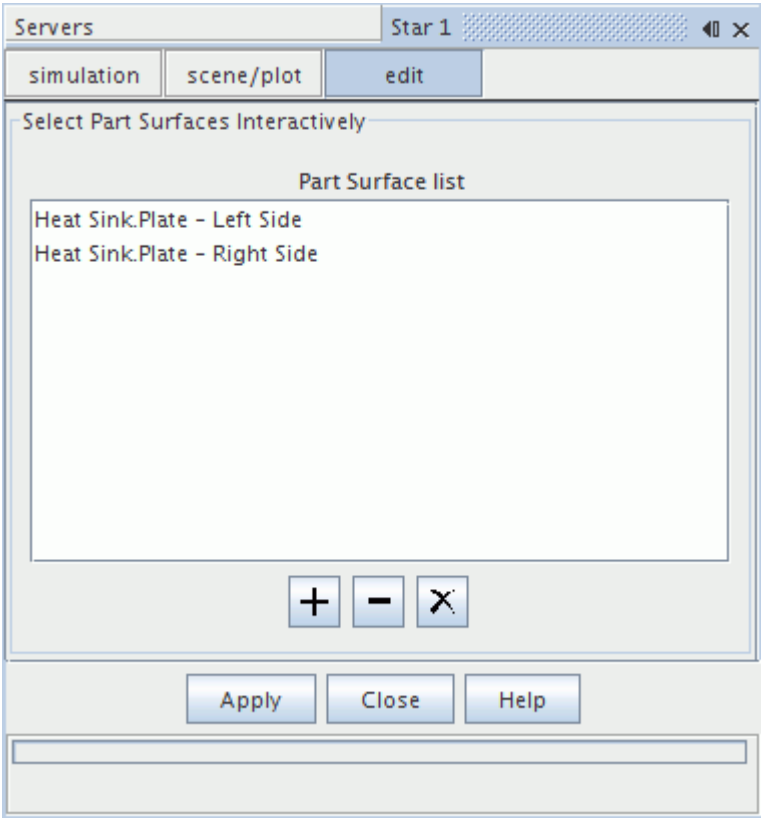
One group contact prevention surface makes sure that all surfaces within the group do not touch.

To prevent contact between surfaces in one group:

1. Right-click the **Operations** > **Surface Wrapper** > **Contact Prevention** node and select **One Group Contact Prevention Set**.

2. To apply the contact prevention to part surfaces:
   - If you want to define the surfaces manually, select the **Contact Prevention** > **Contact Prevention 1** node and set *Surfaces* to the required part surfaces.
   - If you want to define the surfaces interactively, right-click the **Contact Prevention** > **Contact Prevention 1** node, select the part surfaces with the *Graphics* window and selection buttons.

     This option is only available if you have a scene open.



3. Select the **Contact Prevention** > **Contact Prevention 1** node and set *Minimum Size* to the lowest edge size that the wrapper generates in the refinement.

   This value must be greater than 0.

Preventing Contact between Two Groups of Surfaces

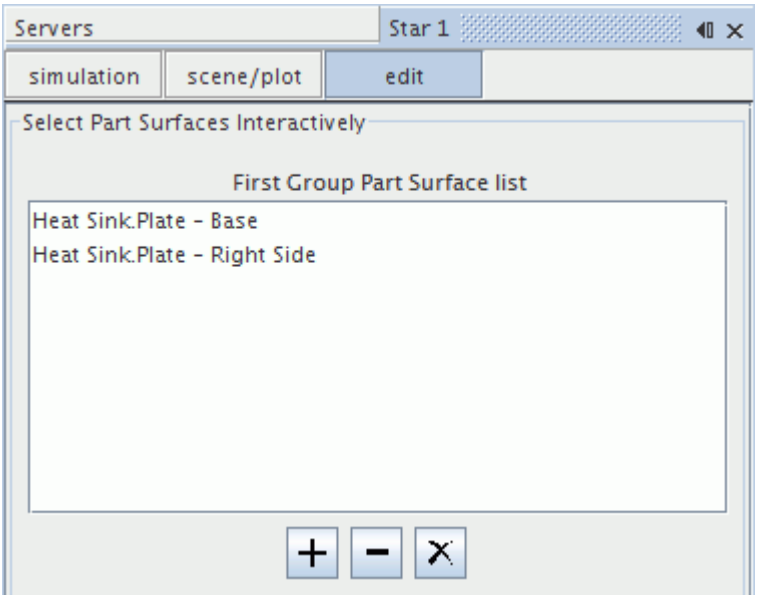
Two group contact prevention makes sure that surfaces between the two groups do not touch, although surfaces within groups can touch.

To prevent contact between two groups of surfaces:

1. Right-click the **Operations** > **Surface Wrapper** > **Contact Prevention** node and select **Two Group Contact Prevention Set**.
2. To apply the contact prevention to part surfaces:
   - If you want to define the surfaces manually, select the **Contact Prevention** > **Contact Prevention 1** node and set *Surface set 1* to the required part surfaces.

- If you want to define the surfaces interactively, right-click the **Contact Prevention** > **Contact Prevention 1** node and select the part surfaces with the *Graphics* window and selection buttons.

  This option is only available if you have a scene open.



3. Repeat the previous step for *Surface set 2*.
4. Select the **Contact Prevention** > **Contact Prevention 1** node and set *Minimum Size* to the lowest edge size that the wrapper generates in the refinement.

   This value must be greater than 0.

See Also:

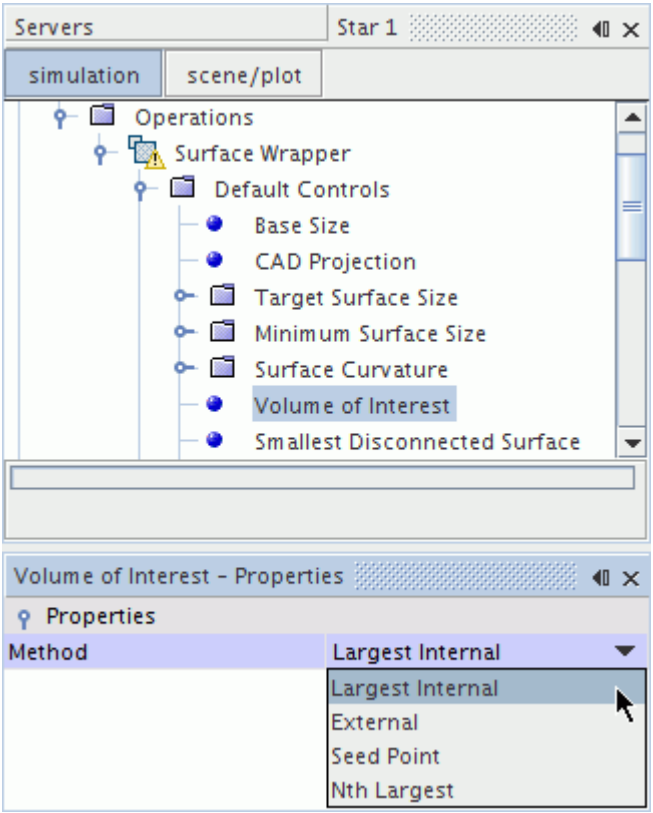
Surface Wrapper

Setting the Volume of Interest and Wrap Properties

The volume of interest control determines the volume that the surface wrapper extracts from the input parts. The wrapper properties control the wrap.

1. Set the volume of interest:
   a) Select the **Surface Wrapper** > **Default Controls** > **Volume of Interest** node and set *Method* to one of the following properties:
      - **Largest Internal** — Specifies that the wrapper extracts the largest internal, self-contained volume from the input geometry surfaces.
      - **External** — Specifies that the surface wrapper extracts the outer surfaces of all parts, that is, the external surfaces.
      - **Seed Point** — Specifies that a user-defined location determines the surface that the wrapper extracts. The resultant mesh is the surface that surrounds this location. See Step 2.

- **Nth Largest** — Specifies that the wrapper extracts a particular surface according to volume size. The surfaces are ranked from largest to smallest, and the user-defined Volume Number determines the output surface. See Step 3.



2. If you set *Method* to **Seed Point**, create a seed point:

   Seed points define a location within the volume of interest for the seed point volume extraction method.

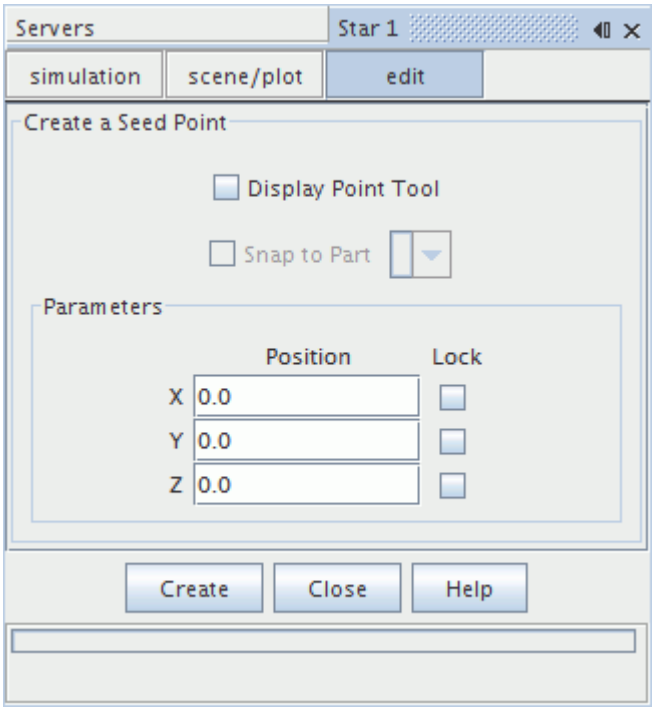
   a) Make sure that you have a scene open.

      If you do not have a scene open, you can only define the seed point using its properties.

   b) Right-click the **Surface Wrapper** > **Volume of Interest** > **Seed Points** node and select **New**.

   c) Define the position of the seed point:

      - Use the *X*, *Y*, and *Z* positions to specify an exact position.
      - Activate the **Display Point Tool** checkbox to define the position interactively in the *Graphics* window.



d) Click **Create** and then **Close**.

3. If you set *Method* to **Nth Largest**, define the volume number:

   The volume number property defines the volume of interest, ranked from largest to smallest, for the nth largest volume extraction method.

   a) Select **Surface Wrapper** > **Default Controls** > **Volume of Interest** > **Volume Number** node, and then set *Volume Number* to define the volume of interest.

      For example, setting *Volume Number* to `1` extracts the largest volume and `2` extracts the second largest volume.

4. Select the **Operations** > **Surface Wrapper** node and set the properties in the *Properties* window. See Surface Wrapper Properties.

Generating a Basic Wrap and Checking the Fidelity

Generate a basic wrap and check the fidelity.

To execute the wrapper:

1. Right-click the **Operations** > **Surface Wrapper** node and select **Execute**.

   Typically, you wrap the surface with coarser settings and proceed to refine the settings.

   The geometry is wrapped.

2. Check the fidelity of the wrap:

   The Deviation Distance field function, which is available when the Surface Wrapper model is active, displays the amount by which the surface wrap deviates from the original geometry.

   a) Create a scalar scene and display the wrapped part.

   b) Set the field function for the scalar scene to **DeviationDistance.Root**.

   c) Expand the **Tools** > **Field Functions** node.

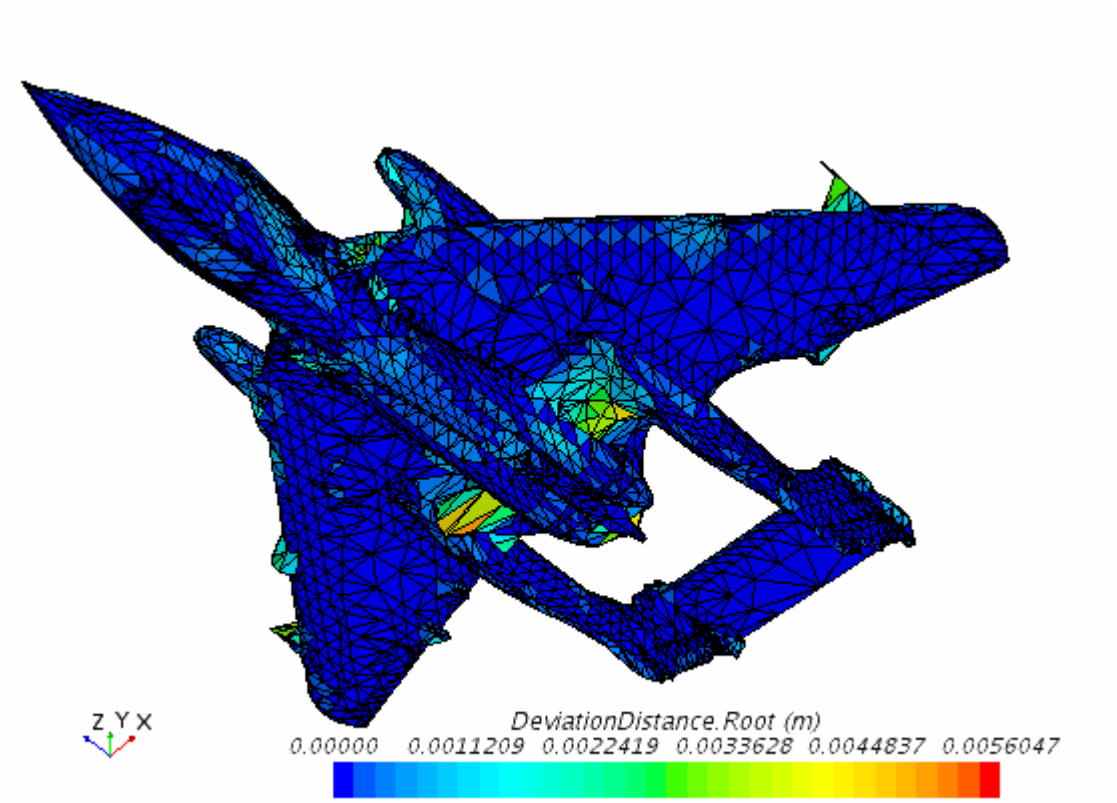
d) Select the **DeviationDistance.Root** and click ... next to the *Part* property.

e) In the *DeviationDistance.Root - Parts* dialog, select the parts that correspond to the original geometry.

If you want to compare the wrap with specific part surfaces, expand the parts nodes and select the desired part surfaces.

f) Click **OK**.

STAR-CCM+ calculates the deviation distance displays this in the scalar scene. The calculation can take time, depending on the complexity of the geometry.



Refining the Wrap

Typically you wrap geometry using coarse settings before refining the wrap. Local wrapping speeds up the re-execution of the surface wrapper after local changes are made to the input geometry or wrapper settings. With this feature, STAR-CCM+ only wraps surfaces within a defined location and leaves other surfaces unchanged.

With local wrapping, you set limits on where you want wrapping to occur by defining local extents. For surface wrapping, there is only one type of local extent—the volume extent, which contains shape parts that define a volume. All part surfaces within the volume are candidates for local wrapping.

Local surface wrapping is only supported with parts-based meshing.

To refine the wrap using local surface wrapping:

1. Select the **Operations** > **Surface Wrapper** node and activate the **Perform Local Surface Wrapping** option.

   The **Local Extents** node is added to the automated mesh operation.
2. Create extents that define the limits of surface wrapping:
   a) Create a shape part that encloses the part surfaces that you want to wrap locally. See Creating Simple Shape Parts.

      You can create multiple shape parts to cover different areas on the geometry.
   b) Right-click the **Surface Wrapper** > **Local Extents** node and select **New**.

      A **Volume Extent** node is added to the **Local Extents** node.
   c) Select the **Volume Extent** node and set *Parts* to the shape part that you created earlier.

      You can select multiple shape parts for the extent.
3. Set the local volume of interest.

   By default, the **Auto volume of interest** option is activated which can successfully determine the volume of interest that is based on the initial global wrapped output.
4. Update the geometry or wrapper settings.
5. To re-execute the surface wrapper, right-click **Operations** > **Surface Wrapper** and select **Execute**.

Local changes are wrapped in the locations that are specified by the extents. The local wrap is then stitched to the global wrap. Seed points are automatically detected and reported in the *Output* window.

If the inputs inside the extents have changed too much, or if the extent is too close to the wrapped surface, automatic mode can fail. If automatic mode fails, deactivate the **Auto VOI seed point** option and set the seed points manually.

When adding volume of interest seed points manually, a general guideline is that the number of seed points must be at least the number of disjointed volume segments you expect to get in a global wrap. Insufficient number of seed points can result in missing regions and failure to stitch the local wrap back to the global wrap. Incorrectly placed seed points, or seed points that are placed too close to either the geometry to be wrapped or the extents, can result in failures.

When setting the seed point manually, you are advised to have a scene open and position the seed using the scene. See Setting the Volume of Interest and Wrap Properties.

With local surface wrapping activated, you can perform leak detection locally as well. See Leak Detection.

Assigning Wrapped Parts to Regions

There are three different procedures to assign wrapped parts to regions, depending on what you want to achieve.

To assign wrapped parts to regions:

1. If you want to create interfaces from contacts between input parts and the output surface wrapper part, execute the **Surface Wrapper** operation with **Create Weak Contacts with input parts** activated before moving on to assign the surface wrapper part to a region:
   a) In the *Surface Wrapper - Properties* window, activate **Create Weak Contacts**.

      To select this option, make sure that the **Per-Part Meshing** option in the *Surface Wrapper - Properties* window is deactivated.

b) Right click the **Geometry** > **Operations** > **Surface Wrapper** node and select **Execute**.

In-place contacts are formed between Part Surfaces.

c) Assign the surface wrapper part to a region. See Assigning Parts to Regions.

The contacts are converted to interfaces.

2. If you do not want to create interfaces from contacts between input parts and the output surface wrapper part, you can:
   - Execute the surface wrapper operation with **Create Weak Contacts** deactivated in the *Surface Wrapper - Properties* window.
   - Assign parts to regions before executing the surface wrapper operation.

     If you do not need to create interfaces, you can execute the Surface Wrapper (with **Create Weak Contacts** deactivated) or Assign the surface wrapper part to a region in either order. If **Create Weak Contacts** is activated, assign parts to regions before executing the Surface Wrapper, otherwise interfaces are created. See Assigning Parts to Regions.
3. If you activate the **Per-Part Meshing** option in the *Surface Wrapper - Properties* window, no output parts form when you execute the **Surface Wrapper** operation. Instead, a new description node appears under the **Descriptions** manager. This description is a new representation of the input part.

Generating Surface and Volume Mesh

Meshing parts generates the surface and volume discretization of the imported CAD geometry. The solvers use this discretization to obtain a physical solution.

Before Generating Surface and Volume Mesh:

1. Complete Preparing Geometry for Meshing.

   See Preparing Geometry for Meshing.
2. If the CAD geometry is poor quality or complex, surface wrap the geometry.

   See Surface Wrapping the Geometry.
3. If part contacts occur between two part surfaces in the geometry and have not been defined automatically, create them now:

   a. To browse or find part contacts automatically, multi-select the required parts, right-click, and choose **Browse Part/Part Contacts**, or **Find Part/Part Contacts.**

   b. To create part contacts manually, multi-select and right-click the parts in contact, then choose **Create Part/Part Contact**.

   See Generating Part Contact Data.

STAR-CCM+ aims to generate a conformal mesh at part contacts. Each part contact becomes an interface in the physics region.

To Generate Surface and Volume Mesh:

1. Create an automated mesh operation:

   a. Right-click the **Geometry** > **Operations** manager node and select **New** > **Mesh** > **Automated Mesh**.

   The *Create Automated Mesh Operation* dialog appears.

b. Select the parts to mesh, then activate the required Meshers and click **OK**.

An unexecuted Automated Mesh node is displayed under the **Geometry** > **Operations** manager.

2. Within the **Automated Mesh** > **Default Controls** node, review the properties that are exposed by each mesher and set values appropriate for your simulation.

For example, you generally want to set the base size for all meshers. To do this, select **Default Controls** > **Base Size** and set *Value* to an appropriate size.

3. If you want to modify meshing behavior in any part of the model, add custom controls on the **Automated Mesh** > **Custom Controls** node.

Some common examples are:

- Deactivate prism layers on interfaces.
- Refine the mesh in chosen locations.
- Define precise size controls on some surfaces.

See Creating And Defining Custom Controls

4. Execute surface meshers.

a. Right-click the **Operations** > **Automated Mesh** node and select **Execute Surface Meshers**.

b. To display the surface mesh, right-click an empty area within a mesh scene and select **Apply representation** > **Mesh Scene 1**.

See What Is a Scene?.

Note: Assigning parts to regions after an automated mesh operation is executed, creates the need for a complete remesh, even if a mesh exists. When you assign parts to regions and part surfaces to boundaries, the perimeters of the boundaries are marked internally as features for the remesher to preserve. Marking additional edges as features, changes the starting surface, and therefore requires a complete remesh of parts.

5. Execute volume meshers.

- Right-click the **Operations** > **Automated Mesh** node and select **Execute**.

Note: If you create, update, or delete part contacts, mesh operations that include these part contacts are also updated. The updated mesh operation icons appear out of date (⚠) and must be re-executed.

Contents:

Selecting Meshers
Setting Default Controls
Creating and Defining Custom Controls
Meshing the Surface
Meshing the Volume

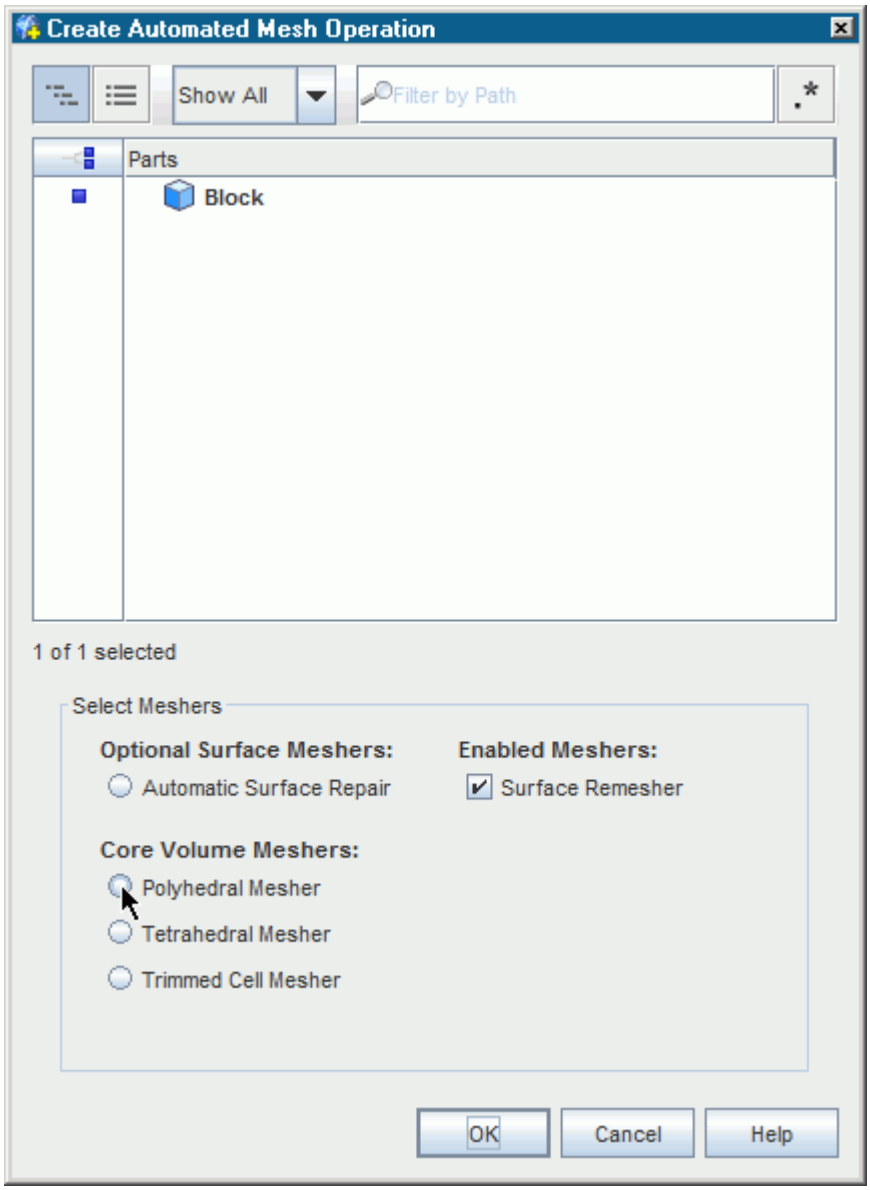
Selecting Meshers

The selected meshers determine the type of surface and volume discretization that you generate. For example, polyhedral, tetrahedral, or trimmed volume meshes.

Selecting Meshers for a New Automated Mesh Operation

To select meshers for a new automated mesh operation:

1. Create an automated mesh operation.
2. In the *Create Automated Mesh Operation* dialog, select the parts for the mesh and the required meshers.



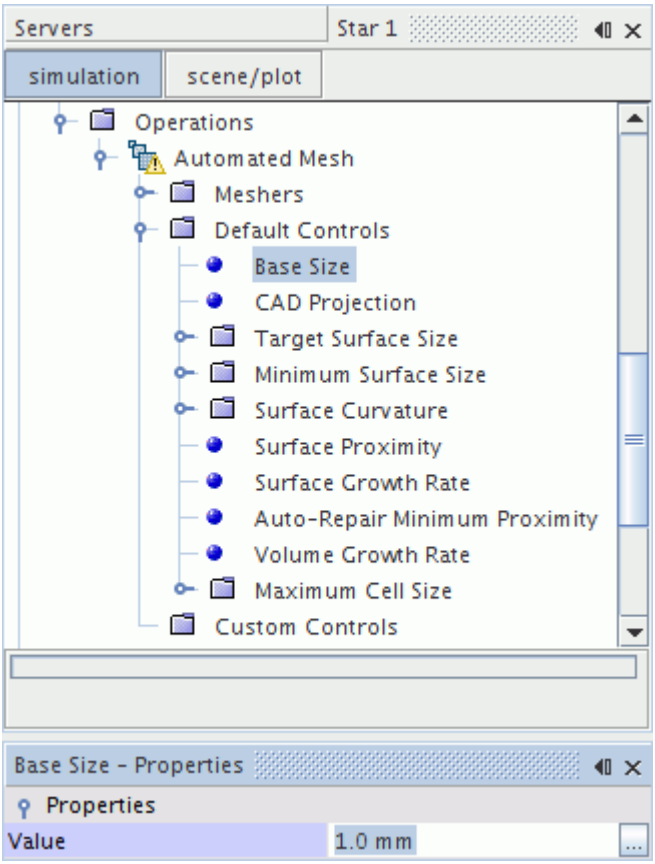
3. To deselect meshers, deactivate the checkboxes in the *Enabled Meshers* group box.

Setting Default Controls

The default mesh controls define the global surface and volume mesh behavior.

To set the default mesh controls:

1. Select a node under the **Operations** > **Automated Mesh** > **Default Controls** manager and set its properties.



See Also:

Extract Volume

Meshing the Surface

Meshing the Volume

Surface Mesher Controls and Values

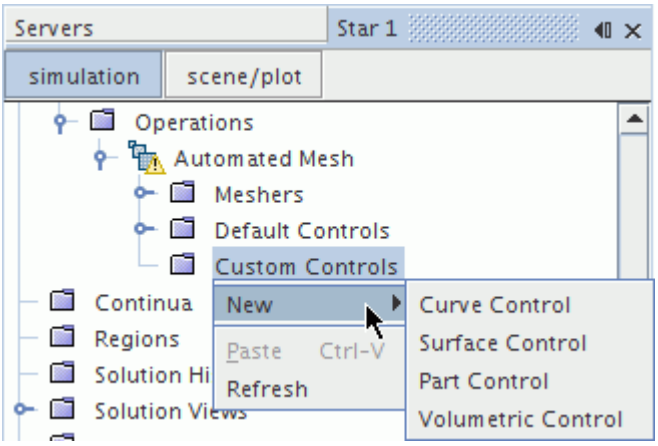
Volume Mesher Controls and Values

Creating and Defining Custom Controls

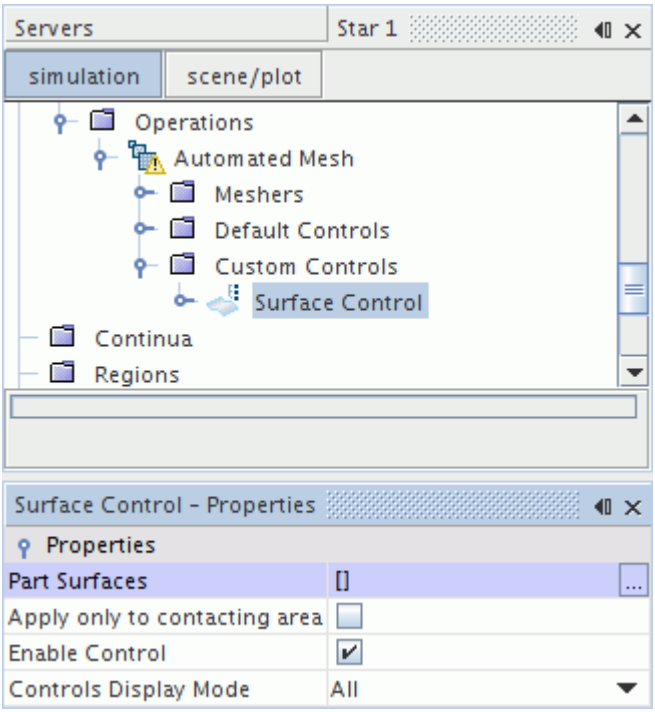
Custom controls refine or coarsen the surface and volume mesh for parts, part surfaces, part curves, and specified volumes.

To create and define custom controls:

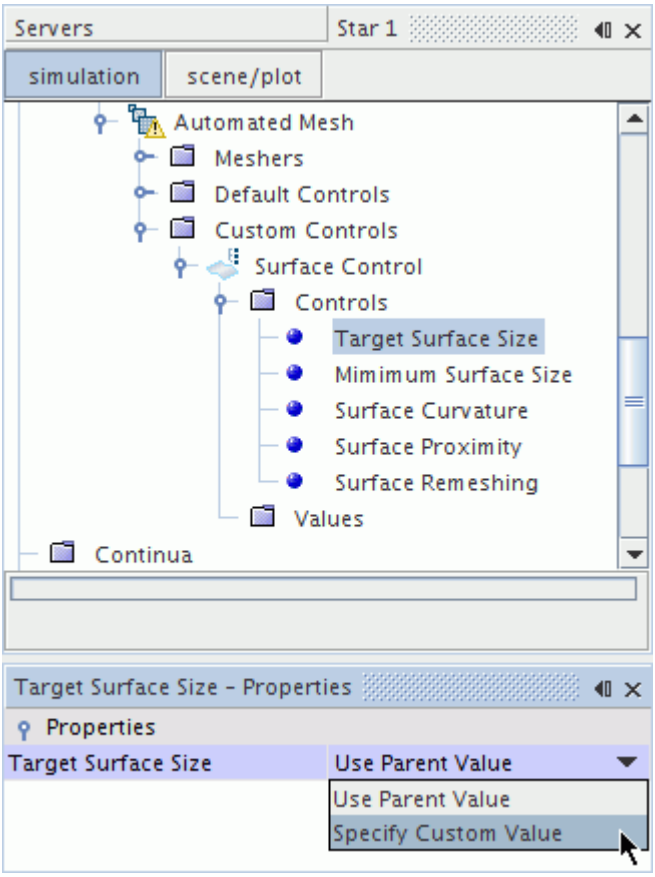
1. Right-click the **Operations** > **Automated Mesh** > **Custom Controls** node and select a control from the **New** submenu.



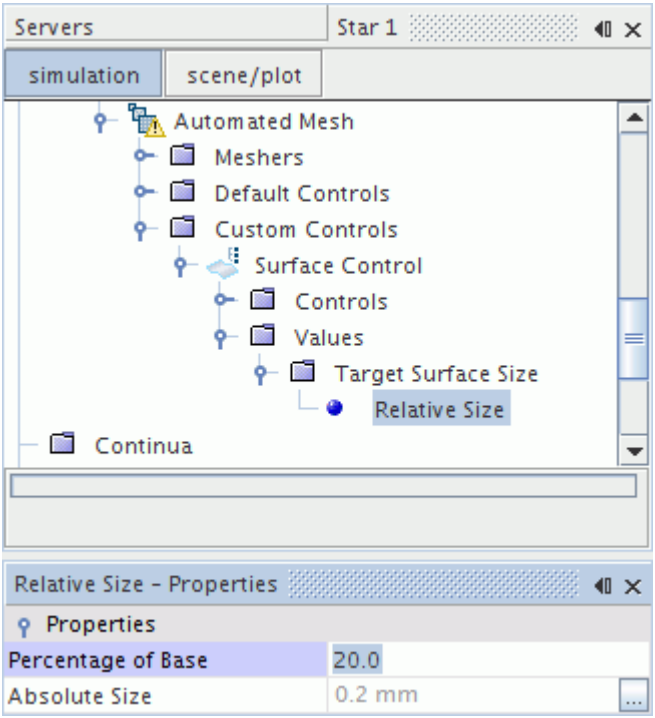
2. To define the applied part entities and control behavior, select the **Custom Controls** > **Control** node and set its properties.



3. To activate a control, select a node under the **Controls** manager and set its properties.



4. To define the custom control value, select a node under the **Values** manager and set its properties.



See Also:

Surface Mesher Controls and Values

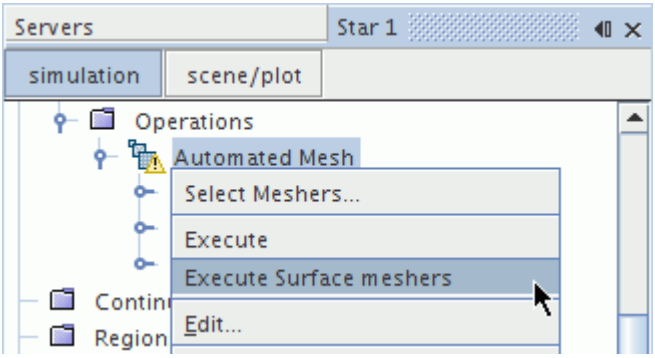
Volume Mesher Controls and Values

Meshing the Surface

Before meshing the volume, generate and check the surface mesh. The volume mesh quality is directly dependent on the surface mesh quality.

To mesh the surface:

1. Right-click the **Operations** > **Automated Mesh** node and select **Execute Surface Meshers**.



See Also:

Meshing the Volume

Extract Volume

Surface Meshers

Meshing the Volume

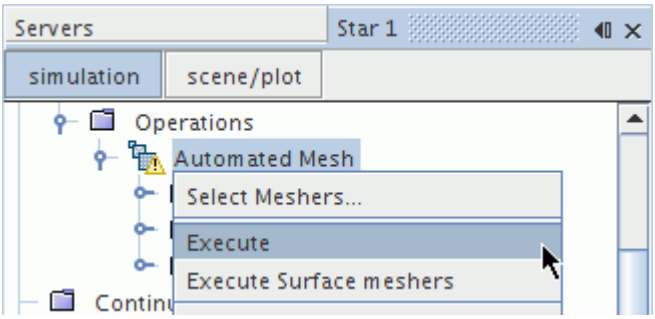
The volume meshers generate the discretized representation of the geometry part that the solvers use to obtain a solution.

Before meshing the volume:

- Assign the parts to regions.

To mesh the volume:

1. Right-click the **Operations** > **Automated Mesh** node and select **Execute**.



See Also:

Extract Volume

Surface Meshers

Local Meshing

Local meshing speeds up the re-execution of the surface remesher after local changes are made to the input geometry or mesh settings. With this feature, STAR-CCM+ only remeshes surfaces within a defined location and leaves other surfaces unchanged.

Depending on your meshing strategy, you can generate the first surface mesh without any local refinements, and then take advantage of local surface meshing to focus only on specific areas as you test different refinements. Similarly, if you expect to modify geometry at a particular location, you can surround the location with a volume that permits STAR-CCM+ to remesh the surfaces within it.

With local meshing, you set limits on where you want remeshing to occur by defining local extents. There are two type of extents that you can define:

- Surface Extent — contains the part surfaces that STAR-CCM+ can remesh
- Volume Extent — contains shape parts that define a volume. All part surfaces within the volume are candidates for remeshing

You can additionally extend the influence of the extents by setting inflation properties on each extent.

Local meshing is only supported with parts-based meshing, and only if the Surface Remesher is selected for the operation.

To use local meshing:

1. Create an automated mesh operation, apply the surface remesher, and generate the global surface mesh.
2. Select the automated mesh operation and activate the **Perform Local Surface Meshing** option.

   The **Local Extents** node is added to the automated mesh operation.
3. Create extents that define the limits of surface remeshing:
   - For surface extents:
     1. Right-click the **Local Extents** node and select **New** > **Surface Extent**.

        A **Surface Extent** node is added to the **Local Extents** node.

2. Select the **Surface Extent** node and set *Surfaces* to include all part surfaces that you want to remesh. You can select multiple part surfaces from different bodies.

- For volume extents:
  1. Create a shape part that encloses the part surfaces that you want to remesh. You can create multiple shape parts to cover different areas on the geometry.
  2. Right-click the **Local Extents** node and select **New** > **Volume Extent**.

     A **Volume Extent** node is added to the **Local Extents** node.
  3. Select the **Volume Extent** node and set *Parts* to the shape part you created earlier. You can select multiple shape parts.

4. Set the extent parameters. See Local Meshing Extents Properties.
5. Update the geometry or mesh settings. For example, you might create a custom mesh control on surfaces within an extent.
6. Re-execute the mesh pipeline.

Local Meshing Extents Properties

Use the inflation properties to extend the scope of the extent into surrounding surfaces. This extension can help to increase the quality of the remeshed surface, as any parts of surfaces that fall within the extended zone are included in the remeshing.

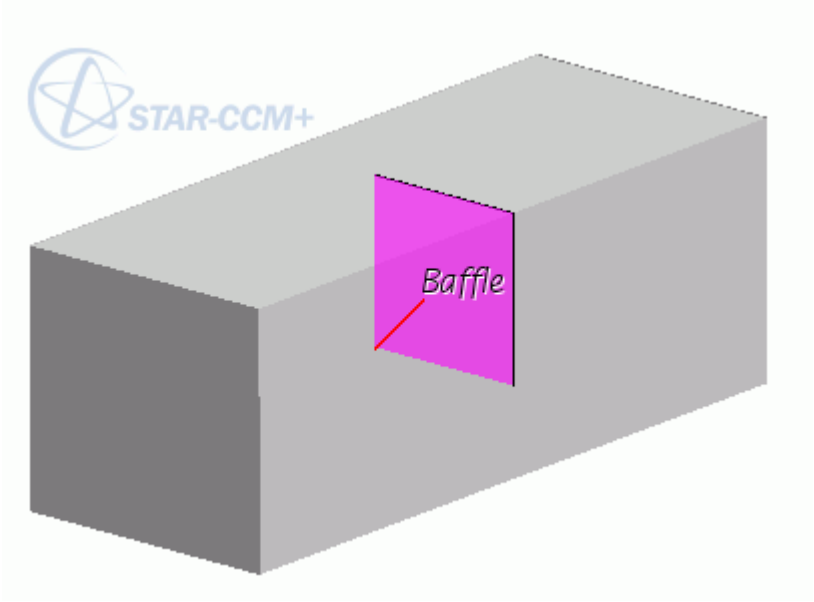
- *Inflation Distance* — specifies a distance beyond the extent. Any surfaces falling within this distance are included in remeshing, regardless of whether they are connected to the original surface parts covered by the extent.
- *Inflation Layers* — specifies a number of adjacent bands of triangles by which the extent can grow. For this option, STAR-CCM+ only enlarges the extent along connected surfaces. If there are two disconnected parts close to each other, the inflation layer does not grow from one part to the other.

Typically, you should aim to have 5 layers of triangles between the surface parts that are remeshed and the surfaces that remain unchanged. You can control this transition either through inflation distance, in which case the you have to roughly convert the existing mesh size to an absolute distance, or with inflation layers, for which you select the number of layers to grow.

Creating a Parts-Based Mesh with Baffles

Baffle contacts are zero-thickness surfaces that the meshers maintain as interfaces in the surface and volume meshes.

Defining baffle contacts maintains the shape of the zero-thickness surfaces in the mesh. The meshers assume that the baffles are two-sided interfaces and generate a conformal mesh on either side.



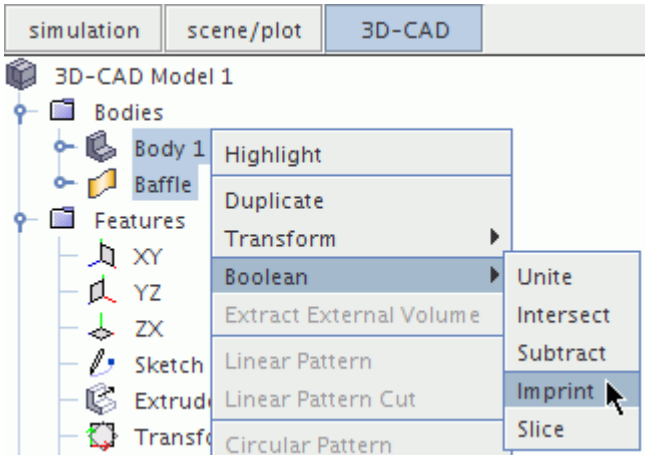
STAR-CCM+ only supports baffles that intersect a fluid volume. STAR-CCM+ does not currently support a line interface between a baffle and an adjacent solid part. Baffles are represented as interfaces within the volume mesh.

To Create a Parts-Based Mesh with Baffles:

1. In 3D-CAD, create solid and sheet bodies for the fluid volume and baffle parts, or import your geometry.

   See Using 3D-CAD or Importing Geometry.
2. If necessary, extract the fluid volume from the geometry.

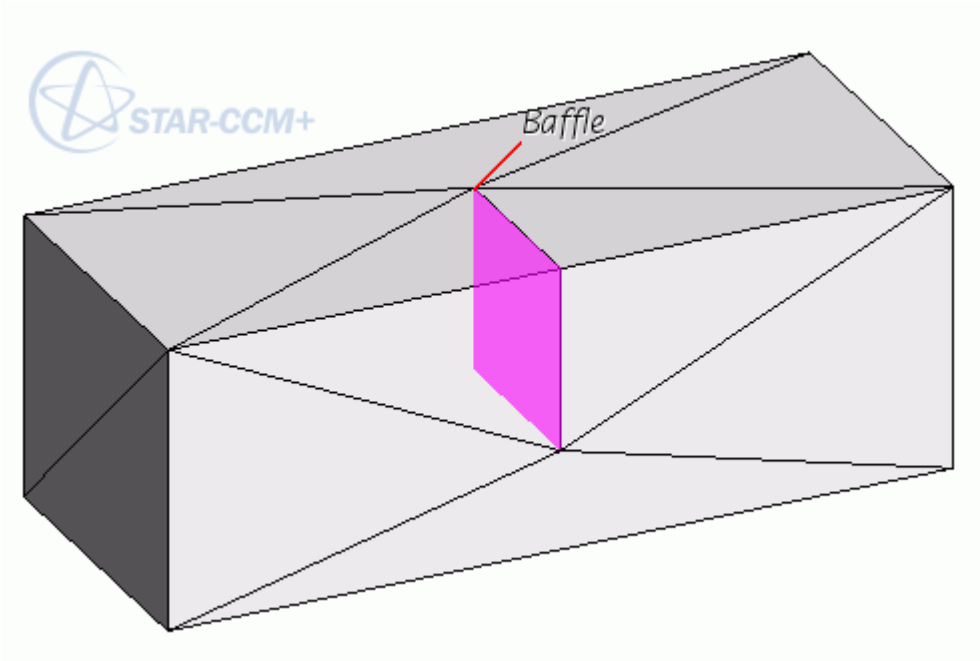
   See Internal Volume Extraction.
3. Imprint the baffles onto the fluid volume body.



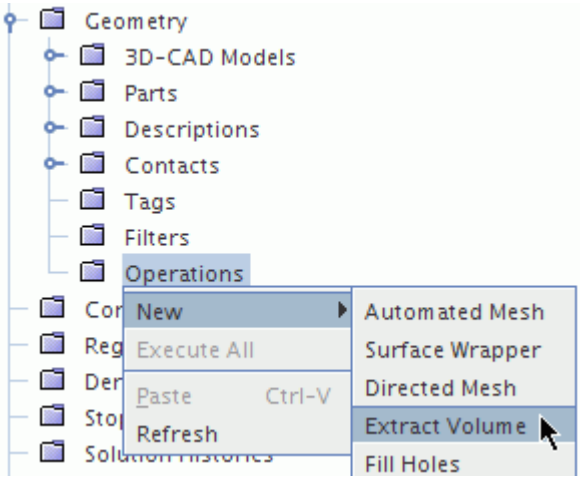
   See Imprinting Bodies.
4. Close 3D-CAD.
5. In STAR-CCM+, create geometry parts from the CAD parts. Ensure that **Create Part Contacts from Coincident Entities** is activated.

See Creating Geometry Parts from 3D-CAD.

The surface tesselation is conformal around the baffle edges.



6. Create and execute an extract volume operation for all parts.



See Extract Volume and Executing Operations.

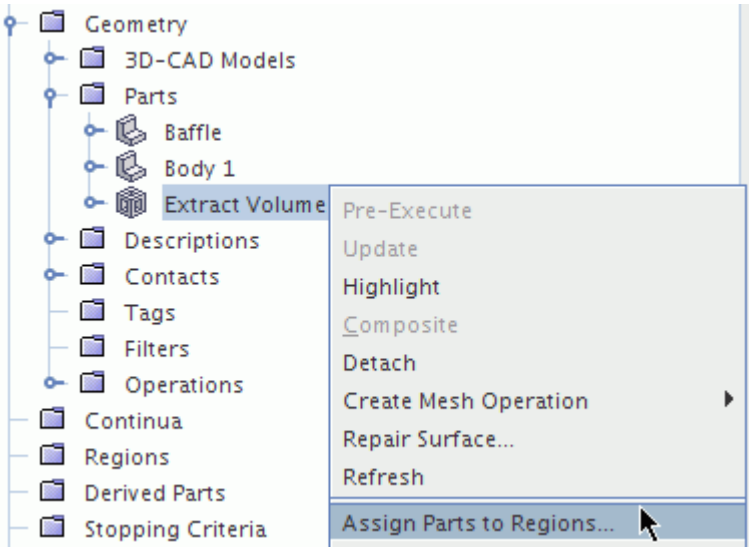
7. Create an automated mesh operation for the extract volume part.

   See Creating Mesh Operations and Selecting Meshers.

8. Set the mesh parameters that you require.

   See Setting Default Controls and Creating and Defining Custom Controls.

9. Assign the extract volume part to a region.

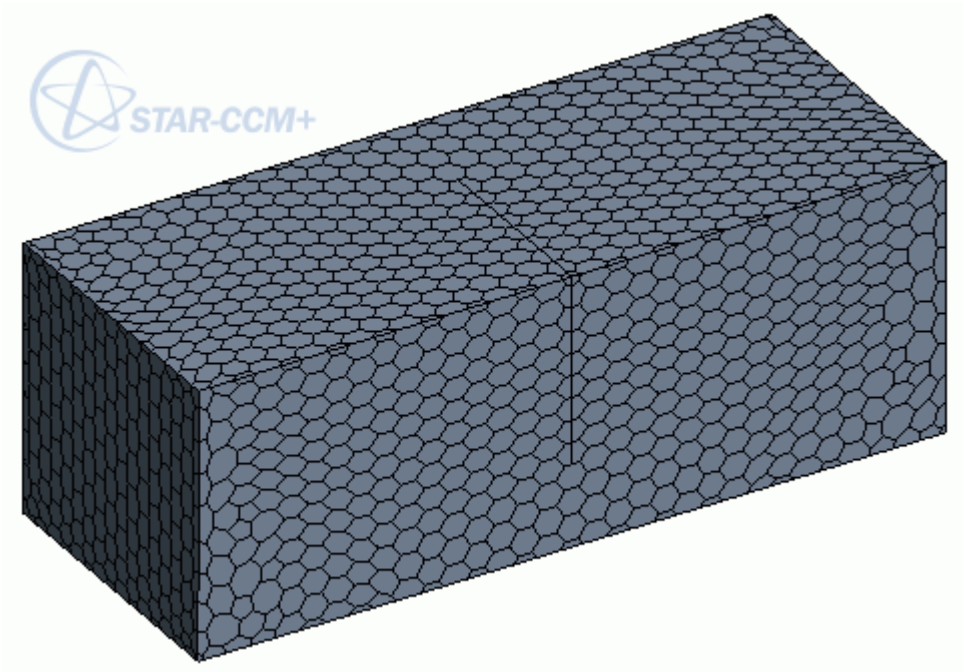


1. Set the following options:
   - Create one Region for All Parts
   - Create a Boundary for Each Part
2. Ensure that *Create Interfaces from Contacts* is activated.

See Assigning Parts to Regions.

10. Execute the automated mesh operation.

A conformal mesh is produced.



See Also:

Surface Wrapper with Internal Baffles

Mesh Operations

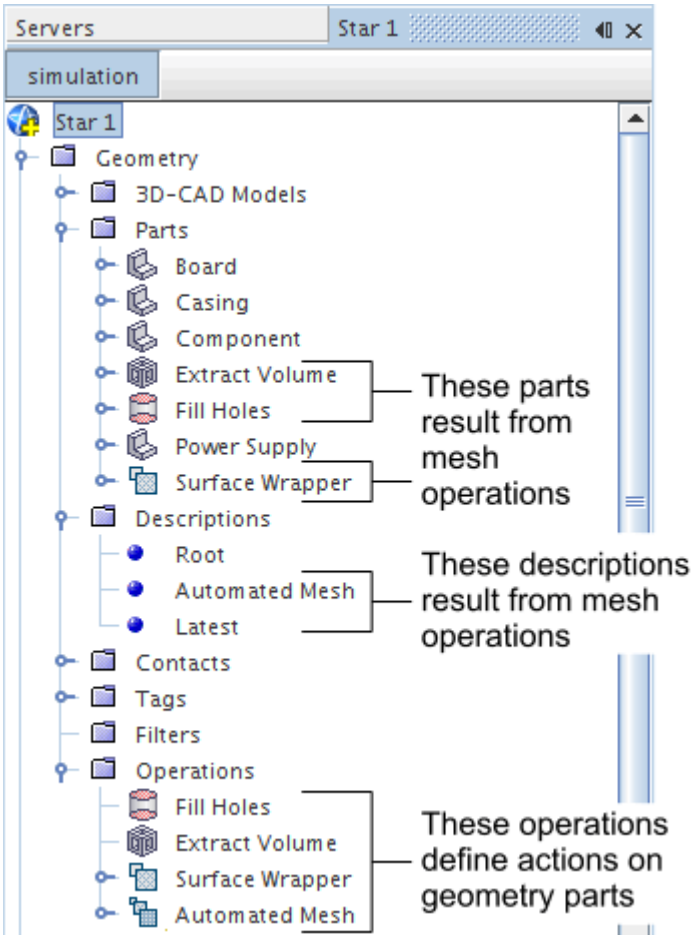
Mesh operations perform actions on geometry parts that you can easily repeat whenever their input parts or properties are changed.

The result of a mesh operation is a new geometry part or description, but the input parts are not changed.

The following mesh operations are available:

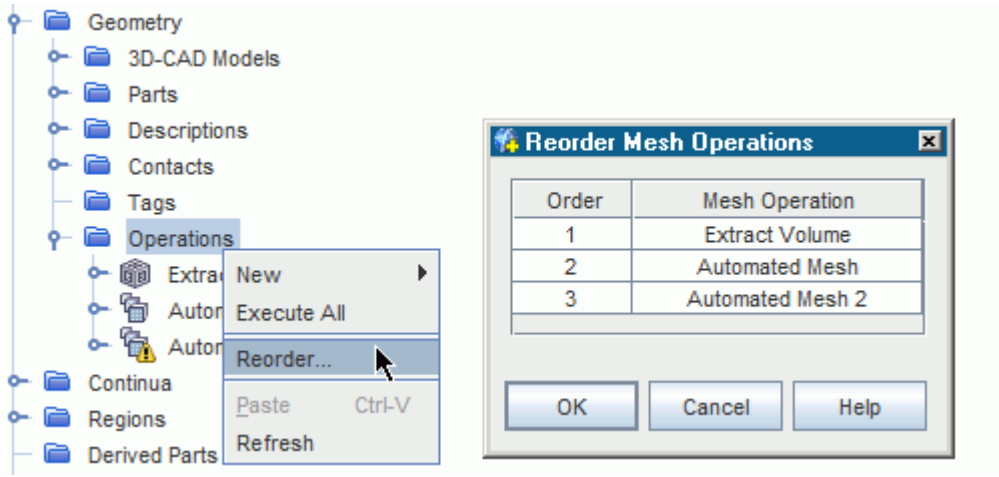
- Surface Preparation:
  - Surface Wrapper
  - Fill Holes
  - Extract Volume
  - Surface Extruder
  - Transform
  - Replace Part
  - Zip Edges
  - Weak Contact Creator
- Boolean:
  - Unite
  - Intersect
  - Subtract
  - Imprint
  - Bounded Shape
- Mesh:
  - Automated Mesh
  - Badge for 2D Meshing
  - Automated Mesh (2D)
  - Volume Extruder
  - Directed Mesh

The **Operations** node holds all mesh operations that are defined in the simulation. An example of how operations, mesh operation parts, and descriptions are linked is provided below.



To reorder the mesh operations:

1. Right-click the **Operations** node and select **Reorder**.
2. In the *Reorder Mesh Operations* dialog, drag the mesh operations into a different order then click **OK**.



Note: Reordering is not possible when the action results in the removal of a Mesh Operation Part from an operation. After reordering, operations are updated automatically, which can make some operations out of date.

Contents:

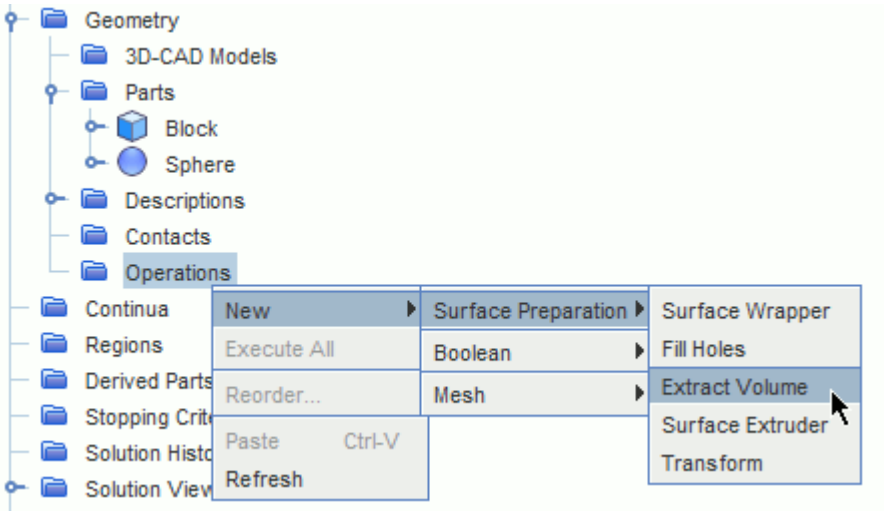
Creating Mesh Operations
Selecting Input Parts for Mesh Operations
Executing Operations
Copying Controls Between Mesh Operations
Mesh Operation Errors
Surface Extruder
Volume Extruder
Extract Volume
Zip Edges
Weak Contact Creator
Fill Holes
Unite Parts
Intersect Parts
Subtract Parts
Imprint Parts
Bounded Shapes
Transform Parts
Boolean Mesh Operation Properties
Intersection Curve Remeshing
Priority Indexing
Inherited Parts
Detaching Parts from Operations
Automated Mesh

Creating Mesh Operations

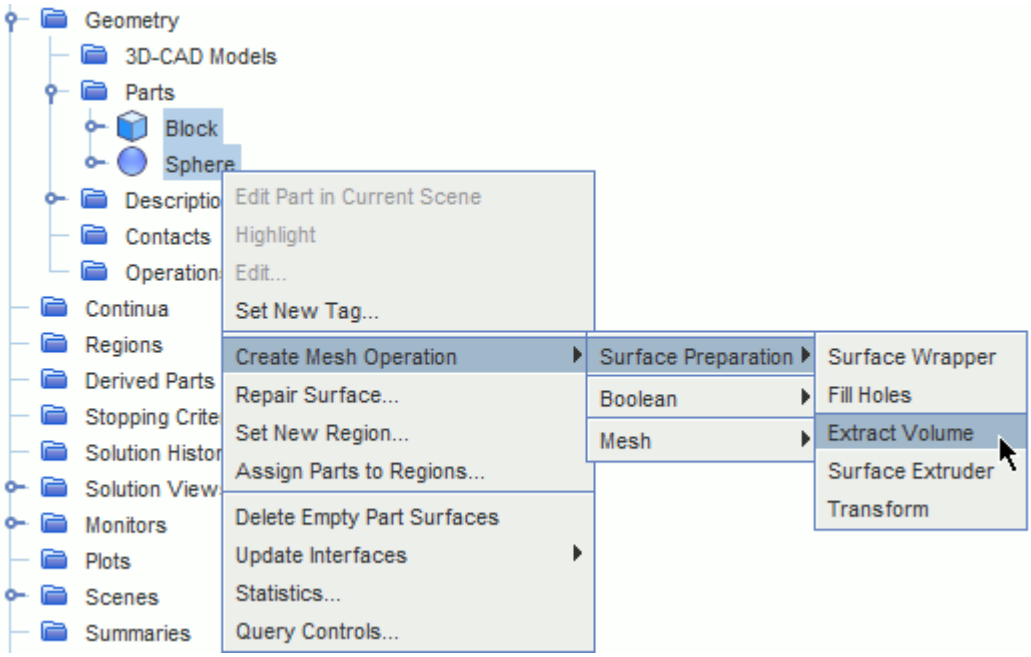
Mesh operations are actions that you can perform on the geometry in a sequential pipeline. This pipeline allows for flexible and repeatable meshing processes.

To create a mesh operation:

1. You can create a mesh operation using two different methods. Either:
   - Right-click the **Operations** node and select the required operation from the **New** > **[Group]** submenu.



- Select one or more geometry parts, right-click one of the selected parts and select the required operation from the **Create Mesh Operation** submenu.



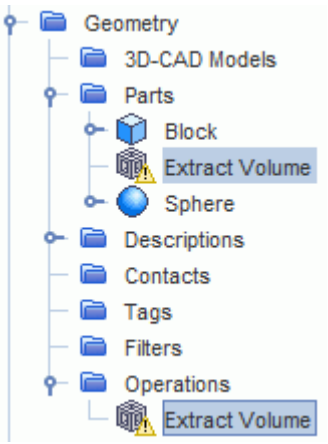
Mesh operations are grouped into three categories: **Surface Preparation**, **Boolean**, and **Mesh**.

When selecting the **Extract Volume** operation, if you have a scene open, a panel appears which allows you to define the properties. For all other operation selections, a dialog appears which allows you to select the input parts. Some mesh operation dialogs provide the option to activate *Execute Operation Upon Creation*. See Executing Operations On Creation. For the mesh operations that are listed below, the dialog also provides the option to *Perform CAD Boolean*.

- **Unite**
- **Subtract**
- **Intersect**

- **Imprint** (the option is called *Perform CAD Imprint*)

When the mesh operation is created, an operation node and a leaf-level part node appear in the tree.



Immediately upon creation, the mesh operation node is selected automatically.

2. Set the operation properties:
   - In the *Properties* window, set the properties that the operation requires.
   - For extract volume operations that are defined using the *Extract Volume* edit panel, you can also edit properties on the Extract Volume node afterwards.

Selecting Input Parts for Mesh Operations

Selecting individual parts

The individual geometry parts that are directly selected, along with part surfaces and part curves that comprise the geometry part, belong to the mesh operation.

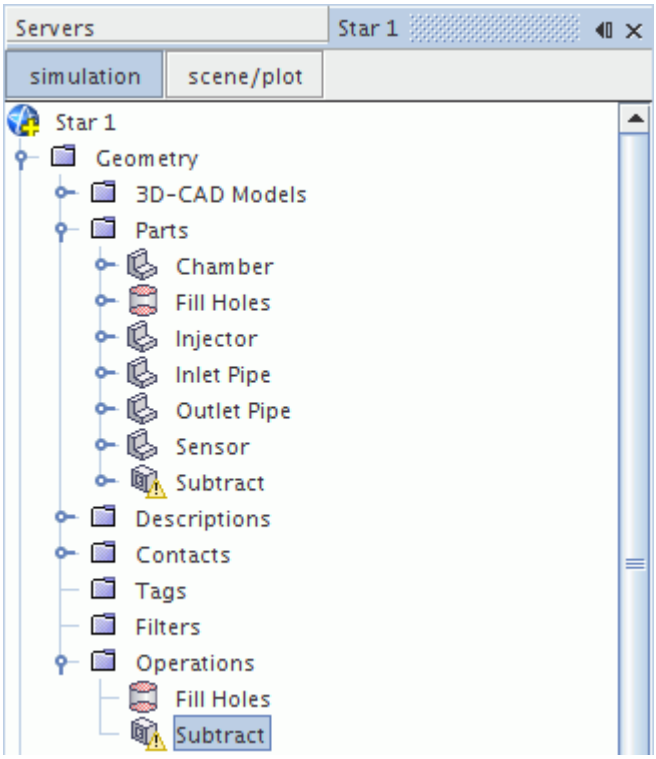
Selecting composite parts

All parts that belong to the composite part, along with part surfaces and part curves that comprise the geometry part, are included in the mesh operation, even if they don't appear selected individually.

Executing Operations

You execute operations to perform a specified action on a geometry part.

When you create an operation, change its properties, or modify the input parts, a warning triangle on the operation and part indicates that the operation needs executing. Mesh operations use the latest descriptions for input parts that are available before that operation. See Mesh Descriptions.



The following methods are available for executing operations:

Contents:

Executing Individual Operations
Executing All Operations
Executing Operations on Creation

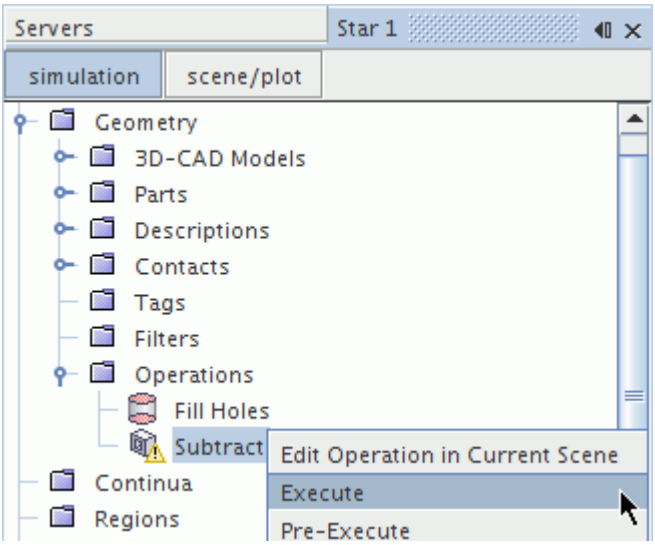
Executing Individual Operations

You can execute individual mesh operations to perform a single action on a geometry part.

Note: Executing an individual operation also executes any out-of-date operations that the selected operation requires as an input.

To execute individual operations:

1. Right-click a mesh operation and select **Execute**.



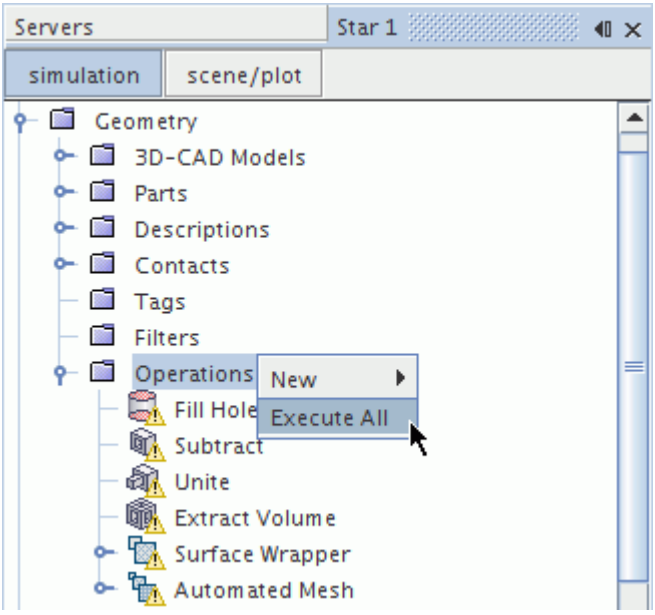
Executing All Operations

You can execute all operations to run the entire meshing pipeline on the geometry parts.

This action executes the operations sequentially in the order that they appear under the **Operations** manager.

To execute all operations:

1. Right the **Geometry** > **Operations** node and select **Execute All**.

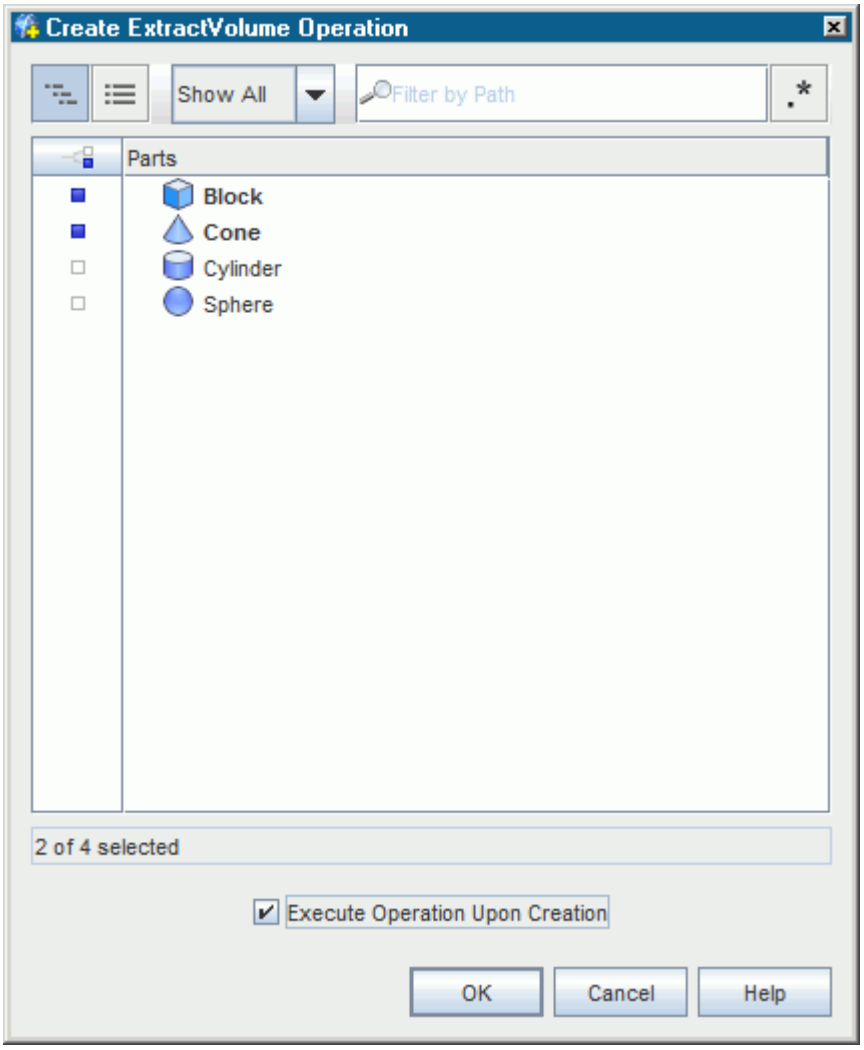


Executing Operations on Creation

When creating some mesh operations, you can activate the option to *Execute Operation on creation*. If this option is activated, the operation is executed when you click **OK**.

The option to activate *Execute Operation Upon Creation* is available for the following mesh operations:

- **Boolean** > **Unite Parts**
- **Boolean** > **Intersect Parts**
- **Boolean** > **Subtract Parts**
- **Fill Holes**
- **Extract Volume**
- **Badge for 2D Meshing**



Copying Controls Between Mesh Operations

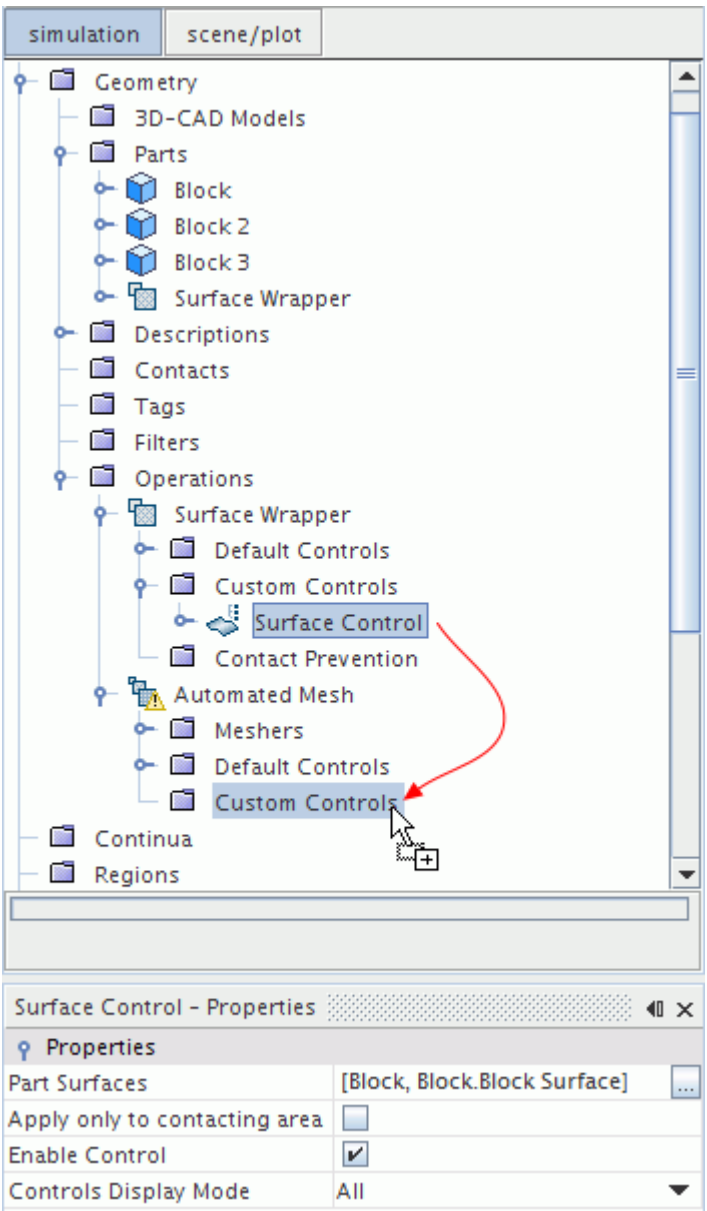
It is possible to copy and paste controls between mesh operations.

Input parts for the controls on the source mesh operation are only copied to the destination controls if the input parts exist in the destination mesh operation (except inherited parts).

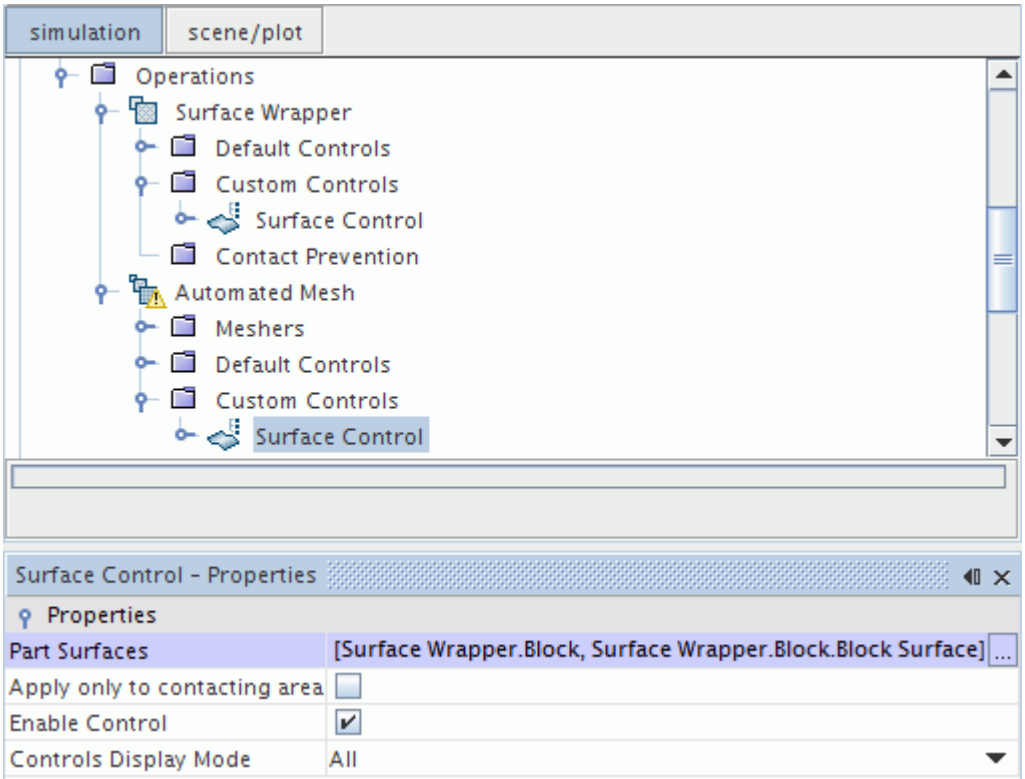
Parts from the source mesh operation are converted to the appropriate inherited parts in the destination mesh operation. It is also possible for inherited parts to be converted into their appropriate parent parts, depending on which operations the controls are copied to and from.

To Copy Controls Between Mesh Operations:

1. Click and hold the source mesh operation **Control** node that you want to copy.
2. Drag and drop the **Control** node onto the corresponding **Controls** manager node in the destination mesh operation.



The **Controls** manager node in the destination mesh operation contains a copy of the **Control**. The Part Surface names in the destination mesh operation **Control** reflect the appropriate inherited part name.



See Also:

Creating Mesh Operations

Mesh Operation Errors

Parts with Errors or Empty Parts

If any of the parts that are selected for an automated mesh operation or surface wrapper operation have errors or contain no surfaces (empty parts), an error message appears in the output window. The names of affected parts are included in the message.

- Multi-Part Meshing
  - If there are multiple parts in the mesh operation and all parts are empty, the following message is displayed:

    ```
    Parts do not contain any faces. Please check surface before meshing.
    ```

  - If there is only one part in the mesh operation, the following message is displayed:

    ```
    Part: Body 2 does not contain any faces. Please check surface before meshing.
    ```

- Per-Part Meshing

In Per-Part meshing where there are multiple parts and one or more parts are empty, the following message is displayed:

```
WARNING: Part: Body 2 is empty and therefore cannot be meshed
WARNING: Part: Body 3 is empty and therefore cannot be meshed
```

Any parts that are not empty are meshed.

Resolving the Empty Faces

After locating the part or parts that are specified in the error message, you can reimport or re-create the mesh, then re-execute the mesh operation.

Surface Extruder

The surface extruder extends the edges of the input surface to form the bounding surfaces of the extruded region. You specify the extrusion distance in the surface extruder operation.

For information on how to use the extruder mesher in parts, see Using the Extruder in Parts.

Surface Extruder Properties

Parameter	Description
Input Parts	Specifies the part that acts as the source for the extruder.
Input Part Surfaces	Specifies the surface to be extruded. The input surfaces are projected a specified distance away from the original surface. The edges of the input surfaces are extruded to form the sides of the extruded region.
Create New Part	Specifies what parts are created as a result of the surface extrude operation. **Single New Part** creates a single part for the surface extruder, even if multiple inputs are selected. **Per Part** creates separate parts for each individual input parts. **Per Part Surface** creates separate parts for each individual input part surfaces.
Output Part Surfaces	Specifies how to pre-execute part surfaces in the output part. **All Pertinent Sides** creates one side part surface in the extruded part for every side part surface of the input parts. **Single Side** creates a single side part surface for all input sides.

Surface Extruder Controls

Parameter	Description
Direction Mode	Specifies the method to use when defining the direction and distance of the extrusion. **Surface Normal** extrudes the region normal to the input part surface. If the input part surface is not completely planar, select **Average Normal**. **Specified** extrudes the region in a specified direction.
└ *Extrusion Distance*	Specifies the distance of the extrusion when you set *Direction Mode* to **Surface Normal** or **Average Normal**.
└ *Specified*	Specifies the coordinate system and direction when you set *Direction Mode* to **Specified**. *Coordinate System* specifies the coordinate system to use for the extrusion direction. You can use custom coordinate systems. *Distance* specifies the magnitude of the extrusion, where each component is the displacement along the corresponding axis of the chosen coordinate system.

Volume Extruder

The volume extruder fills in the extruded region with extruded volume cells from the input surface. You control the amount of divisions in the extruded region the volume extruder operation. You must have a volume mesh set up in parts to use the extruder in parts.

For information on how to use the extruder mesher in parts, see Using the Extruder in Parts.

Volume Extruder Properties

Parameter	Description
Input Parts	Specifies the part that acts as the source for the extruder.

Volume Extruder Controls

Parameter	Description
Number of Layers	Specifies the number of cells in the direction of extrusion. The default value is 1, which is the minimum. There is no maximum limit to the number of layers allowed.
Stretching Method	Specifies the spacing between the layers. **Constant** extrudes in a specified direction with constant thickness layers. **One Sided Geometric** uses

Parameter	Description
	geometric progression to compute cell thickness. **One Sided Hyperbolic** uses a hyperbolic stretching law to compute the cell distribution growth in the direction of the extrusion. This option is geared specifically for aerospace and other applications where the boundary layer resolution is considered critical. **Two Sided Hyperbolic** uses hyperbolic stretching from both ends to compute cell distribution.

Volume Extruder Stretching Options

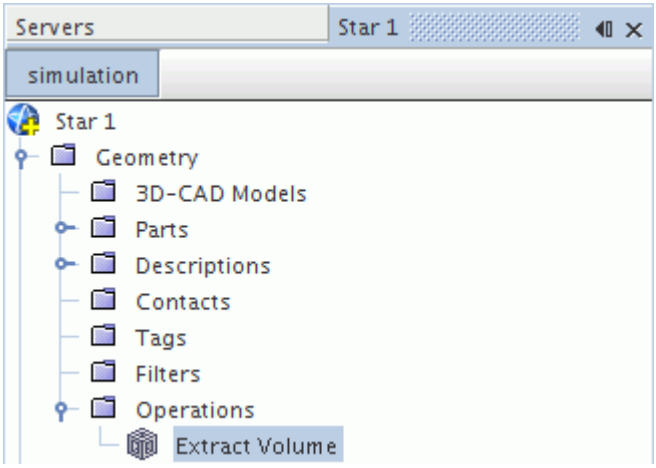
Parameter	Description
Stretching > **Stretch Value**	Specifies the ratio of spacing at layer n and layer n +1.
Initial Wall Thickness > **Initial Wall Thickness**	Specifies the thickness of the first layer of cells near the input part surface.
Thickness Ratio > **Thickness Ratio**	Specifies the layer spacing between the first layer and the last layer of cells.

Extract Volume

The extract volume mesh operation finds enclosed volumes within selected parts and generates a part to fill the volume.

If parts are imprinted before extracting the volume, there are no solid parts to merge, so their tessellated surfaces are merged instead. The tessellated surface allows the volume to be extracted, as long as the combination of merged internal tessellated surfaces are closed.

The **Extract Volume** node represents this operation in the simulation tree.



Extracting the volume in parts follows a similar process to extracting the volume in 3D-CAD. The main difference is that in parts, you must first manually cap all the openings using the fill holes operation before you can extract the volume. See Fill Holes. In 3D-CAD, the holes are automatically capped as part of the operation.

Extract Volume Properties

Use the **Extract Volume** properties to define the input parts, volume extraction mode, and output part display options. You can also identify one or more output parts from this operation.

Input Parts	Specifies the geometry parts from which the operation extracts an internal volume.	
Volume Mode	**Automatic**	Automatically determines the volume of interest for the volume extraction. In the **Volumes** sub-node, you can specify if a single volume or all volumes are automatically extracted. **Single** mode places all closed volumes that are identified into a single output part. Any volume that is already identified as an existing part is ignored. Any volume that spans multiple parts is included in the extracted result. **All** mode creates a new output part for every volume that is identified, even if they match existing parts. Any surfaces that are internal to the final extract volume parts are marked as baffles.
	Manual	Allows you to manually specify the volume of interest for the volume extraction. In the **Volumes** > **Extract Volume** node, set *Mode* to **Surface** or **Point** to define the volume of interest using surfaces or points. Define the surfaces and points (referred to as seed points) in the **Extract Volume** > **Surface** and **Extract Volume** > **Seed Points** nodes respectively. See the Surfaces Properties and Seed Points Properties in the sections below.
Combine By Part Surfaces	When activated, combines any volumes with shared part surfaces.	
Output Parts	Displays the name of the output part that the operation generates (read-only).	

Surfaces Properties

The **Volumes** > **Extract Volume** > **Surfaces** sub-node *Properties* allow you to specify the surfaces:

Surfaces	Specifies the part surface that the **Surface** mode uses to extract an internal volume. For example, you can use the part surfaces from hole filling, where the ends of an inner cylinder are capped off.

Seed Points Properties

The **Volumes** > **Extract Volume** > **Seed Points** node itself has no properties although right-clicking this node allows you to create seed points that are displayed as sub-nodes. The **Seed Points** sub-nodes *Properties* allow you to specify the seed point coordinates:

Coordinates	Specifies the Cartesian coordinates of a seed point that the **Point** mode uses to extract an internal volume. These seed point coordinates are relative to the Laboratory coordinate system.

See Also:

Extracting the Internal Volume

Zip Edges

Use the Zip Edges operation to fix minor issues on a surface—such as the closing of small gaps, or stitching multiple parts together.

The zip edges operation is similar to the **Sew** feature in 3D-CAD, and the **Zip Selected Edges** tool in the Surface Repair Tool. The **Sew** feature requires CAD data, is typically used to stich small gaps between multiple surfaces, and does not form part of the pipeline. The **Zip Selected Edges** tool only works with discrete surfaces (the CAD data is lost after entering Surface Repair), is typically used to repair minor gaps or tears in the surface, and does not form part of the pipeline. The **Zip Edges** operation currently works with discrete surfaces, can heal surfaces (such as tears and small gaps), and can zip edges together—all while being part of a repeatable pipeline.

The zip edges operation works well together with the replace part operation when you have multiple designs that you want to switch between in the pipeline. Using these two operations together, you can, for example, switch between different designs of a car body, stitch them together, and mesh the updated body in a repeatable pipeline. See Replacing a Part in a Parts-Based Mesh.

Zip Edges Options

Set the options to control the outcome of the zip edges operation.

Input Parts	Specifies the parts on which the operation zips the edges. To heal a surface defect, such as a tear, only include the relevant part in the operation. To stitch multiple part surfaces together, include all the parts in the operation.
Zip Mode	Specifies the type of zip operation to execute. **Discrete Part Generating** zips the edges of the selected input parts to produce a single output part. **Discrete Description Generating** zips the edges of the selected input parts to produce a description with new contacts (where appropriate).
Zip Free Edges	When activated, all free edges are added to the collection of edges that are zipped as part of the operation.
Input Curves	Specifies curves that are added to the collection of edges that are zipped as part of the operation.
Auto Tolerance	Activate this option to let the operation choose an appropriate tolerance. The tolerance is calculated based on the size of the input parts. Deactivate this option to manually specify a tolerance for this operation. In some instances, the auto tolerance can include edges that you did not intend to zip. In this situation, deactivate the auto tolerance option and manually specify a suitable tolerance.

Tolerance	Specifies the tolerance for the operation. This field is only relevant when you deactivate the **Auto Tolerance** option.
Remesh After Zipping	When activated, remeshes the portion of the surface mesh around the zipped edges. Activate this option if the input part surfaces have been remeshed and you want to preserve the high-quality surface mesh. With this option activated, you can save time by only remeshing where needed instead of having to remesh the entire surface mesh further down the pipeline.

See Also:

Sewing Sheet Bodies Together

Repair Tools Reference

Weak Contact Creator

You can create weak contacts as part of the pipeline using the weak contact creator operation.

In some situations, where you have many bodies and don't want to imprint them all together, you can choose to generate a mesh per-part and let the solver handle the weak in-place contacts. This method is popular for CHT analyses. You can only generate non-conformal meshes when meshing per-part and using weak in-place contacts.

Without the weak contact creator operation, creating weak in-pace contacts would be a labor-intensive, manual process, or would require the use of Java macros. The weak contact creator automatically creates weak in-place contacts between adjacent part surfaces and maintains this step as part of the pipeline.

Weak Contact Creator Options

Set the options to control the outcome of the weak contact creator operation.

Input Parts	Specifies the pars to include in the operation. Weak in-place contacts are created between these parts.
Tolerance	Specifies the tolerance for the operation. Weak in-place contacts will be created for parts that fall within this tolerance.

Fill Holes

You can use the Fill Holes operation to fill gaps in the geometry with a closed surface.

Fill Holes Properties

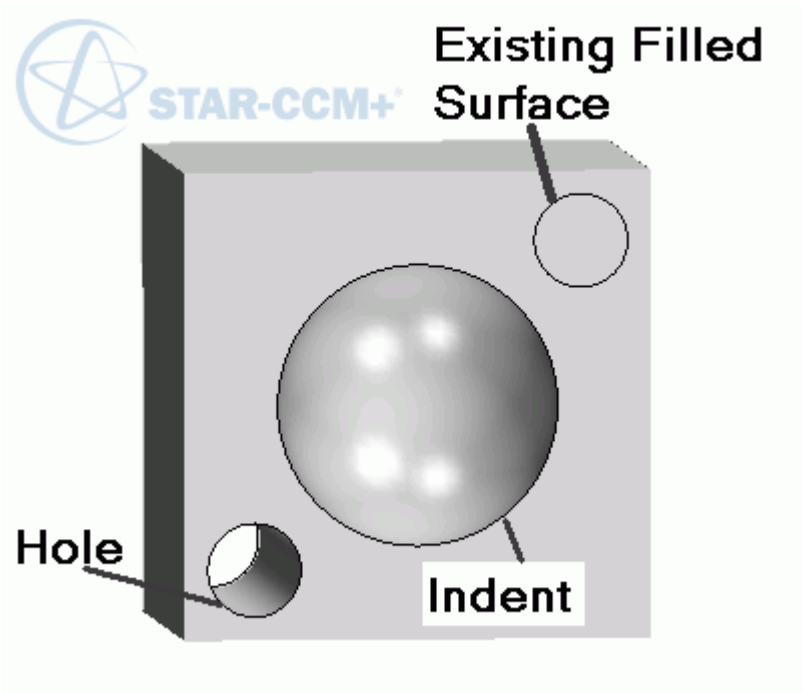
Use the **Fill Holes** properties to define the areas which you want to fill.

Input Parts	Specifies the geometry parts on which the operation fills holes. If you use a derived part that is the result of a mesh operation, also select the inherited parts which contain the inherited part surfaces and inherited part curves.

End Surfaces	Available to specify when **Input Parts** are specified. Specifies the location of the part surfaces that contain gaps that you want to fill.
End Curves	Available to specify when **Input Parts** are specified. Specifies the location of the part curves that you want to fill.
Output Parts	The name of the output part that the operation generates (read-only).

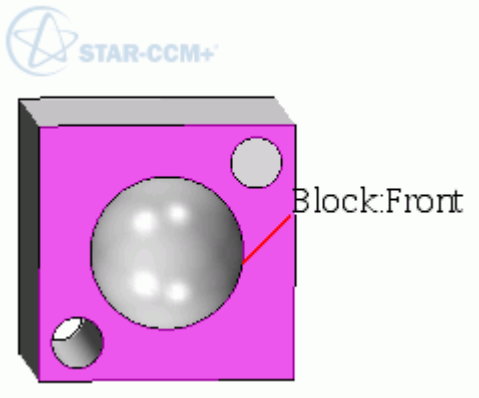
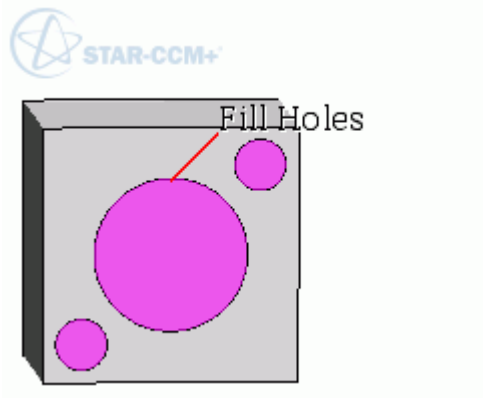
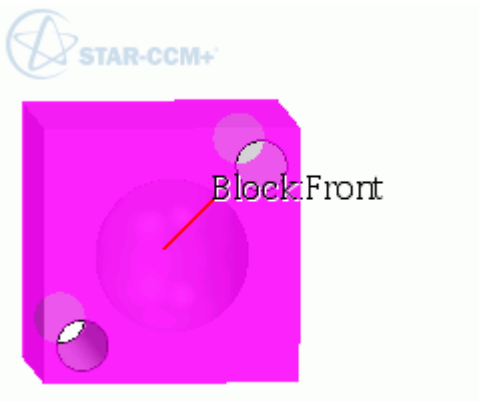
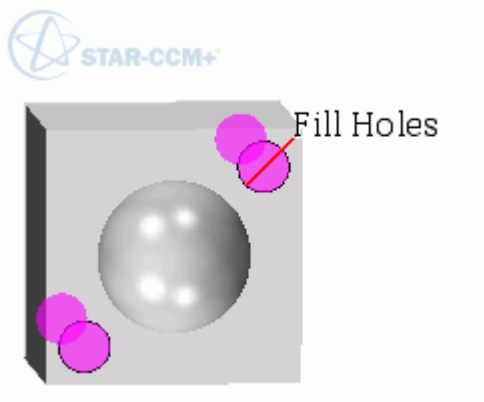
Fill Holes Example

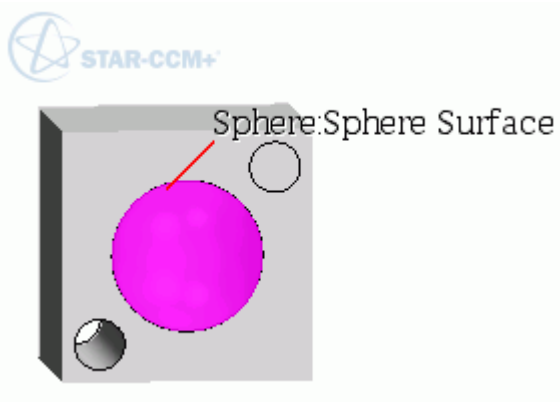
The following geometry contains a hole, an indent, and an existing filled surface. You can fill any or all of these areas using the fill holes operation.



The following table shows examples of using the fill holes operation on the above geometry:

Example Action	Example Result
Set **End Surfaces** as the part surface on which there are gaps that define the hole, the indent, and the existing filled surface.	A new surface is created to fill all of the gaps within the selected part surface.

Example Action	Example Result
	 The surface that is created can be split into separate part surfaces. See Splitting Non Contiguous Part Surfaces.
Set **End Surfaces** as the part surface for the entire geometry part on which there are gaps that define both sides of the hole and both the front and back existing filled surfaces. In this example, the indent is part of the selected surface. 	A new surface is created to fill all of the gaps that define the front and back of the hole and the front and back existing filled surfaces. 
Set **End Surfaces** as the part surface which contains gaps that you want to fill. In this example, the gap is the opening to the indent. If the input part is a derived part that is the result of a mesh operation, you can set **End Surfaces** as the part surface of the input part that created the derived part. In this geometry, the indent was created by subtracting a sphere from a block. In this example, you can set the sphere surface as the **End Surfaces**.	In both of these examples, a new surface is created to fill the gap that defines the opening to the indent.

Example Action	Example Result
	
Set **End Curves** as all of the part curves, including the part curves that define the gaps of the hole, the indent, and the existing filled surface.	A new surface is created to fill all of the gaps within the selected part curves.
Set **End Curves** as the specific part curves that enclose the front and back of the hole.	A new surface is created to fill all of the gaps that define the front and back of the hole.

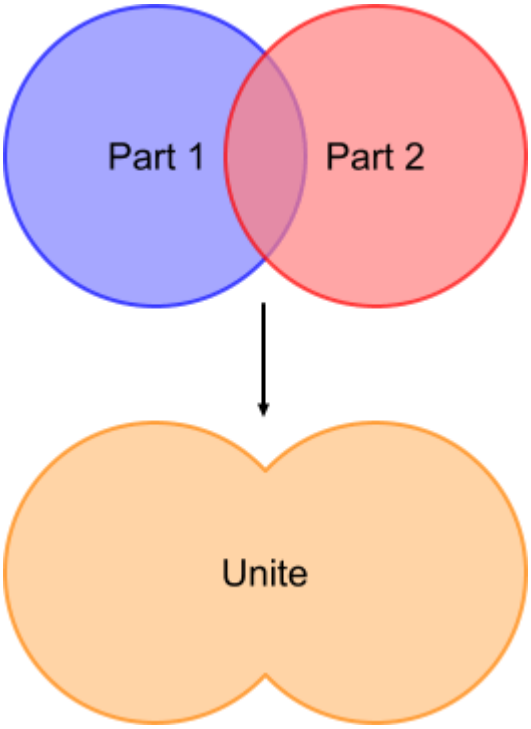
Unite Parts

The unite mesh operation performs a boolean unite on all input parts that are in contact with or overlap each other.

Selecting the **Operations** > **Unite** node allows you to set properties for the unite mesh operation in the properties window.

Example of the Unite Parts Operation

The following diagram illustrates the behavior of the unite parts operation.



See Also:

Boolean Mesh Operation Properties

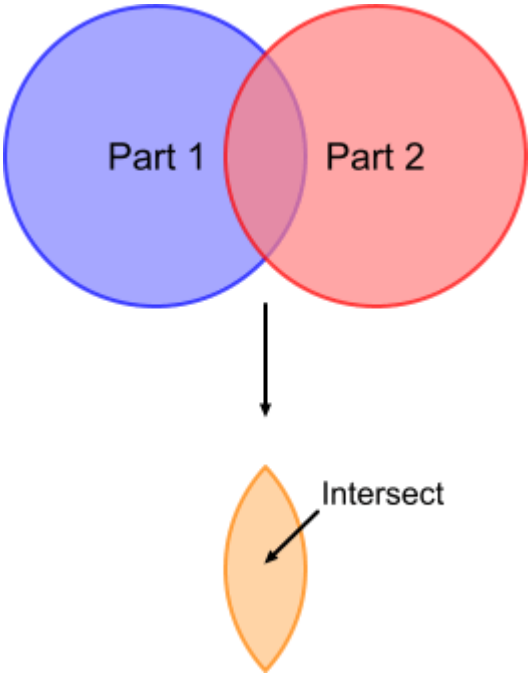
Intersect Parts

The intersect mesh operation performs a boolean intersect on all input parts that overlap each other.

Selecting the **Operations** > **Intersect** node allows you to set properties for the intersect mesh operation in the properties window.

Example of the Intersect Parts Operation

The following diagram illustrates the behavior of the intersect parts operation.



See Also:

Boolean Mesh Operation Properties

Subtract Parts

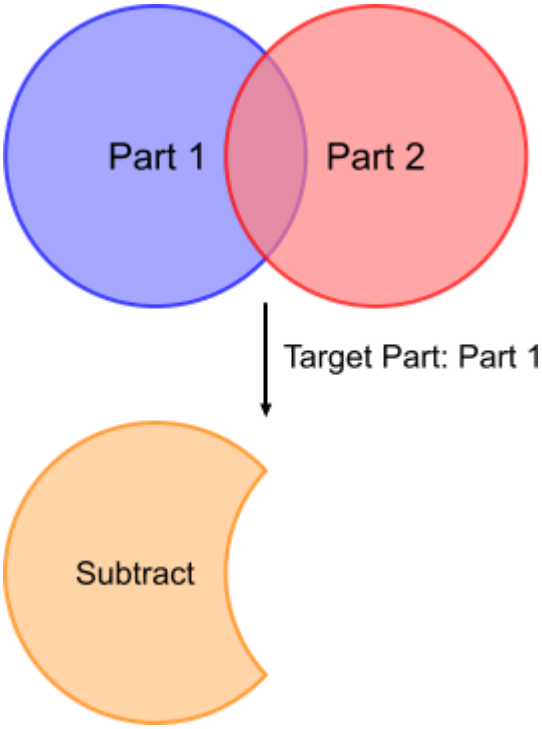
The subtract mesh operation removes geometry parts from the target geometry part.

This operation performs geometrical changes (as opposed to topological) to the target part.

Selecting the **Operations** > **Subtract** node allows you to set properties for the subtract mesh operation in the properties window.

Example of the Subtract Parts Operation

The following diagram illustrates the behavior of the subtract parts operation.



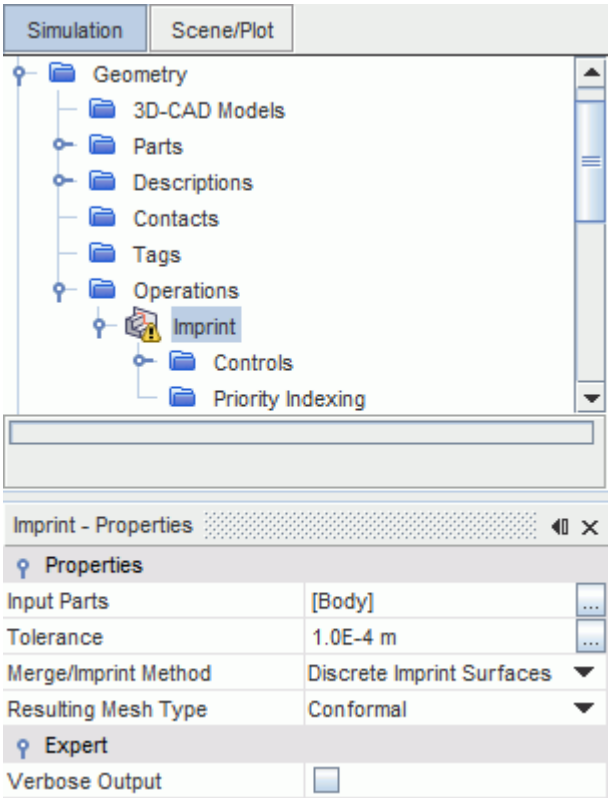
See Also:

Boolean Mesh Operation Properties

Imprint Parts

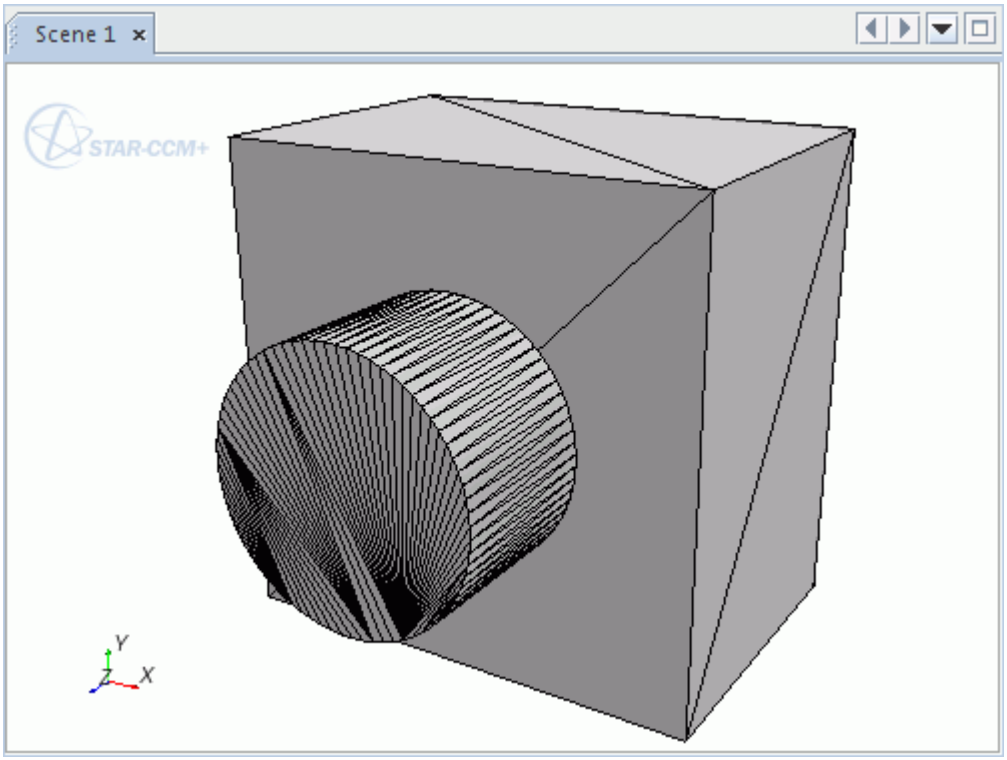
The Imprint operation imprints geometrically close surfaces, or geometrically close surfaces and edges, within a defined tolerance. Part contacts are created between the surfaces that are coincident and you can choose to create a conformal or non-conformal surface mesh between parts.

Selecting the **Operations** > **Imprint** node allows you to set properties for the imprint operation in the properties window.

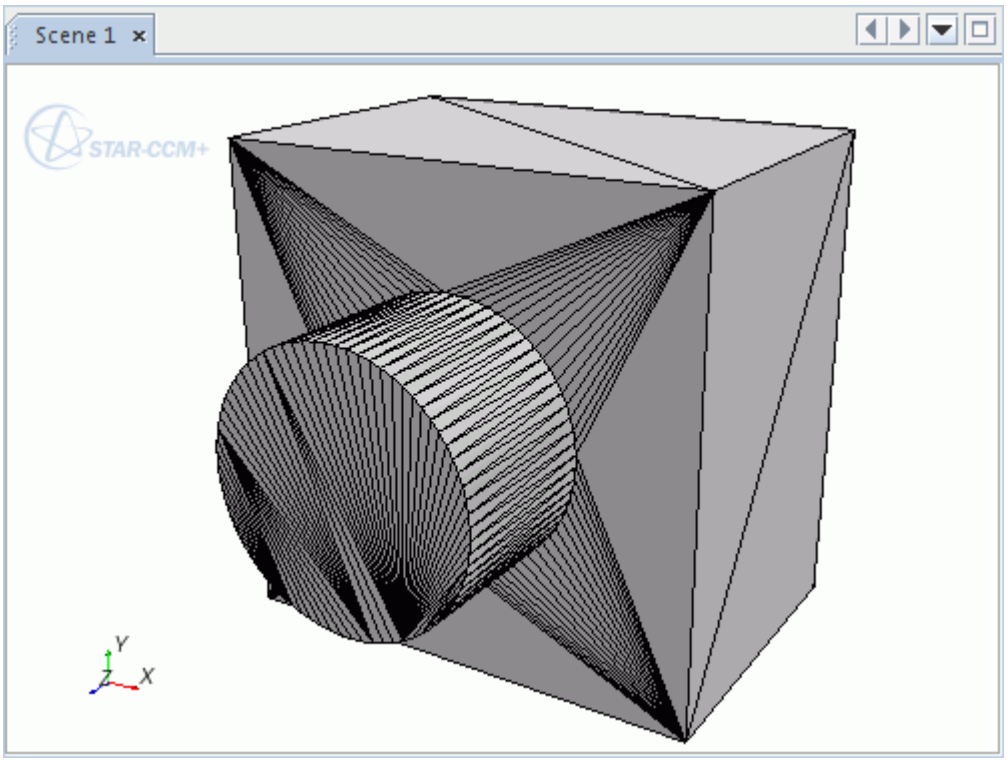


Example of the Imprint Parts Operation for a Conformal Mesh

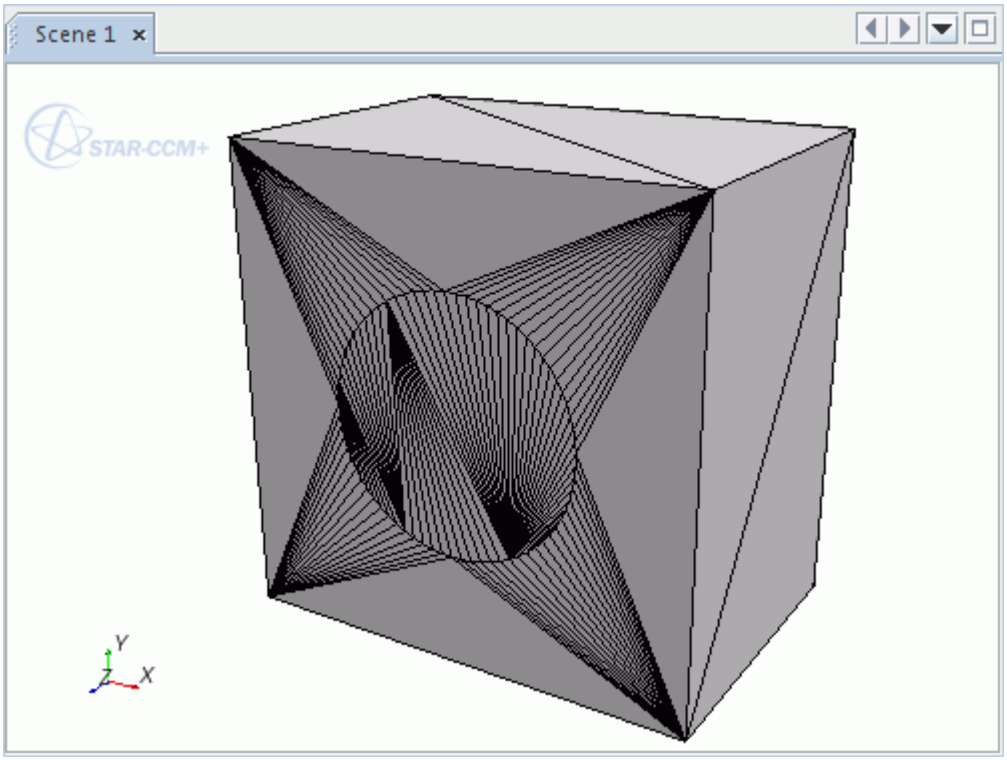
The following geometry shows two parts that have contacting surfaces between each other. To achieve a conformal mesh between the contacting surfaces, before you imprint the parts, set *Resulting Mesh Type* to **Conformal**.



After imprinting, the resultant surfaces are conformal, which means they have the same surface discretization. By default, in-place part contacts are created between the contacting surfaces.



By hiding the cylindrical surface, you can see that the surface discretization on the contacting surfaces is identical.



Contents:

Non-Conformal Imprinting
Imprint Parts Resulting Part Surfaces
Imprint Properties
Imprint Controls

See Also:

Priority Indexing

Non-Conformal Imprinting

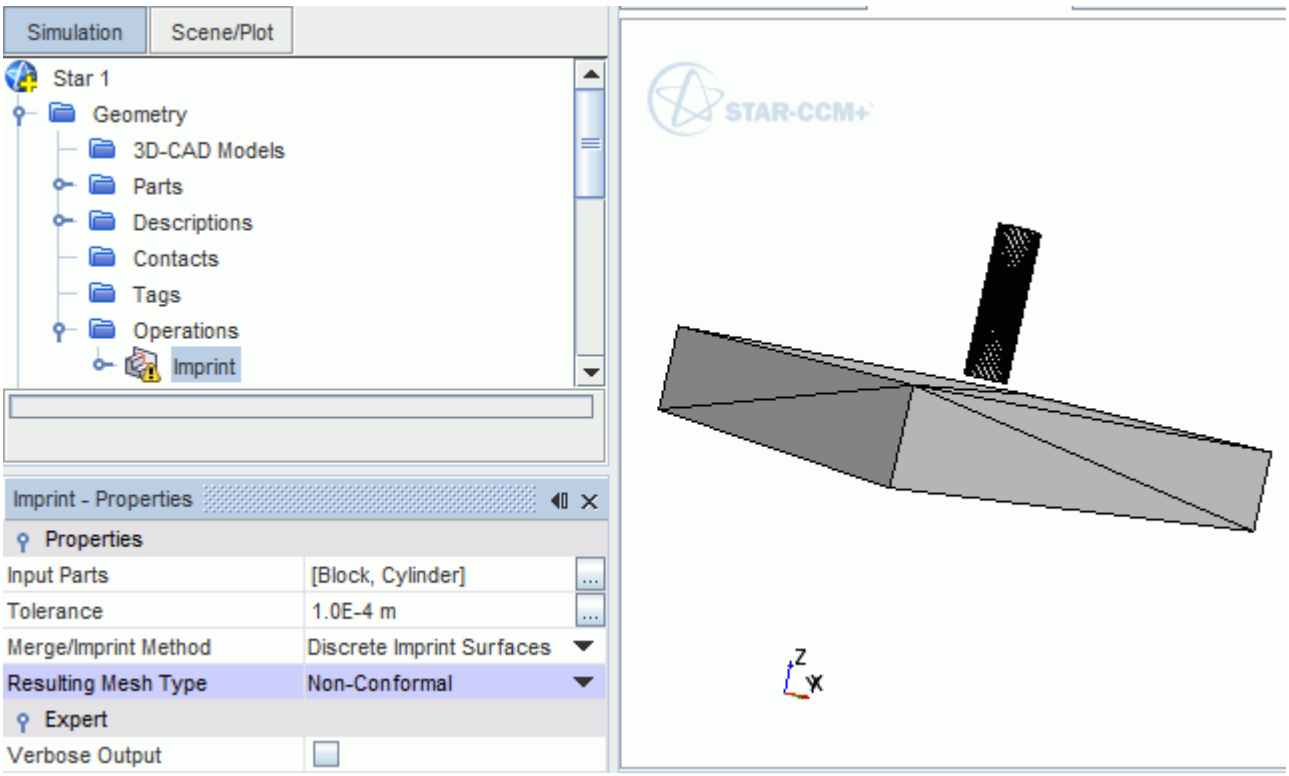
STAR-CCM+ allows you to imprint parts non-conformally. Non-conformal imprinting provides a way to imprint parts efficiently without taking time to imprint CAD parts conformally when it is unnecessary.

This process is useful when:

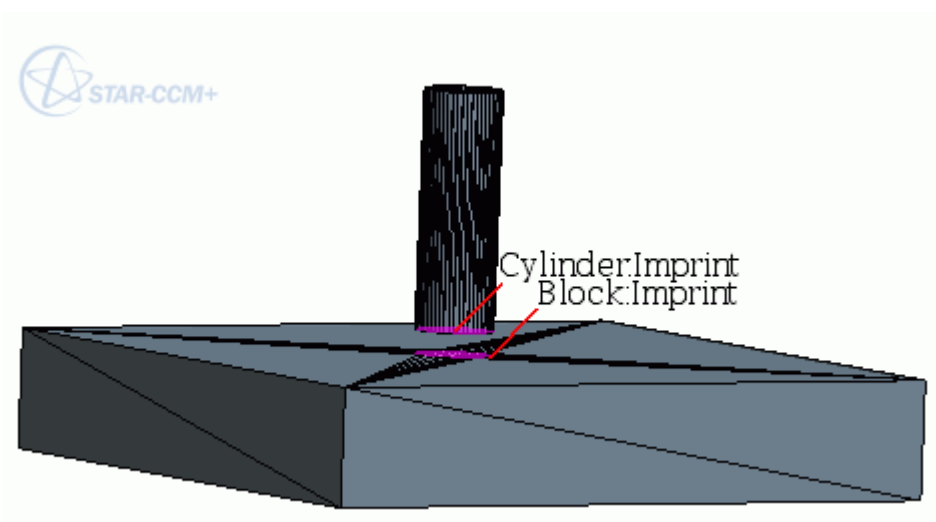
- a conformal interface is not necessary to achieve solution accuracy
- there are many contacting parts
- each part is within many pairs
- there are gaps between the CAD parts

During the process of non-conformal imprinting, pairs of split surfaces are created on contacting parts without altering each individual part.

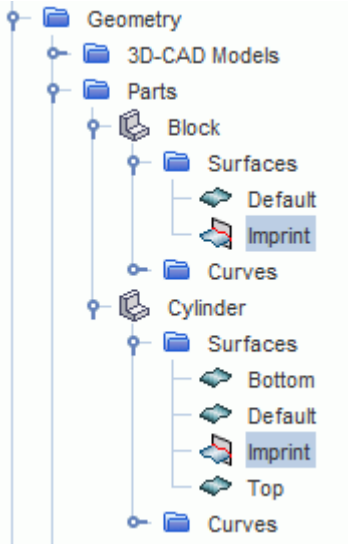
The following geometry shows two parts that are in close proximity with a small gap between the surfaces. To achieve a non-conformal mesh when imprinting these surfaces, before you imprint the parts, set *Resulting Mesh Type* to **Non-Conformal.**



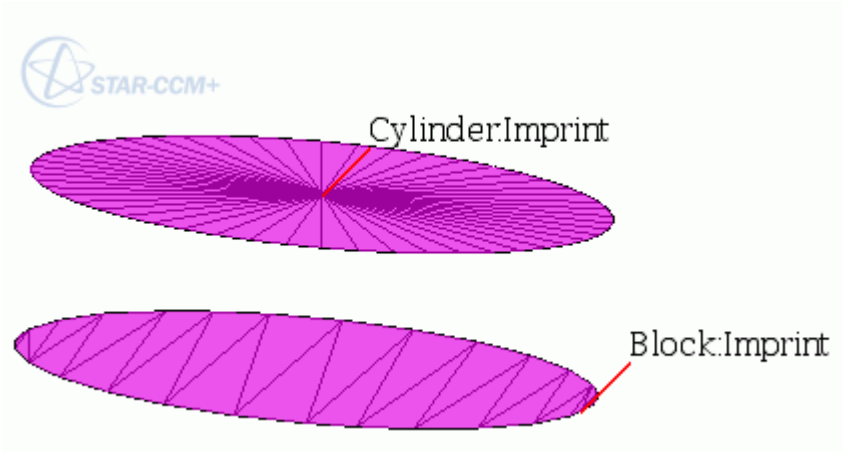
After imprinting the parts, the resultant surfaces are non-conformal, which means that the surface discretization on the imprinted surfaces does not match. Each part remains separate and there is no shared surface between each part. However, the original discretized surfaces are cut and a pair of split part surfaces are created which represent an in-place part contact between the surfaces.



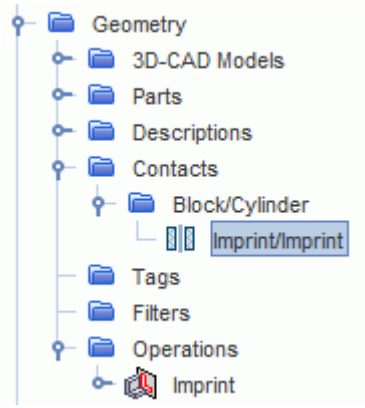
The pair of split part surfaces are displayed in the **[Part]** > **Surfaces** folder:



Taking a closer look at the pair of split surfaces, you can see that the surface discretization on each split part surface is non-conformal:



The in-place part contact which represents the pair of split part surfaces appears in the simulation tree:



Imprint Parts Resulting Part Surfaces

The assignment of imprinted faces to a part surface depends on the **Imprint** > **Controls** > **Resulting Part Surfaces** option that you select.

This option is only available for conformal imprinting as non-conformal imprinting creates new part surfaces by default.

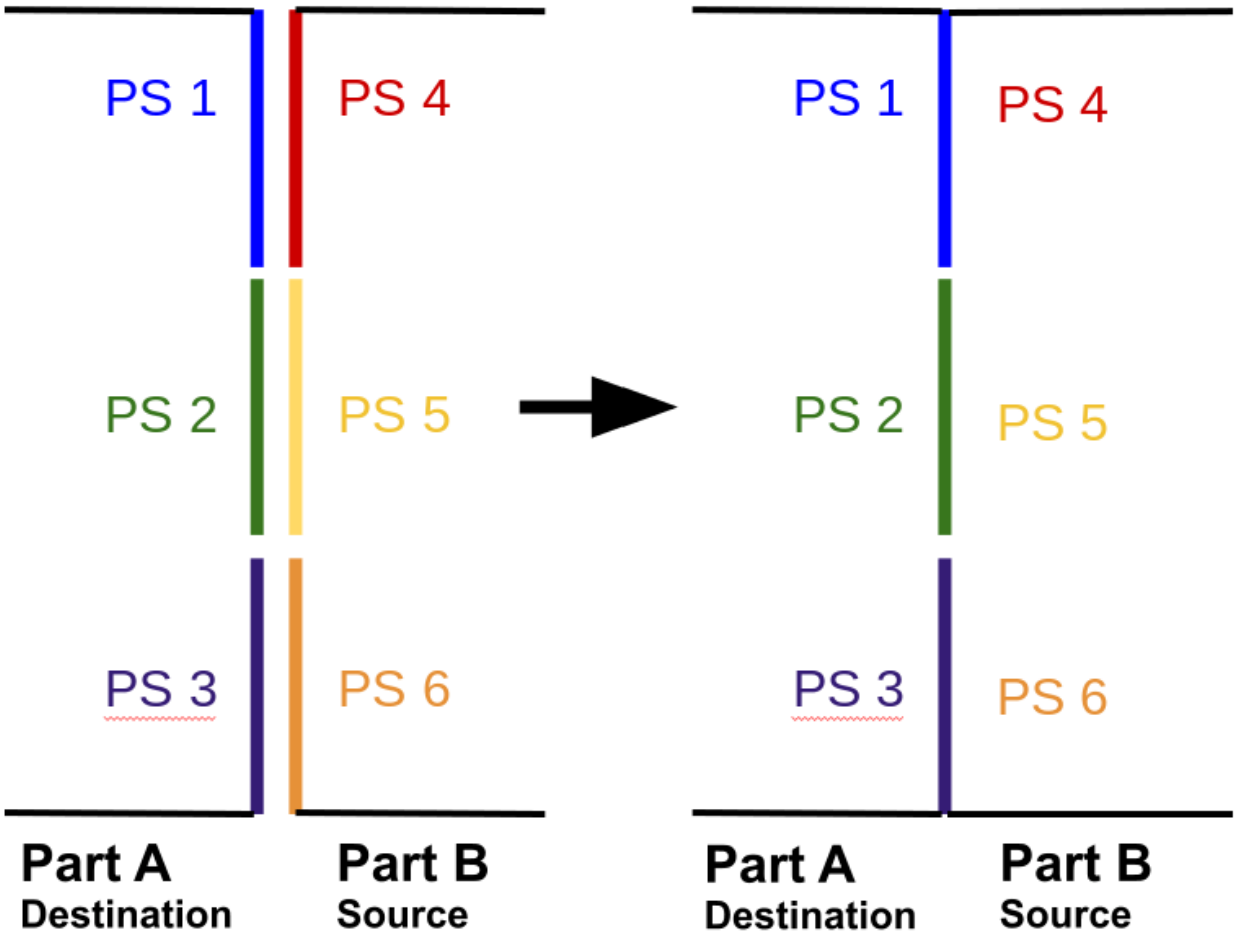
There are two options available:

- Input Part Surfaces
- New Part Surfaces

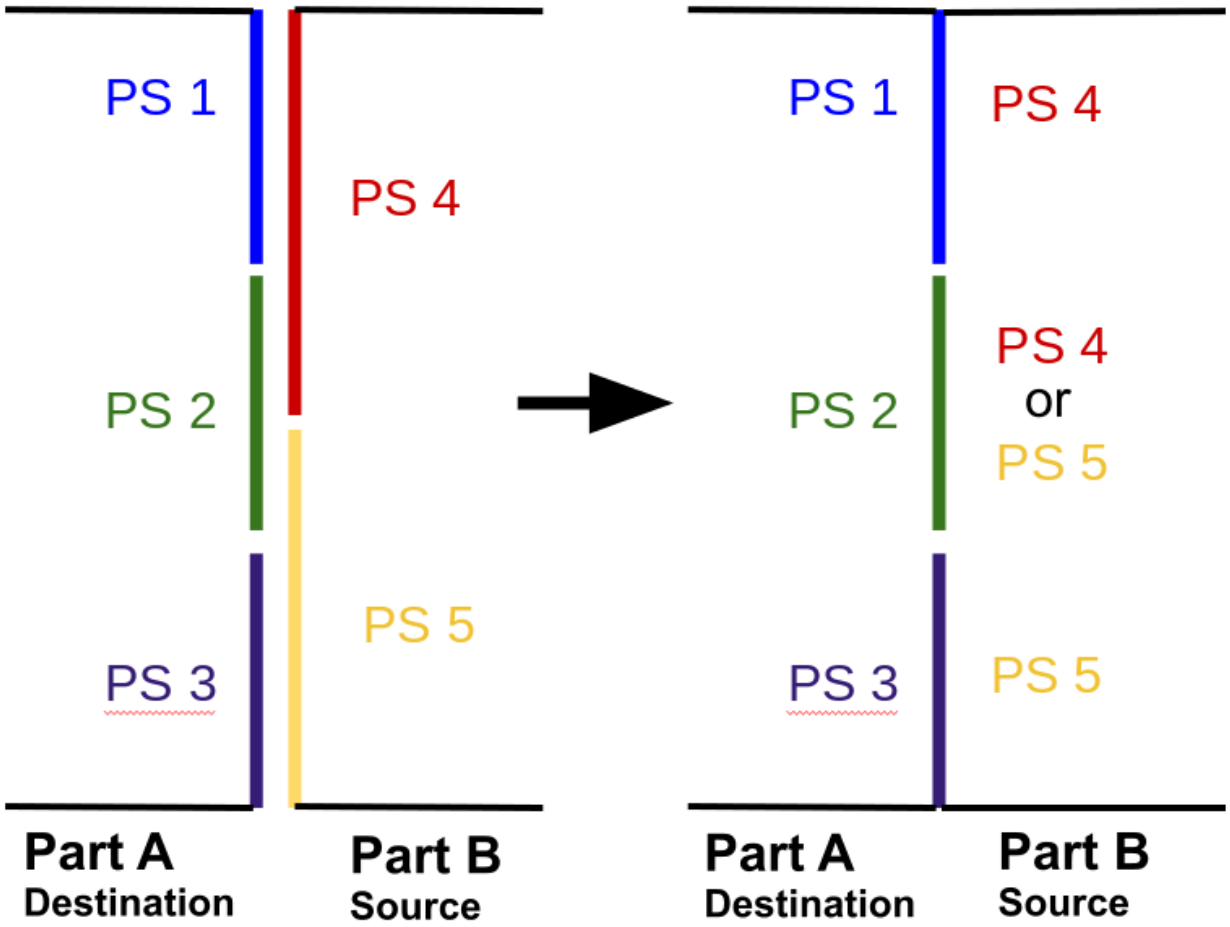
Input Part Surfaces

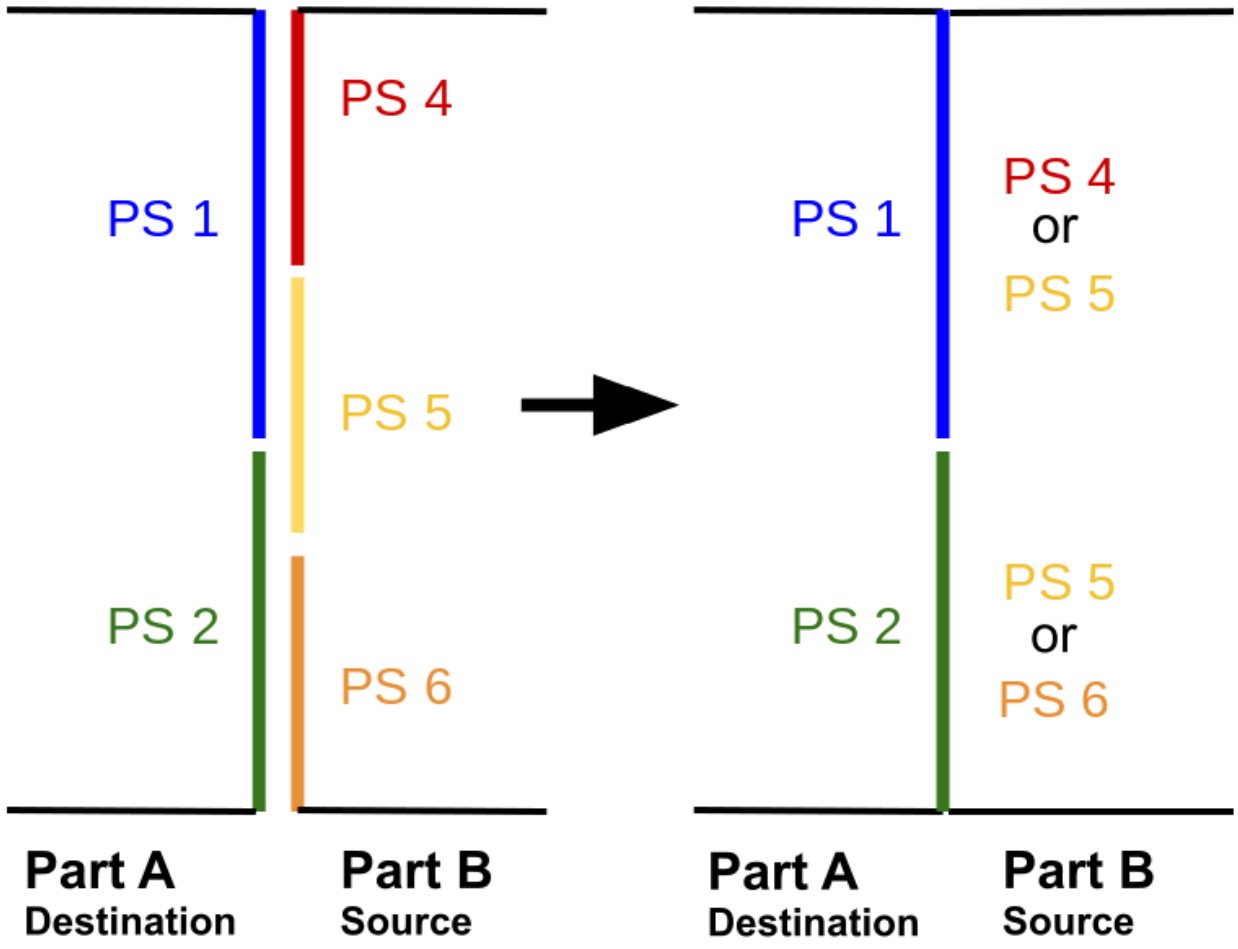
When *Resulting Part Surfaces* is set to **Input Part Surfaces**, the imprinted faces that are created are added to the part surface of the input part.

During the imprint operation, part surfaces from source input parts are assigned to the coincident part surfaces on destination input parts. If the part surfaces on the source part line up well with the destination part surfaces, the imprinted faces are assigned to the correct part surface.



If the part surfaces on the source part do not line up well with the destination part surfaces, the imprint operation only preserves part surface distinctions that are on the destination side of the interface.


PS 1
PS 4
PS 2
PS 5
PS 3
Part A
Destination
Part B
Source
PS 1
PS 4
PS 4
or
PS 5
PS 2
PS 3
PS 5
Part A
Destination
Part B
Source

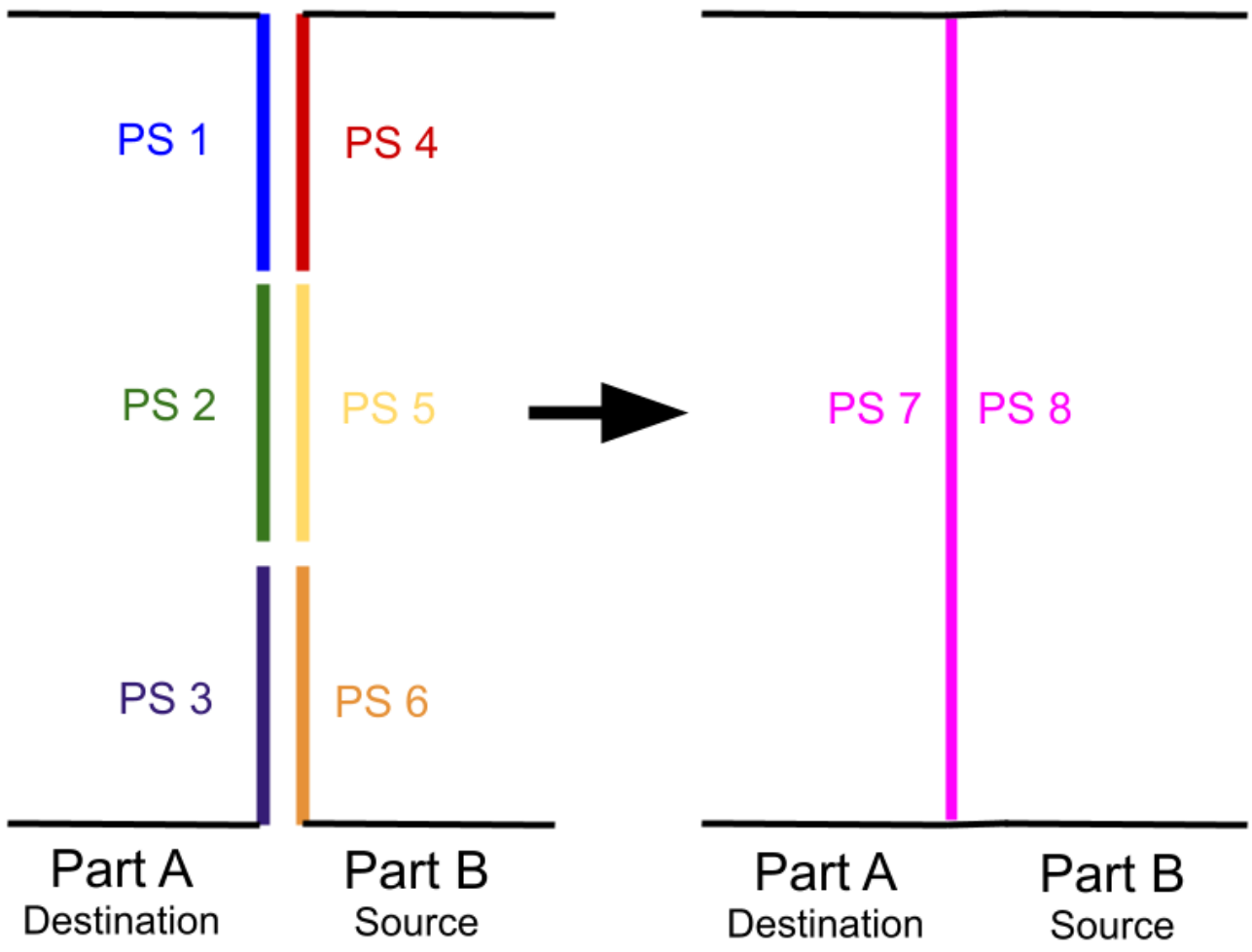


To specify which side of the interface is the destination, use priority indexing. See Priority Indexing.

New Part Surfaces

When *Resulting Part Surfaces* is set to **New Part Surfaces**, the imprinted faces that are created are added to new part surfaces. This behavior also reflects the part surface assignment that occurs by default during non-conformal imprinting.

One new part surface is created for each contiguous component of an imprint in a particular operation. Re-executing the operation reuses the part surfaces that are created from a previous execution whenever possible.



Imprint Properties

Use the **Imprint** properties to define the input parts, imprint tolerance, and whether to create mesh contacts or not.

Input Parts	Specifies the geometry parts that the operation imprints. The operation will not imprint a part on itself.	
Tolerance	Specifies the upper limit on the distance between part surfaces that the operation imprints. When imprinting edges, the upper limit is between the edges and faces. Any part surfaces or edges that are not within this tolerance are not imprinted.	
Merge/Imprint Method	Specifies how faces and edges are chosen and matched for imprinting.	
	Discrete Imprint Surfaces	The imprint is carried out using the tessellated surface mesh of the parts. The original tessellated

		faces are cut to ensure that the imprinted faces match.
	Exact Face Match	The imprint is carried out using the tessellated surface mesh of the parts. Coincident vertices and faces are merged and a part contact is created between the two parts. Specify this option for part pairs that already have exact one-to-one correspondence between their faces and vertices. See Using the Exact Face Match Method.
	CAD Imprint	The imprint is carried out using the CAD description of the parts. If the CAD description is not available on all parts, an error message appears informing you of this.
	Discrete Imprint Curves	The imprint is carried out using the tessellated surface mesh and edges or curves.
Resulting Mesh Type	Determines if the imprinter merges the faces or edges of the input parts.	
	Conformal	The imprinter merges faces of the imprinted parts to produce a conformal mesh on the output surface. For edge imprinting, the imprinter merges edges of imprinted parts to produce a conformal mesh.
	Non-Conformal	The imprinter does not merge faces of the imprinted parts so a non-conformal mesh is produced on the output surface. For edge imprinting, the imprinter does not merge edges of the imprinted parts producing a non-conformal mesh.

Imprint Expert Properties

Verbose Output	Specifies how much information is provided in the *Output* window while executing the Imprint operation.	
	Activated	The report in the *Output* window includes the number of matching source and target surfaces, the

		number of faces that were merged during the Imprint process, and the number of created contacts.
	Deactivated	The report in the *Output* window only displays the number of contacts created during the Imprint operation execution.

See Also:

Mesh Operations

Imprint Controls

Reimprint Existing Contacts (available for all imprinting methods except CAD imprinting)	Reimprints existing contacts. By default, this option is deactivated and only new contacts are imprinted. When activated, all input parts are imprinted without any consideration of existing contacts. This includes contacts between the parts in the current operation, as well as "collateral" contacts between one part in the operation and one part that is not in the operation. This option is not available for CAD imprinting because during CAD imprinting, existing contacts are already reimprinted. This option is useful in cases where you have partial contacts (where there are contacting faces between two parts but other faces still need imprinting and were missed). This option is also useful when imprinting an extract volume part to another part. Since the extract volume operation creates contacts with its input, all faces are in contact. Previously, the imprint operation was unable to imprint these parts.	
Candidate Pairs (available for conformal imprinting only)	Allows you to specify pairs of parts to imprint.	
	All Close Pairs	All close pairs of parts/part surfaces are imprinted.
	Use Existing Contacts	Only pairs of parts/part contacts with existing contacts are imprinted.
Source Curves (available for edge imprinting only)	Controls which edges the imprinter attempts to imprint. The operation will use the union of the following two choices:	
	Imprint Part Curves	Specifies the curves to imprint.
	Imprint Free Edges	Attempts to imprint any free edges on the body.
Preprocess	Allows you to specify if any boolean operations are performed before parts are imprinted.	
	No Preprocess	No boolean operations are performed before parts are imprinted.

	Boolean Subtract	A boolean subtract operation is performed before parts are imprinted.
	Intersect Surfaces (available for non-conformal imprinting only)	An intersect operation is performed to cut the existing surfaces before parts are imprinted. Coincident surfaces that lie inside other surfaces are also cut and automatically included in the resulting part surfaces.
Resulting Part Surfaces (available for conformal imprinting only)	Allows you to specify the surface that the resulting imprinted faces are added to. This option is only available for conformal imprinting as non-conformal imprinting creates new part surfaces by default. See Imprint Parts Resulting Part Surfaces.	
	Input Part Surfaces	The imprinted faces that are created are added to the part surface of the input part.
	New Part Surfaces	The imprinted faces that are created are added to a new part surface.
Resulting Part Curves (available for edge imprinting only)	Controls the part curves created.	
	Input Part Curves	Assigns the imprinted edges to existing part curves.
	New Part Curves	Assigns the imprinted edges to a new set of curves.
Face Orientation	Imprint based on the relative orientation of the outward normals of faces. For edge imprinting, this corresponds to the outward tangent at the edge.	
	Opposing and Aligned	Performs the imprint regardless of normals being opposing or aligned.
	Opposing Outward	Performs the imprint when outward normals are opposing. See Using the Opposing Outward Face Orientation Option. This is the default option.
	Aligned Outward	Perform imprint when outward normals are aligned.
Merge Angle	Specifies the maximum angle between the normals of the faces at which parts merge. Only pairs of faces which lie within the specified *Value* and have normal directions which differ by less than this *Merge Angle* are merged. For edge imprinting, this option specifies the tolerance angle between the edge to be imprinted and the faces.	
Tessellation Options (available for CAD imprinting only)	Controls the tessellation parameters. Click to the right of this property to display the *Tessellation Options* dialog. The options presented here are:	

	• **Mark Feature Edges** — Allows you to specify which of the mesh edges that lie along CAD edges are marked as feature edges. See Mark Feature Edges. • **Sharp Edge Angle (deg)** — Creates feature edges from edges of attached faces that make an angle sharper than this value. The default value is `30 degrees`. • **Tessellation Density** — determines the level of surface refinement on the geometry after imprinting. Activate the **Show Detailed Tessellation Parameters** option to reveal more parameters. See Show Detailed Tessellation Parameters.	
Resulting Part Surfaces (available for CAD imprinting only)	Specifies the assignment of imprinted faces to a part surface. See Imprint Parts Resulting Part Surfaces.	
Tolerance Type (available for CAD imprinting only)	Defines the tolerance method to use in the operation. The tolerance methods used in the imprint operation are the same as that used in 3D-CAD. See Imprinting Bodies: Imprint Bodies Panel.	
	Precise	Uses the default internal model precision of `1.0E-8 m` as the tolerance. This tolerance setting requires that faces of separate bodies be within `1.0E-8 m` of each other. This option takes a minimum amount of computational time and works well for all coincident faces and surfaces.
	Tolerant Faces and Edges	Reduces the model precision by using a higher tolerance value that you provide in the **Tolerance** property on all the edges and faces in the model. This option causes any subsequent operations to use the lower precision model.
Remesh After Imprint (available for discrete imprinting only)	Remeshes the geometry parts after the imprint operation. By default, this option is deactivated. The imprinter can introduce poor-quality sliver triangles (high aspect-ratio cells) to the mesh. In most cases, this is not a problem since you remesh the geometry parts downstream. However, running the imprinter could introduce poor-quality sliver triangles to an existing remeshed surface. Activate the **Remesh After Imprint** option to remesh the geometry parts after the imprinter runs to avoid any sliver cells being introduced to the mesh. This option is not provided for CAD imprinting because that method of imprinting always produces a tessellated surface. This option is not provided for exact face imprinting because that method of imprinting does not create any sliver triangles. Do not activate this option with surfaces that have not been remeshed (have the initial tessellation). Doing so typically produces poor results where the remesher will use very large sizes for the mesh, resulting in an excessively coarsened mesh.	

Bounded Shapes

Use bounded shapes to define your simulation domain or to refine the mesh in certain locations. Bounded shapes are shape parts that are bounded to existing geometry, and resize to maintain their relative shape and size according to the reference geometry.

Shape parts are useful for defining the simulation domain in external flows, or for defining global or local refinement regions; however, they are static shapes that remain in position, and the same size all the time. If you use a shape part to define the local extent for local meshing, and you change the design of your geometry, you have to manually update the size and postion of the shape part to suit the new geometry design. In contrast, a bounded shape automatically updates when you change the geometry.

Currently, you can create bounded shapes using blocks or spheres. Each face on the bounded shape is a separate surface in the output part, making it easier to manage the boundaries in your simulation.

Bounded Shape Operation Properties

Input Parts	Specifies the parts to use as a base for the inflation—that is, the growth, of the bounding shape.
Output Part Types	Specifies the shape of the bounded part. Currently, you can create a **Block** or a **Sphere**.
Coordinate System	Specifies the coordinate system to use in the operation. Custom coordinate systems that you create are available for selection.
Output Parts	Displays the name of the output part that the operation generates (read only).

Bounded Shape Inflation Options

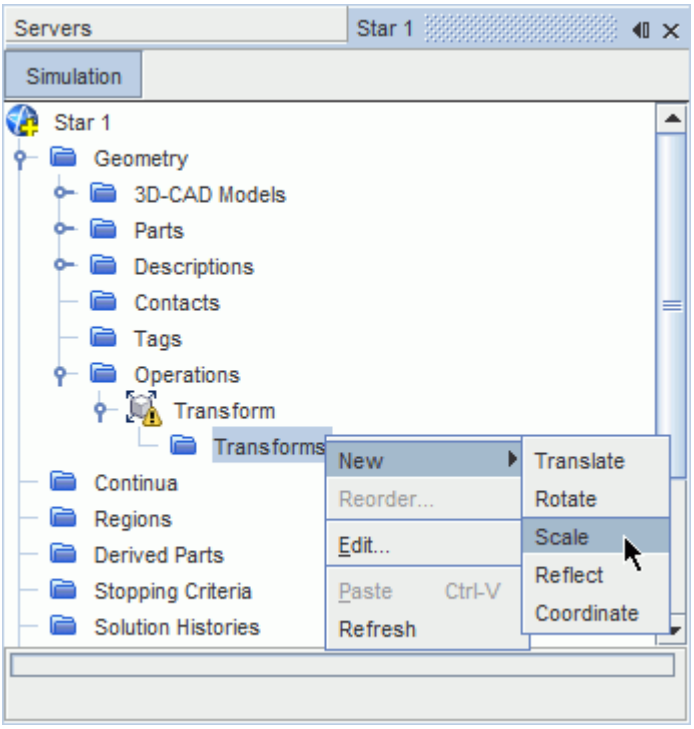
Inflation Control	Specifies the method to inflate—that is, to grow, the bounded shape. **Constant Factor** controls the inflation of the bounded shape using a single factor that inflates the bounded shape in all directions, relative to the input part. **Individual Factors** controls the inflation using three factors, *X Factor*, *Y Factor*, and *Z Factor* and inflates the bounding shape relative to the input part. Each factor controls the inflation of the bounded shape in a specific direction, independently of the other factors. **Constant Offset** specifies a constant offset distance by which the bounding part is inflated in all directions. **Individual Offsets** specifies individual offset distances for all directions, independently of each other, in the +/-X, +/-Y, and +/-Z directions.
Centroid Offset	Specifies the offset by which the bounding shape is translated. Set the offsets in the *X Offset*, *Y Offset*, and *Z Offset* entries.

Transform Parts

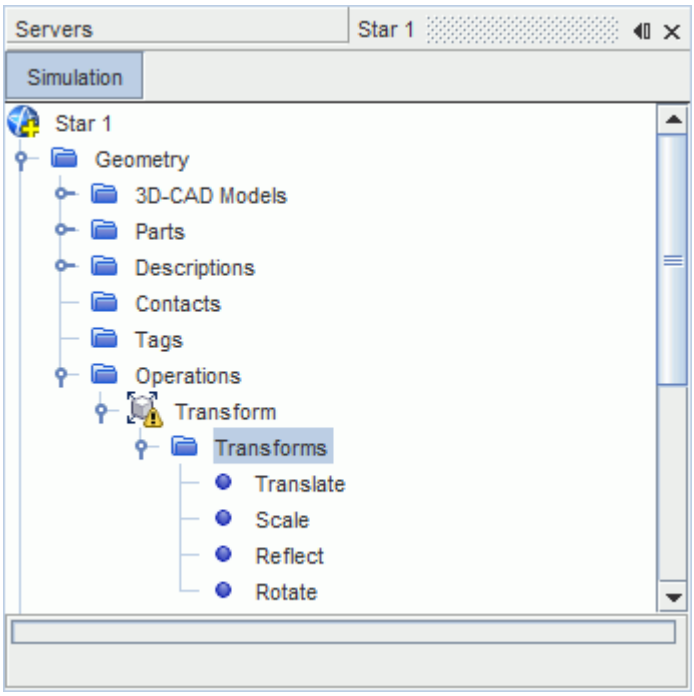
The Transform Parts operation allows you to perform transformations on the input parts. The available transformations include translate, rotate, reflect, scale and coordinate system transform.

To add a transformation to the Transform mesh operation, create a transform control:

1. Right-click the **Transform** > **Transforms** node and select **New** > **[Transform Control]**.



You can perform multiple transformations within the same Transform mesh operation, by creating multiple Transform controls.



The Transforms are applied in the order in which they appear in the tree. To change the order, right-click the **Transforms** node and select **Reorder...**, then use the *Reorder Transforms* dialog as required.

Transform operations generate a new description for all the input parts. When performing a transform operation on parts that have strong contacts, the contact type in the new description might change depending on whether the transformation is applied to one or both parts in contact. A strong contact retains its type only if both parts in contact are transformed. If the transformation is performed on one part only, the contact in the resulting description is weak.

Transform Control common Properties

The following properties are common to all types of transforms.

Enable Transform	Allows you to enable/disable the transform.
Index	Specifies a unique identifier for the transform.

Translate Properties

Coordinate System	Allows you to select the coordinate system for the translation.
Translation Vector	Specifies the translation along the X, Y, Z axes.

Rotate Properties

Coordinate System	Allows you to select the coordinate system for the rotation.
Origin	Specifies the origin for the transform in the specified coordinate system.
Rotation Axis	Specifies the axis about which the input parts are rotated.
Rotation Angle	Specifies the angle of rotation in the preferred units.

Scale Properties

Coordinate System	Allows you to select the coordinate system for the transform.
Origin	Specifies the origin for the transform in the specified coordinate system.
Scale Factor	Specifies the scale factor in each direction.

Reflect Properties

Coordinate System	Allows you to select the coordinate system for the transform.
Origin	Specifies the origin for the transform in the specified coordinate system

Normal Vector	Specifies the normal of the plane about which the input parts are reflected.

Coordinate Properties

From Coordinate System	Specifies the starting coordinate system.
To Coordinate System	Specifies the target coordinate system.

Boolean Mesh Operation Properties

Some boolean mesh operations have common properties which are described in this section.

The properties below apply to the following boolean mesh operations:

- Unite Parts
- Intersect Parts
- Subtract Parts

Properties

Input Parts	Specifies the geometry parts that the boolean operation is performed on.
Target Part (Subtract Parts operation only)	Specifies the single geometry part from which all other input parts are subtracted.
Output Parts	The name of the output parts that the operation generates (read-only).

Expert Properties

Perform CAD Boolean	This property allows you to choose if the boolean operation uses discrete surface mesh from the latest description of input parts for the boolean operation, or CAD bodies.	
	Activated	The boolean operation uses CAD bodies from the latest description for this operation. The root description of the output part has CAD bodies from which the discrete surface mesh is generated. If CAD bodies are not available on the latest description of input parts, the boolean operation performs a discrete boolean on the surface mesh.
	Deactivated	The boolean operation uses the discrete surface mesh from the latest description of input parts for

		this operation. The root description of the output part does not have any CAD bodies and the discrete surface mesh is generated from the input surface meshes. A sub-node appears under the boolean mesh operation node called **Remesh Intersection Curve**.

The **Imprint Parts** boolean operation has a unique set of properties which are described in the Imprint Properties section. See Imprint Properties.

Remesh Intersection Curve Properties

Remesh Intersection Curve	This property is available under a boolean mesh operation node when the boolean mesh operation property **Perform CAD Boolean** is deactivated.	
	Activated	Faces near the intersection curve in the output parts are remeshed. See Intersection Curve Remeshing.
	Deactivated	Deactivated by default. Faces near the intersection curve in the output parts are not remeshed.

See Also:

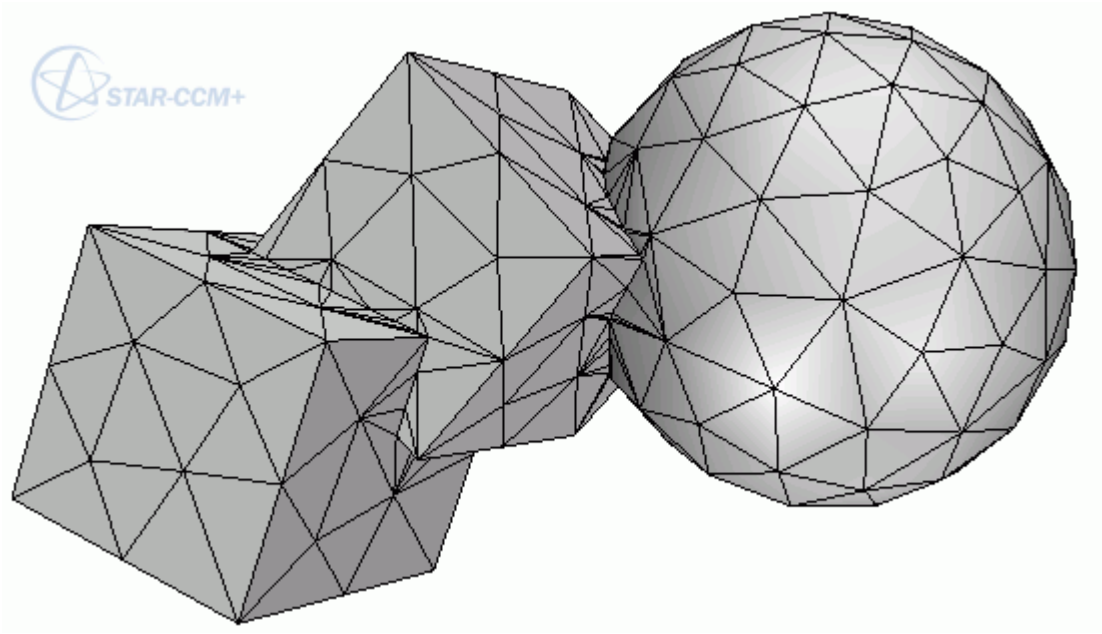
Inherited Parts

Intersection Curve Remeshing

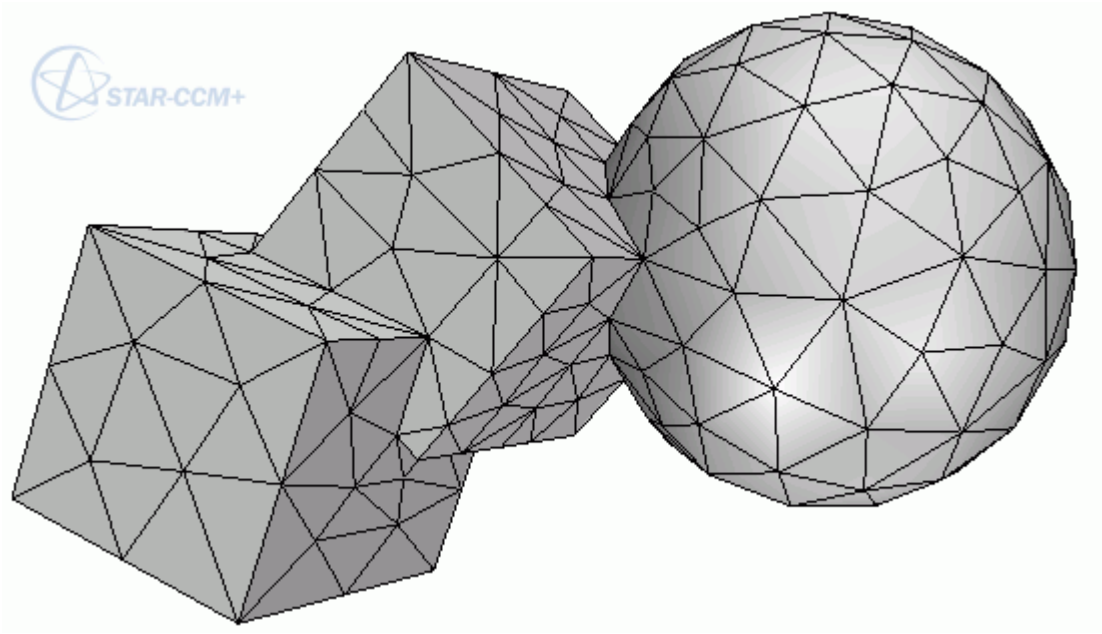
The *Remesh Intersection Curve* option is available when performing one of the following boolean mesh operations:

- Unite Parts
- Intersect Parts
- Subtract Parts

If the input parts for a mesh operation intersect each other, the mesh operation can produce poor quality faces around the intersection curve area on the surface of the output parts. The following image shows the output from a unite operation with *Remesh Intersection Curve* deactivated:



Activating the *Remesh Intersection Curve* option causes the surfaces of the output parts to be remeshed automaticaly after the mesh operation is executed.The following image shows the output from a unite operation with *Remesh Intersection Curve* activated:



Priority Indexing

When imprinting parts, at least one part is designated as the destination part and any other parts are designated as source parts. Priority Indexing lets you control which parts are the destination or source parts. The part or parts with the lowest index value have the highest priority. So any parts with the priority index of 1 are the destination parts.

The features and precise location of the destination parts are maintained during the imprint operation. However, the features of the source parts are modified to fit the features of the destination parts.

The imprint operation moves vertices of the source parts onto vertices of the destination parts. Faces from the source part that are coincident with a destination part are deleted and replaced with faces of the destination part.

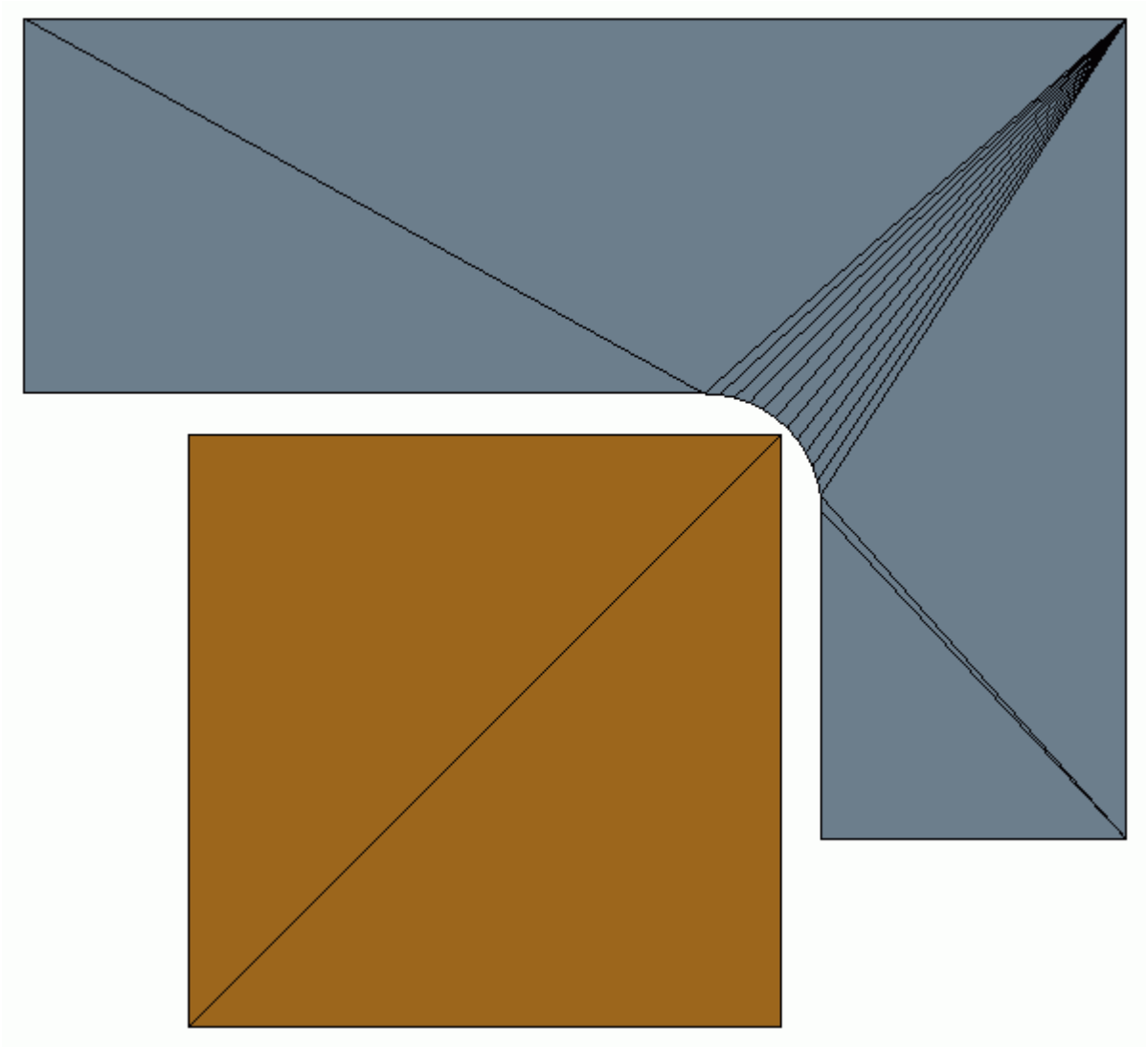
Imprint priority indexing can also be useful to control the division of shared faces into part surfaces.

Part surfaces of all faces in destination parts are preserved despite being re-tessellated. However, part surfaces in the source parts change since duplicate faces from source parts are deleted during imprinting.

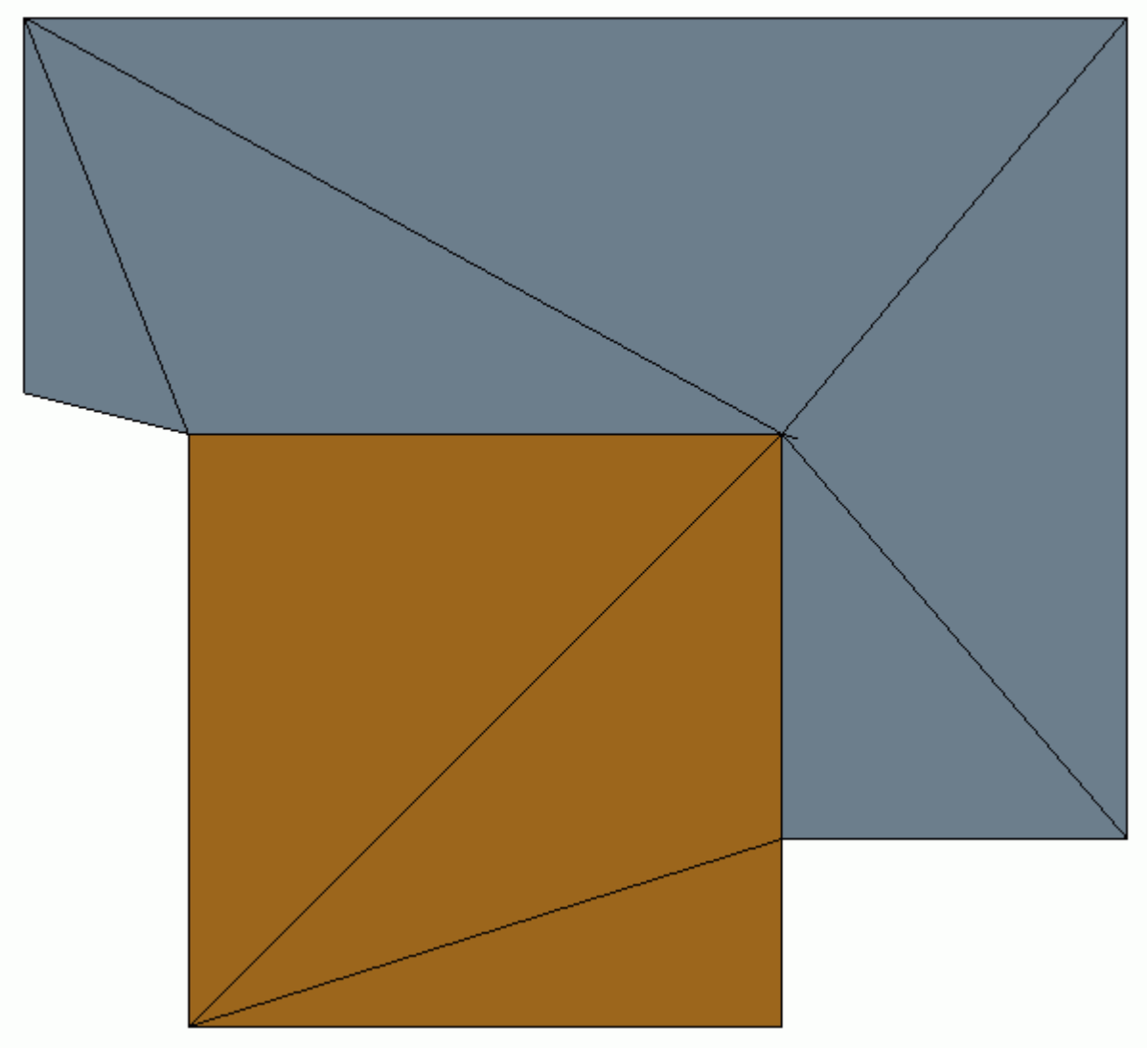
The number of interfaces that exist after imprinting depends on the assignment of priority indexes.

The Effect of Priority Indexing on Part Shapes

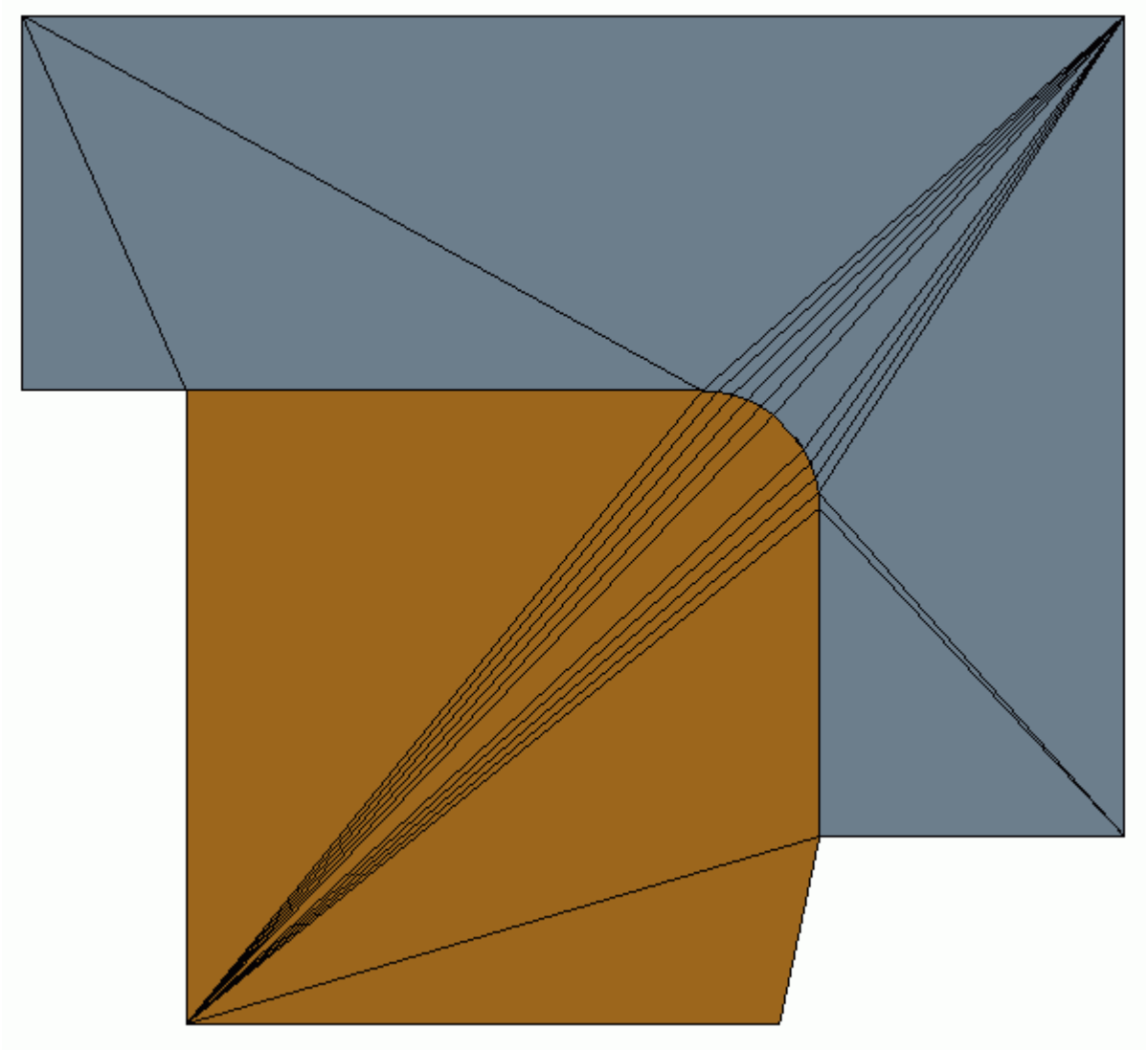
When imprinting the parts displayed below, you can specify either the L-shaped part or the block part as the destination part. Setting the priority index number of a part to 1, defines that part as the destination part.



When the priority index number of the block part is set to 1, the sharp corner of the block is preserved during imprinting:



When the priority index number of the L-shaped part is set to 1, the curve of the L-shaped part is preserved during imprinting:



Contents:

Imprinting Parts With Priority Indexing

See Also:

Imprint Parts

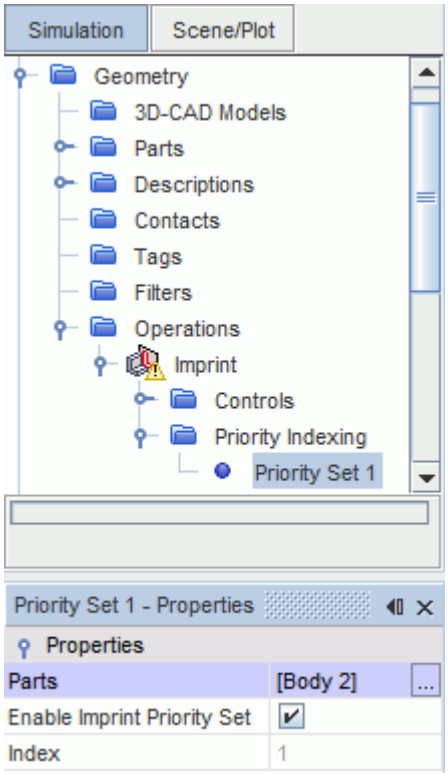
Imprinting Parts With Priority Indexing

Setting a priority index for an imprint operation allows you to have control over imprinting while maintaining the flexibility of performing operations as part of the parts-based meshing pipeline.

Before parts are assigned a priority index number, first create a priority set within the Imprint Operation and assign a part or parts to that priority set. You can then reorder priority sets within the simulation tree to change their priority index number. The higher in the tree a priority index is, the higher the priority it has.

A priority set has the following properties:

- **Parts**
- **Enable Imprint Priority Set**
- A read-only **Index** number



Priority sets are not shared between operations, but you can copy and paste them between operations.

Before creating an imprint operation with a priority index:

1. Import / create parts.
2. If necessary, split parts by patch. See Splitting by Patch.
3. Assign parts to regions. See Assigning Parts to Regions.
   - In the *Assign Parts to Regions* dialog, select:
     - **One region per part**
     - **One boundary per part surface**
     - **One feature curve for all part curves**

   Interfaces are created between part surfaces that are in close proximity.

To create an imprint operation with a priority index:

1. Create an imprint operation.
   a. Right-click **Geometry** > **Operations** and select **New** > **Imprint**.
   b. In the *Create Imprint Operation* dialog, select the parts to include in the imprint operation.

c. Select the**Imprint** node and set the properties that you require. See Imprint Properties.

d. Click **OK**.

2. Set the priority index of the parts you have selected to merge.

a. Right-click **Geometry** > **Operations** > **Imprint** > **Priority Indexing** and select **New**.

A new **Priority Indexing** > **Priority Set 1** node appears.

b. Select the **Priority Set 1**node.

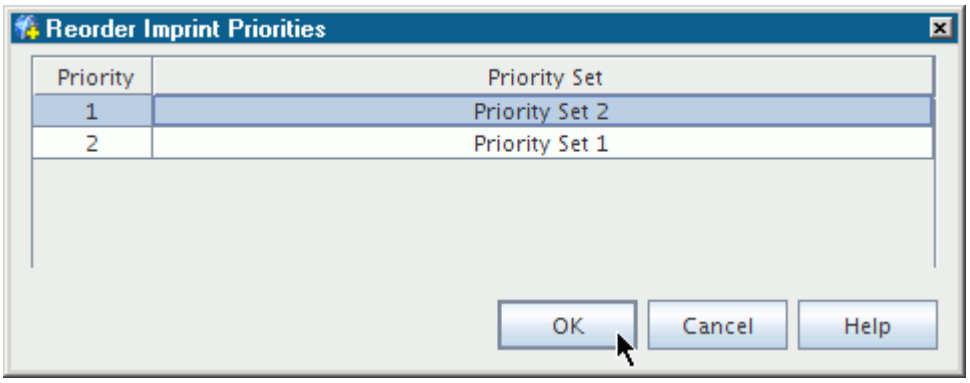
c. In the *Priority Set 1 - Properties* window, set *Parts* to the destination part for the imprint operation.

d. Repeat the previous step to create additional priority indexes for the source parts if you require them.

3. If necessary, reorder priority sets:

a. Right-click the **Priority Indexing** folder and select **Reorder**.

b. In the *Reorder Imprint Priorities* dialog, click and drag the priority sets then release them at a suitable point in the table. The part with Priority 1 is the destination part.



c. When the priority sets are in the correct order, click **OK**.

Note: It is possible to place a single part, or composites containing it, in multiple priority sets. If conflict occurs between two priority sets that contain the same part, the priority set with the highest index value is selected first.

4. Execute the **Imprint** Operation.

a. Right-click the **Operations** > **Imprint** node and select **Execute**.

b. To display the imprinted parts, right-click an empty space in the *Graphics* window and select **Apply Representation** > **Imprint**.

Note: Priority sets that are deactivated and priority sets that contain no parts are displayed with a gray icon.

See Also:

Imprint Parts

Inherited Parts

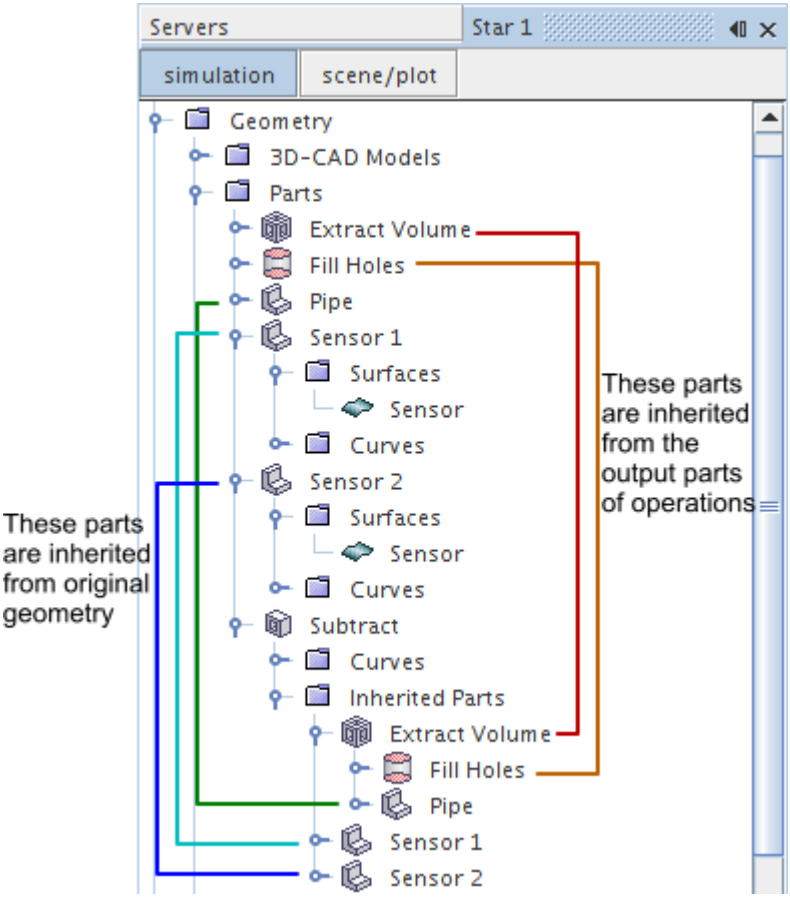
Inherited parts are the input parts to an operation, which are categorized separately from any new surface and curves that the operation creates. This categorization makes it easier to find and apply the input parts to other operations and mesh controls.

The following operations generate inherited parts:

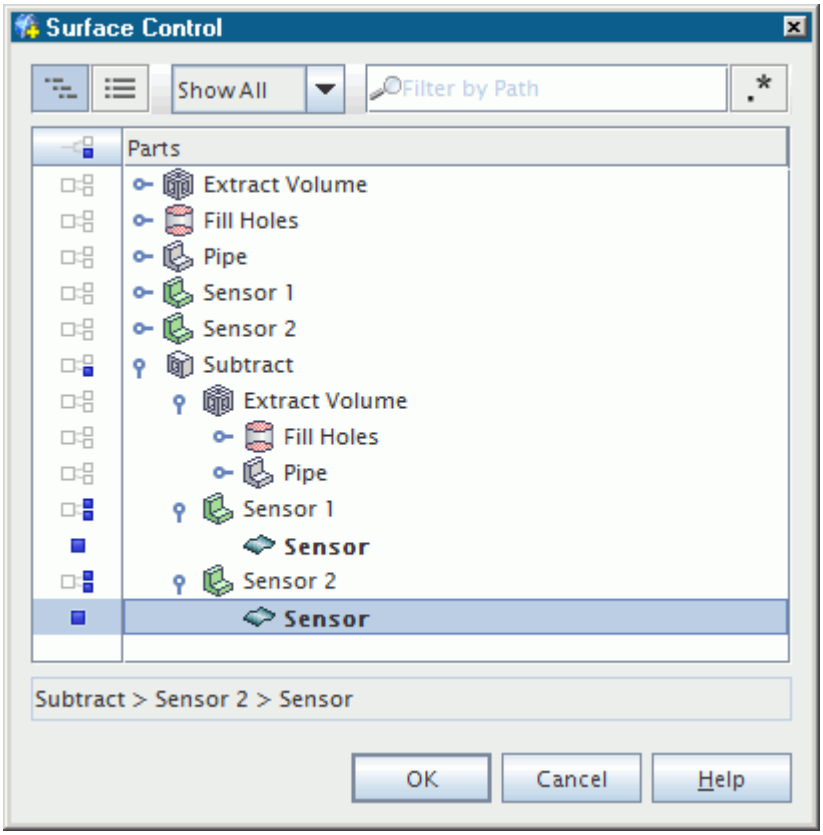
- Surface Wrapper (with the Per-Part Meshing property deactivated)
- Extract Volume
- Unite Parts
- Intersect Parts
- Subtract Parts

Example of Using Inherited Parts for Mesh Controls

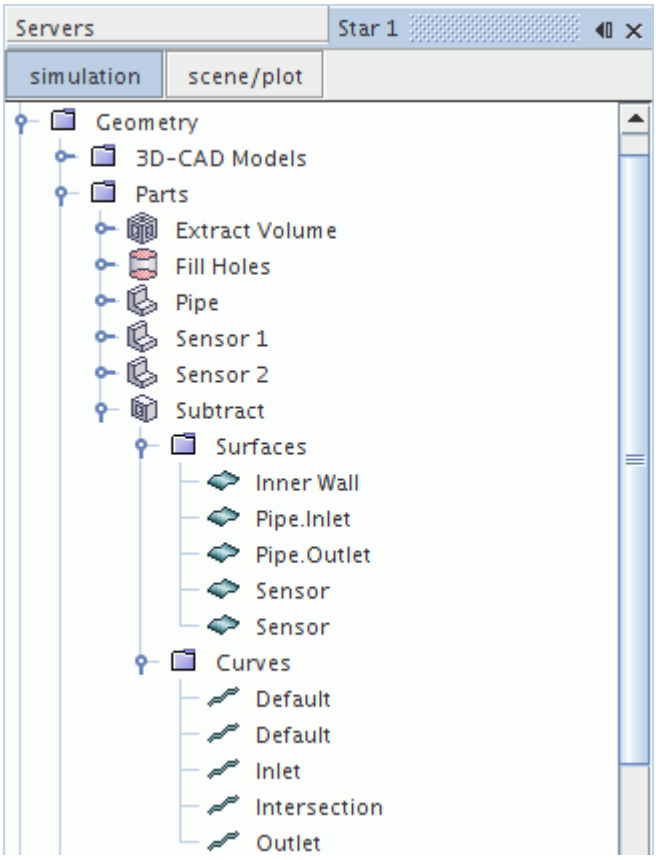
The following example uses the output part from a subtract operation that has several inherited parts. The following image illustrates this example and how the inherited parts are organized.



When using this part as the input part for an automated mesh operation, the categorization of inherited parts makes it easier to find and apply custom controls to the sensor surfaces.



For comparison, the following image shows the categorization for the same subtract part with a flat list of surfaces and curves. This image shows that it is not obvious which parts are associated with the Sensor surfaces and Default curves.



Detaching Parts from Operations

Some operations create parts that contain the output of that operation. The output parts are linked to the operation so that when you make a change to the operation, the output parts are updated. You can break the link between the part and the operation by detaching the parts.

To detach a part from an operation:

1. Right-click the desired output part and select **Detach**.

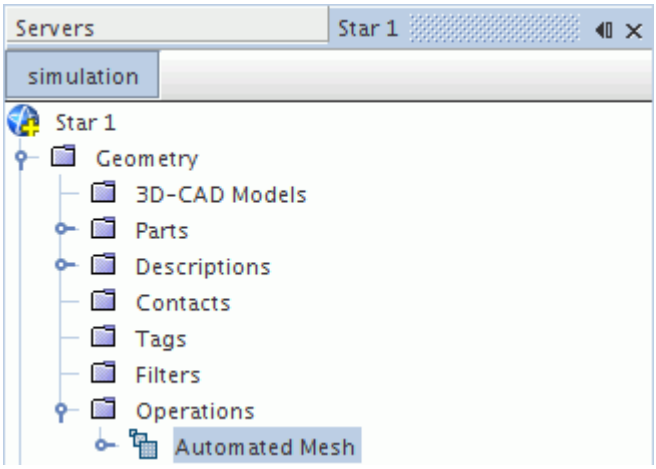
   You can only detach output parts whose associated operation has been executed.

   The link between the output part and the associated operation is broken. Changes that you make in the operation are no longer passed down to the output part.

Automated Mesh

The automated mesh operation generates surface and volumes meshes for its input parts. It manages all of the meshers, default controls, and custom controls.

The **Automated Mesh** node represents this operation in the simulation tree.



STAR-CCM+ automatically removes any volume mesh generated using an **Automated Mesh** operation, when you do any of the following:

- remove the associated input part from all volume mesh operations
- assign the associated input part to a different region
- delete the associated input part

Contents:

Create Automated Mesh Operation Dialog
Select Meshers Dialog
Automated Mesh Properties

Create Automated Mesh Operation Dialog

Use the *Create Automated Mesh Operation* dialog to select the input parts and meshers for the automated mesh operation.

Parts	Specifies the geometry parts for which the operation generates a mesh.
Surface Meshers	Provides the available meshers for generating a surface mesh. Currently, the following surface meshers are available: • Surface Remesher
Optional Surface Meshers	Provides the optional meshers for generating a surface mesh. Currently, the following optional surface meshers are available: • Automatic Surface Repair
Core Volume Meshers	Provides the available meshers for generating a volume mesh. Currently, the following volume meshers are available: • Polyhedral Mesher • Tetrahedral Mesher • Trimmed Cell Mesher

Optional Volume Meshers	Provides the optional meshers for generating a volume mesh. Currently, the following optional volume meshers are available: • Thin Mesher
Optional Boundary Layer Meshers	Provides the optional meshers for generating a boundary layer mesh. Currently, the following optional boundary layer meshers are available: • Prism Layer Mesher • Advancing Layer Mesher
Enabled Meshers	Shows the meshers that you have selected. Deactivating the check box for any of these meshers removes it from the mesh operation.

See Also:

Mesh Operations

Extract Volume

Select Meshers Dialog

Use the *Select Meshers* dialog to redefine the meshers for the automated mesh operation.

Surface Meshers	Provides the available meshers for generating a surface mesh. Currently, the following surface meshers are available: • Surface Remesher
Optional Surface Meshers	Provides the optional meshers for generating a surface mesh. Currently, the following optional surface meshers are available: • Automatic Surface Repair
Core Volume Meshers	Provides the available meshers for generating a core volume mesh. Currently, the following core volume meshers are available: • Polyhedral Mesher • Tetrahedral Mesher • Trimmed Cell Mesher
Optional Volume Meshers	Provides the optional meshers for generating a volume mesh. Currently, the following optional volume meshers are available: • Thin Mesher
Optional Boundary Layer Meshers	Provides the optional meshers for generating a boundary layer mesh. Currently, the following optional boundary layer meshers are available: • Prism Layer Mesher • Advancing Layer Mesher
Enabled Meshers	Shows the meshers that you have selected. Deactivating the check box for any of these meshers removes it from the mesh operation.

See Also:

Mesh Operations

Extract Volume

Automated Mesh Properties

Use the **Automated Mesh** properties and expert properties to define the behavior of the operation.

Properties

Per-Part Meshing	**Activated**	Meshes each part individually, which ignores any interfaces between parts. Therefore the mesh between connected parts is not conformal. This option is useful if the trimmer or thin mesher cannot produce a mesh at the interface.
	Deactivated	Meshes all the input parts at once.
Mesher Execution Mode	Allows you to set the execution mode when running parallel servers.	
	Serial	Executes the meshers on the main processor only.
	Parallel	Executes the meshers on parallel processors, depending on how you launch the simulation. See Parallel Meshing. Parts are subdivided between processors.
	Concurrent	Executes the meshers on separate processors for each individual part. Only available to use with the surface remesher, surface wrapper, trimmed, polyhedral, and tetrahedral meshers - with or without the prism layer mesher. See Concurrent Meshing. When this option is selected, the default control Part Priority appears which allows you to define the order in which parts are meshed concurrently.
Input Parts	Specifies the geometry parts for which the operation generates a mesh.	

Expert Properties

Preserve Surface Perimeters	**None**	Maintains only surface that you have assigned to boundaries.
	Major Mesh Controls Surface Perimeters	Maintains the surface perimeters for surfaces to which you have applied custom surface controls for the prism layer mesher only.
	All Mesh Controls Surface Perimeters	Maintains the surface perimeters for all surfaces to which you have applied custom surface controls.
	All Part Surface Perimeters	Maintains all of the surface perimeters in the surface and volume mesh.
Verbose Output	**Activated**	Provides detailed meshing information in the *Output* window. This option is useful for debugging meshing problems.
	Deactivated	Provides the standard meshing information in the *Output* window.

See Also:

Mesh Operations

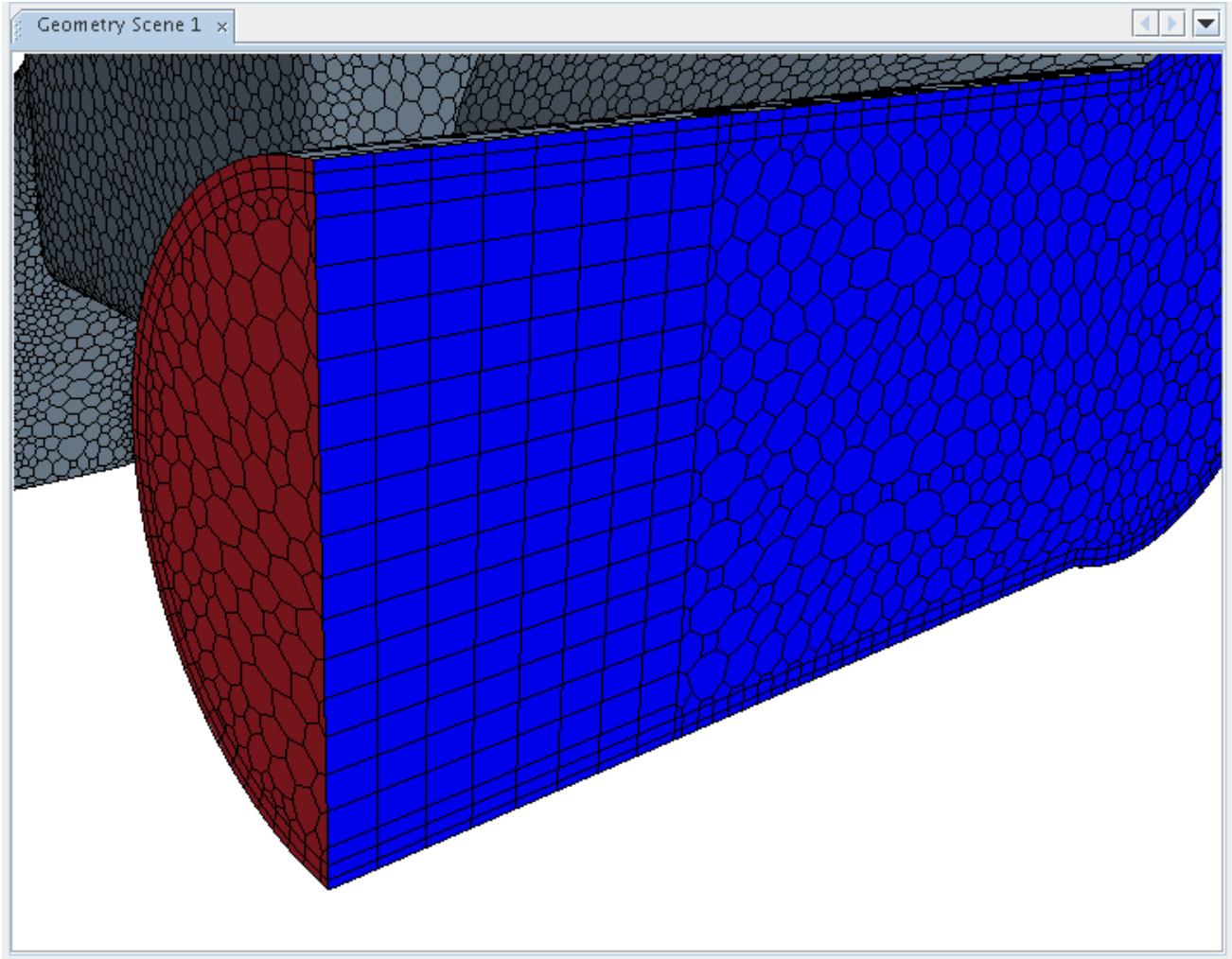
Extract Volume

Extruding Surfaces from a Parts-Based Volume Mesh

Use the extruder to extend the existing volume mesh beyond the original dimensions of the starting surface, normal to the existing volume mesh, or in a specified direction.

The extruder is typically used to extend the simulation domain with orthogonal extruded cells for inlet and outlet boundaries, and for special applications such as in catalysts.

Extending the inlet and outlet boundaries is useful for certain types of geometries, such as curved or short pipes near the boundary surface, where the flow would not otherwise have enough distance to fully develop in the original shape of the geometry. Using orthogonal extruded cells to extend the volume mesh at these boundaries allows the flow to develop fully while maintaining a low cell count. In the following example, a small simulation domain is extended using the Extruder in parts.



The extruder mesher is made up of two operations:

- **Surface Extruder** — The surface extruder extends the edges of the input surface to form the bounding surfaces of the extruded region. You specify the extrusion distance in the surface extruder operation.
- **Volume Extruder** — The volume extruder fills in the extruded region with extruded volume cells from the input surface. You control the number of divisions in the extruded region of the volume extruder operation.

To use the extruder mesher using parts-based meshing:

1. Set up and generate a parts-based volume mesh for the simulation domain. See The Meshing Workflow.

   The surfaces of the volume mesh act as a starting point for the extruder mesh.
2. Create and define the surface extruder operation: Right-click the **Operations** node and select **New** > **Surface Preparation** > **Surface Extruder**.
3. In the *Create Surface Extruder Operation* dialog:
   a) Set *Input Parts* to the part that you want to extrude.
   b) Set *Input Part Surfaces* to the surface that you want to extrude.

      You can only choose surfaces that are associated with the selected input parts.
   c) In *Create New Part*, choose how new parts are treated in the extrusion. See Surface Extruder: Create New Part.

d) In *Output Part Surfaces*, choose how the input part surface is to be extruded. See Surface Extruder: Output Part Surfaces.

If only one part surface is chosen, these options will produce the same result. However, the assignment of the boundaries may differ, depending on the geometry.

e) Set the *Extrusion Distance*.

f) Set the options for creating the surface and volume extruder operations.

To automatically create the Volume Extruder operation, activate the **Create Volume Extruder Operation from Output Parts** option.

To execute the Surface Extruder operation upon closing the *Create Surface Extruder Operation* dialog, activate the **Execute Surface Extruder Upon Creation** option.

g) Click **OK**.

A **Surface Extruder** node is added to the **Operations** node and automatically executed. Here, you can change the direction and distance properties. See Surface Extruder Properties. A **Volume Extruder** node is also added to the **Operations** node. See Volume Extruder Properties.

4. If required, create additional surface extruder operations to extend the simulation domain in multiple directions, and by different amounts.

   This is not necessary if you want to extrude the simulation domain by the same amount in all directions.

5. Set-up and execute the Volume Extruder operation:

   a) Expand the **Operations** > **Volume Extruder** > **Controls** node.

   b) Select the **Set the Number of Layers** node and set *Number of Layers* to the desired amount.

   c) Select the **Stretching Function** node and choose a *Stretching Method* to specify the cell-distribution along the extruded distance.

   For more details, see Volume Extruder Controls.

Assign the extruded parts to regions:

6. Right-click the **Surface Extruder** node and select **Assign Parts to Regions**. See Assigning Geometry Parts to Regions

   If you have multiple surface extruder operations, assign all extruded parts to a separate region.

7. To create multiple extruded regions with different parameters, such as increasing the number of layers, create additional Volume Extruder operations as described in the previous steps.

   Note: If a number of extruded regions have the same properties, additional volume extruder operations are not required. In this case, simply multi-select the surface extruder parts before setting the properties.

8. Right-click the **Operations** > **Volume Extruder** node and select **Execute**.

The selected regions are extended.

Replacing a Part in a Parts-Based Volume Mesh

Replacing a part is useful to speed up the workflow when exploring multiple designs of a particular component. Including the replace part functionality as part of the meshing pipeline can further speed up the workflow.

Without the replace part capability of STAR-CCM+, you would have to switch between parts and manually update the mesh operations and custom controls in your pipeline for each new part. Using the replace part capability removes the need to manually update the downstream mesh operations and custom controls when you switch the source part.

When replacing a part in parts-based meshing, you require a placeholder to hold the replacement part. This placeholder is the target part in the replace part operation. In this context, "part" can refer to an individual part, or a composite part/assembly (a collection of parts grouped together). When replacing a composite part, both the target and source parts must be similar (for example, the source part must be an assembly if the target is an assembly).

The requirements for the target part are listed below:

- You can replace the following type of parts:
  - Leaf-level CAD part
  - Leaf-level mesh part
  - Detached mesh operation part
  - Composite part
- You cannot replace the following type of parts:
  - 3D-CAD part
  - Simple shape part
  - Solid Model part
  - Mesh operation part

You can replace a target part with an existing part from the **Geometry** > **Parts** node or from an external file. When importing from an external file, you can only import a discrete mesh file to replace a discrete leaf-level mesh target part. You can only replace a target CAD part with another existing CAD part. You can replace a discrete part with another discrete part or a CAD part. When choosing a CAD file as a source part for a discrete mesh target part, an additional **Tessellation Options** node appears under the replace part operation node. Here, you can define the tessellation parameters.

To replace a part within the meshing pipeline:

1. Right-click on a supported part that you want to replace and select **Duplicate**.

   This step creates a leaf-level placeholder part that will contain the replacement part in subsequent steps.
2. Right-click the **Geometry** > **Operations** node and select **New** > **Surface Preparation** > **Replace Part**.
3. In the *Replace Part* dialog:
   a) Choose a method for how you want to replace the part: from an existing part in the simulation tree, or from an external file.
   b) Select a source part or source file to.

      This part or file replaces the target part.
   c) Select the target part.

      The source part or external file replaces this part. Typically, the target part is the duplicate leaf-level part from step 1.
   d) Click **OK**.

   See Replace Part Operations Properties.

When replacing a composite part, the operation works through the source assembly and finds the equivalent part in the target assembly. The mesh is then replaced on the target parts if they match a source part. The operation also matches parts at the same hierarchy using part names. The name at the top level of the assembly (the name of the source part and target part) are not matched.

4. Use the placeholder part from step 1 in all subsequent mesh operations instead of the original part.
5. Set up the remaining simulation settings and run the simulation.
6. Change the source part in the **Replace Part** operation to the desired part.

Contents:

Replace Part Operation Properties

Replace Part Operation Properties

The replace part operation allows you to replace parts in a simulation as part of the simulation pipeline.

Obtain Mesh From	Specifies how the part is replaced. Select **Existing Part** to replace the target part with one that is located in the **Geometry** > **Parts** node. Select **File** to use an external file for the replacement.
Source Part	Specifies an existing part to replace the target part. Choose an existing part from the **Geometry** > **Parts** node. This property becomes active when you set *Obtain Mesh From* to **Existing Part**.
File	Specifies an external file that replaces the target part. This property becomes active when you set *Obtain Mesh From* to **File**.
Target Part	Specifies the part to replace in the pipeline. Typically this is a leaf-level part that you create to contain the replaceable part in the pipeline. For a list of parts that you can use as a target part, see Replacing a Part in a Parts-Based Mesh.
Default Target Part Surface/ Curve/Point	All unmapped faces, edges, and vertices are assigned to this part surface/ curve/point respectively.

Assembly Import Properties

The assembly import properties appear when replacing composite parts.

Clear Existing Parts	When activated, clears the mesh on all existing parts in the target assembly. By default, this option is activated. Leave this option activated to retain the unmatched parts in the target assembly and clear the mesh. In this workflow, the parts are preserved so that downstream operations and controls retain the parts in their inputs. This option is useful if you expect to bring the parts back again in the future. Deactivate this option to retain the unmatched parts in the target assembly, including the mesh. This is useful when you are replacing only a few parts in the target assembly and wish to retain all other existing parts in the target assembly.

Create New Parts	When activated, creates new parts in the target assembly for unmatched parts from the source assembly. By default, this option is activated.
Delete Empty Parts	When activated, deletes empty parts in the target assembly. By default, this option is deactivated. Activate this option in conjunction with the **Clear Existing Parts** option to delete all unmatched parts from the target assembly.

Mesh Descriptions

A mesh description is the different versions of a surface mesh that each geometry part has. You create and change descriptions by importing surfaces or executing operations.

The following list summarizes the main types of mesh descriptions:

- Root Descriptions — This description holds the surface mesh from the tessellation of the geometry parts.
- Operation Descriptions — This description holds the surface mesh that was created or changed as a result of a mesh operation.
- Latest Description — This description holds the surface mesh from the most recent mesh operation that you executed on the geometry parts. It appears along with the root description when the part is first imported or created.

The following meshing operations create or change operation descriptions:

- Automated Mesh
- Surface Wrapper (with the Per-Part Meshing property activated)
- Imprint Parts

Whenever you execute a mesh operation on a part, the operation acts on the last description in the tree which contains that part.

If you set a scene to display the latest description, and then you remesh the surface or perform an imprint operation, the scene automatically updates to display the latest description.

Note: Other mesh operations create new parts, not descriptions. If you want to see the result of these operations, modify the visible parts in the current scene.

Contents:

Visualizing Mesh Descriptions
Exporting Mesh Descriptions
Description Properties
Description Context Menu

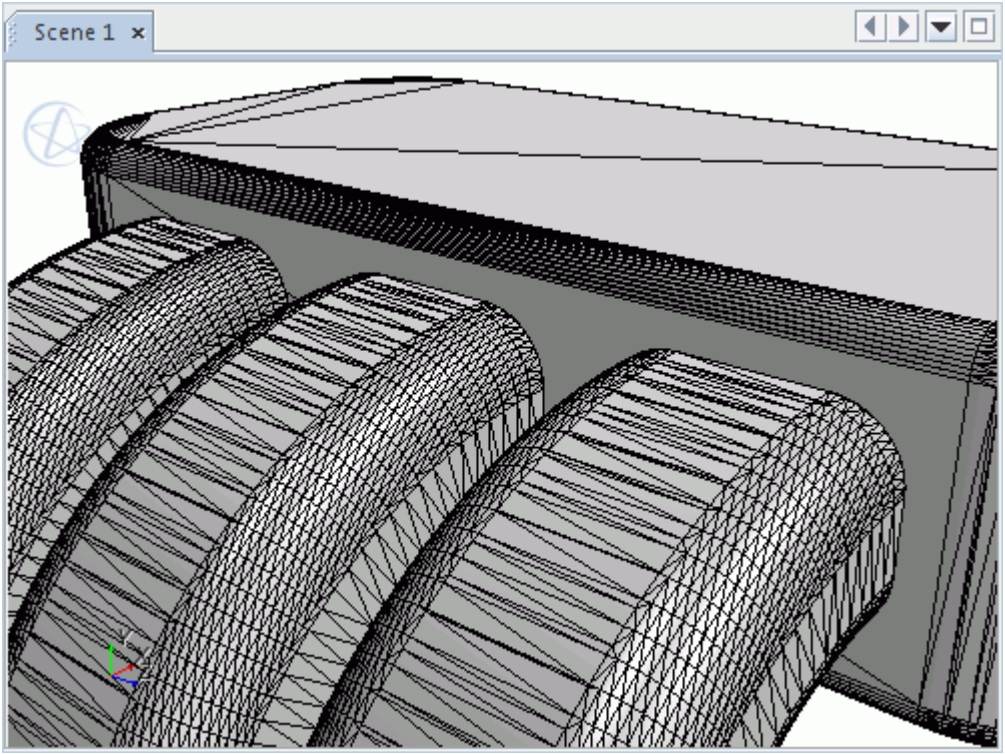
Visualizing Mesh Descriptions

You can use descriptions to visualize the effect of a mesh operation on a geometry part.

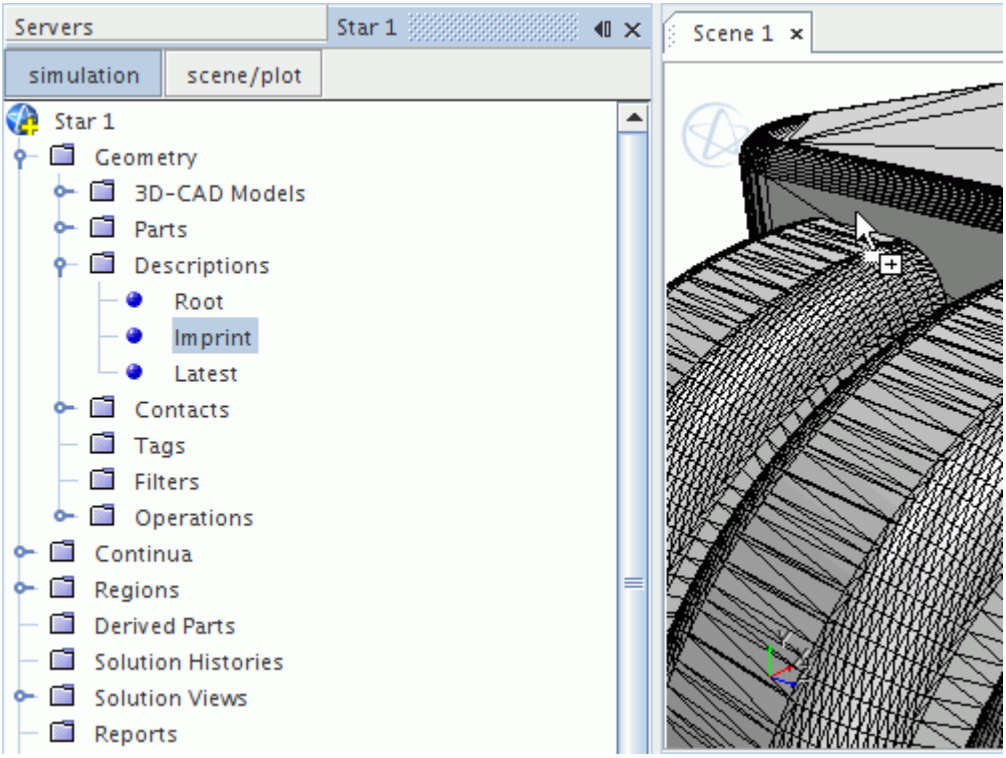
To visualize a mesh description:

1. Make sure that you have a mesh scene open and the geometry part is visible in the scene.

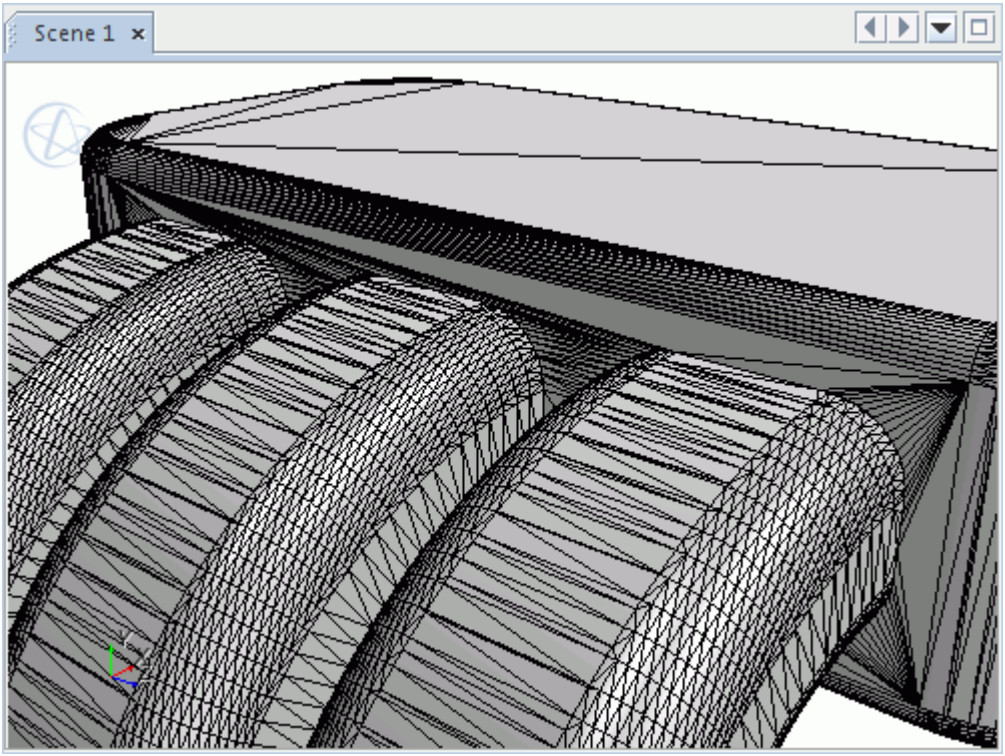
The following image shows the surface tessellation of an imported CAD part. This description is the root description.



2. Drag and drop the required description node into the scene.



The following image shows the imprinted surface mesh.



See Also:

Custom Mesh Controls

Previewing Parts from Mesh Operations

You can preview parts that are created by pre-executed mesh operations. Having this capability is useful for identifying parts and part surfaces when setting up interfaces or custom controls. When previewing, you see a basic surface mesh for all empty mesh operations that are in the `pre-execute` or `out-of-date` state.

To preview mesh operations:

1. Set up a scene to display the mesh operation parts or part surfaces.
2. Right-click the desired mesh operation and select **Pre-Execute**.
3. Select the **Geometry** > **Descriptions** > **Latest** node and activate the **Preview Mesh Operations Parts** option.

Mesh operation parts and part surfaces that are in the latest description in the scene are displayed. The mesh is obtained from the inputs to the mesh operation. If the inputs are empty mesh operation parts, then STAR-CCM+ searches recursively up the mesh operations inputs to find the part with a valid mesh. With this option activated, you can select and highlight parts and part surfaces from the pre-execute mesh operation parts. This function is useful when working with custom controls.

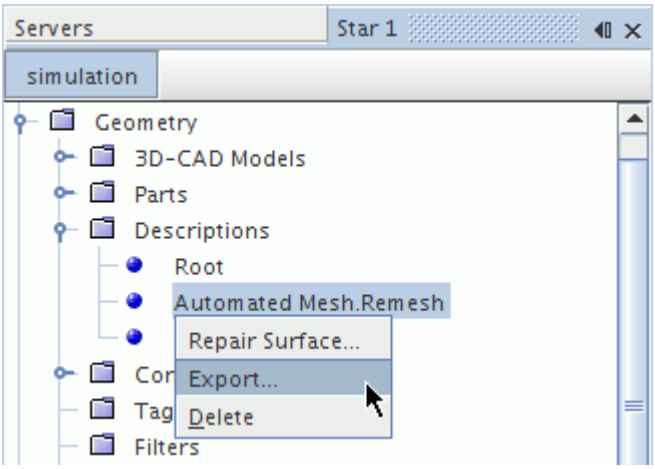
Exporting Mesh Descriptions

You can export any Parts-Based description to the surface mesh in another STAR-CCM+ session or third-party software.

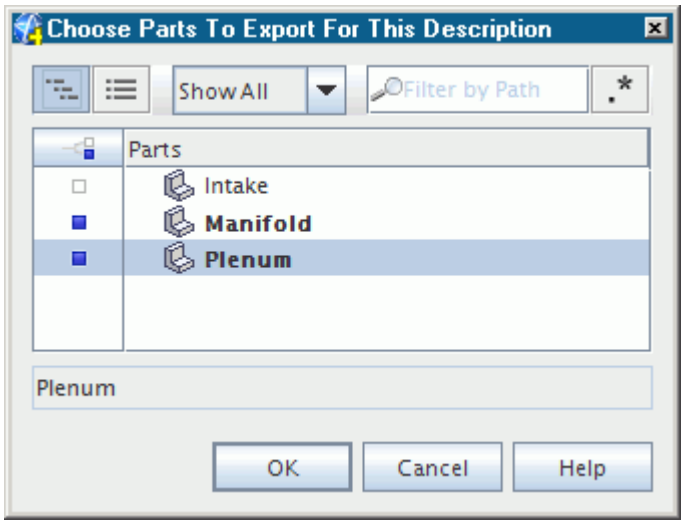
Several operations create descriptions which you can export – including the remesher, imprint operation, and transform operation.

To export a mesh description:

1. Right-click a mesh description and select **Export**.



2. In the *Choose Parts To Export For This Description* dialog, select the parts that you want to export.



3. In the *Save* dialog, set the *File Name* and click **Save**.

Description Properties

The description node properties display information on the selected description.

Described Parts	The geometry parts to which the selected description applies (read-only).
Faces	The number of faces in the surface mesh that the selected description has (read-only).
Vertices	The number of vertices in the surface mesh that the selected description has (read-only).

Description Context Menu

The description context menus can repair and export the description surface mesh.

Repair Surface	Launches the *Surface Repair* tool so that you can manually fix surface mesh issues for the selected description.
Export	Exports the surface mesh for the selected description to one of the following file types: • .dbs — pro-STAR Surface Mesh • .stl — Stereolithography File • .nas — Nastran Shells • .pat — PATRAN Shells • .inp — pro-STAR Shell Input (cell/vertex) File • .fro — FELISA Front Surface • .jt — JT Open

See Also:

The Surface Repair Tool
Exporting Mesh Descriptions

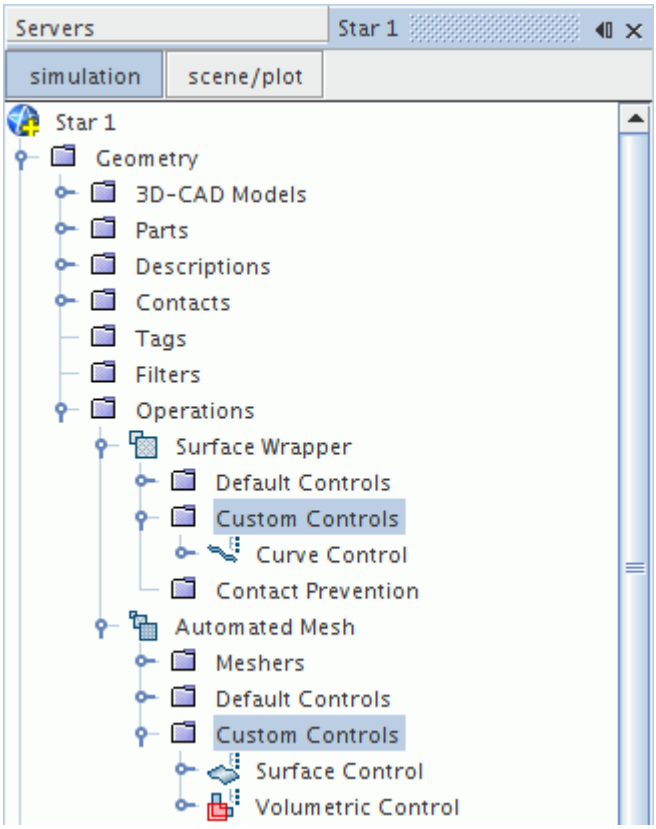
Custom Mesh Controls

Custom controls override any default controls for the surface and volume meshers. This facility allows you to refine or coarsen the mesh for part curves, part surfaces, geometry parts, and specified volumes.

The following types of custom mesh controls are available:

- Curve Controls
- Surface Controls
- Part Controls
- Volumetric Controls

Custom mesh controls are available for the **Surface Wrapper** and **Automated Mesh** operations.



Contents:

Custom Mesh Control Properties
Custom Mesh Control Context Menu
Curve Control
Surface Control
Part Control
Volumetric Control
Mesh Refinement in a Specified Volume
Query Custom Mesh Controls
Mesh Refinement in a Wake

See Also:

Visualizing Mesh Descriptions
Creating and Defining Custom Controls

Custom Mesh Control Properties

Use the custom mesh controls to define the input parts, part curves, or part surfaces, display the entities that belong to the control, and deactivate the control.

Parts	Specifies the geometry parts, part curves, or part surfaces to which the mesh control applies.
Part Curves	
Part Surfaces	
Apply Only to contacting area (Surface Control only)	Allows you to apply specific controls on contacting surfaces only. You can use this property to avoid splitting the contacting surface from the default surface. However, the surfaces must form a strong contact with a conformal interface.

Custom Mesh Control Expert Properties

Enable Control	Allows you to activate or deactivate the control without having to delete or recreate it.	
Controls Display Mode	**All**	Displays all parts, part curves, or part surfaces under the **Parts** manager.
	Customized	Displays only the parts, part curves, or part surfaces to which the mesh control applies under the **Parts** manager. This option quickly identifies the entities that the control affects.

See Also:

Creating and Defining Custom Controls

Custom Mesh Control Context Menu

Custom Controls Manager

Use the context menu for the custom controls manager to create custom controls.

New	Generates a new mesh control. The following controls are available: • Curve Controls • Surface Controls • Part Controls • Volumetric Controls

Control Nodes

Use the context menu for individual controls to duplicate the selected control.

Duplicate	Creates an exact copy of the selected control.

See Also:

Creating and Defining Custom Controls

Curve Control

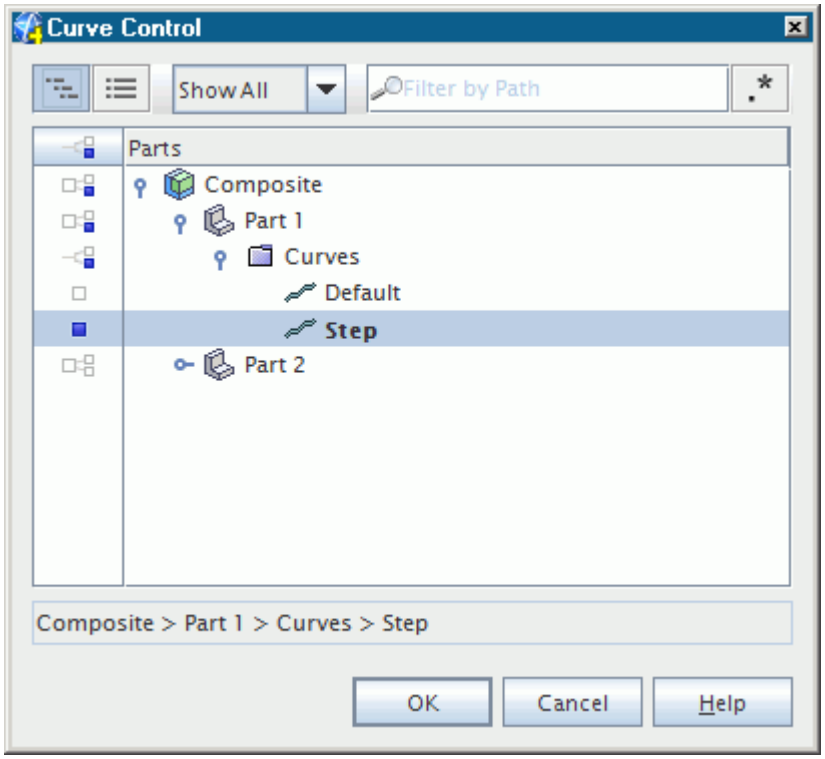
Curve controls specify alternative surface mesh settings for part curves.

If a part curve belongs to several controls, the following hierarchy determines the curve control that takes precedence. This hierarchy depends on how you apply the part curve to the control.

- Selecting the individual part curves.

  Only the individual part curve that is directly selected belongs to the curve control.

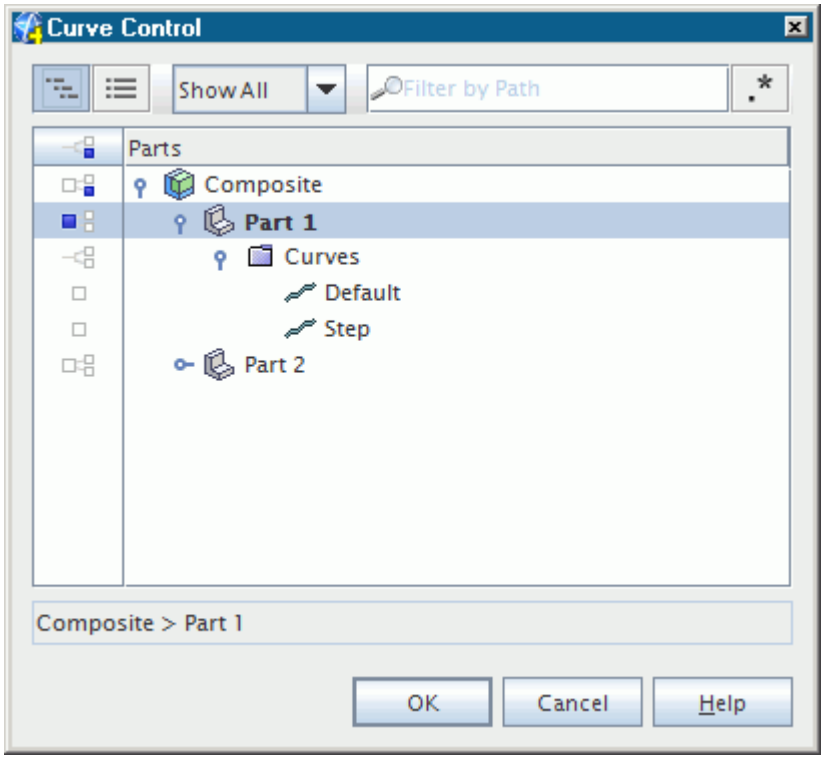
  Any curve controls that are applied directly to an individual part curve take priority over curve controls that are applied to the geometry part or composite part.



- Selecting a geometry part that includes the part curves.

  All part curves that comprise the selected geometry part are included in the curve control, even if they do not appear selected individually.

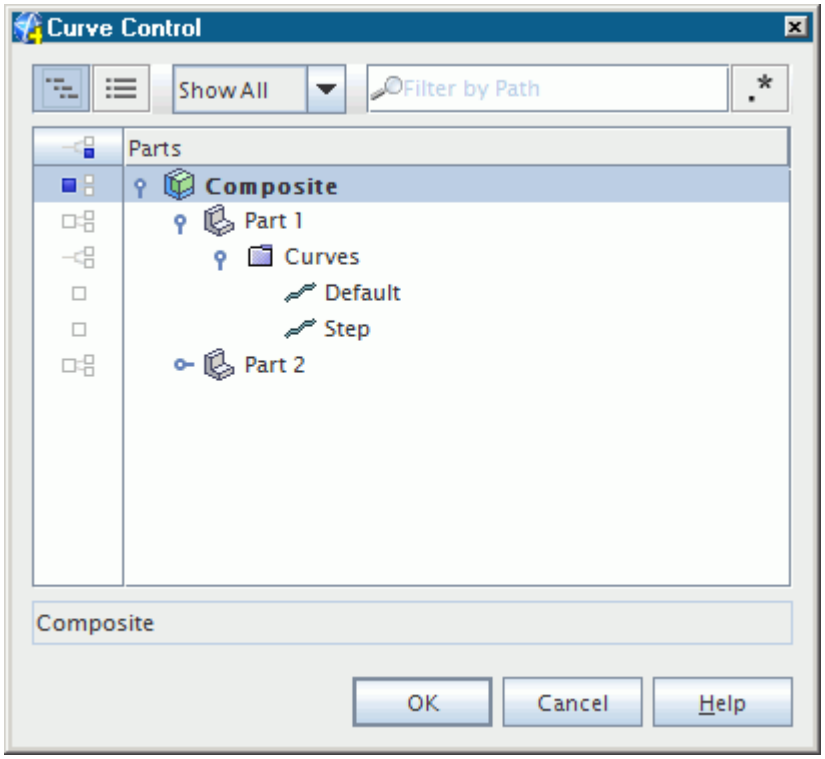
  Any curve controls that are applied to geometry parts take priority over curve controls that are applied to composite parts.



- Selecting a composite part that includes the part curves.

  All part curves on all geometry parts that comprise the composite part are included in the curve control, even if they do not appear selected individually.

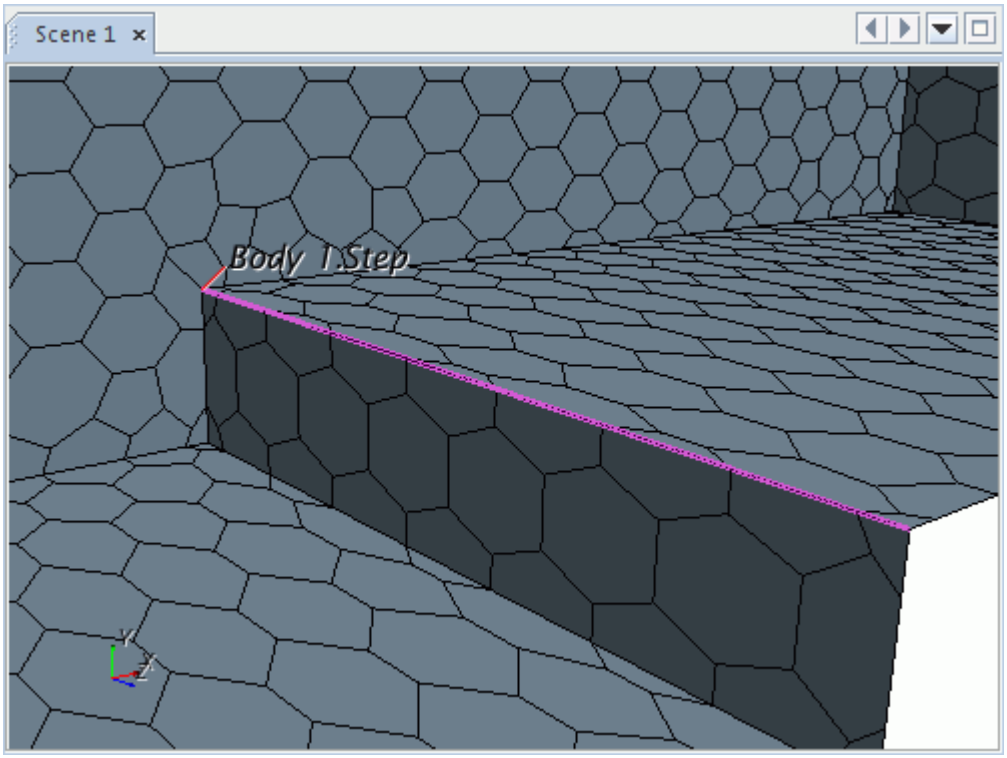
  However, curve controls that are applied to the geometry part or directly to an individual part curve, take priority over curve controls that are applied to the composite part. For composite parts within composite parts, curve controls that are applied to the lowest level composite part take priority.



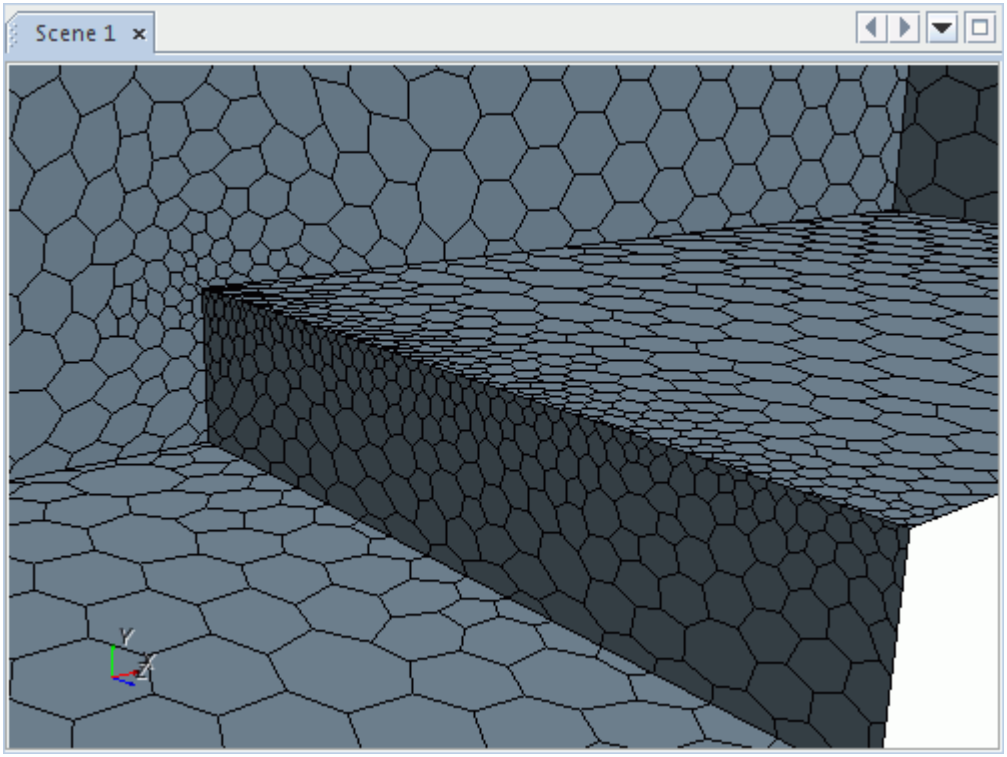
In addition, if a part curve belongs to several controls at the same level in this hierarchy, the control that produces the smallest cell size takes precedence.

Example of a Curve Control

This example refines the mesh along a feature of interest. The following image shows a part curve that defines the edge of a step.



Refining the surface mesh on the curve by 20% gives a higher density of surface discretization. This dense surface mesh produces a dense volume mesh around the feature.



Contents:

Curve Controls and Values

See Also:

Creating and Defining Custom Controls
Custom Mesh Control Context Menu
Custom Mesh Control Properties

Curve Controls and Values

Use the curve controls and values to define the mesh controls that you want to modify.

You can modify several mesh controls with a curve control:

Controls - Non Wake Refinement Properties

For the mesh controls that are listed, you can specify the properties as relative or absolute.

- Minimum Surface Size
- Target Surface Size

Control	**Specify Custom Value**	Allows you to specify a value for this control that overrides the default control.
	Use Parent Value	Uses the same value as the default control.

Controls - Wake Refinement Properties

The curve controls and values for wake refinement are the same as the surface controls and values for wake refinement. See Surface Controls And Values.

See Also:

Custom Mesh Controls
Creating and Defining Custom Controls
Surface Mesher Controls and Values

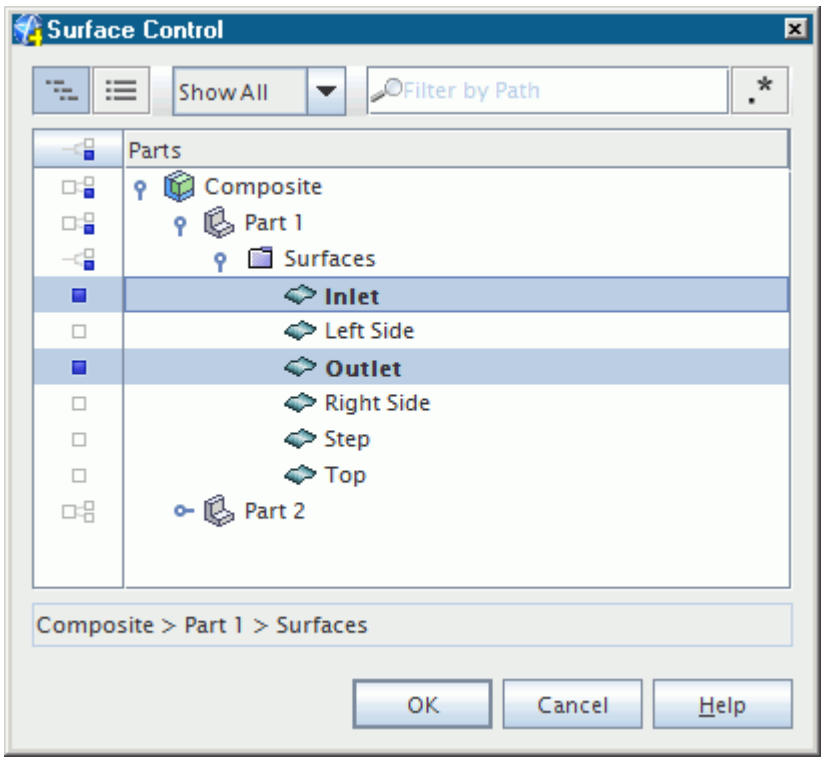
Surface Control

Surface controls specify alternative surface mesh and prism layer settings for part surfaces, geometry parts, and composite parts.

If a part surface belongs to several controls, the following hierarchy determines the surface control that takes precedence. This hierarchy depends on how you apply the part surface to the control.

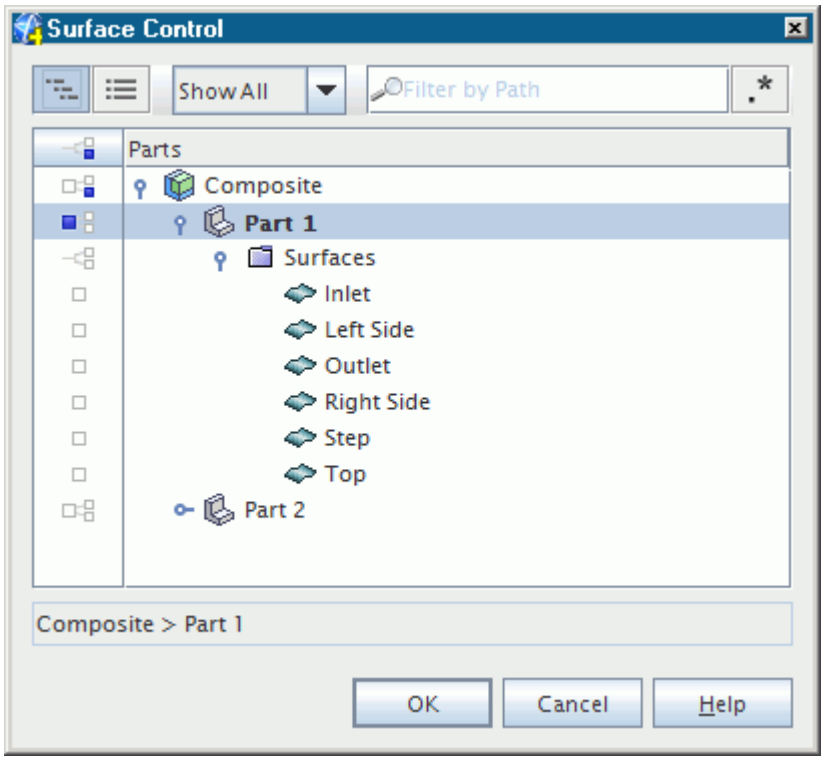
- Selecting the individual part surfaces.

  Only the individual part surface that is directly selected belongs to the surface control



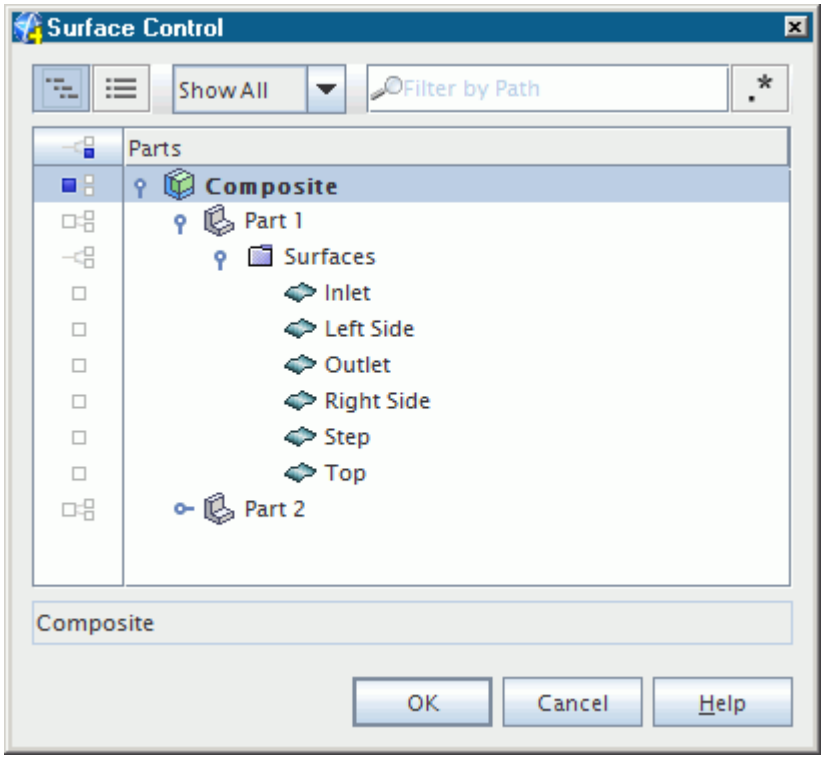
- Selecting a geometry part that includes the part surfaces.

  All part surfaces that comprise the selected geometry part are included in the surface control, even if they don't appear selected individually.



- Selecting a composite part that includes the part surfaces.

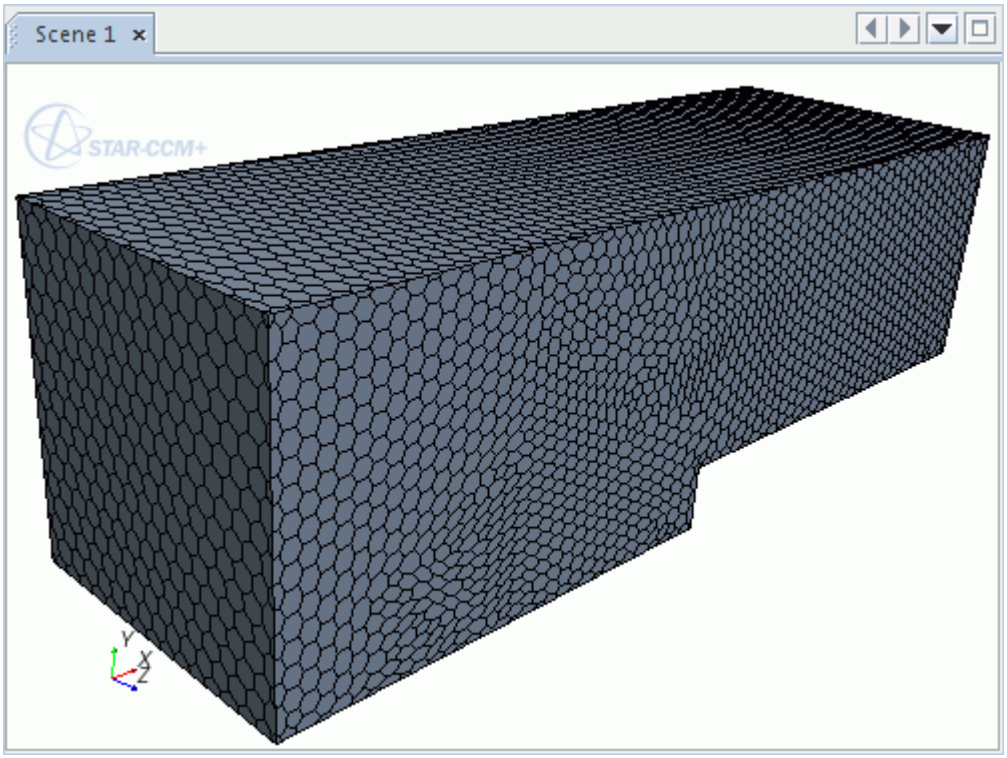
  All part surfaces on all geometry parts that comprise the composite part are included in the surface control, even if they don't appear selected individually.



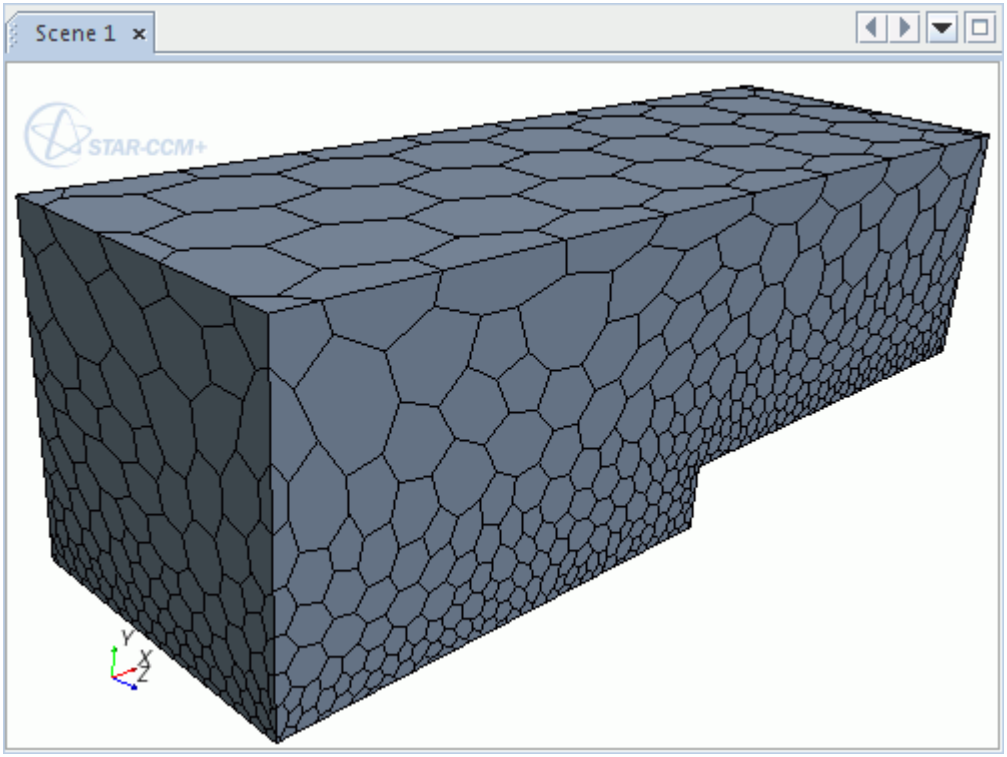
In addition, if a part surface belongs to several controls at the same level in this hierarchy, the control that produces the smallest cell size takes precedence.

Example of a Surface Control

This example coarsens the mesh in the far field, which provides a more efficient solution by only solving detailed flow in areas of interest. The following image shows the original volume mesh.



Coarsening the surface mesh by five times on the far field surfaces gives a lower mesh density. This coarse surface mesh produces a coarse volume mesh.



Contents:

Surface Growth Rate
Surface Controls and Values

See Also:

Creating and Defining Custom Controls
Custom Mesh Control Context Menu
Custom Mesh Control Properties

Surface Growth Rate

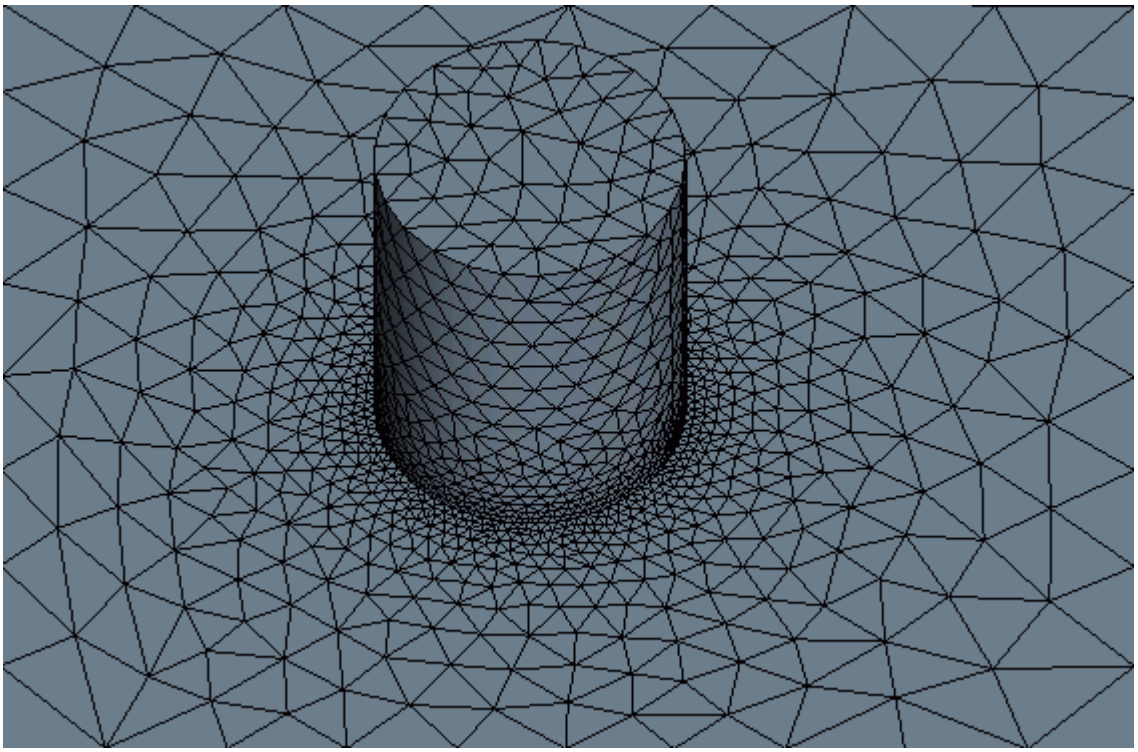
Surface Growth Rate can be specified on a part or part surface to control the triangle growth rate on those surfaces.

This control allows you to match the surface growth rate on symmetry, inlet, outlet, and far field boundaries with the volume growth rate.

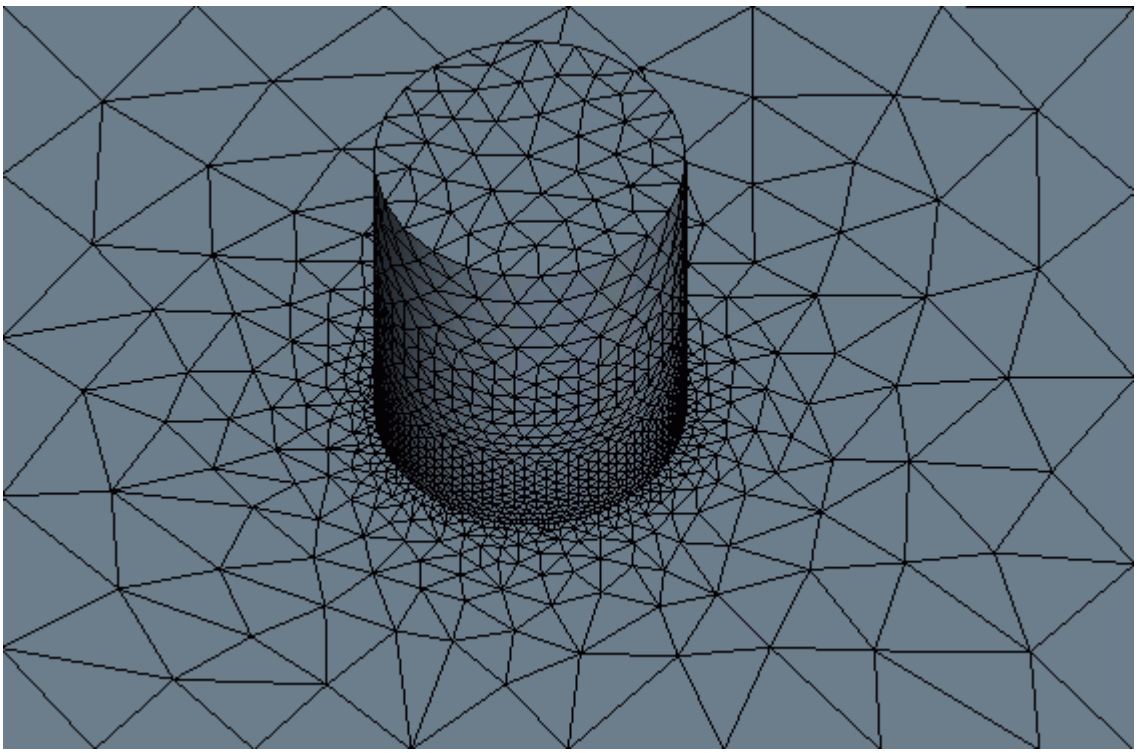
This function is only available in parts-based meshing.

The examples below show a cylinder intersected by a symmetry plane. The *Surface Growth Rate* is based on a curve control from the feature curve where the cylinder meets the symmetry plane.

The following example shows a constant *Surface Growth Rate* of 1.3 on all surfaces:



The example below shows the surface mesh with a *Surface Growth Rate* of 1.1 on the cylinder surface and 2.0 on the plane surface.



Surface Controls and Values

Use the surface controls and values to define the mesh controls that you want to modify–including wake refinement.

You can modify several mesh controls using a surface control.

Controls – Non-Wake Refinement Properties

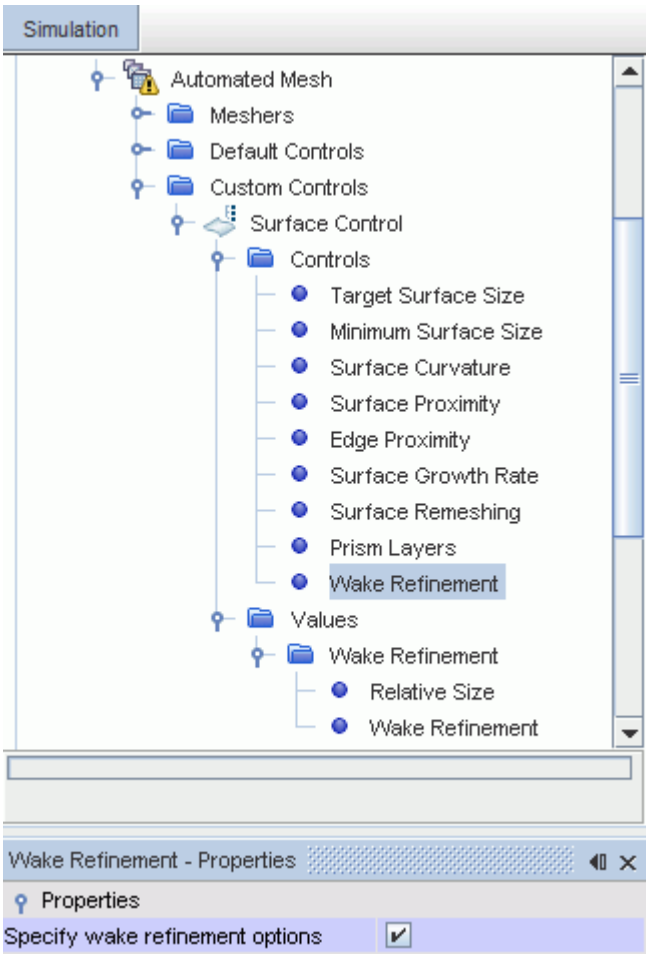
For the mesh controls that are listed, you can specify the properties as relative or absolute.

- Minimum Surface Size
- Prism Layers
- Surface Curvature
- Surface Proximity
- Surface Growth Rate
- Surface Remeshing
- Target Surface Size
- Trimmer Surface Growth Rate

Control	**Specify Custom Value**	Allows you to specify a value for this control that overrides the default control.

	Use Parent Value	Uses the same value as the default control.
	Disable	Does not apply the selected mesh control to the surfaces. This option is only available for the Surface Curvature, Surface Proximity, Surface Remeshing, and Prism Layers controls.
	Enable	Applies the selected mesh control to the surfaces. This option is only available for the Surface Remeshing control.

Controls – Wake Refinement Properties

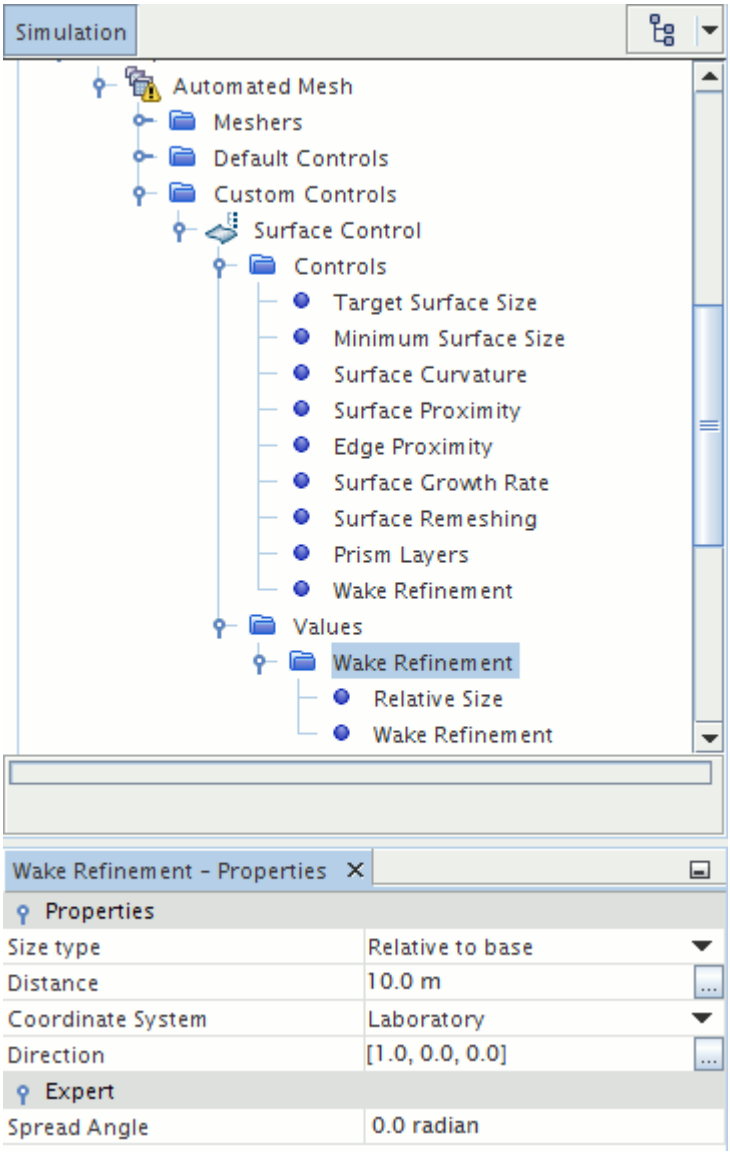


The following wake refinement mesh controls allow you to activate wake refinement and specify wake refinement options:

- Remesher Wake Refinement – appears when you use the Surface Remesher along with the Trimmed Mesher
- Trimmer Wake Refinement – appears when you use the Trimmed Mesher
- Wake Refinement – appears when you use either the Surface Remesher, Polyhedral Mesher, or Tetrahedral Mesher individually, or when you use the Surface Remesher along with either the Polyhedral Mesher or Tetrahedral Mesher

Control	**Activated**	A **Values** > **Wake Refinement** node is created which has properties and sub-nodes that allow you to set values for the wake refinement.
	Deactivated	Wake refinement does not occur.

Values – Wake Refinement Properties

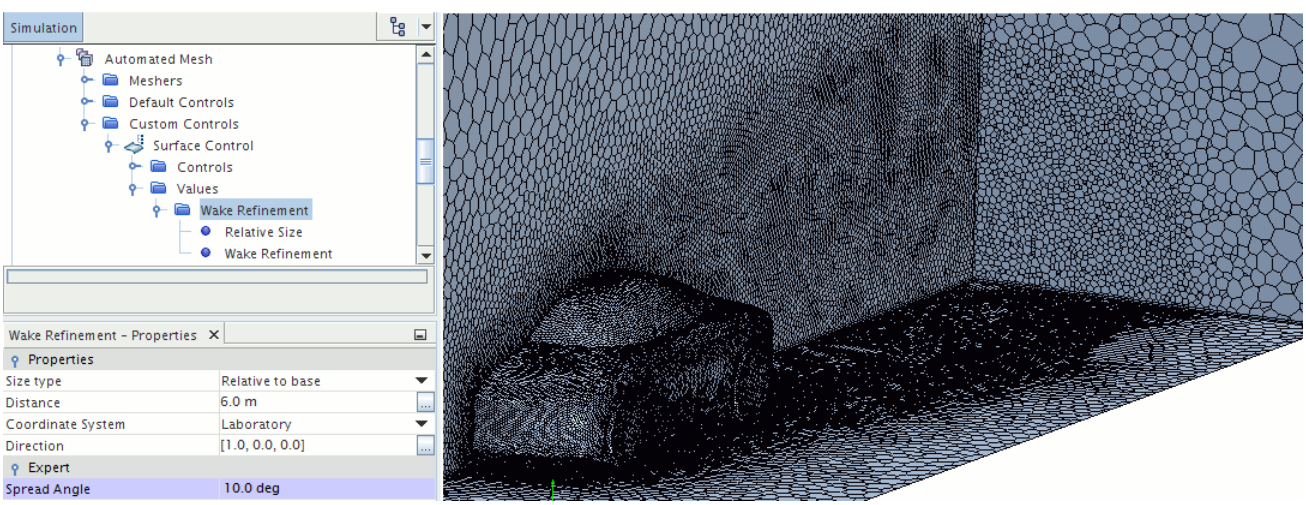


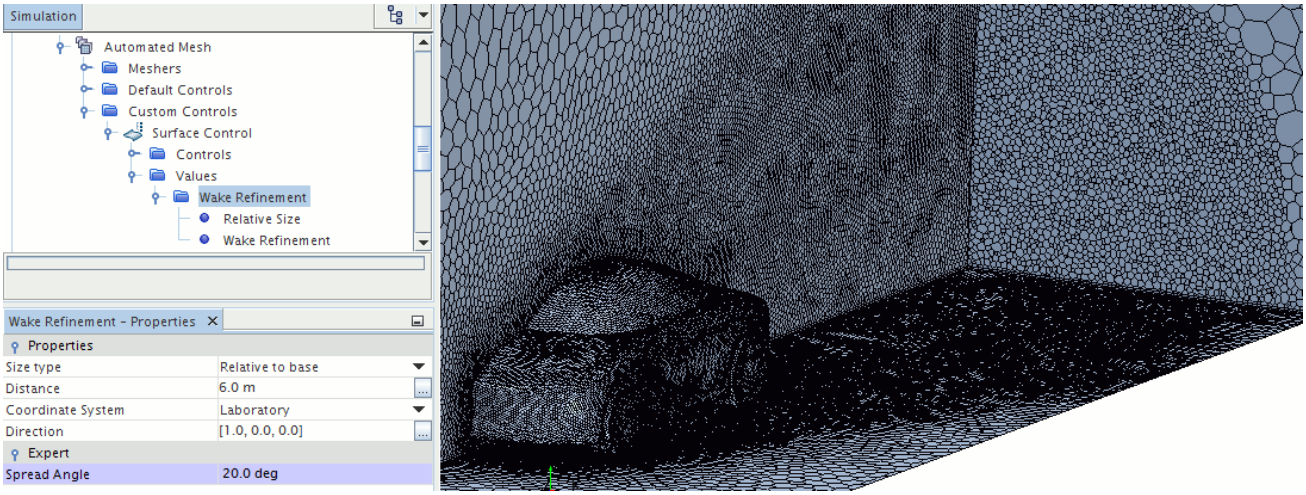
Size Type	**Relative to base**	Creates a **Relative Size** sub-node which allows you to define the upper limit of cell sizes in the wake area as a percentage relative to the base size.
	Absolute	Creates an **Absolute Size** sub-node which allows you to define the upper limit of cell sizes in the wake area as an absolute size.
Distance	The distance that the wake extends from the selected surface or curve.	
Coordinate System	The coordinate system for the wake direction. Once a Cartesian coordinate system has been defined, it can be selected using the pull-down menu.	
Direction	The direction that the wake extends from the selected surface or curve with respect to the selected coordinate system.	

Values – Wake Refinement Expert Properties

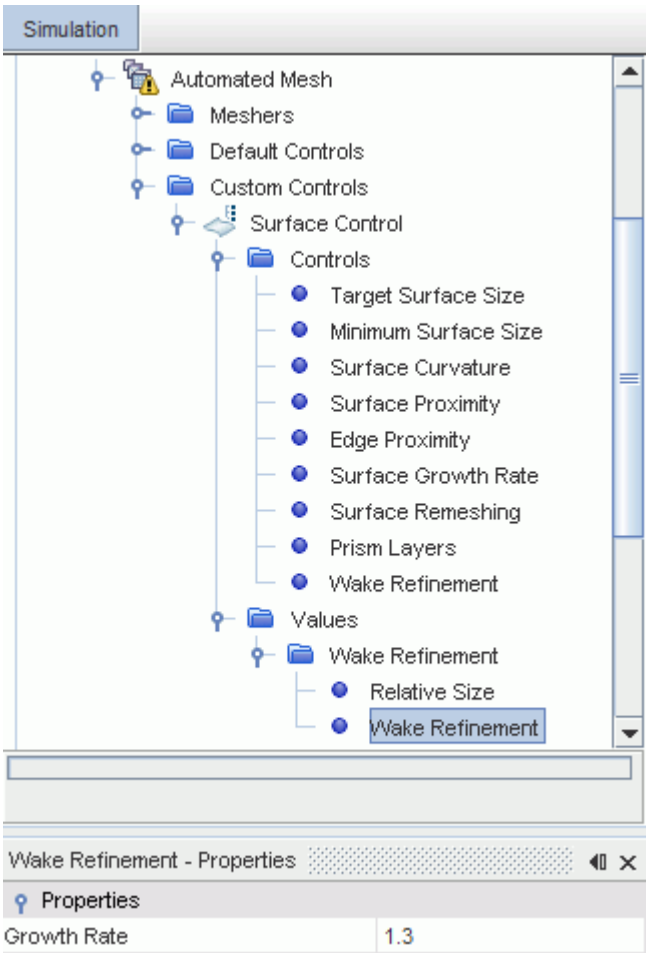
Spread Angle	The angle to which the wake refinement spreads from the specified wake refinement direction.

The following examples show wake refinement that is specified for a surface control on the exterior of a car. The spread angles that are specified below are 10 degrees and 20 degrees from the -Z direction:





Values – Wake Refinement Sub-Nodes Properties



The sub-nodes that appear under the **Values** > **Wake Refinement** node depend on the volume meshing model that the wake refinement is specified in.

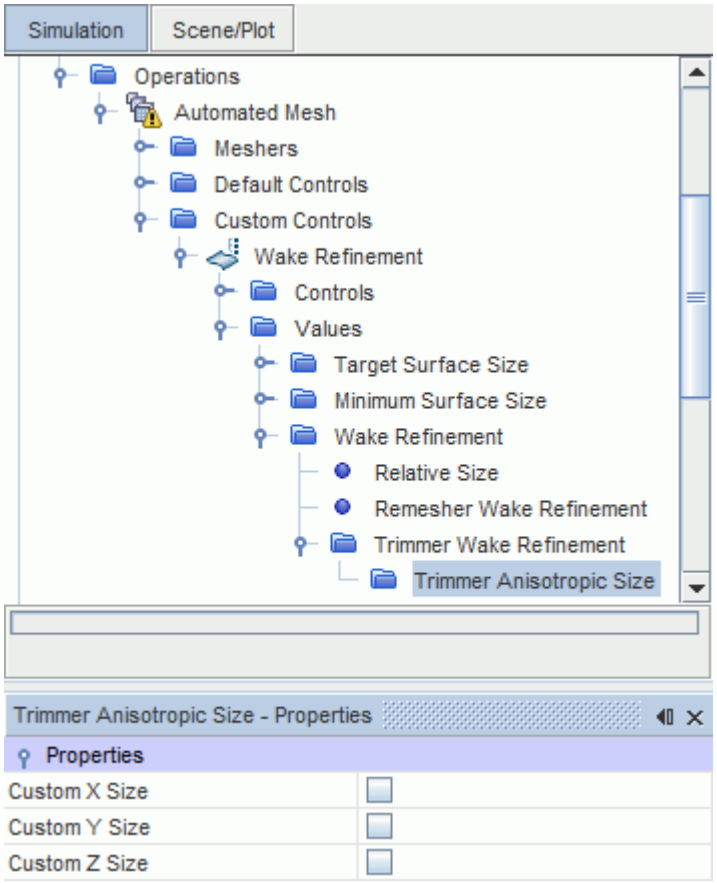
The **Values** > **Wake Refinement** > **Wake Refinement** node appears when either:

- You specify wake refinement in either the Surface Remesher, Polyhedral Mesher, or Tetrahedral Mesher individually
- You specify wake refinement in the Surface Remesher along with either the Polyhedral Mesher or Tetrahedral Mesher

However, if you specify wake refinement in the Trimmed Mesher, the **Values** > **Wake Refinement** > **Trimmer Wake Refinement** node appears. If you specify wake refinement in both the Surface Remesher and the Trimmed Mesher, the **Values** > **Wake Refinement** > **Remesher Wake Refinement** node also appears.

The Properties of these **Values** > **Wake Refinement** sub-nodes are as follows:

Growth Rate	Available for the **Wake Refinement** and **Remesher Wake Refinement** sub-nodes. Allows you to control the speed of cell growth through the wake. The cell size grows from the part surface or part curve into the volume of the wake. The cells within the wake stop growing when the cell size reaches the specified maximum cell size within the wake.	
Default Growth Rate	Available for the **Trimmer Wake Refinement** sub-node. Controls the stepping from one cell size to the next within the wake.	
	Fast	This setting usually equates to having only one layer of each cell size for each transition, which results in the least number of cells being created for the supplied input values.
	Medium	Gives a minimum of two equal sized cell layers per transition.
	Slow	Gives a minimum of four equal sized cell layers per transition.
	Very Slow	Gives a minimum of eight equal sized cell layers per transition.
	Custom	Adds a **Custom Surface Growth Rate** sub-node which allows you to specify the number of cell layers per transition.
Surface Growth Rate	Available for the **Trimmer Wake Refinement** sub-node. Determines the number of cell layers to grow in the transition from cells next to surfaces to cells in the core volume.	
	Disable	Default setting. The default growth rate is also applied to cell layers next to surfaces.
	Fast	Sets one cell layer next to surfaces.
	Medium	Sets two cell layers next to surfaces.
	Slow	Sets four cell layers next to surfaces.
	Very Slow	Sets eight cell layers next to surfaces.

	Custom	Adds a **Custom Surface Growth Rate** sub-node which allows you to specify the number of cell layers per transition.
Customize Anisotropic Size	Available for the **Trimmer Wake Refinement** sub-node. When activated, the Relative or Absolute size that is set for the parent wake refinement node is not acknowledged. Activating this property creates a set of sub-nodes which allow you to set a custom cell size for each coordinate direction relative to the trimmer coordinate system.  	

See Also:

Custom Mesh Controls

Creating and Defining Custom Controls

Surface Mesher Controls and Values

Part Control

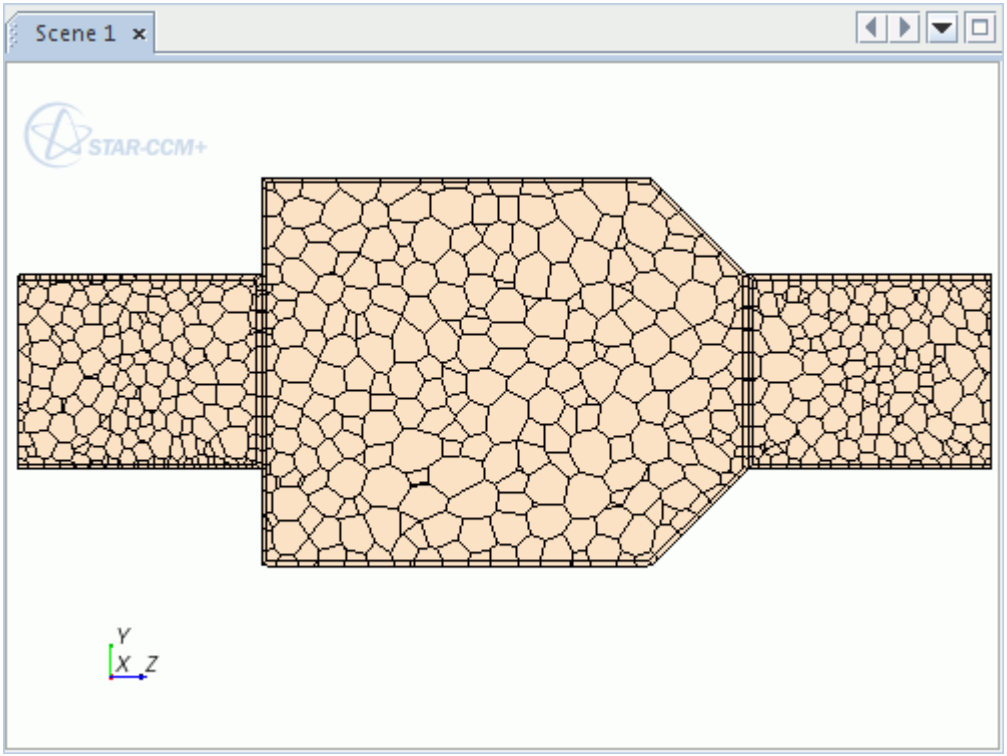
Part controls specify alternative volume mesh settings for the polyhedral and tetrahedral meshers.

If a part belongs to several part controls, the control that produces the smallest cell size takes precedence.

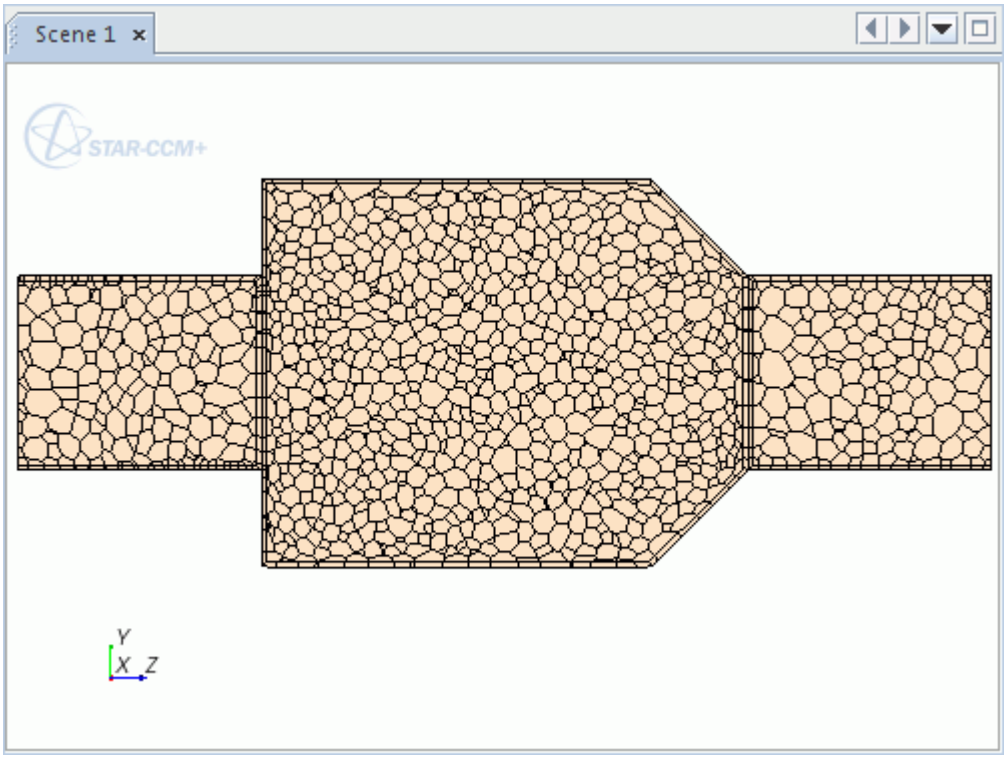
Selecting a composite part selects all geometry parts that comprise the composite part, even if they do not appear selected.

Example of a Part Control

This example refines the core volume mesh in the center section of the following geometry.



Increasing the volume mesh density and growth rate increases the number of cells within the core mesh. In this case, the control does not affect the mesh density and growth in the adjacent parts or prism layers.



Contents:

Part Controls and Values

See Also:

Creating and Defining Custom Controls
Custom Mesh Control Context Menu
Custom Mesh Control Properties

Part Controls and Values

Use the part controls and values to modify the mesh density for the polyhedral and tetrahedral meshers.

Part Control Properties

Mesh Density	**Specify Custom Value**	Allows you to specify a value for this control that overrides the default control.
	Use Parent Value	Uses the same value as the default control.

See Also:

Custom Mesh Controls

Creating and Defining Custom Controls

Volumetric Control

Volumetric controls specify alternative mesh sizes and prism layer settings for the surface and volume mesher.

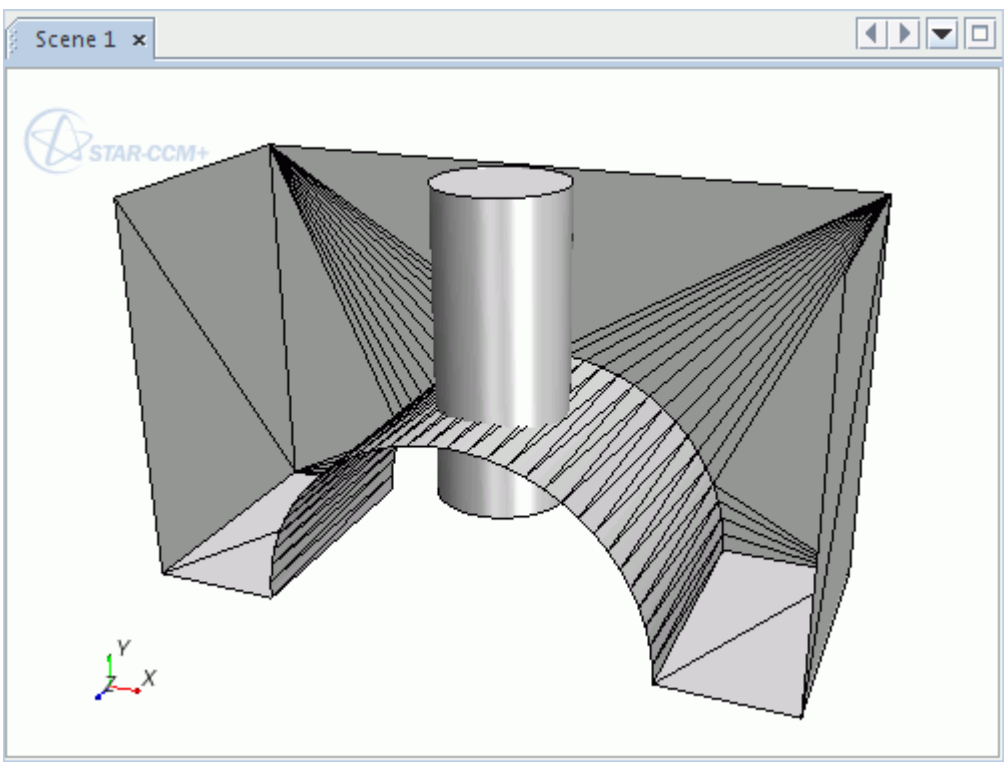
Generally, volumetric controls have common properties that specify an alternate face or cell size (see Common Volumetric Control). You can specify the face or cell size as relative or absolute.

However, for the prism layer mesher and trimmed mesher, the behavior of volumetric controls is different (see Prism Layer Mesher Volumetric Control and Trimmed Mesher Volumetric Control).

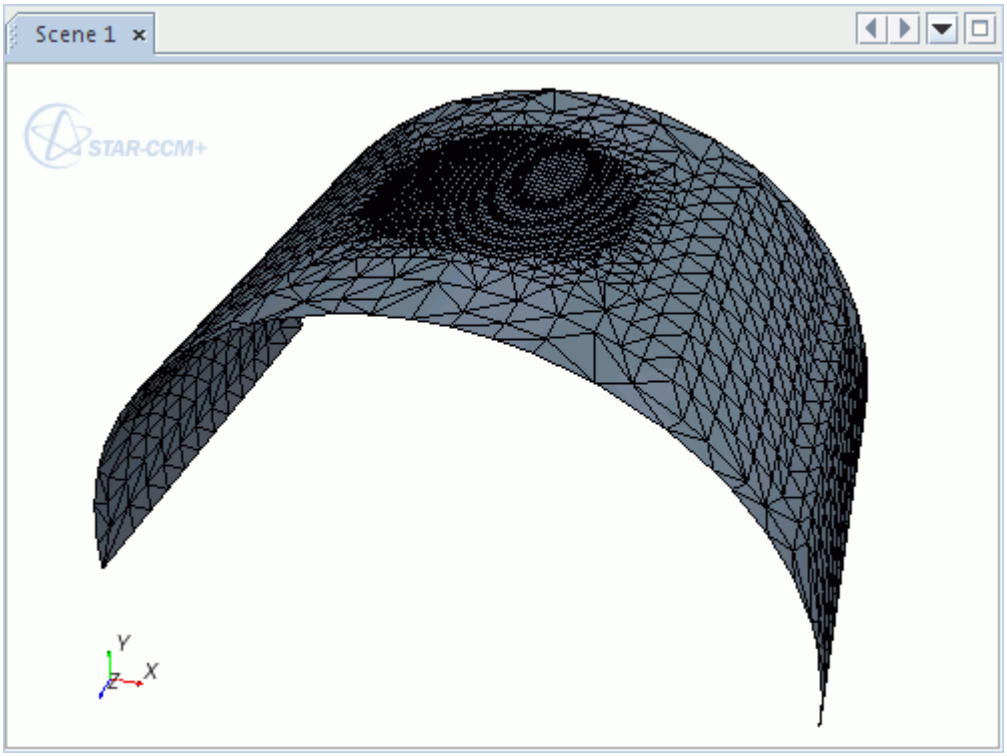
If a volume belongs to several volumetric controls, the control that produces the smallest cell size takes precedence.

Examples of Volumetric Controls

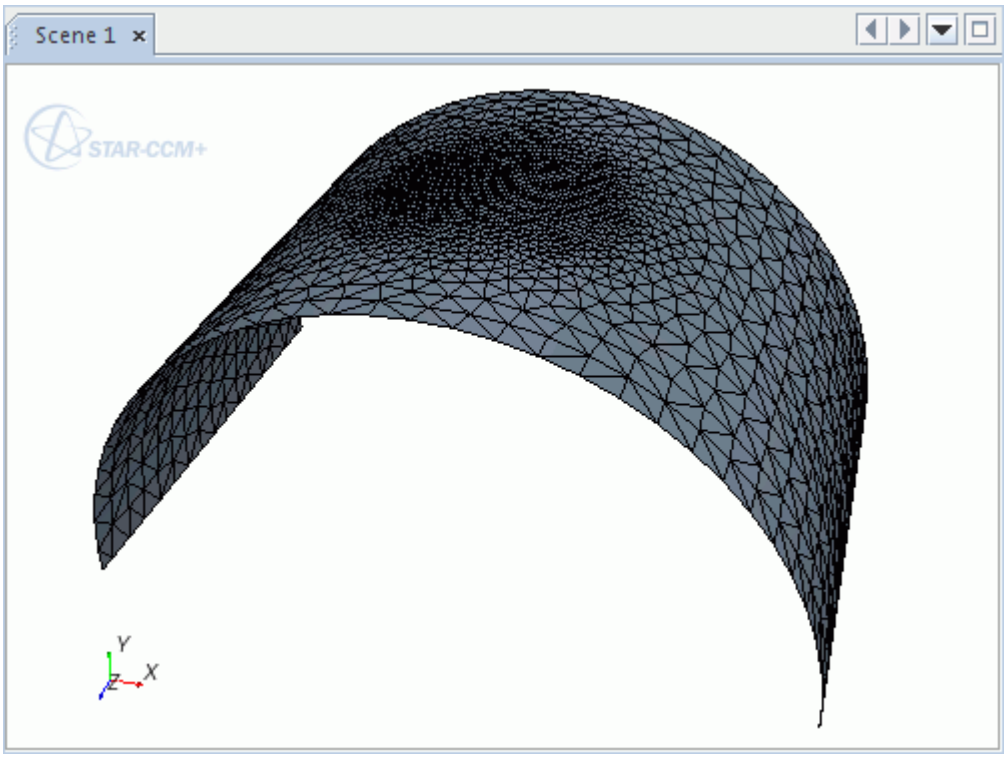
The following examples show 30% mesh refinement with a cylindrical volumetric control. The image below shows the initial geometry and the cylindrical volumetric control.



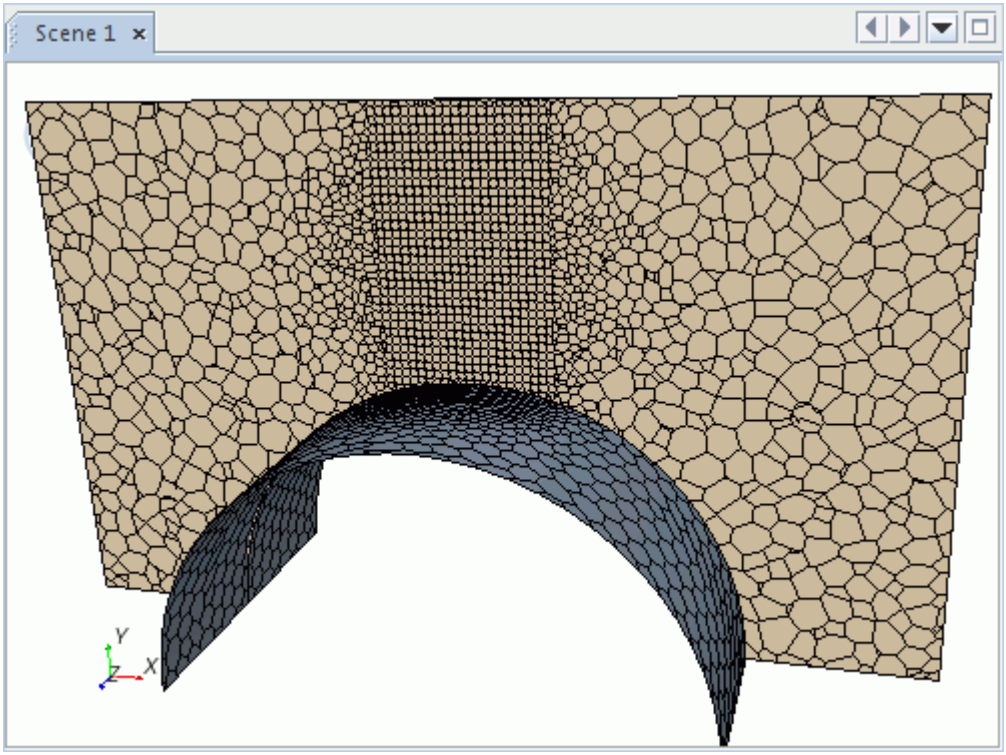
Surface Wrapper



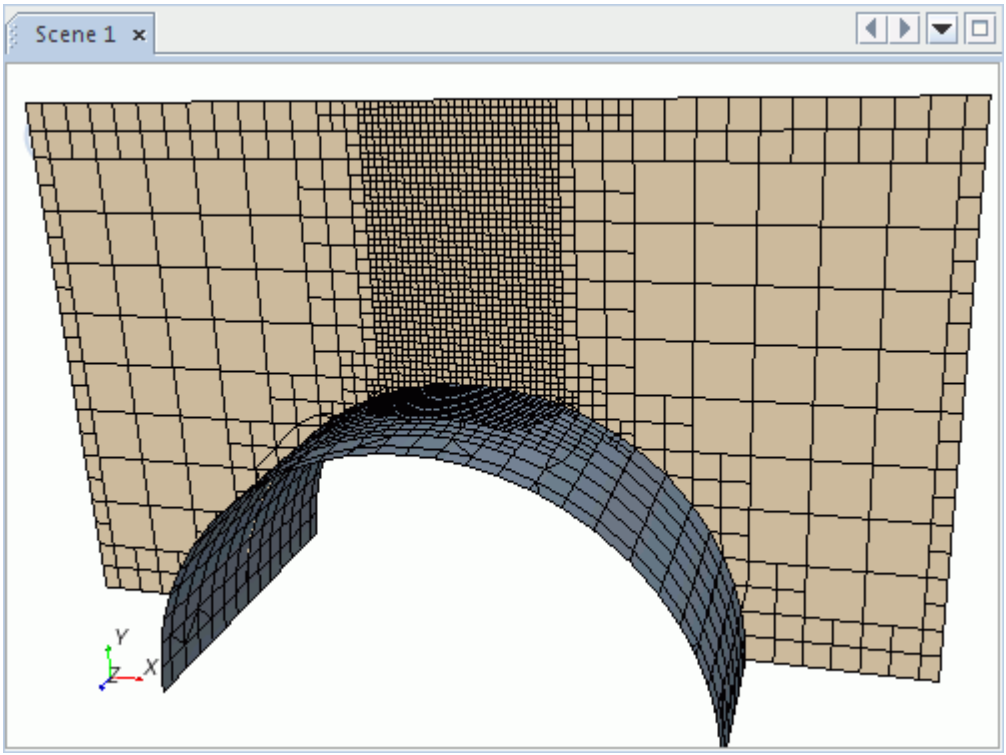
Surface Remesher



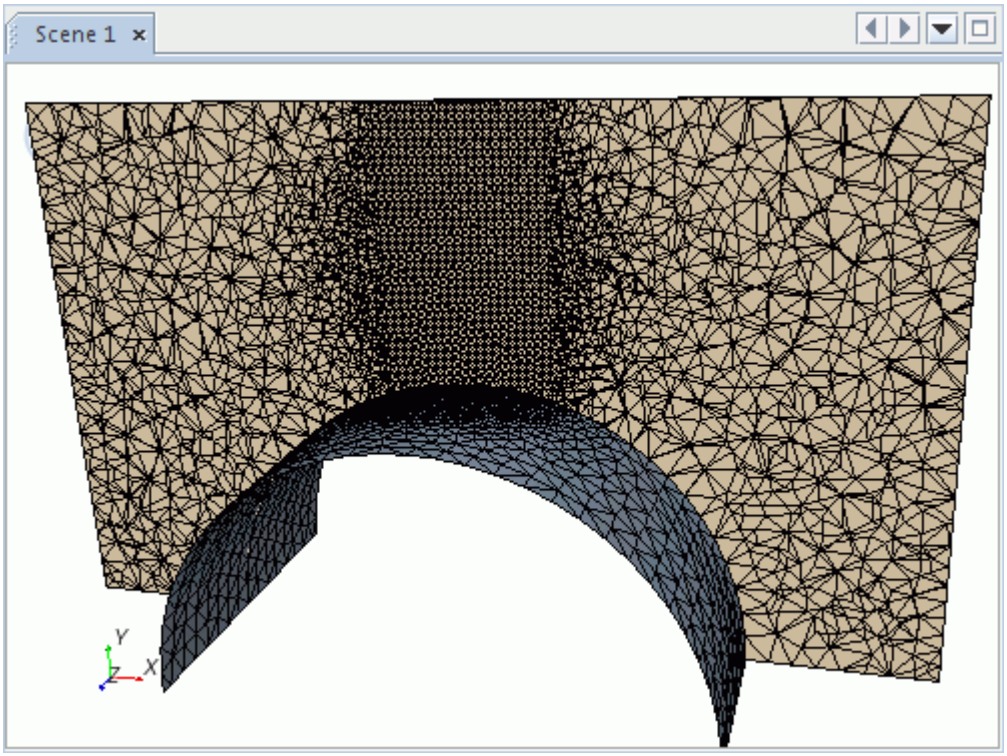
Polyhedral Mesher



Trimmed Mesher

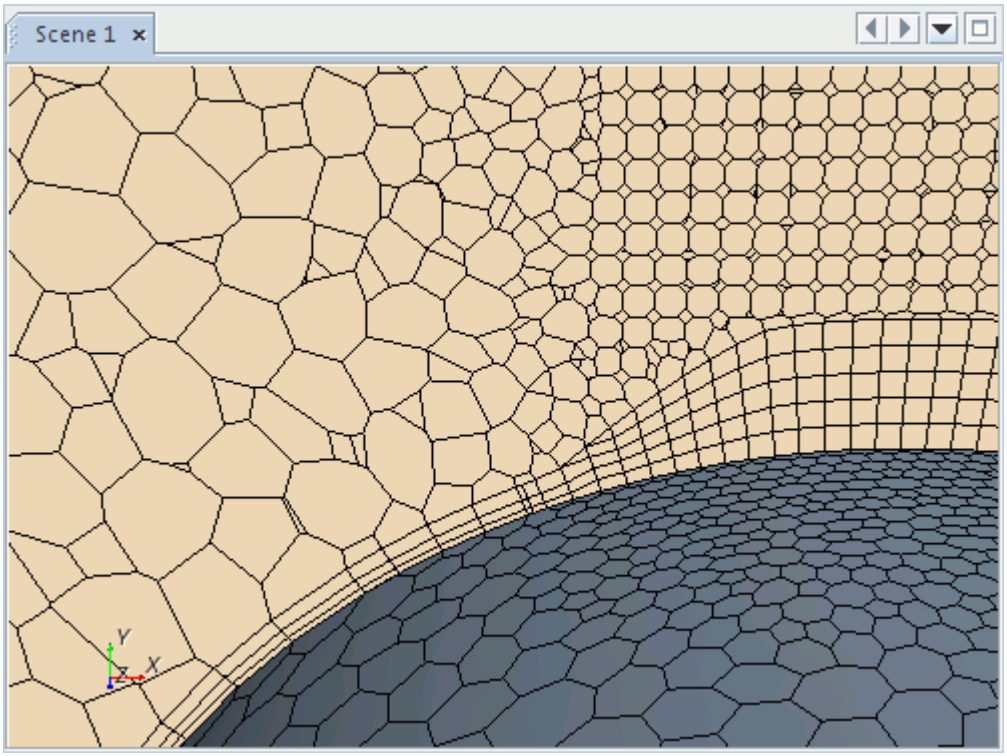


Tetrahedral Mesher



Prism Layer Mesher

This example is slightly different from the others in that the number of layers, and prism layer thickness, changes within the volumetric control. To make sure that the prism layers are influenced by the volumetric control, use a shape that is larger than the target area of interest.



Contents:

Volumetric Controls and Values

Volumetric Controls and Values

Use the volumetric controls and values to define surface and volume mesh sizes within a specified volume.

Common Volumetric Control

These controls are common to most volumetric controls and specify alternative face or cell sizes for the surface or volume mesh respectively.

Common Volumetric Control Properties

Customize	**Activated**	Allows you to apply an alternative cell or face size.
	Deactivated	Uses the cell or face size that the default controls define.

Prism Layer Mesher Volumetric Control

Use these volumetric controls to specify an alternative number of prism layers, prism layer thickness, and prism layer stretching.

Prism Layer Mesher Volumetric Control Properties

Customize Number of Layers	**Activated**	Allows you to specify an alternative number of prismatic cell layers.
	Deactivated	Uses the number of prismatic cell layers that the default controls define.
Customize Total Thickness	**Activated**	Allows you to specify an alternative total thickness for all cell layers.
	Deactivated	Uses the total thickness that the default controls define.
Customize Stretching	**Activated**	Allows you to specify an alternative stretching value between prismatic cell layers.
	Deactivated	Uses the stretching value that the default controls define.

Trimmed Mesher Volumetric Control

Use this volumetric control to specify an alternative isotropic or anisotropic cell size for the trimmed mesher.

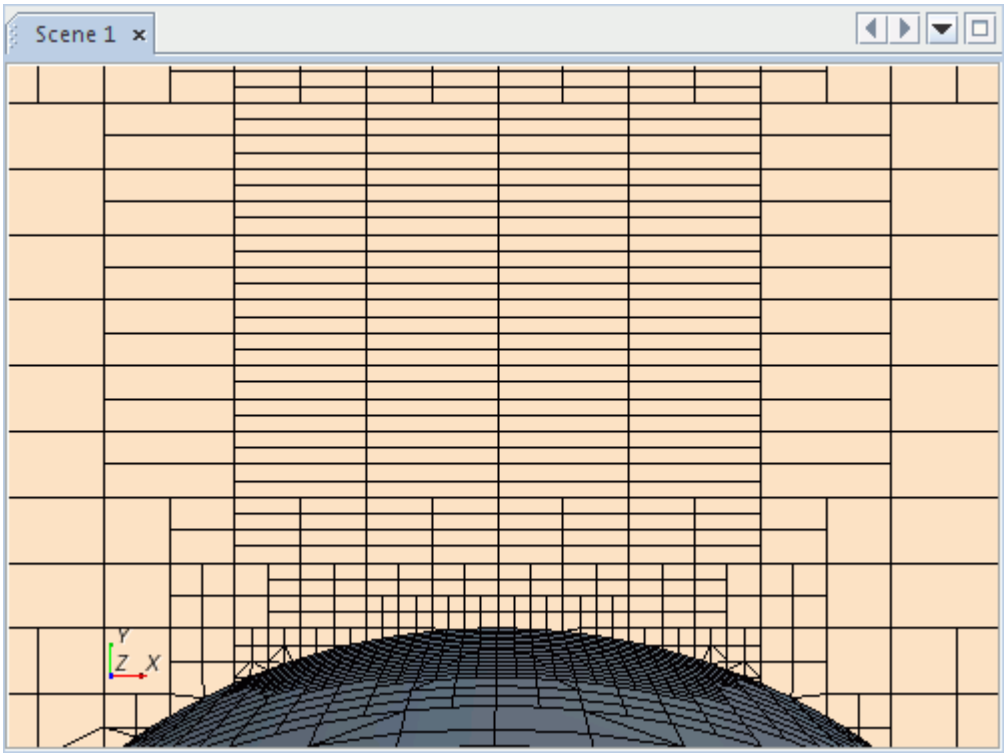
Trimmed Mesher Volumetric Control Properties

Customize isotropic size	**Activated**	Allows you to apply an alternative cell size that is the same in each coordinate direction.
	Deactivated	Uses the cell size that the default controls define.
Customize anisotropic size	**Activated**	Allows you to apply a different cell size in each coordinate direction.
	Deactivated	Uses the cell size that the default controls define.

Trimmed Mesher Anisotropic Size Value

Use this value to specify different cell sizes in each coordinate direction for the trimmed mesher. You can specify these cell sizes as relative or absolute.

The following image shows an example of anisotropic refinement in the y-direction.



Trimmed Mesher Anisotropic Size Value Properties

Custom X size	**Activated**	Specifies an alternative cell size in the x-direction relative to the trimmed mesher coordinate system.
	Deactivated	Uses the cell size that the custom size value specifies.
Custom Y size	**Activated**	Specifies an alternative cell size in the y-direction relative to the trimmed mesher coordinate system.
	Deactivated	Uses the cell size that the custom size value specifies.
Custom Z size	**Activated**	Specifies an alternative cell size in the z-direction relative to the trimmed mesher coordinate system.
	Deactivated	Uses the cell size that the custom size value specifies.

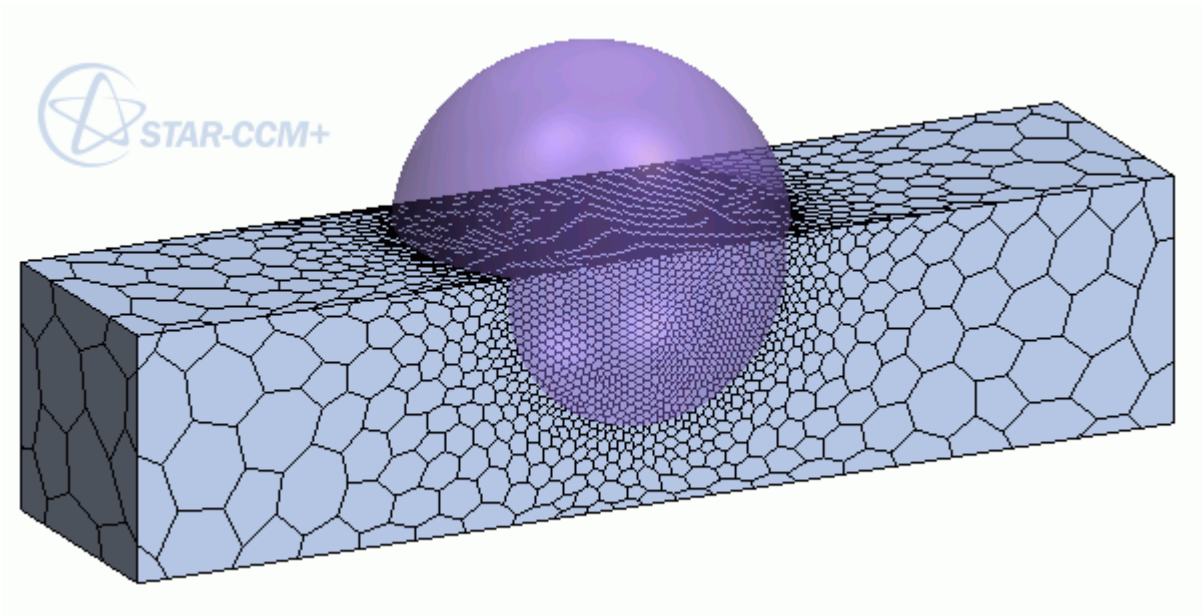
See Also:

Custom Mesh Controls

Creating and Defining Custom Controls
Volume Mesher Controls and Values

Mesh Refinement in a Specified Volume

STAR-CCM+ allows you to refine surface and volume meshes within a specific volume in order to capture localized effects. You define the specific volume using a geometry part, which you can create or import. The volume can be created to intersect specific volumetric parameters within; geometry parts, fluid domains or both.



To refine the mesh:

1. Prepare the Geometry Part that defines the volume.

   See Preparing the Geometry Part that Defines the Volume.
2. Create a Volumetric Control.

   See Creating a Volumetric Control.
3. Set Volumetric Control Values.

   See Editing Volumetric Control Values.

After running surface or volume meshers, the mesh within the volumetric parameters specified now appears refined.

Contents:

Preparing the Geometry Part that Defines the Volume
Creating a Volumetric Control
Editing Volumetric Control Values

Preparing the Geometry Part that Defines the Volume

The volume within which the mesh is refined is defined by a geometry part.

It is not necessary for this geometry part to belong to the geometry you are modeling. You can import or create a separate part.

To obtain a part that defines the volume, choose one of the following options:

- Create a simple shape part in STAR-CCM+

  This option is the simplest way to create a block, cone, cylinder or sphere for the volumetric control shape part.

  See Creating Simple Shape Parts.
- Import a CAD part into STAR-CCM+

  This option allows you to use your own CAD part for the volumetric control shape part.

  See Importing Geometry.
- Create a shape part in 3D-CAD and import it to **Geometry** > **Parts**

  This option is useful if you want to create a more complex shape part for the volumetric control, but do not already have a CAD part to import.

  See Using 3D-CAD and Importing Geometry.

Creating a Volumetric Control

Before Creating a Volumetric Control:

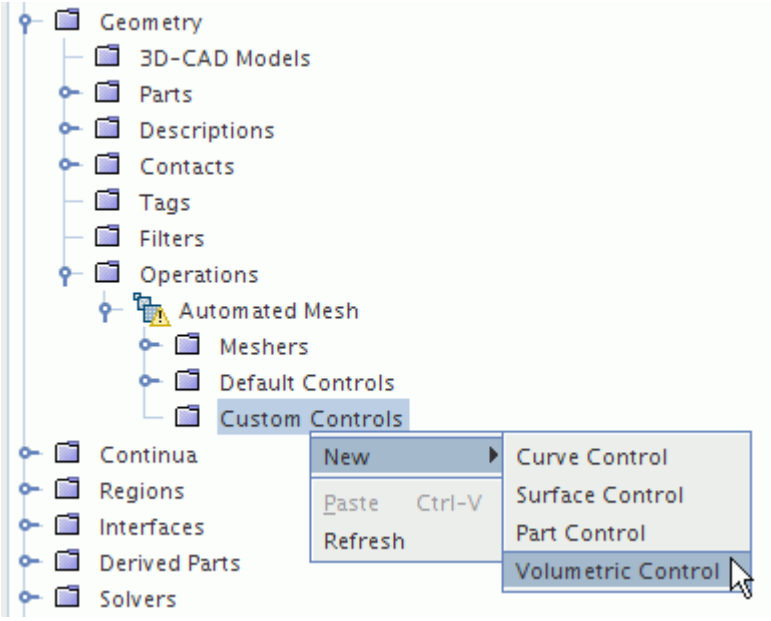
1. Prepare the Geometry Part that Defines the Volume.

   See Preparing the Geometry Part that Defines the Volume.
2. Create an Automated Mesh Operation.

   See Selecting Meshers.

To Create a Volumetric Control:

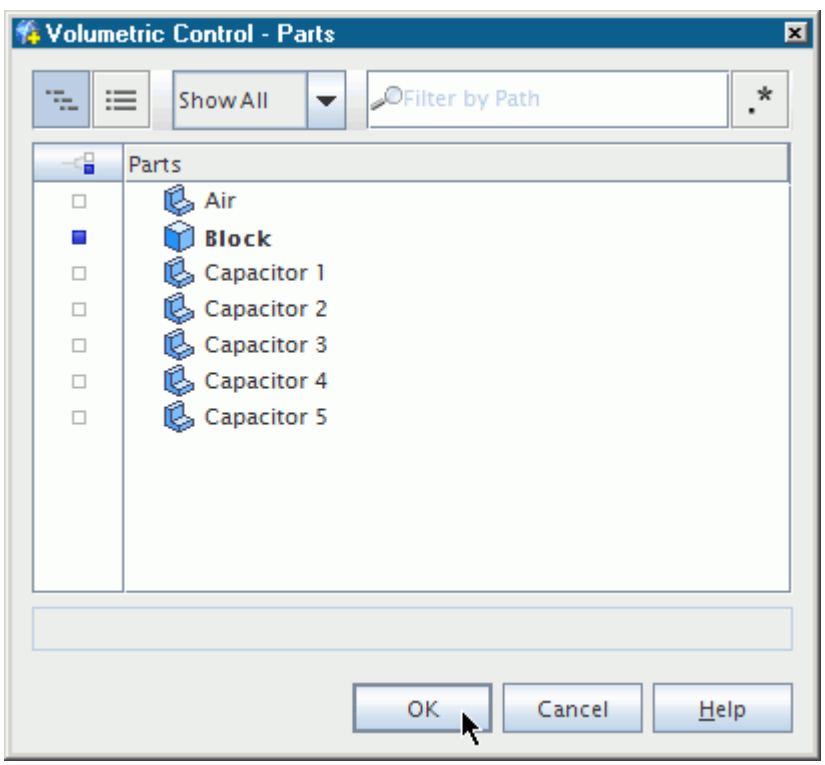
1. Right-click **Geometry** > **Operations** > **Automated Mesh** > **Custom Controls** and select **New** > **Volumetric Control**.



2. Select the new **Volumetric Control**node and click the area to the right of **Parts** in the *Volumetric Control - Properties* pane.

   The *Volumetric Control* dialog appears.

3. Select the shape part that defines the volume.



4. Click **OK**

You now have a Volumetric Control which is ready for you to customize. Until you customize the Volumetric Control, the settings will remain the same as the Automated Mesh which contains the Volumetric Control.

Proceed to Editing Volumetric Control Values.

Editing Volumetric Control Values

Volumetric Controls provide additional options when you choose which meshing models to customize.

You can add custom values for the following:

- Surface Remesher
- Polyhedral Mesher
- Tetrahedral Mesher
- Trimmed Cell Mesher
- Prism Layer Mesher (option only available after selecting a volume mesher)

See Volumetric Controls and Values.

Note: The Thin Mesher, Embedded Thin Mesher, and Advancing Layer Mesher are not yet available in Automated Mesh operations. To use these Meshers within a volumetric control use Region Based Meshing. See Setting Up a Volumetric Control.

Before Editing Volumetric Control Values:

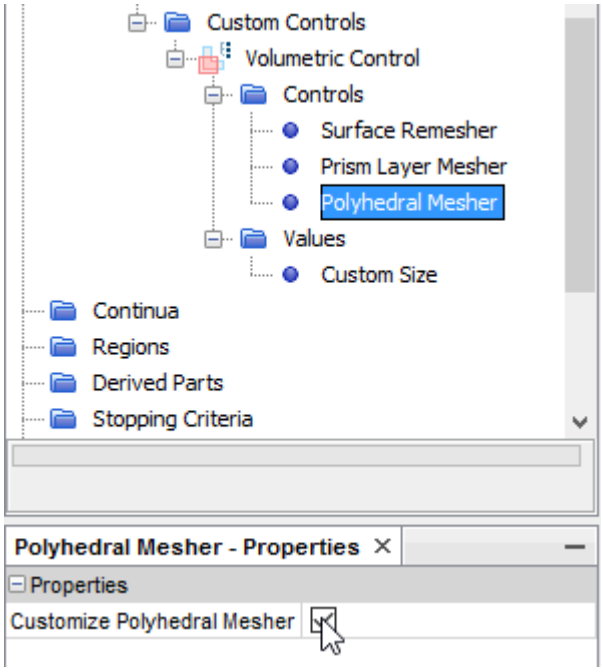
Create a Volumetric Control for the Part that Defines the Volume.

See Creating a Volumetric Control

To Edit Volumetric Control Values:

1. Expand the **Volumetric Control** > **Controls** node and select the relevant Mesher you wish to customize for the selected volume(s).
2. Activate the properties you wish to set in the *[Mesher] - Properties*

   For example, selecting the **Polyhedral Mesher** node displays the following in the *Polyhedral Mesher - Properties* window:



3. Activate the **Customize Polyhedral Mesher** node.

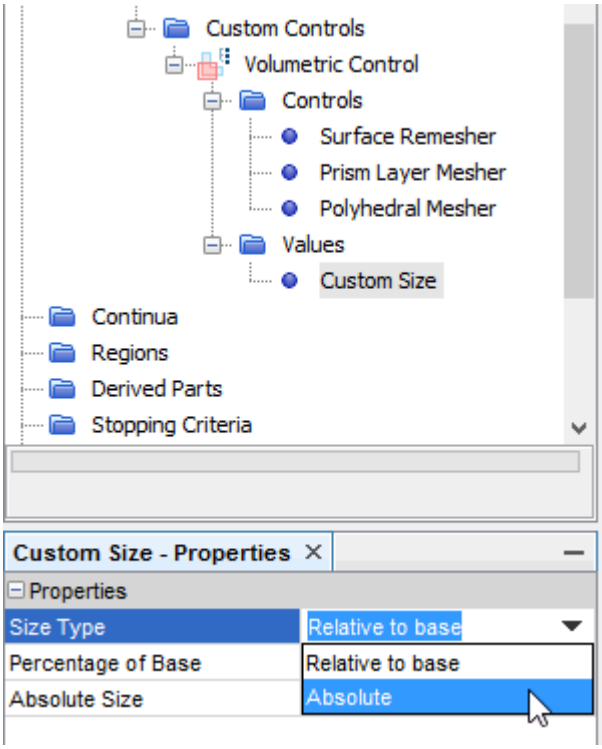
   A new**Custom Size** node appears under the **Values** node.

Editing Custom Sizes

Some Volumetric controls have the option to set values as a percentage that is relative to the base size, or absolute values.

For example:

To apply a specific size in the **Polyhedral Mesher**, select the **Values** > **Custom Size** node. For *Size type* choose either **Relative to base** or **Absolute** as shown:

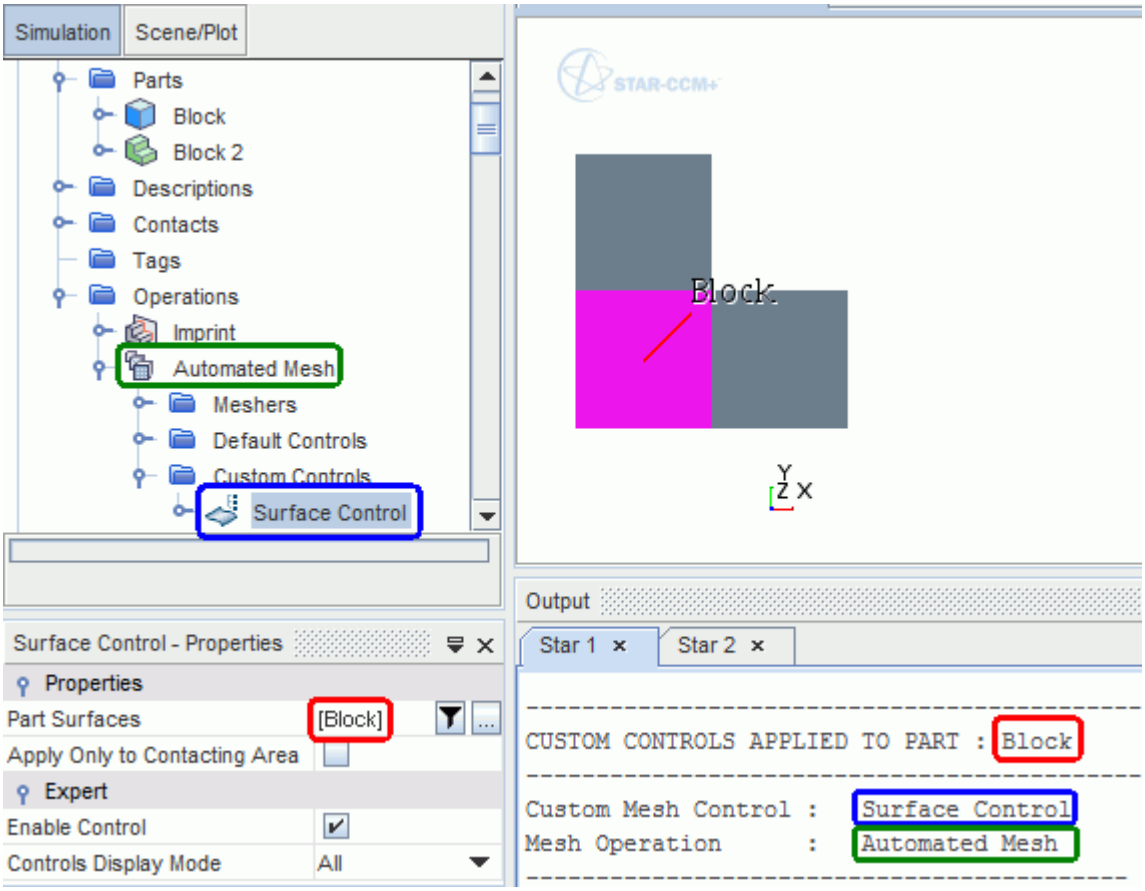


Query Custom Mesh Controls

This feature allows you to display – as a list in the Output window – all of the meshing custom control settings that are applied to a selected part, composite part, part surface, or part curve.

When there is a high level of mesh refinement in a simulation due to custom mesh controls, this feature is useful to find out which custom mesh controls are applied to a specific part, composite part, part surface, or part curve.

For example, upon right-clicking a **Geometry** > **[Part]** and selecting **Query Controls**, the *Output* window updates to display details of all custom mesh controls that are applied to the **[Part]**. The name of the Mesh Operation to which each custom control belongs is also displayed.



The Query Controls option displays all types of custom controls:

- Curve Controls
- Surface Controls
- Part Controls
- Volumetric Controls

Mesh Refinement in a Wake

Wake refinement using the Parts-Based meshing approach allows you to define the direction, distance, and spread angle of the wake relative to a part surface or part curve from which the wake grows.

The wake grows from the part surface or part curve into the fluid volume of the wake which is then refined to the parameters that you specify. Setting a large growth rate means that the cell size increases rapidly from the part surface or part curve into the wake. The cells within the wake stop growing when the cell size reaches the specified maximum cell size within the wake. Using a large part surface to grow a wake from can cause excessive cell counts in the fluid volume. It is computationally less expensive to use part curves which define a specific feature instead of a large part surface. When you intend to view a scalar output in 3D in the wake, use a part surface. However, if you only intend to view a scalar output on a plane propagating downstream, you can use part curves at the trailing edge for wake refinement.

You can perform wake refinement using an Automated Mesh Operation with any of the following meshers:

- Surface Remesher

- Polyhedral Mesher
- Tetrahedral Mesher
- Trimmed Mesher

Note: You can also use the Prism Layer Mesher with a core mesher, although, it is not possible to specify wake refinement settings for the Prism Layer Mesher.

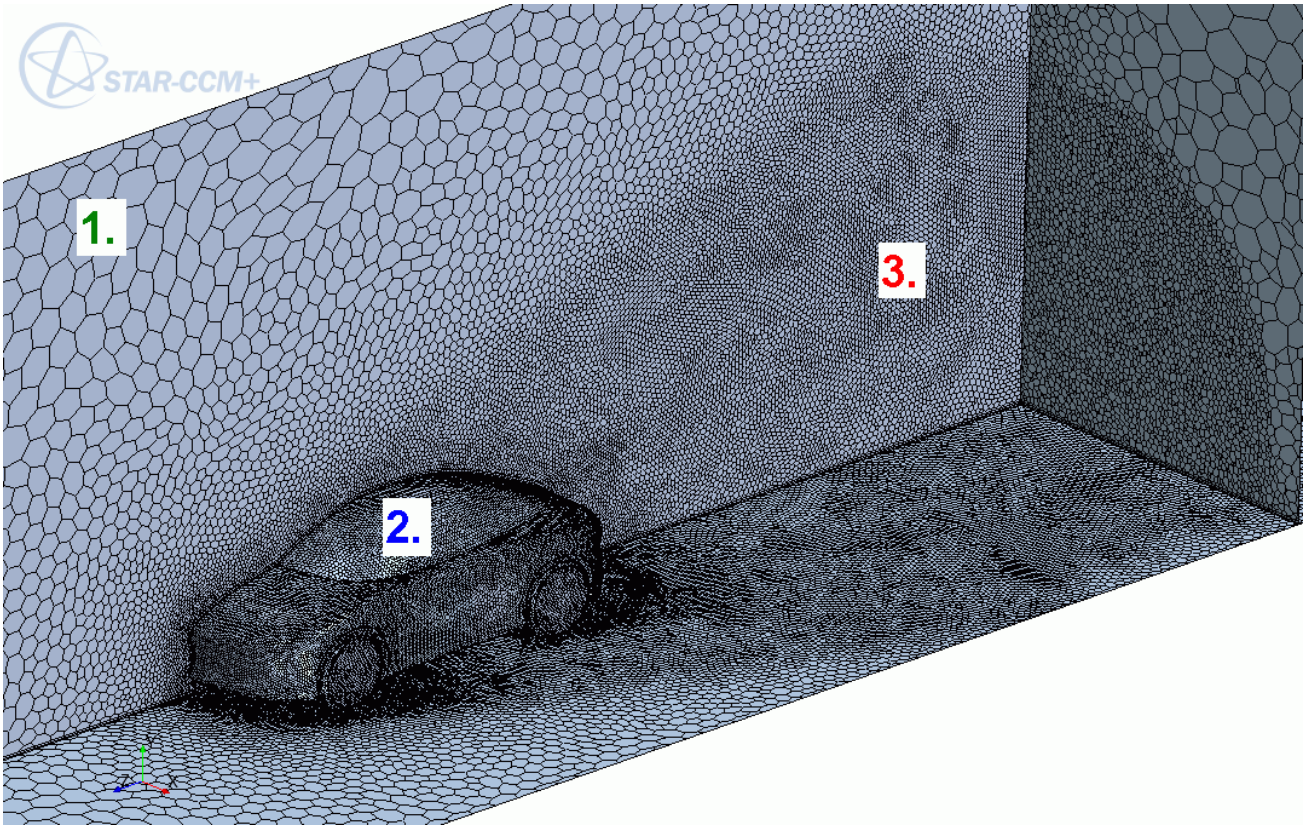
Before refining the mesh in a wake, make sure that the starting geometry represents the fluid that surrounds the object which causes the wake.

1. Right-click the **Geometry** > **Operations** node and select **New** > **Automated Mesh Operation**.
2. In the *Create Automated Mesh Operation* dialog:
   a) Set Parts to the part which represents the fluid volume.
   b) Select the meshing models that you require for meshing the fluid volume.
   c) Click **OK**.
3. Expand the **Automated Mesh** > **Default Controls** node and set the parameters that you require for the overall fluid domain.
4. Right-click the **Automated Mesh** > **Custom Controls** node and select **New** > **Surface Control** or **New** > **Curve Control**.

   To avoid confusion, you can rename the **Surface Control** or **Curve Control** node to represent the Wake Refinement.
5. Select the **Surface Control** or **Curve Control** node and in the *Properties* window:
   a) Ensure that **Enable Control** is activated.
   b) Specify the *Part Surfaces* or *Part Curves* from which the wake refinement should grow.
6. Define the *Part Surfaces* or *Part Curves* custom controls. Either:
   - Expand the **Custom Controls** > **Surface Control** > **Controls** node, then specify the Surface Controls for the Part Surface that is selected and the wake. See Surface Controls And Values.
   - Expand the **Custom Controls** > **Curve Control** > **Controls** node, then specify the Curve Controls for the Part Curve that is selected and the wake. See Curve Controls And Values.

   Note: The quality of the surface mesh affects the quality of the wake directly.

When refining the mesh settings, make sure that you set the controls and values correctly in the area that you intend. The following example shows the wake from a car and the location in the simulation tree where you can refine each area. A surface control is set for the car surface and the wake occurs within the polyhedral volume mesh in a specified direction up to a specified angle and distance from the car surface.



7. Execute the Automated Mesh Operation.
8. If necessary, modify the mesh settings and re-execute the Automated Mesh Operation.

   Note: If the cell size within the wake refinement grows too quickly to the specified maximum cell size, you can increase the maximum cell size that is specified for the wake, or you can decrease the growth rate.

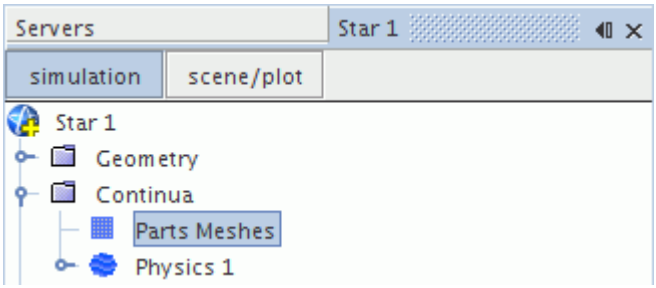
9. To turn off wake refinement temporarily without losing any settings that are already defined, you can either:
   - Deactivate all Surface Control or Curve Control settings and the wake refinement by selecting the **Custom Controls** > **Surface Control** or **Custom Controls** > **Curve Control** node and deactivating *Enable Control*.
   - If you want to maintain all Surface Control or Curve Control settings except the wake refinement, select the **Custom Controls** > **Surface Control** > **Wake Refinement** node or the **Custom Controls** > **Curve Control** > **Wake Refinement** node and deactivate *Specify wake refinement options*.

Parts Meshes Continua

The parts meshes continua represents the volume mesh that you generated for the part as a continuum that the physics region can use.

Conceptually, the parts meshes continua is a bridge between the parts-based mesh and the physics regions.

The **Parts Meshes** node represents this part-based volume mesh.



Contents:

Parts Meshes Continua Properties

Parts Meshes Continua Properties

Use the **Part Meshes** properties to view information on the parts meshes continua and define the method for mapping solution data between different volume meshes.

Properties

Interfaces	The interfaces (or mesh contacts) between parts that are defined (read-only).
Regions	The physics region to which the parts meshes continua applies (read-only).

Expert Properties

Interpolation Option	**Nearest neighbor**	Maps the solution data from one mesh to another by comparing the cell centroids in each mesh. That is, the new cell gets the solution data from an old cell whose cell centroid is closest to its own. This option is quick, but does not provide a smooth interpolation of results.
	Higher-order stencil	Maps the solution data from one mesh to another by using a stencil of cells and faces on the original mesh. A smooth interpolation is achieved by using a distance-weighted, least-square interpolation. This option is

		recommended for transient analyses.

See Also:

Extract Volume

Region-Based Meshing

When using region-based meshing, the meshes that you generate are associated with the regions and continua that you have defined, and mesh controls relate to regions and boundaries.

This approach differs to parts-based meshing, where the meshes are associated with the geometry parts. Review the comparison of meshing strategies before deciding on the strategy that is appropriate for you.

Contents:

Creating a Region-Based Mesh
Feature Curves
Using Feature Curves to Repair a Surface
Volumetric Controls
Mesh Continua

Creating a Region-Based Mesh

Before creating a region-based mesh, complete the following prerequisite stages.

- Import or create a geometry as the basis for the computational domain.
- Repair the surface so that it forms a watertight outer boundary to the computational domain, or be prepared to use the surface wrapper when meshing.
- Define the simulation topology by assigning parts to regions and defining the appropriate regions and boundaries.

Contents:

Creating Mesh Continua
Selecting Meshing Models
Setting Mesh Model Properties
Setting Mesh Controls and Values
Initializing the Mesh
Generating the Surface Mesh
Generating the Volume Mesh
Clearing the Mesh
Replacing a Mesh Region

Creating Mesh Continua

This section outlines how to create mesh continua.

A mesh continuum is required on which to define the appropriate meshing methods.

See Mesh Continua.

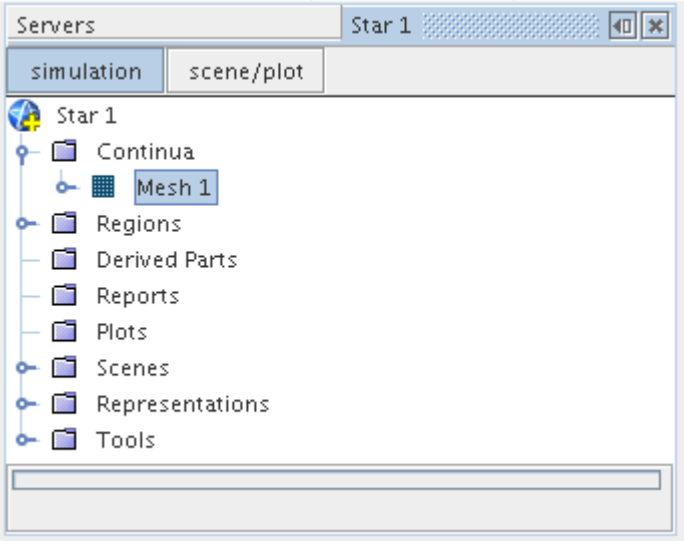
Contents:

Automatically Creating a Mesh Continuum During Import
Manually Creating a Mesh Continuum
Assigning Mesh Continua to Regions
Using Per-Region Meshing

Automatically Creating a Mesh Continuum During Import

A mesh continuum is automatically created when a surface mesh is imported.

When a new simulation is created, it does not contain any continua and the **Continuum** node is empty. As soon as a surface mesh is imported (using the **Create New Region** option for *Import Mode*), a mesh continuum is automatically created:

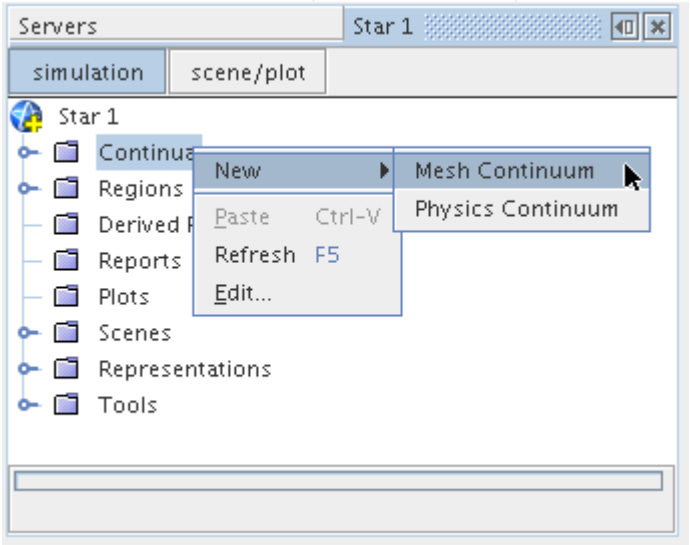


By default, this continuum is called **Mesh 1**, but can be renamed if necessary. All mesh regions that are created during the import process are assigned to this mesh continuum. Importing further surface files for a given simulation does not generate any extra mesh continua but you can manually create a new mesh continuum at any point in the process.

Manually Creating a Mesh Continuum

This section describes how to create a new mesh continuum manually.

A new mesh continuum can be created by right-clicking on the **Continua** node and selecting **New** > **Mesh Continuum**.



The new continuum is given the next logical sequential name such as **Mesh 2** but can be renamed if necessary. Manager nodes for **Models**, **Reference Values**, and **Volumetric Controls** are added as child nodes to the new continuum node. Once the new mesh continuum has been created, you can proceed to select the meshing models for it. If the mesh continuum is the first that is created, then any null regions that currently exist are automatically assigned to this continuum.

You would typically require a new mesh continuum in any of the following circumstances:

- You want to have a different volume mesh type for each region.
- You are using the trimmed cell mesher for a multi-region case.
- You want to use the surface wrapper for a multi-region case.

In each of the above circumstances, a conformal mesh interface between each region would not be possible during the meshing process.

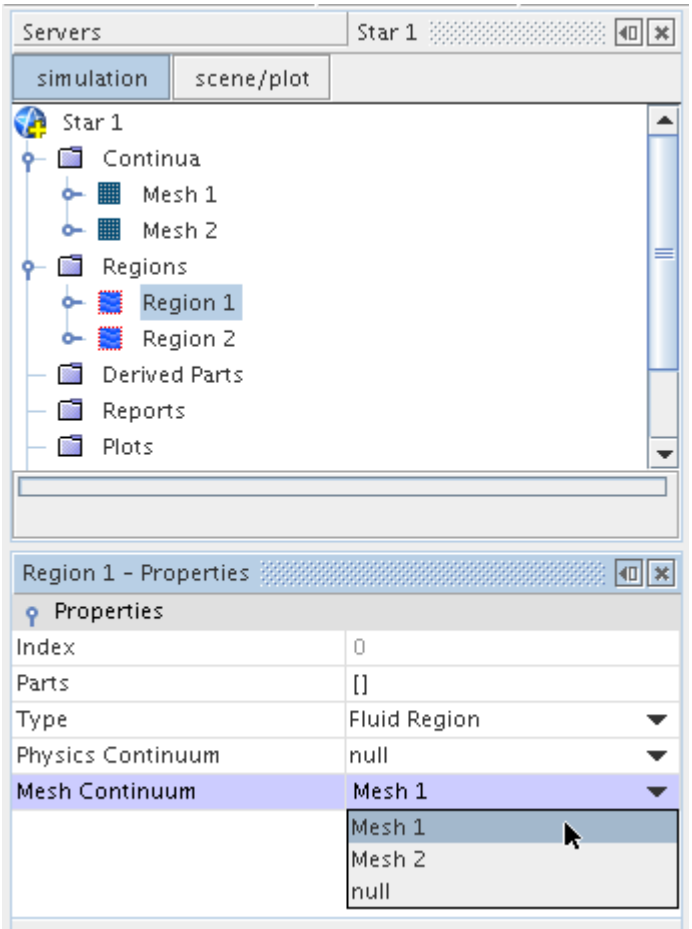
Assigning Mesh Continua to Regions

Regions are automatically assigned to the default mesh continuum on surface mesh import. They can be reassigned in one of the following ways.

- Setting the continuum for the region using the region properties panel
- Dragging and dropping the continuum onto the region

Setting the Region Mesh Continuum

When the node for a region is selected, the **Properties** window shows the currently selected *Mesh Continuum*.



When more than one mesh continuum exists, the appropriate selection can be made from the menu for the highlighted region. Setting the option to None indicates that no mesh operations are performed for that region.

Dragging a Mesh Continuum Onto a Region

A mesh continuum can be assigned to a region by using the drag-and-drop feature in STAR-CCM+ to drag the mesh continuum onto the region.

Using Per-Region Meshing

This section describes how to activate per-region meshing.

By default, each mesh continuum applies its meshing models to all regions associated with the continuum, in a single operation. In general, this means that the meshing models account for any interfaces that exist between regions and create conformal mesh at the interfaces.

Meshing models that support conformal meshes across interfaces are:

- surface remesher
- tetrahedral mesher
- polyhedral mesher
- prism layer mesher

Meshing models that do not support conformal meshes are:

- surface wrapper
- trimmed cell mesher
- thin mesher

Specific situations where per-region meshing is essential are:

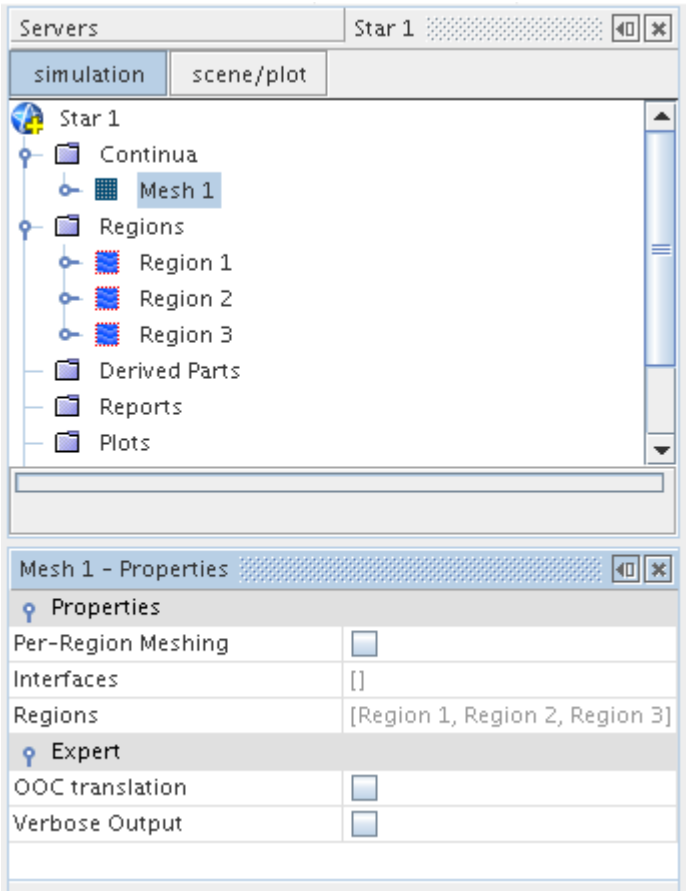
- When using a mesher that does not support conformal interfaces on a multi-region case.

  Meshers in this category are:

  - trimmed-cell mesher
  - thin mesher
  - surface wrapper
- When using overset meshes, and you want to mesh regions that occupy the same physical space using a single mesh continuum.
- For conjugate heat transfer cases where you want to have different cell densities on either side of the solid/fluid interface. The interface itself can be defined as a mapped interface, in order to improve the interpolation of solved quantities across the interface.

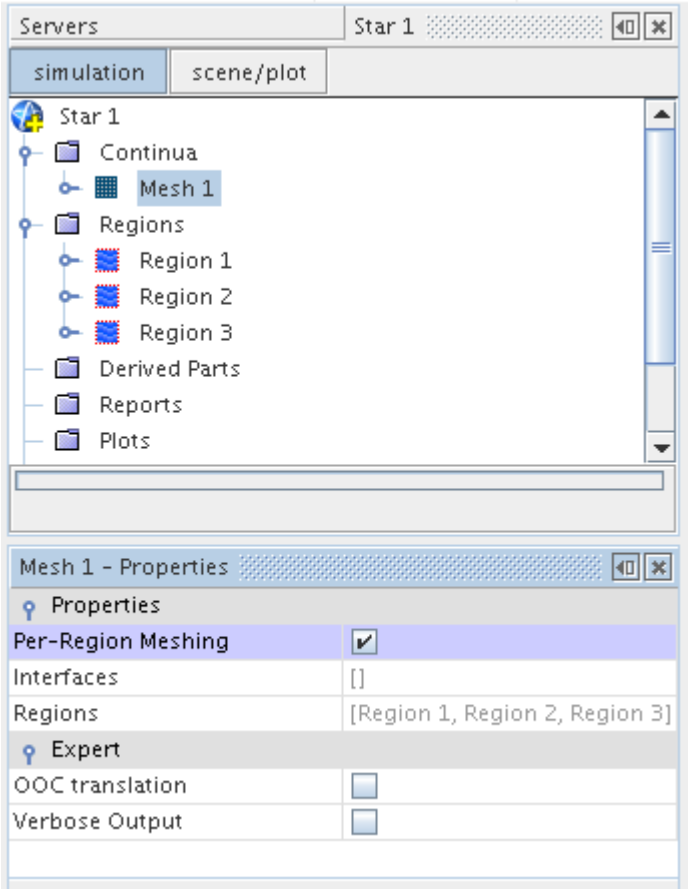
To activate per-region meshing:

1. Select the relevant mesh continuum node:



By default, the *Per-Region Meshing* option is turned off, meaning that any models for the continuum behave as described above.

2. In the *Properties* window, activate the **Per-Region Meshing** option:



The Mesh Conditions manager node is deactivated on interfaces when per-region meshing is activated.

The extruder mesher is not influenced in any way by the per-region meshing option.

Selecting Meshing Models

A meshing model in STAR-CCM+ determines the type of mesh that is used within a given mesh continuum. Models can be selected individually or together.

There are both surface meshing models and volume meshing models. Models can be selected individually or together depending on the desired action. Meshing model selections are executed using the (**Generate Surface Mesh**) and (**Generate Volume Mesh**) buttons in the **Mesh Generation** toolbar. Pressing the **Generate Volume Mesh** button also completes any selected surface meshing operations that have not been performed so far, or that are out-of-date due to parameters that have been modified.

A continuum must contain at least one mesh model in order to be valid. If more than one meshing model is selected, then each operation is carried out in turn during the meshing process. The pipeline sequence in which meshing operations are carried out for a continuum is as follows:

1. Surface wrapper
2. Surface remesher
3. Volume mesher, which can be the:
   - Polyhedral Mesher
   - Tetrahedral Mesher
   - Trimmer Mesher
   - Thin Mesher
   - Advancing Layer Mesher
4. Generalized cylinder and/or embedded thin mesher (for polyhedral)
5. Prism layer mesher
6. Extrusion mesher

The Thin Mesher is a stand-alone volume mesher. If the Thin Mesher is activated, the pipeline can only be composed of the surface meshers plus the selected volume mesher.

For a given continuum, models are selected and changed using the mesh model selection dialog, which assists you with making the appropriate combination of models.

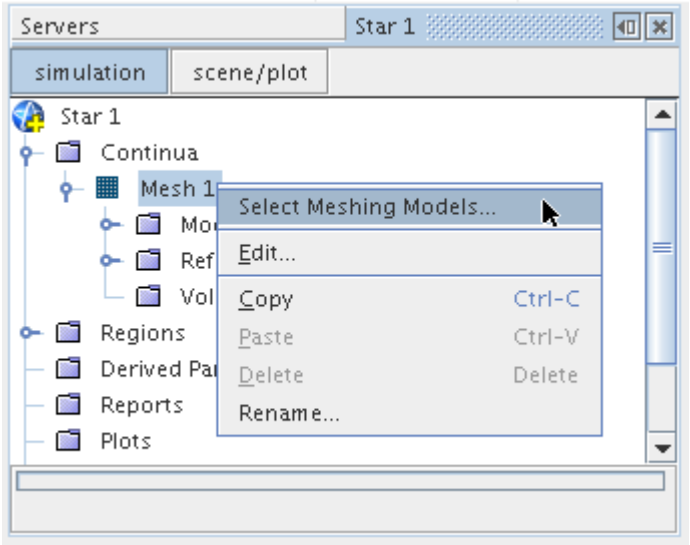
Contents:

Using the Mesh Model Selection Dialog
Viewing the Mesh Models Using the Object Tree

Using the Mesh Model Selection Dialog

The *Meshing Model Selection* dialog is used to select mesh models and/or change the existing model selection for a simulation.

To activate the dialog for a continuum, open the **Continua** node and right-click on the mesh continuum node name (say **Mesh 1**). In the context menu, select the item **Select Meshing Models...**



This same menu item is available in the pop-up menu of the **Models** manager node under the mesh continuum node.

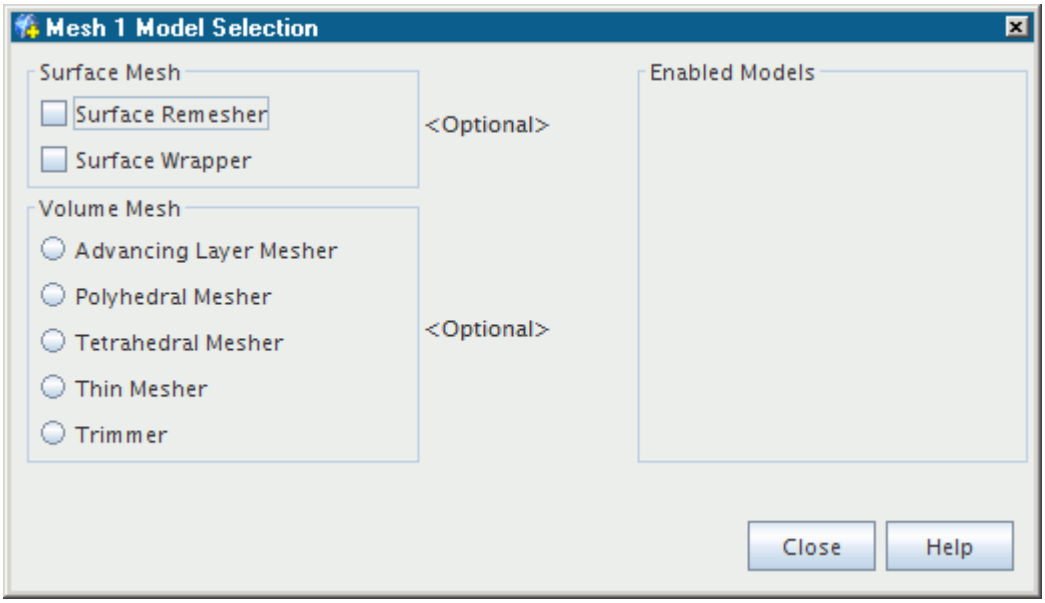
The *Meshing Model Selection* dialog that appears can be used to either select new mesh models, review the existing models, or change the model selection.

Meshing model selection and deletion can only be done through this dialog:

- The dialog is similar to a wizard in that it can assist you through a fairly complex process selection network.
- It is unlike a wizard in that it does not have the traditional **Next** and **Back** buttons.
- It does not restrict you to answering questions in a particular order.

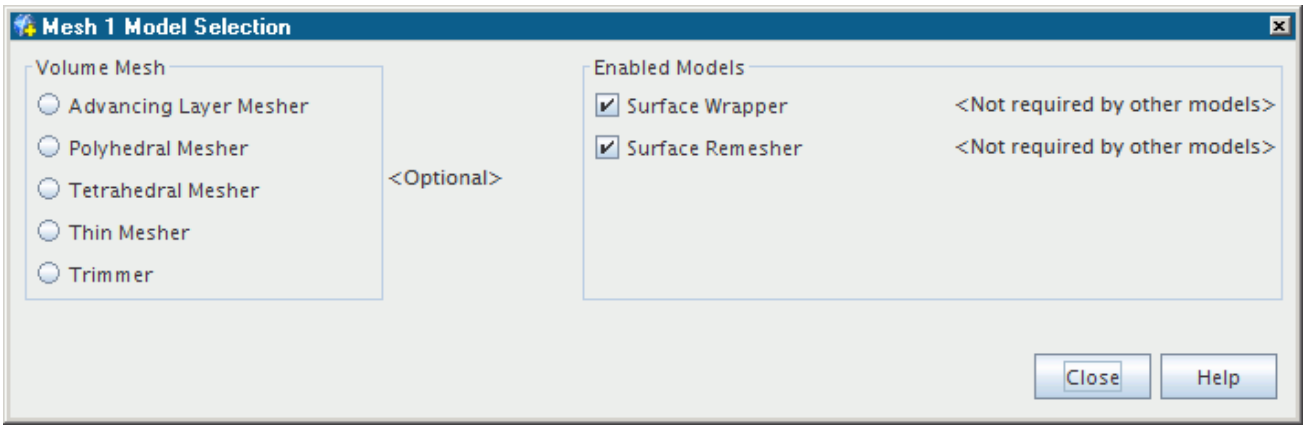
Selecting Mesh Models in the Dialog

The *Mesh 1 Model Selection* dialog appears as shown below when first opened for a mesh continuum that has not yet been defined.

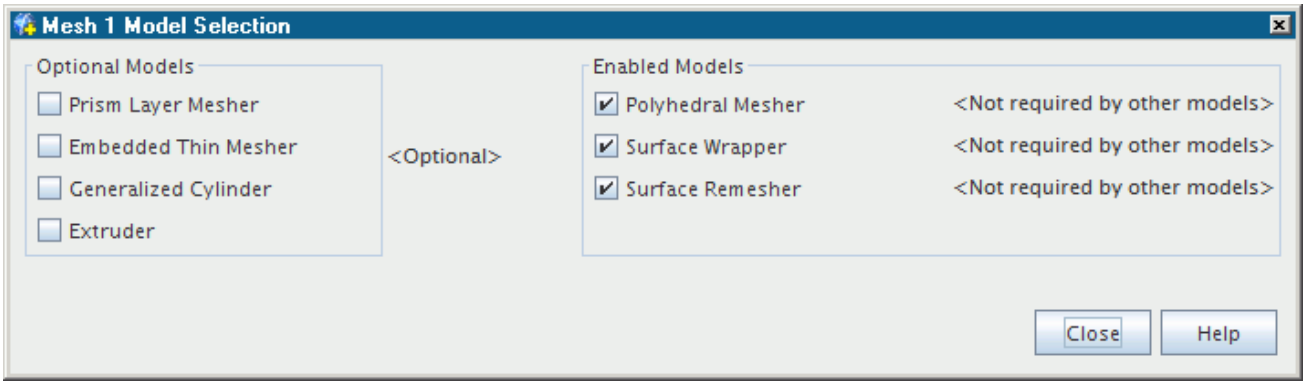


The *Enabled Models* group box on the right-hand side of the panel is empty, indicating that no models have been selected. Two types of models can be selected, namely *Surface Mesh* and *Volume Mesh*. The **<Optional>** labels indicate that the selection of these items is not mandatory.

For example, if you wanted to select the surface meshing models, namely the surface remesher and surface wrapper, then click the checkbox next to each. The models are then activated and listed in the *Enabled Meshing models* box.



The *Mesh 1 Model Selection* dialog changes dynamically in response to the selections made. For instance, if the **Polyhedral Mesher** model option is selected, then the optional **Prism Layer Mesher**, **Embedded Thin Mesher**, **Generalized Cylinder**, and **Extruder** models then become available.



The embedded thin mesher and generalized cylinder meshers are specialized options that are only available when the polyhedral volume meshing option has been selected. These two options are not available for the trimmer and tetrahedral volume meshers, although a tetrahedral mesh option can be selected afterwards using the properties node for the thin mesher model.

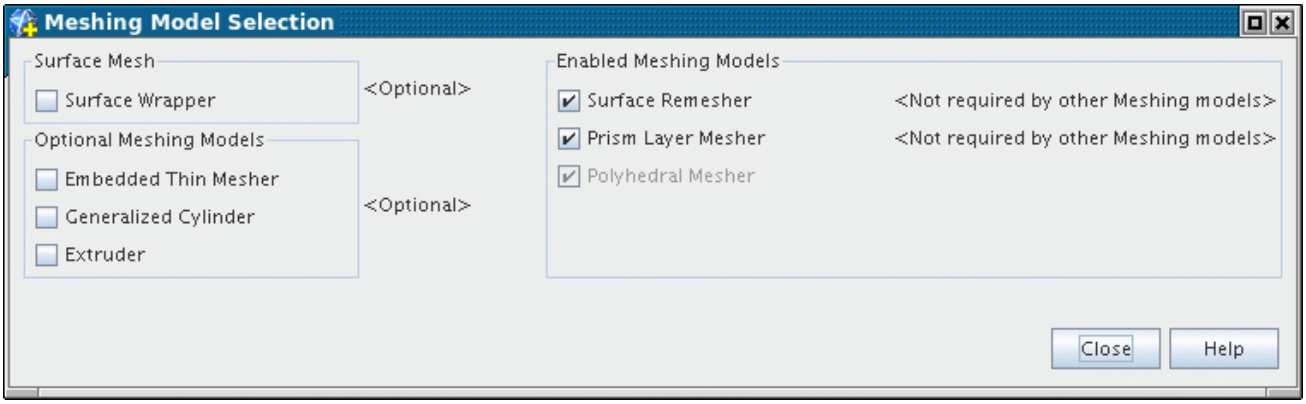
Make the appropriate model selection for the continuum before pressing the **Close** button to complete the model setup. For regular use, the guideline is to follow a particular sequence when selecting your models.

The dialog is “live” in that the server is updated each time that you change the state of a checkbox or radio button. Changes are also reflected in the tree when you have the relevant nodes open.

Reviewing Selected Mesh Models

The *Meshing Model Selection* dialog can be used to view the models that have been activated in a continuum. If the dialog has been closed, it can be reopened by right-clicking on either the **Mesh** node or the **Models** node and choosing the **Select Meshing Models...** item in the context menu.

As shown below, the **<Optional>** models group box on the left-hand side of the dialog lists the models that have not been selected. The *Enabled Meshing models* group box on the right-hand side shows all the meshing models that have been activated. The options that are grayed out are required for other (non-grayed out) models and therefore cannot be deactivated without first deselecting the appropriate model.



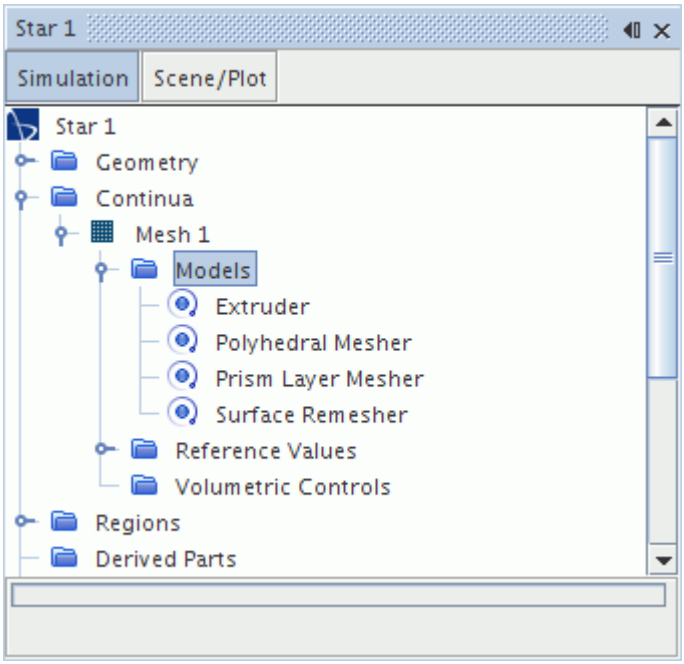
Changing Mesh Model Selection

The meshing models that are selected for a continuum can be modified in order to change the characteristics of the resulting meshing during the mesh generation process. Optional models in the group box on the left-hand side can be activated at any time before mesh generation; conversely, activated models can be deselected and new optional models can be reselected instead.

Viewing the Mesh Models Using the Object Tree

The meshing models that have been activated for a given mesh continuum can be viewed in the object tree by opening the **Models** node.

The tree view expands as shown below:



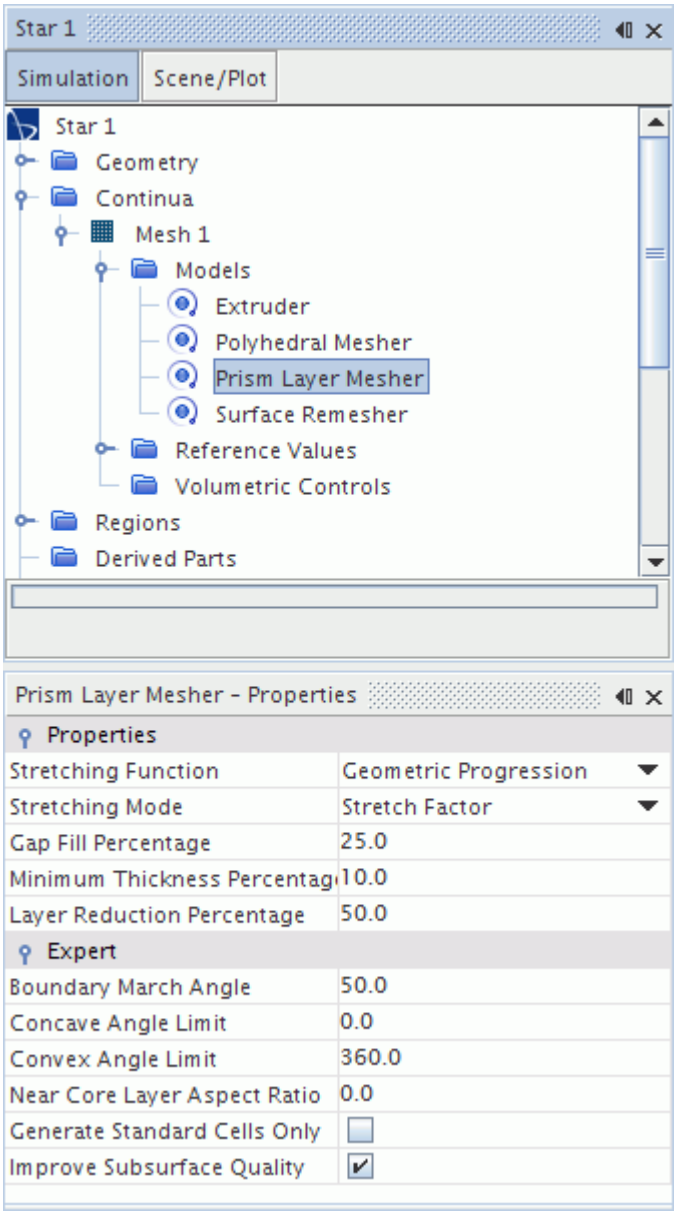
Setting Mesh Model Properties

This section describes how to set the properties of a mesh model

To set the properties of a mesh model:

1. Open the **Models** node in the mesh continuum of interest and then click the desired model name.

The *Properties* of the particular model are accessible for viewing and editing.



In general, the default properties for each model are suited for most applications although some adjustment can be necessary for some examples depending on the complexity of the geometry. Additional details on recommended changes to the properties are discussed in the relevant sections on surface and volume meshing.

For more information, see Surface Meshers Overview and Volume Meshers Overview.

Setting Mesh Controls and Values

This section describes how to set mesh controls and values.

In addition to the reference values on a mesh continuum, mesh values can also be set on the following simulation objects:

- Regions

- Boundaries
- Interfaces
- Feature Curves
- Volumetric Controls

Not all mesh values can be set on all of these objects. The particular values that can be set, and the objects on which they can be set, are determined by the meshing model.

In the absence of any custom mesh value, the reference value on the mesh continuum is used to construct the mesh. Custom mesh values override continuum mesh values according to the following order:

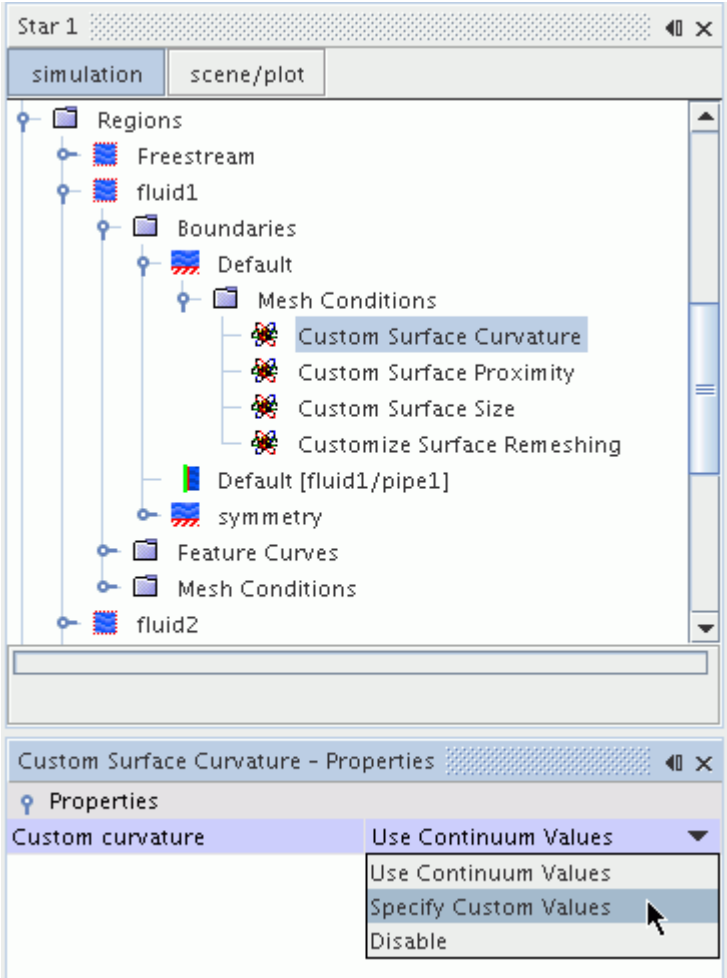
- Mesh values that are set on regions override equivalent values that are set on continua.
- Mesh values that are set on boundaries override equivalent values that are set on regions or continua.
- Mesh values that are set on interfaces override equivalent values that are set on the corresponding boundaries.
- Mesh values that are set on feature curves override equivalent values that are set on boundaries.

Mesh values that are set on volumetric controls also override equivalent values that set elsewhere, but the behavior in relation to boundaries depends on whether a surface or volume mesh is affected.

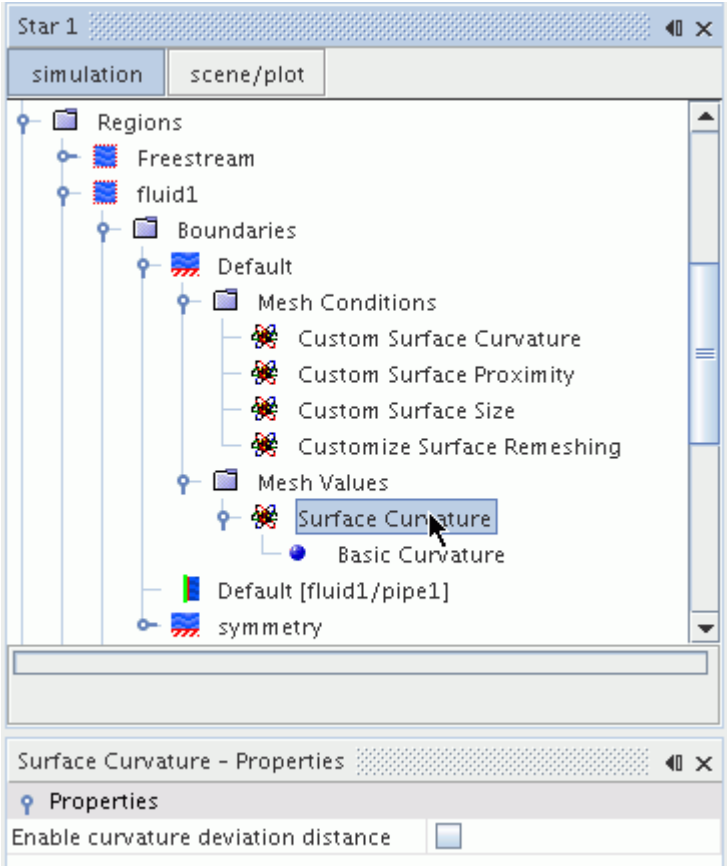
The general procedure for applying a custom mesh condition is as follows:

1. Expand the **Mesh Conditions** node of the relevant simulation object, for example, a boundary.
2. Some customization nodes have a drop-down menu of options; others have a checkbox. Make the required selection or activate the checkbox.

   In the example below, the option to **Specify Custom Values** for surface curvature is chosen.



3. If the customization requires any input, the **Mesh Values** folder is added to the parent object, and appropriate mesh value nodes are shown within it. Set the necessary inputs on the mesh value nodes.



Additional details on custom settings are discussed in the relevant sections on surface and volume meshing.

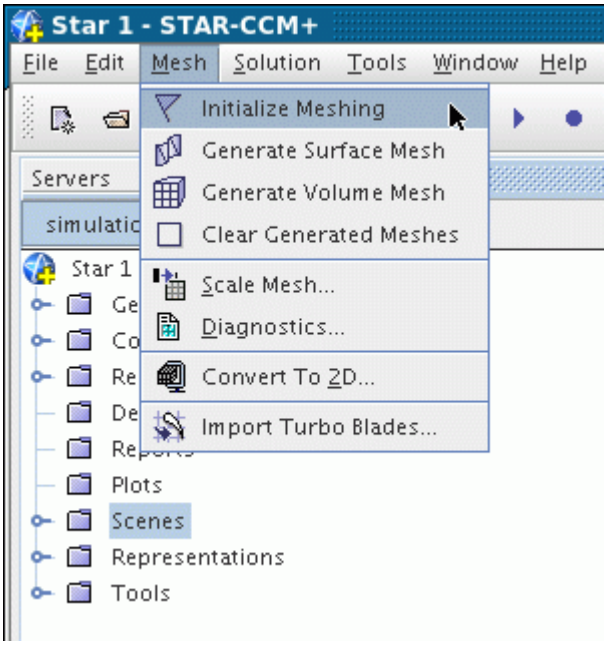
Initializing the Mesh

This section describes how to initialize the mesh.

Meshing initialization is required in two general situations:

- Whenever geometry parts are used to provide the starting surface for the meshing models. See What is a Geometry Part?.
- If design parameters relating to 3D-CAD leaf-level parts are changed, requiring a parts update to be performed, and in-turn, a reinitialization of the starting surface. See What Are Leaf-Level Parts? and Updating 3D-CAD Parts.

The option is available in the meshing toolbar (**Initialize Meshing** button) or by using the **Mesh** > **Initialize Meshing** menu item:

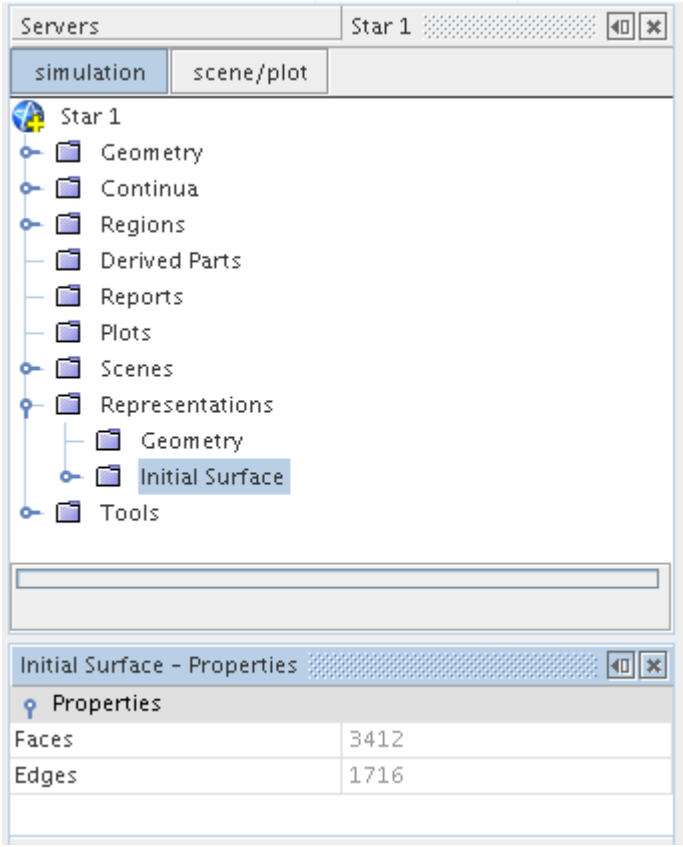


Before meshing initialization takes place, the following should have been completed:

- At least one mesh continuum model has been created.
- One or more leaf-level parts have been assigned to regions.

The meshing initialization option is available as soon as a mesh continuum has been created. However, using the option at this stage results in no initialization being performed. Only when at least one leaf-level part has been assigned to a region can meshing initialization proceed and result in the generation of an **Initial Surface** representation.

Clicking the initial surface node provides a summary of the surface:



The initial surface is the surface representation of whatever leaf-level part surfaces and curves that have currently been assigned to regions. If you change the part surfaces and/or curves that are associated with the region boundaries, then perform the initialize meshing operation again in order to update the meshing pipeline. The link between a part and a region is broken if the part is deleted. Since the initial surface representation is associated with a region, deleting a part does not delete the existing mesh for the region. If you use the initialize meshing option in this case, STAR-CCM+ does not clear the mesh.

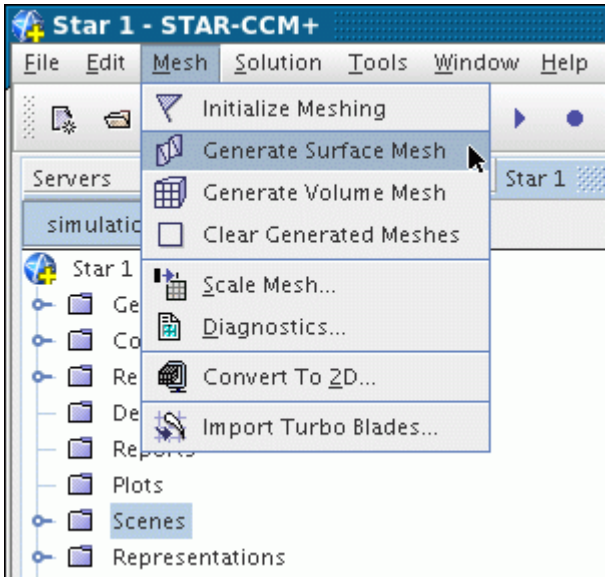
If you forget to perform the meshing initialization step before surface or volume meshing, then is automatically done for you based on the current surface and curve assignments. Additionally, if design parameters relating to 3D-CAD parts have been changed since the last surface or volume mesh operation then the initial surface is automatically updated when reinvoking the meshing pipeline.

Generating the Surface Mesh

Once the surface mesh models and properties have been set up, the surface mesh generator can be launched.

To start the process:

1. Click the (**Generate Surface Mesh**) button in the toolbar or use the **Mesh** > **Generate Surface Mesh** menu item.



The text output from the selected meshing models is shown in the *Output* window. This output includes information on the time it took for each stage of the surface meshing pipeline (for example, surface remeshing), as well as the overall time for the whole process. The surface mesh generation process continues until the completed surface is formed or the mesh generation task is aborted.

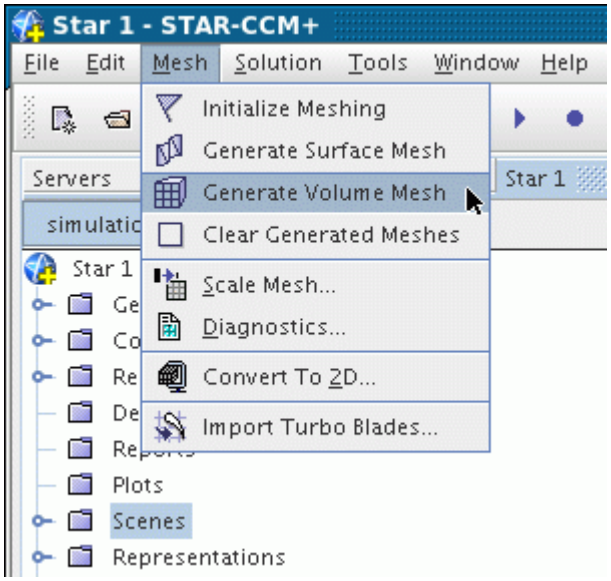
Once the process is complete, the surface mesh representation can be visualized on the plot display in the same way as other mesh representations.

Generating the Volume Mesh

Once the volume mesh models and properties have been set up, the volume mesh generator can be launched.

To start the process:

1. Click the (**Generate Volume Mesh**) button in the meshing toolbar or use the **Mesh** > **Generate Volume Mesh** menu item.



The selected meshing models generate output that is shown in the *Output* window, and the progress of meshing is shown in the status bar. The output includes information on the time it took for each stage of the volume meshing pipeline to complete, as well as the overall time for the whole process.

The volume mesh generation process continues until the completed volume is formed or the mesh generation task is aborted.

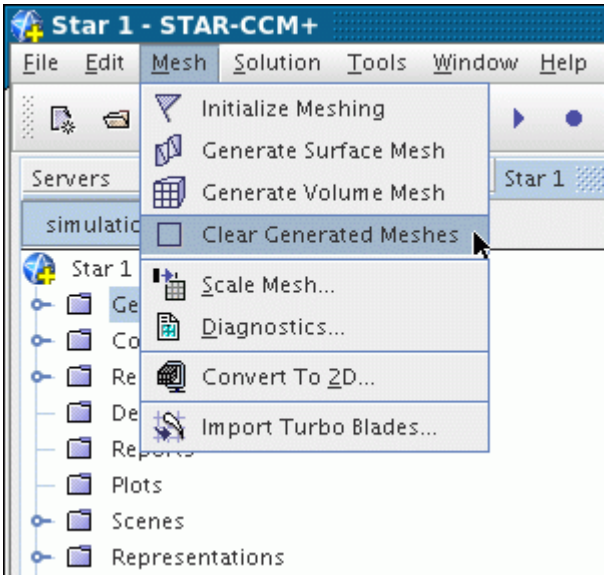
Clearing the Mesh

This section describes how to clear the mesh.

Each surface and volume mesh representation that STAR-CCM+ generates is saved automatically as part of the simulation. When generating a new mesh, the old representation is overwritten as part of the process. Some of the reasons for wanting to clear the generated meshes and representations are:

- to reduce the size of the simulation file
- to start the meshing process over
- to modify the boundaries or patch definitions in the import surface

You can clear all the generated meshes by pressing the (**Clear Generated Meshes**) button in the meshing toolbar or use the **Mesh** > **Clear Generated Meshes** menu item.



The mesh representations nodes are also deleted and any scene displays revert to the import, geometry, or initial representation, which are unaffected by this action.

Replacing a Mesh Region

A region containing a volume mesh in STAR-CCM+ can be replaced by another topologically identical volume mesh at any point during the analysis.

One of the main reasons to replace a mesh is to improve the solution integrity and convergence characteristics by replacing a coarser mesh with a finer one in the middle of the analysis. The current solution is automatically mapped to the new mesh region in order to improve the restart conditions. The mesh replacement option differs from deleting and reimporting a new mesh with respect to this last point; no solution mapping is done in the latter case.

The assumptions that are made when replacing a mesh region with a new one are:

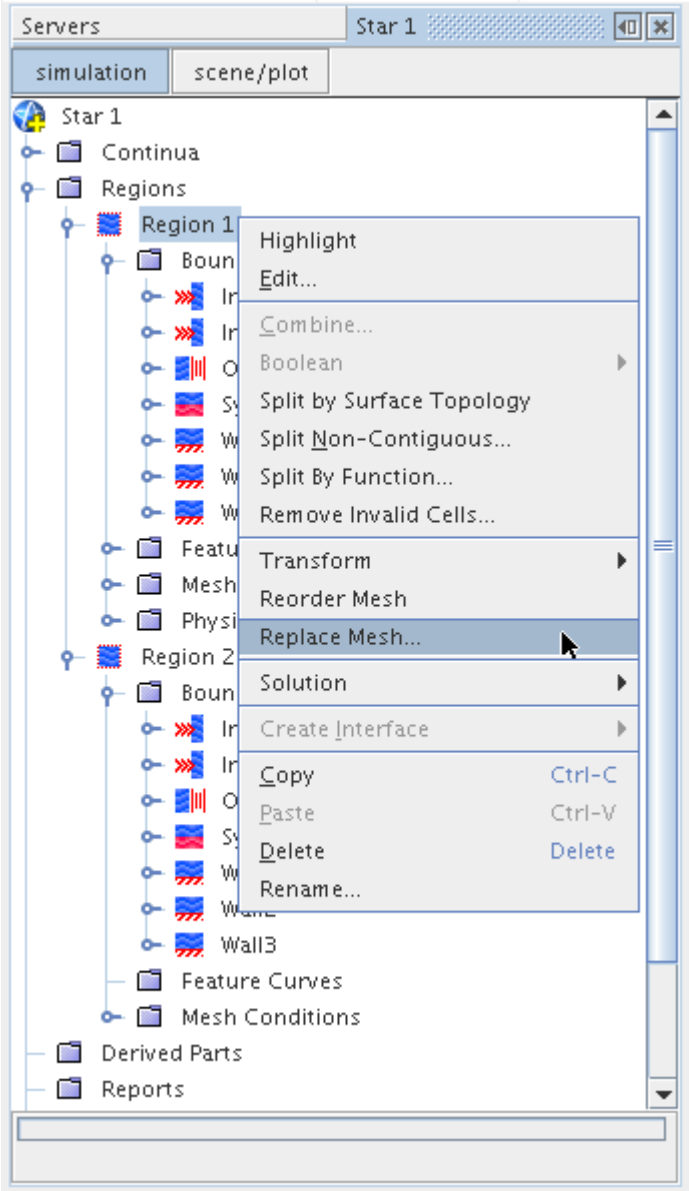
- the region topology is identical in both cases
- the number of boundaries are the same and use the same source names

The grid structure of the two regions does not have to be the same. In other words, a polyhedral type mesh can replace a tetrahedral one, or any other combination of cell shapes allowed.

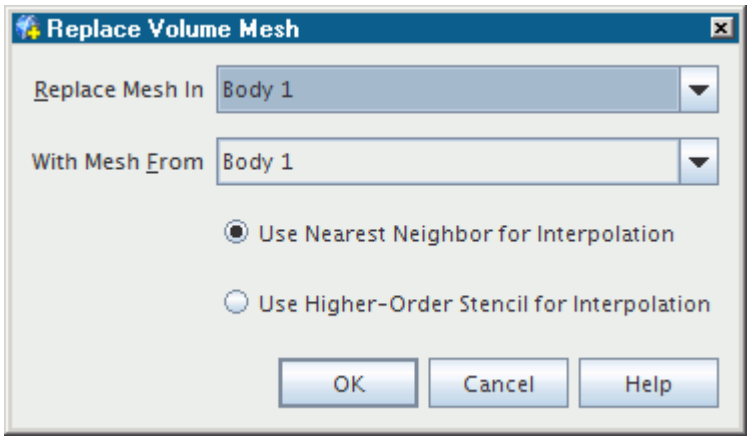
Before replacing a mesh region, import the new volume mesh region into STAR-CCM+ and configure the boundaries to have the same number and source names as the original mesh region.

To replace a mesh region:

1. Open the region node of interest and right-click on the name. In the pop-up menu that appears, select the **Replace Mesh...** option.



The *Replace Volume Mesh* dialog appears.

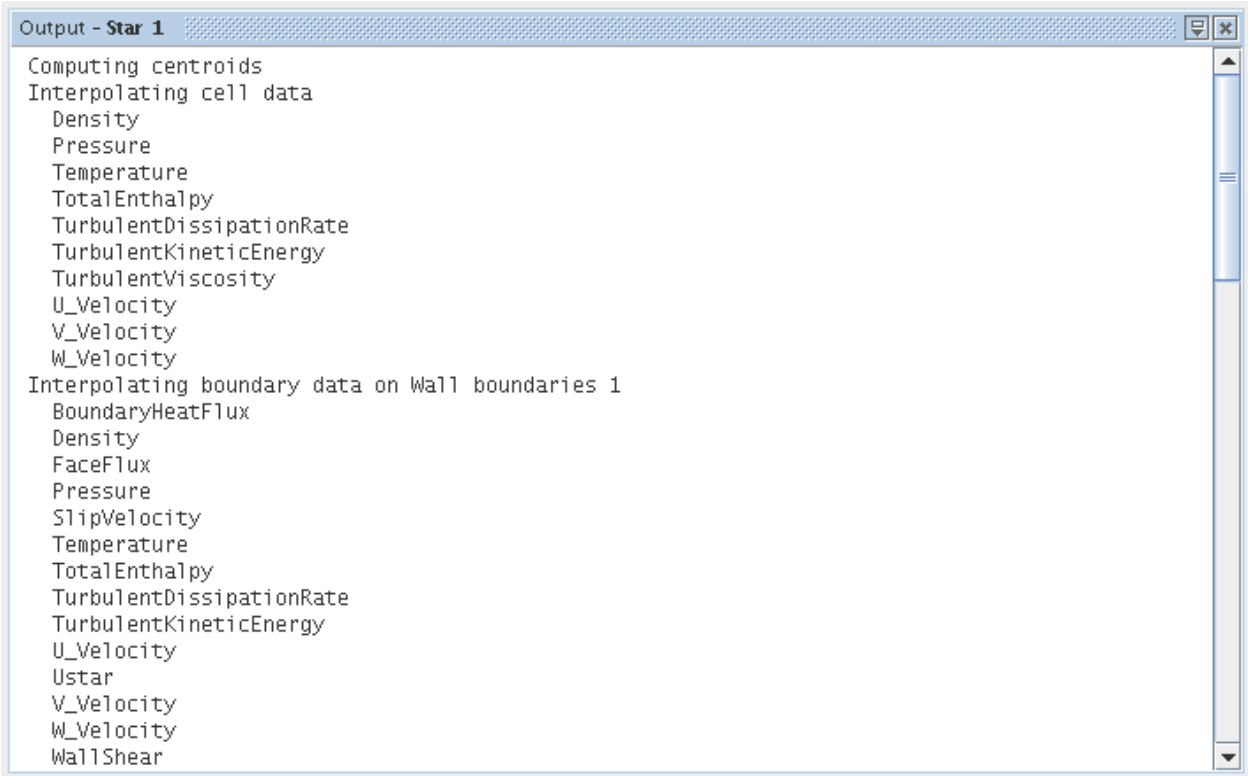


2. Two options are provided for interpolation:
   - **Use Nearest Neighbor for interpolation**

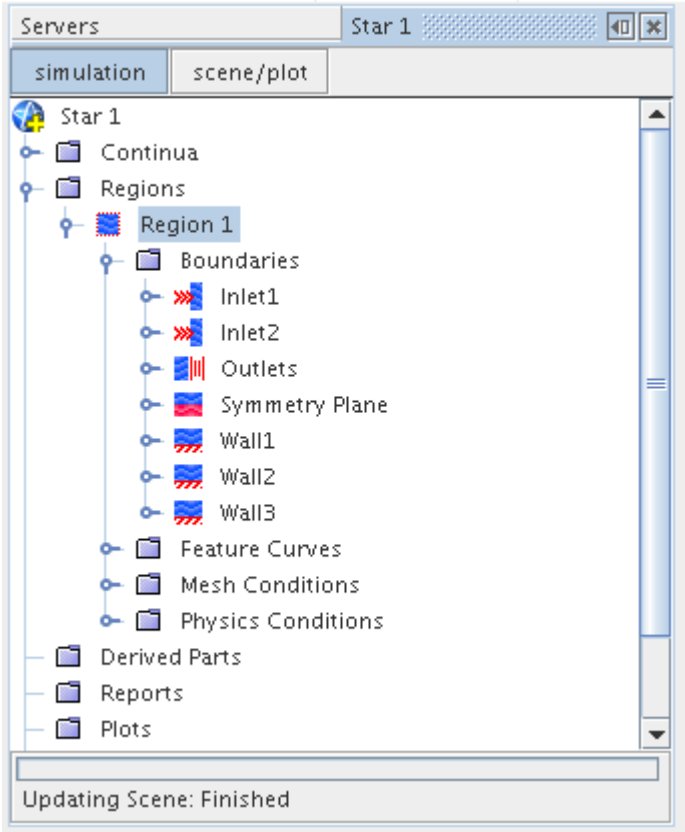
     The cell/face value on the new mesh is assigned the value of the cell/face on the old mesh whose cell/face centroid is nearest to the cell/face centroid of the new mesh. This option does not provide smooth interpolations of the results from one mesh to the other.
   - **Use higher-order stencil for interpolation**

     The cell/face value on the new mesh is smoothly interpolated from a stencil of cells/faces on the old mesh using the distance-weighted, least square interpolation method. Generally speaking, the method would be preferred when replacing the mesh at some point during a transient analysis. It can also be useful for steady problems, but the requirement for smoothness is not as critical, as the simulation can continue to converge to a steady state solution after the mesh replacement. The technique is slightly more computationally expensive than the nearest neighbor mapping, but it can offer more fidelity when mapping with thin parts.
3. Using the pull down menus, select the region to replace and then the region that contains the new mesh. Press the **OK** button.

   The progress bar notifies you once the process is complete and a summary message is displayed in the *Output* window:



Solution data in the old region is mapped to the new one, and the old region is deleted. The simulation tree shows the updated region structure.



The physics and boundary conditions from the original region is copied across to the region that replaces it.

Feature Curves

Feature curves are an important part of the surface definition in order to produce a high-quality mesh (both surface and volume).

As part of the meshing process they are used to define edges and surface features from the surface geometry all the way through to the final volume mesh. If they are not included in the starting surface, then such features are not preserved. Feature curves can also be used to specify custom sizes on edges and features for the surface wrapper, surface remesher, and trimmer meshing models.

Feature curves can be imported from third party pre-processing packages into STAR-CCM+ using any of the allowable surface file formats other than STL and PATRAN. They can also be created automatically during the import of neutral data files such as IGES, STEP, and Parasolid Transmit and native CAD files such as PRO/E, CATIA, and SolidWorks.

Feature curves can be computed automatically after surface import based on a number of criteria. They can also be manually added and edited. Similar to other objects, feature curves can be combined, split, renamed and deleted.

Contents:

What Are Feature Curves?
Computing Feature Curves

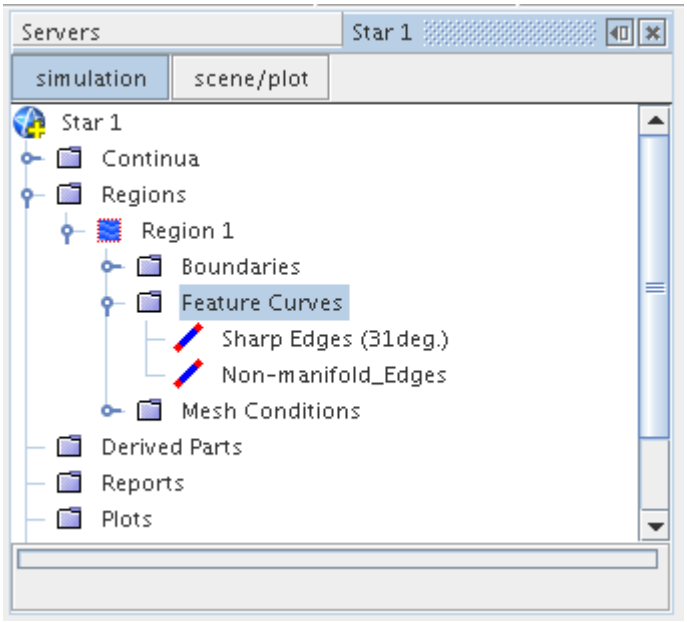
Adding New Feature Curves
Editing Feature Curves in the Current Scene
Splitting Feature Curves
Combining Feature Curves

What Are Feature Curves?

A feature curve in its most basic form is a two-dimensional cell with a vertex at each end.

They are exclusively used in surface meshes to indicate where sharp edges exist, and to define surface details to maintain in the final volume mesh.

When imported or created in STAR-CCM+, feature curves are listed under the **Feature Curves** node for each region. An example showing two sets of feature curves is shown below:



If the surface remesher is used to improve the starting surface quality, then including feature curves is critical to preserve sharp edges and/or surface detail (such as imprints). If the feature curves are not included, then edges are rounded off and surface detail is lost.

Including feature curves for edges when utilizing the surface wrapper is not as critical as the wrapper feature angle, which can be used to extract features to maintain. However, for features that have an angle less than the angle specified, or for surface details such as imprints, then it is recommended that feature curves be used to preserve the definition.

You can have as many feature curve entries as you want for the surface. However, it is typical that only the critical edges (say features over 45 degrees) are included in feature curves to limit the complexity of the model.

Computing Feature Curves

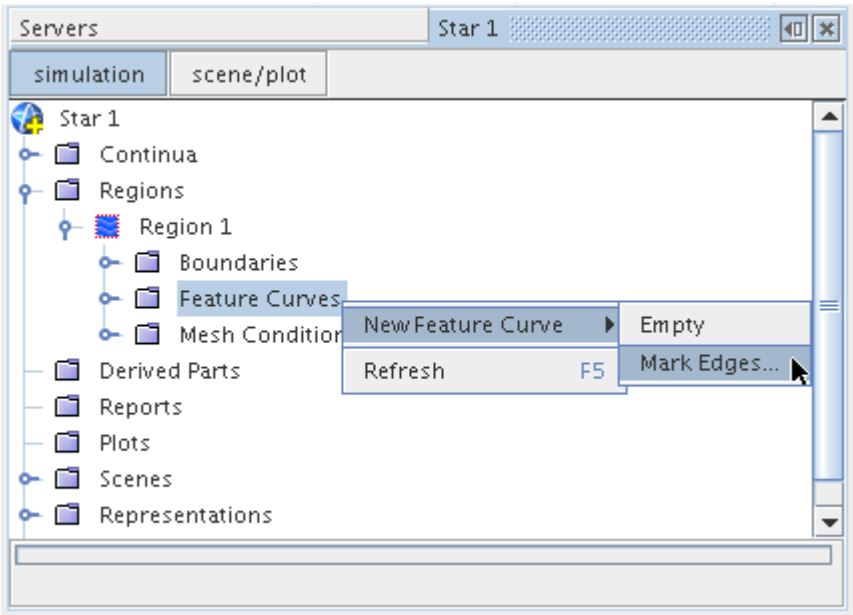
Feature curves can be automatically created for a given set of boundaries by marking edges using different criteria.

Curves can be created for a single boundary or multiple boundaries simultaneously. Thereafter, further feature curves can be manually added or the original set can be edited so that the exact edge curve definition is obtained.

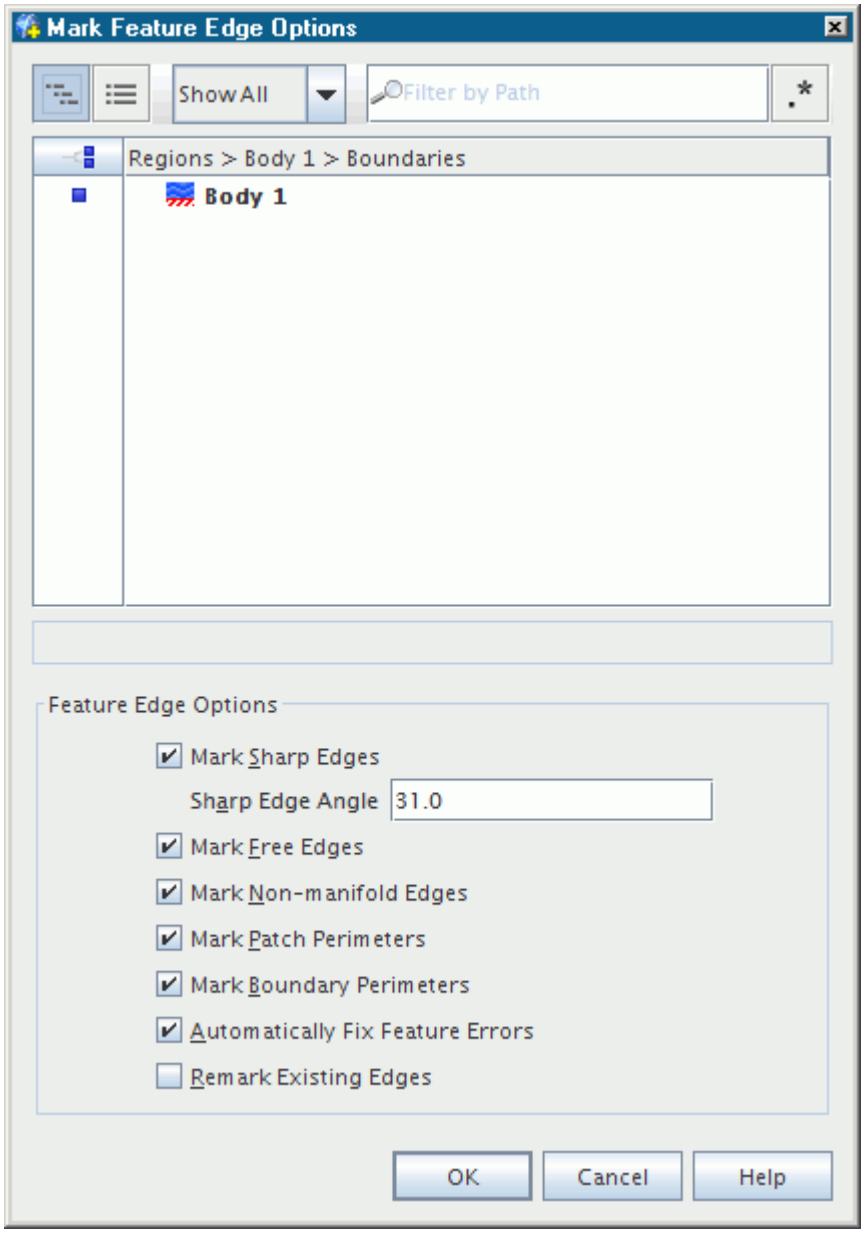
Before feature curves can be created, a surface mesh type representation (such as import or initial surface) has to exist.

To create a new set of feature curves:

1. Right-click on the **Feature Curves** node for the desired region and select **New Feature Curve** > **Mark Edges...** from the pop-up menu:



This action opens the *Mark Feature Edge Options* dialog, which can be used in the same way as the object selection dialog to select the boundaries on which to mark feature edges:



2. First, select the boundaries that require feature curves.

   The *Feature Edge Options* criteria can then be reviewed and the appropriate selections for the given boundary set made. The following options are available:

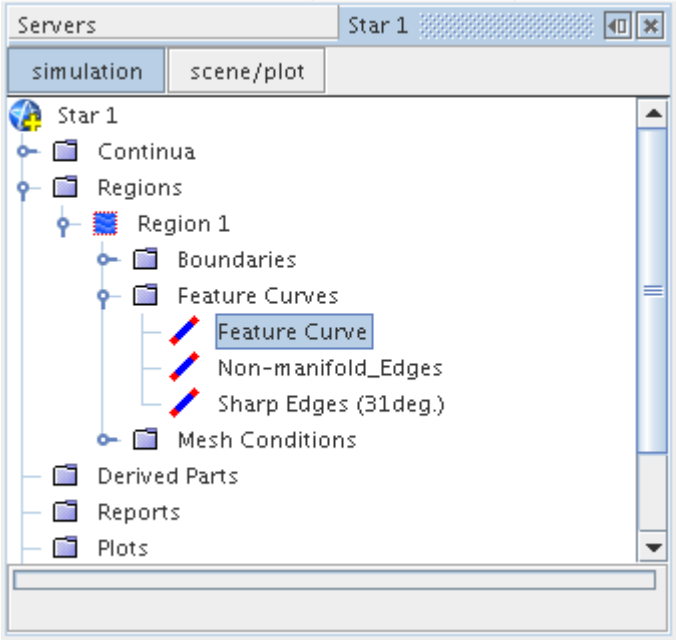
   - **Mark sharp edges** - Allows feature curves to be created on edges whose attached faces make an angle that is sharper than the supplied Sharp edge angle value (31 degrees by default)
   - **Mark free edges** - Allows feature curves to be created for any free edges
   - **Mark non-manifold edges** - Allows feature curves to be created for any non-manifold edges
   - **Mark patch perimeters** - Allows feature curves to be created based on the underlying imported patch definition
   - **Mark boundary perimeters** - Allows feature curves to be created based on the perimeters of each of the currently selected boundaries
   - **Automatically Fix Feature Errors** - Fixes all feature errors in explicit feature edges.

- **Remark existing edges** - Moves any previously marked edges that fit the criterion into a new feature curve. When deactivated, any existing edges remain in their current feature curve

3. To mark feature edges, press the **Apply** button.

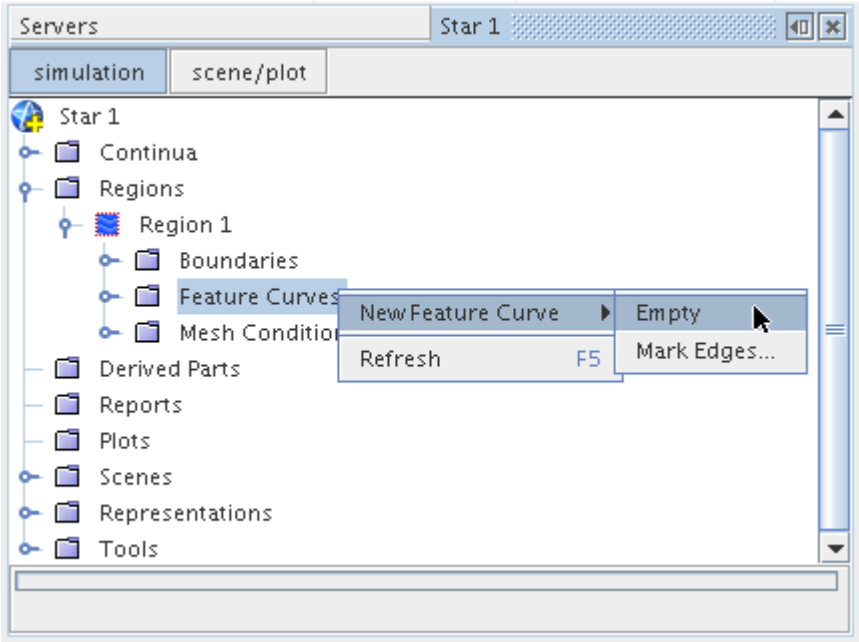
   The feature curve node name for the feature curves that are created is based on the method used. For example, if the **Mark sharp edges** option has been selected with the default angle criteria, then the feature curve node name would be **Sharp Edges (31 deg.)**. When more than one selection criteria are used, the generic node name **Feature Curve** is used.

   An example entry is shown below:



4. If you do not want to create feature curves automatically and prefer to add them manually, right-click on the **Feature Curves** node and select **New Feature Curve** > **Empty** from the context menu.

   This selection is shown below:



A new feature curve node (with the generic name **Feature Curve**) is created to which new feature curves can be added by hand.

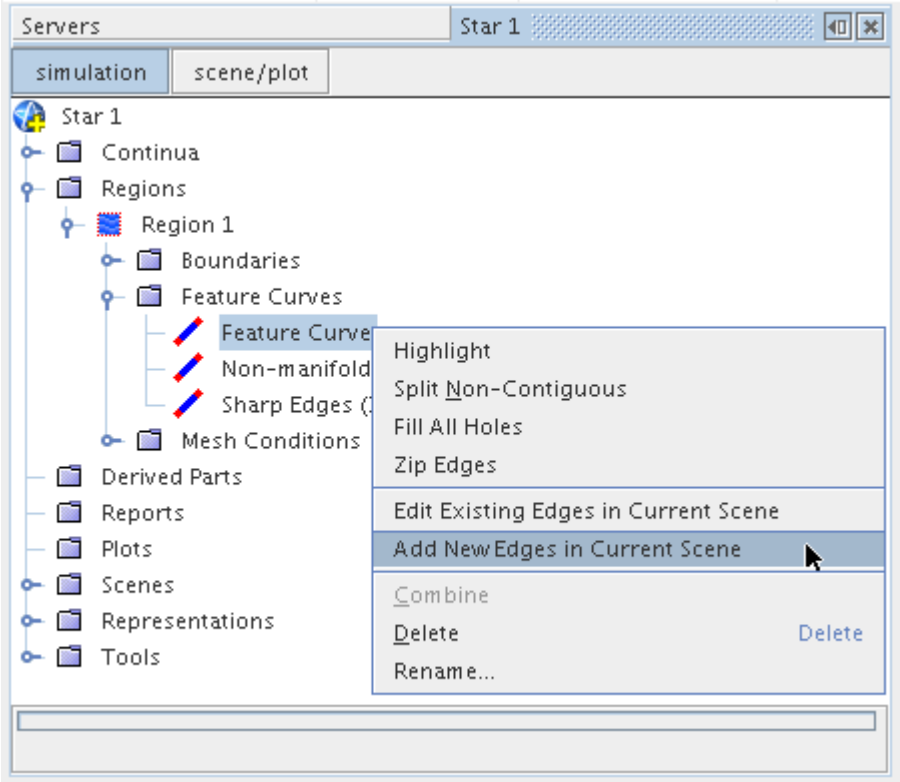
Adding New Feature Curves

Feature curves can be added by hand to an existing feature curve node.

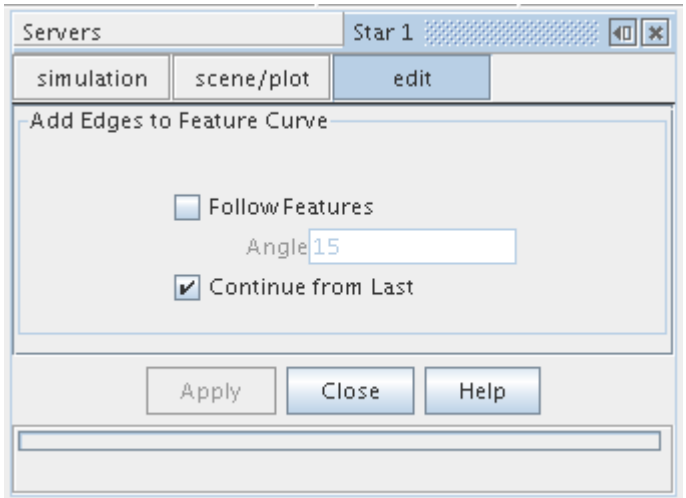
The node can be empty or contain feature curves that were previously imported or computed. Before feature curves can be created, a surface mesh type representation (such as import or initial surface) has to exist.

To add to an existing feature curve node:

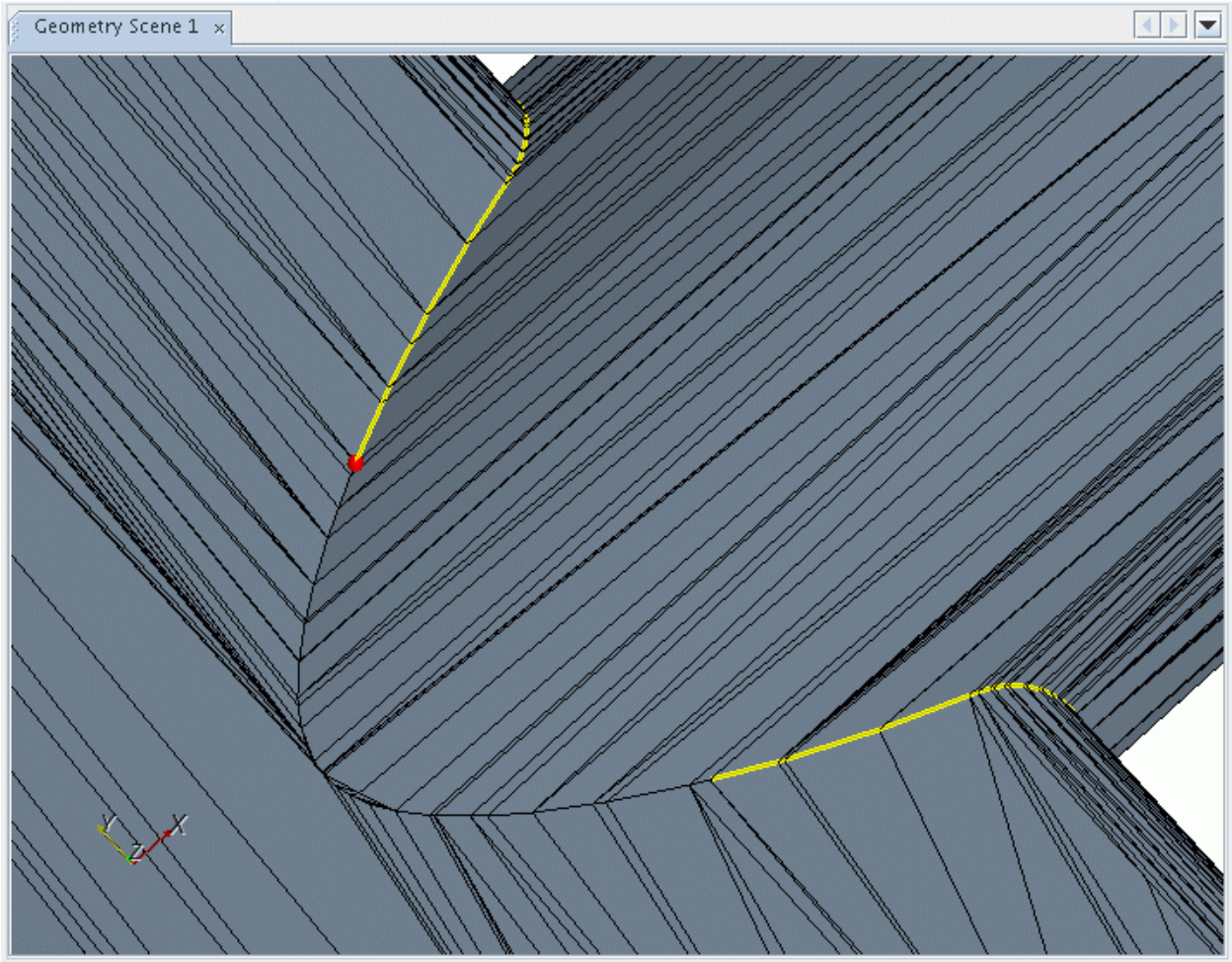
1. Right-click and select the **Add New Edges in Current Scene** option in the pop-up menu, as shown below:



The *Add Edges to Feature Curve edit* tab appears in the explorer pane:

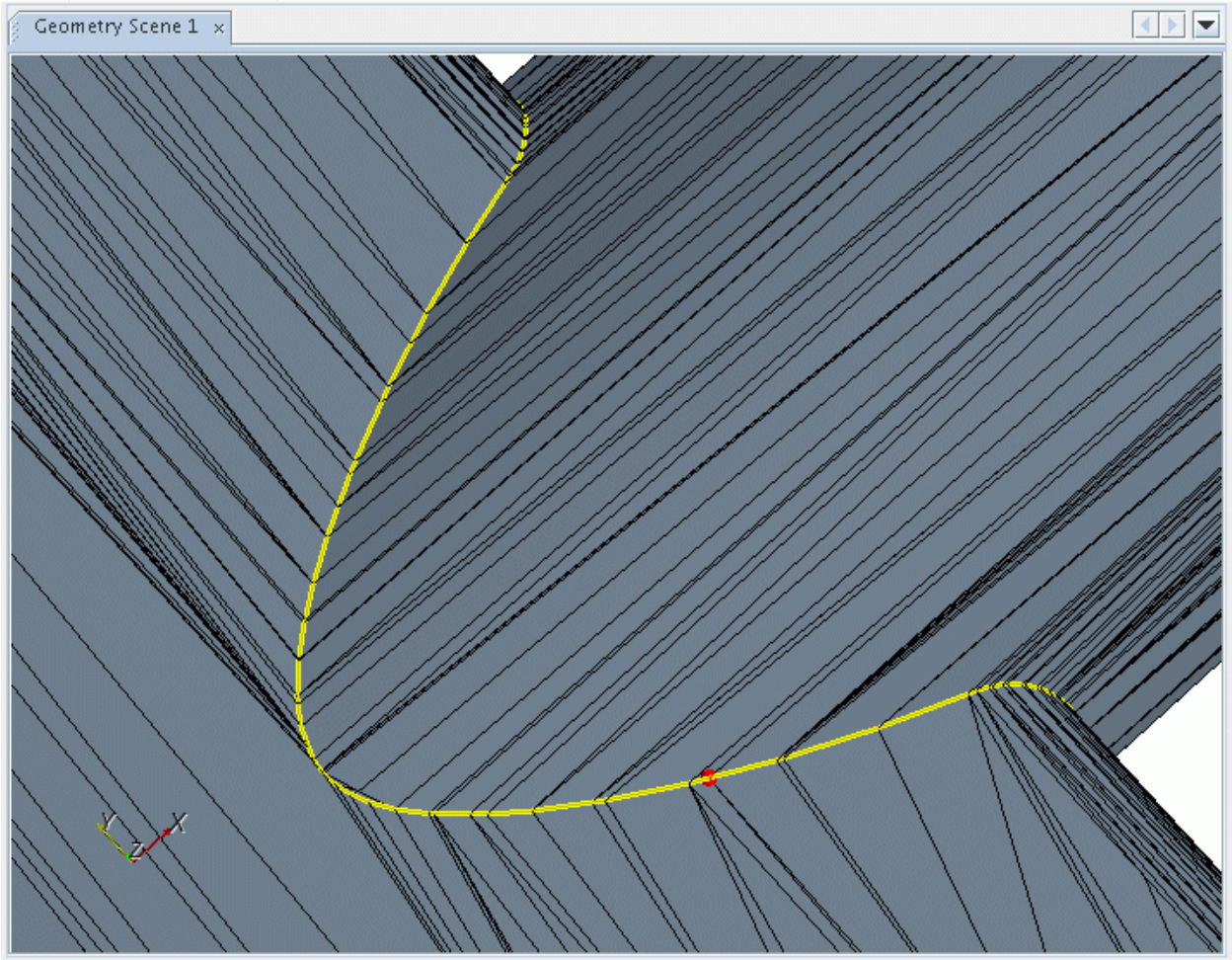


The existing feature curves belonging to the curve set are highlighted in yellow for easier viewing, as shown below:



2. To identify the feature curve start, click a vertex (a red ball appears) and then click a second vertex to identify the feature curve end (the red ball moves to this second location).

   By default the shortest path along the surface mesh is followed to create the feature curve. An example is shown below:



3. If you do not want the feature curve to continue from the last vertex that you selected, deactivate the **Continue from Last** option. You then have to select two new vertices for each set of feature curves you want to create.
4. If you want the feature curve to follow an existing surface feature edge, then turn on the **Follow Features** option and supply an edge **Angle** value (in degrees).
5. Once you have completed adding in feature curves, press the **Close** button to close the dialog and return to the normal scene view.

Any new feature curves that are created during the session are added to the existing feature curve node and displayer. Feature curves can only be added to a topologically connected surface: they cannot be made to span across one surface that is completely disconnected from another one.

Editing Feature Curves in the Current Scene

This section describes how to edit feature curves.

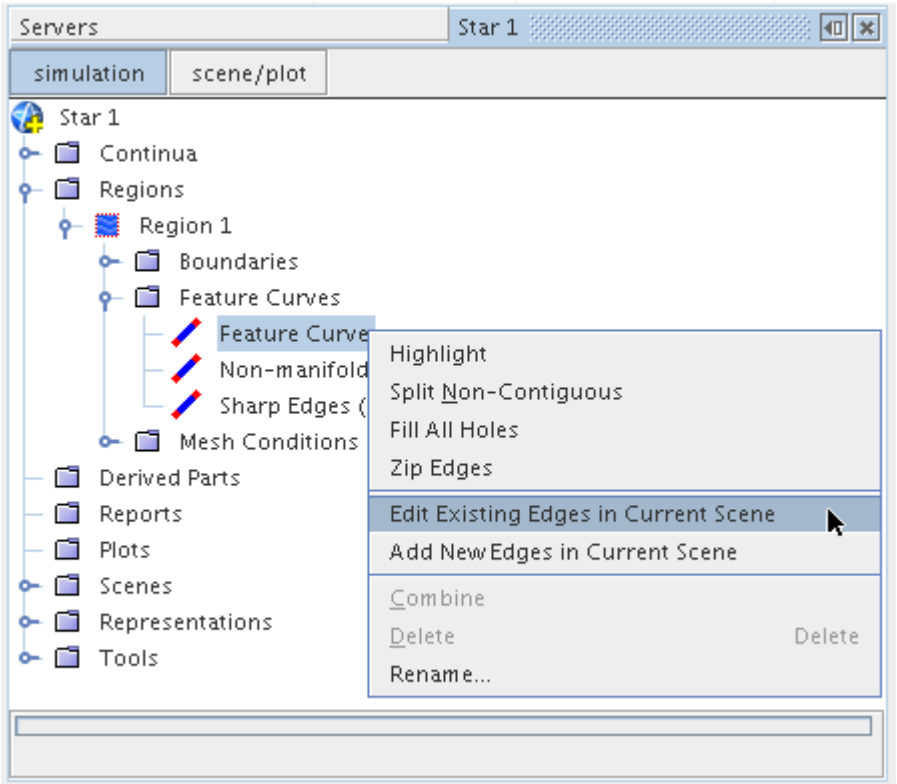
Feature curves can be edited by hand in the following ways:

- delete a part of or the whole of a feature curve
- assign a part of or the whole of a feature curve to a new feature curve node

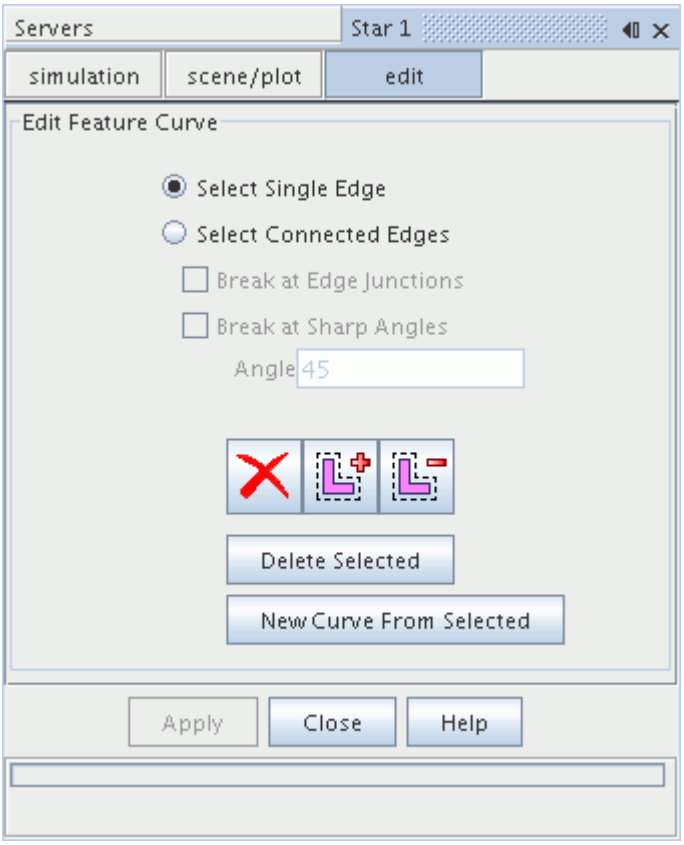
Before feature curves can be created, a surface mesh type representation (such as import or initial surface) has to exist.

To edit an existing feature curve node:

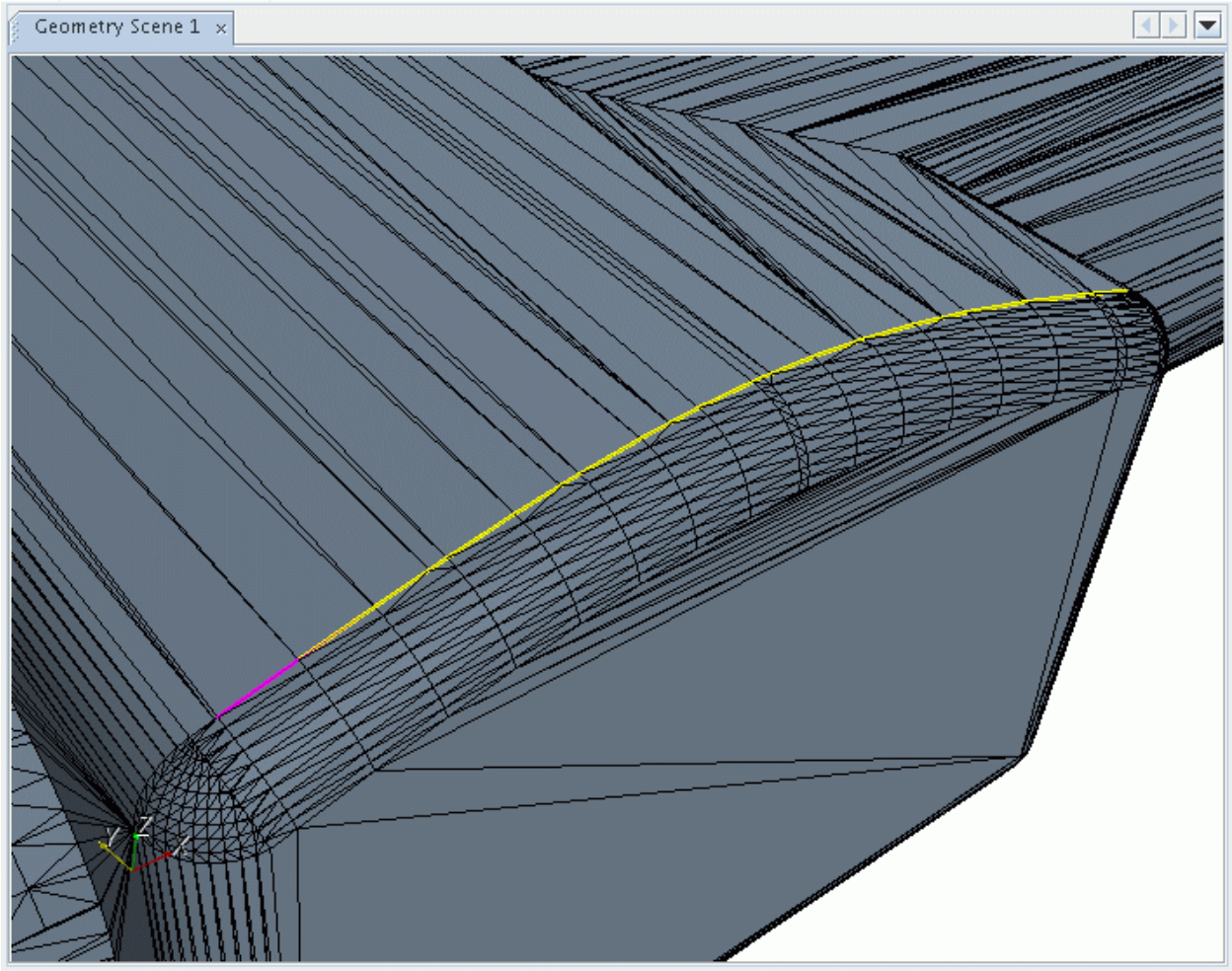
1. Right-click and select the **Edit Existing Edges in Current Scene** option in the pop-up menu, as shown below:



The *Edit Feature Curve* dialog appears in the *edit* tab:

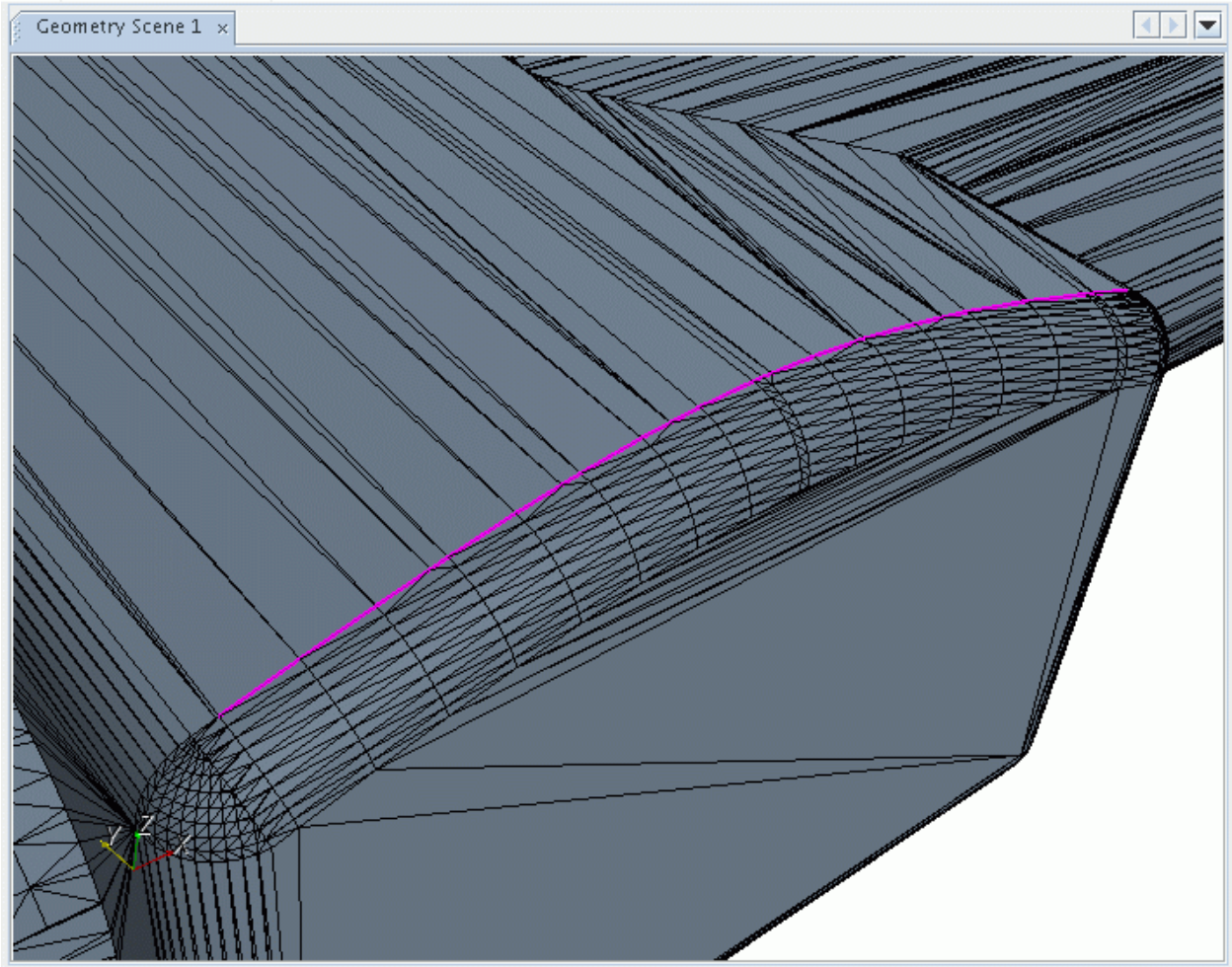


The existing feature curves in the feature curve set are highlighted in yellow for easier viewing. By default, clicking a highlighted feature curve only selects a single edge. An example is shown below:



2. If you want to select a set of continuous feature curves, then use the **Select Connected Edges** option, which allows features curves that share a common vertex to be selected in one go.

   An example is shown below:



3. When using the connected edges mode, you can specify further criteria to determine where to break the edge selection.

   The two options that are provided are:

   - **Break at Edge Junctions** - stop the selection at where three or more feature curves exist
   - **Break at Sharp Angles** - stop the selection at the specified **Angle** along the curve

4. The **Select Zone** and **Deselect Zone** buttons can also be used to select and unselect feature curves respectively, using a polygonal zone.

5. To clear an existing selection from any of the above described methods, press the **Clear Selection** button.

6. Once a set of feature curves has been selected, you can delete them by pressing the **Delete Selected** button or assign them to a new curve node by pressing **New Curve from Selected**.

   The new curve node name is based on the original one with an appropriate numerical suffix. These curves are colored red and can no longer be selected in the current operation.

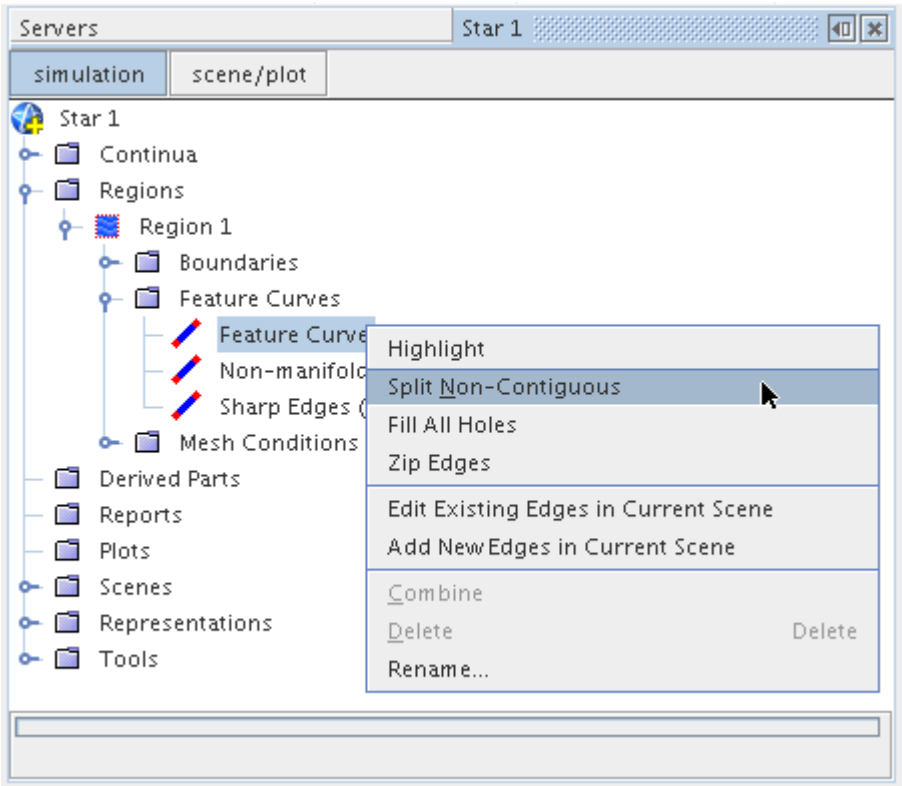
7. Press **Close** to complete the process and return to the normal scene view.

Splitting Feature Curves

Before feature curves can be split, a surface mesh representation (such as import or initial surface) has to exist.

To split non-contiguous feature curves:

1. Right-click on the feature curve name and select **Split Non-Contiguous** from the pop-up menu:



   More than one feature curve can be split (using multiple selection) at the same time.

2. If you want to split feature curves in some other way, then use the edit feature curves option instead, and manually select the curves. See Editing Feature Curves in the Current Scene.

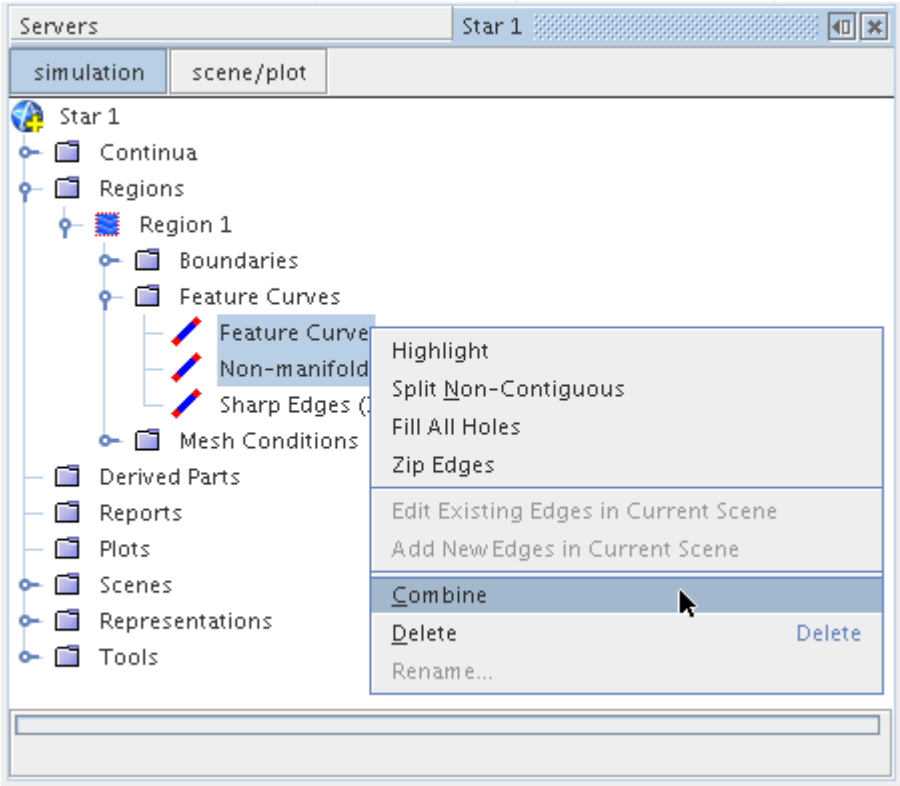
Combining Feature Curves

Two or more feature curves in the same region can be combined into one in order to simplify the meshing setup.

Before feature curves can be combined, a surface mesh representation (such as import or initial surface) has to exist.

To combine the feature curves:

1. Select the names in the simulation tree (using multiple selection) and right-click. In the pop-up menu, select **Combine**, as shown below:



The name that is used for the combined feature curve is the first one selected.

Using Feature Curves to Repair a Surface

Two different tools are available to help with basic surface repair within STAR-CCM+, both of which utilize feature curves.

The two surface repair options are:

- filling holes
- zipping edges

The hole filler can be used to close any closed loop (both single and multiple loops) planar hole within a surface. The edge zipper can be used to join two mismatched surface edges that are within close proximity to one another, so that a closed surface is formed.

A surface mesh representation (such as import or initial surface) has to exist before feature curves can be used to fill holes or zip edges.

Contents:

Filling Holes in the Surface
Zipping Edges

Filling Holes in the Surface

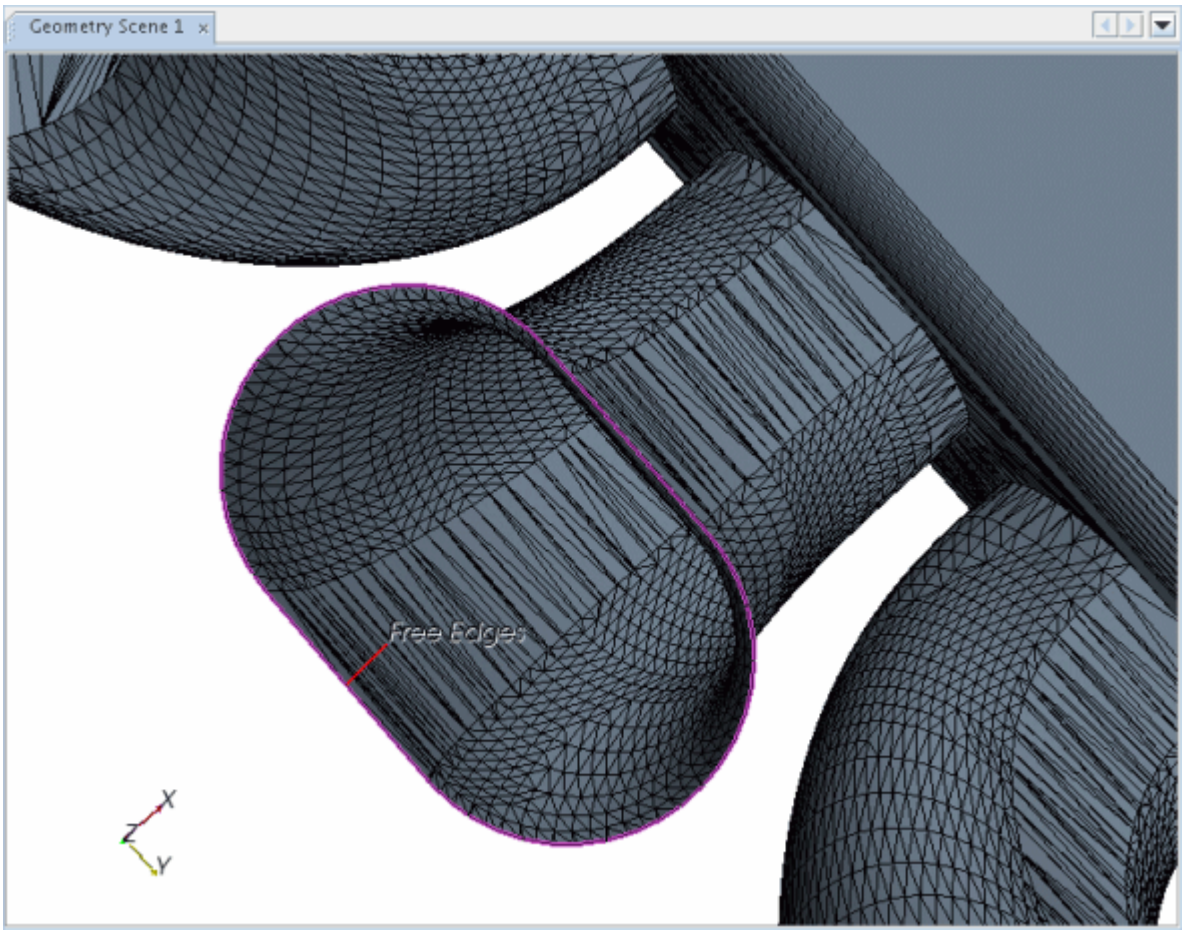
The hole filler can be used to close any arbitrary shaped hole using a planar surface.

Two different types of holes can be filled:

- Simple holes with a single connected loop of free edges
- Multi-loop holes with one loop inside the other

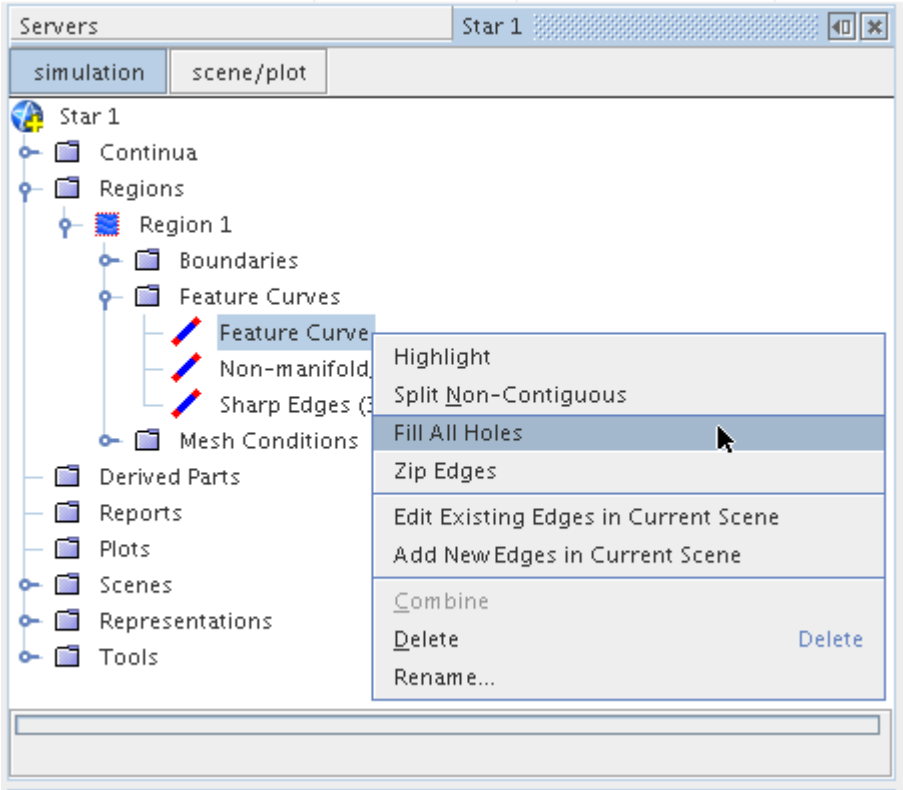
To fill a hole, first define feature curves in a continuous closed loop around the edge of the hole. The easiest way of doing to define the features is to use the **Mark Free Edges** option when computing feature curves, although they can be added manually if preferred. Only include the feature curves relating to the hole as all feature curves are used for hole filling, not just those features that are defined on a free edge. Use the non-contiguous split option to separate the feature curve sets beforehand if necessary or the edit tools to clean up the definition.

An example of a simple hole with the appropriate feature curve set is shown below:

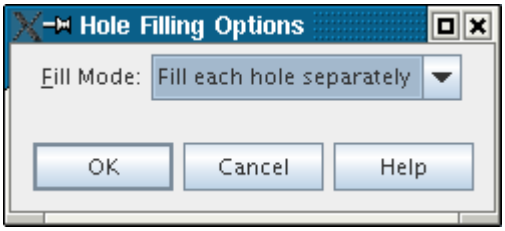


To fill a hole:

1. Select the feature curve node name and right-click. In the pop-up menu select the **Fill All Holes** option:



The *Hole Filling Options* dialog appears:

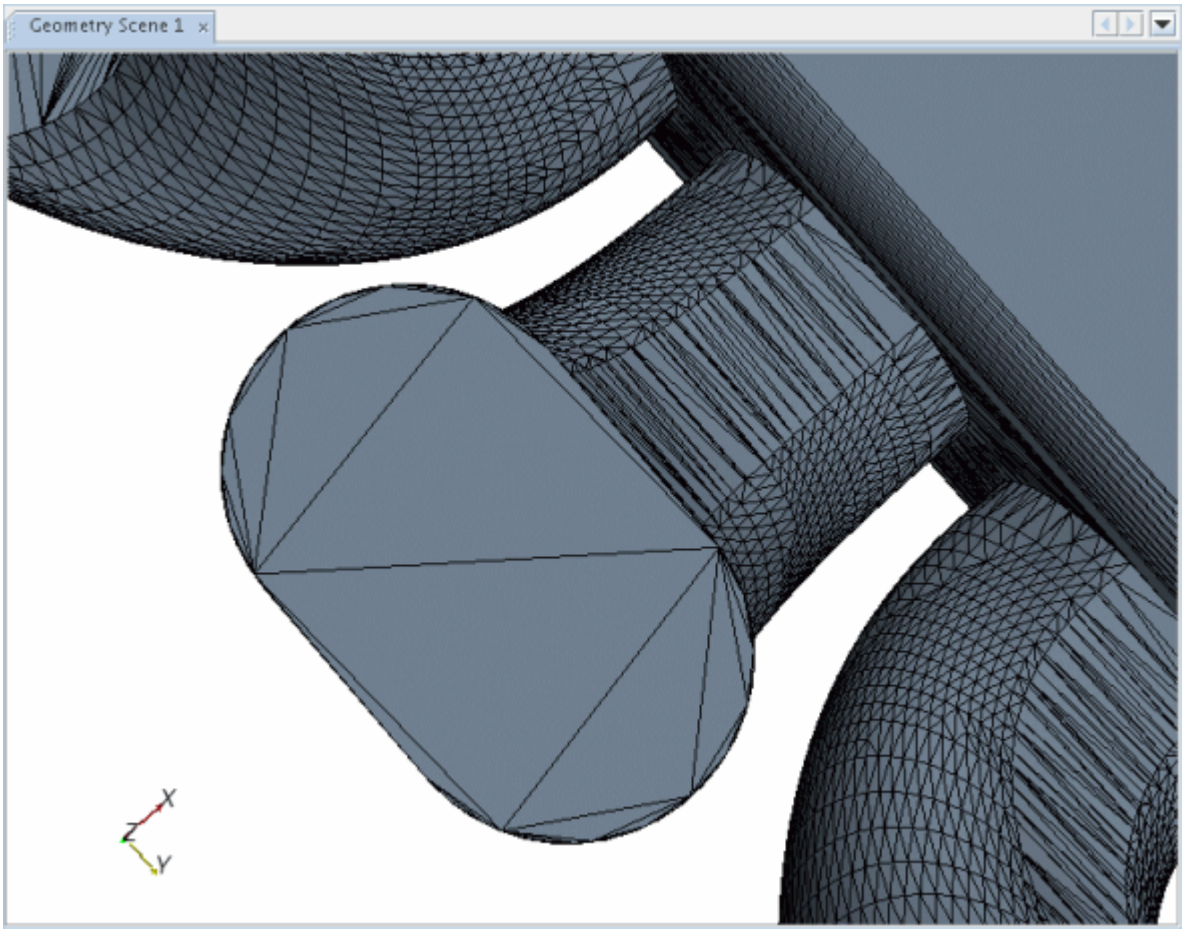


2. The **Fill Mode** option determines the method that is used to fill multiple holes.

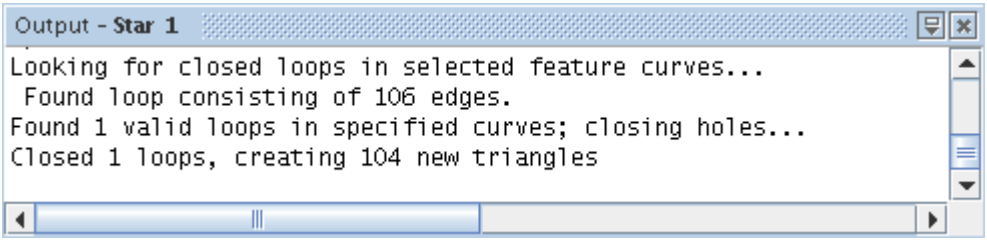
   Two options are provided:

   - **Fill each hole separately** - Treats each connected loop of free edges as a separate (simple) hole.
   - **Fill all holes together** - Looks at all the loops that are selected and fills in the void between them.

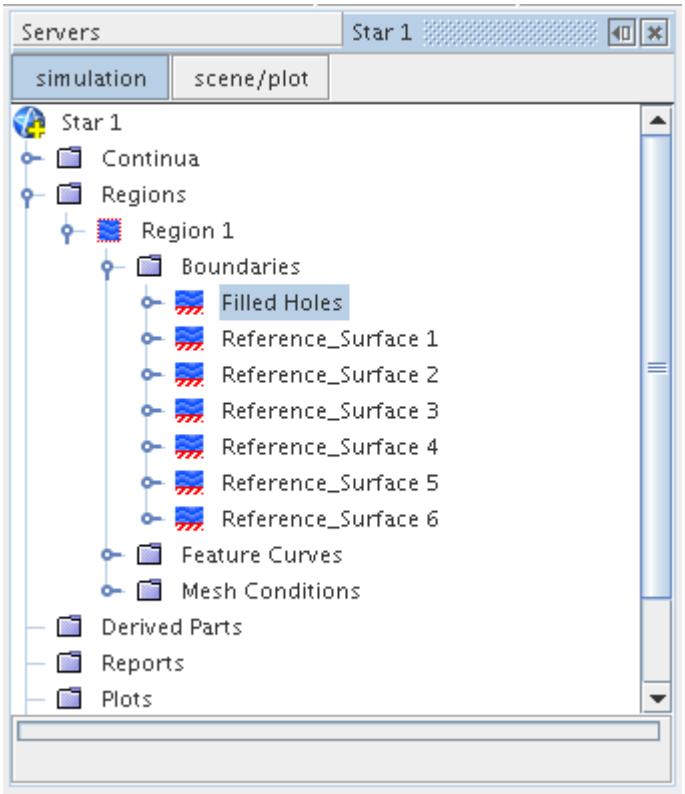
   The default operation is to treat each loop as a separate hole and fill it without reference to any other loops that have been selected. Each hole is filled with a planar surface using the least number of triangular faces possible. An example result of filling a simple hole is shown here.



The output for a successful simple hole fill is shown here:

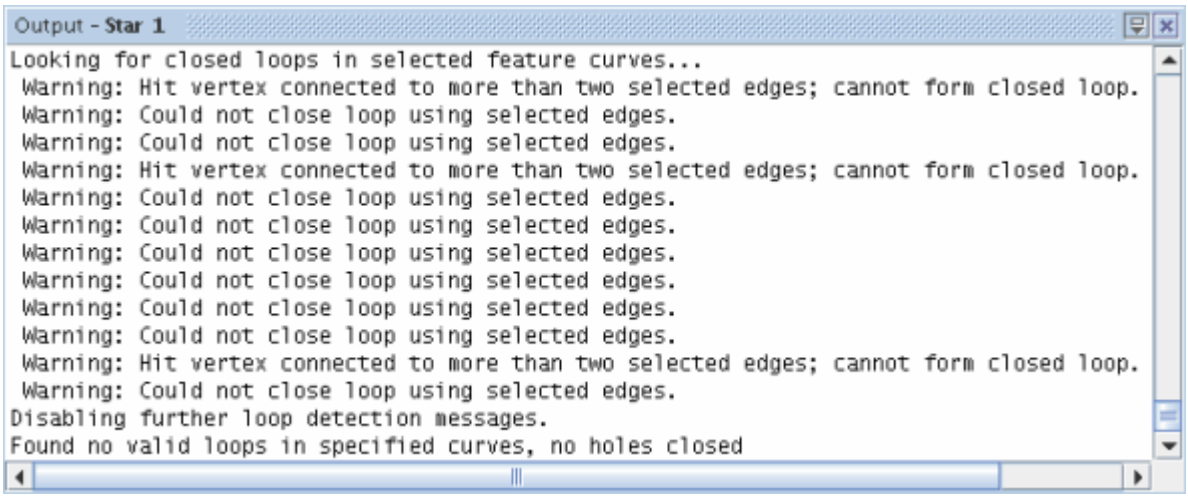


A new boundary representing the filled hole is also created in the simulation tree and a new displayer containing the object is added to the current scene. The default name of the new boundary is **Filled Holes**.



The use of the hole filler usually results in poor quality triangles being produced in the area of the gap. For this reason, it is recommended that you apply the manual surface remesher to the new boundary faces in order to improve the triangulation quality before generating a volume mesh.

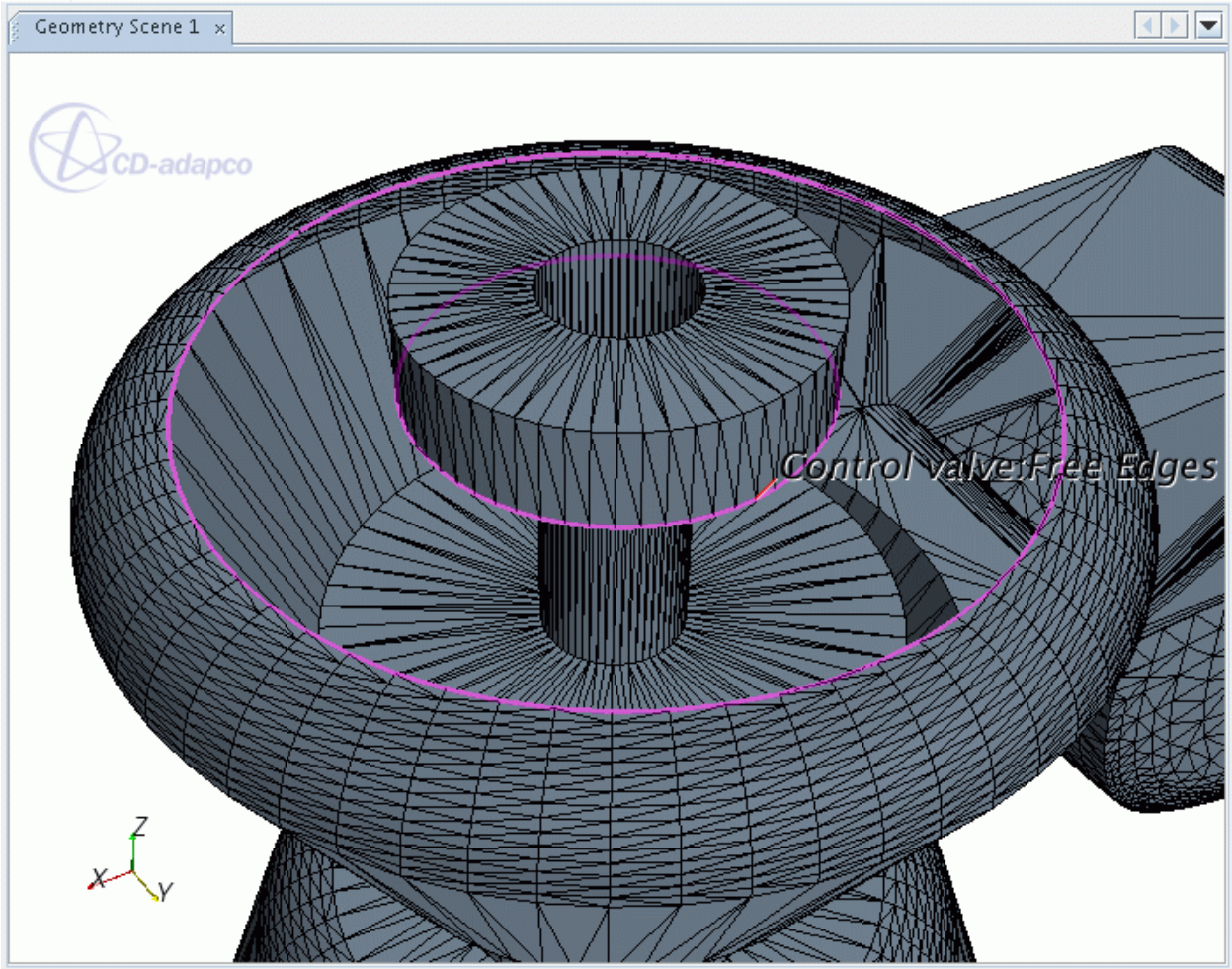
In instances when the feature curves do not form a closed loop or have extra feature curves intersecting the loop, the hole filler fails and no surface is produced. An example output showing when the hole filler is unsuccessful is shown here:



3. In this instance, use the manual add and edit tools to clean up the feature curve definition so that a closed loop is created and then run the hole filler again.

4. To fill more complex holes (for example, holes that have a loop within a loop), first ensure that a connected loop of feature edges exists for each side of the hole.

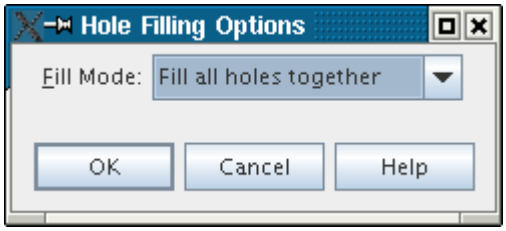
   An example is shown below:



   Each loop of feature edges does not need to belong to the same node.

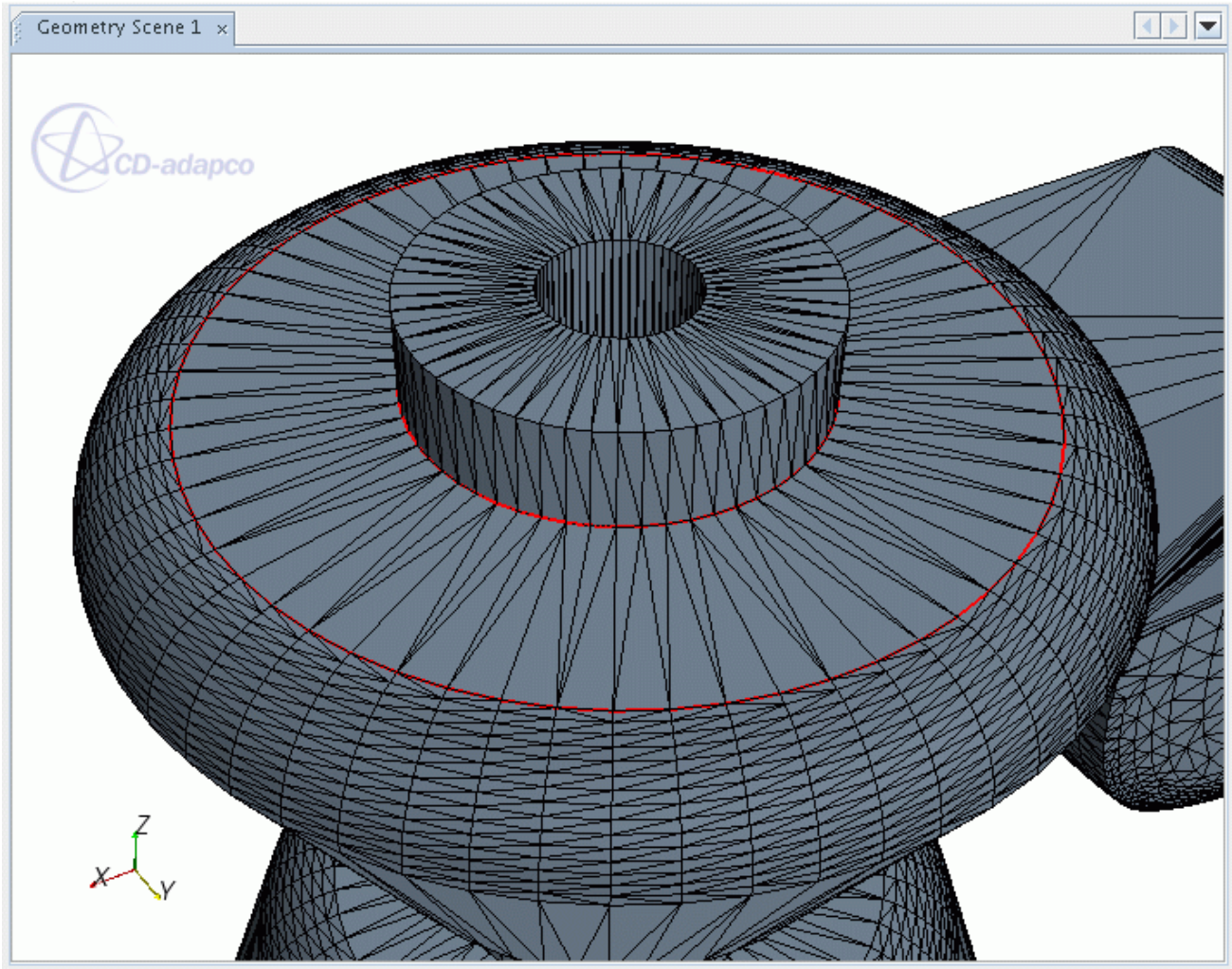
5. To fill the hole, select the feature curve nodes and right-click. In the context menu select the **Fill All Holes** option.

   The *Hole Filling Options* dialog appears:



6. In the dialog, select **Fill all holes together** for the **Fill Mode**, so that the gap between the loops is filled rather than each loop separately (which would result in overlapping surfaces).

An example result of filling a multi-loop hole is shown below:



As for simple holes, a new boundary node that is called **Filled Holes** is created containing the surface.

Zipping Edges

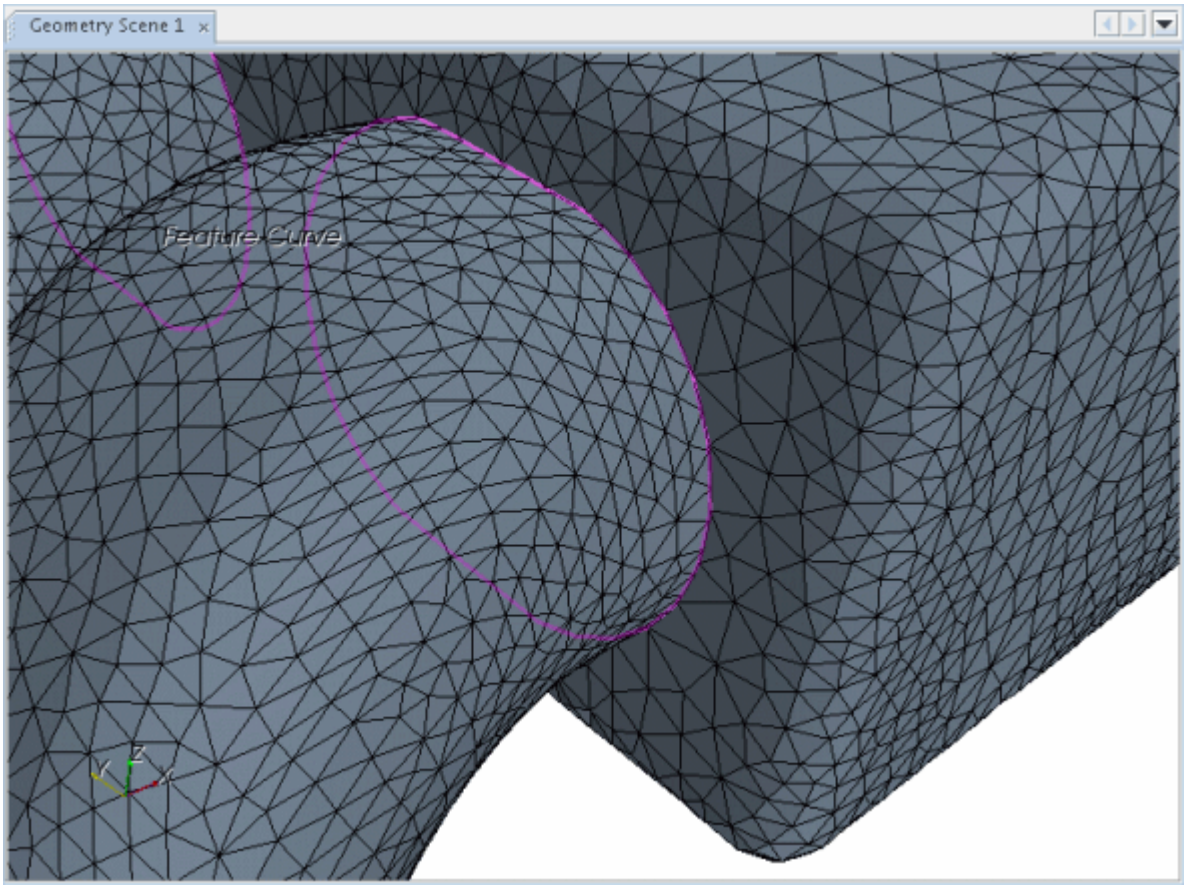
The edge zipper can be used to join mismatched surfaces together to form one continuous surface.

It does this by retriangulating the faces at the edges and adding new vertices so that a one to one connection is obtained.

Before using the edge zipper, make sure that feature curves are defined along both sets of edges that you wish to join. The easiest way of marking the edges is by using the **Mark Free Edges** option when computing feature curves, although they can be added manually if preferred. The feature curves used for the zipping process can belong to the same node or to different nodes, but have to belong to the same region. Extra

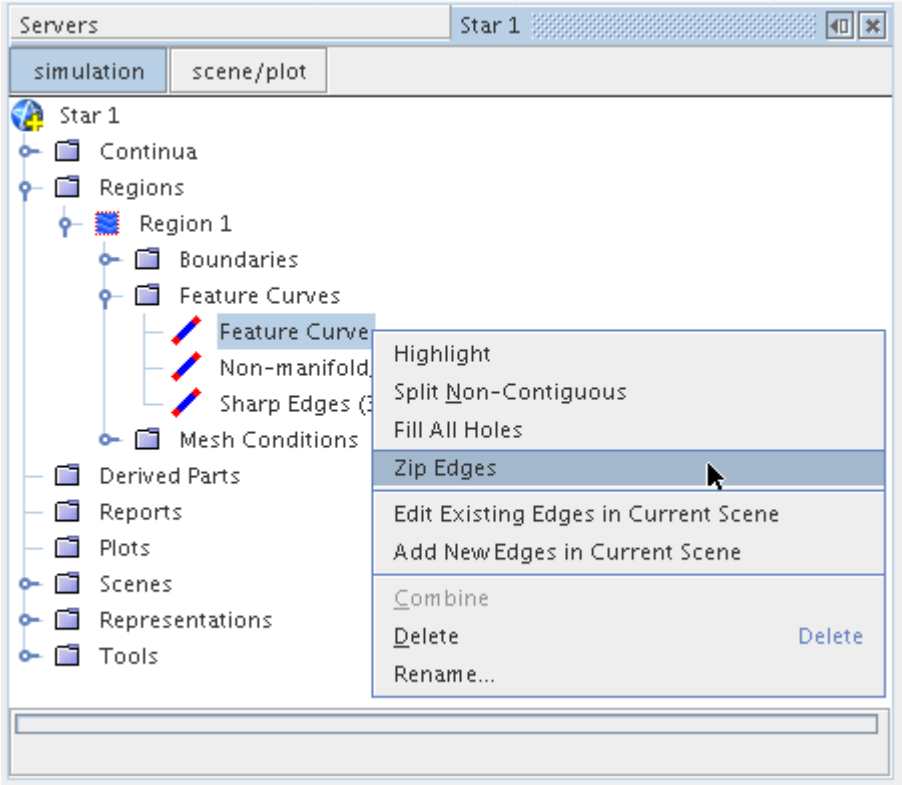
feature curves not belonging to the free edges can also be included but they have no effect on the zipping process.

An example geometry with mismatched surfaces and the appropriate feature curve set is shown below:

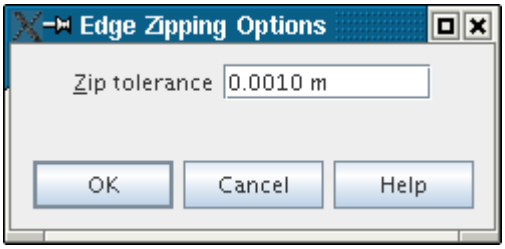


To run the edge zipper:

1. Select the feature curve node name or names (using multiple selection) and right-click. In the pop-up menu select the **Zip Edges** option:



The *Edge Zipping Options* dialog appears:



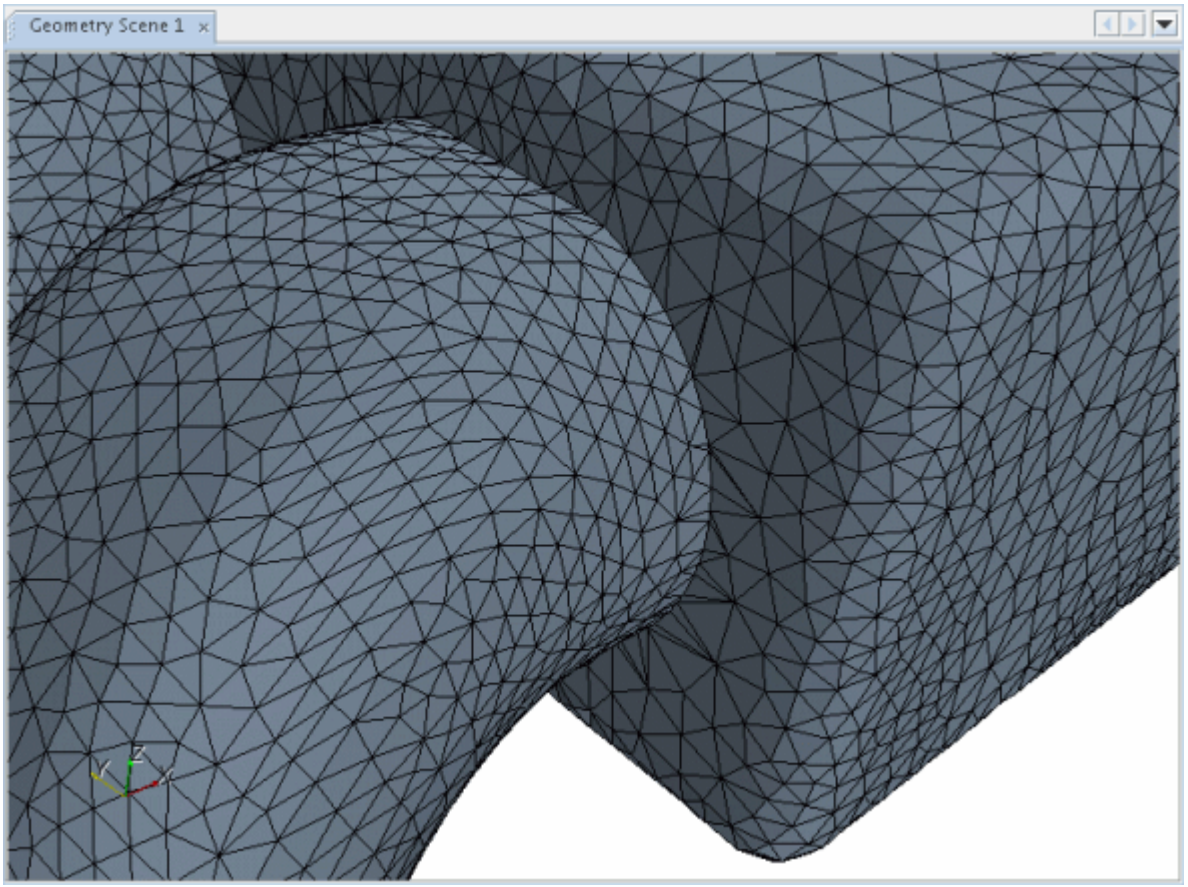
The **Zip tolerance** can be set to determine the tolerance for locating vertices to form the join.

2. To perform the edge zipping operation, press the **OK** button.

   If new faces are required in order to zip the two surfaces, then they are added to the appropriate boundary node in the surface.

   Feature curves can also be deleted depending on the zipping triangulation.

   An example of a zipped surface is shown here.



The output from a successful edge zipping operation appears as follows:

```
Output - Star 1
Zipping edges in selected feature curves...
Found 255 free edges in specified curves; zipping edges...
Deleted 7 edges. 248 edges remain.
```

The edge zipper often produces poor quality triangles being along the length of the surface mismatch. For this reason, it is recommended that you apply the surface remesher to the closed surface in order to improve the triangulation quality before generating a volume mesh.

Volumetric Controls

You use a volumetric control to specify the mesh density in a specific zone for both surface and volume meshes.

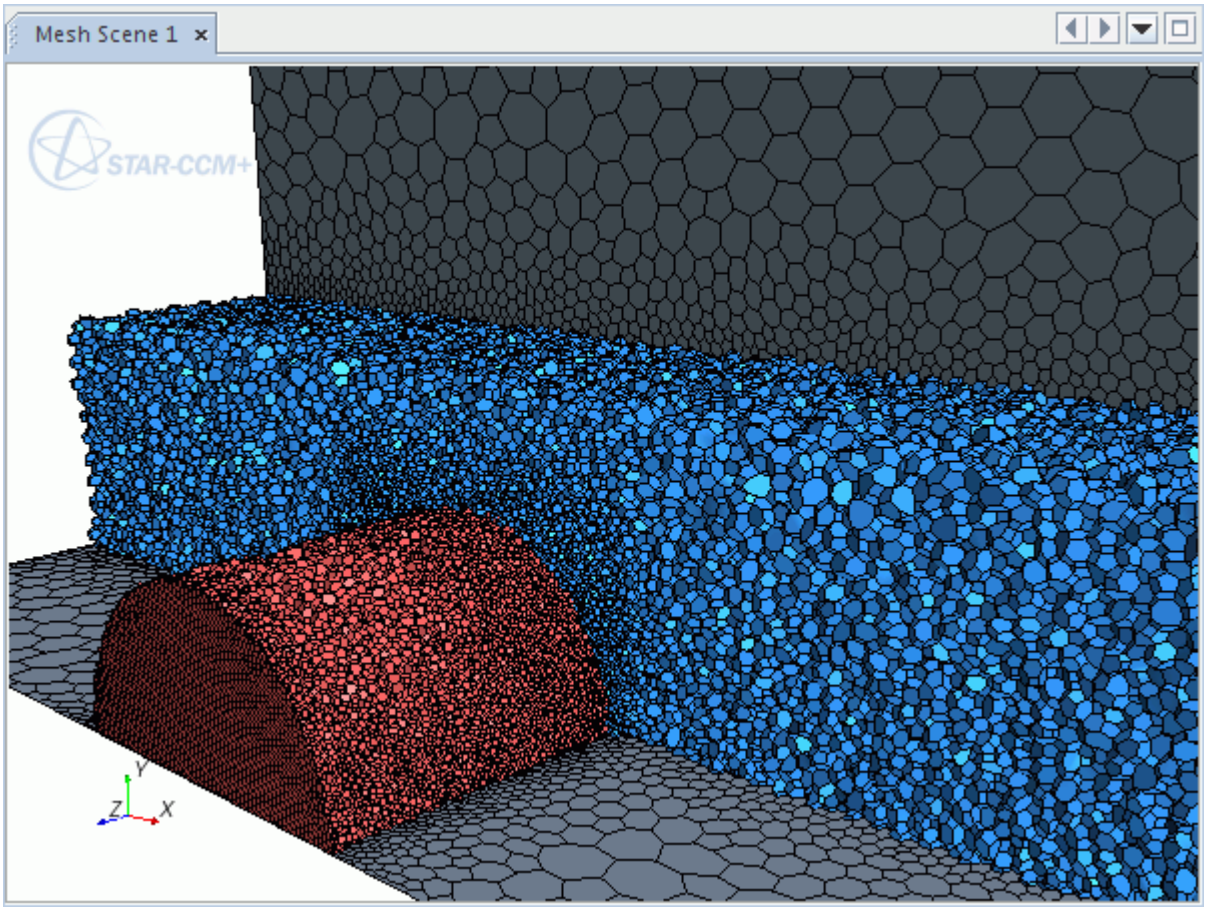
You can define the mesh refinement zone using volume shapes and geometry parts. For geometry parts, the control takes its definition from the Root description only.

You can apply each volumetric control to any combination of meshing models. Therefore, you can set specific cell sizes within the zone for each mesh generation stage.

Volumetric controls can overlap and extend outside the region boundary definition. Volumetric controls can also overlap from one region to another, but the effect is only included if the region belongs to the same mesh continuum as the volumetric control. If two or more volumetric controls overlap, the smallest user-defined cell size takes priority.

The relevant growth rate for each core mesh model determines the transition in cell sizes from the volumetric control to the core mesh.

The following image illustrates volumetric controls. The blue cells are within a block-shaped volumetric control which specifies a cell size that is 25% of the base size. The red cells are within a cylindrical volumetric control with a refinement to 10% of the base size. The image also shows how growth rates influence cell size transition.



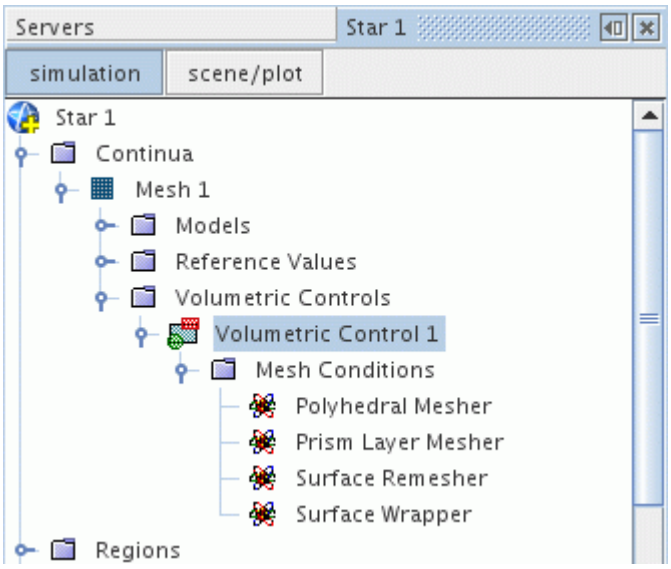
Note: Do not use the same geometry part for the volumetric control that you use as the bounding geometry for the meshing continuum. To make the volumetric control influence the boundaries, extend it a small distance beyond the original geometry.

Volumetric controls affect each meshing model in a different way:

- Surface Wrapper — Refines the surface size. If you use the wrapper scale factor, the surface size within the volumetric control is also scaled.

- Surface Remesher — Refines the surface size.
- Polyhedral Mesher — Polyhedral cell size is uniform within the volumetric control.
- Trimmed Mesher — Refines with either an isotropic or anisotropic cell size. If you specify both options, the smallest cell size in each coordinate direction takes priority.
- Tetrahedral Mesher — Tetrahedral cell size is uniform within the volumetric control.
- Prism Layer Mesher — Refines the number of prism layers, prism layer stretching, and prism layer thickness. To include the prism layer, the extent of the volumetric control must exceed the boundary.
- Advancing Layer Mesher — Refines the polyhedral core cell size and the prismatic layer cell size. However, you cannot refine the number of layers, stretching factor or layer thickness.

The **Volumetric Controls** manager node, which has its own context menu, contains the volume control nodes for the mesh continuum.



Contents:

Examples of Volumetric Control Refinement
Effect of Volumetric Control on Surface and Volume Meshers
Setting Up a Volumetric Control
Volumetric Control Manager Context Menu
Working with Volume Shapes

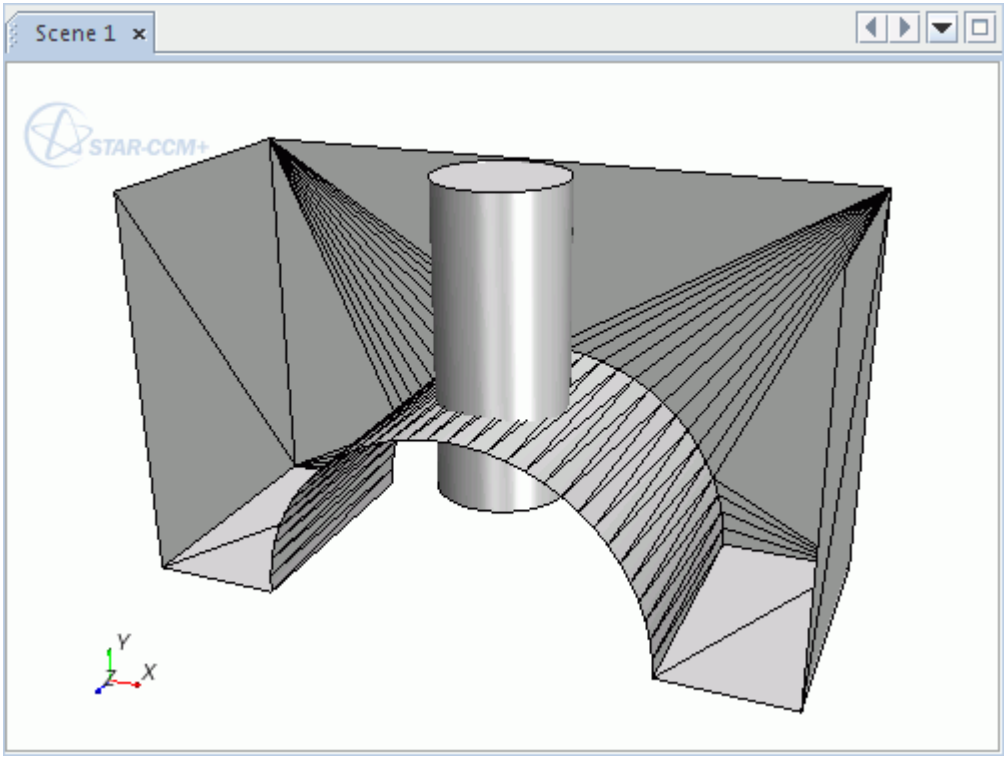
See Also:

Generalized Cylinder Mesher

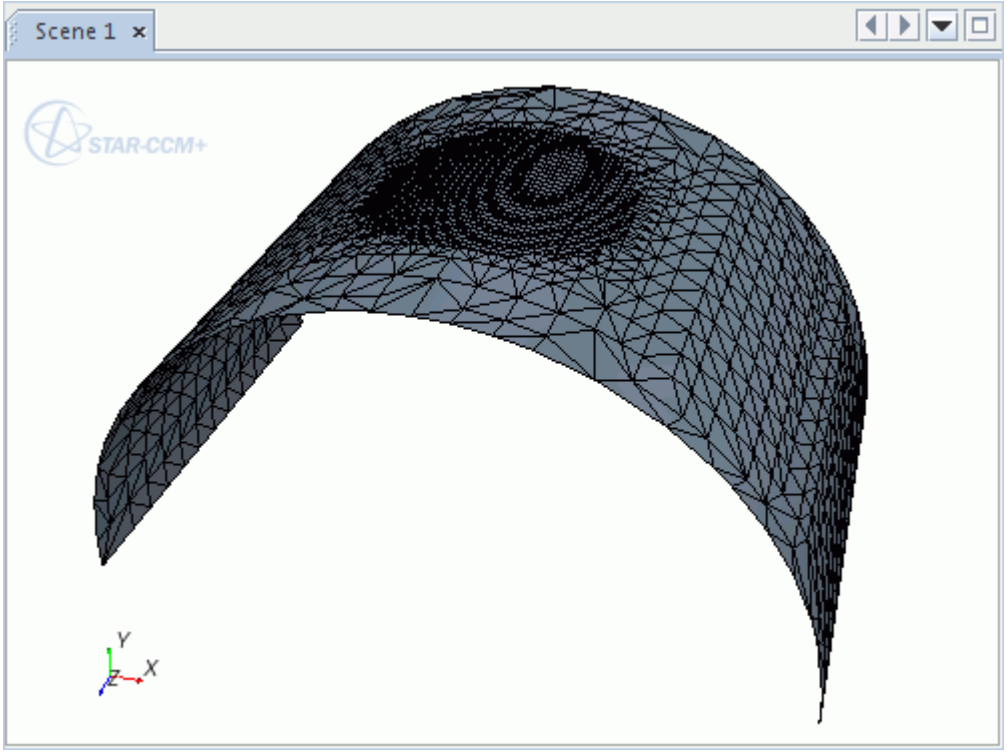
Examples of Volumetric Control Refinement

The following examples show 30% mesh refinement with a cylindrical volumetric control.

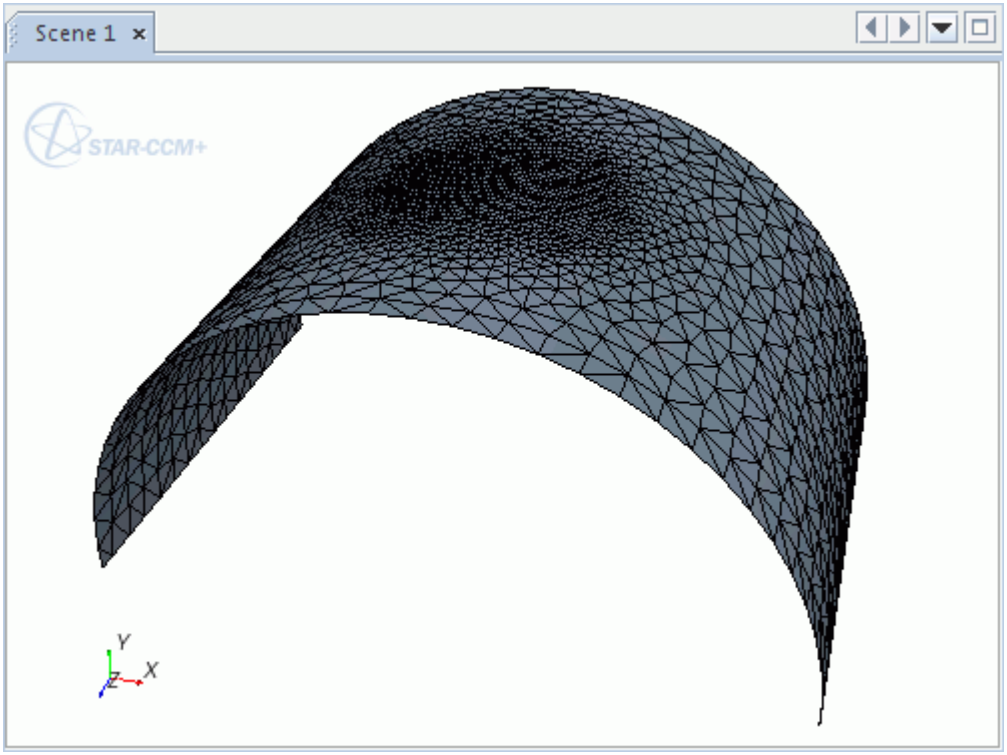
The image below shows the initial geometry and the cylindrical volumetric control.



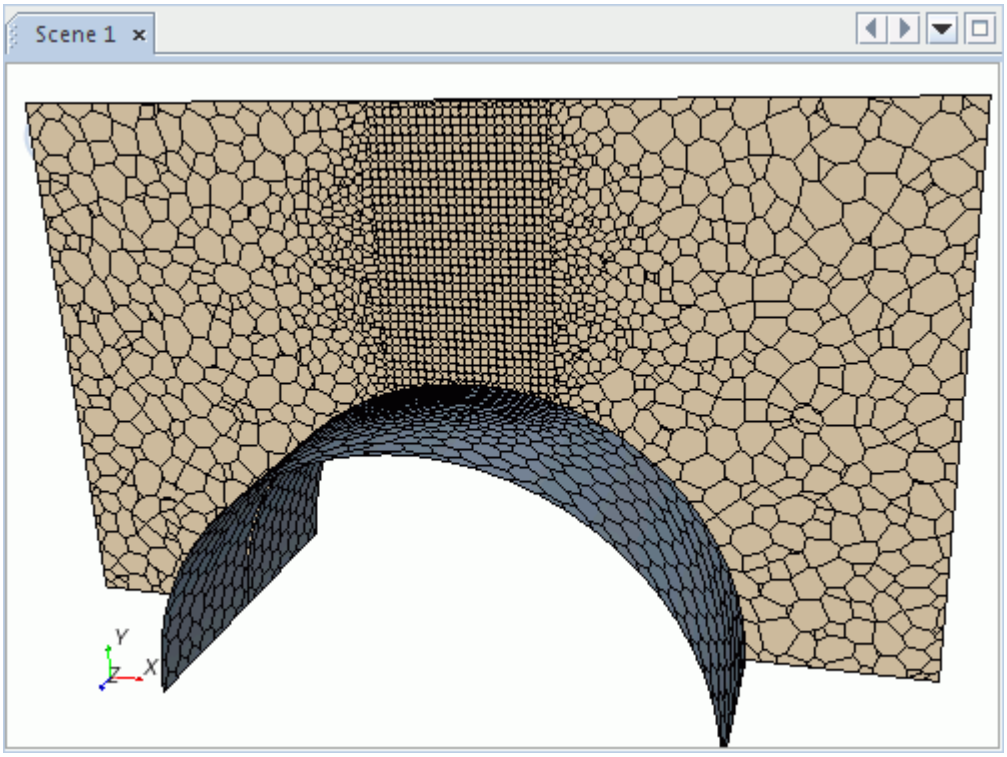
Surface Wrapper



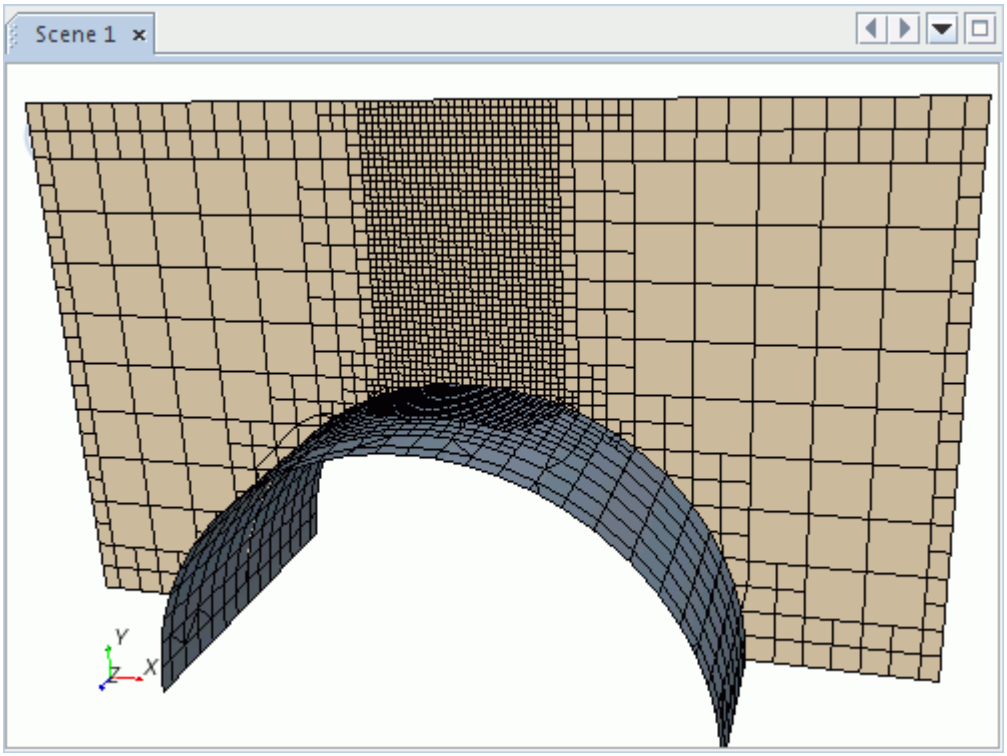
Surface Remesher



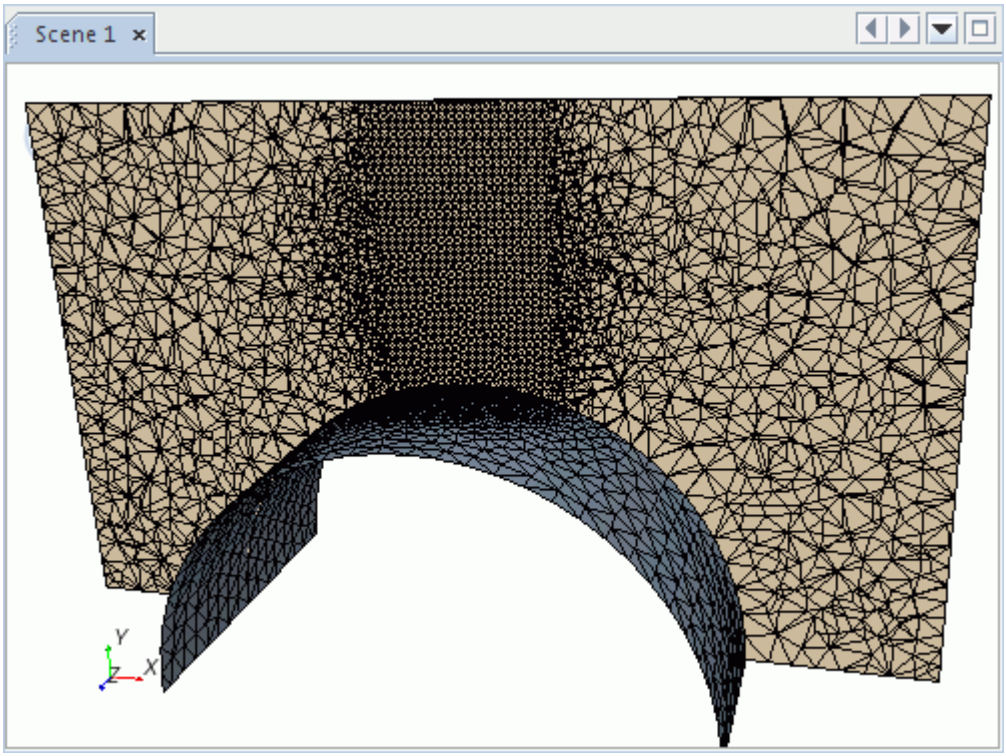
Polyhedral Mesher



Trimmed Mesher

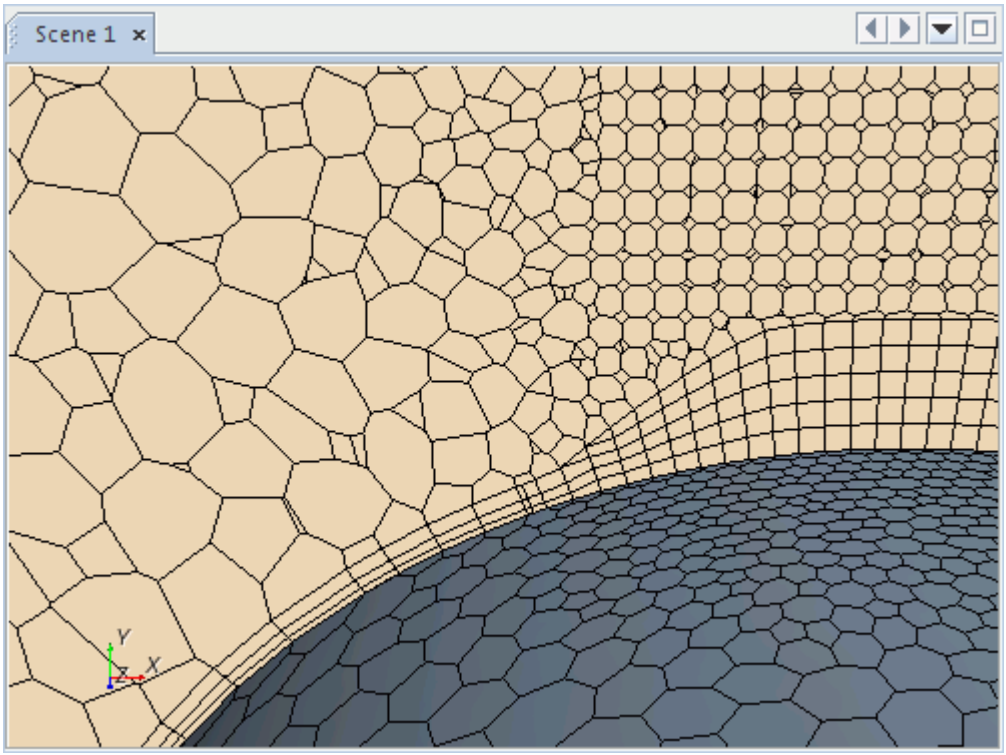


Tetrahedral Mesher



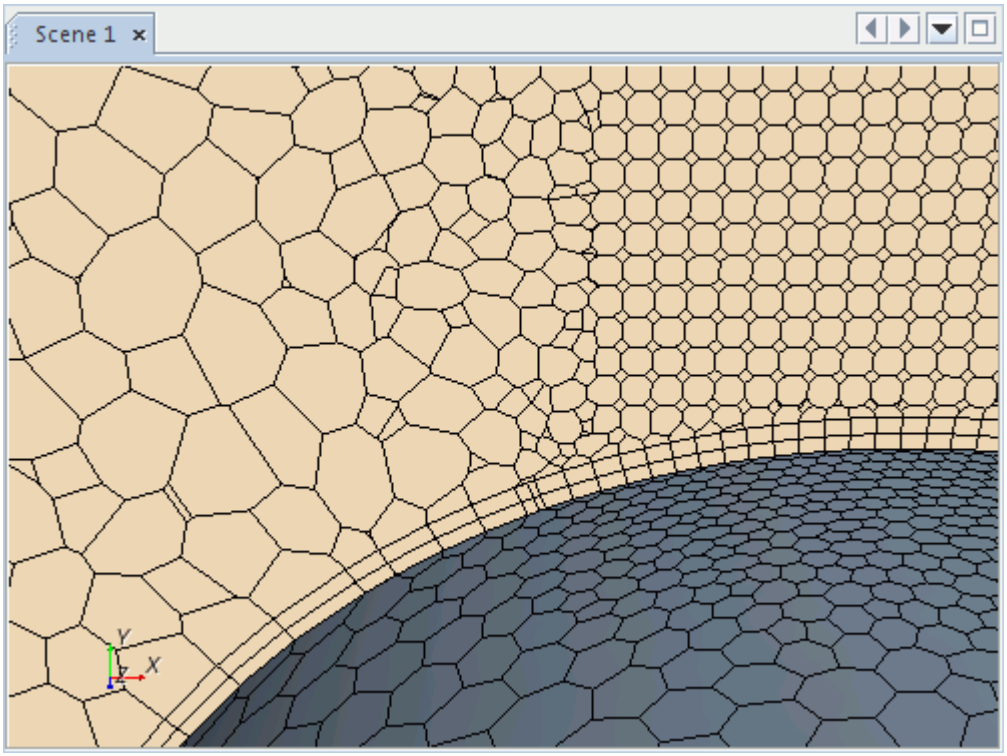
Prism Layer Mesher

This example is slightly different from the others in that the number of layers and prism layer thickness changes within the volumetric control.



Advancing Layer Mesher

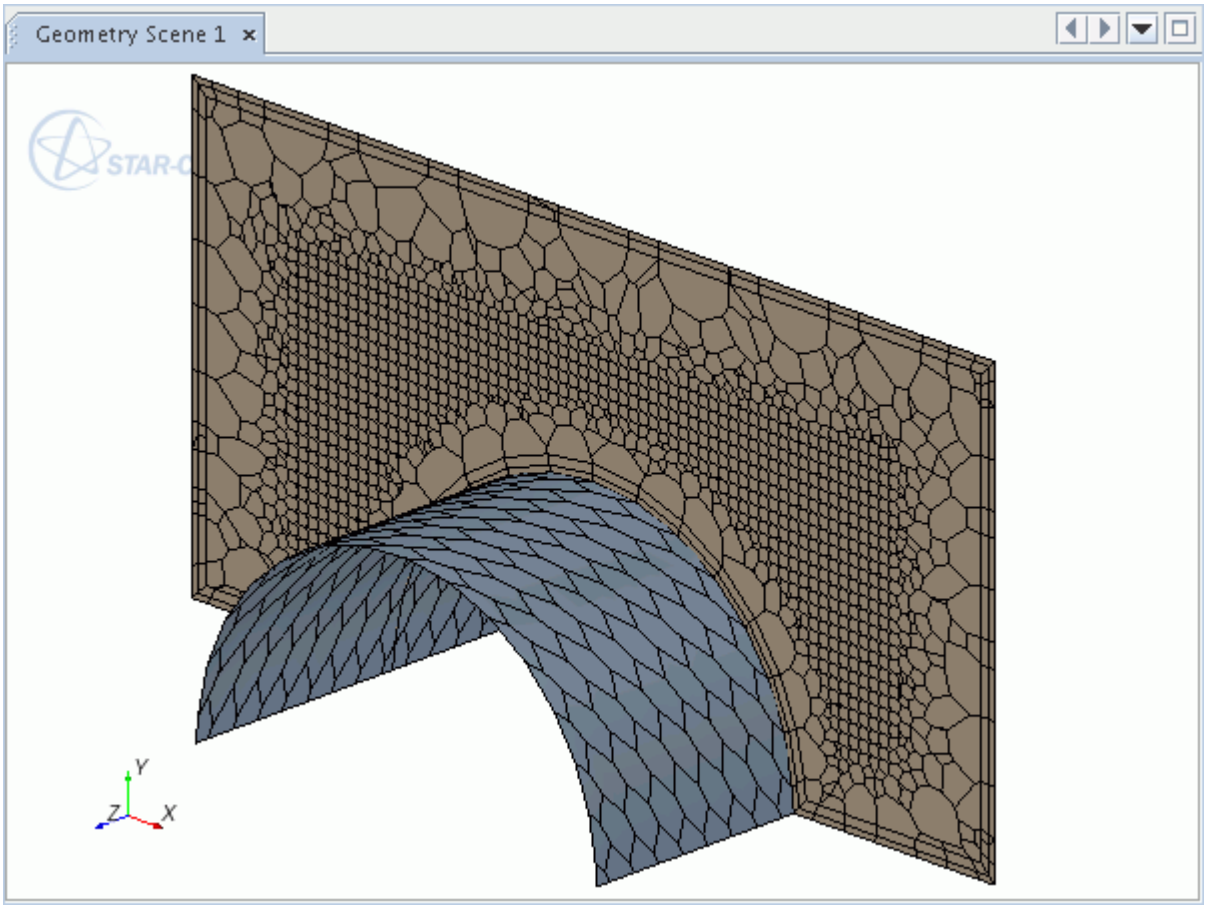
With the Advancing Layer Mesher, a volumetric control can only change the cell size. The number of layers and layer thickness are not available.



Effect of Volumetric Control on Surface and Volume Meshers

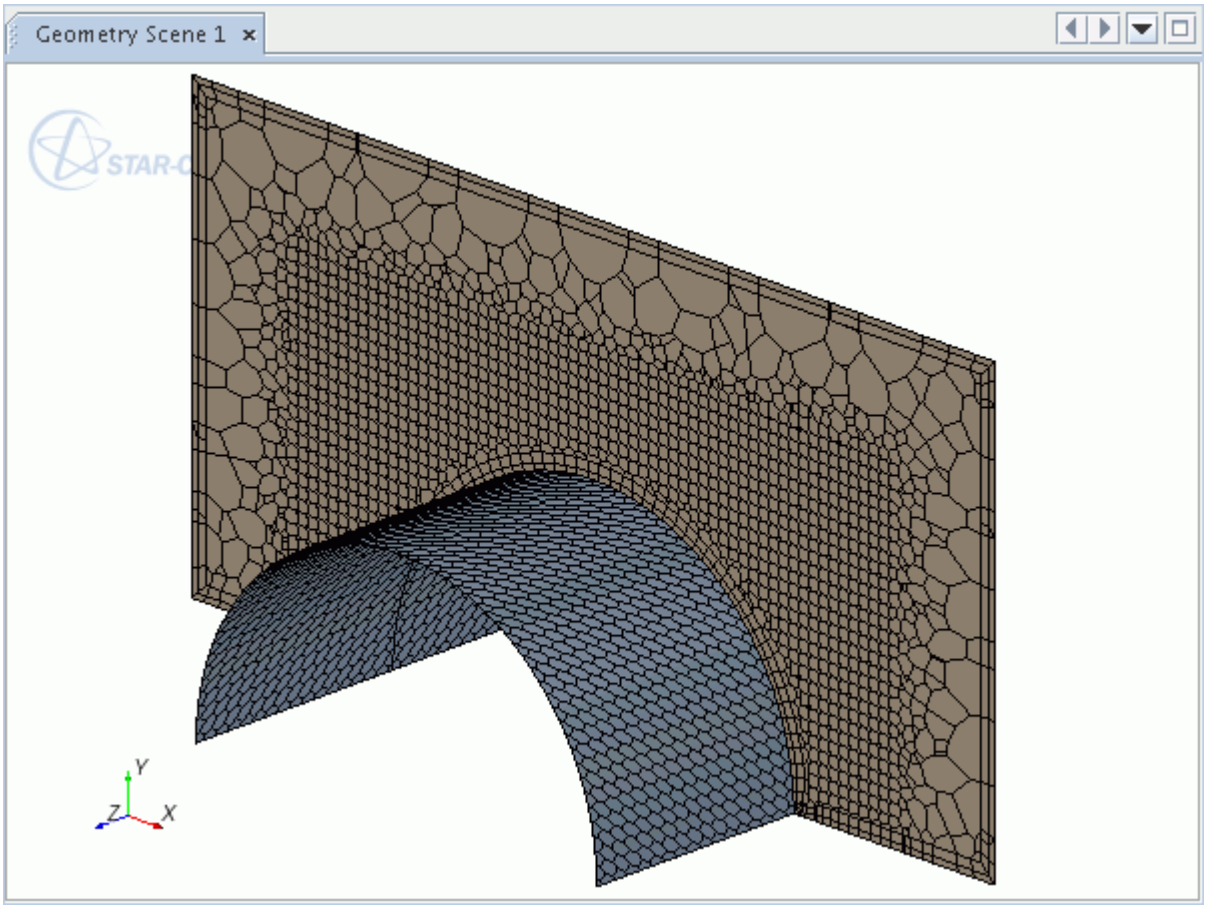
The volumetric control is activated individually for each meshing model. This functionality can cause unexpected results if the volumetric control is activated on the volume mesher, but not the surface mesher.

For example, the following image shows the resultant mesh when a volumetric control is activated on the polyhedral mesher, but not the surface remesher.



In this case, the volume mesh is coarse near the surface, as the volumetric control does not refine the surface mesh.

Activating the volumetric control for both the polyhedral mesher and the surface remesher generates the following mesh.



This image shows that mesh refinement is observed up to and including the surface mesh.

Setting Up a Volumetric Control

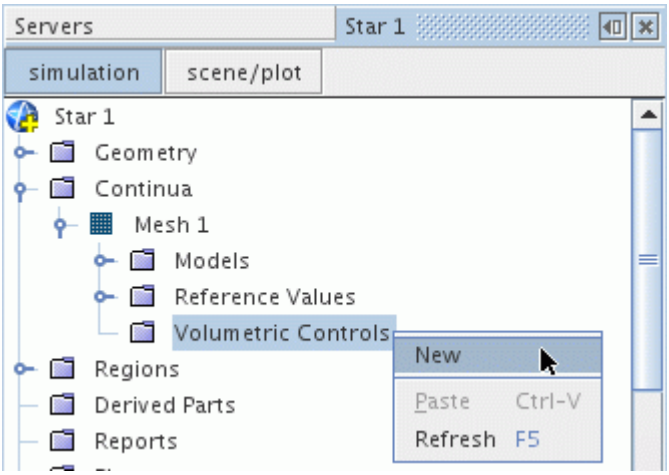
A volumetric control refines cells within a user-defined volume shape or geometry part. You can set this refinement using either an absolute cell size or relative to the mesh base size.

You can create a volumetric control after importing a surface mesh as a region or creating a mesh continuum.

Note: A volumetric control cannot coarsen a mesh — that is, the smallest size takes precedence. Therefore, for the volumetric control to refine the mesh, the size that you specify must be smaller than the corresponding reference or boundary surface size.

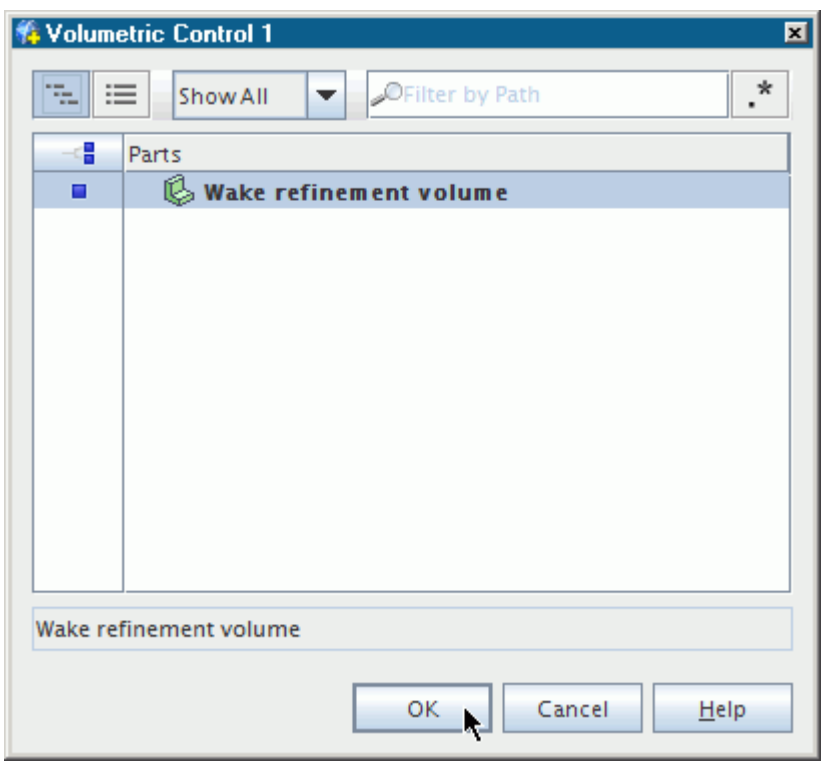
To refine with a volumetric control:

1. Right-click the **Volumetric Controls** node and select **New**.

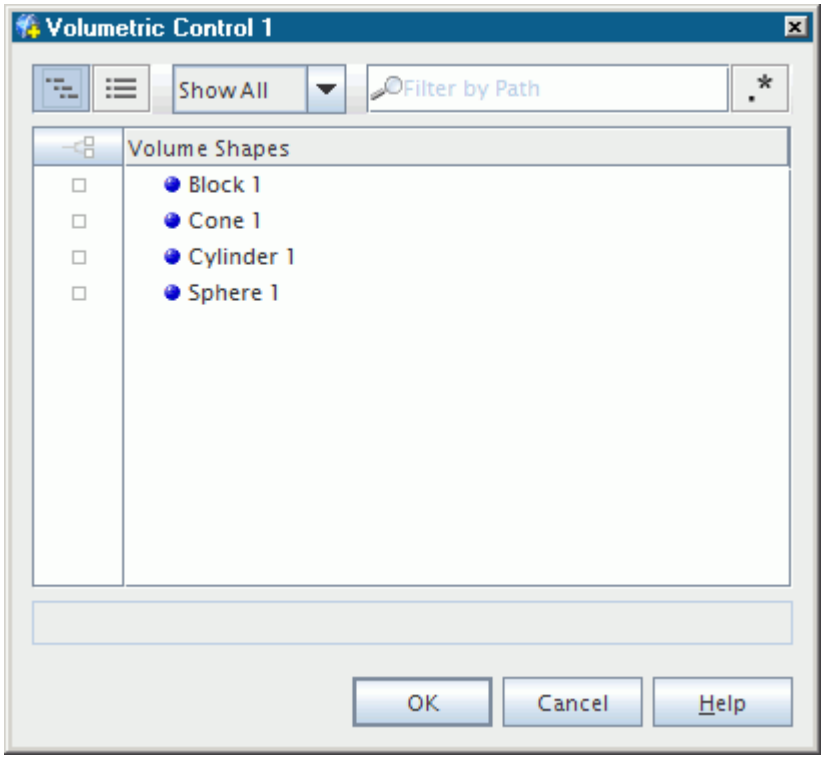


Initially, the volumetric control node is inactive , which means it does not have a geometry part or volume shape that is assigned to it.

2. To assign a geometry part, click the *Parts* property and select the geometry part from the object selection dialog.

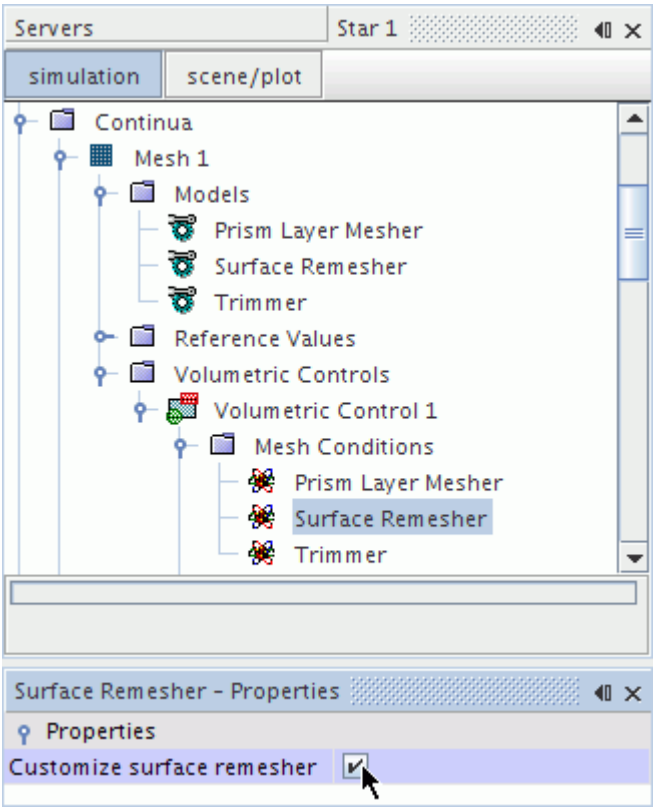


3. To assign a volume shape, click the *Shapes* property and select a previously created volume shape from the dialog.



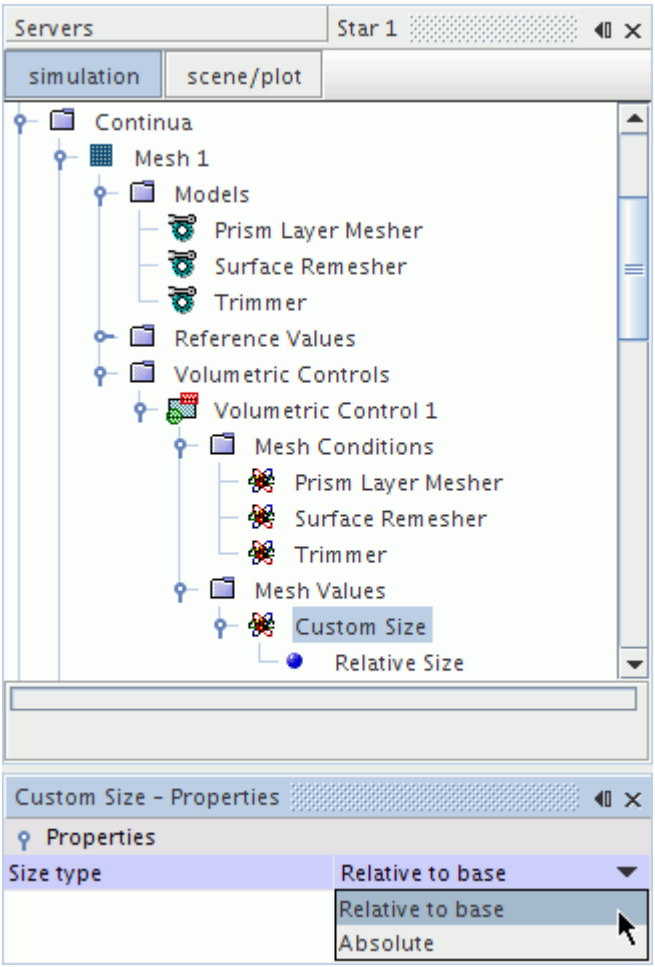
4. Under the **Volumetric Control** > **Mesh Conditions** node, select the required mesh models that you want the volumetric control to influence.

   The available meshers depend on the meshing models that you have selected. In this example, only the Surface Remesher, Trimmer, and Prism Layer Mesher are available.

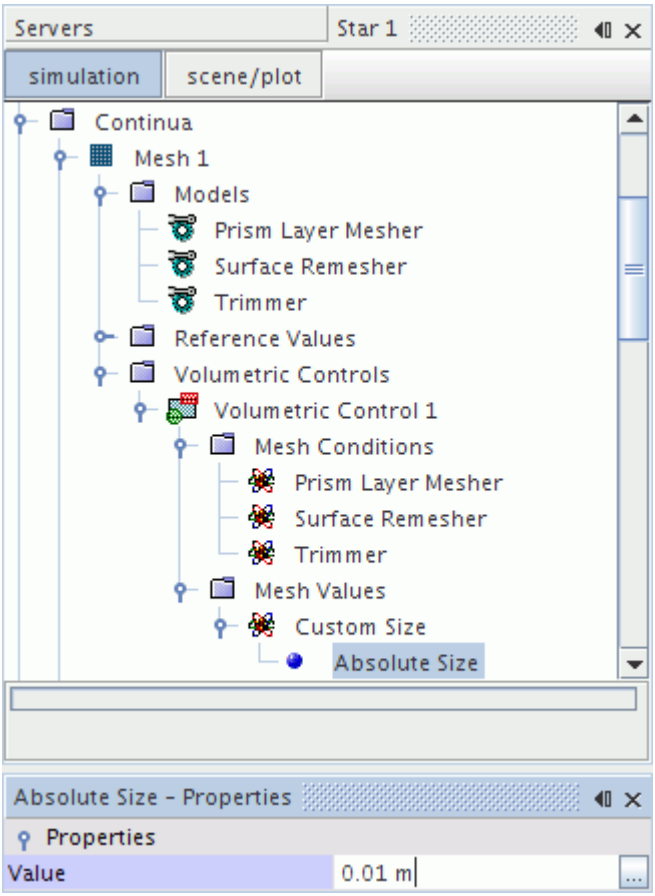


This action adds the **Custom Size** node under the **Mesh Values** manager.

5. Select the **Mesh Values** > **Custom Size** node and define whether the refinement is relative or absolute.



6. Select the **Relative Size** or **Absolute Size** node and specify the level of refinement in the *Properties* window.



Contents:

Setting the Trimmer Isotropic Size
Setting Trimmer Anisotropic Sizes
Setting Prism Layer Volumetric Controls

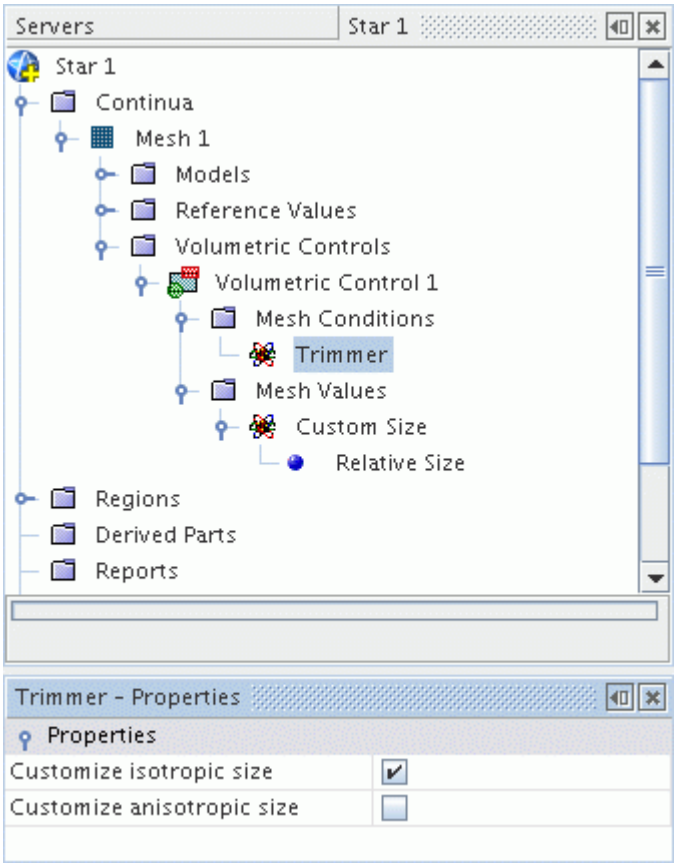
Setting the Trimmer Isotropic Size

The trimmer isotropic size is equivalent to the cell size for other core mesh models. It specifies the same cell size in each coordinate direction.

Note: The surface cell size is subject to the trimmer mesher surface size restriction. Therefore, the core mesh cell size can change inside the volumetric control as it approaches the surface.

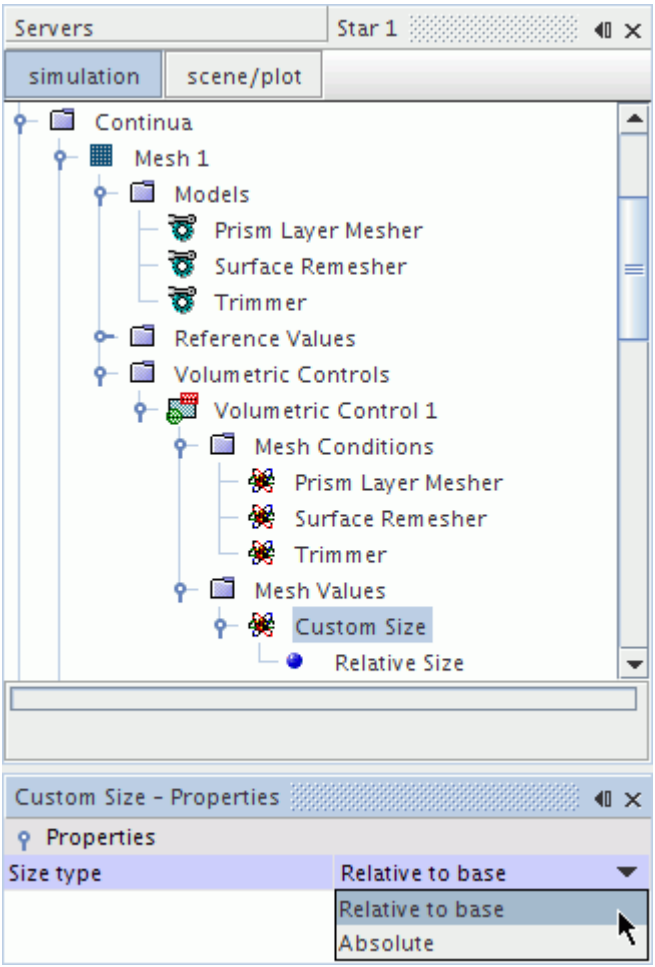
To specify an isotropic cell size:

1. Select the **Volumetric Controls** > **Mesh Conditions** > **Trimmer** node and activate the *Customize isotopic size* property.

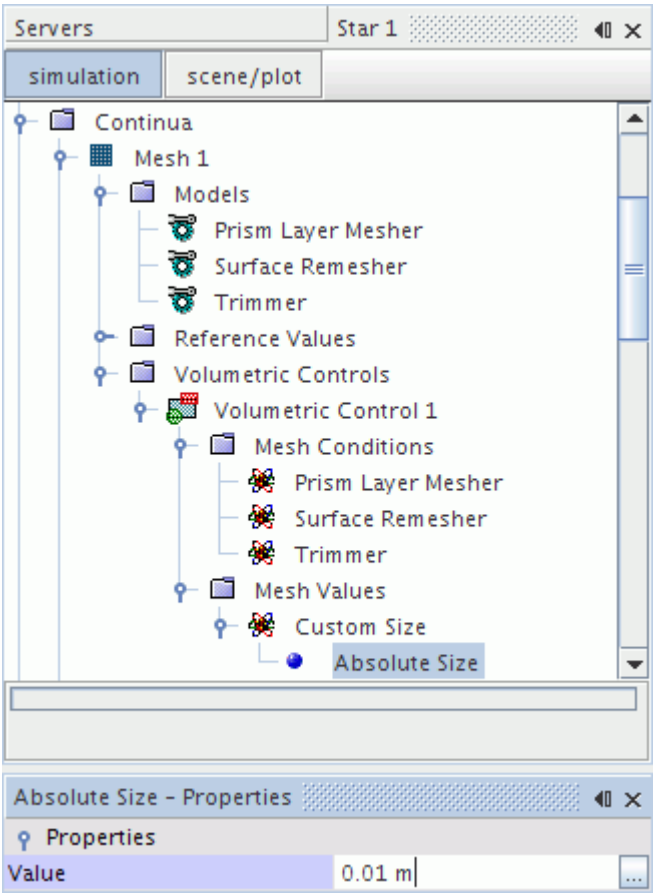


This action adds the **Custom Size** node under the **Mesh Values** manager.

2. Select the **Mesh Values** > **Custom Size** node and define whether the refinement is relative or absolute.



3. Select the **Relative Size** or **Absolute Size** node and specify the level of refinement in the *Properties* window.



Setting Trimmer Anisotropic Sizes

The trimmer anisotropic cell sizes allow you to specify a different cell size in each coordinate direction.

The coordinate directions are relative to the trimmer mesher coordinate system.

If you do not specify an anisotropic size for a coordinate direction, the volumetric control uses the isotropic size. If you also do not specify an isotropic cell size, the volumetric control uses the global reference values.

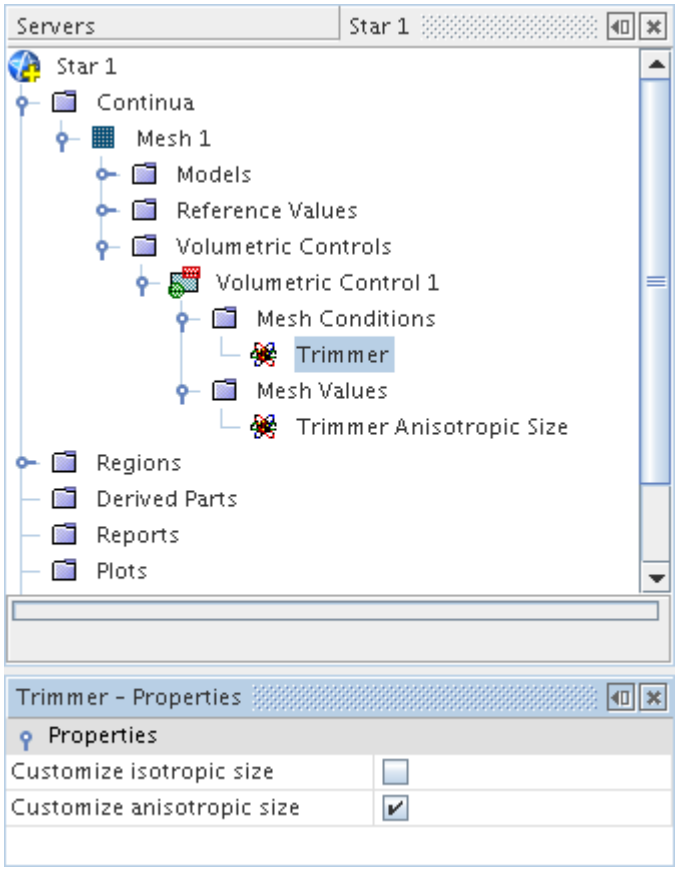
The trimmed mesher can only refine cells by splitting the original cell in half. Therefore, the volumetric control refines the cells to the next lowest allowable cell size. For example, if the core mesh cell size is 8 units, then the cells within the volumetric control can only be split to sizes of 4, 2, and 1 units. So if you specified a cell size of 3, you would actually get a cell size of 2 within the volumetric control.

Cells within an anisotropic volumetric control are not subject to the template growth rate settings. See Trimmed Mesher Properties. Anisotropic cells transition from small cells to large cells as fast as possible (equivalent to the fast growth rate setting).

If the boundary is aligned with the anisotropic coordinate directions, the volumetric control refines cells up to and including the boundary. However, if the surface is not aligned with the coordinate system, then anisotropic refinement reverts to isotropic refinement near the surface, as this behavior maintains good cell quality.

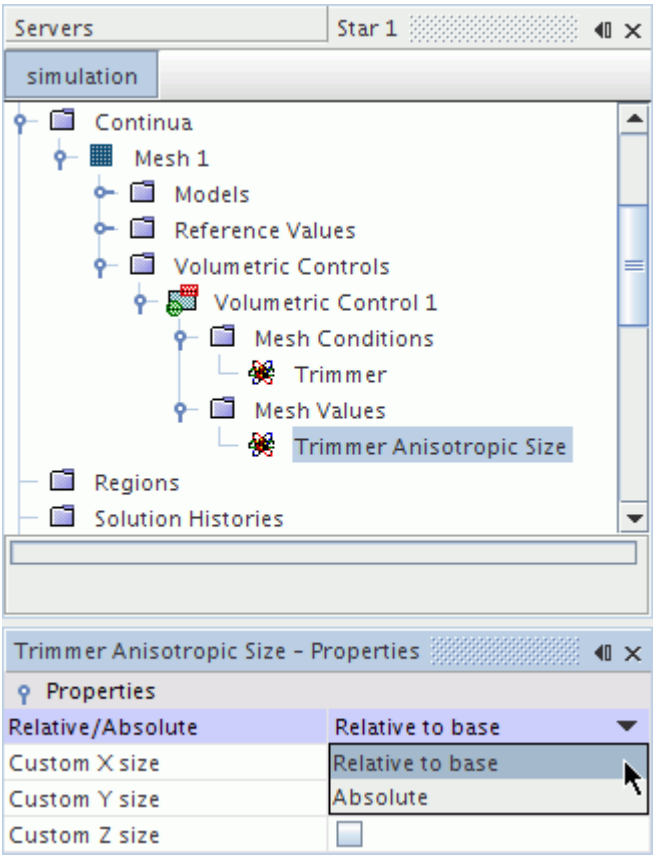
To specify anisotropic cell sizes:

1. Select the **Volumetric Control** > **Mesh Conditions** > **Trimmer** node and activate the *Customize anisotropic size* property.

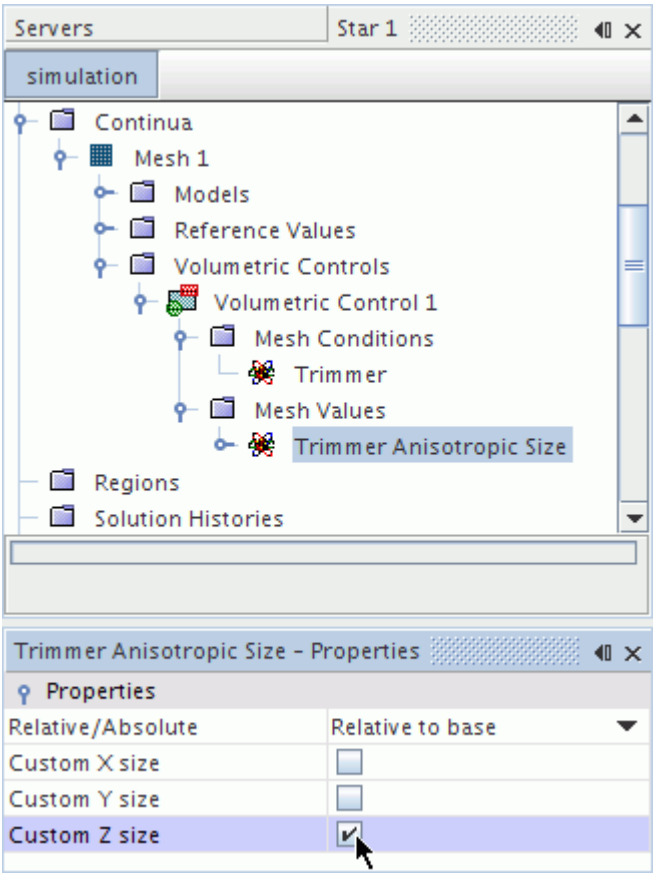


This action adds the **Trimmer Anisotropic Size** node under the **Mesh Values** manager.

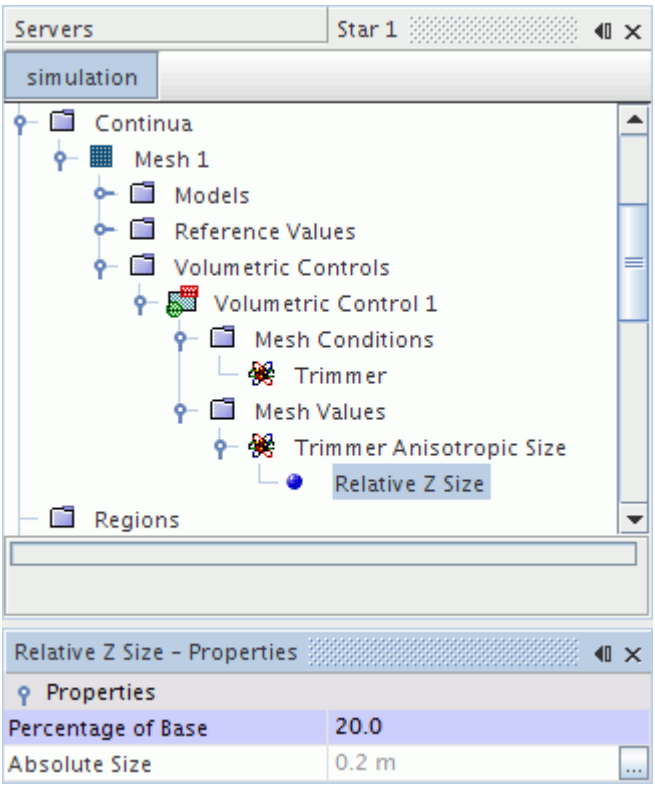
2. Select the **Mesh Values** > **Trimmer Anisotropic Size** node and define whether the refinement is relative or absolute.



3. Activate the following properties for cell sizes in the X-, Y-, and Z-direction respectively:
   - *Custom X size*
   - *Custom Y size*
   - *Custom Z size*

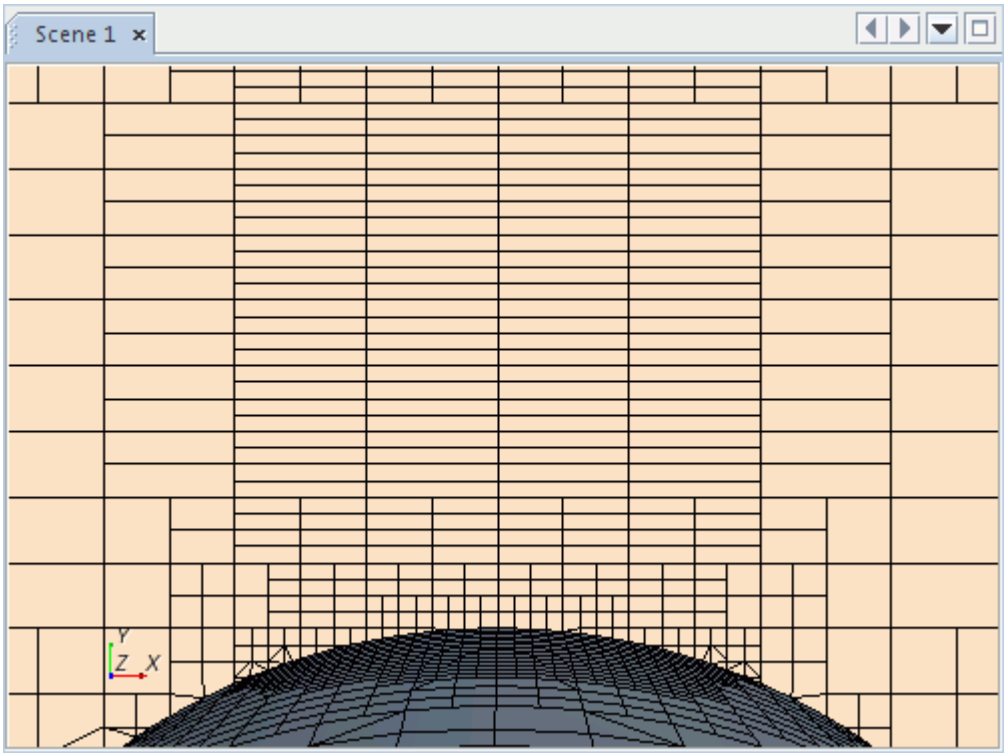


4. Select the sub-node and specify the level of refinement in the *Properties* window.



Example of Anisotropic Cell Sizes

The following example shows anisotropic refinement in the y-direction:

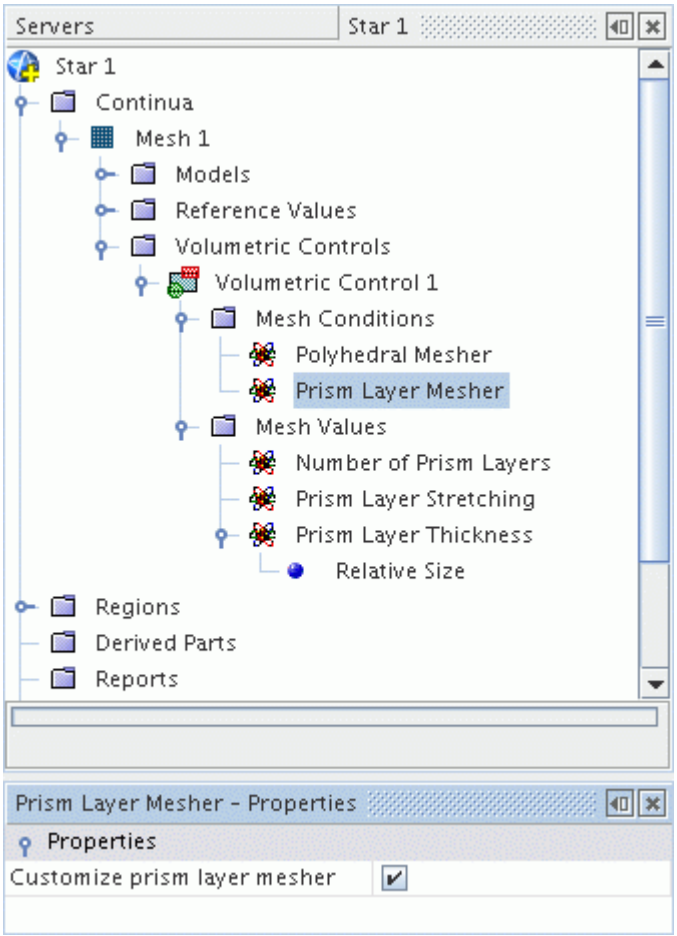


Setting Prism Layer Volumetric Controls

Applying a volumetric control to the prism layer model allows you to refine the number of prism layers, prism layer stretching, and prism layer thickness.

To specify prism layer volumetric controls:

1. Select the **Volumetric Control** > **Mesh Conditions** > **Prism Layer Mesher** node and activate the *Customize prism layer mesher* property.



2. Use the following nodes and their properties customize the prism layers within the volumetric control:
   - Number of Prism Layers
   - Prism Layer Stretching
   - Prism Layer Thickness

     For the prism layer thickness, you can use a relative or absolute value.

Volumetric Control Manager Context Menu

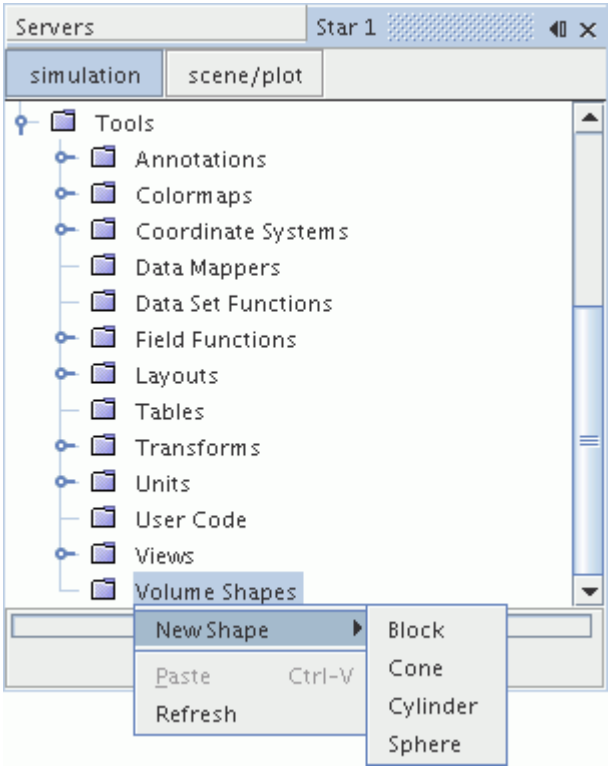
This section describes the context menu options that are available for the Volumetric Control Manager.

New	Executes the create volumetric control action.

Working with Volume Shapes

A volume shape is a closed geometric figure that can be used to specify a volumetric control for surface or volume mesh refinement or coarsening during the meshing process in STAR-CCM+.

The **Tools** > **Volume Shapes** node is the manager object for all volume shapes that have been created in the simulation.



It appears in the object tree as soon as you create a mesh continuum, and is empty to allow the creation of the first volume shape. The volume shapes manager can hold any number of volume shapes up to the memory limits of the computer.

Volume shapes allow any one of four different types of closed volumes to be defined using the context menu:

- Block
- Cone
- Cylinder
- Sphere

These volume shapes can be used to define volumetric controls for mesh continua in STAR-CCM+. The volume shapes can be created manually (using data values) or graphically using the point tool, in either the global coordinate system or any locally defined ones.

Context Menu of the Volume Shapes Manager

These options are the context menu items available when you right-click the **Volume Shapes** manager node.

New	Creates a new volumetric control of the type that is selected in the submenu.	
	Sphere	Executes the create sphere action to add a sphere to the **Volume Shapes** node.

	Block	Executes the create block action to add a block to the **Volume Shapes** node.
	Cylinder	Executes the create cylinder action to add a cylinder to the **Volume Shapes** node.
	Cone	Executes the create cone action to add a cone to the **Volume Shapes** node.

Context Menu of Individual Volume Shape

These options are the context menu items available when you right-click an individual volume shape node.

Highlight	Highlights the volume shape in the current scene. This option is intended for use when the default highlighting has been deactivated, and is only available if a scene is showing in the *Graphics* window.
Edit Volume Shape in Current Scene	Executes the edit action allowing you to set the properties of this object interactively. This option is only available if a scene is showing in the *Graphics* window.

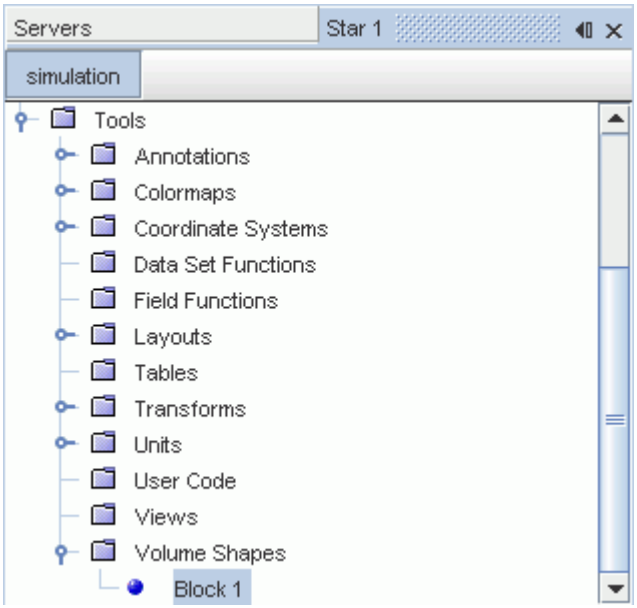
Contents:

Working with Blocks
Working with Cones
Working with Cylinders
Working with Spheres

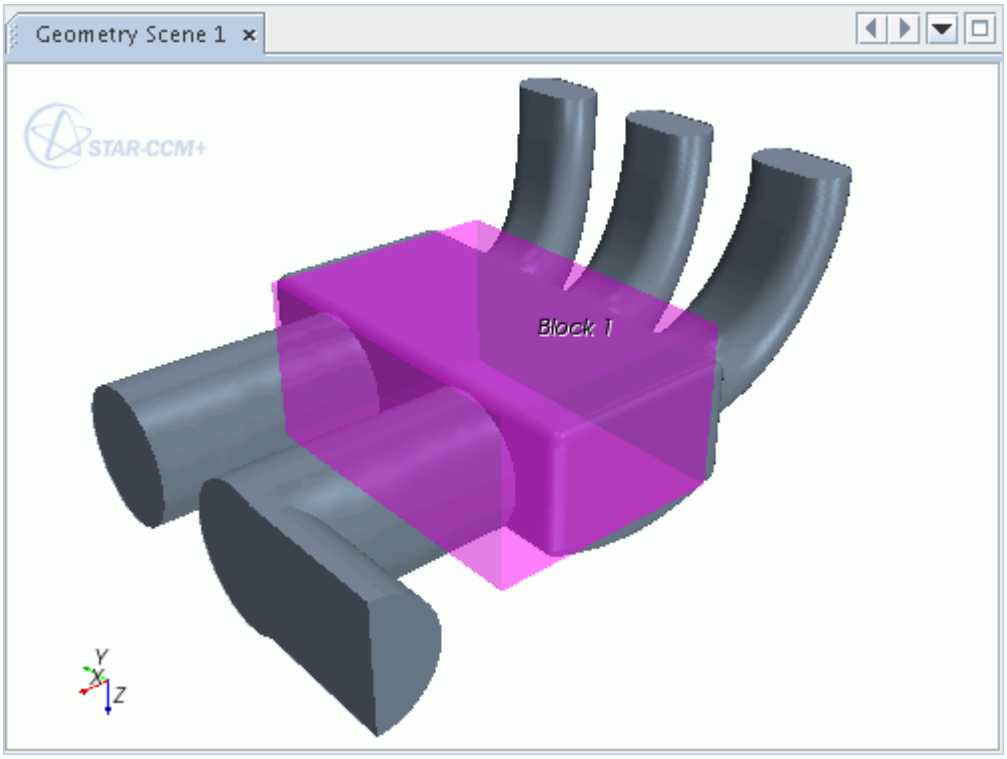
Working with Blocks

The block volume shape is a rectangular volume that is aligned with either the global or a local xyz axis.

It is defined using the coordinates of two opposite corner points. Once defined, this volume shape has a node in the STAR-CCM+ object tree, has its own properties and shares a context menu with other types of volume shapes.

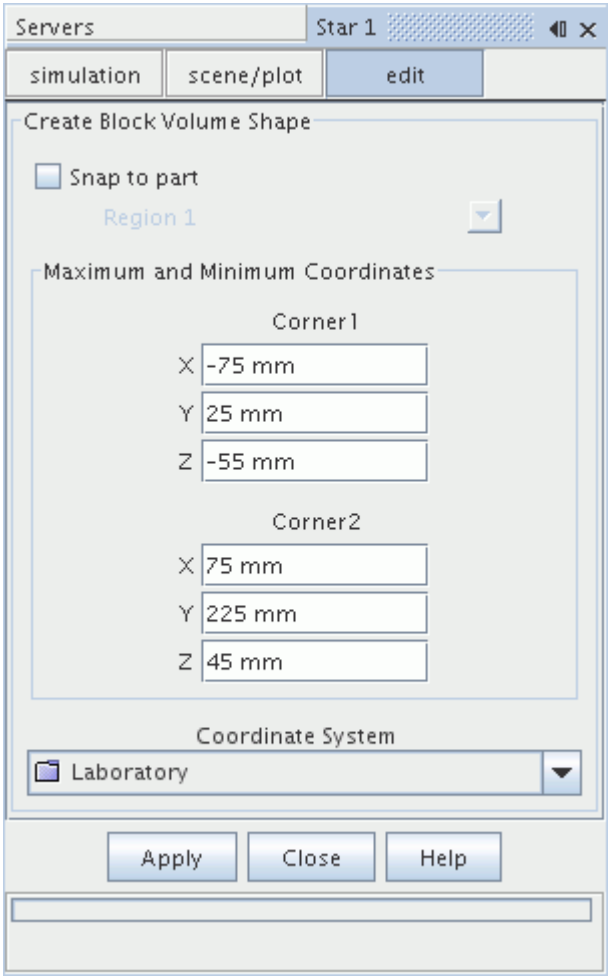


The block shape can be created either manually or graphically using the point tool. An example is shown below:



Creating a Block

On selecting the **Block** option in the **New Shape** pop up menu, the *Create Block Volume Shape* dialog appears.



To define the block, specify two separate locations (Maximum and Minimum Coordinates of the opposite corner points) based on either the global cartesian or a local coordinate system. These locations can be entered manually into the dialog, or the point tool can be used to drag the face center balls graphically to the appropriate locations. Pressing the **Create** button creates a new node that is called **Block 1** under the **Volume Shapes** node. This process can be repeated as many times as necessary to create further block shapes. Press the **Close** button to complete the operation.

Block Properties

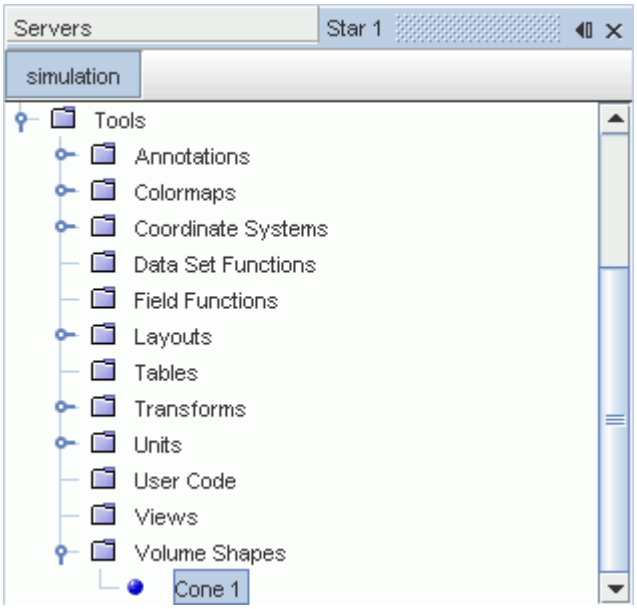
You can use the properties that are listed below to adjust the specifications of the block.

Coordinate System	The coordinate system (either global or local) in which the volume shape is defined.
Corner1	A Cartesian position vector that defines the location of one corner. Note that a comma must separate the numeric values.
Corner2	A Cartesian position vector that defines the location of the other corner, which completes the object. Note that a comma must separate the numeric values.

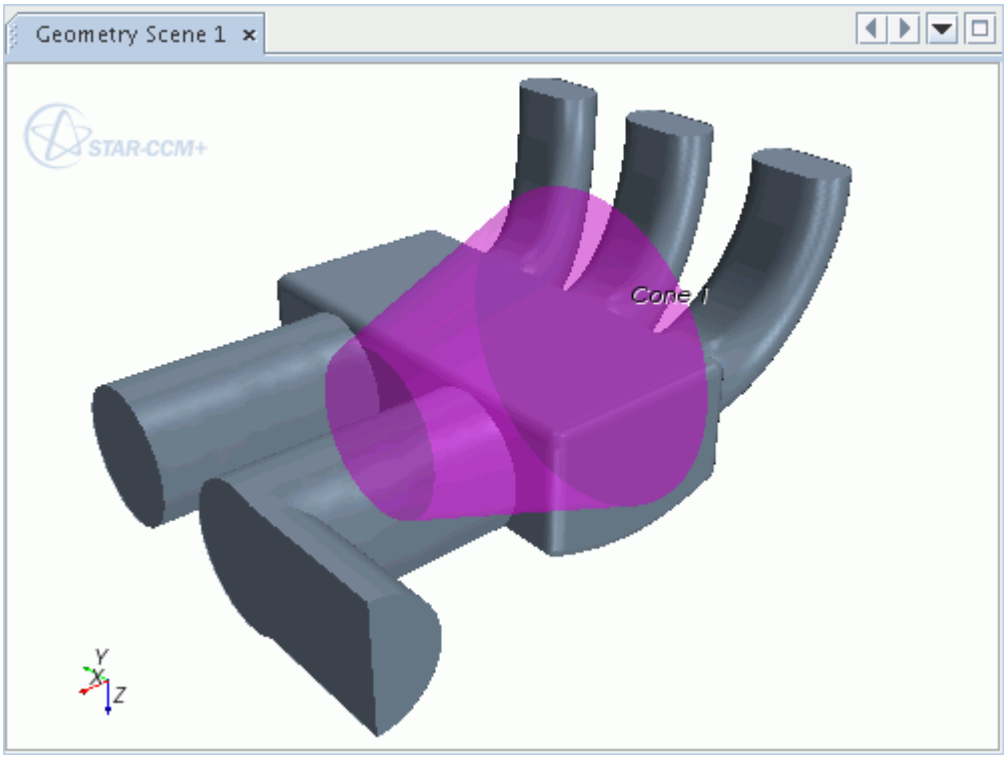
Working with Cones

To define the cone volume shape, specify two radii and the xyz center points of the two cylindrical ends in either the global or local co-ordinate system.

Once created, this volume shape has a node in the STAR-CCM+ object tree, has its own properties and shares a context menu with other types of volume shapes.

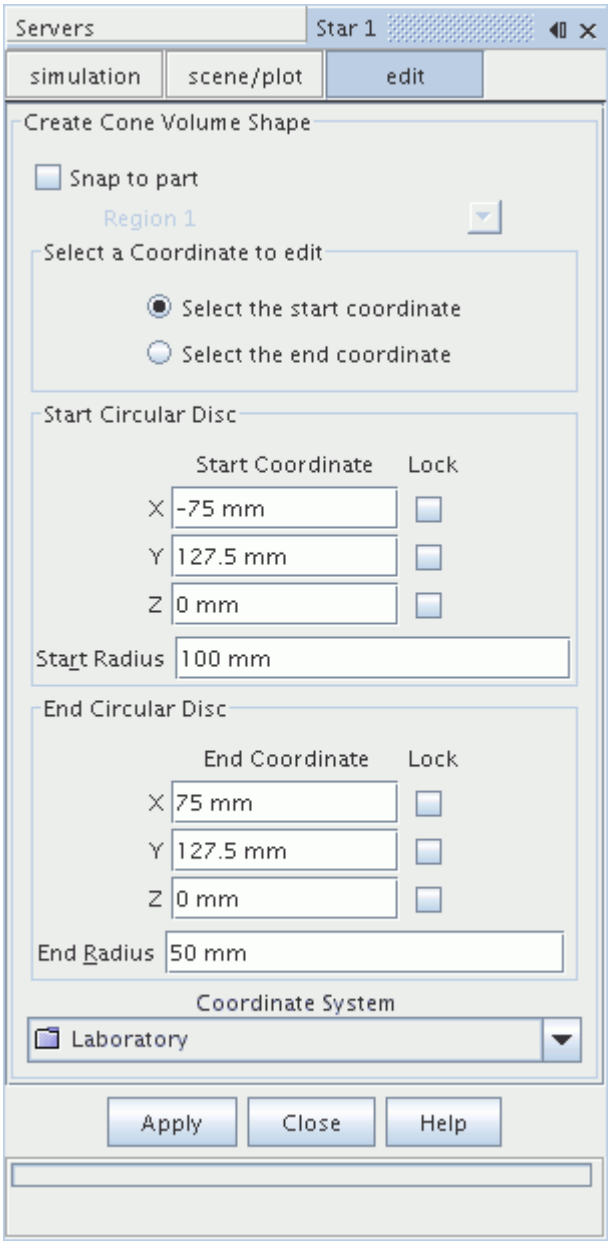


The cone volume shape can be created either manually or graphically using the point tool. An example is shown below:



Creating a Cone

On selecting the **Cone**option in the **New Shape** pop up menu, the *Create Cone Volume Shape* dialog appears on the plot display.



To define a cone, specify two separate locations (Start Circular Disc and End Circular Disc) and two radii (**Start Radius** and **End Radius**) using either the global cartesian or a local coordinate system. These locations can be entered manually into the dialog, or the point tool can be used to drag the start and end points to the appropriate locations. Note that a representation of the cone can be seen by entering the **Start Radius** and **End Radius** values and pressing enter for each. Pressing the **Create** button creates a new node that is called **Cone 1** under the **Volume Shapes** node. This process can be repeated as many times as necessary to create further cone shapes. Press the **Close** button to complete the operation.

Cone Properties

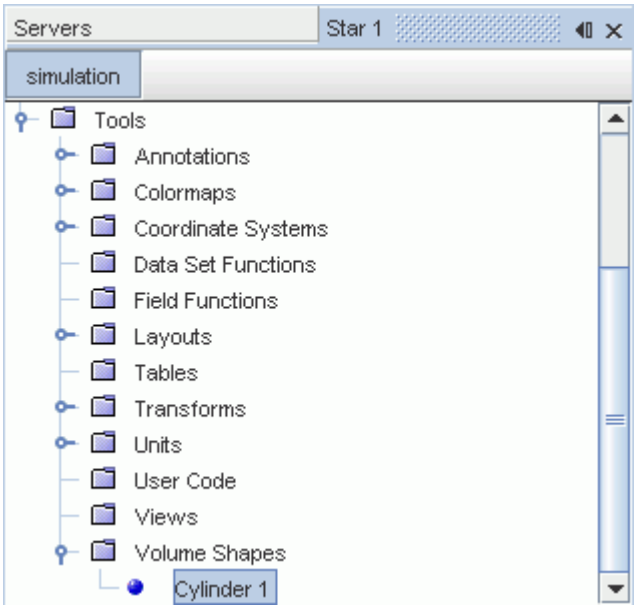
You can use the properties that are listed below to adjust the specifications of the cone.

Coordinate System	The coordinate system (either global or local) in which the volume shape is defined.
Start Coordinate	A Cartesian position vector that defines the center of the base of the cone. Note that a comma must separate the numeric values.
Start Radius	Radius of the base of the cone.
End Coordinate	A Cartesian position vector that defines the center of the tip of the cone. Note that a comma must separate the numeric values.
End Radius	Radius of the tip of the cone.

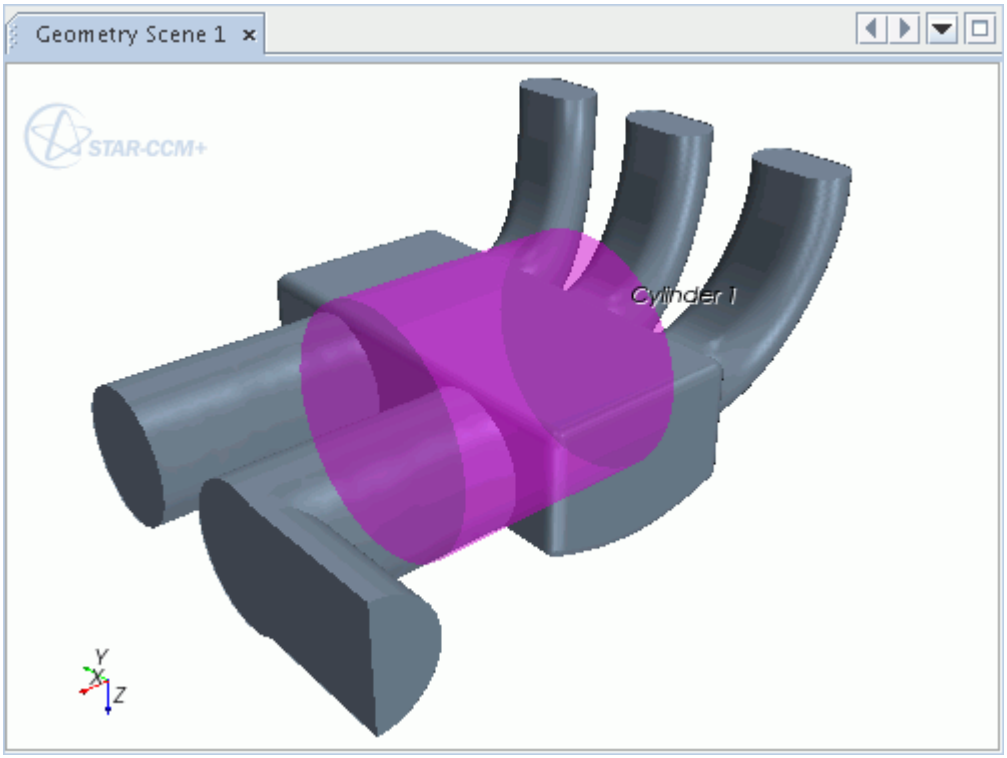
Working with Cylinders

To define a cylinder volume shape, specify a radius and the xyz center points of the two cylindrical ends in either the global or local coordinate system.

Once created, this volume shape has a node in the STAR-CCM+ object tree, has its own properties and shares a context menu with other types of volume shapes.

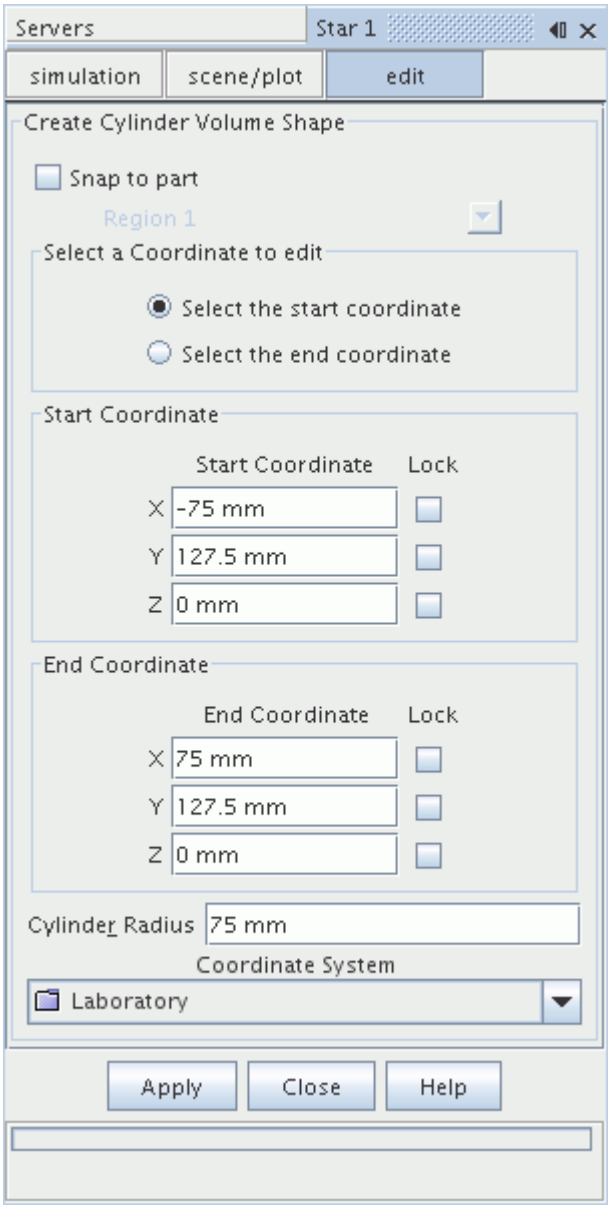


The cylinder volume shape can be created either manually or graphically using the point tool. An example is shown below:



Creating a Cylinder

On selecting the **Cylinder** option in the **New Shape** pop up menu, the *Create Cylinder Volume Shape* dialog appears on the plot display.



To define the cylinder, specify two separate locations (Start Coordinate and End Coordinate) and a **Radius** that is based on either the global cartesian or a local coordinate system. The coordinates can be entered manually into the dialog, or the point tool can be used to drag the start and end points to the appropriate locations. You can see a representation of the cylinder by entering a **Radius** value and pressing Enter. Pressing the **Create** button creates a new node that is called **Cylinder 1** under the **Volume Shapes** node. This process can be repeated as many times as necessary to create further cylinder shapes. Press the **Close** button to complete the operation.

Cylinder Properties

You can use the properties that are listed below to adjust the specifications of the cylinder.

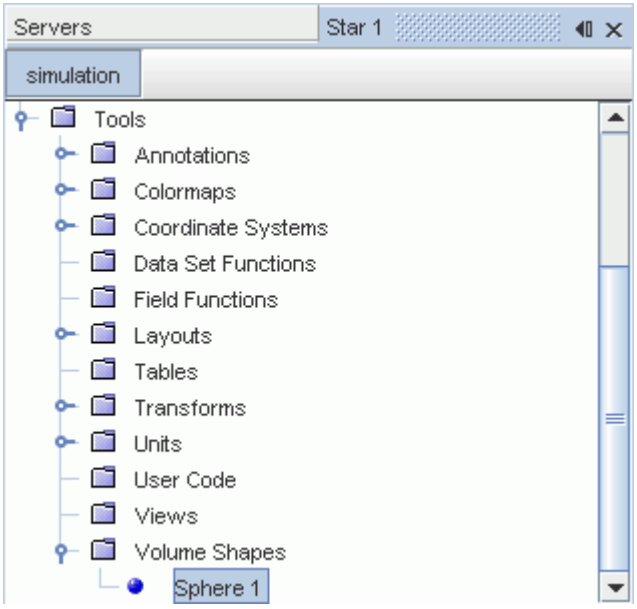
Coordinate System	The coordinate system (either global or local) in which the volume shape is defined.

Start Coordinate	A Cartesian position vector that defines the center of the start of the cylinder. Note that a comma must separate the numeric values.
End Coordinate	A Cartesian position vector that defines the center of the end of the cylinder. Note that a comma must separate the numeric values.
Radius	Radius of the cylinder.

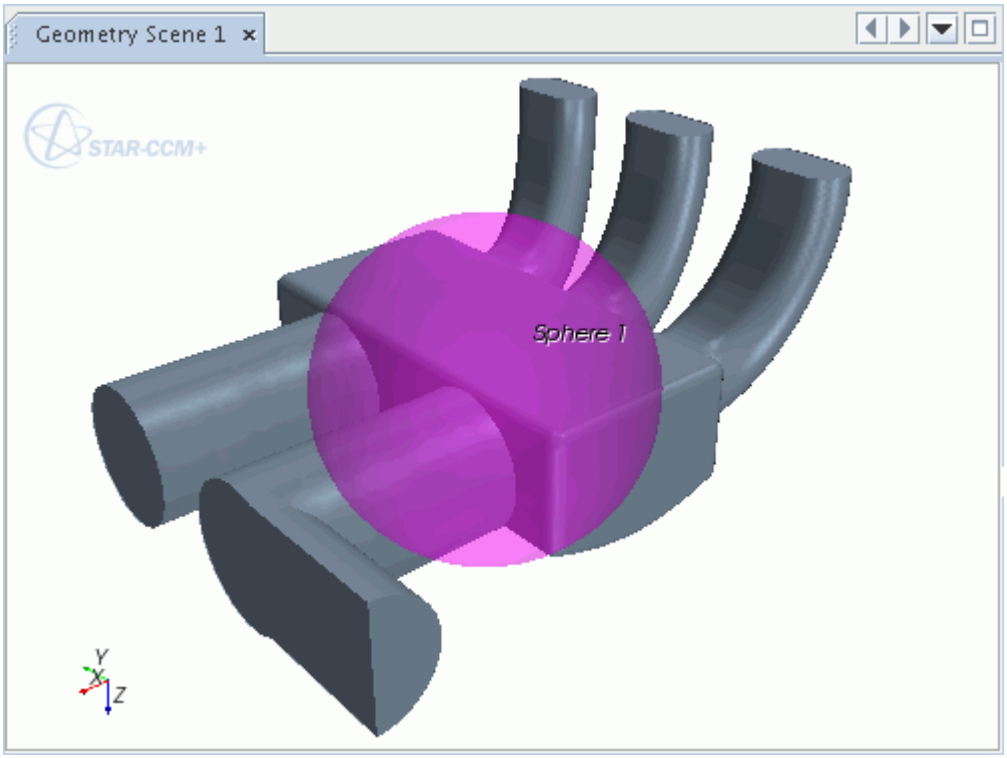
Working with Spheres

To define a spherical volume shape, specify a radius and the xyz center point of the sphere in either the global or local coordinate system.

Once created, this volume shape has a node in the STAR-CCM+ object tree, has its own properties and shares a context menu with other types of volume shapes.

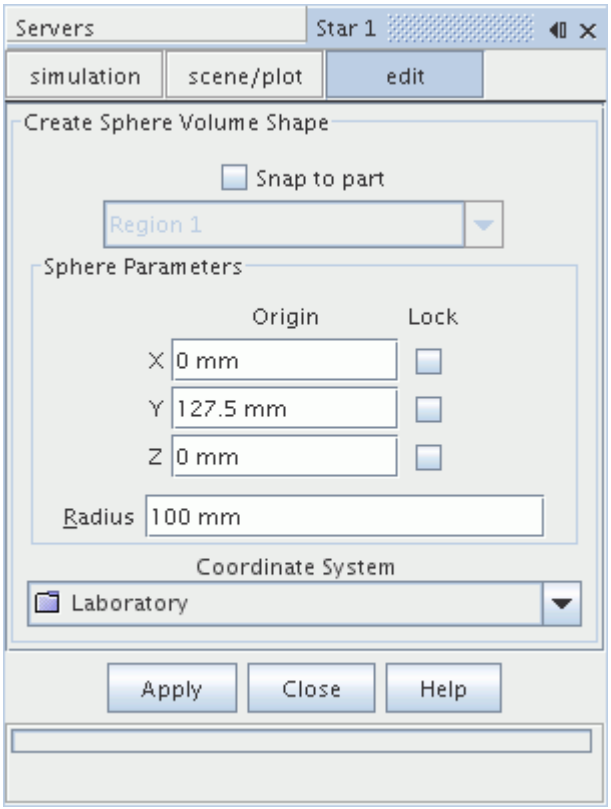


The sphere volume shape can be created either manually or graphically using the point tool. An example is shown below:



Creating a Sphere

On selecting the **Sphere** option in the **New Shape** pop up menu, the *Create Sphere Volume Shape* dialog appears on the plot display.



To define the sphere, specify the **Origin** of the sphere and the **Radius**, using either the global cartesian or a local coordinate system. The point tool allows the origin to be selected graphically by dragging the ball to the appropriate location, or alternatively, the coordinates can be entered directly into the dialog. Note that a representation of the sphere can be seen by entering a **Radius** value and pressing enter. Pressing the **Create** button creates a new node that is called **Sphere 1** under the **Volume Shapes** node. This process can be repeated as many times as necessary to create further spherical shapes. Press the **Close** button to complete the operation.

Sphere Properties

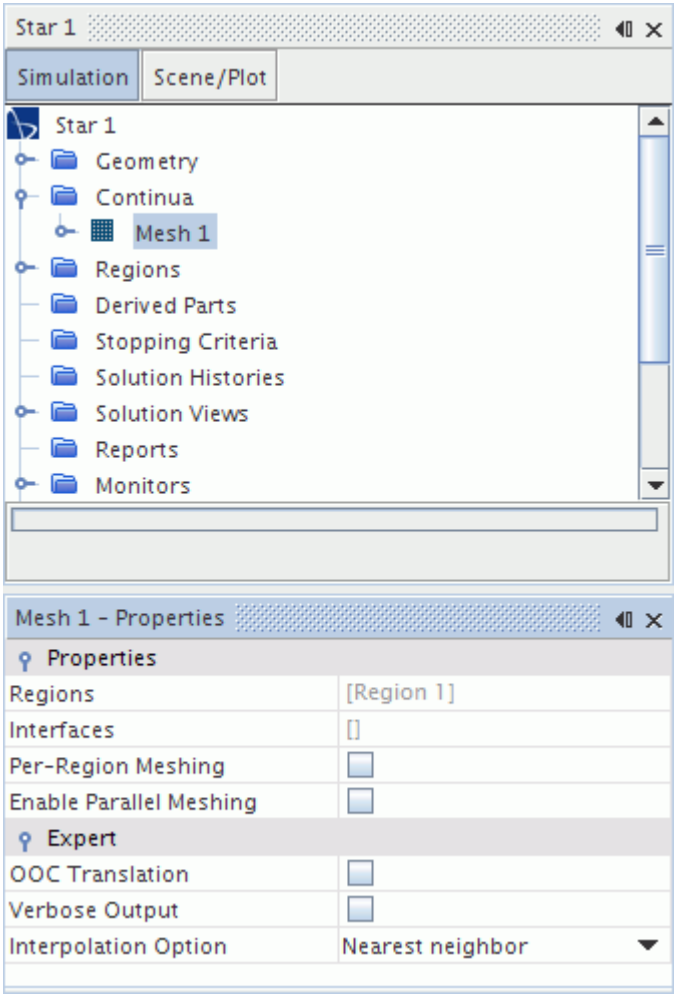
You can use the properties that are listed below to adjust the specifications of the sphere.

Coordinate System	The coordinate system (either global or local) in which the volume shape is defined.
Radius	The radius of the sphere.
Origin	A Cartesian position vector that defines the center of the sphere.

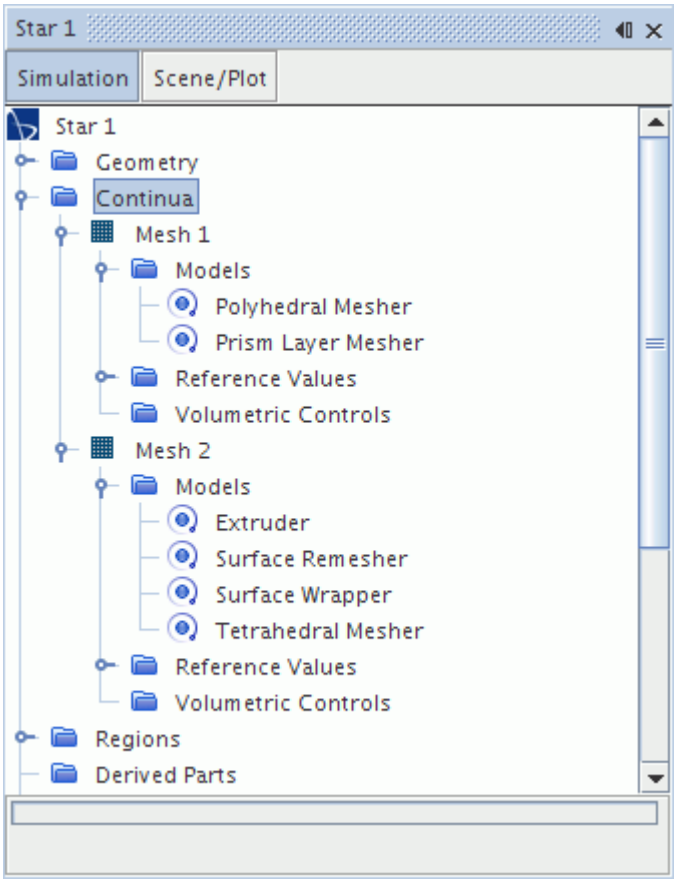
Mesh Continua

A mesh continuum is a computational domain for which a mesh is created using a collection of surface and volume meshing models.

The input geometry and associated regions define the extents and topology of the continuum. A mesh continuum node has its own properties and context menu.



An example showing two mesh continuum nodes with various meshing models activated in STAR-CCM+ is shown below.



A mesh continuum can be defined independently of both the physics and existing regions. The models that are activated in the continuum determine the characteristics of the type of mesh (surface and/or volume) that is created for the regions that are associated with that continuum during meshing.

During meshing, it is not necessary for every region to be associated with a continuum in order for the meshing process to proceed. Those regions that are not associated with a continuum are ignored and no mesh is created for them. By default, if no mesh continua exist then all regions are assigned to the first mesh continuum created. Thereafter, a null region is left unassigned if further mesh continua are created and must be assigned manually. More than one region can be associated with the same continuum.

In addition to the models comprising the continuum, the mesh continuum also contains:

- reference values
- volumetric controls

Contents:

Mesh Continuum Properties
Mesh Continuum Expert Properties
Mesh Continuum Pop-Up Menu
Interpreting the Verbose Output

Mesh Continuum Properties

This section describes the properties of the mesh continuum.

Per-Region Meshing	Activates per-region meshing for regions that are associated with this continuum.
Use Parallel Meshing	Activates parallel meshing for this continuum.
Interfaces	The interfaces that are associated with the continuum (read-only).
Regions	The regions that are associated with the continuum (read only).

Mesh Continuum Expert Properties

This section describes the export properties of the mesh continuum.

OOC translation	Determines the use of file I/O for meshing operations.	
	Activated	Allows file I/O to be used for meshing operations.
	Deactivated	File I/O is not used for meshing operations.
Verbose Output	Provides further output from the meshing models while they are running. This output can be useful for debugging problems when they occur.	
	Activated	Activates verbose output.
	Deactivated	Leaves output at its default.
Interpolation Option	Determines the method that is used to map the solution from an existing volume mesh to a newly generated volume mesh. These options are the same as the options provided for region mesh replacement.	
	Nearest neighbor	The cell/face value on the new mesh is assigned the value of the cell/face on the old mesh whose cell/face centroid is nearest to the cell/face centroid of the new mesh. Does not provide smooth interpolation of results.
	Higher-order stencil	The cell/face value on the new mesh is smoothly interpolated from a stencil of cells/faces on the old mesh. Smooth interpolation is based on a distance-weighted, least squared, interpolation scheme. Preferred method for transient analyses.

Mesh Continuum Pop-Up Menu

This section describes the pop-up menu for the mesh continuum.

Select Meshing Models...	Executes the select models action to allow selection of the models to be activated in this continuum.

Interpreting the Verbose Output

Activating the *Verbose Output* property for a mesh continua reports more detail on the tasks the surface and volume meshers perform when generating a mesh. This information can be useful for debugging and evaluating the volume mesh.

However, note that the extra time taken to generate the verbose output can increase meshing CPU time by approximately 20%, which can be an issue for large meshes.

There are many differences between the normal and verbose output, which would not be feasible or necessary to fully document. Generally, the verbose output provides more information at each stage of the meshing process. For example, when fixing non-manifold baffle surfaces, the normal output reports the following:

```
     Checking baffles for containment
     Fixing non-manifold surface, CPU Time: 0.13, Wall Time: 0.13,
Memory: 215.89 MB
```

Whereas the verbose output reports more detail on what the mesher is doing to fix the non-manifold baffle surface, as shown below:

```
    Checking baffles for containment
    Fixing non-manifold surface
    found 1810 nonmanifold edges
    found 0 nonmanifold vertices
    created face pairs
    created vertex manifolds
    created 1810 new edges
    created 602 new vertices
    done, CPU Time: 0.16, Wall Time: 0.16, Memory: 216.92 MB
```

This type of information can be useful for identifying the cause of errors during the meshing process.

The verbose output also reports more information on the mesh statistics throughout the meshing process. For example, when generating a polyhedral mesh, the normal output reports the following:

```
    Estimating number of cells...
    estimated number of cells: 1468015
    done, CPU Time: 0.03, Wall Time: 0.03, Memory: 225.77 MB
```

Whereas the verbose output reports further statistics on the volume mesh, as shown below:

```
    Estimating number of cells...
    estimated number of cells:   1468015
```

```
    volume enclosed by surface:  3.36484e-003
    area of surface:             2.55161e-001
    average volume:              1.87295e-009
    average area:                2.73713e-006
    average edge length:         2.51419e-003
    done, CPU Time: 0.03, Wall Time: 0.03, Memory: 226.19 MB
```

The key difference between the normal and verbose output is the mesh statistics that are printed at the end of the report. This output includes quality and validity metrics for the volume mesh, as detailed below:

- Polyhedral quality

  This value refers to the cell quality of the polyhedral cells.
- Cell face validity

  This value refers to the face validity of the cells.
- Cell orthogonality

  This value refers to the angle between the normals of two faces in a cell that share an edge. The closer the cell edge angles are to 90 degrees, the more orthogonal the cell is.
- Cell skewness angle

  See cell skewness angle.
- Cell face planarity

  This value is a measure of the how planar the cell face is. A value of 1.0 indicates that face is planar.
- Cell face twist

  This value is a measure of the how twisted the cell face is. The closer the value is to 1.0, the more twisted a face is.
- Cell volume ratio

  This value refers to the volume change between neighboring cells.

Each volume mesh metric is reported in a tabular form, with the metric value range, number of cells and percentage. The minimum or maximum value in the volume mesh (depending on which is less desirable) is also reported for each metric. An example of polyhedral quality output is provided below:

Output - Star 1

```
------------------------------+-----------+-----------
 Polyhedral quality (norm.)   |  # Cells  | % of Total
------------------------------+-----------+-----------
  1.000000000 - 0.750000000   |     44500 |   18.944%
  0.750000000 - 0.500000000   |    115152 |   49.020%
  0.500000000 - 0.250000000   |     63546 |   27.052%
  0.250000000 - 0.100000000   |      9558 |    4.069%
  0.100000000 - 0.010000000   |      1291 |    0.550%
  0.010000000 - 0.001000000   |       857 |    0.365%
  0.001000000 - 0.000100000   |         3 |    0.001%
  0.000100000 - 0.000001000   |         0 |    0.000%
  0.000001000 - 0.000000001   |         0 |    0.000%
  0.000000001 - 0.000000000   |         0 |    0.000%
------------------------------+-----------+-----------
 Minimum Polyhedral Quality: 4.069396e-004
```

Surface Meshers

STAR-CCM+ contains meshing tools which can be used to help prepare a high-quality surface mesh before a volume mesh can be generated.

Depending on the initial quality of the geometry, there are several surface meshing tools available:

- Surface Wrapper
- Surface Remesher
- Automatic Surface Repair

The surface wrapper is used when the imported geometry is of such poor quality (intersecting surfaces, holes, gaps, and so on) that it is not directly usable for volume meshing. In these cases carrying out surface repair and remeshing would typically involve more effort than any extra fidelity obtained would warrant. The surface wrapper is used with the feature extraction and feature editing tools to ensure a high-fidelity representation of the underlying geometry.

The result of a surface wrapping operation is a closed (watertight) surface that may require further remeshing to improve mesh quality.

The surface remesher is used to retriangulate a closed starting surface to obtain a high-quality surface mesh. It is used to improve surface meshes from the surface wrapper or to improve the quality of imported tessellated geometry. When used on imported geometry directly, surface remeshing is used with the surface repair tools such as the hole filling and edge zipping tools to ensure a closed surface. Many of these repair tools can be automated by using the automatic surface repair tool to minimize the manual effort required.

This section also discusses:

- When to use each model
- How to launch the surface mesh generator

Contents:

Surface Meshers Overview
Surface Wrapper
Surface Remesher
Automatic Surface Repair
Surface Mesher Controls and Values

Surface Meshers Overview

This section provides a comparison of the Surface Remesher and Surface Wrapper

To decide which surface meshing approach to use in preparation for volume meshing, weigh up the level of fidelity that is required against the acceptable level of manual effort.

For example, in cases where it is important that there is little or no loss of surface detail, the surface remesher would be used to maintain the original features. In cases where a significant level of de-featuring is required, and there are many gaps and mismatches between surfaces and parts, the surface wrapper would be used.

To decide which model or models are appropriate, use the following guidelines:

Instances when surface remesher would be used:

- When a high level of surface mesh fidelity to the underlying geometry is required
- When the imported surface is closed but composed of bad quality triangles (for example, STL data)
- When the imported surface is closed but does not contain the required triangulation resolution
- When the surface wrapper has been used, and the surface mesh needs further improvement

Instances when the surface wrapper would be used:

- When a high level of surface accuracy is not critical
- When the imported surface contains large number of gaps, holes and/or mismatches that cannot be easily fixed manually
- When the imported surface contains multiple volumes that intersect but are not joined
- When the imported geometry contains multiple overlapping surfaces that must be merged into one
- When the imported geometry contains too much detail, requiring de-featuring

See Also:

Preparing Boundaries for the Generalized Cylinder Mesher

Surface Wrapper

The surface wrapper can be used to provide a closed, manifold, non-intersecting surface when starting from poor quality CAD data.

It is typically used when the imported geometry has:

- Multiple intersecting parts
- Missing data in the form of holes and gaps
- Surface mismatches
- Duplicate and internal surfaces
- Overly complex geometry with too much detail

The surface wrapper can:

- Provide a closed, manifold, triangulated surface definition
- Close holes, gaps, and mismatches
- “Skin” duplicate surfaces and remove internal features
- Simplify the surface geometry by removing unwanted detail
- Include baffle thickness inflation for desired internal features
- Include refinements that are based on curvature, proximity, and individual surfaces

The resulting surface quality from the surface wrapper is not optimal so it is used with the surface remesher to provide a high quality starting surface for the volume meshers.

An example result from the surface wrapper is shown below:

The surface wrapper controls can be set in three different groupings:

- Default controls - the global settings.
- Custom controls - settings that control the mesh density near certain part surfaces, feature curves, or volume parts.
- Contact prevention - settings to prevent surfaces incorrectly connecting.

These groupings let you control the relative level of fidelity versus de-featuring, and control the distribution of surface mesh resolution.

The location of these controls varies in the GUI depending on whether you are using parts-based meshing or region-based meshing. For parts-based meshing, the controls are all under the **Operations** > **Surface Wrapper** node. For region-based meshing, the global controls are under the **Continua** > **Mesh** node. The custom controls are under the region node to which they pertain, and the contact protection controls are under the **Regions** > **Body** > **Mesh Values** node.

The leak detector option can be used to check if holes or gaps exist in the surface either before wrapping or after the surface mesh has been created.

Input part surfaces or are typically used as input parts to a single surface wrapper operation to generate a single wrapped surface. If you want to wrap some parts separately, then use a separate surface wrapper operation (PBM) or mesh continuum (RBM) for each group of surfaces. Alternatively, activate the per-part meshing (PBM) or per-region meshing (RBM) option to generate a separate wrapped surface for each part or region.

If you only want to wrap part of a geometry, you can activate the surface wrapper property, *Perform Partial Wrapping*. This property allows you to select the faces that you want to exclude from wrapping in a surface wrapper operation. Excluded faces are not wrapped but are still included in the final mesh. See Setting the Volume of Interest and Wrap Properties.

Contents:

Gap Closure
Leak Detection
Contact Prevention
Surface Wrapper Defeature Control
Surface Wrapper with Internal Baffles
Surface Wrapper Properties

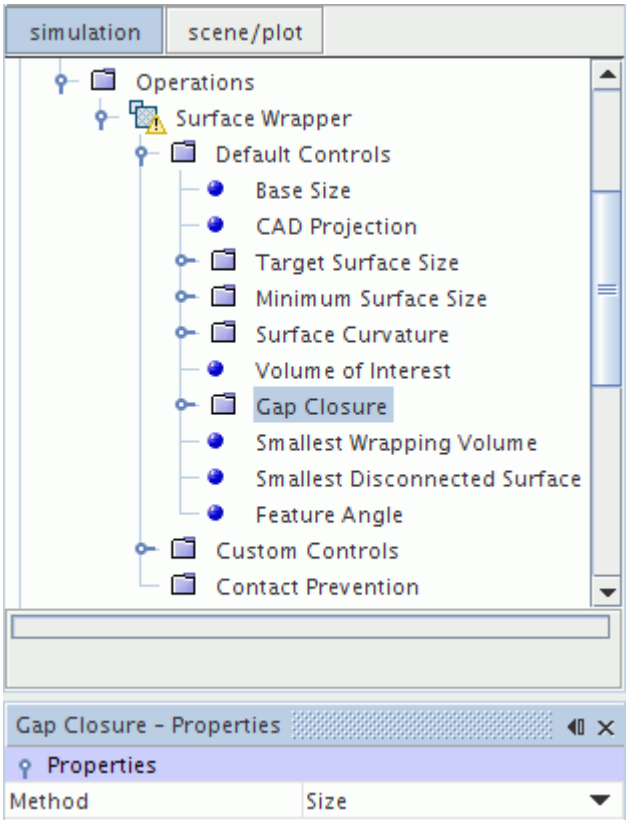
See Also:

Generating a Surface Mesh

Gap Closure

The surface wrapper can close holes in the input geometry during surface wrapping.

Three methods are provided to control the behavior of gap closure, which you select from the properties of the Gap Closure node.



Gap Closure Properties

The following options are available for the *Gap Closure - Properties Method*:

Size	Allows you to specify gap closure **Size type** as an **Absolute** value, or **Relative to base**. Any holes that are smaller than the specified gap closure size are closed during surface wrapping.
Seed Point	Allows you to specify the location of **Seed Points** within your geometry. Only volumes that contain **Seed Points** are closed during surface wrapping. Generally, gaps are closed at the smallest possible opening between the gap closure seed point and the volume of interest. If there are multiple small openings, the wrapper does not guarantee which it closes.
Both	Allows you to specify a **Gap Closure Size** and use gap closure **Seed Points**. During surface wrapping, any holes that are smaller than the **Gap Closure Size** are closed. Holes that are larger than the **Gap Closure Size** are also closed based on the gap closure **Seed Points**.

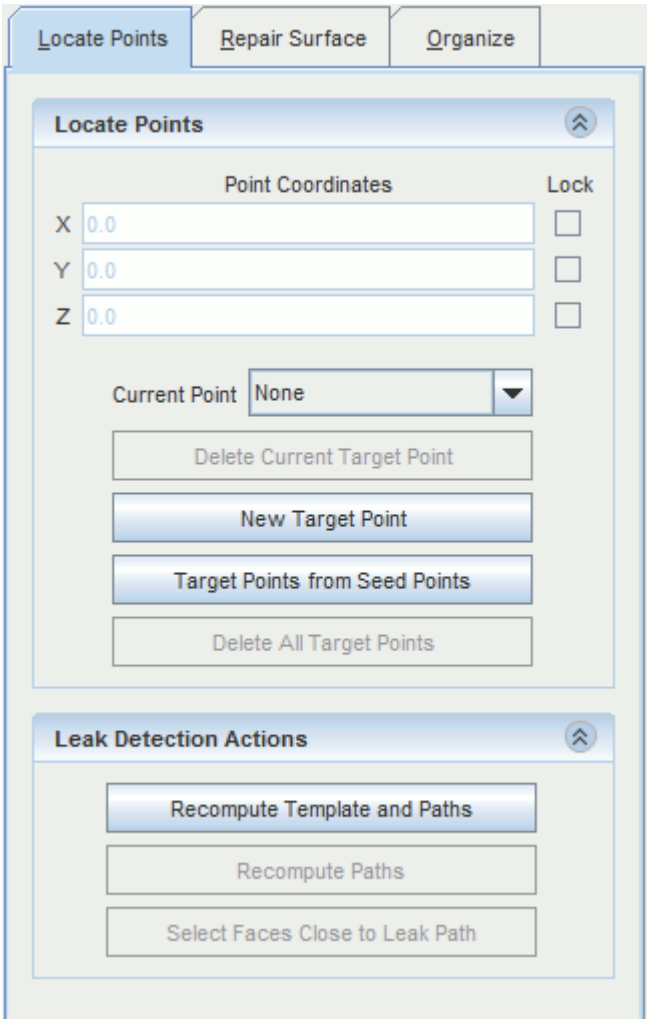
Leak Detection

The surface wrapper leak detection tool is useful for determining whether “leaks” exist in the resultant wrapped surface that is created from an imported surface.

A leak is a gap or a hole that would allow a path to be traced from one volume through to another volume or the outside without crossing through a surface. Therefore, a fully enclosed surface has no leaks.

The leak detection tool can only be used if an import surface representation of some kind exists. The leak is detected in the octree version of the mesh that is built based on the supplied wrapper inputs. However, if a leak is found then the path is displayed using the original import surface representation (not the octree or wrapped mesh). Displaying the leak in this way allows repairs to be made to the surface if necessary to avoid the leak from occurring in the actual wrapped surface representation.

For information on how to use the leak detection tool, see Closing Gaps and Checking for Leaks.



The *Wrapper Leak Detection* edit pane allows you to access three different tools:

- **Locate Points** — Create and move source and target points for the leak detection calculation
- **Repair Surface** — Patch up the surface before and after leak detection; Inflate or translate selected faces or edges
- **Organize** — Create control sets and edit parts. A set is a collection of faces, edges, or vertices, which you can store, recall and browse

By default, the panel shows the **Locate Points** tool, which allows you to define the location of the source and target points using either the *Point Coordinates* entry boxes and/or the graphical point tool. Click **Target Points from Seed Points** to create target points from existing seed points in the surface wrapper. The new target points have the same X, Y, Z coordinates and same name as the seed points they are created from.

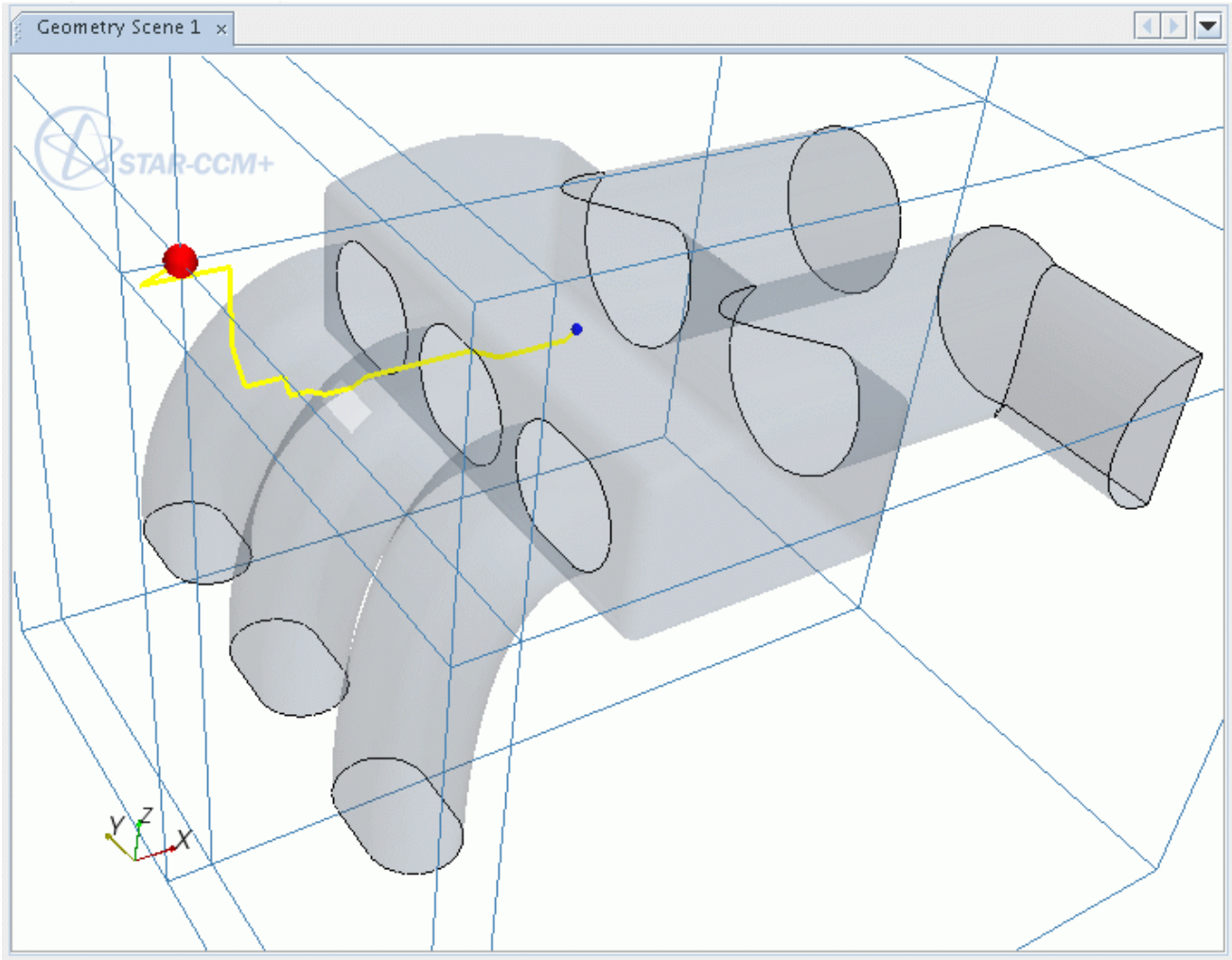
By default, there is a single source and target point. The source point (blue) would typically reside inside the geometry volume while the target point (red) would be anywhere outside (in another volume if necessary).

The tools within the *Repair Surface* and *Organize* tabs are identical to those in the Surface Repair tool. For more details about individual tools, refer to the following sections:

- Repair Tools Reference
- Set Controls

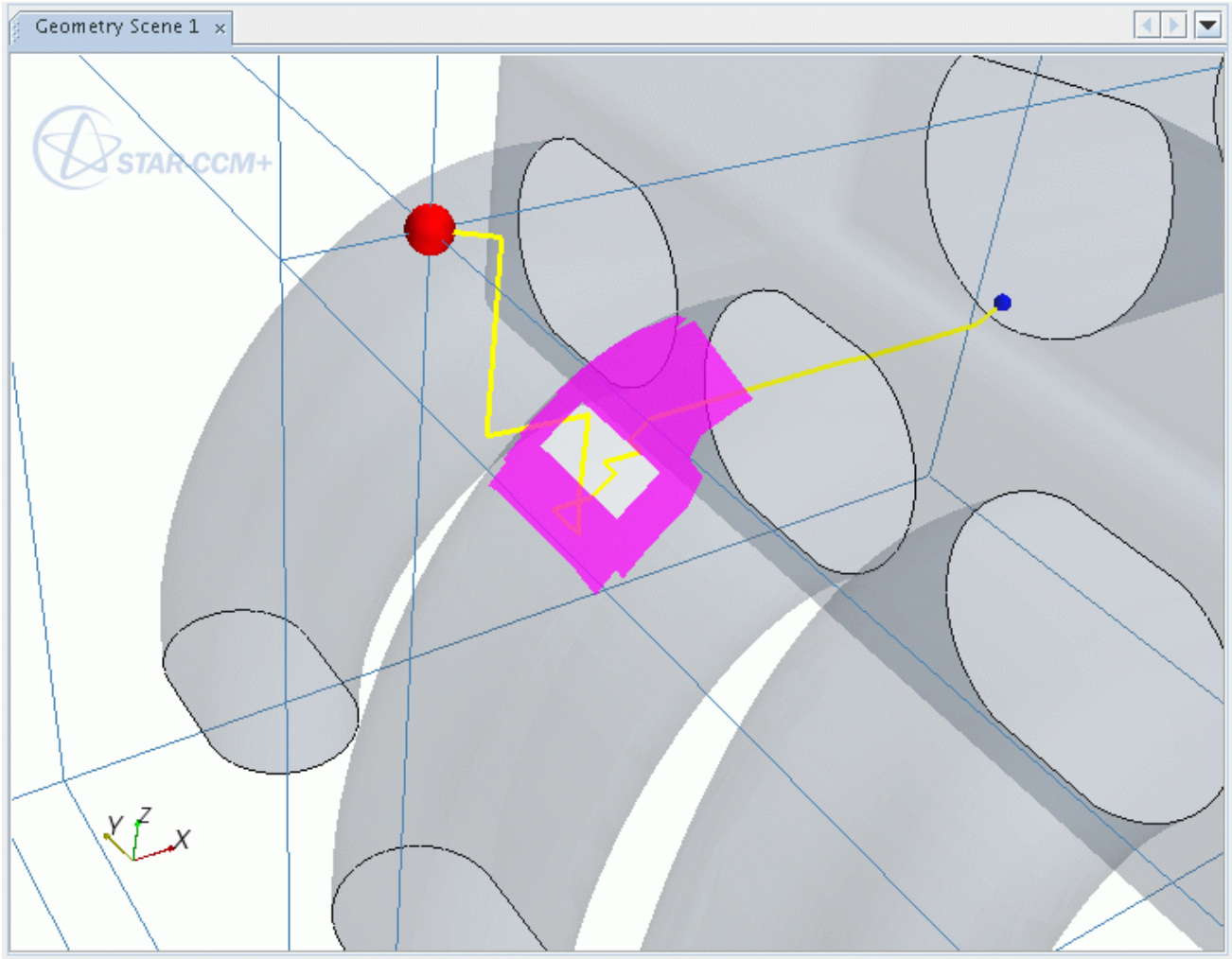
- Part Controls

When running leak detection for the first time on a new geometry, the method builds a template that is based on the supplied input parameters for the surface wrapper. The wrapper then checks to see if a leak would exist based on the template cell sizes. If a leak is found, then it is traced using a line from the source point to the target point, as shown below:

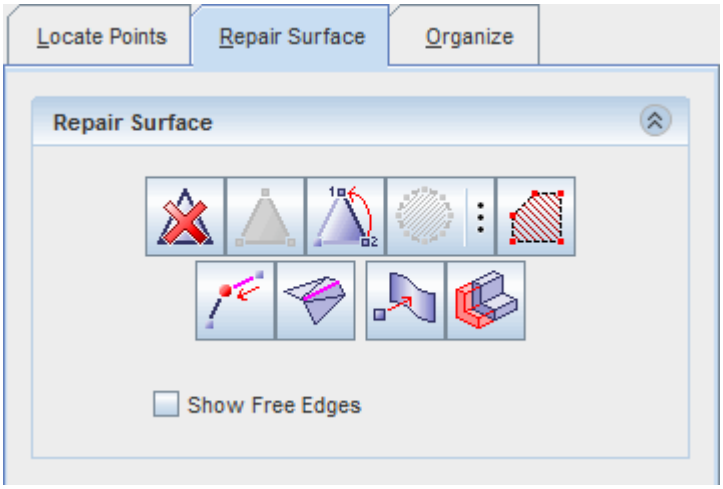


Where the leak path crosses through the surface indicates the location of the hole which must be fixed.

An example leak path face selection is shown below:



Being automatically able to select the faces next to the hole aids with the manual surface repair step that follows, as described below. Once a leak path has been detected, you can access the **Repair Surface** tool by selecting the **Repair Surface** tab in the *Wrapper Leak Detection* dialog.



Using this dialog, any holes in the surface can be repaired using the *Repair Surface* buttons, which are identical in behavior to those in the Surface Repair tool (see Repair Tools Reference). The **Show Free Edges** toggle can be used to highlight (using green lines) the holes within the geometry. Of particular importance in the **Surface Repair** options is the **Fill polygonal patch** button, which can be used to patch arbitrary holes in the surface simply by clicking the surface around the hole. This option does not fill the hole perfectly but covers it sufficiently for the surface wrapper not to detect a hole anymore and produces a uniform surface in its place. The Selection Control, Display Control, and Boundary/Feature Curve Edit options in the dialog can be used the same way as for the normal surface repair dialog.

Note that if you imported the geometry as a part, filling holes at this stage only applies changes to the initial surface representation, whereas the geometry part representation remains unchanged. This fact can cause problems if you reinitialize the mesh, as the surface repairs are lost. To ensure that fixes to the surface are permanent, exit the leak detection tool, and use surface repair at the parts level to fix the underlying geometry. This recommendation does not apply if you imported the surface mesh as a region.

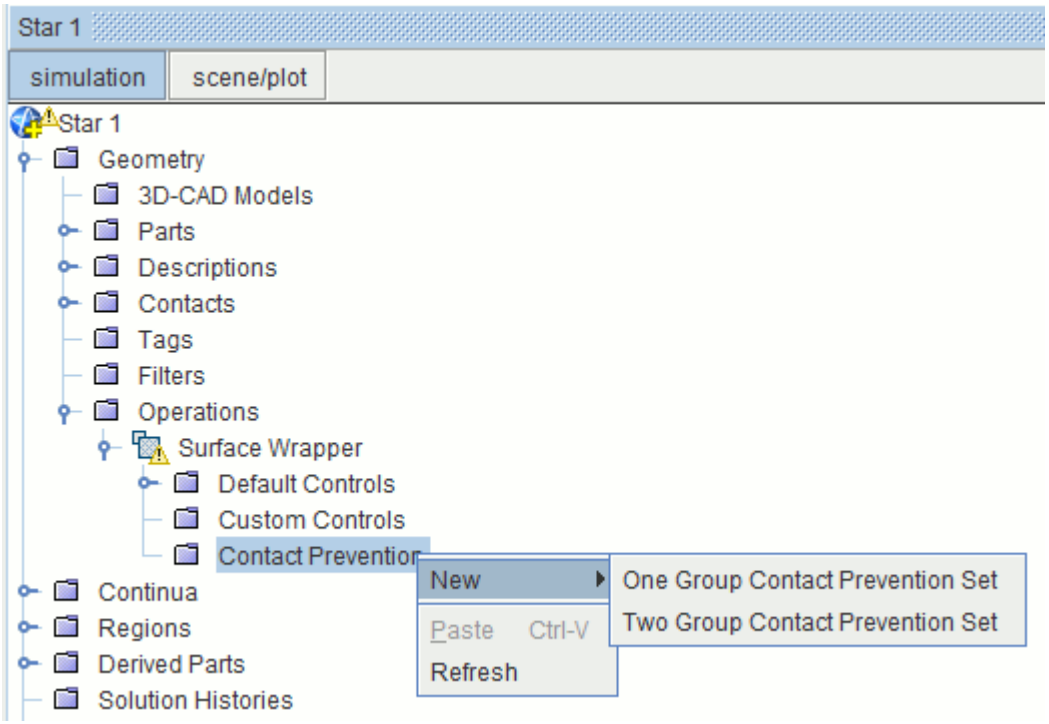
To reiterate - the leak detection option is run before the wrapped surface is created (that is, on the original input surface). The leak detection tool accounts for the current wrapper settings and determines if any gaps in the surface would be closed based on the supplied settings. If the sizes specified are not sufficient, it reports back with the leak path through the hole, which must be fixed before progressing. Alternatively, the wrapper settings, in particular the gap closure size, can be modified so that the wrapper can automatically close the hole itself.

Contact Prevention

Contact Prevention avoids the merging of two or more part surfaces that are close to one another during surface wrapping.

This functionality is useful during meshing as it keeps certain surfaces separated, which would otherwise be amalgamated as one due to the surface mesh size settings.

To activate contact prevention for two or more surfaces when using a parts-based meshing approach, right-click **Operations** > **Surface Wrapper** > **Contact Prevention** and select **New**:



If you are using region-based rather than parts-based meshing, contact prevention is found under **Regions** > **[Body name]** > **Mesh Values** > **Contact Prevention**. It works with region boundaries rather than part surfaces, but otherwise looks and functions in an identical way.

Two options are available:

- One Group Contact Prevention Set
- Two Group Contact Prevention Set

Once you have defined a contact prevention set, the following properties are available:

- Search Floor
- Enable Contact Prevention Set
- Using Parts and Shapes to limit contact prevention

In addition to common menu items, the right-click menu of a contact prevention set contains actions for:

- Interactively selecting surfaces within the set
- Specifically highlighting surfaces when default highlighting has been deactivated.

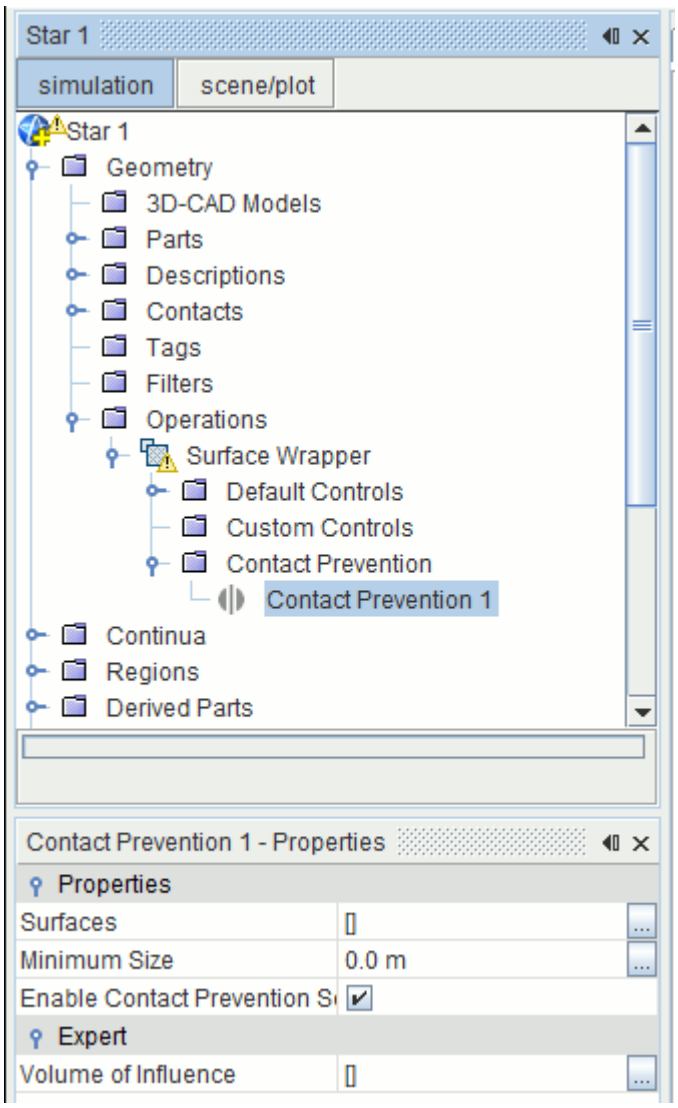
Empty contact prevention sets appear grayed out in the object tree and behave like deactivated sets. These empty sets are still considered active but do not have any effect on the mesh — the mesh pipeline continues normally. If you want to deactivate a particular contact prevention set, see Enable Contact Prevention Set.

One Group Contact Prevention

One Group Contact Prevention Set allows individual part surfaces to be grouped, such that each part surface forms a contact prevention with every other part surface grouped in the set.

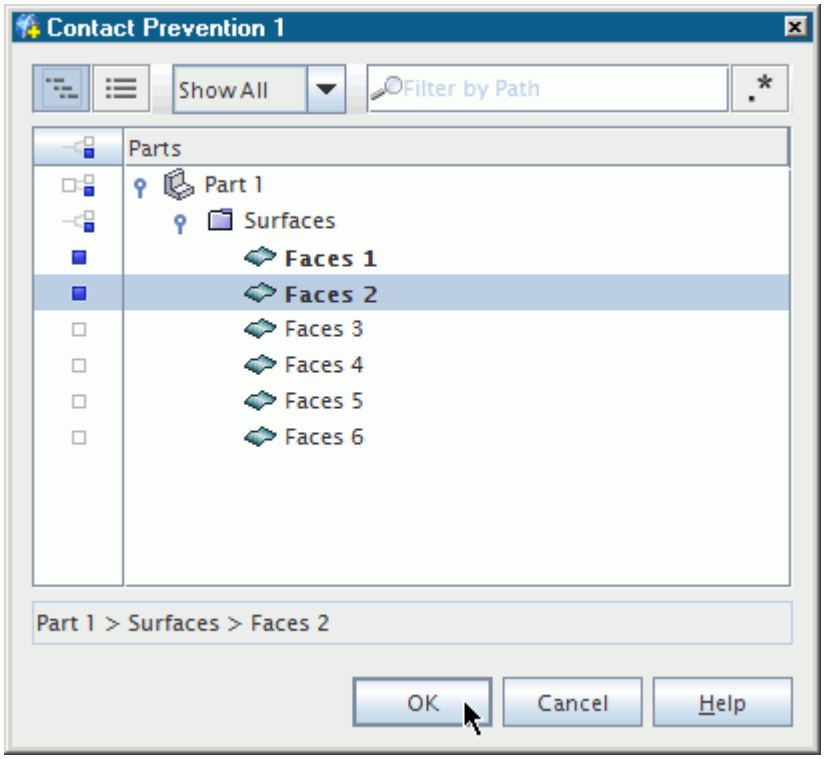
1. Selecting **One Group Contact Prevention Set** creates a node that is called **Contact Prevention 1**.

The properties of this node are shown below for a parts-based meshing approach:



The part surfaces that form the contact prevention set can be selected using two different methods.

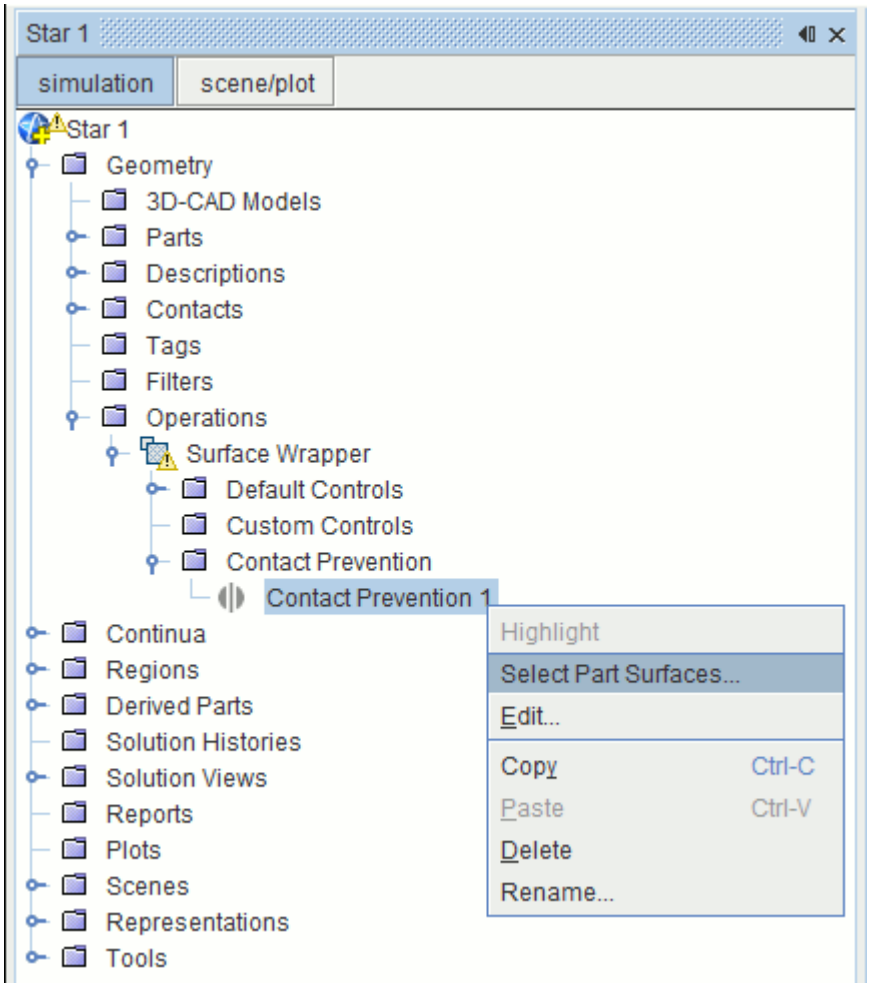
The first option uses the object selection dialog, which is accessible by clicking the ellipsis (**...**) next to *Surfaces* in the *Properties* window.



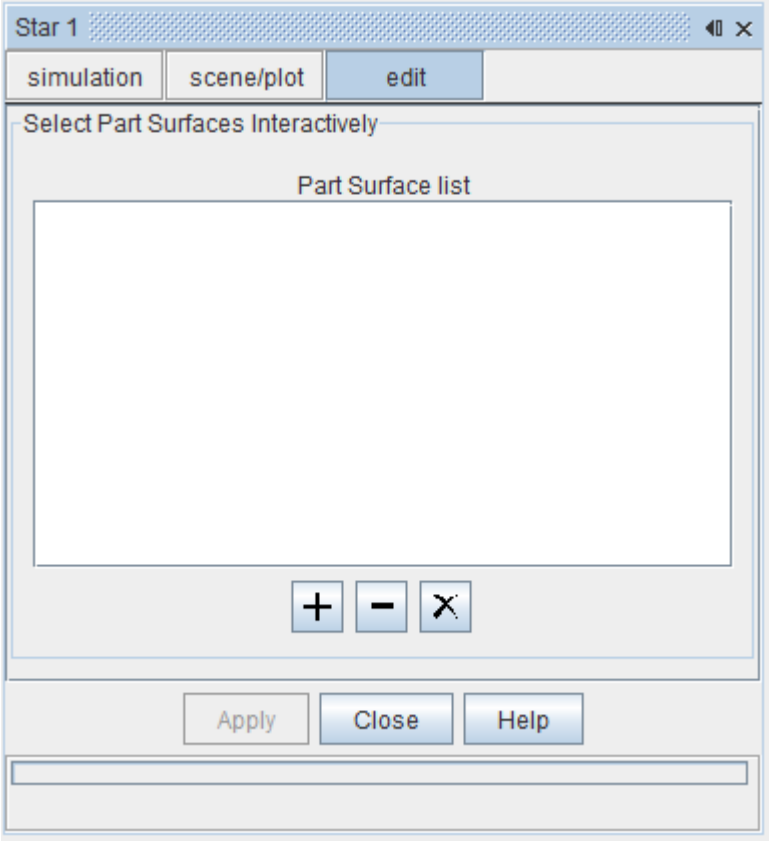
If you only have a few part surfaces in the model and they are easily identifiable, using the object selection dialog to select the part surfaces is the easiest and simplest method.

However, if you have many part surfaces or prefer to select part surfaces interactively in the *Graphics* window, use the *Select Part Surfaces Interactively* dialog.

2. To access the *Select Part Surfaces Interactively* dialog, first ensure that there is a scene open, as the dialog requires selection of part surfaces in the *Graphics* window.
3. Then, right-click the **Contact Prevention** node and select **Select Part Surfaces...**



The *Select Part Surfaces Interactively* in-place dialog opens:



4. Select the part surfaces for the contact prevention set using the mouse in the *Graphics* window (press `<Ctrl>` for multiple selections).

5. Once you have made the appropriate selections, click + to add the part surfaces to the list. Alternatively, right-click a selected part surface and select **Add Part Surface**.

   Selected part surfaces appear blue in the display, indicating they belong to a set.

6. If part surfaces must be removed, select the part surfaces in the list box or in the display and click − to remove them. Alternatively, right-clicking on a part surface in the display provides a **Remove Part Surface** option. Clicking ✕ removes all part surfaces from the list box.

   Select at least two part surfaces in order for the contact prevention to be valid.

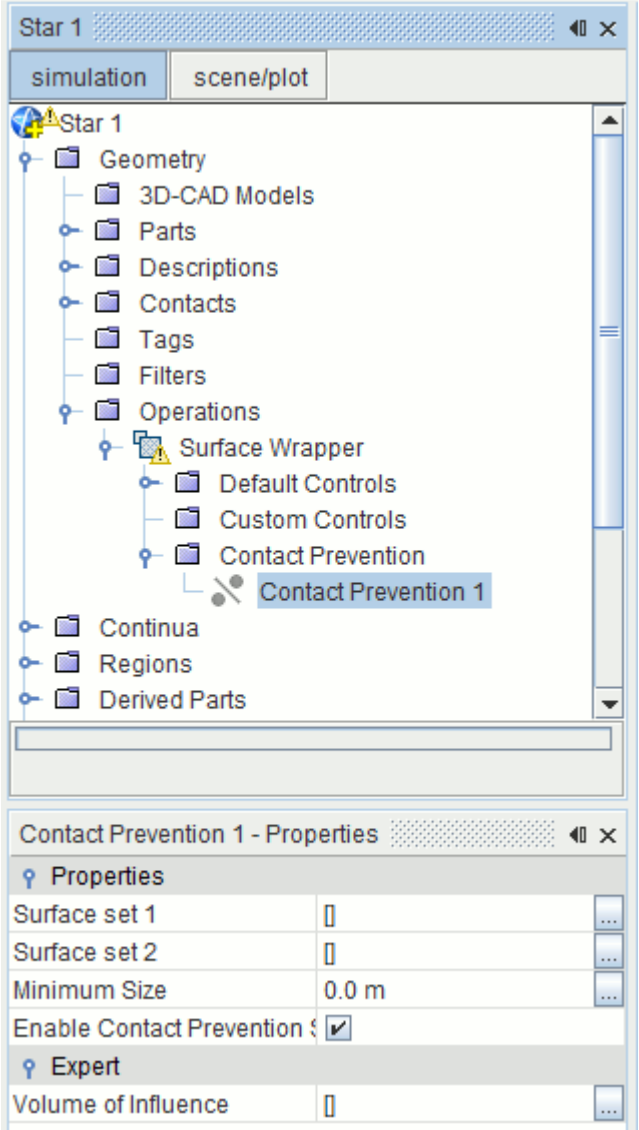
7. Once the selections are complete, click **Apply** to apply the changes, and then **Close** to return to the simulation tree.

Two Group Contact Prevention

Two Group Contact Prevention Set allows part surfaces in one group to form contact prevention sets with individual part surfaces in the other group, but not with part surfaces in their own group.

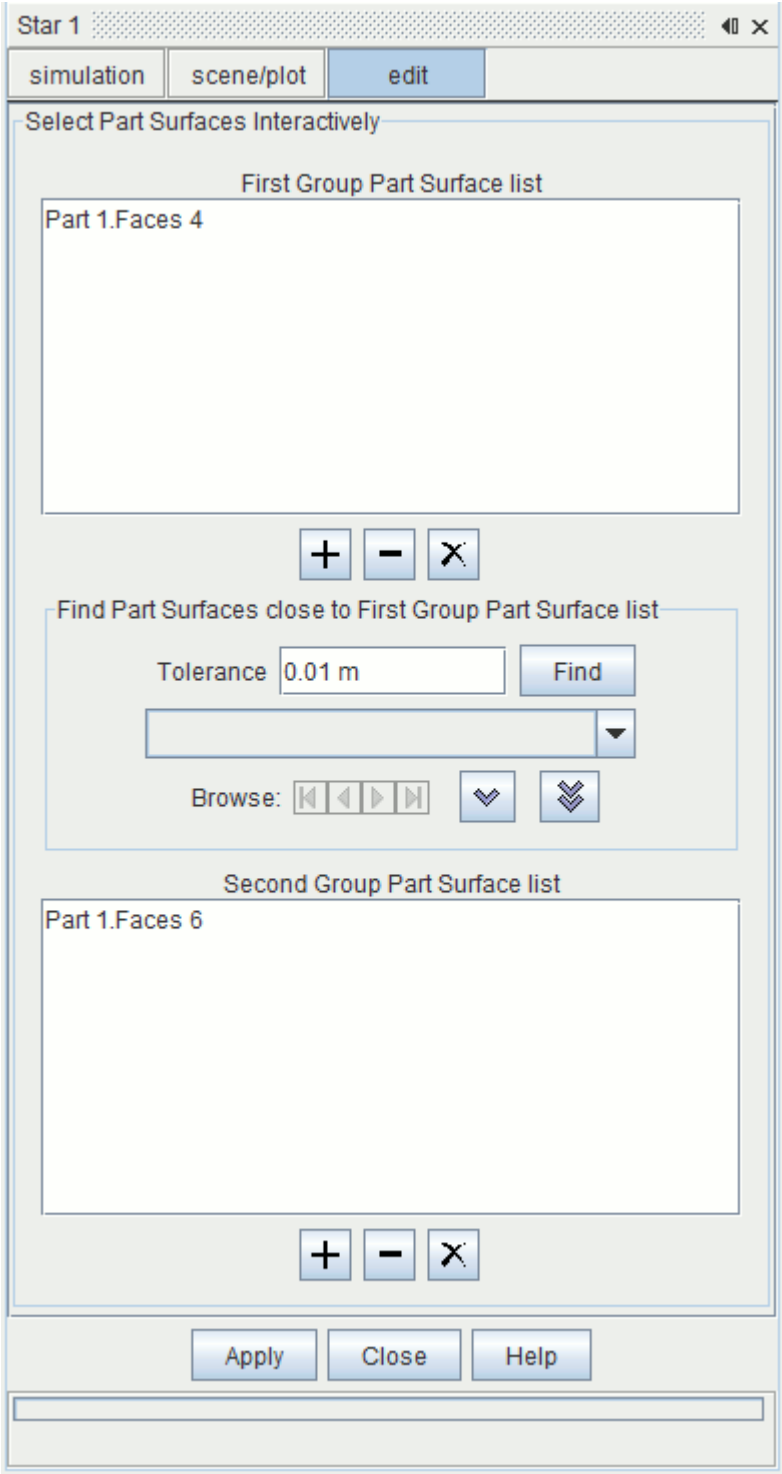
1. Selecting **Two Group Contact Prevention Set** creates a node that is called **Contact Prevention 1**.

   The properties of this node are shown below:



2. Part surfaces for each group are selected in the same way as for a single contact prevention set.

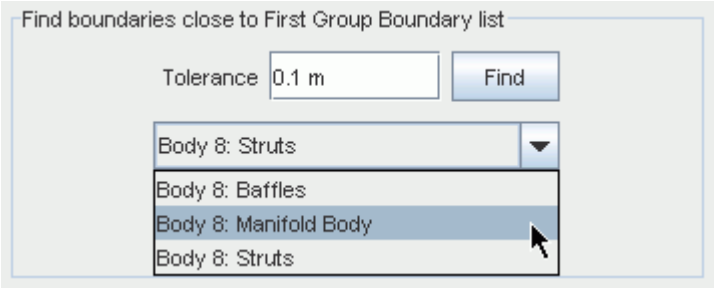
   If part surfaces are selected interactively, a list box for both the first and second group is provided:



3. Part surfaces can be added or removed from groups in the same way as for a single contact prevention set, but care should be taken that they are added to the correct group.
4. You can also use the *Close Part Surfaces* finder to identify part surfaces within a specified tolerance of the surfaces in the first group, and add them to the second group.
5. To use the *Close Part Surfaces* finder, first add one or more part surfaces to the first group.

6. Then enter the **Tolerance**, which specifies the search distance from the surfaces in the first group, and click **Find**.

   The tool finds all surfaces that have any part of their surface within the tolerance, and adds them to the drop-down menu.



   Browse buttons are available to inspect individually each part surface the tool finds.

7. To add a part surface to the second group, select the part surface from the drop-down menu and click . To add all part surfaces in the menu, click .

   Surfaces that are selected for the first group appear blue in the display, while surfaces in the second group appear orange.

8. Select at least one surface for both the first and second group in order for the contact prevention to be valid.

Minimum Size

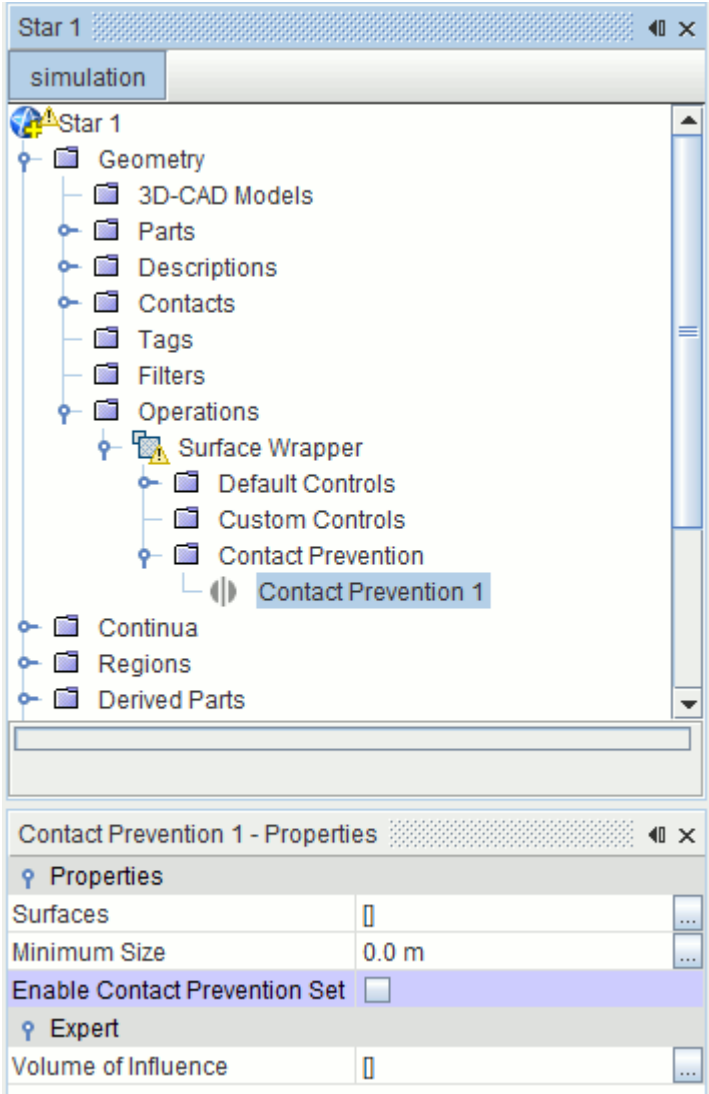
The *Minimum Size* is a minimum distance between the part surfaces at which contact prevention is guaranteed.

As a guideline, set this value to be approximately the largest distance between the surfaces that you want to keep separated.

Although contact prevention occurs if the distance between the surfaces is less than the search distances, it is not guaranteed to work in this case.

Enable Contact Prevention Set

A contact prevention set node can be temporarily deactivated, while still keeping all previously specified inputs, by using the *Enable Contact Prevention Set* property:



By default, the *Enable Contact Prevention Set* option is on.

1. Turning off the option deactivates the contact prevention node and deactivates the settings from any future surface wrapping operation.

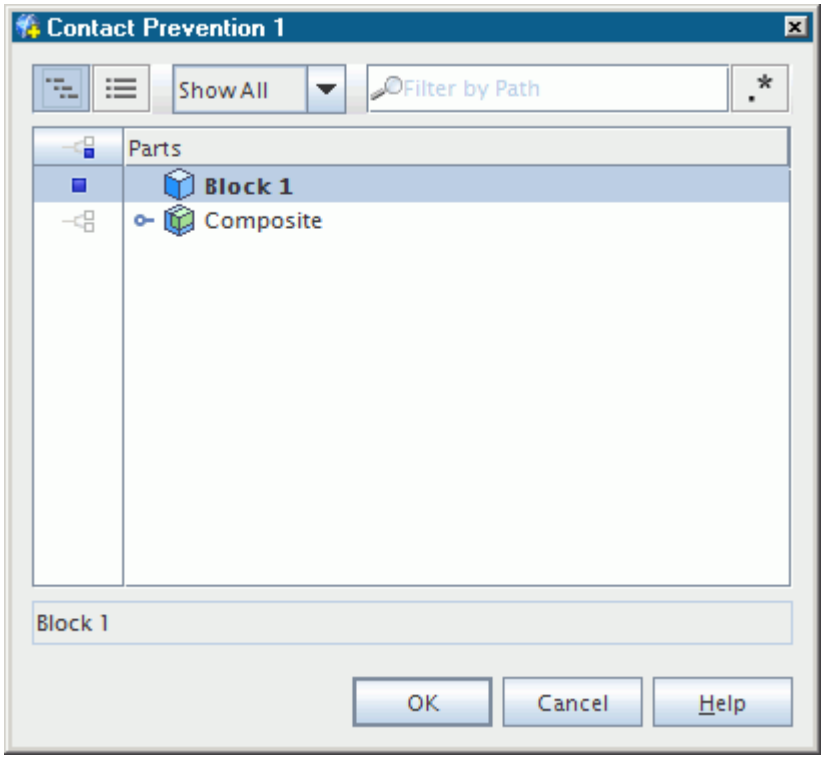
   The node icon that is associated with the set is also grayed out, indicating the change in status.

Using Volume of Influence to Limit Contact Prevention

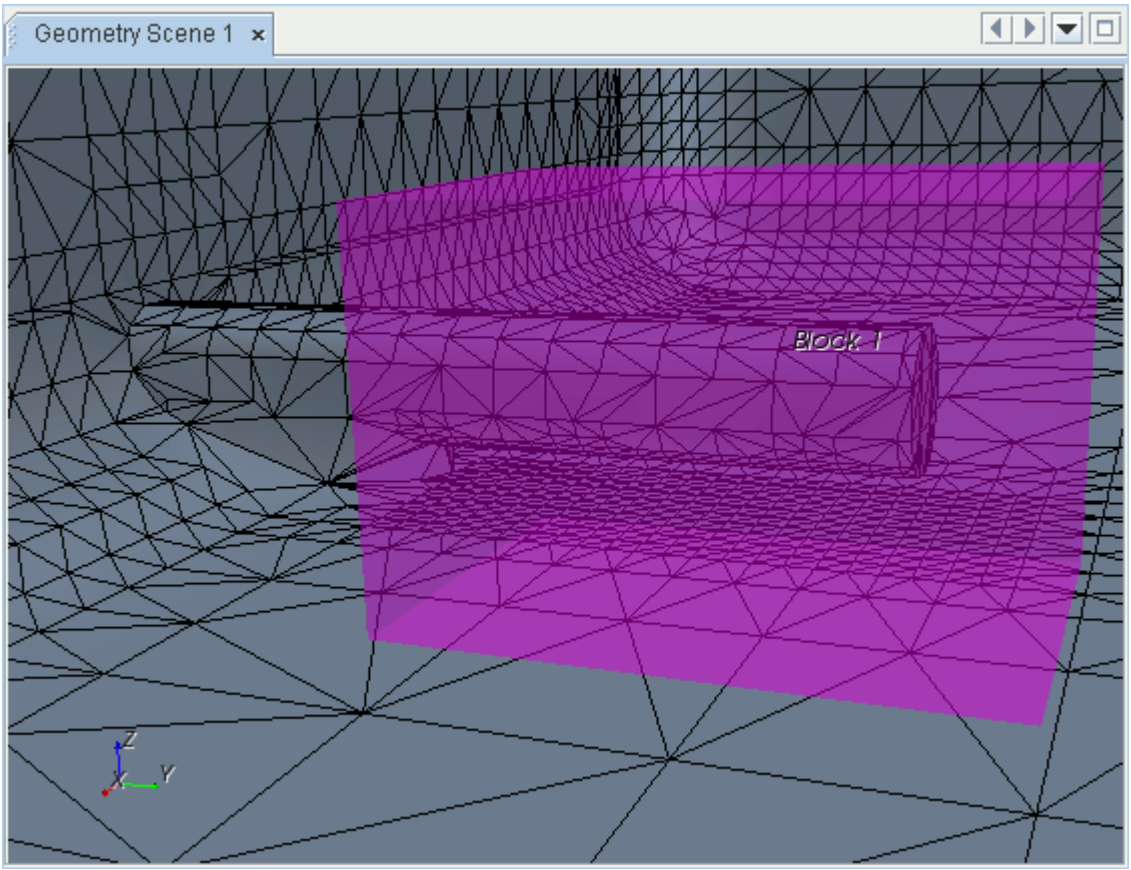
The expert property, *Volume of Influence* is used to define a volume within which contact prevention applies.

1. Click the ellipsis (**...**) to the right of the property to launch the *Volume of Influence* selection dialog.
2. Choose one or more parts that define the volume within which contact prevention is to apply and click **OK**.

   As an example, the dialog below selects **Block 1** to define the volume of influence.



When the surface wrapper runs, contact prevention is only applied to the area of participating surfaces that lie within the specified volume. The wrapper does not prevent contact for the remainder of these surfaces outside the specified volume. For this example, the volume of influence is applied to the contact prevention set created for the sensor boundary in the manifold surface wrapping tutorial. The resulting surface wrap is modified as shown below. The outline of the volume shape **Block 1** is also shown.



Highlighting Contact Prevention Part Surfaces

By default, the part surfaces that are set on a contact prevention set are highlighted in a scene when you select the node for the contact prevention set.

1. If you have deactivated this default highlighting, you can still choose to highlight the part surfaces by right-clicking on the contact prevention set node and choosing **Highlight** from the context menu.

This option is only available if a scene is showing in the *Graphics* window.

Surface Wrapper Defeature Control

Use the defeature controls of the surface wrapper to choose which features are removed, or more precisely, which features are not retained when wrapping geometry parts.

Without the defeature control capability of the wrapper, you would have to adjust the surface size to control feature retention, reducing the size where you want to capture features and increasing the size where geometric accuracy is not important. Due to many factors, such as contact prevention, geometric versus topological distance, etc., it is not always possible to increase the surface size in a specific portion of the model, which can lead to undesired results.

Using the defeature control, you can control what features are retained by creating control sets of part surfaces, between which feature retention does not occur. By defining the control sets, you are essentially choosing the location where the features are not retained.

There are two types of control sets:

- **One Group Wrapper Defeature Control** — Contains a single group of surfaces. Feature retention does not occur between these surfaces.
- **Two Group Wrapper Defeature Control** — Contains two groups of surfaces. Feature retention does not occur between the two groups.

The control sets in the defeature control are similar to contact prevention sets. See Contact Prevention.

To control which features are not retained during the wrap:

1. Create a defeature control:
   - For a single set of surfaces, right-click the **Surface Wrapper** > **Wrapper Defeature** node and select **New** > **One Group Wrapper Defeature Control**.
   - For two groups of surfaces, right-click the **Surface Wrapper** > **Wrapper Defeature** node and select **New** > **Two Group Wrapper Defeature Control**
2. Select the surfaces that define the defeature control set:
   - For a one group control, in the **Surfaces** property, select the surfaces between which you want to stop feature retention.
   - For a two group control, select a group of surfaces for the **Surface set 1** property, and a group of surfaces for the **Surface set 2** property. Feature retention is prevented between the two sets.

Surface Wrapper with Internal Baffles

The surface wrapping process automatically detects and inflates internal surfaces representing baffles.

The total thickness that is applied to each side of the baffle is equal to 0.25 times the smallest refinement size that is used in the surface wrapping process. This size is not necessarily equal to the minimum surface size specified in the **Default Controls** node (PBM) or **Reference Values** node (RBM). If you want to have more control over the thickness of the baffles, then it is recommended that you inflate the surface using the offset surface preparation tool before proceeding with surface wrapping.

Note that internal surfaces (by definition) are disconnected from the main geometry and are such subject to the smallest disconnected number of faces setting. By default, the surface wrapper automatically removes any isolated internal surfaces with less than 1000 faces (unless all surfaces have less than the specified number of faces in which case nothing is deleted). This process removes any small “artificial” surfaces that do not represent any geometry, which simplifies the volume meshing process. However, in some cases using a coarse surface wrap size can inadvertently coarsen an internal baffle surface such that it has less than 1000 faces. In such cases, consider reducing the value or deactivating it altogether by using a value of 0.

Surface Wrapper Properties

This section describes the properties that you can set for the Surface Wrapper.

The surface curvature and surface proximity properties are common to several meshers, and as such are documented in Surface Mesher Controls and Values.

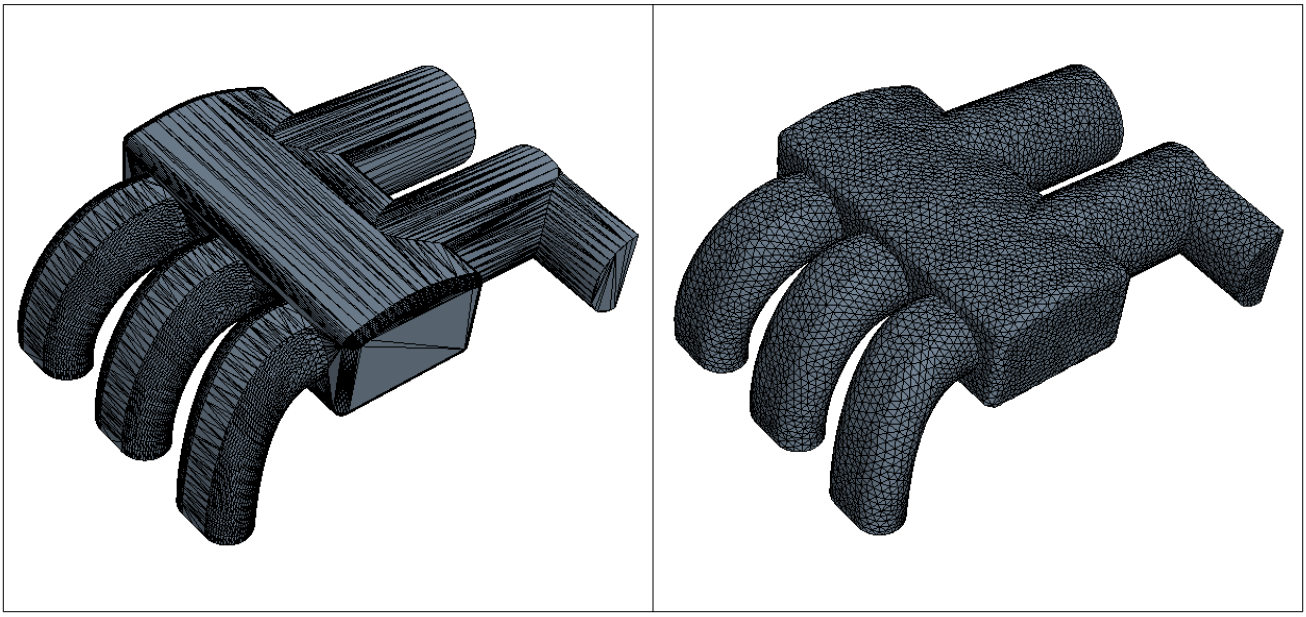
Per-Part Meshing (PBM only)	Activating **Per-Part Meshing** causes STAR-CCM+ to wrap each part individually. In this case, results are stored as a new description for each part. There are no new parts as is the case when wrapping all parts at once. Any interfaces previously defined will be ignored causing the mesh between connected parts to be non-conformal. This option is useful if the Trimmed Mesher or Thin Mesher can not produce a mesh at the interface. When the Per-Part Meshing box is deactivated all input parts will be wrapped at once and the wrapper results are stored in a new part. No new description is created. The equivalent functionality for Region Based Meshing is found under **Continua** > **Mesh**.
Input Parts (PBM only)	Specifies the geometry parts for which the operation generates a mesh. The equivalent Regions input for Region Based Meshing is found under **Continua** > **Mesh**.
Curvature Refinement	Activating Curvature Refinement increases the number of faces based on surface curvature or deviation distance.
Proximity Refinement	Activating Proximity Refinement increases the number of faces based on the distance to nearby surfaces.
Gap Closure	Activating Gap Closure can automatically close holes in the surface that are either smaller than a user-defined size, specified by seed points, or both. Do not use this option in tandem with the smallest wrapping volume specification, to prevent closure of internal geometry parts that have a small diameter.
Mesh Alignment	Selecting the **Perform Mesh Alignment** (PBM) / **Do Wrapper Mesh Alignment** (RBM) option allows the mesh alignment location values to be set. That is, a wrapper mesh line can be "pinned" to a given value in any of the X, Y and/or Z directions.
Create Weak Contacts With Input Parts (PBM only)	Activating the Create Weak Contacts With Input Parts option when the Per-Part Meshing option is deactivated creates new weak contacts. The weak contacts that are created do not result in a conformal mesh. However, when you assign parts to regions, these contacts are converted into interfaces which the solver then uses to make the mesh conformal.
Perform Partial Wrapping (PBM only)	Allows you to create a **Partial Wrap Set** to exclude surfaces which do not require surface wrapping from being wrapped in a Surface Wrapper operation. Surfaces that are excluded from wrapping in the Surface Wrapper operation are copied from the original geometry and fixed to the wrapped surfaces. See Setting the Volume of Interest and Wrap Properties.
Verbose Output (PBM only)	Activating Verbose Output prints more verbose output to the Output window during Meshing.

Surface Remesher

In order to improve the overall quality of an existing surface and optimize it for the volume mesh models, the surface remesher can be used to retriangulate the surface.

The remeshing is primarily based on a target edge length that you supply and can also include feature refinement that is based on curvature and surface proximity. Localized refinement that is based on part surfaces or boundaries can also be included. Specific surfaces or boundaries can also be omitted from the process so that the original triangulation from the imported mesh can be preserved.

The surface remesher is typically used for remeshing surfaces that are output from the surface wrapper and STL type data. As well as improving the surface for the volume meshers it also aids the subsurface generator when the prism mesher option is selected. An example starting surface and the remeshed surface is shown here.



The properties of the surface remesher model can be changed to provide additional meshing control during the meshing process. The input values used for the surface remesher can be set on four different levels:

- Global level
- Part surface or boundary level
- Feature curve level
- Interface level

A volumetric control can be included as part of the surface remeshing operation. This option allows the surface target size to be set for any of the allowable shapes that encompass the import surface.

Aligned meshing can also be included to optimize the resulting surface triangulation further.

Contents:

Aligned Meshing and Geometric Features
Surface Remesher with Interfaces
Surface Remesher Guidelines
Surface Remesher Properties

Surface Remesher Expert Properties

See Also:

Generating a Surface Mesh

Aligned Meshing and Geometric Features

In the surface remesher, aligned meshing is a fully automated process which creates a templated mesh on surfaces with four corners.

Aligned meshing improves the surface mesh representation for 4-sided surfaces, such as fillets and rounds, especially when they are meshed with a coarse mesh. Other benefits include quicker CAD projection and a more efficient use of faces on the surface mesh.

For flat surfaces, the aligned mesher only produces an aligned mesh when the cells that are produced are of higher quality than when using the regular remesher. For example, if the aligned mesher is likely to create poorly-shaped triangles, it automatically reverts to the basic remesher output for that CAD surface. For curved surfaces, an effort is made to align the mesh with the curvature. For this reason, STAR-CCM+ often favors the aligned mesher, while allowing for some degradation in the cell quality, as it captures the geometry more effectively.

In the **Surface Remesher** model expert properties, the **Create Aligned Meshes** option is activated by default.

This feature is fully active in these situations:

- Cases with CAD association, where the **Project to CAD** option is activated (default). The resulting meshes keep their CAD association.
- Cases with CAD association, where the **Project to CAD** option is deactivated. CAD association is lost, but aligned meshing is performed.
- Cases without CAD association of any kind, regardless of how you set the **Project to CAD** option.

You can generate aligned meshes on discrete surfaces. In this context, a discrete surface is defined as:

- Surfaces with no CAD.
- All surfaces in the surface remesher, if **Project to CAD** option is deactivated.
- Any surface which is partially or fully disassociated from CAD coming into the surface remesher.

Limitations of Aligned Meshing

Limitations of the aligned mesher for surfaces with CAD:

- It is possible that simulation files saved in STAR-CCM+ 5.06 or older do not contain sufficient information for aligned meshing and retaining geometric features. To overcome this issue, either reimport the CAD file or re-tessellate the part.
- The **Create Aligned Meshes** option works best with faces and vertices on surfaces that have 100% CAD association. If the surfaces have incomplete CAD maps, a hybrid projection is created. Local CAD projection is used when the information is available. When no information is available, a discrete projection is used back to the initial mesh.

- For Region Based Meshing (RBM), it is highly recommended that the *Retain Geometric Features* option remains activated. If you deactivate *Retain Geometric Features*, ensure that you mark the feature edges. If no feature edges are marked, the remesher has nothing with which to align the mesh.
- When using Parts Based Meshing (PBM), geometric features are retained automatically.
- You can use one or more feature edges as boundaries for aligned meshing, as long as there is only one underlying CAD surface.
- For aligned meshing to be able to work on a surface, the surface must have at least four CAD vertices, four corners, or four feature edges surrounding the area of interest.
- Surfaces that are directly on a periodic interface map are not aligned meshed, even when only one edge is shared. However, in-place maps between neighboring parts can be align meshed.
- There must be no inner loops on the surface.

Limitations of the aligned mesher on surfaces with no CAD:

- The limitations that apply to aligned meshing on surfaces with CAD also apply to aligned meshing on surfaces with no CAD.
- Aligned meshing supports 4-sided and non-4-sided rectangular surfaces, however these surfaces must have 4 distinct corners.
- Aligned meshing is based on patches. More specifically, any discrete region is a candidate for aligned meshing if this region is 4-sided and is surrounded by patch perimeters, part surface boundaries, and/or feature edges.
- Aligned meshing works for geometry models that have a combination of regions/surfaces that are associated with CAD and other regions/surfaces that are not associated with CAD. In such situations, surfaces that are associated to CAD are aligned and remain associated to CAD.
- Cylinders, and other closed surfaces that have 0 or 1 seams, are not aligned if they are discrete but are still supported if they have underlying CAD. To make sure that these surfaces are aligned, break them into at least 2 separate patches/surfaces — by default, STAR-CCM+ does this. You can break these surfaces into 2 or more surfaces by feature edges.

Surface Remesher with Interfaces

In order to achieve a conformal mapped mesh at the interface, the surface remesher can be used with specific direct type interfaces.

This recommended procedure results in a better volume mesh definition for the analysis.

For an in-place type interface, a conformal mesh can be obtained for both single region and multi-region cases, as long as the same mesh continuum is used. For example, in a single region case, an in-place interface can be used to represent an internal baffle of some kind. For multi-region cases, the interface can represent a change of material, such as a fluid/porous interface or fluid/solid of some kind. Additionally, the surface remesher can now be used regardless of the perimeter shape of the boundaries at the interface as long as the interface boundaries are planar. In other words, the boundaries belonging to the interface do not need to have the same shape or area. An imprint method is automatically included as part of the surface remeshing process, meaning that the boundaries are first retriangulated to have a common perimeter before proceeding with the surface remeshing step.

When the surface remesher is used with a periodic interface, then it attempts to produce a conformal mapped mesh for the boundary pair. Both boundaries have to belong to the same region in this instance.

When using one of the above direct type interfaces for surface remeshing, then the interface node controls the settings and sizes that are common to both boundaries. Where remainder boundaries exist for in-place type interfaces (that is, those that belong to a boundary that is part of an interface but is not common to the boundary pair), the normal boundary node settings and sizes are used instead.

When visualizing the remeshed surface, include the interface boundary nodes that are connected to the boundary pair in the displayer parts selection so that the mesh triangulation can be seen on those faces. The interface boundary surfaces are also colored according to their type.

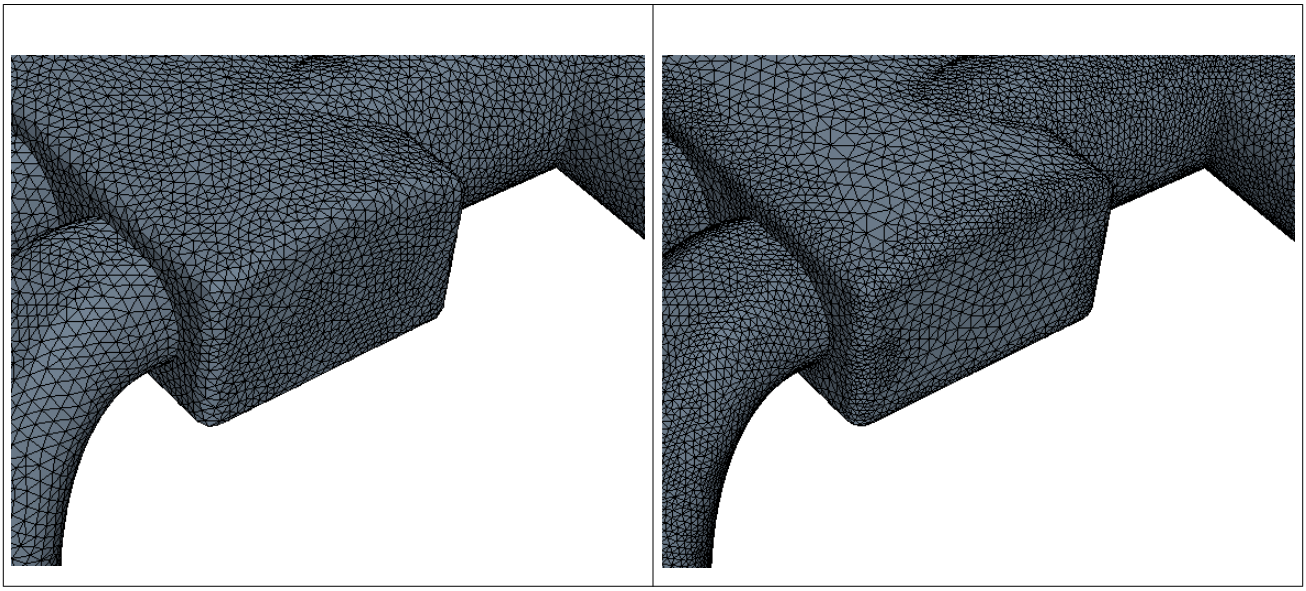
When using the repeating type interface, a conformal mesh interface is not required because the meshes in each region are sliding past each other. Here the per-region meshing option must be activated in order to ensure a good quality surface at the interface.

Additional notes on surface remeshing with interfaces are provided in the guidelines section.

Surface Remesher Guidelines

This section provides guidelines for using the Surface Remesher.

- It is not necessary for the import surface that the surface remesher acts upon to be closed. However, the surface remesher does not close this surface for you - where free edges occur in the starting surface, an equivalent free edge is present in the remeshed surface.
- Feature curves must be added to the import surface if you wish to maintain sharp edges.
- Internal feature surfaces such as baffles can be included in the import surface. Convert these surfaces to an interface before surface remeshing.
- In cases where proximity refinement is desired but the starting triangulation is very coarse (for example, STL data), a uniform triangulation in the proximity area may not be achieved. In this case, perform a surface remesh without proximity effects using an intermediate triangulation density, delete the import representation and rerun the surface remesher again with proximity effects included.
- With the surface curvature option turned off, a uniform surface triangulation that is based on the target surface size is obtained. However, it is recommended that you use surface curvature for most cases as the resultant surface definition is far superior to the definition obtained using uniform triangulation. An example comparison is shown here.



- When using in-place interfaces in a multi-region case, where the perimeter of the two interface boundaries does not match (leading to remainder boundaries), the imprinting process can fail.

  If this scenario occurs, try one of the following:

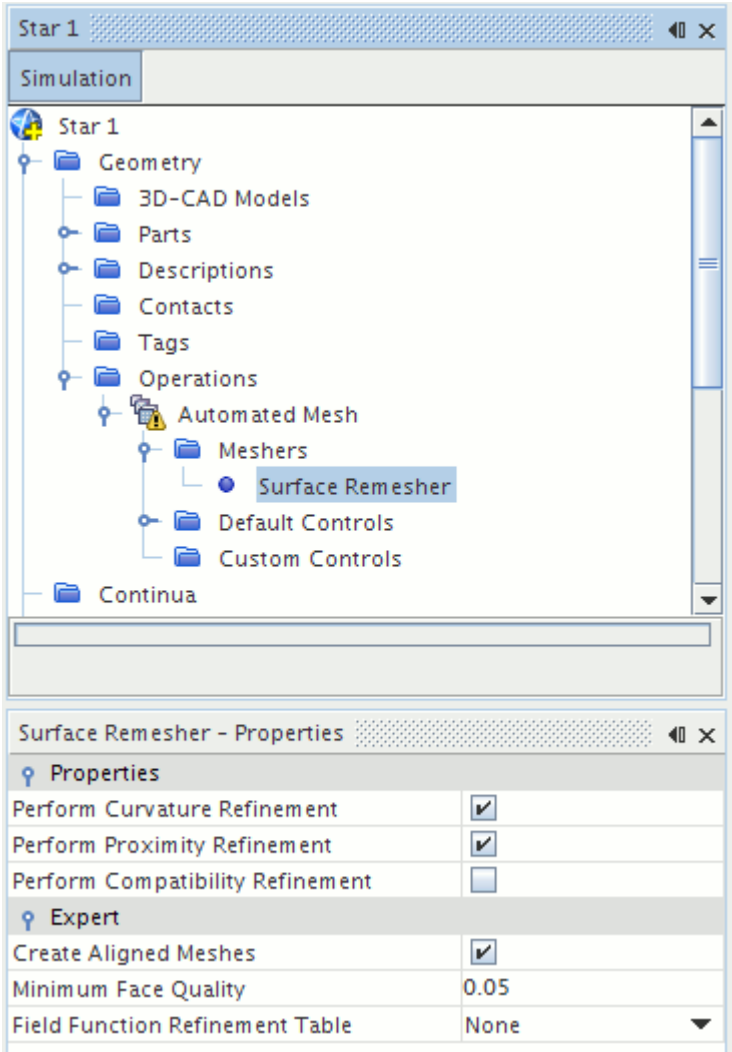
  - Remesh one or both boundaries separately to improve the surface triangulation in these areas before attempting to match them. To remesh the boundaries separately, either use surface remeshing with per-region meshing turned on, or use the remesh selected faces option for surface repair.
  - Reverse the order of the interface boundaries. Delete the interface and then select the two boundaries in the opposite order to that used for the original one before making a new interface.
  - When matching a curved surface to a planar one (for example, a sphere touching the side of a box), project part of the curved surface onto the planar surface first. A common area is then formed between the two surfaces that allows an interface to be created.
- As with all the meshing tools in STAR-CCM+, the surface remesher is subject to continual improvement from release to release in order to increase the overall quality of the surface triangulation produced. As such, an increase in the resultant face and vertex count can be seen for the same geometry when run in two different versions.

  A typical example of this situation is seen in highly curved geometries, where a better definition is achieved in the newer version than the old. The increase in face count can sometimes be undesirable (especially if there is a substantial increase). In this instance, relaxing the surface curvature requirement somewhat helps to reduce the overall face count. Depending on the geometry, further adjustments in user-specified sizes can be required in order to reproduce similar face counts (albeit with a higher quality) when comparing the newer version to the old.

Surface Remesher Properties

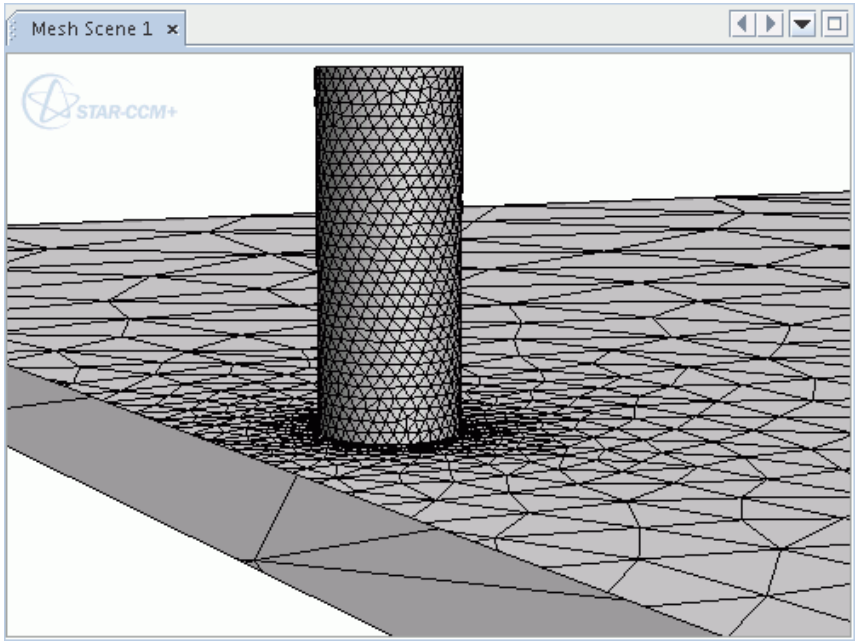
This section describes the properties that you can set for the Surface Remesher.

When the surface remesher has been activated, the properties are found under **Automated Mesh** > **Meshers** > **Surface Remesher** when using Parts Based Meshing (PBM), and under **Continua** > **Mesh** > **Models** > **Surface Remesher** when using Region Based Meshing (RBM).

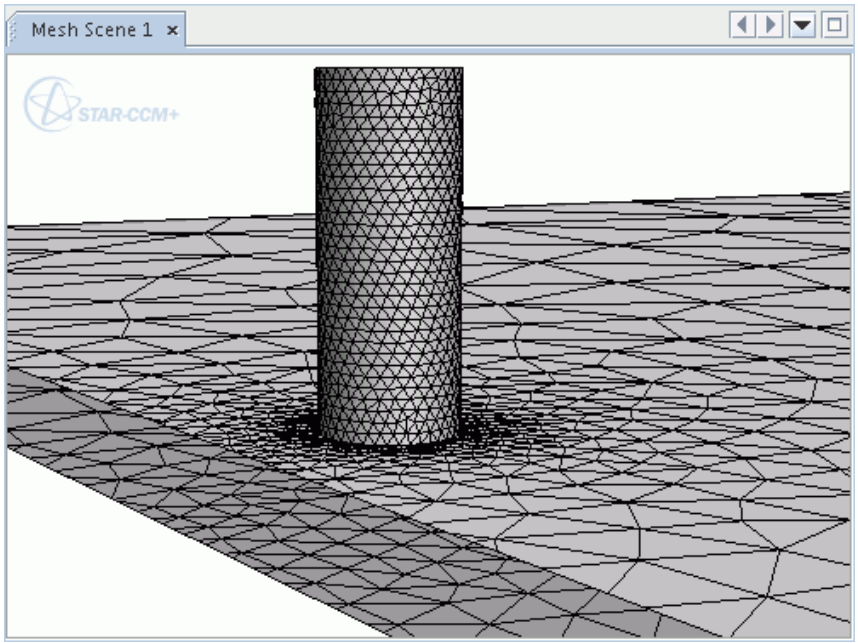


The surface curvature and surface proximity properties are common to several meshers, and as such are documented with the common surface mesher controls and values. See Surface Mesher Controls and Values.

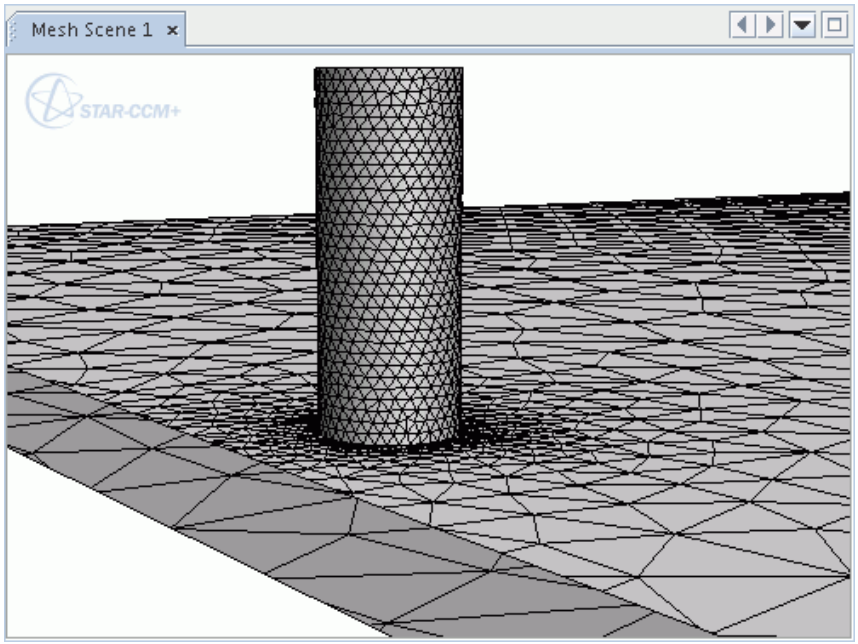
Perform Compatibility Refinement (PBM) / Do Compatibility Refinement (RBM)	Activating this option imposes a surface growth rate of 2 to limit the difference in face sizes across a gap. Therefore, if the gap between faces is small, the relative size of faces on either side of the gap is close to one another. This option is useful when using the thin mesher, and improves volume mesh quality, as face sizes on opposing sides are similar without explicitly stating a size. This option is similar to proximity refinement. Proximity refinement specifies a mesh size that is based on the gap size and the number of points inside the gap. Whereas, compatibility refinement ensures that the mesh size on either side of the gap is similar. In some cases, this results in fewer elements and a more efficient use of the surface remesher. For example, consider the surface mesh below. The curvature of the cylindrical pin reduces the mesh size near the pin:



Applying compatibility refinement has the following effect on the surface mesh opposite the cylindrical pin:



As can be seen above, the surface mesh in the region opposite to the cylindrical pin is refined, as the compatibility refinement makes the mesh size across the gap similar. Note that the surface growth rate also influences the mesh size, so the sizes are not identical. The refinement on the surface mesh in turn affects the volume mesh by creating refined

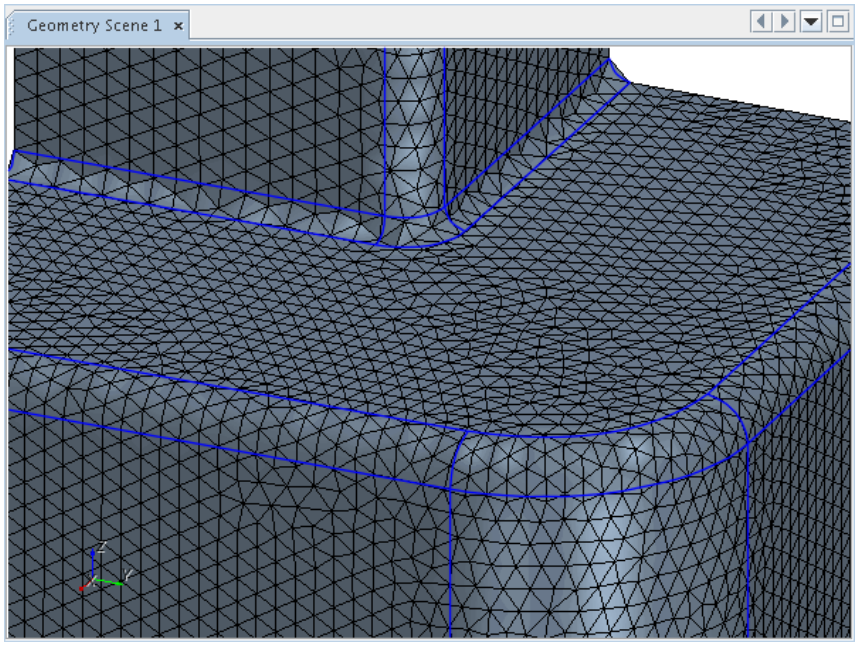
	and better quality cells in the gap between the pin and opposing surface. For comparison, the following image shows the surface mesh with only proximity refinement activated.
	 The mesh size on the opposing surface is independent to the cylindrical pin and the region is only refined due to the gap thickness.

Surface Remesher Expert Properties

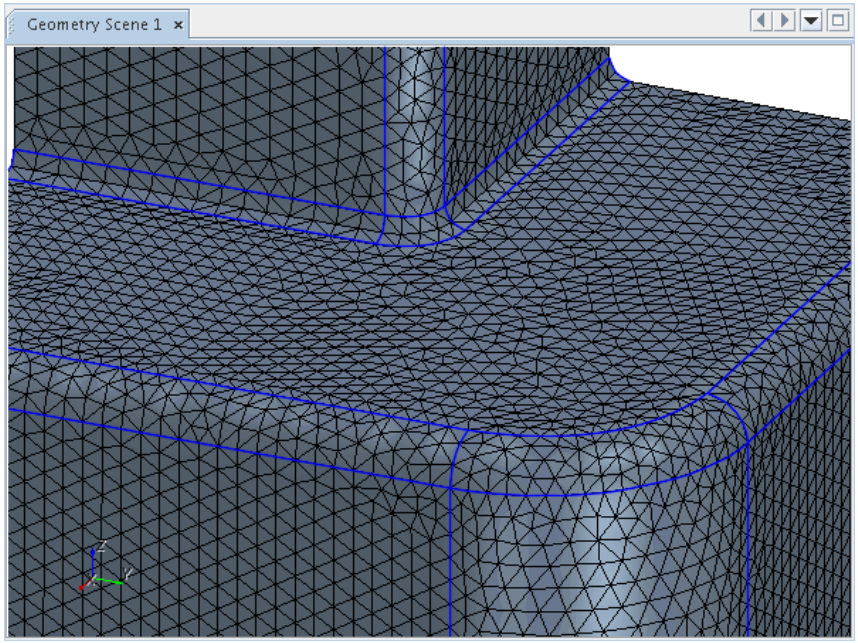
This section describes the expert properties that you can set for the Surface Remesher.

Retain Geometric Features (RBM)	Retaining geometric features aims to preserve all CAD edges when generating a surface mesh, which provides a better representation of the original CAD model. This option also makes the CAD projection process more efficient, as faces on the surface mesh do not span multiple CAD surfaces. The *Retain geometric features* option is a property of the **Surface Remesher** meshing model when using Region Based Meshing (RBM), and is activated by default. However, if geometric features cause poor face quality, the surface remesher is free to ignore them. For Parts Based Meshing (PBM), the equivalent *Retain geometric features* property is not available, as it is activated automatically. The following examples illustrate the effect of the *Retain geometric features* option. The blue sections highlight CAD edges for comparison

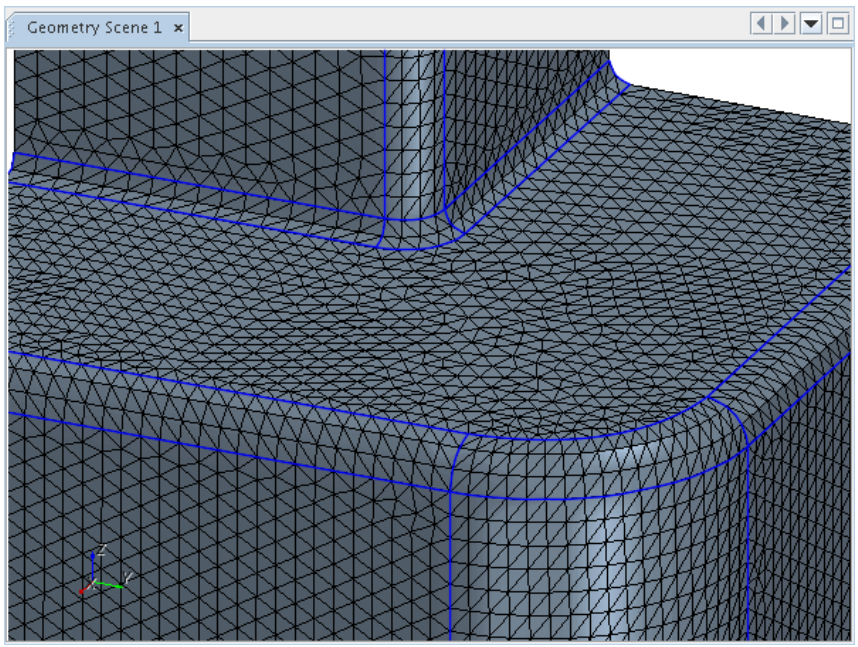
between the two surface meshes. The first image shows a surface mesh with *Retain geometric features* deactivated.

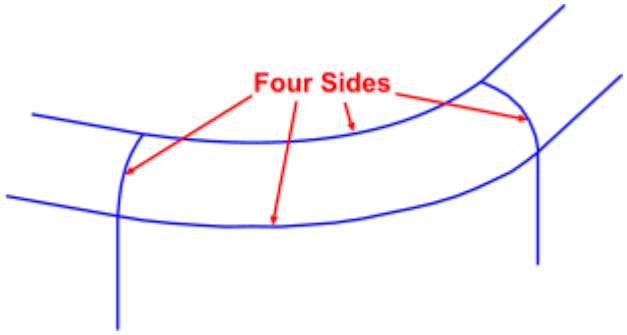


The second image shows a surface mesh with *Retain geometric features* activated.

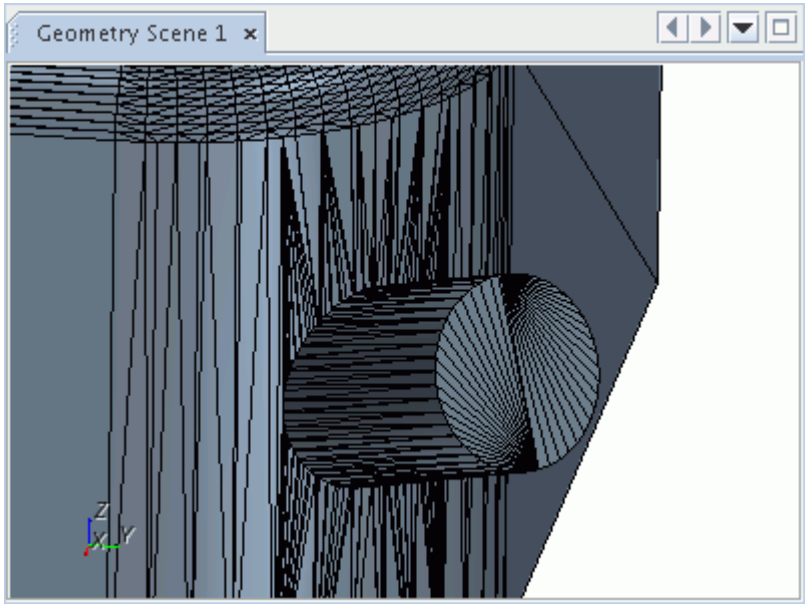


You can identify which edges have CAD association by using the CAD Association option in the *Select Threshold* tool.

Create Aligned Meshes	Activating *Create aligned mesh* generates a surface mesh with face edges that aim to follow the curvature of rounded CAD features, such as fillets. The following issues can prevent the surface remesher from generating an aligned mesh: • An aligned mesh reduces the overall surface mesh quality. • The aligned meshing conditions are not met. • The underlying geometry is not suitable for aligned meshing. The *Create aligned mesh* option is a property of the **Surface Remesher** meshing model, and is activated by default. Use the checkbox to deactivate aligned meshing. The following example illustrates the effect of the *Create aligned mesh* option. The blue sections highlight CAD edges for comparison between this surface mesh and those in the previous section. 
	Currently, the following conditions apply to the *Create aligned meshing* option: • Aligned sections must have 4 sides and have geometric features as closed boundaries, for example:

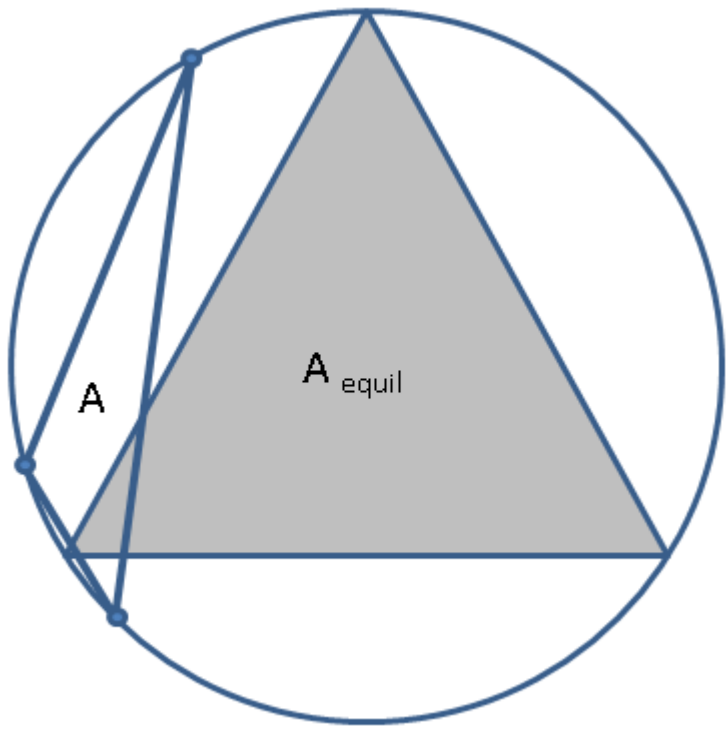


- There must be less than a 30% difference between the number of vertices that the surface remesher creates on opposing sides of an aligned section.
- The remeshed surface must have a relatively uniform surface mesh size. As a general guideline, the minimum surface size (without aligned meshing) must be greater than half of the maximum surface size.
- Aligned meshes cannot lie on a periodic interface boundary, or be connected to one along an edge (in-place interfaces are supported).
- Aligned sections cannot contain inner loops, such as an inlet duct on a filleted edge:



Minimum Face Quality

The *Minimum Face Quality* property for the surface remesher is a value that ranges from 0 to 1, with 0 being the worst and 1 being perfect. The quality of a triangle is calculated as the ratio of the triangle face area to the area of an equilateral triangle that would exactly fit inside the circumcircle of the triangle. The circumcircle is the circle which touches

	all three corner points of the original face. In the diagram below, the Minimum Face Quality is defined as: $$\text{Minimum Face Quality} = A/A_{equil}$$  The default value is `0.05`, and it can be changed with a slider. The surface remesher ensures that all the triangles in the resultant surface mesh have a quality equal to or greater than the specified value. This minimum quality is achieved by allowing the mesher to disregard feature edges, move feature vertices, or avoid projecting vertices if necessary. As a result, the surface remesher removes or improves faces whose quality is below the specified value. Setting the value to `0` deactivates the option. Recommended values are in the range `0.01` to `0.1`. Values cannot exceed 0.1.
Choose Field Function Refinement Table (PBM) / Choose a Mesh Size Table (RBM)	The **Choose Field Function Refinement Table / Choose a Mesh Size Table** property allows for the refinement of the mesh using field functions, based on an existing solution.

Automatic Surface Repair

The automatic surface repair tool provides an automatic procedure for correcting a range of geometric type problems that can exist in the remeshed surface once the surface remeshing process is complete.

The auto repair option is typically used when performing surface remeshing on a wrapped surface.

Up to three different metrics are used to determine if automatic fixes are required:

- Pierced Faces (surface intersections including opposing intersections)
- Surface Proximity
- Surface Quality

The metric for pierced faces is always on and is invoked when the automatic surface repair option is activated. The metric for surface proximity is only used if the surface proximity property has been activated for the surface remesher. The metric for minimum quality is only used when the surface remesher has a minimum face quality value of greater than 0.

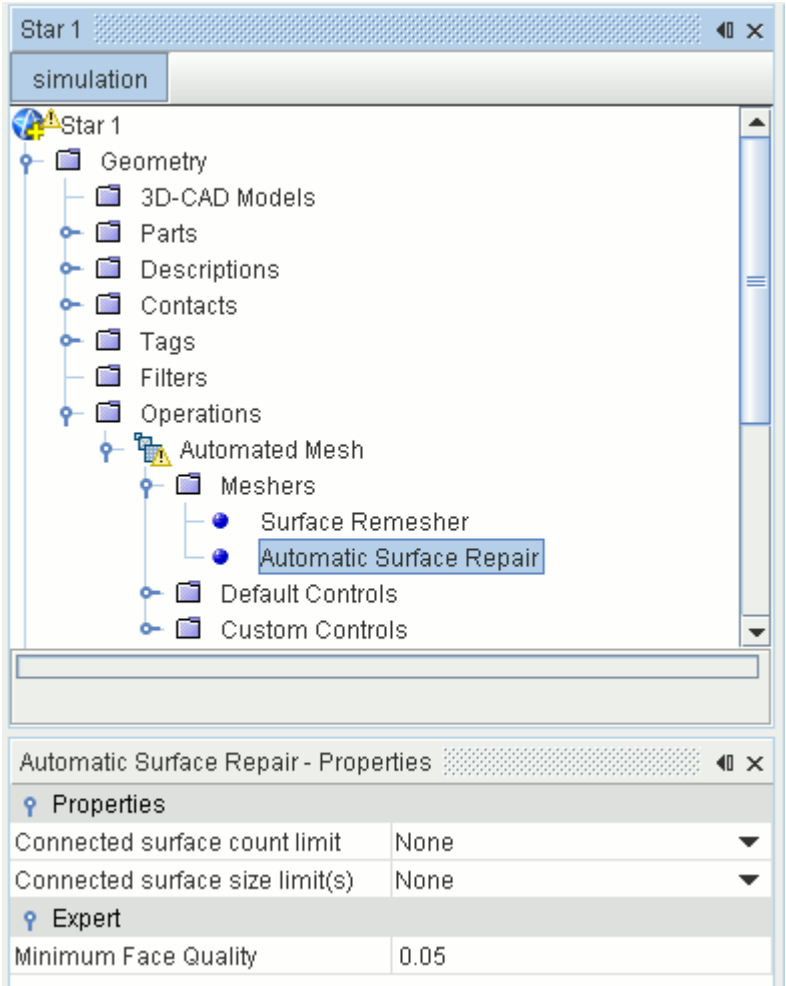
Within each metric, two different fixing mechanisms are used, arranged in best to worst order in terms of preserving the original geometry. The fixing mechanisms are:

1. **Remesh** - Grows out the problem area and then surface remeshes, removing features if necessary.
2. **Patch** - Grows out the problem area, deletes the local surface, and then invokes the hole filler.

During the auto repair fixing operation, it is possible for small disconnected parts of the surface mesh to be formed as a by-product. Two options are provided to allow you to delete these disconnected surfaces automatically:

- Connected surface count limit
- Connected surface size limit

These options can be accessed by selecting the **Reference Values** > **Automatic Surface Repair** (RBM) or **Meshers** > **Automatic Surface Repair** (PBM) node:



The connected surface count limit can be used to control the number of individual isolated or disconnected surfaces. For example, if you are only expecting to have one surface at the end of the resurfacing process, then set a value of 1. The surface with the largest face count or area is retained while all others are eliminated. Alternatively or in addition, the connected surface size limit can be used to specify the minimum face count and/or area that each disconnected surface requires. The above two options can be used separately or in tandem in order to control the resultant surface.

The auto repair option does not fix topological problems such as free edges, non-manifold edges, and non-manifold vertices. Therefore, if these problems exist in the surface, they must be fixed by hand afterwards using the manual repair tools.

Contents:

Automatic Surface Repair Properties

Automatic Surface Repair Properties

This section describes the properties that you can set for the Automatic Surface Repair mesher.

Connected Surface Count Limit	The connected surface count limit provides a method of controlling which connected surfaces are kept (or conversely, which disconnected surfaces are deleted) based on the total number of connected/disconnected surfaces. For example, if you are expecting to retain two surfaces at the end of the surface remeshing and auto repair process, then a limit of 2 can be specified. The two surfaces with the largest face count or area are then retained and all other surfaces deleted. The option to determine whether to use the face count or area specification is set using the *Connected surface count limit* property in the **Automatic Surface Repair** node. By default, the **None** option is specified indicating that the feature is turned off. Selecting either the **Keep Largest Face Counts** or **Keep Largest Areas** option activates the feature and activates the **Maximum # of Connected Surfaces** node: • **Default Controls** > **Maximum # of Connected Surfaces** (PBM) • **Reference Values** > **Automatic Surface Repair** > **Maximum # of Connected Surfaces** (RBM) The default value of `1e9` effectively means that all surfaces are retained by default. Entering a value of `1` for example would specify that only one surface is retained after the auto repair process. The *Maximum # of Connected Surfaces* property can also be set with a slider.
Connected Surface Size Limits	The connected surface size limit provides a method of controlling which connected surfaces are kept based on either the number of faces in the surface and/or the area of the surface. For example, if you only want to maintain surfaces with more than 100 faces, then a minimum limit of 100 faces can be specified. The auto repair process deletes any resultant surface with 100 or less faces in it. By default, the **None** option is specified indicating that the feature is turned off. If either of the face count based options are selected, the **Minimum Connected Surface Face Count** node is activated: • **Default Controls** > **Minimum Connected Surface Face Count** (PBM) • **Reference Values** > **Automatic Surface Repair** > **Minimum Connected Surface Face Count** (RBM) The default value for *Minimum Connected Surface Face Count* is 0, meaning that all surfaces are retained. If a value of 1000 is entered for example, then the auto repair process automatically removes any disconnected surface with 1000 or less faces in it. If either of the area-based options are selected, then the **Minimum Connected Surface Area** node is activated: • **Default Controls** > **Minimum Connected Surface Area** (PBM)

	• **Reference Values** > **Automatic Surface Repair** > **Minimum Connected Surface Area** (RBM) The default value for *Minimum Connected Surface Area* is 0, meaning that all surfaces are retained. If a value of 0.5 m^2 is entered for example, then the auto repair process automatically removes any disconnected surface with an area of less than 0.5 m^2. The value can be specified in any of the allowable units. If both the area and face based limits are selected, both nodes (as discussed above) are activated and both an area and face based limit can be applied. In this instance, the disconnected surfaces are retained if they meet both criteria.

Expert Properties

Minimum Face Quality	The equivalent property for Region Based Meshing is found as a node under **Reference Values** > **Automatic Surface Repair**. The *Minimum Face Quality* property for the auto surface repair is a value that ranges from 0 to 1, with 0 being the worst and 1 being perfect. The quality of a triangle is defined as 2*(r/R). R is the radius of the circle that passes through the three corner points of the triangle and r is the radius of the circle that fits inside the triangle. The default value is `0.05`. The auto surface repair ensures that all triangles that the process produces have a quality equal to or better than the specified value. The mesher disregards feature edges where necessary, moves feature vertices, or avoids projecting vertices in order to remove or improve faces whose quality is below the specified value. Setting the value to `0` deactivates the option. Recommended values are in the range `0.01` to `0.1`. Values cannot exceed 0.1.

Surface Mesher Controls and Values

This section details the controls and values for all surface meshers.

Alignment Location

This control aligns the wrapper mesh lines relative to the global cartesian system. That is, you can direct the mesh lines along the x-, y- and/or z-direction.

This facility is useful when wrapping multiple regions; and the alignment of each wrapped mesh to its neighbor is important for merging and interfacing purposes.

Alignment Location Properties

Location	Specifies the location for aligning the wrapper mesh lines. The current scene highlights the current location of the mesh alignment point with a circular ball.

Automatic Surface Repair

See Automatic Surface Repair

Base Size

The Base Size is a characteristic dimension of the model that you set before using any relative values. As general examples, you can set the base size to the diameter of an inlet, the length of the fluid volume, or a size that is convenient for scaling other values.

Basic Curvature

This value specifies the number of points around a circle on the surface mesh, which can refine the mesh on surface curvature. Increasing the value increases the relative refinement of the cells next to the curved surface.

The following diagram illustrates this value.

For pipes or tubes that have a relatively small diameter when compared to the rest of the mesh region, decrease this value to avoid over-refinement of the area.

Basic Curvature Properties

# Pts/circle	Specifies the number of triangles (for a surface) or cells (for a volume) are used to define a 360 degree cylindrical surface. Any other types of cylindrical surface use an equivalent number of triangles. For example, a half circle use half of this value and an ellipse would use more faces on tighter curvature. When used with the curvature deviation distance, this property is the minimum number of edges that are used around a complete circle.

CAD Projection

This control allows the mesher to project vertices back to the imported CAD surface during surface wrapping or surface remeshing. This capability results in a surface definition that is closer to the CAD geometry.

CAD projection is only available for the following CAD file types:

- Native CAD Formats
- Parasolid Transmit
- IGES
- STEP

- VDA

Projection back to an underlying CAD edge is possible only for feature edges that the surface import process defined.

For the wrapper, you can expect 20% or more CPU time for CAD projection. For the remesher, you can expect 300% or more CPU time.

Note: The surface wrapper cannot project some vertices, such as areas of gap closure or non-CAD data. The *Output* window provides a summary of the projected vertices at the end of the surface wrapping process.

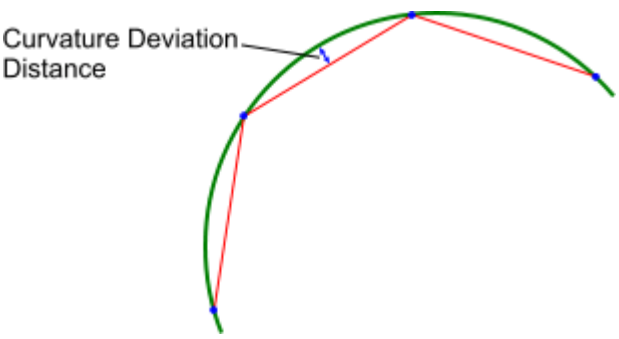
CAD Projection Properties

Project to CAD	**Activated**	Projects the vertices to the original CAD surface.
	Deactivated	Does not project the vertices to the original CAD surface.

Curvature Deviation

The surface wrapper, surface remesher, and trimmed cell mesher use this distance to populate the surface mesh with an appropriate number of edges. This means that the diameter of a curve is accounted for in the mesh distribution.

An illustration of this distance is shown below:



To estimate an initial curvature deviation distance, measure the larger final mesh sizes on curved areas and multiply that distance by between 0.1 and 0.01.

Curvature Deviation Properties

Max # Pts/circle	Upper limit on the number of points that are permitted on a complete circle.
Curvature deviation distance	Maximum distance that is permitted between the center of any mesh edge and its associated input surface.

Edge Proximity

Edge Proximity allows you to specify a target minimum number of mesh faces that you require between opposing perimeter edges on a remeshed part surface.

To apply edge proximity refinement, STAR-CCM+ first calculates the distance between opposite perimeter edges on the input part surface. This distance is divided by the specified target number of faces. The result is then set as a target size for faces on the part surface. Assuming that this target size is greater than the Minimum Size, and less than the current target mesh size, the triangulation on the part surface shows refinement after remeshing.

If one part surface defines an entire geometry part, applying edge proximity has no effect since there are no surface edges present. You can either:

- Split the part surface appropriately and apply edge proximity to the required part surfaces.
- Use the Include Feature Edges option.

Edge Proximity is only available as a custom control for an automated mesh operation and is deactivated by default.

Edge Proximity Properties

Number of Faces	The target number of mesh faces on the surface between each given boundary.	
Enable Search Ceiling	Allows you to specify a Search Ceiling distance. Surfaces with a proximity distance that is larger than this value are excluded from processing to avoid unnecessary refinement. Deactivated by default.	
Search Floor	Surfaces with a proximity distance that is smaller than this value are excluded from processing. Therefore, unnecessary refinement is avoided across very thin patches or part surfaces and in sharp or tapered corners. Set to 0.0 (deactivated) by default.	
Include Feature Edges	Determines if feature edges are added to the collection of edges for edge proximity computation. By default this is deactivated and part surface edges are used for edge proximity computation.	
	Activated	Feature edges are added to the collection of edges that are used for edge proximity computation.
	Deactivated	Feature edges are not added to the collection of edges that are used for edge proximity computation.
Include Patch Perimeters	Determines if patch perimeter edges are added to the collection of edges for edge proximity computation. By default this is deactivated and part surface edges are used for edge proximity computation.	
	Activated	Patch perimeter edges are added to the collection of edges for edge proximity computation.

	Deactivated	Patch perimeter edges are not added to the collection of edges for edge proximity computation.

Feature Angle

This value determines the surface angle where the wrapper generates a feature edge. Use this feature specification instead of generating feature curves in the input surface.

The surface wrapper produces a part curve (PBM)/feature curve (RBM) that is called **Edges from Wrapping** that contains any edges that did not exist in the starting import surface. In parts based meshing, the surface wrapper creates a tag (called **XYZ**) on the **Edges from Wrapping** part curve to differentiate the part curve that the wrapper creates from the pre-existing part curves.

Note: The wrapped surface retains any existing part curves (PBM)/feature curves (RBM) that you previously defined. However, they can be modified according to the new wrapped surface.

Feature Angle Properties

Angle	Specifies the minimum angle between faces that the surface wrapper defines as a feature.

Gap Closure Size

This value controls the filling of holes in the import surface during the surface wrapping process. You can specify the gap closure size as relative or absolute.

To close a hole, the specified gap closure size value must be bigger than the actual hole size. That is, the surface wrapper does not close holes with a dimension that is larger than the specified size.

The gap closure size cannot be smaller than the target surface size; the surface wrapper uses the larger of the two values to close gaps. This rule is also true when you use custom boundary values. This behavior prevents the surface wrapper from refining all of the surface down to the size of the smallest gap size.

Gap closure size works with the smallest wrapping volume option. The surface wrapper only closes gaps on surfaces that enclose a volume that is larger than the specified volume.

The surface wrapper creates a surface that is called **Gap Closure Faces**, which contains the faces that fill the hole.

Minimum Proximity

The minimum proximity specifies the minimum allowable distance between faces. Proximity is defined as a percentage of the average edge length of triangle that is checked.

Note: The default value of 0.05 is suitable for most cases. This means that any two faces cannot be closer than 5% of the average edge length for the initial face.

Minimum Proximity Properties

Minimum Proximity	Minimum allowable distance between two faces before STAR-CCM+ invokes automatic surface repair. A value of zero deactivates this property.

Minimum Quality

The minimum quality specifies the minimum allowable face quality.

Minimum Quality Properties

	Minimum allowable face quality before STAR-CCM+ invokes automatic surface repair. A value of zero effectively deactivates this property.

Minimum Surface Size

This value specifies the lower limit of edge lengths on the surface mesh. You can specify the minimum surface size as relative or absolute.

Decreasing the minimum surface size allows more refinement in regions where the mesher is trying to capture complex features. However, this refinement increases the CPU time and size of the surface mesh.

Maximum Surface Size

This value specifies the upper limit of edge lengths on the surface mesh. You can specify the maximum surface size as relative or absolute.

Increasing the maximum surface size reduces the number of faces in regions where there is little complexity on the surface. However, this coarsening can ignore features that you want to capture in the surface mesh.

Part Priority

This control is only available when the Automated Mesh operation property *Mesher Execution Mode* is set to **Concurrent**, *Per-Part Meshing* is activated, and the simulation is opened on at least three processors.

Sort Criteria	Allows you to distinguish which parts are prioritized when meshing concurrently.	
	Default	Parts are sorted in the order of creation.
	Surface Area	Parts are meshed in order from largest to smallest surface area. The surface areas are computed from the root descriptions of each part.
	Volume	Parts are meshed in order from largest to smallest volume. The volumes are computed from the root descriptions of each part. The absolute value of the volume is used, so open and negative volume parts affect the order. The mesh operation prints an error in the Output window with a list of parts that are not closed and manifold.

	Manual	Allows you to categorize parts using Priority Sets. Parts are sorted by the index of the priority set – which you can reorder. All parts within a priority set are sorted in the order they were created. Parts that are not in any priority set have the least priority and are also sorted by the order of creation. This option is useful to prioritize computationally expensive parts. The Output window displays computational times for each part during meshing.

Smallest Disconnected Surface

This control removes detached volumes from the surface mesh if their total number of faces is below this minimum.

Surface wrapping can generate many disconnected surfaces that you do not want in the surface mesh. This control allows you to remove these surfaces automatically.

Smallest Disconnected Surface Properties

Number of Faces	Specifies the minimum number of faces a wrapped volume can have before it is deleted from the surface mesh. If all disconnected surfaces have less than this number of faces, all surfaces are maintained. A value of zero effectively deactivates this property.

Smallest Wrapping Volume

This control ignores volumes in the wrapped surface if their volume is below this minimum. These volumes are not included in gap closure, or for the largest internal volume or the nth largest volume of interest options.

Smallest Wrapping Volume Properties

Smallest Wrapping Volume	Specifies the minimum volume that a wrapped volume can have before it is deleted from the surface mesh. A value of zero effectively deactivates this property.

Surface Curvature

This control includes cell refinement for the surface wrapper and surface remesher, using the minimum number of points around a circle (#Pts/circle) and the curvature deviation distance.

The number of points around a circle is always available, but you manually activate curvature deviation distance.

Surface Curvature Properties

Enable curvature deviation distance	**Activated**	Allows you to control surface curvature refinement with the curvature deviation distance control.

	Deactivated	Removes the curvature deviation distance control.

Surface Growth Rate

This control determines the rate at which face edge sizes vary from one face to its neighbor. The effect is only seen when there is transition from an area of refinement to the rest of the surface mesh.

As an example, the default value of 1.3 means that a face edge can be 1.3 times its neighbor length when growing from a short to a long edge length. The recommended range for Surface Growth Rate is 1.1 to 1.6.

Surface Growth Rate Properties

Surface Growth Rate	Specifies the maximum rate of face edge transition. Decreasing this value generally produces a surface mesh with more faces.

Surface Proximity

This control provides mesh refinement on surfaces that are closed to one another. The increased number of faces in the surface mesh produces more volume mesh cells between the gap.

Surface Proximity Properties

# Points in a gap	Determines the local face size by dividing the distance from one face to another across a gap by this value. However, if the gap distance is less than the *Search Floor* or more than the *Search Ceiling*, there is no refinement.
Search Floor	Specifies the minimum gap size that the mesher includes in proximity, which prevents unwanted refinement. This property is useful when the surface contains overlapping faces or faces that are too close, due to complex detail or CAD artifacts.
Search Ceiling	Specifies the maximum distance that the mesher includes in proximity, which prevents unwanted refinement. This property is useful if a component has a variable thickness and you need an option to locally refine only those areas where the thickness is less than a specified value.

Surface Size

This control determines the method by which the surface size is defined. For the surface wrapper and surface remesher, it determines the face and edge sizes. For the trimmed cell mesher, it determines the edge size of the cells next to the surface.

Surface Size Properties

Relative/Absolute	**Relative to base**	Specifies that the surface sizes are a percentage of the Base Size.

	Absolute	Specifies that the surface sizes are absolute values.
Size Method	**Min and Target**	The mesher aims for face edges that meet the target size, regardless of the local face size of the input surface. Where there is mesh refinement, the face edges do not go below the minimum size.
	Min and Max	The mesher aims for faces edges that meet current local triangle size of the input surface. Where the local triangle size exceeds the minimum or maximum size, the mesh is refined or coarsened. Where there is mesh refinement, the face edges do not go below the minimum size.
	Min only	The mesher aims for faces edges that meet current local triangle size of the input surface. Where the local triangle size is less than the minimum size, the mesh is coarsened. Where there is mesh refinement, the face edges do not go below the minimum size.

Target Surface Size

This value specifies the edge length that the mesher aims for in the absence of any mesh refinement. You can specify the target surface size as relative or absolute.

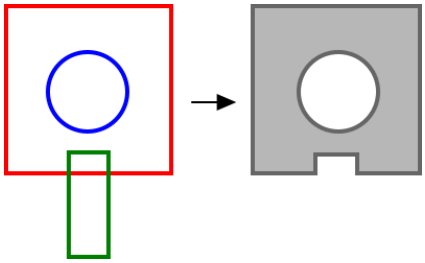
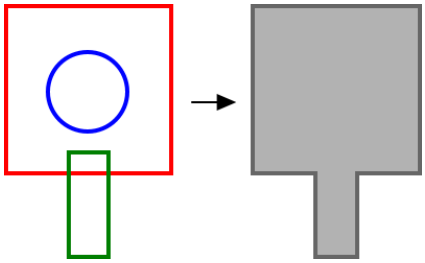
Decreasing the target surface size generates more refined and detailed surface mesh. However, this refinement increases the CPU time and size of the surface mesh.

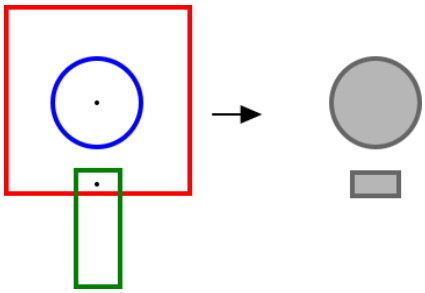
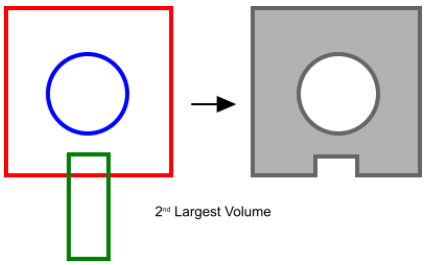
Volume of Interest

This control determines the volume that the surface wrapper extracts from the original geometry surface.

Volume of Interest Properties

Method	**Largest Internal**	Specifies that the wrapper extracts the largest internal, self-contained volume from the input geometry surfaces. In the following example, the largest internal volume is the red box, without the blue and green

		parts, as internal geometries within the volume are maintained.  To ignore internal volumes that you do not want in the surface mesh, use the smallest wrapping volume control.
	External	The *External* method specifies that the surface wrapper extracts the outer surfaces of all parts, that is, the external surfaces. The following example demonstrates this method:  In this example, all internal surfaces are omitted from the final wrapped surface. If they are part of the outer surface, the intersecting surfaces are included to form one volume.
	Seed Point	Specifies that a user-defined location determines the surface that the wrapper extracts. The resultant mesh is the surface that surrounds this location. In this example, two seed points are placed inside the blue and green volumes. The result from the

		wrapper is two closed surfaces that are shown to the right of the diagram. 
	Nth Largest	Specifies that the wrapper extracts a particular surface according to volume size. The surfaces are ranked from largest to smallest, and the user-defined Volume Number determines the output surface. Geometries that intersect one another are considered as both a single volume plus individual volumes when assessing their volumetric size. The following example shows the second largest volume that is extracted from the input geometry, which is the red box minus all other parts.  To ignore internal volumes that you do not want in the surface mesh, use the smallest wrapping volume control.

Wrapper Scale Factor

This control scales all refinement sizes (for example, surface size and volumetric controls) by a common factor during the surface wrapping process.

This facility allows you to generate a wrapped surface with different controls to the other meshers, so that you can generate a finer surface mesh without running the wrapper independently. It also allows quick and easy alteration of all the surface wrapper controls.

The wrapper scale factor is not applied to the following controls:

- Smallest Wrapping Volume
- Smallest Disconnected Faces
- Gap Closure Size
- Contact Prevention
- Wrapper Feature Angle

Wrapper Scale Factor Properties

Percentage	Specifies the scale factor by which all surface wrapper controls are scaled. A higher or lower value results in a coarser or finer mesh respectively.

Volume Meshers

STAR-CCM+ contains different types of meshing models that can be used to generate a volume mesh starting from a suitably prepared surface mesh.

The five types of meshing models are:

- Tetrahedral - tetrahedral cell shape based core mesh
- Polyhedral - arbitrary polyhedral cell shape based core mesh
- Trimmed - trimmed hexahedral cell shape based core mesh
- Thin Mesh - tetrahedral or polyhedral based prismatic thin mesh
- Advancing Layer Mesh - polyhedral core mesh, with in-built prismatic layers that advance inward from a polygonal surface mesh.

Prismatic near wall layers can be included with the first three types of mesh that are listed above by including the prism meshing model as part of the volume meshing process. For multi-region cases, a conformal mesh at the interface between one region and the next is currently possible for the tetrahedral and polyhedral mesh types as part of the meshing process. This option is currently not supported for the trimmed cell mesher.

Different mesh types can also be used for different mesh regions if desired by using a different mesh continuum for each region. A conformal mesh interface at inter-region boundaries is not supported in this case.

Volumetric controls in the shapes of bricks, spheres, cylinders, and cones can be used during the volume meshing of the core mesh to increase or decrease the local cell density. Arbitrary shaped volumes that are imported as parts can also be used to define the refinement volume. Feature refinement is automatically included in the case of the trimmer meshing model.

The extruder mesher can be used with the core mesh methods to extend the original domain by extruding a boundary. The extruder produces orthogonal cell layers from the chosen boundaries. An example use of the extruder mesher would be to extend an inlet duct upstream.

The generalized cylinder mesher can be used with the polyhedral meshing model to generate a mesh on elements in the geometry that are roughly cylindrical. The generalized cylinder mesher generates extruded orthogonal cells along the length of cylindrical boundaries.

In cases where a previous solution exists and the mesh is modified, the solution is automatically mapped to the new mesh in order to restart the simulation.

Various field functions are created during volume meshing operations; some of which are documented in the appropriate reference section.

Once the volume mesh has been built, it can be checked for errors and exported to other packages if desired.

Contents:

Volume Meshers Overview
Polyhedral Mesher
Tetrahedral Mesher
Trimmed Mesher

Prism Layer Mesher
Advancing Layer Mesher
Extruder Mesher
Thin Mesher
Generalized Cylinder Mesher
Volume Mesher Controls and Values

Volume Meshers Overview

This section introduces volume meshing.

Choosing a Volume Mesh Model

In general, the choice of which mesh type model to use for the core volume mesh is dependent on several factors. Some of these factors are:

- the turn around time available for building the mesh
- the desired solution accuracy and convergence rate
- the amount of memory available on your computer
- whether you are building a single or multi-region mesh
- the quality of the starting surface mesh
- whether the geometry is predominantly thin or not.

However, some of these factors are not mutually exclusive. For example, the tetrahedral mesher is the fastest volume mesh generator (in terms of number of cells that are generated per minute) and the one that uses the least amount of memory. However, in terms of solution quality, approximately five to eight times the number of cells are required to produce the same accuracy as the equivalent polyhedral or trimmed cell mesh. The additional solution time, memory, and increased time to convergence when solving on this type of mesh must be factored in against any time gains that are made during the meshing process.

In terms of general accuracy for a given number of cells, the trimmed and polyhedral cell type meshes produce more accurate solutions when compared to a tetrahedral mesh. Mesh independent solutions can be obtained for all three mesh types if the density is increased to the appropriate level.

If you are building a multi-region grid where a conformal mesh is required at interfaces, then use the tetrahedral or polyhedral meshers with a single mesh continuum. However, the trimmer model does not produce conformal mesh interfaces and a different continuum must be used for each region. Alternatively the per-part or per-region meshing option can be activated instead to apply the same continuum to all regions.

Both the tetrahedral and polyhedral volume meshers are directly dependent on the quality of the starting surface triangulation. In other words, a bad quality starting surface leads to a bad quality volume mesh. For complex meshing problems however, the overall cell quality of the polyhedral meshes can suffer when compared to a tetrahedral mesh starting from the same surface. The trimmer model is not directly dependent on the surface quality of the starting surface and as such is more likely to produce a good quality mesh for most situations.

For geometries that are predominantly composed of thin surfaces, it is recommended that the thin mesher is used. The thin mesher produces a better quality prismatic type mesh than would be created using one of the other three volume meshers.

Polyhedral Mesher

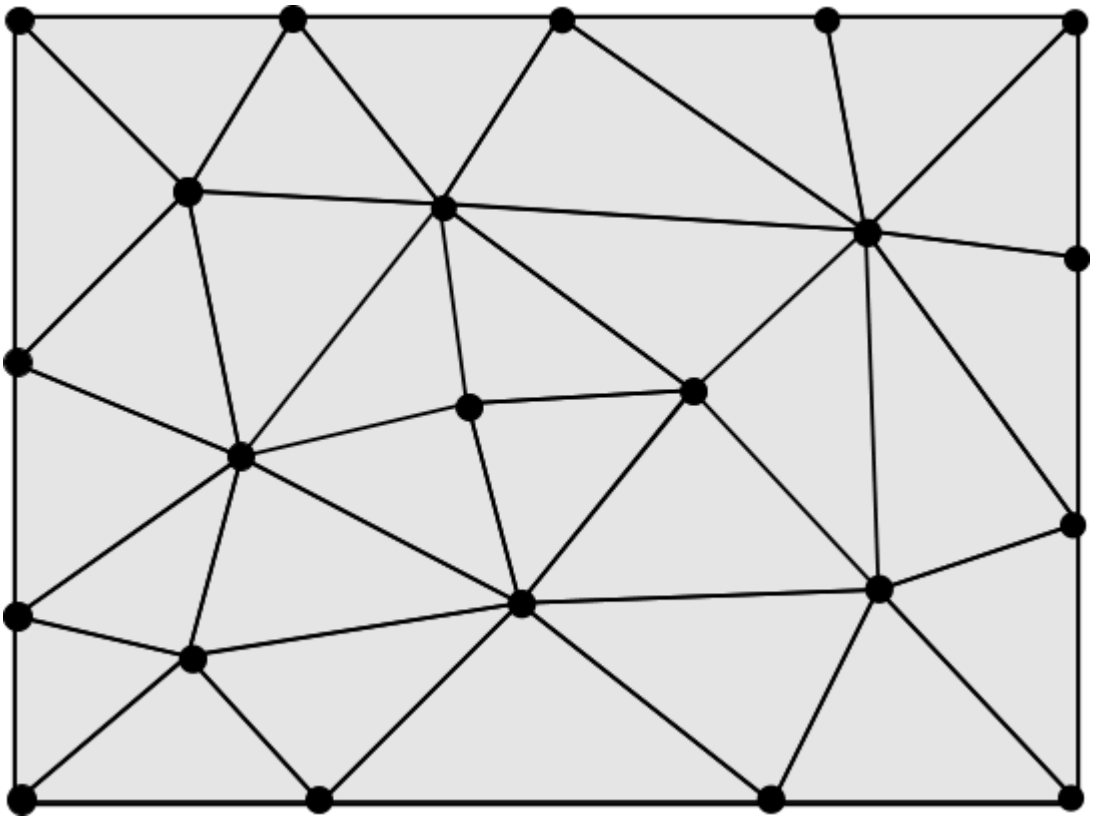
Polyhedral meshes provide a balanced solution for complex mesh generation problems.

They are relatively easy and efficient to build, requiring no more surface preparation than the equivalent tetrahedral mesh. They also contain approximately five times fewer cells than a tetrahedral mesh for a given starting surface. Multi-region meshes with a conformal mesh interface are allowed.

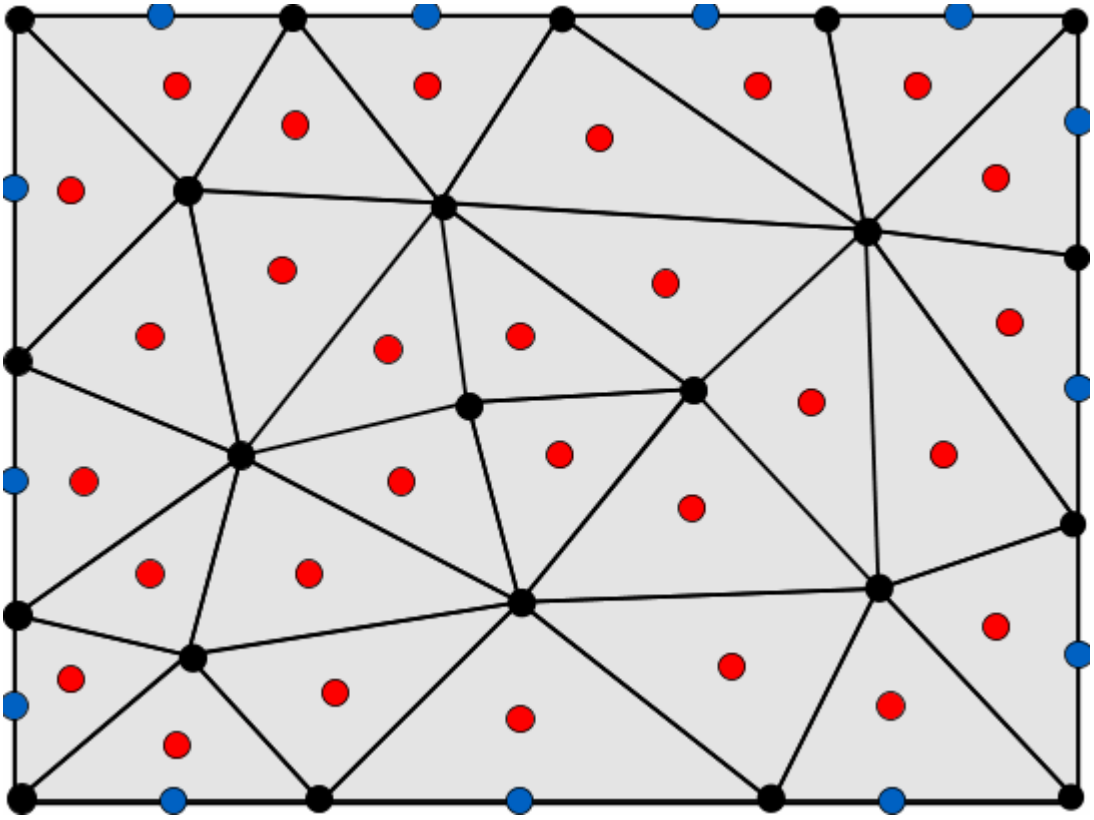
The polyhedral meshing model utilizes an arbitrary polyhedral cell shape in order to build the core mesh. In STAR-CCM+, a special dualization scheme is used to create the polyhedral mesh from an underlying tetrahedral mesh, which is automatically created as part of the process. STAR-CCM+ calculates the initial cell count from this underlying tetrahedral mesh. The polyhedral cells that are created typically have an average of 14 cell faces.

The following sequence of images demonstrate the process of generating a polyhedral mesh in STAR-CCM+.

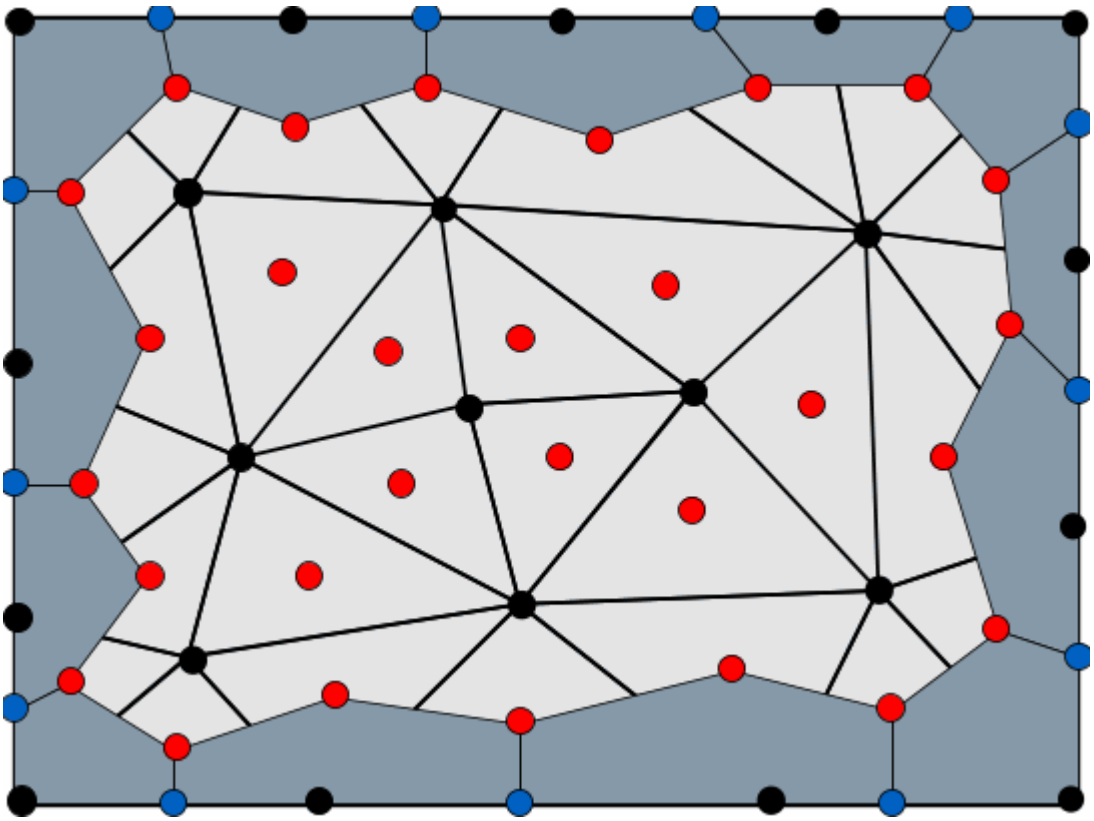
A tetrahedral mesh is generated for the input surface.



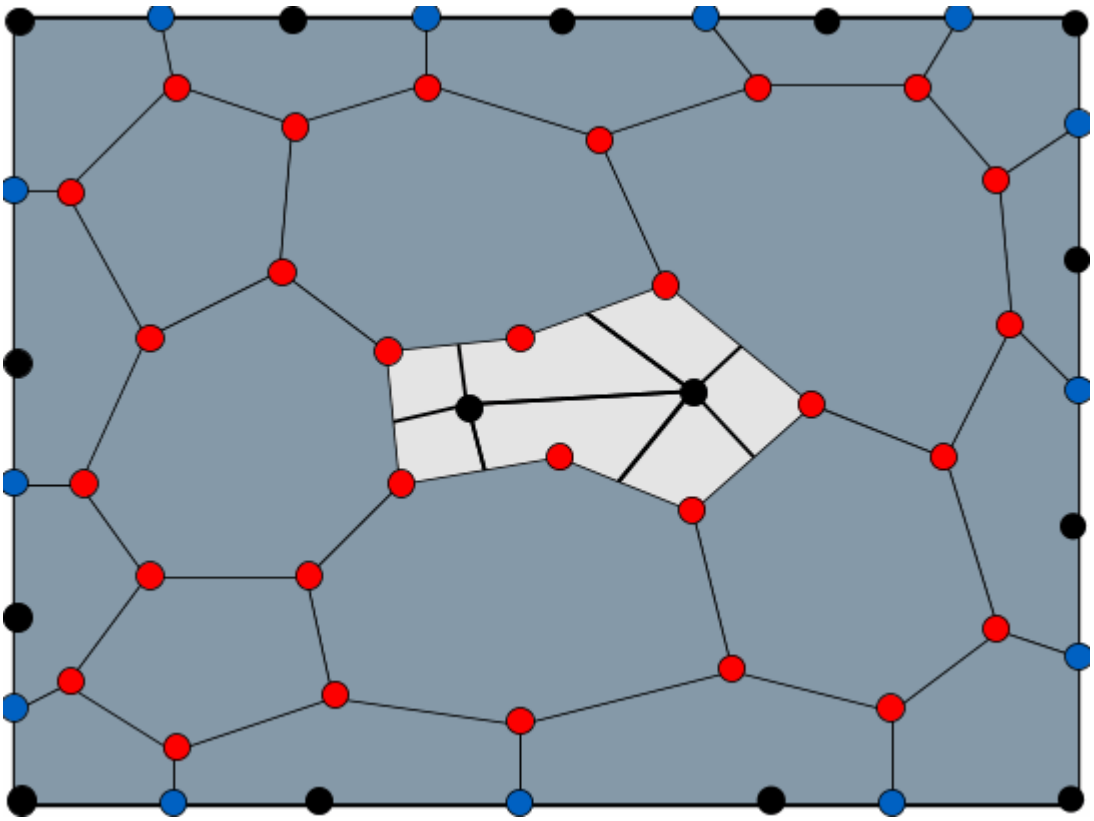
A dualization scheme is used to mark the center of the tetrahedral cells (red dots in example), as well as the midpoints (blue dots in example) on the boundary edges.



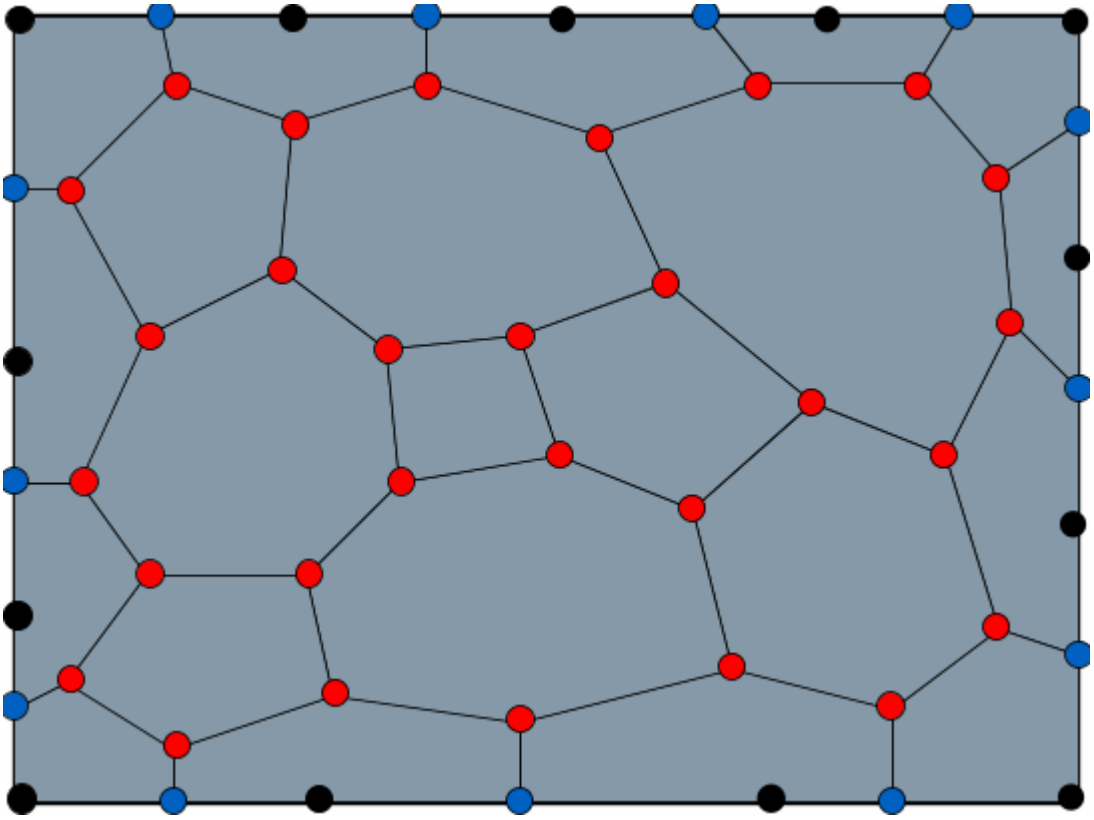
Polyhedral cells are generated, starting from the boundary edges.



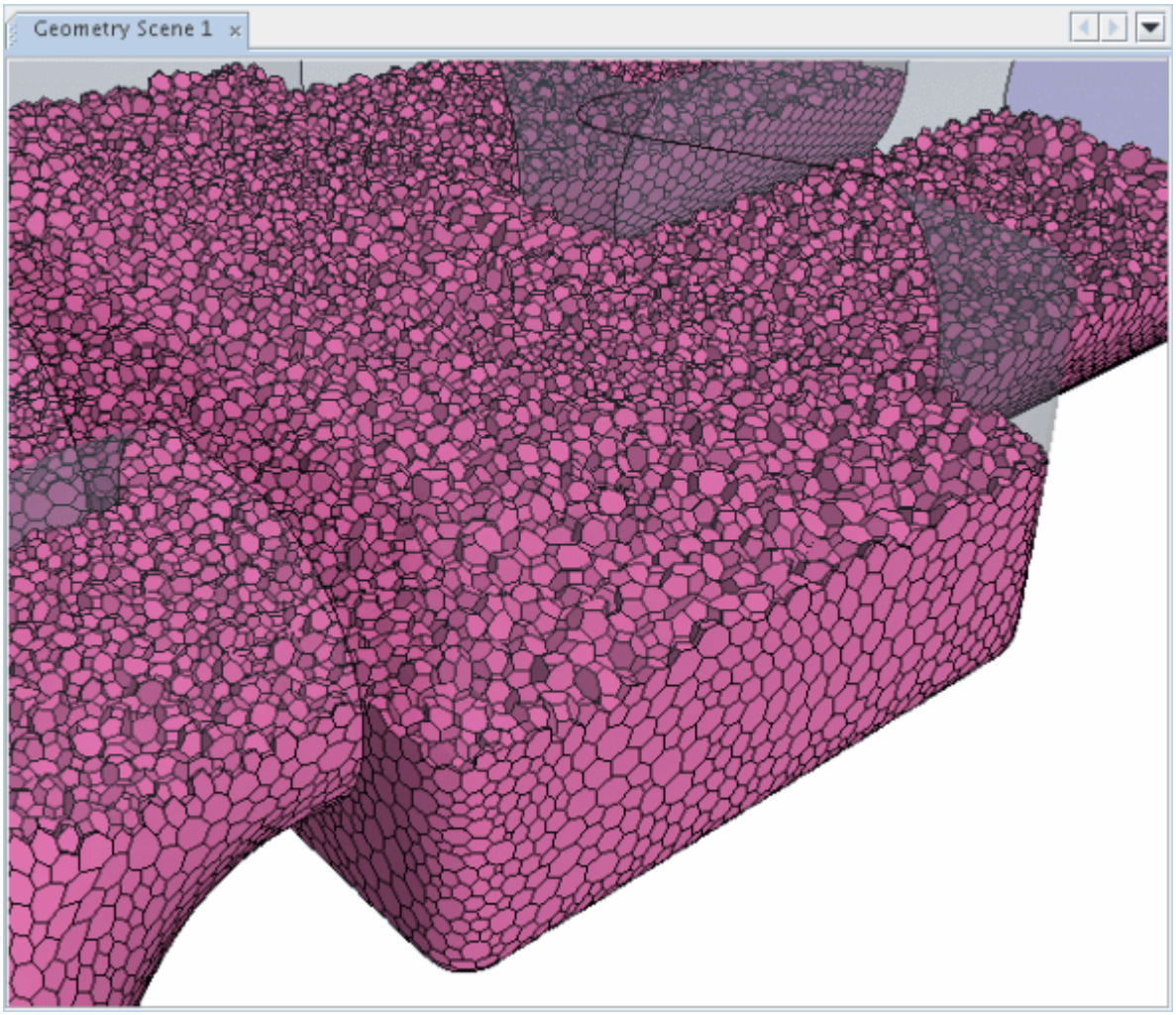
Further polyhedral cells are generated from the outer layer.



The process carries on until all polyhedral cells are generated.



An example polyhedral core mesh is shown below:



The polyhedral core mesh density can be increased or decreased by using the volume mesh density factors. Volumetric controls that are chosen from a range of prescribed shapes can also be included to increase or decrease the mesh density locally. Global refinement can be applied to increase the density for every cell, and field function based refinement can be used to create local density variations which are based on a mesh size table. When the polyhedral mesher is used with a periodic interface, then it attempts to produce a conformal mesh for the boundary pair where possible.

For geometries containing cylindrical elements, the polyhedral mesher can be used with the generalized cylinder mesher, which generates extruded orthogonal cells along the length cylindrical sections. For geometries containing thin walled structures, the polyhedral mesher can be used with the embedded thin mesher, which generates prismatic type cells in areas that are considered to be thin.

Polyhedral meshes can normally be generated with free floating baffles. However, issues are sometimes seen when the baffles do not have prismatic layers. If you encounter such issues, activate the prism mesher for the baffles so that at least one layer of prism cells is generated to help alleviate problems with cell generation at baffle edges.

Some meshes generated by a pipeline that includes the polyhedral mesher can show additional edges on the boundary. These additional edges are added by the polyhedral mesher in order to avoid warped boundary faces and to enhance conformality among regions. The effect is stronger if the triangulation of the input surface is coarse.

Polyhedral Mesher Guidelines

- The maximum limit of polyhedral cells per process is 120 million cells.
- For any polyhedral meshes containing greater than 120 million cells, use parallel meshing with at least one additional process for every 120 million cells.

Contents:

Global Polyhedral Mesh Refinement

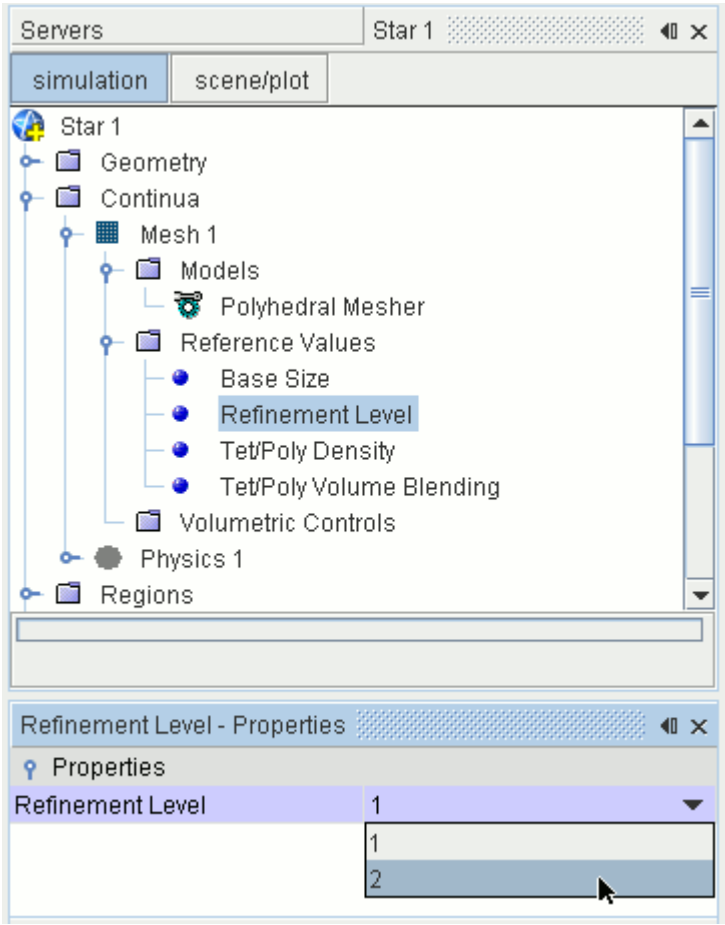
See Also:

Generalized Cylinder Mesher

Global Polyhedral Mesh Refinement

The *Include Refinement* option on the **Polyhedral Mesher** node provides a quick method to increase the cell count in a polyhedral mesh globally.

This option is only available for Region Based Meshing. Ticking this option adds the **Refinement Level** node that is shown below.



Two refinement levels are available:

- Refinement Level 1: Splits each polyhedral cell into approximately 6 or 7 new cells.
- Refinement Level 2: Adds another level; so splits each original polyhedral cell into approximately 40 - 50 new cells.

Therefore, the overall cell count of the polyhedral mesh increases by a factor of 6 - 7 for one refinement level, and by a factor of 40 - 50 for 2 refinement levels.

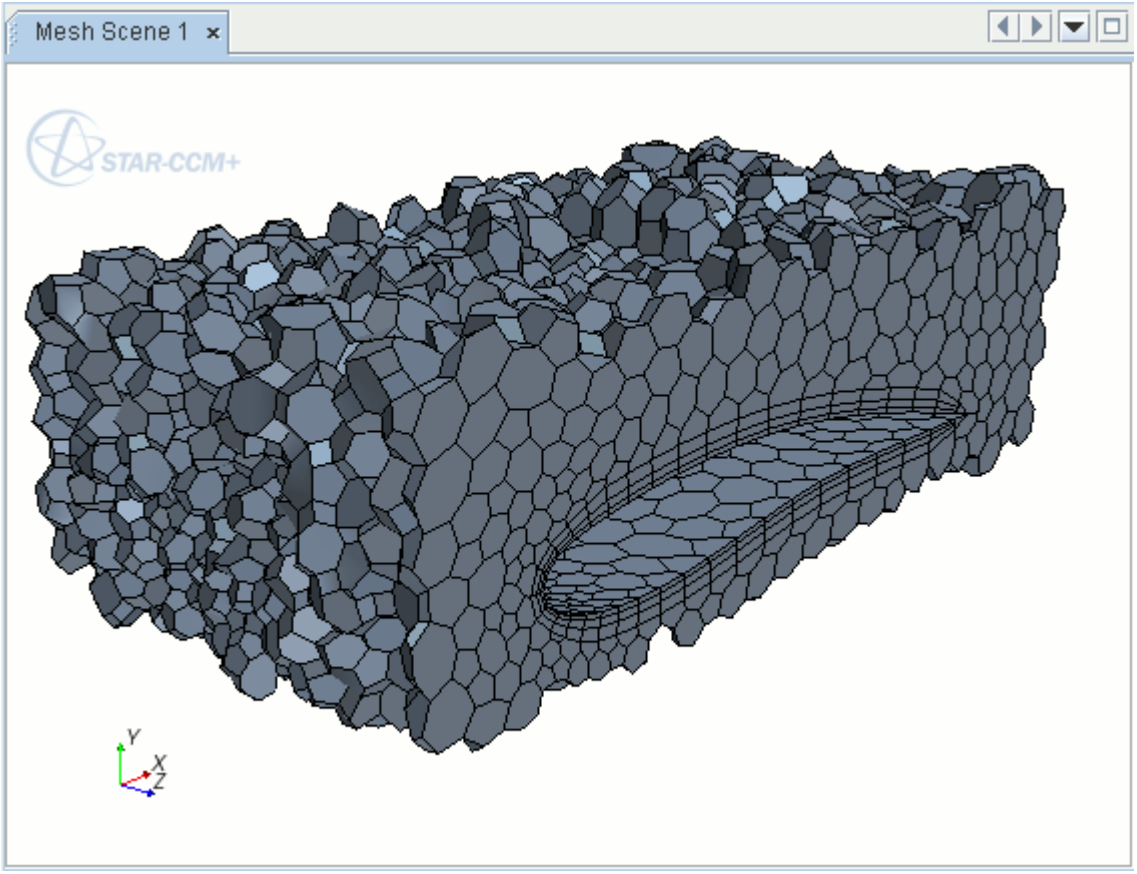
Refining a coarse mesh takes less computational time to increase cell count when compared to creating a fine mesh from the initial surface mesh.

The thickness and number of the prism layers remain unchanged by the global refinement of the polyhedral mesh.

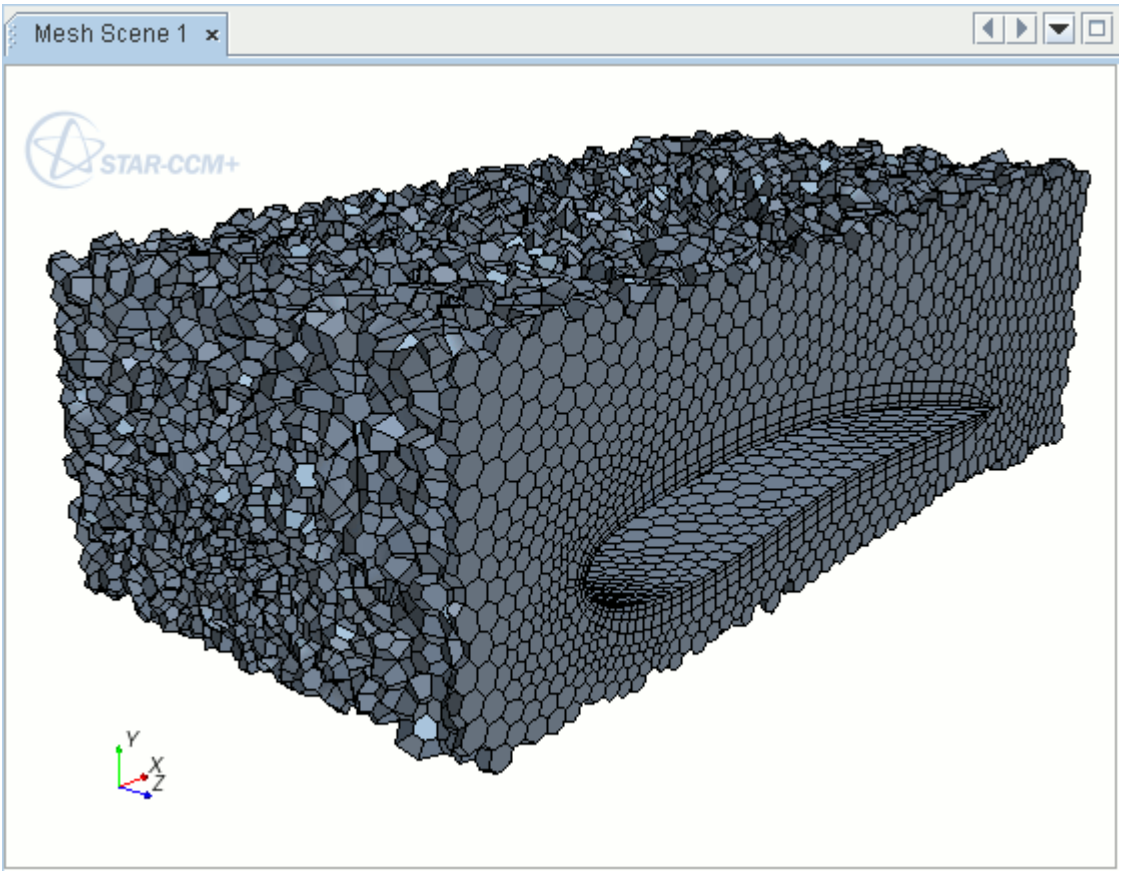
Also, as global refinement splits the polyhedral cells, it does not affect the surface mesh. Therefore, you do not need to run the surface remesher with global refinement and the surface mesh is not as refined as the volume mesh.

As an example of this global refinement, observe the changes that result from one level of refinement, and then from two levels of refinement:

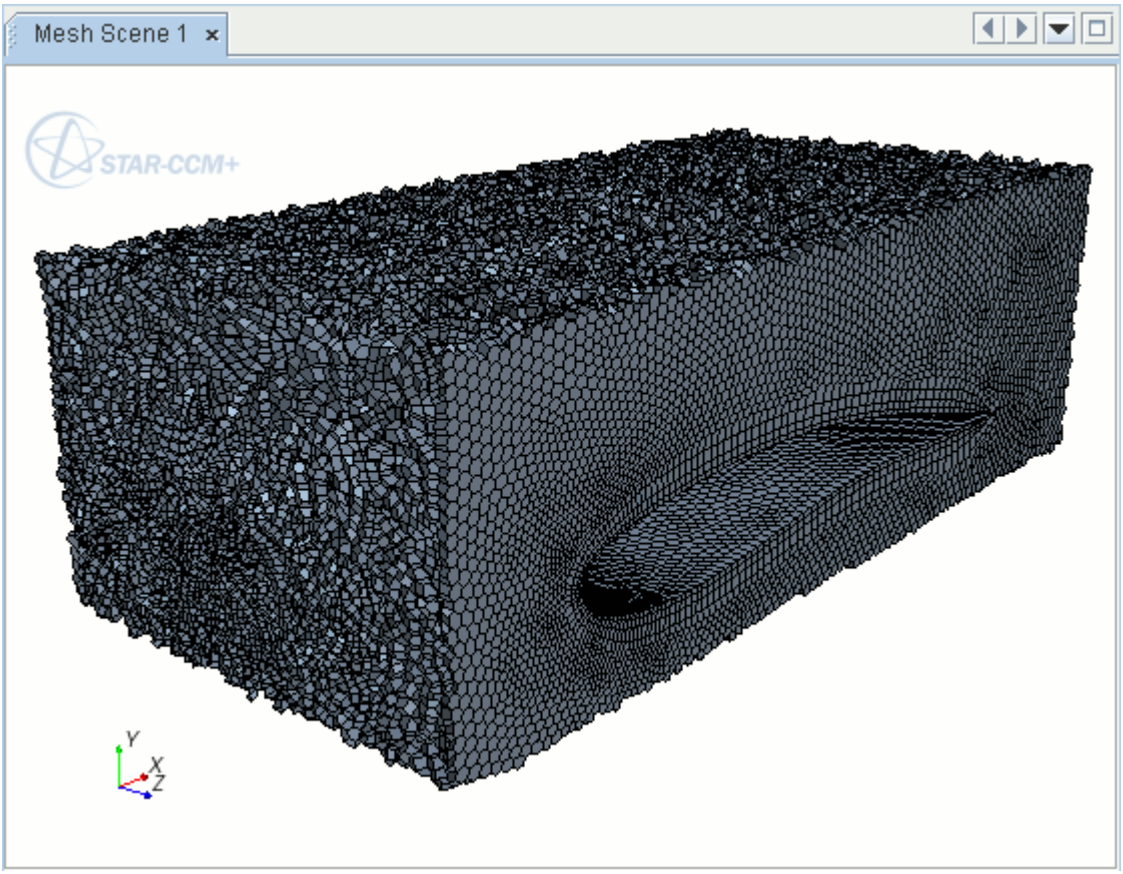
- Initial coarse mesh.



- Refinement level 1.



- Refinement level 2.



Contents:

Behavior of Global Refinement with Field Function Refinement

Behavior of Global Refinement with Field Function Refinement

This section summarizes the behavior of the polyhedral mesher when using global refinement with field function mesh refinement.

For one level of global refinement, consider a cell in the mesh where:

- FF — is the cell size that is prescribed by field function mesh refinement.
- C — is the cell size before refinement.

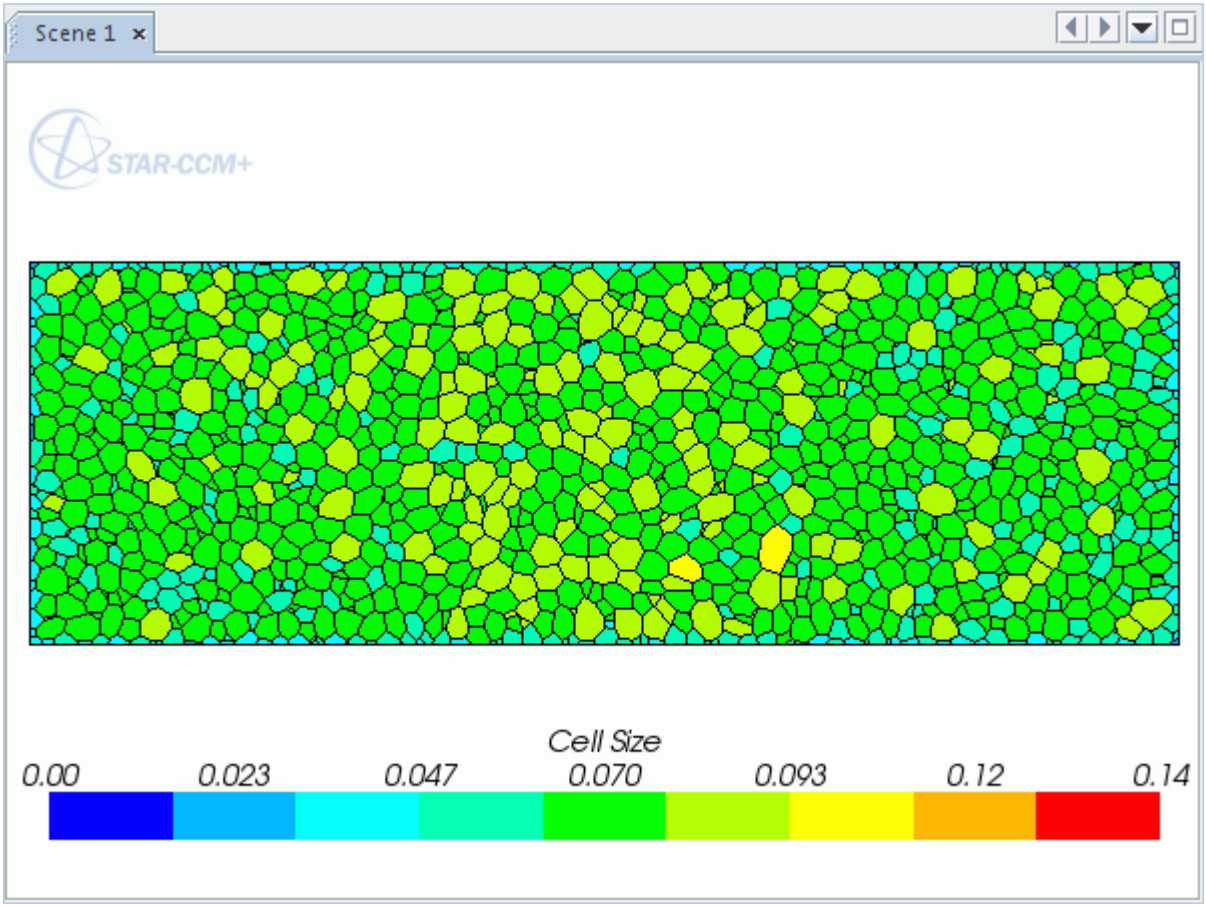
If FF < C, global refinement splits the cell according to the refinement level algorithm. The final cell size does not necessarily meet the requirements of the field function refinement.

For two levels of global refinement, consider the same cell where:

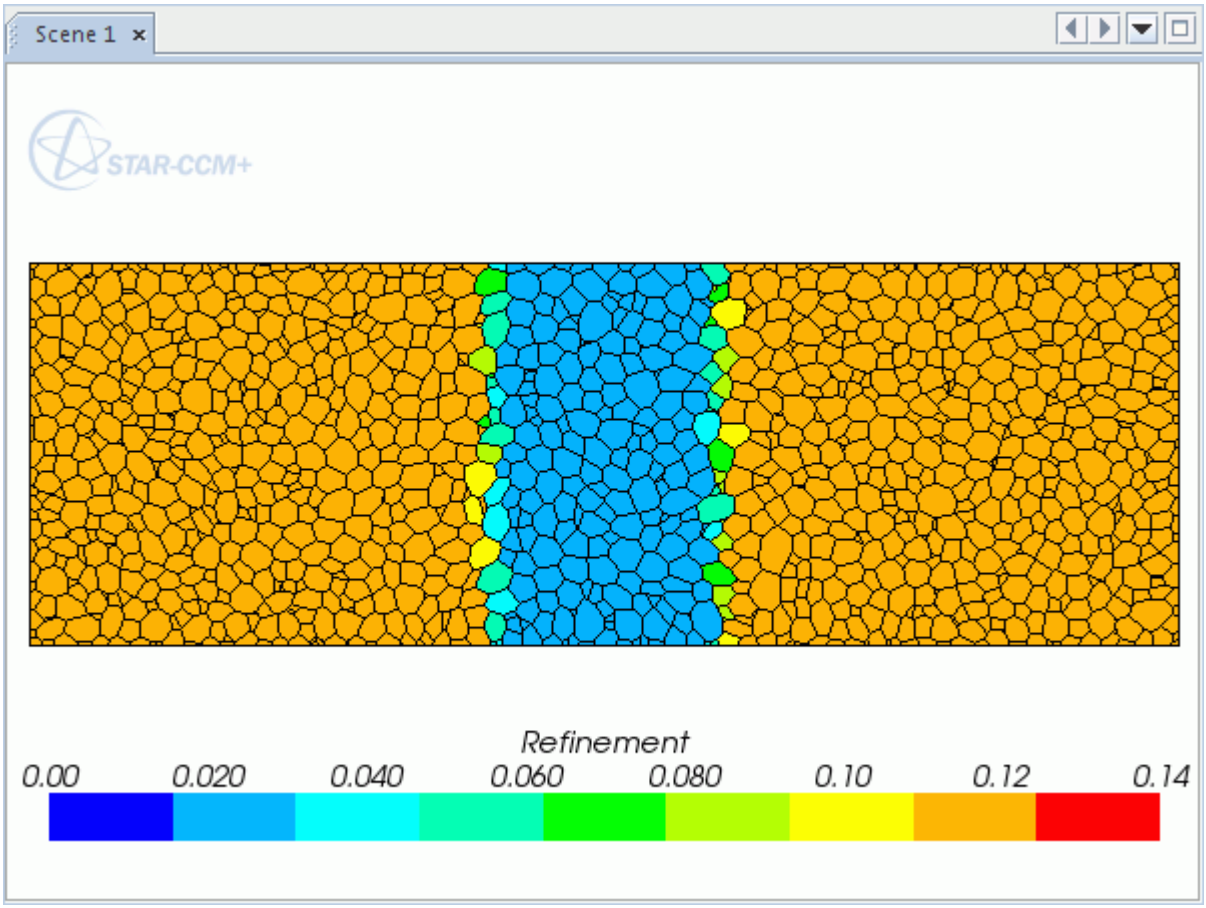
- FF — is the cell size that is prescribed by field function mesh refinement.
- c — is the cell size after one level of global refinement.

Similarly, if FF < c, global refinement splits the cell according to the refinement level algorithm. Again, the final cell size does not necessarily meet the requirements of the field function refinement.

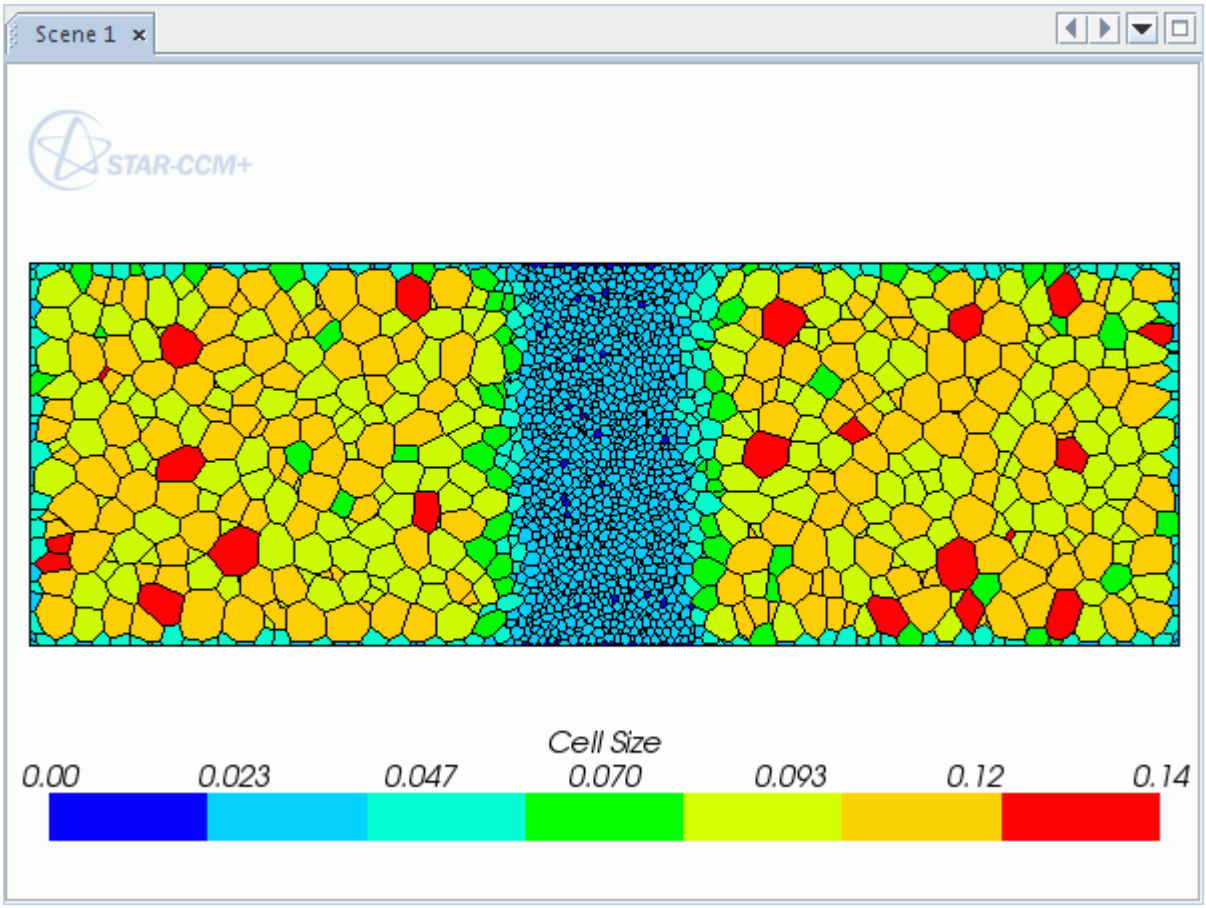
The following example refines a mesh section to less than the original cell size, coarsens the rest of the mesh, and then applies global refinement. The original cell size within the mesh is approximately 0.07 m as shown below.



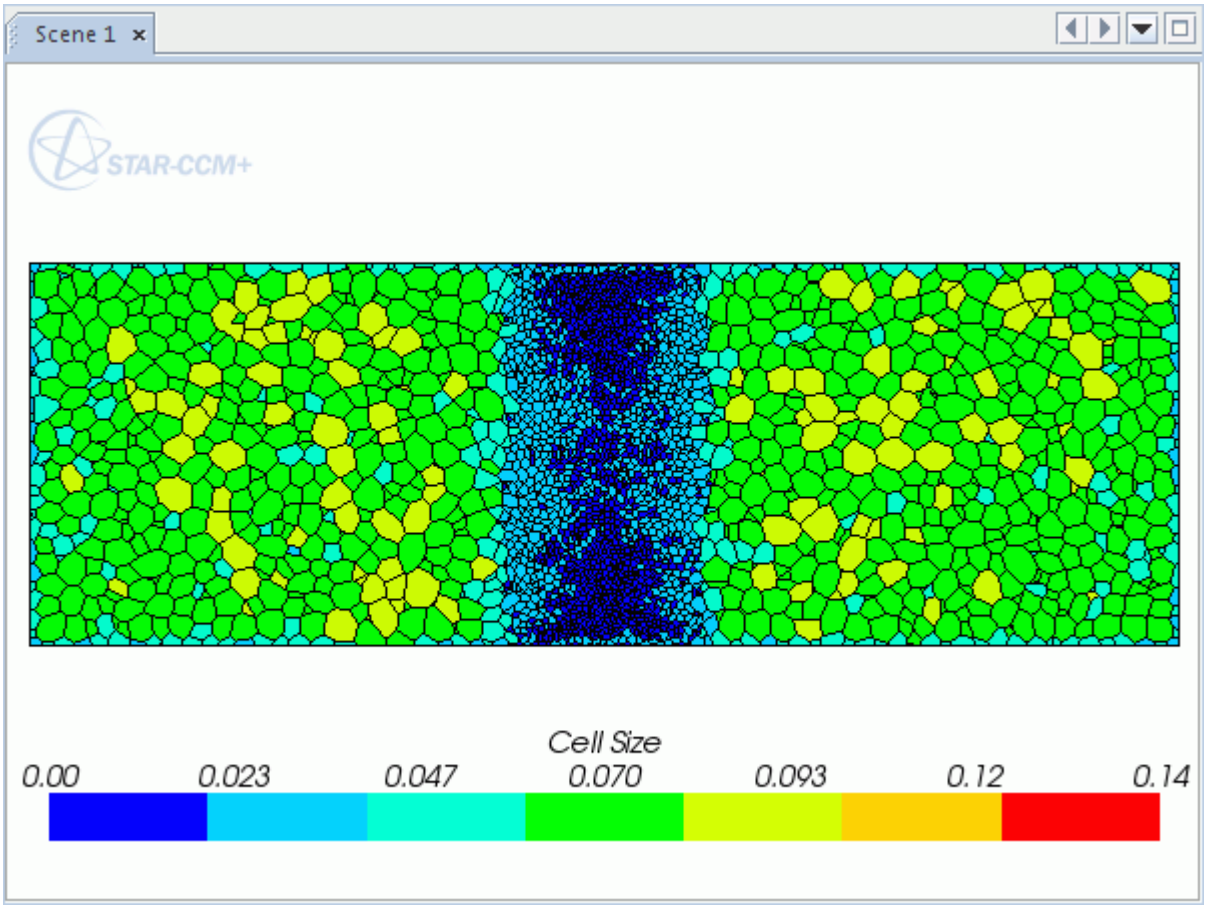
The applied field function refines the mesh to a cell size of 0.02 m in the center and coarsens the mesh to 0.12 m in the rest of the mesh.



The following image shows the mesh when only field function mesh refinement is applied. The coarse section does not have a consistent cell size of 0.12 due to growth rates.



The following image shows the mesh after applying field function refinement and global refinement. In the refined section, the mesher applies global refinement and the cells are approximately six or seven times smaller. In the coarse sections, the mesher does not apply global refinement and the cells still have an approximate cell size of 0.07 m.



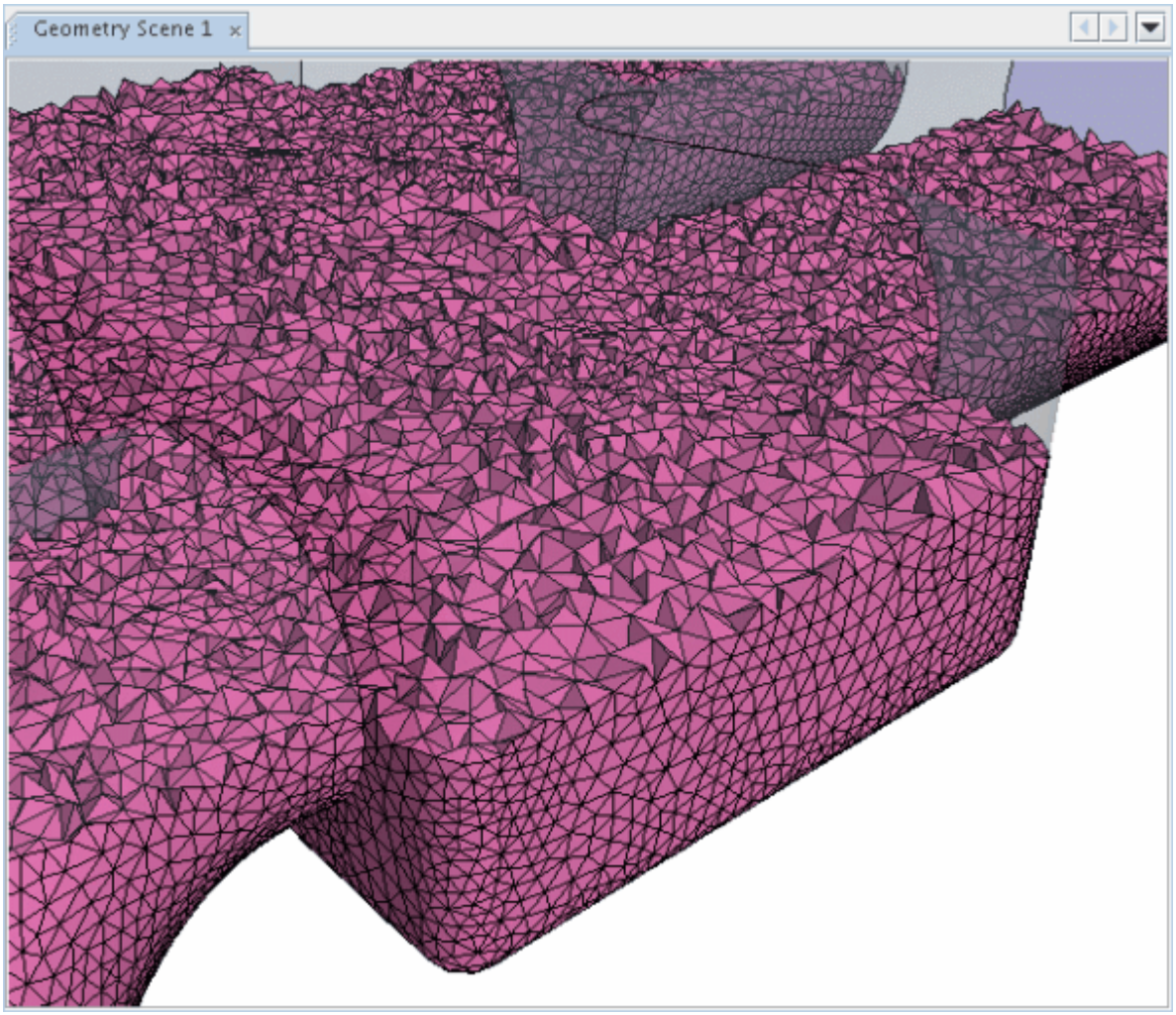
Tetrahedral Mesher

Tetrahedral meshes provide an efficient and simple solution for complex mesh generation problems.

Out of the three provided models, the tetrahedral is the fastest and uses the least amount of memory for a given number of cells. Multi-part or multi-region meshes with a conformal mesh interface are allowed.

The tetrahedral meshing model utilizes the tetrahedral cell shape in order to build the core mesh. In STAR-CCM+, the Delaunay method is used to construct the mesh, which iteratively inserts points into the domain, forming high-quality tetrahedra in the process. The resulting mesh strictly conforms to the triangulation of the surface at the domain boundary, so the quality of the original surface mesh must be good to ensure a good quality volume mesh.

An example tetrahedral core mesh is shown below:



The tetrahedral core mesh density can be increased or decreased by using the volume mesh density factors. Volumetric controls that are chosen from a range of prescribed shapes can also be included to increase or decrease the mesh density locally. When the tetrahedral mesher is used with a periodic interface, it attempts to produce a conformal mesh for the boundary pair where possible.

Tetrahedral Mesher Expert Properties

The tetrahedral mesher expert properties that are common with the polyhedral mesher are documented in Volume Mesher Controls and Values.

Trimmed Mesher

The trimmed cell mesher provides a robust and efficient method of producing a high-quality grid for both simple and complex mesh generation problems.

It combines a number of highly desirable meshing attributes in a single meshing scheme:

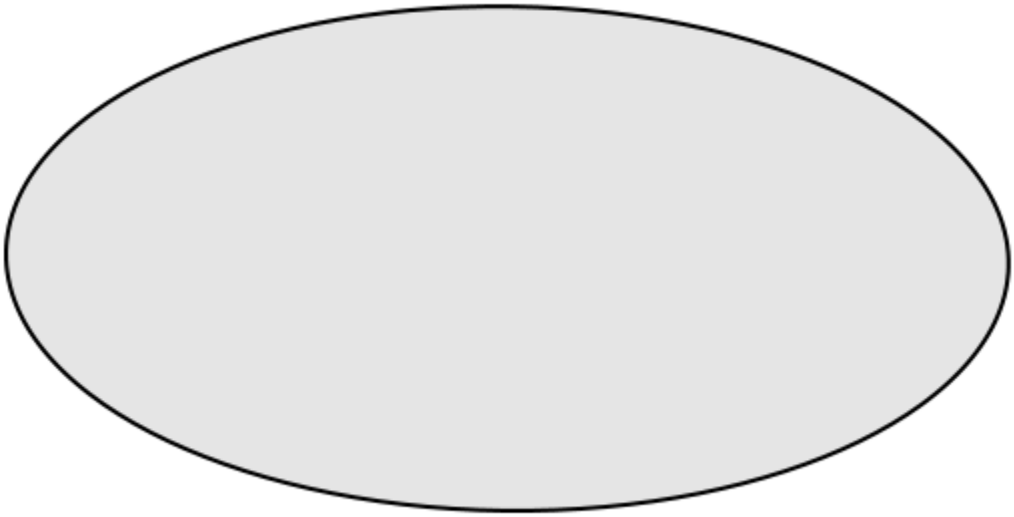
- predominantly hexahedral mesh with minimal cell skewness
- refinement that is based upon surface mesh size and other user-defined refinement controls

- surface quality independence
- alignment with a user specified coordinate system.

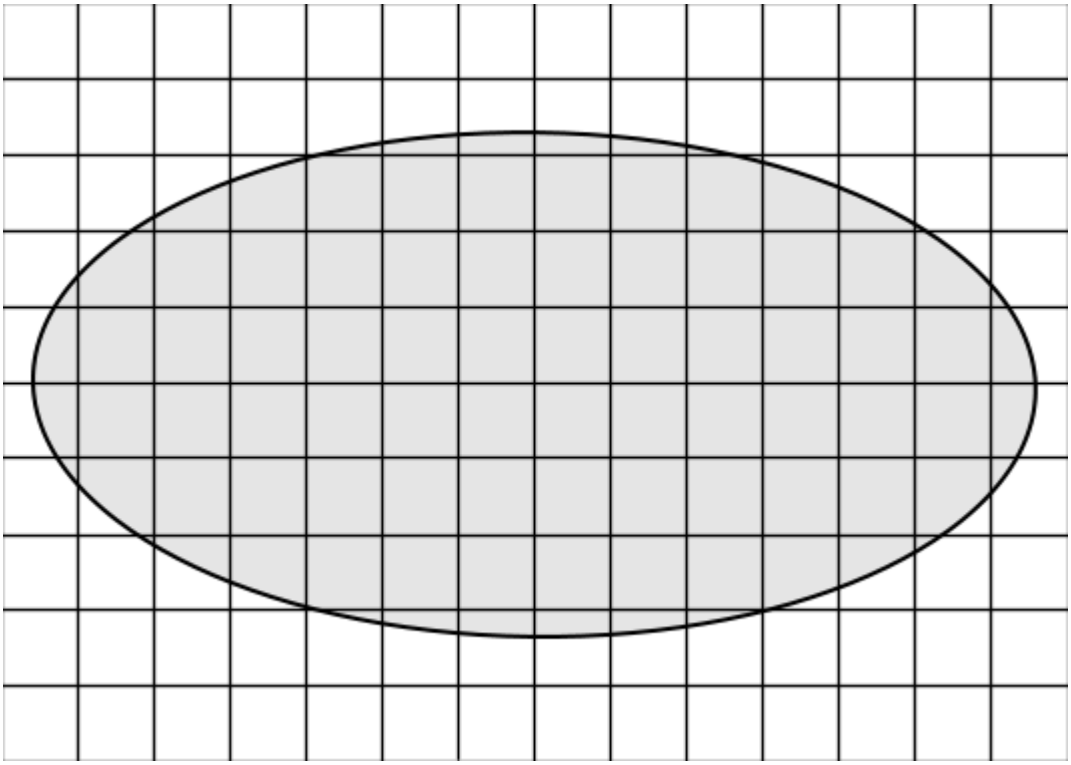
The trimmed cell mesher is available for both the Parts Based Meshing (PBM) and Region Based Meshing (RBM) approaches.

By default, the trimmer meshing model utilizes a template mesh that is constructed from hexahedral cells at the target size from which it cuts or trims the core mesh using the starting input surface. The template mesh contains refinement that is based on the local surface mesh size and local refinement controls. Growth parameters can be used to control the transitioning of the mesh cell sizes from small to large both at the surface and far field. A maximum and/or minimum cell size can be supplied as well to control the upper and lower cell size bounds. The template can be aligned in any direction in a user specified Cartesian coordinate system.

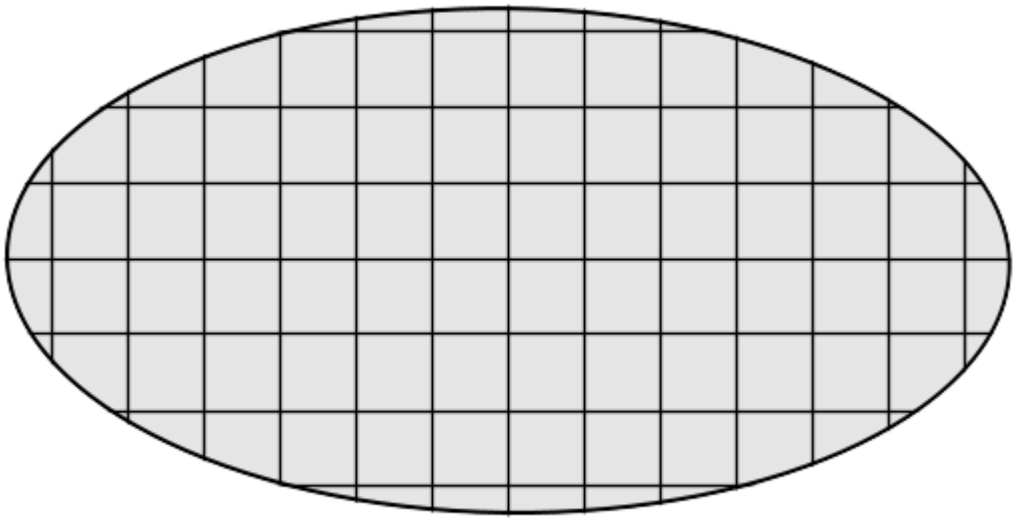
The following sequence of images shows how a trimmed mesh is generated for an input surface in STAR-CCM+.



The template is generated from hexahedral cells using the target size.

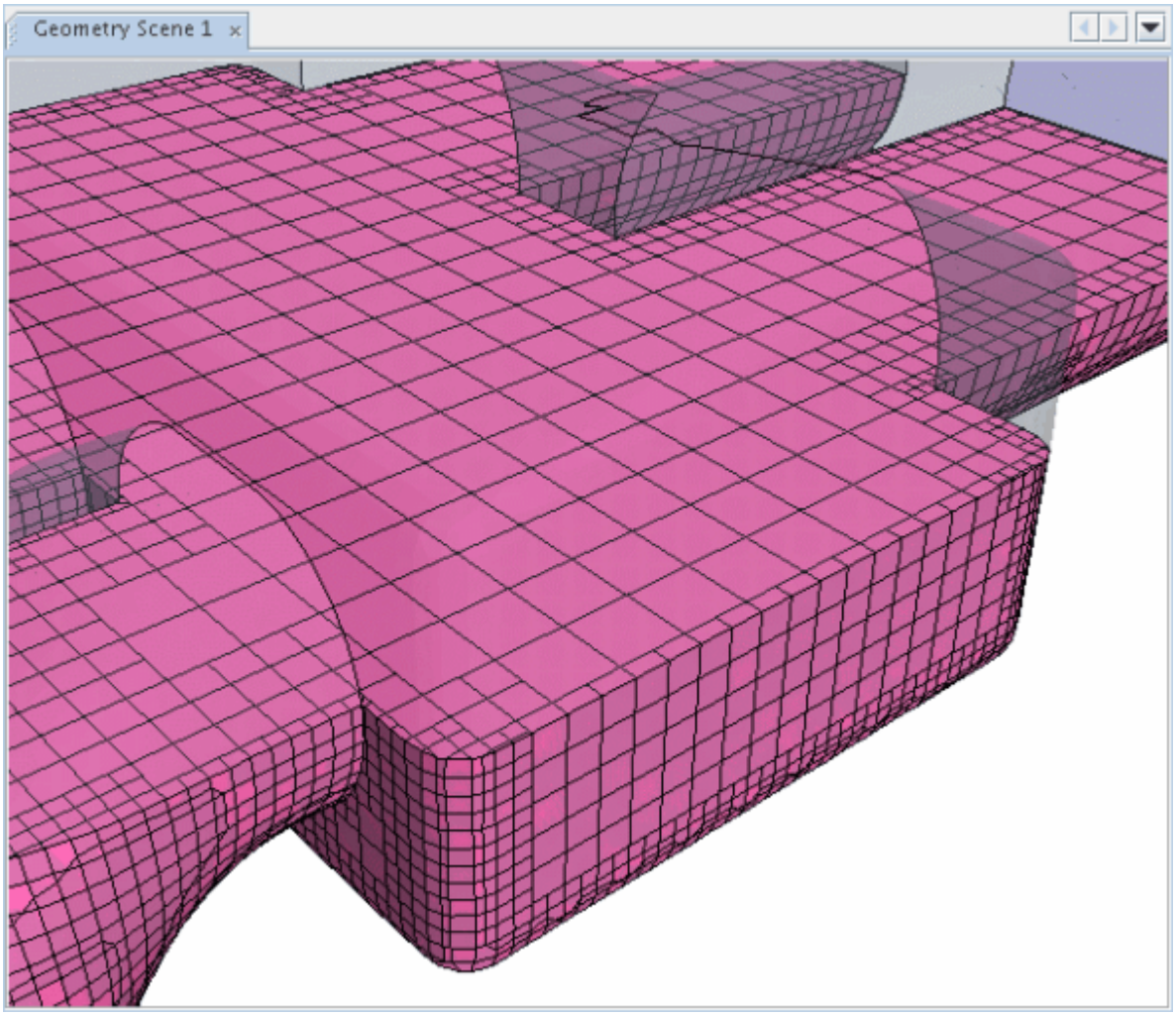


The core mesh is trimmed using the input surface.



The resulting mesh is composed predominantly of hexahedral cells with trimmed cells next to the surface. Trimmed cells are polyhedral cells but can usually be recognized as hexahedral cells with one or more corners and/or edges that are cut off.

An example trimmed cell core mesh is shown below:



An additional feature that is useful in modeling external aerodynamic flows is the ability to refine cells in a wake region. This region is generally the volume of fluid at the rear of a moving body.

The input values used for the trimmer model can be set on four different levels:

- global
- wake refinement region
- part surface or boundary
- part or feature curves

Volumetric controls can also be included to increase or decrease the mesh density locally in the template using a range of prescribed shapes to define the extents.

The current implementation of the **Trimmed Cell Mesher** (PBM) / **Trimmer** (RBM) model is restricted to one trimmed volume mesh per automated mesh operation (PBM) or mesh continuum (RBM). In other words, a

different automated mesh operation or mesh continuum must be used for each trimmed volume mesh that you wish to generate. Alternatively, the per part (PBM) or per-region (RBM) meshing option can be activated to mesh parts or regions independently. At part or region interfaces a non-conformal interface is generated. Additionally, interfaces can only be used with the trimmer meshing model when the boundaries in the interface belong to the same region. For example, an in-place interface could be used to model a baffle in the flow domain, or a periodic interface for periodic flow condition. In all instances, a conformal trimmed mesh across the interface is not guaranteed.

The trimmer model also has a built-in ceiling to control the size of the template cells next to surfaces. This ceiling is to avoid grossly de-featuring the model and generating bad quality cells. At present, the maximum surface template cell size is fixed to 150% of the average triangle edge length size in the local surface. This factor does not affect cells away from the surface.

Trimmed Mesher Guidelines

- The trimmer template mesh size is based on the default target surface size. The trimmer mesh only grows by multiples of 2 from the target surface size. You are advised to set all refinements for the trimmer mesh as multiples of 2 from the target surface size, for example: `25%`, `50%`, `200%`, etc. Using a target surface size of `100%` of the base size can be helpful.
- Activate parallel trimming on meshes with more than 100,000 cells per process. For meshes smaller than this threshold, serial trimming will often outperform parallel trimming due the extra overhead of parallel communication between processes. Adding more processors will improve performance at a diminishing rate of return. At some point adding more processors will result in a negligible increase or even a decrease in performance. The optimal number of processors depends on the case.

Contents:

Trimmed Mesher Properties
Setting Trimmer Default Controls (PBM) / Reference Values (RBM)
Volume Growth Rate (PBM) / Template Growth Rate (RBM)

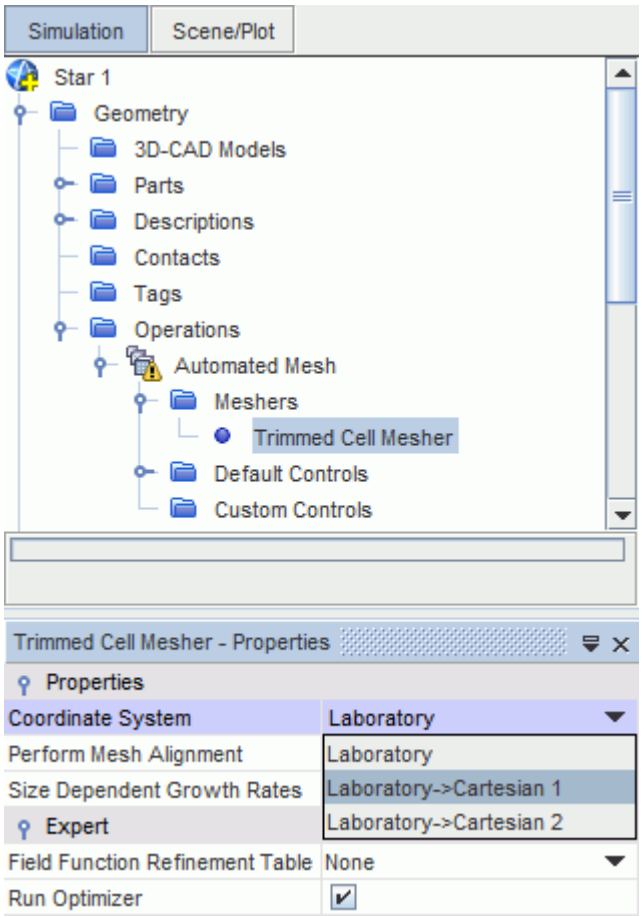
Trimmed Mesher Properties

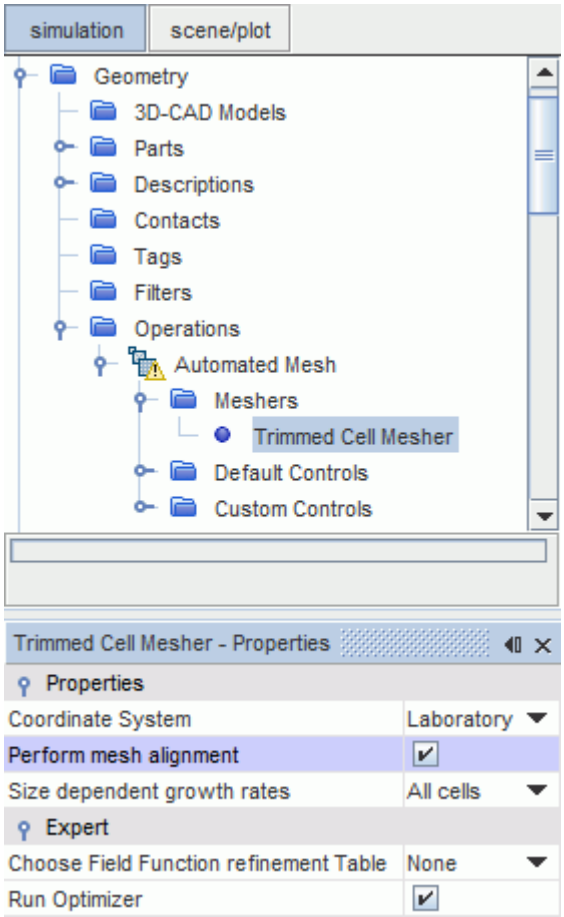
This section covers the trimmed mesher properties for both Parts Based Meshing (PBM) and Region Based Meshing (RBM).

Screenshots are of the Parts Based Meshing approach. Where the names of nodes or properties differ between the approaches, this is indicated by the acronym PBM for Parts Based Meshing and RBM for Region Based Meshing. When using Parts Based Meshing the properties are found under the **Geometry** > **Operations** > **Automated Mesh** > **Meshers** > **Trimmed Cell Mesher** node. When using Region Based Meshing the properties are found under the **Continua** > **Mesh** > **Models** > **Trimmer** node.

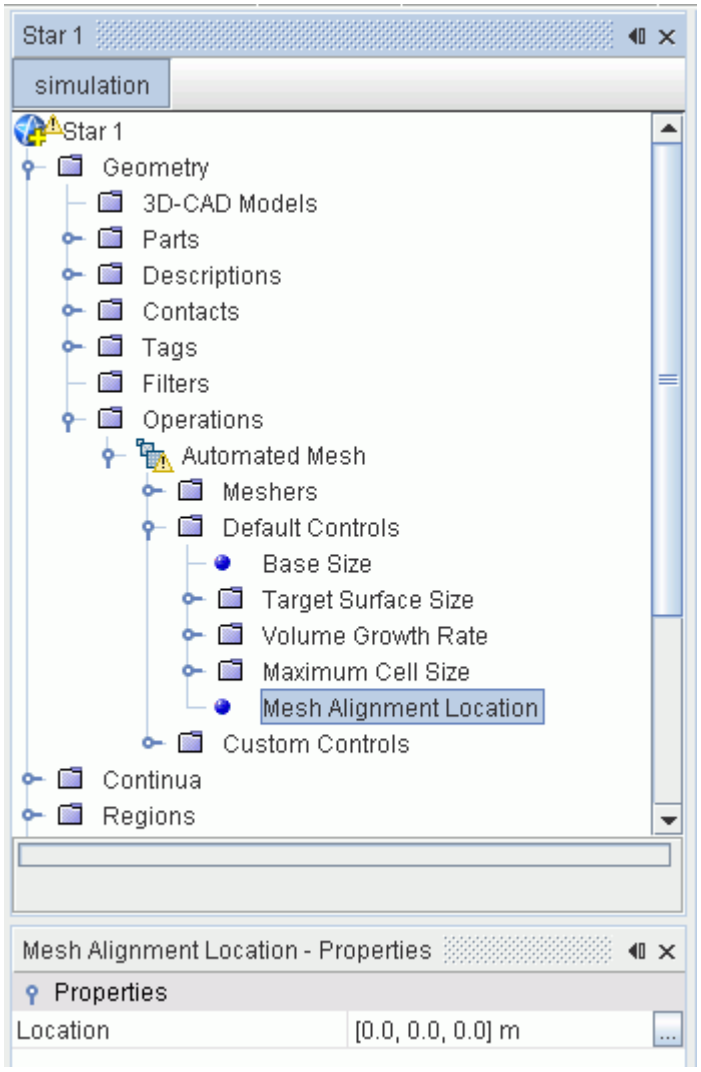
For expert properties of the trimmed mesher, refer to Volume Mesher Controls and Values section.

Coordinate System	The trimmer template mesh is aligned with the global cartesian laboratory coordinate system by default. If the imported surface is not aligned to this coordinate system, then it can be advantageous to realign the template using a user-defined cartesian coordinate system instead. Once a new coordinate system has been defined, it can be

	selected using the *Coordinate System* pull-down menu in the *Properties* dialog:  The trimmed cell mesh is now generated based on the new local coordinate system instead of the global one.
Perform Mesh Alignment (PBM) / Do Mesh Alignment (RBM)	The trimmer mesh alignment option is turned off by default, meaning that the template mesh is centered and aligned based on the global bounds and model alignment in the specified coordinate system. In certain circumstances, such as free surface calculations, it is desirable to align a plane of mesh faces with a physical plane corresponding to a salient feature such as an initial free surface. Such alignment is done, not only in order to make the physical model setup and initialization easier, but also to improve accuracy. Turning the *Perform mesh alignment* (PBM) / *Do mesh alignment* (RBM) option on allows the mesh location coordinates to be specified for the chosen trimmer coordinate system, so that the mesh lines coincide with that location:



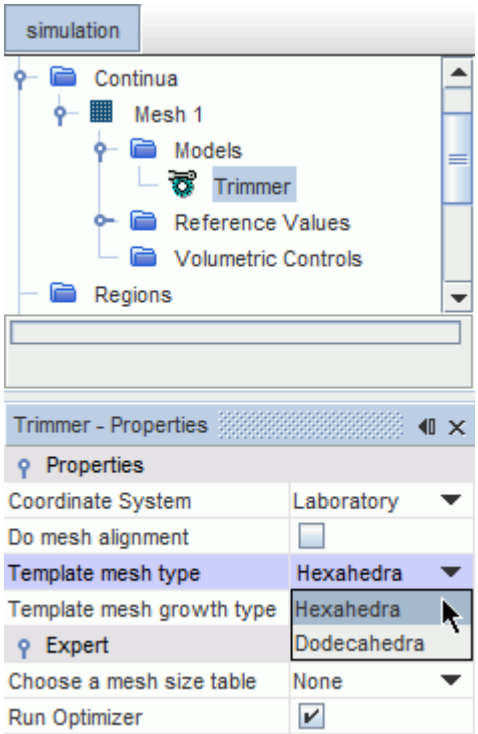
With the perform mesh alignment option turned on, the **Mesh Alignment Location** node under **Default Controls** (PBM)/ **Reference Values** (RBM) can be used to specify a location in the currently specified coordinate system. The trimmer mesh is then aligned so that this location is on a cell corner vertex in the resultant mesh. Selecting the **Mesh Alignment Location** node allows the mesh alignment values to be set:

<table>
<tr>
<td></td>
<td>


The *Location* coordinate can be specified in any of the allowable units. If mesh alignment is only required in one direction, leave the other two location coordinate values set to 0. The current location of the grid alignment point is highlighted in the scene using a circular ball once the **Mesh Alignment Location** has been set.
</td>
</tr>
<tr>
<td>Template Mesh Type (RBM)</td>
<td>
The Template Mesh Type property allows one of two different cell shapes to be used for the underlying template mesh. The two cell shapes are:

hexahedral
dodecahedra

These selections are shown below for the Region Based Meshing approach. Options are found under the Mesh > Models node for the appropriate Continuum:
</td>
</tr>
</table>

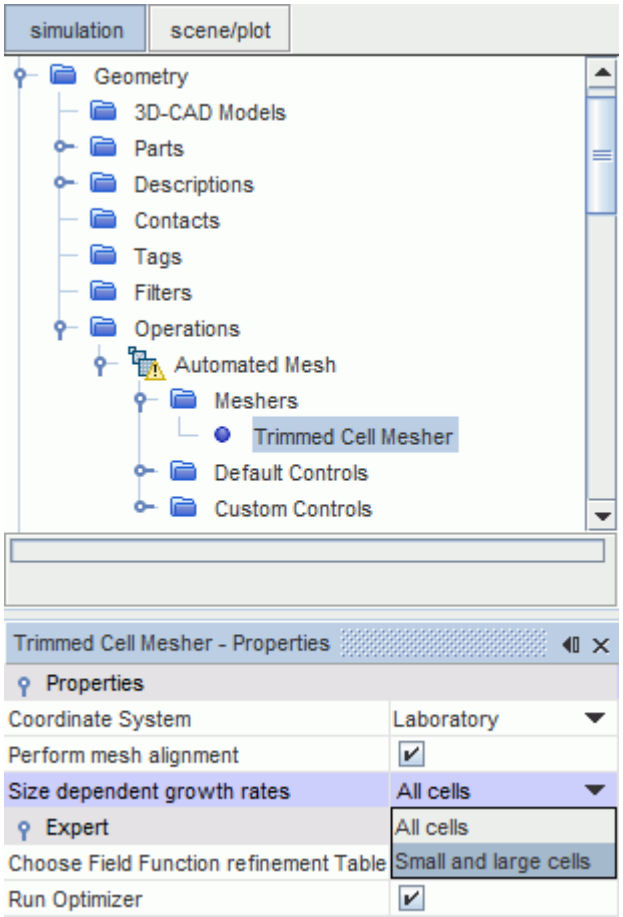


By default, the *Template mesh type* is set to *Hexahedra* meaning that hexahedral shaped cells are used to construct the template. Changing the option to *Dodecahedra* results in a template that is built from regular rhombic dodecahedral cells. A regular rhombic dodecahedral cell has 12 faces, with each face being a rhombus of 4 equal length sides. For PBM, by default, the equivalent template mesh type property is not available to modify in the properties window as it is set to Hexahedra automatically.

To specify the template mesh type as Dodecahedra in PBM, run a macro such as:

```
AutoMeshOperation autoMeshOperation_0 =

((AutoMeshOperation)
simulation_0.get(MeshOperationManager.class).get
Object("Automated Mesh"));

TrimmerAutoMesher trimmerAutoMesher_0 =

((TrimmerAutoMesher)
autoMeshOperation_0.getMeshers().getObject("Trim
med Cell Mesher"));

trimmerAutoMesher_0.getTemplateMeshOption().setS
elected(TemplateMeshOption.DODECAHEDRA);
```

The mesh alignment option cannot be used in conjunction with the dodecahedra template mesher.

Size Dependent Growth Rates (PBM) / Template Mesh Growth Type (RBM)	The *size dependent growth rates / template mesh growth type* property determines the growth rate options that are used to control the refinement next to surfaces and in the core mesh. Two options are provided: • **All cells** (PBM) / **Simple** (RBM) • **Small and large cells** (PBM) / **Two Level** (RBM) These selections are shown below for the Parts Based Meshing approach:  By default, the **Size dependent growth rates** (PBM) / **Template mesh growth type** (RBM) is set to **All Cells** (PBM) / **Simple** (RBM).

Setting Trimmer Default Controls (PBM) / Reference Values (RBM)

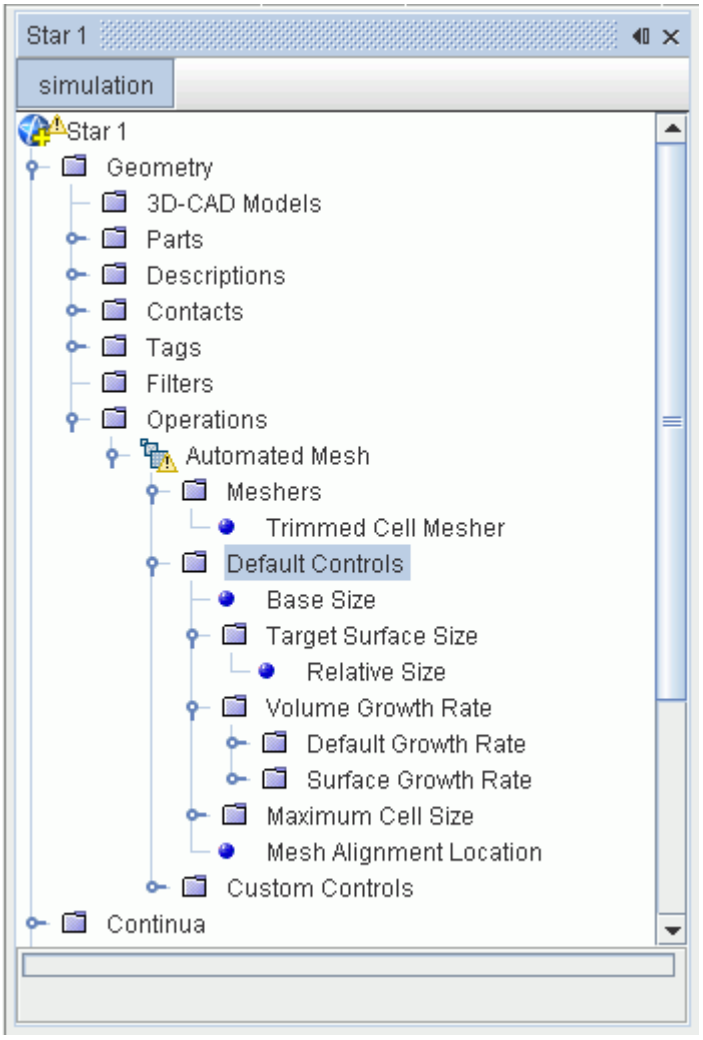
The default controls (PBM) or reference values (RBM) are the global level controls for the trimmer.

The following reference values are applicable for the trimmer:

- base size
- maximum cell size

- mesh alignment location
- surface size
- volume growth rate (PBM) / template growth rate (RBM)

These options can be set within the **Default Controls** node (PBM), or the **Reference Values** node (RBM) in the simulation tree:

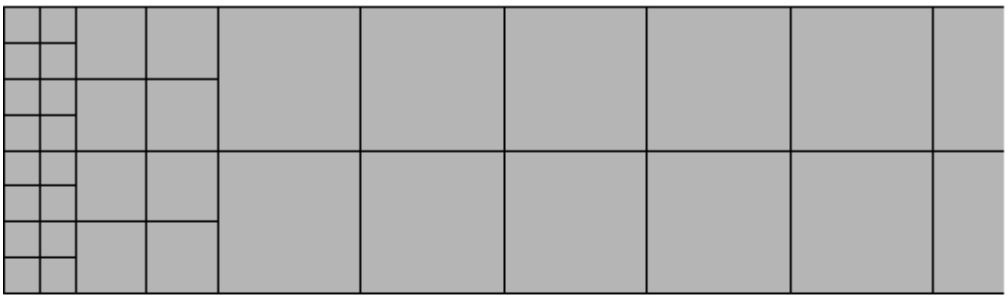


Volume Growth Rate (PBM) / Template Growth Rate (RBM)

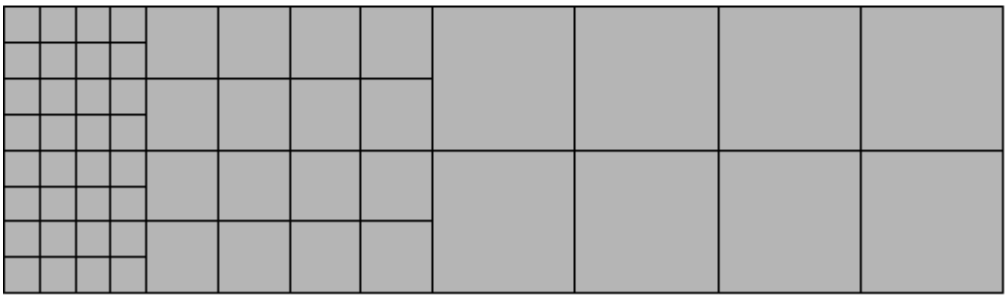
Growth rates control the rate at which cell sizes increase from one cell size to another within the trimmed cell mesh.

A fast growth rate increases cell sizes rapidly; a slow growth rate means that the mesher uses multiple cell layers to provide a gradual transition.

The growth rate defines cell size transition by specifying a minimum number of cell layers of each size. The following diagram illustrates this method:



Medium Growth Rate (2 Layers per Cell Size)

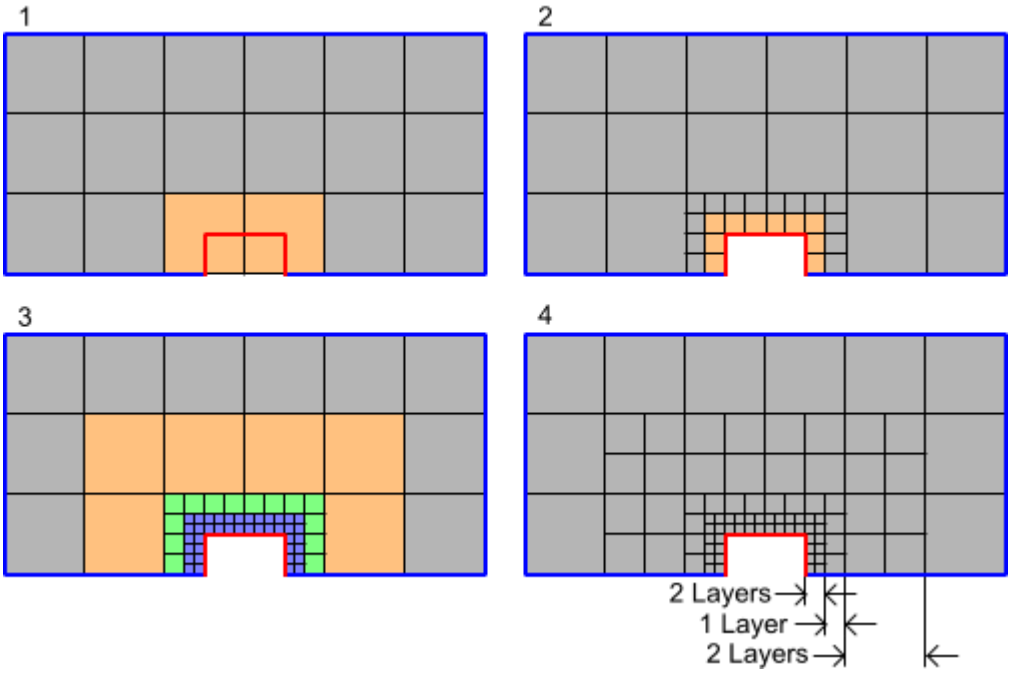


Slow Growth Rate (4 Layers per Cell Size)

To refine cell sizes and maintain connectivity, the mesher subdivides cells into eight smaller cells. In some cases, this method of splitting means that the mesher does not produce the minimum number of cell layers, so the mesh has extra layers. This is shown in the following example. Here, the mesh must meet the following criteria:

- Surface cell size of 1 unit.
- Core cell size of 8 units.
- Fast growth rate, so at least one layer of each cell size (1, 2, 4, and 8 units).

Also, this example uses a two-dimensional mesh, so larger cells split into four smaller cells.



- Image 1: The mesher generates a uniform mesh with cell size 8.
- Image 2: The mesher splits the beige cells in Image 1 to make the cell size closer to surface size on the red boundary.
- Image 3: The mesher performs a second split on the beige cells in Image 2 so that the cell sizes match the surface size.
- Image 4: The blue and green cells in Image 3 are cell layers of 1 and 2 units respectively; however there are no cell layers of 4 units. Therefore, the mesher splits the beige cells in Image 3 to create cells with a size of 4 units.

This final image shows that the splitting of larger cells can result in a number of cell layers that does not match the minimum.

The *Default Growth Rate* controls the rate of size change in the core volume mesh. You set this property for the automated mesh operation (PBM) or mesh continuum (RBM), within the **Default Controls** (PBM) or **Reference Values** (RBM) manager respectively. Two methods of control are available:

- **Volume Growth Rate** (PBM).
- **Template growth rate** (RBM).

The *Surface Growth Rate* (PBM) / *Boundary Growth Rate* (RBM) controls the rate of size changes between the cells that are next to surfaces and the cells in the core. You can set this property at a global or local level on a part surface, boundary, or volumetric control.

This section contains the following topics:

Contents:

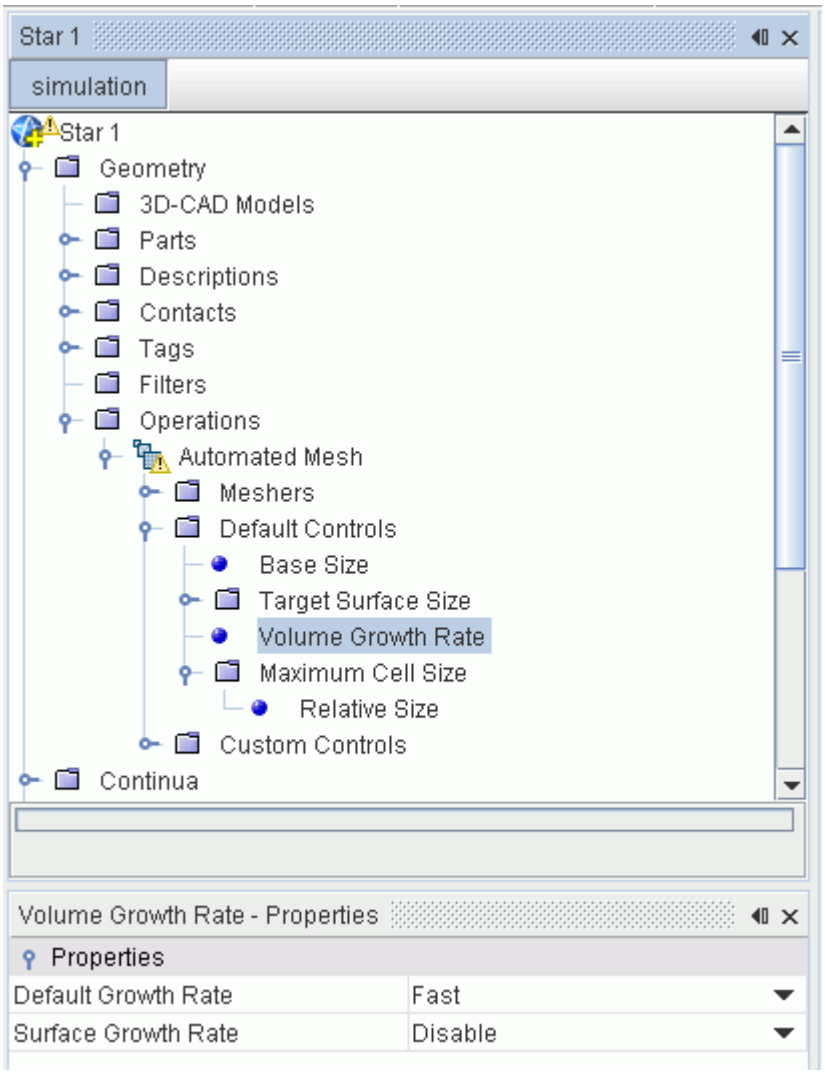
Volume Growth Rate (PBM) / Template Growth Rate (RBM) Properties
Default Growth Rate
Surface Growth Rate (PBM) / Boundary Growth Rate (RBM)

Relative or Absolute Cutoff Size
Custom Growth Rate

Volume Growth Rate (PBM) / Template Growth Rate (RBM) Properties

The **Volume Growth Rate** node (PBM) or **Template Growth Rate** node (RBM) controls the stepping or transition of one cell size to the next within the template that is used for the trimmer.

When the **Trimmed Cell Mesher** (PBM) / **Trimmer**(RBM) node is selected, the *Properties* window displays the *Size dependent growth rates* (PBM) / *Template mesh growth type* (RBM) as**All cells** (PBM) /**Simple** (RBM). Maintaining this default setting then selecting the **Volume Growth Rate** node (PBM) displays the properties as shown below:



Selecting the **Template Growth Rate** node (RBM) with the default **Trimmed Cell Mesher** settings maintained displays similar properties.

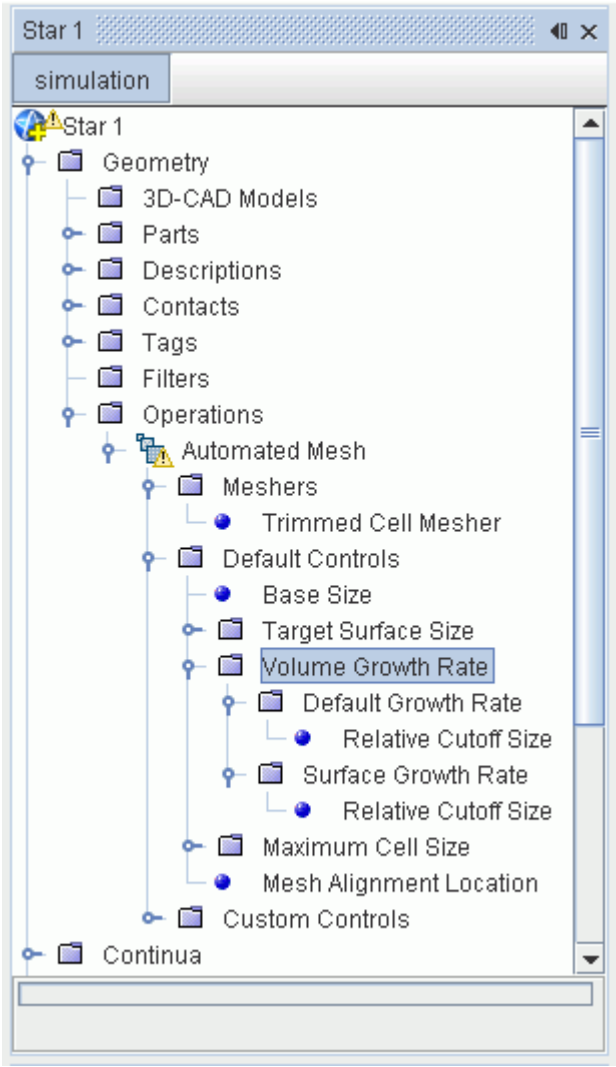
Properties of the Volume Growth Rate (PBM) / Template Growth Rate (RBM)

Default Growth Rate	This option controls the stepping from one cell size to the next within the core mesh. This option can be thought of as non-feature, far field growth control for the core mesh.	
	Fast	Default setting, which means that the fastest transition from the smallest cell size to the largest is used. This setting usually equates to having only one layer of each cell size for each transition, which results in the least number of cells being created for the supplied input values.
	Medium	Gives a minimum of two equal sized cell layers per transition.
	Slow	Gives a minimum of four equal sized cell layers per transition.
	Very Slow	Gives a minimum of eight equal sized cell layers per transition.
	Custom	Adds a Custom Default Growth Rate node on which to set a user-defined number of cell layers per transition.
Surface / Boundary Growth Rate	This option determines the number of cell layers to grow next to surfaces in the template.	
	Disable/None	Default setting, which means that the default growth rate is also applied to cell layers next to surfaces.
	Fast	Sets one cell layer next to surfaces.
	Medium	Sets two cell layers next to surfaces.
	Slow	Sets four cell layers next to surfaces.
	Very Slow	Sets eight cell layers next to surfaces.
	Custom	Adds a Custom Boundary Growth Rate node on which to set a user-defined number of cell layers next to surfaces.

Default Growth Rate

This section describes the properties that you can set for the Default Growth Rate.

If the Size dependent growth rate (PBM) / Template mesh growth type (RBM) property is set to **Small and large cells** (PBM)/**Two Level** (RBM) for the **Trimmed Cell Mesher**(PBM) / **Trimmer** (RBM) meshing model, the **Volume Growth Rate**node (PBM) or**Template Growth Rate** node (RBM) has no properties of its own. Instead, it has two child nodes, one for the **Default Growth Rate**, and another for the **Surface Growth Rate** (PBM) or **Boundary Growth Rate (RBM)**.



Properties on these nodes give a finer level of control over cell size transitions than is possible with the simple template growth rate.

Properties of the Default Growth Rate

Small Cell Cutoff Size Type	Sets the size type for the cutoff size.

	Relative to Base	Set the small cell cutoff size relative to the base size.
	Absolute	Set the small cell cutoff size as absolute value with length units.
Default Growth Rate	This option controls the stepping from one cell size to the next within the core mesh. This option can be thought of as non-feature, far field growth control for the core mesh.	
	Fast	Default setting, which means that the fastest transition from the smallest cell size to the largest is used. This setting usually equates to having only one layer of each cell size for each transition, which results in the least number of cells being created for the supplied input values.
	Medium	Gives a minimum of two equal sized cell layers per transition.
	Slow	Gives a minimum of four equal sized cell layers per transition.
	Very Slow	Gives a minimum of eight equal sized cell layers per transition.
	Custom	Adds a Custom Default Growth Rate node on which to set a user-defined number of cell layers per transition.
Small Cell Growth Rate	This growth rate determines the growth rate in the vicinity of small cells in the template that are less than or equal to the cutoff size.	
	Fast	Default setting, which means that the fastest transition from the smallest cell size to the largest is used. This setting usually equates to having only one layer of each cell size for each transition, which results in the least number of cells being created for the supplied input values.
	Medium	Gives a minimum of two equal sized cell layers per transition.
	Slow	Gives a minimum of four equal sized cell layers per transition.

	Very Slow	Gives a minimum of eight equal sized cell layers per transition.
	Custom	Adds a Custom Small Cell Growth Rate node on which to set a user-defined number of cell layers per transition.

Surface Growth Rate (PBM) / Boundary Growth Rate (RBM)

This section describes the properties of the Surface Growth Rate (PBM) / Boundary Growth Rate (RBM) node

The properties of the **Surface Growth Rate** (PBM) / **Boundary Growth Rate** (RBM) node give access to additional template growth controls for cells next to surfaces of the mesh. The **Surface Growth Rate** (PBM) / **Boundary Growth Rate** (RBM) node is only added under the **Surface Growth Rate** (PBM) / **Template Growth Rate** (RBM) node when the **Small and large cells** (PBM) / **Two Level** option is chosen on the **Trimmed Cell Mesher** (PBM) / **Trimmer** (RBM) meshing model.

Properties of the Surface Growth Rate (PBM) / Boundary Growth Rate (RBM)

Small Cell Cutoff Size Type	Sets the size type for the cutoff size.	
	Relative to base	Sets the small cell cutoff size type relative to the base size.
	Absolute	Sets the small cell cutoff size type as an absolute value with length units.
Surface Growth Rate (PBM) / **Boundary Growth Rate** (RBM)	This setting controls the number of cell layers to use in the transition from cells next to surfaces to cells in the core volume.	
	The trimmer **Surface / Boundary Growth Rate** option determines the number of cell layers to grow next to surfaces in the template.	
	Disable (PBM) / **None** (RBM)	Default setting, which means that the default growth rate is also applied to cell layers next to surfaces.
	Fast	Sets one cell layer next to surfaces.
	Medium	Sets two cell layers next to surfaces.
	Slow	Sets four cell layers next to surfaces.
	Very Slow	Sets eight cell layers next to surfaces.

	Custom	Adds a Custom Surface / Boundary Growth Rate node on which to set a user-defined number of cell layers per transition.
Small Cell Surface Growth Rate (PBM) / **Small Cell Boundary Growth Rate** (RBM)	This growth rate determines the growth rate in the vicinity of small cells in the template that are less than or equal to the cutoff size.	
	Fast	Default setting, which means that the fastest transition from the smallest cell size to the largest is used. This setting usually equates to having only one layer of each cell size for each transition, which results in the least number of cells being created for the supplied input values.
	Medium	Gives a minimum of two equal sized cell layers per transition.
	Slow	Gives a minimum of four equal sized cell layers per transition.
	Very Slow	Gives a minimum of eight equal sized cell layers per transition.
	Custom	Adds a Custom Small Cell Growth Rate node on which to set a user-defined number of cell layers per transition.

Relative or Absolute Cutoff Size

You can set a relative or absolute small cell cutoff size.

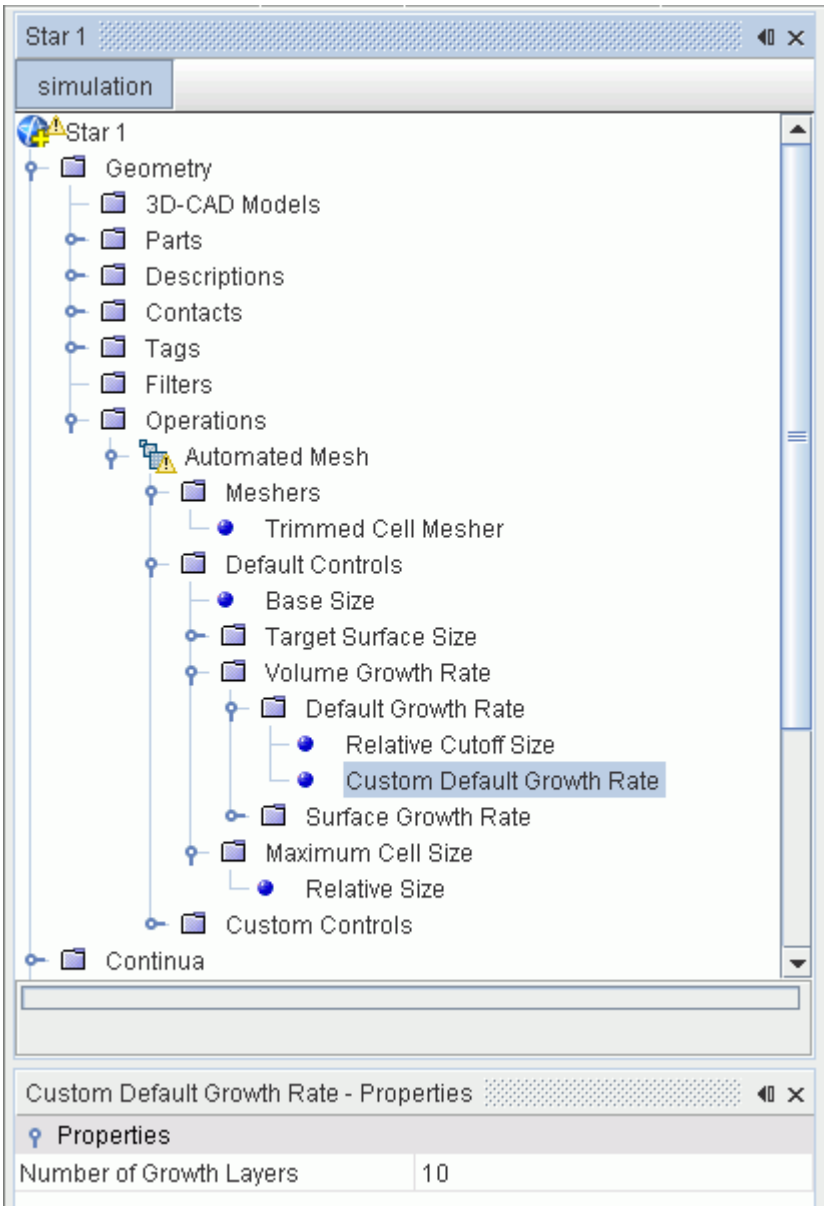
It is possible that refinements due to the initial surface mesh size, or volumetric controls can result in small cells whose size is less than the target size in the template. If the small and large cells growth rate (PBM), or the Two Level (RBM) property has been activated for the **Trimmed Cell Mesher** (PBM) / **Trimmer** (RBM) model, then a relative or absolute small cell cutoff size can be set for both the **Default Growth Rate** and the **Surface Growth Rate** (PBM) / **Boundary Growth Rate** (RBM). For cells smaller than the small cell cutoff size, the trimmer applies the small cell growth rate to create transition cell layers between the small cells and the rest of the mesh.

The small cell growth rate is set as a property of the nodes for **Default Growth Rate** or **Surface Growth Rate** (PBM) / **Boundary Growth Rate** (RBM). On surfaces or boundaries, the growth rate applies to transition cell layers that are grown from small cells on the surface. Elsewhere, the default small cell growth rate applies.

Custom Growth Rate

If the built-in choice of growth rates fails to provide sufficient control over the number of cell layers that are used for transition between sizes, a custom option can be selected instead.

This option adds a custom growth rate child node whose single property sets the number of cell layers to use in size transitions. An example of this node for a custom default growth rate in the **Reference Values** section is shown below:



Properties of the Custom Growth Rate

Number of Growth Layers	This property sets the number of cell layers to use in the transition from one cell size to another. In the case of a custom boundary growth rate,

	this property applies to the transition in size between cells on a surface and cells in the core.

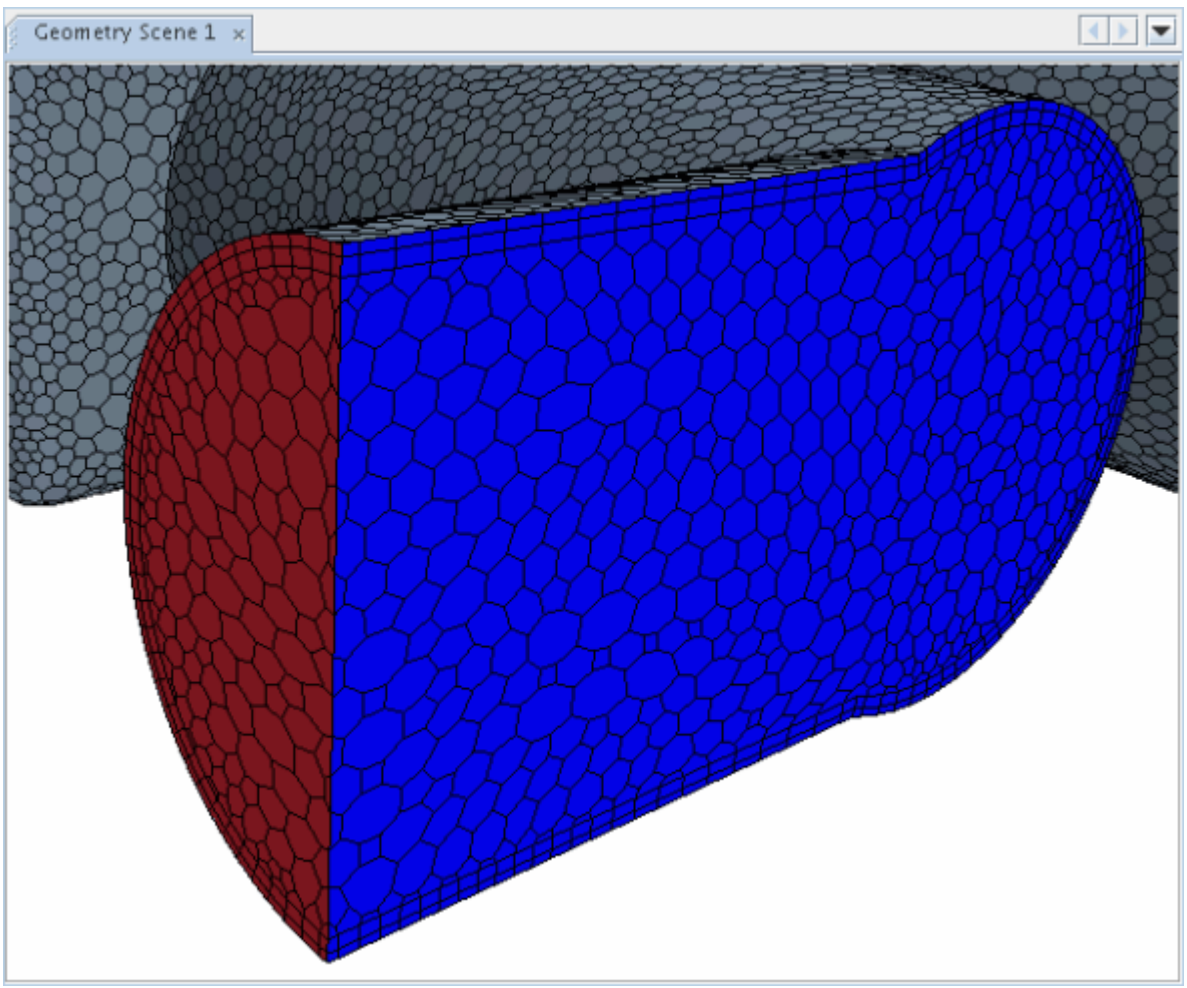
Prism Layer Mesher

The prism layer mesh model is used with a core volume mesh to generate orthogonal prismatic cells next to wall surfaces or boundaries. This layer of cells is necessary to improve the accuracy of the flow solution.

A prism layer is defined in terms of:

- Its thickness
- The number of cell layers within it
- The size distribution of the layers
- The function that is used to generate the distribution -- either by geometric progression or hyperbolic tangent.

These general properties can be defined globally within an automated mesh operation or a mesh continuum, and can also be defined locally on part surfaces, regions, or boundaries, and within volumetric controls. An example volume mesh showing two prism layers next to the wall surface is shown below:

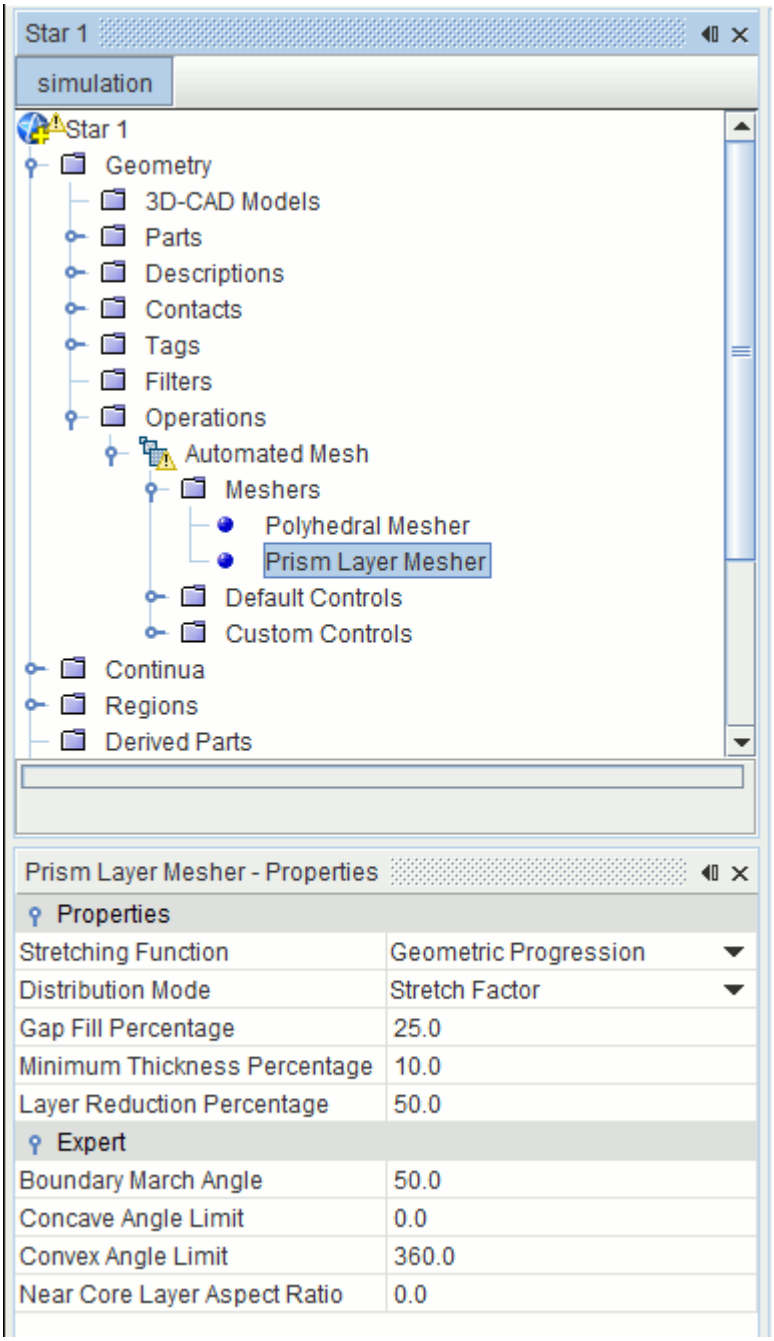


By default, prism layers are not created next to boundaries (or associated part surfaces) unless they are of the wall boundary type. This setting can be changed on a boundary by boundary, or surface by surface basis.

Prism cells are not created on boundary surfaces when there is more than one boundary in a thin portion of geometry. Instead, prism layers are created inside the volume which avoids confusion between prism layers that belong to different boundaries. See Troubleshooting the Thin Meshers: Multiple Boundaries in a Thin Geometry.

By default, STAR-CCM+ aims to generate a conformal prism layer mesh at interfaces of type in-place, periodic, or baffle, regardless of the number of prism layers at each side of the interface.

When the prism layer meshing model is activated, its default and expert properties can be accessed by selecting the **Prism Layer Mesher** node. This node can be found under the **Meshers** node for the **Automated Mesh** operation if using Parts Based Meshing, or under the **Models** node for the mesh continuum if using Region Based Meshing.



Why Is a Prism Layer Mesh Required?

Prism layers allow the solver to resolve near wall flow accurately, which is critical in determining not only the forces and heat transfer on walls, but also flow features such as separation. Separation in turn affects integral results such as drag or pressure drop. Accurate prediction of these flow features depends on resolving the velocity and temperature gradients normal to the wall. These gradients are much steeper in the viscous sublayer of a turbulent boundary layer than would be implied by taking gradients from a coarse mesh. Using a prism layer mesh allows you to resolve the viscous sublayer directly if the turbulence model supports it (low y+ ~1). Alternatively, for coarser meshes it allows the code to fit a wall function more accurately (high y+ > 30).

The turbulence modeling approach that is used and the desired fidelity of the physics determine the thickness, number of layers and distribution of the prism layer mesh. Depending on the Reynolds number, a turbulent shear layer requires in excess of 10-20 cells in the cross-stream direction for accurate resolution of the turbulence flow profiles. To resolve the viscous sublayer (that is, low y+ wall treatment), more cells are required. If only the gross flow features (such as a first-order estimate of skin friction) are required, coarser, high y+ wall-function type meshes with just a few prism layers can yield acceptable results.

Prism layers do not only provide near wall mesh density, they also allow high-aspect-ratio cells to be used, thus providing better cross-stream resolution without incurring an excessive stream-wise resolution.

Prism layers also reduce numerical diffusion near the wall. Numerical diffusion is a discretization error that smears discontinuities and steep gradients in a finite volume advection scheme. Numerical diffusion is minimized when the flow is aligned with the mesh. The use of prism layers greatly enhances accuracy as a result.

How is the Prism Layer Generated?

Before the core volume mesh is created, a subsurface is generated at the specified prism layer thickness values, in effect “shrinking” (for internal flows) or “expanding” (for external flows) the starting surface. The core mesh is built using this subsurface. The prism layer mesh is then generated by extruding the cell faces from the core mesh back to the original starting surface.

When the prism mesher is used with a periodic interface, then it attempts to produce a conformal mesh for the boundary pair where possible. The inclusion of a periodic interface overrules the boundary type that is specified for the boundary pair in this instance. Instead, the interface level prism layer option determines whether a prism mesh is produced or not. By default, no prism layers are produced on interfaces.

Contents:

Controlling the Construction of the Prism Layer
Global Controls
Customizing Prism Layers
Improving Prism Layer Construction in Gaps and Corners
View Prism Layer Cells Using a Cell Set
Prism Layer Model Properties
Prism Layer Model Expert Properties
Prism Layer Mesh Bibliography

See Also:

Advancing Layer Mesher
Generalized Cylinder Mesher

Controlling the Construction of the Prism Layer

The properties of the prism layer mesher model control different aspects of prism layer construction.

They can be split into two categories: the properties that control the general prism layer construction (such as the number of layers); and the properties that govern the behavior of the mesher when constructing a prism layer is difficult.

The following properties that govern general prism layer construction:

- Setting the number of cell layers
- Setting the thickness of the prism layer
- Setting prism layer stretching
- Setting transition ratios between prism layers and trimmed mesh

Global Controls

This section introduces the properties that control the general construction of the prism layer.

Properties that control the general construction of the prism layer can be set globally using the nodes within the **Automated Mesh** > **Default Controls** node when using Parts Based Meshing. If using Region Based Meshing, the same controls are within the **Reference Values** node of the parent mesh continuum.

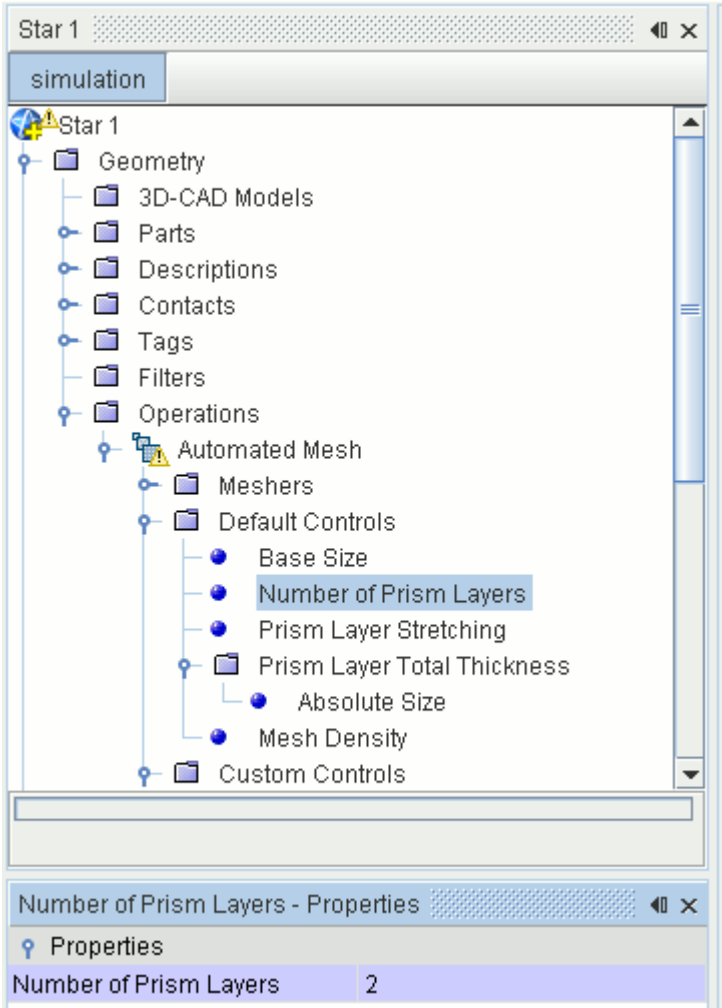
Contents:

Number of Prism Layers
Prism Layer Total Thickness
Prism Layer Stretching
Maximum Core Prism Transition Ratio

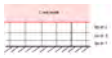
Number of Prism Layers

The number of prism layers parameter controls the number of cell layers that are generated within the prism layer on a part surface or boundary.

This number is set using the **Number of Prism Layers** node.



Number of Prism Layers Properties

Number of Prism Layers	This property sets the number of prism layers to grow unless over-ridden by local settings or quality limits. The default value for *Number of Prism Layers* is to *2*. Any whole value equal to or greater than 1 can be entered. The example below shows a prism mesh with three layers.  When specifying custom values, this node is added to the **Custom Controls** > **[Surface]** > **Controls** node for Parts Based Meshing. When using Region Based Meshing it is added to the **Mesh Values** node of a region or boundary.

Prism Layer Total Thickness

This section describes the Prism Layer Total Thickness properties.

The prism layer thickness controls the total overall thickness of all the prism layers.

Prism Layer Total Thickness Properties

Prism Layer Total Thickness	Selecting the **Prism Layer Total Thickness** node allows the choice of a relative or absolute *Size type* to be made. In the *Prism Layer Total Thickness - Properties* dialog set the **Size type** to **Relative Size** or **Absolute Size** to provide the value of the thickness. The example below shows how the prism layer thickness is measured. When specifying custom values, a **Prism Layers** node is added to the **Custom Controls** > **[Surface Control]** > **Controls** node for Parts Based Meshing. When using Region Based Meshing, it is added to the **Mesh Values** node of a region or boundary.	
	Relative to Base	Sets the prism layer total thickness relative to the base size.
	Absolute	Sets the prism layer total thickness as absolute value with length units.

Prism Layer Stretching

Prism layer stretching sets the target growth rate of successive prism layers away from the wall.

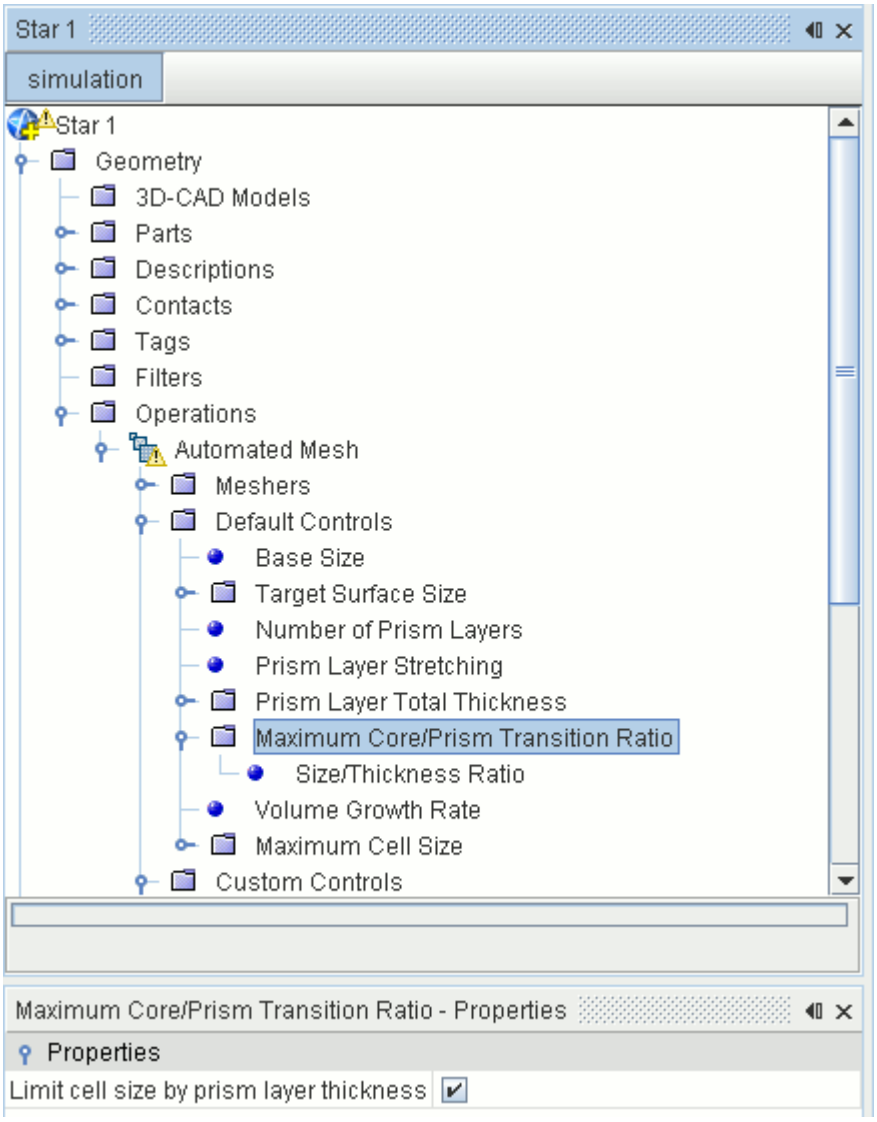
Prism Layer Stretching Properties

Prism Layer Stretching	The default **Prism Layer Stretching** value is 1.5.

Maximum Core Prism Transition Ratio

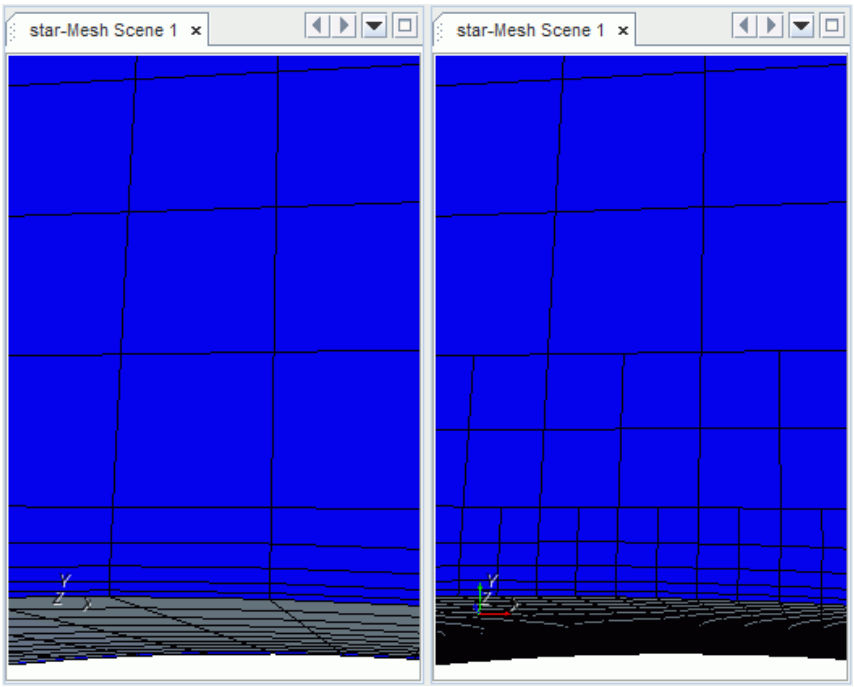
When using the prism layer mesh model with the trimmer model, you can control the ratio between the trimmed and prism layer cell sizes.

This feature, along with the **Near Core Layer Aspect Ratio** option, is designed to help you generate smoother cell volume transitions between the trimmed and prism layer meshes. It is set by activating the checkbox for the **Limit cell size by prism layer thickness** in the *Properties* window of the **Maximum Core/ Prism Transition Ratio** node.



Maximum Core/Prism Transition Ratio Properties

Maximum Size/Thickness Ratio	STAR-CCM+ adds a **Size/Thickness Ratio** node, which lets you set the limiting value for the ratio between the two type of mesh cells. The default value is 5.0, that is, trimmed cells are refined so that they are at most five times the size of the neighboring prism layer cells. The minimum ratio value that you can enter is 2.0. Smaller values result in increased cell count. The effect of using the **Size/Thickness Ratio** option is shown below. The left image is with this feature turned off, and the right shows that the additional refinement has reduced the volume change between the trimmed and prism layer cells.



Customizing Prism Layers

Custom controls can be set for particular part surfaces, individual regions, or boundaries.

Custom controls can be set for particular part surfaces under the **Automated Mesh** > **Custom Controls** node when using Parts Based Meshing. When using Region Based Meshing, the same controls can be set under individual regions or boundaries using the **Customize Prism Mesh** node that exists within the **Mesh Conditions** node for these objects.

Identical prism layer options are provided for part surfaces, regions, boundaries, and volumetric controls.

By default, prism layers are only generated on all wall boundaries or related part surfaces. They are not generated for any other type of boundary or related part surface unless you specifically activate this using custom controls. Activating a prism layer on a part surface or boundary exposes value nodes for the number of layers, total thickness, and layer distribution. For boundaries that lie on an interface, an additional control must be considered, which is the Interface Prism Layer Option that sets whether to grow prism layers from an interface or not. Custom values can also be supplied based on volumetric controls.

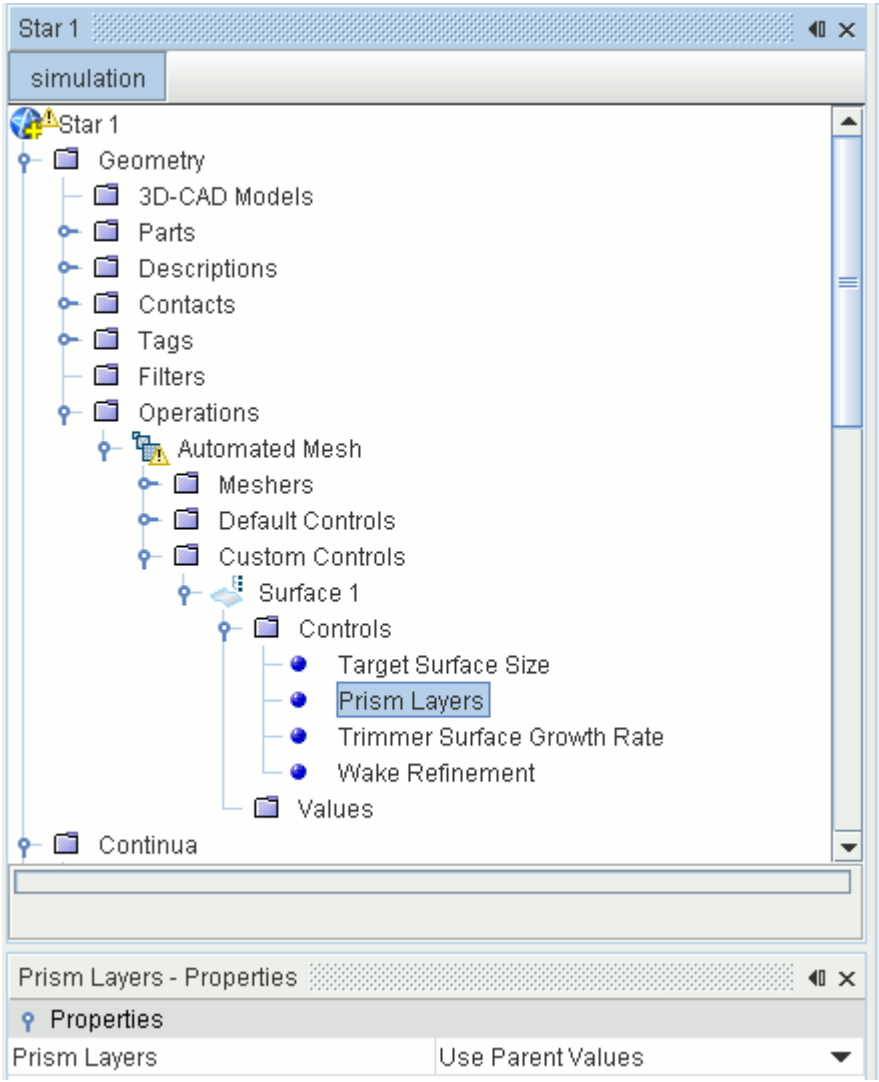
Contents:

Prism Layers (PBM) / Customize Prism Mesh (RBM)
Interface Prism Layer Option

Prism Layers (PBM) / Customize Prism Mesh (RBM)

Custom controls can be activated for particular part surfaces, regions, or boundaries.

The **Custom Controls** > **[Part Surface]** > **Controls** > **Prism Layers** node (PBM) or the **Mesh Conditions** > **Customize Prism Mesh** node (RBM) allow custom controls to be activated for particular part surfaces, regions, or boundaries. A single property then controls what happens on the surface regarding the prism layer.



If the *Specify Custom Values* option is selected, additional properties are activated in the **Prism Layers** > **Customize** (PBM) node or the **Mesh Values** node (RBM) which allow specific prism layer settings to be made.

Properties of the Customize Prism Mesh Node

Prism Layers (PBM) / Customize Prism Mesh (RBM)	Use this property to control the prism layer mesher on a part surface, region, or boundary.	
	Use Parent Values (PBM) / Use Default Values (RBM)	The default setting. Apply the global, higher-level settings, no local customization.

	Specify Custom Values	Choose to define custom controls for the part surface, region, or boundary. Appropriate properties or value nodes are added to the part surface, region, or boundary.
	Disable	Do not create prism layers on this part surface, region, or boundary.

Custom controls set at the local level supersede controls that are set at a global or higher level. Custom controls set on a region are applied to all boundaries within the region unless they have their own custom controls, which in turn supersede the region controls.

For multi-region interfaces, the interface prism layer option supersedes the region level setting for these boundaries and provides an additional level of control. For example, if the generation of prism layers has been deactivated for a region, prismatic layers can still be generated for boundary interfaces using this option using values that are specified at either the boundary or global level.

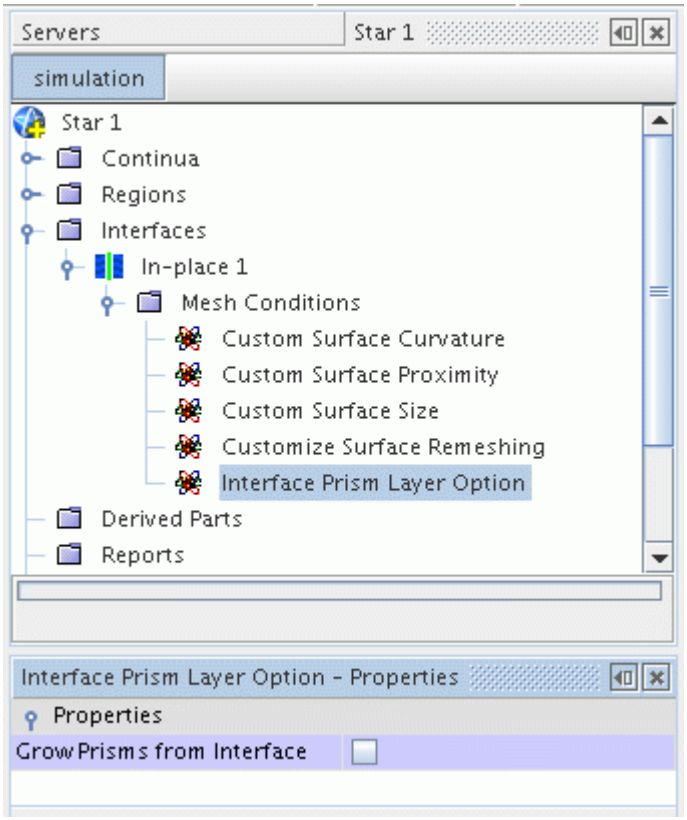
The region type specification, namely fluid, porous, or solid, has no influence on the behavior of the prism mesh settings.

Interface Prism Layer Option

The prism layer settings at the interface level allow you to decide whether prism layers are grown from the interface or not during the volume mesh generation process.

Certain applications such as those that include a porous or a rotating region do not require prism layers at the interface. Others, such as interfaces between fluid and solid regions do require a prism layer on one or both sides of the interface.

The option to activate them or not is provided via the **Interface Prism Layer Option** node, which is located within each interface node that exists. The default state for the *Grow Prisms from Interface* option is off. This option is shown below:



Custom prism values specified for a boundary belonging to an interface are only included when the Grow Prisms from Interface option has been activated.

Improving Prism Layer Construction in Gaps and Corners

This section introduces the controls that help you to improve prism layer construction.

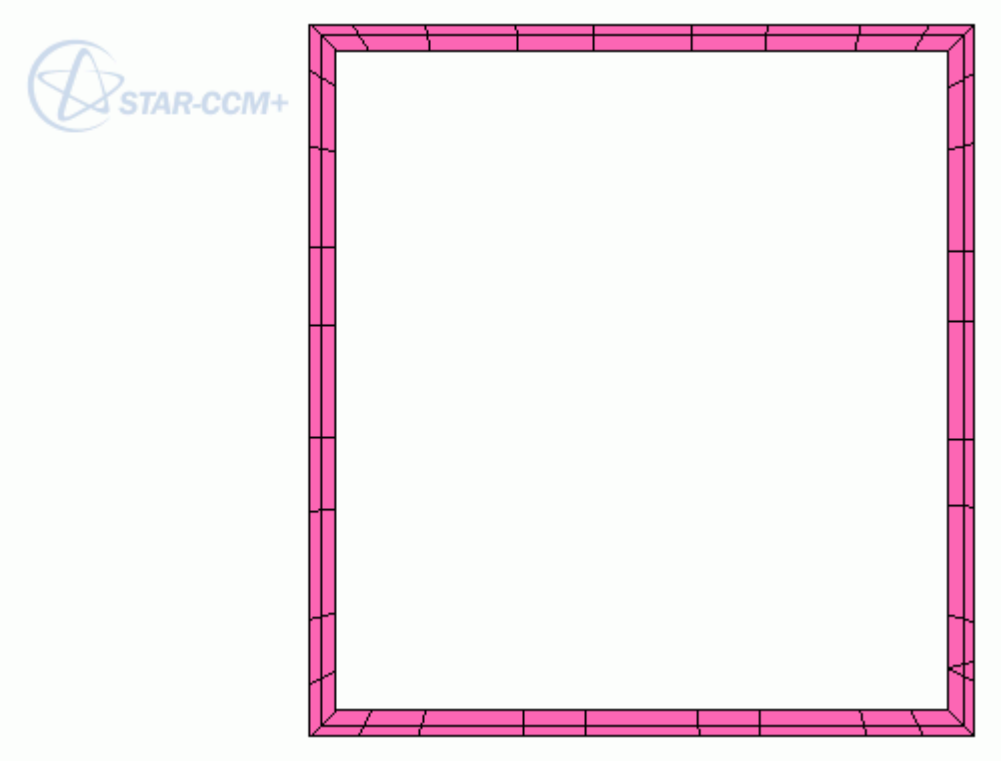
For some geometries, it is not possible to generate a prism layer of acceptable quality on all surfaces, or at least not with the same thickness or number of layers. The prism layer mesher has several controls that determine whether prism layer thickness is reduced, or the number of layers are reduced when local conditions require it. The following controls are provided:

- Gap Fill Percentage
- Minimum Thickness
- Layer Reduction Percentage
- Boundary March Angle
- Concave and Convex Angle Limits
- Generating Standard Cells Only
- Improve Subsurface Quality

View Prism Layer Cells Using a Cell Set

STAR-CCM+ provides a field function that allows you to view cells that are generated by the Prism Layer Mesher. You can use the same function to view prism layers that the Advancing Layer Mesher model creates.

Use this function in combination with a cell set to define the selection of prism layer cells that you want to view in a displayer.



You can view prism layer cells as a cell set in cases which use the prism layer meshing model combined with the following volume meshing models:

- Polyhedral Mesher
- Generalized Cylinder Mesher (also with Polyhedral Mesher)
- Tetrahedral Mesher
- Trimmer Mesher
- Thin Mesher

At present it is not possible to use the prism layer cells field function if you use the following meshing models:

- Extruder Mesher
- Embedded Thin Mesher

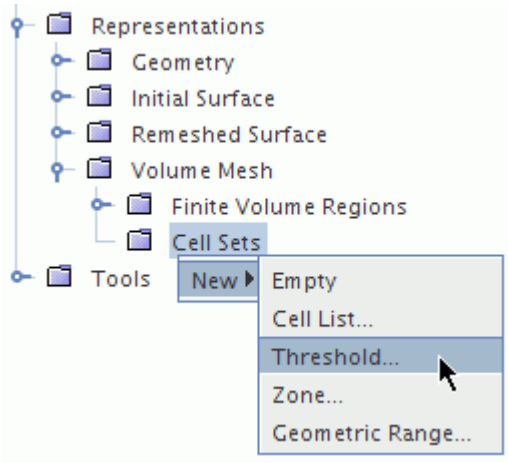
Before Creating the Cell Set:

1. The mesh that you use for this feature must be created in STAR-CCM+ v8.06 or later. If necessary, regenerate the mesh in STAR-CCM+ v8.06 or later.

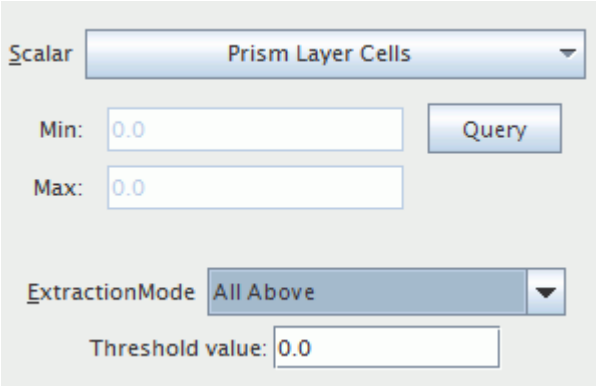
2. This feature does not support imported volume meshes. If necessary, regenerate any imported volume meshes using STAR-CCM+ v8.06 or later.
3. Ensure that your geometry parts are assigned to regions and that they are meshed using the Prism Layer Mesher model.

To create the cell set:

4. Right-click **Representations** > **Volume Mesh** > **Cell Sets**.
5. Select **New** > **Threshold**.



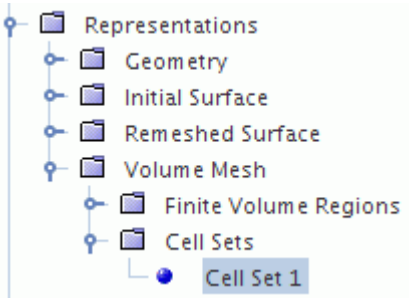
6. In the *Threshold* dialog that appears, set:
   1. *Scalar* to **Prism Layer Cells**.
   2. *Extraction Mode* to **All Above**.



   The *Threshold* value is set to zero by default. However, when *Scalar* is set to **Prism Layer Cells**, there is no need to specify the *Threshold value* as it is automatically set to display all prism layer cells.
7. Set *Display* to **New Geometry Displayer**, then click **OK**.

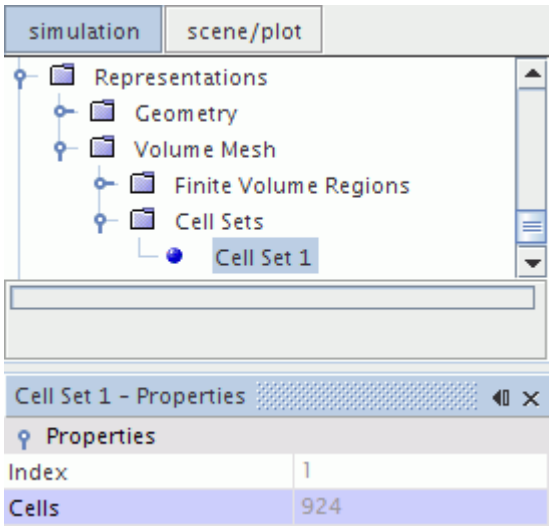
   Selecting **New Geometry Displayer** while a mesh scene is active automatically creates a cell set geometry displayer in the mesh scene.

   A **Cell Set 1** node appears under the **Cell Sets** manager.



8. To determine the number of cells in the cell set, select the **Cell Sets** > **Cell Set** node.

   The number of cells in the cell set is displayed in the *Properties* window, to the right of the *Cells* property.



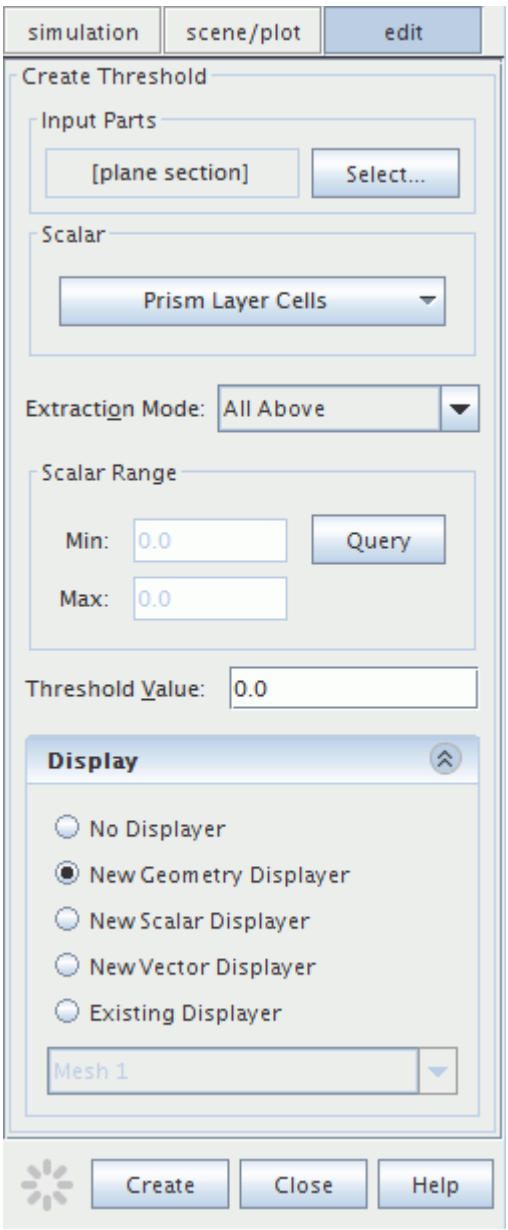
To view the cell set:

9. Select the **Scenes** > **Mesh Scene** node and click the *scene/plot* tab.
10. Expand the**Mesh Scene** > **Displayers**node and multi-select all nodes except the **Cell Set Geometry** node.
11. Right-click the selected nodes and select **Toggle Visibility**.
12. Return to the *simulation* tab.

    The prism layer mesh cell set is displayed in the *Graphics* window.

To view a plane section of the cell set:

13. Create a plane section of the cell set.
14. Create a threshold part for the cell set, using the following settings:
    1. Set *Input Parts* to **plane section**
    2. Set *Scalar* to **Prism Layer Cells**
    3. Set *Extraction Mode* to **All Above**
    4. Set *Display* to **New Geometry Displayer**



You can use the resulting derived part in a scene displayer to visualize the cross-section of the prism layer.

Prism Layer Model Properties

This section describes the properties of the Prism Layer model.

Stretching Function	Selects the stretching function to use in calculating the thickness of each cell layer within the prism layer.	
	Geometric Progression	The thickness of each cell layer is calculated using a geometric

		progression that is based around a constant size ratio. Selecting this option, sets the **Default Controls** > **Prism Layer Stretching** property to 1.5 by default.
	Hyperbolic Tangent	The thickness of each cell layer is calculated using a hyperbolic tangent stretching function. Selecting this option, sets the **Default Controls** > **Prism Layer Stretching** property to 0.5 by default.
Distribution Mode/ Stretching Mode	Selects the distribution/stretching mode to use in specifying the relative size parameters that the stretching function requires.	
	Stretch Factor	The definition of the stretch factor depends on which stretching function is being used. Adds the **Prism Layer Stretching** node for global and custom settings.
	Wall Thickness	Sets the thickness of the near-wall cell layer. Thicknesses of the remaining layers are distributed according to a calculated stretch factor. Adds the Thickness of Near Wall Prism Layer node for global and custom settings.
	Thickness Ratio	Sets the ratio of the thickness of the cell layer furthest from the wall to the thickness of the cell layer nearest the wall. Adds the Prism Layer Thickness Ratio node for global and custom settings.
Gap Fill Percentage	When prism layers are generated in a narrow passage (or gap), it is possible that the prism layer could become larger than the gap. The gap fill percentage sets the maximum allowable proportion of a gap that the prism layer mesh can fill.	
Minimum Thickness Percentage	The minimum thickness percentage is used to calculate a threshold thickness below which the prism layer is retracted.	
Layer Reduction Percentage	Used to calculate a threshold thickness below which the number of cell layers is progressively reduced.	

Contents:

Working with the Stretching Function
Working with the Distribution/Stretching Mode
What is the Gap Fill Percentage?
What is the Minimum Thickness?
What is Layer Reduction?

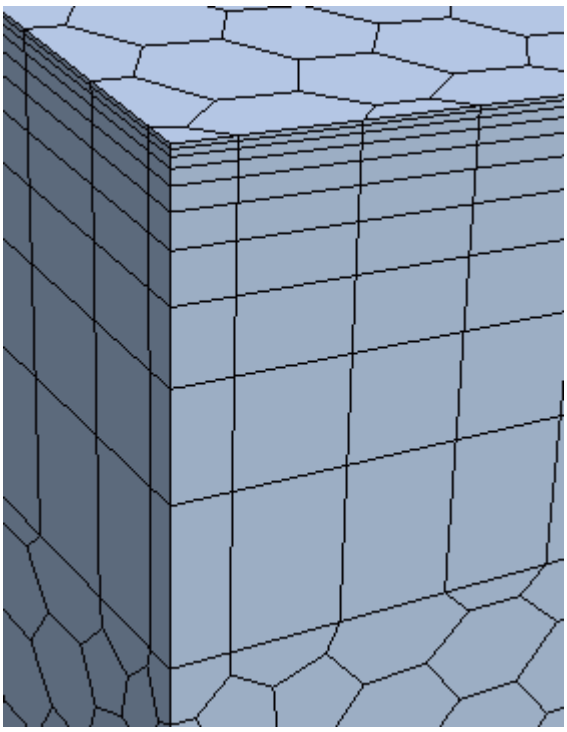
Working with the Stretching Function

The stretching function determines the underlying formula that is used to generate the cell layer thickness distribution.

Two options are provided: geometric progression and hyperbolic tangent.

Geometric Progression

This default option distributes the cell layers using simple geometric relationships. The stretching factor between adjacent cell layers is always calculated as a constant value. An example of a prism layer that is constructed using 10 layers with geometric progression is given below. A **Prism Layer Stretching** factor of 1.5 (default) is used for the entire prism layer.



Hyperbolic Tangent

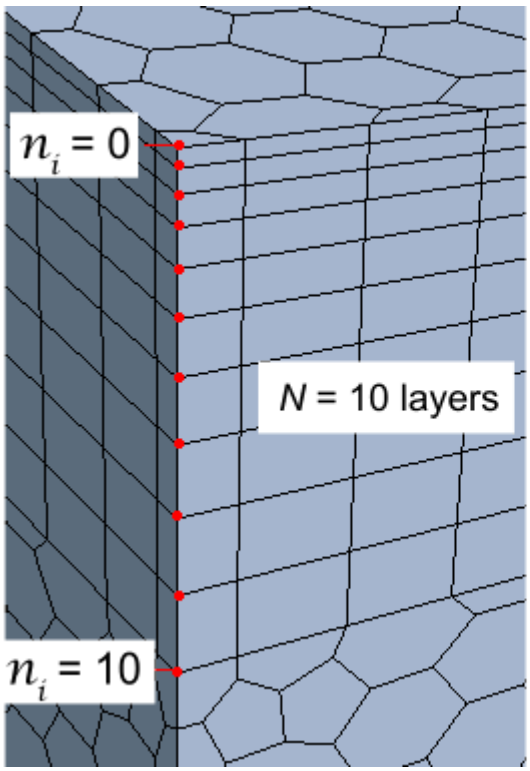
This option distributes the cell layers according to a one-parameter hyperbolic tangent stretching function:

$$s(n_i) = 1 + \frac{\tanh[F_s(n_i/N - 1)]}{\tanh(F_s)} \tag{86}$$

where N is the total number of layers, n_i is the current node, F_s is the overall stretching factor, and $s(n_i)$ is the distribution value for node n_i. Note that $0 \leq n_i \leq N$, so that for 5 layers, n_i takes the values 0, 1, 2, 3, 4, 5.

For more information, see Prism Layer Mesh Bibliography.

The relation between nodes and layers is depicted in the diagram below.



An overall **Prism Layer Stretching** factor of 1.5 for 10 layers gives a distribution as shown above. The size ratio between layers is no longer constant, but is modified to give a smoother transition between cell thicknesses across the whole prism layer. This option allows tighter spacing of near-wall prism layers while avoiding large spacing of the outer prism layers.

See Also:

Advancing Layer Mesher Properties

Working with the Distribution/Stretching Mode

The distribution/stretching mode is a property of the **Prism Layer Mesher** node. It controls the method that is used to calculate the thickness distribution of multiple cell layers within the prism layer.

The mode cannot be set separately for local part surfaces, regions, and boundaries--it is only set once at a global level. However, the chosen option determines the type of node to appear in the **Default Controls** node

of the **Automated Mesh** operation when using Parts Based Meshing. When using Region Based Meshing, it determines the type of node to appear in the **Reference Values** node of the mesh continuum.

These three options, and their corresponding value nodes, are as follows:

Distribution Mode Option	Corresponding Value Node
Stretch Factor	**Prism Layer Stretching**
Wall Thickness	**Thickness of Near Wall Prism Layer**
Thickness Ratio	**Prism Layer Thickness Ratio**

Contents:

Stretch Factor
Wall Thickness
Thickness Ratio

Stretch Factor

When a geometric progression is used for the stretching function, the *Stretch Factor* is the ratio of the thickness of a cell layer to the thickness of the preceding layer.

The minimum allowed value is 1.0, which results in uniformly spaced prism layers. Any value greater than 1.0 results in non-uniformly spaced prism layers with a low prism layer height nearer to the wall which increases progressively in each prism layer away from the wall.

For example, if the stretch factor is 2.0, and the thickness of the first prism layer next to the wall is 0.5 units, the thickness of the next layer would be 1 unit. Continuing to move away from the wall would give thicknesses of 2, 4, 8, and so on.

When the stretching function is set to hyperbolic tangent, the stretch factor becomes a parameter that applies within the stretching function. The actual size ratio between cell layers is not a constant, but is determined by the distribution function.

You can set *Prism Layer Stretching* values according to the following table:

Stretching Function	Valid Values
Geometric Progression	The default value is 1.5. A value between 1.2 through 1.5 is recommended, although any value >= 1.0 is valid.
Hyperbolic Tangent	The default value is 0.5 within default controls and 1.5 within custom surface controls. A value between 2.0 through 5.0 is recommended, although any value > 0.0 is valid.

In addition to the global control, local control is available.

When using Parts Based Meshing:

1. Set the **Custom Controls** > **[Surface]** > **Controls** > **Prism Layers** property to **Specify Custom Values**.
2. In the **Prism Layers** > **Customize** node, activate **Customize Distribution**.
3. Under the **Values** > **Custom Prism Values** node, a local value of the **Prism Layer Stretching** can be set on the chosen part surface.

When using Region Based Meshing:

4. Set the **[Boundary]** > **Mesh Conditions** > **Customize Prism Mesh** property to *Specify Custom Values*.
5. Under the **Mesh Values** node, a local value of the **Prism Layer Stretching** can be set on the chosen boundary.

See Also:

Advancing Layer Mesher Properties

Wall Thickness

The *Wall Thickness* option sets the thickness of the layer that is immediately next to the wall boundary (called the near wall layer).

In order to calculate the appropriate thickness distribution, a stretch factor is automatically calculated and applied to the remaining prism layers (between the near wall layer and the start of the core mesh). The *Wall Thickness* value cannot be greater than the total prism layer thickness divided by the number of layers. If a larger value is supplied, then a linear distribution is used.

The initial wall thickness must always be greater than zero, regardless of the stretching function being used.

On selecting *Wall Thickness* for the distribution/stretching mode, the **Prism Layer Near Wall Thickness** node is added to the **Default Controls** node when using Parts Based Meshing. When using Region Based Meshing, a node **Thickness of Near Wall Prism Layer** is added under the **Reference Values** node of the mesh continuum. These nodes are used to set the global near wall layer thickness.

In addition to the global control, local control is available.

When using Parts Based Meshing:

1. Set the **Custom Controls** > **[Surface]** > **Controls** > **Prism Layers** property to **Specify Custom Values**.
2. In the **Prism Layers** > **Customize** node, activate **Customize Distribution**.
3. Under the **Values** > **Custom Prism Values** node, a local value of the **Prism Layer Near Wall Thickness** can be set on the chosen part surface.

When using Region Based Meshing:

4. Set the **[Boundary]** > **Mesh Conditions** > **Customize Prism Mesh** property to **Specify Custom Values**.
5. Under the **Mesh Values** node, a local value of the **Thickness of Near Wall Prism Layer** can be set on the chosen boundary.

Thickness Ratio

The *Thickness Ratio* is the ratio of the thickness of the prism layer next to the core mesh to the thickness of the prism layer next to the wall.

A stretch factor is automatically calculated for the intermediate layers. The minimum allowed value is 1.0.

The default *Prism Layer Thickness Ratio* value of *1.5* means that the inner most layer is 1.5 times the thickness of its outer most layer. Valid values of thickness ratio are given in the table below.

Stretching Function	Valid Values of Thickness Ratio
Geometric Progression	>= 1.0
Hyperbolic Tangent	> 1.0

In addition to the global control, local control is available.

When using Parts Based Meshing:

1. Set the **Custom Controls** > **[Surface]** > **Controls** > **Prism Layers** property to **Specify Custom Values**.
2. In the **Prism Layers** > **Customize** node, activate **Customize Distribution**.
3. Under the **Values** > **Custom Prism Values** node, a local value of the **Prism Layer Thickness Ratio** can be set on the chosen part surface.

When using Region Based Meshing:

4. Set the **[Boundary]** > **Mesh Conditions** > **Customize Prism Mesh** property to **Specify Custom Values**.
5. Under the **Mesh Values** node, a local value of the **Prism Layer Thickness Ratio** can be set on the chosen boundary.

What is the Gap Fill Percentage?

The Gap Fill Percentage property of the prism layer mesher controls the thickness of the subsurface (and hence the prism layer) where boundary surfaces are close to one another.

For example, two parallel planar boundaries that are close together. It essentially defines the amount of space in a gap that is occupied with a prism layer.

The thickness for each prism layer is calculated by multiplying the **Gap Fill Percentage** value (25% by default) by the intersection distance. This intersection distance is the normal distance between the two boundaries. If one of the boundaries does not have the prism layer option activated, then the maximum allowable prism layer thickness is 2 * Gap Fill Percentage * Intersection Distance.

The gap fill percentage value influences prism generation for surfaces that are perpendicular to one another, for example, a symmetry plane and a wall boundary. In this instance, the symmetry boundary is considered close to the wall leading to a reduction of the subsurface (and hence prism layer) thickness in the vicinity of the intersection of the boundaries. To minimize this effect, increase the gap fill percentage value.

What is the Minimum Thickness?

The Minimum Thickness is the smallest subsurface/prism layer thickness that is allowed below which the thickness is forced to zero.

It is specified as a percentage of the local prism layer thickness reference value.

In order to avoid self intersection or to meet other quality criteria, the subsurface sometimes has to be retracted for certain boundary surfaces (for example, because of proximity effects). If the retraction is severe enough to cause the thickness to fall below the *Minimum Thickness* value, the subsurface is locally set to zero. This approach avoids producing poor quality cells through squeezing several prism layers into a thin section.

What is Layer Reduction?

The Layer Reduction Percentage property controls the point at which the number of layers within a contracting prism layer is reduced.

When generating a prism layer mesh, it can become necessary for the mesher to reduce the overall prism layer thickness due to corners, narrowing gaps, curved surfaces, or other local geometry. In these cases, it is preferable to reduce the number of layers locally as the thickness decreases, maintaining a lower aspect ratio (and hence better quality) for the remaining prism layer cells. The Layer Reduction Percentage property controls the point at which the number of layers within a contracting prism layer is reduced.

By default, the *Layer Reduction Percentage* is specified as a percentage of the global prism layer thickness value. Specifying a value between 0% and 100% activates the feature and the layer reduction thickness is calculated as:

$$\text{Layer Reduction Thickness} = \frac{\text{Layer Reduction Percentage}}{100} \times \text{Prism Layer Thickness}$$

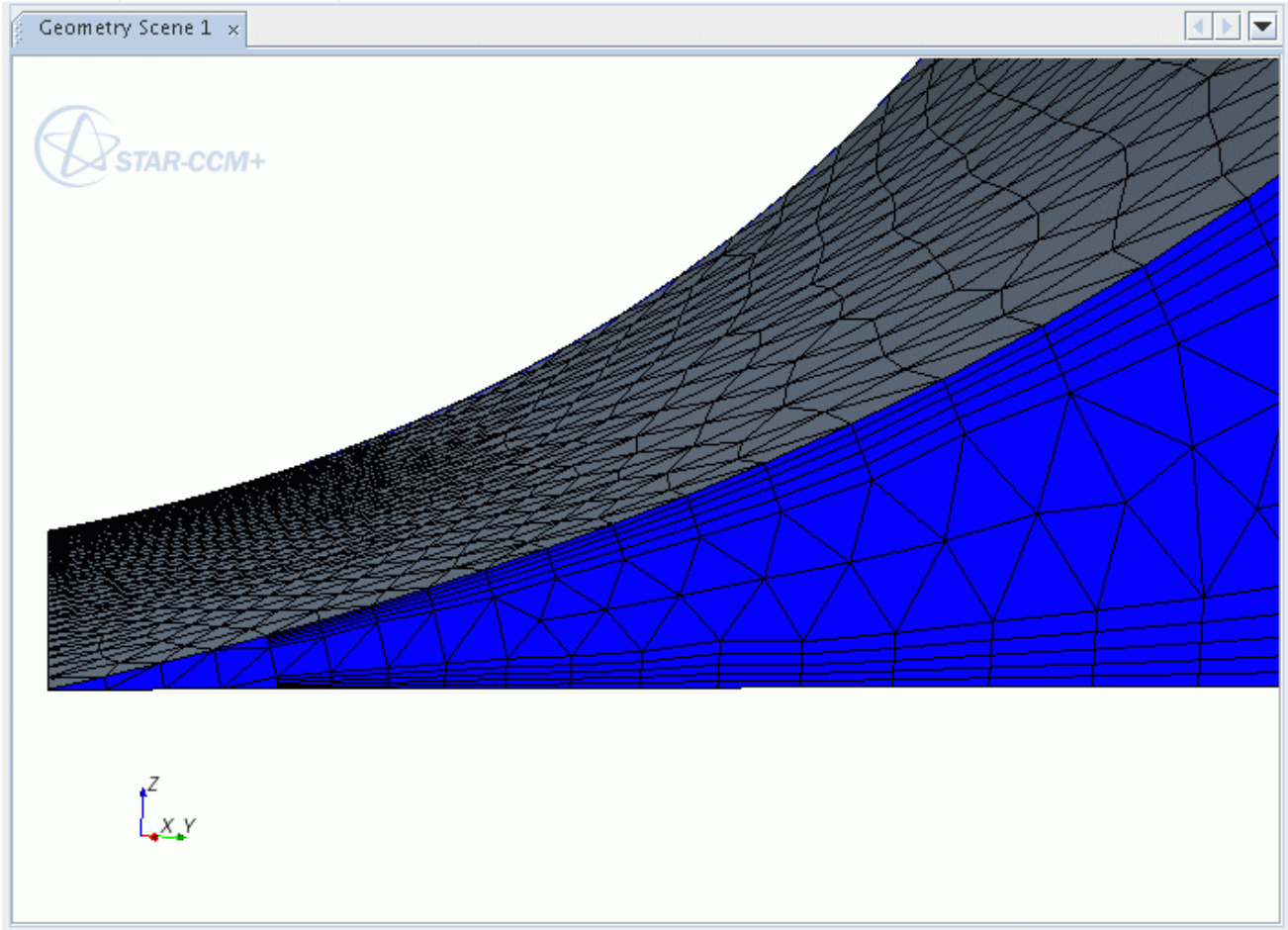
If a prism layer thickness has been specified on a part surface, region, or boundary, or through a volumetric control, then this local value is used instead of the global value.

Wherever the prism layer thickness falls below the Layer Reduction Thickness, prism layers are naturally thinned out as the prism layer thickness reduces. The default *Layer Reduction Percentage* value is 50%, which means that when the prism layer is reduced to half its target thickness, layers begin to be removed. Setting *Layer Reduction Percentage* to 0% deactivates the feature.

The range of influence of the layer reduction scheme on the prism layer mesh is illustrated below:

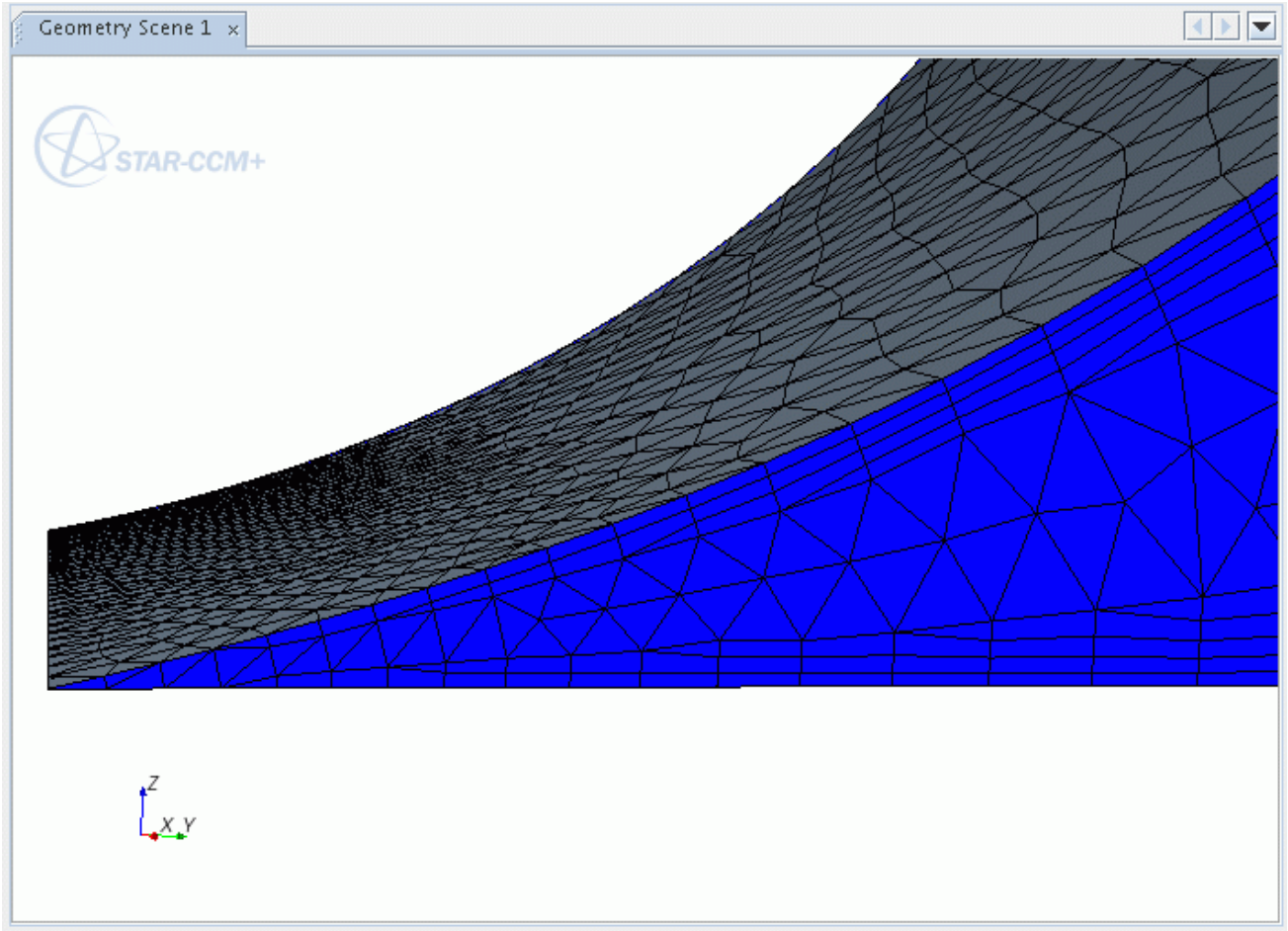
In the diagram that is shown above, the prism layers corresponding to the blue shaded areas would be subject to prism layer reduction based on the layer reduction thickness value as drawn. As this value increases, more of the prism layer is subject to the reduction process.

A comparison of a prism layer mesh without layer reduction and with layer reduction is shown below:



In the above example, no layer reduction (that is, layer reduction percentage = 0) has been applied. As the subsurface becomes thinner towards the angled corner, the prism layers progressively get thinner and eventually truncate where the subsurface retracts to zero height (determined by the minimum thickness value).

With a layer reduction value applied, the above situation can be avoided, as shown below:



In this case, the prism layers start truncating as you move towards the angled corner, reducing the number of layers that remain and increasing the overall quality.

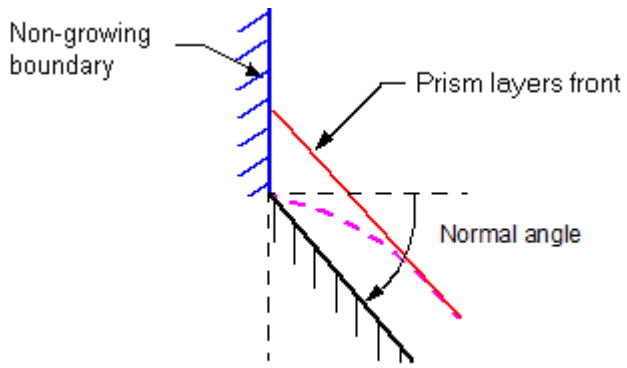
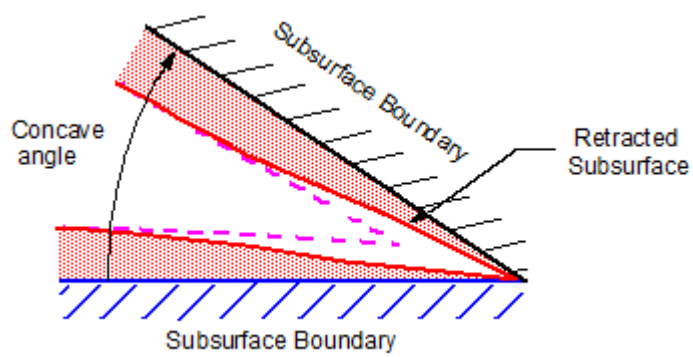
Notes:

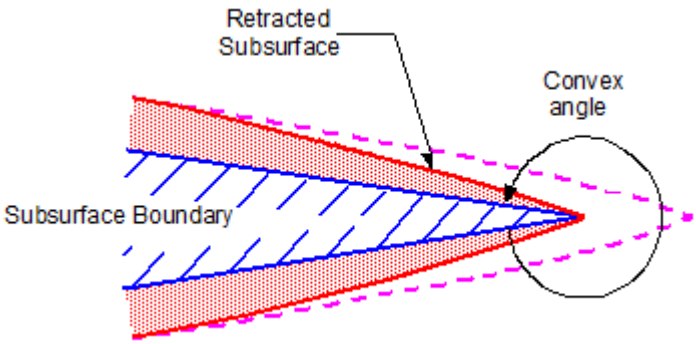
- The layer reduction feature has no effect where there is only one prism layer.
- If a layer reduction percentage value of 100% is used, then the entire prism layer mesh is subject to layer reduction. In general, values of 25 - 75% are reasonable to use for most cases.
- If the option is used with a volumetric control, then the prism layers that are associated with the control shape are also subject to layer reduction in the same way as described above.

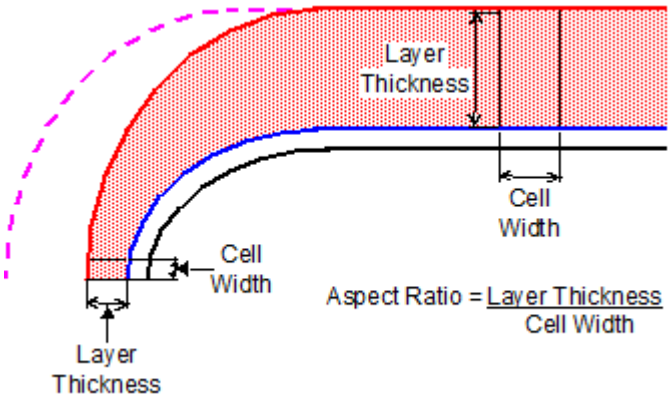
Prism Layer Model Expert Properties

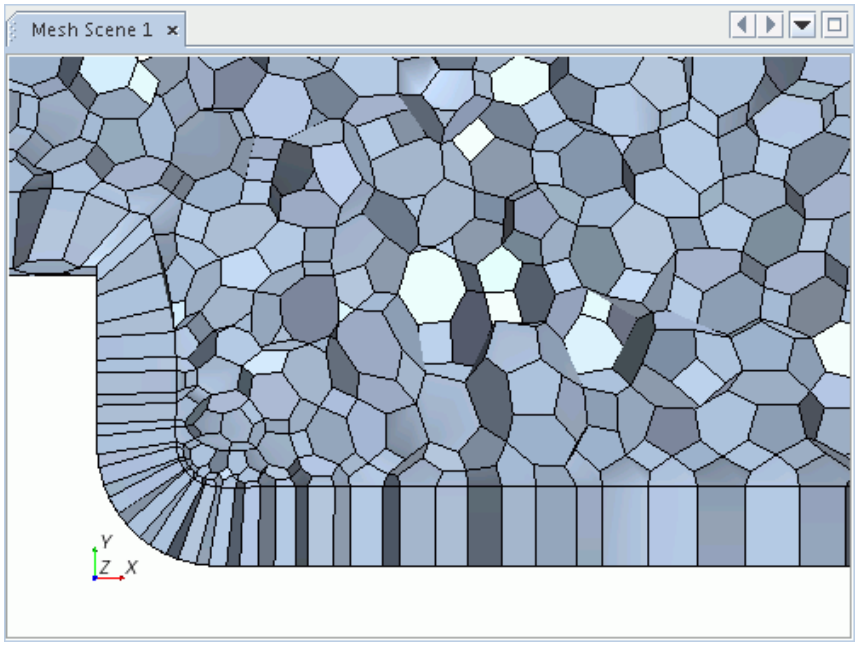
This section describes the expert properties of the Prism Layer model.

Boundary March Angle	The boundary march angle is used to control the behavior when the side of a prism layer lies on a boundary that does not have its own prism layer. The **Boundary March Angle** controls the way the prism layers march along (projected on) a boundary that does not have a prism layer (also known as a non-growing boundary). The angle (in degrees) specifies

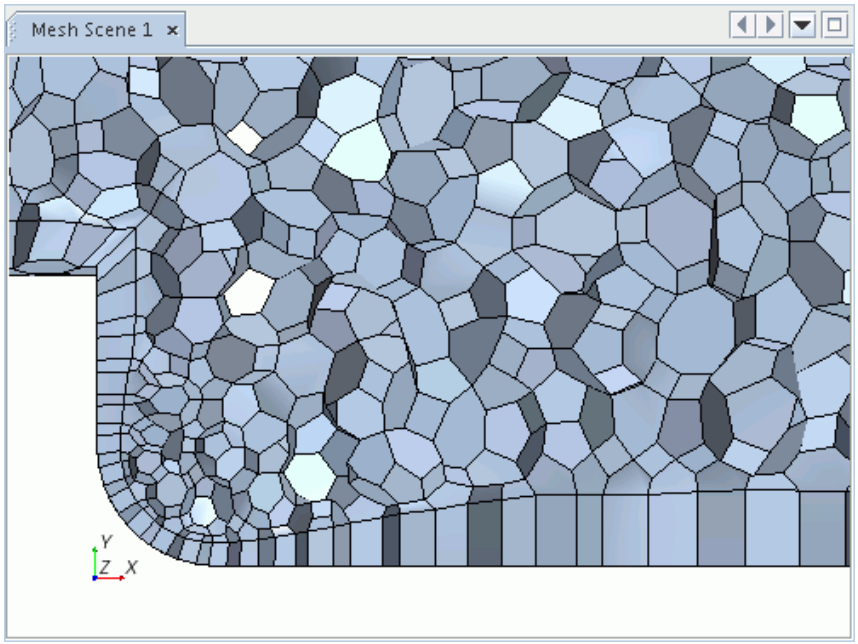
	the deviation from the normal that is allowed in order for the prism layers to be successfully projected on the non-growing boundary.  In the example that is shown, prism layers are “marched” along (projected on) the non-growing boundary (shown by the blue line) only if the *Boundary March Angle* is greater than the normal angle shown. If the normal angle is greater than the *Boundary March Angle*, then the prism layers are retracted, as shown by the magenta dotted line.
Concave Angle Limit	Sets a limiting angle that causes the prism layer to be retracted as it approaches the apex inside a wedge. The concave and convex angle limits result in the local retraction of the subsurface where the surface edge angle is less than the concave angle limit, or greater than the convex angle limit. The purpose of which is to improve the resultant prism layer mesh quality in these areas. For example, in the case of a car wheel meeting the ground, the concave angle limit is used to retract the subsurface where the wheel and the ground meet. Doing so avoids having non-orthogonal and possibly high aspect ratio prism cells at the edge intersection of the two surfaces. The concave angle is illustrated below:  In the above diagram, the black and blue boundaries represent surfaces from which subsurfaces are generated. The dotted magenta line

	represents the subsurface that would have been produced without the concave angle limit being activated. With the concave angle specified to a suitable value (that is, more than the geometry interior angle), the red line represents the retracted subsurface, which results in a higher-quality prism layer mesh.
Convex Angle Limit	Sets a limiting angle that causes the prism layer to be retracted as it approaches the apex outside a wedge. The convex angle can be used in situations such as at the trailing edge of a wing to retract the subsurface back to give an external angle is greater than the specified limit. An example of this situation is illustrated below:  In the above diagram, the blue boundary represents the boundary surface from which a subsurface is generated. The dotted magenta line represents the subsurface that would have been produced without the convex angle limit being activated. With the convex angle specified to a suitable angle (less than the exterior angle of the geometry), the red line represents the retracted subsurface, which results in a higher-quality prism layer mesh.
Near Core Layer Aspect Ratio	Sets an aspect ratio between the layer thickness and cell width of the prism layer closest to the core mesh, which improves the transition from the prism layer to the core mesh. The **Near Core Layer Aspect Ratio** controls the ratio between the cell height (layer thickness) and cell width of the prism cell layer closest to the core mesh. The cell faces on the core mesh (known as the subsurface) govern the prism layer cell width, and so the layer height is adjusted to set the aspect ratio. When the geometry surface is curved, the Surface Curvature control refines the mesh, which results in smaller cell widths across the curved portion. The near core layer aspect ratio becomes effective on these curved portions, as it prevents the mesher from generating long, thin cells with a high aspect ratio. This property only affects the thickness of the cell layer that is next to the core mesh. The thickness of the remaining layers is defined

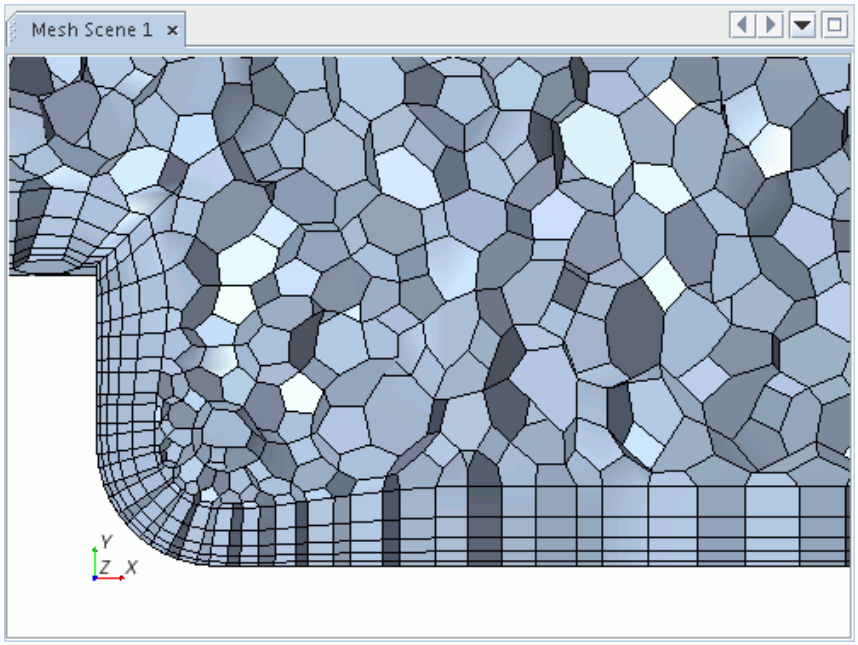
	according to the stretching mode and function, and with any other active constraints. The near core layer aspect ratio is illustrated in the diagram below. The red area represents the prism layer closest to the core mesh (that is, the core mesh begins after the red line). The dashed magenta line shows the profile of the prism layer if no aspect ratio is applied. 
	As shown above, the aspect ratio is defined as the layer thickness divided by the cell width. For example, setting a value of 2 gives a layer thickness twice as large as the cell width. To deactivate the property, set a value of 0. Using an aspect ratio improves the transition from the prism cell layers to the core mesh, as the prism layer cells are a similar size to the core mesh cells. This option also prevents “tall” cells being generated, as the aspect ratio of prism layer cells is explicitly specified. Setting the near core layer aspect ratio influences the overall prism layer thickness, particularly on curved surfaces. However, note that the overall prism layer thickness always remains at or below the user-defined value. As an example, the following image shows a thick boundary layer without the near core layer aspect ratio specified. There are high aspect ratio cells in the prism layer due to the curvature of the geometry and associated increased surface mesh density.



If the near core layer aspect ratio is set to 1, the transition from the prism layer to the core volume mesh is greatly improved. The volume of the prism layer cells is then closer to the volume of the adjacent core mesh cells.



The overall prism layer thickness is reduced in this example. A more uniform thickness would be obtained by increasing the number cell layers, as shown below. You could also increase the specified aspect ratio.

	
Generate Standard Cells Only	This option can be used with the tetrahedral meshing model to produce a volume mesh containing only pure tetrahedral and prismatic cells with triangular end faces. In order to produce a volume mesh containing pure tetrahedral and prismatic cells only, the generate standard cells only option can be used with the tetrahedral meshing model. Ensure that the number of prism layers that are used in the mesh is uniform throughout. That is, no custom number of prism layers have been set for part surfaces, regions, or boundaries. If these conditions are met, then the volume mesh that is produced can be exported to third-party packages that can only accept standard cell shapes. The default state for this option is off. Note: When using Parts Based Meshing, this option is only available to set when using a tetrahedral volume mesher.
Improve Subsurface Quality	This option causes retriangulation of boundary surfaces that do not have prism layers after the subsurface stage of the volume meshing process, and results in a better quality mesh. The improve subsurface quality option, which is on by default, improves the quality of boundaries that do not have prism layers (such as inlets, outlets, and interfaces). This option results in the retriangulation of such surfaces after the subsurface stage of the volume meshing process. With the option deactivated, such boundaries are smoothed as part of the subsurface process. This smoothing can lead to poor quality triangulation, especially when the prism layer thickness is large compared with the size of the boundary mesh triangles. Activating the option leads to better quality triangulation, which results in a higher-quality volume mesh being created.

	Note: This option is hardwired as on and cannot be turned off when using Parts Based Meshing. In this case, the option is not shown.

Prism Layer Mesh Bibliography

This section lists bibliographic references.

[11] *Vinokur, M. 1980. "On One-Dimensional Stretching Functions for Finite-Difference Calculations", NASA Contractor Report 3313.*

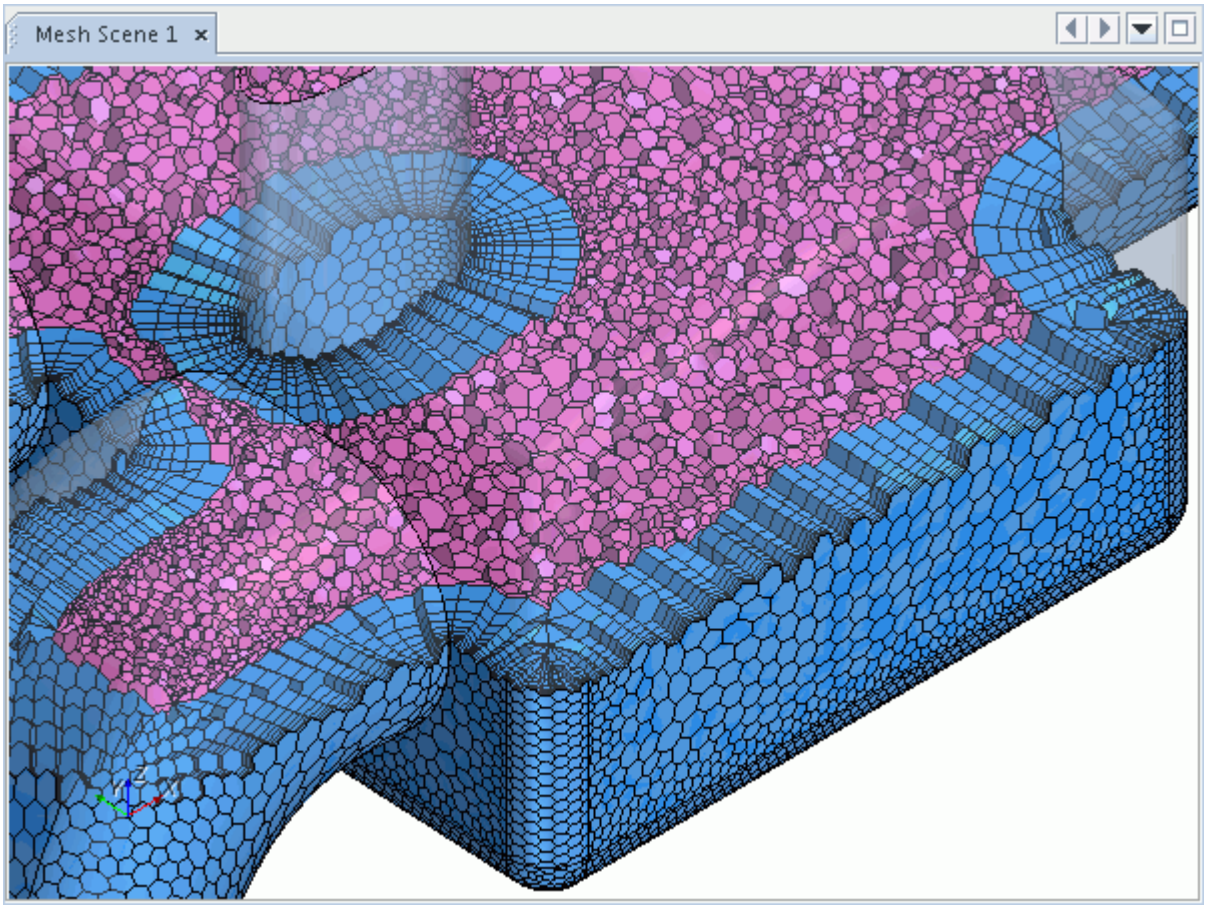
Advancing Layer Mesher

The advancing layer mesher generates layers of prismatic cells around the surfaces of regions or parts and fills the remaining void with polyhedral cells. This mesher either uses the input triangular surface mesh or converts the triangular mesh to a polygonal or quad surface mesh – depending on the *Surface Element Type* that is selected. This surface mesh is then advanced into the region volume to form cell layers. Some advantages of this approach are a conformal match on either side of an interface, and the ability to generate thicker and more uniform cell layers.

The extent and characteristics of the near-wall cell layers are user-defined. In this respect, the advancing layer mesher is a stand-alone volume mesher – it is not the same as the prism layer model. When using a Parts-Based mesh operation, before the Advancing Layer Mesher becomes available, you select either the Polyhedral or Tetrahedral volume mesher – which is used for the core mesh. When using a Regions-Based mesh continuum, no other volume meshers are compulsory with the Advancing Layer Mesher – the core is filled with polyhedral cells by default.

The prismatic cell layers help capture the boundary layer, turbulence effects, and heat transfer near wall boundaries. Prismatic cell layers can also be projected on non-extrusion boundaries such as inlets, outlets, and symmetry planes. The core mesh provides a balanced solution for meshing complex geometries and is relatively quick to generate. This approach makes the advancing layer mesher suitable for various applications.

An example of an advancing layer mesh is shown below. For illustration, the prismatic layers are colored blue and the core mesh is colored pink.



Contents:

Advancing Layer Mesher Properties
Selecting the Advancing Layer Meshing Model
How Is an Advancing Layer Mesh Generated?
What Are the Differences Between the Advancing Layer and Prism Layer Meshers?

See Also:

Advancing Layer Mesher Properties
Prism Layer Mesher

Advancing Layer Mesher Properties

This section contains the properties of the Advancing Layer Mesher when using Parts-Based meshing (PBM) or Region-Based meshing (RBM).

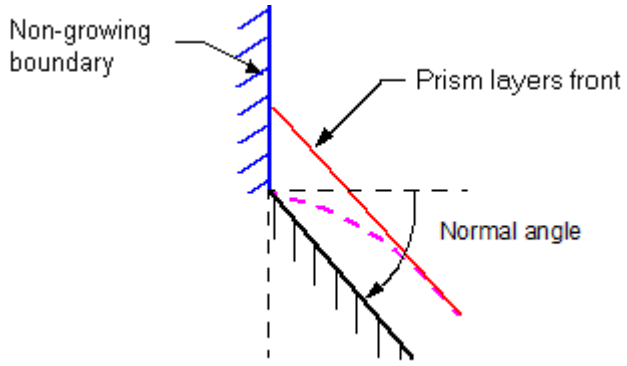
Stretching Function	Selects the stretching function to use in calculating the thickness of each cell layer within the prism layer.	
	Geometric Progression	The thickness of each cell layer is calculated using a geometric

		progression that is based around a constant size ratio.
	Hyperbolic Tangent	The thickness of each cell layer is calculated using a hyperbolic tangent stretching function.
Stretching Mode (RBM) / **Distribution Mode** (PBM)	Selects the stretching mode to use in specifying the relative size parameters that the stretching function requires.	
	Stretch Factor	The definition of the stretch factor depends on which stretching function is being used. Adds the **Prism Layer Stretching** node for global and custom settings.
	Wall Thickness	Sets the thickness of the near-wall cell layer. Thicknesses of the remaining layers are distributed according to a calculated stretch factor. Adds the **Thickness of Near Wall Prism Layer** node for global and custom settings.
	Thickness Ratio	Sets the ratio of the thickness of the cell layer furthest from the wall to the thickness of the cell layer nearest the wall. Adds the **Prism Layer Thickness Ratio** node for global and custom settings.
Run Optimizer (RBM only)	Performs additional optimization of vertices during the meshing process. When using PBM, you can specify this property in the properties of the core mesher.	
	Activated	Activate the optimizer during meshing
	Deactivated	Deactivate the optimizer during meshing

Advancing Layer Mesher Expert Properties

The following table contains expert properties that are specific to the advancing layer mesher model. Common volume mesher expert properties are also available.

Enable Patcher (RBM only)	Patch the surface to smooth cell faces on the surface, while maintaining geometry features. However, this property can generate high aspect ratio cells along sharp edges.	
	Activated	Activates the patcher during meshing

	Deactivated	Deactivates the patcher during meshing
Surface Element Type	Specifies the shape of the faces on the surface.	
	Triangle	Creates the surface mesh using triangles.
	Quad	Creates the surface mesh using quadrilaterals.
	Polygon	Creates the surface mesh using polygons.
Generate Standard Cells Only	Lets you choose if only standard cells are produced or not. In RBM, this property is only valid when *Surface Element Type* is set to **Triangle** or **Quad**. In PBM, this property is activated automatically when the Tetrahedral mesher is selected as the core mesher.	
	Activated	The mesh that is generated can only contain cells that are tetrahedral, hexahedral, wedges, and pyramids.
	Deactivated	The mesh that is generated can contain polyhedral cells.
Boundary March Angle (RBM only)	The boundary march angle is used to control the behavior when the side of a prism layer lies on a boundary that does not have its own prism layer. The **Boundary March Angle** controls the way the prism layers march along (projected on) a boundary that does not have a prism layer (also known as a non-growing boundary). The angle (in degrees) specifies the deviation from the normal that is allowed in order for the prism layers to be successfully projected on the non-growing boundary.  In the example that is shown, prism layers are “marched” along (projected on) the non-growing boundary (shown by the blue line) only if the *Boundary March Angle* is greater than the normal angle shown. If	

	the normal angle is greater than the *Boundary March Angle*, then the prism layers are retracted, as shown by the magenta dotted line.
Choose a Mesh Size Table (RBM only)	Allows you to refine the mesh using field functions based on an existing solution. When using PBM, you can specify this property in the properties of the core mesher.
Sharp Edge Refinement (PBM only)	When activated, this option automatically improves the mesh near sharp edges. Typically, sharp edges such as trailing edges on a wing are problematic for mesh generators as they rely on surface normal vectors to guide them in creating cells next to the surface. Sharp Edge Refinement adds additional surface normal vectors around the sharp edge. Sharp edge refinement is applied to edges where the internal angle between faces is less that than 150 degrees, including baffle edges. Some edges are excluded from the refinement due to topological features—for example, single sharp edges or junctions of more than three sharp edges.

See Also:

Advancing Layer Mesher
Working with the Stretching Function
Stretch Factor

Selecting the Advancing Layer Meshing Model

You can use the Advancing Layer Mesher when using a Parts-Based or Region-Based meshing approach. See Advantages of Parts-Based Meshing.

To select the advancing layer meshing model using a parts-based mesh operation:

1. Right-click the **Geometry** > **Operations** node and select **New** > **Automated Mesh**.
2. In the *Create Automated Mesh Operation* dialog:
   a. Select the *Parts* to mesh.
   b. From the *Core Volume Meshers*, select **Polyhedral Mesher** or **Tetrahedral Mesher**.
   c. From the *Optional Boundary Layer Meshers*, select **Advancing Layer Mesher**.
   d. Click **OK**.

To select the advancing layer meshing model using a region-based mesh continuum:

1. Right-click the **Continua** node and select **New** > **Mesh Continuum**.
2. Right-click the **Mesh** node and select **Select Meshing Models...**
3. In the *Mesh Model Selection* dialog, from the *Volume Mesh* group box, select **Advancing Layer Mesher**.

How Is an Advancing Layer Mesh Generated?

The generation of an advancing layer mesh is outlined in the steps below:

1. Generate a surface mesh from the initial triangulated surface and specify the surface type element shape.
2. Extrude the surface mesh to create prismatic cell layers.
3. Generate a polyhedral core mesh for the remaining volume.
4. Assemble the core and prismatic cell layers.
5. If required, you can view the prism layer cells as a cell set using a field function.

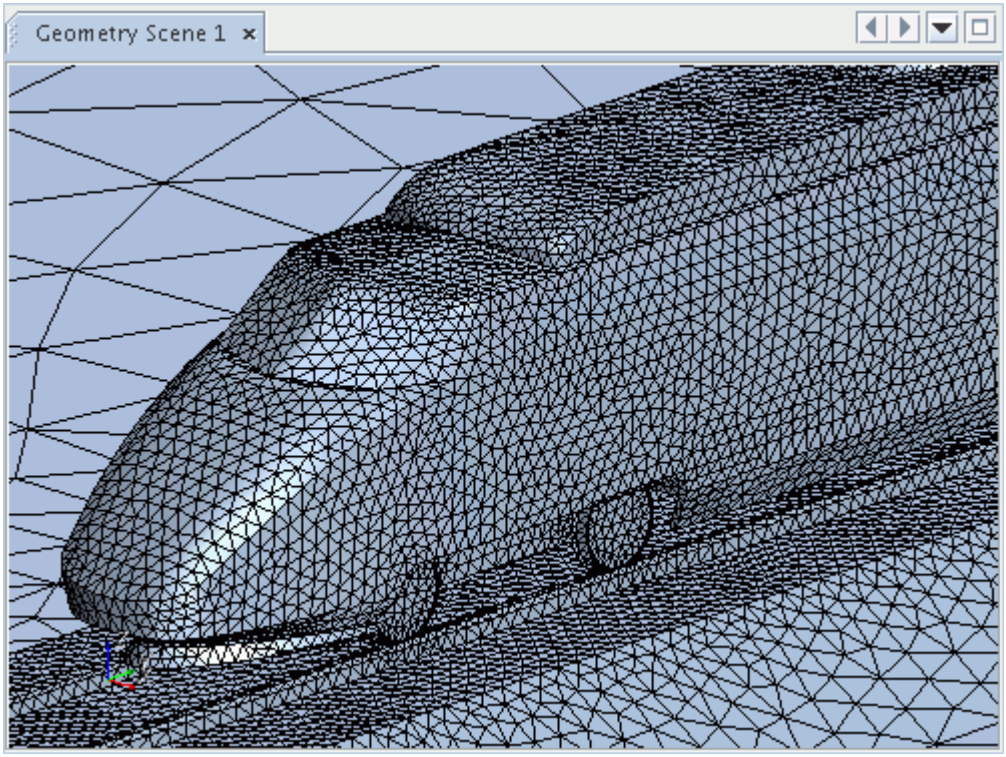
Contents:

Generating a Surface Mesh
Generating the Prismatic Cell Layers
Generating the Core Mesh
Assembling the Volume Mesh

Generating a Surface Mesh

In the first step, the advancing layer mesher generates a surface mesh from the triangulated surface (provided by the Surface Remesher or Surface Wrapper).

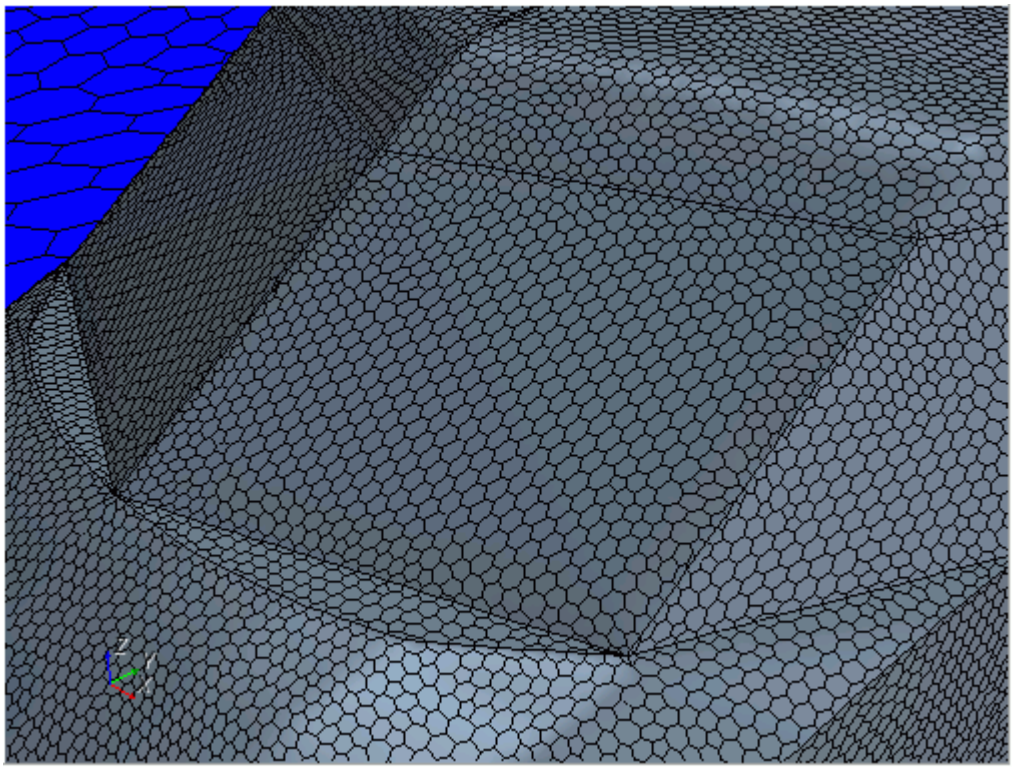
An example of a remeshed surface using triangles is shown below.



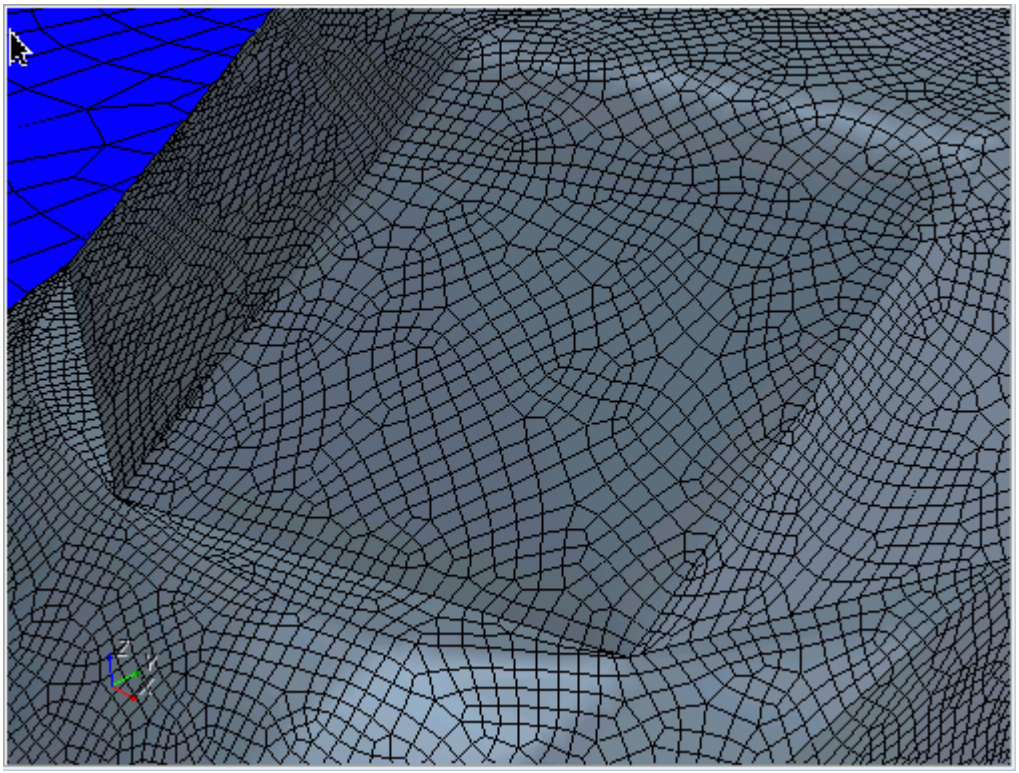
The advancing layer mesher generates the surface mesh either by using the existing triangles, or by converting the triangles to quadrilaterals or polygons, depending on the surface element type property that is specified.

Note: If the quadrilateral or polygonal surfaces that are created contain quality issues–such as intersections and face proximity, the advancing layer mesher uses the original triangulated surface mesh on the affected patches. This behavior can cause some boundaries to remain triangular when the surface element type property is set as **quadrilaterals** or **polygons**. In this situation, the following message appears in the *Output* window: `Warning: initial triangular surface has quality issues. Converted quad/polygon surface will contain triangles.`

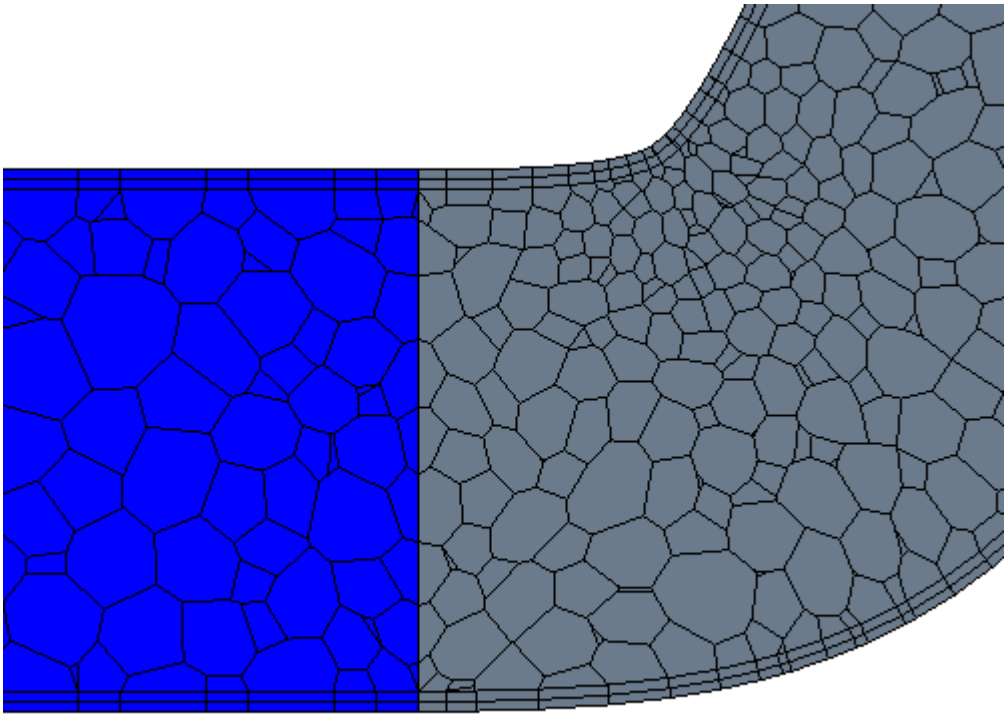
An example of a polygon surface mesh is shown below.



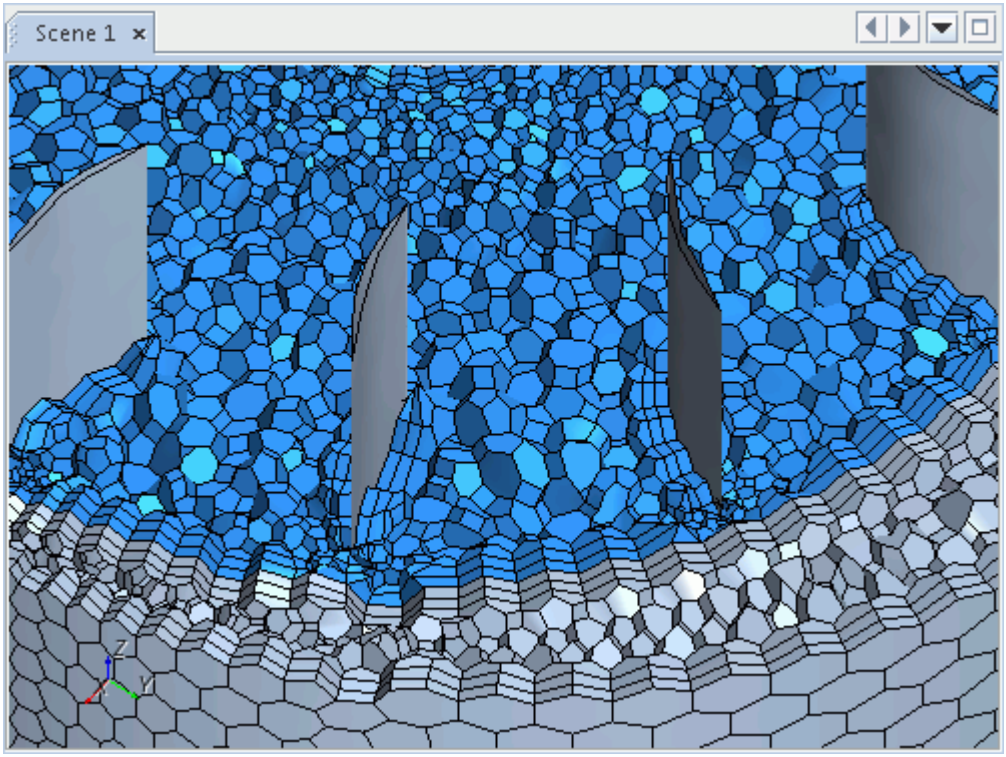
An example of a quadrilateral surface mesh is shown below.



The advancing layer mesher can create a conformal match between cells on either side of a direct or periodic interface. When layers grow from both sides of an interface, a conformal mesh is created. However, to guarantee a conformal mesh when layers are projected on to the interface, layer settings on surfaces that meet at the interface perimeter must be identical. You can choose to deactivate layer growth on the interface itself, as shown in the following image.

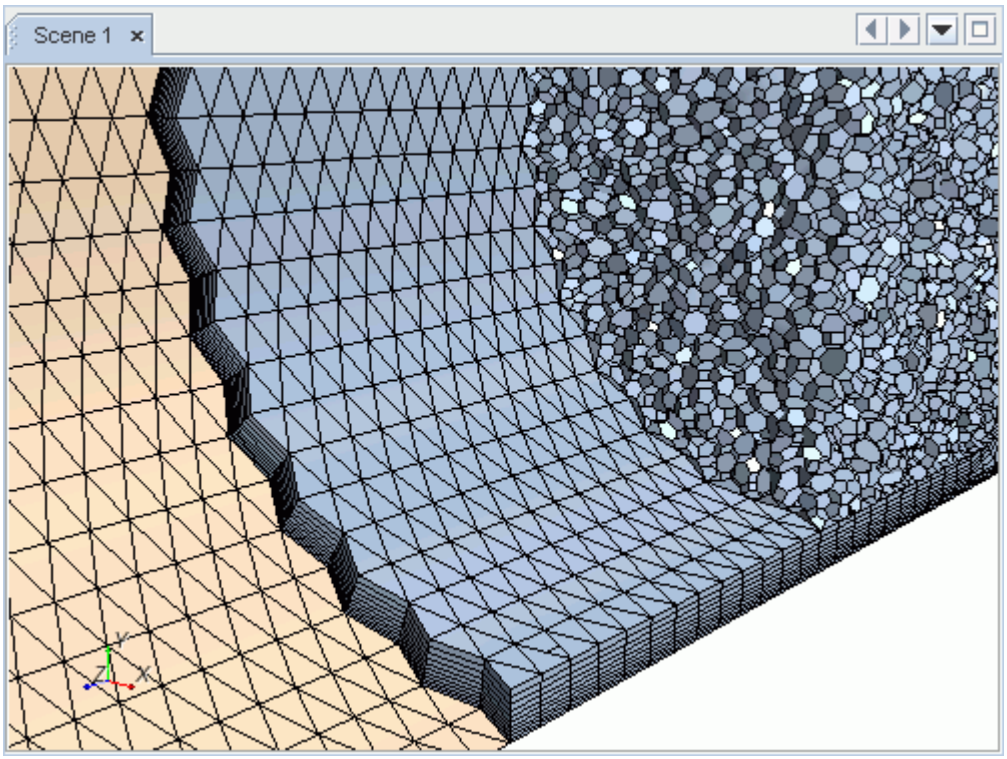


Note: To guarantee conformality ensure that the layer settings (thickness, number of layers, and stretching) are all the same.



Maintaining the Existing Surface Mesh

If you have an existing triangular surface mesh from the Surface Remesher, Surface Wrapper, or third-party software, you can choose to grow the prismatic cell layers from the existing triangular surface mesh. This function can be useful, for example, when mapping to third-party codes that cannot interpolate data from polygonal or tetrahedral cells. An example of prismatic cell layers that are generated from an existing triangular surface mesh is shown below.



To maintain an existing triangular surface mesh, set the Advancing Layer Mesher property, *Surface Element Type* to *Triangle*.

See Also:

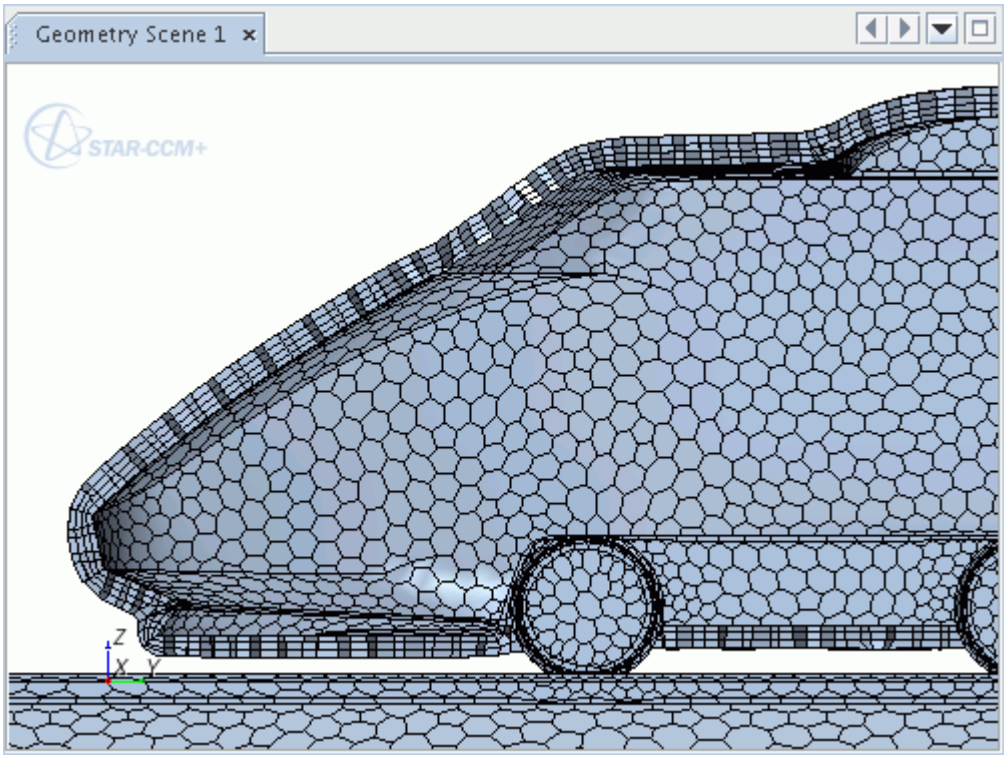
Surface Remesher

Surface Wrapper

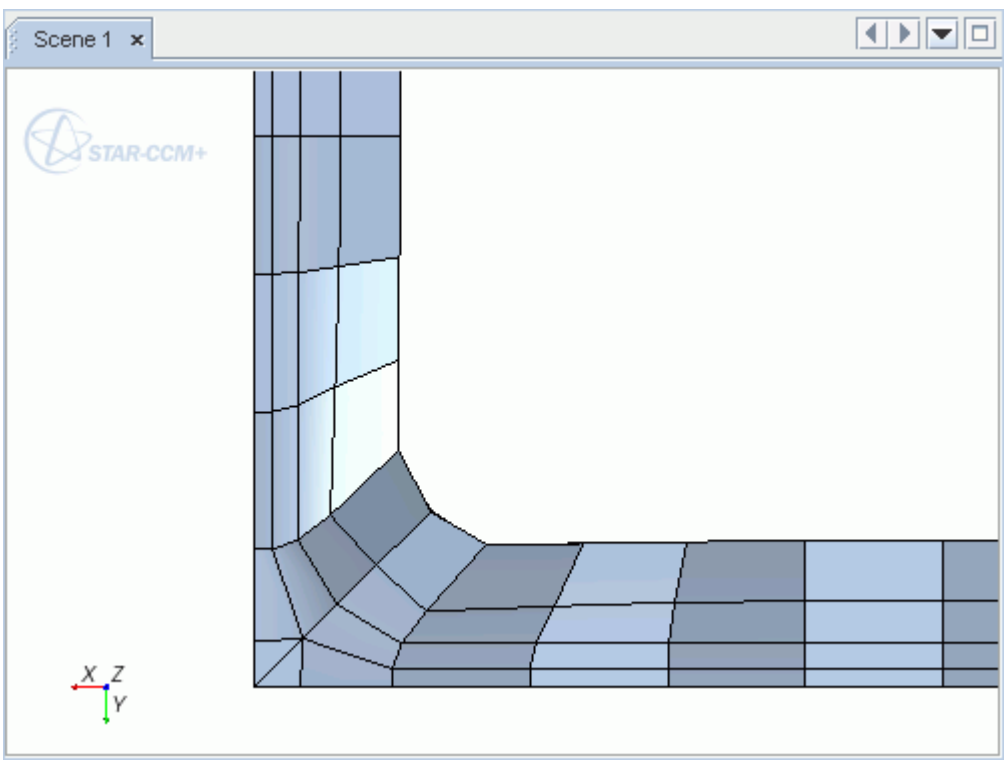
Generating the Prismatic Cell Layers

Prismatic cell layers are generated from the polygonal, quad, or triangular surface mesh. This approach gives an advantage over the prism layer mesher as the prismatic cell layers are less dependent on the original surface topology.

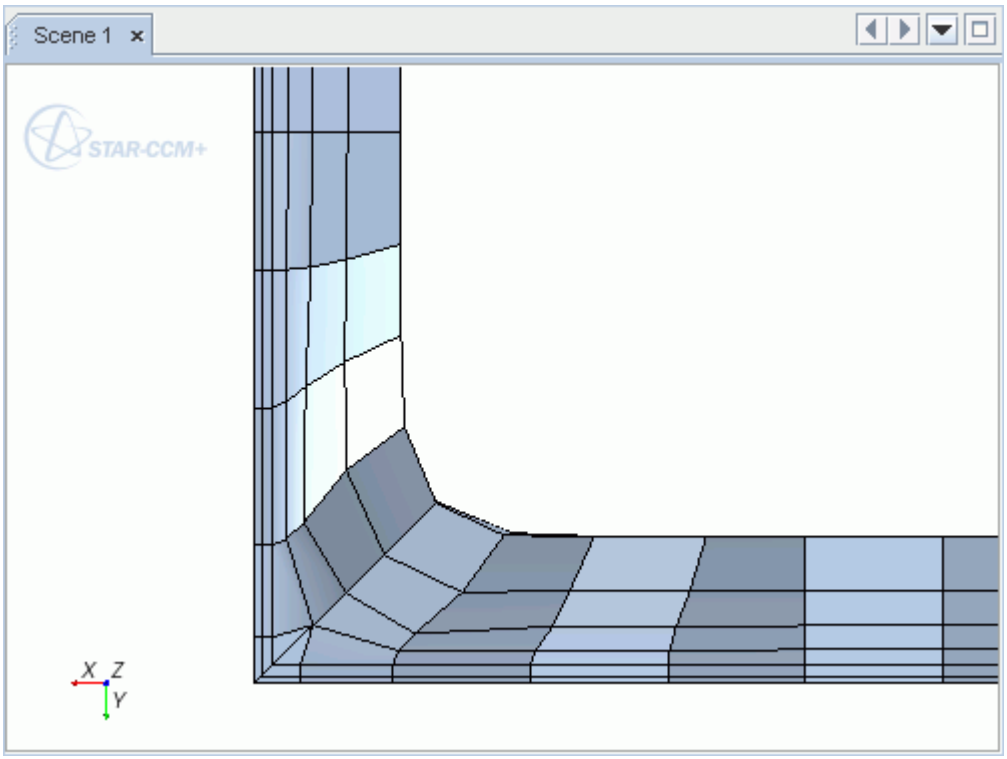
A more uniform thickness is therefore possible across all prismatic cell layers.



In some cases, the first layer next to the boundary may not have a uniform thickness. This scenario occurs when the advancing layer mesher has to collapse faces and edges in order to fit the required number of layers. In the example that is shown below, four prism layers is not sufficient for the mesher to maintain a uniform thickness in the corner of the volume mesh.



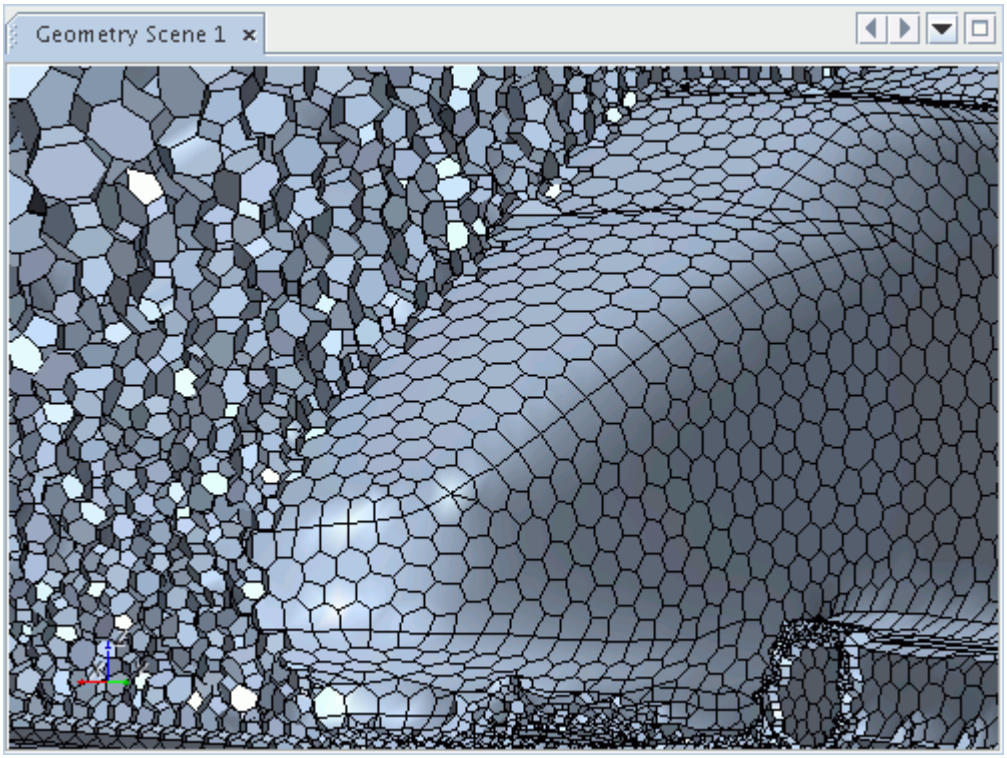
Increasing the number of prism layers to six, so that the first prism layer is thinner, results in a uniform first layer thickness.



Generating the Core Mesh

Once the prismatic cell layers have been generated, the advancing layer mesher proceeds to fill the remaining region volume with polyhedral or tetrahedral cells.

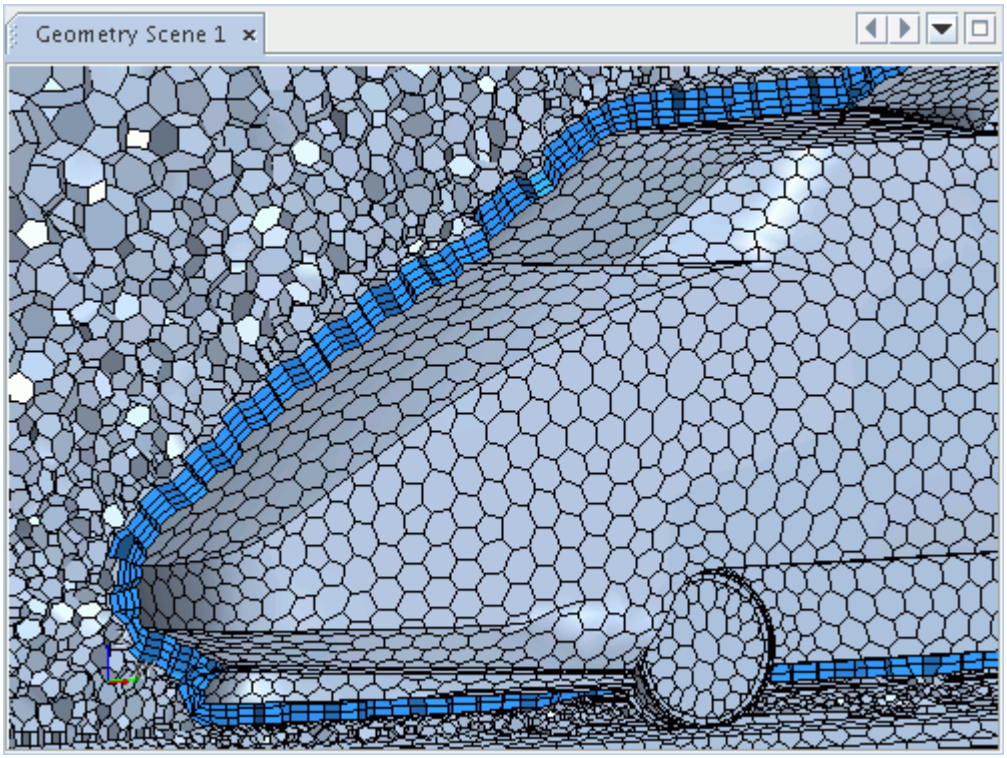
The advancing layer mesher uses the last prismatic cell layer as a basis for the polyhedral or tetrahedral core mesh. This approach ensures a one-to-one correlation between adjacent cell faces in the core mesh and cell layers. When a tetrahedral core mesh is used, the Advancing Layer Mesher guarantees that the mesh only contains standard cells such as tet, hex, wedge, or pyramid cells. These standard cells are suitable to use for Finite Element (FE) Analysis simulations which use the FE Solid Stress model.



Assembling the Volume Mesh

The final stage in generating an advancing layer mesh is to assemble the layered and core mesh.

The mesh is then improved and optimized using cell quality metrics.



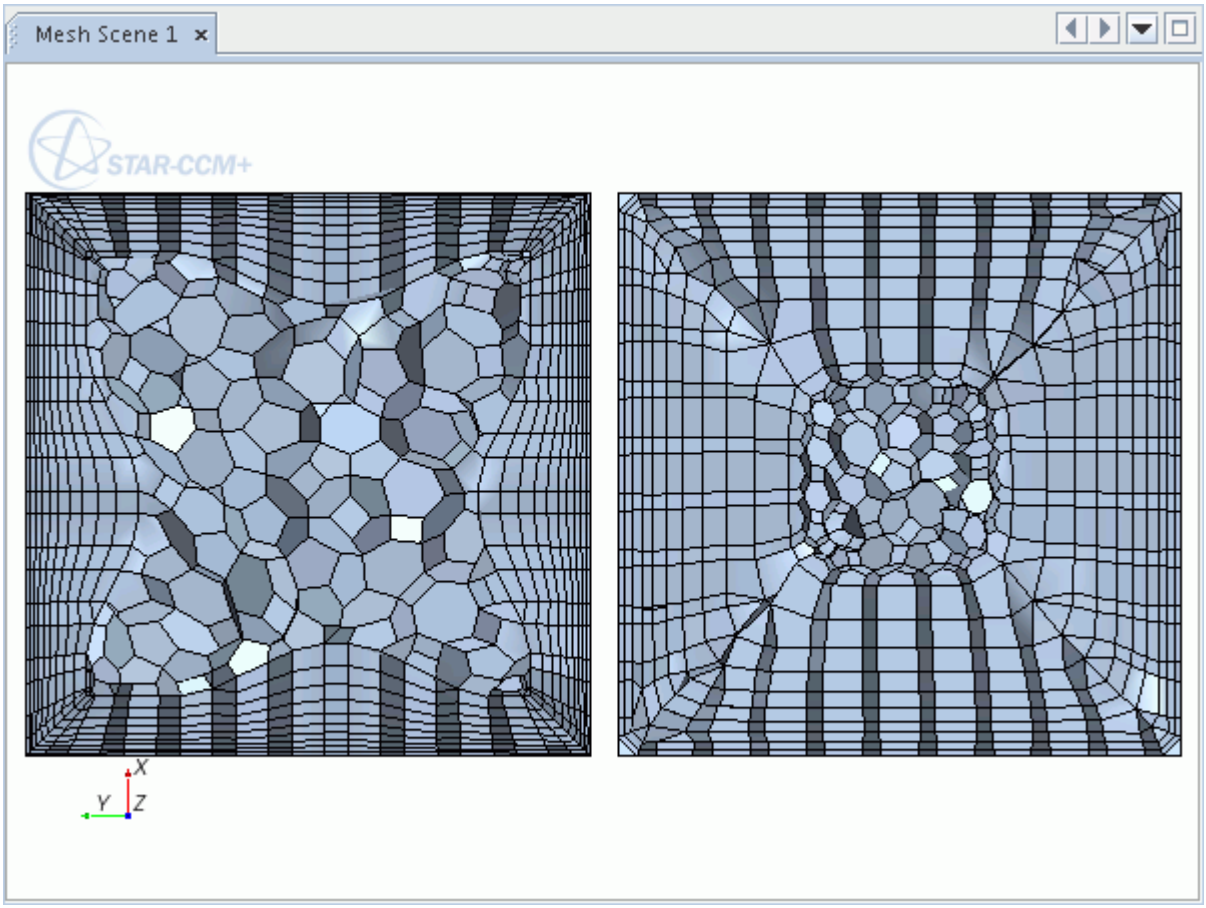
What Are the Differences Between the Advancing Layer and Prism Layer Meshers?

The advancing layer and prism layer meshers both produce prismatic cell layers near wall boundaries. However, there are several differences between the meshers that make them suitable for different applications.

The advancing layer mesher generates a polygonal surface mesh from the initial triangulated surface. The prismatic cell layers are then generated by extruding the polygonal surface into the region volume. This approach relaxes the dependency between the subsurface and region topology, and allows the advancing layer mesher to generate a thicker, more uniform layer than the prism layer mesher can achieve.

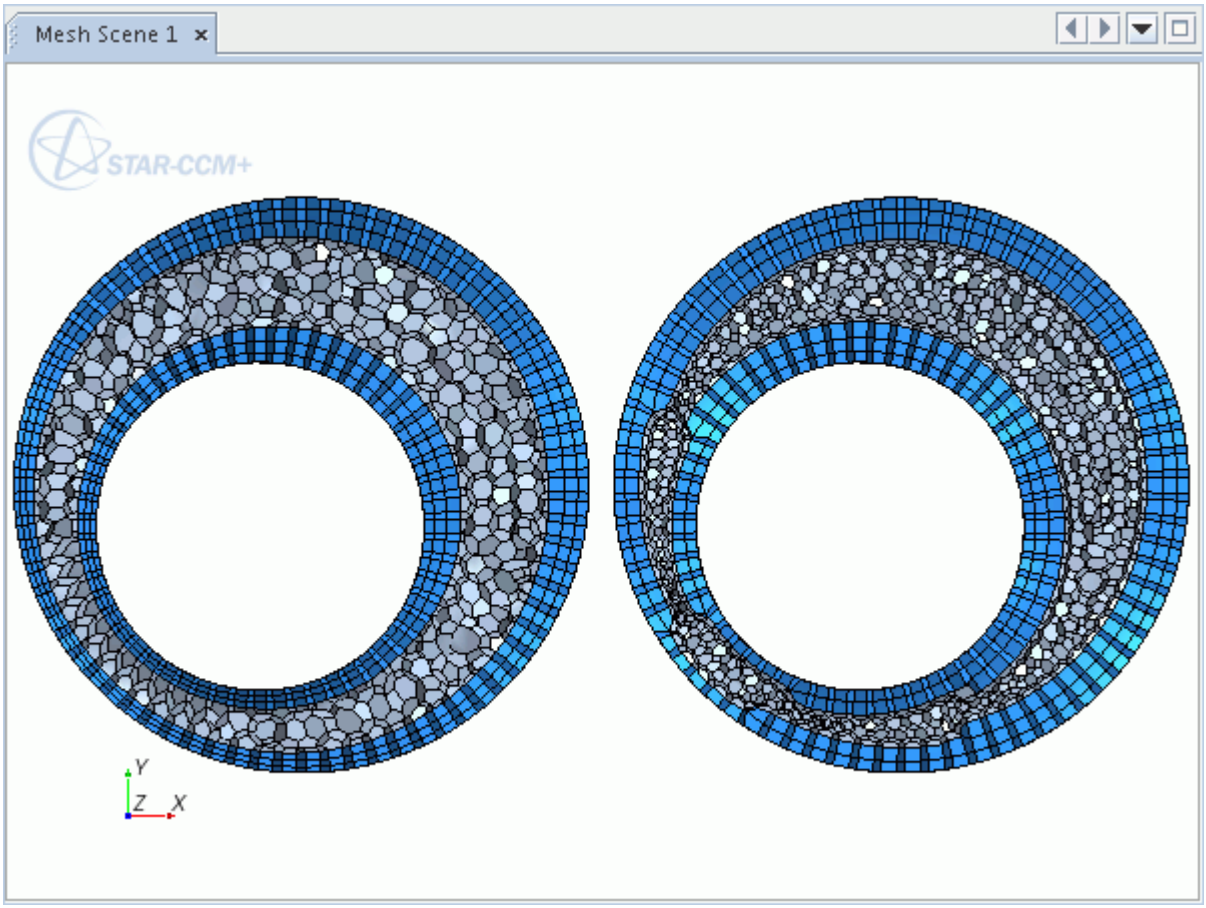
The advancing layer mesher also collapses and splits faces and edges, which helps to improve mesh quality – particularly when reducing cell skewness.

The example below illustrates the difference between a prism layer mesh (left) and advancing layer mesh (right).



Another difference between the prism layer and advancing layer meshers is the way that the two meshers behave in narrow gaps. In order to maintain some polyhedral cells between the opposing prism layers, the prism layer mesher can gradually reduce the layer thickness and remove layers. The advancing layer mesher instead removes cell layers without any reduction in thickness. An example where it could be advantageous to use the advancing layer mesher is when using a non-hybrid wall function turbulence model. The advancing layer mesher maintains the first layer thickness to give more control of the y+ value, whereas the prism layer mesher reduces the first layer thickness and the y+ is changed.

This difference between the two meshers is illustrated in the example below. The left-hand region has been meshed using the polyhedral and prism layer meshers, and the right-hand region has been meshed using the advancing layer mesher. The prism layer mesher maintains three prism layers in the gap by reducing the layer thickness. In contrast, the advancing layer mesher drops to one prism layer in the gap, but maintains the first layer thickness. For clarity, the prism layer is shown in blue.



If a volumetric control is activated which contains cells that are assigned to the Advancing Layer Mesher, both the polyhedral/tetrahedral core cell size and the prismatic layer cell size are refined. When using Region-Based Meshing (RBM), it is not possible to customize the number of layers, stretching factor, or layer thickness using a volumetric control for cells that are assigned to the Advancing Layer mesher. However, when using the Parts Based Meshing (PBM) approach, you can customize prism settings for cells that are assigned to the Advancing Layer mesher using volumetric controls. When using PBM, you can update the selected Optional Boundary Layer Mesher between the Prism Layer Mesher or the Advancing Layer Mesher without losing settings for related default controls settings.

If a volumetric control is activated which contains cells that are assigned to the Prism Layer Mesher, you can customize the number of layers, stretching factor, or layer thickness. To refine the mesh by cell size, activate a volumetric control for the core mesher. The cell size that you set in the volumetric control is applied to the core mesh only.

Extruder Mesher

The extruder mesher can be included with the core mesh when generating a volume mesh in order to produce orthogonal extruded cells for specific boundary surfaces. The extruder mesh is typically used for inlet and outlet boundaries to extend the volume mesh beyond the original dimensions of the starting surface, so that a more representative computational domain is obtained.

However, it can also be used for any boundary type and for combinations of adjacent boundaries, such as expanding the walls of a box in all outward directions.

The extruder mesher can affect some types of interfaces, such as periodic interfaces, if these lie on the extruded boundaries. To avoid any potential issues, it is recommended that you create these types of interfaces after the volume mesh has been generated.

Note: The extruder meshing model cannot be used on an imported volume mesh in order to extend the boundaries. In this instance, the original volume mesh surface can be extracted and a new mesh can be generated using the pipeline in STAR-CCM+, with the extruder model activated to extend the domain.

The extruder meshing model can be selected for any mesh continuum once a core volume mesh model has been selected.

Contents:

What is the Extruder Meshing Model?

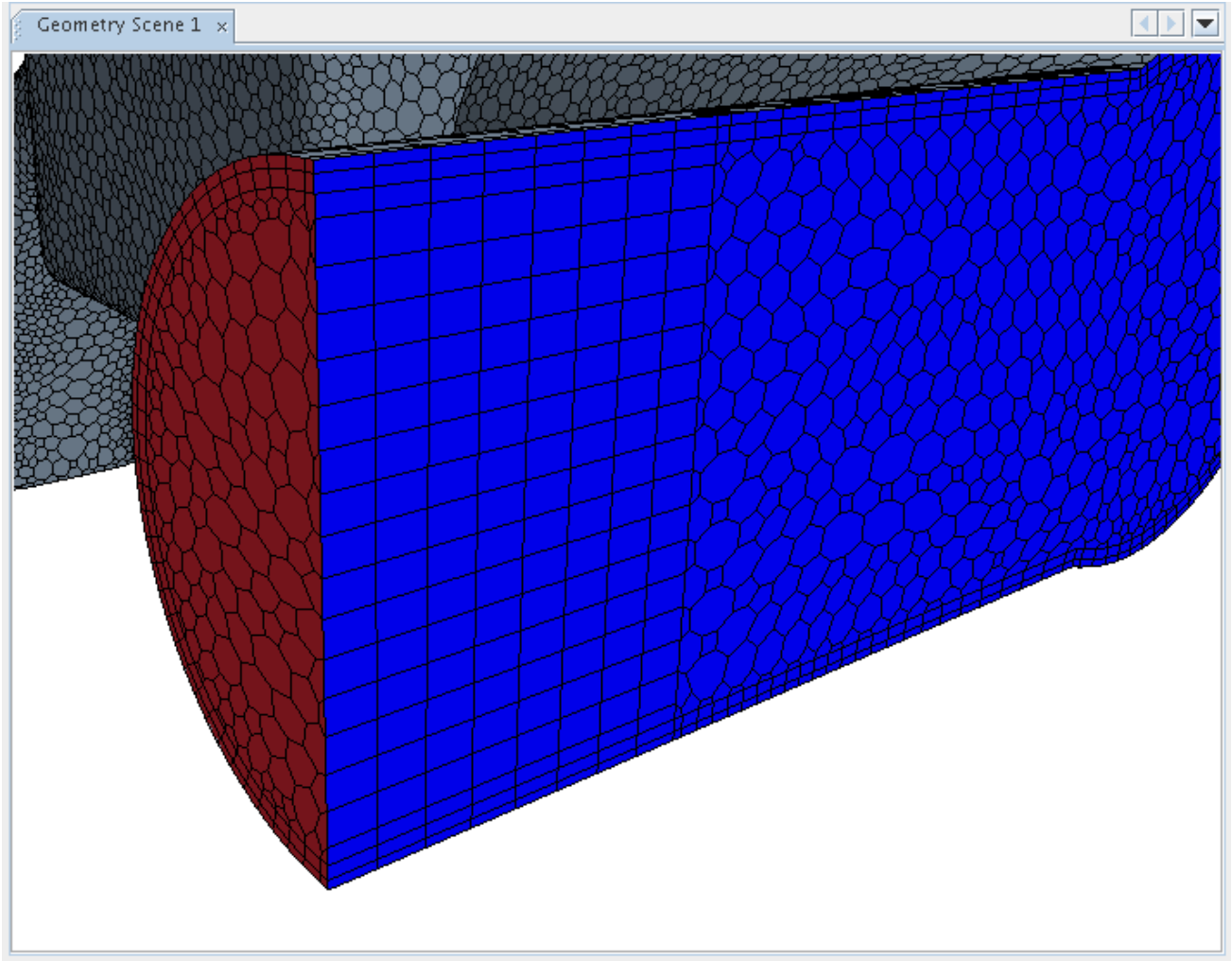
See Also:

Extracting a Surface Mesh
Extruding Surfaces from a Parts-Based Volume Mesh

What is the Extruder Meshing Model?

The extruder meshing model, when activated and enabled for a specific boundary, performs an additional volume meshing step once the core mesh and, optionally, the prism layer mesh, has been generated.

Based on supplied values for the number of layers, extrusion magnitude and stretching, the volume mesh is extruded from the specified boundary either in the normal or specified direction, extending the original domain. An example extrusion for an inlet boundary in the normal direction is shown below:



If the prism layer mesher is included in the volume meshing pipeline, then prism layers are also produced for the adjacent boundaries by virtue of the extrusion, as per the example above. By default the extruded mesh belongs to the same region as the originating boundary, but can be made into a new region if desired. The boundary types that are used for the extruded volume are set to match the types of the boundaries that they have been extruded from.

The extruder meshing model can be activated for any of the core mesh types and enabled for any boundary. Care must be taken however that the extrusion volume does not interfere with the existing mesh by intersecting it in any way. Although such intersection does not invalidate the computational domain, it is unphysical and it presents issues in post processing the analysis results. The properties of the extruder mesher can be changed to alter the behavior of extrusions on neighboring boundaries.

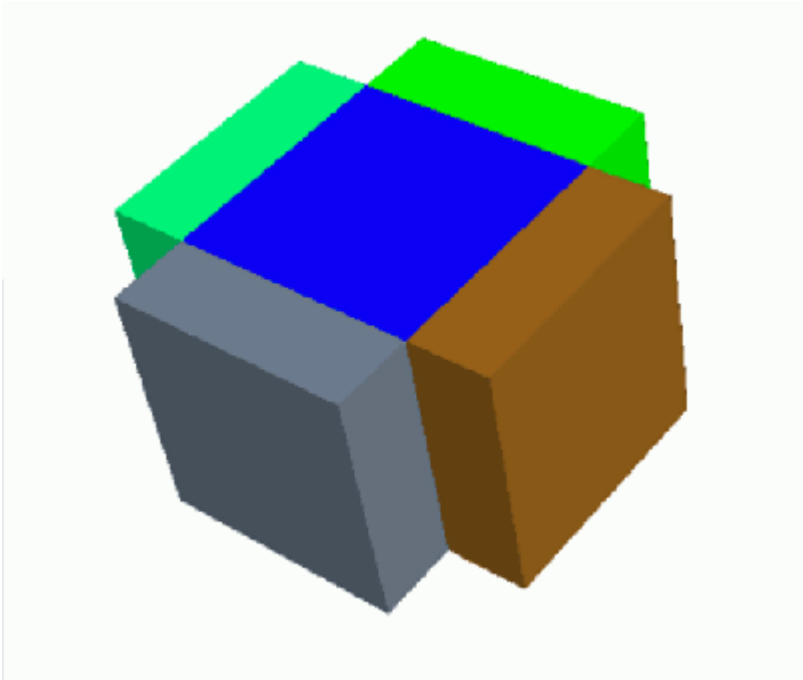
The extruder mesher differs from the prism layer mesher in that the latter cannot increase the domain extents beyond the initial surfaces. The prism mesher is primarily used to ensure suitable near wall mesh resolution for turbulence and heat transfer modeling.

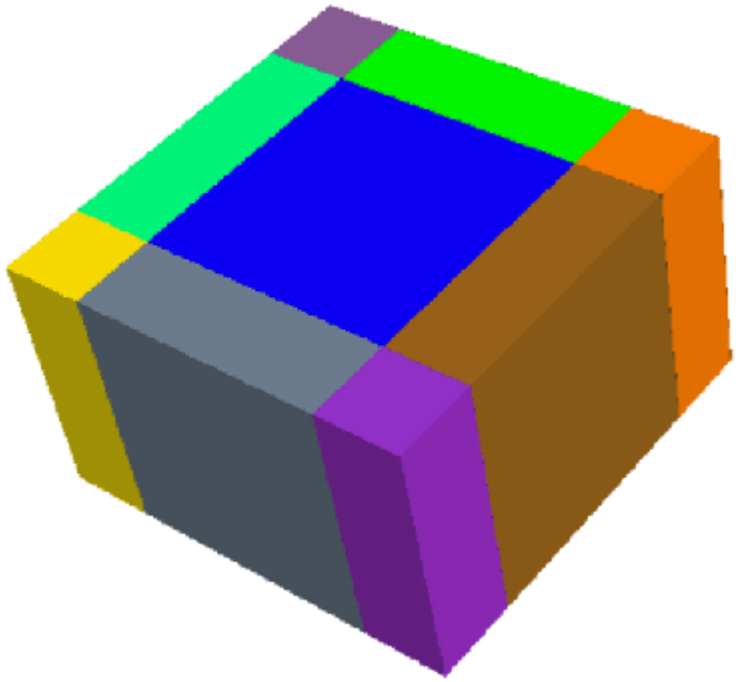
Contents:

Extruder Mesher Properties
Extrusion Type Properties

Extrusion Parameters Properties

Extruder Mesher Properties

Frozen Boundaries	The *Frozen Boundaries* option, which is on by default, determines the behavior of the extruder mesher when extruding a group of boundaries that do not lie on the same plane. In the image below, neighboring boundaries are extruded in their respective normal directions with the *Frozen Boundaries* option activated. The result is a change to the topology of the computational domain. 
	If this behavior is undesirable, deactivate the *Frozen Boundaries* option to allow the automatic creation of additional mesh blocks to retain the original topology. In this example case, the result would be a mesh topology as shown below:

	
	The *Frozen Boundaries* option is turned on by default as it produces the fastest extrusion of all selected boundaries, since a check for new boundaries is not required after each extrusion. Turning off the option can significantly slow down the extrusion process in some cases.
Check Validity	The check validity option, which is on by default, checks the validity of the resultant extruded mesh and takes corrective action if problems are found. It is recommended that this option is left on when using the extruder model.

See Also:

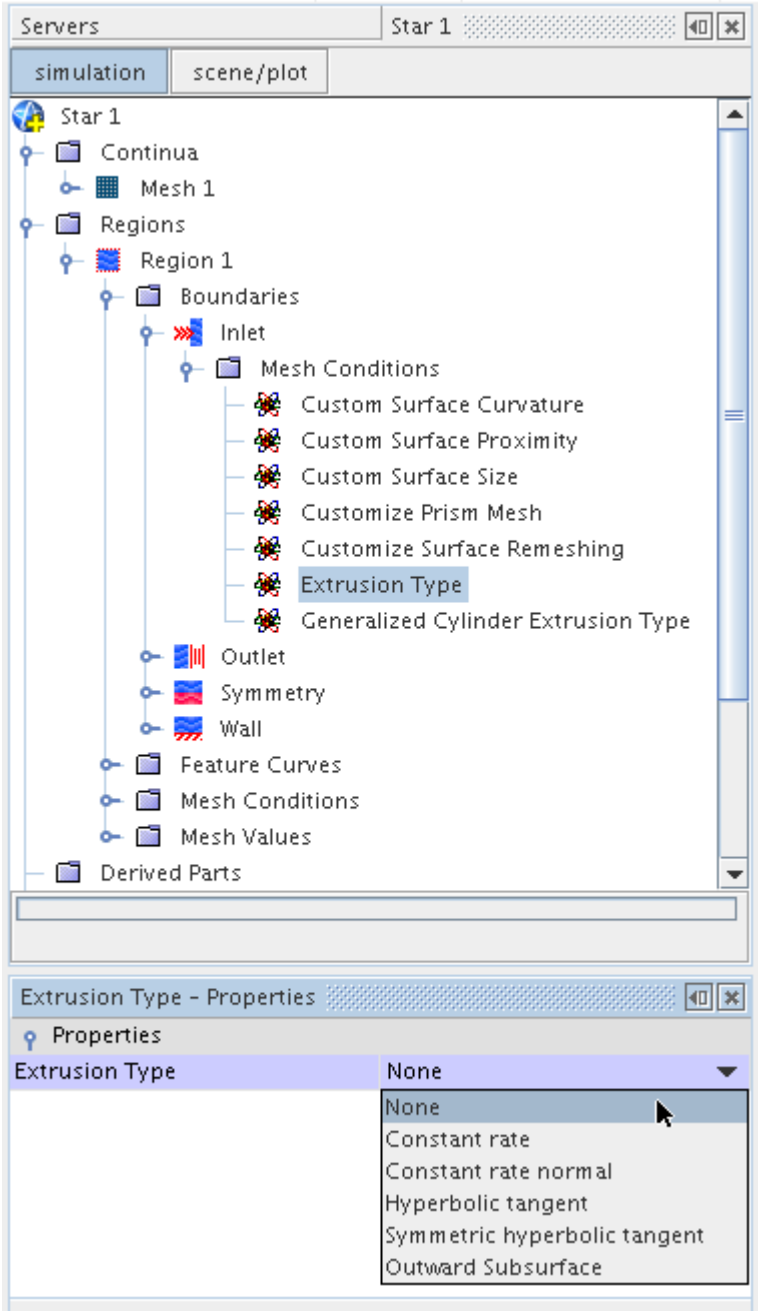
Extrusion Type Properties

Extrusion Parameters Properties

Extrusion Type Properties

When the Extruder Meshing Model has been selected, an additional **Extrusion Type** node is added to the **Mesh Conditions** manager node on each boundary.

The *Extrusion Type* property of this node is used to activate the extruder on that boundary.



Extrusion Type	**None**	The default value of the property indicating that the feature is deactivated.
	Constant Rate	The mesh is extruded in a specified direction with constant thickness layers.
	Constant Rate Normal	The mesh is extruded in the normal direction with constant thickness layers.

	Hyperbolic Tangent	The **Hyperbolic Tangent** option uses a hyperbolic stretching law to compute the cell distribution growth in the direction of the extrusion. This option is geared specifically for aerospace and other applications where the boundary layer resolution is considered critical.
	Symmetric Hyperbolic Tangent	The **Symmetric Hyperbolic Tangent** option grows the mesh hyperbolically from both ends.
	Outward Subsurface	The **Outward Subsurface** option provides an alternative scheme that is based on the subsurface generation method. In general, the other extrusion methods simply extrude the mesh in a normal direction to the local face by the specified distance, which can sometimes lead to problem cells in some areas. The outward subsurface method looks at the surrounding faces and extrudes in a direction that is less likely to cause problems. Additionally, it can partially or wholly retract the extrusion in areas where performing the full extrusion would lead to errors in the resultant mesh. This method is suited to complex meshes that require multiple boundary extrusions to generate a solid exterior or interior domain for the purposes of including heat transfer as part of the solution. The frozen boundaries extruder property must be turned on in order to use the outward subsurface extruder option.

See Also:

Extruder Mesher Properties

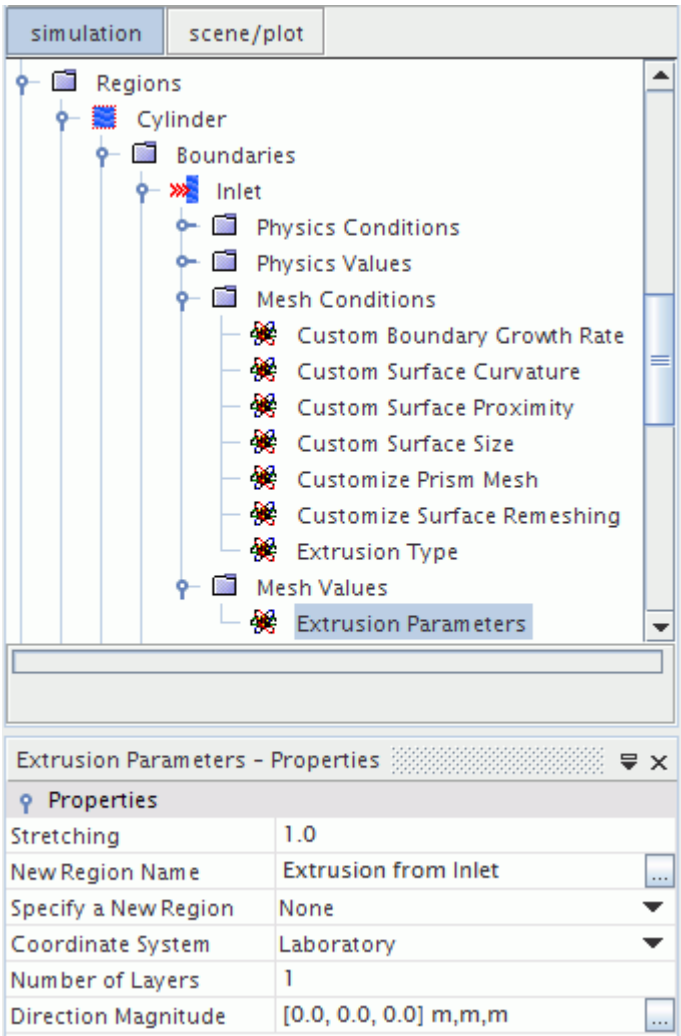
Extrusion Parameters Properties

Extrusion Parameters Properties

All five extrusion types require that the extrusion parameters be set in order to obtain a successful extrusion.

Additionally, when selecting the **Constant rate**, **Hyperbolic tangent** or **Symmetric hyperbolic tangent** options, the direction of the extrusion in the supplied coordinate system must also be specified.

By default, the *Extrusion Type* option is set to **None**, indicating no boundary extrusion. Selecting any of the other options from the pull-down menu activates an additional property node in the **Mesh Values** node which allows the extrusion parameters to be set for the boundary. This node is shown below:



Extrusion parameters can only be set at the boundary level for each boundary and therefore other local or global settings do not affect the extrusion. If the boundary belongs to an interface, then only remainder boundaries that are not participating in the interface are extruded. If there are no remainder boundaries, then the extruder settings have no effect on the meshing process for that boundary.

Extrusion Parameter Properties

Stretching	The meaning of the stretching value depends on the type of extrusion selected. When selecting a constant rate extrusion, the *Stretching* parameter controls the relative length of the extruded cells in the direction of extrusion. The parameter is the ratio of the size of the end most layer to the base or root layer. A value above 1 produces clustering towards the originating boundary while a value below 1 produces clustering away from it. When a hyperbolic tangent type of extrusion option has been chosen, the *Stretching* parameter is a non-dimensional clustering factor. If a value close to zero is used, then a near equal layer spacing is achieved while values above zero result in a clustering of the cell layers towards the originating root boundary. When using the symmetric option, the layers are also clustered towards the end boundary. Any value of *Stretching* greater than zero is allowed. The default value of 1.0 produces different results depending on the extrusion type selected.	
Specify a New Region	The *Specify a New Region* option determines where the extruded mesh goes. The options appear in the *Properties* window for the **Extrusion Parameters** node in the form of a drop-down menu. When a new extruded region is created, it contains the following boundaries: • The base boundary (for any portion of the original boundary that does not fall within the internal interface between the original region and the extruded region. In most cases it will contain zero faces) • The top boundary (a copy of the extruded boundary, displaced according to the extrusion law and extrusion magnitude) • One or more side boundaries, each of them named **extension of {Boundary Name}** • An Internal Interface Boundary New regions are not automatically displayed as part of the model, so to view the result you must add any new region parts to the current displayer	
	None	Creates the extrusion mesh as part of the same region as its source boundary. This option is the default selection.
	Yes, with Wall Boundaries	Sets the extrusion mesh as part of a new region that is called **Extrusion from {Boundary Name}**
	Yes, with Extended Type Boundaries	Sets the extrusion mesh as part of a new region which is an extension

		of the source region with the same extended boundary conditions. The new region that is created is called **Extrusion from {Boundary Name}.** This setting moves the original boundary type from the extruded boundary to the extrusion top. The original extruded boundary itself becomes a **Wall** boundary (as its extrusion base boundary counterpart). Note: The extruder can only generate a new region when the source boundary has a surface representation (that is, from the surface remesher or wrapper). Therefore, generate a surface mesh before specifying a new region.
Coordinate System	Specifies the coordinate system to use for the Constant rate, Hyperbolic tangent, or Symmetric hyperbolic tangent extrusion types. then the coordinate system in which the extrusion is performed must also be specified. Extrusions can be carried out in cartesian, cylindrical, or spherical type coordinate systems.	
Number of Layers	Determines the number of cells in the direction of extrusion. The default value is `1`, which is the minimum. There is no maximum limit to the number of layers allowed.	
Direction Magnitude	The total distance in each coordinate direction by which the mesh is extruded in the specified coordinate system. The direction magnitude value can be set for each direction using any of the allowable units and must be greater than 0. When the normal constant rate or outward subsurface extrusion type has been selected, the **Normal Extrusion Parameters** node is accessible instead: In this instance, specify the *Magnitude* in the normal direction using any of the allowable units.	
Use Average Normal	The *Use Average Normal* option of the **Normal Extrusion Parameters** node forces the extruder to compute an average face normal across the whole boundary from which the extrusion is grown. Using a consistent extrusion normal direction helps to maintain a uniform extrusion along a	

	single direction, and is useful for cases where there are small imperfections on an otherwise planar surface.
	Backward Compatibility If you use this feature in simulation files that were created using versions of STAR-CCM+ older than 7.04, the extruder creates a single side boundary as it did in earlier versions of STAR-CCM+. To change this behavior, two techniques are available: • Extrude the boundaries with the *Create a New Region* option deactivated. • Delete the new region, make sure that the *Create a New Region* option is active, and repeat the extrusion action. This step creates a separate region with the side boundaries automatically split. **Working with the Frozen Boundaries Option** Normally you can deactivate the frozen boundaries option. However, if the *Create a New Region* option is active for two or more adjacent boundaries, the extruder works as if frozen boundaries were activated.

See Also:

Displaying Remainder Boundaries
Extrusion Type Properties
Extruder Mesher Properties

Thin Mesher

The Thin mesher is used to generate a prismatic type volume mesh for thin volumes within parts or regions. Thick or bulk portions of the same geometry are meshed with the core volume mesher.

The different portions of mesh are joined continuously without abrupt changes in the mesh between the bulk and thin portions of the geometry.

When using Parts-Based Meshing (PBM), there is one implementation of the Thin Mesher which can be used for various applications. When using Region-Based Meshing (RBM), there are two implementations of the Thin Mesher within STAR-CCM+ for similar but differing applications.

- **Thin Mesher (PBM)**
- **Thin Mesher (RBM)**
- **Embedded Thin Mesher (RBM)**

Thin Mesher (PBM)

In Parts-Based Meshing, when the Polyhedral Mesher or Tetrahedral Mesher is selected for an Automated Mesh operation, the Thin Mesher is available to select as an Optional Volume Mesher. This implementation of the Thin Mesher automatically recognizes areas of the Geometry that are thin or bulk. The Thin Mesher creates a prismatic type mesh for the areas that are recognized as thin and the bulk areas are meshed with

the core volume mesher that is selected. You can refine the parameters that the Thin mesher uses to recognize thin or bulk areas. See Determining if Geometry is Thin or Bulk.

The Thin mesher creates a conformal mesh between any concurrent parts that are included in the same Automated mesh operation – regardless of the parts having a prismatic or polyhedral volume mesh.

There may be cases where fan-shaped cells are generated resulting from how the thin mesher operates in parts-based meshing.

In some cases, the number of thin layers that are generated by the Thin Mesher can drop to one, regardless of the number of layers that you specify. This behavior can occur because the thin mesher performs checks just before generating the prism cells. The checks attempt to predict the worst cell aspect ratio and rescale the number of layers to find a compromise between the quality, while attempting to honor the number of layers that you specify.

Thin Mesher (RBM)

Select the **Thin Mesher** for geometries that have no other region surrounding them and are predominantly thin or have thin structures included in them.

The **Thin Mesher** is used to generate a prismatic type mesh throughout the thin portions of the geometry instead of using other core volume meshers.

Embedded Thin Mesher (RBM)

Select the **Embedded Thin Mesher** for geometries that are entirely contained within another region and are predominantly thin or have thin structures included in them.

Note: If necessary, to determine which parts of the geometry are considered thin, set the *Customize Thickness Threshold* expert property. Specifying a custom number of thin layers forces STAR-CCM+ to apply the **Embedded Thin Mesher**. See Determining if Geometry is Thin or Bulk.

Using the Thin Meshers

The **Thin Mesher** is typically used for thin plate geometries. In these geometries, good quality cells are required to capture the solid material thickness adequately. Heat transfer analysis can then be performed between any geometries that are in contact with each other.

For example, the **Thin Mesher** can be used during the mesh generation stage of an analysis involving the heat transfer between the thin plates of a car chassis.

Using the same example, if the car chassis is contained in a region representing (say) air, which is also included as part of the analysis, use the **Embedded Thin Mesher**.

Contents:

What Is the Thin Meshing Model?
Determining if Geometry is Thin or Bulk
Customizing Thin Mesher Parameters
Thin Mesher Expert Properties
Troubleshooting the Thin Meshers

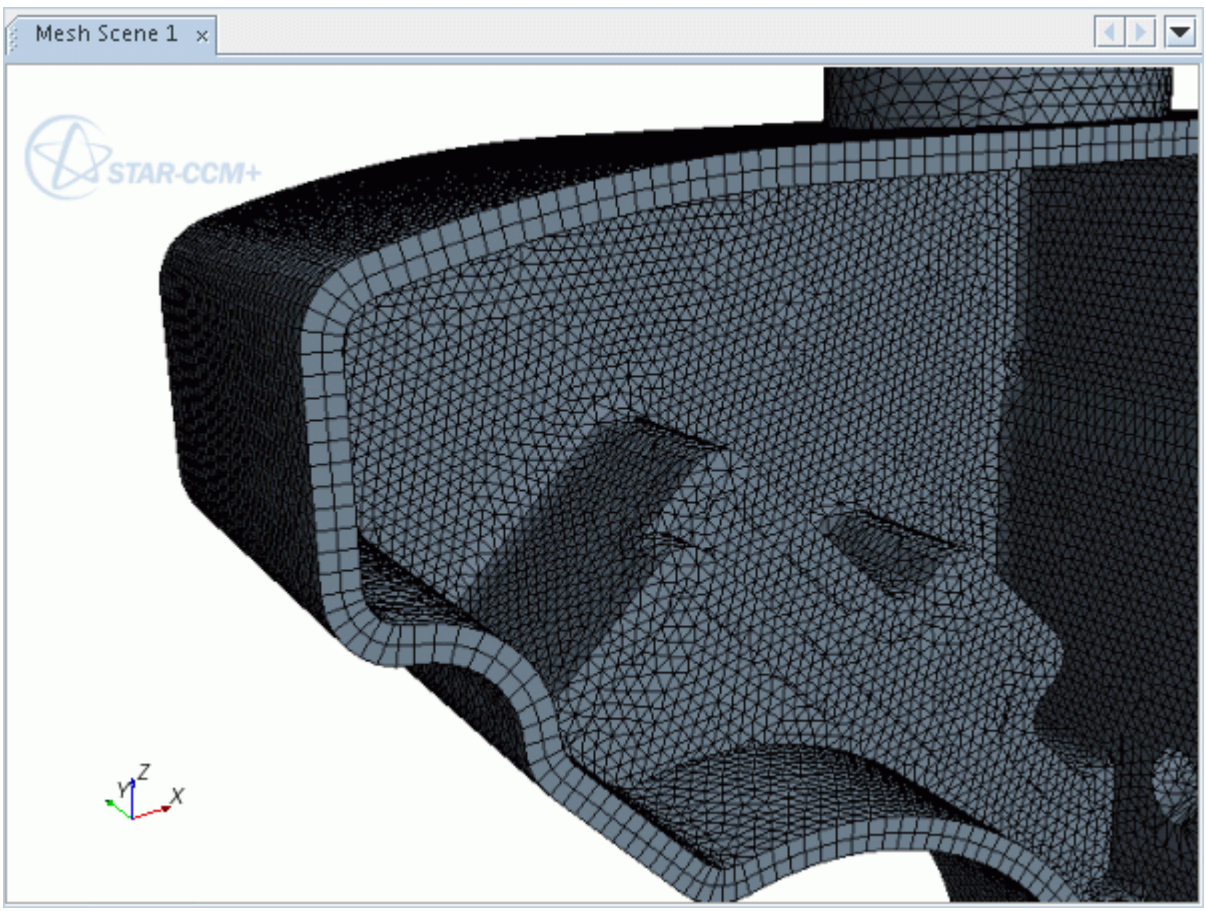
What Is the Thin Meshing Model?

When activated, the thin meshing model allows thin areas in the geometry to have a prismatic type volume mesh. Using this mesh improves the overall cell quality and reduces the cell count when compared with an equivalent tetrahedral or polyhedral type core mesh.

Using this type of mesh can aid the convergence of simulations involving conjugate heat transfer in thin geometries. In Region-Based Meshing (RBM), the Thin Mesher can be activated independently of the other volume meshers or, as part of an “embedded” process within the core volume mesher. In Parts-Based Meshing (PBM), the Thin Mesher is available to select as an Optional Volume Mesher after either the Polyhedral or Tetrahedral volume mesher is selected. You can specify the number of layers in the thin mesh and the threshold size for thin/non-thin structures.

When using Region-Based Meshing, you can choose between a triangular or polyhedral type thin mesh. In Parts-Based Meshing (PBM), the core volume mesher that you select determines if a triangular or polyhedral type thin mesh is generated.

An example thin mesh with two triangular based prismatic layers, generated using the default thin mesher (RBM), is shown below:



A good quality triangulated surface mesh is required as the input for the thin mesher. Failure to provide such a surface results in either a poor quality thin mesh being produced, or no mesh at all. Where the geometry is considered not to be thin by the meshing model, the prismatic mesh in the thin portion transitions naturally to

a normal core type mesh in the non-thin section. The threshold for this transition occurs if the distance between two opposing surfaces (the base surface and the target surface) is greater than four times the average edge length of the base surface.

The properties of the thin mesh can be changed to provide more meshing controls during the mesh process. The input values can be set on two different levels, that are listed below in lowest to highest priority:

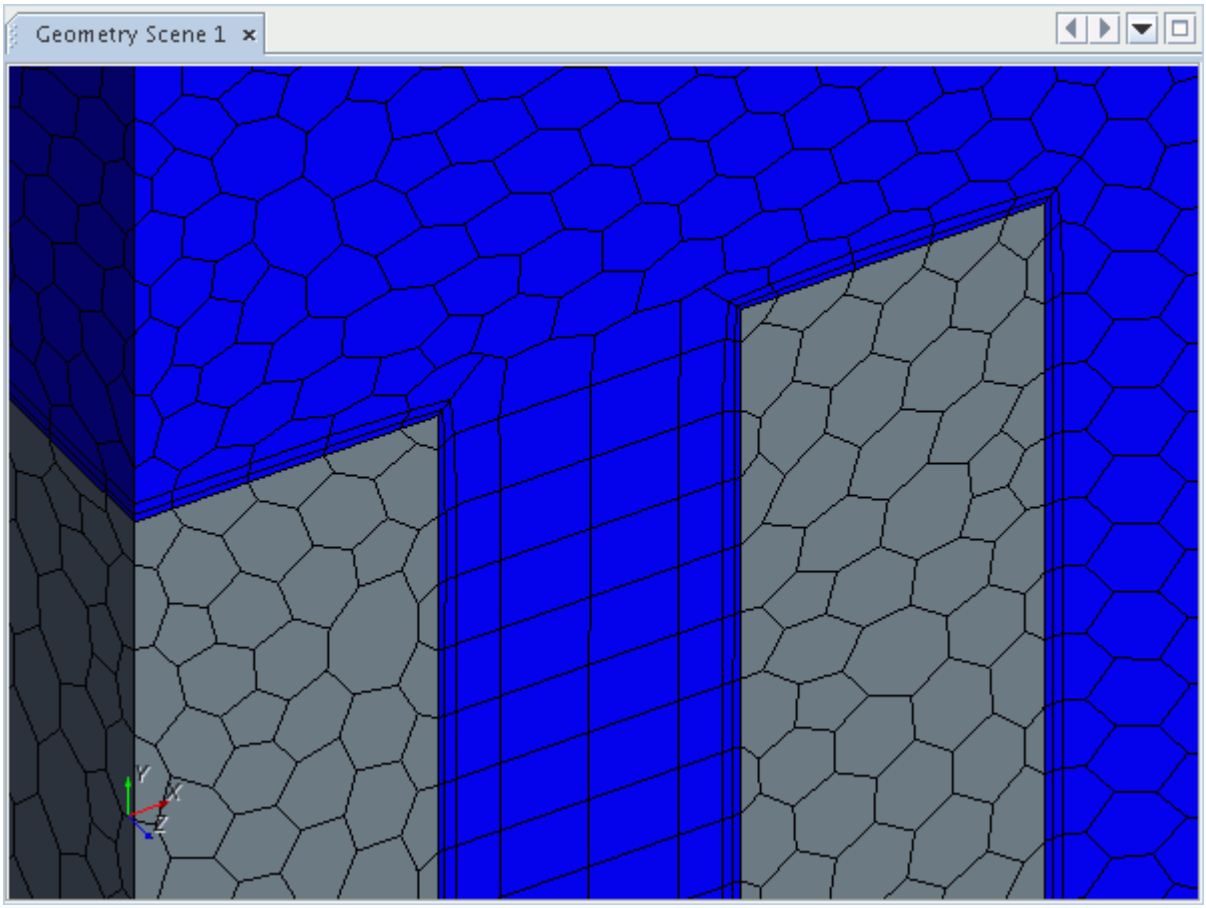
- Global Level
- Region / Parts Level

In Region-Based meshing, both types of thin mesher can be used with any interfaces that exist between two or more regions. The default Thin mesher ensures a conformal mesh between thin geometries in each region that are in contact with one another. However, in the case of the Embedded Thin mesher, a conformal mesh is guaranteed only when the bulk region entirely encloses the thin mesh region. If a boundary plane in the thin region is exposed on the side of the bulk domain, then a conformal mesh is not always possible.

There is no guarantee that a conformal mesh is produced between the thin regions and the surrounding mesh region. However, a mesh is at least guaranteed without needing to activate the per-region or per-part meshing option.

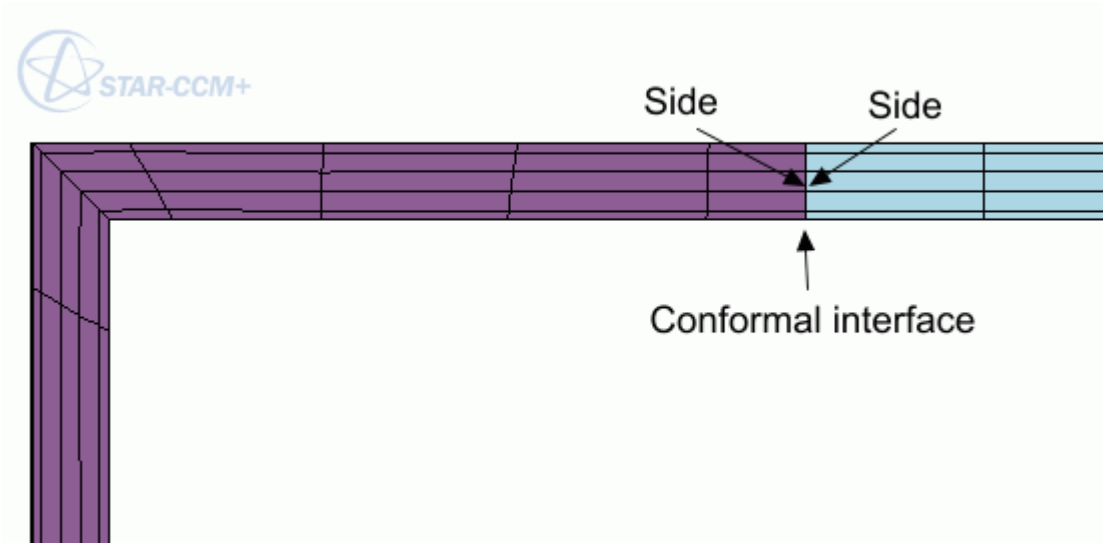
The thin meshers (RBM) cannot currently be used with volumetric controls. No method of specifying the volumetric controls for the default thin mesher (RBM) is provided in this instance. However, when using Parts-Based Meshing, you can create Part Controls in which you can specify customized Values for the Thin Mesher for different geometry parts.

Since the default Thin Mesher (RBM) is used instead of one of the normal core volume meshers, this Thin Mesher cannot be used with the Prism Layer Mesher, Extruder, or Generalized Cylinder models. However, as the Embedded Thin Mesher (RBM) and the Thin Mesher (PBM) work with a core volume mesher, you can use the Prism Layer Mesher within thin sections of the mesh.



Prism meshes cannot grow at thin-thin interfaces because conformality cannot be maintained in such cases. This condition is enforced no matter what other parameter is used.

However, for thin-thin side interfaces, prism layers from the thin mesher are projected onto the interface to produce a conformal mesh.



However, the mesher still collapses prism layers when:

- The number of prism layers on each side are different.
- There are more than two sides that meet at one interface.

See Also:

Thin Mesher Expert Properties

Determining if Geometry is Thin or Bulk

The Thin Mesher meshes as much of a region or part as possible using prism cells. For any areas where the Thin Mesher cannot produce prism cells, polyhedral or tetrahedral cells are generated – depending on the core volume mesher that is used.

When using the Embedded Thin Mesher (RBM), the thin regions of the geometry are sent to the Thin mesher and the bulk regions are sent to the core volume mesher. When using the Parts-Based meshing (PBM) approach, you can set the Thin Mesher to identify thin parts automatically, specify all parts as thin, or specify particular parts as thin.

If the automatic behavior of the Thin Mesher is not suitable, you can manually specify an area as thin or bulk as described below:

To Specify a Part as Thin:

By default, the Thin Mesher (PBM) Property, *Automatic Thin/Bulk Shape Detection*, is deactivated causing all parts to be recognized as thin. However, if you activate this property, the parts are identified as thin or bulk automatically. See Thin Mesh Expert Properties.

To specify a specific selection of parts as thin, or customize thin properties on specific parts:

1. Create a Part Control and set *Parts* to the parts which you want to specify as thin.
2. Select the **[Part Control]** > **Controls** > **Thin Mesher** node and set *Thin Mesher* to **Specify Custom Values**. This setting forces the mesher to recognize the part as thin.
3. If necessary, you can then customize the Thin Mesher Properties. Select the **Thin Mesher** > **Customize** node and activate any *Properties* that you want to customize.

   For each custom thin property that you activate, a node appears under **Values** > **Custom Thin Values** which allows you to specify a custom value.

To Specify a Part as Bulk:

1. Create a Part Control and set *Parts* to the parts which you want to specify as bulk.
2. Select the **[Part Control]** > **Controls** > **Thin Mesher** node and set *Thin Mesher* to **Disable**.

   The parts that are selected are forced to be recognized as bulk.

To Specify a Region as Thin:

1. Expand the **[region]** > **Mesh Conditions** > **Customize Thin Mesher Parameters** node.
2. Set *Customize Thin Mesher Parameters* to **Specify Custom Values**.

3. If the default number of layers is not sufficient, select the **[Region]** > **Mesh Values** > **Thin Mesher Layers** node and update the *Number of Layers*.

Setting this property forces the mesher to recognize the region as thin, regardless of setting any custom values that are set.

To Specify a Region as Bulk:

1. Expand the **[region]** > **Mesh Conditions** > **Customize Thin Mesher Parameters** node.
2. Set *Customize Thin Mesher Parameters* to **Disable**.

Setting this property causes the mesher to recognize the region as bulk. A mesh is produced using the polyhedral volume mesher.

Note: If you deactivate the thin mesher, ensure that there is an alternative volume mesher available.

See Also:

Thin Mesher Expert Properties

Customizing Thin Mesher Parameters

Customizing Thin Mesher Parameters

At the region or parts level, three different parameters can be specified to control the behavior of the thin mesher.

Customize Thin Mesher Parameters (PBM)

In Parts-Based meshing, when a custom Part Control is created and the *Thin Mesher* Control is set to **Specify Custom Values**, three parameters are available to customize:

- Customize Thin Number of Layers
- Customize Thickness Threshold
- Customize Surface Size Ratio Threshold

To customize thin mesher parameters at the part level,

1. Select the **[Part Control]** > **Controls** > **Thin Mesher** node and set *Thin Mesher* to **Specify Custom Values**.
2. Select the **Thin Mesher** > **Customize** node and activate the *Properties* that you want to customize.
3. Expand the **Part Control** > **Values** > **Custom Thin Values** node and set the custom values as required.

Customize Thin Mesher Parameters (RBM)

In Region-Based meshing, when the Customize Thin Mesher Parameters option is activated, three parameters are available to customize:

- Thin Mesher Layers
- Thin Solid Thickness
- Thin Mesher Solid Surface Size Ratio

To customize thin mesher parameters at the region level,

1. Select the **Region** > **Mesh Conditions** > **Customize Thin Mesher Parameters** node and select a customization option from the drop-down menu.

Customize Thin Mesher Parameters Properties

Customize Thin Mesher Parameters	**Use Default Values**	Specifies that the region uses the thin mesher settings that are specified at the global level. This option also allows the mesher to decide whether a region is thin or bulk.
	Specify Custom Values	Specifies that the region uses the thin mesher settings that are specified at the region level. This option also forces the mesher to consider the region as thin.
	Disable	For the embedded thin mesher, the region does not use any thin mesher settings; hence, the mesher considers the region as bulk. For the thin mesher, the mesher still generates prismatic cell layers, as there is no alternative volume meshing model available.

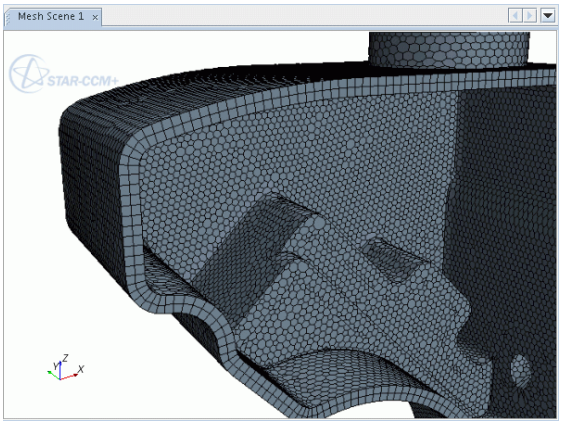
See Also:

Determining if Geometry is Thin or Bulk

Thin Mesher Expert Properties

Thin Mesher Expert Properties

Polyhedral Cells Type (RBM)	The polyhedral cells type option determines the type of prismatic cells that the thin mesher generates.	
	Triangular prisms	This option results in a triangular based mesh being used instead. In regions that are determined to be thin, the cells are regular three sided prisms. In non-thin regions, a regular tetrahedral type mesh is produced.
	Polygonal prisms	This option, which is selected by default, means that a polyhedral based mesh is used. In thin regions, multi-sided prismatic cells are created. In non-thin regions, a regular polyhedral type mesh is produced.

		An example thin mesh using polygonal prisms is shown below: 
Run Optimizer (RBM)	This option is deactivated by default and is only available for the Thin Mesher (RBM) – not for the Embedded Thin Mesher (RBM). Activate this option to allow more optimization of the thin mesh when the polygonal prism mesh type has been set. A better quality mesh is then produced in the thin mesh regions at the cost of longer mesh generation time. This option has no effect for the triangular prism mesh type. Deactivate this option to stop further optimization of the thin mesher.	
Automatic Correction (RBM/PBM)	This option is activated by default. To ensure that a continuous, valid mesh is obtained, leave this option activated where possible. When activated, this option allows the thin mesher model to use cells other than prisms in thin regions when a locally invalid mesh is originally obtained. Deactivate this option to not allow the thin mesher to use cells other than prisms.	
Customize Thickness Threshold (RBM/PBM)	By default, all parts are marked as potentially thin and the thin mesher determines automatically which parts are thin and which are bulk. Activate this option to override this behavior and supply an appropriate cut-off size, where geometry that is larger than this cut-off size is treated as bulk. Alternatively, you can force thin meshing for a region (RBM) using the region level mesh conditions. When activated, this option creates the node: • **Default Controls** > **Thin Mesher Maximum Thickness** (PBM) • **Reference Values** > **Thin Solid Thickness** (RBM) The thin mesher treats geometry that is larger than the specified thickness threshold as bulk. Note that the purpose of this option is to exclude regions from the thin mesher rather than mark regions as thin.	

	When deactivated, the thin mesher determines automatically which parts of the geometry are thin and which are not.
Customize Surface Size Ratio Threshold (RBM/PBM)	Allows you to specify which parts of a mesh are considered thin based on the product of the local surface cell size and a Thin Mesher Surface Size Ratio that you specify. The thin solid surface size ratio can be set at the global level, and also for individual regions (RBM). When activated, this option creates the nodes: • **Default Controls** > **Thin Mesher Surface Size Ratio** (PBM). • **Default Controls** > **Thin Solid Surface Size Ratio** (RBM) The thin mesher determines which parts of the geometry are thin based on the product of the local surface cell size and the Thin (Mesher/Solid) Surface Size Ratio that you specify. Any geometry that is thinner than both this size and the thickness threshold is meshed with prismatic layers. When deactivated, the thin mesher determines automatically which parts of the geometry are thin and which are not.
Automatic Thin/Bulk Shape Detection (PBM)	This option, which is available when using the Parts Based Meshing approach, determines if all parts are thin or not. If parts are considered thin, the parts are meshed with the thin mesher which creates prismatic cells in thin areas, and core volume cells elsewhere. Activate the option so that parts are automatically detected as thin or bulk. Only parts which are determined as thin are meshed with the Thin Mesher. Non-thin parts are meshed with the core volume mesher that is selected. Deactivate this option to consider all parts as thin.
Optimization Cycles (RBM/PBM)	Increasing this value, increases the quality of the volume mesh, at the expense of increased CPU time in generating the mesh. The default is set to 1 and the recommended range is from 1 through 8.
Quality Threshold (RBM/PBM)	Allows you to specify the maximum cell quality value that you want the mesher to aim for. Increasing this value increases the cell quality, at the expense of increased CPU time in generating the mesh. The default is set to 0.4 and the recommended range is from 0.4 through 1.0.
Force Thin # Layers (PBM)	By default, this option is deactivated and the thin mesher can reduce the number of thin layers depending on the quality of the cells. Activate this option to force the thin mesher to use the number layers that you specify, regardless of the quality implications.

Thin Mesher Layers (RBM) / Number of Thin Layers (PBM)

The following nodes are available for you to specify the number of prismatic layers that the thin mesher creates in geometries that are considered to be thin:

- **Automated Mesh** > **Default Controls** > **Number of Thin Layers** (PBM)
- **Mesh 1** > **Reference Values** > **Thin Mesher Layers** (RBM)

Thin Mesher Layers (RBM) / Number of Thin Layers (PBM) Properties

Number of Layers	The default value is `1`. Any integer value greater than 0 can be specified. For a geometry that contains both thick and thin sections, a value between 1 through 5 is suggested in order to prevent low quality cells or meshing failure. A minimum of two layers are always produced when using polygonal prisms. No such restriction exists when using triangular prisms.

Thin Mesher Maximum Thickness (PBM) / Thin Solid Thickness (RBM)

The following nodes are available for you to specify the threshold thickness, below which the geometry is considered to be thin.

- **Default Controls** > **Thin Mesher Maximum Thickness** (PBM)
- **Reference Values** > **Thin Solid Thickness** (RBM)

For these nodes to be accessible, activate the Customize Thickness Threshold model property.

Thin Mesher Maximum Thickness (PBM) / Thin Solid Thickness (RBM) Properties

Size Type	The default value of `0.0 m` indicates that the thin mesher model determines the thin/non-thin thickness threshold itself. Specifying a value greater than zero tells the thin mesher to search through the geometry and determine the thin sections (based on the supplied value). Surfaces that do not meet the proximity threshold are considered non-thin and are meshed with a non-prismatic mesh type.	
	Relative to base	Set the size relative to the base size.
	Absolute	Set the size as absolute value with length units.

Thin Mesher Surface Size Ratio (PBM) / Thin Solid Surface Size Ratio (RBM)

The following nodes are available for you to specify the ratio which is used to determine which parts of a mesh are considered thin. Any geometry that is thinner than, the product of the local surface cell size and the ratio that you specify, and the thickness threshold, is meshed with prismatic layers.

Thin Mesher Surface Size Ratio (PBM) / Thin Solid Surface Size Ratio (RBM) Properties

Thin Mesher Surface Size Ratio / Thin Solid Surface Size Ratio	This ratio is multiplied by the local surface size to provide the thickness threshold – below which geometry is considered thin. Therefore, the surface size settings on a face determine whether to mesh with prismatic layers or not. This option is available when the customize surface size ratio threshold option is activated. This ratio is set to 4.0 by default.

See Also:

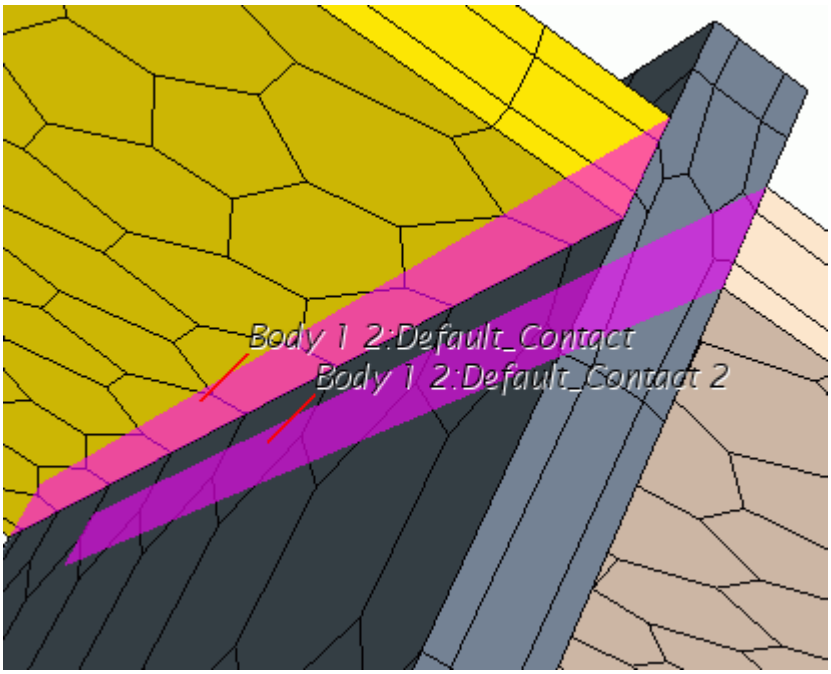
Determining if Geometry is Thin or Bulk
What Is the Thin Meshing Model?
Customizing Thin Mesher Parameters

Troubleshooting the Thin Meshers

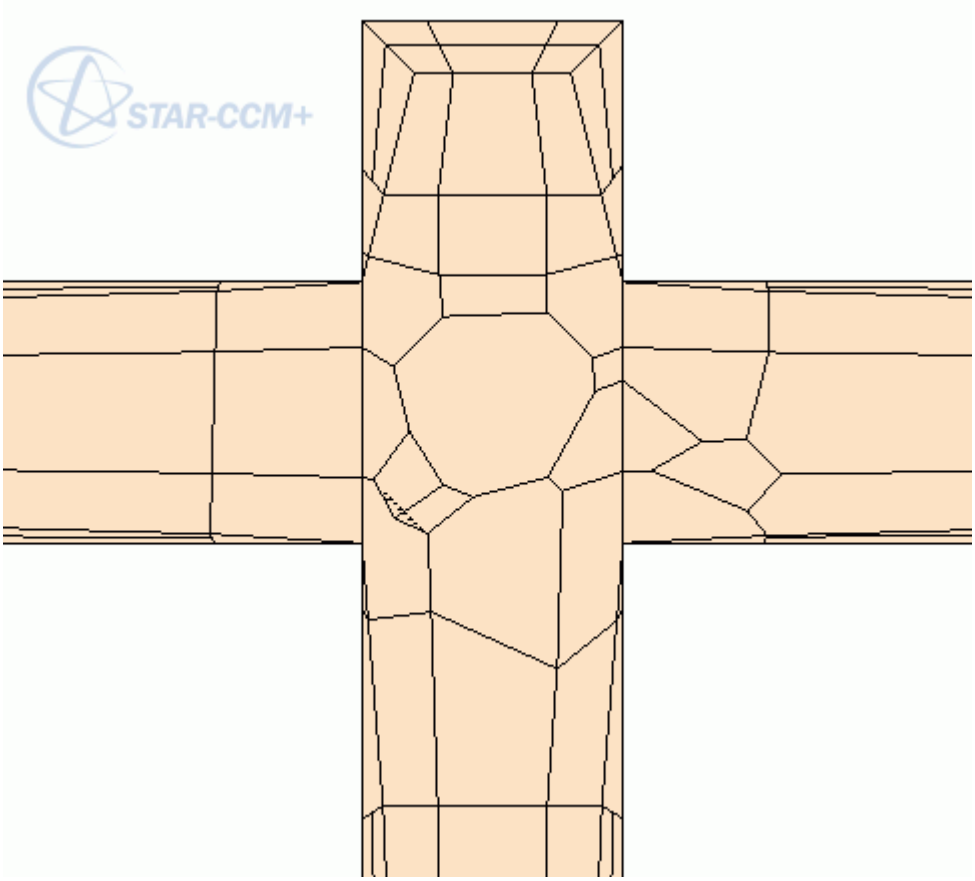
The following section describes issues that can occur due to known behavior of the thin meshers.

Multiple Boundaries in a Thin Geometry

Prism cells are not created on boundary surfaces when there is more than one boundary in a thin portion of geometry. Instead, prism layers are created inside the volume which avoids confusion between prism layers that belong to different boundaries.

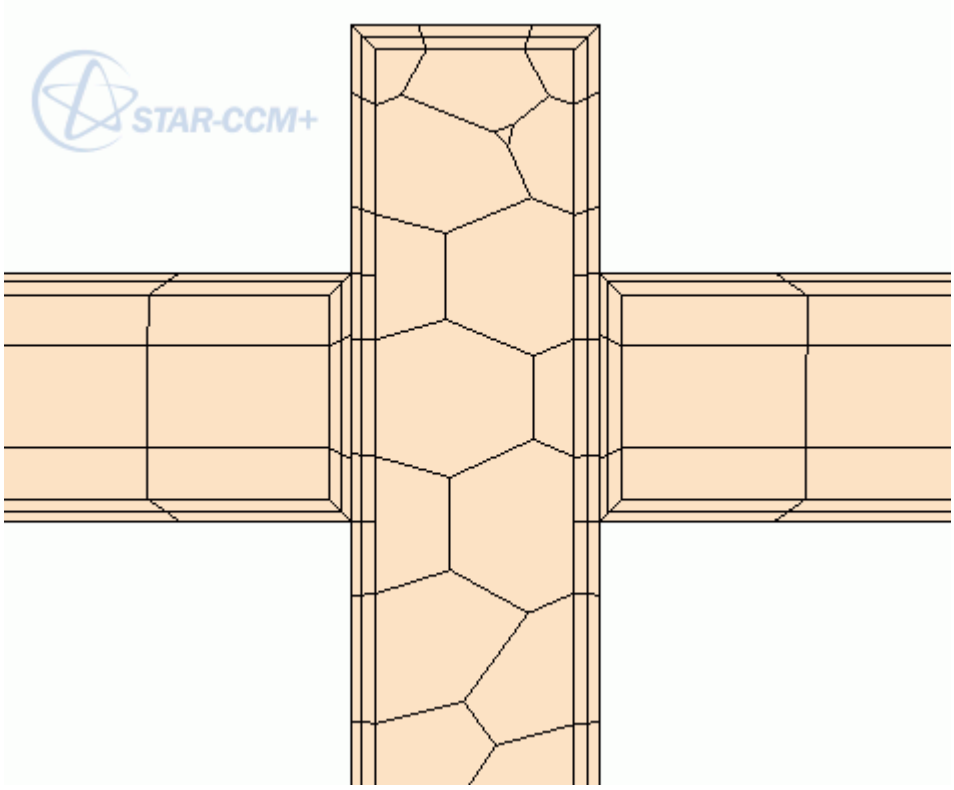


To view the internal prism layers, take a cross section of the geometry.



If you want to maintain prism layers on the two boundaries in the thin portion of the geometry, either:

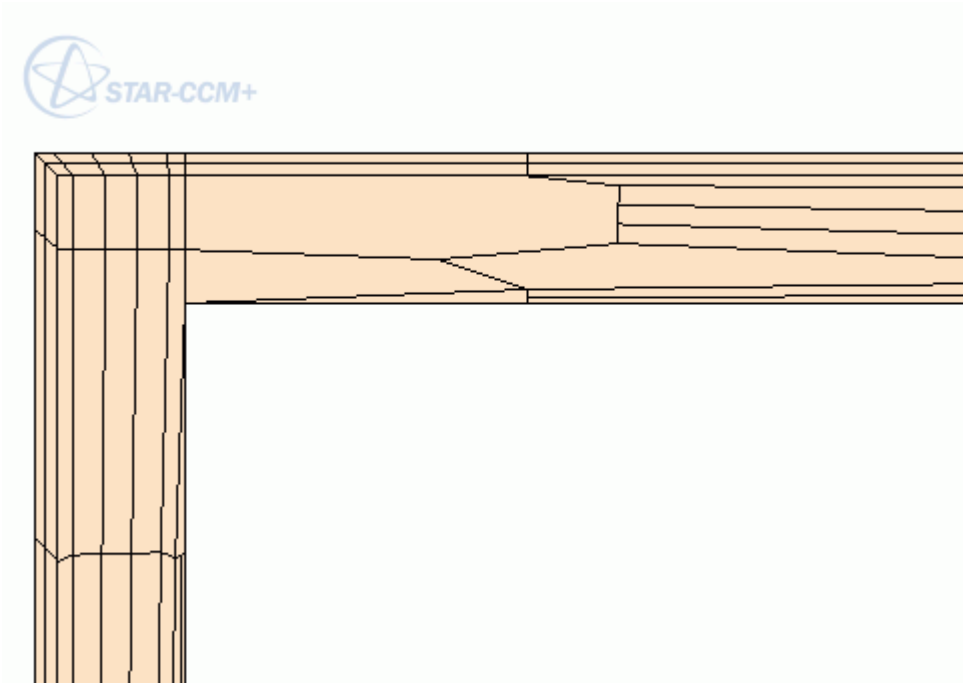
- Activate per-part meshing, or
- Create a separate mesh continuum for the thin portion that has two boundaries. For this mesh continuum, select the prism layer mesher along with a volume meshing model that is not thin.



Using a separate mesh continuum creates non-conformal mesh at the interface.

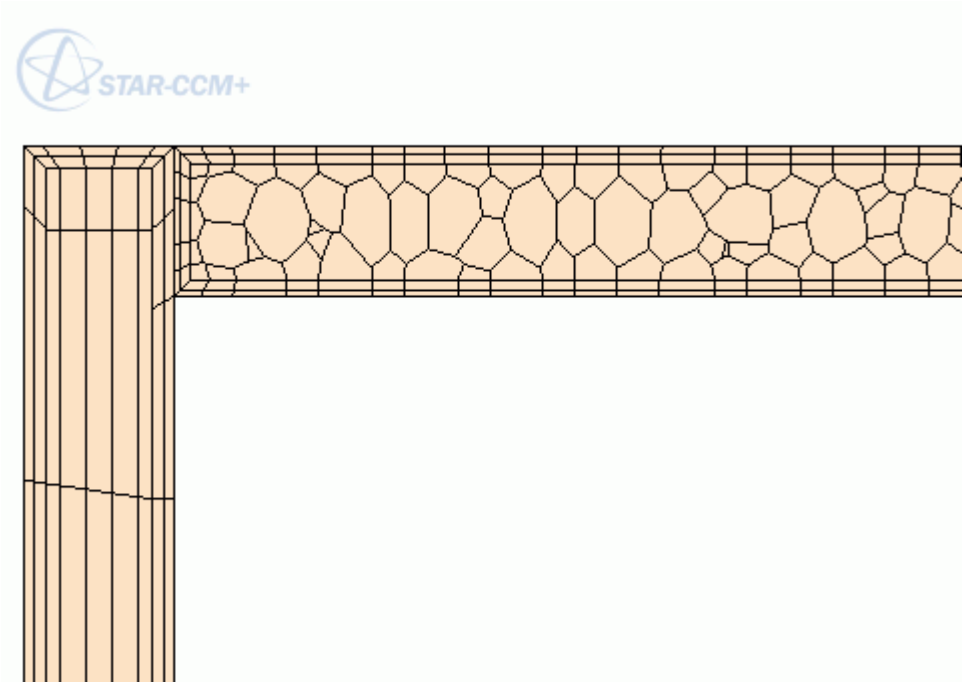
Collapsed Prism Cells at Thin-Thin Interface

When the side of a thin mesh meets the long edge of a thin mesh at an interface, the prism layers in the thin mesh can collapse on one side of the interface. View the internal volume mesh using a cross section.



To overcome this issue and produce a better quality mesh, either:

- specify smaller surface size locally on the interface,
- or, select a non-thin volume mesher along with the prism layer mesher for one of the thin parts.



Using a separate mesh continuum creates non-conformal mesh at the interface.

Corrupted Cells at Sharp Angles

When using the Thin mesher with the Polyhedral mesher, geometry with sharp angles which deviate more than 45 degrees (convex or concave) from a plane surface within a part can unexpectedly cause corrupted cells in the mesh.

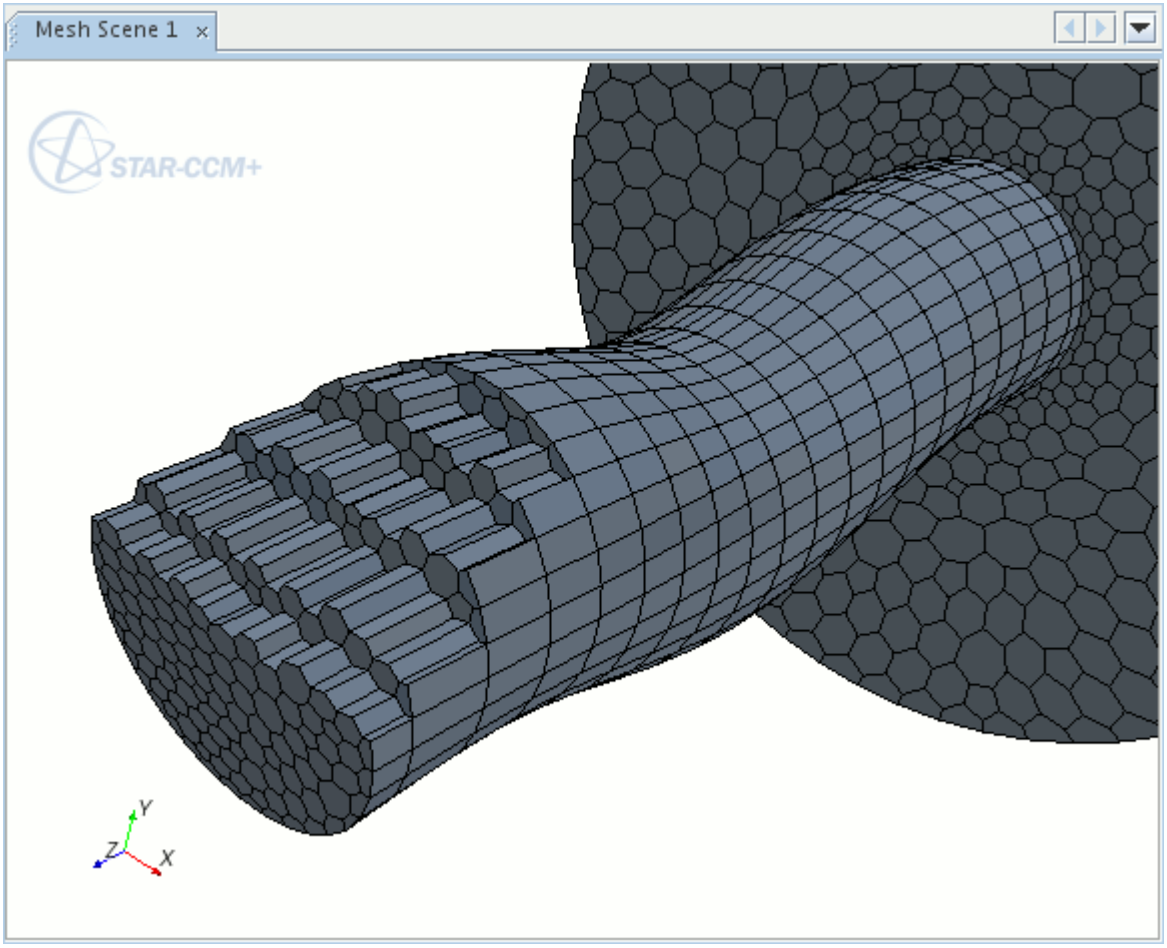
To reduce the possibility of creating corrupted cells, avoid sharp angles in thin parts where possible. Where sharp angles are necessary in thin parts, refine the surface near the sharp angles.

Generalized Cylinder Mesher

The generalized cylinder mesher is used with the polyhedral volume mesher to generate an extruded mesh along the length of parts that are considered to be generalized cylinders.

Using extruded prismatic cells reduces the overall cell count and improves the rate of convergence in some cases. In order to apply the generalized cylinder mesher, candidate cylinders must be defined in separate boundaries. A tool is provided with the mesher that can automatically detect which boundaries are suitable for generalized cylinder meshing, and the properties of the generalized cylinder model are set at the boundary level. The generalized cylinder mesher can be used with the prism layer mesher.

The generalized cylinder mesher is best suited to cases where there are roughly cylindrical sections of geometry, and the direction of flow is parallel to the vessel wall, for example flow down pipes. Such cases can be solved more efficiently by using cells that are oriented to the direction of the fluid flow. An example of a generalized cylinder mesh is shown below; some of the internal cells have been exposed.



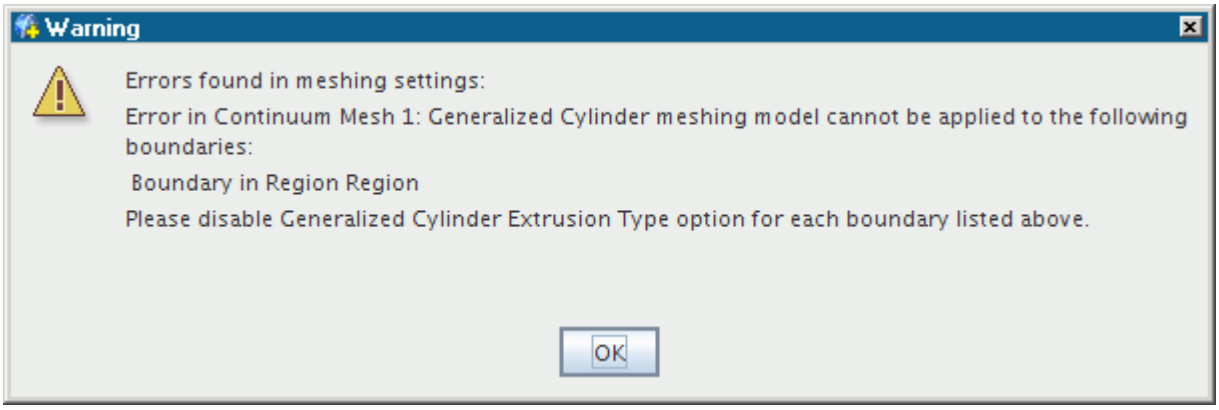
Requirements

The generalized cylinder mesher can automatically detect and mesh boundaries that are suitable for generalized cylinder meshing. However, the generalized cylinder mesher does not automatically mesh any boundaries that it considers unsuitable. If the generalized cylinder mesher does not automatically identify and mesh any boundaries, the following message is displayed in the output window:

```
Candidate cylinder(s) : 0
```

You can manually specify a boundary *Type* as a **Generalized Cylinder Extrusion Type**. See Customizing Generalized Cylinder Conditions and Values.

If you manually specify an unsuitable boundary from which a generalized cylinder mesh cannot be produced, the following message is displayed:



The generalized cylinder mesher does not support cylinders where the axis length is less than 0.0015 m.

Volumetric controls that encapsulate one or both ends of the cylindrical section can indirectly affect the generalized cylinder mesher. When a custom surface size is specified in the volumetric control, this size determines the number of vertices on the cylinder end loops used by the generalized cylinder mesher.

In cases where the generalized cylinder mesher cannot create a mesh using layers of hexahedral cells, it reverts to using a polyhedral mesh instead.

To specify a cylindrical boundary manually, activate the *Generalized Cylinder Extrusion Type* mesh condition for that boundary. See Customizing Generalized Cylinder Conditions and Values.

Contents:

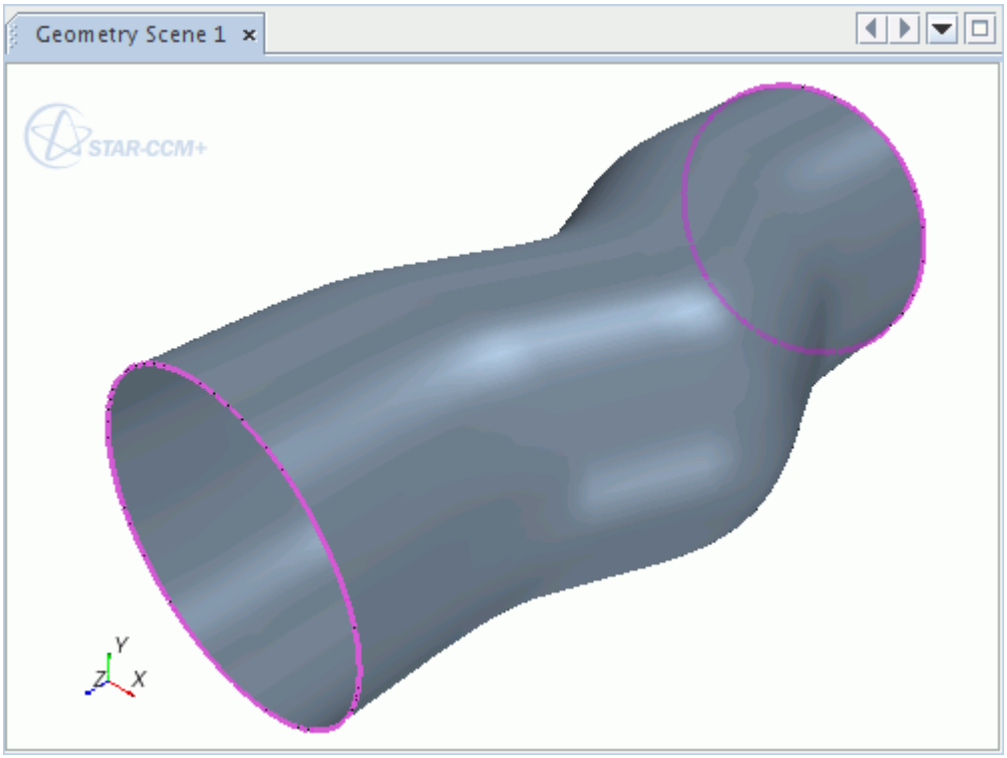
What Is a Generalized Cylinder?
Preparing Boundaries for the Generalized Cylinder Mesher
Generalized Cylinder Mesher Guidelines
Detecting Cylinders Automatically
Clearing All Generalized Cylinder Options
Customizing Generalized Cylinder Conditions and Values

See Also:

Polyhedral Mesher
Prism Layer Mesher
Volumetric Controls
Generalized Cylinder Mesher Guidelines

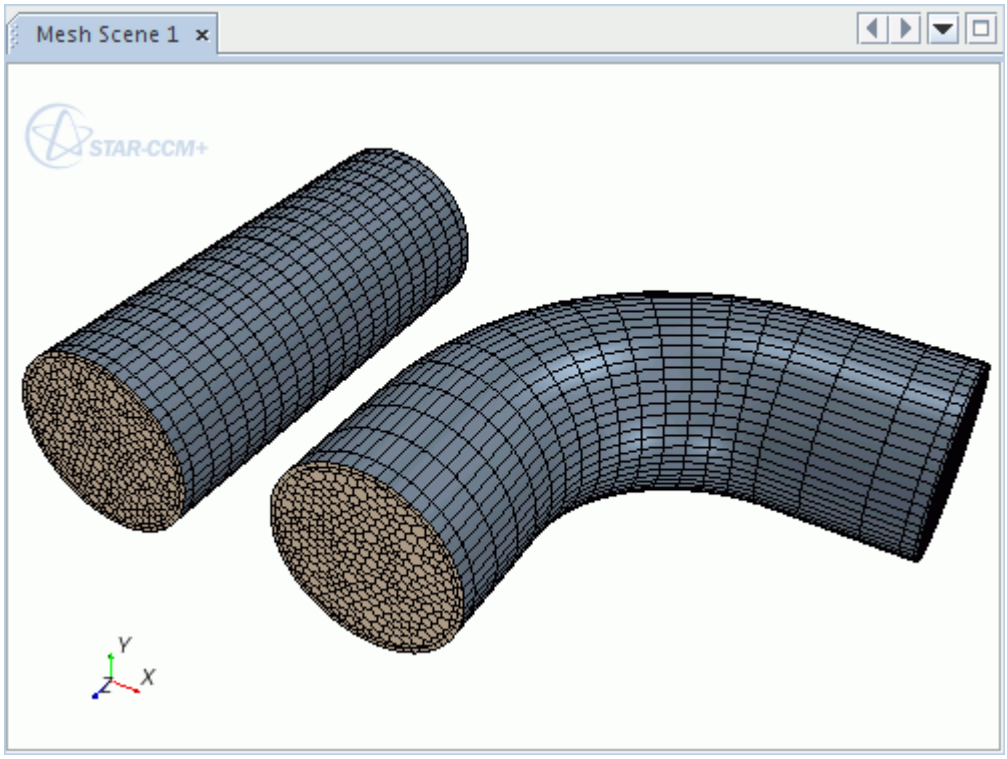
What Is a Generalized Cylinder?

Generalized cylinders are defined as a bounding wall with a closed loop at each end, as shown below.

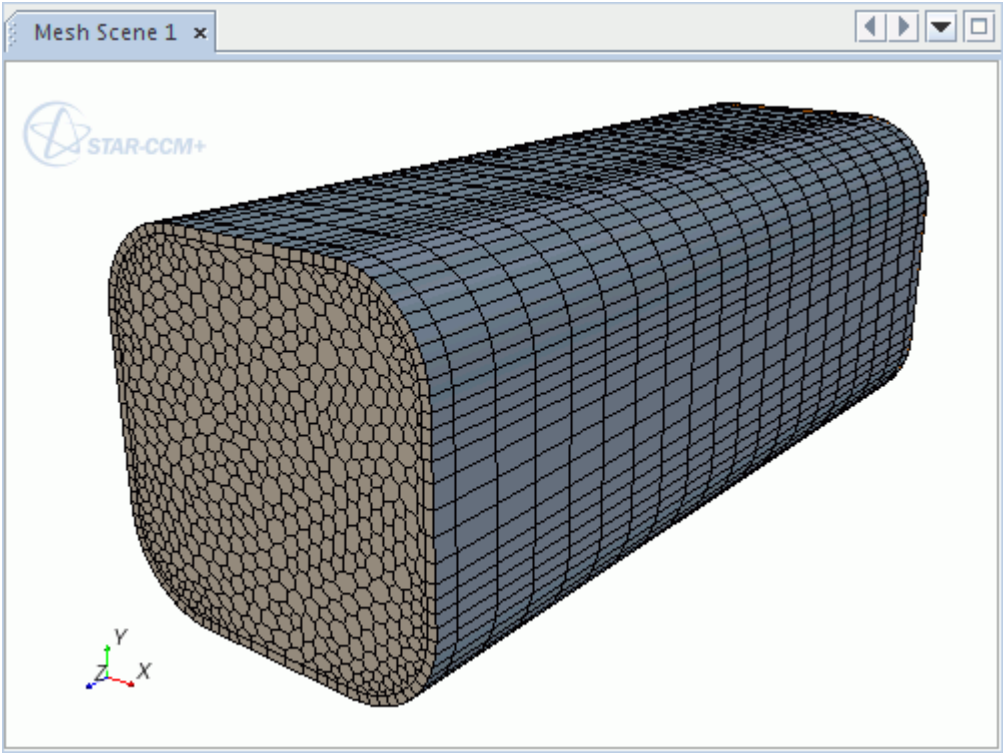


Candidate cylinders have the following characteristics:

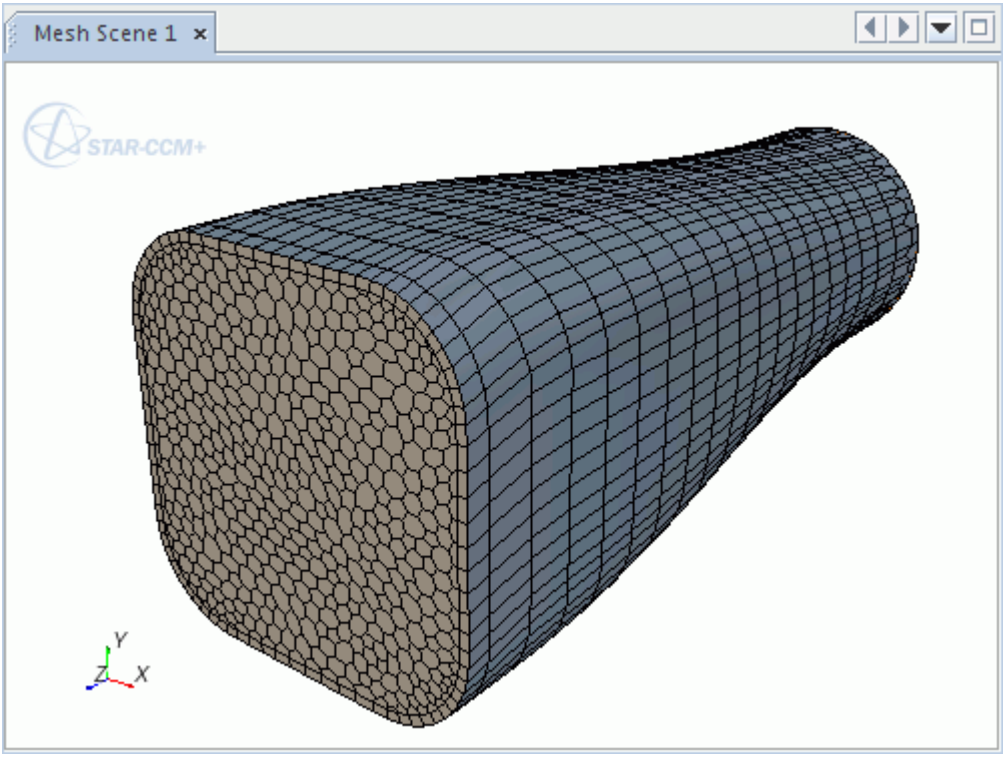
- The cylinder is straight or follows a smooth curve.



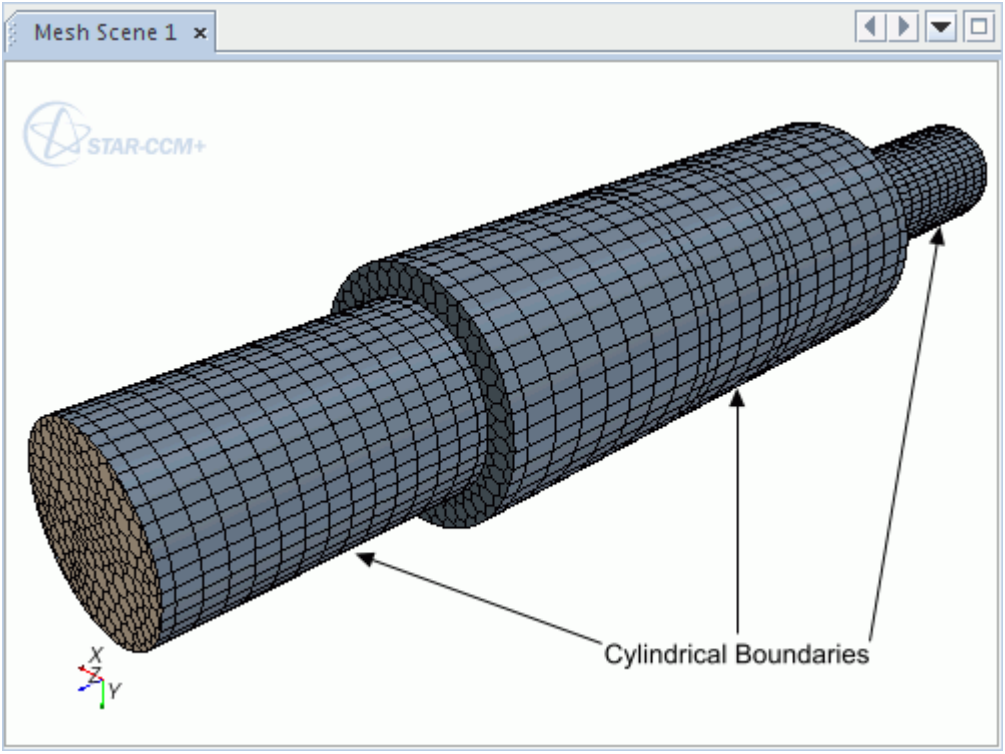
- The cross-sectional shape must have smooth edges—for example, a square with rounded corners, but not a square.



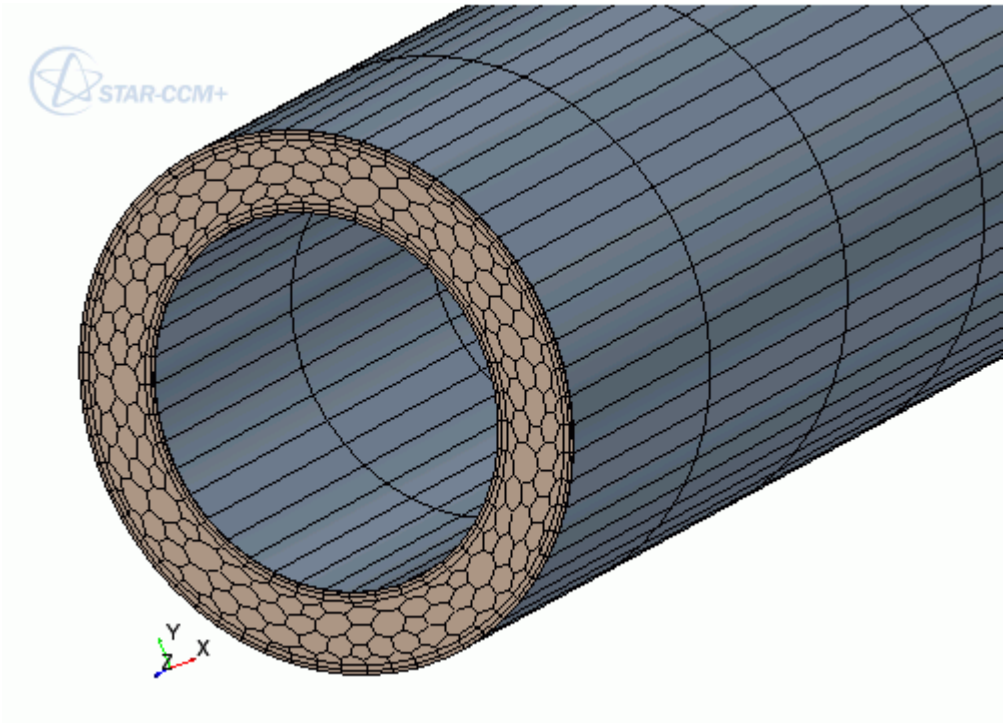
- The size and shape along the length of the cylinder varies, but transitions must be smooth.



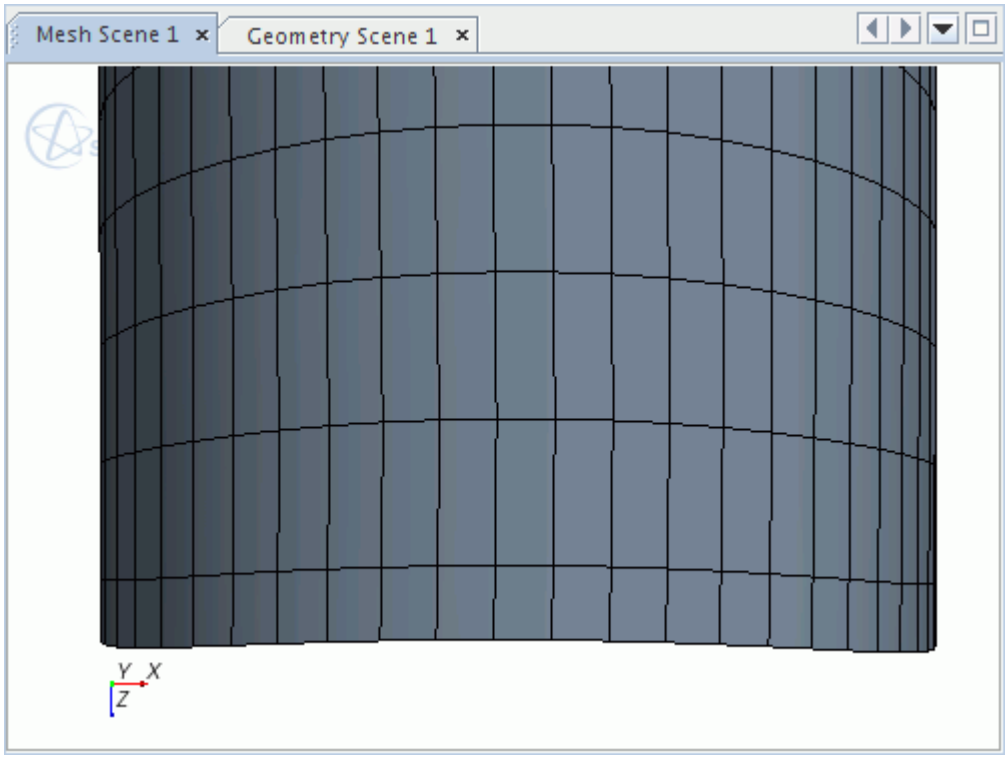
- Cylinders can have a sharp change in cross-section, providing you specify each cylindrical surface as a separate boundary.



- Cylinders can have a hollow centre, as long as the internal and external surfaces of the cylinder are specified as cylindrical boundaries.



- The closed loops that define the start and end of the cylinder can be slightly non-planar.



See Also:

What Are Interfaces?

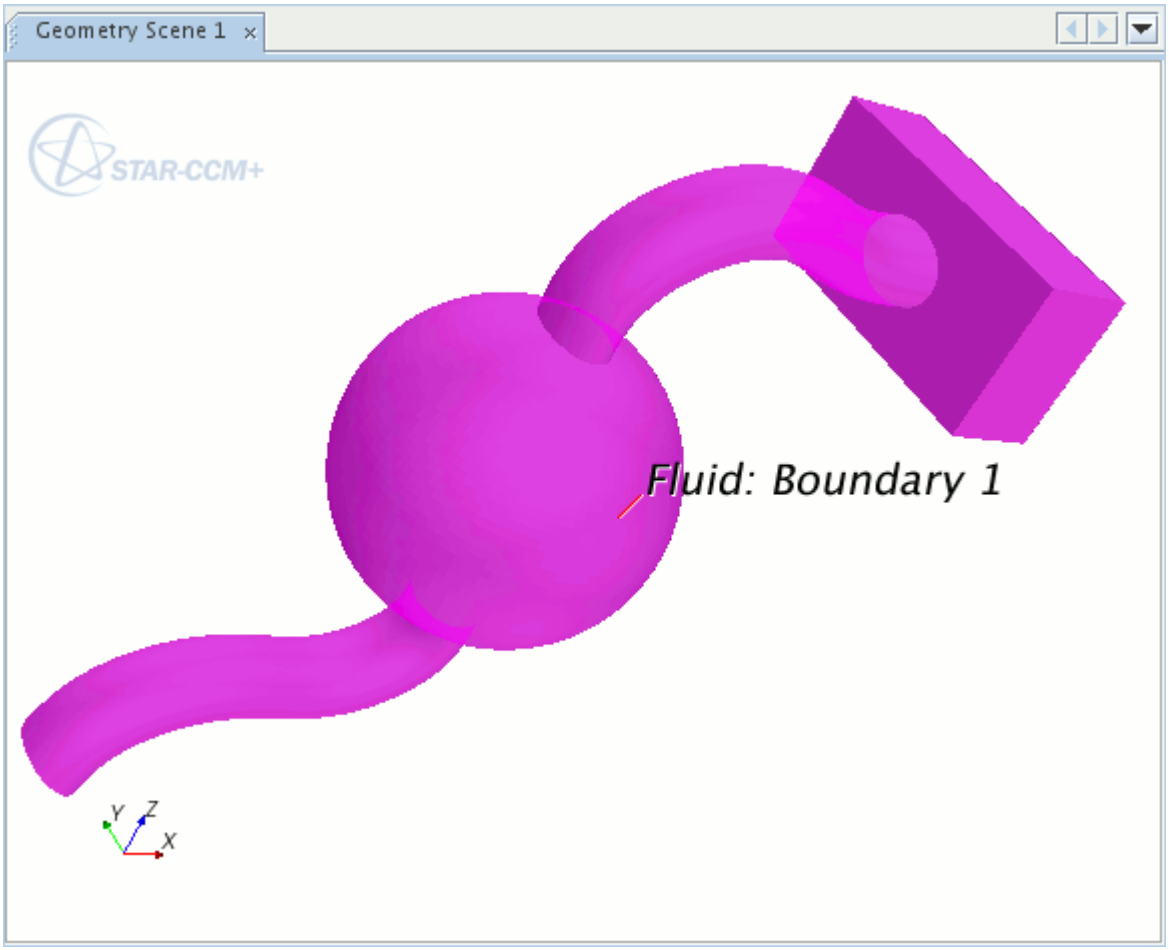
Creating a Boundary Interface

Preparing Boundaries for the Generalized Cylinder Mesher

Before using the generalized cylinder mesher, generate a good quality surface mesh and specify each cylinder as an individual boundary.

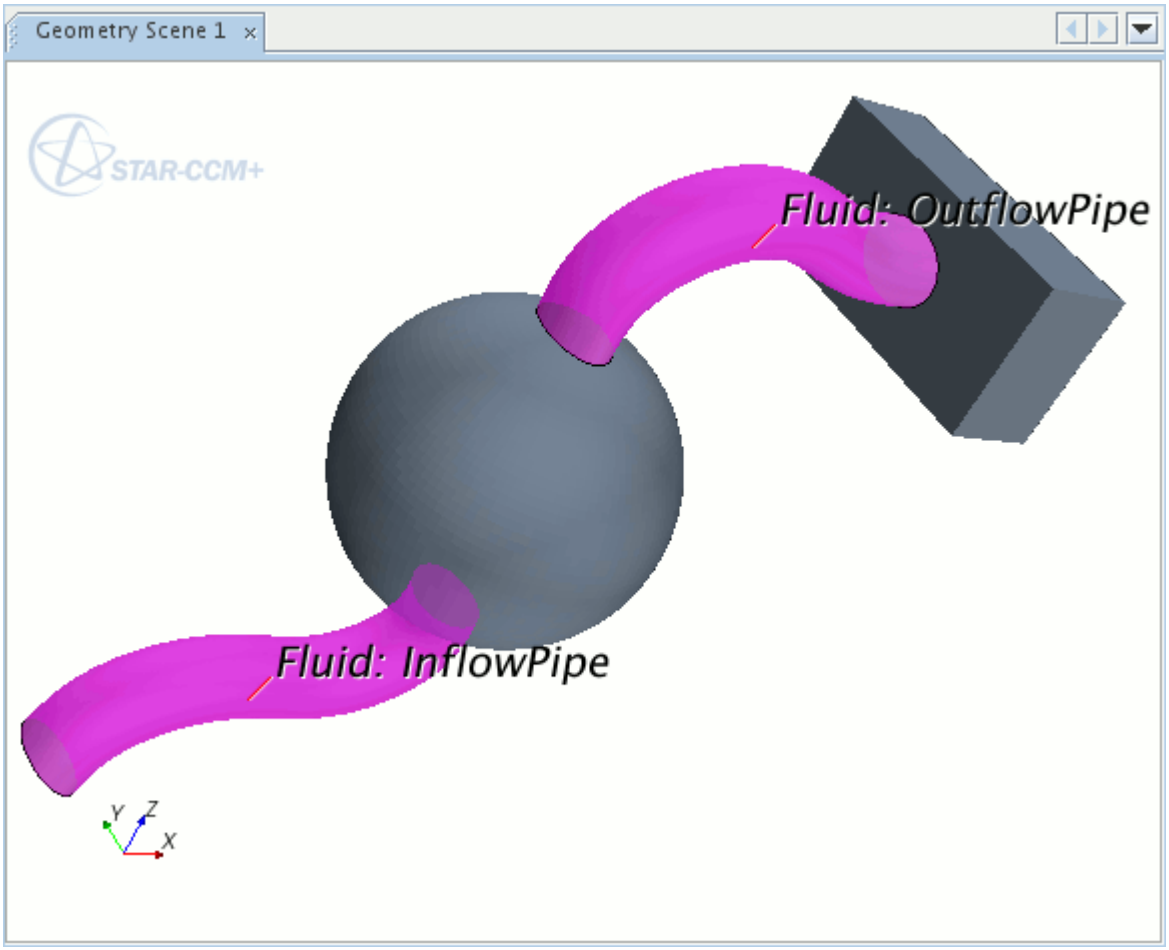
If you do not prepare the simulation in this manner, the generalized cylinder mesher can generate a poor quality volume mesh, or fail. See Generalized Cylinder Mesher Guidelines.

The geometry that is shown in the example below contains two cylindrical pipes that are good candidates for generalized cylinder meshing. However, in this case, the geometry is defined as a single boundary, so the generalized cylinder mesher would not recognize the pipe sections.



So that the generalized cylinder mesher can recognize the pipes, they must be separated into individual boundaries by using one or more of the splitting tools available in STAR-CCM+.

The pipe sections are now defined as separate boundaries, and the generalized cylinder detection tool can be used to assess whether they can be meshed as generalized cylinders.



Currently, the generalized cylinder mesher does not automatically recognize cylindrical elements whose boundaries are interfaces. To use interfaces with a generalized cylinder mesh, generate the volume mesh first and then create interfaces.

Interfaces that exist on the cylindrical boundaries of generalized cylinder mesher regions cannot produce a conformal mesh with regions that are meshed using other volume meshers.

However, the generalized cylinder mesher forms a conformal interface automatically with any region which fills the hollow portion of the cylinder.

See Also:

Surface Meshers Overview

Splitting Boundaries

Detecting Cylinders Automatically

Generalized Cylinder Mesher Guidelines

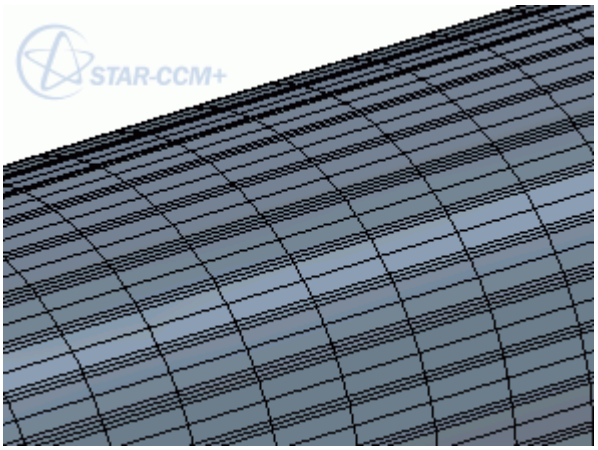
If you encounter problems when identifying cylindrical boundaries or generating a volume mesh, follow these guidelines.

Improve the Surface Definition

The generalized cylinder mesher requires a good surface definition in order to identify cylindrical boundaries. To improve surface definition before using the generalized cylinder mesher, reduce the surface size. If the meshers are still unable to produce a volume mesh, the underlying geometry is unsuitable for cylinder meshing.

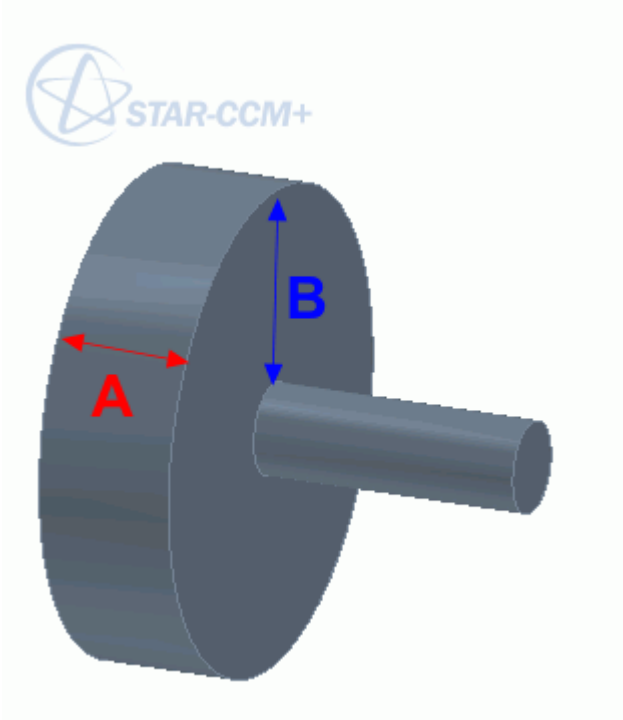
Set the Same Surface Size on all Boundaries

To produce a good quality surface mesh, the vertex density on the inlet and outlet boundaries must match the vertex density on the feature curves. If the Custom Surface Size property is activated on the inlet or outlet boundaries and there is a difference in surface size from the inlet or outlet boundaries to the cylinder boundaries, poor quality faces are created. To resolve this issue, either deactivate the custom surface size property on all boundaries, or set the custom surface size setting to the same value for each boundary.



Where the Radius of the Cylinder is not Constant, Make Sure that the Cylinder Length is at Least Three Times the Change in Radius

If the cylinder length is not sufficient, the generalized cylinder mesher can fail. If it is not possible to adapt the geometry appropriately, deactivate the generalized cylinder mesher for thin cylinders. For example, in the following geometry, the length of distance A is not three times the length of distance B. To mesh this geometry successfully, either increase the length of A, or deactivate the generalized cylinder mesher in the thin cylinder.



Make Sure That Each Cylindrical Boundary is a Complete Cylinder

If more than one boundary represents a cylinder, combine the boundaries to make a complete cylinder.

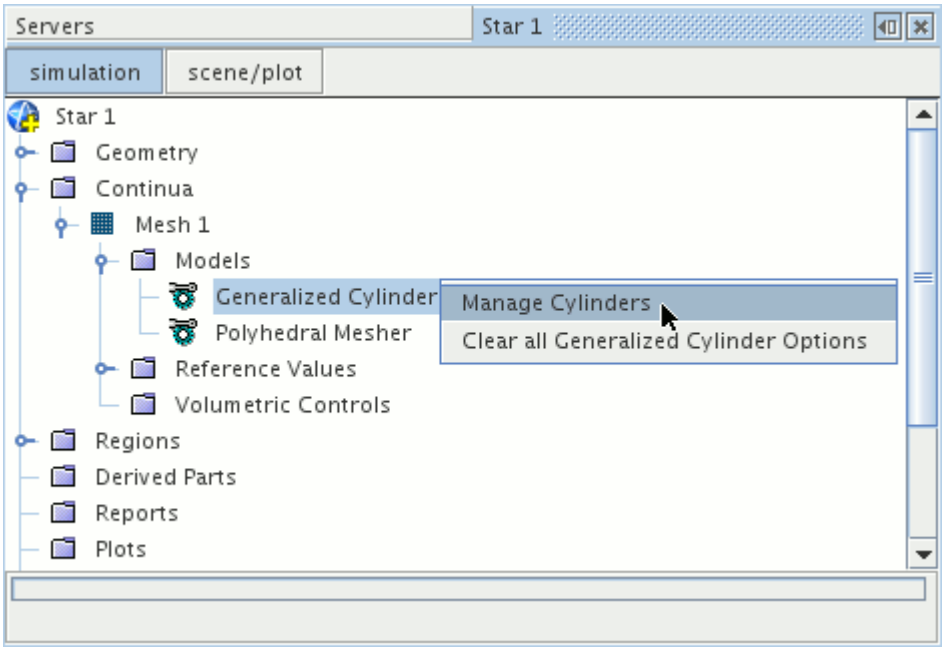
See Also:

Generalized Cylinder Mesher

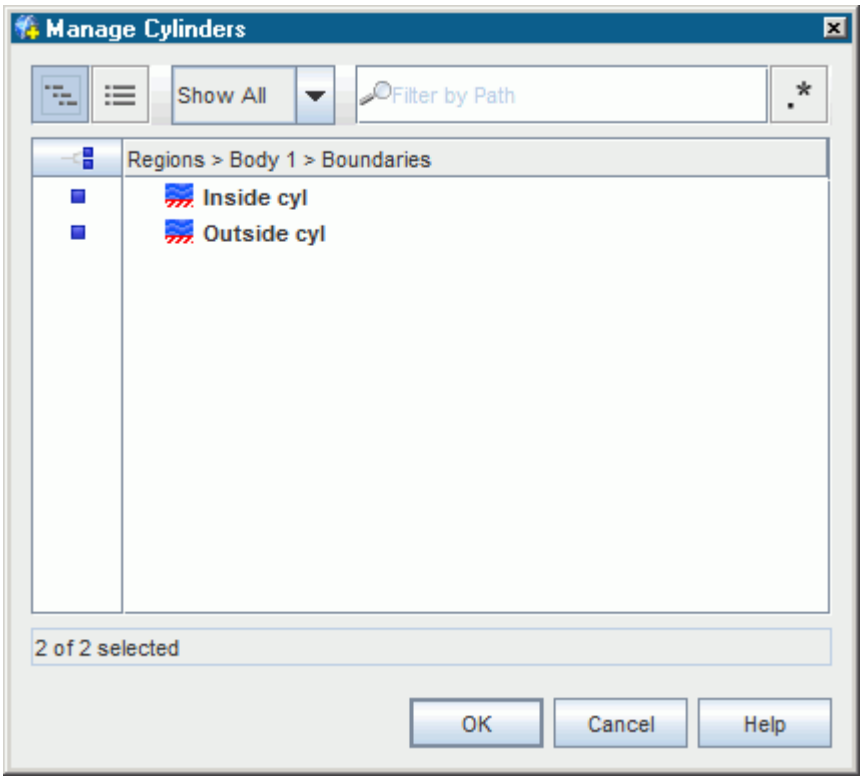
Detecting Cylinders Automatically

The generalized cylinder meshing model detects automatically which boundaries in the simulation are suitable for generalized cylinder meshing.

To detect cylindrical boundaries, first ensure that the generalized cylinder model has been selected, then right-click the **Models** > **Generalized Cylinder** node in the mesh continuum and select **Manage Cylinders**.

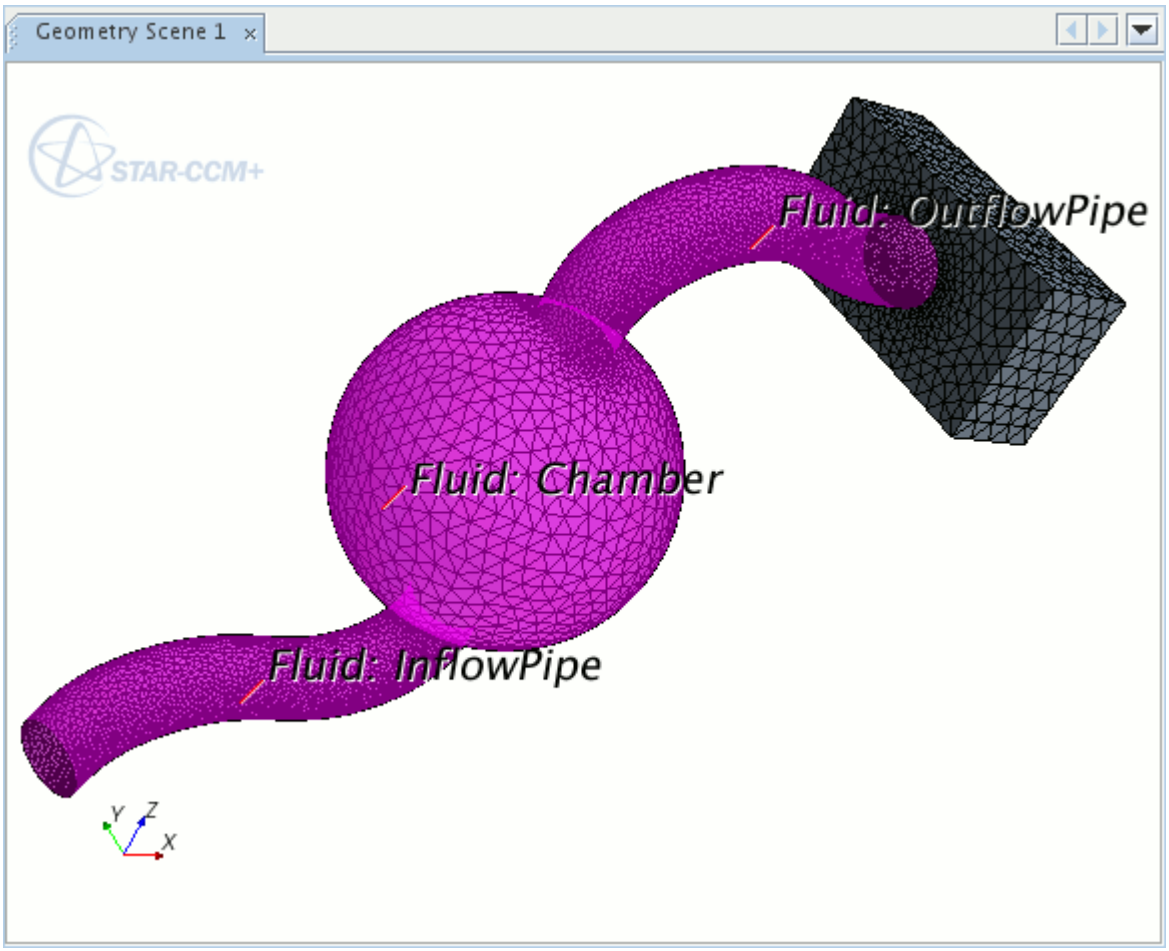


Selecting this option causes STAR-CCM+ to search for any boundaries that are roughly cylindrical. When the search is complete, boundaries that meet the criteria for generalized cylinder meshing are listed in the *Manage Cylinders* dialog. The dialog is used to activate or deactivate the generalized cylinder mesher on the boundaries that are found during the search, by selecting or deselecting the appropriate boundary.



By looking at the highlighted boundaries in the active scene, you can decide which to mesh as generalized cylinders, then activate or deactivate the corresponding boundaries in the *Find Cylinders* dialogand click **OK**.

The following screenshot shows an occasion where the generalized cylinder search found a boundary that is not intended for generalized cylinder meshing, Chamber. Since it would be desirable to generate a generalized cylinder mesh on the InflowPipe and OutflowPipe boundaries, but not on the Chamber boundary, the mesher can be deactivated for the Chamber boundary.

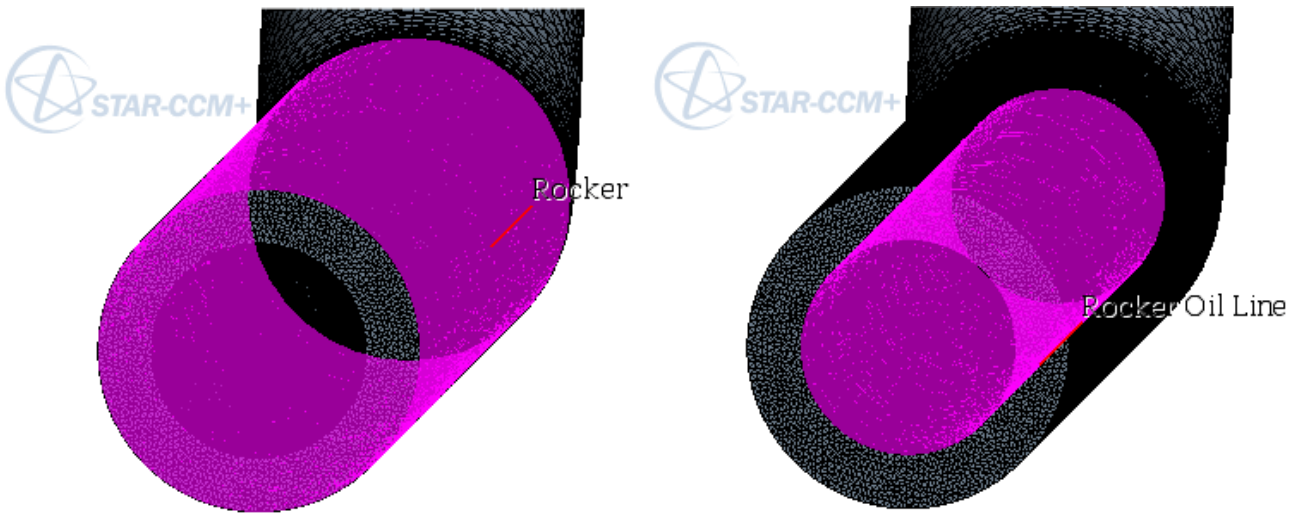


When changes to the *Find Cylinders* dialog are complete, clicking **OK** activates the generalized cylinder mesher for the selected boundaries. This action sets the generalized cylinder extrusion type for each boundary to **Constant**, and the extrusion parameters for the mesh then have the default values. To select an alternative extrusion type or customize the extrusion parameters, changes must be made for each boundary individually. See Customizing Generalized Cylinder Conditions and Values.

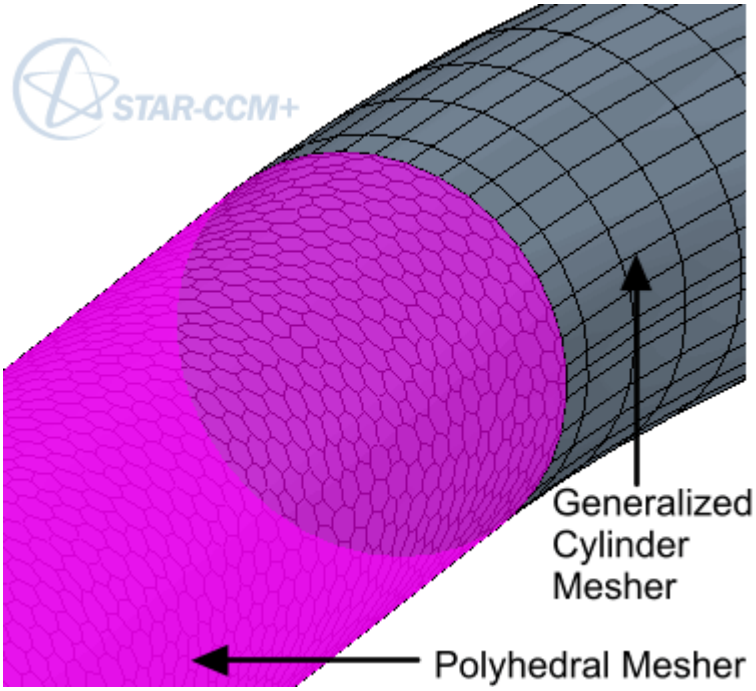
The **Manage Cylinders** option can also be used to remove existing generalized cylinder settings from specific boundaries by deactivating the checkboxes in the *Find Cylinders* dialog for the desired boundaries, and clicking **OK**. Alternatively generalized cylinder settings can be removed from all boundaries by selecting **Clear all Generalized Cylinder Options**.

Detecting Cylindrical Boundaries on Hollow Cylinders

When using the generalized cylinder mesher with hollow cylinders, all external and internal cylindrical boundaries must be selected for the generalized cylinder mesher.



If only one internal or external boundary is selected on a hollow cylinder for the generalized cylinder mesher, the cylinder is meshed using the polyhedral volume mesher.



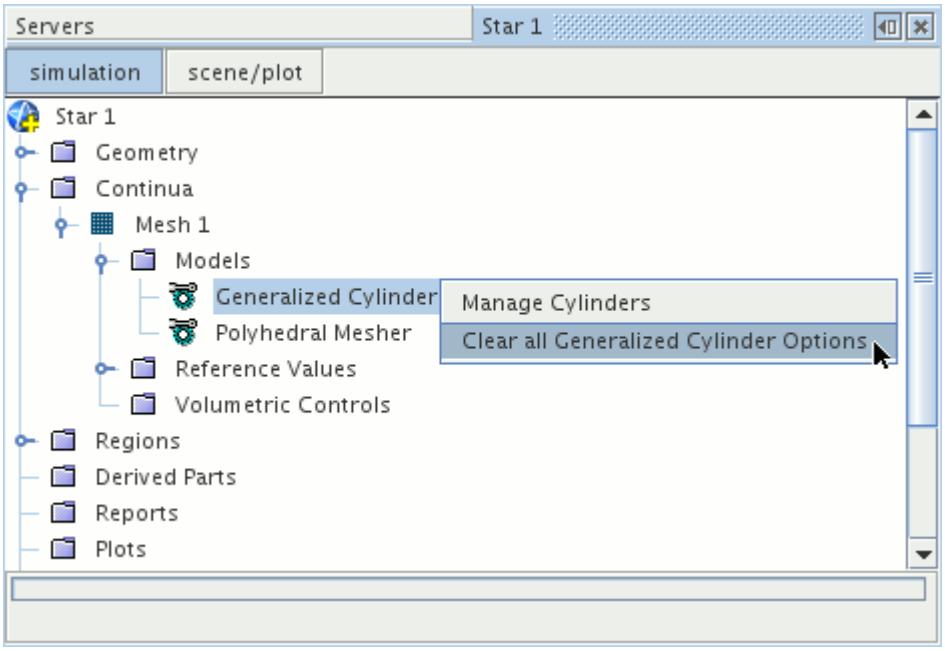
See Also:

Preparing Boundaries for the Generalized Cylinder Mesher

Clearing All Generalized Cylinder Options

You can remove the generalized cylinder settings from all boundaries in the model by right-clicking on the **Models** > **Generalized Cylinder** node in the mesh continuum and selecting **Clear all Generalized Cylinder Options**.

This option resets the generalized cylinder extrusion type mesh condition for all boundaries to **None**, deactivating the generalized cylinder mesher.



Alternatively the generalized cylinder mesher can be deactivated for individual boundaries by using the **Manage Cylinders** option, or by changing the properties of the generalized cylinder extrusion type mesh condition for each boundary.

Customizing Generalized Cylinder Conditions and Values

Properties for the generalized cylinder meshing model are set at the boundary level - no mesh continuum, region, or interface properties are available.

The generalized cylinder mesh is activated or deactivated through the generalized cylinder extrusion type mesh condition. Once activated, the mesh can be controlled by setting the generalized cylinder parameters.

Generalized Cylinder Extrusion Type

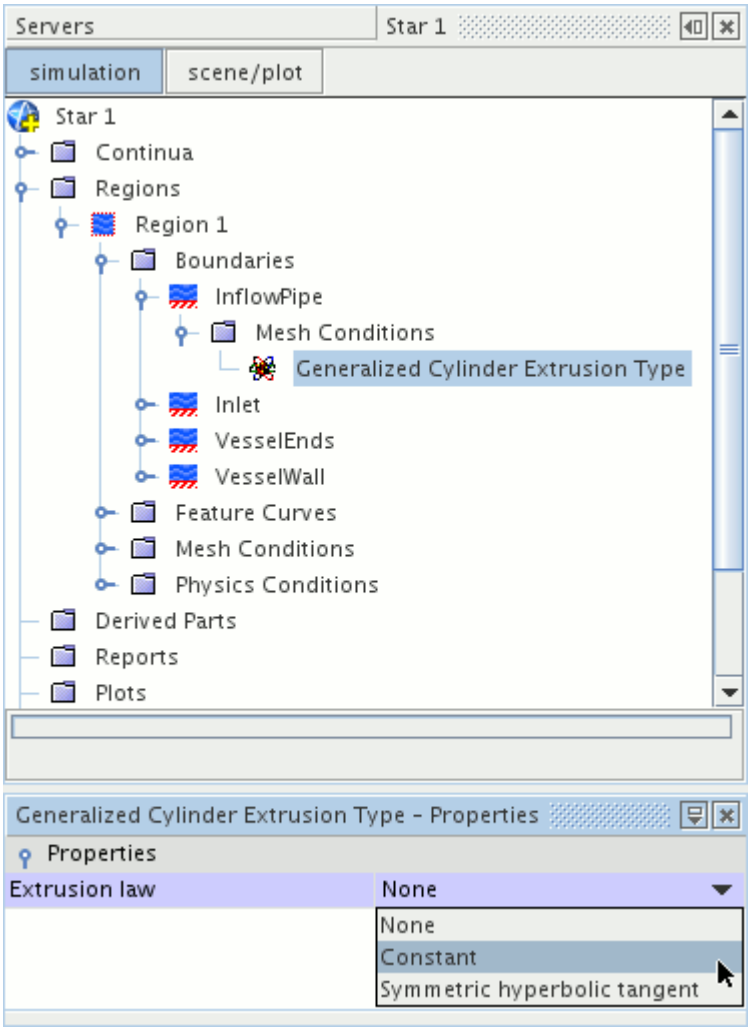
The generalized cylinder extrusion type determines if a generalized cylinder mesh is generated on the boundary, and if so offers a choice for the type of extruded mesh. After the generalized cylinder meshing model is selected, every boundary then has a generalized cylinder extrusion type node.

The *Extrusion Law* property sets the extrusion type, and the options available are:

- none

- constant
- symmetric hyperbolic tangent.

When the generalized cylinder detection tool detects a cylinder, the generalized cylinder mesher is activated for that boundary by setting the *Extrusion Law* to **Constant**.



Generalized Cylinder Extrusion Type Properties

None	The default setting for the extrusion law is None. When this option is selected a generalized cylinder mesh is not generated on the boundary, and a polyhedral mesh is generated instead.
Constant	This option results in a mesh containing cells that are a constant length. The properties of the mesh are controlled through the generalized cylinder parameters node, which is added to the boundary under the **Mesh Values** node.
Symmetric Hyperbolic Tangent	Selecting **Symmetric Hyperbolic Tangent** allows a mesh to be generated that has shorter cells at the start and end of the cylinder, and longer cells in the middle. The properties of the mesh are controlled

	through the generalized cylinder parameters node, which is added to the boundary under the **Mesh Values** node.

Generalized Cylinder Parameters

The generalized cylinder parameters determine the characteristics of the mesh that is generated with the generalized cylinder mesher. This node is activated if the *Extrusion Law* property of the generalized cylinder extrusion type node is set to either **Constant** or **Symmetric Hyperbolic Tangent**. Depending on the selections that have been made, the properties available are:

- number of layers
- stretching.

Generalized Cylinder Parameters Properties

Number of Layers	The Number of Layers property determines how many cell layers are generated along the length of the extrusion. Only integer values can be used to set the number of layers, and the default value is 20. Increasing the number of layers can improve the mesh that is generated on cylindrical elements that have a higher degree of curvature.
Stretching Factor	The *Stretching* property determines the relative size of each cell layer along the length of the cylinder when the **Symmetric Hyperbolic Tangent** option has been selected for the extrusion law. The property can be used to generate longer cells in the middle of the cylinder, and shorter cells at either end. Increasing the value causes increase clustering of the cells towards the end of the cylinder. Any value above 1.0 introduces clustering. The default value property is `1.0`, which generates cell layers of equal size. Values of less than 1.0, although valid, have no effect on the mesh produced.

Volume Mesher Controls and Values

This section details the common controls and values for all volume meshers. Controls that are specific to one mesher are detailed in the properties for that section.

Common Expert Properties

The following expert properties are available for all volume meshers (excluding optional volume meshing models) except where specified:

Choose Field Function Refinement Table (PBM) / Choose a Mesh Size Table (RBM)	Specifies the table for refinement of the mesh using field functions, which are based upon an existing solution. This option is not available for the thin mesher.

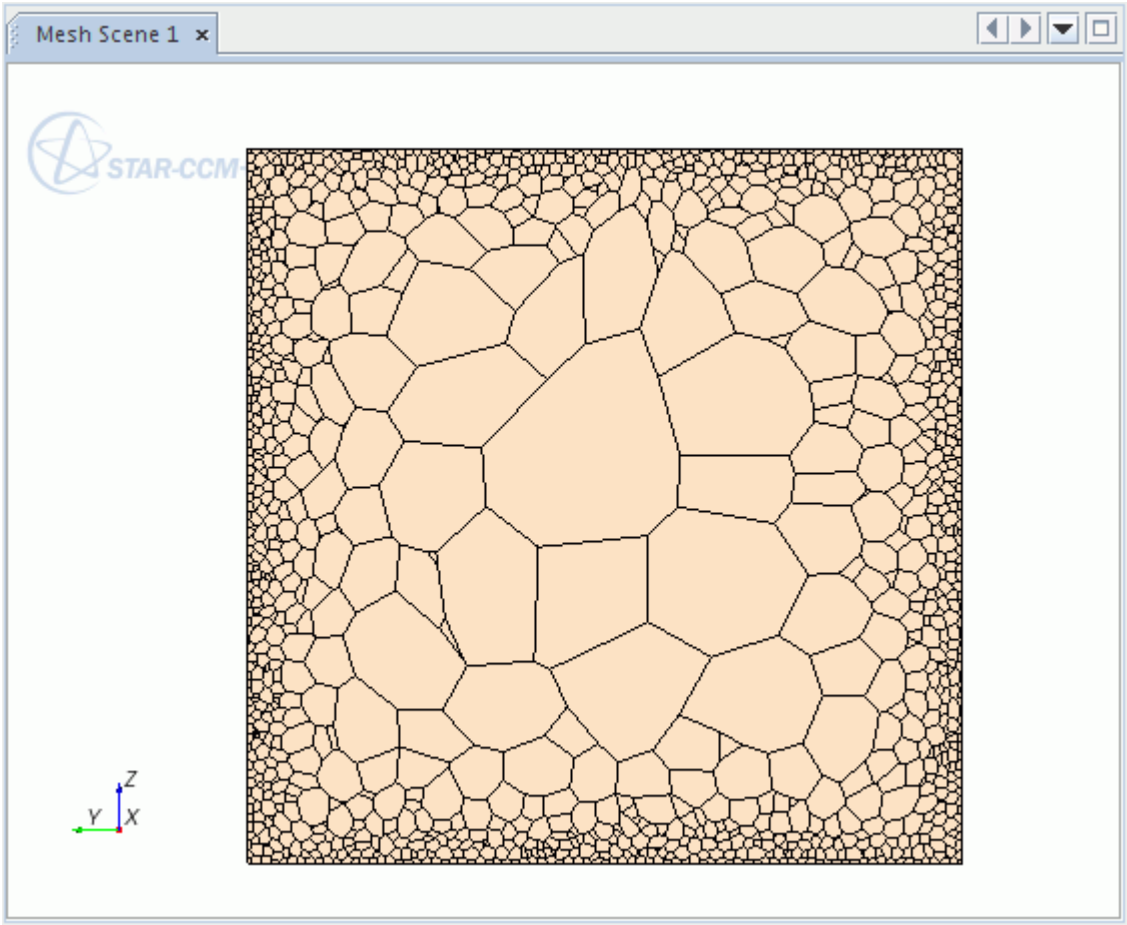
Run Optimizer	Runs a vertex-based optimizer on the assembled mesh for any combination of meshers, resulting in a higher quality mesh. The optimizer moves internal vertices in the volume mesh to improve the mesh metrics. See Mesh Quality. Although this property incurs an additional CPU overhead, you do not generally deactivate it.
Optimization Cycles	Specifies the number of optimization cycles that the mesher runs on the tetrahedral mesh (in the case of a polyhedral mesh, on the initial tetrahedral mesh before generating polyhedral cells). To improve mesh quality, use this property with the *Quality Threshold*, but with increased meshing time. The recommended range is between 1 (lowest quality) and 8 (highest quality). This option is not available for the trimmed cell mesher (PBM)/ trimmer (RBM).
Quality Threshold	Specifies the lowest allowable mesh quality; where 1 is the best mesh that the mesher can generate. Use this property with the *Optimization Cycles* to improve mesh quality, but with increased meshing time. The recommended range is between 0.4 (lowest quality) and 1 (highest quality). This option is not available for the trimmed cell mesher (PBM)/ trimmer (RBM).

When using either the tetrahedral, polyhedral, or advancing layer meshing models, the following expert property is also available:

Enable Growth Rate (PBM) **Enable Mesh Expansion Control** (RBM)	Controls the growth rate by setting a limiting factor for the mesh growth and the speed of the growth. When activated, you set the mesh growth rate on the **Maximum Cell Size** (PBM) / **Tet/Poly Expansion Max Cell Size** (RBM) and **Volume Growth Rate** (PBM) / **Tet/Poly Expansion Rate** (RBM) nodes under the **Default Controls** (PBM) / **Reference Values** (RBM) manager.

Volume Growth Rate (Tetrahedral or Polyhedral Mesher)

When using either the tetrahedral or polyhedral meshing models, a volume growth rate can be implemented to control the density of the core mesh, both globally and on a local basis. Below is a screenshot of the plane section through a cube on which the volume growth rate has been used to decrease the density of the core mesh cells, away from the boundaries.



The **Enable Growth Rate** (PBM) / **Enable Mesh Expansion Control** (RBM) option on the **Tetrahedral** or **Polyhedral Mesher** node provides global control over the expansion of the mesh. Activating this option adds the **Maximum Cell Size** (PBM) / **Tet/Poly Expansion Max Cell Size** (RBM) and**Volume Growth Rate** (PBM) / **Tet/Poly Expansion Rate** (RBM) nodes.

Optimization Cycles and Quality Threshold Properties

The Tetrahedral and Polyhedral Meshers provide two expert properties, Optimization Cycles and Quality Threshold, that you can use to improve the volume mesh quality. However, the increase in quality results in longer processing times during mesh generation.

The Polyhedral Mesher generates a mesh by creating an initial tetrahedral mesh, which it then converts to a polyhedral mesh using a process that is called dualization.

The optimization process is carried out on the tetrahedral mesh for both meshers. The expert properties can be used to increase the number of optimization cycles and improve the quality of the tetrahedral mesh. If the mesh is polyhedral, improvements to the underlying tetrahedral mesh have a direct, beneficial effect on the final polyhedral mesh quality.

The Quality Threshold limits the Tetrahedron Mean Ratio, which is defined as:

$$\text{Tetrahedron Mean Ratio} = \frac{12 \times (3v)^{2/3}}{\sum_i l_i^2}$$

where:

- v — Cell Volume
- l — Edge Length

The Optimization Cycles are processes that the mesher performs to improve the quality and suitability of the volume mesh for the geometry. The polyhedral mesher runs these cycles on the underlying tetrahedral mesh, which has a direct effect on the subsequent polyhedral mesh quality.

As a general guide, applying the highest-quality levels can increase the meshing time by a factor of 10, and the cell count by 15%.

Note: Increase the *Optimization Cycles* and *Quality Threshold* properties together. Changing only one of these properties provides little improvement to mesh quality.

To improve the mesh quality:

- Select the **Operations** > **Automated Mesh** > **Meshers** > **Tetrahedral/Polyhedral Mesher** node for Parts Based Meshing or the **Continua** > **Mesh** > **Models** > **Tetrahedral/Polyhedral Mesher** node for Region Based Meshing.
- Set the *Optimization Cycles* and *Quality Threshold* to one of the following recommendations:

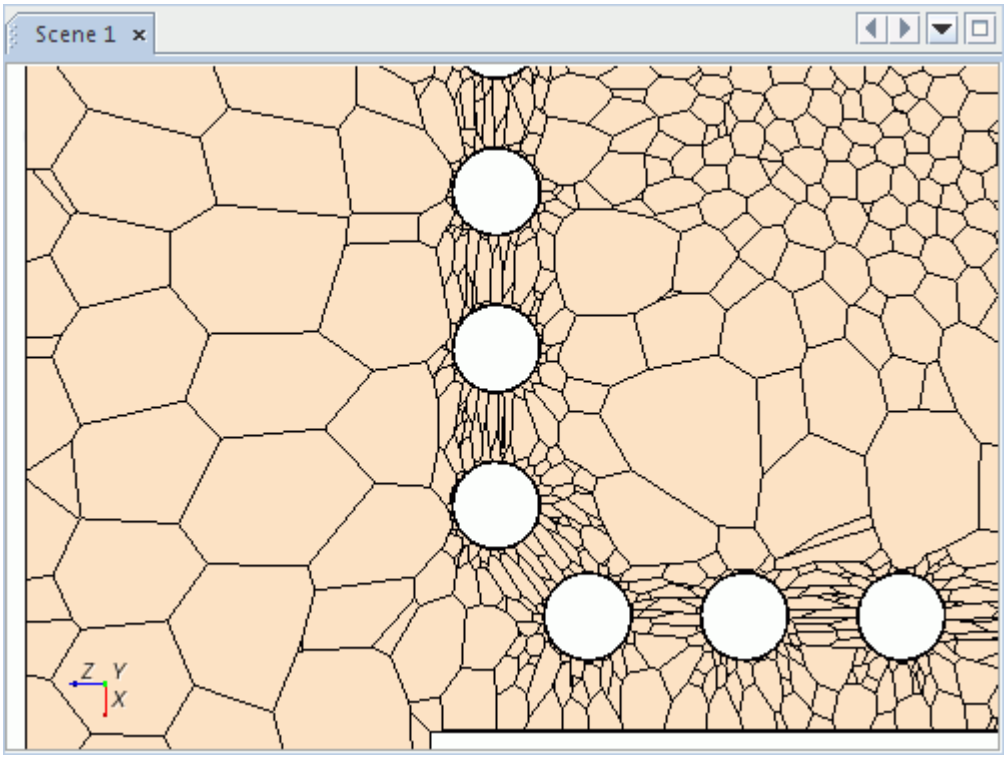
Quality Level	Optimization Cycles	Quality Threshold
Default	1	0.4
1	2	0.6
2	4	0.8
3	8	1.0

- Generate a volume mesh.

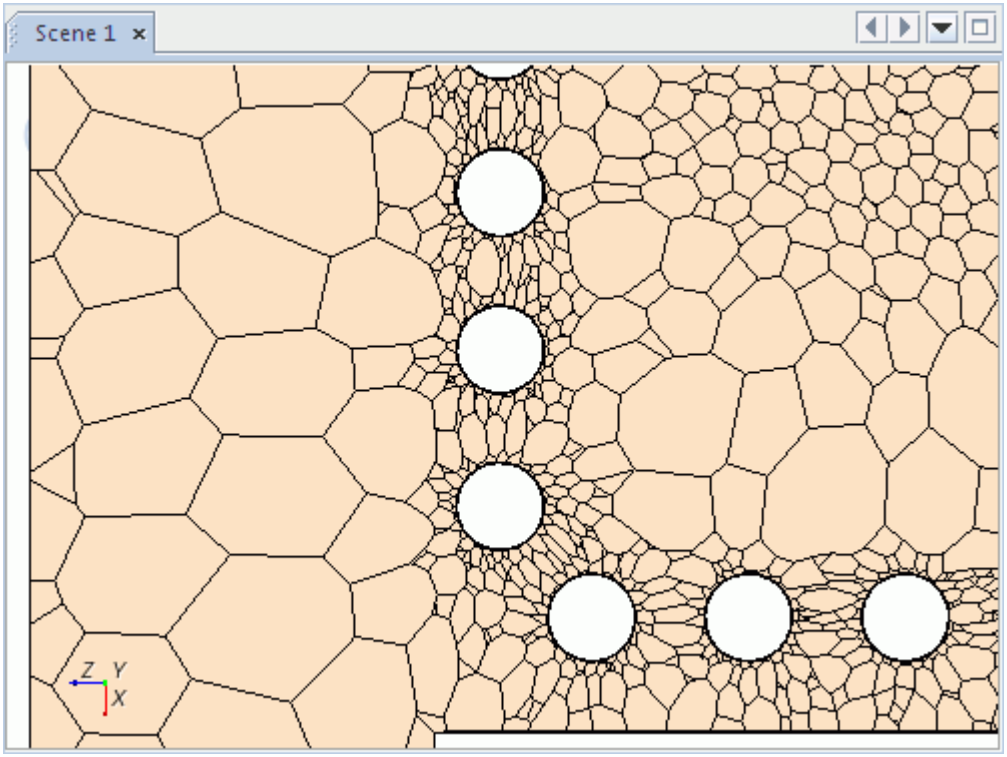
Example of Mesh Quality Improvement

This example shows four volume meshes where the quality of the cells between the circular pegs improves with each level of mesh quality improvement.

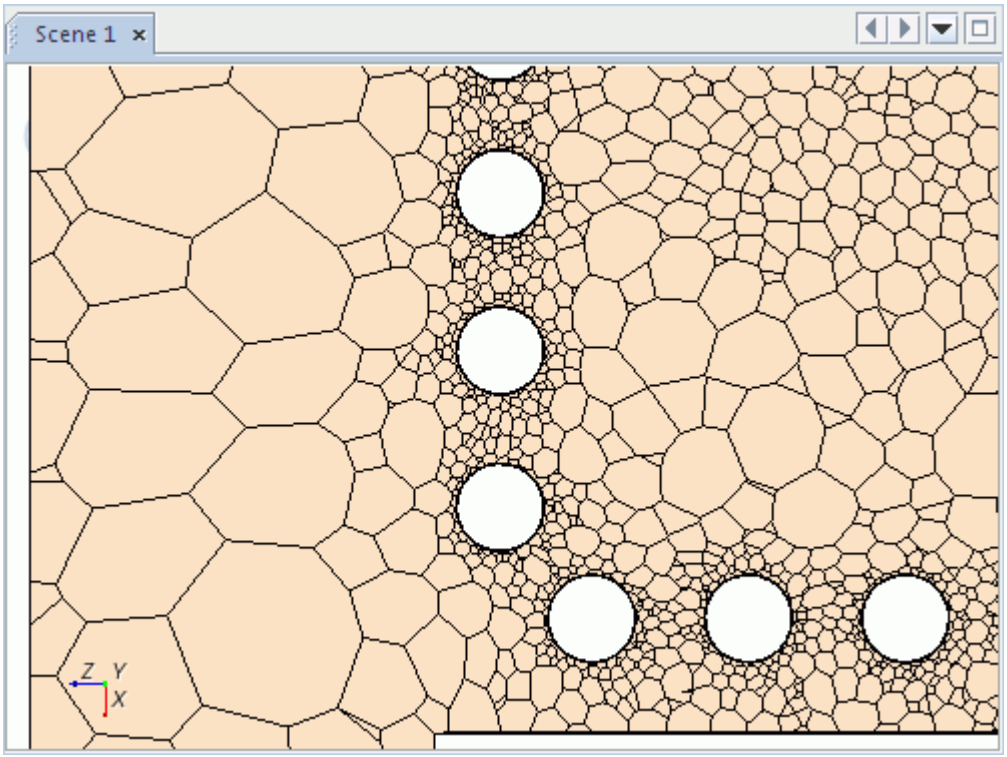
One Optimization Cycle and 0.4 Quality Threshold (Default)



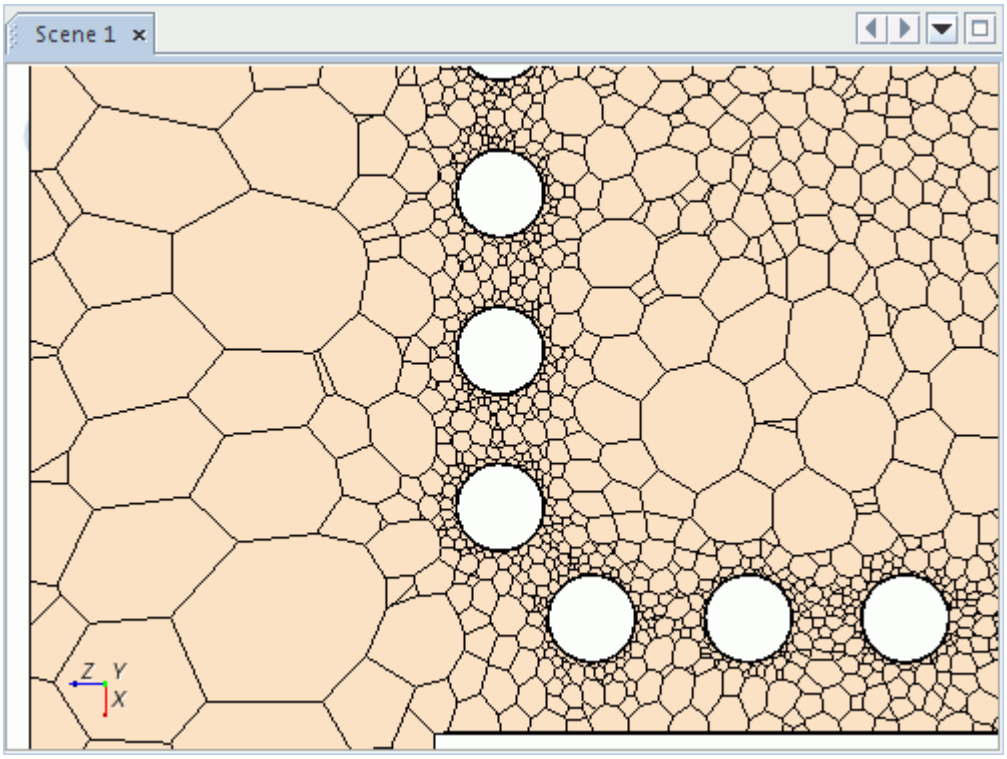
Two Optimization Cycles and 0.6 Quality Threshold



Four Optimization Cycles and 0.8 Quality Threshold

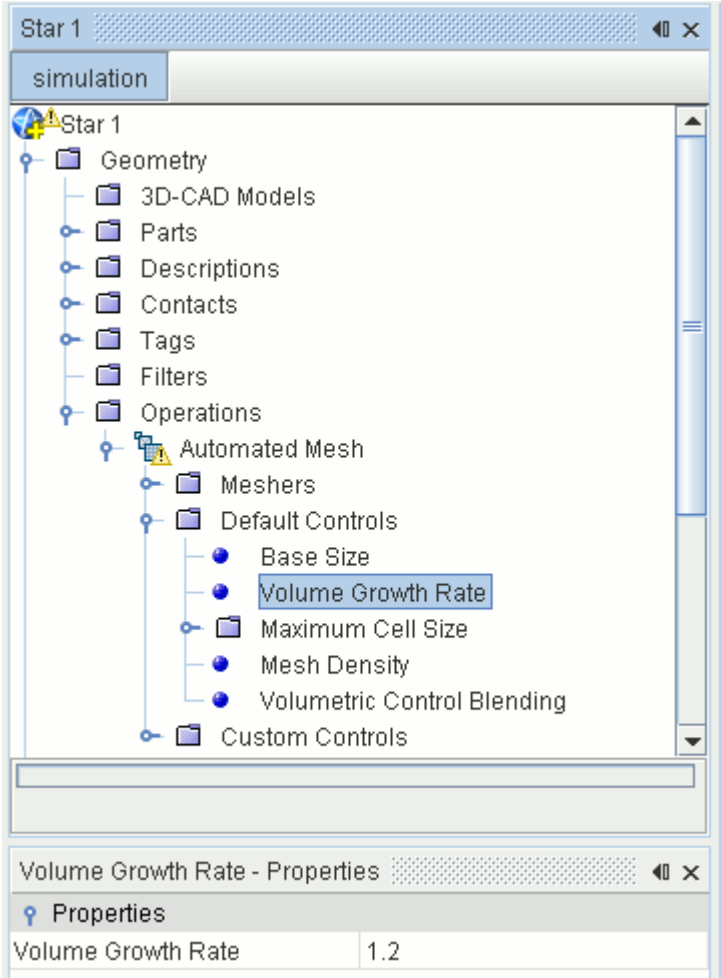


Eight Optimization Cycles and 1.0 Quality Threshold



Volume Growth Rate (PBM) / Tet/Poly Expansion Rate (RBM)

The **Default Controls** > **Volume Growth Rate** node (PBM) or the **Reference Values** > **Tet/Poly Expansion Rate** node (RBM) controls the growth rate of the volume mesh away from boundaries.



Volume Growth Rate Properties

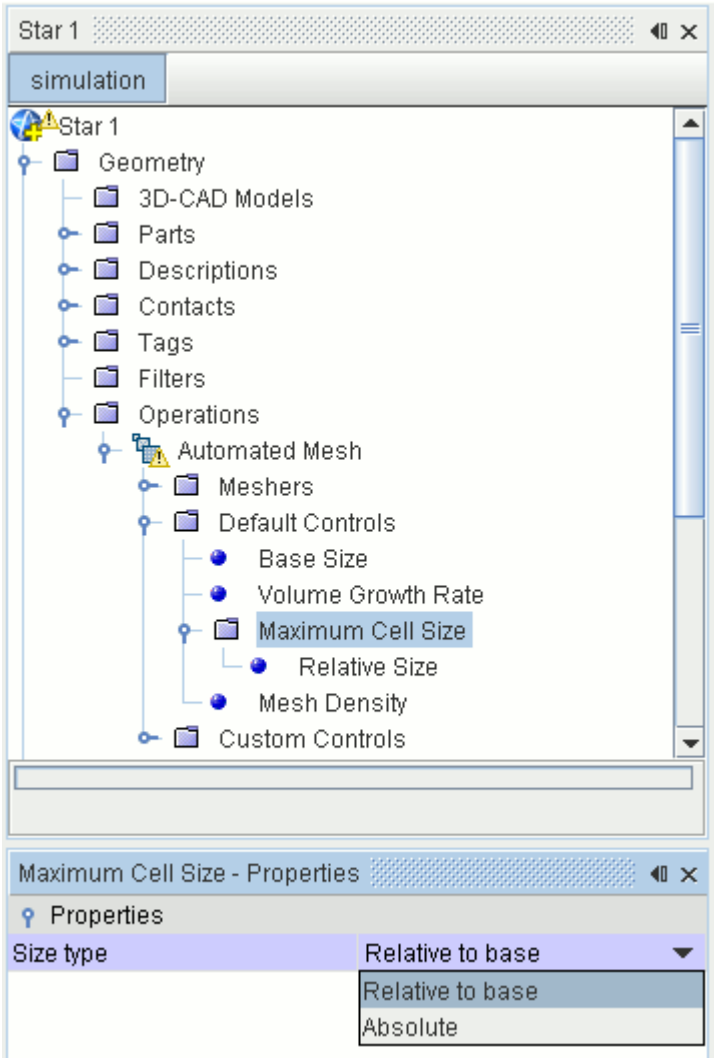
Volume Growth Rate (PBM) **Tet/Poly Expansion Rate** (RBM)	This property defines the growth of the mesh size. This value should be set to between 1.0 and 2.0. Expansion rate of 1.0 is associated with a mesh that has uniform density throughout the assigned region. Increasing the growth factor results in larger mesh sizes in the transition from boundary size to the size away from the boundary. That is, increasing the value closer to 2.0 results in mesh sizes that grow quicker with each layer of cells away from the boundary.

Maximum Cell Size (PBM) / Tet/Poly Expansion Max Cell Size (RBM)

The **Maximum Cell Size** (PBM) / **Tet/Poly Expansion Max Cell Size** (RBM) node limits the largest cell size so that the mesh is not too coarse in the center of the domain.

For a polyhedral mesh, the value that you set here is applied to an initial tetrahedral mesh that the polyhedral mesher generates. The tetrahedral mesh is then converted to a polyhedral mesh during a process that is known as dualization. For this reason, the maximum cell size limit does not apply directly to the polyhedral mesh. It indirectly affects the cell size through the intermediate tetrahedral mesh.

You have the option of setting the size type to absolute or relative to the base size in the **Properties** window of the node.



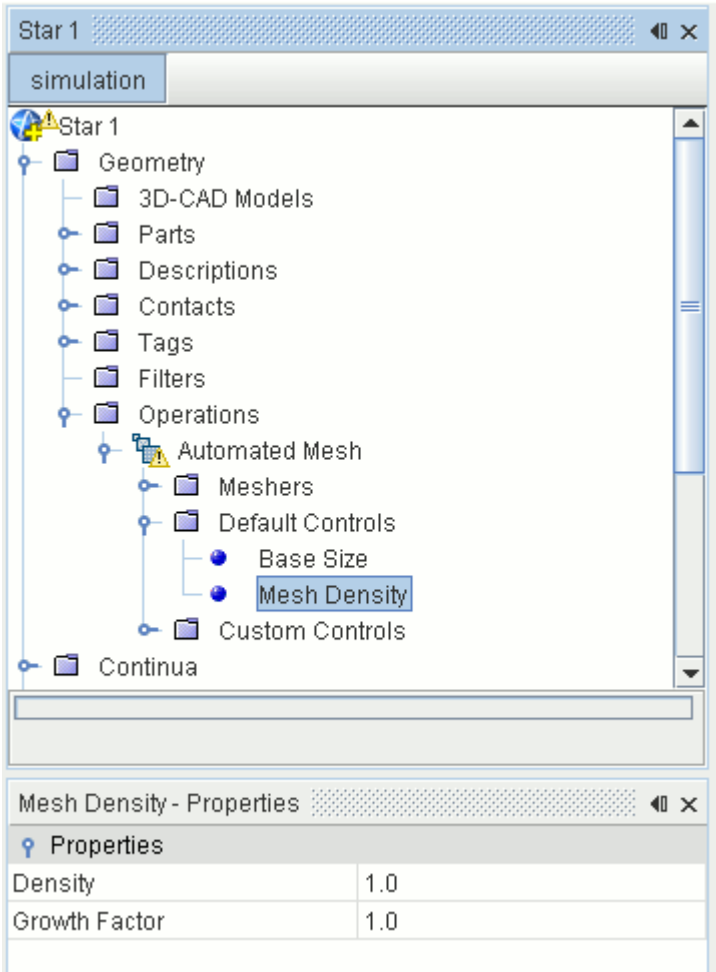
The maximum cell size parameter also controls the maximum allowable cell size in the template refinement scheme when used with the trimmer model.

Maximum Cell Size Properties

Size Type	Selecting the **Maximum Cell Size** node allows the choice of a relative or absolute *Size type* to be made.	
	Relative to Base	Set the maximum cell size relative to the base size.
	Absolute	Set the maximum cell size as absolute value with length units.

Mesh Density

When using the tetrahedral or polyhedral meshing models, a volume growth and/or density factor can be used to increase or decrease the mesh density, both globally and on a local basis. Select the **Default Controls** > **Mesh Density** node (PBM) or the **Reference Values Tet/Poly Density** node (RBM) to access the properties.



Mesh Density Properties

Density	The *Density* value (1.0 by default) can be used to change the overall density of the mesh everywhere in the bulk volume. Increasing the

	value to say 2.0 would approximately double the number of cells that are generated while decreasing it to 0.5 would approximately halve the number of cells.
Growth Factor	The *Growth Factor* can be used to increase or decrease the mesh density of the core mesh by changing the rate at which cells grow from coarse to fine areas. In other words, it affects the grading of the mesh in a localized fashion where finer surface cells exist. Increasing the value above the default of 1.0 would mean that the cell size would grow faster, resulting in fewer cells. Conversely, decreasing the value below 1.0 would make the cell size grow more slowly, resulting in more cells being created.

To demonstrate the effect of each parameter, consider the example of two opposing surfaces that are separated by a gap. If the surface mesh sizes are the same, then the growth factor has no effect and only the density factor could be used to change the volume mesh density. If the surface mesh sizes differ from one surface to the other, then one or both factors could be used to influence the resulting density.

Neither the volume mesh growth nor density factor affects the cells directly next to the starting surface. Neither parameter has priority over the other.

Part Priority

This control is only available when the Automated Mesh operation property *Mesher Execution Mode* is set to **Concurrent**, *Per-Part Meshing* is activated, and the simulation is opened on at least three processors.

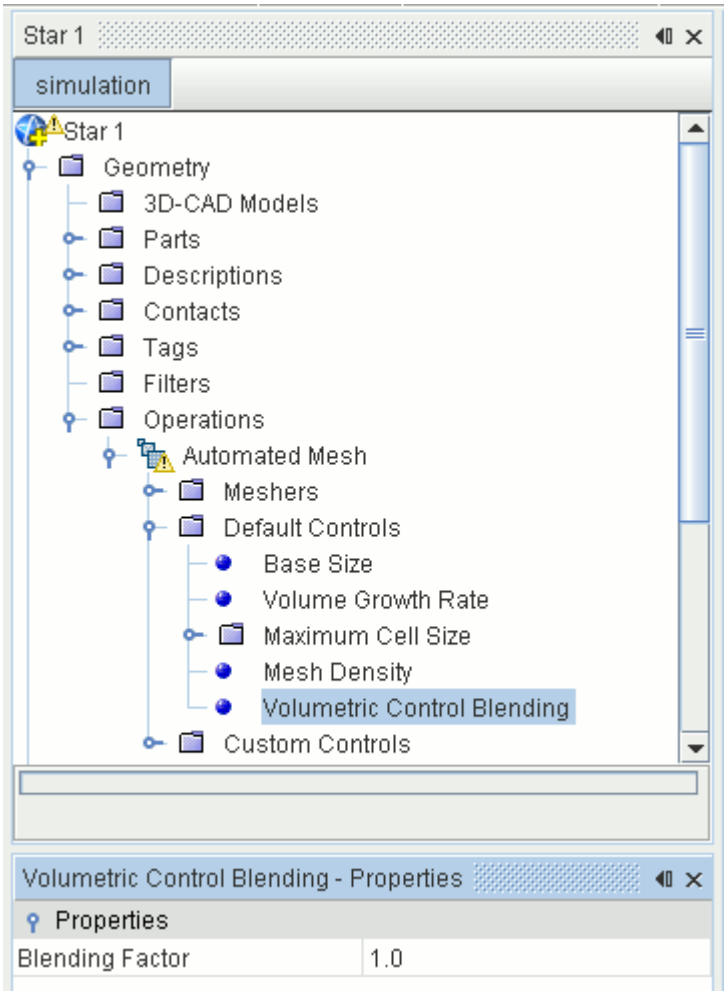
Sort Criteria	Allows you to distinguish which parts are prioritized when meshing concurrently.	
	Default	Parts are sorted in the order of creation.
	Surface Area	Parts are meshed in order from largest to smallest surface area. The surface areas are computed from the root descriptions of each part.
	Volume	Parts are meshed in order from largest to smallest volume. The volumes are computed from the root descriptions of each part. The absolute value of the volume is used, so open and negative volume parts affect the order. The mesh operation prints an error in the Output window with a list of parts that are not closed and manifold.
	Manual	Allows you to categorize parts using Priority Sets. Parts are sorted by the index of the priority set – which you can reorder. All parts within a priority set are sorted in the order they were created. Parts that are not in any priority set have the least priority and are also sorted by the order of creation. This option is useful to prioritize computationally expensive parts. The Output window displays computational times for each part during meshing.

Volumetric Control Blending (PBM) / Tet/Poly Volume Blending (RBM)

The global volume blending factor can be used to control the mesh density transition when volumetric controls overlap or are in close proximity to the surface mesh boundary or interface. This blending factor avoids sharp transitions in mesh density in these areas which could lead to numerical instability during the analysis.

Select the **Default Controls** > **Volumetric Control Blending** node (PBM) or the **Reference Values** > **Tet/Poly Volume Blending** node (RBM) to access the properties.

When using Parts Based Meshing, this node appears in the tree only when at least one volumetric control has been activated.



Volume Control Blending Properties

Blending Factor	The default *Blending Factor* of `1.0` provides a reasonable transition in mesh density between the cell size that is used for the volumetric control and the surface mesh size. Decreasing the value (to say 0.5) provides a smoother transition by making the local mesh size closer to the boundary/interface surface meshing size. Such a value results in fewer cells in the core mesh (assuming that the volumetric control uses a cell size smaller than the boundary/interface triangle edge size).

	Increasing the value (to say 1.5) provides a sharper transition by making the local mesh size closer to the volumetric control mesh size. Such a value results in more cells in the core mesh (again assuming that the volumetric control uses a cell size smaller than the boundary/interface triangle edge size). When the prism mesher is active, the surface size that is seen by the tetrahedral/polyhedral mesher is not the remeshed size, but rather the subsurfacer size from the prism mesher. In cases of sharp corners, this means that the local sizes of the subsurfacer are often bigger than the underlying remeshed surface sizes, and the final tetrahedral/polyhedral transition happens between the volume control size and this subsurfacer size. Note that the volume blending factor has little or no effect on the core mesh if the volumetric control is far away from the surface boundaries. It can also be used to control the mesh density transition between two or more neighboring or overlapping volumetric controls which use different cell sizes.

Mesh Refinement

STAR-CCM+ can refine the mesh by approximating the wake region or using solution-based approach. These features allow you to achieve efficient refinement by reducing the cell size in areas of interest.

This section provides information on the following methods of mesh refinement:

Contents:

Wake Refinement
Field Function Mesh Refinement

Wake Refinement

It is beneficial to refine the mesh downstream of a body as boundary layer separation causes unsteady and turbulent fluid phenomena. In external aerodynamic simulation, this separated region is known as the wake.

You can perform wake refinement using two different approaches:

- Wake refinement using the Parts-Based meshing (PBM) approach
- Wake refinement using the Regions-Based meshing (RBM) approach

The following meshers are supported:

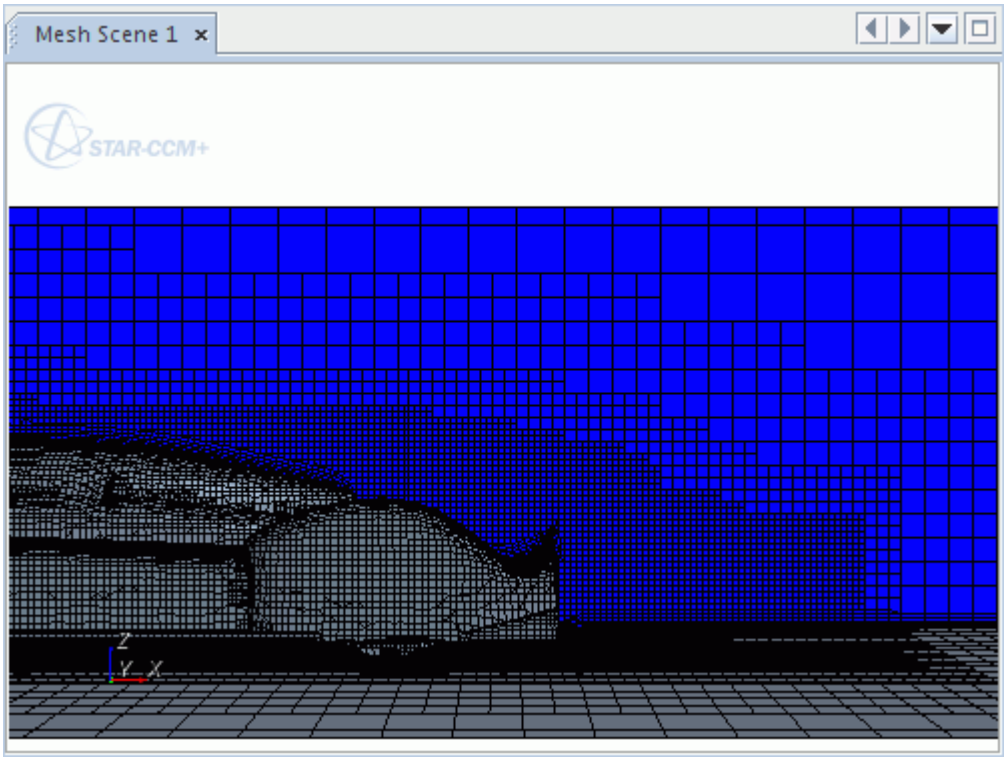
Mesher	PBM approach	RBM approach
Surface Remesher	✓	✗
Polyhedral Mesher	✓	✗
Tetrahedral Mesher	✓	✗
Trimmed Cell Mesher	✓	✓

Note: Wake Refinement does not affect the Prism Layer Mesher. You can still perform wake refinement when using the Prism Layer mesher in addition to a core mesher.

The wake area is formed by sweeping a surface in a user-defined direction and distance. A refinement size and growth parameter determine the cell sizes in the refinement zone.

The main advantage in using the wake refinement option rather than a volumetric control is that you define the refinement zone with a surface. Therefore, if the boundary surface changes, the wake refinement changes also. This characteristic is useful in parametric or batch studies where the same settings are applied to several geometries.

Example of Trimmed Cell Mesh Wake Refinement



Contents:

Refining the Mesh in the Wake
Defining the Shape of the Wake Refinement
Setting the Wake Refinement Size

Refining the Mesh in the Wake

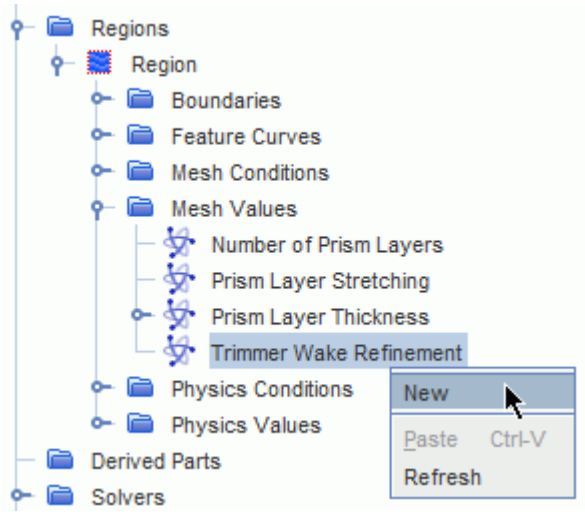
You refine the mesh in the wake to capture the turbulent and transient flow conditions that occur behind a body.

In region-based meshing (RBM), wake refinement is only possible using the Trimmed Mesher. You can use wake refinement with other volume meshers when using parts-based meshing (PBM). See Mesh Refinement in a Wake.

Note: You can create multiple **Wake Refinement Sets**. This facility is useful for marine simulations where the wake broadens downstream of the hull.

To activate the wake refinement option for a region:

- Right-click the **Mesh Values** > **Trimmer Wake Refinement** node and select **New**.



- Select the **Trimmer Wake Refinement** > **Wake Refinement Set 1** node and set the Properties to control the characteristics of the wake refinement

Wake Refinement Set Properties

Relative/Absolute	Specifies whether you define the cell size in the refined region as relative or absolute. The actual size value is set on the child node.
Boundaries/Feature Curves	Specifies the boundaries and feature curves that you use to define the shape of the wake refinement region.
Distance	Specifies the length by which the wake refinement region extends beyond the selected boundaries and feature curves.
Direction	Specifies the direction in which the wake refinement region extends beyond the selected boundaries and feature curves. Enter as vector physical quantity relative to the trimmer coordinate system.
Default Growth Rate	Sets the default growth rate within the wake region (see Setting Wake Refinement Growth Rates).
Boundary Growth Rate	Sets the boundary growth rate within the wake region (see Setting Wake Refinement Growth Rates).
Customize Anisotropic Size	When activated, this property allows you to set a custom cell size for each coordinate direction relative to the trimmer coordinate system.

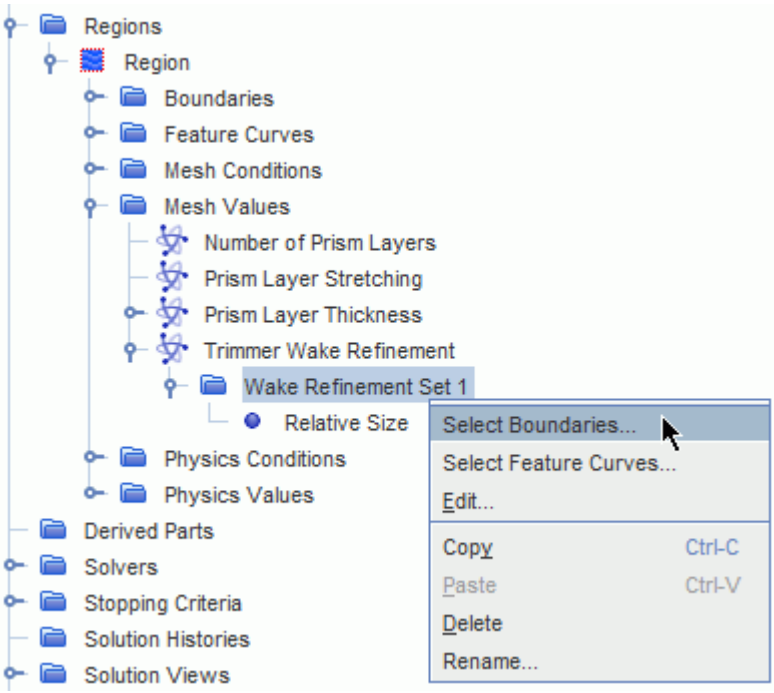
Defining the Shape of the Wake Refinement

The shape of the wake refinement region depends on the shape of the boundary surfaces and feature curves that you define. For the wake refinement to be valid, select at least one boundary.

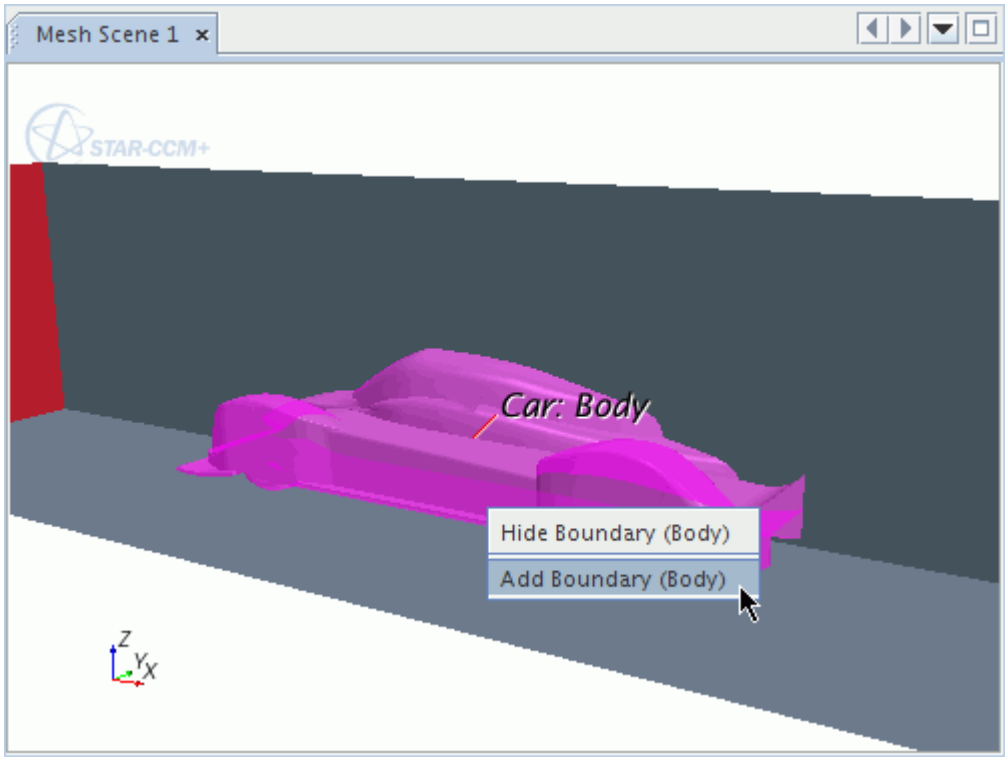
This section covers the interactive selection of boundaries in the *Graphics* window—the procedure for selecting feature curves is the same. Alternatively, you can select boundaries and feature curves using the object selection dialog.

To define the shape of the wake refinement:

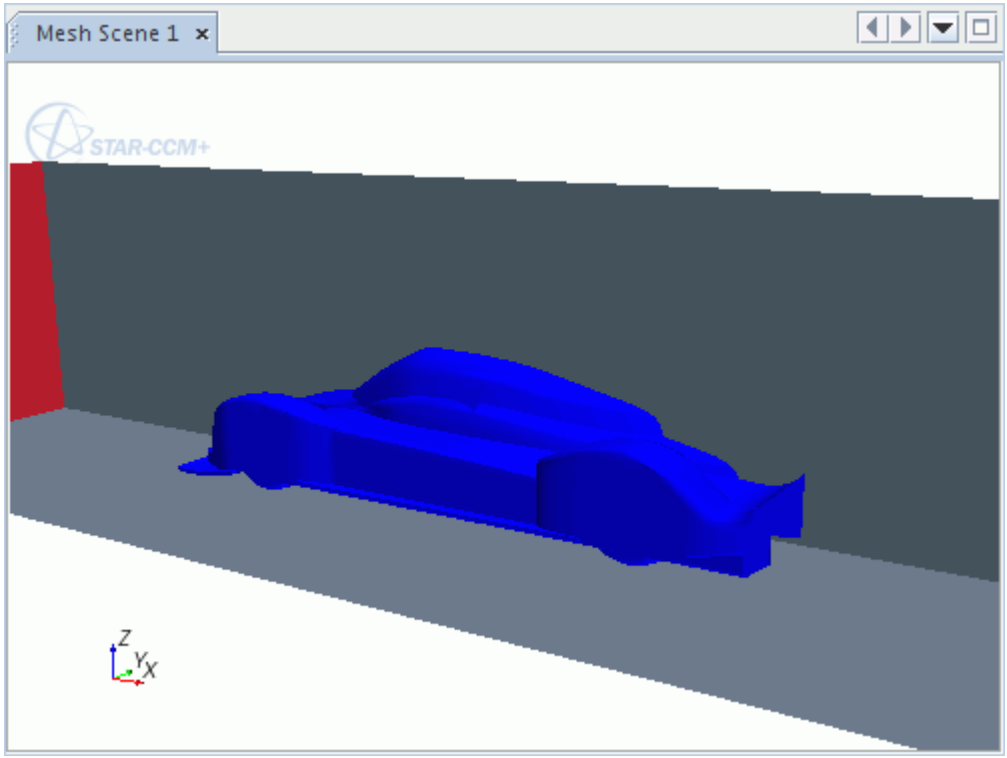
1. Ensure that you have a scene open in the *Graphics* window that includes the region for which you are applying wake refinement.
2. Right-click the **Wake Refinement Set 1** node and select **Select Boundaries**.



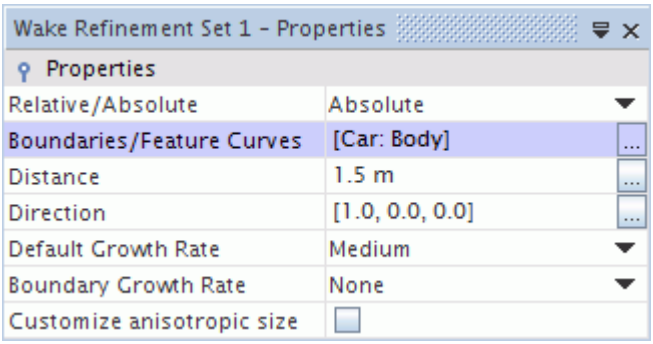
3. In the *Graphics* window, select the required boundaries curves.
4. Right-click one of the selected boundaries and select **Add Boundary** to add the selections to the *Boundary list*.



5. Once you have added a boundary to the list, it appears blue in the *Graphics* window.

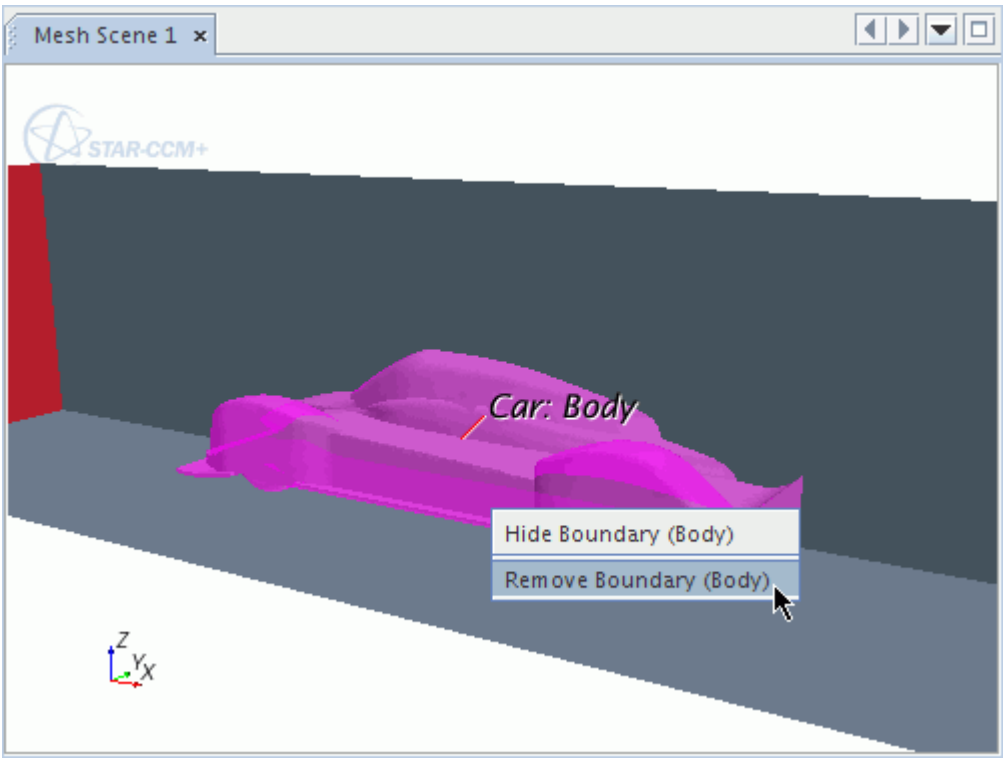


6. Click **Apply** and then **Close** to define the *Boundaries/Feature Curves* property.



To remove boundaries from the *Boundary list*:

7. In the *Graphics* window, right-click an added boundary and select **Remove Boundary**.



8. Click to remove all boundaries from the list.

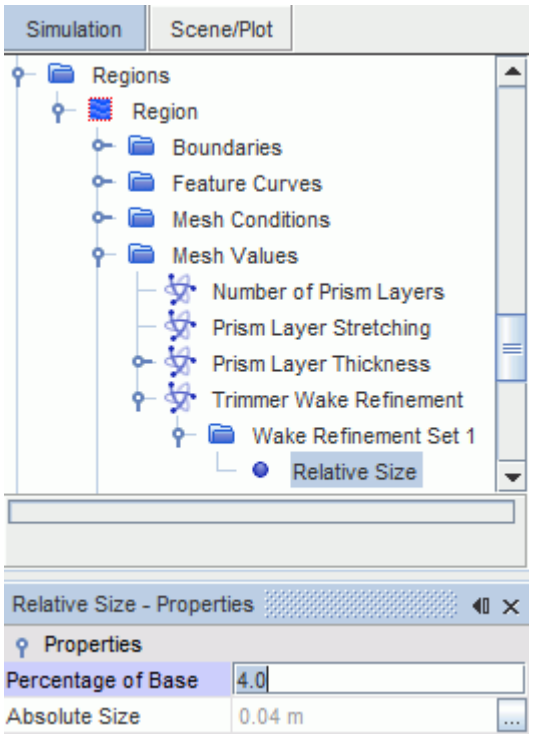
Setting the Wake Refinement Size

The wake refinement size determines the cell size within the refined zone. This refinement size can be either absolute or relative to the base size.

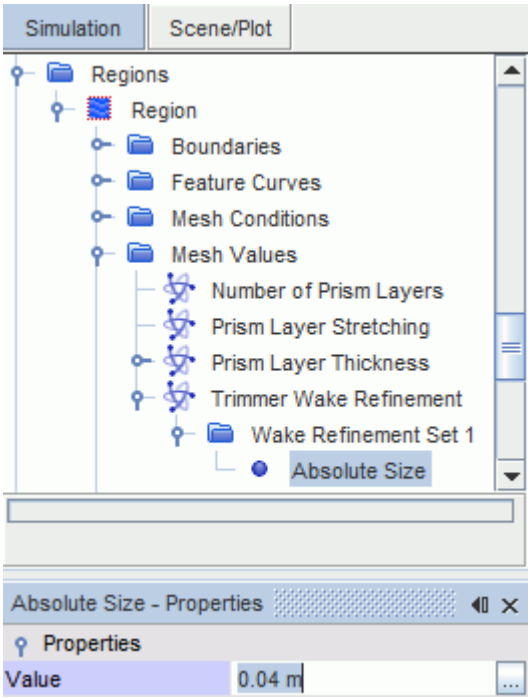
Note: If the wake refinement size is larger than the original cell size, there is no refinement.

To set the wake refinement size:

1. Select the **Wake Refinement Set 1** node and set the *Relative/Absolute* property to define either a relative or absolute size.
2. For a relative cell size:
   - Select the **Wake Refinement Set 1** > **Relative Size** node and set *Percentage of Base*.
     - The *Absolute Size* displays the cell size as a read-only property.



3. For an absolute cell size:
   - Select the **Wake Refinement Set 1** > **Absolute Size** node and set the *Value* property.



Contents:

Setting an Anisotropic Size
Setting Wake Refinement Growth Rates for the Trimmed Mesher

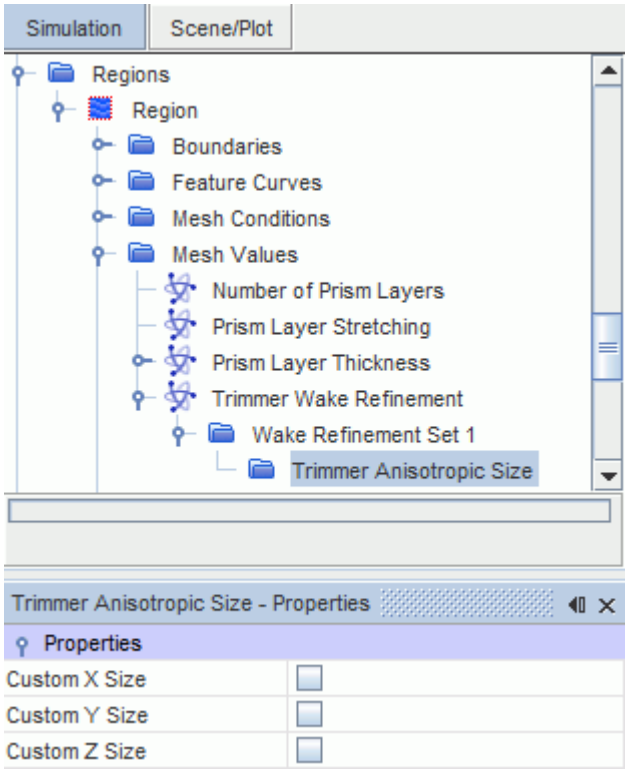
Setting an Anisotropic Size

Anisotropic sizes allow you to define a different cell size in each coordinate direction.

Note: These coordinate directions are relative to the trimmer coordinate system.

To define anisotropic cell sizes:

- Select the **Wake Refinement Set 1** node and activate the *Customize anisotropic size* property.
- Select the **Wake Refinement Set 1** > **Trimmer Anisotropic Size** node and set the properties to define which coordinate directions are anisotropic.



- Select the child nodes and set the relative or absolute cell sizes.

Trimmer Anisotropic Size Properties

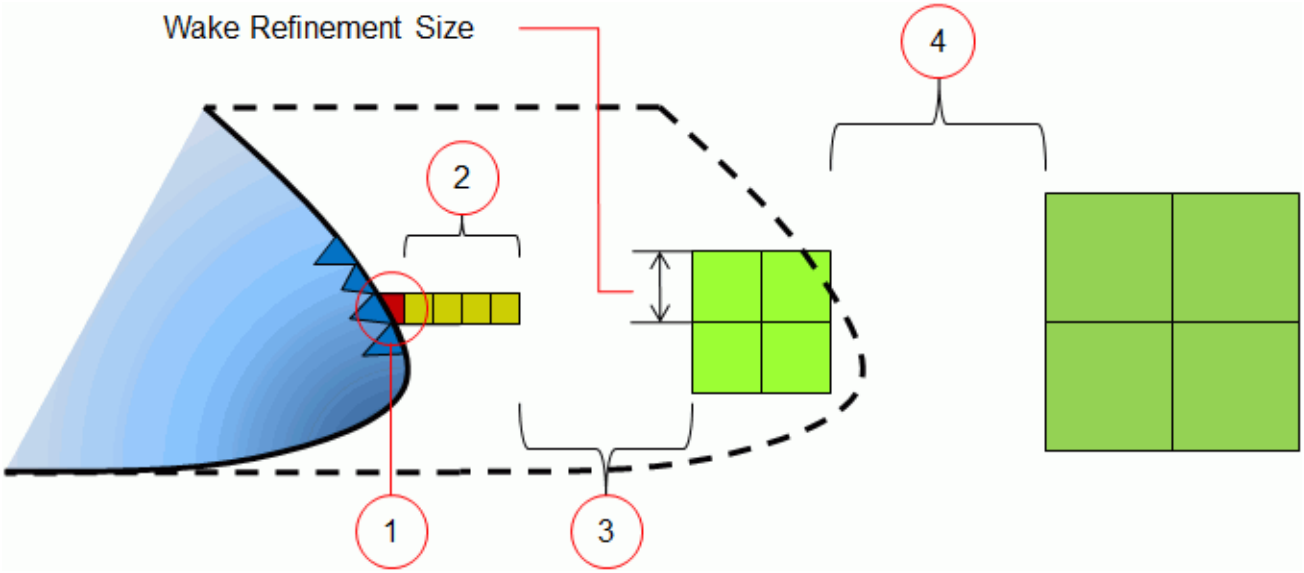
Custom X size	Activate this property to define a relative or absolute X size.
Custom Y size	Activate this property to define a relative or absolute Y size.
Custom Z size	Activate this property to define a relative or absolute Z size.

Setting Wake Refinement Growth Rates for the Trimmed Mesher

The *Default Growth Rate* and *Boundary Growth Rate* control the cell size transition in the wake refinement region.

These options have the same behavior as their global equivalents, the Trimmer Default Growth Rate and the Trimmer Boundary Growth Rate.

The following example illustrates and explains growth rates:



1. Depending on which is smaller, the surface triangulation, custom boundary size, or custom feature curve size determine the initial cell size.
2. The *Wake Region Boundary Growth Rate* maintains the initial cell size for a number of layers that are adjacent the boundary surface or feature curve. The number of layers depends on other meshing properties and the extent of the wake zone, as the mesher aims for a smooth transition in cell sizes. As an example, setting *Boundary Growth Rate* to **Medium** maintains the initial trimmed cell size for at least four layers.
3. The *Wake Region Default Growth Rate* determines the speed of transition from the initial cell size to the wake refinement cell size. Thereafter, the mesher maintains the wake refinement cell size to the limit of the wake refinement zone.
4. Beyond the wake refinement region, the default growth rate determines the speed of transition from the wake refinement cell size to the global cell size.

To set the growth rates:

- Select the **Wake Refinement Set 1** node and set the Boundary Growth Rate and Default Growth Rate properties.

Field Function Mesh Refinement

To achieve successively more accurate solutions, you can refine the mesh with respect to a solution quantity (such as turbulent kinetic energy). In STAR-CCM+, you achieve successive refinement using mesh refinement tables, which are generated from field functions.

Currently, the following meshing models support field function mesh refinement:

- Surface Remesher
- Trimmed Cell Mesher
- Polyhedral Mesher
- Tetrahedral Mesher
- Advancing Layer Mesher

Growth rates determine the transition of cell and face sizes from refined to unrefined zones. The following table details the growth rate properties for each meshing model:

Mesher	Growth Rate
Surface Remesher	Surface Growth Rate
Trimmed	Template Growth Rate
Polyhedral	Tet/Poly Density Growth Factor
Tetrahedral	Tet/Poly Density Growth Factor
Advancing Layer	Tet/Poly Density Growth Factor

If you apply other mesh controls—such as custom mesh values and volumetric controls—to a particular portion of the mesh, the smallest size takes priority.

The following steps outline the procedure for field function mesh refinement:

1. Obtain a solution in the usual manner on the initial mesh.
2. Create a field function to return a mesh size value with respect to a solution quantity.
3. Use this field function to fill an XYZ internal table with mesh sizes. The table properties define the regions and boundaries for which you require mesh refinement.
4. Assign the table as a property of the meshing model.
5. Run the surface mesher or volume mesher to produce the refined mesh.
6. If you require further stages of mesh refinement from new solutions, repopulate the internal table using the extract option.

As this feature provides absolute mesh sizes, you can also coarsen the mesh when the field function returns a mesh size that is larger than the original.

Note: The Advancing Layer mesher also refines near-wall prism layers in addition to the core mesh.

Mesh Size

Meters are the only length unit that field function mesh refinement recognizes—preferred units do not apply. Therefore, ensure that your field function returns the mesh size in meters.

The effect of the mesh size on the refinement is different for the trimmed, polyhedral, and tetrahedral mesher, due to the nature in which these meshes are generated:

- For the trimmed cell mesher, the mesh size is equivalent to the base size in the refinement zone. If the field function returns a value of zero, the mesh is not refined in that zone.
- For the polyhedral mesher, you can approximate the mesh size as follows:

  $$\text{Mesh Size} \approx 1.2 \times \text{Cell Volume}^{(1/3)}$$

  Therefore, if the field function returns a value of `1.2*pow($Volume, 1/3)`, the polyhedral mesher maintains a similar mesh size in that zone.
- For the tetrahedral mesher, you can approximate the mesh size as follows:

  $$\text{Mesh Size} \approx 2 \times \text{Cell Volume}^{(1/3)}$$

Therefore, if the field function returns a value of `2*pow($Volume, 1/3)`, the tetrahedral mesher maintains a similar mesh size in that zone.

Examples of Field Function Mesh Refinement

The following examples show mesh refinement using the trimmed, polyhedral, and tetrahedral mesher. The field function refines the mesh in an area where the turbulence kinetic energy is greater than 4.0 J/kg:

- Trimmed Cell Mesher:

  ```
  ($TurbulentKineticEnergy>4)? 0.1 : 0
  ```

- Polyhedral Mesher:

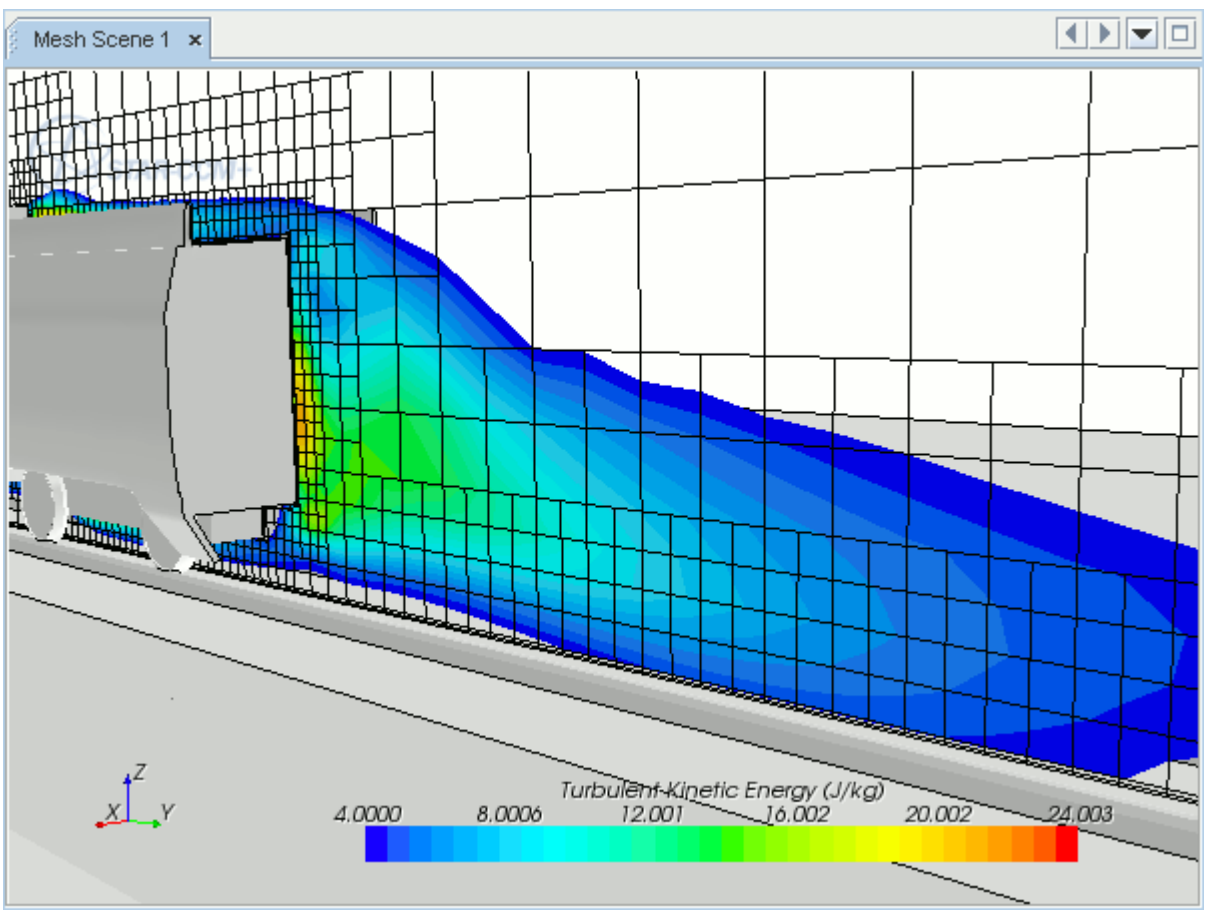
  ```
  ($TurbulentKineticEnergy>4)? 0.1 : (1.2*pow($Volume, 1/3))
  ```

- Tetrahedral Mesher:

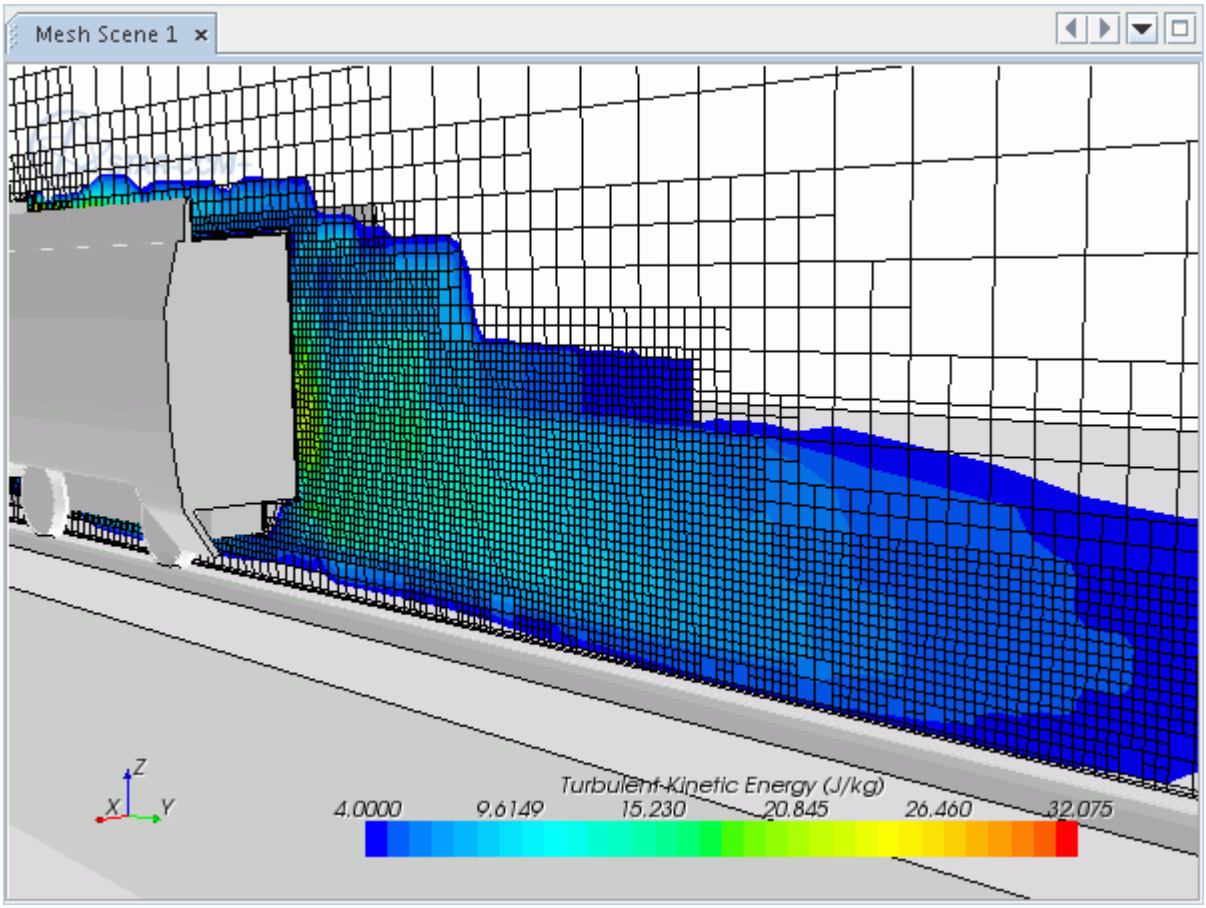
  ```
  ($TurbulentKineticEnergy>4)? 0.1 : (2*pow($Volume, 1/3))
  ```

Trimmed Cell Mesher

Before mesh refinement:

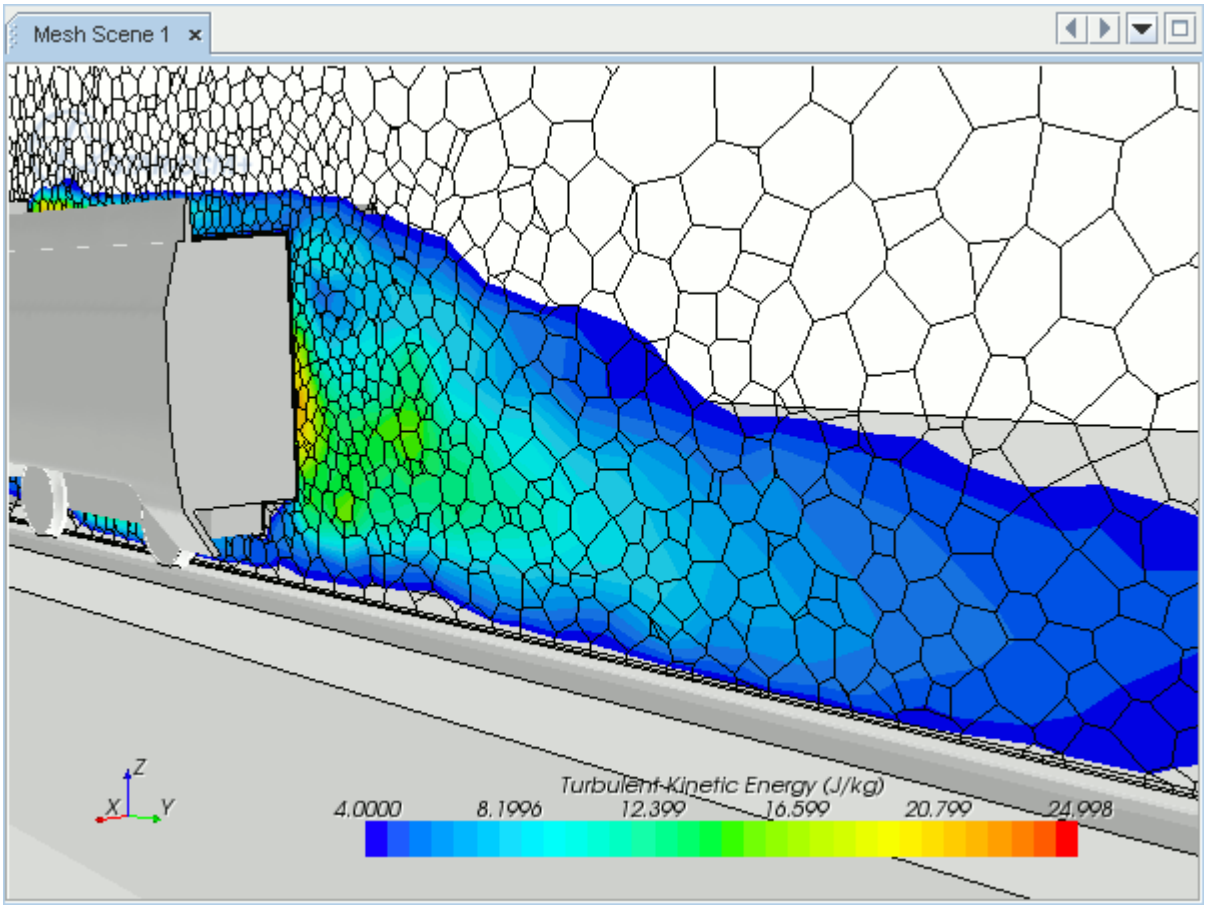


After mesh refinement:

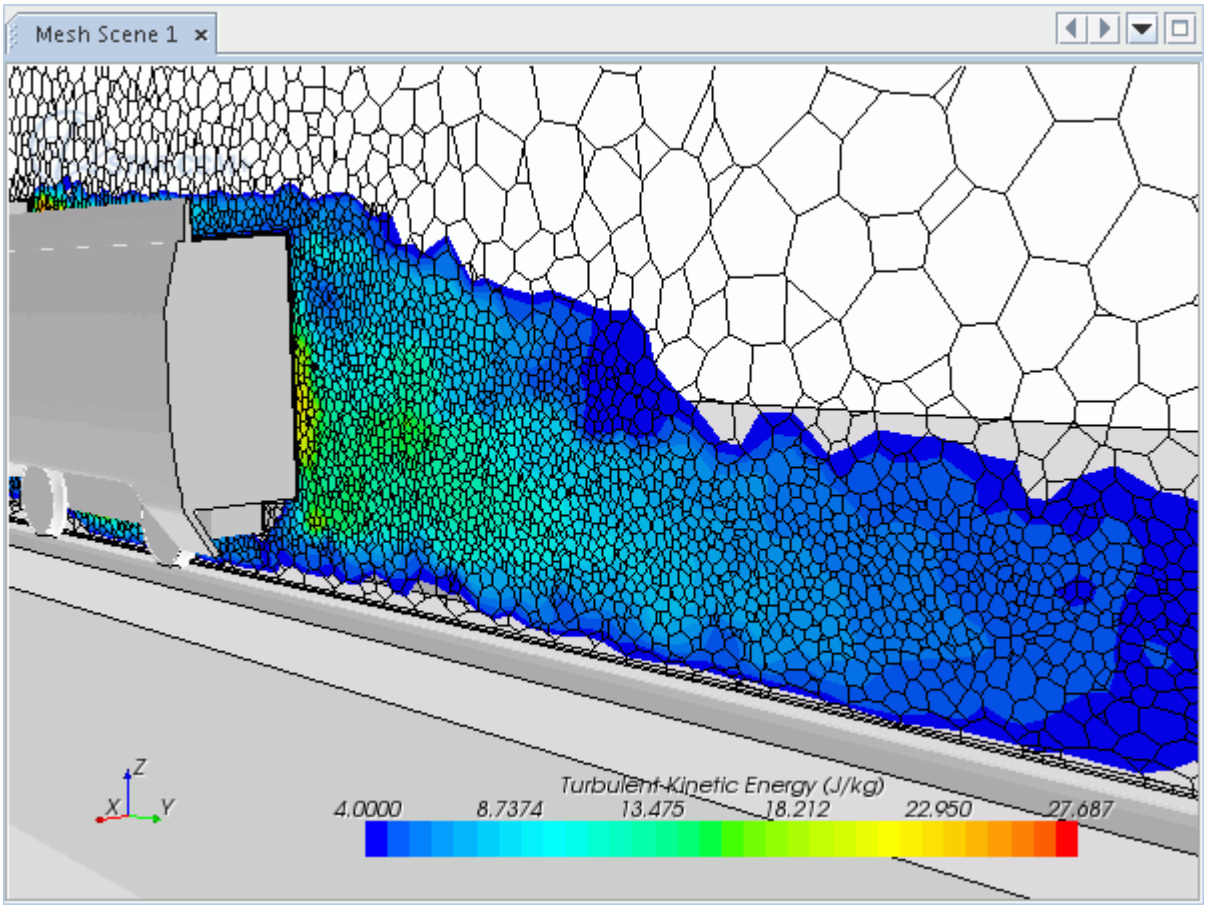


Polyhedral Mesher

Before mesh refinement:

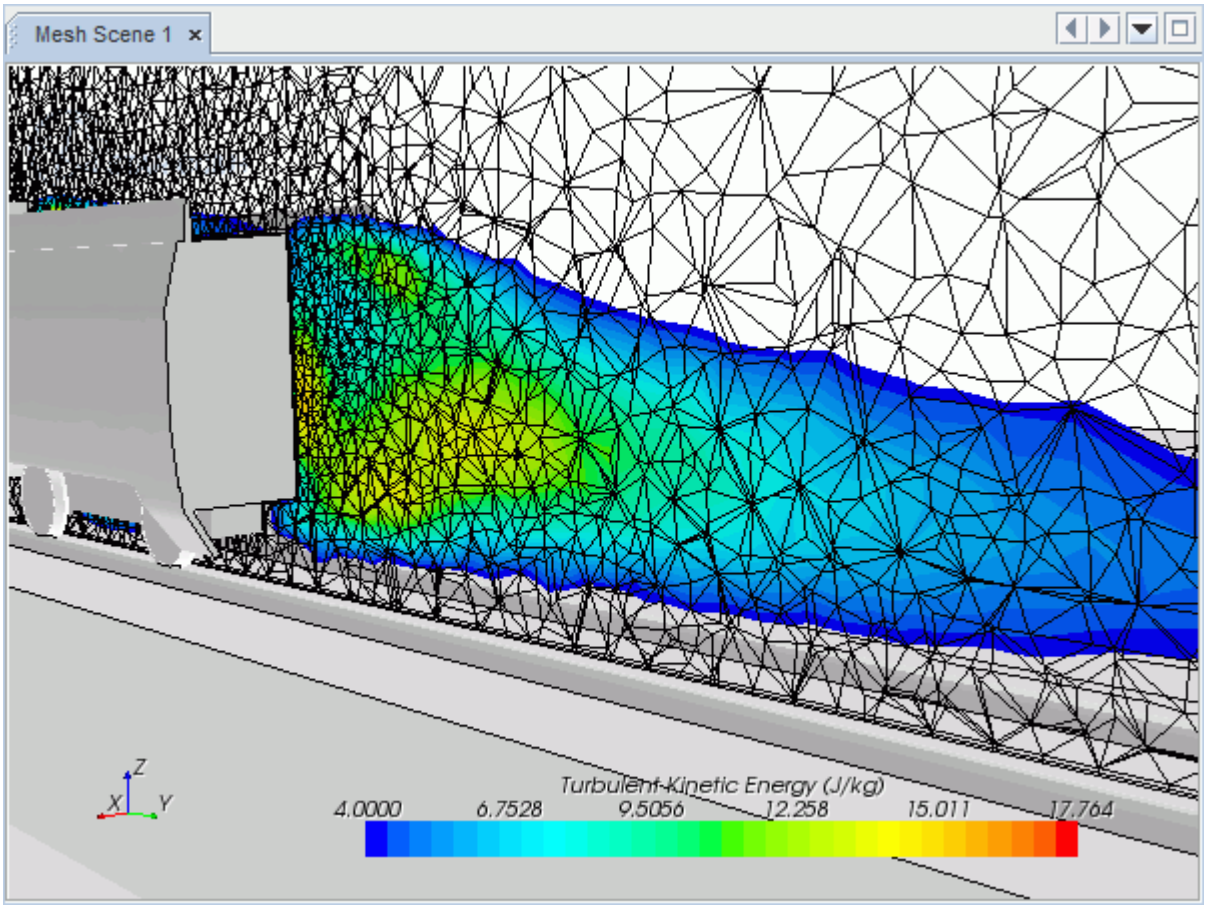


After mesh refinement:

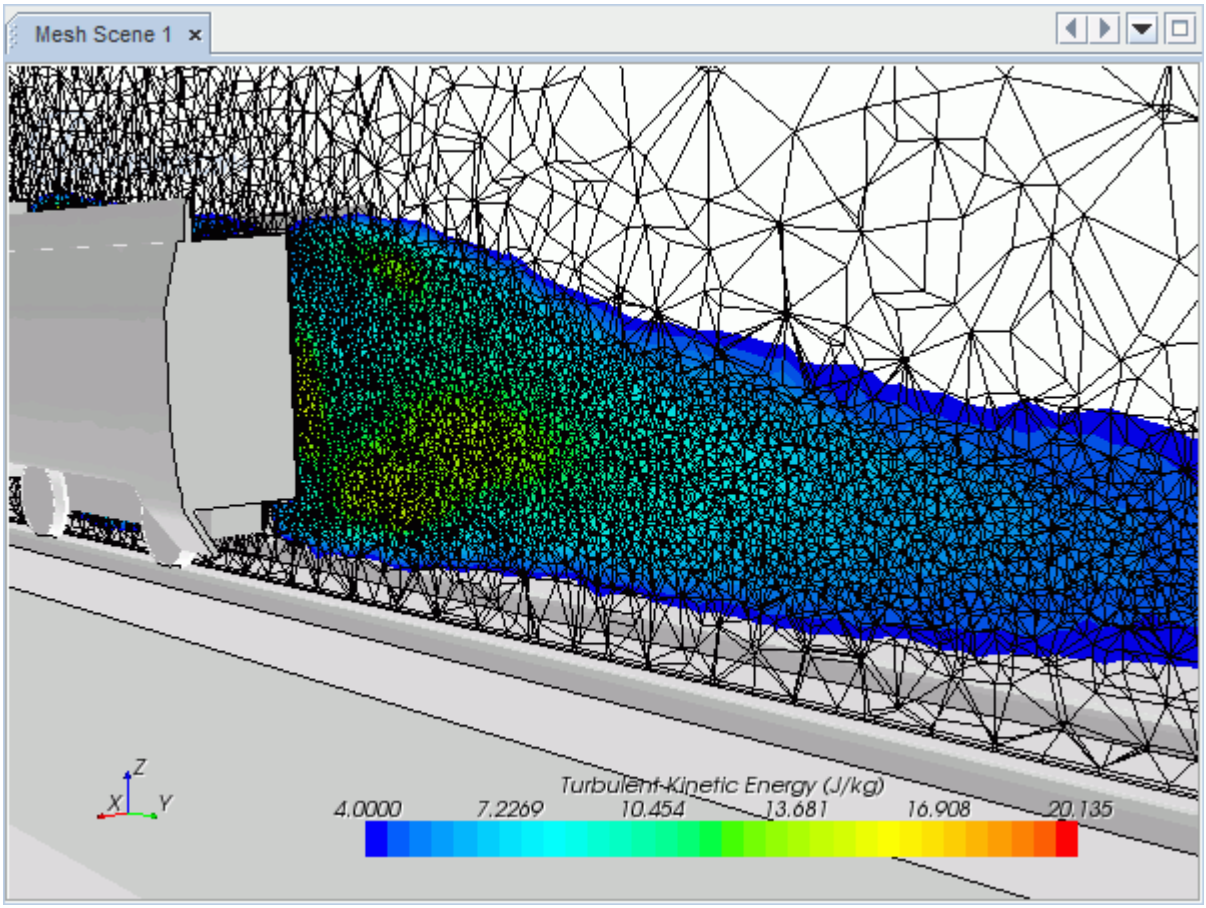


Tetrahedral Mesher

Before mesh refinement:



After mesh refinement:



Contents:

Creating a Field Function for Mesh Refinement
Creating an XYZ Internal Table for Mesh Refinement
Applying the Table of Mesh Sizes

Creating a Field Function for Mesh Refinement

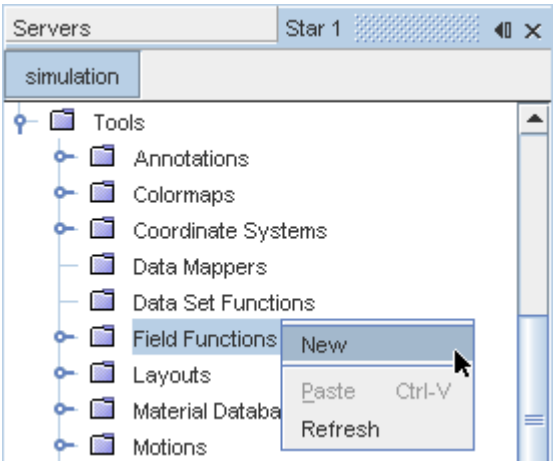
The first step in field function mesh refinement is to create a field function that returns the mesh size with respect to an appropriate solution quantity.

In this example, if the turbulent kinetic energy is greater than 4 J/kg, the mesh size is 0.1 m; otherwise it is 5 m.

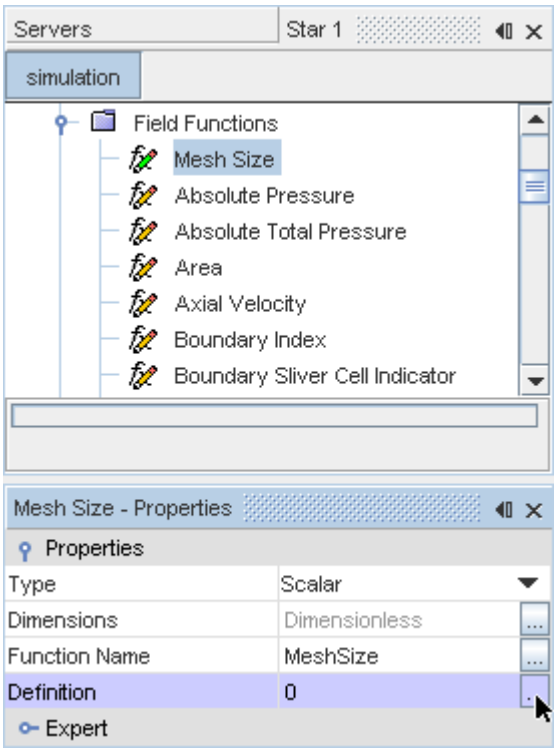
Note: It is important to consider your initial mesh resolution when creating a mesh refinement field function. If your initial mesh is coarse, define a field function that ensures you get a smooth variation in cell sizes. A refinement field function that defines an abrupt change in cell sizes results in poor mesh grading and quality.

To create a field function:

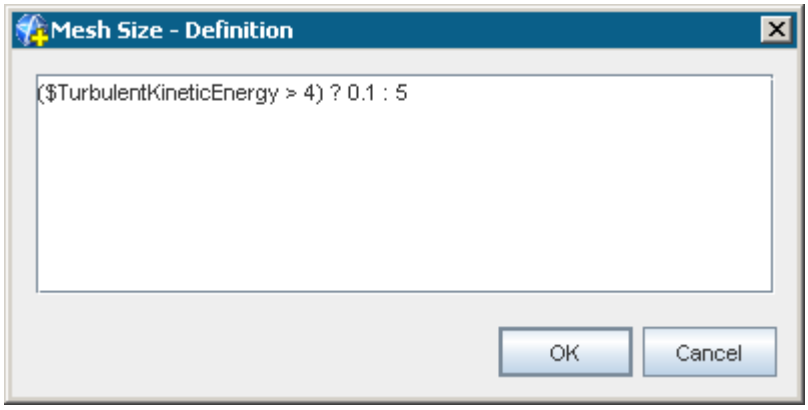
1. Right-click the **Tools** > **Field Function** manager and select **New**.



2. Select the **Field Function** node and click the ellipsis next to the *Definition* property.
   - In this example, the field function name is **Mesh Size**.



3. In the *Field Function - Definition* dialog, enter a conditional expression that returns the mesh size with respect to a solution characteristic.



4. Click **OK** to save the field function definition.

Note: The mesh size units are always meters; regardless of the preferred length unit.

Next, use this field function to create an internal table for mesh refinement.

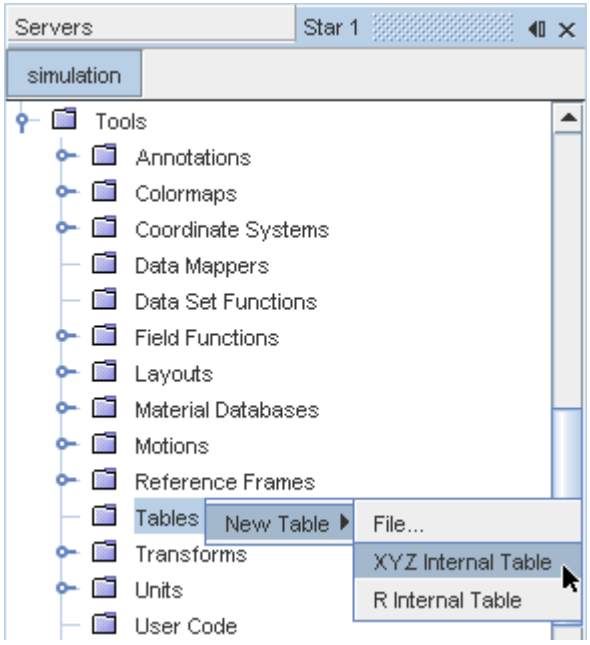
Creating an XYZ Internal Table for Mesh Refinement

You use the mesh refinement field function to populate an XYZ internal table of mesh sizes, with respect to each cell or face centroid. Populating the table applies the field function to the existing volume mesh and its solution data.

Note: If you are using the Trimmer meshing model, ensure that the coordinate systems of the Trimmer and the XYZ internal table are the same.

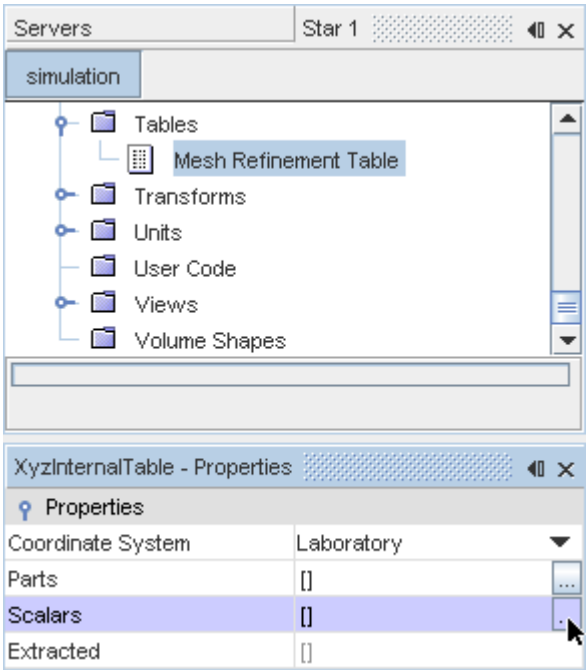
To create an XYZ internal table:

1. Right-click the **Tools** > **Tables** node and select **New Table** > **XYZ Internal Table**.

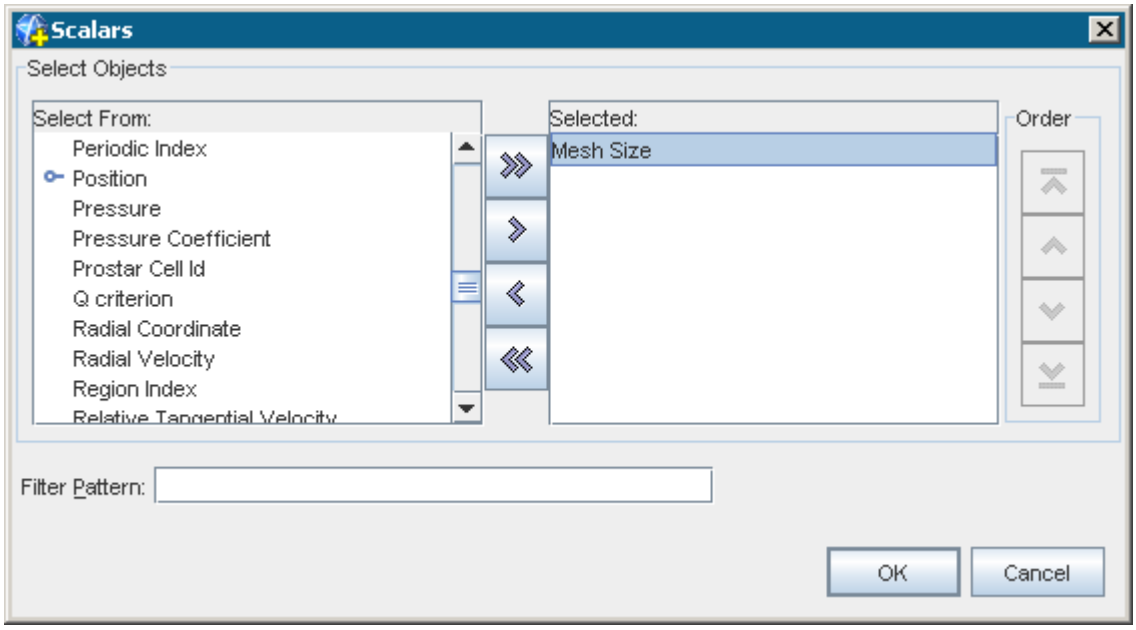


2. Select the **XYZ Internal Table** node and click the ellipsis next to the *Scalars* property.

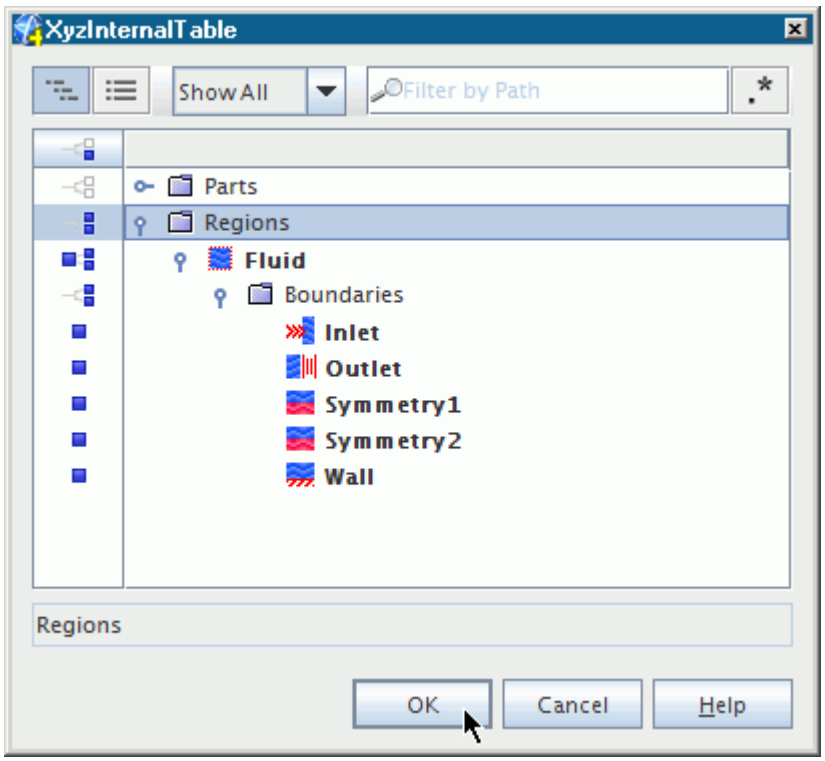
In this example, the table name is **Mesh Refinement Table**.



3. In the *Scalars* object selection dialog, move your previously defined mesh refinement field function to the *Selected* box and click **OK**.



4. Similarly, click the ellipsis next to the *Parts* property and expand the regions node.
5. Select the regions and boundaries that you want to refine, and click **OK**.



6. Right-click the **XYZ Internal Table** node and select Extract to compute the field function for each of the selected region cells or boundary faces.
7. To display the contents of the mesh refinement table, right-click the table node and select Tabulate.

As shown in the following image, the mesh size is given for the cell and face centroids of each selected region and boundary.

Tabular Data

	Mesh Size	X (m)	Y (m)	Z (m)
187	5	0.675329983234	-36.531034987747	0.4111494543
188	0.1	0.675329983234	-36.371993026883	0.254308285842
189	0.1	0.675329983234	-36.359524521476	-0.001391675148
190	0.1	0.675329983234	-36.450970056558	0.026693820725
191	0.1	0.675329983234	-36.602629971532	0.164914876204
192	0.1	0.675329983234	-36.642705568733	0.261697093148
193	0.1	0.675329983234	-36.256248740624	0.081044657548
194	0.1	0.675329983234	-36.20712458638	0.121548854232
195	5	0.675329983234	-36.076168080704	0.005244237562
196	0.1	0.675329983234	-36.254069077788	-0.011615332361
197	5	0.675329983234	-50.594304324404	0.580038228804
198	5	0.74018119816	-50.613782345044	0.570898678579
199	5	0.675329983234	-50.519325994633	0.811923251723
200	5	0.683407646946	-50.517212626608	0.809935350978
201	5	0.675329983234	-50.574336904329	0.724442821671
202	5	0.683407646946	-50.550739657193	0.770534730738

Export... Close

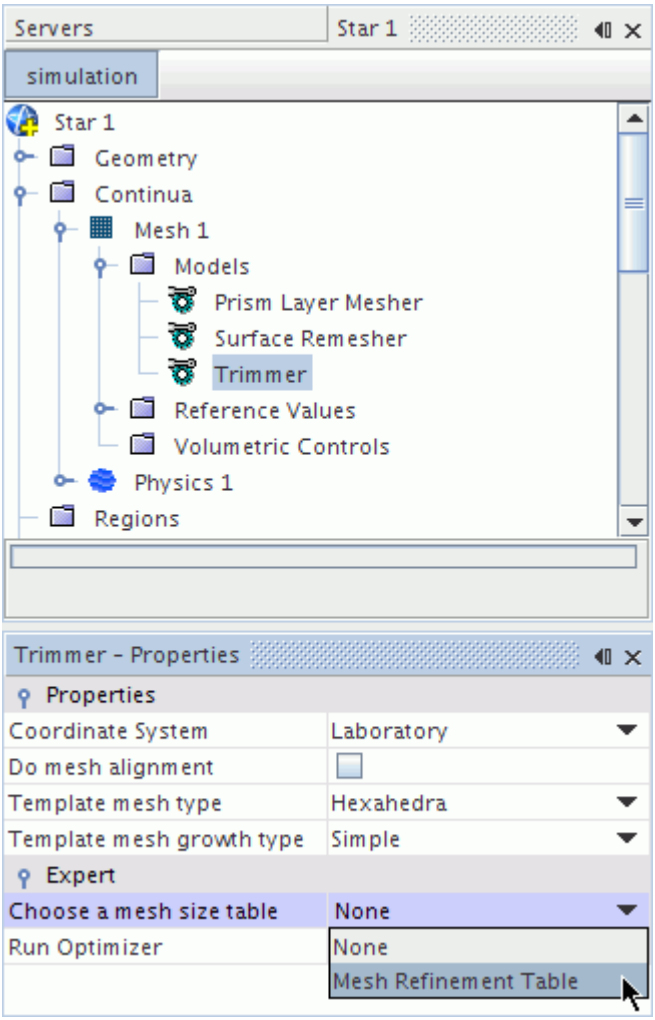
Next, assign this table to the appropriate meshing model.

Applying the Table of Mesh Sizes

The final stage in field function mesh refinement is to apply the table of mesh sizes to the meshing model properties.

To assign the mesh refinement table to the meshing model:

1. Select the **Mesh** node and set *Choose a mesh size table* to the name of the table that you previously created.

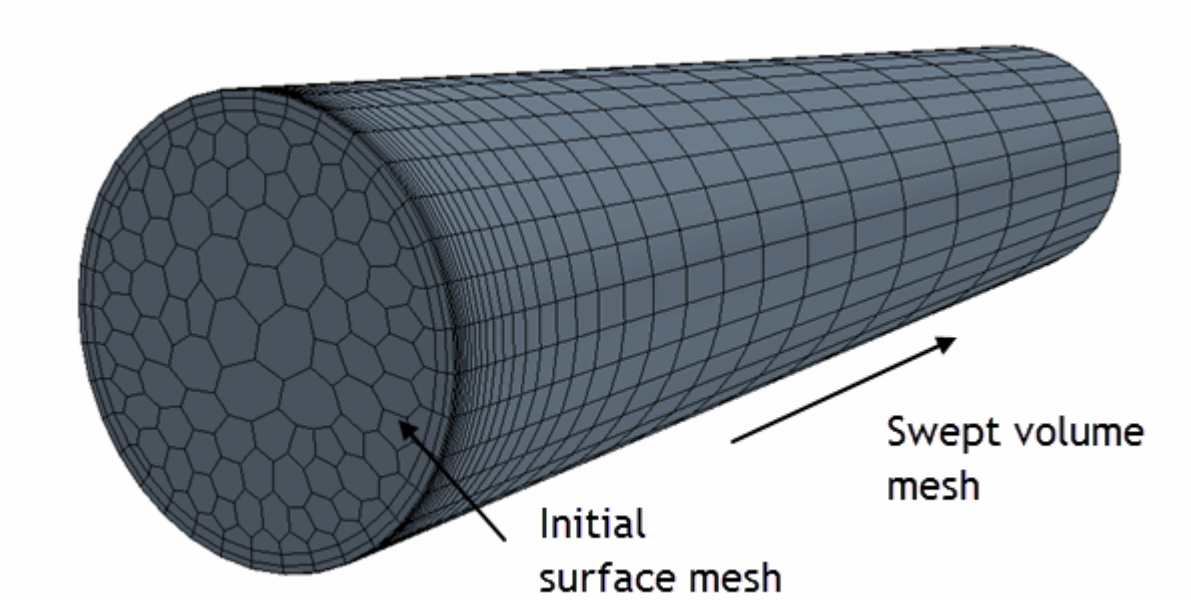


You can now regenerate the mesh in the usual manner for surface and volume meshers.

Directed Meshing

Directed meshing is a method for generating high quality swept meshes on CAD geometries in STAR-CCM+. The method operates by sweeping a mesh from the surfaces of a CAD geometry through its volume onto a facing target surface.

You can use a mesh that is created using the directed mesh tool in any STAR-CCM+ analysis, along with meshes created using other methods. Directed meshing is ideal for meshing FV (Finite Volume) fluid flow simulations as it provides a structured mesh in the axial direction. When using the Solid Stress model for FE (Finite Element) stress analysis, you can use the directed mesher to generate 3D hexahedral elements. For more information on the mesh requirements for FE simulations, see Finite Element Mesh: Requirements and Guidelines.



The directed mesh feature gives you a high degree of control when generating a volume mesh. For example, you can create hexahedral meshes for highly irregular geometries in one operation. You can also create meshes that are flow-aligned, resulting in better accuracy within solvers.

The sweep paths can be linear or curved and the cross section may change in size and shape. Source and target surfaces do not need to be co-planar.

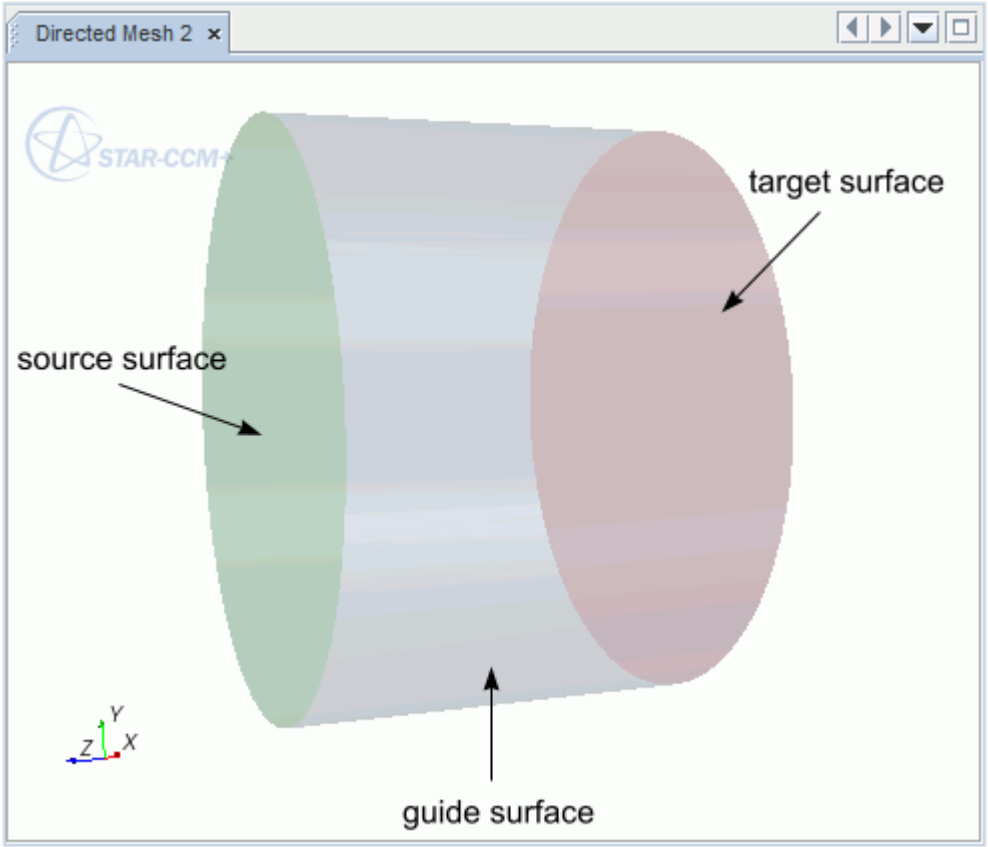
You can use the directed meshing feature on geometry parts whose surfaces are designated as:

- Source surfaces, where you generate the initial surface mesh
- Target surfaces, where the directed mesh ends
- Guide surfaces along which the mesher sweeps the initial mesh from the source surfaces to the target surface

The guide surfaces in the region that you intend to mesh with directed meshing must have complete CAD association.

It is also possible to create internal mesh faces using empty part surfaces (1-1 meshing only).

Note: You can create and select settings for a directed-meshing operation node, but you must assign parts to regions before executing it.



There are two directed meshing strategies:

1. **Many sources to one target (M-1)** - Meshes from multiple source faces are swept through the volume to a single target surface in one operation. M-1 allows you to mesh many-sided part geometries, with source surfaces at different layers of the sweep. Some examples are shown below.
2. **One source to one target (1-1)** - A directed mesh is produced for each part collection. If necessary, you can split-up complex parts so as to provide the necessary guide surfaces.

M-1 is an automated approach, whereby STAR-CCM+ submaps the guide surfaces to enable the mesh to be swept through the volume. There are some restrictions related to the M-1 approach that are discussed in context within the following sections. In some instances, it will be prudent to mesh sets of parts individually, using a 1-1 approach.

Directed meshes can be created in several ways:

- Creating patches on the surfaces and defining quad cells within the patches (Note: 1-1 directed meshing only) . See Creating a Source Mesh Using Patches.

- Extracting meshes from existing volume meshes on which the surfaces lie. See Creating a Source Mesh Using an Existing Mesh.
- Create 2D source meshes on which you can include prism layers. See Creating a 2D Source Mesh.

Note: You cannot use different mesh types in combination for the same source surface of a part. Patch meshing is not supported for M-1 meshing.

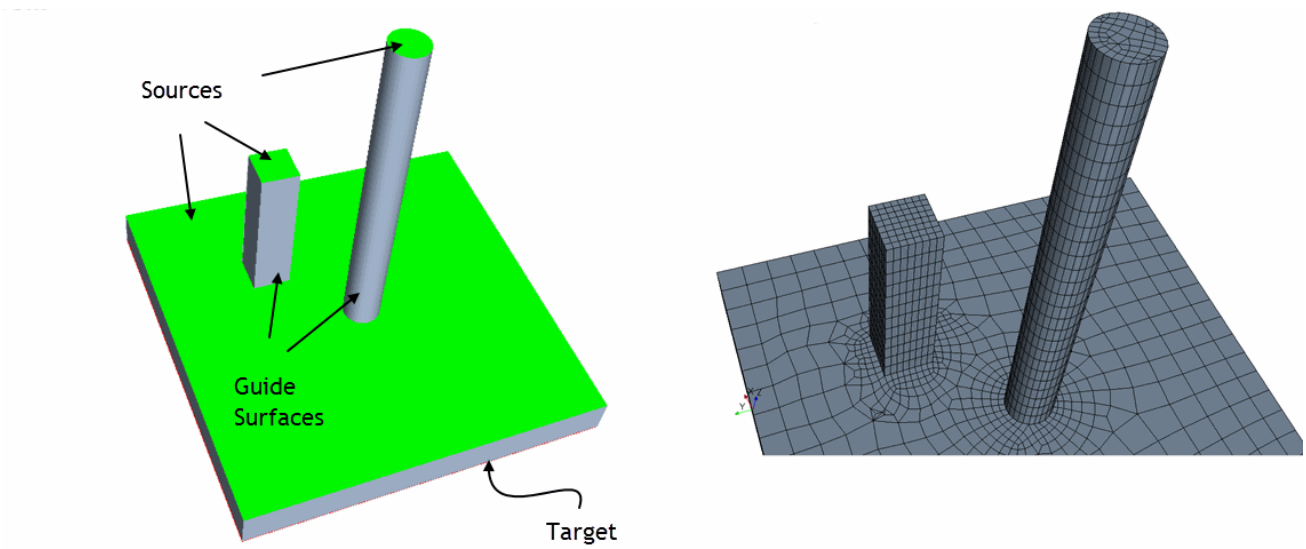
Each source surface mesh is associated with a volume mesh distribution that is used to generate the final volume mesh.

Currently, the smoothing algorithms within the mesher aims to replicate the mesh on the source surface onto the target surface whenever possible. When source or target surfaces are non-planar, this process can generate poor quality and negative volume cells—especially when using M-1. In some cases (such as for standard cube and cylinder shapes), it is not necessary to smooth the target surface mesh. You can deactivate this option. See Directed Meshing Properties: Connected Parts - Part Properties.

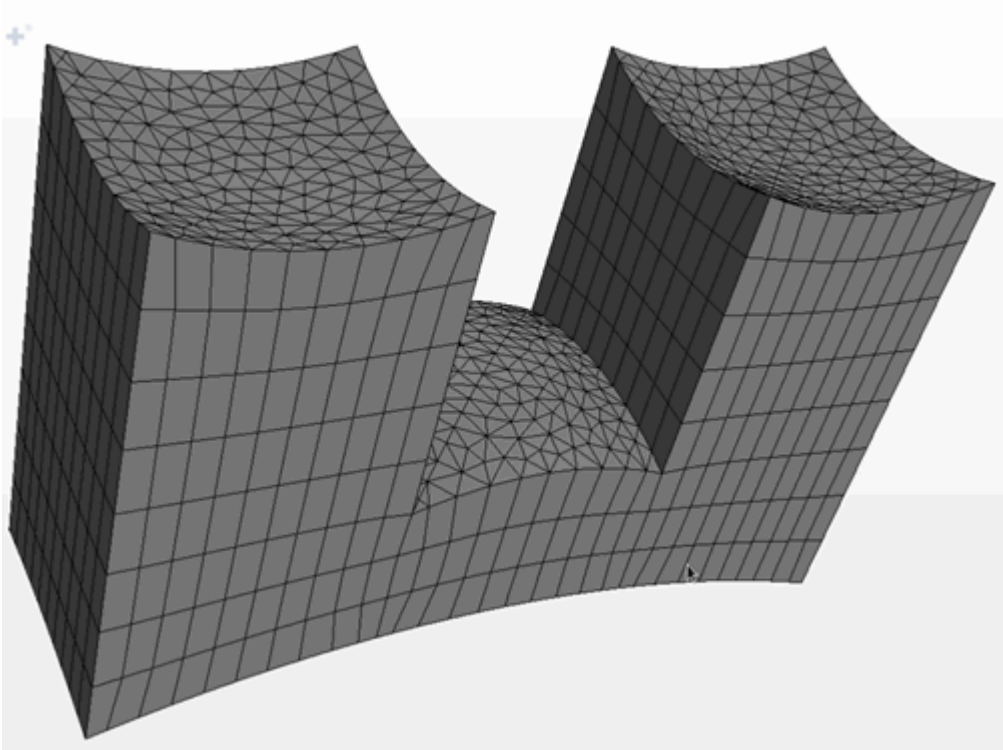
STAR-CCM+ can perform directed meshing when there are spanning faces on both source and target surfaces. However, if the target surface has multiple CAD faces, and the edges between those CAD faces are not marked as features, these edges are not resolved and, in effect, there is only a single target surface and the directed mesher may fail.

Examples showing directed mesher sweep capabilities:

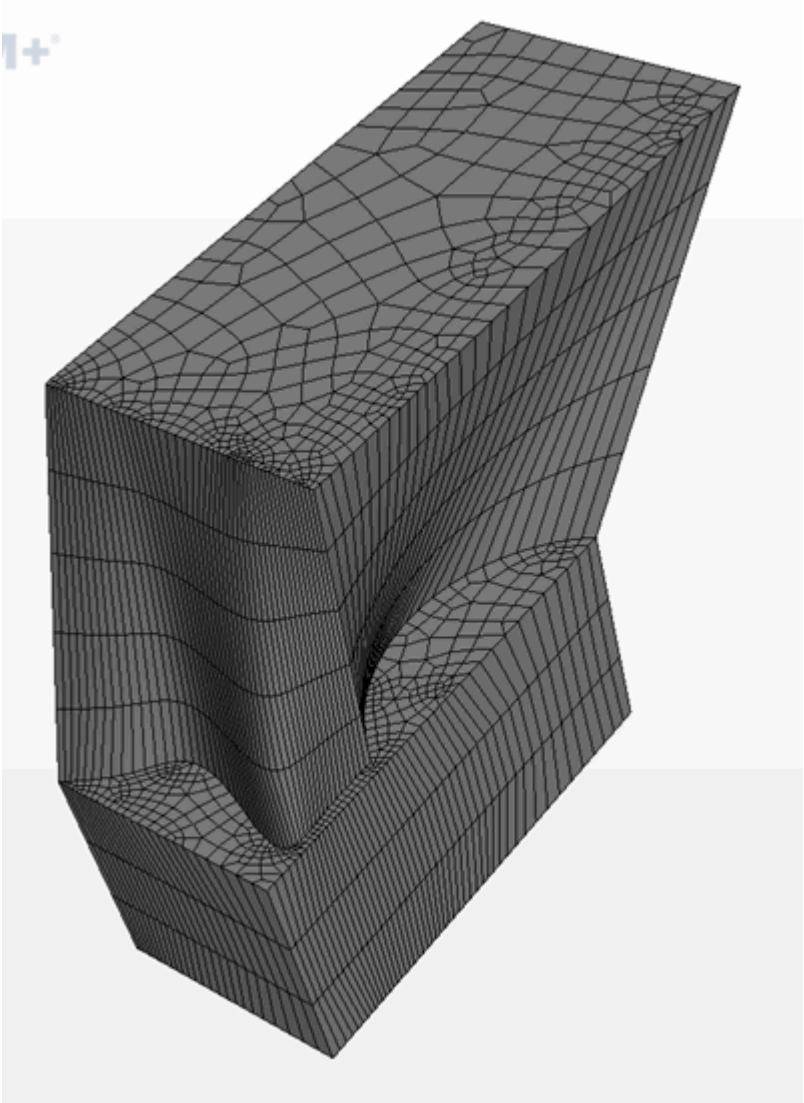
- Three co-planar sources at different sweep layers swept to a single target (M-1). Here, the extended pin surfaces are unconnected:



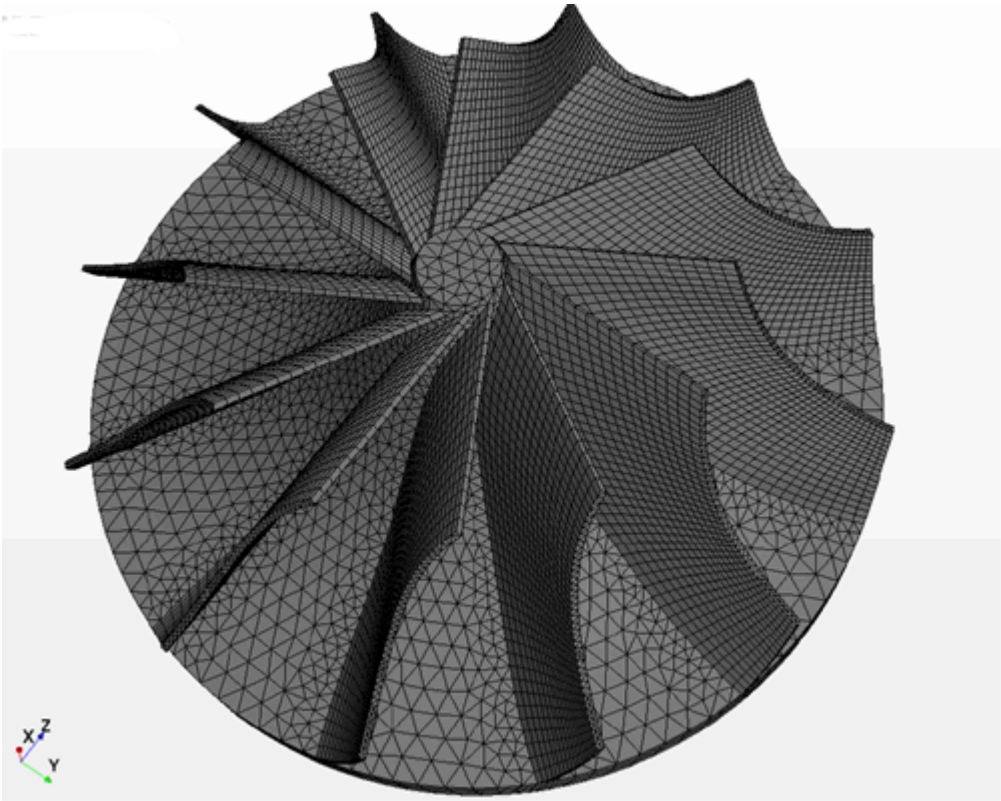
- Part with 3 x curved triangular source meshes and curved target:



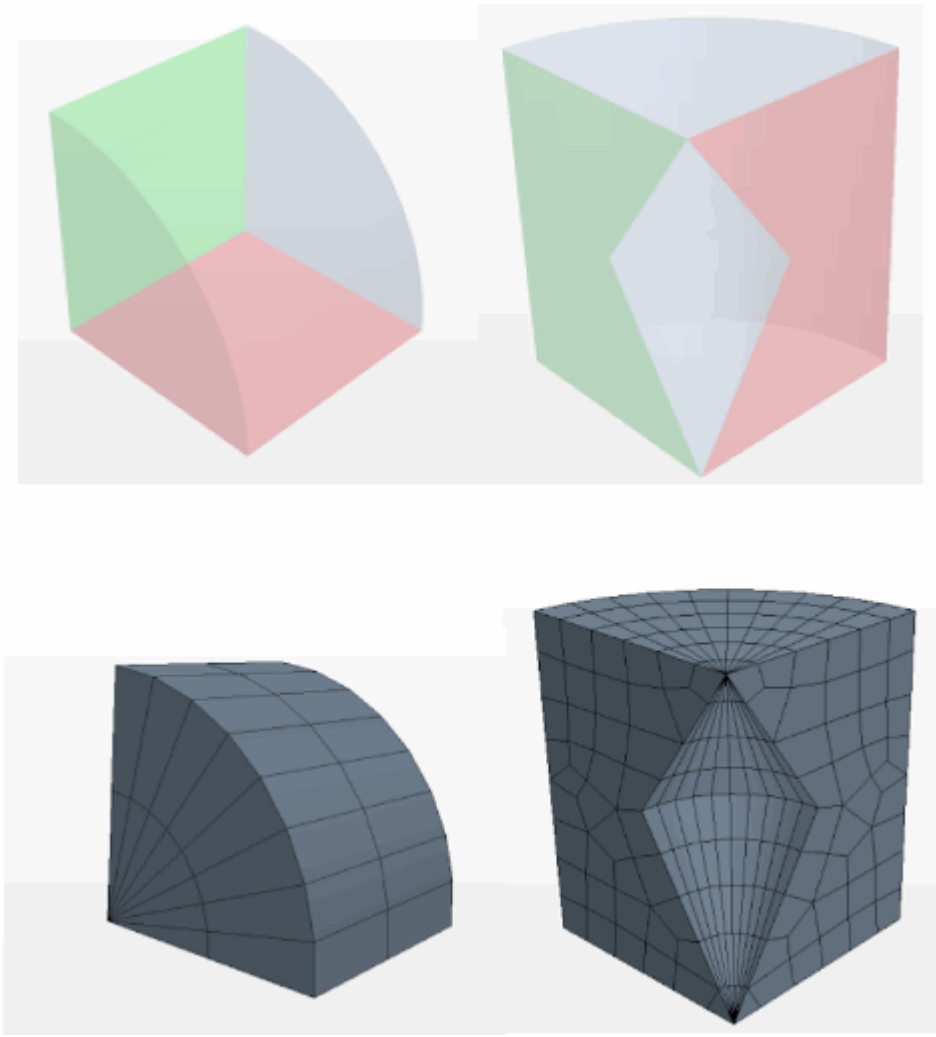
- Part with quadrilateral source meshes, loft and curved submapped surfaces in the sweep direction:



- Impeller with curved sweep in two planes:



- Rotational sweeps about a common edge or connected vertices



Creating Directed Meshes in STAR-CCM+

The subsequent sections provide information on setting up and generating directed meshes in STAR-CCM+.

Contents:

Preparing CAD Parts for Directed Meshing
Directed Meshing Workflow
Choosing Source and Target Surfaces
Creating a Source Mesh Using Patches
Creating a Source Mesh Using an Existing Mesh
Creating a 2D Source Mesh
Deleting a Source Mesh
Creating a Volume Distribution
Deleting a Volume Distribution
Generating the Directed Volume Mesh
Meshing a Stack of Parts

Meshing a Single Part in a Closed Loop with a Baffle
Directed Meshing Properties

Preparing CAD Parts for Directed Meshing

This section provides information on preparing CAD parts for directed meshing.

You can only use directed meshing with geometry parts that have CAD association. You can either import CAD geometry, or create geometry using the 3D-CAD modeler. Directed meshing operations use the latest CAD associated description for a part – which can be either the root description, or the description after a transform operation. See Mesh Descriptions.

The following requirements must be met regardless of the sources used on the geometry parts:

- Each part must have a regular section profile along its length.

  There cannot be any features intruding into the sweep path between the source surfaces and target surface. For example, shapes such as Y-junctions cannot be meshed without being split into two mirrored parts.
- Each part must have at least three surfaces, representing the source, target, and guide surfaces.

  If neccessary, you can subdivide any part into the necessary surfaces using the split by angle or split by patch options (1-1 directed meshing only).
- A logically rectangular guide surface with five or more CAD edges is allowed.

  Include CAD vertices in the distribution path even if the distribution does not land on those points. Directed meshing smooths over any CAD vertices that are not designated as a feature.

  Note: In cases where there is a dramatic bending of the sweep path, there may be a requirement to manually create four-sided guide surfaces in order to ensure the part is meshed successfully.
- Imprint contacting parts on each other and ensure that part contacts are correctly defined on each part.

  Use the CAD-based imprinter under Parts, or ensure that imprinting is active for features in the 3D-CAD modeler. Do not use the surface repair imprinter or the discrete imprinter under Parts as doing so disassociates the CAD from the surface.
- If a part, or set of parts, is associated with a region that is already meshed using a mesh continuum, then that part is not available for use within the directed meshing selection dialog.

  If this is the case, delete the region or disassociate the part from the region in order for it to become available for directed meshing.

  Note: Changing a 3D-CAD model whose parts are used in a directed mesh operation can result in the directed mesh operation becoming invalid. In this case, apply an update and review the settings in the directed mesh operation.

Directed Meshing Workflow

In STAR-CCM+, a directed mesh has its own operation node that identifies the parts in which the mesh belongs, and the sources and target surfaces within those parts.

Make sure that you have CAD geometry that meets the requirements for a directed mesh. See Preparing CAD Parts for Directed Meshing.

To create a mesh using the directed mesh approach:

1. Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Directed Mesh**.
2. In the *Create Directed Mesh Operation* dialog, select the parts that you want to use for the directed meshing operation and click **OK**.

   STAR-CCM+ creates an **Operations** > **Directed Mesh** node.

   Parts that are selected for directed meshing are listed under the **Directed Mesh** > **Connected Parts** node. Multiple parts that form a contiguous set of bodies are represented by a single connected parts object.

   Note: The yellow warning triangles indicate that further action is needed before the operation is fully defined. The triangles disappear when you generate the volume mesh.

For subsequent steps, you are generally advised to work with the *Directed Mesh* panel activated, as this mode allows interactive editing within the graphics window. However, you can also choose particular child nodes in the simulation tree, and edit their properties directly.

To activate the Directed Mesh panel:

3. Right-click the **Operations** > **Directed Mesh** node and choose **Edit...**
4. Choose the source surface(s) and target surface for each connected part.

   See Choosing Source and Target Surfaces.
5. Create the starting mesh on the source surfaces. Choose from one of the following methods:
   - Draw patches on the source surface and have STAR-CCM+ mesh these with quad faces (applies to 1-1, single-source to single-target directed meshing only). See Creating a Source Mesh Using Patches.
   - Take the source meshes from the side of an existing volume mesh. See Creating a Source Mesh Using An Existing Mesh.
   - Create a 2D source mesh using an 2D Auto Mesh operation. See Creating a 2D Source Mesh.
6. Define a volume distribution for the swept cells between the source and target surfaces.

   See Creating a Volume Distribution.
7. Generate the volume mesh.

   See Generating the Directed Volume Mesh.

Choosing Source and Target Surfaces

Before creating a surface meshes for directed meshing, identify the sources and target surface for each part.

Connected parts must share the source surface(s) and the target surface. The same volume mesh distribution parameters are applied to all of the connected parts. If the connected parts are of different lengths, then STAR-CCM+ makes an estimate with regards to the distribution in the sweeping dimension. The directed meshing process transfers the mesh from source(s) onto the target surface. Therefore, target surfaces cannot contain internal feature edges.

Note: Naming the faces on either side of the interfaces in a stack of parts makes it easier to identify the source and target surfaces for directed meshing. See Meshing a Stack of Parts for additional information.

Contents:

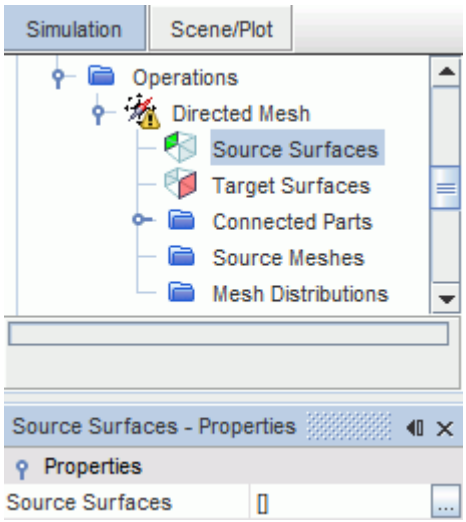
Selecting Source and Target Surfaces Using the Tree
Selecting Source and Target Surfaces Interactively
Setting Source and Target Parts as Periodic Interfaces

Selecting Source and Target Surfaces Using the Tree

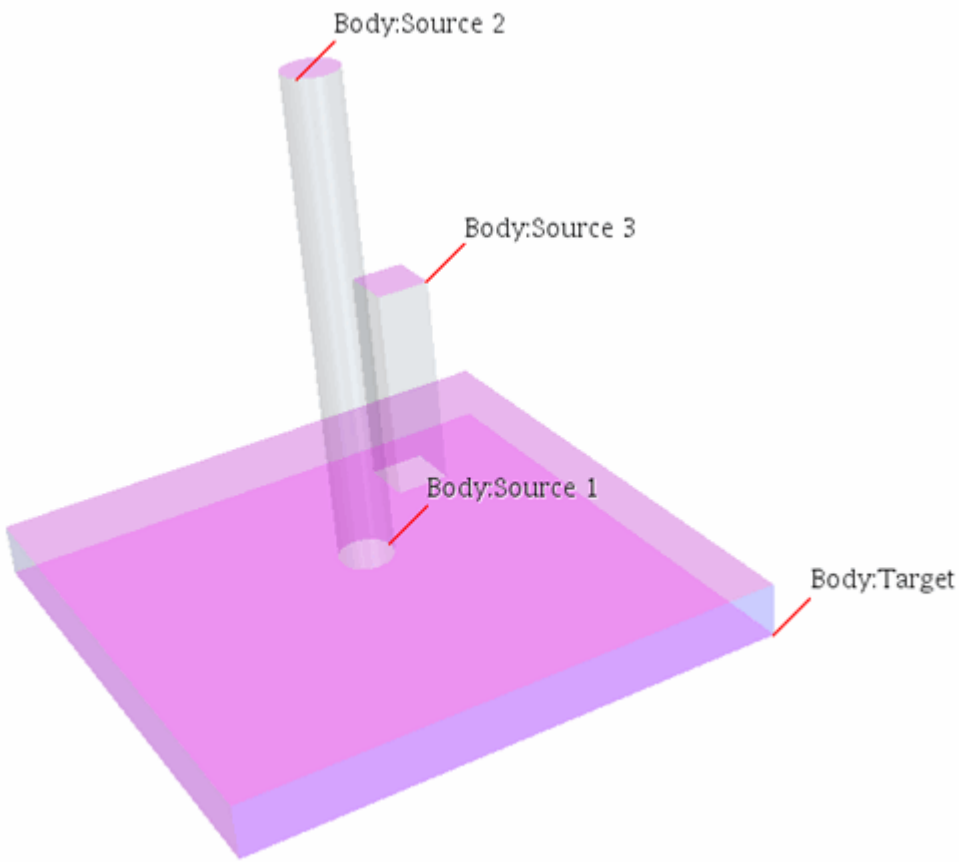
This section describes how to select source and target surfaces.

To select a source surface:

1. Select the **Directed Mesh** > **Source Surfaces** node and in the *Properties* window, click the ellipsis button ... to the right of *Source Surfaces*.

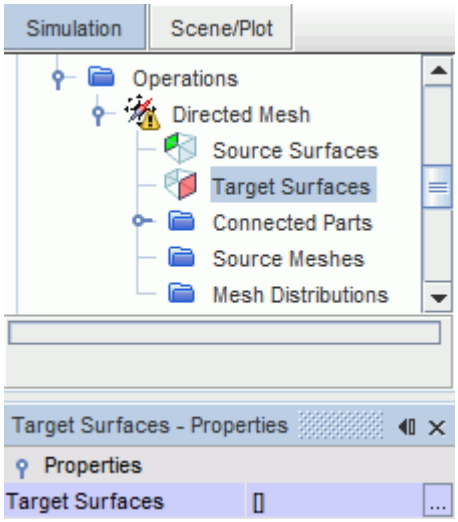


2. In the *Source Surfaces - Source Surfaces* dialog, select at least one surface to represent the source surface(s). For M-1 sweeping, discontiguous source surfaces can be used which are at different sweep levels.



To select a target surface:

3. Select the **Directed Mesh** > **Target Surfaces** node and in the *Properties* window, click ... to the right of *Target Surfaces*.



4. In the *Target Surfaces - Target Surfaces* dialog, select at least one face to represent the target surface.

To remove any faces from source or target surfaces:

5. Select the **Source Surfaces/Target Surfaces** node.
6. In the *Properties* window, click ... for *Source Surfaces/Target Surfaces*.
7. In the object selection dialog, deselect any surfaces that are not required as a source or target surface.

Selecting Source and Target Surfaces Interactively

In the *Graphics* window, you can select surfaces on the geometry and add them to the source or target surfaces for the mesh.

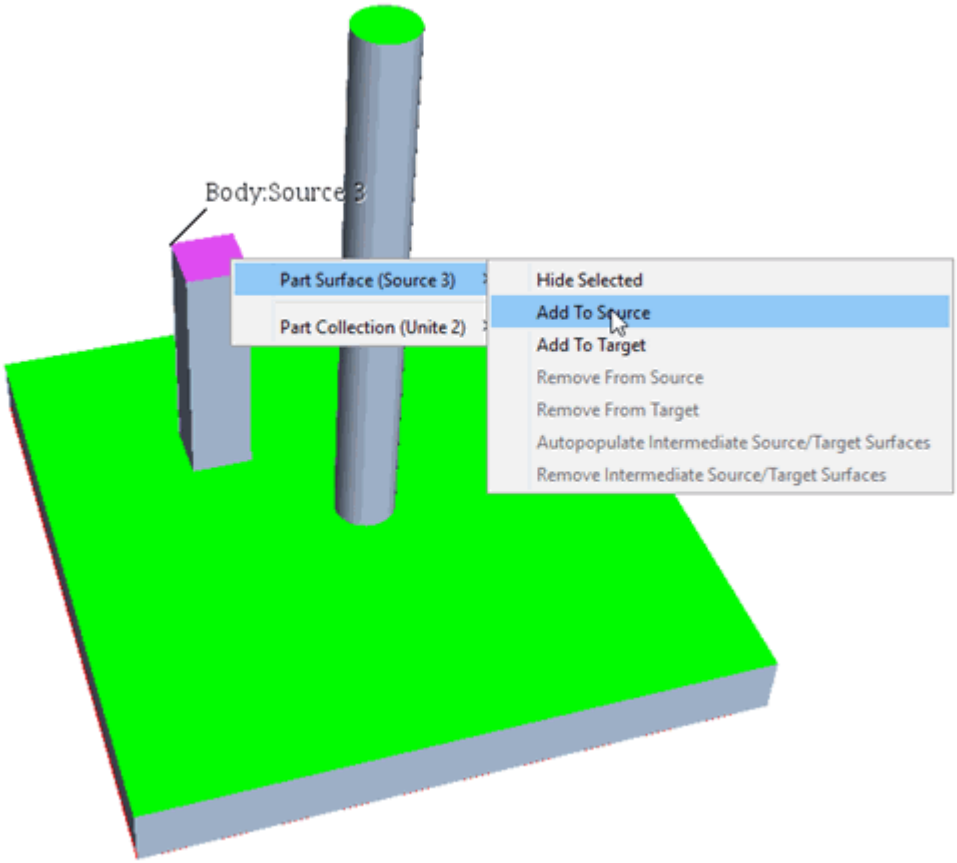
To select the source and target surfaces interactively in the *Graphics* window:

1. Edit the **Operations** > **Directed Mesh** node.
2. Select the surfaces that represent the source or target surface:
   - To select a single surface in the *Graphics* window, click the surface that you require.
   - To select multiple surfaces in the *Graphics* window, there are several methods:
     - Use `<Ctrl>` + click
     - Double-click

       When you double-click a part surface, all contiguous part surfaces that meet within an angle of 30 degrees are selected.
     - Use the rubberband select tool

       Activate the rubberband tool and select the required surfaces. See Selecting in a Rubberband.

   Note: If you are meshing a part collection using the 1-1 directed mesher approach, STAR-CCM+ can automatically account for intermediate source and target surfaces that exist between the first source surfaces and the final target surfaces. This feature is useful when you mesh a stack of parts as you do not have to specify each contacting surface as either source or target throughout the stack. In this situation, only define the first source surfaces and final target surfaces at this stage. This feature is not available for M-1 directed meshing.
3. In the *Graphics* window, right-click the selected surfaces.

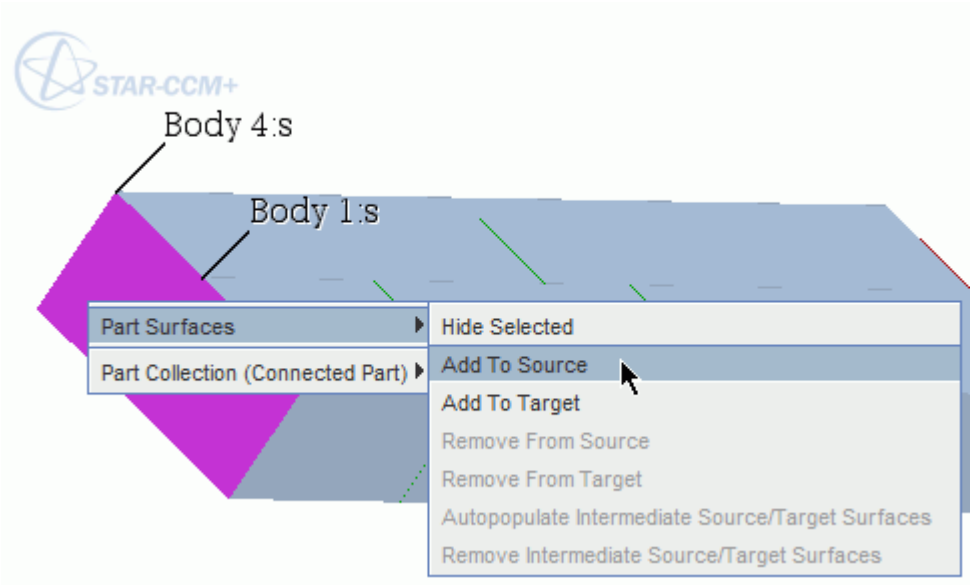


4. In the pop-up menu, select the appropriate option.

   For surfaces on individual parts:

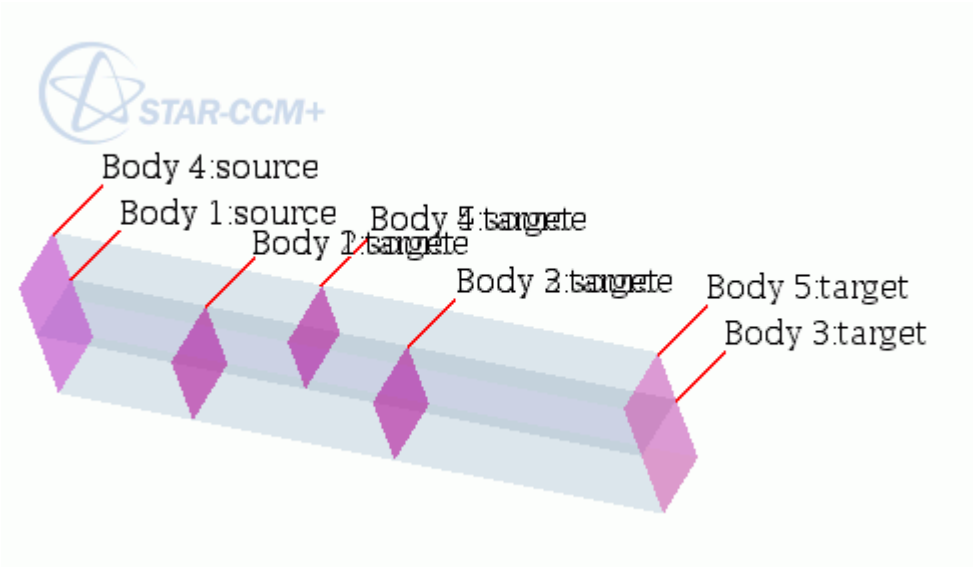
   - **Part Surfaces** > **Add to Source**
   - **Part Surfaces** > **Add to Target**

   For part collections, you only need define the first surfaces as source, and the final surfaces as target.



5. If you are meshing a part collection, right-click either the first source surfaces or the final target surfaces, and select **Part Surfaces** > **Autopopulate Intermediate Source/Target Surfaces.**

   Intermediate source and target surfaces are defined on contacting surfaces that exist between the first source surfaces and the final target surfaces.



To remove any surfaces from source, target, or intermediate surfaces:

6. Right-click the source or target surface that you want to remove. For intermediate source and target surfaces, right-click either a source or target surface.
7. Select the appropriate option:
   - **Part Surface (source)** > **Remove From Source**
   - **Part Surface (target)** > **Remove From Target**
   - **Part Surfaces** > **Remove Intermediate Source/Target Surfaces**

Setting Source and Target Parts as Periodic Interfaces

It is possible to use boundaries that are specified as periodic interfaces for source and target surfaces in directed meshing.

You can also specify surfaces that are already defined as source and target parts as periodic interfaces:

- Create the periodic contact between the Source and Target surfaces.
- Assign Parts to Regions.
- Verify that the Rotational and Translation properties are correct.
- Proceed to Meshing:
  - Creating a Source Mesh Using Patches
  - Creating a Source Mesh Using an Existing Mesh
  - Creating a 2D Source Mesh

Creating a Source Mesh Using Patches

For a source mesh using patches, you first define the patches, and then generate quadrilateral faces within the patches. This option is only available as part of the 1-1 meshing strategy.

A patch is a four-sided shape that is made up of four patch vertices. The curves that define a patch can be straight or curved, and can be derived from CAD feature curves on the original geometry parts.

You can create as many patches as you like on the source surfaces providing you meet the following conditions:

- There is at least one patch
- The patches cover the entire surface

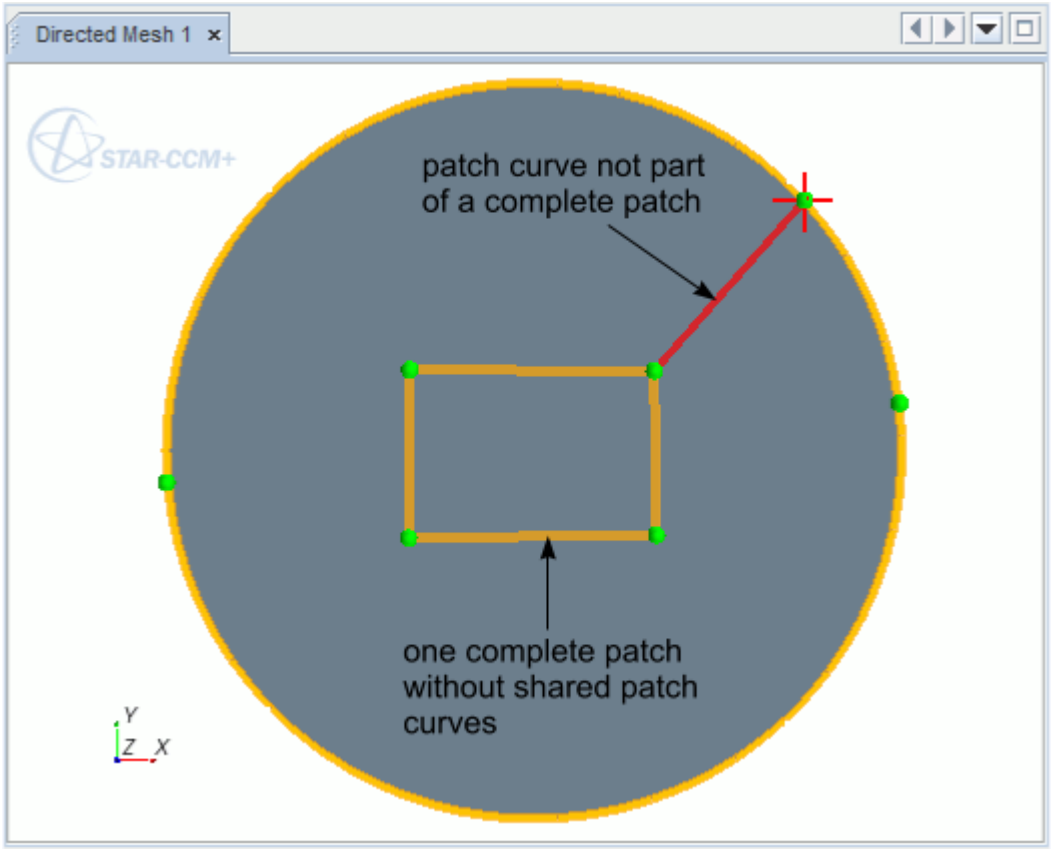
Currently, it is not possible to specify the exact location of patch vertices that connect the patch curves using coordinate values.

For patch meshing, all patch curves must be part of a closed loop. Patch curves that are not on feature edges must also be shared between patches.

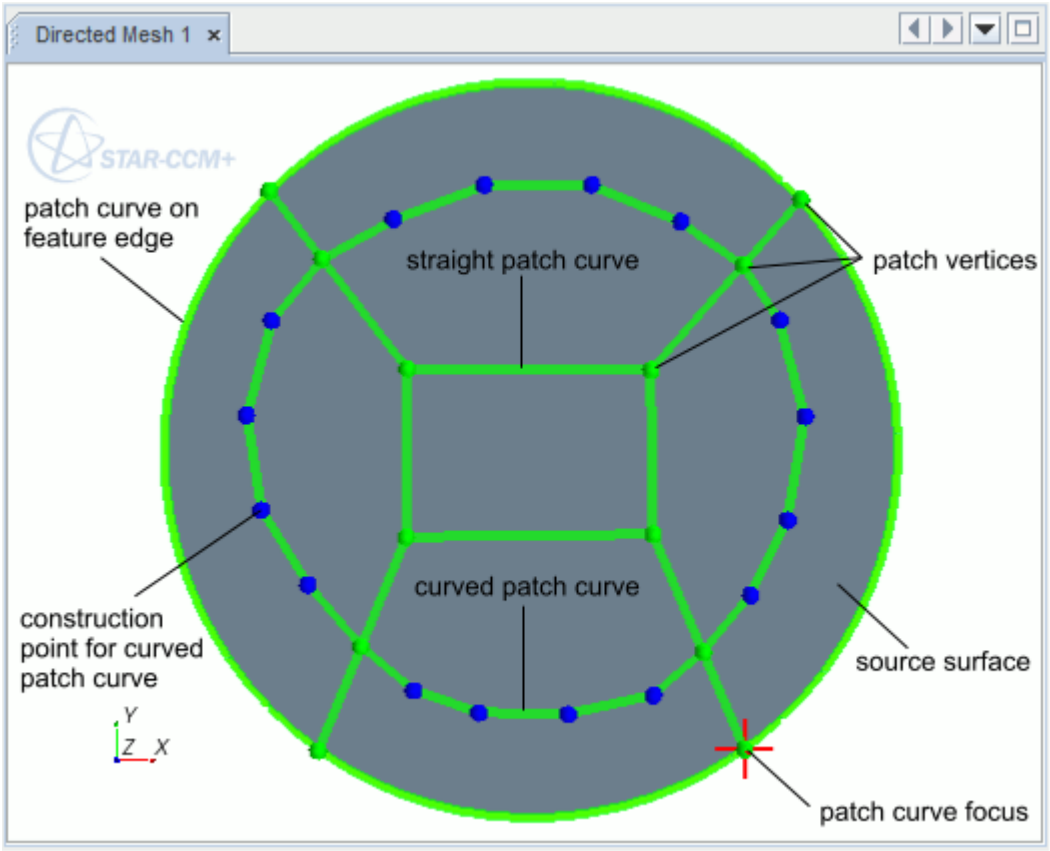
Patch curves are colored according to their status:

- Red indicates that the patch curve is not part of a closed loop; add additional curves to create a complete patch
- Yellow indicates that the curve is part of a closed loop but it is only associated with a single patch
- Green indicates that the curve is part of a valid patch

You can only create a patch mesh when all the patch curves are green.



Patch vertices (green points) connect patch curves. A patch curve is created from a patch vertex that has the “focus”, shown using red lines passing through the vertex point. For curved patch curves, blue construction points are used to guide the curve, but do not act as patch vertices.

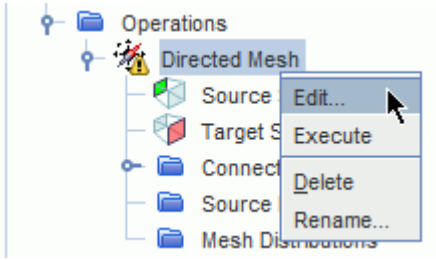


Patch meshing uses a patch *edit* panel. The panel consists of two modes:

- Creating a patch topology
- Defining the quadrilateral mesh that STAR-CCM+ generates within patches.

To create a patch mesh:

1. Enter the directed mesh edit panel by right-clicking **Operations** > **Directed Mesh** and selecting **Edit**.

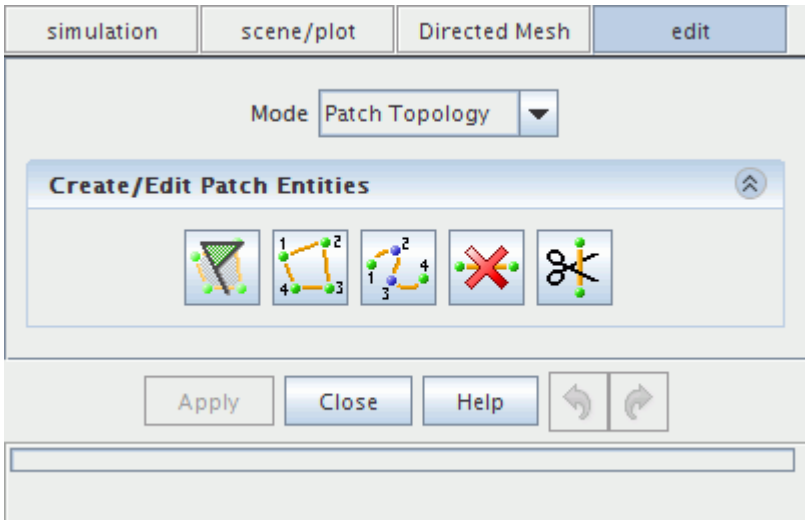


2. In the *Directed Mesh* panel, right-click the **Source Meshes** node and select **New Source Mesh** > **Patch Mesh**.

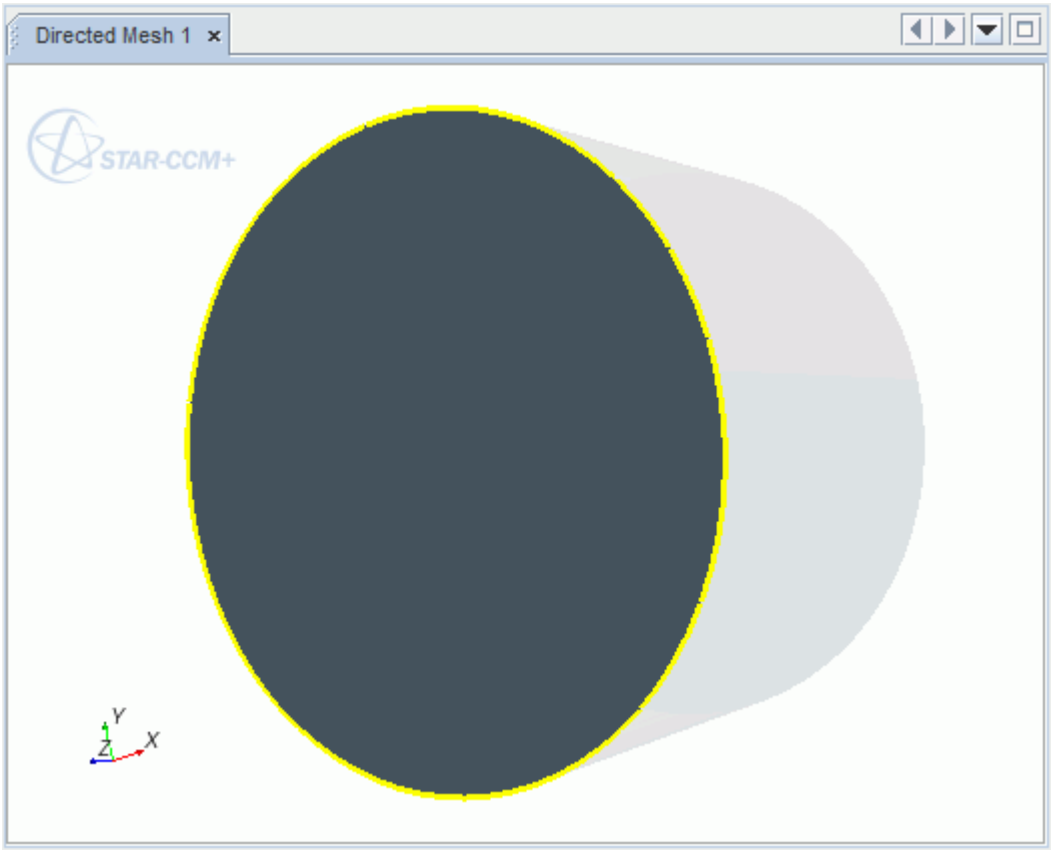
   Note: Make sure that source and target surfaces are already specified before creating the source mesh.

3. In the *Part Collection For This Source Mesh* dialog, select the connected parts on which you are going to create the patch mesh and click **OK**.

Doing so opens an *Edit* panel that is used to create patch curves:



The *Graphics* window updates to show the selected parts. The source surface on which you create the patches appears gray and the remaining surfaces are translucent for reference. Feature curves appear yellow.



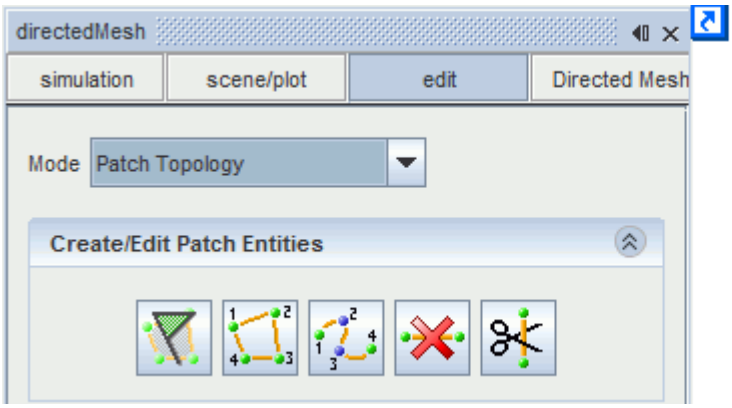
Contents:

Creating the Patch Topology
Undo and Redo in Patch Topology Mode
Creating Quadrilateral Meshes on Patches

Creating the Patch Topology

You use the tools in the Patch Topology mode of the patch mesh *Edit* panel to create patches on the source surface.

By default, the patch mesh *Edit* panel opens in this mode.



The available buttons in the *Create/Edit Patch Entities* box are used to create, delete, and split patch curves. The buttons in this panel are “sticky” buttons. That is, once pressed, they stay activated until another button is pressed or the `<Esc>` key is pressed to cancel the action.

You can undo or redo any of these actions with the undo and redo buttons at the bottom of the panel.

Contents:

Creating a Patch
Specifying Exact Patch Vertex Location
Moving Patch Vertices
Splitting and Deleting Patch Curves

Creating a Patch

This section describes how to create patch curves.

There are three ways to create patch curves using the patch entities box:

- Auto-populate feature edges with patch curves
- Create patch curve using straight edges
- Create patch curve using construction points

These three methods can be used alone or in combination.

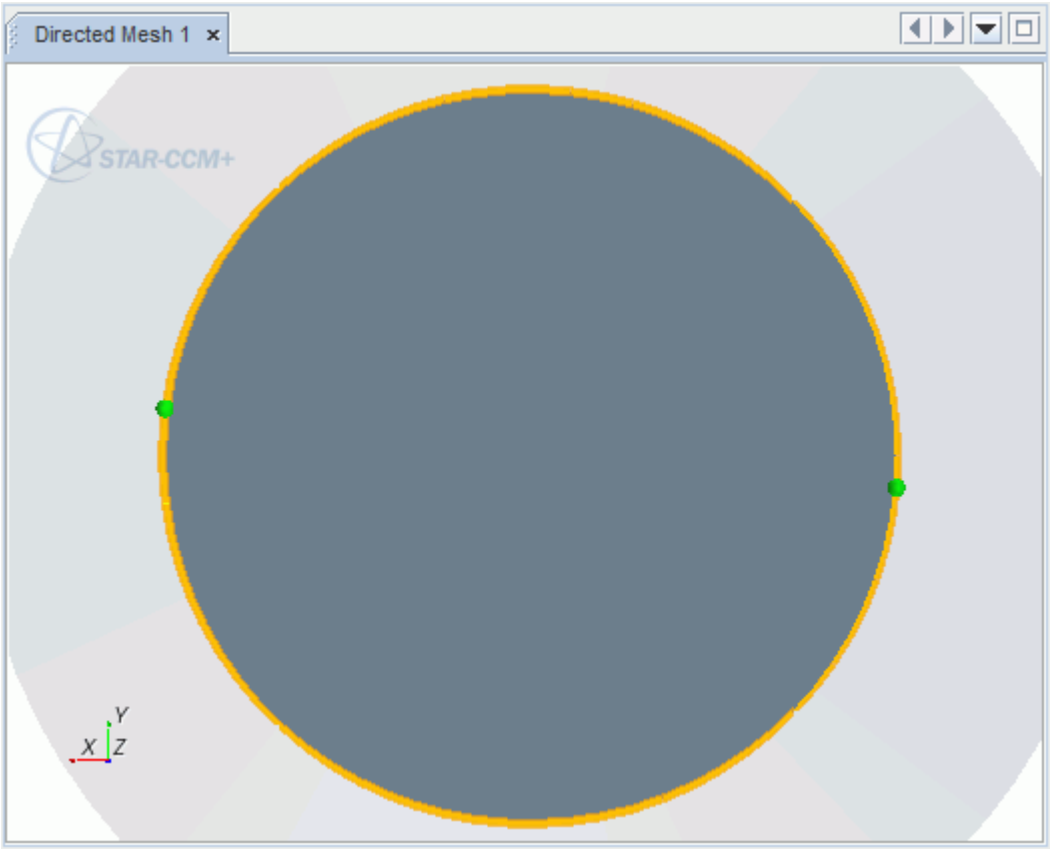
Initially, the patch mesh is created on the tessellated surface and then upon meshing, the patch mesh is projected on to the corresponding CAD surface.

Extracting Patch Curves from Feature Edges

To create a patch curve on the feature edges:

1. Click the (**Auto-populate feature edges with patch curves**) button to create patch curves on the feature edges of a source surface. It also places patch curve vertices at the locations of feature vertices.

   If the feature edge is an ellipse, STAR-CCM+ places two vertices on opposite sides of the curve, resulting in two patch curves.



Contents:

Creating Patch Curves with Straight Edges
Creating Patch Curves with Constructions Points
Changing the Patch Curve Focus

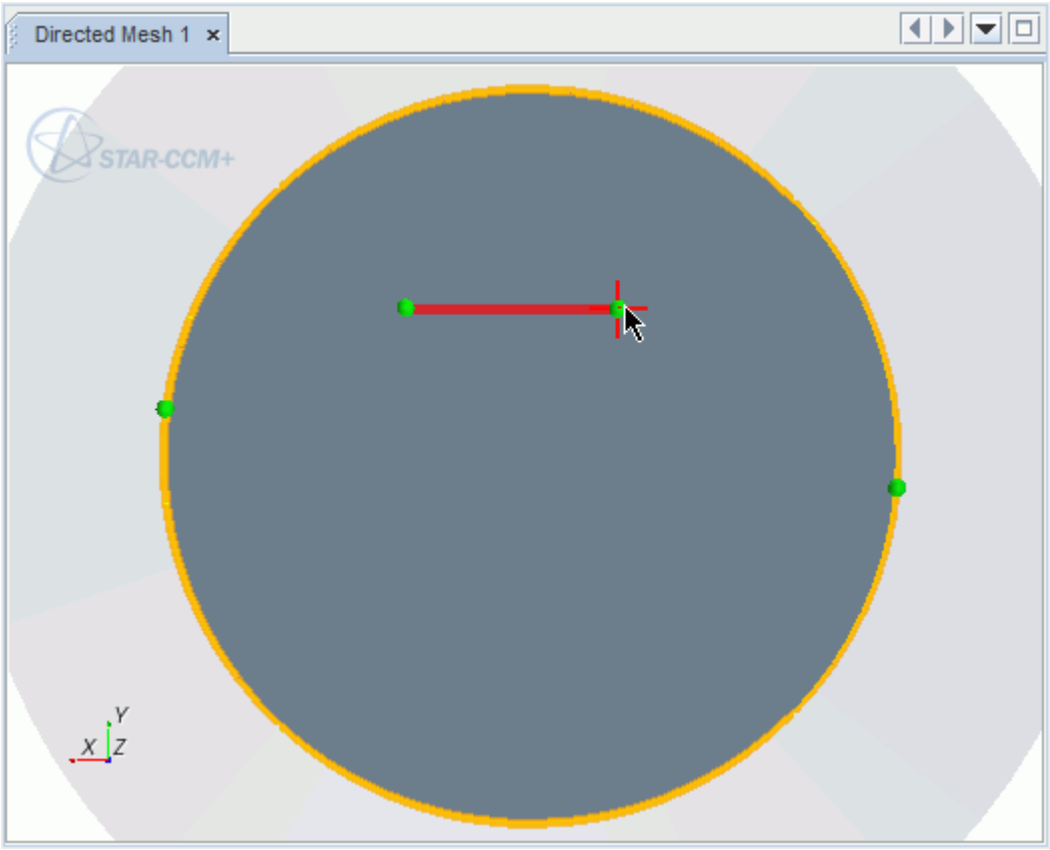
Creating Patch Curves with Straight Edges

You can create a patch curve as a straight edge between two vertices.

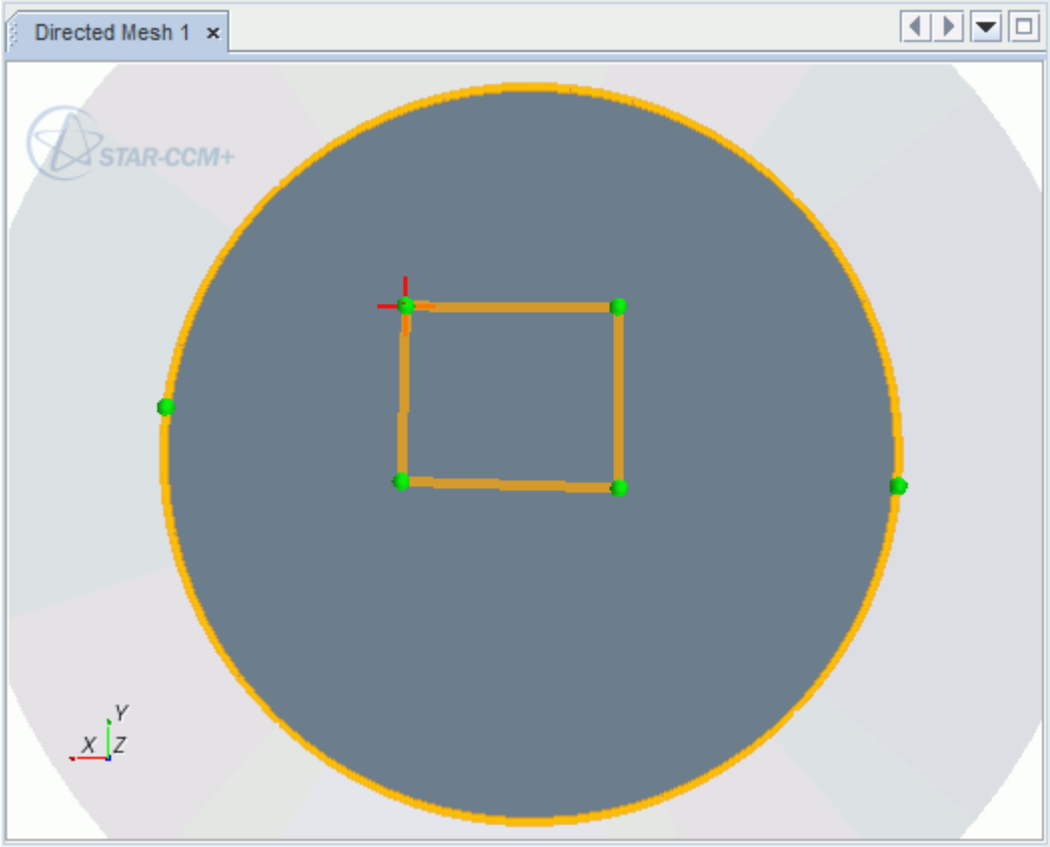
To create a patch curve out of two vertices:

1. Click the (**Create patch curve**) button and move the mouse anywhere over the source surface.
2. Click the location of the surface where you want to put the first patch vertex point.

   A green dot represents the vertex point. Click again where you wish to put the second patch vertex. A red curve is drawn between the two points. The most recently placed vertex has the focus.



3. To close the loop, create three more patch curves.



Creating Patch Curves with Constructions Points

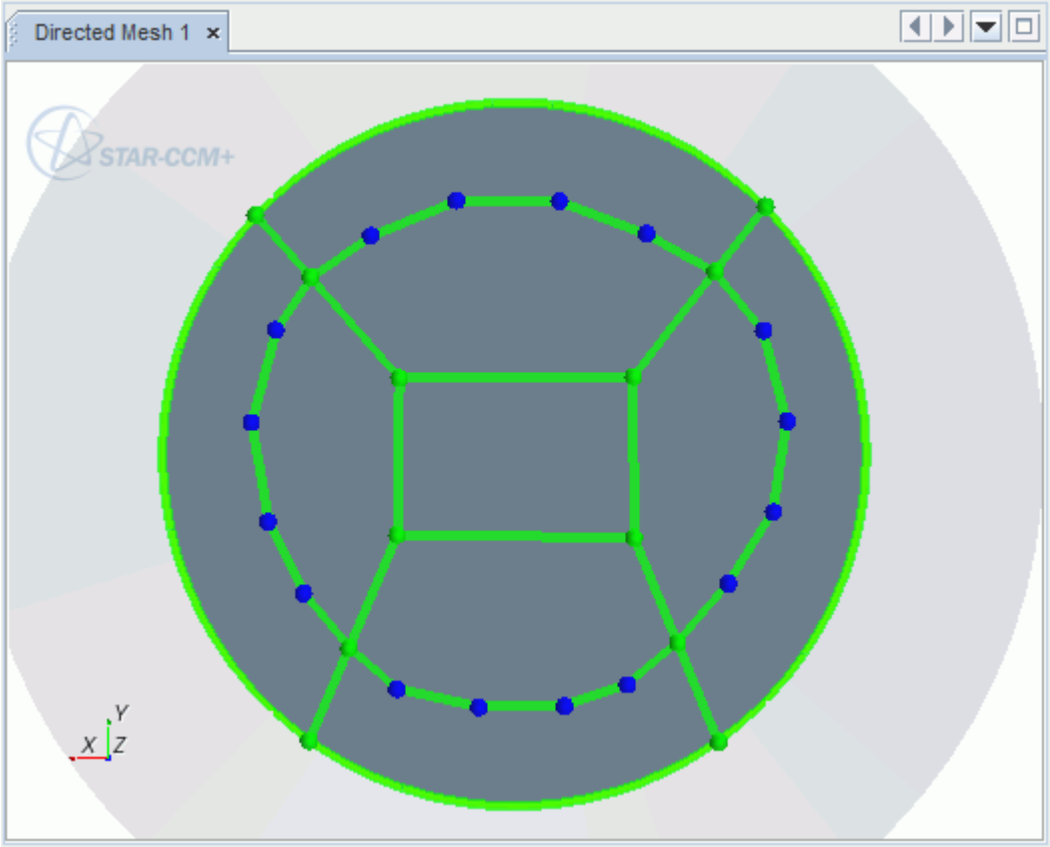
You can create a patch curve that is not straight.

To create a patch curve that is not straight:

1. Click the (**Create patch curve with construction points**) button and move the mouse anywhere over the source surface.
2. Click the location of the surface where you want to put the first patch vertex point.

   This vertex point is a normal patch vertex. Continue adding points along a curve you want to make. These subsequent points are in blue to represent the curve construction points.
3. To end the curve, use one of the following options:
   - Click a patch vertex (green point) from another curve.
   - Right-click the last construction point that you made for the curve. This point is then converted to a regular patch vertex.
   - Press `<Esc>` to convert the last construction point into a regular patch vertex.

   The following patches are created using the patch curve tools in combination with the split curve and move patch vertex features.



Changing the Patch Curve Focus

At some point after creating one or more patches, the focus is no longer in a usable location.

To change the focus of the patch curve:

1. Right-click the patch vertex where you want the new focus point to be.
2. To clear the focus, click anywhere outside of the source surface, and then click again on the patch vertex to set the new focus location.
3. Press <Esc> to cancel the patch creation process and then press the **Create Patch Curve** button again to start a new patch in any desired location.

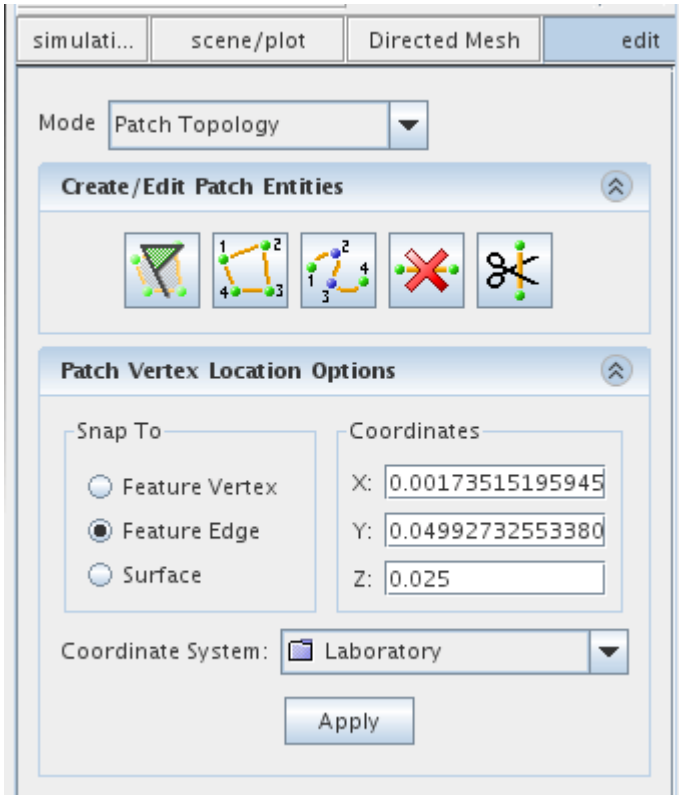
Specifying Exact Patch Vertex Location

You can move patch vertices by defining the exact coordinates of the new location.

To specify an exact patch vertex location:

1. Press <Esc> to cancel any current patch operations. Doing so removes the focus.
2. Click any existing patch vertex.

   The *Patch Vertex Location Options* appears in the patch mesh *Edit* panel.



3. The Snap To panel identifies the location of the patch vertex:
   - Feature Vertex (grayed out if none exist in the source mesh)
   - Feature Edge
   - Surface
4. The Coordinates panel displays the current x, y, z coordinate location of the patch vertex. Enter the specific coordinates for the patch vertex.
5. Select the Coordinate System.
6. Click **Apply**.

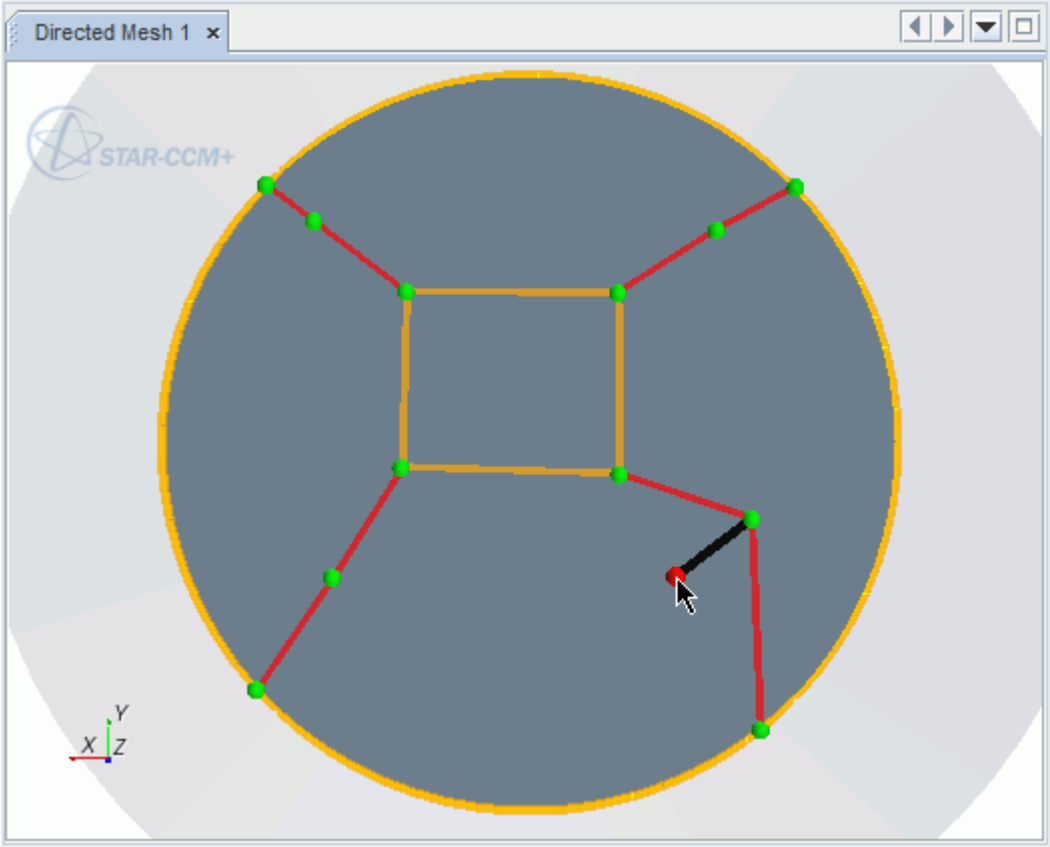
   The patch vertex moves to the specified location.

Moving Patch Vertices

You can move patch vertices or curve construction points to new locations.

To move a patch vertex:

1. Press `<Esc>` to cancel any current patch operations. Doing so removes the focus.
2. Click and drag the patch vertex that you want to move to its new location.



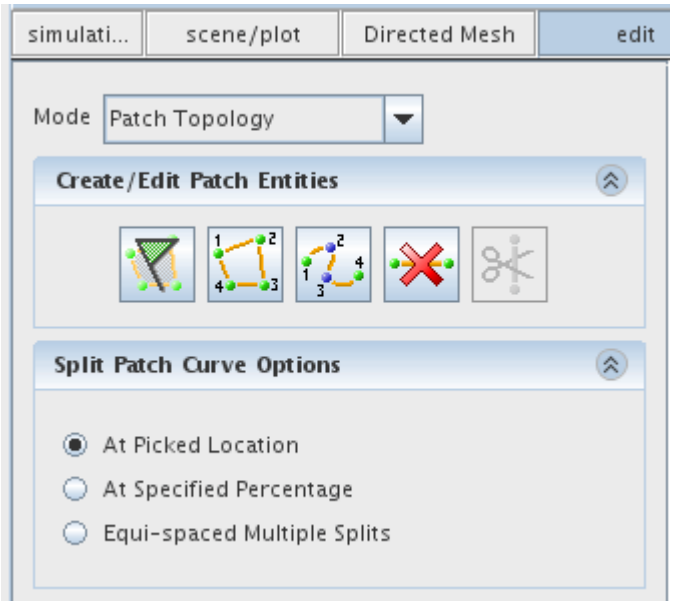
Note: Moving a vertex to the same location as another vertex could invalidate the patch topology. STAR-CCM+ removes any zero-length patch curve.

Splitting and Deleting Patch Curves

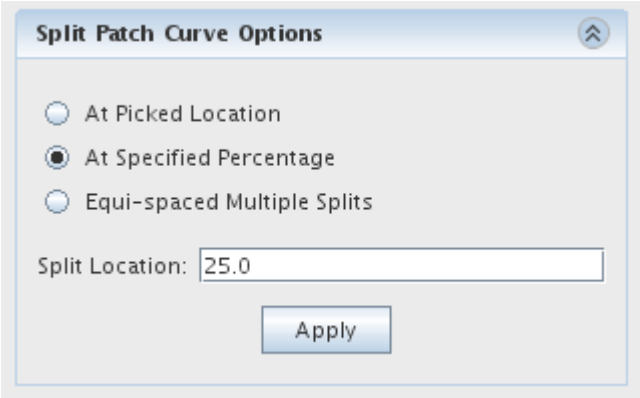
If you want to create a patch curve that crosses another patch curve, split the curve first.

To split a curve:

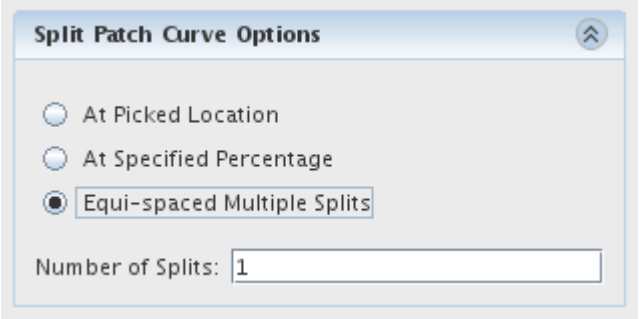
1. Click the (**Split the selected patch curve**) button. The *Split Patch Curve Options* appears in the patch mesh *Edit* panel.



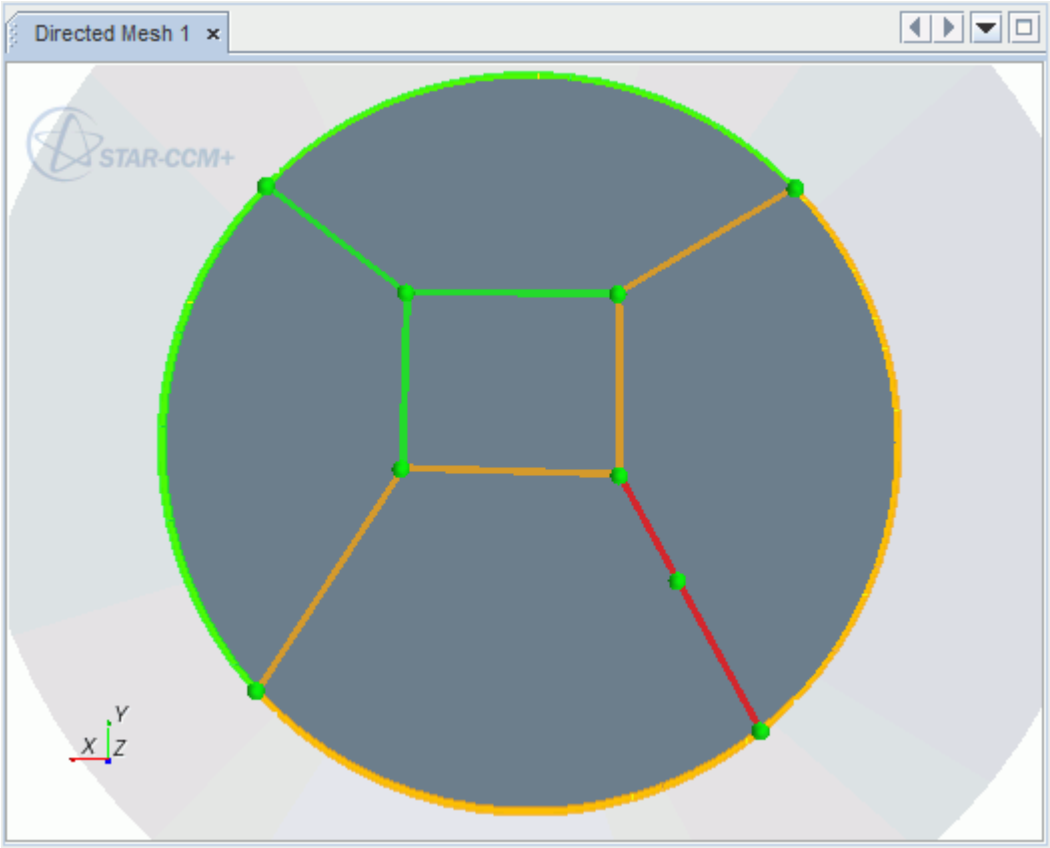
2. To split a curve at a picked location, click *At Picked Location* and then click the curve location where you want the split to happen.
3. To split a curve at a specified percentage, click *At Specified Percentage*. Enter the percentage and click **Apply**.



4. To split a curve into an equal number of splits, click *Equi-spaced Multiple Splits*. Enter the Number of Splits and press `<Enter>`.



Note: Splitting a curve into two usually results in a topologically incorrect set of patches. In the following example, splitting a shared curve results in five sided patches.



To delete a patch curve:

5. Click the (**Delete the selected patch curve**) button and click any curve that you want to delete.
   Deleting a patch curve with construction points deletes the entire curve and not just the clicked segment. Removing a patch curve could cause the patch definition to become topologically incomplete.

Undo and Redo in Patch Topology Mode

When creating or editing the patch topology, you can undo and redo actions using the undo and redo buttons at the bottom of the panel.

The undo tool reverses the previous action

The redo tool repeats a previously undone action

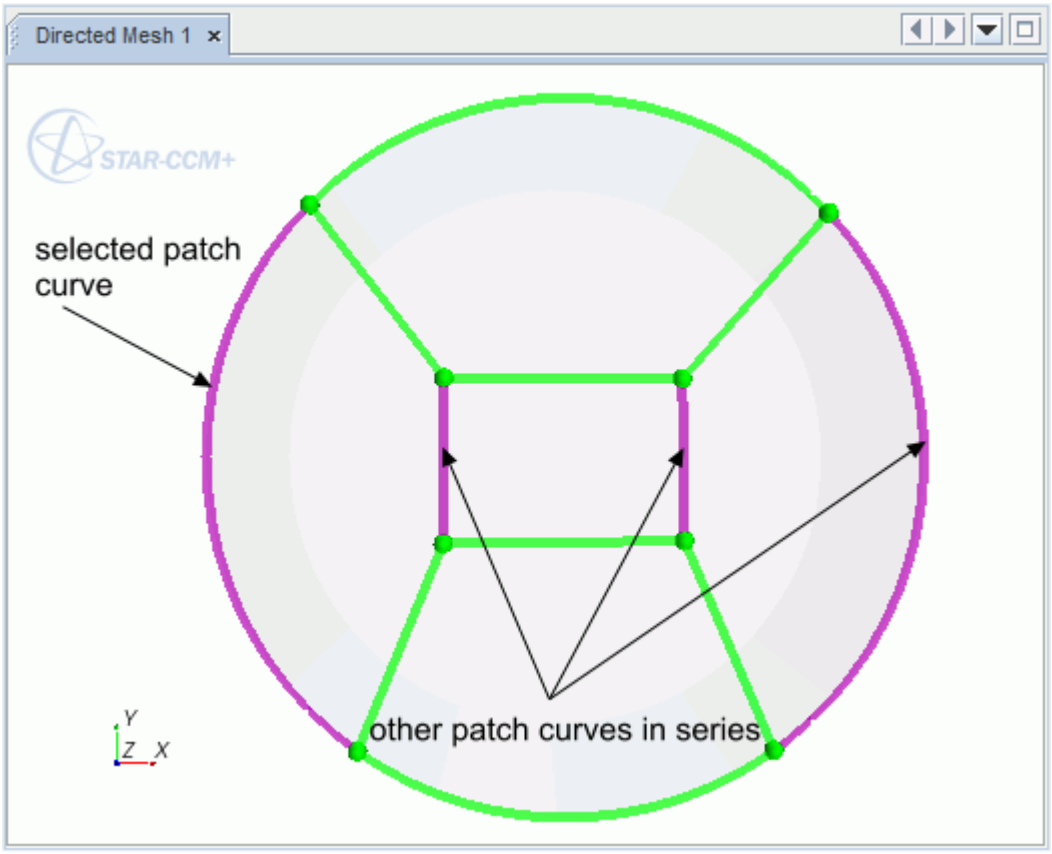
Hovering the cursor over the button indicates the action that the button reverses or repeats.

Note: If you change between modes (from **Patch Topology** to **Patch Mesh**) or close the *edit* panel the undo and redo memory is lost.

Creating Quadrilateral Meshes on Patches

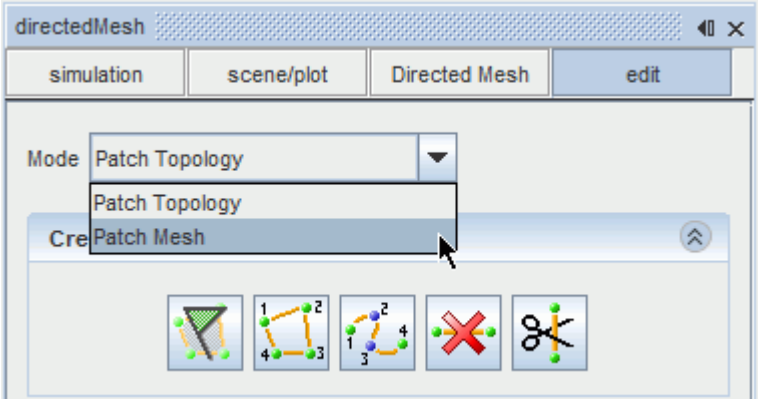
This section describes how to create quadrilateral meshes on patches.

The **Patch Mesh** mode in the *Edit* panel is activated once the patch topology is complete, that is, all the patch curves are green. The patch mesh mode is used to define the mesh properties for the patch curves, including the number of divisions on each mesh curve, stretching distribution and patch mesh smoothing. STAR-CCM+ applies the same mesh properties to a selected patch curve and its corresponding mapped curves. Mesh curves for which the mesh properties are defined appear in black. Once all curve series have a defined mesh distribution, STAR-CCM+ automatically constructs and displays the mesh.

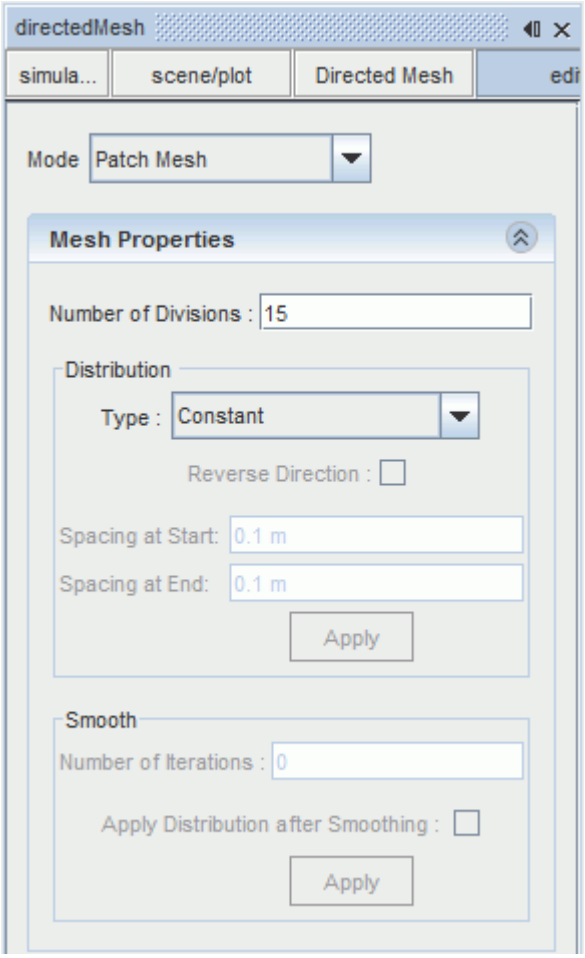


To create a patch mesh:

1. In the *Edit* panel, select the Patch Mesh mode.



This mode contains the *Mesh Properties* box.



2. Select the patch curve for which you want to set mesh properties.

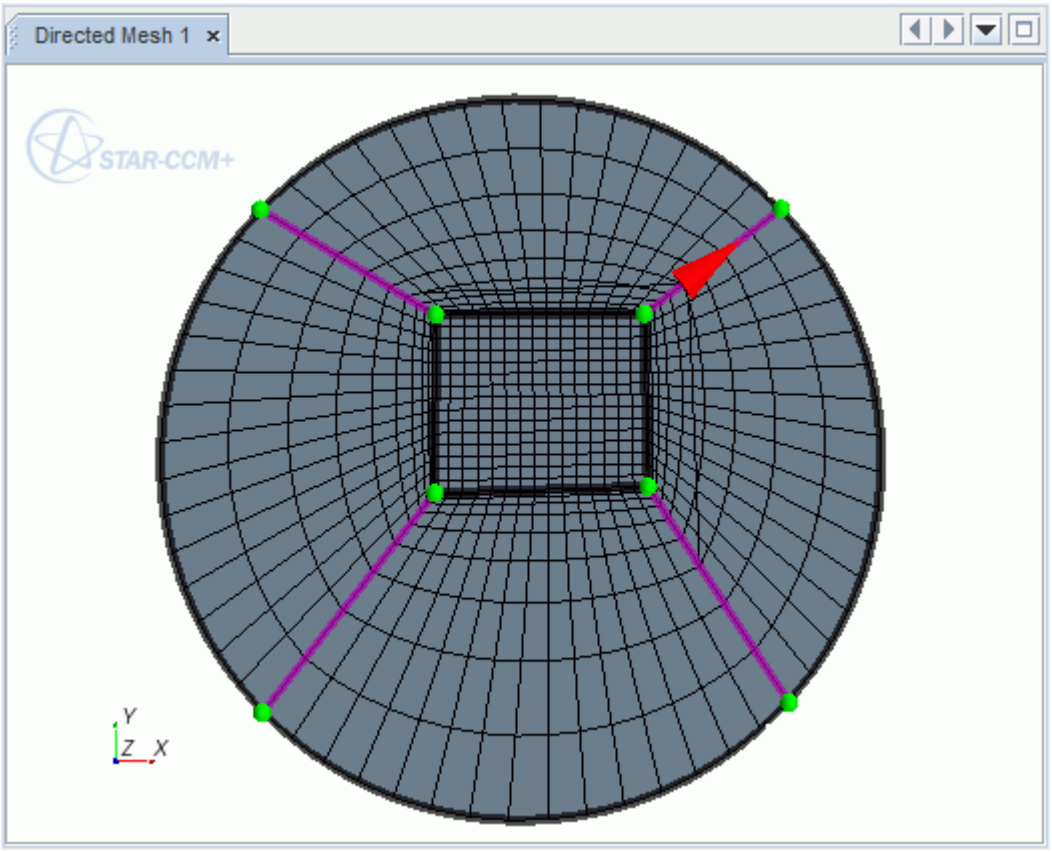
   STAR-CCM+ automatically selects all associated curves. Also, you can select more than one base curve, and STAR-CCM+ automatically includes all the associated curves.

3. In the *Mesh Properties* box, set the **Number of Divisions** to the desired mesh division amount.

   After entering this value, press `<Enter>`.

4. In the *Distribution* group box:
   1. Set the type of stretching for the mesh:
      - **Constant** — The space between divisions is equal.
      - **One Sided Geometric** — The space between divisions increases in one direction with a geometric function. Specify the spacing at the start of the stretching.
      - **One Sided Hyperbolic** — The space between divisions increases in one direction with a hyperbolic function. Specify the spacing at the start of the stretching.
      - **Two Sided Hyperbolic** — The space between divisions increases in both directions with a hyperbolic function. Specify the spacing at the start and end of the stretching.
   2. To reverse the stretching direction, activate the **Reverse Direction** checkbox.
   3. Click **Apply** to apply the distribution settings and generate the patch mesh.

   The direction of the stretching is shown in the *Graphics* window with a red arrow. The following is an example of using the distribution tool with the surface mesh.

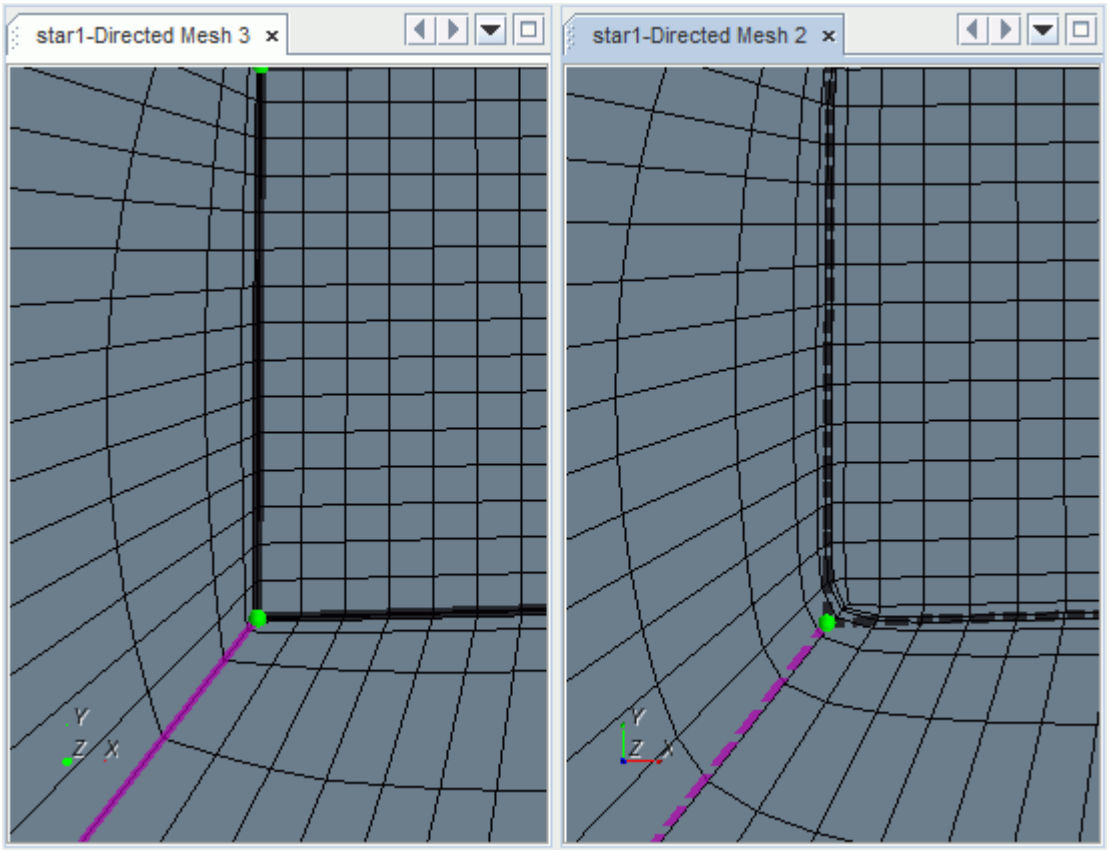


5. In the *Smooth* group box:
   1. To smooth the patch mesh when the patch topology is distorted or uneven, set the **Number of Iterations**.
   2. Activate the **Apply Distribution after Smoothing** box when the smoothing process distorts the selected stretching function.
   3. Click **Apply** to apply the smoothing settings and generate the patch mesh.

STAR-CCM+ applies the smoothing after the patch mesh is generated. Patch vertices are moved from their original position so that they no longer lie on the patch curves. The original vertex locations of are still shown as are the original curves. Dashed lines represent the original curves.

Note: If you increase the number of iterations, you achieve greater smoothing at a greater computational effort. The value of smoothing iterations is absolute and not cumulative from previous iteration values.

In the following screenshot, the mesh on the right shows smoothing of the mesh on the left by moving the vertices.



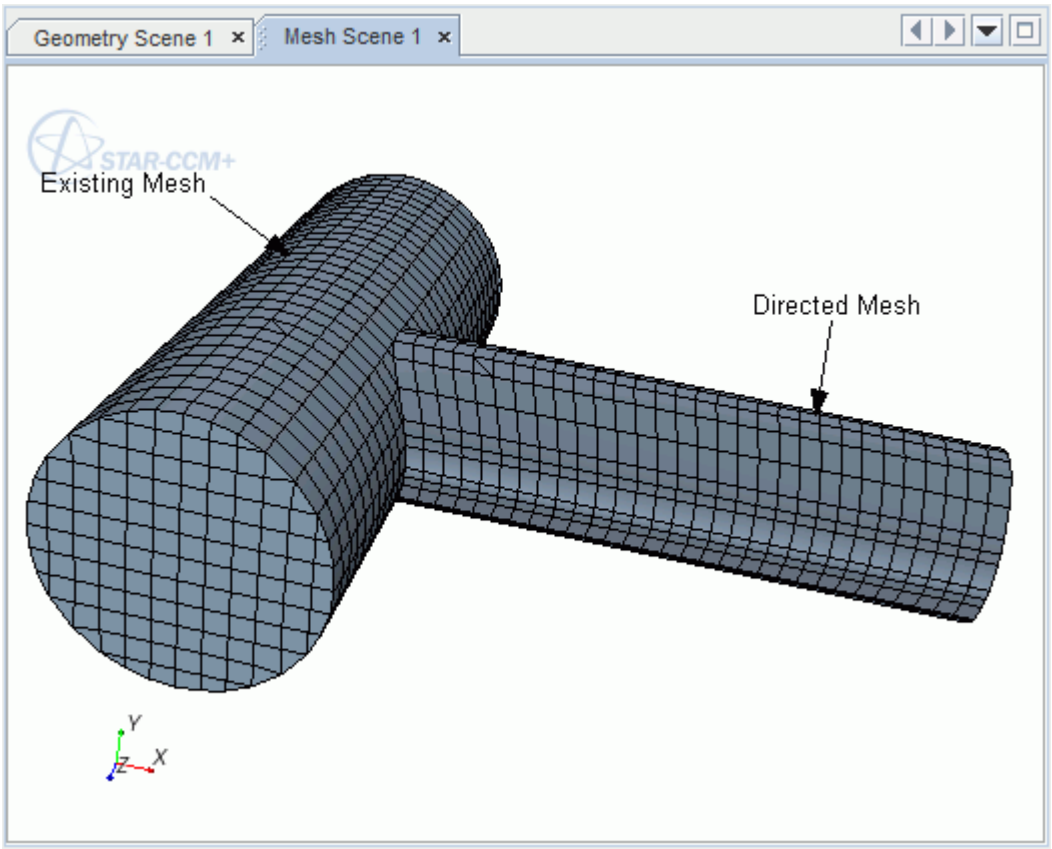
If you want to change the patch layout after creating the patch mesh, change to *Patch Topology* mode. Any changes that you make to patch mesh properties are automatically applied to the mesh shown in the display.

A new **Patch Mesh** node is added under **Source Meshes** once you close the patch mesh *Edit* panel. If you right-click the new node and select **Edit...**, you can return to the *Edit* panel and review or edit the patch definition.

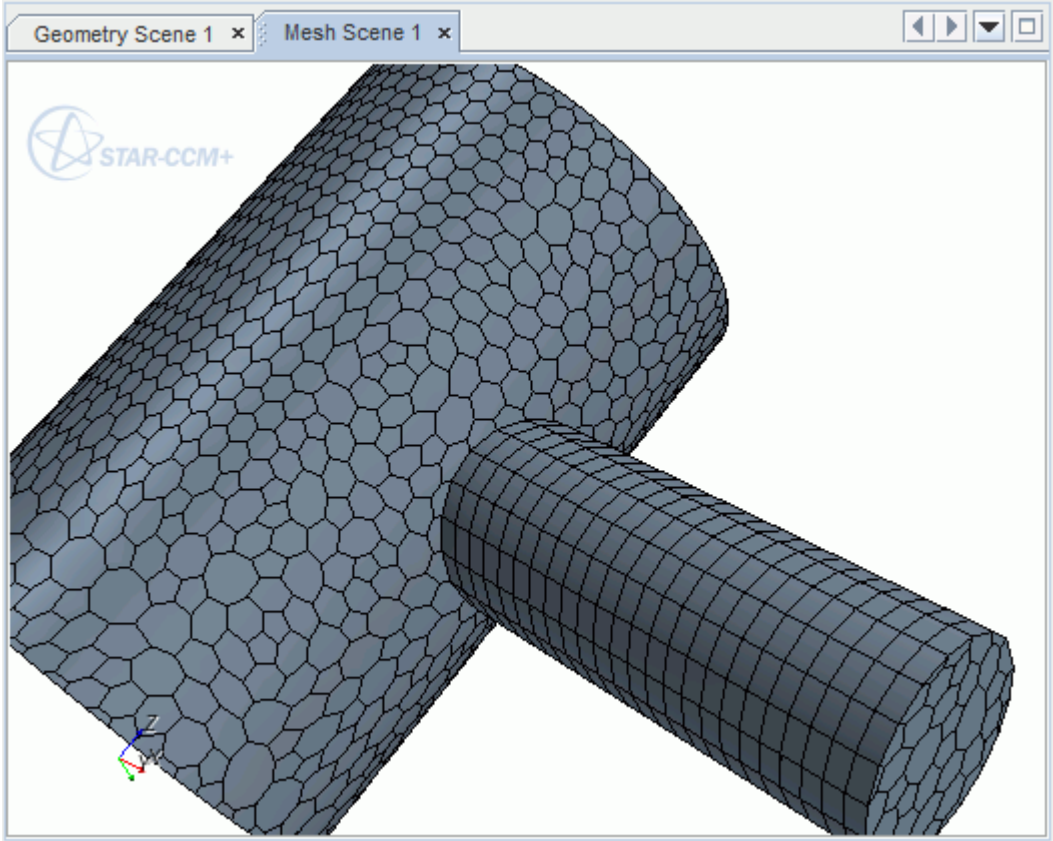
Creating a Source Mesh Using an Existing Mesh

It is possible to use an existing volume mesh as the source surface mesh.

The resulting directed mesh is conformal at the interface between the two volume meshes.



You can map an unstructured mesh to the source surface.



The following requirements must be met before an existing mesh can be extracted for the directed meshing part:

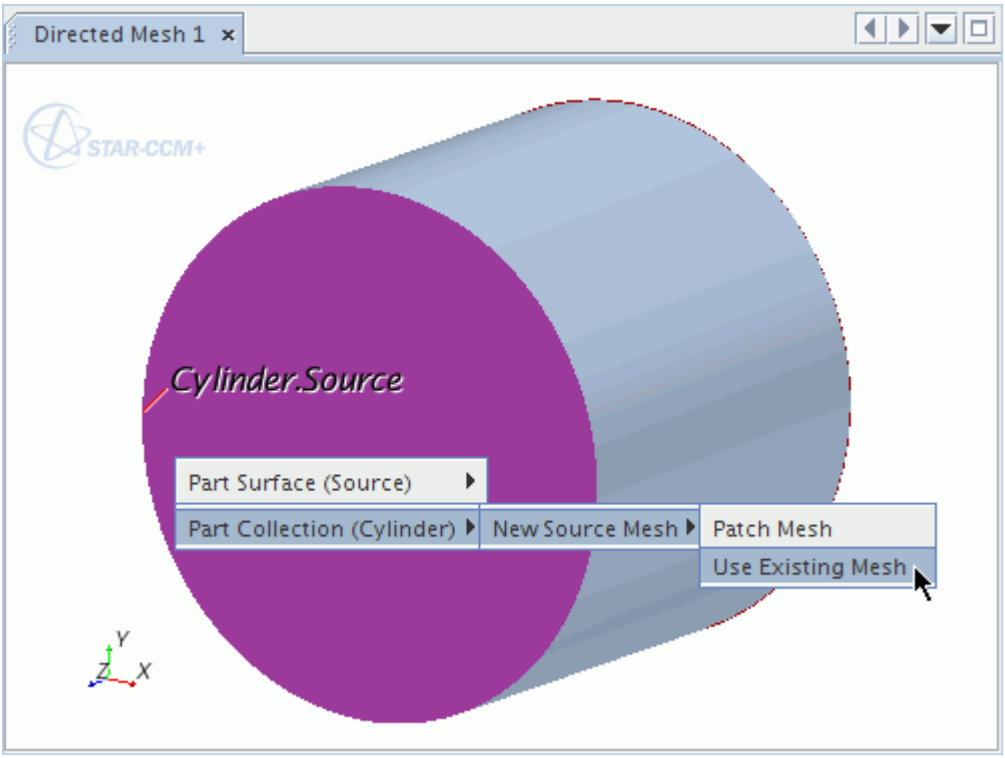
- The directed meshing part must be CAD imprinted to the existing volume mesh part so that an in-place part contact is created.
- If you are connecting to multiple existing volume meshes, all the existing meshes must belong to the same mesh operation or mesh continuum.
- Where the existing volume mesh and directed mesh part surfaces are coincident, the part surface boundaries must match up and share edges. You can use the split by patch option to separate the necessary part surfaces.
- Create a separate part surface for each side of the interface.

  See Splitting A Part Surface By Patch

Note: It is not possible to use an existing 2D mesh that is created outside of the Directed Meshing tool as a source surface mesh for directed meshing. However, you can create a 2D source surface mesh specifically for Directed Meshing. See Creating a 2D Source Mesh.

To create a source mesh using a previously defined volume mesh:

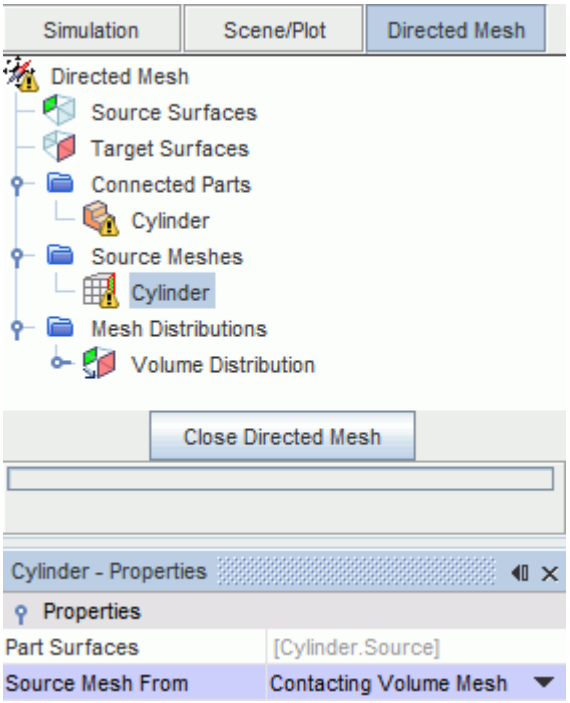
1. In the *Graphics* window, select one or more source surfaces.
2. Right-click one of the source surfaces and select **Part Collection** > **New Source Mesh** > **Use Existing Mesh**.



Alternatively:

1. Right-click the **Source Meshes** node in the *Directed Mesh* panel and select **New Source Mesh** > **Use Existing Mesh**.
2. In the *Part-collection for this source mesh* dialog, select one or more parts for the source mesh.

A **Source Meshes** > **[Existing Mesh]** node appears. Each source mesh node is named after the contacting part collection which the source mesh is defined on.



The properties of the **[Existing Mesh]** node contain the Part Collections that are selected for this distribution. See Source Meshes – Existing Mesh Properties.

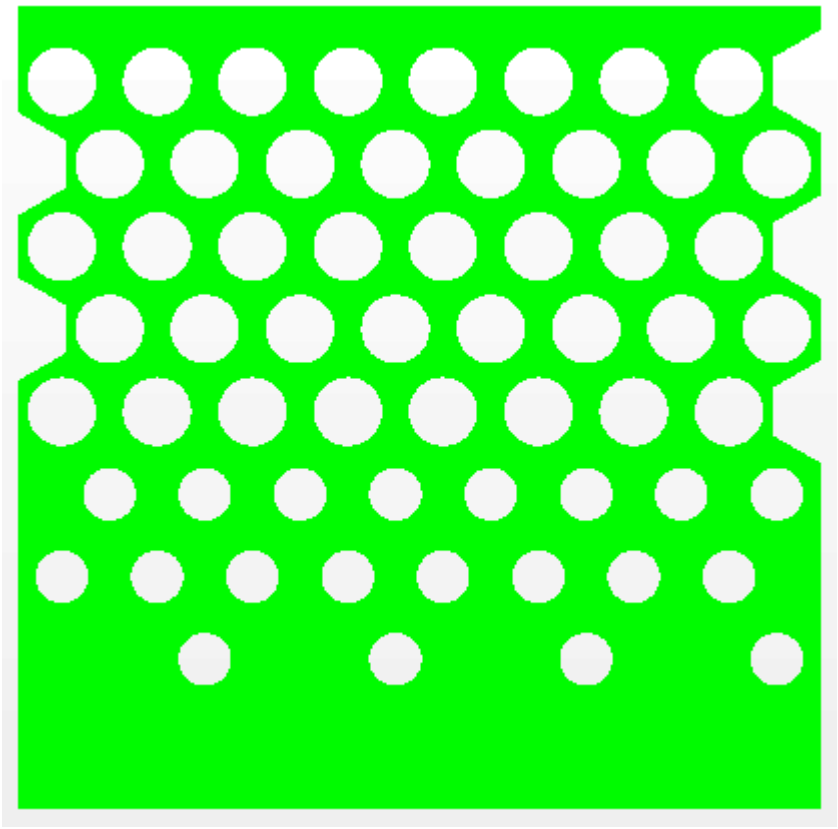
Note: If the *Source Mesh From* option is changed to **Quadrilateral Mesh**, the dirty flag is set on the **[Existing Mesh]** node, the corresponding Part collection, and the Directed Mesh operation.

Updating the Source Mesh can be accomplished using one of the following options:

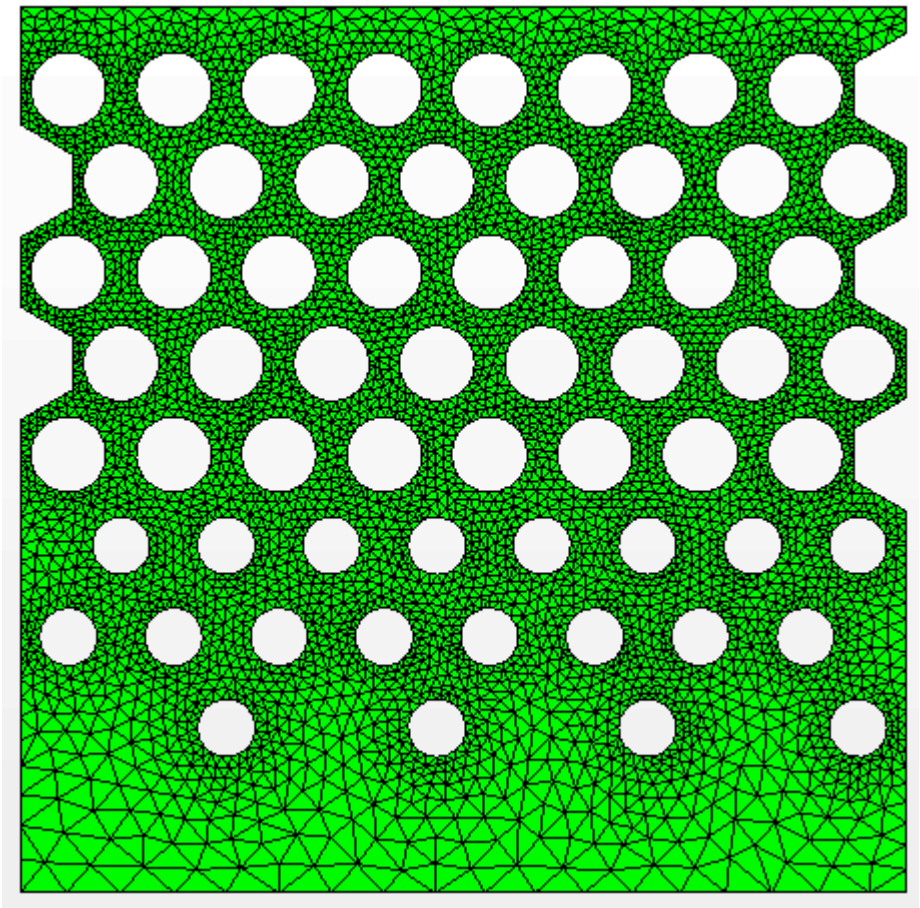
- Right-click the **[Existing Mesh]** node and select **Update**.
- Right-click the **[Part Collection]** and select **Generate Volume Mesh**.
- Right-click the **Directed Mesh** operation and select **Execute**.

When using the Automatic source meshing, all parts in the source surfaces have remeshed surfaces. If they do not, then the initial surface for any non-remeshed part is used as the source surface for all the parts in the collection.

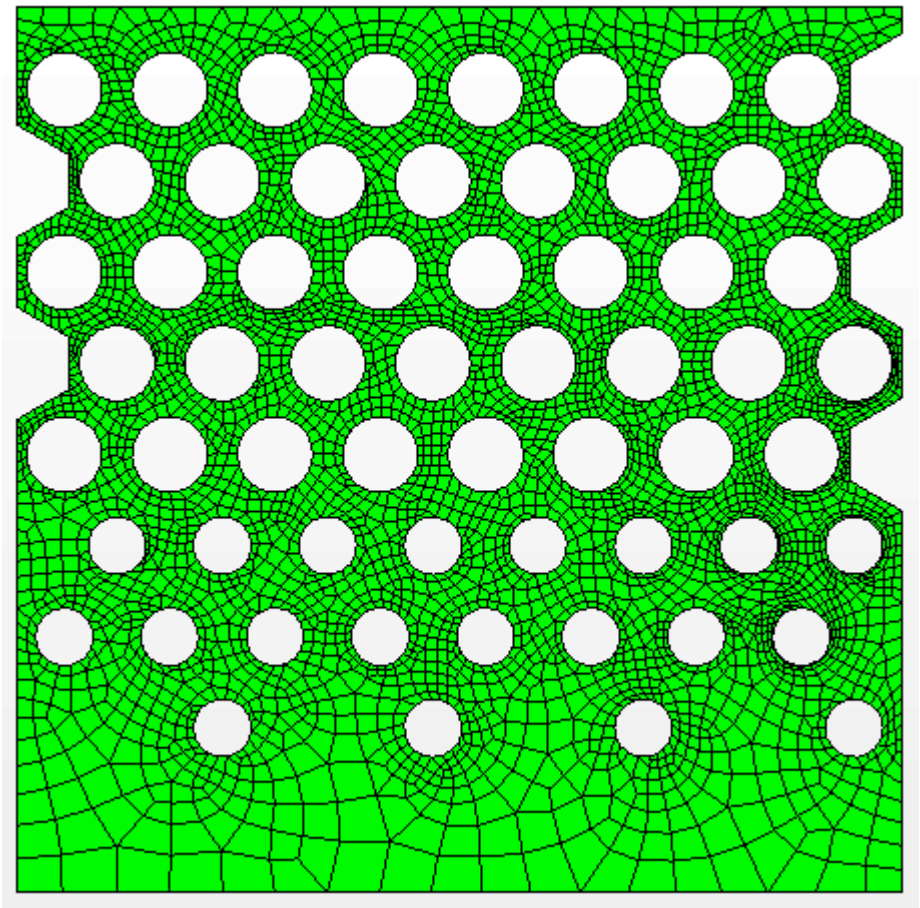
Initial Source Surface



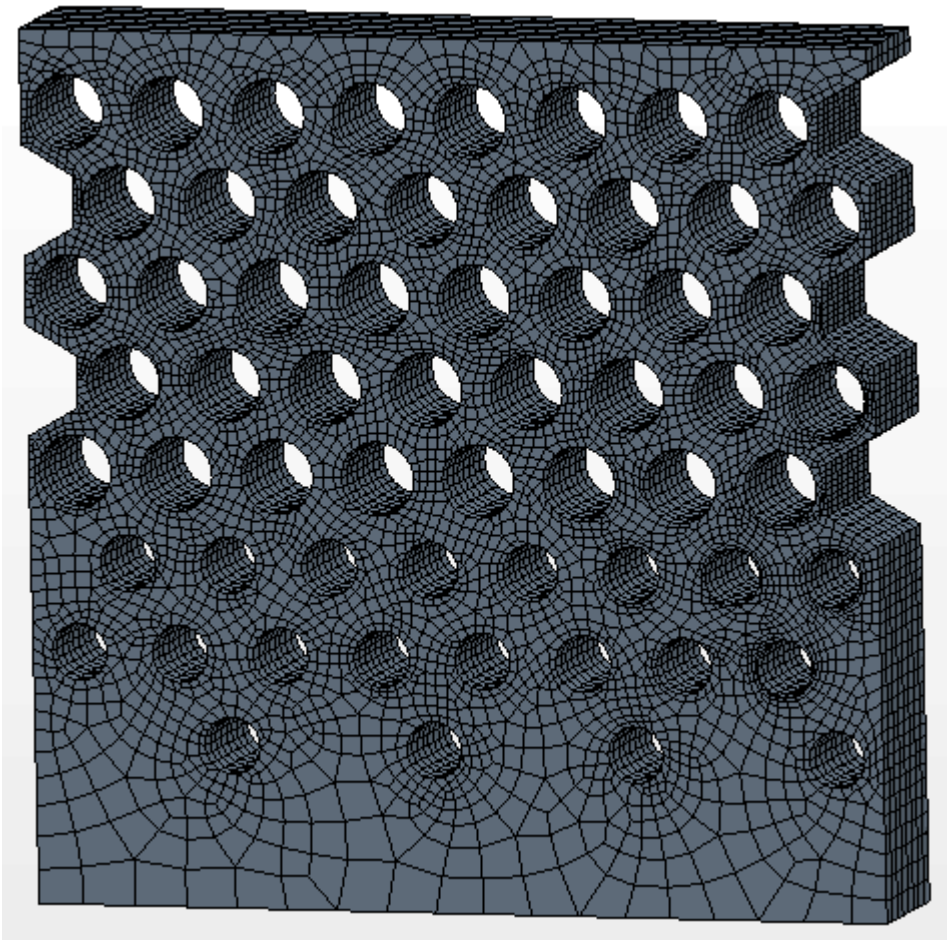
Extracted Triangle Mesh



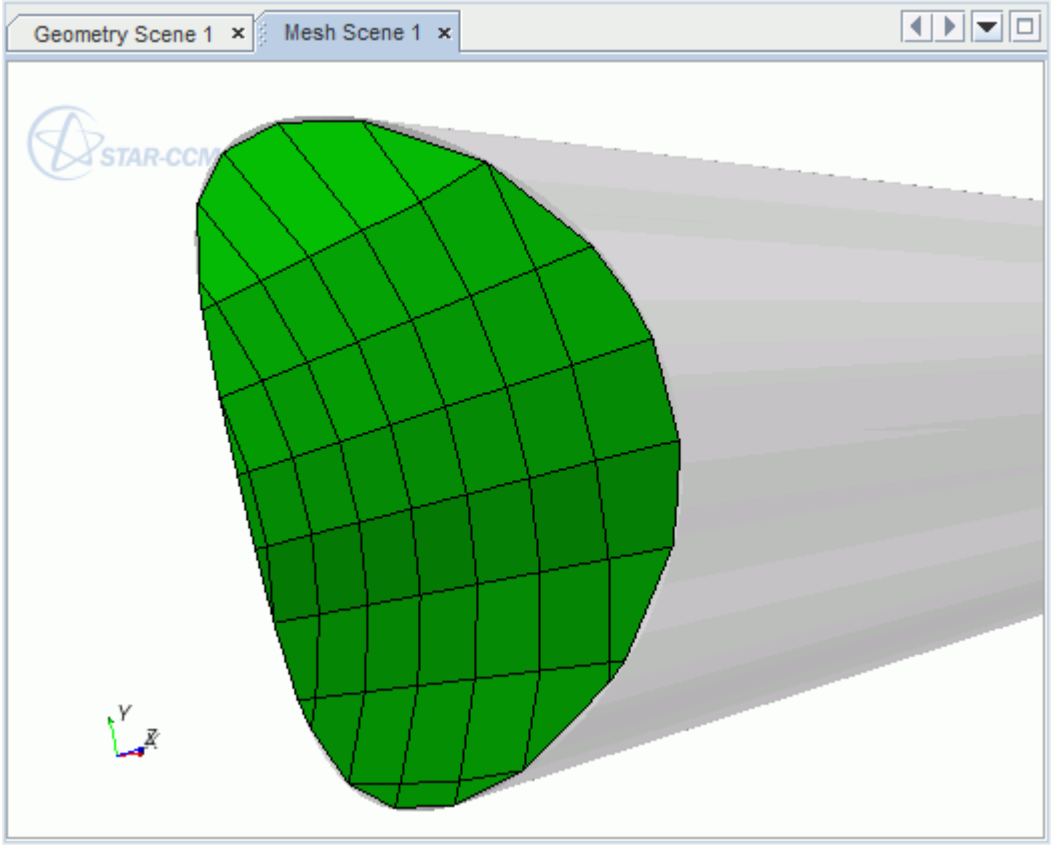
Quad/Quad-Dominant Mesh obtained from triangulated mesh



Updated Volume Mesh



STAR-CCM+ maps the existing mesh to the source surface.



Creating a 2D Source Mesh

When using Directed Meshing, you can create 2D source surface meshes using the automated 2D mesh operation. This approach allows you to include prism layers on the source surface, which are then swept along the length of the connected parts.

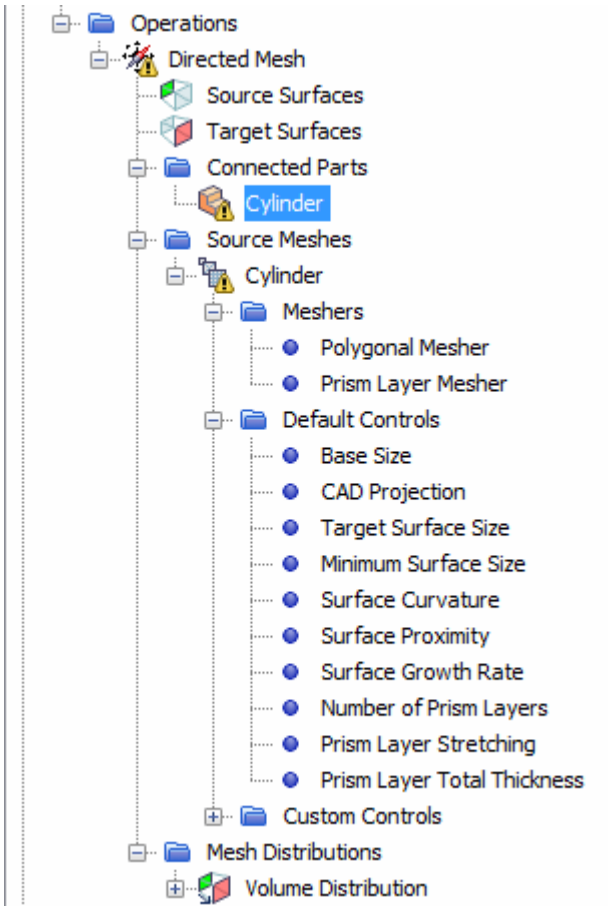
The 2D source surface mesher behaves in the same way as the automated mesh operations that are available in the main simulation tree, allowing you to update the Directed Mesh as part of a repeatable pipeline of parts-based mesh operations.

Before creating 2D source surface meshes, make sure that you have created a Directed Mesh operation and defined the Source faces and Target surface. The Source surfaces can be planar or non-planar.

You can create 2D source surface meshes within the Directed Mesh operation in the simulation tree, or in the *Directed Mesh* in-place edit dialog. This workflow specifies the latter.

1. Within the *Directed Mesh* in-place dialog, right-click the **Source Meshes** node and select **New Source Mesh** > **Automated Source Mesh**.
2. In the *Part Collection For This Source Mesh* dialog:
   a) Select the part collection on which to create the 2D source surface mesh.
   b) Select the necessary **Volume Meshers** and **Optional Volume Meshers**.
   c) Click **OK**.

A **[2D Auto Mesher]** node appears under the **Source Meshes** node which provides the same functionality as an Automated Mesh operation node. The 2D source surface Mesh Default Controls are comparable to the Surface Mesher Default Controls.

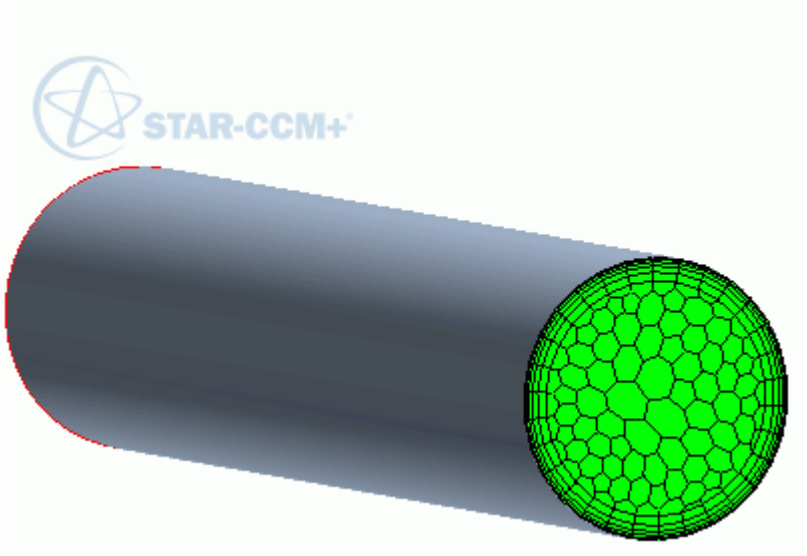


3. Define the parameters of the 2D source mesh. In particular, expand the **Default Controls** and set the **Base Size**. See Surface Mesher Controls And Values.

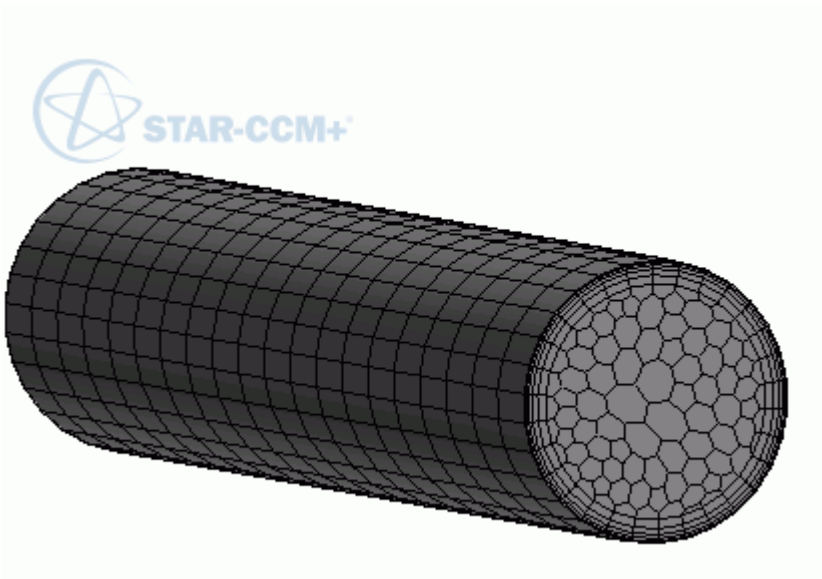
Within the Directed Mesh in-place edit dialog, you can create the 2D source surface mesh without executing the Directed Mesh operation. However, in the simulation tree, the 2D source surface mesh cannot be updated without executing the directed mesh operation.

4. In the *Directed Mesh* in-place edit dialog, right-click the **Source Meshes** > **[2D Auto Mesher]** node and select **Update**.

   The 2D source surface mesh is displayed in the display window. You can adjust the parameters of the 2D source surface mesh and Update it as necessary before executing the Directed Mesh operation.



When the Directed Mesh operation is executed, the 2D source surface mesh is swept from the source surface along the guide surface to the target surface. If prism layers are included on the 2D source surface mesh, prism layers are also created through the volume of the Directed Mesh.

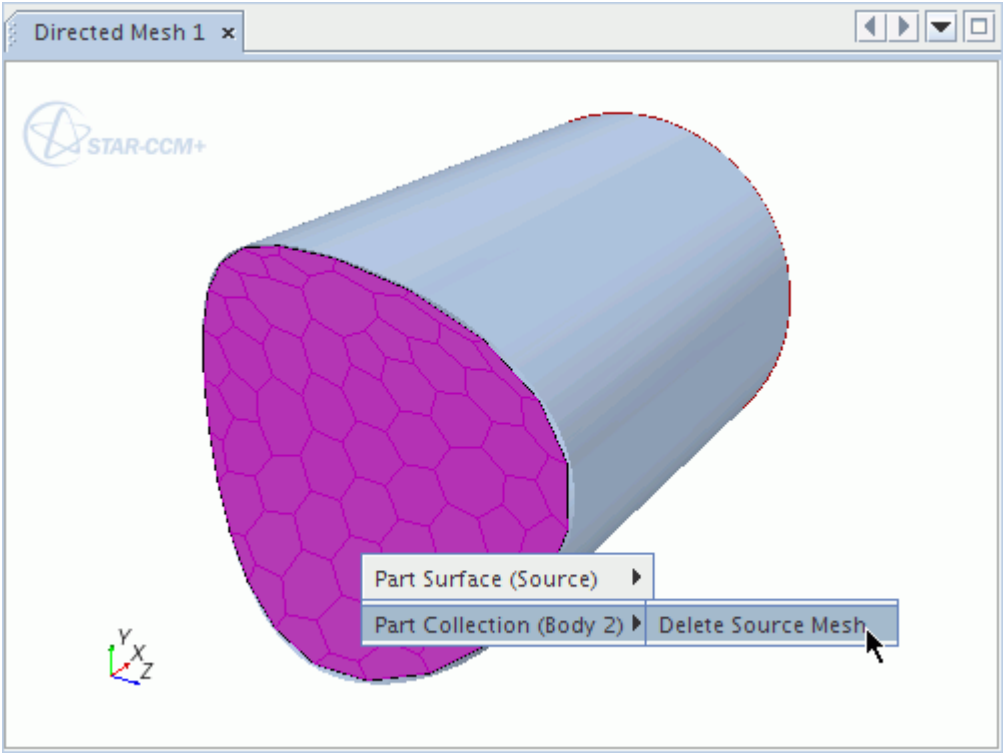


Deleting a Source Mesh

This section describes how to delete a source mesh from within the directed mesher menu.

To delete a source mesh:

1. Right-click the source mesh in the *Graphics* window and select **Delete Source Mesh**.

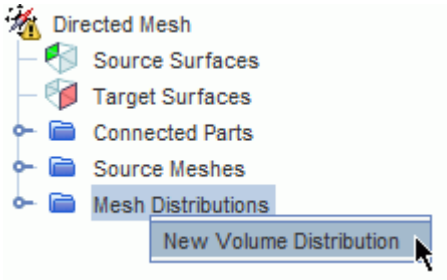


Creating a Volume Distribution

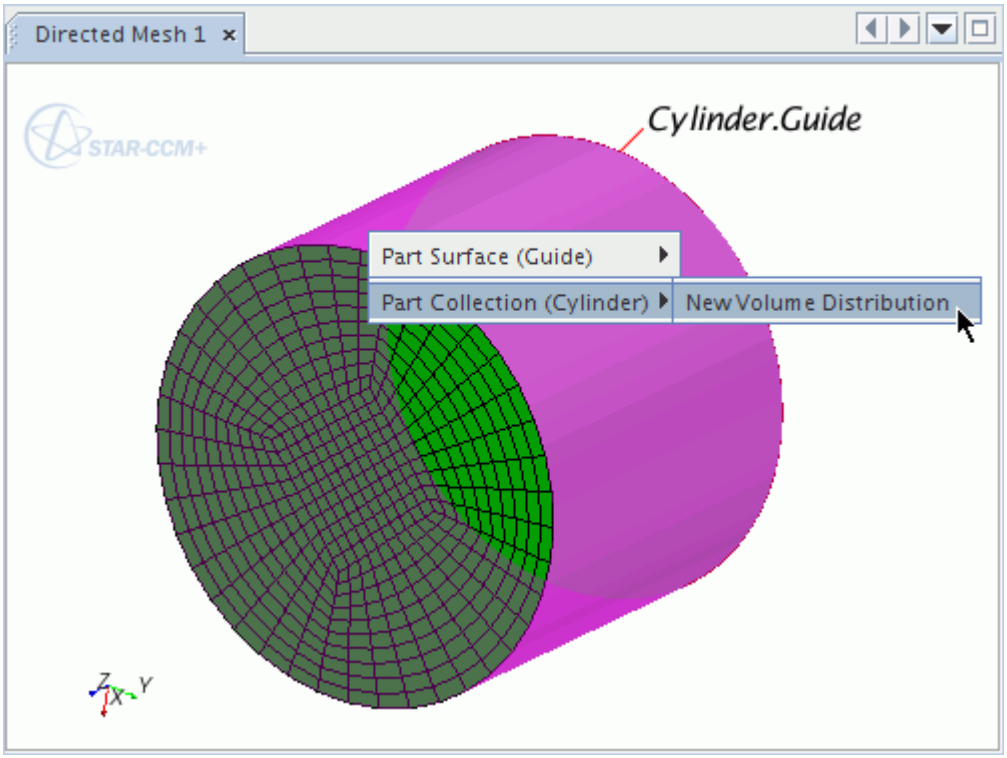
After you have specified a patch mesh (1-1 Directed Meshing only) or an existing mesh for the source surfaces, you assign a volume mesh distribution to the parts through which STAR-CCM+ sweeps the source. This distribution sets the number of cell layers that STAR-CCM+ creates throughout the volume, and how it computes their sizes.

To define the cell-layer distribution:

1. Create a volume distribution using one of the following approaches:
   - Right-click the **Mesh Distributions** node and select **New Volume Distribution**.



   - Right-click the guide surface in the *Graphics* window and select **Part Collection** > **New Volume Distribution**.



2. In the *Parts for creating a new mesh distribution* dialog, select the **Connected Parts** to which you want to apply the distribution. You can apply the same distribution to any number of connected parts, regardless of how the source surface mesh is defined.

   A new **Volume Distribution** node is added under **Mesh Distributions**.

You can choose between setting the number of layers and distribution for the part collection as a whole, or else applying specific settings to parts within the collection.

3. To set the layering properties for the whole part collection:
   a) Expand **Volume Distribution** > **Default Controls** and set the **Number of Layers** and **Stretching Function**.
4. To set the layering properties for one or more parts within a part collection:
   a) Select **Volume Distribution** and set *Mesh Controls* to **Specify Custom**.
   b) Right-click **Custom Controls** and select **New Part Control**.
   c) Select the part control and set *Parts* to include the parts on which the custom control applies.
   d) Within **Part Control** > **Controls**, select each value that you wish to customize and set its corresponding property to *Specify Custom Value*.
   e) Within **Part Control** > **Values**, select each value node and set its property.
   f) Select the parent **Volume Distribution** node and, within **Default Controls**, set the **Number of Layers** to an amount that exceeds the number of layers required by the part control.

Note: When you mesh a single part collection, with all parts connected, the default **Number of Layers** for a volume distribution must be greater than the sum of layers required by all part controls that the distribution contains.

5. To apply the volume distribution multiple times within the same part collection, select the **Volume Distribution** node and set **Number of Repetitions** to the amount you require.

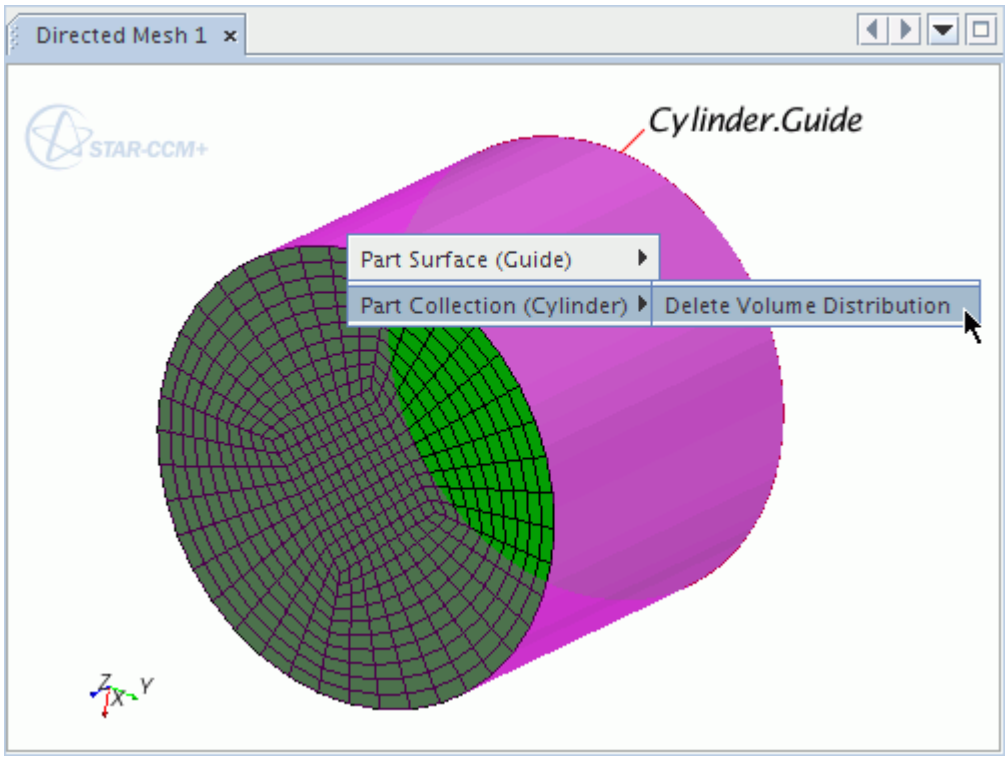
Note: Instead of producing an invalid directed mesh for some geometries when the barycentric mapping fails, STAR-CCM+ improves the quality of the mesh by switching to an advanced mapping technique. When this switch happens, the following message appears in the output window: `Unable to use barycentric mapping. Switching to alternate method...`

Deleting a Volume Distribution

This section describes how to delete a volume distribution.

To delete a volume distribution from within the Directed Mesher menu :

1. Right-click the guide surface in the *Graphics* window and select **Part Collection** > **Delete Volume Distribution**.



Alternatively:

2. Right-click the **Mesh Distributions** > **Volume Distribution** node and select **Delete**.

Generating the Directed Volume Mesh

You can generate the volume mesh for a single part, or for several parts at the same time, in the Directed Mesh panel. This operation can involve single (1-1) or multiple source meshes (M-1) sweeping to a single target surface.

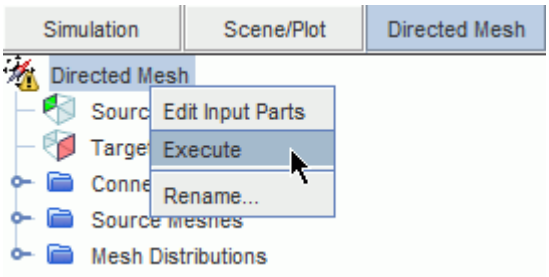
Assign parts to a region before generating a volume mesh.

For a single part:

1. Right-click a specific part under the **Connected Parts** node and select **Generate Volume Mesh**.

For all the parts:

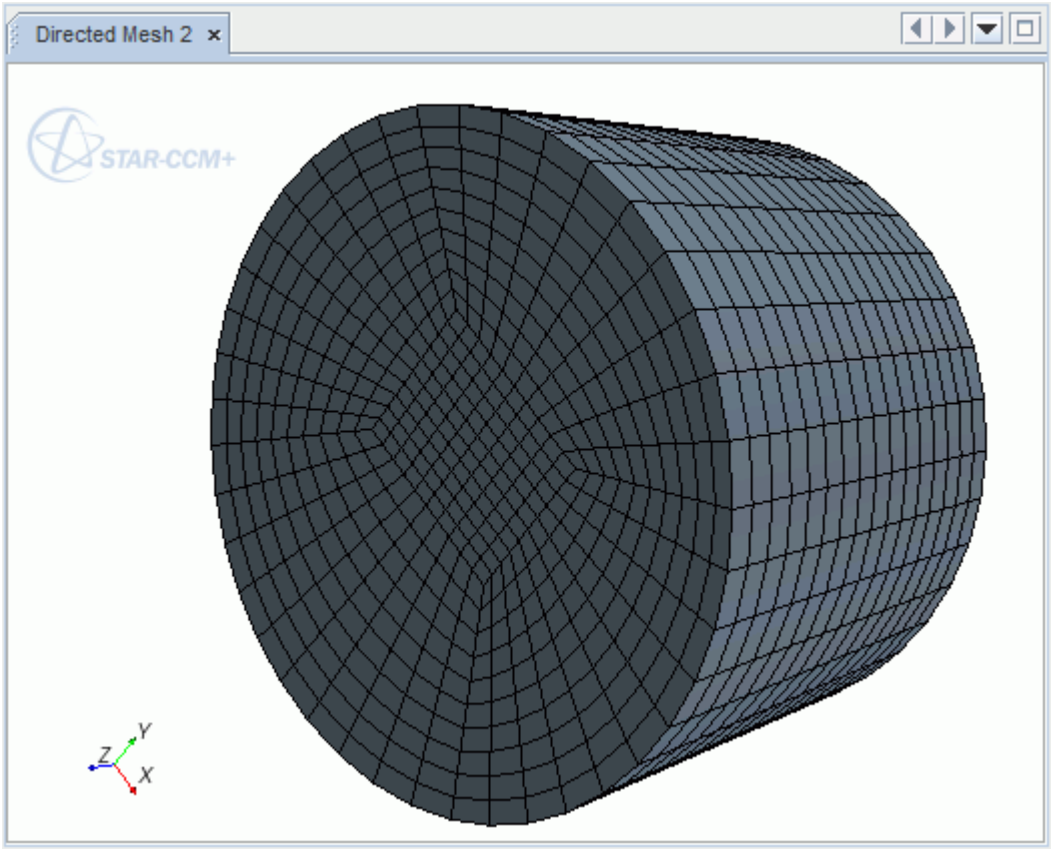
2. Choose one of the following options:
   - Right-click the **Directed Mesh** node and select **Execute** to generate volume meshes for all connected parts at the same time.



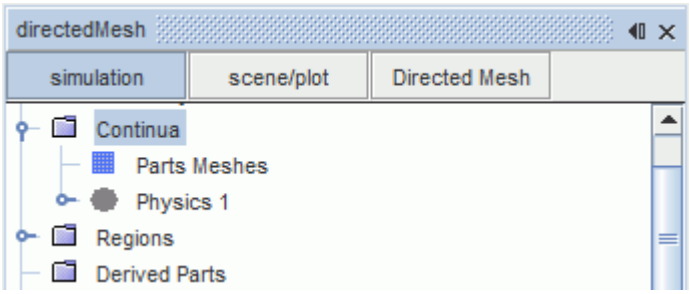
   - Click the (**Generate Volume Mesh**) button in the toolbar.

     This option generates all meshes that are defined for the regions, that is, directed meshes and those set using the standard STAR-CCM+ mesh continuum models. The directed meshes are generated after any region meshes.

The display is updated with the volume mesh.



The volume mesh is automatically transferred to the regions associated with the directed meshing parts. STAR-CCM+ also creates a **Parts Meshes** mesh continuum, and a basic physics continuum.



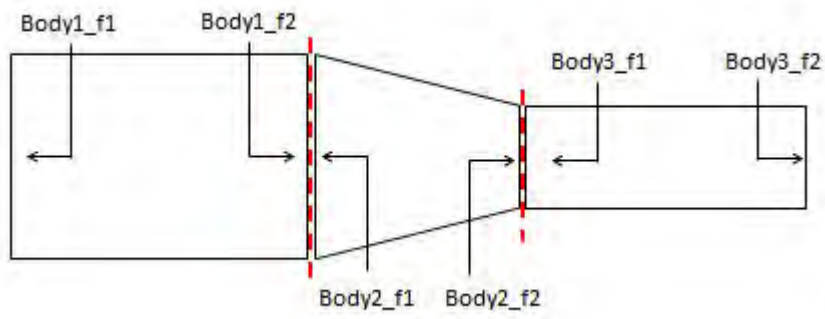
Note: If you clear the mesh or change the parts to regions assignment, use the **Generate Volume Mesh** button to re-execute the directed mesh operation. Doing so updates the regions and the mesh appropriately.

Note: If you clear the generated mesh from the toolbar, all source meshes are cleared and the source mesh node is marked as "dirty". When you subsequently enter the *Directed Meshing* panel for a patch mesh, the patch mesh is rebuilt automatically.

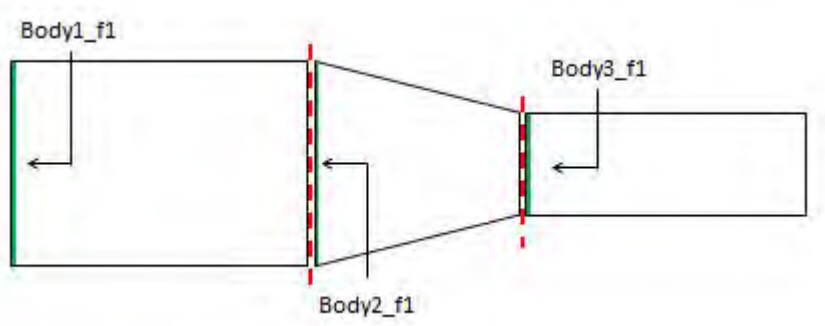
Meshing a Stack of Parts

This section describes how to mesh a stack of parts.

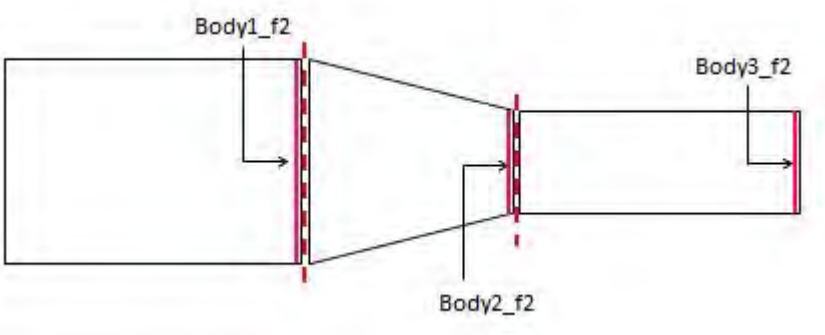
When meshing a stack of parts, prepare the parts with a consistent naming convention so that you can identify each side of the interface throughout the stack.



When setting the Source Surfaces for the Directed Mesh operation, choose the faces on the upstream side of the initial part in the stack.



When setting the Target Surfaces for the Directed Mesh operation, choose the faces on the downstream side of the final part in the stack.



Note: The directed meshing process transfers mesh from source surfaces on to target surfaces. Therefore, target surfaces cannot contain internal feature edges.

1. To identify source and target interfacing surfaces in the stack, use one of the following options:
   - Selecting Source And Target Surfaces Interactively

     Use this option to autopopulate intermediate source and target surfaces.
   - Selecting Source And Target Surfaces Using The Tree
2. After identifying the source and target surfaces, define a mesh by either:
   - Patch Meshing
   - A previously defined volume mesh

Contents:

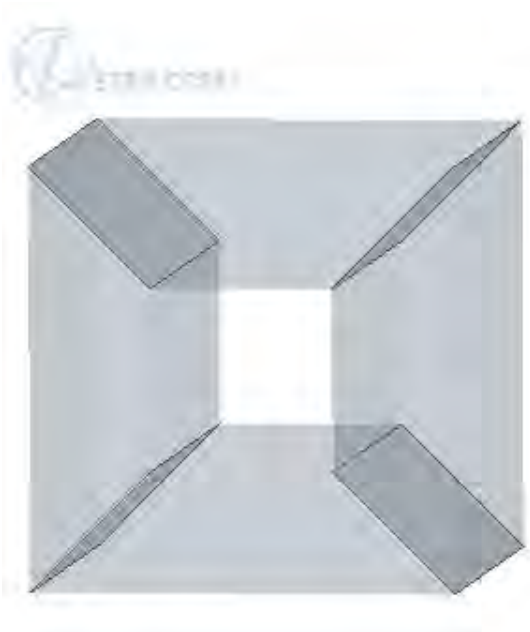
Meshing Multiple Parts in a Closed Loop

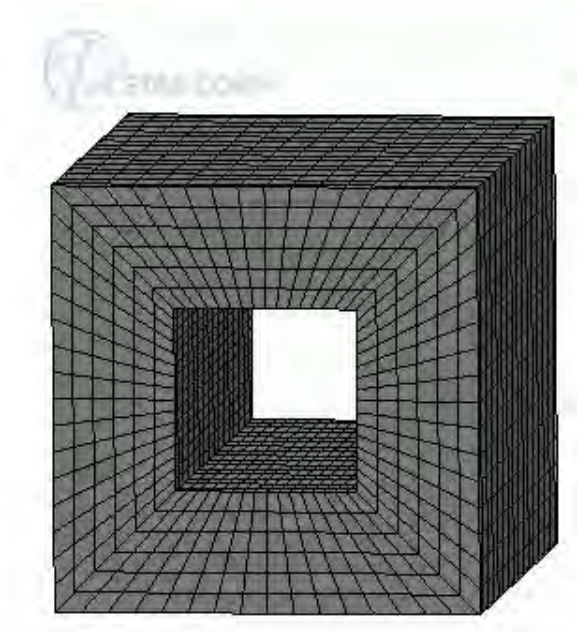
Meshing Multiple Parts in a Closed Loop

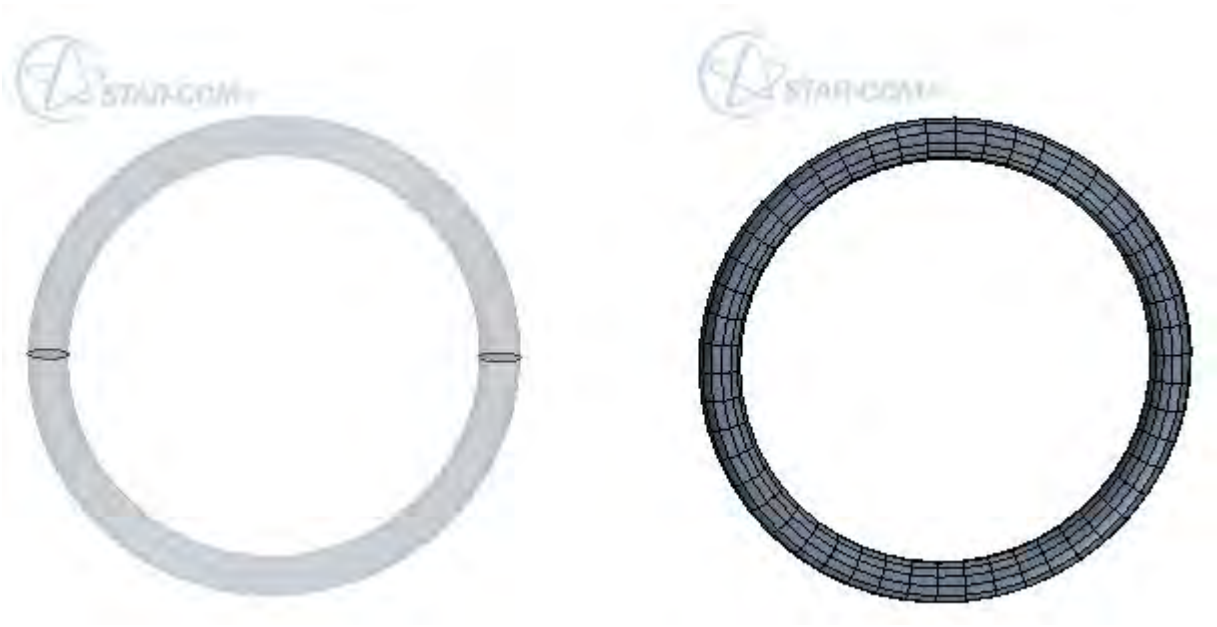
STAR-CCM+ can mesh two or more parts that form a closed loop, using a single Directed Mesh Operation. This operation is only possible using one source to one target (1-1).

You specify the source and target surfaces in the same way as for a stack of parts.

The examples below show the results of meshing multiple parts in a closed loop, using the directed mesher.



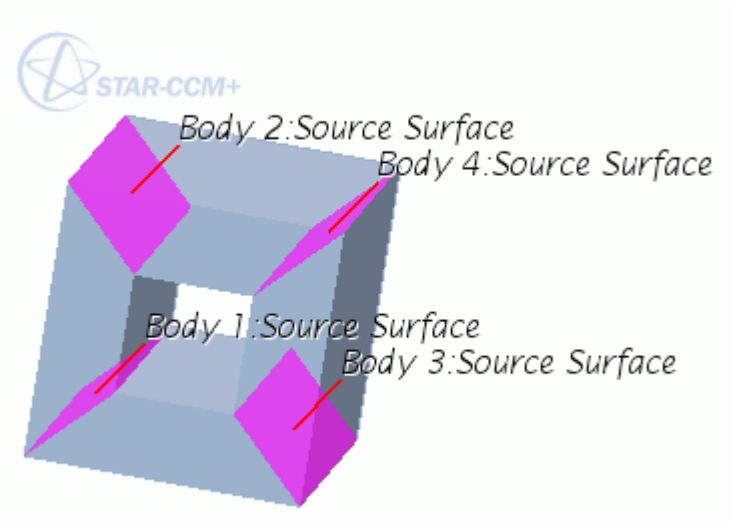


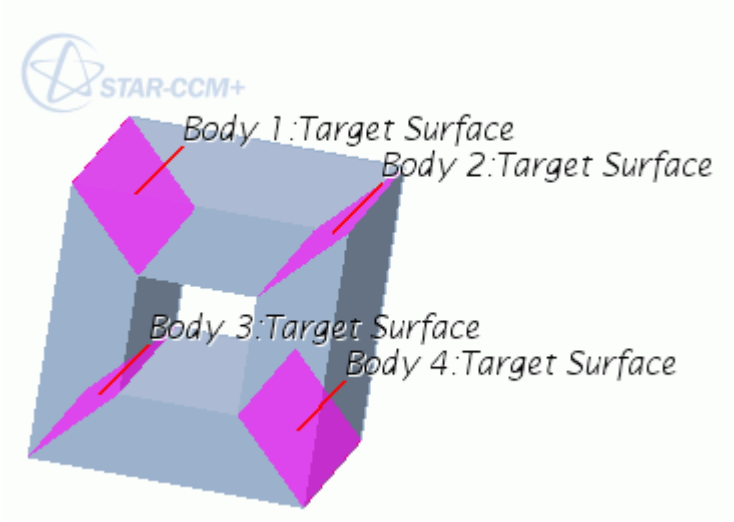


To produce a directed mesh in a multi-part closed loop:

1. Set the **Source Surfaces** and **Target Surfaces** for the directed mesh.

   At each boundary of coincident surfaces, ensure that there is one **Source Surface** and one **Target Surface**.



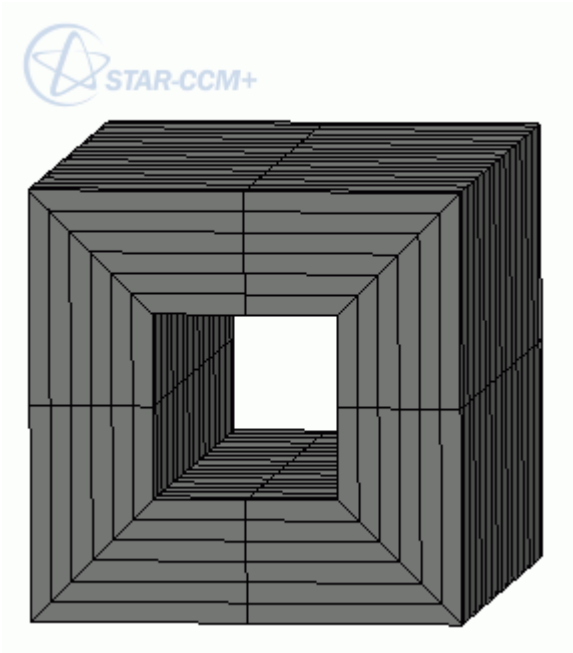


2. Define the source mesh by:
   - Using a **Patch Mesh**

     See Creating a Source Mesh Using Patches.
   - **Using Existing Mesh**

     See Creating a Source Mesh Using an Existing Mesh

   The directed mesh is displayed in the *Graphics* window:

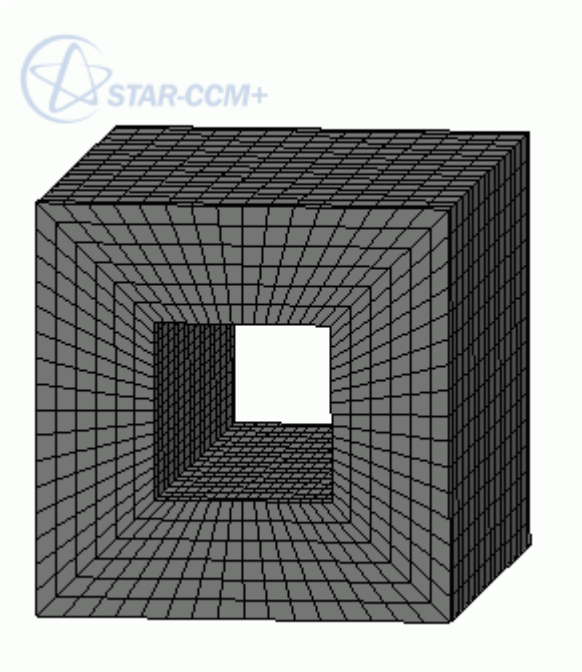


To refine the mesh quality across each part:

3. Select the **Mesh Distributions** > **Volume distribution** node.

4. In the *Properties* window, update the *Number of Repetitions* to match the number of parts that compose the closed loop.
5. To increases the refinement, increase the **Volume distribution** > **Number of Layers** in the loop.

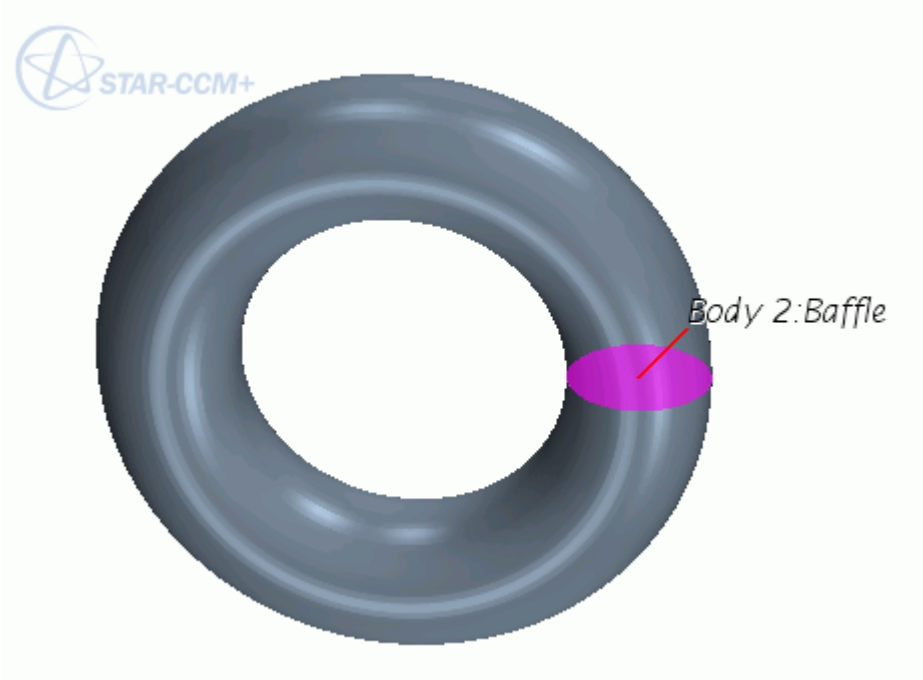
   The following model is set to `16` layers in `4` repetitions:

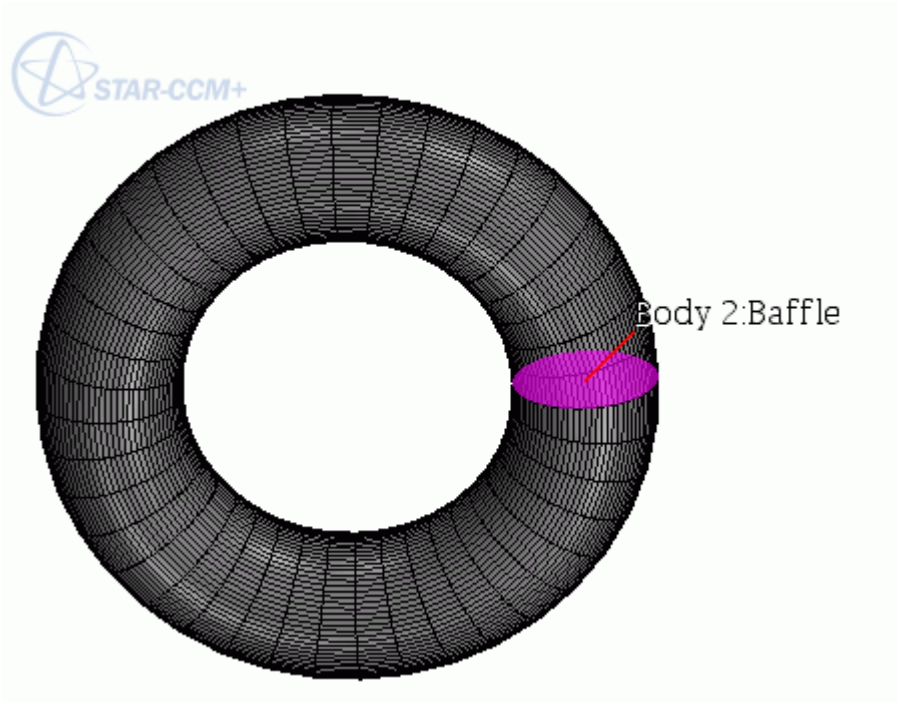


Meshing a Single Part in a Closed Loop with a Baffle

This section describes how to mesh a single part in a closed loop with a baffle. This operation is only possible using one source to one target (1-1).

STAR-CCM+ can perform directed meshing on a single part in a closed loop using one baffle surface as the designated source and target part (1-1 direct meshing only).





To mesh a single part in a closed loop with a baffle:

1. In 3D-CAD, imprint the solid body and sheet body that represent your closed loop part and baffle respectively. See Imprinting Bodies.

Note: Parts must be imprinted in 3D-CAD to make sure that the baffle part is recognized as two-sided when you set the source and target surfaces.

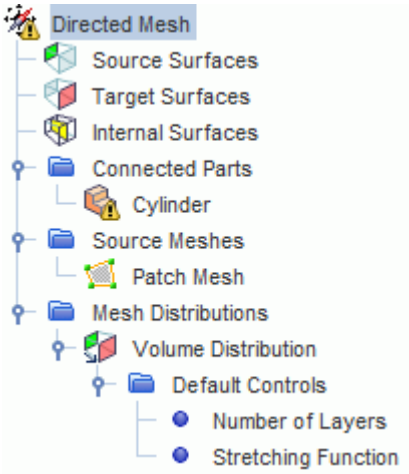
2. In STAR-CCM+, create the geometry parts.
3. Combine the solid body and sheet body parts.
4. Expand the **[Body]** > **Surfaces** node.
5. Rename the surface that represents the baffle.
6. Right-click the **[Baffle]** surface and select **Create Baffle**.
7. Launch a directed mesh operation.
8. Set Source and Target surfaces.
   - Set Source Surface to **[Baffle]**.
   - Set Target Surface to **[Baffle] 2**.
9. Define the source mesh by creating a source mesh using an existing mesh or creating a source mesh using patches.
10. Create a Volume distribution.
11. Generate the Directed Mesh Operation.

Requirements

- The body that is meshed can only contain one baffle.
- There can be no baffles within the path of the sweep between the source and target.
- The baffle surfaces must be designated as source and target surfaces. Partial baffles are not allowed.
- The baffle surface should not have any free edge.

Directed Meshing Properties

This topic describes the *Properties* of each of the nodes that are available for directed meshing.



Directed Mesh Properties

The following *Properties* are available when you select the **Directed Mesh** node. Some of these are only relevant to 1-1 Direct Meshing, as indicated.

Input Parts	Allows you to select one or more suitable parts from which to create a directed mesh. See Preparing CAD Parts For Directed Meshing.	
Generate Internal Surfaces (1-1 Direct Meshing only)	Provides the option of creating internal surfaces within the directed mesh.	
	Activated	Creates a **Directed Mesh** > **Internal Surfaces** node which allows you to define an internal mesh faces between the channels of a directed mesh.
	Deactivated	No internal faces are created within the directed mesh.

Source Surfaces Properties

The following property is available to set when you select the **Directed Mesh** > **Source Surfaces** node.

Source Surfaces	Allows you to select one or more surfaces to represent the source surface.

Target Surfaces Properties

The following property is available to set when you select the **Directed Mesh** > **Target Surfaces** node.

Target Surfaces	Allows you to select one or more surfaces to represent the target surface.

Internal Surfaces Properties

The following property is available to set when you select the **Directed Mesh** > **Internal Surfaces** node.

Internal Surfaces (1-1 Direct Meshing only)	Allows you to select a previously created empty part surface on which to create internal mesh faces. Mesh faces run along the internal and external sides of a channel running from a source mesh face, through the volume mesh, to the corresponding target mesh face. If the part surface is assigned to a wall boundary, cross flow is not allowed between the channels. This functionality is useful when performing CFD analyses of heat exchangers which model hot and cold fluid streams through channels in a porous region between periodic interfaces.

Connected Parts – Part Properties

The following read-only property is available when you select the **Directed Mesh** > **Connected Parts** > **[part]** node. Each **[part]** node is named after the connected geometry parts that are selected for directed meshing.

Parts	Displays the names of the connected geometry parts that are selected for directed meshing.
Source Mesh Node	Displays the source mesh that is associated with the connected part.
Target Surface Mesh Smoothing	Controls whether the target surface mesh is smoothed. Activate this option to use the smoothing algorithms within the mesher. In some cases, such as simple cube and cylinder shapes, it is not necessary to smooth the target surface mesh. In other cases, smoothing a target surface mesh can deteriorate the surface mesh quality (this is especially true of M-1 direct meshing of more complex parts. In such cases, deactivate the smoothing of the target surface mesh.

Source Meshes – Existing Mesh Properties

The following *Properties* are available to set when you select the **Directed Mesh** > **Source Meshes** > **[existing mesh]** node. Each source mesh node is named after the contacting part collection which the source mesh is defined on.

Source Mesh From	Specifies the source mesh.	
	Contacting Volume Mesh	Extracts the mesh from a volume mesh that contacts the specified source surfaces.
	Triangle Mesh	Extracts the latest triangle mesh of the part or parts being meshed. If the original triangulation is obtained, the source mesh that is extracted can be of poor quality if it contains long, narrow triangles. If the part is remeshed, the remeshed triangulation is obtained.
	Quadrilateral Mesh	Obtained from the extracted triangulation. The mesh is dependent upon the quality of the extracted triangulation. The quadrilateral mesh that is obtained is quad-dominant with a mix of quads and triangles. The quadrilateral mesh does not faithfully replicate the extracted triangle mesh density.

Part Surfaces	Displays the part surfaces that are specified as the source surface meshes on the connected part.

Mesh Distributions – Volume Distribution Properties

The following *Properties* are available to set when you select the **Directed Mesh** > **Mesh Distributions** > **Volume Distribution** node.

Part Collections	Sets the part collection to which this distribution is applied.	
Mesh Controls	Allows you to create custom controls for each connected part in the directed mesh operation.	
	Use Default	Maintains the **Volume Distribution** > **Default Controls** node which allows you to specify the number of layers and stretching function for all connected parts together.
	Specify Custom	Removes the **Volume Distribution** > **Default Controls** > **Stretching Function** node with a **Volume Distribution** > **Custom Controls** node which allows you to create part controls for customizing the number of layers and stretching function for each connected part.
Number of Repetitions	Sets the number of times the volume mesh distribution is repeated between the source and the target surface. This repetition can be useful when meshing stacks of parts.	

Default Controls

Default Controls – Number of Layers

To set the number of layers using default controls within the volume distribution, select the **Volume Distribution** > **Number of Layers** node.

Number of Layers	Sets the number of cell layers to create between the source and target surfaces in the volume mesh. For a stack of connected parts, this is the sum of cell layers through all connected parts.

Default Controls – Stretching Function

To set the stretching function using default controls within the volume distribution, select the **Volume Distribution** > **Stretching Function** node.

Stretching function	Sets the stretching function for the mesh layers.	
	Constant	Constant cell distribution
	One Sided Geometric	Geometric progression to compute cell thickness
	One Sided Hyperbolic	Hyperbolic based stretching to compute cell distribution
	Two Sided Hyperbolic	Hyperbolic stretching from both ends to compute cell distribution

Note: When working with stacks of parts, and parts that are connected at the guide surface, STAR-CCM+ attempts to fit the requested distribution to the stack of parts. The requested distribution in the resulting mesh cannot always be guaranteed since divisions close to the end of the part are moved to the end of that part.

If the stretching function is set to One Sided Geometric or One Sided Hyperbolic, a **One Sided Options** node is added under **Volume Distribution** > **Stretching Function**. Select the stretching mode that you want to use in the *Properties* window of this node and its value in its sub-node.

- **One Sided Options Properties:**

Stretching mode	Specifies the type of parameter that calculates the stretch value.	
	Stretching	Ratio of spacing at layer n and layer n+1
	Initial Wall Thickness	Spacing at the first layer of cells
	Thickness Ratio	Ratio of spacing between first layer and last layer of cells
Reverse Direction	When activated, reverses the direction of the stretching function.	

If the stretching function is set to Two Sided Hyperbolic, the **Two Sided** node is added under **Volume Distribution** > **Stretching Function**. Set the initial wall spacing at the source and target surfaces in the *Properties* window of this node.

- **Two Sided Properties**

Spacing at Source	Sets the initial wall spacing at the source surface.
Spacing at Target	Sets the initial wall spacing at the target surface.
Reverse Direction	When activated, reverses the direction of the stretching function.

Custom Controls

Custom Controls – Context Menu

New Part Control	Creates a **Part Control** sub-node which allows you to customize the *Properties* of a part or parts that you select.

Custom Controls – Part Control Properties

Parts	Allows you to select parts for which the custom control applies.

Custom Controls – Part Control Expert Properties

Enable Control	Allows you to activate and deactivate the functionality of the Part Control.	
Controls Display Mode	**All**	Displays all available controls in the tree.
	Customized	Allows you to select which of the directed meshing part controls to display in the simulation tree.

Custom Controls – Number of Layers Properties

Controls:

Number of Layers	**Use Auto Computed Value**	Calculates the Number of Layers in each part automatically using the value that is set for the sum of all parts in the default controls.
	Specify Custom Value	Creates a **Values** > **Number of Layers** node.

Values:

Number of Layers	**Number of Layers**	Allows you to Specify the number of layers in the volume mesh for the parts in the Part Control. This custom Number of Layers overrides the default Number of Layers and prints a warning message in the *Output* window if: • The custom Number of Layers exceeds the default Number of Layers that is set. • All parts in the stack have a custom Number of Layers defined. When the custom Number of Layers for a part is subtracted from the default Number of Layers, the remaining number of layers are distributed to parts that are not in this Part Control.

If a part is assigned to more than one Part Control, the highest value that is set for the Number of Layers is used for that part.

Custom Controls – Stretching Function Properties

Controls:

Stretching Method	**Use Auto Computed Value**	Distributes the layers automatically throughout all parts. Uses the layer thickness from preceding/ succeeding portions to smoothly transition the distribution without abrupt jumps in layer thickness.
	Specify Custom Value	Creates a **Values** > **Stretching Function** node.

Values:

Stretching function	Sets the stretching function for the mesh layers. When stretching functions are defined, the distribution of cell layers transitions smoothly between preceding or succeeding parts without any abrupt jumps in the layer thickness.	
	Constant	Constant cell distribution
	One Sided Geometric	Geometric progression to compute cell thickness
	One Sided Hyperbolic	Hyperbolic based stretching to compute cell distribution
	Two Sided Hyperbolic	Hyperbolic stretching from both ends to compute cell distribution

If a part is assigned to more than one Part Control with the same type of stretching function, the Stretching Function Value with the smallest layer thickness is set for that part.

See Also:

Creating a Volume Distribution

Two Dimensional and Axisymmetric Meshes

The use of a one-cell-thick three-dimensional mesh is much less efficient than using a real two-dimensional mesh for two-dimensional and axisymmetric simulations. When circumstances allow, use real two-dimensional meshes.

There are several ways to obtain a 2D mesh in STAR-CCM+:

- Create a 2D mesh directly in STAR-CCM+.

  This option provides the most efficient and computationally least expensive method for creating a 2D mesh in STAR-CCM+. See Creating a Two-Dimensional Mesh.
- Create a 2D mesh in an external program and import it directly into STAR-CCM+.

  The Gridgen mesh generator, for example, supports 2D meshes.
- Create a 3D mesh and use the STAR-CCM+ option, **Mesh** > **Convert to 2D** to extract the cell faces that lie on the Z=0 plane into a 2D mesh.

  See Examples Of Two Dimensional Mesh Extraction.

  During the conversion process, the choice is given to either keep the original 3D mesh or delete it automatically and maintain only the derived 2D mesh. The choice of which region or regions to convert is also provided.

To practice creating a 2D mesh, try completing the Steady Flow: Backward Facing Step tutorial.

Contents:

Creating a Two-Dimensional Mesh
Extracting The Two Dimensional Mesh
Examples of Two-Dimensional Mesh Extraction

Creating a Two-Dimensional Mesh

Follow these instructions to produce a real two-dimensional mesh in STAR-CCM+ from selected two-dimensional part surfaces.

Creating a 2D mesh is more efficient and less computationally expensive than extracting a 2D mesh from a 3D mesh. 2D meshes are useful for some electric machine and aerospace industry analyses.

Before creating a 2D mesh:

1. The whole part surface on which a 2D mesh is required must be coincident with the Z=0 plane, within a tolerance of 1.0e-6. If necessary, transform the parent part so that the target part surface lies on the Z=0 plane. You can also use splitting tools to break out the target surface into its own part surface. See Manipulating Part Surfaces.

   Note: For 2D axisymmetric analyses, the axis must lie on the X-axis and the mesh must also be above the local Y=0 plane. If necessary, transform the parts. See Transforming Parts to a Coordinate System.

To create a 2D mesh:

2. Badge the 2D parts that contain the Z=0 surfaces.
   See Identifying the Target Part Surfaces for 2D Meshing.
3. Assign parts to 2D regions. See Assigning Parts to Two-Dimensional Regions.

   Separate regions are created for 2D and 3D parts.

   Note: Parts that are badged as 2D cannot belong to the same region as parts that are not badged.
4. Create an **Automated Mesh (2D)** operation.
   See Creating the Two-Dimensional Automated Mesh Operation.
5. Generate the volume mesh.

   The 2D mesh is generated only on the Z=0 surfaces that you identify using the **Badge For 2D Meshing** operation.

Contents:

Identifying the Target Part Surfaces for 2D Meshing
Assigning Parts to Two-Dimensional Regions
Creating the Two-Dimensional Automated Mesh Operation

Identifying the Target Part Surfaces for 2D Meshing

You identify part surfaces on which a 2D mesh is required using the **Badge for 2D Meshing** operation. This operation identifies part surfaces that lie on the Z=0 plane. If any part surfaces lie partially on the Z=0 plane, the **Badge for 2D Meshing** operation fails.

To identify parts for 2D meshing:

1. Create a **Badge for 2D Meshing** operation.
   You can create any number of **Badge for 2D Meshing** operations, however, a part can only belong to one **Badge for 2D Meshing** operation.
   a) Right-click **Geometry** > **Operations** and select **New** > **Badge for 2D Meshing** .
   b) In the *Create Badge for 2D Meshing Operation* dialog, select the parts that are required to produce a 2D mesh. It is not necessary to select part surfaces, as STAR-CCM+ identifies the part surfaces that lie on the Z=0 plane automatically.
2. Execute the **Badge for 2D Meshing** operation.
   Parts and part surfaces are badged and available to select for **Automated Mesh (2D)** operations as follows:

   = Available to select as 2D region

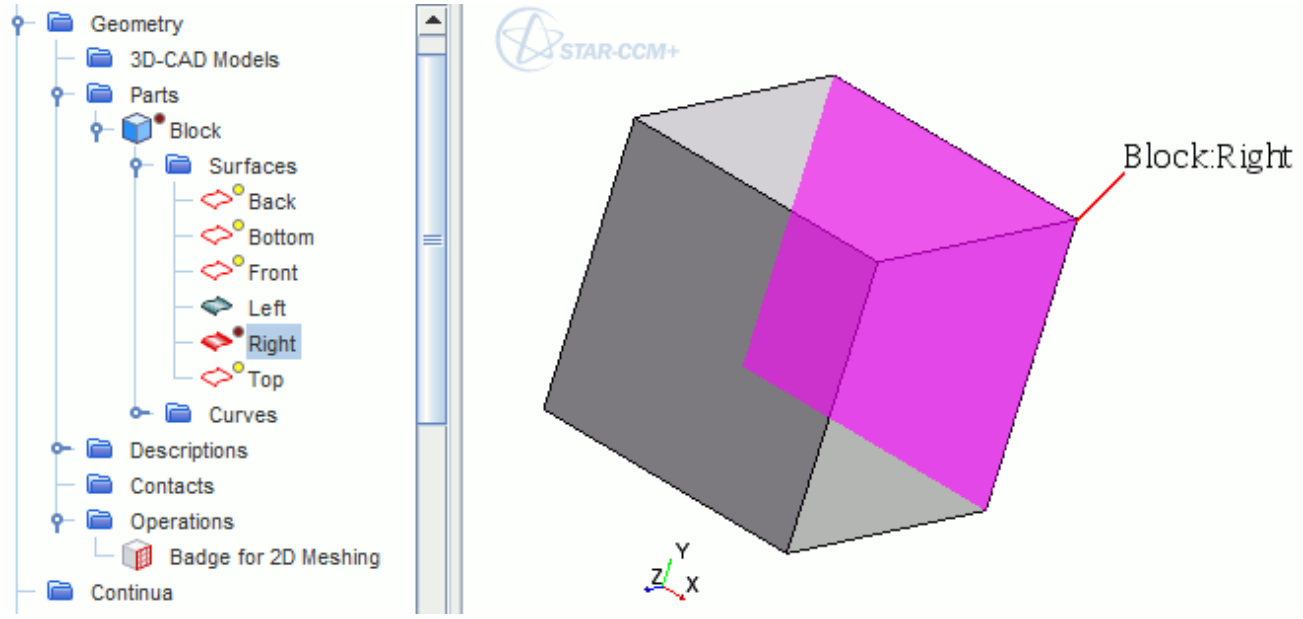
   = Available to select as 2D boundary

   Note: Deleting a **Badge for 2D Meshing** operation removes the badge from the input parts and part surfaces.

Understanding the Results of a Badge for 2D Meshing Operation

The following example shows how part surfaces on a simple three-dimensional block part are badged during a **Badge for 2D Meshing** operation.

On a simple block that is split into six part surfaces, only one part surface can wholly lie on the Z=0 plane. Only the part surface that lies on the Z=0 plane is badged as a 2D region.



The four part surfaces that are connected to the part surface on the Z=0 plane, are badged as boundaries. The edges of these part surfaces, that lie on the Z=0 plane, form the boundaries of the 2D region containing the final 2D mesh.

Any part surfaces that are disconnected from the Z=0 plane, are not badged.

Parts which contain badged part surfaces, also display a badge.

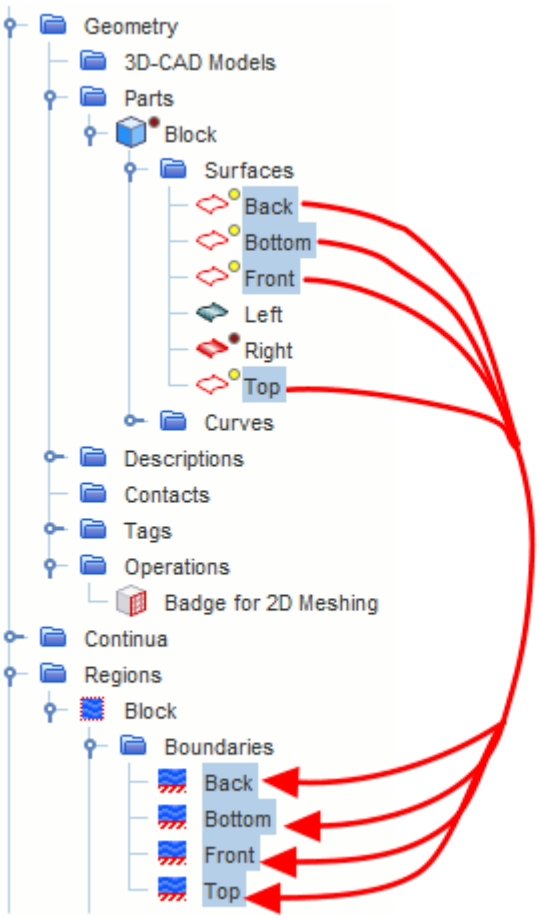
Assigning Parts to Two-Dimensional Regions

This section describes how to assign parts which are badged as two-dimensional to regions.

1. Select the **Geometry** > **Parts** that you are assigning to 2D regions.
2. Right-click one of the selected parts and select **Assign Parts to Regions**.
   Parts that are badged as 2D are assigned to different regions from parts that are not badged.
   - When parts that are not badged as 2D are assigned to regions, all part surfaces on a part are assigned to boundaries as specified in the *Assign Parts To Regions* dialog.

See Assigning Parts To Regions

- When parts that are badged as 2D are assigned to regions, only part surfaces that are badged as 2D boundaries, are assigned to boundaries.



Boundaries are not created for part surfaces that are disconnected from the 2D region. These disconnected boundaries are disregarded as they do not have any boundary conditions and they are not part of the final 2D mesh.

Creating the Two-Dimensional Automated Mesh Operation

The **Automated Mesh (2D)** operation generates the 2D mesh using the mesher that you choose. Only parts that are badged for 2D are available for selection as input parts for an **Automated Mesh (2D)** operation.

Before creating the **Automated Mesh (2D)** operation:

1. Identify and badge the 2D parts. See Identifying the Target Part Surfaces for 2D Meshing

To create the **Automated Mesh (2D)** operation:

2. Right-click **Geometry** > **Operations** and select **New** > **Automated Mesh (2D)**.
3. In the Create **Automated Mesh (2D) Operation** dialog:

a) Select the parts from which to create the 2D mesh.

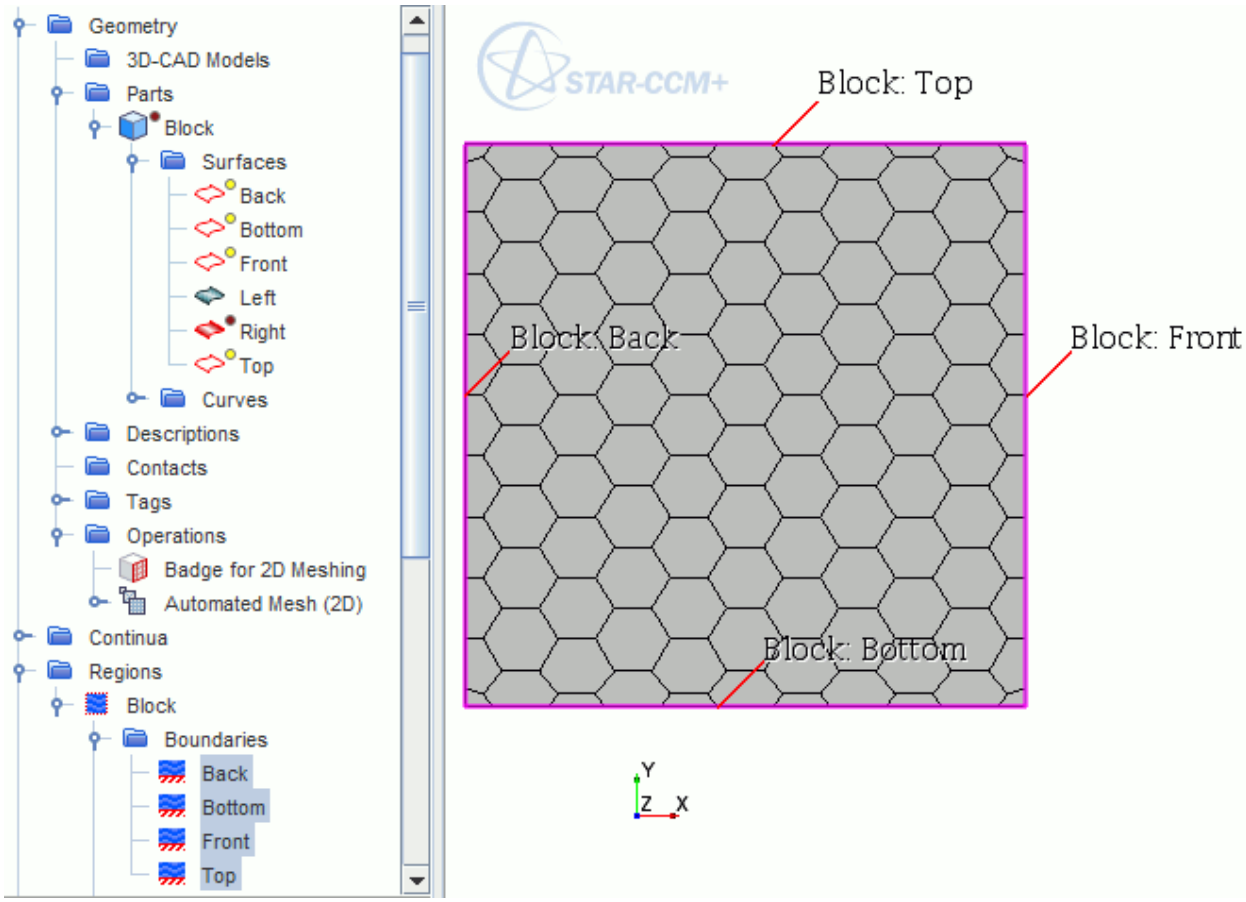
b) Select a volume Mesher:

- Polygonal Mesher
- Quadrilateral Mesher
- Triangular Mesher

You can also select the optional Prism Layer Mesher.

4. Execute the **Automated Mesh (2D)** operation.

Parts that are badged as 2D are meshed. Any part surfaces that are badged as 2D boundaries, create boundaries which appear as edges within the 2D mesh.



Note: You can copy default controls between 3D and 2D meshes.

Extracting The Two Dimensional Mesh

This section describes how to extract the two dimensional mesh from the surface of an existing three-dimensional mesh. This existing mesh can either be imported into STAR-CCM+ or generated using the three-dimensional volume meshers.

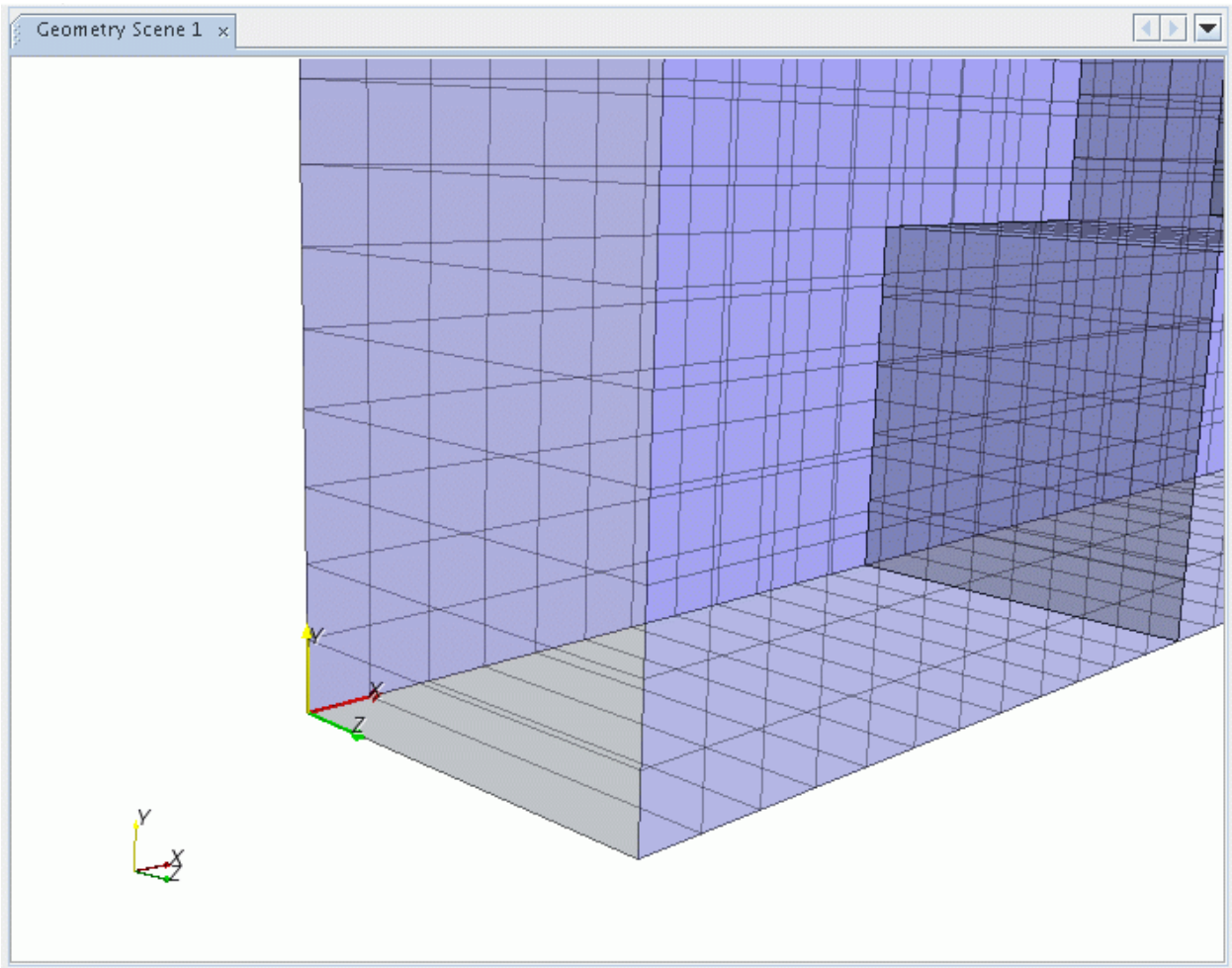
1. For the existing three-dimensional mesh, check that the boundary that you want to extract lies in the Z = 0 plane.

   You can check the alignment in a number of ways:

   - To ensure that the Z extent includes a zero, review the extents of the mesh by:
     - Looking at the results of the import process in the *Output* window
     - Running a mesh diagnostic report and looking at the report in the *Output* window

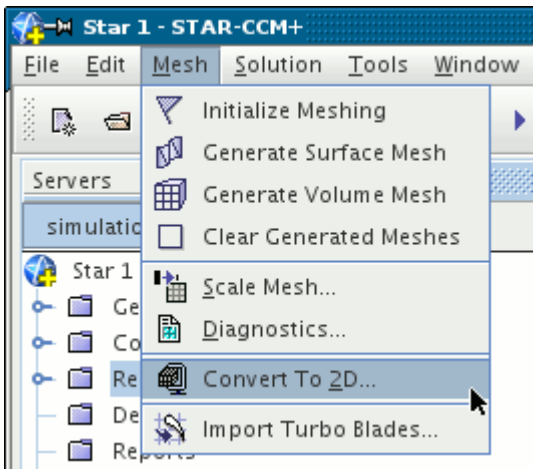
       See Running Mesh Diagnostics.
   - Create a new geometry scene with the default laboratory coordinate system included as part of the view. You can then verify visually the location of the Z = 0 plane. See Displaying the Geometry.

   The example below shows the default laboratory cartesian coordinate system included in the model display. As can be seen, one of the boundaries is aligned with the Z = 0 plane:

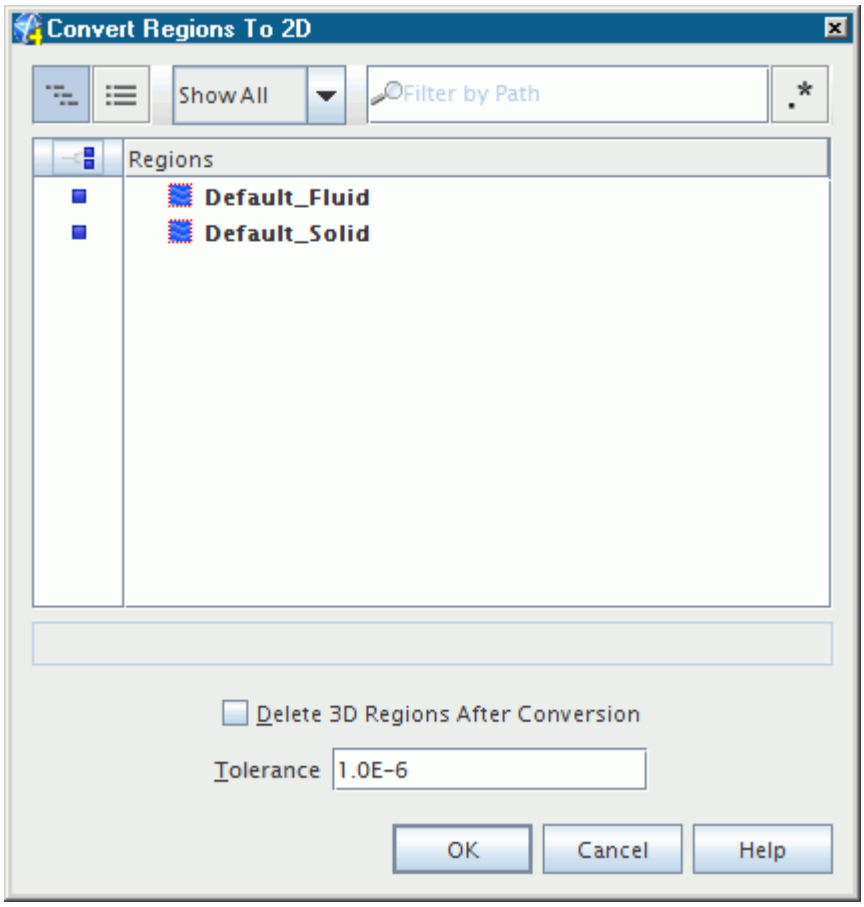


If the model is not oriented correctly, rotate and/or translate the model until the face of interest lies on the Z = 0 plane. If a boundary lies near the Z = 0 plane but not consistently on it, you can project the boundary vertices to that plane.

2. Once you are satisfied with the positioning of the boundary that you wish to extract, convert the mesh by selecting **Mesh** > **Convert To 2D...**.



The *Convert Regions to 2D* dialog opens:

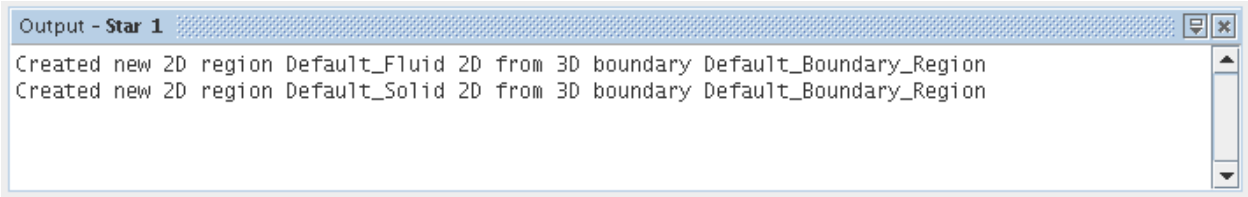


3. Select the regions to convert. Turning the **Delete 3D regions after conversion** option on would automatically delete the original regions that are selected for conversion. The default option is not to delete the regions.

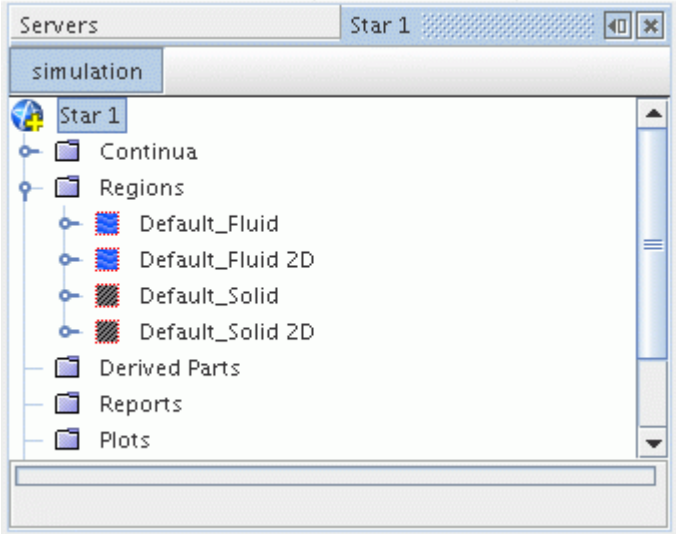
The **Tolerance** value determines the allowable deviation from the Z=0 plane of the mesh boundary that you wish to convert. For example, if the selected 3D regions have boundaries that are not exactly on the Z=0 plane, then the **Tolerance** value can be increased so that a complete boundary plane is found.

4. Pressing **OK** starts the conversion process.

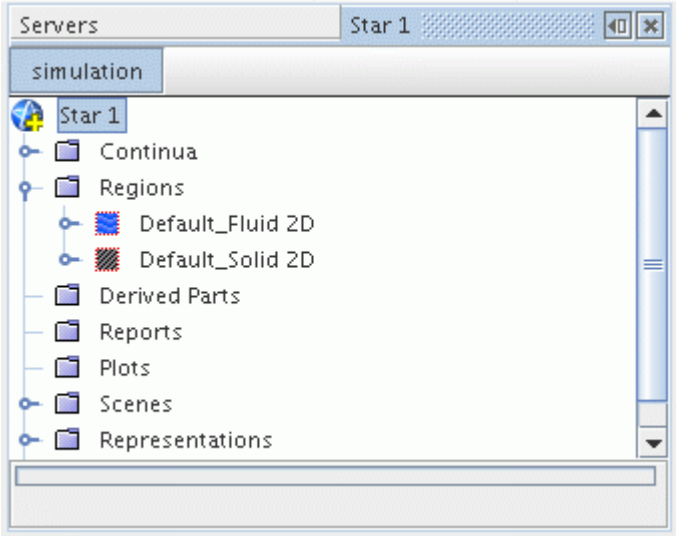
   Once the conversion is complete, the results are reported in the *Output* window where the names of the new regions and boundaries are listed along with the number of cells, faces, and vertices.



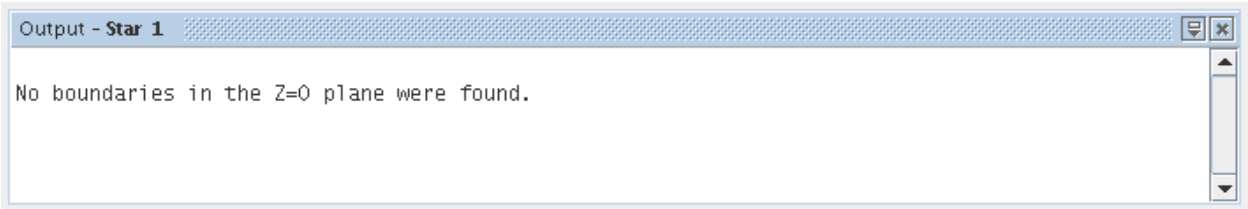
The simulation tree is also updated with the new region specification. Each converted region name is appended to include the 2D specifier, as shown below:



If the option to delete the original 3D regions was activated, then only the new 2D regions remain:



If STAR-CCM+ cannot find any faces in the z = 0 plane, then a message in the *Output* window reports that no suitable boundaries could be found:



Notes:

- If no boundary faces are found in the Z = 0 plane, no action is taken.
- Every distinct boundary with faces on the Z = 0 plane becomes a new, separate region.
- After extracting the two-dimensional mesh, you will need to reconstruct all Part Displayers.
- The extracted two-dimensional mesh is treated as being 1 meter in depth by the solver (1 radian for axisymmetric cases), even if you selected different units for length (such as feet or inches). Compensate manually for different depths.
- If only a partial two-dimensional mesh is created, you can delete the regions and try again with a larger tolerance (assuming the original 3D regions were not deleted).
- If the original 3D region has an associated physics continuum, then a new physics continuum is created for the 2D converted region. Further, if the 3D space model is active for the original physics continuum then the space model is automatically reset to be 2D in the new physics continuum for the two-dimensional region. All other models that are currently selected for the continuum are also carried over to the new continuum.

See Also:

Projecting Boundaries to a Plane

Translating Regions

Rotating Regions

Examples of Two-Dimensional Mesh Extraction

The two-dimensional extraction feature is useful when the mesh generator does not support true two-dimensional mesh creation and you have:

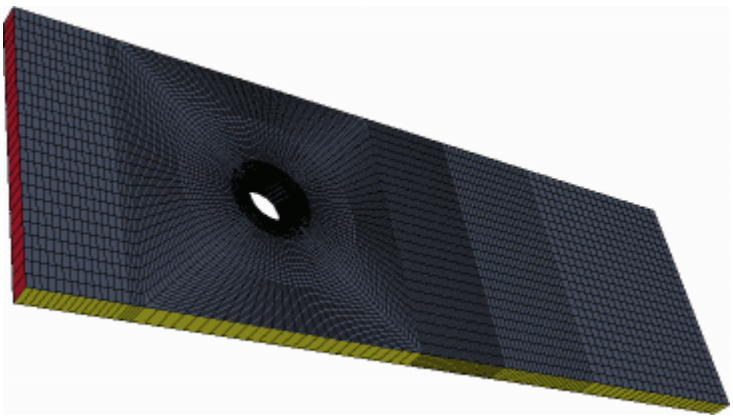
- Two-dimensional meshes that are extruded into the third dimension to have one cell thickness. Use 2D mesh extraction to run such cases as true two-dimensional simulations in STAR-CCM+
- Axisymmetric meshes that are created as annular segments. Use 2D mesh extraction to run such cases as 2D axisymmetric simulations in STAR-CCM+
- Models with symmetry planes that are representative of the whole model. Use 2D mesh extraction to run simplified 2D simulations in STAR-CCM+to investigate parameters such as model selection and boundary conditions.

Contents:

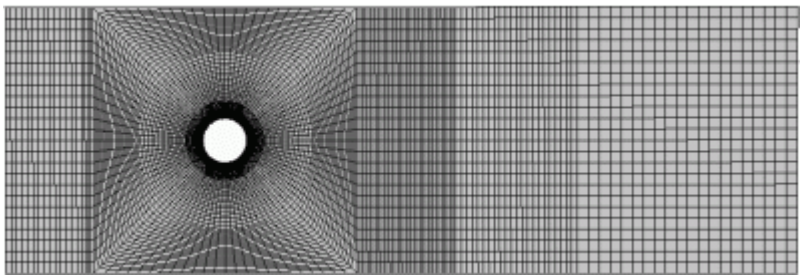
One Cell Thick 2D Extruded Meshes
Axisymmetric Annular Segment
Symmetry Plane

One Cell Thick 2D Extruded Meshes

The three-dimensional mesh that is shown below has been created to simulate flow over a cylinder. It is composed of one layer of hexahedral cells using symmetry boundaries in the spanwise direction.

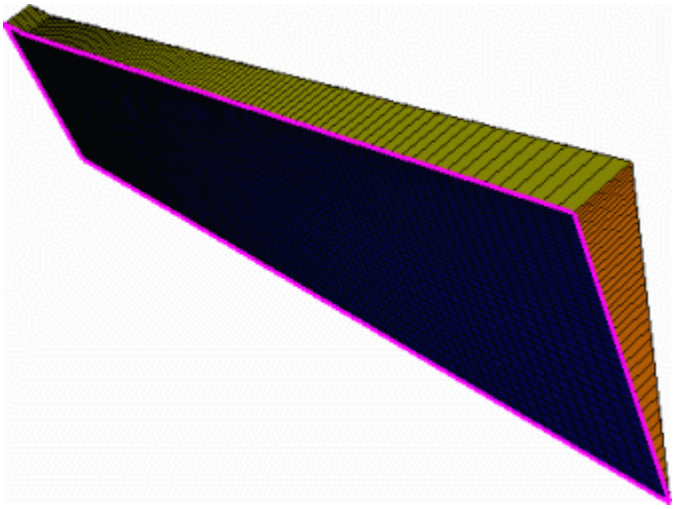


The two-dimensional mesh that is shown below is the result of the `Mesh > Convert To 2D` command.

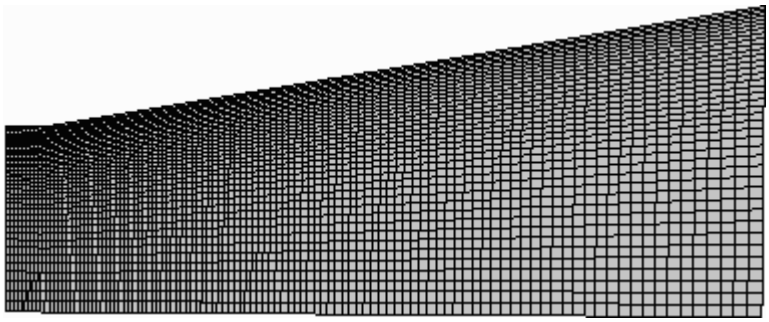


Axisymmetric Annular Segment

The image below shows an example of a one-cell-thick hexahedral mesh with wedge-shaped cells next to the axis for simulating flow through an axisymmetric diffuser. The original model is three-dimensional with cyclic boundaries.

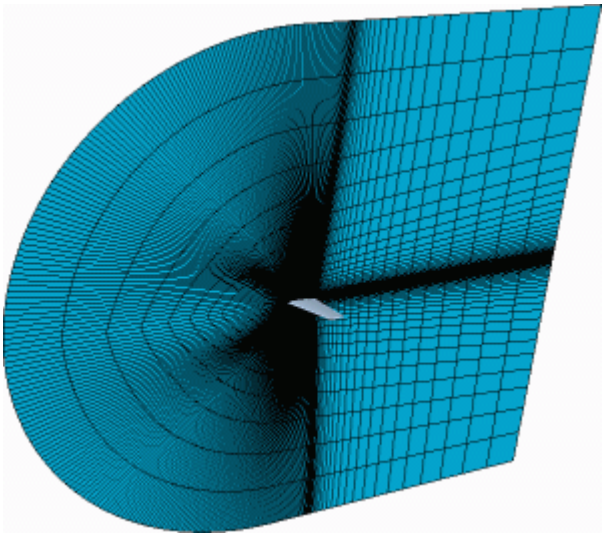


STAR-CCM+ is not used to create the mesh above. However, you can extract the 2D axisymmetric version of the mesh from the surface mesh on one of the cyclic boundaries. The resulting 2D axisymmetric mesh – which you can use in STAR-CCM+ – is composed of quadrilateral cells.

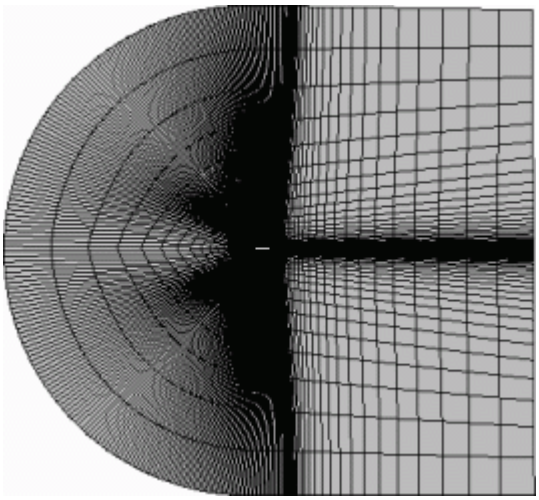


Symmetry Plane

The three-dimensional hexahedral mesh below simulates the flow over a wing that is attached to a symmetry plane.



The two-dimensional version that is shown below, composed of quadrilateral cells, has been extracted from the surface mesh on the symmetry plane to run a simplified simulation.



Mesh Visualization

You can visualize any mesh representation in STAR-CCM+. After you have imported or generated a mesh, you visualize it to check the cell quality and mesh validity.

You visualize the surface or volume mesh using displayers in scenes. These displayers have properties that allow you to change the mesh display characteristics. You can display either the wrapped surface, remeshed surface, or volume mesh in the scene by using mesh representations.

This section focuses on the display of surface and volume meshes only. However, you can also visualize solution data and mesh quality parameters on the mesh surfaces.

Contents:

Displaying the Mesh
Changing the Surface Opacity
Displaying the Volume Mesh on a Plane Section

Displaying the Mesh

Displaying a mesh allows visual inspection of a surface or volume mesh for quality and validity.

You can display a mesh by using the toolbar buttons or by creating a mesh scene.

Using the Toolbar Buttons

The easiest way to activate and deactivate the mesh display is with the buttons in the **Visualization** toolbar:

- To display a mesh, click (**Show All Meshes**). Clicking this button once adds the mesh lines, and overrides the settings of all the displayers in the scene. Clicking it a second time reverts to the displayer settings.
- To hide the mesh, click (**Hide All Meshes**). Clicking this button once removes the mesh lines, and overrides the settings of all the displayers in the scene. Clicking it a second time reverts to the displayer settings.

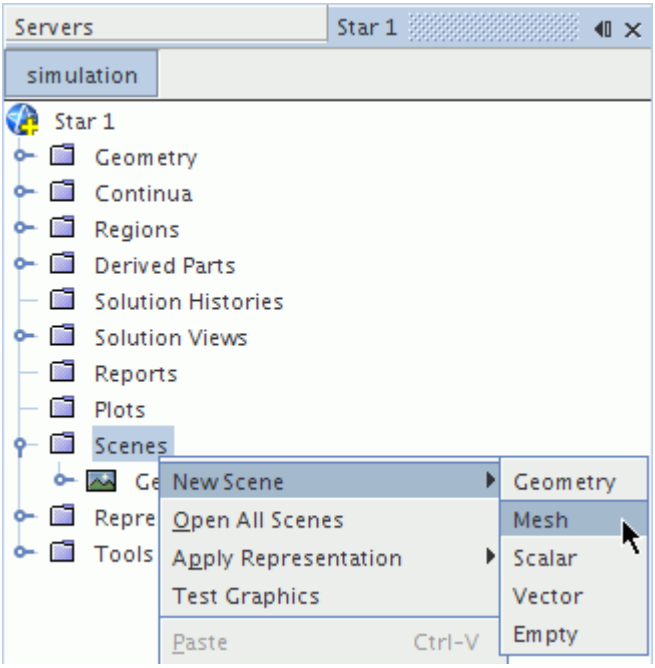
Using a Mesh Scene

To display a mesh, you can also create a mesh scene. A mesh scene contains a mesh displayer that shows the geometry surface and mesh lines for the selected regions. By default, the mesh displayer colors the geometry surface by boundary type.

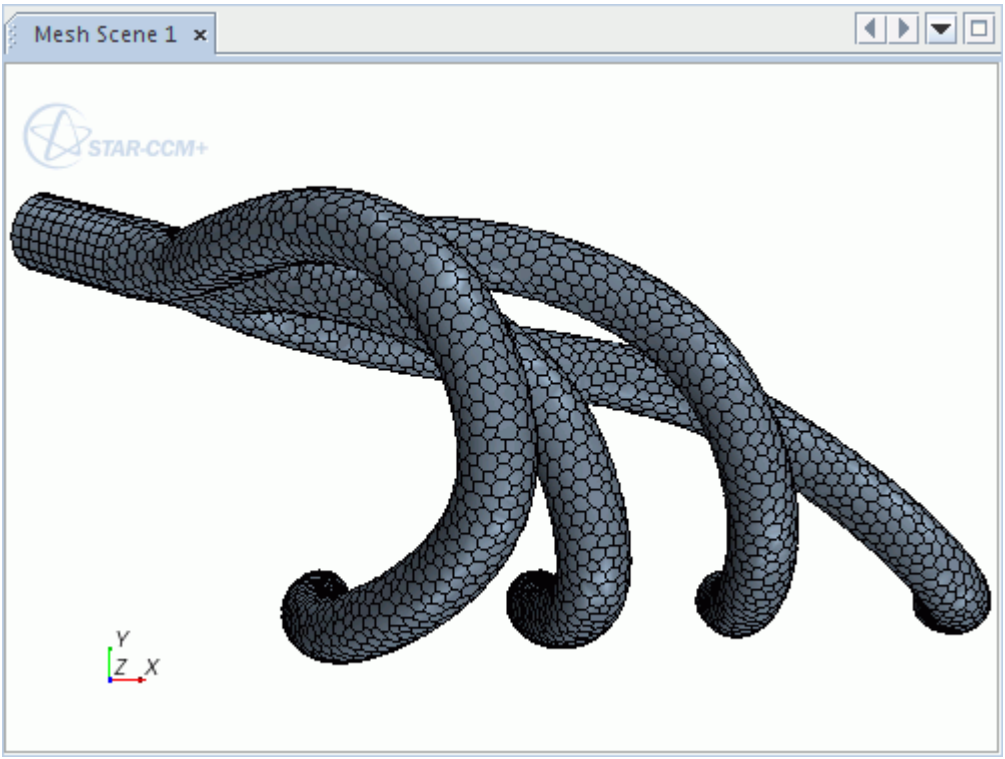
Creating a mesh scene displays the most recent mesh representation. For example, if you run the volume mesher and then create a mesh scene, the scene displays the volume mesh. To change the displayed mesh, select a new representation.

To create a mesh scene:

1. Right-click the **Scenes** node and select **New Scene** > **Mesh**.



The mesh scene appears in the *Graphics* window:



2. You can use the various view manipulations to navigate around the mesh by panning, rotating, and zooming.

By default, the mesh scene shows all geometry. You can identify and hide regions and boundaries to simplify the display.

You can change the displayer properties to alter the visualization characteristics. Some common examples are included in this section:

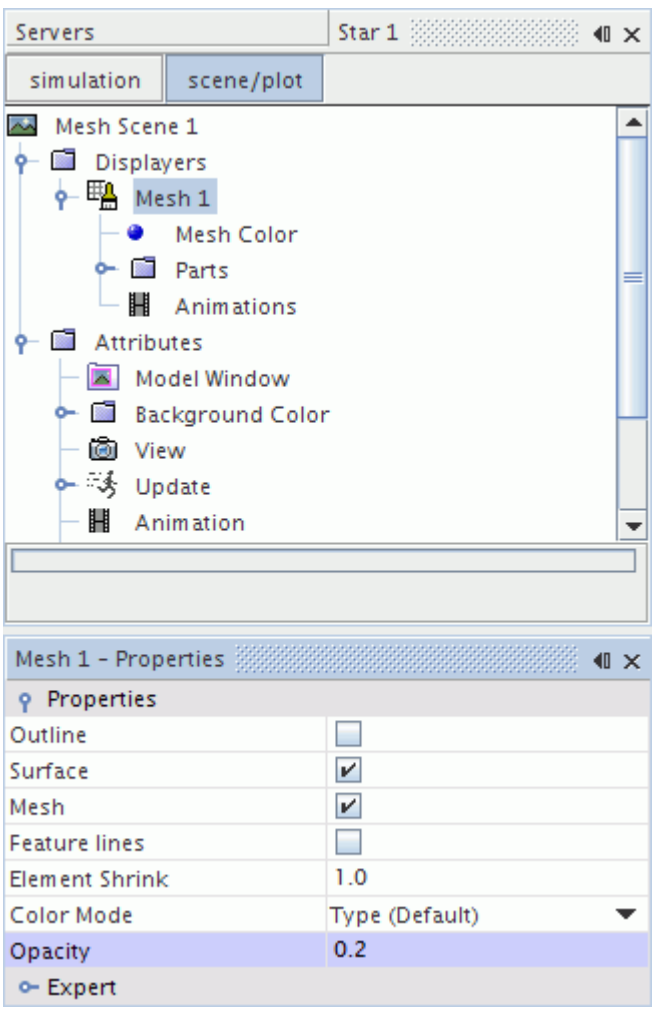
- Changing the Surface Opacity
- Displaying the Volume Mesh on a Plane Section

Changing the Surface Opacity

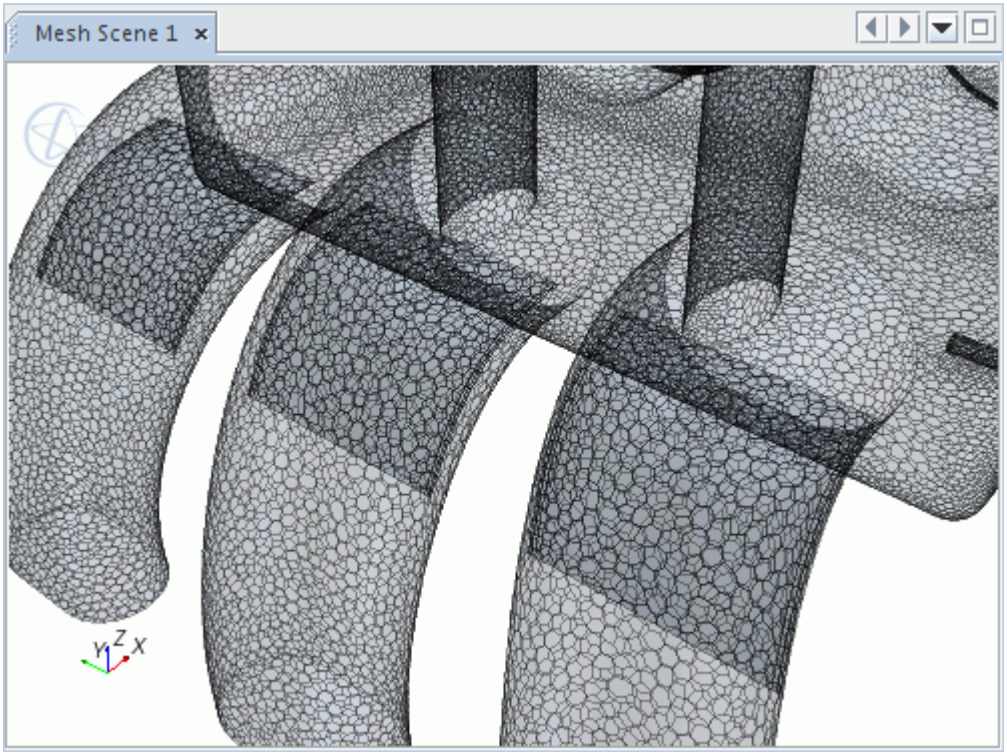
Changing the surface opacity allows you to view internal structures within a surface or volume mesh.

To change the surface opacity:

1. Make sure that you have a mesh scene open.
2. In the *Explorer Pane*, click **scene/plot**.
3. Select the **Displayer** > **Mesh** node and set the *Opacity* property.
   - A value of 1 gives a solid surface; lower values give more transparency



- The surface and mesh lines appear transparent and reveal internal surfaces with darker shades.



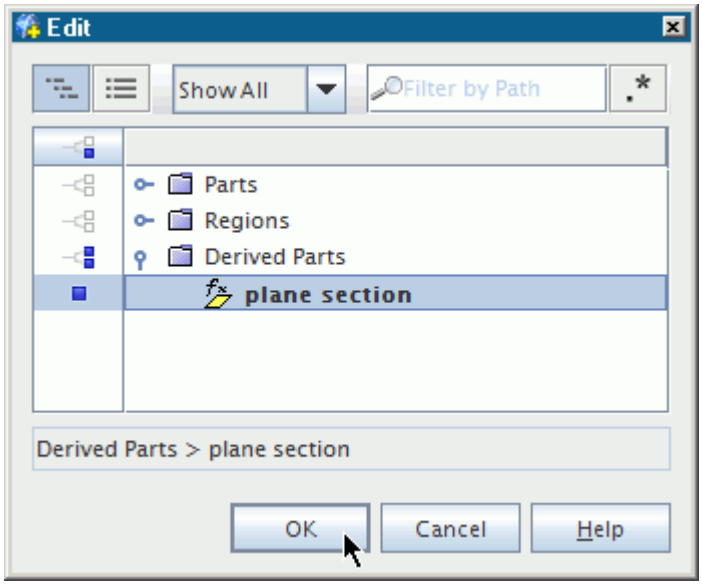
Displaying the Volume Mesh on a Plane Section

You can visualize the volume mesh on a plane section to view the internal cell structure.

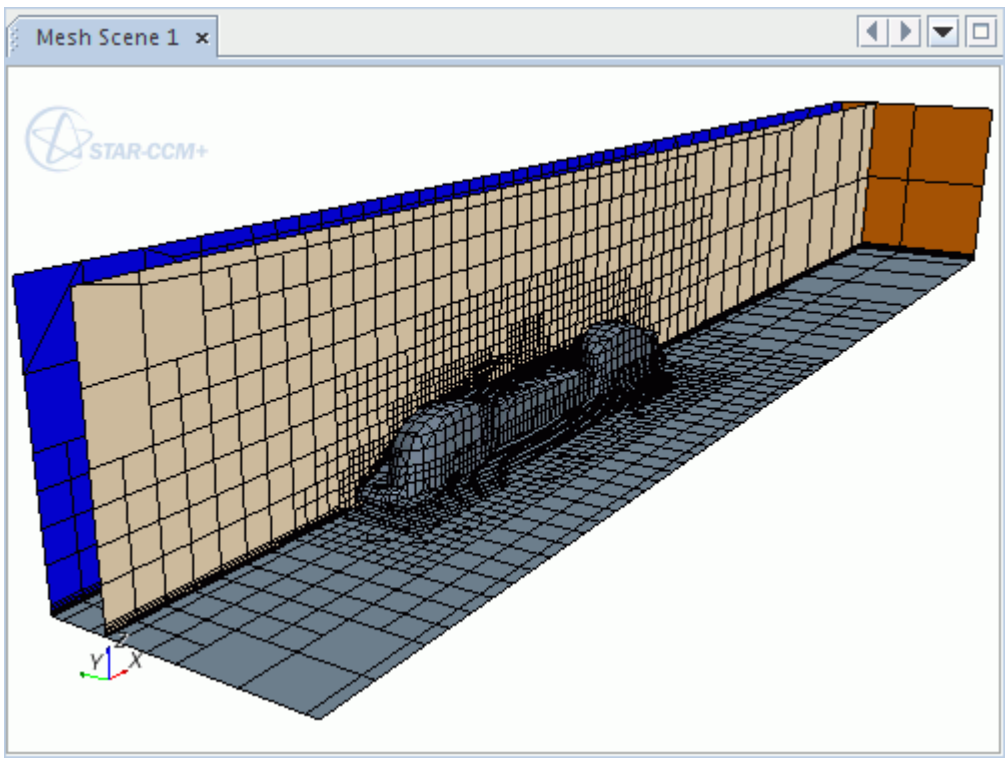
Note: As the plane section displays a two-dimensional cut through the volume mesh, some cells can appear incorrectly distorted.

To display the volume mesh on a plane:

1. Make sure that you have a mesh scene open.
2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane**.
   - You can also use a plane that you have created for other scenes.
3. In the *Create Section* in-place dialog, define the plane parameters.
4. In the *Explorer Pane*, click **scene/plot**.
5. Right-click the **Displayers** > **Mesh** > **Parts** node and select **Edit**.
6. In the object selection dialog, select the plane section.



The plane section displays the volume mesh where it cuts through the region.

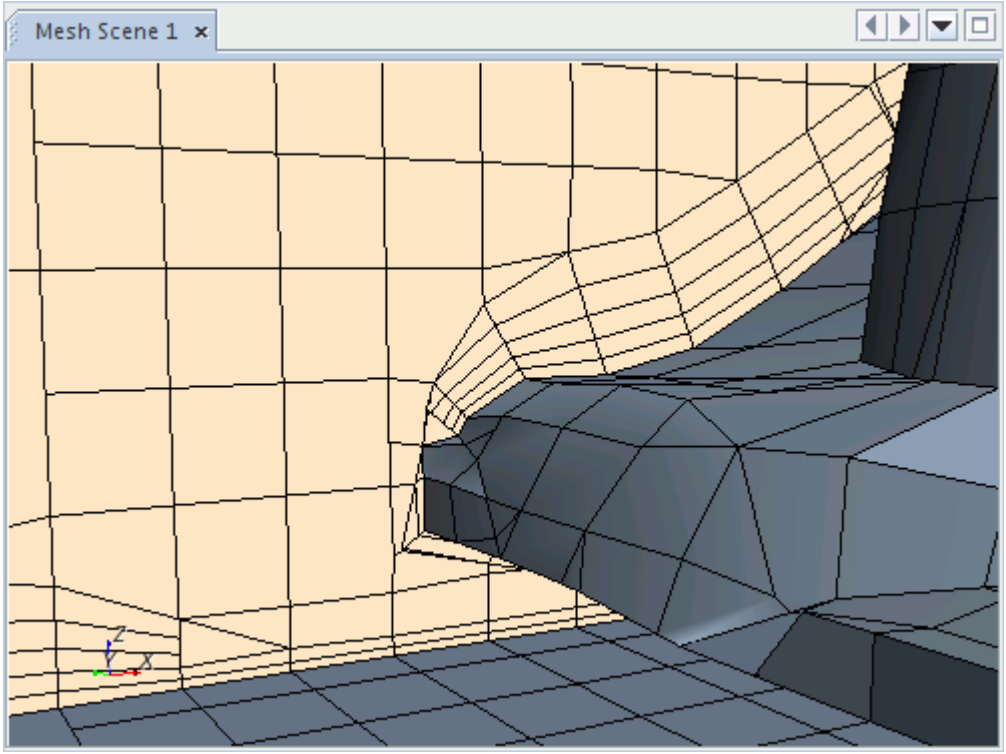


You can also sweep planes to view the mesh throughout the entire volume in an animation, or create multiple sections to view the mesh on several pre-defined planes with the same orientation.

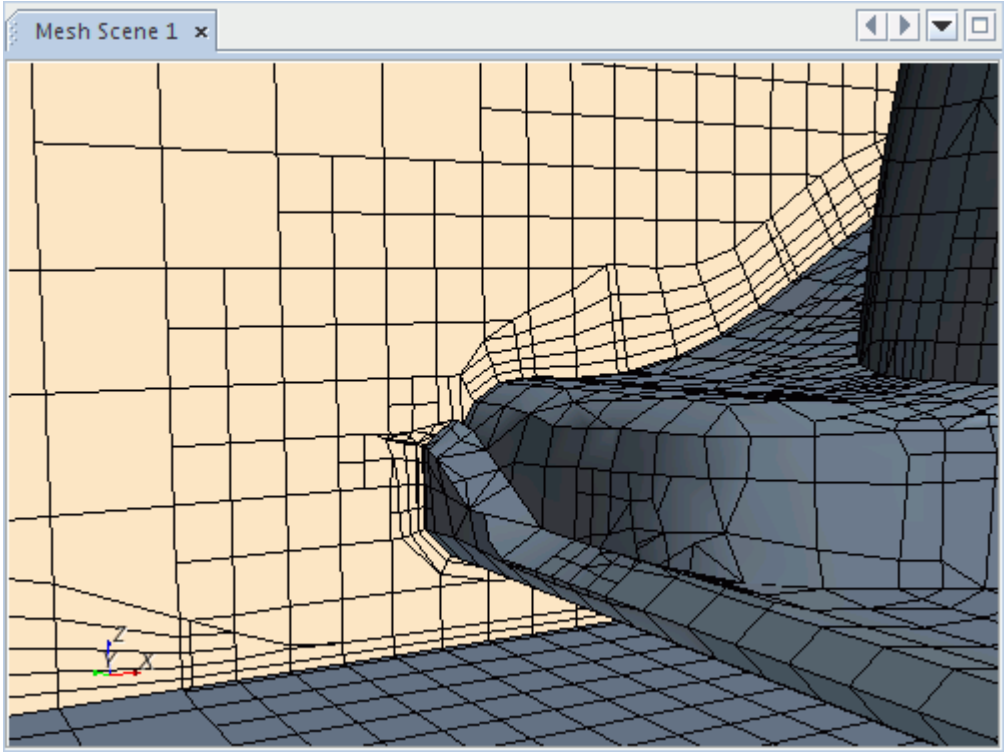
Example of Poor Surface Representation in a Volume Mesh

Plane sections are useful for viewing the internal cell structure around complex geometry to ensure that you have accurately captured the surface in the volume mesh.

The following image shows a volume mesh that does not accurately capture a complex surface.



By producing a finer mesh around the surface, you can improve the discretization. You can validate this improvement by displaying the new volume mesh on the same plane section.



Checking the Surface Mesh

The volume mesh quality is directly dependent on the surface mesh quality. Therefore, it is crucial to make sure that all surface meshes are of good quality.

STAR-CCM+ provides a range of tools which allow you to visualize, repair, and refine the surface mesh where necessary. You can also compare the level of mesh distortion between an original root description and a remeshed surface description.

See:

- Mesh Visualization
- Preparing Surfaces
- Manipulating Parts
- Automatic Surface Repair

Contents:

What is Face Quality?
Checking Mesh Distortion Levels

What is Face Quality?

Face Quality represents a value that is given for the quality of triangulated faces on a surface mesh.

The value that is given for face quality is determined by:

$$face\ quality = factor \times \frac{r}{R} \tag{87}$$

where:

$factor$ = 2.0

r = triangle inradius

R = triangle circumradius

The triangle inradius, r, is calculated using:

$$r = \frac{A}{s} \tag{88}$$

where:

A = triangle area

s = triangle half perimeter

The triangle circumradius, R, is calculated using:

$$R = \frac{a \times b \times c}{4 \times A} \tag{89}$$

where:

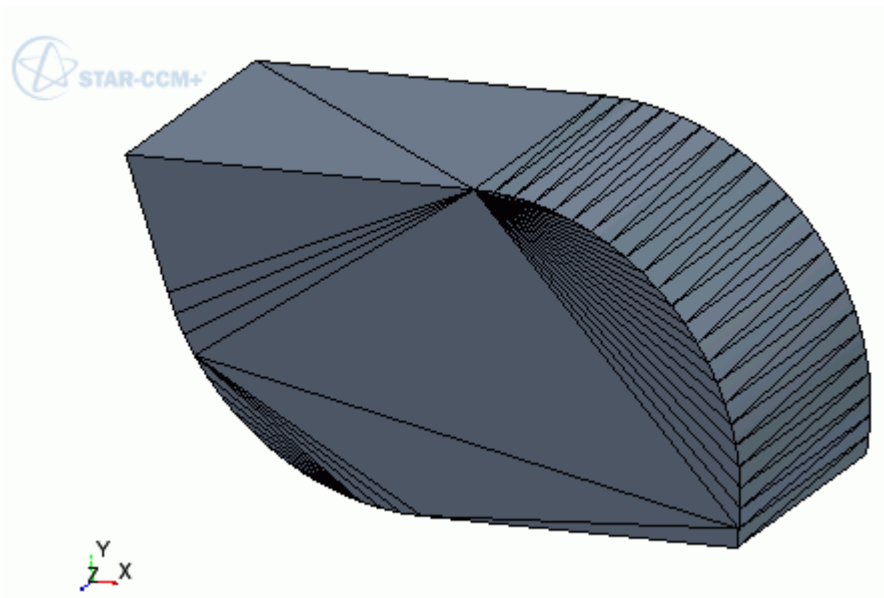
a, b, and c are the lengths of the triangle edges.

You can use the FaceQuality field function to display the face quality values on faces of a surface mesh. See Surface Meshing Field Functions Reference.

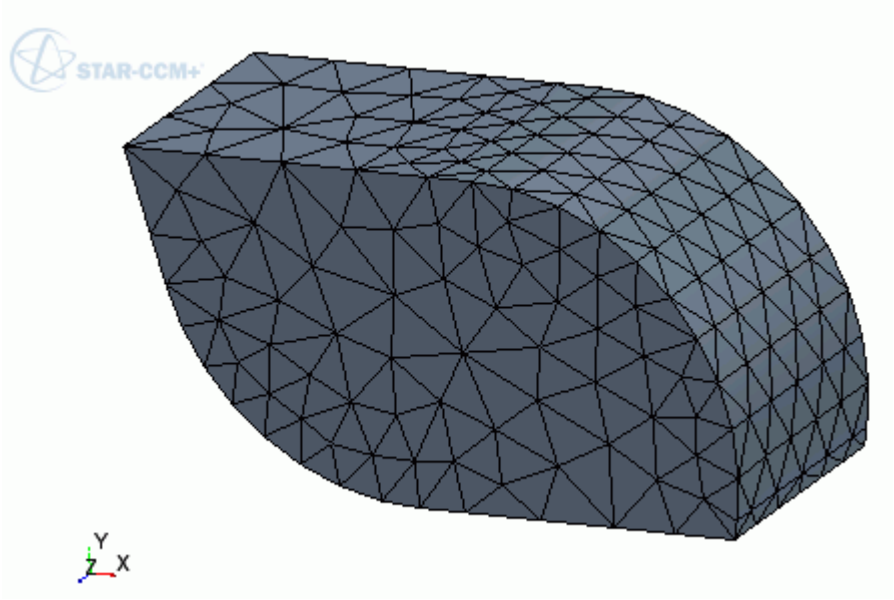
Checking Mesh Distortion Levels

STAR-CCM+ provides a series of field functions that allow you to visualize the distance by which one part-surface description deviates from an earlier description of the same part-surface. This deviation distance is the physical distance between the two triangulated surfaces. You set the reference part-surface in the field function that corresponds to the description you require, and set the comparison part-surface in a scalar scene that plots the function. The scalar scene must be set to use a later description than the one used by the field function you are plotting.

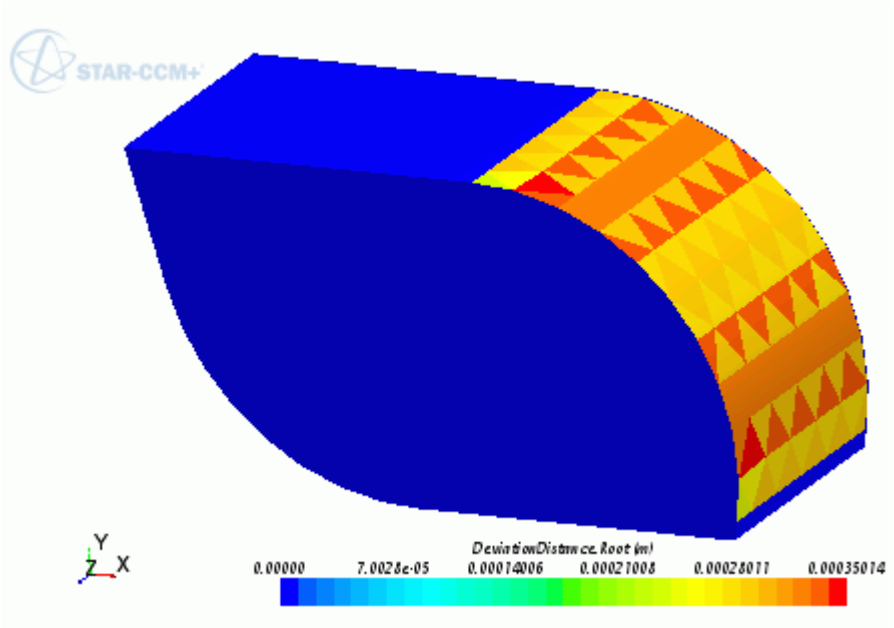
As an example, consider the tessellated surface for the root description of a part:



After remeshing in an Automated Mesh operation, a further description, **Automated Mesh.Remesh**, exists for the part surface:



By choosing this part surface in the **DeviationDistance.Root** field function, and plotting the field function in a scene that shows the **Automated Mesh.Remesh** description for the part surface, you can observe the deviation between the surfaces:



The following deviation distance field functions are available:

DeviationDistance.Root

Generated when geometry parts are imported or created. For the chosen part-surface, compares the description displayed in the scalar scene to the **Root** description.

DeviationDistance.Imprint

Generated when an Imprint operation is added to the meshing pipeline. For the chosen part-surface, this function compares the description displayed in the scalar scene to the **Imprint** description.

DeviationDistance.SurfaceWrapper

Generated for a Surface Wrapper operation that has *Per-Part Meshing* activated. Compares the description in the scalar scene to the **Surface Wrapper.Wrap** description.

DeviationDistance.AutomatedMesh

Generated when an Automated Mesh operation is added to the meshing pipeline. For the chosen part-surface, compares the description displayed in the scalar scene to the **AutomatedMesh.Remesh** description.

Before attempting to visualize the deviation distance between two triangulated surfaces, make sure that you have executed the relevant operations to create the description that you are assessing.

To visualize the deviation distance on a surface:

1. Expand the **Tools** > **Field Functions** node, select the deviation distance field function corresponding to the datum description, and set the *Parts* property to the part surface you are working with.
2. Create a scalar scene which contains the same part surface as the deviation distance field function.

   Note: the surface description chosen for the scalar scene must occur later in the pipeline than the description associated with the particular deviation field function you are using. Otherwise, no deviation distance is displayed.

3. For the scalar displayer of the scene, set the **Scalar Field** > *Function* to the particular Deviation Distance function that you require.

   The part in the scalar scene updates to display the level of mesh distortion on the triangulated surface.

   Note: You cannot delete descriptions manually—they are updated after related parts or operations are deleted.

Checking the Volume Mesh

This section describes the tools available for checking the validity and quality of the volume mesh, and for diagnosing problem areas. Such problems can then be fixed, either inside STAR-CCM+ or in the originating mesh generator.

When using an external mesh generator, it is important to ensure that the volume mesh file or files were imported properly, by reviewing the import progress report and visually inspecting the mesh.

After the mesh has been successfully imported or generated, check the coordinate extents of the mesh to verify that the imported mesh is scaled to the correct dimensions. Incorrect scaling of the mesh adversely affects the solution. The mesh diagnostic report allows you to verify the mesh scaling by checking the mesh extents. If there is a problem, STAR-CCM+ provides functionality to rescale the entire mesh accordingly.

A common symptom of an invalid mesh is zero or negative cell volumes. In some cases negative volume cells are due to incorrect face orientation, which must be addressed in the mesh generator. It is often worth verifying that there are no negative cell volumes before any interfaces are created, in order to remove the effects of the interfaces on the mesh. In addition, checks for near-zero area faces are also explained.

Even though a mesh is valid and the solution can be successfully initialized with an appropriate Space model, poor mesh quality can negatively impact the solution. The direct measures of mesh quality in STAR-CCM+ are:

- Cell Skewness Angle
- Boundary Skewness Angle
- Face Validity Metric
- Cell Quality Metric
- Volume Change Metric

A description of how to check the cell quality metric is provided, which is also applicable to checking and visualizing the other cell metrics. For more detailed diagnosis of mesh quality issues, the visualization tools in STAR-CCM+ can be used for identifying and visualizing bad cells.

In extreme cases, invalid cells can be removed so that you can continue running the analysis and obtain a solution.

Contents:

What Is a Valid Mesh?
Mesh Quality
Running Mesh Diagnostics
Checking The Mesh Validity
Comparing a Volume Mesh with the Geometry
Using The Connected Cells Field Function
Checking The Skewness Angle
Identifying And Visualizing Bad Cells
Removing Invalid Cells

What Is a Valid Mesh?

Fundamentally, a valid mesh is one that allows the simulation to be initialized and then run so that a solution is achieved. An invalid mesh is typically one for which no solution can be obtained or cannot even be initialized. As an example of the latter situation, a mesh that contains a negative volume cell would be considered invalid.

Based on that definition, the determination of whether a mesh is valid is not only dependent on the underlying mesh quality, but also the physics models in use with the mesh. For example, meshes with cells with high concave angles would not normally cause a problem, but in cases with high gradients or complex physics, a solution may not be obtainable. In this case, the mesh is considered invalid.

Other factors that can affect whether a mesh is valid include:

- choice of solver (coupled or segregated)
- under-relaxation or Courant number specification; and
- discretization scheme that is used

The approach that is used to define a CFD analysis is also important. For example, adding the physical models in one by one, iterating and restarting for each might lead to a valid solution. However, adding in all the physical models at once and attempting to run the simulation might cause the solution to diverge for a given mesh.

The mesh validity checks that are performed in STAR-CCM+ are:

- Unclosed cells;
- Invalid cell/vertex references;
- Zero area faces; and
- Zero or negative volume cells

Mesh Quality

STAR-CCM+ provides a number of cell metrics that you can use to determine the overall quality of the volume mesh. In general, a poor quality volume mesh does not necessarily result in solver run time problems, but it does reduce the accuracy and efficiency of the solution obtained.

When assessing the overall quality of the mesh, consider the following factors:

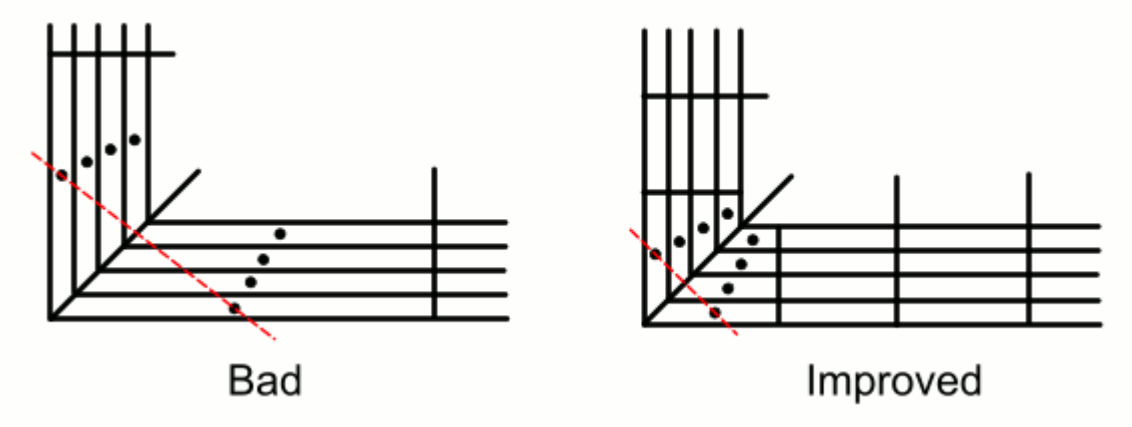
- **Mesh Density** - Is the mesh sufficiently dense in high-gradient areas to capture flow features accurately?
- **Mesh Distribution** - Is the distribution of the mesh between high- and low-gradient areas optimal?
- **Near Wall Layers** - Does the mesh contain near-wall layers for calculating turbulence and heat transfer effects?

On a local (cell to cell) level, consider the following factors:

- **Aspect Ratio** - Is the aspect ratio reasonable for the expected flow regime?
- **y+ Range** - Are the y+ values reasonable for the turbulence model selected?
- **Volume Ratio** - Is the ratio of a cell volume to its neighbor reasonable?
- **Skewness Angle** - Are the cell skewness angles extreme?

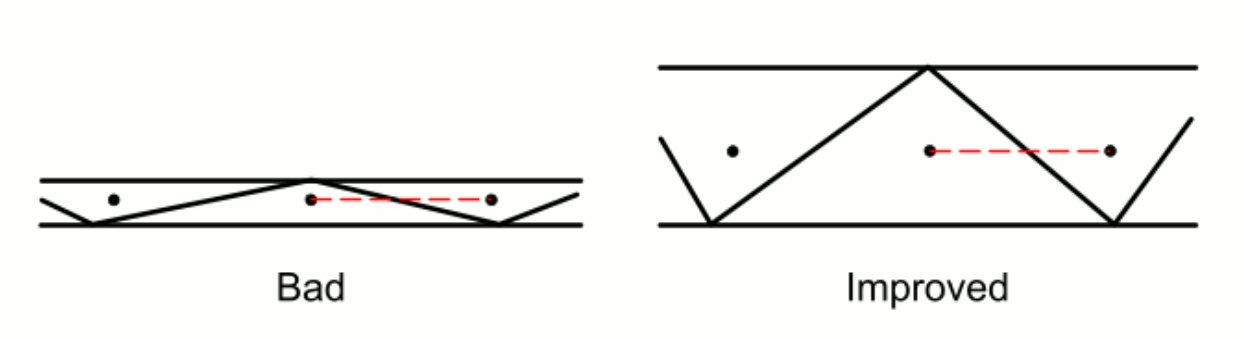
For fluid flow calculations, grid quality has most impact on the calculation of the convective and diffusive fluxes:

- To obtain convective fluxes with optimum accuracy and robustness, the line connecting the centroids of neighboring cells must ideally pass as closely as possible to the centroid of the face separating the cells. A two-dimensional example of this problem is shown below:



In this example, refining the mesh to create lower aspect ratio cells near the corner would improve the quality of the mesh.

- To obtain diffusive fluxes with optimum accuracy and robustness, the line connecting the centroids of neighboring cells must ideally be as close to orthogonal as possible to the face separating the cells. This angle is the cell skewness angle in STAR-CCM+. A two-dimensional example with high skewness angles is shown below:



In this example, refining the mesh to create lower aspect ratio cells would improve the quality of the mesh.

Mesh Metrics

There are many measures of cell quality, some of which are best suited to specific cell shapes. The metrics currently available in STAR-CCM+ are:

- Face Validity
- Cell Quality
- Volume Change
- Cell Skewness Angle
- Chevron Quality Indicator

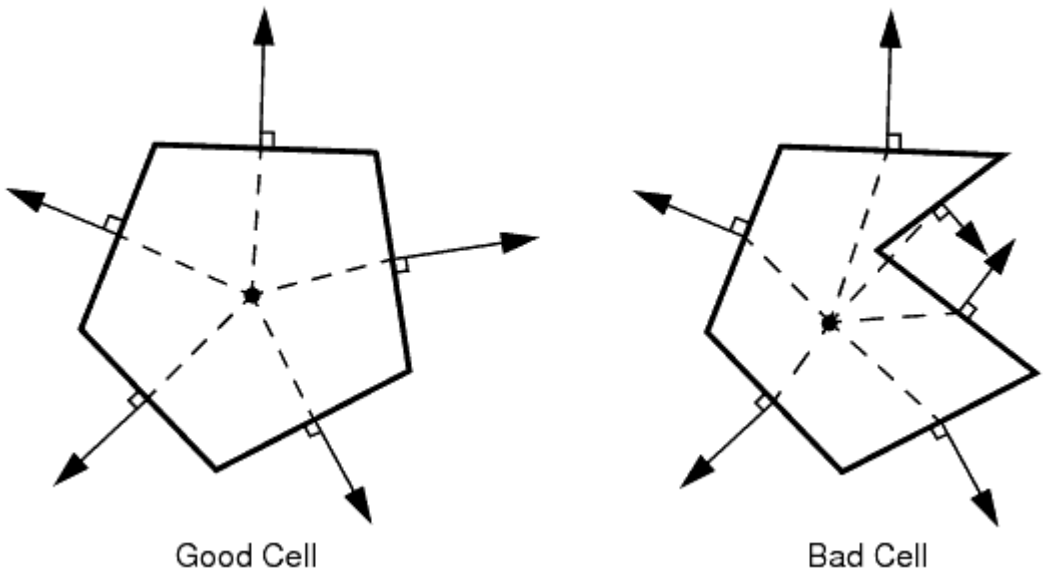
- Least Squares Quality
- Cell Warpage Quality

In order to create and visualize the face validity and volume change metrics, first initialize the solution with the appropriate physical models activated. For the above instances, you need only have the **Three Dimensional** space model selected before doing the initialization.

Each of the above metrics can be identified and visualized using a threshold as part of the geometry or scalar scene. See Identifying And Visualizing Bad Cells.

Face Validity

The face validity is an area-weighted measure of the correctness of the face normals relative to their attached cell centroid.

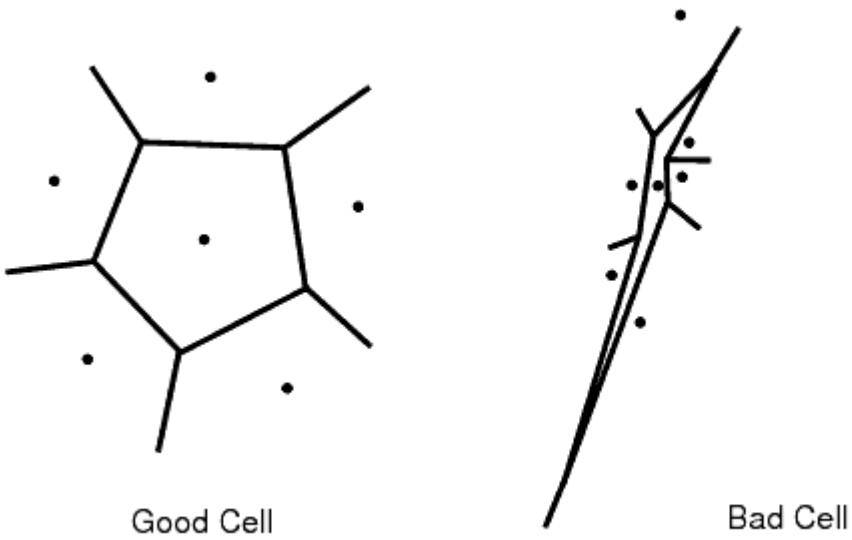


In a good quality cell, the face normals point outwards, away from the cell centroid. In a cell with bad face validity, one or more of the face normals point inwards, towards the cell centroid. A face validity of 1.0 means that all face normals are correctly pointing away from the cell centroid. Values below 1.0 mean that some of the cell faces have normals pointing inward towards the cell centroid, indicating some form of concavity. Values of below 0.5 signify a negative volume cell. Cells with a face validity below 1.0 are considered bad.

A minimum required value for the face validity metric can be entered on the dialog for removing invalid cells.

Cell Quality

The cell quality metric algorithm is based on a hybrid of the Gauss and least-squares methods for cell gradient calculation methods. It is a function not only of the relative geometric distribution of the cell centroids of the face neighbor cells, but also of the orientation of the cell faces. Generally, flat cells with highly non-orthogonal faces have a low cell quality.



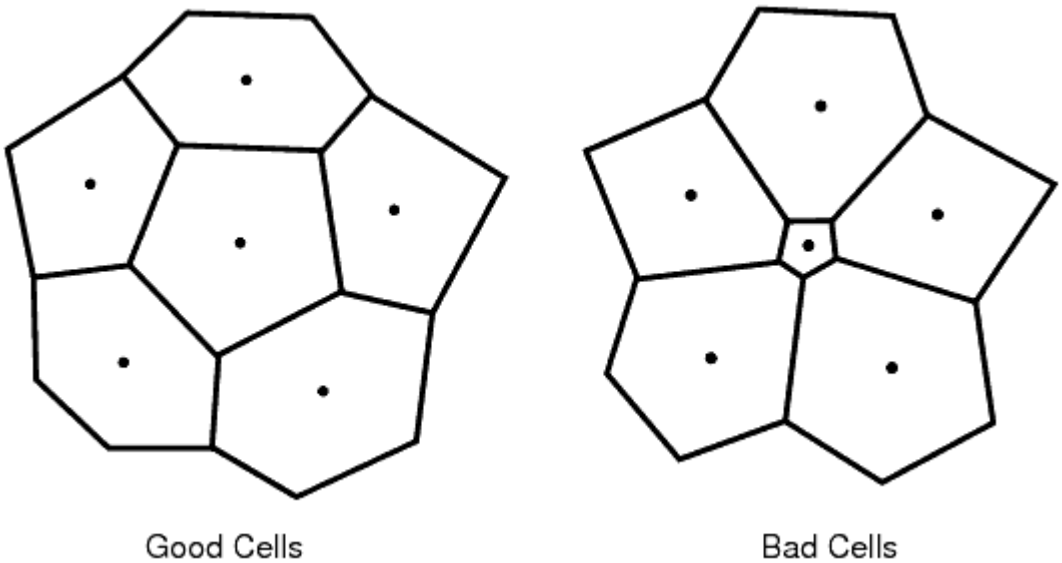
A cell with a quality of 1.0 is considered perfect. A cubic cell is an example of such a perfect cell, but other polyhedral cell shapes can also have a cell quality approaching unity. A degenerate cell has a cell quality approaching zero. Cells with a face quality less than 1.0e-5 are considered bad.

Depending on the physics that has been selected for the analysis, the cell quality of a cell can be fairly low and still provide a valid solution. However, poor cell quality is likely to affect both the robustness and accuracy of the solution. Cell quality is only defined for three-dimensional meshes.

A minimum required value for the cell quality metric can be entered on the dialog for removing invalid cells.

Volume Change

The volume change metric describes the ratio of the volume of a cell to that of its largest neighbor.



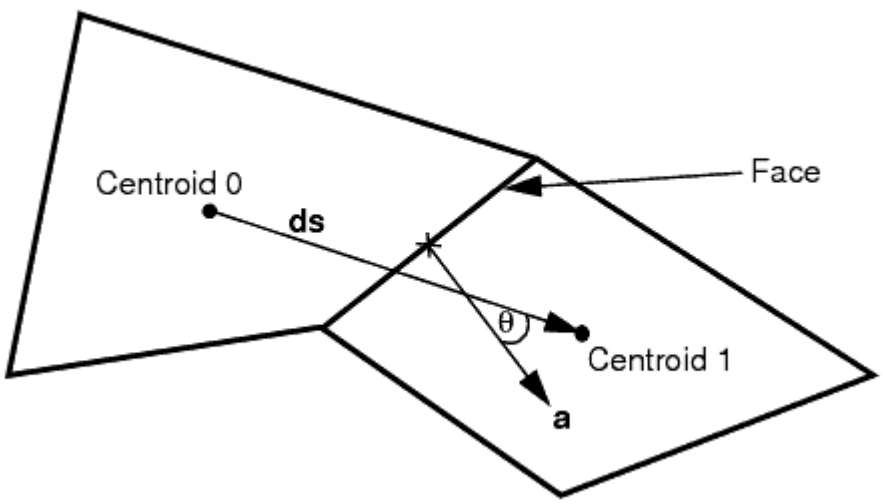
A value of 1.0 indicates that the cell has a volume equal to or higher than its neighbors. As the cell volume decreases relative to its neighbors (such as a sliver or flat cell), then the volume change metric can be used to flag this cell. A large jump in volume from one cell to another can cause potential inaccuracies and instability in the solvers. Cells with a volume change of 0.01 or lower are considered bad cells.

A minimum required value for the volume change metric can be entered on the dialog for removing invalid cells.

Cell Skewness Angle

This skewness measure is designed to reflect whether the cells on either side of a face are formed in such a way as to permit diffusion of quantities without these quantities becoming unbounded.

To understand this measure, consider the following two-dimensional diagram of a face and the cell centroids on either side of the face:



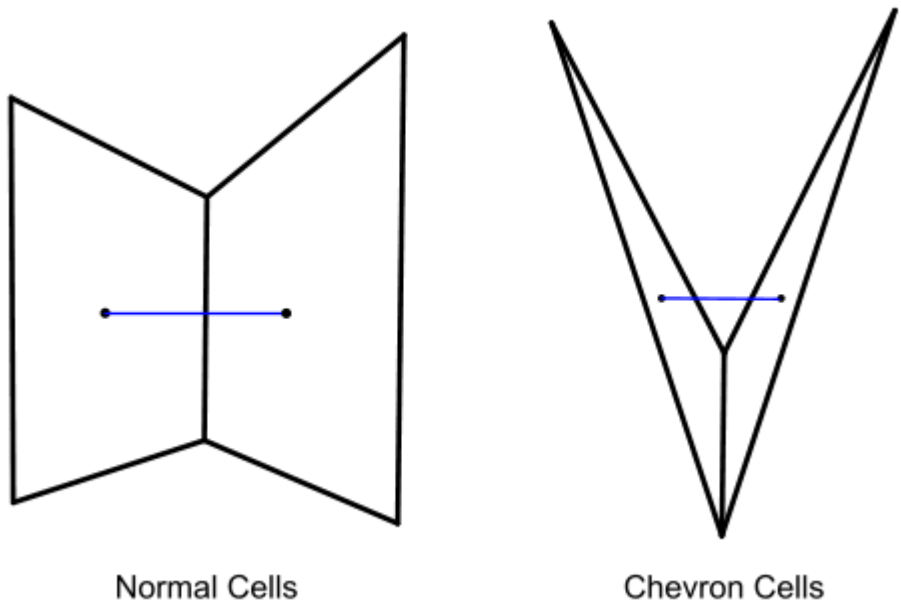
The skewness angle θ is the angle between the face area vector $\mathbf{a}$ (face normal) and the vector connecting the two cell centroids, $d\mathbf{s}$. An angle of zero indicates a perfectly orthogonal mesh. Variables that are stored at interior faces in STAR-CCM+ cannot be displayed; instead, the worst skewness angle for all the faces of a cell are stored in that cell. Thus, two cells often have the same skewness angle corresponding to the face that they share.

Cells with a skewness angle greater than 85° are considered bad cells.

Skewness angles of 90° or greater (which can occur in concave cells where the centroid of both cells lies on the same side of the boundary face), typically result in solver convergence issues. Problems result because the diffusion term for transported scalar variables contains the dot product $\mathbf{a} \cdot d\mathbf{s}$ in the denominator, and this dot product is zero when the angle between $\mathbf{a}$ and $d\mathbf{s}$ is 90°. Even though STAR-CCM+ takes care to avoid such a divide-by-zero error, the accuracy of the diffusion calculation is reduced, leading to a loss of robustness in such cases.

Chevron Quality Indicator

Chevron cells are pairs of thin slender cells which meet at a common face at an angle such that the line joining the cell centers does not pass through the common face.



A chevron quality norm, c, is used to determine whether a cell can be classified as a chevron cell or not:

$$c = 1 - \frac{d\mathbf{p} \cdot d\mathbf{p}}{max(d\mathbf{p} \cdot d\mathbf{v})}$$

where:

- $d\mathbf{p}$ is a vector that is projected from the face center to the line joining the cell centers.
- $d\mathbf{v}$ is the vector from the face vertex to the face center.
- The max function, in evaluating the chevron quality norm, is taken over all the face vertices.

The chevron quality norm is computed using a vector $d\mathbf{p}$:

$$d\mathbf{p} = \left(\frac{\mathbf{A}_f \cdot d\mathbf{x}}{\mathbf{A}_f \cdot d\mathbf{s}}\right) d\mathbf{s} - d\mathbf{x}$$

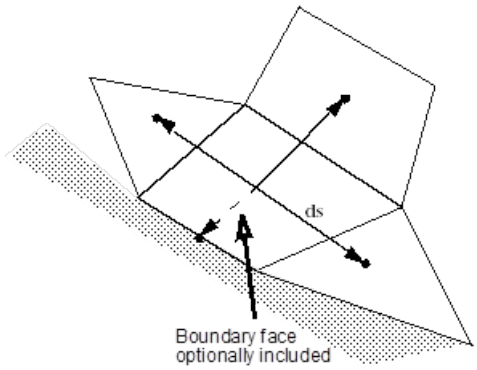
where:

- $\mathbf{A}_f$ is the face area vector.
- $d\mathbf{x}$ is the vector from the face center to the cell center.
- $d\mathbf{s}$ is the vector connecting the two cell centers.

Chevron cells are marked as 1.0 and every other cell as 0. Chevron cells are considered bad cells.

Least Squares Quality

This field function is an indicator of the quality of a cell, using the physical location of a cell centroid relative to the cell centroid locations of its face-neighbors.



The field function value is based on the ratio of the maximum and minimum eigenvalue ratios of the symmetrical matrix:

$$\overline{\overline{M}} = \sum_{\text{cell faces } i}^{n} \frac{d\mathbf{s_i} \otimes d\mathbf{s_i}}{\left|d\mathbf{s_i}\right|^2}$$

where:

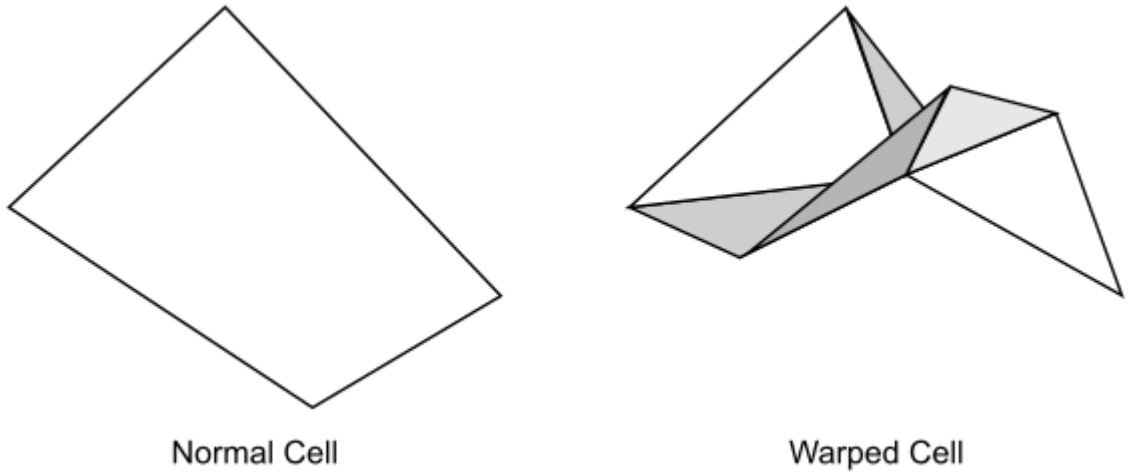
- $d\mathbf{s}_i$ represents the vector from the cell to the neighbor centroid.

Two methods are optionally provided to evaluate the eigenvalue ratio: approximate and exact. The exact method uses a Jacobi transformation. The approximate method uses the Frobenius norm of the matrix to estimate the eigenvalue ratio. It is slightly more economical to evaluate than the exact method, and is guaranteed to be less than or equal to the exact value. A perfect cell has a Least Squares Quality computed by the exact algorithm equal to unity. The same perfect cell would have an approximate value of 1/N, where N is the dimension of the mesh (2 or 3). As the Least Squares Quality becomes small (< 0.1), the approximate and exact values yield almost identical results. An option also exists to include the boundary faces of the cell in the computation for boundary cells.

Cells with a least squares quality less than 1.0e-3 are considered bad.

Cell Warpage Quality

Cell warpage quality identifies thin and warped cells. These types of cells can cause issues for the flow solver, especially for high fidelity simulations such as LES/DES and acoustics, where minimal dissipation is required.



The warpage quality is based on the measured cell warpage angle of the face. The cell warpage angle is defined by:

$$2 * \arctan\left(\frac{d}{L}\right)$$

where:

- d is the reciprocal distance projected on the face normal, which is defined by $d = \max\left[\left|\left(\mathbf{x}_i - \mathbf{x}_{i+1}\right) \cdot \mathbf{n}\right|\right]$. The maximum is taken over the successive nodes i of the face, and n is the normal unit vector of the face.
- L is the characteristic length scale of the face, which is defined by $L = \sqrt{Area}$. Area is the area magnitude of the face.

The cell warpage angle is non-linearly scaled and transformed into the cell warpage quality. The cell warpage quality ranges from 0.0, which corresponds to the worst warpage, to 1.0, which corresponds to no warpage.

Cells with a cell warpage quality lower than 0.15, which corresponds to a cell warpage angle of 50°, are considered bad cells. In general, the smaller the cell warpage angle, the better the quality is.

Only cells that have an aspect ratio smaller than 0.2 can have a warpage quality of less than 1.0.

Cell warpage quality is not included in the Cell Quality Remediation or Bad Cell Detection models. You can manually add cell warpage quality to these criteria through a Java macro, where the minimum value is 0.15 and the maximum value is 1.0.

Running Mesh Diagnostics

This section describes how to run a mesh diagnostic report and how to interpret the output.

A mesh diagnostics report is a useful way of checking the validity of the mesh and getting statistics on the mesh quality. Two different types of reports are possible, a compact one with basic information on the model

and a more extensive full report with more information on the boundaries. It is also possible to produce a diagnostics report for a surface mesh.

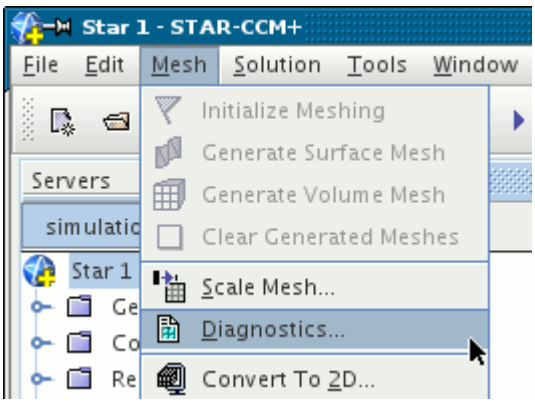
Contents:

Running the Mesh Diagnostics Report
Interpreting the Compact Report
Interpreting the Full Report
Interpreting the Report for a Surface Mesh

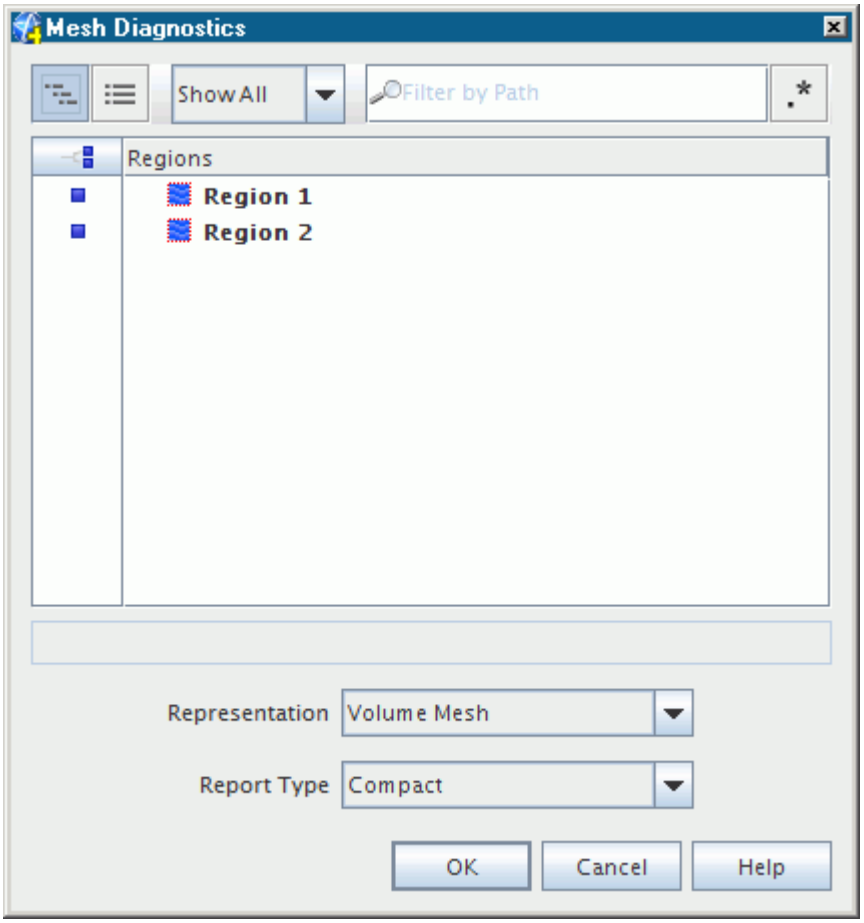
Running the Mesh Diagnostics Report

A mesh diagnostics report is a useful way of checking the validity of the mesh.

To run a mesh diagnostics report for a volume mesh, select the **Mesh** > **Diagnostics...** menu option.



The *Mesh Diagnostics* dialog then opens with all regions selected.



Deactivate the checkbox of any regions for which you do not want a report. The **Representation** selection lists all the current representations available that a diagnostics report can be produced for. If a volume mesh exists, the **Volume Mesh** option is selected by default, but a surface mesh representation can be selected instead. Select a **Report Type**, either **Compact** or **Full**.

Pressing the **OK** button generates the specified report and displays it in the *Output* window of the current simulation. The information that is displayed is dependent on whether the compact or full option is selected for **Report Type**. For a surface mesh representation, a surface mesh report is produced. In all instances information summarizing the entity count, mesh extents and mesh quality is produced.

If a Space model has not been selected before running the mesh diagnostics report for a volume mesh, then a three-dimensional physics model is automatically created in the **Continua** node for the simulation.

Interpreting the Compact Report

A compact mesh report for a volume mesh consists of statistics that are split up into statistics for each mesh region, and overall statistics.

For each region, the following subsections exist:

- Entity count
- Mesh extents
- Mesh validity check

- Face validity
- Volume change

The overall section also consists of the above items for all regions, minus the mesh extents summary. An example compact output for a single region is shown below:

```
Output - Star 1
Loading module: MetricsModel
--------------------------------------------------
--- Computing statistics in Region: Region 1
--------------------------------------------------
-> ENTITY COUNT:
   # Cells: 53192
   # Faces: 276966
   # Verts: 226439
-> EXTENTS:
   x: [ 0.0000e+00,  4.3180e-01] m
   y: [-1.0155e-01,  0.0000e+00] m
   z: [-1.0160e-01,  1.0160e-01] m
-> MESH VALIDITY:
   Mesh is topologically valid and has no negative volume cells.
-> FACE VALIDITY STATISTICS:
Minimum Face Validity: 1.000000e+00
Maximum Face Validity: 1.000000e+00
        Face Validity <  0.50          0    0.000%
0.50 <= Face Validity <  0.60          0    0.000%
0.60 <= Face Validity <  0.70          0    0.000%
0.70 <= Face Validity <  0.80          0    0.000%
0.80 <= Face Validity <  0.90          0    0.000%
0.90 <= Face Validity <  0.95          0    0.000%
0.95 <= Face Validity <  1.00          0    0.000%
1.00 <= Face Validity              53192   100.000%
-> VOLUME CHANGE STATISTICS:
Minimum Volume Change: 3.741970e-03
Maximum Volume Change: 1.000000e+00
         Volume Change <  0e+00          0    0.000%
0e+00 <= Volume Change <  1e-06          0    0.000%
1e-06 <= Volume Change <  1e-05          0    0.000%
1e-05 <= Volume Change <  1e-04          0    0.000%
1e-04 <= Volume Change <  1e-03          0    0.000%
1e-03 <= Volume Change <  1e-02         15    0.028%
1e-02 <= Volume Change <  1e-01       1963    3.690%
1e-01 <= Volume Change <= 1e+00      51214   96.281%
--------------------------------------------------
--- Computing statistics in Region: Region 2
--------------------------------------------------
```

For a compact report, the entity count includes a summary of the number of cells, faces, and vertices for the region. A more extensive breakdown of the different cell shapes used is provided in the full report. See Interpreting the Full Report.

The mesh extents are always reported in meters based on the global Cartesian system. STAR-CCM+ computes all physical quantities in SI and therefore assumes that the model is dimensioned correctly in meters. If a mesh was created using some other unit and was not properly scaled before import, it will result in incorrect physics being simulated. The mesh extents are originally reported at the end of the import report in the *Output* window for the simulation. Verify the coordinate extents for each region to make sure that they correspond with the expected dimensional range. If the extents are incorrect, you can:

- Scale the mesh in the original meshing pre-processor and re-import.
- Scale the entire mesh in STAR-CCM+. See Scaling the Entire Mesh.
- Scale a specific region in STAR-CCM+. See Scaling Regions.

Once you have scaled the mesh or region, its worth running the mesh diagnostics report and checking the mesh extents again.

Interpreting the Full Report

The full diagnostics report is structured in a similar way to the compact report and contains all the information the compact report produces.

In addition however, the full report includes the following extra items for each region and the overall summary:

- Boundary details
- Entity count per cell shape
- Volume range
- Maximum interior skewness angle

The boundary details can be further split up as follows:

- Number of faces
- Boundary extent
- Surface area
- Maximum boundary skewness angle

An example output of the full report with the boundary information is shown below:

```
Output - Star 1

Boundaries of region Region 1:
Boundary Interface: 18568 faces (378 quadrilateral, 18190 polygonal)
 Extents:
  x: [0.0000000000e+00, 4.3180000000e-01] m
  y: [-1.0155037851e-01, 0.0000000000e+00] m
  z: [-1.0160000000e-01, 1.0160000000e-01] m
 surface area: 2.2217960656e-01 m^2
 maximum boundary skewness angle = 5.2533542633e+01 deg
Boundary Inlet: 60 faces (23 quadrilateral, 37 polygonal)
 Extents:
  x: [4.3180000000e-01, 4.3180000000e-01] m
  y: [-1.8945986797e-02, 0.0000000000e+00] m
  z: [1.9050000000e-02, 5.7150000000e-02] m
 surface area: 5.6100578513e-04 m^2
 maximum boundary skewness angle = 3.7604660034e+01 deg
Boundary Outlet: 61 faces (26 quadrilateral, 35 polygonal)
 Extents:
  x: [4.3180000000e-01, 4.3180000000e-01] m
  y: [-1.8958928660e-02, 0.0000000000e+00] m
  z: [-5.7150000000e-02, -1.9050000000e-02] m
 surface area: 5.6184205459e-04 m^2
 maximum boundary skewness angle = 3.4550983429e+01 deg
Boundary Symmetry: 3108 faces (803 quadrilateral, 2305 polygonal)
 Extents:
  x: [0.0000000000e+00, 4.3180000000e-01] m
  y: [-3.9837506798e-17, 0.0000000000e+00] m
  z: [-1.0160000000e-01, 1.0160000000e-01] m
 surface area: 3.0161229894e-02 m^2
 maximum boundary skewness angle = 5.7758831024e+01 deg
Boundary Interface [In-place 1]: 0 faces
```

An example output of the full report with the other additional information is shown below

```
Output - Star 1
Region Region 1:
     430 hexahedral cells
      57 wedge cells
   52705 polyhedral cells
   53192 cells total
  276966 interior faces (9562 triangular, 120452 quadrilateral, 146952 polygonal)
  226439 vertices
 Extents:
  x: [0.0000000000e+00, 4.3180000000e-01] m
  y: [-1.0155037851e-01, 0.0000000000e+00] m
  z: [-1.0160000000e-01, 1.0160000000e-01] m
 Maximum interior cell index delta: 459, average: 1.4594911650e+02
 Maximum cell face index delta: 2230, average: 1.2235293277e+03
 Volume range: [1.9417767394e-10, 5.1441674032e-07] m^3
 Minimum distance between centroids of neighbor cells = 6.1174513819e-04
Calculating skewness angle in Region 1
 Maximum interior skewness angle = 7.4360519409e+01 deg
```

Interpreting the Report for a Surface Mesh

The mesh diagnostics report for any of the surface type representations (import, wrapped surface, remeshed surface, or extracted surface) consists of sections for each boundary and each region.

Each of these sections has subsections for the following:

- Triangles, vertices, and marked feature edges count
- Boundary or region extents
- Boundary or region triangle quality
- Edge length statistics

Triangle quality is listed in 10 separate bands from 0 to 1, with 1 being the best quality. The edge length statistics are useful for gauging initial base values to be used as input for the surface wrapper and/or surface remesher tools.

An example output is shown below:

Output - Star 1

```
***** Mesh report for representation: Import
---------------------------------------------------
Boundary Wall Boundaries:
    4438 triangles
    2221 vertices
     165 marked feature edges
Extents:
  x: [-2.7559058873e-01, 1.0254813695e-02] m
  y: [-1.1218700822e-01, 1.1219020904e-01] m
  z: [-8.2640535249e-02, 1.4221760232e-01] m
Boundary triangle quality:
 Face Quality    Faces    Percent
  0.0 - 0.1          0       0.0
  0.1 - 0.2          0       0.0
  0.2 - 0.3          0       0.0
  0.3 - 0.4          0       0.0
  0.4 - 0.5          1       0.0
  0.5 - 0.6          1       0.0
  0.6 - 0.7          0       0.0
  0.7 - 0.8          3       0.1
  0.8 - 0.9         24       0.5
  0.9 - 1.0       4409      99.3
 Worst quality is 4.9e-01 on face 4438
 Edge lengths:
  Min: 5.4e-03
  Avg: 1.1e-02
  Max: 1.8e-02
---------------------------------------------------
Region Region 1:
    4438 triangles
    2221 vertices
     165 marked feature edges
Extents:
  x: [-2.7559058873e-01, 1.0254813695e-02] m
  y: [-1.1218700822e-01, 1.1219020904e-01] m
  z: [-8.2640535249e-02, 1.4221760232e-01] m
Region triangle quality:
 Face Quality    Faces    Percent
  0.0 - 0.1          0       0.0
  0.1 - 0.2          0       0.0
  0.2 - 0.3          0       0.0
  0.3 - 0.4          0       0.0
  0.4 - 0.5          1       0.0
  0.5 - 0.6          1       0.0
  0.6 - 0.7          0       0.0
  0.7 - 0.8          3       0.1
  0.8 - 0.9         24       0.5
  0.9 - 1.0       4409      99.3
 Worst quality is 4.9e-01 on face 4438
 Edge lengths:
  Min: 5.4e-03
  Avg: 1.1e-02
  Max: 1.8e-02
```

Checking The Mesh Validity

The mesh validity is automatically checked whenever a mesh diagnostics report is produced or when the simulation is initialized for the first time.

STAR-CCM+ performs the following mesh validity checks:

- Unclosed cells;
- Invalid cell/vertex references;
- Zero area faces; and
- Zero or negative volume cells.

If any of the above errors are found, except for near-zero area faces, then the mesh is invalid and must be fixed or regenerated in order to continue with the simulation. Alternatively, if the problem area is next to a surface (most issues usually are) improve the surface mesh in that area by hand before regenerating the volume mesh.

Contents:

What Are Unclosed Cells
What Are Invalid Cell Vertex References
Checking For Zero Area Faces
Checking For Zero Or Negative Volume Cells

What Are Unclosed Cells

A topologically closed volume cell consists of edges that are connected to exactly two faces, and faces with normals pointing outwards. If a cell is missing a face, or the outward normals are inconsistent, it is considered unclosed and invalidates the mesh.

Sometimes this problem originates from the mesh exporter/importer or from the mesh generator itself. Fix any unclosed cells in the mesh by either deleting the cell and replacing it with a valid cell, or orienting the face normals in the outward direction.

Unclosed cells (as defined above) can be created during interface generation process as the intersecting vertices are not included in the vertex lists of the adjacent interior faces. In most instances, unclosed faces that are reported after interface generation are not a problem.

Unclosed cells can be visualized by creating a threshold part for cells with an unmatched edge count greater than zero. To create a new threshold part, set the Scalar field function to `Unmatched Edge Count`, set the Extraction Mode to **All Above** and enter a **Threshold Value** of `0`.

What Are Invalid Cell Vertex References

Invalid cell or vertex references usually originate from the mesh exporter/importer or the mesh generation process and must be fixed before proceeding any further. Deleting the cell and replacing it is usually the quickest and easiest solution to this problem.

STAR-CCM+ stores the mesh connectivity as an array of cells that are adjacent to each face together with a list of vertices belonging to the face. In a valid mesh, each cell interior face belongs to exactly two cells. Invalid cell and vertex references arise when the arrays storing the cell connectivity and vertices have entries that make reference to faces or vertices that fall outside the expected range of values.

Checking For Zero Area Faces

It is possible to generate a volume mesh containing cells that have faces of zero area magnitude. This situation would occur if all the vertices belonging to a face were coincident or collinear.

Since most calculations performed on faces involve flux calculations, a zero area face simply results in a zero value for the relevant flux. However, some calculations require a face unit normal, which is defined as the face area vector normalized by its magnitude. For this reason, STAR-CCM+ limits the face area magnitude that is used to a finite small value (less than 10^{-10} for a single-precision simulation) to prevent a divide-by-zero error. Zero area faces therefore do not usually cause a problem and can be ignored.

A face area calculation is performed whenever a mesh diagnostics report is produced or when the simulation is initialized. If any zero area faces are found, then the following message is displayed:

```
Output - Star 1
Initializing metrics
Loading module: MetricsModel
Calculating cell metrics in region Volume Mesh:Region_1
Info: mesh contains 1 zero-area faces.
-----------------------------------------------------
--- Computing statistics in Region: Region_1
-----------------------------------------------------
-> ENTITY COUNT:
   # Cells: 36953
   # Faces: 106020
   # Verts: 43669
-> EXTENTS:
```

If desired, you can identify and visualize the cells that have near zero, or zero face areas.

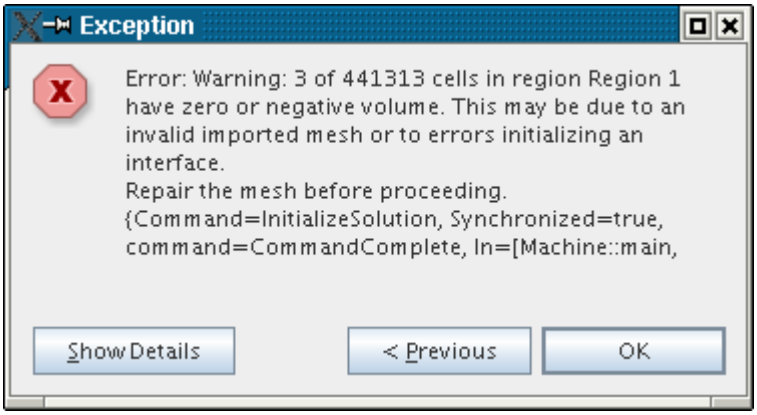
Checking For Zero Or Negative Volume Cells

For a mesh to be valid for finite volume calculations, each cell must have a positive volume, otherwise the initialization process fails, and the simulation cannot be run. Negative volume cells can occur in a number of different ways but must be fixed or removed from the mesh in order to proceed any further.

1. To check if the volume mesh has zero or negative volume cells, you can either run the mesh diagnostics report or initialize the solution. An example report output for a simulation containing negative volume cells is shown below:

```
Output - Star 1
   # Verts: 1548971
-> EXTENTS:
   x: [ 3.2635e+01,  3.3716e+01] m
   y: [-1.2690e+01, -1.1889e+01] m
   z: [ 1.2094e+00,  2.5007e+00] m
-> MESH VALIDITY:
   Mesh is INVALID:
    # Negative Volume Cells: 3
-> FACE VALIDITY STATISTICS:
Minimum Face Validity: 5.621198e-01
Maximum Face Validity: 1.000000e+00
```

The total number of zero and negative volume cells are reported, along with a message reporting that the mesh is invalid. During the initialization step, an *Exception* dialog is displayed if there are zero or negative volume cells in the mesh:



It is recommended that you check for zero or negative volume cells by running the mesh diagnostics report immediately after importing or generating a volume mesh in STAR-CCM+. Do so avoids wasting time setting up a simulation for a mesh that is not valid.

Cells that have zero or negative volumes can be identified and visualized using threshold values.

Contents:

How Do Negative Cell Volumes Occur

How Do Negative Cell Volumes Occur

Negative volume cells occur for a number of different reasons.

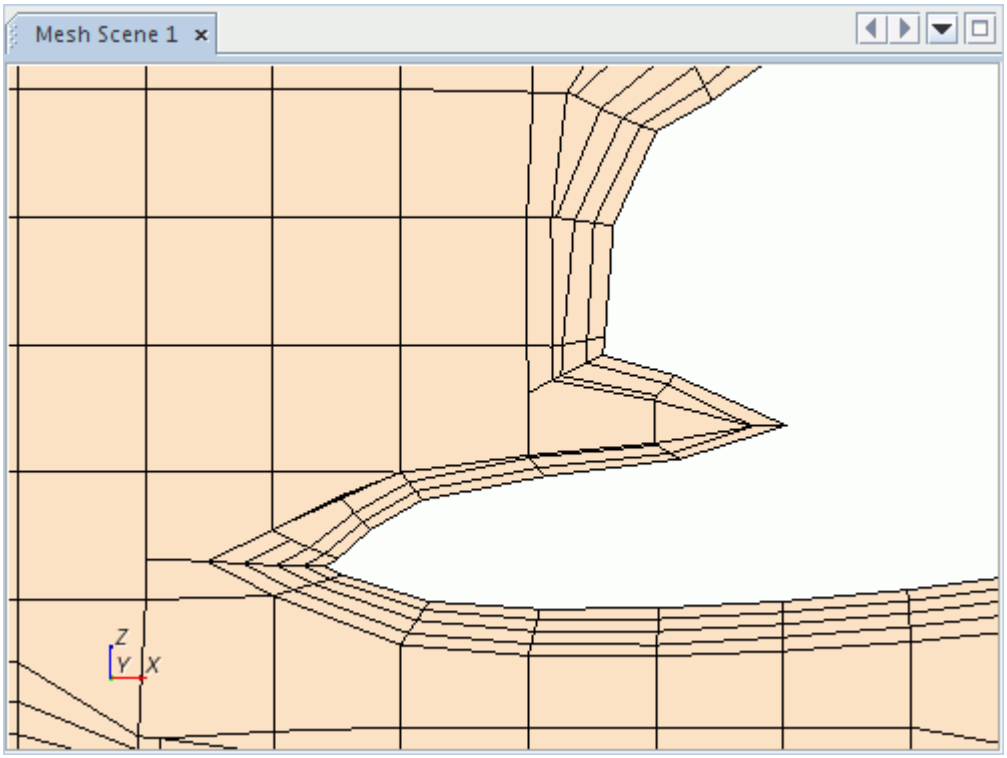
These reasons include:

- Incorrect face orientation, which occurs if the pre-processing package used to create the mesh writes out the vertices using a left-handed rule, rather than the right-handed rule that STAR-CCM+ expects. In this situation, every cell in the region has a negative volume reported. Some tools, such as pro-STAR, have operations that can correct this problem and turn left-handed cells into right-handed ones.
- Badly formed cells. For hexahedral cells, the faces of high-aspect ratio cells can be highly warped leading to poorly defined face and cell centroids. For polyhedral shapes, negative volume cells can occur due to the dualization of poor quality tetrahedral cells, resulting in badly formed, and sometimes twisted shapes. Once you have identified these cells, go back to the pre-processor to repair them or remove them directly in STAR-CCM+.
- Interfaces that are created from incorrect or misplaced boundaries. Make sure that you check for negative volumes after you create interfaces in STAR-CCM+.

Comparing a Volume Mesh with the Geometry

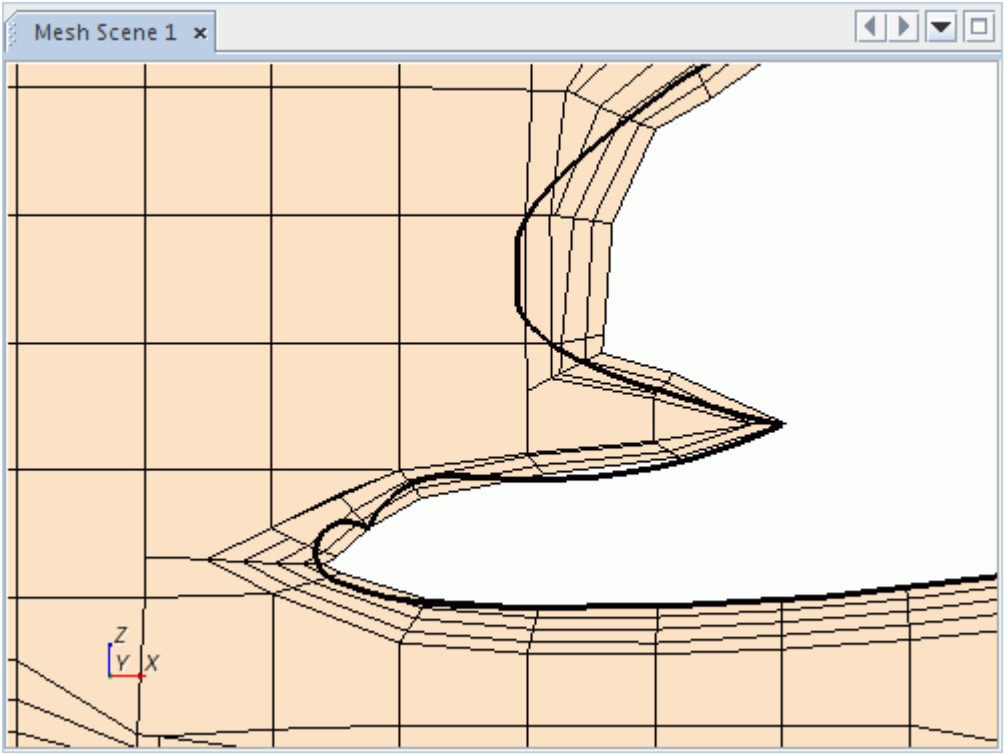
By displaying two different mesh representations, you can compare the volume mesh with the geometry surface to determine the accuracy of the discretization.

In the following example, the original mesh scene shows the volume mesh on a plane section that cuts through the region.

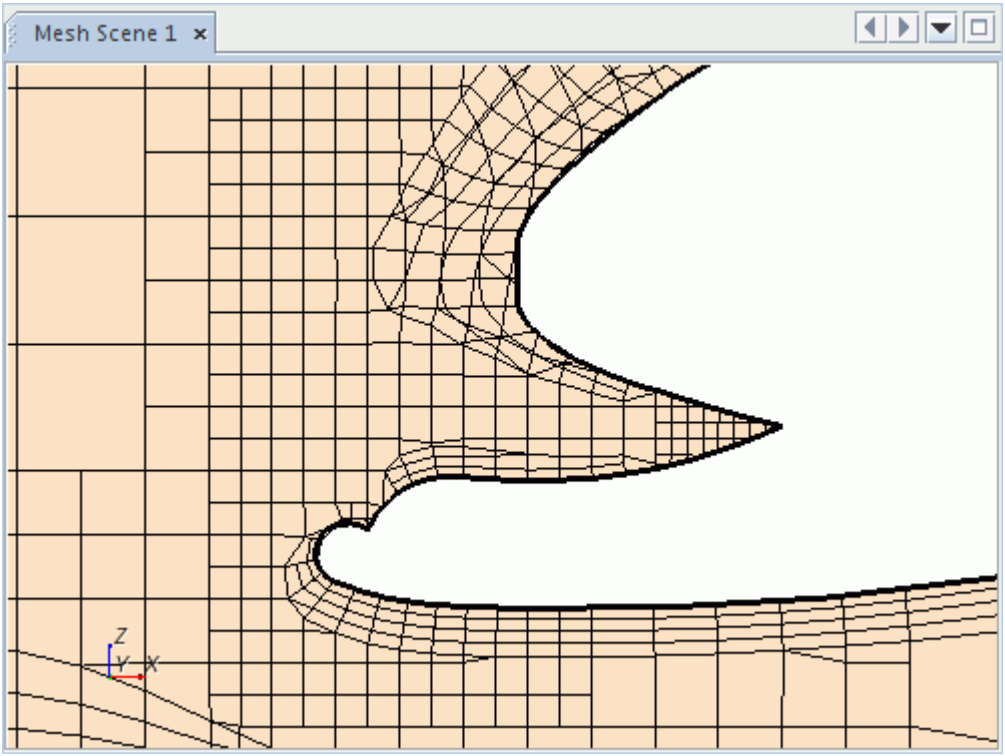


To compare the volume mesh with the geometry:

1. Right-click the **Displayers** node and select **New Displayer** > **Geometry**.
   - Adding a displayer allows you to apply two different mesh representations.
2. Select the geometry displayer and set the properties to control the display.
   - To highlight the difference between the two mesh representations, this example sets the geometry displayer properties to:
     - *Outline*: Activated
     - *Line Width*: 3.0
3. Select the **Parts** node and set the *Parts* property as required.
   - To display the comparison on the same part, this example sets **Parts** to same plane section.
4. Finally, select the geometry displayer and set *Representation* to **Geometry**.
   - Now, the mesh displayer displays the volume mesh and the geometry displayer displays the original geometry surface.



This image shows that the volume mesh is a poor representation of the original geometry. By refining the surface size on the boundary, you can improve the accuracy of the discretization, but this results in a higher cell count.



Using The Connected Cells Field Function

When the volume mesh generation is complete, STAR-CCM+ exposes a field function, **Connected Cells**, that has the same value for any pair of cells that share a face (interior or interface).

To check the connectivity of the volume mesh, use this field function in scalar scenes, threshold parts, reports, and other analysis tools. Initialize the solution when you want to check connectivity across interfaces between regions, or within a region that is divided into separate section by baffles.

This function could be useful when you want to:

- Check for any small non-contiguous cell islands in a single region which can cause the solvers problems. To remove problem cells, use the remove invalid cells option.
- Check that all pieces of a multi-region simulation are thermally connected. Interfaces must be intersected by initializing the solution.

Checking The Skewness Angle

The Cell skewness angle and boundary skewness angle are important measures of cell quality. This section describes the meaning of the terms and how to find the worst (highest) values in the mesh diagnostic report.

If the values for some cells are beyond desired limits, you can identify and visualize the bad cells before returning to the meshing pre-processor to correct the problems.

To view the maximum skewness angles, run the mesh diagnostic report using the full report type option. The cell skewness angle is reported for each mesh region:

Output - Star 1

```
Region Region 1:
    1027 tetrahedral cells
   23317 hexahedral cells
    6285 wedge cells
      31 pyramid cells
    6293 polyhedral cells
   36953 cells total
  106020 interior faces (9688 triangular, 90425 quadrilateral, 5907 polygonal)
   43669 vertices
 Extents:
  x: [7.4997538600e-05, 3.8107493000e-01] m
  y: [2.7863855055e-05, 6.6675762000e-02] m
  z: [1.4901161460e-08, 1.7780002540e-01] m
 Maximum interior cell index delta: 1217, average: 4.1822294850e+02
 Maximum cell face index delta: 3504, average: 1.5822524558e+03
 Volume range: [2.9906282312e-11, 4.7686967264e-06] m^3
 Minimum distance between centroids of neighbor cells = 2.7314738334e-04
 Maximum interior skewness angle = 8.0427368164e+01 deg

Face validity:
Minimum Face Validity: 1.0000000000e+00
Maximum Face Validity: 1.0000000000e+00
        Face Validity <  0.50          0    0.000%
0.50 <= Face Validity <  0.60          0    0.000%
```

The boundary skewness angle is reported for each boundary:

```
Output - Star 1

Boundaries of region Region 1:
Boundary Symmetry Plane: 2397 faces (104 triangular, 1977 quadrilateral, 316 polygonal)
 Extents:
  x: [7.4997538600e-05, 3.8107493000e-01] m
  y: [6.6675762000e-02, 6.6675762000e-02] m
  z: [1.4901161460e-08, 1.7780000000e-01] m
 surface area: 3.2238636166e-02 m^2
 maximum boundary skewness angle = 6.6457313538e+01 deg
Boundary Wall: 7620 faces (1910 triangular, 5147 quadrilateral, 563 polygonal)
 Extents:
  x: [7.4997538600e-05, 3.8107493000e-01] m
```

Scroll through the report and look for any `Maximum interior skewness angle` and `Maximum boundary skewness angle` values of more than 85°. If you find any large skewness angles, identify and visualize the bad cells before taking corrective action.

Contents:

What Is Cell Skewness Angle?
What Is Boundary Skewness Angle

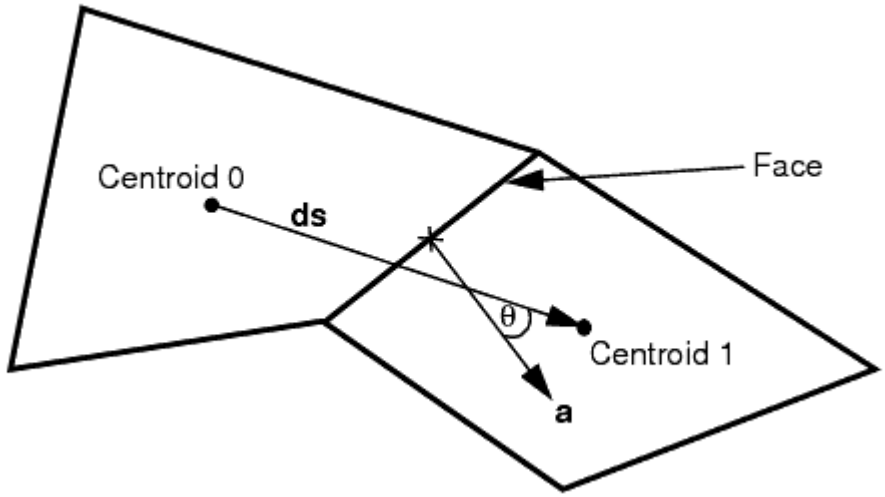
See Also:

Omitting Secondary Gradients

What Is Cell Skewness Angle?

This skewness measure is designed to reflect whether the cells on either side of a face are formed in such a way as to permit diffusion of quantities without these quantities becoming unbounded.

To understand this measure, consider the following two-dimensional diagram of a face and the cell centroids on either side of the face:



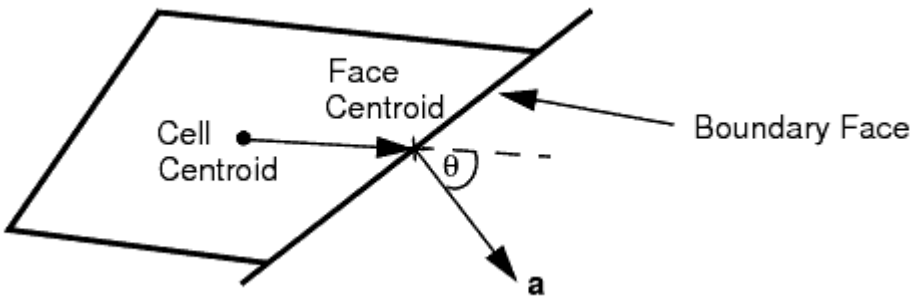
The skewness angle θ is the angle between the face area vector $\mathbf{a}$(face normal) and the vector connecting the two cell centroids, $\mathbf{ds}$. An angle of zero indicates a perfectly orthogonal mesh. Variables that are stored at interior faces in STAR-CCM+ cannot be displayed; instead, the worst skewness angle for all the faces of a cell are stored in that cell. Thus, two cells often have the same skewness angle corresponding to the face that they share.

Skewness angles of 90° or greater, which can occur in concave cells where the centroid of both cells lies on the same side of the boundary face, typically result in solver convergence issues. Problems result because the diffusion term formulation for transported scalar variables contains the dot product $\mathbf{a} \cdot \mathbf{ds}$ in the denominator, and this dot product is zero when the angle between $\mathbf{a}$ and $\mathbf{ds}$ is 90°. Even though STAR-CCM+ takes care to avoid such a divide-by-zero error, the accuracy of the diffusion calculation is reduced and there is a loss of robustness in such cases.

To reduce the impact on robustness, avoid skewness angles greater than 85°, as reported in the mesh diagnostics full report.

What Is Boundary Skewness Angle

The boundary face skewness can be as important as cell skewness. Boundary face skewness is defined as the angle between the area vector and the vector connecting the cell centroid and the boundary face centroid.



This angle is important since the dot product of these two vectors appears in a denominator in a component of the equation for diffusive fluxes at a boundary face.

In turbulent flows where wall functions are used, the equation for diffusive fluxes at a boundary is not used to compute diffusion fluxes at walls, so boundary face skewness become less important. However, it is important for laminar flows and for heat transfer in solids.

To reduce the impact on robustness, avoid boundary skewness angles greater than 85 degrees, as reported in the mesh diagnostics full report.

Identifying And Visualizing Bad Cells

To identify and visualize cells in STAR-CCM+ according to a given criteria, such as one of the quality metrics, use threshold values.

Two main types of threshold are available to achieve this goal:

- Derived part threshold
- Cell set threshold

Two techniques are available when using derived part thresholds to identify poor quality cells in the mesh:

- Identification and visualization using cell metrics. This method can be used for finding cells with negative volumes, high cell skewness angles, and low values of face validity, cell quality, and volume change.
- Identification and visualization using boundary metrics. This method is useful for finding cells with zero area faces and high boundary skewness angles.

The cell-set threshold options allow multiple threshold criteria to be defined for identifying cells, as well as providing tools for manipulating the cell collections.

Other than a tool to remove invalid cells, STAR-CCM+ does not currently have any volume mesh fixing tools to correct problems found using the threshold checks. However, it is useful to extract information such as the cell index ID or the location of the cell centroid to help identify the cells in the original pre-processing package. An example way of creating a XY plot showing bad cells that have been identified using a threshold is presented.

Contents:

Identification And Visualization Based On Cell Metrics

Identification And Visualization Based On Boundary Metrics
Creating Threshold XY Plots

Identification And Visualization Based On Cell Metrics

Using cell-based threshold metrics, the bad cells in the interior of the mesh can be identified and isolated for further investigation.

This technique is applicable for finding cells that are based on:

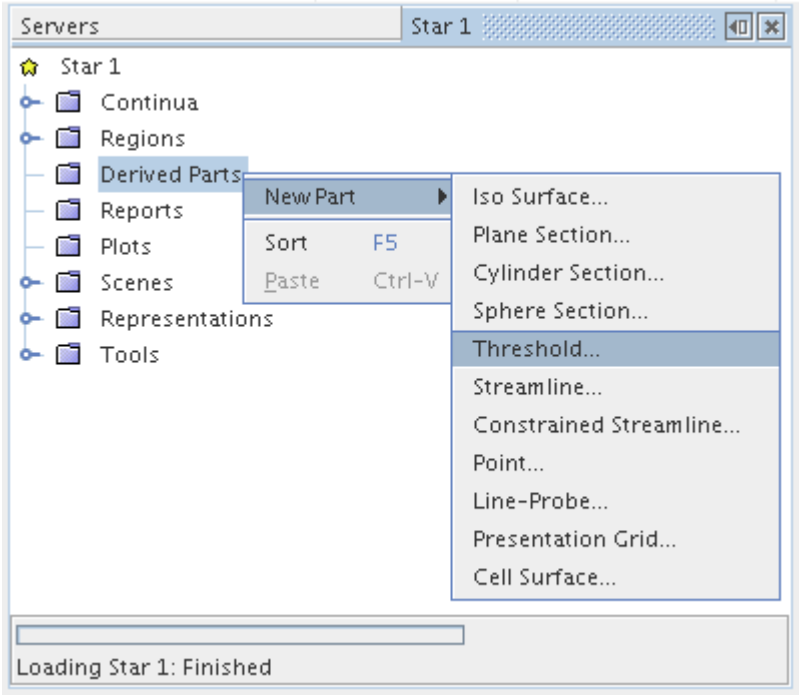
- negative volumes;
- cell skewness;
- face validity metric;
- cell quality metric; and
- volume change metric

The method that is used to identify and visualize cells is identical for all the above scalars and is interchangeable once a threshold has been created. That is, the same threshold can be used to investigate any or all of the scalars one at a time by changing the properties of the node.

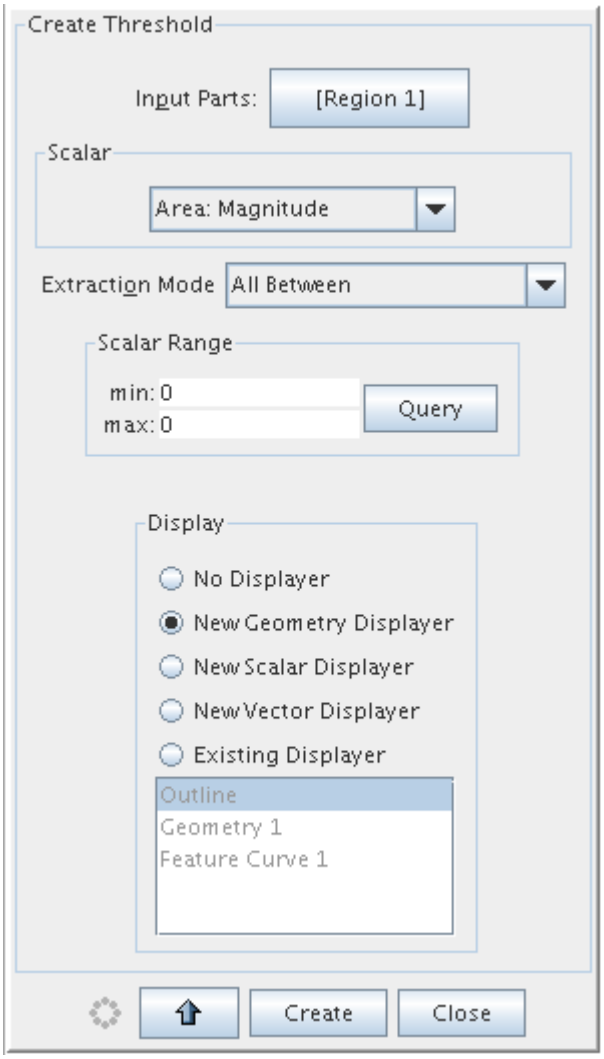
To access the above cell metrics as part of a threshold, first initialize the solution with the appropriate physical model mesh metrics activated. In this case, to access the cell metrics, select the **Three Dimensional** *Space* model before doing the initialization. If you do not perform this step, only the cell quality metric is then available by default.

A geometry scene must exist or one must be created in order to display the model. Simplifying the model to show only the outline of the geometry can assist the visualization of threshold cells. To display only the outline, edit the **Geometry 1** node for the scene so that it contains no parts, or hide all the parts. Alternatively, you can create a new displayer.

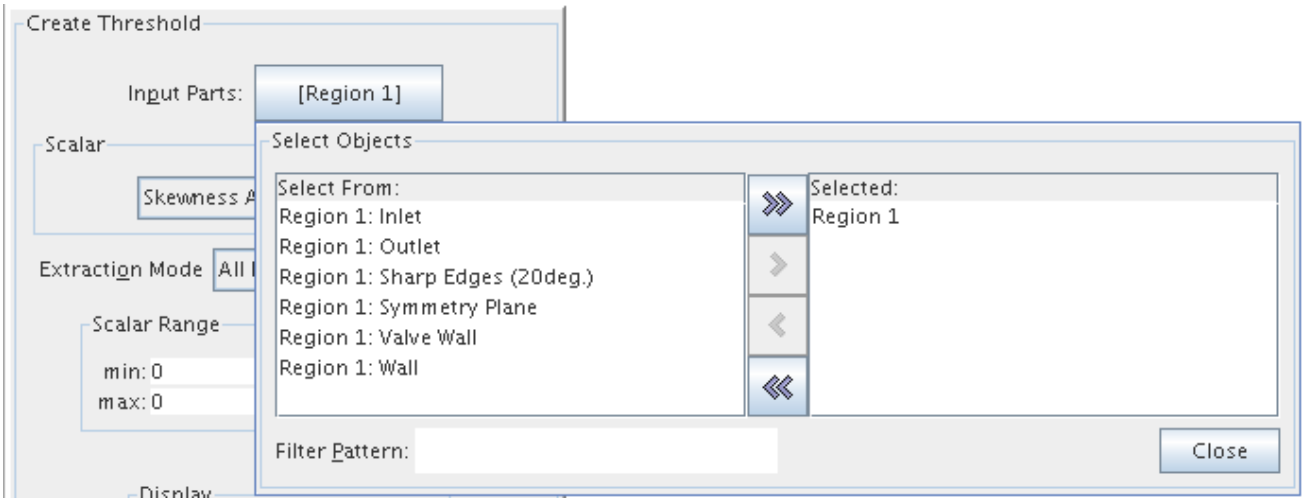
To create a threshold, right-click on the **Derived Parts** node and select **New Part** > **Threshold...** from the context menu.



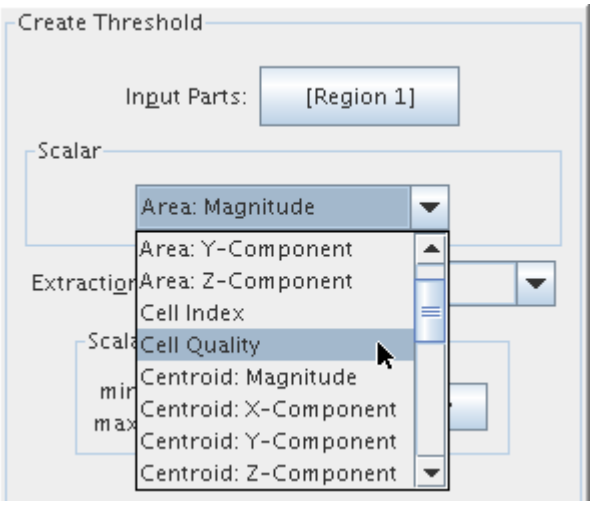
The *Create Threshold* dialog appears, which allows the selection of the input parts, threshold scalar, extraction mode, value range, and display options.



To identify interior cells (as opposed to boundaries), the **Input Parts** list must contain the mesh region or regions nodes of interest. To edit the list, click the **Input Parts** list box to open the *Select Objects* parts editor:



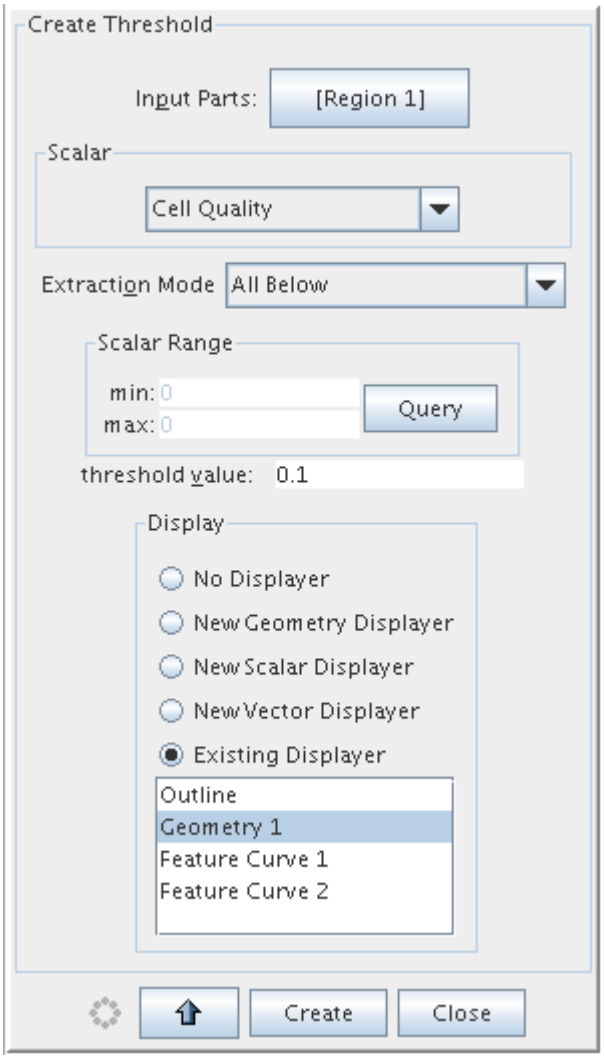
Once you have selected the desired region or regions, click the **Close** button. The threshold scalar field function can be selected next. Click the *Scalar* pull-down menu, and scroll through the list to select the one of interest.



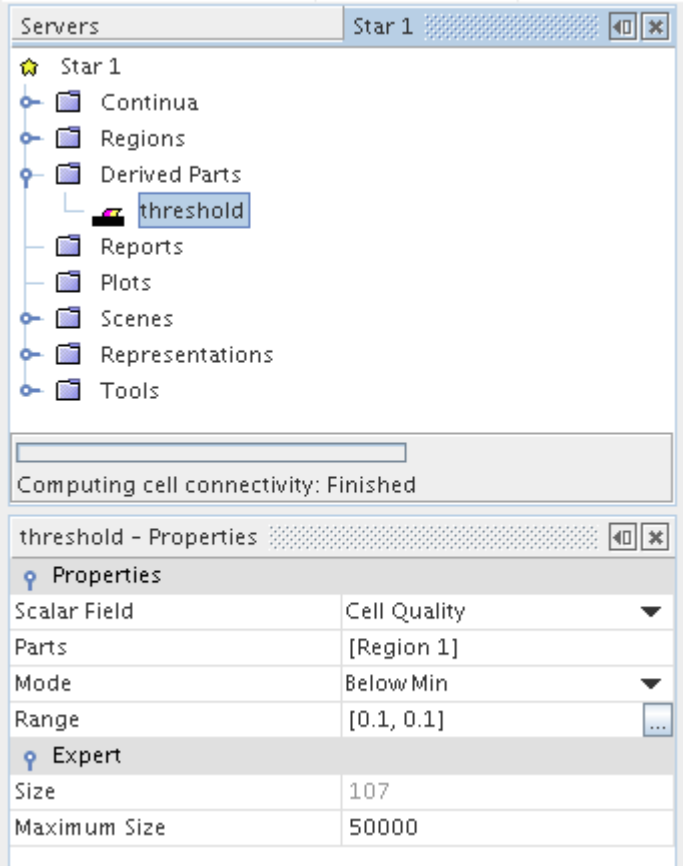
To check for one of the cell metrics that were listed earlier, choose from the scalars *Cell Quality*, *Volume*, *Skewness Angle*, *Face Validity*, or *Volume Change*.

To get an idea of the scalar value range, the *Scalar Range* **Query** button can be pressed. The *Extraction Mode* option can then be used to select how you want to identify the cells for the scalar threshold. Typically, you would want to select cells either above or below a given value but you can also select cells within a range or outside (below min and above max) a range. Use the **All Below** option here and a **threshold value** of `0.1`, meaning that all cells with a cell quality below 0.1 are identified.

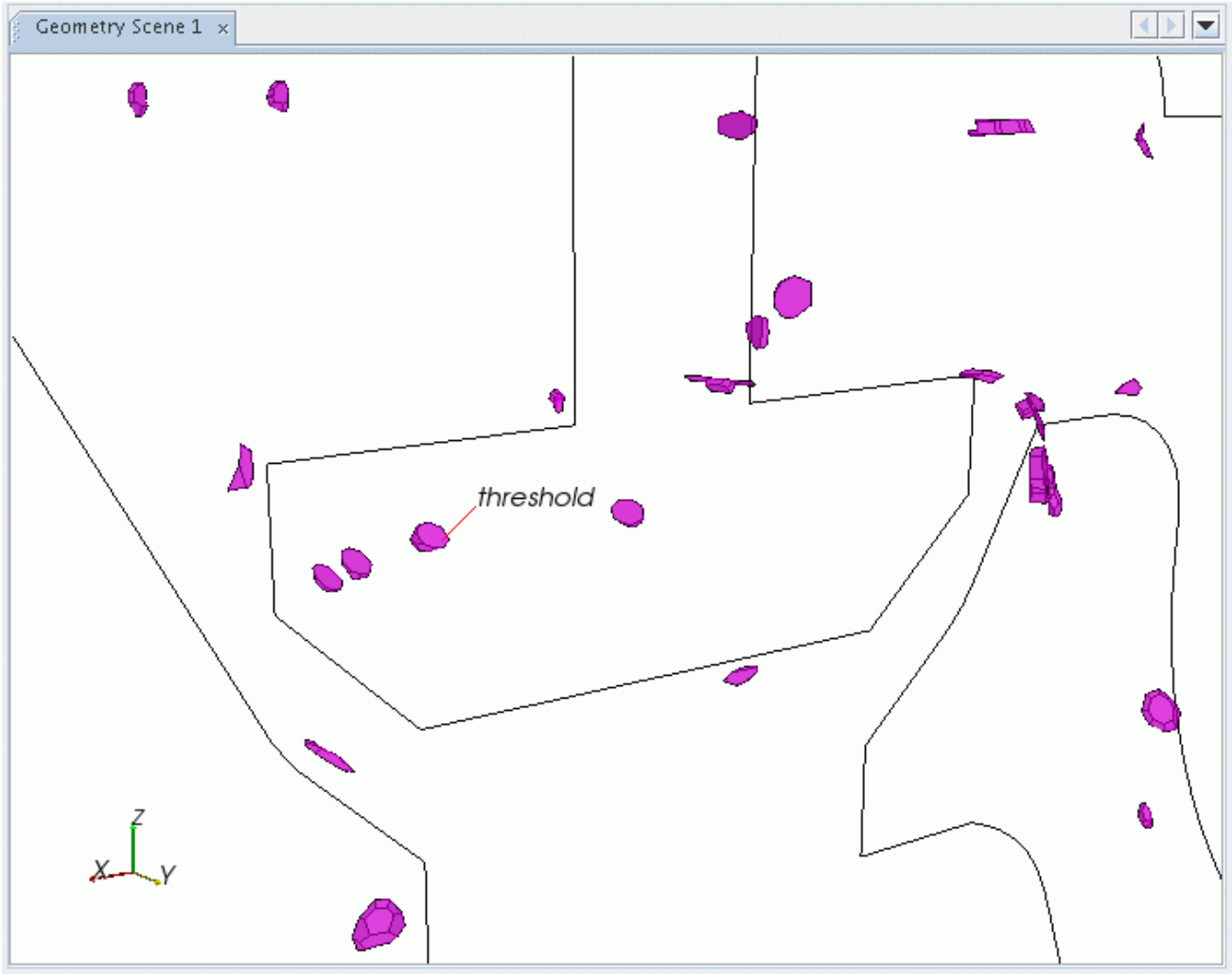
The last item to decide is how the new threshold node is displayed. You can select not to create a displayer, to create a new geometry, scalar, or vector displayer, or add to an existing displayer. When selecting the last option, you can decide which pre-existing displayer the new node is added to. Use this option here and add the threshold part node to the **Geometry 1** displayer.



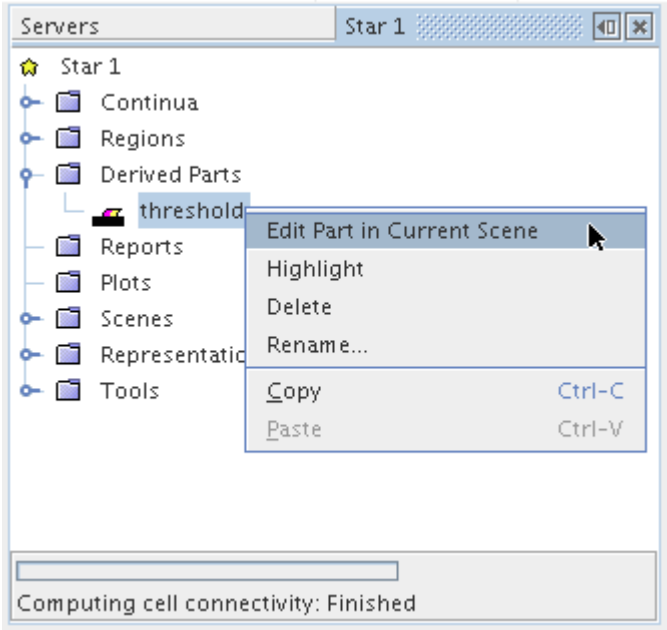
Click the **Create** button to create the threshold part. This may take several seconds depending on the number of regions and type of scalar selected. Press the **Close** button on the dialog once the process is complete. A new threshold node is created in the **Derived Parts** node. To view the new threshold node properties, click the node:



In the above case, the current geometry scene is updated with the results of the threshold check. With the **threshold** node selected, the cells are shown highlighted in the display:



You can rotate, translate, and zoom the display in order to investigate the poor quality cells. Thereafter, the *Properties* of the threshold node can be changed directly in order to update the display. Alternatively, you can right-click on the **threshold** node and select **Edit Part in Current Scene** in order to return to the original dialog used to create the threshold.



Identification And Visualization Based On Boundary Metrics

Boundary metrics can be used to identify bad cells on the boundaries.

This technique is applicable for finding boundary cells which have:

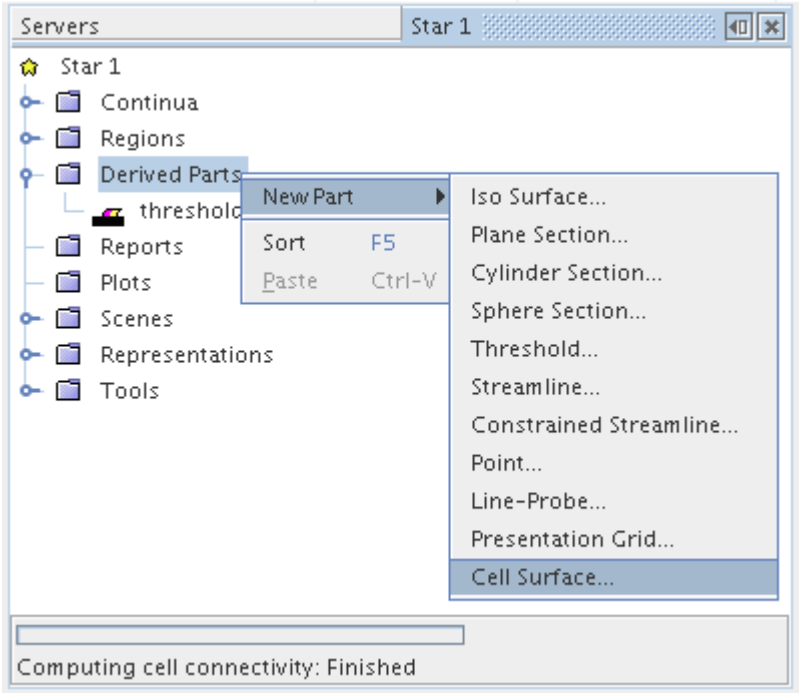
- zero area faces; and
- poor boundary skewness angles;

To identify the boundary surfaces using the boundary metrics that are listed above, follow the same procedure that was used for identifying and visualizing cells using cell metrics. However, for the **Input Parts** list in the *Create Threshold* dialog, select the desired boundaries instead of regions to operate on in the *Select Objects* editor. Additionally, to check for zero-area faces, use the Area: Magnitude option for *Scalar*.

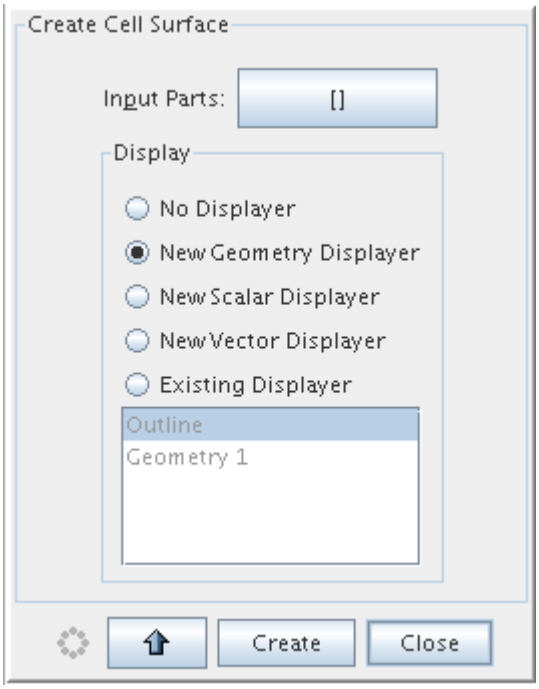
This results in the threshold boundary surfaces being displayed at the end. However, this type of display is limited for the following reasons:

- Displaying the value of the scalar quantity on the boundary does not indicate the shape of the cell.
- If the quantity of interest is the face area magnitude and the face areas are almost zero, then the polygon representing the face is typically hard to locate and view.

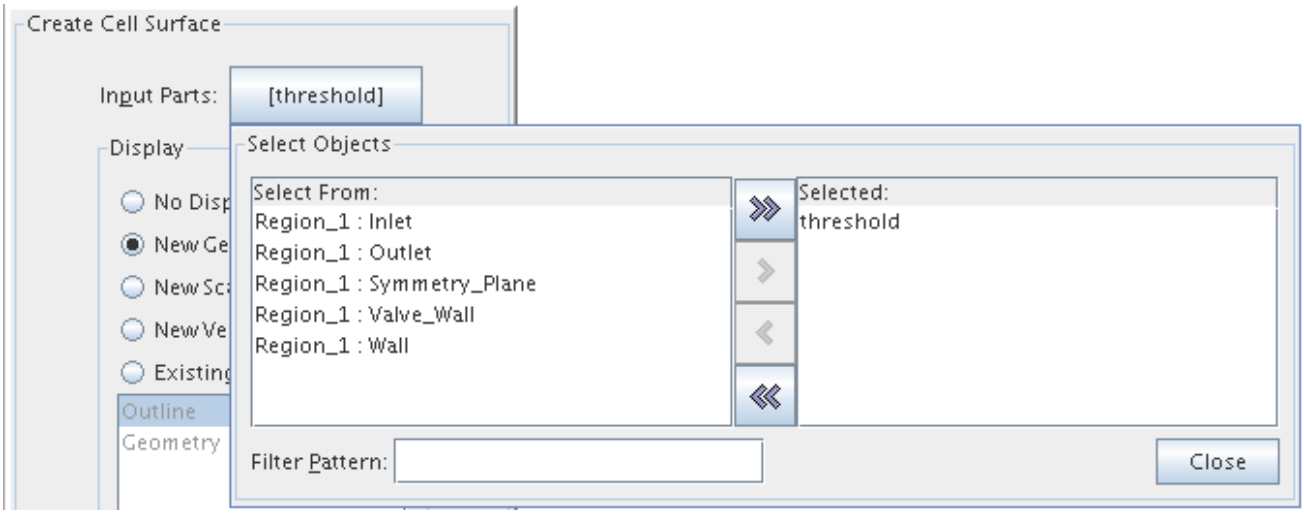
To visualize the cells next to the threshold surfaces, create a new cell surface derived part using the threshold by right-clicking on the **Derived Parts** node. Select **New Part** > **Cell Surface...** from the context menu.



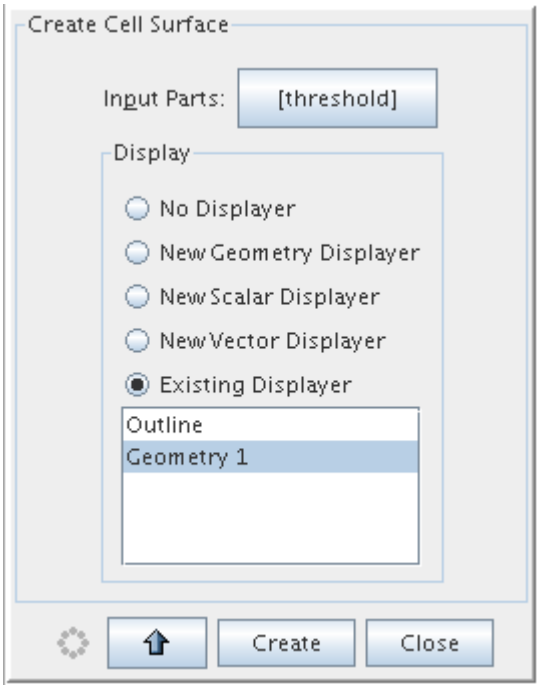
The *Create Cell Surface* dialog appears, which allows the selection of the input parts and display options for the new part.



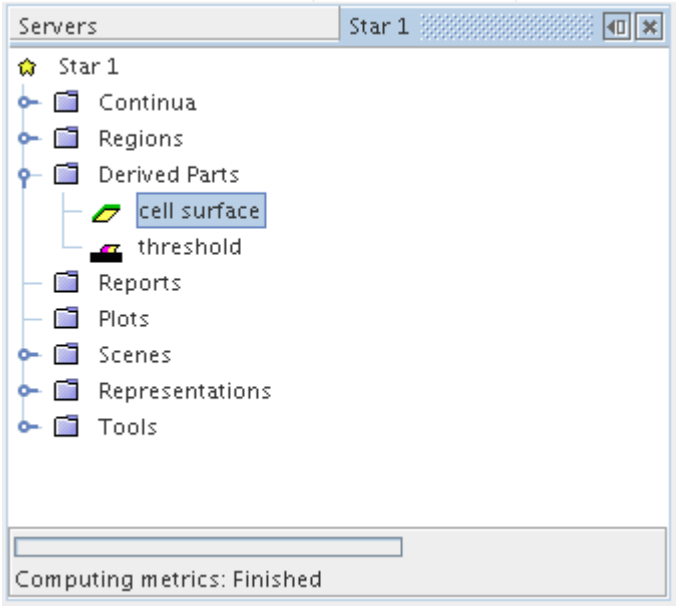
The **Input Parts** list in this case contain the threshold parts previously created. To select the nodes of interest, click the **Input Parts** list box and use the *Select Objects* parts editor:



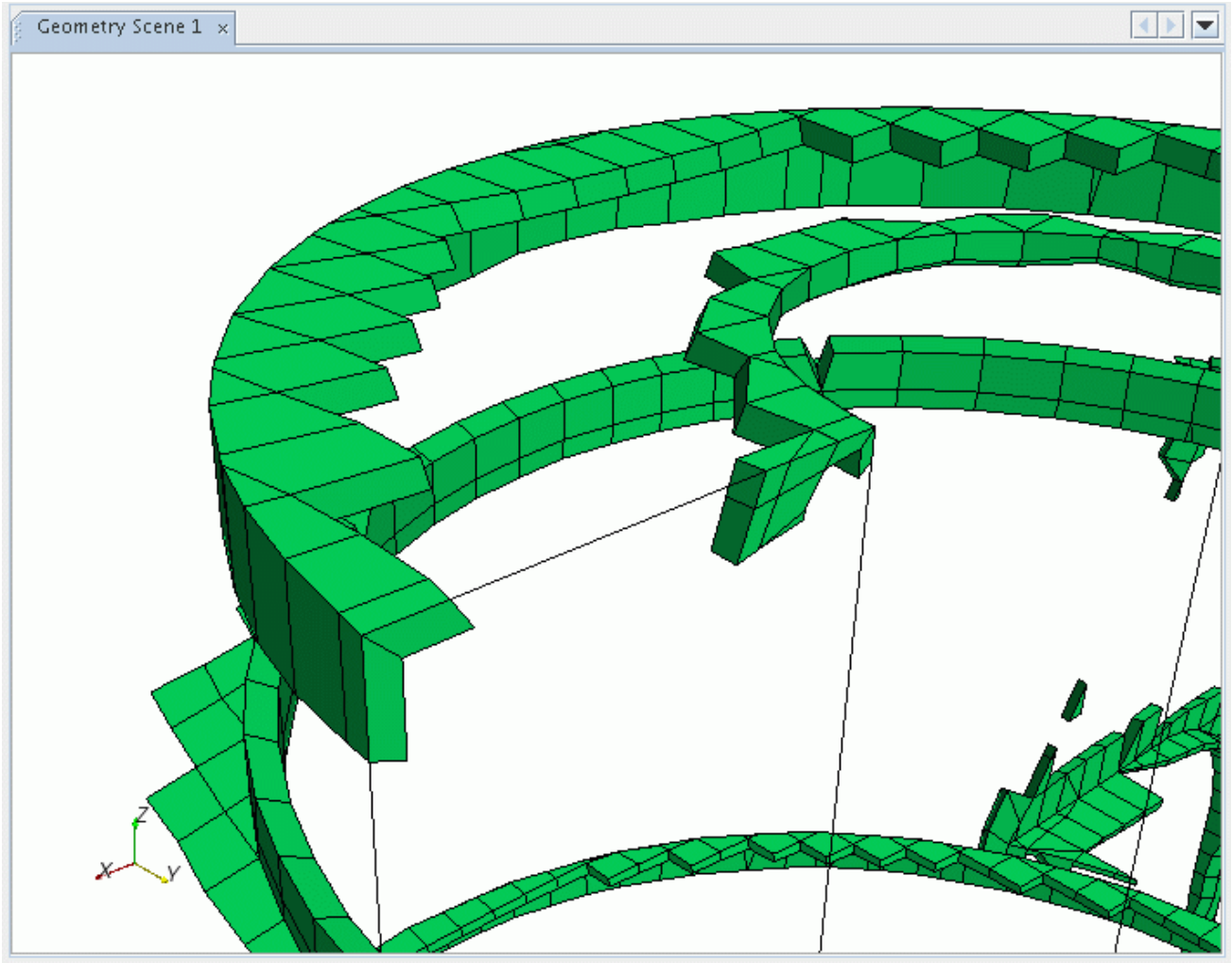
Press the **Close** button on the editor when done. To display the new part, select an appropriate displayer. The easiest method is to add the new part to an existing geometry displayer.



To create the new part, click the **Create** button, and then **Close** to close the dialog. A new **Derived Parts** node that is called cell surface appears in the simulation tree:



The geometry scene is also updated with the results of the cell surface creation which uses green colored cells by default.



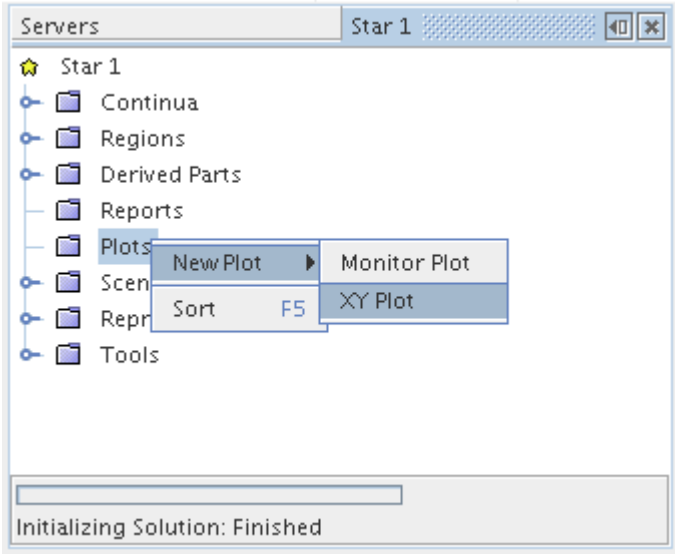
The full shape of the boundary cells can now be seen and inspected further by using the normal zoom, translate, and rotate options.

Creating Threshold XY Plots

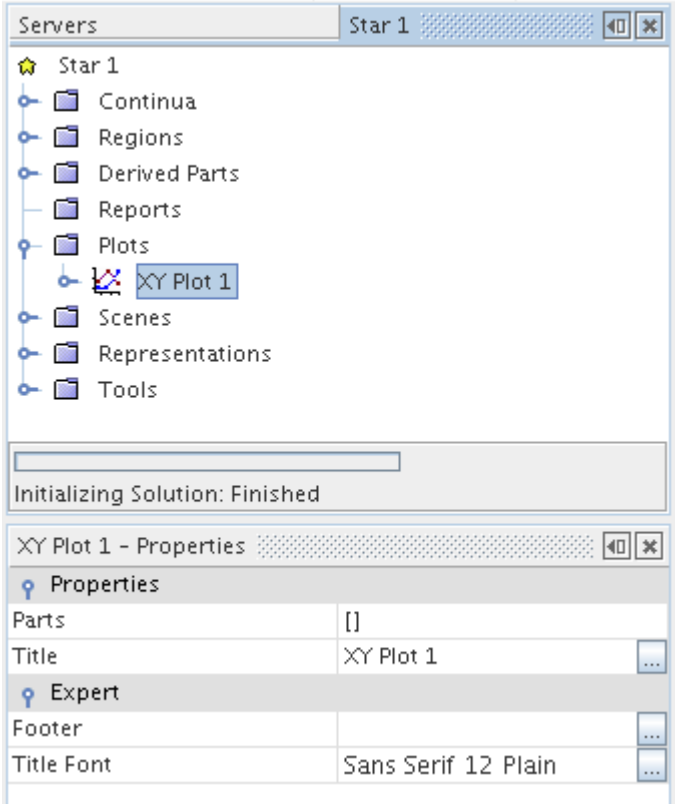
A XY scatter plot of the cell or boundary thresholds against the pro-STAR cell index can be made in order to visualize the distribution of cell quality in the mesh.

An appropriate cell metric or boundary metric based threshold must be made first before you can analyze the bad cells in the mesh.

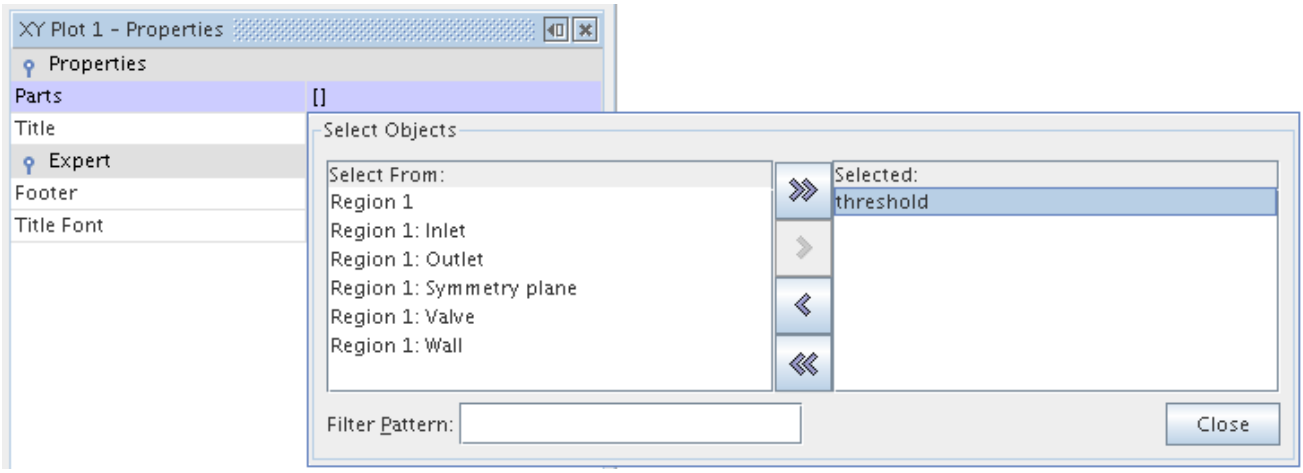
To create an XY plot, first right-click on the **Plots** node and select `New Plot > XY Plot`.



A new **XY Plot 1** node is created with the properties shown below:

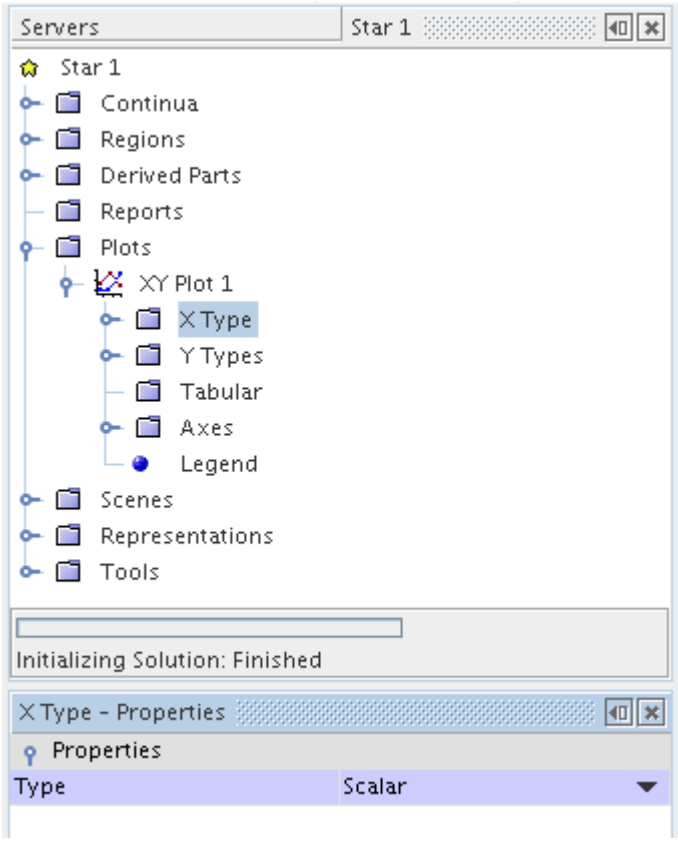


Edit the *Parts* list in order to add the cell or boundary-based threshold. To open the *Select Objects* in-place dialog, click the *[] Parts* property, and select the threshold part that was created earlier. Add it to the Selected list by pressing the > button.

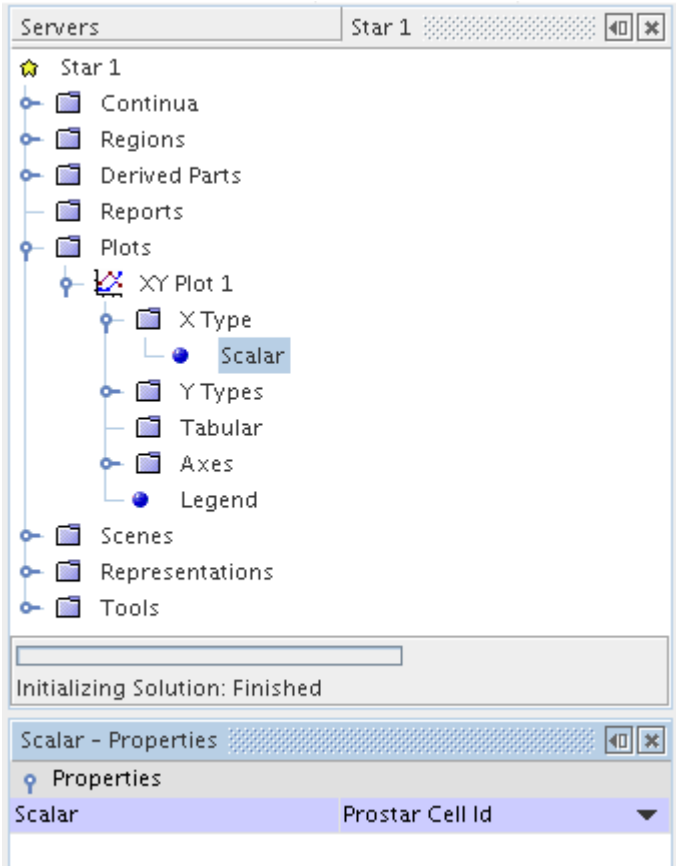


To close the editor, press the **Close** button. The cells in the threshold part are used as the basis to extract data for the XY plot, which means that only data for the bad cells are shown. Next, set the properties for the X and Y axes, both of which use scalars.

Open the **XY Plot 1** node and click the **X Type** node in order to view its properties. Set the *Type* property to **Scalar** by using the pull-down menu, as shown below.

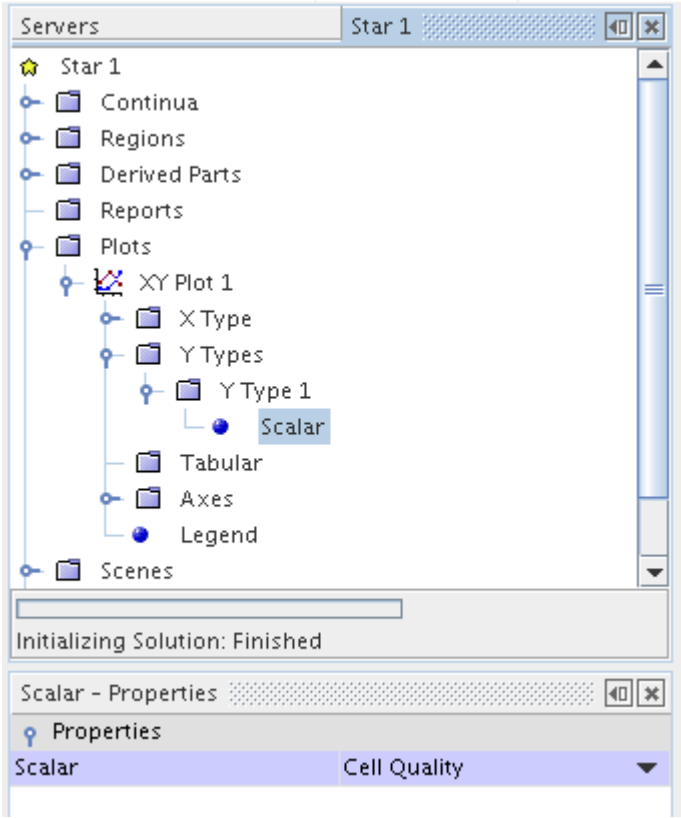


Setting the *Type* to be a scalar, allows the use of pro-STAR cell index numbers for the X-axis. Open the **X Type** node and click the **Scalar** node underneath. In the *Properties* window for the node, set *Scalar* to be **Prostar Cell Id** by using the pull-down menu.

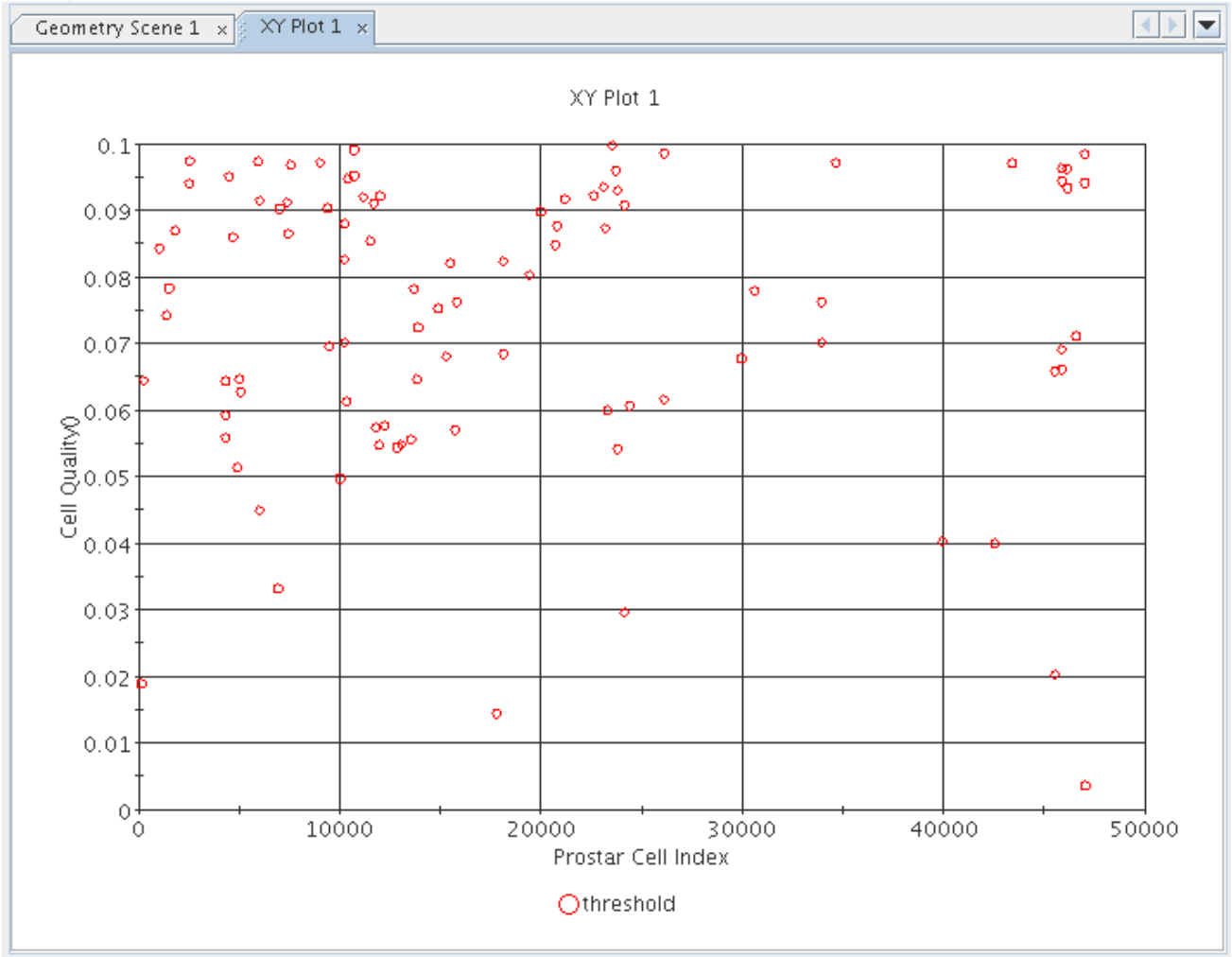


The X-axis of the plot then uses the pro-STAR cell index values from the threshold part, which are typically anywhere in the range of 1 to the highest cell number in the mesh.

The Y-axis scalar can now be set. For this example, use the same scalar as used for the cell threshold that you previously defined, namely cell quality. Open the **Y Types** node and click the **Y Type 1** > **Scalar**node. In the Properties window, select **Cell Quality** for the *Scalar* option.



The XY plot now has sufficient information to update the scene with the data points.



The original derived part threshold properties can be used to control the number of cells and range that are used in the graph. The plot is useful to determine clustering or banding of data points for different cell and boundary metrics for the mesh.

Removing Invalid Cells

Poor quality cells can cause the solution to diverge or compromise the accuracy of the results. STAR-CCM+ provides a tool for removing poor quality mesh cells based on specified mesh criteria.

In STAR-CCM+, the **Removing Invalid Cells** tool allows you to remove mesh cells from the volume mesh region definition based on one or more of the following mesh criteria:

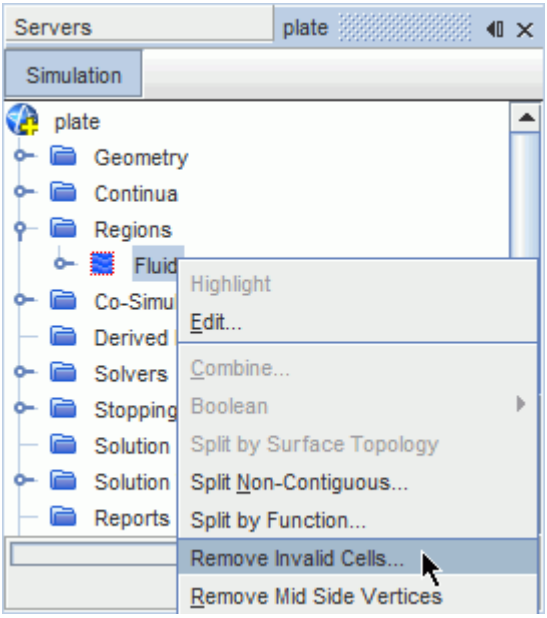
- Face Validity
- Cell Quality
- Volume Change
- Contiguous Cells
- Connected Face Area
- Volume

- Function

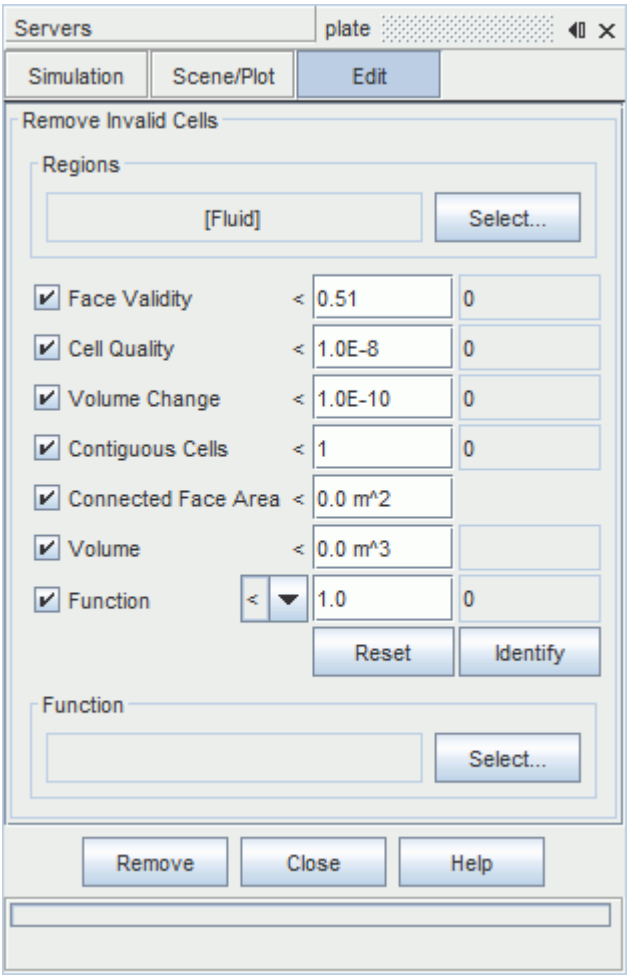
STAR-CCM+ moves all the removed cells into a separate region that is not used in the analysis. Symmetry plane boundaries are automatically added to the neighboring cell faces of cells that have been removed, minimizing the impact of these removed areas on the solution.

To remove invalid cells:

1. Open a mesh scene in the *Graphics* window.
2. Right-click the region or regions of interest and select **Remove Invalid Cells...**



The *Remove Invalid Cells* dialog appears.



3. In the *Regions* box, click **Select...** and use the *Select Objects* dialog to specify the relevant regions.
4. Enable the mesh criteria that you want to use to identify invalid cells.
5. For the enabled criteria, specify the desired threshold values.

   The following threshold values are suggested guidelines:

   - **Face Validity:** 0.95 to 0.51 (default 0.51)
   - **Cell Quality:** 1e-05 to 1e-12 (default 1e-08)

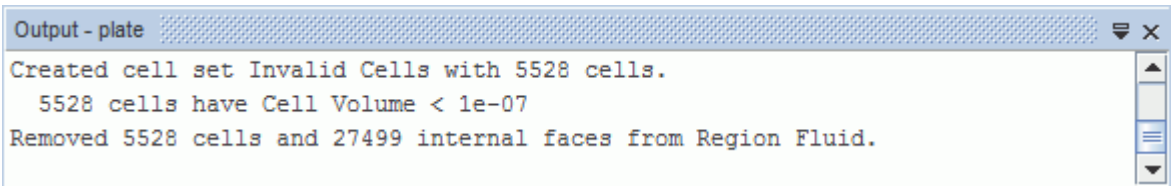
     As the **Cell Quality** metric can sometimes flag valid trimmed cells (especially if they are anisotropic), you are recommended to use a low value.
   - **Volume Change:** 1e-05 to 1e-12 (default 1e-10)
   - **Contiguous Cells:** 1 to 1000000 (default 1)
   - The **Connected Face Area** is dependent on the case dimensions and can be estimated based on the average mesh cell size.

   If you change the metric values then they are automatically preserved as part of the STAR-CCM+ workspace and will be used instead of the default values in future sessions. To restore the default factory values, click **Reset**.
6. To preview the number of cells that each criterion would flag for removal, click **Identify**.

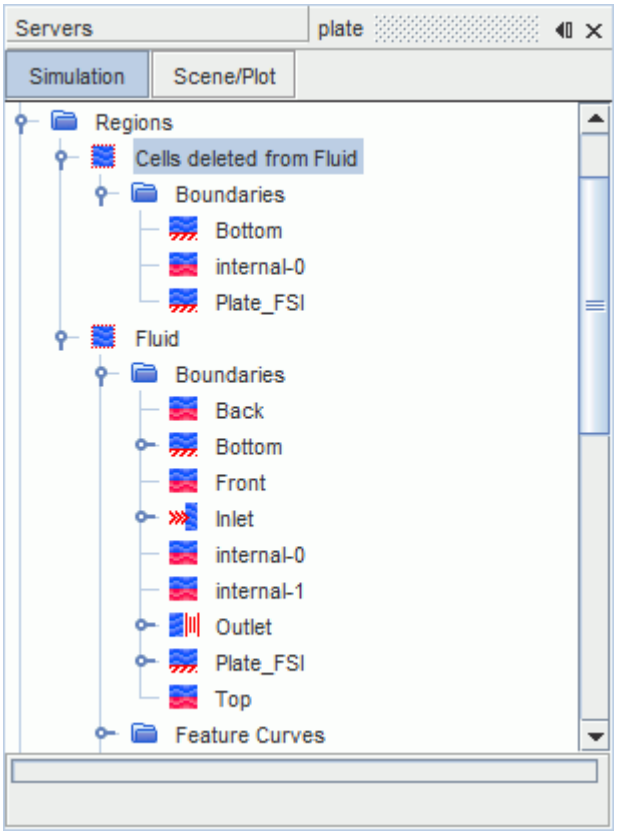
   The cells that meet the specified criteria are displayed in the active mesh scene.

7. When you are satisfied with the criteria, click **Remove** to remove the cells. If you do not want to remove cells, just close the dialog without selecting **Remove**.

   The *Output* window displays the result:



Cells that meet the provided criteria are moved into a new region called **Cells deleted from [Region]**.



New symmetry plane boundaries with the names **internal-0** and **internal-1** are created in each region accordingly. The new region has no associated physics continuum, so it is not necessary to delete it in order to run the analysis.

Contiguous Cells

The contiguous cells metric can be used to identify discontiguous regions of cells (as determined by the connected face area metric). These cell regions can either pre-exist in the volume mesh or be created when bad quality cells are removed based on the other three invalid cell criteria. Discontiguous fluid cells would cause the solver to fail under most circumstances.

A value in the range 1 - 1000000 can be entered, where the value indicates the number of cells in the discontiguous region. On utilizing the feature, first the other criteria are applied and any invalid cells are flagged. Then any discontiguous regions of unflagged cells are checked and compared to the supplied metric value, as well as the supplied connected face area value. If the number of cells is less than the amount specified, and/or the area criterion is not met, then the region is also marked for removal.

Connected Face Area

The connected face area is the smallest area that is allowed between two connected cells, below which the cells would be considered discontiguous and subject to the contiguous cells check criterion.

It is sometimes possible during meshing and cell removal that is based on the other criteria, that two cells are left connected but with an extremely small face area between the two. Such scenarios can lead to the flux calculation becoming zero and the solution becoming unstable. Although topologically connected, if the face area between the two cells is below the specified value, then the cells are no longer considered to be connected. In this case, remove the smaller group of cells using the above contiguous cell check.

Volume

Use the volume metric to remove all the cells with volume below a specified value. You can specify any value above zero.

Function

The function metric allows you to remove the cells for which a selected field function has a value below

( ) or above ( ) a specified limit. To specify the field function, click **Select...** in the *Function* group box and use the *Select Objects* dialog.

Meshing Field Functions Reference

This section contains a list of field functions that are related to meshing.

Surface Mesh Field Functions

Several field functions are made available in STAR-CCM+ as a result of importing parts or using the surface remesher.

Area	Displays the area of the cell faces. Only available for surface meshes.	
	Magnitude	Displays the absolute value of area.
	Laboratory	Displays the value of area normal to the i, j, or k directions.
Centroid	See Space Field Functions Reference.	
DeviationDistance.AutomatedMesh	Compares the latest surface that is displayed in the scalar scene to an earlier description that is created from the automated mesh operation. Displays how far the surface in the scene deviates from the earlier description.	
DeviationDistance.Root	Compares the latest surface that is displayed in the scalar scene to the root description and displays how far the surface in the scene deviates from the earlier root description.	
EdgeLength	Displays the average edge length on faces. Only available for surface meshes.	
FaceQuality	Displays the face quality value on faces, where 1.0 is the highest quality and 0.0 is the lowest quality. Only available for surface meshes.	
FreeEdges	Displays any free edges on the surface. Only available for surface meshes.	
NonmanifoldEdges	Displays any non-manifold edges on the surface. Only available for surface meshes.	
Position	See Space Field Functions Reference.	
SurfacePatch	Displays each surface patch as a unique number. Only available for surface meshes.	

Volume Mesh Field Functions

Several field functions are made available in STAR-CCM+ as a result of mesh model selections or meshing operations.

Area	The **Area** field function displays the area of the cell faces. Area can be shown as either a Magnitude or as Laboratory coordinate values. From STAR-CCM+ version 10.02, you can also run the Area field function on an unmeshed surface.

	Magnitude	Displays the absolute value of area.
	Laboratory	Displays the value of area normal to the i, j, or k directions.
Boundary Index	This field function displays the index number that is associated with the boundaries. Each boundary has a unique boundary index that distinguishes it.	
Boundary Sliver Cell Indicator	This field function indicates cells that are “boundary slivers”. Boundary slivers are flat cells on the outer boundaries of the solution domain that can be removed without affecting the topology of the mesh. A value of 0 indicates that the cell is not a boundary sliver cell, and a value of 1 indicates that the cell is a boundary sliver cell. (Available when the Cell Quality Remediation model is used). For more information, see the Remedying Cell Quality Field Functions Reference page.	
Cell Aspect Ratio	This field function is used to identify regular polygonal cells from skewed or stretched cells. A value of 1 indicates that the cell is a regular polygon, and a value close to 0 would represent a highly stretched cell. For more information, see the Modeling Space Field Functions Reference page.	
Cell Index	This field function displays the index for each cell. The index is only unique within the portion of the mesh for a given region on the local partition. So if you have R regions that are partitioned across P processors there are R*P cells with index 0 (assuming enough cells to allocate at least one per region per partition). It is not an ID, it is an offset.	
Cell Quality	Cell quality is a function that is determined from the relative geometric distribution of the cell centroids of the face neighbor cells and of the orientation of the cell faces. Generally, flat cells with highly non-orthogonal faces have a low cell quality. A cell with a quality of 1.0 is considered perfect. A degenerate cell has a cell quality approaching 0. Depending on the physics that has been selected, the cell quality of a cell can be fairly low and still provide a valid solution. However, poor cell quality is likely to affect both the robustness and accuracy of the solution. Cell quality is only defined for three-dimensional meshes. See Mesh Quality: Cell Quality.	
Centroid	The centroid is the location vector of the center of mass of the volume (or shell). The center of mass can be displayed as a Magnitude or by Laboratory coordinates. For more information, see the Modeling Space Field Functions Reference page.	
	Magnitude	Displays the absolute location of the center of mass.

	Laboratory	Displays the location of the center of mass in either the X, Y, or Z direction.
Chevron Quality Indicator	This field function identifies chevron cells. Chevron cells are pairs of thin slender cells which meet at a common face at an angle such that the line joining the cell centers does not pass through the common face. Chevron cells are marked as 1.0 and every other cell as 0. See Mesh Quality: Chevron Quality Indicator.	
Element Type	The element type identifies the specific star cell topology for each cell. Each number corresponds to a different type of topology, as shown below:	
	One Dimensional	
	3	STAR_LINE_CELL
	21	STAR_QUADRATIC_EDGE
	Two Dimensional	
	5	STAR_TRI_CELL
	7	STAR_POLYGON_CELL
	9	STAR_QUAD_CELL
	22	STAR_QUADRATIC_TRIANGLE
	23	STAR_QUADRATIC_QUAD
	Three Dimensional	
	10	STAR_TET_CELL
	12	STAR_HEX_CELL
	13	STAR_WEDGE_CELL
	14	STAR_PYRAMID_CELL
	24	STAR_QUADRATIC_TETRA
	25	STAR_QUADRATIC_HEXAHEDRON
	26	STAR_QUADRATIC_WEDGE
	29	STAR_TRIQUADRATIC_HEXAHEDRON
	41	STAR_UNIDENTIFIED_CELL
	42	STAR_POLYHEDRON
Face Index	Face index displays the partition-local index for each boundary face.	

Face Type	Face type displays the number of vertices per face. For example, a triangular face has a Face Type of 3 and a hexagonal face has a Face Type of 6.	
Face Validity	Face validity is an area-weighted measure of the correctness of the face normals relative to their attached cell centroid. In a good quality cell, the face normals point away from the cell centroid. In a cell with bad face validity, one or more face normals point towards the attached cell centroid. A face validity of 1.0 means that all face normals are correctly pointing away from the centroid. Values below 1.0 mean that some cell face normals point towards the centroid, indicating some form of concavity. Values of below 0.5 signify a negative volume cell. See Mesh Quality: Face Validity.	
Interface Parent Face Index	This field function is the offset of the parent face for each direct boundary interface boundary face. Each face on a direct boundary interface boundary is a piece of an intersection of one face from each of the contributing boundaries. This index identifies those faces from the contributing boundaries.	
Least Squares Quality	Least squares quality is an indicator of the quality of a cell that is based on the physical location of a cell centroid relative to the cell centroid locations of its face-neighbors. (Available when the Cell Quality Remediation model is used). See Mesh Quality: Least Squares Quality.	
Partition	The primary partition that the cell/face/vertex is on. When a simulation is run in parallel, the regions are decomposed across the number of processors such that each cell is allocated to a primary partition.	
Periodic Index	Repeating interfaces can require multiple passes to intersect all faces. For example, a rotor-stator (sliding interface) case intersects once in situ, rotates/translates one side by the periodic transform and intersects again, and repeats until all faces are consumed. For a given intersected face, this value reports the number of steps that were required until it was matched. So it indicates how many multiples of the periodic transform must be applied to align it with the other side of the interface.	
Position	This field function displays the position of the vertices relative to the origin. Position can be displayed as a Magnitude or as Laboratory coordinates. For a cell, the position is returned as the average of the positions of the vertices of the cell. For more information, see the Modeling Space Field Functions Reference page.	
	Magnitude	Displays the absolute position.
	Laboratory	Displays the position in either the X, Y, or Z direction.
Prostar Face Id	When a face on an imported STAR-CD mesh has been assigned a boundary, this field function returns the index of the original pro-STAR	

	boundary. In pro-STAR, this number would be plotted in response to the `NUMB,BOUND,ON` command.
Prostar Cell Id	This field function contains the original pro-STAR cell number for the imported STAR-CD mesh. In pro-STAR, this number would be plotted in response to the `NUMB,CELL,ON` command.
Prostar Cell Type	This field function contains the pro-STAR cell table index number for the imported STAR-CD mesh. In pro-STAR, this number would be plotted in response to the `NUMB,TYPE,ON` command.
Radial Coordinate	If an axis is defined, this field function provides the distance from the axis.
Region Index	This field function contains the region index. Each region has a unique region index that distinguishes it from other regions.
Skewness Angle	This field function contains the skewness angle of the cells. It indicates whether the cells on either side of a face are formed in such a way as to permit diffusion of quantities without these quantities becoming unbounded. To reduce the impact on robustness, avoid skewness angles greater than 85 degrees.
Unmatched Edge Count	This field function displays any unclosed cells. A topologically closed volume cell consists of edges that are connected to exactly two faces, and faces with normals pointing outwards. If a cell is missing a face or the outward normals are not consistent, then it is considered unclosed and invalidates the mesh.
Volume	This field function contains the volume of each cell.
Volume Change	This field function describes the ratio of the volume of a cell to that of its largest neighbor. A value of 1.0 indicates that the cell has a volume equal to or higher than its neighbors. A large jump in volume from one cell to another can cause potential inaccuracies and instability in the solvers. Investigate cells with a volume change value of 1e-05 or below as they can be considered suspect. See Mesh Quality: Volume Change.
Zero Face Area Indicator	This field function has the value of unity if the area of any of the faces approaches zero; otherwise it has a value of zero. It is used to diagnose potential mesh problems. (Available when the Cell Quality Remediation model is used). For more information, see the Remedying Cell Quality Field Functions Reference page.

Working With Representations

This section provides information on representations as a useful part of the meshing process.

This section describes what representations are, how they are used as part of the meshing process and how they can be visualized. Replacing, exporting and extracting a surface mesh are also discussed. The creation of cell sets for volume mesh representations is also presented.

Contents:

What Are Representations?
Displaying Representations
Using Proxy Representations
Replacing a Surface Mesh
Preparing a Surface Mesh
Exporting a Surface Mesh
Extracting a Surface Mesh
Using Cell Sets

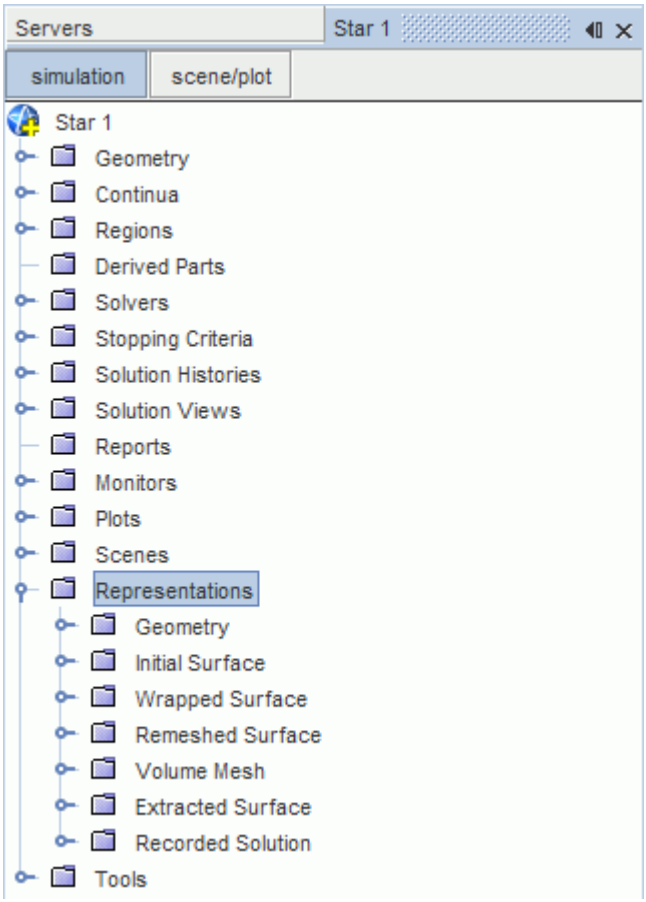
What Are Representations?

Representations help you keep track of the surface and volume meshes that STAR-CCM+ uses as part of the meshing and simulation setup process. Several different representations are available.

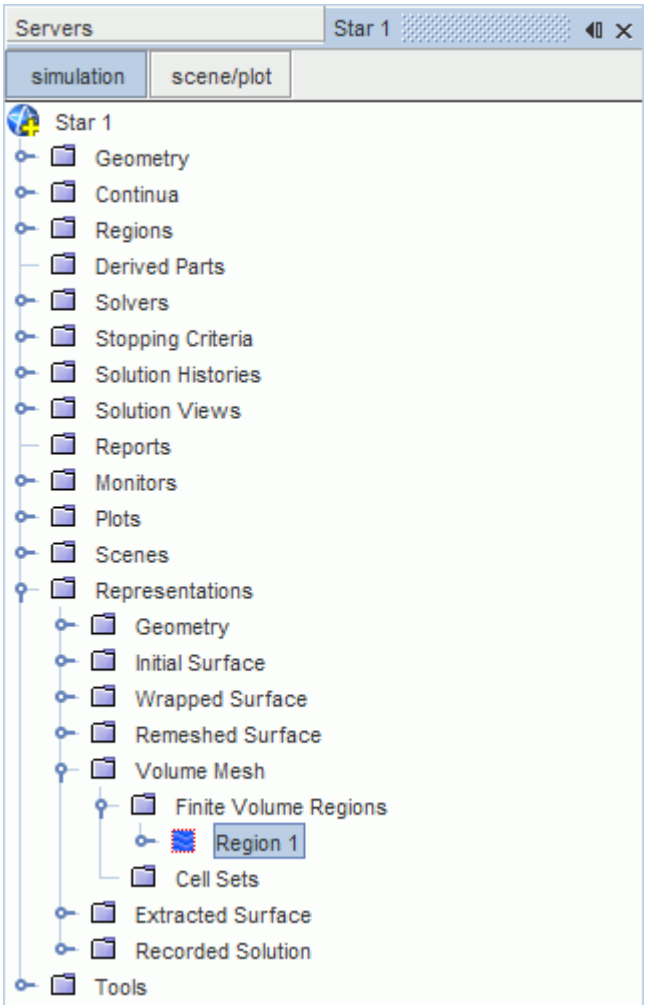
The types of representations are:

- **import** - imported surface mesh (using the regions import mode option)
- **geometry** - geometry surface (using the parts import mode option)
- **initial surface** - initial surface from geometry parts
- **wrapped surface** - surface wrapper mesh
- **remeshed surface** - surface remesher mesh
- **volume mesh** - volume mesh (with optional prismatic layers)
- **extracted surface** - surface boundary mesh that is extracted from a volume mesh
- **proxy** - placeholder that you apply to scenes, report, and plots so that you can control the representation of these items from one place
- **recorded solution** - copy of the volume mesh that is stored on an external solution history file (containing selected solution data)
- **overset mesh** - contains the background mesh and all overset region cells, regardless of whether they are active or inactive

The representations can be seen by clicking the **Representations** node in the simulation tree. For example:



A representation node is added whenever an operation results in the creation or import of that kind of mesh. For example, when a new simulation is made, no representations exist. When a surface mesh is imported using the regions import mode option, then an **Import** node appears in the simulation tree. Similarly, importing a surface as a part results in a **Geometry** node being created instead. If a volume mesh is imported or generated, then a **Volume Mesh** representation node is shown. Except for the **Geometry** node (which simply acts as a place holder), the properties of the mesh can be seen by clicking the node and viewing the *Properties* for it. For example, the total number of cells, interior faces and vertices for a volume mesh region can be seen by selecting the region name of interest:



Only one of each kind of mesh representation can exist for a given simulation although multiple extracted surfaces are allowed. The result of a second import operation adds the properties of the new mesh to the existing node. The result of generating a new mesh replaces the properties of the existing node. Representation nodes can be deleted like any other object, but at least one representation must always remain. All meshes (and appropriate representations) generated within STAR-CCM+ can be cleared in one go if preferable.

See Also:

The Surface Repair Tool

Displaying Representations

The new surface or volume mesh representations that are generated using the meshing tools are not automatically displayed in the plot window once the meshing process is complete.

The mesh representations can be visualized using three different methods:

- Selecting the appropriate representation from the scene displayer pull-down menu in the *Properties* window.

- Selecting the appropriate representation by right-clicking the graphics display.
- Dragging and dropping the representation node object onto the graphics display.

Only the representations that have so far been imported or generated are listed in the menus.

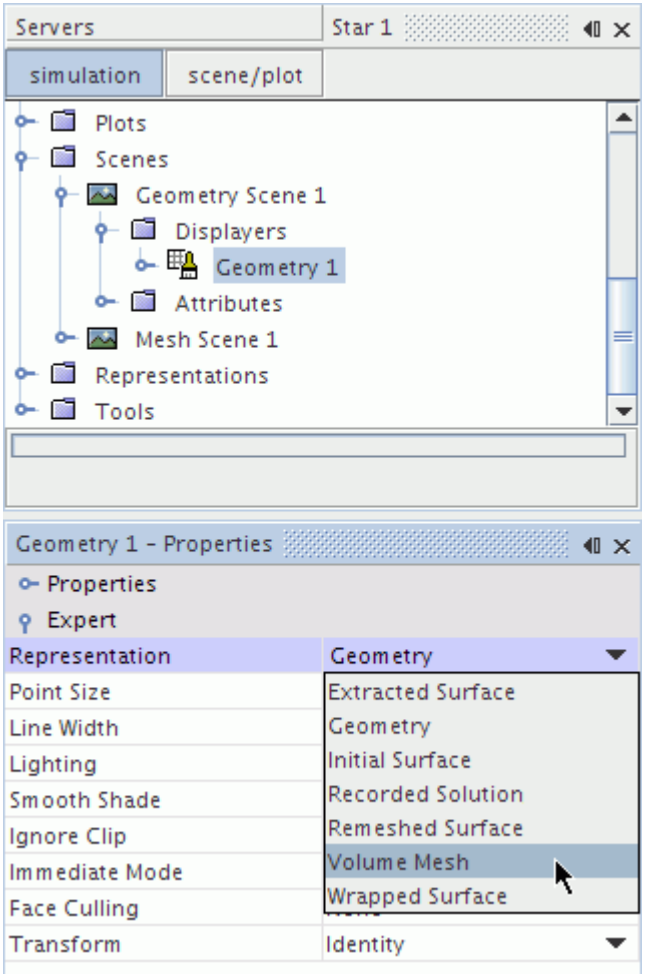
Contents:

Displaying Representations using the Scene Displayer
Displaying Representations via the Graphics Display
Displaying Representations by Dragging and Dropping

Displaying Representations using the Scene Displayer

Each scene displayer node provides access to the mesh representations that currently exist.

To change the current representation, select the scene displayer node of interest (for example, **Geometry 1**) and then click the *Representation* option in the expert *Properties* window. The currently available mesh representations are listed in the pull-down menu.

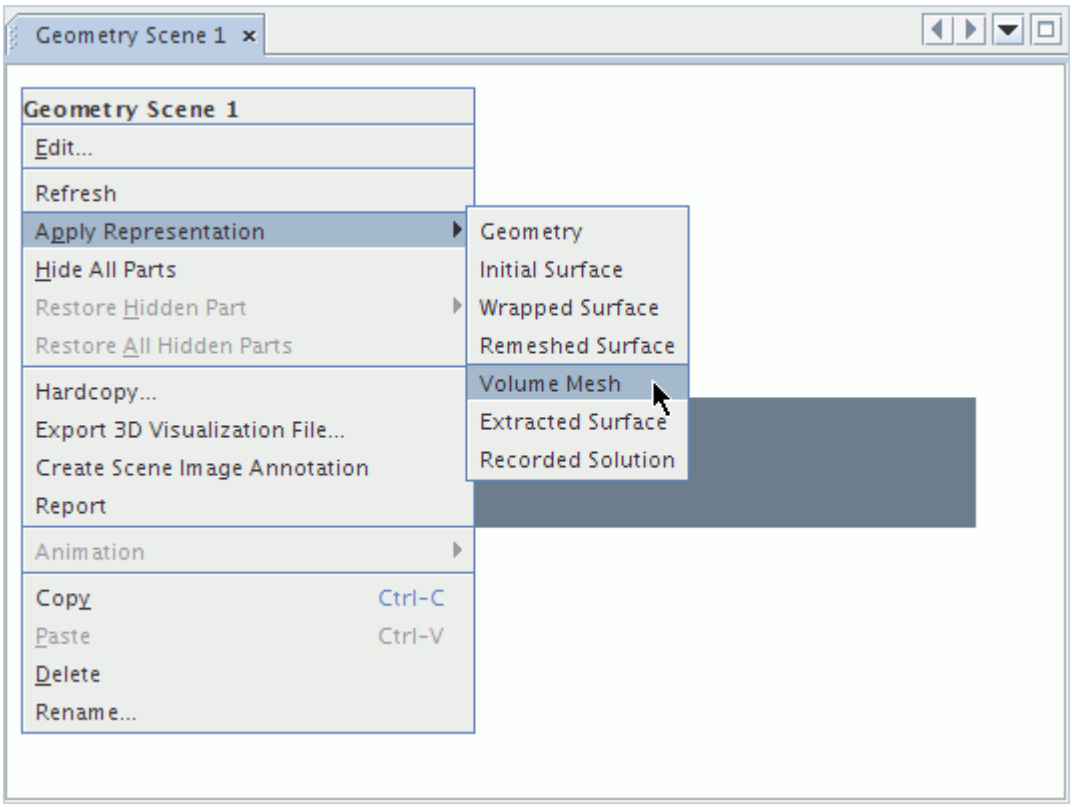


Select the appropriate mesh representation from the list and the scene updates with the new mesh. Any other scene displayer nodes (for example, **Outline**) for the same scene does not have the same representation selected for them - update these using the same technique as described above.

Displaying Representations via the Graphics Display

The pop-up menu in each scene display provides access to the mesh representations that currently exist.

To change the current representation, right-click any portion of the display that the model does not occupy and select **Apply Representation** in the pop-up menu that appears. The currently available mesh representations are listed.

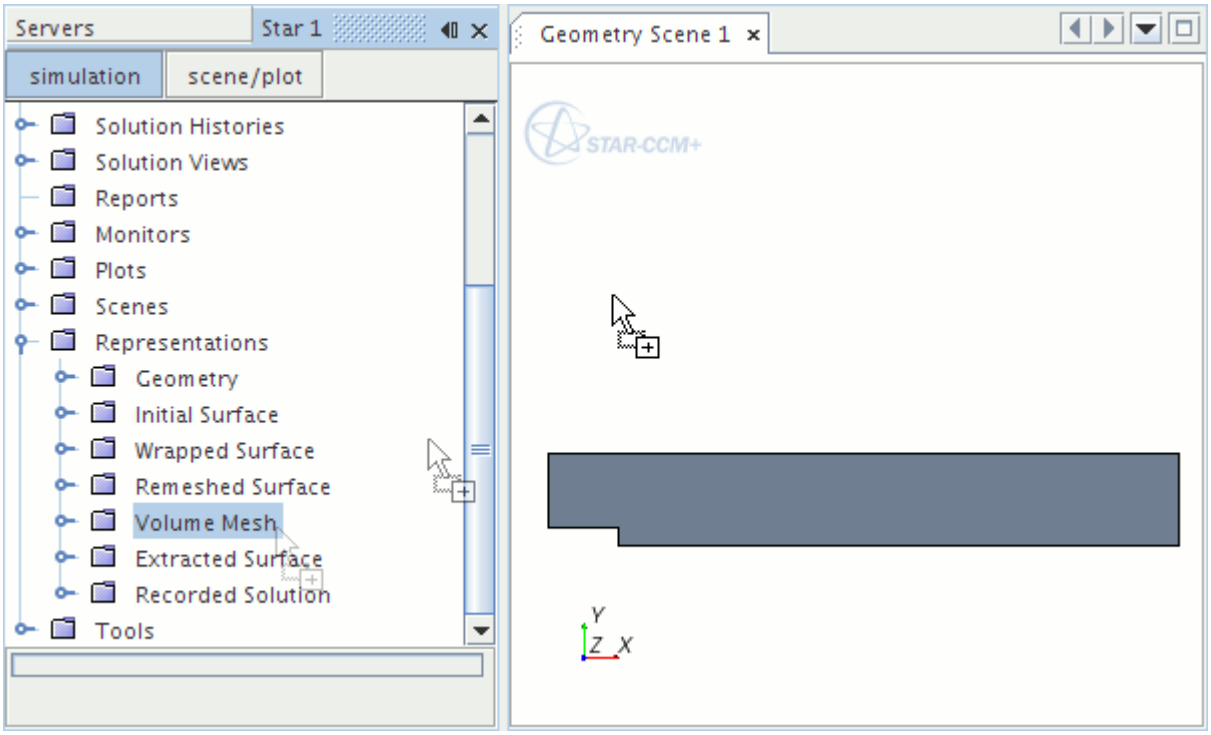


Select the appropriate mesh representation from the list and the scene is updated with the new mesh. Any other scene displayer nodes belonging to the scene (for example, **Outline**) have the same representation selected for them automatically using this method.

Displaying Representations by Dragging and Dropping

In order to update the scene with the new mesh representation, each mesh node in the **Representation** node can be dragged and dropped onto the graphics display scene.

An example of using the drag-and-drop feature is illustrated below:



All displayers belonging to the scene are automatically updated to have the same representation using this method.

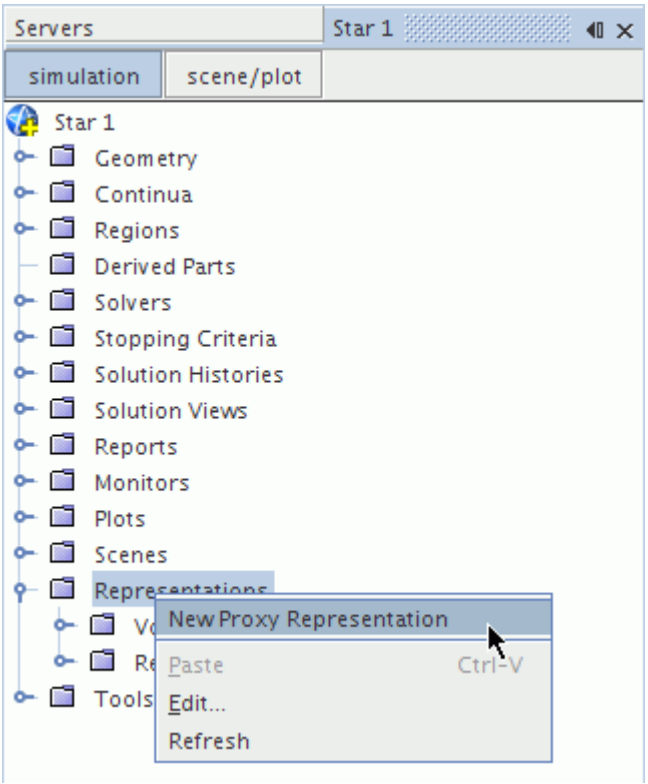
Using Proxy Representations

A proxy representation acts as a placeholder and allows you to control the representation of multiple objects (scenes, reports, and plots) from a single location.

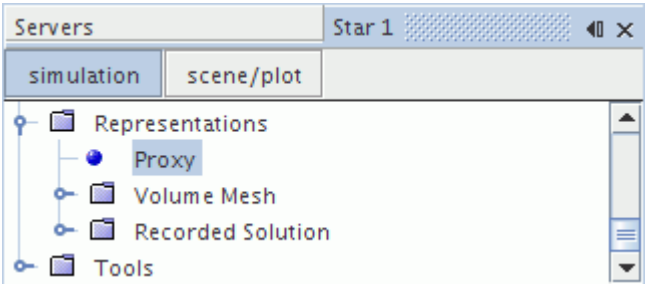
Often, you have multiple plots and reports in your simulation. If you want to apply the same representation to all these objects, you could manually edit each object in turn and change the representation. Instead, by using a proxy representation you only need to set the representation once in the various objects. You can then control these representations using the proxy representation.

To create a proxy representation:

1. Right-click on the **Representation** node in the object tree and select **New Proxy Representation**.

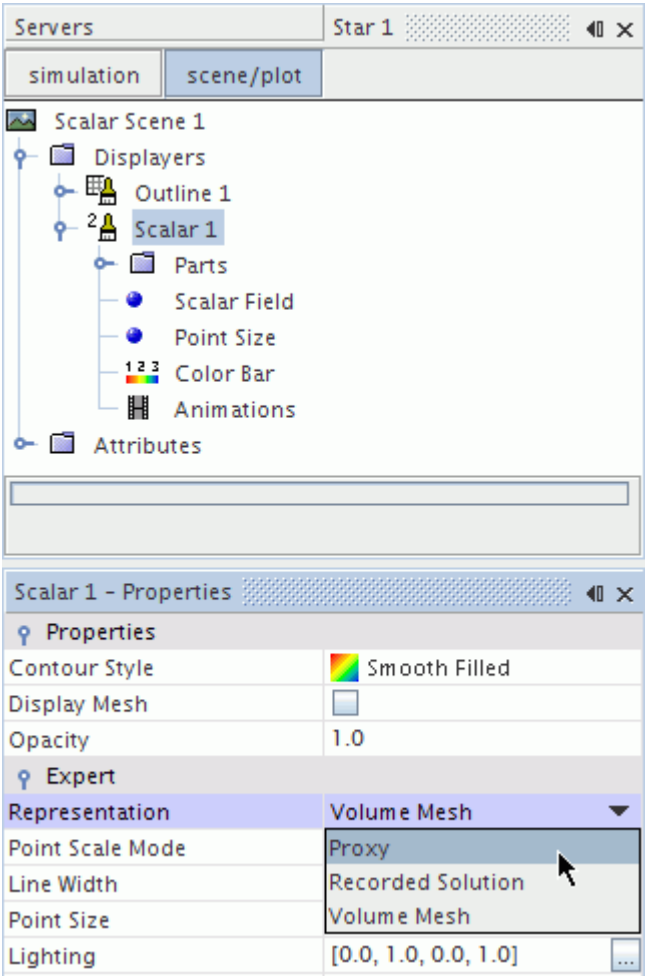


A proxy representation, **Proxy**, appears in the object tree.

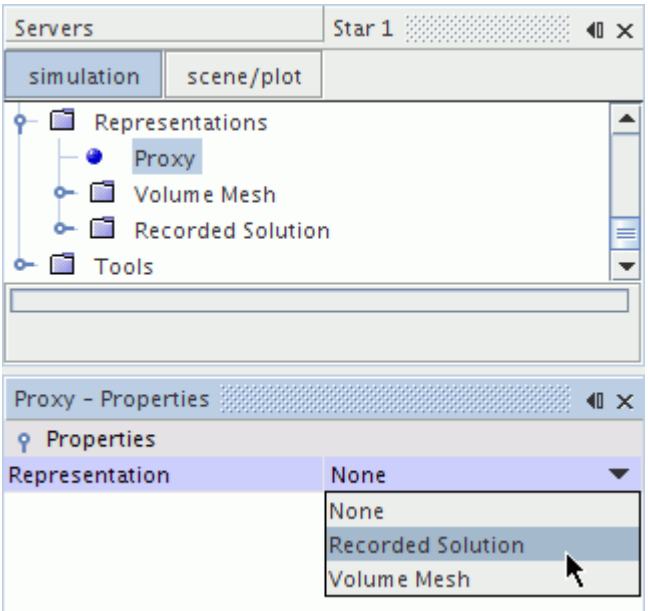


Note: You can create and use as many proxy representations as you want.

2. Navigate to the objects where you want to change the representation. Set the *Representation* to the **Proxy** representation.



3. Return to the **Representation** > **Proxy** node. In the *Properties* window, select the representation you want to apply to the objects. In the example, a solution history representation is selected.



The representation you select in the proxy is applied to every object that is assigned the proxy representation. You can quickly change the representation that is applied to the object by changing the proxy representation.

Replacing a Surface Mesh

Any of the five types of surface mesh representations, namely import, initial surface, remeshed surface, wrapped surface and extracted surface, can be replaced at any given time by a new surface read in from an external file.

There are two main reasons for wanting to replace a surface mesh:

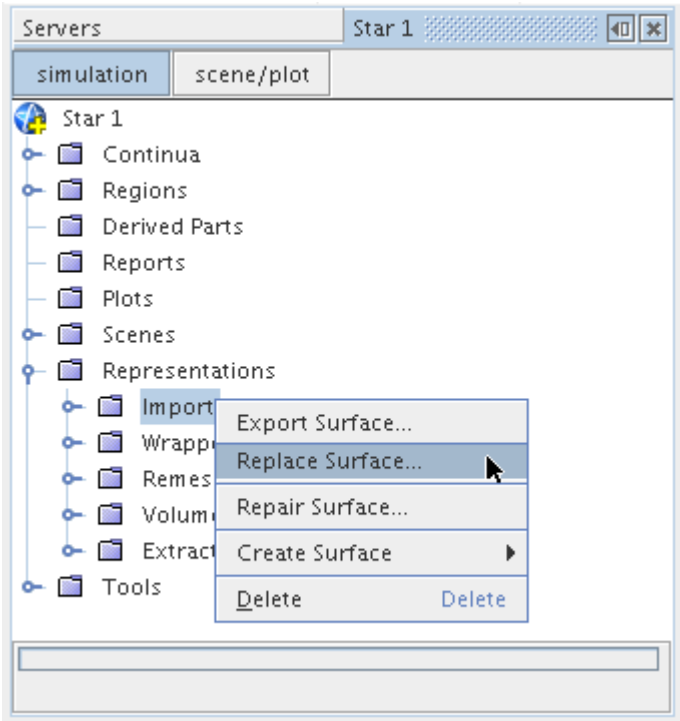
1. The new surface contains fixes performed by some other third-party pre-processing software; and
2. The new surface contains a geometrical design change from the original

Replacing a surface does not disrupt the meshing pipeline and is a convenient method for performing multiple mesh runs on different surfaces since all parameter settings and values are preserved.

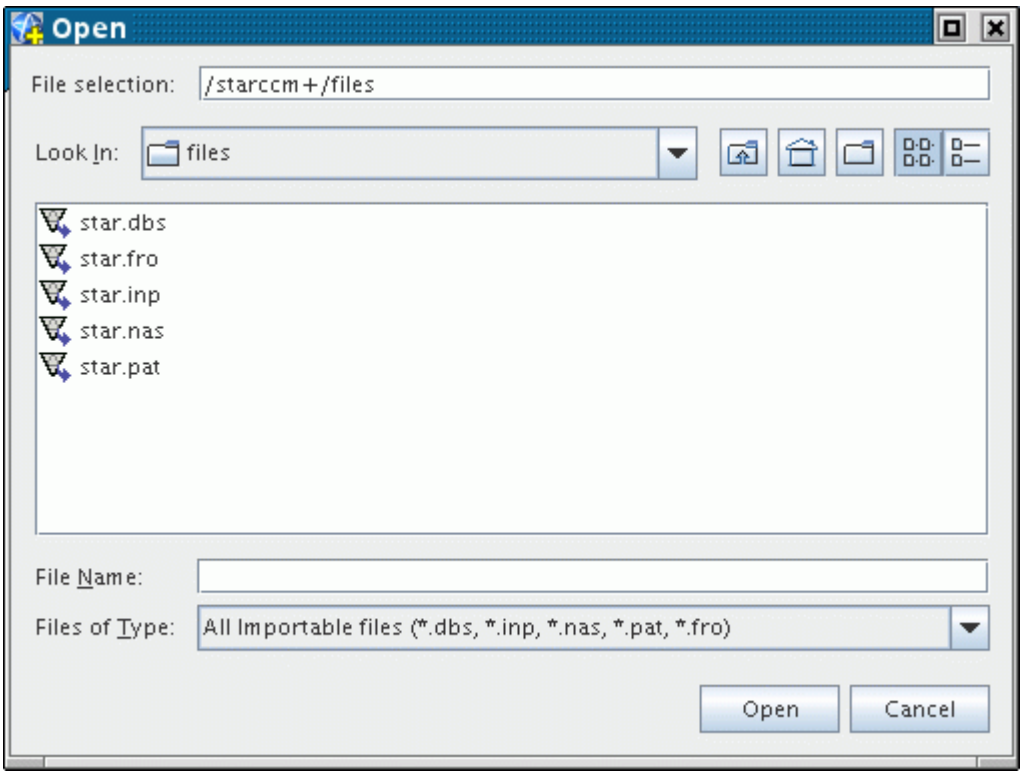
The currently allowable surface mesh replacement formats are:

- .dbs - pro-STAR surface database mesh file
- .inp - pro-STAR cell/vertex input file
- .nas - NASTRAN shell file
- .pat - PATRAN shell file
- .fro - FELISA front surface file

To replace a surface, right-click on the surface mesh name and select **Replace Surface...** from the pop-up menu. An example for the **Import** node is shown below:



The *Open* dialog appears.



Select the surface file that you want to import and click **Open** to import the surface. You are then prompted for the units for the new surface file before it is imported. If the underlying patch definitions for the original and replacement surface are the same, then the current boundary list is maintained. If extra patches exist in the new surface, then new boundaries are created to represent them. In instances where patches are missing from the new surface, deselect existing boundaries that become empty.

When using the .dbs or .inp formats for replacing a mesh, an option to **Match Boundaries by Name** is provided in the interim dialog. This option is turned off by default, meaning that no attempt is made to match old and new boundaries based on their names but instead their cell types are used. Turning on this option matches patches to boundaries based on their names instead of cell types. The process of matching boundaries by name is case-sensitive.

If you are replacing a surface mesh with a surface of .inp format that was exported from an existing STAR-CCM+ simulation, underscores replace any spaces in boundary names in the .inp file. As a result, STAR-CCM+ may be unable to match boundaries by name when the surface mesh is replaced. Under these circumstances, it is recommended that you export the surface mesh to the .dbs format instead.

To replace the volume mesh, use the import option in the **File** menu to reimport a whole new mesh or alternatively you can replace a single mesh region.

Preparing a Surface Mesh

Any of the surface mesh representations can be manually prepared using the Surface Preparation tool that is contained within STAR-CCM+.

Exporting a Surface Mesh

Any of the five types of surface mesh representations, namely import, initial surface, remeshed surface, wrapped surface and extracted surface can be exported from STAR-CCM+ for further manipulation.

The currently allowable export formats for surfaces are:

- .dbs - pro-STAR surface database mesh file
- .inp - pro-STAR cell/vertex input file
- .nas - NASTRAN shell file
- .pat - PATRAN shell file
- .stl - Stereolithography file
- .fro - FELISA front surface file
- .bgp - XML boundary grouping

All the above formats support single or multiple region definitions. All the formats other than PATRAN and STL support feature curves. It is possible to export the entire representation, single regions within the representation, or boundaries belonging to regions within the representation.

Contents:

Exporting the Complete Surface
Exporting Individual Regions
Exporting Boundaries

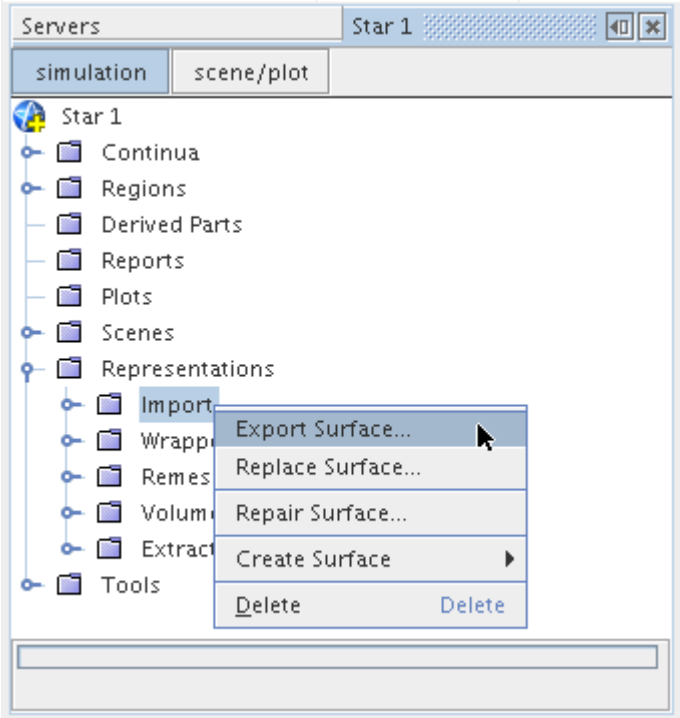
Exporting the Complete Surface

You can export any surface representation to one of multiple file formats.

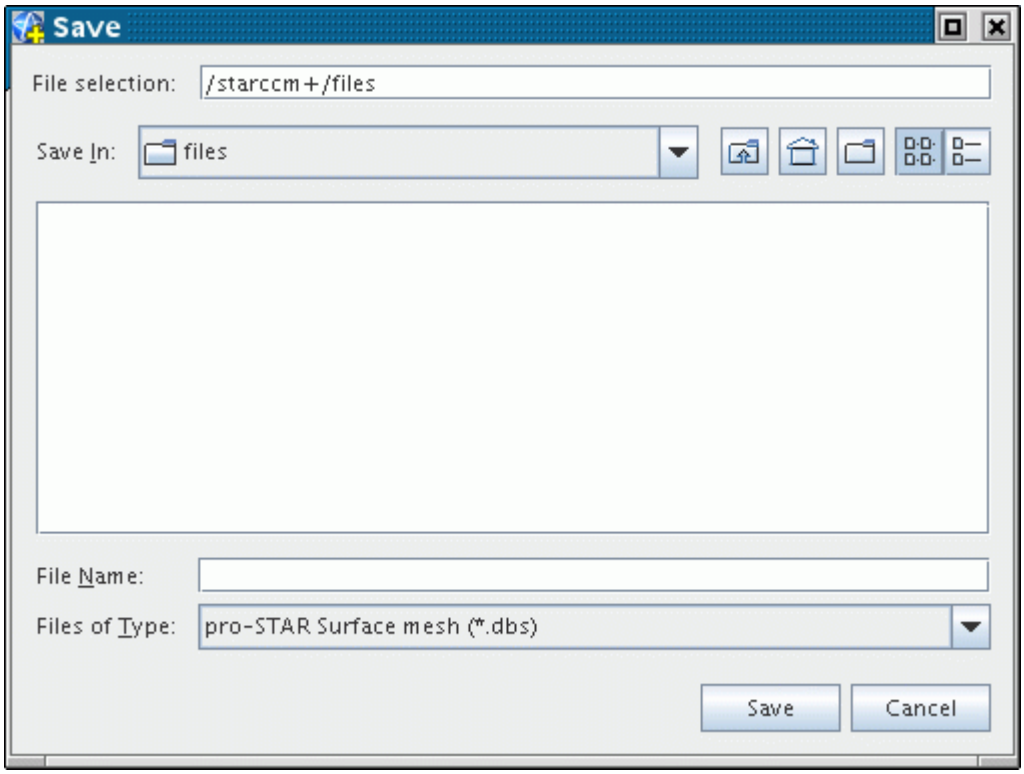
When remeshing a region using the Region-Based meshing approach, a representation is created which you can export. However, if you remesh a part using the Parts-Based meshing approach, a description is created instead of a representation. To export a surface description, see Exporting Mesh Descriptions.

To export the complete surface representation:

1. Right-click on the representation node of interest and select **Export Surface...** from the pop-up menu:



The *Save* dialog appears:



2. Select the file type from the **Files of Type** pull-down menu.
3. Enter the **File Name** and click **Save**.

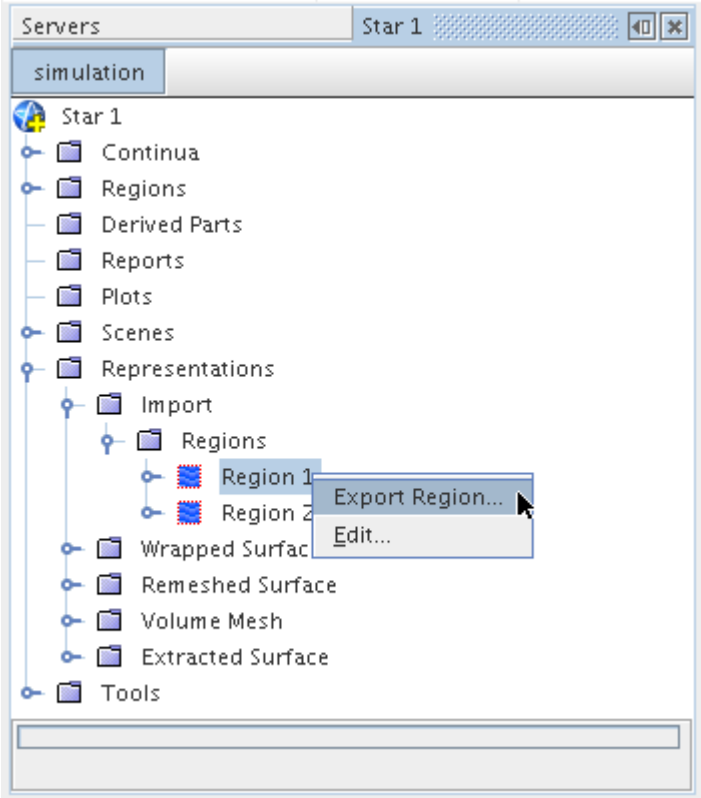
Before the file is saved, details that are particular to the file format can be required. When using the Nastran file type, the default option is to export the geometry using the **Extended** file format, which provides greater node precision (G16.9) as compared to the **Standard** option (E8.3). A loss in surface quality is therefore encountered with the latter.

An export only format (XML Boundary Grouping) is allowed here if you wish to reconstruct the regions/boundaries hierarchy that is established in STAR-CCM+. Reconstruction is possible when the surface patch data (exported using any of the other allowable formats) is imported into a third-party package that also supports the XML boundary grouping format.

Exporting Individual Regions

As an alternative to exporting the entire surface, you can export individual regions belonging to each surface type representation.

1. To export a region, right-click the region of interest and select **Export Region...** from the pop-up menu:

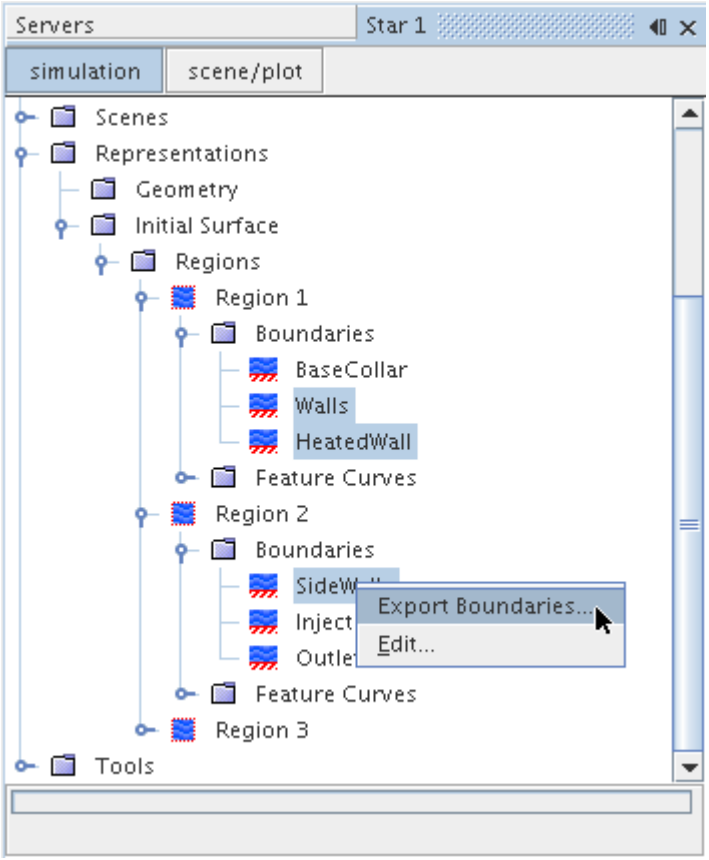


The *Save* dialog appears as before, allowing just the surface mesh for the selected region to be saved in the desired format. Only one region can be selected at a time for this export process.

Exporting Boundaries

Boundaries belonging to regions within each surface representation can be exported, either singly, or with others selected at the same time. It is also possible to export multiple boundaries from different regions in a single operation.

1. To export one or more boundaries, first select their nodes, right-click, and choose **Export Boundaries...**



The *Save* dialog appears as for other export operations, allowing the selected boundary or boundaries to be saved in the desired format. The *File Name* field is automatically populated with the name of the selected boundary (or the first boundary if more than one is selected).

Not all formats preserve boundary names: only the two pro-STAR formats and the NASTRAN format retain actual names (with duplicate names receiving an additional integer index); PATRAN and FELISA formats retain individual boundary definitions without the names; the Stereolithography format combines all boundaries into one. Region definitions are not retained in any format.

Extracting a Surface Mesh

This facility extracts a surface mesh from a volume mesh representation so that you can generate a new volume mesh using the STAR-CCM+ meshing tools.

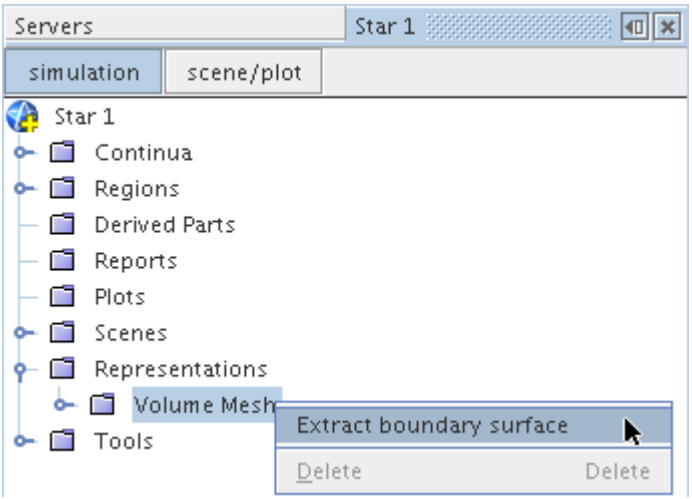
Surface mesh extraction is useful when you have a volume mesh (for example, a tetrahedral mesh) from another package, but want to remesh it using the different mesh types in STAR-CCM+. This can result in a lower cell count or better cell quality (or usually both).

Extracting a surface mesh maintains the boundary and interface definitions. However, exporting a surface mesh representation does not preserve boundary types, except from name association, and never preserves interfaces. Therefore, after importing an extracted surface, check the surface mesh, redefine interfaces and possibly redefine boundary types. If you are importing the surface mesh as a part, you may also need to

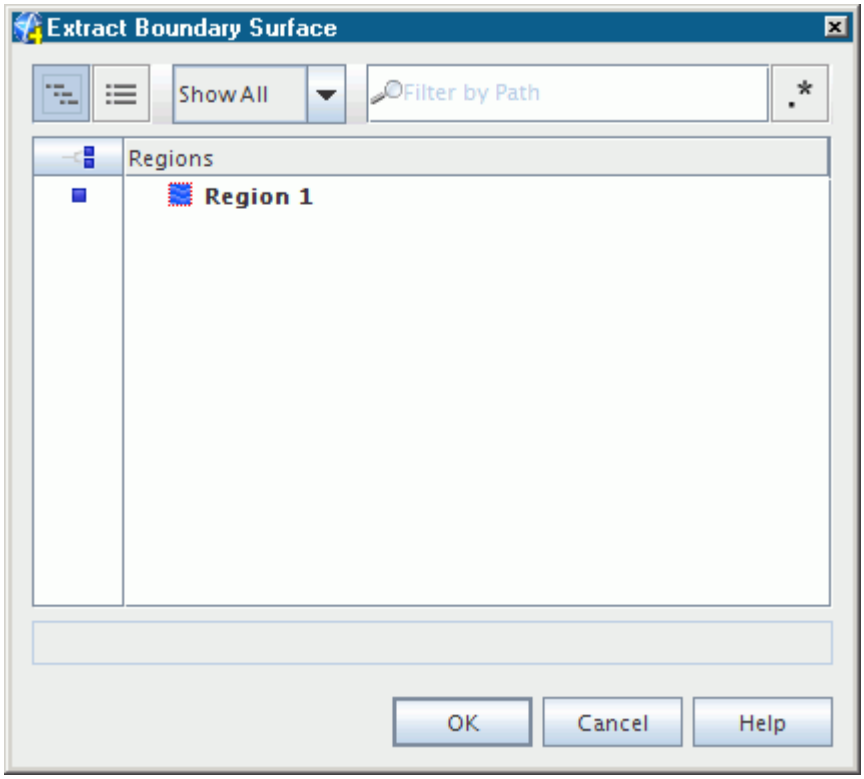
merge/imprint coincident surfaces, so that interfaces are conformal. Also note, volume meshes do not store feature edges from the surface mesh. Therefore, when extracting a surface, redefine the feature edges on the extracted surface mesh.

Before extracting a surface mesh, a volume mesh representation must exist (generated when importing a volume mesh). To extract a surface mesh from a volume mesh:

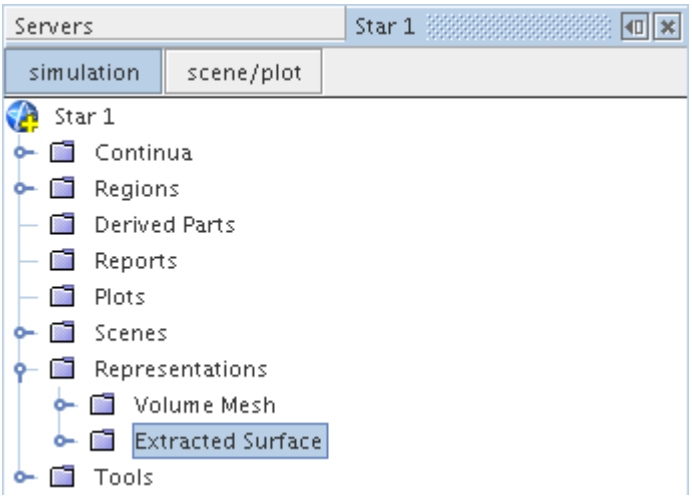
1. Right-click the **Volume Mesh** node and select **Extract boundary surface** from the pop-up menu.



2. In the *Extract Boundary Surface* dialog, select the required region(s).



When the process is complete, a new representation called **Extracted Surface** appears in the simulation tree. Each extracted surface representation contains a closed surface mesh of the selected region(s)



If you extract multiple regions, multiple nodes appear in the **Representations** node with numerically incremented names.

The extracted surface can now be used as the initial surface mesh for a new meshing process within STAR-CCM+. If other surface mesh representations exist, delete them so that the meshing pipeline uses the correct surface.

To access the full range of boundary manipulation options after extracting a surface mesh, create a mesh continuum manually if one does not exist before performing the operation.

See Also:

Extruder Mesher

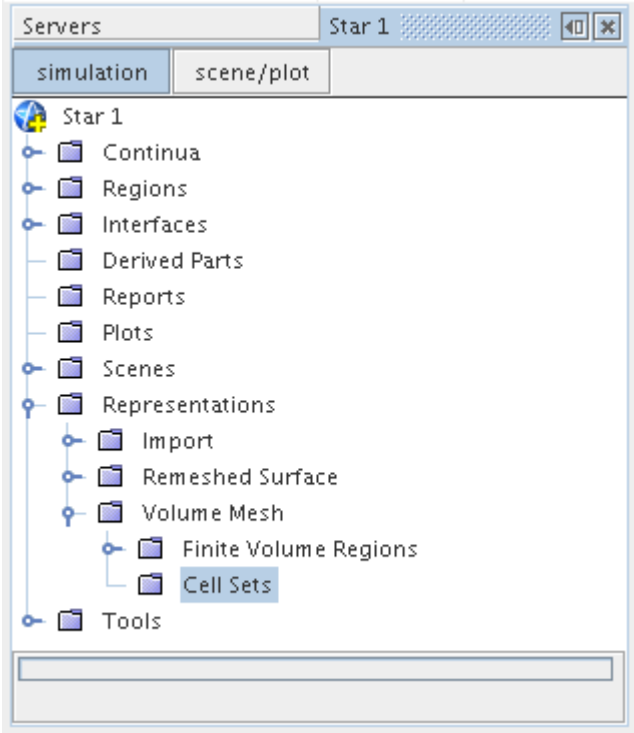
Using Cell Sets

The purpose of a cell set is to allow you to identify collections of cells that are useful for visualizing the model, such as during post processing or mesh checking.

STAR-CCM+ allows cell sets to be created once you have a 3D volume mesh representation. Any number of cell sets can be created for a given volume mesh, with each one containing anywhere from zero to the total number of cells in the model.

A cell set is similar to a derived part in many respects except that it is more versatile and can be derived from more than one condition. For example, you can create a cell set based on a combination of a threshold and a zone in order to obtain the cell grouping you want. For this reason, it is a powerful tool for isolating and storing cell-based parts for further use.

All operations relating to cell sets take place via the **Cell Sets** node, which appears under the **Finite Volume Regions** node in the **Volume Mesh** representation in the simulation tree:



A new cell set can be created based on a cell list, a scalar threshold or a zone. Once a cell set has been created, it can be modified if desired using a number of global options such as fill, clear, grow and shrink. Boolean operations are also possible to add to a set, take a subset of a set, remove from a set and replace a set altogether, using the same options as used to create the original set.

Cell set operations only allow the groupings of cells that exist in the model. No cells can be deleted or created by this process. The cell set operations described here are also currently only applicable to 3D meshes - if you have a 2D mesh then a cell set cannot be displayed.

Each cell set created within STAR-CCM+ has a corresponding field function whose *Function Name* is a contracted form of the cell set name. This field function can be used within other field function definitions to detect whether a cell belongs to the specified cell set or not. This feature is useful when applying region sources to a particular set of cells, for example.

Contents:

Creating a New Cell Set
Modifying an Existing Cell Set
Copying and Pasting Cell Sets

Creating a New Cell Set

Generate a new cell set before any further manipulation can be done on the set or the set contents displayed.

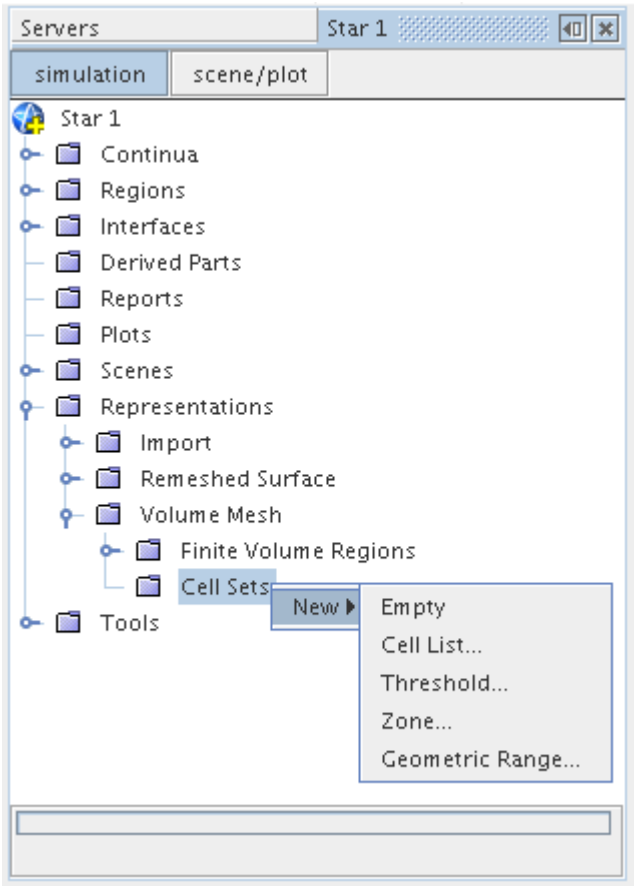
By default a 3D volume mesh representation has no cell sets defined for it. If you have a 2D volume mesh, then a cell set cannot be displayed for it.

Four methods are available for creating a cell set:

- Using a cell list;
- Using a scalar threshold;
- Using a zone; and
- Using a geometry range

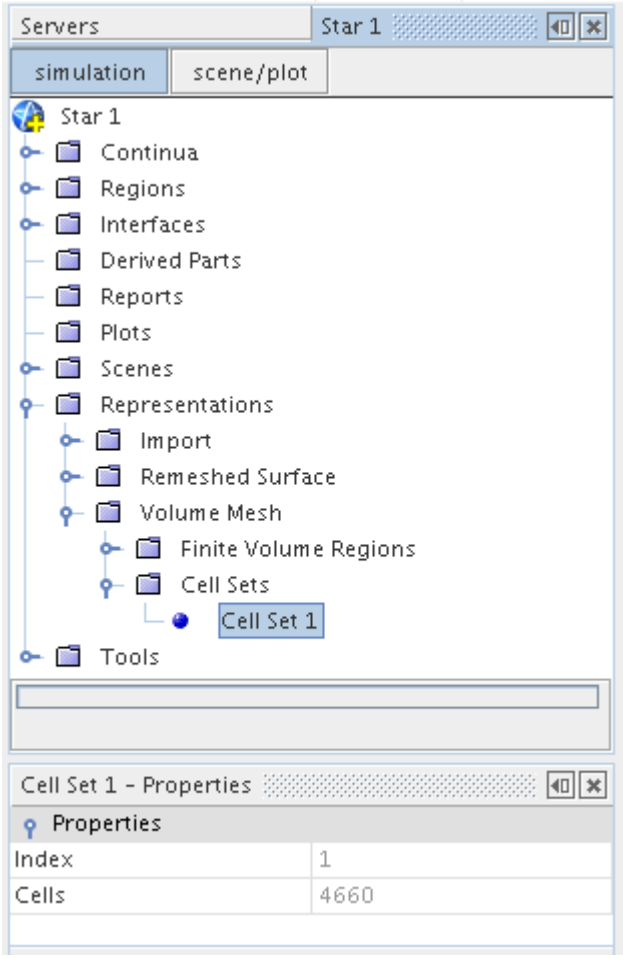
Each new cell set created is based on all the cells in the model, regardless of any previously generated sets. For the cell list and threshold options, a dialog window appears with the appropriate options to allow you to create the cell set, while the zone and geometric range options allow you to select the zone or block of cells of interest manually. In all four instances however, you are prompted as how to add the new cell set part to the current scene in order for it to be displayed.

To view the GUI options available for creating a cell set, right-click on the **Volume Mesh** > **Cell Sets** node and select the **New** option:

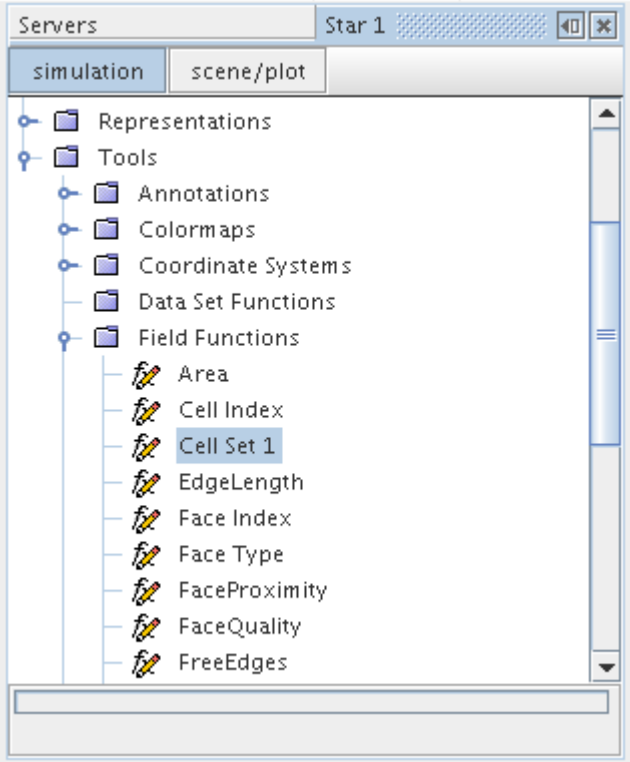


The **Empty** cell set option allows you to create an empty cell set node, which can be modified afterwards to include cells using any combination of the tools provided.

Once a cell set has been created, a new sequentially numbered node, starting from **Cell Set 1**, is created under the **Cell Sets** node. Clicking the node provides a summary of the number of cells that are contained in the set:



A field function is also created for each cell set by default, which appears under the **Tools** > **Field Functions** manager node. The field function has the same name as the cell set by default. For example:



This cell set field function can be useful is situations such as when a region needs to be split based on an existing collection of cells. Once the original cell set is deleted, the associated field function is also deleted.

The new cell set creation tools only allow a single action to create the original cell set. If you want to apply additional criteria to manipulate the set further, then modify the cell set using the range of global and Boolean tools available.

Cell set nodes can be renamed and deleted in the same way as other nodes. They can also be copied and pasted to perform Boolean operations between two different sets.

Contents:

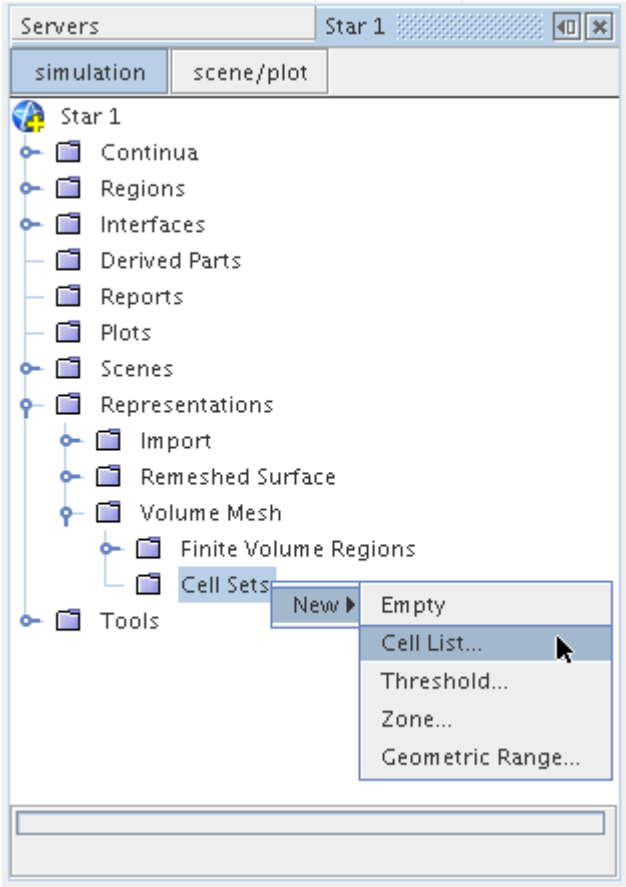
Creating a New Cell Set Using a Cell List
Creating a New Cell Set Using a Threshold
Creating a New Cell Set Using a Zone
Creating a New Cell Set Using a Geometric Range
Creating a New Empty Cell Set
Example: Creating a Cell Set to View the Outer Layer of Cells on a Mesh

Creating a New Cell Set Using a Cell List

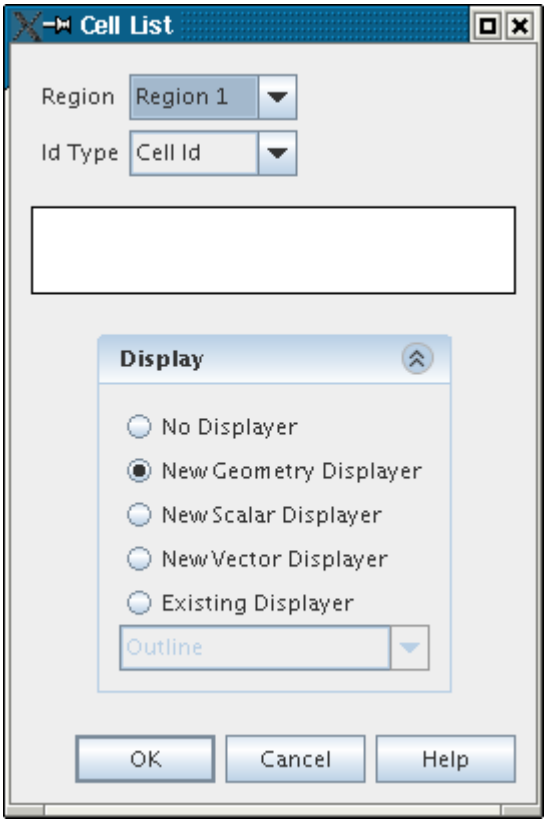
The cell list option for creating a cell set allows individual cell ID's or offsets to be used to identify cells. In general, these options are included for developers to debug problems as the ID's and offsets are not generally available or known to the average user.

Cell ID's are unique values assigned to each cell and correspond with the pro-STAR cell ID index. Cell offsets are used when the mesh has been partitioned for a parallel analysis.

To access the cell list options, right-click on the **Cell Sets** node and select **New** > **Cell List...** from the pop-up menu:



The *Cell List* dialog appears.



In the dialog, select the region of interest using the **Region** pull-down menu and then the method that is used to set the **Id Type**, namely **Cell Id** or **Cell Offset**. The individual cell ID's or offset values can then be entered into the white box, separated by spaces.

Then select the *Display* option to determine how the new cell set part is added to the current scene. Click **OK** to create the cell set based on the supplied information or **Cancel** to void the operation.

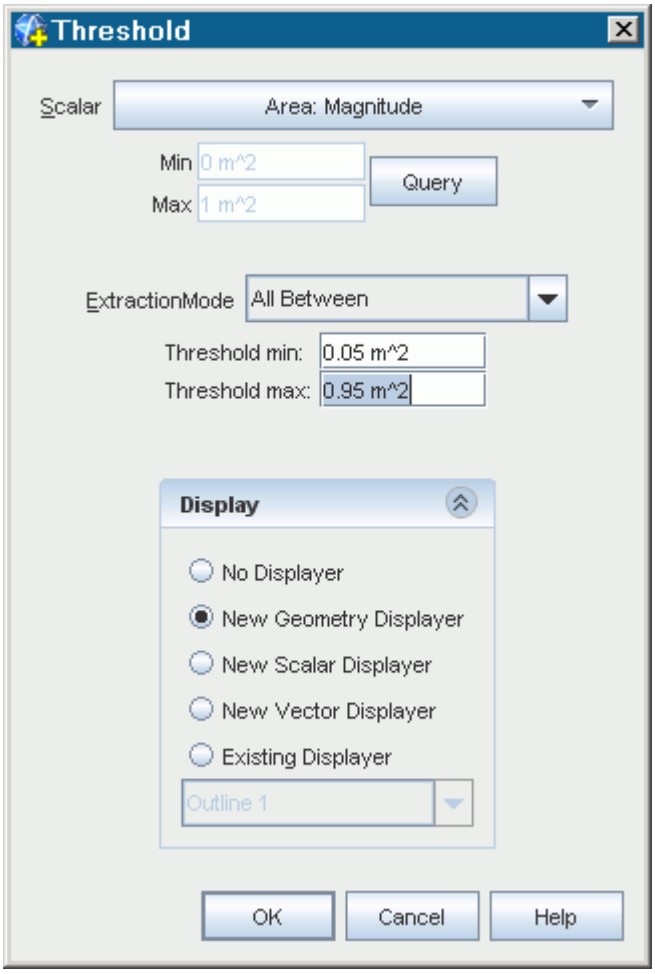
The *Cell List* dialog is automatically closed once the operation is complete. If you want to apply additional criteria to manipulate the set further, then modify the cell set using the range of global and Boolean tools available.

Creating a New Cell Set Using a Threshold

The threshold option allows cell sets to be created based on any of the normally available scalars such as centroid geometric range, face area magnitudes and volumes, as well as post processing items such as pressure, temperature and velocity etc, should they be available.

To access the threshold options, right-click on the **Cell Sets** node and select **New** > **Threshold...** from the pop-up menu.

The *Threshold* dialog appears:



First select the **Scalar** field function of interest using the pull-down menu. If desired, click **Query** to query the range of field values in the entire cell set. Set the **Extraction Mode** to determine the method that is used to extract the cells for the threshold. Enter the appropriate minimum/maximum or threshold value depending on the option selected.

Then select the *Display* option to determine how the new cell set part is added to the current scene. Click **OK** to create the cell set based on the supplied information or **Cancel** to void the operation.

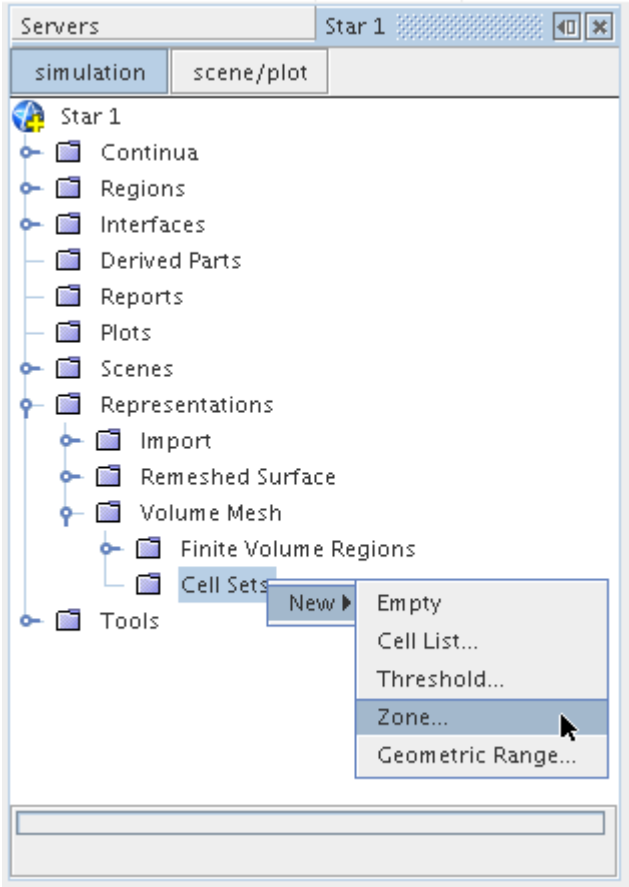
The *Threshold* dialog is automatically closed once the operation is complete. If you want to apply additional criteria to manipulate the set further, modify the cell set using the range of global and Boolean tools available.

Creating a New Cell Set Using a Zone

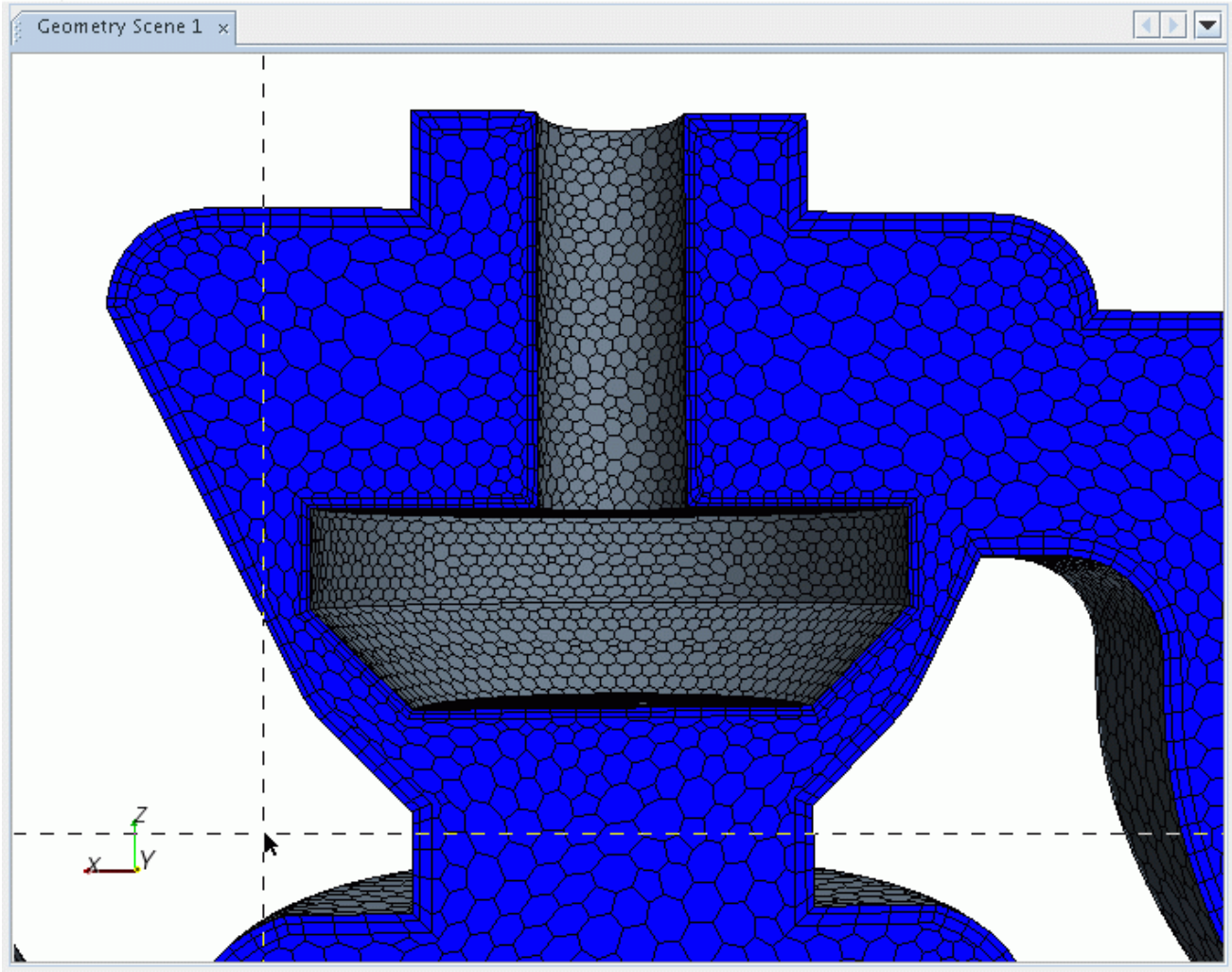
The zone option allows cell sets to be created based on cells that are contained fully within an arbitrary closed loop polygon. As such, it is a very flexible and quick method of isolating cells with just a few mouse clicks.

Note: A valid scene must be displayed in order for this option to be active.

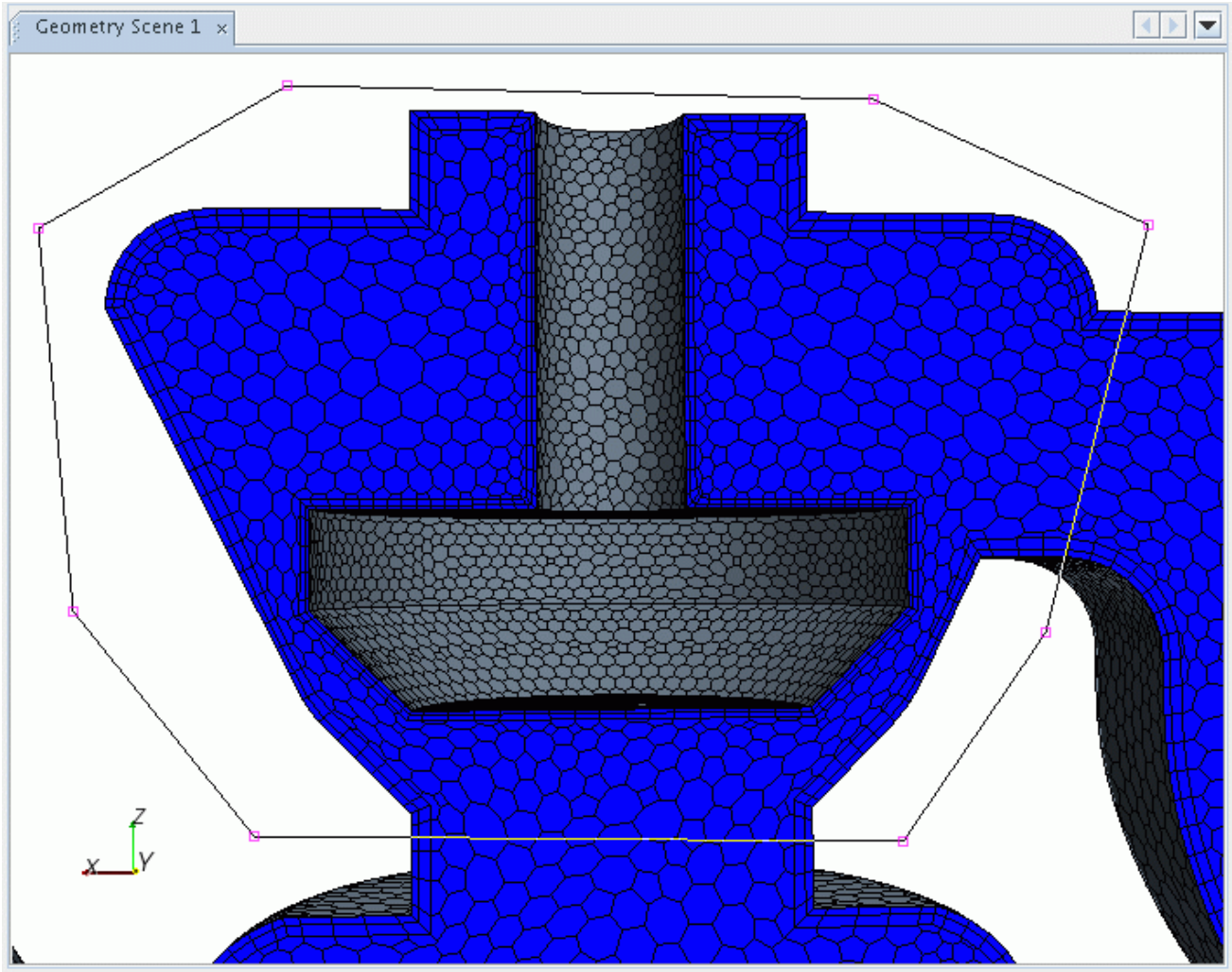
To access the zone options, right-click the **Cell Sets** node and select **New** > **Zone...** from the pop-up menu:



This activates the mouse cross-hair selection tool, shown below:

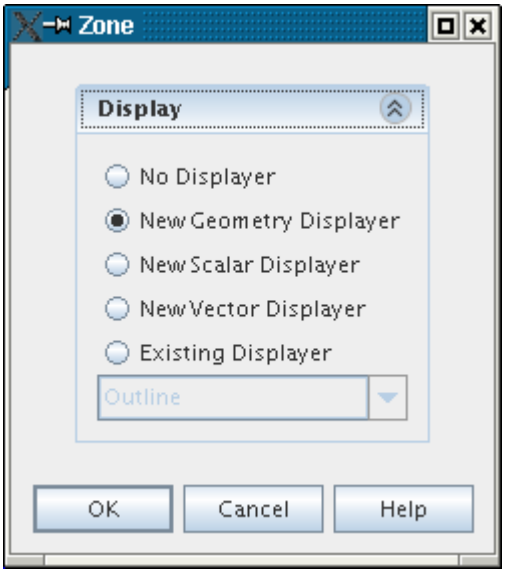


Click the scene to indicate the starting point of the polygon zone and then continue to click around the model to create a closed loop zone of the area of interest. An example is shown below.

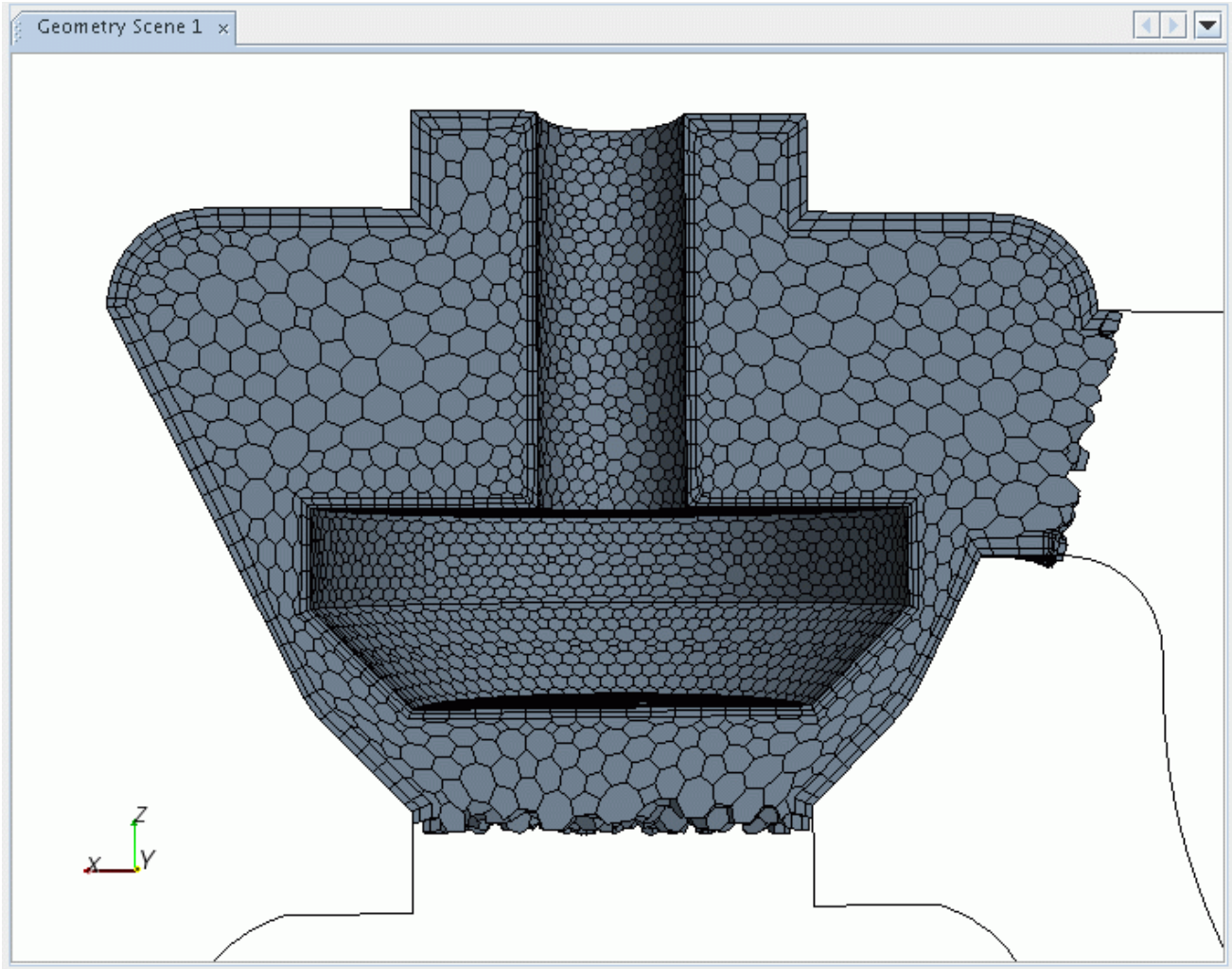


To complete the zone, you can either right mouse-click or click back on the starting point of the zone. If at anytime you want to discontinue the zone, press the Escape key. Press the D key to undo the previous zone point selection.

On the successful completion of a zone, the *Zone* dialog appears to indicate the display choices:



Then select the *Display* option to determine how the new cell set part is added to the current scene. Click **OK** to create the cell set. If the appropriate display options are selected, the cell set automatically appears within the scene:



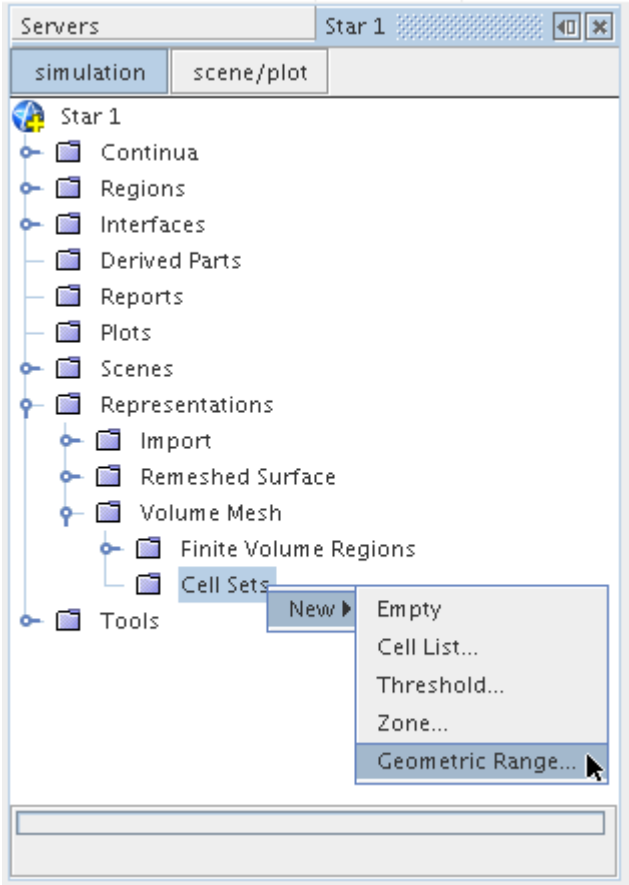
If you want to apply additional criteria to manipulate the set further, modify the cell set using the range of global and Boolean tools available.

Creating a New Cell Set Using a Geometric Range

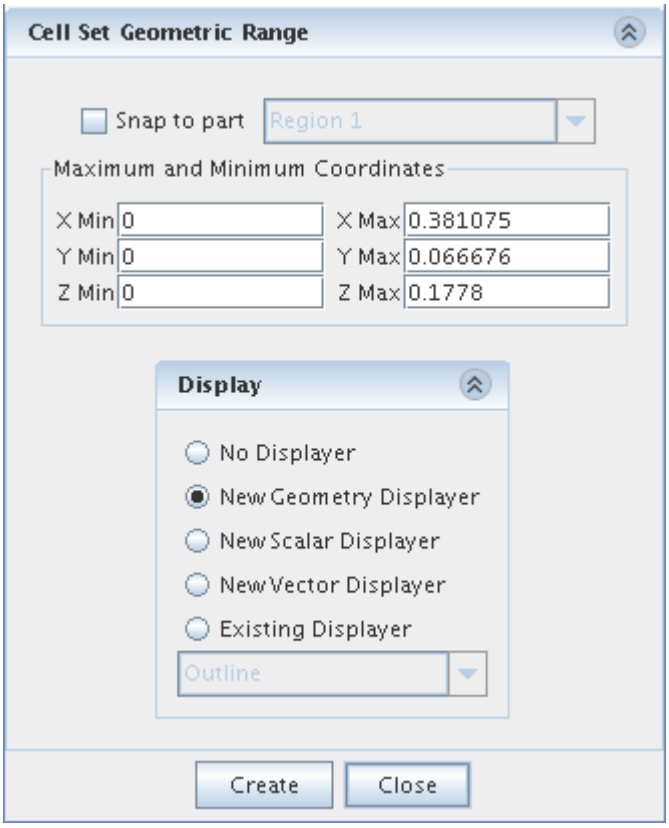
The geometric range option allows cell sets to be created based on cells that are contained fully within a Cartesian based block, extending in the x, y and z directions.

If a valid scene is currently displayed, the minimum and maximum geometric coordinates can be set either graphically or manually, otherwise they can be entered manually by hand if a scene is not open.

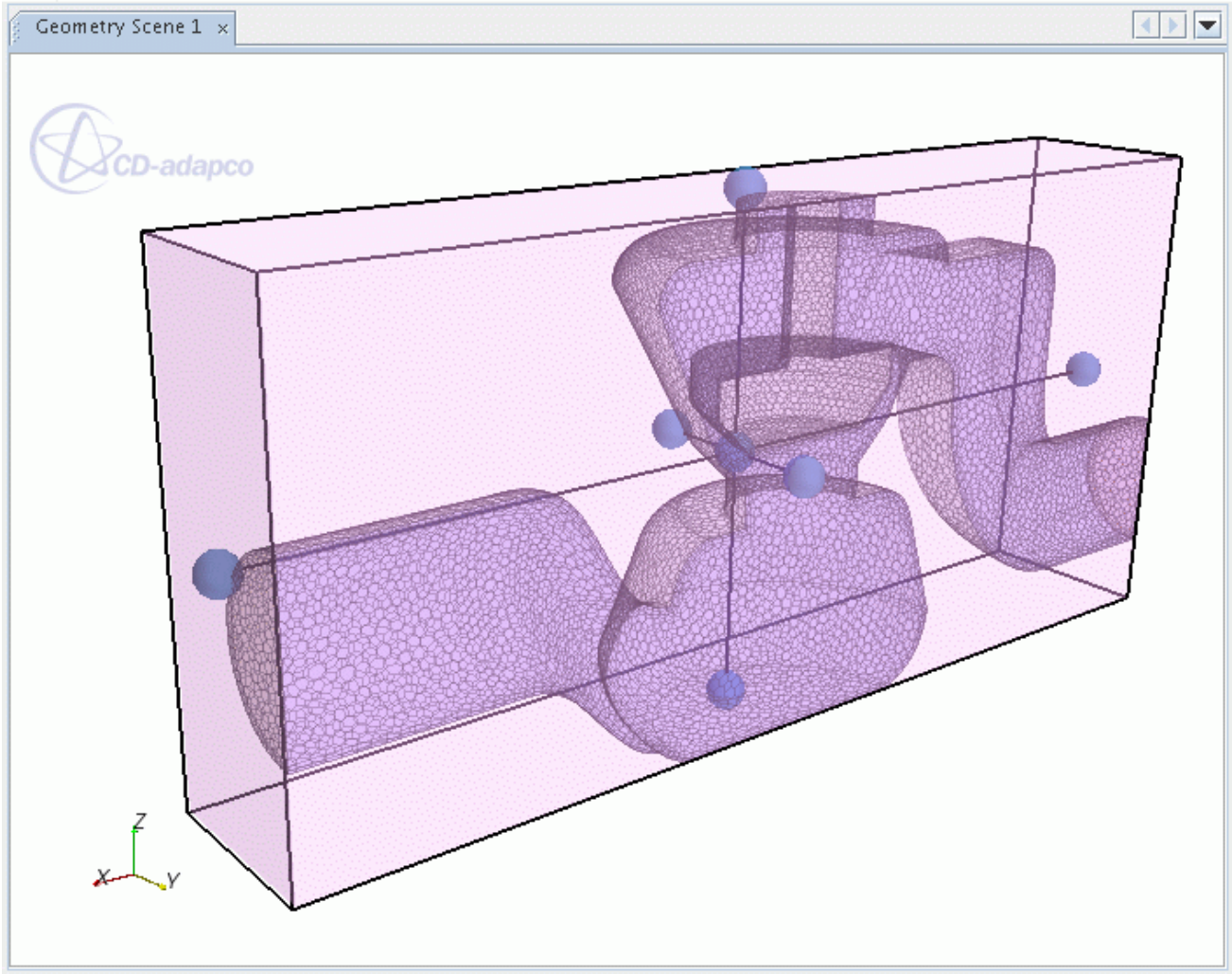
To access the geometric range options, right-click on the **Cell Sets** node and select **New** > **Geometric Range...** from the pop-up menu:



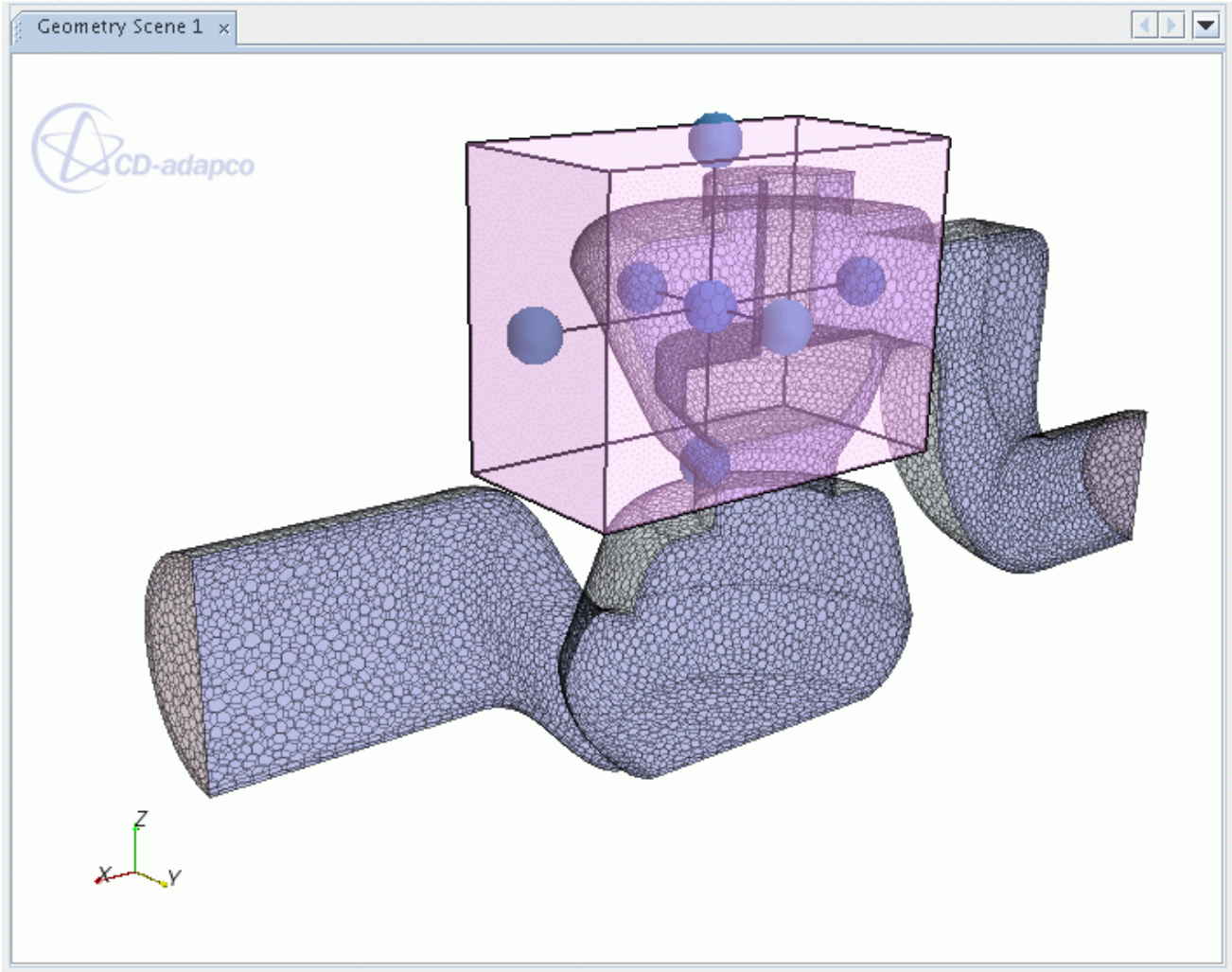
The *Cell Set Geometric Range* dialog appears:



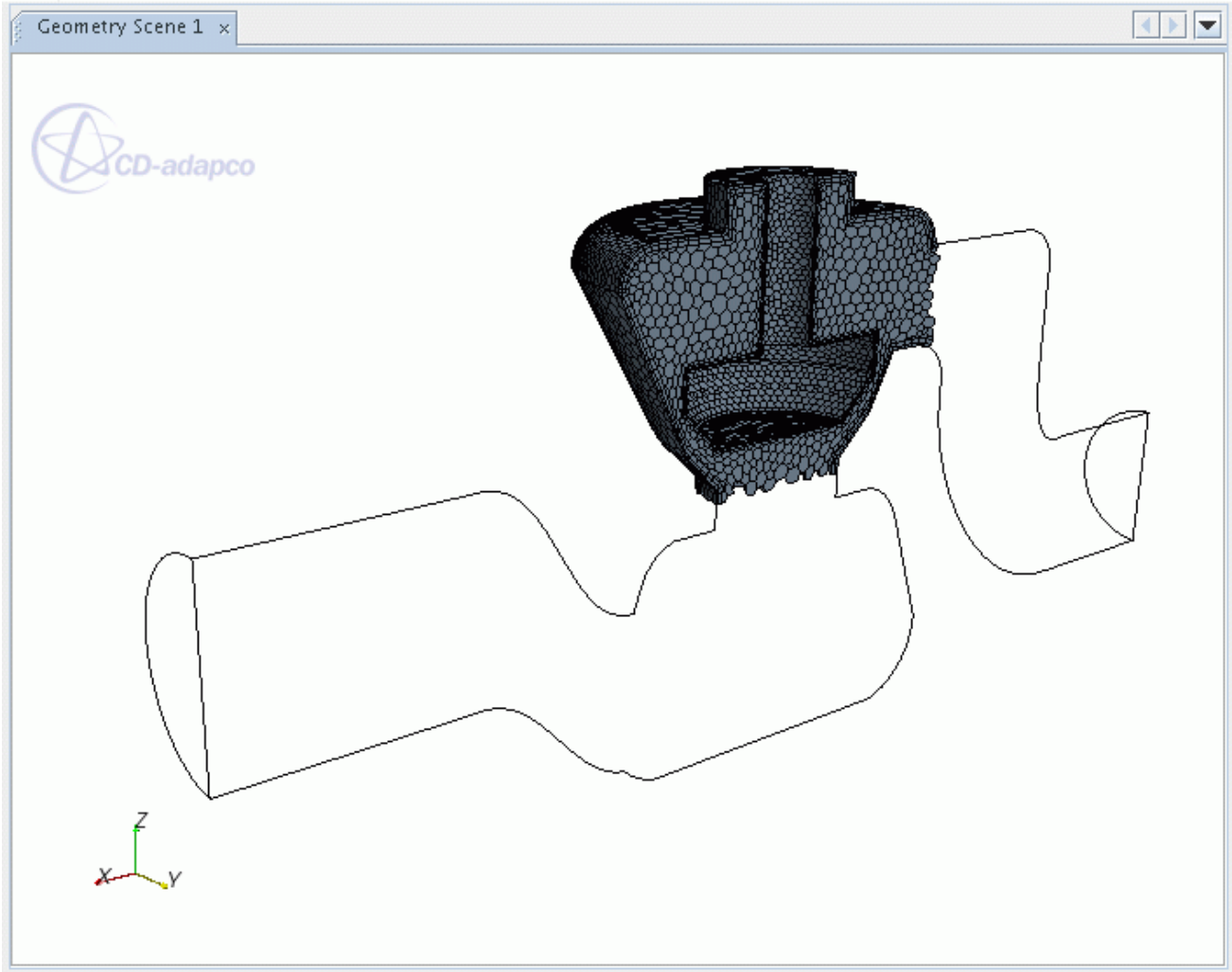
The dialog can be used to enter the maximum and minimum coordinate values in each of the three cartesian directions. Alternatively, the graphical display tool, which is automatically displayed in the scene with the bounding box extending to the model maximum and minimum coordinate directions, can be used to specify the geometric range:



Click and drag on the spherical balls to extend or reduce the box size in each direction. Right-clicking and dragging vertically on the box allow you to resize all the coordinates of the box simultaneously. Clicking using the middle mouse button on the box allows it to be translated in each direction. An example showing a reduced box size in the vicinity of the valve region of the model is shown below:



Then select the *Display* option on the dialog to determine how the new cell set part is added to the current scene. Click **Create** to create the cell set based on the supplied information. The new cell set is then displayed depending on the *Display* options selected:

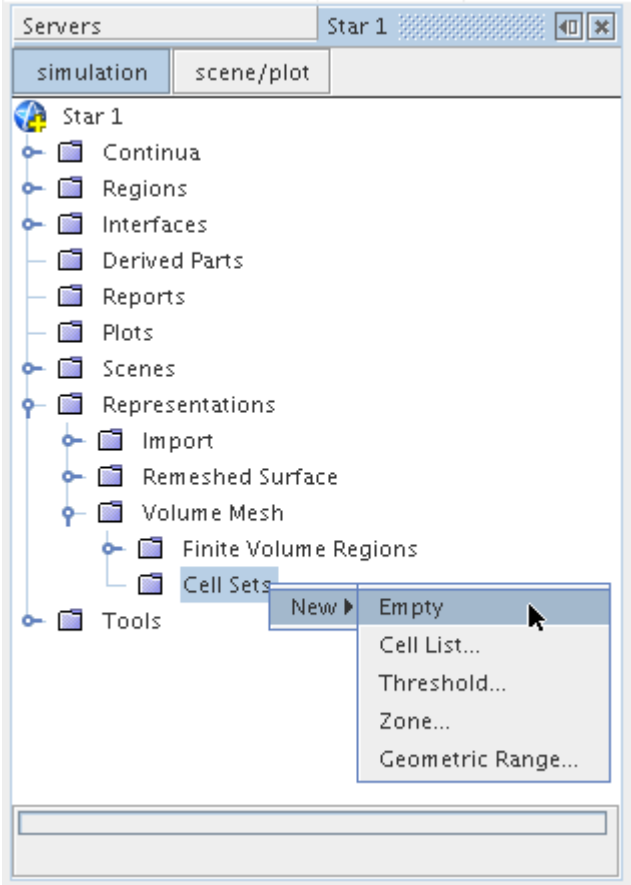


Additional cell sets can be created by repeating the above operation. Press the **Close** button when you are done.

Creating a New Empty Cell Set

The advantage of using an empty cell set is that you are initially not presented with any options for how the set is created or displayed as part of the scene. You can then modify the cell set using the range of global and Boolean tools available.

An empty cell set node, that is, one containing no cells, can be created by right-clicking the **Cell Sets** node and selecting **New** > **Empty** from the pop-up menu:



Example: Creating a Cell Set to View the Outer Layer of Cells on a Mesh

This example describes how to create and modify a cell set using some of the cell set tools that are available.

You can create an empty cell set and customize the cell set using the modification tools that are available. See Modifying an Existing Cell Set. The following steps describe one example workflow that you can use to visualize the outer layer of cells on a part.

Before creating a cell set, make sure that you have a mesh scene open which contains the meshed parts on which you want to view the outer layer of cells. The outer layer of cells can be a prism layer, or the outer layer of volume meshed cells.

1. Create an empty cell set. See Creating a New Empty Cell Set.
2. Right-click the empty cell set node and select **Edit Cell Set in Current Scene**.
3. In the *Edit Cell Set* in-place dialog:
   a) Click **Fill**.
   In the mesh scene, the entire volume mesh appears selected.
   b) Click **Shrink All**.
   The selection shrinks away from all part surfaces by one cell depth. Each additional click of **Shrink All**, shrinks the selection by another cell depth from all part surfaces.
   c) Click **Invert**.

The cells that were selected become deselected and the outer layer of cells that were deselected become selected instead.

d) Click **Close**.

The Properties of the **[Cell Set 1]** node display the number of cells in the cell set that represent the outer layer of cells.

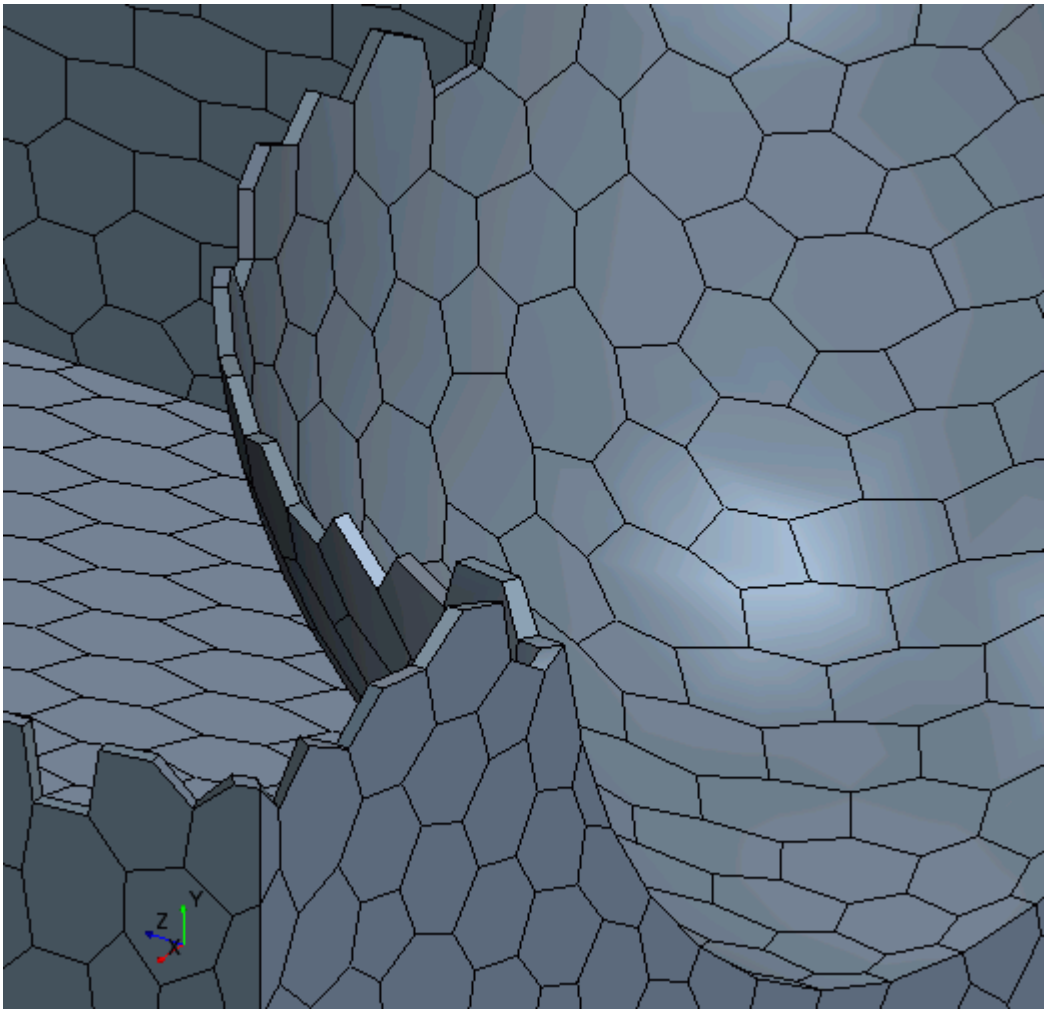
To visualize a section of the cell set, you can create a subset of the cell set (see Using the Subset Option).

For example, to create a geometrical section of the cell set:

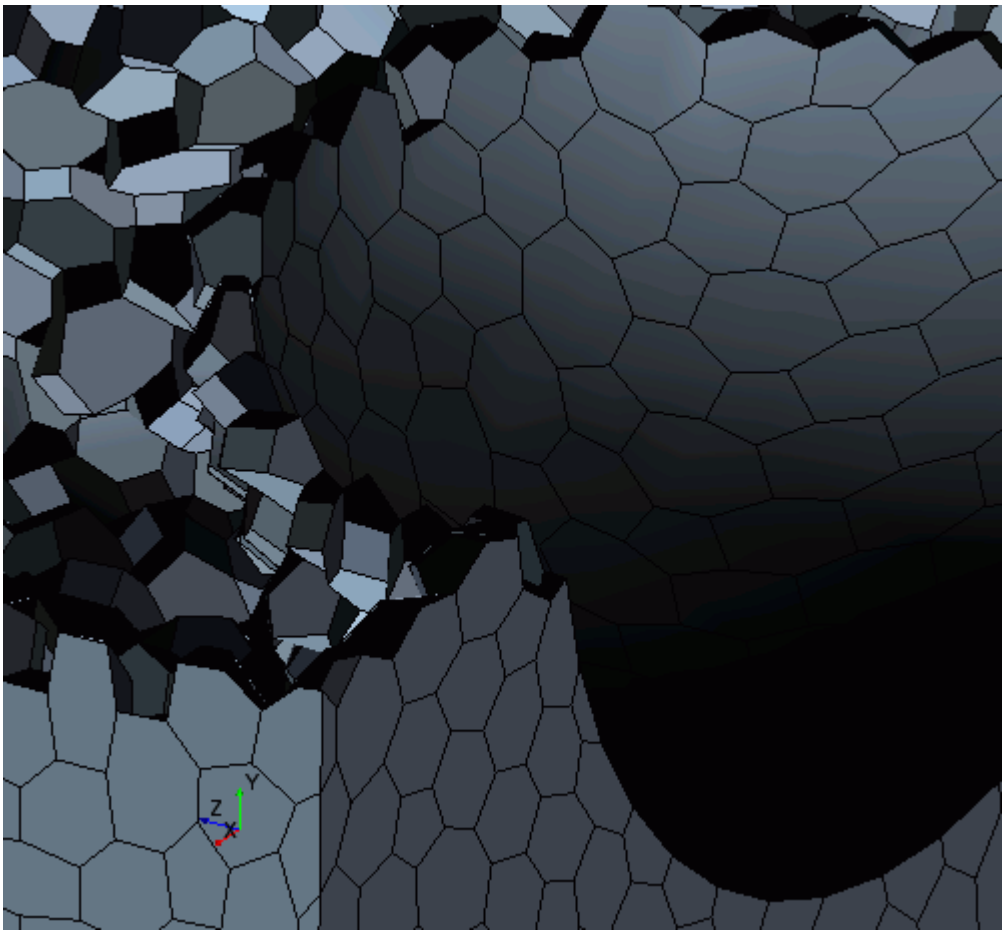
4. Right-click the **[Cell Set 1]** node and select **Subset** > **Geometric Range**.
5. Define the geometric range in the scene, or specify the coordinates in the *Cell Set Geometric Range* dialog.
6. Click **Create**, then **Close**.

   The mesh scene updates to show the subset of the cell set, and the *Properties* of the **[Cell Set 1]** node update to show the number of cells in the subset.

In this example, an outer layer of prism layer cells that is one cell thick is displayed:



In this example, an outer layer of polyhedral cells that is one cell thick is displayed:



Modifying an Existing Cell Set

Once a cell set has been created, it can be easily modified using a range of global and/or Boolean operations in order to manipulate the cell collection further.

In general, global actions are those which allow a single operation on the cell set resulting in collecting all the cells, removing all the cells or growing or shrinking the cell set in some way. Boolean operations allow the existing cell set to be modified using the same modes used for the creation of the original cell set, namely based on a cell list, zone or threshold.

The global actions that you can perform on a cell set are:

- fill - Replace the existing cell set with all the cells in the model
- clear - Remove all cells from the cell set
- invert - Invert the cell set to give all remaining cells
- grow - Grow the existing cell set by one layer based on exposed faces
- grow across interfaces - Grow cells across an interface

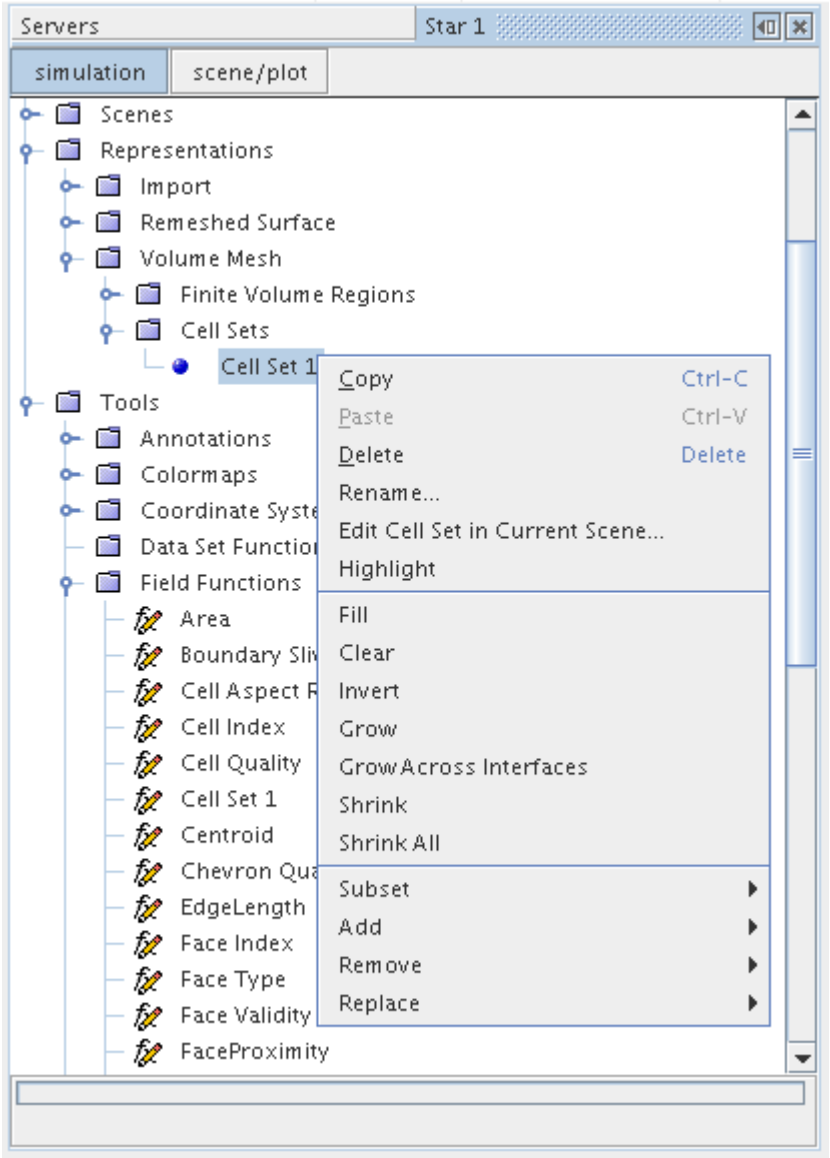
- shrink - Shrink the existing cell set by one layer based on internal faces
- shrink All - Shrink the existing cell set by one layer based on all faces

The Boolean operations that are allowed are:

- subset - Create a set based on the existing set only
- add - Add to the existing set based on all the cells in the model
- remove - Remove cells from the existing set
- replace - Create a set based on all the cells in the model

Boolean operations between different cell sets are also possible using the copy and paste options.

To access the global and Boolean operations, right-click on the cell set node as shown below:



You can also edit the cell set in the current scene, which provides all the above operations in one single in-place dialog. The advantage of this method is that multiple operations can be performed on the cell set without the dialog being closed each time.

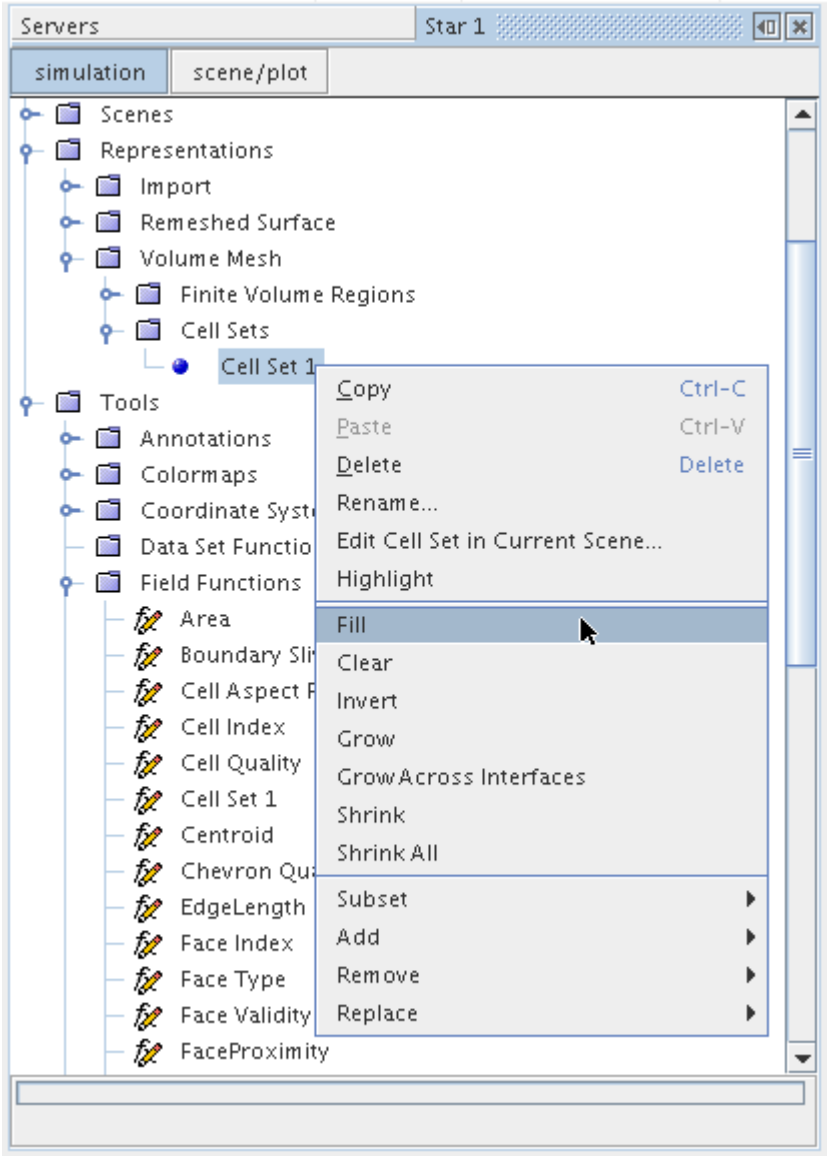
Contents:

Using the Fill Option
Using the Clear Option
Using the Invert Option
Using the Grow Option
Using the Grow Across Interfaces Option
Using the Shrink Option
Using the Shrink All Option
Using the Subset Option
Using the Add Option
Using the Remove Option
Using the Replace Option
Editing a Cell Set in the Current Scene

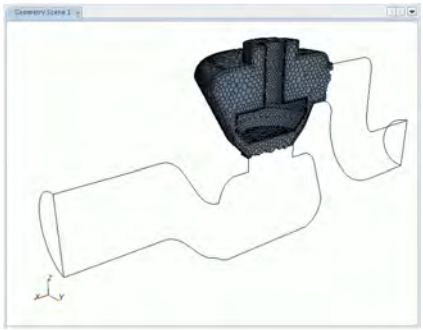
Using the Fill Option

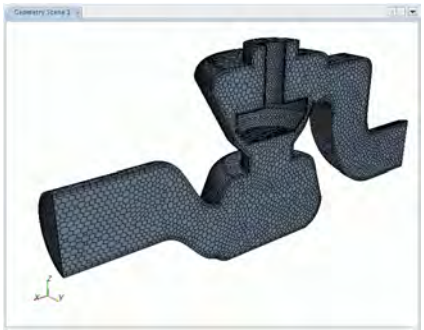
The fill option replaces a previous collection of cells with all the cells that currently exist in the model.

To access the cell set fill option, right-click the desired cell set node and select **Fill** from the pop-up menu:



An example before and after snapshot of the fill operation is shown below:



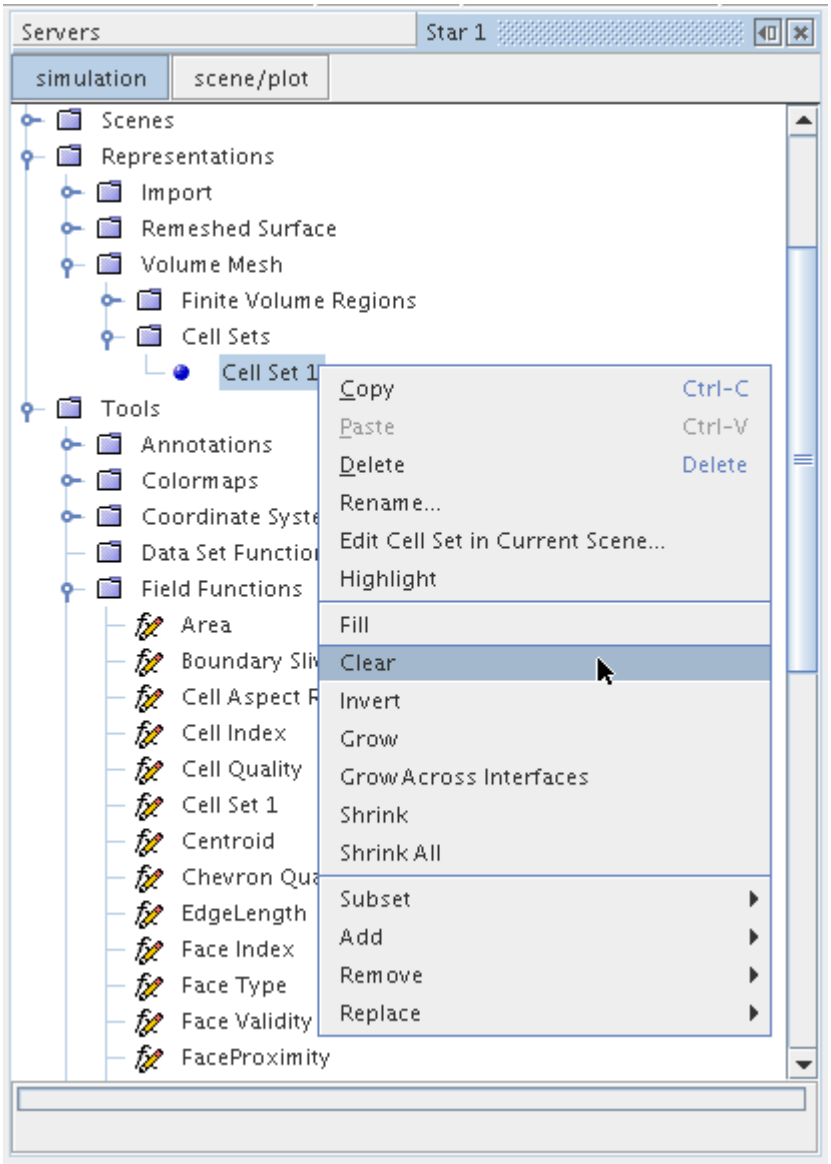


In this example, the original cell set contained a portion of the model cells. After the fill operation, the cell set contains all the model cells.

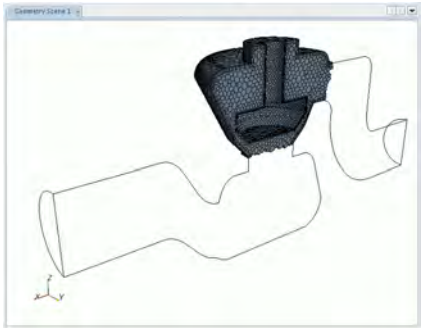
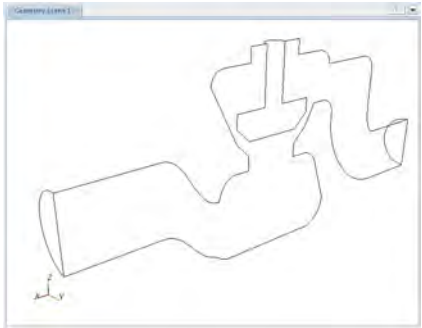
Using the Clear Option

The clear option can be used to clear all the cells from the cell set. The previous collection of cells is removed leaving the cell set empty.

To access the cell set clear option, right-click the desired cell set node and select **Clear** from the pop-up menu:



An example before and after snapshot of the clear operation is shown below:

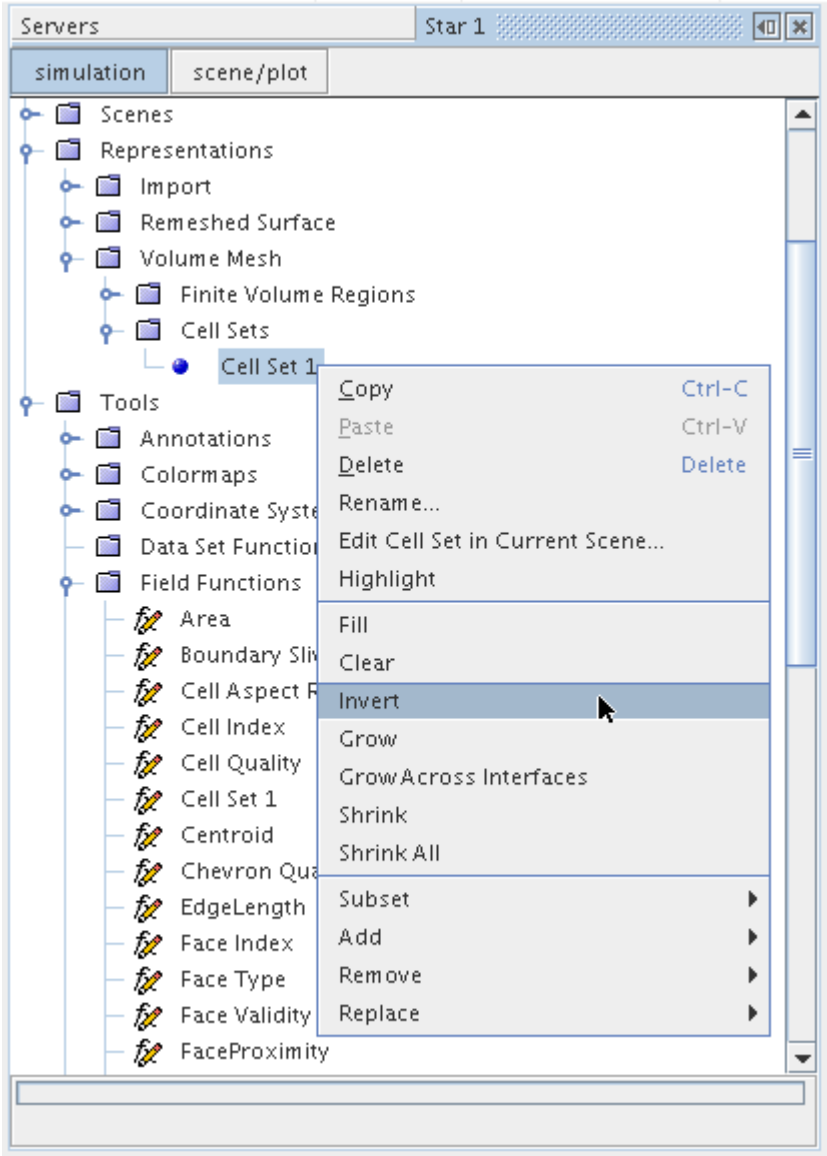
		

In this example, the original cell set contained a portion of the model cells. After the clear operation, the cell set contains no cells.

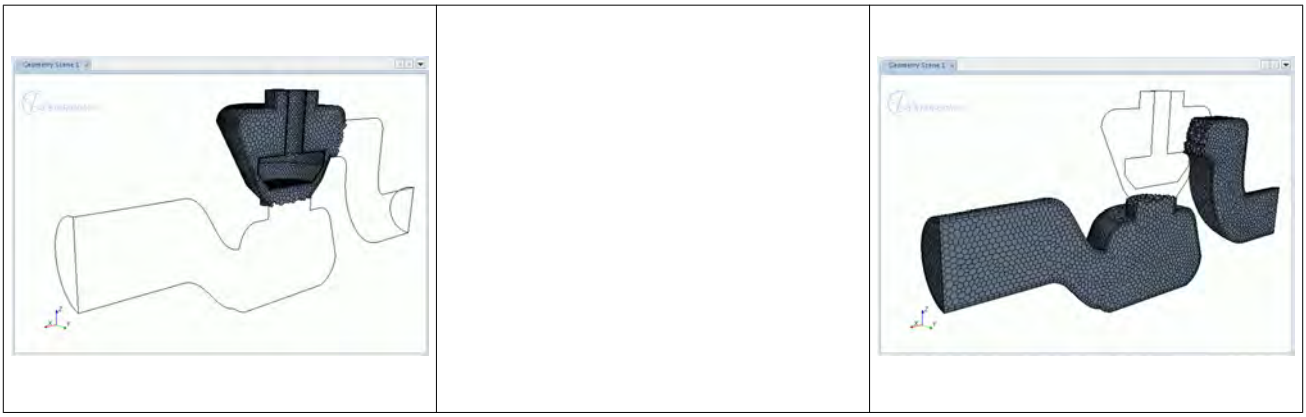
Using the Invert Option

The invert option deletes the existing cells from the set and replaces them with all other cells in the model, that is, those cells that were previously outside the set.

To access the cell set invert option, right-click the desired cell set node and select **Invert** from the pop-up menu:



An example before and after snapshot of the invert operation is shown below:

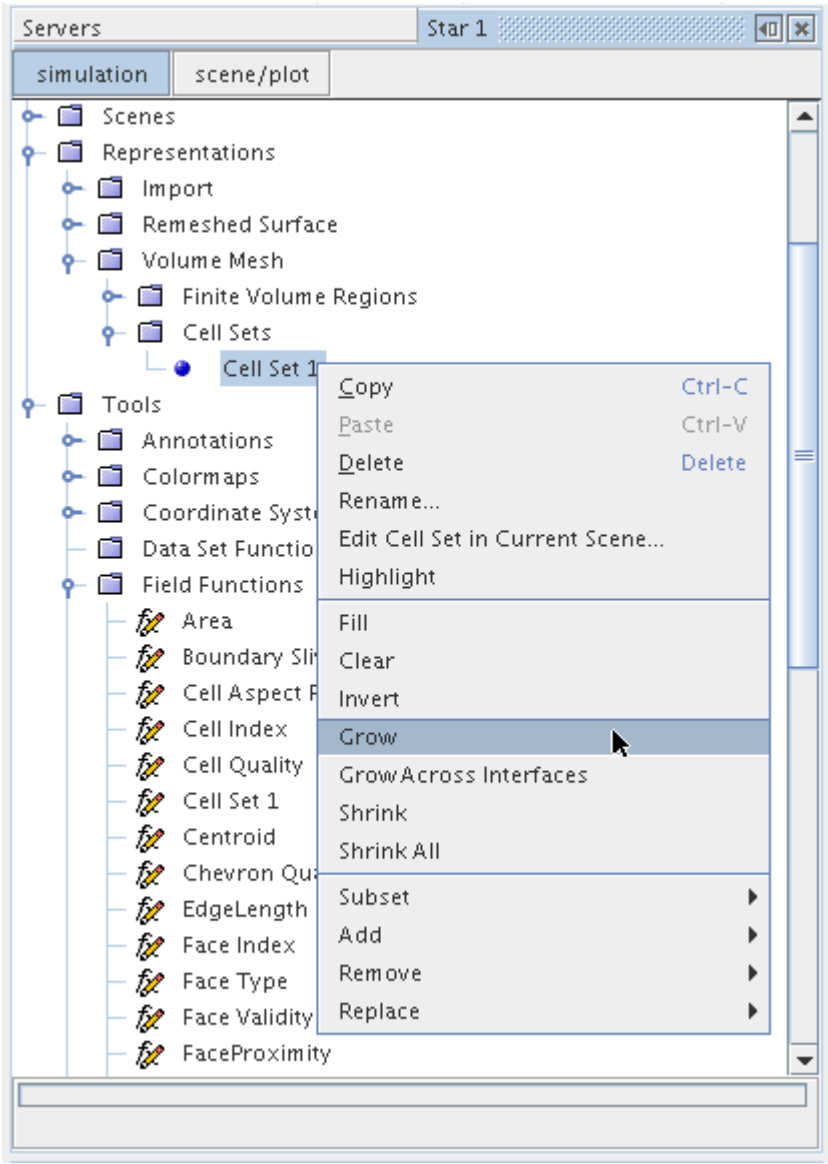


In this example, the original cell set contained a portion of the model cells. After the invert operation, the cell set contains all remaining cells.

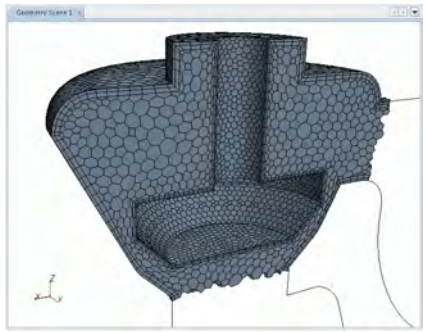
Using the Grow Option

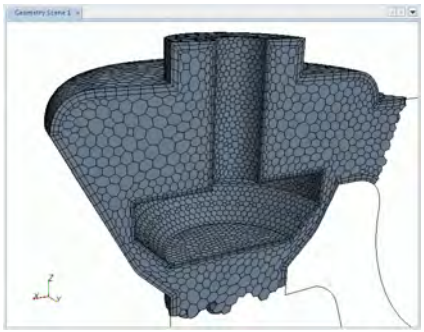
The grow option can be used to expand the current cell set by a single layer of cells in all directions.

To access the cell set grow option, right-click on the desired cell set node and select **Grow** from the pop-up menu:



An example before and after snapshot of the grow operation is shown below:





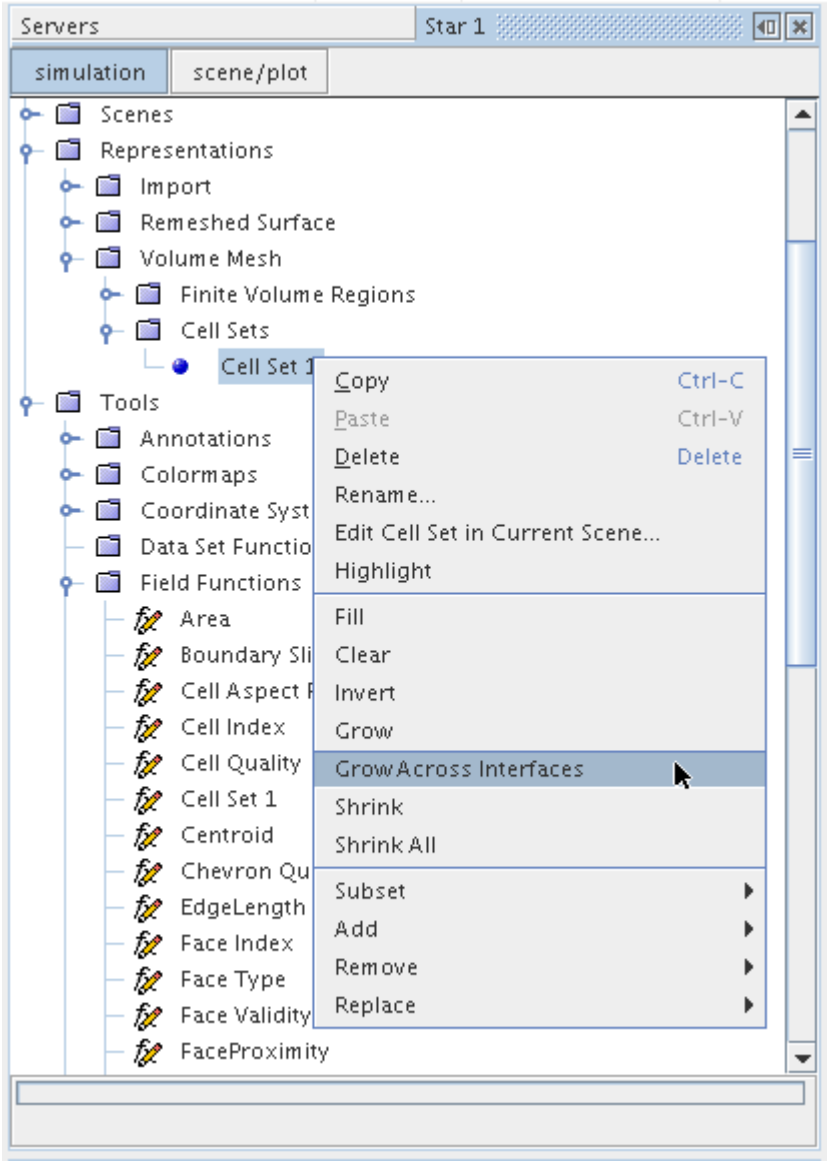
In this example, the original cell set contained a portion of the model cells. After the grow operation, the cell set has expanded to include the nearest cell neighbors based on the faces of the original cell set.

Using the Grow Across Interfaces Option

The `Grow Across Interfaces` option can be used with the grow option to expand the current cell set by one layer across an interface.

A single layer of cells is grown from the interface faces of the existing cell set cells and added to the set from another region or from the opposite side of a baffle. Before this operation can be used however, the simulation must be initialized.

To access the cell set grow across interfaces option, right-click on the desired cell set node and select **Grow Across Interfaces** from the pop-up menu:

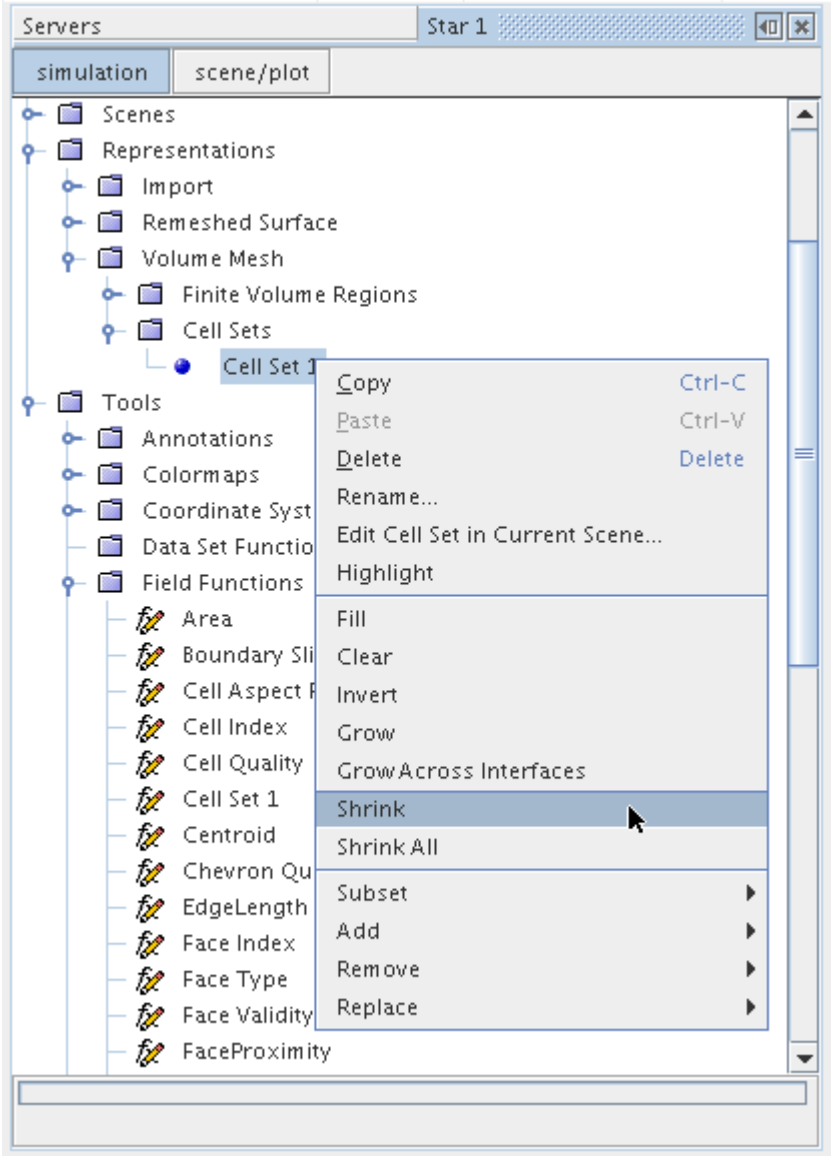


Wherever a cell face in the cell set belongs to an interface then the opposite cell connected to that face is grown. Thereafter, the normal grow option can be used to continue growing the cells.

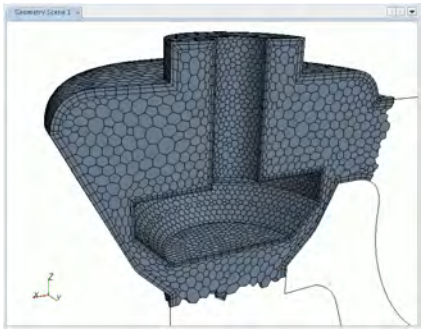
Using the Shrink Option

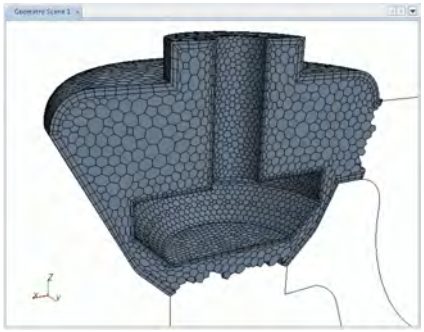
The shrink option can be used to reduce the current cell set by one layer, removing cells that have a face exposed in the interior of the model. Cells with faces that are on a boundary are ignored.

To access the cell set shrink option, right-click the desired cell set node and select **Shrink** from the pop-up menu:



An example before and after snapshot of the shrink operation is shown below:



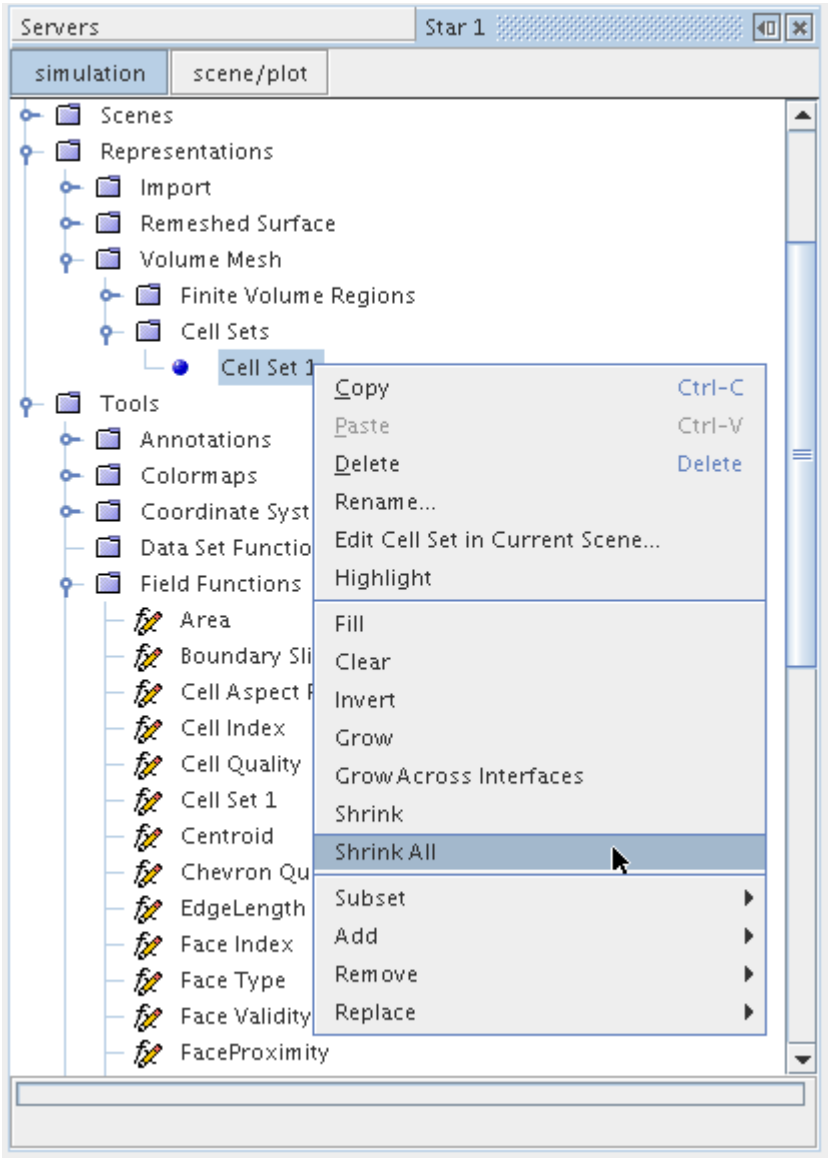


In this example, the original cell set contained a portion of model cells. After the shrink operation, the cell set has been reduced in size by one neighbor layer where the neighbor cells have a face exposed in the model interior.

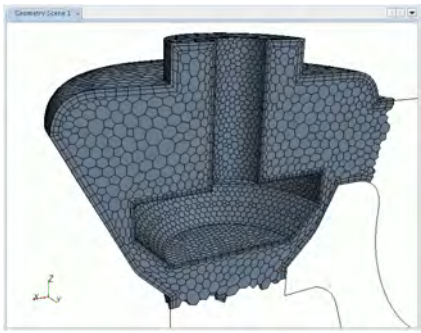
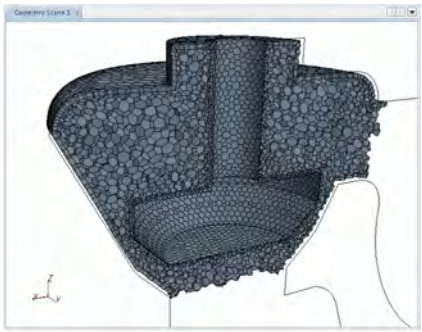
Using the Shrink All Option

The shrink all option can be used to reduce the current cell set by one layer, removing cells that have a face exposed in the interior of the model or on a boundary.

To access the cell set shrink all option, right-click on the desired cell set node and select **Shrink All** from the pop-up menu:



An example before and after snapshot of the shrink all operation is shown below:

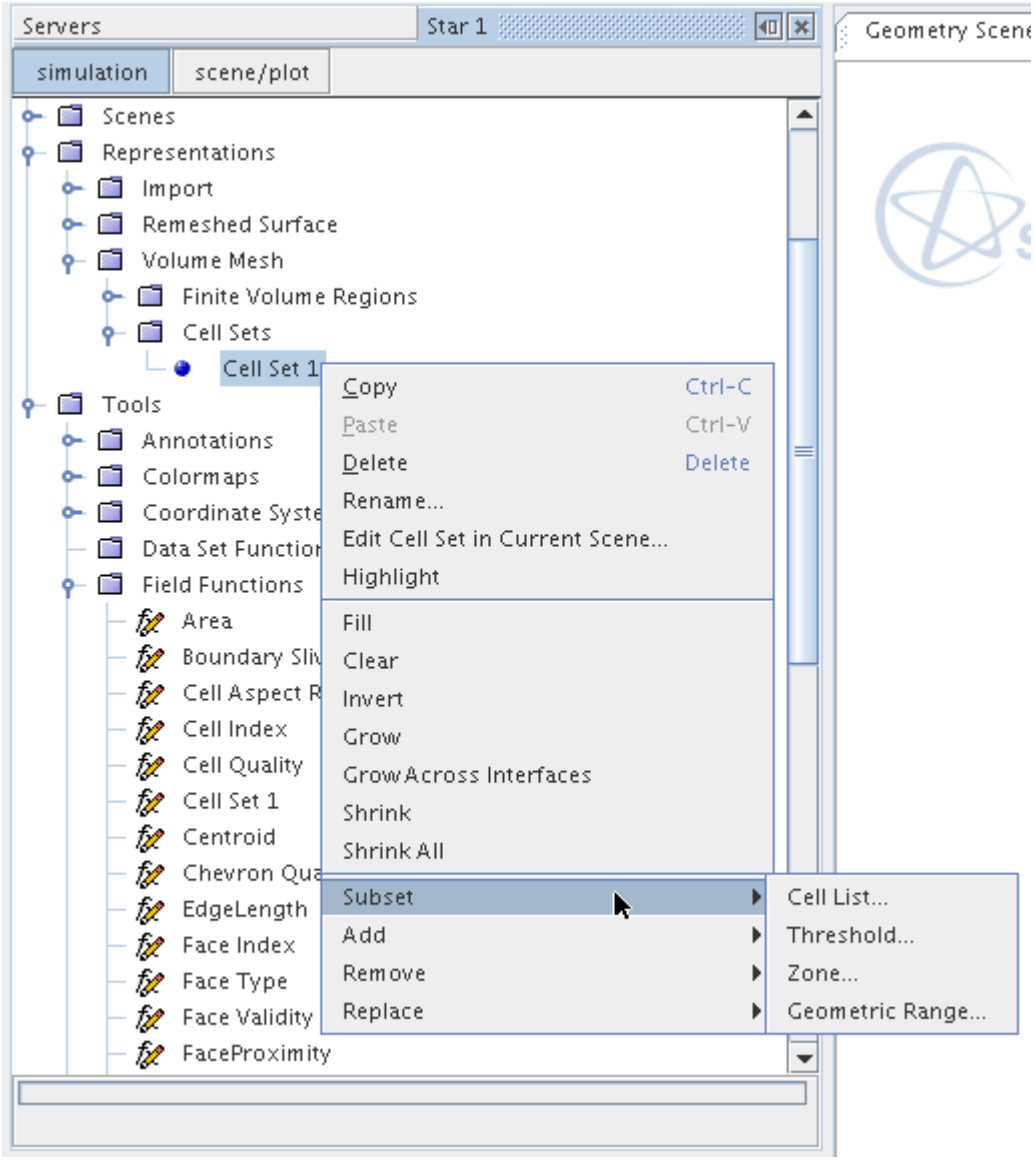
		

In this example, the original cell set contained a portion of the model cells. After the shrink all operation, the cell set has been reduced in size by one neighbor layer where the neighbor cells have a face exposed on the boundary or the model interior.

Using the Subset Option

The subset option allows you to create a set of cells based on the existing cell set only, rather than the entire model. Cells that are not contained in the original cell set are ignored.

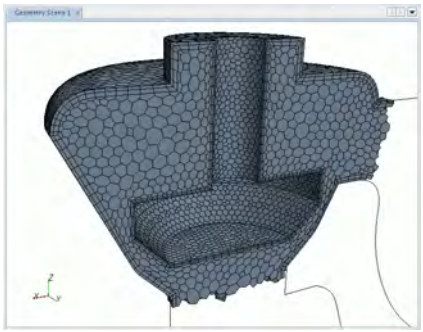
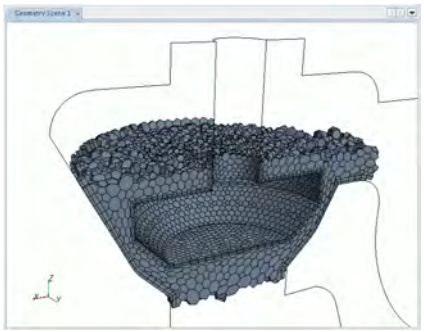
To access the cell set subset options, right-click the desired cell set node and select **Subset** from the pop-up menu:



Four different subset options are allowed:

- cell list - Create a subset based on cell index ID's or offsets.
- threshold - Create a subset based on a scalar threshold.
- zone - Create a subset based on a zone.
- geometric range - Create a subset based on a geometric range.

An example before and after snapshot of a subset operation using a zone is shown below:

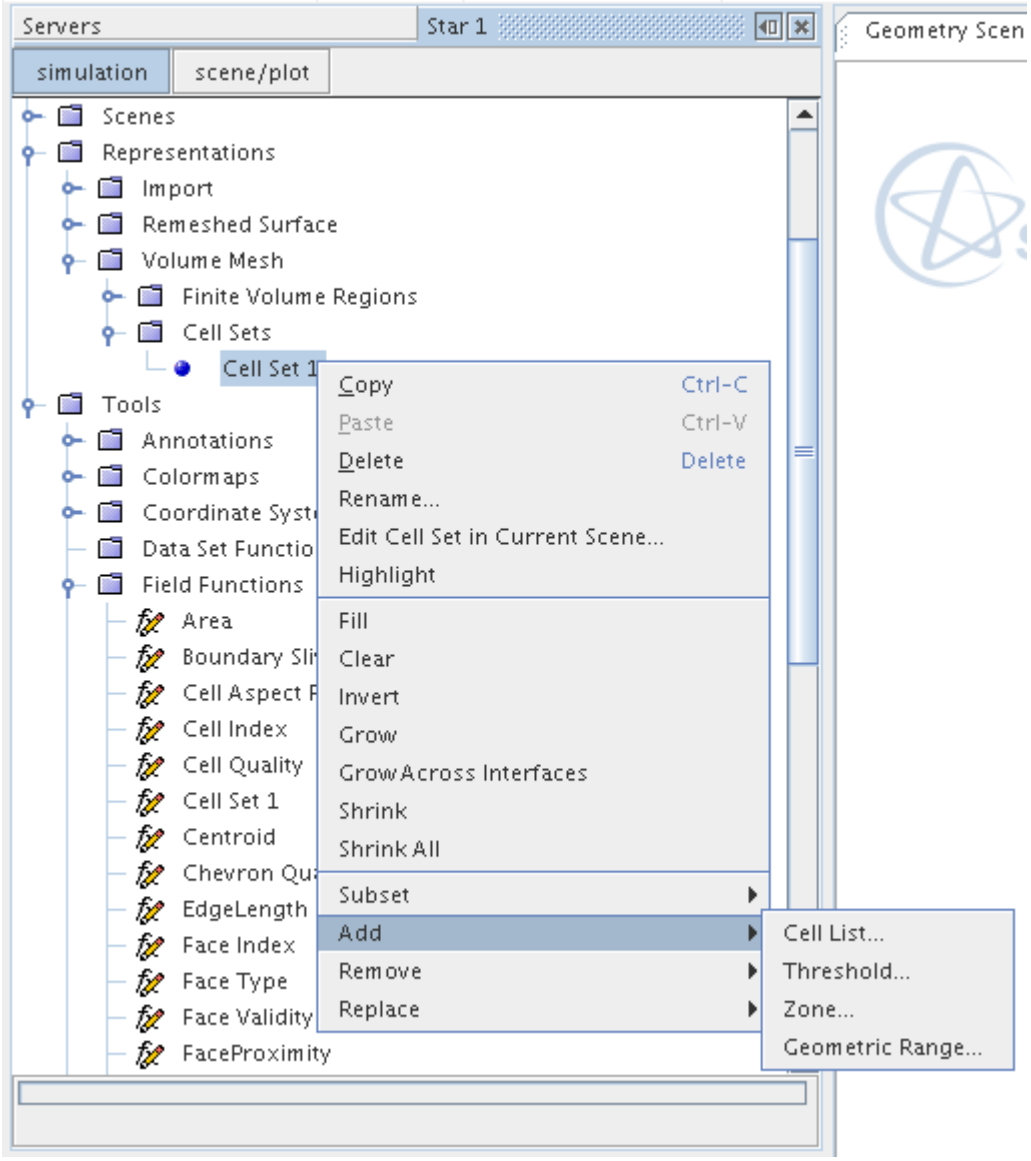
		

In this example, the original cell set contained a portion of the model cells. The subset operation using a zone was used to select the lower half of the set. Any cells in the model not contained in the original cell set were ignored in the new cell set.

Using the Add Option

The add option allows you to add cells to the existing cell set based on all the cells in the model.

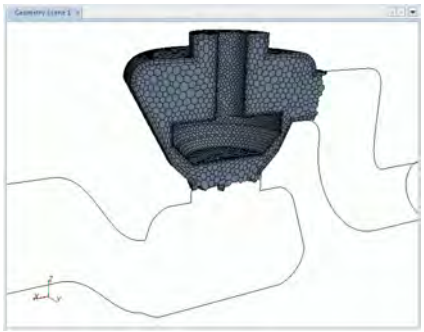
To access the cell set add option, right-click the desired cell set node and select **Add** from the pop-up menu:



Four different add options are allowed:

- cell list - Add to the set based on cell index ID's or offsets.
- threshold - Add to the set based on a scalar threshold.
- zone - Add to the set based on a zone.
- geometric range - Add to the set based on a geometric range.

An example before and after snapshot of an add operation using a zone is shown below:

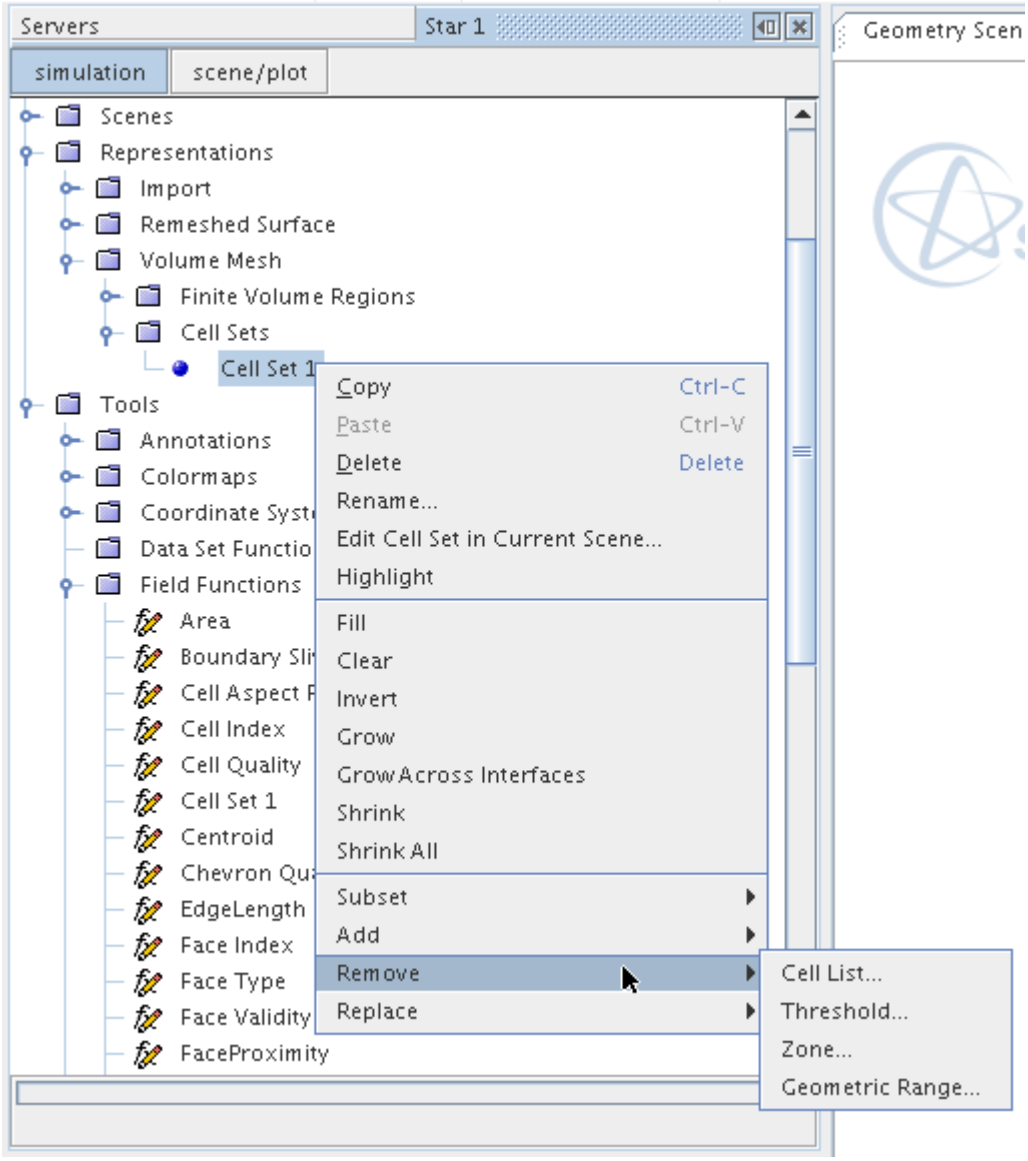


In this example, the original cell set contained a portion of the model cells. The add operation using a zone was used to select an additional area below the original to add to the set.

Using the Remove Option

The remove option allows you to remove cells from the existing cell set.

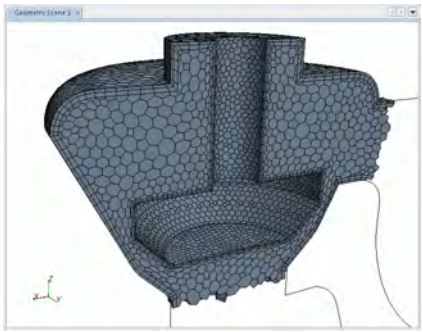
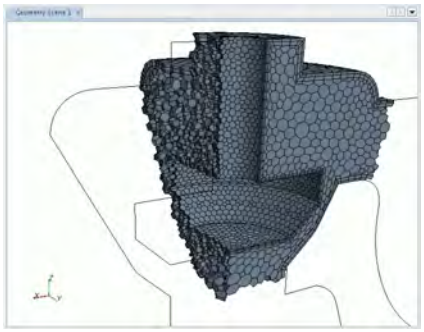
To access the cell set remove option, right-click on the desired cell set node and select **Remove** from the pop-up menu:



Four different remove options are allowed:

- cell list - Remove cells from the set based on cell index ID's or offsets.
- threshold - Remove cells from the set based on a scalar threshold.
- zone - Remove cells from the set based on a zone.
- geometric range - Remove cells based on a geometric range.

An example before and after snapshot of a remove operation using a zone is shown below:

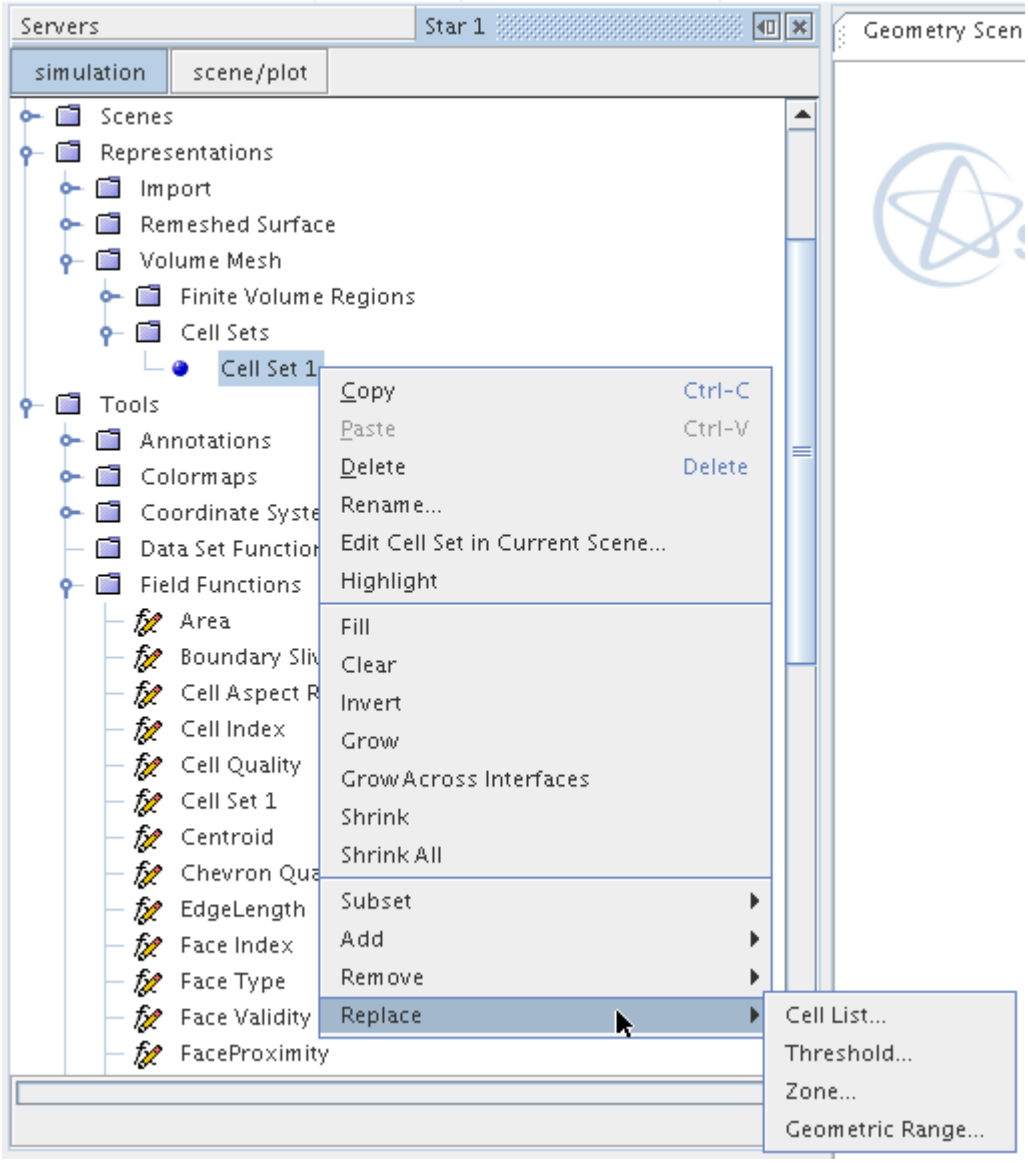
		

In this example, the original cell set contained a portion of the model cells. The remove operation using a zone was used to select the left half of the set, which was removed, leaving the right half in the set.

Using the Replace Option

The replace option allows you to replace the existing cell set with a new one based on the cells in the entire model. Cells that are contained in the original set can be reselected if desired.

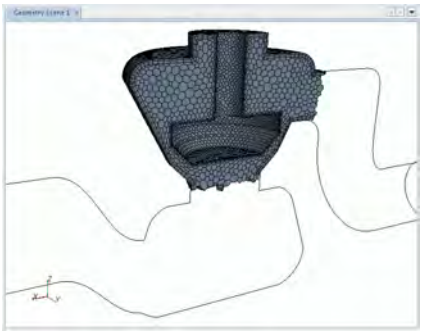
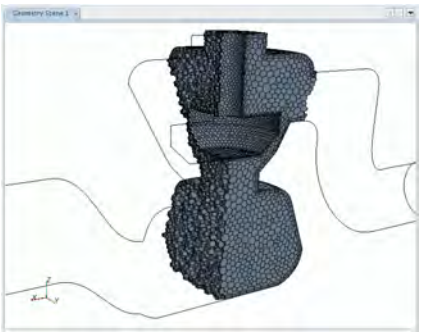
To access the cell set replace option, right-click on the desired cell set node and select **Replace** from the pop-up menu:



Four different replace options are allowed:

- cell list - Replace cells based on cell index ID's or offsets.
- threshold - Replace cells based on a scalar threshold.
- zone - Replace cells based on a zone.
- geometric range - Replaced cells based on a geometric range.

An example before and after snapshot of a replace operation using a zone is shown below:

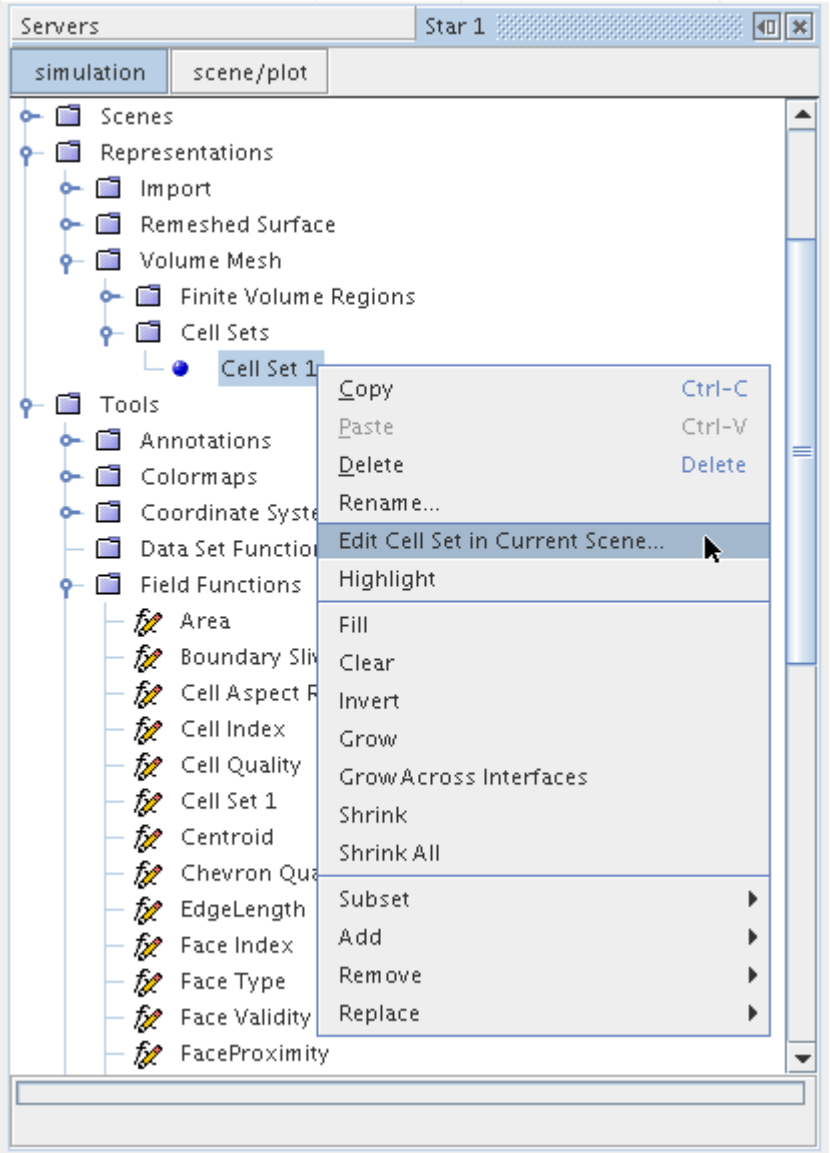
		

In this example, the original cell set contained a portion of the model cells. The replace operation using a zone was used to select a new set containing some cells below the original set, as well as the right half portion of the original set.

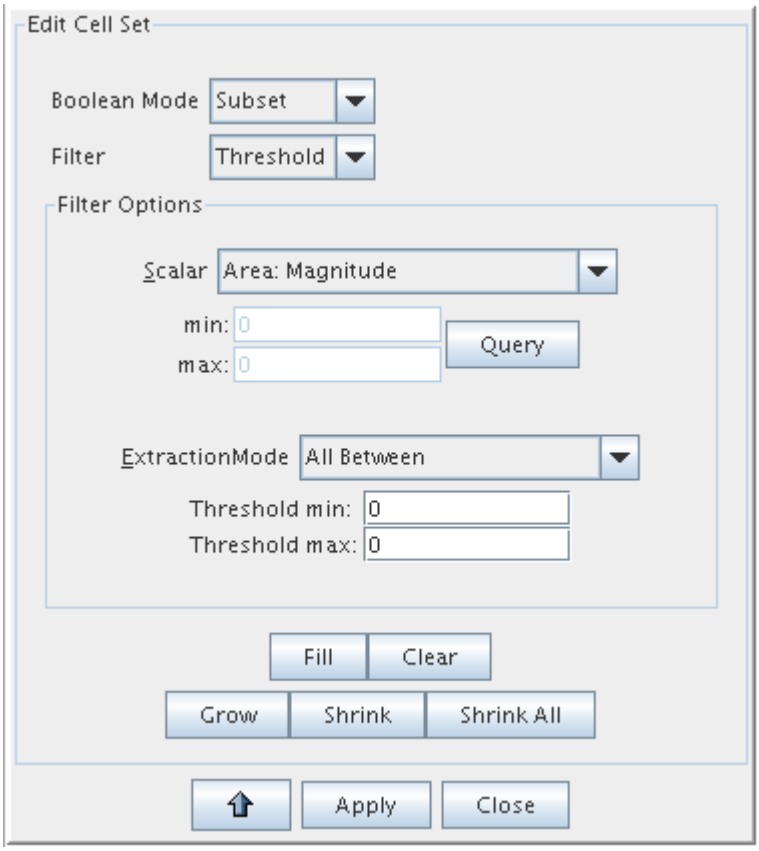
Editing a Cell Set in the Current Scene

In order to allow multiple set operations to be performed without the dialog closing each time, a cell set can be edited using an in-place dialog as part of the current scene. The same global and Boolean operations as described above are allowed.

To edit the cell set in the current scene, right-click the cell set node and select **Edit Cell Set in Current Scene...** from the pop-up window:



The in-place *Edit Cell Set* dialog opens:



To perform Boolean operations, such as Subset, Add, Remove or Replace, select the appropriate **Boolean Mode** option. The **Filter** option determines whether a Cell List, Threshold or Zone is used. Depending on the latter option, the *Filter Options* groupbox displays the appropriate selections to complete the operation. Press **Apply** to perform the change.

To perform global operations, the Fill, Clear, Grow, Shrink and Shrink All buttons can be pressed.

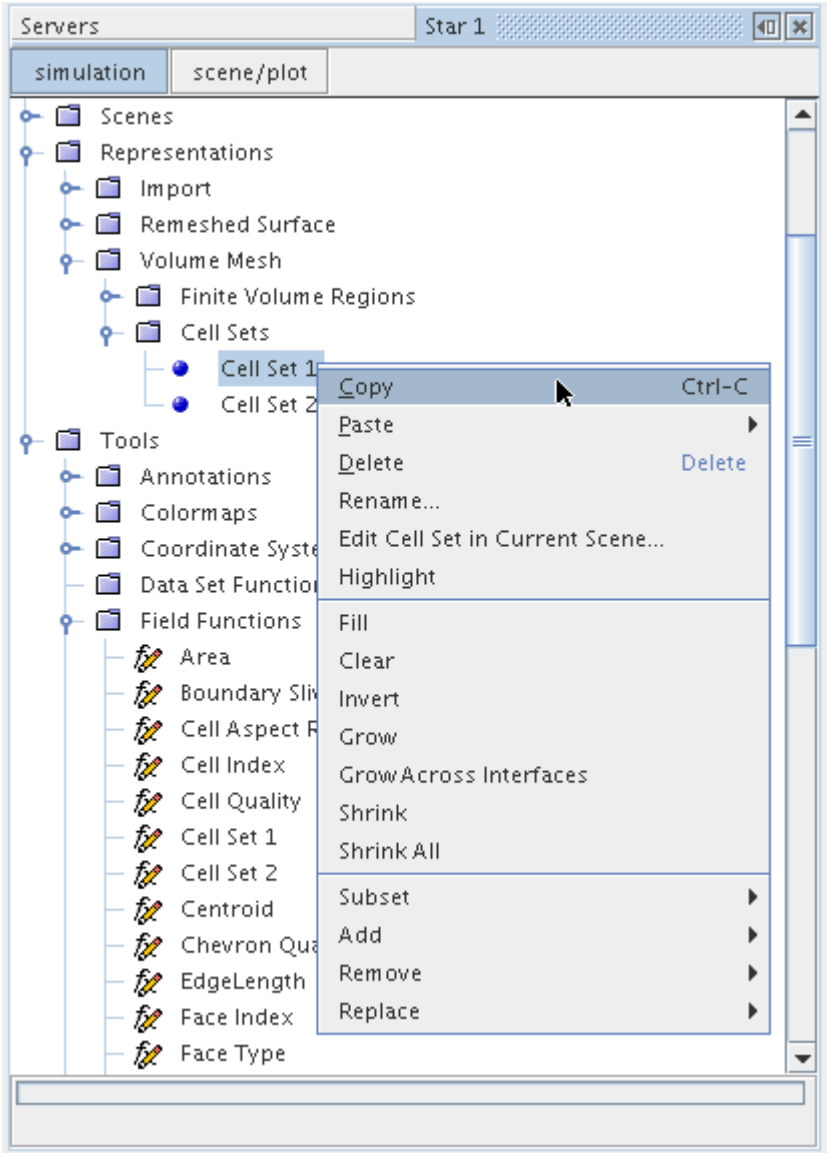
Copying and Pasting Cell Sets

Boolean operations between two cell sets can be performed by copying and pasting from one cell set to another.

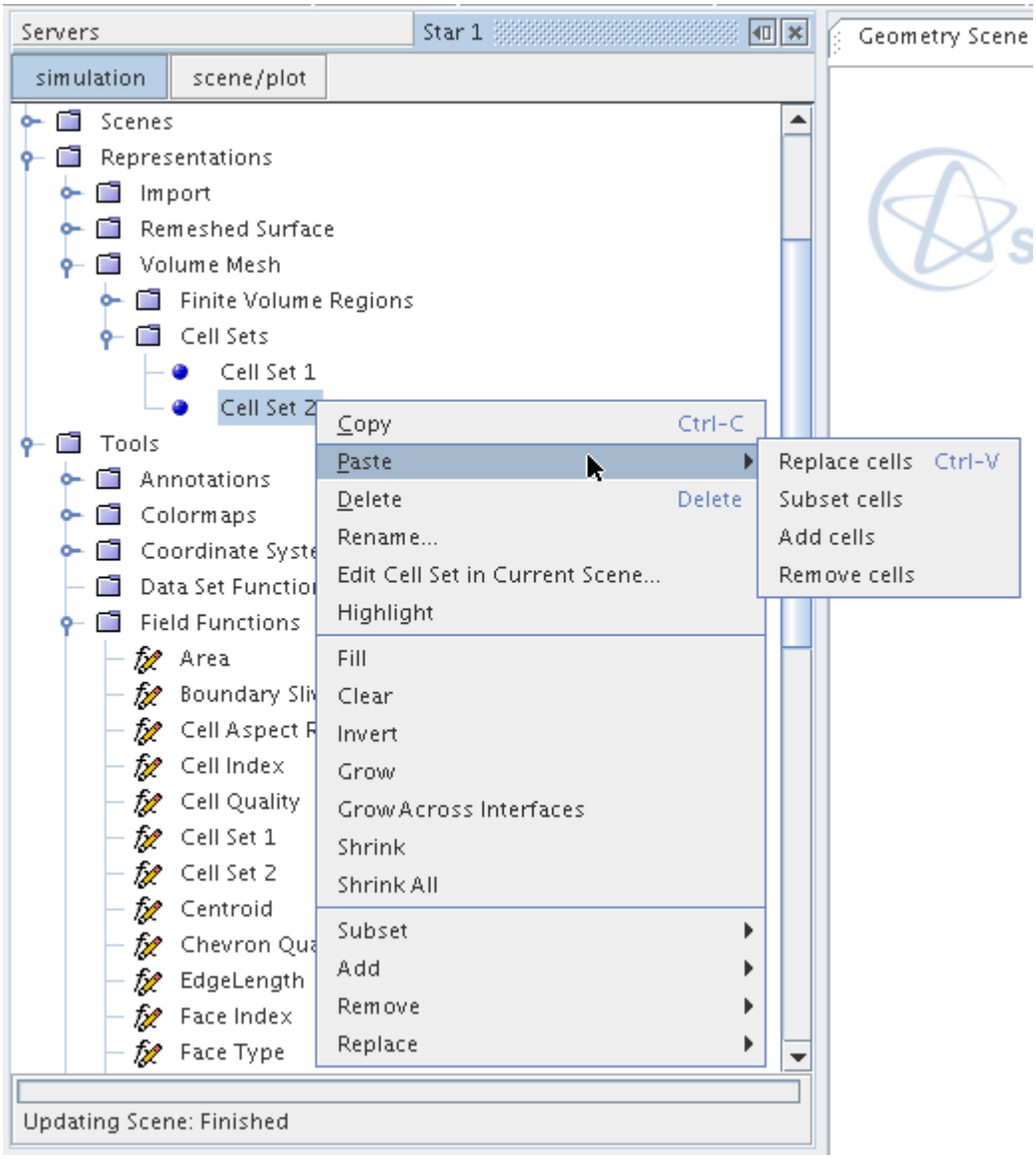
The following options are allowed when you paste a cell set:

- Replace cells - Replace the target cell set with cells from the copied set
- Subset cells - Form a new set of the common cells between the copied cell set and the target one
- Add cells - Add the cells from the copied set to the target set
- Remove cells - Remove common cells in the copied set from the target set

To copy a cell set, right-click on the cell set node and select **Copy**from the pop-up menu, as shown below:



The Boolean operation is performed when pasting to another cell set. Right-click on the target cell set node and select **Paste** from the pop-up menu, along with the desired operation as described above:



The behavior of the Boolean operations between cell sets is the same as previously described for individual sets. However, they provide a powerful manipulation tool to help isolate cells quickly for further analysis.

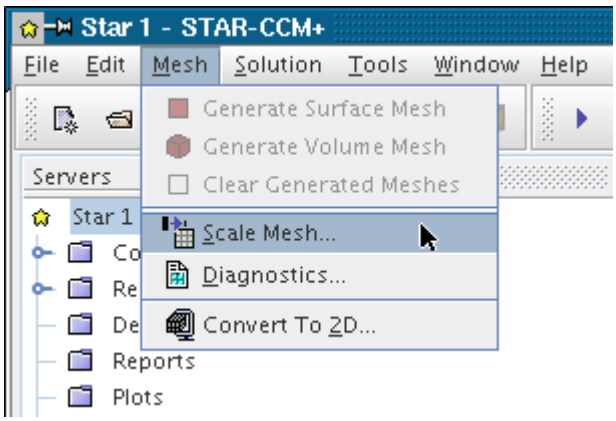
Scaling the Entire Mesh

Scaling the mesh is important as the units of the model must be in meters before running an analysis.

If the surface or volume mesh was imported using different units, then the process below must be followed.

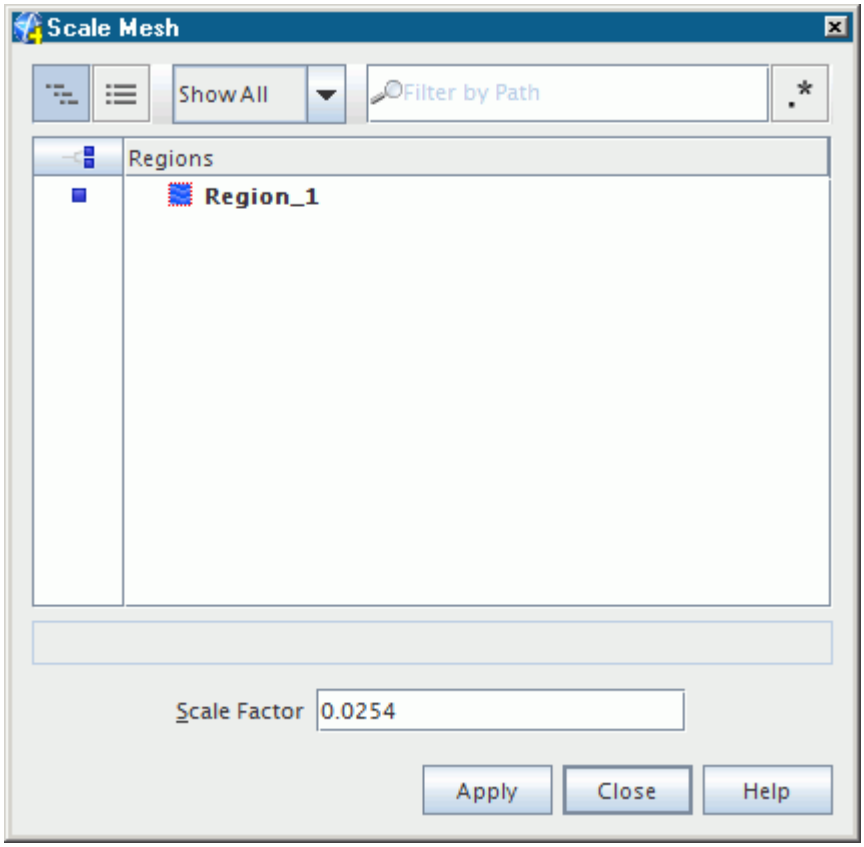
To scale the entire mesh inside STAR-CCM+:

1. Use the **Mesh** > **Scale Mesh...** menu option:



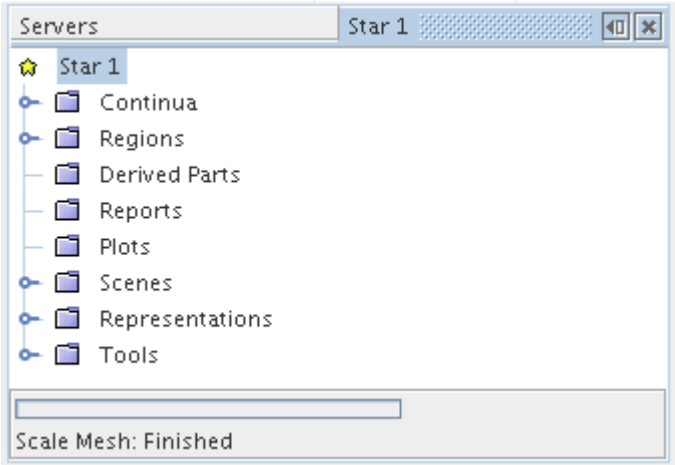
   The *Scale Mesh* dialog opens with all regions selected by default.

2. Enter the scale factor into the **Scale Factor** text box. This value is used to multiply each dimension to scale the model.
   In the example below, the scale factor would convert a model that had been constructed in inches to meters.

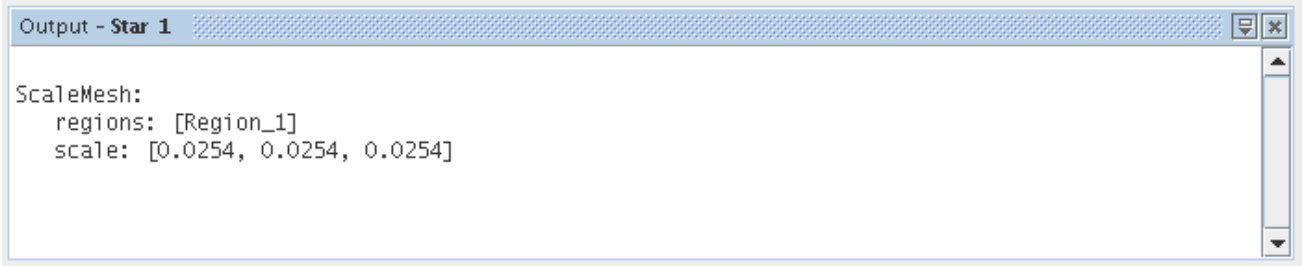


3. Click **Apply**.

   The progress bar at the bottom of the simulation tree view indicates the progress:



Once the operation is complete, the summary information is written to the *Output* window. This output is useful to check that the scale factor has been applied to the appropriate regions:



All active scenes are updated to reflect the new mesh configuration.

Working with Baffles

In STAR-CCM+, there are two methods to accommodate 2D surfaces that are intended for use as thin-walled baffles. The first method is to create interfaces that are based on imported surfaces and so have a true zero-thickness baffle. The second method is to inflate the imported surface to create a baffle of finite physical thickness. Both of these methods are discussed in this section.

Contents:

Generating Baffles with Zero Thickness
Generating Baffles with Finite Thickness

Generating Baffles with Zero Thickness

When correctly defined, zero thickness baffles are represented as an interface within the **Interfaces** manager node.

To define a zero thickness baffle:

1. Unless you intend to create a floating baffle, ensure that the baffle is properly connected to the surrounding geometry. If the baffle is a separate geometry part from the part in which the baffle is contained, combine these parts. See Combining Parts.
2. Create a separate boundary for the baffle and assign the baffle surface to it.
3. Convert the baffle boundary to an interface of the relevant type.
4. To distinguish which side of the baffle belongs to the original baffle boundary, use the offset surface mode of the *Surface Preparation Tool*.

   A section detailing the process for updating baffle geometry is also provided.

For more information, see the following sections.

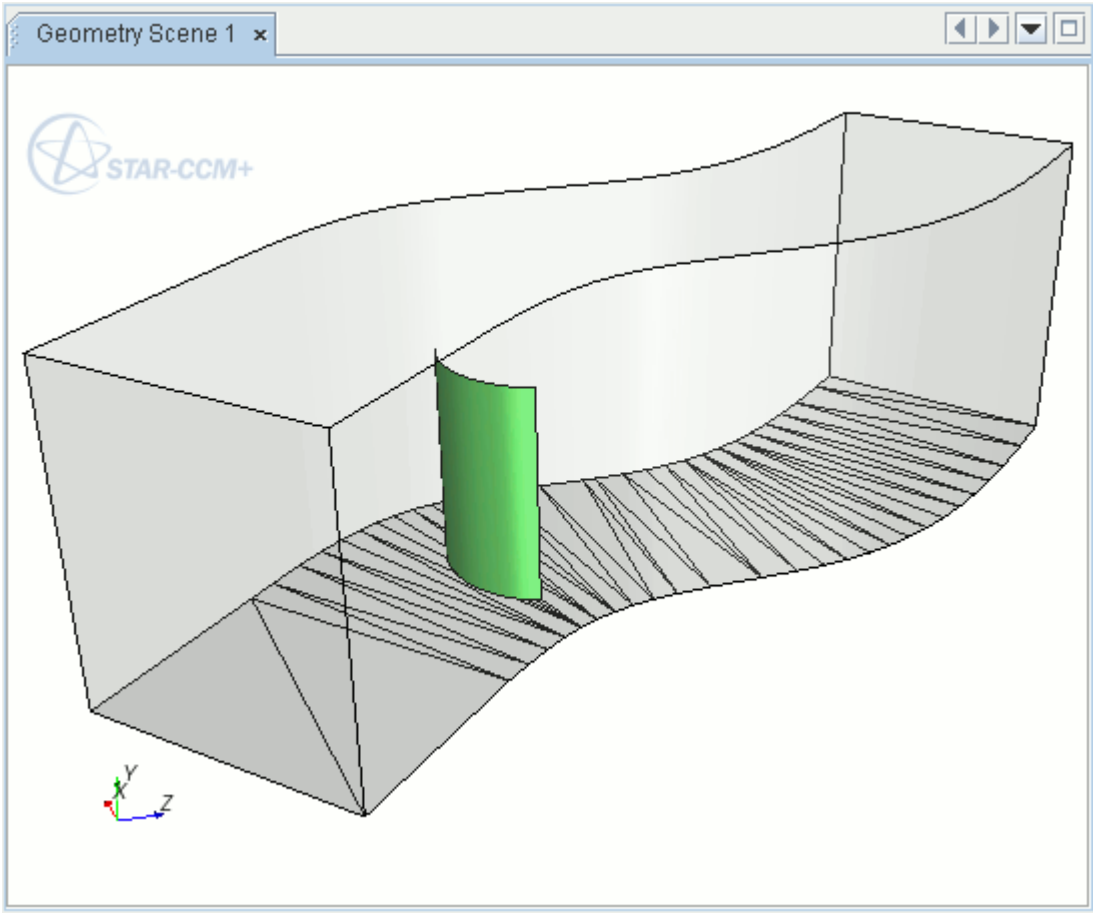
Contents:

Ensuring the baffle is a Connected Surface
Converting a Baffle Boundary to an Interface
Distinguishing Between the Two Sides of a Baffle
Updating Baffle Geometry

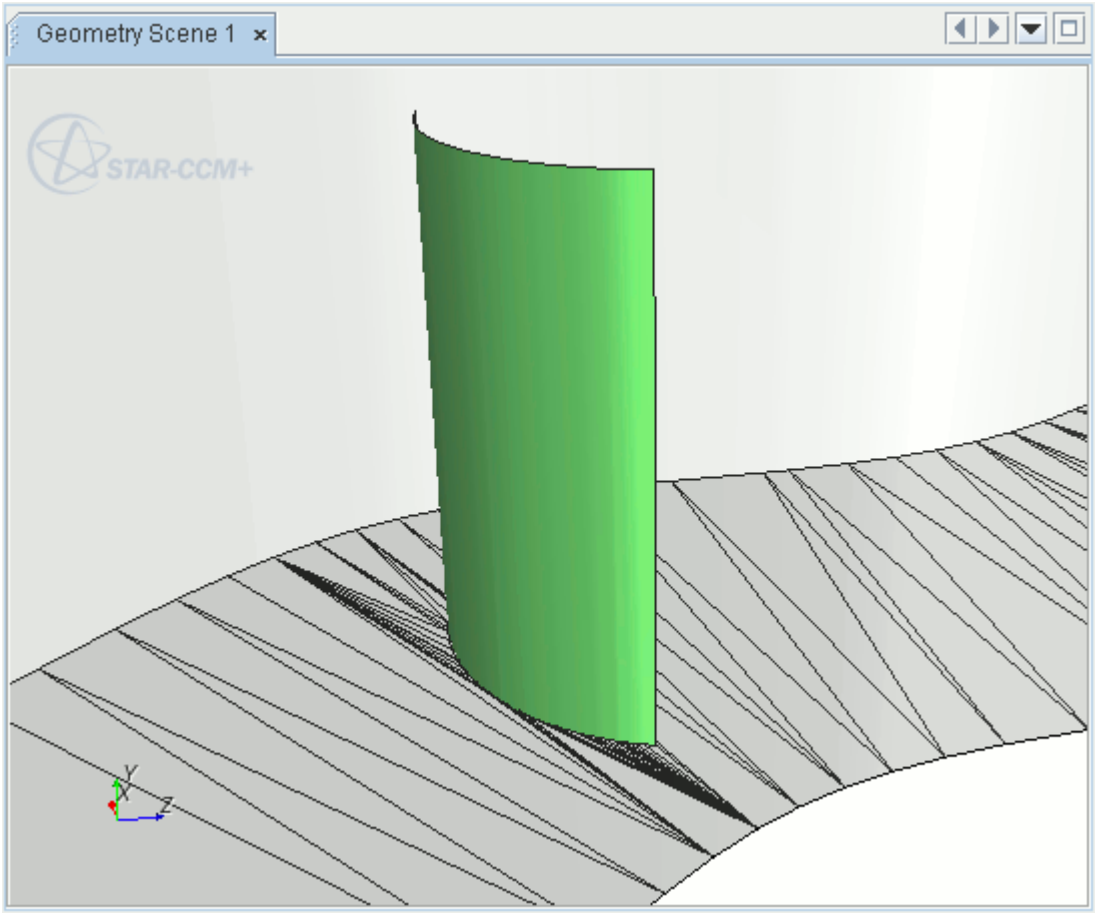
Ensuring the baffle is a Connected Surface

The merge/imprint mode of the Surface Preparation tool can be used to connect a baffle surface into the surfaces surrounding it. See Imprinting Baffle Edges onto Faces .

In the example that is shown below, a two-dimensional surface is located within the duct, but is not connected to the base surface of the duct.



In the merge/imprint options, the edges along the lower side of the baffle are added to the *Source*, and the faces forming the base of the duct are added to the *Destination*. On executing the **Imprint** operation, the lower side of the baffle is stitched into the base of the duct.



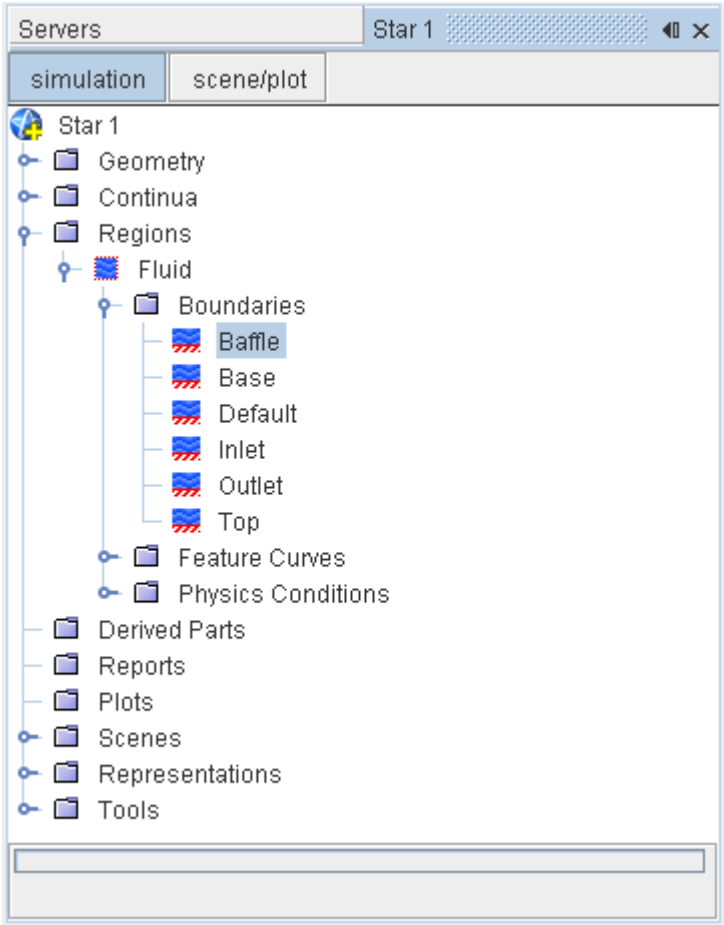
See Also:

Merge and Imprint Tool

Converting a Baffle Boundary to an Interface

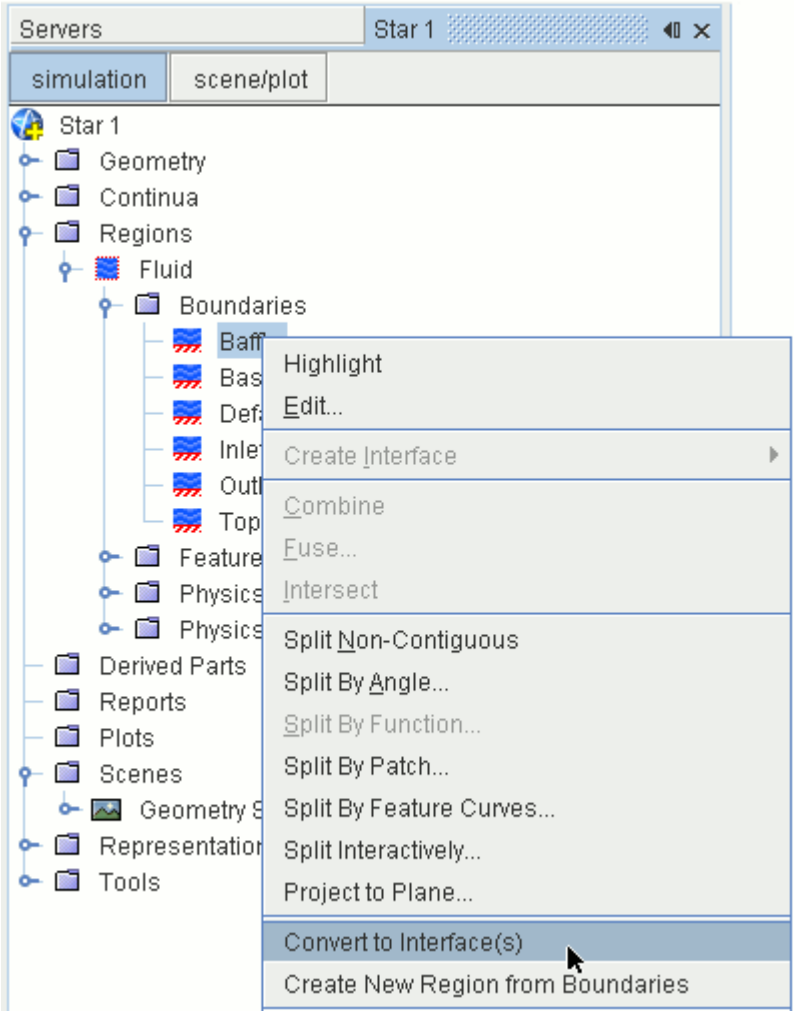
Represent a zero-thickness baffle by an interface.

The surface that is used to create a baffle must be assigned to its own boundary within the region to which it belongs.



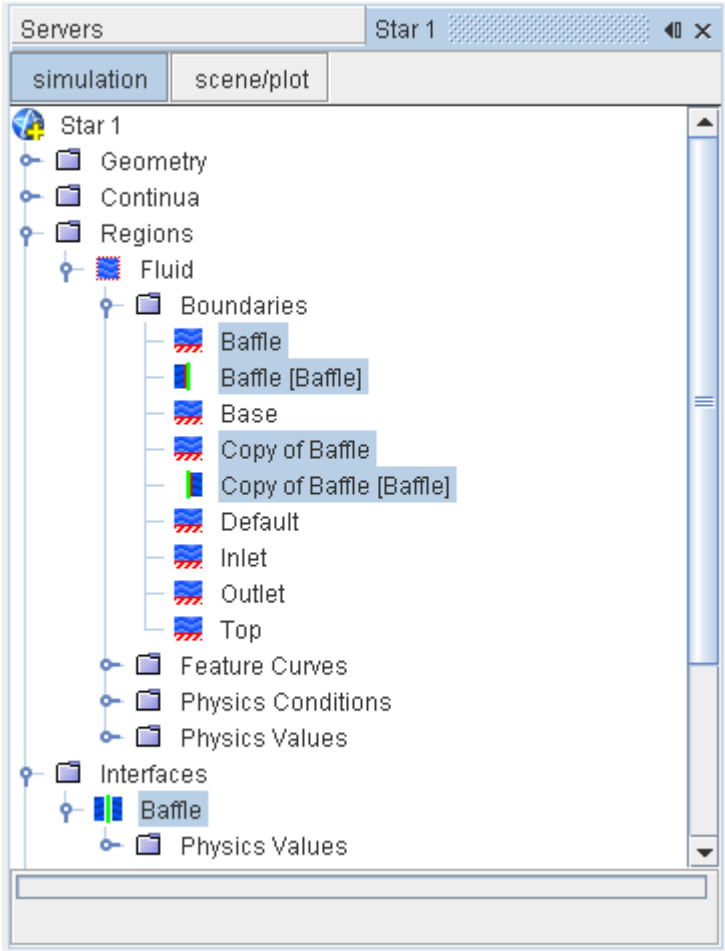
To convert this boundary to an interface:

1. Right-click the boundary representing the baffle and select **Convert to Interface(s)**.



Note that a mesh continuum must be present in the simulation, and the meshing pipeline must be initialized, before this option is available on a boundary. For more information, see Initializing the Mesh.

The result of the operation is the creation of an additional three boundaries, two of which form the opposite sides of the in-place interface. The original boundary, and the newly created copy, are used to govern any surface area that does not form part of the interface (and are therefore redundant in most cases).



In the example above, the boundaries **Baffle [Baffle]** and **Copy of Baffle [Baffle]** refer to the two sides of the interface. If necessary, different mesh or physical properties can be set for each side of the interface, for example, a prism layer on one side of the interface only. Finding out which boundary applies to which side of the baffle is described in the next step.

Distinguishing Between the Two Sides of a Baffle

The general rule for distinguishing between the boundaries on either side of a baffle is that the interface boundary with the same name as the original boundary has positive face normals.

This positive face normal is colored green when the baffle surfaces are viewed within the offset faces mode of the *Surface Preparation tool*.

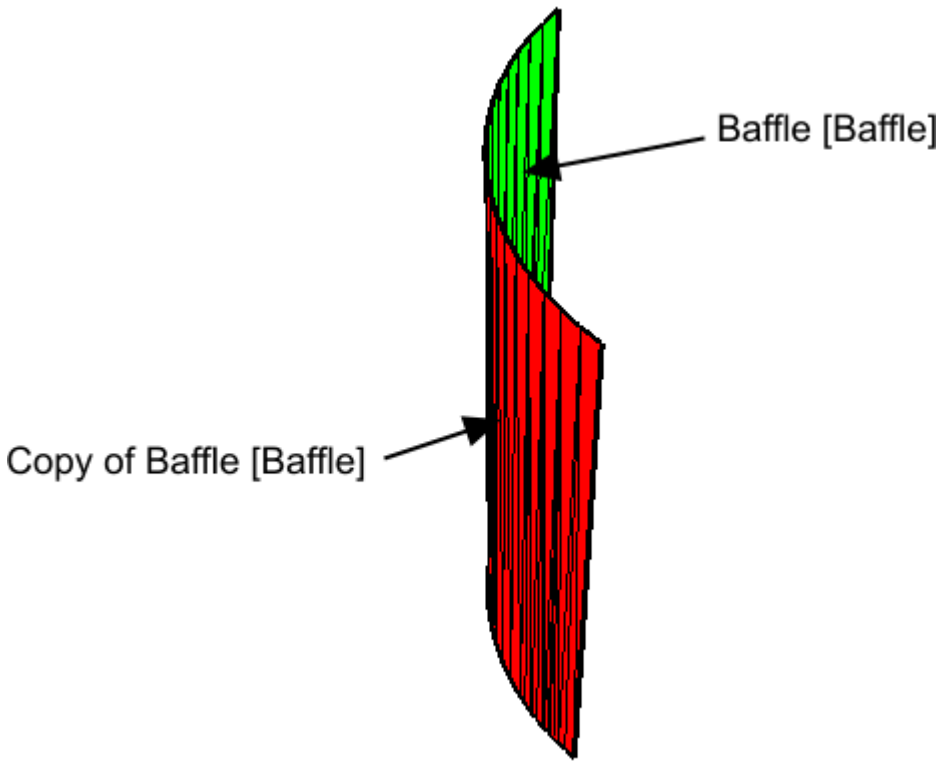
To access this view:

1. Activate the *Surface Preparation tool* from the appropriate surface representation of the model.

   See The Surface Repair Workflow.
2. Switch to the offset faces mode within the Surface Preparation tool.

   See Offset Faces/Edges Tool.
3. Activate the **Color Face Normals** checkbox.

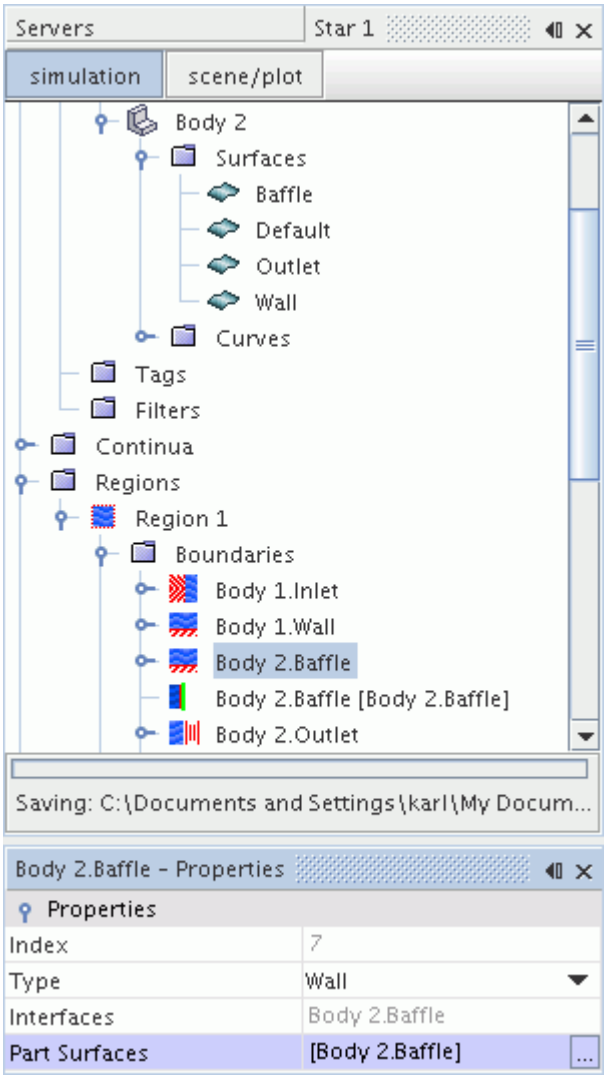
In the example that is shown here, the **Baffle [Baffle]** boundary belongs to the side colored green, and the **Copy of Baffle [Baffle]** boundary belongs to the side colored red.



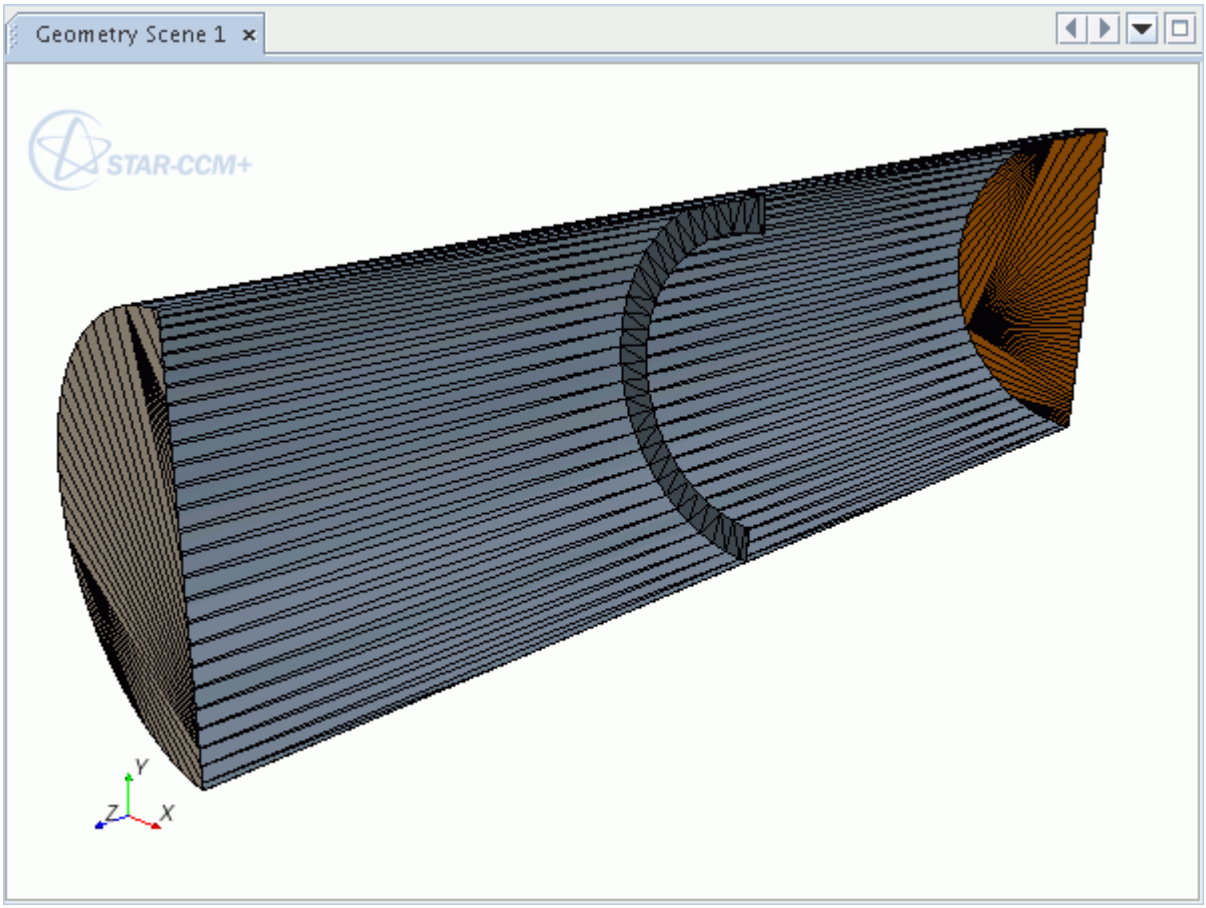
Updating Baffle Geometry

Changes that are made to the baffle geometry surface are automatically transferred to the fluid boundary, once you have connected the baffle to a surface, and converted it to an interface.

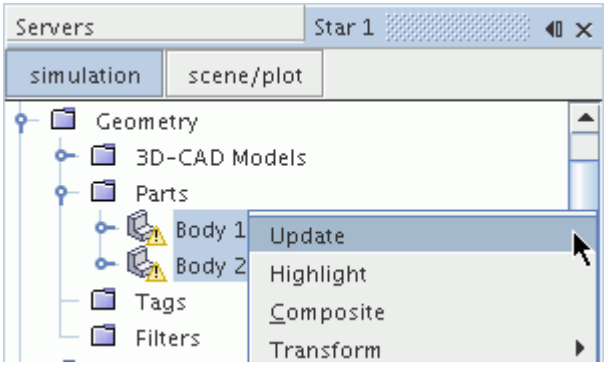
The baffle boundary must also be associated with a part surface, as illustrated in the image below. For more information, see Assigning Surfaces to Boundaries Using the Region Boundary Nodes.



The following example uses 3D-CAD to reduce the inner diameter of an orifice baffle within a pipe. If you are using an imported baffle, importing a new baffle surface as a geometry part has the same effect.



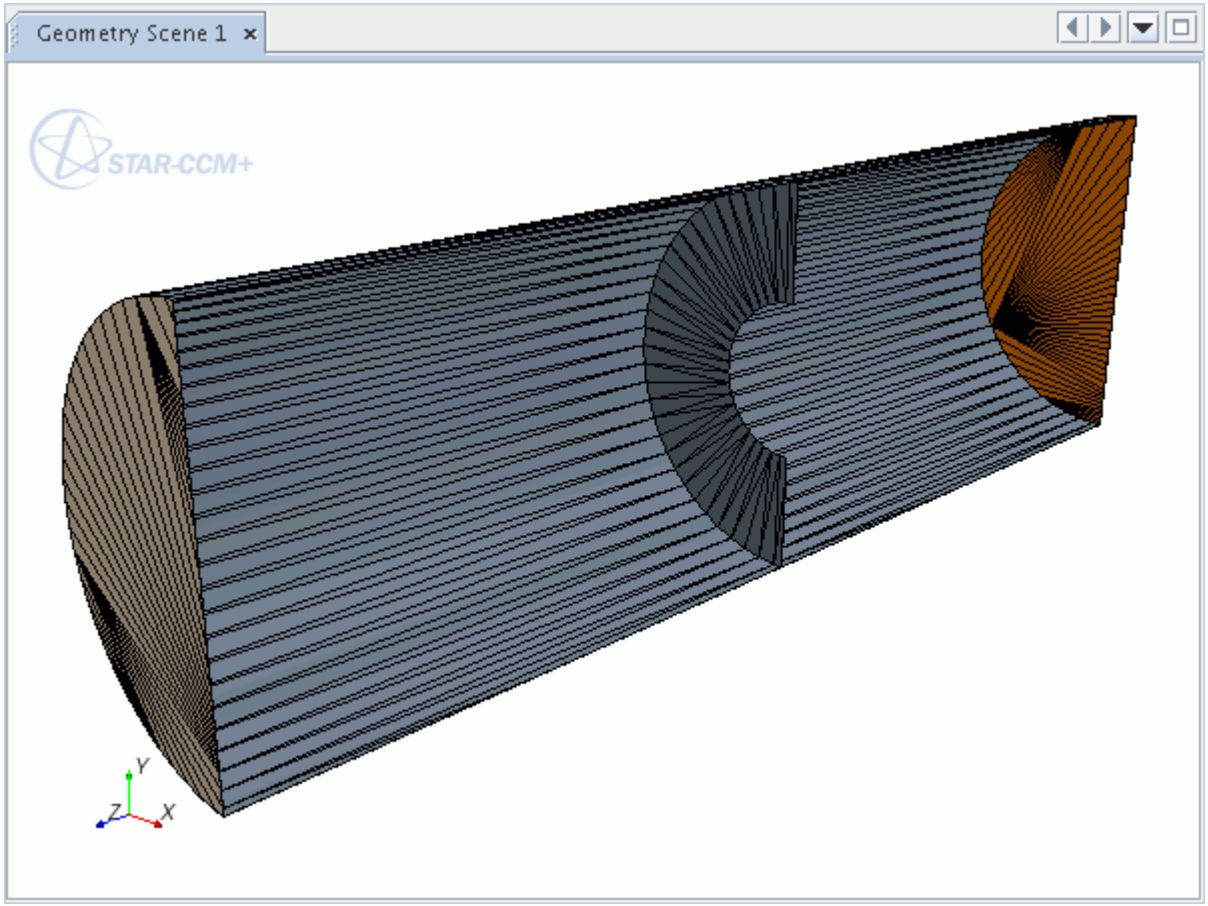
When a geometry part is changed in 3D-CAD, it must be updated before the changes are made to the geometry part.



Note that after updating or importing a part, any modifications to the surface using the surface preparation tools are over-written. Therefore, if you used the surface repair tools to connect the baffle to a surface, repeat the operation after updating the part.

1. Click **Initialize Mesh** to initialize the fluid region, which transfers the modified baffle surface to the fluid boundary.

   The updated baffle is then shown in the scene.



See Also:

Ensuring the baffle is a Connected Surface
Converting a Baffle Boundary to an Interface
Introducing 3D-CAD Modeling

Generating Baffles with Finite Thickness

If the surface wrapper is used on an imported surface containing 2D baffle faces, and the procedure for generating zero-thickness baffles is not followed, then the surface wrapper automatically inflates the 2D faces. The inflation thickness is 0.25 times the smallest refinement size that is used in the surface wrapping process (not necessarily equal to the minimum surface size specified in the **Reference Values** node).

If the surface wrapper is not used, or you require more control over the thickness of the baffles, inflate the two-dimensional baffle surface using the Offset Faces mode of the *Surface Preparation tool*. For more information, see Offset Faces/Edges Mode.

In the case where the surface wrapper has not been used, it will be necessary to perform some manual cleanup after inflating the baffle to ensure that the resulting surfaces are properly connected. It may also be necessary to duplicate some faces so that a separate solid region can be created for the baffle if it is expected to participate in a conjugate heat transfer simulation.

Exporting Mesh and Solution Data

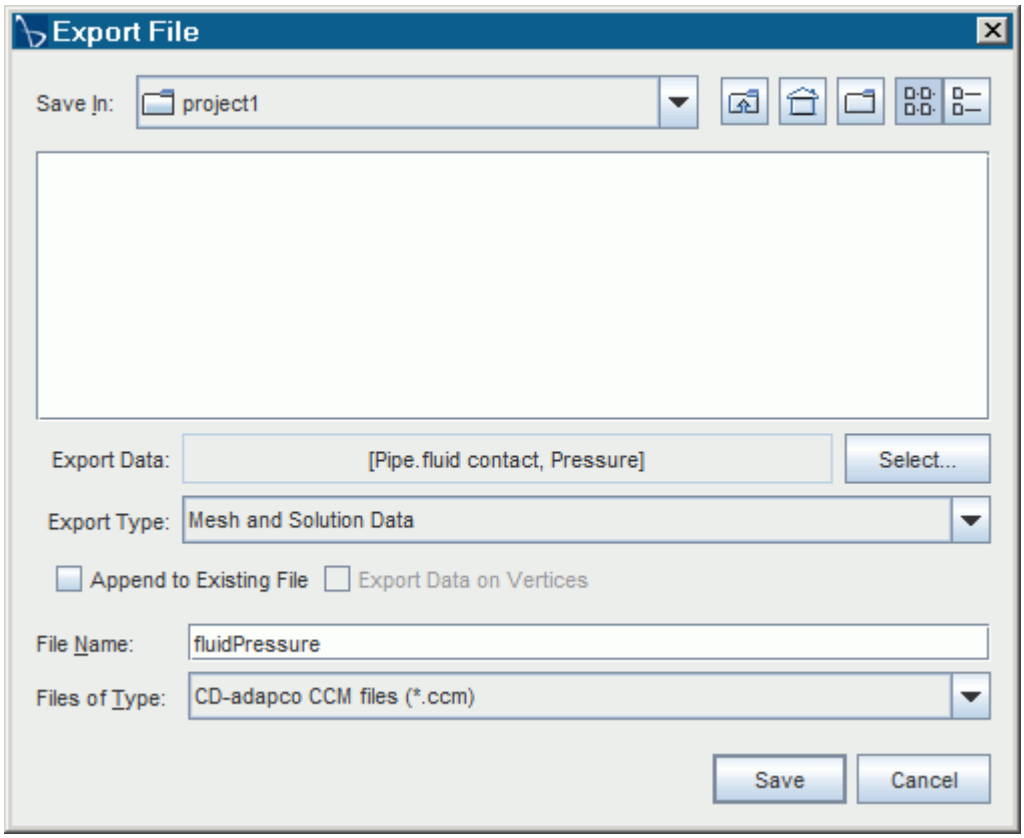
This section outlines the steps for exporting STAR-CCM+ mesh and solution data to formats that can be used in other CAE packages.

Separate sections provide instructions and format information for exporting the following:

- 3D-CAD geometry
- Mesh surfaces

1. Select **File** > **Export ...**

   This menu item opens a modified form of the operating system-specific standard *Save To File* dialog. In this dialog, you can specify the file type, the name of the new file, and which types of data to export to the file.



2. Choose the scope of the data to select from the *Export Type* drop-down list.

   The following options are available:

   - **Mesh Only**
   - **Mesh and Solution Data**
   - **Solution Data Only**

3. Use the *Files of Type* drop-down list to choose the output format.

   The currently supported export formats include:

Type of File	Type of Data Exported	Further Information
CCM (*.ccm)	Mesh data and solution data	Exports volume mesh and solution data. If you select the CCM format, you can activate Append to existing file to add the exported information to a .ccm file that you select. For example ESI/VA One, FFT/Actran, and LMS/Virtual.Lab (used in noise propagation cases), can read this format. When exporting a user-defined field function, make sure that the function property, *Function Name*, only contains characters of the following types: • A-Z, a-z, 0-9 • special characters other than the forward slash "/" (that is, ASCII values from 32 to 126, except for 47) **Note:** In CCM files, mesh indices are written as 32-bit integers. This approach sets a limit to the size of meshes that you can export using the CCM format. Estimated upper limits are ~700 million cells for trimmed meshes and ~350 million cells for polyhedral meshes.
CGNS, CFD General Notation System (HDF5) (*.cgns) **CGNS, CFD General Notation System (ADF)** (*.cga)	Mesh data and solution data	Exports volume mesh and solution data. Open format, CFD General Notation System. Appending is allowed. When exporting polyhedral volume meshes to CGNS format, regions and boundaries can import into some third-party codes as a single region and single boundary. This limitation only applies to meshes containing polyhedral cells.
Siemens Virtual.Lab Custom CGNS (HDF5) (*.cgns)	Mesh data and solution data	Exports volume mesh and solution data in a format that is suitable for LMS Virtual.Lab. This option is intended for exporting transient pressure data to Siemens Virtual.Lab, for aeroacoustic analysis. As LMS Virtual.Lab does not support polyhedral meshes, do not export regions with polyhedral meshes to this file format. However, you can export boundaries with polygonal surface meshes. Before exporting to CGNS,

Type of File	Type of Data Exported	Further Information
		STAR-CCM+ merges the boundaries and triangulates any polygonal faces. For automatic export of the transient data, set *Trigger* to either **Time Step** or **Delta Time**, as required. When you export solution data, but not mesh data, make sure that the mesh does not change during the analysis. If the simulation requires mesh repartitioning, choose to export both mesh and solution data. STAR-CCM+ automatically appends the exported data to the file that is saved during the first export. See Automatically exporting Mesh and Solution data at Regular Intervals.
Ensight Gold (*.case)	Mesh data and solution data	Exports surface mesh and solution data—from cells and vertices—although, you can also export the solution on the surface. However, when a region is selected, only surface mesh is exported. Appending to the file is allowed.
FIELDVIEW (*.fvuns)	Mesh data and solution data	Exports surface mesh and solution data. When exporting to this format from a parallel session, STAR-CCM+ includes grid partition information in the exported file. A parallel session of FIELDVIEW can use this grid partition information to distribute the data among the processes that are available to it.
Tecplot 360 (Version 2008 onwards) (*.plt)	Mesh data and solution data	Exports surface mesh and solution data—from cells and vertices—although, you can also export the solution on the surface. However, when a region is selected, only surface mesh is exported. Appending to the file is not allowed.
Boundary Shell (*.cel)	Multiple files	Creates files with `.cel` and `.vrt` formats. A third format, `.daten`, appears when you specify data by selecting field functions, boundaries, and/or part surfaces. **Note:** When exporting polyhedral meshes to `Boundary Shell (*.cel)` format, the surface mesh is triangulated before exporting.

Type of File	Type of Data Exported	Further Information
STAR-CCM+ **Boundary Data** (*.sbd)	Mesh data and solution data	Exports surface mesh and solution data. Post-Processing data. Appending to the file is not allowed.
RadTherm Patran Neutral (*.trn)	Mesh data and solution data	Exports surface mesh and solution data. RadTherm/Patran recognizes this format. Exports heat transfer coefficient and reference temperature data.
Surface FFT (*.trn)	Mesh data and solution data	Exports surface mesh and time history of the solution.
Nastran load history (*.bdf)	No mesh data—only STAR-CCM+ load data	Intended for writing time history of fluid load data for file-based FSI coupling with Nastran. For use with Auto-Export.

4. To specify the exported data, click **Select...** next to *Export Data*.
5. Use the *Select Objects* dialog to select objects for export. The objects that are available for selection depend on the specified *Export Type* and output format (that is, *Files of Type*).

   More than one variable can be selected using `<Ctrl><Click>` to select discontinuous items and `<Shift><Click>` to select a continuous range of items. To deselect an item in a list, use `<Ctrl><Click>`.

   - You can export solution data by selecting the corresponding field functions.
   - You can export mesh data by selecting the corresponding objects (that is, regions, boundaries, parts, or parts surfaces, based on the output format). The `Boundary Data (*.sbd)` and `RadTherm Patran Neutral (*.trn)` output formats only allow you to export boundaries. The `Ensight Gold files (*.case)` or `Surface fft (.trn)` allow you to export mesh data available on derived parts. The `Boundary Shell (*.cel)` output format also allows you to export mesh data available on parts and part surfaces.
6. Enter the output *File Name* and click **Save**.

   It is also possible to export automatically at regular intervals.

Contents:

Automatically Exporting Mesh and Solution Data at Regular Intervals
Automatically Exporting Mesh and Solution Data at Regular Intervals

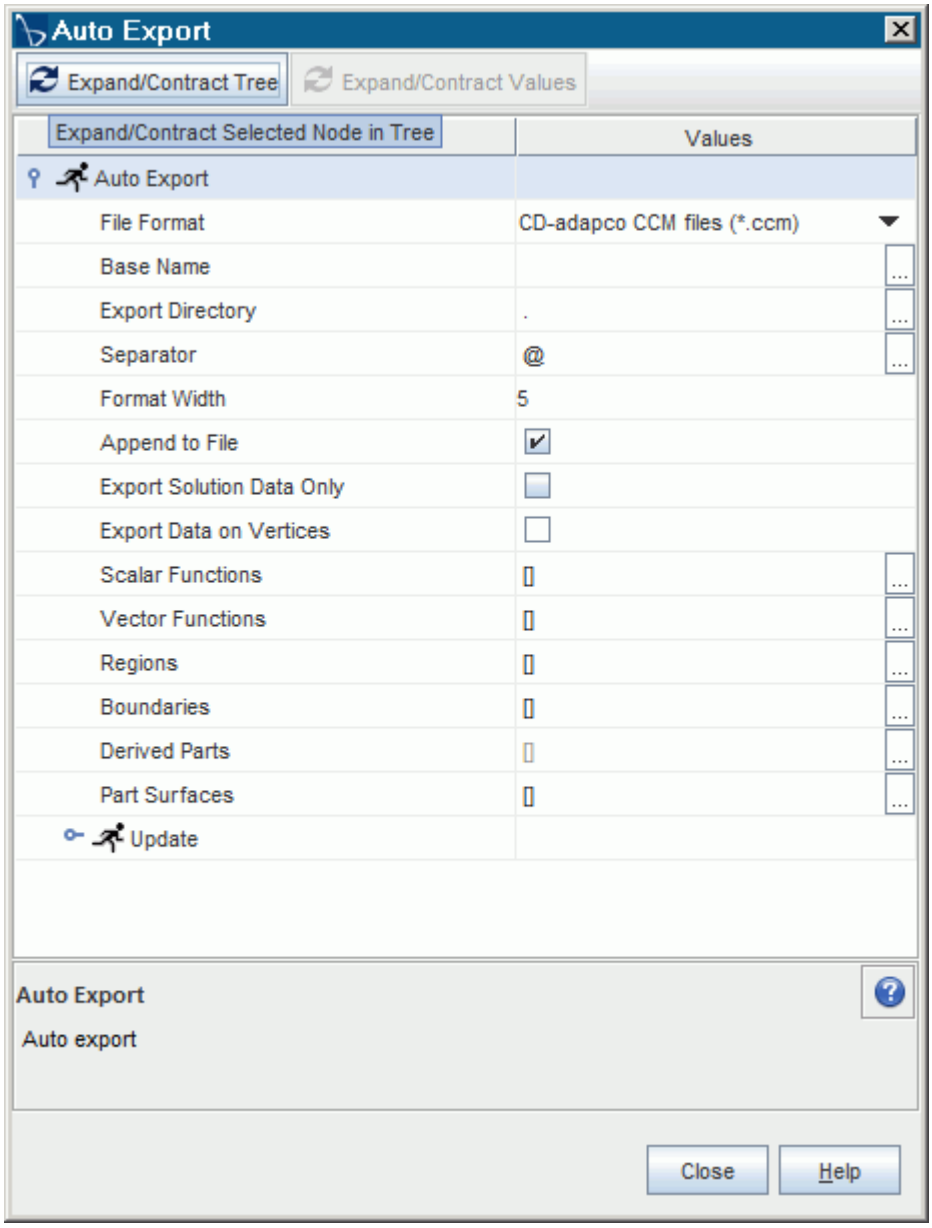
Automatically Exporting Mesh and Solution Data at Regular Intervals

When running a long simulation (either interactively or in batch mode) it is useful to enable the auto export feature. This feature causes STAR-CCM+ to export the solution data automatically at regular intervals during the run, when the **Export...** command is disabled. This section includes a description of the properties of the *Auto Export* dialog.

To set up the auto-export of solution data:

1. Select **File** > **Auto Export...** from the menu.

   The *Auto Export* dialog appears.



2. Expand the **Auto Export** > **Update** node and activate *Enabled* .

To specify the file format that you want to export to:

3. Select the file format from the drop-down list in the Values column of the *File Format* property.
4. Specify the *Base Name*.

   The base name is the prefix of the name of the file that you want to export your data to.
5. Specify the *Export Directory* if necessary.

This directory is the location of the file that you want to export your data to. The default path is the same as the path of your simulation file. Clicking ... (**Custom Editor**) activates a dialog similar to the *Save* dialog. You can use this dialog to navigate to a new location for the file.

6. Enter the type of separator that should be used in the *Separator* property.

   By default, when STAR-CCM+ saves copies of the simulation file, it places a `@` character to separate the base file name and the number of the trigger event. If the use of this character in this manner causes any problems with your system, you can enter a different one as the separator. If you attempt to leave this field blank, STAR-CCM+ puts the `@` character there by default.
7. If you want to add the exported data to an existing file, activate *Append To File*.
8. If you prefer not to export the mesh data, activate *Export Solution Data Only*.
9. To export vertex data, activate *Export Data on Vertices*.

   This option is only available when your export includes solution data, and when you are exporting to an Ensight Gold file.

To select the scalar values that you want to export:

10. Click ... (**Custom Editor**) next to *Scalar Functions*.
11. In the *Auto Export - Scalar Functions* dialog, select the desired functions and click **OK**.
12. Using the same method as defining the scalar functions, you can select vector functions, regions, boundaries, derived parts, and parts surfaces.
13. Select the trigger type from the drop down menu of the *Trigger* property.

    This property specifies how the auto export operation is triggered. Unless you are running a transient analysis, leave the setting at `Iteration`. For more information, see Trigger Types.
14. To control the number of intervals that pass between auto exports, expand the **Trigger** > **Frequency** node and enter the value that you want. See Trigger Properties.
15. Click **Close**.

STAR-CCM+ automatically exports during the next run at the intervals you specified. To deactivate this feature, simply reopen the *Auto Export* dialog and deactivate *Enabled*.

Contents:

Auto Export Properties
Exporting Region and Boundary Data
Auto Export Properties
Exporting Region and Boundary Data

Auto Export Properties

You can use the following properties to activate and customize this feature:

File Format

Selects the format of the file from a drop-down list.

Base Name

The prefix of the file name to use when exporting your data.

Export Directory

The directory that contains your file with exported data.

Separator

Specifies the character that appears between the base file name and the number of the trigger event (`@` by default).

Format Width

Width of the iteration, time, or step number in the formatted file name.

Append to File

When activated, adds the exported information to an existing file. This file would be the one you specify in the *Base Name*, *Export Directory* and *File Format* properties.

This option works for `.ccm`, Ensight and `.trn` file formats only.

Export Solution Data Only

When activated, exports solution data without any mesh data.

Export Data on Vertices

When activated, exports vertex data. This option is only available when your export includes solution data, and when you are exporting to an Ensight Gold file.

Scalar Functions

Determines which scalar values to export.

Vector Functions

Determines which vectors to export.

Regions

Selects regions to export. See Exporting Region and Boundary Data.

Boundaries

Selects boundaries to export. See Exporting Region and Boundary Data.

Derived Parts

Selects derived parts to export. See Accessing Solution Data through Derived Parts.

Part Surfaces

Selects part surfaces to export.

Update

When expanded, this property makes available the controls listed below.

Enabled

When activated, enables the auto export feature.

Trigger

Provides choices for triggering the automatic export. See Trigger Types.

Iteration Frequency

Controls the number of trigger events between exports. See Trigger Properties.

The settings are saved in the simulation database.

Exporting Region and Boundary Data

The following rules apply when exporting region and/or boundary data:

- If no regions and no boundaries are selected, data for all the regions (including the boundaries) in the simulation is exported.
- Selecting a region automatically selects the boundaries that are associated with that region. Data for the selected region (including the associated boundaries) is exported. This action has the same effect as manually selecting a region and the boundaries in that region.
- Selecting a boundary does not automatically select the region that is associated with that boundary. Data for the selected boundary is exported.
- Selecting boundaries without also selecting regions results in a `.ccm` file that contains only boundary data. This option is not compatible with pro-STAR.

Multi-Region Meshing

This section outlines the surface requirements for performing multi-region meshing, as well as step-by-step instructions when starting from surface imports that result in single region and multiple regions definitions. Additionally, guidelines are provided on what the best approaches are for exporting surface data from CAD packages when multiple regions are required.

Contents:

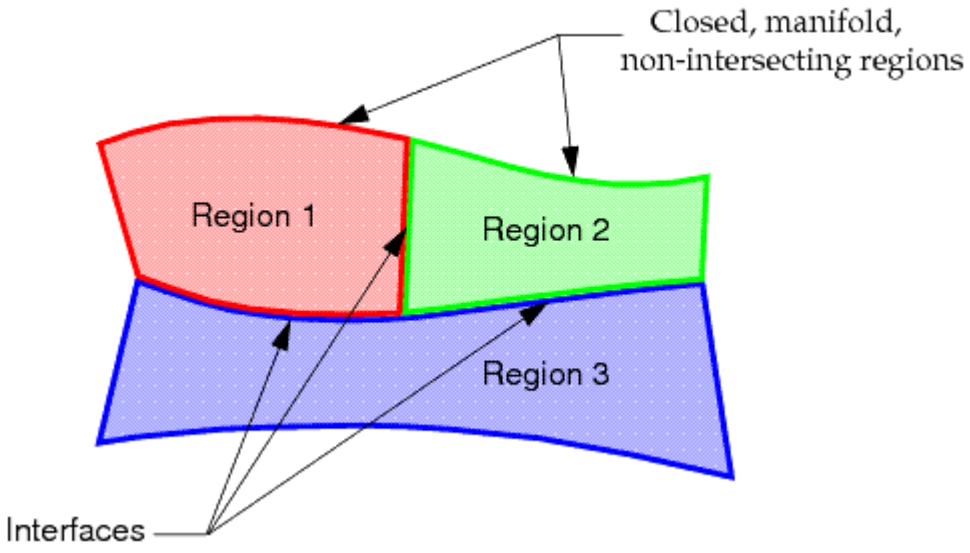
Multi-Region Surface Requirements
Starting from a Single Region
Starting From Multiple Regions
CAD Data Export Guidelines for Assemblies

Multi-Region Surface Requirements

For STAR-CCM+ to remesh surfaces successfully, and/or generate a volume mesh for a multi-region case with conformal mesh between regions, prepare the surface to adhere to the following conditions:

- each region must be closed, manifold, and non-intersecting
- boundaries that exist between regions must be grouped and flagged as interfaces

The above requirements are illustrated here:



The surface remesher can automatically produce a conformal mesh across an in-place interface containing mismatched boundaries. To do this, STAR-CCM+ imprints one boundary on the other so that a mapped boundary surface is produced, resulting in a conformal triangulation. For a case such as that above, use a single mesh continuum to mesh all regions in order to ensure that a conformal surface and volume mesh is produced at interfaces.

In instances when a conformal volume mesh is not required between each region, then the requirement to have interfaces and imprints defined is relaxed. In this case, each region can be meshed independently by using a different mesh continuum or by turning on per-region meshing. The meshes can later be joined back together either by using an in-place interface or by fusing the boundaries. In some respects, this approach is far easier than ensuring that the region interfaces and imprints are properly defined. However, the process results in increased overhead at the simulation stage, due to the extra interface information that is stored and the extra face divisions in the boundary fusing process.

When dealing with multi-region topologies, there are many factors in the definition of the CAD data that affect the requirements for STAR-CCM+. These factors include:

- the format that is used to export the CAD data, for example, STL, IGES, VDA
- whether each part in the assembly is exported to the same file or to separate files
- whether duplicate surfaces exist at inter-region boundaries
- whether inter-region surfaces have been imprinted or not
- whether the assembly contains all relevant fluid/solid parts or only the solid material portion of the geometry.

For example, certain CAD packages such as SolidWorks allow you to save an assembly as a single STL file or each part as a separate STL file. Similarly, in most CAD packages, you can elect to open parts individually and save them as separate IGES files or other format. Alternatively, the entire assembly can be saved to a single file, which is usually the default (or only option allowed). Depending on the method that is used to save the import file or files, inter-region boundaries can have either single or duplicate surfaces, which do not necessarily have identical perimeter triangulation. Additionally, some CAD packages allow imprinting (whereby one part can be “stamped” onto another, leaving an edge definition that represents the common area) while others do not have this feature. Finally, the assembly typically only represents the solid material, rather than the fluid volume that is required for CFD simulations. Some CAD packages easily allow the fluid portion to be extracted and combined back into the assembly while others do not. In summary, many different factors influence the level of effort that is required to prepare the import geometry for STAR-CCM+.

In general, regardless of the factors that affect the multi-region geometry definition, STAR-CCM+ contains the required tools to prepare the surface for meshing. In particular, two operations specific to multi-region meshing are essential for manipulating improperly prepared surfaces in order to conform to the above requirements. They are:

1. create a (duplicate) interface boundary
2. create a new region from a collection of boundaries.

Being able to create a duplicate boundary at an inter-region interface allows two regions that were previously “joined” to be separated out, allowing a new region to be created from individual boundaries. The surface boundary that was duplicated maintains an in-place interface definition as part of its make up so no further action is required once the region has been redefined.

The use of the in-place interface option is important for defining the surface correctly in cases where regions are defined at import but no connectivity information between regions is included. By including an in-place interface, conformal meshes at interface boundaries can be created for the surface and volume meshers.

Two different scenarios are discussed here showing how the initial CAD surface data can be manipulated in STAR-CCM+ so that a conformal volume mesh can be constructed. The two different scenarios are:

- the CAD import results in a single region
- the CAD import results in multiple regions.

See Also:

Surface Remesher
In-Place Interface Topology
Using Per-Region Meshing
Fusing Boundaries
Converting Boundaries to Interfaces
Creating Regions From Boundaries

Starting from a Single Region

The discussion here focuses on multi-region cases that start off as a single region as a result of the import process into STAR-CCM+.

For example, the following operations all result in the creation of a single region definition during import:

- Selecting the **One region for all files** option for **Region Mode** when importing multiple files at the same time
- Importing an assembly that has been saved to a single file.

Depending on the surface export options in the CAD package and the import options in STAR-CCM+, surfaces at inter-region boundaries are sometimes duplicated. This leads to two different scenarios:

- Inter-region boundaries are common/shared (and therefore have same perimeter triangulation)
- Inter-region boundaries are duplicated and can have dissimilar areas.

Contents:

Common Inter-Region Boundaries
Inter-Region Boundaries Are Duplicated

Common Inter-Region Boundaries

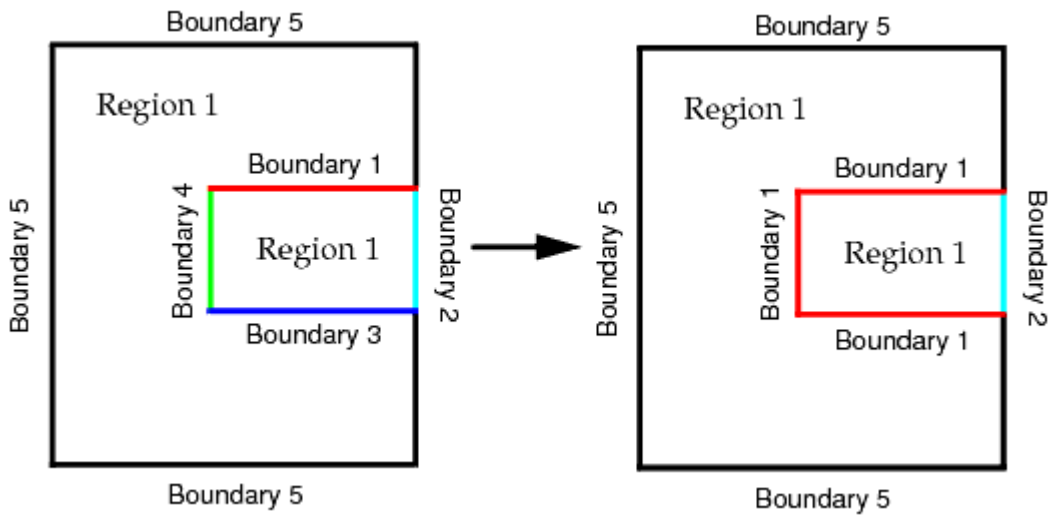
In cases with common inter-region boundaries, the split by surface topology option can be used to split regions into their constituent parts and generate the appropriate boundary interfaces to ensure a conformal mesh.

This option effectively performs steps 1 through 3 as described below and also allows regions that originally belong to the same boundary to be combined automatically, reducing the overall number. For more information, see Splitting by Surface Topology.

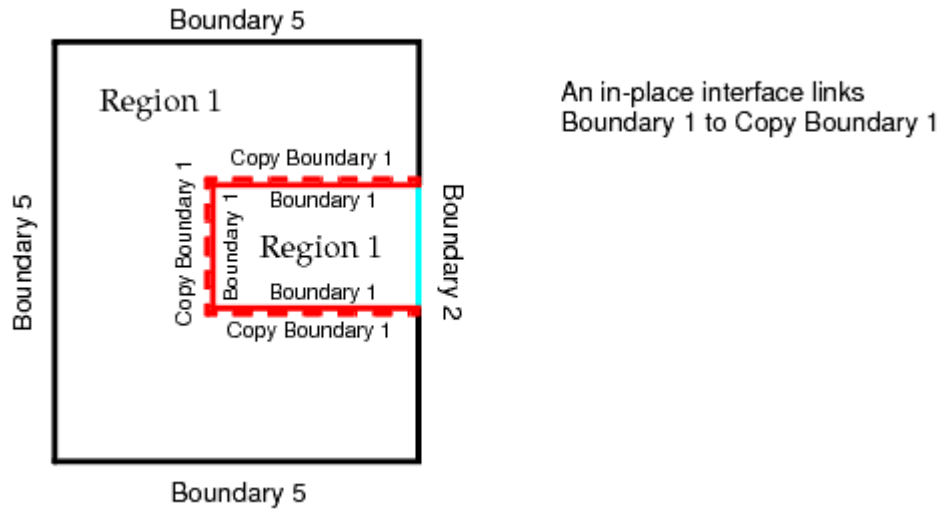
The above option is the recommended route for all such cases, from simple to complex, as no manual interaction is required. However, if you prefer to define the individual regions by hand, then follow the guidelines below to prepare the surface further:

1. Combine boundaries that belong to the same inter-region interface together into one boundary. Repeat as necessary for all inter-region boundaries.

   See Combining Boundaries.



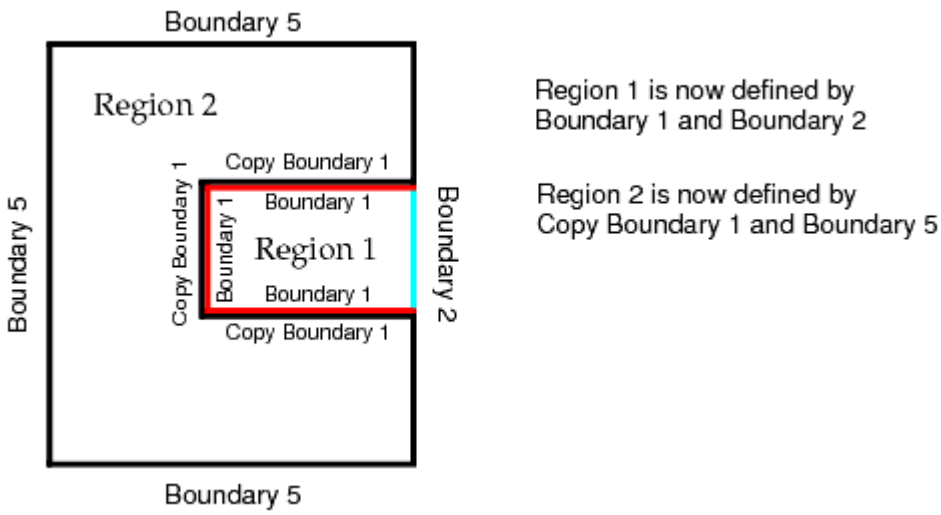
2. Use the **Convert to Interface(s)** option to create a duplicate boundary for all the inter-region boundaries. See Converting to an Interface.



A pair of in-place internal interfaces are generated at the same time linking the boundary pair.

3. For each new region in turn, select the boundaries that make up the region and select the **Create New Region from Boundaries** option.

   A new region is created. One half of the in-place internal interface is also transferred into the new region. Alternatively, if several new regions are required, the split non-contiguous region option can be used instead to perform the operation in one go. For more information, see Splitting Non-Contiguous Regions.



4. Continue with the meshing process as normal.

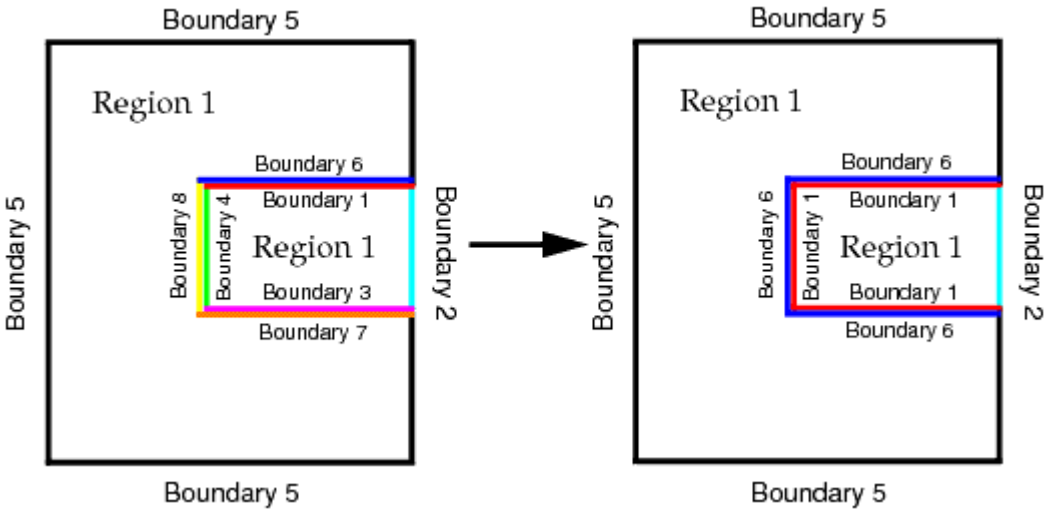
See Also:

Creating Regions From Boundaries

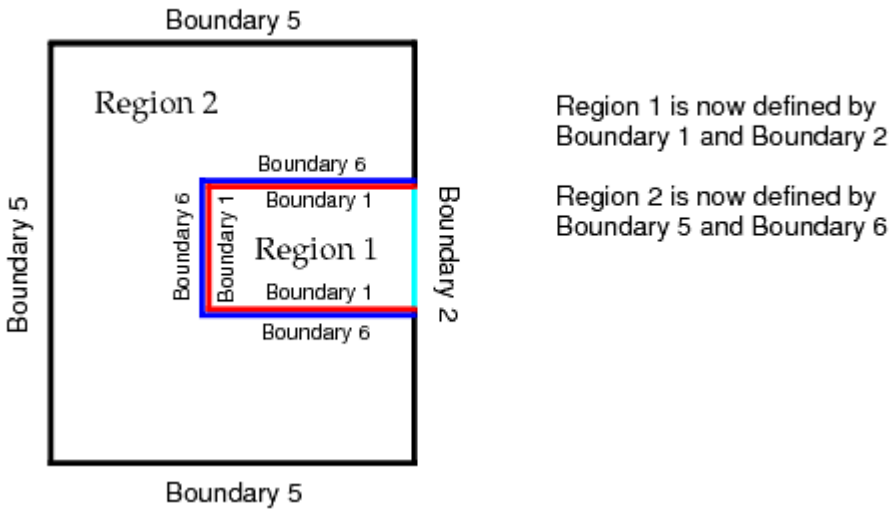
Inter-Region Boundaries Are Duplicated

In cases where the inter-region boundaries are not common (that is, duplicated) and the perimeter triangulation and boundary areas may not be the same, follow the guidelines below:

1. Combine boundaries that belong to the same inter-region interface together into one boundary. Repeat as necessary for all inter-region boundaries.



2. Separate out each region either by using the **Create New Region from Boundaries** or the **Split Non-Contiguous Region** options, such that each region is composed of unique boundaries.



3. Create an in-place interface between each boundary pair (**Boundary 1** and **Boundary 6** in the above example) and then proceed with the meshing process as normal.

   In this instance, the surface remesher imprint process is used to produce the same triangulation for inter-region boundaries before proceeding with producing new triangulation for each region. A conformal mesh is produced at the inter-region boundary interface. The original import or initial surface representation is left unaffected by this process. For additional information on using the surface remesher with interfaces, see Surface Remesher with Interfaces.

See Also:

Combining Boundaries

Creating Regions From Boundaries

Splitting Non Contiguous Regions

Surface Remesher

Starting From Multiple Regions

The discussion here focuses on multi-region cases that have individual regions that are defined during the import process in STAR-CCM+.

For example, the following operations all result in the creation of multi-region definitions during import:

- Selecting the **One region per file** option for **Region Mode** when importing multiple files at the same time
- Importing each file one at a time.

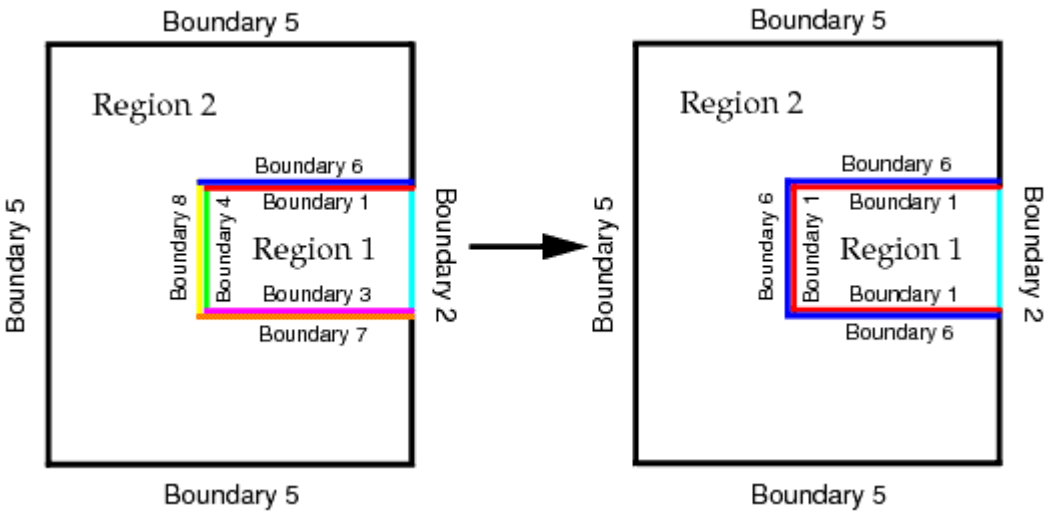
Surfaces at inter-region boundaries can have dissimilar triangulation, depending on how they were prepared. This leads to two different scenarios existing:

- Inter-region boundaries have the same triangulation or the same perimeter triangulation

- Inter-region boundaries do not have the same perimeter triangulation or do not have the same areas.

Regardless of which of the above scenarios you have, the same procedure is used to prepare the surface ready for meshing. Follow the guidelines below:

1. Combine boundaries that belong to the same inter-region interface together into one boundary. Repeat as necessary for all inter-region boundaries.



2. To create an interface between each inter-region boundary pair, use the in-place interface option.
3. Continue with the volume meshing process as normal.

   In this instance, the surface remesher imprint process is used to produce the same triangulation for inter-region boundaries before proceeding with producing new triangulation for each region. A conformal mesh is produced at the inter-region boundary interface. The original import or initial surface representation is left unaffected by this process.

See Also:

Combining Boundaries

Surface Remesher

Surface Remesher with Interfaces

CAD Data Export Guidelines for Assemblies

The text below provides guidelines for the best approach for how to export multi-region cases from CAD packages and what issues are typically avoided or encountered when they are imported into STAR-CCM+.

Guidelines are provided for the following formats:

Contents:

STL

NASTRAN or PATRAN

IGES, STEP, or VDA
Parasolid Transmit

STL

Save each part in the assembly as a separate STL file but use the same edge length deviation and curvature angle settings for each. Import the separate STL files into STAR-CCM+ using the **One part per file** option which results in separate parts automatically being defined. Split the single boundary definition for each region up by angle and add feature curves as normal to define the edges. The perimeter triangulation at the inter-region boundaries will most likely be different (dependent on the CAD package) but a single mesh continuum with interfaces can still be used, resulting in a conformal mesh at the inter-region interfaces. If the surface triangulation is such that a single mesh continuum cannot correctly edge zip and swap out inter-region boundaries, then a different mesh continuum must be used for each region, resulting in a non-conformal volume mesh interface.

If you save the assembly to a single STL file, the import process usually results in a single region with a single boundary, regardless of the option that is selected for **Boundary Mode.** Few CAD programs support the multiple solid definition that STL allows. Split the single boundary up using the non-contiguous option and then split the regions up using the non-contiguous option. Add features curves as normal and continue with the meshing process for multiple regions.

See Also:

Starting From Multiple Regions

NASTRAN or PATRAN

Where the CAD program can export tessellated surfaces as Nastran or Patran files, save the assembly to a single file if the inter-region boundaries are common, or have the same perimeter triangulation. When this file is imported into STAR-CCM+, the boundaries can then be separated out into interface and non-interface groups and duplicate copies can be made of the boundaries at the inter-region interface. New regions can then be created and a single mesh continuum can then be used to generate the volume mesh for the entire geometry resulting in a conformal mesh at the boundary interface.

If the assembly contains duplicate inter-region boundaries with different triangulation on each, then save each part to a separate file if possible. When you import these files into STAR-CCM+, you are presented with the choice of whether to create one region per file (recommended), or one region for all files. Regardless of the choice that you make, individual regions often have dissimilar triangulation at the inter-region boundaries. Usually STAR-CCM+ can resolve different inter-region boundary triangulation by using the edge zipper to match the perimeter nodes and then deleting one of the boundaries to allow the creation of a conformal interface. However, depending on the variation in perimeter node positioning, the edge zipping can fail, in which case, use a different mesh continuum for each region, which results in a non-conformal volume mesh interface.

IGES, STEP, or VDA

If using IGES, STEP, or VDA formats to export from the CAD package, save the assembly to a single file. STAR-CCM+ can read this file and translate the surface. At inter-region boundaries, STAR-CCM+ automatically detects surfaces that are within close proximity and attempts to merge them, resulting in a

common boundary at the interface (note that this process is not always successful). It also provides the option for generating feature curves. The boundaries can then be separated out to interface and non-interface groups and duplicate copies are made of the boundaries at the inter-region interface. New regions can then be created which have the same triangulation for the boundaries at the interface. A single mesh continuum can be used to volume mesh the entire geometry, resulting in a conformal mesh at the inter-region boundary interface.

If you save the individual parts in the assembly as separate IGES, STEP, or VDA files, then on importing them into STAR-CCM+ (either at the same time or one after the other), you are given the choice of whether to create one region per body, one region per file (recommended) or one region for all files. Regardless of the choice that you make, individual regions often have dissimilar triangulation at the inter-region boundaries. Usually STAR-CCM+ can resolve different inter-region boundary triangulation by using the edge zipper to match the perimeter nodes and then deleting one of the boundaries to allow the creation of a conformal interface. However, depending on the variation in perimeter node positioning, the edge zipping can fail, in which case, use a different mesh continuum for each region, which results in a non-conformal volume mesh interface. For this reason, it is recommended that you use the first approach that is discussed above to simplify the meshing process and guarantee a successful conformal interface.

See Also:

Parasolid Transmit

Parasolid Transmit

If using Parasolid Transmit to export from the CAD package, save the assembly to a single file. This file can then be imported into STAR-CCM+ directly, maintaining the names of the parts in the assembly. Refer to the importing IGES/STEP/VGA section for more information on working with multi-region geometries.

See Also:

IGES, STEP, or VDA

Simulating Physics

This part of the documentation describes the models in STAR-CCM+ and how to use them.

Separate chapters deal with:

- Space, time, and motion
- Materials
- Flow and energy
- Species
- Turbulence and transition
- Radiation
- Aeroacoustics
- Combustion
- Multiphase flow
- Dynamic fluid body interaction
- Harmonic balance
- Solid stress
- Electromagnetism
- Electrochemistry
- Casting

Contents:

Defining Physics Workflow
Setting Up Physics
Running Physics Simulations
Using Overset Meshes
Modeling Space, Time, and Motion
Modeling Materials
Modeling Flow and Energy
Modeling Viscous Flow
Modeling Heat Transfer
Modeling Species
Modeling Porous Media
Solving the Flow Adjoint

Modeling Fans
Virtual Disk Model
Modeling Turbulence and Transition
Modeling Transition
Modeling Wall Distance
Modeling Radiation
Modeling Aeroacoustics
Combustion and Other Reacting Flows
Modeling Multiphase Flow
Modeling Dynamic Fluid Body Interaction
Modeling with Harmonic Balance
Solid Stress
Modeling Solid Stress
Modeling Electromagnetism
Modeling Electrochemistry
Modeling Plasma
Modeling Batteries
Modeling Region Sources
Remedying Cell Quality
Controlling Domain Decomposition
Guidelines for Applying STAR-CCM+
Solving Transport Equation

Defining Physics Workflow

STAR-CCM+ contains a wide range of physics models and methods for the simulation of single- and multi-phase fluid flow, heat transfer, turbulence, solid stress, dynamic fluid body interaction, aeroacoustics, and related phenomena. These physics models are all selected using a physics continuum.

Although you can create and define physics continua without having any regions or volume mesh present, you must have both in order to run a simulation.

To create physics continua and define supporting objects:

1. Create one physics continuum for each material or material mixture that you require in the simulation. If you have multiple adjacent solid parts, you can use the multi-part solid model to define multiple solid materials in the same continuum. To create a physics continuum:

   a) Right-click the Continua node and select **New** > **Physics Continuum**

   The new continuum receives the next logical sequential name such as Physics 2 but can be renamed if necessary. Manager nodes for **Models**, **Reference Values**, and **Initial Conditions** are added as child nodes to the new continuum node. Once the new physics continuum has been created, you can proceed to select the physics models for it.

   Note: If you import a mesh into the simulation, STAR-CCM+ automatically adds a physics continuum for you.

2. Choose the physics models and materials. For each physics continuum:

   a) Right-click the **Continua** > **[physics continuum]** > **Models** node and click **Select Models...**

   b) In the *Physics Model Selection* dialog, for each group of choices on the left-hand side, choose a model. Some groups require a choice, indicated by **<Select One>**. Other groups present optional choices.

      By default, the option **Auto-select recommended models** is active. When this option is active, STAR-CCM+ selects default models for some groups in the dialog. For example, if you select the **K-Epsilon Turbulence** option from the *Turbulence* group, STAR-CCM+ automatically chooses the **Realizable Two-Layer K-Epsilon** model.

   c) If STAR-CCM+ makes an automatic selection that you wish to change:

      1. Deactivate **Auto-select recommended models**.
      2. Within *Enabled Physics Models*, deactivate unwanted selections until you return to the earlier state.
      3. Choose the correct sequence of models.

3. Assign a physics continuum to each region in the simulation. To do this, for each region, either:

   - select the **Regions** > **[region]** node and set *Physics Continuum*, or
   - drag the physics continuum object onto the region node.

Region nodes update their icon according to whether the region is assigned to a solid or fluid continuum.

4. Complete the setup for materials and physics models.

   The actual steps you must follow varies depending on the physics models that you choose. Other parts of the user guide provide detail on specific models.

5. Define global parameters.

   If you wish to use the same input value (scalar or vector quantity) in more than one location in a simulation, you can set the value using a global parameter, and access this where required. For details, see Global Parameters.

6. For each physics continuum, set reference values.
7. For each physics continuum, set initial conditions:
   a) Expand the **Continua** > **[physics continuum]** > **Initial Conditions** node.
   b) Select the child nodes in turn and set the *Method* and *Value* properties appropriately.
8. For each region in the simulation, set any quantities that act on the region as a whole:
   a) Expand **Regions** > **Physics Conditions**, select each child node, and set options appropriately. Some nodes, such as the **Initial Condition Option**, allow you to override continuum values at the region level.
   b) If you have multiple parts within the region, decide whether you want to set values for the whole region, or for subgroups of parts within the region.
      - To set values on subgroups of parts, select the **[region]** node and activate **Allow Per-Part Values**.
   c) Within **[region]** > **Physics Values**, select each node and set values according to the real scenario you are modeling.
      - To set values on subgroups of parts, select the value node and set **Method** to **By Part Subgroup**. Then select the child nodes within **By Part Subgroup** and set the *Method* and *Value* properties for the parts within the subgroup.
   d) If you wish to add motion to a region, review the information provide on motion objects.
9. Define boundary conditions for each region. Within a region:
   a) Expand the **Boundaries** > **[boundary]** > **Physics Conditions** node and its corresponding **Physics Values** node.
   b) Review each condition. Choose methods or options that suit the data that you have for that boundary. As you modify conditions, STAR-CCM+ adds or removes nodes within the **[boundary]** > **Physics Values** manager node.
   c) If you have multiple part surfaces within the boundary, decide whether you want to set values for the whole boundary, or for subgroups of part surfaces within the boundary.
      - To set values on subgroups of part surfaces, select the **[boundary]** node and activate **Allow Per-Surface Values**.
   d) Within **[boundary]** > **Physics Values**, select each node and set values that match the real scenario that you are modeling.
      - To set values on subgroups of part surfaces, select the value node and set **Method** to **By Surface Subgroup**. Then select the child nodes within **By Surface Subgroup** and set the *Method* and *Value* properties for the part surfaces within the subgroup.
10. Define interface conditions. This procedure requires similar steps to those for boundary conditions:

a) Expand the **Interfaces** > **[interface]** > **Physics Conditions** node and its corresponding **Physics Values** node (if present).

b) Review each condition. If you choose an option that requires a value,STAR-CCM+ adds the node within the **Physics Value** node for the interface.

c) If the interface is based on part surfaces, and has multiple parts assigned to it, you can choose to set values for the whole interface, or for subgroups of part contacts within the interface.

- To set values on subgroups of part contacts, select the **[interface]** node and activate **Allow Per-Contact Values**.

d) Within **[interface]** > **Physics Values**, select each node and set values that give you the behavior you require.

- To set values on subgroups of part contacts, select the value node and set **Method** to **By Contact Subgroup**. Then select the child nodes within **By Contact Subgroup** and set the *Method* and *Value* properties for the part contacts within the subgroup.

11. Many physics models expose specific solvers that control the solution of the model equations. Expand the **Solvers** node and review the properties provided for each solver.

    For models that solve transport equations using the finite-volume method, the *Under-relaxation Factor* is often a property that you can adjust to reduce solution times. However, you must make sure that transport equations are adequately converged by the end of the simulation.

Setting Up Physics

This section provides information on how to set up physics models in STAR-CCM+.

First, to represent the different materials present in your simulation, create physics continua. Then select the physics models for each physics continuum, and assign the physics continua to the appropriate regions in the CFD model. Afterward, proceed to set appropriate conditions and values for the boundaries in your model, and also implement changes to region or interface types.

Contents:

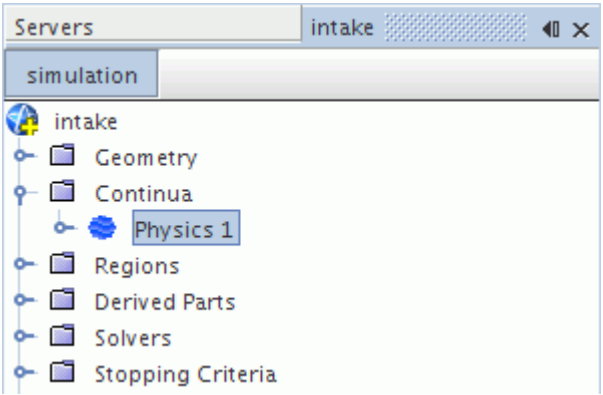
Using Physics Continua
Selecting Physics Models
Setting Types
Region Types Reference
Boundary Types Reference
Applying Quantities by Subgroup
Extensive and Intensive Quantities in Values

See Also:

Conditions, Values, and Profiles

Physics Continua

A physics continuum node, which has its own properties and pop-up menu, contains all of the model nodes in the object tree.



Generally, a physics continuum is defined as continuous and nonspatial such that no part of it is distinct or distinguishable from adjacent parts. This definition is also reflected in STAR-CCM+, which logically interprets a continuum as a collection of models that represents the substance (fluid or solid) being simulated.

In keeping with the nonspatial characteristic, you can define physics continua in STAR-CCM+ independently of the mesh and independently of existing regions. The models that are activated within a physics continuum determine the following:

- Whether the continuum is three-dimensional
- Whether it comprises a solid, liquid, or gas
- Whether it is stationary in time or transient
- The quantities that are defined within it

Certain models, when activated in a physics continuum, require the activation of other models in that continuum. For instance, once a physics continuum contains a liquid or a gas, it needs a flow model. Once it has a flow model, it needs a viscous model (inviscid, laminar, or turbulent). Once turbulence is activated within a fluid continuum, a turbulence model must be selected.

A simulation can contain many physics continua, each with its own name and properties. Each region must be associated with a physics continuum, and conversely each physics continuum can be associated with one or more regions.

In addition to the models comprising the physics continuum, the continuum also contains:

- Initial conditions
- Reference values

Copying Continua with Tables

STAR-CCM+ excludes an excessively large table from the process of copying a physics continuum; the rest of the continuum does get copied, however.

To copy a continuum with a table:

- Attempt to copy the continuum and paste it in its destination.
- If the table is too large, STAR-CCM+ generates a warning in the *Output* window.

```
Output - Star 3
/u/xeona38/people/ivana/.star-8.03.017/var/journal/star8614173457560781837.java
Loading module: MaterialModel
Loading module: StarReactions
Loading module: CombustionModel
Loading module: PvmModel
Reading material property database "/u/xeona38/people/ivana/STAR_C++/gitStar/star/props.mdb
Loading module: SegregatedFlowModel
Loading module: KeTurbModel
Loading module: RadiationCommon
Loading module: RadiationDoModel
WARNING: Ppdf Equilibrium table with dimensions [101, 98, 20] is too big and was not copied
 Please export the original table and then import it manually
WARNING: Ppdf Equilibrium table with dimensions [101, 98, 20] is too big and was not copied
 Please export the original table and then import it manually
```

- Export the table to a `.tbl` file using the right-click menu action.
- Import the table into the destination continuum.

Physics Continuum Properties

Interfaces

The interfaces that are associated with the continuum (read-only).

Regions

The regions that are associated with the continuum (read-only).

Active

When **Off**, halts solution updates for the continuum domain. The default is **On**.

Physics Continuum Pop-Up Menu

Select Models...

Allows selection of the models to be activated in this continuum.

See Also:

Using Initial Conditions

Conditions, Values, and Profiles

Selecting Physics Models

A physics model in STAR-CCM+ defines how a physical phenomenon in a continuum is represented.

Essentially, physics models define the primary variables of the simulation (such as pressure, temperature, and velocity) and what mathematical formulation is used to generate the solution. An appropriate combination of models is necessary for the complete definition of a physics continuum.

Models in STAR-CCM+ have varying levels of complexity and functionality, but their major purpose is to work with solvers to obtain a solution and to help present the information to you. The tasks of a typical model include making relevant field functions available and placing initial conditions and reference values in its continuum.

In situations where transport equations are required, the model provides these equations to the solvers. Therefore model properties often include coefficients relevant to the transport equations, as well as parameters relevant to the discretization approach.

In any continuum, models are selected and changed using the Physics Model Selection dialog, which assists you in recommending models and in selecting the appropriate combination of models.

Contents:

Using the Physics Model Selection Dialog
Selecting Models in the Dialog

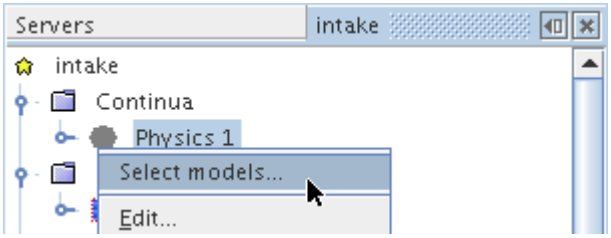
See Also:

Using Initial Conditions

Using the Physics Model Selection Dialog

The *Physics Model Selection* dialog is used to select physics models and to change the model selection in a simulation.

To activate the dialog in a physics continuum (say **Continuum 1**), open the **Continua** node and right-click on the **Physics 1** node. In the pop-up menu, select the item **Select models...**

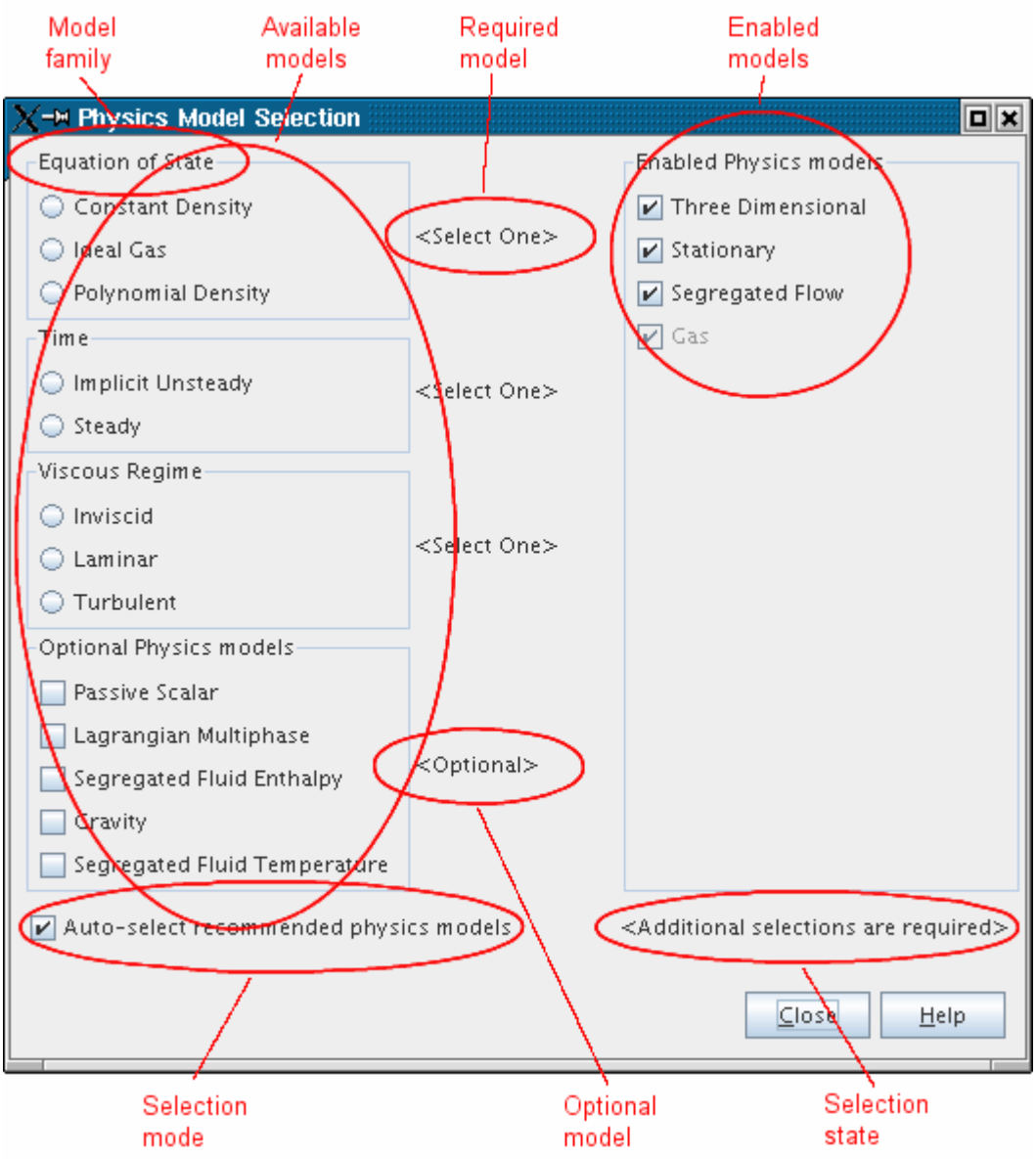


This same menu item is available in the pop-up menu of the **Models** manager node under the continuum node.

Model selection and deletion can only be done through this dialog:

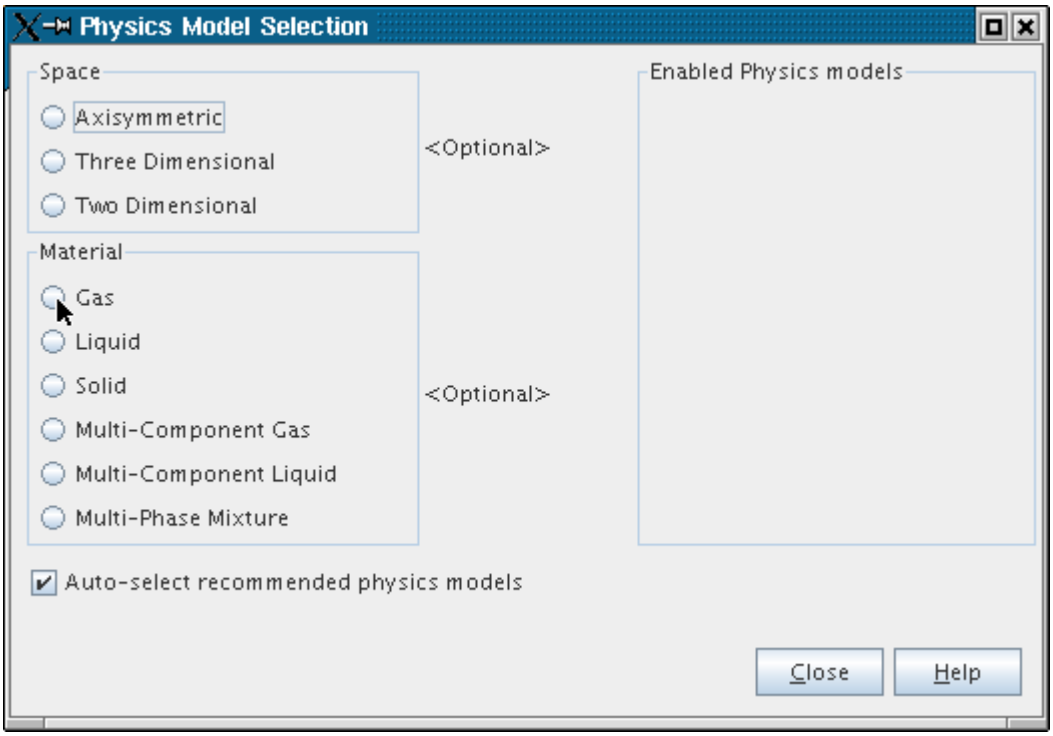
- This dialog is similar to a wizard in that it can assist you through a fairly complex process selection network.
- It is unlike a wizard in that it does not have the traditional **Next** and **Back** buttons.
- It does not restrict you to answering questions in a particular order.
- Models not available with STAR-CCM+ Lite are grayed out.

In this section and the following ones, the descriptions and illustrations of model selection refer to physics models; however, the general principles apply to meshing models as well. The following screenshot shows the dialog part way through a selection process, where each significant aspect of the dialog can be seen.



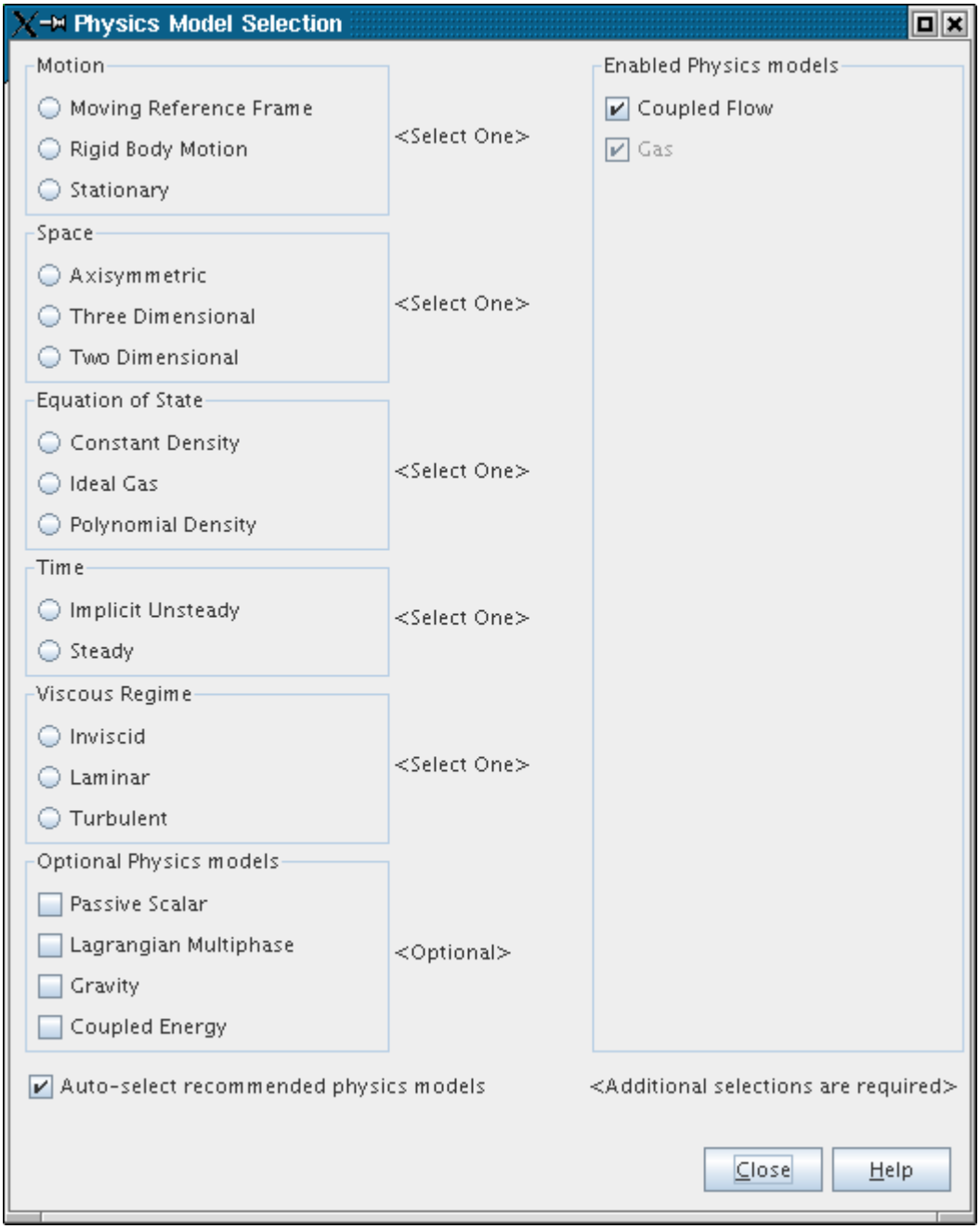
Selecting Models in the Dialog

The *Physics Model Selection* dialog is used to select physics models and to change the model selection in a simulation.

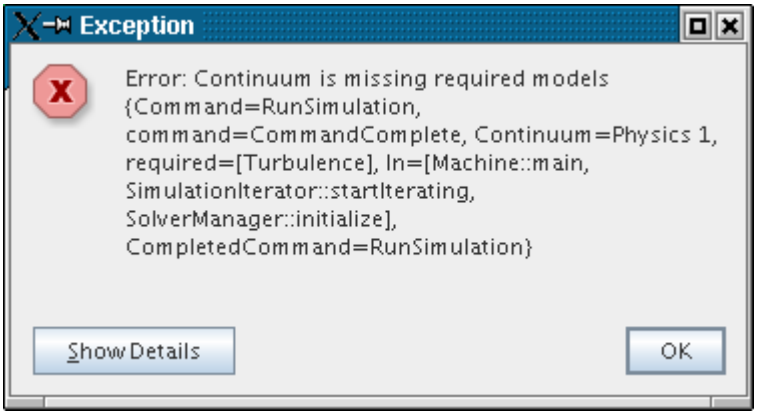


The *Enabled Models* group box on the right-hand side of the dialog is empty, indicating that no models have yet been selected. Two models are available for selection, the **Space** model and the **Material** model. The <Optional> labels indicate that the selection of these items is not mandatory. The **Auto-select recommended models** checkbox provides an option for the automatic selection of recommended models.

The *Model Selection* dialog changes dynamically in response to previous selections. For instance if the **Gas** and **Coupled Flow** model options are selected, it appears as below. Now, you are prompted to select models for motion, space, Equation of State and other categories. The <Additional selections are required> label tells you that the model selection is incomplete. An incomplete model selection means that the continuum cannot be initialized in its current state.



Trying to iterate when you have not completed the model selection results in an error similar to the one shown below.

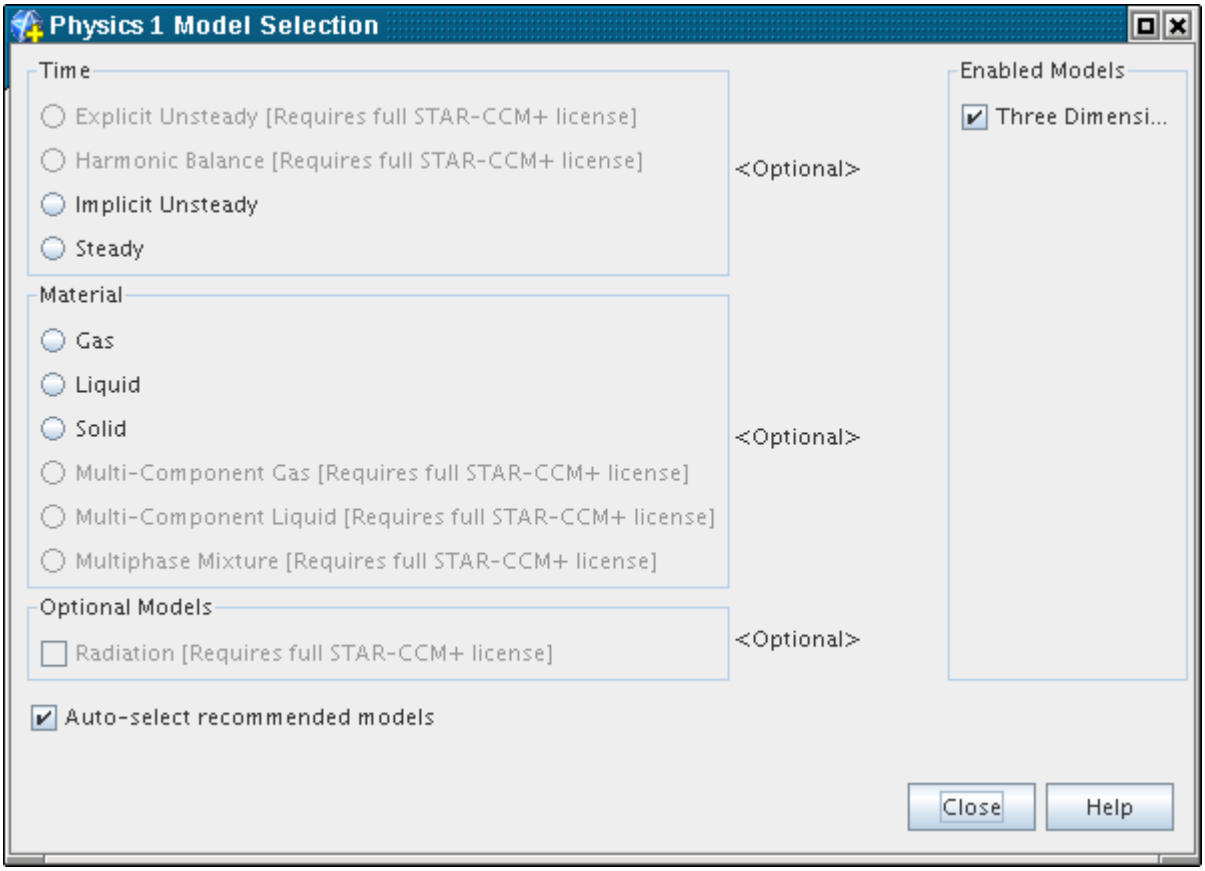


The most effective way to make model selections is to keep making choices from the model family that appears at the top of the available models side of the dialog. This technique adds models to the *Enabled Models* group box in the same order that they were selected (by you or by STAR-CCM+when in auto select mode).

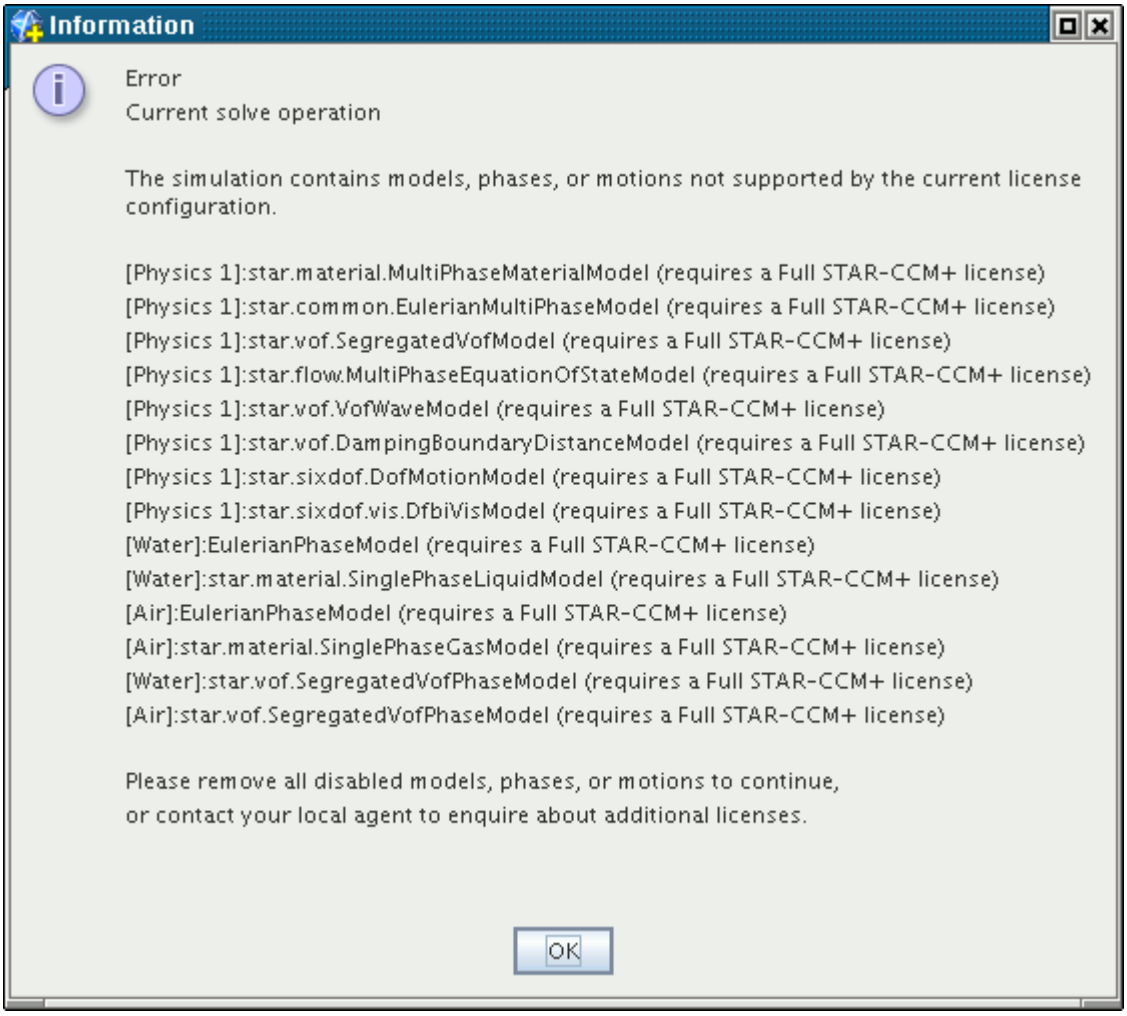
The dialog is “live” in that the server is updated each time that you change the state of a checkbox or radio button. If you have the relevant nodes open, the object tree reflects these updates.

Model Selection in STAR-CCM+ Lite

If you are running STAR-CCM+ Lite, some of the models are grayed out as shown below. These models are available for selection only in the full version of STAR-CCM+.



STAR-CCM+ Lite lets the *Physics Model Selection* dialog display any selected models, even ones that were activated in the full version of STAR-CCM+. However, STAR-CCM+ Lite cannot run the simulation. An *Information* dialog such as the one shown below appears when you attempt to run the case.


Information
Error
Current solve operation
The simulation contains models, phases, or motions not supported by the current license configuration.
[Physics 1]:star.material.MultiPhaseMaterialModel (requires a Full STAR-CCM+ license)
[Physics 1]:star.common.EulerianMultiPhaseModel (requires a Full STAR-CCM+ license)
[Physics 1]:star.vof.SegregatedVofModel (requires a Full STAR-CCM+ license)
[Physics 1]:star.flow.MultiPhaseEquationOfStateModel (requires a Full STAR-CCM+ license)
[Physics 1]:star.vof.VofWaveModel (requires a Full STAR-CCM+ license)
[Physics 1]:star.vof.DampingBoundaryDistanceModel (requires a Full STAR-CCM+ license)
[Physics 1]:star.sixdof.DofMotionModel (requires a Full STAR-CCM+ license)
[Physics 1]:star.sixdof.vis.DfbiVisModel (requires a Full STAR-CCM+ license)
[Water]:EulerianPhaseModel (requires a Full STAR-CCM+ license)
[Water]:star.material.SinglePhaseLiquidModel (requires a Full STAR-CCM+ license)
[Air]:EulerianPhaseModel (requires a Full STAR-CCM+ license)
[Air]:star.material.SinglePhaseGasModel (requires a Full STAR-CCM+ license)
[Water]:star.vof.SegregatedVofPhaseModel (requires a Full STAR-CCM+ license)
[Air]:star.vof.SegregatedVofPhaseModel (requires a Full STAR-CCM+ license)
Please remove all disabled models, phases, or motions to continue,
or contact your local agent to enquire about additional licenses.
OK

Setting Types

The simulation of continua requires a bridge between what is understood in the material world and the mathematical constructs of transport equation models. In the physical world, we understand concepts such as “walls” or “porous media”. In the modeling world, we think in terms of “boundary conditions” or “volumetric source terms”.

These concepts are bridged in STAR-CCM+ by formally defining types, conditions, and values in such a way that there is no ambiguity. These three entities are used to provide to the continuum models complete information for dealing with boundaries, regions, and interfaces.

This section includes information about:

- Setting types
- Specific region types
- Specific boundary types

Separate sections cover the following topics in detail:

- Conditions and values
- Specific interface types

Contents:

What Are Types?
Setting Types

What Are Types?

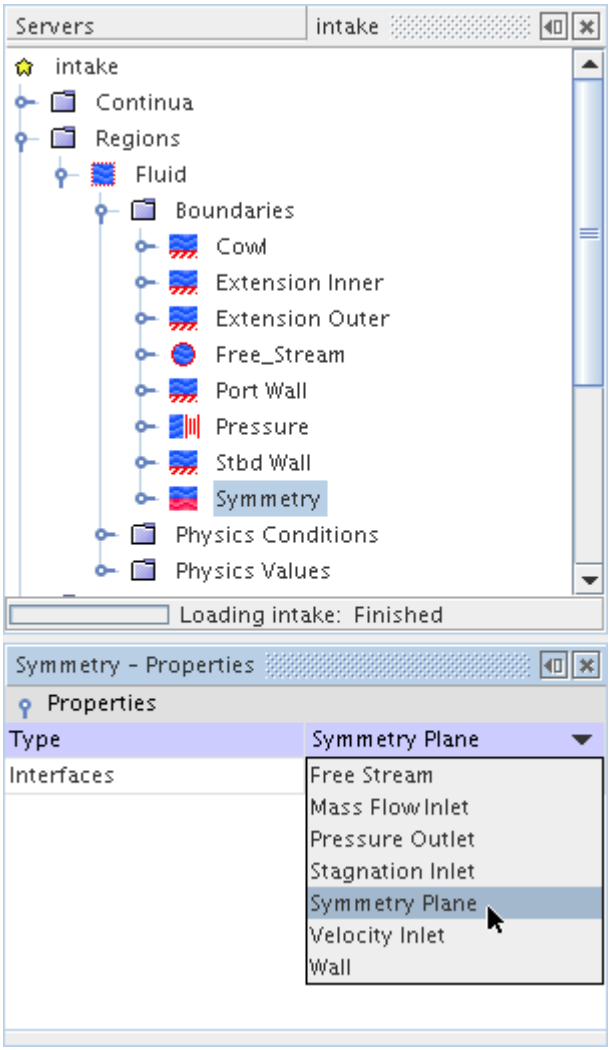
In STAR-CCM+, types are assigned to regions, boundaries, and interfaces as a way of informing the models of the associated continua how these entities must be dealt with.

As much as possible, the types are analogous to physical entities so that they are as unambiguous as possible. For instance, a typical boundary type is a wall, which is a concept that most people can understand and identify with. All models in STAR-CCM+ must know how to deal with the predefined types.

Setting Types

STAR-CCM+ initially assigns types during mesh import (based on information that is contained in the mesh being imported) or when a region or boundary is created from within the simulation.

To change the type of a region, boundary or interface, select the region, boundary, or interface in the object tree. In the *Properties* window, select one of the available types from the drop-down list corresponding to the *Type* property.



The models that are available in the continuum determine the list of available types. For instance, the **Free Stream** boundary type is only valid for ideal gas flow. So if the continuum does not contain the Ideal Gas Model, the **Free Stream** boundary type is not presented.

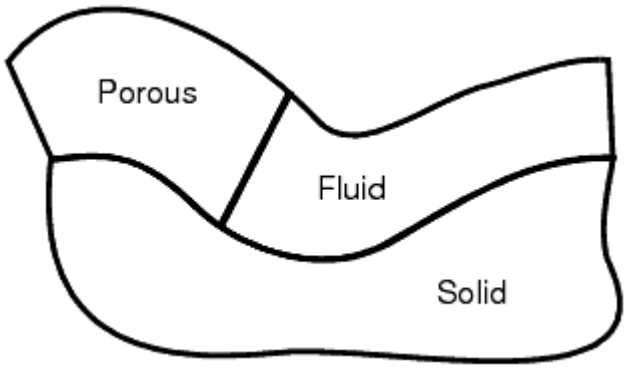
See Also:

Conditions, Values, and Profiles

Region Types Reference

Regions are assigned types that models within their associated continuum use to determine what specific conditions and values are required.

STAR-CCM+ recognizes three different region types: fluid, porous, and solid. The availability of each type depends on the continuum that is associated with the region. Fluid and porous regions are available only if the continuum is a gas or liquid. Solid regions are available only if the continuum is a solid.



STAR-CCM+ initially assigns regions types upon mesh import (based on information that is contained in the mesh being imported). Region types can also be reassigned within STAR-CCM+.

Contents:

Fluid
Porous
Solid

Fluid

Regions that are assigned to gas or liquid continua are automatically assigned the fluid type.

The physical models, such as flow, energy, and turbulence, make the appropriate modeling assumptions governing flow in these regions.

Application

- General fluid flow in the absence of porous media

Compatibility

- Available only for regions containing a fluid material

Porous

Regions that are assigned to gas or liquid continua can be designated with the porous type.

The physical models, such as flow, energy, and turbulence, make the appropriate modeling assumptions governing porous media flow in these regions.

Applications

- Filters
- Tube bundles
- Flow straighteners
- Sponges
- Radiators

Compatibility

- Available only for regions containing a fluid material

Solid

Regions that are assigned to solid continua are automatically assigned the solid type.

The physical models, such as energy and stress, make the appropriate modeling assumptions governing solids in these regions.

Applications

- Solid regions in which conductive heat transfer is to be modeled
- Solid regions in which stresses are to be modeled

Compatibility

- Available only for regions containing a solid material

Boundary Types Reference

Boundaries are assigned types that models within their associated continuum use to determine what conditions and values are required.

STAR-CCM+ recognizes nine different user selectable boundary types:

- Axis
- Flow split outlet
- Free-stream
- Mass flow inlet
- Pressure outlet
- Stagnation inlet
- Symmetry plane
- Velocity inlet
- Wall

The availability of these boundary types is dependent on the specific models that have been activated within the continuum that is associated with the boundary.

STAR-CCM+ initially assigns boundary types upon mesh import (based on information that is contained in the mesh being imported). These types can also be reassigned within STAR-CCM+.

Boundaries added through the creation of interfaces are assigned interface boundary types and names that you do not change.

Contents:

Axis
Flow Split Outlet
Free-Stream
Mass Flow Inlet
Pressure Outlet
Stagnation Inlet
Symmetry Plane
Velocity Inlet
Wall
Interface Boundary Types
Interface Boundary Names

Axis

The axis boundary represents the axis of a two-dimensional axisymmetric region.

Application

- The centerline of a pipe or other axisymmetric duct

Compatibility

- Available only for boundaries of regions containing the Axisymmetric model

Outlet

The outlet boundary represents the outlet of a duct.

When multiple outlets bound a fluid continuum, you specify the fraction of the total mass flow through each of the outlet boundaries. The sum of all flow split fractions from all of the outlet boundaries in a simulation must equal 1.0. When only one flow split boundary bounds the continuum, it is assumed that all the flow exits through this boundary.

The flow split can be specified by ratio or by mass flow rate. The pressure is not specified at an outlet.

For more information, see Outlet.

Application

Incompressible internal flows, with multiple outlets, for which the flow split between the outlets is specified.

Compatibility

- Should not be used with compressible flows
- Cannot exist in the same continuum as pressure, stagnation and mass flow boundaries, or mixing-plane interfaces
- Not suitable for outflow boundaries at which recirculation (inflow) occurs

Free-Stream

The free-stream boundary represents the boundary of a bubble of fluid surrounding an object flying through space.

The free-stream boundary condition is based on extrapolation of Riemann invariants under the assumption of irrotational, quasi-1D flow in the boundary-normal direction. This condition generally applies to external flows when the boundary is placed sufficiently far from the body. Internal flows often have walls, possibly of irregular shape, immediately next to the flow boundary. These walls can produce boundary layers, vortices, or other multi-dimensional flow structures such that the irrotational, quasi-1D flow assumption breaks down.

For more information, see Free-Stream.

Application

- The free-stream or far-field of an external aerodynamic simulation
- If using this condition for internal flows, position the flow boundary in such a way that the irrotational, quasi-1D flow assumption is valid.

Compatibility

- Available only for boundaries of regions in which the ideal gas model is activated

Mass Flow Inlet

A mass flow inlet boundary represents an inlet for which the mass flow rate is known.

For more information, see Mass Flow Inlet.

Application

The inlet to a simulation of compressible internal flows for which the mass flow rate is known

Compatibility

- Should be used in combination with pressure outlet boundaries

Pressure Outlet

A pressure outlet boundary is a flow outlet boundary at which the pressure is specified.

In a multiphase flow simulation, the Mach number determines whether the ambient pressure is used as a boundary condition (cell-wise). This condition means that the simulation can become unstable when the Mach number is close to 1 at the pressure outlet boundary. In a simulation that models trans-sonic multiphase flow, place the pressure outlet boundaries in such a way that the flow over them is either clearly subsonic or clearly supersonic. Ensure that the flow velocity at the pressure outlet boundary is not close to Mach 1.

For more information, see Pressure Outlet.

Applications

- Outflow of compressible and internal flows
- A plenum, for example, the boundary surrounding a jet emanating into a large chamber

Compatibility

- A pressure outlet boundary cannot exist in the same continuum as a flow split outlet, as a combination of pressure and flow split outlet boundaries results in an ill-posed problem.
- Although STAR-CCM+ does not prevent a pressure outlet boundary from being used as an inlet, a stagnation boundary is a better choice.

Stagnation Inlet

The stagnation inlet boundary is an inlet condition that is well-posed for compressible flows, although it is equally valid for incompressible flows.

The stagnation conditions refer to the conditions in an imaginary plenum, far upstream, in which the flow is completely at rest. For incompressible flows, Bernoulli's equation is used to relate total pressure, static pressure, and velocity magnitude. For compressible ideal-gas flows, isentropic relations are used, and

characteristic variables determine the propagation properties of the flow. In compressible flow, the total pressure, total temperature, and flow direction are always used to obtain the inlet flow conditions.

Supersonic Flow

Confusion often arises, however, about the need to specify the static pressure at the inlet, termed supersonic static pressure in STAR-CCM+. The characteristics of the flow dictate that when the inlet flow is supersonic, all information comes from upstream conditions, including static pressure. This flow regime is termed hyperbolic, and the situation dictates that supersonic static pressure must be appropriately specified. When the inlet flow is subsonic, the flow regime is termed elliptic, which means that the static pressure is obtained from the downstream flow. In this case, the value that is specified for supersonic static pressure is ignored, and ideally, its value could be left at the default. This is not always advisable, as in some situations the inlet flow can momentarily become supersonic during the simulation iteration. If this occurs, and the supersonic static pressure is improperly specified, the solution might diverge. In these situations, it is recommended that isentropic relations be used to specify a value for supersonic static pressure that corresponds to a slightly supersonic Mach number.

As an example of using isentropic relations to set the supersonic static pressure, consider the isentropic relation for the ratio of static pressure to total pressure in an ideal gas:

$$\frac{p_t}{p} = \left\{1 + \frac{(\gamma-1)}{2} M^2\right\}^{\gamma/(\gamma-1)} \tag{90}$$

In this relation, p and p_t are the absolute static pressure and absolute total pressure, respectively, and M is the Mach number. To obtain the static pressure, one could set $M = 1.02$ for a mildly supersonic flow and use the known total pressure to compute the required value.

For more information, see Stagnation Inlet.

Multiphase Flow

The formulation for multiphase flow is not as well-defined as for single phase flow, as the isentropic relations that relate pressure to temperature do not hold for a multiphase mixture. The same theory is extended to compressible flow, but is expected to break down at high speeds.

The stagnation flow inlet allows you to specify the total pressure of the mixture with the volume fraction per phase at the inlet boundary. For multiphase flow, the total pressure is defined as the sum of the static pressure with the dynamic pressure of each phase weighted with the volume fraction:

$$p_t = p_s + \sum_{k=1}^{n} \frac{1}{2}\alpha_k \rho_k v_k^2 \tag{91}$$

where α_k is the k-phase volume fraction, ρ_k is the phase density, and v_k is the velocity of the k-phase. This relation is exact for constant density and approximate for compressible and non-isothermal cases.

Applications

The inlet to a simulation of internal flows for which the stagnation conditions are known.

Compatibility

- Should be used in combination with pressure outlet boundaries

- A stagnation boundary cannot exist in the same continuum as a flow split outlet boundary.

See Also:

Total Temperature

Symmetry Plane

A symmetry plane boundary represents an imaginary plane of symmetry in the simulation.

The solution that is obtained with a symmetry plane boundary is identical to the solution that would be obtained by mirroring the mesh about the symmetry plane (in half the resulting domain).

For more information, see Symmetry Plane.

Application

An imaginary plane of symmetry in a simulation that would be physically symmetrical if modeled in its entirety.

Velocity Inlet

A velocity inlet boundary represents the inlet of a duct at which the flow velocity is known.

For more information, see Velocity Inlet.

Application

The inlet to a simulation of incompressible internal flows.

Compatibility

- Can be used in combination with pressure outlet and flow split outlet boundaries
- Should not be used for compressible flow as this boundary does not prescribe the total energy of the system--which is important for compressible flow. (STAR-CCM+ does not, however, specifically prevent the use of velocity inlet boundaries for compressible flows).

Wall

A wall boundary represents an impermeable surface.

For more information, see Wall.

Applications

- An impenetrable boundary for inviscid flows
- An impenetrable, no-slip boundary for viscous flow simulations
- “Slip walls” representing an impenetrable but traction-free surface
- The boundary of a thermally active solid region at which the temperature or heat flux is specified

Interface Boundary Types

When a new interface is successfully intersected, new boundaries are created, appear in the simulation tree, and have types that are associated with them defined by the type of the interface. These boundary types cannot be changed except by changing the interface type.

The possible generated boundary types are:

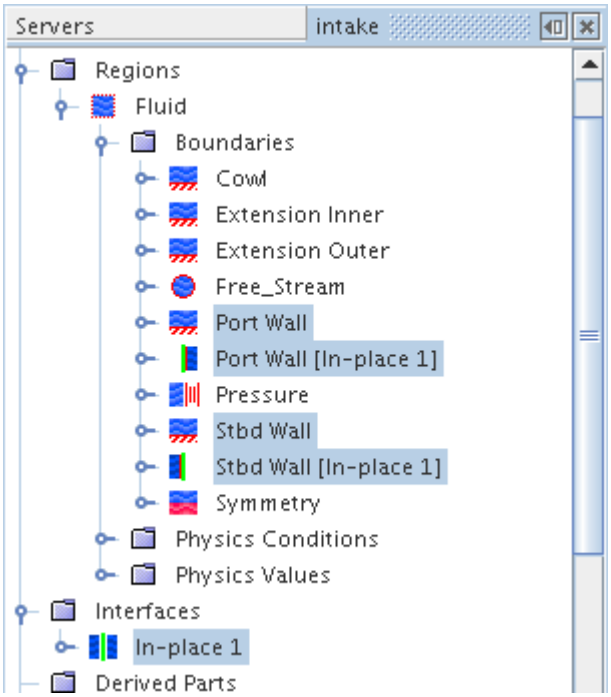
- Baffle boundary
- Contact interface boundary
- Fan interface boundary
- Fully developed interface boundary
- Internal interface boundary
- Mixing plane inflow boundary
- Mixing plane outflow boundary
- Porous baffle boundary

In addition, the name of the interface boundary cannot be changed except by changing the names of the original boundary and the interface.

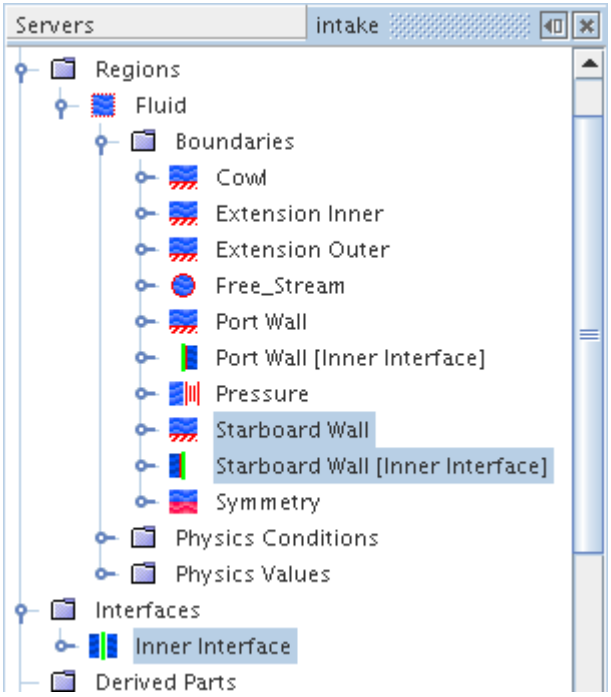
Interface Boundary Names

Unlike the names of other boundary types, you cannot directly change the interface boundary names. STAR-CCM+ constructs them. The rule for these names is `<originalboundary> [<interface>]`.

For example, in the following screenshot, the selected nodes are of an interface, the original boundaries of that interface and the interface-type boundaries that were created along with the interface. The default name of each interface-type boundary contains the default names of the original boundary in its respective region and of the interface.



When the nodes of the interface and/or either original boundary is renamed, that name change is reflected in the corresponding nodes of the interface-type boundaries.



Applying Quantities by Subgroup

For region objects (that is, region, boundaries, and interfaces) that address multiple parts, you can apply quantities to subgroups based on parts within the object. For example, if you have a region with 10 parts, you can apply one amount of heat source to 4 parts, and another amount to the rest. This subgrouping facility extends to other features including parts-based reports, parts-based visualization, and part-based materials (multi-part solids).

The benefits of applying parts-based subgroups are:

- Improved performance and scalability. Using parts-based subgroups allows you to reduce the number of regions and boundaries in the simulation. In some cases you can reduce the number of regions and boundaries by a factor of 100 or more. This reduction generally improves both performance and scalability, with the improvement generally being proportional to the reduction in the number of regions and boundaries.
- Specification of physics using familiar geometric components – parts and part surfaces – used in parts-based meshing and/or in imported CAD.
- Simplified physics specification. Parts-based subgroups often reduce the number of regions and boundaries required to specify the problem. This in turn often simplifies workflow and eliminates redundant input.

Parts-based subgroups require that before you generate the mesh, you associate geometry parts with regions, part surfaces with boundaries, and part contacts with interfaces. Each geometry part is associated with only one region (multiple parts per region); each part surface is associated with only one boundary (multiple surfaces per boundary); and each part contact is associated with only one interface (multiple part contacts per interface). Maximum benefit applies when multiple parts are assigned to a region, multiple part surfaces to a boundary, and multiple part contacts to an interface.

The overall strategy is to reduce the number of region objects present in the simulation so that they represent domains of common physics rather than single geometry parts. For instance, a particular region can include all the geometry parts that are solid and have common physics (such as conduction), while a particular boundary can include all the part surfaces that have a common set of boundary condition types.

Do not confuse the boundary condition types that are specified under **Physics Conditions** (such as **Thermal Specification**) with the values for the boundary conditions that are specified under **Physics Values**. The physics conditions are the same for the entire boundary (for example, the whole boundary has the same thermal specification such as a fixed temperature specification). However, with the use of parts-based subgroups, the physics values can vary across the boundary down to the level of granularity of the part surfaces nested within that boundary. For example, each part surface could have a different temperature value.

Provided the geometry layout defines the assignment of geometry objects to their corresponding region objects, parts-based subgroups can be applied within regions, boundaries, and interfaces.

General Steps

The steps in using parts-based subgroups are:

1. Apply parts-based meshing.

Associate geometry parts with regions, part surfaces with boundaries, and part contacts with interfaces, and then mesh with parts-based meshing. Minimize the number of region objects that the model requires by grouping parts, part surfaces, and part interfaces according to the physics you expect to apply.

2. Divide the part collections nested in region objects into divisions that have common physics values.

   Each division is known as a Part Subgrouping and is composed of a number of Subgroups.

   Every part surface nested in a boundary belongs to only one part Subgroup in a given Subgrouping.

   For instance, the part surfaces in a boundary can be grouped into subsets that each have common a boundary temperature or emissivity specification.

3. Assign values to the various subgroups that are defined in the previous step.

Details

You can apply values to selected parts and collections of parts within a region or part surfaces within a boundary. The steps described here apply to regions, but analogous steps apply to boundaries.

To define part Subgroups to which profiles can be applied:

1. Select a region, interface, or boundary and activate the *Allow Per-[Object] Values* property. This action creates a **Part Subgroupings** node under the parent node, and creates a default subgrouping called **Subgrouping 1** under that node. The default subgrouping contains one subgroup that includes all the parts in the region.
2. To create additional subgroupings, select the **[object] Subgroupings** node, right-click, and select **New**.

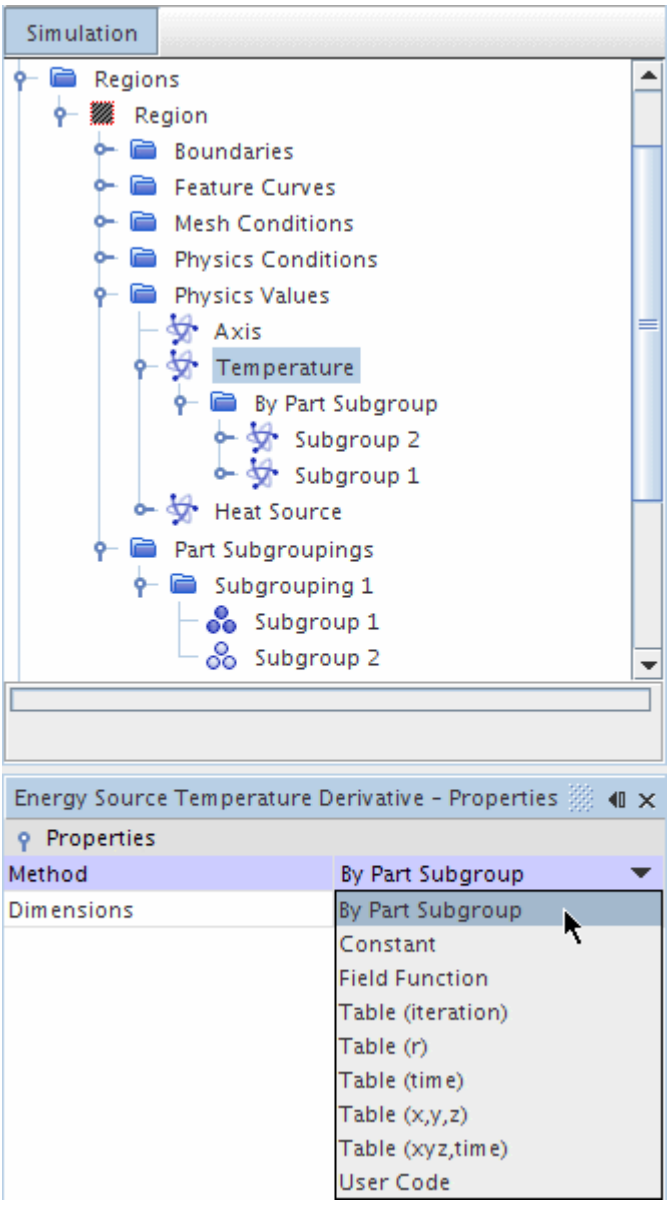
   This action creates **Subgrouping 2**, **Subgrouping 3**, and so forth.
3. To create more subgroups after the initial default (**Subgroup 1**), select a subgrouping node, right-click, and select **New**. This action creates a **Subgroup 2** node. Repeat to create **Subgroup 3**, **Subgroup 4**, and so forth.
   - You can rename subgrouping nodes and subgroup nodes, including the default nodes.
   - You can delete subgrouping nodes and subgroup nodes until there is only one remaining. The last node cannot be deleted.
   - You can reassign which subgroup within a subgrouping is the default group: edit the *Default Subgroup* property of the subgrouping.

   Use multiple subgroups only to apply different quantities to different collections of parts. For example, take a case where volumetric heat source has a unique value in parts A and B, but porosity has a unique value in parts B and C. One subgrouping can be defined for use with volumetric heat source that has a subgroup with parts A and B, and another subgrouping can be defined for use with porosity that has a subgroup with parts B and C.
4. To assign or unassign parts to a subgroup, edit the *Objects* property of the subgroup.
5. When you unassign a part, it returns to the default subgroup of the subgrouping. Trying to unassign a part from the default subgroup fails and generates a warning message. When the *Allow Per-Part Values* property is activated, every part is always in exactly one subgroup in each subgrouping. Each subgrouping contains all the parts that are assigned to the region.

To assign quantities to subgroups:

1. Under the **Physics Values** node, select the physics value that you want to set by part subgrouping, for example **Physics Values** > **Temperature**.

2. Set the *Method* property to **By [part-object] Subgroup**. This action creates a **By [part-object] Subgroup** node under the **Physics Value** node. Under **By [part-object] Subgroup** are profile nodes for every subgroup in the first subgrouping.
3. To change the subgrouping that is assigned to the **By [part-object] Subgroup** node, edit the *Subgrouping* property of the **By [part-object] Subgroup** node.
4. Select a value node under the **By [part-object] Subgroup** node and edit the *Method* property.



Extensive and Intensive Quantities in Values

When assigning Physics Values, it is important to remember which quantities are extensive and which are intensive.

Extensive quantities apply to an entire geometrical object. (Example objects are regions, boundaries, parts, or subgroups of parts (see Quantities That Apply to a Part).)

Intensive quantities apply per unit of the object.

For example:

- A mass flow rate of 5 kg/s through an inlet is an extensive quantity, because it applies to (*extends* over) the entire inlet.
- A mass flux of 10 g/cm^2-s through an inlet is an intensive quantity, because it applies to each unit, each square centimeter, of the inlet area.

Extensive and intensive quantities can be two different ways of quantifying the same thing. Continuing the inlet example, if you give the mass flux through the inlet in an extensive form, the simulation can then calculates the intensive form, that is, the mass flux through each square centimeter of inlet area. From there, the simulation can calculate the mass flow through each cell face across the inlet.

Other examples of extensive quantities are Total Heat Source and Force. The equivalent intensive quantities are Heat Flux and Pressure.

Running Physics Simulations

This part of the documentation describes preparation and procedures for running a STAR-CCM+ simulation.

Contents:

Using Initial Conditions
Using Solvers
Setting Up Stopping Criteria
Obtaining a Solution Interactively
Troubleshooting the Solution

Using Initial Conditions

Initial conditions in a physics continuum specify the initial field data for the simulation.

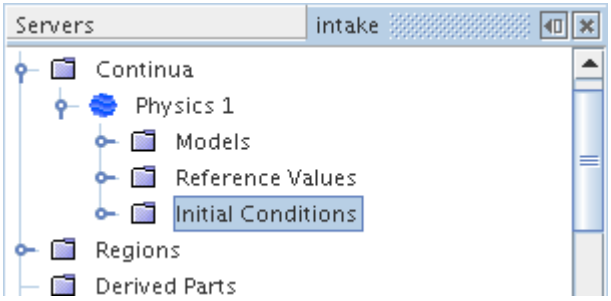
For steady-state simulations, the converged solution should be independent of the initial field. However, the path to convergence, and hence the computational effort that is required to reach convergence, is affected. Therefore, choose the initial conditions and values judiciously, particularly when the physics is complex.

Each model requires sufficient information for the primary solution data for the primary variables that are associated with the model to be set. In most cases, the initial conditions are set up directly. For some models, such as turbulence, there is the option of specifying the information in a more convenient form (turbulence intensity and turbulent viscosity ratio instead of turbulent kinetic energy and turbulent dissipation rate).

Examples of initial conditions are:

- Pressure
- Temperature
- Velocity components
- Turbulence quantities

The **Initial Conditions** manager node, which is subject to common menu items, contains the initial condition nodes for the continuum.



Contents:

Setting Initial Conditions
Setting Initial Conditions for a Particular Region

See Also:

Physics Continua
Selecting Physics Models
Troubleshooting the Solution
Getting Past a Poor Initial Solution

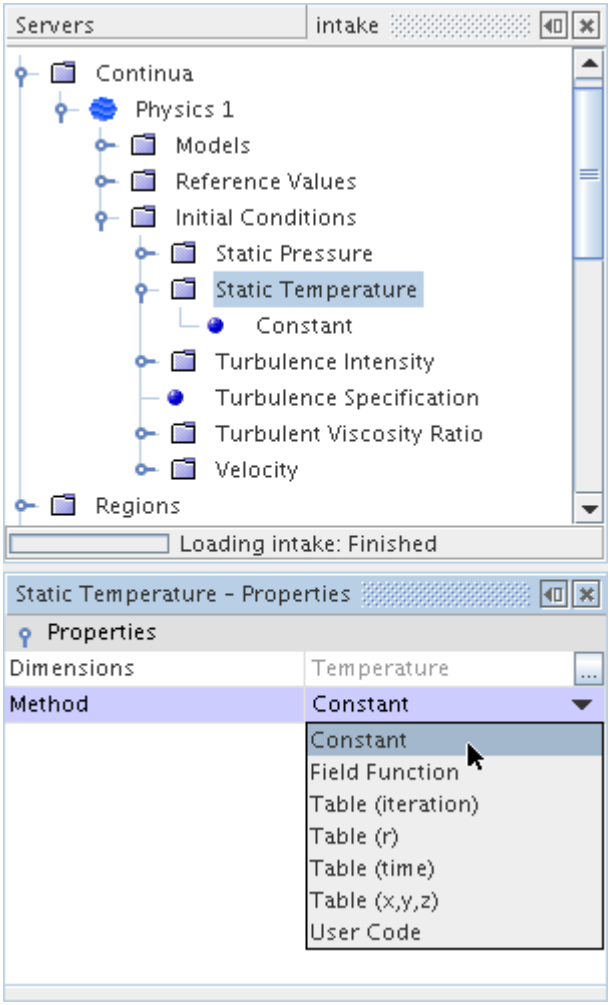
Setting Initial Conditions

Initial conditions are set for a continuum. They can be set with constant values, field function values, or tabular data values.

Initial conditions can also be set for a particular region.

To specify the method to use for setting an initial condition:

- Under the **Initial Conditions** node, select the node for the relevant condition.
- In the *Properties* window, click the drop-down list corresponding to the *Method* property.



This action permits the selection of one of the following:

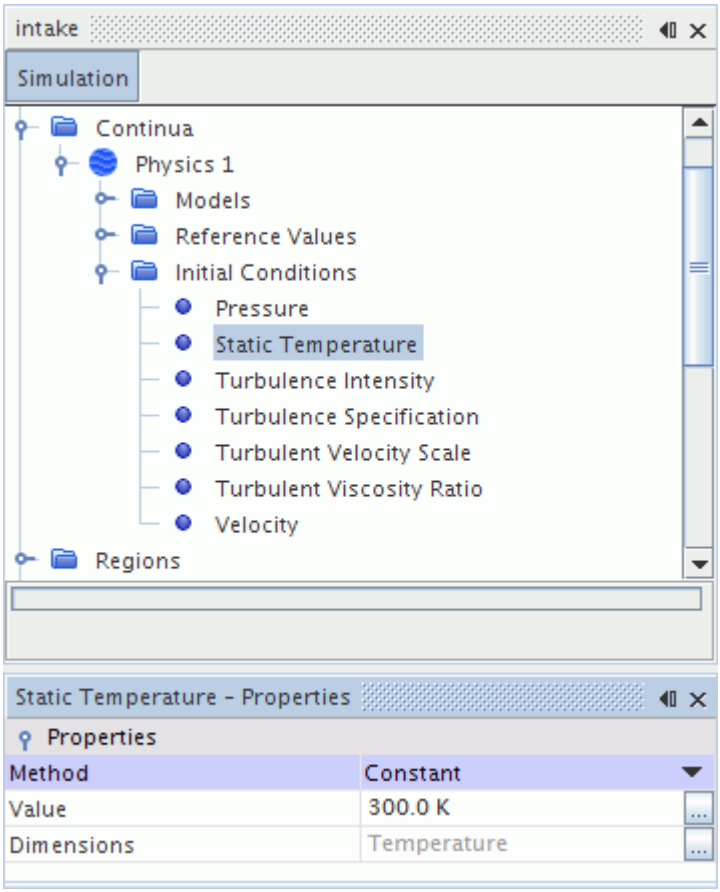
- **Constant**
- **Field Function**
- **Table (iteration)**
- **Table (r)**
- **Table (time)**
- **Table (x,y,z)**

- **User Code**

Using Constant Values

Generally, most initial conditions are specified as constants.

This value is entered as either a scalar or vector value, depending on the specific initial condition.



Using Field Function Values

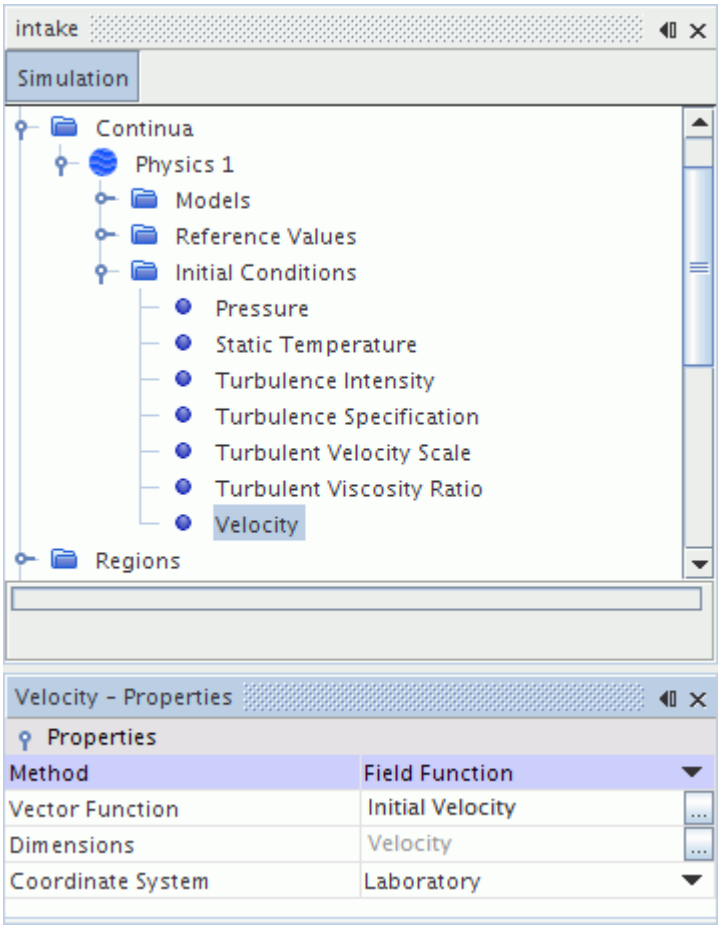
Field functions are useful for specifying initial conditions that are not constant values.

For instance, if you initialized the velocity in the L-shaped duct that is shown schematically below:

An initial velocity of 0 m/s is acceptable. Although, a better initial condition would be 1 m/s in the x-direction in the first part of the duct and 1 m/s in the -y direction after the bend. To set up this scenario, use a vector field function which could be defined as follows:

```
[($$Centroid[0] < 5) ? 1 : 0, ($$Centroid[0] < 5) ? 0 : -1, 0]
```

Suppose that this field function was named **Initial Velocity**. This field function would then be selected in the list corresponding to the *Vector function* property in the *Properties* window.



Although the above example was for a vector value, and thus required a vector field function, the process would be identical for scalar values using scalar field functions.

Take care to use a field function that is likely to be initialized before being used. For instance, problems are likely to occur if you initialize pressure in terms of velocity, since it is not clear that velocity is available at the time that pressure is initialized.

Using Tabular Data Values

Tabular data sets are useful for transferring the data from one solution to use as an initial condition for another solution.

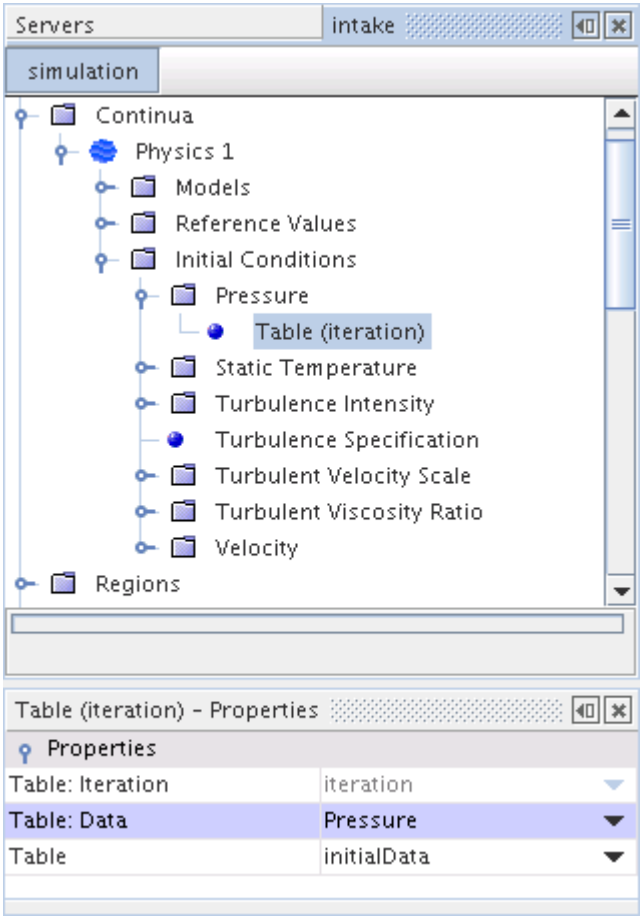
Since the tabular data file format is not compact, this method is not a suitable method of data transfer between large simulations. However, this method is effective for transferring a coarse-mesh solution to a finer mesh.

If you were to do the following:

- Extract a tabular data set containing pressure from a previous simulation.
- Export the data set to a file called `initialData`.

- Read the tabular data file in the current simulation.
- Assign the name of this tabular data set to the *Table* property in the *Properties* window of the table node (**Table (iteration)** in the following screenshot) under **Pressure**.

Then the field in the tabular data corresponding to pressure would also be assigned to the *Data* property.



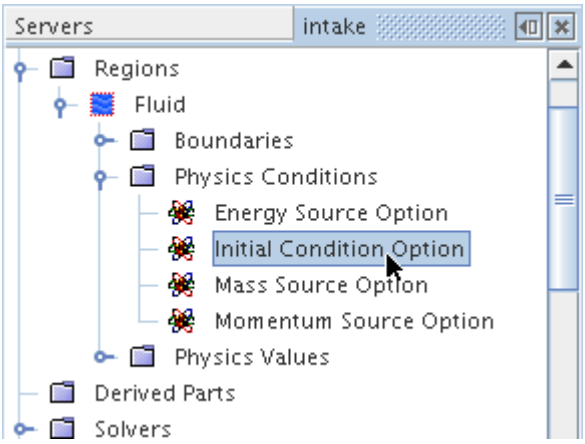
The tabular data set is not partitioned for a parallel case which means that the entire data set is loaded into each parallel process that is allocated to the simulation. This process could result in memory overflow, as multiple copies of the data set are created. For this reason, initialization using tables is not recommended for parallel runs.

Setting Initial Conditions for a Particular Region

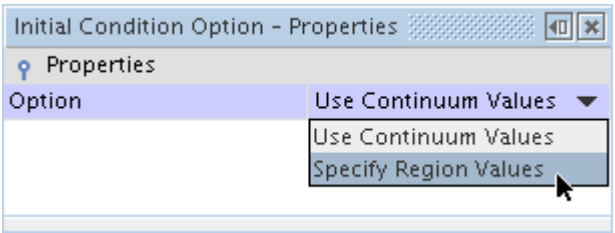
By default, the initial conditions for every region in a simulation come from the settings in the continuum. However, you can customize initial conditions for an individual region.

To customize initial conditions:

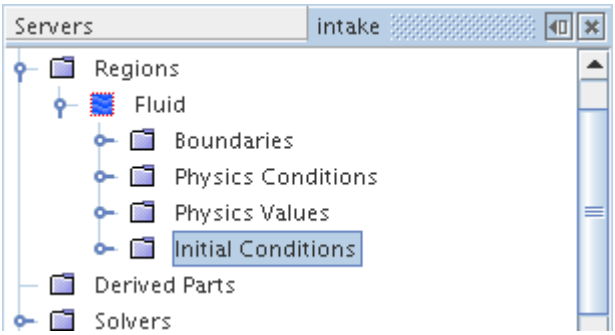
1. Select the **Regions** > **[region]** > **Physics Conditions** > **Initial Condition Option** node.



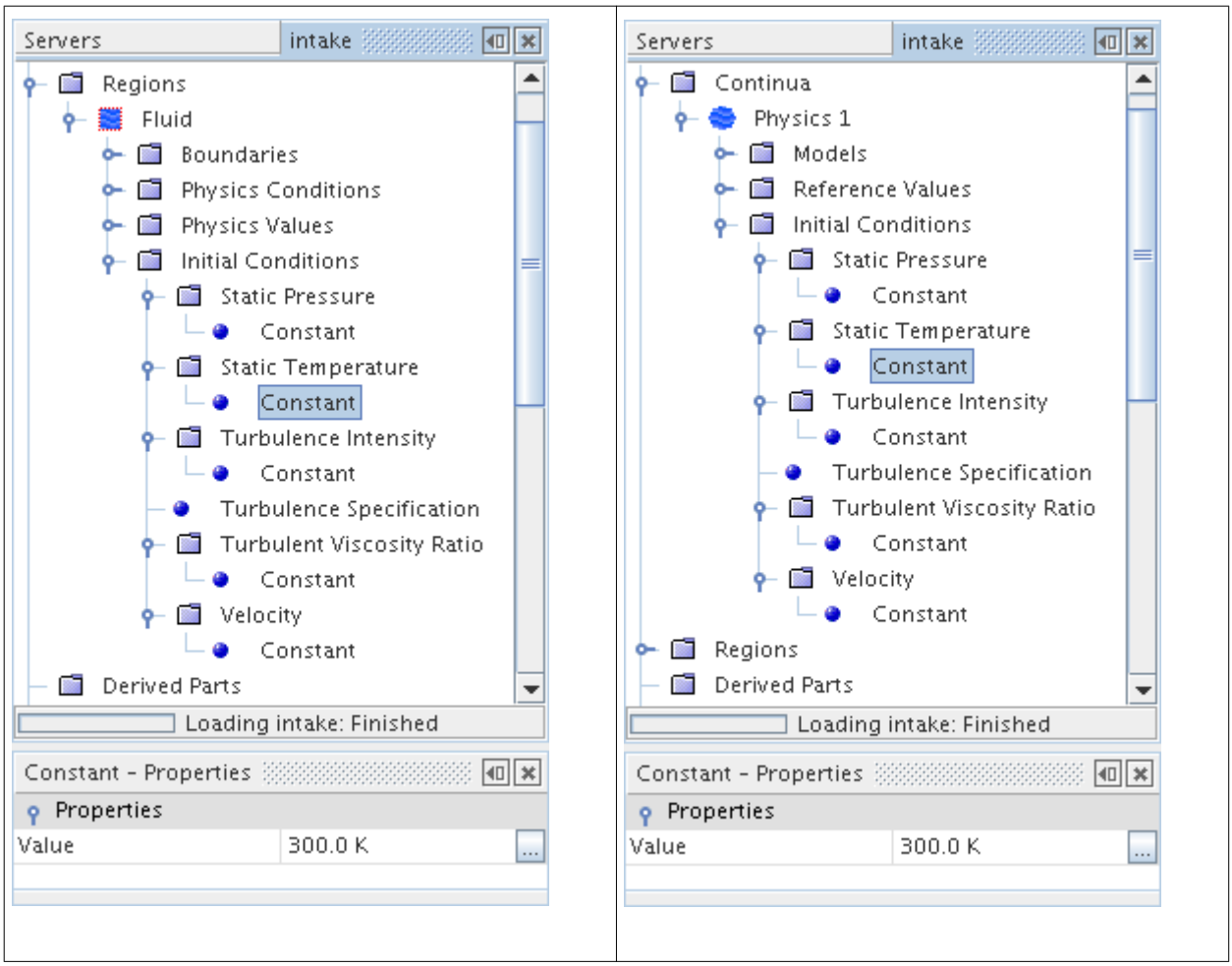
2. In the Properties window, set *Option* to `Specify Region Values`.



A new **Initial Conditions** manager node appears within the region node.



Its sub-nodes, and their default values, appear identical to the STAR-CCM+ defaults of the **Initial Conditions** manager node in the continuum.



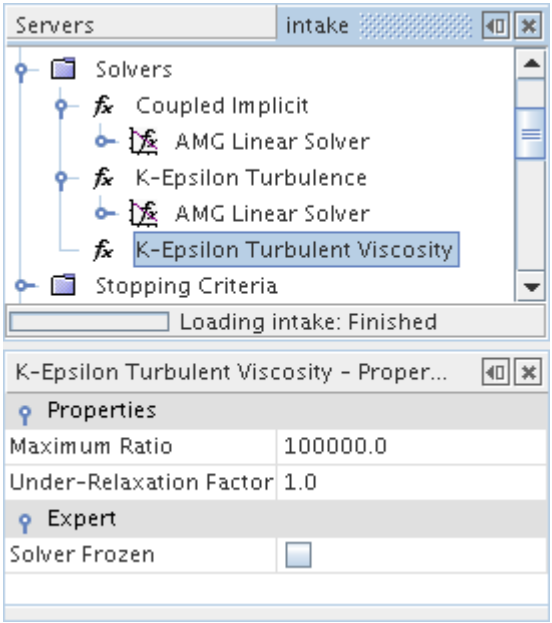
The initial condition nodes of the region, rather than the ones in the continuum, now control the initial condition values for this region.

The initial condition settings in the continuum still affect any other regions in the simulation that have not been changed in this way. However, any changes you make to the initial conditions of the continuum, either before or after you add initial condition objects to a region, will not affect the initial conditions of the region.

Using Solvers

Most solvers that are seen within the **Solvers** node of a simulation tree are closely related to the models that are selected in a continuum.

To access solver properties, select the node for that solver in the **Solvers** node as shown below.



When solver nodes are opened, they contain algebraic multigrid solvers. Selecting the **AMG Linear Solver** node allows you to adjust the multigrid parameters.

The solver parameters that most commonly need to be adjusted are as follows:

- The Coupled Implicit solver parameters and Coupled Explicit solver parameters for the Coupled Flow model and Coupled Energy model.
- The solver parameters for the pressure solver and the velocity solver of the Segregated Flow model.
- The solver parameters for the energy solver that the Segregated Fluid Energy model uses.

See Also:

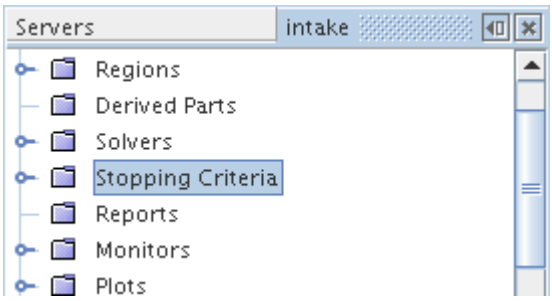
What Are Simulations?
Working with the Object Tree

Setting Up Stopping Criteria

Stopping criteria allow you to specify how long the solution runs for and under what conditions it stops iterating and/or marching in time.

Each specified stopping criterion is evaluated at the completion of every simulation step and a logical rule is used to determine if the interaction of all of the criteria stops the solver.

The **Stopping Criteria** manager node, which has its own pop-up menu, contains all of the stopping criterion nodes in the simulation tree.



Pop-Up Menu of the Stopping Criteria Manager

The following are the pop-up menu items that are available when you right-click the **Stopping Criteria** manager node.

Create New Criterion	Creates a new stopping criterion of the type that is selected in the submenu.	
	Create Stopping Criterion from Monitor	Creates a stopping criterion that is based on any one of the monitors available in the monitor manager.
	Minimum Inner Iterations	Creates a Minimum Inner Iterations stopping criterion (available in transient simulations only).
Enable All Criteria	Activates *Enabled* for all stopping criteria in the stopping criteria manager. Activating this property is the equivalent of visiting each stopping criterion and activating the *Enabled* property.	
Disable All Criteria	Deactivates *Enabled* for all stopping criteria in the stopping criteria manager. Deactivating this property is the equivalent of visiting each stopping criterion and deactivating the *Enabled* property.	

Contents:

Using Automatically Generated Stopping Criteria
Using Manually Generated Stopping Criteria

Using Monitors as Stopping Criteria
Using Multiple Stopping Criteria

See Also:

What Are Simulations?
Working with the Object Tree

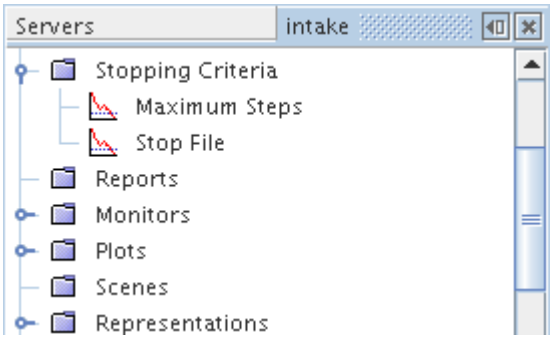
Using Automatically Generated Stopping Criteria

Certain stopping criteria are generated automatically when a solver is chosen.

To view the available criteria, expand the **Stopping Criteria** node.

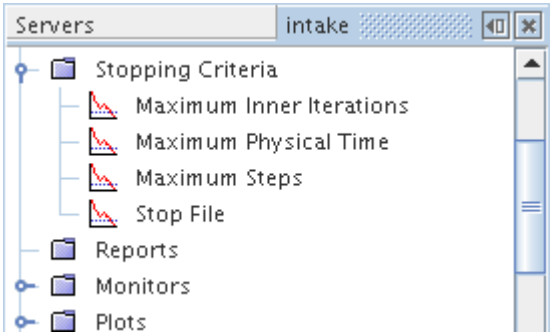
For steady simulations, there are two automatically-generated stopping criteria:

- Maximum Steps
- Stop File



For unsteady simulations, there are four automatically-generated stopping criteria:

- Maximum Inner Iterations
- Maximum Physical Time
- Maximum Steps
- Stop File



Automatically generated stopping criteria cannot be deleted, but the *Enabled* property can be activated or deactivated.

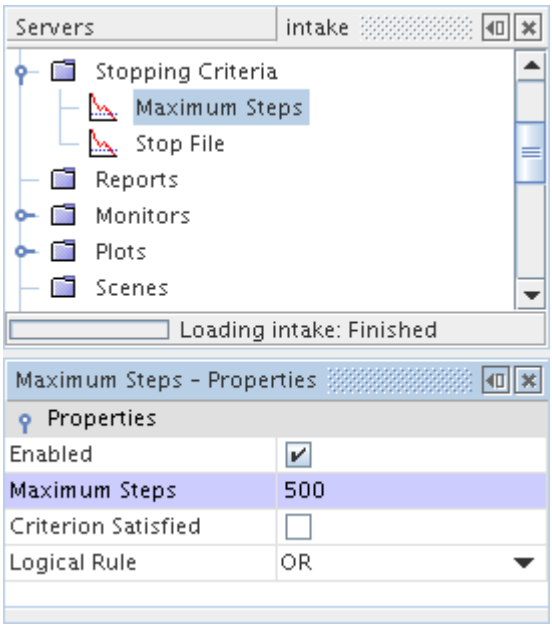
Contents:

Setting the Maximum Number of Steps
Setting a Stop File
Setting the Maximum Physical Time
Setting the Maximum Number of Inner Iterations
Disabling/Enabling Stopping Criteria

Setting the Maximum Number of Steps

The **Maximum Steps** stopping criterion allows you to specify the maximum number of iterations in a steady solver or the maximum number of time-steps in an unsteady solver.

The node of this criterion has its own properties, and is added automatically to the object tree by selecting any of the *Time* models (explicit unsteady, implicit unsteady, or steady).



This criterion cannot be manually deleted.

The stopping decision is based on the number of steps that the solver executes, including any steps that are executed in a previous session. If you Clear the solution, the counter resets to zero, if you initialize the solution it does not.

To change the number of steps, click the **Maximum Steps** criteria node and modify the value of the *Maximum Steps* property in the *Properties* window.

Properties

Enabled	Determines if this criterion is used for stopping the solver.	
	Activated	Use this criterion.
	Deactivated	Do not use this criterion.
Maximum Steps	The maximum number of steps to take before this criterion is satisfied. The default is 1000 steps.	
Criterion Satisfied	Indicates if the stopping criterion has been satisfied (read-only).	
Logical Rule	Defines how this criterion interacts with other stopping criteria that have been enabled.	
	AND	Requires that this criterion, along with one or more other criteria with the *Logical Rule* set to **AND**, all be satisfied before stopping the solver.
	OR	Indicates that satisfying only this criterion is sufficient to stop the solver.

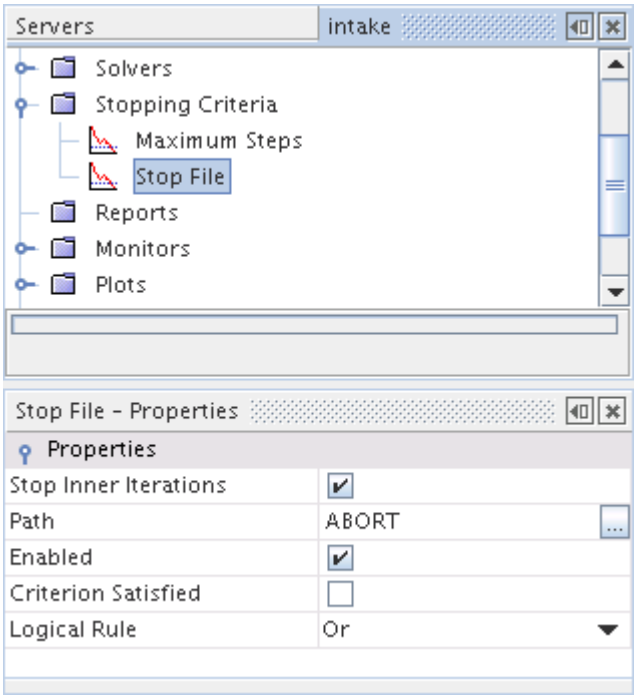
See Also:

Using Solvers
Clearing the Solution
Initializing the Solution
Using Multiple Stopping Criteria
Common Pop-Up Menu Items

Setting a Stop File

The **Stop File** criterion allows you to specify the pathname of a file (named `ABORT` by default) that, once in place, causes the solver to stop.

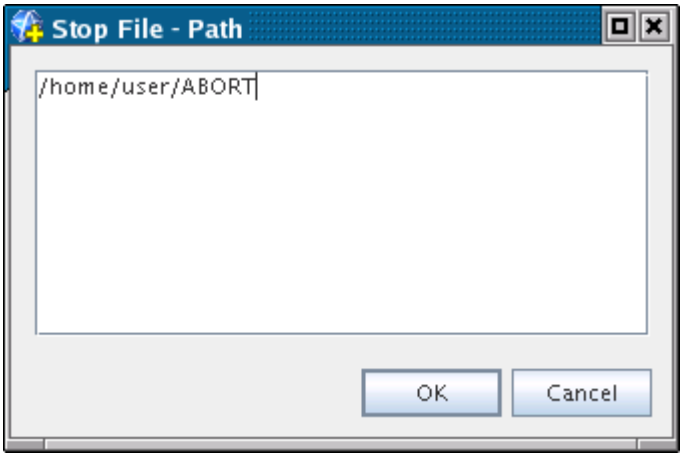
This action can be useful for stopping a batch job, for example. The node of this criterion has its own properties, and is added automatically to the tree by selecting any of the *Time* models (explicit unsteady, implicit unsteady, or steady).



Note: Activating this criterion does not stop the run immediately because STAR-CCM+ checks once every 60 seconds for the stop file.

This criterion cannot be manually deleted.

To set the path name of the file, click the **Stop File** criterion node and modify the name and the path in the *Path* property. You can either modify the name directly with a text entry or click the ellipses button and enter the path in the *Stop File - Path* dialog.



To use this feature while a simulation is running, you can simply add a file as follows:

- On Windows, place a file with the correct name in the correct path; even an empty text file would suffice for this purpose.

- On Linux, use the line command:

```
% touch ABORT
```

Properties

Stop Inner Iterations	Controls if this criterion responds to the completion of an inner iteration or a time-step.	
	Activated	Stops after the current internal iteration.
	Deactivated	Stops when the current time-step is finished (when the appropriate number of internal iterations are complete).
Path	The fully qualified file name of the stop file, `ABORT` by default.	
Enabled	Determines if this criterion is used for stopping the solver.	
	Activated	Use this criterion.
	Deactivated	Do not use this criterion.
Criterion Satisfied	Indicates if the stopping criterion has been satisfied (read-only).	
Logical Rule	Defines how this criterion interacts with other stopping criteria that have been enabled.	
	AND	Requires that this criterion, along with one or more other criteria with the *Logical Rule* set to **AND**, all be satisfied before stopping the solver.
	OR	Indicates that satisfying only this criterion is sufficient to stop the solver.

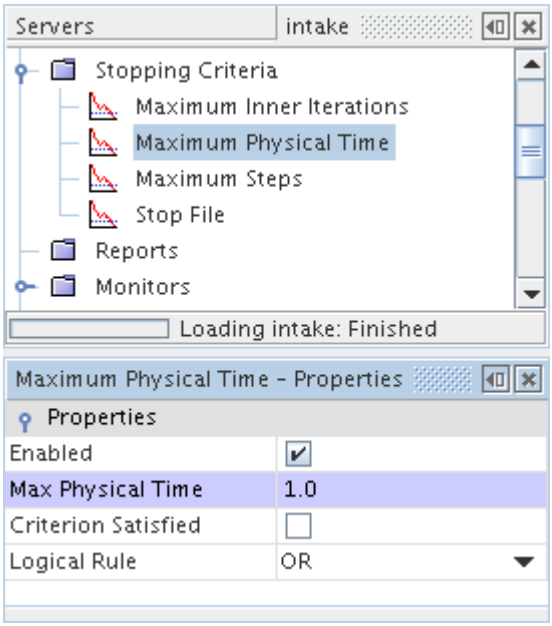
See Also:

Using Solvers
Working with the Object Tree
Using Multiple Stopping Criteria
Common Pop-Up Menu Items

Setting the Maximum Physical Time

The **Maximum Physical Time** stopping criterion is based on the simulation time that has elapsed in a transient analysis.

The node of this criterion has its own properties, and appears when either the implicit unsteady model or the explicit unsteady model is chosen.



When used with the implicit unsteady model, this stopping criterion is linked to the *Time-Step* property of the **Implicit Unsteady** node in the **Solvers** node. If the *Time-Step* is set to 1 second, and the *Maximum Physical Time* is set to 10 seconds, then the simulation runs for 10 time-steps.

To set the maximum time, click the **Maximum Physical Time** node and modify the *Max Physical Time* property in the *Properties* window.

Properties

Enabled	Determines if this criterion is used for stopping the solver.	
	Activated	Use this criterion.
	Deactivated	Do not use this criterion.
Max. Physical Time	The amount of physical time before this criterion is satisfied. The default is 1 second.	
Criterion Satisfied	Indicates whether the stopping criterion has been satisfied (read-only).	
Logical Rule	Defines how this criterion interacts with other stopping criteria that have been enabled.	
	AND	Requires that this criterion, along with one or more other criteria with the *Logical Rule* set to **AND**, all be satisfied before stopping the solver.

	OR	Indicates that satisfying only this criterion is sufficient to stop the solver.

See Also:

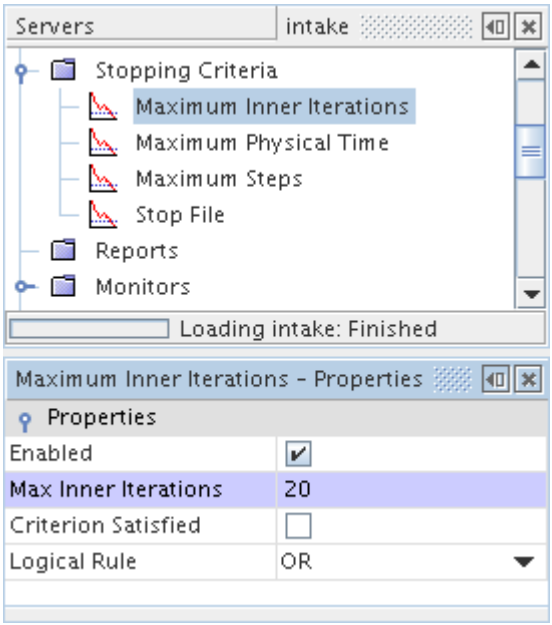
Using Multiple Stopping Criteria

Common Pop-Up Menu Items

Setting the Maximum Number of Inner Iterations

The **Maximum Inner Iterations** stopping criterion is based on the number of inner iterations that the solver executes for transient analyses.

The node of this criterion has its own properties, and appears when the implicit unsteady model is chosen.



If the implicit unsteady solver is used, this stopping criterion can control the number of inner iterations to be run at each physical time-step. The **Maximum Inner Iterations** stopping criterion differs from other stopping criteria. This stopping criterion does not control when to stop the solver, only when to stop the inner iterations of the implicit solver and march the solution to the next time-step.

You can change the number of inner iterations by clicking the **Maximum Inner Iterations** criterion node and then modifying the *Max Inner Iterations* property.

Properties

Enabled	Determines if this criterion is used for stopping the solver.	
	Activated	Use this criterion.

	Deactivated	Do not use this criterion.
Max. Inner Iterations	The number of inner iterations to take before this criterion is satisfied. The default is 5 iterations.	
Criterion Satisfied	A read-only property that indicates when the stopping criterion has been satisfied.	
Logical Rule	Defines how this criterion interacts with other stopping criteria that have been enabled.	
	AND	Requires that this criterion, along with one or more other criteria with the *Logical Rule* set to **AND**, all be satisfied before stopping the solver.
	OR	Indicates that satisfying only this criterion is sufficient to stop the solver.

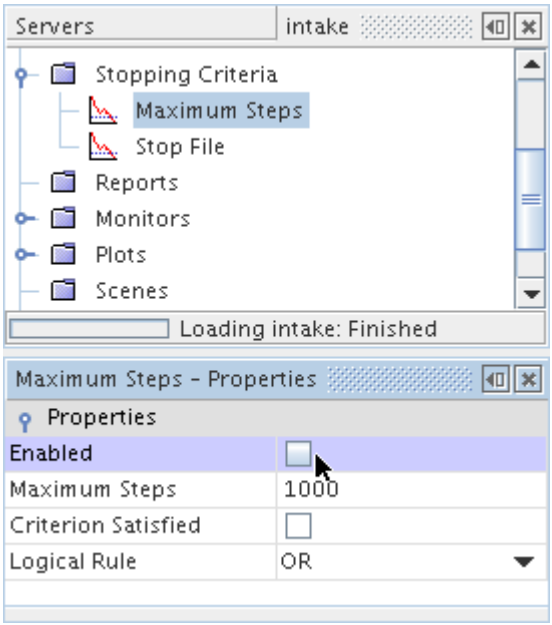
See Also:

Using Solvers
Using Multiple Stopping Criteria
Common Pop-Up Menu Items

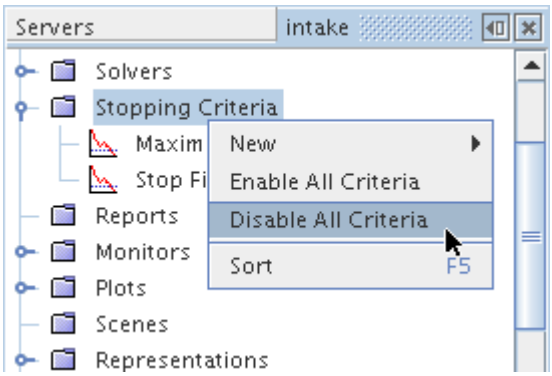
Disabling/Enabling Stopping Criteria

This section describes how to enable and disable stopping criteria.

To deactivate an automatically generated stopping criterion, click the criterion node and deactivate the *Enabled* property in the *Properties* window.



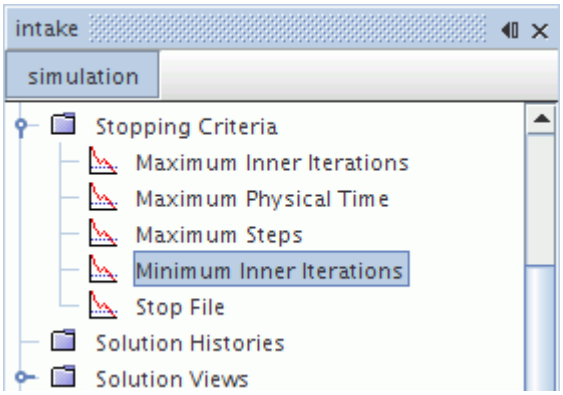
All criteria can be activated or deactivated at the same time. To activate or deactivate all criteria, right-click the **Stopping Criteria** node and select **Enable All Criteria** or **Disable All Criteria** respectively.



Using Manually Generated Stopping Criteria

Residuals do not always reflect changes after the first inner iteration in a transient analysis. The **Minimum Inner Iterations** stopping criterion prevents a time-step from stopping until the solver finishes the specified number of inner iterations.

The node of this criterion has its own properties.

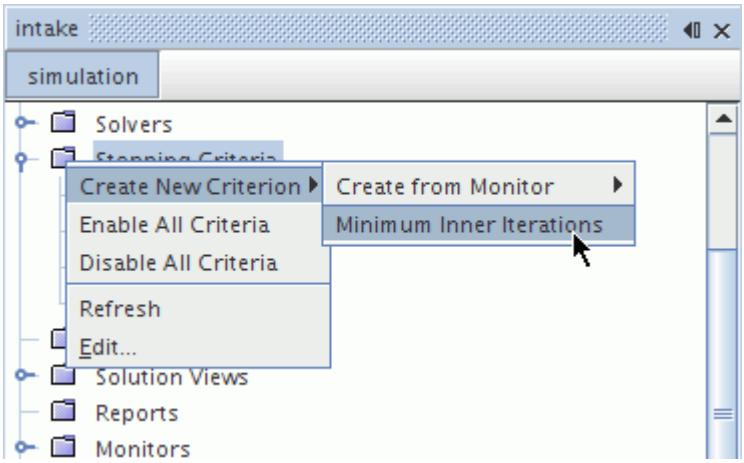


Minimum Inner Iterations Properties

Minimum Inner Iterations	The number of inner iterations to take before this criterion is satisfied. The default is 1 iteration.	
Enabled	Determines whether this criterion is used for stopping the solver.	
	Activated	Use this criterion.
	Deactivated	Do not use this criterion.
Criterion Satisfied	A read-only property that indicates whether the stopping criterion has been satisfied.	
Logical Rule	Defines how this criterion interacts with other stopping criteria that have been enabled.	
	AND	Requires that this criterion, along with one or more other criteria with the *Logical Rule* set to **AND**, all be satisfied before stopping the solver.
	OR	Indicates that satisfying only this criterion is sufficient to stop the solver.

Adding the Minimum Inner Iterations Stopping Criterion

To make this stopping criterion available, right-click the **Stopping Criteria** node and select **Create New Criterion** > **Minimum Inner Iterations**. (This option is available for transient simulations only.)



See Also:

Using Solvers

Using Multiple Stopping Criteria

Using Monitors as Stopping Criteria

It is possible to create stopping criteria that are based on existing monitors. This function lets you use more meaningful criteria to judge convergence.

The node of a monitor-based stopping criterion has properties just like the automatically generated kind.

These monitor-based stopping criteria could include a reduction in a residual, or could be based upon some physical quantity that you are interested in obtaining from the solution. For example, if you are simulating the flow over an airfoil you can stop iterating when the drag and lift coefficients of the airfoil have reached steady values.

You can create stopping criteria that are based on any existing monitor. These criteria check the value of the associated monitor and return a satisfied condition when that value reaches some user-specified minimum, maximum, asymptotic limit, or standard deviation. You can also copy properties of these criteria.

Using Monitor-Based Stopping Criteria in Transient Simulations

If you are using the implicit unsteady model, these criteria apply automatically to the inner iterations. To apply the criteria to the outer iterations, so that they stop the entire simulation, tick the *Stop Outer Iterations* property of the corresponding criterion node.

In a transient analysis these criteria, when satisfied, allow only one inner iteration by default. To require more than one inner iteration:

- Manually create the Minimum Inner Iterations criterion.
- Set the properties of the Minimum Inner Iterations criterion:
  - Set the *Minimum Inner Iterations* property to `3` or `4`.
  - Make sure that the *Logical Rule* property is set to `And`.
- Set the *Logical Rule* property of the other monitor-based criteria to `And`.

Properties

Monitor	The name of the monitor that is used to create this stopping criterion (read-only).	
Stop Inner Iterations	Stops the inner iterations of the transient simulation when the criterion (as specified in *Criterion Option*) is satisfied.	
	Activated	Use this criterion.
	Deactivated	Do not use this criterion.
Criterion Option	The method to use for evaluating this criterion.	
	Minimum	Specifies that this criterion is satisfied when the monitor reaches a user-specified minimum value. A Minimum Limit node is added as a sub-node.
	Maximum	Specifies that this criterion is satisfied when the monitor reaches a user-specified maximum value. A Maximum Limit node is added as a sub-node.
	Asymptotic	Specifies that this criterion is satisfied when the monitor has stabilized to a particular range during a number of iterations. The range and iteration window are specified in the Asymptotic Limit sub-node.
	Standard Deviation	Specifies that this criterion is satisfied when the monitor reaches a user-specified standard deviation. A Standard Deviation node is added as a sub-node.
	Relative Change	Specifies that this criterion is satisfied when the ratio of current value to initial value of the monitor reaches a user-specified relative change value. The initial value is determined at the start of each time-step for transient cases.
Stop Outer Iterations	When **On**, stops the transient simulation when the criterion (as specified in *Criterion Option*) is satisfied.	
Enabled	When **On**, determines if this criterion is used to stop the solver.	
Criterion Satisfied	Indicates if the stopping criterion has been satisfied (read-only).	

Logical Rule	Defines how this criterion interacts with other stopping criteria that have been enabled.	
	AND	Requires that this criterion, along with one or more other criteria with the *Logical Rule* set to **AND**, all be satisfied before stopping the solver.
	OR	Indicates that satisfying only this criterion is sufficient to stop the solver.

Contents:

Creating a Monitor-Based Criterion

See Also:

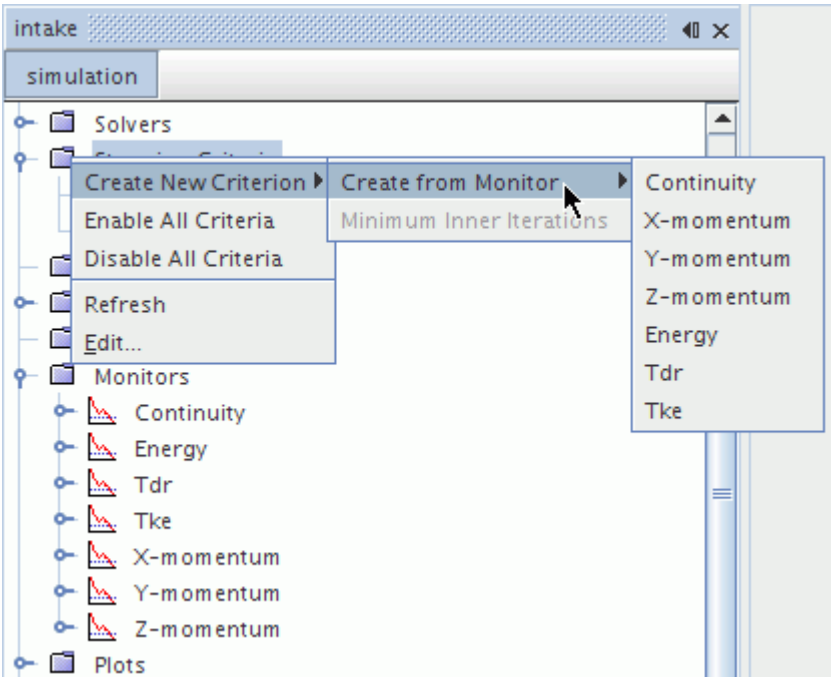
What Is a Monitor?
Using Solvers
Judging Convergence
Using Manually Generated Stopping Criteria
Using Multiple Stopping Criteria

Creating a Monitor-Based Criterion

This section describes how to create a stopping criterion from an existing monitor.

To create a stopping criterion from an existing monitor:

1. Right-click on the **Stopping Criteria** node and select **Create New Criterion** > **Create from Monitor**.
2. Select from the list of existing monitors.

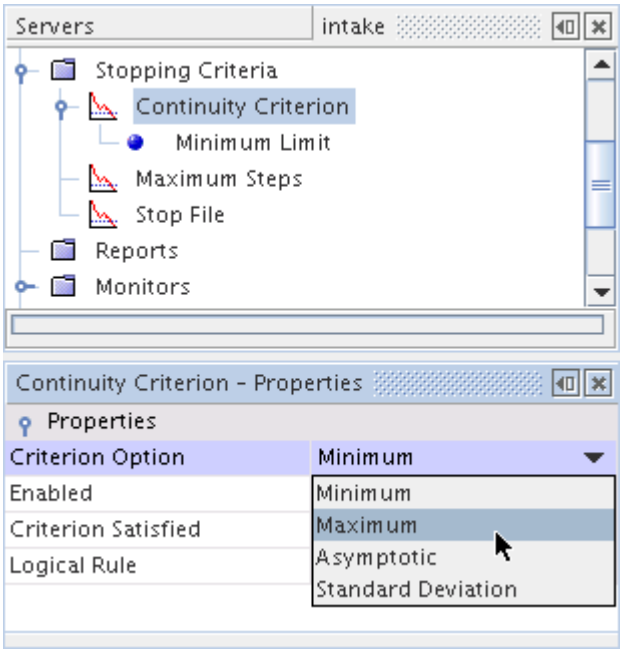


A new stopping criterion node is added in the **Stopping Criteria** manager node. The new node is named after the selected monitor with `Criterion` appended to the name, and possibly a digit to make the new criterion unique among others with similar names. However, renaming the stopping criterion is possible. The new criterion is activated by default, with the *Criterion Option* set to `Minimum`, and the *Logical Rule* set to `OR`.

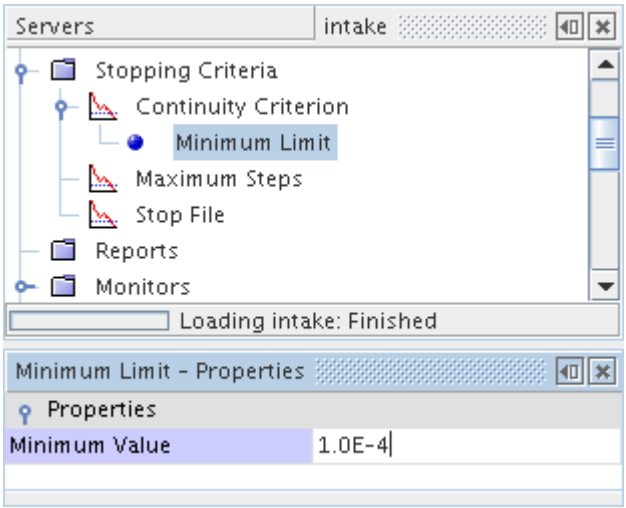
Min/Max Monitor

By default all criteria that are created from a monitor use a minimum limit to determine when to stop the simulation. The type of limit can be changed to a maximum, an asymptote or a standard deviation.

To change the type of limit, select the monitor stopping criterion node, then from the pull-down menu under the *Criterion Option* property in the *Properties* window, select the type. You can use a monitor to create more than one criterion. For example, a monitor could be used to stop the iteration when its value reaches a maximum or an asymptote.



If you select a *Minimum* or *Maximum* limit, you can set the limiting value by selecting the **Minimum Limit** or **Maximum Limit** node in the **Stopping Criteria** manager. In the properties window, you can then modify the *Minimum Value* or *Maximum Value* property.



Minimum Limit Properties

Minimum Value	The minimum monitor value that satisfies this criterion. The solver stops as soon as the monitor value is less than this specified value.

Maximum Limit Properties

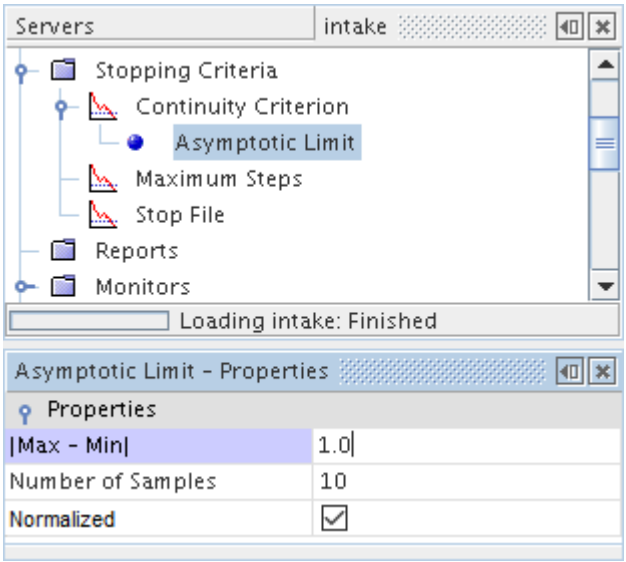
Maximum Value	The maximum monitor value that satisfies this criterion. The solver stops as soon as the monitor value is greater than this specified value.

Asymptotic Monitor

It is also possible to set the criterion to stop once it has reached an asymptotic limit. It allows for definition of the range over which the value can fluctuate in a given number of samples.

This feature is useful when the exact limiting value is unknown but you do know the maximum change in the monitored value that you consider sufficient for convergence.

To select the asymptotic limit option, select the monitor stopping criterion node and select the **Asymptotic** option in the drop-down list under the *Criterion Option* property in the *Properties* window. The limit is set by selecting a maximum permissible absolute change in the monitored value over a given number of iterations. Expand the monitor stopping criterion node, select the **Asymptotic Limit** child node and edit the absolute change in the monitored value and the number of samples in the *Properties* window.



Asymptotic Monitor Properties

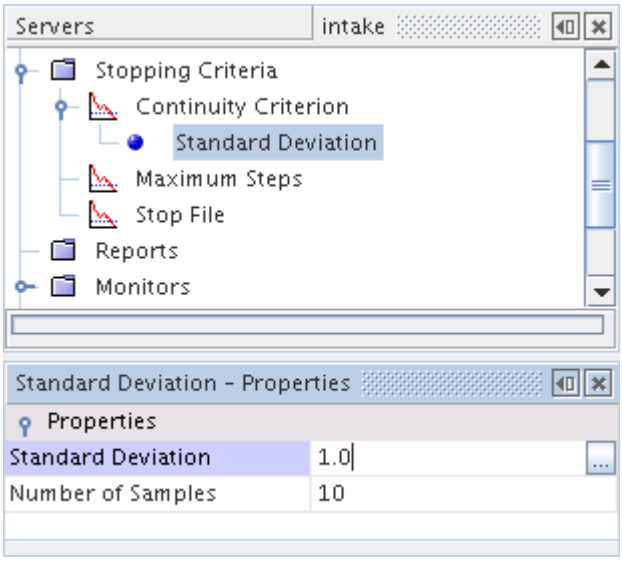
|Max - Min|	The difference between the smallest and largest value that is collected during the previous *Number of Samples*. If this difference is not exceeded, then this criterion is satisfied.
Number of Samples	The number of samples that the monitor values check to see if the absolute change is exceeded.
Normalized	When **On**, this normalizes the **|Max - Min|** values to the average value over **Number of Samples**. The default value for **|Max - Min|** is then 0.1. The default is **On**.

Standard Deviation Monitor

You can use the standard deviation for a given number of samples.

To select this option, select the monitor stopping criterion node and select the **Standard Deviation** option in the drop-down list under the *Criterion Option* property in the *Properties* window. Expand the monitor stopping

criterion node, select the **Standard Deviation** child node and edit the standard deviation and the number of samples in the *Properties* window.



Standard Deviation Monitor Properties

Standard Deviation	The standard deviation of the samples, entered as text.
Number of Samples	The number of samples that the monitor values check.

Copying Properties of Monitor-Based Stopping Criteria

You can copy a property from one limiting value node to another.

If you have two (or more) monitor-based stopping criteria and you want them to have the same limiting value, you can copy that property from one limiting value node to another. You can only copy from the same type of limiting value node (minimum, maximum, or asymptotic).

Renaming Monitor-Based Stopping Criteria

Similar to other objects, a monitor-based stopping criterion can be renamed within STAR-CCM+.

See Also:

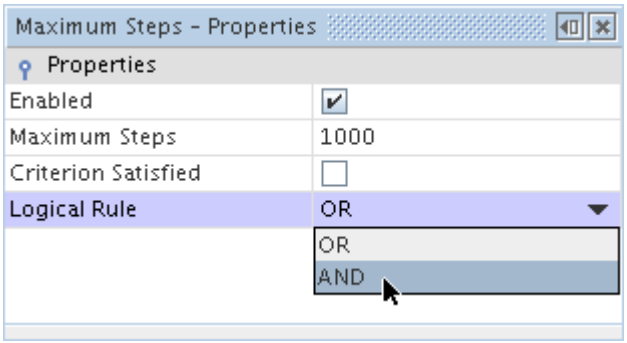
What Is a Monitor?

Using Multiple Stopping Criteria

If more than one stopping criterion is activated, it is necessary to construct a logical rule to determine when the criteria stops the solver.

For example, you can stop the solver when all criteria generated from monitors have been satisfied. However, you can also stop the solver when the number of iterations exceeds a limit.

Logical rules (AND, OR) are assigned to each individual criterion and are used to determine how they interact as a group. If a criterion is assigned an OR logical rule, the solver stops when it is satisfied. The solver also stops when all the criteria assigned the AND logical rule are satisfied. By default all criteria are assigned the OR logical rule. To change the logical rule that is associated with a criterion, select the criterion node then in the *Properties* window, set the option that you require for the *Logical Rule* property.



For example, if you have four criteria named A, B, C, and D, the Logical Rule is set to OR for A and B, then C and D have the Logical Rule set to AND. In that case, the boolean expression that is tested is:

```
A OR B OR (TRUE AND C AND D)
```

The `TRUE` at the beginning of the bracketed `AND`s caters for the (unusual) situation where there is only one criterion with an `AND`.

Obtaining a Solution Interactively

This section explains how to use the interactive features of STAR-CCM+ to obtain a solution.

Other STAR-CCM+ features such as scenes and plots, which are normally associated with post-processing, can help you examine results in real time while the simulation is running. These functions are described in the chapter on analyzing.

Contents:

Initializing the Solution
Clearing the Solution
Running the Simulation
Stepping the Solution
Watching Residuals
Stopping the Solution
Steering the Solution
Saving Intermediate Solutions

See Also:

Troubleshooting the Solution

Initializing the Solution

The solution must be initialized to apply the initial conditions.

Once initial conditions have been set for each model in the simulation, the solution must be initialized, either automatically or manually, to apply the initial conditions. In general, you do not need to initialize manually as the automatic initialization is always run at the appropriate times.

In general, solution initialization requires no explicit action on your part. Before running or stepping, the solution is initialized automatically as required.

During solution initialization, each model that is activated in the simulation is instructed to allocate storage for its solution data. The data is then set according to the specified initial values for the model.

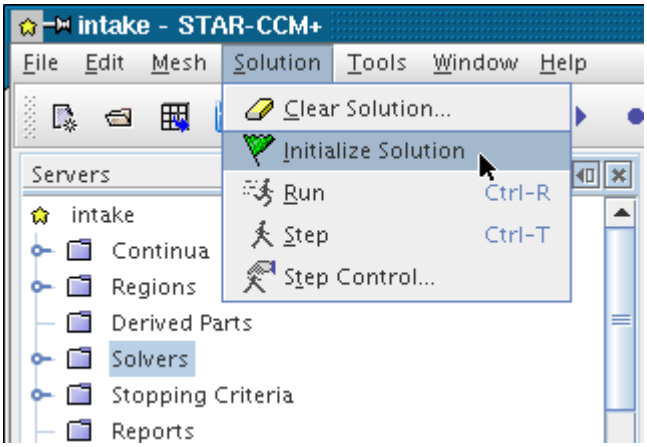
If storage exists and is initialized, it is not changed. To reinitialize an existing solution completely, first clear the solution.

If the mesh is changed in any way, then mesh variables such as volume are cleared and need reinitialized. This phenomenon occurs in rotor stator problems, for example, each time the rotor is stepped and a new interface intersection is calculated.

Initializing Manually

If you want to visualize the initialized solution, initialize manually.

To initialize, click the (**Initialize Solution**) button in the toolbar or use the **Solution** > **Initialize Solution** menu item.



If the solution has already been initialized (manually or automatically), then this action has no effect. Executing this action after one or more iterations does not reinitialize the solution data. Any values that have been calculated for the solution variables to that point are untouched.

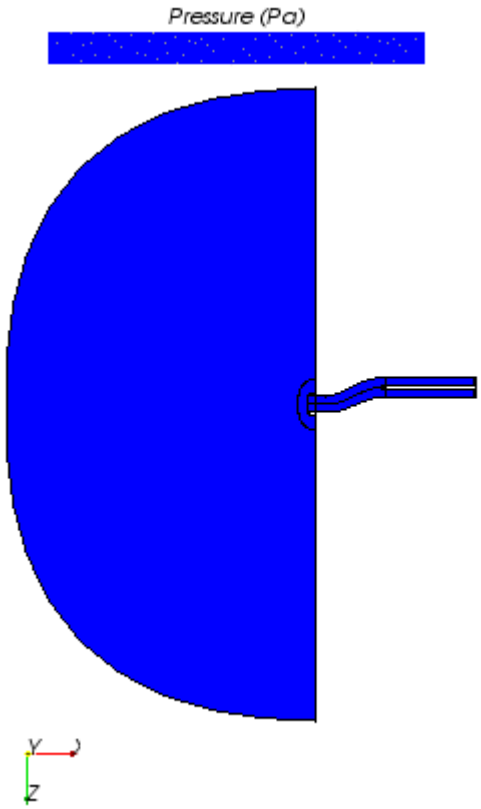
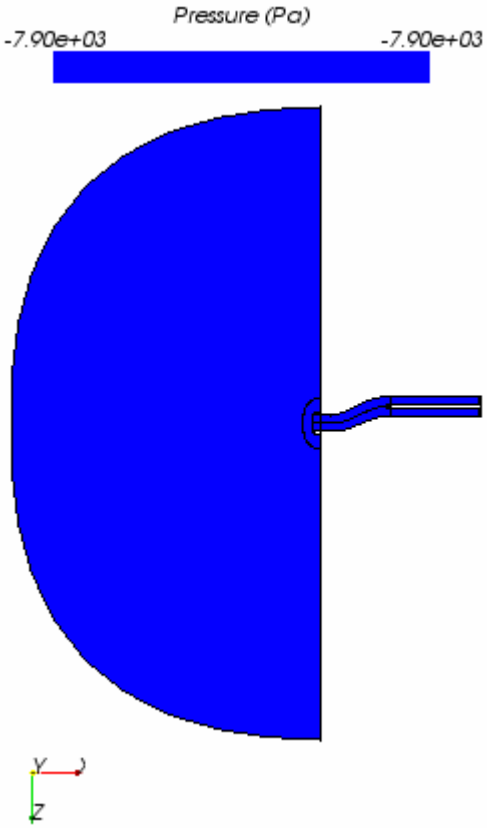
If you then change or add models to the simulation, an initialization sets the solution variables for the changed models.

You can use this feature with a vector displayer to display the direction of quantities on boundaries, even before you run the simulation.

Viewing the Initialized Solution

It can sometimes be informative to view the initialized solution to understand whether the initial data make sense.

Before scalars or vectors can be displayed, or reports generated, an initialized solution must exist. If it does not, no data is displayed. The omission of a scale on the scalar or vector bar of the scalar or vector display indicate the lack of data (rather than just a constant initial value).

	
Before Initialization	After Initialization

See Also:

Setting Convergence Accelerators

Clearing the Solution

The action of clearing a solution causes all storage that is associated with the solution to be cleared from the memory of the computer.

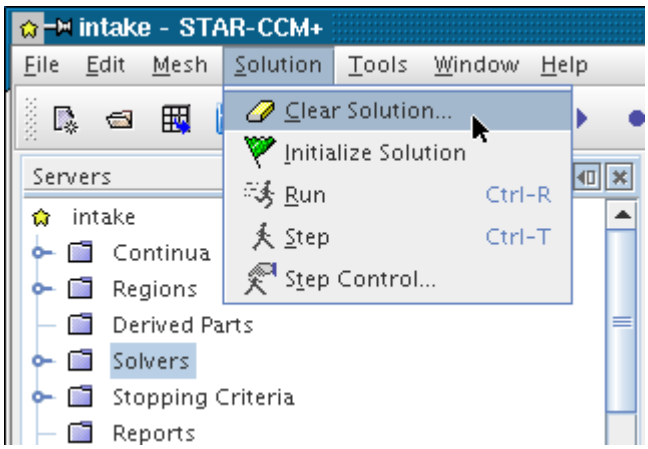
This storage includes the contents of the Residuals plot of the *Graphics* window, as well as monitor information. If the simulation is saved with the solution cleared, no solution data is stored in the simulation file.

If you want to reinitialize a solution and start again, clear the solution. An alternative approach, however, is clearing the solution for a particular region or a particular boundary.

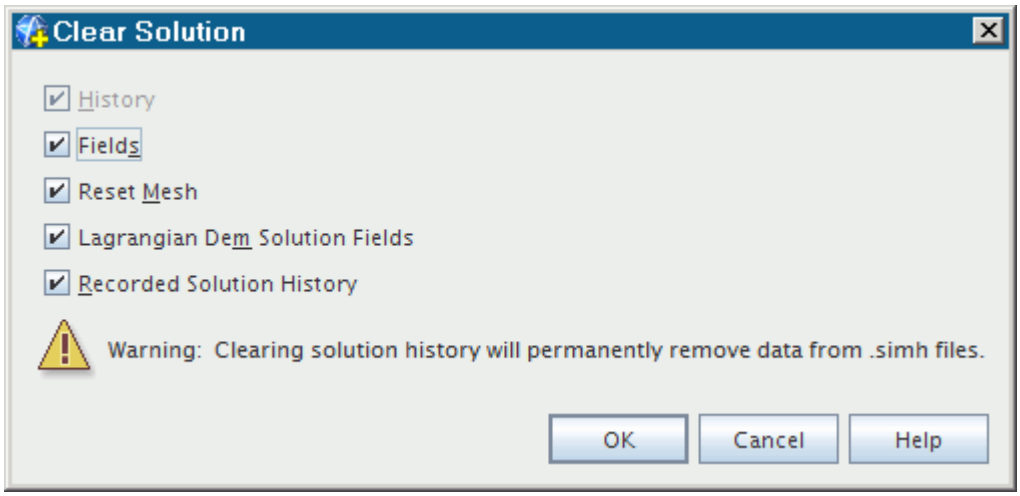
Storage that is associated with any model is not cleared when a model is deactivated. Clearing the solution is the only way of ensuring that unnecessary storage from deactivated models is not contained within the simulation. However, in some situations you can choose not to clear the solution after changing models. For example, if it is beneficial to use the solution data from one model as the initial conditions for another model.

To clear the solution:

1. Select **Solution** > **Clear Solution...**.



The *Clear Solution* dialog appears:



The options available on this dialog are described as follows:

- **History** – This option clears the current iteration number and time (for an unsteady case), as well as all monitors and residual plots. This option is always activated.
- **Fields** – Activating this option clears all data fields from the solution. To maintain the data fields so that they remain in place as initial conditions, deactivate this option. If you selected a motion model, the initial position of the mesh is also cleared when this option is activated, so that you can modify the initial mesh position before commencing the next simulation.
- **Reset Mesh** – This option is only available when you have selected a motion model or a turbulence model. If a motion model is active, activate this option to reset all mesh vertices to their original position at time zero. This action will also clear the storage for initial position, so that you can manipulate regions before commencing the next simulation. If a turbulence model is active, activate this option to clear the wall distance solution.
- **Lagrangian DEM Solution Fields** – This option is only available when you have selected the Discrete Element Model (DEM). To clear all DEM particles from the solution, activate this option.

- **Recorded Solution History** - This option is only available when you have created a simulation history file. To clear the solution data that is stored in the simulation history file, activate this option.

2. Deactivate any of the options if necessary, and click **OK**.

Clearing Solution Data from a Region

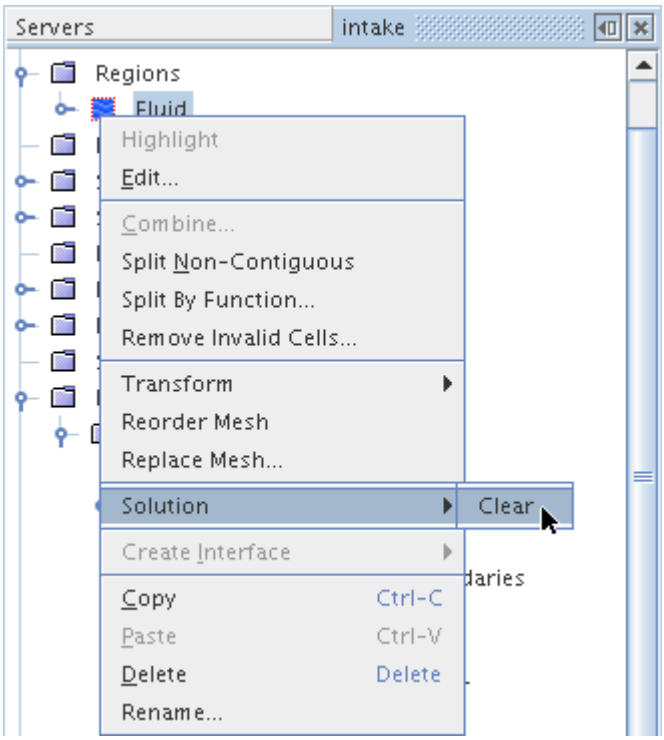
By clearing the solution for an individual region rather than for the simulation as a whole, you retain much of the material that is produced during the simulation run.

This information includes plots, monitor information and, for a transient analysis, the physical time (which is reset to zero upon the clearing of the entire solution). You can also step the solution extra iterations in addition to the iterations of the original run.

This option is also helpful if the solution results for one region are less satisfactory than for others. If you do not need to keep solution results for all regions, it can also help to reduce simulation file size.

To clear the solution for an individual region:

- Right-click the node for that region and select **Solution** > **Clear**.



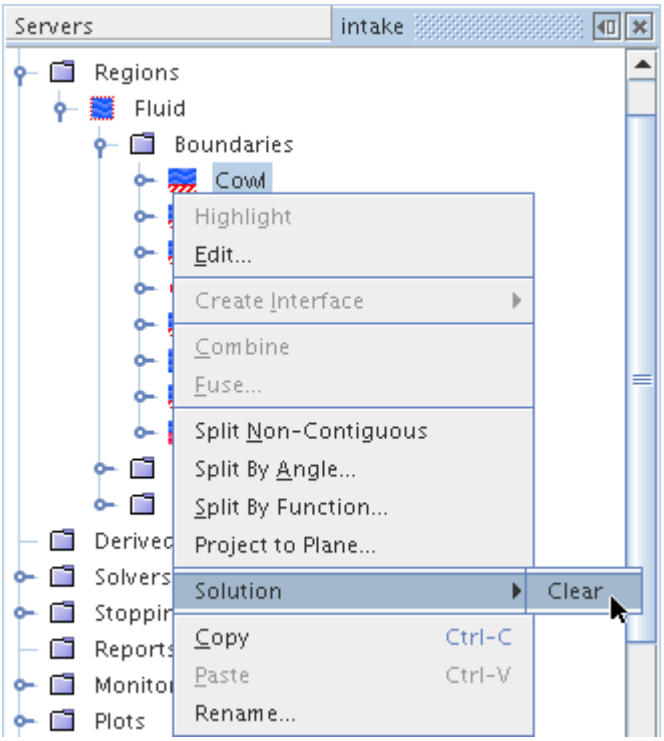
This step leaves in place the Residuals plot of the *Graphics* window, as well as monitor information.

Clearing Solution Data from a Boundary

The procedure for clearing the solution for a single boundary is similar to the procedure for a particular region.

To clear the solution for a single boundary:

- Right-click the node for that boundary and select **Solution** > **Clear**.



Resetting for Restart

Earlier versions of STAR-CCM+ had a **Reset for Restart** option on the **Solution** menu. This operation is now accomplished using the checkboxes that are provided on the *Clear Solution* dialog. To reset only the iteration number, current time, and residual and monitor plots, clear the **Fields** option on the dialog before clicking OK. Field data will be retained to act as the initial condition for the next simulation.

See Also:

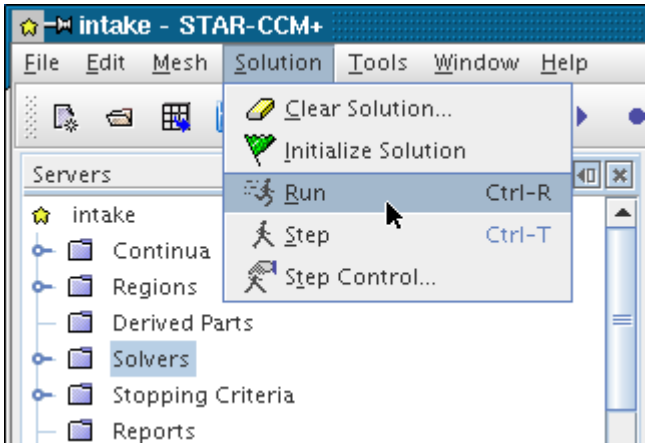
Troubleshooting the Solution

Running the Simulation

Once a simulation has been set up and there is reasonable confidence that the setup is correct, solution iteration can commence.

To start iterating:

1. Click the (**Run**) button in the toolbar, press `<Ctrl><R>`, or use the **Solution** > **Run** menu item.



If the solution is not already initialized before iterations commence, it is automatically initialized. The iterations are displayed in the residual plot in real time.

In addition to the residual plot, STAR-CCM+ provides other features that can help you examine results while the simulation is running. These features are described in the chapter on analyzing.

The simulation runs until the specified stopping criteria are met.

See Also:

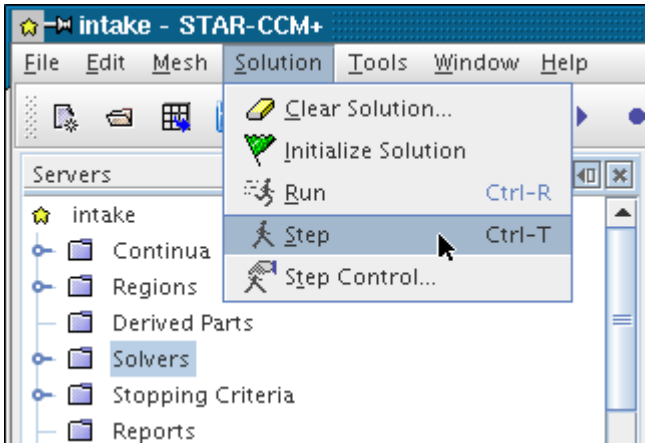
Troubleshooting the Solution

Stepping the Solution

You can step the solution a fixed number of iterations, or time-steps if running an unsteady simulation.

To step the solution:

1. Click the (**Step**) button in the toolbar, press `<Ctrl><T>` on the keyboard, or use the **Solution** > **Step** menu item.

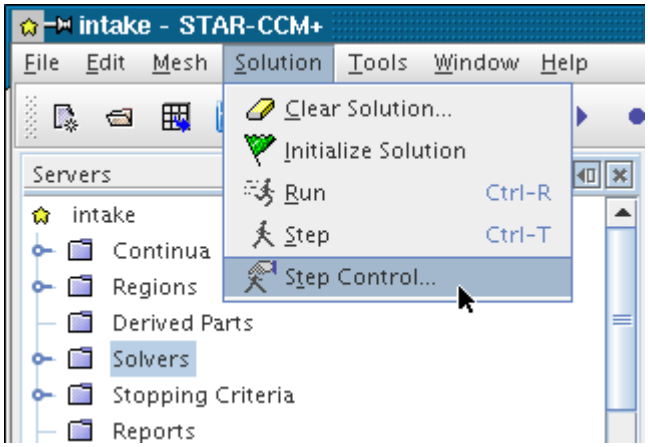


If the solution is not already initialized before iterations commence, it is automatically initialized. The stopping criteria are ignored.

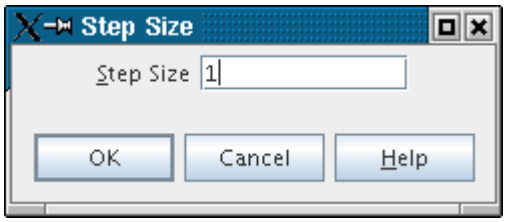
You can also adjust the size of a step with the step control feature.

The number of iterations or time-steps taken for each step has a default value of 1.

2. Select the **Solution** > **Step Control** menu item.



The *Step Size* dialog appears.



3. Enter the step size and click **OK**.

   Each subsequent step uses this step size.

The iteration or iterations are displayed in the residual plot.

See Also:

Watching Residuals
Running the Simulation
Setting a Stop File

Watching Residuals

This section explains the concept of calculating residuals, as well as working with residual monitors, and residual plots.

For using residuals to monitor a solution, see the section on judging convergence.

What Is a Residual?

The residual in each cell represents the degree to which the discretized equation is satisfied. For example, consider the transport equation for a passive scalar and its component terms (the full description can be found at Passive Scalar Formulation). When the solver is run, a discretized version of this equation is solved for each cell in the mesh. The residual of this equation is:

$$r = \left| \frac{\partial}{\partial t} \int_V \alpha_i \rho \chi \phi_j \, dV + \oint \alpha_i \rho \chi \phi_j \left(\mathbf{v} - \mathbf{v}_g \right) \cdot d\mathbf{a} - \left(\oint \left[\alpha_i \chi \left(\frac{\mu}{\sigma} + \frac{\mu_t}{\sigma_t} \right) \nabla \phi_j \right] \cdot d\mathbf{a} + \int_V S_{\phi_j} \, dV \right) \right| \tag{92}$$

where:

r is the absolute error in the solution of a particular variable. In a perfectly converged solution, the residual for each variable for each cell would be equal to machine roundoff.

In STAR-CCM+, the RMS (Root Mean Squared) value of a residual for all cells is determined from:

$$R_{rms} = \sqrt{\frac{1}{n} \Sigma_n r^2} \tag{93}$$

where:

n is the number of cells.

Residual monitors keep a record of this global quantity for each of the transport equations solved in the continua within the simulation. As the residuals can vary widely in absolute value, by default a normalized residual, R_{pres}, is presented. This function allows you to compare multiple residuals as all residuals have a value from 0 through 1. R_{pres} is determined from:

$$R_{pres} = \frac{R_{rms}}{R_{norm}} \tag{94}$$

The normalization value can be set manually to any non-zero value. However, if left at its default, R_{norm}, the normalization is automatic:

$$R_{norm} = \max \left\{ \left| R_1 \right|, \left| R_2 \right|, \ldots, \left| R_m \right| \right\} \tag{95}$$

where:

$R_1, R_2, \ldots, R_m$ are the residuals at each iteration.

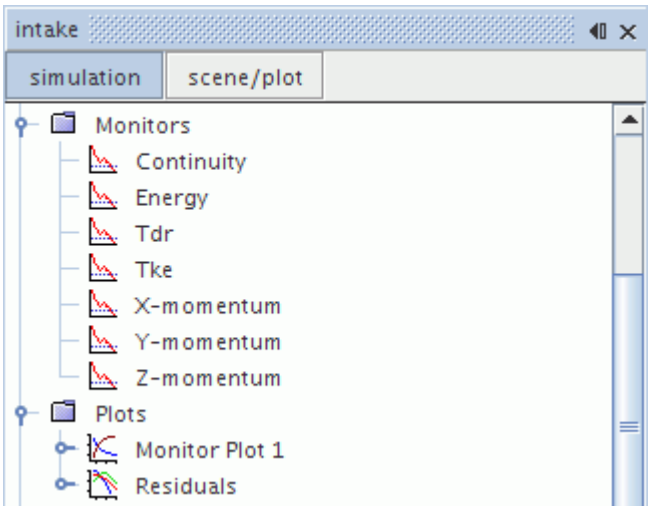
m is the Normalization Number property that selects the number of iterations to compare before holding a single normalization value. By default, m is set to 5, as this setting removes the influence of the initialization values (reflected in R_1, R_2) but provides a sensible value for observing convergence that does not change after every iteration.

In general, use normalized residuals to monitor the behavior of the solvers at each iteration to make sure that the solution is converging toward the machine precision. If the simulation is steady state, the convergence is

displayed over the total number of iterations. If the solution is time-dependent, make sure that convergence occurs within each time-step. For more details, see the section on judging convergence.

Adding Residual Monitors

Residual monitors are created automatically after model selection, and cannot be deleted or renamed.



When a solver runs, a residual plot is automatically created that displays the residual monitors on semi-log axes.

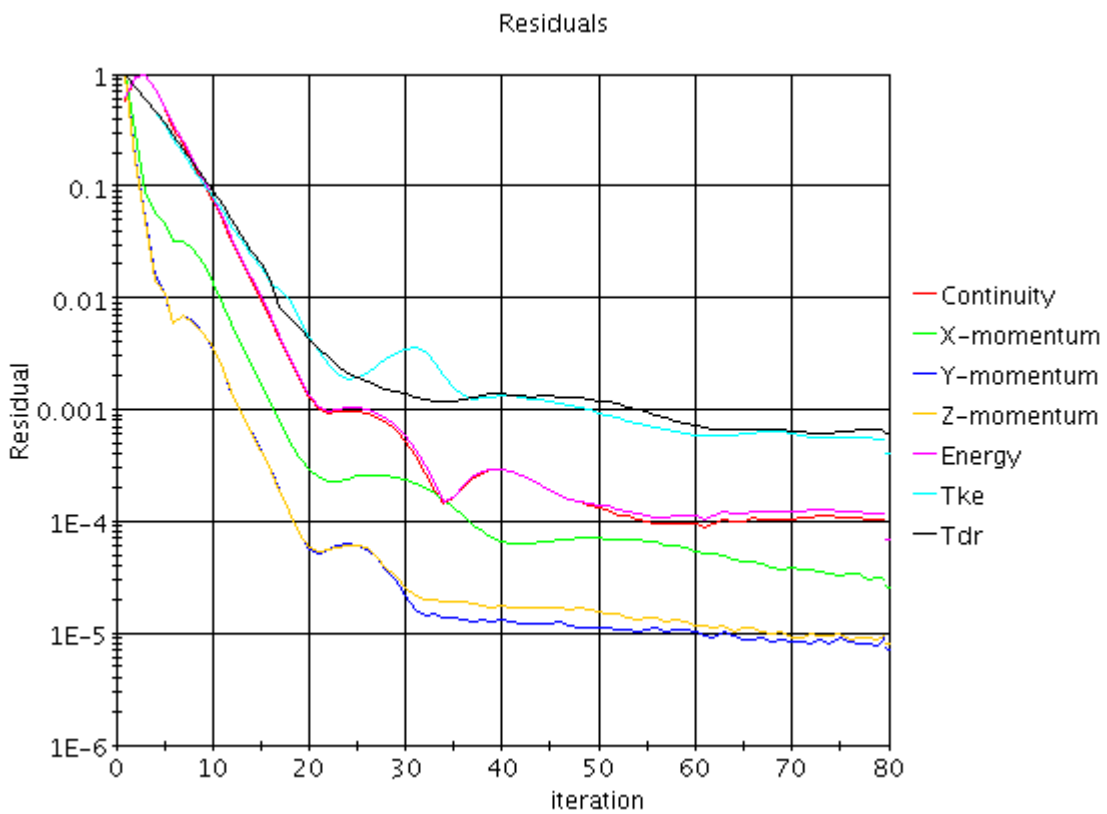
Residual Monitors Properties

Active	Read-only property to indicate if the models have been changed in this simulation after a simulation was run.	
	Activated	The monitor is active in the current simulation.
	Deactivated	Any monitors specific to the previous models become deactivated unless you delete them.
Trigger	Defines what action triggers the collection of a sample. The default setting depends on whether you are using the Steady model (iterations) or Implicit Unsteady model (time-steps).	
	Iteration	A sample is taken every `N` iterations.
	Time-Step	A sample is taken every `N` time-steps.
	Inactive	No sample is taken.

Maximum Plot Samples	Limits the number of sample points that used in the monitor plot.	
Frequency	Determines how often the sample is taken using the units that are specified in *Trigger*.	
Normalization Option	Defines how the monitor data is normalized.	
	Off	Do not normalize.
	Auto	Lets STAR-CCM+ do the normalization as derived in the Automatic Normalization node that is added as a sub-node when you choose this option.
	Manual	Lets you define a specific normalization factor in the Manual Normalization node that is added as a sub-node when you select this option.

What Is a Residual Plot?

The residual plot is a monitor plot that is automatically created from the active residual monitors on semi-log axes when iterating starts. A sample residual plot is shown below.



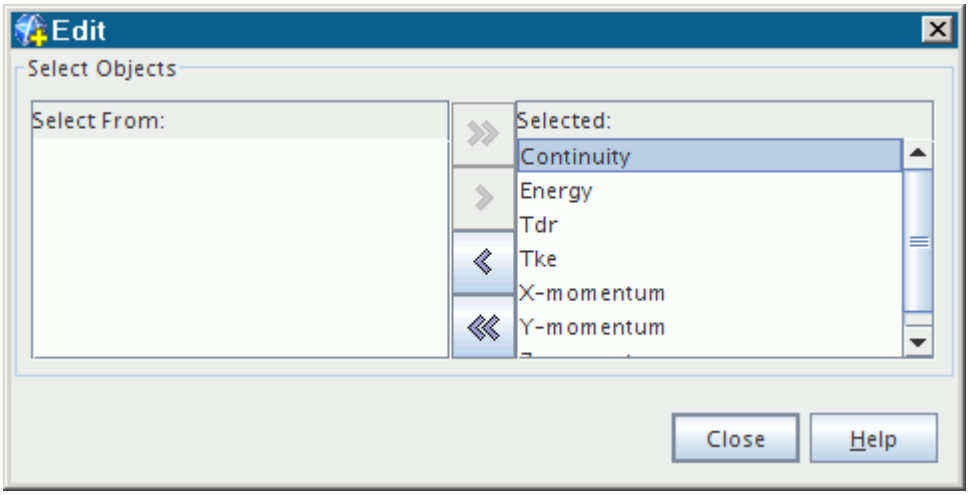
By default, all active residuals are displayed in the residual plot. The residual plot can be renamed but it cannot be deleted, though individual data series can be deleted from the plot.

The residual plot uses the same properties as the monitor plot.

Adding and Removing Residuals from the Residual Plot

All monitored residuals are displayed in the residual plot by default. Specific residual monitors can be removed from the plot by right-clicking on the node or nodes in the **Monitors** node under the **Residual Plot** node and selecting **Remove**. Alternatively, you can right-click on the **Monitors** node and select **Edit...** to access the *Customizer* dialog to add and remove monitors from the plot.

The residual plot monitors customizer allows you to add and remove specific residual monitors from the plot.



Adding or removing residual monitors does not affect the currently selected models or the stored data; only the plotting of the residuals is affected. However, these changes are not saved to the simulation file. The residual plot is reset to include the residuals from all of the selected models when you change the currently selected models.

See Also:

Setting Maximum Plot Samples
Creating a Monitor Plot
Running the Simulation
Monitor Plot Properties

Stopping the Solution

This section describes how to stop the solution.

Once is solution is iterating, the solution can be interrupted by clicking the (**Stop**) button in the toolbar. The solution can be resumed again by running or stepping.

If the solution is stopped during an inner iteration, the time-step can be incomplete for transient simulations. In this case, a message similar to the one below appears in the *Output* window.

```
Output - intake
      5   1.563991e-02 3.964559e-01 2.019408e-01 1.942943e-01 8.230692e+01
      6   1.449795e-02 2.656780e-01 1.917255e-01 1.872454e-01 6.161633e+01
      7   1.336314e-02 1.779128e-01 1.732034e-01 1.671565e-01 5.817261e+01
      8   1.250677e-02 1.658569e-01 1.477493e-01 1.382004e-01 5.610940e+01
Breaking out of time step iteration.
Time solution is in an inconsistent state.
Step once more to complete the time step.
      9   1.173682e-02 1.643345e-01 1.223129e-01 1.134307e-01 5.296078e+01
Stopping criterion Maximum Physical Time satisfied.
```

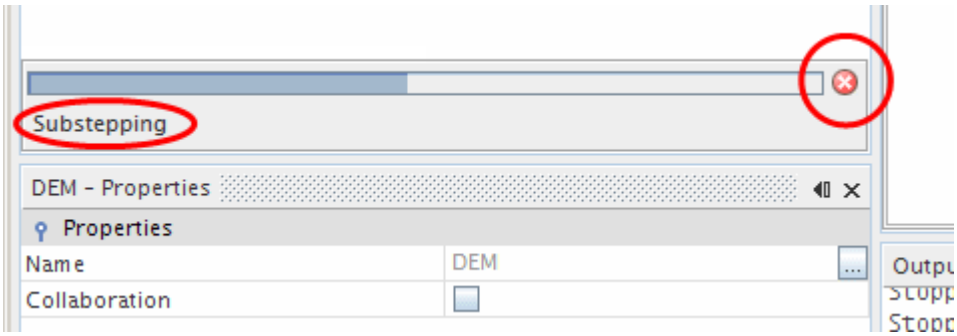
As indicated, stepping the solution once completes the time-step.

Although primarily intended for batch operation, the stop file stopping criterion also aborts the job while the solution is iterating interactively.

Stopping Lagrangian Iterations

In some Lagrangian simulations, particularly those using DEM models, clicking the (**Stop**) button does not immediately stop the simulation. This phenomenon is due to the large number of sub steps that are left in the iteration.

You can force an immediate stop by using the (**Substepping Stop**) button. This button appears at the bottom of the simulation window, next to a substepping progress bar, during runs of Lagrangian simulations. It becomes active after 100 substeps. To stop a Lagrangian simulation without completing that iteration or the substepping process, click the **Substepping Stop** button.



After you click this button, particle time and iteration time fall out of synchronization. For this reason, do not use the **Substepping Stop** button as a routine way to control simulations.

Steering the Solution

STAR-CCM+ allows you to monitor the progress of a solution in several ways and to change simulation parameters while running.

STAR-CCM+ contains state-of-the-art numerical methods that are both accurate and robust. However, the equations of fluid flow are highly non-linear and can be difficult to solve for situations that involve complex physics, complicated geometries and/or poor-quality meshes. Human intervention is therefore often required to obtain a solution in the most efficient manner. For this reason, STAR-CCM+ allows you to monitor the progress of a solution in several ways and to change simulation parameters while running.

If you decide that the simulation is complete after the stopping criteria have been met, save the simulation and commence post-processing.

However, you can also change the stopping criteria and continue running. For instance, if a simple stopping criterion such as number of iterations was used, the iterations can stop before acceptable convergence has been reached. In this case, the stopping criteria can be changed and the solution continued by running again.

Contents:

Monitoring Solution Progress
Changing the Simulation While Running
Judging Convergence

Monitoring Solution Progress

In addition to the residual plot, the progress of the solution while iterating can be monitored either by using monitors, or by visualizing the solution with scalar displayers and/or vector displayers.

Both monitor update and displayer update policies can be set to regular intervals of iterations or time-steps if transient.

Other STAR-CCM+ features that can help you monitor solution progress are described in the chapter on analyzing.

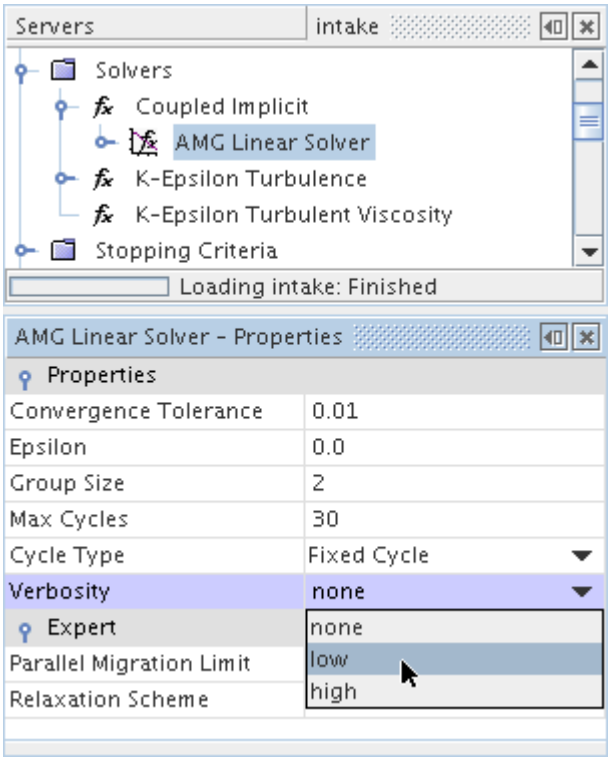
Although the computational expense of visualization or monitors can be small compared to the overall solution iteration process, it is not always the case. For simulations that contain a small numbers of cells, the cost of monitoring is significant. Especially if the monitoring includes a number of scalar and vector displayers as well as monitors that work over a volume (for example, maximum, minimum or volume integrals). To reduce this monitoring overhead, you can reduce the update frequency of the displayers or monitors, or reduce the quantity of displayers and monitors. However, if monitoring is important, the overhead is likely to be worthwhile.

The ability to observe the inner convergence of the AMG solution of the linear systems that are associated with solvers can be useful for obtaining the most efficient solution or for debugging the simulation.

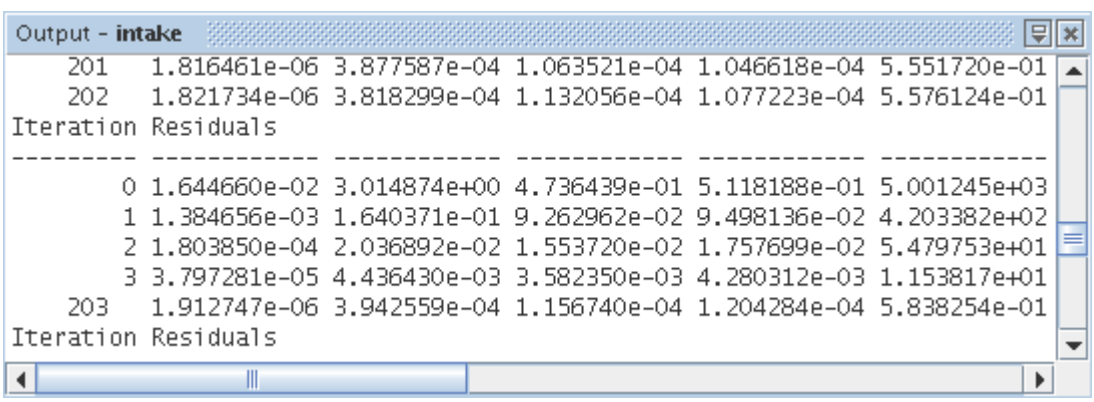
Observing the AMG Inner Convergence

Solvers in STAR-CCM+ that solve systems of algebraic linear equations resulting from the discretization of transport equations, rely on the algebraic multigrid (AMG) linear solver. A residual measure, described by Eqn. (2514), also indicates the solution of these linear equations. In some situations, it is beneficial to monitor

these residuals. To monitor the residuals, open the **AMG Linear Solver** node under the required solver. Then in the *Properties* window, set the *Verbosity* property to `low`.



When you continue iterating, extra information appears in the *Output* window. Nested within the customary residual output window are the AMG residuals. As a rule, the magnitude of the residuals is less important than how many iterations are being executed in a single iteration. Generally speaking, it is recommended that the number is between 2 and 5.



Changing the Simulation While Running

The client-server architecture of STAR-CCM+ allows you to make minor changes to the simulation while the solution is iterating.

The client-server architecture of STAR-CCM+ allows minor changes to be made to the simulation while the solution is iterating without stopping the solution:

- Opening plots and scenes
- Changing properties
- Adding monitors or reports

Therefore, it is acceptable to adjust solver properties such as Courant number (for coupled solvers) and under-relaxation factors (for segregated solvers) while continuing to iterate. However, if there is some risk of these changes causing solution convergence problems, then stop the solution, save the simulation, implement the changes, and continue with the solution.

However, major changes to the simulation are not advisable while the solution is iterating. As a general guideline, avoid any actions that involve the deletion of simulation objects or the creation of certain new ones (such as activating a new model) during this process.

Adding Monitors or Reports While Running

During a simulation run, it is possible to create new annotations, monitors, or reports, as well as new related annotations and plots, using the following methods:

- Creating a report
  - Creating an annotation from a report
  - Creating a monitor from a report
  - Creating a monitor and a plot at once from a report
- Creating a monitor
- Creating a plot from a monitor

The STAR-CCM+ server only handles such requests at the end of an iteration cycle. Pending requests appear in the object tree using the (**Pending**) icon. At the end of the iteration, the new object becomes available for use.

When you create a monitor during a simulation run, its *Trigger* property setting is `Inactive` by default. This default setting prevents error messages. To include the new monitor in the calculations:

- Set the *Field Function* property.
- Set the *Parts* property.
- Set any other appropriate properties.
- Select one of the two active *Trigger* options, `Time Step` or `Iteration`.

See Also:

Troubleshooting the Solution
Getting Past a Poor Initial Solution
Gradually Activating Models

Judging Convergence

Residual monitor plots are useful for judging the convergence (or divergence) of a solution.

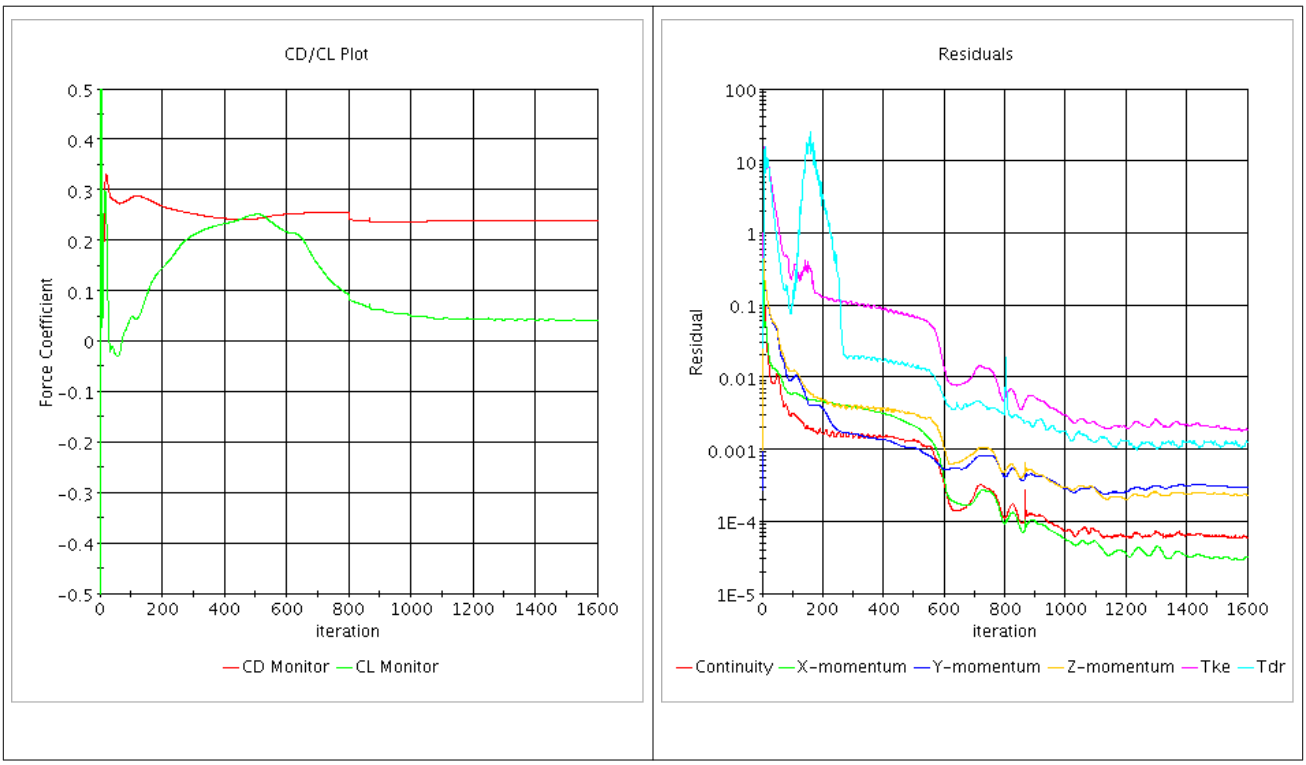
Residual monitor plots are useful for judging the convergence (or divergence) of a solution, and they are created automatically within every simulation. However, it is important to understand both the significance of residuals and their limitations. While it is true that the residual quantity tends toward a small number when the solution is converged, the residual monitors cannot be relied on as the only measure of convergence. The limitations of residuals are as follows:

- The amount that a residual decreases by depends on the particulars of the simulation. Therefore, a three-order-of-magnitude drop in residuals is possibly acceptable for one simulation, but not another. The initial guess also strongly influences the amount that residuals are reduced. If the initial solution satisfies the discretized equations perfectly, the residuals do not drop at all.
- There are two types of discretization errors: dissipative errors and dispersive errors. Dissipative errors are characteristic of first-order upwind schemes; they are inherently stabilizing and produce residual plots that tend to decrease monotonically. Dispersive errors are characteristic of second-order upwind schemes which tend to “smear” solutions less than first-order schemes. While dispersive errors tend to produce residual plots that are not monotonic. This outcome is generally an acceptable price to pay for the enhanced accuracy.

  In some cases, often because of poor mesh quality, dispersive errors result in oscillating solutions (that is, changing from one iteration to the next) within a few cells. The result is that the residual plots can indicate that the solution is not “converged”. You have a choice to either accept the solution, or to try to stabilize it by choosing a lower-order numerical scheme. Frequently, it is better to accept the solution.
- Residuals do not necessarily relate to quantities of engineering interest in the simulation such as integrated forces, pressure losses, or mass flow rates.

With the issues above in mind, it is advisable to monitor quantities of engineering interest, such as integrated forces, pressure changes, or mass flow rates as well as the residuals. STAR-CCM+ features such as scenes and plots can help you examine these quantities while the solution progresses. The features are described in detail in the chapter on analyzing.

The choice of the engineering quantity, as well as the convergence criterion, is your judgment call. In the example below, taken from a large external aerodynamics solution, both lift and drag coefficients are monitored as well as the residuals. It is clear that not all quantities reach an asymptotic limit at the same time. Use your judgment and decide which coefficient is the most critical.



Monitor-based stopping criteria can be used to stop the simulation when an appropriate level of convergence has been reached, based on an appropriate combination of residual monitors or monitors of other physical quantities.

See Also:

Plotting Results
Watching Residuals
Using Monitors as Stopping Criteria
Ramping Up Solution Accuracy

Saving Intermediate Solutions

This section deals with different approaches to saving the simulation while still iterating toward a final solution.

Three topics are discussed:

- Saving the simulation before making steering changes
- Saving multiple checkpoints
- Automatically saving at regular intervals

Saving Before Making Steering Changes

It is good practice to save the simulation before making changes to solver parameters.

The section on steering the solution describes how you can change the simulation while it is running. When solver parameters that could result in solution divergence are changed, such as under-relaxation factors or Courant numbers, it is good practice to first save the simulation. This step allows the saved solution to be

recovered in case divergence occurs after the change. The same can be said for other major changes to the simulation, such as models changes or boundary condition changes.

Saving Multiple Checkpoints

It can be desirable to compare solutions from two different time-steps or two different times in a transient calculation.

You can save a copy of the simulation to allow this comparison to be done at a later stage.

Contents:

Automatically Saving at Regular Intervals
Auto Save Properties
Saving with a Checkpoint File

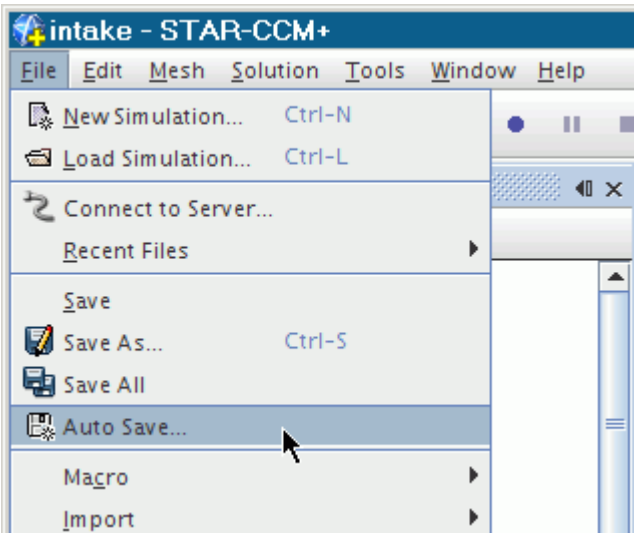
Automatically Saving at Regular Intervals

When running a long simulation (either interactively or in batch mode), it is useful to activate the auto save feature.

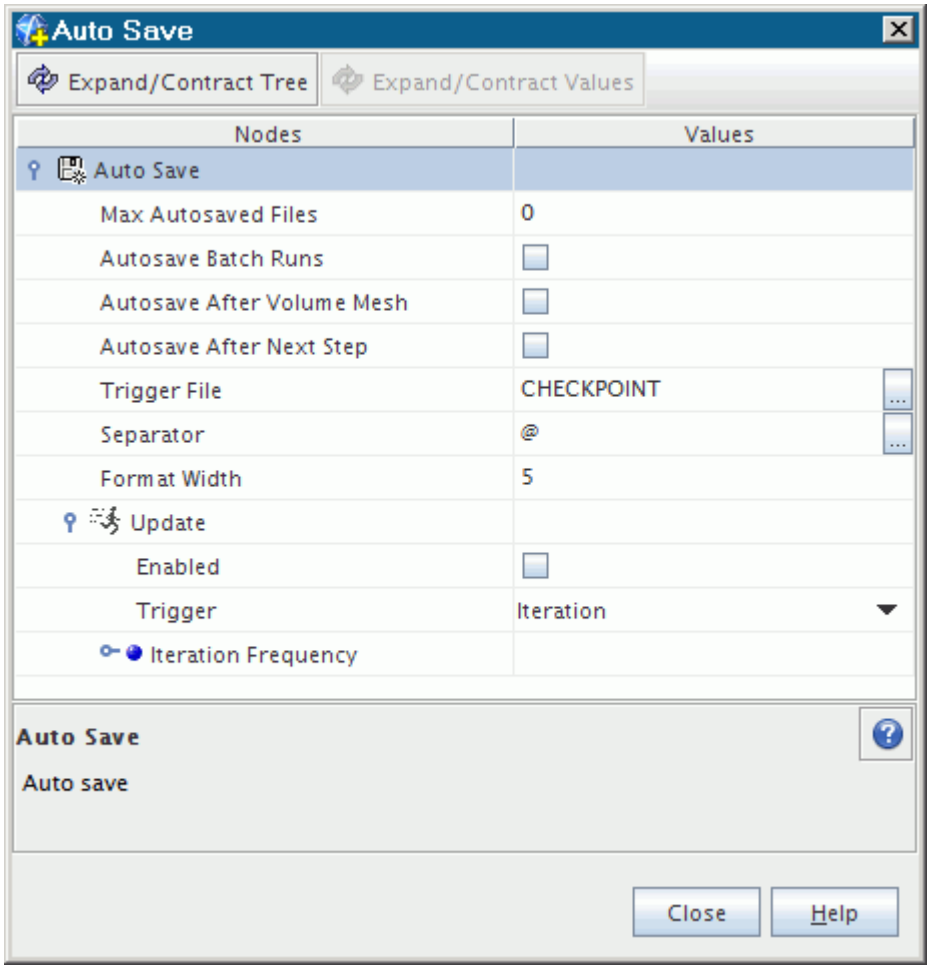
This option causes STAR-CCM+ to save the simulation automatically at regular intervals during the run, when the **Save**, **Save As...** and **Save All** commands are deactivated. The auto save feature is also useful for recovering from computer system crashes, or saving transient analyses at predetermined times. This section includes a description of the properties of the *Auto Save* dialog.

To set up the auto save feature:

1. Select **File** > **Auto Save...** from the menu.



The *Auto Save* dialog appears.



2. Activate the auto save feature:
   a) To expand the *Update* property, use the node handle.
   b) Activate the *Enabled* checkbox.
3. Define the number of saves that you want STAR-CCM+ to create in the *Max Autosaved Files* property.

   This property controls the number of extra files that are created. The default setting of zero instructs STAR-CCM+ to keep overwriting the same file—the same as executing the **Save** command.
4. To save the simulation at the end of the run in a batch job, tick the *Autosave Batch Runs* checkbox.

   It is not necessary to specify this action in the macro. This feature works even if the simulation diverges.
5. If you want the simulation to save automatically after generating a volume mesh, activate the *Autosave After Volume Mesh* checkbox.
6. To save the simulation at the end of the next iteration or time-step, activate the *Autosave After Next Step* checkbox.
7. To trigger the auto save operation with a file, use the *Trigger File* property.
8. Enter the type of separator that you want to use in the *Separator* property.

   By default, when STAR-CCM+ saves copies of the simulation file, it places an `@` character to separate the base file name and the number of the trigger event. If the use of this character in this manner causes any problems with your system, you can enter a different one as the separator. If you attempt to leave this field blank, STAR-CCM+ puts the `@` character there by default.

9. Select the *Trigger Type* from the drop-down menu.

   This property specifies how the auto save operation is triggered. See Trigger Types. Unless you are running a transient analysis, leave the setting at `Iteration`.

10. To control the number of intervals that pass between auto saves, expand the **Trigger Frequency** node and enter the value that you want.

    See Trigger Properties.

11. Click **Close**.

STAR-CCM+ saves automatically during the next run at the intervals you specified. Depending on the Max Autosave Files that you specified, STAR-CCM+ runs and saves a separate copy of the simulation file at each instance of the auto save operation. These copies are saved in the same directory as the original simulation file. When the specified number of files is reached, the next auto save action deletes the oldest copy of the file. The name of each copy includes the existing simulation name, along with the number of the particular interval to ensure that each file is unique.

To deactivate this feature, reopen the *Auto Save* dialog and clear the checkbox of the *Enabled* property.

Note: You can use the auto save feature with the Thermal Comfort Wizard to save the thermal properties of the occupants during a TCM simulation.

See Also:

Running the Client in Batch

Triggers and Update Events

Auto Save Properties

This section describes the properties of the Auto Save feature.

You can use the following properties to activate and customize this feature:

Max Autosaved Files	Sets the number of copies of the simulation file to be created.	
Autosave Batch Runs	Saves at the end of a batch run when the simulation has not been saved since the last iteration.	
	Activated	Automatically saves simulations at the end of a batch run.
	Deactivated	Does not automatically save simulations at the end of a batch run.
Autosave After Volume Mesh	Saves when a volume mesh is generated.	
	Activated	Automatically saves simulations when a volume mesh is created.
	Deactivated	Does not automatically save simulations when a volume mesh is created.

Autosave After Next Step	Saves when the next iteration or time-step is completed. This option triggers the same behavior as the *Trigger File* property. After the auto save occurs, the checkbox is deactivated automatically to avoid triggering the auto save again.	
	Activated	Automatically saves simulations after the next step.
	Deactivated	Does not automatically save simulations after the next step.
Trigger File	Uses a file to control the auto save operation.	
Separator	Specifies the character that appears between the base file name and the number of the trigger event (`@` by default).	
Format Width	Width of the iteration, time, or step number in the formatted file name.	
Update	When expanded, this property makes available the controls that are listed below.	
Enabled	Activates or deactivates the auto save feature.	
	Activated	Auto save is turned on.
	Deactivated	Auto save is turned off.
Trigger Type	Provides choices for triggering the automatic save. See Trigger Types.	
Update Frequency	This property controls the number of trigger events between saves. See Trigger Properties.	

The settings are saved in the simulation database.

See Also:

Saving with a Checkpoint File
Trigger Types
Triggers and Update Events

Saving with a Checkpoint File

The *Trigger File* property of the *Auto Save* dialog allows you to specify the pathname of a file (named `CHECKPOINT` by default) that, once in place, triggers the auto save operation.

This feature can help you control the auto save process when accessing the *Auto Save* dialog is inconvenient, for example in a cluster environment with the compute nodes in a separate subnet.

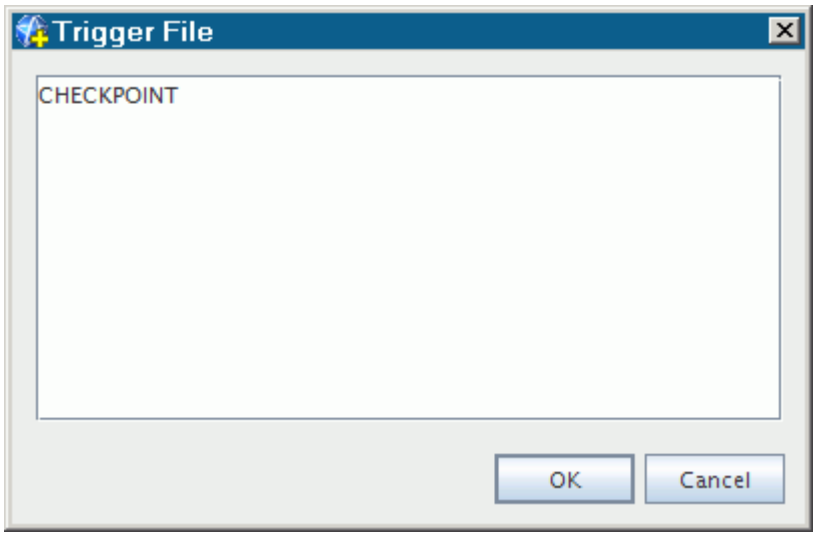
If STAR-CCM+ finds the trigger file, the auto save occurs at the end of the current step (iteration in a steady case, time-step in an implicit unsteady case).

This feature is designed to use a particular file once. After the auto save, STAR-CCM+ renames the trigger file by appending a tilde character (`~`) to the filename.

To set the path name of the file:

1. Activate the *Auto Save* dialog.
2. Modify the name and the path in the *Trigger File* property.

   You can either modify the name directly with a text entry or click the ellipses button and enter the path in the *Trigger File* dialog.



   If you specify a relative path, or no path at all, STAR-CCM+ checks for the trigger file in the same directory as the simulation file.
3. To use this feature while a simulation is running, you can simply add a file as follows:
   - On Windows, place a file with the correct name in the correct path; even an empty text file would suffice for this purpose.
   - On Linux, use the line command:

```
% touch CHECKPOINT
```

   Note: This feature does not execute the save action immediately because STAR-CCM+ checks once every 60 seconds for the trigger file.

Troubleshooting the Solution

Divergence of the solution is the situation where cell residuals increase from one iteration to another. If left, it can eventually result in a floating point exception being reported. Generally, the behavior of the residual monitor plots indicates when divergence happens.

An increase in one or more of the residual monitors during the first few iterations is normal and acceptable. For example, consider that a (constant) initial solution often satisfies the discretized equations exactly, everywhere except at the boundaries. The boundary conditions propagate into the interior as the solution iterates, causing the global residuals to increase for several iterations before decreasing. However, if this situation does not change after 5-10 iterations, it is likely to be an indication of divergence.

Solution divergence could be due to one of two reasons:

- In a case with difficult physics, the initial conditions specify that the initial solution is inappropriate and special steps must be taken to get past a poor initial solution.
- There is some other fundamental issue with the simulation, such as the mesh quality, the boundary conditions or the solution parameters, and special steps must be taken to debug the simulation setup.

If divergence occurs, you do not need to exit from the simulation. Instead, you can clear the solution and run it again after investigating the cause of divergence and making the required changes to the simulation.

Contents:

Getting Past a Poor Initial Solution
Debugging the Simulation Setup

See Also:

Obtaining a Solution Interactively
Using Initial Conditions
Clearing the Solution
Running the Simulation
Changing the Simulation While Running

Getting Past a Poor Initial Solution

A possible cause of divergence can be related to the initial solution being radically different from a converged solution. This situation occurs because implicit numerical methods rely on linearization techniques to advance the solution of equations toward convergence, and it is possible that the linearization is inadequate to advance the solution.

There are two techniques to deal with this issue:

- Improve the initial conditions. Alternatively, it can be achieved using techniques such as gradually activating models, changing boundary conditions, activating unsteady models from a steady solution, ramping up solution accuracy, or adjusting the normalization factor.
- Advance the solution less by ramping up Courant numbers and relaxation factors.

When using these techniques, be sure to follow the guidelines which tell you if you are required to stop the solution or not before you change the conditions. These guidelines are described in the section that deals with changing the simulation while running.

Contents:

Gradually Activating Models
Changing Boundary Conditions or Ramping Boundary Values
Activating Unsteady Models from a Steady Solution
Ramping Up Solution Accuracy
Adjusting the Normalization Factor
Ramping Up Courant Numbers and Relaxation Factors

See Also:

Using Initial Conditions
Changing the Simulation While Running

Gradually Activating Models

There is no need to activate all the models at the beginning of the solution; models can be activated during the solution.

Examples of where it can be beneficial to activate models during the solution process are as follows:

- Compressible flows can be started inviscid (or laminar) before turning on turbulence. However, take care to ensure that the meshes are appropriate for the substituted models. It is possible that the Courant numbers and relaxation factors need ramped up after the models are changed.
- Unsteady models can be activated from a steady solution.

See Also:

Changing the Simulation While Running
Ramping Up Courant Numbers and Relaxation Factors
Activating Unsteady Models from a Steady Solution

Changing Boundary Conditions or Ramping Boundary Values

In some simulations, it can be difficult to start the solution with the final boundary conditions and boundary values.

Possible examples of where it is advantageous to start with one set of boundary conditions or boundary values before changing them are as follows:

- When solving high-Mach number flows over bluff bodies, it can be beneficial to ramp-up the free-stream Mach number gradually.
- In some complex turbomachinery cases that use a combination of stagnation and pressure boundaries, establishing the correct flow direction at the inlet boundary can be difficult. It is therefore prudent to obtain

an initial solution with another sort of inlet boundary (mass flow or velocity inlet) before switching to a stagnation inlet.

When changing the boundary type, take care to ensure that the types are compatible. For example, exercise caution in the following scenarios:

- For a highly compressible flow, it is possible that a velocity inlet is not a well-posed type.
- In incompressible flows, exercise caution when changing a flow split outlet to a pressure outlet. The pressure at a flow split outlet is not specified and is potentially considerably different from the pressure that is specified at a pressure outlet.

Activating Unsteady Models from a Steady Solution

It is common practice to start an unsteady simulation with a steady solution.

Potentially advantageous examples of this practice are:

- When doing a simulation of vortex shedding from a bluff body, you can often first obtain a steady-state solution, though there is no guarantee that the solution you obtain is converged. In some cases, if a converged steady-state solution exists, it is possible that a change in inlet boundary conditions for a few time-steps is necessary. This action can cause a perturbation and prompt the shedding to occur.
- In some turbomachinery problems where a transient, sliding mesh calculation is required, you can often start with a steady solution using the multiple-reference frame (MRF) approach. Then you can initiate the unsteady and mesh motion models.

See Also:

Gradually Activating Models

Ramping Up Solution Accuracy

It can be beneficial to start the solution with a first-order numerical scheme.

As discussed in the section on judging convergence, the dissipative errors resulting from first-order solutions are stabilizing and can enhance convergence. Therefore, it can be beneficial to start the solution with a first-order numerical scheme. For example:

- Models involving convective transport equations have either a *Convection* property (for segregated models), or a *Discretization* property (for the Coupled Flow model), that can be set to first order. The *Convection* property for segregated models affects only the upwind scheme for convection, leaving the diffusion and other gradients still fully second-order. The *Discretization* property for the Coupled Flow model neglects higher-order reconstruction for convection and diffusion.
- Solvers that are related to models that involve convective transport equations have a *Reconstruction Zeroed* expert property. Irrespective of the model type, this property causes reconstruction gradients to be neglected, effectively resulting in a first-order scheme.

See Also:

Judging Convergence

Adjusting the Normalization Factor

To improve the appearance of the residuals, especially with turbulent cases, adjust the normalization factor.

The initial conditions for turbulence (and also velocity) often lead to situations where there are low levels of turbulence within the first 5 iterations, which is when STAR-CCM+ calculates the residual normalization. As a result, as turbulence develops within the domain, the normalized turbulence peaks at a value much larger than 1.0. If you are concerned that the final value of this turbulence can hover higher than the other residuals, change the normalization factor to get a peak residual of 1.0.

For details on how to set the normalization factor, see Normalizing the Monitor Data.

Ramping Up Courant Numbers and Relaxation Factors

This section provides information on modifying values of Courant number and under-relaxation factors.

The default values of Courant number and the under-relaxation factors in STAR-CCM+ work in many situations, but are unlikely to be optimal in all situations, particularly when dealing with poor initial solutions. Suggestions for suitable alternative values are contained in the troubleshooting sections of the following relevant modeling guides:

- Coupled Flow and Coupled Energy
- Segregated Flow
- Segregated Fluid Energy
- Turbulence

See Also:

Gradually Activating Models

Debugging the Simulation Setup

When a simulation is diverging or a floating-point error appears, try to investigate where the problem is occurring; either in the solution process or its physical location in the mesh.

STAR-CCM+ provides several tools to aid debugging that are not necessarily typically employed. Examples include:

- Using maximum and minimum reports
- Using threshold parts and cell surfaces
- Turning on temporary storage
- Zeroing reconstruction gradients
- Omitting secondary gradients

Contents:

Using Maximum and Minimum Reports
Using Threshold Parts and Cell Surfaces

Turning on Temporary Storage
Zeroing Reconstruction
Omitting Secondary Gradients

Using Maximum and Minimum Reports

In cases where the solution is “blowing” up, it is a good idea to create report-based monitors and to plot them to see when the solution starts diverging.

Typically a maximum report of velocity magnitude is a good indicator. In addition, minimum values of positive-definite quantities such as density, temperature, or turbulence quantities can be equally informative.

See Also:

Using Report-Based Monitors
Creating a Monitor Plot

Using Threshold Parts and Cell Surfaces

Once it has been determined that the solution is growing unbounded, it is useful to identify the physical location of the problem. Locating the problem can help identify if it is occurring near a boundary, or near a poor-quality cell.

Threshold parts are often the best way to view specific cells. Set the threshold to the required value (of, say, velocity magnitude), and once the threshold part is created, drag it into the scene. You can also create a cell surface from any other surface which lets you visualize a cell with a bad solution on a cutting plane.

Turning on Temporary Storage

You can turn on temporary storage for a particular solver in the Expert Properties category of that solver.

That action adds the field functions that let you view data such as Ap coefficients, residuals, corrections, reconstruction gradients, and cell gradients.

The Ap coefficients are the diagonal coefficients in the matrix of the linear system. Ensure that they are always positive-definite (that is, greater than 0).

The residuals are the imbalance in the discretized equations. The corrections are what is added to the solution of the previous iteration to get the updated solution. Generally, if the solution is not converging you can look where the residuals or the corrections are comparatively large. (Large corrections typically correspond to large residuals.)

Reconstruction gradients are used to interpolate cell values to faces, while gradients are used to compute values such as diffusion and strain rates. Visualizing these quantities can help to diagnose if something is going wrong at a flow boundary, say, where large gradients are not expected.

Field functions added when temporary storage is turned on are not saved to the simulation file. It is necessary that at least one step is iterated before the additional field functions become available.

Zeroing Reconstruction

When you zero reconstruction, the interpolation to the faces is a simple arithmetic averaging.

This averaging causes a less accurate solution, but improves the robustness. It is a useful thing to do if your solution is diverging, since it can reduce the magnitudes of variables like pressure and velocity, and keep the parts that are diverging more “local”. However, a converged solution with reconstruction zeroed does not have high accuracy.

Omitting Secondary Gradients

Secondary gradients (also called cross-diffusion) are used in the diffusion terms and also in the pressure-velocity coupling for SIMPLE.

When the skewness angle gets large (89 degrees or more), these secondary gradients can become destabilizing. You can turn them off in the model, such as the segregated flow model. Do not accept a solution with secondary gradients off, since it is inaccurate unless the mesh is orthogonal. However, it can provide a clue and show if the diffusion terms are causing problems with the solution. If that is the case, improve the mesh and reduce the skewness angles. There are two separate switches: one for the boundary faces, and one for the interior faces.

See Also:

Checking The Skewness Angle

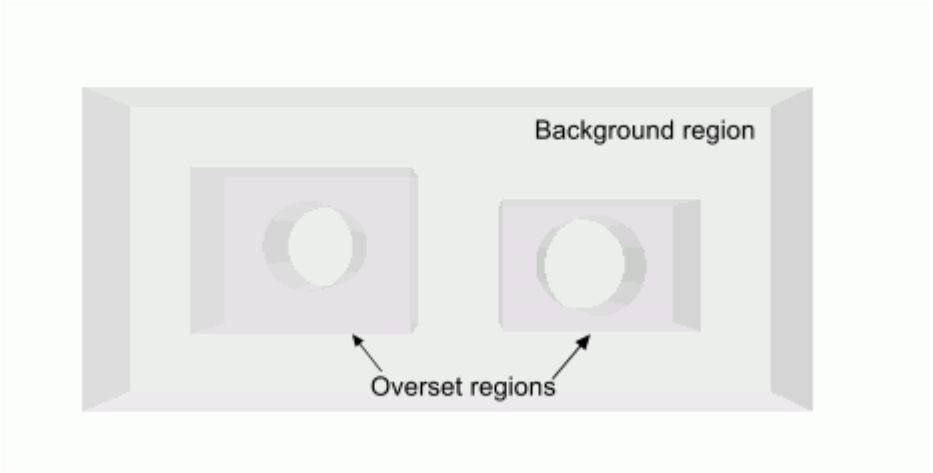
Using Overset Meshes

Overset meshes, also known as “Chimera” or overlapping meshes, are used to discretize a computational domain with several different meshes that overlap each other in an arbitrary manner. They are most useful in problems dealing with multiple or moving bodies, as well as optimization studies.

In most cases, using overset meshes does not require any mesh modification after generating the initial mesh, thus offering greater flexibility over the standard meshing techniques.

Overview

Any study involving overset meshes has a background region enclosing the entire solution domain and one or more smaller regions containing the bodies within the domain. A schematic is shown below:



In an overset mesh, cells are grouped into active, inactive, or acceptor cells. Within active cells, discretized governing equations are solved. Within inactive cells, no equation is solved, however, these cells can become active if the overset region is moving. Acceptor cells separate active and inactive cells in the background region and are attached to the overset boundary in the overset region. Acceptor cells are used to couple solutions on the two overlapping grids. Variable values at donor cells of one mesh express variable values at acceptor cells in the other mesh, through interpolation. The donor cells are the active cells from the other mesh that are nearest the acceptor cell. The available interpolation options are described in the coupling regions section.

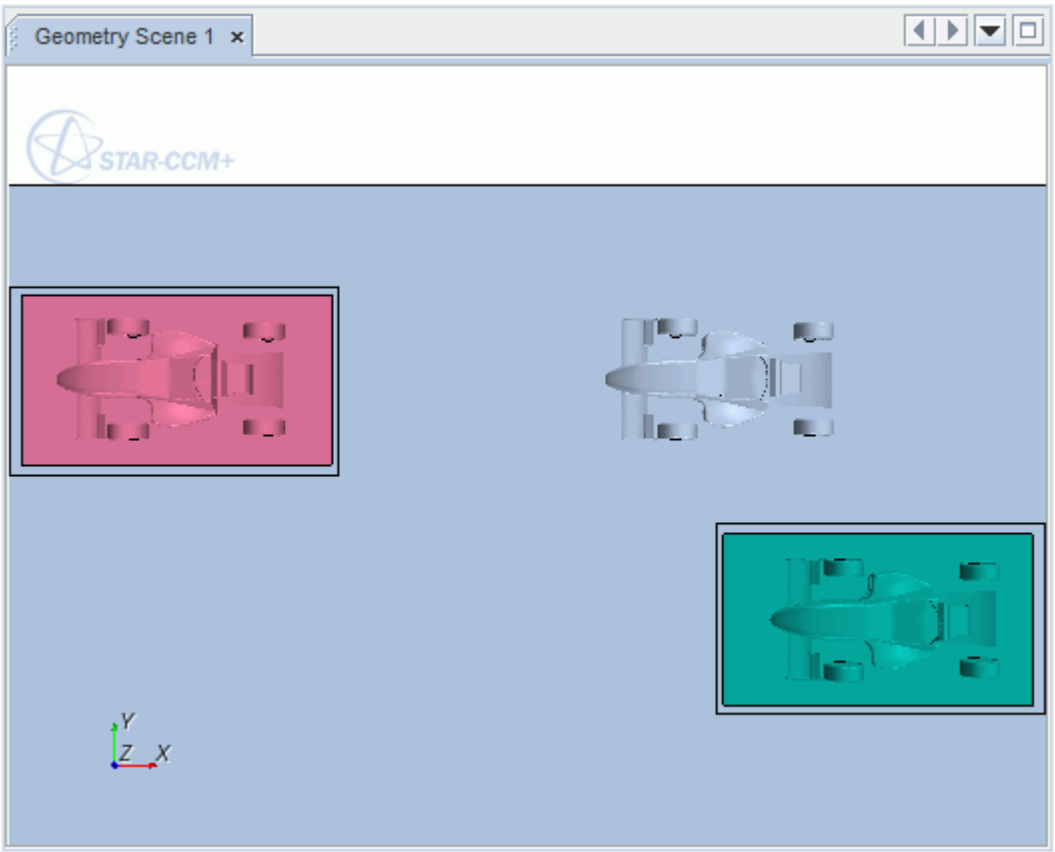
The solution is computed for all active cells in all regions simultaneously, that is, the meshes are implicitly coupled. When a reference is made to the variable value in an acceptor cell of one mesh within the discretized equations, a blend of variable values at donor cells from the other mesh is used. This value is reflected directly in the coefficient matrix of the algebraic equation system. This tight coupling of the overset and

background regions allows for a solution that is within an arbitrary low level of iteration errors. The rate of convergence of the iterative solution method is therefore similar to that of a single mesh of the same resolution. Refer to the overset mesh methodology section for further description.

Applications

Overset meshes are most applicable in the following cases:

- Parametric studies, such as identical bodies at different relative positions or different bodies in the same environment. Fewer meshes are generated for these cases when using overset meshes.
- Complicated body motions, such as those involving bodies that are close together, have intersecting paths, or have large motions. Below is an example of two cars overtaking the third. Overset meshes are used for the two colored cars in motion.
-



Current Implementation

The overset mesh feature is supported for use with the following:

- Segregated and coupled flow
- DFBI
- Discrete Element Method (DEM)
- Motion models and reference frames
- Single phase

- VOF Multiphase
- Lagrangian Multiphase
- Eulerian Multiphase
- Fluid Film
- Combustion
- Dispersed Multiphase Model (DMP)
- Passive Scalar

Current implementation of the overset mesh feature has the following limitations. Some of these restrictions will be lifted in future releases.

- Mass conservation is not strictly imposed with the current interface treatment. The usual error is on the order of 0.1% or less, that is, the error does not affect the solution to the first 3 - 4 significant digits.
- Regions that are involved in creating an overset mesh must be assigned to the same physics continuum.
- When using overset mesh together with the fluid film model, certain requirements must be met. For more information, see Setting up Overset Mesh with Fluid Film.
- The feature is not supported for the following STAR-CCM+ models:
  - De-Fogging
  - Surface-to-Surface Radiation
  - Harmonic Balance
  - Grid Sequencing
  - Batteries

Current Overset Capabilities

- Overlapping of multiple overset meshes
- Overset regions can extend beyond the solution domain

Working with Overset Grids in STAR-CCM+

The subsequent sections provide information on setting up and using overset meshes in STAR-CCM+:

- Summary of the overset mesh user procedure
- Creating meshes for the background and overset regions
- Creating an Overset Mesh Interface
- Guidelines
- Overset mesh methodology
- Overset mesh field function
- Troubleshooting

Contents:

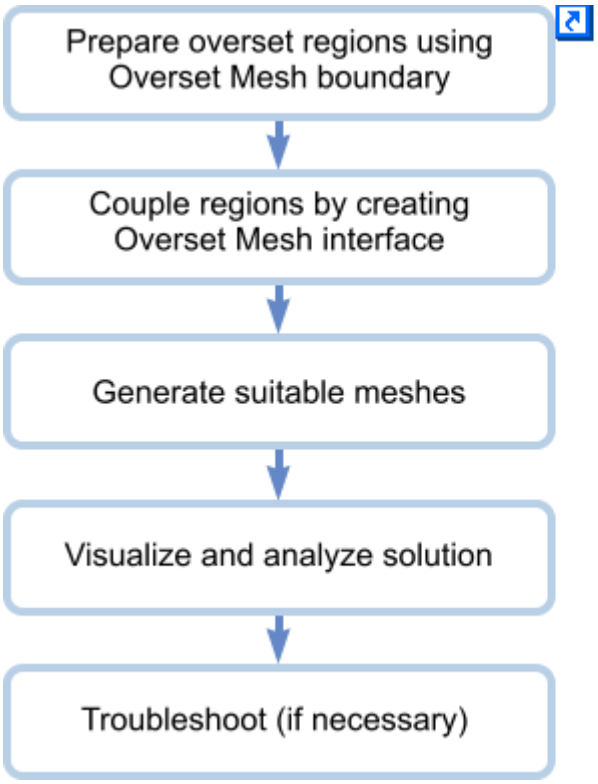
Understanding the Overset Workflow
Preparing Overset Regions
Coupling Regions
Enforcing Mass Conservation

Dynamic Overset Behavior
Overset Topology
Setting up Overset Mesh with Fluid Film
Generating Suitable Meshes
Visualizing and Analyzing Solution
Activating Load Balancing for Parallel Overset Mesh Simulations
Overset Mesh Best-Practice Guidelines
Overset Setup Examples and Best Practices
Overset Mesh Methodology
Overset Mesh Reference
Overset Mesh Field Function Reference
Troubleshooting Overset Mesh
Overset Mesh Bibliography

Understanding the Overset Workflow

The general workflow for using an overset mesh involves the steps that are outlined below.

Two tutorials, transonic flow over an airfoil and DFBI Lifeboat, using overset meshes are available in the tutorial guide.

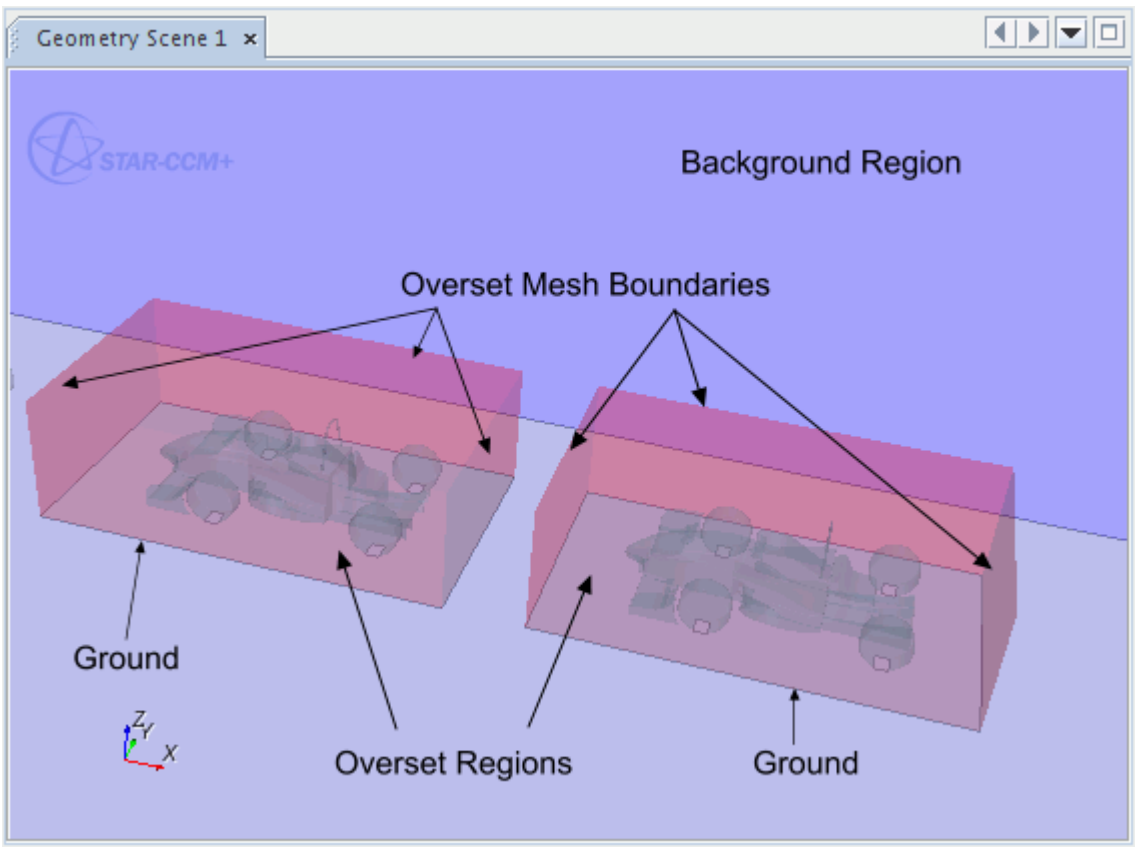


Preparing Overset Regions

Using an overset mesh requires creating not more than one background region, which is a closed surface solution domain, and one or more overset regions, which contain the physical bodies. More than one overset region can be used on top of the background region. These overset regions can also overlap each other.

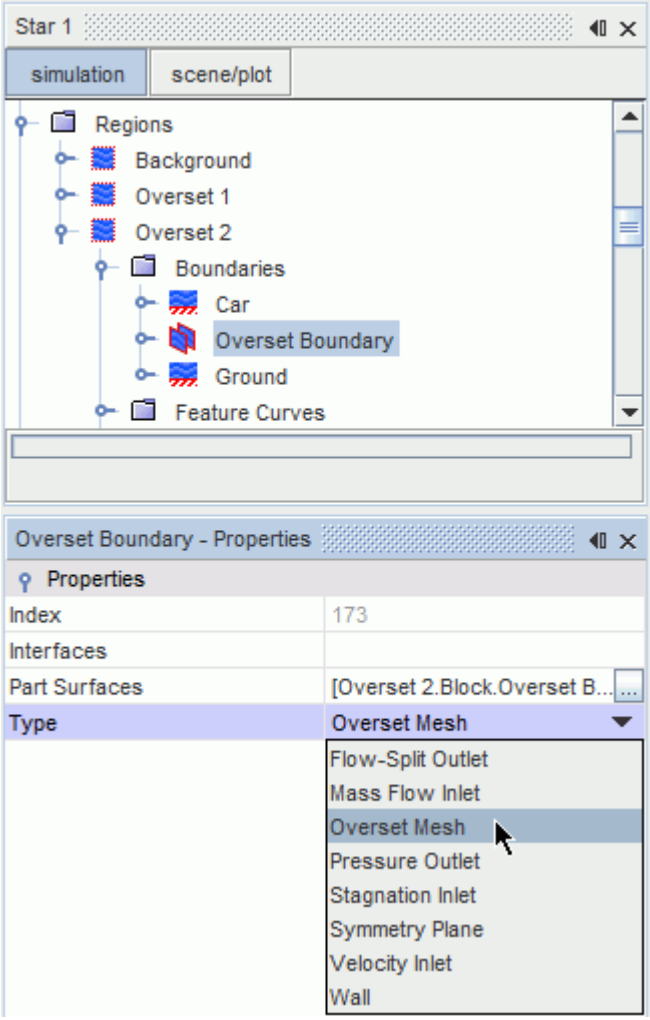
Setting the Overset Mesh Boundary

The overset mesh is created using the Overset Mesh boundary, which is the outer boundary of the overset region that is expected to be coupled with the background mesh. If part of the overset region boundary coincides with the boundary of the background region, that part must be of the same boundary type as the background region. An example is shown below. The ground boundary in this case is part of both the background and the overset regions. It is not considered to be part of the Overset Mesh boundary.



To assign an Overset Mesh boundary:

1. Navigate to the designated overset mesh boundary under the **Boundaries** node in the overset region.
2. In the *Properties* window, set the type to **Overset Mesh**.


Star 1
simulation
scene/plot
Regions
Background
Overset 1
Overset 2
Boundaries
Car
Overset Boundary
Ground
Feature Curves
Overset Boundary - Properties
Properties
Index
173
Interfaces
Part Surfaces
[Overset 2.Block.Overset B...
Type
Overset Mesh
Flow-Split Outlet
Mass Flow Inlet
Overset Mesh
Pressure Outlet
Stagnation Inlet
Symmetry Plane
Velocity Inlet
Wall

Coupling Regions

The Overset Mesh Interface is used to couple the overset regions with the background region. If you want to define overset regions that overlap each other, create overset mesh interfaces between each set of overlapping overset regions. The interfaces for the overlapping regions are in addition to the overset mesh interface between the respective overset region and the background mesh.

To create the interface, the overset regions must have an Overset Mesh boundary defined. The option to create this interface only becomes available once a physics continuum is defined and assigned to both the background and overset region. This interface ensures that both regions use the same physics continuum.

You have two options for creating such interfaces:

- Creating Overset Mesh Interfaces
- Creating Overset Mesh Zero Gap Interfaces

Contents:

Creating Overset Mesh Interfaces
Creating Overset Mesh Zero Gap Interfaces

See Also:

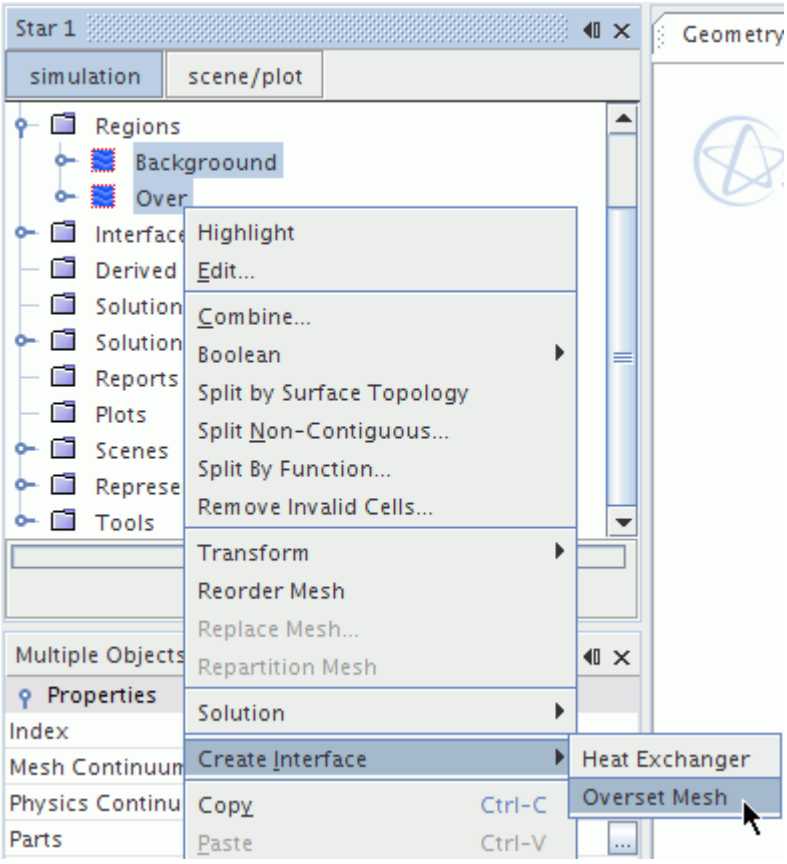
Preparing Overset Regions

Creating Overset Mesh Interfaces

Use this type of overset mesh interface for any overset mesh simulation except for a zero gap configuration

To create the interface between the background and the overset region:

1. Select the background region and the overset regions. You can use <Ctrl><Click> to select both regions.
2. Right-click on the selected regions and choose **Create Interface** > **Overset Mesh**.



An **Overset Mesh 1** node is added under **Interfaces**.

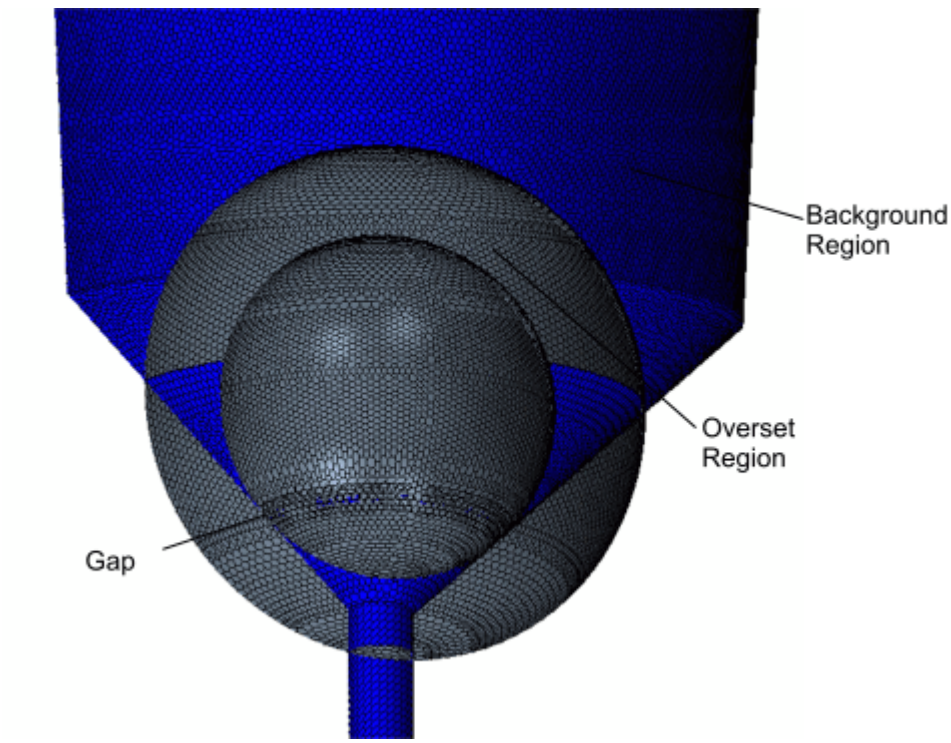
To create an interface between overset regions that overlap each other:

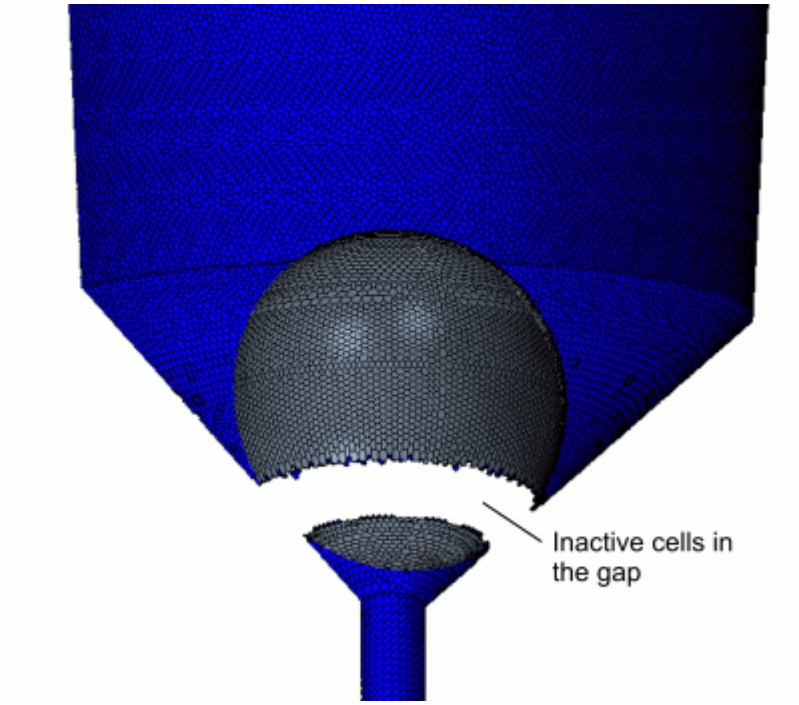
3. Select each pair of overlapping overset regions. You can use <Ctrl><Click> to select both regions.
4. Right-click on the selected regions and choose **Create Interface** > **Overset Mesh.**
5. Select the **Interfaces** > **Overset Mesh 1** node and set the Overset Mesh Interface Properties.

Creating Overset Mesh Zero Gap Interfaces

Use this type of overset mesh interface if you expect a zero gap configuration to occur in your simulation. Zero gap configuration means that two regions, either one background region and one overset region, or two overset regions, approach each other very closely or make contact.

The following diagrams of a ball-type check-valve show an example of such a configuration:





The upper diagram shows the mesh in its initial state, that is, before the simulation starts. The overset region (gray) consists of the mesh of the ball. The overset region extends beyond the boundaries of the background region (blue). You can see a very small gap between the background region and the overset region. The distance between the background region and the overset region is less than two cell layers and the zero gap algorithm applies.

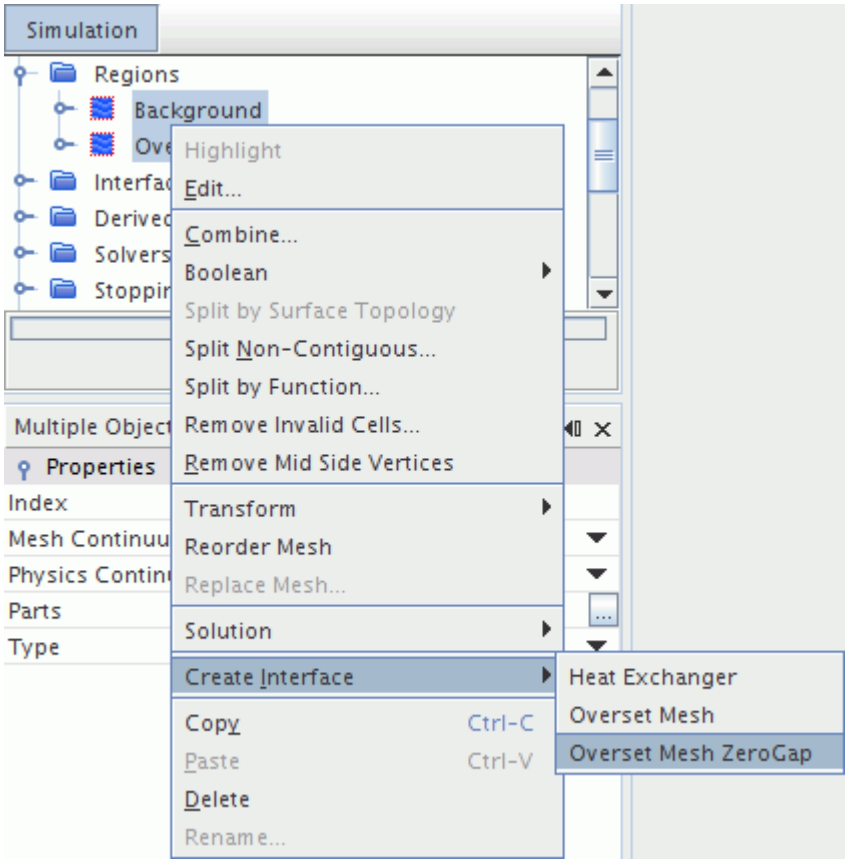
The lower diagram shows the mesh after the simulation started. You see that the cells (overset and background) in the gap and the overset cells outside of the background region are inactive.

Note: The Overset Zero Gap Distance Field Function can be used for determining the gap distance on a boundary; the minimum gap distance can be monitored and plotted for cases like this one—where the minimum gap size may vary with flow rate and time (see Overset Mesh Field Function Reference).

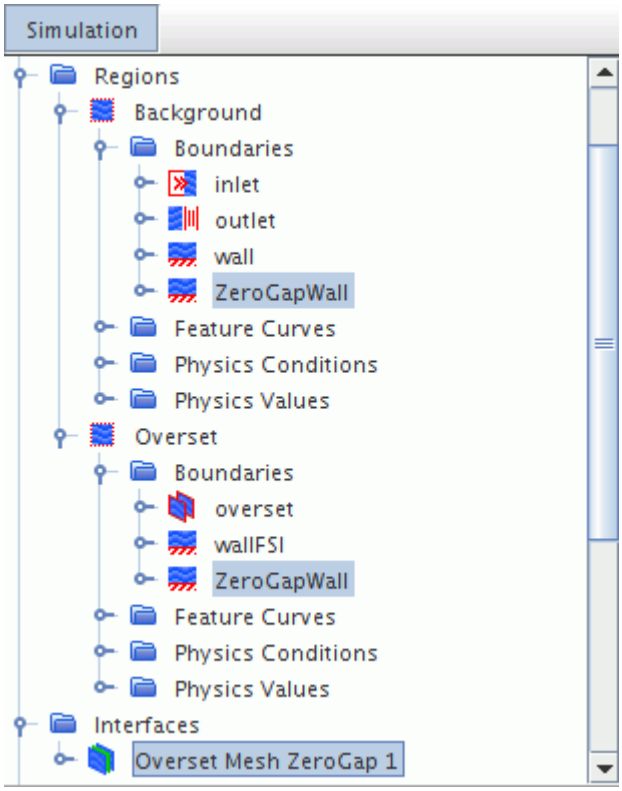
When you create the overset mesh zero gap interface between two regions, a zero gap wall boundary is automatically created in each of the regions. During the simulation, if the Overset Assembly process detects that the distance between the two zero gap wall boundaries of both regions is less than the default value of three cell layers, the cells in the gap become inactive. If you wish, you can change the number of cell layers below which STAR-CCM+ deactivates the cells. Cell faces between active and inactive cells are placed within the zero gap wall boundaries.

To create an overset mesh zero gap interface:

1. Multi-select the background region and the overset regions.
2. Right-click on the selected regions and choose **Create Interface** > **Overset Mesh** > **Overset Mesh ZeroGap**.



A **ZeroGapWall** boundary and **Number of Zero Gap Layers** node are added under each **Boundaries** node. An **Overset Mesh ZeroGap 1** node is added under the **Interfaces** node.



3. (Optional) Select the **Boundaries** > **Number of Zero Gap Layers** node and set the Number of Zero Gap Layers Properties.

   For a given mesh resolution in the gap, this option lets you change the width of the gap. The cells in the gap are deactivated and flow through it is prevented.

4. Select the **Interfaces** > **Overset Mesh ZeroGap 1** node and set the Overset Mesh Interface Properties.

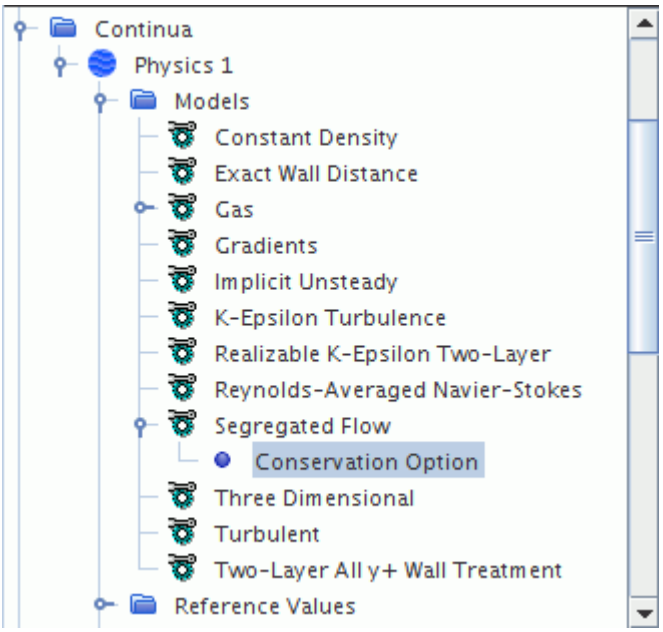
Enforcing Mass Conservation

The overset method is not conservative and comes with an interpolation error at the overset interface leading to a mass imbalance. To enforce mass conservation for the overset grid method, you must select the appropriate conservation option for the segregated flow model.

Before the **Conservation Option** node appears as a sub-node of the **Continua** > **[Physics Continuum]** > **Models** > **Segregated Flow** node, you must create at least one overset interface between a background and an overset region.

To enforce mass conservation:

1. Select the **Continua** > **[Physics Continuum]** > **Models** > **Segregated Flow** > **Conservation Option** node and set *Option* to the appropriate Conservation Option Properties.



Dynamic Overset Behavior

Use dynamic overset behavior for overset mesh applications where an automatic switch from a wall boundary type to an overset boundary type or vice versa is required.

Examples for such applications are:

- Closely counter-rotating objects such as screw pumps
- IC engine piston motion where the cylinder has a variable radius

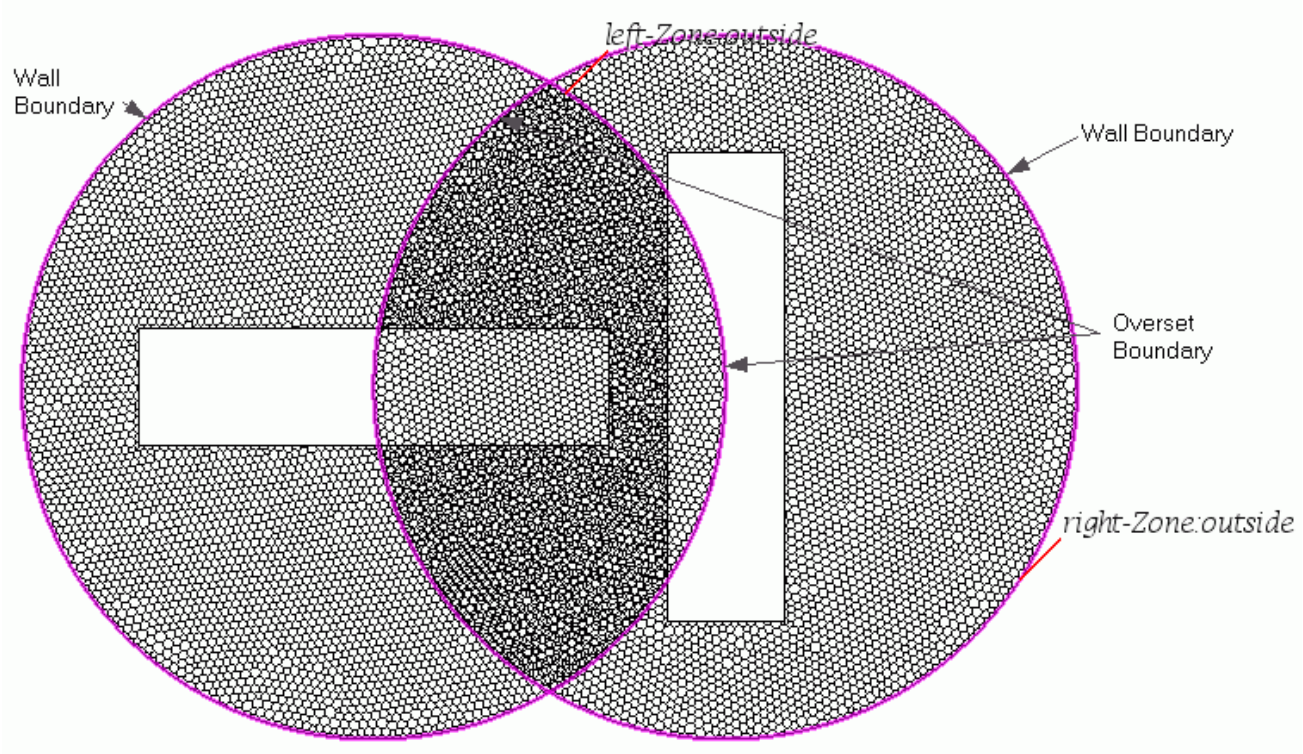
The dynamic wall boundary is an option that is available for wall boundaries after an overset mesh interface is created. There are two main scenarios for this option:

- Overset simulations without background mesh
- Overset simulations with background mesh

Simulations Without Background Mesh

When you activate the dynamic overset behavior to model a scenario without a background mesh, the hole-cutting process checks whether a boundary face resides inside or outside of the other overset region. If the hole-cutting process detects that a boundary face lies within the other overset region, the cells that are adjacent to the boundary become acceptor cells. The previous wall boundary becomes an overset boundary or the overset boundary becomes again a wall boundary depending on position.

The following diagram shows two counter-rotating square objects. The objects are embedded within two overset regions. There is no background mesh present. The highlighted boundaries use the dynamic overset behavior.

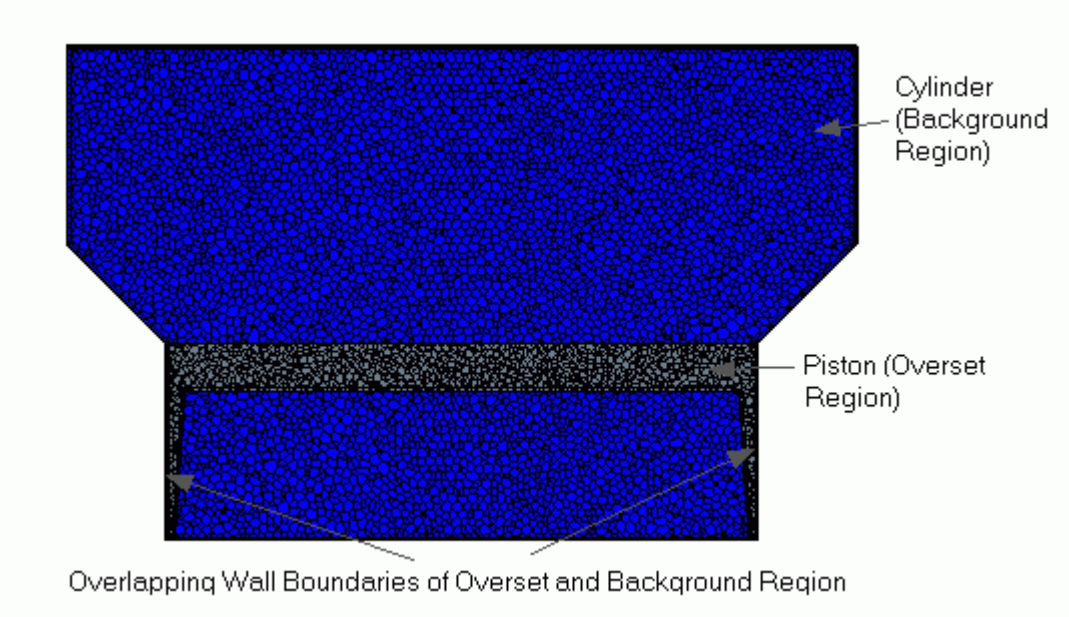


The advantage of this functionality is the fact that you do not need to set up a background mesh.

Simulations with Background Mesh

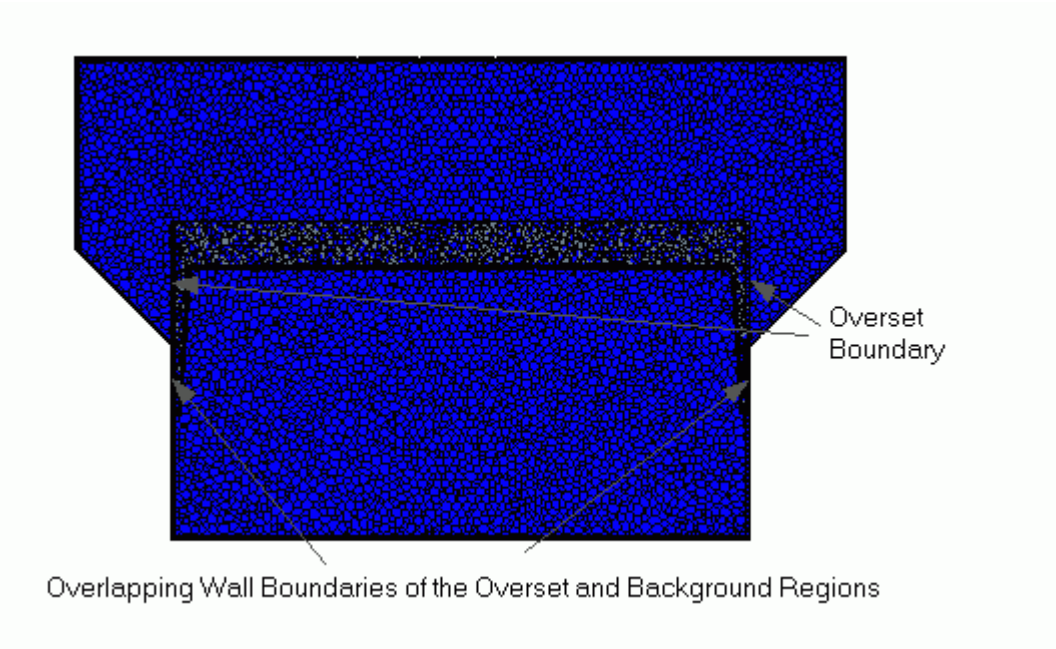
When you activate the dynamic overset behavior to model a scenario with a background mesh, the hole-cutting process checks whether the moving wall boundary of the overset region still coincides with the wall boundary of the background mesh. If the hole-cutting process detects that a wall boundary face of the overset region no longer coincides with a wall boundary face of the background region, the cells that are adjacent to the boundary become acceptor cells. The previous wall boundary of the overset region becomes an overset boundary. If such an overset boundary moves so that it coincides with a background region wall boundary, it reverts to a wall boundary.

The following screenshot shows an example application—the translational motion of a simplified piston within a cylinder of variable radius.



Here, the overset region (gray), that is, the piston is in its initial position. The side wall boundaries of this overset region coincide with the side wall boundaries of the background region (blue), which is the cylinder. You must set up the sliding side walls of the piston as a wall boundary (and not as an overset boundary), because if part of the overset region boundary coincides with the boundary of the background region, that part must be of the same boundary type as the background region. For more information, see Preparing Overset Regions.

The piston moves upward. The following screenshot shows the mesh after the piston moved upwards.



The upper part of the side boundary of the piston no longer coincides with the wall boundary of the background mesh. The dynamic overset behavior automatically detects this situation and changes the

previous wall boundary to an overset boundary. The lower part of the side boundary that still coincides with the wall boundary of the background region remains a wall boundary.

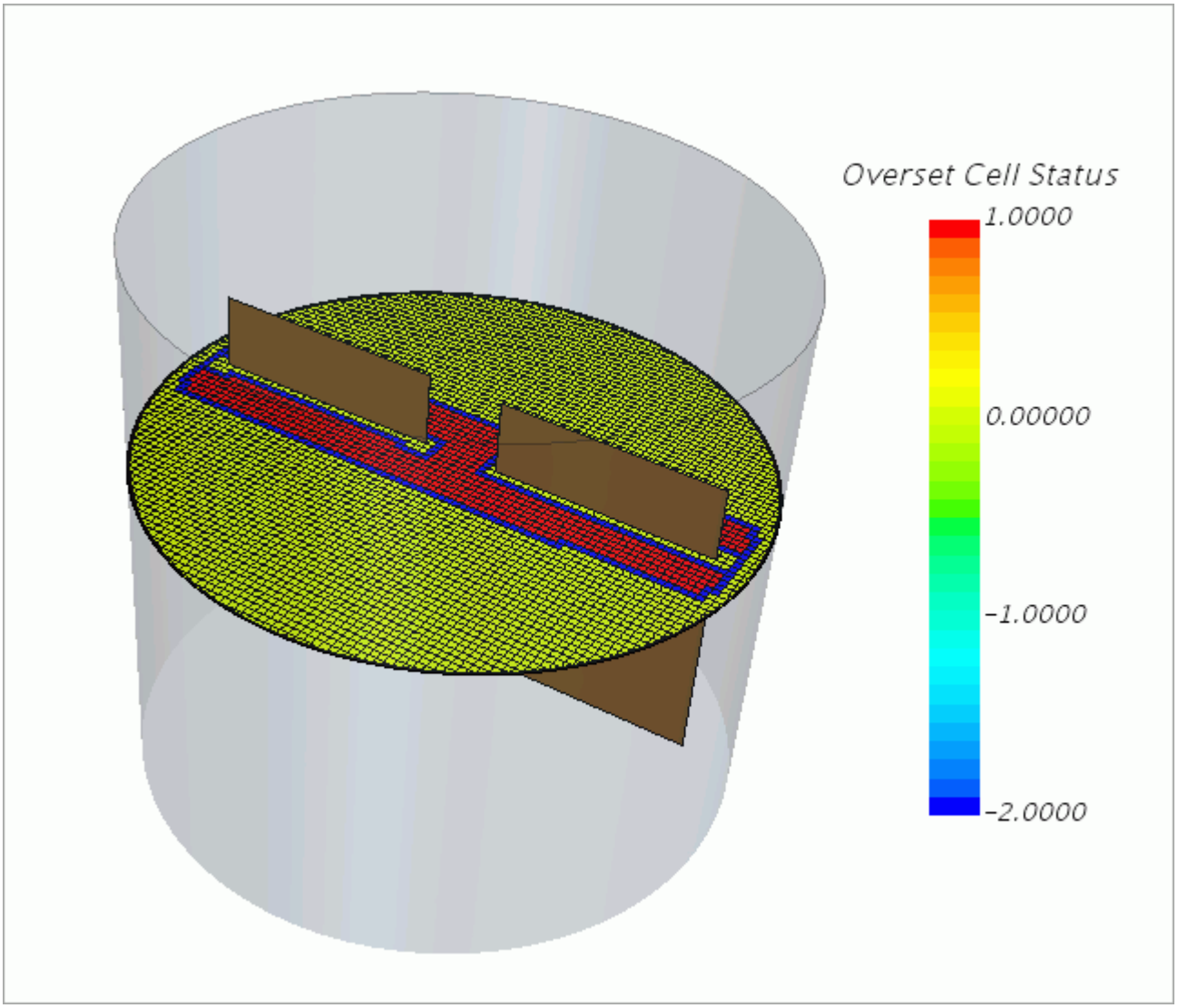
Artificial Thickening for Thin Structures

You are generally advised to generate meshes for the overset and the background regions with similar cell size distributions on both meshes. Therefore, when you simulate a thin geometrical structure within a background or overset region, for example a thin flap or a baffle, together with a comparatively larger geometrical structure in the corresponding overset or background region, you are required to resolve the large geometrical structure with a mesh of a similar cell size as the thin structure. This overset requirement can lead to high cell counts. If this requirement is not fulfilled, that is, there is a difference in mesh resolution of the background mesh and the overset mesh, the hole-cutting algorithm can face problems and lead to unexpected results.

When you apply the **Active** dynamic overset option to the wall boundary of the thin structure, this requirement is relieved and you can maintain a discrepancy in cell sizes on both meshes. You specify a number of protective layers around the thin structure to thicken it artificially. The number of layers should be such that the total thickness of the layers approximately matches the cell size of the coarse mesh of the other region.

The boundary remains a wall boundary and the cells that are adjacent to the boundary are always active.

The following screenshot shows an example:



You see two baffles within a large vessel that is colored with the overset cell status of the background mesh. The baffles are each modeled inside two overset regions and the vessel is the background region. For the wall boundaries of the baffles, the *Allowable Overset Faces States* is set to **Active**. This setting is required to ensure that the baffle is recognized as internal object with a wall boundary. One protective cell layer is specified. The overset cell status for the cells that are adjacent to the boundary is 0. This status corresponds to an active state of the cells in the background mesh

Contents:

Using Dynamic Overset Behavior
Preparing Overset Regions for Dynamic Overset Behavior

Using Dynamic Overset Behavior

To use dynamic overset behavior:

1. Prepare the overset regions for dynamic overset behavior.

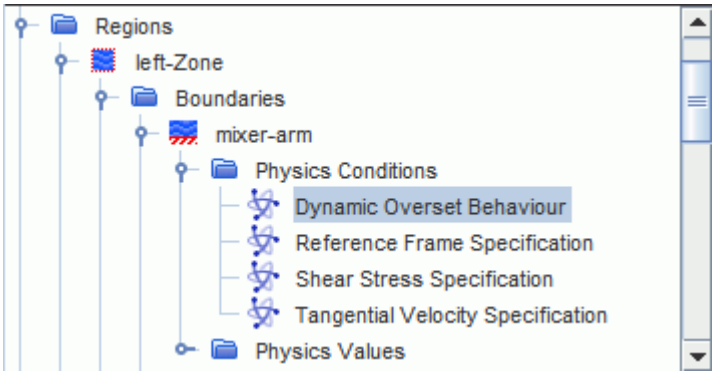
See Preparing Overset Regions for Dynamic Overset Behavior

2. Multi-select the overset regions or the overset region and the background region and create an overset interface between them.

   If you set up a simulation using a background region, the sequence of the multi-select operation is important. First, select the overset region. Then, select the background region. The overset region must always be Region-0, the background region must always be Region-1.

   See Coupling Regions

3. For each wall boundary that uses dynamic overset behavior:
   a) Select the **Regions** > **[Region]** > **Boundaries** > **[Boundary]** > **Physics Conditions** > **Dynamic Overset Behaviour** node and set *Allowable Overset Face States* to the appropriate Dynamic Overset Properties.



Preparing Overset Regions for Dynamic Overset Behavior

In contrast to Preparing Overset Regions, you do not prepare a background region and you do not assign an overset boundary type to the boundaries that use dynamic overset behavior. Instead, you apply the wall boundary type for those boundaries that use dynamic overset behavior.

To prepare the overset regions:

1. Select the mesh boundaries under the **Boundaries** node for each region for which you want to apply dynamic overset behavior.
2. Set *Type* to **Wall** for each of the boundaries.

Overset Topology

Overset topology describes how the regions for an overset mesh simulation are related to each other. The overset topology determines which region is the background region and which region is the overset region.

This information is crucial for the hole-cutting process. For many situations, the algorithm can detect this region relation automatically. However, for some applications, such as a piston within a cylinder configuration, a more detailed specification is necessary.

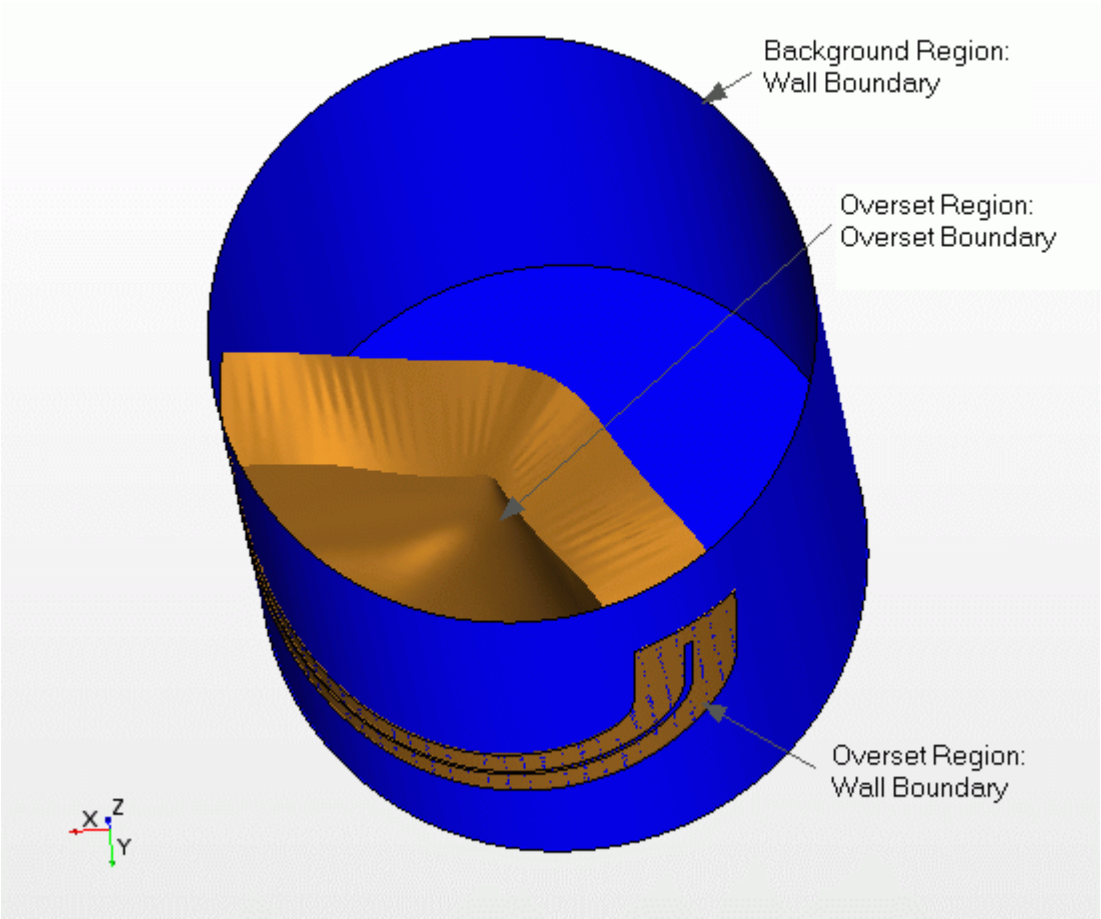
The following additional overset topology methods are available:

- Piston

  Select this option to simulate a piston within a cylinder configuration.
- Direction

  This is an expert option. Select this option only, if you wish to simulate a situation where the overset boundary does not completely enclose the overset region and this boundary is curved strongly at the same time. The remaining parts of the enclosing surface consist of wall or symmetry boundaries coinciding with respective boundaries of the background mesh.

  The following diagram shows an example of such a configuration:

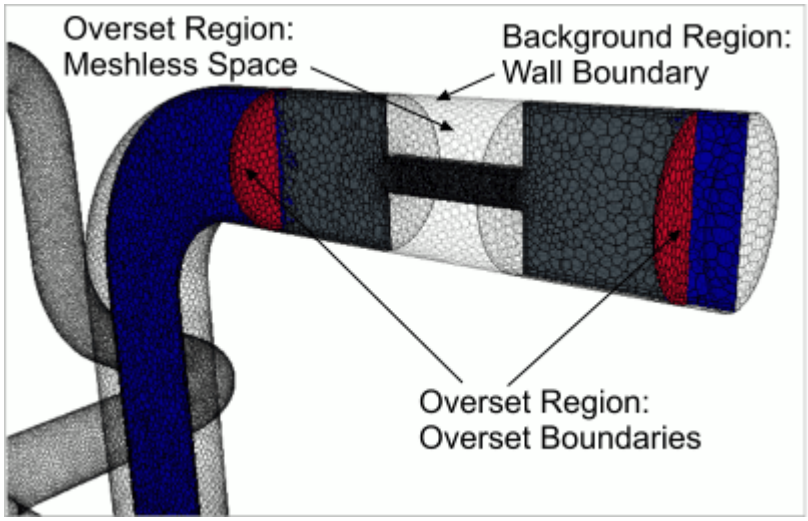


Here, the cylinder is the background region (blue) where the side boundary is a wall boundary. The overset region (bronze) consists of a curved inner overset boundary and a wall boundary coinciding with the wall boundary of the background mesh. The hole-cutting algorithm requires the direction in which the overset boundary appears to be closed. For this example, this direction is the z-direction.

- Bounded

  This is an expert option. Select this option only if you wish to simulate a situation where you have meshless space within your overset region that is bounded by a wall boundary of the background region.

  The following diagram shows an example (pigging case) of such a configuration:



The overset region (grey) slides inside the background region (blue) cylinder. The overset region is bounded by overset boundaries (red) and wall boundaries. The meshless space of the overset region is bounded by the wall boundary of the background region.

Contents:

Setting up Overset Topology

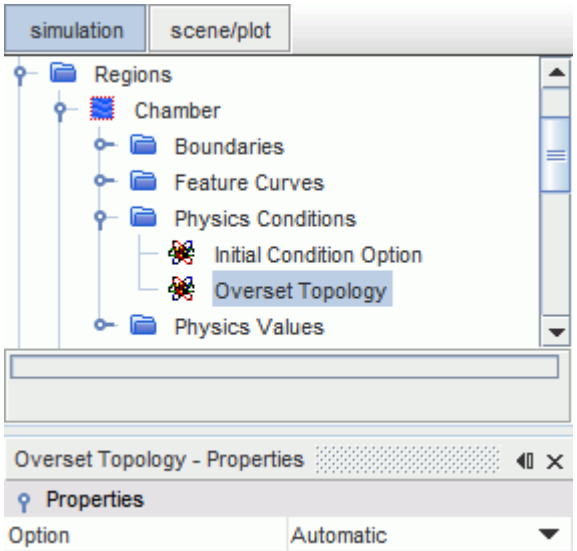
Setting up Overset Topology

Automatic overset topology is activated by default. This option is applicable to most situations. Only for specific situations, the topology algorithm fails and you must select the appropriate overset topology method.

To set up overset topology:

1. Select the **Regions** > **[Region]** > **Physics Conditions** > **Overset Topology** node and set *Option*to the desired overset topology method.

   See Overset Topology Properties.



2. If you set *Option* to **Piston**or **Direction**, select the **[Region]** > **Physics Values** > **Overset Direction Specification**node and select **Direction**.

   See Overset Direction Specification Properties

Setting up Overset Mesh with Fluid Film

For certain applications, it is useful to combine overset mesh with fluid film.

Examples of such applications are:

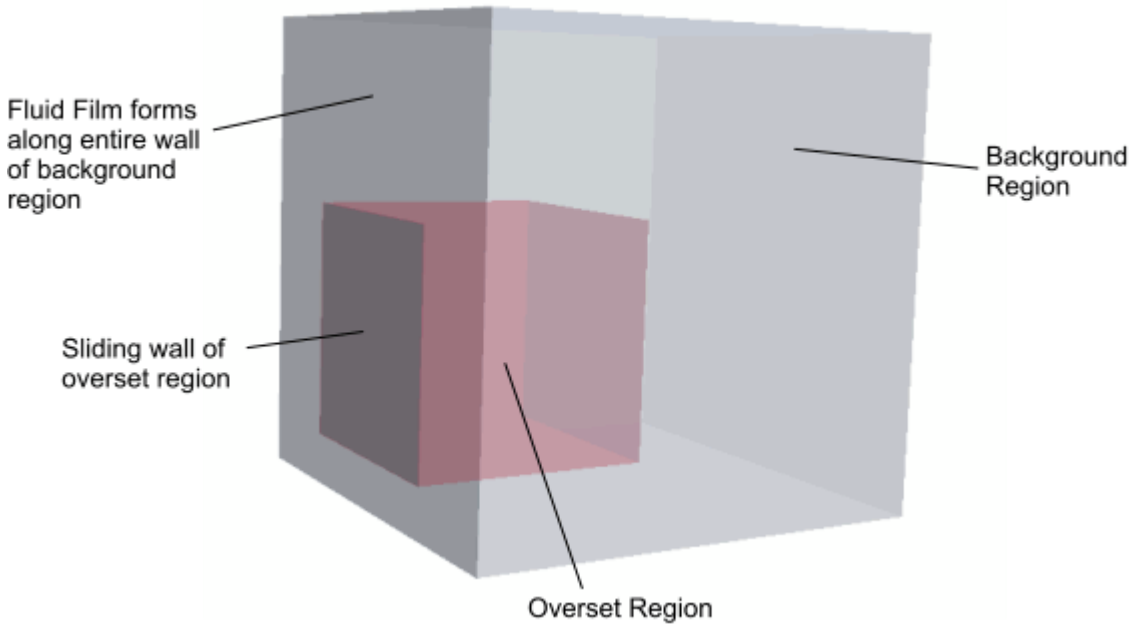
- automotive window wipers
- fuel flow on an engine cylinder with a moving piston

You can now model a fluid film shell region passing from one overset region to another or to a background region.

For the combination of overset mesh with liquid film, the following requirements apply:

- The overset model must be set up in a sliding configuration, that is, a boundary of the overset region must coincide with a boundary of the background region.
- You must create the shell regions on the sliding boundaries of the overset region and the background region.
- Each overset interface for shell regions must have a corresponding three-dimensional overset interface.

Before you begin to set up overset mesh with fluid film, generate a suitable mesh considering these requirements. In addition, the size of volume cells adjacent to the fluid film must be consistent for the background and the overset region. The following diagram shows an example:



The small cube is modeled as the overset region (red) and slides along the side wall of the large cube, which is modeled as the background region (grey). The fluid forms over the entire side wall of the background region, which coincides with the sliding wall of the overset region. You create two shell regions for the fluid

film: one on the sliding wall of the red cube and one on the side wall of the grey cube. By creating an overset interface between the two shell regions, the fluid film can pass from one shell region to the other.

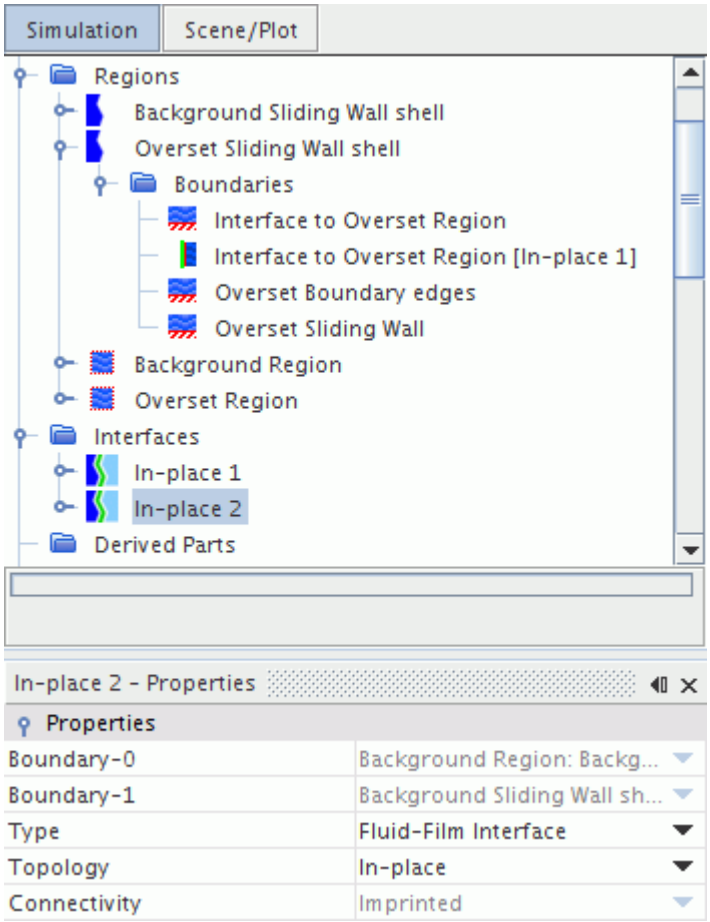
To set up overset mesh with fluid film:

1. Set up the fluid film model.

   See Setting Up a Fluid Film Simulation.

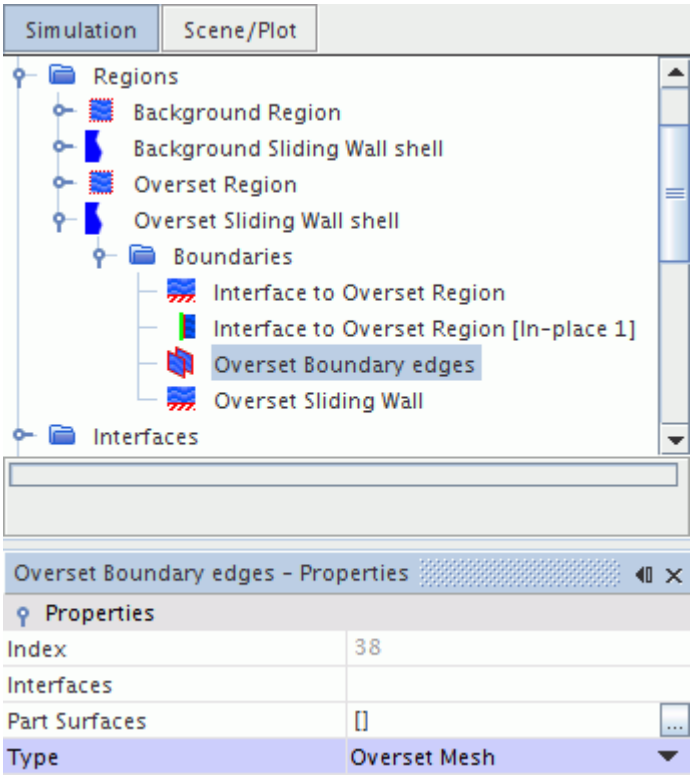
2. Set up the shell regions for the fluid film model. You must create one shell region on a boundary of the background region and one shell region on a boundary of the overset region.

   Fluid-Film interfaces and edges of the imprinted sliding overset boundary are created automatically as part of this process.



3. Set the boundary type of the imprinted sliding overset boundary edges to **Overset**:

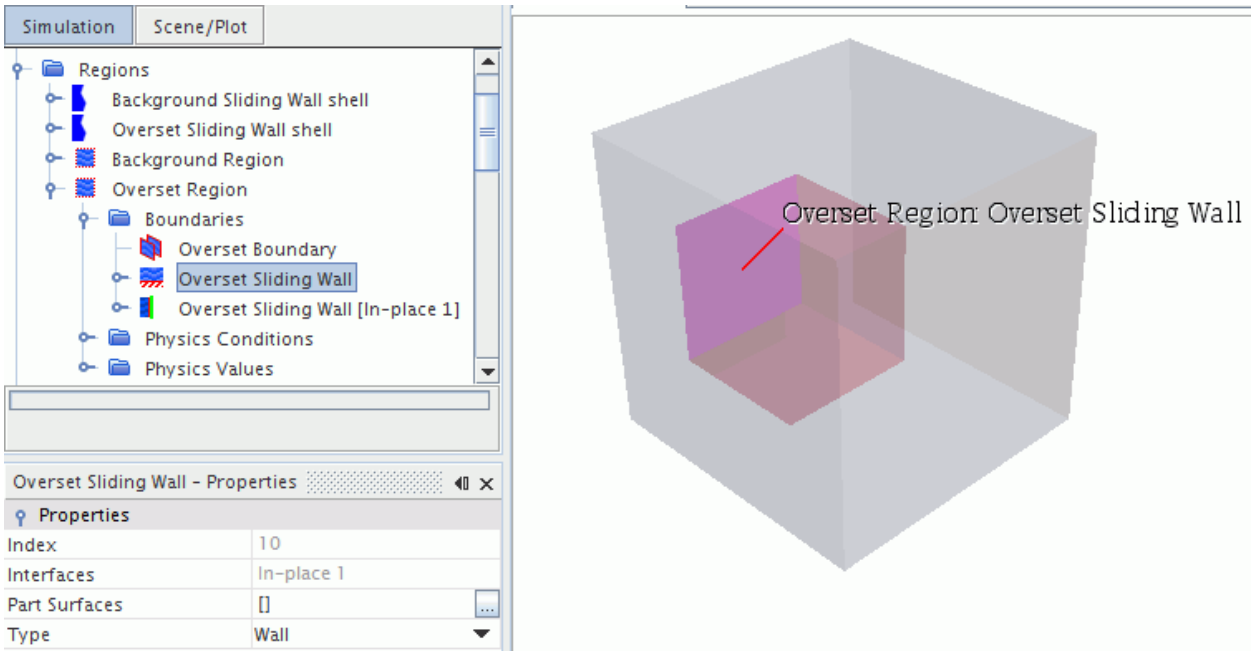
   a) Select the **Regions** > **[Overset Shell Region shell]** > **Boundaries** > **[Overset Boundary edges]** node and set *Type* to **Overset Mesh**.



4. Set up the overset mesh boundary on the overset region.

   See [Setting the Overset Mesh Boundary]

5. In the overset region, select the sliding boundary and set its *Type* to match that of the contacting boundary within the background region. Usually, this boundary is a wall boundary.



6. Select the **Regions** > **[Overset Shell Region shell]** > **Boundaries** > **[Overset Sliding Wall]** > **Physics Conditions** > **Reference Frame Specificiation** node and set *Option* to **Lab Frame**.
7. Create the overset mesh interface between the background and the overset regions.

   See Creating Overset Mesh Interfaces

Note: Currently, overset mesh with fluid film is not compatible with zero gap interface creation.

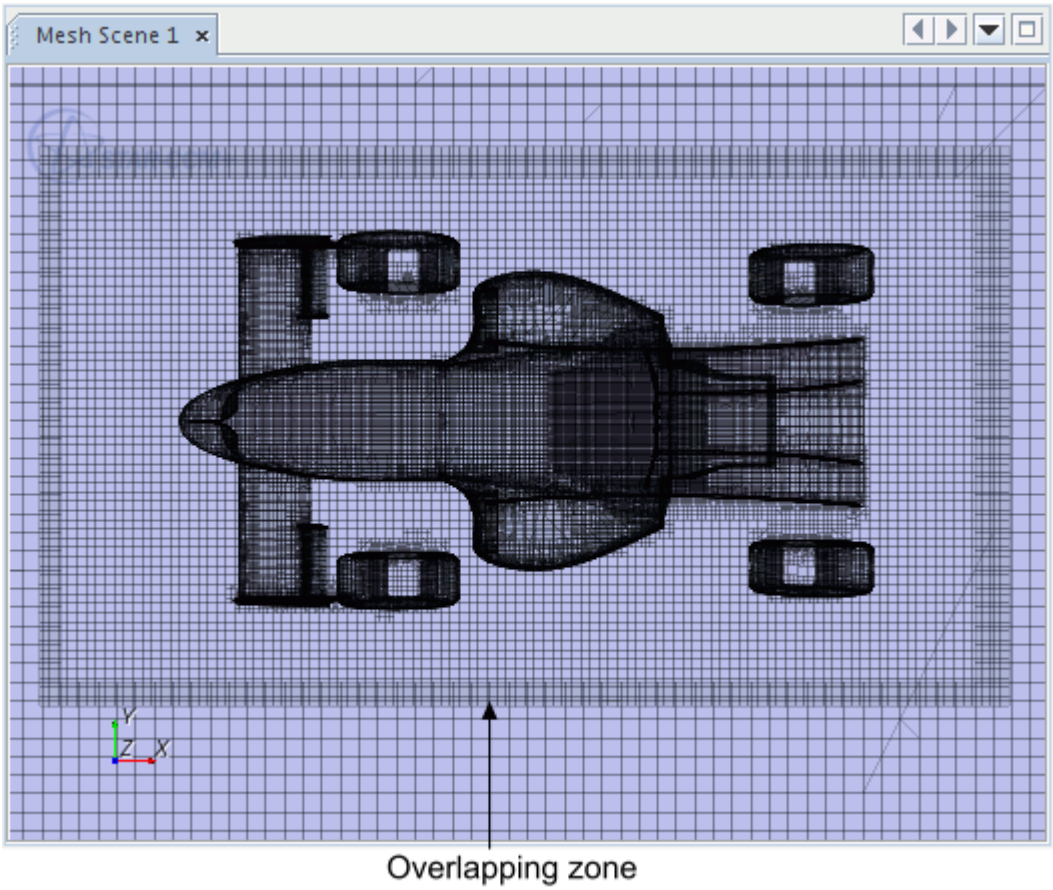
8. Create an overset interface between the fluid film shell regions of the background region and of the overset region.
   a) Multi-select the shell regions of the background region and of the overset region.
   b) Right-click on any of the highlighted items and select **Create Overset Mesh Interface**.

The fluid film can now pass from the shell region on the background region to the shell region on the overset region and vice versa.

Generating Suitable Meshes

A critical factor in successfully applying overset meshes is to have enough cells across the overlapping zone between background and overset regions. This overlapping zone must contain at least 4–5 cell layers in both background and overset meshes.

To minimize discretization error, ensure that the cell sizes on both meshes are similar within the overlapping zone. The coarser of the two coupled meshes determines the error level in the interpolation. An example is shown below.



A successful coupling by use of an overset interface results in a “hole” being cut in the background mesh (hole-cutting process). During this hole-cutting process, the status of the involved cells is determined, that is, it is determined whether the cells are active or inactive. There are field functions available that allow the visualization of the cell status (Overset Mesh Field Function Reference).

Using multiple overlapping overset meshes can reduce mesh cell count, as described in the following example:

Consider meshing a small gap, for example the gap between the rotor teeth of a gear pump, using multiple overset meshes overlapping each other. Due to the requirement that the overlapping zone must at least

contain four cell layers in both background and overset meshes, you need at least eight cell layers in the gap (four cell layers from each overlapping region). The Overset Zero Gap Distance Field Function can be used for determining the gap distance on a boundary; the minimum gap distance can be monitored and plotted cases where the gap size may neither be fixed in time or location (see Overset Mesh Field Function Reference).

If the two overlapping overset regions overlap such that the background mesh is completely covered, the background mesh does not have to fulfill the overlapping four cell layer requirement. Therefore, the cell size of the background mesh in the gap does not have to be as small as for the overset regions. However, you must ensure that the background mesh in this area is always covered completely by the overset regions. Then, you can use a coarser background mesh for those cells that are never activated. These cells are never visible when visualizing the active cells by using the Volume Mesh representation.

Special Criteria for Transient Problems

For transient problems with moving bodies, set the time step such that the movement of the overset region within one time step meets the following conditions:

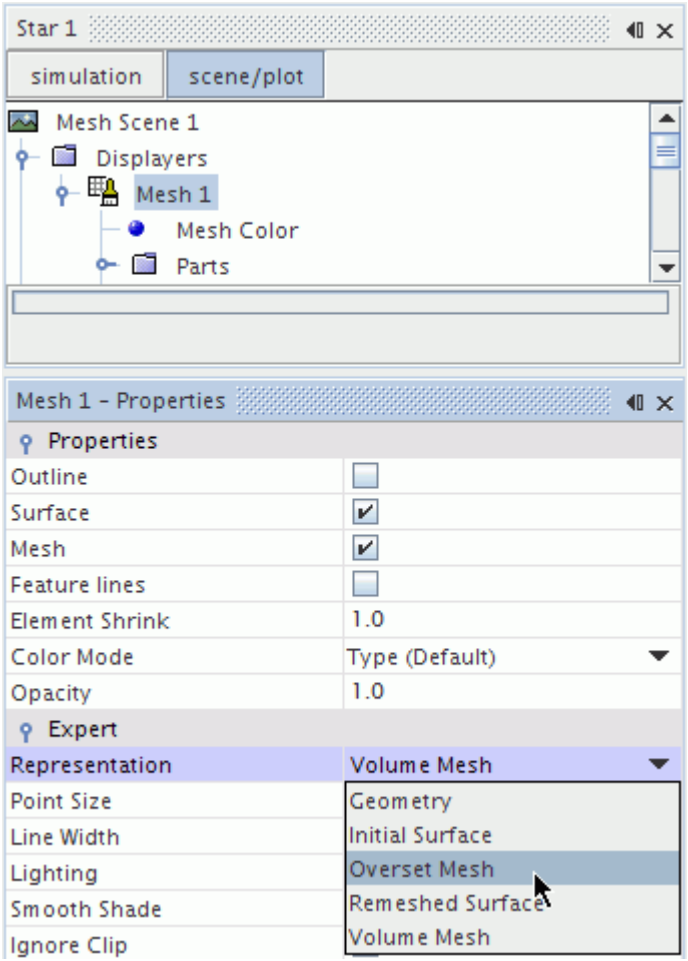
- When the time integration is set to the 1st-order Euler scheme, the maximum movement is the smallest cell in the overlapping zone.
- When the time integration is set to the 2nd-order implicit time integration scheme, the maximum movement is half the smallest cell size in the overlapping zone.

  If one inactive cell becomes an active cell without being an acceptor cell for at least one time step, there is no previous time step solution at that cell to use to compute the rate of change term. The 1st-order Euler scheme needs one previous time step value and the 2-order implicit scheme needs values from the previous two time steps.

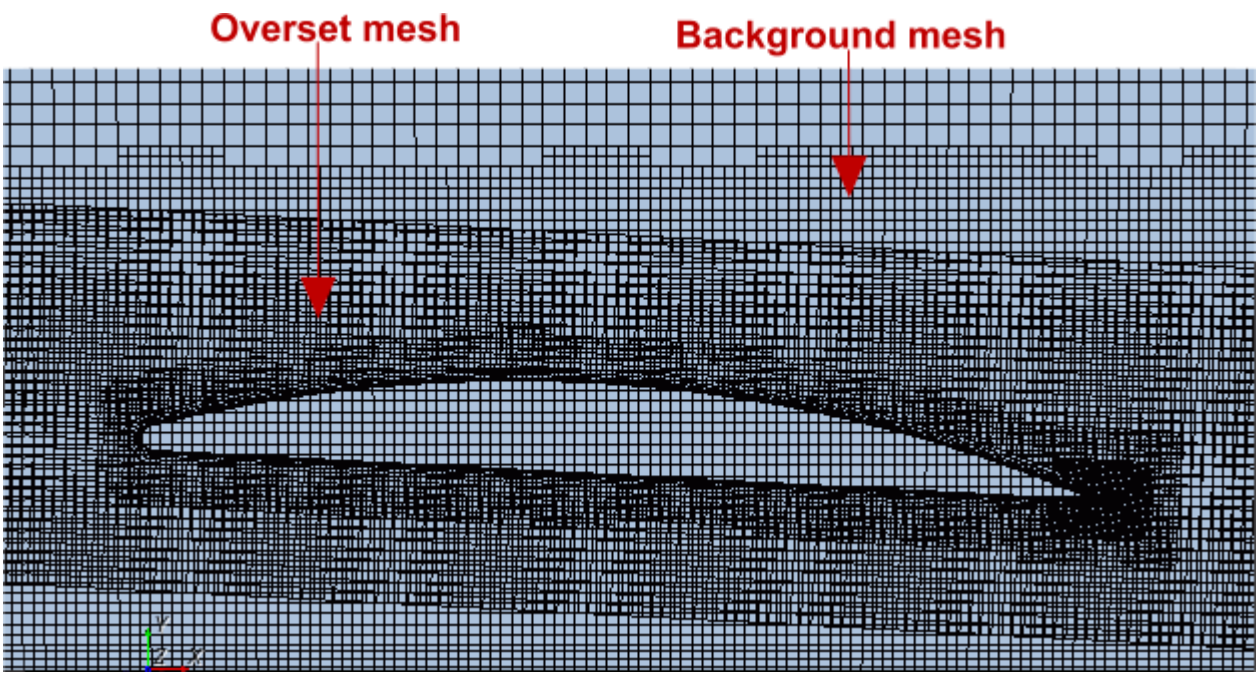
Visualizing and Analyzing Solution

Overset meshes are automatically created while running a simulation. Two mesh representations are available to visualize the mesh: volume mesh and overset mesh.

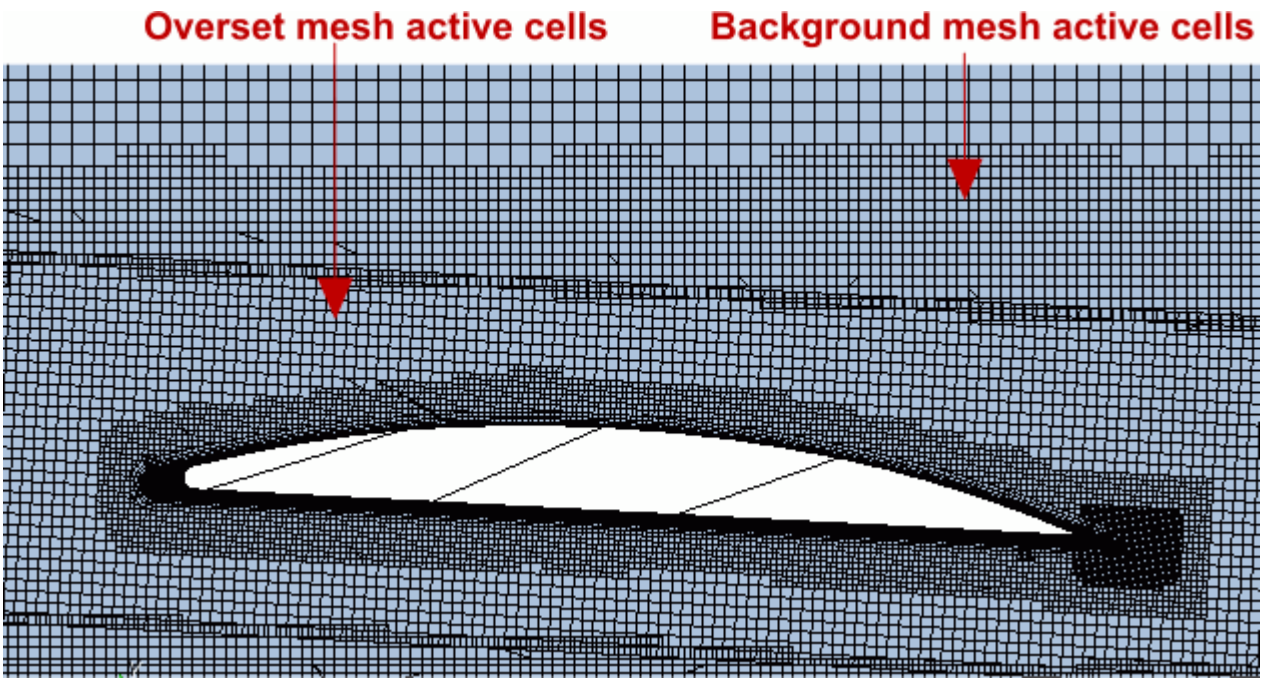
By default, the mesh scene is opened using the volume mesh representation. You can change the type in the *Properties* window of the Mesh node.



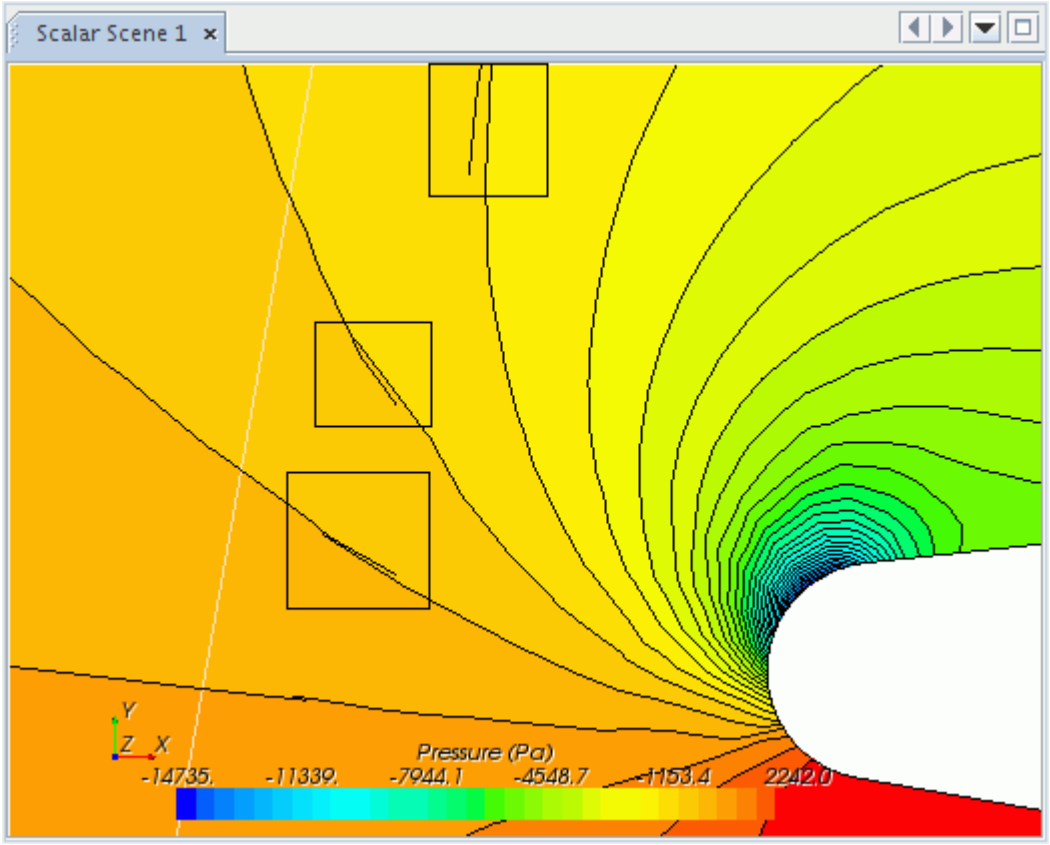
Below is an example of an Overset Mesh representation. This representation shows all background and overset region cells, whether they are active or inactive.



To see only the active cells, select the Volume Mesh representation. Visualization and reporting is done based on this representation.



You can visualize the solution on each mesh individually. The nodes of the background mesh most often do not coincide with the nodes of the overset mesh in the overlapping region. If the meshes are not fine enough, this discontinuity can create some visualization issues. One such issue is with contour lines. The discontinuity in the contour lines that are shown below is the result of discretization error and reduces as the mesh is refined.



The overlapping of active cells of the background and the overset region (overlapping zone) also affects the volume integrals. For the volume integrals, the volumes of all active cells are integrated, that is, the overlapping volumes of both the cells in the background and in the overset region are integrated. This way, the volume integrals introduce some inaccuracies for the whole solution domain because of the duplicate integral evaluation of the active cells in the overlapping zone.

This issue does not apply to surface integrals when walls or physical boundaries are either in a background or an overset region. However, when you have boundaries from the background mesh and the overset mesh overlapping each other, this results in overlapping active faces. In this case, you can get inaccuracies for the surface integrals because all active faces are integrated. There is also a duplicate integral evaluation in the overlapping zone of the boundaries.

Activating Load Balancing for Parallel Overset Mesh Simulations

When running an overset mesh simulation in parallel on multiple cores, the traditional mesh-based domain decomposition can lead to imbalanced CPU loads across the parallel cores. This CPU load imbalance is due to the overset assembly process, which poses a substantial computational cost on the processors where the acceptor cells reside. Overset mesh load balancing rebalances the domain decomposition such that the acceptor cells are distributed most evenly across the parallel cores.

To perform overset mesh load balancing, there are two solvers that are involved: the partitioning solver and the overset load balancing solver. These two solvers interact during the simulation. The overset load balancing solver collects the information that is required to rebalance the domain and passes it to the partitioning solver. In order to reduce the load imbalance, the acceptor cells are weighted more than other cells to account for their computational cost. The partitioning solver redistributes the cell load on each core.

To activate load balancing for overset mesh:

1. Select the **Solvers** > **Load Balancing** node and set the Load Balancing Properties.
2. Select the **Solvers** > **Partitioning** node and set the Partitioning Properties.

See Also:

Load Balancing

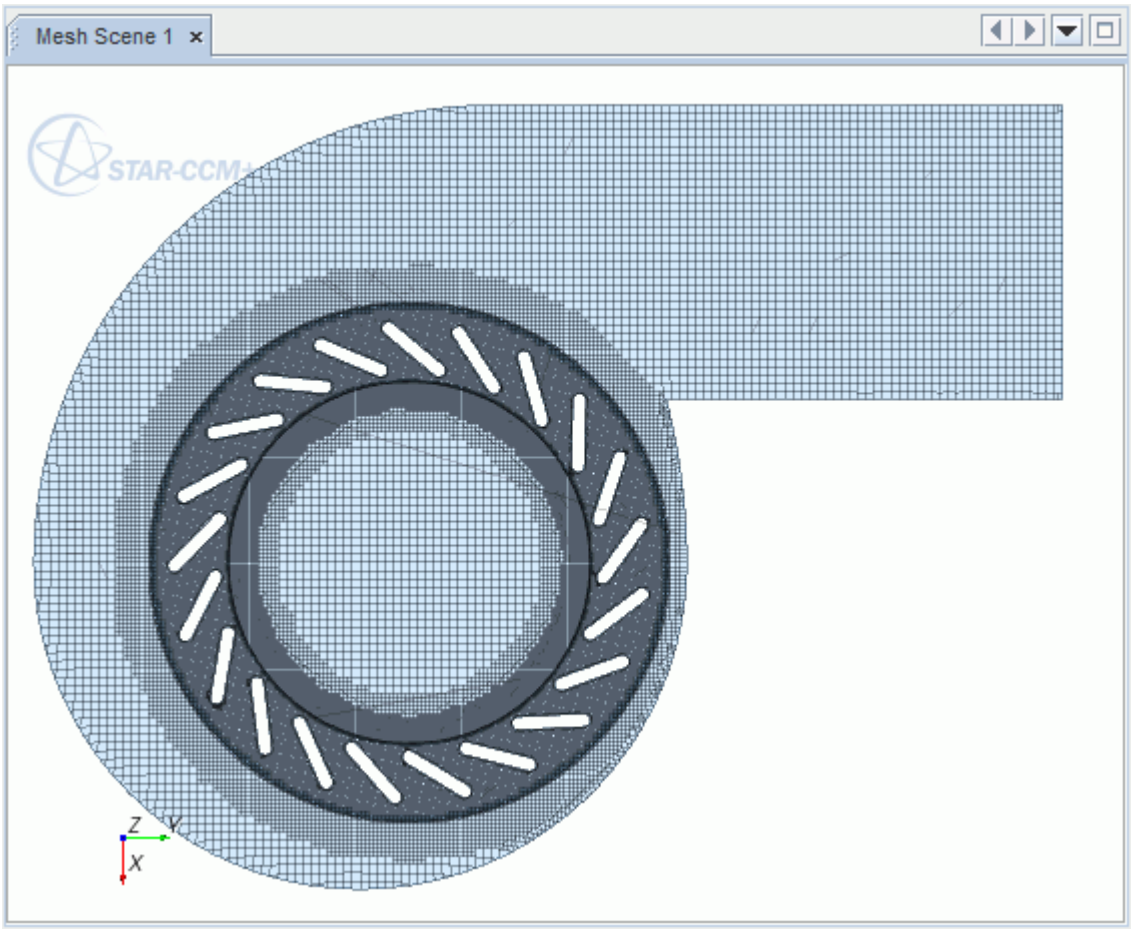
Overset Mesh Guidelines

This section outlines some guidelines for setting up an appropriate mesh in the overlapping region of an overset mesh.

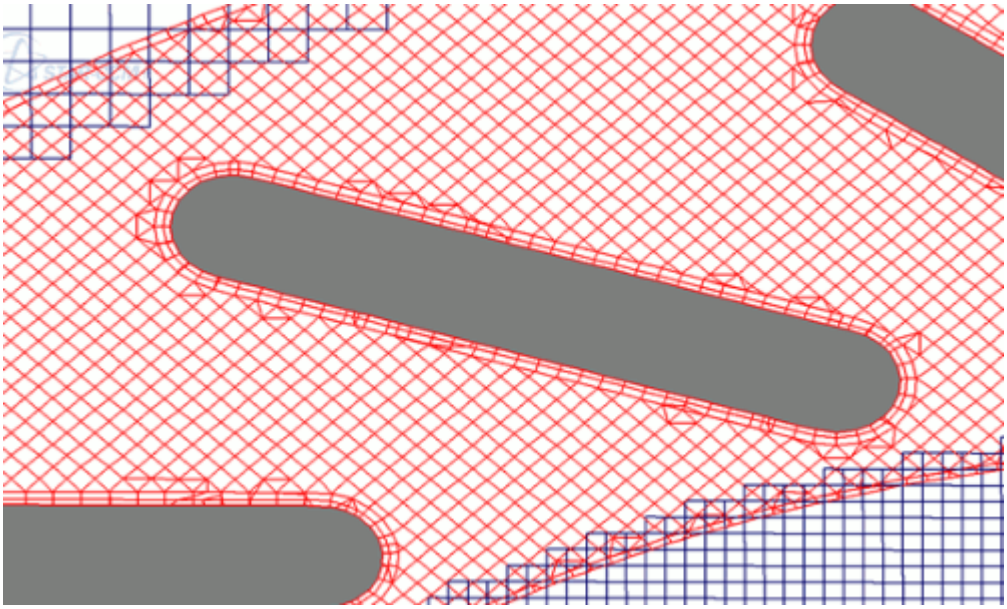
The required features of overset regions and meshes are:

- Between the wall and the overset boundary of an overset region, there must be at least 4–5 cells in both the overset and the background mesh.

To demonstrate this point, consider a cross section of a simplified blower model that is shown below:

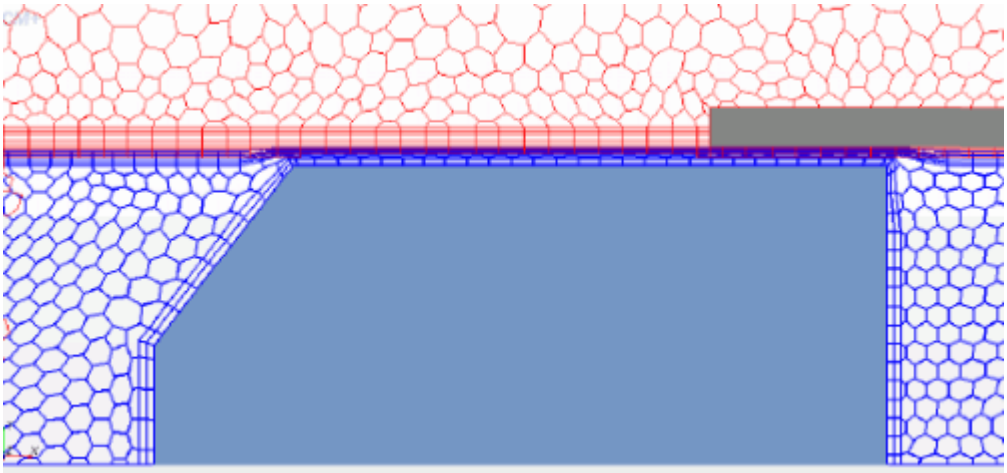


The overset region includes the set of rotor blades. It extends to the top and bottom walls of the blower in the axial direction, but two cylindrical surfaces limit the region in the radial direction. The space between the leading edge of the blades and the outer cylinder is about twice as large as the space between the trailing edge and the inner cylinder. Based on the given requirement, you need a background mesh that is twice as fine in the inner overlapping zone as in the outer overlapping zone. The cell sizes can be clearly seen when you zoom in on one of the blades. The background mesh is shown in blue while the overset mesh is in red. Only the active cells are shown.



- To couple the overset and the background meshes, there must be at least 4–5 cells in each mesh in the overlap region.

This requirement is illustrated below with a body moving through a channel with a small clearance at the top. The image below is zoomed in on the body. The background mesh is shown in red while the overset mesh is in blue.



In this example, the overset region is kept completely within the background region. Therefore, the overset region covers only a small distance above the body, limited by the width of the gap between the body and the channel wall. To account for this limitation, the background mesh has a prism layer with five cells in it along the channel wall. The thickness of the prism is made equal to the width of the gap between the body and the channel wall.

In the overset mesh for this example, there is a prism layer along the overset boundary containing four cells, and a prism layer along the body wall containing two cells. As both prism layers are present in the gap region, there are enough overlapping cells to ensure successful coupling of the two meshes as the body moves along the channel.

The minimum number of 4–5 cells that are required to couple the overset and background regions refers to cell count in the direction normal to the boundary. Depending on the cell aspect ratio and the orientation of cells on the two meshes relative to each other, this requirement does not apply. This criterion is also not applicable if an overset region is not fully embedded in the background region, that is, the overset and background regions share a boundary. In the overset region for such a case, the boundary must be of the same type as in the background region, and not an Overset Mesh boundary.

If the mesh does not fulfill all of the required features, the initialization of the overset region interface fails. It is not always obvious where the problem arises. For a moving overset region where a new Overset Mesh Interface is created at every time-step, it could be that one or more requirements are violated at only one particular time. The design of the overset and background meshes thus requires a careful assessment of all possible situations during the simulation.

Refer to the troubleshooting section for how to visualize the overlapping cell status.

In addition to the above requirements, the following mesh features are desirable for accuracy reasons:

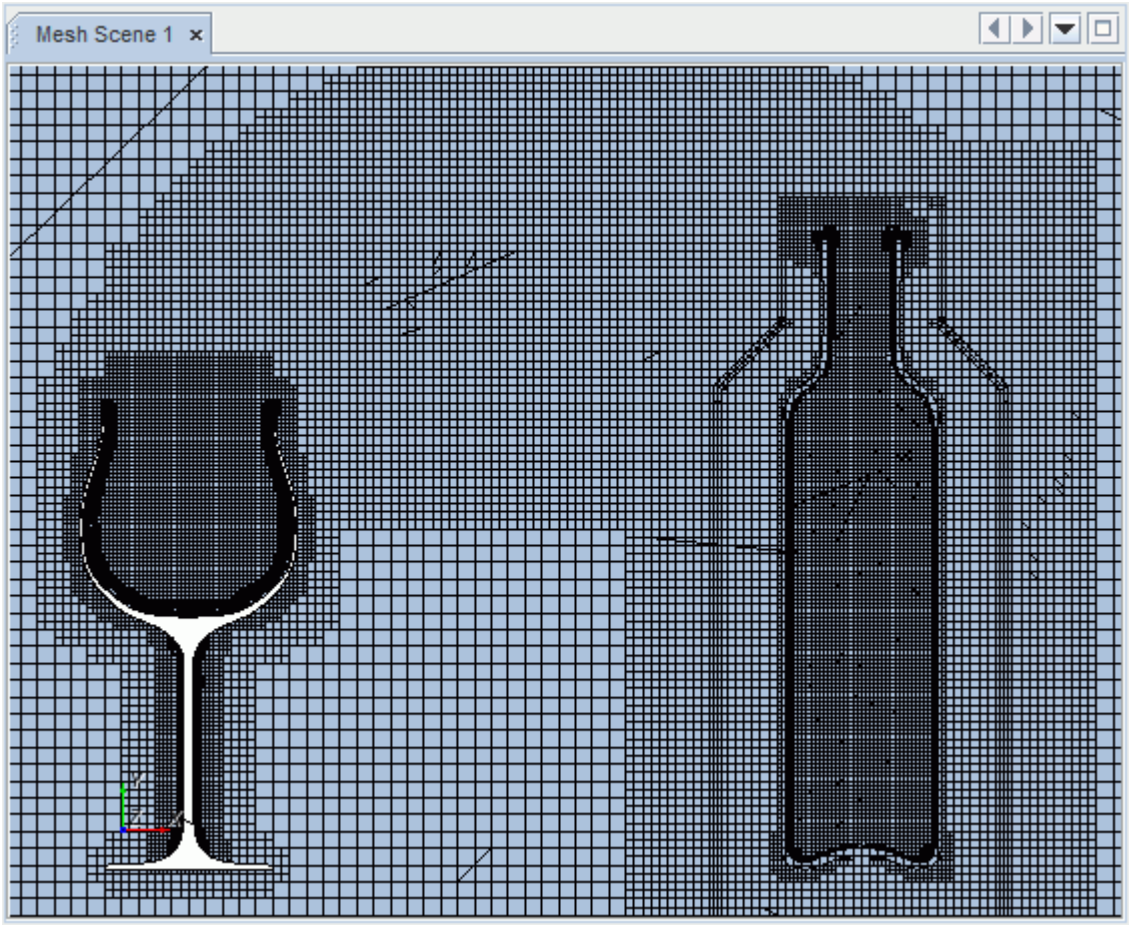
- The cells in the overlapping region are of similar size on both the background and overset meshes.

  Interpolation errors are introduced by expressing field values at acceptor cells in one mesh using field values at donor cells in the other mesh. If cells in both sides of the overlapping region are of similar size, these errors are reduced to the same order of magnitude as other discretization errors.

  For example, the same interpolation method is used in overset meshes as in computing field values at cell faces when convective fluxes are approximated. If the cells are of the same size, shape and orientation, then the distance between the acceptor cell centroid and the nearest donor cell centroid is never larger than half the cell width. This distance is the distance between cell face center and the cell centroid. If the cells on one mesh are substantially larger, interpolation errors when using donor cells from this mesh are larger than in the other direction. This error is not important when the variation of field values is small. However, this error significantly affects the solution when the overlapping zone is part of a region where there is a significant variation in the solution.
- Select the extent of the overset region such that variables do not vary much in the overlapping zone.

  If you cannot minimize this variation in the overlapping zone, refine the mesh to satisfy accuracy requirements.

Mesh adaptation and interpolation accuracy is demonstrated in the following example in which wine is poured from a bottle into a glass. The bottle is initially vertical and positioned to the right of the glass. The bottle is placed back into this position after the pouring process.



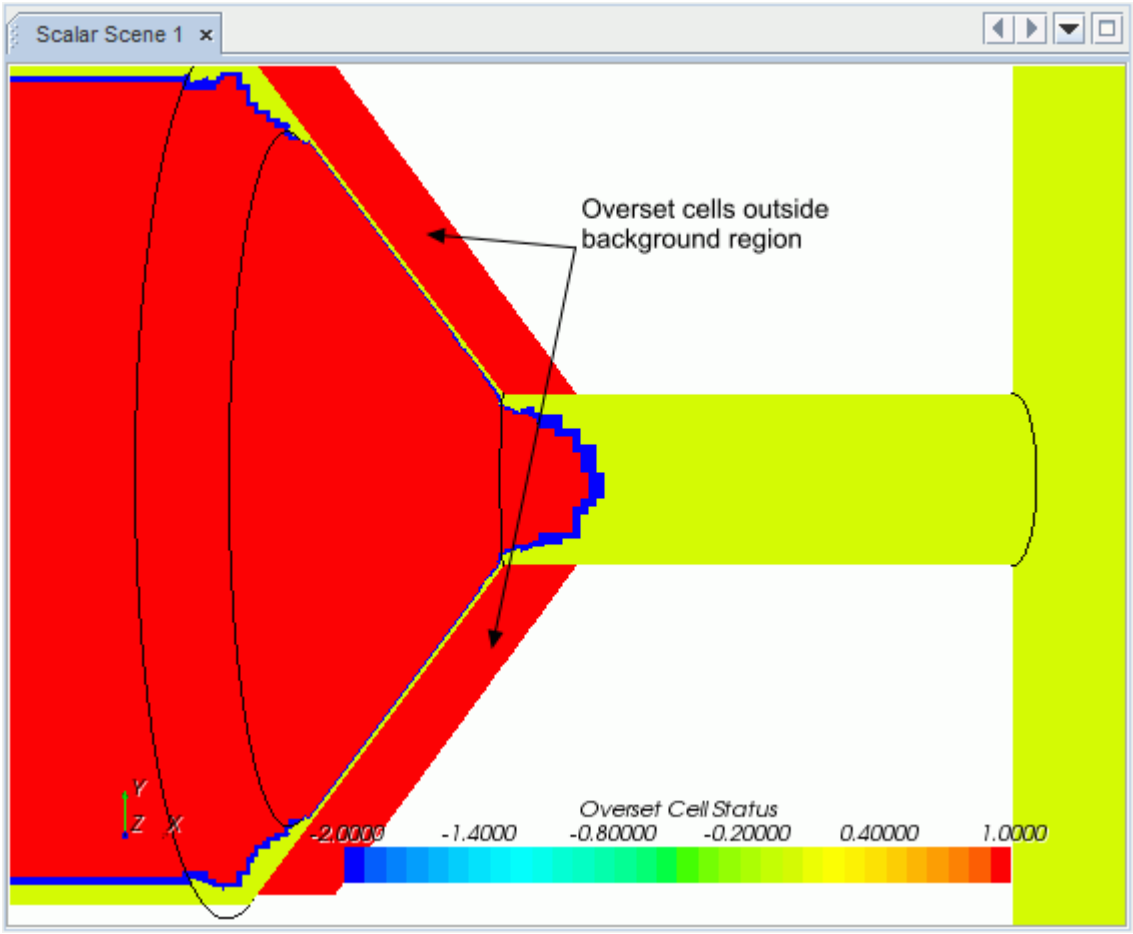
The cell size in the background mesh is adapted to the size of the mesh around the bottle in the overset region within the zone of its motion. The background mesh is two times coarser than the overset mesh around the bottle opening while it is away from the glass and no wine is being poured. At this stage, the velocities are small in this region with insignificant interpolation errors. Once the wine is poured and the bottle is brought to a nearly horizontal position, the overset mesh overlaps the refined background mesh above the glass. The interpolation accuracy matches that of other discretization approximations.

Overlapping the Background Region

To simplify cases that deal with small gaps between walls, you can set the overset region to extend beyond the solution domain. Any mesh cells of an overset region that lie outside of the background region are inactive. These cells can only become active again if they enter the background region, which always represents the solution domain. If the overset region moves outside the solution domain, new acceptor cells are found that are inside the background region such that the coupling criteria are satisfied.

Of the overset region, only boundaries of the type Overset Mesh or Symmetry Plane can protrude the background mesh. Wall boundaries are not allowed outside of the background mesh. However, if you use overset zero gap interfaces, this restriction is no longer valid. For overset zero gap interfaces, wall boundaries are allowed outside of the background mesh.

You can see this condition in the following screenshot that shows motion using an injector.



The overset region in this case is over the needle, which has part of its geometry outside of the nozzle. Those cells are inactive and are in red. The acceptor cells are updated to represent the cells that are inside the nozzle. These cells are shown in blue.

Refer to the troubleshooting section for further information on analyzing cell types.

Which Hole-Cutting Approach Should You Use?

When you create an overset interface, you can first use the default layered approach. This approach is computationally more efficient than the global approach and suitable for most cases. If the overset interface creation gives overset intersector errors, you can activate the *Alternate Hole Cutting* to use the global hole-cutting approach (see Overset Mesh Interface Properties). The global approach is more robust when you run the simulation on many cores. The global approach is more sensitive to non-standard overset topologies where the overset boundary does not form a closed surface (see Overset Topology).

Overset and Lagrangian Multiphase Simulations

When using overset meshes in a Lagrangian multiphase simulation with two-way coupling, you are advised to activate the following overset mesh interface properties:

- *Alternate Hole Cutting*
- *Close Proximity*

When you apply two-phase coupling in a Lagrangian simulation, the particles influence the flow field. This influence depends on the number of particles that are present. In the area where background mesh and overset mesh overlap—the overlapping zone—the cells from each mesh cover the same space. The influence of a particle that enters the overlapping zone is distributed between the cells from the background mesh and the overset mesh in the same location. Therefore, the influence of that particle on the flow field in the overlapping zone is lower than in the non-overlapping portion of the mesh. Activating *Alternate Hole Cutting* and *Close Proximity* reduces this effect and also improves the calculation of particle volume fraction.

Overset Setup Examples and Best Practices

You can use overset mesh when modeling complex motion setups. Overset mesh is also compatible with many physics scenarios, for example, multiphase flow or DFBI. Some of these combinations with other physics models in STAR-CCM+ require you to make specific settings or to observe certain requirements in addition to the general overset workflow.

The following sections present examples and setup descriptions for frequently used overset configurations.

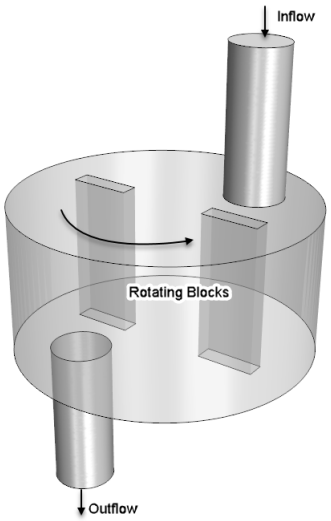
Combination of Overset Interfaces with In-Place Interfaces

Generally, you can use overset interfaces and in-place interfaces, (such as internal or contact interfaces), in one simulation. Both boundary-based and parts-based interfaces are valid. Specifically, you can define the overset region such that its enclosing boundaries—either overset or sliding wall—overlap the other interfaces.

With this compatibility, you can create overset configurations such as in the following example.

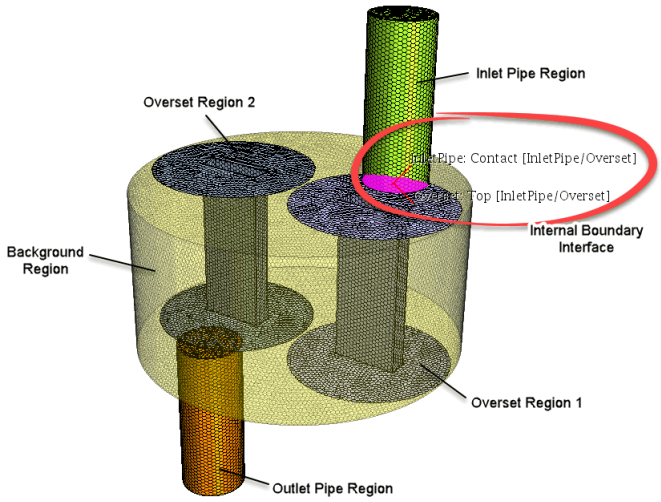
Physical Problem Description

Two blocks rotate within a cylinder. The top and the bottom sides of the blocks slide along the top and the bottom walls of the cylinder, respectively. As the blocks rotate, they also slide across the openings for the inflow and outflow pipes.



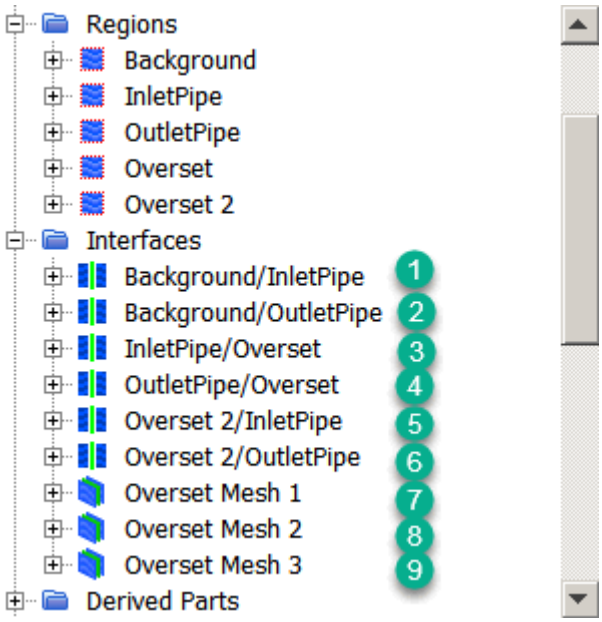
Overset Simulation Setup

As shown in the following diagram, the two blocks are enclosed by two cylindrical overset regions (grey)—Overset Region 1 and Overset Region 2. The cylinder (yellow) constitutes the background region. The inflow (green) and outflow (gold) pipes are modeled as two separate regions—Inlet Pipe and Outlet Pipe. These regions are connected to the background region and the overset region through boundary interfaces of type internal.



Regardless of whether an overset region or the background region passes the internal interfaces, the overset assembly process automatically determines active and inactive cells. The boundary interface intersector uses only active boundary faces of parent boundaries for the intersection. See Interfaces Overview and Intersecting an Interface.

In this example setup, overset mesh interfaces and internal in-place interfaces (boundary-based or parts-based) are present as shown in the following screenshot:



The interfaces are defined as:

1. Internal in-place interface between the background region and the inlet pipe region
2. Internal in-place interface between the background region and the outlet pipe region
3. Internal in-place interface between the inlet pipe region and overset region 1
4. Internal in-place interface between the outlet pipe region and overset region 1

5. Internal in-place interface between the inlet pipe region and overset region 2
6. Internal in-place interface between the outlet pipe region and overset region 2
7. Overset mesh interface between background region and overset region 1
8. Overset mesh interface between background region and overset region 2
9. Overset mesh interface between overset region 1 and overset region 2

Overset Mesh Methodology

To establish the connectivity between the background and the overset regions, a two-step Overset Assembly process takes place:

1. Hole-cutting, which determines which cells are active, inactive, or acceptor cells.
2. Donor Search, which ensures that donor cells are found for each acceptor cell.

Contents:

Hole-cutting process
Donor Search
Mass Conservation

Hole-cutting process

The hole-cutting process is part of the coupling of the overset region with the background region through an overset interface. A successful coupling by use of an overset interface results in a “hole” being cut in the background mesh - the hole-cutting process. The hole-cutting process determines whether cells are active or inactive in the coupled simulation. Currently, STAR-CCM+ provides two methods for hole-cutting:

- Layered approach
- Global approach

Layered Approach

Along the boundary of the overset region, a layer of cells is identified that forms the donor cell layer. The cells in the background region in immediate vicinity to the donor cell layer become the acceptor cells. These layers of cells must form a watertight container around the overset region. The cells of the background mesh that are completely covered by the overset region cells inside this closed container become inactive.

Global Approach

For each cell of the background region, the algorithm verifies whether the cell centroid of that cell is outside or inside the overset region. If the cell centroid of a background region cell is inside the overset region, it becomes inactive.

Donor Search

For each acceptor cell, donor cells must be found. The set of donor cells depends on the interpolation option that is chosen and on the number of active cells in the donor region around the acceptor cell centroid.

The interpolation options are:

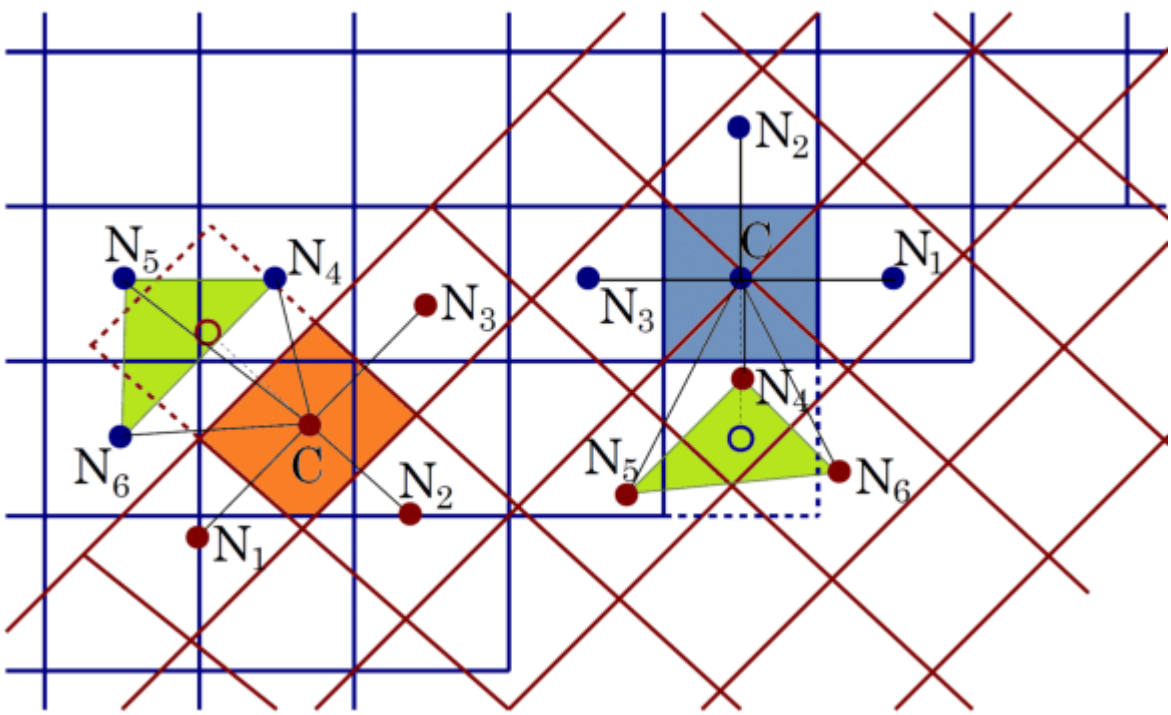
- Distance-weighted interpolation, where the interpolation factors are inversely proportional to the distance from acceptor to donor cell center, resulting in the closest cell giving the largest contribution. This involves 3 donor cells (in 2D) or 4 (in 3D). The complete overset assembly process is performed at each time-step.
- Linear interpolation using shape functions spanning a triangle (in 2D) or a tetrahedron (in 3D) defined by centroids of the donor cells.

  This option is more accurate but also more expensive. It is important in simulations involving moving meshes as it ensures that interpolation elements do not overlap. The choice of donor cells is not unique since the available donor cell centroids that enclose the acceptor cell centroid can define more than one triangle or tetrahedron. This way the interpolation is continuous as the acceptor cell centroid passes from one interpolation element to the next.

  For moving mesh simulations, when the movements of the overset region compared to the cell size during one time-step are small, the donor triangle (in 2D) or the donor tetrahedron (in 3D) can still be valid. If this is the case, only the interpolation factors are re-calculated instead of executing the entire overset assembly process. For such situations, the linear interpolation provides an optimization regarding runtime. For overset motion setups involving DFBI, the probability of STAR-CCM+ applying this optimization is high, because of typically small motions being performed during each inner DFBI iteration.
- Least squares interpolation.

  For a description of the least squares interpolation method, see Least Squares Interpolation.

  The interpolation function is built directly into the coefficient matrix of the algebraic equation system. This approach ensures implicit coupling of the overset meshes. In the image below, two acceptor cells are shown using dashed lines, one in the background mesh and one in the overset mesh.



The fluxes through the cell face between the last active cell and the acceptor cell are approximated in the same way as between two active cells. However, whenever the variable value at the acceptor cell

centroid (marked by the open symbols in the above figure) is referenced, the weighted variable values at the donor cells are substituted:

$$\Phi_{\text{acceptor}} = \sum \alpha_i \Phi_i \tag{96}$$

In this equation, α_i are the interpolation weighting factors, Φ_i are the values of the dependent variable Φ at donor cells N_i and subscript i runs over all donor nodes of an interpolation element (denoted by the green triangles in the figure). This way, the algebraic equation for the cell "C" in the above figure involves three neighbor cells from the same mesh (N_1 to N_3) and three cells from the overlapping mesh (N_4 to N_6). The coefficient matrix of each equation solved (both for the segregated and coupled solution method) is updated accordingly to ensure that equations can be solved up to the round-off level of residuals.

Mass Conservation

The interpolation of dependent variables at the overset interface is not conservative. Although this lack of conservation is usually acceptable, for the segregated flow model, the pressure-correction equation requires special attention. Strict conservation of mass is essential for the solution of this equation. For overset mesh computations, additional treatment is necessary to enforce the global mass conservation.

In the pressure-correction equation, the source term for a cell represents the sum of mass fluxes across all cell faces of that cell. The overset boundary is not the outer boundary of the overset region, but consists of cell faces that separate the interpolation cells from the discretization cells. It is similarly defined for the overlapping and the background regions. The interface fluxes contribute only to one cell. To ensure that mass is conserved, the sum of mass fluxes must be zero, that is, the total mass flux across all interfaces must be equal to zero. However, because of the non-conservative nature of the inter-grid interpolation, this condition is not automatically fulfilled and a mass imbalance results:

$$\Delta\dot{m} = \sum_i \dot{m}_i \tag{97}$$

where $\dot{m}_i$ is the mass flux through the interface cell face i. This mass imbalance appears as a net mass source in the pressure-correction equation and must be eliminated. There are two methods available [12].

Source Term Correction

Mass conservation can be enforced by adding a fraction of the mass source due to the non-conservative overset interpolation given by Eqn. (97) to the right-hand side of the discretized pressure-correction equation Eqn. (1641). This addition is such that sum of the new right-hand side is zero. The source term of the pressure-correction equation then is:

$$r^{corr} = r + \beta_s \Delta\dot{m} \tag{98}$$

where β_s is the fraction of mass source that is added to each cell. For each cell, the factor β_s is computed proportional to cell volume.

Flux Correction

Mass conservation can be enforced by adjusting the interface fluxes such that their sum Eqn. (97) is zero. The adjustment of interface fluxes is achieved by subtracting a part of mass source $\Delta\dot{m}$ from each mass flux $\dot{m}_i$:

$$\dot{m}^{corr} = \dot{m}_i - \beta_i \Delta\dot{m} \tag{99}$$

where the factor β_i is proportional to the share of the mass flux $\dot{m}_i$ in the total mass flux through the overset interface $\dot{m} = \left|\sum_i \dot{m}_i\right|$. The proportionality factor is given by:

$$\beta_i = \frac{\left|\dot{m}_i\right|}{\dot{m}} \tag{100}$$

The mass flux correction defined in Eqn. (99) ensures that the sum in Eqn. (97) is zero, because there is no contribution from the overset interfaces. The continuity equation is satisfied for each cell and a divergence-free flow field is obtained.

Overset Mesh Reference

You control the overset mesh simulation through various settings and properties that STAR-CCM+ applies at boundaries, regions, and interfaces.

Boundary Settings

Wall

Dynamic Overset Behaviour

Selects different types of dynamic overset behavior. Apply the dynamic overset behavior options depending on the overset configuration in the model.

Allowable Overset Face States	Corresponding Physics Value Nodes
Active, Inactive Dynamic overset behavior is deactivated. The default setting. The overset hole cutting algorithm determines automatically whether the cells in the background mesh are active or inactive.	None
Active, Acceptor Activates dynamic overset behavior. Depending on position, the cells that are adjacent to the wall boundary are either acceptor cells or active cells. If they are acceptor cells, the wall boundary performs as an overset boundary. If they are active cells, the wall boundary remains unchanged. For more information, see Dynamic Overset Behavior.	None
Active Activates dynamic overset behavior where the cells adjacent to the wall boundary are always active. Thus, the boundary remains a wall boundary. This option is applicable when you have a thin geometrical structure, for example, a baffle in the background or the overset region and a comparatively large geometrical structure in the corresponding overset or background region. Selecting this option prevents problems with the hole-cutting algorithm due to discrepancies in mesh resolution of the background and the overset mesh. For more information, see Artificial Thickening for Thin Structures.	**Number of Active Layers** Sets the number of cell layers next to the wall boundary that are active. These active cell layers serve as an artificial thickening of thin geometrical structures. You are advised to set the *Number of Active Layers* so that the total thickness of the layers approximately matches the cell size of the coarse mesh of the other region.

ZeroGapWall

This boundary type is created automatically when you create an overset zero gap interface. It provides the same conditions and values as a regular wall boundary.

Region Settings

Applies to both background and overset regions.

Number of Zero Gap Layers

Added to the **Physics Values** of each region when you create an overset zero gap interface. For an overset zero gap configuration, an overset region and a background region or two overset regions approach each other closely or make contact. With *Number of Zero Gap Layers*, you set the number of remaining cell layers allowed in the gap before they become inactive. You are advised to specify the same *Number of Zero Gap Layers* value for all regions. For a given mesh, this setting lets you control the width of the zero gap that is filled with inactive cells.

Overset Topology

Controls the overset topology of the regions.

Option	Corresponding Physics Value Nodes
Automatic The algorithm detects automatically which region is background region and which region is overset region.	None
Piston Specific option for applications where a piston moves within a cylinder. The cells underneath the piston are deactivated.	**Overset Direction Specification** Sets the direction to control overset mesh hole cutting. Specifies the *Direction* underneath the piston. This part of the background mesh is no longer covered by the piston overset region. However, the cells of this background mesh portion must be deactivated. Normally, the hole-cutting process detects cells to be deactivated only when the background mesh is covered by an overset region. Consequently, the direction in which cells of the background mesh underneath the piston must be deactivated is specified. The direction is specified in the *Coordinate System* selected in the associated expert property.
Direction Specific expert option for applications where the overset boundary does not completely enclose the overset region and this boundary is curved strongly at the same time.	**Overset Direction Specification** Sets the *Direction* to control overset mesh hole cutting. The *Direction* is specified in the *Coordinate System* selected in the associated expert property.
Bounded Specific expert option for applications where the overset boundary contains meshless space that is bounded by a wall boundary of the background region.	None

Overset Mesh Interface Properties

Verbosity

Specifies how much extra information is provided in the *Output* window while the simulation is running. The default is to provide no additional information (**None**). Increasing verbosity is helpful when you are troubleshooting problem cases.

Region-0

Identifies the Overset region (read-only) with higher priority. The priority controls the hole-cutting between **Region-0** and **Region-1**. To swap the association of the Overset region with **Region-0** to **Region-1**, right-click the **Interfaces** > **Overset Mesh 1** node and select **Swap Regions**.

Region-1

Identifies the Background region or the Overset region with lower priority (read-only). The priority controls the hole-cutting between **Region-0** and **Region-1**. To swap the association of the Background region with **Region-1** to **Region-0**, right-click the **Interfaces** > **Overset Mesh 1** node and select **Swap Regions**.

Close Proximity

Activate this option when there is a narrow gap between the moving body and a boundary, such as a wall. This option allows more active cells in the overset mesh between the moving body and the boundary. This feature helps to prevent mesh intersector failure when the overset region overlaps the physical boundaries of the background region.

When the moving body goes close to a boundary, the overlapping region of the overset mesh can extend over the boundary. Normally any cell in the overset mesh that has a vertex outside the boundary (and therefore outside the background region) is deactivated. However, when this option is activated, those cells remain active as long as the cell centroid is within the background region. The cell is deactivated only when the cell centroid is outside the boundary.

Alternate Hole Cutting

Determines whether the global or the layered hole-cutting approach is used. For more information, see Hole-cutting process and Which Hole-Cutting Approach should you use?. When on, activates the alternate hole-cutting process: the Global Approach. When off, activates the default hole-cutting process: the Layered Approach. The default option.

Treat Error as Warning

When activated, overset errors that occur during an overset simulation are treated as warnings. Interpreted as a warning, an overset error does not prevent the simulation that is run from continuing. This option is useful for detecting and investigating overset errors without stopping the simulation. These overset warnings are displayed in the output window.

Tip: To make the overset warnings more prominently visible, right-click in the white space of the output window and deselect **Information**.

Type

Defines the interface type as Overset Mesh Interface.

Interface Settings

Interpolation Option

Controls the overset mesh interpolation. STAR-CCM+ couples the solution in the overset and background regions by interpolating donor cell values to provide data for acceptor cells.

Options	Corresponding Physics Value Nodes
Distance weighted Weighted based on distances to the four nearest neighbors (Default).	None
Linear Linear for 2D and 3D meshes.	None
Linear Quasi2D Linear for 3D meshes with only one layer of cells.	None
Least Square Least square for 2D and 3D meshes.	None

Conservation Option Properties

When you create an overset mesh interface, STAR-CCM+ provides the **Continua** > **[physics]** > **Models** > **Segregated Flow** > **Conservation Option** node. Selects the conservation option for the segregated flow model.

None

No special mass conservation treatment is applied to the pressure-correction equation.

Source Term

Mass conservation is enforced at the overset interface by adding a distribution of the overset mass imbalance to the source term of the pressure-correction equation. For more information, see Source Term Correction.

Use this option only for incompressible flows that are enclosed by walls. Activating this option in other situations can cause non-physical solutions.

Flux Correction

Mass conservation is enforced at the overset interface by adjusting interface fluxes such that their sum is zero. For more information, see Flux Correction.

You can apply this option generally to compressible and incompressible, non-closed flow situations. Applicable examples are rotating gear pumps or moving cars. This option must not be used for moving piston simulations with dynamic cell activation and deactivation.

Overset Load Balancing Solver Properties

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes or to keep the partitioning fixed. Deactivated by default.

Enabled

When activated, overset load-balancing is turned on for parallel servers.

Acceptor Weight

Controls the relative weight of the acceptor cells compared to other cells. The default value of 1 corresponds to weighting the acceptor cells twice as much as other cells. A value of 0 does not assign a specific weight to acceptor cells.

Rebalance Limit

Sets the point at which the load balance is sufficiently distributed and the solver can stop. The limit is the ratio of the loading between the most and least loaded domains. Choose a value of the limit that avoids rebalancing efforts beyond useful levels. The lower the limit, the less frequent the rebalancing. Setting the value close to 1 results in 'always rebalance' behavior. The default value is 0.8.

Rebalance Period

Number of iterations or time-steps after which the overset load balancing solver checks if a load imbalance prevails. If it prevails, that is, the *Rebalance Limit* is not reached, rebalancing takes place.

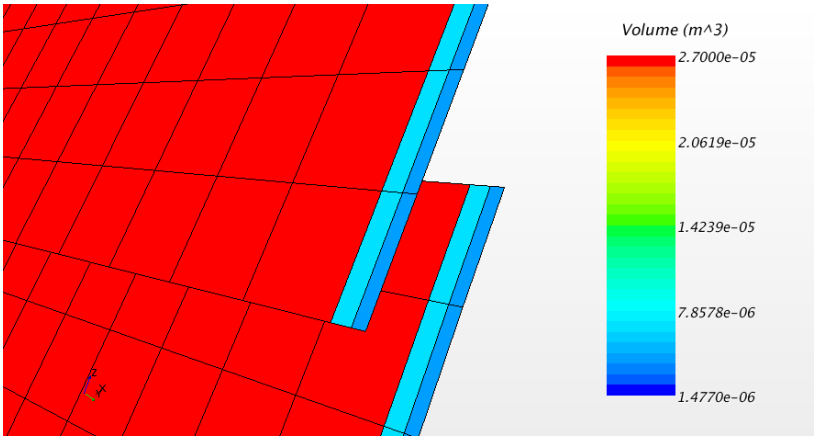
If you combine overset mesh with a Lagrangian multiphase simulation, for which load balancing is also available, the *Rebalance Frequency* of the **Solvers** > **Partitioning** > **Load balancing options** node influences the rebalance period. Rebalancing is then performed every *Rebalance Frequency* * *Rebalance Period* iterations or time-steps. For example, if you set *Rebalance Period* to 10 and *Rebalance Frequency* to 2, the load balancing solver checks every 20 iterations or time-steps if a load imbalance still exists.

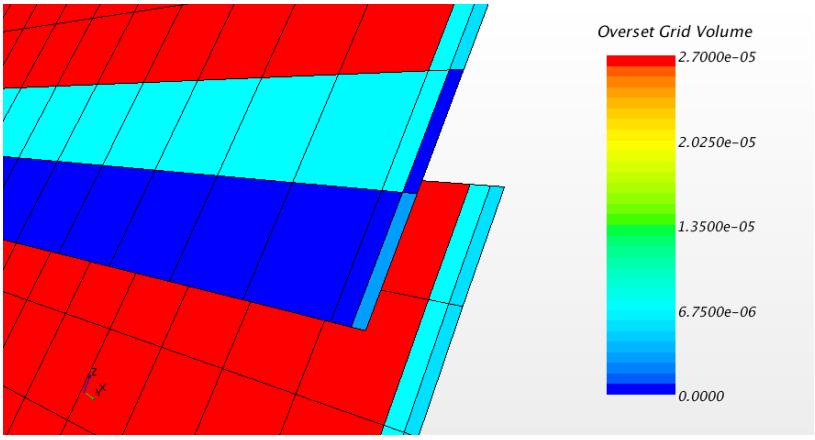
Overset Mesh Field Function Reference

The following primitive field functions are made available to the simulation with overset mesh. Both the volume mesh as well as the overset mesh representation can be used to display these field functions.

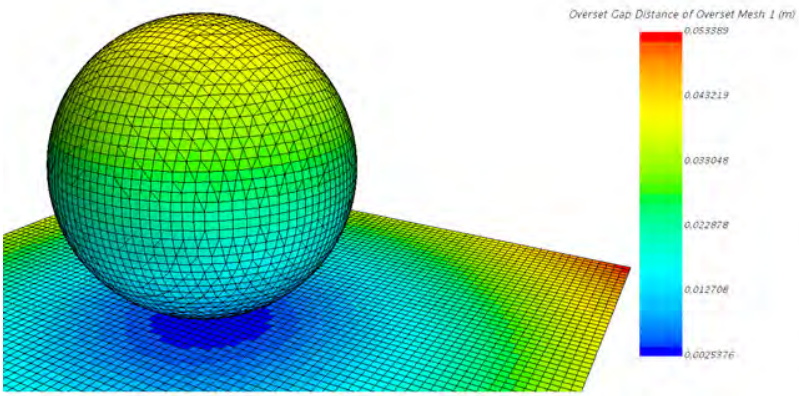
The volume mesh representation displays only the active cells. If you want to see the acceptor cells, use the overset mesh representation. These field functions have a typical set of base properties.

ChimeraGridTestData	Used to check the interpolation error at the Overset Mesh Interface.
Overset Cell Status	Used internally as a flag in the solvers to identify the status of each cell during the solution process of the equations. This function provides more information about the result of the Overset Assembly process. The available values are: • 1 - inactive • 0 - active • -2 - acceptor • -1 - acceptor that is used as donor
Overset Cell Type	Used by the Overset Assembly process to identify the status of each cell. It gives more information about the cells with regard to the Overset Assembly process. Can be used to analyze errors. The available values are: • 0 - active • -1 - inactive cell that resides outside of the background mesh • -2 - inactive cell that is covered by another overset region. Consequently, this type can only occur for multiple overset meshes. • -3 - inactive cell inside a zero gap zone • 1 - donor • 2 - active, intermediate cell layer used by the hole cutting process. • 3 - acceptor
Overset Error Status	Indicates the status of each cell that has an error. The available values are: • 0 - cell has no error • 5 - cell is an acceptor cell that has inactive donor cells. • 6 - cell is an acceptor cell that has zero donor cells. • 20 - cell is a donor cell, but is inactive. • 30 - an inactive face: that is, a face between an active cell and an inactive cell.

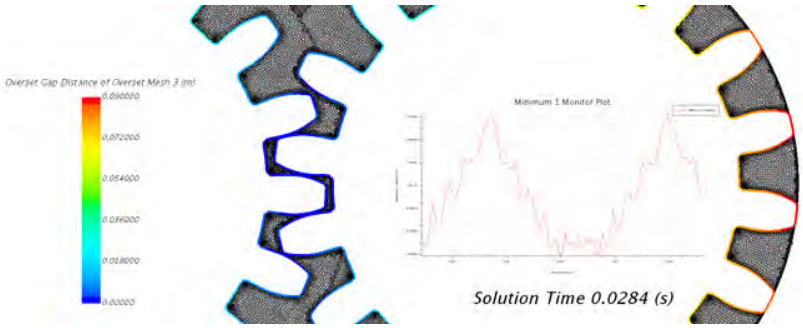
Overset Grid Area	Estimates the area of all regions in the simulation. For overset mesh simulations, this area estimation is more accurate than the area computation of the **Area** field function, because it avoids double-summation of the area of overlapping cells in overlapping regions. The **Overset Surface Integral** report uses this field function for a more accurate integration compared to the standard **Surface Integral** report. See Overset Surface Integral Report.
Overset Grid Volume	Estimates the volume of all regions in the simulation. For overset mesh simulations, this volume estimation is more accurate than the volume computation of the **Volume** field function, because it avoids double-summation of the volume of overlapping cells in overlapping regions. The **Overset Volume Integral** report uses this field function for a more accurate integration compared to the standard **Volume Integral** report for overset simulations. See Overset Volume Integral Report. When you use this field function, either through an **Overset Volume Integral** or for visualization in a scalar scene, the simulation runtime can increase significantly. To speed up the simulation time, do not evaluate the field function at every time step, but select **Monitors** > **[Overset Volume Integral Monitor]** > **Time-Step Frequency** and increase *Frequency*. You can similarly increase the time-step frequency for a scene by selecting **Scenes** > **[Scene]** > **Attributes** > **Update** > **Time-Step Frequency**. 

	
Overset Interpolation Type	Indicates which interpolation type STAR-CCM+ is using for the interpolation between background mesh and overset mesh. Although you specify the interpolation type (see Overset Interpolation Option), STAR-CCM+ can switch to a more robust interpolation scheme in the course of the simulation when the current mesh configuration makes it impossible to apply the selected interpolation scheme. For example, when you select a **Linear** or a **LinearQuasi2D** interpolation scheme, STAR-CCM+ can revert to the more robust **Distance weighted** scheme if the conditions for the selected interpolation scheme cannot be met. The possible values are: • 0 – Distance Weighted • 1 – Linear • -1 – No overset interface present • -2 – Not an acceptor cell, that is, no interpolation takes place from donor to acceptor • 4, 5, or 6 – Inverse Distance type interpolation. Indicates that the interpolation is difficult and less accurate. When you see these values, you are advised to improve the mesh, for example, by refining the overlap area or by ensuring similar cell sizes for background and overset mesh in the overlap area.
Overset Zero Gap Distance Field Function	Provides the minimal distance between wall-type boundaries of connected overset regions, or an overset region with a background region for the purposes of determining the cell-density required on the boundary surfaces to resolve the flow in small gaps (using prism layers) The gap distance is calculated from the boundary to each cell face-centroid on the boundaries of the connected regions. This can be between boundaries on two overlapping overset regions or between an overset region and the background region.

The **Overset Zero Gap Distance Field Function** is only available when you create overset mesh interfaces. See Preparing Overset Regions. This field function is only available for determining the minimal distance between wall-type boundaries—such as wall, contact, baffle, or porous-baffle boundaries.



The zero gap distance can be visualized through creating reports, plots or scalar scenes. For example, it may be prudent to report and monitor the minimum gap between two surfaces as they move together or apart over time (for example, a valve plug closing against its seat, or teeth on a gear-wheel engaging with its counterpart).

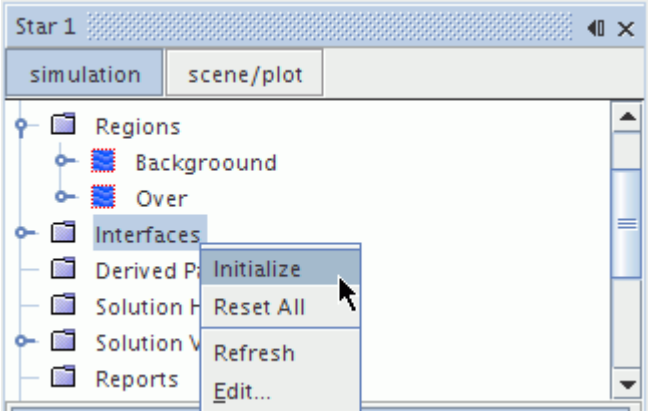


Troubleshooting Overset Mesh

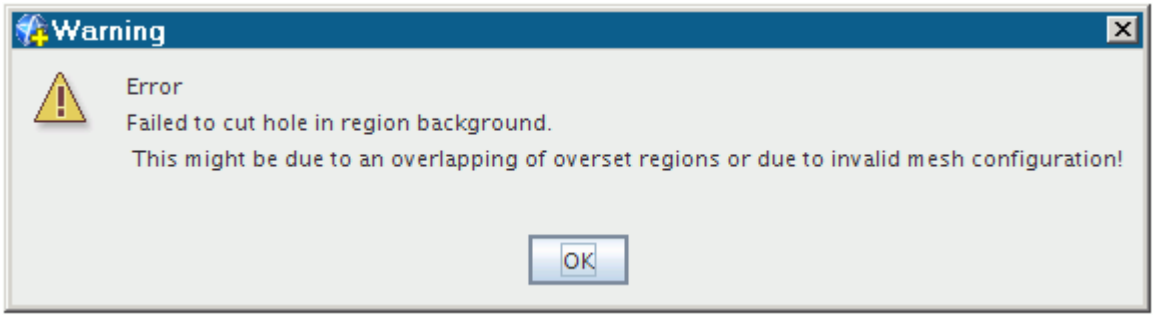
Several troubleshooting features are available for analyzing the quality of the overset mesh.

You can generate the overset mesh without running the simulation by initializing the Overset Mesh Interface:

- Right-click the **Interfaces** node and select **Initialize**.

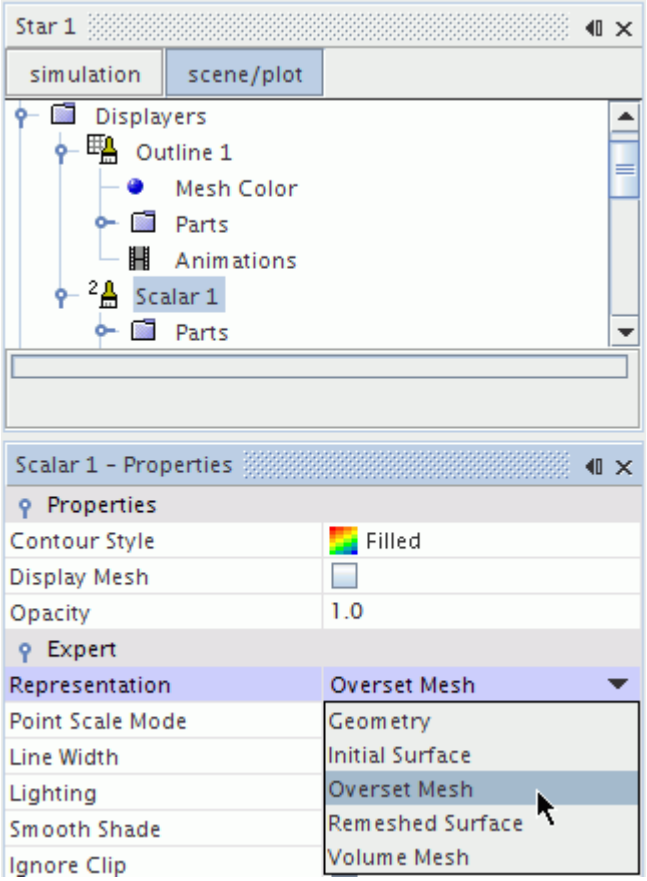


If the background and overset regions were not successfully coupled, the following error message appears.



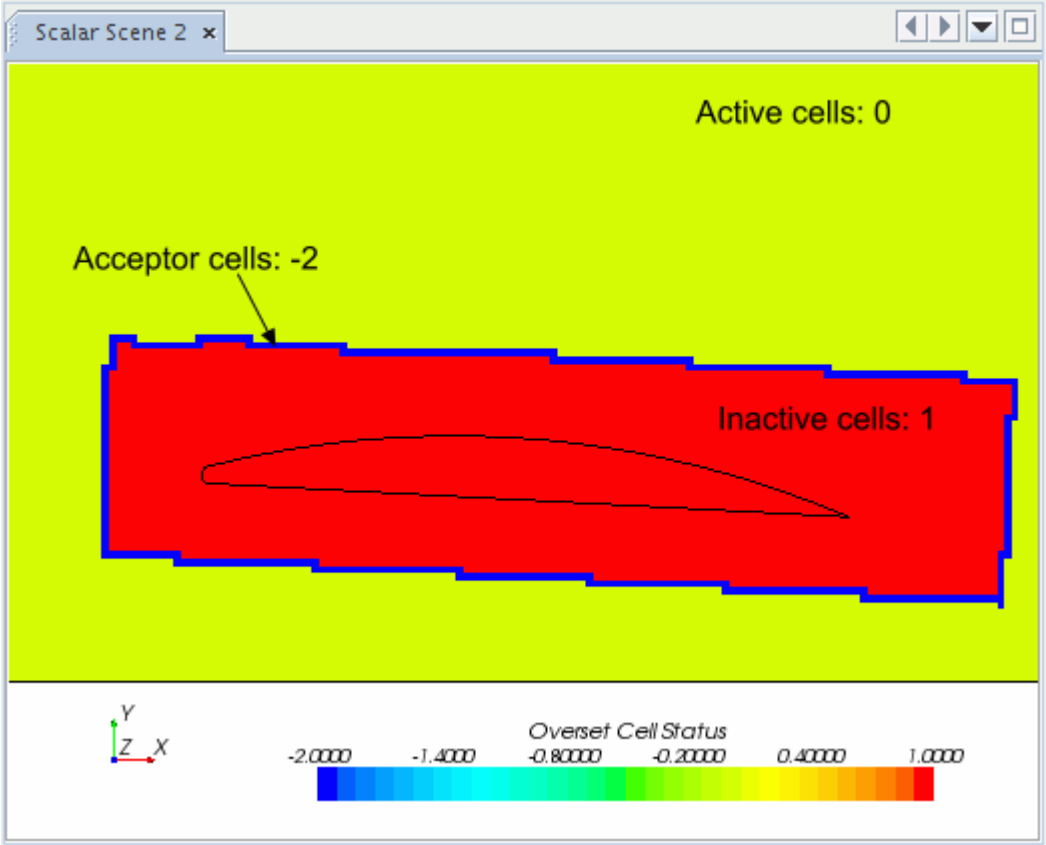
You can check the cell status of the overset mesh by creating an Overset Cell Status scalar field. For a list of overset cell status values, see Overset Mesh Field Function Reference.

- Set the scalar mesh representation to **Overset Mesh**.

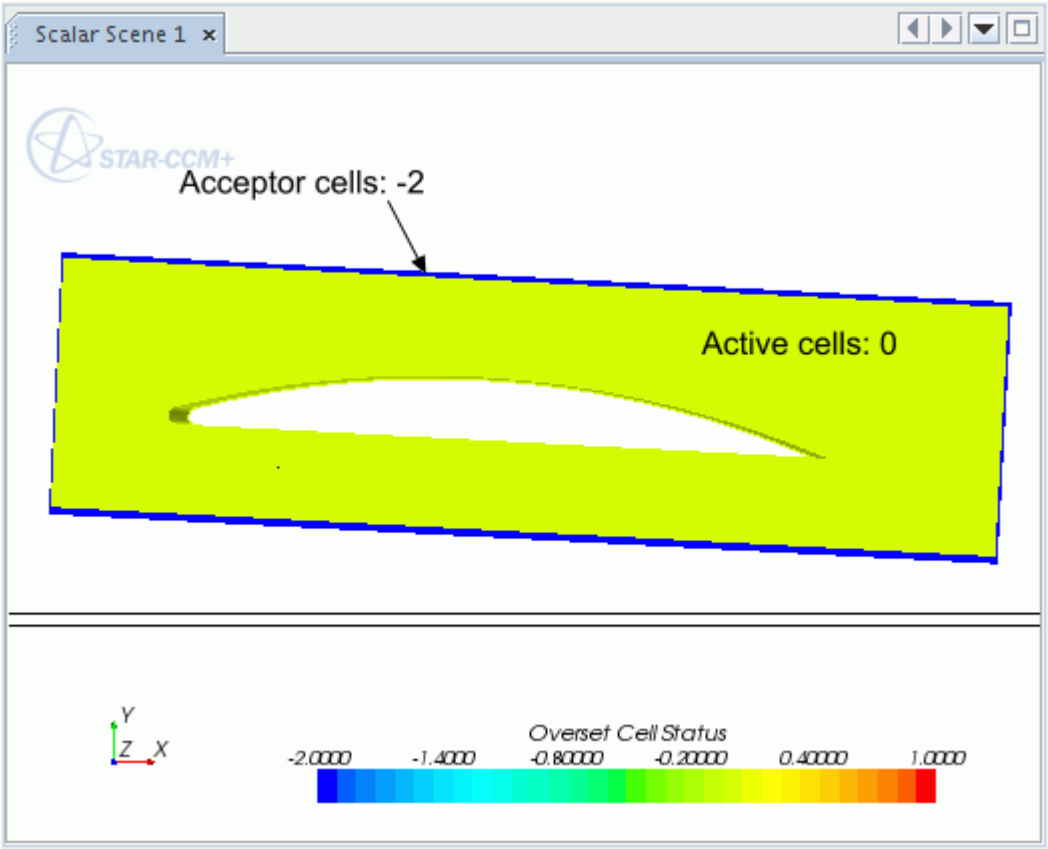


Check each region separately. It is recommended that you manually change the scalar bar range from -2.0 to 1.0. In this case, all active cells have a value of 0, inactive cells have a value of 1, and acceptor cells have a value of -2. Acceptor cells must separate the active cells from inactive cells; direct contact between active and inactive cells is not allowed.

Below is the scalar field for the background region. The overset region cells have been deactivated.



The overset region only includes mesh cells for which the governing equations are solved.



All other scalar field options available for mesh analysis can be found under field function reference.

Viewing Overset Error Cell Sets

When an error occurs during the overset assembly process at the overset interface, a cell set is created that contains all cells with **Overset Error Status** not equal to 0. To create successive cell sets in the course of the simulation:

- Select **Interfaces** > **Overset Mesh 1** and activate *Treat Error as Warning*.

Analyzing the Overset Interpolation Type

The overset interpolation type can vary during the simulation when the mesh requirements for the specified interpolation type are not met. To analyze which overset interpolation type is actually used:

- Create a scalar scene and display the field function **Overset Interpolation Type**.

For more information, see Overset Mesh Field Function Reference.

Reporting the Interpolation Quality

To monitor the quality of the interpolation between the overset mesh and the background mesh over time, you can create a specific overset report—the Overset Interpolation Quality Report.

See Also:

Overset Mesh Field Function Reference

Overset Mesh Bibliography

[12] *Hadzic, H. 2005. "Development and application of Finite Volume Method for the Computation of Flows Around Moving Bodies on Unstructure, Overlapping Grids", Ph.D. Dissertation, Technische Universität Hamburg-Haburg.*

Modeling Space, Time, and Motion

Contents:

Modeling Space
Coordinate Systems
Modeling Time
Using Reference Frames
Modeling Motion

See Also:

Modeling Heat Transfer in Solid Shells

Modeling Space

The primary function of the Space models in STAR-CCM+ is to provide methods for computing and accessing mesh metrics. Examples of mesh metrics include cell volume and centroid, face area and centroid, cell and face indexes, and skewness angle.

This section describes the four Space models in STAR-CCM+ and how to select one:

- The Axisymmetric Model
- The Shell Three-Dimensional Model
- The Two-Dimensional Model
- The Three-Dimensional Model

Reference information is also provided for field functions.

What Is an Axisymmetric Model?

The Axisymmetric model is designed to work on two-dimensional axisymmetric meshes. When using the model, orient the mesh such that the axis of rotation is at y = 0 in global coordinate space. No part of the mesh can be below y = 0, and the boundary edge that lies along the axis must be of type Axis.

For boundary conditions and reporting purposes, the mesh is assumed to be swept through an angle of 1 radian. For applications that use a mass flow inlet, the mass flow is therefore given in kg/rad s. Any volumetric or area quantities reported for the Axisymmetric model are assumed to be for a 1-radian sector. If you want the value of a quantity for the full 360° revolution, multiply reported quantities by 2π radians.

What Is a Shell Three-Dimensional Model?

The Shell Three-Dimensional model represents a volume in space whose thickness is sufficiently small that it can be considered as a surface with a single-cell thickness in the normal direction. This model must be associated with a shell region, and is required when:

- Modeling heat conduction within a solid shell; and,
- Modeling fluid films.

What Is a Two-Dimensional Model?

The Two-Dimensional model is designed to work on two-dimensional meshes. Activate this model only if the mesh is indeed two-dimensional. In this model, the mesh is assumed to have a unit depth (in SI units) so that any volumetric or area quantities reported for the two-dimensional model are assumed to be “per meter”.

What Is a Three-Dimensional Model?

The Three-Dimensional model is designed to work on three-dimensional meshes. Activate this model only if the mesh is indeed three-dimensional. The use of a one-cell-thick three-dimensional mesh is much less efficient than using a true two-dimensional mesh for two-dimensional and axisymmetric simulations. A better approach is to extract a two-dimensional mesh.

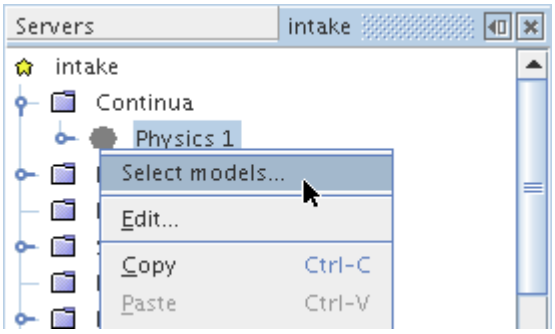
Contents:

Selecting a Space Model
What Is a Shell Region?
Setting an Axis
Space Field Functions Reference

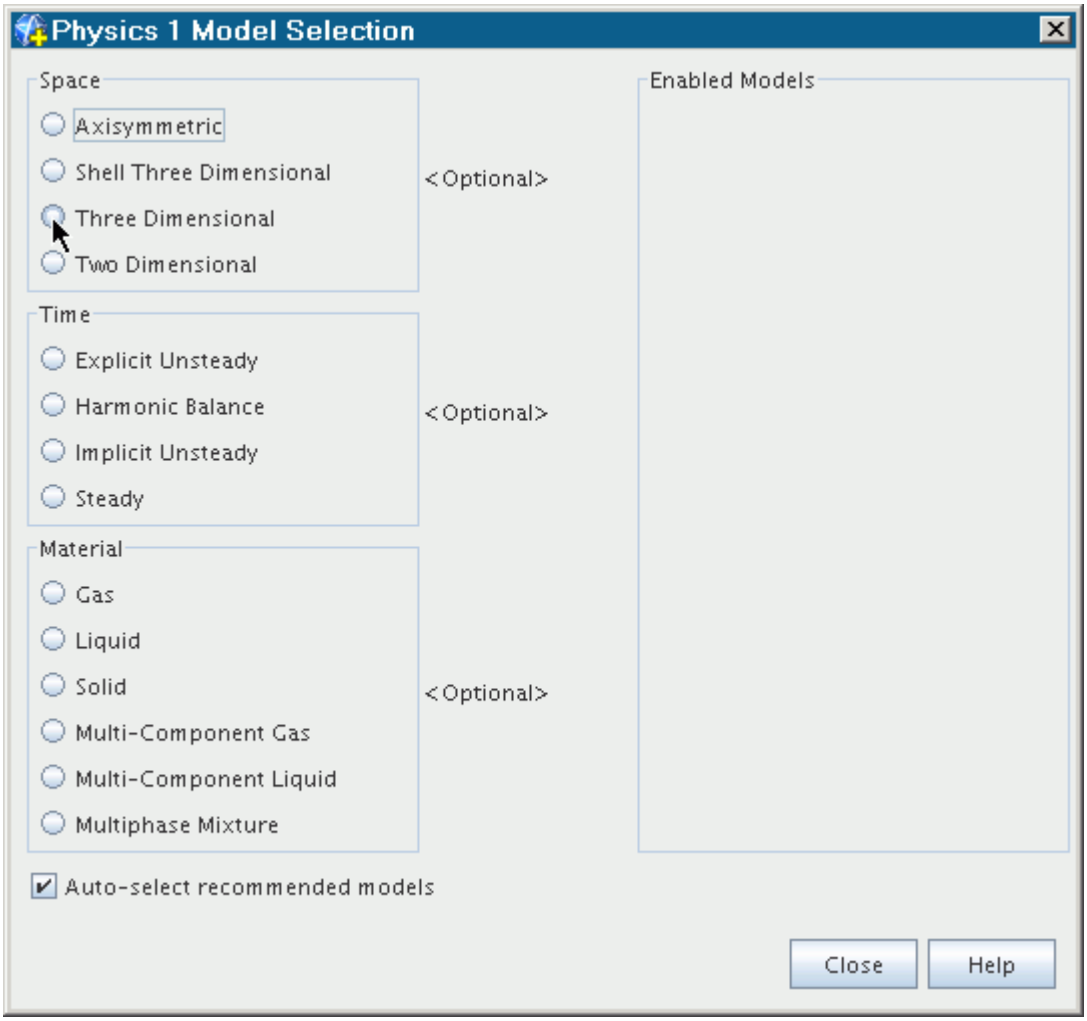
Selecting a Space Model

To select a Space model, open the **Continua** node of the simulation tree, then right-click the **Physics 1** node.

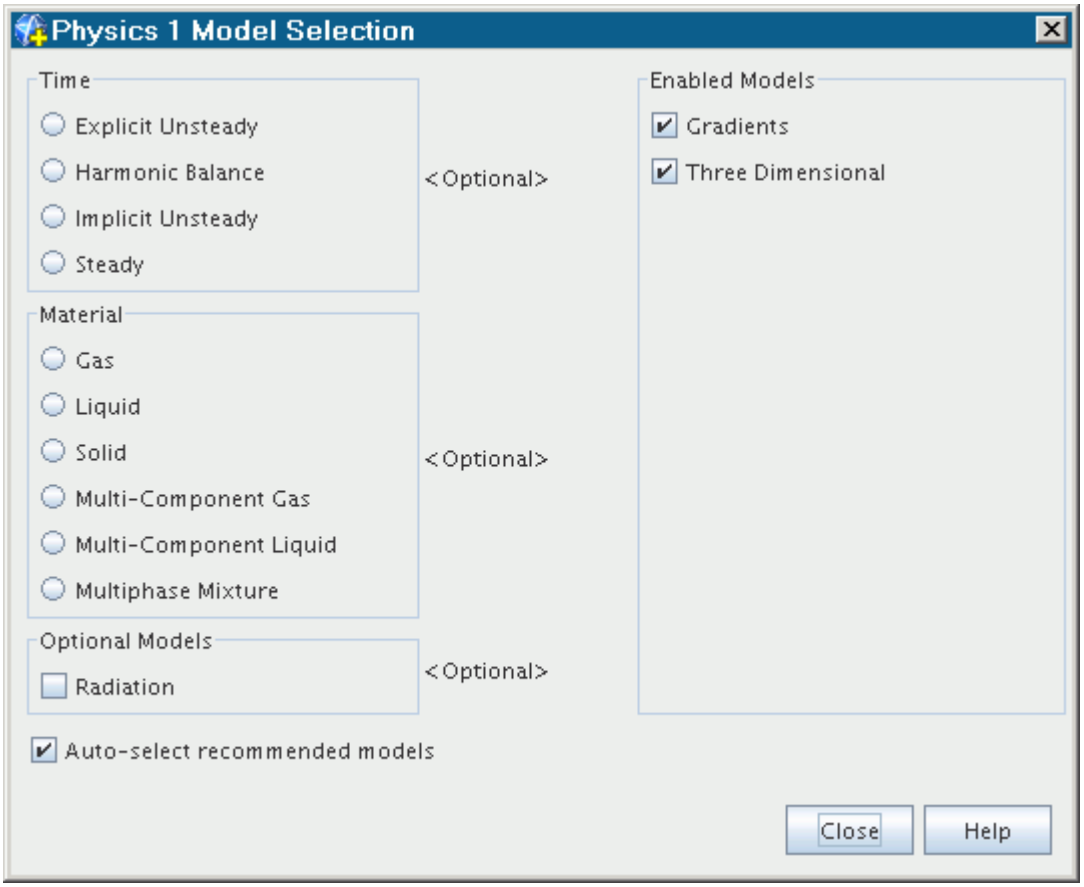
Select the **Select models...** item from the pop-up menu.



This command opens the *Model Selection* dialog. Select the correct Space model for the simulation by clicking the relevant radio button.

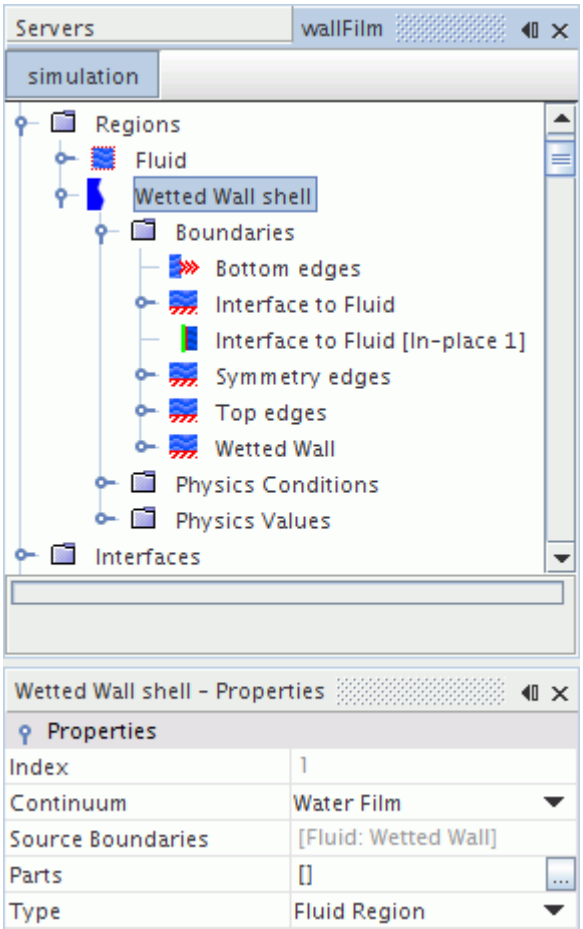


The following model selection dialog shows the Space model activated.



What Is a Shell Region?

The Shell Three-Dimensional model is used with shell regions. Shell regions, which have properties and a pop-up menu, are created from existing surfaces (boundaries) contained within regular regions.



It is not possible to create a shell region directly from geometry parts or baffles. However, you can use the Insert Shell Region Interface option to create a shell region at an in-place (baffle) interface.

Shell Region Properties

Index	Index reference for shell region (read-only).
Continuum	Defines the fluid film phase or physics continuum that is associated with this region. Only physics continua where the Shell Three-Dimensional space model is selected is shown in the drop-down menu.
Source Boundaries	The boundaries that are used to create this shell region (read-only).
Parts	List of parts that are associated with this region. Editing the property accesses the in-place object selector to add or remove parts.
Type	Defines the type of the shell region depending on the region from which it was created: fluid, solid, or porous.

Shell Region Pop-Up Menu

Transform	Scales, translates, or rotates the selected regions and all associated boundaries. These actions operate on one or multiple regions, and if

	more than one region is selected, the transform action is applied to each region in sequence.	
	Scale...	Executes the scale region action.
	Translate...	Executes the translate region action.
	Rotate...	Executes the rotate region action.
Reorder Mesh	Reorders the internal references to cells in the mesh region in order to optimize the solver performance in terms of memory usage.	
Solution	Applies operations in the submenu to the individual region.	
	Clear	Executes the clear region solution action.

Contents:

Creating a Shell Region

See Also:

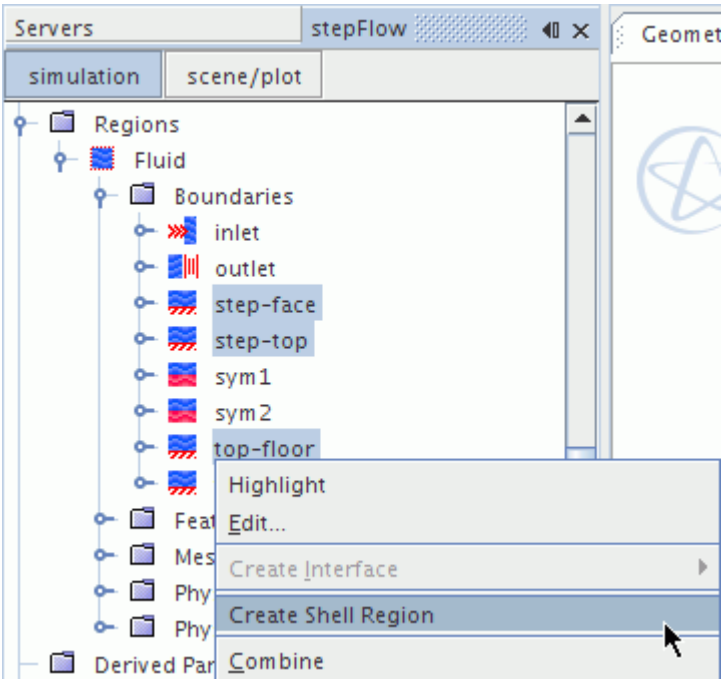
Inserting a Shell Region Inside an Interface

Defining the Regions Layout

Creating a Shell Region

Shell regions are created from existing region boundaries.

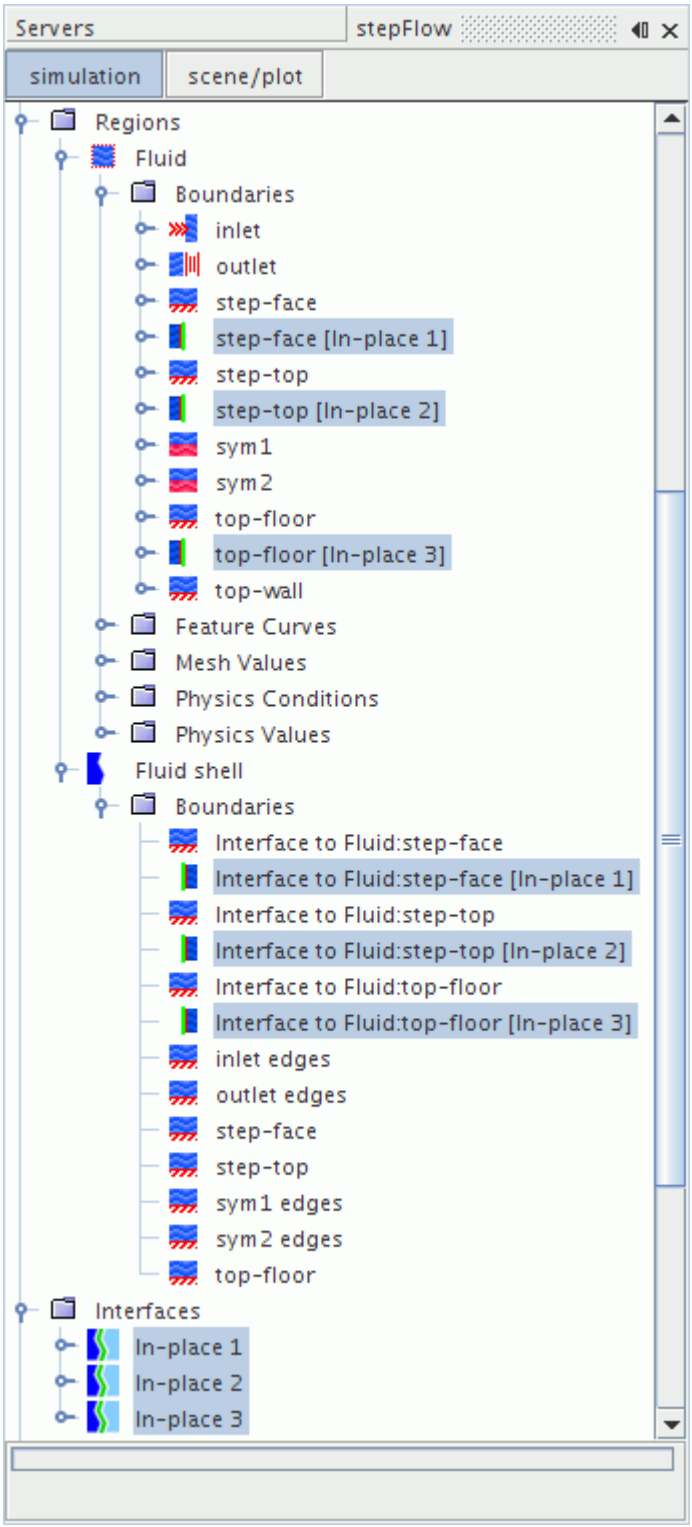
- Right-click the region boundary or boundaries from which you wish to create a shell region and choose `Create Shell Region` from the pop-up menu.



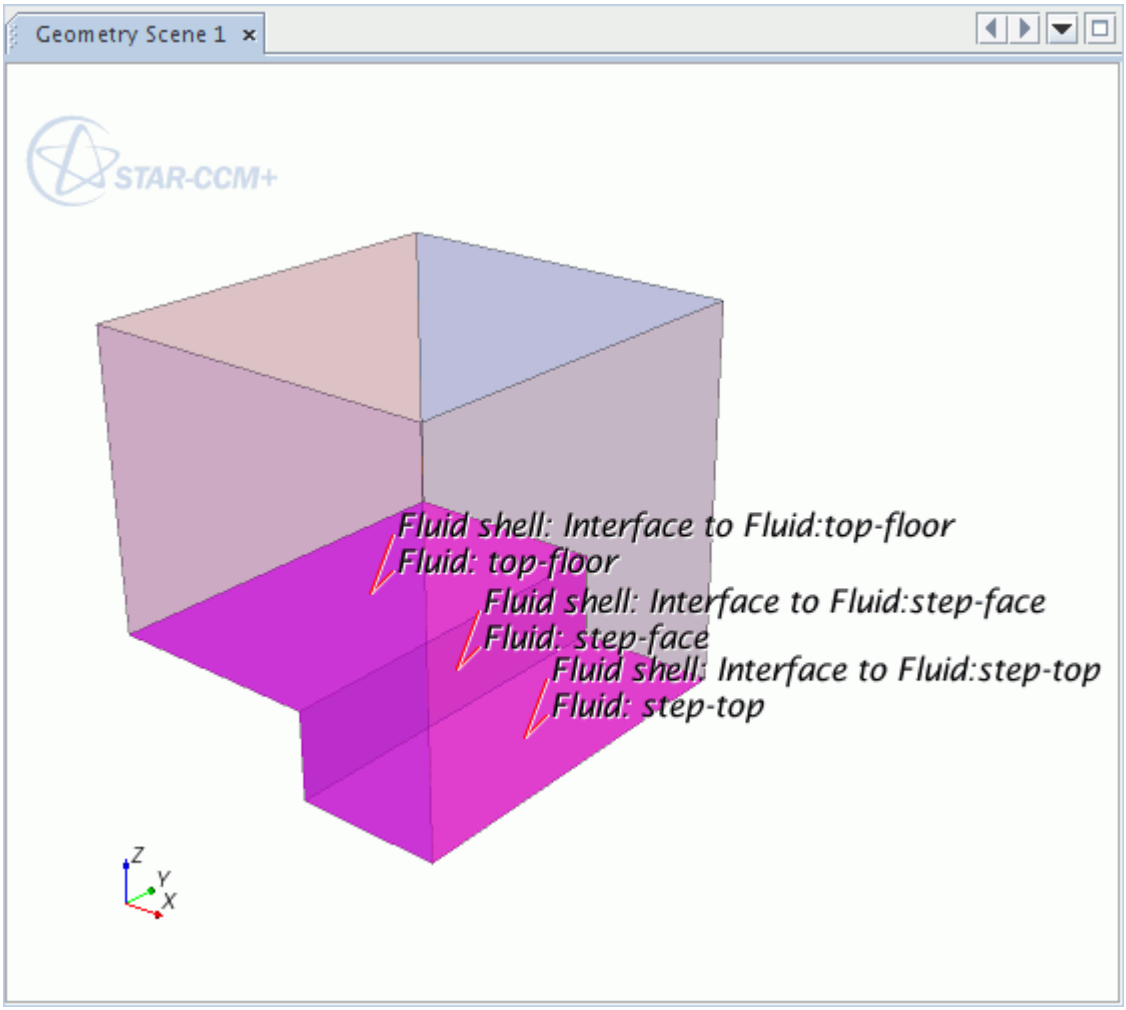
Several objects are added to the simulation as a result of this operation:

- A single shell region is created when multi-selecting the contiguous boundaries belonging to a single region (as is shown in the preceding example). If the selected boundaries belong to different regions, several shell regions are created. These shell regions must be connected with interfaces.
- Interfaces between the surface of the shell region and the boundaries from which it was created.
- The corresponding interface boundaries in both the source region and the shell region.

These objects are shown in the example below:



In the example that was used above, the following shell region was created.



- After you have created the shell region, set its *Continuum* property to point to either of the following items:
  - A separate physics continuum that uses the Shell Three-Dimensional model. Use this property assignment when modeling heat conduction or electrical conduction in solid shells.
  - A fluid film phase.

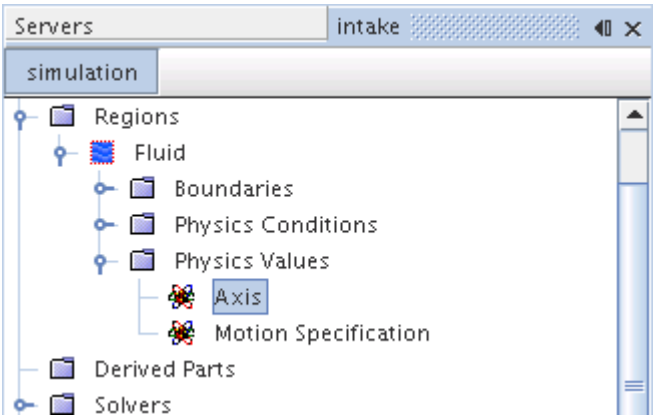
Setting an Axis

An axis is used to define an axis of rotation when the simulation requires it.

The **Axis** node, which represents the axis, appears as:

- A region value when a space model has been selected for the associated physics continuum, and
- A boundary value on a non-slip wall boundary when the tangential velocity specification condition is set to local rotation rate.

  The properties of the node define the axis.



This axis definition provides information that is required when STAR-CCM+ creates periodic interfaces for a region, and for transformations involving these interfaces. The tangential velocity specification at a boundary also uses the axis definition to specify the local axis and origin of wall rotation.

Axis Properties

Direction	A Cartesian unit vector specifying the direction of the rotation.
Origin	Cartesian position vector specifying the origin of the axis.
Coordinate System	Specifies the coordinate system (laboratory by default) for defining the origin and axis. The drop-down list of this property includes any local coordinate systems that exist in the simulation.

Space Field Functions Reference

The following field functions are made available to the simulation when any one of the Space models is used. They include a typical set of base properties.

Cell Aspect Ratio	This field function is used to identify regular polygonal cells from skewed or stretched cells. A value of 1 indicates that the cell is a regular polygon, and a value close to zero would represent a highly stretched cell. This ratio is defined as: $$CAR = \frac{ND \times NF \times max(V,0)}{\left(\sum_f \left\|A_f\right\|\right) \cdot \left(\sum_f \left\|\frac{A_f \cdot dx}{\left\|A_f\right\|}\right\|\right)}$$ where ND is the number of dimensions (three for a 3D mesh and two for a 2D mesh). NF is the number of faces of the cell. V is the volume of the cell. A_f is

	the cell face area vector and dx is the position vector from the face centroid to the cell centroid.
Centroid	Displays the position of the face or cell centroid. At face and cell centroids, the value is the same as the Position field function.
Chevron Quality Indicator	This field function is used to identify chevron cells. These cells are pairs of thin slender cells that share a common face whose angle is such that the line joining the cell centers does not pass through the common face. This indicator function marks chevron cells as 1 and every other cell as 0. A chevron quality norm c , as defined below, is used to determine whether a cell can be classified as a chevron cell or not. The chevron quality norm is computed using a vector dp that is projected from the face center to the line joining the cell centers: $$dp = \left(\frac{A_f \cdot dx}{A_f \cdot ds}\right) ds - dx$$ $$c = 1 - \frac{dp \cdot dp}{max(dp \cdot dv)}$$ where: A_f is the face area vector dx is the vector from the face center to the cell center ds is the vector connecting the two cell centers dv is the vector from the face vertex to the face center. The max function, in evaluating the chevron quality norm, is taken over all the face vertices. The chevron quality norm is: • 1 for orthogonal cells in which the line connecting the two cell centers passes through the face centroid • 0 when the line passes through an edge of the face • negative when the line is outside the face.

	The chevron quality indicator is marked 1 when the chevron quality norm is negative.
	Two more properties are provided: • *Chevron Threshold Angle*—the maximum angle included by the two cells, beyond which the chevron quality indicator is not computed and is set to 0. You specify this value. • *Cell Aspect Ratio Tolerance*—if the Cell Aspect Ratios of both the cells in question are greater than this specified tolerance, then the chevron indicator is not computed and is set to 1. This tolerance prevents STAR-CCM+ from marking two irregular sized cells connected by a small face.
Position	The position vector represents the position or coordinates of a given location. It is available at vertices, faces, and cells. At faces and cells, the value represents the position of the respective face and cell centroids. This field function is the preferred option for specifying position dependencies of profiles and material properties when using field function methods.

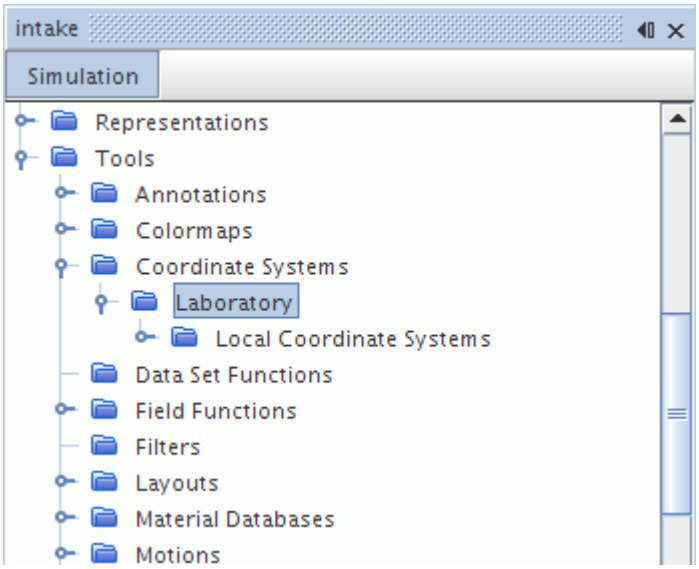
Coordinate Systems

Coordinate systems are used to specify spatial vectors, such as position and direction for boundary and region values, initial conditions, and properties of location and direction for derived parts. Coordinate systems are also used to define the position of sketch planes in 3D-CAD. For simple applications, the default laboratory coordinate system is sufficient. However, more complex situations can benefit from the definition and use of a local coordinate system or a block-mapped coordinate system.

In STAR-CCM+, any coordinate system is represented by a node under **Tools** > **Coordinate Systems**.

Laboratory Coordinate System

The laboratory coordinate system is the base system in which all other local coordinate systems are defined. The laboratory coordinate system is a Cartesian system. It exists in every STAR-CCM+ simulation by default, and cannot be deleted.

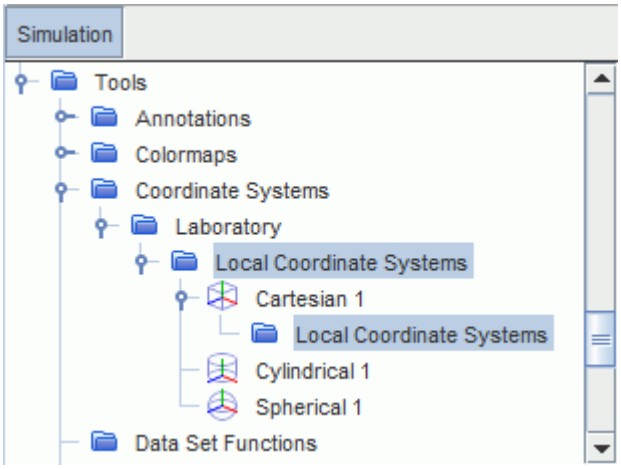


The laboratory coordinate system is represented by the **Laboratory** node. This node contains a child node, **Local Coordinate Systems**, that is used to manage all local coordinate systems (and their derived systems) within the simulation.

Local Coordinate System

A local coordinate system can be defined in terms of cartesian, cylindrical, or spherical coordinates, and can take either the base laboratory coordinate system, or a previously defined cartesian local coordinate system, as the reference system. A local coordinate system can also be exported from a sketch plane in 3D-CAD for use in STAR-CCM+.

In the simulation tree, the reference system acts as a parent node. Several layers of nesting are permissible, up to the memory limit of the computer. The **Local Coordinate Systems** manager node manages local coordinate systems and is a child node of the relevant reference coordinate system.



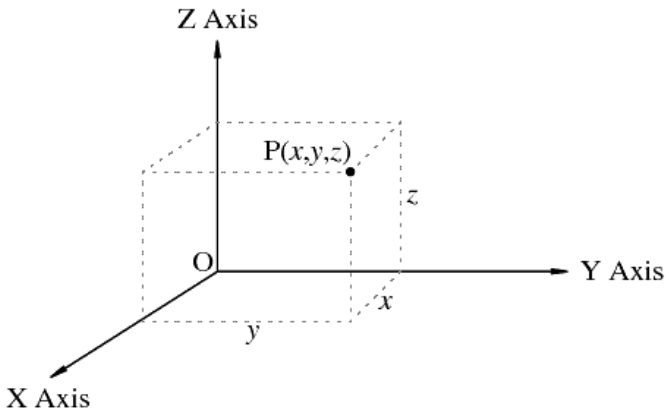
Local coordinate systems can be accessed by derived parts and reports, and also serve as scalar and vector field functions. Examples of how they can be used are:

- Initializing the velocity in a region to solid body rotation
- Providing a swirling inlet velocity profile

Three types of user-defined local coordinate systems are available:

Cartesian

This type of coordinate system is defined by an origin and three right-handed orthogonal direction vectors for the X Axis, Y Axis, and Z Axis. Points are described using x, y, and z coordinates in the reference coordinate system (identified as the coordinate system containing the **Local Coordinate System** manager node to which the current system belongs). Vectors are described using basis vectors aligned with the X, Y, and Z Axes of the reference coordinate system.

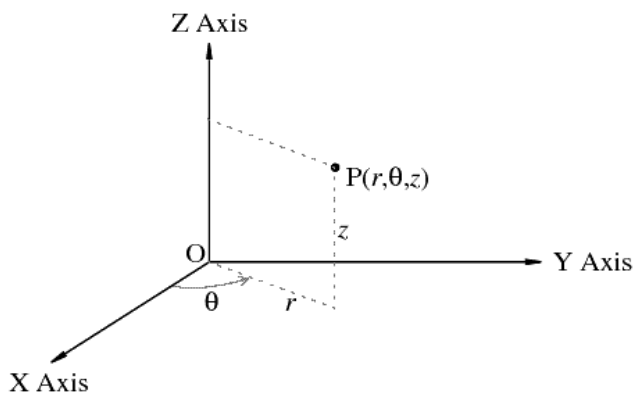


A variation of this coordinate system is available with 6-DOF motion: the body-fitted coordinate system. Unlike the other local coordinate systems, the body-fitted one is only created automatically when you create a 6-DOF body with using 6-DOF motion in the simulation. A set of axes that stays “fixed” to a rigid body as it undergoes motion helps with reporting various quantities, especially forces, and moments in a body reference frame.

Cylindrical

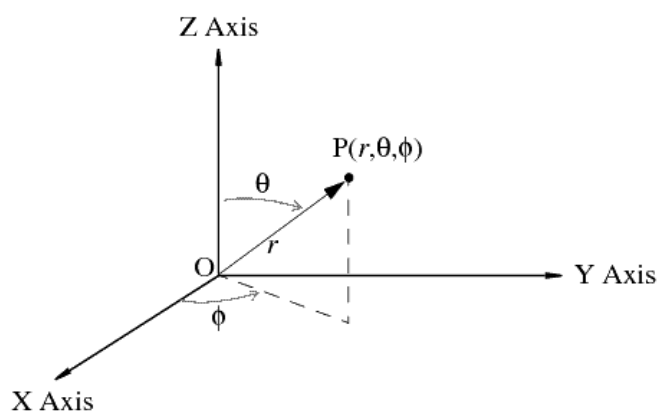
This type of coordinate system is defined by a reference Cartesian coordinate system with an origin and three right-handed orthogonal direction vectors for the radial, tangential, and axial axes. The reference coordinate system can be either the Laboratory system, or a Cartesian local coordinate system.

Points are described using r, θ, and z coordinates. Vectors are described as the radial, tangential, and axial components of the vector in the local basis of that vector (see: Transformations of Points and Vectors to Curvilinear Coordinate Systems). This coordinate system is common in simulations that are rotating about an axis, or where a swirling velocity profile is used.



Spherical

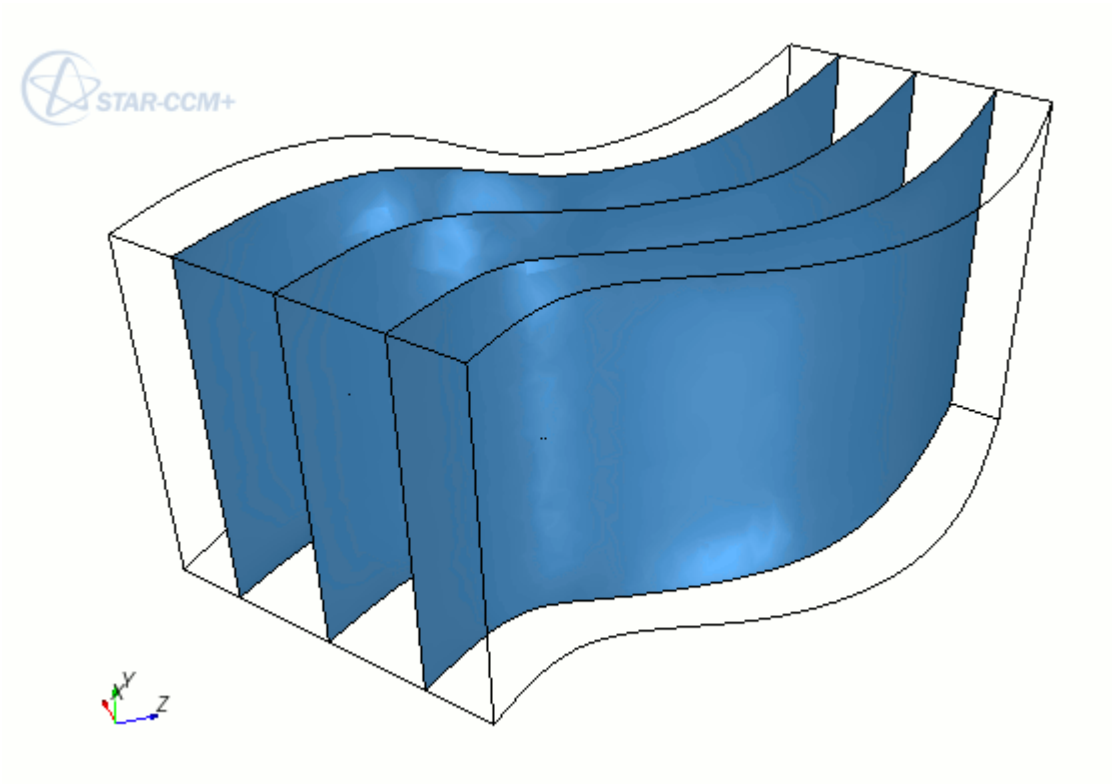
An underlying Cartesian coordinate system, with an origin and three right-handed orthogonal direction vectors for the X Axis, Y Axis, and Z Axis defines this type of coordinate system. Points are described using r, θ, and ϕ coordinates. Vectors are described as the radial, polar, and azimuthal components of the vector in the local basis of that vector (see: Transformations of Points and Vectors to Curvilinear Coordinate Systems). This coordinate system is useful for simulations in which there is spherical symmetry or rotation about a point.



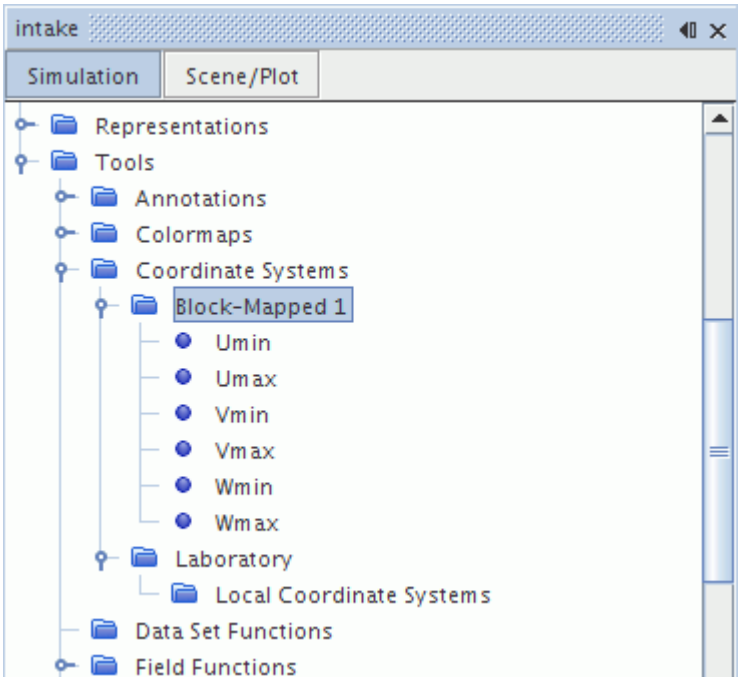
Block-Mapped Coordinate System

A block-mapped coordinate system is a type of user-defined coordinate system in STAR-CCM+, that defines points based on evenly distributed gradations between selected end surfaces in a topological cube. Each coordinate direction has a range of 0 to 1, where [0, 0, 0] defines one corner of the cube, and [1, 1, 1] defines the diagonally opposite corner of the cube.

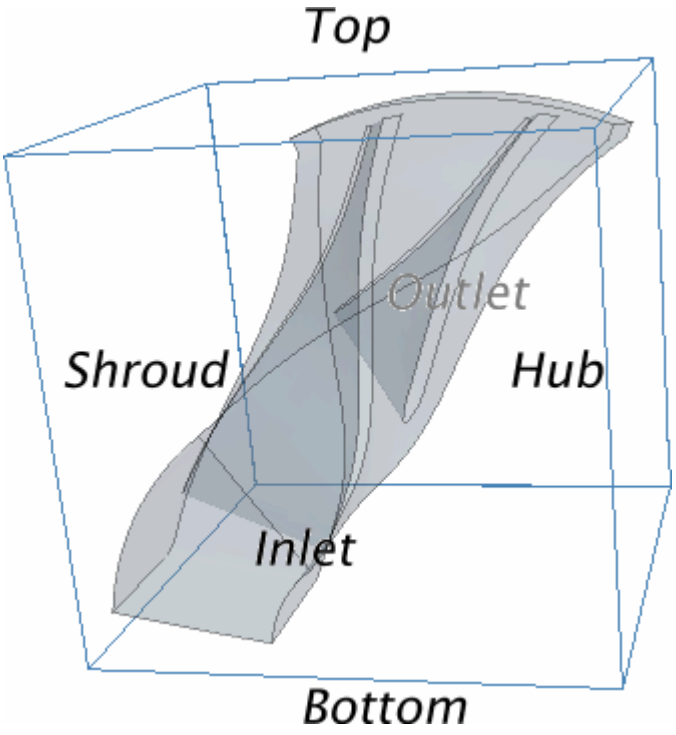
This type of coordinate system makes it possible to create iso surfaces that follow the contours of the bounding surfaces, as would be desired, for example, in turbomachinery post-processing for generating blade-to-blade or meridional views.



The system is represented in the simulation tree by a node that has its own properties and is subject to common menu items. This coordinate system permits the creation of section surfaces along periodic directions, and has its own field functions.



The six subnodes in the **Block-mapped** node, known as block-mapped boundaries, represent the six sides of the topological cube. Each side should consist of one or more boundaries that you select when creating this coordinate system. All boundary edges in the group should be connected to one other boundary to form a closed cube.



Contents:

Transformations of Points and Vectors to Curvilinear Coordinate Systems
Creating a Local Coordinate System
Creating a Local Coordinate System Interactively
Editing a Local Coordinate System Interactively
Transforming a Local Coordinate System
Copying Properties of Local Coordinate Systems
Creating a Block-Mapped Coordinate System
Displaying a Coordinate System in a Scene
Coordinate Systems Reference

Transformations of Points and Vectors to Curvilinear Coordinate Systems

STAR-CCM+ supports two types of curvilinear coordinate systems: cylindrical coordinates and spherical coordinates.

The transformation of a point $\mathbf{p}$ to a local curvilinear coordinate system from its parent coordinate system is done in two steps:

- Computation of $\mathbf{p}$'s coordinates with respect to the local Cartesian system.
- Conversion of $\mathbf{p}$'s local Cartesian coordinates into the curvilinear (cylindrical or spherical) coordinates.

If (x_l, y_l, z_l) denotes $\mathbf{p}$'s local Cartesian coordinates, then its cylindrical and spherical coordinate conversions are:

$$r = \sqrt{x_l^2 + y_l^2} \tag{101}$$

$$\theta = \tan^{-1}\frac{y_l}{x_l} \tag{102}$$

$$z = z_l \tag{103}$$

and

$$r = \sqrt{x_l^2 + y_l^2 + z_l^2} \tag{104}$$

$$\theta = \cos^{-1}\frac{z_l}{r} \tag{105}$$

$$\phi = \tan^{-1}\frac{y_l}{x_l} \tag{106}$$

respectively. For spherical coordinate systems, STAR-CCM+ follows the physics convention of using θ to denote the polar angle – that is, the angle from the z-axis.

However, the transformation of a vector or tensor to a local curvilinear coordinate system involves a change of basis which depends on the location within the local Cartesian frame. The transformation process in this case is to:

1. Compute the position of the vector or tensor with respect to the local Cartesian frame.
2. Perform a change of basis using the basis of the curvilinear coordinate system at that point.

The following examples represent this transformation for cylindrical and spherical coordinates.

Cylindrical Coordinates

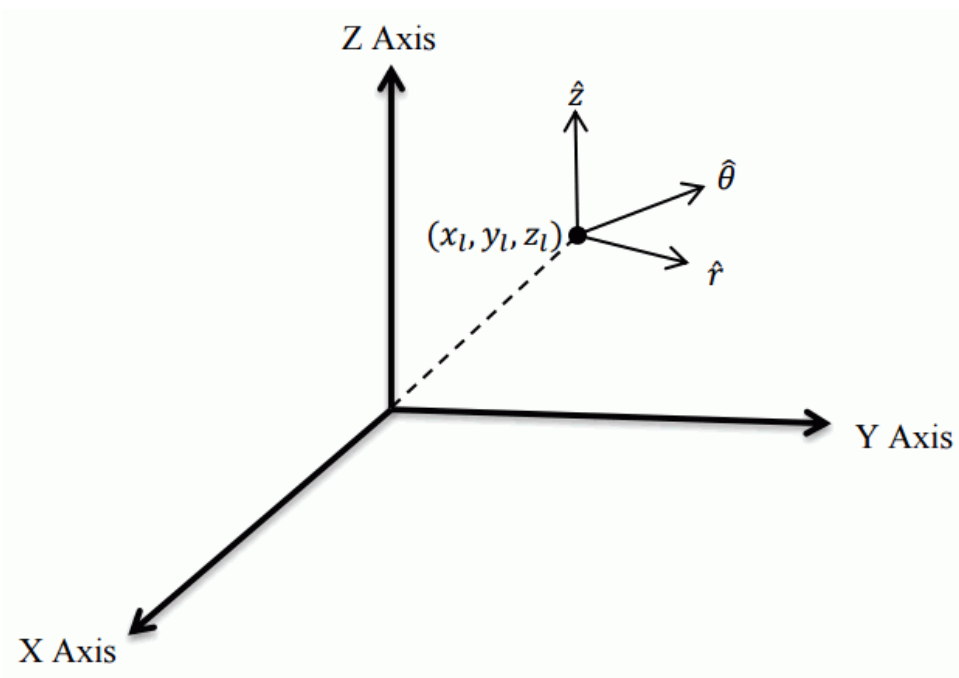
The following coordinates give the cylindrical basis vectors at a local coordinate, (x_l, y_l, z_l):

$$\hat{r} = \frac{x_l\hat{i} + y_l\hat{j}}{\sqrt{x_l^2 + y_l^2}} \tag{107}$$

$$\hat{\theta} = \hat{k} \times \hat{r} \tag{108}$$

$$\hat{z} = \hat{k} \tag{109}$$

These vectors are illustrated here:



Spherical Coordinates

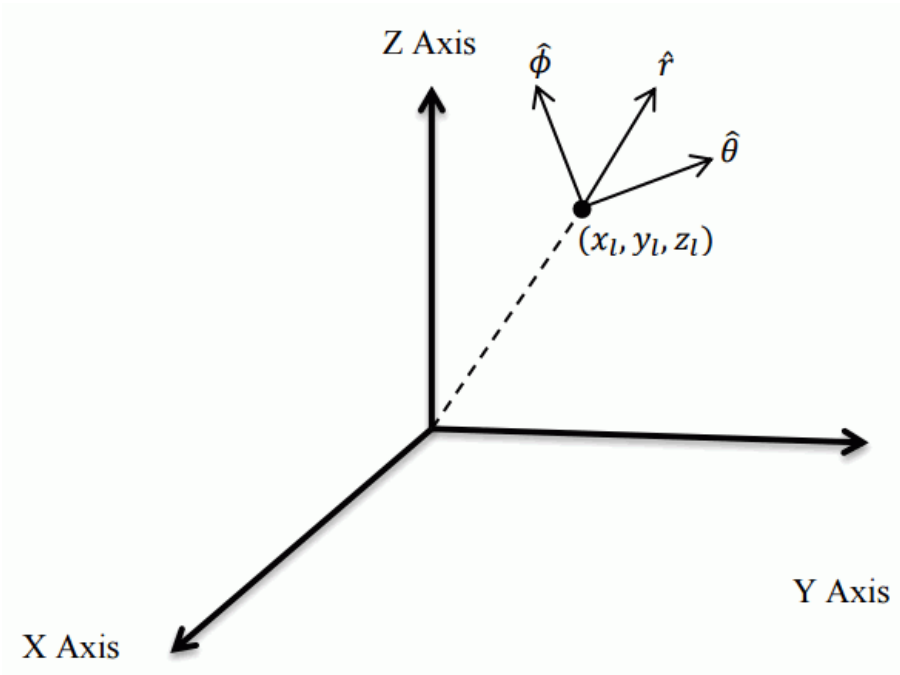
The following coordinates give the spherical basis vectors at a local coordinate, (x_l, y_l, z_l):

$$\hat{r} = \frac{x_l\hat{i} + y_l\hat{j} + z_l\hat{k}}{\sqrt{x_l^2 + y_l^2 + z_l^2}} \tag{110}$$

$$\hat{\theta} = \frac{\hat{k} \times \hat{r}}{\|\hat{k} \times \hat{r}\|} \tag{111}$$

$$\hat{\phi} = \hat{r} \times \hat{\theta} \tag{112}$$

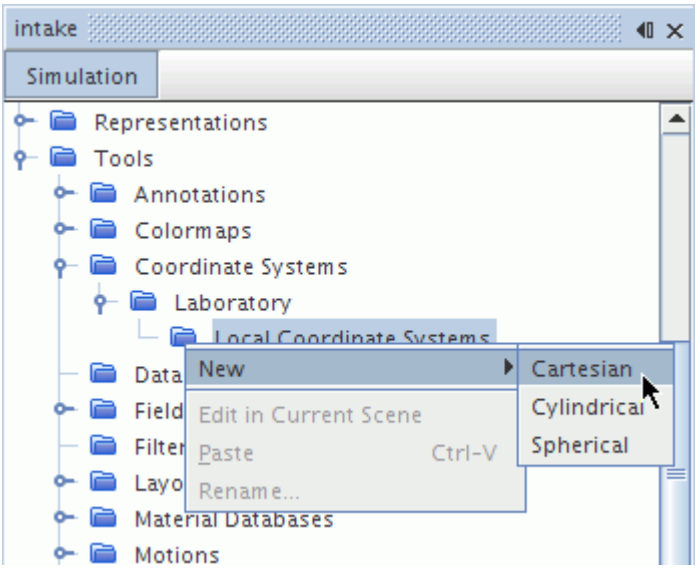
These vectors are illustrated here:



Creating a Local Coordinate System

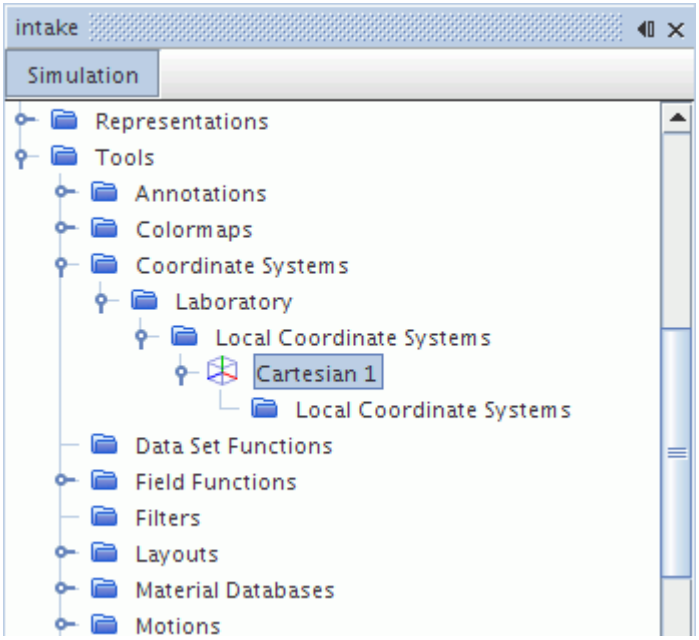
To create a new local coordinate system:

1. Right-click on the **Local Coordinate Systems** manager node of either the base laboratory system, or a previously defined Cartesian local coordinate system and choose **New** followed by the type of local coordinate system you wish to create:
   - **Cartesian**
   - **Cylindrical**
   - **Spherical**



If you have a scene display open, you are able to create the system interactively. See Creating a Local Coordinate System Interactively.

Otherwise the new coordinate system appears in the **Local Coordinate Systems** manager node belonging to the parent system.

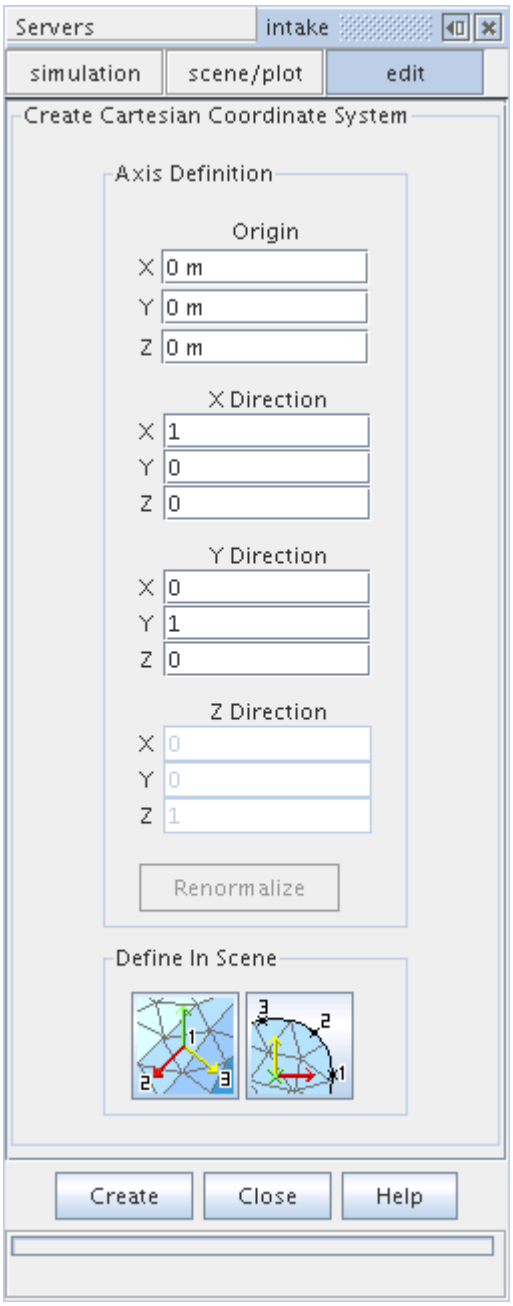


If you want to create a duplicate of an existing Cartesian coordinate system, you can do it using the technique for copying properties. See Copying Properties of Local Coordinate Systems.

You can rename the coordinate system and set its axis and origin properties. You can also edit it interactively in a scene.

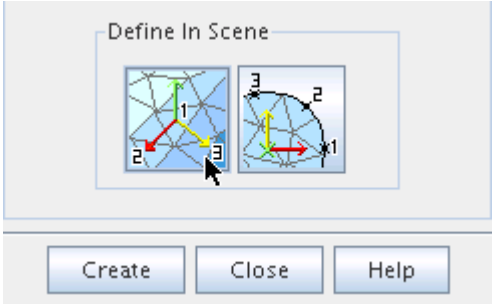
Creating a Local Coordinate System Interactively

When you have a scene display active, visual tools are activated when you choose to create a new local coordinate system. An in-place dialog appears in the *edit* tab of the simulation window.



The text boxes allow you to specify coordinates as text entries. You can enter strings with mixed units (see Entering Mixed Units). You can also set vector coordinates with a single copy-and-paste action (see Pasting Coordinate Sets Directly into Dialog Fields). To apply your entries, click the **Renormalize** button. When you click this button, STAR-CCM+ makes sure that the vectors are normalized and orthogonal, which typically results in some of the entered values changing.

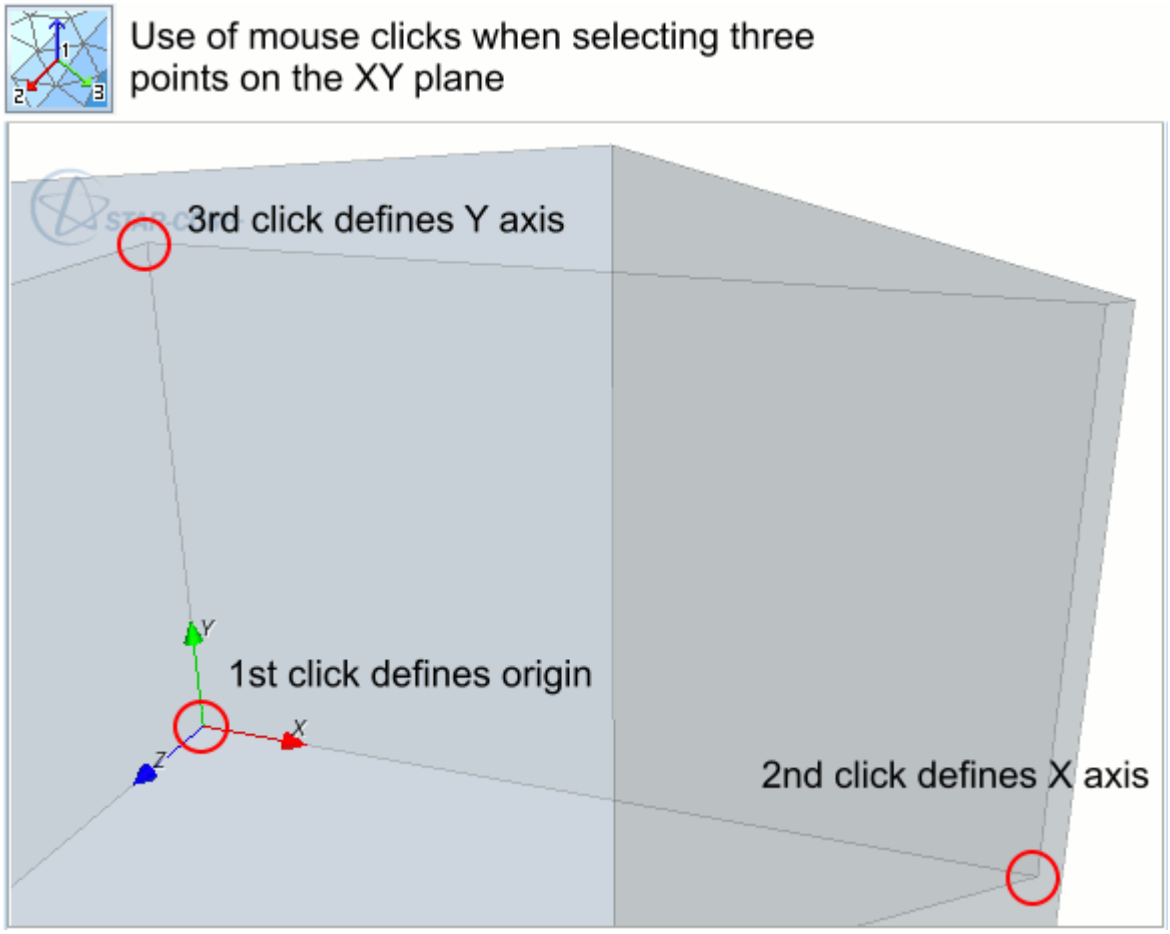
Alternatively, you can click one of two mouse icons in the dialog.



The icon on the left activates the tool for specifying three points that define the XY plane.

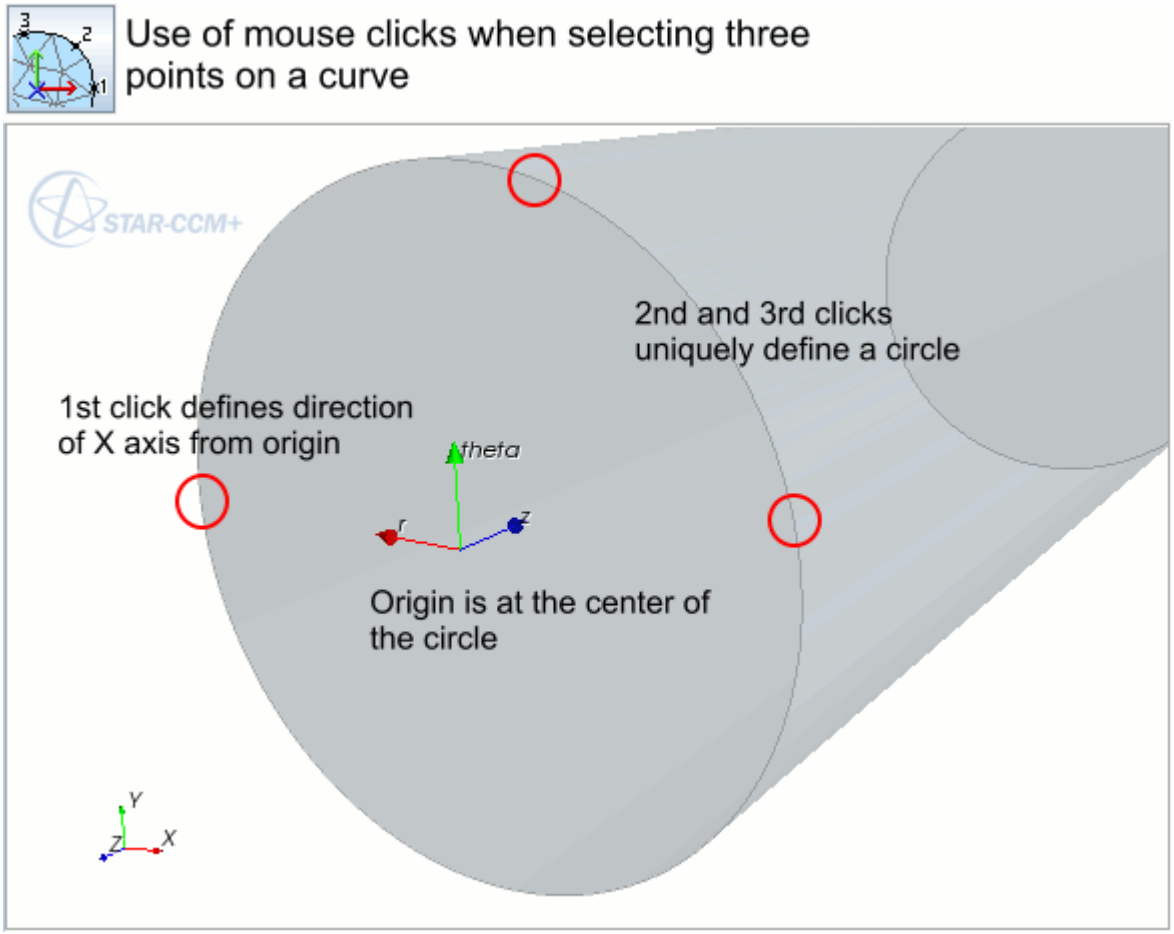
1. The first click defines the origin of the coordinate system.
2. The vector from the first click to the second click defines the X axis.
3. The third click defines the direction of the Y axis. The actual Y axis needs to be perpendicular to the X axis.

The Z axis is defined as orthogonal to the X and Y axes. These mouse clicks are illustrated below.



The icon on the right activates the tool for specifying three points on a circle. You should click three points that are not co-linear, and that uniquely define a circle. The origin of the new coordinate system is at the center of

this circle. The X axis points towards the first click, and the Z axis is perpendicular to the plane of the circle with its direction determined by the order in which the points are defined. In the example that is shown below, the three clicks were made clockwise beginning from the first click.



You can adjust the display while this tool is active.

When you finish setting the parameters, click the **Renormalize** button to apply your entries. Click `Create` to add the coordinate system to the simulation tree, then click `Close`. Alternatively, click `Close` to cancel the creation process and discard all your settings.

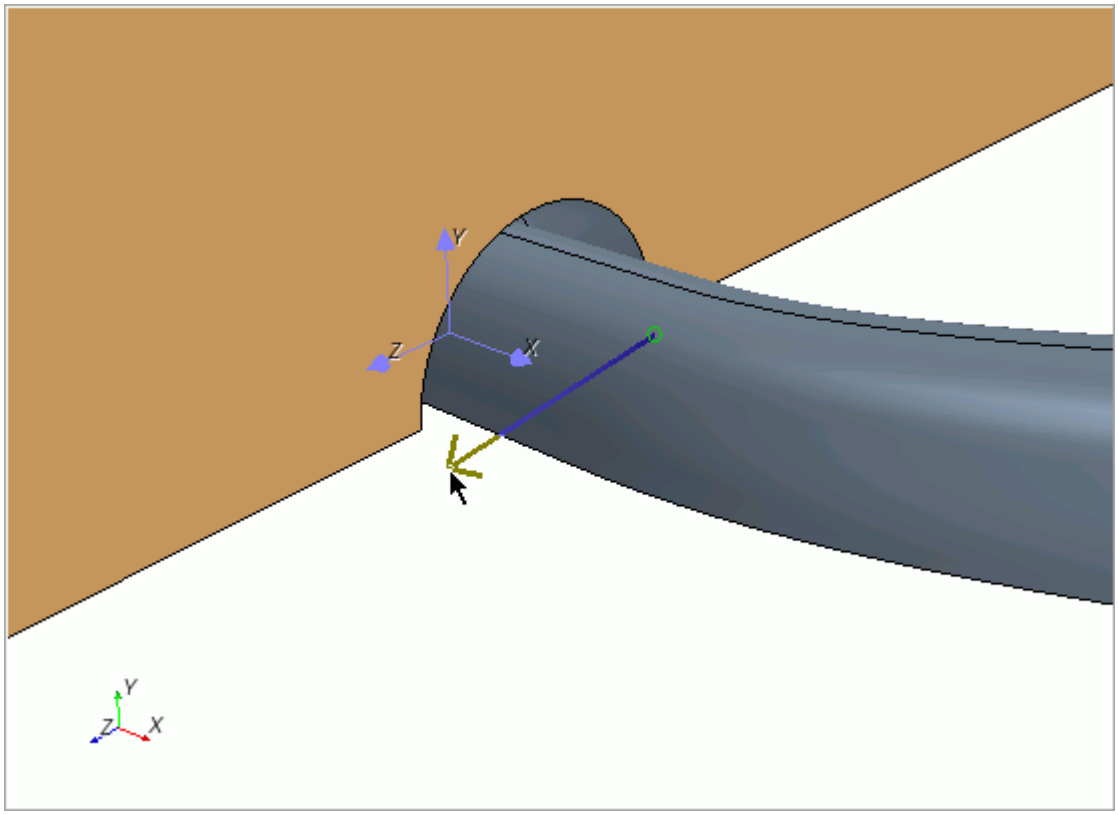
Contents:

Adjusting the Display While Creating a Coordinate System

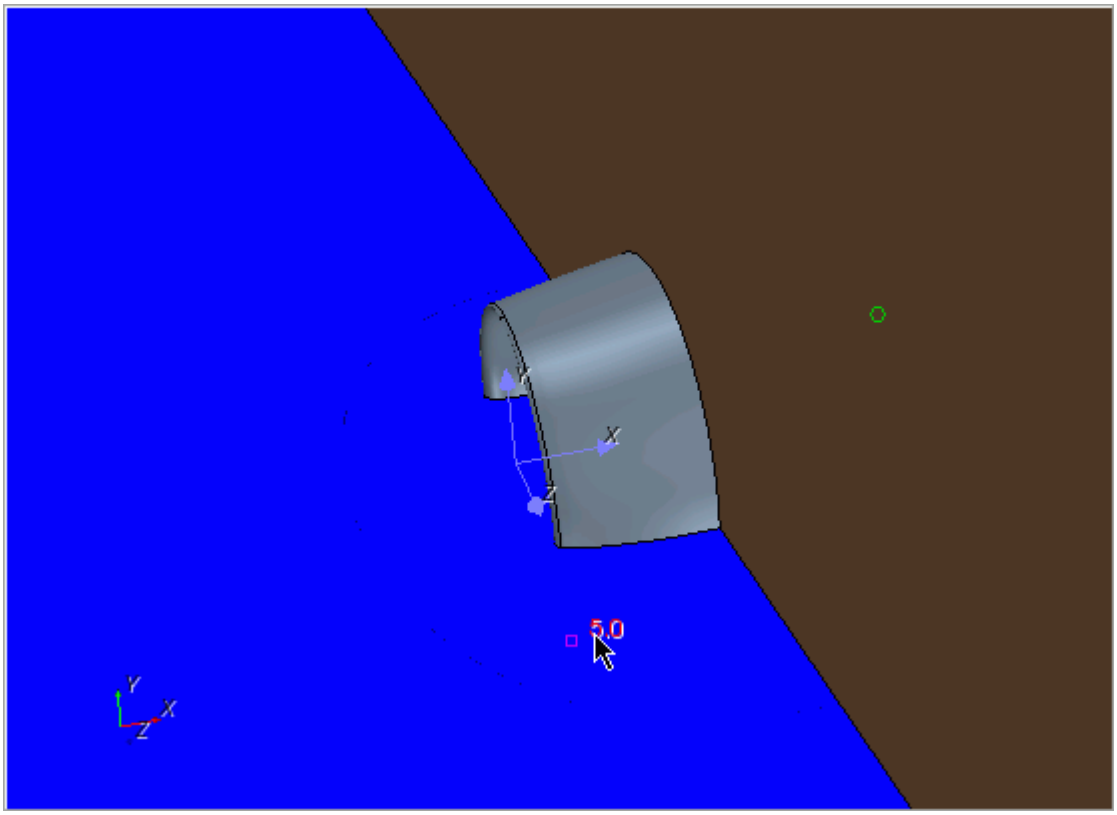
Adjusting the Display While Creating a Coordinate System

To change the view in the scene display while the coordinate system creation tool is active, use normal mouse movements for changing a 3D display or changing a 2D display.

1. For example, select the first of three points on the XY plane.

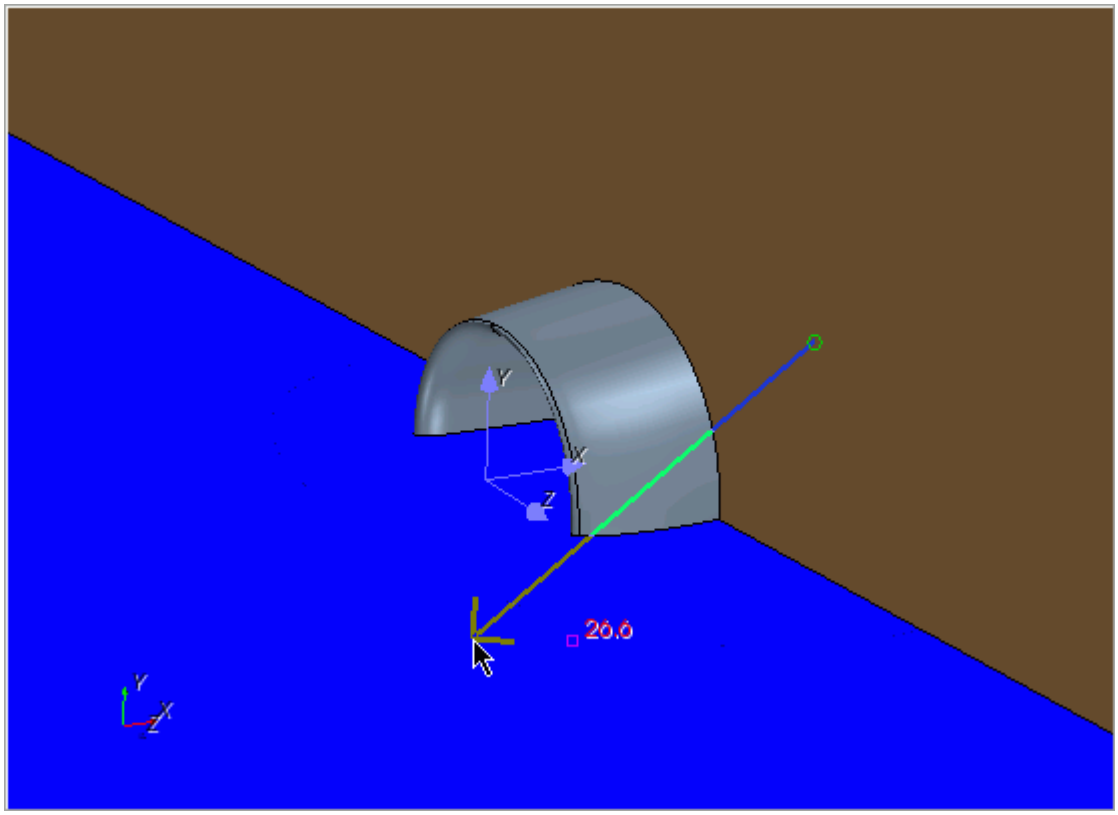


2. To get access to an area that is behind the view, move the view as needed:
   - To rotate, click and hold the mouse button, then drag.
   - To zoom, use the standard techniques with the middle mouse button.



3. After changing the view as desired, release the mouse button.

   The vector of the creation tool reappears.

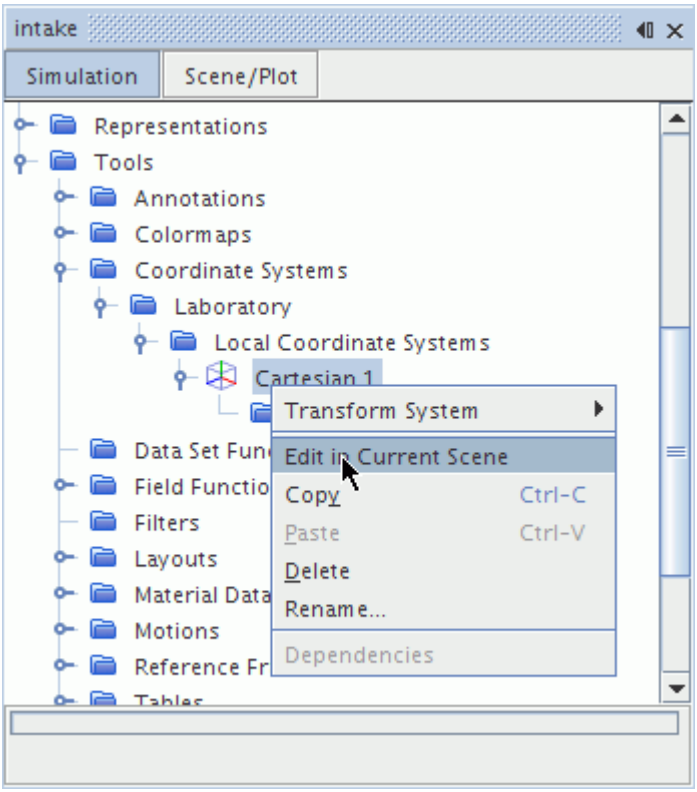


4. Continue defining the coordinate system, or make other adjustments to the view.

Editing a Local Coordinate System Interactively

To edit a local coordinate system interactively:

1. If you have a scene display active, right-clicking on it and select **Edit in Current Scene**.



This pop-up menu item is disabled (grayed out) if a scene display is not active.

2. Modify the local coordinate system.

   The subsequent steps for modifying a local coordinate system in a scene are the same as those used for creating one. See Creating a Local Coordinate System Interactively

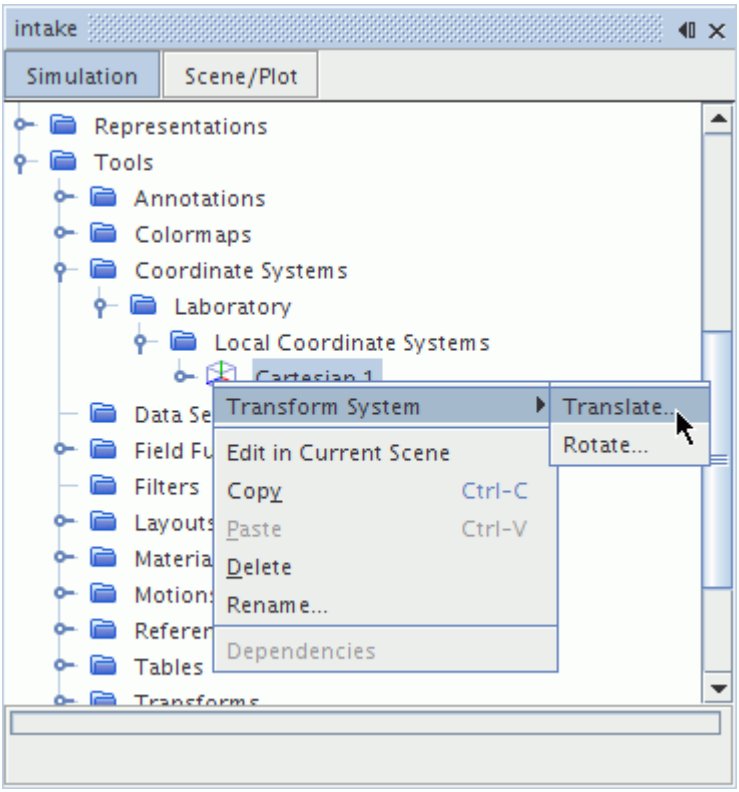
Transforming a Local Coordinate System

STAR-CCM+ allows you to translate or rotate a local coordinate system to a new location. You access these transformations through a right-click menu on the local coordinate system node.

Translate

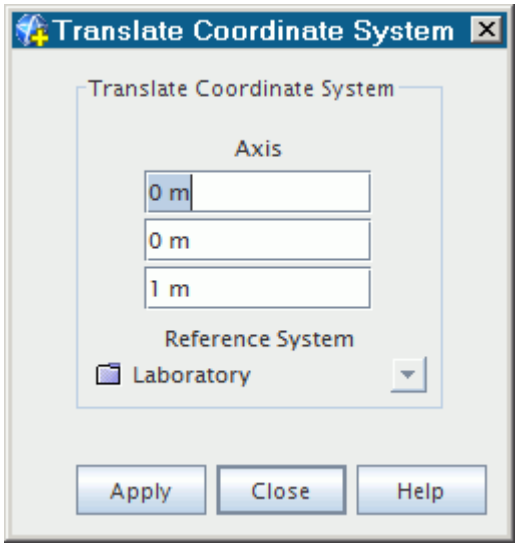
To translate the coordinate system:

1. Right-click on the node and select **Transform System** > **Translate...** in the pop-up menu.



2. In the *Translate Coordinate System* dialog, enter the component distances by which you wish to translate the local coordinate system.

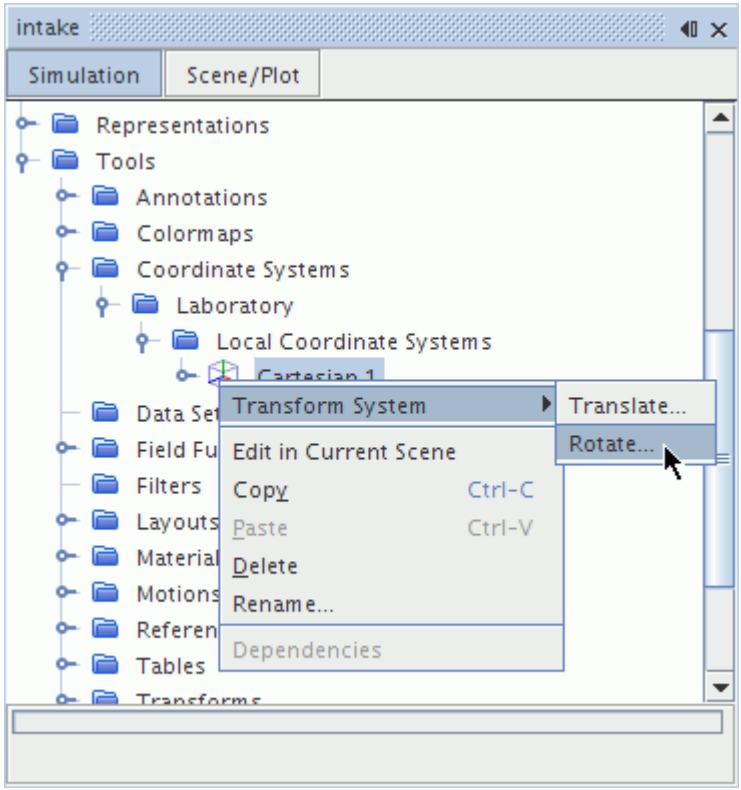
   Distances can be either positive or negative, and can be entered in any valid length unit.
3. Click the Reference System drop-down list for a choice of existing coordinate systems.
4. Click **Apply** to complete your changes. Click **Close** to exit the dialog.



Rotate

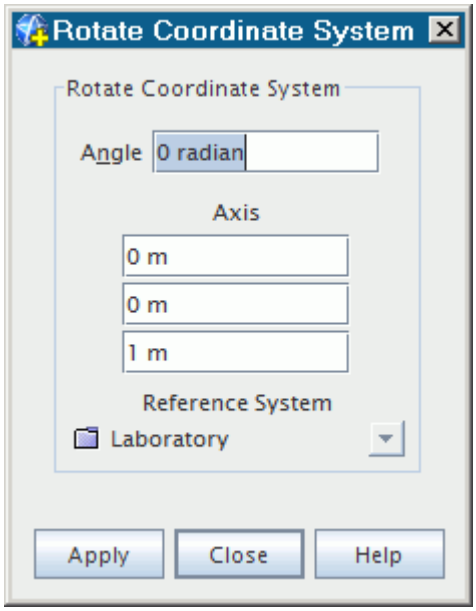
To rotate the coordinate system:

1. Right-click on the node and select **Transform System** > **Rotate...** in the pop-up menu.



2. In the *Rotate Coordinate System* dialog, enter the angle by which you wish to rotate the local coordinate system, and the components of the vector to rotate it around.

   The *Reference System* drop-down list offers a choice of coordinate systems in the simulation.
3. Click **Apply** to complete your changes.
4. Click **Close** to exit the dialog.



Copying Properties of Local Coordinate Systems

When properties are copied from one local coordinate system to another, all of the editable properties are applied. The read-only *Reference System* remains the same.

It is not possible to copy properties between a user-defined coordinate system and the laboratory coordinate system, which is pre-defined.

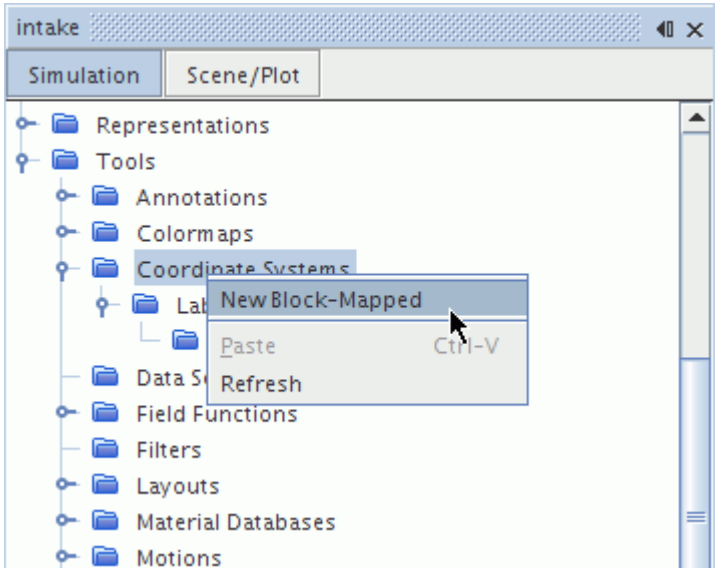
You can use these copying techniques to create a duplicate of a local coordinate system. Simply apply the **Paste** command to the **Coordinate Systems** manager node (or make it the destination of your drag-and-drop operation).

A copy of the coordinate system appears in the tree.

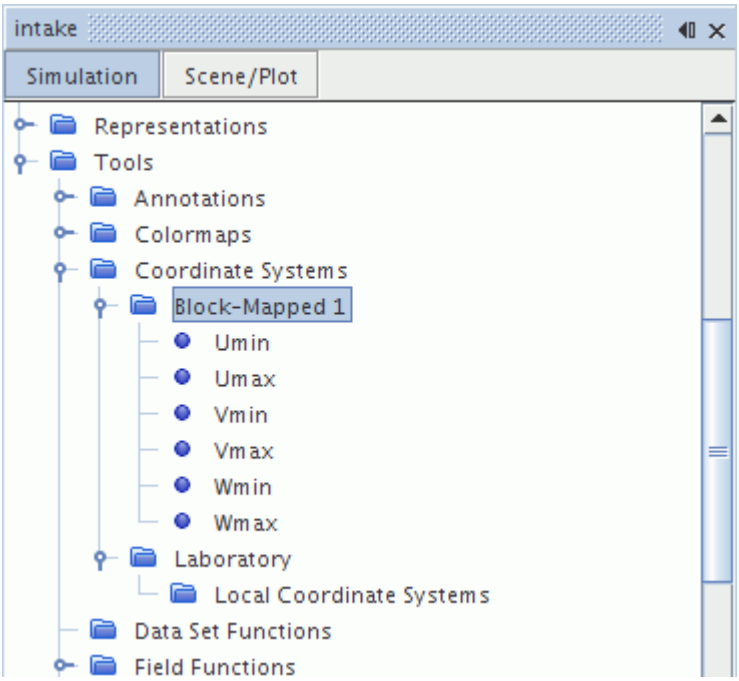
Creating a Block-Mapped Coordinate System

To create a new block-mapped coordinate system:

1. Expand the **Tools** node, right-click on the **Coordinate Systems** node, and select **New Block-mapped**.



The new coordinate system appears in the **Coordinate Systems** node. It contains six subnodes representing the six sides of the model.



You can then rename the coordinate system and set its properties.

Contents:

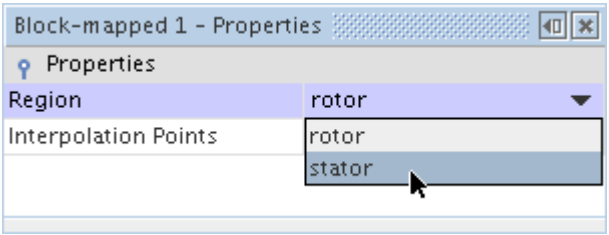
Setting Properties of a Block-Mapped Coordinate System
Working with Field Functions of the Block-Mapped Coordinate System

Setting Properties of a Block-Mapped Coordinate System

Specify a region from which the boundaries are taken.

To do this:

1. make sure that you have the node of the coordinate system selected, and then choose the region from the drop-down list of the *Region* property.

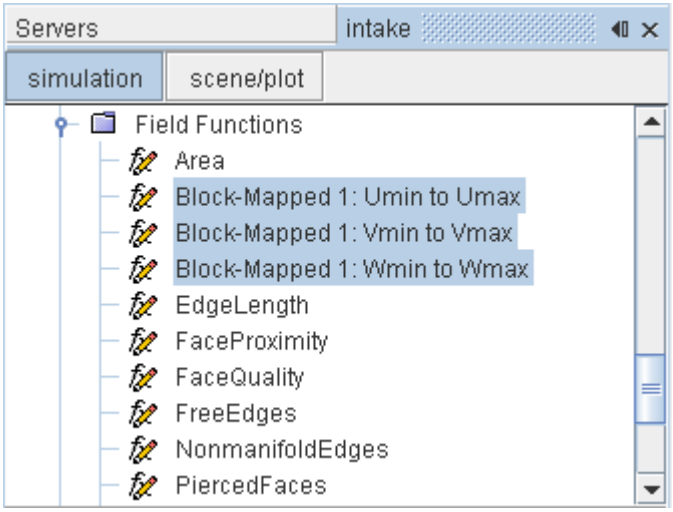


2. Set up the boundary groups by selecting each of the block-mapped boundaries nodes and clicking the right half of its *Boundaries* property.
3. Then specify the boundaries in the object selector.

Working with Field Functions of the Block-Mapped Coordinate System

Three field functions are used with the block-mapped coordinate system.

When they are calculated, their nodes are added under the **Tools** > **Field Functions** manager node: **Block-mapped: Umin - Umax**, **Block-mapped: Vmin - Vmax**, and **Block-mapped: Wmin - Wmax**. The names of these field functions can be altered by renaming the coordinate system and/or any of its six subnodes.



The values in the field functions are calculated when they are needed. A section surface can be defined by creating an iso surface of one of these field functions.

The field functions' values vary from about 0 to 1, where 0 is the first edge of a boundary group pair and 1 is the other edge. The maximum and minimum are not exactly 0.0 and 1.0 due to approximation errors.

The field functions are computed by creating a background grid and interpolating where each point in the actual mesh would be located. If this background mesh is too coarse, the resulting section surfaces will not be smooth. In this case, increase the resolution of the background grid by changing the coordinate system's *Interpolation Points* property.

Displaying a Coordinate System in a Scene

When a coordinate system is in a scene, it functions as an alternate set of axes. The original set of axes is itself a laboratory coordinate system, though it is relative to the screen rather than to the origin in the space being modeled.

You can display a coordinate system in a scene using one of two techniques: drag-and-drop or object selection.

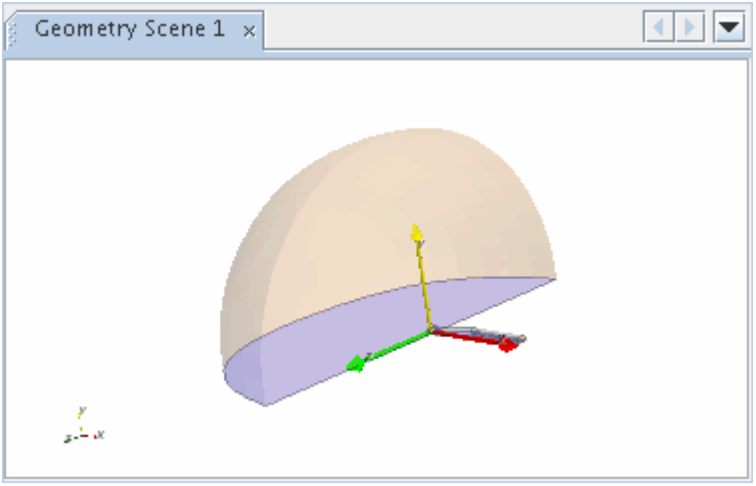
Displaying a Coordinate System Using Drag-and-Drop

To display a coordinate system in a scene with the drag-and-drop technique;

- Drag the node of the particular coordinate system onto the scene display, and release the mouse button when your pointer is within the display.

  If you do not have the scene open, you can also drag the coordinate system node onto the scene node.

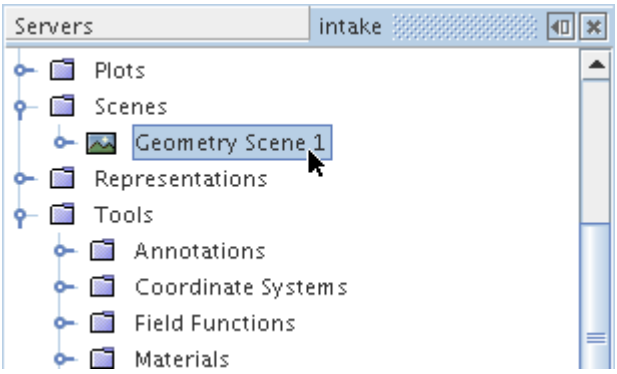
The axes of the coordinate system appear in the display. In the following screenshot, a Cartesian coordinate system has been added to the scene.



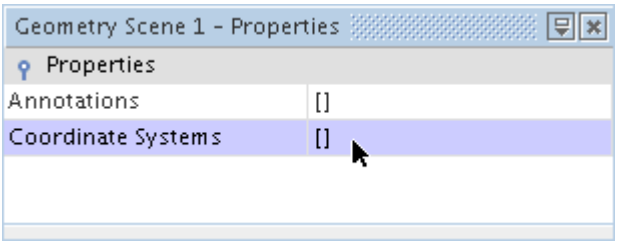
Displaying a Coordinate System Using Object Selection

To display a coordinate system in a scene by selecting it as an object:

1. Select the node of the particular scene.

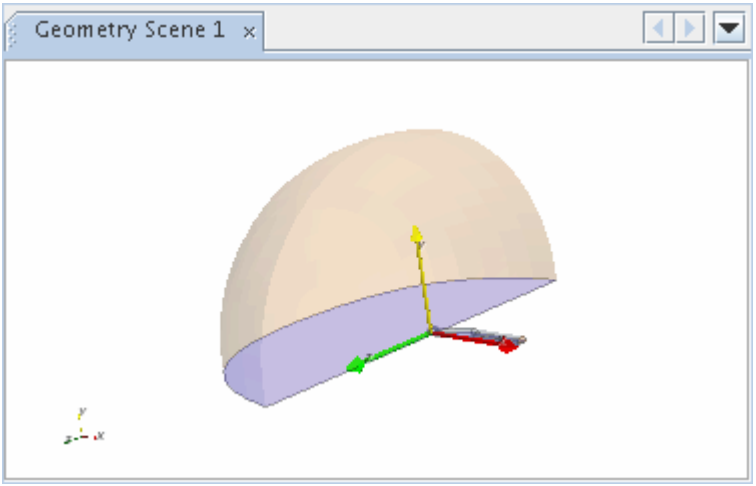


2. In the *Properties* window, click in the right half of the *Coordinate Systems* property.



This activates a standard object selection dialog for selecting coordinate systems for the scene.

The axes of the coordinate system appears in the display. In the following screenshot, a Cartesian coordinate system has been added to the scene.



Coordinate Systems Reference

Coordinate Systems Manager Right-Click Actions

New Block-Mapped	Executes the create block-mapped coordinate system action to add a new block-mapped coordinate system to the **Coordinate Systems** node. See Block-Mapped Coordinate Systems.

Local Coordinate Systems Reference

Local Coordinate System Right-Click Actions

Transform System	Transforms the coordinate system in a manner selected in the submenu.	
	Translate	Activates a dialog for translating the coordinate system.
	Rotate	Activates a dialog for rotating the coordinate system.

Cartesian and Body-Fitted Local Coordinate Systems Properties

X Axis Direction	A Cartesian unit vector that, along with the *Origin*, defines the x-axis.
Y Axis Direction	A Cartesian unit vector that, along with the *Origin*, defines the y-axis.
Z Axis Direction	A Cartesian unit vector that, along with the *Origin*, defines the z-axis (read-only).
Origin	The Cartesian position vector defining the origin of the x-, y-, and z-axes.
Reference System	The system to which the current coordinate system refers (read-only).

Cylindrical Coordinate System Properties

Radial Axis Direction	A Cartesian unit vector that, along with the *Origin*, defines the radial axis.
Tangential Axis Direction	A Cartesian unit vector that, along with the *Origin*, defines the tangential axis.
Axial Axis Direction	A Cartesian unit vector that, along with the *Origin*, defines the axial axis (read-only).
Origin	The Cartesian position vector defining the origin of the radial, tangential, and axial axes.
Reference System	The system to which the current coordinate system refers (read-only).

Spherical Coordinate System Properties

Radial Axis Direction	A Cartesian unit vector that, along with the *Origin*, defines the radial axis.
Polar Axis Direction	A Cartesian unit vector that, along with the *Origin*, defines the polar axis.
Azimuthal Axis Direction	A Cartesian unit vector that, along with the *Origin*, defines the azimuthal axis (read-only).
Origin	The Cartesian position vector defining the origin of the radial, polar, and azimuthal axes.
Reference System	The system to which the current coordinate system refers (read-only).

Block-Mapped Coordinate System Reference

Region	A drop-down list of the regions in the simulation, from which you select a region to specify boundary groups.
Interpolation Points	A set of points along the three dimensions, representing the number of cells in the background grid along the Umin/Umax, Vmin/Vmax, and Wmin/Wmax directions. Increasing the number of points improves the accuracy with which the coordinate system follows the contours of the mapped surfaces.

Modeling Time

The primary function of the Time models in STAR-CCM+is to provide solvers that control the iteration and/or unsteady time-stepping.

This section contains the following:

Contents:

What Is the Steady Model?
What Is the Harmonic Balance Model?
What is the PISO Unsteady Model?
What Is the Implicit Unsteady Model?
What Is the Explicit Unsteady Model?
Selecting a Time Model
Setting High-Accuracy Temporal Discretization
Setting the Time-Step for Implicit Unsteady Simulations
How Do I Choose Between Steady and Unsteady?
How Do I Choose a Time-Step?

What Is the Steady Model?

The Steady model is used for all steady-state calculations. When this model is activated, the concept of a physical time-step is meaningless.

For those objects that offer a choice between **Iteration** or **Time-Step** in a trigger (such as Monitors or Scenes), choose only **Iteration**.

What Is the Harmonic Balance Model?

The Harmonic Balance time model is always used with the Harmonic Balance method which is designed for solving periodic unsteady problems, where the solution repeats itself with a known frequency.

See Modeling with Harmonic Balance.

See Also:

What Is the Harmonic Balance Flow and Energy Model?

What is the PISO Unsteady Model?

The PISO Unsteady time model is for use with the PISO algorithm, in transient cases.

The PISO Unsteady time model applies the PISO algorithm when solving the discretized system of equations for a fluid continuum. For each time interval, PISO performs a predictor step followed by a number of corrector steps. During these steps, linear equation sets are solved iteratively for each main dependent variable. STAR-CCM+ decides the number of corrector steps based on a splitting error and inner residual levels.

See PISO Algorithm.

Contents:

The PISO Unsteady Solver

The PISO Unsteady Solver

The PISO Unsteady solver controls the solution update for the Segregated Flow model according to the PISO algorithm.

The PISO Unsteady solver is available when the PISO Unsteady time model is active. Use the PISO algorithm in transient cases with small time-steps.

PISO Unsteady Solver Properties

Time-Step

The physical time-step. In addition to a constant, you can enter an expression directly.

Verbosity

Specifies the amount of feedback (directed to the *Output* window) by the PISO solver.

None

Provides no feedback. The PISO solver runs silently.

Low

Provides the number of PISO iterations taken to converge

High

Provides the same as Low, plus the residuals of the individual PISO iterations.

Diagnostics

Provides the highest degree of information. This option can affect performance, and therefore is not recommended for normal iteration.

Solver Frozen

When **On**, prevents the solver from updating the solution while iterating or responding to other solvers. This feature is primarily a “debugging” tool and is not supported for normal simulation work.

What Is the Implicit Unsteady Model?

The Implicit Unsteady model is the only unsteady model available with the Segregated Flow and Segregated Fluid Energy models. It uses the Implicit Unsteady solver.

With the Coupled Flow and Coupled Energy models, the implicit unsteady approach is the alternative to the Explicit Unsteady one. The choice between these two approaches is based on the time scales of the phenomena of interest.

When this model is activated, objects that offer a choice between *Iteration* or *Time-Step* for a trigger (such as Monitors or Scenes), can be set to update at each time-step.

Contents:

What Is the Implicit Unsteady Solver?

See Also:

Running a Transient Analysis

What Is the Implicit Unsteady Solver?

The Implicit Unsteady solver becomes active when the Implicit Unsteady model is active. Its primary function is to control the update at each physical time for the calculation. It also controls the time-step size.

Properties

Time-Step	The physical time-step. In addition to a constant, you can enter an expression directly. The default Time-Step is 0.001 s.	
Temporal Discretization	Selects the temporal discretization scheme. For guidance on scheme selection, see the discussion on the transient term.	
	1st-order	Selects the first-order temporal scheme. When this scheme is selected together with a rigid body, morphing, or user-defined vertex motion model, the discretized grid flux is of first order (see Eqn. (145)).
	2nd-order	Selects the second-order temporal scheme. When this scheme is selected together with a rigid body, morphing, or user-defined vertex motion model, the discretized grid flux is of second order (see Eqn. (146)).

Expert Properties

Freeze Time	Controls whether to advance time or fix it at the current time-step.	
	Activated	Time does not advance.
	Deactivated	Time advances. This option is the default setting.
Solver Frozen	Prevents or allows the solver from updating the solution while iterating. Activating this option is primarily a debugging tool that is not normally required for simulation work.	
	Activated	The solver does not update the solution.
	Deactivated	The solver updates the solution.

What Is the Explicit Unsteady Model?

The Explicit Unsteady model is available only with the Coupled Energy model. It is only compatible with the Inviscid and Laminar viscous regime models. It uses the Explicit Unsteady solver.

The explicit approach is the alternative to the Implicit Unsteady model. The choice between these two approaches is based on the time scales of the phenomena of interest.

When this model is activated, objects that offer a choice between *Iteration* or *Time-Step* for a trigger (such as Monitors or Scenes), can be set to update at each time-step.

Contents:

What Is the Explicit Unsteady Solver?

See Also:

Running a Transient Analysis

What Is the Explicit Unsteady Solver?

The Explicit Unsteady solver becomes active when the Explicit Unsteady model is active. Its primary function is to control the iterative update for the calculation.

Properties

None.

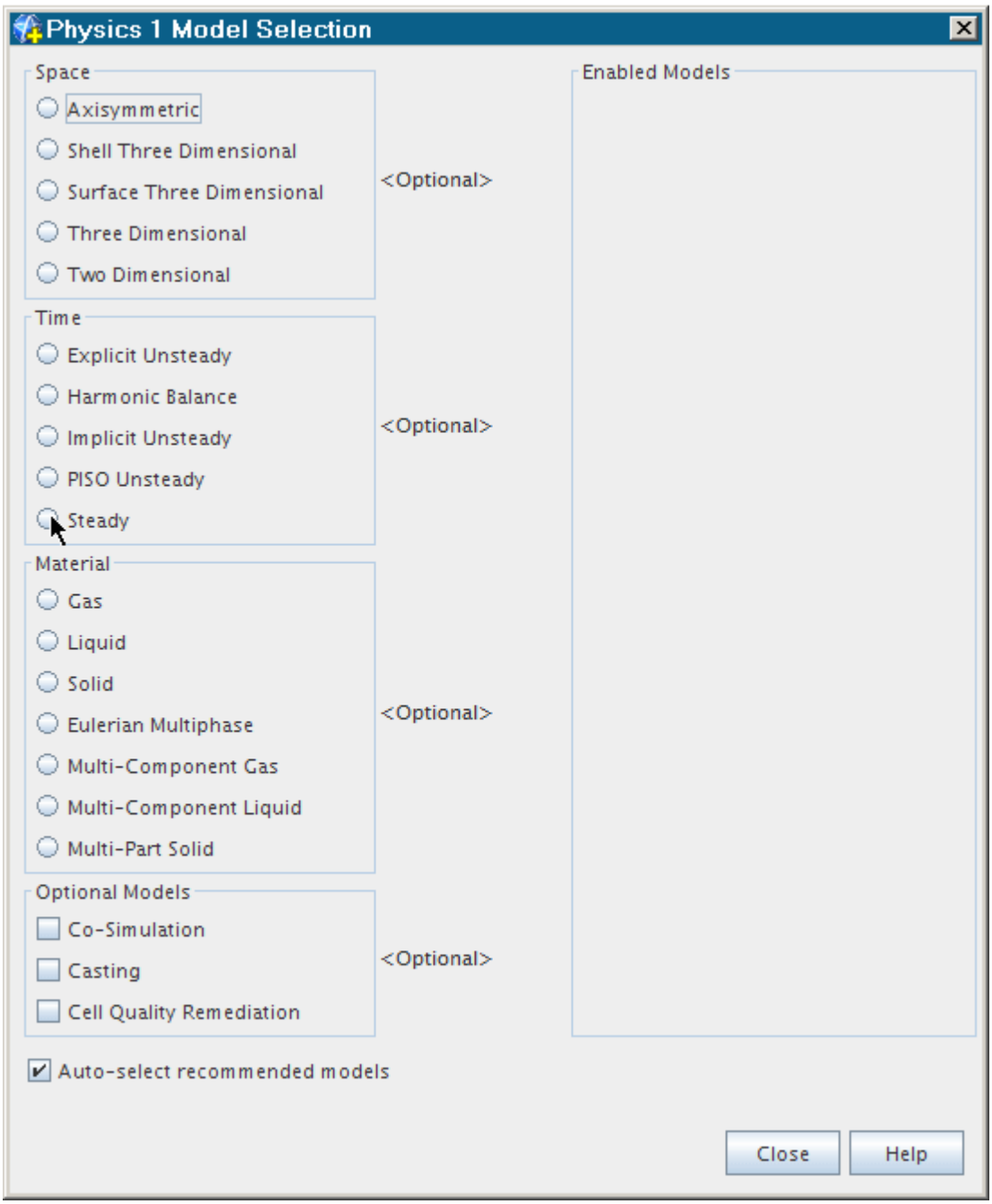
Expert Properties

Solver Frozen	Prevents or allows the solver from updating the solution while iterating. Activating this option is primarily a debugging tool that is not normally required for simulation work.	

	Activated	The solver does not update the solution.
	Deactivated	The solver updates the solution.

Selecting a Time Model

Before or after selecting the Equation of State model and the Viscous Regime, select the relevant Time model radio button.



Setting High-Accuracy Temporal Discretization

High-accuracy temporal discretization provides the option of doing 2nd-order time discretization with four or five time levels.

High-accuracy discretization schemes give faster unsteady solutions by using larger time steps and are more accurate. However, they are harder to stabilize and require more attention to mesh quality. Start with lower-order schemes and switch to high-accuracy schemes, if necessary, when all strong initial transients have been eliminated.

To use high-accuracy temporal discretization, select the **Implicit Unsteady** solver node and set *Temporal Discretization* to **2nd-order**. The **High-Accuracy Temporal Discretization** node becomes available under the following solver nodes:

- Coupled Implicit
- Segregated Flow
- Segregated Energy, when *Enable High-Accuracy Temporal Discretization* is also activated.
- Segregated Species, when *Enable High-Accuracy Temporal Discretization* is also activated.

High-Accuracy Temporal Discretization Properties

Option

Specifies the number of time levels that are used for second-order temporal discretization:

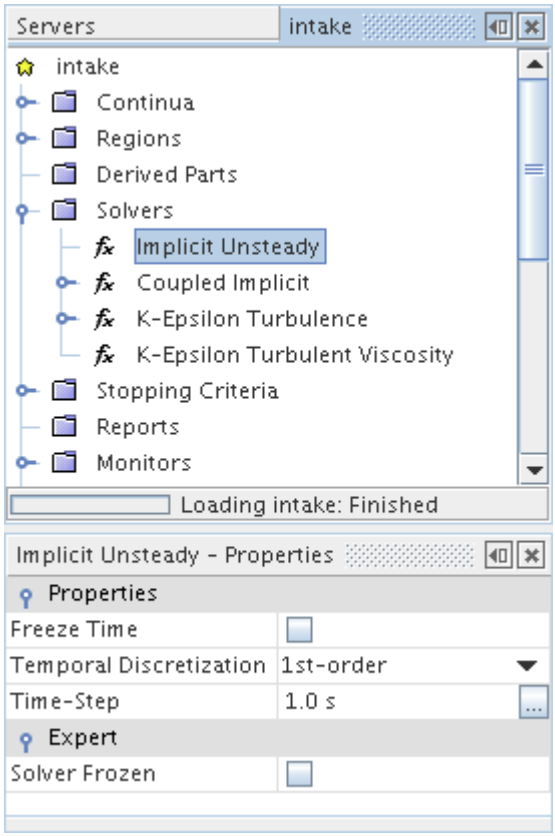
- **NONE**: Adds no time steps to 2nd-order temporal discretization. This option leaves the number of time levels at three. This is the default.
- **Optimized 2nd-order (4)**: Sets the number of time levels at four. Reduces leading-order truncation error by a factor of 2 compared to base-level 2nd-order temporal discretization.
- **Optimized 2nd-order (5)**: Sets the number of time levels at five. Reduces leading-order truncation error by a factor of 2.64 compared to base-level 2nd-order temporal discretization.

Setting the Time-Step for Unsteady Simulations

To set the time-step for unsteady simulations, enter a constant or an expression in the related solver or select one or both time-step control models.

Use one of the following methods:

- To set the time-step to a constant, select the **Implicit Unsteady** or **PISO Unsteady** solver under the **Solvers** node and set the *Time-Step* property in *Properties* window.



- To set the time-step to an expression, such as `10 * sin(3.14 * $Time)`, enter the expression in the same *Properties* window.
- To set the time-step automatically, based on CFL number, the convective time scale, select the Convective Time-Step Control model and set the properties of the related subsolver. The model adjusts the time-step so that the actual CFL number approaches either a maximum or a mean CFL number, whichever yields the minimum time-step.
- To set the time-step automatically, based on von Neumann number, $\mu \Delta t/(\Delta x)^2$ where μ is thermal diffusivity, select the Energy-Based Time-Step Control model and set the properties of the related subsolver. The model adjusts the time-step so that the actual von Neumann number approaches either a maximum or a mean von Neumann number, whichever yields the minimum time-step.

The Time-Step Control models can be used separately or together. They are suitable when time scales vary over the course of the simulation.

Contents:

Time-Step Control Models

Time-Step Control Models

The Convective CFL and Energy-Based Time-Step Control models automatically adjust the length of time-step according to the CFL and von Neumann numbers you specify.

The Time Step Control models can be used separately or together. At any given point, the simulation uses the shorter of the time-steps from the two models.

Convective CFL Time-Step Control Reference

Provided By	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Convective CFL Time-Step Control**	
Requires	*Time*: **Implicit Unsteady** or **PISO Unsteady** *Material*: one of **Gas**, **Liquid**, **Multi-Component Gas**, or **Multi-Component Liquid** *Reaction Regime*: **Non-reacting** (if **Multi-Component Gas** or **Multi-Component Liquid** is selected) *Flow*: any	
Activates	Solvers	**Time-Step Control** > **Time-Step Control Sub Solvers** > **Convective CFL Time-Step Control**

Energy-Based Time-Step Control Reference

Provided By	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Energy-Based Time-Step Control**	
Requires	*Time*: **Implicit Unsteady** or **PISO Unsteady** *Material*: one of **Gas**, **Liquid**, **Multi-Component Gas**, or **Multi-Component Liquid** *Reaction Regime*: **Non-reacting** (if **Multi-Component Gas** or **Multi-Component Liquid** is selected) *Flow*: any *Energy*: any	
Activates	Solvers	**Time-Step Control** > **Time-Step Control Sub Solvers** > **Energy-Based Time-Step Control**

The models adjust the time-step so that the actual CFL and von Neumann numbers approaches either specified maxima or specified means, whichever yield the minimum time-step. To remove any of these numbers from consideration, set them to an arbitrarily high value. The time-step of the Implicit Unsteady or PISO Unsteady solver becomes the maximum permitted time-step overall.

Time-Step Control Solver Properties

Min. Time Step

Sets the lower limit on time-step; appears as $\Delta t_{\min}$ in Eqn. (1673).

Max. Time Step Change Factor

Applies a factor f that limits the amount by which the time-step can change in a single increment.

Verbose

When activated, the time step control solver sends additional information on its progress to the *Output* window.

Convective CFL Time-Step Control Sub-Solver Properties

Target Mean CFL Number

Target Max. CFL Number

Parameters CFL_{mean}^{tar} and CFL_{max}^{tar} as in Eqn. (1673). The default values are `0.5` and `5.0`, respectively.

Energy-Based Time-Step Control Sub-Solver Properties

Target Mean Von Neumann Number

Target Max. Von Neumann Number

Parameters VNN_{mean}^{tar} and VNN_{max}^{tar}, when used in Eqn. (1673) is defined in terms of von Neumann instead of CFL. The default values are `0.5` and `5.0`, respectively.

How Do I Choose Between Steady and Unsteady?

In most situations, the physics being simulated indicates the choice between the Steady and Unsteady (Implicit or Explicit) models.

Simulations that are time-dependent, and hence require the Unsteady model include:

- Time-varying boundary conditions
- Sliding or deforming mesh problems
- Free surface (VOF) problems
- Transient heat transfer

Other simulations are separate cases, namely those using harmonic balance.

In some situations, however, unsteadiness in the physical problem is not readily apparent. It can manifest itself in a failure to converge when running with the Steady model. One example is the solution of flow past a cylinder. At certain Reynolds numbers, the flow is unsteady because of vortex shedding, and it is incorrect to use a steady solution approach.

When physical instabilities do exist, the correct way to model them is to perform a time-dependent simulation. Choose an appropriate time-step that is based on a representative Strouhal number. Accumulate time-averaged statistics only after the transient simulation has achieved a periodic state.

Using the Steady model for unsteady physical phenomena is not recommended. Running an unsteady problem with a Steady model is equivalent to using an inaccurate time-step in an unsteady simulation.

How Do I Choose a Time-Step?

The choice of time-step is an engineering judgment (just like grid refinement). Be sure to understand the time scales of the physical phenomenon.

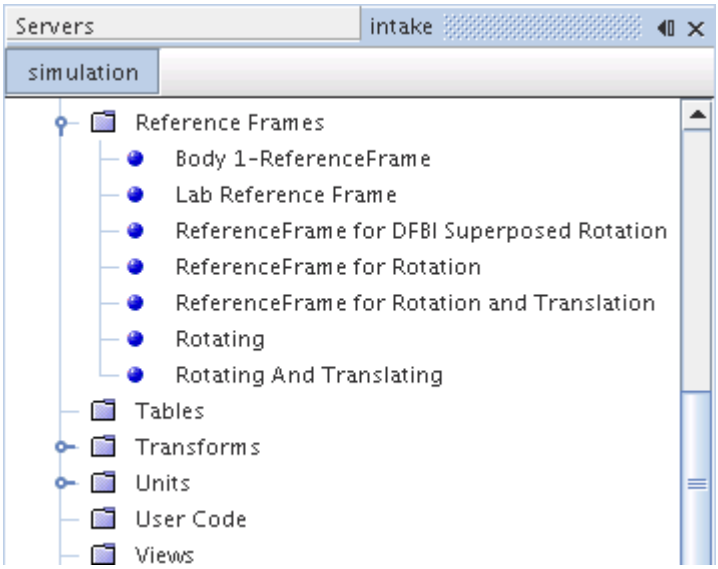
The convective time scale (CFL) is important if, for example, you are tracking the convective dispersion of a scalar through the flow. If molecular diffusion is dominant, consider the Von Neumann number, $\mu \Delta t/(\Delta x)^2$. For Large Eddy Simulation (LES), consider the turbulent time scale. For vortex shedding (for example, with Detached Eddy Simulation, or DES) consider the Strouhal number.

Reference Frames

In STAR-CCM+, it is possible to specify multiple frames of reference for use in applying a constant grid flux to the governing equations of fluid motion, or for post-processing the results of a simulation. Reference frames are also added when particular types of motion are added to the **Motions** manager.

Each reference frame is represented by a node under **Tools** > **Reference Frames**.

The default type is the Laboratory Reference Frame, and the two other general types are Rotating and Rotating and Translating. Depending on how they are added, the latter two types may be either read-only or user-defined, and can vary in their labeling. Only user-defined reference frames can be added or modified.



The following sections describe the various types of reference frames. For more information on all available properties, refer to the section, Reference Frames Reference.

Lab Reference Frame

The **Lab Reference Frame** node is the default stationary reference frame in a STAR-CCM+ simulation.

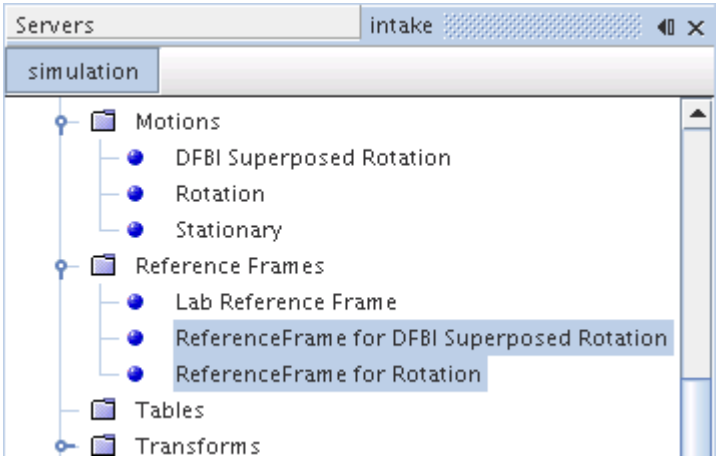
This frame is the one type of reference frame present in the reference frames manager when a simulation is created, and is the default selection for all regions in the simulation. It cannot be deleted, and more reference frames of this type cannot be created.

Read-Only Rotating Reference Frame

A read-only Rotating reference frame is created automatically for each of the following objects that you add to the simulation:

- Rotation motion
- DFBI Superposed Rotation motion

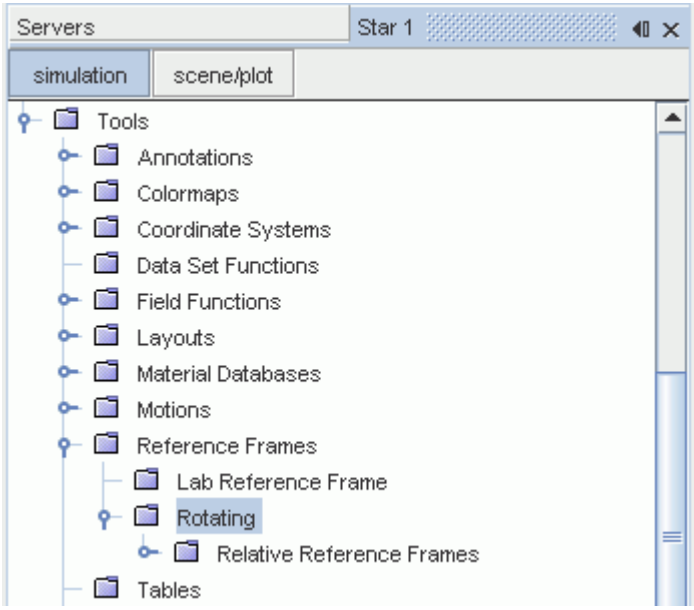
The label of the read-only Rotating reference frame includes the name of the object with which it was created.



If you rename the related object, the label of the Rotating reference frame changes accordingly.

User-Defined Rotating Reference Frame

A user-defined rotating reference frame is a rotating frame of reference that can be applied to regions to generate a constant grid flux, or used in post-processing as an alternative frame of reference for certain quantities. Add this type of frame using the pop-up menu of the **Reference Frames** node, and define it using its properties. For more information, see Reference Frames Reference.



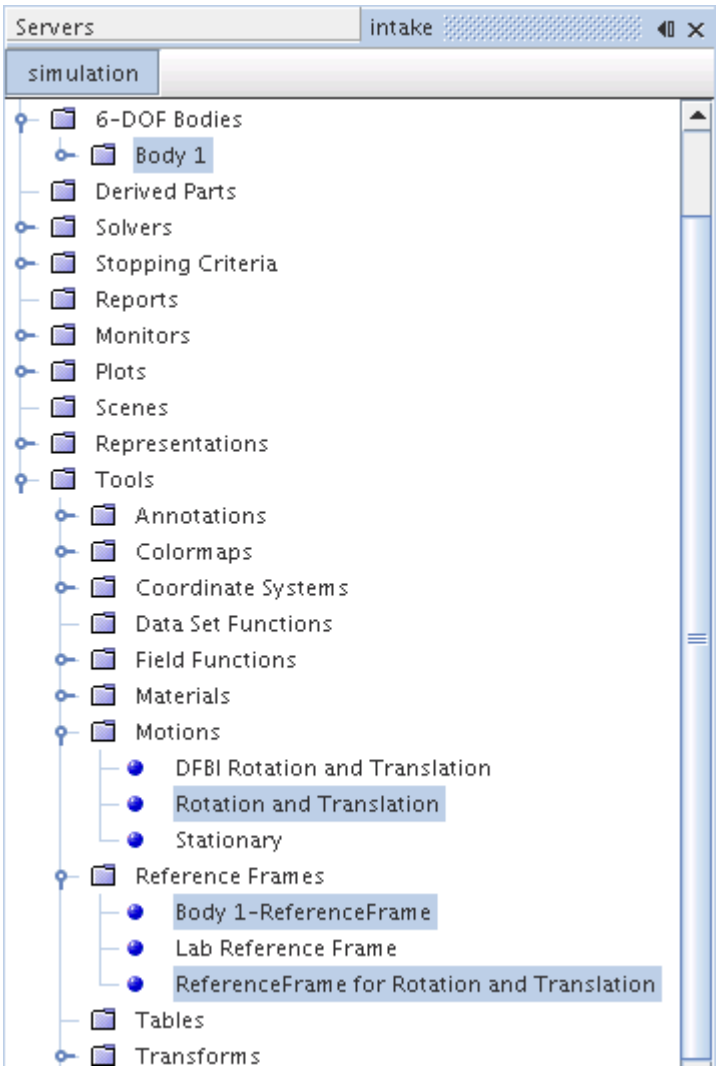
User-defined rotating reference frames can have further relative rotating frames defined within them. This creates a nested hierarchy of rotating reference frames, where child frames are defined relative to the parent frame.

Read-Only Rotating and Translating Reference Frame

A read-only Rotating and Translating reference frame is created automatically for each of the following objects that you add to the simulation:

- Rotation and Translation motion
- 6-DOF body

The label of the read-only Rotating and Translating reference frame includes the name of the object with which it was created.

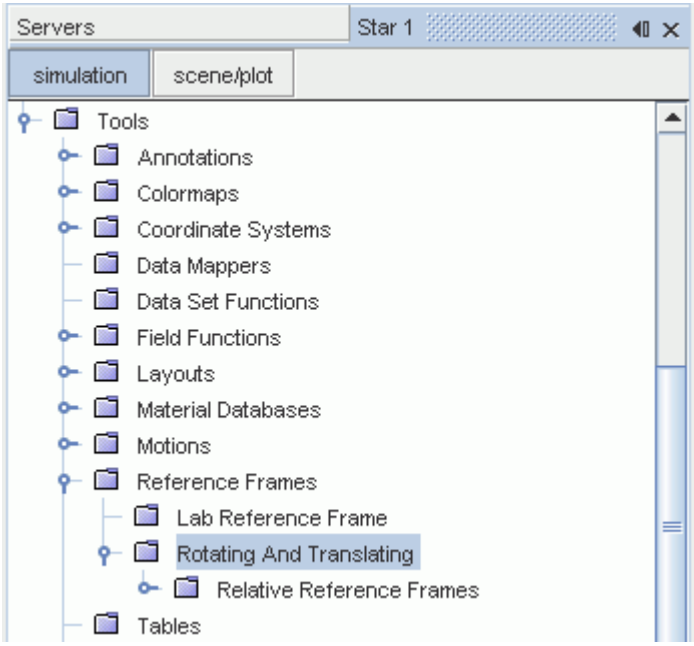


If you rename the related object, the label of the Rotating and Translating reference frame changes accordingly.

User-Defined Rotating and Translating Reference Frame

A user-defined rotating and translating reference frame is a moving frame of reference that can be applied to regions to generate a constant grid flux, or used in post-processing as an alternative frame of reference for certain quantities. It has inputs for both rotation and translation.

Add this type of reference frame using the pop-up menu of the **Reference Frames** manager node, and define it using its properties. For more information, see Reference Frames Reference.

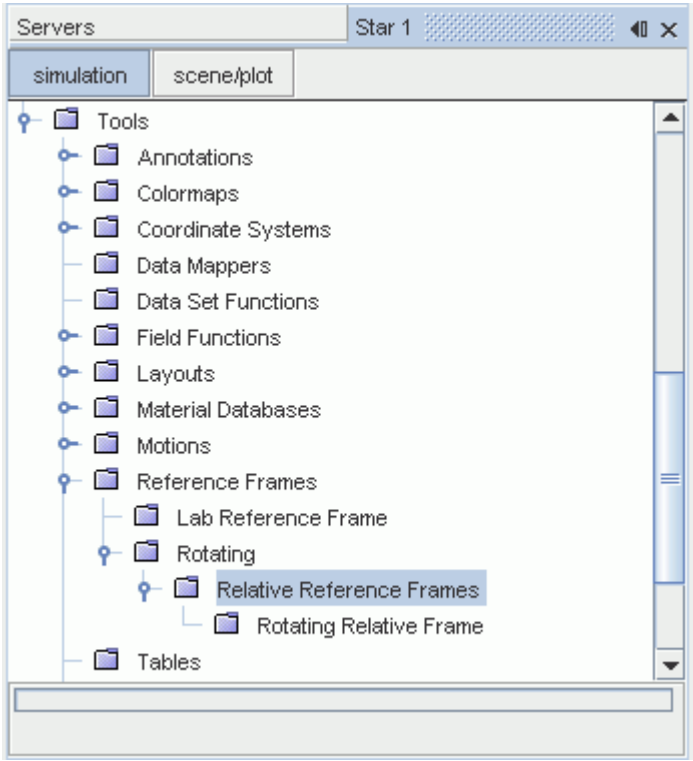


User-defined rotating and translating reference frames can have further relative rotating frames defined within them. This creates a nested hierarchy of reference frames, where child frames are defined relative to the parent frame. The child reference frames have rotation only.

Relative Reference Frame

Relative reference frames are rotating reference frames defined as children of other reference frames. They are contained within the **Relative Reference Frames** manager node, which is a child of both Rotating and Rotating and Translating reference frames. You add a new relative reference frame by using the right-click menu of the **Relative Reference Frames** manager node.

Relative reference frame nodes can be renamed, copied, and deleted as required. Nodes of relative reference frames which are assigned to the Motion Specification of regions cannot be deleted. They must be unassigned before the option to delete becomes available.



For more information on all available properties, see Reference Frames Reference.

Contents:

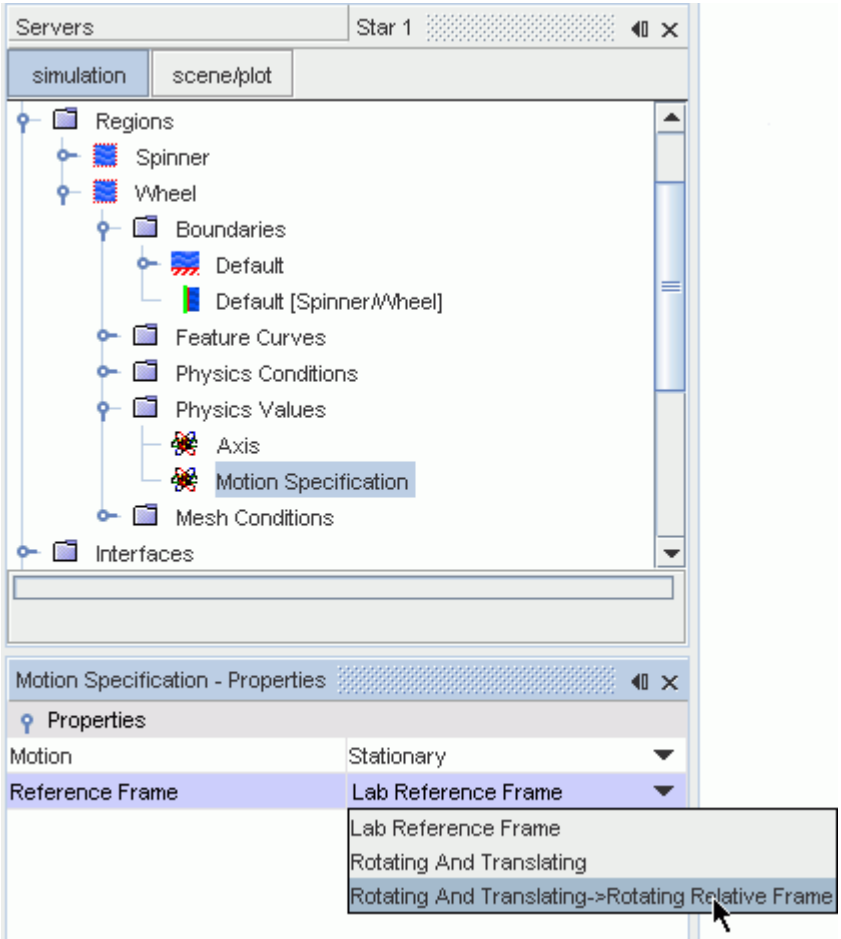
Working with Reference Frames
Reference Frames Reference
Governing Equations in Moving Reference Frames

Working with Reference Frames

One use of reference frames is to mimic the effect of a constant motion without actually moving the vertices of the mesh. This approach means that a steady solution only is required, without the need for a transient method. This is a simplification, however, and an accurate analysis of the fluid flow in such a system is likely to require use of an appropriate motion in transient flow.

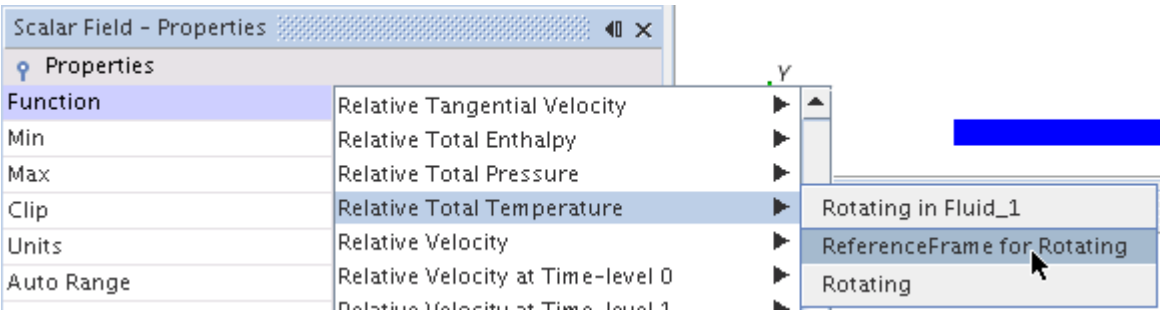
To apply a reference frame to a region:

1. Create the required reference frame (rotating, or rotating and translating).
2. Add any further relative reference frames.
3. Select the **Motion Specification** node of the relevant region and set *Reference Frame* to the newly created frame.



Currently, only the morphing and solid displacement motions can be used with user-defined reference frames. All other motions require either the Laboratory reference frame, or add their own reference frames automatically.

All reference frames defined in a simulation become available in any drop-down list of scalar or vector values, as an option of relative values.



You can display reference frames in scalar or vector scenes, add them to reports, or plot them in monitor and XY plots.

Reference frames can be renamed, and user-defined reference frames can also be copied. You can copy the properties of one user-defined reference frame onto another, provided they are of the same type.

User-defined reference frames may be deleted using the standard methods, unless the reference frame is being used by a region. Read-only reference frames are deleted only when the object associated with them is deleted. This is possible provided they are not being used by any region.

Reference Frames Reference

Reference Frames Right-Click Actions

These pop-up menu items are available when you right-click the **Reference Frames** manager node.

New	Creates a new user-defined reference frame.	
	Rotating	Adds a user-defined Rotating reference frame to the **Reference Frames** node.
	Rotating and Translating	Adds a user-defined Rotating and Translating reference frame to the **Reference Frames** node.

Rotating Reference Frame Properties

These properties are shared by both the read-only and user-defined rotating reference frames, but are editable only for the user-defined type.

Axis Direction	The Cartesian direction vector specifying the axis of rotation.
Axis Origin	The Cartesian position vector specifying the origin of the axis of rotation.
Coordinate System	Specifies the coordinate system (laboratory by default) for defining the origin and axis. The drop-down list of this property includes any local coordinate systems that exist in the simulation. See Coordinate Systems.
Rotation Rate	The rotation rate around the axis specified by *Direction* and *Origin*.

Rotating and Translating Reference Frame Properties

These properties are shared by both the read-only and user-defined Rotating and Translating reference frames, but are editable only for the user-defined type.

Axis Direction	The Cartesian direction vector specifying the axis of rotation.
Axis Origin	The Cartesian position vector specifying the origin of the axis of rotation.
Coordinate System	Specifies the coordinate system (laboratory by default) for defining the origin and axis. The drop-down list of this property includes any local coordinate systems that exist in the simulation. See Coordinate Systems.
Rotation Rate	The rotation rate around the axis specified by *Direction* and *Origin*.
Translation Velocity	The translation velocity of the region. This is additive with *Rotation Rate*.

Relative Reference Frames Right-Click Actions

These pop-up menu items are available when you right-click the **Relative Reference Frames** manager node.

New	Creates a new rotating relative reference frame.

Relative Reference Frame Properties

Axis Direction	The Cartesian direction vector specifying the axis of rotation.
Axis Origin	The Cartesian position vector specifying the origin of the axis of rotation.
Coordinate System	Specifies the coordinate system (laboratory by default) for defining the origin and axis. The drop-down list of this property includes any local coordinate systems that exist in the simulation. See Coordinate Systems.
Rotation Rate	The rotation rate around the axis specified by *Direction* and *Origin*.

Governing Equations in Moving Reference Frames

Moving Reference Frames (MRF) are reference frames that can rotate and translate with respect to the laboratory reference frame.

MRF models in STAR-CCM+ assume that:

- The angular velocity of the body is constant.
- The mesh is rigid.

The governing equations can be written in terms of:

- the velocity in the laboratory reference frame (absolute velocity), or
- the velocity relative to the moving reference frame (relative velocity).

Contents:

Incompressible Navier-Stokes Equations in the Laboratory Reference Frame
Incompressible Navier-Stokes Equations in a Moving Reference Frame with Relative Velocity
Incompressible Navier-Stokes Equations in a Moving Reference Frame with Absolute Velocity
Embedded Reference Frames

Incompressible Navier-Stokes Equations in the Laboratory Reference Frame

This section presents the incompressible Navier-Stokes equations for the sake of simplicity. The extra terms that appear in the compressible Navier-Stokes equations receive no contribution from the motion of the reference frame.

In the laboratory reference frame, the incompressible Navier-Stokes equations are written as:

$$\nabla \cdot \mathbf{u}_0 = 0 \tag{113}$$

$$\frac{D\mathbf{u}_0}{Dt} = -\nabla(p/\rho) + \nu\nabla \cdot \nabla\mathbf{u}_0 \tag{114}$$

Expanding the substantial derivative in the momentum equation:

$$\frac{\partial\mathbf{u}_0}{\partial t} + \nabla \cdot \left(\mathbf{u}_0 \otimes \mathbf{u}_0\right) = -\nabla(p/\rho) + \nu\nabla \cdot \nabla\mathbf{u}_0 \tag{115}$$

The following relation is used to transform the convection term in the conservative form:

$$\nabla \cdot \left(\mathbf{u}_0 \otimes \mathbf{u}_0\right) = \mathbf{u}_0 \cdot \nabla\mathbf{u}_0 + \underbrace{\left(\nabla \cdot \mathbf{u}_0\right)}_{=0} \mathbf{u}_0 \tag{116}$$

Using Eqn. (116), the momentum equation becomes:

$$\frac{\partial\mathbf{u}_0}{\partial t} + \mathbf{u}_0 \cdot \nabla\mathbf{u}_0 = -\nabla(p/\rho) + \nu\nabla \cdot \nabla\mathbf{u}_0 \tag{117}$$

Incompressible Navier-Stokes Equations in a Moving Reference Frame with Relative Velocity

Consider a moving reference frame p rotating with a constant angular velocity $\boldsymbol{\omega}_p$ as measured in the laboratory frame 0. The velocity $\mathbf{u}_p$ relative to the moving reference frame is related to the absolute velocity $\mathbf{u}_0$:

$$\mathbf{u}_0 = \mathbf{u}_p + \boldsymbol{\omega}_p \times \mathbf{r} \tag{118}$$

where $\mathbf{r}$ denotes the position vector from the origin of the moving reference frame p to the center of a control volume.

Eqn. (118) is used to obtain the Navier-Stokes equations in terms of the relative velocity. First, the left-hand side of Eqn. (114) is transformed:

$$\frac{D\mathbf{u}_0}{Dt} = \frac{D\mathbf{u}_p}{Dt} + \frac{\mathrm{d}\,\boldsymbol{\omega}_p}{\mathrm{d}\,t} \times \mathbf{r} + 2\boldsymbol{\omega}_p \times \mathbf{u}_p + \boldsymbol{\omega}_p \times \boldsymbol{\omega}_p \times \mathbf{r} \tag{119}$$

Using Eqn. (116),

$$\frac{D\mathbf{u}_0}{Dt} = \frac{\partial\mathbf{u}_p}{\partial t} + \nabla \cdot \left(\mathbf{u}_p \otimes \mathbf{u}_p\right) + \frac{\mathrm{d}\,\boldsymbol{\omega}_p}{\mathrm{d}\,t} \times \mathbf{r} + 2\boldsymbol{\omega}_p \times \mathbf{u}_p + \boldsymbol{\omega}_p \times \boldsymbol{\omega}_p \times \mathbf{r} \tag{120}$$

The diffusive term is:

$$\begin{aligned} \nabla \cdot \nabla\mathbf{u}_0 &= \nabla \cdot \nabla\left[\mathbf{u}_p + \boldsymbol{\omega}_p \times \mathbf{r}\right] \\ &= \nabla \cdot \nabla\mathbf{u}_p + \underbrace{\nabla \cdot \nabla\left(\boldsymbol{\omega}_p \times \mathbf{r}\right)}_{=0} \\ &= \nabla \cdot \nabla\mathbf{u}_p \end{aligned} \tag{121}$$

Applying all transformations, the momentum equation becomes:

$$\frac{\partial \mathbf{u}_p}{\partial t} + \nabla \cdot \left(\mathbf{u}_p \otimes \mathbf{u}_p\right) + \frac{\mathrm{d}\,\boldsymbol{\omega}_p}{\mathrm{d}\,t} \times \mathbf{r} + 2\boldsymbol{\omega}_p \times \mathbf{u}_p + \boldsymbol{\omega}_p \times \boldsymbol{\omega}_p \times \mathbf{r} = -\nabla(p/\rho) + \nu \nabla \cdot \nabla \mathbf{u}_p \tag{122}$$

Both the convection and convected velocities are in terms of velocity relative to the rotating frame. This formulation is also known as the relative velocity formulation.

Similarly, the continuity equation is transformed using:

$$\begin{aligned} \nabla \cdot \mathbf{u}_0 &= \nabla \cdot \left[\mathbf{u}_p + \boldsymbol{\omega}_p \times \mathbf{r}\right] \\ &= \nabla \cdot \mathbf{u}_p + \underbrace{\nabla \cdot \left(\boldsymbol{\omega}_p \times \mathbf{r}\right)}_{=0} \\ &= \nabla \cdot \mathbf{u}_p \end{aligned} \tag{123}$$

giving

$$\nabla \cdot \mathbf{u}_p = 0 \tag{124}$$

Eqn. (124) and Eqn. (122) form the incompressible Navier-Stokes equations in the moving frame, with relative velocity.

Incompressible Navier-Stokes Equations in a Moving Reference Frame with Absolute Velocity

In this form, the convection velocity is the velocity $\mathbf{u}_p$ relative to the moving reference frame and the convected velocity is in terms of the velocity $\mathbf{u}_0$ in the laboratory frame (that is, the absolute velocity). The convection term is of the form $\nabla \cdot \left(\mathbf{u}_p \otimes \mathbf{u}_0\right)$.

Starting from Eqn. (122), the term corresponding to the convective fluxes $\nabla \cdot \left(\mathbf{u}_p \otimes \mathbf{u}_p\right)$ is written as:

$$\begin{aligned} \nabla \cdot \left(\mathbf{u}_p \otimes \mathbf{u}_p\right) &= \nabla \cdot \left[\mathbf{u}_p \otimes \left(\mathbf{u}_0 - \boldsymbol{\omega}_p \times \mathbf{r}\right)\right] \\ &= \nabla \cdot \left(\mathbf{u}_p \otimes \mathbf{u}_0\right) - \underbrace{\nabla \cdot \mathbf{u}_p}_{=0} \left(\boldsymbol{\omega}_p \times \mathbf{r}\right) - \mathbf{u}_p \cdot \nabla\left(\boldsymbol{\omega}_p \times \mathbf{r}\right) \end{aligned} \tag{125}$$

The term $\mathbf{u}_p \cdot \nabla\left(\boldsymbol{\omega}_p \times \mathbf{r}\right)$ is expanded as follows:

$$\begin{aligned} \mathbf{u}_p \cdot \nabla\left(\boldsymbol{\omega}_p \times \mathbf{r}\right) &= \left(\mathbf{u}_p \cdot \underbrace{\nabla \boldsymbol{\omega}_p}_{=0}\right) \times \mathbf{r} + \boldsymbol{\omega}_p \times \left(\mathbf{u}_p \cdot \nabla \mathbf{r}\right) \\ &= \boldsymbol{\omega}_p \times \mathbf{u}_p \quad \text{with} \quad \nabla \mathbf{r} = \mathbf{I} \end{aligned} \tag{126}$$

Therefore, Eqn. (125) becomes:

$$\nabla \cdot \left(\mathbf{u}_p \otimes \mathbf{u}_p\right) = \nabla \cdot \left(\mathbf{u}_p \otimes \mathbf{u}_0\right) - \boldsymbol{\omega}_p \times \mathbf{u}_p \tag{127}$$

Simplifying the steady terms in the left-hand side of Eqn. (122):

$$\begin{aligned}\nabla \cdot \left(\mathbf{u}_p \otimes \mathbf{u}_p\right) + 2\boldsymbol{\omega}_p \times \mathbf{u}_p + \boldsymbol{\omega}_p \times \boldsymbol{\omega}_p \times r \\ &= \nabla \cdot \left(\mathbf{u}_p \otimes \mathbf{u}_0\right) + \boldsymbol{\omega}_p \times \mathbf{u}_p + \boldsymbol{\omega}_p \times \boldsymbol{\omega}_p \times r \\ &= \nabla \cdot \left(\mathbf{u}_p \otimes \mathbf{u}_0\right) + \boldsymbol{\omega}_p \times \left(\mathbf{u}_p + \boldsymbol{\omega}_p \times \mathbf{r}\right) \\ &= \nabla \cdot \left(\mathbf{u}_p \otimes \mathbf{u}_0\right) + \boldsymbol{\omega}_p \times \mathbf{u}_0\end{aligned} \tag{128}$$

The unsteady term of Eqn. (122) is transformed as:

$$\begin{aligned}\frac{\partial \mathbf{u}_p}{\partial t} &= \frac{\partial \mathbf{u}_0}{\partial t} - \frac{\partial\left(\boldsymbol{\omega}_p \times \mathbf{r}\right)}{\partial t} \\ &= \frac{\partial \mathbf{u}_0}{\partial t} - \underbrace{\frac{\partial \boldsymbol{\omega}_p}{\partial t}}_{=0} \times \mathbf{r} - \boldsymbol{\omega}_p \times \underbrace{\frac{\partial \mathbf{r}}{\partial t}}_{=0}\end{aligned} \tag{129}$$

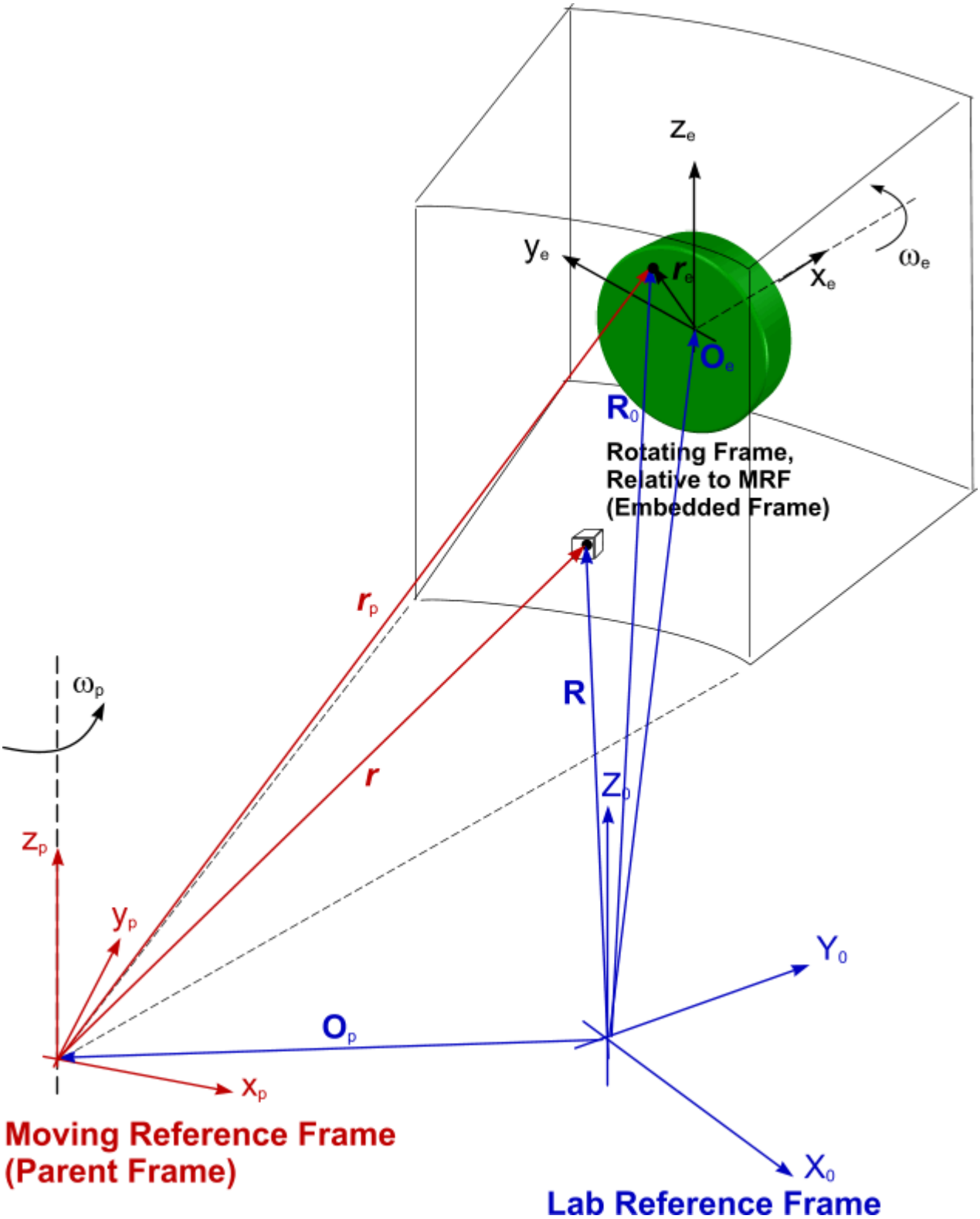
Using all the transformed terms, the momentum equation in terms of the absolute velocity is:

$$\frac{\partial \mathbf{u}_0}{\partial t} + \nabla \cdot \left(\mathbf{u}_p \otimes \mathbf{u}_0\right) + \boldsymbol{\omega}_p \times \mathbf{u}_0 = -\nabla(p/\rho) + \nu \nabla \cdot \nabla \mathbf{u}_0 \tag{130}$$

Eqn. (130) is the momentum equation in the moving reference frame with the absolute velocity formulation. In this formulation, Eqn. (113) is the continuity equation. This formulation is solved in STAR-CCM+.

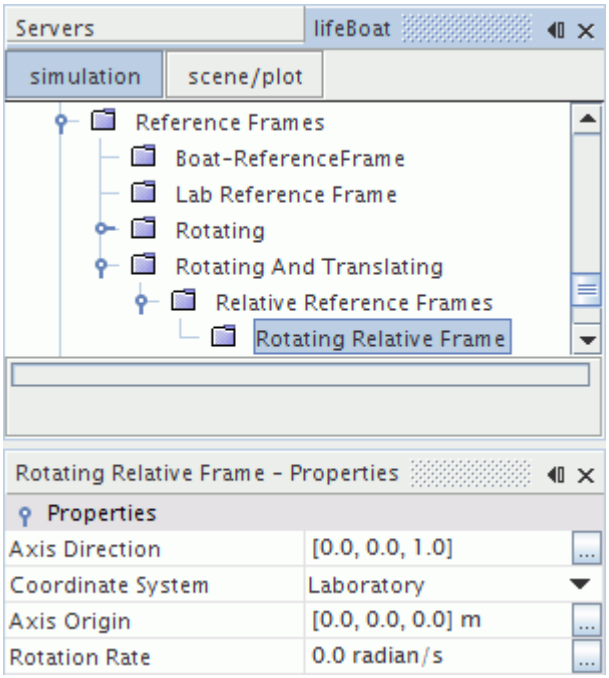
Embedded Reference Frames

An embedded reference frame e allows you to specify rotational motion with respect to a parent frame p that can be both rotating and translating. The diagram below shows an example of an embedded, rotating frame that is defined with respect to a moving reference frame and the laboratory reference frame. X, Y, and Z represent the axes for each frame. O represents the origin in each frame.



The Moving Reference Frame model handles motion in moving reference frames. The Relative Moving Reference Frame model handles motion in embedded reference frames. Both models are formulated in terms of the absolute velocity formulation that is given by Eqn. (124) and Eqn. (130).

Embedded reference frames are represented by nodes under the **[Moving Reference Frame]** > **Relative Reference Frames** manager.



Both models require a single initial evaluation of grid flux throughout the domain. The source term, being a function of the velocity field, is evaluated at each iteration.

Moving Reference Frame

In the Moving Reference Frame, the grid flux is calculated as:

$$G_f = \left(\mathbf{a} \cdot \mathbf{v}_g\right)_f \tag{131}$$

with

$\mathbf{v}_g = \boldsymbol{\omega}_p \times \mathbf{r} + \mathbf{V}_{tp}$

where

- the subscript f corresponds to a face value
- the subscript c corresponds to a cell value
- $\mathbf{v}_g$ is the grid velocity
- $\boldsymbol{\omega}_p$ is the angular velocity of the moving frame p. This angular velocity is measured in the laboratory reference frame.
- $\mathbf{r} = \mathbf{R} - \mathbf{O}_p$ is the position vector with respect to the parent frame. $\mathbf{O}_p$ is the origin of the parent frame.
- $\mathbf{V}_{tp}$ is the translational velocity of the parent frame. This translational velocity is measured in the laboratory reference frame.
- $\mathbf{R}$ is the position vector of a control volume in the parent frame, as measured in the laboratory reference frame.

The source term is calculated as $\boldsymbol{\omega}_p \times \mathbf{v}_c$, where $\mathbf{v}_c$ is the velocity at the cell center as measured in the laboratory reference frame.

Embedded Reference Frame

In the embedded Moving Reference Frame, the grid flux is calculated as:

$$G_f = \left(\mathbf{a} \cdot \mathbf{v}_{total}\right)_f \tag{132}$$

where the total grid velocity $\mathbf{v}_{total}$ is calculated as the sum of grid velocities of the parent frame p and the embedded frame e:

$$\mathbf{v}_{total} = \mathbf{v}_p + \mathbf{v}_e = \boldsymbol{\omega}_p \times \mathbf{r}_p + \mathbf{V}_{tp} + \boldsymbol{\omega}_e \times \mathbf{r}_e \tag{133}$$

where

- $\boldsymbol{\omega}_e$ is the angular velocity of the embedded moving frame.
- $\mathbf{r}_p = \mathbf{R}_0 - \mathbf{O}_p$ is the position vector with respect to the parent frame.
- $\mathbf{r}_e = \mathbf{R}_0 - \mathbf{O}_e$ is the position vector with respect to the embedded frame. $\mathbf{O}_e$ is the origin of the embedded frame.
- $\mathbf{R}_0$ is the position vector of a control volume in the embedded frame, as measured in the laboratory reference frame.

The source term is calculated as $\boldsymbol{\omega}_{total} \times \mathbf{v}_c$, where the angular velocity is the vector sum of the angular velocities of the parent and embedded frames:

$$\boldsymbol{\omega}_{total} = \boldsymbol{\omega}_p + \boldsymbol{\omega}_e \tag{134}$$

Modeling Motion

In engineering, motion can be defined as the change in location of a body relative to a particular frame of reference. Such motion is imparted to any fluid in contact with the surfaces of the body, and must be considered in an analysis of the fluid behavior.

Some examples of motion-induced flows are:

- Mixing vessels, where mixer blades drive the flow
- Turbomachinery
- Flexible diaphragms in medical applications
- Ship and propeller motion
- Fluid structure interaction

In STAR-CCM+, it is also possible to model the motion of rigid bodies in six degrees of freedom with the fluid system.

Contents:

Understanding Categories of Motion
What Is a Motion?
Working with Motions
Simulating Constant Rigid Motion
Working with Morphing
Working with User-Defined Vertex Motion
General Motion Formulation

Understanding Categories of Motion

There are four broad categories for defining motion in STAR-CCM+:

- The first category contains those methods that involve actual displacement of mesh vertices in real time: use these methods with a transient analysis.
- The second category contains the method whereby an unsteady motion problem is converted to a steady-state problem by imposing a moving frame of reference on a static mesh. Rigid rotation and/or translation can be simulated in this way.
- The third type of motion is a corollary of the real-time displacement, and applies only to harmonic balance flutter. The imposed modal displacement of the flutter boundary is applied to the frequency domain solver.
- The fourth category is a motion that is performed in steady-state where the shape of the solution domain is part of the solution.

  This type of motion applies, for example, to fluid-structure-interaction simulations where you calculate the steady-state flow around a static deformation of the structure under the action of the steady-state fluid loads (aeroelastic or hydroelastic equilibrium). Since the velocity of the structure is zero (or almost zero), the grid flux is zero. For this motion type, the morpher is used to deform the mesh at the boundaries.

Another use case for this morphing motion in steady-state is the simulation of die extrusion of non-Newtonian fluids. The extrudate is modeled as a free surface using the Arbitrary Lagrangian-Eulerian (ALE) method to track the shape of the free surface boundary.

For more information, see Setting up Extrusion and Free Surface Flow.

The specific methods available in each category are listed as follows:

Category 1: Mesh Displacement in Real Time

Several methods are available in this category:

- Rigid Motion
  - Rotation
  - Translation
  - Rotation and Translation
- Morphing (deforming mesh)
- Dynamic Fluid Body Interaction
  - DFBI Rotation and Translation
  - DFBI Embedded Rotation
  - DFBI Morphing
  - DFBI Superposed Rotation
- Solid Displacement
- User-Defined Vertex Motion

The space conservation law is enforced in non-rigid body motion types by solving for a volume conservation equation.

Category 2: Moving Reference Frame in Steady-State

The reference frame types that can be used in this category are:

- Rotating
- Rotating and Translating

For more information, see Reference Frames.

Category 3: Harmonic Balance Flutter

There is only one option for this approach.

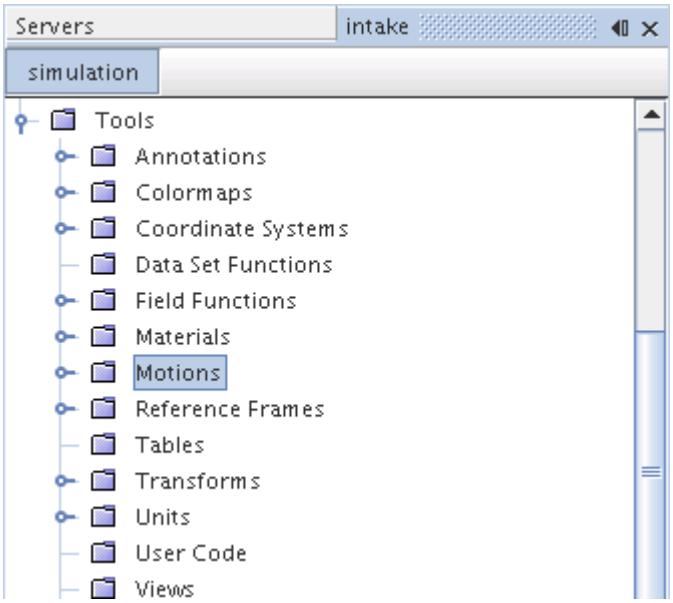
Category 4: Morphing in Steady-State

Morphing (deforming mesh)

What Is a Motion?

In STAR-CCM+, motion is defined using motion objects. These objects are contained within the **Motions** manager node in the main simulation tree and encapsulate the different methods available for introducing motion.

Setting a region to use a motion object other than **Stationary** results in movement of the mesh vertices during the simulation. This manager node uses a pop-up menu to add motion objects.



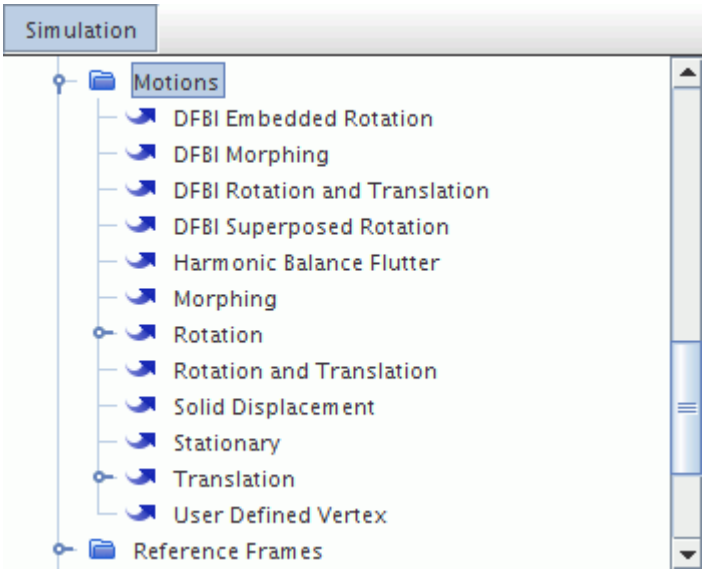
As motion is defined relative to a reference frame, STAR-CCM+ provides reference frame objects to complete the motion specification for a region. These objects are contained within the **Reference Frames** manager node in the main simulation tree.

While the types of motion objects differ, they are all subject to common procedures for working with motions.

The currently available motions are:

- Stationary
- Rotation
- Translation
- Rotation and Translation
- Morphing
- DFBI Rotation and Translation
- DFBI Embedded Rotation
- DFBI Morphing
- DFBI Superposed Rotation
- Harmonic Balance Flutter
- Solid Displacement
- User-Defined Vertex

You can add more than one of any types other than Stationary. This feature is helpful when you want to retain more than one set of property specifications for a particular motion.



If there is more than one motion of a particular type in the simulation, then the second and subsequent ones have numbers at the end of the labels. (Assuming they have not been renamed).

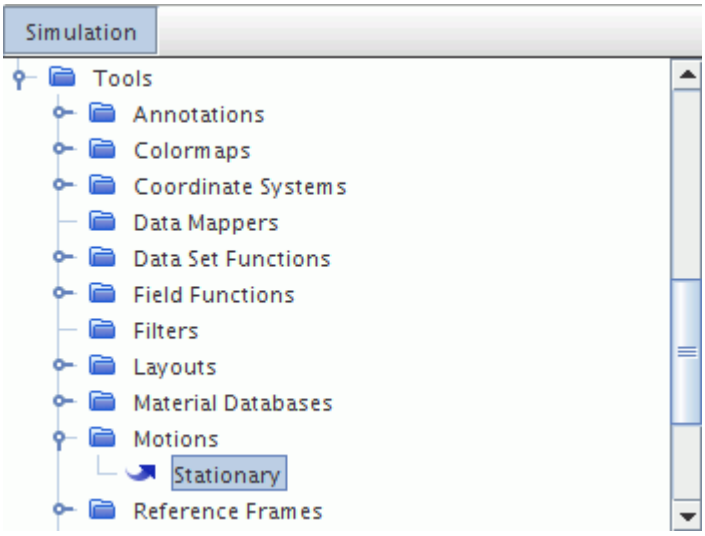
Motions can be copied, renamed, or deleted.

Contents:

What Is the Stationary Motion?
What Is a Solid Displacement Motion?
Pop-Up Menu of the Motions Manager

What Is the Stationary Motion?

The Stationary motion is used for simulations that do not involve any kind of motion. The **Stationary** motion node, which is subject to common menu items, is the default motion in a STAR-CCM+ simulation.

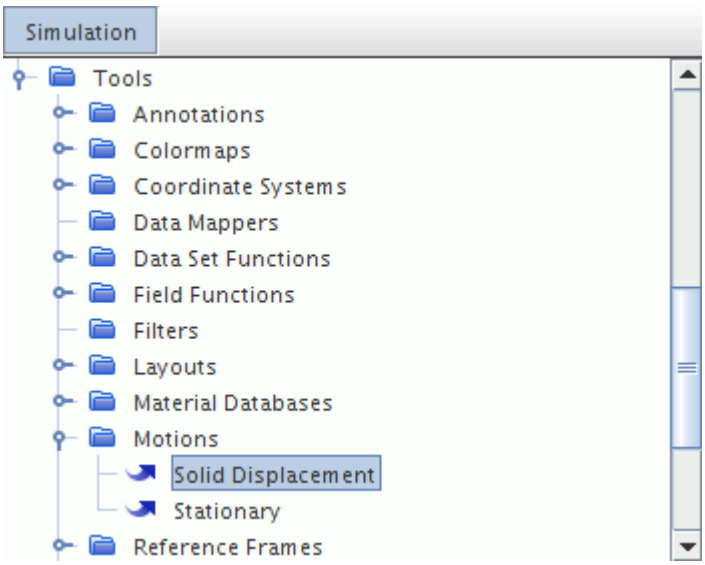


Stationary is the only type of motion present in the motions manager when a simulation is created, and cannot be deleted. It is not necessary to create more motions of this type.

What Is a Solid Displacement Motion?

Solid displacement motion, also referred to as the large displacement model, is used with the solid stress model to allow the solid mesh to be geometrically updated in response to solid stress displacements. It is useful for cases involving fluid structure interaction.

The **Solid Displacement** motion node represents this motion and is subject to common menu items.



To use solid displacement in a simulation, activate the Solid Stress continuum model, add this motion, and then select it for a region in the Motion Specification region value.

Pop-Up Menu of the Motions Manager

The following are the pop-up menu items available when you right-click the **Motions** manager node.

New	Creates a motion.	
	Rotation	Adds a Rotation motion to the **Motions** node.
	Translation	Adds a Translation motion to the Motions node.
	Rotation And Translation	Adds a Rotation and Translation motion to the **Motions** node.

	Morphing	Adds a new Morphing motion to the **Motions** node.
	DFBI Rotation and Translation	Adds a DFBI Rotation and Translation motion to the **Motions** node.
	DFBI Embedded Rotation	Adds a DFBI Embedded Rotation motion to the **Motions** node.
	DFBI Morphing	Adds a DFBI Morphing motion to the **Motions** node.
	DFBI Superposed Rotation	Adds a DFBI Superposed Rotation motion to the **Motions** node.
	Harmonic Balance Flutter	Adds a Harmonic Balance Flutter motion to the **Motions** node.
	Solid Displacement	Adds a new solid stress Solid Displacement motion to the **Motions** node.
	User-Defined Vertex	Adds a User-Defined Vertex motion to the **Motions** node.

Working with Motions

Motion objects are applied to regions using the Motion Specification node that can be found in the **Physics Values** manager node for a region. By default this node is set to use the Stationary motion with the Lab reference frame.

To add more motions and set them for use on a region, use the following procedure:

- Open the **Tools** node. Add a motion to the **Motions** manager by right-clicking the node and choosing **New** and then the appropriate type of motion.
- Depending on the type of motion you create, STAR-CCM+ automatically adds a read-only reference frame to the simulation tree. If you require an alternative reference frame, or to add a reference frame without first adding a motion:
  a) right-click the **Tools** > **Reference Frames** manager node
  b) choose **New** and then the appropriate type of reference frame.
- Having created the necessary motion and/or reference frame, open the **Physics Values** node of the region, click the **Motion Specification** node, and select the desired motion and reference frame in the *Properties* window.
- Depending on the motion that is specified, set the Axis or control point list region values.

Currently it is not possible to use non-trivial combinations of a motion and a reference frame. If a motion is selected, the reference frame is automatically set to Lab reference frame. On the other hand, if a reference frame other than Lab is selected, the motion is automatically set to Stationary in that reference frame.

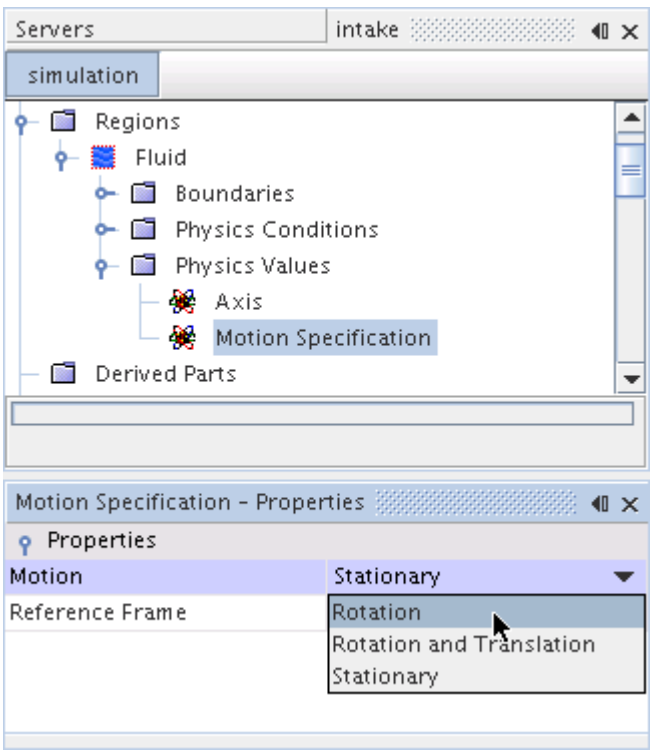
Contents:

Setting the Motion Specification for the Region
Adding Motions
Copying Motions
Renaming Motions
Deleting Motions

Setting the Motion Specification for the Region

Motion is applied on a region basis in STAR-CCM+. Each region has a **Motion Specification** node that defines the motion and reference frame that is applied to the region.

The default motion is the stationary motion with the Lab Reference Frame. The governing equations for that region are therefore solved within the Lab Reference Frame. To access the properties of the motion specification, select the **Motion Specification** node.

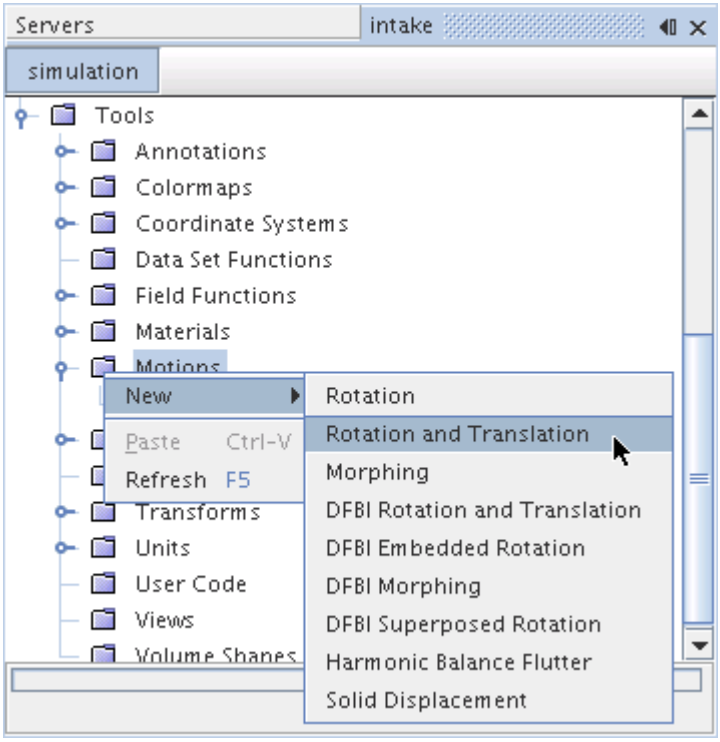


Motion Specification Properties

Motion	Selects one of the existing motions in the simulation from a drop-down list.
Reference Frame	Selects one of the existing reference frames in the simulation from a drop-down list.

Adding Motions

To create a motion (other than Stationary), right-click the **Motions** node and select the motion type that you wish to create from the **New** pop-up menu.



Copying Motions

A motion can be copied. When a motion is copied, all properties are copied, resulting in an exact duplicate of the original.

You can also copy the properties of one motion onto another, provided they are of the same type.

Copying Properties of a Motion

It is possible to copy properties among similar motions of all categories except for motions that do not have properties.

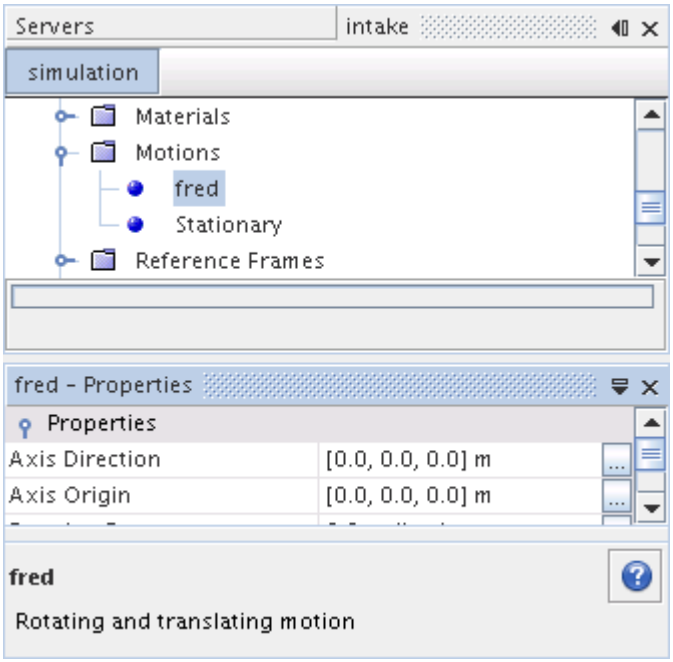
Among motions of the same type, this operation applies all of the properties of the source motion.

It is not possible to copy properties between different types of motions.

Renaming Motions

Any of the motions can be renamed using the standard mechanisms, but be careful that the name you apply still reflects the type of motion.

If it does not (as in the example below), identify the type of the motion from the description area at the bottom of the *Properties* window.



Deleting Motions

Motions other than Stationary can be deleted using the standard methods, unless a region uses the motion. In this case the **Delete** pop-up menu item is grayed and the <Delete> key has no effect (no warning is issued).

Simulating Constant Rigid Motion

Examples of systems involving rigid motion are mixers, turbines, compressors, pumps, and other devices where an internal or external component rotates about a fixed axis, or translates along a fixed path. It is assumed that the moving component does not deform.

For an approximate analysis of a constant rigid motion, it is possible to apply a moving reference frame to the region that contains the moving part, and to solve the problem in steady-state. This approach gives a solution that represents the time-averaged behavior of the flow, rather than the time-accurate behavior. The section, Using Moving Reference Frames in Steady-State discusses this matter more fully.

When the time-accurate behavior is required, STAR-CCM+ provides two motion options for rigid motion. Applying either of these motions results in mesh vertices being displaced in each time step. Details are given in the section, Using Rigid Motion.

Contents:

Using Moving Reference Frames in Steady-State
Using Rigid Motion of Mesh Vertices

What Is a Rotation Motion?
Superposing Additional Rigid Rotations
What Is a Translation Motion?
What Is a Rotation and Translation Motion?

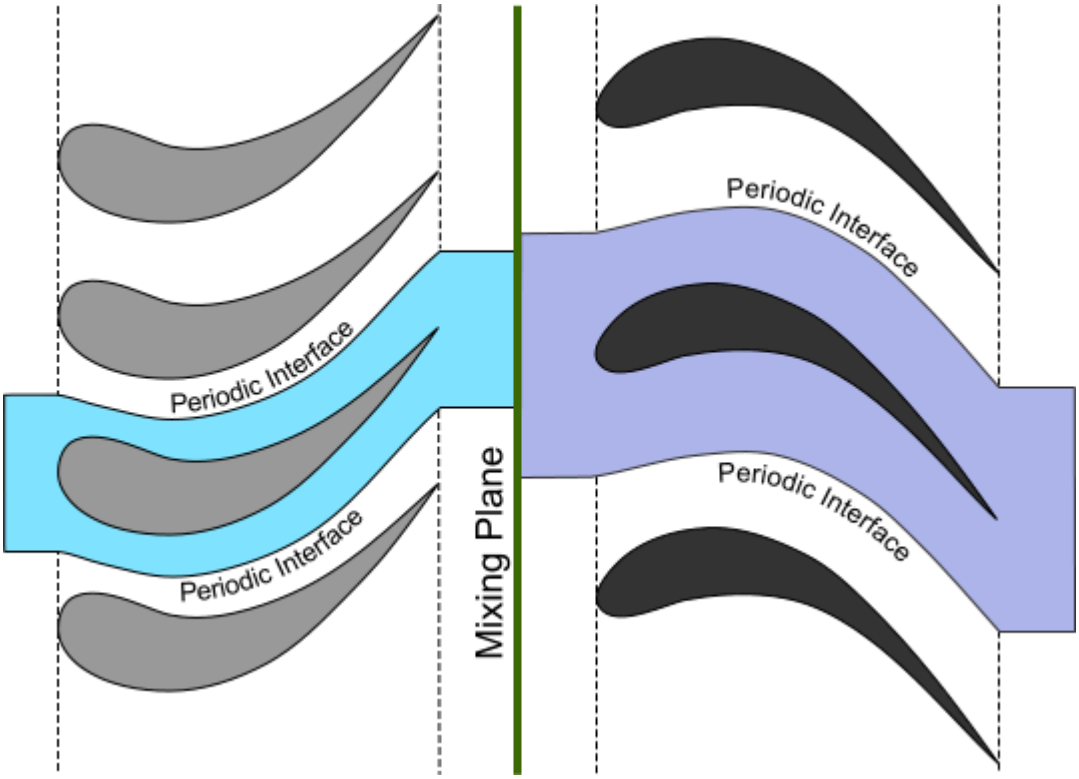
Using Moving Reference Frames in Steady-State

Applying a moving reference frame to a region generates a constant grid flux in the appropriate conservation equations. This grid flux is calculated based on the properties of the reference frame rather than the local motion of cell vertices. A moving reference frame does not change the position of cell vertices.

Do not use moving reference frames if any considerable component of external flow velocity is perpendicular to the axis of rotation. The moving reference frame gives physically realistic results only if the flow in the surrounding volume is axisymmetric. If any component of the surrounding flow is perpendicular to the axis of rotation, unphysical results proportional to the perpendicular component follow.

For the analysis of turbomachinery flows, moving reference frames can be used for two types of steady-state simulation: frozen rotor and mixing plane. In a frozen rotor simulation, a rotating reference frame is applied to the region of mesh containing the rotor blades. The remainder of the mesh is either static or has a different rotating reference frame applied (as multiple regions can have different reference frames). As the rotor blades are frozen in space with respect to the stator blades, the flow field that the stator blades see depends on the relative position of the stages. Different relative positions result in different solutions, which is a major disadvantage of this approach.

In a mixing plane analysis, the problem of relative positioning is overcome by using an indirect mixing plane interface to couple the rotor and stator regions. Flow conditions on either side of the mixing plane interface are averaged circumferentially to provide a single radial profile that is applied to all cell faces at the interface. The basis for this treatment can be appreciated by considering what would be observed when looking at a fast rotating fan: that is, a single rotating surface emitting a constant stream of air independent of blade position. Increasing the number of blades on the fan, increasing the rotational speed, or simply moving farther away from the fan, would all tend to increase the accuracy of this interpretation. The following figure illustrates a rotor and stator assembly for which a mixing plane simulation would be appropriate. If periodic interfaces are applied mid-way between the blades, it is only necessary to include a single blade passage for each stage.



Place the mixing plane interface at a point where the time-averaged properties rather than instantaneous flow details are likely to influence the flow. To create the mixing plane interface, see Creating a Boundary Interface. Select the newly created node under **Interfaces** and set *Type* to **Mixing-Plane Interface**.

The mixing plane approach has the advantage that the relative positioning of the stator and rotor is irrelevant. However, it suffers from the fact that wake effects are lost due to the circumferential averaging. If it becomes necessary to model the interaction between stages in greater detail, STAR-CCM+ provides two further options: rigid motion of mesh vertices in a transient analysis, or a harmonic balance analysis of a single stage with a prescribed wake.

Using Rigid Motion of Mesh Vertices

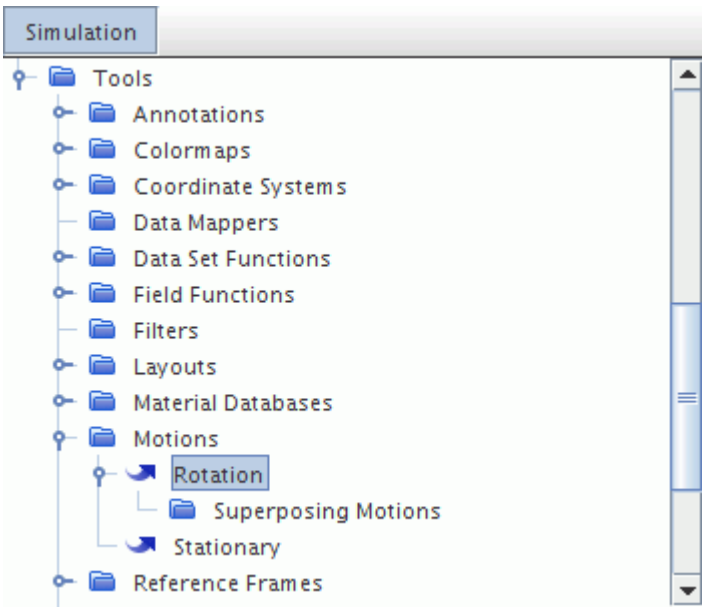
For cases where the motion of an entire region is considered, STAR-CCM+ provides the option to move the mesh vertices of a region during a transient analysis. This mesh motion is specified using either a Rotation motion or a Rotation and Translation motion.

For the Rotation motion, it is possible to superpose additional rigid rotations on top of the original rigid rotation. This superposition permits rotations around multiple axes to take place at the same time.

What Is a Rotation Motion?

A Rotation motion is used to move the mesh vertices of a region around a specified axis.

The **Rotation** node represents this motion and has properties and is subject to common menu items. The **Superposing Motions** child node has its own pop-up menu that is used to add more rigid rotations on top of the original motion.



Adding this motion results in a read-only Rotating reference frame node being added to the **Reference Frames** manager. This node reflects the properties that are entered for the Rotation motion.

Rotation Motion Properties

The *Axis Direction*, *Axis Origin*, and the *Rotation Rate* scalar property, can be specified in the *Properties* window. In addition to constants, you can enter field function syntax directly. You can select a local coordinate system to define the axis and origin. Also, the *Managed Coordinate System* property lets you select the coordinate system that the motion moves.

Axis Direction	The Cartesian direction vector specifying the axis of rotation.
Axis Origin	The Cartesian position vector specifying the origin of the axis of rotation.
Rotation Rate	The rotation rate around the axis that with the specified *Direction* and *Origin*.
Managed Coordinate Systems	Specifies coordinate systems that the motion is to move. The *Select Objects* dialog of this property includes any local coordinate systems that exist in the simulation.
Coordinate System	Specifies the coordinate system (laboratory by default) for defining the origin and axis. The drop-down list of this property includes any local coordinate systems that exist in the simulation.

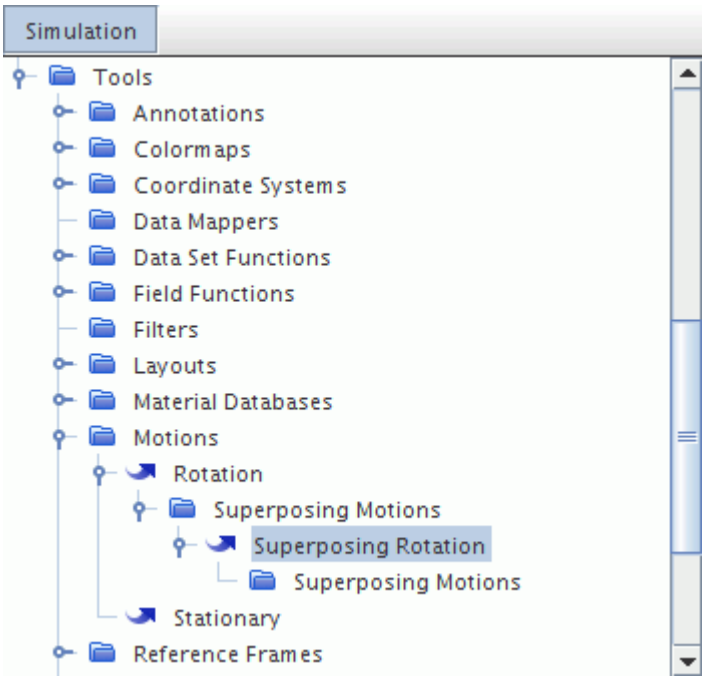
Superposing Additional Rigid Rotations

The **Superposing Motions** node allows you to add more rigid motions to Translation and Rotation motions.

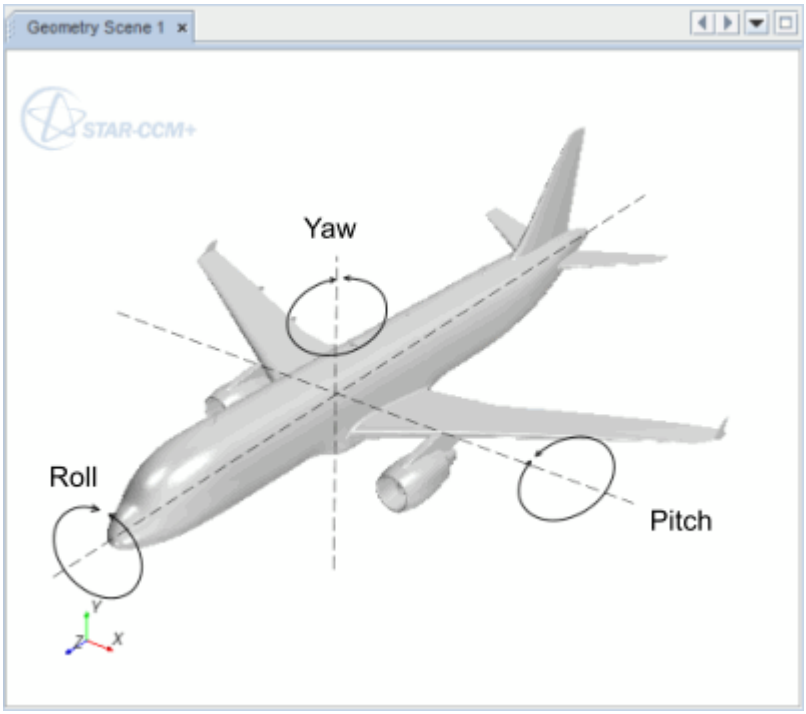
To add a motion, right-click the node and select **New** from the pop-up menu and choose the option **Rotation** or **Translation**, as appropriate.

A **Superposing Rotation** (or **Superposing Translation**) node is added beneath the **Superposing Motions** node. This **Superposing Motions** > **[Motion]** node requires the same set of properties as the **Motions** > **[Motion]** node, and contains its own **Superposing Motions** child node to enable the nesting of further motions beneath it.

Note: You can have up to eight levels of superposed motions nested beneath the parent motion node.

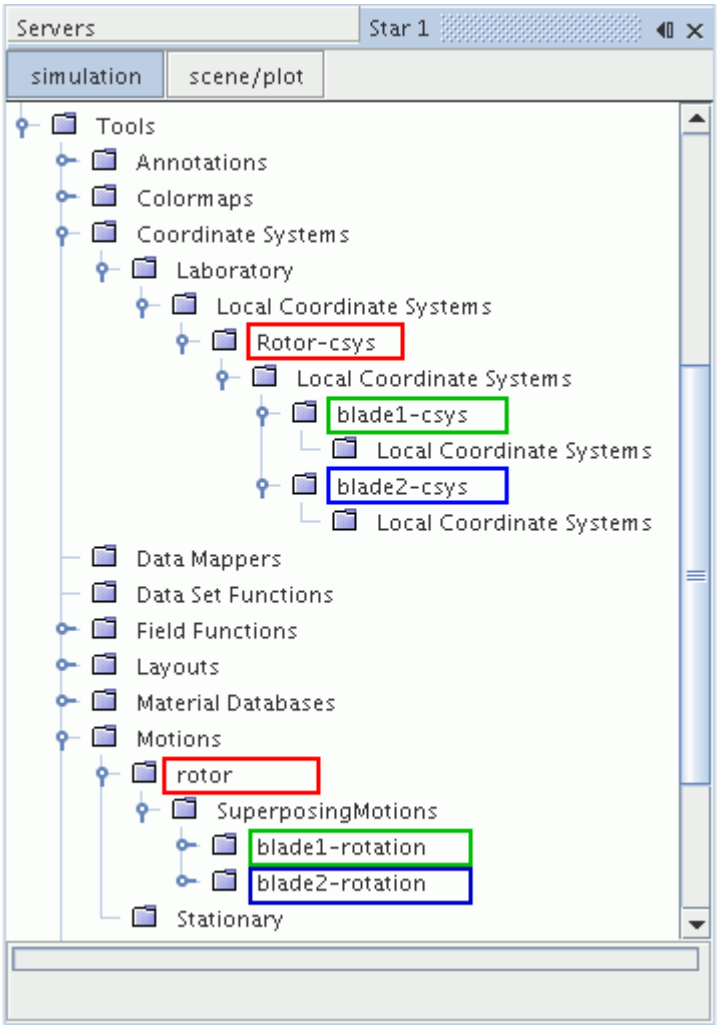


By successively adding superposing motions in this way, it is possible to construct complex motions that apply to the whole mesh and in the regions for which the original motion is active. For example, pitch, yaw and roll for a flying object can be modeled:



The local coordinate system in which the superposing motion is defined must be a nested beneath that of the parent motion.

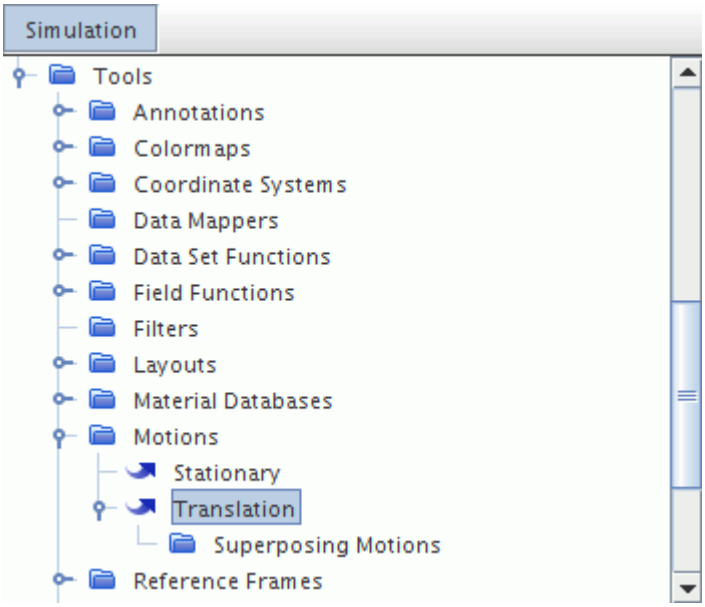
An example of such coordinate systems is shown below. The **rotor** motion is defined in the **Rotor-csys** coordinate system. Coordinate systems **blade1-csys** and **blade2-csys** define the superposed motions: **blade1-rotation** and **blade2-rotation**, respectively. These two latter coordinate systems are children of the **Rotor-csys** coordinate system.



What Is a Translation Motion?

A Translation motion is used to move the mesh vertices of a region with a specified velocity.

The **Translation** node represents this motion. This node has properties and is subject to common menu items. The **Superposing Motions** child node has its own pop-up menu that is used to add more rigid rotations on top of the original motion.



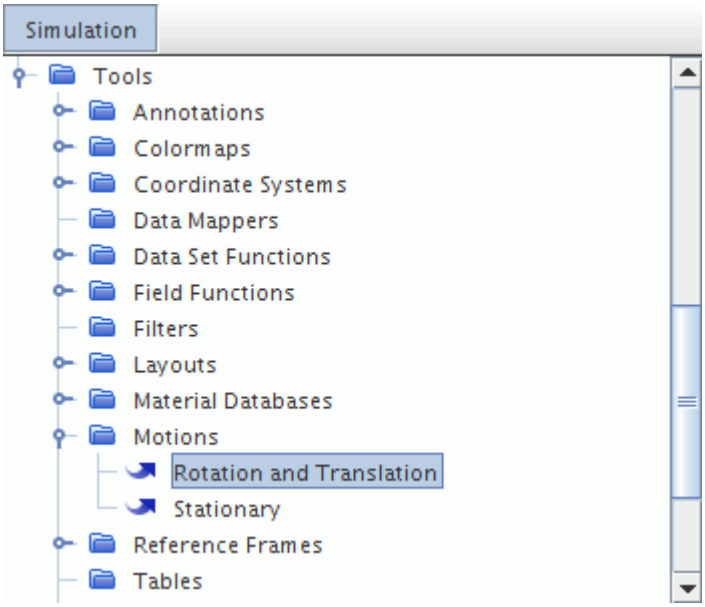
Translation Motion Properties

Translation Velocity	The translation velocity of the region.
Managed Coordinate Systems	Specifies the coordinate system that uses the motion. The *Select Objects* dialog of this property includes any local coordinate systems that exist in the simulation.
Coordinate System	Specifies the coordinate system (laboratory by default) for defining the velocity. The drop-down list of this property includes any local coordinate systems that exist in the simulation.

What Is a Rotation and Translation Motion?

This motion is used to rotate and translate the mesh vertices of a region.

The **Rotation And Translation** node represents this motion. This node has properties and is subject to common menu items.



Adding this motion results in a Rotating and Translating reference frame node being added to the **Reference Frames** manager.

Rotation and Translation Motion Properties

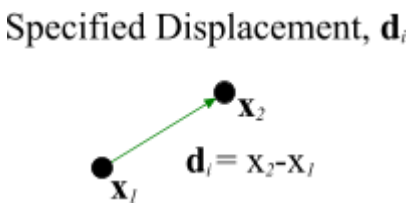
In addition to Rotation motion properties, this motion lets you specify translation velocity.

Axis Direction	The Cartesian direction vector specifying the axis of rotation.
Axis Origin	The Cartesian position vector specifying the origin of the axis of rotation.
Rotation Rate	The rotation rate around the axis that with the specified *Direction* and *Origin*. This option is additive with *Translation Velocity*.
Translation Velocity	The translation velocity of the region. This property is additive with *Rotation Rate*.
Managed Coordinate Systems	Specifies the coordinate system that uses the motion. The *Select Objects* dialog of this property includes any local coordinate systems that exist in the simulation.
Coordinate System	Specifies the coordinate system (laboratory by default) for defining the origin and axis. The drop-down list of this property includes any local coordinate systems that exist in the simulation.

Working with Morphing

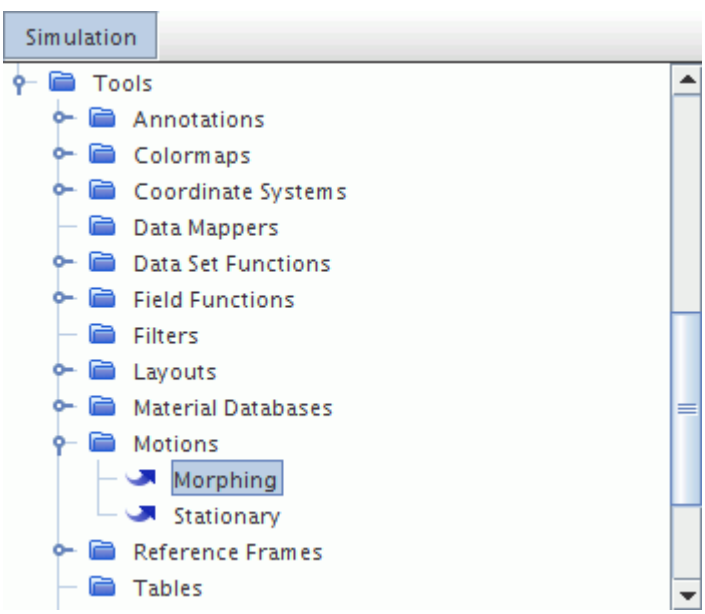
The Morphing motion in STAR-CCM+ redistributes mesh vertices in response to the movement of control points. The mesh morpher uses control points and their associated displacements to generate an interpolation field throughout the region which can then be used to displace the actual vertices of the mesh.

Each control point has an associated distance vector which specifies the displacement of the point within a single time-step.



When control points are collected from a boundary, it is not always necessary to create one control point per mesh vertex. A lower ratio can work equally well, and reduce the calculation time required. An automatic thinning algorithm is provided. This algorithm uses an estimated deformation of the mesh to reduce (or thin-out) the number of control vertices. Manual control is also provided through Thin Factor values on morphed boundaries. A Thin Factor of 1 means that all boundary mesh vertices are used, and a factor less than one results in a proportional reduction.

The node, **Morphing**, is subject to common menu items and has one property. When it is selected in the **Motion Specification** node for a region, this motion activates a solver and a morpher boundary condition. In turn the morpher boundary condition accesses a selection of boundary values that control the morphing process.



Morphing Model Properties

Morphing Order	The morpher can operate in two different modes: one region at a time, or one motion at a time. This property selects the mode to use.	
	Regionwise	Each region that is associated with this motion is morphed independently, one region at a

		time. Use this option, for example, in a fluid-solid case, where the solid region has a solid displacement motion and the fluid region has a morphing motion.
	Motionwise	All regions that are associated with this motion are morphed at the same time. This method is recommended for use in cases where internal interfaces are present between regions, if: • the floating morpher boundary method is applied to the boundaries on either side of the interface, and; • both regions are set to use the same morphing motion.
Linear Fitter	This option extracts a linear transformation from the prescribed motion, and uses this transformation to move vertices. The main multiquadric procedure is then used to treat any remaining motion. If the linear transformation accounts for most of the motion, activate the Morph From Zero property on the Morpher solver.	
	Activated	Linear motion is treated separately from multiquadric interpolation.
	Deactivated	All motion is treated in multiquadric interpolation.

Morphing Model Expert Properties

Automatic Thin-out	Applies an additional stage of control vertex thinning after the user-defined thin factors have been applied. Tick the Morph from Zero property on the Morpher solver when using this option.	
	Activated	Automated thinning is applied.
	Deactivated	No automated thinning is applied. Only user-defined thin factors are accounted for.
Automatic Thin-out CI	This property controls the target spacing between control points. If the value is smaller, the spacing is smaller resulting in denser control point distribution. If the value is larger, it coarser control point distribution.	

In cases with multiple regions, if a given region has no specified control points, the morphing solver is skipped in that region.

Note: In some cases involving interfaces, STAR-CCM+ must intersect the interfaces at the beginning of each time step. This reintersection adds to the computational cost. Further information is given in the section, Working with Interfaces and Morphing.

There is nothing to prevent morphing from being used in 2D and 2D axisymmetric cases. The axis boundary type is treated in the same way as a symmetry boundary type; the in-plane condition is applied to them both.

Alternative frames of reference can be used with the morphing motion. Choose alternative frames of reference in the *Reference Frame* property of the Motion Specification belonging to the region being morphed.

Contents:

Understanding the Morphing Procedure
Preparing for a Morpher Analysis
Working with Interfaces and Morphing
Understanding the Interaction of Control Planes and Control Points
Troubleshooting Negative Volume Cells
Morphing with Multiple Regions
What Is the Mesh Morpher Solver?
Using a Control Point List
Morpher Boundary Condition
Visualizing the Morpher Displacement
Morphing Motion Formulation
Morphing Field Functions
Motion Bibliography

See Also:

Harmonic Balance Flutter Boundary Values

Understanding the Morphing Procedure

Control points and their associated displacements form the underlying data that the morpher requires to generate an interpolation field. This interpolation field can be considered to act as a field function that is called for every mesh vertex in the region to find the new position for the vertex.

A high-level view of the morphing procedure is therefore:

1. The morpher collects control points and their known displacements from boundaries and control point tables within the region. User-specified thin factors are applied. If specified, the linear fitter is applied to update the current location of the vertices.
2. Automatic thinning (if activated) is applied to reduce the number of control points. Automatic thinning is activated by default.
3. After some initial preparation (based on Eqn. (140)), these control points and their known displacements are used to generate the interpolation field (by solving Eqn. (135) and Eqn. (138)).
4. The interpolation field (Eqn. (139)) is applied to all mesh vertices in the region.
5. Some final adjustments are made to mesh vertices on or near boundaries.

Control points can be specified using vertices that lie on mesh boundaries, and can also be specified using points that are read from a table. These two approaches are discussed below.

Control Points Taken from Mesh Boundaries

When the morpher motion is active for a region, every boundary (apart from boundaries of type symmetry plane) within the region receives a **Morpher Specification** node in its **Physics Conditions** folder. Additional nodes for morphing appear in the **Physics Values** folders, depending on the setting that is chosen for the **Morpher Specification** node. Boundaries of type symmetry plane have a single morphing-related node in their **Physics Values** folders.

The **Morpher Specification** node in the **Physics Conditions** node has a *Method* property that provides several different ways in which boundaries can interact with the morphing process. Deciding which method to use for each boundary is an important step in setting up a simulation for morphing.

In **Morpher Specification**, the morpher uses boundary vertices as control points for the following types of boundary conditions:

- **Fixed**
- **Displacement**
  - **Total**
  - **Incremental**
  - **Solid Stress**
  - **Rigid Body**

Often, the morpher problem can be solved correctly with a reduced set of control points. You control the proportion of mesh vertices that is used to create control points through the **Morpher Thinout Factor** node in the **Physics Values** node of those boundaries where the boundary condition is applied. The **Constraint** boundary conditions select the thinning factor automatically. **Floating** boundaries do not have control points.

Automatic Thin-Out

When automatic thin-out is activated on the morpher motion, the morpher thins out input vertices automatically to reduce the morpher execution time. The automatic thinner attempts to thin out control vertices by estimating the deformation of the grid due to morphing and placing control vertices efficiently according to this information. This option is active by default.

An automatic thin-out factor, CI, is provided to control the strength of the automatic thinning algorithm. Control vertices are thinned more aggressively as the factor increases. The suggested range of values is from 0.5 (mild thinning) to 5.0 (strong thinning).

Studies have shown that automatic thin-out works best when the *Morph From Zero* property is activated on the Morpher solver. Further improvements are possible when the *Linear Fitter* is also activated, as the linear fitter separates any rigid body motion of mesh vertices from their deformation motion.

Linear Fitter Option

The basic idea of the linear fitter is to use the control point data to make a least square fit to a linear transformation between new and old coordinates. Motion due to this linear fit is subtracted from the overall motion during multiquadric interpolation, and added back in afterwards to get the final motion.

A linear transformation includes all the rigid body modes:

- 3 degrees of freedom (DOF) for translation,
- 3 DOF for rotation,

and stretching modes:

- 6 DOF for stretching (that is, the strain in the fluid is a constant).

Therefore the linear fitter solves for a total of 12 unknown degrees of freedom.

Having deduced the linear transformation, the multiquadric interpolation is then used only for the portion of the overall motion that the linear transformation does not cover. The complete motion of the control points is the sum of the linear transformation of the points from their linear location, plus the deformation that is propagated from the multiquadric interpolation.

For example, if the control points were undergoing pure rigid body motion, then the fit to the linear transformation would be exact. The difference between actual motion and the linear transformation would be zero. In this case, the morpher would move the vertices exactly as if you had selected a rigid body motion model, and there would be no morphing due to deformation.

As another example, consider a case where the control points are on a body which is slightly flexible but which has predominately rigid body modes. The linear fit for this case is not exact, but the differences between that and the full motion is small. These small variations would be interpolated to the moving vertex and then the overall rigid body motion (the linear fit) would be added back in for the total morph.

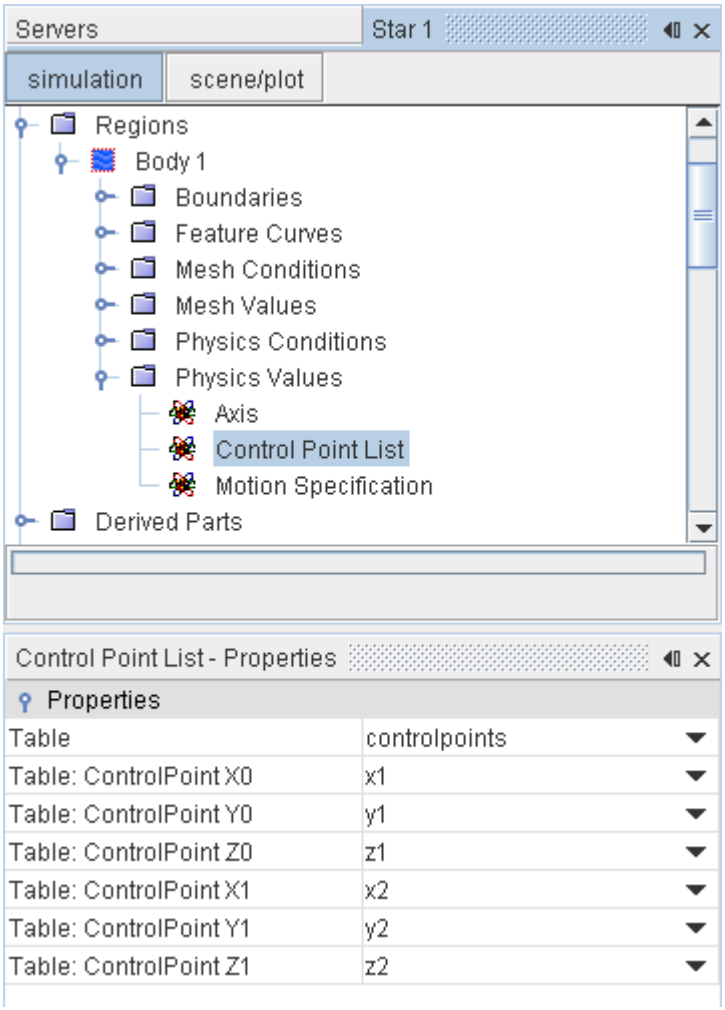
This linear fit option is therefore suited for cases where the rigid body motion usually governs the fluid grid motion.

Thinned Vertex Corrector

When the Vertex Thin Factor on a boundary is set to less than 1, the boundary vertices excluded due to thinning are initially moved according to the interpolation field in step 3 above. It is possible, however, that the final position of these thinned vertices is not precisely the position they would have obtained had they been moved according to the specified displacement on the boundary. A small error could result for other reasons even if vertex thinning is not used. To correct this error, a thinned vertex corrector algorithm is executed within step 4 of the morpher process. The corrector algorithm ensures that the thinned vertices receive the same displacement as the control vertices taken from their respective boundaries.

Control Points Taken from a Table

It is possible to define control points and their displacements directly in space for a morphing analysis. These control points are defined on the **Control Point List** node that are contained within the **Physics Values** folder of a region.

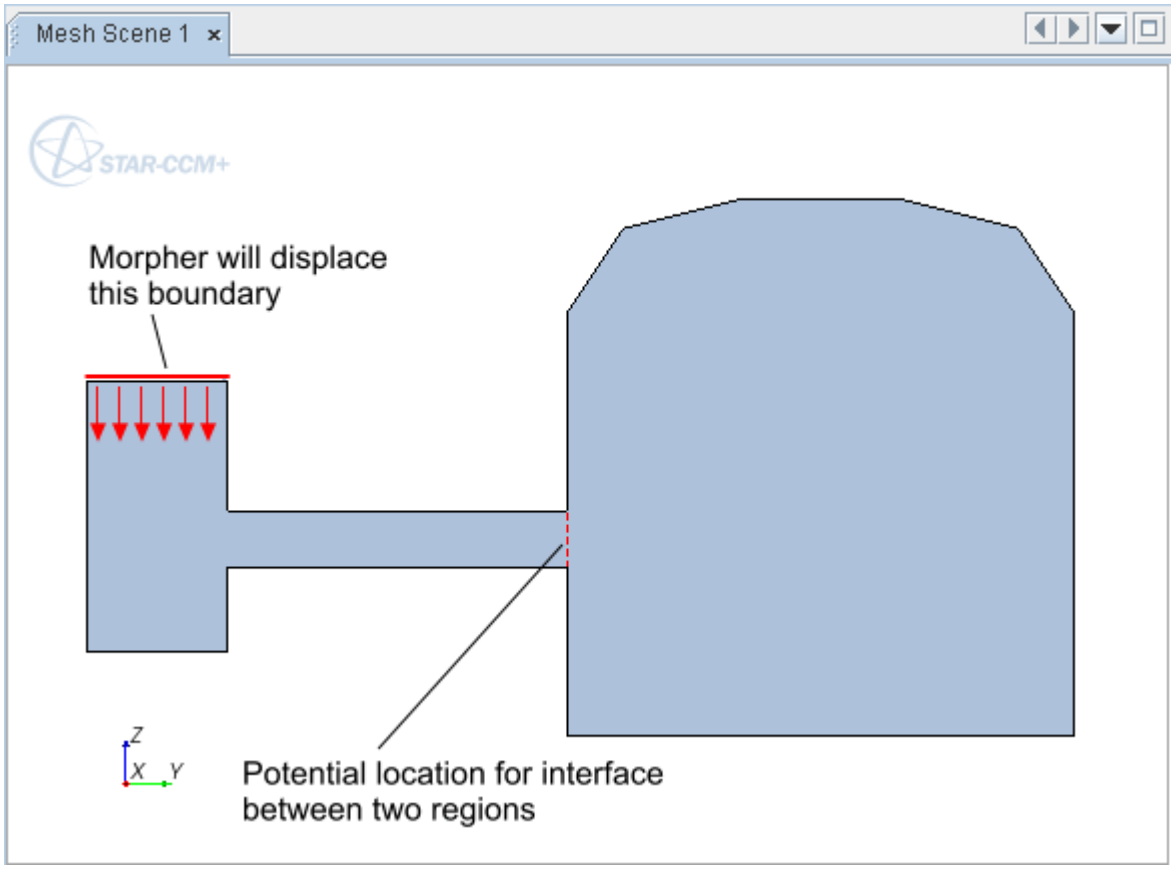


For further information, see the section on Using a Control Point List.

Preparing for a Morpher Analysis

The computational resource that is required to run the morpher is directly related to the number of control points present in the regions for which the morpher is active. For this reason, keep the number of control points as low as possible.

1. If the morphing only affects a portion of the mesh, consider creating a separate region for the portion that is morphed. Having a separate region means that control points are only required in the region being morphed. In the example below, morphing is restricted to the small plunger on the left, which means that the large vessel could be placed in its own region. The morphing motion would then only apply to the region containing the plunger.



2. In regions for which morphing is active, identify those boundaries that can be set to use the **Fixed Boundary Plane** or the **Boundary Plane** morpher constraint specification. These boundary methods can be dealt with far more efficiently in the morpher than is possible with the **Fixed** morpher specification method. Note, however, that these planar boundaries define an infinite plane that must not intersect the mesh at any other point. Further information is provided in the section, Understanding the Interaction of Control Planes and Control Points.
3. If a body is moving parallel to a planar surface, then apply the **Boundary Plane** morpher constraint specification to the planar surface.
4. The *Vertex Thin Factor* is set to 1.0 by default, but can be safely reduced to 0.25 in many cases. For boundaries that are at a distance from surfaces being actively morphed, a value of 0.1 would not be unreasonable. The automatic thin-out also helps to reduce the number of control points in cases where the *Vertex Thin Factor* has been set too high.
5. For the initial setup, do not reduce any vertex thin factors but rather activate the automatic thin-out algorithm on the Morpher motion node. This algorithm can provide sufficient thinning without user-specified intervention. Run a short trial case to see if the algorithm reduces the number of control vertices sufficiently.

Working with Interfaces and Morphing

It is important to apply the correct conditions at interfaces between the regions as failing to do so could result in the mesh being badly distorted. Two different scenarios can exist: internal interfaces and sliding interfaces.

Internal Interfaces Between Regions

For internal interfaces between regions, set the *Morphing Order* property of the **Morpher** motion node to **Motionwise**. Set the morpher boundary conditions on both sides of the interface to Floating if the interface is expected to morph in tandem with vertices surrounding the interface. This condition does not apply to fluid-solid cases where the solid region is using the solid displacements motion and the fluid region uses the morphing motion.

Deactivate the *Recalculate Interfaces* expert property on the Morpher Solver for internal interfaces between regions.

Sliding Interfaces

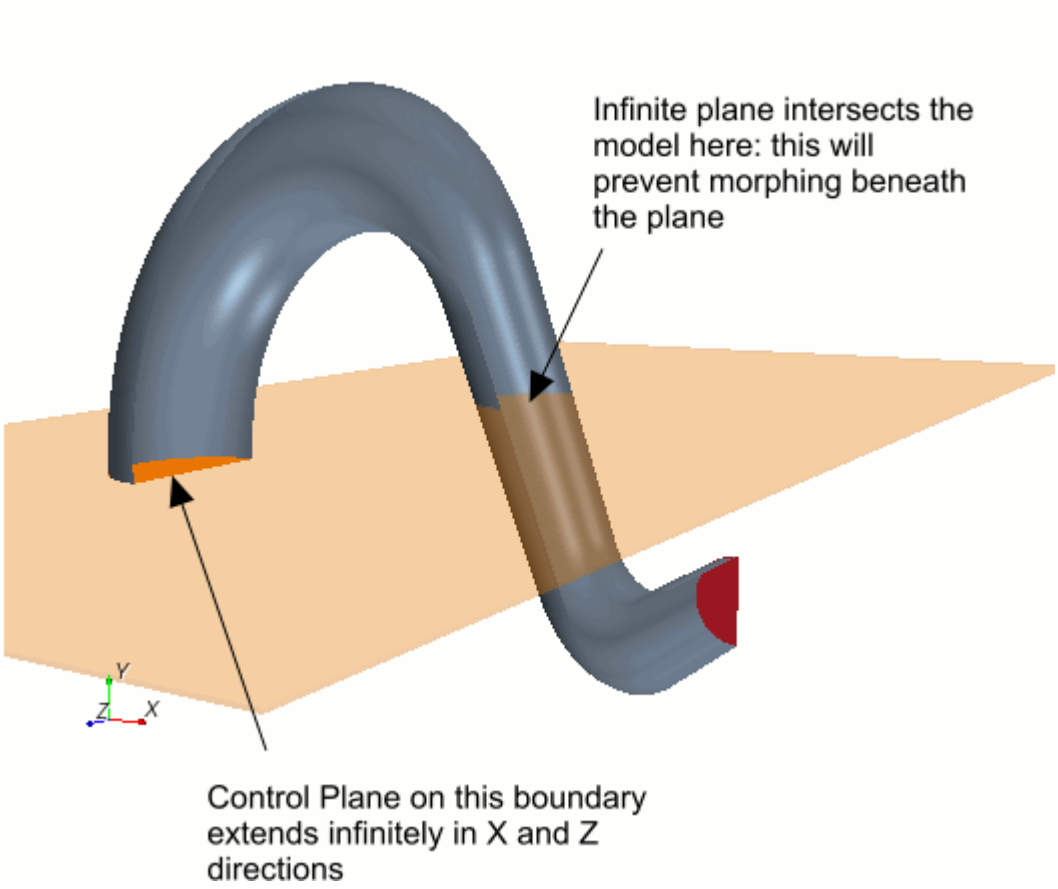
At such interfaces, the intention is for the mesh to move differently on the two sides, either to accommodate large motion, or to model a physical boundary. It is still necessary to ensure that the faces on the two sides lie close enough to be intersectable. Appropriate boundary conditions would be the In Plane condition, or the Displacement condition where the normal component on both sides is identical. The other components could be specified differently on the two sides.

Tick the *Recalculate Interfaces* expert property on the Morpher Solver for sliding interfaces.

Understanding the Interaction of Control Planes and Control Points

Control planes (in the **Fixed Boundary Plane** and **Boundary Plane** morpher constraint specification methods) work by damping out the displacements smoothly to zero over a specified zone of influence. For the **Fixed Boundary Plane** type, the damping applies to displacements in all directions; for the **Boundary Plane** type, the damping only applies to normal displacements.

Control planes extend infinitely in each direction tangential to the plane. The infinite extension means that it is possible to prevent morphing in other parts by setting a plane that intersects the model. The morpher only considers control points above the plane (the outward direction of the surface normal). Any control points on the opposite side of the plane are ignored. The following diagram illustrates such a case.



(This behavior can be avoided by using the **Part Surface** from the **Morpher Constraint Specification** or the **Boundary Plane** with the check box **Use Triangulated Plane** in **Morpher Plane Constraint Specification**.)

When a mesh vertex is located near or on a control plane, its final displacement due to morphing is calculated by multiplying the predicted displacement from Eqn. (139) by a damping function. The damping function gives zero displacements (or zero normal displacements for an In Plane type) at the plane, and smooth displacements for points away from the plane. The Damping Factor sets the strength of the damping function. It has units of length, and defines the distance normal to the plane within which damping is applied. As a guide, set the *Damping Factor* to a length that is about a quarter of the distance from the control plane to other boundaries.

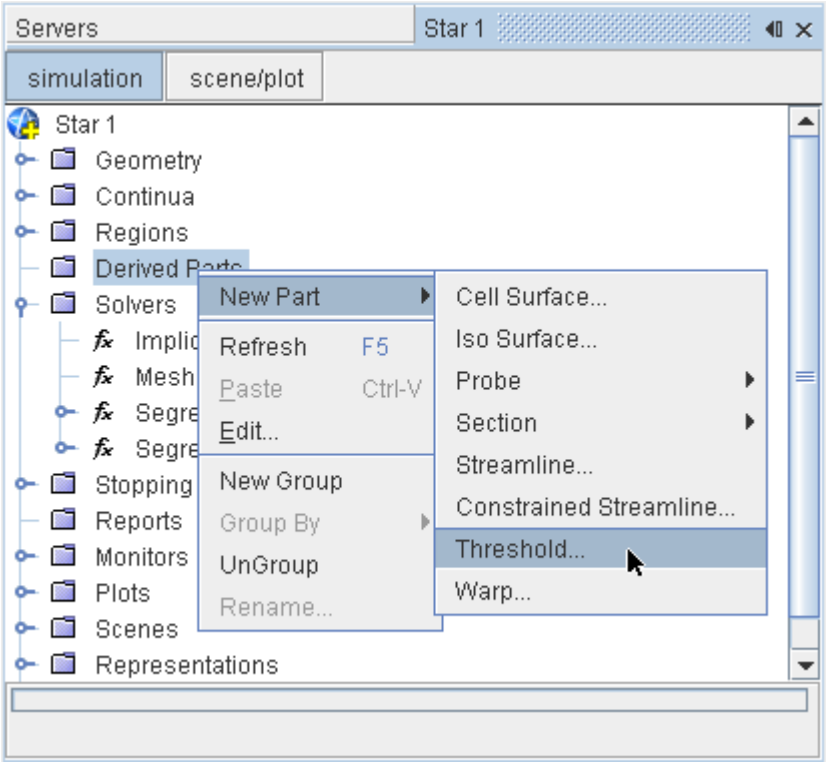
Two damping functions are provided to smooth vertices within the region the *Damping Factor* defines. These functions are identified in the formulation as C_0 continuous, Eqn. (141), and C_2 continuous, Eqn. (142). A plot is given of both functions.

Potential conflicts can occur at the intersection of boundaries that use control plane methods and boundaries that use methods which supply control points. An example would be the intersection of a boundary using the fixed plane method and a boundary using the displacement method. The prescribed displacement could potentially conflict with the fixed plane. For such cases, the morpher can be set to ignore control points that lie within a certain distance from the control plane. The distance is calculated as the *Ignore Factor* * *Damping Factor*. Both of these values are given as properties within the Morpher Plane Options boundary value.

Troubleshooting Negative Volume Cells

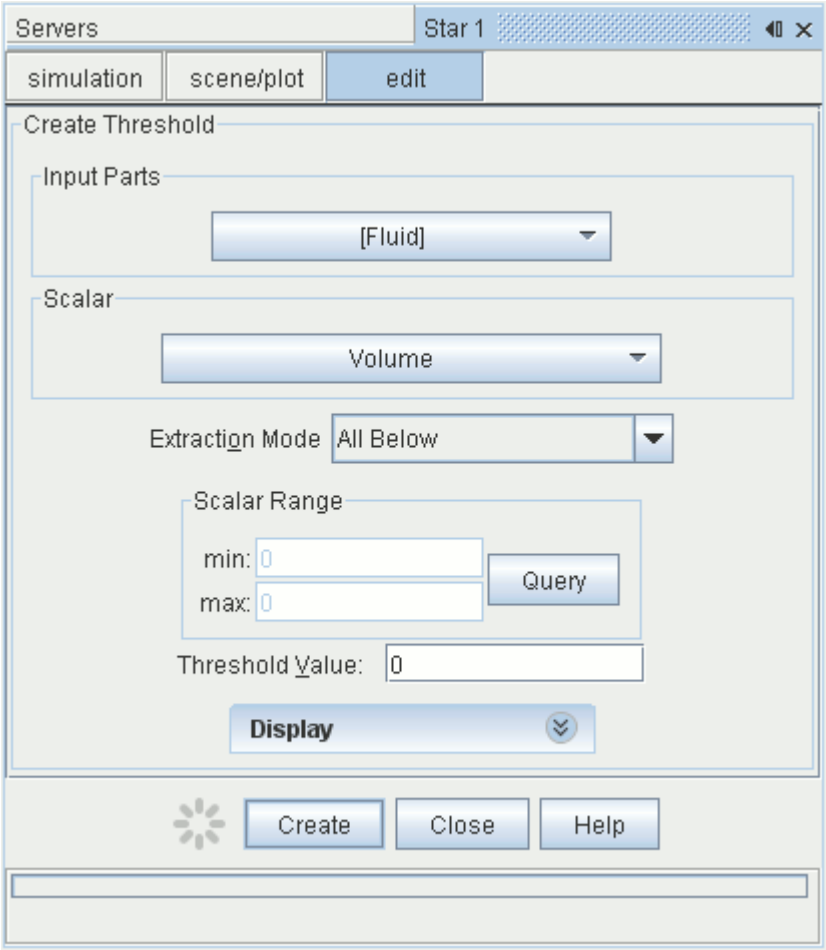
It is not uncommon to have reports of negative volume cells when setting up a morpher analysis. When negative volume cells are reported, the first step is to create a threshold derived part using the `Volume` field function for a maximum value of `0`.

1. Right-click the **Derived Parts** node and choose **New Part** > **Threshold...**

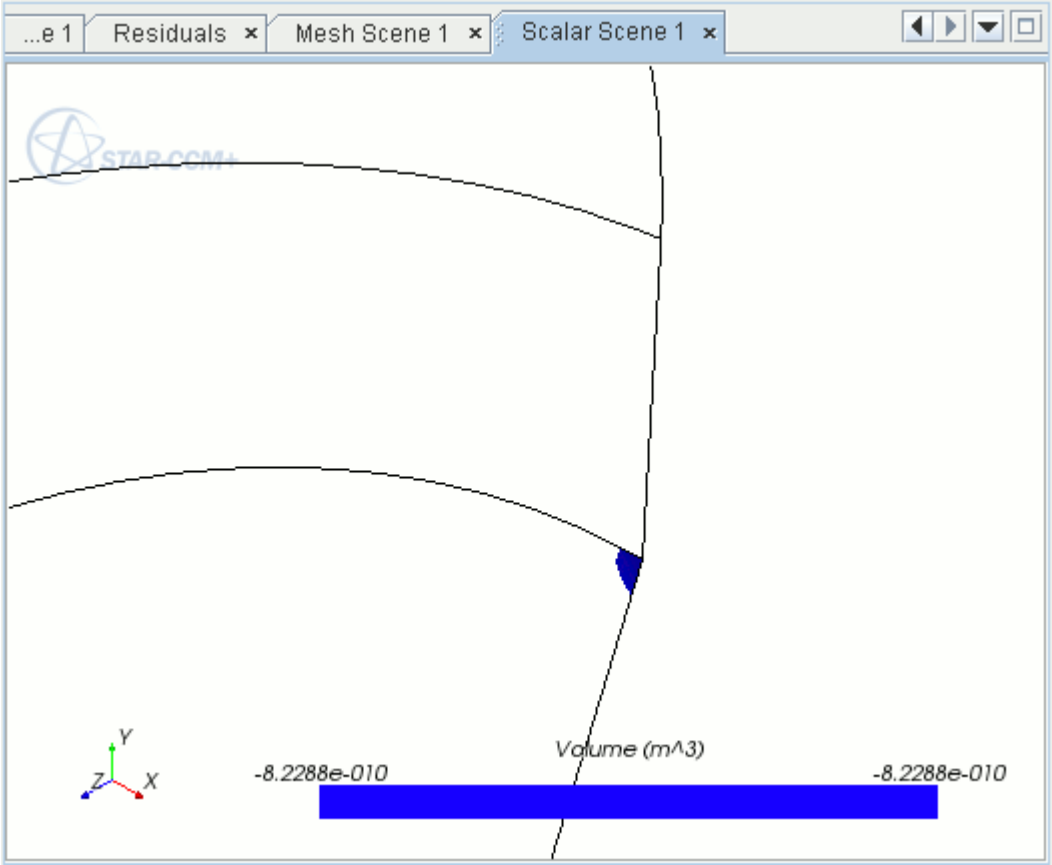


2. In the *Create Threshold* edit panel, do the following:
   - Ensure that the morphed region is selected in the **Input Parts** menu.
   - Choose `Volume` from the **Scalar** menu.
   - Set the **Extraction Mode** to **All Below**.
   - Leave the **Threshold Value** at its default setting of `0`.

   An example of a panel that is set with these properties is shown below. Click **Create** to generate the threshold part.



The part is added to a scalar scene to show the location of negative volume cells.



Suggestions for Correcting the Model

1. If the negative volume cells are close to a moving body, increase the *Vertex Thin Factor* for the boundary on the body. If possible, split the moving body boundary and apply a larger *Vertex Thin Factor* just for the area near the negative volume cells. This arrangement allows you to use a lower value (0.25 or below) for the remainder of the boundary.
2. If the negative volume cells are close to a fixed boundary, follow the same procedure as in (1).
3. If the negative volume cells are close to a fixed plane boundary, check to see if compression normal to the boundary is causing the negative volumes. If so, increase the damping factor so that a greater number of cell layers are accounted for in the damping zone. If the negative volume cells are being stretched parallel to the boundary, change the boundary to the in-plane type instead.
4. If the negative volume cells are close to an in-plane boundary, increase the damping factor.

Morphing with Multiple Regions

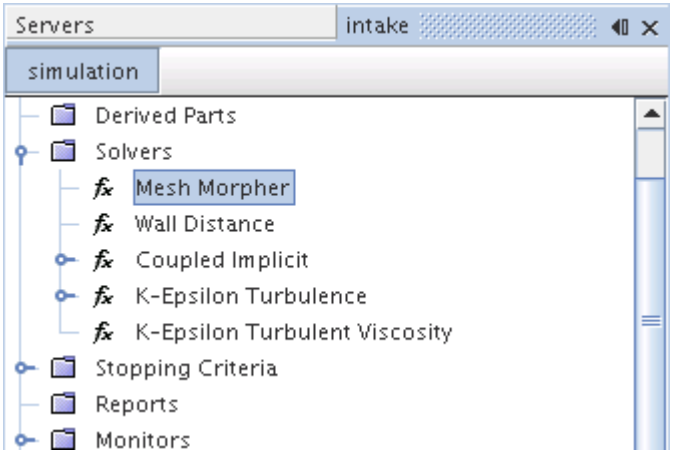
To reduce computational costs, the domain can be divided into regions. In any given region, if there are no control points, the morpher is automatically turned off.

Normally, the **Fixed** morpher boundary method creates control points. If, however, there are no other control points, these **Fixed** points are discarded and the morpher solution is skipped in that region.

Multiple regions can also be used in cases where grossly different behavior is desired in each region (for example, to preserve some mesh features like boundary extrusion layers). In such cases, the Interface Boundaries must be given identical boundary conditions on the two adjacent regions.

What Is the Mesh Morpher Solver?

The **Mesh Morpher** solver is designed to control the morphing motion calculations with specialized properties. It becomes available when a region has the Morphing motion selected.



Mesh Morpher Solver Properties

Verbosity	Provides additional output for this value while the simulation is running. This property is useful for debugging problems when they occur.	
	None	Leaves output at its default.
	Low	Adds messages as to which region is morphed.
	High	Generates precise equation information about the morphing process as the solver converges.
Morpher Tolerance	Non-dimensional parameter that is used to specify acceptable accuracy in morphing. This parameter is multiplied by the longest edge length of the bounding box enclosing the morphing region. If the mesh to be morphed contains extremely long and thin prism layers cells, you are advised to reduce the **Morpher Tolerance** to 10^{-7} to 10^{-9}. This setting can avoid the occurrence of negative cell volumes during morphing.	
Initialization needs morphing	Selecting this option causes the morpher to run when the simulation is initialized. Use this option when the control point specifications at time zero define an initial deformation of the mesh.	

	Activated	The morpher runs at time zero.
	Deactivated	The morpher does not run at time zero, but runs during the first time-step to apply deformations at time equal to Δt .

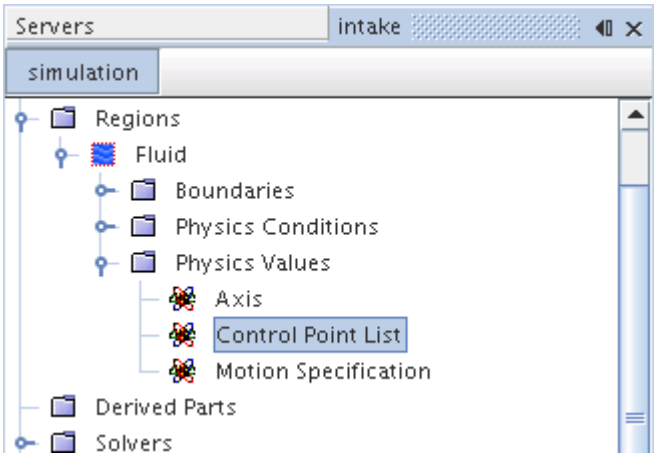
Mesh Morpher Solver Expert Properties

Temporary Storage Retained	Specifies whether to retain temporary storage at the end of the iteration, or not. Examples include the ap coefficients, the residual, the corrections, the reconstruction gradients, and the cell gradients. These quantities become available as field functions during subsequent iterations.	
	Activated	Retains temporary storage.
	Deactivated	Does not retain temporary storage.
Morph At Inner Iterations	Repeats mesh morphing at each inner iteration during the time-step. This action is only meaningful for cases where some boundary condition depends on the solution; it is off by default to reduce computational costs. For a **DFBI morphing** motion, this option is activated by default and cannot be deactivated. DFBI uses an implicit algorithm for coupling fluid flow and body motion, which requires a morphing update at each inner iteration.	
	Activated	Repeats mesh morphing at each inner iteration.
	Deactivated	Leaves off this option.
Morph From Zero	Begins each mesh morphing from the same (zero time) positions. This behavior is helpful for periodic body motions, to ensure that the mesh is also periodic. It is also recommended when automatic thinning is applied.	
	Activated	Activates this option.
	Deactivated	Leaves off this option.
Ignore Gridflux Terms	When this option is activated, the mesh still moves but the “grid” flux is not included in the transport equations. This option provides more stability when marching towards a steady solution with an unsteady solver. It is also useful when you are simply adjusting the shape of the region without solving the transport equations.	
	Activated	Activates this option.
	Deactivated	Leaves off this option.

Recompute Interfaces	When activated, this causes STAR-CCM+ to recompute all interfaces at each time-step. Use this option only if the morphing changes the mesh in different ways on opposite sides of the interfaces.	
	Activated	Activates this option.
	Deactivated	Leaves off this option.
Skip Thinned Vertex Corrector	When this option is activated, it prevents STAR-CCM+ from attempting to correct thinned vertices.	
	Activated	Activates this option.
	Deactivated	Leaves off this option.
Solver Frozen	Prevents or allows the solver from updating the solution while iterating. Do not activate this feature as your first choice, as it is primarily a “debugging” tool and not supported for normal simulation work.	
	Activated	The solver does not update the solution.
	Deactivated	The solver updates the solution.

Using a Control Point List

The Control Point List is a region value, added by the Morphing motion or the Harmonic Balance Flutter motion.



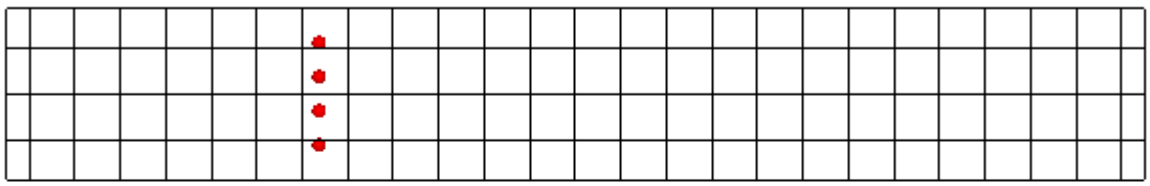
The Control Point List can be used to specify control points and their associated displacements without reference to any boundary. The list is defined as a table containing six columns. The first three columns specify the cartesian components of the initial position of the control points. The second set of three columns specifies the cartesian components of the target position of the control points. An example fragment of such a table is given as follows:

x1	y1	z1	x2	y2	z2
0.55	0.06	0.06	0.56	0.06	0.06

x1	y1	z1	x2	y2	z2
0.55	0.06	0.12	0.56	0.06	0.12
0.55	0.06	0.18	0.56	0.06	0.18

This table is defining a line of three control points (spaced along the Z-axis) whose displacement is 0.01 units in the X direction. The initial position of each control point is set to [x1,y1,z1] at the beginning of each time step. The target position is set to [x2,y2,z2] in every time step. This behavior is illustrated below. The mesh vertices at the end of the simulation are displaced a considerable proportion, but the control points based on the table remain in their original location.

(1) Initial grid (before morphing)



(2) Grid after several time steps in which morphing was driven by the Control Point List

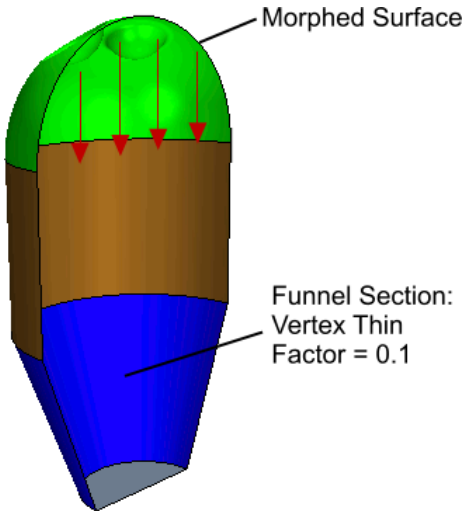
Control Point List Properties

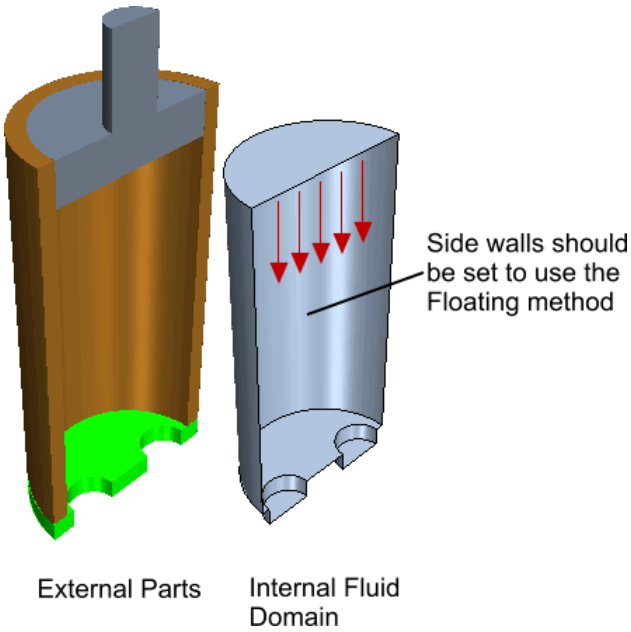
Table	Selects a table from the **Tools** > **Tables** sub-nodes.
Table: ControlPoint X0	Selects the column within the table for the initial x-coordinate of the control point.
Table: ControlPoint Y0	Selects the column within the table for the initial y-coordinate of the control point.
Table: ControlPoint Z0	Selects the column within the table for the initial z-coordinate of the control point.
Table: ControlPoint X1	Selects the column within the table for the final x-coordinate of the control point.
Table: ControlPoint Y1	Selects the column within the table for the final y-coordinate of the control point.
Table: ControlPoint Z1	Selects the column within the table for the final z-coordinate of the control point.

Morpher Boundary Condition Reference

Morpher Specification

Controls how the morphing for a boundary is performed. Activated when the Morpher motion is selected.

Specification	Corresponding Value Nodes
Fixed The vertices of a boundary are fixed in space. The Fixed morpher boundary method supplies control points for use in calculating the interpolation field. In this case, however, the displacement that is associated with the control points is zero. For non-planar boundaries close to the zone of influence, keep the *Vertex Thin Factor* close to 1.0, as the shape of the **Fixed** boundary could otherwise be distorted. For planar boundaries far from the zone of influence, the **Vertex Thinout Factor** can be reduced. In the example below, the *Vertex Thin Factor* on the walls of the funnel section has been set to 0.1.  For a planar boundary where the infinite plane of the boundary surface does not intersect with the model geometry at any other point, use the **Fixed Boundary Plane** morpher constraint specification method instead.	**Morpher Thinout Factor** Reduces the number of vertices that are used as control points in order to reduce computational costs. Specifies the *Vertex Thinout Factor* to use only this fraction of boundary vertices as control vertices. Vertices that are not chosen are given the applied boundary condition (so that the boundary movement is correct).
Floating Vertices on the boundary move according to the interpolation of the displacement vector field. With the floating method, mesh vertices on the boundary are not constrained. They move in response to the interpolation	None

field created by the morpher. An example of where this morpher boundary type could be used is for the side walls of a piston undergoing one-dimensional expansion or contraction, as shown in the illustration below.  	
Displacement The boundary vertices are moved by a specified distance. There are several types of displacement types available. Adds the **Morpher Displacement Specification** node.	**Morpher Thinout Factor** Reduces the number of vertices that are used as control points in order to reduce computational costs. Specifies the *Vertex Thin Factor* to use only this fraction of boundary vertices as control vertices. Vertices that are not chosen are given the applied boundary condition (so that the boundary movement is correct).
Constraint Boundary vertices are forced to stay within a plane or a triangulated surface.	None

Morpher Displacement Specification

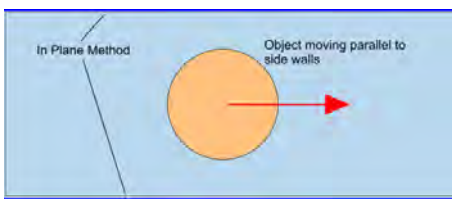
Controls which type of displacement is used to morph the boundary vertices.

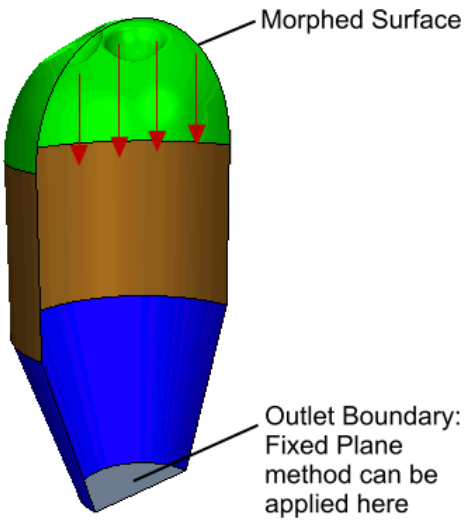
Specification	Corresponding Value Nodes
Incremental Linear displacement of boundary vertices relative to the vertex positions of the previous time-step or iteration. The boundary target vertex coordinates are computed as: `target_coordinate = current_coordinate + incremental_displacement`.	**Morpher Incremental Linear Displacement** Specified as a vector profile with respect to *Coordinate System*. **Morpher Rigid Boundary Motion** Allows you to combine the incremental linear displacement morphing motion of this boundary with a rigid motion. All types of rigid motions are compatible. Selects the *Rigid Motion* that you define under **Tools** > **Motion**.
Total Linear total displacement per time-step of boundary vertices relative to the initial position of the boundary vertices at time zero. The boundary target vertex coordinates are computed as: `target_coordinate = saved_coordinate + total_displacement`.	**Morpher Total Linear Displacement** Specified as a vector profile with respect to *Coordinate System*.
Grid Velocity Vertices on this boundary are moved using the specified velocity. The resulting displacement is proportional to the time step used for the simulation. The boundary target vertex coordinates are computed as `target_coordinate = current_coordinate + grid_velocity * time_step`.	**Grid Velocity** Input as a vector profile. If a user coded profile is used to set this value, the profile must be of type region, rather than type boundary. In this instance, however, only the boundary vertices are used.
Internal The displacement field is computed internally by other STAR-CCM+ models and is treated as an incremental displacement. This method is used by the **Fluid Film Solidification** model (for example for wing de-icing simulations) and the **Viscous Flow** model for modeling free-surface tracking (for example for extrusion simulations).	None
Solid Stress Used by the STAR-CCM+ **Solid Stress** model. The **Solid Stress** model computes a vector displacement field. This field is mapped from faces to vertices and it is applied as incremental displacement.	None

Coordinate Offset Specifies an incremental displacement with respect to any type of coordinate system, for example, a cylindrical coordinate system. For more information, see What is the Coordinate Offset.	**Coordinate Offset** You specify the coordinate offset value as a vector profile with respect to *Coordinate System*.

Morpher Constraint Specification

Specifies the type of constraint that is applied to the boundary.

Constraint	Corresponding Value Nodes
Boundary Plane The boundary vertices can move on the boundary, but they remain in the plane of the boundary. Results in the creation of an infinite plane. The boundary vertices are permitted to move within the plane of the boundary itself. This behavior is appropriate for simulations similar to the one shown below.  Adds the **Morpher Plane Constraint Specification** node.	None
Fixed Boundary Plane Mesh vertices lying on a fixed boundary plane are not moved in response to the interpolation field calculated by the morpher. This boundary differs from the fixed boundary in that no control points are taken from the boundary this method is applied to. Ultimately this means that a more computationally efficient approach can be taken. Note, however, that applying this boundary condition results in the creation of an infinite plane that could potentially intersect other parts of the model geometry.	**Morpher Plane Options** Determines the morpher behaviour for points near this plane. Starting at a distance that you specify as *Damping Factor*, the displacements at the boundary are damped smoothly to zero. The *Damping Factor* corresponds to L_p in Eqn. (143). The default value of 0.0 means that the distances are calculated automatically. The *Continuity Level* specifies which damping function to use in smoothing vertex displacements approaching the plane. For more information, see Boundary Plane Conditions.

The outlet of the funnel section below would be a suitable boundary on which to apply the **Fixed Plane** method.  	The *Ignore Factor* is multiplied by the *Damping Factor* to calculate a finite distance. Control points closer than the specified distance to this plane are given special treatment. A value of zero is acceptable.
STL Surface (deprecated) The boundary vertices slide along a triangulated STL surface that is used as guide surface. The STL surface is imported as an external file. **Note:** This constraint is available for simulation files that used this option in versions of STAR-CCM+ prior to 12.02. From STAR-CCM+ 12.02 onwards, use the **Part Surface** constraint instead.	**STL File Settings** Through *File Path*, you import the binary STL file that is located anywhere on your file system. **Morpher Rigid Boundary Motion** Allows you to combine the **STL Surface** constrained morphing motion of this boundary with a rigid motion. All types of rigid motions are compatible. Selects the *Rigid Motion* that you define under **Tools** > **Motion**.
Initial Boundary The boundary vertices slide along the original boundary as guide surface.	None
Part Surface The boundary vertices slide along a part surface that is used as guide surface.	**Part Surface** You select the surface of a geometry part. The geometry part can originate from the following sources: • Conversion of 3D-CAD bodies • Imported surfaces • Creation of shape geometries within STAR-CCM+

- Results of mesh operations

Morpher Rigid Boundary Motion

Allows you to combine the **Part Surface** constrained morphing motion of this boundary with a rigid motion. All types of rigid motions are compatible. Selects the *Rigid Motion* that you define under **Tools** > **Motion**.

Morpher Plane Constraint Specification

Defines a plane for constrained boundary vertices to slide on. STAR-CCM+ implements two possibilities to create such a plane:

- by using the equation of the plane
- by triangulating the plane

Use Triangulated Plane	Corresponding Value Nodes
When **Activated**, the plane is defined by finding an equation of the plane for the boundary by using a least-square fitting method. (The equation of the plane is not further used for constraining the boundary vertices.) The plane is then triangulated. For constraining the boundary vertices to the boundary plane, the following steps are performed: 1. The displacement is calculated and all vertices are morphed. The vertices on the boundary are also displaced and treated as floating. 2. The displaced boundary vertices are projected back onto the closest triangle of the boundary plane. 3. The whole region is morphed one more time, but now the sliding boundary is treated as a displacement boundary that uses the projected coordinates as target coordinates. The three steps are performed inside STAR-CCM+ and you observe only the final result. This procedure is more accurate than using the non-triangulating approach (*Use Triangulated Plane* is deactivated), but it is computationally more expensive, because of the second morphing step. The boundary plane can be moved additionally by a rigid body motion.	**Morpher Rigid Boundary Motion** Allows you to combine the **Triangulated Plane** constrained morphing motion of this boundary with a rigid motion. All types of rigid motions are compatible. Selects the *Rigid Motion* that you define under **Tools** > **Motion**.
When **Deactivated**, an infinite plane is defined by finding an equation of the plane for the boundary by a least-square fitting method. For constraining the boundary vertices to the boundary plane, the following steps are performed:	**Morpher Plane Options** Determines the morpher behaviour for points near this plane.

1. The displacement is calculated and all vertices are morphed.
2. The normal component of the displacement is damped. The damping magnitude is related to the distance of the vertices to the boundary plane. The closer the vertices to the boundary plane, the higher the damping. At the boundary, the displacement of the vertices is damped to zero. Through this damping, the boundary vertices are projected onto the boundary plane.

The damping options can be adjusted.

Starting at a distance that you specify as *Damping Factor*, the displacements at the boundary are damped smoothly to zero. The *Damping Factor* corresponds to L_p in Eqn. (143). The default value of 0.0 means that the distances are calculated automatically.

The *Continuity Level* specifies which damping function to use in smoothing vertex displacements approaching the plane. For more information, see Boundary Plane Conditions.

The *Ignore Factor* is multiplied by the *Damping Factor* to calculate a finite distance. Control points closer than the specified distance to this plane are given special treatment. A value of zero is acceptable.

Visualizing the Morpher Displacement

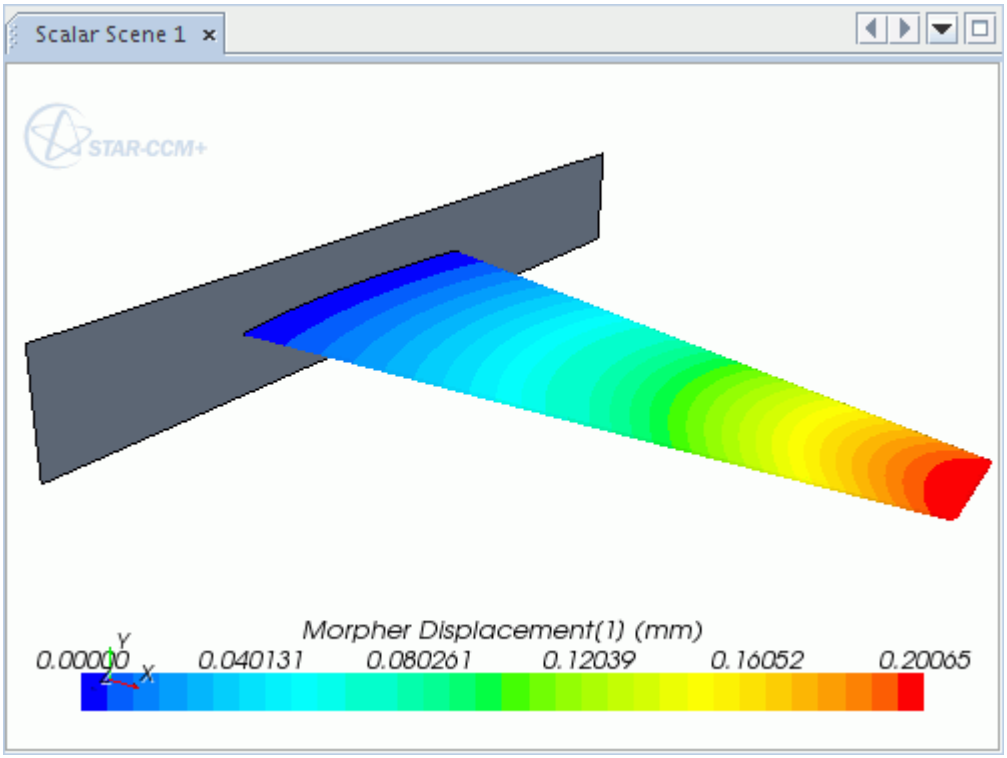
You can use the Morpher Displacement field function to analyze how the morpher affects the volume mesh and compare its values with expected results.

Note: Morpher displacement is based upon values at vertices, not cell centers.

To visualize the morpher displacement:

1. In a scalar scene, select the **Displayers** > **Scalar 1** > **Parts** node and set *Parts* to the boundary on which you want to visualize morpher displacement.
2. Select the **Scalar Field** node and set *Function* to one of the following:
   - **Morpher Displacement** > **Magnitude**: The displacement magnitude of all components.
   - **Morpher Displacement** > **0**: The x-component of displacement.
   - **Morpher Displacement** > **1**: The y-component of displacement.
   - **Morpher Displacement** > **2**: The z-component of displacement.

The following example shows a scalar scene that visualizes the y-component of morpher displacement.



Morphing Motion Formulation

In STAR-CCM+, a set of control vertices defines the initial movement that the morpher imposes on the mesh. Each control vertex is associated with a displacement vector that the morpher uses to move nearby vertices.

Using multiquadric interpolation theory [13], these specified displacements are used to generate an interpolation field that applies to the entire region for which the morpher motion applies. When this field has been generated, the morpher uses it to move mesh vertices to their new position.

To generate the interpolation field, a system of equations is created using the control vertices and their specified displacements, where the known displacement $\mathbf{d}'_i$ of each control vertex i is expanded as

$$\mathbf{d}'_i = \sum_{j=1}^{N} f_{b,j}\left(r_{ij}\right)\boldsymbol{\lambda}_j + \boldsymbol{\alpha} \tag{135}$$

in which $f_{b,j}\left(r_{ij}\right)$ is a radial basis function of the form

$$f_{b,j}\left(r_{ij}\right) = \sqrt{r_{ij}^{\ 2} + c_j^{\ 2}} \tag{136}$$

and r_{ij} is the magnitude of the distance between two vertices:

$$r_{ij} = \left|\mathbf{x}_i - \mathbf{x}_j\right| \tag{137}$$

In these equations, $\boldsymbol{\lambda}_j$ is the expansion coefficient, $\mathbf{x}_i$ is the position of vertex i, N is the number of control vertices, and c_j is the basis constant. The basis constant is set to zero in STAR-CCM+. The constant vector $\boldsymbol{\alpha}$ is used to satisfy the additional constraint that

$$\sum_{j=1}^{N} \boldsymbol{\lambda}_j = 0 \tag{138}$$

This additional constraint bounds the expansion for large $\mathbf{x}$. The system that is described in Eqn. (135) is non-singular if $\mathbf{x}_i \neq \mathbf{x}_j$ for all $i \neq j$.

Equations Eqn. (135) and Eqn. (138) are solved to give the Cartesian components of all $\boldsymbol{\lambda}_j$ and the components of constant vector $\boldsymbol{\alpha}$. This results in the desired interpolation field:

$$\mathbf{d}'(\mathbf{x}) = \sum_{j=1}^{N} f_{b,j}(r)\boldsymbol{\lambda}_j + \boldsymbol{\alpha} \tag{139}$$

where $f_{b,j}(r)$ applies at all vertices in the mesh, that is,

$$f_{b,j}(r) = \sqrt{r^2 + c_j^{\,2}}$$
$$r = \left|\mathbf{x}\text{-}\mathbf{x}_j\right|$$

This equation can now be used to move mesh vertices by the calculated displacement $\mathbf{d}'$.

The matrices that are involved in equations Eqn. (135) and Eqn. (139) are dense, and a straightforward matrix operation would require $\mathrm{O}(N^2)$ operations. Fortunately, an $\mathrm{O}(N\log(N))$ method has been developed to solve the linear equation Eqn. (135) and evaluate equation Eqn. (139) at many vertex locations. The method involves a conjugate gradient method with a special preconditioner ([14], [15]) and a fast multipole method [16] to accelerate the matrix-vector multiplication.

Boundary Plane Conditions

Two plane boundaries are provided for use with the morpher motion: boundary plane and fixed boundary plane. For the fixed boundary plane condition, the vertices which lie on the specified plane are fixed, while for the boundary plane condition they are permitted to move within the plane itself. An effect similar to the fixed boundary plane condition can be realized by using control vertices but the fixed boundary plane condition itself can be more efficient. The movement of vertices near the constraint planes is such that they join smoothly with the movement of vertices on the plane. This movement near the constraint planes can be controlled using the damping distance L_p that is described below.

These constraints are imposed before solving Eqn. (135). In preparation, user-specified displacements $\mathbf{d}_i$ are transformed to $\mathbf{d}'_i$ by solving

$$\mathbf{d}_i = \prod_{p=1}^{N_p} \left[s_p \mathbf{n}_p \mathbf{n}_p^T + t_p\left(1 - \mathbf{n}_p \mathbf{n}_p^T\right)\right] \mathbf{d}'_i \tag{140}$$

where $\mathbf{n}_p$ is the normal of the p-th plane and N_p is the number of planes. The displacements $\mathbf{d}'_i$ are transformed back to $\mathbf{d}_i$ together with other mesh vertices at the end of the morphing process. s_p and t_p are functions that depend on the plane type and the continuity condition you choose. These functions are given as follows:

	C_0 continuous		C_2 continuous	
	s_p	t_p	s_p	t_p
Fixed plane	$f_0(l)$	$f_0(l)$	$f_2(l)$	$f_2(l)$
In-plane	$f_0(l)$	1	$f_2(l)$	1

Here:

$$f_0(l) = \begin{cases} 1 & ;\ l \geq 1 \\ 1-(1-l)^3; & 1 > l \geq 0 \\ 0 & ;\ l < 0 \end{cases} \tag{141}$$

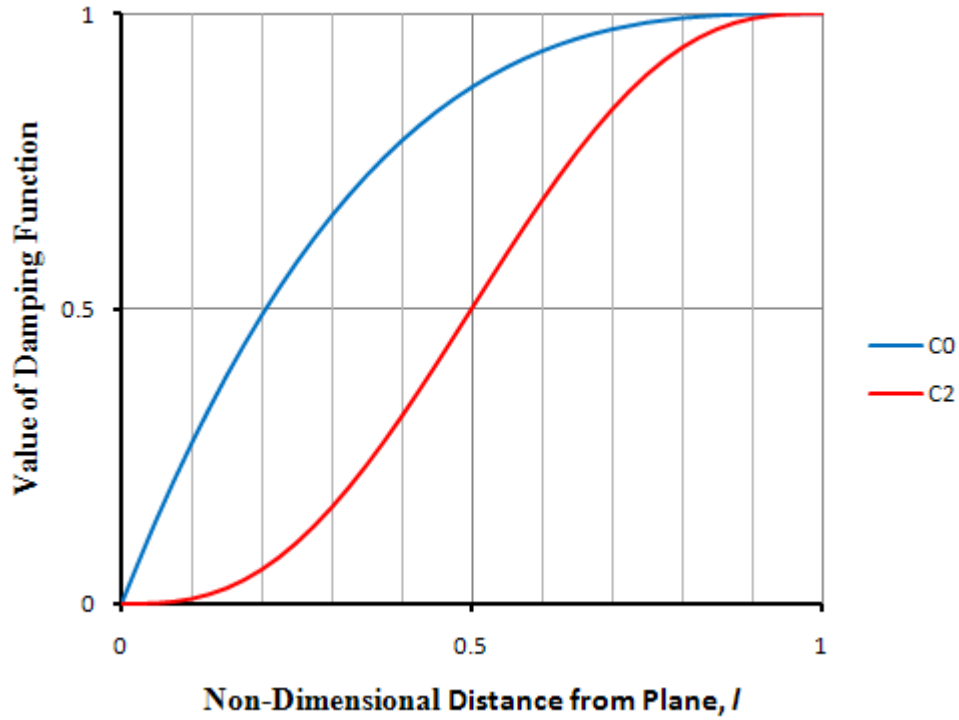
$$f_2(l) = \begin{cases} 1 & ;\ l \geq 1 \\ l^3\ \left(10-15l+6l^2\right); & 1 > l \geq 0 \\ 0 & ;\ l < 0 \end{cases} \tag{142}$$

in which

$$l = \frac{\mathbf{n}_p\left(\mathbf{x}-\mathbf{x}_{0,p}\right)}{L_p} \tag{143}$$

is the ratio of the signed distance from the p-th plane to $\mathbf{x}_i$ and the damping distance L_p. $\mathbf{x}_{0,p}$ is a position vector on the p-th plane and L_p is either entered as a finite distance or calculated automatically as $\left(L_{min}+L_{max}\right)/2$. L_{min} and L_{max} are respectively the distances from the plane to the nearest and farthest control vertices on the active side of the plane.

Functionsf_0 andf_2 in the range from 0 to 1 are plotted below for comparison. A continuity level of C_2 results in a smooth transition.



Solution Steps

The steps that are involved in the morpher stage within a time step are:

- The control vertex displacements $\mathbf{d}_i$ are transformed to $\mathbf{d}'_i$ according to Eqn. (140).
- Eqn. (135) is solved with Eqn. (138) to provide an interpolation field.
- The displacement vectors are transformed from $\mathbf{d}'_i$ to $\mathbf{d}_i$ using Eqn. (140).

Morphing Field Functions

The following primitive field function is made available to the simulation when the morpher motion model is used. This field function has a typical set of base properties.

SavedCoord	This field contains the coordinates before the morphing.
Morpher Displacement	Vector value of the distance by which the morpher has moved the mesh vertices (not face centers) in the current time-step. As this field is a vector field function, you can plot the x-, y-, z-components, and magnitude.

Motion Bibliography

[13] *Hardy, R.L. 1990. "Theory and applications of the multiquadric-biharmonic method", Comput. Math. Applic., 19, pp. 163-208.*

[14] *Faul, A.C., Goodsell, G. and Powell, M.J.D. 2005. "A Krylov subspace algorithm for multiquadric interpolation in many dimensions.", IMA Journal of Numerical Analysis, 25, pp. 1-24.*

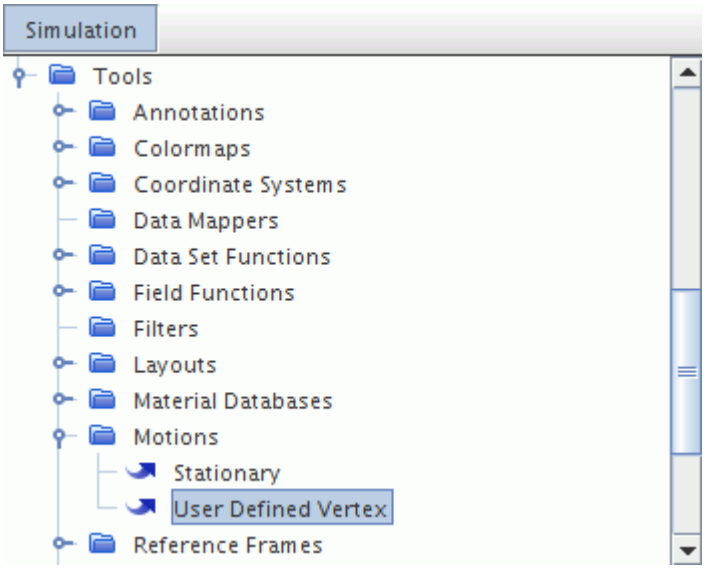
[15] *Gumerov, N.A. and Duraiswami, R. 2007. "Fast radial basis function interpolation via preconditioned Krylov iteration.", SIAM Journal on Scientific Computing, 29, pp. 1876-1899.*

[16] *Beatson, R.K., Powell, M.J.D. and Tan, A.M. 2006. "Fast evaluation of polyharmonic splines in three dimensions", DAMTP 2006/NA03.*

Working with User-Defined Vertex Motion

The User-Defined Vertex motion allows you to set a displacement or velocity for every vertex position in the simulation. It is typically intended for use in cases where the other motions cannot give the precise control that is required over the mesh movement.

The **User-Defined Vertex** motion node represents this type of motion. You can add this node using the right-click menu of the **Motions** manager.



When you select this motion for the region you wish to move, an additional condition, the **User-Defined Vertex Option**, is added to the **Physics Conditions** folder of the region. Use this node to set the quantity by which the motion is defined.

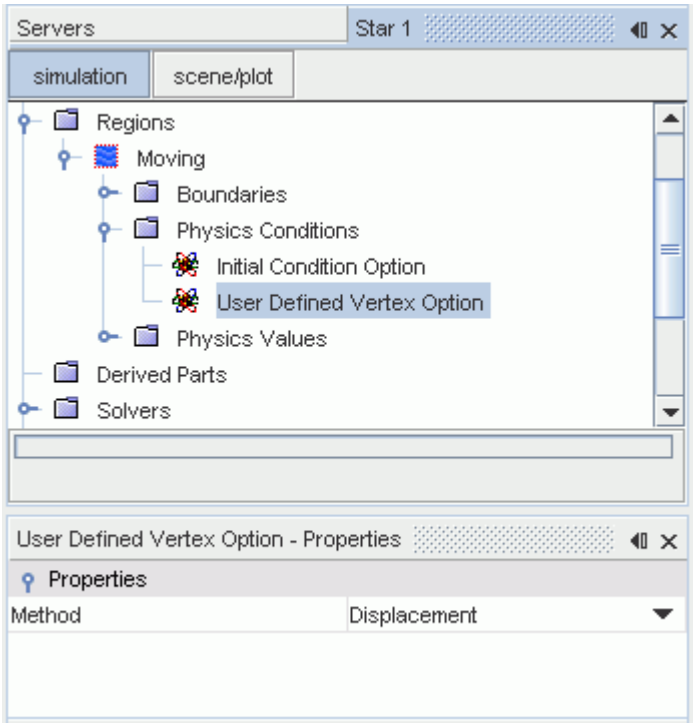
Contents:

Setting the Quantity for User-Defined Motion

What Is the User-Defined Vertex Motion Solver?
An Example of User Coding

Setting the Quantity for User-Defined Motion

User-defined motion can be specified in terms of displacement, grid velocity, new coordinates or coordinate offset. The choice between these quantities is made using the *Method* property of the **User-Defined Vertex Option** node.



Method	Specifies the method by which the mesh motion is defined.	
	Fixed	No vertex motion. Use this method when solving for initial flow fields, particularly in cases for fluid-structure interaction.
	Displacement	Move vertices by a specified displacement. Adds the Linear Displacement node to the **Physics Values** folder of the region.
	Grid Velocity	Calculate vertex movement within a time step using a grid velocity. Adds the Grid Velocity node to the **Physics Values** folder of the region.

	New Coordinate	Specify new coordinates at each time step. Adds the New Coordinate node to the **Physics Values** folder of the region.
	Coordinate Offset	Specify an offset by which vertices move at each time step. Adds the Coordinate Offset node to the **Physics Values** folder of the region.
Initial Displacement	Specifies whether to modify the mesh generation position by adding a displacement.	
	Activated	The vertices of the mesh are moved by the initial displacement. Adds the Initial Displacement node to the **Physics Values** folder of the region. For more information, see What Is the Initial Displacement?.
	Deactivated	The vertices of the mesh remain in the mesh generation position. This position is used as the initial position.

What Is the Linear Displacement?

Use this method to set the linear displacement of vertices in the parent region for which a user-defined vertex motion is active. This method is valid only with Cartesian coordinate systems. Do not use this method in cylindrical or spherical coordinates. For vertex motion in these coordinate systems, use the New Coordinate or Coordinate Offset methods.

The **Linear Displacement** node acts as a vector profile and has a single additional property for the type of displacement.

Properties

Total Displacement	Use this property to specify whether the displacement is taken as incremental (applying the same displacement at each time step), or whether it is taken as a total displacement relative to the initial mesh at time zero. Use this latter option when you supply displacements as an absolute value.	
	Activated	The specified displacement is treated as a total value relative to the initial position at time zero.
	Deactivated	The specified displacement is applied to the vertices based on

		their position at the start of each time step.
Coordinate System	Coordinate system in which the displacements are defined.	
Method	The method that is used to define the value for the Linear Displacement.	
Dimensions	The dimensionality of the property that is expressed in terms of the basic dimensions, such as length, time (read-only).	

What Is the Grid Velocity?

Use this node to set the grid velocity that is applied to vertices in the parent region when a user-defined vertex motion is active. This node acts as a vector profile.

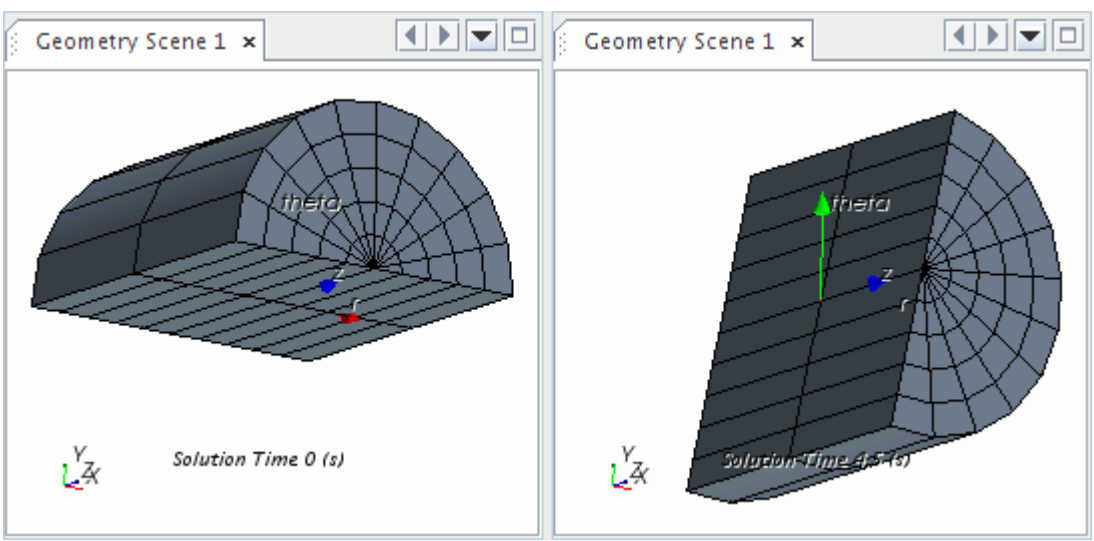
This method is valid only with Cartesian coordinate systems. Do not use this method in cylindrical or spherical coordinates. For vertex motion in these coordinate systems, use the New Coordinate or Coordinate Offset methods.

The coordinate system of the field function must be identical to the coordinate system that is used in the field function definition.

What Is the New Coordinate?

Use this node to specify the new coordinates for the vertices at each time step. The coordinates can be specified in the Cartesian, Cylindrical, or Spherical coordinates system.

Consider a half-section of a cylinder for rotation with a constant angular velocity about its axis:



You can use a field function to rotate this cylindrical section:

- Set the *Coordinate System* to the cylindrical coordinate system of the mesh for rotation, for example: **Laboratory-** > **Cylindrical 1**

- Create a field function with a *Definition* of the form:

```
[ $$Position(@CoordinateSystem("Cylindrical 1"))[0],
$$Position(@CoordinateSystem("Cylindrical 1"))[1]+0.1,
$$Position[2] ]
```

- Set the **New Coordinate***Method* to **Field Function** and select the field function that is defined in the previous step.

The coordinate system of the field function must be identical to the coordinate system that is used in the field function definition.

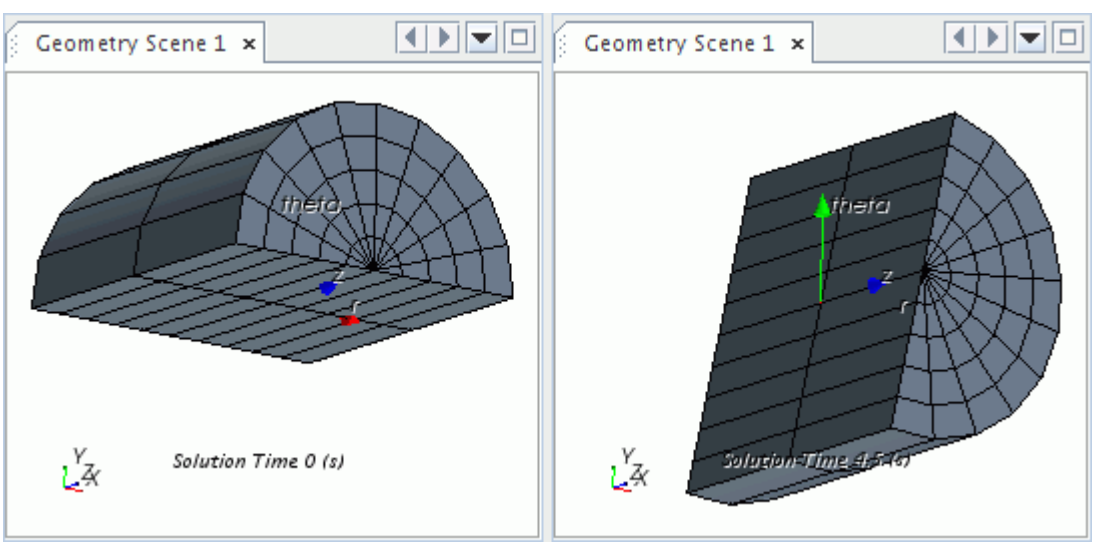
Properties

Coordinate System	Coordinate system in which the origin is defined.	
Method	The method that is used to define the value for the New Coordinate.	
	Field Function	Defines the value using a field function (typically user defined).
	User Code	Defines the value using a user subroutine.
Vector Function	Activates an object selector dialog from which you can choose a field function. This property is only available when **Method** is set to **Field Function**.	
Function	Provides a drop-down list from which you can choose a user function. This property is only available when **Method** is set to **User Code**.	

What Is the Coordinate Offset?

Use this node to specify the change in coordinates of the vertices at each time step. The change in coordinates can be specified in the Cartesian, Cylindrical, or Spherical coordinates system.

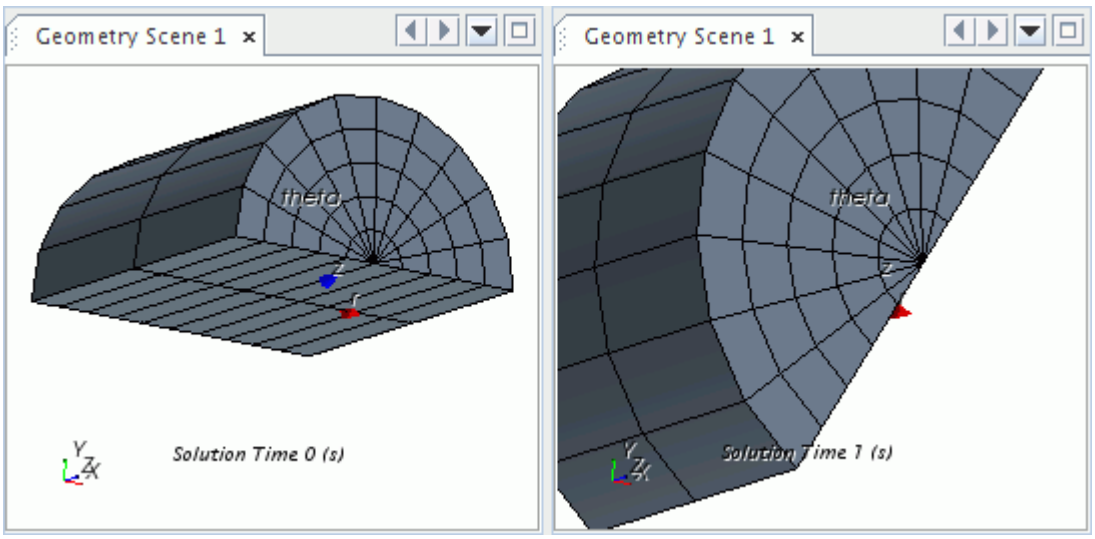
Consider the following section of a cylinder for rotation with a constant angular velocity about its axis:



- Set the *Coordinate System* to the cylindrical coordinate system of the mesh for rotation, for example: **Laboratory-** > **Cylindrical 1**
- Use the **Constant** method with a value of `[0.0, 0.1, 0.0]` for example.

If the mesh must grow in the radial direction as it is rotated, you cannot use the **Constant** method. Each radial vertex value increases at a different rate, depending on its radial position. Instead, use the **Field Function** method with a field function such as:

```
[ $$Position(  @CoordinateSystem("Cylindrical 1"))[0] * 0.05,  0.1,
0.0 ]
```



The coordinate system of the field function must be identical to the coordinate system that is used in the field function definition.

Properties

Coordinate System	Coordinate system in which the origin is defined.	
Method	The method that is used to define the value for the Coordinate Offset.	
	Constant	Specifies the value as a constant array.
	Field Function	Defines the value using a field function (typically user defined).
	User Code	Defines the value using a user subroutine.
Value	Specifies the x, y, and z values. These values are separated by commas in a text entry. You can also enter an expression, such as `10 * sin(3.14 * $Time)`, directly. The property customizer opens a	

	dialog which contains separate fields for these values. This property is only available when **Method** is set to **Constant**.
Vector Function	Activates an object selector dialog from which you can choose a field function. This property is only available when **Method** is set to **Field Function**.
Function	Provides a drop-down list from which you can choose a user function. This property is only available when **Method** is set to **User Code**.

What Is the Initial Displacement?

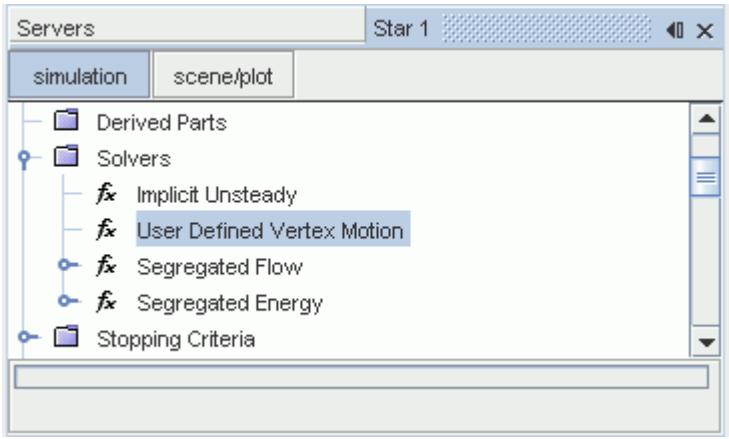
Use this node to adjust the mesh from the mesh generation position to the initial mesh position for the simulation. You specify an initial displacement by which the vertices are moved from the mesh generation position to the initial mesh position of the simulation. The initial displacement is applied only once at the start of the simulation to obtain the initial position for the simulation. The initial position is used to initialize the flow field. Note that selecting Solution > **Clear Solution...** > **Reset Mesh**resets the mesh back to the mesh generation position.

Properties

Coordinate System	Coordinate system in which the initial displacement vector is defined.
Method	The method that is used to define the value for the Initial Displacement.
Dimensions	The dimensionality of the property that is expressed in terms of the basic dimensions, such as length (read-only).
Value	Specifies the x, y, and z values. These values are separated by commas in a text entry. You can also enter an expression, such as `10 * sin(3.14 * $Time)`, directly. The property customizer opens a dialog which contains separate fields for these values. This property is only available when **Method** is set to **Constant**.
Vector Function	Activates an object selector dialog from which you can choose a field function. This property is only available when **Method** is set to **Field Function**.
Function	Provides a drop-down list from which you can choose a user function. This property is only available when **Method** is set to **User Code**.

What Is the User-Defined Vertex Motion Solver?

The user-defined vertex motion solver, which has expert properties, applies your specified displacement or velocity to the mesh.



Expert Properties

Update At Inner Iterations	Repeats the movement at each inner iteration during the time-step. This option is only meaningful for cases where some boundary condition depends on the solution; it is off by default to reduce computational costs.	
	Activated	Repeats the motion at each inner iteration.
	Deactivated	Leaves off this option.
Zero grid flux	When this option is activated, the mesh still moves but the grid flux is not included in the transport equations. This option provides more stability when marching towards a steady solution with an unsteady solver. It is also useful when you are simply adjusting the shape of the region without solving the transport equations.	
	Activated	Ignores the grid flux.
	Deactivated	Includes the grid flux.
Solver Frozen	Prevents or allows the solver from updating the solution while iterating. Do not activate this feature, as it is primarily a “debugging” tool and not supported for normal simulation work.	
	Activated	The solver does not update the solution.
	Deactivated	The solver updates the solution.

An Example of User Coding

User coded functions can be used to define the displacements or grid velocity that STAR-CCM+ requires.

Such a function requires access to all the vertices in the mesh, obtained as follows:-

```
ucarg((void *)userMotionFunc, "Vertex", "Coord", sizeof(CoordReal[3]));
```

The argument type here is “`Vertex`”, rather than “`Cell`” or “`Face`”, and the array being requested is “`Coord`”. The argument belongs to the function, `userMotionFunc`, which could have a declaration as follows:

```
void userMotionFunc(Real (*result)[3], int size, CoordReal (*vpos)[3])
{
   // user coding here to calculate the displacement
   // or grid velocity, which will be returned
   // in the result array
 }
```

In this function, the result array would return one of the following quantities:

- incremental displacement;
- total displacement;
- grid velocity.

The particular quantity depends on the choice you have made on the **User Defined Vertex Option** node and its associated value node.

When a suitable function has been compiled and linked, it appears in the Field Functions list as a vector function. Select the **Field Function** option for the *Method* property of the **UDV: LinearDisplacement** node or **Grid Velocity** node, and choose the user coded field function from the list.

General Motion Formulation

This section discusses how the motion modeling affects the continuum transport equations.

The motion enters these equations in two ways:

- As a grid flux in the convective terms
- As a body force due to system rotation in the momentum equations

Contents:

Grid Flux
Body Force Due to System Rotation
Common Field Functions

Grid Flux

The grid flux at a face is calculated as follows.

For a moving reference frame:

$$G_f = \mathbf{v}_g \cdot \mathbf{a}_f \tag{144}$$

where $\mathbf{v}_g$ is the grid velocity and $\mathbf{a}$ is the face area.

For rigid body motion, morphing, and user-defined vertex motion, the first-order grid flux is:

$$G_f = \frac{\delta V_f^n}{\Delta t} \tag{145}$$

The second-order grid flux is:

$$G_f = \frac{\left(\alpha^2 - 1\right)\delta V_f^{n+1} - \delta V_f^n}{\alpha(\alpha - 1)\Delta t^{n+1}} \tag{146}$$

where

$$\begin{aligned} \alpha &= 1 + \frac{\Delta t^{n+1}}{\Delta t^n} \\ \Delta t^{n+1} &= t^{n+1} - t^n \\ \Delta t^n &= t^n - t^{n-1} \end{aligned}$$

and

δV_f^{n+1}	Volume swept by cell face f over time step Δt^{n+1} .
δV_f^n	Volume swept by cell face f over time step Δt^n .

In Eqn. (146), the swept volumes are calculated from the grid position at two time levels. n denotes the current time level, $n-1$ the previous time level, and $n+1$ the next time level. Δt is the time step.

The grid flux G_f modifies the convective flux at a face:

$$\left[\phi\rho\left(\mathbf{v} \cdot \mathbf{a} - G\right)\right]_f = \left(\dot{m}\phi\right)_f = \dot{m}_f \phi_f \tag{147}$$

where ϕ_f and $\dot{m}_f$ are the scalar values and mass flow rates at the face, respectively.

Body Force Due to System Rotation

If the reference frame is rotating, the rotational body force ($\mathbf{f}_r$ in Eqn. (1650)) is:

$$\mathbf{f}_r = \rho\boldsymbol{\omega} \times \mathbf{v} \tag{148}$$

where $\boldsymbol{\omega}$ is the rotational velocity.

Common Field Functions

The following primitive field function is made available to the simulation with motions.

This field function has a typical set of base properties.

Grid Flux	The rate at which a face sweeps out volume due to grid motion.

Modeling Materials

Material models simulate substances, including various mixtures.

Some simulations that you perform in STAR-CCM+ involve modeling a substance such as air, water, or aluminum. Others involve several substances such as a combusting mixture of gases, or a sloshing mixture of air and water. The Material model handles the definition of these substances.

Material Models

A material model is responsible for managing the material, that is, substance or substances, being simulated in the continuum.

The simulated material is responsible for managing the various thermodynamic and transport properties relevant to that material and to the physical processes being modeled in the continuum.

There are three general types of material models available in STAR-CCM+: single-component, multi-component, and multiphase mixture. Each of these material models manages one specific type of material: a pure substance, a multi-component mixture, or a multiphase mixture.

Selecting a material model early in the model selection process allows STAR-CCM+ to assist you in making subsequent modeling choices appropriately.

A material model is only required if you plan to model fluid flow or heat transfer. If, for example, all you are doing is checking the mesh, a material model is not needed.

Single-Component Material Models

The single-component material models are for modeling pure substances such as air, water, or aluminum.

There are three single-component material models in STAR-CCM+: Gas, Liquid, and Solid models. Each manages one single-component material representing the pure substance being modeled. The Gas model manages a gas material (that is, a pure substance in the gaseous phase). The Liquid model manages a liquid material (that is, a pure substance in the liquid phase). The Solid model manages a solid material (that is, a pure substance in the solid phase).

Multi-Component Material Model

The multi-component material models are single-phase models for simulating fully miscible mixtures of two or more pure substances in the same phase, such as a combusting mixture of fuel, oxidizer and product.

There are four multi-component material models in STAR-CCM+, for gases, liquids, and solids:

- The Multi-Component Gas and Multi-Component Liquid models. Each manages a multi-component material representing the single-phase, miscible mixture being modeled. The Multi-Component Gas model manages a Gas Mixture material (that is, a miscible mixture of two or more pure substances in the gaseous phase). The Multi-Component Liquid model manages a Liquid Mixture material (that is, a

miscible mixture of two or more pure substances in the liquid phase). These models are specified as either reacting or non-reacting.

- The Multi-Part Solid model and the more specialized Multi-Layer Solid model. Each manages multiple solid materials in a single physics continuum, mapped to individual parts.

Multiphase Mixture Material Model

The multiphase mixture material model is for simulating two or more immiscible phases, where each phase is composed of a pure gas or liquid substance, such as air and water.

The Multiphase Mixture model in STAR-CCM+ manages objects that control properties for the mixture as a whole. The phase materials and equation of state specifications are managed with the Eulerian Multiphase model.

Contents:

Managing Single-Component Materials
Managing Multi-Component Materials
Modeling Multi-Component Solids
Modeling a Multiphase Mixture
Modeling Linear Isotropic Elastic
Setting Material Properties Methods
Using Material Databases

Modeling a Single-Component Material

To model a single-component material, that is, a pure substance, activate either the Gas, Liquid, or Solid model by selecting the appropriate radio button in the Material section of the model selection dialog.

Eulerian phases include a Particle model when the Multiphase Segregated Flow model is selected.

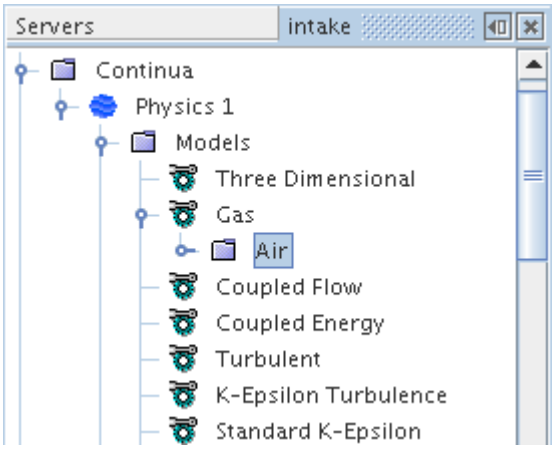
Model Names	**Gas**	
	Liquid	
	Solid	
Provided By	**[physics continuum]** > **Models** > *Material*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Gas**	
Requires	Immediately available.	
Activates	Physics Models	*Flow* for **Gas** or **Liquid**
	Model Controls (child nodes)	A node for the specific gas, liquid, or solid. The defaults are: • **Gas**: **Air** • **Liquid**: **H2O** (water) • **Solid**: **Al** (aluminum)

Each model manages the material as outlined in Managing Single-Component Materials.

If you wish to model a material other than the default, you can replace the existing material with a different one from the material database. This procedure is outlined in the section describing how to manage single-component materials. Alternatively, you can edit the properties of the gas directly as outlined in Setting Mixture Properties.

Managing Single-Component Materials

The single-component material node, which is a child node of the material model, represents the specific type of substance being modeled. For example, as shown in the following screenshot, the **Air** node exists within the **Gas** model node, since air is the default gas type when the Gas material model is selected.



The type of material defaults to air, water, or aluminum, depending on the type of the material model (Gas, Liquid, or Solid). A node with a corresponding name appears with properties and a pop-up menu.

While you can rename this node using standard techniques, you can as easily replace the type of material used. It is also possible to set material properties.

Single-Component Material Properties

Database Material

The name of the material as it is listed in the material database (read-only).

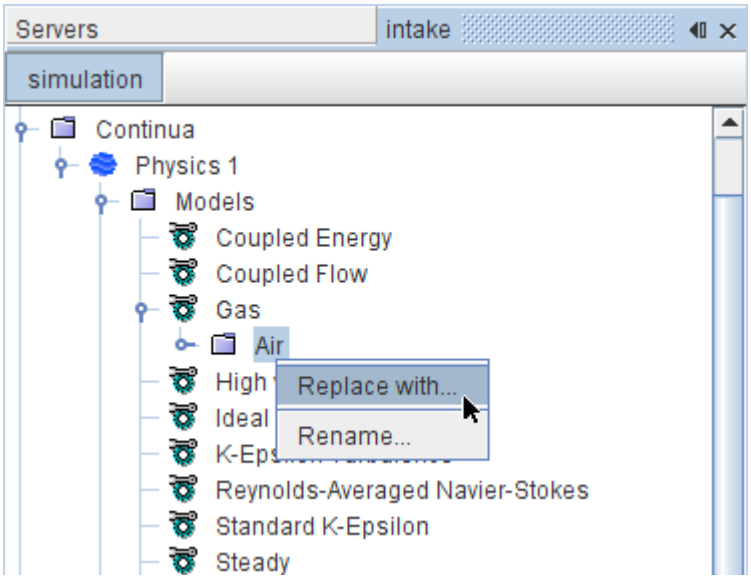
Single-Component Material Pop-Up Menu

Replace with

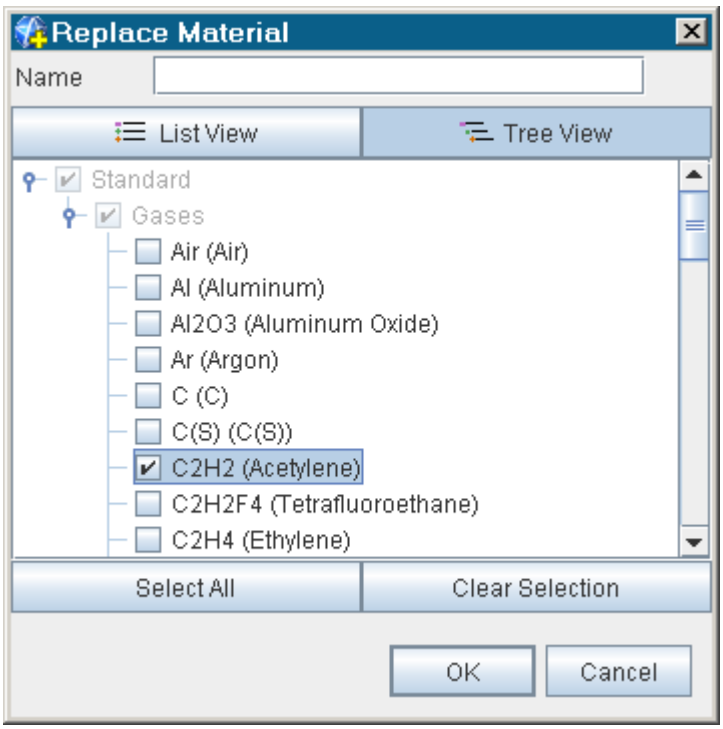
Opens a submenu of all materials of the same phase in the material database.

Replacing Single-Component Materials

You can control the choice of material in your simulation by using the Replace feature. To replace a single-component material, right-click the node of the existing material type and select **Replace with...**



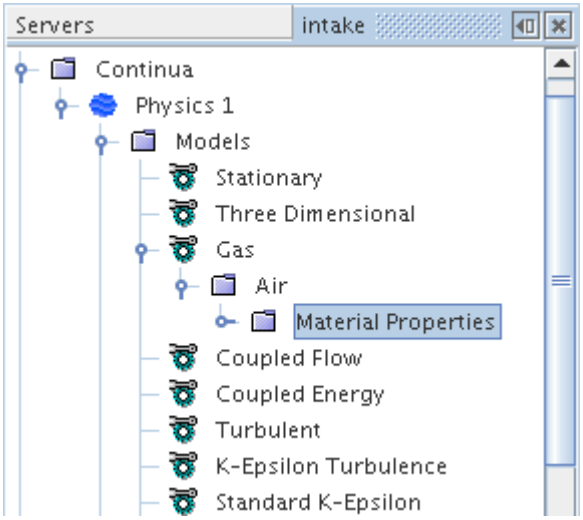
Then select a replacement material from the list in the *Replace Material* dialog and click **OK**.



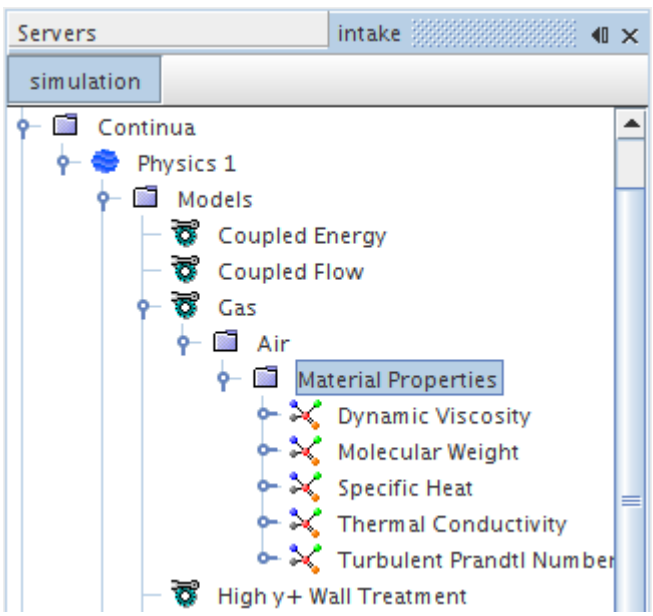
The node of that selected material appears in the place of the existing material node.

Setting Material Properties

The **Material Properties** manager node is a placeholder for the properties of a substance. This node is empty until either an Equation of State model or a Viscous Regime is selected, after which it becomes populated with material property nodes.



A fully populated node is shown below.



Modeling a Multi-Component Mixture

A multi-component mixture is a miscible mixture of two or more pure substances in the same phase.

Model Names and Abbreviations	**Multi-Component Gas**	
	Multi-Component Liquid	
	Multi-Component Solid (See also Modeling Multi-Part Solids and Modeling Multi-Layer Solids.)	
Provided By	**[physics continuum]** > **Models** > *Material*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Multi-Component Gas**	
Requires	Immediately available.	
Activates	Physics Models	*Reaction Regime* for **Multi-Component Gas** or **Liquid** (See Combustion and Other Reacting Flows.) *Multi-Component Solid* for **Multi-Component Solid** (See Managing Multi-Component Solids.)
	Model Controls (child nodes)	**Mixture Components** for **Multi-Component Gas** or **Liquid** (See Fluid Mixtures below.)

Fluid Mixtures

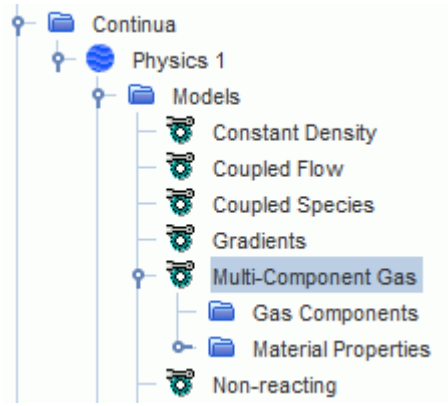
A gas or liquid mixture is a material representing a miscible mixture of two or more pure fluids. The Multi-Component Gas or Fluid model manages the material as outlined in Managing Multi-Component Materials.

The **Gas Components** node and **Liquid Components** node are manager nodes for the fluids composing the mixture. Initially, the **Components** node is empty. It is here that you add, remove, and replace fluids in the mixture as outlined in the section on managing multi-component materials. The material properties of each individual component in the mixture are then accessible and can be set as outlined in Setting Material Properties.

The **Mixture Properties** node is a manager node for the material properties of the entire fluid mixture as a whole, distinct from the properties of each individual component. This node is empty initially, but is populated with material property nodes once other physical models are selected. The material properties of the mixture can be set as outlined in Setting Mixture Properties.

Managing Multi-Component Materials

The multi-component mixture node, which is a child node of the material model, is a placeholder for two manager nodes: one for the components of the mixture where you would add substances, the other for the properties of the mixture. For example, in the following screenshot, the **Gas Mixture** node contains a **Gas Components** node and a **Mixture Properties** node.

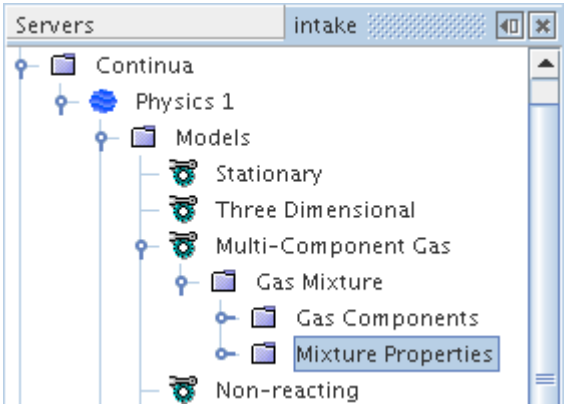
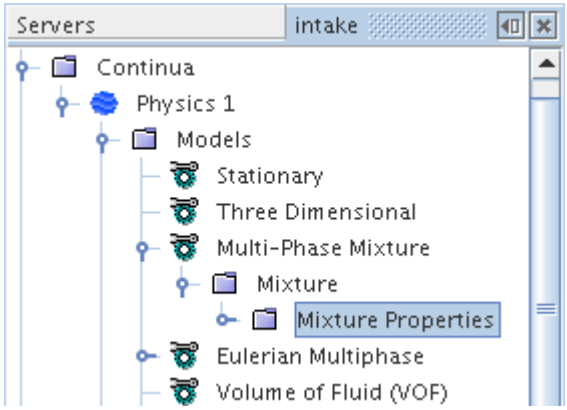


It is possible to rename the multi-component mixture node in a manner that reflects the specific aspects of the material you are modeling. A mixture can also be copied to another simulation by copying the continuum that contains it.

Setting Mixture Properties

The **Mixture Properties** node is a manager node for the material properties of the entire mixture as a whole, distinct from the properties of each individual component.

The information in this section applies equally well to the corresponding nodes inside a multi-component or multiphase material model node, since these nodes function in a similar fashion.

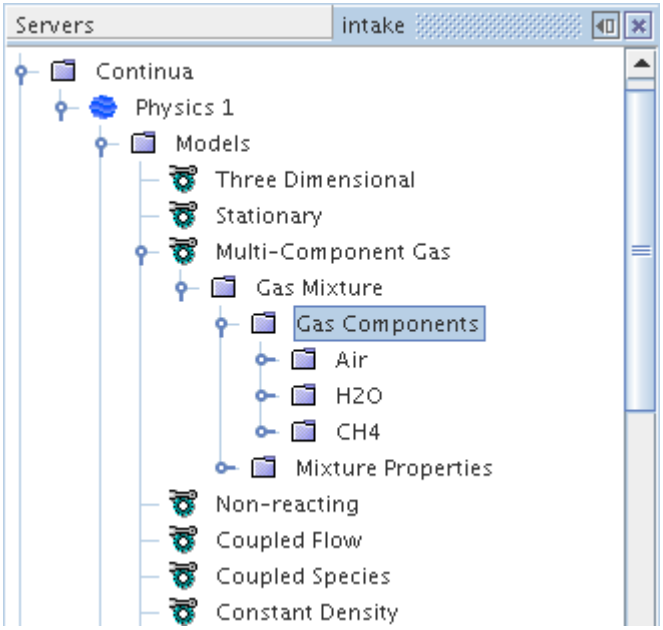
	
Multi-Component	Multiphase

This node functions the same as the material properties manager of a single-component material. It is empty initially, but is populated with material property nodes once other physical models are selected. The material properties of the fluid mixture can be set as outlined in the section describing setting material properties.

Managing Mixture Components

The mixture components manager node is a placeholder for the nodes of the substances that comprise a mixture.

This node (**Gas Components** in the following screenshot) remains unpopulated until you add components.



You can also modify a mixture through the following actions:

- Replacing components
- Reordering components
- Setting component properties

A material can be renamed using the standard techniques.

Materials can be deleted from a mixture by right-clicking on the node and selecting **Delete**. This action removes the material from the mixture, but not from the material database.

The number of components is given in the *Number of Components* property (Read Only) of the **Components** node under **Gas Mixture** or **Liquid Mixture** or under **Multi-Part Solid** > **Solids** or **Multi-Layer Solid** > **Solids**.

Mixture Components Manager Pop-Up Menu

Gas Mixture

New Gas Components

Opens the *Select Mixture Components* dialog, which displays all gas materials in the material database. Activate the checkboxes in the order in which you want to add components, and click **OK**.

Liquid Mixture

New Liquid Component

Opens a submenu of all liquid materials in the material database from which you can add a component to the mixture.

Adding Mixture Components

The same method is used to add components to a mixture irrespective of whether a multi-component gas or liquid is being modeled. The component names and material properties are retrieved from the material database, and constant property values are assigned by default.

As part of setting up the simulation, review the default material property settings of the mixture as well as the settings of each new component. If necessary, change the settings of the mixture properties, and then the settings of each new component as required (see Managing Multi-component Materials).

The ordering of components can be important for reasons of numerical accuracy. See the section on using the Species model for details. The following steps explain how to add a component to a mixture:

1. Right-click the **Gas or Liquid Components** node and choose **Select Mixture Components**.

   The *Select Mixture Components* dialog appears. This dialog is essentially the object selection dialog that is applied to material selection.
2. In the *Tree View* panel, expand the **Material Databases** > **Standard** node (corresponding to the Standard materials database) and its child node (**Gases** or **Liquids**).
3. Activate the boxes corresponding to the desired components, and click **OK**.
4. The new components appear beneath the **Gas** or **Liquid Components** node.

Frequently Used Mixtures

If you often use specific multi-component mixtures, you can use material database folders to define these mixtures and then save them in the material database for later re-use. Use the following steps:

1. Using the pop-up menu of the **Standard** node, create a folder for the mixture and give it an appropriate name, such as `Air`.
2. Copy and paste the components of the mixture from pre-existing databases. For example, copy **N2 (Nitrogen)**, **O2 (Oxygen)**, and **Ar (Argon)** from the supplied **Standard** > **Gases** folder.
3. Right-click the parent database node and select **Save**.
4. To use the Air mixture, select the **Air** folder in the *Select Mixture Components* dialog box. All three components are selected together.

Replacing Mixture Components

You can control the choice of mixture component in your simulation by using the Replace feature.

To replace a mixture component, right-click the node of the component and select **Replace With...**. The *Replace Material* dialog appears. Expand the nodes as necessary and activate the box corresponding to the replacement component. Click **OK** to confirm the selection.

Reordering Mixture Components

You can control the order in which the mixture components appear in the object tree.

Right-click the **Gas** or **Liquid Components** node and choose **Reorder Mixture Components...** from the menu.

The *Reorder Mixture Components* dialog appears, showing the names of selected components in their order on the tree.

To move the position of a component:

- Click and hold the left mouse button on its name;
- Drag the mouse pointer until it hovers over the destination entry in the list; and,
- Release the mouse button.

The dragged component now occupies the targeted entry, and the component that was there moves up one position. Click **OK** to save the changes.

Setting Component Properties

Each component of a mixture has properties that you can set. The mixture-component material node exists as a child node of the mixture components node. It contains a sub-node that functions the same as the material properties of a single-component material.

Mixture Component Properties

Database material

The name of the material (read-only).

ID

Unique index for this component within the simulation. If this component is copied to another continuum, the copy within the target continuum is assigned a new, unique, ID.

Mixture Component Pop-Up Menu

Replace with...

Opens the *Replace Material* dialog from which you can replace the selected component.

Modeling Multi-Component Solids

Some simulations require you to model many solid parts of different materials.

Examples of such simulations are:

- Electronics cooling studies where a motherboard contains tens or hundreds of different components
- Vehicle underhood thermal analysis

Use the Multi-Component Solid model and then either the Multi-Part Solid or the Multi-Layer Solid model. These models allow you to define multiple solid materials in a single physics continuum, and to map those materials to individual parts in the region to which the continuum applies. The **Multi-Part Solid** and **Multi-Layer Solid** nodes represent these models in a physics continuum.

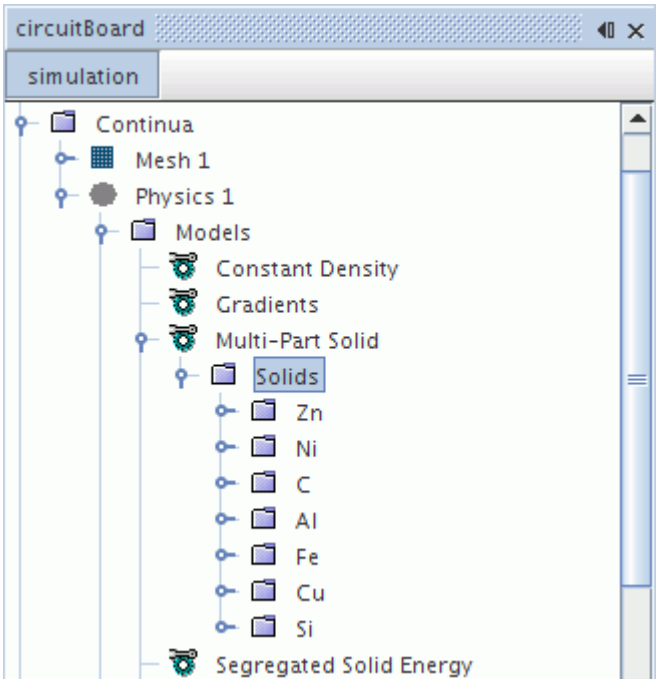
Multi-Component Solid Model

Provided By	**[physics continuum]** > **Models** > *Material*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Multi-Component Solid**	
Requires	Immediately available.	
Activates	Physics Models	*Multi-Component Solid* (See Multi-Part Solid and Multi-Layer Solid models.)

Defining Solid Components

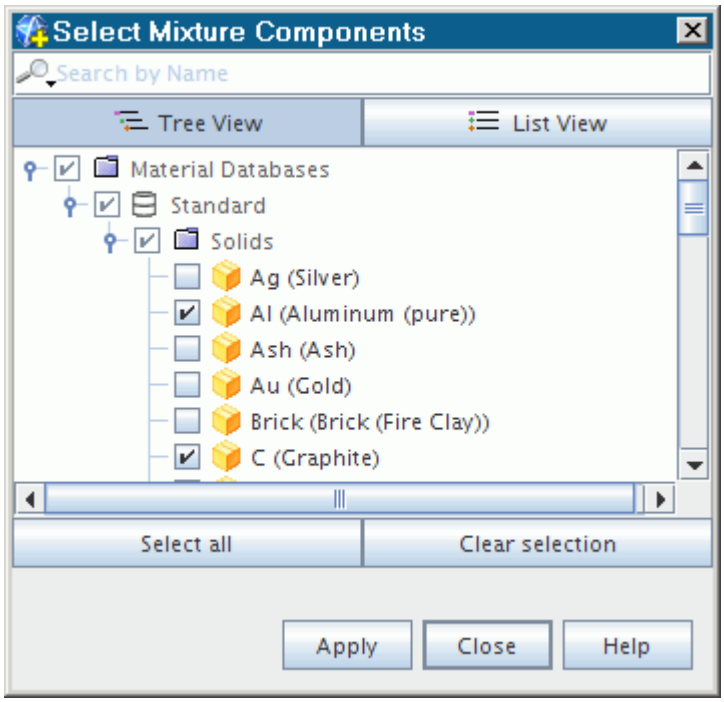
The Multi-Part Solid and Multi-Layer Solid models require you to add the component materials in the same way as you add mixture components for a multi-component liquid or gas.

The **Solids** node manages the component materials. Default properties for each of these components come from the material database.



To add solid components:

1. Right-click the **Solids** node and choose **Select Mixture Components...** The *Select Mixture Components* dialog appears.
2. Expand the nodes within the *Select Mixture Components* dialog.
3. Activate the checkboxes next to the materials you require; click **Apply**, then **Close,** when you are done.



You can manage solid components in the same way as you manage gas or liquid components.

Once these components are in place, work with material properties methods as needed. The choices of available material properties and associated methods are the same as with a Solid model, except that the Anisotropic method for Thermal Conductivity is not available with the Multi-Part Solid model.

Contents:

Managing Multi-Part Solids
Modeling Multi-Layer Solids
Assigning Parts to Materials
Post-Processing Multi-Part Solid Data

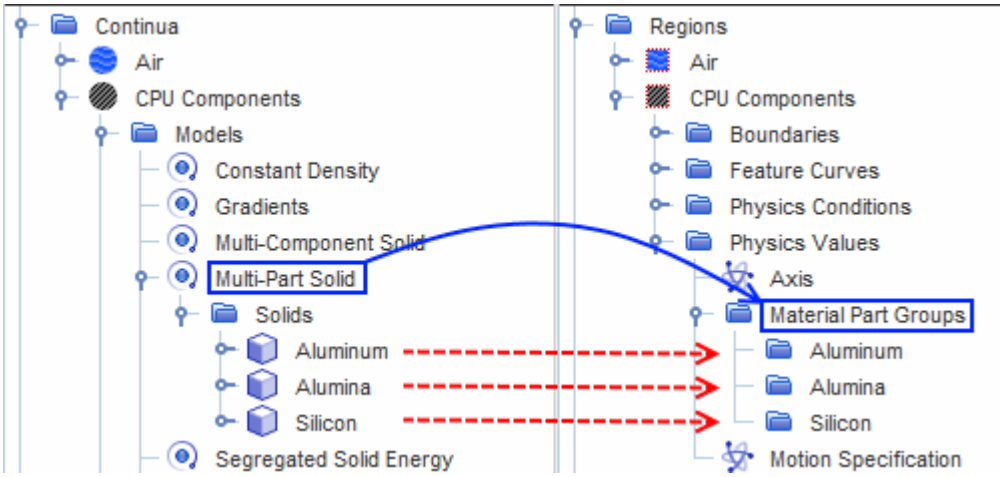
Modeling Multi-Part Solids

The Multi-Part Solid model can simulate several parts of different compositions in a single region.

Provided By	**[physics continuum]** > **Models** > *Multi-Component Solid*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Multi-Part Solid**	
Requires	*Space*: any *Time*: **Steady** or **Implicit Unsteady** *Material*: **Multi-Component Solid**	
Activates	Model Controls (child nodes)	Solids. (See Defining Solid Components.)
	Region Inputs	**Physics Values** > **Material Part Groups** (See Assigning Parts to Materials.)
	Field Functions	**Cell Part Index**, **Solid Material ID**. (See Multi-Part Solid Field Functions Reference.)

When a physics continuum containing the Multi-Part Solid model is applied to a region, a **Material Part Groups** node is added to the **Physics Values** manager for that region. You use the **Material Parts Group** node to map between solid materials and geometry parts.

The following diagram shows the **Multi-Part Solid** node in the physics continuum, and the **Material Part Groups** node in a region to which the physics continuum is applied:



Requirements

- Multi-part solids can only be used in simulations where geometry parts define the geometry. The identity of these geometry parts must be retained at the region level. Therefore, when you assign geometry parts to regions, include multiple geometry parts in the same region.

  Select the **One region for all parts** option in the *Region Mode* menu of the *Assign Parts to Regions* dialog.
- When you have interfaces between parts, the best accuracy can be obtained by setting the interface type to **Contact**. In this instance, the thermal flux across the interface is calculated in a manner that accounts for the presence of discontinuous properties. Furthermore, Contact interfaces support the application of a Thermal Contact Resistance and/or Heat Source within the interface if necessary. In contrast, if you set the interface between parts to type **Internal**, the heat flux at the interface is calculated based on averaged properties. This calculation results in a certain amount of smoothing of the temperature field across the interface, which is not always realistic. Therefore, set interfaces between parts of dissimilar solid materials to **Contact**.
- If any material in the simulation has anisotropic thermal conductivity, create a dedicated physics continuum for that material. The Multi-Part Solid model does not currently support materials that require the **Anisotropic** method for *Thermal Conductivity*.

Workflow

To set up a simulation in STAR-CCM+ with multi-part solids:

1. Select the Multi-Part Solid model.
2. Select Coupled Solid Energy or Segregated Solid Energy from the *Optional Models* group.
3. Select a model from the *Equation of State* group.
4. Assign each region to the proper continuum. Only a solid region is compatible with a continuum that contains the Multi-Part Solid model.
5. Assign parts to materials.
6. Change the interface type between solid parts that are in contact.

This model activates field functions that are useful for post-processing.

Multi-Part Solid Field Functions Reference

The following primitive field functions are made available to the simulation when the Multi-Part Solid model is used.

Cell Part Index	The Index value of the part that is associated with a given cell. Generated by the meshing process. Also used in parts-based reporting.
Solid Material ID	The ID value of the solid material that is associated with a given cell. Identifies which solid component in the continuum defines the material properties for that cell. Also allows visual association of cells defining parts with a particular material.

Modeling Multi-Layer Solids

Frequently, solids take the form of multi-layer systems. The Multi-Layer Solids material model is specific for simulations within multi-layer solids. Uses for the Multi-Layer Solids model include printed circuit boards (PCBs), thin heat-conducting shells in vehicles, and solid stress applications for thin composite structures. The model applies to both solid volume regions and solid shell regions.

The Multi-Layer Solid model specifies the number and properties of the laminates in the simulation, and of the layers within the laminates.

Provided By	**[physics continuum]** > **Models** > *Multi-Component Solid*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Multi-Layer Solid**	
Requires	*Space*: any *Time*: **Steady** or **Implicit Unsteady** *Material*: **Multi-Component Solid**	
Activates	Model Controls (child nodes)	**Solids**. (See Defining Solid Components.) **Material Properties**: The information under this node is read-only. It consists of the types of properties available and the methods set for each property (**Multi-Layer**). The properties are calculated internally by the Multi-Layer Solids model. **Laminates** (See below.)
	Region Inputs	**Physics Values** (See below.)

Workflow

A typical simulation using the Multi-Layer Solids model includes several similar but distinct laminates, for instance several layers on a Printed Circuit Board (PCB) composing a volume region. General steps to using this model are:

1. In the *Physics Model Selection* dialog for a physics continuum, select the models that you require for the simulation, including the **Multi-Layer Solids** model.
2. Choose all the solid materials that can appear in any layer within the board that you are modeling:
   a. Select **[physics continuum]** > **Models** > **Multi-Layer Solid** > **Solids**, right-click, and select **Select Mixture Components...**
   b. In the *Select Mixture Components* dialog, select components from the *Material Databases* to match the materials you require. If specific materials are not available, select arbitrary materials, rename them, and redefine their properties manually.
3. Define a laminate object as a series of layers:
   a. Select **Multi-Layer Solid** > **Laminates**, right-click, and select **New** to create a Laminate node. Rename the node if necessary.
   b. Select the **Laminate** node, right-click, and select **New** to create a layer. Rename the **Layer n** node if necessary.
   c. Create any additional layers for the laminate.
   d. Select each **Layer n** node in turn. Specify the thickness of the layer as a normalized proportion of the overall thickness of the laminate. Within a laminate, the thickness of all layers must sum to 1.0. Also set the volume fraction of each solid component present in the layer.
4. Repeat the previous step until you have defined all distinct laminates for the simulation.
5. Depending on the geometry you are modeling, assign laminates to regions, or parts within regions. If you have many connected geometry parts in your simulation, assign them to a single region and use the part subgrouping method to assign laminates to specific parts within the region.

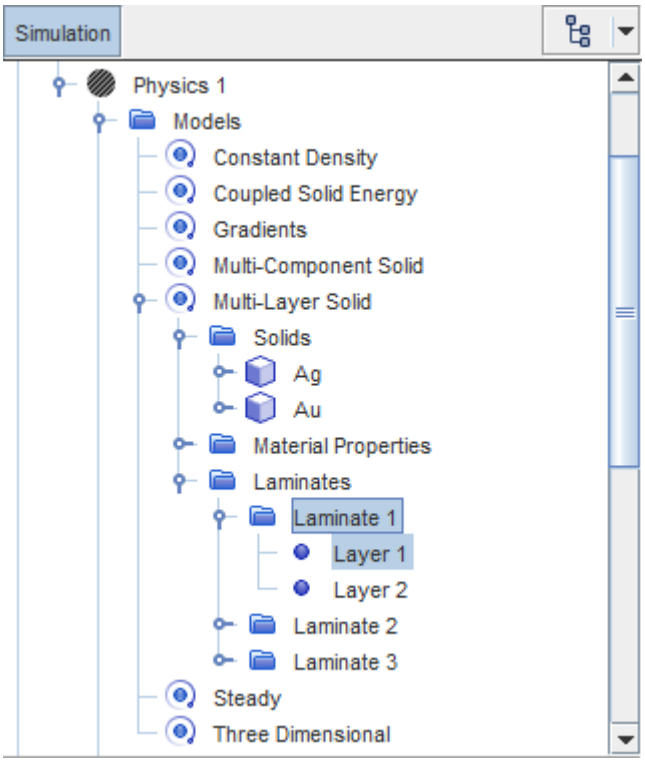
   To assign laminates using part subgrouping:
   1. Make sure that the geometry parts are assigned to the target region.
   2. Select the target region and activate *Allow Per-Part Values*.
   3. Expand the **Regions** > **[region]** > **Part Subgroupings** node. Within the default subgrouping, create as many subgroups as you require. You can associate one laminate object with each subgroup.
   4. Select the subgroups in turn and set their *Objects* to one or more geometry parts from the set of parts that are assigned to the region.
   5. Select the **[region]** > **Physics Values** > **Laminate Specification** node and activate *Specify by Part Subgroup*.
   6. Expand the **Laminate Specification** node, select each subgroup in turn, and set their *Laminate* property to the appropriate laminate object.

   To assign laminates per region:
   1. Make sure that the geometry parts are assigned to the target region.
   2. Select the **[region]** > **Physics Values** > **Laminate Specification** node and set *Laminate* to the appropriate laminate object.

   If you are using the Shell Three Dimensional space model, you can assign the laminate to a shell region.
6. Select **Physics Values** > **Laminate Normal** and specify the normal vector that applies to all laminates in the region unless you have chosen the Shell Three Dimensional model for space. In that case, the only normal vector specification is **Shell Normal**.

Laminates

The **Laminates** manager node has a **Laminate** sub-node for each laminate in the simulation. In turn, each **Laminate** sub-node has a **Layer** sub-node for each layer in the laminate. Right-click on the **Laminates** manager to create new **Laminate** nodes; right-click on a **Laminate** node to create new **Layer** nodes. Rename **Laminate** and **Layer** nodes to suit the simulation.



Layer Node Properties

Thickness

Determines the relative thickness of the layer. Enter the fraction of the laminate thickness that is occupied by the layer. For example, if the layer is 1 mm thick and the whole laminate is 4 mm thick, enter `0.25`.

Volume Fraction

Determines the composition of the layer, by volume fraction, based on the materials listed under the **Solids** node. The composition is expressed as an ordered list, following the order of materials under the **Solids** node.

Example: The solids listed are, first, `Ag` (silver) and, second, `Au` (gold). The entry `[0.75, 0.25]` indicates that the layer is three quarters silver and one quarter gold, by volume.

Physics Values

When the Multi-Layer Solid model is selected, the following values become available under the **Region** nodes that are associated with **Laminate** nodes:

Laminate Normal

Describes the properties of the normal vector to the laminate.

Method

Specifies the method for defining the normal vector from a list of profile options.

When the Shell Three Dimensional model is selected for *Space*, *Method* is set to **Shell Normal**; the shell and the laminate use the same normal vector.

When any other space model is used, common methods for a vector profile can be used. When *Method* is set to one of the **Table** options, it makes a sub-node available for the table under the **Laminate Normal** node. The default method is **Constant**.

Value

The `[x,y,z]` coordinates for the normal vector. This property is only available when *Method* is set to **Constant**.

Vector Function

Opens a dialog offering a choice of field functions for the normal vector. This property is only available when *Method* is set to **Field Function**.

Function

Provides a drop-down menu from which to select a user function. This property is only available when *Method* is set to **User Code**.

Dimensions

Read only. Gives the dimensions of the normal vector.

Coordinate System

Specifies the coordinate system for the normal vector. The default is **Laboratory**.

Laminate Specification

Specifies the **Laminate** nodes that are associated with the **Region** node.

Specify by Part Subgroup

When **On**, makes available the **By Part Subgroup** node under **Laminate Specification** and the **Part Subgroupings** node under the **Region** node. See Applying Quantities by Subgroup.

Laminate

Available when *Specify by Part Subgroup* is **Off**. Select the laminate from the drop-down menu. The default is **None**.

Multi-Layer Solid Formulation

Properties for the laminates in multi-layer solids are calculated from weighted averages of the properties of the constituent layers of the laminates.

In the following equations, t_i is the thickness of the ith layer.

Effective Anisotropic Thermal Conductivity

The normal component of the thermal conductivity is:

$$K_{nr}^{ef} = \frac{1}{\sum_{i=1}^{n} \frac{t_i}{K_{i,nr}}} \tag{149}$$

where:

- $K_{i,nr} = \sum_{j=1}^{m} \alpha_j \cdot K_{j,nr}$ is the normal component of the ith layer of the laminate.
- α_j is the volume fraction of the jth component in the ith layer.
- $K_{j,nr}$ is normal thermal conductivity of the jth component.

The planar component of the thermal conductivity is:

$$K_{pl}^{ef} = \sum_{i=1}^{n} t_i \cdot K_{i,pl} \tag{150}$$

where:

- $K_{i,pl} = \sum_{j=1}^{m} \alpha_j \cdot K_{j,pl}$ is the planar component of the ith layer of the laminate.
- $K_{j,nr}$ is planar thermal conductivity of the jth component.

Effective thermal conductivity tensor:

$$\mathbf{K}^{ef} = \begin{bmatrix} K_{nr}^{ef} & 0 & 0 \\ 0 & K_{pl}^{ef} & 0 \\ 0 & 0 & K_{pl}^{ef} \end{bmatrix} \tag{151}$$

Effective Density

The effective specific heat is:

$$Cp_{ef} = \frac{\sum_{i=1}^{n} t_i \rho_i Cp_i}{\sum_{i=1}^{n} t_i \rho_i} \tag{152}$$

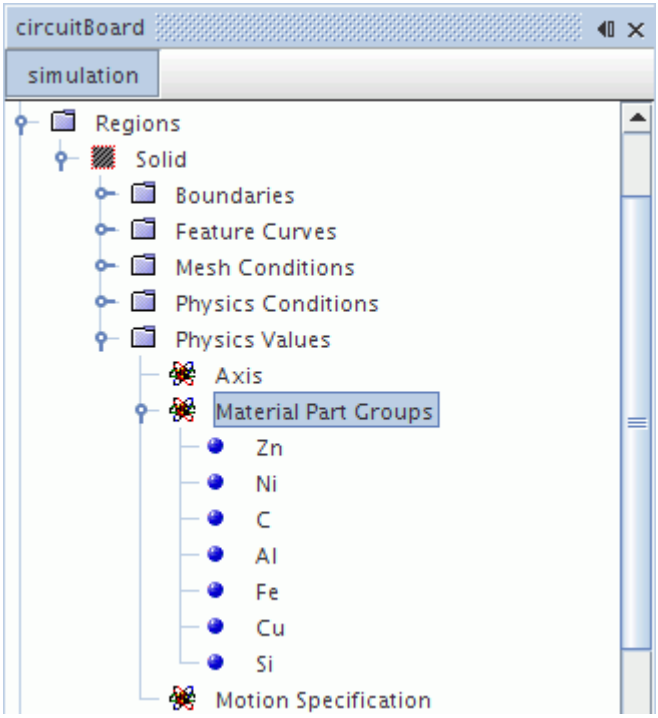
where:

- $Cp_i = \frac{\sum_{i=1}^{m} \alpha_i \rho_i Cp_i}{\sum_{i=1}^{m} \alpha_i \rho_i}$ is the specific heat of the ith layer of the laminate.
- Cp_j is the specific heat of the jth component.

Assigning Parts to Materials

When you associate a multi-part solids physics continuum with a region, STAR-CCM+ adds a **Material Part Groups** node to the **Physics Values** for the region.

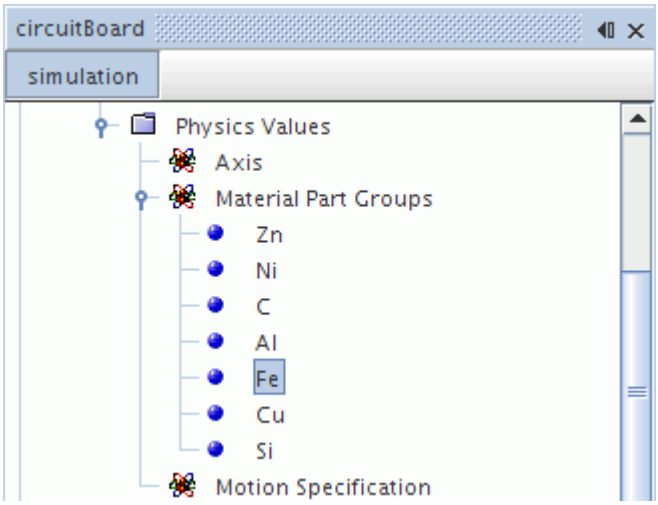
Material part groups corresponding to the solid components of the Multi-Part Solid model is listed beneath the **Material Part Groups** node.



- Select each of the material part groups in turn and assign geometry parts to the *Parts* property.

If you switch the region to another multi-part solid physics continuum, STAR-CCM+ repopulates the **Material Part Groups** node with the solids selected in the newly assigned continuum.

In a multi-part solid simulation, each material appears in an associated region as a material part group node (**Fe** in the following screenshot). To assign parts, select the node of the desired material for access to its properties.



The *Parts* property of each of these objects opens the object selection dialog.

In this example, the **Fe** node has Chip3 and Chip8 selected, meaning that those chips are made of iron.

Parts can correspond to only one particular material at a time. If you reassign a part to another material, the part disappears automatically from the list for the previous material.

Material Part Group Properties

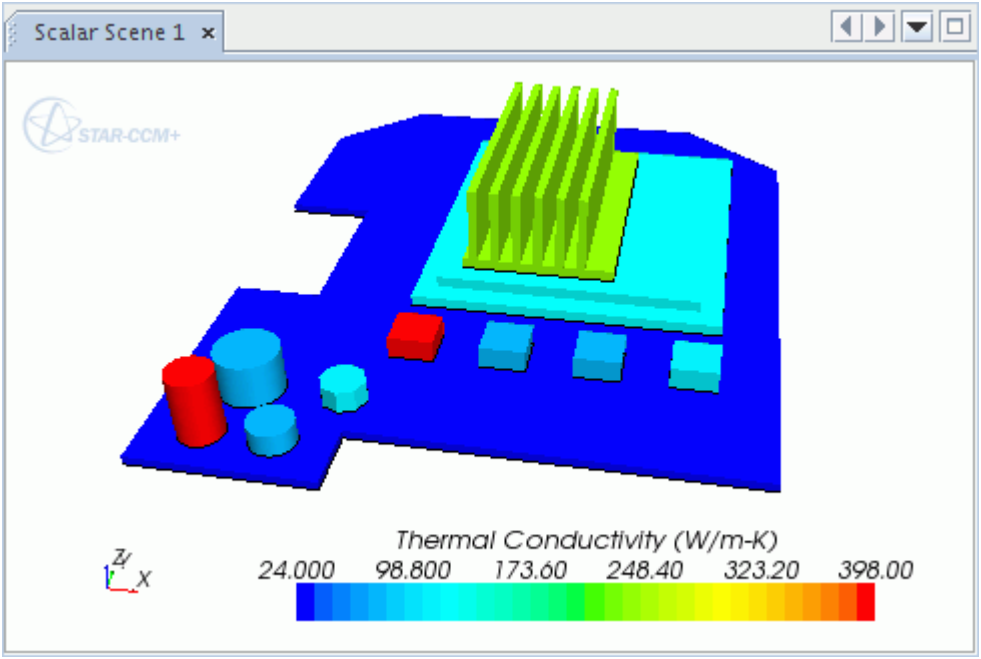
Parts

List of parts in this material part group. Clicking the property customizer (...) accesses the object selection dialog to add or remove parts.

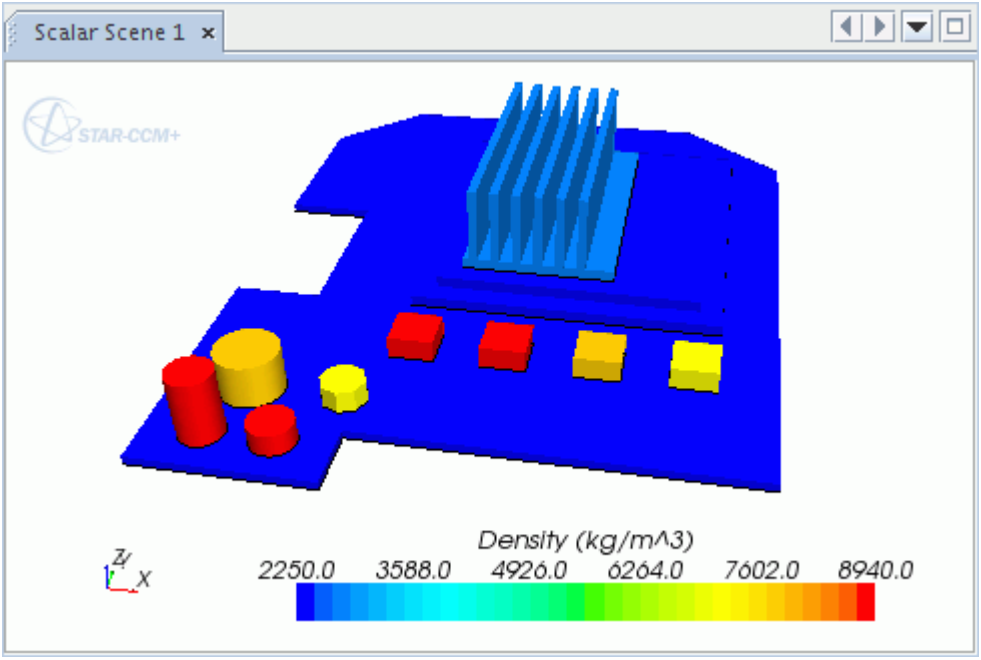
Post-Processing Multi-Part Solid Data

Various relevant field functions can be post-processed in scenes.

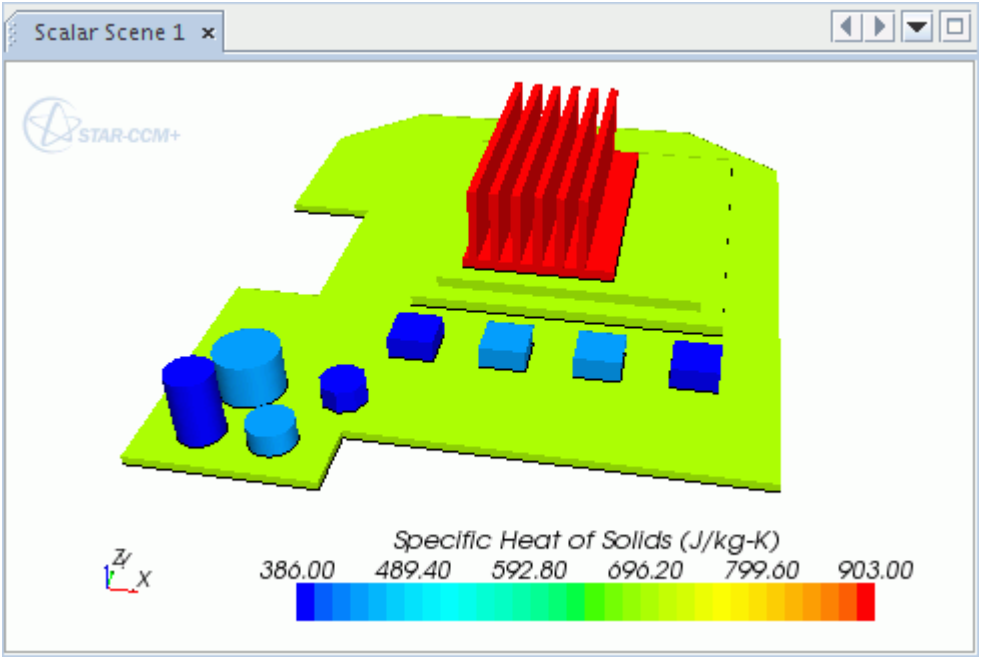
- Thermal Conductivity



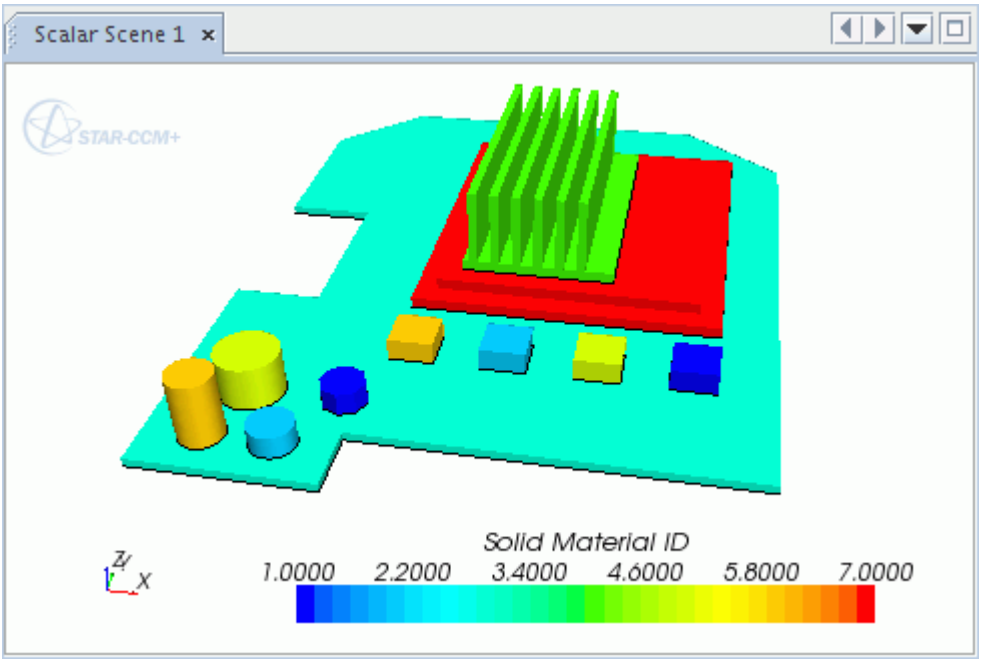
- Density



- Specific Heat of Solids



- Solid Material ID -- associating parts with materials visually



For example, suppose that you want to see which parts in the scene are made of zinc. Select the **Zn** component node in the **Solids** manager in the continuum model, and get the ID number, which in this case is 1. To associate that number with the color of certain parts in the scalar scene, look at the color bar in the scene -- the parts with the associated color are made of zinc.

Modeling a Multiphase Mixture

A multiphase mixture is a fluid that is composed of multiple phases: an immiscible mixture of two or more pure substances in either the gas or liquid phase.

Multiphase mixture modelling can be activated by using either of the following combination of models:

- The Eulerian Multiphase model and the Eulerian Multiphase Mixture model
- The Eulerian Multiphase model, the Volume of Fluid (VOF) model, and either the Laminar or the Turbulent models for *Viscous Regime*

Modeling Linear Isotropic Elastic Stress

In STAR-CCM+ the available solid stress material model is Linear Isotropic Elastic.

A material is considered elastic when its stress is linearly related to the strain. If an elastic material is loaded, and then unloaded, it returns to it original undeformed shape.

The deformation does not depend on the loading history or the rate of loading and the material exhibits no damping behavior on its own. In other words, it is a reversible process.

The Linear Isotropic Elastic material model works with the Solid Stress model in STAR-CCM+.

Theory	See Linear Isotropic Elastic Formulation.
Provided By	**[physics continuum]** > **Models** > *Constitutive Law*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Linear Isotropic Elastic**
Requires	*Material*: **Solid** or **Multi-Component Solid** *Space*: **Three Dimensional** *Optional Models*: **Solid Stress**

Linear Isotropic Elastic Formulation

The relationship between the stress and strain in an isotropic elastic material is:

$$\sigma = 2G\varepsilon + \lambda tr\left(\varepsilon\right)\delta \tag{153}$$

where G is the shear modulus and λ is Lame's modulus both with units of stress and δ is the Kronecker delta. Two independent parameters describe the material modulus parameters. Other commonly used parameters are Young's Modulus E, Poisson's ratio ν, and the bulk modulus K. Young's modulus and Poisson's ratio have a simple interpretation for an uniaxial stress loading typical in coupon material tests or a bar loaded only down its axis.

$$\sigma = \begin{bmatrix}\sigma_{xx} & 0 & 0 & 0 & 0 & 0\end{bmatrix}^T \tag{154}$$

In this case, the axial stress is the product of Young's modulus and the axial strain $\sigma_{xx} = E\varepsilon_{xx}$. As the bar is loaded in tension, it elongates down the axis, but it shrinks lateral to the axis. The amount of lateral shrinking is proportional to the axial elongation by the amount of Poisson's ratio:

$$\varepsilon_{yy} = \varepsilon_{zz} = -\nu\varepsilon_{xx} \tag{155}$$

For pure shear loading, the shear stress is product of the shear modulus and the engineering shear strain:

$$\sigma_{xy} = 2G\varepsilon_{xy} = G\gamma_{xy} \tag{156}$$

The bulk modulus relates the mean stress to the volumetric strain:

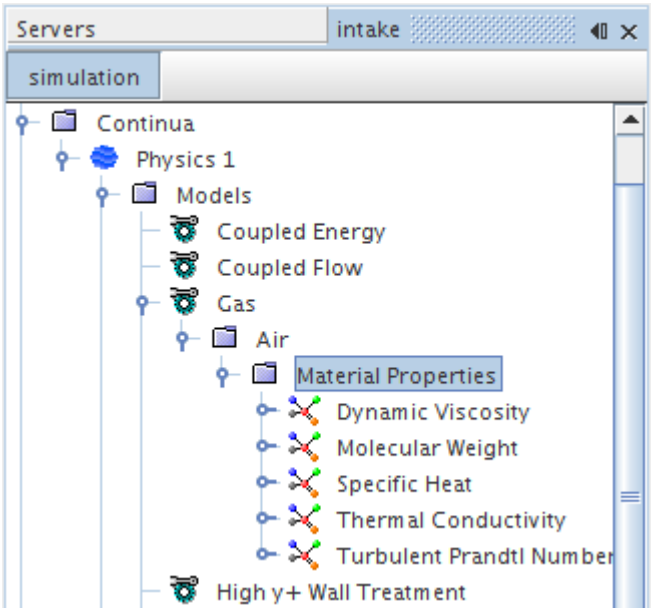
$$\sigma_m = \frac{\left(\sigma_{xx} + \sigma_{yy} + \sigma_{zz}\right)}{3} = K\left(\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}\right) \tag{157}$$

The elastic properties for a solid material are prescribed under Using Physics Continua.

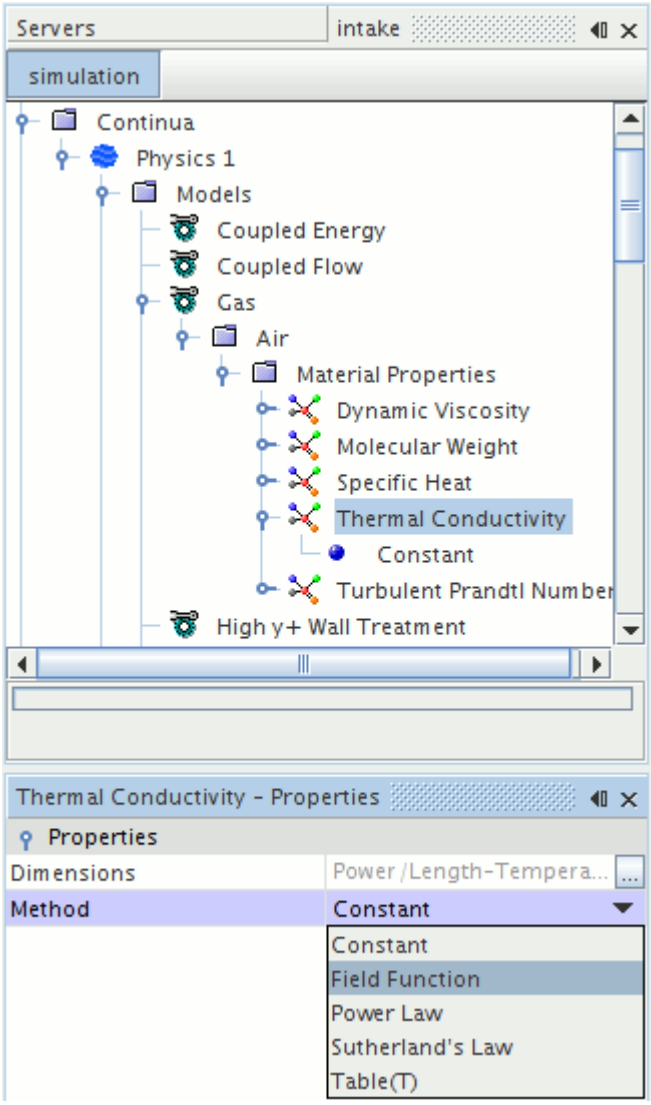
Setting Material Properties Methods

After the selection of an Equation of State model or a Viscous Regime, the **Material Properties** manager node becomes fully populated with material property nodes, which can be accessed through their properties.

Material properties that use polynomials can be plotted instantly.



To change the method, open the node for the relevant material property. In the *Properties* window, click the drop-down list corresponding to the *Method* property. This list permits the selection of **Constant**, **Field Function**, or one or more material property functions such as **Power Law**.



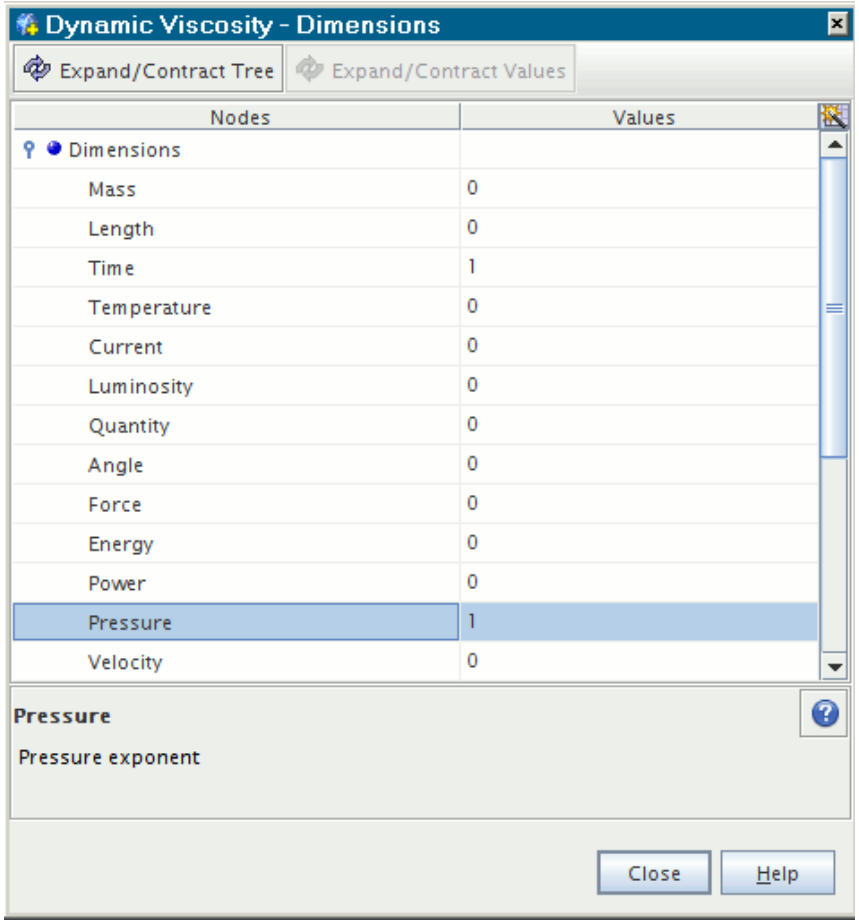
Contents:

Material Property Node Properties
Plotting Variable Material Properties
Using Constant Values
Using Field Functions
Using Polynomial in T
Using Thermodynamic Polynomial Data
Using IAPWS-IF97
Using Sutherland's Law
Using Table (Reaction Time)
Using Table(T)
Using Table(T,P)
Using the Density Table Method for Density Derivatives
Using the Enthalpy Table Method for Specific Heat

Using the Specific Heat Method for Enthalpy
Using the Enthalpy Difference Method for Heat of Vaporization
Using Entropy
Using Table(B,H) Method for Permeability
Using Tabular Data
Using the Power Law
Using the Temperature Shift Factor
Using the Non-Newtonian Generalized Models
Using the Schmidt Number
Using a Volume-Weighted Mixture
Using the Mass-Weighted Mixture
Using the Lewis Number
Using the PPDF Table
Using the PCFM Method
Using the Mixture Method for Critical Temperature and Pressure
Using the Mixture Method for Molecular Weight
Using the Mixture Method for Surface Tension
Using the Mathur-Saxena Averaging Method for Dynamic Viscosity
Using the Chapman-Enskog Method for Dynamic Viscosity
Using the Refutas Method for Dynamic Viscosity
Using the Mathur-Saxena Averaging Method for Thermal Conductivity
Using the Kinetic Theory Method for Thermal Conductivity
Using the Elemental Composition Method
Using the Anisotropic Method
Using the Dars-CFD Viscosity Method
Using the Dars-CFD Thermal Conductivity Method
Using the Gas Kinetics Method for Specific Heat
Using the Antoine Equation
Using the Wagner Equation
Using the Clausius-Clapeyron Method
Using the Heat of Formation
Using the Leidenfrost Temperature
Using the Standard State Temperature
Using the Binary Diffusion Coefficient Method for Molecular Diffusivity
Using the Kinetic Theory Method for Molecular Diffusivity
Using the Component Molecular Diffusivities Method for Molecular Diffusivity
Using the Nernst-Einstein Method for Charged Species
Using the Electrochemical Species method for Electrical Conductivity
Using the Thermal Diffusion Coefficient
Equilibrium Speed of Sound
Materials Bibliography

Material Property Node Properties

This section lists the settings for various material property nodes.

Dimensions	The dimensionality of the property that is expressed in terms of the basic dimensions, such as length, time (read-only). Accessing the customizer opens a *Dimensions* dialog to allow setting the dimensionality of a value or unit by setting the exponent of each of the base dimensions.  However, changing the dimensionality in this dialog has no effect on this property.	
Method	The method that is used to define the value for this material property.	
	Constant	The material property is input as constant. A Constant node is added as a child to this node. Applicable to all material properties.

	Field Function	The material property is entered using a field function. A Function node is added as a child to this node. Only applicable to some material properties.
	Power Law	Specifies the value using the Power Law. A node by that name is added as a child to this node. Only applicable to some material properties.
	Non-Newtonian Generalized Power Law	Specifies the value using the Non-Newtonian Generalized Power Law. A node by that name is added as a child to this node. Only applicable to dynamic viscosity when the models “Liquid” and “Laminar Flow” are selected.
	IAPWS-IF97	The material property is calculated using IAPWS-IF97. A node by that name is added as a child to this node. Only applicable to some material properties.
	Sutherland’s Law	The material property is entered using Sutherland’s Law. A node by that name is added as a child to this node. Only applicable to some material properties.
	Table (Reaction Time)	The material property is entered using table (reaction time). A node by that name is added as a child to this node. Only applicable to some material properties.
	Table(T)	The material property is entered using a table. A node by that name is added as a child to this node. Only applicable to some material properties.
	Tabular Data	The material property is entered using a table interpolation method. A **Tabular** node is added as a child to this node. Only applicable to some material properties.
	Polynomial in T	The material property is entered using a Polynomial in T. A node by

		that name is added as a child to this node. Only applicable to some material properties.
	Schmidt Number	The material property is entered using the Schmidt number. A node by that name is added as a child to this node. Only applicable to some material properties.
	Volume-Weighted Mixture	The material property is entered using the volume-weighted mixture. A node by that name is added as a child to this node. Only applicable to some material properties.
	Mass-Weighted Mixture	The material property is entered using the mass-weighted mixture. A node by that name is added as a child to this node. Only applicable to some material properties.
	Lewis Number	The material property is entered using the Lewis number. A node by that name is added as a child to this node. Only applicable to some material properties.
	PPDF Table	The material property is entered using the PPDF Table. A node by that name is added as a child to this node. Only applicable to some material properties.
	PCFM Table	The material property uses the PCFM Method. A node by that name is added as a child to this node. Only applicable to some material properties.
	Mixture	The material property uses a Mixture. A node by that name is added as a child to this node. Only applicable to some material properties.
	Anisotropic	The material property uses the Anisotropic method. A node by that name is added as a child to this node. Only applicable to some material properties.

	Multi-Component (DarsCFD)	The material property uses the Multi-Component (Dars-CFD) method. A node by that name is added as a child to this node. Only applicable to some material properties.
	Dars-CFD Viscosity Method	The material property uses the Dars-CFD Viscosity Method. A node by that name is added as a child to this node. Only applicable to some material properties.
	Dars-CFD Thermal Conductivity Method	The material property uses the Dars-CFD Thermal Conductivity Method. A node by that name is added as a child to this node. Only applicable to some material properties.
	Antoine Equation	The material property uses the Antoine Equation. A node by that name is added as a child to this node. Only applicable to the **Saturation Pressure** material property.
	Wagner Equation	The material property uses the Wagner Equation. A node by that name is added as a child to this node. Only applicable to the **Saturation Pressure** material property.
	Clausius-Clapeyron	The material property uses the Clausius-Clapeyron method. A node by that name is added as a child to this node. Only applicable to the **Latent Heat of Vaporization** material property with the **Antoine Equation**, **Wagner Equation**, or **Polynomial in T** methods for the **Saturation Pressure**.
	Linear	The solid volume fraction changes linearly with temperature between the solidus and liquidus temperatures (Eqn. (902)).

	Carman-Kozeny	The material property uses the Carman-Kozeny Mushy Permeability method. A node by that name is added as a child to this node. Only applicable to the **Mushy Zone Permeability** material property with the **Melting-Solidification** model.
	Metzner	The material property uses the Metzner method. A node by that name is added as a child to this node. Only applicable to the **Slurry Viscosity** material property with the **Melting-Solidification** model.
	Nernst-Einstein	The material property uses the **Nernst-Einstein** method. A node by that name is added as a child to this node. Only available for the Charged Species Mobility Material Property.
	Component Thermal Diffusion Coefficients	The material property uses the **Component Thermal Diffusion Coefficients** method. A node by that name is added as a child to this node. Only available for the Thermal Diffusion Coefficient Material Property.
	Warnatz Model	The material property uses the **Warnatz Model** method. A node by that name is added as a child to this node. Only available for the Thermal Diffusion Coefficient Material Property.

Plotting Variable Material Properties

When you work with a material property that uses a polynomial function, you can generate an approximate plot preview of the values before you run the simulation.

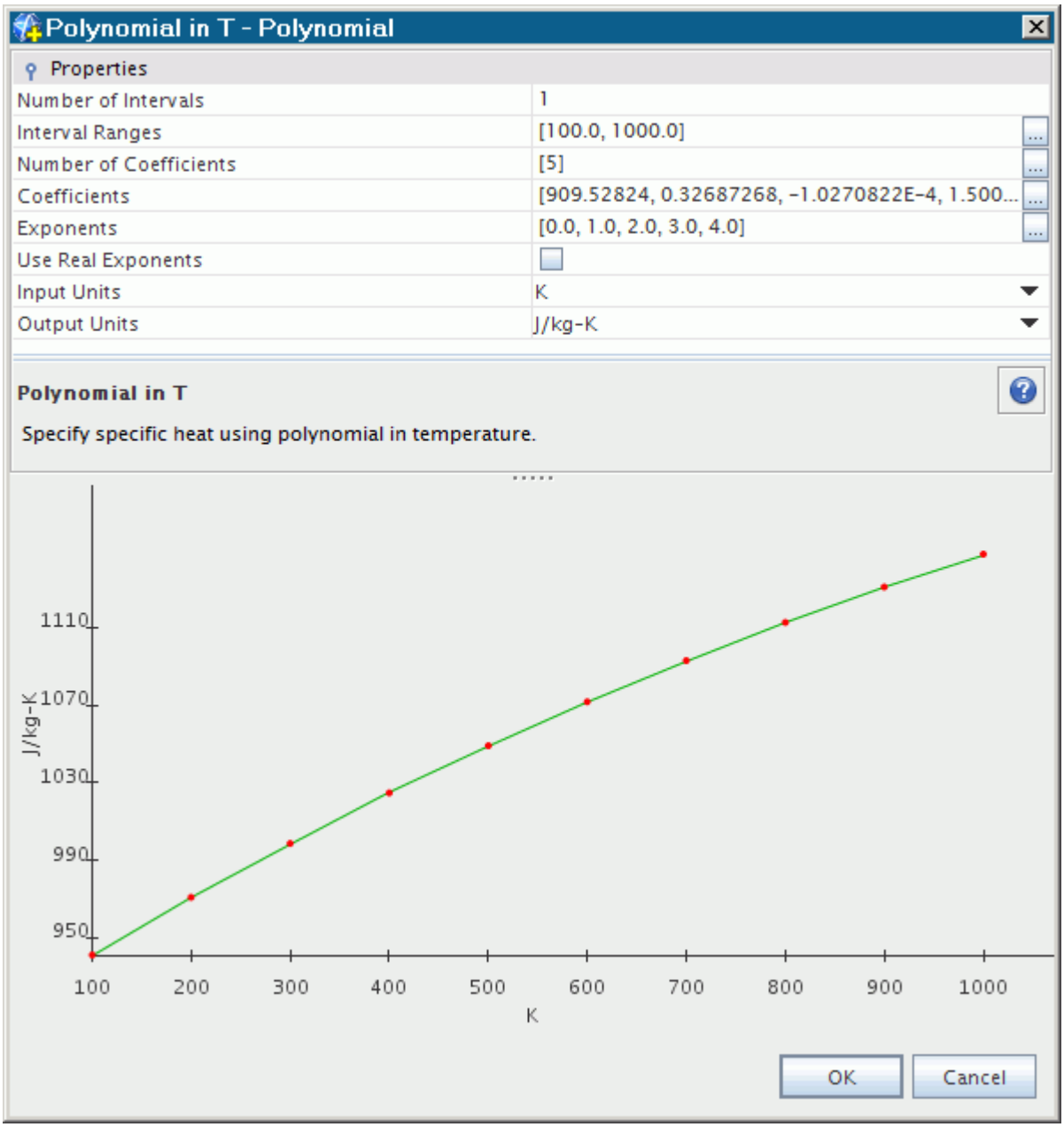
The “pre-processing check” lets you verify that you have entered the values correctly, or otherwise observe the behavior of the polynomial.

To generate the plot preview:

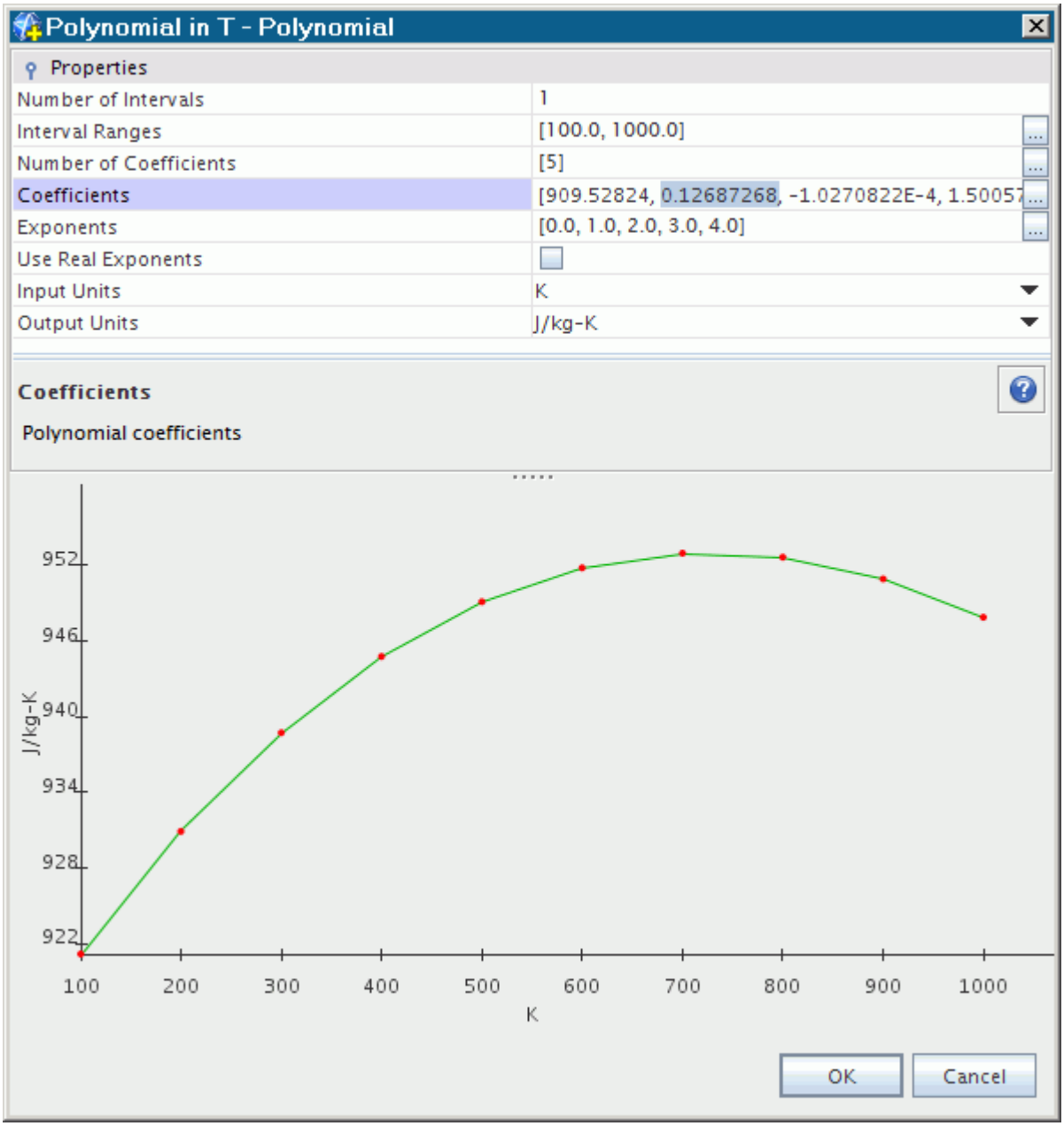
1. Select the node of a material property polynomial method. An example of such a node would be **Continua** > **Physics 1** > **Models** > **Gas** > **Air** > **Material Properties** > **Specific Heat** > **Polynomial in T**.

2. Click ... (**Custom Editor**) in the Polynomial property.

A customizer dialog that includes a graph opens.



As you modify the values, the graph changes in response to your entries.

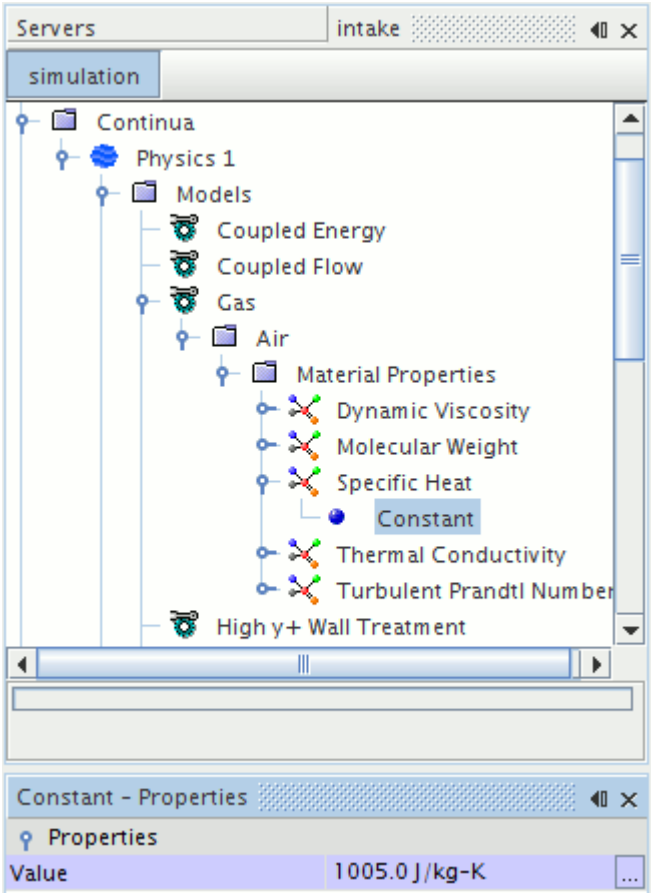


You can also choose units. The *Input Units* property specifies units for the x-axis, while the *Output Units* property does so for the y-axis. Unlike constant values, the polynomial coefficients and polynomial ranges (and so x- and y-axes on the polynomial plot) do not rescale when you change the units. These unit selections serve as an indicator of the units that STAR-CCM+ uses when calculating quantities with this polynomial.

Using Constant Values

Generally, most material property values are specified as constants.

The constant value is entered as a scalar value in the properties as shown below for **Specific Heat**.



Constant Value Properties

Constant value	The constant scalar value for the material property.

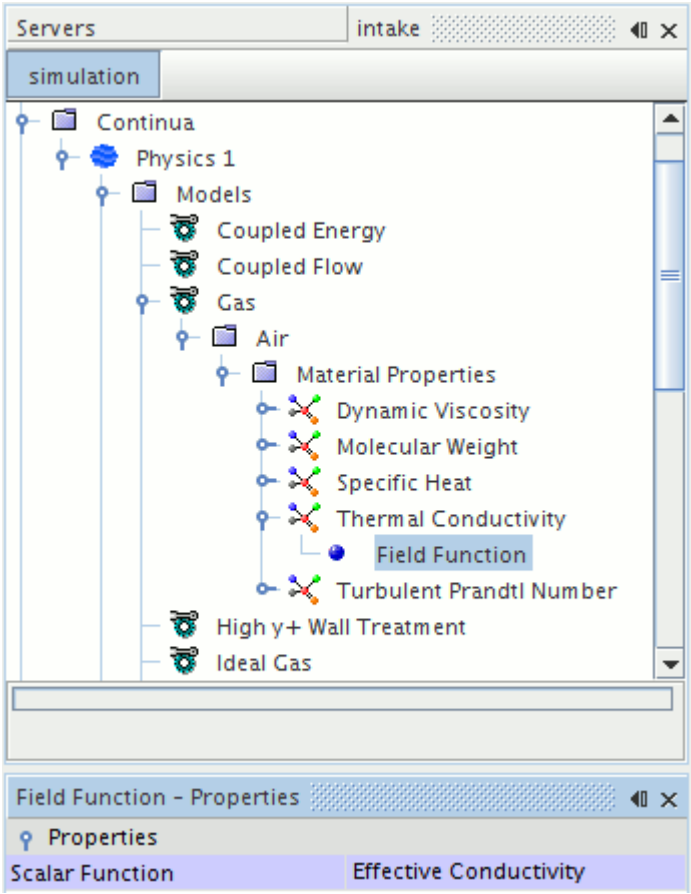
Using Field Functions

Field functions can be used to create non-standard functions for material properties.

Suppose that a scalar field function that is called **Effective Conductivity** is defined as

```
1.5207e-11 * pow($Temperature,3) - 4.857e-8 * pow($Temperature,2)
+ 1.0184e-4 * $Temperature - 3.9333e-4
```

This field function would then be selected in the list corresponding to the *Scalar Function* property in the *Properties* window.



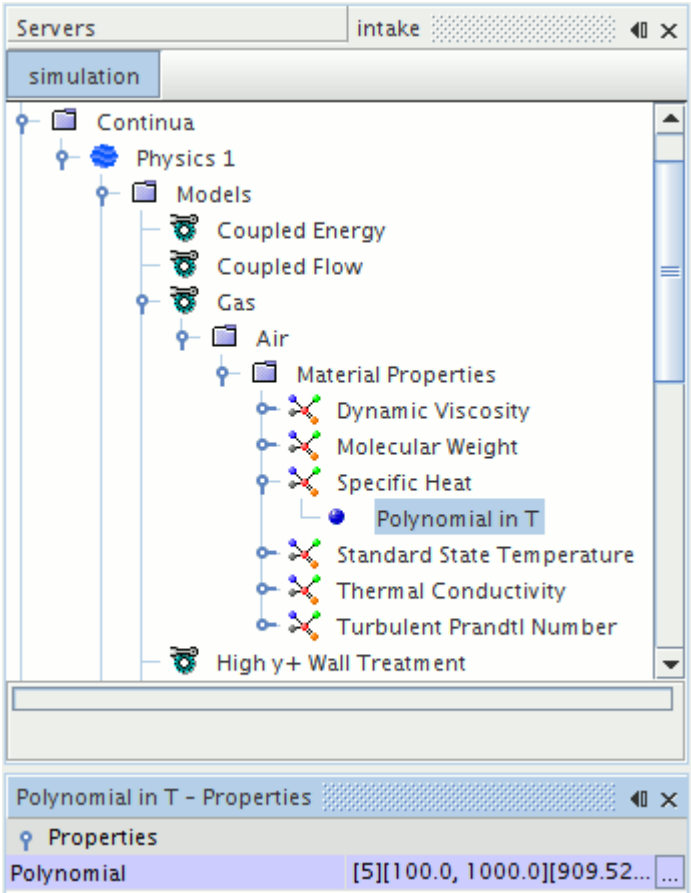
Field Function Value Properties

Scalar Function	The field function that is used to define this material property. A list of available scalar functions is provided.

Using Polynomial in T

This polynomial function is available for certain material properties.

The properties for this function are set in the *Properties* window.



Temperature polynomials are specified in piece-wise fashion where each interval i spans the range Tr_i to Tr_{i+1}, has m_i number of terms, and is of the form:

$$X_{(T)} = \sum_{k=0}^{m_i} a_{i,k} \cdot T^{bi,k}; \; Tr \leq T \leq Tr_{i+1}$$

The polynomial has n intervals with interval ranges:

$$Tr_0, Tr_1, Tr_2, ..., Tr_{n+1}$$

The number of terms in each interval are:

$$m_0, m_1, m_2, ..., m_n$$

The exponents are:

$$b_{0,0}, b_{0,1}, b_{0,2}, ..., b_{0,m_0}$$

$$b_{1,0}, b_{1,1}, b_{1,2}, ..., b_{1,m_1}$$

$$\vdots \quad \vdots \quad \vdots \quad \vdots$$

$$\vdots \quad \vdots \quad \vdots \quad \vdots$$

$$b_{n,0}, b_{n,1}, b_{n,2}, ..., b_{n,m_n}$$

The coefficients for the polynomial are:

$a_{0,0}, a_{0,1}, a_{0,2}, \ldots, a_{0,m_0}$

$a_{1,0}, a_{1,1}, a_{1,2}, \ldots, a_{1,m_1}$

$\vdots \quad \vdots \quad \vdots \quad \vdots$

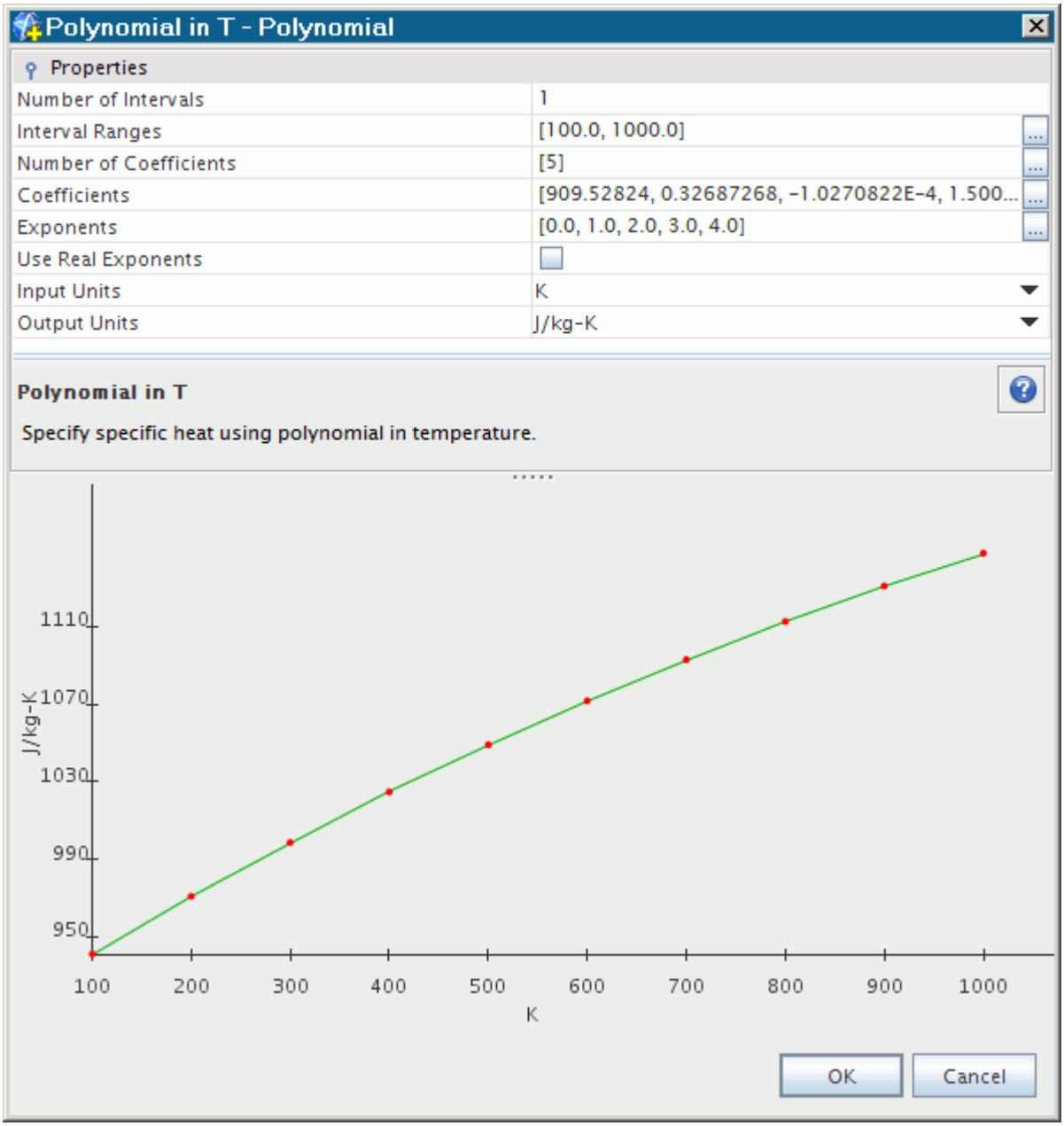
$\vdots \quad \vdots \quad \vdots \quad \vdots$

$a_{n,0}, a_{n,1}, a_{n,2}, \ldots, a_{n,m_n}$

Specify a temperature range of the polynomial sufficiently large such that the polynomial is not evaluated outside the range during the analysis. If the polynomial is ever evaluated outside the range, the property is assigned a value of the polynomial at the respective range limit.

Polynomial in T Properties

In the *Properties* window, a single property that is called *Polynomial* includes a property customizer (...). Click that button to activate a customizer dialog with properties for working with polynomials in STAR-CCM+.



The plot in this dialog gets updated instantly with your changes. You can select units separately for each axis: *Input Units* for the x-axis, and *Output Units* for the y-axis. Unlike constant values, the polynomial coefficients and polynomial ranges (and the corresponding x- and y-axes on the polynomial plot) do not rescale when you change the units. These unit selections serve as an indicator of the units that STAR-CCM+ uses when calculating quantities with this polynomial.

For the *Coefficients* property, commas separate the entries and the coefficients are in the following order:

$$C_p(T) = C_{p0} + \left(C_{p1} \cdot T\right) + \left(C_{p2} \cdot T^2\right) + \ldots + \left(C_{pn} \cdot T^n\right)$$

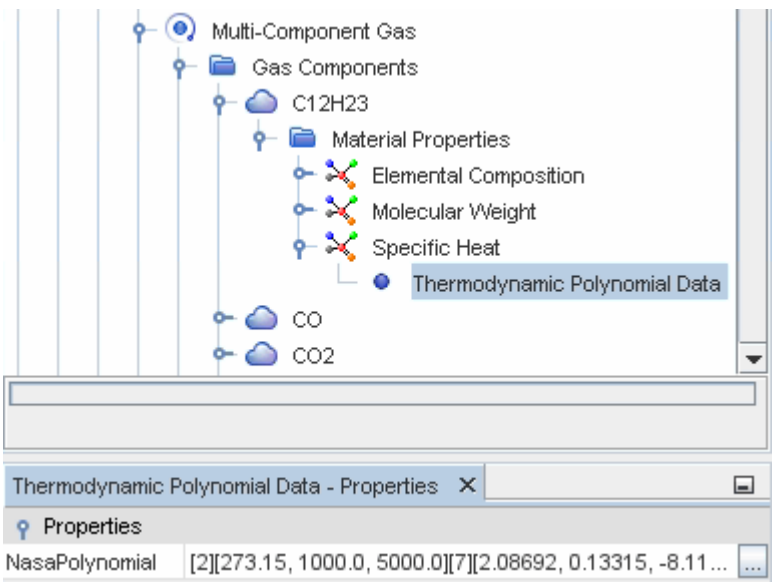
A separate section describes an example of working with these properties.

Note: When using a polynomial in T to specify the specific heat of a material, you are advised to define the polynomial on an absolute temperature scale by selecting either **K** and **J/kg-K** or **R** and **Btu/lb-R** as the values for *Input Units* and *Output Units* respectively. These selections avoid any difficulties that are associated with the integrations used to evaluate certain thermodynamic properties such as entropy and total pressure when $C_p(T)$ is specified on a non-absolute temperature scale that can pass through $T = 0$.

Using Thermodynamic Polynomial Data

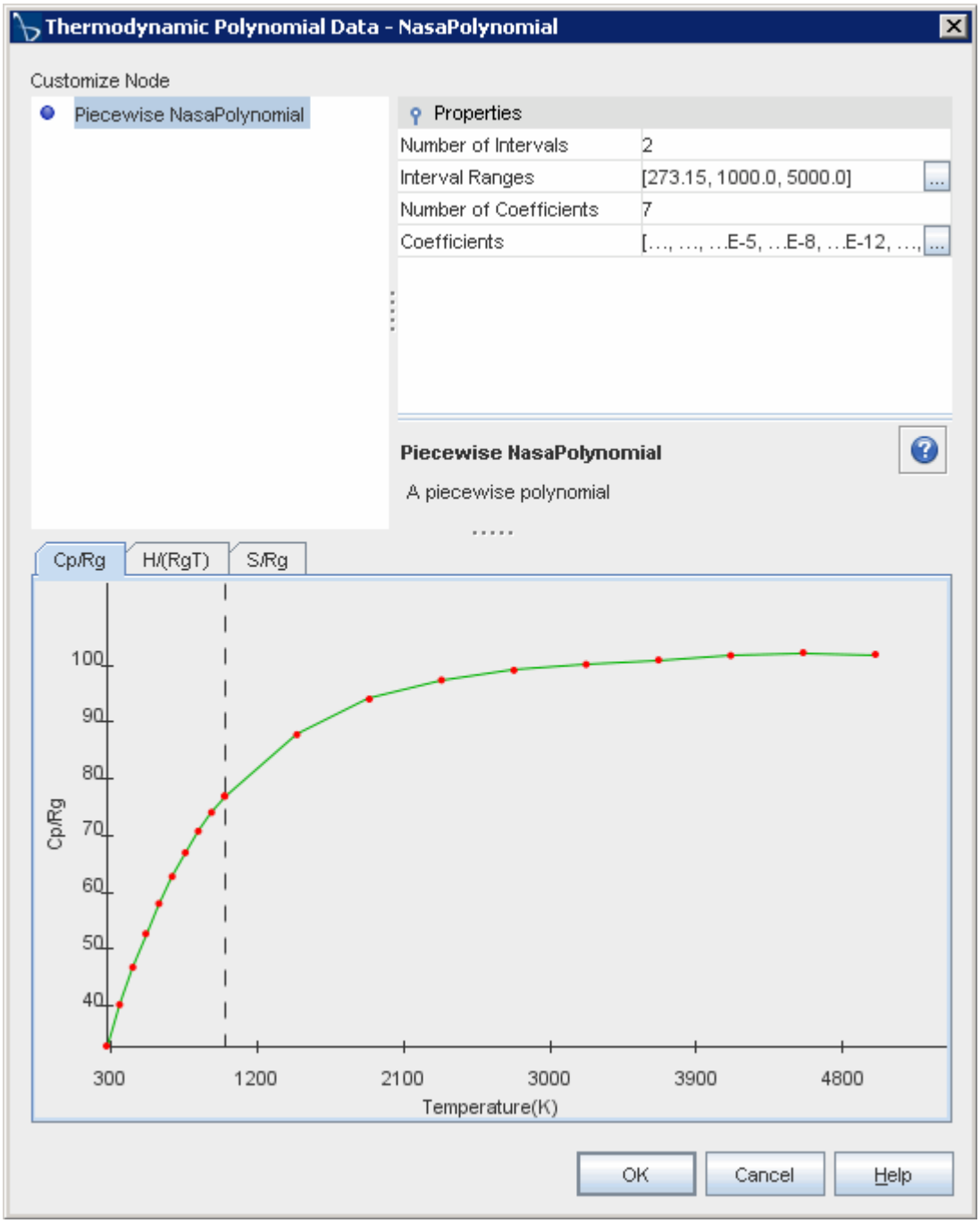
Thermodynamic Polynomial Data provides seven (or nine) coefficients—five (or seven) coefficient polynomial curve fits for specific heat C_p, a coefficient polynomial curve fit for static enthalpy h, and a coefficient polynomial curve fit for entropy S.

The curve fits are read in from either the thermodynamic properties file that is imported (see Gas / Surface Thermodynamic Properties Files), or the selected material database (most often from the standard material database, `props.mdb`, that is provided as a part of STAR-CCM+).



Since enthalpy of formation $h_f{}^0$ and reference temperature T_{ref} are implicit in the curve fit for h, you do not need to supply them as you would if you were to use the Polynomial in T method for C_p.

In the *Properties* window, a single property that is called *NasaPolynomial* includes a property customizer (...). Click that button to activate a customizer dialog with properties for working with polynomials in STAR-CCM+.



The plot in this dialog gets updated instantly with your changes. Separate plots, accessible through tabs, exist for C_p, h, and S.

Note: If STAR-CCM+ ever evaluates the polynomial outside the range, it assigns to the property a value of the polynomial at the respective range limit. Depending on the demands of the particular simulation, you can specify a temperature range of the polynomial large enough that the polynomial is not evaluated outside the range during the analysis. However, sometimes the polynomials are not valid beyond the range that is given, so manipulate the temperature range with caution.

Until STAR-CCM+ version 5.04, the values of the polynomial coefficients were scaled with $R_u/M_r(X)$, where R_u is the universal gas constant [J/(kmol K)] and $M_r(X)$ is the molecular weight (kg/kmol) of the substance X. They appeared scaled in this manner in the GUI. Now the values appear unscaled, as they are found in the literature.

7-Coefficient NASA Thermodynamic Polynomials

For 7-coefficient NASA thermodynamic polynomials, the coefficients a_1 through a_5 for various temperature ranges define the specific heat capacity, C_p, as follows:

$$\frac{C_p}{R_u} = a_1 + a_2T + a_3T^2 + a_4T^3 + a_5T^4 \tag{158}$$

These coefficients can be set with a customizer dialog similar to the dialog used for polynomial properties. However, since it can be difficult to make them consistent, it is advisable not to change the default numbers.

The curve fit is:

$$\frac{h}{R_uT} = a_1 + \frac{a_2}{2}T + \frac{a_3}{3}T^2 + \frac{a_4}{4}T^3 + \frac{a_5}{5}T^4 + \frac{a_6}{T} \tag{159}$$

where:

$$h = \int_{T_{ref}}^{T} C_p(T)\ \mathrm{d}\ T + h_f{}^0\Big|_{T_{ref}} \tag{160}$$

The coefficients for various temperature ranges define S as follows:

$$\frac{S}{R_u} = a_1\ln T + a_2T + \frac{a_3T^2}{2} + \frac{a_4T^3}{3} + \frac{a_5T^4}{4} + a_7 \tag{161}$$

9-Coefficient NASA Thermodynamic Polynomials

For 9-coefficient NASA thermodynamic polynomials, the coefficients a_1 through a_7 for various temperature ranges define C_p as follows:

$$\frac{C_p}{R_u} = a_1T^{-2} + a_2T^{-1} + a_3 + a_4T + a_5T^2 + a_6T^3 + a_7T^4 \tag{162}$$

These coefficients can be set with a customizer dialog similar to the dialog used for polynomial properties. However, since it can be difficult to make them consistent, it is advisable not to change the default numbers.

The curve fit is:

$$\frac{h}{R_uT} = -a_1T^{-2} + \frac{a_2\ln T}{T} + a_3 + \frac{a_4T}{2} + \frac{a_5T^2}{3} + \frac{a_6T^3}{4} + \frac{a_7T^4}{5} + \frac{a_8}{T} \tag{163}$$

where:

$$h = \int_{T_{ref}}^{T} C_p(T)\ \mathrm{d}\ T + h_f^{\ 0}\Big|_{T_{ref}} \tag{164}$$

The coefficients for various temperature ranges define the entropy, S, as follows:

$$\frac{S}{R_u} = \frac{-a_1 T^{-2}}{2} - a_2 T^{-1} + a_3 \ln T + a_4 T + \frac{a_5 T^2}{2} + \frac{a_6 T^3}{3} + \frac{a_7 T^4}{4} + a_9 \tag{165}$$

Troubleshooting Thermodynamic Polynomial Data

It is expected that the thermodynamic NASA polynomials provide continuous data and as such, if a discontinuity in data is detected, STAR-CCM+ attempts to correct the coefficients internally. If STAR-CCM+ corrects any coefficients, the corrected coefficients are displayed in the GUI instead of the coefficients that were specified. However, if STAR-CCM+ cannot correct the coefficients automatically (for a large discontinuity in data), a warning is displayed in the Output window and the coefficients must be fixed manually.

Contents:

Modifying a Copy of the Chemkin Database
Loading a Modified Copy of the Chemkin Database
Chemkin Database Format

Modifying a Copy of the Chemkin Database

The Chemkin database file (`thermdat.dbs`) is essentially a text file, and can be modified with any text editor.

Caution: The `thermdat.dbs` file is the source of the Chemkin database. Before modifying it, make a copy of this file for modification, and keep the original in a safe place.

The file is located as follows (these locations are examples using STAR-CCM+ version 7.04.006):

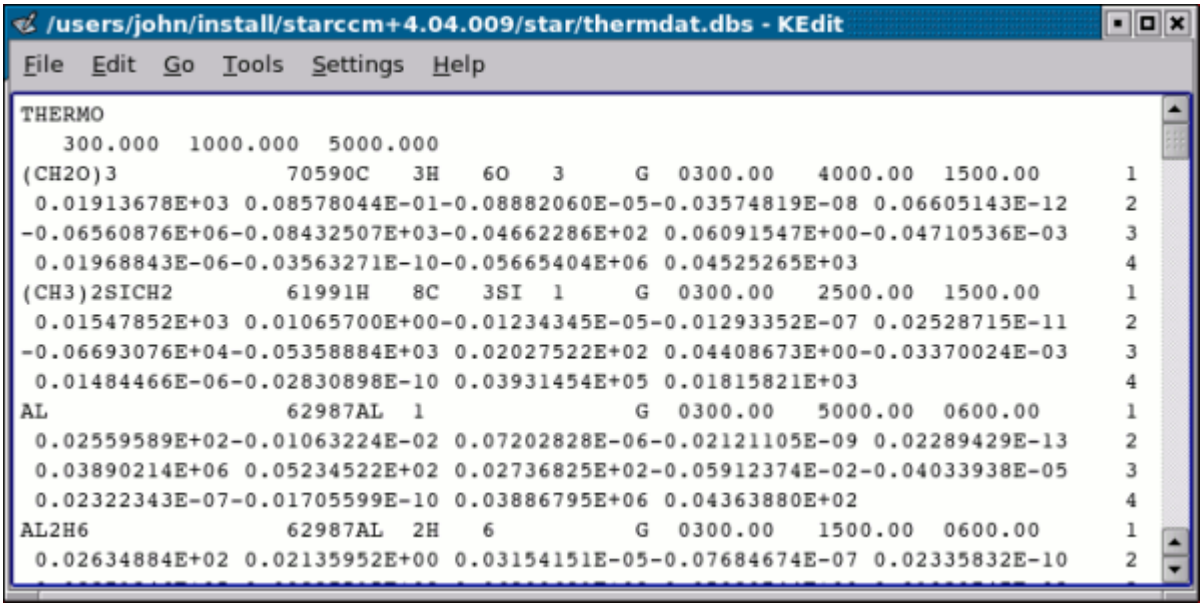
- On Unix:

```
[INSTALL.DIR]/cd-adapco/starccm+7.04.006/star
```

- On Windows:

```
C:\Program Files\CD-adapco\STAR-CCM+ 7.04.006
```

Open your copy of the `thermdat.dbs` file in a text editor.



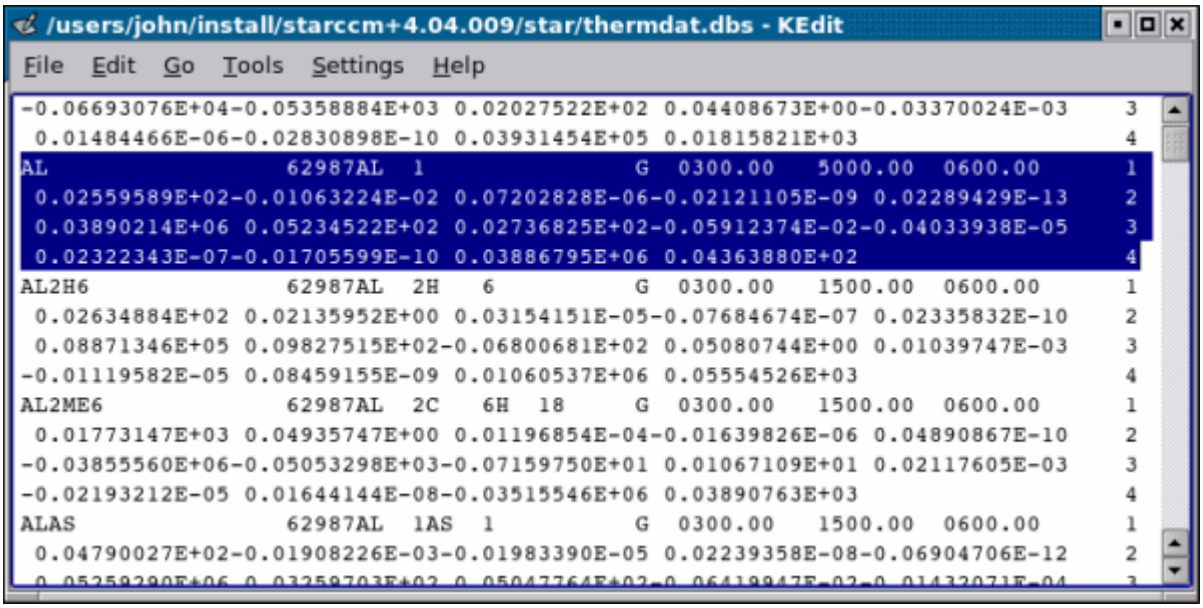

```
THERMO
   300.000  1000.000  5000.000
(CH2O)3            70590C   3H   6O   3     G  0300.00   4000.00  1500.00      1
 0.01913678E+03 0.08578044E-01-0.08882060E-05-0.03574819E-08 0.06605143E-12    2
-0.06560876E+06-0.08432507E+03-0.04662286E+02 0.06091547E+00-0.04710536E-03    3
 0.01968843E-06-0.03563271E-10-0.05665404E+06 0.04525265E+03                   4
(CH3)2SICH2        61991H   8C   3SI  1     G  0300.00   2500.00  1500.00      1
 0.01547852E+03 0.01065700E+00-0.01234345E-05-0.01293352E-07 0.02528715E-11    2
-0.06693076E+04-0.05358884E+03 0.02027522E+02 0.04408673E+00-0.03370024E-03    3
 0.01484466E-06-0.02830898E-10 0.03931454E+05 0.01815821E+03                   4
AL                 62987AL  1               G  0300.00   5000.00  0600.00      1
 0.02559589E+02-0.01063224E-02 0.07202828E-06-0.02121105E-09 0.02289429E-13    2
 0.03890214E+06 0.05234522E+02 0.02736825E+02-0.05912374E-02-0.04033938E-05    3
 0.02322343E-07-0.01705599E-10 0.03886795E+06 0.04363880E+02                   4
AL2H6              62987AL  2H   6          G  0300.00   1500.00  0600.00      1
 0.02634884E+02 0.02135952E+00 0.03154151E-05-0.07684674E-07 0.02335832E-10    2
```

This editing technique lets you add new species as well as modify species properties.

Adding New Species to the Database

Gaseous species and condensed species with only one condensed phase can be in any order. However, the sets for two or more condensed phases of the same species must be adjacent. If there are more than two condensed phases of a species, their sets must be either in increasing or decreasing order according to their temperature intervals.

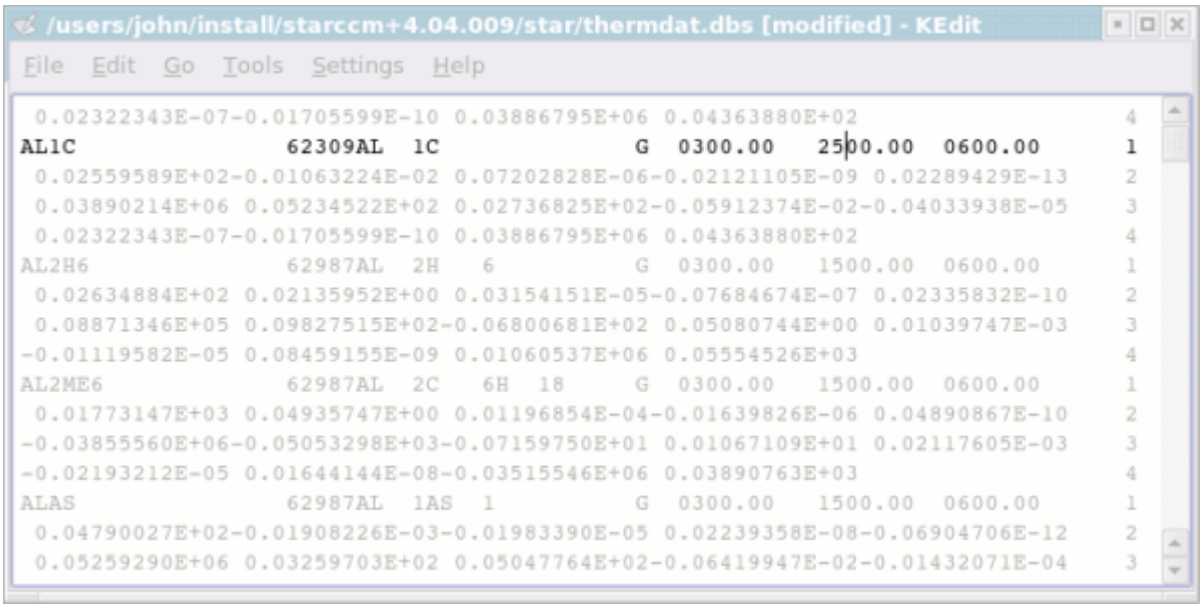
With the file open in the text editor, scroll to a set of data for any species, preferably one that belongs to the same phase as the species you wish to add.




```
-0.06693076E+04-0.05358884E+03 0.02027522E+02 0.04408673E+00-0.03370024E-03    3
 0.01484466E-06-0.02830898E-10 0.03931454E+05 0.01815821E+03                   4
AL                 62987AL  1               G  0300.00   5000.00  0600.00      1
 0.02559589E+02-0.01063224E-02 0.07202828E-06-0.02121105E-09 0.02289429E-13    2
 0.03890214E+06 0.05234522E+02 0.02736825E+02-0.05912374E-02-0.04033938E-05    3
 0.02322343E-07-0.01705599E-10 0.03886795E+06 0.04363880E+02                   4
AL2H6              62987AL  2H   6          G  0300.00   1500.00  0600.00      1
 0.02634884E+02 0.02135952E+00 0.03154151E-05-0.07684674E-07 0.02335832E-10    2
 0.08871346E+05 0.09827515E+02-0.06800681E+02 0.05080744E+00 0.01039747E-03    3
-0.01119582E-05 0.08459155E-09 0.01060537E+06 0.05554526E+03                   4
AL2ME6             62987AL  2C   6H  18     G  0300.00   1500.00  0600.00      1
 0.01773147E+03 0.04935747E+00 0.01196854E-04-0.01639826E-06 0.04890867E-10    2
-0.03855560E+06-0.05053298E+03-0.07159750E+01 0.01067109E+01 0.02117605E-03    3
-0.02193212E-05 0.01644144E-08-0.03515546E+06 0.03890763E+03                   4
ALAS               62987AL  1AS  1          G  0300.00   1500.00  0600.00      1
 0.04790027E+02-0.01908226E-03-0.01983390E-05 0.02239358E-08-0.06904706E-12    2
```

Select that entire set of data and use the Copy command.

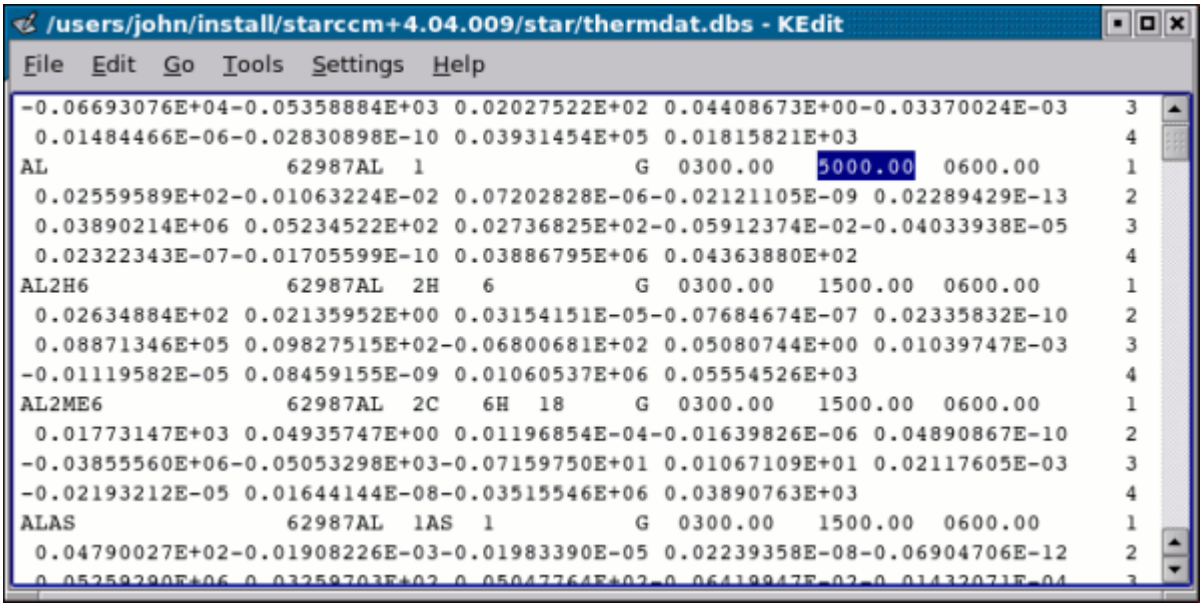
Then paste that data into an appropriate location, and change the entries so that they reflect the data of your species. In order for STAR-CCM+ to read your information correctly, ensure that the spacing is the same as that of other entries.



To complete the addition of a new species, enter data for your species in the proper positions. The details of how various types of data appear in the file are described in the section on the species database format.

Modifying Properties in the Database

Once you have located the value that you want to change, scroll to that value and change it to the new one. In order for STAR-CCM+ to read your information correctly, ensure that the spacing is the same as that of other entries.



Loading a Modified Copy of the Chemkin Database

After you have modified a copy of the Chemkin database, you can get STAR-CCM+ to recognize the changes by loading the `thermdat.dbs` file.

You can load the `thermdat.dbs` file one of three ways:

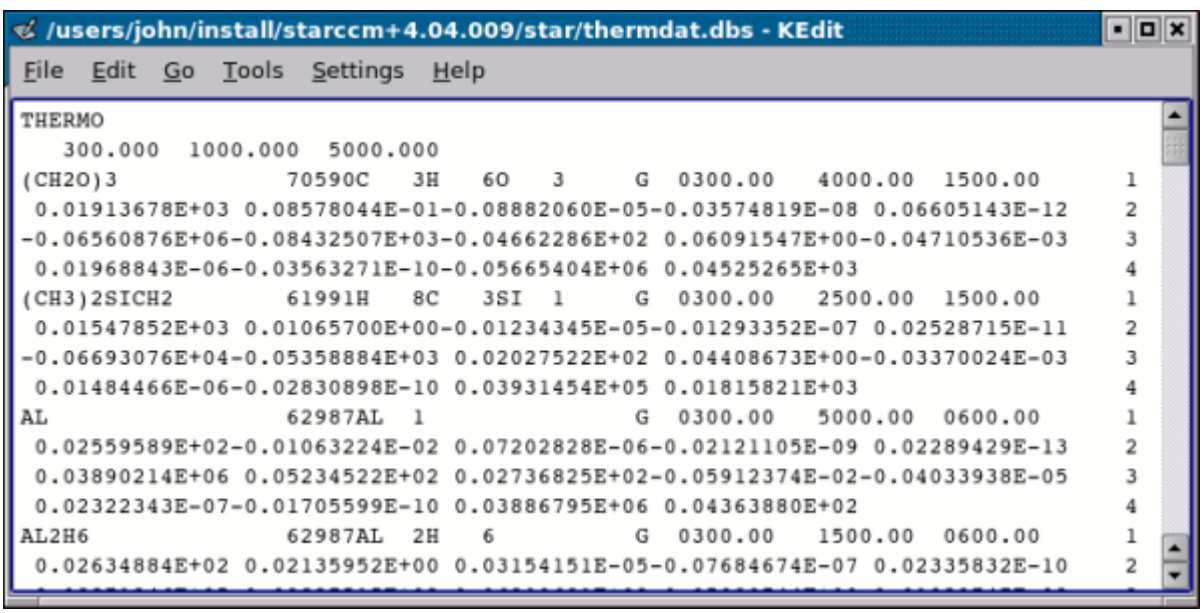
- In the *File Path* property of the **Tools** > **Materials Database** > **[material database]** node, enter the absolute path of your `thermdat.dbs` file. This works in a manner similar to loading material databases.
- If you are using the Complex Chemistry model with the DARS-CFD ODE solver, you can use a special dialog to import the Chemkin file. This feature also lets you use multiple `.dbs` files.
- In your file management program, move the original `thermdat.dbs` file and transfer the copy in its place, making sure that the copy has the same name, `thermdat.dbs`. Be careful not to overwrite one with the other.

Chemkin Database Format

The information in the `thermdat.dbs` file is organized in a precise structure so that STAR-CCM+ can read it into the simulation.

In order for STAR-CCM+ to read the information correctly, any modifications must follow a precise syntax of position and spacing. Never modify the original `thermdat.dbs` file that is supplied with STAR-CCM+. Make a copy of the file for modifications and keep the original somewhere safe.

The following screenshot shows an example of the top of a `thermdat.dbs` file in a text editor.



The file begins as follows:

- The top line starts the file with THERMO.
- The second line from the top contains temperature ranges for two sets of coefficients: lowest, common, and highest.

The body of the file contains data for the species. Each set of data for a particular species is composed of four lines, numbered 1 through 4 at the end:

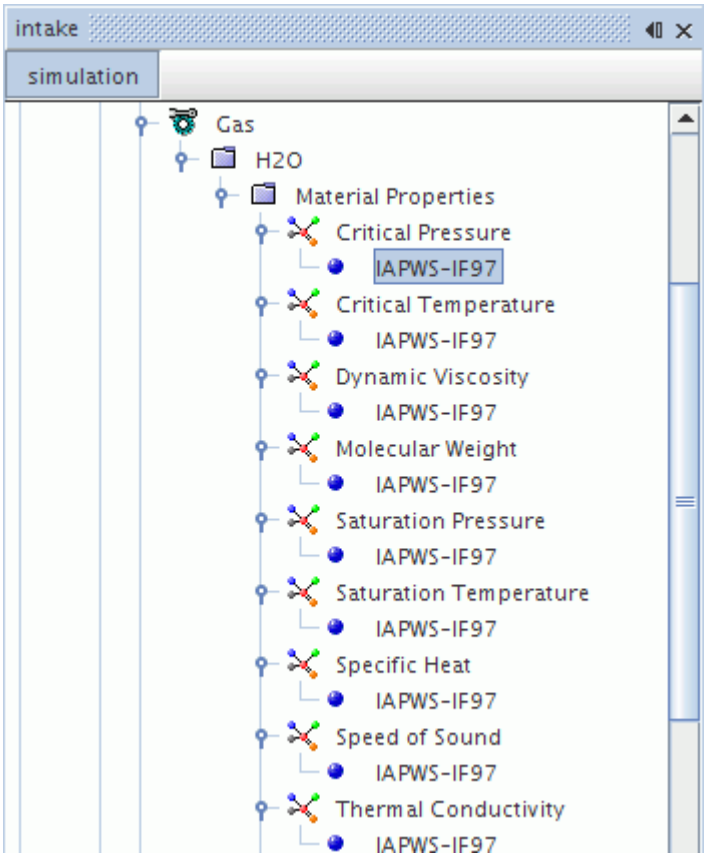
- Line 1 consists of:
  - The name of the species
  - The date
  - The atomic symbols and formula of the species
  - The phase (G = gas; L = liquid; S = solid)
  - The temperature range
- Line 2 contains coefficients a_1 through a_5 for the upper temperature interval.
- Line 3 contains coefficients a_6 and a_7 for the upper temperature interval, and coefficients a_1 through a_3 for the lower temperature interval.
- Line 4 contains coefficients a_4 through a_7 for the lower temperature interval.

After the last set of numbered lines, the bottom line terminates the file with END.

Using IAPWS-IF97

The IAPWS-IF97 specification is available for various material properties.

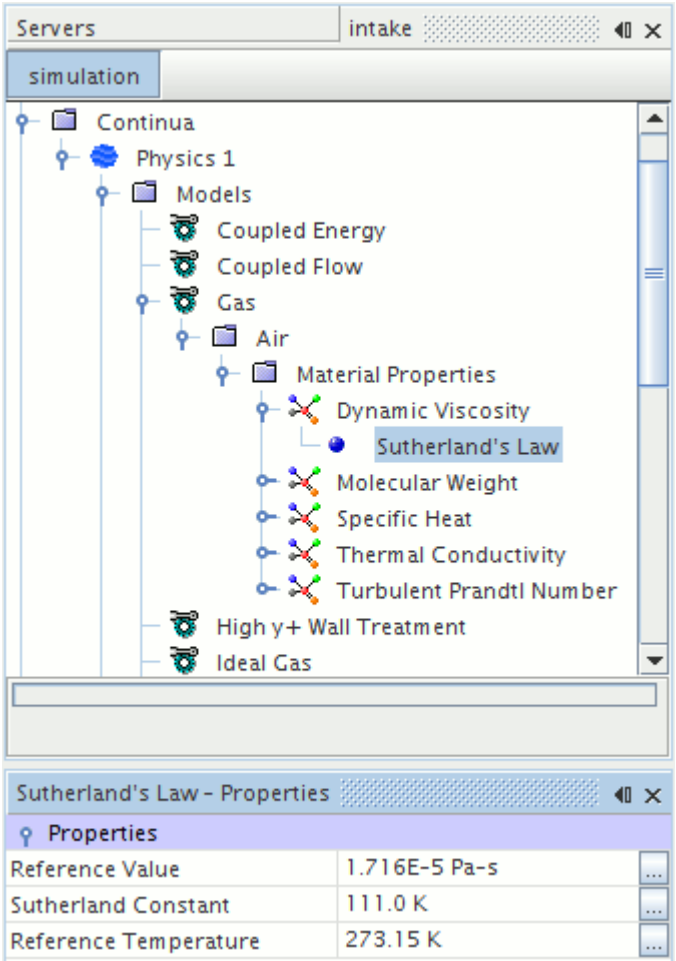
This method node has no properties. It acts simply as an indicator that the associated material property follows the IAPWS-IF97 specification [21], [22], [23]. This node is activated automatically when you select one of the IAPWS-IF97 continuum models.



Using Sutherland's Law

The Sutherland's law function is available for various material properties.

The coefficient properties for this function are set in the *Properties* window.



Sutherland's law for viscosity is:

$$\frac{\mu}{\mu_0} = \left(\frac{T}{T_0}\right)^{\frac{3}{2}}\left(\frac{T_0 + S}{T + S}\right) \tag{166}$$

where T_0 and μ_0 are the reference temperature and viscosity, respectively, and S is the Sutherland constant.

Sutherland's law for thermal conductivity is:

$$\frac{k}{k_0} = \left(\frac{T}{T_0}\right)^{\frac{3}{2}}\left(\frac{T_0 + S}{T + S}\right) \tag{167}$$

where T_0 and k_0 are the reference temperature and conductivity, respectively, and S is the Sutherland constant.

Sutherland's Law Properties

Reference Temperature	The reference temperature in Eqn. (166) or Eqn. (167) (depending on the material property).	

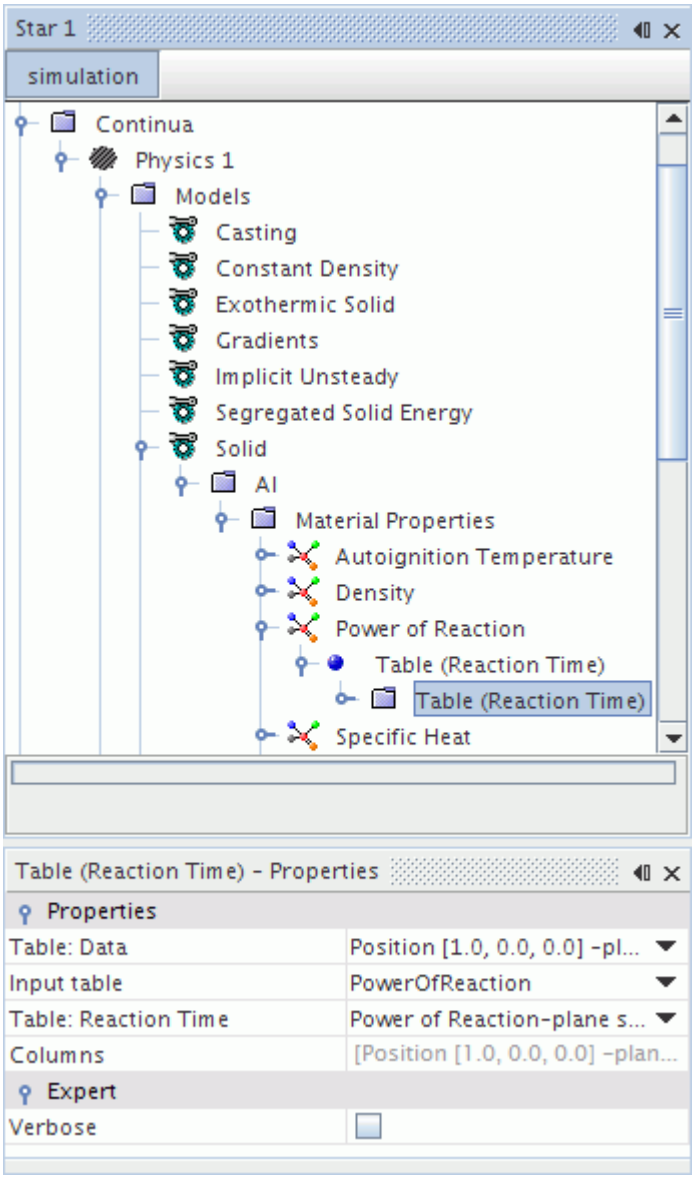
Reference Value	The reference viscosity or conductivity (depending on the material property) in Eqn. (166) or Eqn. (167) .
Sutherland Constant	The Sutherland Constant value in Eqn. (166) or Eqn. (167) (depending on the material property).

Using Table (Reaction Time)

A (Reaction Time) table has a column of Reaction Time values that specify the physical time of the reaction, in seconds, of each point (row) in the table.

The points must be ordered by increasing values of time. By default, the linear-piecewise method is used for interpolating between data points in the table. The properties for this function are set in the *Properties* window.

This type of table is only applicable to unsteady calculations using the Exothermic Solid model, where “Reaction Time” is defined as the accumulated physical time for which the temperature exceeds the autoignition time.



Note: A profile in Reaction Time can also be specified analytically with a user-defined field function that uses the `$ReactionTime` variable.

Table (Reaction Time) Properties

Table: Data	Identifies the selected column in the table file as the `Data` (for example, Position) column.
Table: Reaction Time	Identifies the selected column in the table file as the `Reaction Time` column.
Input Table	Selects a table from among the ones that appear as nodes in the **Tables** node of the **Tools** node.
Columns	Displays headings of columns in the table (read-only).

Table (Reaction Time) Expert Properties

Table Format	Read-only. Reports the format of the table data. Before the data is read and processed, the state is **Unprocessed**. Afterward, the state is one of these three: • **Cartesian**: Data is stored on equally spaced grid points. • **Structured**: Data is stored on grid points that are regularly spaced but not equally spaced. • **Unstructured**: Data is stored on irregularly spaced grid points. See Cartesian and Structured Tables.
Verbose	Provides more output for this value while the simulation is running. This property can be useful for debugging problems when they occur.
Cubic Interpolation	When **On**, provides smoother interpolation between table values than the default linear-piecewise method, and reduces simulation iteration time. This option is supported only for 1D and 2D Cartesian and Structured tables, as described in the *Table Format* property. This requires relatively smooth data to produce reliable results. For sudden jumps such as phase changes, use the default linear-piecewise method.

Using Table(T)

A (T) table has a column of temperature values for each point (row) in the table.

By default, a linear-piecewise interpolation method is used for interpolating between data points in the table. If the point lies outside the table range, the function value is computed at the nearest data point in the table. The properties for this function are set in the *Properties* window.

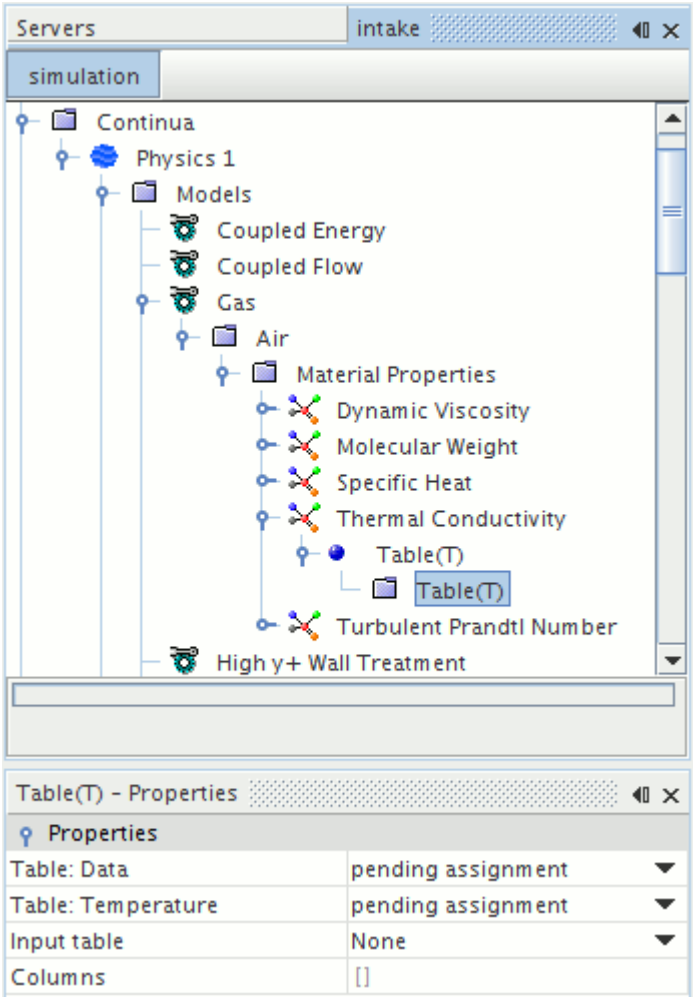


Table (T) Properties

Table: Data	Identifies the selected column in the table file as the Data (for example, Thermal Conductivity) column.
Table: Temperature	Identifies the selected column in the table file as the Temperature column.
Input Table	Selects a table from among the ones that appear as nodes in the **Tables** node of the **Tools** node.
Columns	Displays headings of columns in the table (read-only).

Table (T) Expert Properties

Table Format	Read-only. Reports the format of the table data. Before the data is read and processed, the state is **Unprocessed**. Afterward, the state is one of these three: • **Cartesian**: Data is stored on equally spaced grid points.

	• **Structured**: Data is stored on grid points that are regularly spaced but not equally spaced. • **Unstructured**: Data is stored on irregularly spaced grid points. See Cartesian and Structured Tables.
Verbose	Provides more output for this value while the simulation is running. This property can be useful for debugging problems when they occur.
Cubic Interpolation	When **On**, provides smoother interpolation between table values than the default linear-piecewise method, and reduces simulation iteration time. This option is supported only for 1D and 2D Cartesian and Structured tables, as described in the *Table Format* property. This requires relatively smooth data to produce reliable results. For sudden jumps such as phase changes, use the default linear-piecewise method.

Using Table (T,P)

A (T,P) table consists of three columns, one for the material property itself, and two additional columns that give corresponding values of temperature and absolute pressure. STAR-CCM+ interpolates the material property value from the table.

By default, a linear-piecewise interpolation method is used for interpolating between data points in the table. If the point lies outside the table range, the function value is computed at the nearest data point in the table. The properties for this function are set in the *Properties* window.

This method is available with density, thermal conductivity, dynamic viscosity, and electrical conductivity. An example application is a circuit breaker, where an electrical arc passes through a gas and causes a large increase in the local air temperature and pressure. These changes exert a strong influence on the local value of the electrical conductivity.

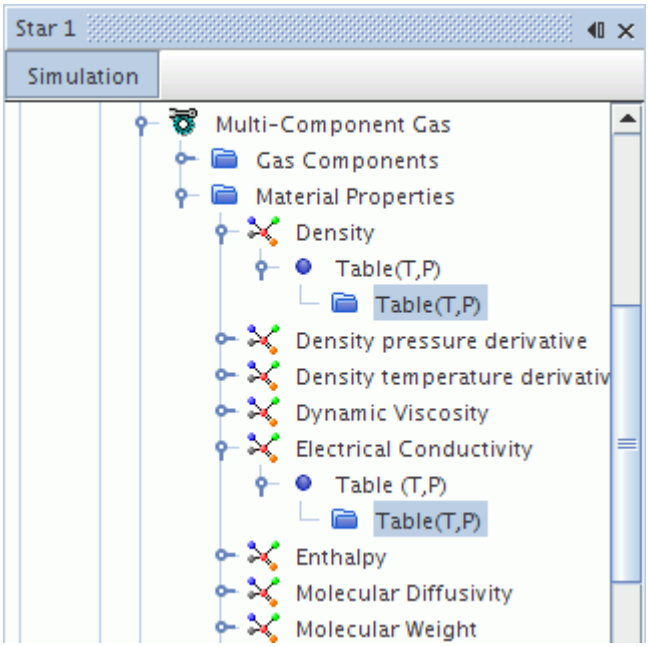


Table (T,P) Properties

Columns	Displays column headings in the table (read-only).
Table: Data	Identifies the selected column in the table file as the Data (for example, density or electrical conductivity) column.
Table: Temperature	Identifies the table column containing the temperature values.
Table: Absolute Pressure	Identifies the table column containing the absolute pressure values.
Input Table	Selects a table from among the ones that appear as nodes in the **Tables** node of the **Tools** node.

Table (T,P) Expert Properties

Table Format	Read-only. Reports the format of the table data. Before the data is read and processed, the state is **Unprocessed**. Afterward, the state is one of these three: • **Cartesian**: Data is stored on equally spaced grid points. • **Structured**: Data is stored on grid points that are regularly spaced but not equally spaced. • **Unstructured**: Data is stored on irregularly spaced grid points. See Cartesian and Structured Tables.
Verbose	Provides more output for this value while the simulation is running. This property can be useful for debugging problems when they occur.
Cubic Interpolation	When **On**, provides smoother interpolation between table values than the default linear-piecewise method, and reduces simulation iteration time. This option is supported only for 1D and 2D Cartesian and

	Structured tables, as described in the *Table Format* property. This requires relatively smooth data to produce reliable results. For sudden jumps such as phase changes, use the default linear-piecewise method.

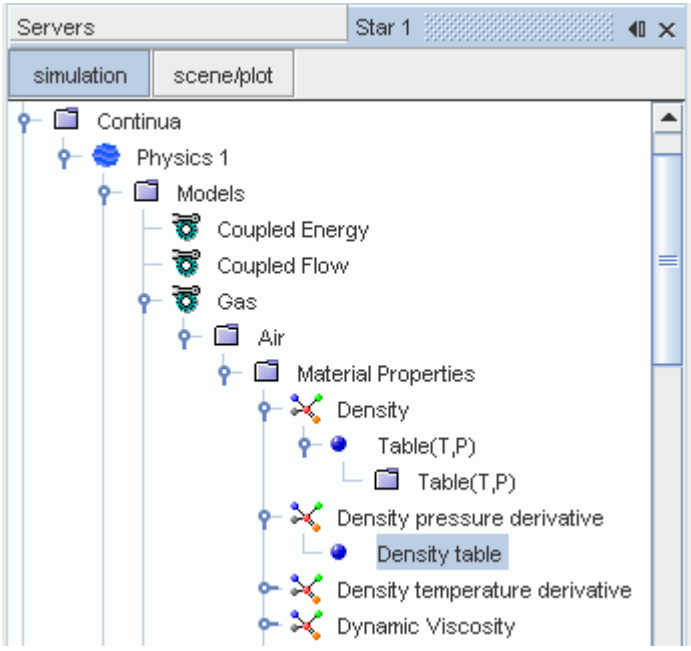
Using the Density Table Method for Density Derivatives

When density is defined by a table of data in temperature (or temperature and pressure for compressible flows), STAR-CCM+ can numerically differentiate the table of data for density.

When density is specified with the **Table(T)** or **Table(T,p)** method, STAR-CCM+ uses numerical differentiation to evaluate the following properties:

- The density pressure derivative, derived by differentiating the density table with respect to pressure;
- The density temperature derivative, derived by differentiating the density table with respect to temperature.

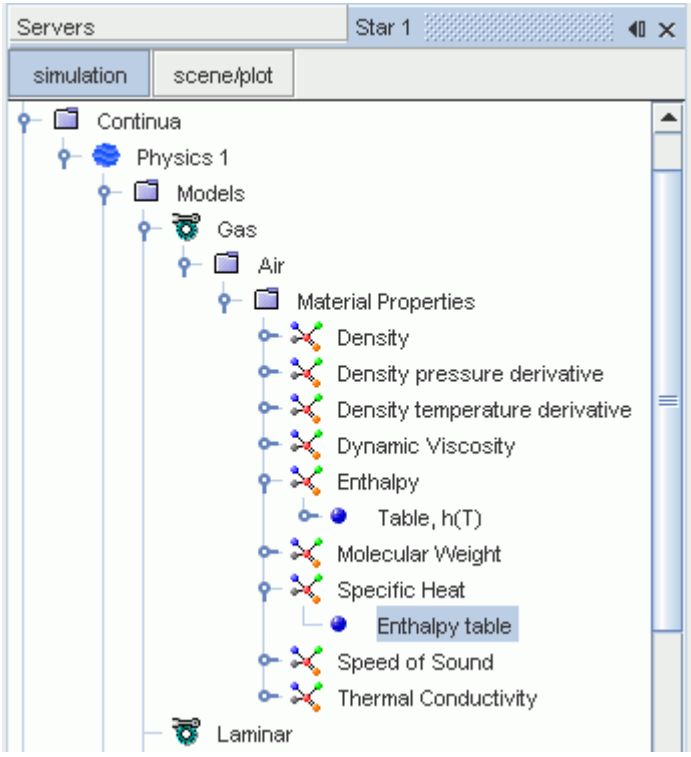
The User-Defined Equation of State model requires these properties. For an incompressible equation of state in the presence of an energy model, only the density temperature derivative is required. For a compressible equation of state, both the density pressure and density temperature derivatives are required.



Using the Enthalpy Table Method for Enthalpy Derivatives

This material property method is available for specifying specific heat when the enthalpy is defined by a table of values in temperature or in temperature and pressure. This method can also be used to specify the enthalpy pressure derivative when enthalpy is defined by a table of values in temperature and pressure.

When this option is selected, STAR-CCM+ calculates the specific heat by numerically differentiating an **Enthalpy table**, a specified **Table, h(T)**, or a specified **Table, h(T,p)** with respect to temperature at constant pressure. The table nodes have no properties.

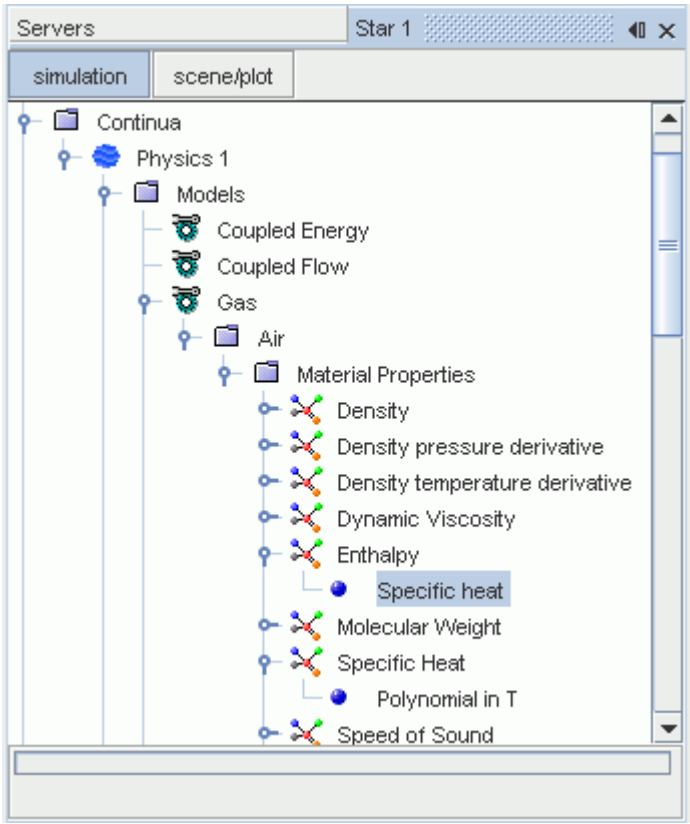


If the **Table h(T,p)** is selected, the **Enthalpy Pressure Derivative** node becomes available under **Material Properties**. If the **Enthalpy table** option is selected for this property, STAR-CCM+ calculates the enthalpy pressure derivative by numerically differentiating the enthalpy table with respect to pressure at constant temperature.

Using the Specific Heat Method for Enthalpy

This material property method is available for specifying enthalpy when the User-Defined Equation of State model is active.

When selected, STAR-CCM+ calculates the enthalpy by integrating specific heat. It is the default method that is used for calculating enthalpy in STAR-CCM+. The **Specific heat** node has no properties.



Using the Enthalpy Difference Method for Heat of Vaporization

This material property method is available for specifying latent heat of vaporization when certain droplet evaporation models are active.

These models are:

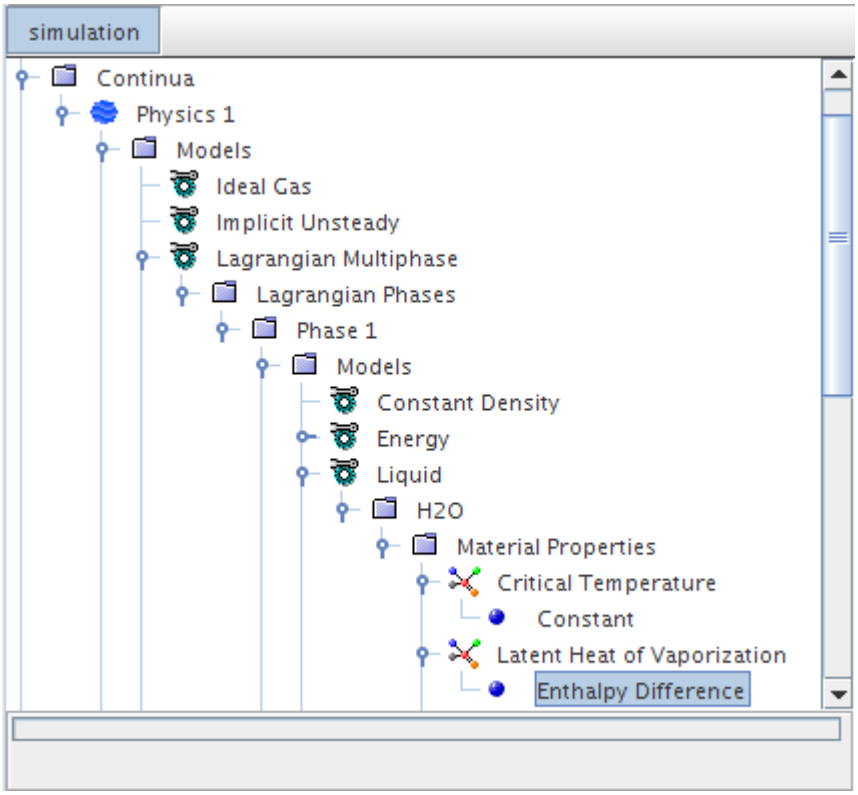
- Quasi-Steady Single-Component Droplet Evaporation
- User-Defined Single-Component Droplet Evaporation
- Multi-Component Droplet Evaporation

When this option is selected, STAR-CCM+ calculates the latent heat of vaporization directly as a difference in the enthalpy of the gas and liquid states:

$$L\left(T_{\mathrm{drop}}\right) = h_{\mathrm{vap}}\left(T_{\mathrm{drop}}\right) - h_{\mathrm{liq}}\left(T_{\mathrm{drop}}\right) \tag{168}$$

where:

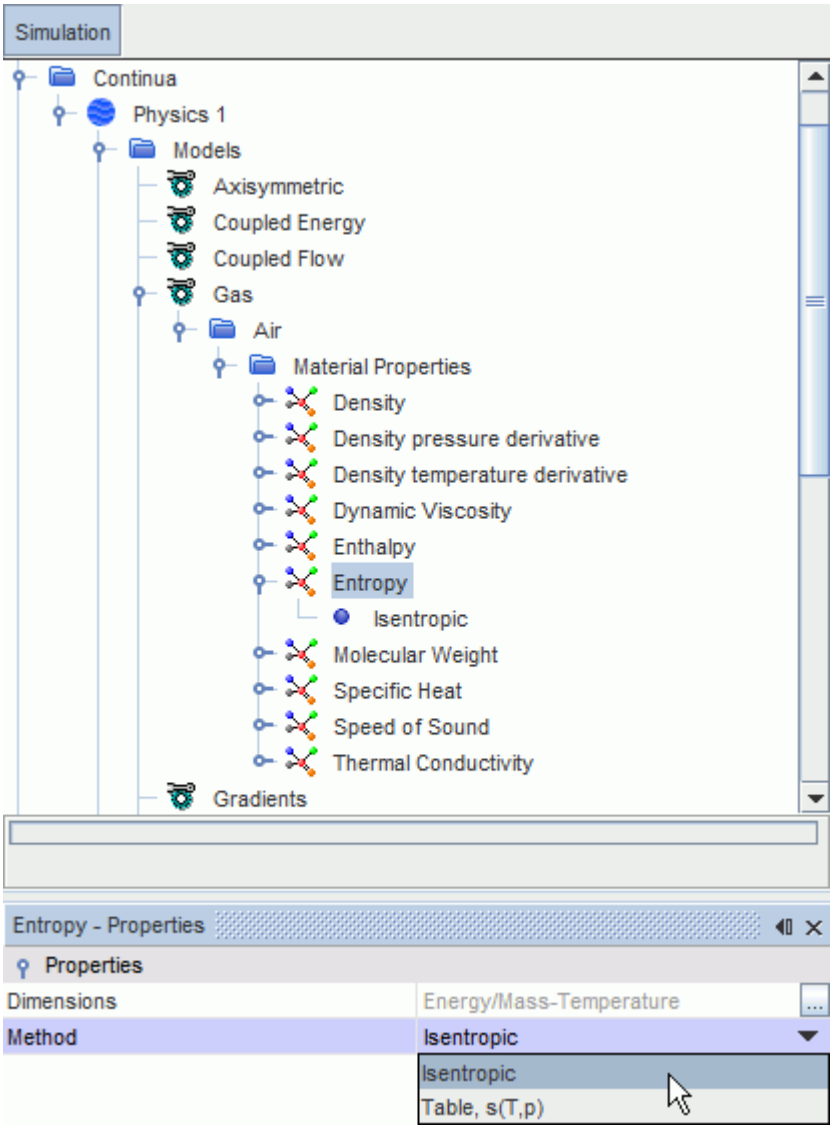
- $L\left(T_{\mathrm{drop}}\right)$ is the latent heat of the droplet material at the droplet temperature.
- $h_{\mathrm{vap}}\left(T_{\mathrm{drop}}\right)$ is the static enthalpy of the vapor component corresponding to the droplet material, evaluated at the droplet temperature.
- $h_{\mathrm{liq}}\left(T_{\mathrm{drop}}\right)$ is the static enthalpy of the droplet material, evaluated at the droplet temperature.
-



Using Entropy

When you use the User Defined Equation of State model with compressible flow, you have the option to specify the entropy of a material.

You can specify the entropy of a material as isentropic or as a table of values for particular temperatures and pressures.



Entropy Methods

Method	Description
Isentropic	Treats the entropy as a constant value, independent of temperature and pressure. For more information, see the entropy formulation. This method is the default for existing simulations (those created with STAR-CCM+ 10.04 and earlier).
Table, s(T,P)	Specifies the entropy using a table of values. Entropy is tabulated as a function of temperature and pressure. This method is the default for new simulations.

Method	Description
	The tables that you use to specify the entropy, and the other material properties (such as density and enthalpy), must cover the full range of temperatures and pressures that are encountered in the simulation. The range must cover the total pressure and total temperature values, which are higher than the corresponding static quantities. If the table does not cover the full range, the total quantities may not converge.

Using Table(B,H) Method for Permeability

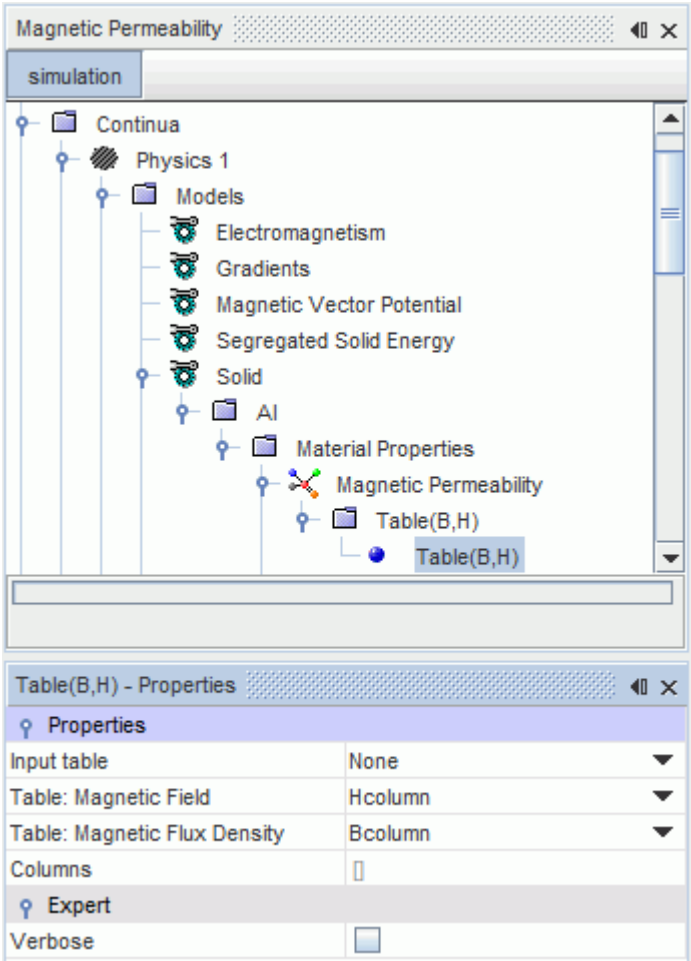
This material property provides a stable iterative scheme for setting magnetic permeability for both convex and concave B-H curves.

By default, a linear-piecewise interpolation method is used for interpolating between data points in the table. If the point lies outside the table range, the function value is computed at the nearest data point in the table. The properties for this function are set in the *Properties* window.

This method is a preferred method for setting permeability when using a table over the tabular data method.

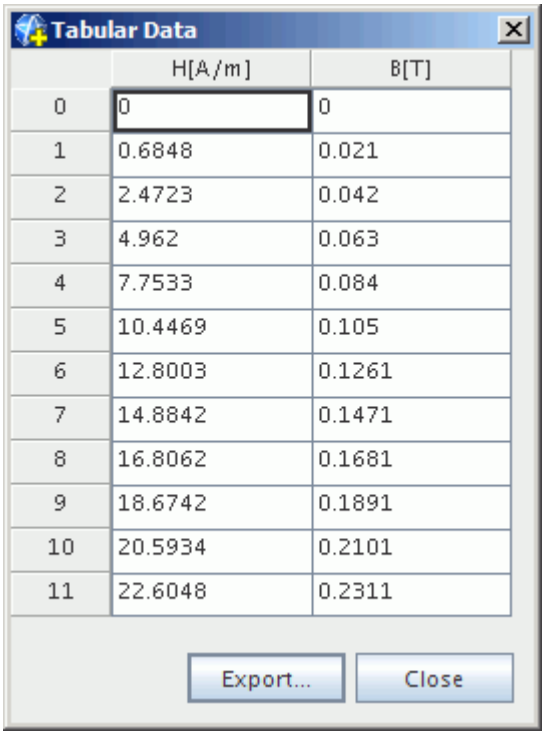
If you want to select the B-H-mu table that is in the material database, you use this method. The tabular data method does not let you select the permeability table from the material database.

To use Table(B,H), input the table and the columns corresponding to the magnetic field H and the magnetic flux density B.



The first entry in the table must be (0,0). When $\mathbf{B}$ becomes greater than the maximum flux density that is provided in the table, the B-H curve is extended at a slope of μ_0 from the last point.

The provided table for this method can look as follows:



	H[A/m]	B[T]
0	0	0
1	0.6848	0.021
2	2.4723	0.042
3	4.962	0.063
4	7.7533	0.084
5	10.4469	0.105
6	12.8003	0.1261
7	14.8842	0.1471
8	16.8062	0.1681
9	18.6742	0.1891
10	20.5934	0.2101
11	22.6048	0.2311

Using Tabular Data

The tabular data method is used to provide values of a material property. The multi-dimensional bilinear interpolation method is used for interpolating between data points in the table.

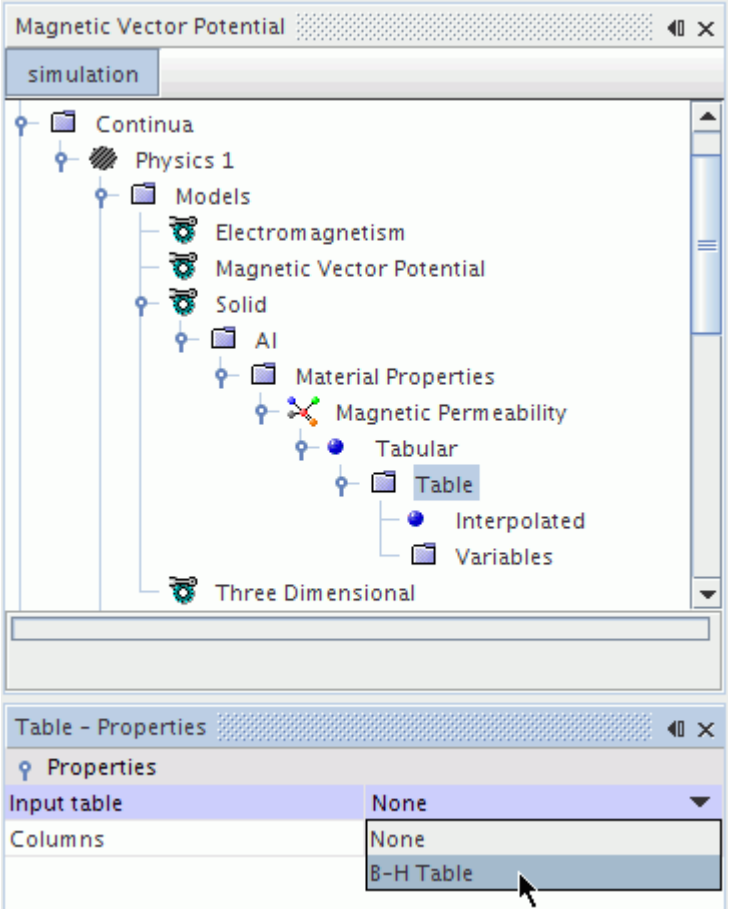
By default, a linear-piecewise interpolation method is used for interpolating between data points in the table. If the point lies outside the table range, the function value is computed at the nearest data point in the table. The properties for this function are set in the *Properties* window.

As an example, the tabular data method is described here for the magnetic permeability material property of the Magnetic Vector Potential model.

If you want to select the B-H-mu table that is in the material database, you use the Table(B,H) method. The tabular data method does not let you select the permeability table from the material database.

The tabular data method works as follows:

- Select the **Tabular** > **Table** node and, in the *Properties* window, select the table containing the material property data.

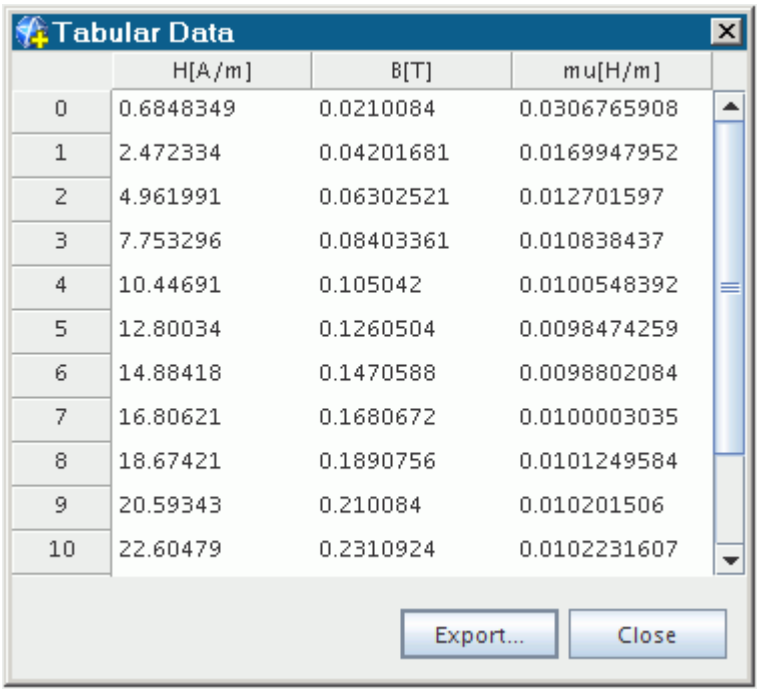


- Select the **Table** > **Interpolated** node and, in the *Properties* window, set the **Column title** that matches the material property to be returned by the method.
- Select the input quantity against which the material property is interpolated:
  - Right-click the **Table** > **Variables** node and choose **New**.

    A new **Variable** child node is added.
  - Select the newly added child node. In the *Properties* window, set the **Scalar Function** property to the desired quantity. Set the **Column title** property to the table column that contains values of this quantity.

Setting Magnetic Permeability Property

Following the above procedure, first set the **Input table** property to the B-H table. The B-H table usually includes three columns: one for the magnetic flux density $\mathbf{B}$, one for the magnetic field $\mathbf{H}$ and one for the calculated permeability μ that is determined through the relation $\mu = \mathbf{B}/\mathbf{H}$. An example table is shown below. Only two columns are needed for the tabular data method. For this case, either $\mathbf{H}$-μ or $\mathbf{B}$-μ is used.



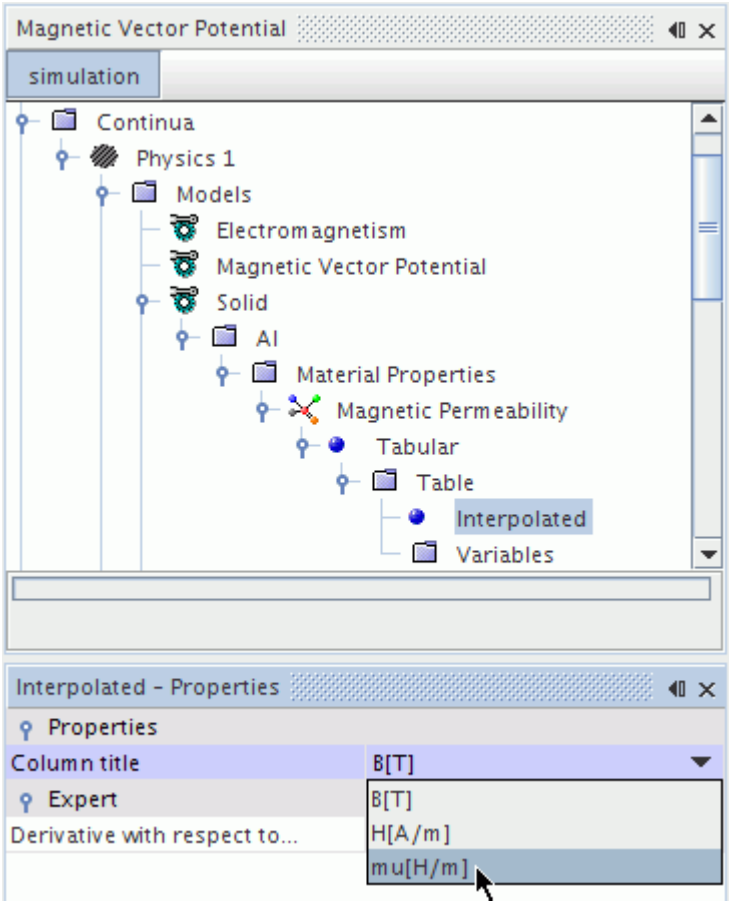
Tabular Data

	H[A/m]	B[T]	mu[H/m]
0	0.6848349	0.0210084	0.0306765908
1	2.472334	0.04201681	0.0169947952
2	4.961991	0.06302521	0.012701597
3	7.753296	0.08403361	0.010838437
4	10.44691	0.105042	0.0100548392
5	12.80034	0.1260504	0.0098474259
6	14.88418	0.1470588	0.0098802084
7	16.80621	0.1680672	0.0100003035
8	18.67421	0.1890756	0.0101249584
9	20.59343	0.210084	0.010201506
10	22.60479	0.2310924	0.0102231607

Export... Close

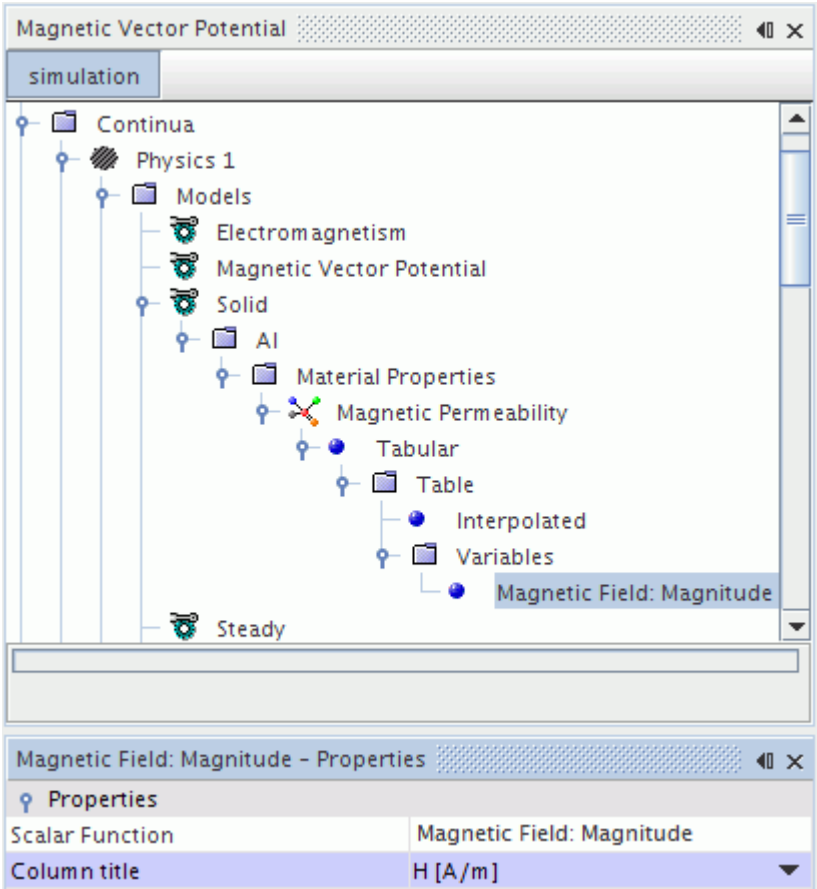
Columns that are extracted from this table appear in the **Columns** property of the **Table** node.

Having selected the table, choose the mu[H/m] column for the **Column title** property of the **Interpolated** child node. This column matches the Magnetic Permeability material property for which the Tabular method has been selected.



Finally, choose either the **H** or **B** scalar field function against which permeability is interpolated. Determining which of these variables to use as the interpolation variable has a direct effect on the convergence of the magnetic vector potential solution (as discussed in the troubleshooting section of electromagnetism modeling).

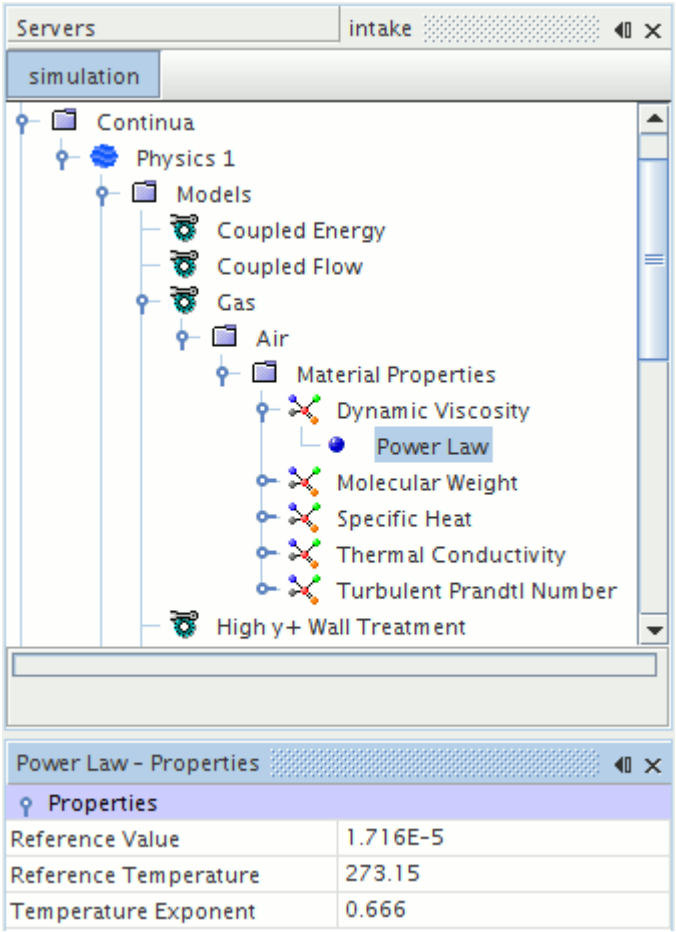
Here, create a variable and set its **Scalar Function** property to Magnetic Field: Magnitude, and its **Column title** property to H[A/m].



Using the Power Law

The Power law function is available for various material properties.

The coefficient properties for this function are set in the *Properties* window.



The Power law for viscosity is:

$$\frac{\mu}{\mu_0} = \left(\frac{T}{T_0}\right)^n \tag{169}$$

where T_0 and μ_0 are the reference temperature and viscosity, respectively and n is the Power law exponent.

The Power law for thermal conductivity is:

$$\frac{k}{k_0} = \left(\frac{T}{T_0}\right)^n \tag{170}$$

where T_0 and k_0 are the reference temperature and conductivity, respectively and n is the Power law exponent.

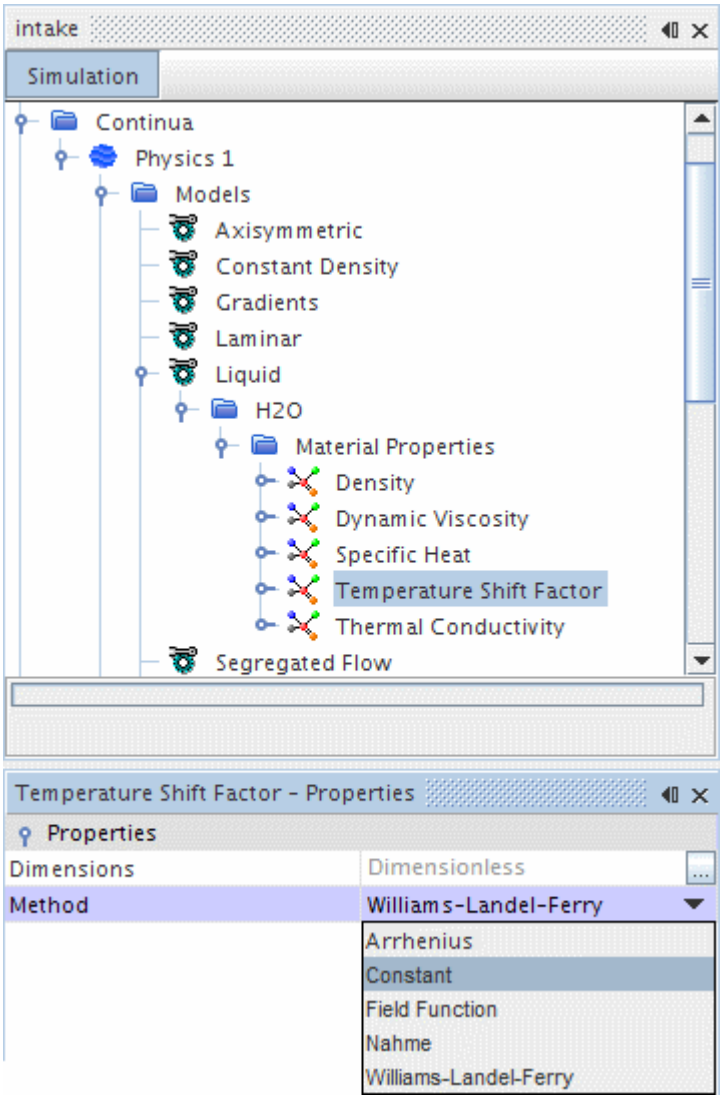
Power Law Properties

Reference Temperature	The reference temperature in Eqn. (169) or Eqn. (170) (depending on the material property).	
Reference Value	The reference viscosity or conductivity (depending on the material property) in Eqn. (169) or Eqn. (170).	

Temperature Exponent	The temperature exponent in Eqn. (169) or Eqn. (170) (depending on the material property).

Using the Temperature Shift Factor

When temperature effects are present, the non-Newtonian liquid models use a shift factor a_T to adjust viscosity with temperature.



When the simulation is isothermal, $a_T = 1$. For polymers:

$$a_T = \frac{\eta_T}{\eta_{T0}}$$

where η is the viscosity at temperature T and T_0.

The shift factor can be calculated by the Arrhenius model, the Nahme model, or the Williams-Landel-Ferry (WLF) model.

Temperature Shift Factor Methods

Arrhenius Method

Exposes the following terms from Eqn. (1510):

- E_a: the activation energy.
- T_0: a reference temperature in K.

Use the Arrhenius model when the temperature is at least 100°C above the glass transition temperature of the polymer.

Nahme Method

Exposes the following terms from Eqn. (1511):

- E_a: the activation energy.
- T_0: a reference temperature in K.

Use the Nahme model when the temperature is close to the reference temperature T_0.

Williams-Landel-Ferry (WLF) Method

Exposes the following terms from Eqn. (1512):

- C_1 and C_2: positive empirical constants that depend on the material and the reference temperature. See [20] for tables of values for C_1, C_2
- T_0: a reference temperature in K.

Use the WLF model when the temperature is close to the glass transition temperature of the polymer.

Using the Non-Newtonian Generalized Models

To model a non-Newtonian liquid, STAR-CCM+ supports three methods.

The methods are:

- Power Law
- Cross Fluid
- Carreau-Yasuda Fluid

Using the Non-Newtonian Generalized Models in Turbulent Flows

The Non-Newtonian Generalized models are now available for turbulent flows; previously they were available for laminar flows only.

Note that this approach is not strictly valid, as the turbulence models have been designed for Newtonian fluids; the production at the wall is known to be different for non-Newtonian fluids. Also the turbulence models do not account for any viscoelasticity of the flow. However, in many cases, the results that you obtain from

using the (incorrect) turbulence model can be better than the results that you would obtain by assuming a laminar flow.

If you use the Non-Newtonian Generalized models in a turbulent flow, you do so at your own risk.

Using the Non-Newtonian Models in Multiphase Flows

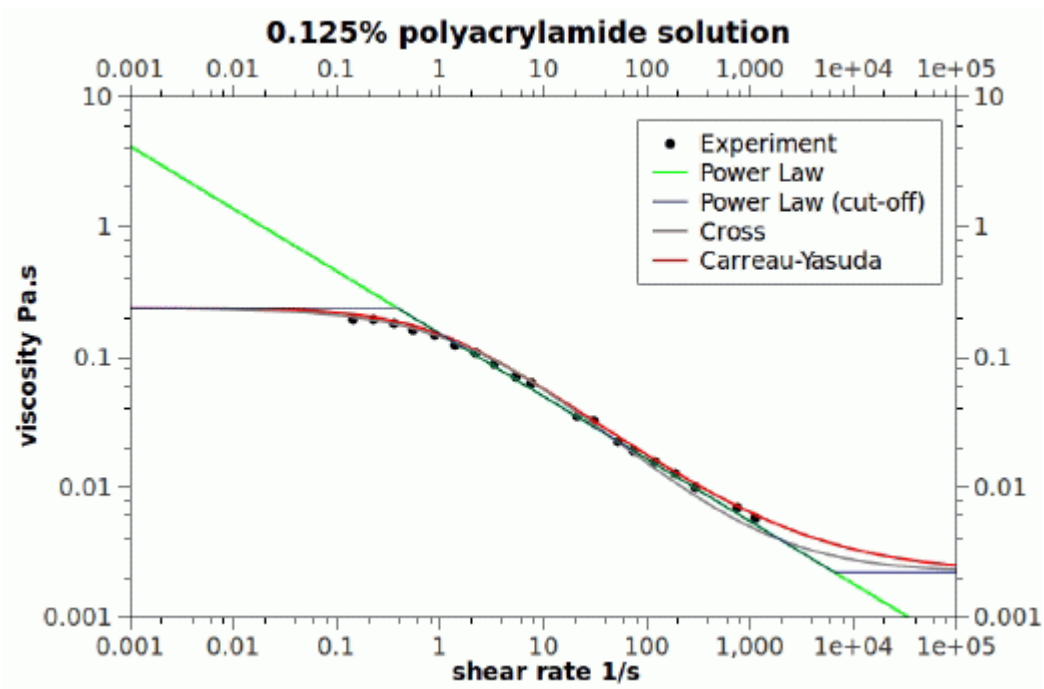
The Non-Newtonian Generalized models are available for multiphase flows. However, the following restrictions apply:

- The Non-Newtonian Generalized Cross Fluid model is not compatible with the Eulerian Multiphase model.
- The dispersed phase cannot be specified as a Non-Newtonian fluid. Consequently, you cannot define any interphase interactions between Non-Newtonian phases (which have to be continuous).

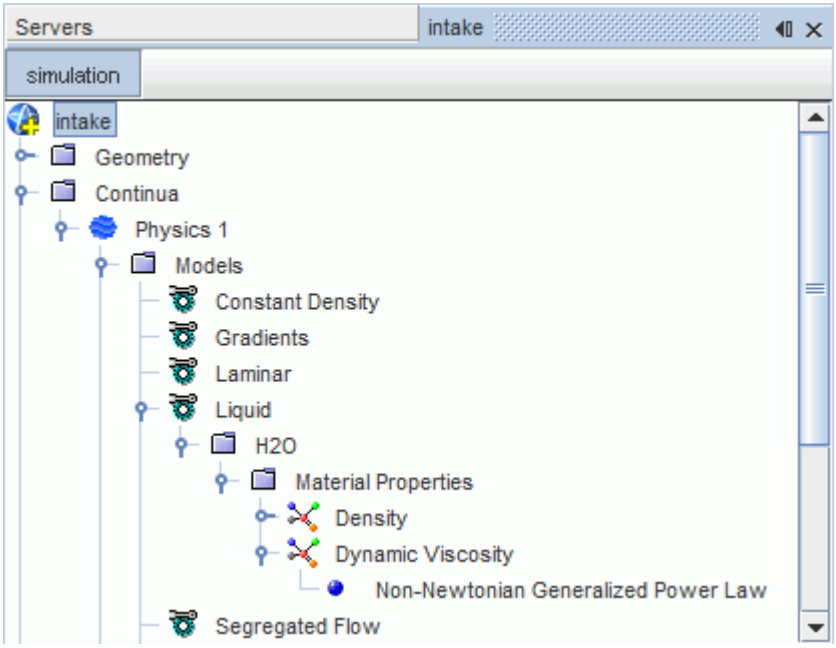
Selecting a Non-Newtonian Model

The graph below shows a comparison of the models from [19]. The three-parameter Cross model and the four-parameter Carreau-Yasuda model both display good agreement with the experimental data for this shear-thinning material.

However, these models require you to provide the values of zero-shear and infinite shear at the Newtonian plateaus. In contrast, the two-parameter Power Law requires fewer inputs and has good agreement in the main-shear thinning (or shear thickening) zone but loses accuracy as the fluid approaches a Newtonian plateau.



These methods are available for the material property Dynamic Viscosity for both laminar and turbulent liquid flow. They can be used with single phase and multiphase flows. The coefficient properties for this function are set in the *Properties* window.



Non-Newtonian Generalized Power Law Properties

Non-Newtonian Generalized Power Law

This method comprises two rheology models into one:

- Hershel-Bulkley
- Power Law/Ostwald de Waele

Exposes the following terms from Eqn. (1497):

- *Consistency Factor* k
- *Power Law Exponent* n
- *Yield Stress Threshold* τ_0
- *Yielding Viscosity* μ_0
- *Minimum Viscosity Limit* and *Maximum Viscosity Limit*: minimum and maximum values allowed for μ

To obtain the Power Law/Ostwald de Waele viscosity function, set the yield stress $\tau_0 = 0$.

Non-Newtonian Generalized Cross Fluid Properties

Non-Newtonian Generalized Cross Fluid

Exposes the following terms from Eqn. (1494):

- *Cross Rate Constant* m
- *Zero Shear Viscosity* μ_0
- *Infinite Shear Viscosity* μ_∞
- *Critical Shear Rate* $\dot{\gamma}_c$

Non-Newtonian Generalized Carreau-Yasuda Fluid Properties

Non-Newtonian Generalized Carreau-Yasuda Fluid

Exposes the following terms from Eqn. (1497):

- *Power Constant* n
- *a Parameter* a
- *Zero Shear Viscosity* μ_0
- *Infinite Shear Viscosity* μ_∞
- *Relaxation Time* λ

Also provides the *Viscosity Under-Relaxation Factor*, which you can reduce to enhance convergence.

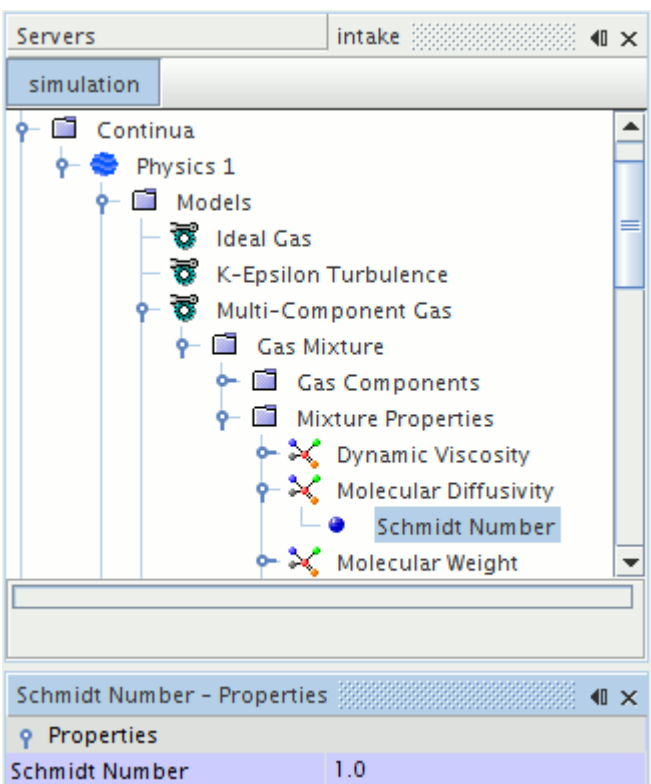
Using the Schmidt Number

The Schmidt number function is available for specifying molecular diffusivity of components in a mixture.

The Schmidt number is a non-dimensional parameter that is defined as:

$$S_c = \frac{\nu}{D_{im}} \tag{171}$$

where ν is the kinematic viscosity and D_{im} is the molecular diffusivity of component-i into the mixture.



The properties for this method are set in the *Properties* window.

Schmidt Number Properties

Schmidt Number	The value of Schmidt number that is used for specifying molecular diffusivity.

Using a Volume-Weighted Mixture

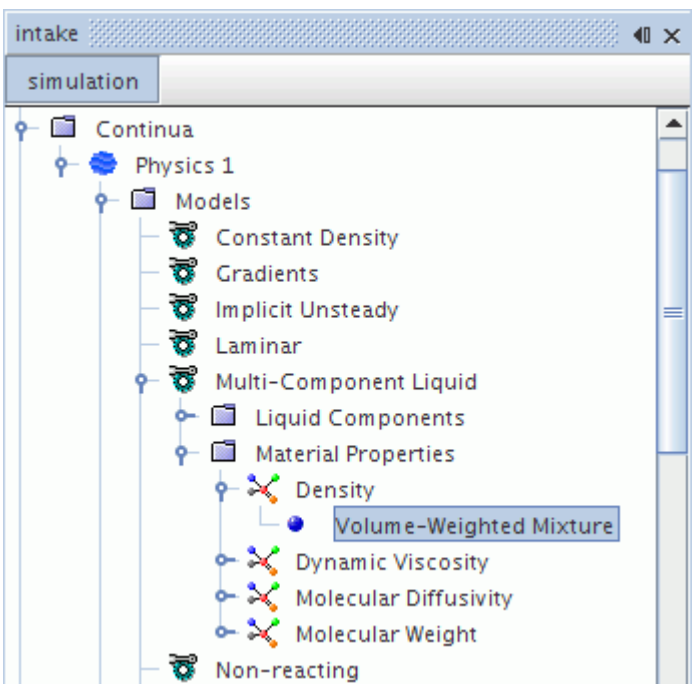
The volume-weighted mixture function is for non-ideal gas mixtures.

Using this function, a given mixture property ϕ_{mix} is defined as:

$$\phi_{mix} = \frac{1}{\sum_1^N \left(\frac{y_i}{\phi_i}\right)} \tag{172}$$

where y_i and ϕ_i are the mass fraction and individual material property of component-i in the mixture of N components.

This method is not available for conductivity or viscosity in single-phase multi-component mixtures. It is, however, available for these properties in a multiphase mixture.

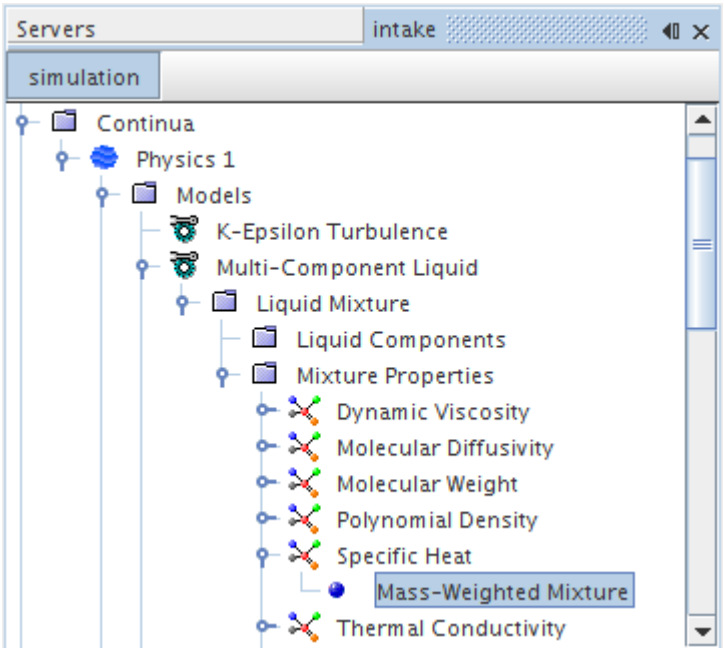


Volume-Weighted Mixture Properties

None

Using the Mass-Weighted Mixture

The mass-weighted mixture method calculates a given mixture property by mass-weighting the component property values.



For example, mixture property ϕ_{mix} is:

$$\phi_{mix} = \sum_{i=1}^{N} y_i \phi_i \tag{173}$$

where y_i and ϕ_i are the mass fraction and property values of mixture component-i and N is the total number of components in the mixture.

Mass-Weighted Mixture Properties

None

Using the Lewis Number

The Lewis number L_e is defined as the ratio of thermal diffusion to mass diffusion D_m.

The Lewis number is expressed as follows:

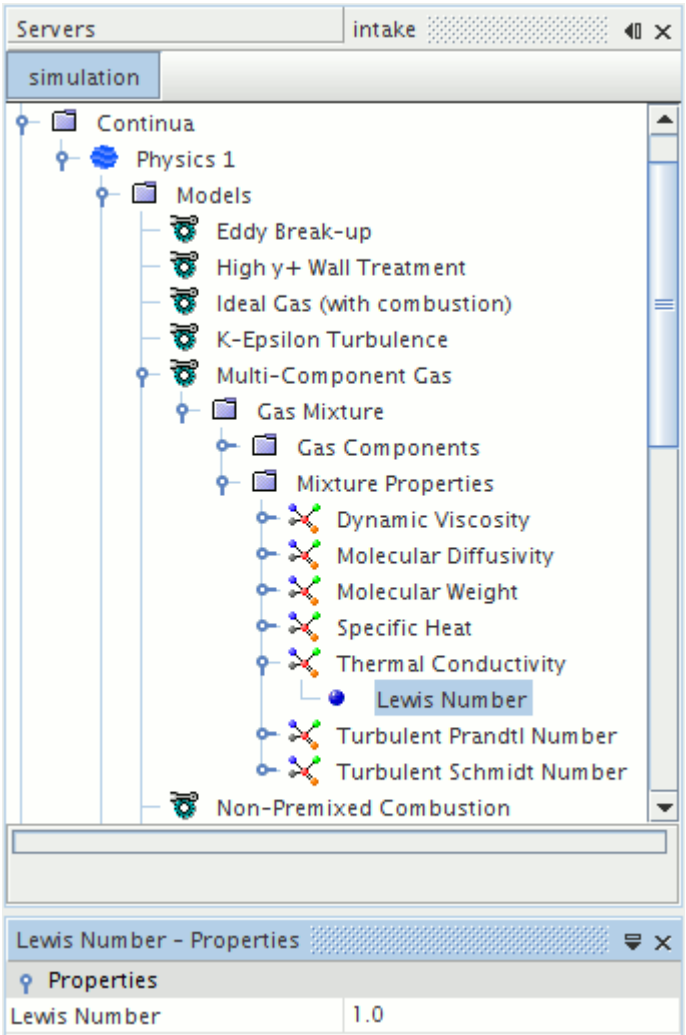
$$L_e \equiv \frac{k/\rho C_p}{D_m} \tag{174}$$

where ρ, C_p, and k are the density, specific heat, and thermal conductivity of the mixture, respectively.

Given values for L_e, D_m, ρ, and C_p, thermal conductivity k can be calculated as follows:

$$k = L_e D_m \rho C_p \tag{175}$$

The properties for this method are set in the *Properties* window.



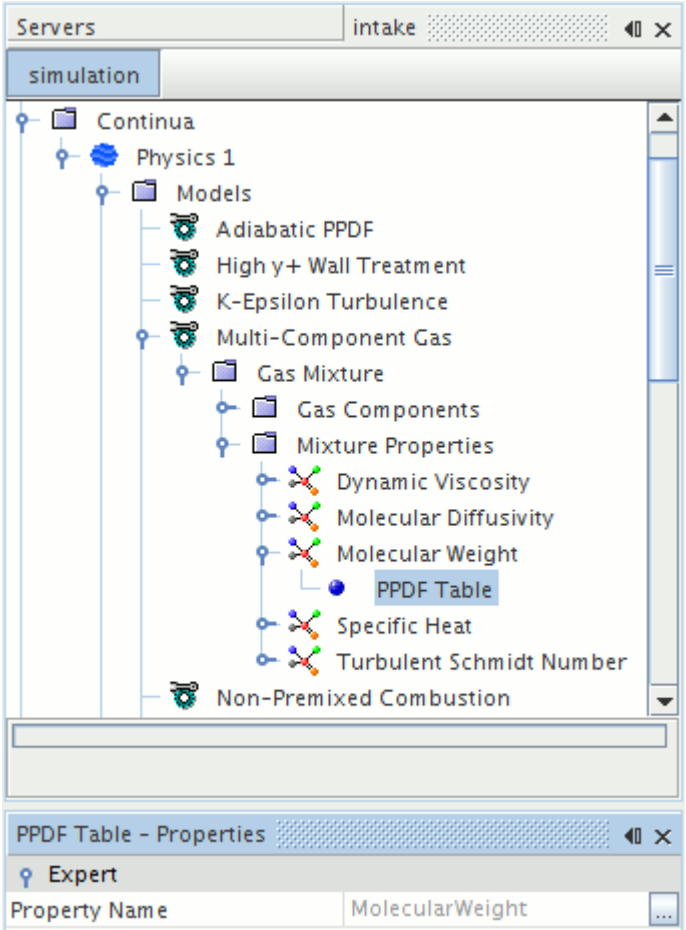
Lewis Number Properties

Lewis Number	The value of Lewis number that is used for specifying the ratio of thermal diffusion to mass diffusion.

Using the PPDF Table

The PPDF Table method interpolates property values from the PPDF table. The Molecular Weight and Specific Heat material properties of the mixture use this method.

This method is the only one available for these properties when the Presumed Probability Density Function (PPDF) model is active.



PPDF Table Properties

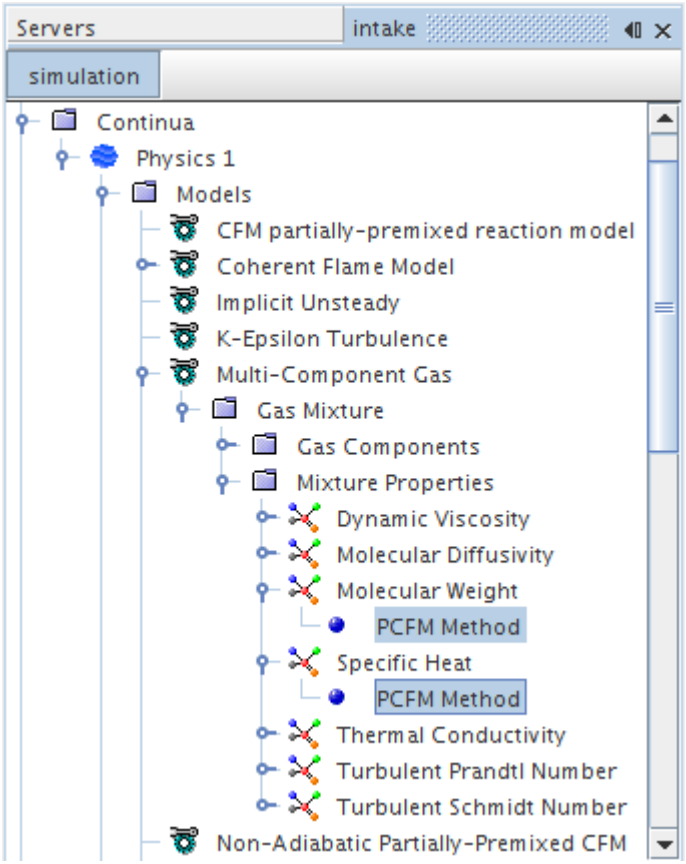
None

Expert Properties

Property Name	Identifies the material property that is being interpolated from the table (read-only).

Using the PCFM Method

The PCFM method becomes active when the Partially-Premixed Coherent Flame Model (PCFM) is used. The Molecular Weight and Specific Heat material properties of the mixture use this method.



This method is the only one available for these properties when the PCFM model is active.

PCFM Method Properties

None

Expert Properties

Property Name	Identifies the material property (read-only).

Using the Mixture Method for Critical Temperature and Pressure

The Mixture method calculates the critical temperature and pressure of a mixture.

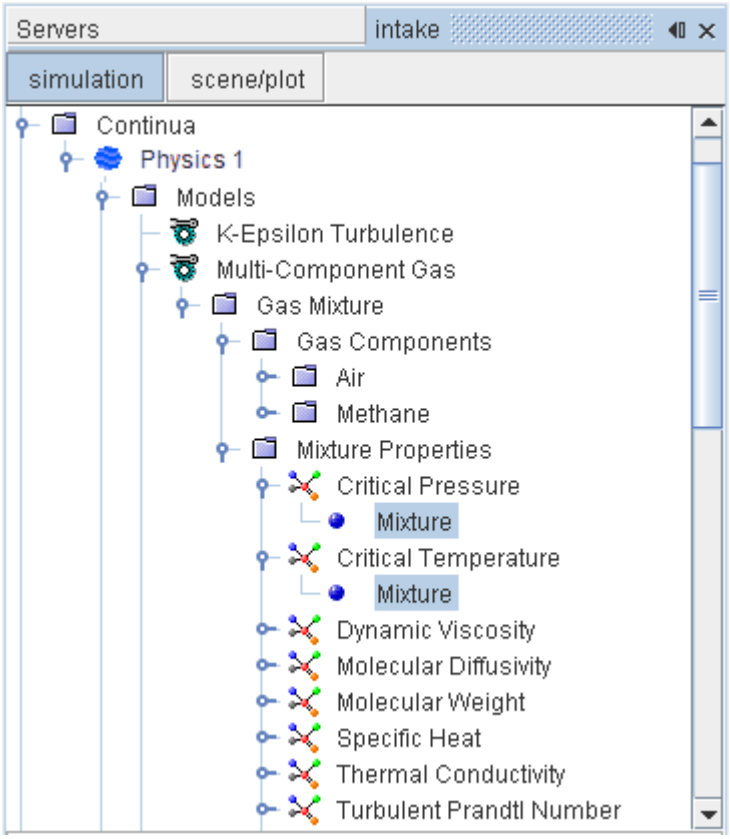
The Mixture method calculates the critical temperature $T_{c,m}$ of a mixture as:

$$T_{c,m} = \frac{\sum_i T_{c,i} Y_i / W_i}{\sum_i Y_i / W_i} \tag{176}$$

where W_i is the molecular weight, Y_i the mass fraction and $T_{c,i}$ the critical temperature of species i. The summation is over all species in the mixture. The critical pressure $P_{c,m}$ of a mixture is similarly calculated as:

$$P_{c,m} = \frac{\sum_i P_{c,i} Y_i / W_i}{\sum_i Y_i / W_i} \tag{177}$$

where $P_{c,i}$ is the critical pressure of component i.



Properties

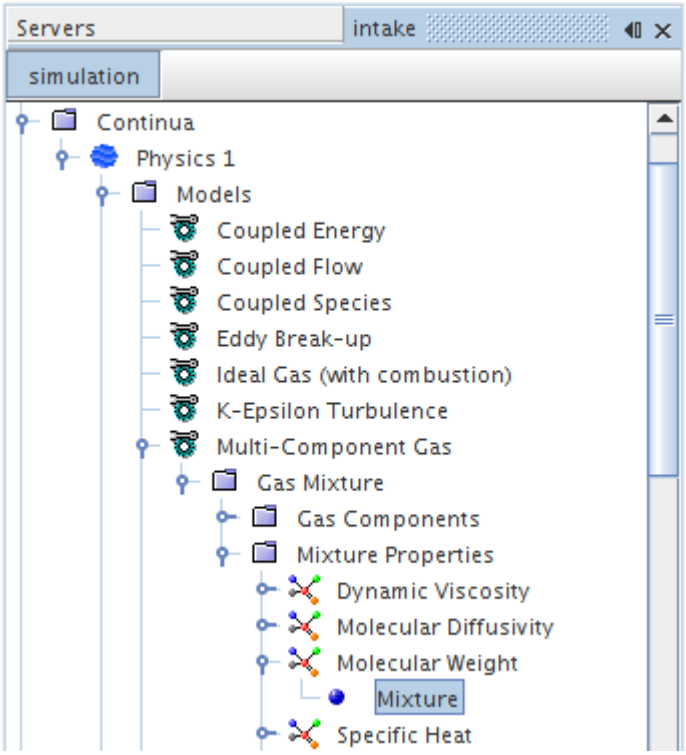
None.

Using the Mixture Method for Molecular Weight

When the Mixture method is used for molecular weight, the mixing law is:

$$W_m = \frac{1}{\sum_i Y_i / W_i} \tag{178}$$

where W_i is the molecular weight of species i, Y_i is the mass fraction of species i, and the summation is over all species in the mixture.



Properties

None.

Using the Mixture Method for Surface Tension

When the Mixture method is used for surface tension, the mixture surface tension is:

$$\sigma_m{}^r = \sum_i X_i \sigma_i{}^r \tag{179}$$

where σ_i is the surface tension of component i, X_i is its mole fraction, and r is the exponent property of the mixing law.

Mixture Method Properties

Exponent	The exponent property r of the mixing law that is used to determine the surface tension of a mixture (Eqn. (179)).

Using the Mathur-Saxena Averaging Method for Dynamic Viscosity

This method becomes available when the multi-component gas model is used. Selecting this method provides the choice of the Chapman-Enskog method for Dynamic Viscosity on the Mixture Component level.

The following properties show up on the mixture component level:

- Dipole momentum – a measure of polarity of a covalent bond in the molecule (always given in Debye)
- Lennard-Jones characteristic length – the collision diameter (always given in Angstroms)
- Lennard-Jones energy – the potential energy of attraction (given in J/kmol)

These properties have only one method available for their calculation: *Constant*. These values are used for calculating the dynamic viscosity of the individual component for the Chapman-Enskog method.

When the *Method* property of the **Dynamic Viscosity** node is set to **Mathur-Saxena Averaging**, the **Multi-Component Gas** > **Material Properties** > **Dynamic Viscosity** > **Mathur-Saxena Averaging** node appears.

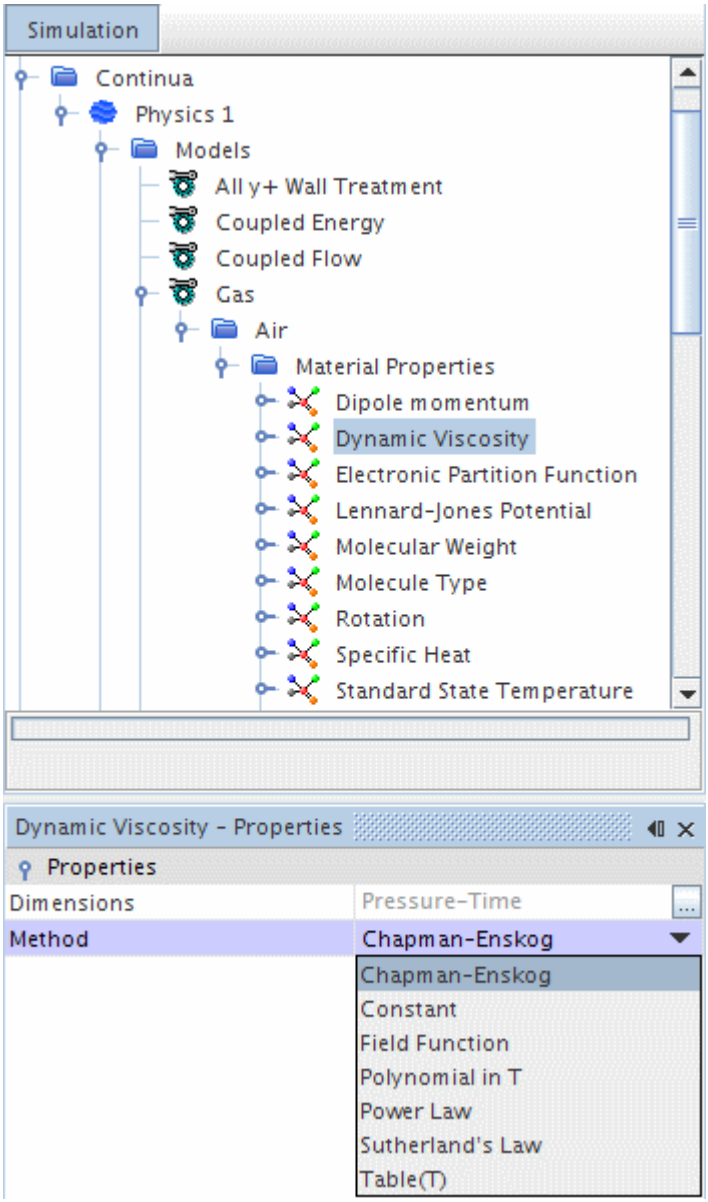
The Mathur-Saxena Averaging methods for Dynamic Viscosity and Thermal Conductivity are specifications of the Multi-Component Mathur-Saxena Averaging Property method <Φ> where Φ is a material property. This method uses the following formula to compute the property on the mixture level, using values that are given for the individual mixture components (X_i is the mole fraction of component i):

$$\Phi = \frac{1}{2}\left(\sum_1^n X_i \Phi_i + \left(\sum_1^n \frac{X_i}{\Phi_i}\right)^{-1}\right) \tag{180}$$

When the option is set, each component within the **Gas Components** manager has a **Dynamic Viscosity** node that contains the **Chapman-Enskog** node. See Using the Chapman-Enskog Method for Dynamic Viscosity.

Using the Chapman-Enskog Method for Dynamic Viscosity

This method becomes available for single- and multi-component gases once an energy model is included. This method is also activated when the Mathur-Saxena Averaging method is selected for dynamic viscosity.



The dynamic viscosity of a simple gas, or of an individual component of a multi-component gas, is calculated with the Chapman-Enskog equation as follows:

$$\mu_i = 2.6693 \times 10^{-6} \frac{\sqrt{M_i T}}{\sigma_i^2 \Omega(T^*)} \tag{181}$$

where:

- $T^* = kT/\varepsilon_i$ is reduced temperature (dimensionless)
- k is the Boltzmann constant
- ε_i is the potential energy of attraction of component i; values given in the standard material database, `props.mdb`, provided as a part of STAR-CCM+
- $\Omega(T^*)$ is the collision integral

- μ_i is the viscosity of component i
- M_i is the molecular weight of component i
- T is temperature (in K)
- σ_i is the collision diameter of component i; values given in the standard material database, `props.mdb`, provided as a part of STAR-CCM+

For a simple gas, there is only one value of i.

Using the Refutas Method for Dynamic Viscosity

This method becomes available when the Multi-Component Liquid model is included and density is calculated using the **Volume Weighted Mixture** method.

The kinematic viscosity ν_i of a multi-component liquid is calculated with the Refutas equation [reflink] as follows:

$$\nu = \exp\left(\exp\left(\frac{\sum_{i=1}^{n} y_i b_i - 10.975}{14.534}\right)\right) - 0.8 \tag{182}$$

where y_i is the mass fraction of the *i*th component of the mixture and b_i is the viscous blending number of the *i*th component:

$$b_i = 14.534\ \ln\left(\ln\left(\nu_i + 0.8\right)\right) + 10.975$$

and ν_i is the kinematic viscosity of the *i*th component in units of centiStokes.

Using the Mathur-Saxena Averaging Method for Thermal Conductivity

This method becomes available when the multi-component gas model is used. Selecting this method activates the choice of the Kinetic Theory method for Thermal Conductivity on the Mixture Component level.

The following properties show up on the mixture component level:

- Dipole momentum – a measure of polarity of a covalent bond in the molecule (always given in Debye)
- Lennard-Jones characteristic length – the collision diameter (always given in Angstroms)
- Lennard-Jones energy – the potential energy of attraction (given in J/kmol)
- Rotation – rotational relaxation collision number (non-dimensional)
- Molecule Type – describes molecular structure; can be one of the following:
  - 0: atom
  - 1: linear molecule
  - 2: nonlinear molecule

All these properties have only one method available for their calculation: Constant. These values are used for calculating the thermal conductivity of the individual component for the Kinetic Theory method.

When the *Method* property of the **Thermal Conductivity** node is set to **Mathur-Saxena Averaging**, the manager for a **Multi-Component Gas** > **Material Properties** > **Thermal Conductivity** > **Mathur-Saxena Averaging** node appears.

The Mathur-Saxena Averaging methods for Dynamic Viscosity and Thermal Conductivity are specifications of the Multi-Component Mathur-Saxena Averaging Property method $\langle \Phi \rangle$; see Eqn. (180).

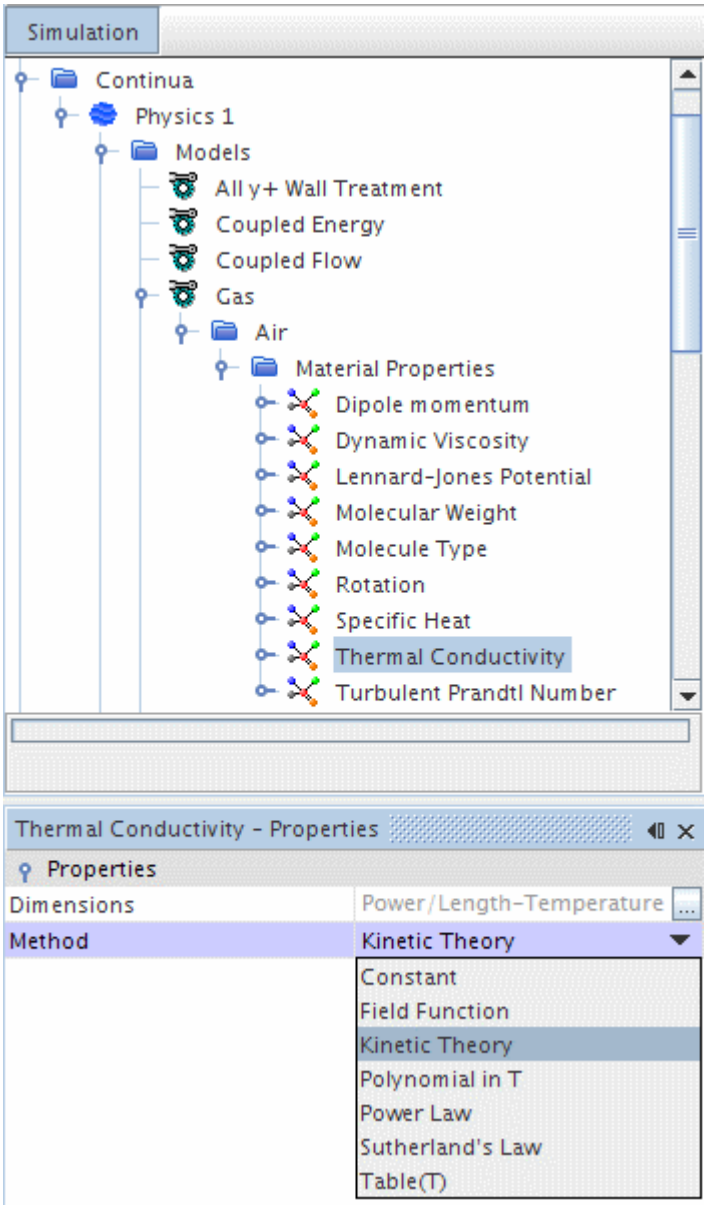
When the option is set, each component within the **Gas Components** manager has a **Thermal Conductivity** node that contains the **Kinetic Theory** node. See Using the Kinetic Theory Method for Thermal Conductivity.

Using the Kinetic Theory Method for Thermal Conductivity

This method is available for both single- and multi-component gases.

For single-component gases, the **Kinetic Theory** method is available under **Gas** > **[gas]** > **Material Properties** > **Material Properties**.

For multi-component gases, the **Kinetic Theory** method becomes available under **Material Properties** > **Thermal Conductivity** for each gas when you select **Mathur-Saxena Averaging** under **Multi-Component Gas** > **Material Properties** > **Thermal Conductivity**.



The thermal conductivity of a simple gas, or of an individual component of a multi-component gas, is calculated as follows:

$$\lambda_i = \frac{\mu_i}{M_i}\left(f_{trans}C_{v,trans} + f_{rot}C_{v,rot} + f_{vib}C_{v,vib}\right) \tag{183}$$

where:

$$f_{trans} = \frac{5}{2}\left[1 - \frac{2}{\pi}\frac{C_{v,rot}}{C_{v,trans}}\frac{A}{B}\right] \tag{184}$$

$$f_{rot} = \frac{\rho D_{ii}}{\mu_i}\left[1 + \frac{2}{\pi}\frac{A}{B}\right] \tag{185}$$

$$f_{vib} = \frac{\rho D_{ii}}{\mu_i} \tag{186}$$

$$A = \frac{5}{2} - \frac{\rho D_{ii}}{\mu_i} \tag{187}$$

$$B = z_{rot} + \frac{2}{\pi}\left(\frac{5}{3}\frac{C_{v,rot}}{R_u} + \frac{\rho D_{ii}}{\mu_i}\right) \tag{188}$$

$$C_{v,trans} = \frac{3}{2}R_u \tag{189}$$

$$z_{rot} = z_{rot}(298)\frac{F(298)}{F(T)} \tag{190}$$

$$F(T) = 1 + \frac{\pi^{3/2}}{2}\left(\frac{\varepsilon/k}{T}\right)^{1/2} + \left(\frac{\pi^2}{4} + 2\right)\frac{\varepsilon/k}{T} + \pi^{3/2}\left(\frac{\varepsilon/k}{T}\right)^{3/2} \tag{191}$$

and $z_{rot}(298)$ is the value of the rotation collision number at 298 K, defined in the transport properties of the component.

For a linear molecule:

$$C_{v,rot} = R_u \tag{192}$$

$$C_{v,vib} = C_v - \frac{5}{2}R_u \tag{193}$$

and:

- μ_i is the viscosity of component i, $\mu_i = \mu_i(T)$.
- M_i is the molecular weight of component i.
- λ_i is the thermal conductivity of component i.
- C_v is the molar specific heat at constant volume of the component.
- R_u is the universal gas constant.

For a simple gas, there is only one value of i.

Using the Elemental Composition Method

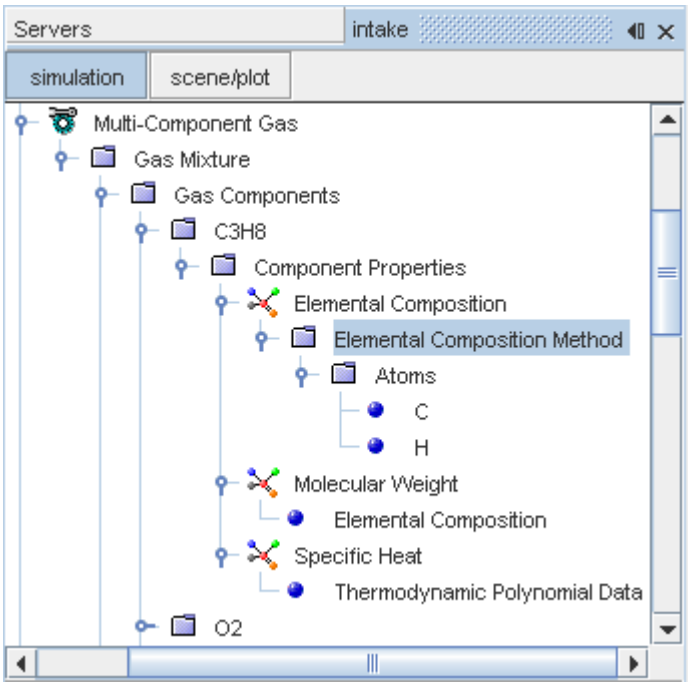
In order to calculate the molecular weight of a molecule, this method is used to describe its elemental composition.

The advantages of this method (as opposed to setting the molecular weight by a constant) are that:

- The molecular weight is automatically updated when you change the composition of the molecule; and,
- No round-off is applied to the result. Using the constant method for molecular weight incurs a round-off to the second decimal place.

The constant method is still provided for use with those materials that do not have an elemental composition (such as air), and also to provide backwards compatibility with simulations created in earlier versions of STAR-CCM+.

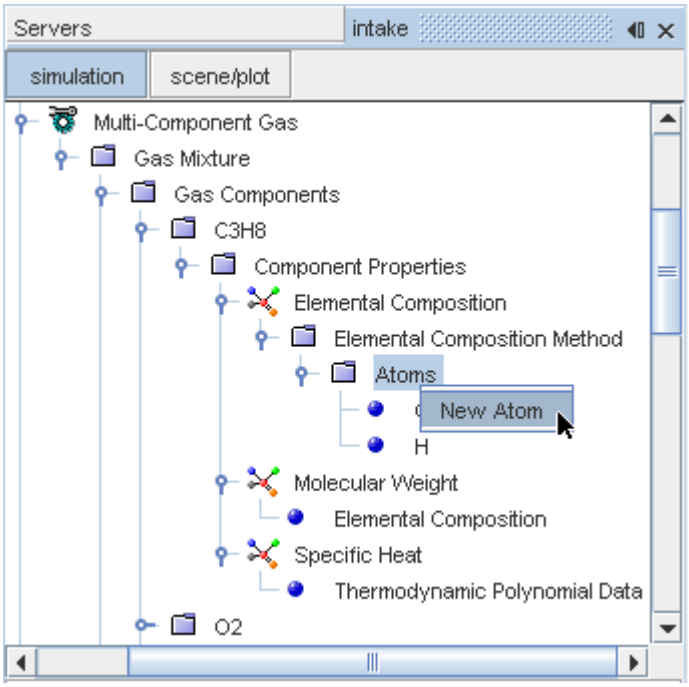
The **Elemental Composition Method** node contains a single child node, **Atoms**, that is used to manage the individual substances that make up the molecule.



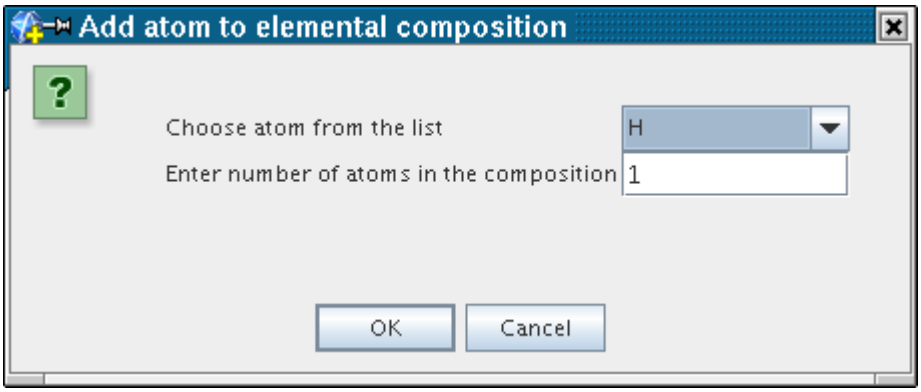
In most instances, the **Atoms** node is populated automatically based on information that is contained in the materials property database. It is, however, possible to populate this node manually.

Populating the Composition

Individual substances are added to the composition by right-clicking on the **Atoms** node and choosing **New Atom** from the pop-up menu.



Choosing **New Atom** opens the *Add atom to elemental composition* dialog, which you can use to set the substance and its quantity in the molecule. A list of available substances is provided in the drop-down menu, and an entry box accepts the number of atoms of the chosen substance



Clicking **OK** completes the specification and add a corresponding Atom child node to the **Atoms** manager node.

Removing Substances

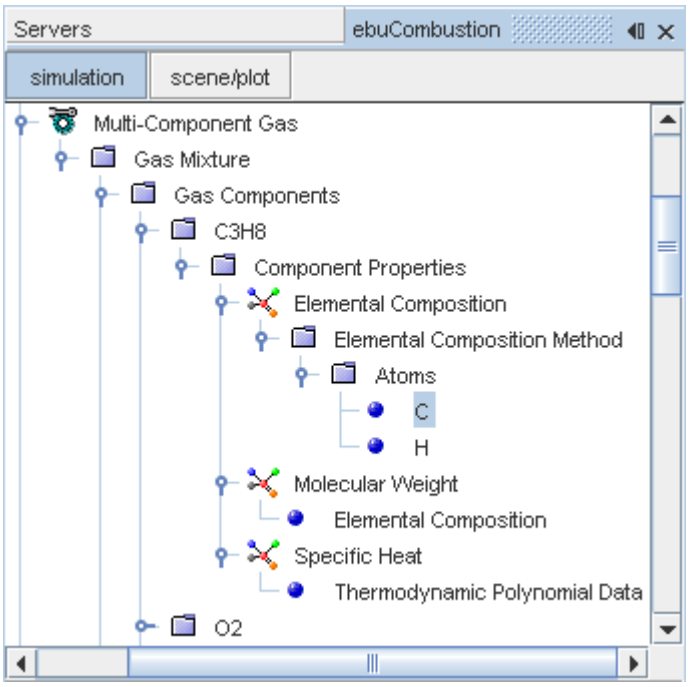
Substances can be removed from the composition by right-clicking on their respective **Atom** nodes and choosing **Delete**.

Contents:

What Is an Atom?
Working with Atoms

What Is an Atom?

The **Atom** node represents a single substance and its quantity within an elemental composition.



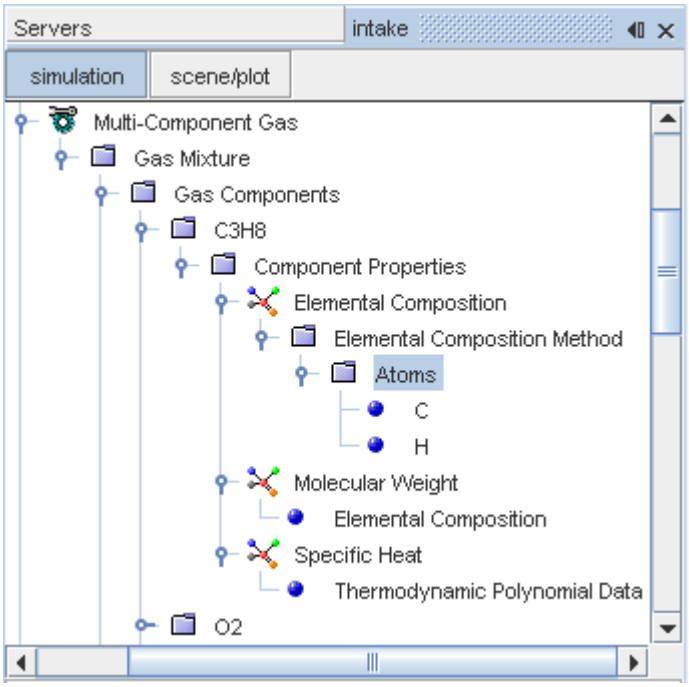
Atom Properties

Value	The number of atoms of this substance in the composition.

Working with Atoms

The **Atoms** node, which is a child of the **Elemental Composition Method** node, contains the individual substances (represented by Atom nodes) that make up a particular molecule.

This node also provides the molecular weight as a read-only property. A pop-up menu is used to populate the node.



The molecular weight W_m of the composition is calculated by summing up the individual molecular weights of each atom multiplied by the atom's quantity, that is:

$$W_m = \sum_i n_i W_i$$

where W_i is the molecular weight of a constituent atom and n_i is the number of such atoms in the molecule.

Atoms Properties

Molecular Weight	The calculated molecular weight of the composition (read-only).

Atoms Pop-Up Menu

New Atom	Opens the Add atom to elemental composition dialog for choosing an atom and its quantity.

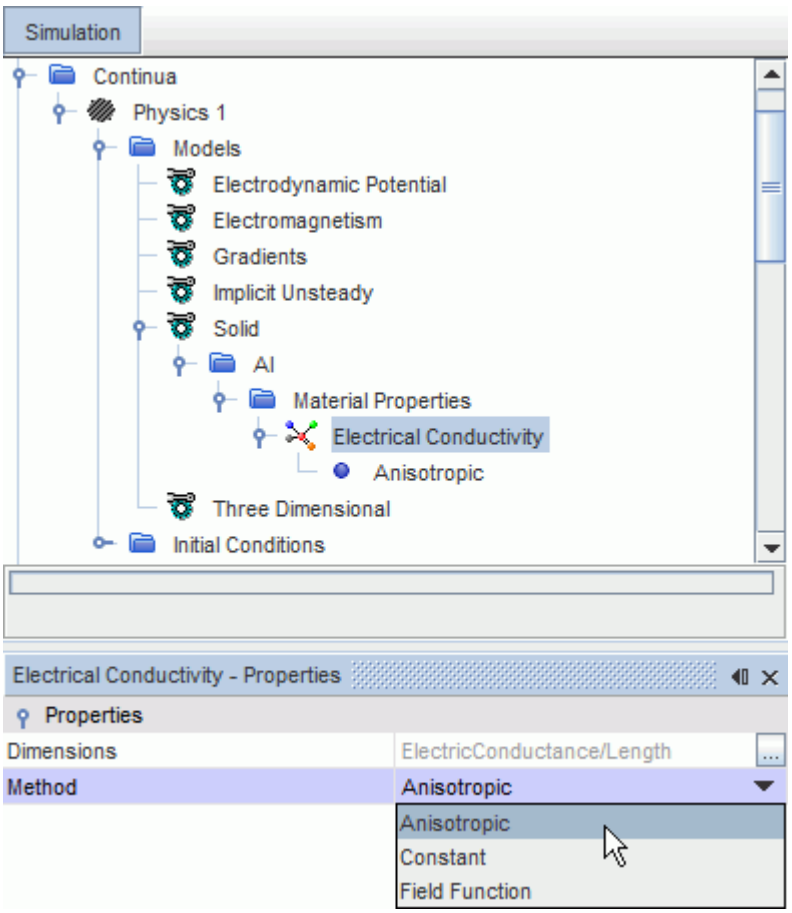
Using the Anisotropic Method

The Anisotropic method is a placeholder object to indicate that an anisotropic approach to a material property is being used.

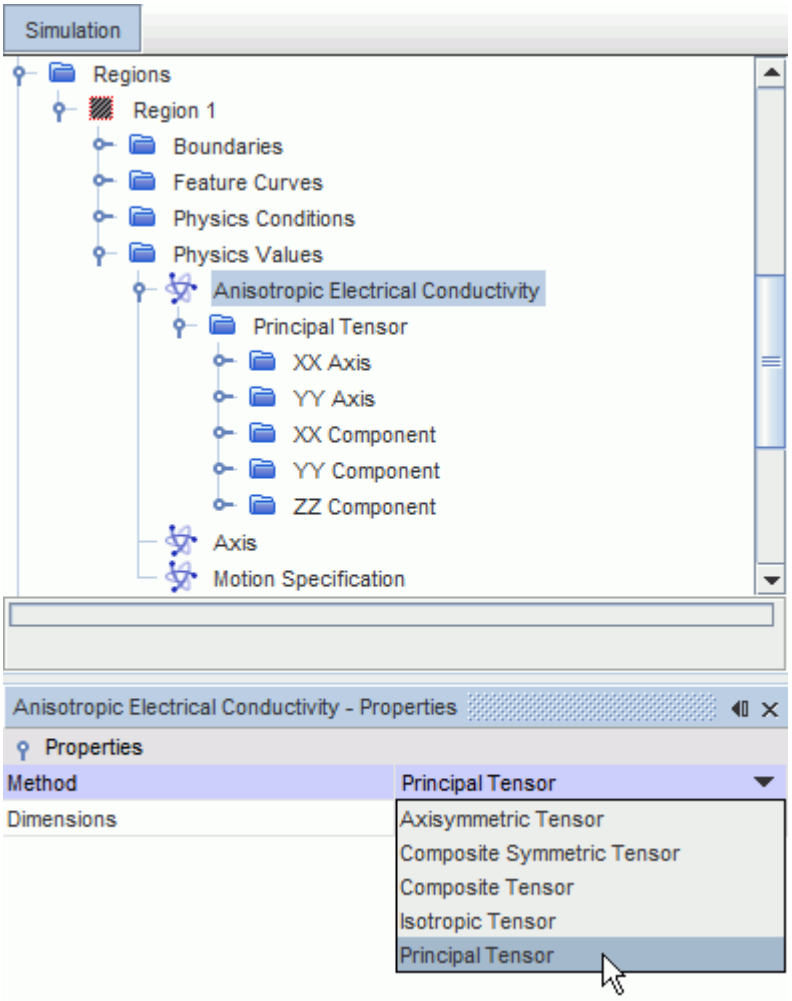
This method is available for the following material properties:

- Permittivity of a solid when the Electrostatic Potential model is active.
- Electrical Conductivity of a solid when the Electrodynamic Potential model or Magnetic Vector Potential model is active.
- Thermal Conductivity of a solid when the Coupled Solid Energy or Segregated Solid Energy model is active.

The example below is for the Electrical Conductivity material property of a solid.



The selection of this method adds the corresponding **Physics Values** node to the solid region of the simulation. The example below shows the **Anisotropic Electrical Conductivity** node.



Anisotropic Method Properties

Method	The anisotropic value is entered as a tensor profile.

Using the Viscosity Method

The dynamic viscosity of individual components is calculated with a variant of the Chapman-Enskog equation, Eqn. (181).

The dynamic viscosity of the mixture is given with the following equation:

$$\mu = \frac{1}{2}\left(\sum_1^n X_i \mu_i + \left(\sum_1^n \frac{X_i}{\mu_i}\right)^{-1}\right) \tag{194}$$

where:

- μ is the viscosity of the mixture
- μ_i is the viscosity of component i

- X_i is the mole fraction of component i

Using the Thermal Conductivity Method

The thermal conductivity of individual components is calculated from kinetic theory. The thermal conductivity of the mixture is given with the following equation:

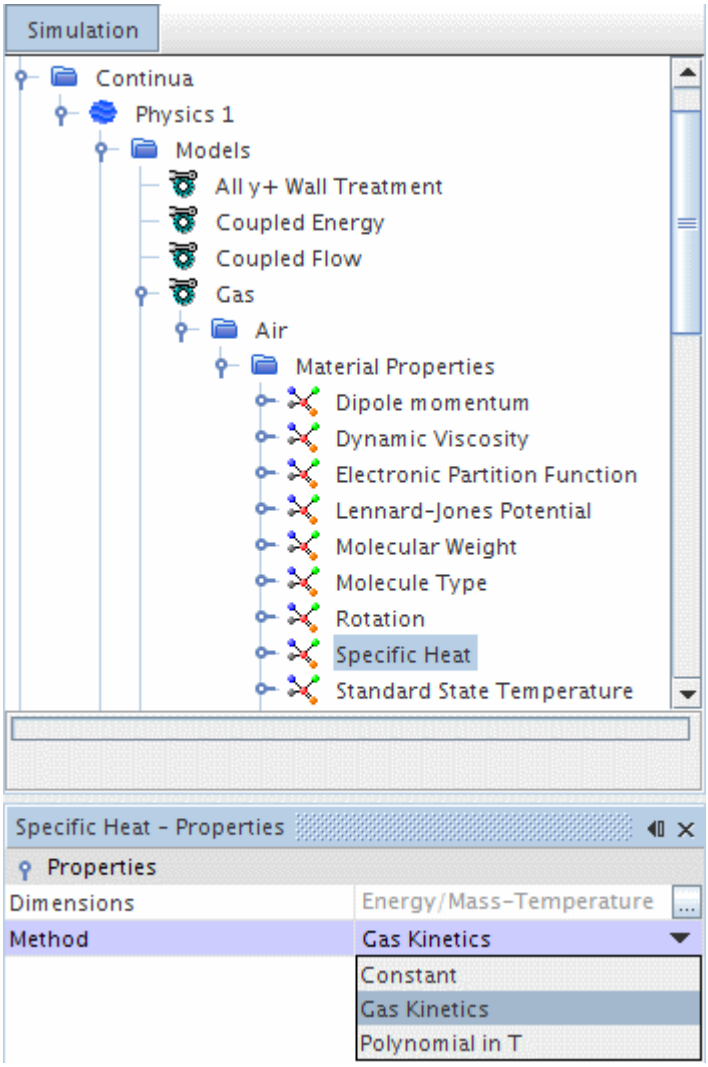
$$\lambda = \frac{1}{2}\left(\sum_1^n X_i \lambda_i + \left(\sum_1^n \frac{X_i}{\lambda_i}\right)^{-1}\right) \tag{195}$$

where:

- λ is the conductivity of the mixture
- λ_i is the conductivity of component i
- X_i is the mole fraction of component i

Using the Gas Kinetics Method for Specific Heat

This method becomes available with single- and multi-component gases, together with the Ideal Gas, Constant Density, or Thermal Non-Equilibrium Ideal Gas models, or with any of the Real Gas models: Peng-Robinson, Redlich-Kwong, Soave-Redlich-Kwong, Modified Soave-Redlich-Kwong, and Van der Waals).



The specific heat of a gas consists of contributions from translational, rotational, vibrational and electronic energy [29]:

$$C_{p,total} = C_{p,trans} + C_{v,rot} + C_{v,vib} + C_{v,elec} \tag{196}$$

where:

- C_p values are specific heats at constant pressure.
- C_v values are specific heats at constant volume.

For each energy type, the contribution to the specific heat is calculated based on the associated partition function, through the following formula:

$$\begin{gathered} \frac{e_i}{R} = T^2 \frac{\partial}{\partial T} \ln(Q_i) \\ \frac{C_{v,i}}{R} = \frac{T^2}{Q_i}\left[\frac{\partial^2 Q_i}{\partial T^2} + \left(\frac{\partial Q_i}{\partial T}\right)^2 + \frac{2\partial Q_i}{T\partial T}\right] \end{gathered} \tag{197}$$

where:

- e_i is energy of type i.
- i is $trans$ (translational), rot (rotational), vib (vibrational), or $elec$ (electronic)
- R is the gas constant.
- T is the temperature.
- Q_i is the partition function for energy of type i.

To determine the total specific heat, the partition function for each energy type must be specified. The combined translational and rotational energy $C_{p,TR} = C_{p,trans} + C_{v,rot}$ can be simplified because:

- For the translational energy, the specific heat is effectively constant due to the small spacing between quantum energy levels.
- The rotational energy is assumed fully excited, since the spacing between quantum energy levels is negligible at all but cryogenic temperatures. Rotational energy is zero for monatomic molecules.

The combined translational-rotational specific heat is given as:

$$\begin{aligned} C_{p,TR} &= \frac{5}{2}R \quad \text{for monatomic molecules} \\ C_{p,TR} &= \frac{7}{2}R \quad \text{for linear molecules} \\ C_{p,TR} &= \frac{8}{2}R \quad \text{for non-linear molecules} \end{aligned} \tag{198}$$

The partition function for electronic energy must be specified explicitly for this method. For polyatomic molecules, the partition function for vibrational energy must also be specified. These functions are both specified using a series of modes with characteristic temperature (θ_i) and degeneracy (g_i). The partition function for each of these modes is defined as:

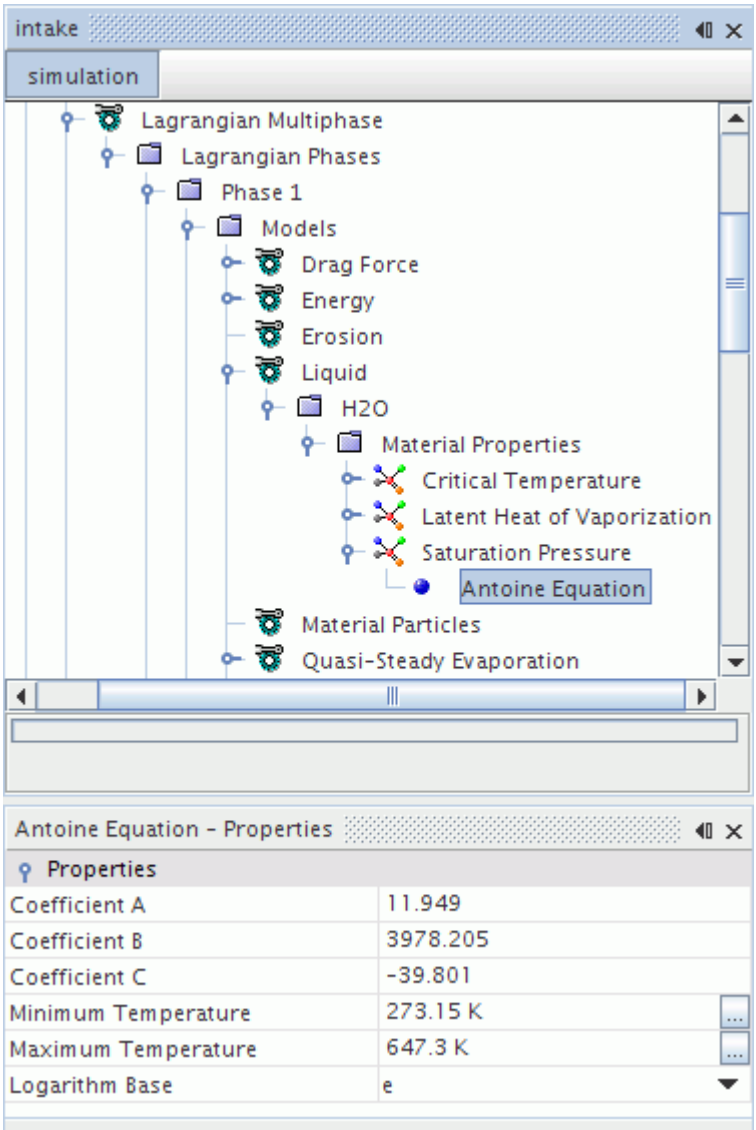
$$\begin{aligned} Q_{vib} &= \prod_i \left(1 - e^{-\theta_i/T}\right)^{-g_i} \quad \text{for polyatomic molecules} \\ Q_{el} &= \sum_i g_i e^{-\theta_i/T} \quad \text{for all molecules} \end{aligned} \tag{199}$$

These degeneracies and characteristic temperatures can be found for various molecules in [25] and [28]. For N_2, O_2, NO, N, and O, these values are in the standard material database provided as part of STAR-CCM+.

Using the Antoine Equation

The Antoine equation is available for the **Saturation Pressure** material property.

The coefficient properties for this equation are set in the *Properties* window.



The Antoine equation for saturation pressure [27] is based on the logarithm of the ratio P_{sat}/P_{atm} to a base of e or 10:

$$\log_n\left(\frac{p_{sat}}{p_{atm}}\right) = A - \frac{B}{T + C} \tag{200}$$

where p_{atm} is 1 bar, T is temperature (K), n is the base of the logarithm, and A, B and C are equation coefficients. Values of the coefficients for many substances are tabulated in [27]. Coefficients must be consistent with the temperature, T, being an absolute temperature in K.

For temperatures above the Maximum Temperature, the saturation pressure is evaluated using the Maximum Temperature; similarly for temperatures below the Minimum Temperature, the Minimum Temperature is used.

Antoine Equation Properties

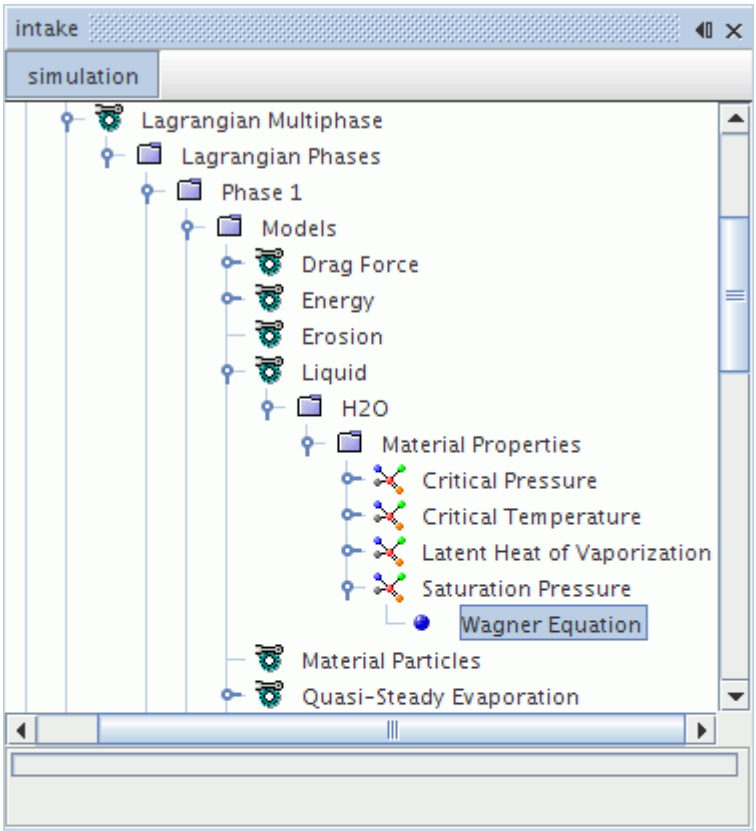
Coefficient A	The coefficient A in Eqn. (200) .	

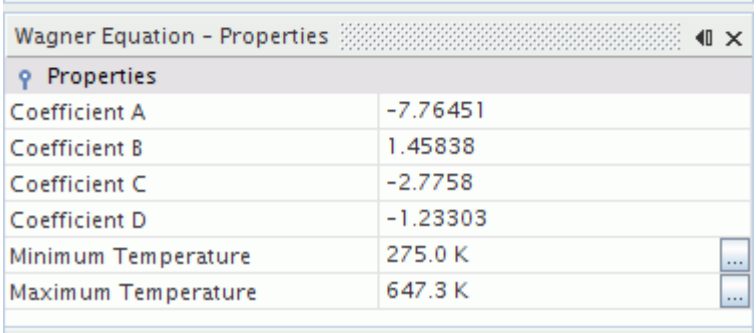
Coefficient B	The coefficient B in Eqn. (200) .	
Coefficient C	The coefficient C in Eqn. (200) .	
Minimum Temperature	Specifies the lower limit of the temperature range over which this equation is valid. The default value, 273.15K, is for water. Tables such as those in [27] give minimum temperature values for various substances.	
Maximum Temperature	Specifies the upper limit of the temperature range over which this equation is valid. The default value, 647.3K, is for water. Tables such as those in [27] give maximum temperature values for various substances.	
Logarithm Base	Specifies the base n in Eqn. (200) .	
	e	Coefficients are calibrated for base e, as given in [27].
	10	Coefficients are calibrated for base 10.

Using the Wagner Equation

The Wagner equation is available for the **Saturation Pressure** material property.

The coefficient properties for this equation are set in the *Properties* window.





The Wagner equation for saturation pressure [27] is:

$$p_{sat} = p_c \exp\left(\frac{A\tau + B\tau^{1.5} + C\tau^3 + D\tau^6}{T_r}\right) \tag{201}$$

in which p_c is the **Critical Pressure** material property, T_r the reduced temperature T/T_c, T_c the **Critical Temperature** material property, $\tau = 1 - T_r$ and A, B, C and D are equation coefficients. Values of the coefficients for many substances are tabulated in [27].

For temperatures above the Maximum Temperature, the saturation pressure is evaluated using the Maximum Temperature; similarly for temperatures below the Minimum Temperature, the Minimum Temperature is used.

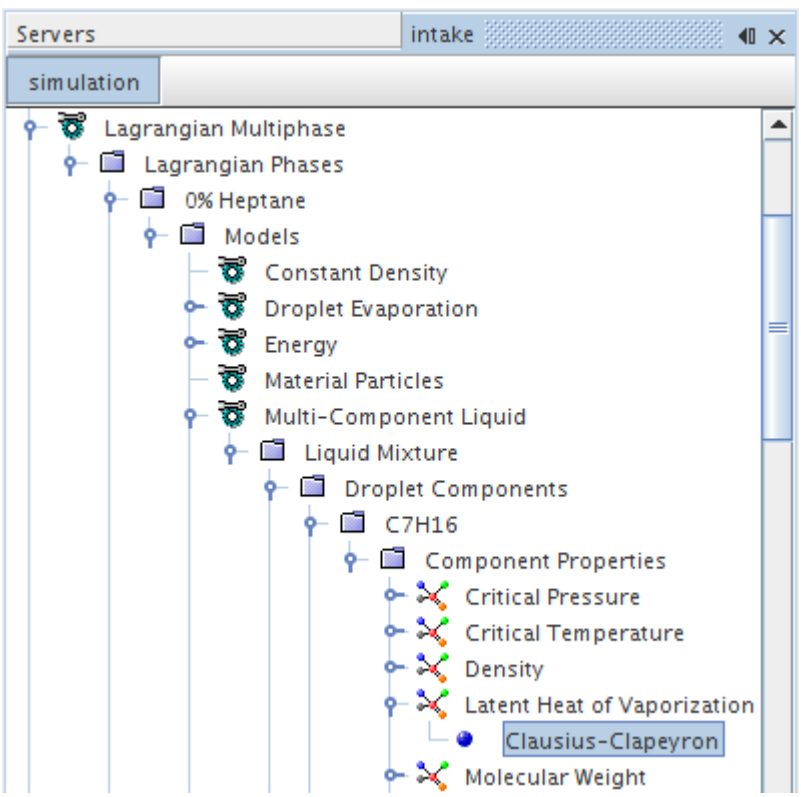
Wagner Equation Properties

Coefficient A	The coefficient A in Eqn. (201) .	

Coefficient B	The coefficient B in Eqn. (201) .
Coefficient C	The coefficient C in Eqn. (201) .
Coefficient D	The coefficient D in Eqn. (201) .
Minimum Temperature	Specifies the lower limit of the temperature range over which this equation is valid. The default value, 275K, is for water. Tables such as those in [27] give minimum temperature values for various substances.
Maximum Temperature	Specifies the upper limit of the temperature range over which this equation is valid. The default value, 647.3K, is for water. Tables such as those in [27] give maximum temperature values for various substances.

Using the Clausius-Clapeyron Method

The Clausius-Clapeyron method uses the Clapeyron equation and the assumption of an ideal gas equation of state for the vapor to relate the latent heat of vaporization to the saturation pressure.



The Clausius-Clapeyron method is available for the **Latent Heat of Vaporization** if the **Saturation Pressure** is given by the Antoine Equation, Wagner Equation, Polynomial in T, or Table(T) methods.

The Clausius-Clapeyron method uses the following expression:

$$L = \frac{RT^2}{p_{sat}} \cdot \frac{dp_{sat}}{dT} \tag{202}$$

It is the need for dp_{sat}/dT which limits the choice of compatible methods for p_{sat}. The ideal gas constant implies that the **Molecular Weight** material property must be supplied.

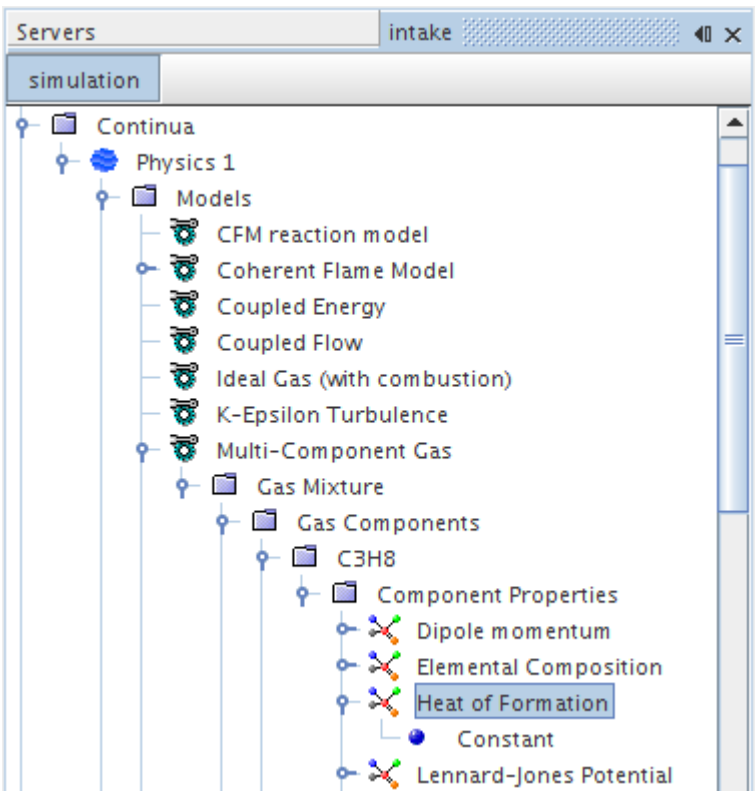
Clausius-Clapeyron Properties

None

Using the Heat of Formation

The Heat of Formation is the heat that is evolved when 1 kilogram of the substance is formed from its elements in their respective standard states [J/kg].

This value node (**Heat of Formation** in the following screenshot) is activated if the *Specific Heat* property of a mixture component in an EBU or CFM combustion model is to be calculated by a method other than **Thermodynamic Polynomial Data**. The supplied quantity is always a constant value.



Using the Leidenfrost Temperature

The Leidenfrost temperature is the temperature at which the Leidenfrost effect appears. This effect creates a vapor barrier between liquid and solid phases, reducing heat transfer and adhesion.

Choosing Bai-Gosman Wall Impingement or the Fluid Film Boiling model makes the Leidenfrost temperature available in nodes for material properties of liquids under **Lagrangian Multiphase** > **Lagrangian Phases** > **Phase 1** > **Models** or under **Fluid Film** > **Fluid Film Phases** > **Film Phase** > **Models**. You can set the Leidenfrost temperature by selecting one of five methods:

- Constant: Set the temperature to a constant value, or use the default value, 546.03 K, for water at 1 atm.
- Field Function: Define a scalar function determining the temperature.
- Spiegler method:

$$T_L = \frac{27}{32} T_c = 0.84375 T_c \tag{203}$$

  Use this method when pressure is much lower than the critical pressure P_c.

- Lienhard method:

$$T_L = T_c \left(0.905 + 0.095 \left(\frac{T_{sat}}{T_c} \right)^8 \right) \tag{204}$$

  Use this method when pressure is close to the critical pressure P_c.

- Habchi method:

$$T_L = T_{sat} + \Delta T \tag{205}$$

  where

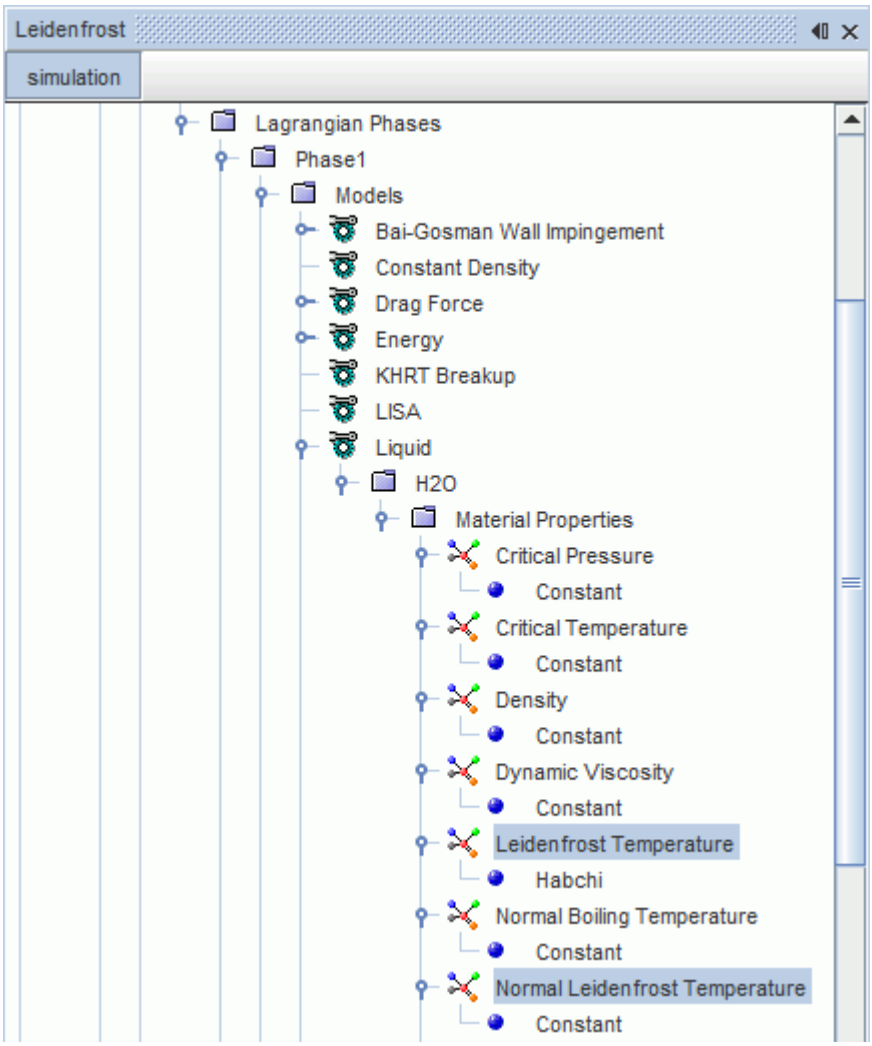
$$\Delta T = \begin{cases} T_L{}^1 - T_b & \text{if } (P \leq 1) \\ \frac{\left(T_L{}^1 - T_b\right) - A}{T_c - T_b} \left(T_c - T_{sat}\right) + A & \text{if } (P > 1) \end{cases}$$

  Use this method when pressure P is close to the critical pressure P_c.

In the preceding equations:

- T_L is the Leidenfrost temperature.
- T_c is the critical temperature. For multi-component drops, T_c is the highest of the critical temperatures of the components.
- T_{sat} is the saturation temperature (boiling point). The saturation temperature is the temperature at which the vapor pressure of the liquid equals the pressure in the cell. It is calculated internally from the saturation pressure p^*. You can define saturation pressure under the **Material Properties** > **Saturation Pressure** node of the Liquid model. See Calculating Saturation Temperature.
- $T_L{}^1$ is the “normal Leidenfrost temperature,” the value at 1 atm. This is a model parameter, to be determined experimentally or estimated. For multi-component drops, choose $T_L{}^1$ as a function of composition.

- T_b is the normal boiling temperature at 1 atm. For multi-component drops, choose T_b as a function of composition.
- P is pressure.
- A is the larger of 1 or $T_L{}^1 - T_c$.



Contents:

Calculating Saturation Temperature

See Also:

Modeling Bai-Gosman Wall Impingement
Modeling Bai-ONERA Wall Impingement

Calculating Saturation Temperature

The Speigler, Leinhard and Habchi methods for Leidenfrost temperature require the saturation temperature T_{sat} as part of their solution.

When the saturation temperature method is set to **Iterative**, the saturation temperature for the particle is calculated implicitly from the saturation pressure relations of the material. The temperature is calculated using a hybrid bisection/Newton scheme, which finds the temperature at which the vapor pressure of the droplet material is equal to the absolute pressure in the cell.

For multi-component droplet materials, the partial pressure of each component is mass summed to obtain the total vapor pressure. For the numerical method to be successful, the vapor pressure vs. temperature relation must be monotonically increasing.

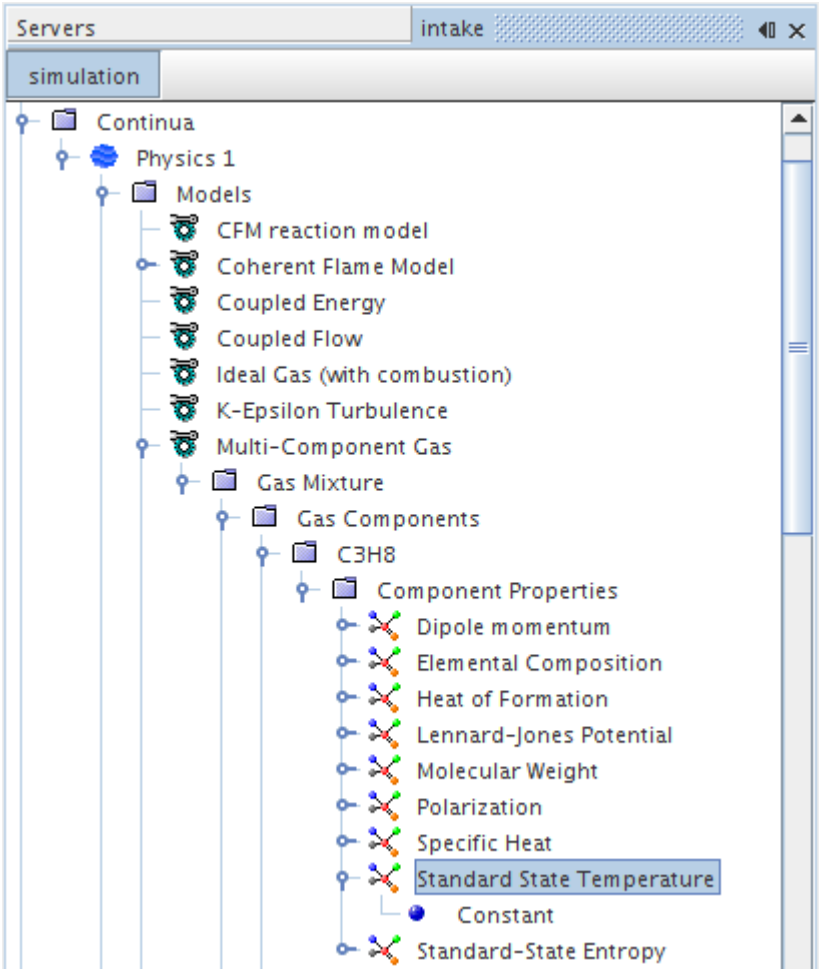
This temperature is available under the **Material Properties** > **Saturation Temperature** node of the Liquid model. Once calculated, it becomes available as a field function. It does not override the Lower Transition Temperature specified in the Bai-Gosman Wall Impingement model.

Using the Standard State Temperature

The Standard State Temperature is the temperature at which the standard state of a substance is defined. The commonest choice is 298.15 K (25.00 C).

Standard State Temperature is used to calculate enthalpy. The enthalpy value varies with Standard State Temperature, but in direct proportion, so differences in enthalpy remain the same for different values of Standard State.

Among other circumstances, the **Standard State Temperature** node appears when the chosen method for specific heat is either **Gas Kinetics** or **Polynomial in T**. The supplied quantity is always a constant value. Be sure to supply thermodynamic data for different phases or species in a consistent manner.



See Also:

Latent Heat of Vaporization

Modeling Energy in Multiphase Segregated Flow

Using the Binary Diffusion Coefficient Method for Molecular Diffusivity

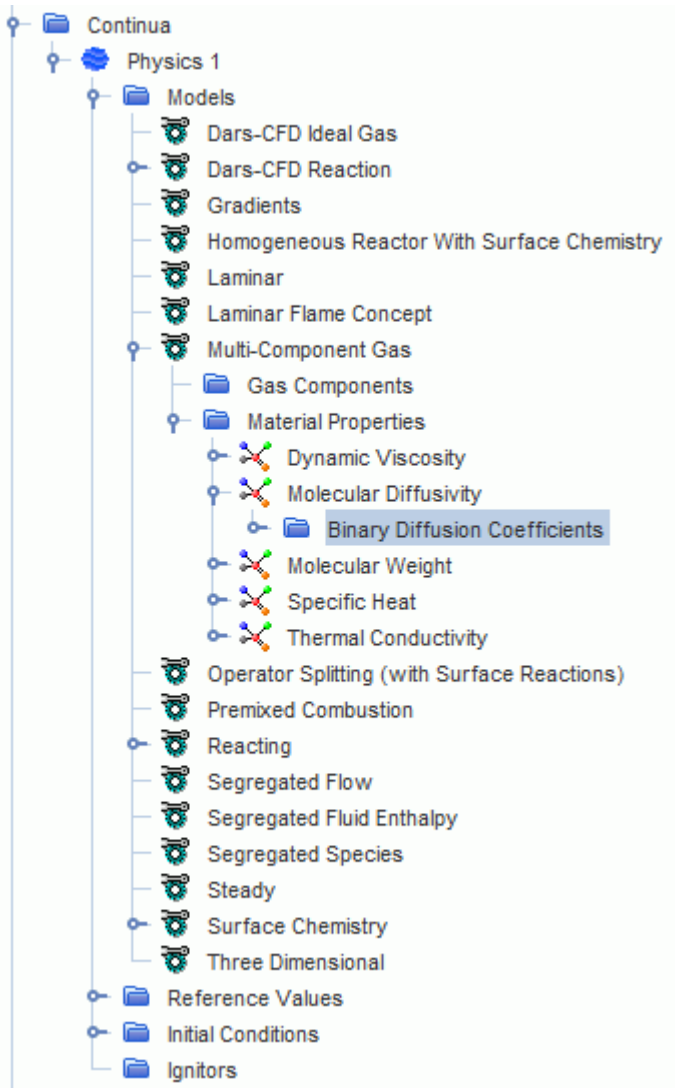
This method uses specified binary diffusion coefficients between the components of the mixture to calculate a molecular diffusivity for each component that is diffusing into the mixture as a whole.

For a mixture containing N components, this method requires the specification of $(N(N-1))/2$ diffusion coefficients.

This specification enforces the fact that self-diffusion is zero and the binary diffusion coefficients are symmetric. An array profile specifies these binary diffusion coefficients with all the standard methods available for specifying the individual profiles. The diffusivity of each compound is calculated as:

$$D_m = \frac{1 - x_m}{\sum_{i \neq m} \frac{x_i}{D_{i,m}}} \tag{206}$$

To expose this method, select the **Material Properties** > **Molecular Diffusivity** node for a multi-component gas or liquid, then set *Method* to **Binary Diffusion Coefficients**. The **Molecular Diffusivity** > **Binary Diffusion Coefficients** node appears.



Using the Kinetic Theory Method for Molecular Diffusivity

The Kinetic Theory method calculates the molecular diffusivity of an individual component using kinetic theory.

When this method is used, the following properties show up on the mixture component level:

- Dipole momentum—a measure of polarity of a covalent bond in the molecule (Debye)
- Lennard-Jones characteristic length—the collision diameter (always given in Angstroms)

- Lennard-Jones energy—the potential energy of attraction (given in J/kmol)
- Polarization—polarizability of a molecule (Cubic Angstroms)

These properties are entered as **Constant** values only. These values are used for calculating the molecular diffusivity of the individual component for the Kinetic Theory method.

To expose this method, select the **Material Properties** > **Molecular Diffusivity** node for a multi-component gas or multi-component liquid, then set *Method* to **Kinetic Theory**. The **Molecular Diffusivity** > **Kinetic Theory** node appears.

The Molecular Diffusivity property on a mixture level is different than other properties – it actually calculates the molecular diffusivities for each individual component.

The molecular diffusivity D_m of component m is calculated as:

$$D_m = \frac{1 - X_i}{\sum_{i=1, i \neq m}^{N} \frac{X_i}{D_{i,m}}} \tag{207}$$

where the mole fraction, X_i, is related to the mass fraction, Y_i, as:

$$X_i = \frac{Y_i M_w}{M_i} \tag{208}$$

- M_i is the molecular weight of component i
- M_w is the molecular weight of the mixture
- $D_{i,m}$ is the binary diffusivity of components i and m

From gas kinetic theory, the expression of $D_{i,m}$ is based on Chapman-Enskog [26]:

$$D_{i,m} = \frac{2.66 \times 10^{-7} T^{3/2}}{p M_{i,m}^{1/2} \sigma_{i,m}^2 \Omega(T^*)} \tag{209}$$

where:

$$M_{i,m} = \frac{2 M_i M_m}{M_i + M_m} \tag{210}$$

and:

- $\sigma_{i,m}$ is the Lennard-Jones characteristic length, the collision diameter for the pair of component i and component m. $\sigma_{i,m}$ is a function of the Lennard-Jones collision diameter material property, σ_i, that is specified for each component i of the mixture. $\sigma_{i,m}$ is different for polar and non-polar molecules (gases).
- p is the (absolute) static pressure
- $\Omega(T^*)$ is a collision integral that is a function of the reduced temperature, T^*. T^* is defined similarly to the reduced temperature for the viscosity calculation:

$$T^* = \frac{kT}{\varepsilon_{i,m}} \tag{211}$$

where:

- k is the Boltzmann constant $= 1.3806503 \times 10^{-23}$ m^2 kg s^{-2} K^{-1}
- T is temperature
- $\varepsilon_{i,m}$ is the characteristic Lennard-Jones energy for the pair of component i and component m. $\varepsilon_{i,m}$ is a function of the Lennard-Jones potential energy material property, ε_i, that is specified for each component i of the mixture. $\varepsilon_{i,m}$ is different for polar and non-polar molecules (gases).

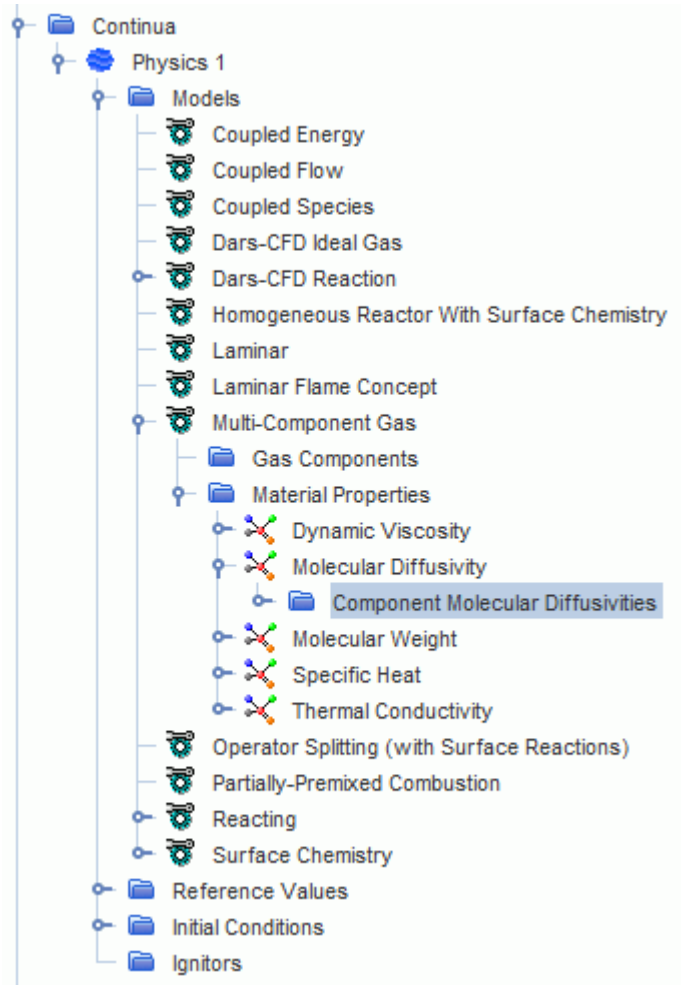
Using the Component Molecular Diffusivities Method for Molecular Diffusivity

This method allows you to specify the molecular diffusivity of separate components in a multi-component mixture, for cases where the other methods are not sufficient to predict it accurately.

Some examples of this situation are:

- Simulations with diffusion dominated flows (that is, where convection is negligible or where there is no turbulence); and,
- Simulations containing mixture components that have significantly different molecular weights (such as H2 with fast diffusion and C10H22 with slow diffusion).

To expose this method, select the **Material Properties** > **Molecular Diffusivity** node for a multi-component gas or liquid. Then, set *Method* to *Component Molecular Diffusivity*. The **Molecular Diffusivity** > **Component Molecular Diffusivities** node appears



When this option is set, you can specify the diffusivities for each component directly using an array profile.

All the usual methods for a scalar profile quantity are available.

Using the Nernst-Einstein Method for Charged Species

The Nernst-Einstein method can be set to determine the mobility of charged species from the diffusivity, or the diffusivity from the mobility.

You can use this method to define the charged species mobility or diffusivity and determine the overall migration of ionic species in electrolytic solutions.

	Computing diffusivity from mobility (mobility is set to constant and molecular diffusivity is set to Nernst-Einstein)	**Computing mobility from diffusivity** (mobility is set to Nernst-Einstein and molecular diffusivity is set to constant)

When mobility specified as u_i, $\left[\mathrm{m^2 kmol/J\,s}\right]$:	Diffusivity, D_i, is determined from the mobility, u_i, using Eqn. (1112). (This scenario is default when using the Electrodynamic Potential model.)	Mobility, u_i is determined from the diffusivity, D_i, according to Eqn. (1112).
When mobility specified as K_i, $\left[\mathrm{m^2/Vs}\right]$:	Diffusivity, D_i, is determined from the mobility, K_i, using Eqn. (1112) (in which u_i is determined from Eqn. (1124)).	Mobility, K_i, is determined from the diffusivity, D_i, according to Eqn. (1124) (in which u_i is determined from Eqn. (1112)). (This scenario is default when using the Electrostatic Potential model.)

Nernst-Einstein can be set as the Method for either Mobility or Molecular Diffusivity, but not for both. Therefore, to change the default settings for Mobility and Molecular Diffusivity, de-select Nernst-Einstein from one of the material properties methods before attempting to select it for the other. You can manually change the mobility dimensions from the default settings using the Electrochemical Species model property Charged Species Mobility Dimension.

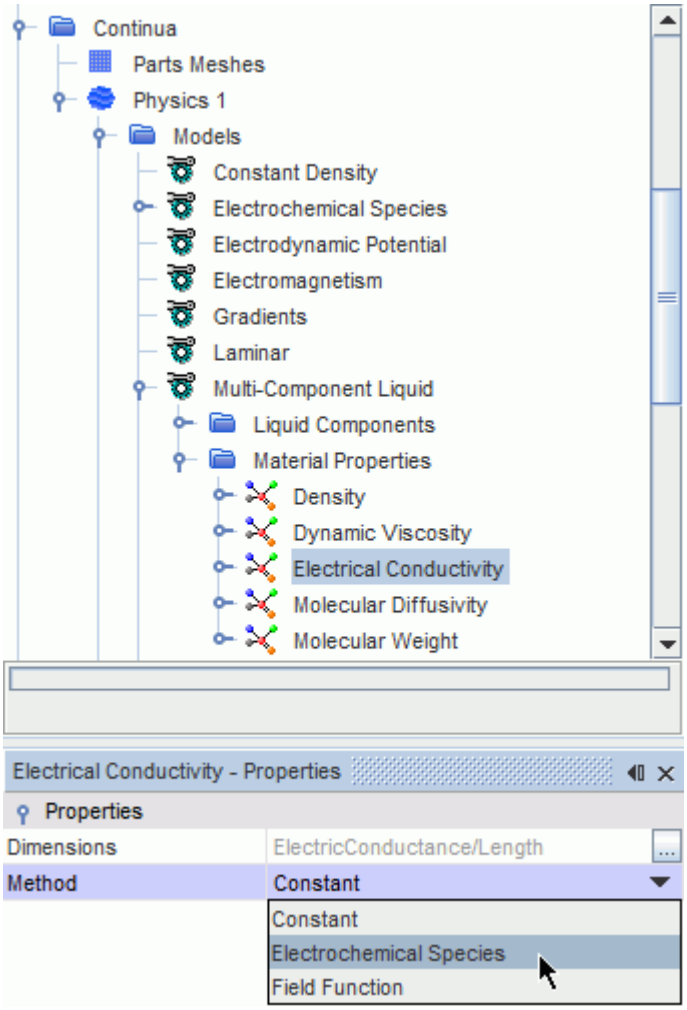
You can also define the *Method* using a **Field Function**. See Using Field Functions.

Note: If the physics continuum contains an energy model, the value of the local temperature is used for T in Eqn. (1112). If no energy model is active, a default value of $293.15K = 20°C$ is used.

Using the Electrochemical Species method for Electrical Conductivity

When using the Electrodynamic Potential model, to obtain a consistent setup for applications which require electroneutrality, specify the Electrical Conductivity *Method* as **Electrochemical Species**.

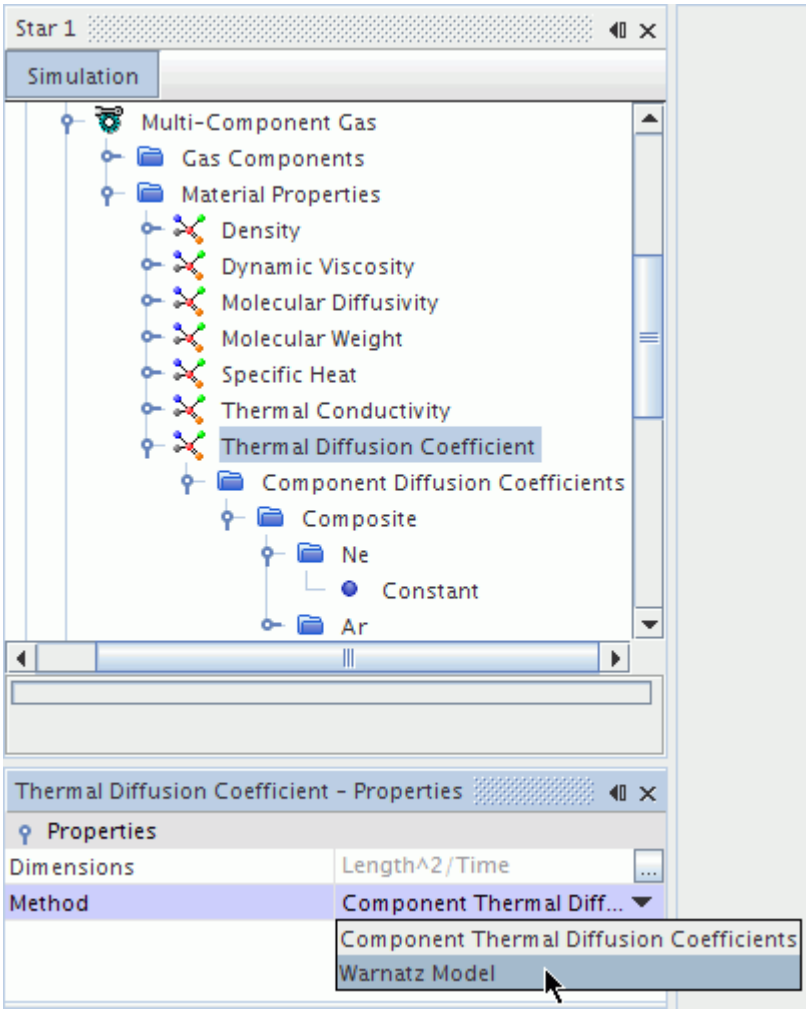
This option sets the electrical conductivity according to Eqn. (1111).



Using the Thermal Diffusion Coefficient

When the Soret Effect option is selected, you must specify the thermal diffusion coefficient for each component.

The coefficients can be specified directly using the **Component Thermal Diffusion Coefficients** material property method. For gases, the coefficients can also be specified using the **Warnatz Model** method, which calculates the coefficient based on kinetic theory (see Eqns. Eqn. (358) through Eqn. (360)).



Equilibrium Speed of Sound

This method of calculating the speed of sound applies in a wide range of compressible fluids in equilibrium states—that is, with properties dependent only on temperature and pressure.

This method gives the speed of sound c in terms of density derivatives:

$$c^2 = \frac{1}{\left(\frac{\partial \rho}{\partial P}\right)_T - \frac{T}{\rho^2 C_p}\left(\frac{\partial \rho}{\partial T}\right)_P^2} \tag{212}$$

where C_p is specific heat and density ρ is a function of temperature T and pressure P.

Materials Bibliography

[17] *Carreau, P.J. 1972. "Rheological equations from molecular network theories", J. Rheol., 16(1), pp. 99-127.*

[18] *Cross, M.M. 1965. "Rheology of non-Newtonian fluids: A new flow equation for pseudoplastic systems", J. Colloid Sci., 20(5), pp. 417-437.*

[19] *Escudier, M.P., Poole, R.J., Dales, C., Nouar, C., Desaubry, C., Gresham, L. and Pullum, L. 2005. Observations of asymmetrical flow behaviour in transitional pipe flow of yield-stress and other shear thinning liquids J. Non-Newtonian Fluid Mechanics, 127, pp. 143-155.*

[20] *Ferry, John D., 1980. Viscoelastic Properties of Polymers, 3rd edition, Wiley.*

[21] *International Association for the Properties of Water and Steam. 2007. Revised Release on the IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam, IAPWS Secretariat, Lucerne, Switzerland.*

(This material is available at www.iapws.org.)

[22] *International Association for the Properties of Water and Steam. 2008. Release on the IAPWS Formulation 2008 for the Viscosity of Ordinary Water Substance, IAPWS Secretariat, Berlin, Germany.*

(This material is available at www.iapws.org.)

[23] *International Association for the Properties of Water and Steam. 2011. Release on the IAPWS Formulation 2011 for the Thermal Conductivity of Ordinary Water Substance, IAPWS Secretariat, Plzen, Czech Republic.*

(This material is available at www.iapws.org.)

[24] *Maples, R. E. 2000. Petroleum Refinery Process Economics, Pennwell Books.*

[25] *Moore, C. E. 1971. Atomic Energy Levels as Derived From the Analyses of Optical Spectra.*

[26] *Poling, B.E., Prausnitz, J.M., O'Connell, J.P., 2001. The Properties of Gases and Liquids, 5th edition, McGraw-Hill, New York.*

[27] *Reid, R.C., Prausnitz, J.M., and Poling, B.E., 1987. The properties of gases and liquids, 4th edition, McGraw-Hill, New York.*

[28] *Rosen, B., ed. 1970. "Spectroscopic Data Relative to Diatomic Molecules", International Tables of Selected Constants, 17.*

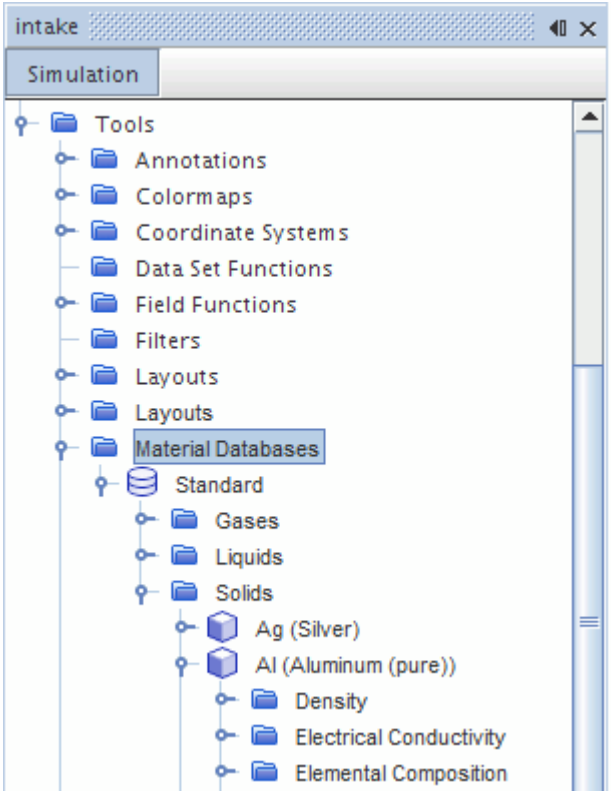
[29] *Vincenti, W. G., and Kruger, C. H. 1965. Introduction to Physical Gas Dynamics.*

[30] *Yasuda, K. 1979. Investigation of the analogies between viscometric and linear viscoelastic properties of polystyrene fluids. PhD Thesis, Dept. of Chemical Engineering, Massachusetts Institute of Technology.*

Material Databases

The material database node within the **Tools** > **Material Databases** node contains the objects that represent the information from the material database.

When you create a simulation, this node appears with its sub-nodes in the object tree as soon as you select a material model.



Its contents are grouped among three sub-nodes that represent database folders: **Gases**, **Liquids**, and **Solids**. You can also create database folders as well as create database material properties.

The source of the material database is a `props.mdb` file, that is located in your main STAR-CCM+ installation directory (Windows) or in the star subdirectory of that directory (Linux). It is possible to load one or more alternate databases. You can also update the database through the pop-up menu.

A second material database is supplied called `propsExt.mdb`, which contains a larger selection of materials that are rarely used.

If you set up a database, the data you enter in the STAR-CCM+ GUI is not saved until you explicitly export it to an external file.

Material Databases Manager Right-Click Menu

Right-click the **Material Databases** node to access the following menu:

Open Material Database...

Opens an available material database file (`.mdb` format only) from a drop-down list in a separate dialog. This list is updated when the simulation is opened.

New Material Database...

Adds a material database object to the simulation.

Material Database Properties

Individual databases, such as **Standard** in the preceding screenshot, have the following properties:

File name

The name of the database file from which the material database object gets its material data.

File path

The path to the database file.

Save When Simulation Is Saved

When this property is **On**, changes to the database are saved when you save the simulation. The default is **Off**.

Material Database Right-Click Menu

Right-click the node for the individual database, such as **Standard** in the preceding screenshot, to access the following menu:

New Folder

Creates a database folder.

Import Properties...

Imports material data from a file (`.dbs` format only).

Reload

Reloads the material database.

Save

Saves the material database changes to the external file. Do not use this command with the **Standard** material database object before you archive a copy of the supplied `props.mdb` file.

How the Material Data Is Used in the Simulation File

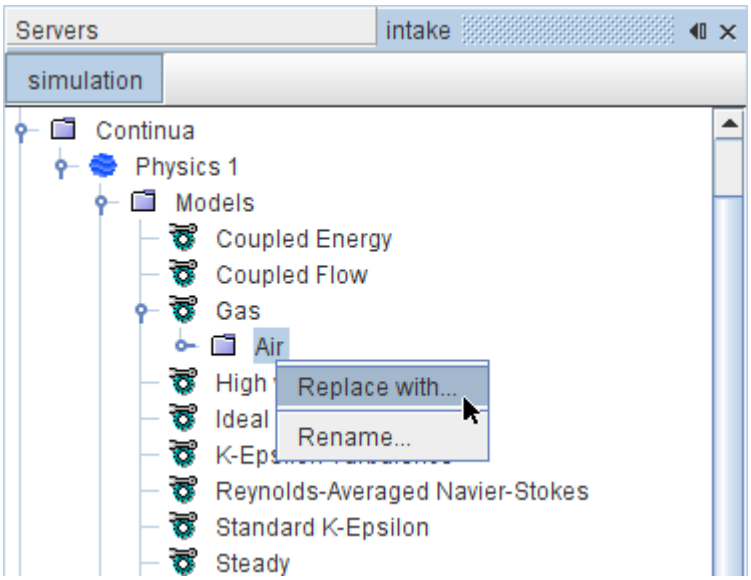
It would be too costly in terms of disk space to duplicate all material data from the `props.mdb` file into every simulation file. Therefore a simulation file only retains the information on materials that are referenced by material model selection.

This information includes:

- Names of referenced materials

- The database folder (gas, liquid, or solid) to which each referenced material belongs
- Properties of referenced materials
- Values of the properties

To make data of other materials available, STAR-CCM+ accesses the `props.mdb` file directly. This action occurs when, for example, you want to replace a selected material.



Available replacement materials are listed in the *Replace Material* dialog.

This list is why the `props.mdb` file must be in its proper place when you start a new simulation, or when you open an existing simulation and intend to replace materials. The proper place is your main STAR-CCM+ installation directory (Windows) or the star subdirectory of that directory (Linux).

Note: When the materials database does not contain specific properties for a material, STAR-CCM+ inserts default values instead. The default values are those for air (gases), water (liquids), and aluminum (solids).

Additionally, if multiple databases were used in creating a simulation, all of those databases must be in the places they occupied when STAR-CCM+ loaded them into the simulation. Otherwise any unique data that they contain is not accessible within the simulation unless you load the databases again.

Database Folders

In the default **Standard** material database, the database folders are used to group database materials as gases, liquids, and solids.

Database Folder Pop-Up Menu

New Material

Creates a database material from a submenu: **New Gas**, **New Liquid**, or **New Solid**.

New Folder

Creates a folder within the database folder. Folders can be useful for organizing many materials. For an example, see "Mixture Folders," following.

Mixture Folders

If you often use multi-component mixtures, you can use material database folders to define these mixtures and then save them in the material database for later re-use. Use the following steps:

1. Using the pop-up menu of the **Standard** node, create a folder for the mixture and give it an appropriate name, such as `Air`.
2. Copy and paste the components of the mixture from pre-existing databases. For example, copy **N2 (Nitrogen)**, **O2 (Oxygen)**, and **Ar (Argon)** from the supplied **Standard** > **Gases** folder.
3. Right-click the parent database node and select **Save**.
4. To use the Air mixture, select the **Air** folder in the Adding Mixture Components dialog box. All three components are selected together.

Database Materials

The database material node represents the substance in the material database. It is a manager node for database material properties nodes. The nodes of these materials have sub-nodes representing database material properties, such as dynamic viscosity and molecular weight. The properties nodes in turn contain objects that represent methods for calculating property values.

Database Material Node Properties

Symbol

The chemical name of the database material where applicable, such as `Al` for aluminum.

Title

The common name of the database material, such as `Aluminum (pure)`.

Database Material Node Pop-Up Menu

New Database Properties...

Activates a dialog that selects database properties from a list.

Database Material Property Nodes

The database material property node represents a particular property of the substance in the material database, such as **Density** or **Specific Heat**.

Database Material Property Method Nodes

Each property is associated with one or more of the following property methods:

Constant

Most properties have a Constant method available, which defines the constant value of the property.

To change this property, select the **Constant** sub-node and make an entry in the *Value* property.

The value node inside the database material property node represents the value that is set for that property.

Polynomial

Polynomial in T is a method for calculating certain quantities as polynomial functions of temperature. The **Polynomial** node represents polynomial coefficients for density, dynamic viscosity, electrical conductivity, saturation pressure, specific heat, and thermal conductivity. This particular type of polynomial method is not available in the supplied material database by default, instead you can create it. See Creating Material Databases.

Thermodynamic Polynomial Data

Thermodynamic Polynomial Data is a method for calculating specific heat with a piecewise, fourth-order (five-coefficient) polynomial in temperature. Two additional coefficients are provided in each temperature range to allow the calculation of enthalpy and entropy.

Table(T)

A material property that is imported from an XML file uses the **Table(T)** property method. The data for the material property is contained in a property table, **Table(T)** > **Material Database Table** node.

To replace the contents of a material database table:

- Right-click the **Material Database Table** node and select **Import Material Table from a file...**
- Open the `.csv` file containing the new data.

Table(T,c)

The Table(T,c) material property method option is only available when the material is selected from the Electrolytes Material Database. However, if you select the Table(T,c) material property and then replace the material with a non-electrolyte, the Table(T,c) method remains available—although the table values are no longer predefined.

This option specifies the Values for the material property at various Temperatures (K) and Li/Ion salt concentrations (mol/l). These values are predefined and stored in the electrolytes material database:

Values > **Material Databases** > **Electrolytes** > **Liquids** > **[electrolyte]** > **[material property]** > **Table(T,c)** > **Material Database Table**.

Material Database Table Properties

Interpolated value

Property value that is interpolated from the database table. (Read-only)

Extracted

Properties that are contained in the database table. (Read-only)

Material Database Table Pop-Up Menu

Export...

Export the material table to a `.csv` file.

Import Material Table from a file...

Replace the material data with the contents of a `.csv` file.

Tabulate...

Display the data in tabular form.

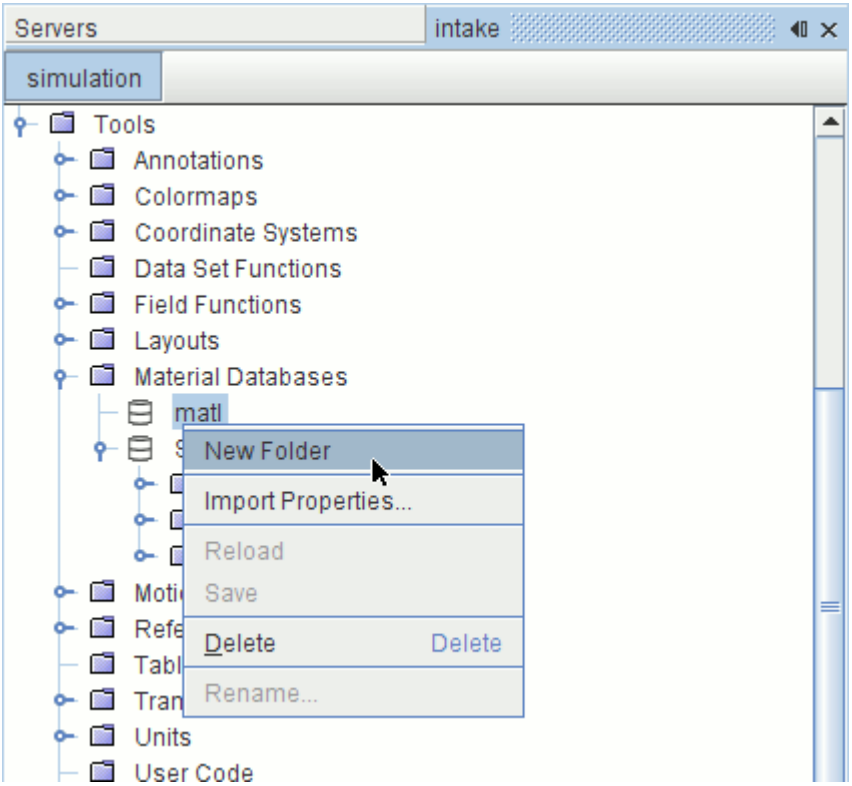
Contents:

Creating Material Databases
Using Material Databases

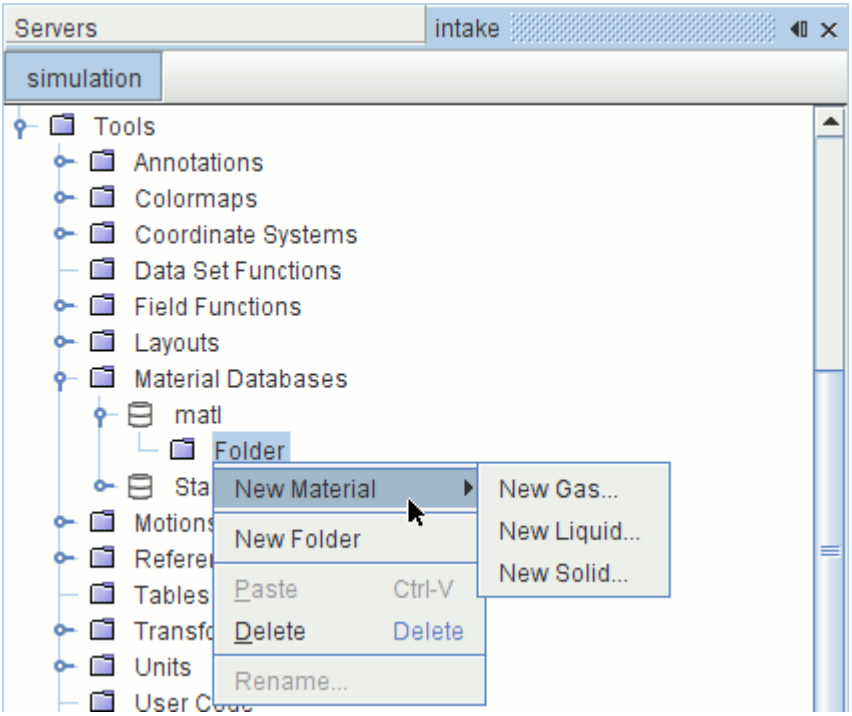
Creating Material Databases

These instructions describe how to set up a material and its properties in a new material database object.

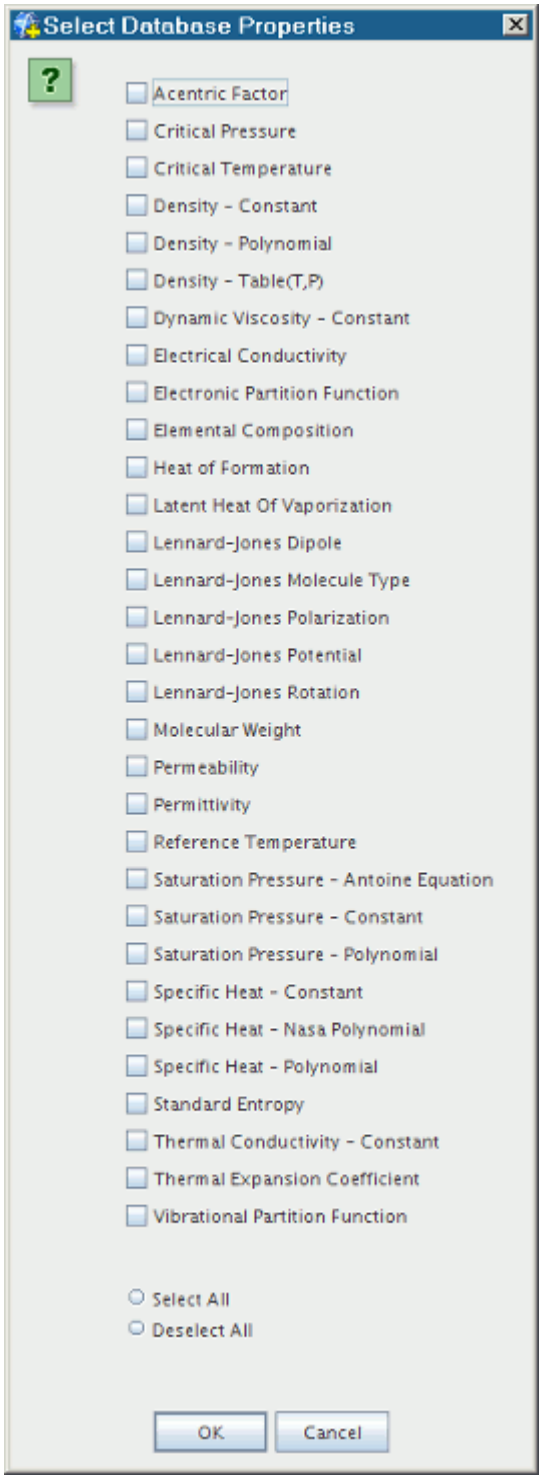
1. Right-click the material database node and select **New Folder**.



2. Right-click on the database folder node and select a new gas, liquid, or solid from the **New Material** pop-up menu item.



3. Right-click on the database material node and select **New Database Properties...**
4. Select database properties with corresponding methods from the list in the *Select Database Properties* dialog that appears.

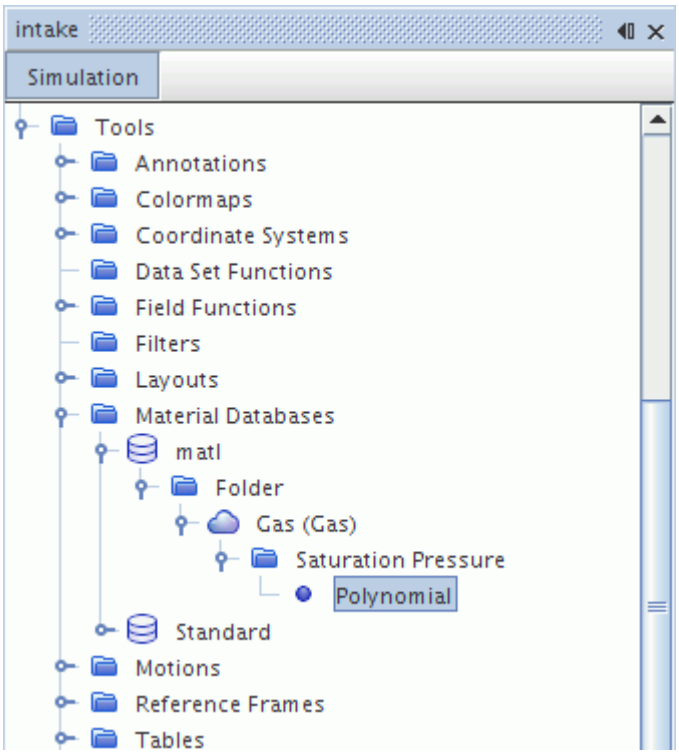


The constant values for these new database properties are zero by default.

The **Select All** button lets you manage the selection of many items efficiently. Simply click it to select the entire group, and then deactivate the individual checkboxes of the ones you wish to exclude. If you want to start over, deactivate all the checkboxes at once by clicking the **Deselect All** button.

5. Click **OK** when finished.

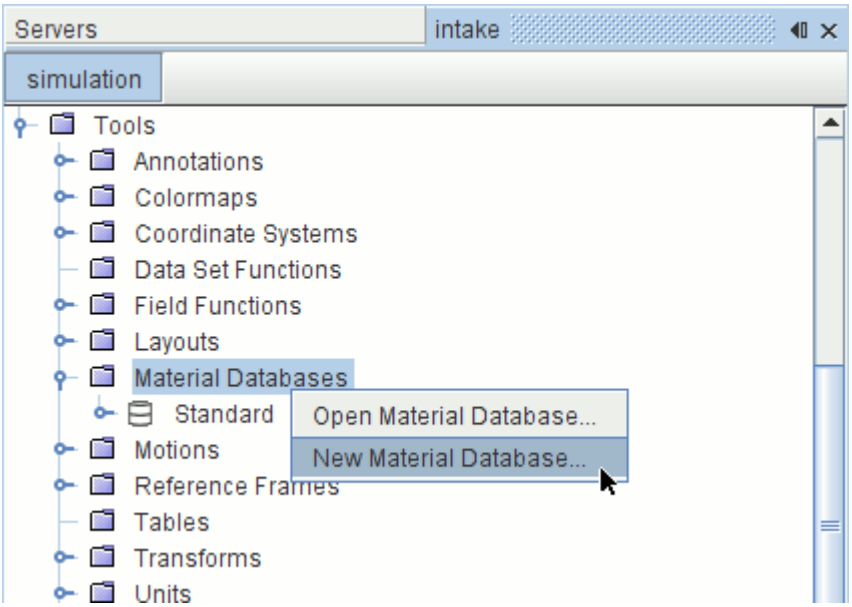
When a material property option is selected in the dialog, the option disappears from the dialog and a corresponding node appears in the object tree. In the example below, the option **Saturation Pressure - Polynomial** was selected. The resulting additions to the object tree are nodes representing the Saturation Pressure material property with the Polynomial in T method.



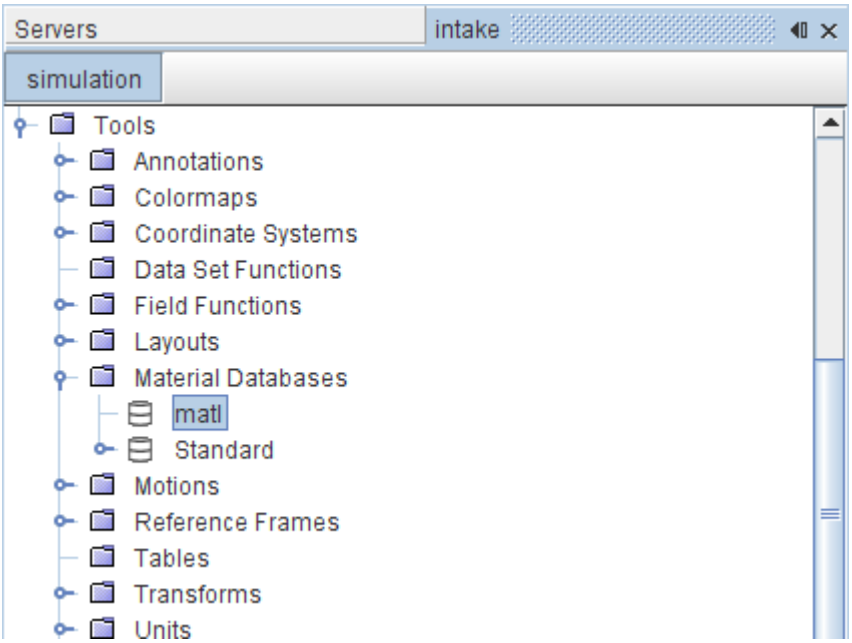
Adding Database Objects

You can add material database objects to the STAR-CCM+ simulation manually. This capability can be helpful for importing legacy (`.dbs`) data.

1. To create a material database object, right-click on the **Material Databases** manager node and select **New Material Database...**



A new, empty node appears with the default name **matl**.



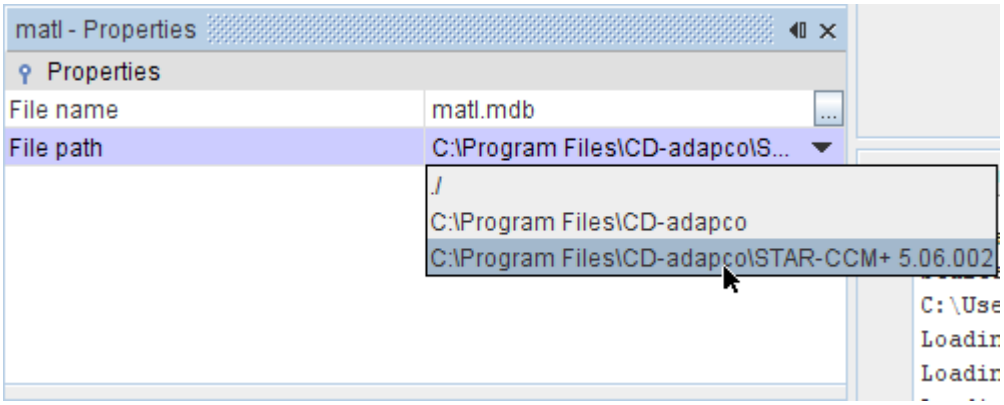
Subsequent objects have default names that are concatenated with a number, and can be renamed.

2. To populate this database from within the GUI, create folders. Right-click on the database node and select **New Folder**. You can then rename the folders as required to organize your database materials.
3. Once you have prepared the folders, right-click each one to select the database material to create. The database material objects have default names that are based on the type that is selected from the pop-up submenu, concatenated with numbers. Rename these objects according to the database materials you want them to represent. Make entries in both the *Symbol* and *Title* properties, though your choice of symbol entry is the more important.

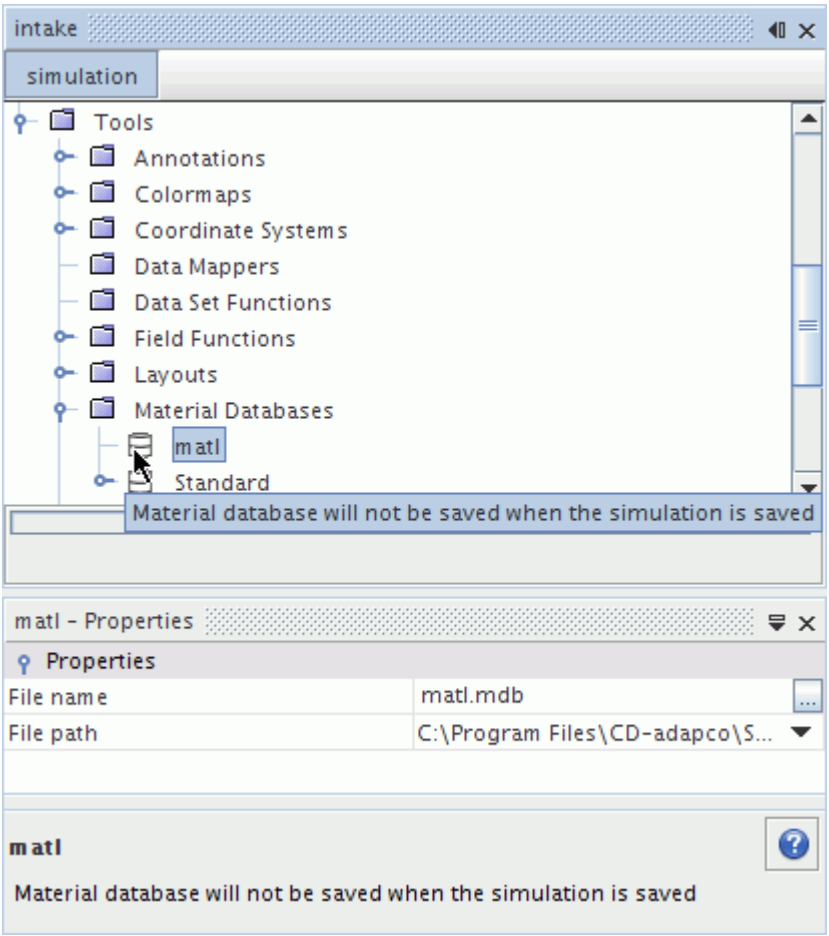
4. Next, open the database material nodes, select the database material property nodes and enter values for the properties. If you are setting up multiple database materials that have similar properties, you can use the copy-and paste technique as follows:
   a. Create one of the database materials and set up all its property values.
   b. Right-click on the node of the database material and select **Copy**.
   c. Create another node for the similar database material.
   d. Right-click that newer node and select **Paste**. It has all the properties and subobjects of the previous database material.
   e. Adjust the properties of that newer database material as needed.

   Using the *File path* property of the database object, you can select the file path of your new database file from among the following locations:

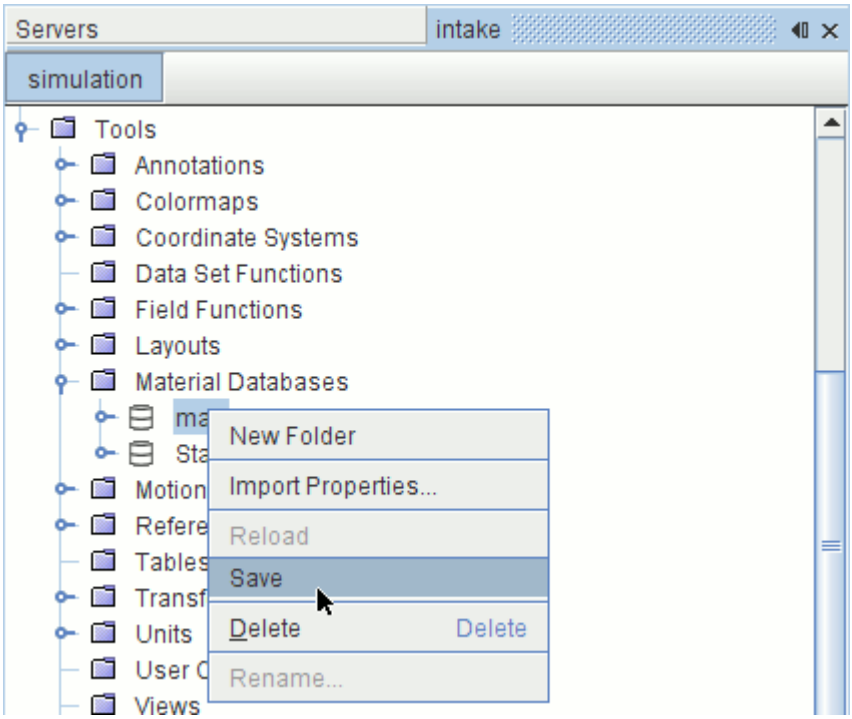
   - The root directory
   - The STAR-CCM+ installation directory
   - The same location as the current simulation file
   -



   However, the database you create is not saved until you explicitly export it to an external file. STAR-CCM+ reminds you of this fact in the tooltip and *Properties* window description area for every database object, whether the database it represents is saved or not.



To export your new database into an external file, simply right-click on the node and select **Save**.



The file is saved in `.mdb` format with a default name matching the name of the corresponding node.

Using Material Databases

You can load additional material databases and edit existing ones.

Loading Material Database Files

By default, a material database object named **Standard** is loaded automatically in a new simulation once you select a material model. This object accesses the `props.mdb` file that is located in your main STAR-CCM+ installation directory (Windows) or in the star subdirectory of that directory (Linux).

To add a material or change the property of one, you can load a separate material database file. Alternatively, you can modify a database file for this purpose and reload it into the simulation. (If you are working with the database file supplied with STAR-CCM+, be sure to make a copy and then move the original to another location.) You can also load legacy material data from `.dbs` files.

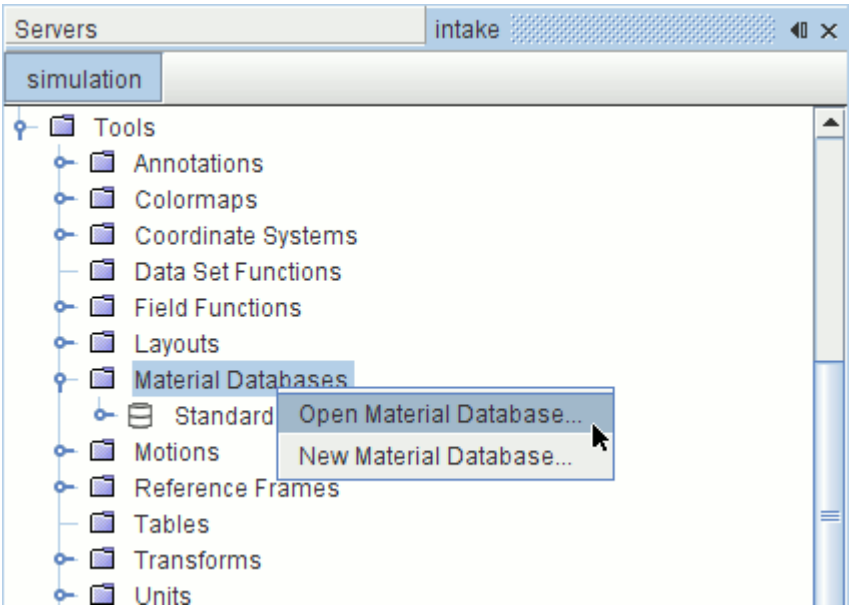
STAR-CCM+ can accommodate multiple material databases, making them all available in the material model selection process.

The simplest way to prepare to load an additional database file is to place it in the same directory as the ones supplied by default, but with a different name. This location would be your main STAR-CCM+ installation directory (Windows) or the `star` subdirectory of that directory (Linux). Make these preparations before opening your simulation file.

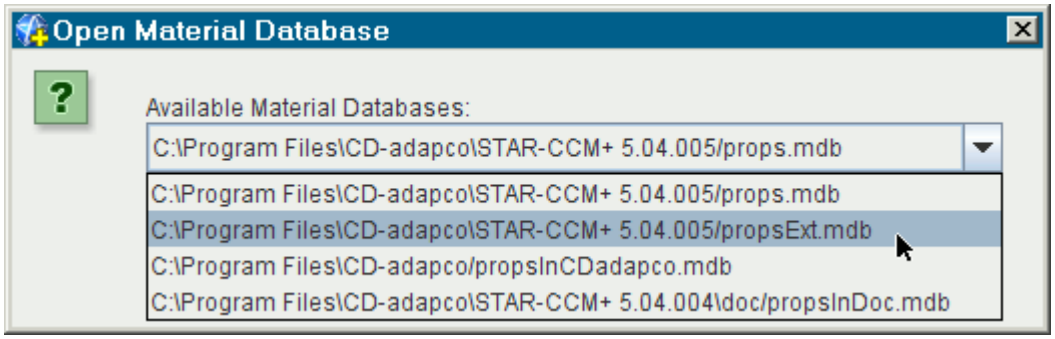
Alternatively, you can set an environment variable:

1. Select **Tools** > **Options** and, in the *Options* dialog, select the **Environment** node.
2. Select the *Material Database Load Path* property.

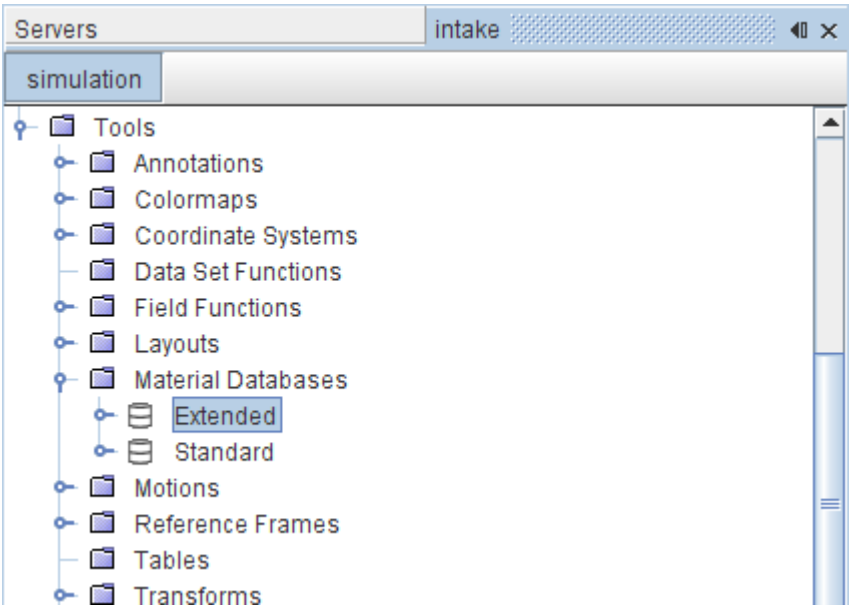
3. Click the property customizer ( ) open a dialog that provides a complete view of your entries. These paths are in addition to the default path. The dialog allows you to enter absolute paths of multiple databases, with each entry separated by a semicolon. Click **OK** when finished.
4. Next, right-click on the **Material Databases** manager node and select **Open Material Database...**



A separate *Open Material Database* dialog appears with a list of the `.mdb` files available, including the one(s) you added.



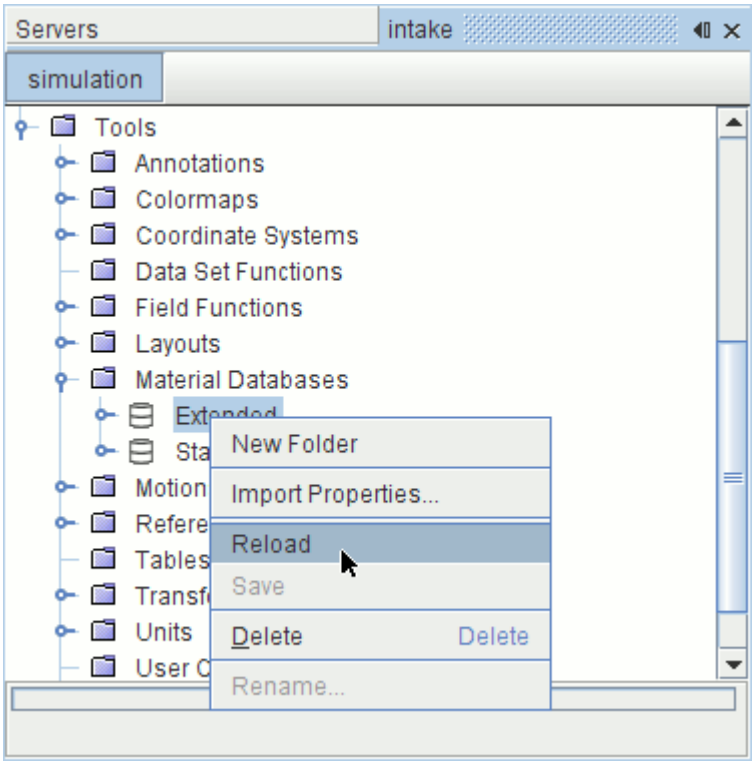
5. Select one and click **OK**. A new node for that database appears in the object tree, and its database file path appears in the *Properties* window.



Reloading a Database File

This action updates all the information in the simulation for that database.

To reload the material database file, right-click the material database node and select **Reload**.

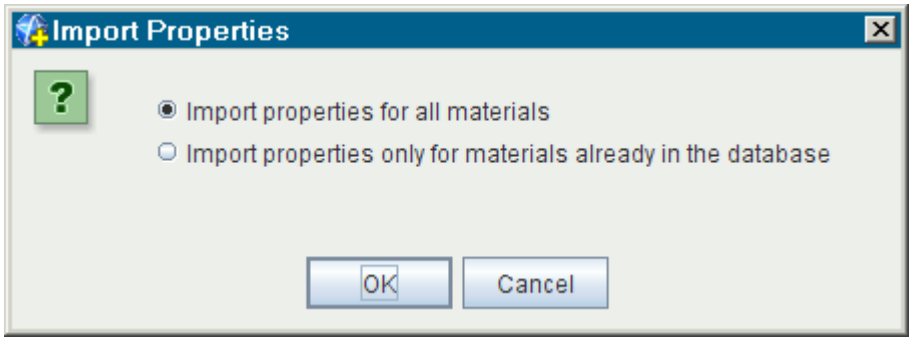


Loading a Legacy Database File

STAR-CCM+ can import the older `.dbs` database format.

To import this type of file: Then select an option from the *Import Properties* dialog.

1. Create a new (empty) material database object.
2. Right-click on that object and select **Import Properties...**The dialog that appears is set to filter the `.dbs` format.



3. Navigate to your file and click **Open**.

The first option imports the contents of the entire file. The second imports properties only for the materials that are already represented in the database object. This action populates the material database node with three database folders, similar to the structure of the **Standard** database object.

You can also add more properties to the database object by selecting **Import Properties...** again, and selecting other files.

Database files that are supplied with STAR-CCM+ include `props.mdb`, `thermdat.dbs` (NASA polynomials), and `moldat.dbs` (transport properties). To combine these files with one such as `props.mdb`, repeat the import action, but select the option **Import properties only for materials already in the database** in the dialog that appears.

Changing Database Material Properties

Generally material properties can be changed by opening the database material property nodes and entering values in the properties of the sub-nodes. However, your actions in the GUI affect how permanent these changes are.

In principle, changes are made permanently to the material database by modifying a copy of that database. (Take care not to overwrite the original supplied file.) The simulation file itself only retains changes that are made to properties of the database materials that are used to create substances in the continuum (within single, multiphase, or multi-component material models). Therefore, if you modify a group of database materials, and then select only part of them in your continuum, the other modified database materials revert to their original values when you close and reopen the simulation file.

A good approach for modifying properties is as follows:

1. For a trial run, change material properties in the continuum.
2. Run the simulation.
3. If you are satisfied with the results from the new property values of the materials in the continuum:
   a. Make a copy of the external database file for use by the simulation.

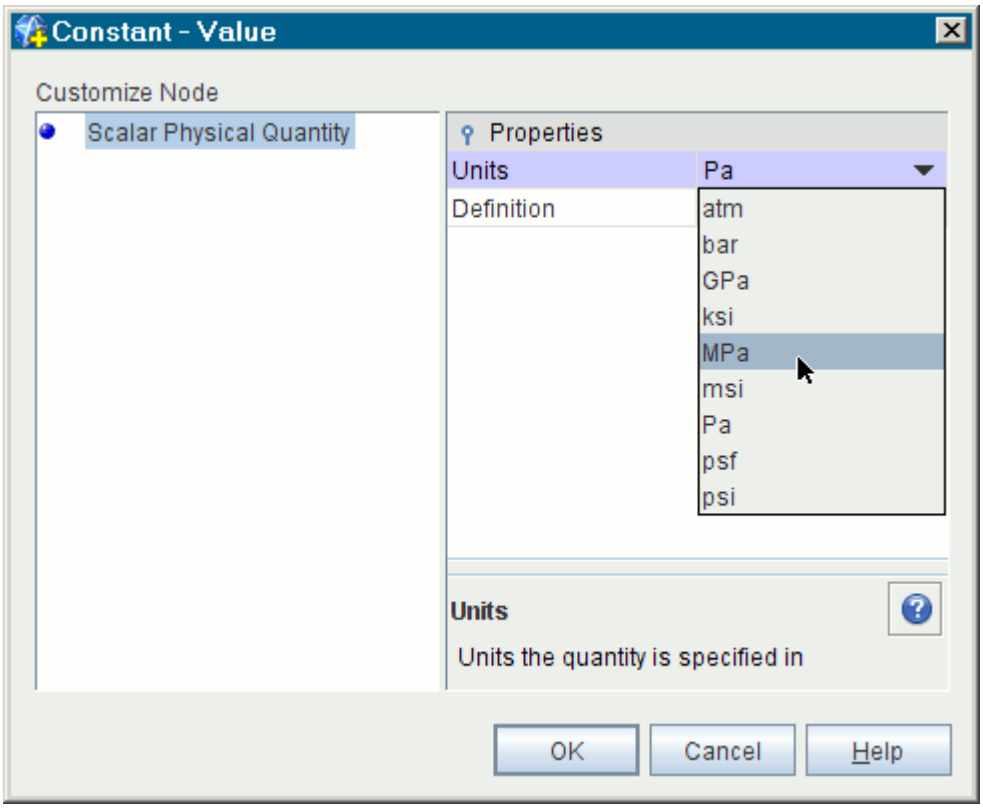
b. In the database material, enter the custom values that you used in the continuum. Specifically, enter the values in the database material properties values of the database material (for example **Tools** > **Material Databases** > **Standard** > **Gases** > **Air** > **Density** > **Constant**).

c. Save the values to the database file by right-clicking the material database node and selecting **Save**.

Changing the Units of the Database Material Properties

By default the database materials have properties that are measured in SI (Systeme Internationale) units. However, you can change the units as they are represented in the database material properties, using the standard technique for changing units in STAR-CCM+.

After making the switch once, you will have the added option of changing the units used for individual database material property values. To access this option for a particular value, click the property customizer (...) in the *Value* property of the node to activate the Customizer dialog. The properties within that dialog include a drop-down list with a choice of units.



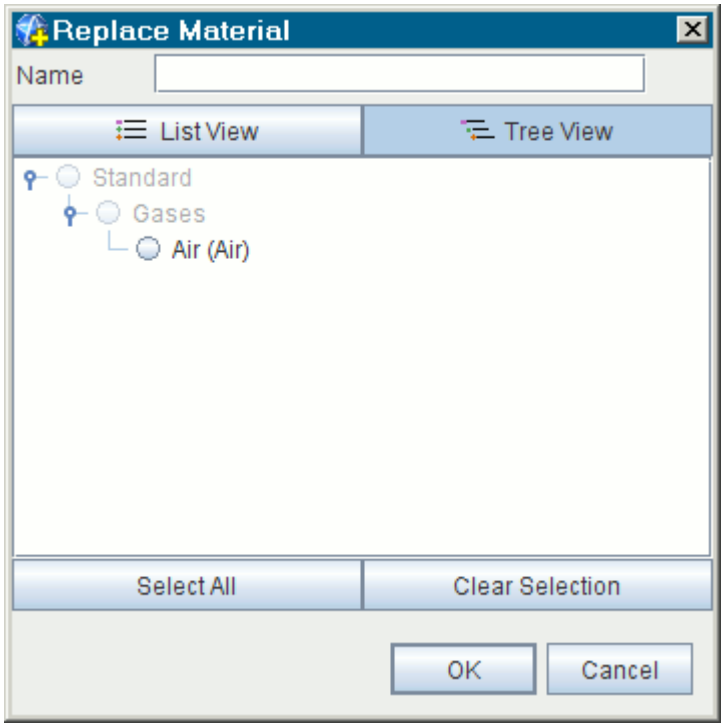
What If the Database File Is Missing?

Most of the work you would do with material models depends on the supplied `props.mdb` file residing in its normal place. That place is your main STAR-CCM+ installation directory (Windows) or the `star` subdirectory of that directory (Linux). You can load a material database into an open simulation.

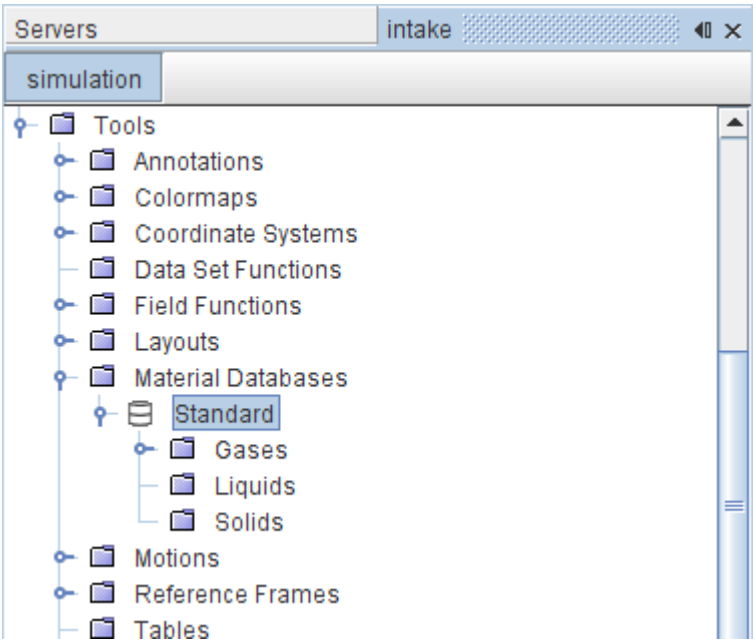
Without a database, you cannot select materials in a new simulation, or change the material selections in an existing simulation. Your only option is to modify properties of material selections that were saved previously in a simulation file.

In a new or existing simulation, STAR-CCM+ lets you know in several ways that the database is missing (in the following example, the standard material database is missing):

- When you attempt to select a new or alternate material model, you get an empty selection dialog, except for any materials selected in a previously saved simulation. (In this example, the simulation file contains only the material **Air**.)



- After you attempt to select a material model, the material database node appears in the object tree, but with no database folder nodes. (In an existing simulation file, the database folder nodes are also empty except for the referenced materials.)



Modifying a Copy of the Material Database

Located in your main STAR-CCM+ installation directory (Windows) or in the star subdirectory of that directory (Linux), the `props.mdb` file is modified through the STAR-CCM+ GUI, rather than opening it with another program.

Possible reasons for changing the material database include:

- Adding material data
- Importing a PPDF table (equilibrium or flamelet)
- Importing Chemkin-formatted data

Caution: Before changing the material database, make a copy of the database file for modification, and keep the original in a safe place. An additional helpful measure is to set the original file as read-only.

After you make a copy of the external `props.mdb` file and relocate the original, change properties of database materials and/or add new materials with new properties and enter values for them. Once you are finished, right-click the node of the material database and select **Save**. (Saving the simulation file alone does not save your database material properties changes to the external database file.)

Modeling Flow and Energy

Many engineering design projects require you to predict the effect of flowing fluids on containing structures or immersed objects. While you can analyse simple scenarios with hand calculations, complex scenarios require you to apply numerical methods for accurate solutions.

In STAR-CCM+, you can model the following types of flow:

Inviscid Flow

Inviscid flows are an idealization resulting from neglecting the viscous effects in simulating the equations of motion.

The solution of the resulting Euler equations (as opposed to the Navier-Stokes equations) generally results in significant savings of computer resources. Boundary layers and other viscous effects are not resolved. This approximation is only valid for certain physical situations, such as high-Reynolds number compressible aerodynamics.

Viscous Flow

Viscous flows are typically of high viscosity and can be classified as either laminar or turbulent. Laminar and turbulent flows occur in nature. Both types are described using the Navier-Stokes equations which include the effects of viscosity, thermal conductivity, and mass diffusion. For a detailed treatment of laminar viscous flow, see Modeling Viscous Flow.

Laminar Flow

The term laminar refers to a well-ordered flow, free of macroscopic, non-repeating fluctuations. Laminar flows occur in nature when the Reynolds number (the ratio of viscous to inertial forces) is low enough that transition to turbulence does not occur.

In computational simulations, numerical instabilities can arise from simulating laminar flows at Reynolds numbers that are too large. If you are seeking a steady solution, these instabilities can impede convergence. Therefore, laminar flow simulation is appropriate if you already know that the Reynolds number of the problem is sufficiently low.

Transitional Flow

Transition is a term that refers to the breakdown of laminar flow, through amplification of infinitesimal disturbances, to turbulence. A transitional flow can be defined as one that encompasses this process.

The occurrence of physical instabilities in laminar simulations cannot be relied on as an accurate indication of transition. Furthermore, when using turbulence models in low-Reynolds number simulations, the onset of turbulence in viscous layers cannot be predicted with any reliability by the turbulence model itself.

Strictly speaking, STAR-CCM+ has no means of predicting transition. However, a Transition model is available that allows you to mimic the effect of transition by suppressing the turbulence in a certain pre-defined region.

Turbulent Flow

A flow that is in a state of continuous instability, exhibiting irregular, small-scale, high-frequency fluctuations in both space and time is termed turbulent.

It is strictly possible to simulate turbulent flow directly by resolving all the scales of the flow (termed direct numerical simulation). However, the computer resources that are required are too large for practical flow simulations. Therefore, a suitable turbulence modeling approach must be selected.

As fluids appear in a wide range of scenarios, and exhibit different behaviors depending on their properties, STAR-CCM+ provides several flow models so that you can address most applications. These models are summarized as follows:

Segregated Flow and Energy

The Segregated Flow model is an alternative for incompressible or mildly compressible flows, particularly when computational resources are an issue; it is used with the Segregated Fluid Energy models.

Coupled Flow and Energy

The Coupled Flow and Coupled Energy models are best for compressible flows, natural convection problems, and flows with large body forces or energy sources. The Coupled Flow model includes the optional AUSM+ inviscid flux scheme, which is advantageous for flows involving high supersonic and/or hypersonic regimes.

Viscous Flow and Energy

The Viscous Flow model is intended for viscoelastic materials and other highly viscous non-Newtonian fluids, and is the prerequisite for all viscous flow simulations. The Viscous Energy model simulates energy transfer due to internal forces associated with complex deformations of viscous fluids. For further information, see Viscous Flow Models.

Harmonic Balance

For complex time-periodic flows.

Whichever flow and energy models you choose, modeling options that interact with them include:

- Equation of state—models that compute the density and the density derivatives with respect to temperature and pressure
- Passive scalar—transport equations for user-defined variables that can be represented using arbitrary scalar values
- Gravity—specifically including the buoyancy source terms in the momentum equations

Additional models accompany the flow and energy ones to create simulations for specific applications:

- Thin film—deicing or defogging on a surface such as a cabin windshield
- Porous media—commonly used to show the macroscopic effect of the porous medium on the overall fluid flow
- Non-reflecting boundaries and mixing-plane interfaces—allowing accurate simulations of applications such as multi-stage turbomachinery
- Vorticity confinement—preventing the rapid dissipation of vortices
- Fans—specifically the fan momentum source

In addition to descriptions of the models that are mentioned above, the following sections help clarify how this modeling works:

- Explanation of how pressure is simulated in STAR-CCM+
- Suggestions for initializing the flow
- Formulation of the boundary conditions

When you activate these models, other objects appear in the GUI that are designed to give you added control over the flow modeling. Some of them are common to more than one flow or energy model, or common to flow and energy models. They are listed in the following sections:

- Flow reference
- Energy reference
- Field functions common to flow and energy models

Contents:

Flow and Energy Workflow
High-Speed Flow Regimes
Pressure
Modeling the Equation of State
Modeling Flow and Energy Using a Coupled Approach
Coupled Flow Controls and General Setup Recommendations
Segregated Flow Guidelines
Segregated Flow Model Reference
Initializing Flow
Initial Conditions
Flow Boundaries Reference
Partial Slip
Boundary Conditions Formulation
Reporting Mass Flowrate
Modeling Passive Scalars
Modeling Gravity
Modeling Axisymmetric Swirl
Modeling Fully Developed Energy
Confining Vorticity
Vorticity Confinement Formulation
Region Values
Flow Reference
Fluid Flow and Energy Field Functions Reference
Fluid Flow Formulation

See Also:

Real Gas Models
What Is the AUSM+ Scheme?

Modeling Fluid Energy Using a Segregated Approach
What Is the Harmonic Balance Flow and Energy Model?

Flow and Energy Workflow

STAR-CCM+ provides multiple flow solvers that address different categories of fluid flow. These solvers include the viscous flow solver, the segregated flow solver, and the coupled flow solver. Choosing the correct solver is one of the first steps in a flow analysis.

To model fluid flow in STAR-CCM+:

1. Assess the flow that you see in the real situation, and extract the quantities that define the flow, particularly the Reynolds number.

   For high Reynolds number flows, consider the additional information that is given in High-Speed Flow Regimes.
2. Design the simulation topology for the flow scenario you intend to model.
   - If you want to include porous media, or sections that rotate or translate, design the simulation topology so that it has distinct regions for these physics modifiers.
   - Set boundary types that support the flow conditions you see in the real situation.
3. Determine which category of flow you intend to simulate, and decide which solver is most appropriate. For external flows of Newtonian fluids, use the following table:

External Flow Categories

Flow Category	Range of Conditions	Suggested Solver
Incompressible flow	Ma $\leq$ 0.2	Segregated Flow is normally preferred, but Coupled Flow can work well if computational resources are not an issue.
Compressible subsonic flow	0.2 < Ma < 0.7	Coupled Flow is preferred, but Segregated Flow sometimes works.
Transonic flow	0.7 < Ma < 1.2	Coupled Flow
Supersonic flow	Ma $\geq$ 1.2	
Hypersonic flow	Ma $\geq$ 5	

For Non-Newtonian flows of substances such as molten rubber or liquid polystyrene, use the Viscous Flow model.

In addition to compressible flows, choose the Coupled Flow and Coupled Energy models for natural convection problems and flows with large body forces or energy sources.

Some general observations about the segregated and coupled flow solvers are:

- The segregated solver uses less memory than the coupled.
- The coupled solver yields more robust and accurate solutions in compressible flow, particularly in the presence of shocks.
- The coupled solver is more robust for high-Rayleigh number natural convection.

- The number of iterations that the coupled solver requires to solve a given flow problem is independent of mesh size. However, the number of iterations that the segregated algorithm requires increases with mesh size.
- In some situations, the solver algorithm can be combined with the implicit solver to permit large CFL numbers. This scenario would be analogous to an under-relaxation factor of 1 for all variables in a segregated algorithm. In contrast, the segregated algorithm needs significant under-relaxation for both velocity and pressure and, in compressible flows, energy.

4. Prepare a mesh that matches the requirements of the solver, and that follows guidelines for the flow scenario you are working with.
   - The segregated and coupled flow solvers accept any two or three-dimensional mesh that the STAR-CCM+ meshers can generate.
   - The viscous flow solver can only accept meshes that come from the following meshers:
     - Tetrahedral
     - Extruder
     - Directed Mesh

     In general, the viscous flow solver cannot accept any mesh that contains polyhedral cells.
5. For each physics continuum that contains a fluid material, select one model from each of the following group boxes:
   a) *Space*: choose any model apart from **Shell Three Dimensional**. Only choose **Two Dimensional** or **Axisymmetric** if the mesh has zero-thickness and lies along a plane.
   b) *Time*: choose a compatible option as described in the following table:

Time Model	Compatible Flow Solvers and Fluids	Analysis Type
Steady	All flow solvers, all materials	Steady-state analysis
Implicit Unsteady	All flow solvers, all materials	Transient analysis with several control options
PISO Unsteady	Segregated Flow only, not Eulerian Multiphase	Transient analysis with short time-steps
Explicit Unsteady	Coupled Flow only, Gas or Multi-Component Gas	Transient analysis of high-speed events at low computational cost
Harmonic Balance	Specialized solver, only for Gas.	Specialized form of steady-state analysis

   For steady simulations, the coupled solver in STAR-CCM+ employs a time marching scheme to drive the unsteady form of the governing equations to a steady state. In this case, a pseudo-transient term replaces the physical time-derivative. The solution advances in pseudo-time to drive this term to zero.

   c) *Material*: choose either **Gas** or **Liquid**, or the equivalent multi-component versions. A multi-component fluid is composed of a number of individual species that contribute to the properties of the fluid.
      - The viscous flow solver only supports liquids.

d) *Flow*: choose the solver strategy.

Options are:

- **Coupled Flow**
- **Segregated Flow**
- **Viscous Flow**

If you are using the Harmonic Balance approach, STAR-CCM+ automatically selects the **Harmonic Balance Flow and Energy** model.

e) *Viscous Regime* (or *HB Viscous Regime* if using Harmonic Balance): choose from **Inviscid**, **Laminar**, or **Turbulent**.

For Viscous Flow, **Laminar** is the only option.

f) *Equation of State*: choose the model that best represents the behavior you are modeling. The choices are:

Model	Suitable For...
Constant Density	Incompressible gas or liquid flows. This option is the only option available for Viscous Flow.
IAPWS-IF97 (Water) [Segregated Flow Only]	Liquid water only
Ideal Gas	Single or multi-component gases
Polynomial Density	Liquids or gases
Real Gas	Single or multi-component gases. Requires further selection from: Peng-Robinson, Redlich-Kwong, Soave-Redlich-Kwon, Modified Soave-Redlich-Kwong, Van der Waals, Equilibrium Air, or IAPWS-IF97 (Steam).
Thermal Non-Equilibrium Ideal Gas [Coupled Flow Only]	High temperature and low densities (for example, hypersonic flows)
User Defined EOS	Implementing an equation of state using field functions or user code

6. If you want to include thermal effects in the simulation, activate one of the energy models.
   - When an equation of state model requires that you solve the energy equation, the *Energy* group box appears.
   - When solving the energy equation is optional, the energy models appear within the *Optional Models* group box.

   The models that you choose from are:
   - Coupled flow solver: **Coupled Energy**
   - Segregated flow solver: **Segregated Fluid Enthalpy**, **Segregated Fluid Isothermal**, or **Segregated Fluid Temperature**
   - Viscous flow solver: **Viscous Energy**
7. If you choose the Turbulent Viscous Regime, make further selections starting from the *Turbulence* group box. See Selecting a Turbulence Model.

8. To include the effects of gravity, from the *Optional Models* group box, activate **Gravity**.

   For constant density flows, you can additionally choose to activate the **Boussinesq Model**. See Modeling Gravity.

9. Choose any additional models that you require for the physics continuum, then close the dialog.
10. Review flow model properties on the model node. Your scenario could require different settings from the defaults.

    For model reference information, see the following:

Flow Model	Property Reference
Coupled	Coupled Flow Model Reference
Segregated	Segregated Flow Model Reference
Viscous	Viscous Flow Model

11. Expand the **[physics continuum]** > **Models** > **[Initial Values]** node and set initial values of pressure and velocity.

    For inviscid problems, the initial velocity must be non-zero. Zero initial velocity is also inappropriate if the Turbulence Intensity method is being used to specify initial turbulence values.

    Review the content within Initializing Flow for specific guidelines.

12. For improved convergence, set some or all of the following properties on the **Segregated Flow** solver node and its children, or on the **Coupled Implicit** solver node and its children:
    - Segregated flow solver:
      - Set *Under-Relaxation Factor* properties on the Velocity and Pressure Solvers
      - Set *Continuity Initialization*
      - Activate *Enabled Enhanced Stability Treatment*
    - Coupled flow solver:
      - Activate Grid-Sequencing initialization
      - Activate the Expert driver
      - Activate Convergence Accelerators, particularly when mass-balance convergence is slow.
13. To control under-relaxation or number of iterations for linear systems of equations, set the algebraic multigrid parameters for either the pressure solver or the velocity solver:
    a) Open the node for the pressure or velocity solver under the **Coupled Implicit** node in the **Solvers** node.
    b) Open the node for the **AMG Linear Solver**.
    c) Enter the algebraic multigrid parameters using the *Properties* window.

    Note: In most situations, it is not necessary to change these parameters.

14. When using the Implicit Unsteady time model for a transient analysis, set the discretization, physical time-step, and inner iterations criteria:
    a) In the simulation tree, select the **Solvers** > **Implicit Unsteady** node and set an appropriate *Time-Step*.
    b) Within the same node, choose the order of *Temporal Discretization*.

15. When using the PISO Unsteady time model for a transient analysis, set the physical time-step and verbosity level:
    a) In the simulation tree, select the **Solvers** > **PISO Unsteady** node and set an appropriate *Time-Step*.
    b) Within the same node, choose the level of *Verbosity*.
16. For any transient analysis, select the **Stopping Criteria** > **Maximum Inner Iterations** node and set *Maximum Inner Iterations*.

    As this value directly impacts the number of iterations that STAR-CCM+ performs for an analysis, you are advised to plan some numerical experiments for finding the minimum value that gives convergence within each time-step.
17. Within the **Regions** node, set boundary conditions for each flow and wall boundary.

    See Flow Boundaries Reference.

High-Speed Flow Regimes

At high Mach numbers, extreme temperatures occur and dissociation, or even ionization, of the gas must be considered. First establish that the continuum mechanics approach used by STAR-CCM+ is valid, then assess which other factors you must account for.

Flow problems can be divided into four general regimes that are based on the Knudsen number as shown below. These limits are not precise, but rather rules of thumb.

$Kn < 0.01$	Continuum (for which STAR-CCM+ is valid)
$0.01 < Kn < 0.1$	Slip flow
$0.1 < Kn < 3$	Transitional
$Kn > 3$	Free molecular flow

STAR-CCM+ can be used in the continuum regime where the Navier-Stokes equations are valid.

In the slip flow regime, the Navier-Stokes equations can produce good results. However, the effects of rarefaction must be taken into account through slip velocity boundary conditions and accounting for temperature jump on the surface.

Solving for transitional flows requires a statistical method, such as a direct simulation Monte Carlo approach. In the free molecular regime, the flow is sufficiently rarefied that molecular collisions can be neglected. The continuum assumption that is built into the Navier-Stokes equations is not applicable to these rarefied gas regimes.

For high Reynolds number, the Knudsen number is:

$$Kn = \frac{M}{Re}\sqrt{\frac{\gamma\pi}{2}} \tag{213}$$

where M is the Mach number, γ is the ratio of specific heats, and the Reynolds number Re is dependent on the characteristic length of the body, L.

Mach Number Regimes

For flows that STAR-CCM+ can support, the Mach number regimes, and the appropriate models to use, vary by gas and upstream temperature and pressure. For air however, the following general guidelines can be used:

Mach Number	Flow Regime	Modeling
$M < 3$	Calorically Perfect	Ideal gas Constant specific heat
$3 < M < 8$	Thermally Perfect	Ideal gas Specific heat using polynomial in T
$8 < M < 30$	Dissociation	Real gas model and/or chemically reacting flow

$M > 30$	Ionization	Requires solving the Navier-Stokes equations together with the Maxwell equations. Not inherently supported in STAR-CCM+

Pressure

STAR-CCM+ simulates various pressure quantities, depending on the model selection.

- Reference Pressure— a device that reduces the numerical roundoff error in the numerical calculations involving pressure.
- Working Pressure— the solution variable p that is used in the flow models in STAR-CCM+. In the code and in the documentation, this variable is mentioned most of the time as Pressure. Its definition is a matter of computational convenience and depends on the model selection. It is always expressed relative to the Reference Pressure.

  In laminar flows without the influence of gravity, Pressure is equal to the static gauge pressure. If the gravity model is active, Pressure is equal to the piezometric pressure, as shown in Eqn. (1571) for variable and constant density flows.

  When a turbulence model is used, Working Pressure includes the contribution of $\frac{2}{3}\rho k$. This contribution can be considered negligible in most flows of engineering interest.
- Static Pressure— the spherical part of the stress tensor acting in fluids (see Eqn. (1649)). It happens that the static pressure is the same as the thermodynamic pressure, which is one of the state variables.
- Absolute Pressure— static pressure + reference pressure.
- Relative Total Pressure— the static pressure that results from isentropically bringing the flow to rest in the relative frame of motion.
- Total Pressure— For compressible fluid, total pressure is calculated assuming an isentropic expansion from the total (stagnation) to the actual flow conditions.

  For incompressible fluid, total pressure is working pressure + dynamic pressure, $p + \frac{1}{2}\rho v^2$.
- Absolute Total Pressure— total pressure + reference pressure + $\rho_{ref} g(x_0 - x)$ whereρ_{ref} is the reference density, x is the altitude, and x_0 is the reference altitude.

Contents:

Defining the Reference Pressure
Providing a User Specified Pressure Reference Location
When Is the Pressure Reference Location Required?

See Also:

Modeling Gravity

Defining the Reference Pressure

The reference pressure is simply a device that is used to reduce the numerical roundoff error in the numerical calculations involving pressure.

This reduction is necessary since the differences in pressure are important, and these differences can be small relative to the absolute value of the pressure, as for example in fully incompressible or very low-Mach number flows. By subtracting a suitable constant reference pressure, a working pressure is obtained that is less prone to roundoff errors.

For simulations that use the Constant Density model or the Polynomial Density model, the actual value of the reference pressure has no relevance to the calculations. However, when using the Ideal Gas model, the reference pressure is used in the definition of the Absolute Pressure field function according to Eqn. (1466).

For simulations using the Ideal Gas model the following guidelines are suggested for setting the reference pressure:

- If the Mach numbers are low (say 0.3 or less), set the reference pressure to some ambient value such that the working pressures are small.
- If the Mach numbers are substantial, you can do one of the following:
  - Set the reference pressure to an ambient value such that the working pressures are small.
  - Set the reference pressure to zero so that working pressures are identical to the absolute pressures.

Specify the pressure location manually when conditions require it.

Reference Pressure Properties

Value	The value of the reference pressure. The customizer triggers the define value action.

See Also:

General Equation of State Models

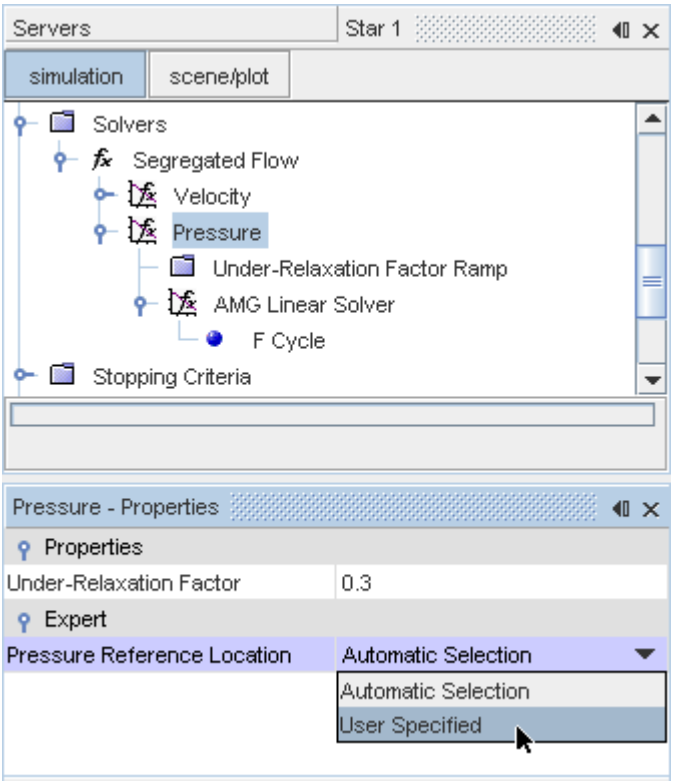
Providing a User Specified Pressure Reference Location

It is possible to set the pressure reference location manually when it becomes necessary to do so.

(See When is the Reference Pressure Location Required?).

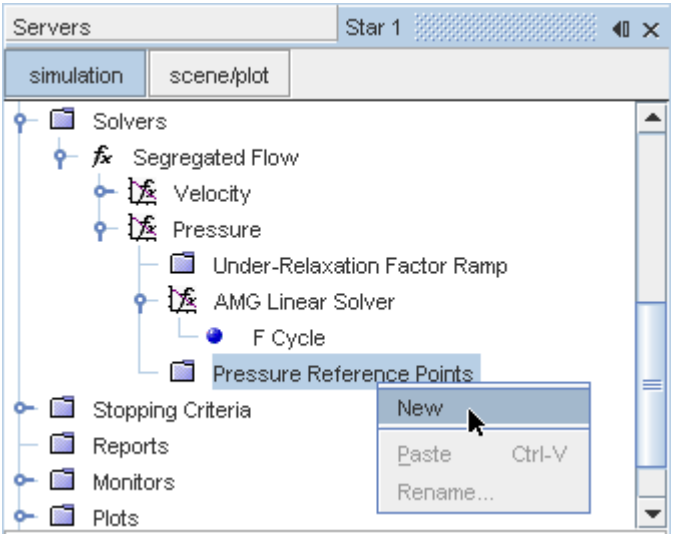
To set the pressure reference location:

1. Select the appropriate solver node and choose **User Specified** for the *Pressure Reference Location* property in the *Properties* window.

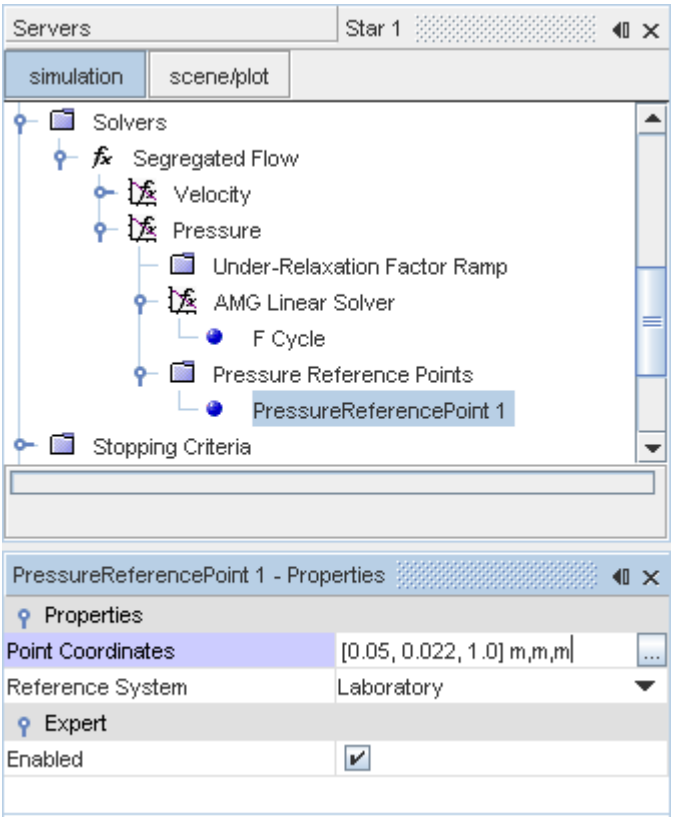


The **Pressure Reference Points** manager node, which has its own pop-up menu, is added as a child.

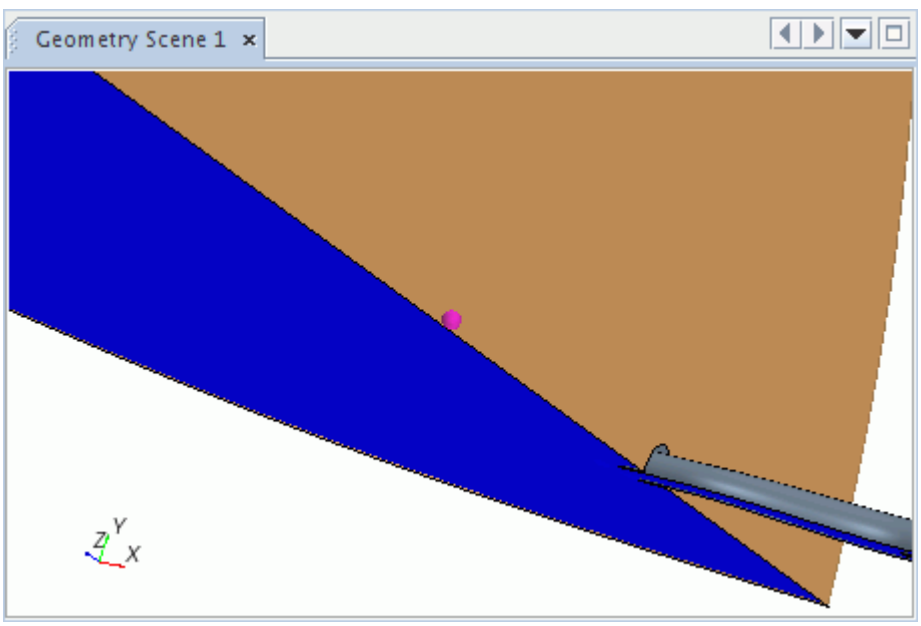
2. Right-click this node and choose **New**. A Pressure Reference Point node is added.



3. Select the newly added **Pressure Reference Point** node and set the point coordinates (along with the reference coordinate system) in the *Properties* window.



When the **Pressure Reference Point** node is selected, the point is displayed in an active scene.



What Is a Pressure Reference Point?

The properties of a *Pressure Reference Point* node are used to define a single geometric location that STAR-CCM+ uses in finding the pressure reference cell for a fluid stream. For each fluid stream, the cell closest to

one of the defined pressure reference points is used as the pressure reference cell. Multiple pressure reference points are only necessary for problems having multiple disconnected fluid streams; you set one pressure reference point in each fluid stream.

Properties

Point Coordinates	The geometric location of the pressure reference point in the selected coordinate system.
Reference System	A list of local coordinate systems in the simulation for use in defining the point coordinates.

Expert Properties

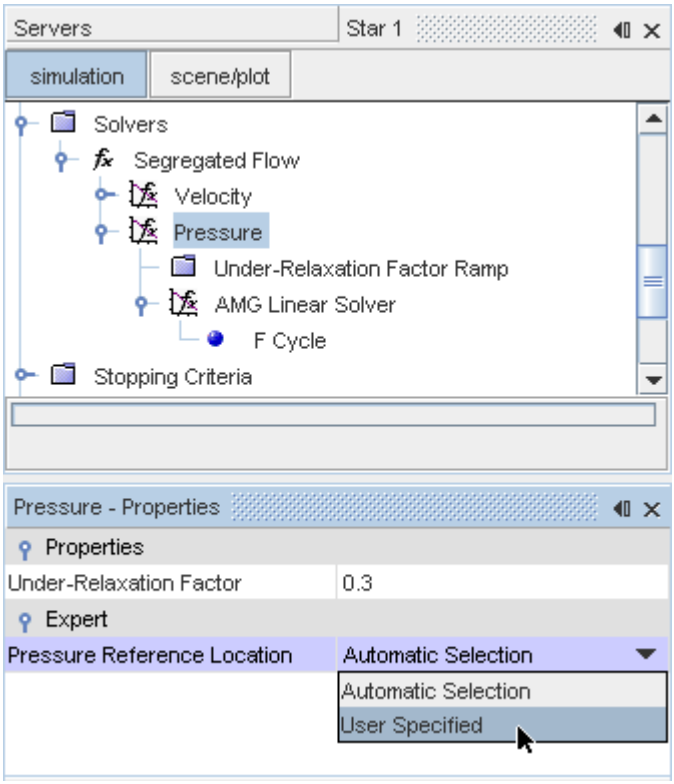
Enabled	Determines whether the solver users this pressure reference point in finding the reference cell.	
	Activated	The point is included.
	Deactivated	The point is excluded.

Pressure Reference Points Manager Pop-Up Menu

New	Adds a Pressure Reference Point node to the manager.

When Is the Pressure Reference Location Required?

The Pressure Reference Location is an expert property on the Pressure solver for the Segregated Flow model, and on the Coupled Implicit and Coupled Explicit solvers for the Coupled Flow model. It is used to determine whether STAR-CCM+ automatically calculates a location for the reference pressure, or whether you provide the location manually.



When simulating steady flow, or the flow of incompressible fluids, the equation for pressure is singular. If the boundary conditions do not define the pressure level (that is, there are no pressure boundaries), you can anchor the pressure level to a datum value in the solution domain. If the working pressure is not anchored to a known value somewhere in the domain, it can float arbitrarily high and cause numerical problems.

- If absolute pressure levels are not important, set the *Pressure Reference Location* property to **Automatic Selection**.

  STAR-CCM+ identifies the cell such that the cell centroid has the min(x), min(y), min(z) in the domain. This condition can only occur in cells that are next to boundaries.

  To determine the appropriate cell, STAR-CCM+ does the following:

  1. Checks the centroid of every cell that is next to a boundary, and selects all of the cells that have the smallest X component.

     STAR-CCM+ checks all of the boundaries: wall boundaries as well as flow boundaries.
  2. From the cells that are selected in the previous step, STAR-CCM+ selects the cells that have the smallest Y component.
  3. From the cells that are selected in the previous step, STAR-CCM+ selects the cell that has the smallest Z component.

  STAR-CCM+ uses the selected cell as the pressure reference location.

  Note: There is known limitation with this method. Consider a rectangular prism as a flow domain: in the ideal case STAR-CCM+ would select the cell at the lower left front edge of the prism. However, small numerical rounding errors in the centroid calculation and meshing can cause STAR-CCM+ to select a different cell nearer the middle of the left side of the prism.

- If you want to determine absolute pressure levels accurately, set the pressure reference location manually as described in the section Providing a User Specified Pressure Reference Location. For example, when you want to evaluate the pressure force on wall boundaries. Choose a location where the pressure is known, and where a fixed value is appropriate for the simulation.

The working pressure at the pressure reference location is set using the initial pressure value; the absolute pressure is the sum of the reference pressure plus the initial pressure. If the initial pressure value is changed during a simulation (by using a field function, for example), the pressure at the pressure reference location changes accordingly.

In buoyant flows, the definition of an operating altitude is required. The operating altitude is a position vector defining the location at which the static and piezometric pressures are equal. For more information, see the sections on modeling gravity for variable density and constant density flows.

See Also:

Coupled Flow Solvers

Segregated Flow Solvers

Modeling the Equation of State

The Equation of State model is used to compute the density and the density derivatives with respect to temperature and pressure.

This section describes the various models that are available in STAR-CCM+:

Contents:

General Equation of State Models
Real Gas Models
Entropy Reference
Equation of State Field Functions Reference

See Also:

Coupled Flow Solvers

General Equation of State Models

General equation of state models compute the density and density derivatives with respect to temperature and pressure.

STAR-CCM+ uses the following models:

- Constant Density, based on the assumption that density is invariant throughout the continuum. It is available for gases, liquids, and solids.
- Ideal Gas, available for gases only. It uses the ideal gas law to express density as a function of temperature and pressure.
- IAPWS-IF97 (Water), available for water only. For more information on this model and the companion model for steam, see The IAPWS-IF97 Models under Real Gas Models.
- Polynomial Density, available for gases, liquids, and solids. It works on the assumption that the density is a function of temperature only.
- Real Gas, a family of models for situations outside the scope of the idea gas law. See also Real Gas Models
- Thermal Non-Equilibrium Ideal Gas, for use at high temperatures and low densities, where the vibrational/electronic energy modes become active yet the density is low enough that equilibration does not occur. See also Thermal Non-Equilibrium Model.
- User Defined EOS. This model allows you to specify the equation of state for a physics continuum using your own approach. See User-Defined EOS Material Properties.

Model Names and Abbreviations	**Constant Density**	**CD**
	Ideal Gas	**Ideal**
	IAPWS-IF97 (Water)	**IAPWS-w**

	Polynomial Density	**PolyD**
	Real Gas	**Real**
	Thermal Non-Equilibrium Ideal Gas	**TNEq**
	User Defined EOS	**UDEOS**
Theory	Ideal Gas Formulation Polynomial Density Formulation Real Gas Formulation Thermal Non-Equilibrium Formulation	
Provided by	**Continua** > **[physics continuum]** > **Models** > *Equation of State* box	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Constant Density**	
Requires	*Flow*: **Coupled Flow** or **Segregated Flow** **TNEq** specifically requires **Coupled Flow** **Ideal**, **Real**, and **TNEq** also require a gaseous material model **IAPWS-w** also requires *Material*: **Liquid**	
Properties	General Equation of State Properties Lookup and General Equation of State Properties	
Activates	Physics Models	For **CD**, *Optional Models*: **Boussinesq Model**. For **Ideal**, **IAPWS-w**, **PolyD**, **Real**, and **TNEq**, *Energy*: **Coupled Energy** becomes available. For **TNEq**, *Thermal Non-Equilibrium*: **Thermal Non-Equilibrium** becomes available. For **Real**, the *Real Gas Equation of State* box becomes available. See Real Gas Models.
	Model Controls (child nodes)	For **ID** and **TNEq**: Density Limits
	Materials	See General Equation of State Material Properties Lookup
	Reference Values	**Minimum Allowable Temperature**, **Maximum Allowable Temperature** (for **Ideal**, **IAPWS-w**, **PolyD**, **Real**, **TNEq**)
	Initial Conditions	Static Temperature (for **Ideal**, **IAPWS-w**, **PolyD**, **Real**, **TNEq**)
	Monitors	**Vib. Energy** (for **TNEq**, **UDEOS**)

General Equation of State Properties Lookup

This table shows which material properties are used by which equation of state models.

	CD	Ideal	IAPWS-w	PolyD	TNEq	UDEOS
Compressible						✓
Density Limiting		✓			✓	
Incompressible		✓			✓	

See Real Gas Models for the properties of individual real gas models.

General Equation of State Properties

Compressible

When turned **On**, this property specifies that the material is compressible (dependent on working pressure, p).

Density Limiting

When turned **On**, this property activates the *Density Limits* option that is used to prevent negative densities.

Incompressible

When turned **On**, this property specifies that the gas is incompressible (not dependent on working pressure p).

General Equation of State Material Properties Lookup

This table shows which material properties are used by which equation of state models.

	CD	Ideal	IAPWS-w	PolyD	TNEq	UDEOS
Critical Pressure			✓			
Critical Temperature			✓			
Density	✓					✓
Mode Relaxation Time					✓	
Molecular Weight		✓	✓		✓	
Molecule Type					✓	
Polynomial Density				✓		
Saturation Pressure			✓			
Saturation Temperature			✓			
Specific Heat		✓	✓	✓		✓
Speed of Sound			✓			

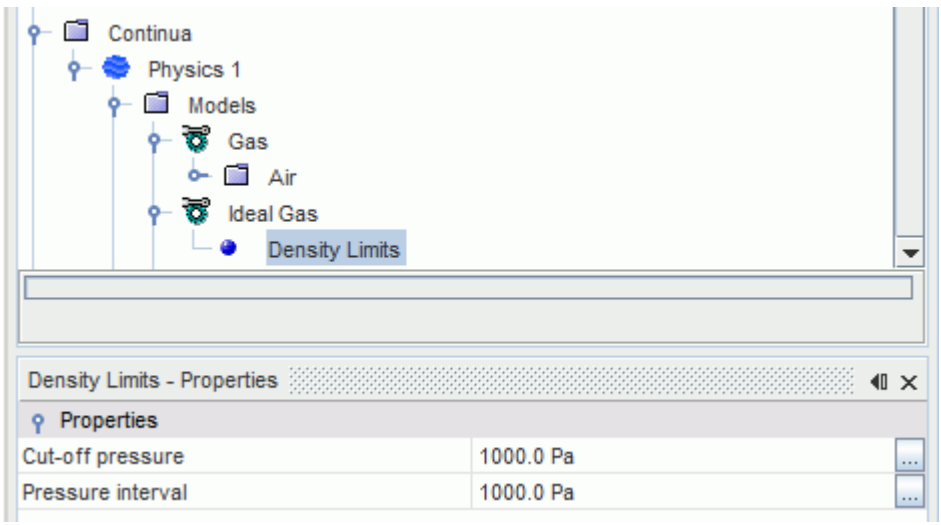
	CD	Ideal	IAPWS-w	PolyD	TNEq	UDEOS
Vibrational-Electronic Specific Heat					✓	

See Real Gas Models for the material properties of individual real gas models.

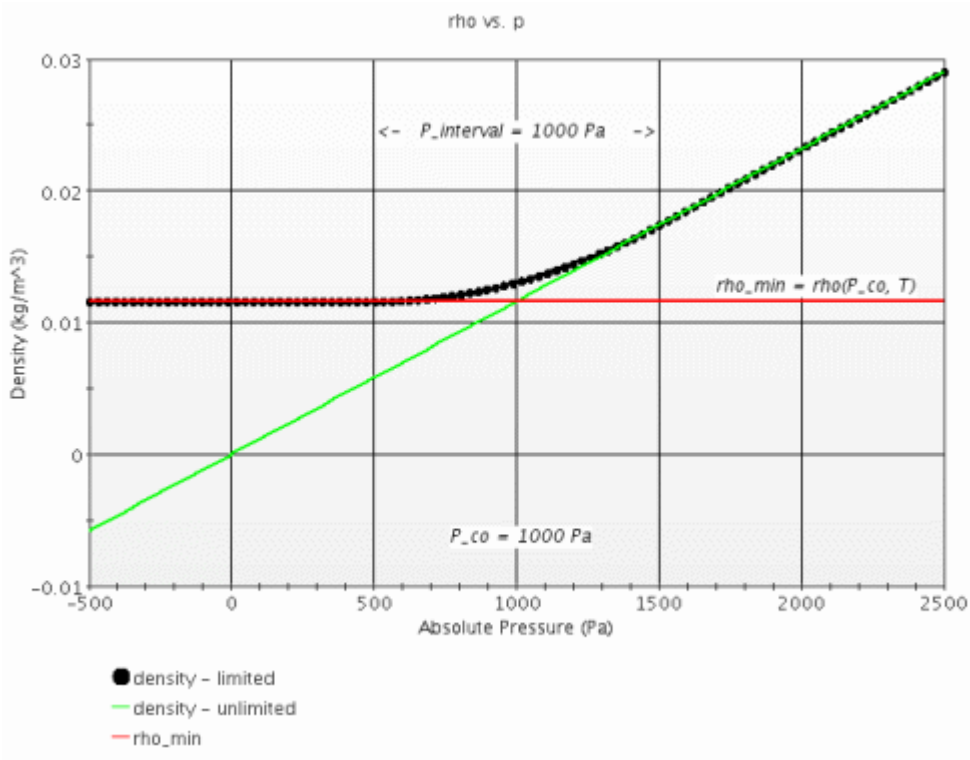
Density Limits

The **Density Limits** node lets you specify the density limiting parameters. This node appears under the model nodes when the *Density Limiting* property is turned **On**.

The following figures show the **Density Limits** node, with the *Cut-off pressure* and *Pressure interval* properties exposed.



The following figure shows a comparison of the limited and unlimited density as a function of absolute pressure. As seen from the figure, the limited density smoothly and continuously transitions from its unlimited value to a constant, minimum value, *rho_min*, over the course of the interval, *P_interval*, as specified by the Pressure Interval. The value *rho_min* is the density that is calculated at the existing temperature. An absolute pressure value as specified by the Cut-off Pressure, *P_co*.



Thermal Non-Equilibrium Model

This model becomes available when you select the Thermal Non-Equilibrium Ideal Gas (**TNEq**) model, and is selected automatically when **Auto-select recommended models** is active, since the TNEq requires it. Non-equilibrium occurs at high temperatures and low densities, where the vibrational/electronic energy modes become active yet the density is low enough such that equilibration does not occur.

Examples of problems include high-speed atmospheric re-entry flows and nozzles. For problems in this regime, thermal non-equilibrium is required in order to capture temperature profiles and heating accurately.

The Thermal Non-Equilibrium energy model is currently limited to the Coupled Flow model only. The Thermal Non-Equilibrium Ideal Gas model requires the use of the Coupled Thermal Non-Equilibrium model. It is a required model for the Equation of State.

The principal application for this model is hypersonic flows (mainly atmospheric re-entry). Nozzle flows where the effects of vibrational freezing are important can also be simulated. When Thermal Non-Equilibrium modeling is activated, the following material properties are evaluated:

- Vibrational-Electronic Specific Heat, with the methods Thermodynamic Polynomial Data or Gas Kinetics
- Mode Relaxation Time, with the methods Landau-Teller or Inverse Mole-Fraction Average
- Vibrational Conductivity, with the method Non-Equilibrium Kinetic Theory

Theory	Thermal Non-Equilibrium Formulation
Provided by	**Continua** > **[physics continuum]** > **Models** > *Thermal Non-Equilibrium* box
Example	**Continua** > **Physics 1** > **Models** > **Thermal Non-Equilibrium**

Node Path		
Requires	**Thermal Non-Equilibrium Ideal Gas**	
Properties	See Thermal Non-Equilibrium Properties	
Activates	Materials	The following material properties under the **Gas** model node: • Mode Relaxation Time • Molecule Type
	Monitors	Vib. Energy
	Field Functions	Combined Total Enthalpy Mode Relaxation Time for Air Temperature of Vibrational Mode Vibrational Enthalpy of Air Vibrational Specific Heat of Air

Thermal Non-Equilibrium Properties

Equalize Temperatures

When **On**, forces the Vibrational-Electronic temperature to equal the Translational-Rotational temperature.

This option is useful for problems with a low degree of thermal non-equilibrium and can be used to overcome start-up problems that are associated with a poor initial condition. Turn it **Off** when the problem begins to converge towards a solution.

Freeze Vibrational Temperature

When **On**, freezes the Vibrational temperature at its current value.

This option is useful for problems with a large degree of thermal non-equilibrium and can be used to overcome start-up problems that are associated with a poor initial condition. Turn it **Off** when the problem begins to converge towards a solution.

User-Defined EOS Material Properties

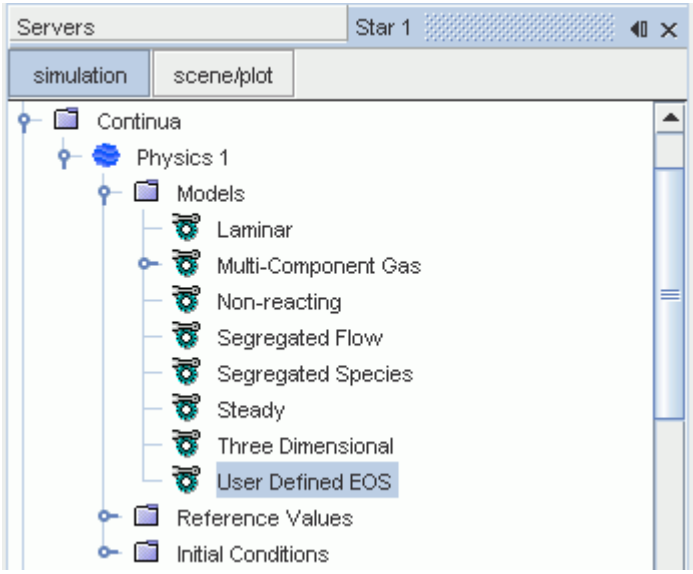
When you use the User Defined EOS model, some or all of the following material property nodes are added for each material in the continuum, depending on the other model selections you have made for a continuum:

- Density, ρ
- Density pressure derivative, $d(\rho)/d(p)$
- Density temperature derivative, $d(\rho)/d(T)$
- Density mass fraction derivative, $d(\rho)/d(Y_i)$
- Specific heat, C_p
- Enthalpy, h
- Enthalpy pressure derivative, $d(h)/d(p)$

- Entropy, s
- Speed of sound, c

The User Defined EOS model provides you with various methods for specifying these thermodynamic properties. These methods include constant, field function, temperature polynomial, and the option to input tabular data from a file.

The model also allows you to indicate whether the fluid is compressible, that is, whether density is a function of pressure, by activating the *Compressible* property of the **User Defined EOS** model node.



When the working fluid is declared compressible, the material property Density Pressure Derivative appears with the material. This property defines the partial derivative of the user-defined density with respect to pressure. The Speed of Sound material property is included for compressible materials as well.

If an Energy model is activated in the physics continuum, the material property Density Temperature Derivative is activated. This property defines the partial derivative of the user-defined density with respect to temperature. Nodes are also added for Enthalpy and Entropy.

If a Multi-Component Gas or Multi-Component Liquid model is activated in the physics continuum, the material property Density Mass-Fraction Derivative is activated for each mixture component. This property defines the partial derivative of the user-defined density with respect to mass-fraction of the component.

The User Defined Equation of State model is applicable in instances where the density of the working fluid best represented with some user-defined expression or a table of property data. You use this model when the other Equation of State models (Constant Density, Polynomial Density, or Ideal Gas) do not adequately define the density.

As an example, one can choose to model a compressible liquid by specifying density using a field function that is defined as:

$$\rho = \rho_0 + \frac{p}{c^2} \tag{214}$$

where ρ_0 is a constant, p is the pressure, and c is the (constant) speed of sound. In this example, the following function would be specified for Density Pressure Derivative:

$$\frac{d(\rho)}{d(p)} = \frac{1}{c^2} \tag{215}$$

and the constant c would be specified for the speed of sound.

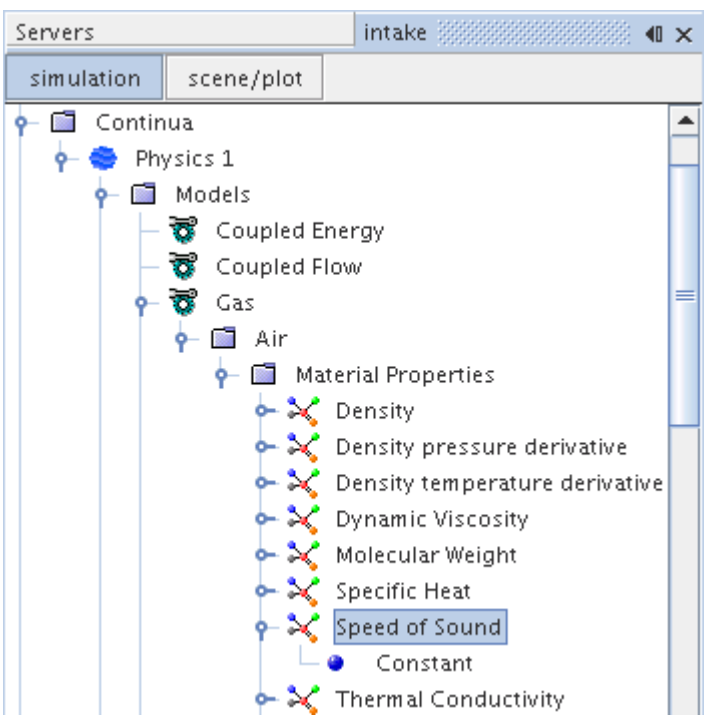
When you specify thermodynamic properties such as density and enthalpy using tabular input data, STAR-CCM+ can evaluate the necessary derivative terms with respect to pressure and temperature by numerically differentiating the tabular data. This calculation eliminates the need to provide more columns of data for the derivative terms. To increase accuracy and robustness of the solver, this derivative data can also be specified explicitly using tabular input data. When explicit derivative values are available, it is best to provide this derivative data to STAR-CCM+.

All other models that interact with the Equation of State model continue to function as usual when the user-defined model is active.

Using Speed of Sound

The speed of sound is the rate of acoustic propagation in the medium. It is used at certain boundary conditions (pressure outlet, mass flow inlet, and free-stream) when the flow is compressible. (It also appears in the compressibility modification (dilatation dissipation) for the K-Epsilon and Reynolds Stress turbulence models.)

This material property becomes available once you have selected the User Defined EOS model and activated its *Compressible* property.



You can set the speed of sound for a material using a Constant, the Equilibrium method, a Field Function, or (if the energy model is available) a Table(T,P).

For a multi-component material, you can set the speed of sound for the material as a property of the mixture. For a Eulerian multiphase material, you can set the speed of sound for each phase of the material as a property of the phase. Eulerian multiphase materials use the speed of sound in the dissipation term in the K-Epsilon and Reynolds Stress turbulence models only.

Note: Eulerian multiphase models do not have a full implementation of this property for high Mach number (M > 0.3).

See Also:

Isentropic Efficiency Report

Defining the Reference Pressure

Equation of State Field Functions Reference

Real Gas Models

Real Gas models are used in situations where the gas properties cannot be explained using the ideal gas law —for example, near the condensation point of gases, near critical points, and at high pressure.

Real Gas models take into account non-ideal behavior such as compressibility effects, variable specific heat, van der Waals forces, and non-equilibrium thermodynamic effects. The Real Gas models are suitable for modeling complex chemistry combustion applications at high pressures and low temperatures in transcritical and supercritical environments, such as in liquid rocket engines. These applications can have flow regions in which the equation of state departs substantially from ideal gas behavior.

The following Real Gas models are available in STAR-CCM+:

- Peng-Robinson, for use at high pressures and low temperatures
- Redlich-Kwong, for use at high pressures and low temperatures
- Soave-Redlich-Kwong, for more complex treatment of temperature
- Modified Soave-Redlich-Kwong, which is designed to handle phase equilibrium calculations for non-polar molecules ([211], [212])
- Van der Waals, which is designed to handle intermolecular attractive forces
- Equilibrium Air, for high temperatures, to handle effects such as ionization
- IAPWS-IF97 (available for steam or liquid water)

Model Names and Abbreviations	**Peng-Robinson**	**PR**
	Redlich-Kwong	**RK**
	Soave-Redlich-Kwong	**SRK**
	Modified Soave-Redlich-Kwong	**mSRK**
	Van der Waals	**VdW**
	Equilibrium Air	**EA**
	IAPWS-IF97 (Steam)	**IAPWS-s**

Theory	Peng-Robinson Formulation Redlich-Kwong Formulation Soave-Redlich-Kwong Formulation Modified Soave-Redlich-Kwong Formulation Van der Waals Formulation Equilibrium Air Formulation IAPWS-IF97 Models	
Provided by	**Continua** > **[physics continuum]** > **Models** > *Real Gas Equation of State* box	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Peng-Robinson**	
Requires	*Equation of State*: **Real Gas**	
Properties	**PR**, **RK**, **SRK**, **mSRK**, **VdW**, **EA**: *Density Limiting* **IAPWS-IF97 (Steam)**: none	
Activates	Model Controls (child nodes)	**Density Limits** This node appears under **PR**, **RK**, **SRK**, **mSRK**, **VdW**, and **EA** when the *Density Limiting* property is turned **On**.
	Materials	See Real Gas Material Properties
	Monitors	**Vib. Energy**

Real Gas Material Properties

This table shows which material properties are used by which real gas models.

	PR	RK	SRK	mSRK	VdW	EA	IAPWS-s
Acentric Factor ω	✓		✓	✓			
Critical Pressure p_c	✓	✓	✓	✓	✓		✓
Critical Temperature T_c	✓	✓	✓	✓	✓		✓
Molecular Weight	✓	✓	✓	✓	✓	✓	✓
Saturation Pressure							✓

	PR	RK	SRK	mSRK	VdW	EA	IAPWS-s
Saturation Temperature							✓
Specific Heat	✓	✓	✓	✓	✓	✓	✓
Speed of Sound						✓	✓

These properties are constant. Default values for these constants are provided from the material database, though you can change them if desired.

Equilibrium Air Model

The equilibrium air equation of state is used for the simulation of high-temperature air, where the effects of molecular dissociation, internal energy excitation, and ionization become significant.

For situations where the time scale of these effects is much shorter than the time scale of the flow, these effects can be treated in an equilibrium manner (see Equilibrium Air Formulation). With the equilibrium assumption, this model can capture the effects of dissociation and ionization in air without the computational expense of a multicomponent, reacting simulation.

The curve fits of this model are valid for temperatures less than 30,000 K and for pressures between 10^{-4} and 10^{2} atmospheres. Since the curve fits are based on the behavior of air, the Equilibrium Air equation of state can only be used with air as the working fluid.

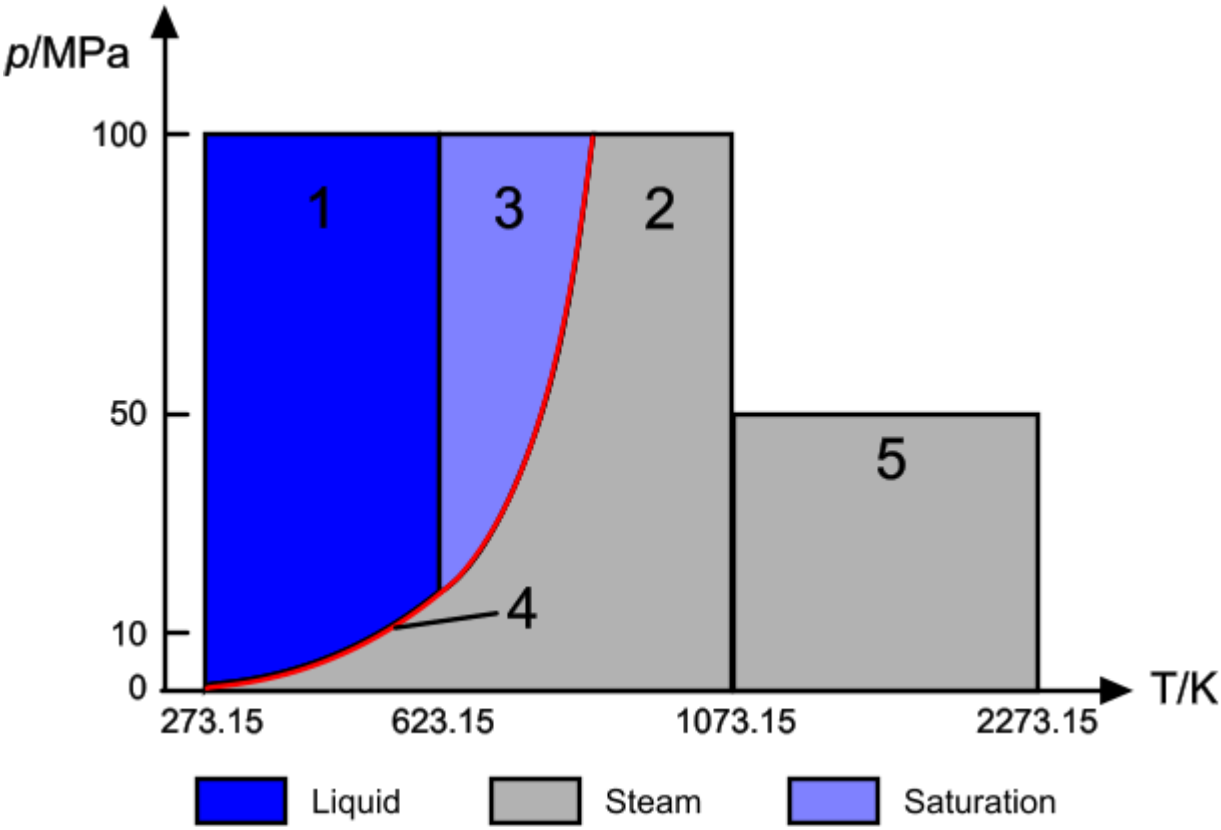
This equation of state is useful for the simulation of hypersonic flows, such as nozzle flows or atmospheric re-entry. When Equilibrium Air modeling is activated, the following material properties are evaluated:

- Specific Heat
- Dynamic Viscosity
- Thermal Conductivity
- Speed of Sound

The IAPWS-IF97 Models

These models include calculations of density and other thermodynamic properties. The IAPWS-IF97 provides fundamental polynomial equations for the specific Gibbs free energy, $g(p,T)$. Specific volume, internal energy, entropy, enthalpy, heat capacity, and speed of sound are derived from the fundamental equation using appropriate combinations of the dimensionless Gibbs free energy and its derivatives [217].

Based on the Gibbs free energy, the IAPWS-IF97 models in STAR-CCM+ are only valid within certain ranges of temperature and pressure. These ranges are represented in a diagram of regions [217] which are essentially the phases of water and steam.



The following table lists the components of the region diagram and how they work with IAPWS-IF97 models in STAR-CCM+.

Component	Ranges of Validity		STAR-CCM+ Models
Region 1 (liquid)	273.15 K ≤ T ≤ 623.15 K	p ≤ 100 MPa	IAPWS-IF97 (Water)
Region 2 (steam)	273.15 K ≤ T ≤ 1073.15 K	p ≤ 100 MPa	IAPWS-IF97 (Steam)
Region 3 (saturation)	This phase region is not supported in STAR-CCM+.		
Boundary 4	This boundary lies between liquid and saturation on one side, and steam on the other.		
Region 5 (steam)	1073.15 K < T ≤ 2273.15 K	p ≤ 50 MPa	IAPWS-IF97 (Steam)

The enthalpies H_{sh}^{L} of superheated liquid and H_{sc}^{V} of subcooled vapor are calculated as:

$$H_{sh}^{L}(T,P) = H_{r1}(T_{sat},P) + cp_{r1}(T_{sat},P)(T - T_{sat}) \ \ (T > T_{sat}) \tag{216}$$

$$H_{sc}^{V}(T,P) = H_{r2}(T_{sat},P) + cp_{r2}(T_{sat},P)(T - T_{sat}) \ \ (T < T_{sat}) \tag{217}$$

where:

- $T_{sat}(P)$ is the saturation temperature
- The subscripts $r1$ and $r2$ refer to Region 1 and Region 2 respectively.

Selecting an IAPWS-IF97 Model

Activate one of the two models depending on whether you are simulating liquid water or gaseous steam:

- For liquid water, select the IAPWS-IF97 (Water) model from the *Equation of State* group. It is available once the Liquid material model is activated.
- For gaseous steam, select the IAPWS-IF97 (Steam) model from the *Real Gas Equation of State* group. It is available once the Gas material model and Real Gas model are selected.

When either IAPWS-IF97 model is activated, the IAPWS-IF97 method is available for the following material properties:

- Dynamic Viscosity
- Thermal Conductivity

When either IAPWS-IF97 model is activated, the IAPWS-IF97 method is the *only* one available for these properties:

- Critical Pressure
- Critical Temperature
- Molecular Weight
- Saturation Pressure
- Saturation Temperature
- Speed of Sound

Also when this model is activated, the following fundamental quantities are calculated using IAPWS-IF97:

- Entropy
- Specific Heat

See Also:

Modeling Flow and Energy

Equation of State Field Functions Reference

Entropy Reference

When the Ideal Gas model is used, entropy is defined as:

$$S = \frac{\left(T/T_{ref}\right)^{\left(C_p/R\right)}}{\left(P/P_{ref}\right)} \tag{218}$$

where:

- T is temperature K and T_{ref} is the reference temperature.
- P is absolute pressure Pa and P_{ref}, is the reference pressure.
- C_p is specific heat J/(kg K), a constant.
- R is a specific gas constant J/(kg K).

This expression can be rewritten as:

$$R\ \ln(S) = C_p \ln\left(T/T_{ref}\right) - R\ \ln\left(P/P_{ref}\right) \tag{219}$$

The more familiar representation of entropy s is:

$$s = C_p \ln\left(\frac{T}{T_{ref}}\right) - R\ \ln\left(\frac{P}{P_{ref}}\right) \tag{220}$$

Comparison of Eqn. (219) and Eqn. (220) shows that entropy s can be related to the STAR-CCM+ Entropy Function field function S as follows:

$$s = R\ \ln(S)$$

where $T_{ref} = 1\ \mathrm{K}$ and $P_{ref} = 1\ \mathrm{Pa}$.

When the IAPWS-IF97 model is activated, entropy is calculated using that method.

Entropy Field Functions

STAR-CCM+ provides the following entropy field functions:

Field Function	Definition
Entropy	The true thermodynamic entropy. For an ideal gas, the entropy s is as defined in Eqn. (220). For the other equations of state, different formulas are used to ensure that the correct value for entropy is calculated and displayed in the field function. This field function is available with the Ideal Gas Equation of State model, and with the following Real Gas Equation of State models: • Peng-Robinson • Redlich-Kwong • Soave-Redlich-Kwong • Modified Soave-Redlich-Kwong • Van Der Waals • Equilibrium Air • IAPWS The entropy value depends on the *Specific Heat* material property method that you use. If you change the *Specific Heat* method, the field function value changes accordingly. These Equation of State models use a reference temperature and a standard-state entropy. If these values are not specified, the defaults are used: the reference temperature is assumed to be 298.15 K and the standard-state entropy is assumed to be 0. In

Field Function	Definition
	this case, negative entropy values are possible as a negative value implies entropy levels below the standard state level. When you use thermodynamic polynomials to set the *Specific Heat*, the polynomials have a reference temperature and entropy built into the specification. These values are used to calculate entropy. Therefore, if you change to a different *Specific Heat* method, the entropy can be expected to jump due to the inclusion of a non-zero standard state entropy.
Entropy Function	The value S where $s = R\ \ln(S)$ as shown by comparison of Eqn. (219) and Eqn. (220). This field function measures the degree to which a flow is isentropic, with a value of 1 indicating isentropic flow.

Equation of State Field Functions Reference

The following primitive field functions are made available to the simulation when the Ideal Gas model, a Real Gas model (Redlich-Kwong, van der Waals), or the IAPWS-IF97 model is used.

These field functions have a typical set of base properties.

Ratio of Specific Heats

Expressed as:

$$\gamma = \frac{C_p}{C_v} = \frac{C_p}{\left(C_p - R\right)}$$

where C_p is specific heat J/(kg K), and R is a specific gas constant J/(kg K).

Real Gas/IAPWS-IF97 Field Functions Reference

The primitive field functions in the table below are made available to the simulation when one of the following models is used:

- Peng-Robinson
- Redlich-Kwong
- Soave-Redlich-Kwong
- Modified Soave-Redlich-Kwong
- van der Waals
- IAPWS-IF97

See Real Gas Models.

These field functions have a typical set of base properties.

Critical Pressure	Expressed as p_c .
Critical Temperature	Expressed as T_c .
Reduced Pressure	Expressed as $p_r = p/p_c$.
Reduced Temperature	Expressed as $T_r = T/T_c$.
Compressibility Factor	Represents the amount of deviation of the real gas from an ideal gas, expressed as $Z = p/(\rho\, R\, T)$. This factor usually takes on values around 1. For the Equilibrium Air model, the compressibility factor represents the degree of dissociation, which is non-ideal behavior, but has values that vary more widely (approximately between 1 and 6). A simplified formula that applies to the equilibrium air model is given as Eqn. (1481).

The following primitive field function is available only with the Peng-Robinson, Redlich-Kwong, Soave-Redlich-Kwong, Modified Soave-Redlich-Kwong, or van der Waals model. It has a typical set of base properties.

Real Gas Specific Heat	The value of specific heat at constant pressure of the real gas. This field function accounts for the effects of variations in both temperature and pressure by including the departure from the ideal gas value. The last three terms on the right-hand side of Eqn. (221) give the departure from ideal: $$C_p = C_p^0 + T\int_{\infty}^{v}\left(\frac{\partial^2 P}{\partial T^2}\right)_v dv - T\left(\frac{\partial P}{\partial T}\right)_v^2 \left(\frac{\partial P}{\partial v}\right)_T^{-1} - R \quad (221)$$

The following primitive field function is available only with either IAPWS-IF97 model. It has a typical set of base properties.

IAPWS Region ID	Data that STAR-CCM+ calculates for a particular IAPWS validity region.

The following primitive field functions are available only with the Thermal Non-Equilibrium model. They have a typical set of base properties.

Temperature of Vibrational Mode	Vibrational temperature (characteristic temperature of the vibrational/electronic modes).
Vibrational Enthalpy	Energy associated with the vibrational and electronic energy modes.

Combined Total Enthalpy	Sum of total translational-rotational enthalpy with vibrational enthalpy.
Vibrational Specific Heat	Specific heat for the vibrational/electronic modes.
Mode Relaxation Time	Relaxation time between the vibrational mode and translational mode.
Vibrational Conductivity	Thermal conductivity associated with the vibrational/electronic energy modes.

See Also:

Real Gas Models

General Equation of State Models

Modeling Flow and Energy Using a Coupled Approach

This section explains the following:

- The coupled flow and coupled energy models
- The coupled implicit and coupled explicit solvers
- Coupled flow controls and general setup recommendations
- How to select the coupled flow and coupled energy models
- Running a steady state analysis
- Running a transient analysis
- Troubleshooting

Contents:

Coupled Flow Model Reference
What Is the AUSM+ Scheme?
Coupled Energy Model Reference
Running a Steady State Analysis
Running a Transient Analysis
What Is Implicit Unsteady?
What Is Explicit Unsteady?
What Is the Coupled Implicit Solver?
Setting Grid Sequencing Parameters
Using the Expert Driver
Setting Convergence Accelerators

Coupled Flow Model Reference

The Coupled Flow model solves the conservation equations for mass, momentum, and energy simultaneously using a pseudo-time-marching approach.

One advantage of this formulation is its robustness for solving flows with dominant source terms, such as rotation. Another advantage of the coupled solver is that CPU time scales linearly with cell count; in other words, the convergence rate does not deteriorate as the mesh is refined.

This model can also evaluate inviscid fluxes with the optional AUSM+ scheme, which offers advantages for various cases.

Coupled Flow Model Reference

Theory	See Coupled Flow Solver.
Provided By	**[physics continuum]** > **Models** > *Flow*

Example Node Path	**Continua** > **Physics 1** > **Models** > **Coupled Flow**	
Requires	*Material*: one of **Gas**, **Liquid**, **Multi-Component Gas**, **Multi-Component Liquid** (For multi-component materials) *Reaction Regime*: any option	
Properties	See Coupled Flow Properties.	
Activates	Physics Models	Model Groups: *Equation of State* *Viscous Regime*
	Reference Values	**Reference Pressure**. See Defining the Reference Pressure.
	Initial Conditions	**Pressure** The initial value at any pressure boundary, entered as a scalar profile. **Velocity** The initial fluid velocity that is entered as a scalar profile, required non-zero for inviscid simulations and for initial turbulence intensity.
	Boundary Inputs	Free Stream: **Flow Direction**, **Mach Number**, **Pressure** Mass Flow Inlet: **Mass Flow Rate**, **Supersonic Static Pressure** Pressure Outlet: **Pressure** Stagnation Inlet: **Supersonic Static Pressure**, **Total Pressure** Velocity Inlet: **Velocity Magnitude** See Flow Boundaries Reference.
	Region Inputs	**Mass Source Option** See Mass Source Option under "Flow Node Reference." **Momentum Source Option** See Momentum Source Option under "Flow Node Reference."
	Solvers	Coupled Implicit
	Monitors	**Continuity** **X-, Y-, and Z-Momentum**
	Field Functions	**Absolute Pressure**, **Absolute Total Pressure**, **Axial Velocity**, **Cell Relative Velocity**, **CFL Number**, **Density**, **Flow Direction**, **Helicity**, **Lamb Vector**, **Lambda 2 Criterion**, **Local Time Step**, **Mass Flow Rate**, **Mass Flux**, **Mass Imbalance**, **Pressure**, **Pressure Coefficient**, **Q-Criterion**, **Radial Velocity**, **Reference Velocity**, **Relative Tangential Velocity**, **Static Pressure**, **Tangential Velocity**, **Total Pressure**, **Total Pressure Coefficient**, **Turbulent Charge**, **Velocity**, **Vorticity** See Common Flow and Energy Field Functions.

Coupled Flow Properties

Integration

Switches between implicit and explicit pseudo-time integration.

Implicit

The coupled implicit solver is invoked. Adds a Coupled Implicit node to the **Solvers** node with the option to set a Courant number ramp, grid sequencing, and expert driver for the solver.

Explicit

The coupled explicit solver is invoked. Adds a Coupled Explicit node to the **Solvers** node with the option to set a Courant number ramp.

Discretization

Selects the discretization scheme that is used for evaluating face values for convection and diffusion fluxes.

1st-order

Selects the first-order upwind discretization scheme.

2nd-order

Selects the second-order upwind discretization scheme.

Central

Selects the central-differencing discretization scheme (available with Large Eddy Simulation (LES) only).

Bounded-Central

Selects the bounded central-differencing discretization scheme (available with Large Eddy Simulation (LES) only). Adds a Bounded Differencing sub-node to the **Coupled Flow** node.

Hybrid

Selects the hybrid second-order upwind/central-differencing discretization scheme (available with Detached Eddy Simulation (DES) only).

Hybrid-BCD

Selects the hybrid second-order upwind/bounded central-differencing discretization scheme (available with Detached Eddy Simulation (DES) only). Adds a Bounded Differencing sub-node to the **Coupled Flow** node.

MUSCL 3rd-order/CD

Selects the hybrid MUSCL 3rd-order/CD convection scheme. Adds a Bounded Differencing sub-node to the **Coupled Flow** node.

Positivity Rate Limit

Maximum rate by which temperature is allowed to decrease.

Explicit Relaxation

A scaling factor that is used to relax all corrections explicitly before they are applied to the variables.

Minimum Absolute Pressure

Minimum allowable absolute pressure.

Preconditioning Enabled

When **On**, activates *Unsteady Low-Mach Preconditioning*. This property includes a preconditioning matrix in the governing equations for the Coupled Flow model, in both original and discretized form. Using this matrix improves convergence and accuracy. This improvement applies to solving both compressible and incompressible flows at all speeds.

Unsteady Low-Mach Preconditioning

When **On**, activates unsteady preconditioning to improve the accuracy for sound propagation in low-speed compressible regimes. Only active when *Preconditioning Enabled* is active. The default is **Off**.

Unsteady Preconditioning Max Factor

The maximum factor for unsteady preconditioning, a value from 0 through 1. The default is 0.95. Larger values can reduce robustness. At zero, the solver uses no high Strouhal number unsteady numerical dissipation.

The *Unsteady Low-Mach Preconditioning* and *Unsteady Preconditioning Max Factor* properties control treatment of mass-flux pressure dissipation, D_p (described in [31]). When you activate unsteady preconditioning, D_p is a function of the ratio of a local time-step to the user-defined time-step. This local time-step is obtained without preconditioning, assuming CFL = 1.

At high speeds, no unsteady modification is done for D_p. At low speeds (approaching and in the incompressible regime), D_p depends on local flow conditions and also on the local Strouhal number.

At high Strouhal numbers (small time-steps), D_p corresponds to the unpreconditioned mass-flux pressure dissipation. At low Strouhal numbers and towards steady-state (when time-step tends to infinity), D_p moves towards the steady-state low-Mach preconditioned mass-flux pressure dissipation. This move happens because the dissipation depends on the ratio of the local time-step to the user-defined time-step, which decreases as the time-step increases.

Pressure Difference Scale Factor

ϵ from Eqn. (1656).

Minimum Reference Velocity

$U_{r,\ \min}$ from Eqn. (1656).

Maximum Reference Velocity

$U_{r,\ max}$ from Eqn. (1656).

Flow Boundary Diffusion

When **On**, activates the flow-boundary diffusion fluxes (or viscous fluxes for flow models). See Finite Volume Discretization boundary conditions. The default is **Off**.

Secondary Gradients

There are two sources of secondary gradients in STAR-CCM+ flow solvers:

- Boundary secondary gradients for diffusion
- Interior secondary gradients at cell faces

Use this property to control which gradients are included in the solver. **On** gives both gradients while **Off** excludes them. **Interior Only** and **Boundaries Only** select the corresponding gradients.

Coupled Inviscid Flux

Selects the discretization scheme for the coupled inviscid flux.

Roe FDS

Roe's flux-difference splitting scheme, adapted for use with Weiss-Smith preconditioning for all-speed flows. At the incompressible limit, the pressure difference at the flux interface is a blend of reconstructed and average states. This blending results in a less dissipative scheme for low speed, constant density flows.

Roe FDS - no blending	Same as the **Roe FDS** option, but without blending at the incompressible limit.
AUSM+ FVS	Liou's AUSM+ flux-vector splitting scheme.

Bounded Differencing

Bounded differencing lets you specify parameters for:

- The bounded central-differencing discretization scheme when you work with Large Eddy Simulation (LES).
- The hybrid second-order upwind/bounded central-differencing discretization scheme when you work with Detached Eddy Simulation (DES).

The **Bounded Differencing** node is activated when the *Discretization* property of the Coupled Flow model is set to **Bounded-Central** (with LES) or **Hybrid-BCD** (with DES).

Bounded Differencing Properties

Upwind blending factor

Specifies the proportion of upwind differencing, related to Eqn. (1587).

The default maximum of this property, `0.15`, provides the most robustness for the scheme. Reducing it would, in principle, increase accuracy. However, unless you are thoroughly familiar with the theoretical aspects of bounded differencing, do not change this property. The default value reflects optimization for accuracy and performance.

See Also:

What Is the AUSM+ Scheme?

What Is the AUSM+ Scheme?

The AUSM+ scheme is an optional scheme for evaluating inviscid fluxes.

An optional scheme for evaluating inviscid fluxes is Liou's AUSM+ flux-vector splitting scheme (see [221] and [222].) Like the Weiss-Smith preconditioned Roe's flux-difference splitting scheme, this scheme is also based on the upwind concept, which is applied consistently on the two physically distinct parts of the inviscid flux: convective and pressure. The AUSM+ inviscid flux scheme is the more appropriate choice for solving flows involving high supersonic and/or hypersonic regimes (Mach number of around 3 and above), when compared with the Roe scheme.

The main advantages of the AUSM+ scheme are:

- Algorithmic simplicity and straightforward extension for complex conservation laws.
- Accurate capture of shock- and contact-discontinuity.
- Numerical solutions that preserve positivity and satisfy entropy.
- Reduced susceptibility to "carbuncle" phenomena.

AUSM+-up is the low-Mach preconditioned formulation that is used in STAR-CCM+. It is included primarily as an option for use with high supersonic and/or hypersonic flows. It also provides uniform accuracy and robust convergence for all Mach number flows.

The AUSM+-up scheme can be employed for solving a wide range of flow problems including incompressible to hypersonic aerodynamics, turbomachinery flows, LES/DES of turbulent flows and aeroacoustics.

See Also:

Modeling Flow and Energy

Coupled Flow Model Reference

Coupled Energy Model Reference

The Coupled Energy model is an extension of the Coupled Flow model.

Together they solve the conservation equations for mass, momentum, and energy simultaneously using a time- (or pseudo-time-) marching approach. This formulation is robust for solving compressible flows and flows with dominant source terms, such as buoyancy.

Due to the preconditioned form of the governing equations that the Coupled Flow and Coupled Energy models use, convergence rate is effectively independent of Mach number, ranging from incompressible through to supersonic regimes.

Coupled Energy Model Reference

Theory	See Coupled Flow Solver.
Provided By	**[physics continuum]** > **Models** > *Optional Models*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Coupled Energy**
Requires	*Material*: one of **Gas**, **Liquid**, **Multi-Component Gas**, **Multi-Component Liquid**

	(For multi-component materials) *Reaction Regime*: any option *Flow*: **Coupled Flow**	
Properties	*Enthalpy Formulation* When **On**, activates the enthalpy formulation for the coupled energy model. When activated, enthalpy is the dependent variable instead of temperature. The default is **Off**. **Flow Boundary Diffusion** When **On**, this property includes the flow-boundary diffusion fluxes (or viscous fluxes for flow models) that come from Eqn. (1592). The default is **On**.	
Activates	Physics Models	*Optional Models*: **Boiling**, **Thermal Comfort**, and **Thin Film**
	Reference Values	**Minimum Allowable Temperature** **Maximum Allowable Temperature** See Reference Values under "Flow Node Reference."
	Initial Conditions	**Static Temperature**
	Boundary Inputs	**Free Stream, Pressure Outlet** **Static Temperature** Specify the temperature at the interface. **Inlet Boundaries** **Total Temperature** The total temperature of the boundary.
	Region Inputs	**Energy Source Option** See Energy Source Option under "Modeling Heat Transfer."
	Monitors	**Energy**
	Field Functions	**Specific Heat**, **Temperature**, **Temperature Coefficient**, **Total Enthalpy**, **Total Pressure** See Common Flow and Energy Field Functions.

Coupled Energy Properties

Enthalpy Formulation

When **On**, uses enthalpy as the dependent variable when solving the energy conservation equation. When **Off**, uses temperature as the dependent variable. The default is **Off**.

Flow Boundary Diffusion

When **On**, activates the flow-boundary diffusion fluxes (or viscous fluxes for flow models). See Eqn. (1592). The default is **Off**.

See Also:

Coupled Flow Solvers

Running a Steady State Analysis

For steady simulations, the coupled solver in STAR-CCM+ employs a time marching scheme to drive the unsteady form of the governing equations to a steady state.

In this case, a pseudo-transient term replaces the physical time-derivative. The solution advances in pseudo-time to drive this term to zero. The solution in each cell is advanced independently with an optimal pseudo-time step computed locally according to stability constraints. In this way, convergence to steady state is achieved in the most efficient manner.

Running a Transient Analysis

Transient analysis in STAR-CCM+ is used to solve time-accurate problems, including initial value problems.

Two different time integration schemes are available for transient analysis:

- Implicit Unsteady
- Explicit Unsteady

You choose between these two by activating either the Implicit Unsteady Model or the Explicit Unsteady Model, while also selecting the Coupled Flow model. The Coupled Energy model can be chosen as an option.

See Also:

What Is Explicit Unsteady?
What Is Implicit Unsteady?
What Is the Implicit Unsteady Model?
What Is the Explicit Unsteady Model?

What Is Implicit Unsteady?

The Implicit Unsteady approach is appropriate if the time scales of the phenomena of interest are either of the following:

- The same order as the convection and/or diffusion processes (for example, vortex shedding).
- Related to some relatively low frequency external excitation (for example, time-varying boundary conditions or boundary motion).

In the Implicit Unsteady approach, each physical time-step involves some number of inner iterations to converge the solution for that given instant of time. These inner iterations can be accomplished using implicit spatial integration or explicit spatial integration schemes. You specify the physical time-step size that is used

in the outer loop. The integration scheme marches inner iterations using optimal pseudo-time steps that are determined from the Courant number.

With the Implicit Unsteady approach, you are required to set the physical time-step size, the Courant number, and the number of inner iterations at each physical time-step.

The transient phenomena being modeled generally governs the physical time-step size. The time step must at least satisfy the Nyquist sampling criterion: more than two time-steps per period are required.

In general, the guidelines for setting the Courant number in steady-state integration schemes also apply to the transient integration schemes.

The number of inner iterations per physical time-step is harder to quantify. Generally, you determine this number by observing the effect that it has on results. Select a number of inner iterations, plot a monitor for one or more specific quantities against iteration, and see whether these monitors are converging within each time-step.

Smaller physical time-steps generally mean that the solution is changing less from one time step to the next; fewer inner iterations are then required. There is an optimal balance of time-step size and number of inner iterations for a given problem and desired transient accuracy.

See Also:

Running a Transient Analysis

What Is Explicit Unsteady?

If the unsteady time scales are of the order of the acoustic processes (for example, shock front tracking), the Explicit Unsteady approach is the proper choice.

The Explicit Unsteady approach is effectively the same as the explicit spatial integration scheme, but using the same physical time step for all cells in the domain (rather than local time-stepping). In this case, the solver automatically determines the size of the time-step as Eqn. (1670) such that one value satisfies the Courant condition at all points; the minimum allowable time-step is used everywhere. Thus, each iteration becomes a time-accurate advancement of the solution. In this case, the Courant number must be 1 or less, resulting in a physical time-step that varies from one iteration to the next as the flow field changes. Furthermore, preconditioning of the governing equations is omitted; the Explicit Unsteady approach is therefore unsuited for incompressible flow simulations.

See Also:

Running a Transient Analysis

Coupled Flow Solvers

The coupled approach can use either implicit or explicit pseudo-time integration. The default is implicit.

Coupled Implicit Solver

The Coupled Implicit solver controls the solution update for the Coupled Flow model. If the Coupled Energy Model and the Coupled Species Model are activated, the Coupled Implicit solver controls these models also.

The solver is used for implicit spatial integration in both steady and unsteady analyses, using a coupled algebraic multi-grid method.

To set the Courant number for this solver, enter the value in the properties. The default value of Courant number for the implicit scheme is 50 for unsteady flows, and 5 for steady flows. While these values are fine for many simulations, they are not optimal in all cases. Generally, there is some optimal value of Courant number that balances the total number of iterations against the time per iteration. This optimum value is a function of many factors: whether the flow is compressible, how close the solution is to convergence, mesh quality and the magnitude of the body forces.

Solving the governing discretized equations implicitly in pseudo-time results in a wider stability margin; this approach permits Courant numbers greater than unity. The large resulting local pseudo-time steps $\Delta\tau$, given by Eqn. (1673), provide relatively fast convergence rates. The penalty for this approach, however, is a larger storage requirement than the explicit spatial integration scheme.

Guidelines:

- If the solution diverges for more than a few iterations after initialization, or if the solver reports that corrections are being limited, it is necessary to reduce the Courant number.
- If the solver reduces the Courant number automatically for more than 1 or 2 iterations, the Courant number must be manually reduced.
- If the Courant number is too high, especially during initial iterations, the convergence can be slowed due to the multigrid solver working too hard. A suggested procedure for monitoring this behavior is as follows:
  - Select the **Solvers** > **Coupled Implicit** > **AMG Linear Solver** node.
  - To display the number of inner multigrid, set the *Verbosity* property to low.
  - There must be about 3–4 inner iterations per global iteration. If there are many more, reduce the Courant number.
- If the Courant number is set high and the residuals do not decline at some point after many iterations, try reducing the Courant number.
- Consider using ramping to increase the CFL automatically to its final value during the early iterations.

The following sections discuss how to set the following:

- Algebraic multigrid parameters for the Coupled Implicit solver
- Expert initialization for grid-sequencing parameters
- Expert driver parameters
- Convergence accelerators
- High-Order Temporal Discretization

Coupled Implicit Solver Properties

Courant Number

The value of CFL in Eqn. (1673) that controls the local time-step. The default Courant number is 50 for unsteady flows, and 5 for steady flows. In general, and depending on the initial conditions, it is good practice to start at a lower Courant number, and then increase it.

Freeze Flow

When **On**, deactivates the flow solver solution update. The default is **Off**.

Pressure Reference Location

Provides a choice between using the automatic algorithm ofSTAR-CCM+ for the reference location or providing it manually, as discussed elsewhere.

Automatic Selection

STAR-CCM+ sets the reference location at the cell that is next to the boundary face that has the smallest X, Y, Z location in the domain.

User Specified

The Pressure Reference Points child node is added. Use this property to add one or more reference locations depending on the number of non-contiguous regions.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Grid Sequencing and Expert Initialization

With the Coupled Solver, you can select the grid-sequencing (GS) expert initialization. In essence, this action performs the normal initialization followed by the computation of an approximate inviscid solution to the flow problem. Only flow variables (such as pressure, velocity, and temperature) and species (for mixtures of gases) are initialized.

The **Grid Sequencing** node allows you to set parameters with its properties. It becomes available when the *Method* property of the **Expert Initialization** node is set to **Grid Sequencing**.

When you activate this feature as you manually initialize the flow solution or start a simulation, the GS initialization performs the following steps:

1. A series of coarse meshes is generated.
2. Normal initialization of the flow solution is performed on each of the coarse meshes.
3. Starting with the coarsest mesh:
   - A number of iterations are run to compute an approximate solution on the current mesh.
   - If either convergence or the maximum number of iterations is reached, the solution is interpolated on the next finer mesh and (unless the next mesh is the finest), the previous step is repeated.
4. The process stops when GS has reached the finest mesh level.

GS uses a full implicit incomplete-Newton solution algorithm to compute a first-order inviscid flow solution, which has the following advantages:

- Increased robustness, due to the full implicit algorithm.
- Faster convergence, due to relatively large CFL numbers that can be used.
- Fully automatic coarse-mesh generation and solution.
- An inviscid solution that is a much better approximation of the flow solution than that obtained through normal initialization.
- Faster and more robust convergence of the flow solution, after GS initialization (for both the Coupled and Segregated solvers).

Limitations

GS is not available as an expert initialization method for simulation of heat conduction in solids.

GS, which is regarded as a tool to improve robustness and speed for convergence, is not guaranteed to work in all foreseeable cases. Deactivate it if you encounter situations such as lack of robustness or convergence during the GS process.

Grid Sequencing Properties

Maximum grid levels

Number of coarse grid levels, created automatically, which the grid-sequencing (GS) algorithm uses to compute inviscid solutions, starting from the coarsest mesh.

Maximum iterations per level

Number of solver iterations that are performed during GS, to compute an approximate inviscid solution of the flow problem, on the current coarse mesh.

Convergence tolerance per level

Solution convergence criterion which, together with the maximum number of iterations per level, the solver uses to stop cycling on the current mesh level.

CFL number

Courant-Friedrichs-Levy (CFL) number that the flow solver uses during the grid-sequencing initialization. This number can be different from the Courant number that the solver uses during iterations on the finest mesh (normal run).

Expert Initialization Sub-Solver Properties

The **Expert Initialization** node is a sub-node of the Coupled Implicit solver.

Method

Activates or deactivates grid sequencing for calculation.

None

Deactivates grid sequencing.

Grid Sequencing

Activates grid sequencing. The Grid sequencing node becomes available in the **Expert Initialization** node.

Coupled Explicit Solver

The Coupled Explicit solver controls the solution update for the Coupled Flow model and Coupled Energy model. It is used for explicit integration using a Runge-Kutta multi-stage scheme. The **Coupled Explicit** node represents this solver in the **Solvers** node. To add this node to the simulation tree, select the **Coupled Flow** model node and set the *Integration* property to **Explicit**.

To set the Courant number for this solver, enter the value in the properties. Generally, the *Courant Number* property is chosen to be 1 for unsteady analysis. For steady analysis, a value of 2 can often be used with two residual smoothing iterations. This restriction is much more severe than the implicit spatial integration scheme, although much less storage is required for the same size problem.

Residual smoothing can be employed along with the explicit spatial integration scheme. This approach helps widen the stability margin and allows slightly larger Courant numbers to be used.

Coupled Explicit Solver Properties

Courant Number

Controls the local time-step at each iteration according to Eqn. (1662).

Freeze Flow

When **On**, deactivates the flow solver solution update. The default is **Off**.

Multi-stage Coefficients

Sets the multi-stage coefficient values, see Eqn. (1662). The number of values in the vector indicates the number of stages.

Dissipation-stage Flags

Indicates the stages of the multi-stage scheme on which dissipation (and viscous fluxes if appropriate) is evaluated. The number of these flags must be the same as the number of stages in the multi-stage coefficients.

Residual Smoothing

When **On**, activates residual smoothing.

Residual Smoothing Iterations

Sets the number of residual smoothing iterations using Eqn. (1665).

Residual Smoothing Underrelaxation

Sets the residual smoothing under-relaxation factor ε , see Eqn. (1664). The default value is 0.5.

Pressure Reference Location

Provides a choice between using the automatic algorithm of STAR-CCM+ for the reference location or providing it manually, as discussed elsewhere.

Automatic Selection

STAR-CCM+ sets the reference location at the cell that is next to the boundary face that has the smallest X, Y, Z location in the domain.

User Specified

Adds the **Pressure Reference Points** child node. Use this node to add one or more reference locations depending on the number of non-contiguous regions.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

See Also:

When Is the Pressure Reference Location Required?

Coupled Energy Model Reference

Setting Algebraic Multigrid Parameters

Setting Grid Sequencing Parameters

Setting Convergence Accelerators

Modeling the Equation of State

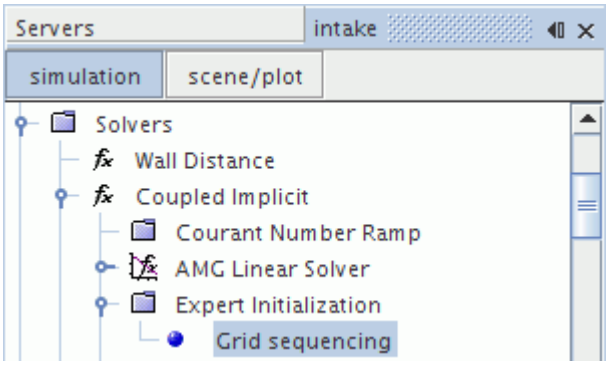
Coupled Flow Controls and General Setup Recommendations

Setting Grid Sequencing Parameters

With the Coupled Solver, you can select the grid-sequencing (GS) expert initialization, which in essence performs the normal initialization followed by the computation of an approximate inviscid solution to the flow problem.

Only flow variables (such as pressure, velocity, and temperature) and species (for mixtures of gases) are initialized.

The **Grid Sequencing** node allows you to set parameters with its properties. It becomes available when the *Method* property of the Expert Initialization node is set to Grid Sequencing.



When you activate this feature when the flow solution is initialized, the GS initialization performs the following steps (this applies to initializing manually or starting a simulation run):

- A series of coarse meshes is generated.
- Normal initialization of the flow solution is performed on each of the coarse meshes.
- Starting with the coarsest mesh:
  - A number of iterations are run to compute an approximate solution on the current mesh.
  - If either convergence or the maximum number of iterations is reached, the solution is interpolated on the next finer mesh and (unless the next mesh is the finest), the previous step is repeated.
- The process stops when GS has reached the finest mesh level.

GS uses a full implicit incomplete-Newton solution algorithm to compute a first-order inviscid flow solution, which has the following advantages:

- Increased robustness, due to the full implicit algorithm.
- Faster convergence, due to relatively large CFL numbers that can be used.
- Fully automatic coarse-mesh generation and solution.
- An inviscid solution that is a much better approximation of the flow solution than that obtained through normal initialization.
- Faster and more robust convergence of the flow solution, after GS initialization (for both the Coupled and Segregated solvers).

Flexibility

You can interrupt GS at any time by clicking the **(Abort)** button next to the status bar. (In this event, if a valid coarse mesh solution exists, it is interpolated onto the finest mesh, and it is fully usable.)

You can change GS settings at any time during the initialization. (The solver acknowledges the change through a message in the *Output* window.)

If GS initialization fails (for example due to aggressive settings), the solver attempts to recover from it. At worst, you can simply get a normal (uniform) initialization. In this case, change the GS settings. For example, choose smaller values for the CFL number or the maximum iterations per level, clear the solution and try the initialization again. Also check that the solver is working with a reasonable number of cells per level to

converge. The default number of maximum grid levels may not be suitable for problems with a relatively small number of cells in the mesh.

Limitations

GS is not available as an expert initialization method for simulation of heat conduction in solids.

GS, which is regarded as a tool to improve robustness and speed for convergence, is not guaranteed to work in all foreseeable cases. Deactivate it if you encounter situations such as lack of robustness or convergence during the GS process.

Grid Sequencing Properties

Maximum grid levels	Number of coarse grid levels, created automatically, which the grid-sequencing (GS) algorithm uses to compute inviscid solutions, starting from the coarsest mesh.
Maximum iterations per level	Number of solver iterations that are performed during GS, to compute an approximate inviscid solution of the flow problem, on the current coarse mesh.
Convergence tolerance per level	Solution convergence criterion which, together with the maximum number of iterations per level, the solver uses to stop cycling on the current mesh level.
CFL number	Courant-Friedrichs-Levy (CFL) number that the flow solver uses during the grid-sequencing initialization. This number can be different from the Courant number that the solver uses during iterations on the finest mesh (normal run).

See Also:

Coupled Flow Solvers

Expert Driver Reference

STAR-CCM+ features an automated convergence control tool within the Coupled Implicit solver. This tool, which is known as the Expert Driver, can be activated through the properties of the **Solution Driver** node that is located beneath the **Coupled Implicit** solver node.

Solution Driver Properties

Driver

Activates or deactivates the Expert Driver for calculation.

Expert driver

Activates the Expert Driver. The Expert Driver node becomes available in the **Solution Driver** node.

None

Deactivates the Expert Driver.

This section covers the following:

- Setting Expert Driver parameters
- Expert Driver properties
- Plotting solver quantities to assess Expert Driver performance

Setting Expert Driver Parameters

The Expert Driver is an automatic convergence control for the Coupled Implicit solver. It is intended for steady-state simulations although a subset of its functionality is also available with unsteady simulations. The Expert Driver improves the robustness of the Coupled Implicit solver, and increases the speed of convergence. However, it is not guaranteed to provide the best solution control in all cases. If the solution slows down or becomes unstable, you can tune (or deactivate) the feature.

The Expert Driver provides the following levels of control:

1. *CFL Control*: Automatically modifies the CFL (Courant) number each iteration to ensure that the AMG linear solver converges. The Courant Number property of the Coupled Implicit solver node sets the maximum CFL number. On the Expert Driver node, properties that are connected to this control are:
   - *CFL Ramp: Start Iteration*, *End Iteration*, and *Start Value*. These properties provide an initial linear ramping of the CFL number.
   - *CFL Recovery Rate*: when the AMG Linear Solver fails to converge, the Coupled solver reduces the CFL number. The Expert Driver can reduce the impact of this reduction by helping to recover the CFL number back to its desired ramp or target value. This property defines this recovery rate.
2. *CFL Limiting*: Adds an extra layer of solver protection by adjusting the CFL number to achieve a balance between the flow solution coupling and the efficiency of the linear system solution. The Expert Driver achieves this balance by controlling the AMG linear solver convergence behavior so that the number of AMG cycles is as close as possible to the target value given in the *CFL Limiting: Target AMG Cycles* property.
3. *Corrections Control*: Computes the under-relaxation factor that gives the quickest rate of decrease in the residual for each coupled flow model. The CFL number control is independent of this property. On the Expert Driver node, properties that are associated with the Corrections Control are:
   - *Corrections Control: Minimum Explicit Relaxation*, *Maximum Explicit Relaxation*, *URF Recovery Rate*, and *Relative Residual Cutoff*.
   - *Corrections Control: Update Frequency*: Unlike *CFL Control* and *CFL Limiting*, which apply to every iteration, the *Corrections Control* is only updated at the frequency this property sets.
4. Expert-Driver Start Iteration and Cutoff Iteration: Automatically activates and deactivates the expert driver after the specified number of iterations.
5. *Enhanced Dissipation*: Significantly improves the convergence and robustness of the Coupled solver during the initial stages of a simulation. The increased numerical dissipation that it provides can lead to inaccuracies in the flow solution. Enhanced Dissipation is available for incompressible flows only. When the simulation reaches the iteration count set by *Enhanced Dissipation: Start transition*, the solver starts to reduce the amount of extra dissipation. The smooth non-linear reduction continues until the simulation reaches the iteration count set by Enhanced Dissipation: End transition. At this point, the extra dissipation is removed completely.

All relevant controls are provided in the properties of the Expert Driver node.

Note: To turn off this property, set both *Start* and *End transition* to 1. In all cases, the solver uses extra dissipation for the first 25 iterations, even when Enhanced Dissipation is deactivated.

Expert Driver Properties

Verbose

When **On**, provides additional output for this value while the simulation is running. This property can be useful for debugging problems when they occur.

CFL Ramp: Start Iteration

Specifies the iteration at which ramping begins.

CFL Ramp: End Iteration

Specifies the iteration at which ramping ends.

CFL Ramp: Start Value

The CFL value before the ramping begins.

Corrections Control

When **On**, optimizes the solution update. The CFL number control is independent of this property.

Corrections Control: Update Frequency

Specifies the number of iterations to perform before invoking this property.

Corrections Control: URF Recovery Rate

Specifies the rate of increase for the explicit relaxation after a solver-controlled reduction. The default value is 1.1, which allows a maximum increase of 10%.

CFL Control

When **On**, modifies the CFL number appropriately at each iteration to ensure the AMG linear solver convergence, and provides a smooth recovery to the desired value when the CFL number is reduced.

CFL Limiting

When On, adjusts the CFL number to strike a balance between the flow solution coupling (as given by the CFL) and the efficiency of the linear system solution.

Expert-Driver Start Iteration

Automatically activates the expert driver after the specified number of iterations. The default is 1.

Expert-Driver Cutoff Iteration

Automatically deactivates the expert driver after the specified number of iterations. The default is 10000000.

Corrections Control: Minimum Explicit Relaxation

Sets the minimum scale to the update with which the *Corrections Control* property optimizes the solution update.

Corrections Control: Maximum Explicit Relaxation

Sets the maximum scale to the update with which the *Corrections Control* property optimizes the solution update.

Corrections Control: Relative Residual Cutoff

As the solution progresses towards convergence, this feature limits the *Corrections Control* property to the initial phases of the simulation as specified here.

Enhanced Dissipation: Start transition

Sets the iteration at which the solver begins to reduce the amount of extra dissipation. The default value is 1.

Enhanced Dissipation: End transition

By the iteration set in this property, no extra dissipation is included. The default value is 100.

CFL Control: Courant Number Recovery Rate

Specifies the rate of recovery to the desired value, in cases where the *CFL Control* property reduces the CFL number.

CFL Limiting: Target AMG Cycles

Controls CFL reduction. If the solver reduces the CFL number (when the AMG linear solver did not converge) and the number of AMG cycles (during the CFL recovery period) is larger than the specified “optimal” (or target) number of AMG cycles, the CFL number is additionally reduced (by 10% at a time).

Expert Driver Plots

STAR-CCM+ provides three field functions that allow you to monitor how well the Expert Driver controls the parameters of the Coupled Implicit solver. The functions are: **CFL Number**, **Explicit Relaxation**, and **Number of AMG Cycles**. Plotting these field functions during a simulation help you optimize the problem setup to achieve fastest possible convergence.

CFL Number

A history plot of the **CFL Number** that shows oscillatory behavior (around a mean value) suggests that you can lower the target **CFL Number**. The *Courant Number* property value of the **Solvers** > **Coupled Implicit** node defines this target. Conversely, if the **CFL number** plot shows a monotone increase towards its peak value, you can increase the **Courant Number**. (In this instance, the increase must be no more than 15–20% at a time).

When the **AMG Linear Solver** fails to converge, the Coupled solver reduces the **CFL Number**. The **Expert Driver** CFL Control helps to recover the **CFL Number** back to the desired ramp or target value. As recovering from a low **CFL Number** takes time, it is better to have an appropriate initial **CFL Number** that does not need to be reduced. Plotting the **CFL Number** helps in this respect.

Explicit Relaxation

If the history plot of **Explicit Relaxation** shows general oscillation, it is likely that the fluid problem being solved has a high degree of non-linearity. In this situation, you are recommended to lower the value of *Corrections Control: Maximum Explicit Relaxation* (and eventually also reduce the value of *Corrections Control: Minimum Explicit Relaxation*).

Mostly high values of **Explicit Relaxation** (close to or above 1.0) indicate rapid and eventless convergence. A higher **CFL Number** could potentially be used.

Number of AMG Cycles

A history plot of the **Number of AMG cycles** shows the variations in computational load inside the linear-system solver (AMG). If the Number of AMG Cycles is much larger than the value that the *CFL Limiting: Target AMG Cycles* property sets, reduce the target **CFL Number** for the Implicit Coupled solver.

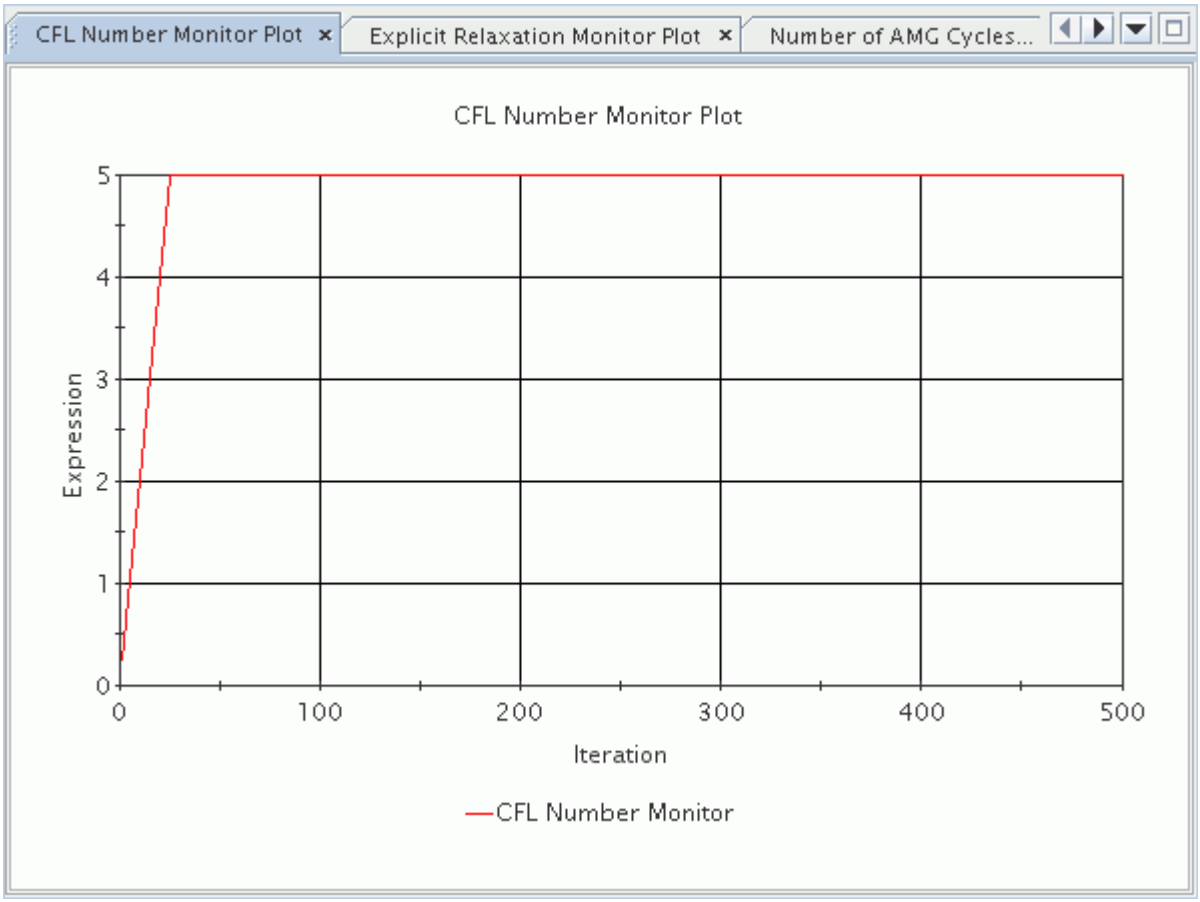
A low **Number of AMG cycles** over an extended part of the simulation suggests that you could accelerate convergence by increasing the target **CFL Number** (but by not more than 15–20% at a time).

Creating the Plots

A plot of the **CFL Number** can be created from an Expression report:

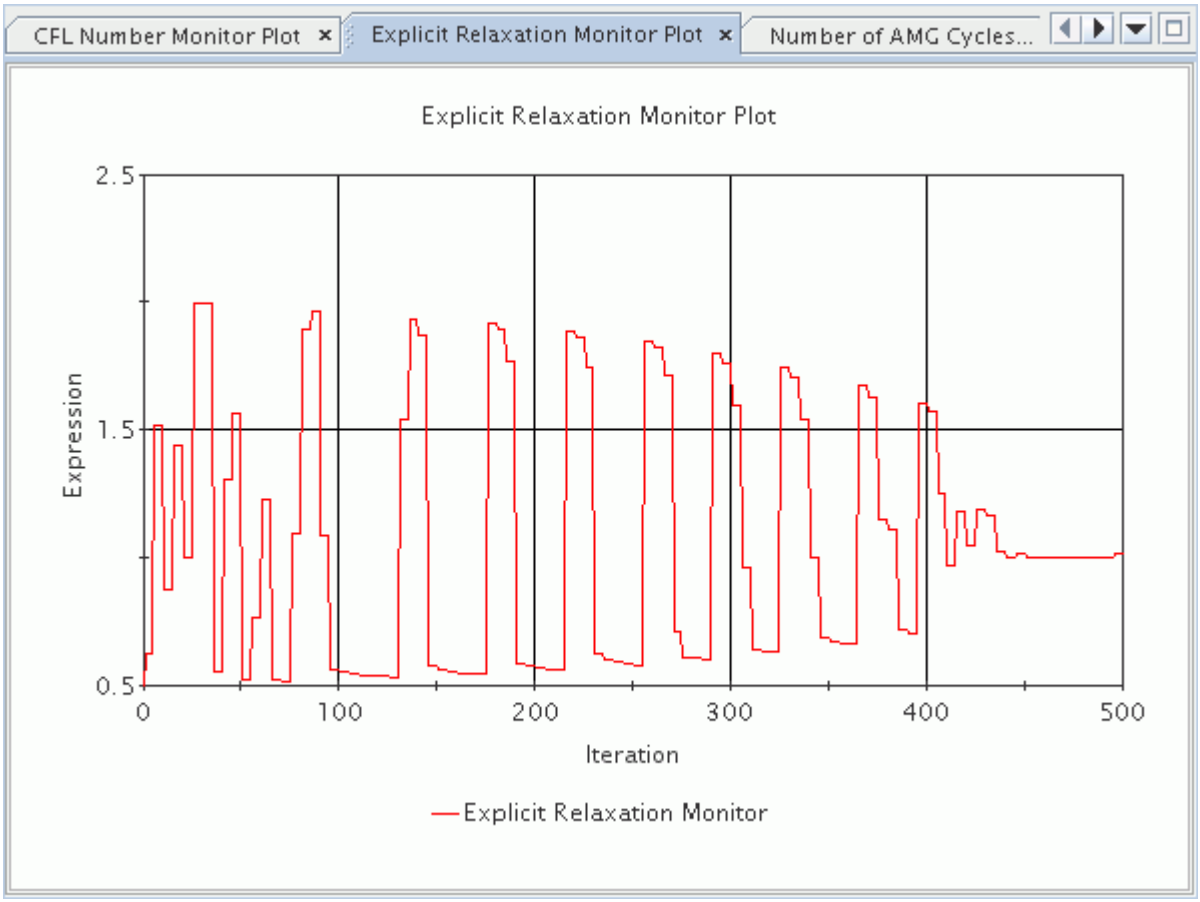
- Right-click **Reports** and select **New Report** > **Expression**.
- Rename the newly created **Expression 1** report to **CFL Number**.
- In the *Properties* window, set the *Definition* property to $CFL. This definition corresponds to the field function, **CFL Number**.
- Right-click **Reports** > **CFL Number** and select **Create Monitor and Plot from Report**.

The *CFL Number Monitor Plot* can be visualized when a simulation is running by right-clicking the **Plots** > **CFL Number Monitor Plot** node and selecting **Open**. An example is shown below.

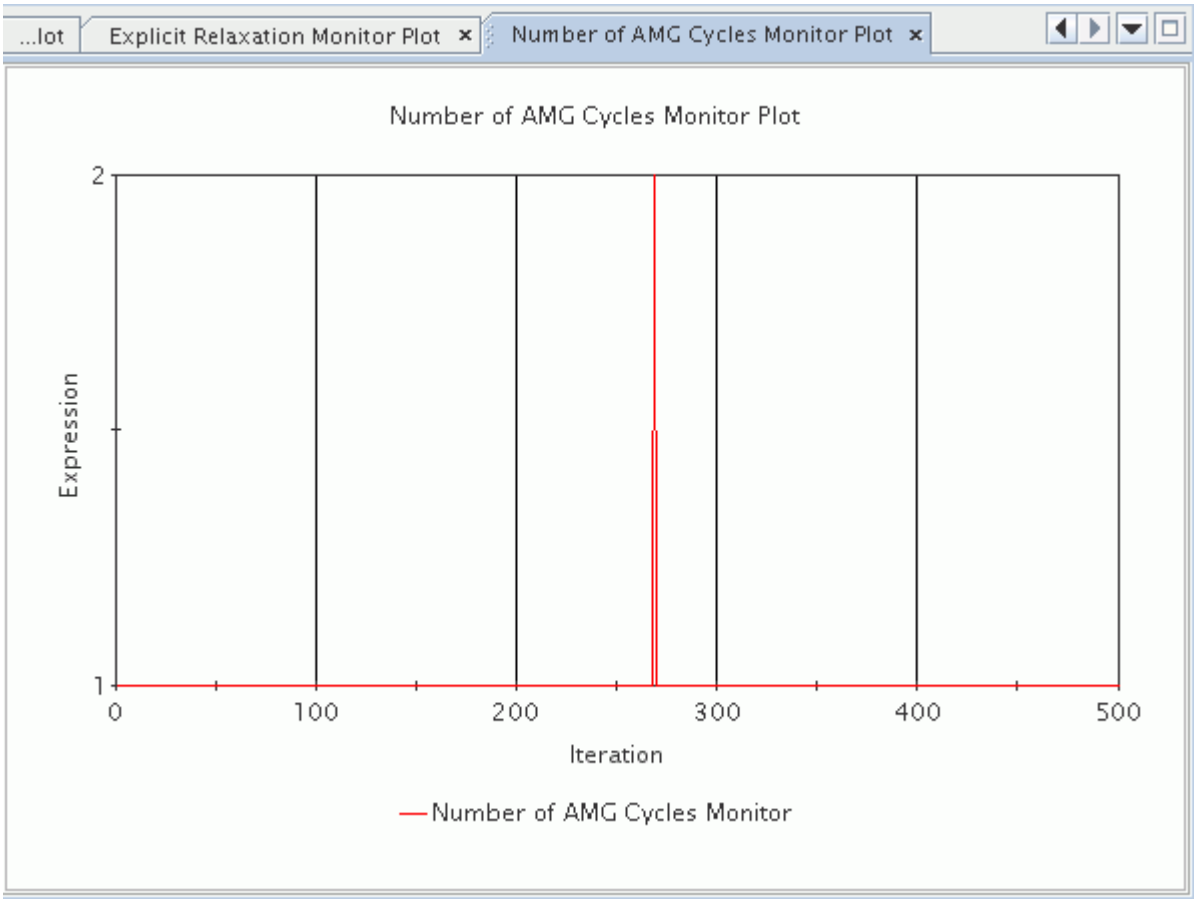


Plots of **Explicit Relaxation** and **Number of AMG cycles** are available only when the **Expert Driver** is activated. The **CFL Number** can be plotted even though **Expert Driver** is not activated.

Create a plot of the **Explicit Relaxation** in the same way as shown for the **CFL Number**. In this instance, however, rename the Expression report to Explicit Relaxation, and set the *Definition* property to `$UrfED`. An example of this plot is shown below.



Create a plot of the **Number of AMG cycles** in the same way as shown for the CFL Number. Rename the Expression report to Number of AMG Cycles, and set the *Definition* property to $NoAMGCyclesED. An example of this plot is shown below.

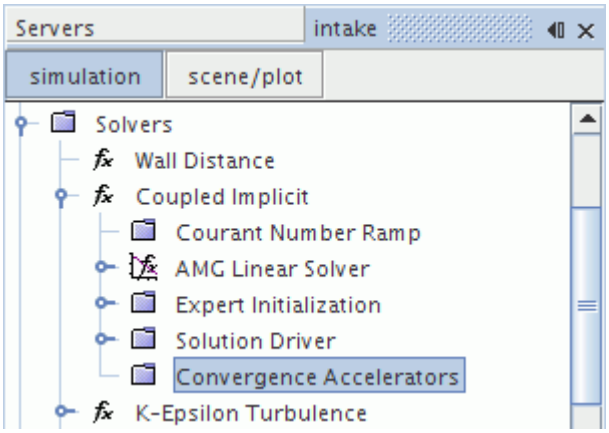


Setting Convergence Accelerators

Some flow problems—for example, cases with a wide range of speeds in the domain (including super- or hypersonic flow)—require a lower CFL number for stability. This can happen when using both constant or flow-dependent material properties.

Typically, the mass-flow convergence is slow for these cases and these simulations can take many iterations to converge.

To speed up these computations, STAR-CCM+ features a mass convergence acceleration tool within the Coupled Implicit solver. This tool, the Continuity Convergence Accelerator, is activated through the **Convergence Accelerators** node that is located beneath the **Coupled Implicit** solver node.



Convergence Accelerators Properties

Convergence Accelerators	Activates or deactivates the Continuity Convergence Accelerator.	
	None	Deactivates the Continuity Convergence Accelerator. (Default)
	Continuity convergence accelerator	Activates the Continuity Convergence Accelerator. The **Continuity Convergence Accelerator** node becomes available in the **Convergence Accelerators** manager.

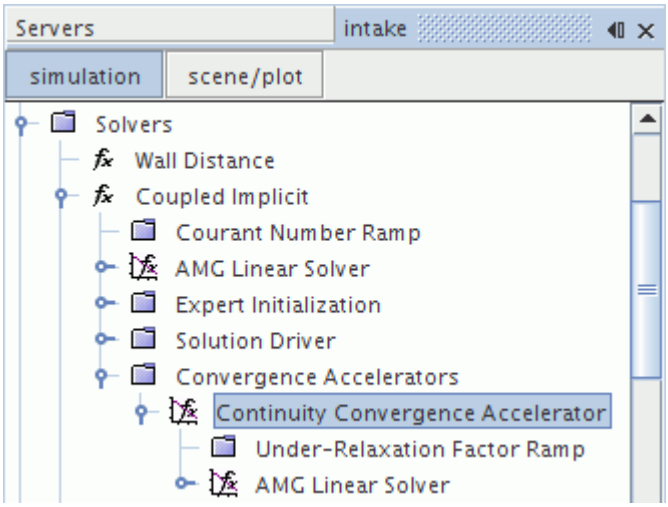
This section provides documentation on the Continuity Convergence Accelerator as follows:

- Setting Continuity Convergence Accelerator parameters
- Continuity Convergence Accelerator guidelines
- Continuity Convergence Accelerator algorithm

Setting Continuity Convergence Accelerator Parameters

The Continuity Convergence Accelerator (CCA) formulates and solves a pressure-correction equation using the Density-based/Riemann flux discretization. The solution of this additional equation provides updates for pressure, velocity, and other flow field variables in such a way that the overall and individual cell mass imbalances are minimized at each iteration.

The **Continuity Convergence Accelerator** node represents this tool in the simulation tree.



Continuity Convergence Accelerator Properties

Enhanced Stability Treatment	When **On**, activates the *Bounded Velocity Corrections* and the *Enhanced Mass-Imbalance Calculations* properties. Sets the AMG Acceleration Method to **None**. The default is **Off**. Switching from **On** back to **Off** does not restore the other properties to their defaults. They must be reset individually if required.
Verbose	When **On**, provides more output for this value while the simulation is running. This property is useful for debugging problems when they occur. The default is **Off**.
Convergence Accelerator Update Frequency	The iteration frequency at which the Continuity Convergence Accelerator is invoked. The default value is 1, meaning that the Continuity Convergence Accelerator is invoked at each iteration.
Under-Relaxation Factor	ω in Eqn. (1676). The default value is 0.1.
CCA Start Iteration	Enables the Continuity Convergence Accelerator at the specified iteration. The default is 1.
CCA Cutoff Iteration	Disables the Continuity Convergence Accelerator after the specified number of iterations. The default is 10000000.
Frozen	When **On**, prevents the Continuity Convergence Accelerator from updating the solution while iterating. The default is **Off**.
Bounded Velocity Corrections	When **On**, makes bounded velocity corrections and therefore limits the pressure correction gradient in the formula for velocity correction, Eqn. (1678). This ensures that velocity corrections are “smooth,” providing bounded velocity updates. **On** is the default.
Enhanced Mass-Imbalance Calculations	When **On**, calculates the final cell mass balances in step 6 of the Continuity Convergence Accelerator Algorithm using the Riemann solver fluxes depending on the flux scheme (see Eqn. (1658) or Eqn. (1661)). This option is more accurate, at the expense of

	computational cost. When **Off**, only corrects the mass flux. See Eqn. (1640). This option is sufficient in most cases and is the default.
Minimum Pressure-Correction Scaling	Minimum value for additional pressure-correction scaling in the presence of strong initial transients, large pressure changes, or rotational motion. The default is 0.1
Pressure Reference Location	Provides a choice between allowing STAR-CCM+ to set the reference pressure location or providing it manually. This property is required for incompressible cases with no pressure boundaries and is ignored otherwise.
Automatic Selection	STAR-CCM+ sets the reference location at the cell that is adjacent to the boundary face that has the smallest X, Y, Z location in the domain. This is the default.
User Specified	The Pressure Reference Points child node is added. Use this method to add one or more reference locations, depending on the number of non-contiguous regions.

Continuity Convergence Accelerator Guidelines

The Continuity Convergence Accelerator does not modify the final converged solution; it only modifies the path to convergence.

The Continuity Convergence Accelerator has a positive effect on convergence for cases with slow mass balance convergence due to strong non-linearity of the flow and the CFL limitations of the solver, for instance when the Coupled solver is used with an Outlet boundary type. However, the Continuity Convergence Accelerator is not a universal remedy to all convergence or robustness issues in the Coupled Implicit solver. Use the Continuity Convergence Accelerator only when mass-balance convergence is slow.

If convergence issues occur with the Continuity Convergence Accelerator, reduce the under-relaxation factor. For highly rotational and/or high speed flows, lower under-relaxation values and careful under-relaxation ramping (from 0.01 or lower) may be required.

Once activated, the Continuity Convergence Accelerator can be deactivated at any point in the simulation without any negative effect on the Coupled Implicit solver.

For maximum convergence efficiency of the Coupled Implicit solver with convergence acceleration, tune the value for the *CCA Under-Relaxation Factor* for the specific flow problem. For difficult problems with stiff numerics, values as low as 0.01 or lower can be used to allow CCA to smoothly correct or reduce the remainder of the continuity equation residual.

If the values for the CCA under-relaxation factor are too large for a model, the result can be strong solution oscillations, lack of convergence, or even divergence. In these cases, reduce the *CCA Under-Relaxation Factor* tenfold and restart the simulation. In these cases, you are advised to use the Expert Driver for the Coupled Implicit solver, where *Corrections Control: Minimum* and *Maximum Explicit Relaxation* need to be reduced with respect to default values, for example using minima of 0.1 and 0.2 and maxima of 0.3 and 0.5.

Use *Enhanced Stability Treatment* for those cases where the Coupled Implicit solver with CCA shows poor convergence. When it is selected, both *Bounded Velocity Corrections* and *Enhanced Mass-imbalance Calculations* are activated and the AMG Acceleration Method is set to **None**. Any of these properties can then be activated or deactivated independently. However, when *Enhanced Stability Treatment* is deactivated and

activated again, *Bounded Velocity Corrections* and *Enhanced Mass-Imbalance Calculations* are also re-activated.

See Also:

Coupled Flow Solvers

Initializing the Solution

Coupled Flow Controls and General Setup Recommendations

The STAR-CCM+ Coupled Flow model and Coupled Implicit Solver make available several parameters which allow control of stability and convergence for practical CFD cases:

- The *Courant Number* (CFL) property of the Coupled Implicit and Coupled Explicit solvers controls the size of the local time-steps that are used in the time-marching procedure these two solvers employ. The Courant number plays the same role as under-relaxation parameters in the segregated solver.

  Generally speaking, for a steady-state simulation, a larger CFL number increases the local pseudo-time step size and produces faster convergence. Thus, use the largest CFL number possible while still ensuring that the solver remains within the bounds of stability. (This means that the solver does not start reducing corrections, or hitting temperature and pressure limits, and that the residual has a healthy downward trend.)
- The *Explicit Relaxation* (an expert property of the Coupled Flow Model) is a scaling factor that is used to relax all corrections explicitly before they are applied to the variables. It is intended to add stability in cases where the solver's first-order linearization of a second-order discretization has trouble. The default value is 1.0, which results in 100% of the corrections being applied. Use a value less than 1.0 to limit the applied corrections explicitly.
- The *Positivity Rate Limit* (an expert property of the Coupled Flow Model) defines the amount that temperature, a positive definite quantity, is allowed to decrease when corrections are applied at each iteration. It is expressed as a percentage of the current value.

  The default is 0.2, which says temperature is not allowed to decrease by more than 20% (there is no upward limit). If temperature corrections are reduced to meet the Positivity Rate Limit, then all other variable corrections are equally scaled in an attempt to keep everything changing at the same rate.

  Note: The last two Coupled Flow model expert properties are linked, in that the Explicit relaxation factor is applied to the corrections first, then the Positivity Rate Limit condition is checked. It is possible that the Explicit relaxation factor could reduce temperature corrections such that the Positivity Rate Limit does not get invoked where it otherwise would.
- Deactivate the *Preconditioning Enabled* option (an expert property of the Coupled Flow Model) when you are trying to capture unsteady phenomena at small time scales (for instance, in an aeroacoustics analysis). Deactivate this property when the *Integration* property is set to either **Implicit** or **Explicit**.

Setup Recommendations

- The default CFL number for the Coupled Implicit solver is 50.0.
- You can often increase the CFL number value to 100, 200 or even more, depending on the physical and numerical difficulty of the actual flow problem being solved and grid quality. Laminar flow and good grid quality allow even higher values and very rapid convergence. When using the Coupled Explicit Solver, limit the CFL number to values near unity.
- Turbulent flows require lower CFL numbers and more iterations than laminar flows, since equations for turbulence quantities are solved in a segregated manner outside the coupled solver. However, even for turbulent flows, the CFL number can be increased beyond 1000, especially when the mesh is refined and the simulation is initialized with a coarser mesh solution.

- In many situations, it is helpful to perform some iterations in the segregated solver before switching to the coupled solver. This procedure provides a better initial solution and allows you to set a higher Courant number in the coupled solver.
- If an approximate initial solution is used (uniform values), smaller CFL number values are required at startup.
- You can employ the Linear Ramp capability of the Coupled Implicit solver, to automate desired change of the CFL number during the simulation.
- A low CFL number value of 0.1, or even lower, may be required for startup of some difficult problems (for example, hypersonics or combustion). To ensure that the solver stability is maintained during the entire simulation, use the Linear Ramp to increase the CFL number based on iteration count.
- Depending on the problem (such as very high-speed flow or mesh quality issues) you could use an Explicit relaxation factor as low as 0.2 or 0.25. Very difficult cases may require a value of 0.15 to be able to maintain non-linear stability.
- Typically an Explicit relaxation factor of 0.75–0.85 helps convergence for most problems, without greatly affecting the speed of convergence for those cases that do not require any explicit relaxation.
- Besides the CFL number and the Explicit relaxation factor, a practical way to keep hypersonic cases within physical “bounds” is to use a lower value for the Positivity Rate Limit. The default value is 0.2, although for extreme hypersonic cases it may be necessary to use values as low as 0.05.
- For time accurate simulations, choose the time step to keep the CFL number around 1.0 on average. In this case, it is possible to set under-relaxation factors to 1.0 for all transport equations. The CFL value can be larger where the mesh is fine, and smaller where cells are large.

Mesh Quality Considerations

- Mesh quality does influence the quality of the numerical results, irrespective of the setup: coupled vs. segregated solver, discretization schemes...
- On a mesh of a given quality and sufficient fineness, higher-order schemes yield more accurate results than lower-order ones. For a given cell count, aim to optimize the grid quality because a better mesh gives a more accurate solution regardless of the differencing schemes that are used and other parameters.
- With a poor-quality mesh, the gradient limiter is invoked more often, thus affecting its nominal second-order accuracy. Grid skewing is an important contributing factor (much more than grid stretching) to the loss in nominal accuracy of the solution. Grid design (distribution of cell size, local refinement using feature edges, boundary region, or volume shapes) is important in maximizing the accuracy for a given effort. Two grids with the same number of cells may lead to discretization errors that differ by an order of magnitude. Thus, improving mesh quality is a top priority.
- When solving steady-state problems on fine meshes, start with a much coarser mesh, and then successively refine the mesh by reducing the base size. For example:
  - Design the desired mesh for the highest cell count affordable.
  - Increase the base size by a factor of 8 and start the computation with this coarse mesh.

    You may require lower under-relaxation values for the coarsest mesh, but iterations are fast and this does not increase the computing effort much.
  - After the convergence criteria are satisfied or maximum specified number of iterations is reached, halve the mesh base size, remesh, and restart the simulation. Repeat until the final mesh size is reached.

In this way, you obtain faster convergence on finer grids by providing a good initial solution. Usually, one needs only a third or a quarter of the number of iterations that would be needed if starting with an initial guess, like constant or zero values.

In addition, you obtain solutions on a series of grids which are systematically refined (same design, only base sizes reduced), which allows for an estimate of discretization errors using Richardson extrapolation. For example, if base size is reduced by a factor of 2 and second-order discretization is used, discretization errors on the finest mesh are equal to one third of the difference between solutions on the finest and next coarser grid. For base size reduction by a factor of 1.5, the errors amount to about 80% of the difference in the two solutions.

Troubleshooting

Flow solution diverges during initial iterations

If the residuals show that the solution is diverging during the first several iterations, and there is no sign that they are decreasing, this indicates a problem.

- Make sure that the overall setup is correct. Some things to look for are:
  - Mesh extents for viscous flow
  - Mesh quality
  - Boundary conditions for both flow and turbulence, if turbulent flow is activated
- Try reducing the Courant number.
  - It is possible to make the Courant number as small as desired to get the solution started. If it seems necessary to run at a Courant number significantly less than 1 for more than just a few iterations, there is most likely something wrong with your solution setup.
  - Once the residuals start coming down again, it is possible to increase the Courant number.

Warnings occur of corrections being limited

Because temperature is a positive definite quantity, STAR-CCM+ tries to limit the corrections so that large changes do not tend to drive this quantity negative.

- It is acceptable for these warnings to occur during startup, as long as they cease after several iterations. If they do not, it is necessary to reduce the Courant number.

See Also:

Coupled Flow Solvers

Segregated Flow Guidelines

STAR-CCM+ solvers provide several controls that help simulations remain stable and achieve a converged solution.

Under-Relaxation Factors

The following guidelines are helpful for setting the under-relaxation parameters for the velocity solver and the pressure solver:

Steady State Simulations

- The default pressure and velocity under-relaxation factors are conservative. They lead to a converged solution in most cases, including when the grid is poor or the equations are stiff.
- Optimal values of under-relaxation parameters are problem-dependent. The grid aspect also affects the optimum values. In a batch job or any other computation where the progress cannot be monitored, it is safer to use the default values.
- When performing similar calculations, check how the simulation reacts to the variation of under-relaxation factors. Since convergence behavior is similar on different grid densities, it is worthwhile to perform a test on a coarse grid to determine the optimum values for the case. These optimum values can then be applied to a calculation on a denser grid.
- It is an accepted rule of thumb that optimum under-relaxation factors for velocity and pressure add up to 1 for steady-state simulations. If the grid is highly non-orthogonal, keep the under-relaxation factor for pressure lower than this rule suggests. (0.1 is a typical value.)
- If the residuals are converging well, it is acceptable to try increasing the under-relaxation factor for velocity (up to 0.9), and decreasing the pressure under-relaxation factor (to 0.1).
- For compressible flows, you must sometimes decrease the under-relaxation factor for velocity to 0.5 and possibly the pressure under-relaxation factor to 0.1.
- For porous media, you must sometimes to decrease the under-relaxation factor for velocity to 0.5 and possibly the pressure under-relaxation factor to 0.1.
- When solving steady-state problems on fine meshes, start with a much coarser mesh, and then successively refine the mesh by reducing the base size. For example:
  - Design the desired mesh for the highest cell count affordable.
  - Increase the base size by a factor of 8 and start the computation with this coarse mesh.

    You sometimes need lower under-relaxation values for the coarsest mesh, but iterations are fast and this requirement does not increase the computing effort much.
  - After the convergence criteria are satisfied or maximum specified number of iterations is reached, halve the mesh base size, remesh, and restart the simulation. Repeat until the final mesh size is reached.

  In this way, you obtain faster convergence on finer grids by providing a good initial solution. Usually, one needs only a third or a quarter of the number of iterations that would be needed if starting with an initial guess, like constant or zero values.

In addition, you obtain solutions on a series of grids which are systematically refined (same design, only base sizes reduced), which allows for an estimate of discretization errors using Richardson extrapolation. For example, if you reduce base size by a factor of 2 and use second-order discretization, discretization errors on the finest mesh are equal to one third of the difference between solutions on the finest and next coarser grid. For base size reduction by a factor of 1.5, the errors amount to about 80% of the difference in the two solutions.

Transient Simulations

- For transient simulations, provided a reasonably well-resolved time-step is chosen, you can often increase the under-relaxation factor for velocity as high as 1.0 and for pressure up to 0.9 (for example, for LES-type simulations).
- Under-relaxation factors of 0.9 (velocities and scalars) and 0.5 (pressure) work well in most cases.
- These values are higher than in steady-state cases since the contribution of the transient term to the discretized equations has the same effect as under-relaxation.

Transient Time-Step and Inner Iterations

The transient phenomenon being simulated generally governs the physical time-step.

Physical Time-Step

For periodic flow, an order of 50–100 time steps per period is appropriate.

The convective Courant number is a helpful indication for selecting the time step size: for time-accurate simulations, the convective Courant number should be 1.0 on average in the zone of the interest. This value implies that the fluid moves by about one cell per time step.

For flows with free surfaces, if a second-order scheme for time integration is used, the Courant number must be less than 0.5 in all cells. If a free surface moves more than half a cell per time step, the High-Resolution Interface-Capturing scheme can lead to overshoots or undershoots, and eventually to divergence.

When computing wave propagation, use more than twice as many time steps per wave period than cells per wavelength.

Maximum inner iterations stopping criterion

Smaller physical time-steps generally mean that the solution is changing less from one time-step to the next, so that fewer inner iterations are required. There is clearly an optimal balance of time-step size, under-relaxation factors, and the number of inner iterations for a given problem and desired transient accuracy.

If you are performing a time-accurate simulation with adequate time step and under-relaxation parameters, 5 iterations per time step are usually sufficient. More iterations per time step (an order of 10) and lower under-relaxation factors (around 0.8) are sometimes required for tough coupled problems like flow with free surface and motion of a floating body. VOF simulations with melting and solidification also require many inner iterations.

If the convergence of iterations is slow, it is a sign that the time step is too large and will result in significant temporal discretization errors. It is better to reduce the time step than to do many iterations within a large time step. For example, halving the time step allows the number of inner iterations to be reduced, so that the computing effort is not much higher while the accuracy of solution is increased.

Contents:

Troubleshooting

Troubleshooting

Flow solution diverges during initial iterations

If the momentum and continuity residuals show solution divergence during the first several iterations and the residuals show no signs of decreasing, this indicates a problem.

- Make sure that the overall setup is correct. Some things to look for are:
  - Mesh extents for viscous flow
  - Mesh quality
  - Boundary conditions for both flow and turbulence, if turbulent flow is activated
- If the flow is incompressible, a slight reduction of the under-relaxation factors for pressure and velocity can stabilize the solution. It is also appropriate to try setting the model properties to use a first-order convection scheme.
- If the flow is compressible, reduce the under-relaxation factor for energy (say to 0.3). Also, the under-relaxation factor for pressure can be set to a small number (say 0.01), for just a few iterations before resuming iterating with the default value of 0.3.

Warnings occur of pressure corrections being limited for compressible flow

For compressible flow, the absolute pressure must be a positive definite quantity. STAR-CCM+ tries to limit the pressure corrections so that large pressure changes do not drive the absolute pressure negative.

- It is acceptable for these warnings to occur during the startup for compressible flow, as long as they cease after several iterations. If they do not, it is necessary to reduce the under-relaxation factor for pressure. This factor can be set to a small number (say 0.01), for just a few iterations, before resuming iterating with the default value of 0.3.
- Ensure that the under-relaxation factor for energy is set to an appropriate value (say 0.3).

Warnings occur about reverse flow at a flow split boundary

In STAR-CCM+, flow split outlet boundaries are not designed to permit recirculation.

- If warnings occur during initial convergence but cease after several iterations, there is no cause for concern.
- If warnings persist, there are two options:
  - Rebuild the mesh but extend the outlet region in such a way as to minimize the risk of recirculation.
  - Change the boundary type to a pressure outlet.

Warnings occur about reverse flow at a pressure outlet boundary

In STAR-CCM+, pressure outlet boundaries are designed to permit recirculation, although a unique solution can not exist resulting in convergence problems.

- Recirculation causes a non-uniform pressure distribution at the boundary.

- If a uniform pressure distribution is desired, rebuild the mesh with the outlet region extended in such a way as to prevent recirculation.
- Convergence problems due to unsteady recirculation are indicated by the number of faces at which reversed flow occurs changing from one iteration to the next. (The number of faces is printed in the warning messages in the output window.) If unsteady recirculation occurs, extend the outlet region in such a way as to prevent it.

Energy solution diverges during initial iterations

If the energy residuals show solution divergence during the first several iterations and the residuals show no signs of decreasing, this scenario indicates a problem.

- Make sure that the overall setup is correct. Some things to look for are:
  - whether the flow field is diverging; if so, refer to the Segregated Flow model troubleshooting
  - mesh quality
  - boundary conditions for both flow and energy.
- A reduction of the under-relaxation factor for energy can stabilize the solution.

Energy solution converges slowly

The under-relaxation factor can be adjusted to control the convergence rate.

- The default *Under-Relaxation Factor* value of 0.9 is aggressive. It is not advisable to use a larger value than 0.9 for flow simulations.
- For compressible flow problems, the under-relaxation factor is normally kept in the range 0.1-0.5.

Segregated Flow Model Reference

The segregated flow model invokes the segregated solver which solves each of the momentum equations in turn, one for each dimension.

The linkage between the momentum and continuity equations is achieved with a predictor-corrector approach.

The complete formulation can be described as using a colocated variable arrangement (as opposed to staggered) and a Rhie-and-Chow-type pressure-velocity coupling combined with a SIMPLE-type algorithm.

This model has its roots in constant-density flows. Although it can handle mildly compressible flows and low Rayleigh number natural convection, it is not suitable for shock-capturing, high Mach number, and high Rayleigh-number applications.

Segregated Flow Model Reference

Model Name	**Segregated Flow**	
Theory	See Fluid Flow in the Theory Guide.	
Provided By	**[physics continuum]** > **Models** > *Flow*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Segregated Flow**	
Requires	Pre-requisite selections: *Material*: one of **Gas**, **Liquid**, **Multi-Component Gas**, **Multi-Component Liquid**, **Eulerian Multiphase** > **Volume of Fluid (VOF)** (For multi-component materials) *Reaction Regime*: any option	
Properties	Key properties are: *Convection*. See Segregated Flow Properties.	
Activates	Physics Models	Model Groups *Equation of State* *Viscous Regime*
	Model Controls (child nodes)	**Bounded Differencing**, when *Convection* is set to **MUSCL 3rd-order/CD**. See Bounded Differencing.
	Materials	**Dynamic Viscosity**. Other properties depend on the Equation of State.
	Reference Values	**Reference Pressure**. See Reference Values.
	Initial Conditions	**Pressure**, **Velocity**. See Initial Conditions.
	Boundary Settings	Inflow: **Velocity**, **Mass Flow Rate**, **Total Pressure** Outflow: **Pressure**, **Target Mass Flow Rate**, **Average Pressure**, **Pressure Jump** Domain: **Shear Stress Specification**, **Tangential Velocity Specification** For a complete list, see Flow Boundaries Reference.

	Region Settings	Region sources are available for mass and momentum. See Region Values.
	Solvers	Segregated Flow Velocity Pressure See Segregated Flow Solvers.
	Monitors	**Continuity** **X-, Y-, and Z-Momentum**
	Field Functions	See Flow and Energy Field Functions.

Segregated Flow Properties

Convection

Sets the discretization scheme that STAR-CCM+ uses for computing the convection flux on a cell face in appropriate transport equations. More information is given in the related topic for the Convection Term:

- **1st-Order**: First-order upwind scheme. This scheme scales the transported quantity by the upstream or downstream mass flowrate depending on flow direction. Only use when a higher-order scheme fails to give convergence, or in order to obtain an initial solution before switching to a higher-order scheme.
- **2nd-Order**: Second-order upwind scheme. This scheme introduces linear interpolation of cell values on either side of the upstream or downstream face. Using this scheme can lead to poorer convergence properties, but gives accuracy as good as or better than the first-order scheme.
- **Central**: Central-differencing scheme. This scheme is second-order accurate but is prone to dispersive error and stability problems. However, central-differencing is useful in large eddy simulation (LES) where upwind schemes accelerate the rate at which turbulent kinetic energy decays.
- **Bounded-Central**: Bounded-central differencing. This scheme is recommended for large eddy simulation (LES) of complex turbulent flows. This scheme applies a boundedness criterion that makes the scheme more robust than central-differencing alone.
- **Hybrid**: Hybrid second-order upwind/central. This scheme blends second-order upwind and central differencing, and is appropriate for simulations that use detached eddy simulation (DES).
- **Hybrid-BCD**: Hybrid second-order upwind/bounded-central. This scheme is the default scheme for detached eddy simulation (DES).
- **MUSCL 3rd-order/CD**: Hybrid MUSCL third-order/central-differencing. Adds a **Bounded Differencing** node to the flow model node.

If you choose either the **Bounded-Central** or **Hybrid-BCD** schemes, set the *Upwind blending factor* in the **Bounded Differencing** node.

Minimum Absolute Pressure

Minimum allowable absolute pressure (for compressible flows).

Positivity Rate Limit

Maximum allowable pressure correction update, expressed as a fraction of the difference between the current pressure and the minimum pressure. Applies only to compressible flows. The default value is `0.2`,

meaning that any negative pressure correction is limited to 20% of current pressure minus minimum pressure.

Secondary Gradients

There are two sources of secondary gradients in STAR-CCM+ flow solvers:

- boundary secondary gradients for diffusion
- interior secondary gradients for diffusion at cell faces

Use this property to control which gradients are included in the solver. **On** gives both gradients while **Off** excludes them. **Interior Only** and **Boundaries Only** select the corresponding gradients.

Limiting Acoustic-CFL

The limiting acoustic CFL, applied for unsteady simulations, to scale the mass flux dissipation. Larger values cause more local dissipation and damp more spurious oscillations, but at the expense of accuracy. Make sure that non-constant values vary smoothly in space. When set to **Per-Model**, you specify the value in the corresponding child node. When set to **Per-Region**, you specify the value within the **Physics Values** node for each region.

Delta-V Dissipation

When **On**, this property enables an extra "delta-v" dissipation (in addition to Rhie-Chow) that resembles a term in the coupled flow solver. This dissipation comes from the right-most term in Eqn. (1629). This enhancement allows a robust treatment of porous media and similar large body forces by making the segregated solver behave more like the coupled solver, and fixing checker-boarding at a porous/non-porous interface. This property is deactivated by default.

Flow Boundary Diffusion

When activated, this property includes the flow-boundary diffusion fluxes (or viscous fluxes for flow models) as given by [eqnlink]. This property is activated by default.

Unsteady Flux Dissipation Corrections

When activated, this property includes mass flux dissipation corrections for unsteady simulations, modifying the coefficients $\overline{a}_0$ and $\overline{a}_1$ in Eqn. (1628), based on CFL. This property is deactivated by default.

Reference Values

Reference Pressure

The pressures referred to by the flow models in STAR-CCM+ are implicitly gauge pressures. This is important for minimizing roundoff errors, particularly for incompressible problems where pressure changes might be small.

If the Constant Density model is being used, the operating pressure has no bearing on the calculation. However, when the Ideal Gas model is chosen, the absolute pressure is used to obtain the density so that the operating pressure must be specified appropriately.

Initial Conditions

Pressure (initial condition)

Specify the working pressure in Pa. For an initial condition, the outlet pressure is reasonable to use for the initial value of pressure. If there are no pressure boundaries that are attached to the given continuum, the initial value is arbitrary.

Velocity

Relative Velocity

Sets the velocity vector in the chosen *Coordinate System*. When relative, the vector is defined within the reference frame applied to the boundary.

Contents:

Segregated Flow Solver
Bounded Differencing
Setting Algebraic Multigrid Parameters
Modeling Fluid Energy Using a Segregated Approach

Segregated Flow Solvers

The Segregated Flow solver controls the solution update for the Segregated Flow model according to the SIMPLE algorithm. For improved solution times, activate *Continuity Initialization* on the solver node.

Two additional solvers :

- Velocity solver
- Pressure solver

For each of these solvers, you can set under-relaxation factors and algebraic multigrid parameters.

Segregated Flow Solver Properties

Freeze Flow	When **On**, deactivates the flow solver solution update while allowing the solver to continue responding to other solvers. This property is **Off** by default.
Enable Enhanced Stability Treatment	When **On**, activates the **Pressure Corrections** and **Velocity Corrections** expert properties to increase the robustness of iterations for each time-step.This property is **Off** by default, making these expert properties unavailable.

Segregated Flow Solver Expert Properties

Pressure Corrections: Bad Cell Minimum Scaling

Sets a scale factor for pressure corrections, for bad cells, and for cells that have a volume change lower than the acceptable limit. See Volume Change. The default value is 0.8.

Pressure Corrections: Acceptable Cell Volume Change

Defines a limit of acceptable volume change for the solver enhanced robustness treatment. See Volume Change. The default value is 0.001.

Velocity Corrections: Maximum Unlimited Velocity

Defines the value below which there are no damping corrections to velocity magnitude. The default value is 20 m/s.

Velocity Corrections: Acceptable Velocity Increase Rate [<1]

Defines an acceptable limit that reduces the rate of increase of velocity magnitude due to pressure corrections. (There is no limit for rate of decrease.) See the SIMPLE Solver Algorithm. The default value is 0.15.

Continuity Initialization

When this property is **On**, the solver uses the optimized values that are given by the initialization as initial conditions to solve the flow equations. The improvement of the initial conditions due to initialization helps to accelerate the convergence of the solution. Adds a node on which to set Continuity Initialization properties. When it is **Off**, the solver uses the initial conditions that you have specified to solve the flow equations.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

For information about other properties within the *Expert* category, see Expert Properties.

Continuity Initialization Properties

Continuity Initialization Iterations	Specifies the number of iterations that are performed during the initialization process. The default value of 3 can be increased for further optimization of the initial conditions.
Continuity Initialization Tolerance	The convergence tolerance during the flow field initialization. Its value overwrites the convergence tolerance for the Pressure Solver during the initialization cycle. When the initialization is complete, the convergence tolerance of the pressure AMG Linear solver is set back to the value

	you have specified. The default value for the **Continuity Initialization Tolerance** is 1E-6.

Velocity Solver

The velocity solver controls the under-relaxation factor and algebraic multigrid parameters for the momentum equations. More specifically, it solves the discretized momentum equation to obtain the intermediate velocity field.

Properties

Under-Relaxation Factor	In order to promote convergence, this property is used to under-relax changes of the solution during the iterative process. For steady flows, the default value is 0.7; for unsteady flows, the default value is 0.8. The under-relaxation is implemented in an implicit manner. For more information, see Finite Volume Discretization.

Expert Properties

Dynamic Local Under-Relaxation	For unsteady flows, the time-derivative term in the momentum equation acts as a form of under-relaxation for the system of equations. Because of this, the user-defined under-relaxation factor can cause a higher level of under-relaxation than the solution requires. Dynamic Local Under-Relaxation causes the solver to compare the relaxation provided by the time derivative term of the momentum equation with the user-defined under-relaxation. The solver then modifies the actual under-relaxation applied based on the result of the comparison.	
	Activated	Solver compares the two sources of under-relaxation and modifies the final under-relaxation accordingly.
	Deactivated	Solver applies the user-defined under-relaxation factor without modifying it. (The default.)

Pressure Solver

The pressure solver controls the under-relaxation factor and algebraic multigrid parameters for the pressure correction equation. More specifically, it solves the discrete equation for pressure correction, and updates the pressure field.

Properties

Under-Relaxation Factor	At each iteration, this property governs the extent to which the newly computed solution supplants the

	old solution. This quantity is ω in Eqn. (1643). For steady flows, the default value is 0.3; for unsteady flows, the default value is 0.2.

Expert Properties

Pressure Reference Location	Provides a choice between using the automatic algorithm of STAR-CCM+ for the reference location or providing it manually, as discussed elsewhere.	
	Automatic Selection	STAR-CCM+ sets the reference location at the cell that is next to the boundary face with the smallest X, Y, Z location in the domain.
	User Specified	The Pressure Reference Points child node is added. Use this method to add one or more reference locations depending on the number of non-contiguous regions.

See Also:

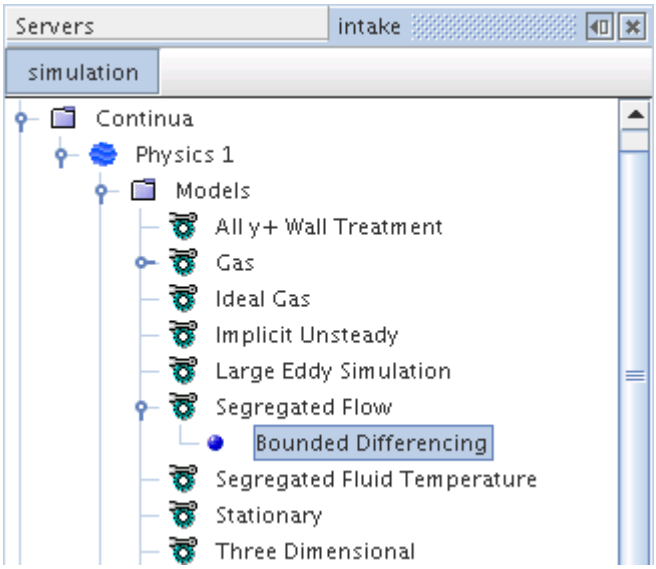
When Is the Pressure Reference Location Required?

Bounded Differencing

The Bounded Differencing sub-node lets you specify parameters for the bounded differencing when you select any of these convection schemes: Bounded-Central (flow solver enabled for LES), Hybrid-BCD (flow solver enabled for DES), or MUSCL 3rd-order/CD.

The **Bounded Differencing** node is activated under any of these conditions:

- When the *Convection* property of the Segregated Species model is set to **MUSCL 3rd-order/CD**.
- When the *Convection* property of the Segregated Fluid Temperature model is set to **MUSCL 3rd-order/CD**.
- When the *Convection* property of the Segregated Multiphase Temperature model is set to **MUSCL 3rd-order/CD**.
- When the *Convection* property of the Segregated Fluid Enthalpy model is set to **MUSCL 3rd-order/CD**.
- When the *Convection* property of the Segregated Flow model is set to **Bounded-Central** (with LES), **Hybrid-BCD** (with DES), or **MUSCL 3rd-order/CD** (with either).
- When the *Discretization* property of the Coupled Flow model is set to **Bounded-Central** (with LES), **Hybrid-BCD** (with DES), or **MUSCL 3rd-order/CD** (with either).



Properties

Upwind blending factor	Specifies the proportion of upwind differencing, ς_{ubf} related to Eqn. (1586). The default value of this property is 0.15 for **Bounded-Central** or **Hybrid-BCD**. This value provides the most robustness for the scheme. Reducing it would, in principle, increase accuracy. However, unless you are thoroughly familiar with the theoretical aspects of bounded differencing, do not change this property. The default value reflects optimization for accuracy and performance. The default value of this property is 1.0 for **MUSCL 3rd-order/CD**. Use lower values to reduce the numerical dissipation of the scheme.

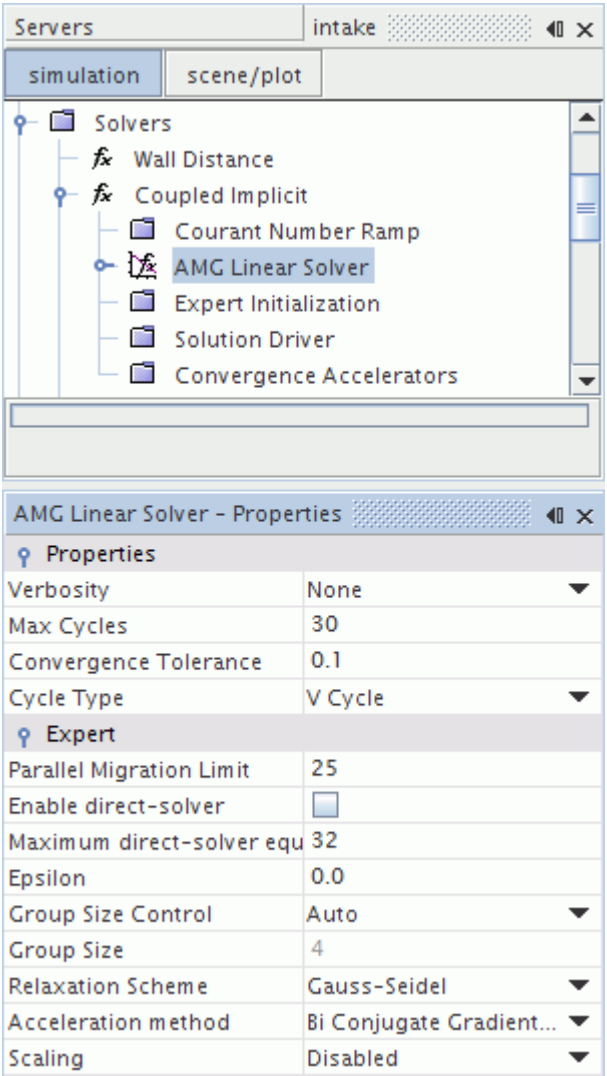
See Also:

Detached Eddy Simulation (DES)

Setting Algebraic Multigrid Parameters

To set the algebraic multigrid parameters for either the pressure solver or the velocity solver:

- Open the node for that solver under the **Coupled Implicit** node in the **Solvers** node.
- Open the node for the **AMG Linear Solver**.
- Enter the algebraic multigrid parameters using the *Properties* window.



In most situations, it is not necessary to change these parameters.

See Also:

Coupled Flow Solvers

AMG Linear Solver Reference

AMG Linear Solver Reference

Modeling Fluid Energy Using a Segregated Approach

This section describes the Segregated Fluid Energy models and the Energy solver. Instructions are provided for selecting a Segregated Fluid Energy model and setting model properties.

A troubleshooting section is provided and the formulation of the equations and discretization approach is also presented.

There are three Segregated Fluid Energy models, which are companion models to the Segregated Flow model:

- Segregated Fluid Enthalpy
- Segregated Fluid Temperature
- Segregated Fluid Isothermal

The first two solve the total energy equation in a continuum using a segregated formulation, while the third uses a constant setting for temperature.

Segregated Fluid Enthalpy

The Segregated Fluid Enthalpy model solves the total energy equation with chemical thermal enthalpy as the solved variable. Temperature is then computed from enthalpy according to the equation of state.

This model is recommended for any simulation involving combustion. It is required with the non-adiabatic PPDF equilibrium combustion models.

Properties

Convection	For guidance on selecting a convection scheme, see Segregated Flow Properties.	
	1st-order	Selects the first-order convection.
	2nd-order	Selects the second-order convection.
	MUSCL 3rd-order/CD	Selects the hybrid MUSCL 3rd-order/CD convection scheme. Adds a **Bounded Differencing** sub-node to the **Segregated Fluid Enthalpy** node.

Expert Properties

Flow Boundary Diffusion	When **On**, activates the flow-boundary diffusion fluxes that are given in Eqn. (1593). This option is **On** by default. A realistic temperature value must be set on flow boundaries when this option is activated.	
Secondary Gradients	Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces.	
	On	Include both secondary gradients.
	Off	Exclude both secondary gradients.
	Interior Only	Include the interior secondary gradients only.
	Boundaries Only	Include the boundary secondary gradients only.

Segregated Fluid Temperature

The Segregated Fluid Temperature model solves the total energy equation with temperature as the solved variable. Enthalpy is then computed from temperature according to the equation of state.

This model is appropriate for simulations that do not involve combustion.

Properties

Convection	For guidance on selecting a convection scheme, see Segregated Flow Properties.	
	1st-order	Selects the first-order convection.
	2nd-order	Selects the second-order convection.
	MUSCL 3rd-order/CD	Selects the hybrid MUSCL 3rd-order/CD convection scheme. Adds a Bounded Differencing sub-node to the **Segregated Fluid Temperature** node.

Expert Properties

Flow Boundary Diffusion	When **On**, activates the flow-boundary diffusion fluxes (or viscous fluxes for flow models) given by Eqn. (1593). This option is **On** by default. A realistic temperature value must be set on flow boundaries when this option is activated.	
Secondary Gradients	Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces.	
	On	Include both secondary gradients.
	Off	Exclude both secondary gradients.
	Interior Only	Include the interior secondary gradients only.
	Boundaries Only	Include the boundary secondary gradients only.

Segregated Fluid Isothermal

The Segregated Fluid Isothermal model keeps the temperature in the continuum constant. For problems where the temperature variations are small and negligible, it would be computationally expensive to solve an ordinary energy transport equation when it would yield a nearly constant field of temperature. Hence, STAR-CCM+ includes the isothermal energy option to provide a constant temperature field for all models that require temperature.

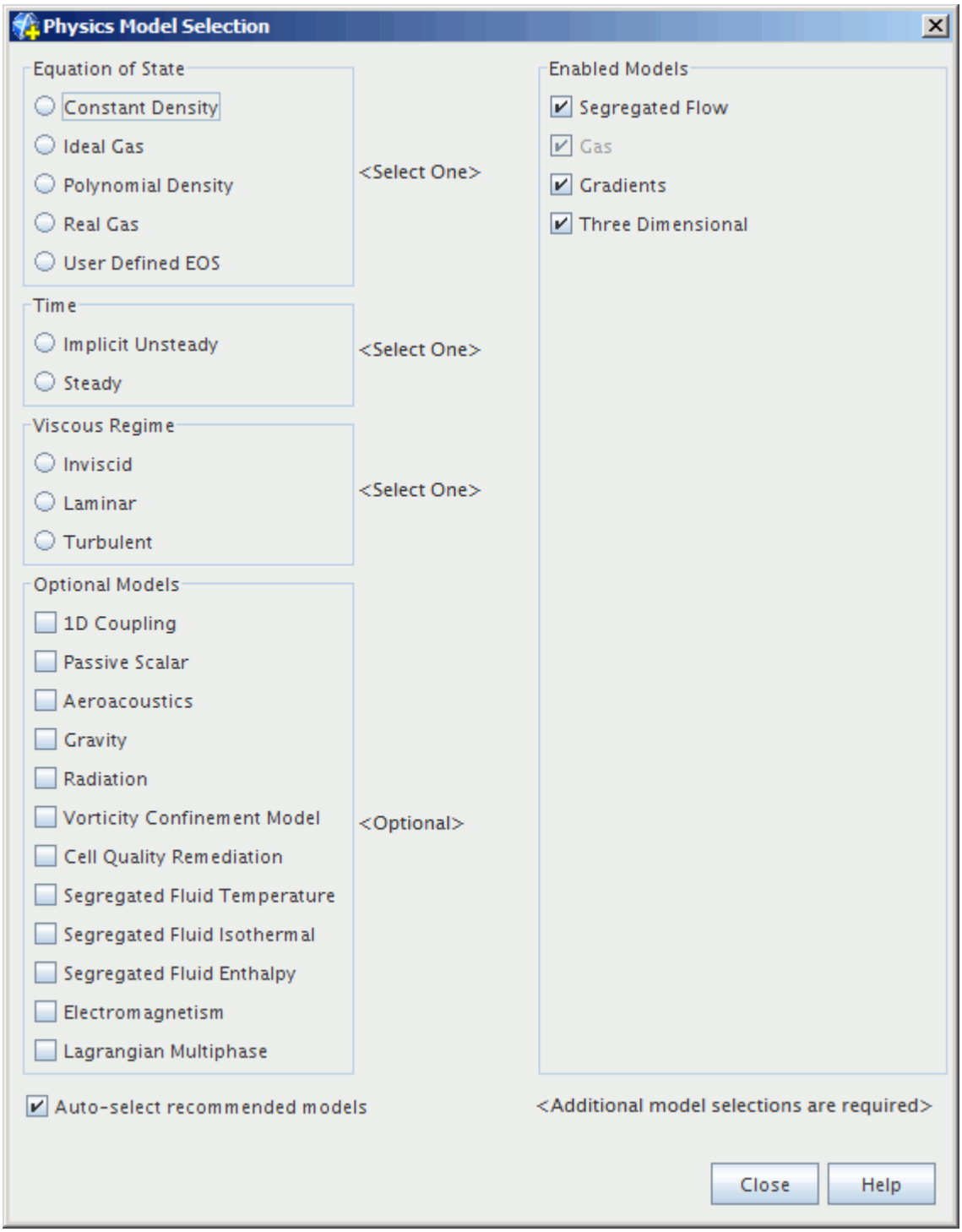
Properties

Continuum Temperature	Specifies the temperature for the continuum with a text entry.

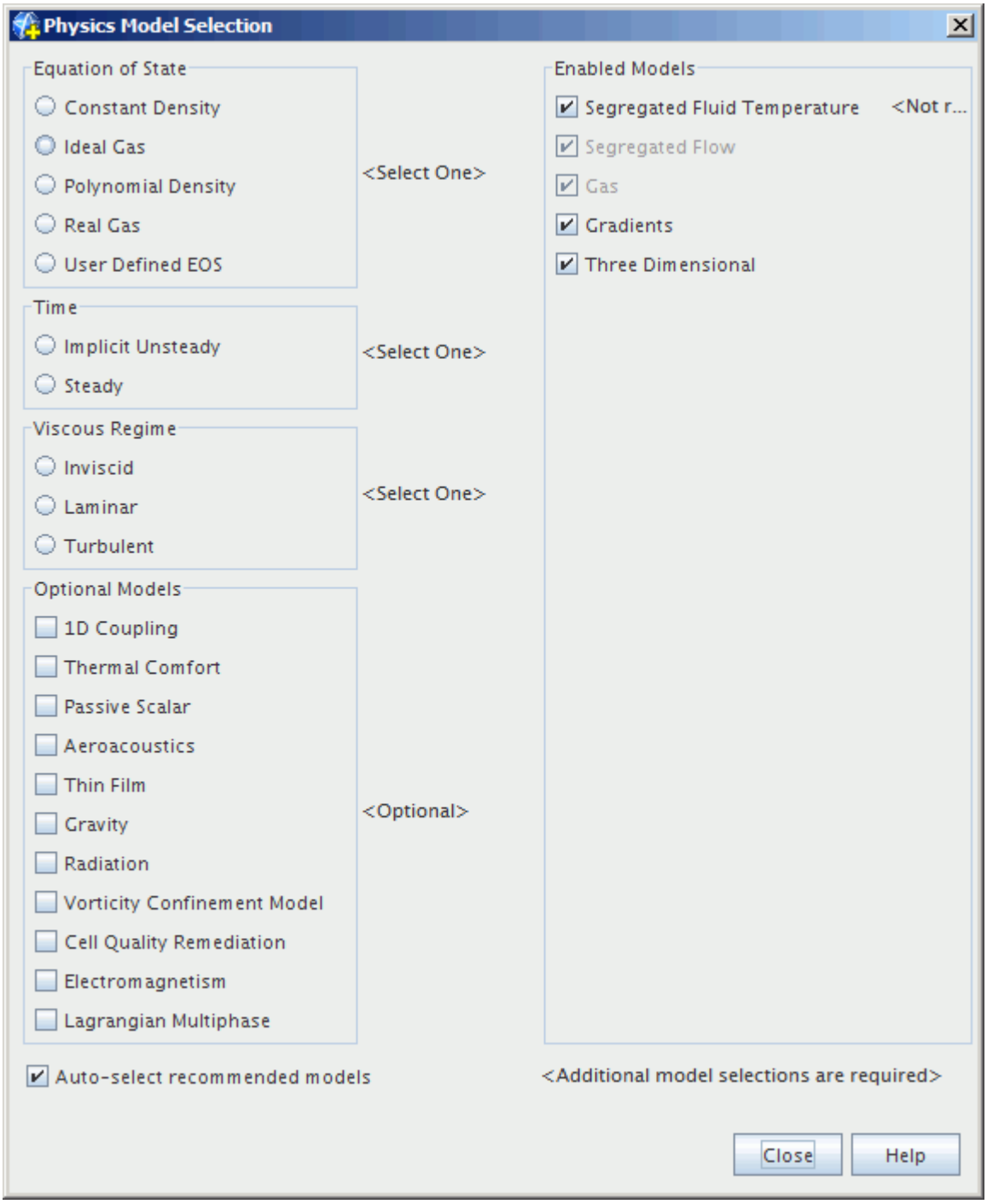
Selecting a Segregated Fluid Energy Model

To select a Segregated Fluid Energy model, first select the Segregated Flow model.

The **Segregated Fluid Enthalpy, Segregated Fluid Temperature**, and **Segregated Fluid Isothermal** radio buttons become available for selection in the *Optional Models* box:



The *Physics Model Selection* dialog with one of the Segregated Fluid Energy models activated is shown below:

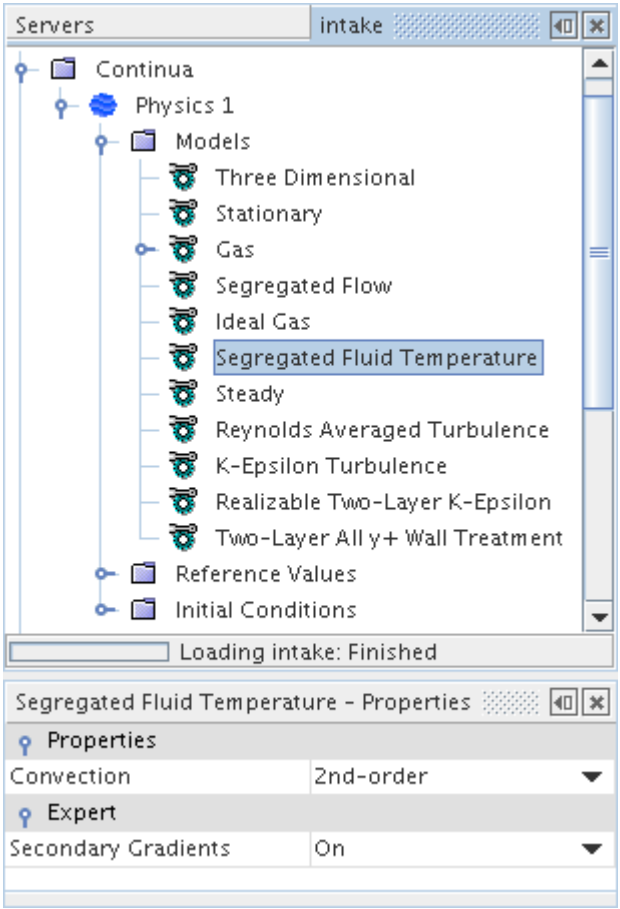


If a non-adiabatic PPDF combustion model is active, Segregated Fluid Enthalpy is selected automatically provided the Auto-select recommended physics models checkbox is activated. This model is the only appropriate choice for an energy model.

Setting Model Properties

To set the properties for the Segregated Fluid Enthalpy model or the Segregated Fluid Temperature model, select its node in the **Models** node under the required continuum node.

The properties are available under the Properties and Expert categories.



The *Convection* property switches between first-order and second-order upwind convection schemes.

The Segregated Fluid Isothermal model simply has a constant value to set for its property, *Continuum Temperature*.

Energy Solver

The Energy solver controls the solution update for the Segregated Fluid Energy model. It is used to set the under-relaxation factor and algebraic multigrid parameters for the energy equation.

Properties

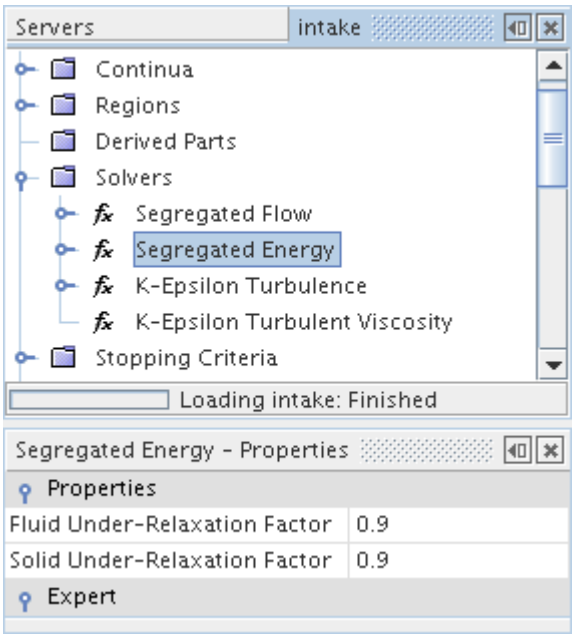
Under-Relaxation Factor	At each iteration, governs the extent to which the old solution is supplanted with the newly computed solution. For the theoretical background, see Finite Volume Discretization.

Expert Properties

For information about properties in the *Expert* category, see Expert Properties.

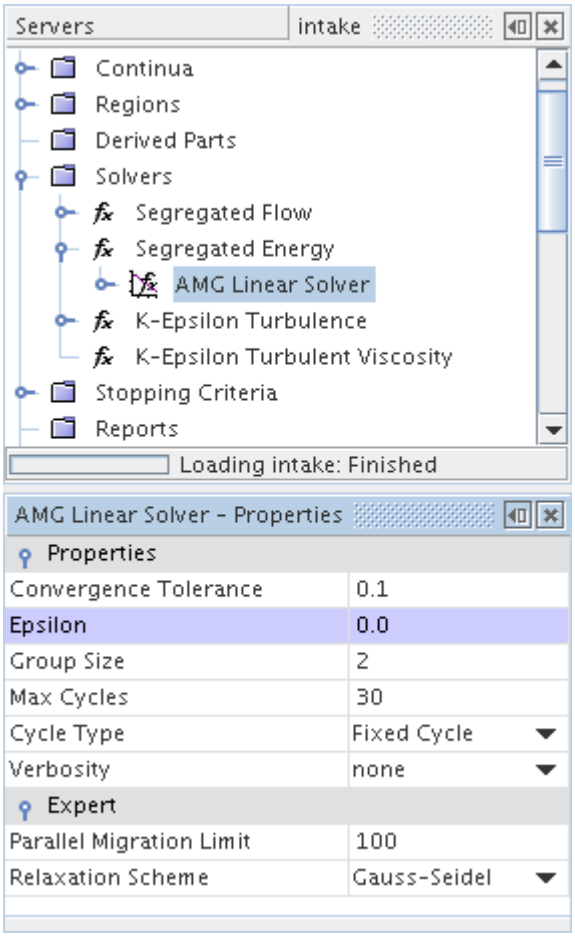
Setting the Under-Relaxation Factor

To set the under-relaxation factor for energy, select the **Energy Solver** node in the **Solvers** node as shown below. The *Under-Relaxation Factor* property for either a fluid or solid material, depending on what you model, can be entered in the *Properties* window.



Setting Algebraic Multigrid Parameters

To set the algebraic multigrid parameters for the energy solver, open the **Energy Solver** node in the **Solvers** node. Then also open the node for the **AMG Linear Solver** as shown below. The algebraic multigrid parameters can be entered using the *Properties* window.



See Also:

Modeling Flow and Energy

Initializing Flow

The process of solution initialization requires that suitable values are specified for the initial conditions. This section provides suggestions for setting initial conditions in both incompressible flows and compressible flows.

STAR-CCM+ also provides the following initialization tools to simplify the process:

- Continuity Initialization for the Segregated Flow Solver
- Activating Grid Sequencing for the Coupled Flow Solver

Initializing Incompressible Flow

Initializing velocity:

- If the solution domain has a prevalent flow direction, an approximate velocity can be specified in that direction.
- For ducts of complex shapes that do not have a prevalent flow direction, zero is an acceptable guess.

Initializing pressure:

- If the continuum has regions with pressure boundaries, the pressure can generally be set to the value specified on one of the pressure boundaries.
- If there are no pressure boundaries and the flow is incompressible, the initial pressure is arbitrary.

Initializing Compressible Flow

This section presents some suggestions for initializing velocity and for initializing pressure in compressible flows. See also Isentropic Flow Formulation.

Initializing velocity:

- For uni-directional ducts and free-stream flows over slender bodies, it is best to initialize the flow to the inlet or free-stream values.
- For bent ducts or bluff bodies, it can help to reduce the velocity significantly in these cases. Furthermore, techniques such as changing boundary conditions or ramping boundary values may be required to aid solution convergence.

Initializing pressure:

- If there are no pressure boundaries, the specified initial pressure must not result in a non-physical absolute pressure.
- For free-stream flows, choose the initial pressure such that it is equal to the free-stream pressure.
- For duct flows, choose the initial pressure such that it is equal to or higher than the outlet pressure. Choosing a value higher than the outlet pressure can inhibit temporary reversed flow occurring at the outlet.

The following isentropic relations are useful when initializing compressible flow cases:

$$\frac{T_t}{T} = 1 + \frac{(\gamma-1)}{2}M^2 \tag{222}$$

$$\frac{P_t}{P} = \left(1 + \frac{(\gamma-1)}{2}M^2\right)^{\gamma/(\gamma-1)} \tag{223}$$

$$V = M\sqrt{\gamma RT} \tag{224}$$

$$\frac{\dot{m}}{A} = \sqrt{\frac{\gamma}{R}}\frac{P_t}{\sqrt{T_t}}M\left(1 + \frac{(\gamma-1)}{2}M^2\right)^{1/2-\gamma/(\gamma-1)} \tag{225}$$

$$\gamma = \frac{C_p}{C_p - R} \tag{226}$$

$$R = \frac{R_u}{M_r(X)} \tag{227}$$

where:

- A is area
- $\dot{m}$ is mass flow rate
- M is Mach number
- $M_r(X)$ is molecular weight (relative molecular mass) of molecular entity X
- P is static pressure
- P_t is total pressure
- R is specific gas constant
- R_u is universal gas constant [8314.4621 J/kmol K]
- T is static temperature
- T_t is total temperature
- V is velocity magnitude
- γ is ratio of specific heats

Initial Conditions

Several initial conditions define the starting flow field in STAR-CCM+.

(Static) Pressure

The outlet pressure is reasonable to use for the initial value of pressure. If there are no pressure boundaries that are attached to the given continuum, the initial value is arbitrary. It is entered as a scalar profile.

Static Temperature

Energy models require this initial condition.

Velocity

For inviscid problems, the initial velocity must be non-zero. Zero initial velocity is also inappropriate if the Turbulence Intensity method is being used to specify initial turbulence values. It is entered as a vector profile.

Flow Boundaries Reference

STAR-CCM+ provides a wide range of conditions with which you can describe how flow behaves as it meets or passes through the boundaries of the solution domain. Setting these conditions correctly is a vital step in achieving a successful flow solution.

For flow simulations, STAR-CCM+ requires that you set quantities and choices on the following boundary types:

- Velocity Inlet
- Mass Flow Inlet
- Stagnation Inlet
- Free Stream
- Wall
- Pressure Outlet
- Outlet

Click on any link to see the condition and value nodes available for each type.

Velocity Inlet

Velocity Specification

Controls how you define velocity on the boundary.

Method	Corresponding Value Nodes
Components Define all the components of the velocity vector.	**Velocity** **Relative Velocity** Sets the velocity vector in the chosen *Coordinate System*. When relative, the vector is defined within the reference frame applied to the boundary.
Magnitude + Direction Define the magnitude of the velocity vector, and take the direction based on the **Flow Direction Specification**.	**Velocity Magnitude** **Relative Velocity Magnitude** Sets the velocity magnitude by the selected *Method*. When relative, the magnitude is defined within the reference frame applied to the boundary.
Background This method is available in the Dispersed Multiphase model only. The velocity of the dispersed phase is taken from the background flow.	None

Flow Direction Specification

Relative Flow Direction Specification

Controls how you define flow direction. When relative, the quantities are set in the frame of reference that you apply to the boundary.

Method	Corresponding Physics Value Nodes
Boundary Normal Sets flow direction normal to the boundary face.	None
Components	**Flow Direction** **Relative Flow Direction** Sets the direction vector in the chosen *Coordinate System* by the selected *Method*.
Angles Sets direction using Euler angles.	**Flow Angles** Defines the flow direction as the composition of rotations of a unit *Reference Vector* around certain axes of a reference *Coordinate System*. These rotations are known as Euler angles. When you set *Method* to **Constant**, set *Value* to the φ, θ, ψ rotations. When *Method* is **Field Function**, set *Angles Function* to an appropriate function. You can also provide tables. Three different rotation conventions are supported to define the coordinate axes that are used, and the order in which the rotations apply. Choose a *Rotation Convention* as follows: • **X Convention (Z-X-Z)**: For Euler angles (φ , θ , ψ), the first rotation is by angle φ about the z-axis, the second is by angle θ about the x-axis, and the third is by angle ψ about the z-axis. • **Y Convention (Z-Y-Z)**: For Euler angles (φ , θ , ψ), the first rotation is by angle φ about the z-axis, the second is by angle θ about the y-axis, and the third is by angle ψ about the z-axis. • **Yaw, Pitch, Roll Convention (Z-Y-X)**: For Euler angles (φ , θ , ψ), the first rotation is by angle φ about the z-axis, the second is by angle θ about the y-axis, and the third is by angle ψ about the x-axis. The *Axes Convention* applies rotations as follows: • **Fixed Axes**: relative to the original axes • **Moving Axes**: relative to the rotated axes

Reference Frame Specification

Allows you to apply a rotating reference frame to the boundary, and the source of the reference frame.

Option	Corresponding Physics Value Nodes
Lab Frame Uses the Laboratory reference frame (default). No rotation.	None
Region Reference Frame Uses the reference frame of the parent region. When the region has a rotating reference frame attached, a physically valid setup using this condition puts the boundary in the reference frame of the region. This implies that the boundary is rotating with the region at the same RPM, and has the same rotation axis and origin as the region.	None
Local Reference Frame This option allows the boundary to have its own frame (only rotating frame) attributes. For example, this can be used when both the boundary and the region are rotating at a different RPM but have the same axis and origin. A setup where the region and boundary have different values for the RPM, axis, and origin is also possible.	**Boundary Reference Frame Specification** Applies the chosen *Reference Frame* to the containing boundary.

Mass Flow Inlet

When you set a mass flow rate, STAR-CCM+ determines the velocity from the mass flow rate using $\dot{m} = \rho\, \mathbf{a} \cdot \mathbf{v}$.

A mass flow inlet differs from a velocity inlet in the way that density is used:

- For constant density flows, the two approaches are identical.
- For variable density flows, if the mass flow is specified, the velocity changes when the density is changed. If the velocity is specified, the mass flow changes when the density is changed.

You can specify a negative mass flow rate, negative mass flux, or a negative flow direction at a mass flow inlet boundary so that it behaves like an outlet. This feature is compatible with both segregated and coupled solvers. However, do not use it if any of the following conditions apply (STAR-CCM+ has no automated checks against them):

- Outflow exceeds inflow (for example, +1 kg/s at one inlet and -2 kg/s at the other); this condition would cause the mass flow boundary to attempt to pull more mass than is available.
- The flow near the boundary is much more than Mach 0.2.
- The flow is choked.

In a multiphase simulation, a mass flow inlet allows you to specify the mass flow rate or mass flux per phase.

Supersonic Static Pressure

Sets the static pressure upstream of an inlet boundary when supersonic conditions apply. STAR-CCM+ uses this value in conjunction with the Total Pressure to calculate the velocity of the flow entering the simulation.

Mass Flow Option

Specification Option	Corresponding Value Nodes
Mass Flux	**Mass Flux** **Relative Mass Flux** Scalar quantity for the mass flow rate per unit area. Spatial variations across the inlet can be specified using functions or tables. For more information, see Mass Flow Inlet in the Theory Guide.
Mass Flow Rate	**Mass Flow Rate** **Relative Mass Flow Rate** Scalar quantity for the mass flow rate. The Mass Flow Rate represents the total mass per unit time (kg/s) for the whole boundary. You can use field functions and tables to describe a dependence on Iteration or Time Step, but the mass flow rate cannot vary spatially across the boundary. The total mass flow is distributed over all of the faces of the part as described in Mass Flow Inlet in the Theory Guide. A reverse flow (negative mass flow rate) can be specified for a multiphase flow. However, this option should be used with caution.

Reference Frame Specification

As for Velocity Inlet.

Flow Direction Specification

Relative Flow Direction Specification

As for Velocity Inlet.

Stagnation Inlet

For this boundary type, you always set the total pressure upstream of the simulation domain.

Total Pressure

Scalar profile value.

Supersonic Static Pressure

Sets the static pressure upstream of an inlet boundary when supersonic conditions apply. STAR-CCM+ uses this value in conjunction with the Total Pressure to calculate the velocity of the flow entering the simulation.

Stagnation Inlet Option

Option	Corresponding Value Nodes
None	
Pressure Jump Adds an extra condition node, **Pressure Jump Option**.	

Pressure Jump Option

Option	Corresponding Value Nodes
None	
Fan The pressure rise is obtained from a fan curve. At a Stagnation Inlet boundary, the pressure rise is applied in the direction going into the domain. At a Pressure Outlet boundary, the pressure rise is applied in the direction going out of the domain. Selecting this option adds an additional condition node, **Fan Curve Type**.	Fan value nodes are controlled by the **Fan Curve Type**.
Porous Computes the pressure loss as $\alpha\rho\|\mathrm{v}\|^2 + \beta\rho\|\mathrm{v}\|$ where α is the **Porous Inertial Resistance** and β is the **Porous Viscous Resistance**. The pressure loss is always applied in the direction of flow.	**Porous Inertial Resistance** Scalar profile value for α. **Porous Viscous Resistance** Scalar profile value for β.
Loss Coefficient Computes the pressure loss as $0.5K\rho\|\mathrm{v}\|^2$ where K is the **Pressure Loss Coefficient**. The pressure loss is always applied in the direction of flow.	**Pressure Loss Coefficient** Scalar profile entry for K.

Fan Curve Type

Curve Type	Corresponding Value Nodes
Table	**Fan Curve Table** The fan curve table defines pressure rise as a function of four possible input variables. The accepted input variables, selected using *Method*, are: • **Local Velocity** • **Mass Average Velocity** • **Volumetric Flow Rate** • **Mass Flow Rate** After choosing the previously imported *Table*, set *Table: X* to the column that contains the input variable data, and *Table: P* to the column that contains the pressure rise data. Set the respective units using *Units: X* and *Units: P*. Use *Pressure Rise* to inform STAR-CCM+ about the basis on which the data is defined:

- **Standard**: pressure rise is the difference between the static pressure downstream and the total pressure upstream
- **Static to Static**: pressure rise is the difference between the static pressure downstream and the static pressure upstream. This definition also matches the difference between total pressure downstream and total pressure upstream

When you choose the **Static to Static** option, ensure that the specified fan curve is a decreasing function of the input variable. When you choose the **Standard** option, ensure that [specified fan curve + $\frac{1}{2}pv^2$] is a decreasing function of input variable.

Of the four fan curve *Method* options, only the **Mass Average Velocity** option is not linearized, and is considered to be less robust. The other options apply a different pressure rise to each face at the fan interface or boundary. However, for some problems, it can be more meaningful to apply the same pressure rise to all of the faces on the fan interface or boundary. In that scenario, you can use the mass averaged velocity options.

Enter the reference fan speed to which the performance tests apply in *Data Rotation Rate*, and the actual fan speed for the current simulation in *Operating Rotation Rate*. STAR-CCM+ applies fan scaling laws to the reference data contained within the table.

Fan Verbosity

When *Verbosity* is **On**, STAR-CCM+ prints detailed information on the behavior of the fan model.

Polynomial

Fan Curve Polynomial

Models the fan pressure rise as a polynomial in one of the following variables, which you select using *Method*:

- **Local Velocity**
- **Mass Average Velocity**
- **Volumetric Flow Rate**
- **Mass Flow Rate**

Enter the polynomial parameters using the custom property editor for *Polynomial*.

The properties, *Pressure Rise*, *Operating Rotation Rate*, and *Data Rotation Rate*, have the same purpose as for the **Fan Curve Table**.

Free Stream

Free Stream Option

Option	Corresponding Value Nodes
Mach Number + Pressure + Temperature Default freestream option.	**Mach Number** Accepts two alternative sets of inputs, depending on what you choose for **Free Stream Option**. You either set the *Mach Number* directly, or else specify an appropriate *Reynolds Number* and *Length Scale* from which STAR-CCM+ computes the Mach number.

	Pressure (boundary) Specify the working pressure in Pa. For both as an initial condition and as a boundary value the pressure is entered as a pressure profile. On a boundary the node has an additional option in the *Method* property—**Radial Equilibrium**. **Static Temperature** Energy models require this initial condition.
Altitude + Length Scale + Reynolds Number STAR-CCM+ adds the **Atmosphere Type Option** node for use in conjunction within this choice. Specify the appropriate Reynolds number and length scale values in the Mach Number node.	**Altitude** Sets the height at which STAR-CCM+ extracts the freestream pressure and temperature from the chosen Atmosphere Table. **Mach Number** Accepts two alternative sets of inputs, depending on what you choose for **Free Stream Option**. You either set the *Mach Number* directly, or else specify an appropriate *Reynolds Number* and *Length Scale* from which STAR-CCM+ computes the Mach number.
Altitude + Mach Number STAR-CCM+ adds the **Atmosphere Type Option** node for use in conjunction within this choice.	**Altitude** Sets the height at which STAR-CCM+ extracts the freestream pressure and temperature from the chosen Atmosphere Table. **Mach Number** Accepts two alternative sets of inputs, depending on what you choose for **Free Stream Option**. You either set the *Mach Number* directly, or else specify an appropriate *Reynolds Number* and *Length Scale* from which STAR-CCM+ computes the Mach number.

Atmosphere Type Option

For atmospheric flows, specifies the method by which STAR-CCM+ computes the freestream pressure and temperature from the altitude:

Option	Corresponding Value Nodes
Standard Uses the US 1976 Standard Atmosphere as a model of the Earth's atmosphere.	None.
User Table Uses your own imported table. Make sure that you first import the table within the **Tools** > **Table** node.	**Atmosphere Table** Use this node to configure your own imported table from which STAR-CCM+ extracts freestream pressure and temperature. Set *Table* to the imported table. Match the relevant table columns to *Table: Altitude*, *Table: P*, and *Table: T*. Set the corresponding units using *Units: Altitude*, *Units: P*, and *Units: T*.

Flow Direction Specification

Relative Flow Direction Specification

As for Velocity Inlet.

Wall

Reference Frame Specification

As for Velocity Inlet.

Shear Stress Specification

Defines how a wall surface acts on a fluid passing across it.

Method	Corresponding Physics Value Nodes
No-Slip Relative fluid velocity tangential to the wall is set to zero. When you choose `No-Slip`, STAR-CCM+ adds the **Tangential Velocity Specification.**	
Slip Fluid velocity at the wall is extrapolated from the parallel component of the velocity in the cell next to the wall.	
Specified Shear	**Wall Shear Force** Sets the wall shear force $(\mathbf{T} \cdot \mathbf{a})_f$. Enter as a vector profile in the chosen *Coordinate System*. **Wall Shear Velocity Derivative** Enter this quantity as a tensor profile.

Tangential Velocity Specification

When a Tangential Velocity Specification is used at a wall f, only the tangential component v_τ of the specified velocity $\mathbf{v}_{\text{spec}}^{\text{lab}}$ is used. If you specify a velocity with a component normal to the wall, the normal component of $\mathbf{v}_{\text{spec}}^{\text{lab}}$ is ignored, since the velocity $\mathbf{v}_f$ at the face is computed as:

$$\mathbf{v}_f = \mathbf{v}_{\text{spec}}^{\text{lab}} - \left(\mathbf{v}_{\text{spec}}^{\text{lab}} \cdot \mathbf{a} - G_f\right)\frac{\mathbf{a}}{a^2} \tag{228}$$

where $\mathbf{a}$ is the face area vector and $\mathbf{v}_{\text{spec}}^{\text{lab}}$ is the velocity in the laboratory frame. The normal component contribution at the face f can only come from the grid flux G_f that is given by Eqn. (144). That is, if the wall itself is moving at a velocity with a non-zero component in the wall normal direction.

Note that:

$$\mathbf{v}_{\text{spec}}^{\text{lab}} = \mathbf{v}_{\text{spec}}^{\text{mesh}} + \mathbf{v}_{\text{mesh}}^{\text{ref}} + \mathbf{v}_{\text{ref}}^{\text{lab}} \tag{229}$$

where:

- $\mathbf{v}_{\text{spec}}^{\text{lab}}$ is the specified velocity measured in the laboratory frame.
- $\mathbf{v}_{\text{spec}}^{\text{mesh}}$ is the specified velocity measured relative to the mesh.
- $\mathbf{v}_{\text{mesh}}^{\text{ref}}$ is the velocity of the mesh relative to the reference frame. This quantity is in **Region** > **Physical Values** > **Motion Specification**..
- $\mathbf{v}_{\text{ref}}^{\text{lab}}$ is the velocity of the reference frame relative to the laboratory frame.

The net tangential velocity in the laboratory frame is the vector sum of the tangential velocity with respect to the reference frame selected under the condition Reference Frame Specification and the velocity of the selected frame with respect to the laboratory frame.

Unless specified in an **Axis** node, axis of rotation is defined by information in the **Motion Specification** node under the **Physics Values** node.

Method	Corresponding Physics Value Nodes
Fixed Tangential velocity is zero with respect to the applicable reference frame.	
Vector Tangential velocity is specified by rotation rate with respect to the applicable reference frame.	**Velocity** **Relative Velocity** Sets the velocity vector in the chosen *Coordinate System*. When relative, the vector is defined within the reference frame applied to the boundary.
Rotation Rate Tangential velocity is specified as a vector with respect to the applicable reference frame.	**Wall Rotation** **Wall Relative Rotation** Sets the wall rotation rate around the axis of the reference frame applied to the containing boundary.
Local Rotation Rate Tangential velocity is defined by the rotation rate around the axis specified in the Axis node of the boundary and with respect to the applicable reference frame.	**Axis** Specifies a vector whose *Origin* and **Direction** define an axis of rotation. **Wall Rotation** **Wall Relative Rotation** As for **Rotation Rate**.

Pressure Outlet

For all but the Average Pressure option, you specify the pressure across the outlet.

Pressure (boundary)

Specify the working pressure in Pa. For both as an initial condition and as a boundary value the pressure is entered as a pressure profile. On a boundary the node has an additional option in the *Method* property—**Radial Equilibrium**.

Flow Direction

Relative Flow Direction

Sets the direction vector in the chosen *Coordinate System* by the selected *Method*.

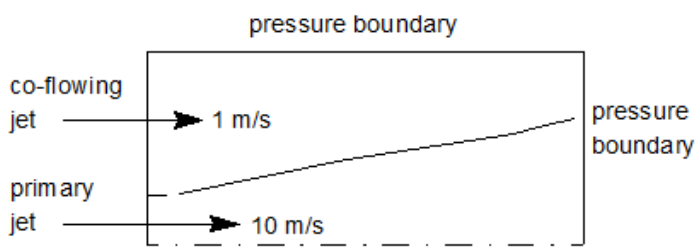
Backflow Specification

Sets the method by which STAR-CCM+ computes the direction of any flow that enters the simulation through an outflow boundary. For some methods, you must also set the associated Reference Frame Specification.

Pressure

Specifies how the pressure is computed.

- **Environmental**: Include dynamic head at pressure outlet boundary.
- **Static**: Do not use dynamic head in the case of inflow. Pressure is maintained at the pressure specified by the user.

Direction	Corresponding Physics Value Nodes
Extrapolated Extrapolates the flow direction from the interior of the domain. In most situations, this is less stable than the Boundary Normal option. However, it is recommended for situations where the flow is known to be parallel to the pressure boundary, such as a co-flowing jet. This option is set by default for the Coupled Flow model.  	None
Boundary Normal Assumes that the incoming flow enters the simulation along a vector that is orthogonal to the boundary surface, in the specified reference frame for the boundary. This option is set by default for the Segregated Flow model and generally provides the most robust convergence. If the flow is largely parallel to the boundary, the Extrapolated option should be used instead.	None

Components The inflow direction vector $\mathbf{V}_{user}$ is explicitly defined in the specified reference frame. See Eqn. (238). When the **Reference Frame Specification** is set to Lab Frame, this value is defined in the **Flow Direction** node under **Physics Values**. When the **Reference Frame Specification** is set to an option other than Lab Frame, this value is defined in the **Relative Flow Direction** node.	**Flow Direction** **Relative Flow Direction** Sets the direction vector in the chosen *Coordinate System* by the selected *Method*.
Angles Sets direction using Euler angles.	**Flow Angles** Defines the flow direction as the composition of rotations of a unit *Reference Vector* around certain axes of a reference *Coordinate System*. These rotations are known as Euler angles. When you set *Method* to **Constant**, set *Value* to the φ, θ, ψ rotations. When *Method* is **Field Function**, set *Angles Function* to an appropriate function. You can also provide tables. Three different rotation conventions are supported to define the coordinate axes that are used, and the order in which the rotations apply. Choose a *Rotation Convention* as follows: • **X Convention (Z-X-Z)**: For Euler angles (φ , θ , ψ), the first rotation is by angle φ about the z-axis, the second is by angle θ about the x-axis, and the third is by angle ψ about the z-axis. • **Y Convention (Z-Y-Z)**: For Euler angles (φ , θ , ψ), the first rotation is by angle φ about the z-axis, the second is by angle θ about the y-axis, and the third is by angle ψ about the z-axis. • **Yaw, Pitch, Roll Convention (Z-Y-X)**: For Euler angles (φ , θ , ψ), the first rotation is by angle φ about the z-axis, the second is by angle θ about the y-axis, and the third is by angle ψ about the x-axis.

The *Axes Convention* applies rotations as follows:

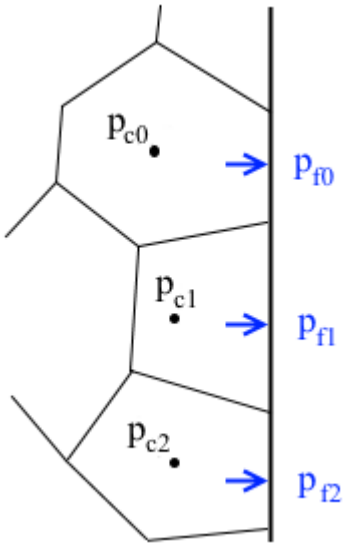
- **Fixed Axes**: relative to the original axes
- **Moving Axes**: relative to the rotated axes

Pressure Outlet Option

Provides a range of options for constraining pressure on an outflow boundary.

Option	Corresponding Physics Value Nodes
None	
Non-Reflecting Prevents spurious numerical reflection of the solution into the solution domain.	**Non-Reflecting Mode Specification** Specify the *Number of modes* to retain. Specify a number of modes less than the number of cells in the circumferential direction.
Target Mass Flow Automatically applies a correction to the specified Pressure boundary value in an attempt to yield the specified mass flow rate.	**Target Mass Flow Rate** Sets the *Massflow Rate* that you want to achieve, defined as $\dot{m}_{target}$ in Eqn. (244). In each iteration, STAR-CCM+ applies a *Pressure Adjustment* δp to the specified pressure. The correction process is constrained by average pressure values that you set, the *Minimum Allowable Avg Pressure* and the *Maximum Allowable Avg Pressure*. The *Under-Relaxation Factor* enters the pressure adjustment equation, Eqn. (244), as ω. While the adjustment is applied at every iteration, a new pressure adjustment value is computed according to the specified **Update Frequency** (iterations or time-steps). Activate *Verbose* for additional debugging output.
Pressure Jump Adds an extra condition node, **Pressure Jump Option**.	
Average Pressure The pressure profile from the cells next to the pressure outlet boundary is adjusted so that the average of that profile is equal to the average pressure value specified under Average Pressure	**Average Pressure** Sets the average pressure p_{avg} as a scalar profile.

node. This adjusted profile is then applied to the faces of the pressure outlet boundary.



The pressure at the boundary face of cell i is computed as:

$$p_{fi} = p_{avg} + \beta\left(p_{ci} - p_{c,avg}\right)$$

where:

- p_{avg} is the average pressure.
- β is the blending factor that you select.
- p_{ci} is the pressure in cell i.
- $p_{c,avg} = \frac{\sum_i p_{ci} A_{fi}}{\sum_i A_{fi}}$ where A_{fi} is the area of the boundary face of cell i.

These computations are made only for faces that do not have backflow. Faces with backflow or supersonic flow are treated as for the default pressure outlet boundary condition. See Pressure Outlet.

When using this option with Coupled Flow, you are advised to use the Continuity Convergence Accelerator for faster convergence. STAR-CCM+ issues warnings to this effect.

Average Pressure Adjuster

Sets the factor β that STAR-CCM+ uses to smooth out the pressure profile that is extrapolated from cells next to the pressure outlet boundary. The value can range from 0 to 1. The default value is 0.5.

Pressure Jump Option

Option	Corresponding Value Nodes
None	
Fan The pressure rise is obtained from a fan curve. At a Stagnation Inlet boundary, the pressure rise is applied in the direction going into the domain. At a Pressure Outlet boundary, the pressure rise is applied in the direction going out of the domain. Selecting this option adds an additional condition node, **Fan Curve Type**.	Fan value nodes are controlled by the **Fan Curve Type**.
Porous Computes the pressure loss as $\alpha\rho\|\mathrm{v}\|^2 + \beta\rho\|\mathrm{v}\|$ where α is the **Porous Inertial Resistance** and β is the **Porous Viscous Resistance**. The pressure loss is always applied in the direction of flow.	**Porous Inertial Resistance** Scalar profile value for α. **Porous Viscous Resistance** Scalar profile value for β.
Loss Coefficient Computes the pressure loss as $0.5K\rho\|\mathrm{v}\|^2$ where K is the **Pressure Loss Coefficient**. The pressure loss is always applied in the direction of flow.	**Pressure Loss Coefficient** Scalar profile entry for K.

Pressure Jump Under-Relaxation Factor

Stabilises the simulation and improves convergence.

The default setting is 0.5 if the Pressure Jump Option is enabled when the Segregated Solver is active and is 1.0 if the Pressure Jump Option is enabled when the Coupled Solver is active. Once the Pressure Jump Option is enabled, the Pressure Jump Under-Relaxation Factor setting does not change if you subsequently switch between solvers. However, you can manually change the Pressure Jump Under-Relaxation Factor as required. Additionally, when the Segregated Solver is active, you can modify the Pressure Jump Under-Relaxation Factor in conjunction with the Velocity and Pressure Under-Relaxation Factors.

Fan Curve Type

Curve Type	Corresponding Value Nodes
Table	**Fan Curve Table** The fan curve table defines pressure rise as a function of four possible input variables. The accepted input variables, selected using *Method*, are: • **Local Velocity** • **Mass Average Velocity** • **Volumetric Flow Rate** • **Mass Flow Rate** After choosing the previously imported *Table*, set *Table: X* to the column that contains the input variable data, and *Table: P* to the column that contains the pressure rise data. Set

	the respective units using *Units: X* and *Units: P*. Use *Pressure Rise* to inform STAR-CCM+ about the basis on which the data is defined: • **Standard**: pressure rise is the difference between the static pressure downstream and the total pressure upstream • **Static to Static**: pressure rise is the difference between the static pressure downstream and the static pressure upstream. This definition also matches the difference between total pressure downstream and total pressure upstream When you choose the **Static to Static** option, ensure that the specified fan curve is a decreasing function of the input variable. When you choose the **Standard** option, ensure that [specified fan curve + $\frac{1}{2}pv^2$] is a decreasing function of input variable. Of the four fan curve *Method* options, only the **Mass Average Velocity** option is not linearized, and is considered to be less robust. The other options apply a different pressure rise to each face at the fan interface or boundary. However, for some problems, it can be more meaningful to apply the same pressure rise to all of the faces on the fan interface or boundary. In that scenario, you can use the mass averaged velocity options. Enter the reference fan speed to which the performance tests apply in *Data Rotation Rate*, and the actual fan speed for the current simulation in *Operating Rotation Rate*. STAR-CCM+ applies fan scaling laws to the reference data contained within the table. **Fan Verbosity** When *Verbosity* is **On**, STAR-CCM+ prints detailed information on the behavior of the fan model.
Polynomial	**Fan Curve Polynomial** Models the fan pressure rise as a polynomial in one of the following variables, which you select using *Method*: • **Local Velocity** • **Mass Average Velocity** • **Volumetric Flow Rate** • **Mass Flow Rate** Enter the polynomial parameters using the custom property editor for *Polynomial*. The properties, *Pressure Rise*, *Operating Rotation Rate*, and *Data Rotation Rate*, have the same purpose as for the **Fan Curve Table**.

Reference Frame Specification

As for Velocity Inlet.

Outlet

Mass Flux Specification

Mass Flux Specification	Corresponding Value Nodes
Specified Split Ratios	**Split Ratio** When there are multiple outlet boundaries on one continuum, the fraction of the mass flow passing through each of the boundaries must be specified. The specified fraction is ignored if there is only one outlet boundary in the continuum. This is a simple value with only one method of entering the data-type in a value from 0 to 1.
Specified Mass Fluxes	**Mass Flow Rate** **Relative Mass Flow Rate** Scalar quantity for the mass flow rate. The Mass Flow Rate represents the total mass per unit time (kg/s) for the whole boundary. You can use field functions and tables to describe a dependence on Iteration or Time Step, but the mass flow rate cannot vary spatially across the boundary. The total mass flow is distributed over all of the faces of the part as described in Mass Flow Inlet in the Theory Guide.

Outlet Flow Specification

Outlet Flow Specification	Corresponding Value Nodes
Specified Split Ratios	**Split Ratio** When there are multiple outlet boundaries on one continuum, the fraction of the mass flow passing through each of the boundaries must be specified. The specified fraction is ignored if there is only one outlet boundary in the continuum. This is a simple value with only one method of entering the data-type in a value from 0 to 1.
Specified Mass Fluxes	**Mass Flow Rate** **Relative Mass Flow Rate** Scalar quantity for the mass flow rate. The Mass Flow Rate represents the total mass per unit time (kg/s) for the whole boundary. You can use field functions and tables to describe a dependence on Iteration or Time Step, but the mass flow rate cannot vary spatially across the boundary. The total mass flow is distributed over all of the faces of the part as described in Mass Flow Inlet in the Theory Guide.

Specified Volumetric Fluxes	**Volumetric Flow Rate** Scalar quantity for the volumetric flow rate. The volumetric flow rate represents the total volume per unit time for the whole boundary. In cases when there are large differences between the phase densities, using the volumetric flow rate of the multiphase mixture allows a smoother outlet profile. If a constant mass flow rate is specified, the flow velocity can become unstable. For example, a bubble of air in water will experience a large velocity increase as it passes through the outlet boundary. See Specified Volumetric Fluxes Formulation.

A reverse flow (negative mass flow rate) is not supported as the volume fraction cannot be specified on the boundary.

Partial Slip

Use the slip condition models—Maxwell Slip and von Smoluchowski Slip—in situations where the no-slip boundary condition does not apply. Such cases include rarefied flows where the Knudsen number is between 0.01 and 0.1.

These models apply to walls with thermal boundary conditions, **Environment**, **Convection**, **Heat Flux**, **Adiabatic**, and **Temperature**. See Boundary Conditions, High-Speed Flow Regimes and Modeling Fully Developed Flow and Energy.

Maxwell Slip Model

Provided by	**[physics continuum]** > **Models** > **Optional Models**	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Maxwell Slip**	
Requires	*Material*: **Gas** or **Multi-Component Gas** *Flow*: **Coupled Flow** or **Segregated Flow** *Viscous Regime*: **Laminar**	
Activates	Physics Models	None
	Material Properties	**Mean Free Path** **Momentum Accommodation Factor** See Maxwell Slip Material Properties
Properties	None	
Actions	See Maxwell Slip Action.	

Maxwell Slip Material Properties

Selecting the Maxwell Slip model exposes the following properties under the **Material Properties** node for the **Gas** phase, or for each component of **Multi-Component Gas**:

Mean Free Path

L in Eqn. (230).

Method	Corresponding Value Nodes
Kinetic Theory Use kinetic theory to calculate the mean free path.	None

Momentum Accommodation Factor

σ_v in Eqn. (230). For multi-component gases, this value is calculated as a mass-weighted average of the gas components. The value must be in the range from 0 through 1.

Maxwell Slip Actions

Apply the Maxwell Slip model to a specific wall boundary.

Select `Physics Conditions` > **Shear Stress Specification** for the wall boundary and set **Method** to **Partial-Slip**.

Von Smoluchowski Slip Model

Model Names	**von Smoluchowski Slip**	
Provided by	**[physics continuum]** > **Models** > **Optional Models**	
Example Node Path	**Continua** > **Physics 1** > **Models** > **von Smoluchowski Slip**	
Requires	*Material*: **Gas** *Flow*: **Coupled Flow** or **Segregated Flow** *Viscous Regime*: **Laminar** *Optional Models*: **Maxwell Slip** *Optional Models*: an Energy model	
Activates	Material Properties	**Thermal Accommodation Factor** See Von Smoluchowski Slip Material Properties
Properties	None	
Actions	See Von Smoluchowski Slip Actions.	

Von Smoluchowski Slip Material Properties

Selecting the von Smoluchowski Slip model exposes the following property under the **Material Properties** node for the **Gas** phase, or for each component of **Multi-Component Gas**:

Thermal Accommodation Factor

σ_t in Eqn. (232). For multi-component gases, this value is calculated as a mass-weighted average of the gas components. The value must be in the range from 0 through 1.

Von Smoluchowski Slip Actions

Apply the von Smoluchowski Slip model to a specific wall boundary.

Select `Physics Conditions` > **Shear Stress Specification** for the wall boundary and set **Method** to **Partial-Slip**.

Select `Physics Conditions` > **Thermal Specification** for the wall boundary and set **Condition** to **Temperature**, **Convection**, **Environment**, **Heat Flux**, or **Adiabatic**. Then set the corresponding value in **Physics Values**.

Partial Slip Formulation

The partial slip formulation consists of velocity- and temperature-dependent boundary conditions.

The velocity-dependent slip boundary condition, by Maxwell, is:

$$u_{gas} - u_{wall} = \frac{2 - \sigma_v}{\sigma_v} L \frac{\partial u}{\partial n}\bigg|_{wall} \tag{230}$$

where:

- u_{gas} is the tangential slip velocity.
- u_{wall} is the wall velocity.
- σ_v is the tangential momentum accommodation factor, an empirical parameter ranging from 0 through 1, denoting the gas-surface momentum exchange.
- L is the mean free path of the gas:

$$L = \frac{\kappa T}{\sqrt{2}\pi P \sigma^2} \tag{231}$$

  where:

  - κ is the Boltzmann constant, 1.38066 × 10^{-23}J/K.
  - T is temperature.
  - P is pressure.
  - σ is the Lennard Jones characteristic length or collision diameter.
- u is velocity and n is normal distance from the wall.

The temperature-dependent slip boundary condition, by von Smoluchowski, is:

$$T_{gas} - T_{wall} = \frac{2 - \sigma_t}{\sigma_t} \cdot \frac{2\gamma}{\gamma + 1} \cdot \frac{L}{Pr} \cdot \frac{\partial T}{\partial n}\bigg|_{wall} \tag{232}$$

where:

- T_{gas} is the gas temperature at the wall.
- T_{wall} is the wall temperature.
- σ_t is the thermal accommodation factor, an empirical parameter ranging from 0 through 1, denoting the gas-surface energy exchange.
- γ is the specific heat ratio.
- Pr is the Prandtl number.

Boundary Conditions Formulation

This section describes the boundary condition formulation for the flow and energy models, considering each of the following boundary types in turn.

Contents:

Velocity Inlet
Symmetry Plane
Wall
Stagnation Inlet
Pressure Outlet
Mass Flow Inlet
Free-Stream
Flow-Split Outlet
Modeling Non-Reflecting Boundaries and Mixing-Plane Interfaces

Velocity Inlet

Velocity

The inlet face velocity vector is specified directly.

Pressure

The boundary face pressure is extrapolated from the adjacent cell using reconstruction gradients.

Temperature

The inlet static temperature is specified.

If applicable to the case, it is possible to specify inlet boundary conditions on a reference frame.

Symmetry Plane

Velocity

The shear stress at a symmetry boundary is zero. The face value of velocity is computed by extrapolating the parallel component of velocity in the adjacent cell using reconstruction gradients.

Pressure

The boundary face pressure is extrapolated from the adjacent cell using reconstruction gradients.

Temperature

The heat flux at a symmetry face is zero. The face value of temperature is extrapolated from the adjacent cell using reconstruction gradients.

Wall

Velocity

For slip walls, the face value of velocity is computed by extrapolating the parallel component of velocity in the adjacent cell using reconstruction gradients.

For no-slip walls, the tangential velocity is explicitly set either to zero (for the case of no wall motion) or to a specified value.

Pressure

The boundary face pressure is extrapolated from the adjacent cell using reconstruction gradients.

Temperature

For inviscid flow, the face value of temperature is extrapolated from the adjacent cell using reconstruction gradients.

For viscous flows, either the temperature or the heat flux is specified. If the heat flux is specified, the wall boundary temperature is computed from the appropriate laminar or turbulent heat flux relationship.

At walls, the wall heat flux is:

$$q_w = -k\frac{\mathrm{d}\,T}{\mathrm{d}\,n} \tag{233}$$

This flux is linearized in terms of the cell and wall temperatures using Eqn. (308). The coefficients of this linearization are functions of the wall model (for example, laminar, turbulent, with or without radiation, with or without boiling). When the cell temperatures are updated, the wall temperature is computed using Eqn. (308).

Stagnation Inlet

Velocity

The boundary face velocity magnitude $\left|\mathbf{v}_f\right|$ is obtained from:

$$\left|\mathbf{v}_f\right| = \sqrt{2\,C_p\left(T_{tf} - T_f\right)} \tag{234}$$

where C_p is the specific heat, T_{tf} is the total temperature and T_f is the static temperature at boundary face f.

The direction vector is explicitly specified.

Pressure

The boundary face total pressure P_{tf} is specified. For a supersonic flow, the boundary face pressure P_f is specified, while for subsonic flow it is extrapolated from the adjacent cell using reconstruction gradients.

You can specify a pressure jump across the boundary.

This option also applies to pressure outlet boundaries.

Temperature

The boundary face total temperature T_{tf} is specified, and the static temperature is obtained as follows:

$$T_f = \frac{T_{tf}}{\left(P_{tf}/P_f\right)^{R/C_p}} \tag{235}$$

If applicable to the case, it is possible to specify inlet boundary conditions on a reference frame.

See Also:

Specifying the Fan Performance Curve as a Polynomial Function

Pressure Outlet

Velocity

The boundary face velocity is extrapolated from the interior using reconstruction gradients. In the case of the Segregated Flow model, an option exists to use only the normal component of the velocity for inflow. This option can stabilize the solution in certain situations.

The continuity equation imposes an internal constraint on the velocity normal to the boundary surface. The velocity normal, U_n, is given as:

$U_n = \mathbf{U} \cdot \mathbf{a}/|\mathbf{a}|$

where a is the area vector.

This constraint means that U_n, the normal component of the velocity $\mathbf{U}$ in the absolute (Laboratory) reference frame must be left unchanged regardless of the overall direction defined for the backflow. If the overall direction is defined by $\mathbf{k}$, then the following formula applies to all **Backflow Specification** options, apart from the **Extrapolated** option (where the velocity at the boundary is equal to the internal velocity):

$$\mathbf{U}' = \frac{\mathbf{W}\cdot\mathbf{n}}{\mathbf{k}\cdot\mathbf{n}}\,\mathbf{k} + \mathbf{V}_{frame} \tag{236}$$

where $\mathbf{U}'$ is the new inflow velocity in the absolute frame and $\mathbf{W} = \mathbf{U} - \mathbf{V}_{frame}$ is the relative velocity.

Eqn. (236) preserves U_n as $\mathbf{U}' \cdot \mathbf{n} = \mathbf{U} \cdot \mathbf{n}$. Eqn. (236) is effectively scaling the desired velocity in order to match U_n.

For the **Backflow Specification** options, $\mathbf{k}$ is defined as follows:

- *Boundary Normal*

$$\mathbf{k}=\mathbf{n} \tag{237}$$

- *Components*

$$\mathbf{k} = \mathbf{V}_{user} \tag{238}$$

  where the vector inflow $\mathbf{V}_{user}$ is explicitly defined in the specified reference frame.

- *Angles*

$$\mathbf{k} = \mathbf{n}^{\alpha}{}_{user} \tag{239}$$

  where the direction angles of the inflow velocity are explicitly defined in the specified reference frame. The specified direction angles are internally converted into the components of a unit vector, where $\mathbf{n}^{\alpha}{}_{user}$ are the unit vector cosine angles.

Inflow Boundary Pressure

When inflow occurs, the boundary pressure p_f is assumed to be given by:

$$p_f = p_{specified} - \frac{1}{2}\rho_f |\mathrm{v}_n|^2 \tag{240}$$

where v_n is the normal component of the boundary inflow velocity. This approach discourages backflow from occurring.

Outflow Boundary Pressure

For subsonic outflow, the boundary pressure p_f is specified.

For supersonic outflow, the boundary face pressure is extrapolated from the adjacent cell value using reconstruction gradients.

For radial equilibrium outflow, the radial pressure gradient exactly balances the centrifugal forces due to rotation. To apply radial equilibrium to a specified pressure boundary condition, one must impose the condition:

$$p_f = p_i + \int_{r_i}^{r} \frac{\rho \mathrm{v}_\theta^2}{r} \, \mathrm{d}\, r \tag{241}$$

where the subscript i refers to the inner location (that is, the hub in a turbomachine) and v_θ is the tangential component of the velocity.

To obtain the function $\left(\rho\ \mathrm{v}_\theta^2\right)/r$ as a function of r only, the data is sorted into bins. An arbitrary number of 100 bins is chosen. If any bins are empty, linear interpolation fills them. Since the interpolated data is going to be numerically integrated, the interpolation order is not very important.

Target Mass Flow Rate and Pressure Correction

When the **Target Mass Flow Rate Option** is enabled, the boundary condition automatically applies a correction to the pressure value specified at the boundary in an attempt to yield the specified mass flow rate.

At each iteration, the value of pressure that is applied at the boundary is given by the specified pressure value plus an adjustment:

$$p_f = p_{specified} + \delta p \tag{242}$$

A pressure correction of $\Delta p \equiv {p_f}^{n+1} - {p_f}^{n}$ is calculated as follows:

$$\Delta p = \frac{\omega \cdot |\dot{m}^n| \cdot \left(\dot{m}^n - \dot{m}_{target}\right)}{\left({\rho_{ave}}^n \cdot A^2\right)} \tag{243}$$

where $\dot{m}^n$ and ${\rho_{ave}}^n$ are the mass flow rate and average density at the current iteration, and $\dot{m}_{target}$ and A are the target mass flow rate and total area of the boundary, respectively. ω is an under-relaxation factor.

Combining Eqns. Eqn. (242) and Eqn. (243), the new pressure adjustment value becomes:

$$\delta p^{n+1} = \frac{\omega \cdot |\dot{m}^n| \cdot \left(\dot{m}^n - \dot{m}_{target}\right)}{\left({\rho_{ave}}^n \cdot A^2\right)} + \delta p^n \tag{244}$$

While the adjustment is applied at every iteration, a new pressure adjustment value is computed according to a specified update frequency.

Dynamic Head and Pressure Specification

Problems arise when flow recirculates at the pressure outlet boundary, in which case that boundary is no longer a real outlet. To prevent this, the model automatically subtracts the dynamic head from pressure only on the faces where recirculation occurs. (This is in addition to the pressure terms given in Eqn. (240).) For some applications, however, it is important to maintain the specified pressure at the outlets. In such cases, the user can disable the use of the dynamic head by setting the **Pressure specification** option to **Static** for the pressure outlet boundary concerned.

The pressure specification method is present for Pressure Outlet boundary conditions. When this option is set to **Static**, Eqn. (240) no longer applies and the solver disables the subtraction of the dynamic head from the specified pressure in case of reversed flow or inflow at the pressure outlet boundary.

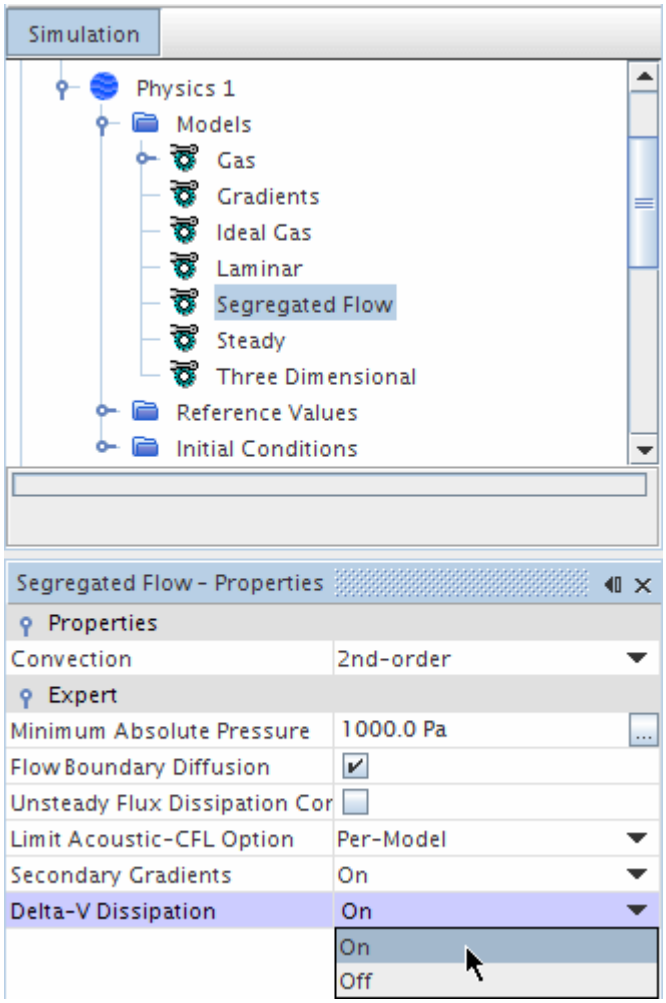
Setting the Pressure Jump Option

You can specify a pressure jump across the boundary. This option also applies to Stagnation Inlet boundaries.

Resolving Flow Convergence Failure

When you use the Pressure Jump Option with the Segregated Flow Solver, there can be times when the flow solution does not converge.

- To resolve this problem, activate *Delta-V Dissipation* in the **Segregated Flow** model:



Temperature

For outflow, the boundary face temperature is extrapolated from the adjacent cell using reconstruction gradients. For inflow, the specified static temperature is used.

See Also:

Specifying the Fan Performance Curve as a Polynomial Function

Mass Flow Inlet

A mass flow inlet allows you to specify the mass flow rate or mass flux per phase.

Mass Flow Rate

The Mass Flow Rate represents the total mass per unit time (kg/s) for the whole inlet. You can use field functions and tables to describe a dependence on Iteration or Time Step, but the mass flow rate cannot vary spatially across the inlet.

The total mass flow is distributed over all of the faces of the part as follows:

For single-phase flow, the mass flux is uniform. The mass flow rate formula is:

$$\dot{m}_f = \dot{m}\frac{\left|a_f\right|}{\sum_{f'}\left|a_{f'}\right|} \tag{245}$$

For multiphase flow, the mass flux is uniform over the available area according to the volume fraction profile that you specify. For this purpose, the inlet volume fraction profile can be interpreted as the phase fraction of the physical injection area. The mass flow rate formula is:

$$\dot{m}_f = \dot{m}\frac{\alpha_f\left|a_f\right|}{\sum_{f'}\alpha_{f'}\left|a_{f'}\right|} \tag{246}$$

where:

- $\dot{m}$ is the mass flow rate specified for the whole inlet.
- $\dot{m}_f$ is the mass flow rate applied to face f of the inlet.
- α_f is the volume fraction on the face.
- $\left|a_f\right|$ is the face area.

Velocity is also uniform, unless density is non-uniform. The velocity is determined as follows:

For single-phase flow, STAR-CCM+ determines the velocity using $\dot{m} = \rho\ \mathbf{a}\cdot\mathbf{v}$.

For multiphase flow, STAR-CCM+ determines the velocity of each phase using $\dot{m}_k = \alpha_k\rho_k\mathbf{a}\cdot\mathbf{v}_k$ where:

- α_k is the k-phase volume fraction
- ρ_k is the density, which is calculated for compressible flows
- $\mathbf{a}$ is the inlet area
- $\mathbf{v}_k$ is the resulting velocity of the k-phase.

Mass Flux

The Mass Flux is a scalar quantity for the mass flow rate per unit area.

For single-phase flow, STAR-CCM+ determines the velocity using $\dot{m}'' = (\rho\ \mathbf{a}\cdot\mathbf{v})/|\mathbf{a}|$.

For multiphase flow, STAR-CCM+ determines the velocity of each phase using $\dot{m}_k'' = \frac{\alpha_k\rho_k\mathbf{a}\cdot\mathbf{v}_k}{|\mathbf{a}|}$ where α_k is the k-phase volume fraction.

Spatial variations across the inlet can be specified using functions or tables.

Pressure

For supersonic inflow, the specified pressure is used. For subsonic flow, the boundary face pressure is extrapolated from the adjacent cell using reconstruction gradients.

Temperature

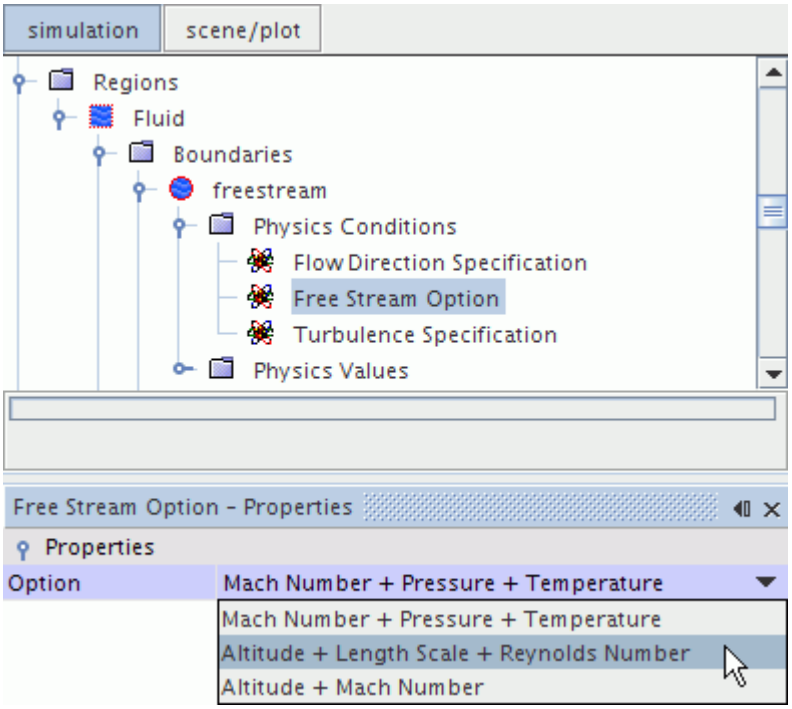
The static temperature is obtained from the specified total temperature using the isentropic definition of total enthalpy.

If applicable to the case, it is possible to specify inlet boundary conditions on a reference frame.

Free-Stream

For a freestream boundary, there are three ways that you can specify the freestream pressure, temperature, and Mach number.

You select the appropriate option as a property of the **Region Name** > **Boundary** > **Boundary Name** > **Physics Conditions** > **Free Stream Option** node.

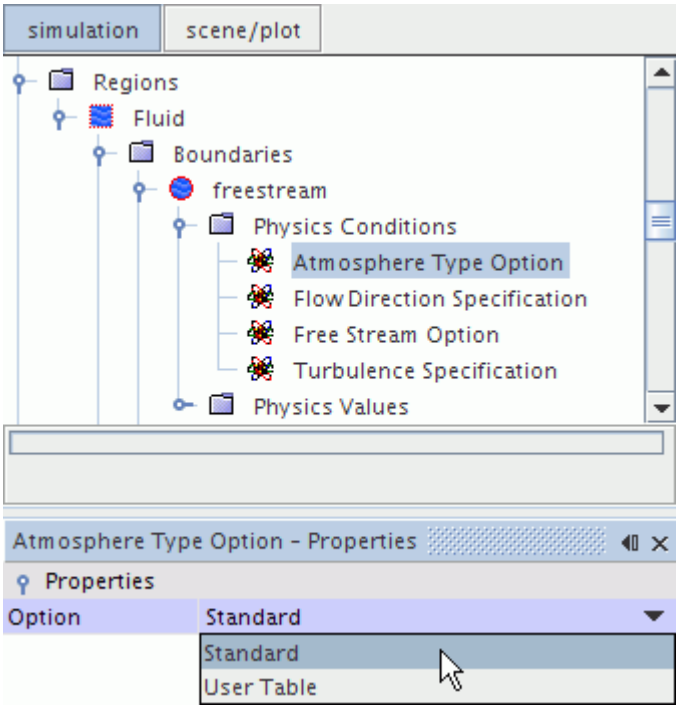


Free Stream Option Properties

Mach Number + Pressure + Temperature	For this option, STAR-CCM+ adds **Mach Number**, **Pressure**, and **Static Temperature** nodes to the Physics Values for the boundary. This option is the default freestream option.
Altitude + Length Scale + Reynolds Number	For this option, STAR-CCM+ adds the **Altitude** node to the Physics Values for the boundary. The freestream pressure and temperature are computed internally using the specified altitude. Specify the appropriate Reynolds number and length scale values in the **Mach Number** node under the **Physics Values** node for the boundary. The freestream Mach number is computed internally using the specified altitude, Reynolds number, and length scale.

Altitude + Mach Number	For this option, STAR-CCM+ adds the **Altitude** and **Mach Number** nodes to the Physics Values for the boundary. The freestream pressure and temperature are computed internally using the specified altitude.

When you use the *Altitude + Length Scale + Reynolds Number* or *Altitude + Mach Number* options, you also specify the appropriate atmosphere type as a property of the **Physics Conditions** > **Atmosphere Type Option** node.



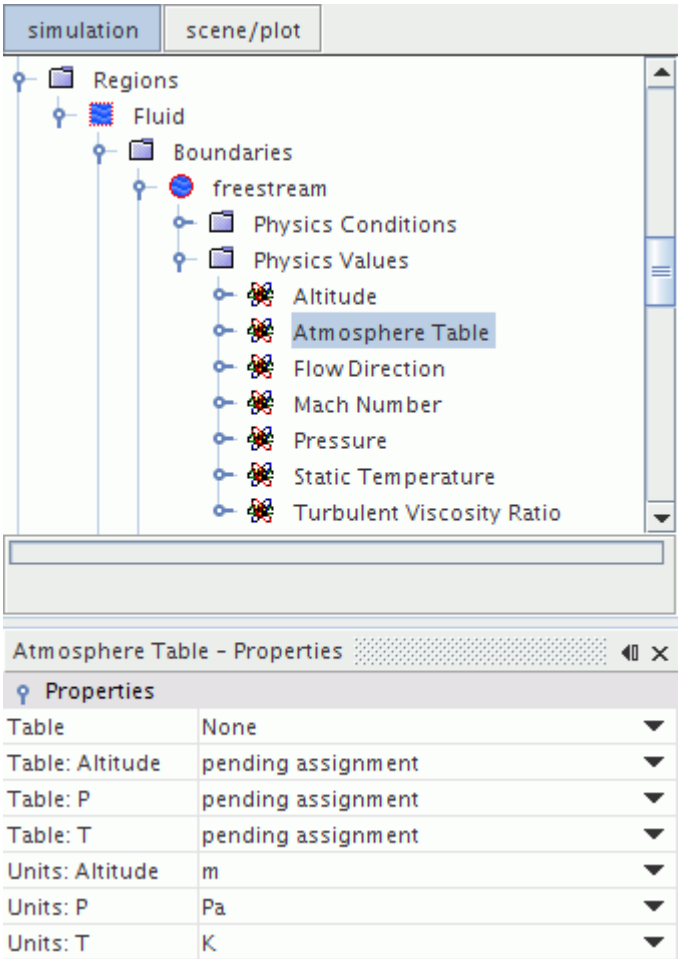
Atmosphere Type Option Properties

Standard	The US 1976 Standard Atmosphere is used to compute the freestream pressure and temperature from the altitude.
User Table	A user-supplied table is used to compute the freestream pressure and temperature from the altitude.

Using an Atmosphere Table

If you select the **User Table** option, you must first import the appropriate atmosphere table into STAR-CCM+.

You then set up the atmosphere table properties in the **Atmosphere Table** node under the **Region Name** > **Boundary** > **Boundary Name** > **Physics Values** node.



Note: When you use the *Altitude + Length Scale + Reynolds Number* or *Altitude + Mach Number* options, ensure that you set initial values that are consistent with the freestream values. You set the initial values in the physics continuum, under the **Initial Conditions** node, in the **Pressure** and **Static Temperature** nodes.

Velocity

For supersonic inflow, the velocity is taken to be the free-stream velocity, computed from the specified Mach number and temperature.

For supersonic outflow, the boundary face velocity is extrapolated from the cell value using reconstruction gradients.

For subsonic inflow, the tangential component of the free-stream velocity is combined with the boundary normal velocity computed from characteristics as follows:

$$\mathbf{v}_f = \mathbf{v}_\infty + \left(\mathrm{v}_{fn} - \mathrm{v}_{\infty\ n}\right)\frac{\mathbf{a}}{|\mathbf{a}|} \tag{247}$$

where $\mathrm{v}_{\infty n}$ is the component of free-stream velocity normal to the boundary and v_{fn} is the boundary normal velocity that is obtained from characteristics:

$$\mathrm{v}_{fn} = \frac{1}{2}\left(\mathrm{v}_{0\ n}^{r} + \mathrm{v}_{\infty\ n}\right) + \frac{c_0 - c_\infty}{\gamma - 1} \tag{248}$$

In this expression, v_{0n}^{r} is the normal component of the boundary velocity that is extrapolated from the cell. c_∞ is the speed of sound obtained from the free-stream temperature. c_0 is the speed of sound obtained from T_0^r, the boundary temperature that is extrapolated from the cell, that is:

$$c_0 = \sqrt{\gamma R T_0^r} \tag{249}$$

For subsonic outflow, the tangential component of the boundary velocity, that is extrapolated from the cell using the reconstruction gradient $\mathbf{v}_{0n}^{r}$, is combined with the boundary normal velocity computed from characteristics. The result is:

$$\mathbf{v}_f = \mathbf{v}_0^r + \left(\mathbf{v}_{fn} - \mathbf{v}_{0\ n}^{r}\right)\frac{\mathbf{a}}{|\mathbf{a}|} \tag{250}$$

where $\mathbf{v}_{fn}$ is again computed using Eqn. (248).

Pressure

For supersonic inflow, the pressure is taken to be the free-stream pressure.

For supersonic outflow, the boundary face pressure is extrapolated from the adjacent cell value using reconstruction gradients.

For subsonic inflow, the pressure is computed from the free-stream pressure using the following isentropic relation:

$$p_f = p_\infty \left(\frac{T_f}{T_\infty}\right)^{C_p/R} \tag{251}$$

For subsonic outflow, the pressure is computed from the adjacent cell value using the following isentropic relation:

$$p_f = p_0^r \left(\frac{T_f}{T_0^r}\right)^{C_p/R} \tag{252}$$

Temperature

For supersonic inflow, the boundary face temperature is taken to be the free-stream temperature.

For supersonic outflow, the boundary face temperature is extrapolated from the adjacent cell value using reconstruction gradients.

For subsonic flow, the boundary face temperature is computed as:

$$T_f = \frac{c_f^2}{\gamma R} \tag{253}$$

where:

$$c_f = \frac{1}{2}\left(c_\infty + c_0\right) + \frac{1}{4}\left(\gamma - 1\right)\left(\mathrm{v}_{0\ n}^r - \mathrm{v}_{\infty\ n}\right) \tag{254}$$

Outlet

Velocity

The flow velocity at an outlet boundary *i* is given by:

$$\mathbf{v}_f = \mathbf{v}_0^r + \frac{x_i}{\rho_f}\frac{\mathbf{a}}{|\mathbf{a}|} \tag{255}$$

where $\mathbf{v}_0^r$ is the velocity that is extrapolated from the adjacent cell value using reconstruction gradients and the vector $\mathbf{a}/|\mathbf{a}|$ is the outward-pointing normal. x_i is a scale factor that is computed for outlet boundary i as follows:

$$x_i = -\frac{f_i}{\sum_{j=1}^{n\ \mathrm{outlets}} f_j}\ \frac{\left(\dot{m}_{\mathrm{in}} - \dot{m}_{\ i}{}^*\right)}{\sum_{\mathrm{outlet}\ i\ \mathrm{faces}} |\mathbf{a}|} \tag{256}$$

where f_i is the specified fraction of the flow leaving outlet i, and $\dot{m}_{\mathrm{in}}$ is the total inlet flow, that is defined as:

$$\dot{m}_{\mathrm{in}} = \sum_{\mathrm{non\text{-}outlet\ faces}} \rho_f\left(\mathbf{v}_f \cdot \mathbf{a} - G\right) \tag{257}$$

and $\dot{m}^*_i$ is the (unscaled) outlet mass flow rate through outlet i:

$$\dot{m}^*_i = \sum_{\mathrm{outlet}\ i\ \mathrm{faces}} \rho_f \max\left(\mathbf{v}_0 \cdot\ \mathbf{a} - G, 0\right) \tag{258}$$

Pressure

The boundary face pressure is extrapolated from the adjacent cell using reconstruction gradients.

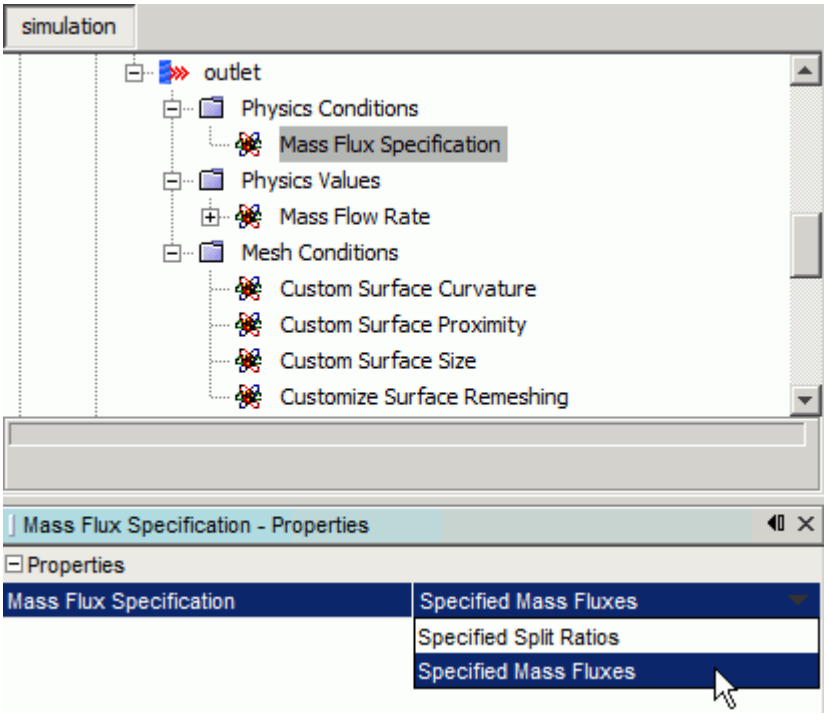
Temperature

The boundary face temperature is extrapolated from the adjacent cell using reconstruction gradients.

Setting the Mass Flux Specification

The flow split can be specified by flow ratio or by mass flow rate. However, all of the flow split boundaries in a continuum must use the same type of mass flux specification. That is, all of the flow split boundaries specify the flow rate using split ratios, or they all specify the flow rate using mass flow rates.

You specify the flow split type in the **Mass Flux Specification** node.



Mass Flux Specification Properties

Specified Split Ratios	Lets you specify the fraction (f_i) of the total flow that passes through the outlet. This boundary type cannot be used in the same continuum as pressure or stagnation boundaries.
Specified Mass Fluxes	Lets you specify the mass flow rate through the outlet. See Mass Flow Rate and Mass Flux.

Using a Specified Mass Flow Rate

When you specify a mass flow rate, at every iteration the solver makes a correction to the flow rate at this boundary to ensure that the total flow matches the specified flow rate. If the compressible solver is used, the extrapolated density is used for computing the mass flow rate.

This boundary type can be used in the same continuum as a pressure outlet. However, you need to consider the following:

- In scenarios where the pressure outlet may act as an inlet, you need to be careful how you specify the pressure condition. In the event of reversed flow, the dynamic head is taken out of the specified pressure, which can cause an incorrect pressure profile at the pressure outlet.
- If you specify the mass flow rate using a field function, an average value on that boundary is used as the specified mass flow rate. The mass flow rate must be a single value, not a field function with a value that varies along the boundary.

Setting the Outlet Flow Specification

The flow split can be specified by flow ratio, mass flow rate, or by volumetric flow rate. All of the flow split boundaries in a continuum must use the same type of outlet flow specification.

You specify the flow split type in the **Outlet Flow Specification** node.

Outlet Flow Specification Properties

Specified Split Ratios	As defined in Mass Flux Specification Properties.
Specified Mass Fluxes	As defined in Mass Flux Specification Properties.
Specified Volumetric Fluxes	Lets you specify the volumetric flow rate through the outlet. See Specified Volumetric Fluxes Formulation.

Specified Volumetric Fluxes Formulation

The mass flow rate is scaled by the density to give a volumetric flux:

$$T = \sum_f \frac{\dot{m}_f \left|\mathbf{A}_f\right|}{\rho_f} \tag{259}$$

For a multiphase flow, the mixture density is:

$$\rho_m = \sum_k^{phases} \alpha_k \rho_k \tag{260}$$

So the total mass*area is calculated by adding an additional sum over the phases:

$$T = \sum_k^{phases} \sum_f \frac{1}{\rho_{mf}} \dot{m}_{kf} \left|\mathbf{A}_f\right| \tag{261}$$

Modeling Non-Reflecting Boundaries and Mixing-Plane Interfaces

STAR-CCM+ can work effectively with non-reflecting boundaries and mixing-plane interfaces, allowing accurate simulations of applications such as multi-stage turbomachinery.

The implementation of non-reflecting boundaries and explicit mixing-plane interfaces is based on the work of Giles [207], [208], [209] and Saxer [235]. The theory is based on a Fourier analysis of the linearized Euler equations. The approach assumes that the solution at the boundary is decomposed, circumferentially in three-dimensional space, into Fourier modes.

However, these entities differ in the zeroth mode:

- For non-reflecting boundaries, the zeroth mode corresponds to the average solution and receives the standard boundary condition treatment.
- For explicit mixing-plane interfaces, conditions that are used for the zeroth mode solution are reached through an averaging of conserved quantities across the mixing-plane interface.

Additional information is available on these features as follows:

- The non-reflecting boundary condition—its theoretical background and purpose. This information includes setting average total pressure, temperature, and flow angle at the inlet, and pressure at the outlet.
- Details of setting up the non-reflecting option
- The mixing-plane interface—its uses and options.

Reporting Mass Flowrate

To obtain an accurate value of the computed mass flowrate in a simulation, apply the Mass Flow report to a boundary or interface.

STAR-CCM+ stores the mass flowrate on each face centroid on a boundary or interface. When you run a mass flow report on a boundary or interface, STAR-CCM+ sums the individual mass flowrates from all the faces on the surface.

Although you can run the mass flow report on an arbitrary plane section, interpolation errors mean that the computed value is an approximation of the true value. Choosing boundary or interface parts for the report allows STAR-CCM+ to access the stored quantity.

To compute mass flowrate:

1. Right-click the **Reports** node and choose **New Report** > **Mass Flow**.
2. Select the newly created report and set *Parts* to the boundary or interface through which you require the mass flowrate.

   Note: The Smooth Values property has no impact on the mass flowrate calculation; leave the property in its deactivated state.

3. For an instant calculation using the current simulation data, right-click the report and select **Run Report**.

   STAR-CCM+ writes the mass flowrate, in the selected units, to the *Output* window.

Modeling Passive Scalars

Passive scalars are user-defined variables of arbitrary value, assigned to fluid phases or individual particles. They are passive because they do not affect the physical properties of the simulation. An intuitive way to think of passive scalars is as tracer dye in a fluid, but with numerical values instead of colors, and with no appreciable mass or volume.

More elaborate physical effects can be modeled with passive scalars, by careful definition of the source terms, for example, change of composition through radioactive decay.

You can use more than one passive scalar in a simulation. The following situations are examples where this capability can be useful:

- Choosing the best position for a sensor while designing a part with multiple inlet pipes and an exhaust pipe.

  You can use the passive scalars to check the scalar values at discrete points in the computational domain. Then you can determine which of the scalars sends the strongest signal to the sensor.
- Analyzing the mixing of two fluid streams that have the same properties.

  The fluid is represented as single-phase, but you can use multiple passive scalars to examine the effect of mixing.
- Tracing how smoke or any other vapor would convect and diffuse in a room or any fluid domain.

  More specifically, you can model the dissolution of a gas in a liquid.

Contents:

Passive Scalars Workflow
Passive Scalar Model Reference
Example of Using a Passive Scalar for Residence Time
Passive Scalar Field Functions Reference

Passive Scalars Workflow

To set up passive scalar tracking, you choose the model on the physics continuum and then create one or more passive scalar objects. STAR-CCM+ exposes further controls to complete the definition.

For a single-phase flow, one of the following models must be activate within the physics continuum:

- Segregated Flow model
- Coupled Flow model
- Harmonic Balance model

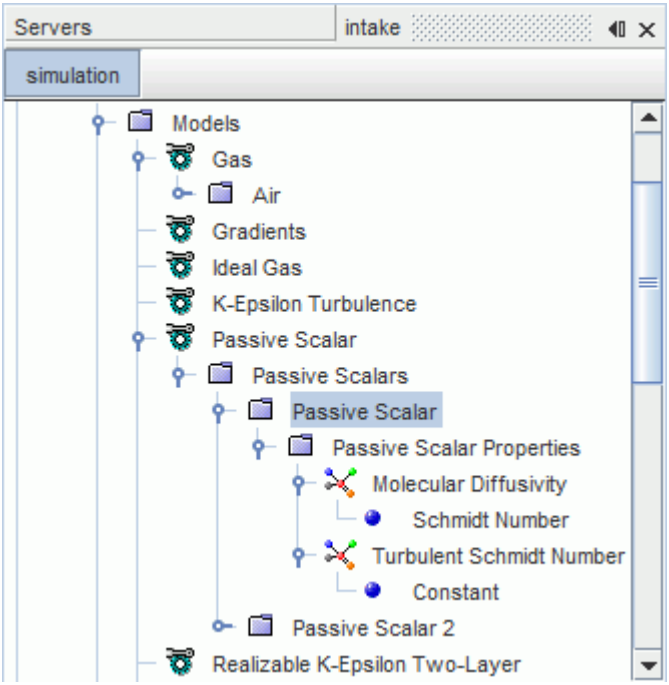
For a multiphase flow, you choose the passive scalar model within the phase model selection for each phase.

To set up passive scalar tracking within a flow simulation:

1. Within the *Optional Models* groupbox of the physics model selection dialog for a physics continuum (single phase flow) or an individual phase (multiphase flow), activate *Passive Scalar*.

2. Expand the Models manager for the physics continuum or phase, right-click the **Passive Scalar** > **Passive Scalars** node and select **New**.

   STAR-CCM+ adds a passive scalar object to the tree.



3. Give the passive scalar a name appropriate for its role within the simulation.
4. Select the **Passive Scalar** node and set its properties.

   Particular properties to consider are:

   - Clipping properties: these properties set minimum and maximum bounds on the values that a scalar can take
   - Transport: this property allows you to control which terms of the transport equations apply in the simulation

   See Passive Scalar.
5. Set the amount of passive scalar entering the simulation through inflow or outflow boundaries, if any. For example, on an inflow boundary, select the **Physics Values** > **Passive Scalar** node and set **Value** to `0.5`.
6. Each wall boundary contains the *Wall Passive Scalar Option*, which is set to **Zero Flux** by default. Change this option to set a specified value, or a specified flux, as required.
7. Provide an initial value for each passive scalar in the simulation.

Passive Scalar Model Reference

Provided By	**[physics continuum]** > **Models** > *Optional Physics Models*

Example Node Path	**Continua** > **[Fluid Continuum]** > **Models** > **Passive Scalar**	
Requires	*Flow*: any model	
Activates	Physics Models	None
	Model Controls (child nodes)	**Passive Scalars** manager
	Reference Values	None
	Initial Conditions	**Passive Scalar**
	Boundary Inputs	**Passive Scalar** value on flow boundaries, **Wall Passive Scalar Option** on walls. See Passive Scalar Boundary Conditions.
	Region Inputs	See Passive Scalar Region Source
	Solvers	Passive Scalar solver
	Monitors	One residual monitor per passive scalar
	Report Options	No specific reports
	Field Functions	For each passive scalar: **Diffusion Coefficient of [passive scalar]** **Diffusivity of [passive scalar]** **[passive scalar]**
Properties	See Model Properties below.	
Actions	On **Passive Scalars** manager node.	

Passive Scalar Model Properties

Secondary Gradients

Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces.

- **On**: Default value. Solves for interior and boundary types of secondary gradient.
- **Off**: Does not solve for either type of secondary gradient.
- **Interior Only**: Solves for the interior secondary gradients only.
- **Boundaries Only**: Solves for the boundary secondary gradients only.

Convection

In transport equations, you can choose from a range of schemes that calculate the convection term at a cell face. This calculation requires STAR-CCM+ to compute the face value of a quantity from the surrounding cell values. The method used for computing this face value has a profound effect on the stability and accuracy of the numerical scheme. For guidance on selecting a convection scheme, see Convective Flux.

- **1st-order**: First-order convection scheme.

- **2nd-order**: Second-order convection scheme.

The properties *Secondary Gradients* and *Convection* are available with the Segregated Flow and Coupled Flow models. The property *SUPG* is available with the Viscous Flow model.

Passive Scalar Solver

The passive scalar solves a transport equation, Eqn. (286), for each passive scalar defined in the simulation.

Under-Relaxation Factor

At each iteration, governs the extent to which the new solution replaces the old one. The default value is 0.8. For the theoretical background, see Eqn. (1625) in "Implicit Iteration."

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Relative Tolerance

Sets the relative tolerance of the solver.

Secondary Gradients

Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces.

- **On**: Default value. Solves for interior and boundary types of secondary gradient.
- **Off**: Does not solve for either type of secondary gradient.
- **Interior Only**: Solves for the interior secondary gradients only.
- **Boundaries Only**: Solves for the boundary secondary gradients only.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Common Values

Passive Scalar

Amount of passive scalar, set as either a constant array profile or a composite array profile.

Passive Scalar Source

Source term setting for the passive scalar, either **Mass Flux**, using a source term alone, or **Scalar rate with inferred density**, using a source term and automatically using the density of the material composing the phase.

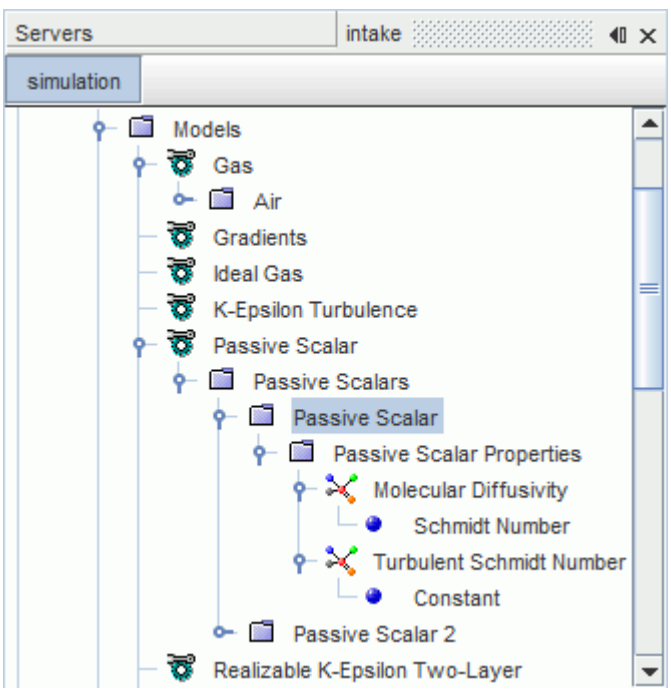
Contents:

Managing Passive Scalars
Setting an Initial Value of Passive Scalar
Setting Passive Scalar at Boundaries
Setting a Passive Scalar Region Source

Passive Scalar Material Node

A **Passive Scalar** node represents a passive scalar material in the simulation tree. Its child objects function in the same way as standard material property methods.

Passive scalars use the Schmidt number function by default; this is useful for when you want the passive scalar to diffuse at a different rate from the fluid. Alternatively, the *Method* property of the **Molecular Diffusivity** node can be set to a field function. If you use turbulence, then an additional material property node is activated: **Turbulent Schmidt Number**. This can be set for a constant or a field function.



Passive scalar materials can be renamed or deleted.

Passive Scalar Properties

Number	A read-only property that indicates the number of passive scalar materials preceding the current one. For example, for the first passive scalar material that you create, this property would equal zero. These values remain unchanged even when you delete one or more of these objects.	
Maximum clip value	Sets the maximum scalar value used for clipping.	
Minimum clip value	Sets the minimum scalar value used for clipping.	
Clip	Clipping prevents scalar values from exceeding limits you set.	
	Off	No clipping applied.
	Below Min, Above Max	Prevents scalar values from rising above the *Maximum clip value*, and falling below the *Minimum clip value*.
	Below Min	Prevents scalar values from falling below the *Minimum clip value*.
	Above Max	Prevents scalar values from rising above the *Maximum clip value*.
Transport	Provides the option to deactivate the convection and/or diffusion term in Eqn. (286). The source term is always active. The transient term is only active for transient simulations.	
	Convection and Diffusion	Both diffusion and convection are activated.
	Convection only	The diffusion term is deactivated.
	Diffusion only	The convection term is deactivated.
	No Convection or Diffusion	Both convection and diffusion terms are deactivated.
Use GGDH	Uses the Generalized Gradient Diffusion Hypothesis. A Reynolds Stress Turbulence model must be activated first.	
	Activated	Use GDDH in calculating passive scalar transport.
	Deactivated	Do not use GDDH. The default.

Passive Scalars Manager Pop-Up Menu

Use this pop-up menu to add passive scalar material objects to the simulation.

New	Adds a passive scalar material to the simulation.

Setting an Initial Value of Passive Scalar

Initial values of passive scalars can be set using the **Passive Scalar** node that appears in the **Initial Conditions** folder of physics continua. This value can also be set on a region when custom initial conditions are selected.

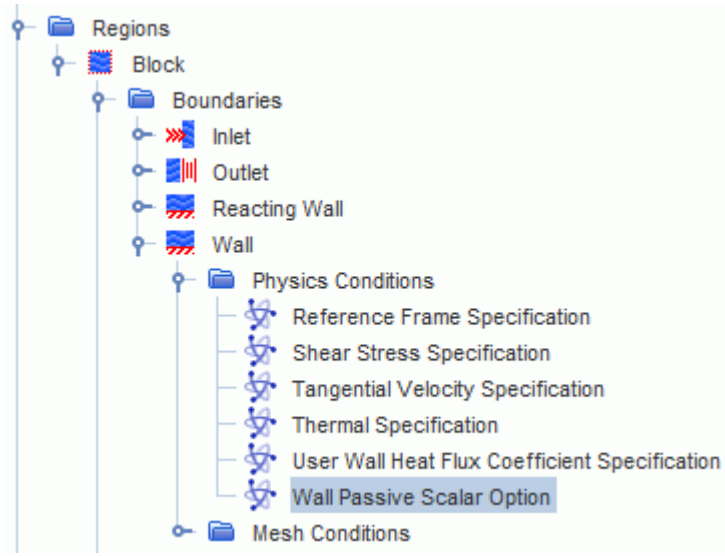
Passive Scalar Properties

Value	The value and units of the passive scalar initial condition. Clicking the customizer executes the define value action.

Setting Passive Scalar at Boundaries

For flow boundaries, you can set the amount of passive scalar crossing the boundary as a simple value. For wall boundaries, a choice is presented between a simple value, zero flux, specified flux, and flux derivative.

This choice is made on the **Wall Passive Scalar Option** node that is located within the **Physics Conditions** for a wall boundary. Select this node to access the properties of this option.



Wall Passive Scalar Option Properties

Wall Passive Scalar	Defines whether you want to enter a scalar value or a flux for this condition.	
	Specified Value	Specify a scalar value term. A Passive Scalar Value node is added to the **Physics Values** node.

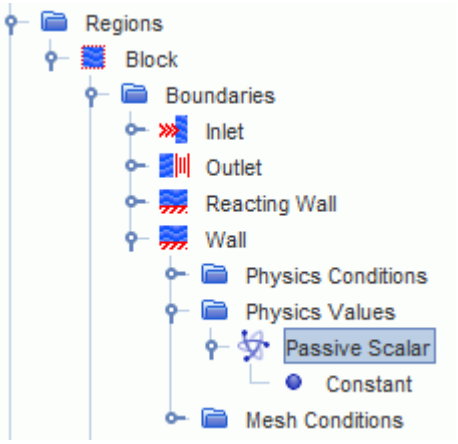
	Specified Flux	Specify a flux. A **Physics Values** > **Passive Scalar Flux** node and a **Physics Conditions** > **Wall Passive Scalar Flux Derivative Option** node appear.
	Zero Flux	Do not specify a value.

Wall Passive Scalar Flux Derivative Option Properties

Method	Allows you to specify a Passive Scalar Flux Derivative	
	None	No option is provided to specify a Passive Scalar Flux Derivative.
	Specified	A **Physics Values** > **Passive Scalar Flux Derivative** node appears which allows you to specify a Passive Scalar Flux Derivative using a scalar array profile.

Setting Passive Scalar as a Value

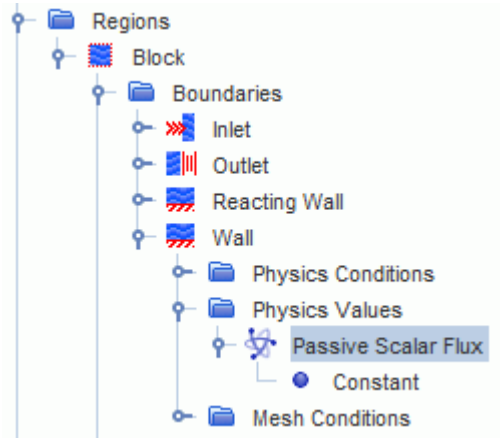
Passive scalar values can be set at inflow and outflow boundaries using the **Passive Scalar** node. This node is added to their **Physics Values** folders when the Passive Scalar model is selected. It is entered as a scalar array profile. For Eulerian and Lagrangian phases, the node is supplied for each phase containing passive scalars.



This node can also be added to a wall boundary depending on the setting of the Wall Passive Scalar Option condition of that boundary.

Setting Passive Scalar as a Flux

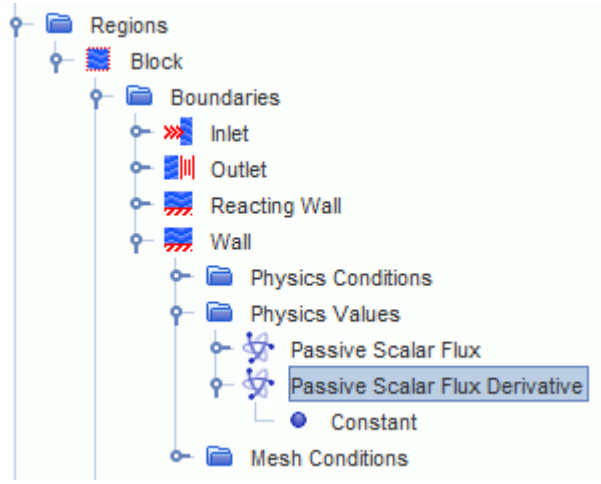
Passive scalar can be set as a flux at a wall boundary using the **Passive Scalar Flux** node. This node can be added to a wall boundary depending on the setting of the Wall Passive Scalar Option condition of that boundary. It is entered as a scalar array profile.



Note: The passive scalar flux value is the mass flux, not the volumetric flux.

Setting Passive Scalar as a Flux Derivative

You can set passive scalar as a flux derivative at a wall boundary using the **Passive Scalar Flux Derivative** node. Activating the derivative speeds up convergence. When the derivative is deactivated, oscillations occur in the passive scalar values near the wall and the solution does not converge fully.

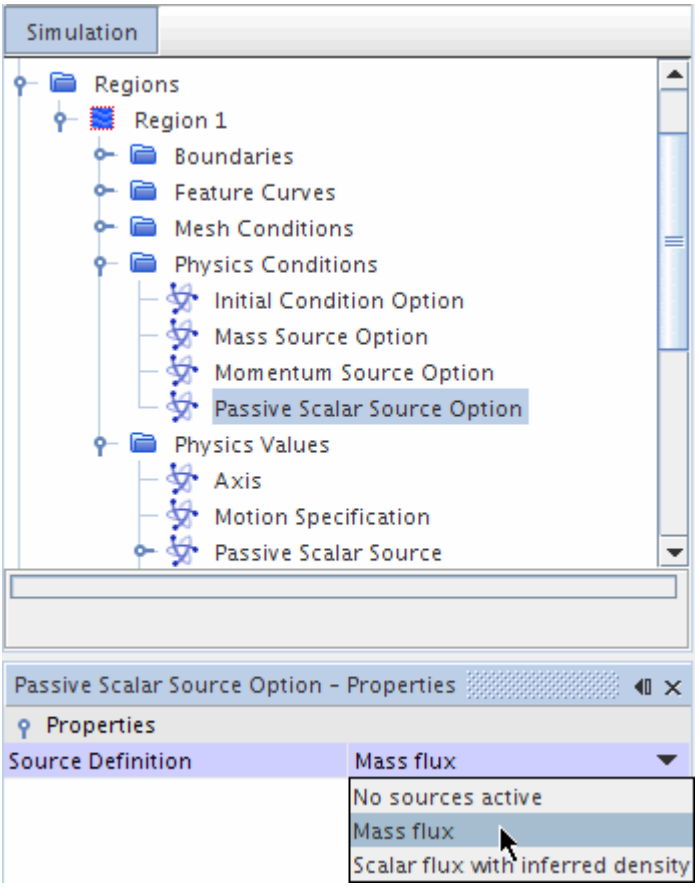


- If a field function which depends on the Passive Scalar value at the wall defines the Passive Scalar Flux at the wall, define the Passive Scalar Flux Derivative using a field function.
- If a constant Passive Scalar Flux, or a flux which is independent of the Passive Scalar value is defined at the wall, the Passive Scalar Flux Derivative is unnecessary.

Setting a Passive Scalar Region Source

To define a passive scalar source within a region with a single phase flow:

1. Select the **Physics Conditions** > **Passive Scalar Source Option** node for the region and **Mass flux** or **Scalar flux with inferred density** for *Source Definition*. See Passive Scalar Source.

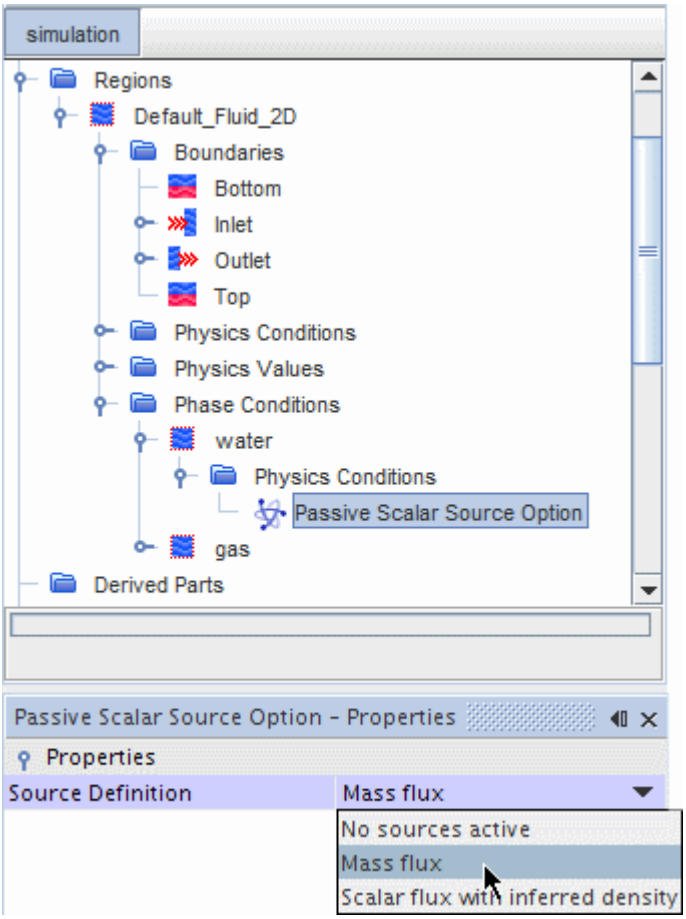


Two nodes are added to the **Physics Values** folder for the region: the Passive Scalar Source and Passive Scalar Source Derivative nodes.

2. Use these nodes to set the value and derivative of the source that you wish to set.

To define a passive scalar source within a region with a multiphase flow:

3. Select the **Physics Conditions** > **Passive Scalar Source Option** node for the region and select **Mass flux** or **Scalar flux with inferred density** for *Source Definition*. See Passive Scalar Source.



Two nodes are added to the **Physics Values** folder for the phase: the Passive Scalar Source and Passive Scalar Source Derivative nodes.

4. Use these nodes to set the value and derivative of the source that you wish to set.

The source term S_{ϕ_j} in Eqn. (286) is not multiplied internally by the volume fraction. When you set the scalar source for an Eulerian or Lagrangian phase in a multiphase simulation, include the phase volume fraction explicitly in the implementation of the source. For example, a passive scalar source term for air in a two-phase Eulerian multiphase simulation containing air and water is:

$VolumeFractionAir*$DensityAir

Passive Scalar Source

Use this node to set the source term S_{ϕ} in Eqn. (286). It is entered as a scalar array profile. Include the phase volume fraction explicitly for an Eulerian or Lagrangian phase.

Passive Scalar Source Derivative

This node represents the derivative of each component of the Passive Scalar Source regarding the corresponding passive scalar component,S_{ϕ_j} given in Eqn. (286). The derivative of the user source term is used to linearize Eqn. (286). Its value is set to zero by default. Providing a value for the derivative helps stabilize the solution, assuming that the source is set by a differentiable equation. It is entered as a scalar array profile.

Example of Using a Passive Scalar for Residence Time

You can use the Passive Scalar model to simulate residence time (the mean age of air) for a single phase.

Consider a chamber with air flowing in at one opening and flowing out at another, modeled as a steady-state Eulerian flow. A passive scalar is defined on the air, increasing steadily over time in value. In effect, defining the passive scalar attaches a clock to each volume element of air. These virtual clocks can be used to measure the residence time of the air in different parts of the chamber.

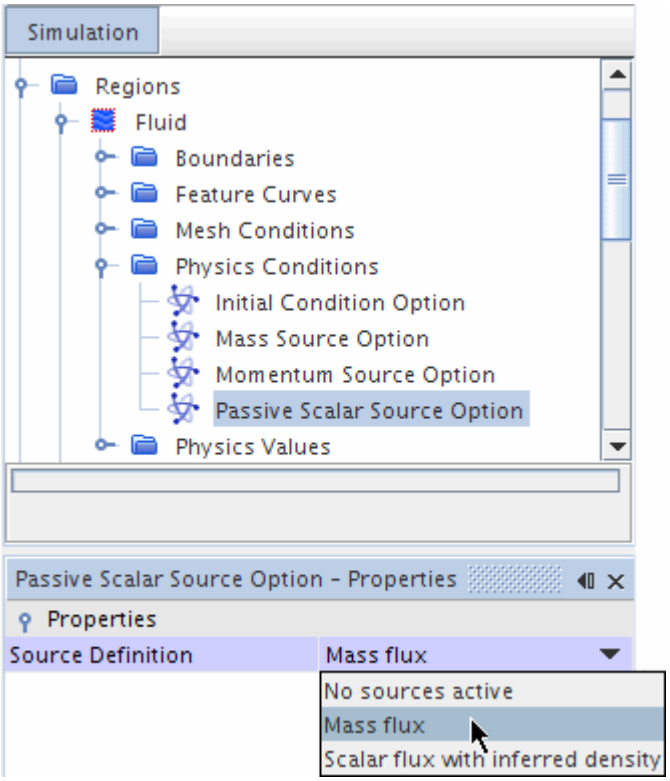
Example procedure:

- Set up all your necessary models, boundary conditions, and initial conditions.
- Activate the Passive Scalar model.
- Create a passive scalar and rename it **ResidenceTimeAir**.
- For all wall boundaries, set the *Physics Conditions* > *Wall Passive Scalar* option to **Zero Flux**.
- Create a field function and rename it **ResidenceTimeAirSource**.
- In the definition of the field function, supply this syntax:

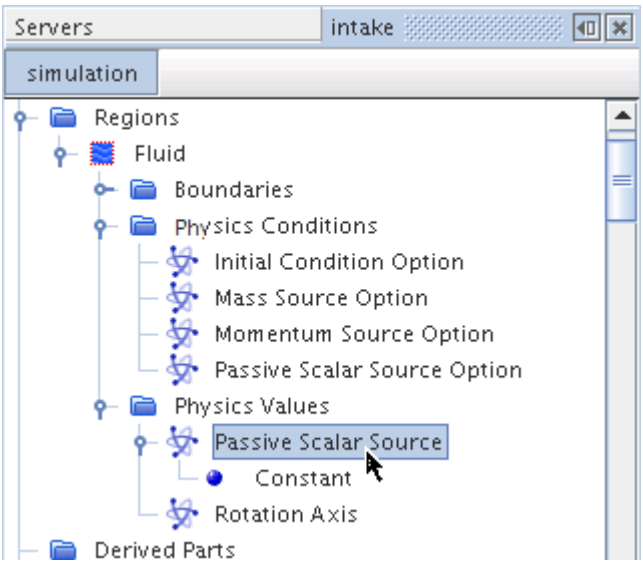
  ```
  ($ResidenceTimeAir >1000)?0:$Density
  ```

  The value of 1000 represents a maximum time, which is necessary if the flow field has a vortex or recirculation. Otherwise, time would grow to infinity.

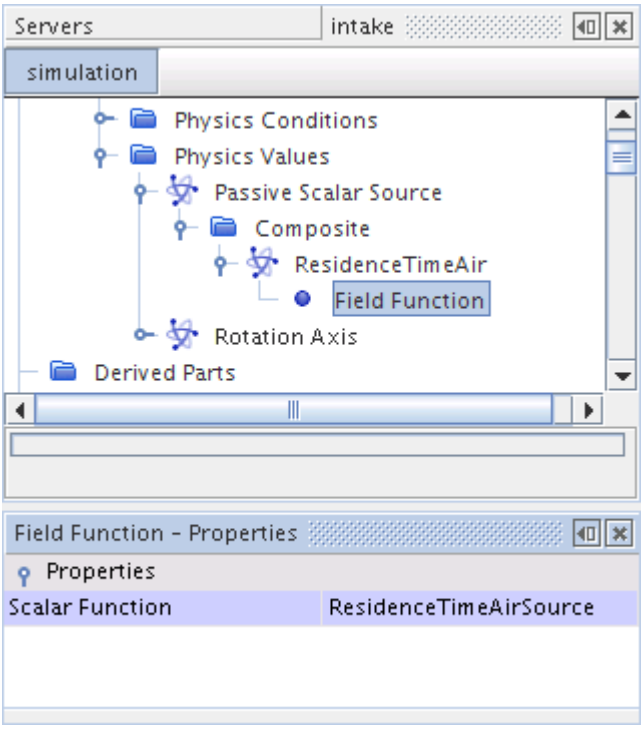
  The `$Density` value is present to scale the source term in the passive scalar transport equation.
- Select the **Regions** > **Fluid** > **Physics Conditions** > **Passive Scalar Source Option** node and select **Mass flux** for *Source Definition*. (See Passive Scalar Properties.)



- Open the **Regions** > **[Region]** > **Physics Values** node and select the Passive Scalar Source node.



- In the *Method* property, select Composite.
- Open the **Composite** node and select the **ResidenceTimeAir** node.
- In the *Method* property of the **ResidenceTimeAir** node, select *Field Function*.
- Select the **Field Function** node and set its *Scalar Function* property to *ResidenceTimeAirSource*, which is the field function that you defined.



- Create a scalar scene to display *ResidenceTimeAir*.

The simulation is ready to run.

A similar approach can be used for unsteady simulations.

Passive Scalar Field Functions Reference

The following primitive field functions are made available to the simulation when the Passive Scalar model is used. These field functions have a typical set of base properties.

Boundary Passive Scalar Flux of <passive scalar>	The rate per unit area at which a passive scalar quantity flows across a boundary.
Diffusion Coefficient of <passive scalar>	Diffusion coefficient of a particular passive scalar material, relating the diffusive transport of the material to the gradient. It has units of mass/length/time.
Diffusivity of <passive scalar>	Diffusivity of a particular passive scalar material, giving the rate at which it diffuses into another material. It has units of length^2/time. This field function is activated only when you select the laminar viscous option. This can use the Schmidt number method or a field function.
<Passive Scalar>	The field function object that corresponds to a particular passive scalar material. This object makes the passive scalar available as a field function.
Turbulent Schmidt Number of <passive scalar>	Turbulent Schmidt Number of a particular passive scalar material. This field function is activated only when you select the turbulent viscous option. This can be a constant or a field function.

Modeling Gravity

STAR-CCM+ can account for the effect of gravitational acceleration on materials in a physics continuum.

For fluids, it provides two effects:

- The working pressure becomes the piezometric pressure.
- The body force due to gravity is included in the momentum equations.

  The Gravity model also affects other types of simulation, such as material particle flows under the Lagrangian Multiphase framework.

For constant density flows, you can additionally select the Boussinesq model. The Boussinesq model provides a buoyancy source term that only applies when there are small variations of density due to temperature variations.

Working Pressure

When the Gravity model is active, the STAR-CCM+ working pressure becomes the piezometric pressure, in which the reference altitude (which you set) is taken into account. The piezometric pressure is given as:

$$p_{piezo} = p_{static} - \rho_{ref}\,\mathbf{g}\cdot\left(\mathbf{x}\text{-}\mathbf{x}_0\right)$$

For both variable and constant density flows it is necessary to set the reference altitude $\mathbf{x}_0$, which is used in the calculation of piezometric pressure as shown above. The reference altitude is the point at which the piezometric and static pressure are considered equal.

When setting initial or boundary pressures in simulations for which the Gravity model is active, it is the piezometric pressure value that is being set. STAR-CCM+ calculates the corresponding static pressure.

Body Force Source Term

The Gravity model permits the inclusion of the body force due to gravity in the momentum equations when using the Segregated Flow model or the Coupled Flow model.

For variable-density flows (for example, when using the Ideal Gas model or the Polynomial Density model), the buoyancy effects due to gravity and density variation are modeled directly and are always present. For problems that use the Constant Density model, the buoyancy effects due to small temperature and, so, small density variations, are only included when the optional Boussinesq Model is selected in addition to the Gravity model. The formulation of these source terms is presented below.

For Solids, the Gravity model permits the inclusion of a body force source term, $\mathbf{b}$ in the solid momentum equation Eqn. (3815). This body force is calculated and applied to each cell of the solid continuum.

Contents:

Gravity Model Reference

Boussinesq Model Reference

See Also:

Pressure

Gravity Model Reference

The Gravity model accounts for the action of gravitational acceleration in STAR-CCM+simulations.

Summary

Location within the model selection dialog	**[physics continuum]** > **Models** > *Optional Physics Models*
Example object path	**Continua** > **[Water Continuum]** > **Models** > **Gravity**
Pre-requisites	Physics continuum models: • Segregated Flow or • Coupled Flow
Adds	Models: • Boussinesq Model Reference nodes: • Gravity • Reference Altitude • Reference Density
Properties	None
Actions	None

Reference Values

The working pressure at pressure boundaries has to be consistent with the selected reference altitude, reference density, and reference pressure. If any of these values are changed, it is necessary to make sure that the working pressures set on pressure boundaries are still consistent.

Gravity

Value of gravity $\mathbf{g}$ appearing in equations Eqn. (1572) and Eqn. (1573). Set as a constant vector value.

Reference Altitude

Value h_{ref} in Eqn. (1571) and Eqn. (1571). At this point, the static and piezometric pressures are equal. Set as a constant vector value.

Although its exact position is arbitrary (similar to the values of reference pressure and reference density), it is reasonable to locate it at a central point within the solution domain. This means that the expected range of values for working pressure is balanced around zero, and so provide an optimum condition for the STAR-CCM+ numerical schemes.

Reference Density

Value ρ_{ref} in Eqn. (1572). Required for variable density flows.

Boussinesq Model Reference

The Boussinesq model provides a buoyancy source term that only applies when there are small variations of density due to temperature variations.

Summary

Obtain from	**[physics continuum]** > **Models** > *Optional Physics Models*
Example Object Path	**Continua** > **[Water Continuum]** > **Models** > **Boussinesq Model**
Requires	Physics continuum models: • Gravity • One from *Energy*
Adds	Reference nodes: • Reference Temperature Material property: • Thermal Expansion Coefficient
Properties	None
Actions	None

Reference Values

Reference Temperature

Value T_{ref} in Eqn. (1570).

Material Properties

Thermal Expansion Coefficient

Coefficient of bulk thermal expansion, β in Eqn. (1570).

Modeling Axisymmetric Swirl

The Axisymmetric Swirl model models swirling or rotating flow about a central axis. You can use this model when your simulation includes a swirling flow from an inlet such as a fan interface.

The Axisymmetric Swirl model accounts for the rotational motion in the momentum equations when using the Segregated Flow model or the Coupled Flow model.

For more information, refer to the following:

- Selecting the Axisymmetric Swirl Model
- Using the Axisymmetric Swirl Model
- Axisymmetric Swirl Model Formulation

Contents:

Selecting the Axisymmetric Swirl Model
Using the Axisymmetric Swirl Model
Axisymmetric Swirl Model Formulation

Selecting the Axisymmetric Swirl Model

The Axisymmetric Swirl model is available for gas, liquid, or multiphase axisymmetric region. This model can be used with the Segregated Flow model or the Coupled Flow model.

To select the Axisymmetric Swirl model in the *Physics Model Selection* dialog:

1. In the *Space* box, select **Axisymmetric**.
2. In the *Material* box, select **Gas**, **Liquid**, or **Eulerian Multiphase**.
3. Make the appropriate flow selection:
   - For Gas or Liquid materials, in the *Flow* box, select **Coupled Flow** or **Segregated Flow**.
   - For Eulerian Multiphase materials, in the *Eulerian Multiphase Model* box, select **Eulerian Multiphase Mixture**, **Multiphase Segregated Flow**, or **Volume of Fluid (VOF)**.

The **Axisymmetric Swirl** model is available for selection under *Optional Models*.

After you select the Axisymmetric Swirl model, you must also select an *Equation of State* model, a *Time* model, and a *Viscous Regime* model.

When you close the *Physics Model Selection* dialog, the Axisymmetric Swirl model is shown in the list under the **Continua** > **Physics** > **Models** node. The model has no user-defined properties.

Using the Axisymmetric Swirl Model

In an axisymmetric space, all three velocity components are functions of only the axial and radial coordinates, *z* and *r*. You define the swirling flow in your simulation by specifying the appropriate *z* and *r* components of the velocity field.

The fluid rotational motion is typically from a swirling inlet such as a fan interface. You model this swirling flow by setting the appropriate boundary conditions at the relevant inlets and walls.

To specify the swirl in your simulation, do the following for each inlet that contributes to the swirling flow:

- If necessary, create the field functions, tables, or user code that you want to use to define the velocity field at the inlet boundary.
- Under **Regions** > **[Region Name]** > **Boundaries** > **Inlet** > **Physics Conditions**, select the **Velocity Specification** node.
- In the *Properties* window, in the **Method** property drop-down menu, select **Components**.
- Under **Regions** > **[Region Name]** > **Boundaries** > **Inlet** > **Physics Values**, select the **Velocity** node.
- In the *Properties* window, in the **Method** property drop-down menu, select the appropriate method.

  The velocity is set using a vector profile. You can define the swirling flow by using field functions, a table, or user code.
- Under the **Velocity** node, select the sub-node and then, in the *Properties* window, specify the appropriate function or table.

At high rotation velocities, the strong coupling between the momentum equations can make the flow difficult to solve. The high rotation velocity creates a large radial pressure gradient which, in turn, affects the swirl in the fluid. You may find it necessary to first solve the flow for a low rotational velocity, and then gradually increase the swirl velocity until you reach the appropriate values.

Axisymmetric Swirl Model Formulation

The equations being modeled by the Axisymmetric Swirl model are the 2D conservation equations for mass and momentum.

In a cylindrical coordinate system, the 2D conservation equation is:

$$\frac{\partial}{\partial t}\int_V \rho \boldsymbol{u} r \,\mathrm{d}\, V + \oint \left(\rho \boldsymbol{u} \otimes \boldsymbol{u} + p\boldsymbol{I}\right) \cdot r \; d\mathrm{a} \; = \oint T \cdot r \; d\mathbf{a} \; + \int \frac{1}{r} \begin{bmatrix} 0 \\ p + \rho {u_\theta}^2 - \tau_{\theta\theta} \\ -\rho u_r u_\theta + \tau_{\theta r} \end{bmatrix} r \,\mathrm{d}\, V \tag{262}$$

where

$u = \left(u_z u_r u_\theta\right)^T$

and

$\frac{\partial}{\partial \theta}(\,) = 0$

and

$A_\theta = 0$

and T is the stress tensor

$$T = \mu \begin{bmatrix} 2\frac{\partial u_z}{\partial z} - \frac{2}{3}\nabla\cdot u & \frac{\partial u_z}{\partial r} + \frac{\partial u_r}{\partial z} & \frac{\partial u_\theta}{\partial z} \\ \frac{\partial u_z}{\partial r} + \frac{\partial u_r}{\partial z} & 2\frac{\partial u_r}{\partial r} - \frac{2}{3}\nabla\cdot u & \frac{\partial u_\theta}{\partial r} - \frac{u_\theta}{r} \\ \frac{\partial u_\theta}{\partial z} & \frac{\partial u_\theta}{\partial r} - \frac{u_\theta}{r} & 2\frac{u_r}{r} - \frac{2}{3}\nabla\cdot u \end{bmatrix}$$

and I is the identity matrix.

Modeling Fully Developed Flow and Energy

This section explains what fully developed flows are, presents formulation of the conditions for fully developed energy and the algorithm, and describes situations where errors are incurred. Information is provided about variable heating situations, validation, and recirculation.

What Are Fully Developed Flows?

“Fully developed flows” occur in long tubes or tube banks. For simple flows without heat transfer or transported scalars, all flow variables except pressure can be treated as periodic.

Pressure is typically modeled with a “jump” condition at the interface that compensates for the streamwise pressure drop through the system.

When energy is present, a fully developed situation can be deemed to occur if the profiles of temperature and/or enthalpy become self-similar when using an appropriate scaling. In these cases, the self-similarity relations can be used to rescale the thermal field leaving the domain in order to reintroduce it at the inflow periodic boundary. This rescaling is done in STAR-CCM+ by using the Fully Developed Flow Option. The treatment of the thermal field is more complex than pressure, since both convection and diffusion are taken into account. In addition, the limitations on which physical models are present in order for the fully developed condition to occur must also be considered.

In Which Situations Will Errors Be Incurred?

Errors occur with mesh issues, compressibility effects, property variations, and non-contiguous regions.

- **Mesh Issues**

  In addition to regular mesh refinement considerations, particularly in the wall-normal direction where large gradients occur, care must be taken to resolve the flow in the streamwise direction. Unlike simple fully developed duct flows without energy, which can often use as few as 3 - 5 cells in the streamwise direction, a larger number of cells is required when energy is considered. This requirement is true of the constant wall temperature heating condition, since the fluid heats up rapidly as the flow enters the domain. Grid refinement studies are recommended to confirm that the grid resolution is adequate.
- **Compressibility Effects**

  In truly compressible flows, viscous dissipation causes the fluid to heat up. With adiabatic walls, this heating effect causes the flow to accelerate. This scenario is called “fanno flow.” The fully developed energy treatment in STAR-CCM+ is not designed for fanno flows or compressible duct flows with other heating situations. However, use of the ideal gas law is not precluded and if employed, the Mach number must be sufficiently low that these effects do not dominate and cause stability problems.
- **Property Variations**

  The existence of a similarity solution for energy in a fully developed duct flow is most easily demonstrated for constant property flows. However, the use of temperature-dependent fluid properties is not precluded, and small property variations do not have a large effect on the solution. However, caution must be exercised when using variable properties in case they affect the accuracy or stability of the solution.
- **Non-Contiguous Regions**

STAR-CCM+ permits regions to be non-contiguous, in general. However, the fully developed energy algorithm must search through all regions that are connected to the interface to evaluate contributions from associated cells and boundaries. For this reason, it is recommended that non-contiguous regions are not used. If in any doubt, use the Split Non-Contiguous function on the region.

What About Variable Heating Situations?

The similarity solutions that cause a fully developed energy situation in a duct are limited to:

- Constant wall heat flux
- Constant wall temperature

However, given the flexibility of the boundary conditions in STAR-CCM+, no attempt is made to ensure that the values specified for wall heat flux or temperature are indeed constant. If a boundary condition is specified that deviates from the profile scaling that is used at the fully developed interface, an error is incurred. The significance of the error clearly depends on how incorrect the assumption is.

The specified wall heat flux situation is likely to be more forgiving than the specified wall temperature situation because the linear behavior of the energy equation permits superposition of solutions. Therefore a wall heat flux that varies periodically (such as a sinusoidal wave) most likely produces a well-behaved solution.

How Can One Verify That Fully Developed Energy Is Valid for a Specific Heating Situation?

The similarity solutions that cause a fully developed energy situation in a duct are limited to:

- Constant wall heat flux
- Constant wall temperature in **constant property flows**

Any deviations from these situations cast doubt on the validity of the approach. One way to check that the fully developed approach is valid is to create a simplified version of the domain and manually replicate it 20 or more times. Then, verify that the velocity and thermal fields are indeed periodic (using the appropriate scaling). This check requires a significant effort, but is necessary to remove residual doubt.

What Happens If Recirculation Occurs at the Fully Developed Interface?

Care has been taken to test the fully developed interface for an interface at which recirculating flow occurs. However, problems can arise if the portion of flow entering approaches the portion of the flow leaving. Therefore, if possible, it is preferable to place the interface at a location where recirculation does not occur.

Contents:

Creating a Fully-Developed Flow Simulation
How Does the Fully Developed Energy Algorithm Work?

See Also:

Fully Developed Interface

Creating a Fully-Developed Flow Simulation

This procedure explains how to set up and run a simulation to model fully developed turbulent flow of a fluid through a straight pipe in STAR-CCM+.

STAR-CCM+ allows you to model a fully-developed flow using a short section of pipe with periodic interfaces, rather than using a length of pipe with inlet and outlet boundaries.

1. Import the geometry that represents a short section of the interior volume of the pipe, or create it in 3D-CAD.
2. Make sure that the upstream and downstream flow boundaries are defined as separate part surfaces. If necessary, rename the part surfaces in 3D-CAD, or split them using one of the splitting actions.
3. Define the contact between the upstream and downstream flow boundaries.
   a) Within **Geometry** > **Parts**, multi-select the part surfaces that represent the upstream and downstream boundaries, right-click one of the selected nodes, and select **Create Periodic**.
   a) Select the **Contacts** > **[Part/Part]** > **[Periodic Contact]** > **Periodic Transformation** node and in the *Properties* window, set *Periodicity* to **Translational**.
4. Select the geometry part, right-click, and choose **Assign parts to regions...**
   a) In the *Assign Parts to Regions* dialog, select the following options:
      - **Create a Region for Each Part**
      - **Create a Boundary for Each Part Surface**
      - **Create One Feature Curve for all Part Curves**
   b) Make sure that **Create Interfaces From Contacts** is activated.
   c) Click **Apply** then **Close**.

An **Interfaces** > **[Periodic]** node appears. STAR-CCM+ recognizes that the two surfaces that form the periodic interface are from the same part and sets the *Interface Type* to **Internal Interface**. However, as you wish to model fully-developed flow, you must manually change this type:

5. Select the **Interfaces** > **[Periodic]** node and set *Type* to **Fully-Developed Interface**.
6. For the benefit of the prism layer mesher, set the upstream flow boundary type to Velocity Inlet, and the downstream flow boundary type to Pressure Outlet. These settings make sure that the prism layer mesher does not apply layers to these surfaces.
7. Mesh the geometry.
   Since you are only modeling a small portion of the pipe flow, the computational expense of refining the mesh is much reduced compared to modeling a longer section of pipe.
8. Create a physics continuum and select the appropriate physics models for the simulation. An example for turbulent flow is as follows:

Group box	Model
Material	**Liquid**
Flow	**Segregated Flow**
	Gradients (selected automatically)
Space	**Three Dimensional**
Equation of State	**Constant Density**

Group box	Model
Time	**Steady**
Viscous Regime	**Turbulent**
	Reynolds-Averaged Navier-Stokes (selected automatically)
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**
	Realizable K-Epsilon Two-Layer (selected automatically)
	Two-Layer All y+ Wall Treatment (selected automatically)

9. Define the fluid properties.
10. Define the target flow condition for the interface:
    a) Select the **Interfaces** > **[region/region]** > **Physics Conditions** > **Fully Developed Flow Option** and choose either **Pressure Jump** or **Mass Flow Rate** as the controlling factor for flow through the pipe section.
    b) Within the **Physics Values** node for the interface, select the corresponding **Pressure Jump** or **Mass Flow Rate** node and set the target value.
11. Set up relevant XY plots and scenes to display the values that you require.
12. Run the Simulation.

Fully Developed Flow and Energy Formulation

How Does the Fully Developed Energy Algorithm Work?

The basic mechanism for the fully developed energy algorithm is the rescaling of the temperature profile from the interface boundary at the outflow to use at the inflow interface boundary. This mechanism is different according to whether the constant temperature wall assumption or constant heat flux assumption is being used. For the constant temperature wall assumption, the scaling is:

$$\frac{T_w - T}{T_w - T_b} = \text{const} \tag{263}$$

so that the wall temperature and the bulk temperature must be specified.

For the constant heat flux assumption, the scaling law is:

$$T - T_b = \text{const} \tag{264}$$

so that bulk temperature must be specified.

Under Which Conditions Is the Fully Developed Assumption Valid for Energy?

Consider the flow in a duct in which the streamwise coordinate is x, and the cross-stream coordinate is y. By definition, fully developed heat transfer exists when the heat transfer coefficient h is independent of x:

$$h = \frac{\dot{q}_w}{T_w - T_b} = -\frac{k}{T_w - T_b}\frac{\partial T}{\partial y}\bigg|_w \neq f(x) \tag{265}$$

For a constant-property fluid, the bulk temperature T_b is defined:

$$T_b = \frac{1}{AU_m}\int_A uT \text{ d } A \tag{266}$$

and the mean velocity U_m is defined as:

$$U_m = \frac{1}{A}\int_A u \text{ d } A \tag{267}$$

It is assumed that u and U_m are independent of x, but T_b is a function of x. Since T_w and T_b are independent of y, the following is obtained:

$$Nu = \frac{hl}{k} = -\frac{\partial}{\partial(y/l)}\left(\frac{T_w - T}{T_w - T_b}\right)\bigg|_w \neq f(x) \tag{268}$$

implying the existence of a dimensionless temperature profile:

$$\theta(y/l) = \frac{T_w - T}{T_w - T_b} \neq f(x) \tag{269}$$

or equivalently, since $\partial\theta/\partial x = 0$:

$$\frac{\partial T_w}{\partial x} - \frac{\partial T}{\partial x} = \frac{T_w - T}{T_w - T_b}\left(\frac{\partial T_w}{\partial x} - \frac{\partial T_b}{\partial x}\right) \tag{270}$$

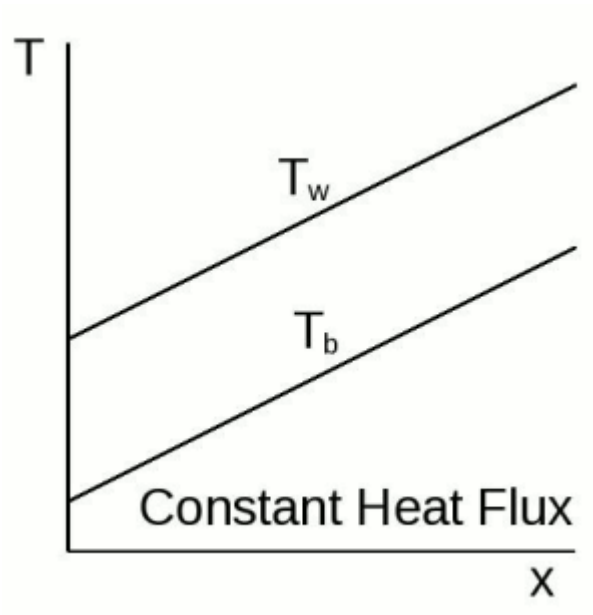
In the constant wall heat flux case, since both h and $\dot{q}''$ are constant, $T_w - T_b$ equals a constant, so that:

$$\frac{\partial T}{\partial x} = \frac{\partial T_w}{\partial x} = \frac{\partial T_b}{\partial x} \tag{271}$$

In other words, for a constant wall heat flux, both the bulk temperature and the wall temperature increase linearly with streamwise distance (see the figure of constant heat flux). This linear increase means that the temperature profiles at various streamwise locations differ only by a constant offset as follows:

$$T(r,x) - T_b(x) = T(r, x+L) - T_b(x+L) \tag{272}$$

This means that the problem can be solved without knowledge of the wall temperature, if one knows the increase in bulk temperature that is due to the (constant) wall heat flux.

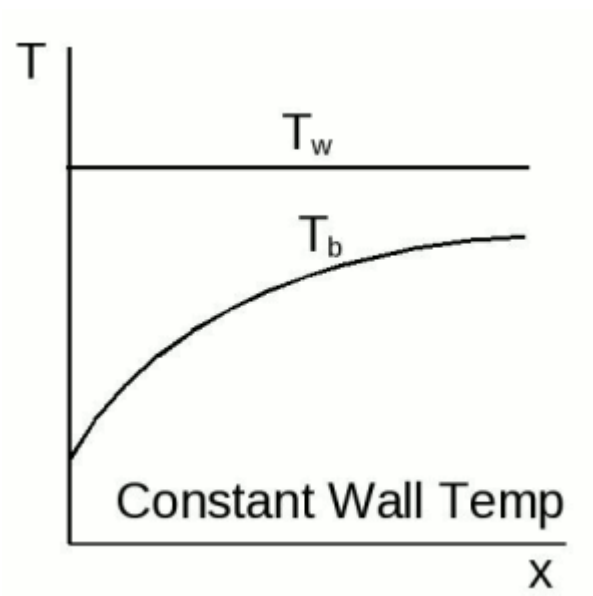


Behavior of bulk and wall temperatures for fully developed thermal fields with a constant heat flux

In the constant wall temperature case, the partial derivative of temperature in the streamwise direction in the energy equation, which results from the convective term, is replaced with:

$$\frac{\partial T}{\partial x} = \theta \frac{\partial T_b}{\partial x} \tag{273}$$

such that the rest of the energy equation can be written as a function purely of y/l. However, the temperature profiles must be scaled using both the bulk temperature and the wall temperature. Furthermore, the bulk temperature increases non-linearly with streamwise distance x in the duct, as shown in the following figure, resulting in a corresponding non-linear increase in local temperature.

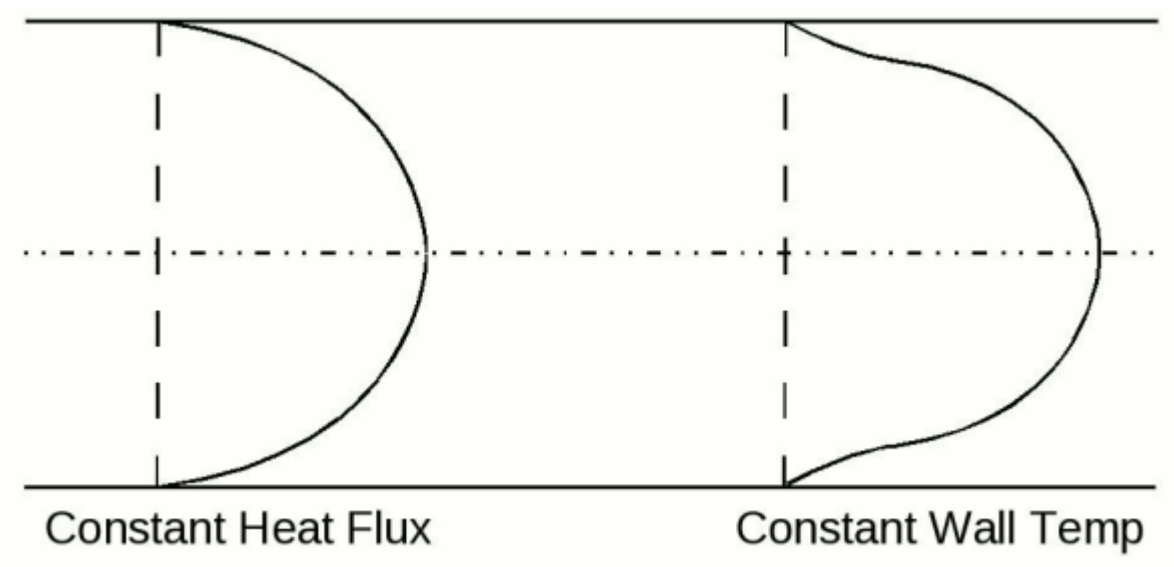


Behavior of bulk and wall temperatures for fully developed thermal fields with a wall temperature

Therefore this section defines two and only two situations that permit fully developed heat transfer in a duct:

- Wall with constant heat flux
- Walls with a constant temperature

The resulting temperature profiles can be different, depending on the geometry. The following figure shows the profile shapes for laminar constant-property flow through circular ducts.



Temperature profiles for fully developed thermal fields in laminar pipe flows

The constant heat flux boundary condition applies to a wall subject to electrical resistance heating or to a radiant flux. The constant temperature wall is approached when the outer wall is subject to phase change such as condensation or boiling. The two heating situations are also applicable to heat exchangers. The constant heat flux situation can correspond to a counterflow heat exchanger when the fluid capacity rates are the same. The constant temperature solution is applicable to any heat exchanger where one material has a very much larger capacity rate than the other. In addition to evaporators and condensers, the other material can be a heated solid of large thermal capacity.

Tracking Vortices in the Flow

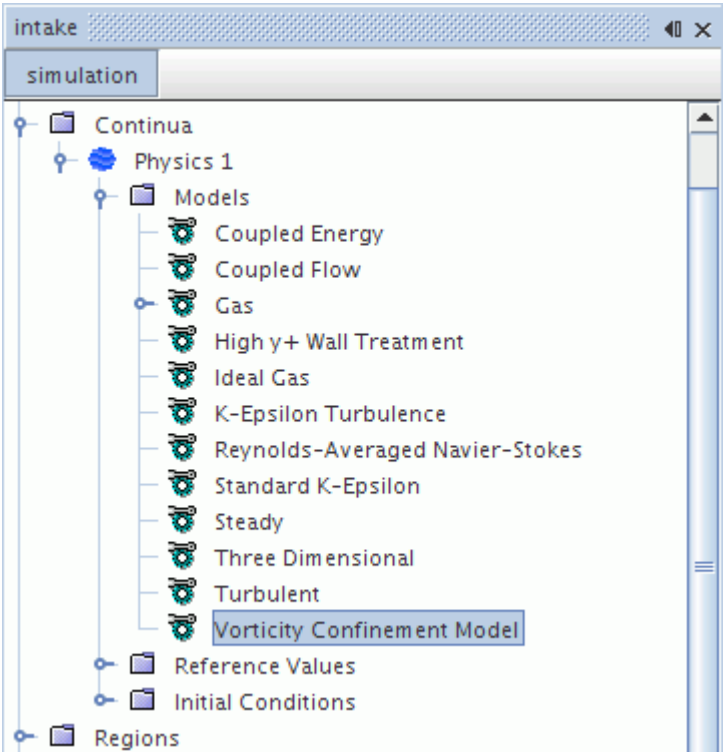
Many areas of engineering today require tracking isolated vortices accurately over long distances.

Such areas include:

- Aerospace—vortices of aircraft wing-tips, helicopter blades, and other components
- Submarines—tip vortices of various control surfaces
- Wind turbines—tip vortices and their environmental impact.

Unfortunately, due to the built-in dissipation of state-of-the-art numerical schemes that are used in modern CFD codes, any such simulation rapidly dissipates these vortices, unless a special treatment is applied.

One such treatment is the vorticity confinement concept that John Steinhoff et al. introduced ([237], [225]). Others have refined this concept [224]. It is implemented in STAR-CCM+ as the Vorticity Confinement model. This model is represented in the object tree by the **Vorticity Confinement Model** node, which has properties and is subject to common menu items.



An isolated vortex can be defined as the region in space with high values—and sometimes much higher values—of vorticity ($\mathrm{curl}(\mathbf{v})$), also referred to as “vortex tubes”. Tracking isolated vortices over long distances is important because of the induced velocity they produce. This velocity in turn influences the flow-field and aerodynamic processes—sometimes at great distances from the place where these strong isolated vortices have been produced. For example, helicopter blades generate strong blade-tip vortices which have to be either resolved or modeled (with a model such as the Vorticity Confinement model). Otherwise, the simulation of such a flow situation (especially of the rotor in ground effect) can be seriously inaccurate.

The current implementation of the Vorticity Confinement model in STAR-CCM+ is valid for both the Coupled Flow and Segregated Flow models in a low-speed flow regime ($\mathrm{Mach}_\infty \leq 0.3$). However, this model must be chosen with care. The $\mathbf{f}(d^*)$ factor (Eqn. (279)) is used to deactivate effectively the vorticity confinement force next to solid walls (for example inside a boundary layer) for viscous (laminar and turbulent) simulations. The model is deactivated next to the wall, as suggested by the authors [237], [225], [224]. Deactivating the model prevents the "vorticity confinement force" from affecting the boundary layer velocity distribution (and with it, factors such as skin friction).

This section includes instructions on selecting the Vorticity Confinement model, choosing the values for the Vorticity Confinement parameters, as well as its formulation and associated field functions.

Contents:

Vorticity Confinement Model Properties
Selecting the Vorticity Confinement Model
Choosing the Vorticity Confinement Model Parameter Values
Vorticity Confinement Formulation

Vorticity Confinement Model Properties

The Vorticity Confinement Model has the following properties:

Verbosity	Provides more output for this value while the simulation is running. This option can be useful for debugging problems when they occur.	
	Activated	Activates verbose output.
	Deactivated	Leaves output at its default.
Temporary Storage Retained	Retains or deletes temporary storage at the end of the iteration.	
	Activated	Retains temporary storage.
	Deactivated	Does not retain temporary storage.

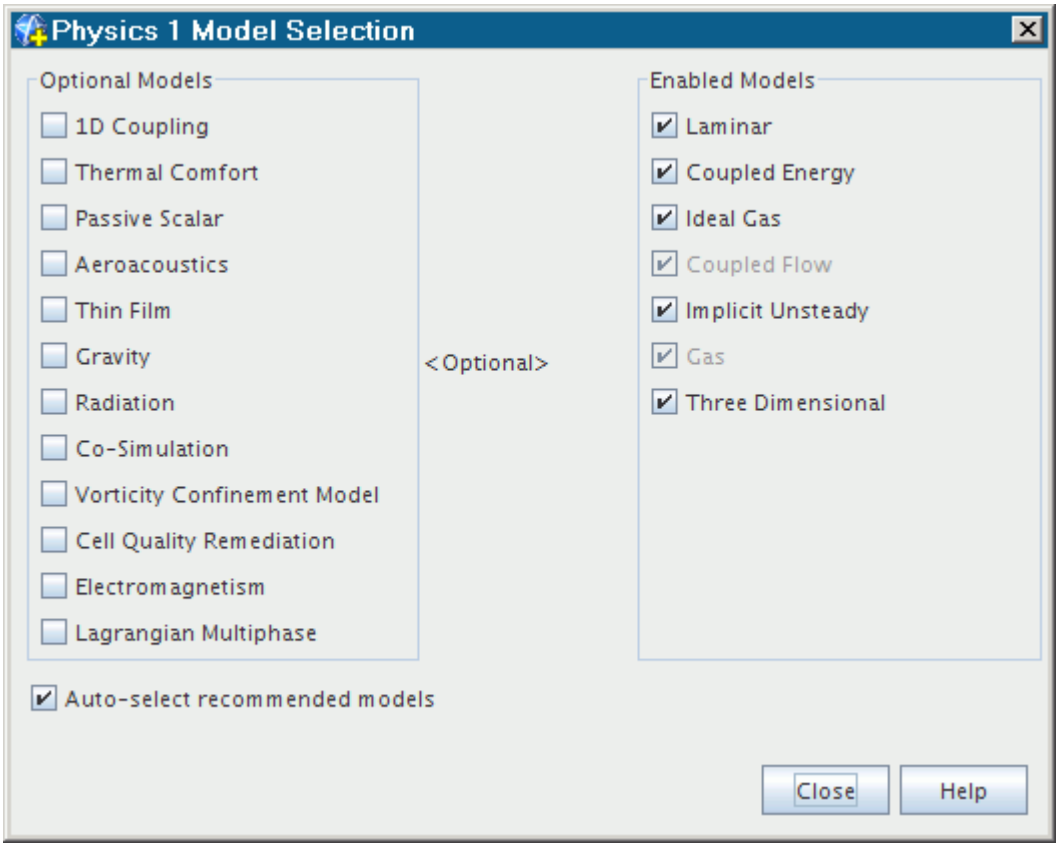
Expert Properties

Non-Dimensional Wall Distance Coefficient	Prevents the vorticity confinement term from acting inside boundary layers (d^*_{cutoff} in Eqn. (279)).	
Model Coefficient	Tunes the strength of vorticity confinement (ε_0 in Eqn. (275)).	

Selecting the Vorticity Confinement Model

To select the Vorticity Confinement model, first select the Segregated Flow model or select the Coupled Flow model.

The **Vorticity Confinement Model** radio button becomes available for selection under *Optional Models*.

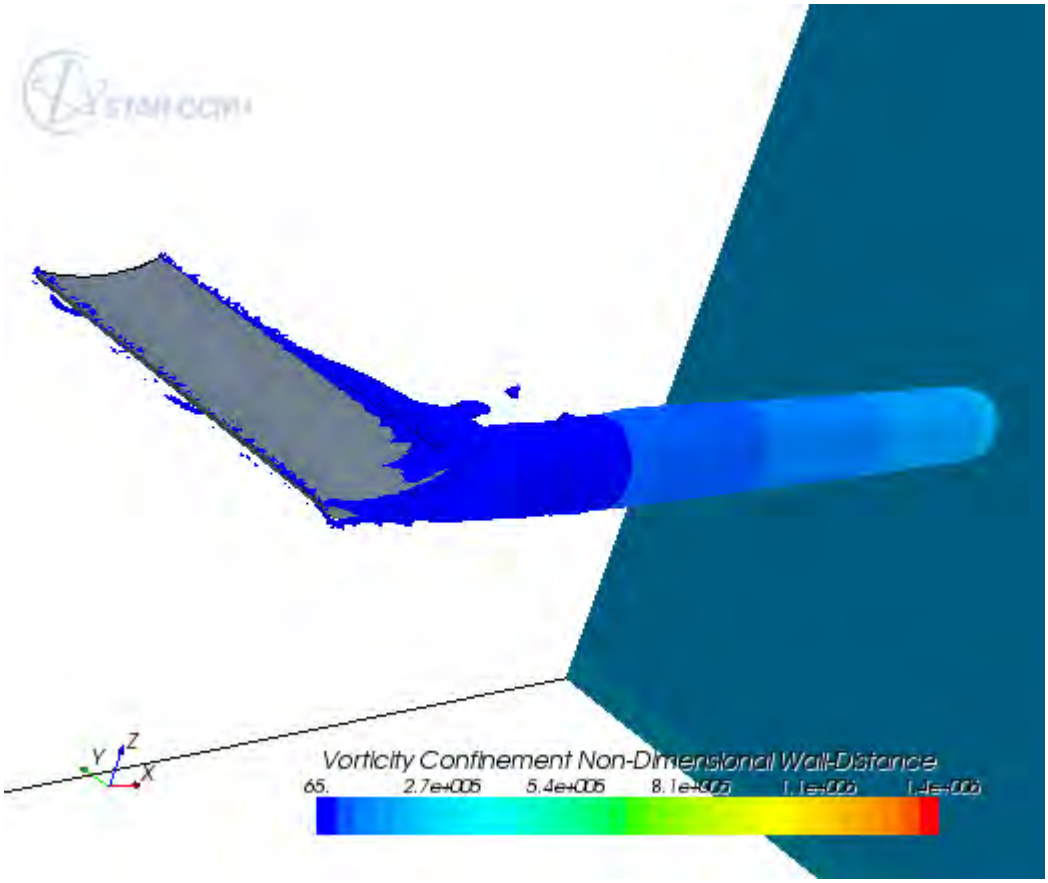


Choosing the Vorticity Confinement Model Parameter Values

To choose the values for the non-dimensional wall-distance coefficient d^*_{cutoff} and model coefficient ε_0 that are appropriate for your simulation:

1. Create a scalar scene and plot the field function **Vorticity Confinement Non-Dimensional Wall-Distance** on the parts of interest.

   In the following screenshot, this field function is plotted on the region of interest: the wing surface.

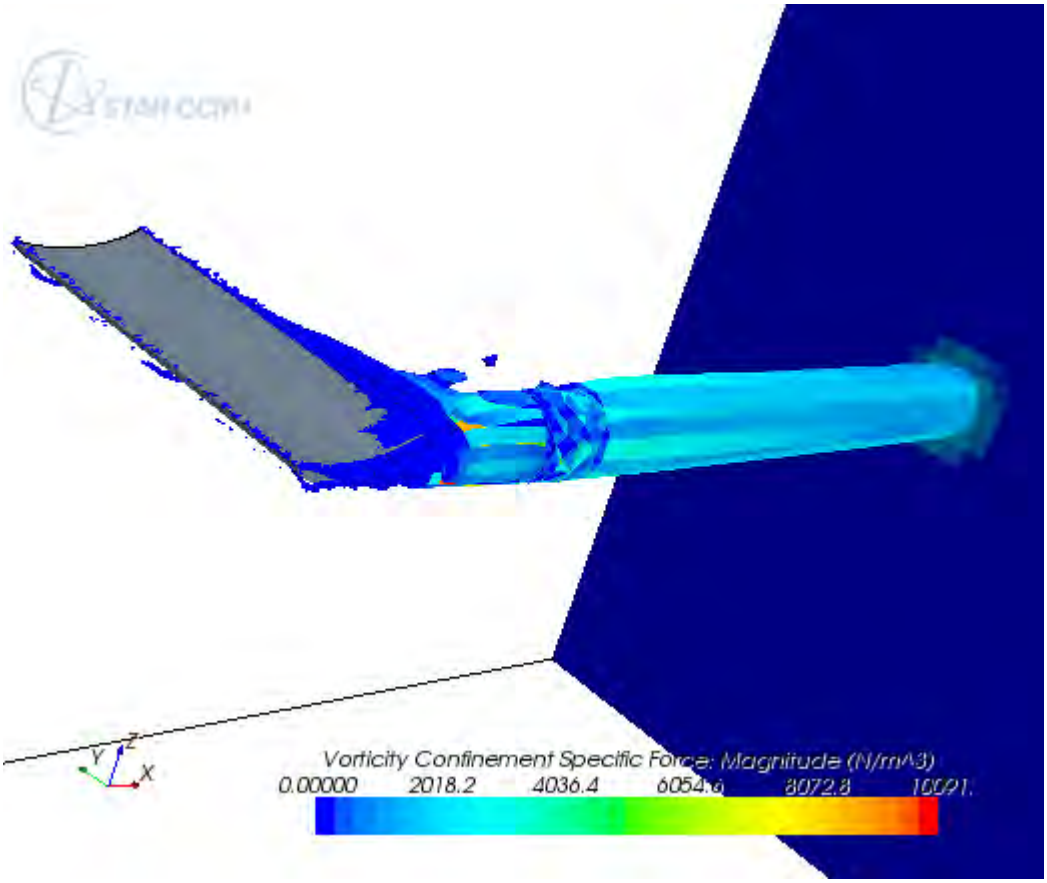


2. Set the non-dimensional wall-distance coefficient d^*_{cutoff} to the value that is appropriate for your case.

   In the above example, the computed vorticity confinement force is switched off near the wing surface. Therefore, d^*_{cutoff} is chosen to be 65.

3. Run the case.
4. Plot the **Vorticity Confinement Specific Force** field function and verify that:
   - The Vorticity Confinement specific force is zero inside the boundary layer.
   - The Vorticity Confinement specific force is sufficient to preserve the vortex for a sufficiently long distance. The actual value of this specific force depends on the simulation.

The Vorticity Confinement specific force plot for the above example is shown below.



Note: To increase the Vorticity Confinement specific force, you can increase the model coefficient ε_0.

Vorticity Confinement Formulation

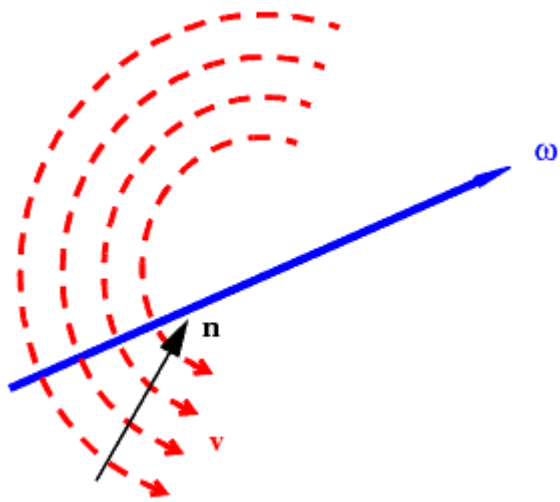
The basic approach of the Vorticity Confinement model consists of adding a force term $\mathbf{f}_\omega$ to momentum equations to help “confine” the vorticity.

Vorticity confinement is a force per unit volume $\left[N/m^3\right]$. Its purpose is to keep the vorticity from spreading. The source and its derivation can be found in these references: [224], [225], [237].

It is computed as:

$$\mathbf{f}_\omega = -\varepsilon\rho\left(\mathbf{n} \times \omega\right) \tag{274}$$

where $\omega = \mathrm{curl}(\mathbf{v})$, ρ is fluid density and $\mathbf{n}$ is the unit vector that is aligned with $\nabla\omega \ / \ |\nabla\omega|$. The following diagram depicts the basic spatial relationship among ω and the vectors $\mathbf{n}$ and $\mathbf{v}$.



(275)

where the model coefficient ε_0 (non-dimensional) is set to 0.04 in three-dimensional cases and 0.02 in two-dimensional cases by default, and:

$$V_{\omega_1} = |\nabla|\omega||\, h^2 \tag{276}$$

where h is the local grid size, computed as:

$$h = ND\sqrt{\mathrm{Vol}_{\mathrm{cell}}} \tag{277}$$

d^* is the non-dimensional wall distance:

$$d^* = W_d\left(\frac{V_{\mathrm{scale}}}{\nu}\right) \tag{278}$$

where W_d is wall distance, ν is kinematic viscosity of fluid, and V_{scale} is the mass average of velocity magnitude.

$$\mathbf{f}(d^*) \triangleq \min\left(\max\left(\left(\frac{d^* - d^*_{\mathrm{cutoff}}}{0.3d^*_{\mathrm{cutoff}}}\right), 0\right), 1\right) \tag{279}$$

where d^*_{cutoff} is user-specified non-dimensional wall distance.

Both the model coefficient ε_0 and non-dimensional cutoff wall distance can be retuned for different simulation. Default values have been found to be appropriate for external aerodynamics.

Vorticity Confinement Formulation

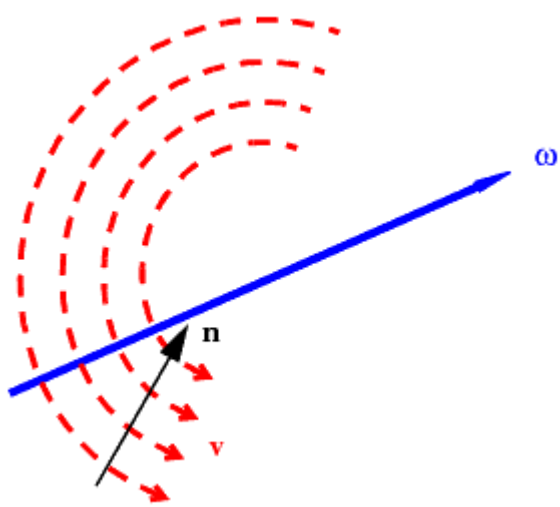
The basic approach of the Vorticity Confinement model consists of adding a force term $\mathbf{f}_{\omega}$ to momentum equations to help “confine” the vorticity.

Vorticity confinement is a force per unit volume $\left[N/m^3\right]$. Its purpose is to keep the vorticity from spreading. The source and its derivation can be found in these references: [224], [225], [237].

It is computed as:

$$\mathbf{f}_{\omega} = -\varepsilon\rho\left(\mathbf{n} \times \omega\right) \tag{280}$$

where $\omega = \mathrm{curl}(\mathbf{v})$, ρ is fluid density and $\mathbf{n}$ is the unit vector that is aligned with $\nabla\omega \ / \ |\nabla\omega|$. The following diagram depicts the basic spatial relationship among ω and the vectors $\mathbf{n}$ and $\mathbf{v}$.



(281)

where the model coefficient ε_0 (non-dimensional) is set to 0.04 in three-dimensional cases and 0.02 in two-dimensional cases by default, and:

$$V_{\omega_1} = |\nabla|\omega||\, h^2 \tag{282}$$

where h is the local grid size, computed as:

$$h = ND\sqrt{\mathrm{Vol}_{\mathrm{cell}}} \tag{283}$$

d^* is the non-dimensional wall distance:

$$d^* = W_d\left(\frac{V_{\mathrm{scale}}}{\nu}\right) \tag{284}$$

where W_d is wall distance, ν is kinematic viscosity of fluid, and V_{scale} is the mass average of velocity magnitude.

$$\mathbf{f}\left(d^*\right) \triangleq \min\left(\max\left(\left(\frac{d^* - d^*_{\text{cutoff}}}{0.3d^*_{\text{cutoff}}}\right), 0\right), 1\right) \tag{285}$$

where d^*_{cutoff} is user-specified non-dimensional wall distance.

Both the model coefficient ε_0 and non-dimensional cutoff wall distance can be retuned for different simulation. Default values have been found to be appropriate for external aerodynamics.

Region Values

This section provides information about the region values that are available for flow and energy models in STAR-CCM+.

This section also provides a detailed reference of the methods of setting the values. Region, boundary, and interface values are elected by the region, boundary, or interface types, and/or by methods that are selected in corresponding nodes in the boundary conditions node.

Mass Source	This node is added when the Mass Source Option region condition is activated. It is entered as a scalar profile.
Mass Source Pressure Derivative	This node represents linearization of the Mass Source with respect to pressure, S_u in Eqn. (1650) (coupled) and $S_u dV$ in Eqn. (1461) (segregated).
Mass Source Temperature Derivative	See S_u in Eqn. (1650). The derivative of the user source term is used to linearize Eqn. (1650). Its value is set to zero by default. Providing a value for the derivative helps stabilize the solution.
Mass Source Velocity Derivative	This node represents linearization of the Mass Source with respect to x-, y-, [z-] components of velocity. See S_u in Eqn. (1650). The derivative of the user source term is used to linearize Eqn. (1650). Its value is set to zero by default. Providing a value for the derivative helps stabilize the solution.
Momentum Source	This node is added when the Momentum Source Option region condition is set to Specified. It is input as a vector profile.
Momentum Source Pressure Derivative	This node represents linearization of the Momentum Source with respect to pressure, S_u in Eqn. (1650).
Momentum Source Temperature Derivative	This node represents linearization of the Momentum Source with respect to temperature, S_u in Eqn. (1650).
Momentum Source Velocity Derivative	This node represents linearization of x-, y-, [z-] components of the Momentum Source with respect to x-, y-, [z-] components of velocity, respectively, S_u in Eqn. (1650) (coupled) and $S_u dV$ in Eqn. (1650) (segregated).

Flow Node Reference

The object-oriented nature of STAR-CCM+ means that several objects are shared among nodes in the simulation tree.

Contents:

Conditions
Reference Values

Conditions

The boundary conditions that are available to all flow models in STAR-CCM+ are:

- Flow direction specification
- Reference frame specification
- Backflow direction specification
- Mass flow option
- Pressure outlet: pressure jump option
- Pressure outlet: target mass flow option
- Shear stress specification
- Tangential velocity specification
- Turbulence suppression option
- Velocity specification

Under the pressure outlet boundary condition:

- The non-reflecting option is available to the Coupled Flow, and Harmonic Balance Flow and Energy models.
- The average pressure is available to the Segregated Flow model.

The region conditions that are available to all flow models in STAR-CCM+ are:

- Mass source option
- Momentum source option

This section provides a detailed property reference for these conditions. Region, boundary, or interface conditions are elected by the region, boundary, or interface types, which in turn are limited by the models that have been turned on in the continuum.

Contents:

Mass Source Option
Momentum Source Option
Non-Reflecting Option

Mass Source Option

A mass source option can be allowed or not.

Properties

Mass source option	Defines whether you want to enter a mass source term.	
	Activated	Allows a mass source term to be defined. The following nodes are added to the **Physics Values** node: • Mass Source • Mass Source Pressure Derivative • Mass Source Temperature Derivative • Mass Source Velocity Derivative
	Deactivated	Does not allow a mass source term to be defined.

Momentum Source Option

This condition allows you to select one of the following options:

- No momentum source defined
- A momentum source term defined
- Fan properties and device data defined

Properties

Momentum source option	Defines whether you want to enter a momentum source term, and which type.	
	None	Does not allow a momentum source term to be defined.
	Specified	Allows a momentum source term to be defined. The following nodes are added to the **Physics Values** node: • Momentum Source • Momentum Source Pressure Derivative • Momentum Source Temperature Derivative

		• Momentum Source Velocity Derivative
	Fan	Allows fan properties and device data to be defined. A Fan Momentum Source node is added to the **Physics Values** node.

Non-Reflecting Option

This boundary condition is available on stagnation inlet and pressure outlet boundaries through Pressure Outlet Option and Stagnation Inlet Option respectively.

This option is available only when the Coupled Flow Ideal Gas, Steady, and Two-Dimensional or Three-Dimensional models are enabled, and only on boundaries of linearly (two-dimensional) or rotationally (three-dimensional) periodic problems. It is appropriate in instances where the flow normal to the non-reflecting boundary remains subsonic.

In three-dimensional simulations, the boundary must be a surface of revolution. Furthermore, the boundary mesh must be a structured, mapped mesh of quadrilateral faces. (This requirement does not apply to steady-state calculations using the Coupled Flow model.) If these conditions are not satisfied, an error is generated during pre-processing of the boundary mesh before the application of the non-reflecting boundary condition procedure.

Properties

Non-Reflecting Option	Specifies whether to use the non-reflecting option.	
	Activated	Use the non-reflecting option.
	Deactivated	Do not use the non-reflecting option.

Note: No additional nodes appear in the **Values** node when this option is activated. However, there are some restrictions as to which methods are appropriate for specifying other boundary values. The table below indicates which methods can be used with which boundary values when this option is activated.

	Method			
Boundary Value	Constant	Radial Equilib.	Tabular Data	Field Function
Total pressure	Yes	No	3D[1]	No
Pressure	Yes	Yes[2]	3D[1]	No
Total temperature	Yes	No	3D[1]	No
Flow direction	Yes	No	3D[1]	No

Note: [1] External radial tabular data files only, not internal or xyz formatted.

[2] At a pressure outflow boundary only.

All existing Methods for turbulence quantities and other transported scalars are acceptable.

Reference Values

This section lists all the reference values that must be set for flow models in STAR-CCM+ and provides a detailed reference of the methods of setting them.

- Gravity
- Maximum allowable temperature
- Minimum allowable temperature
- Minimum allowable wall distance
- Maximum number of eddies
- Reference altitude
- Reference density
- Reference pressure

Maximum Allowable Temperature

The largest temperature value that is permitted anywhere in the continuum.

The Energy models (Coupled Energy, Coupled Solid Energy, Segregated Solid Energy, Segregated Fluid Enthalpy, Segregated Fluid Temperature) limit temperature corrections such that the corrected value does not exceed this maximum. If this occurs, a message is printed to the *Output* window.

Properties

Value

The value and units of the maximum allowable temperature. Clicking the customizer executes the define value action. (Default value is 5000 K.)

Minimum Allowable Temperature

The smallest temperature value that is permitted anywhere in the continuum.

The Energy models (Coupled Energy, Coupled Solid Energy, Segregated Solid Energy, Segregated Fluid Enthalpy, Segregated Fluid Temperature) limit temperature corrections such that the corrected value does not go below this minimum. If this occurs, a message is printed to the *Output* window.

Properties

Value

The value and units of the minimum allowable temperature. Clicking the customizer executes the define value action. (Default value is 100 K.)

Maximum Number of Eddies

The maximum number of eddies parameter sets an upper limit on the number of initial eddies that are generated within the flow domain by the Synthetic Eddy Method.

These initial eddies prepare the flow for an analysis that uses the Large Eddy Simulation approach for turbulence. If the maximum limit is reached, a warning is shown in the *Output* window, as this could potentially destroy the desired effect of the initial turbulence. It should not be set too low.

Fluid Flow and Energy Field Functions Reference

This section contains the following sub-sections:

- Common Flow and Energy Field Functions
- Porosity_Field_Functions including:
  - Porosity_Parameter_Field_Functions
  - Viscous_Term_Field_Function
  - Inertial_Term_Field_Function
  - Region Porosity

Contents:

Common Flow and Energy Field Functions
Porosity Field Functions

Common Flow and Energy Field Functions

These field functions are available to both the flow and energy models. These field functions are also available to the Eulerian Multiphase models, except where indicated.

These objects all have the same base properties, with some field functions also having reference value properties (indicated by a ref subscript in the following descriptions).

Absolute Pressure	The sum of the static pressure and the reference pressure. It is used to obtain density by the Ideal Gas Law, Eqn. (1466).
Absolute Total Pressure	The sum of the total pressure P_t and the reference pressure p_{ref} : $P_{t,\mathrm{abs}} = (P_t + p_{\mathrm{ref}})$
Axial Velocity	The axial component of velocity v_{ax} relative to the axis and origin in the coordinate system given under the **Axis** node of the region. This field function is not available in Eulerian Multiphase models for 3-D cases.
Cell Relative Velocity	The fluid flow velocity at the cell centroid relative to the cell centroid due to motion.
CFL Number	The CFL number controlled by the Expert Driver.
Convective Courant Number	The local convective Courant number, which is the ratio of the physical time-step to the mesh convection

	time scale, $\Delta t/(dx/V)$ or $V^*\Delta t/dx$. For moving mesh cases, V is the velocity relative to the mesh. When you use this field function in a Eulerian Multiphase model, it plots the value for the phase that has the maximum convective Courant number in a cell.
Density	The density of the fluid.
Energy Residual	This residual becomes available in simulations involving heat transfer for fluids alone when the Energy solver has the **Temporary Storage Retained** property activated.
Flow Direction	Defined on any boundary that has a Flow Direction physics condition. For each Flow Direction option, it gives a vector which is parallel to the flow direction specification. See the Flow Direction boundary value.
Flow Work Energy Source	The rate of viscous dissipation or viscous work done as fluid is moved through a boundary, the dot product of the viscous stress tensor and the velocity with the boundary area $(\mathbf{T} \cdot \mathbf{v}) \cdot \mathbf{a}$. See Eqn. (1463) and Eqn. (2233). This field function is only available with Segregated Flow, with *Temporary Storage Retained* activated in the Segregated Flow solver.
Grid Relative Velocity	The velocity vector field relative to the grid. This field function is not available in Eulerian Multiphase models.
Helicity	The scalar defined by $H = V \cdot \Omega$ where V is velocity and Ω is vorticity.
Lamb Vector	The cross product of velocity and vorticity, which is defined as $\mathbf{L} \equiv \mathbf{v} \times \nabla \times \mathbf{v}$.
Lambda 2 Criterion	The scalar defined by the second eigenvalue of $\left(\mathbf{S}^2 + \Omega^2\right)$ when sorted from minimum to maximum, where $\mathbf{S}$ is the strain-rate tensor and Ω is the spin tensor. Values of $\lambda_2 < 0$ can be interpreted as vortex regions, while values equal to or greater than 0 have no physical interpretation.
Local Time Step	The time-step used in the cell under consideration.
Mass Flow Rate	The rate at which mass flows through a boundary.
Mass Flux	The face mass flux, $\dot{m}_f$.

Mass Imbalance	The imbalance of mass in any cell that results from evaluating the discrete continuity equation at the given iteration or time-step. For incompressible flows with no source terms this equation reduces to $\sum_f \dot{m}_f$ for each cell.
Pressure	The working pressure that is the solution variable in the flow models in STAR-CCM+. It is the quantity p in Eqn. (1650).
Pressure Coefficient	The value that is defined as $C_p = (p - p_{\mathrm{ref}}) / \left(\frac{1}{2}\rho_{\mathrm{ref}}\ \mathrm{v}_{\mathrm{ref}}^2\right)$ where p_{ref} , ρ_{ref} and $\mathrm{v}_{\mathrm{ref}}$ are the reference pressure, density, and velocity, respectively. This field function is not available in Eulerian Multiphase models.
PressureGrad	The pressure gradient $\Delta\ P$ times the cell volume. This becomes available when you activate the Coupled Solver and the *Temporary Storage* property.
Q Criterion	The scalar value defined by: $Q := \frac{1}{2}\left(\lVert\Omega\rVert^2 - \lVert\mathbf{S}\rVert^2\right)$ where Ω is the spin tensor and $\mathbf{S}$ is the strain-rate tensor. Where positive, this is vorticity-dominated flow; a negative value implies strain-dominated.
Radial Velocity	The radial component of velocity v_r relative to the axis and origin in the coordinate system given under the **Axis** node of the region. This field function is not available in Eulerian Multiphase models for axisymmetric cases.
Relative Total Pressure	The pressure that is obtained from isentropically bringing the flow to rest in the relative frame of motion. This field function is not available in Eulerian Multiphase models.
Skin Friction Coefficient	Defined as $C_f = \lvert\vec{\tau}_w\rvert / \left(\frac{1}{2}\rho_{\mathrm{ref}}\ \mathrm{v}_{\mathrm{ref}}^2\right)$ where ρ_{ref} and $\mathrm{v}_{\mathrm{ref}}$ are the reference density and velocity, respectively. This field function is not available in Eulerian Multiphase models.
Specific Heat	The specific heat of the model material.

Static Pressure	The spherical part of the stress tensor acting in fluids, which is the same as the actual thermodynamic pressure of the fluid. When the gravity model is active, it is related to the working pressure, (which becomes the piezometric pressure), by Eqn. (1572) for variable density flows and by Eqn. (1571) for constant density flows.
Tangential Velocity	The tangential component of velocity v_t relative to the axis and origin in the coordinate system given under the **Axis** node of the region. This field function is not available in Eulerian Multiphase models for axisymmetric cases.
Temperature	The temperature field.
Temperature Coefficient	See Temperature Coefficient Field Function Properties.
Total Enthalpy	The total enthalpy H is the sum of static enthalpy plus kinetic energy, $H = h + \frac{v^2}{2}$.
Total Pressure	The total pressure (gauge) is the pressure that is obtained from isentropically bringing the flow to rest. For an ideal gas with constant specific heat, it is defined as $P_t = P_{\mathrm{abs}}\left[\left(1 + \frac{(\gamma-1)}{2}M^2\right)^{\gamma/(\gamma-1)}\right] - P_{\mathrm{ref}}$ where P_{abs} , P_{ref} , M and γ are the absolute pressure, reference pressure, Mach number, and ratio of specific heats, respectively. For alternate equations of state, or a non-constant specific heat, P_t is obtained by integrating $\mathrm{d}\,h = \mathrm{d}\,P/\rho$ from static conditions to total conditions.
Total Pressure Coefficient	The value that is defined as $C_p = \left(P_t - P_{\mathrm{ref}}\right)/\left(\frac{1}{2}\rho_{\mathrm{ref}}\ \mathrm{v}_{\mathrm{ref}}^2\right)$ where P_{ref} , ρ_{ref} and $\mathrm{v}_{\mathrm{ref}}$ are the reference pressure, density, and velocity, respectively. This field function is not available in Eulerian Multiphase models.
Turbulent Charge	The value that is defined as $n = \nabla \cdot \mathbf{L}$, where $\mathbf{L}$ is the lamb vector. This field function is not available in Eulerian Multiphase models.

Velocity	The velocity vector field.
Vorticity	The vector variable with components ζ_x , ζ_y , ζ_z defined as: $\zeta_x = \frac{\partial w}{\partial y} - \frac{\partial v}{\partial z}, \quad \zeta_y = \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x}, \quad \zeta_z = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$ where u , v and w are velocity components in the x, y and z directions. $\nabla\times v$ is the curl of the velocity field.
Vorticity Confinement Non-Dimensional Wall-Distance	d^* in Eqn. (278).
Vorticity Confinement Specific Force	$\mathbf{f}_\omega$ in Eqn. (274).
Wall Shear Stress	The wall shear stress vector, which is defined at a wall face as $\vec{\tau}_w = (-\mathbf{T}\cdot\mathbf{a}/\lvert a\rvert)$, where $\mathbf{T}$ is the stress tensor (see coupled viscous fluxes or segregated viscous fluxes). For a multiphase fluid, the wall shear stress vector is the sum of the wall shear stress vectors for each phase. This summation is valid because the phase-specific coefficients are already volume fraction weighted.
Wall Shear Stress of <phase	The wall shear stress vector for the specified phase. A version of this field function is created for each phase in a multiphase continuum. This field function is volume fraction weighted for each phase.

Porosity Field Functions

This section includes the following:

- Porosity_Parameter_Field_Functions
- Viscous_Term_Field_Function
- Inertial_Term_Field_Function
- Region Porosity

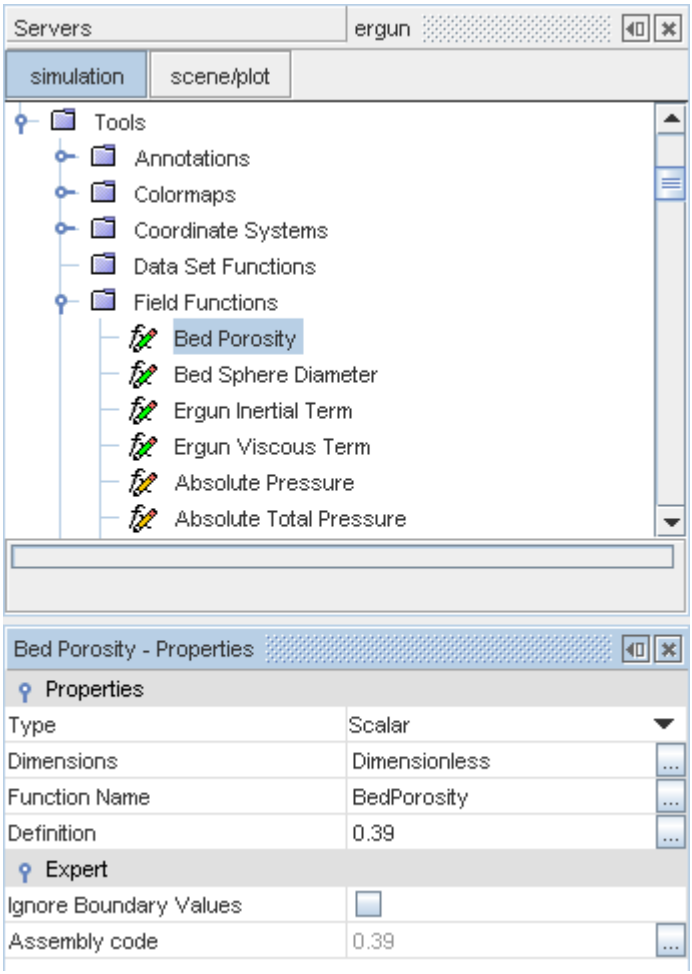
Contents:

Porosity Parameter Field Functions
Viscous Term Field Function
Inertial Term Field Function

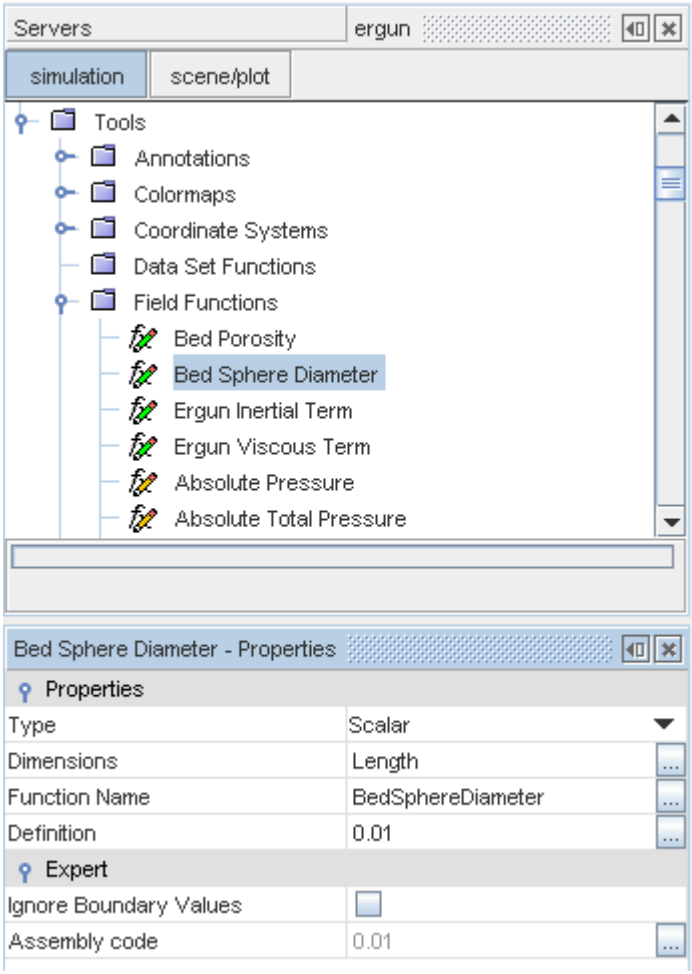
Region Porosity

Porosity Parameter Field Functions

The volume porosity is defined as a scalar field function with no dimensions, and whose *Definition* is set to the required value (0.39 in this example).



The particle diameter (0.01 m in this example) is defined in a similar way, except that the *Dimensions* property is set to Length.

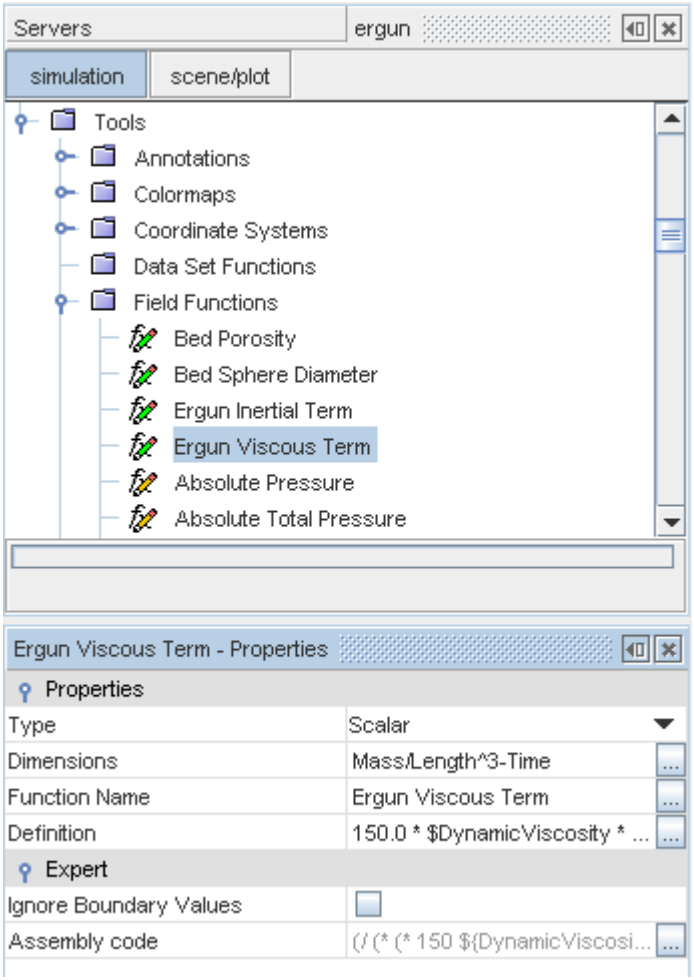


Viscous Term Field Function

The function uses the **Bed Porosity** and **Bed Sphere Diameter** functions that were defined previously.

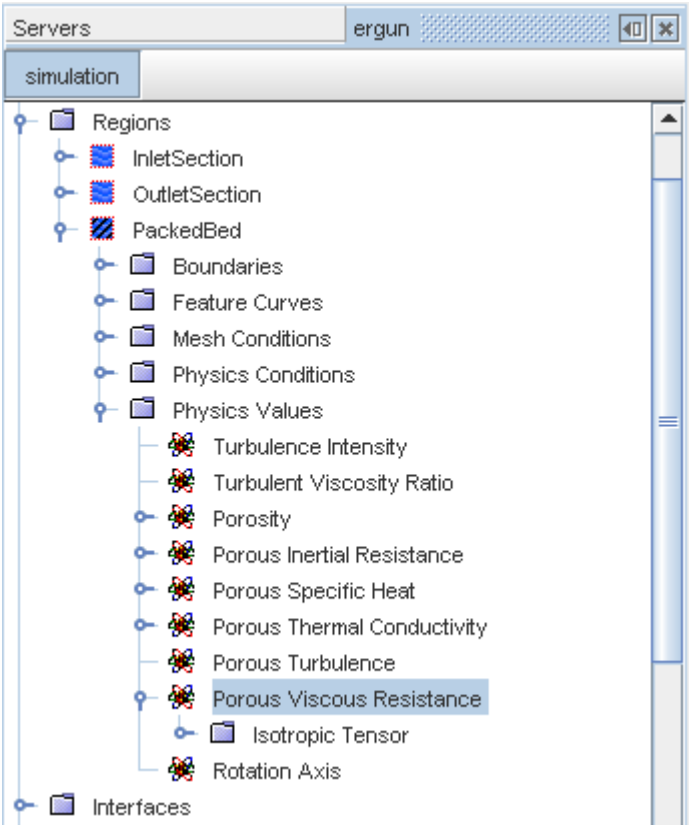
The scalar field function for the viscous term is dimensioned as $Mass \cdot Length^{-3} \cdot Time^{-1}$, and the full *Definition* is:

```
150.0*$DynamicViscosity*pow((1-$BedPorosity),2)/
(pow($BedPorosity,3)*pow($BedSphereDiameter,2))
```

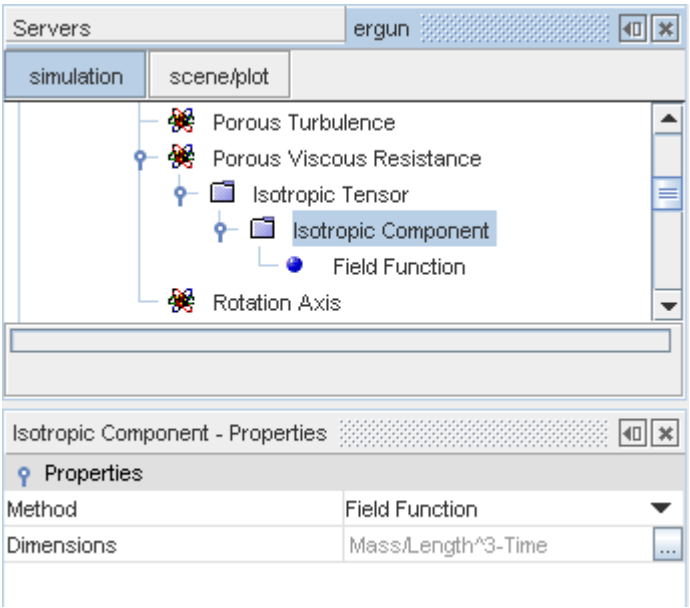


This field function is applied to the **Porous Viscous Resistance** node as an isotropic tensor.

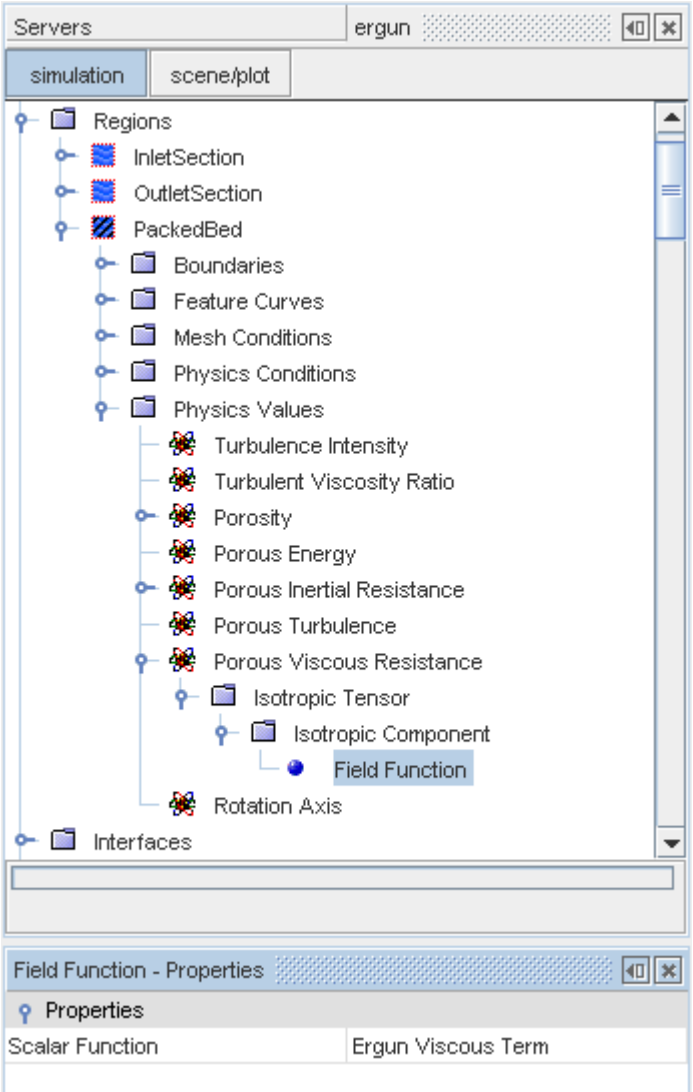
- Select the **Physics Values** > **Porous Viscous Resistance** node on the porous region.



- In the *Properties* window, set the *Method* to **Isotropic Tensor** using the drop-down menu. The **Isotropic Tensor** child node is shown underneath the **Porous Viscous Resistance** node.
- Select the **Isotropic Tensor** > **Isotropic Component** node and, in the *Properties* window, set the *Method* to **Field Function**.



- Finally, select the **Isotropic Component** > **Field Function** node and choose **Ergun Viscous Term** from the *Scalar Function* drop-down menu in the *Properties* window.

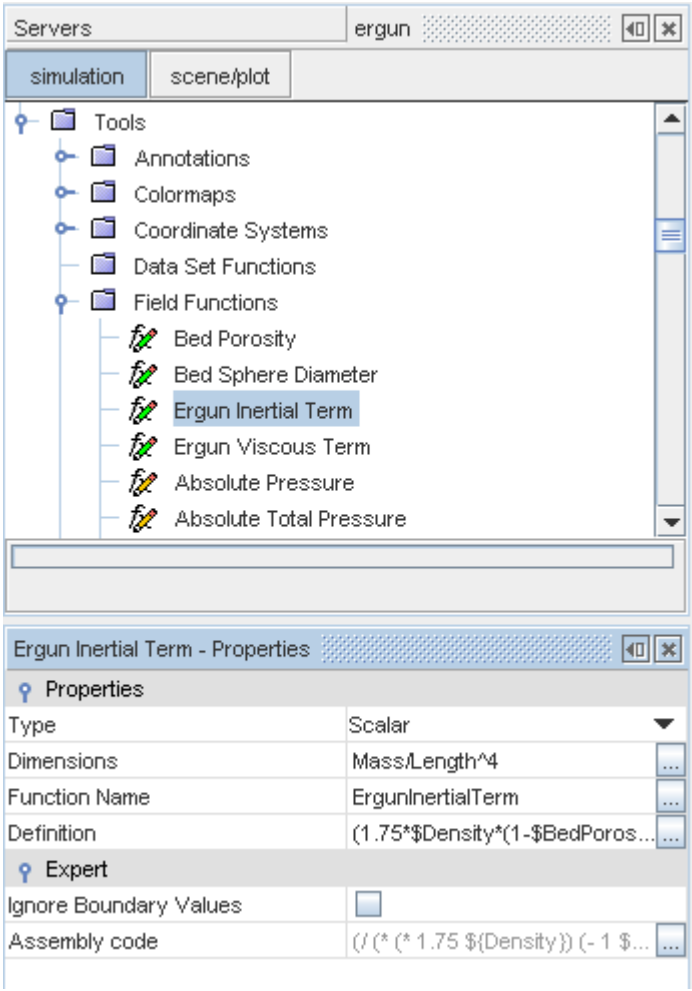


Inertial Term Field Function

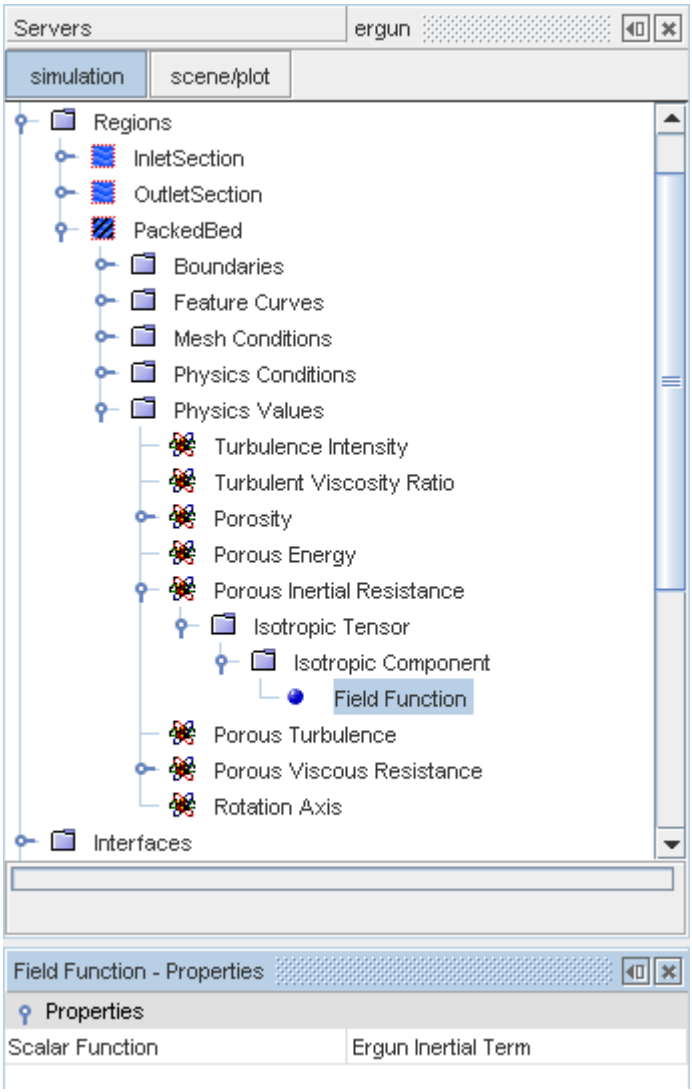
The fluid density is included in this field function. Any variations in the fluid density therefore affect the inertial resistance.

The scalar field function for the inertial term is dimensioned as $Mass \cdot Length^{-4}$, and the *Definition* is:

```
(1.75*$Density*(1-$BedPorosity))/
(pow($BedPorosity,3)*$BedSphereDiameter)
```



This function is applied to the **Porous Inertial Resistance** using the isotropic tensor method as before:

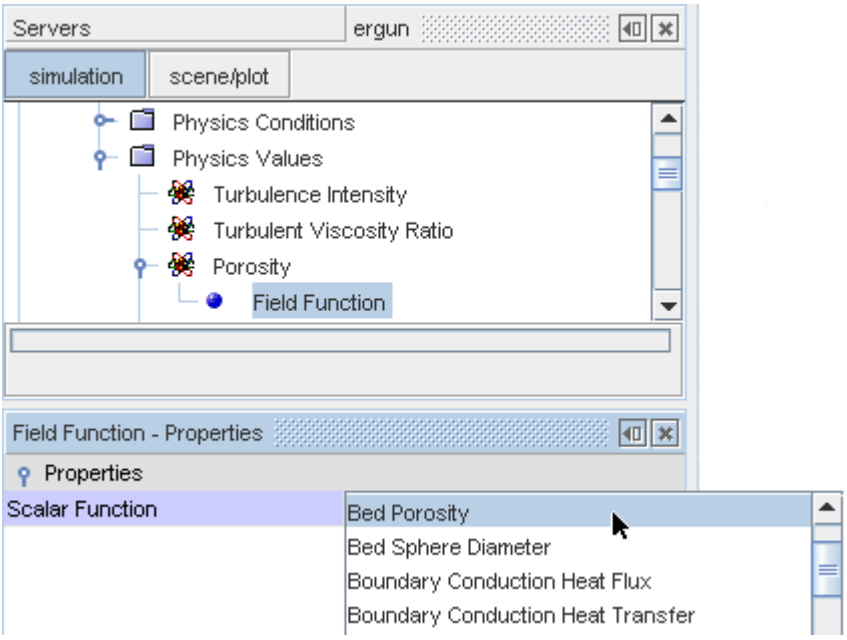


Region Porosity

For an unsteady calculation, the volume porosity χ can be set on the region using the **Bed Porosity** field function.

To set the volume porosity:

- Select the **Physics Values** > **Porosity** node of the porous region (**PackedBed** in this example).
- In the *Properties* window, set the *Method* property to **Field Function** using the drop-down menu.
- Select the **Porosity** > **Field Function** node and, in the *Properties* window, set the *Scalar Function* property to **Bed Porosity**.


Servers
ergun
simulation
scene/plot
Physics Conditions
Physics Values
Turbulence Intensity
Turbulent Viscosity Ratio
Porosity
Field Function
Field Function - Properties
Properties
Scalar Function
Bed Porosity
Bed Sphere Diameter
Boundary Conduction Heat Flux
Boundary Conduction Heat Transfer

Fluid Flow Formulation

This section contains the following sub-sections on formulations used in modeling fluid flow.

Contents:

Passive Scalar Formulation
Isentropic Flow Formulation

Passive Scalar Formulation

This formulation includes:

- The transport equation for passive scalar
- Passive scalar diffusion flux formulation
- Wall boundary conditions

Transport Equation

The transport equation for the passive scalar component, ϕ_j, is:

$$\frac{\partial}{\partial t}\int_{\widetilde{V}} \rho\phi_j d\widetilde{V} + \oint_A \rho\phi_j\left(\mathbf{v}-\mathbf{v}_g\right)\cdot d\widetilde{\mathbf{a}} = \oint_A \mathbf{J}_j \cdot d\widetilde{\mathbf{a}} + \int_{\widetilde{V}} S_{\phi_j} d\widetilde{V} \tag{286}$$

where:

- j is the component index.
- $d\widetilde{V} = \alpha_i \chi dV$, where α_i is the volume fraction of phase *i* (set to 1 for single phase flows) and χ is the void fraction.
- $\mathbf{J}_j$ is the diffusion flux. See the following section.
- $d\widetilde{\mathbf{a}} = \alpha_i \chi d\mathbf{a}$
- S_{ϕ_j} is a source term for passive scalar component j. The source term is not multiplied internally by the volume fraction. ϕ_j is assumed to be positive-definite.

Passive Scalar Diffusion Flux

The diffusion flux $\mathbf{J}_j$can be expressed with either the linear eddy-diffusivity model (the default) or the Generalized Gradient Diffusion Hypothesis (GGDH) model.

In the linear eddy-diffusivity model, the diffusion flux is:

$$\mathbf{J}_j = \left(\frac{\mu}{\sigma} + \frac{\mu_t}{\sigma_t}\right)\nabla\phi_j$$

where:

- μ is the viscosity.

- μ_t is the turbulent viscosity.
- σ is molecular Schmidt number—a material property.
- σ_t is the turbulent Schmidt number.

The Generalized Gradient Diffusion Hypothesis model (GGDH) provides improved prediction for passive scalar diffusion in highly anisotropic turbulent flows. This hypothesis models the passive scalar diffusion flux as a tensor quantity proportional to the Reynolds stresses:

$$\mathbf{J}_j = \frac{\mu}{\sigma}\nabla\phi_j + \frac{C_s}{\sigma_t}\tau\mathbf{R}\cdot\nabla\phi_j \tag{287}$$

- ϕ_j is the value of the passive scalar component j.
- $C_s = 0.2$, a model constant.
- $\tau = k/\epsilon$, where k is kinetic energy and ϵ is the dissipation rate.
- $\mathbf{R}$ is the Reynolds stress tensor.

Wall Boundary Conditions

At walls, use either a specified flux or a specified value condition. See Setting Passive Scalars at Boundaries.

Isentropic Flow Formulation

The following isentropic relations are useful when initializing compressible flow cases:

$$\frac{T_t}{T} = 1 + \frac{(\gamma-1)}{2}M^2 \tag{288}$$

$$\frac{P_t}{P} = \left(1 + \frac{(\gamma-1)}{2}M^2\right)^{\gamma/(\gamma-1)} \tag{289}$$

$$V = M\sqrt{\gamma RT} \tag{290}$$

$$\frac{\dot{m}}{A} = \sqrt{\frac{\gamma}{R}}\frac{P_t}{\sqrt{T_t}}M\left(1 + \frac{(\gamma-1)}{2}M^2\right)^{1/2-\gamma/(\gamma-1)} \tag{291}$$

$$\gamma = \frac{C_p}{C_p - R} \tag{292}$$

$$R = \frac{R_u}{M_r(X)} \tag{293}$$

where:

- A is area
- $\dot{m}$ is mass flow rate
- M is Mach number
- $M_r(X)$ is molecular weight (relative molecular mass) of molecular entity X
- P is static pressure

- P_t is total pressure
- R is specific gas constant
- R_u is universal gas constant [8314.4621 J/kmol K]
- T is static temperature
- T_t is total temperature
- V is velocity magnitude
- γ is ratio of specific heats

Modeling Viscous Flow

Viscous Flow is a finite element approach for use with viscoelastic materials and other highly viscous non-Newtonian fluids, such as liquid plastics and rubber, dough and similar foodstuffs, molten glass, and mud. Viscoelastic materials resemble elastic materials, but also exhibit viscous effects, rebounding slowly from deformations.

Use the Viscous Flow models to calculate:

- Heating and normal stresses inside the viscous flows
- Dissipation energies
- Non-linear behavior of viscoelastic fluids
- Residence time distributions
- Species distributions

Different viscoelastic models can be coupled with each other, each model using the material properties under all models, to improve convergence. See Viscoelastic Model.

Viscous flow simulations are designed to run in double precision. You can deactivate the **Solve in double precision property** of the Viscous Flow solver, but do so only when memory is severely restricted. See Viscous Flow Solver.

Several rheological models/constitutive equations are available to model non-Newtonian fluids:

Use Cases	Model	Rheology Law
High-viscosity fluids in which stress, fluid velocity, and viscosity are interrelated.	Generalized Non-Newtonian	Generalized Newtonian Fluid
High-viscosity fluids with elastic properties	Viscoelastic	Viscoelastic Fluids
∟ Uncured molten rubber and similar materials	∟ Viscoelastic mode	Giesekus Leonov
∟ Academic cases with multiple normal stresses and constant shear viscosity	∟ Viscoelastic mode	Oldroyd-B
∟ Molten polyethylenes and similar materials	∟ Viscoelastic mode	Linear Phan Thien-Tanner / JS(Johnson-Segalman)
		Exponential Phan Thien-Tanner / JS(Johnson-Segalman)

Use Cases	Model	Rheology Law
∟ Complex geometries with elongation and shear flow	∟ Viscoelastic mode	eXtended Pom-Pom

Scope

- The Viscous Flow models are best suited to low Reynolds numbers, less than 10 and often less than 1.
- Viscous Flow is not suitable for simulating convection, though it is compatible with diffusion.
- The fluid flows treated by Viscous Flow are laminar; turbulence models are not compatible with Viscous Flow.
- Use Viscous Flow models only with single-phase flows.
- Viscous Flow models are not compatible with overset meshes or conjugate heat transfer.
- Viscous Flow models do not work with radiation modeling.

Use Case: Rubber Seals

A manufacturing process runs liquid rubber through a mixer and extrudes it in a strip. The strip is then cooled in water and sliced into thin sheets to be used as seals. Viscous Flow models can simulate the behavior of the rubber in the mixer, at the point of extrusion, and as it is cooled. This way, the manufacturer can evaluate the heat and forces in the mixer, the normal stresses affecting extrusion, and the rate of cooling in the water. All these factors affect the maximum speed for the process. See the tutorials Generalized Non-Newtonian Fluid Flow in a Static Mixer and Viscoelastic Flow: Basic Extrusion.

Contents:

Modeling Rheology
Setting Up Viscous and Viscoelastic Fluid Flow
Extrusion and Free Surface Flow
Setting Up Extrusion and Free Surface Flow
Setting Up a Film Casting Simulation
Viscous Flow Models
Mesh Requirements and Guidelines
Guidelines for Co-Extrusion
Guidelines for the eXtended Pom-Pom Model (XPP)
Viscous Flow Field Function Reference

Modeling Rheology

Rheology distinguishes materials with "memory" from those without.

Materials with memory retain deformation even after the deforming force stops, and only gradually resume their initial shape. Materials having no memory can be described by an instantaneous relationship between the shear rate in a general flow and the stress. The shear rate is evaluated through the second invariant of the rate of deformation tensor, or strain rate tensor, which describes the rate of change of deformation. These are the Generalized Newtonian materials. STAR-CCM+ models these materials under the **Material Properties** node for the fluid, using the options **Dynamic Viscosity** > **Carreau-Yasuda**, **Cross Fluid**, or **Power Law**.

Materials such as polymers and rubber often exhibit memory effects. In those cases, the shear stress tensor is the sum of a solvent component and a viscoelastic component. STAR-CCM+ models the solvent component as a Generalized Newtonian material, using the **Dynamic Viscosity** property mentioned above. It models the viscoelastic component as the sum of 1 to 8 modes. Each mode provides a simplified or partial model of the viscoelastic behavior and obeys one of the following differential constitutive equations:

- Oldroyd-B
- Linear Phan-Thien Tanner (and Johnson-Segalman)
- Exponential Phan-Thien Tanner (and Johnson-Segalman)
- Giesekus-Leonov
- eXtended Pom-Pom (XPP)

Each viscoelastic mode is coupled to the others, to improve convergence, but this improvement comes at a considerable computational and memory cost that increases rapidly with the number of modes. This is especially true in 3D simulations.

The only variables required by Generalized Newtonian flows are velocity and pressure. Additional variables, such as temperature, are optional.

Viscoelastic flows require additional variables to model the viscoelastic component:

- The components of the Rate of Deformation tensor (the same for all modes), $\mathbf{D}$ in Eqn. (1499)
- Components of the Extra-Stress tensor for each viscoelastic mode, $\mathbf{T}^p$ in Eqn. (1498), determined by the "well-established DEVSS" numerical technique [233]

Two important effects can be modelled by viscoelastic fluids that cannot be modeled with a Generalized Newtonian approach alone:

- Normal stresses, which drive effects such as extrudate swelling
- Elongational flow resistance (sometimes called “extensional viscosity”), which has a major impact in the vortex growth mechanism

Setting Up Viscous and Viscoelastic Fluid Flow

The workflows for setting up generalized Newtonian fluids and for setting up viscoelastic fluids are similar. To model a non-Newtonian fluid with a high viscosity or a variable viscosity depending on shear-rate without any elastic effects, choose the **Generalized Non-Newtonian Model**. If the non-Newtonian fluid exhibits normal stress or memory effects, work with the **Viscoelastic Model**. High shear rates in the flow can lead to a temperature increase—viscous heating. To account for this effect, STAR-CCM+ provides the **Viscous Energy Model**.

The Finite Element discretization imposes certain requirements on the meshes that are suitable for using with the viscous flow model. Generate either tetrahedral meshes, extruded meshes, or directed meshes in combination with prism layers. You can also use imported hexahedral meshes. The viscous flow model is not compatible with polyhedral or trimmed meshes.

If you are replacing an existing mesh with a new mesh, STAR-CCM+ automatically interpolates any vertex data, available on the existing mesh, to the new mesh vertices. For more information on the **Replace Mesh** operation, see Replacing a Mesh Region.

To set up generalized non-Newtonian or viscoelastic fluid flow:

1. Create a volume mesh that is suitable for the Finite Element discretization. See Mesh Requirements and Guidelines.

   If you are modeling co-extrusion, and hence have one fluid region in contact with another, use a conformal mesh at the interface between the fluid regions.
2. Create a physics continuum for each non-Newtonian fluid material in the simulation.
3. For the physics continuum of each non-Newtonian fluid, under **Continua** > **[Physics Continuum]** > **Models**, select the following models in order:

Group Box	Model
Time	**Steady** or **Unsteady** To model cases with Weissenberg number > 1 and using the eXtended Pom-Pom model, select **Unsteady**.
Material	**Liquid**
Flow	Viscous Flow **Laminar** (selected automatically) **Constant Density** (selected automatically)
Rheology	Select one of the following: • To model a shear-rate dependent viscosity (generalized Newtonian) fluid, select Generalized Non-Newtonian. • To model a viscoelastic fluid, select Viscoelastic.
Optional Models	Select one of the following:

Group Box	Model
	• To model temperature effects such as viscous heating, select Viscous Energy. • To model extrusion or any free surface flow of a non-Newtonian fluid, select Free Surface. • To model partial filling of a domain, such as for injection molding, select Partial Fill. Partial filling is only available for unsteady simulations. • To model the slippage of the fluid at boundaries, use the Partial Slip models. • To model passive scalar components, use the Passive Scalar model.

4. Define the material properties. These liquids are incompressible. Depending on the type of fluid you model, do one of the following:

For the Generalized Non-Newtonian model:

Edit the **Models** > **Liquid** > **[material]** > **Material Properties** node and set the Dynamic Viscosity Properties.

For the Viscoelastic model:

1. Select the **Models** > **Viscoelastic** node and set the *Number of Modes*. You can specify 1–8 modes. The behavior of the viscoelastic model is the sum of the behaviors of the modes. You can select a different viscoelastic material model for each mode. For more information, see Viscoelastic Properties.
2. Edit the **Models** > **Liquid** > **[material]** > **Material Properties** node and set
   - Dynamic Viscosity Properties
   - Viscoelastic Mode Properties for each mode. See Viscoelastic Fluid for descriptions of the rheological laws. If you select **Extended PomPom**, see the Guidelines for the eXtended Pom-Pom Model (XPP) for that method.

5. To simulate temperature-dependence of the dynamic viscosity, select the Viscous Energy model, then select **Liquid** > **[material]** > **Material Properties** > **Temperature Shift Factor** and set the Temperature Shift Factor method.
6. To simulate free surface flow, perform the additional steps that are described in Setting Up Extrusion and Free Surface Flow.
7. Set up boundary conditions. If there is more than one fluid, separate them with an immiscible wall within the die.

   Because viscous flow simulations use finite-element discretization, mesh vertices sometimes lie on two boundaries at once. If the boundaries specify different boundary condition values, STAR-CCM+ uses the average of the two. This averaging is the default procedure. If you want to promote one boundary condition over another, apply a slip condition to the boundary with a less dominant contribution and set its slip coefficient to a value on the order of `10^6` to `10^8`, depending on the polymer viscosity.
8. Set solver parameters.

   For time-dependent cases where the viscous fluid flows in to fill a region, you are advised to have a Courant number between 0.1 and 1.

For strongly non-linear cases with a Weissenberg number > 1, raise the *Relaxation over # iterations* property of the Viscous Flow Solver above the default value of `10`.

9. Run the simulation and analyze results.

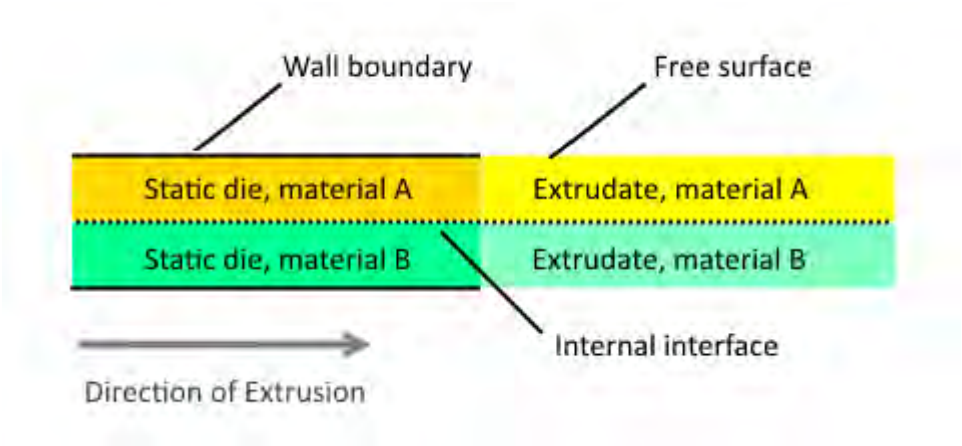
   When using field functions in a scene or report, make sure that you use *Smooth Values* , so that you visualize the nodal values.

Extrusion and Free Surface Flow

You can use the **Free Surface** model to simulate extrusion processes of polymer melts and other processes involving free surface flow of generalized Newtonian and viscoelastic fluids. A free stream boundary represents the outer surface of the extrudate, and an internal interface identifies the boundary between extrudates in contact. Since its shape is not known at the start of the simulation, the free surface is calculated as part of the solution. Typically, you set up these simulations as steady-state. You can also use time-dependent simulations, in which case the free surface position propagates at the material velocity.

STAR-CCM+ employs an Arbitrary Lagrangian-Eulerian (ALE) approach [220] to calculate the free surface of a polymer. The ALE method models the free surface through a mesh boundary that deforms depending on the shape and position of the free surface. This mesh motion is performed through the mesh morpher. You must provide appropriate boundary conditions and constraints for the mesh motion. The mesh motion is performed in steady-state that reaches an equilibrium shape and position when the solution converges.

You prepare a multi-part mesh using at least two parts for each extrudate. One part represents the static die or fixed geometry and the other part represents the extrudate or free surface. If you use the Line Kinematic condition, which is the default, the volume mesh of the free surface must be a directed mesh that is directed in the direction of the extrusion. You generate the mesh for the free surface in its initial undeformed state. For the Line Kinematic condition, the initial free surface mesh must be aligned with the direction of extrusion.

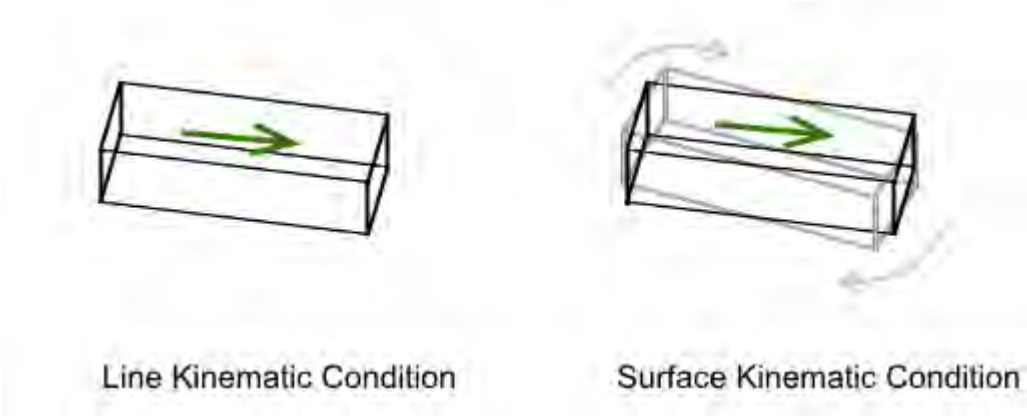


Kinematic Conditions

The **Kinematic Condition** setting puts constraints on cells in regions with Free Surface boundaries, limiting the ways that the cells can move and change shape on the faces of a Free Surface boundary. The **Line** Kinematic Condition applies only in cases of directed meshing, whereas the **Surface** Kinematic Condition uses all mesh types but is less appropriate for corners in extrudate regions.

The **Line** kinematic condition requires the direction of the directed meshing to be parallel to the stream velocity. Thus, the cell surfaces that are part of the free surface are always tangent to the stream velocity.

The **Surface** kinematic condition is less restrictive. It requires the free stream cell faces to be tangent to the stream velocity, but it does not require cell edges to be parallel to the velocity. Thus, a cell can rotate with respect to the velocity as long as the free stream surface remains tangent to the velocity.


Line Kinematic Condition
Surface Kinematic Condition

Setting Up Extrusion and Free Surface Flow

Use the following steps for each extrudate continuum in the simulation.

1. Create a multi-part volume mesh with an internal interface between the static and the free surface regions.

For the free surface, you must specify the extrusion direction and certain constraints subject to which the free surface deforms.

2. Edit the **Models** > **Free Surface** node and set the Free Surface Properties.

The outward boundary of the free surface must be a free stream boundary. To model symmetric extrudates, you can additionally apply a symmetry boundary condition on the corresponding boundary surface.

3. Edit the **Regions** > **[free surface region]** > **Boundaries** > **[outer free surface boundary]** and set *Type* to **Free Stream**.

The motion of the free surface is performed through the morpher. To control the motion, you specify constraints at the free stream boundaries. See Morpher Boundary Conditions Reference. You must apply different conditions to boundaries that are parallel to the extrusion direction from boundaries that are normal to the extrusion direction. Hence, before applying boundary conditions, split the free surface boundary into at least one boundary surface that is initially parallel to the extrusion direction and one outlet boundary surface that is normal to the direction.

4. Edit the **Regions** > **[free surface region]** > **Boundaries** > **[outer free surface boundary parallel to the extrusion direction]** > **Physics Condition** node and set the following:

Node	Property	Setting
Morpher Specification	*Specification*	**Displacement**
Morpher Displacement Specification	*Specification*	**Internal**

These settings cause a free surface to be calculated on this boundary.

5. Edit the **Regions** > **[free surface region]** > **Boundaries** > **[outlet boundary normal to the extrusion direction]** > **Physics Condition** node and set the following:

Node	Property	Setting
Morpher Specification	*Specification*	**Constraint**
Morpher Constraint Specification	*Specification*	**Boundary Plane**
Traction	*Method*	**Aligned Exit**

These settings put no kinematic condition at the boundary. The constraint condition leaves the boundary free to move along with activating free surface tracking.

6. To model co-extrusion, then:

- Edit the **Regions** > **[extrudate region]** > **Boundaries** > **[area of contact between extrudates]** and set *Type* to **Internal Interface Boundary**.
- For the same boundary, edit **Physics Condition** and set the following:

Node	Property	Setting
Morpher Specification	*Specification*	**Displacement**
Morpher Displacement Specification	*Specification*	**Internal**

These settings cause a free surface to be calculated on the interface.

7. To visualize the deformation of the free surface, you can set up a scalar scene and activate the *Surface* option on the free surface part.

   When using field functions in a scene or report, make sure that you use *Smooth Values*, so that you visualize the nodal values.

Setting Up a Film Casting Simulation

Film casting is modeled as a free-surface flow in two dimensions, for a thin sheet of viscoelastic fluid extruded into free space. The shape of the film is modeled, but the shape of the die is not included.

Use the following steps for a film-casting simulation.

1. Create a two-dimensional mesh of the film, with edges designated as inlet, outflow and walls or symmetry planes.
2. In the physics continuum, select the following models:

Group Box	Model
Time	**Steady**
Flow	**Viscous Flow**
Optional Models	**Two Dimensional**
Polymer Displacement	**Free Surface**

3. Edit the **Models** > **Free Surface** node and set the Free Surface Properties to specify the direction of flow.
4. Edit the **Regions** > **[viscous film region]** > **Boundaries** > **[inlet boundary]** node and set *Type* to **Velocity Inlet**.
5. Edit the **Regions** > **[viscous film region]** > **Boundaries** > **[free surface boundary]** node and set *Type* to **Free Surface**.
6. Edit the other nodes under **Regions** > **[viscous film region]** > **Boundaries** node and set *Type* for each to **Free Stream** or **Symmetry Plane**. The boundary for the outlet must be **Free Stream**.
7. Edit the **Boundaries** > **[inlet boundary]** > **Physics Values** node and set the values for **Velocity** and **Film Thickness**. See Boundary Settings.
8. Edit the **Boundaries** > **[outlet boundary]** > **Physics Condition** > **Traction** node and the value for **Method** to **Traction Velocity** or **Normal Velocity**.

   For **Traction Velocity**, the *velocity* component normal to flow is set to zero; for **Normal Velocity**, the *force* component normal to flow is set to zero.

   With either method, the **Physics Values** > **Traction Velocity** node appears.
9. Edit the **Physics Values** > **Traction Velocity** node and select the value for **Method**.
10. Edit the **Physics Condition** nodes for the free surface and set the following:

Node	Property	Setting
Morpher Specification	*Specification*	**Displacement**
Morpher Displacement Specification	*Specification*	**Internal**

If the case is non-isothermal, use the following steps:

11. In the physics continuum, select Viscous Energy from the *Optional Models* box .
12. Edit the **[inlet boundary]** > **Physics Values** > **Static Temperature** node and set the **Value** property.

13. For the side boundaries, edit the **[side boundary]** > **Physics Conditions** > **Thermal Specification** node and set **Condition** property. In most cases, since the film thickness is small, use **Adiabatic**.
14. Edit the **Regions** > **[viscous film region]** > **Physics Conditions** > **Flux Option** node and set **Option**, most commonly to **Convection & Radiation**.
15. Edit the **Regions** > **[viscous film region]** > **Physics Values** node and set the values for **Ambient Temperature**, **Heat Flux**, **Heat Transfer Coefficient**, and **Surface Emissivity**.

Viscous Flow Models

This section describes the physics models specific to viscous flow.

Contents:

Viscous Flow Model
Rheology Models
Free Surface Model
Viscous Energy Model
Partial Fill Model
Film Casting Model

Viscous Flow Model

The Viscous Flow model is the prerequisite for all viscous flow simulations.

Theory	Non-Newtonian Fluids	
Provided by	**[physics continuum]** > **Models** > **Flow**	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Viscous Flow**	
Requires	*Space*: **Three Dimensional**, **Two Dimensional**, or **Axisymmetric** *Time*: **Steady** or **Implicit Unsteady** *Material*: **Liquid**	
Properties	See Viscous Flow Properties	
Activates	Physics Models	Model groups: *Rheology* Automatically selected models: *Equation of State*: **Constant Density** *Viscous Regime*: **Laminar**
	Material Properties	**Density** **Dynamic Viscosity** See Material Property Methods for Rheology
	Reference Values	**Reference Pressure** **Velocity**
	Initial Conditions	**Velocity**
	Boundary Inputs	See Viscous Flow Boundary Settings
	Solvers	Viscous Flow Solver
	Monitors	**Continuity** **Momentum** **Pressure**

		Velocity
	Field Functions	**Pressure** **Shear Rate** **Velocity**

Viscous Flow Properties

Petrov Galerkin

(Also called PSPG for Pressure Stabilization Petrov Galerkin.) A non-dimensional coefficient that controls numerical stabilization of incompressibility. The default value is `1.0`.

Inertia

When **On**, this property has STAR-CCM+ simulate the inertial mass of the viscous fluid. If fluid mass is not a consideration for your application, set the property **Off** to lower computational costs. This property is **On** by default.

Viscous Flow Boundary Settings

Free Stream

Pressure

See entry in Flow Boundaries Reference.

Surface Tension

The magnitude of the surface tension on the exposed surfaces of the free stream. The default is zero.

Mass Flow Inlet

Mass Flow Rate

See entry in Flow Boundaries Reference.

Velocity Inlet

Average Velocity

The velocity of the fluid coming through the inlet, averaged over all the cell faces of the inlet.

Pressure Outlet

Pressure

See entry in Flow Boundaries Reference.

Stagnation Inlet

Total Pressure

See entry in Flow Boundaries Reference.

Flow Direction Specification

See entry in Flow Boundaries Reference.

Reference Frame Specification

See entry in Flow Boundaries Reference.

Wall

Reference Frame Specification

See entry in Flow Boundaries Reference.

Shear Stress Specification

See entry in Flow Boundaries Reference.

Tangential Velocty Specification

See entry in Flow Boundaries Reference.

Viscous Flow Solver

The Viscous Flow Solver is exposed when you select the Viscous Flow physics model.

Note: This solver does not give accurate results if the solvent viscosity (μ_s in Eqn. (1499)) is unrealistically low compared to the viscoelastic viscosity.

This solver has the Sparse Direct Solver as a child node.

Properties

Relaxation over # iterations

The number of iterations over which relaxation reaches full effect. The default value is `10`.

If the Deborah number > 1, non-linear effects dominate and the system does not converge. In such cases, raise the value above 10, experimenting to find the lowest value that gives convergence. Do not lower the value below 10 for any viscoelastic constitutive models.

Solve in double precision

When **On**, the solver works in double precision; when **Off**, the solver works in single precision. This flag is independent of the precision mode in which STAR-CCM+ is started; the precision mode only influences how variables are computed. Mixed modes can lead to truncation errors. By default, this property is **On**.

Verbose

When **On**, provides more output for this solver while the simulation is running, which can be useful for debugging. By default, this property is **Off**.

Parametric Elements

When **On**, the solver uses parametric shape functions; when **Off**, the solver uses barycentric shape functions. By default, this property is **On**.

With parametric elements, all variables are interpolated by a set of explicit functions. These functions allow the evaluation of the variables in the interior points of the element and thus permit higher-order integration.

With barycentric elements, the interpolating functions are provided only implicitly and all variables are evaluated only at element vertices. As a result, the numerical integration is less accurate than with parametric shape functions.

Solver Frozen

When **On**, prevents the solver from updating the solution while iterating. By default, this property is **Off**.

Contents:

Passive Scalar Model

Passive Scalar Model

Viscous flow uses a distinct version of the Passive Scalar model.

Theory	See Passive Scalars in Viscoelastic Fluid Flow	
Provided By	**[physics continuum]** > **Models** > *Optional Physics Models*	
Example Node Path	**Continua** > **[Fluid Continuum]** > **Models** > **Passive Scalar**	
Requires	*Flow*: Viscous Flow	
Properties	See Passive Scalar Model Properties below.	
Specific Right-Click Actions	See Passive Scalar Right-Click Actions below.	
Activates	Model Controls (child nodes)	**Passive Scalars** manager
	Materials	See Passive Scalar Material Node.
	Initial Conditions	See Setting an Initial Value of Passive Scalar.
	Boundary Inputs	See Setting Passive Scalars at Boundaries.

Passive Scalar Model Properties

SUPG

The value for the Streamline-Upwind-Petrov-Galerkin advection stabilization term, β_{SUPG}in Eqn. (1732).

The default value is `1.0`.

Passive Scalar Right-Click Actions

To introduce a node for a new passive scalar, right-click the manager node at **Passive Scalar** > **Passive Scalars** and select **New**.

Rheology Models

The available rheology models for viscous flow are the Generalized Newtonian model and the Viscoelastic model.

Generalized Newtonian Model

The Generalized Newtonian model simulates the behavior of viscous non-Newtonian liquids. The type of Generalized Newtonian fluid can be selected through the Dynamic Viscosity property.

Theory	Generalized Newtonian Fluid	
Provided by	**[physics continuum]** > **Models** > **Rheology**	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Generalized Newtonian**	
Requires	*Flow*: **Viscous Flow**	
Activates	Physics Models	Automatically selected models: *Equation of State*: **Constant Density** *Viscous Regime*: **Laminar**
	Material Methods	See Dynamic Viscosity

Viscoelastic Model

The Viscoelastic model expresses the behavior of the fluid as the sum of the behaviors of 1–8 modes. Each mode acts according to a rheology law specified for it under the **Material Properties** node of the model. In this way, you can model behaviors that are intermediate between those behaviors produced by individual models.

Theory	Viscoelastic Fluids	
Provided by	**[physics continuum]** > **Models** > **Rheology**	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Viscoelastic**	
Requires	*Flow*: **Viscous Flow**	
Properties	See Viscoelastic Properties	
Activates	Physics Models	Automatically selected models:

		Equation of State: **Constant Density** *Viscous Regime*: **Laminar**
	Material Methods	Viscoelastic mode n
	Monitors	**Constitutive *n*** **ExtraStress *n***
	Field Functions	**Extra Stress** n **Rate of Deformation** **Total Viscosity** **Viscoelastic Equivalent Viscosity**

Viscoelastic Properties

Number of Modes

The number of nodes **Viscoelastic mode n** that appear under **Liquid** > **[liquid]** > **Material Properties**. See Viscoelastic Mode N.

SUPG

The value for the Streamline-Upwind-Petrov-Galerkin advection stabilization term. The default value is `0.3`.

DEVSS

The Deviatoric Explicitly Viscous Split Stress stabilization factor for velocity-stress coupling. The default value is `1.0`.

Damping

A non-dimensional parameter for use with the **PhanThien-Tanner/JS.-exp** method. Set the parameter to a non-zero value to make convergence easier, at a cost of additional iterations. The default value is `0.0`.

Steady-State terms only

When **On**, this property removes the transient term dT/dt in the viscoelastic constitutive equation and introduces pseudo-equilibrium stresses. See Viscoelastic Fluids. This property is **Off** by default.

Material Property Methods for Rheology

Dynamic Viscosity

You can obtain suitable parameter values for the following viscosity models by curve-fitting an experimentally determined flow curve, giving the viscosity as a function of shear-rate, to the viscosity functions given by Eqn. (1514), Eqn. (1515), and Eqn. (1516).

Method	Corresponding Method Node
Constant	**Constant** Exposes *Viscosity* μ in Eqn. (1488). The value is independent of shear rate and temperature. See Newtonian Fluids.
Field Function	**Field Function** Allows you to set a scalar field function to specify viscosity.
Newtonian Allows you to specify a viscosity that depends on temperature but is independent of shear rate. Temperature dependency is achieved by time-temperature superposition. For viscoelastic models, the specified viscosity and the relaxation time λ are shifted according to the selected temperature-shift law. The shifted viscosity replaces the polymer viscosity μ_0 in the viscoelastic consitutive equations Eqn. (1502) to Eqn. (1507). For generalized Newtonian fluids, the specified viscosity is shifted according to the selected temperature-shift law.	**Newtonian** Exposes *Viscosity* μ_{ref} in Eqn. (1513) and Eqn. (1517). This option is available only when the Viscous Energy model is activated and the Temperature Shift Factor is set to **Arrhenius**, **Nahme**, or **Williams-Landel-Ferry**. See Time-Temperature Superposition.
Non-Newtonian Generalized Carreau-Yasuda Fluid	**Non-Newtonian Generalized Carreau-Yasuda Fluid** Exposes the following terms from Eqn. (1516): • *Power Constant* n • *a Parameter* a • *Zero Shear Viscosity* μ_0 • *Infinite Shear Viscosity* μ_∞ • *Relaxation Time* λ Also provides the *Viscosity Under-Relaxation Factor*, which you can reduce to improve convergence.
Non-Newtonian Generalized Cross Fluid	**Non-Newtonian Generalized Cross Fluid** Exposes the following terms from Eqn. (1515): • *Cross Rate Constant* m • *Zero Shear Viscosity* μ_0 • *Infinite Shear Viscosity* μ_∞ • *Critical Shear Rate* $\dot{\gamma}_c$

Non-Newtonian Generalized Power Law This method combines two rheology models: • Herschel-Bulkley • Power Law/Ostwald-de Waele	**Generalized Power Law** Exposes the following terms from Eqn. (1514): • *Consistency Factor* k • *Power Law Exponent* n • *Yield Stress Threshold* τ_0 • *Yielding Viscosity* μ_0 • *Minimum Viscosity Limit* and *Maximum Viscosity Limit*: minimum and maximum values allowed for μ To obtain the Power Law/Ostwald de Waele viscosity function, set the yield stress $\tau_0 = 0$. See Eqn. (1493).

Viscoelastic Mode N

You set the number of Viscoelastic Modes in the Viscoelastic model. STAR-CCM+ adds this number of material property nodes. For each Viscoelastic Mode, choose the *Method* then set the properties within the corresponding method node.

Method	Corresponding Method Node
Oldroyd-B	**Oldroyd-B** Exposes *Viscosity* μ_0 and *Lambda* λ from Eqn. (1502).
GiesekusLeonov	**GiesekusLeonov** Exposes *Viscosity* μ_0, *Lambda* λ, and *Alpha* α from Eqn. (1506).
PhanThien-Tanner/JS.-lin	**PhanThien-Tanner/JS.-lin** Exposes *Viscosity* μ_0, *Lambda* λ, *Epsilon* ϵ, and *Xi* ξ from Eqn. (1504).
PhanThien-Tanner/JS.-exp	**PhanThien-Tanner/JS.-exp** Exposes *Viscosity* μ_0, *Lambda* λ, *Epsilon* ϵ, and *Xi* ξ from Eqn. (1505).
Extended PomPom	**Extended PomPom** Exposes *Viscosity* μ_0, *Lambda* λ, and *Alpha* α from Eqn. (1507). Also *Q* and *1/eps* $1/\epsilon$ from Eqn. (1508).

See Also:

Viscous Flow Model

Free Surface Model

The Free Surface model allows for the simulation of movement and change of shape in the surface of a viscous liquid.

Provided by	**[physics continuum]** > **Models** > **Optional Models**	
Example Node Path	**Continua** > **Physics** > **Models** > **Free Surface**	
Requires	*Flow*: **Viscous Flow**	
Properties	See Free Surface Properties.	
Activates	Boundary Inputs	See Free Surface Boundary Settings.
	Solvers	Mesh Morpher
	Field Functions	**Boundary Displacement** **Grid Flux** **Grid Displacement** **Morpher Displacement** **SavedCoord**
	Motions	Morphing

Free Surface Properties

Extrusion Direction

Specifies the axis along which the viscous flow moves, **X**, **Y**, or **Z**. The default is **Z**.

Kinematic Condition

Locates the boundary according to the velocity direction. **Line** constrains the cell edges and the extrusion direction to be aligned with the velocity vector on the boundary. The looser **Surface** setting only constrains the cell faces to be parallel to the velocity vector. The default is **Line**. See Kinematic Conditions.

Restrain X

Restrain Y

Restrain Z

When one of these properties is **On**, it sets the integral of the velocity component in the given direction to zero. Use these properties when there is no plane of symmetry in a given direction. The default is **Off**.

Restrain Torque

In some cases, the free surface has rotational freedom around the axis of extrusion—for example, when periodic boundary conditions apply, but without a symmetry boundary condition. This rotational degree of freedom can cause arbitrarily large rotations of the free surface, leading to negative volume errors. When this property is **On**, this rotation is constrained. The default is **Off**.

Relax Free Surfaces

When this property is **On**, STAR-CCM+ introduces the kinematic condition progressively. Use it if the fixed domain model converges but the free surface one does not, to locate free surface topological problems. By default, the property is **Off**.

Extrusion Moments

When any of the Restrain properties is **On**, this property specifies the number of moments in the extrusion direction that are set to zero. The higher the number, the stiffer the extrudate becomes. It is seldom useful to go higher than 5. The default value is `3`.

Cross-FS Stabilization

A numerical stabilization term to eliminate irregularities that result from the Line Kinematic Condition. The default value is `0.01`.

Free Surface Boundary Settings

All boundary types but Symmetry Plane

Morpher Specification

Controls how the morphing for a boundary is performed. For more information, see Morpher Boundary Condition Reference. Activated when the **Free Surface** model is selected. For some boundaries, only a subset of the methods that are listed below is available.

Method	Corresponding Physics Value Nodes
Fixed	**Morpher Thinout Factor**
Constraint Adds the following nodes: **Morpher Constraint Specification** **Morpher Plane Constraint Specification** **Traction**	**Morpher Plane Options**
Displacement Adds the **Morpher Displacement Specification** node.	**Morpher Thinout Factor**

Traction

Becomes available when **[boundary]** > **Physics Conditions** > **Morpher Specification** is set to **Constraint**.

Method	Corresponding Physics Value Nodes
No Traction No force is applied at the free surface exit.	**Morpher Thinout Factor** See Morpher Boundary Condition Reference.
Traction Velocity Applies a pulling velocity at the end of the free surface.	**Traction Velocity** Sets the pulling velocity using the chosen *Method*.

Aligned Exit

Applies zero normal force and zero tangential velocity in the extrusion direction, leaving the aligned component free.

Traction Velocity

See above.

Symmetry Plane

Morpher Plane Options

This node gets added to **Physics Values**. For more information, see **Morpher Plane Options**.

See Also:

Viscous Flow Model

Viscous Energy Model

The Viscous Energy model simulates the energy that drives the expansion of a viscous fluid due to shear stress.

Provided by	**[physics continuum]** > **Models** > **Optional Models**	
Example Node Path	**Continua** > **Physics** > **Models** > **Viscous Energy**	
Requires	*Flow*: **Viscous Flow**	
Properties	See Viscous Energy Properties.	
Activates	Material Properties	*Thermal Conductivity* *Specific Heat* *Temperature Shift Factor.* See Materials and Methods.
	Reference Values	**Reference Temperature**
	Initial Conditions	**Static Temperature**
	Boundary Inputs	See Viscous Energy Boundary Settings
	Monitors	**Energy** **Temperature**
	Field Functions	**Specific Heat** **Temperature** **Thermal Conductivity**

Viscous Energy Properties

SUPG

The value for the Streamline-Upwind-Petrov-Galerkin advection stabilization term. The default value is `1.0`.

Materials and Methods

Temperature Shift Factor.

When temperature effects are present, the non-Newtonian liquid models use a shift factor a_T to adjust viscosity with temperature. The *Temperature Shift Factor* applies to both polymer viscosity and relaxation times of viscoelastic modes.

Method	Corresponding Method Node
Arrhenius	**Arrhenius** Exposes the following terms from Eqn. (1510): • Activation energy E_a • Reference temperature T_0 Use the Arrhenius model when the temperature is at least 100°C above the glass transition temperature of the polymer.
Nahme	**Nahme** Exposes the following terms from Eqn. (1511): • Activation energy E_a • Reference temperature T_0 Use the Nahme model when the temperature is close to the reference temperature T_0.
Williams-Landel-Ferry (WLF)	**Williams-Landel-Ferry** Exposes the following terms from Eqn. (1512): • Coefficients C_1 and C_2, positive empirical constants that depend on the material and the reference temperature. See [20] for tables of values for C_1, C_2 • Reference temperature T_0 Use the WLF model when the temperature is close to the glass transition temperature of the polymer.

Viscous Energy Boundary Settings

Free Stream and Wall

Thermal Specification

See the entry in Energy Reference.

Mass Flow Inlet and Velocity Inlet

Static Temperature

See the entry in Energy Reference.

See Also:

Viscous Flow Model

Partial Fill Model

The Partial Fill model allows the simulation of viscous flows moving in a region they do not fully occupy. It facilitates simulation of injection molding, mixing, and compounding.

The model uses the **Filled** physics value to track which parts of the domain are wet, which are dry, and which are partially wet. The viscous liquid fully occupies the wet sections, so the viscous fluid models have full weight there. Partial Fill implements a Level Set method to track the wet regions.

Provided by	**[physics continuum]** > **Models** > **Optional Models**	
Example Node Path	**Continua** > **Physics** > **Models** > **Partial Fill**	
Requires	*Space*: **Three Dimensional** or **Two Dimensional**, or **Axisymmetric** *Time*: **Implicit Unsteady** *Material*: **Liquid** *Flow*: **Viscous Flow**	
Key Properties	***SUPG*** The value for the Streamline-Upwind-Petrov-Galerkin advection stabilization term. The default value is `10`.	
Activates	Physics Models	Automatically selected models: *Equation of State*: **Constant Density** *Viscous Regime*: **Laminar**
	Initial Conditions	**Filled**: see Partial Fill Initial Conditions
	Boundary Inputs	See Partial Fill Boundary Settings
	Solvers	**Filling Solver**
	Field Functions	**Filled**

Partial Fill Initial Conditions

Use this property to define the parts of the domain that are to be considered full at start time.

Filled

Method

The profile that is selected to determine the distribution of the Filled property in the domain.

Value

The volume fraction that is occupied by viscous fluid at the start of the simulation, a value ranging from 0 through 1. The default is `1.0`.

Dimensions

The Filled value is non-dimensional.

Partial Fill Boundary Settings

Free Stream

Filling Condition Specification

Method	Corresponding Value Nodes
Impose no condition	None
Impose filling condition	**Filled** The volume fraction that is occupied by viscous fluid, a value ranging from 0 through 1. The default is `1.0`.

Mass Flow Inlet and Velocity Inlet

Filled

Tracks which parts of the domain are "wet" (flagged with a `1`), which are "dry" (flagged with a `0`), and which are partially wet (between `0` and `1`). The viscous liquid fully occupies the wet sections, so the viscous fluid models have full weight there. Partial Fill implements a Level Set method to track the wet regions.

Method

The profile that is selected to determine the distribution of the Filled property in the domain.

Value

The volume fraction that is occupied by viscous fluid, a value ranging from 0 through 1. The default is `1.0`.

Dimensions

The **Filled** value is non-dimensional.

Filling Solver

The Filling Solver is exposed when you select the Partial Fill physics model.

Properties

Threshold

A value ranging from 0 to 1. When the fraction of a cell that is filled with the viscous liquid is above the threshold value, the cell is treated as part of the viscous flow. The default value is `0.5`.

Solver Frozen

When **On**, prevents the solver from updating the solution while iterating. By default, this property is **Off**.

Film Casting Model

The Film Casting model simulates a thin film of viscoelastic fluid extruded into free space.

Theory	See Film Casting.	
Provided By	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Film Casting**	
Requires	*Optional Models*: **Two Dimensional** *Flow*: **Viscous Flow** *Time*: **Steady** *PolymerDisplacement*: **Free Surface**	
Activates	Boundary Inputs	See Boundary Settings
	Monitors	**Continuity**, **Film Thickness**

Boundary Settings

Velocity Inlet

Film Thickness

The thickness of the film as it leaves the die, h in Eqn. (1519) to Eqn. (1521).

Velocity

The velocity of the film as it leaves the die, $\mathbf{v}$ in Eqn. (1519) to Eqn. (1521).

Mesh Requirements and Guidelines

As the Viscous Flow Solver uses a Finite Element (FE) approach, the volume mesh must meet certain requirements.

The Viscous Flow Solver requires a volume mesh (either imported or generated in STAR-CCM+) of one of the following types:

- 2D mesh of triangular or quadrilateral elements.
- 3D mesh of tetrahedral, hexahedral, prism, or pyramidal elements.

  You are advised to use elements whose aspect ratio is not too large (typically, not more than 10 or 20). When using quad or hex elements, the angles should be as close as possible to 90°, in order to keep Jacobian variations to a minimum.

  Pyramids are generally used as transition elements between different mesh topologies. As this type of elements is highly inaccurate, make sure that the mesh does not contain a large number of pyramids.

  Polyhedral or trimmed meshes are not supported.

Note: The Viscous Flow Solver uses lowest order (linear) elements. If you import a mesh with higher order elements, the solver treats the elements as linear.

The following table shows how to generate compatible mesh topologies using parts-based mesh operations in STAR-CCM+. If you import a CAE model mesh, and you wish to remesh it in STAR-CCM+, you can follow the same procedure.

Dim	Element Type	Mesh Operation	Meshers
2D	Triangle	**Automated Mesh (2D)**	**Triangular Mesher**
	Quad	**Automated Mesh (2D)**	**Quadrilateral Mesher**
3D	Tetrahedron	**Automated Mesh**	**Tetrahedral Mesher**
	Prism	**Automated Mesh**	• **Tetrahedral Mesher** with **Thin Mesher**. In regular thin geometries, such as a thin plate, this combination generates prism elements. In more complex geometries, this combination generates a mix of different elements (mainly tetrahedra and prisms). The meshers can generate pyramids as transition elements. • **Tetrahedral Mesher** with **Prism Layer Mesher**. This combination generates a mesh of tetrahedral elements, with prism boundary layers. When using this combination, always set the *Layer Reduction Percentage* to `0`.
		or **Directed Mesh**	**Directed Mesher** (extrude triangular surface mesh). See Directed Meshing.

Dim	Element Type	Mesh Operation	Meshers
			In extrusion problems, you typically generate an automatic tetrahedral mesh for the die and then extrude the triangular surface mesh to form prisms.
	Hexahedron	**Directed Mesh**	**Directed Mesher** (extrude quadrilateral surface mesh). See Directed Meshing.

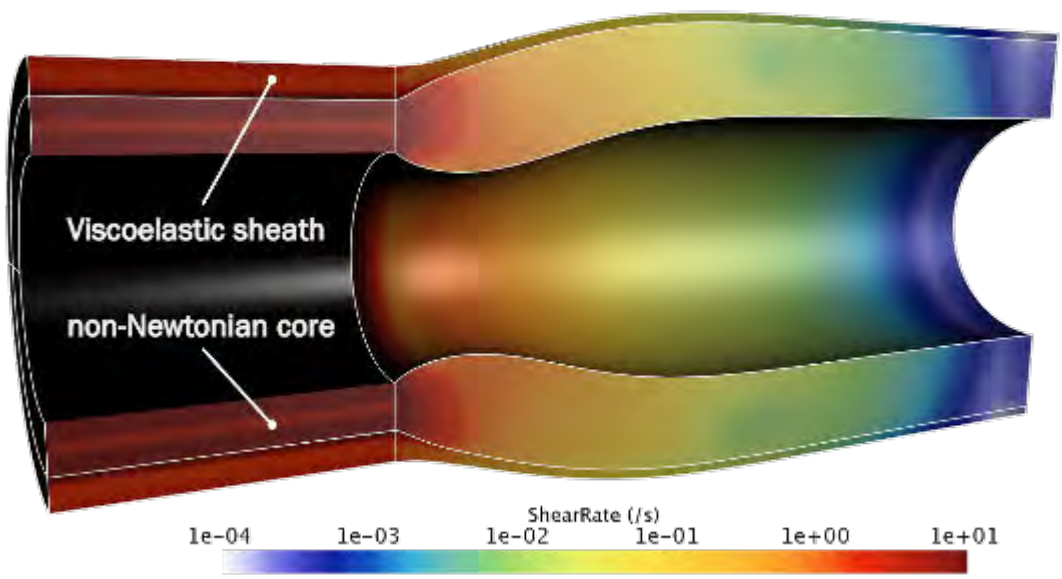
Guidelines for Co-Extrusion

Co-extrusion is extruding two or more materials in contact with each other, for example when an object and its coating are extruded together. The materials must be defined separately, then related to each other by specifying the interface between them.

Use the following guidelines:

- Create a separate region and a separate physics continuum for each extrudate.
- For stable solutions, the material properties of the extrudates must be reasonably comparable. Make sure, for example, that dynamic viscosities are within an order of magnitude of each other. Similarly, keep the shear rate, Reynolds number, and relaxation time within comparable scales.
- Make the aspect ratio of the mesh as close to 1 as possible, to promote convergence.
- Do not use the **No-Slip** method for the **Shear Stress Specification** condition on walls when dealing with two or more materials inside a single die or channel. Use **Slip** instead.
- Under **Tools** > **Motions** > **Morphing**, activate the *Linear Fitter* property. This calculates a least-squares fit for a linear transformation between old and new coordinates. See the Linear Fitter Option.
- Prevent the morpher from overshooting along the symmetry plane. To achieve this, apply a non-zero morpher Damping Factor, typically 0.05–0.1.
- Use Internal Interface Boundary for the contact areas between extrudates.

The figure below shows an example of co-extrusion, with the extrudates for a hose and sheath in contact. Note that the two extrudates use different physics models.



Guidelines for the eXtended Pom-Pom Model (XPP)

The eXtended Pom-Pom (XPP) model applies to complex flows of branched polymers (the "pom-poms") with significant shear flow and extension. To enhance convergence of simulations that use the XPP model, you are advised the following:

- Create a locally refined mesh with a cell aspect ratio close to 1 in regions of the geometry where variables change suddenly, such as stagnation points and next to corners or solid walls. This high mesh resolution is particularly important for advection-dominated cases with high Weissenberg (Wi) or Péclet (Pe) numbers in non-isothermal flows.
- Gradually increase cell size and aspect ratio moving away from the critical regions.
- Use parametric shape functions in the Viscous Flow solver, not barycentric ones.
- In general, keep the SUPG property at the default value. In some cases, increasing the value of *SUPG* to 1 can improve convergence, but this increase can also impair accuracy. To check this potential problem, make sure the results do not depend on the aspect ratio or scale of the mesh.

Viscous Flow Field Function Reference

The following primitive field functions become available with the activation of viscous flow models. The models required are noted in each entry.

Boundary Displacement

The displacement of a free surface or exterior interface, given to the morphing function for mesh adaptation. (Free Surface)

Extra Stress n

The terms **Extra Stress 1**, **Extra Stress 2**, ... that sum to $\mathbf{T}^p$ in Eqn. (1498). There is a term for each viscoelastic mode used. (Viscoelastic)

Filled

The fraction of volume in a region occupied by the viscous flow, expressed as a value between `0` and `1`. (Partial Fill)

Grid Flux

The rate at which a face sweeps out volume due to grid motion. $G_f = \mathbf{v}_g \cdot \mathbf{a}_f$, where $\mathbf{v}_g$ is the grid velocity and $\mathbf{a}_f$ is the face area. (Free Surface)

Morpher Displacement

Vector value of the distance by which the morpher has moved the mesh vertices (not face centers) in the current time-step. As this field is a vector field function, you can plot the x-, y-, z- components, and magnitude. (Free Surface)

Pressure

The pressure of the viscous flow p which is a factor in the stress tensor $\boldsymbol{\sigma}$ in Eqn. (1451). (Viscous Flow)

Rate of Deformation

The rate of deformation tensor $\mathbf{D}$ in Eqn. (1487). (Viscoelastic)

SavedCoord

This field contains the coordinates before the morphing. (Free Surface)

Shear Rate

Flow shear rate $\dot{\gamma}$ in Eqn. (1493). (Viscous Flow)

Specific Heat

The specific heat of the viscous fluid. (Viscous Energy)

Temperature

The temperature of the viscous fluid. (Viscous Energy)

Thermal Conductivity

The thermal conductivity of the viscous fluid. (Viscous Energy)

Total Viscosity

The sum of the solvent viscosity μ_s in Eqn. (1499) and the **ViscoelasticEquivalentViscosity** (see below).

Use **TotalViscosity** for shear driven flows or elongational flows where there are non-zero shear strain rates. Do not use it in flows with free surfaces or axes of symmetry. In complex flows where the shear strain rates drop to very small values, **TotalViscosity** becomes very high and can result in singularities. (Viscoelastic)

Velocity

The velocity of the viscous flow. (Viscous Flow)

Viscoelastic Equivalent Viscosity

The ratio $\frac{II_T}{II_D} = \frac{(tr[\mathbf{T}])^2 - tr\left(\mathbf{T}^2\right)}{(tr[\mathbf{D}])^2 - tr\left(\mathbf{D}^2\right)}$ of the second invariants of the stress tensor $\mathbf{T}$ and the rate of deformation tensor $\mathbf{D}$, providing an estimate of the shear resistance of the viscoelastic material with respect to the local shear rate in the system. See Eqn. (1500) and Eqn. (1501). (Viscoelastic)

Modeling Heat Transfer

Heat transfer is the study of energy in transit due to a temperature difference in a medium or between media. Heat transfer extends thermodynamic analysis through the study of the modes of energy transfer and through development of relations to calculate energy transfer rates.

This transfer of heat takes place through three distinct modes of heat transfer:

- Heat Conduction
- Convective Heat Transfer
- Thermal Radiation

In most problems, heat transfer is coupled to fluid flow through convection, and coupled to thermodynamics through fluid properties (for example, density, material conductivity, and fluid viscosity) and phase change (for example, condensation, evaporation, melting, and freezing). Therefore, from a modeling perspective, heat transfer must be seen as a multi-physics activity.

STAR-CCM+ provides additional models that let you create sophisticated simulations for specific applications:

- Boiling — where you are interested in modeling the improved heat transfer due to boiling, and the creation of the vapor bubbles has little effect on the flow
- Moisture Boiling in Solids — where you are interested in modeling the boiling of moisture within a solid material
- Thin Film — where you are interested in simulating deicing or defogging on a surface
- Heat exchangers — heat transfer between two fluid streams: a hot and a cold stream (such as in radiators and air coolers)

Contents:

Heat Conduction
Convective Heat Transfer
Thermal Radiation
Verification and Validation of Results
Designing a Heat Transfer Optimization Study
Convective Heat Transfer Coefficients
Setting an Environment Boundary Condition for Heat Transfer
Modeling Boiling
Modeling Moisture Boiling in Solids
Modeling Thin Film: Deicing and Defogging
Modeling Heat Exchangers
Modeling Energy in Solids

Energy Reference
Heat Transfer Field Functions Reference
Energy Formulation

Heat Conduction

Heat conduction is the flow of internal energy from a region of higher temperature to a region of lower temperature by the interaction of the adjacent particles (for example, atoms, molecules, ions, or electrons) in the intervening space. This flow can be viewed as the transfer of energy from the more energetic particles to the less energetic particles of a substance due to interactions between the particles.

Conduction takes place in all forms of matter: solids, liquids, gases, and plasmas. In solids, heat conduction is due to the combination of vibrations of the molecules in a lattice and diffusion of free electrons. In gases and liquids, heat conduction is due to collisions between molecules and the diffusion of molecules during their random motion. Conduction is greater in solids because of the relatively close fixed spatial relationships between atoms, whereas fluids and gases are less conductive due to the large distance between atoms.

The law of heat conduction, also known as Fourier's law, provides a means of calculating the local heat flux. This law can be stated in two equivalent forms:

- The integral form, which considers the amount of energy flowing into or out of a body as a whole.
- The differential form, which considers the flow rates or fluxes of energy locally:

$$\mathbf{q}'' = -k\nabla T \tag{294}$$

  In this expression, $\mathbf{q}''$ $\left[W/m^2\right]$ is the local heat flux vector, k $[W/mK]$ is the thermal conductivity of the material, and ∇T $[K/m]$is the temperature gradient. The thermal conductivity of the material can be a strong function of temperature, pressure (for gases), spatial location, and can exhibit anisotropic behavior (that is, it varies with orientation and is represented by a second-order tensor). The negative sign in the expression is because $k > 0$ and heat always moves from a high temperature region to a low temperature (as defined in the Second Law of Thermodynamics).

Thermal conductivities vary from material to material, being the largest for metallic solids, smaller for nonmetallic solids, very small for liquids, and extremely small for gases.

The major objective of performing a simulation that includes conductive heat transfer is to determine the temperature field in a medium, taking into account the conditions that are imposed on the boundaries. The First Law of Thermodynamics (that is, Conservation of Energy) and Fourier's law can be used to derive the Heat Diffusion Equation (a second-order partial differential equation in temperature, or enthalpy, or energy) from which the temperature field can be calculated. This temperature solution can be steady (a function of space only and not changing with time), or unsteady (the temperature field varies in both space and time as, for example, in metal quenching applications).

Convective Heat Transfer

Convective heat transfer is the transfer of thermal energy by the movement of a fluid. In engineering practice, it is used to provide specific temperature changes. Examples include heat exchangers, electronic device cooling, and cooling of turbine blades.

Convective heat transfer comprises two mechanisms: energy transfer due to random molecular motion (diffusion), and energy transfer due to the larger-scale bulk motion of the fluid (advection - the collective movement of groups of molecules).

Convective heat transfer to or from a solid requires the transfer of heat by the bulk motion of the fluid and the transfer of heat by diffusion. The contribution due to random molecular motion dominates near the surface where the fluid velocity is low. At the interface between the surface and the fluid, the fluid velocity is zero (that is, there is no relative motion between the fluid and surface) and heat transfer is transferred by diffusion only.

Convection is usually described as being either natural (”free”) or forced, although other mechanisms also exist (for example, gravitational, granular, thermomagnetic, capillary action-driven, or Marangoni and Weissenberg driven effects).

Convective heat transfer at a surface is governed by Newton’s law of cooling:

$$q_s'' = h\left(T_s - T_{ref}\right) \tag{295}$$

In this expression, q_s'' $\left[W/m^2\right]$ is the local surface heat flux, h $\left[W/m^2K\right]$ is the local convective heat transfer coefficient, T_s is the surface temperature, and T_{ref} is a characteristic temperature of the fluid moving over the surface.

Newton’s law of cooling expresses a linear relationship between the local surface heat flux and the difference between the local surface and fluid temperatures. This linear relationship is only an approximation: in reality the relationship can be strongly nonlinear. Because flow conditions can vary from point to point on the surface, both q_s'' and h can vary as a function of space and time.

Contents:

Natural Convection
Forced Convection

Natural Convection

In a gravitational field, natural convection occurs due to temperature differences which affect the density, and thus the relative buoyancy, of the fluid. Denser components fall, while lighter (less dense) components rise, leading to bulk fluid movement.

Natural convection is more likely and more rapid when any of the following conditions apply:

- a greater variation in density between two mixing fluids
- a larger acceleration due to gravity that drives the convection
- a larger distance through the convecting medium.

Natural convection is less likely and/or less rapid when any of the following conditions apply:

- a more rapid diffusion (which acts to diffuse away the thermal gradient that causes the convection)
- a more viscous (sticky) fluid.

To quantify when natural convection is possible, the dimensionless Rayleigh number is defined and can be viewed as the ratio of buoyant and viscous forces times the ratio of momentum and thermal diffusivities. Natural convection becomes possible when the Rayleigh number exceeds 10^5.

Forced Convection

In forced convection, fluid movement results from external sources (for example, a fan, pump, or the action of a propeller), and is typically used to increase the rate of heat transfer at a surface in cooling or heating applications.

In cases of mixed convection (that is, natural convection and forced convection occurring together) you can determine how much is due to forced convection and how much is due to natural convection. The relative magnitudes of the Rayleigh (Ra), Prandtl (Pr), and Reynolds (Re) numbers indicate which form of convection dominates:

If $\frac{Ra}{Pr\mathrm{Re}^2} \gg 1$, natural convection dominates, forced convection is neglected.

If $\frac{Ra}{Pr\mathrm{Re}^2} \ll 1$, forced convection dominates, natural convection is neglected.

If $\frac{Ra}{Pr\mathrm{Re}^2} \simeq 1$, mixed convection occurs, neither forced or free convection dominate, so both are taken into account.

Thermal Radiation

Thermal radiation is the emission of electromagnetic waves from all matter that has a temperature greater than absolute zero, and represents a conversion of thermal energy into electromagnetic energy. It is generated by the thermal motion of charged particles in matter, which results in charge-acceleration and dipole oscillation. This drives the electrodynamic generation of coupled electric and magnetic fields, which cause the emission of thermal radiation.

Electromagnetic radiation, or light, does not require the presence of matter to propagate, and travels the fastest in a vacuum. All forms of matter emit radiation. For gases and some semi-transparent solids (for example, glass and salt crystals at elevated temperatures) emission is a volumetric phenomenon (that is, emission is an integrated effect of local emission throughout the volume). For most solids and liquids, radiation that is emitted from the interior molecules is strongly absorbed by adjoining molecules. Therefore, because all of the emitted radiation originates from molecules near the surface, it can be considered as a surface phenomenon.

Electromagnetic energy propagation can be described at various levels of complexity, depending on the features of the radiation field to be described. For example, if polarization is important, the Tokes parameters are used to describe the intensity field. If wave effects are important, the vector wave quantities from electromagnetic theory such as electric vector, magnetic vector, and the Poynting vector are used. Wave effects become significant when the characteristic length scale for transport, such as spacing between the surfaces, is similar to the wavelength.

For more information, see Modeling Radiation.

For calculating heat transfer in most situations, however, this level of description is overly complicated. For most heat transfer applications, thermal radiation can be treated as unpolarized (multiple reflections and scattering usually nullify polarization effects) and incoherent (waves or photons are usually out of phase). The length scale for transport is usually much larger than the wavelength of radiation, so the limiting description of geometric optics (that is, the wavelength approaches zero) can be applied: the waves are described as bundles of rays carrying energy in a small volume that is associated with the solid angle in the direction of the rays.

The maximum flux at which radiation can be emitted from a surface is given by the Stefan-Boltzmann law:

$$\mathrm{q}_{bb}'' = \sigma {T_s}^4 \tag{296}$$

In this expression, q_{bb}'' $\left[W/m^2\right]$ is the local surface heat flux, T_s is the temperature of the surface, and σ is the Stefan-Boltzmann constant $\sigma = 5.67 \times 10^{-8}$ $\left[W/m^2K^4\right]$. Such a surface is called an ideal radiator or blackbody.

The heat flux that is emitted by a real surface is less than the heat flux of an ideal radiator, and is given by:

$$\mathrm{q}_e'' = \varepsilon q_{bb}'' = \varepsilon\sigma {T_s}^4 \tag{297}$$

where $0 \leq \varepsilon \leq 1$ is the surface emissivity (that is, the relative ability of a surface to emit energy by radiation).

If radiation is incident upon a surface, a portion is absorbed, a portion is reflected, and a portion is transmitted through the material:

$$q_{inc}'' = q_{abs}'' + q_{ref}'' + q_{trans}'' = \alpha q_{inc}'' + \rho q_{inc}'' + \tau q_{inc}'' \tag{298}$$

The constants in this expression are:

$0 \leq \alpha \leq 1$	The surface absorptivity: the fraction of incident radiation that gets absorbed.
$0 \leq \rho \leq 1$	The reflectivity: the fraction of incident radiation that gets reflected away from the surface.
$0 \leq \tau \leq 1$	The transmissivity: the fraction of incident radiation that gets transmitted through the material.

Absorptivity, reflectivity, and emissivity depend on the surface temperature, surface roughness, emission angle, and wavelength of the radiation (the radiation is not monochromatic, but comprises a continuous dispersion of photon energies).

Verification and Validation of Results

Verification and validation are independent procedures that are used together to ensure that a code meets requirements and specifications, and that it fulfills its intended purpose. In simple terms, validation can be expressed by the query ”Are you building the right thing?” and verification by the query ”Are you building it right?”

In other words, verification determines whether the programming and computational implementation of the conceptual model (that is, the partial differential equation system) is correct by comparison to exact analytical results. For example, does the temporal and spatial rate of convergence match the theory? Verification is more concerned with mathematics than engineering. By contrast, validation determines whether the computational simulation agrees with physical reality by comparison to experimental results. For example, in quenching, does the temporal variation of the surface temperature match experimental results?

Because experimental data is not always available for complex multi-physics applications, you must use a building-block approach consisting of phases that involve successively more complex flow physics, geometry, and their interactions. When validating multi-physics problems combined with complex geometry and forcing, the initial validation problem should be as simple as possible and only involve single physics and simple geometry. For example, if you are interested in forced flow over a cooled turbine blade, you could start with isothermal flow over a flat plate. Once the simulation code has been validated using single physics and simple geometry, you can add more physics and increased geometrical complexity until the actual complete system is being simulated (that is, all of the relevant physics and geometry are present).

Designing a Heat Transfer Optimization Study

In mathematics, optimization is the selection of the best element (with respect to some criteria) from a set of available alternatives. In the simplest case, an optimization problem consists of maximizing or minimizing a real function (that is, the objective function) by systematically choosing input values from within an allowed set and computing the value of the function.

More generally, optimization includes finding the “best available” values of some objective function given a defined domain, including a variety of different types of objective functions and constraints. Multi-objective optimization (that is, an optimization problem that has more than one objective function) adds significant complexity to optimization problems. It is not always possible to satisfy all of the objective functions, so a trade-off can be required. For example, you cannot minimize both the surface temperature and the surface heat flux. Optimization problems are also often multi-modal: they possess multiple good solutions that can all be globally good (that is, they have the same objective function value) or the solutions could be a mix of globally and locally good solutions. When classical optimization techniques are used to obtain multiple solutions, they do not perform satisfactorily due to their iterative approach. That is, it is not guaranteed that different solutions are obtained even with different starting points in multiple runs of the algorithm. However, evolutionary algorithms are able to obtain multiple solutions and have become a popular technique for cases in which more than one solution is desired.

As far as the engineer is concerned, there are two important objectives that optimization can help with:

- Code tuning and validation

  For example, to reproduce previous computations or experiments of interest.
- Meeting design specifications and constraints by allowing you to choose the ”best” set of models, parameters, and geometry for the current application.

  For example, what shape maximizes heat flux but is constrained to have a fixed volume.

When you use optimization in flow and heat transfer applications, address the following points:

- Decide on the objective function (also known as a cost function or utility function) that you are interested in minimizing or maximizing.

  For example, minimize the maximum surface temperature, minimize the downstream vorticity after a pipe bend or junction, or maximize the surface heat flux.
- Decide on the set of models and parameters that are important and need to be considered in the search for objective function extrema.

  For example, material properties, length of a boundary, surface shape, domain size, and boundary conditions.

  It is useful to perform a one-parameter-at-a-time sensitivity study over the input parameter space, and identify any parameters that have a significant impact on the solution (these parameters are sometimes referred to as ”big knobs”). You can measure sensitivity by monitoring changes in the objective function value, looking at its partial derivatives, or by using linear regression. However, this approach does not fully explore the input space, since it does not take into account the simultaneous variation of input variables, and therefore cannot detect input variable interactions. Modelers frequently prefer this technique because of its simplicity and because when the model fails the modeler knows immediately which input parameter caused the failure.

- The mathematical techniques that are used to solve optimization problems depend on the form of the objective function and constraints, as well as the desired use of the results (for example, code tuning or design).

  You must decide which optimization technique best fits the current problem: is finding a local solution good enough, or must you find the global solution, or do you require multiple solutions?

The simplest situations are unconstrained problems, or problems in which all the constraints can be expressed as equality relationships. For these types of problems, the technique of Lagrangian multipliers can be used efficiently to find the optimal solution.

Another technique that is frequently used is response surface analysis. In this technique, you define a finite set of plausible parameter value collections (for example, based on uniform latin hypercube sampling) using the truncated input parameter space (obtained from, for example, the one-parameter-at-a-time sensitivity analysis), and then fit a response surface (either local or global) to the objective function results. Once the response surface has been created, it is easy to use calculus to define a system whose solutions represent the minimum or maximum of the objective function over the truncated input parameter space considered. However, solving this system is not always easy. This technique works well when the objective function is locally or globally smooth with respect to input parameter space. This smoothness makes it possible to get a good representation of the solution hyper-surface with the least number of objective function values.

For problems that involve inequality relationships, the Lagrangian multiplier method is not suited to solving problems of this type efficiently. Modern mathematical programming techniques, such as linear- and nonlinear-programming, in which both the objective and constraint can be linear or non-linear functions, must be used to find optimal solutions.

Convective Heat Transfer Coefficients

Convective heat transfer is the transfer of energy between a surface and a fluid moving over the surface. It is primarily used as a boundary condition when solving solid/fluid heat transfer problems, such as quenching, electronic device cooling, and heat exchangers.

Convective heat transfer is follows Newton's law of cooling:

$$q_s'' = h\left(T_s - T_{ref}\right) \tag{299}$$

where:

- q_s'' is the local surface heat flux (that is, power per unit area).
- h is the local convective heat transfer coefficient.
- T_s is the surface temperature.
- T_{ref} is a characteristic temperature of the fluid moving over the surface.

Newton's law of cooling expresses a linear relationship between the local surface heat flux and the difference between the local surface and fluid temperatures, and is only an approximation. Because flow conditions can vary from point to point on the surface, both q_s'' and h can vary as a function of space and time.

In Eqn. (299), q_s'', T_s, and T_{ref} are fundamental in nature, while the heat transfer coefficient is a constant of proportionality that relates the three fundamental parameters. While q_s'' and T_s are unambiguous, there is some latitude in the choice of the fluid temperature T_{ref}. Depending upon the choice of T_{ref}, the heat transfer coefficient is different in order to satisfy Eqn. (299). Therefore, heat transfer coefficients cannot be defined without also defining a reference temperature: there is an infinite number of heat transfer coefficient and reference temperature combinations that give the same surface heat flux.

In STAR-CCM+, the conceptual centerpiece for modeling convective heat transfer at the wall for turbulent flows stems from the standard wall functions (SWF). Consider the expression for the local surface heat flux:

$$q_s'' = \frac{\rho_f(y_c)C_{p,f}(y_c)u^*}{T^+(y^+(y_c))}\left(T_s - T_c\right) \tag{300}$$

You can then equate the terms from Eqn. (299) and Eqn. (300) and define an h in terms of the local flow conditions.

In this equation:

- ρ_f is the fluid density.
- $C_{p,f}$ is the fluid-specific heat capacity.
- u^* is a reference velocity.
- T^+ is the dimensionless temperature.
- $y^+ = \left(u^* y_c\right)/\upsilon_f$ is the dimensionless wall distance.
- y_c and T_c are the normal distance and temperature of the near-wall cell, respectively.

The Standard Wall Functions (SWFs) are a set of semi-empirical functions that are used to satisfy the flow physics in the near-wall region (the boundary layer). They give relationships for T^+ and u^* in terms of the laminar and turbulent Prandtl numbers, the dimensionless near-wall flow velocity, and the turbulent kinetic energy (see Wall Laws Formulation).

Four heat transfer coefficients are defined in STAR-CCM+:

- Local Heat Transfer Coefficient
- Specified y+ Heat Transfer Coefficient
- Heat Transfer Coefficient
- Virtual Local Heat Transfer Coefficient

These heat transfer coefficients are all grounded to varying degrees in the standard wall function treatment. The local heat transfer coefficient is the one that is used internally within the STAR-CCM+ solver. The other three heat transfer coefficients are computed during post-processing. In the case, of multi-phase flow, volume-fraction-weighted versions of these heat transfer coefficients and reference temperatures are defined for use with co-simulation or coupling to other CAE codes.

The differences between these three heat transfer coefficient post-processing options are discussed in Guidelines for Heat Transfer Coefficients.

Contents:

Local Heat Transfer Coefficient
Specified y+ Heat Transfer Coefficient
Heat Transfer Coefficient
Virtual Local Heat Transfer Coefficient
Guidelines for Heat Transfer Coefficients

Local Heat Transfer Coefficient

The heat transfer coefficient option that is alluded to by Eqn. (300) is the one that the STAR-CCM+ code uses internally to calculate the local surface heat flux.

If you equate the Local Heat Transfer Coefficient and Local Reference Temperature terms from Eqn. (299) and Eqn. (300):

$$h_1 = \frac{\rho_f(y_c)C_{p,f}(y_c)u^*}{T^+(y^+(y_c))} \tag{301}$$

and $T_{ref,1} = T_c$

you have the pair (h_1 and $T_{ref,1}$) which can be viewed or exported.

In STAR-CCM+, the local surface heat transfer coefficient is always calculated (and used internally) in the code using the Standard Wall Functions and Eqn. (301). However, there can be times when you want to view or export heat transfer coefficients that are defined in other ways. For example, to compare the results to previous work or to use with other applications. In general, you can accomplish this by post-processing the simulation results and defining the one-to-one linear mapping (based on equating the surface heat fluxes):

$$h_{user} = h\left(\frac{T_s - T_c}{T_s - T_{ref}}\right) \tag{302}$$

In this way, you can define your own h and T_{ref} pair, where h, T_c, and T_s are known from the simulation, and you pick T_{ref} to suit your requirements.

Specified y+ Heat Transfer Coefficient

The first built-in post-processing option in STAR-CCM+ calculates the heat transfer coefficient as in Eqn. (301), but at a user-specified y^+ value instead of the value that is associated with the near-wall cell:

$$h_2 = \frac{\rho_f(y_c)C_{p,f}(y_c)u^*}{T^+(y^+{}_{user})} \tag{303}$$

Then, using the surface heat flux as calculated in Eqn. (300), you solve Eqn. (299) for the reference temperature:

$$T_{ref,2} = T_s - \frac{q_s''}{h_2} \tag{304}$$

The result is the pair (h_2 and $T_{ref,2}$) which can be viewed or exported.

This definition of the reference temperature does not guarantee that $T_{ref,2}$ lies within the minimum and maximum temperature range of the local fluid region. If $T_{ref,2}$ lies outside this temperature range, then either the minimum or maximum temperature is used in its place, and the heat transfer coefficient is modified accordingly.

Heat Transfer Coefficient

The second built-in post-processing option in STAR-CCM+ calculates the surface heat flux using Eqn. (300), and then recasts Eqn. (299) as:

$$h_3 = \frac{q_s''}{T_s - T_{ref,3}} \tag{305}$$

The heat transfer coefficient is calculated using a user-defined reference temperature ($T_{ref,3}$).

The result is the pair (h_3 and $T_{ref,3}$) which can be viewed or exported.

Virtual Local Heat Transfer Coefficient

The third built-in post-processing option in STAR-CCM+ does not need to solve the energy equation in the fluid.

Instead, this option uses the Local Heat Transfer Coefficient (Eqn. (301)), which is calculated using only flow field information. You specify the specific heat, the molecular Prandtl number, and the turbulent Prandtl number as reference values for this heat transfer coefficient.

The result is the pair (h_1 and $T_{ref,4}$) which can be viewed or exported.

Guidelines for Heat Transfer Coefficients

This section discusses the recommendations for accurate use of the Standard Wall Functions and the built-in post-processing heat transfer coefficients.

STAR-CCM+ always uses Eqn. (300) to obtain the local surface heat flux when solving for fluid temperatures (and/or solid temperatures in CHT applications). This expression embodies the important boundary layer physics, and it is important that you follow the recommendations to ensure its proper application.

The heat transfer coefficients are STAR-CCM+ post-processing results which can be used for comparing to other solutions, visualization, or exporting to other applications such as Abaqus or Nastran. The particular choice of heat transfer coefficient does not affect the simulation results, except in the case when STAR-CCM+ is two-way-coupled to another application that solves for the temperature in the solid. The choice of heat transfer coefficient that is applied in an external application does affect the results in that application. In this case, it is important that you choose the correct option, as the choice of heat transfer coefficient that is exported to the external application does affect the STAR-CCM+ results.

However, if you are solving for both the fluid and solid temperatures within STAR-CCM+ (that is, Conjugate Heat Transfer), your choice of post-processing heat transfer coefficients option has no effect on the simulation results.

In general, standard wall functions give reasonable accuracy for most high-Reynolds-number, wall-bounded flows. However, standard wall functions reach their limitation when the flow conditions differ too much from the ideal conditions that are used to define the functions. This limit can be reached in the following cases:

- Pervasive low-Reynolds-number or near-wall effects (for example, flow through a small gap or highly viscous low-velocity fluid flow).
- Massive transpiration through the wall (such as blowing or suction).
- Severe pressure gradients leading to boundary layer separation.
- Strong body forces (for example, flow near rotating disks or buoyancy-driven flows).
- High three-dimensionality in the near-wall region (for example, Ekman spiral flow or strongly skewed 3D boundary layers).

Thus to get good results when using the standard wall functions, the problem should not contain any of the physics conditions that are outlined above. The y^+ of the near-wall cell should be in the inertial sublayer ($30 < y^+ < 150$), and temperature-dependent fluid properties should be used to ensure maximum accuracy. If the near-wall cell lies in the buffer layer ($5 < y^+ < 30$) between the viscous and inertial sublayers, the “All y^+” option (default) should be used which smoothly connects the viscous and inertial sublayers.

The differences between the heat transfer coefficient post-processing options are discussed in the following sections:

- Heat Transfer Coefficient
- Local_Heat_Transfer_Coefficient
- Specified_y__Heat_Transfer_Coefficient
- Virtual_Local_Heat_Transfer_Coefficient

Contents:

Heat Transfer Coefficient
Local Heat Transfer Coefficient
Specified y+ Heat Transfer Coefficient
Virtual Local Heat Transfer Coefficient

Heat Transfer Coefficient

This option is primarily used for comparing heat transfer coefficients to experimental data or other previous work.

When you use this option, note the following:

- It is recommended that you follow all recommendations for getting accurate results using the standard wall functions.
- For external flows where there is a well-defined free-stream temperature outside the thermal boundary layer, the usual practice is to use this temperature as the reference temperature.

  For internal flows without axial recirculation, it is often practical to define the reference temperature as the local bulk temperature that is obtained by solving the one-dimensional (stream-wise) energy equation. If the bulk temperature varies little relative to the overall fluid-wall temperature difference, you can set the reference temperature to the inlet temperature.
- It is recommended that temperature-dependent fluid properties be used to ensure maximum accuracy.

  If you use temperature-dependent fluid properties, all fluid properties should be temperature-dependent, rather than just some of the properties.
- A poor choice of reference temperature can lead to a negative heat transfer coefficient.
- This option is not recommended for coupled use with other applications.

Local Heat Transfer Coefficient

When you use this option, note the following:

- To ensure maximum accuracy, use temperature-dependent fluid properties.

  If you use temperature-dependent fluid properties, all fluid properties should be temperature-dependent, rather than just some of the properties. For example, to get the Prandtl Number correct you must account for temperature dependence in both the momentum and thermal diffusivity.
- Because y^+ of the near-wall cell can lie in the viscous sublayer, the inertial sublayer, or anywhere in between, this option can give heat transfer coefficients that are different from the values that are found in classic textbooks.
- This option is not recommended for coupled use with other applications unless there is a reason for not using Specified y^+ Heat Transfer Coefficient.

  For more information, see Specified_y+_Heat_Transfer_Coefficient.
- The local heat transfer coefficient takes the temperature of the wall of the adjacent cell as the reference temperature unless there is fluid film on the cell surface. In that case, the local heat transfer coefficient uses the temperature of the fluid film as the reference temperature.

Specified y+ Heat Transfer Coefficient

When you use this option, note the following:

- This heat transfer coefficient and reference temperature pair is recommended to be used when coupling STAR-CCM+ to other applications.

  This heat transfer coefficient and reference temperature pair is less dependent on the mesh than the Local Heat Transfer Coefficient, gives better conditioned systems at solve time, and gives the fastest convergence for a given number of coupling-iterations. One coupling-iteration involves solving for the flow field given the initial solid wall temperature and then solving for the solid temperature field given h and T_c (that is, the near wall cell temperature) from the flow field computation. A single coupling-iteration can give good results, although at least two coupling-iterations are recommended.

  For accuracy and numerical stability reasons, it is not recommended that you use alternative techniques of exchanging boundary conditions between STAR-CCM+ and other applications. For example, when the fluid code passes the local heat flux to the solid code and the solid code passes the local wall temperature to the fluid code.
- This option can have implications for cycle averaging and mapping.
- It is recommended that you follow all recommendations for getting accurate results using the standard wall functions.
- The user-specified value of y^+ should be less than approximately 150. A value of 100 has been shown to work well.
- It is recommended that temperature-dependent fluid properties be used to ensure maximum accuracy.

  If you use temperature-dependent fluid properties, all fluid properties should be temperature-dependent, rather than just some of the properties.

Virtual Local Heat Transfer Coefficient

When you use this option, note the following:

- It is recommended that you follow all recommendations for getting accurate results using the SWFs.
- This option is useful when you are not interested in the temperature distribution in the fluid and it is easy to define a reasonable global reference temperature. For example, in an external/internal flow where the temperature of the fluid is not expected to change much due to surface heat flux.

  For external flows where there is a well-defined free-stream temperature outside the thermal boundary layer, the usual practice is to use this temperature as the reference temperature. For internal flows without axial recirculation, it is often practical to define the reference temperature as the local bulk temperature that is obtained by solving the one-dimensional (stream-wise) energy equation. If the bulk temperature varies little relative to the overall fluid-wall temperature difference, you can set the reference temperature to the inlet temperature.
- The y^+ of the near-wall cell should be in the inertial sublayer $30 < y^+ < 150$ because it is more likely that the user specified reference temperature is close to what is used in the SWFs to calculate h.
- This option should not be used for laminar flows because calculating the surface heat flux accurately requires the near-wall cell to lie in the viscous sublayer. However, as mentioned above, having the near-wall cell in the viscous sublayer is not recommended.
- This option is not recommended for coupled use with other applications.

Setting an Environment Boundary Condition for Heat Transfer

The main purpose of this feature is to permit the modeling of both convection and radiation heat transfer between the external sides of domain boundaries and the environment.

This modeling is achieved through a certain type of thermal boundary condition (or more specifically, an option for Thermal Specification) called the Environment condition (in contrast to a Temperature, Flux, or Convection condition).

This condition is analogous to a Baffle Interface where the physics continuum on one side of the baffle is replaced with the environment. Similar to a baffle interface, the Environment condition can have radiative transfer to/from both surface sides and distinct surface temperatures for each side. In contrast to the baffle interface, the Environment condition is applied using a single boundary (using the Thermal Specification option). This single boundary behaves like a dual-sided boundary when the condition is activated. In contrast, the baffle interface is applied using two single-sided boundaries.

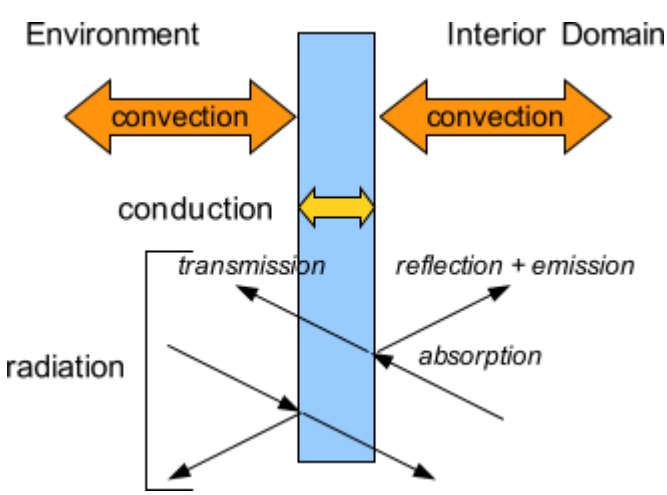
Modeling external radiation transfer with the environment condition is only available with the Surface-to-Surface radiation transfer model.

Contents:

Understanding the Physics and Modeling
Setting the Environment Condition

Understanding the Physics and Modeling

The physics that is being considered is shown in the following diagram.



Specifically, the physics of interest consists of:

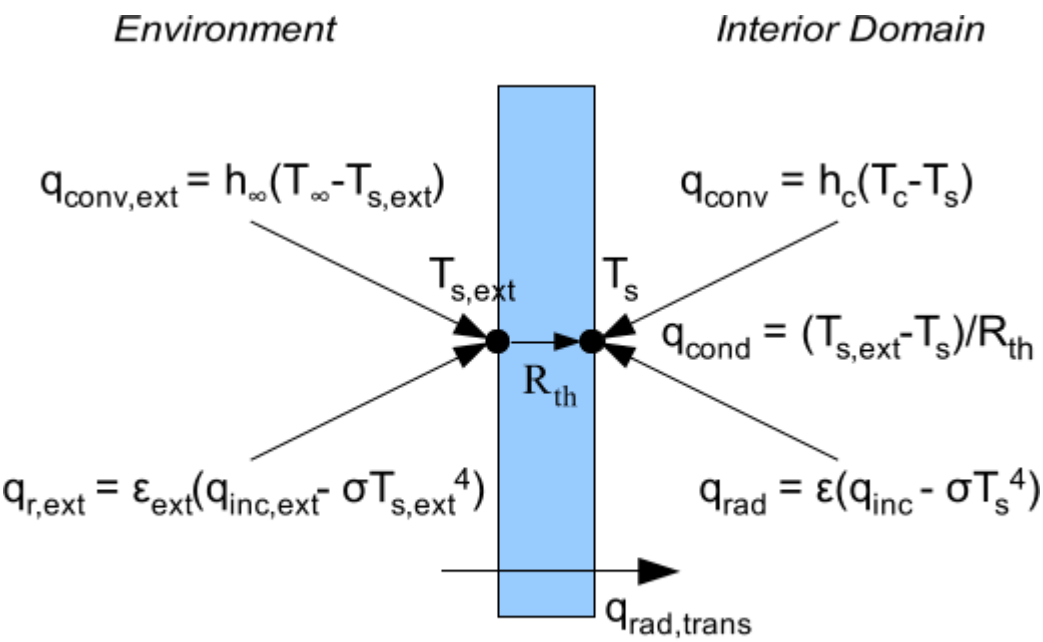
- Convection and radiation from the internal domain to the internal side of the boundary. The internal surface can absorb, emit, reflect, and transmit radiation.
- Conduction through the boundary. The boundary can have any finite thermal resistance, including zero.

- Convection and radiation from the environment to the external side of the boundary. The external surface can absorb, emit, reflect, and transmit radiation.

The modeling of this physics is depicted in the following diagram. In contrast to the existing Convection thermal condition, the Environment thermal condition additionally considers the following:

- Absorption and emission of radiation on the external side.
- A non-zero wall thermal resistance.
- An external surface temperature distinct from the internal surface temperature. This distinction is the result of allowing from a non-zero wall thermal resistance.

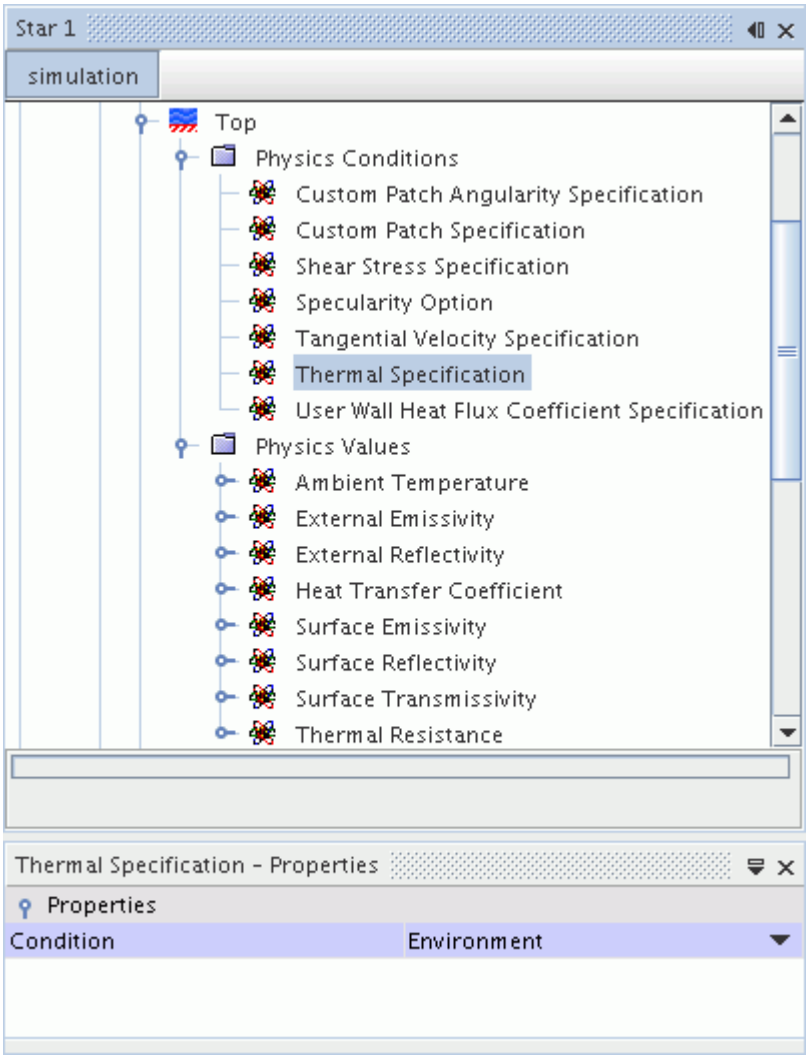
The last two items are not necessarily unique to the Environment condition (that is, they could potentially be applied for the other thermal conditions). Currently, however, the non-zero thermal resistance is only available with the Environment condition. Thermal heat capacity is not considered with the Environment condition. If the wall has sufficient thermal capacity and an unsteady analysis is being considered, then the wall can be modeled using a thin solid. An Environment condition can optionally be applied to the external boundary of the solid.



$$q_{inc,ext} = q_{inc,sol} + q_{inc,env} + q_{inc,other\ bodies}$$

Setting the Environment Condition

The feature is activated on a per-boundary basis by changing the **Thermal Specification** option for the given boundaries.



Specifically, the steps are:

1. Activate the Environment condition:

   Select the **Physics Conditions** > **Thermal Specification** node for the boundary and set the *Condition* property to **Environment**.

2. Specify boundary values under the **Physics Values** node:
   - Specify the External Emissivity and External Reflectivity for the outward facing side.

     These settings are in addition to the standard Surface Emissivity and Surface Reflectivity for the inward-facing side. The existing Surface Transmissivity serves as the transmissivity for both sides (it must be symmetrical to satisfy the second law, just like at interfaces).
   - Specify the Heat Transfer Coefficient and Ambient Temperature for convection from the external side to the environment.

     These settings are the same profiles that are currently used for the Convection condition.
   - Specify the Thermal Resistance for the wall.

The resistance can be specified as a total wall resistance or as a multi-layer resistance. The multi-layer resistance is specified using the Multi-Layer Resistance method on the Thermal Resistance profile. This setting is similar to the multi-layer thermal resistance for baffles.

3. Specify radiation environment attributes in the radiation-specific models as usual.

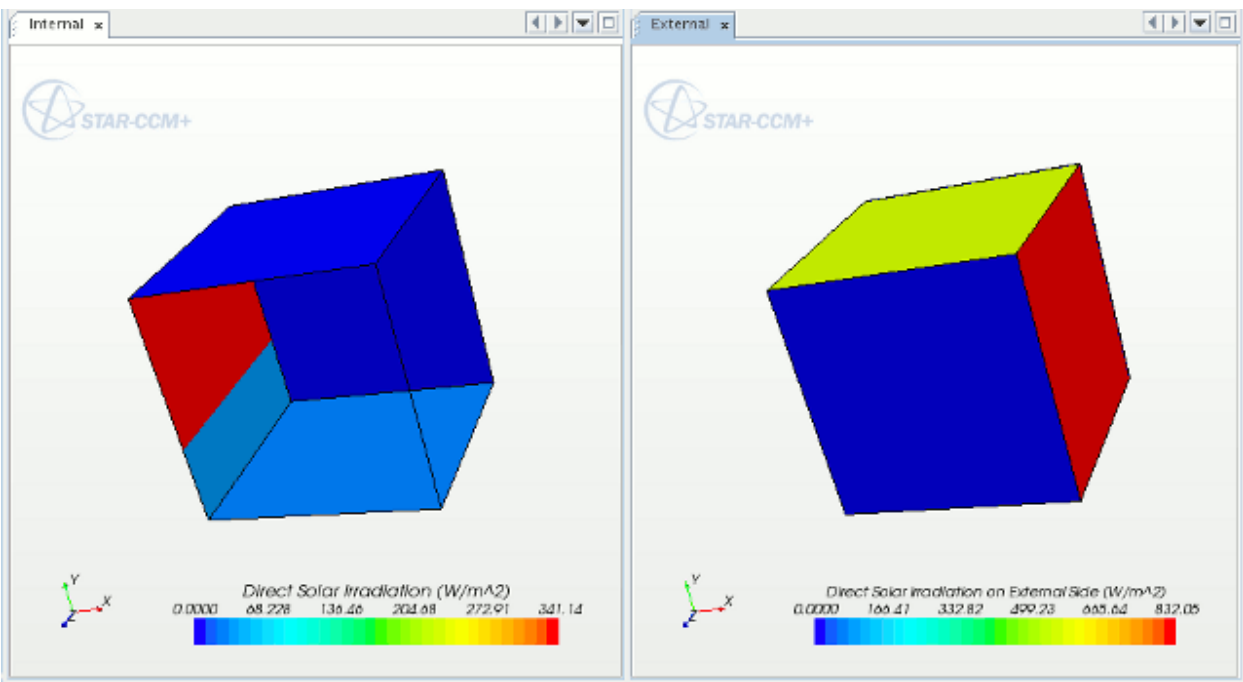
   The attributes include the environment radiation temperature and the solar loads. Unlike the per-boundary inputs, these attributes are global (for example, the specified solar loads are applied to all relevant boundaries). This step has not changed from previous usage, but the specified radiative environment conditions now interact with the external sides of boundaries that have the Environment condition activated.

4. Run the simulation.

5. Post-process the results.

   The field functions for the inward facing side have the standard names/titles (such as Temperature or Boundary Irradiation).

   The field functions for the outward facing side have names/titles that reflect the “External” namespace. The field function name is prefixed with “External” (such as ExternalTemperature). The field function titles are suffixed with “on External Side” (for example Temperature on External Side). Typical results are shown in the following screenshot, an example of a field function display for internal and external sides of boundaries.



Contents:

Using the Environment Condition with S2S External Ray Tracing

Using the Environment Condition with S2S External Ray Tracing

The external sides of the boundaries with the Environment condition activated are patched, and view factors are computed for these patches as with any other patches.

Often the external sides only see the environment and thus have a single unity view factor to the environment. However, if the external side of the boundary “sees” any other boundaries that also have an environment condition (or have an outward S2S surface orientation), then the boundary has view factors to these other boundaries.

Limitations and Restrictions

- External radiation modeling is only available for S2S (not currently available for DOM).
- External surfaces with the Environment condition activated must only see other boundaries that have either of the following:
  - Have the Environment condition activated
  - Are attached to an opaque solid region that has its S2S surface orientation that is set to Outward.

  This restriction is similar to that governing the outward S2S surface orientation.
- The Environment condition must not be applied to boundaries of regions that have an outward S2S surface orientation. The Environment condition provides an outward facing surface (the primary role of the outward orientation).
- The Environment condition is not currently available for flow openings such as inlets and outlets.

Example: Net Radiative Transmission Calculation on External Transmissive Boundaries

With the environment condition, there is an easy way to calculate the net radiative transmission on external transmissive boundaries. The following information illustrates the complete energy balance for a simulation with transparent external boundaries. It involves a window with convection from the outside: `hinf=10W/m2-K, Tinf=350`.

For this case with the environment condition, the net transmission in the window balances the heat transfer at all the boundaries (to about five digits relative to rate at the inlet):

```
        Surface Integral of Net Transmission on Volume Mesh
Part                                   Value (W)
------------------------------         -------------
Body 2: Window                          5.895503e-01
-------------
                        Total:          5.895503e-01

        Heat Transfer of Boundary Heat Transfer on Volume Mesh
Part                                   Value (W)
------------------------------         -------------
Body 2: Box                             0.000000e+00
Body 2: Inlet                          -4.314370e+02
Body 2: Outlet                          4.323347e+02
Body 2: Side                            0.000000e+00
```

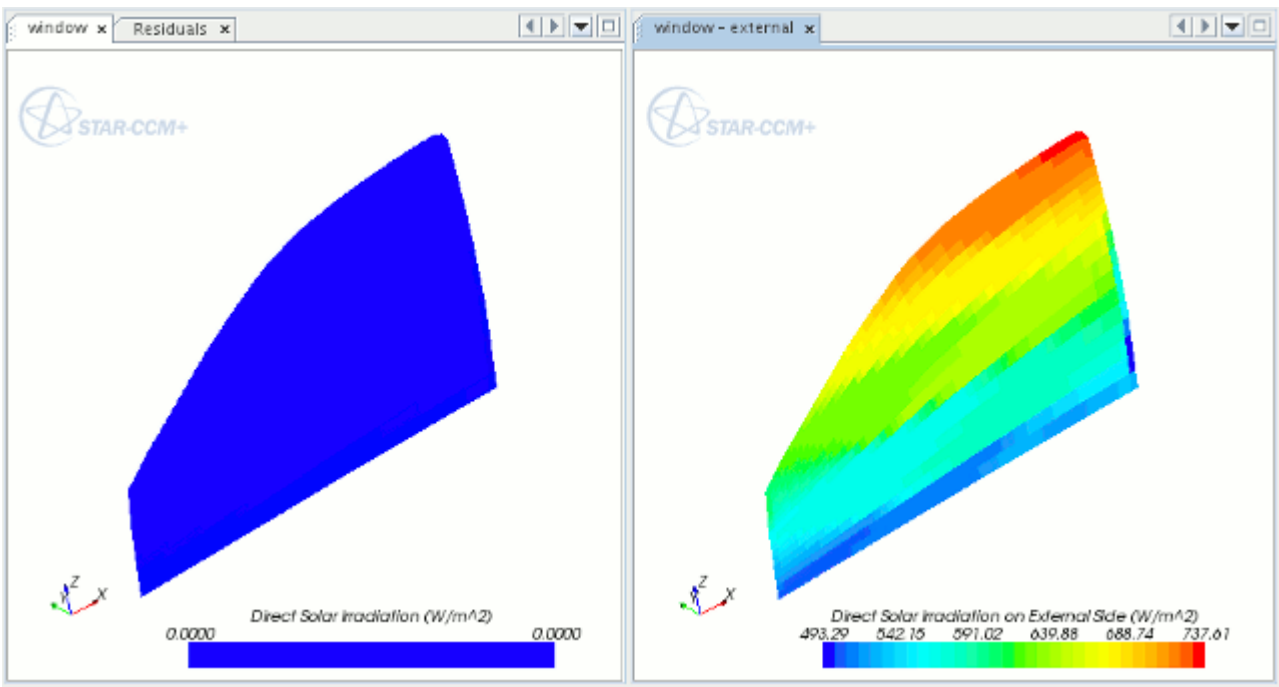
```
Body 2: Tubes                              0.000000e+00
Body 2: Window                            -3.105699e-01
                                          -------------
                            Total:         5.871046e-01
```

The transmission was computed using a user field function [`$BoundaryTransmissivity*($ExternalBoundaryIrradiation - $BoundaryIrradiation)`] that has been surface integrated over the transmissive boundary (that is, the window).

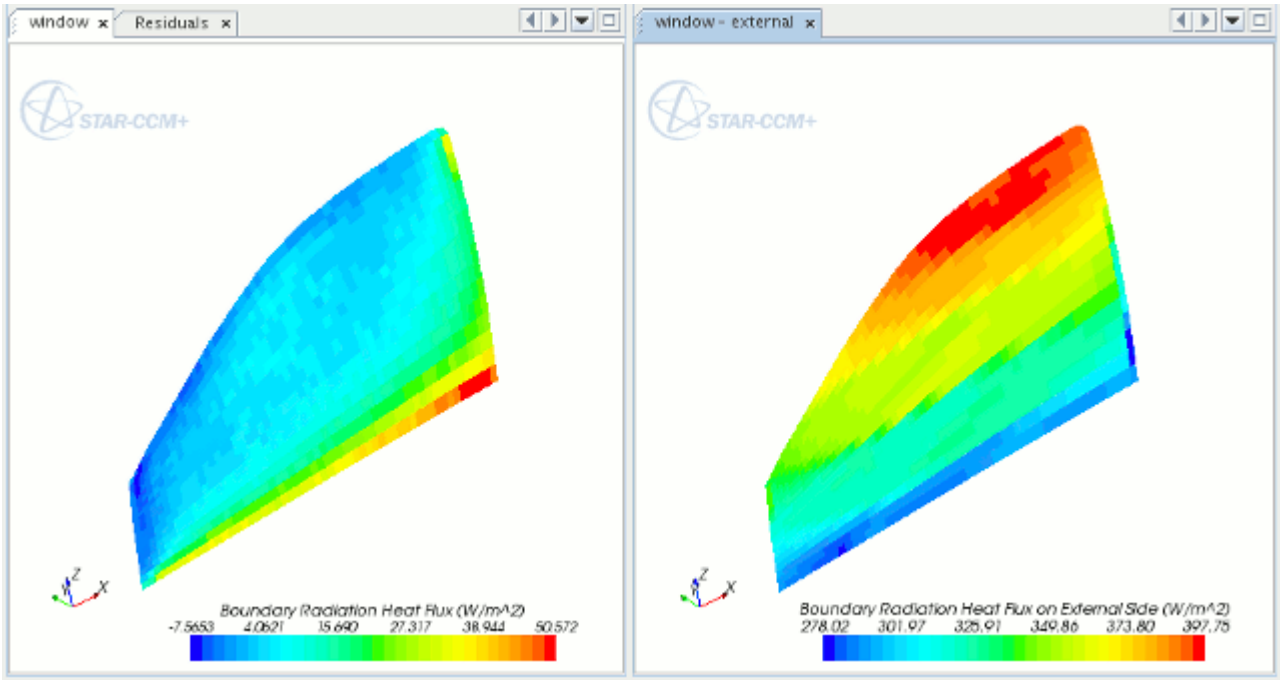
Example: Results for an Auto Cabin

The feature is applied to a thermal analysis of an auto cabin. Specifically, the Environment condition is applied to the boundaries representing the windows of the car. The windows are partially transparent, allowing both the solar radiation and the environment radiation from the outside to be absorbed on the external side and to be transmitted into the cabin interior. In addition, the windows have finite thermal resistance to conduction.

The following four illustrations display various field functions for the internal and external sides of the driver side window—the internal side being displayed on the left and the external on the right. The following screenshot displays the direct solar irradiation on internal and external sides of the driver window. As expected, the flux on the internal side is zero, and the flux on the external size is finite. The highest flux is near the top as the window bends further from vertical.

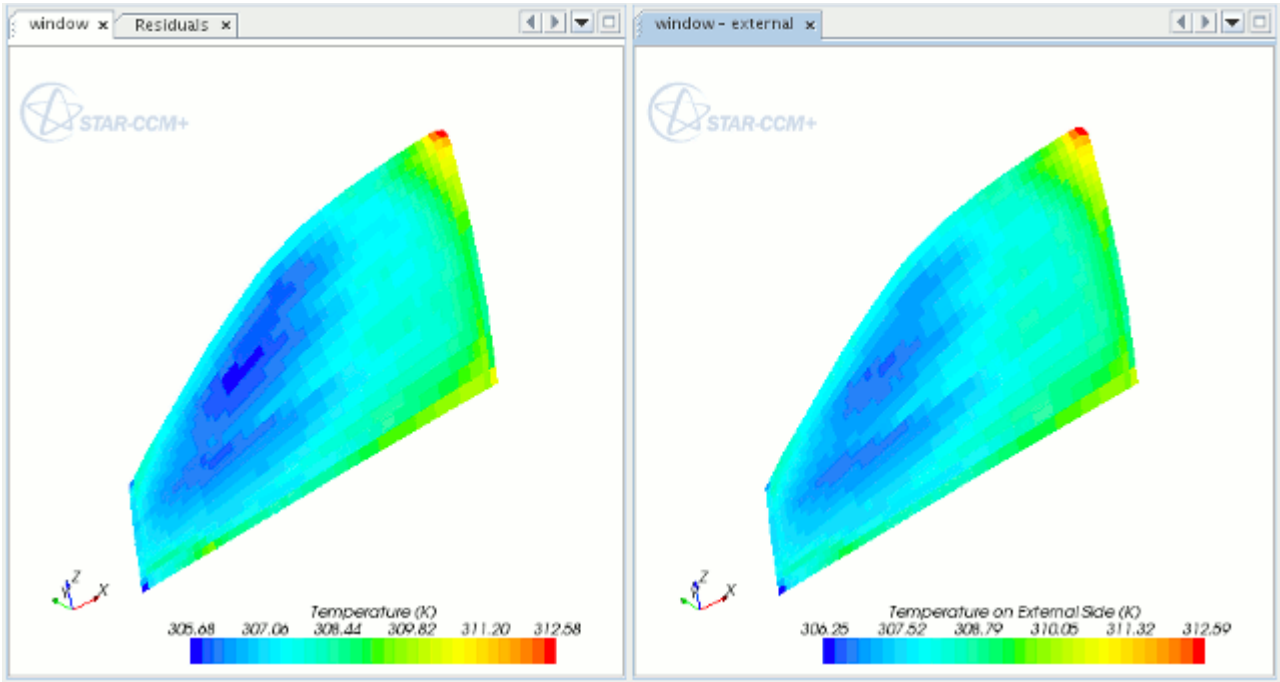


The following screenshot shows the net radiative heat flux on both sides.



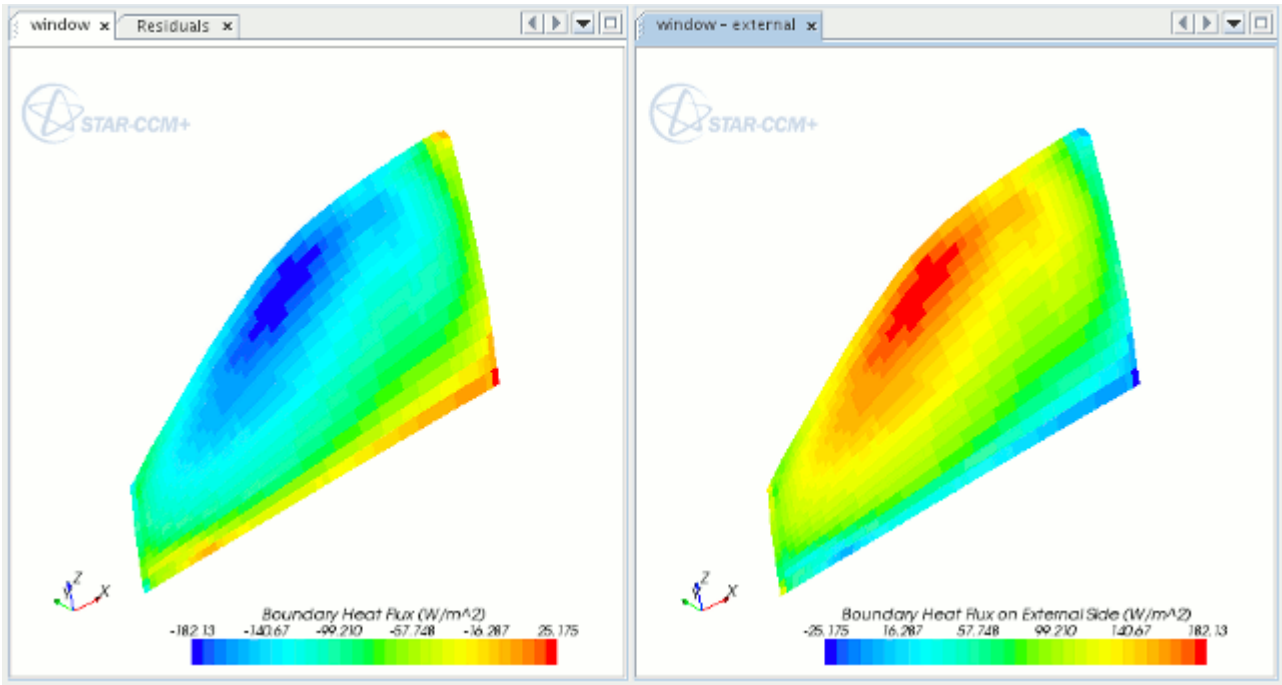
The flux on the external side is highly influenced by the solar flux (both direct and diffuse). The flux is relatively high over the entire glass. In contrast, the net radiative flux on the internal side is much less. The internal side largely interacts with the interior of the cabin which has temperatures similar in scale to the window temperature. The higher fluxes on the bottom of the internal side are due to solar radiation coming through the window. This radiation hits the horizontally oriented part of the door just below the window, and is then diffusely reflected back into the internal side of the window.

The following screenshot shows the temperature on the internal and external sides.



The window has a thermal resistance so the temperatures can be different. However, since the resistance is relatively small, the temperatures end up being similar, with the external temperatures slightly higher than the internal ones.

Lastly, this next screenshot displays the net boundary heat fluxes for the internal and external sides.



In order to conserve energy at the window, these fluxes must be equal and of opposite sign. As seen in the figure, this behavior is realized. The external side generally has a net heat flux to the window from the environment. The internal side generally has a net heat flux from the window to the cabin interior. The fluxes on the opposite sides balance each other to machine precision.

Modeling Boiling

Some flows are characterized by subcooled nucleate boiling on solid surfaces, where the creation of the vapor bubbles is insignificant for the flow.

In such situations, if you are interested in modeling the improved heat transfer due to boiling, you can add boiling to a STAR-CCM+ simulation with the flow of a single-component liquid.

This section describes the following:

- Selecting a boiling model
- Setting boiling model parameters

For information about the formulation that is implemented in STAR-CCM+, see Single-Phase Boiling Formulation.

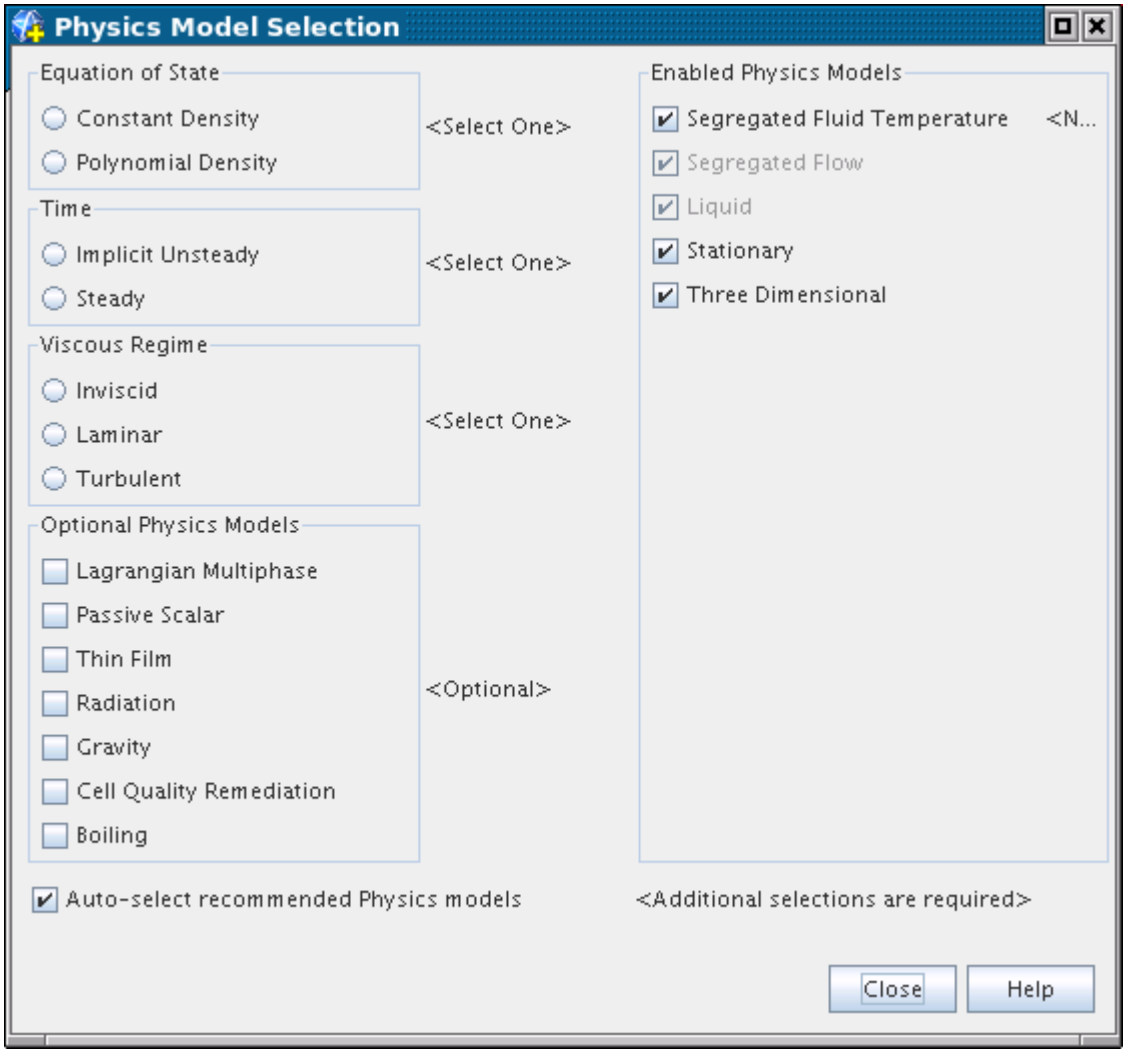
Contents:

Selecting a Boiling Model
Setting Boiling Model Parameters

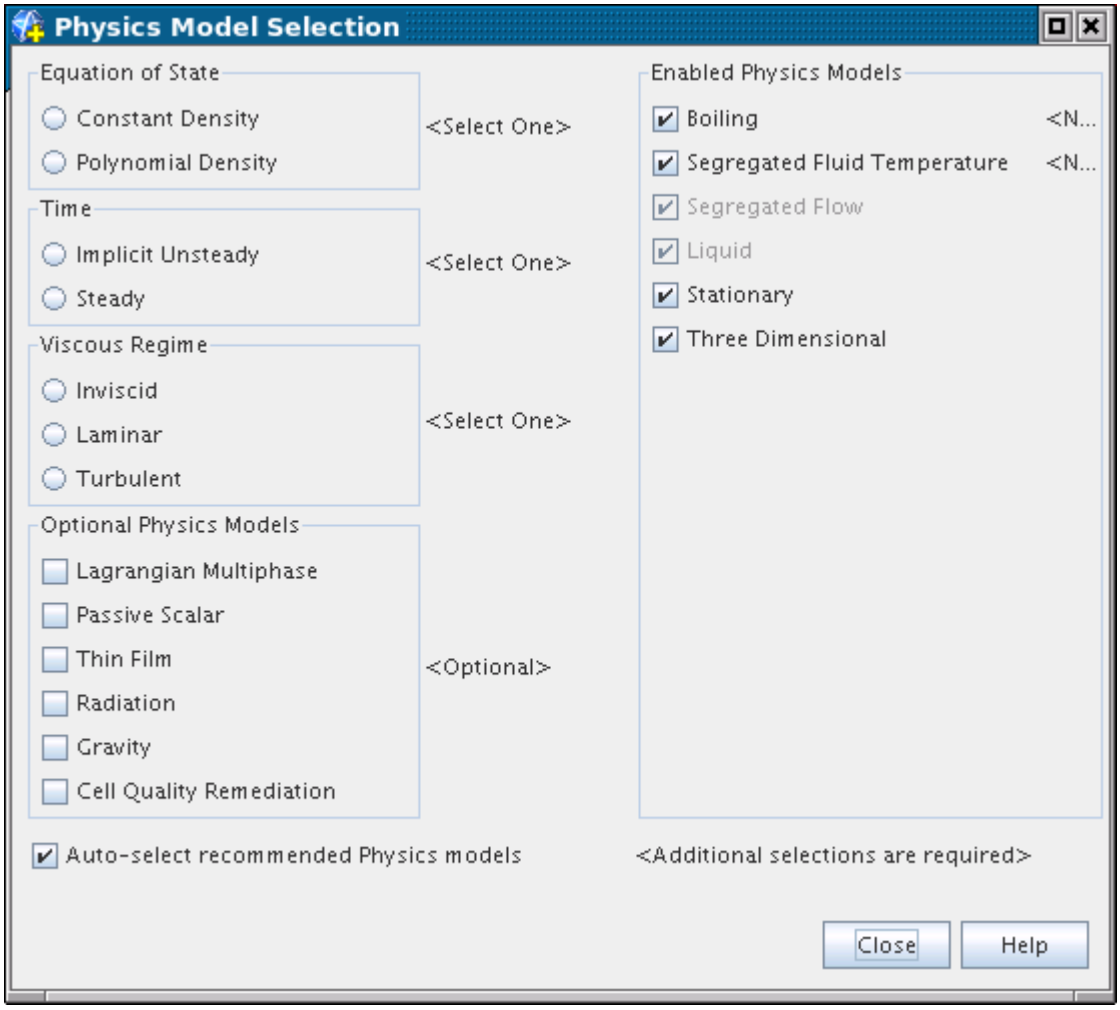
Selecting a Boiling Model

To select the boiling model that is based on Rohsenow's expression for the nucleate boiling:

1. In the *Physics Model Selection* dialog, specify the material model as a single-component liquid by selecting **Liquid** in the *Material* group.
2. Make appropriate selections of flow and energy models. The Boiling model becomes available in the *Optional Models* box.

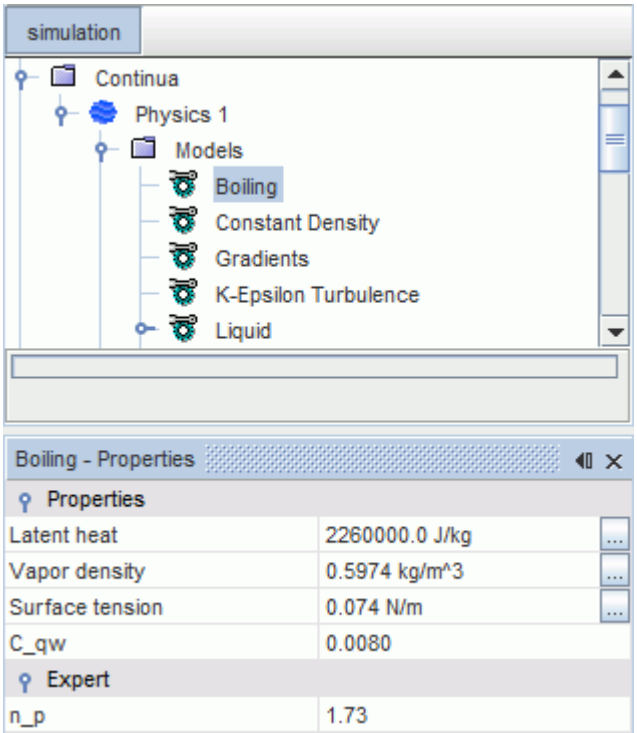


3. When you select **Boiling** the *Physics Model Selection* dialog shows the model as activated, with other model options remaining to be selected.



Setting Boiling Model Parameters

When a simulation with boiling is selected, STAR-CCM+ creates a node that is called **Boiling** in the simulation tree, as shown in the following screenshot.



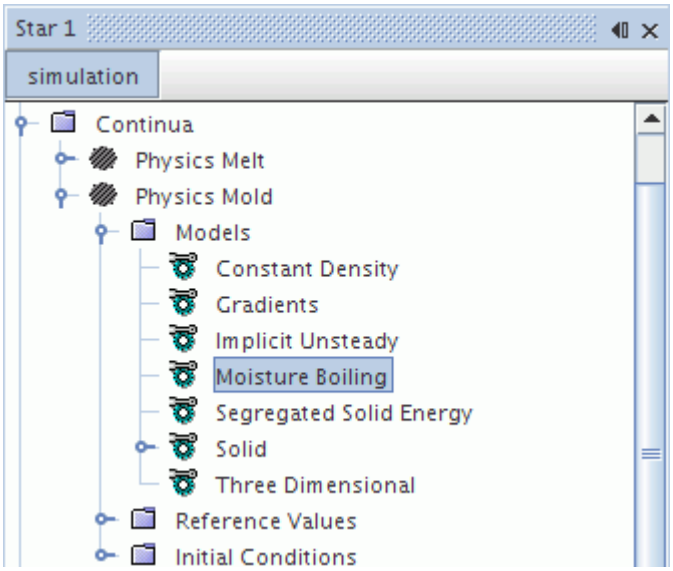
In the *Properties* window, specify the boiling model parameters:

Boiling Properties

Latent heat	The latent heat of evaporation.
Vapor density	The density of vapor.
Surface tension	The surface tension coefficient between the liquid and its vapor.
C_qw	The empirical coefficient C_{qw} in the Rohsenow expression for the wall heat flux, Eqn. (313). This value varies with the liquid-surface combination.
n_p	The Prandtl number exponent n_p (1.73 by default) in Eqn. (313).

Modeling Moisture Boiling in Solids

When a wet solid substance is heated above the boiling temperature of the moisture, it can become a significant heat sink due to the latent heat of vaporization. The Moisture Boiling model in STAR-CCM+ simulates this heat transfer effect.



This model computes the boiling of a liquid (moisture) within a solid material. The model considers only energy transport -- it does not simulate the evaporated gas phase and distribution. However, this model does calculate distribution and volume fraction of moisture within the solid.

The moisture in the solid does not influence the material properties of the solid such as conductivity or density. The boiling process influences the temperature distribution within the solid. (Boiling is an isothermal process, so the temperature remains constant.)

Moisture Boiling Properties

Density	The density of the moisture.
Latent heat	The latent heat of vaporization of the moisture.
Boiling Temperature	The boiling temperature of the moisture.

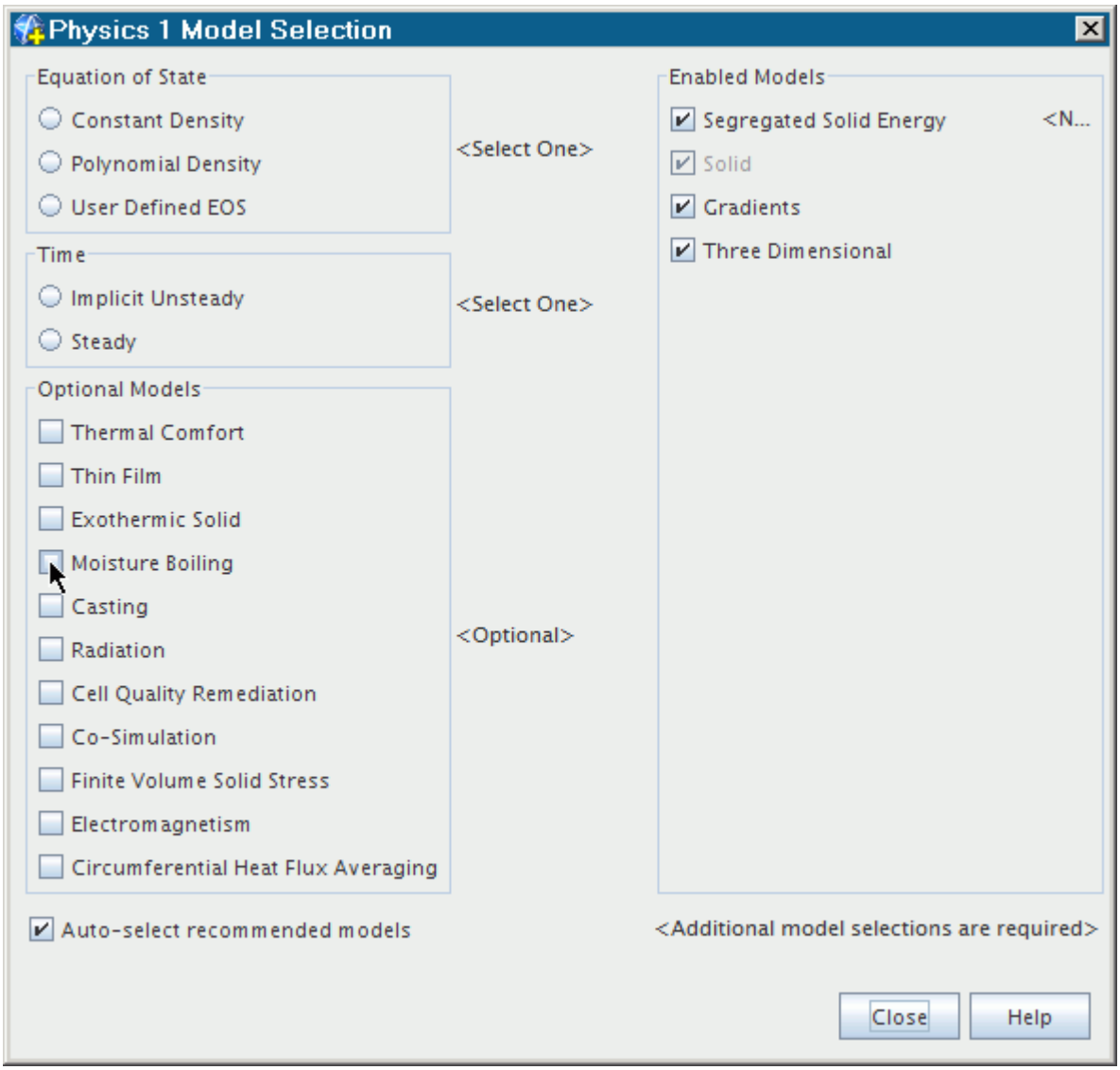
Contents:

Setting up a Moisture Boiling Simulation
Moisture Boiling Field Functions Reference

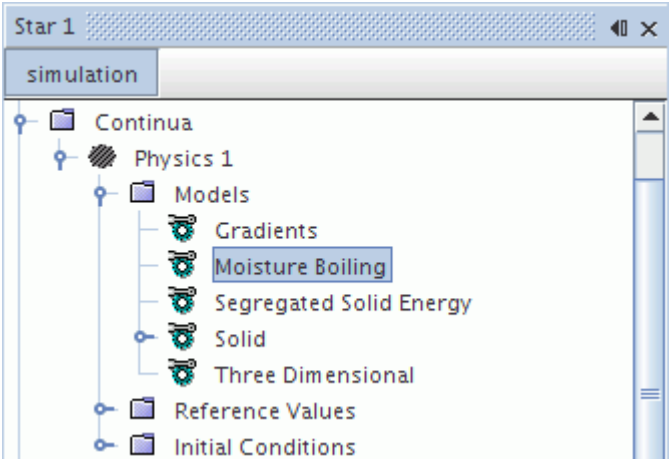
Setting up a Moisture Boiling Simulation

To use moisture boiling in a STAR-CCM+ simulation:

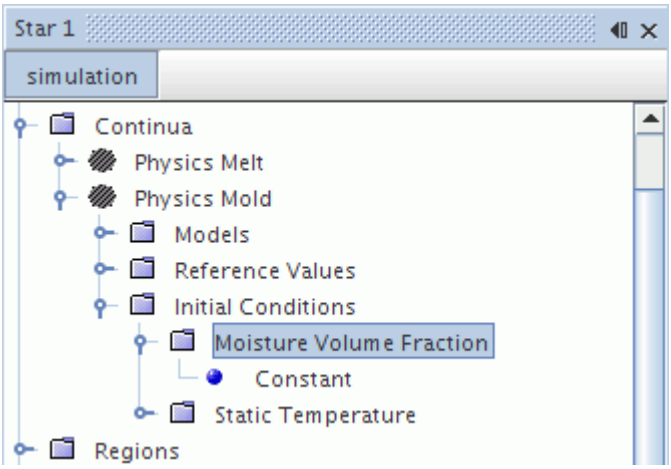
1. Select the Moisture Boiling model:
   a) Include the following models with your selections:
      - The Solid material model.
      - The Segregated Solid Energy model.
   b) Select the **Moisture Boiling** model from the *Optional Models* group box.



When you close the model selection dialog, the Moisture Boiling model is shown in the list under the **Continua** > **Physics** > **Models** node.



2. Specify the initial moisture volume fraction as an initial condition.



3. Set the static temperature which is also an initial condition.
4. Set the boiling temperature in the *Boiling Temperature* property of the **Moisture Boiling** model node.

   The temperature values have the following effects:

   - When the initial temperature is greater than the boiling temperature, the phase change (boiling) of moisture results in a temperature decrease down to boiling temperature. This process requires the volume fraction of moisture to exceed zero.
   - When the initial temperature is less than boiling temperature, and if the cell temperature increases due to a net heat flux into the cell, temperature first increases up to the boiling temperature. Heat is consumed to vaporize moisture isothermally at the boiling temperature. When the moisture volume fraction reaches zero, the temperature increases further.

Moisture Boiling Field Functions Reference

The following primitive field function is made available to the simulation when the Moisture Boiling model is used.

It has a typical set of base properties.

Moisture Volume Fraction	The fraction of volume that the moisture represents.

Modeling Thin Film: Deicing and Defogging

The Thin Film model provides a simplified approach for simulating deicing or defogging inside a cabin windshield or other surface. It addresses cases where the only features of interest about the ice or fog layer are where the layer is distributed and how thick it is. It assumes that the thickness is negligible compared to cell size.

For an example where the changing layer thickness is geometrically and dynamically significant, see the tutorial Dispersed Multiphase: Airfoil Icing. This tutorial describes ice accumulation on an airfoil, which changes shape as the ice layer grows.

This section covers the following:

- The Thin Film model
- The Deicing and Defogging models
- The Deicing and Defogging solvers

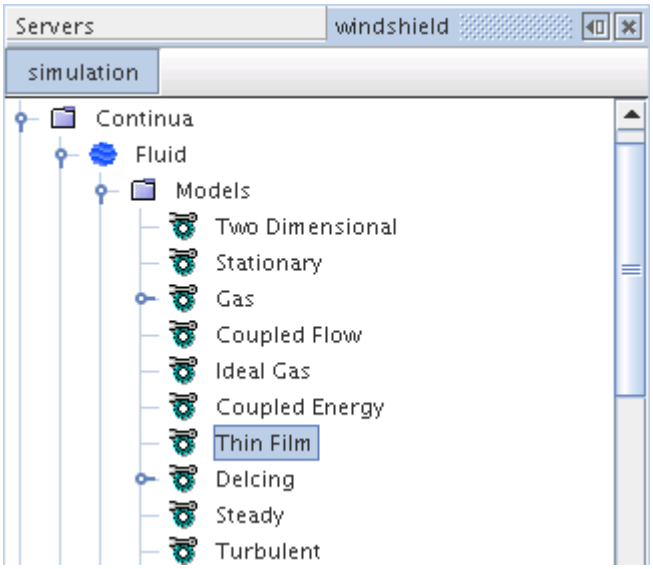
Also included is the following reference information:

- Boundary conditions
- Boundary values and initial conditions
- Field functions that this model activates

The Thin Film Model

The Thin Film model provides a simplified approach for simulating deicing or defogging inside a cabin windshield or other surface.

The object has no properties of its own, but is subject to common menu items.



The limitations on this model are as follows:

Thin Film Model

- Once a boundary is switched from deactivated to having a fog or ice layer, the simulation must be run in transient mode.
- It is possible to have Deicing and Defogging in the same region, but not on the same boundary.

Deicing Model

- You cannot use a contact boundary. Deicing can only be applied to wall boundaries and the type must be convection or heat flux.

Defogging Model

- Defogging can only be applied to wall and contact interfaces (solid/fluid).
- Currently the saturation vapor pressure curve is based on water vapor.

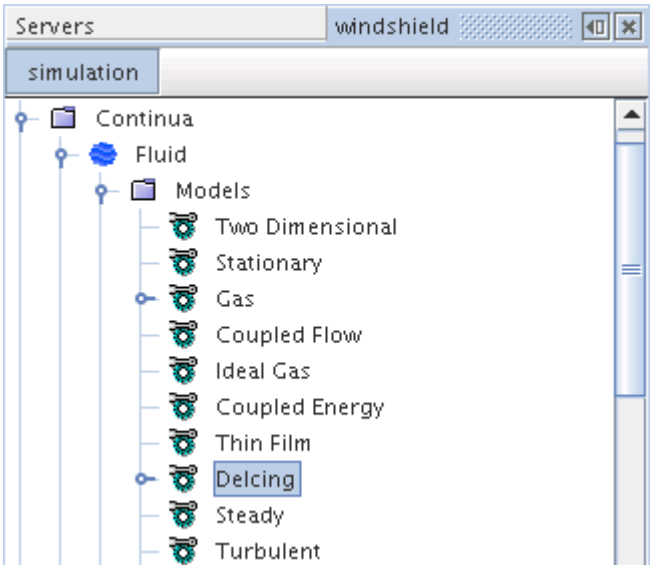
Contents:

What Is the Deicing Model?
What Is the Defogging Model?
Selecting the Thin Film Model
What Is the Deicing Solver?
What Is the Defogging Solver?
Boundary Conditions Reference
Boundary Values and Initial Conditions Reference
Thin Film Field Functions Reference

What Is the Deicing Model?

The Deicing model, which is made available by the Thin Film model, is a simplified approach to obtain deicing on a windshield or other surface.

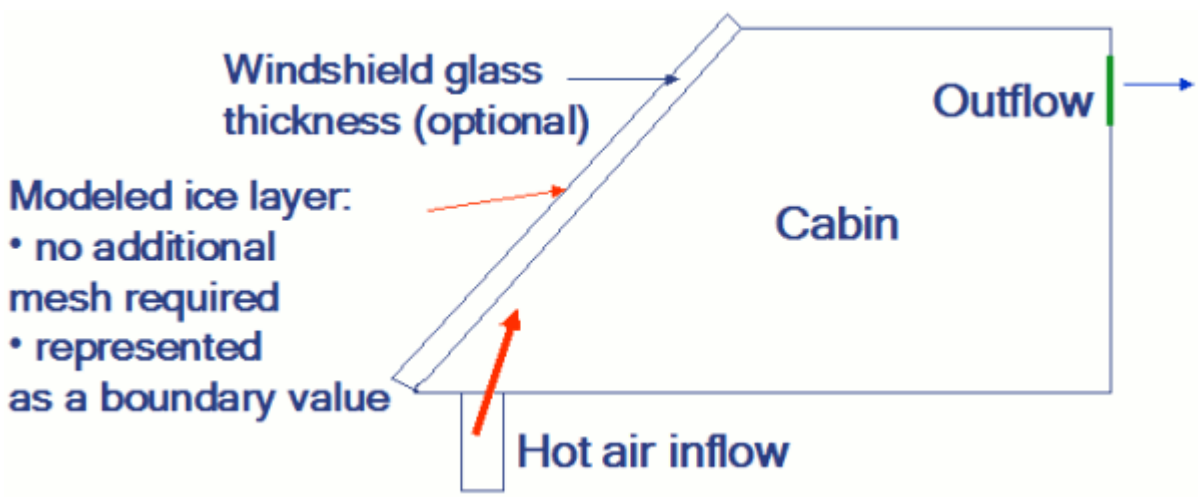
The model object has an expert property and is subject to common menu items.



This object also contains the Ice material, which lets you work with material properties for this film.

Major hypotheses are:

- The ice layer is not represented as a phase in the volume mesh, but is modeled using functions, thus the ice layer does not flow.
- The model formulation implicitly assumes that the temperature surrounding the ice is above the freezing temperature. If this assumption is not correct, the model could produce some non-physical values (for example, the ice could start melting at temperatures which are below the freezing temperature).
- A steady-state calculation is performed to initialize the convection flow field.
- The best method is convection, which you would select for the Thermal Specification boundary condition.
- During the transient run, it is recommended that you leave all equations on. However, you can use a large time-step because the melting of the ice layer does not affect the flow field.



Deicing Model Properties

None

Deicing Model Expert Properties

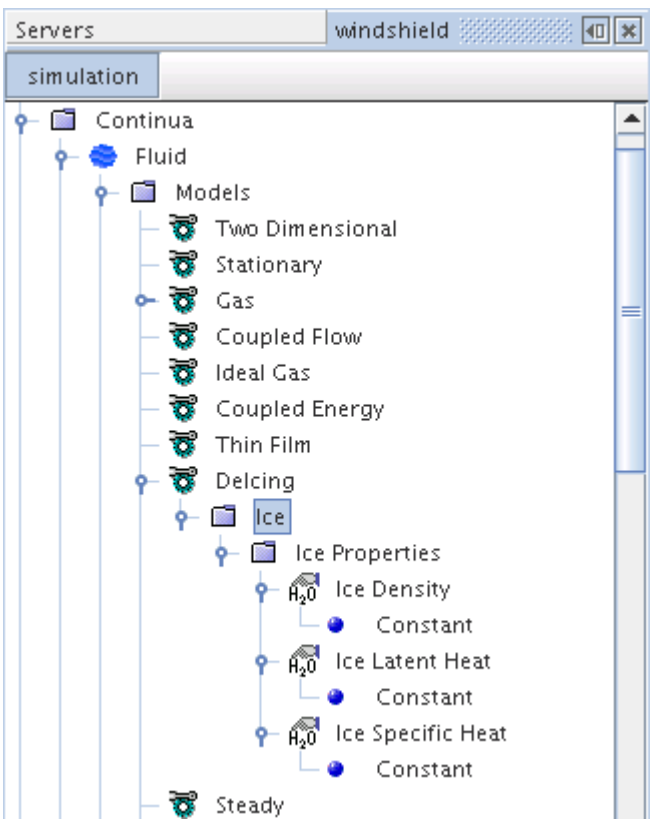
Cemp	Specifies the C_{emp} empirical coefficient in Eqn. (324). This coefficient specifies the proportion of heat flux that is absorbed at the ice-gas interface.

Contents:

What Is the Ice Material?

What Is the Ice Material?

This object has a sub-node that works like the material properties manager: it contains material properties objects that are specific to this type of thin film.



Ice Material Properties

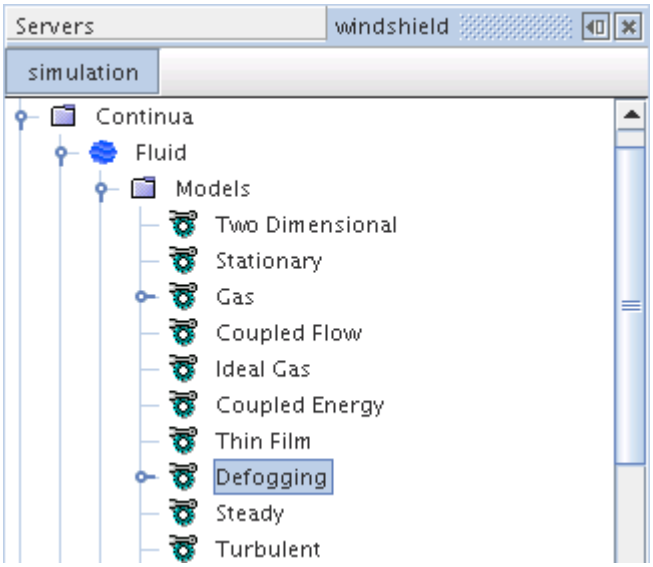
None

See also: common menu items.

What Is the Defogging Model?

The Defogging model, which is made available by the Thin Film model, is a simplified approach to obtaining a defogging simulation inside a cabin windshield or surface.

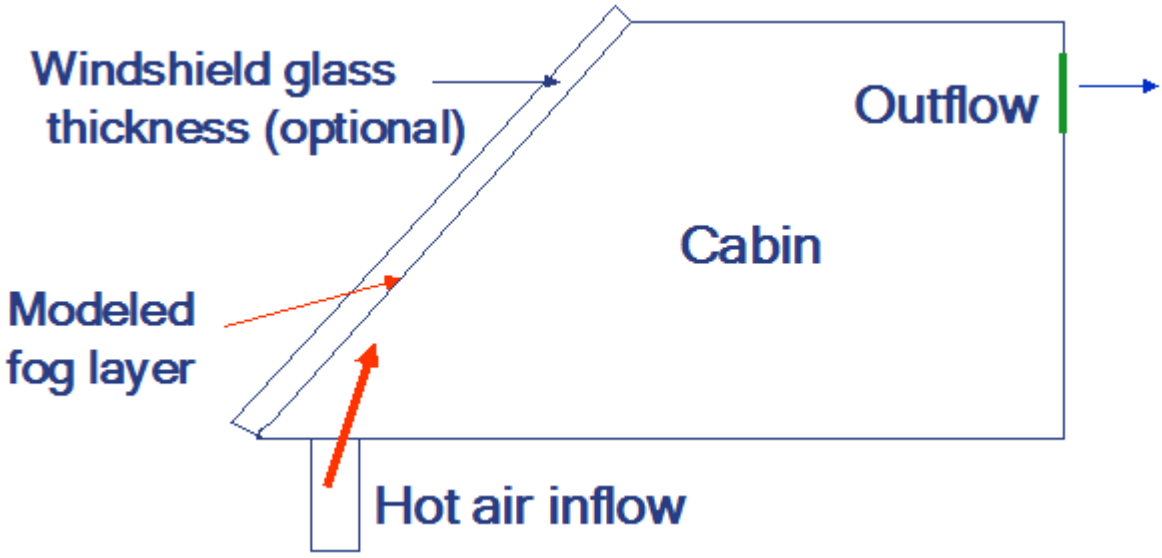
The model object can be controlled with properties and is subject to common menu items.



This object also contains the film material, which lets you work with material properties for this film.

Major hypotheses are:

- The fog layer is not represented as a phase in the inner volume mesh, but is modeled using functions, thus the fog layer does not flow.
- A steady state calculation is performed to initialize the convection flow field.
- During the transient run, it is recommended that you leave all equations on. However, you can use a large time-step because the condensation/evaporation of the fog layer does not affect the flow field.



Contents:

Defogging Model Properties
What Is the Film Material?

Defogging Model Properties

LimitSuperSaturation	Controls whether supersaturation (humidity exceeding 100%) is prevented.	
	Activated	Prevents supersaturation.
	Deactivated	Allows supersaturation.

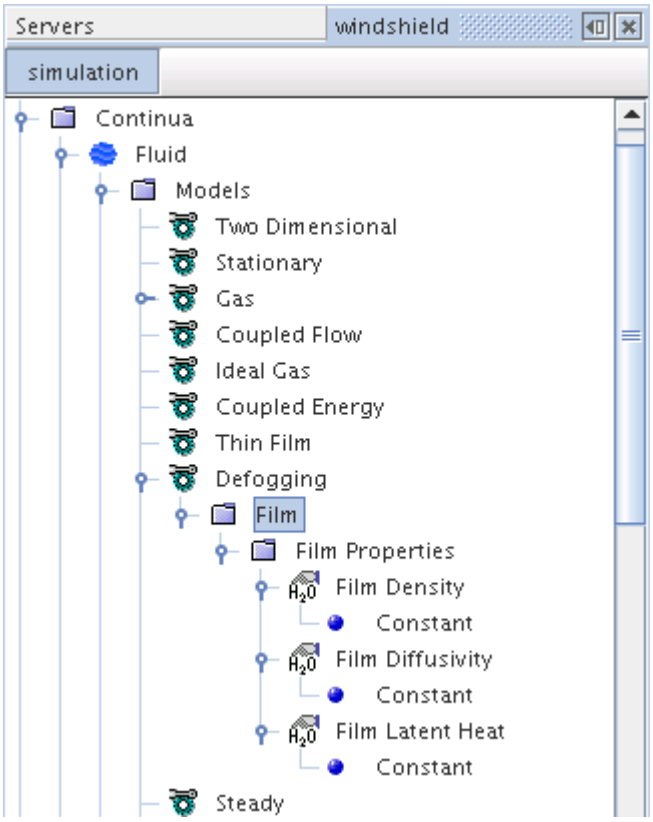
Defogging Model Expert Properties

The settings within the *Expert* category can be useful if you are thoroughly familiar with the theoretical aspects of this model and the discretization techniques that are used in STAR-CCM+. The values in that category reflect both the design of the model and the discretization approaches that have been optimized for accuracy and performance. If you are not familiar with these details, do not change any settings.

Secondary Gradients	Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces.	
	On	Include both secondary gradients.
	Off	Exclude both secondary gradients.
	Interior Only	Include the interior secondary gradients only.
	Boundaries Only	Include the boundary secondary gradients only.
Convection	Selects the convection scheme that is used. For guidance as to convection scheme selection, see Diffusion Term, p.3755.	
	1st-order	Selects the first-order upwind convection scheme.
	2nd-order	Selects the second-order upwind convection scheme.
Cemp	Specifies the C empirical coefficient.	

What Is the Film Material?

This object has a sub-node that works like the material properties manager: it contains material properties objects that are specific to this type of thin film.



Film Material Properties

None

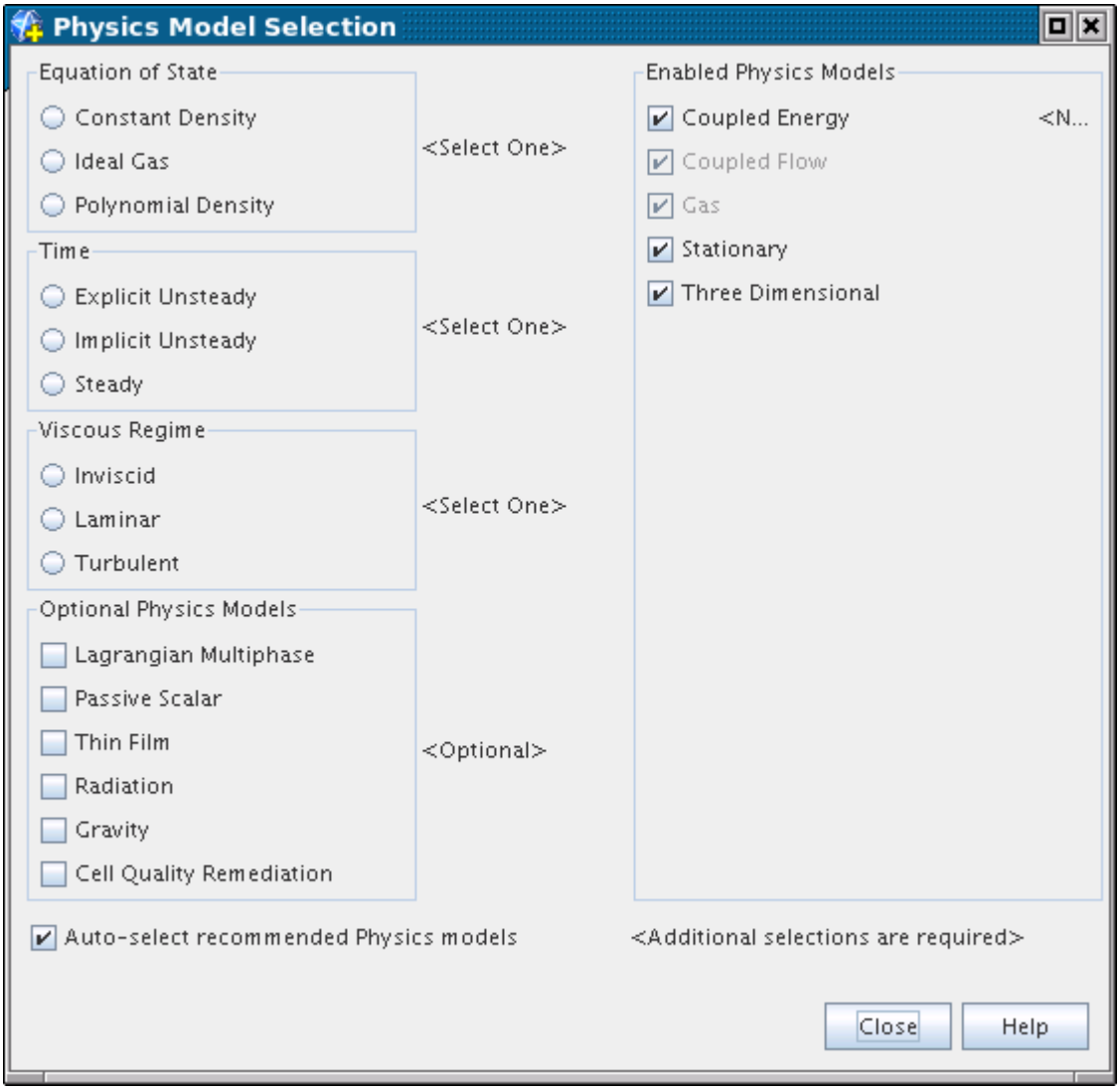
See also: common menu items.

Selecting the Thin Film Model

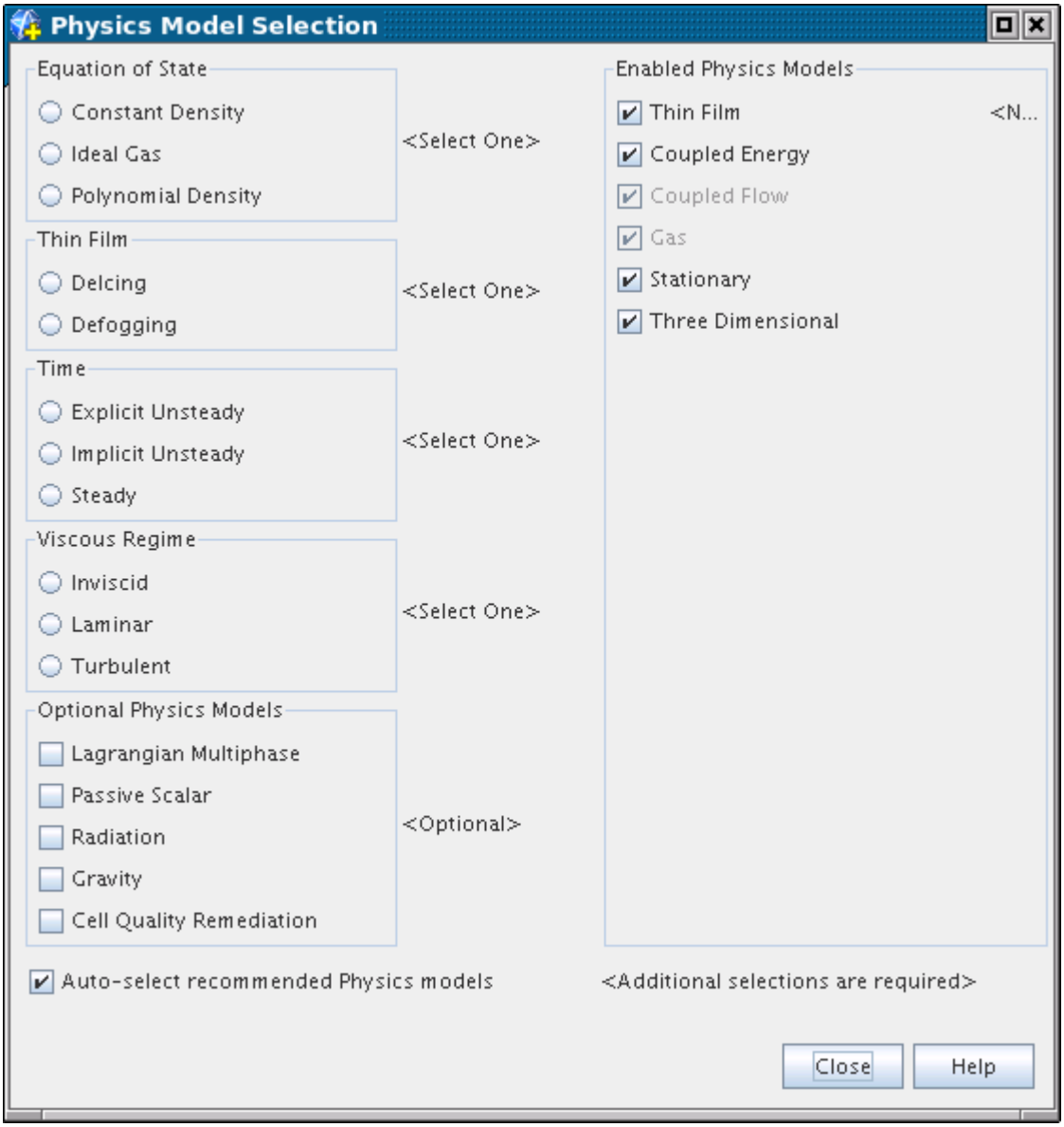
To select the Thin Film model:

1. In the *Physics Model Selection* dialog, specify a material model in the *Material* group.
2. Make appropriate selections of flow or solid models, and an energy model, depending on whether you are using the Deicing or Defogging model. If you are using Deicing, you can apply it to a fluid or a solid region. However, defogging requires a flow model for a fluid region.

The Thin Film model becomes available in the *Optional Models* group.



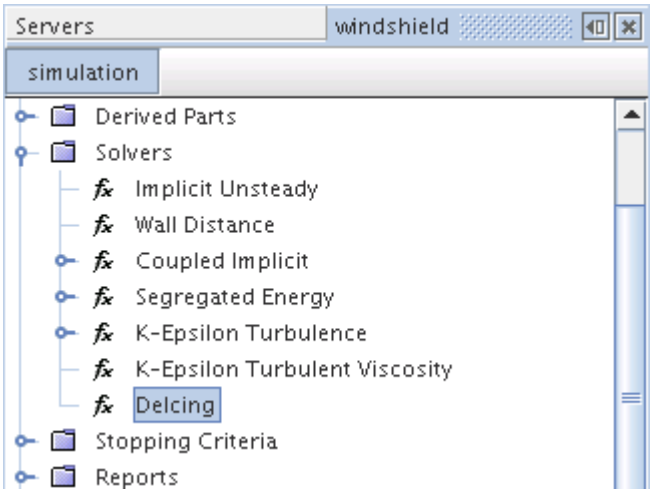
Once you activate the Thin Film model, the Deicing and Defogging models become available in the *Thin Film* group.



Select one of these models, depending on the type of film you are simulating.

What Is the Deicing Solver?

This solver, which is subject to common menu items, controls the solution of the Deicing model in all the continua for which the model is activated.



Deicing Solver Properties

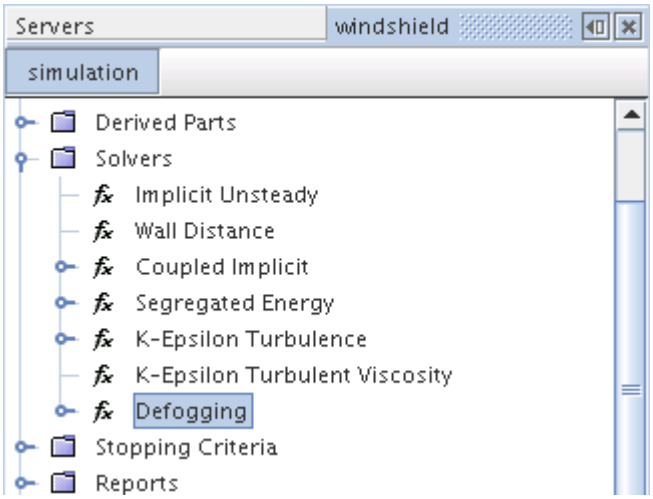
None

Deicing Solver Expert Properties

Solver Frozen	Prevents or allows the solver from updating the solution while iterating. This feature is primarily a “debugging” tool and is not supported for normal simulation work.	
	Activated	The solver does not update the solution.
	Deactivated	The solver updates the solution.

What Is the Defogging Solver?

This solver, which has properties and is subject to common menu items, controls the solution of the Defogging model in all the continua for which the model is activated.



Defogging Solver Properties

Under-Relaxation Factor	At each iteration, this property governs the extent to which the old solution is supplanted with the newly computed solution. The default value is 0.9.

Defogging Solver Expert Properties

The properties in the *Expert* category are designed for use in a temporary “debug” situation, at the expense of simulation accuracy and extra memory usage. It is not recommended that you change any of these properties unless you are thoroughly familiar with the discretization techniques used in STAR-CCM+.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Boundary Conditions Reference

This section details boundary conditions that are available for the Thin Film model.

Thin Film Option Properties

This node allows you to determine how the thin film across a wall boundary is specified. It is activated when the Thin Film model is activated.

Option	Activates defogging or deicing for the boundary.	
	Disable	The option is turned off for the boundary. No nodes are added to the **Physics Values** node.
	Fog Layer	This option, which is available when the Defogging model is activated, adds a Fog Layer Thickness node to the **Values** node, which must be entered as a scalar profile.
	Ice Layer	This option, which is available when the Deicing model is activated, adds an Ice Layer Thickness node to the **Values** node, which must be entered as a scalar profile.

Vapor Mass Fraction Option Properties

This node, which is activated when the Defogging model is activated, allows you to determine how the defogging across a wall boundary is specified.

Vapor Mass Fraction	Activates a fog layer at the boundary.	
	Zero Flux	No fog layer is present. No nodes are added to the **Physics Values** node.
	Film	Specifies saturation conditions are specified for the scalar when a fog layer is present.

Vapor Mass Fraction Specification Properties

This node, which is activated when the Defogging model is activated. It allows you to determine how the defogging is specified across the following types of boundaries:

- Free stream
- Mass flow inlet
- Velocity inlet

- Pressure outlet
- Stagnation inlet

This object is also an initial condition.

Method	The method that is used to define the defogging at the boundary.	
	Vapor Mass Fraction	This option adds a Vapor Mass Fraction node to the **Values** node, which must be entered as a scalar profile.
	Vapor Humidity	This option adds a Vapor Humidity node to the **Values** node, which must be entered as a scalar profile.

Boundary Values and Initial Conditions Reference

This section contains reference descriptions of boundary and initial condition values that are available for the Thin Film model.

The boundary values are set according to the boundary type, and by methods that are selected in corresponding nodes in the **Conditions** node. They are entered as scalar profiles.

Fog Layer Thickness

This value node is available when the Defogging model is selected. It is visible if **Fog Layer** is selected as the *Option* property for the **Thin Film Option** node of the **Conditions** node.

Ice Layer Thickness

This value node is available when the Deicing model is selected. It is visible if **Ice Layer** is selected as the *Option* property for the **Thin Film Option** node of the **Conditions** node.

Vapor Humidity

This value node is available when the Defogging model is selected. It is visible if **Vapor Humidity** is selected as the *Option* property for the **Vapor Mass Fraction Specification** node of the **Conditions** node.

Vapor Mass Fraction

This value node is available when the Defogging model is selected. It is visible if **Vapor Mass Fraction** is selected as the *Option* property for the **Vapor Mass Fraction Specification** node of the **Conditions** node.

Thin Film Field Functions Reference

The following field functions are made available to the simulation when the Thin Film model is used with the Defogging model.

Film Density	Density of liquid film [kg/m^3].
Film Diffusivity	The diffusion of vapor in air when the liquid film evaporates [m^2/s].
FilmLatentHeat	Latent heat due to condensation/evaporation of the film vapor [J/kg].
Fog Layer Thickness Value	Thickness of the film [m].
Vapor Humidity Value	Partial pressure of water vapor in a mixture to the saturated vapor pressure of the water at the mixture temperature [%].
Vapor Mass Fraction Diffusion Coefficient	Diffusion of vapor mass fraction in air [m/s^2].
Vapor Mass Fraction Value	Ratio of the vapor mass fraction to the total mass fraction for each cell [*mv*/*mt*].

The following field functions are made available to the simulation when the Thin Film model is used with the Deicing model.

Ice Density	Density of the ice layer [kg/m^3].
Ice Latent Heat	Latent heat due to melting/solidification of the ice layer [J/kg].
Ice Layer Thickness Value	Thickness of the ice layer [m].
Ice Specific Heat	Specific heat of the ice layer [J/kg-K].

Modeling Heat Exchangers

Heat Exchangers in STAR-CCM+ model the heat transfer between two fluid streams—a hot and a cold stream (such as in radiators and air coolers). This section describes how to use the single stream and dual stream heat exchanger options in STAR-CCM+.

In the single stream option, only one of the fluid streams (usually the cold stream) is modeled explicitly, while the second stream is assumed to have a uniform temperature that you specify. In the dual stream option, both the streams are modeled explicitly by defining two overlapping regions and creating a heat exchanger interface between them. In both the options, heat is added or removed from the participating cells as a source or sink term in the corresponding fluid energy equation. It is assumed in this section that both the streams exist only in a single phase (liquid or gas) and no phase change occurs inside the heat exchangers.

Before choosing any heat exchanger option, an energy model must have been selected as part of the physics continuum of the heat exchanger region.

Contents:

Using the Single Stream Heat Exchanger Option
Using the Dual Stream Heat Exchanger Option

Using the Single Stream Heat Exchanger Option

In a single stream heat exchanger, one stream is assumed to have a uniform temperature and the other stream is modeled by specifying the heat exchanger enthalpy source.

This section defines what is meant by the heat exchanger enthalpy source in STAR-CCM+, and describes how to activate this type of heat exchanger and specify its properties. The formulation of this heat exchanger is described Single-Stream Heat Exchanger Formulation.

The Heat Exchanger Enthalpy Source

The heat exchanger enthalpy source is a device which introduces or removes a specified amount of heat flow in a region.

Local temperature with respect to a user-specified reference temperature establishes the distribution of this set amount of heat source or sink through the space of the region. In this way, the distribution of heat within the fluid is similar to what would be seen with a heat exchanger. It is associated with one fluid stream only and is a net source or sink with the model instead of a transfer from one continuum to another.

This device is a simple way to incorporate a known heat load into (or from) a model with a minimum of effort and a minimal computational burden. Sample applications include air conditioner evaporators, condensers, charge air coolers, radiators, electric heaters, and electronic devices.

The method is relatively insensitive to mesh size and utilizes resolution corresponding the grid fineness.

The Energy Source Option activates the heat exchanger enthalpy source.

Single-Stream Heat Exchanger Workflow

1. Select an energy model.
2. Select **Regions** > **[region node]** > **Physics Conditions** > **Energy Source Option** and set the *Energy Source Option* property to **Heat Exchanger**.
3. Select **[region node]** > **Physics Conditions** > **Heat Exchanger Exit Temperature Specification** and set the *Outlet Temperature* property to **Inferred** or **Specified**:
   - **Inferred** means the outlet temperature T_{out} is inferred. You must then specify the heat transfer rate Q using the **Heat Exchanger Energy Transfer** node under **Physics Values**.
   - **Specified** means that you specify the outlet temperature T_{out} using the **Target Exit Temperature** node under **Physics Values**. The heat transfer rate Q is then inferred.
4. Set heat exchanger properties. Nodes for these properties become available under **[region node]** > **Physics Values**:
   - **Heat Exchanger Minimum Temperature Difference**
   - **Heat Exchanger Boundaries**
   - **Heat Exchanger First Iteration**
   - **Heat Exchanger Temperature**

Physics Values

Heat Exchanger Energy Transfer

This node's *Value* sets the fixed rate of heat transfer, Q_{total}, from or to the device, typically measured in Watts. This quantity represents the total energy transfer for the heat exchanger enthalpy source.

Heat Exchanger Minimum Temperature Difference

This node's *Value* represents the minimum allowable temperature difference between the following:

- The specified heat exchanger temperature and the maximum fluid temperature for a heater (that is, when the specified Heat Exchanger Energy Transfer is positive)
- The specified heat exchanger temperature and the minimum fluid temperature for a cooler (that is, when the specified Heat Exchanger Energy Transfer is negative).

For a heater, the hot stream is assumed to have an infinite thermal capacitance, and the specified heat exchanger temperature T_h refers to the constant temperature of the hot stream. Now, given the heat exchanger energy transfer rate Q, the inlet cold stream temperature T_{ci}, and the cold stream specific heat c_{pc}, an energy balance could be used to predict the cold stream outlet temperature T_{co} as follows:

$$T_{co} = T_{ci} + \frac{Q}{\dot{m}_c c_{pc}} \tag{306}$$

where $\dot{m}_c$ is the mass flow rate of the cold stream. Physically, it is impossible to have $T_h < T_{co}$. However, if the heat exchanger temperature value is erroneously specified below the maximum cold stream temperature in the heat exchanger, the specified heat exchanger minimum temperature difference is used to correct the heat exchanger temperature. The heat exchanger temperature stays above the predicted maximum cold stream temperature. When this error occurs, a warning message appears on the console, which is an indication that the heat exchanger temperature specification is determined to be unphysical.

Similarly, for a cooler, the cold stream is assumed to have an infinite thermal capacitance, and the specified heat exchanger temperature T_c refers to the constant temperature of the cold stream. Now, given the heat exchanger energy transfer rate Q, the inlet hot stream temperature T_{hi}, and the hot stream specific heat c_{ph}, an energy balance could be used to predict the hot stream outlet temperature T_{ho} as follows:

$$T_{ho} = T_{hi} - \frac{abs(Q)}{\dot{m}_h c_{ph}} \tag{307}$$

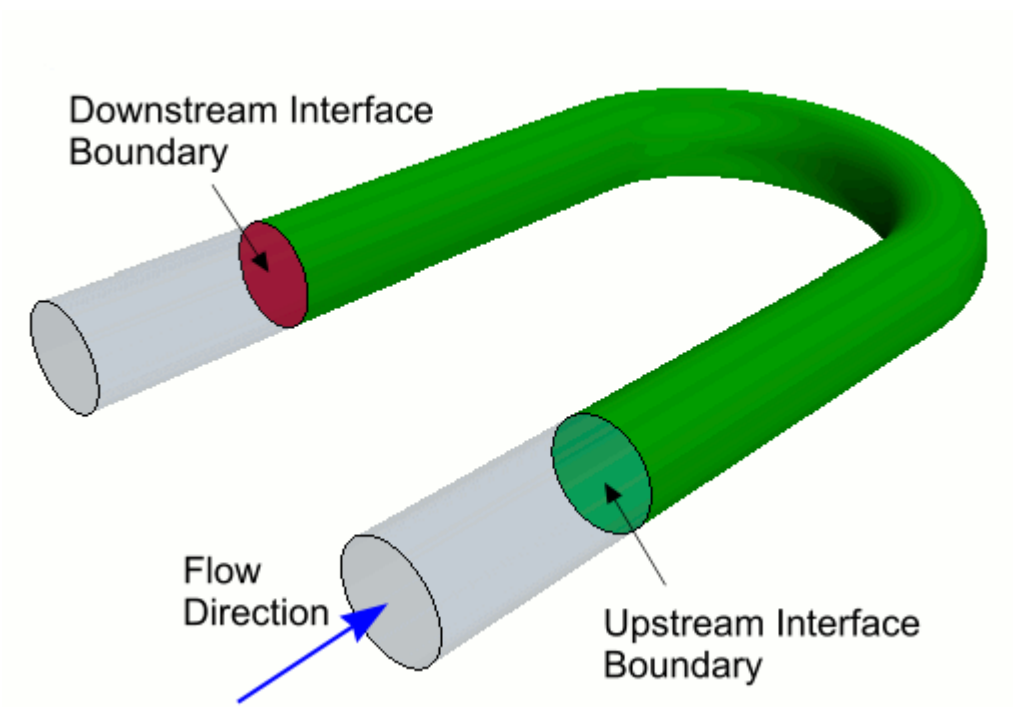
where $\dot{m}_h$ is the mass flow rate of the hot stream. Physically, it is impossible to have $T_c > T_{ho}$. However, if the heat exchanger temperature value is erroneously specified above the minimum hot stream temperature in the heat exchanger, the specified heat exchanger minimum temperature difference is used to correct the heat exchanger temperature. The heat exchanger temperature stays below the predicted minimum hot stream temperature. When this error occurs, a warning message appears on the console, which is an indication that the heat exchanger temperature specification is determined to be unphysical.

Heat Exchanger Boundaries

The properties of this region value node are used to specify the upstream and downstream interface boundaries that the Heat Exchanger (Single Stream) report uses:

- *Upstream Interface Boundary*: Select the interface boundary through which fluid enters the heat exchanger region.
- *Downstream Interface Boundary*: Select the interface boundary through which fluid leaves the heat exchanger region.

This node assumes that the heat exchanger region interfaces with two other regions as shown in the example below.



Heat Exchanger First Iteration

Use this node to set the *First Heat Exchanger Iteration* from which the heat exchanger enthalpy source is active and begins its contribution to the source term of the energy equation.

Ideally, activate the heat exchanger source term when the flow solution is beginning to conform to its final pattern (at least in terms of direction). Conformity could typically be achieved within the first 50 iterations of the flow solver for a moderately complex geometry.

Heat Exchanger Temperature

This region value node represents the reference temperature T_{ref} of the heat exchanger enthalpy source. It is entered as a scalar profile.

The local fluid temperature at each cell is subtracted from the reference temperature T_{ref} to determine the relative amount of heat that is transferred from and/or to the cell. See Eqn. (334).

Target Exit Temperature

The properties of this node allow you to specify the temperature at the outlet, (the *Target Outlet Temperature*), while allowing STAR-CCM+ to compute the overall *Heat Transfer Rate* (shown as a read-only property). You can optionally set the *Under-Relaxation Factor*, which governs the extent to which the new calculation supplants the old one as STAR-CCM+ computes the heat transfer rate.

This node becomes available when the *Outlet Temperature* property of the **Heat Exchanger Exit Temperature Specification** region condition node is set to Specified.

Using the Dual Stream Heat Exchanger Option

Two versions of the dual stream heat exchanger option exist in STAR-CCM+: basic and actual.

This section describes the heat exchanger topology necessary to simulate the dual stream heat exchanger in STAR-CCM+, how to create the heat exchanger topology, and define a heat exchanger interface between two regions.

The dual stream heat exchanger model supports the following:

- A fluid-solid interface, where the heat exchange occurs between a solid region and a fluid region.
- A fluid-fluid interface, where the heat exchange takes place between two fluid regions.

For a fluid-fluid type heat exchanger, the following options are available for specifying the heat transfer rate in the heat exchanger:

- UAL Polynomial
- UAL Table
- UAG Table
- Q Table
- Q Map
- UAL Map

An option is included to specify the target heat rejection rate to enable predicting the hot flow inlet boundary temperature.

For a fluid-solid type heat exchanger, the following option is made available:

- Heat Exchanger UAL

The formulation for this option describes the above quantities and discusses the implementation aspects.

Contents:

What Are the Basic and Actual Dual Stream Heat Exchanger Options?
Dual-Stream Heat Exchanger Workflow
Creating the Heat Exchanger Topology
Heat Exchanger Data Specification Reference
What Is Target Heat Rejection?
What Is the Heat Exchanger UAL?

The Basic and Actual Dual Stream Heat Exchanger Options

The basic and actual dual stream heat exchanger options use different approaches for the solution.

- The basic option solves for a simpler version of the energy equation, Eqn. (2076). The simpler version only uses the convective term and the source term to yield a solution independent of the solution that is generated for the overall simulation. This separate solution consists of pseudo-temperatures for the hot and cold regions of the heat exchanger, which are used to compute the mean temperature difference and hence the local heat transfer coefficient, UAL.
- In order to compute the mean temperature difference and hence the UAL, the actual option uses the flow and energy values that are calculated for the simulation itself.

The basic option uses a wider range of properties settings than the actual. Specifics are given in the descriptions of affected objects.

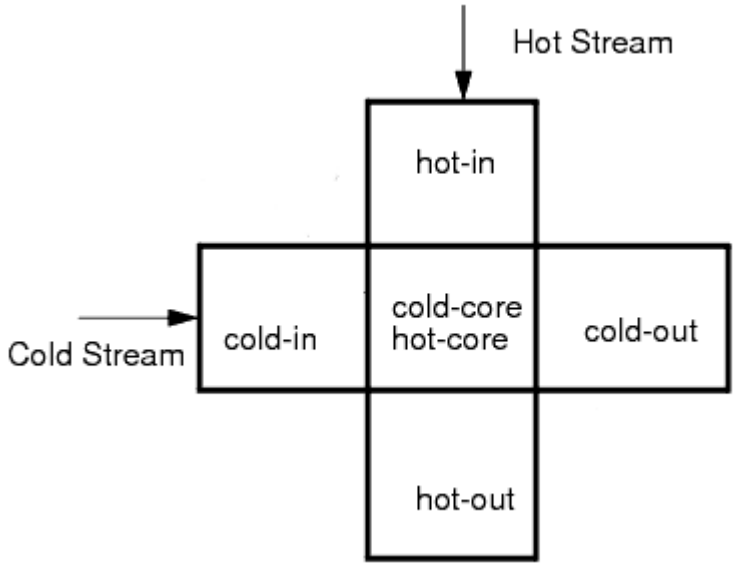
When using the basic option, ensure that the core mesh is constructed of uniform cells. That is, all the cells in the heat exchanger region must be of comparable cell volume, as the cell volumes do not enter the calculation of the UAL table in the basic model. This restriction does not apply when using the actual option.

Note: UA refers to the product of the Overall heat transfer coefficient (U) and the Area of the heat exchanger (A), specified in W/K. UAL refers to the local (or cell value of) heat transfer coefficient (in W/K), with L referring to local.

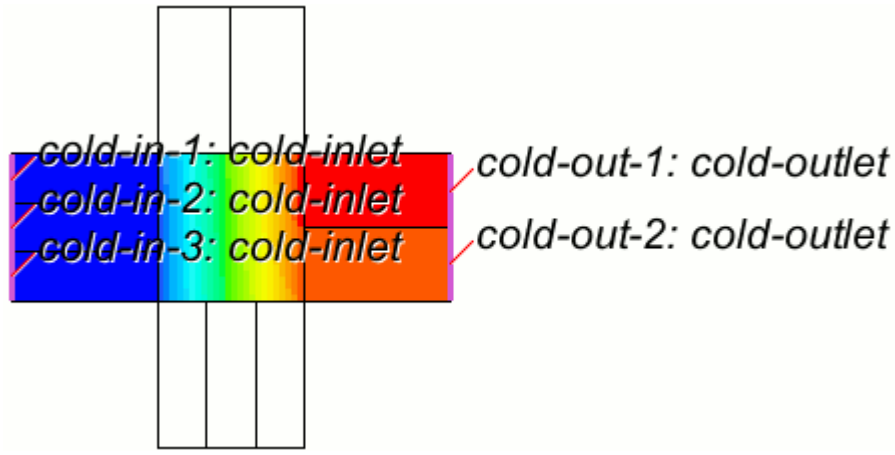
Dual-Stream Heat Exchanger Workflow

To use the dual-stream heat exchanger option, you must define the interface between two streams using a common mesh and separate physics continua. One physics continuum represents the hot stream, and the other represents the cold stream.

1. For each physics continuum, make sure that an energy model is active.
2. Define a region topology with six regions that conform to the following structure (actual shapes are determined by the real geometry):



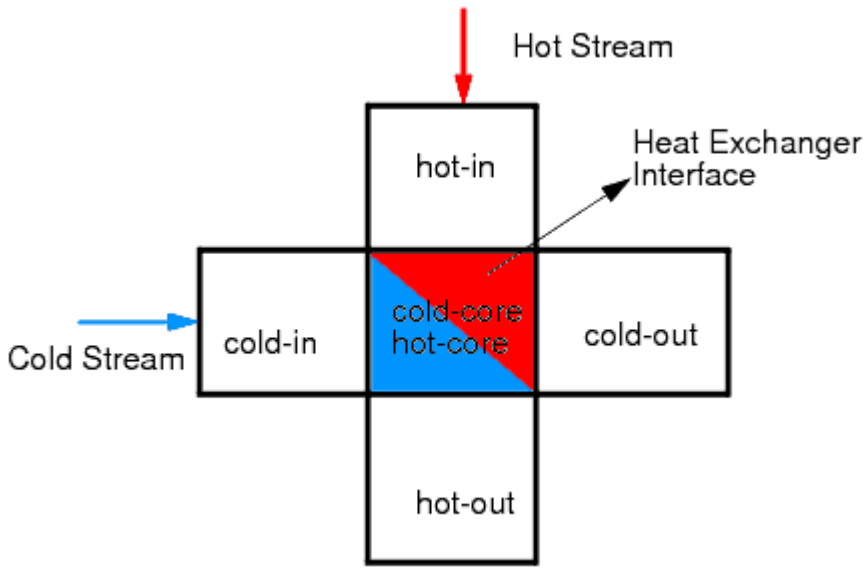
- Create the regions **cold-core** and **hot-core** from the same mesh. This continuum is where the heat exchange occurs. For maximum accuracy when using the basic dual stream heat exchanger, ensure that the mesh that is used for the core region is constructed of uniform cells. To do this, make a copy of the initial core region. The accuracy of results that the actual dual stream heat exchanger predicts does not depend on the uniformity of the mesh.
- The regions **cold-in**, **cold-core**, and **cold-out** represent the cold stream. Assign the equivalent regions of the simulation to a single continuum, the "**Cold Continuum**."
- The regions **hot-in**, **hot-core**, and **hot-out** represent the hot stream. Assign the equivalent regions of the simulation to a single continuum, the "**Hot Continuum**."
- The diagram shows only one inlet and one outlet for each stream, but you can also specify multiple inlets and outlets for a heat exchanger. The figure below shows a heat exchanger with three inlets and two outlets on the cold stream.



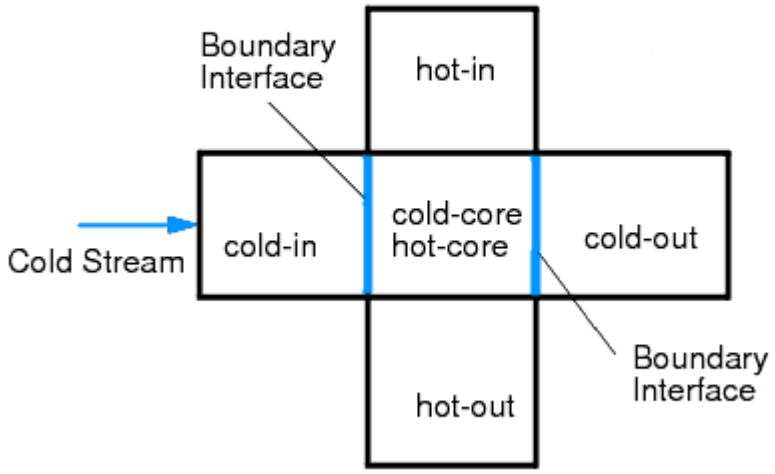
3. Define the heat exchanger interface:

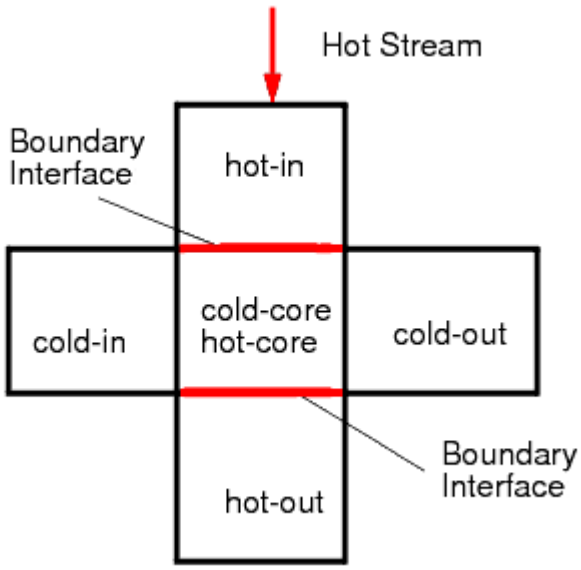
a) Multi-select the cold-core and hot-core regions, right-click and select **Create Interface** from the menu.

b) Set *Type* to **Heat Exchanger**.



4. Define internal (or in-place) boundary interfaces between the common boundaries of the regions equivalent to **cold-in** and **cold-core**, **cold-core** and **cold-out**, **hot-in** and **hot-core**, and **hot-core** and **hot-out**.



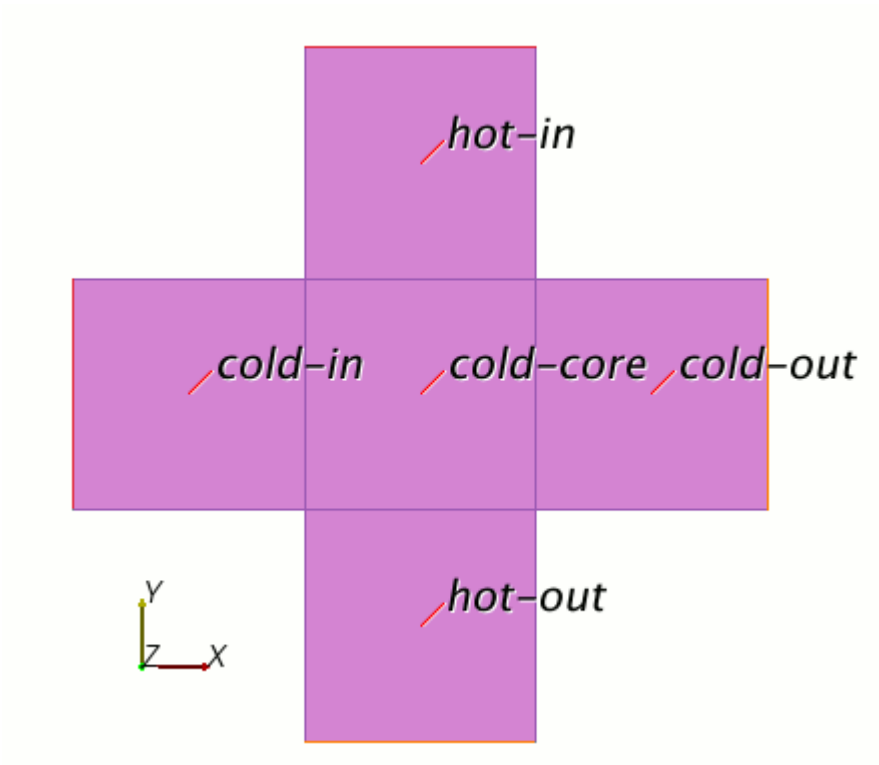


5. Select **Basic** or **Actual Flow Dual Stream** as the heat exchanger method:
   a) Select **Interfaces** > **[heat exchanger]** > **Physics Conditions** > **Heat Exchanger Method**
   b) Set the *Option* property to **Basic Dual Stream** or **Actual Flow Dual Stream**. See The Basic and Actual Dual Stream Heat Exchanger Options for guidelines on choosing between them.

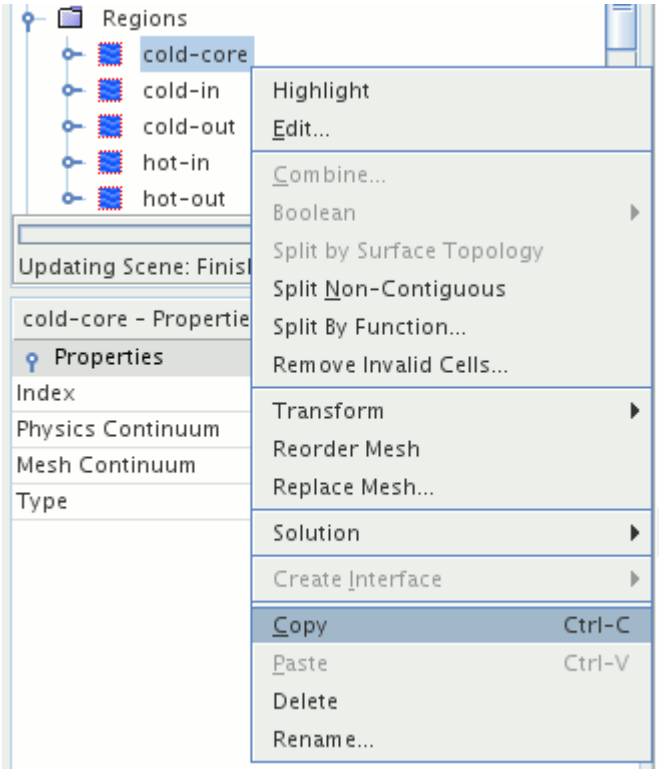
   By default, the **Heat Exchanger Method** stays **Inactive**, which indicates that the source terms of the energy equation are not affected.
6. Set the heat transfer rate.
   a) Select **Physics Conditions** > **Heat Exchanger Data Specification**.
   b) Set the Method property. See Heat Exchanger Data Specification Reference for guidelines on choosing the method.
7. Set target heat rejection, if necessary.
   a) Select **Interfaces** > **[heat exchanger]** > **Physics Conditions** > **Hot Stream Inlet Temperature Specification**
   b) Set the *Hot Inlet Temperature* property to **Inferred** if you do not want to use the default setting of **Specified**. See Target Heat Rejection for guidelines on choosing between them.

Creating the Heat Exchanger Topology

This section explains how to create the heat exchanger topology, starting with the following set of regions:



The figure above shows five regions - **cold-in**, **cold-core**, **cold-out**, **hot-in** and **hot-out**. Ensure that the regions **cold-in**, **cold-out** and cold-core have their physics continuum set as **Cold Continuum** and the regions **hot-in** and **hot-out** have their physics continuum set as **Hot Continuum**. Copy the **cold-core** region as shown in the following screenshot:

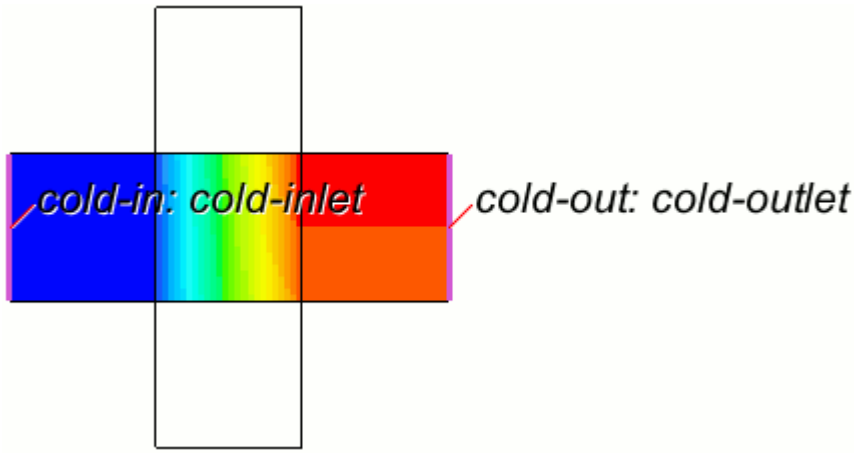


To ensure that both the regions are created from the same mesh continuum, copy the region after the volume mesh has been generated.

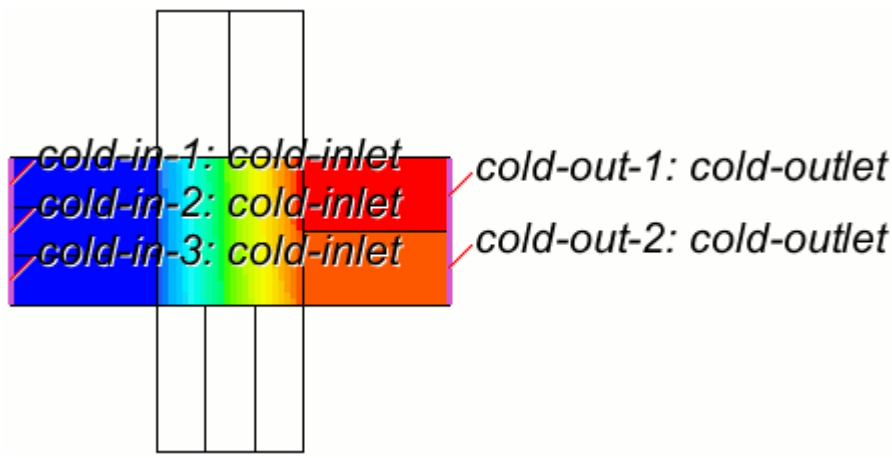
Paste the copied region into the **Regions** node as shown below:

Rename the new region as **hot-core** and change its physics continuum to **Hot Continuum**. Now, the simulation has six regions - **cold-in**, **cold-core**, **cold-out**, **hot-in**, **hot-core**, **hot-out**. Define internal (or in-place) boundary interfaces between the common boundaries of **cold-in** and **cold-core**, **cold-core** and **cold-out**, **hot-in** and **hot-core**, and **hot-core** and **hot-out**. The creation of the heat exchanger topology is now complete.

You can also specify multiple inlets and outlets for a heat exchanger. The figure below shows a heat exchanger with a single inlet and a single outlet.



The next figure shows the heat exchanger with three inlets and two outlets.



Heat Exchanger Data Specification Reference

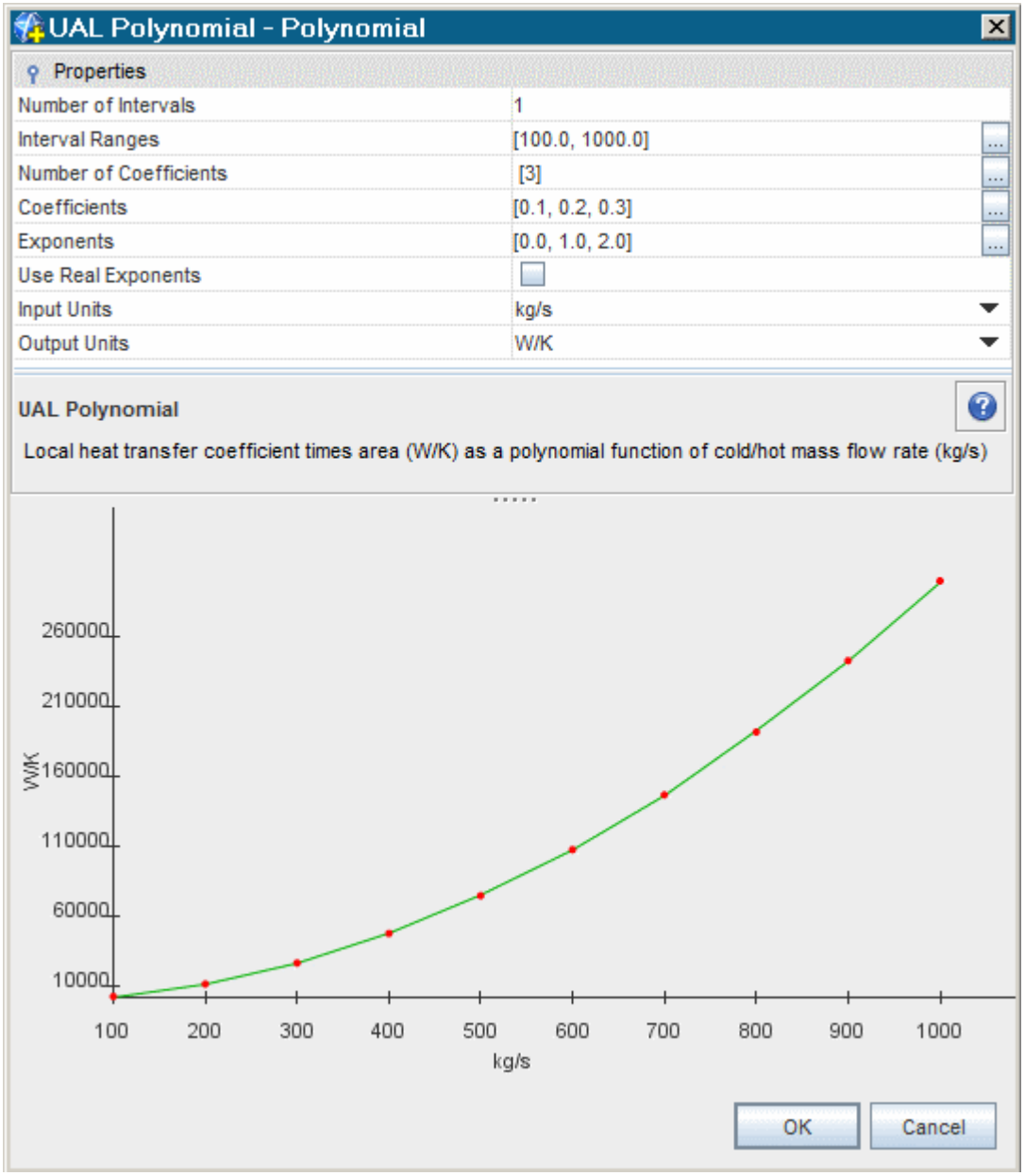
When the **Basic Dual Stream** or **Actual Flow Dual Stream** option is chosen as the **Heat Exchanger Method**, a new physics condition, **Heat Exchanger Data Specification** appears. This node allows you to choose an input option.

Heat Exchanger Data Specification Methods

UAL Polynomial

This polynomial option assumes that UAL is known as a function of the cold flow velocity v for the **Basic Dual Stream** option and as a function of the cold stream mass flow rate for the **Actual Flow Dual Stream** option.

The known polynomial can be specified by selecting the **UAL Polynomial** node. The following example illustrates the specification of the function $UAL = 0.1 + 0.2v + 0.3v^2$ defined within the range $v = 100\frac{kg}{s}$ to $v = 1000\frac{kg}{s}$



UAL Table

This option assumes that the local heat transfer coefficient, UAL, is known as a table of the cold flow velocity for the **Basic Dual Stream** option and as a table of the cold stream mass flow rate for the **Actual Flow Dual Stream** option.

For a known local cell velocity of the cold stream, the UAL is determined by interpolating from this table. The UAL table must contain two columns of data representing the cold stream velocity and UAL. The table must be arranged in an ascending order of the velocities.

UAG Table

This option assumes that the overall heat transfer coefficient UAG across the heat exchanger is known as a table of the cold stream mass flow rate (say, from an experimental source) for one value of hot stream mass flow rate.

UAG refers to global (or) overall heat transfer coefficient for the entire heat exchanger, with the G referring to global.

The UAG Table is first converted into a UAL table (that is, for each row of the UAG table, a UAL is computed), which is ultimately used to compute the local heat transfer coefficient for every cell of the heat exchanger region, and hence the energy source for the energy equation.

In order to compute UAL from UAG, with **Basic Dual Stream** option, the simulation solves a linear system of algebraic equations by considering the heat balance between the cold and hot streams while preserving the overall heat transfer rate across the heat exchanger.

With the **Actual Flow Dual Stream** heat exchanger option, the simulation computes UAL from UAG by using the hot and cold temperatures from the actual simulation, thus eliminating the need to solve any linear system of equations.

With the **Basic Dual Stream** heat exchanger option, the UAL table is computed from the UAG table in the very first iteration and then held constant for the rest of the simulation, unless any property of the UAG table is changed by the user. Any such change triggers a re-compute of the UAL table.

With the **Actual Flow Dual Stream** heat exchanger option, the UAL table is recomputed at every solver iteration as the local hot stream and cold stream temperatures converge within the heat exchanger region.

Using the computed UAL table, both the **`Basic`** and **`Actual Flow`** options interpolate the UAL value for each cell of the heat exchanger region from the UAL table for a given cold stream local velocity. This UAL is then used to compute a local energy source or sink for the energy equations of the cold and hot fluid streams respectively.

Q Table

This option assumes that the overall heat transfer Q across the heat exchanger is known as a table of the cold stream mass flow rate (say, from an experimental source) for one value of hot stream mass flow rate.

The user-supplied Q Table is first converted internally into a UAG table using the expression:

$$UAG = \frac{Q}{T_{hot}^{inlet} - T_{cold}^{inlet}}$$

where T_{hot}^{inlet} and T_{cold}^{inlet} are the test rig hot and cold temperatures that are specified as properties of the Q Table.

After converting the Q Table into a UAG Table, the local heat transfer coefficients (UAL) are computed as described in the UAG Table section.

Q Map

A Q Map provides STAR-CCM+ with a series of curves from which it calculates the local heat transfer coefficient, UAL.

One set of curves on the map provides UAL values for constant cold stream mass flow rates m_{cold} and varying hot stream mass flow rates m_{hot}. These curves are intersected by other curves that provide UAL values for varying m_{cold} with constant m_{hot}.

UAL Map

This method is available with the **Actual Flow Dual Stream** heat exchanger option only.

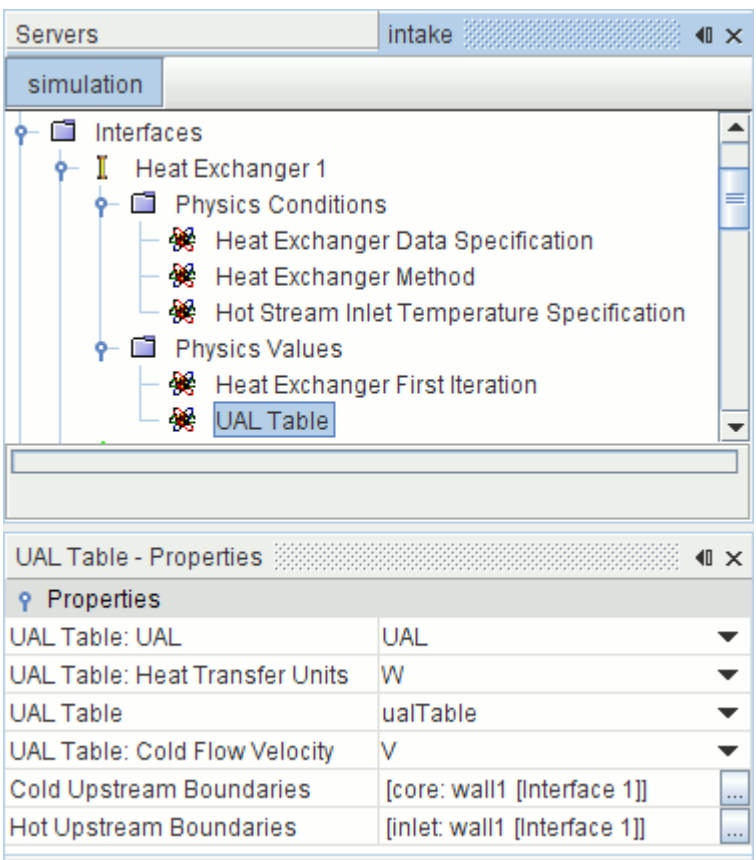
The UAL Map allows a map of curves (m_{cold}, m_{hot} and UA_l) to specify the UAL directly when multiple heat exchanger curves are available. This option requires you to specify a table containing m_{cold}, m_{hot} and UA_l. That is, specify the mass flow rates of the cold and hot streams rather than the velocities. If the velocities of the two streams are known, they can be multiplied by the product of the average density and inlet flow area, thus creating the UAL map table.

Volumetric UAL Table

This option is a variation on the UAL Table option, accepting input in volumetric form, in units such as `W/K-m^3` rather than `W/K`. It is available only with the **Actual Flow Dual Stream** heat exchanger option.

Input Tables

All the methods other than **UAL Polynomial** use input tables. These tables can be imported into the simulation through the **Tools** node. Once imported, the input table can be selected into the method node. The following screen shot shows the **ualTable** selected for the **UAL Table** value node.



Properties Lookup

This table shows which properties are used by which input methods.

Use the following abbreviations:

UP = UAL Polynomial
UalT = UAL Table
UagT = UAG Table
QT = Q Table
QM = Q Map
UM = UAL Map
VT = Volumetric UAL Table

Entries marked **A** apply only with the **Actual Flow Dual Stream** option. Entries marked **B** apply only with the **Basic Dual Stream** option.

	UP	UalT	UagT	QT	QM	UM	VT
Coefficients C0, C1, C2 Coefficients C_0, C_1, and C_2 in Eqn. (342). These computed values are read-only.							
↳ Applies to:					**B**		
Cold Flow Exponent Exponent ec in Eqn. (342).							
↳ Applies to:					**B**		
Cold Fluid Density Test condition density of the cold fluid.							
↳ Applies to:			**B**	**B**	**B**		
Cold Fluid Inlet Temperature Test condition inlet temperature of the cold stream.							
↳ Applies to:			**B**	**A, B**	**A, B**		
Cold Fluid Specific Heat Test condition specific heats of the cold stream.							
↳ Applies to:			**B**	**B**	**B**		
Cold Upstream Boundaries Upstream boundary interfaces of the cold stream.							
↳ Applies to:	**A, B**	**A, B**	**A, B**	**A, B**	**A, B**	**A**	**A**
Convergence Tolerance Tolerance on converging the solver to the target heat transfer rate.							
↳ Applies to:			**B**	**B**	**B**		

	UP	UalT	UagT	QT	QM	UM	VT
Goodness of fit A read-only value ranging from 0 through 1 that indicates how well the fitted curve represents the data. Typically, this value can be expected to be within 0.96 and 0.98 for a reasonable fit.							
↳ Applies to:					**B**		
Hot Flow Exponent Exponent eh in Eqn. (342).							
↳ Applies to:					**B**		
Hot Fluid Density Test condition density of the hot fluid.							
↳ Applies to:					**B**		
Hot Fluid Inlet Temperature Test condition inlet temperature of the hot stream							
↳ Applies to:			**B**	**A, B**	**A, B**		
Hot Fluid Mass Flow Rate Test condition mass flow rate of the hot stream.							
↳ Applies to:			**B**	**B**			
Hot Fluid Specific Heat Test condition specific heats of the hot stream.							
↳ Applies to:			**B**	**B**	**B**		
Hot Mass Flow Rate Specified When **On**, causes the hot stream to use the mass flow rate that the table specifies.							
↳ Applies to:	**A**	**A**	**A**	**A**			**A**
Hot Upstream Boundaries Upstream boundary interfaces of the hot stream.							
↳ Applies to:	**A, B**	**A, B**	**A, B**	**A, B**	**A, B**	**A**	**A**
Inlet Temperature Averaging Selects the form of averaging used when computing UAL table values, either **Area Averaged** (see Eqn. (343)) or **Mass Flow Averaged** (see Eqn. (344)). An expert property.							
↳ Applies to:			**A**	**A**	**A**		

	UP	UalT	UagT	QT	QM	UM	VT
Maximum Solver Iterations The maximum number of iterations permissible for solving the linear system of equations to compute UAL.							
↳ Applies to:			**B**	**B**	**B**		
Polynomial The dialog for editing the polynomial properties. Applies only to the **UAL Polynomial** method.							
↳ Applies to:	**A, B**						
Q Map Set the appropriate the input table and assign columns to the following properties: *Q Table: Q* *Q Table: Cold Mass Flow Rate* *Q Table: Hot Mass Flow Rate* Also set *Q Map: Heat Transfer Units*.							
↳ Applies to:					**A, B**		
Q Table Set the appropriate input table and assign columns to the following properties: *Q Table: Q* *Q Table: Mass Flow Rate* (**A** only) *Q Table: Cold Mass Flow Rate* (**B** only) Also set *Q Table: Heat Transfer Units*.							
↳ Applies to:				**A, B**			
UAG Table Set to the imported table from the **Tools** > **Table** node. Assign table columns to the following properties: *UAG Table: UAG* *UAG Table: Cold Mass Flow Rate* (**B** only) *UAG Table: Mass Flow Rate* (**A** only) Also set *UAG Table: Heat Transfer Units*.							
↳ Applies to:			**A, B**				
UAL Map Set to the appropriate input table and assign columns to the following properties: *UAL Map: Cold Mass Flow Rate* *UAL Map: Hot Mass Flow Rate* *UAL Map: UAL* Also set *UAL Map: Heat Transfer Units*.							
↳ Applies to:						**A**	

	UP	UalT	UagT	QT	QM	UM	VT
UAL Table Set to the imported table from the **Tools** > **Table** node. Assign table columns to the following properties: *UAL Table: Mass Flow Rate* (**A** only) *UAL Table: UAL* Also set *UAL Table: Heat Transfer Units*.							
↳ Applies to:		**A, B**					
Volumetric UAL Table Set to the appropriate input table and assign table columns to the following properties: *Volumetric UAL Table: Mass Flow Rate* *Volumetric UAL Table: UAL* Also set *Volumetric UAL Table: Heat Transfer Units*.							
↳ Applies to:							**A**

Target Heat Rejection

STAR-CCM+ allows you to set a target heat rejection rate for the heat exchanger without setting the temperature at the hot inlet. STAR-CCM+ then predicts the temperature that is required at the inlet to sustain the heat rejection rate.

This Target Heat Rejection option is selected by setting the *Hot Inlet Temperature* property of the **Hot Stream Inlet Temperature Specification** node to **Inferred**.

When you select this option, STAR-CCM+ iteratively modifies the hot stream inlet boundary temperature until the specified heat rejection rate is achieved.

The default option for the **Hot Stream Inlet Temperature Specification** node is **Specified**. This option indicates that you know and specify the hot stream inlet boundary temperature. The **Target Heat Rejection** node is added to the interface **Physics Values** when you choose **Inferred**.

Target Heat Rejection Properties

Target Heat Rejection Rate

The heat transfer rate that is desired from the heat exchanger.

Hot Inlet Boundaries

The hot stream boundaries at which the temperature must be predicted.

Temperature Jump Solver URF

Under-relaxation factor for the heat exchanger temperature jump solver.

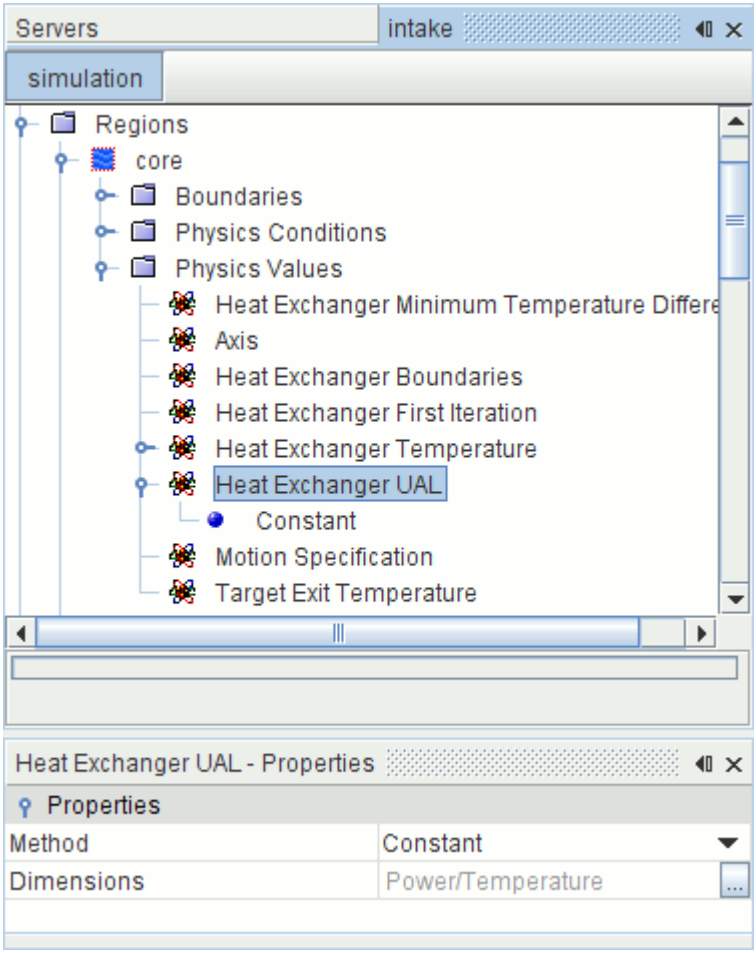
Temperature Prediction Accuracy

An advanced option which allows you to control the accuracy of temperature prediction.

What Is the Heat Exchanger UAL?

This option is made available for a solid-fluid type dual stream heat exchanger, which is created between a solid region and a fluid region or vice versa.

For a solid-fluid type heat exchanger, when the basic dual stream option is chosen as the Heat Exchanger Method, the **Heat Exchanger UAL** option is added as a physics value node to the fluid region. This option is entered as a scalar profile.



The heat exchanger UAL scalar field represents the local (or cell) heat exchange rate, that may either be specified as a constant or a function of the fluid properties. This user-specified value is added to the source term of the fluid energy equation and subtracted from the source term of the solid energy equation.

Modeling Energy in Solids

This section provides information about the properties of the solid energy models in STAR-CCM+.

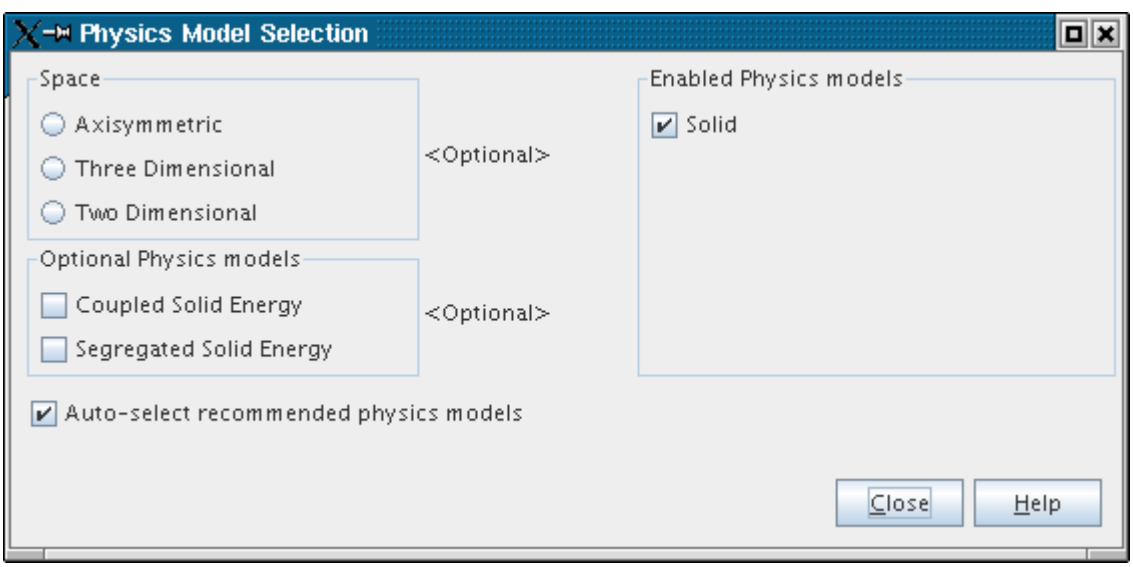
- Coupled Solid Energy
- Segregated Solid Energy
- Modeling Heat Conduction in Solid Shells
- Modeling Exothermic Solids
- Solid Energy Formulation

Contents:

Modeling Solid Energy Using a Coupled Approach
Modeling Solid Energy Using a Segregated Approach
Modeling Heat Transfer in Solid Shells
Modeling Exothermic Solids
Modeling Average Circumferential Heat Flux

Modeling Solid Energy Using a Coupled Approach

The Coupled Solid Energy model is available for selection after you enable the Solid model during material model selection.



Properties

Implicit	Uses a checkbox to select the integration scheme to use.

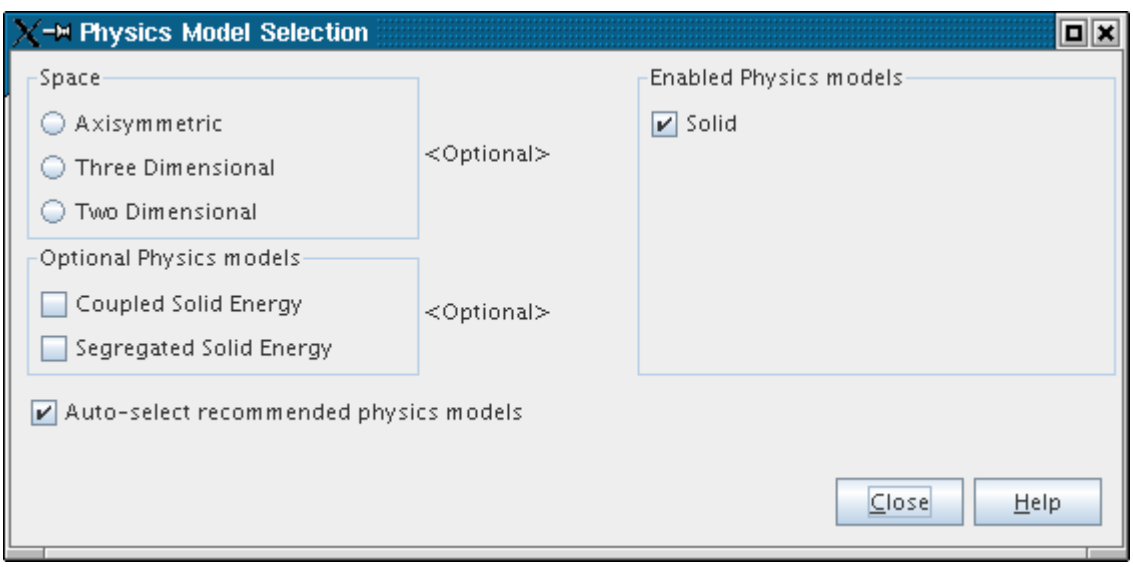
	Activated	The coupled implicit solver is invoked. Adds a Coupled Implicit node to the **Solvers** node.
	Deactivated	The coupled explicit solver is invoked. Adds a Coupled Explicit node to the **Solvers** node.

Expert Properties

Secondary Gradients	Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces.	
	On	Include both secondary gradients.
	Off	Exclude both secondary gradients.
	Interior Only	Include the interior secondary gradients only.
	Boundaries Only	Include the boundary secondary gradients only.

Modeling Solid Energy Using a Segregated Approach

The Segregated Solid Energy model is available for selection after you enable the Solid model during material model selection.



Properties

None.

Expert Properties

Secondary Gradients	Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces.	
	On	Include both secondary gradients.
	Off	Exclude both secondary gradients.
	Interior Only	Include the interior secondary gradients only.
	Boundaries Only	Include the boundary secondary gradients only.

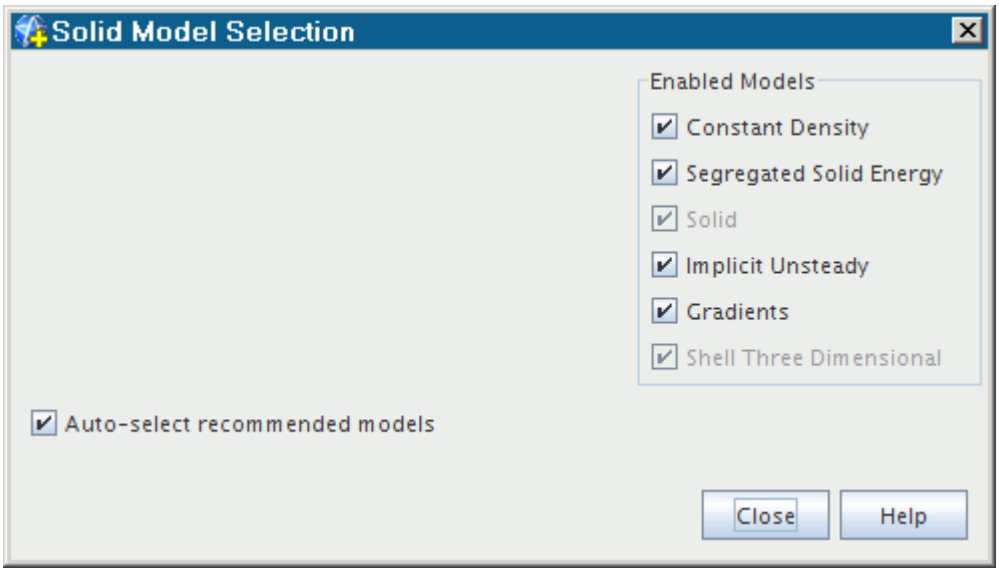
Modeling Heat Transfer in Solid Shells

Problems where thin solid shells are involved in heat transfer are common in engineering practice. For thin solid shells, STAR-CCM+ can model heat conduction within the shell, and radiative heat transfer between the surface of the shell and the surfaces of other objects.

Modeling of in-plane shell heat conduction is required when the following apply:

- The heat conduction in the lateral direction of the shells cannot be neglected.
- The shells are so thin that their meshing is not practical.

Heat transfer in solid shells happens on the surface of the shell. Therefore, to provide support for the discretization on surfaces, include the **Shell Three-Dimensional** model in a physics continuum for solid shells.



The rest of the model selection is identical to that of solid materials in conjugate heat transfer problems.

A shell region is a surface domain in space and can be created using region boundaries. Each shell region is assigned a unique name that is based on the name of the boundary which is used for the shell region creation, although it can be renamed.

A shell region is joined to another region, or shell region, using an interface, so that information can be passed between them. An interface between a shell region and the region which owns the boundary that is used to create the shell region is created automatically during the shell region creation.

You can model radiation in solid shells as you do in other volumetric solids. Specifically, you can:

- Set emissivity, reflectivity, specularity, and transmissivity in shell boundaries.
- Model radiative transfer between the external side of the shell and the surfaces of other objects, using the environment boundary condition.
- Model opaque shells by turning off the radiative transfer option within the shell region. (The option is on by default.)

Contents:

Specifying the Solid Shell Thickness for the Region

See Also:

Modeling Space, Time, and Motion

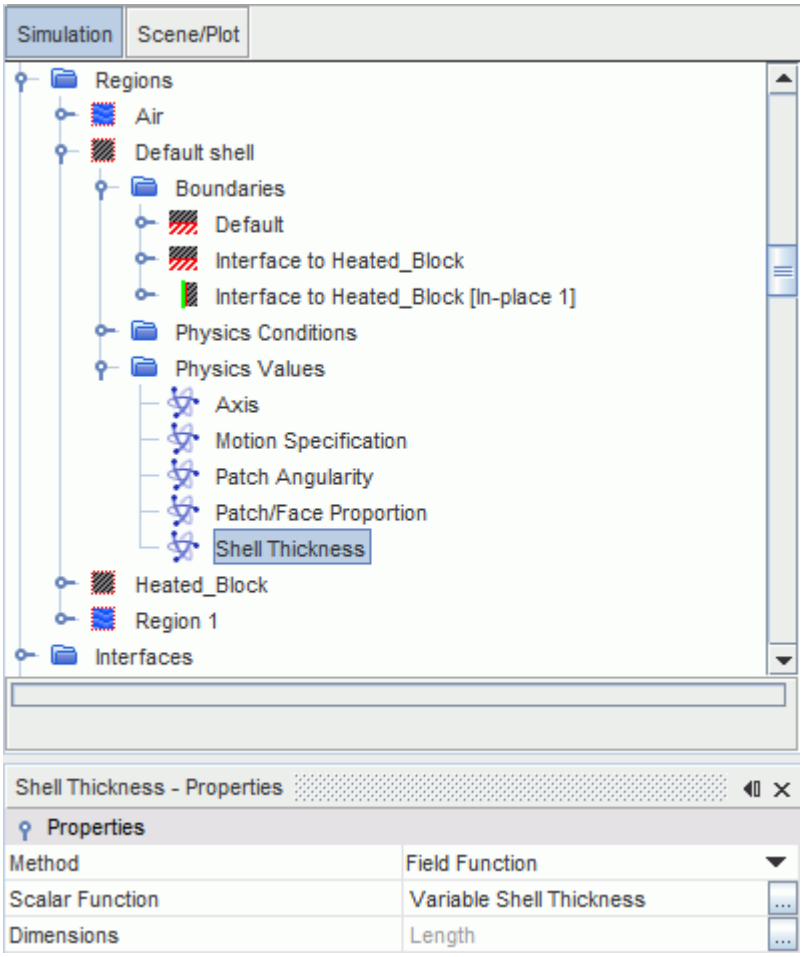
Specifying the Solid Shell Thickness for the Region

You specify the shell thickness as a scalar profile in the **Shell Thickness** node under the **Physics Values** of a shell region.

The shell thickness is available as a field function: **Solid Shell Thickness**.

You can modify the shell thickness as the simulation progresses. If you specify the shell thickness with a user-specified field function, the field function is evaluated with each iteration and the shell thickness is updated accordingly.

The scalar function **Variable Shell Thickness** that is shown in the screenshot below is an example of a user field function that specifies the shell thickness. This field function is not to be confused with the **Solid Shell Thickness** field function that is provided by STAR-CCM+ to make the result of the simulation available. You do not use that field function to specify the input value.



Modeling Exothermic Solids

If temperature exceeds the ignition temperature of an exothermic (mold) material for the first time, a certain amount of energy is released over time.

In a simulation, the cells that represent the exothermic material each burn for the same amount of time, but at different times. This difference in timing makes it difficult to model an exothermic solid through a user-defined volumetric heat source.

To quantify this energy in STAR-CCM+, use the Exothermic Solid model. This model is a simple combustion model that works like a heat source. When temperature in the solid exceeds ignition temperature, a user-specified heat source simulates combustion.

Two material properties, Autoignition Temperature and Power of Reaction, suffice to describe the entire effect of the exothermic solid material on the energy balance. For example, if a simulation involves two different exothermic solid materials, they would differ only in those material properties. There are no reaction constants or, for instance, no Arrhenius equation from which the energy release over time is computed; the Power of Reaction material property gives this quantity directly.

The main simplification is the Reaction Time field function. It returns the time that the temperature exceeds the Auto-Ignition Temperature.

The application for this model is exothermic mold materials. They are used to enhance feeding systems by extending the solidification time of the reservoirs of liquid metal which are intended to compensate for shrinkage on solidification. The exothermic reaction is initiated when molten metal meets the exothermic mold material.

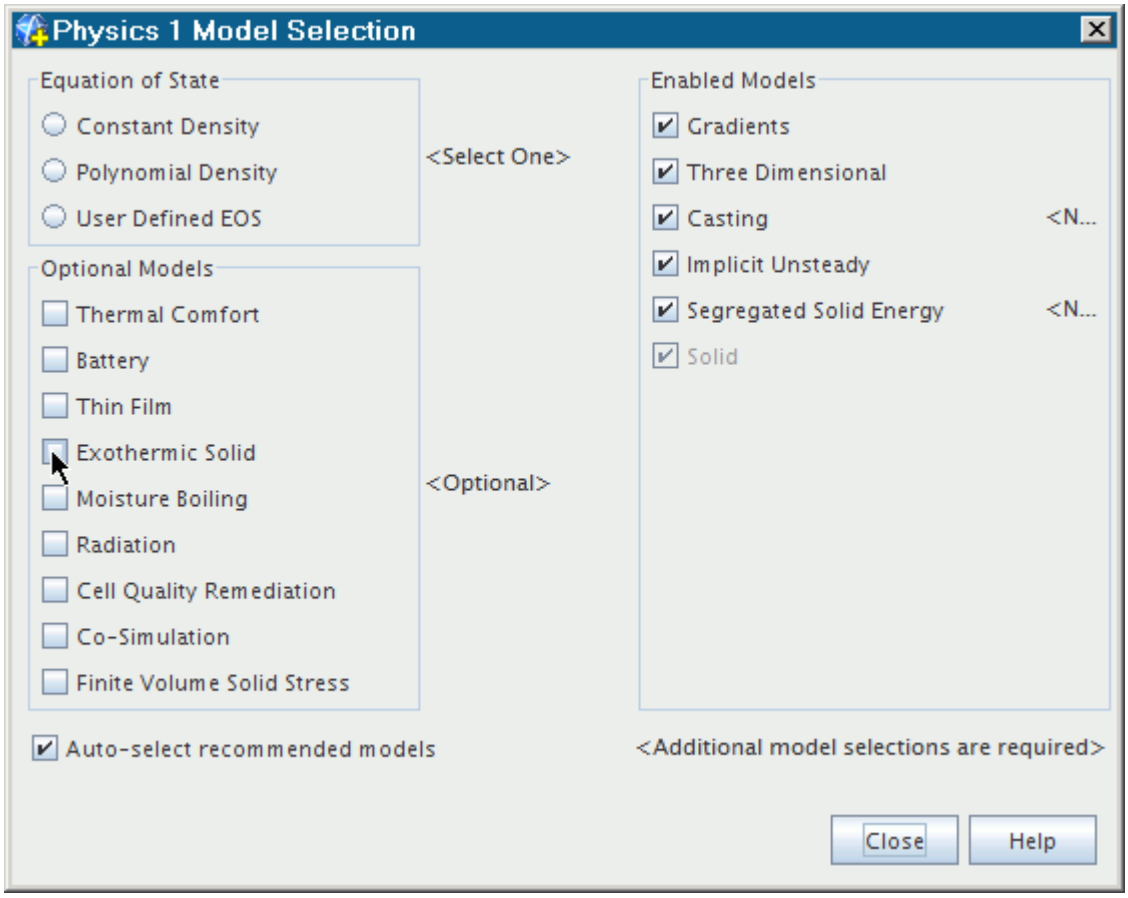
Contents:

Selecting the Exothermic Solid Model
Working with the Autoignition Temperature
Working with the Power of Reaction
Exothermic Solid Field Functions Reference

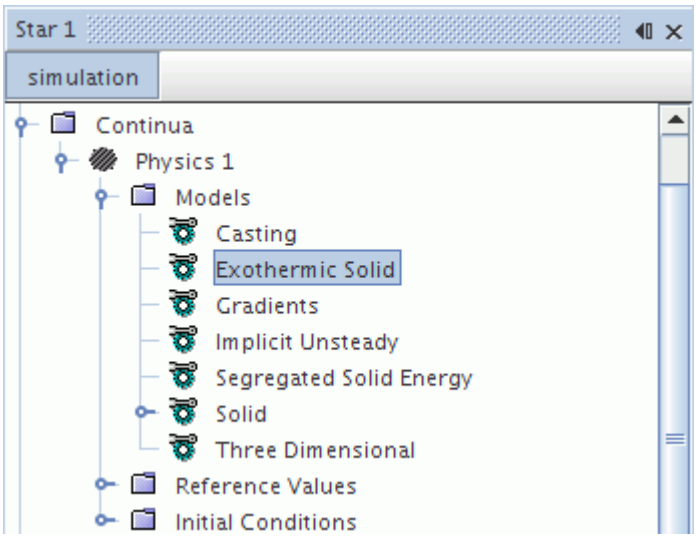
Selecting the Exothermic Solid Model

To select the Exothermic Solid model:

1. Include the following models with your selections:
   - The Solid material model
   - The Segregated Solid Energy model
   - The Implicit Unsteady time model
   - The Casting optional model
2. Select the **Exothermic Solid** model from the *Optional Models* group box.



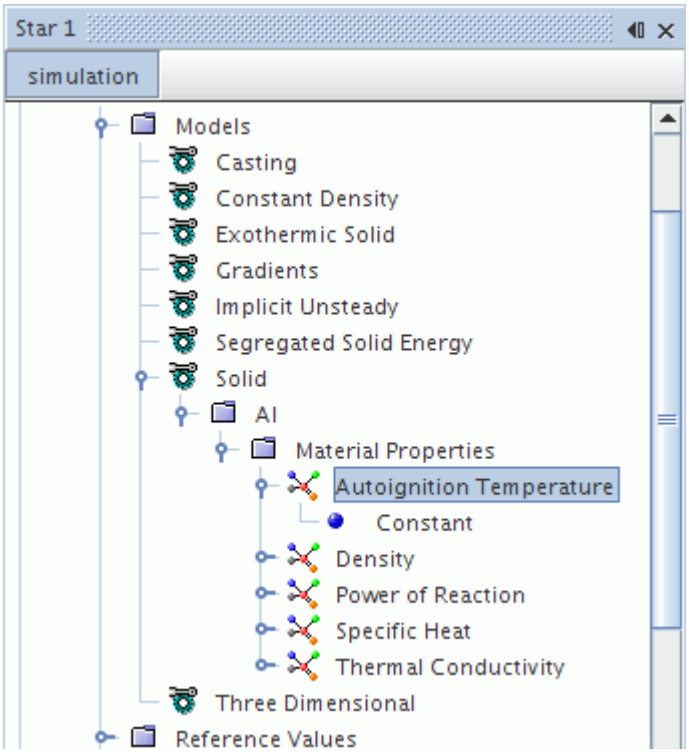
When you close the model selection dialog, the Exothermic Solid model is shown in the list under the **Continua** > **Physics** > **Models** node. The model has no user-defined properties.



Working with the Autoignition Temperature

When you activate the Exothermic Solid model, you can specify the autoignition temperature as a material property.

In the following screenshot, the example shows this property available in the **Continua** > **Physics** > **Models** > **Solid** > **Al** > **Material Properties** node.

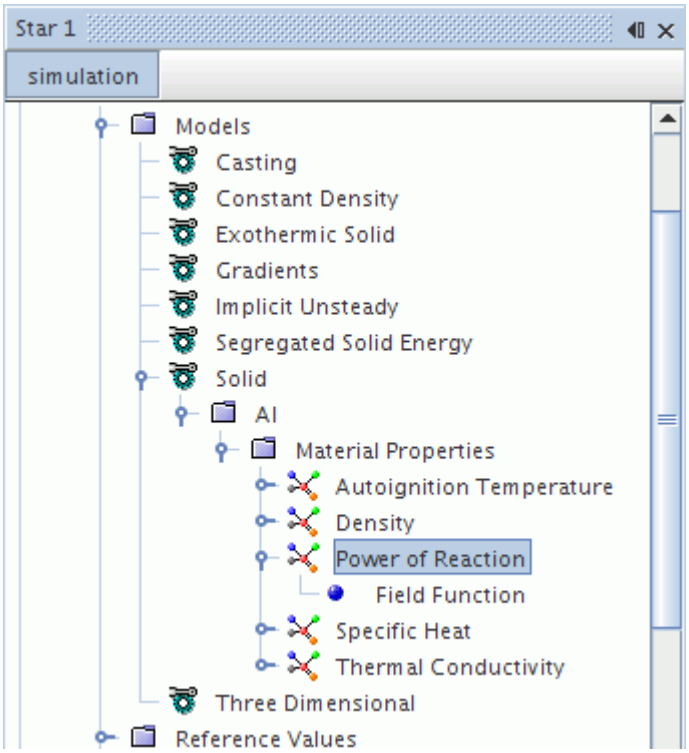


For a calculation method, choose a constant value (K) or a user-defined field function.

Working with the Power of Reaction

When you activate the Exothermic Solid model, you can specify the power of reaction as a material property.

In the following screenshot, the example shows this property available in the **Continua** > **Physics** > **Models** > **Solid** > **Al** > **Material Properties** node.



For a calculation method, choose a constant (W/kg), a user-defined field function, or a Table (Reaction Time) table method. With the Table (Reaction Time) method, the reaction time is used for interpolation of the table and the physical time (as for the regular Table (Time) method).

Exothermic Solid Field Functions Reference

The following primitive field functions are made available to the simulation when the Exothermic Solid model is used.

They have a typical set of base properties.

Autoignition Temperature	The temperature at which a material ignites, and at which reaction time begins to accumulate.
Power of Reaction	The energy release during the reaction time of the material.
Reaction Time	Accumulates the time that the temperature exceeds the autoignition Temperature. If the temperature decreases below the autoignition temperature, the reaction time remains constant until the temperature exceeds the autoignition temperature again.

Modeling Average Circumferential Heat Flux

In a system with a rapidly spinning, axially symmetric body, thermal flux characteristics average out around the circumference of the spinning body. A rotating drive shaft near an engine and a spinning motor in a casing give examples of such systems.

Use the Circumferential Flux Averaging model for such systems. This is a heat transfer model for the special case of rapidly rotating bodies, part of the moving reference frame feature. The model maps out equal-width strips of constant temperature along the axis of the spinning body, using the mesh boundaries for averaging.

The model covers two general cases:

- A solid body spinning rapidly in an interfacing fluid medium. Convection, conduction, and radiation operate within in the fluid. Conduction and radiation operate within the solid. The interface is opaque to radiation.
- A solid or fluid body spinning rapidly in a region without mesh, with no medium between the body and its environment. The body and the external environment interact only by thermal radiation between their boundaries, across the empty meshless region. Within the body, convection (if it is fluid), conduction, and radiation can all operate. Convection and radiation heat transfer operate between the boundaries of the body and the environment.

Only use the Circumferential Heat Flux Averaging model under the following conditions:

- The body in question must be axisymmetric, with little or no circumferential variation in any properties or physical conditions. Minor variations from symmetry may still give acceptable results, but use careful judgement.
- The axisymmetric body must be spinning very fast, and the difference between the angular velocity of the spinning body and any other bodies in the system is either very large or very small.

  “Very fast” means the spinning body must complete several rotations within its thermal relaxation time for this model to be applicable.
- If you want to include radiation modeling in the simulation, interface boundaries and wall boundaries between the rotating region and the surroundings must be opaque to radiation. (The model does not allow radiation from a non-rotating body to pass through the boundaries of a rotating body.)

For more information, see Circumferential Heat Flux Averaging Formulation.

Contents:

Circumferential Heat Flux Averaging Workflow

Circumferential Heat Flux Averaging Workflow

You can use Circumferential Heat Flux Averaging for any **Physics** node in the simulation tree, if the conditions for the model apply.

There are two methods, both based on the Circumferential Flux Averaging model:

External Boundary

Use this method in the case of a solid or fluid body spinning rapidly in a region without mesh, with no medium between the body and its environment, represented by the boundary of the region beyond the meshless one.

Interface

Use this method in the case of a solid body spinning rapidly in an interfacing fluid medium.

Use the following steps:

1. Select any energy model, but do not select any of these incompatible models:
   - Eulerian models
   - Harmonic Balance models
   - The Fluid Film model
   - The PISO Unsteady model
2. Select the **Circumferential Flux Averaging** model from the *Optional Models* group box.

   The model has no user-defined properties.
3. Set up a rotating reference frame for the spinning region.

   For the External Boundary case, see A Body Spinning in a Meshless Region.
   For the Interface case, see A Body Spinning in an Interfacing Medium.
4. Set **Physics Conditions**:

   For the External Boundary case, open the **Regions** > **Physics Conditions** node for the rotating region.
   For the Interface case, open the **Interface** > **Physics Conditions** node for the interface being averaged.
5. Open the **Physics Values** node for the boundary.
6. Select **Axial Strip Specification Method** and, in the *Properties* window, enter a value for the *Number of Axial Strips*.
7. If you are using an unsteady time mode, set the time-step to be at least as long as one complete rotation of the rotating frame. Accuracy improves as the number of rotations per time-step increases. If the time-step is too short, an error message appears on the console.

Run the simulation.

Energy Reference

This section provides reference information that is common to the energy models in STAR-CCM+.

The common information that is covered here includes:

- Conditions (boundary and region)
- Values (boundary and region)

Static temperature is the initial condition available for energy models in STAR-CCM+.

Contents:

Conditions
Values

Conditions

This section lists all the boundary conditions available with the energy models in STAR-CCM+:

- Thermal specification
- User wall heat flux coefficient specification

It also lists region conditions:

- Energy source option
- Convective Velocity Option

In addition, detailed property information is provided.

Boundary conditions are elected by the boundary type, which in turn are limited by the models that have been turned on in the continuum. The boundary types that are available for each of the energy models are listed in the Boundary Types section for the relevant model.

Contents:

Thermal Specification
User Wall Heat Flux Coefficient Specification
Energy Source Option
Convective Velocity Option

Thermal Specification

This node allows you to determine how the energy flow across the boundary is specified.

Properties

Method	The method that is used to define the heat flow at the boundary.

	Heat Flux	Specifies the amount of energy flowing across the boundary in W/m^2. Activating this option adds a Heat Flux node under the **Physics Values** node, which must be entered as a scalar profile.
	Heat Source	Specifies a user-defined total heat source in W. Activating this option adds a Heat Source node under the **Physics Values** node, which must be entered as a scalar profile.
	Temperature	Specifies the boundary temperature in K. Activating this option adds a Temperature node under the **Physics Values** node, which must be entered as a scalar profile.
	Adiabatic	Specifies that there is no energy transfer across the boundary. Activating this option does not require definition of any values.
	Convection	Specifies convection flux across the boundary in W/m^2. Activating this option adds Ambient Temperature and Heat Transfer Coefficient nodes to the **Physics Values** node, which must be entered as scalar profiles.
	Environment	Enables the Environment condition across the boundary. Activating this option adds the following nodes, which generally must be entered as scalar profiles, to the **Physics Values** node: • Ambient Temperature • External Emissivity • External Reflectivity • Heat Transfer Coefficient • Thermal Resistance

User Wall Heat Flux Coefficient Specification

This node allows you to control the heat flux relationship at walls and interfaces.

The wall heat flux $\dot{\mathbf{q}}''$ is linearized as:

$$\dot{\mathbf{q}}'' = A + BT_c + CT_w + D\left(T_w\right)^4 \tag{308}$$

where T_c and T_w are the cell and wall temperatures, respectively. A, B, C, and D are the net wall heat flux coefficients.

In the absence of radiation, the heat flux from the wall into the domain is given by $k\nabla T_w$, where ∇T_w is the temperature gradient at the wall.

This is discretized as described by Eqn. (1593):

$$D_f = \Gamma_f \nabla\phi_f \cdot \mathbf{a} = \Gamma_f\left[\left(\phi_1 - \phi_0\right)\vec{\alpha} \cdot \mathbf{a} + \overline{\nabla\phi} \cdot \mathbf{a} - \left(\overline{\nabla\phi} \cdot \mathbf{ds}\right)\vec{\alpha} \cdot \mathbf{a}\right]$$

where:

- $\Gamma_f = k$
- $\nabla\phi_f = \nabla T_w$
- $\nabla\phi_0 = \nabla T_c$ (the temperature gradient in the near-wall cell)
- $\phi_f = T_w$
- $\phi_0 = T_c$

Prior to taking the dot product with area a, this expression is:

$$k\nabla T = k\left[\left(T_w - T_c\right)\vec{\alpha} + \nabla T_c - \left(\nabla T_c \cdot \mathbf{ds}\right)\vec{\alpha}\right]$$

which can be rearranged to:

$$k\nabla T = k\left[\nabla T_c - \left(\nabla T_c \cdot \mathbf{ds}\right)\vec{\alpha}\right] - k\vec{\alpha}T_c + k\vec{\alpha}T_w + 0{T_w}^4$$

so the internal heat flux coefficients are:

- $A_{\text{internal}} = k\left[\nabla T_c - \left(\nabla T_c \cdot \mathbf{ds}\right)\vec{\alpha}\right]$
- $B_{\text{internal}} = -k\vec{\alpha}$
- $C_{\text{internal}} = k\vec{\alpha}$
- $D_{\text{internal}} = 0$

User Wall Heat Flux Coefficient Specification

The User Wall Heat Flux Coefficient Specification is available in both single phase and multiphase simulations. However, there are some differences between the single phase and multiphase implementations. For information on User Wall Heat Flux Coefficient Specification for multiphase simulations, see Multiphase User Wall Heat Flux Coefficient Specification.

Single Phase Implementation

For single phase simulations, the net wall heat flux coefficients A, B, C, and D are composed of the sum of internally calculated and user-defined coefficients:

$$\begin{aligned} A &= A_{\text{internal}} + A_{\text{user}} \\ B &= B_{\text{internal}} + B_{\text{user}} \\ C &= C_{\text{internal}} + C_{\text{user}} \\ D &= D_{\text{internal}} + D_{\text{user}} \end{aligned} \tag{309}$$

The internal wall heat flux coefficients are evaluated first, in the usual manner according to the active physical processes (for example conduction, radiation and/or boiling). Then the net coefficients are evaluated by summing the internal and user-defined values.

Multiphase Implementation

In Multiphase Segregated Flow simulations, the net wall heat flux coefficients A, B, C, and D are simply the user-defined coefficients:

$$\begin{aligned} A &= A_{\text{user}} \\ B &= B_{\text{user}} \\ C &= C_{\text{user}} \\ D &= D_{\text{user}} \end{aligned} \tag{310}$$

The internal coefficients can be included as necessary in the user-defined coefficients using the field functions:

A_{internal} = `$InternalWallHeatFluxCoefficientA ("Internal Wall Heat Flux Coefficient, A")`

B_{internal} = `$InternalWallHeatFluxCoefficientB ("Internal Wall Heat Flux Coefficient, B")`

C_{internal} = `$InternalWallHeatFluxCoefficientC ("Internal Wall Heat Flux Coefficient, C")`

D_{internal} = `$InternalWallHeatFluxCoefficientD ("Internal Wall Heat Flux Coefficient, D")`

So, for example, you can completely replace net coefficient A with some function f by defining:

$$A_{\text{user}} = f - A_{\text{internal}} \tag{311}$$

such that the net coefficient becomes:

$$A = A_{\text{internal}} + \left(f - A_{\text{internal}}\right) = f \tag{312}$$

The resultant net wall heat flux coefficients are available using the field functions:

A = `$NetWallHeatFluxCoefficientA ("Net Wall Heat Flux Coefficient, A")`

B = `$NetWallHeatFluxCoefficientB ("Net Wall Heat Flux Coefficient, B")`

C = `$NetWallHeatFluxCoefficientC ("Net Wall Heat Flux Coefficient, C")`

D = `$NetWallHeatFluxCoefficientD ("Net Wall Heat Flux Coefficient, D")`

The user-defined heat flux coefficients do not affect the amount of heat that enters the domain. These coefficients determine how much of the heat flux goes towards increasing the wall temperature, and how much goes towards increasing the cell temperature.

User-defined wall heat flux coefficients can be enabled on Wall or Contact Interface boundaries by setting the *Method* property of this node.

User Wall Heat Flux Coefficient Specification Properties

Method	Controls whether to specify the heat flux relationship.	
	None	Leaves off this feature. The User Wall Heat Flux Coefficient is not specified.
	Specified	Activates this feature. This option adds the following nodes to the **Physics Values** node, which must be entered as scalar profiles: • User Wall Heat Flux Coefficient, A • User Wall Heat Flux Coefficient, B • User Wall Heat Flux Coefficient, C • User Wall Heat Flux Coefficient, D The user specified coefficients are added to the internally calculated net coefficients as given by Eqn. (309).
	Specified Net Coefficients	Sets the net boundary heat flux coefficients equal to the user specified heat flux coefficients as given by Eqn. (310). This method is available in Multiphase Segregated Flow simulations only.

Energy Source Option

Properties

Unless otherwise specified, the conditions below apply to fluid, porous, and solid regions.

Energy Source Option

Provides energy source options for the region.

None

Do not specify an energy source.

Volumetric Heat Source

Specifies a user-defined volumetric heat source in W/m^3.

Activating this option when using the Segregated solver adds the following nodes under the **Physics Values** node:

- Volumetric Heat Source
- Energy Source Temperature Derivative

Activating this option when using the Coupled solver adds the following nodes under the **Physics Values** node:

- Volumetric Heat Source
- Energy Source Pressure Derivative
- Energy Source Temperature Derivative
- Energy Source Velocity Derivative

Total Heat Source

Specifies a user-defined total heat source in W.

Activating this option when using the Segregated solver adds the following nodes under the **Physics Values** node:

- **Heat Source**
- **Energy Source Temperature Derivative**

Activating this option when using the Coupled solver adds the following nodes under the **Physics Values** node:

- **Heat Source**
- **Energy Source Pressure Derivative**
- **Energy Source Temperature Derivative**
- **Energy Source Velocity Derivative**

Specific Heat Source

Specifies a user-defined specific heat source in W/kg.

Activating this option when using the Segregated solver or the Coupled solver adds the following nodes under the **Physics Values** node:

- **Specific Heat Source**
- **Energy Source Temperature Derivative**

Heat Exchanger

Specifies a heat exchanger source term.

Activating this option when using the Segregated solver or the Coupled solver adds the following node under the **Physics Conditions** node:

- **Heat Exchanger Exit Temperature Specification**

Activating this option when using the Segregated solver or the Coupled solver adds the following nodes under the **Physics Values** node:

- **Heat Exchanger Minimum Temperature Difference**
- **Heat Exchanger Energy Transfer**
- **Heat Exchanger Boundaries**
- **Heat Exchanger First Iteration**
- **Heat Exchanger Temperature**

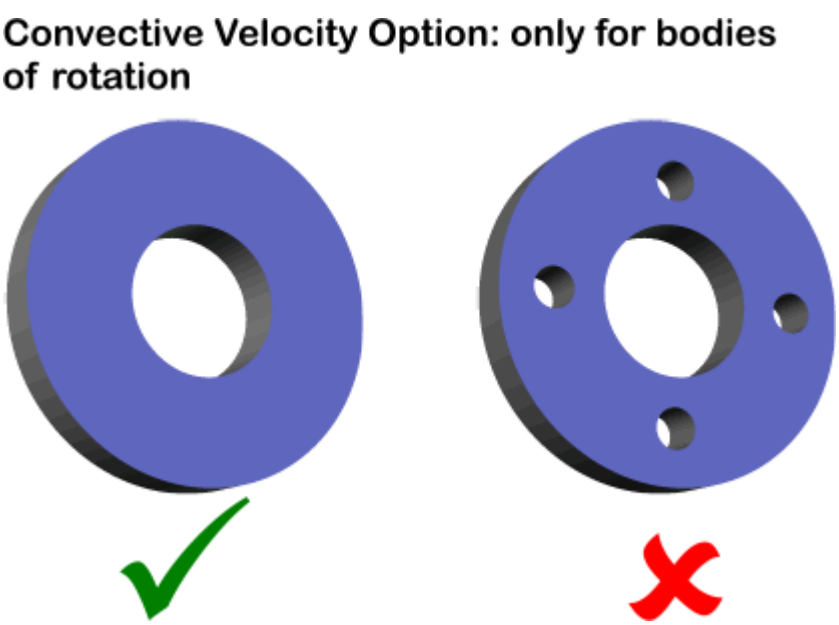
This option is only available with fluid and porous regions.

Convective Velocity Option

This node, which appears in solid regions when either of the Segregated Solid Energy or Coupled Solid Energy models are enabled, allows you to activate a convective velocity for simulating energy convection within the solid region.

(See $\mathbf{v}_s$ in Eqn. (349).) The specified velocity field is a field of pure rotation or translation, and is intended for use in steady cases only. It is recommended that the solid energy under-relaxation factor be reduced to a value between 0.5 and 0.6 when convection is included.

This option should only be used when the region is a pure body of rotation or translation. This is not the case when holes or grooves are present, as illustrated below. All free surfaces should be tangential to the direction of rotation or translation.



Properties

Convective Velocity Option	Provides a convective velocity field for the region.	
	None	Do not specify a convective velocity.
	Specified	Specify a convective velocity. A Convective Velocity node is added to the **Values** node.

Values

This section lists all the boundary values that are available for energy models in STAR-CCM+ and it provides a detailed reference for the methods of setting them.

- Ambient temperature
- External emissivity
- External reflectivity
- Heat flux
- Heat transfer coefficient
- Static temperature
- Thermal resistance
- Total temperature
- User wall heat flux coefficient, A
- User wall heat flux coefficient, B
- User wall heat flux coefficient, C
- User wall heat flux coefficient, D

It also lists region values:

- Anisotropic thermal conductivity
- Volumetric heat source
- Heat source
- Specific heat source
- Energy source pressure derivative
- Energy source temperature derivative
- Energy source velocity derivative
- Solid density
- Solid specific heat
- Solid thermal conductivity
- Convective Velocity

Boundary values are elected by the boundary type, and/or by methods that are selected in corresponding nodes in the boundary conditions node. The boundary types that are available are listed in the Boundary Type section of each model.

Contents:

Ambient Temperature
External Emissivity
External Reflectivity
Heat Flux
Heat Transfer Coefficient
Static Temperature
Thermal Resistance
Total Temperature
User Wall Heat Flux Coefficient, A
User Wall Heat Flux Coefficient, B
User Wall Heat Flux Coefficient, C
User Wall Heat Flux Coefficient, D
Anisotropic Thermal Conductivity
Volumetric Heat Source
Heat Source
Specific Heat Source
Energy Source Pressure Derivative
Energy Source Temperature Derivative
Energy Source Velocity Derivative
Solid Density
Solid Specific Heat
Solid Thermal Conductivity
Convective Velocity

Ambient Temperature

This value node is visible if Convection or Environment is selected as the *Method* property for the Thermal Specification node of the **Conditions** node. It is input as a scalar profile.

External Emissivity

Certain energy and radiation models use this boundary value. It is only visible if the Environment method is chosen as the Thermal Specification and is entered as a scalar profile.

External Reflectivity

Certain energy and radiation models use this boundary value.

It is only visible if the Environment method is chosen as the Thermal Specification. It can be entered as a scalar profile or can be calculated automatically using the Auto-Calculate method, which is available for reflectivity.

Heat Flux

This value node is only visible if Heat Flux is selected as the *Method* property for the Thermal Specification node of the **Conditions** node. It is input as a scalar profile in W/m^2.

Specify a positive heat flux value for heat flowing into the domain, or a negative value for heat flowing out.

Heat Transfer Coefficient

Energy models require this boundary value. It is visible if the Convection or Environment method is chosen as the Thermal Specification and is entered as a scalar profile.

Static Temperature

The temperature of the boundary. This value node is visible if Temperature is selected as the *Method* property for the Thermal Specification node of the **Conditions** node. It is entered as a scalar profile.

Thermal Resistance

This boundary value appears when the Environment method is chosen as the Thermal Specification. It functions in a manner similar to the Thermal Resistance baffle interface value.

Total Temperature

The total temperature of the boundary. It is entered as a scalar profile. Not available in regions of type solid.

See Also:

Stagnation Inlet

User Wall Heat Flux Coefficient, A

This boundary value is available with energy models. It is only visible if the Specified method is chosen as the User Wall Heat Flux Coefficient Specification and is entered as a scalar profile. By default it has a constant value of zero.

User Wall Heat Flux Coefficient, B

This boundary value is available with energy models. It is only visible if the Specified method is chosen as the User Wall Heat Flux Coefficient Specification and is entered as a scalar profile. By default it has a constant value of zero.

User Wall Heat Flux Coefficient, C

This boundary value is available with energy models. It is only visible if the Specified method is chosen as the User Wall Heat Flux Coefficient Specification and is entered as a scalar profile. By default it has a constant value of zero.

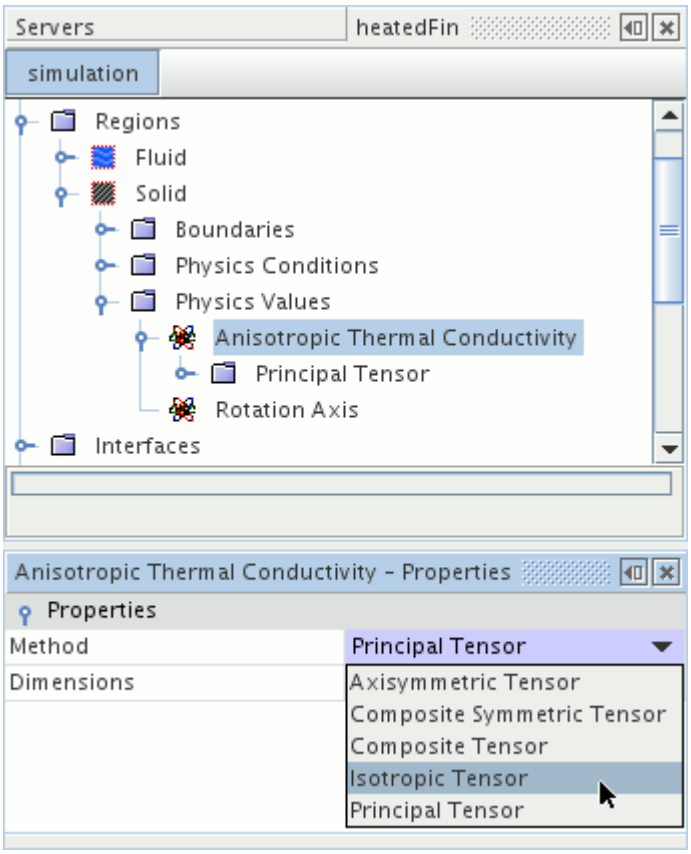
User Wall Heat Flux Coefficient, D

This boundary value is available with energy models. It is only visible if the Specified method is chosen as the User Wall Heat Flux Coefficient Specification and is entered as a scalar profile. By default it has a constant value of zero.

Anisotropic Thermal Conductivity

This value node is added to a solid region when the thermal conductivity material property of a solid is set to the Anisotropic method.

It is entered as a tensor profile. The method of the tensor profile is set by selecting the **Anisotropic Thermal Conductivity** node and using the drop-down menu in the *Properties* window, as shown below.



Volumetric Heat Source

This node is added when the Energy Source Option region condition is set to **Volumetric Heat Source**.

It is entered as a scalar profile.

Heat Source

This value specifies the total heat source in W. It is available when you activate either the **Heat Source** Energy Source Option for a region, or the **Heat Source** Thermal Specification for a boundary.

Specific Heat Source

This node is added when the Energy Source Option region condition is set to **Specific Heat Source**. It is entered as a scalar profile.

This value specifies the heat source in W/kg.

Energy Source Pressure Derivative

This node represents the derivative of the Energy Source with respect to pressure.

See S_u in Eqn. (1650). The derivative of the user source term is used to linearize Eqn. (1650). Its value is set to zero by default. Providing a value for the derivative helps stabilize the solution when the function for the source term is a function of pressure. If the source is constant, or not a function of pressure, leave this value at zero.

The Energy Source Pressure Derivative is available when coupled flow is used. This object is added when the Energy Source Option region condition is set to Specified, and is entered as a scalar profile.

Energy Source Temperature Derivative

This node represents linearization of the Energy Source with respect to temperature.

See S_u in Eqn. (1650) (coupled) and $s_u dV$ in [eqnlink] (segregated). The derivative of the user source term is used to linearize these equations. Its value is set to zero by default. Providing a value for the derivative helps stabilize the solution when the function for the source term is a function of temperature. If the source is constant, or not a function of temperature, leave this value at zero.

The Energy Source Temperature Derivative is available when either coupled or segregated flow is used. This object is added when the Energy Source Option region condition is set to Specified, and is entered as a scalar profile.

Energy Source Velocity Derivative

This node represents linearization of the Energy Source with respect to x-, y-, [z-] components of velocity.

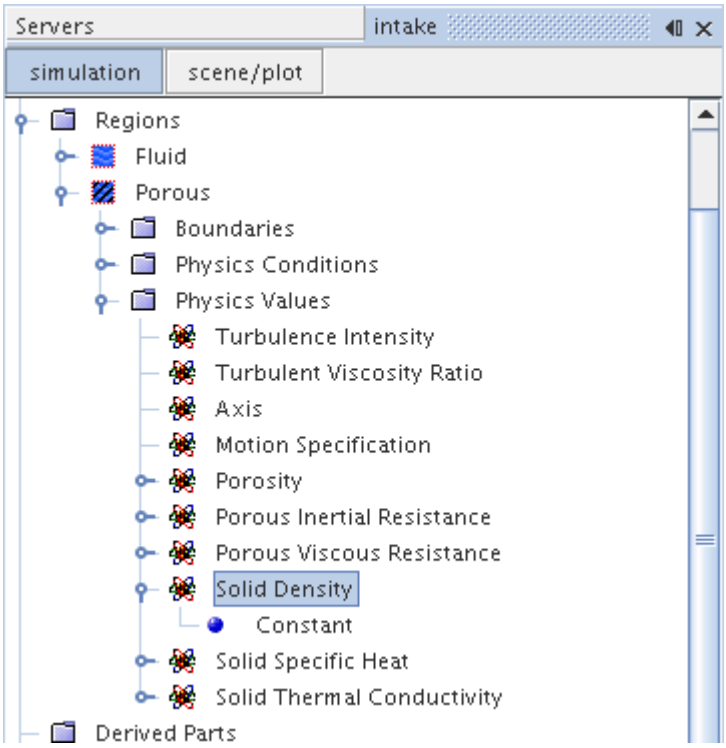
See S_u in Eqn. (1650). The derivative of the user source term is used to linearize Eqn. (1650). Its value is set to zero by default. Providing a value for the derivative helps stabilize the solution when the energy source is a function of velocity. If the source is constant, or not a function of velocity, leave this value at zero.

The Energy Source Velocity Derivative is available when coupled flow is used. This object is added when the Energy Source Option region condition is set to Specified, and is entered as a vector profile.

Solid Density

This node is only available in regions of the porous type.

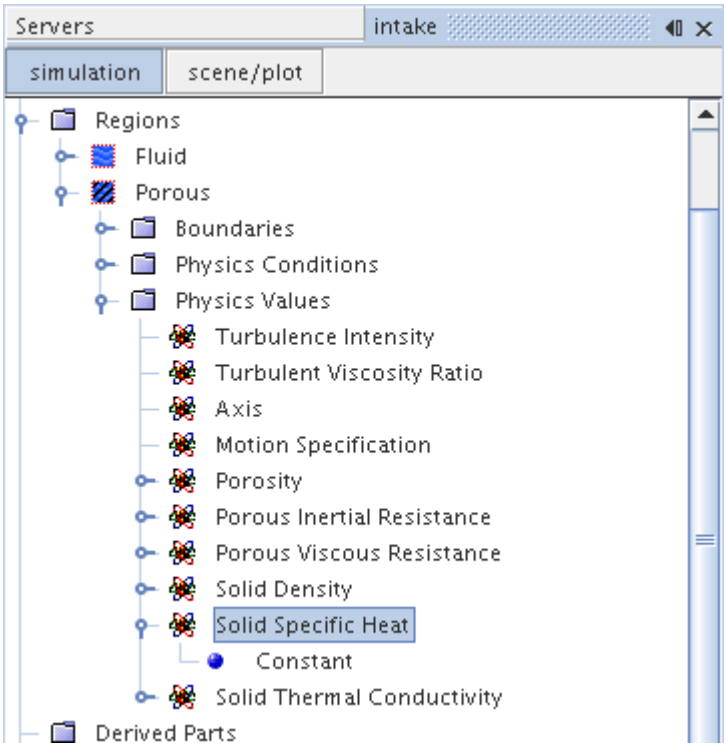
The density that is provided here is used to specify the solid density ρ_{solid} in Eqn. (2186). The fluid density ρ_{fluid} in Eqn. (2186) is the density property that is specified for the fluid continuum. It is entered as a scalar profile.



Solid Specific Heat

This node is only available in regions of the porous type.

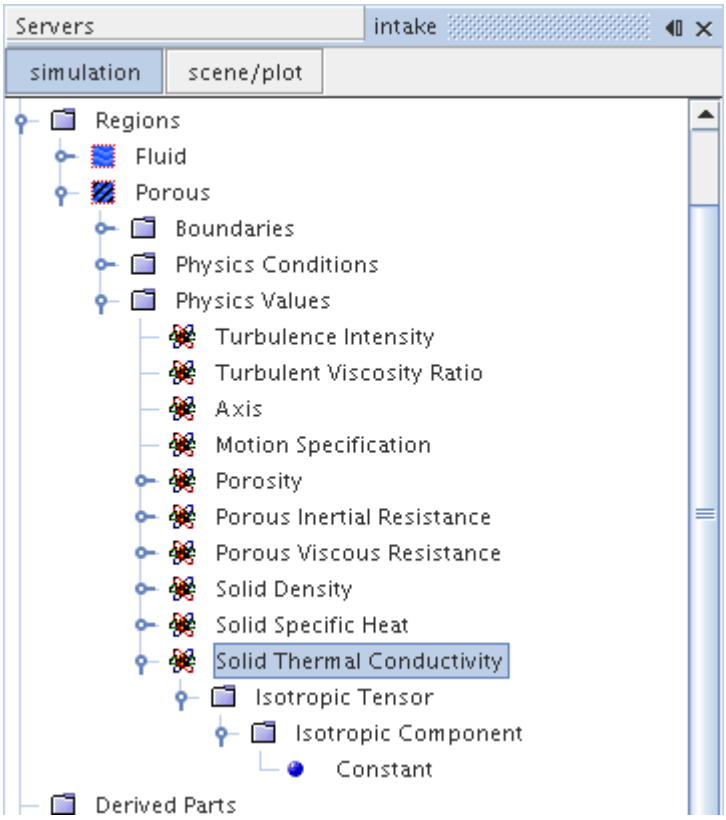
The specific heat that is provided here is used to specify the solid specific heat $C_{p\text{-}solid}$ in Eqn. (2186). It is entered as a scalar profile.



Solid Thermal Conductivity

This node is only available in regions of the porous type.

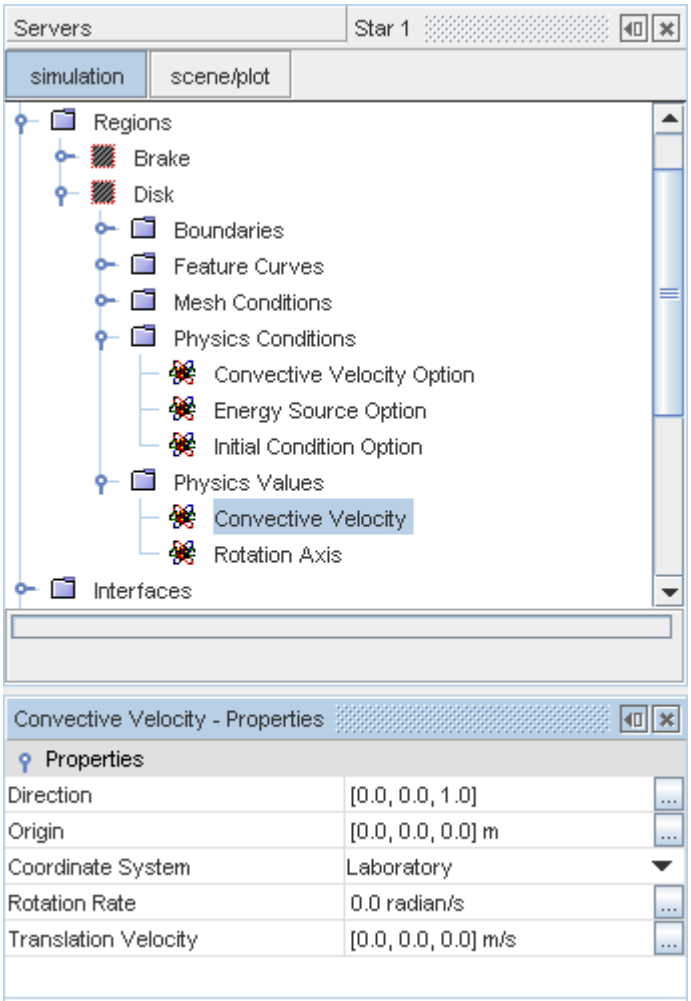
The thermal conductivity that is provided here is used to specify k_{solid} in Eqn. (2188). The fluid thermal conductivity k_{fluid} in Eqn. (2188) is extracted from the property that is specified for the fluid continuum. It is entered as a tensor profile, and it is possible to use field functions and other profile methods to define the components of the tensor.



Convective Velocity

This node, which has properties, is added when the Convective Velocity Option for a solid region has been set to **Specified**.

Rotation is specified by providing a rotation axis, an origin for that axis, and a rotation rate. Translation is specified by providing the translation velocity. User-defined coordinate systems can be used when specifying quantities.



Properties

Direction	Provides a convective velocity field for the region.
Origin	The Cartesian position vector specifying the origin of the axis of rotation.
Coordinate System	Specifies the coordinate system (laboratory by default) for defining the origin and axis. The drop-down list of this property includes any local coordinate systems that exist in the simulation.
Rotation Rate	The rotation rate around the axis that is specified by *Direction* and *Origin*. This is additive with *Translation Velocity*.
Translation Velocity	The translation velocity of the region. This is additive with *Rotation Rate*.

Heat Transfer Field Functions Reference

The following primitive field functions are made available to the simulation when you activate either Coupled Energy or Segregated Fluid Energy in a physics continuum.

These objects all have the same base properties, with some field functions also having reference value properties (indicated by a ref subscript in the following descriptions).

In the case of boundary heat flux functions, positive values mean heat is flowing out of the domain. This is opposite to the convention used with the Heat Flux boundary and region values.

Boundary Advection Heat Flux

The magnitude of the advection heat flux vector normal to the boundary. The quantity includes the energy that is carried by the fluid across the domain boundary and is only defined at flow openings. Consistent with the other heat fluxes, the sign convention is positive for transfer out of the domain.

A version of this field function is created for each phase in a multiphase continuum.

Boundary Advection Heat Transfer

The heat transferred by advection at the boundary. The quantity is essentially the rate that energy is advected into and out of the domain by the moving fluid. Consistent with the other heat transfer rates, the sign convention is positive for transfer leaving the domain.

A version of this field function is created for each phase in a multiphase continuum.

Boundary Conduction Heat Flux

The magnitude of the conduction heat flux vector normal to the boundary, defined as $\left(\dot{\mathbf{q}}''_{\text{conduction}} \cdot \mathbf{a}\right) / |\mathbf{a}|$.

This quantity includes molecular and turbulent diffusion effects at the boundary (positive denotes transfer out of the domain). The advection contribution at open flow boundaries is separately included in the Boundary Advection Heat Flux.

For a multiphase fluid, the boundary conduction heat flux is the sum of the boundary conduction heat fluxes for each phase. This summation is valid because the phase-specific coefficients are already volume fraction weighted.

Boundary Conduction Heat Flux of <phase>

The magnitude of the conduction heat flux vector normal to the boundary for the specified phase. A version of this field function is created for each phase in a multiphase continuum.

This field function is volume fraction weighted for each phase.

Boundary Conduction Heat Transfer

The heat transferred by conduction at the boundary, defined as $\dot{\mathbf{q}}''_{\text{conduction}} \cdot \mathbf{a}$.

This quantity includes molecular and turbulent diffusion effects at the boundary (positive denotes transfer out of the domain). The advection contribution at open flow boundaries is separately included in the Boundary Advection Heat Transfer.

For a multiphase fluid, the boundary conduction heat transfer is the sum of the boundary heat transfer values for each phase. This summation is valid because the phase-specific coefficients are already volume fraction weighted.

Boundary Conduction Heat Transfer of <phase>

The heat transferred by conduction at the boundary for the specified phase. A version of this field function is created for each phase in a multiphase continuum.

This field function is volume fraction weighted for each phase.

Boundary Heat Flux

The magnitude of the heat flux vector normal to the boundary, defined as $(\dot{\mathbf{q}}'' \cdot \mathbf{a})/|\mathbf{a}|$, summed from all forms of heat flux—boundary conduction, convection, and radiation.

For a multiphase fluid, the boundary heat flux is the sum of the boundary heat fluxes for each phase. This summation is valid because the phase-specific coefficients are already volume fraction weighted. When the VOF model is used, this sum includes the wall boiling heat flux.

For harmonic balance cases, the value of the heat flux is a time-mean.

Boundary Heat Flux of <phase>

The magnitude of the heat flux vector normal to the boundary for the specified phase. A version of this field function is created for each phase in a multiphase continuum.

This field function is volume fraction weighted for each phase.

Boundary Heat Flux Radiation Coefficient

The boundary heat flux radiation coefficient, defined by $\sigma\varepsilon$, where σ is the Stefan-Boltzmann constant and ε is the total emissivity. This quantity can be used in external CAE packages to express the radiative emission flux in terms of the boundary temperature:

$$\dot{\mathbf{q}}''_{\text{rad, emission}} = D\,T^4{}_w$$

where D is the boundary heat flux radiation coefficient.

Boundary Heat Transfer

The heat transferred at the boundary, defined as $\dot{\mathbf{q}}'' \cdot \mathbf{a}$.

For a multiphase fluid, the boundary heat transfer is the sum of the boundary heat transfer values for each phase. This summation is valid because the phase-specific coefficients are already volume fraction weighted.

For harmonic balance cases, the value of the heat transfer is a time-mean.

Boundary Heat Transfer of <phase>

The heat transferred at the boundary for the specified phase. A version of this field function is created for each phase in a multiphase continuum.

This field function is volume fraction weighted for each phase.

Effective Conductivity

The effective thermal conductivity given by $k_{\text{eff}} = k + \left(\mu_t C_p\right)/\sigma_t$ where σ_t is the turbulent Prandtl number.

For porous regions this field function is given by

$$k_{\text{eff}} = \alpha\left[k + \left(\mu_t C_p\right)/\sigma_t\right] + (1-\alpha)k_{solid}$$

External Ambient Temperature

The boundary ambient temperature that you specify for convection thermal boundary conditions. This field function is only available on boundaries for which the Convection option has been selected as the *Thermal Specification*.

External Heat Transfer Coefficient

The boundary external heat transfer coefficient that you specify for convection thermal boundary conditions. This field function is only available on boundaries for which the Convection option has been selected as the *Thermal Specification*.

Heat Transfer Coefficient

The wall boundary heat transfer coefficient defined by:

$$h = \left(\dot{\mathbf{q}}''_{\text{conduction}} \cdot \mathbf{a}\right)/\left[|\mathbf{a}|\left(T_{\text{ref}} - T_f\right)\right]$$

where T_f is the wall boundary temperature and T_{ref} is the specified reference temperature in the field function.

For a multiphase fluid, the heat transfer coefficient is defined using the volume-fraction-weighted version of the **Boundary Conduction Heat Flux** field function.

For harmonic balance cases, the value of the coefficient is a time-mean.

Local Heat Transfer Coefficient

The local heat transfer coefficient is the heat transfer coefficient between the near-wall cell and the boundary. It is dependent upon the distance between the boundary and the near-wall cell, the thermal properties of the media and, in the case of turbulent flow, the near-wall turbulence state. Because of the dependence on the mesh, do not consider this quantity as a bulk heat transfer coefficient.

For a multiphase fluid, the local heat transfer coefficient is the sum of the local heat transfer coefficients for each phase. This summation is valid because the phase-specific coefficients are already volume fraction weighted.

For harmonic balance cases, the value of the coefficient is a time-mean.

Local Heat Transfer Coefficient of <phase>

The local heat transfer coefficient of the specified phase. A version of this field function is created for each phase in a multiphase continuum.

This field function is volume fraction weighted for each phase.

Local Heat Transfer Reference Temperature

The local heat transfer reference temperature is the effective temperature that can be used to linearize the total heat flux to the boundary with respect to the boundary temperature: $\dot{\mathbf{q}}''_{\text{total}} = h_{\text{local}}\left(T_{\text{ref}} - T_w\right)$, where h_{local} is the local heat transfer coefficient and T_w is the boundary temperature. The effect of radiation is explicitly included in this quantity. This quantity is useful for export to external CAE packages with the local heat transfer coefficient.

For a multiphase fluid, the local heat transfer reference temperature is given by:

$\sum(h_i \times Tr_i) / \sum h_i$

where h_i is the local heat transfer coefficient of phase i and Tr_i is the local heat transfer reference temperature of phase i .

For harmonic balance cases, the value of the temperature is a time-mean.

Local Heat Transfer Reference Temperature of <phase>

The local heat transfer reference temperature of the specified phase. A version of this field function is created for each phase in a multiphase continuum.

This field function is not volume fraction weighted.

Nusselt Number

The non-dimensional heat transfer coefficient is given by

$$Nu = \frac{\dot{q}''_{wall}}{T_{ref} - T_{wall}} \frac{L_{ref}}{k_{ref}}$$

where $\dot{q}''_{wall}$ is the wall heat flux, k is the thermal conductivity and L_{ref} is the specified reference length in the continuum.

Porous Solid Conductivity

Solid conductivity that is valid only for porous regions. This field function reflects the data that you enter as part of the Solid Thermal Conductivity physics value.

Relative Total Temperature

The temperature obtained by isentropically bringing the flow to rest in the relative frame of motion.

Specified Y+ Heat Transfer Coefficient

The specified Y+ heat transfer coefficient is calculated in the same way as the local heat transfer coefficient (see Eqn. (303)), but at a user-specified y^+ value instead of the value that is associated with the near-wall cell.

For a multiphase fluid, the specified Y+ heat transfer coefficient is the sum of the specified Y+ heat transfer coefficients for each phase. This summation is valid because the phase-specific coefficients are already volume fraction weighted.

For harmonic balance cases, the value of the coefficient is a time-mean.

To make this field function available, the Turbulence models must be activated.

Specified Y+ Heat Transfer Coefficient of <phase>

The specified Y+ heat transfer coefficient of the specified phase. A version of this field function is created for each phase in a multiphase continuum.

This field function is volume fraction weighted for each phase.

To make this field function available, the Turbulence models must be activated.

Specified Y+ Heat Transfer Reference Temperature

The specified Y+ heat transfer reference temperature as given by Eqn. (304).

For a multiphase fluid, the specified Y+ heat transfer reference temperature is given by:

$$\sum\left(h_i \times Tr_i\right) / \sum h_i$$

where h_i is the specified Y+ heat transfer coefficient of phase i and Tr_i is the specified Y+ heat transfer reference temperature of phase i .

For harmonic balance cases, the value of the temperature is a time-mean.

To make this field function available, the Turbulence models must be activated.

Specified Y+ Heat Transfer Reference Temperature of <phase>

The specified Y+ heat transfer reference temperature of the specified phase. A version of this field function is created for each phase in a multiphase continuum.

This field function is not volume fraction weighted.

To make this field function available, the Turbulence models must be activated.

Temperature of External Side

The temperature of the external side of a boundary between the environment and an internal domain. Available only when an environment condition is applied.

Temperature Residual

This residual becomes available in simulations involving heat transfer for solids when the Energy solver has the **Temporary Storage Retained** property activated.

Thermal Conductivity

Scalar value that reflects this material property according to the method that you defined for the physics continuum.

Note: This field function is a scalar value and therefore not populated in regions where thermal conductivity is defined by an anisotropic tensor.

Thermal Conductivity Tensor <nn>

Any of six components of the symmetric tensor describing thermal conductivity of solids in anisotropic regions. <nn> can be XX, YY, ZZ, XY, XZ, or YZ.

Total Enthalpy

The total enthalpy H is the sum of static enthalpy plus kinetic energy, $H = h + \frac{v^2}{2}$.

Total Temperature

The total temperature is the temperature that is obtained by isentropically bringing the flow to rest. For an ideal gas with constant specific heat, it is defined as:

$$T_t = T\left(1 + \frac{(\gamma-1)}{2}M^2\right)$$

where T , M , and γ are the static temperature, Mach number, and ratio of specific heats, respectively.

Wall Boiling Heat Flux

The magnitude of a heat flux vector normal to the wall, expressing an estimate of how much the conductive heat transfer increases due to boiling at the wall boundary.

Energy Formulation

This section contains the following sub-sections on formulations used in modeling energy:

- Single-Phase Boiling Formulation
- Thin Film Formulation
- Single-Stream Heat Exchanger Formulation
- Dual-Stream Heat Exchanger Formulation
- Circumferential Heat Flux Averaging Formulation
- Solid Energy Formulation

Contents:

Single-Phase Boiling Formulation
Thin Film Formulation
Single-Stream Heat Exchanger Formulation
Dual-Stream Heat Exchanger Formulation
Circumferential Heat Flux Averaging Formulation
Solid Energy Formulation

Single-Phase Boiling Formulation

There are many models of nucleate-boiling heat transfer, and none of them is accepted as a standard, more-or-less reliable model.

The main mechanisms which these models try to mimic are summarized below:

- Evaporation (latent-heat) transport—Heat is supplied near the wall to bubbles which then move away into the bulk liquid. In subcooled boiling, there could be simultaneous evaporation at the base of a bubble and condensation at its tip.
- Micro-convection—Growth and collapse of bubbles in the wall proximity cause random liquid motion.
- Vapor-liquid exchange—A “Reynolds analogy” model in which bubble growth causes an exchange of liquid between the wall and bulk regions.
- Surface quenching—A variation of the vapor-liquid exchange, assuming transient conduction to the cold liquid contacting the wall after bubble departure.
- Wake flow—Liquid motion behind a departing bubble causes convection from the wall.
- Enhanced natural convection—Bubble columns produce a cellular flow pattern similar to natural convection above a large horizontal surface.
- Thermo-capillary flow—Small variations in surface tension due to temperature differences between the base and tip of a bubble cause a jet of hot liquid to flow away from the wall.

The empirical correlation that Rohsenow [601] presented is used to calculate the surface heat flux due to boiling:

$$q_{bw} = \mu_l h_{lat} \sqrt{\frac{g(\rho_l - \rho_v)}{\sigma}} \left(\frac{C_{p_l}(T_w - T_{sat})}{C_{qw} h_{lat} Pr_l^{n_p}} \right)^{3.03} \tag{313}$$

In this equation, μ_l, h_{lat}, C_{p_l}, ρ_l, and Pr_l are the dynamic viscosity, latent heat, specific heat, density, and Prandtl number of the liquid phase, n_p is the Prandtl number exponent (1.73 by default), g is gravity, and ρ_v is the vapor density, σ is the surface tension coefficient at the liquid-vapor interface, T_w is the wall temperature, T_{sat} is the saturation temperature, and C_{qw} is an empirical coefficient varying with the liquid-surface combination.

The vapor mass generation rate $\dot{m}_{ew}$ over the area that nucleation sites cover is:

$$\dot{m}_{ew} = \frac{C_{ew} q_{bw}}{h_{lat}} \tag{314}$$

where C_{ew} is a model constant stating how much of the boiling heat flux is used for creation of vapor bubbles.

If the Rohsenow correlation is applied outside its range of applicability (for example, to the film boiling regime), unrealistically high heat fluxes could result. Fluid temperatures can become higher than the near wall temperature. This behavior stems from the fact that the Rohsenow correlation does not depend on the fluid temperature; heat enters the domain irrespective of the fluid temperature.

To prevent this condition, the heat flux that the correlation calculates is multiplied by:

$$max\left[0, min\left(\left(\frac{T_w - T}{Tw - T_{sat}}\right), 1\right)\right] \tag{315}$$

where T is the fluid temperature near the heated wall. Thus, if $T < T_{sat}$, the Rohsenow correlation is used directly, and if $T > T_w$ the boiling heat flux is zero. For fluid temperatures which are between the wall and the saturation temperature, only a fraction of the heat flux predicted by the correlation is used.

Thin Film Formulation

This section includes the following formulation:

- Defogging
- Deicing

Contents:

Defogging
Deicing

Defogging

The defogging model is based on solving an additional scalar transport equation that represents the mass fraction of water vapor. A source/sink term for the scalar is considered for condensation/evaporation of the fog layer as well as the latent heat required for transition.

When there is a difference between the water vapor content at the fog layer and the cell next to this surface, the model calculates a rate of evaporation or condensation depending on the conditions. The assumptions are:

- The vapor content in the air does not affect the thermal properties of the vapor-air mixture.
- The water vapor mass is neglected with respect to the total mass in a cell.

The rate of mass transfer per unit surface $\left[\mathrm{kg/s\text{-}m^2}\right]$ is:

$$\dot{m} = \rho_g \beta_g \cdot \ CG \ \ln \frac{CG\left(1-C_s\right)}{CS\left(1-C_g\right)} \tag{316}$$

where ρ_g is density $\left[\mathrm{kg/m^3}\right]$.

$$\beta_g = C_{\mathrm{emp}} \cdot \ D_V/L \cdot \ 0.045 \cdot \ Re^{0.8} Sc^{0.43} \left[\mathrm{m/s}\right] \tag{317}$$

where C_{emp} is the empirical constant useful for calibration (0.05-0.9) for windshield applications, L is a characteristic length (cubic root of cell volume next to fog layer boundary) [m], D_V is diffusion of vapor in air $\left[\mathrm{m^2/s}\right]$, Re is the Reynolds number (which is based on the gas state and the characteristic length L), and Sc is the Schmidt number, and:

$$CG = 0.622 + 0.378 C_g \tag{318}$$

$$CS = 0.622 + 0.378 C_s \tag{319}$$

where C_g is the actual concentration of vapor in air and C_s is the saturation concentration of vapor in air.

For a given temperature, the saturation pressure is calculated from the following expression:

$$p_s\left(T\right) = 611.85 e^{\frac{17.502(T+-273.15)}{T-32.25}} \tag{320}$$

The amount of evaporated/condensed vapor in one time-step is:

$$d\dot{m} = \dot{m} \cdot \ dt \tag{321}$$

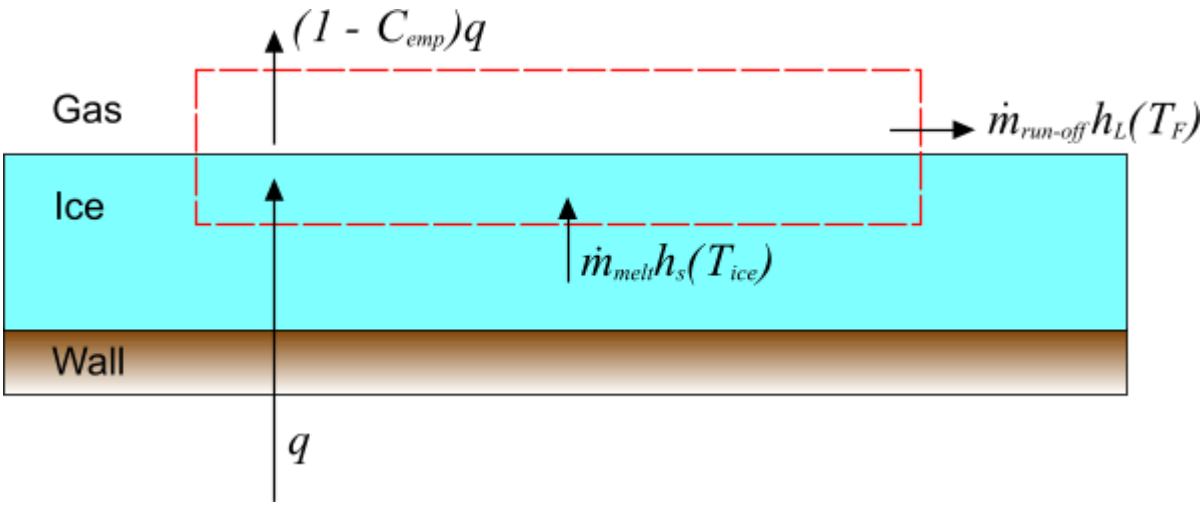
The thickness of the liquid film is updated as:

$$s = s_{t-1} + dm/\rho_l \tag{322}$$

Deicing

Deicing is assumed to be a quasi-steady process.

An ice layer at temperature T_{ice} is melted with a wall heat flux q, of which a proportion C_{emp} is absorbed in the ice layer. Melted ice at the freezing temperature T_F escapes as run-off which cannot refreeze in the current model. The mass and energy fluxes that are associated with this process are illustrated in the diagram below:



The mass balance for the deicing process is simply

$$\dot{m}_{melt} = \dot{m}_{run-off} \tag{323}$$

As it is assumed that run-off cannot refreeze, there is the additional constraint that $\dot{m}_{run-off} \geq 0$ and hence from Eqn. (323), $\dot{m}_{melt} \geq 0$.

The energy balance for this process, neglecting the kinetic energy that is associated with the mass fluxes, is:

$$\dot{m}_{melt} h_s(T_{ice}) + q = \dot{m}_{run-off} h_L(T_F) + (1 - C_{emp}) q \tag{324}$$

where $h_s(T_{ice})$ is the enthalpy of the solid ice at the temperature of the ice layer, and $h_L(T_F)$ is the enthalpy that is associated with the melting ice at the freezing temperature. These enthalpies are related to temperature by:

$$h_s(T) = C_{p,s} T \tag{325}$$

$$h_L(T) = C_{p,s} T + L \tag{326}$$

Using these relations with the mass balance, and rearranging, gives an explicit expression for the melting rate:

$$\dot{m}_{melt} = \frac{C_{emp} q}{C_{p,s}(T_F - T_{ice}) + L} \tag{327}$$

The constraint, $\dot{m}_{melt} \geq 0$, must be imposed on this equation. If Eqn. (327) predicts a negative $\dot{m}_{melt}$, it is clipped to zero.

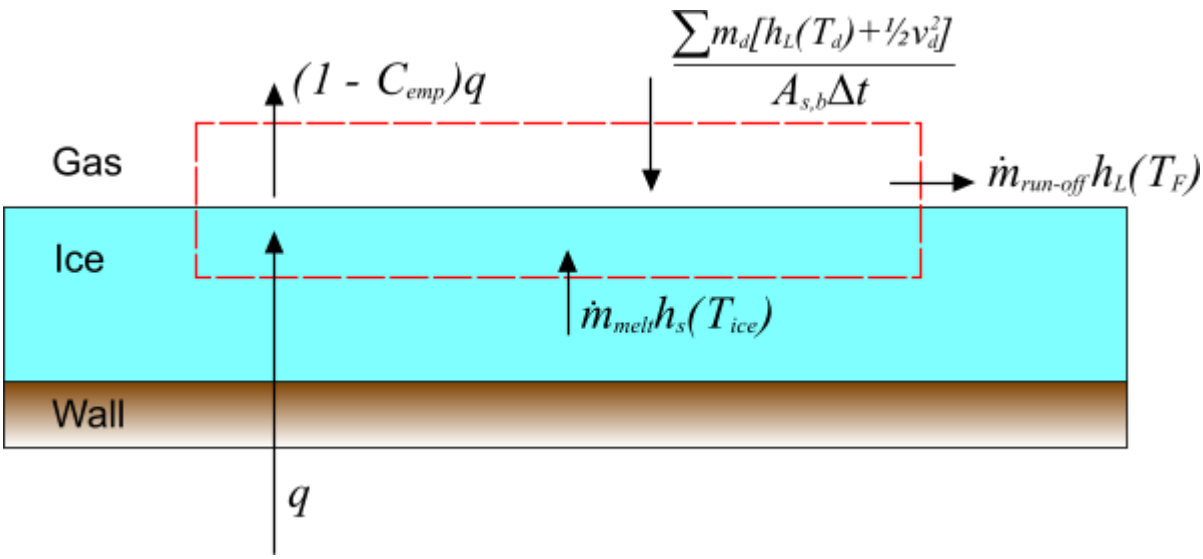
Given $\dot{m}_{melt}$, the ice thickness, s, can be updated using:

$$\frac{ds}{dt} = -\frac{\dot{m}_{melt}}{\rho_s} \tag{328}$$

where ρ_s is the density of the ice layer.

Accretion

The preceding analysis is extended to include an additional incident mass flux of liquid droplets, each with mass m_d, temperature T_d and velocity v_d. This expression is intended to represent liquid droplets impinging on the surface. The various contributions to the overall energy balance are depicted below:



In the above diagram, $A_{s,b}$ is the surface area of the boundary face onto which the droplets impinge and Δt is the time-step. The term:

$$\frac{\sum m_d\left[h_L\left(T_d\right)+\frac{1}{2}v_d^2\right]}{A_{s,b}\Delta t}$$

indicates the overall energy flux due to the impinging droplets.

The mass balance is therefore given as:

$$\dot{m}_{melt} + \dot{m}_d = \dot{m}_{run-off} \tag{329}$$

where:

$$\dot{m}_d = \frac{\sum m_d}{A_{s,b}\Delta t} \tag{330}$$

In addition to the constraint that $\dot{m}_{run-off} \geq 0$, there is also $\dot{m}_d \geq 0$. The constraint on $\dot{m}_{melt}$ is now $\dot{m}_{melt} \geq -\dot{m}_d$, that is, freezing is allowed but no faster than the incoming liquid rate.

The energy balance for this process is:

$$\dot{m}_{melt}h_s\left(T_{ice}\right) + \frac{\sum m_d\left[h_L\left(T_d\right)+\frac{1}{2}v_d^2\right]}{A_{s,b}\Delta t} + q = \left(1 - C_{emp}\right)q + \dot{m}_{run-off}\ h_L\left(T_F\right) \tag{331}$$

giving:

$$\dot{m}_{melt} = \frac{C_{emp}\ q + \frac{\sum m_d\left[C_{p,s}\left(T_d - T_F\right) + \frac{1}{2}v_d^2\right]}{A_{s,b}\Delta t}}{C_{p,s}\ \left(T_F - T_{ice}\right) + L} \tag{332}$$

A similar analysis can be made for the accretion of solid particles.

Accretion Ratio

Following Makkonen [226], it is useful to define the accretion ratio for non-zero $\dot{m}_d$ as:

$$\alpha = -\frac{\dot{m}_{melt}}{\dot{m}_d} \tag{333}$$

This quantity characterizes the nature of the accretion process. When $\alpha = 1$ there is no run-off and the accretion is “dry”, which tends to form rime ice. Otherwise there is some run-off, and the “wet” growth tends to form glaze ice. The accretion ratio is sometimes also known as the “freezing fraction”.

Single-Stream Heat Exchanger Formulation

STAR-CCM+ computes the local cell heat exchange, Q_c, from the specified target total heat exchange, Q_{total}.

Q_{total} is given as:

$$Q_c = Q_{total}\frac{V_c v_c\left(T_{ref} - T_c\right)}{\sum_c V_c v_c\left(T_{ref} - T_c\right)} \tag{334}$$

where:

- Q_{total} is the total heat source (positive Q_{total}) or sink (negative Q_{total}) transferred to/from the region, in Watts.
- T_{ref} is the reference temperature, which is the constant temperature of the fluid stream that is not modeled in the simulation. You specify this value in the simulation, in kelvins.

  You also specify $\Delta\ T_{min}$, which is the minimum temperature difference that must exist between the two streams.

  However, if the user-specified value is physically incorrect, the T_{ref} value is changed in the code as follows:

  - For a heater (that is, Q_{total} is positive), T_{ref} (which is the constant hot stream temperature in this case) is set to:

    $T_{ref} = T_{max} + \ \Delta\ T_{min}$ if $T_{ref} < T_{max}$

    In other words, if you specify a (hot) reference temperature that is less than the maximum cold stream temperature T_{max}, the reference temperature is reset. The cold stream is the one that is being modeled in the simulation. Therefore the reference temperature is set equal to the maximum

cold stream temperature plus the minimum temperature difference that must exist between the two streams.

- For a cooler (that is, Q_{total} is negative), T_{ref} (which is the constant cold stream temperature in this case) is set to:

  $T_{ref} = T_{min} - \Delta T_{min}$ if $T_{ref} > T_{min}$

  In other words, if you specify a (cold) reference temperature that is more than the minimum hot stream temperature T_{min}, the reference temperature is reset. The hot stream is the one that is being modeled in the simulation. Therefore the reference temperature is set equal to the minimum hot stream temperature minus the minimum temperature difference that must exist between the two streams.

- Q_c is the cell heat source/sink, in Watts.
- V_c is the cell volume, in m^3.
- v_c is the cell velocity, in m/s.
- T_c is the cell temperature, in kelvins.

Dual-Stream Heat Exchanger Formulation

This section provides formulation for the two methods of heat exchanger modeling:

- Basic Method
- Actual Method

Contents:

Basic Method
Actual Method

Basic Method

In this section, “local” refers to the value stored in each cell. The goal in the dual stream heat exchanger is to specify or compute a local/cell heat transfer coefficient (multiplied by the local heat exchanger area) UA_l as a function of the local velocity of the cold stream.

The local heat exchange between the hot and the cold stream, Q_{cell}, is given by:

$$Q_{cell} = \pm UA_l\left(T_{hot} - T_{cold}\right) \tag{335}$$

where the “+” is for the cold stream and “-” is for the hot stream (that is, hot stream loses heat to the cold stream). T_{hot} and T_{cold} refer to the local temperatures of the hot and cold streams respectively. This heat exchange is then added as a source term to the corresponding fluid energy equation after scaling by the cell volume, that is:

$S_{cell} = Q_{cell} V_{cell} (Nc) / V_{Hx}$

where S_{cell} is the cell source term, V_{cell} is the cell volume, Nc is the number of cells in the heat exchanger, and V_{Hx} is the volume of all the cells in the heat exchanger.

UAL Polynomial Option

In the UAL Polynomial option, the specified polynomial function is used to compute UA_l as a function of the local velocity of the cold stream. In the UAL Table option, UA_l is interpolated from the specified table of cold flow velocity vs. UA_l.

UAG Table Option

In the UAG Table option, UA_g refers to the overall heat transfer coefficient (multiplied by the heat exchanger area) across the entire heat exchanger, such that,

$$Q = UA_g\left(T_{\text{hot,in}} - T_{\text{cold,in}}\right) \tag{336}$$

where Q is the overall heat transfer across the entire heat exchanger, and $T_{\text{hot,in}}$ and $T_{\text{cold,in}}$ refers to the inlet temperatures of the hot and cold streams respectively. The UAG Table is specified as a table of cold flow mass flow rate vs. UA_g, which may be obtained from test/experimental data. You also specify other data pertaining to the test conditions including:

- Hot stream mass flow rate $\dot{m}'_h$.
- Inlet temperatures of the cold $T_{\text{cold,in}}$ and hot $T_{\text{hot,in}}$ streams.
- Specific heats of the cold Cp_c and hot Cp_h streams.
- Density of the cold fluid ρ_c.

Using these input data, the specified UAG table is internally converted into a UAL table (as in the UAL Table option). This conversion is performed by considering a heat balance between the cold and hot streams as detailed in the following algorithm.

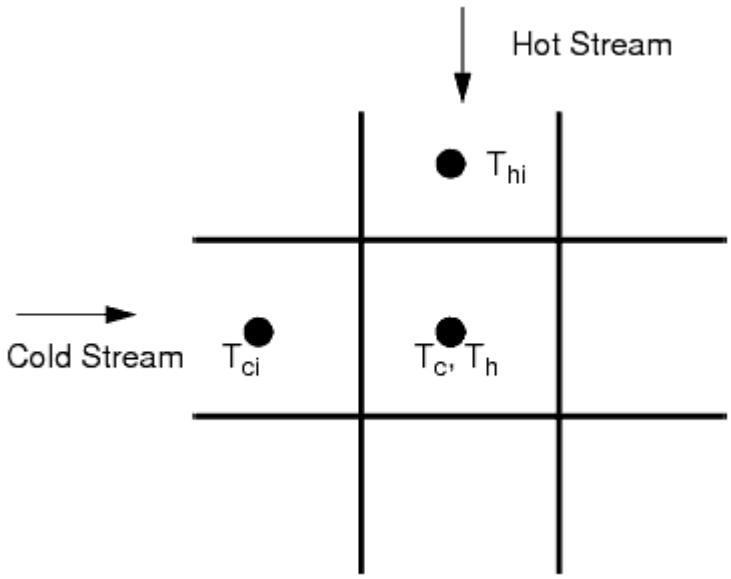
Repeat the following steps for each entry in the UAG Table:

- Retrieve cold stream mass flow rate $\dot{m}'_c$ and UA_g from the UAG Table.
- Compute the target overall heat transfer rate:

  $Q_{\text{target}} = UA_g\left(T_{\text{hot,in}} - T_{\text{cold,in}}\right)$
- Guess UA_l = UA_g / Nc, where Nc is the total number of cells in the heat exchanger.
- Build a linear system of equations, by constructing the following set of equations in each cell:

$$\begin{aligned} \dot{m}'_c Cp_c\left(T_c - T_{ci}\right) &= UA_l\left(T_h - T_c\right) \\ \dot{m}'_h Cp_h\left(T_{hi} - T_h\right) &= UA_l\left(T_h - T_c\right) \end{aligned} \tag{337}$$

  where T_c, T_h, T_{ci} and T_{hi} are local cell unknown temperatures as shown in the following figure.



Assemble the system for all the cells in the heat exchanger, and solve for the unknown cell temperatures T_c and T_h. Boundary conditions are required only at the inlet boundaries, at which the cell temperatures are set to the user-specified $T_{\text{cold,in}}$ and $T_{\text{hot,in}}$ for the cold and hot streams respectively.

- Compute the mean temperature difference between the cold and hot streams:

$$\overline{\Delta T} = \frac{\sum (T_h - T_c)}{(N_c)} \tag{338}$$

- Compute the estimated overall heat transfer $Q = UA_l \overline{\Delta T} Nc$.
- Update $UA_l = Q_{\text{target}} / (\overline{\Delta T} Nc)$.
- Repeat steps 4 to 7 until the user-specified Maximum Solver Iterations have reached or the convergence criteria $\left(Q - Q_{\text{target}}\right) / Q_{\text{target}} < \varepsilon$ is met, where ε is the user-specified convergence tolerance.
- Store the converged UA_l and cold flow velocity $V_c = \dot{m}_c / (\rho_c A_c)$ where A_c is the heat exchanger cold stream inlet boundary area into the internal UAL table.

The above algorithm is invoked once before the first energy solver iteration, and the internally stored UAL table is reused in the subsequent iterations for computing the energy source term as in Eqn. (335). Whenever any UAG Table input is changed, the internal UAL table is recomputed. UA_l in Eqn. (335) is then interpolated based on the local cold stream velocity from this internal UAL table.

The maximum allowable heat transfer rate in a heat exchanger occurs when the temperatures of each stream do not change at all from inlet to outlet (maximum temperature difference). Then the heat transfers to or from each stream are:

$$Q_{hot} = \dot{m}_h Cp_h \left(T_{\text{hot,in}} - T_{\text{cold,in}}\right) \tag{339}$$

and

$$Q_{cold} = \dot{m}_c Cp_c \left(T_{\text{hot,in}} - T_{\text{cold,in}}\right) \tag{340}$$

These heat transfers would have to be equal, and therefore the maximum heat transfer is limited to the minimum of these two values: the one for which $\dot{m}\,Cp$ is a minimum. Usually, this minimum value is for the hot fluid since coolants have high specific heats.

The target heat transfer rate Q_{target} must be chosen such that it is always smaller than the minimum of Eqn. (339) and Eqn. (340), that is:

$$Q_{transfer} < min\left(\dot{m}_h Cp_h, \dot{m}_c Cp_c\right)\left(T_{\text{hot,in}} - T_{\text{cold,in}}\right) \tag{341}$$

Q Table Option

The Q Table option is identical in functionality to the UAG Table option, except that you specify the overall heat transfer Q instead of UA_g in the input table. UA_g is then computed using Eqn. (336).

Q Map Option

To compute the local UAL, a linear system of algebraic equations is solved iteratively by considering the heat balance between the cold and the hot streams, preserving the target heat exchange rate, for each row of the Q Map (that is, for each value of the cold stream and hot stream mass flow rates). This UAL table is computed internally only once initially (or whenever any input for the Q Map is changed). The UAL table is then used to determine the coefficients in the following curve using least squares regression.

$$\left(\frac{1}{UAL}\right) = C_0 + \frac{C_1}{V_c^{\,ec}} + \frac{C_2}{V_h^{\,eh}} \tag{342}$$

In Eqn. (342), C_0, C_1 and C_2 are the coefficients that are determined by fitting the UAL table data to the above curve using least squares regression, V_c and V_h are the cold flow and hot flow local velocities, and ec and eh are the cold flow and hot flow exponents (which by default are taken to be 0.8). After the coefficients are determined, Eqn. (342) is used to determine the local UAL for a given cold flow and hot flow local cell velocities in the energy equation.

Actual Method

Application of local heat exchanger energy sources into the hot/cold energy equations

At every solver iteration, the following steps are carried out to include the actual flow dual stream heat exchanger sources into the energy equations of the hot and cold fluid streams:

1. Solver computes the UAL Table.
2. For every cell, the mass flux $\dot{m}_{ci}$ entering the cell is first computed as:

   $$\dot{m}_{ci} = \rho_{ci} v_{ci}$$

   where ρ_{ci} and v_{ci} are the density and magnitude of velocity of the cold stream entering the cell.
3. For every cell i of the heat exchanger region, the UAL Table is interpolated to compute the local heat transfer coefficient ual_i, given the cold and hot mass fluxes that are entering cell i
4. The local energy source is computed as:

$$src_i = \frac{ual_i(T_{hi} - T_{ci})V_i}{\frac{1}{NC}\sum_{j=1}^{NC} V_j}$$

where:

- T_{hi} and T_{ci} are the temperatures of the hot and cold streams in cell i
- V_i is the volume of cell i
- NC is the total number of cells in the heat exchanger region

5. Apply src_i as an energy source to the cold stream energy equation and as an energy sink to the hot stream energy equation.

Computation of the UAL Table

For each entry of the user input table (UAG Table, Q Table, or Q Map), a corresponding UAL value is computed, thus generating the UAL Table.

The UAL value is computed using the following expression:

$$UAL = \frac{\Gamma}{\Delta T_{net} \cdot NC}$$

where

When a UAG Table is supplied:

$$\Gamma = UAG\left(T_{hot}^{inlet} - T_{cold}^{inlet}\right)$$

where:

UAG is the user-table value, and T^{inlet}, hot or cold, is the inlet temperature for the hot or cold streams, either area averaged or mass-flow averaged over the inlet interface boundary. The summation is over all the cell faces.

- Area averaged:

$$T^{inlet} = \frac{\sum_f \left[T_f A_f\right]}{\sum_f A_f} \tag{343}$$

  T_f and A_f are the cell face values of temperature and surface area.

- Mass flow averaged:

$$T^{inlet} = \frac{\sum_f \left[T_f \rho_f \boldsymbol{V}_f \cdot \boldsymbol{A}_f\right]}{\sum_f \left[\rho_f \boldsymbol{V}_f \cdot \boldsymbol{A}_f\right]} \tag{344}$$

  ρ_f and $\boldsymbol{V}_f$ are the cell face values of density and velocity.

When a Q Table or a Q Map is supplied:

$$\Gamma = \frac{Q\left(T_{inlet}^{hot} - T_{inlet}^{cold}\right)}{T_{user}^{hot} - T_{user}^{cold}}$$

where:

Q is the user-table value,

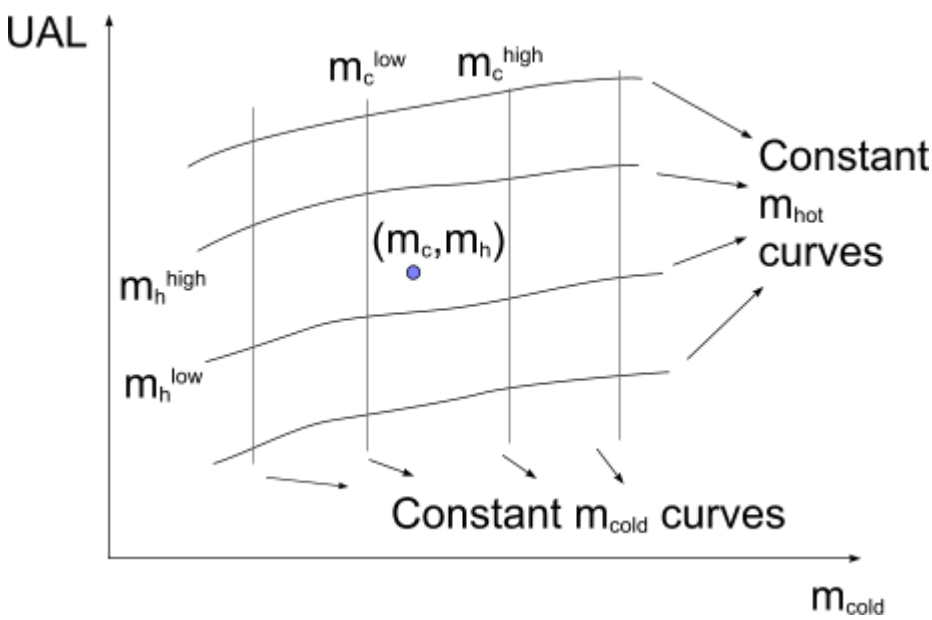
T_{user}^{hot} and T_{user}^{cold} are the user-supplied test rig inlet temperature values of the hot and cold streams respectively (these values are specified in STAR-CCM+ as properties of Q Table or Q Map).

ΔT_{net} is the volume averaged temperature difference between the hot and cold streams, over the exchanger region.

$$\Delta T_{net} = \frac{\sum_{i=1}^{NC} (T_{hi} - T_{ci}) V_i}{\sum_{i=1}^{NC} V_i}$$

Q Map Option

The **Q Map** option of the **Actual Flow Dual Stream** heat exchanger model uses bilinear interpolation to compute UAL values from an UAL map. Consider the graphical representation of a computed UAL map:



The points of intersection between the constant m_{cold} and m_{hot} represent data in the computed UAL map. The UAL at a general point (m_c, m_h) is given by:

$$UAL = u_{c1} + q_c(u_{c2} - u_{c1})$$

where

$$u_{c1} = u_1 + q_c(u_2 - u_1)$$
$$u_{c2} = u_3 + q_c(u_4 - u_3)$$

and

$$q_c = \frac{m_c - m_c^{low}}{m_c^{high} - m_c^{low}}$$

$$q_c = \frac{m_h - m_h^{low}}{m_h^{high} - m_h^{low}}$$

u_1, u_2, u_3 and u_4 are the UAL values at points $\left(m_c^{low}, m_h^{low}\right)$, $\left(m_c^{low}, m_h^{high}\right)$, $\left(m_c^{high}, m_h^{low}\right)$, and $\left(m_c^{high}, m_h^{high}\right)$, respectively.

For example, consider the following UAL table for a given **Q Map** table:

*m*cold	*m*hot	*UAL*
1.0	4.0	10.0
2.0	4.0	20.0
1.0	8.0	30.0
2.0	8.0	40.0

The UAL for a cold mass flow rate of 1.5 kg/s and a hot mass flow rate of 5.0 kg/s is computed as:

$$UAL = \frac{10 + 20}{2} + \left(\frac{5.0 - 4.0}{8.0 - 4.0}\right) \times \left[\frac{30.0 - 40.0}{2} - \frac{10.0 + 20.0}{2}\right] = 20$$

Circumferential Heat Flux Averaging Formulation

STAR-CCM+ bases the averaging of heat flux around the circumference of a spinning body on either of two cases:

- A body spinning in an interfacing medium.
- A body spinning in a meshless region, interacting across the body boundary.

Both cases use the following conditions:

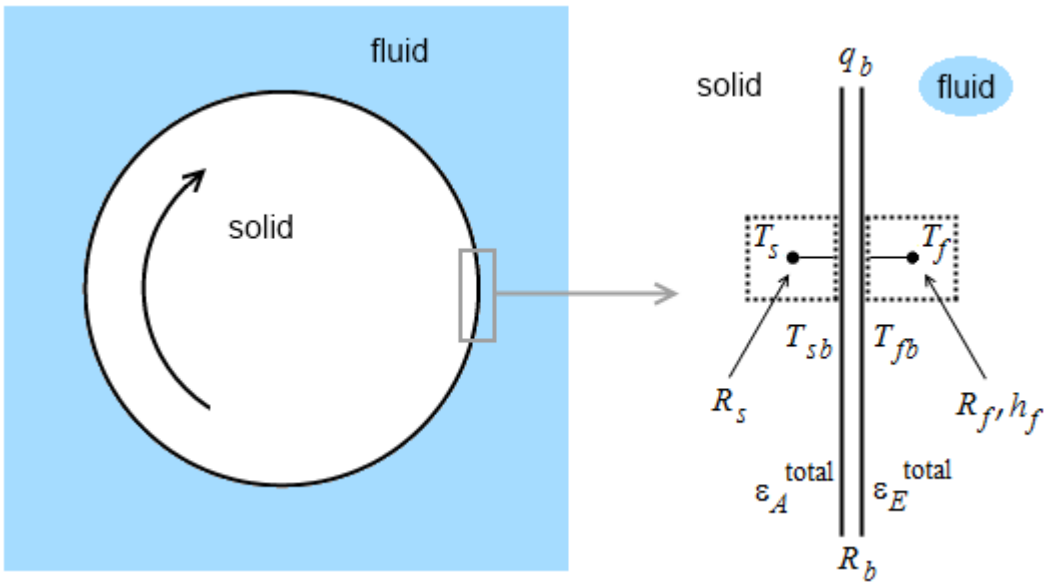
- The body in question is axisymmetric, with little or no circumferential variation in any properties, physical conditions, or geometry.
- Differences between the angular velocity of the axisymmetric body and any other bodies in the system are either very large or very small.
- Interface boundaries and wall boundaries between the rotating region and the surroundings are opaque. (The model does not allow radiative heat transfer between rotating and non-rotating regions.)

Contents:

A Body Spinning in an Interfacing Medium
A Body Spinning in a Meshless Region

A Body Spinning in an Interfacing Medium

In this case, a solid body spins rapidly in an interfacing fluid medium. Convection, conduction, and radiation operate within in the fluid. Conduction and radiation operate within the solid. The interface is opaque to radiation.



Using averaging, the boundary energy balance for the fluid-side interface is:

$$\left(\overline{h_f T_f} - \overline{h_f} T_{fb}\right) + \overline{\left(T_f / R_f\right)} - \overline{\left(1/R_f\right)} T_{fb} + \overline{q_{\mathrm{af}}} - \sigma \varepsilon_E^{\mathrm{total}} T_{fb}^{\ 4} = -\frac{q_b}{2} + \frac{\left(T_{fb} - T_{sb}\right)}{R_b} \tag{345}$$

where:

- The overline bar indicates averaging. The average value of a variable x is weighted by boundary face area A over a given axial strip n:

  $$\overline{x} = \frac{\sum_{\mathrm{strip}} x_n \times A_n}{\sum_{\mathrm{strip}} A_n}$$

- h_f is the convection coefficient of the fluid side.
- T_f is the cell-center temperature on the fluid side.
- T_{fb} is the fluid-boundary temperature.
- T_{sb} is the solid-boundary temperature.
- R_f is the thermal resistance from the fluid cell center to the boundary.
- q_{af} is the radiation heat flux in the fluid due to absorption.
- σ is the Stefan-Boltzmann constant.

- $\varepsilon_E{}^{\text{total}}$ is the total boundary emissivity.
- q_b is the heat flux in the interface.
- R_b is the thermal resistance of the boundary.

On the left-hand side of Eqn. (345):

- $\left(\overline{h_f T_f} - \overline{h_f} T_{fb}\right)$ is the convection term.
- $\overline{\left(T_f / R_f\right)} - \overline{\left(1/R_f\right)} T_{fb}$ is the conduction term.
- $\overline{q_{af}} - \sigma \varepsilon_E{}^{\text{total}} T_{fb}{}^4$ is the radiation term.

On the right-hand side of Eqn. (345), the term $\frac{\left(T_{fb} - T_{sb}\right)}{R_b}$ implies conduction through the boundary.

The boundary energy balance for the solid is:

$$\frac{1}{R_s}\left(T_s - T_{sb}\right) + q_{as} - \sigma \varepsilon_A{}^{\text{total}} T_{sb}{}^4 = -\frac{q_b}{2} + \frac{\left(T_{fb} - T_{sb}\right)}{R_b} \tag{346}$$

where:

- R_s is the thermal resistance from the solid cell center to the boundary.
- T_s is the cell-center temperature on the solid side.
- q_{as} is the radiation heat flux in the solid due to absorption.
- $\varepsilon_A{}^{\text{total}}$ is the total boundary absorbtivity.

On the left-hand side of Eqn. (346):

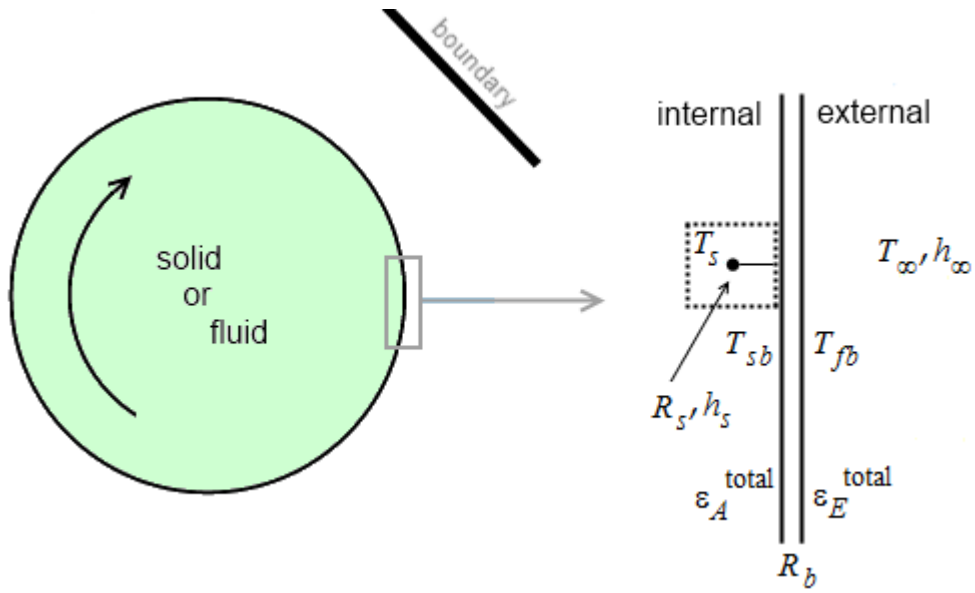
- $\frac{1}{R_s}\left(T_s - T_{sb}\right)$ is the conduction term.
- $q_{as} - \sigma \varepsilon_A{}^{\text{total}} T_{sb}{}^4$ is the radiation term.

For both equations, assume $T_f > T_{fb} > T_{sb} > T_s$.

A Body Spinning in a Meshless Region

In this case, a solid or fluid body spins rapidly in a region without mesh, with no medium between the body and its environment, represented in the following figure by an external boundary. The body and the external environment interact only by thermal radiation between their boundaries, across the empty meshless region.

Within the body, convection (if it is fluid), conduction, and radiation can all operate. Convection and radiation heat transfer operate between the boundaries of the body and the environment.



Using averaging, the boundary energy balance for the external side of the body is:

$$\left(\overline{h_\infty T_\infty} - \overline{h_\infty} T_{fb}\right) + \overline{q_{af}} - \sigma \varepsilon_E^{\text{total}} T_{fb}^{\ 4} = \frac{\left(T_{fb} - T_{sb}\right)}{R_b} \tag{347}$$

where:

- h_∞ is the convection coefficient of the external side.
- T_∞ is the temperature on the external side.

Other variables are defined as in the previous section.

On the left-hand side of Eqn. (347):

- $\left(\overline{h_\infty T_\infty} - \overline{h_\infty} T_{fb}\right)$ is the convection term.
- $\overline{q_{af}} - \sigma \varepsilon_E^{\text{total}} T_{fb}^{\ 4}$ is the radiation term.

The boundary energy balance for the internal side of the body is:

$$h_s\left(T_s - T_{sb}\right) + \frac{1}{R_s}\left(T_s - T_{sb}\right) + q_{as} - \sigma \varepsilon_A^{\text{total}} T_{sb}^{\ 4} = -\frac{\left(T_{fb} - T_{sb}\right)}{R_b} \tag{348}$$

Variables are defined as in the previous section.

On the left-hand side of Eqn. (348):

- $h_s\left(T_s - T_{sb}\right)$ is the convection term.
- $\frac{1}{R_s}\left(T_s - T_{sb}\right)$ is the conduction term.
- $q_{as} - \sigma \varepsilon_A^{\text{total}} T_{sb}^{\ 4}$ is the radiation term.

Solid Energy Formulation

The governing equation for energy transport in a solid is given as follows:

$$\frac{d}{dt}\int_V \rho C_p T dV + \oint_A \rho C_p T \mathbf{v}_s \cdot d\mathbf{a} = -\oint_A \mathbf{q}'' \cdot d\mathbf{a} + \int_V s dV \tag{349}$$

where:

- ρ is the solid density.
- C_p is the specific heat.
- T is the temperature.
- $\mathbf{q}''$ is the heat flux vector.
- $\mathbf{v}_s$ is the solid convective velocity (only applicable when the region is a pure body of rotation).

Contents:

Conducting Solid Shells Formulation

Conducting Solid Shells Formulation

The modeling of heat transfer in thin solid structures is based on the modeling assumption that the temperature variation normal to the shell surface is piecewise-linear. The consequence of this assumption is that the relevant heat fluxes can be estimated having only one cell across the shell thickness.

The energy transport in solid (in-plane conducting) shells is described with the same transport equation as the energy transport in solids.

Modeling Species

This chapter provides information about the chemical species models in STAR-CCM+. A species model is activated whenever a multi-component liquid or multi-component gas is chosen from the *Material* model section of the *Physics Model Selection* dialog.

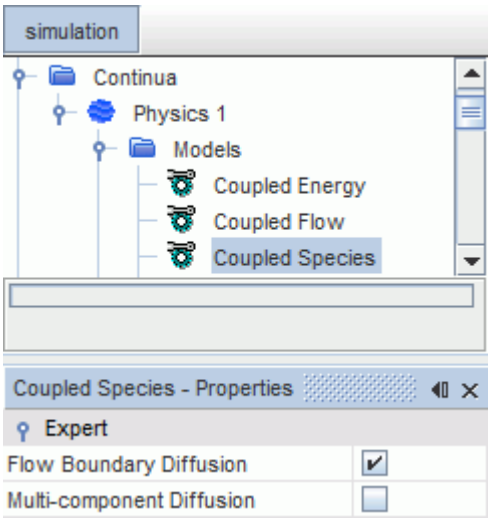
The specific models that are discussed are the Coupled Species model and the Segregated Species model. Some general advice on working with species is also given.

The Coupled Species Model

The Coupled Species model is an extension of the Coupled Flow model. Together with the Coupled Energy model, it solves the conservation equations for mass, momentum, and energy of a multi-component fluid mixture simultaneously using a time- (or pseudo-time-) marching approach.

For a mixture of N components, the Coupled Species model introduces N transport equations into the system. Together with global mass continuity, these equations provide a means for updating the field of N mass fractions defining the mixture composition. Inter-equation coupling is fully maintained between all equations in the system, including species transport with respect to momentum and energy, and vice versa.

For isothermal simulations of multi-component fluid mixtures, it is possible to omit the Coupled Energy model, in which case the Coupled Species model, together with the Coupled Flow model, solves the conservation equations for mass and momentum simultaneously.



The Segregated Species Model

The Segregated Species model solves the species continuity equations for a multi-component fluid mixture. For a mixture of N components, the Segregated Species model solves N transport equations sequentially. Together with global mass continuity, these equations provide a means for updating the field of N mass fractions defining the mixture composition.

Contents:

Working with Species
What Is the Segregated Species Solver?
Setting a Species Region Source
Species Transport Formulation
Species Transport Bibliography
Species Transport Field Functions Reference

Working with Species

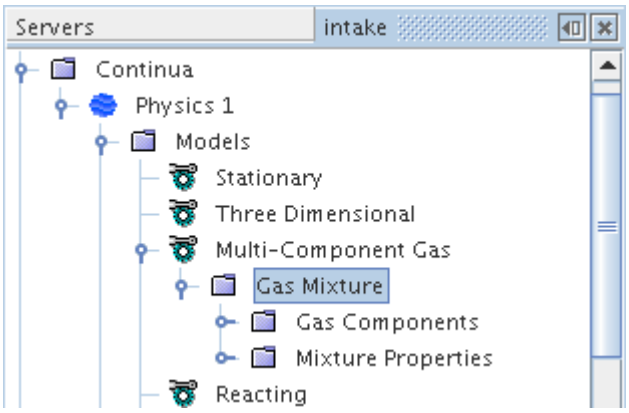
The Species models are often used with combustion models, and so part of the process for setting up a simulation that involves a Species model also involves selecting a combustion model.

In general, however, when a Species model is active it is necessary to:

- Specify mixture components (the different species) and associated properties
- Set species initial distribution
- Set species fractions at relevant boundaries

Specifying Mixture Components

A particular species is defined as one component in a multi-component gas mixture or liquid mixture. For a gas, the component collection is held within the Gas Mixture node of the Multi-Component Gas model node in the **Models** tree of the relevant physics continuum. These nodes are shown below. For liquids, this arrangement becomes the Liquid Mixture node of the Multi-Component Liquid model node.



The components of the gas or liquid mixture are defined by adding new components to the **Gas Components** or **Liquid Components** node. This can be done manually as described in the section, Adding Mixture Components, or automatically by importing Chemkin-formatted data.

Each new component's name and material properties are retrieved from the database with default property settings and values. You should review the default material property settings of the mixture as well as those of each new component. If necessary, change the settings of the mixture properties and then those of each new component accordingly (see Managing Multi-component Materials).

Note that:

- In order for a chemical species to be available for any given chemical reaction equation (see Defining Chemical Reactions) it must first be added to the **Gas Components** node.
- The order in which the mixture components are added is not important as all species are solved for.
- If the database does not include a component that is needed for your combusting mixture, make a local copy of the STAR-CCM+ material database (file `props.mdb`) and add the required component and its physical properties to it.

- For an EBU or CFM combustion simulation, you most likely want to select **Thermodynamic Polynomial Data** (if it is available) as the method for the *Specific Heat* property of each mixture component. If you choose an alternative method for *Specific Heat* (for example, **Constant** or **Polynomial in T**), then you are required to provide Heat of Formation and Standard State Temperature values for that component as well.

Segregated Species and Coupled Species Model Properties

The Segregated Species and Coupled Species models share certain properties.

Properties

Convection (Segregated Species only)	Selects the convection scheme that is used.	
	1st-order	Selects the first-order upwind convection scheme.
	2nd-order	Selects the second-order upwind convection scheme.
	MUSCL 3rd-order/CD	Selects the hybrid MUSCL 3rd-order/CD convection scheme. Adds a **Bounded Differencing** sub-node to the **Segregated Species** node.

Expert Properties

Flow Boundary Diffusion	Allows you to activate or deactivate diffusion at flow boundaries. Activated by default.
Multi-component Diffusion	When **On**, diffusive flux for a given component becomes a function of the mass fraction gradient of all other components. You can only calculate Molecular Diffusivity using the Binary Diffusion Coefficients method or Kinetic Theory Diffusivity method. This option is **Off** by default, in which case Fick's Law is used to calculate the diffusive flux in the species transport equations instead of multi-component diffusion..
Soret Effect	The Soret effect accounts for mass diffusion due to the presence of a temperature gradient. For most problems, this effect is negligible; however, it can play a significant role in problems with large temperature gradients and problems with species of widely varying molecular weights. In practice, the Soret effect can be important for chemical vapor deposition, isotope separation and combustion of light components at low temperatures [31]. When activated, it is represented in the diffusive flux term of the species transport equations (see Eqns. Eqn. (356) and Eqn. (357)). This option is deactivated by default.

	Note: When you activate this option, you must specify the thermal diffusion coefficient for each component. See Using the Thermal Diffusion Coefficient.	
Secondary Gradients (Segregated Species only)	Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces.	
	On	Include both secondary gradients.
	Off	Exclude both secondary gradients.
	Interior Only	Include the interior secondary gradients only.
	Boundaries Only	Include the boundary secondary gradients only.

Species Boundary and Initial Conditions

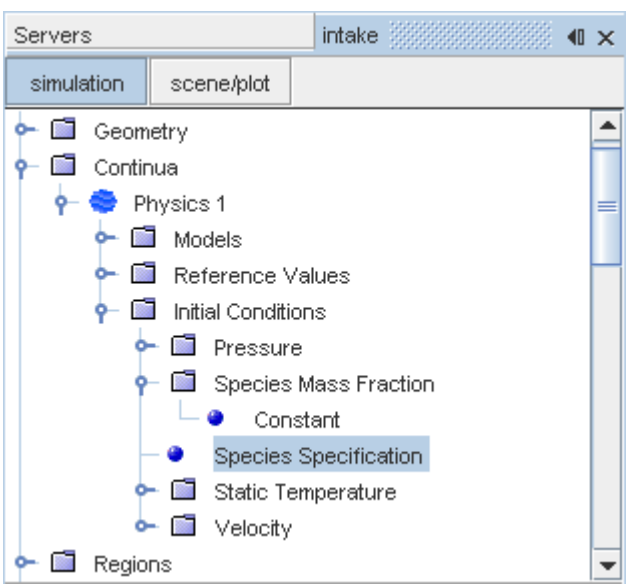
The distribution of species within a continuum or across flow boundaries can be defined using either mass fraction or mole fraction. In STAR-CCM+, the same set of nodes are provided for flow boundaries and initial conditions.

Use the Species Specification node within the **Physics Conditions** folder of a boundary or initial condition node to choose between mass fraction or mole fraction as the method for specifying species distribution. The corresponding value node, Species Mass Fraction or Species Mole Fraction, is added to the relevant **Physics Values** folder.

On wall boundaries, the Wall Species Option allows you to define the wall as impermeable to species transfer, or one having a specified mass fraction or specified flux.

Species Specification Node

This node is added to the **Initial Conditions** node of physics continua, and to the **Physics Conditions** node of flow boundaries, after you have selected components for the mixture.



The single *Method* property of this node is used to switch between specifying the species distribution in terms of mass fraction or mole fraction.

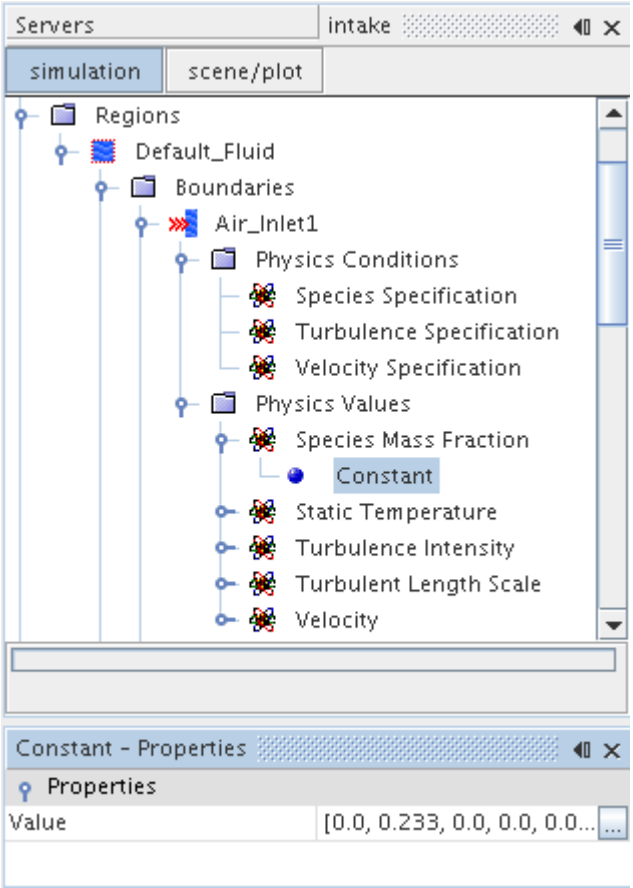
Species Specification Properties

Method	Defines the basis on which initial species distribution is defined.	
	Mass fraction	Mixture components are defined in terms of mass fraction. When this option is selected, a **Species Mass Fraction** sub-node is added under the **Initial Conditions** node (or the **Physics Values** node for a flow boundary).
	Mole fraction	Mixture components are defined in terms of mole fraction. When this option is selected, a **Species Mole Fraction** sub-node is added under the **Initial Conditions** node (or the **Physics Values** node for a flow boundary).

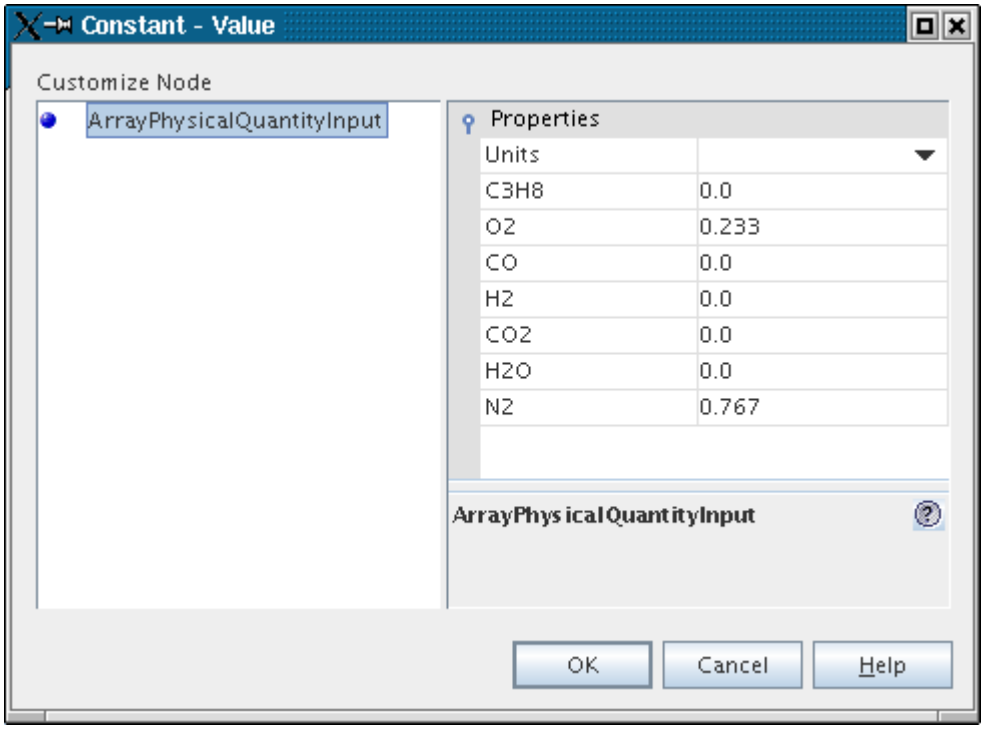
Using Species Mass Fraction

It is a scalar quantity that is used for both initial and boundary conditions, and is defined by one of the following options:

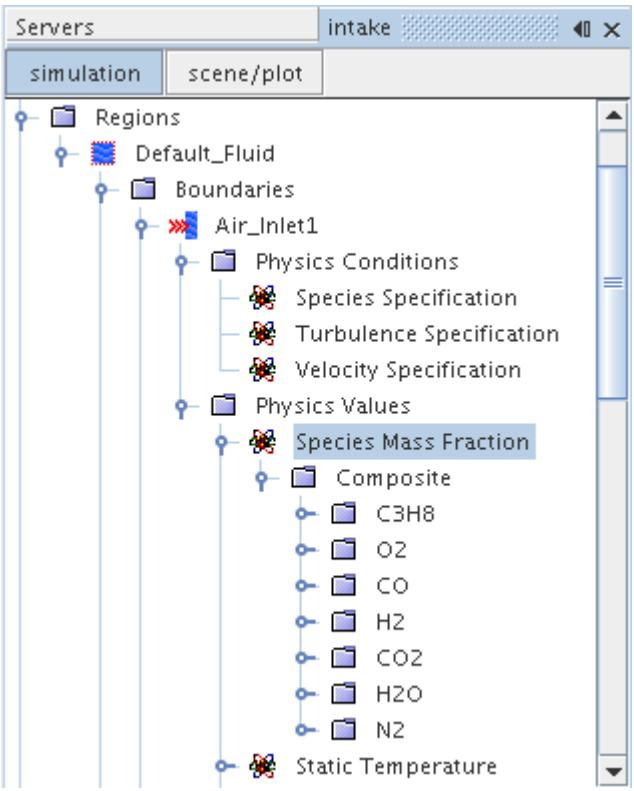
- **Constant** specifies a fixed value for all mixture components. The required values are entered in the *Properties* window as a scalar array profile in the order in which the components were defined.
-



Alternatively, you may click the property customizer box, marked with an ellipsis and located to the right of the *Value* property array. This opens an input editor, shown below, which allows a more convenient way of entering these values as it lists component names side-by-side with the specified values.



- **Composite** specifies the distribution of each component separately; distributions for each component are entered as scalar profiles.
-

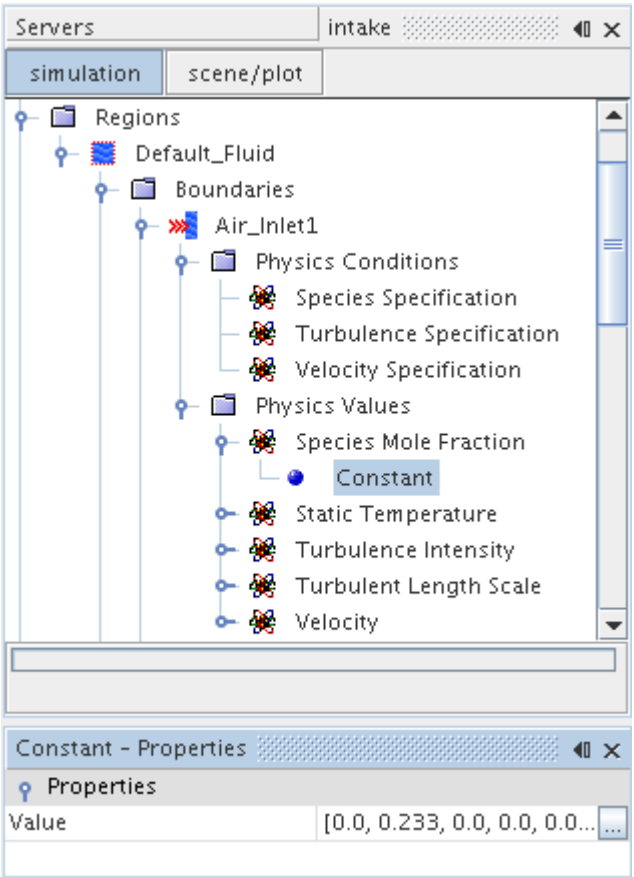


Both initial and boundary values that you specify are normalized prior to being used so that they sum to 1.0. Before normalizing, any value entered larger than 1.0 is set to 1.0. After this, normalization will be applied across all fractions. For example, [1.0, 2.0, 3.0] becomes [1.0, 1.0, 1.0], which normalizes to [0.333, 0.333, 0.333].

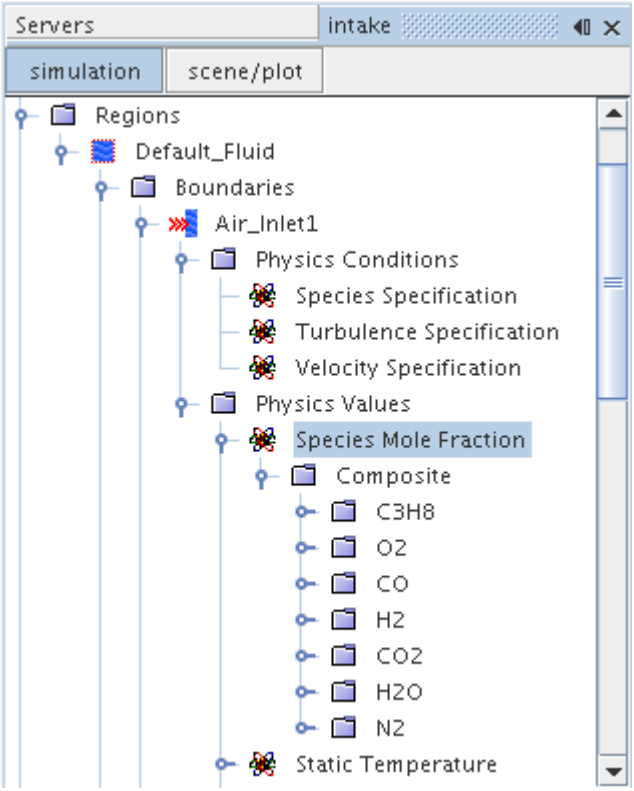
Using Species Mole Fraction

The mole fraction of a species is the ratio of its molar amount over the mixture molar amount, and is represented by the **Species Mole Fraction** node.

It is a scalar quantity that is used for both initial and boundary conditions, and is defined by one of the following options:

- **Constant** specifies a fixed value for all mixture components. The required values are entered in the *Properties* window as a scalar array profile in the order in which the components were defined.
- 

  As for the Species Mass Fraction, the property customizer can be accessed by clicking the ellipsis to the right of the Value property array. This opens an input editor that identifies the different components in the array.
- **Composite** specifies the distribution of each component separately; distributions for each component are entered as scalar profiles.
-



The Wall Species Option

For species transfer, this option defines a wall boundary as impermeable or as having a specified value or flux. The Coupled Species model or the Segregated Species model activates it.

Limitations:

- The bulk velocity of the wall species influx or outflux is negligible.
- The gain or loss of momentum due to wall species influx or outflux is negligible.

Properties of the Wall Species Option

Method	Defines the permeability of the boundary.	
	Impermeable	Specifies the boundary as impermeable.
	Specified Flux	Specifies a species flux at the boundary. The energy that is associated with the wall species influx or outflux is appropriately added. Adds a Species Flux node to the **Physics Values** node.
	Specified Value	Specifies a species mass fraction at the boundary. Adds a Species

		Mass Fraction node to the **Physics Values** node.

Post-Processing

You can monitor the species flux at the wall with the **Boundary Species Flux** field function. To activate this field function, enable the *Temporary Storage Retained* expert property of the Segregated Species solver or the Coupled Implicit solver. This feature lets you verify your specification of the boundary conditions.

Using the Species Flux

The species flux is the mass flow rate of species added or removed per unit area. The **Species Flux** node is added when the *Method* property of the Wall Species Option boundary condition is set to Specified Flux. It is entered as a scalar array profile.

What Is the Segregated Species Solver?

This object controls the solution update for the Segregated Species model.

Properties

Under-Relaxation Factor	At each iteration, this property governs the extent to which the old solution is supplanted by the newly computed solution. The default value is 0.9.

Expert Properties

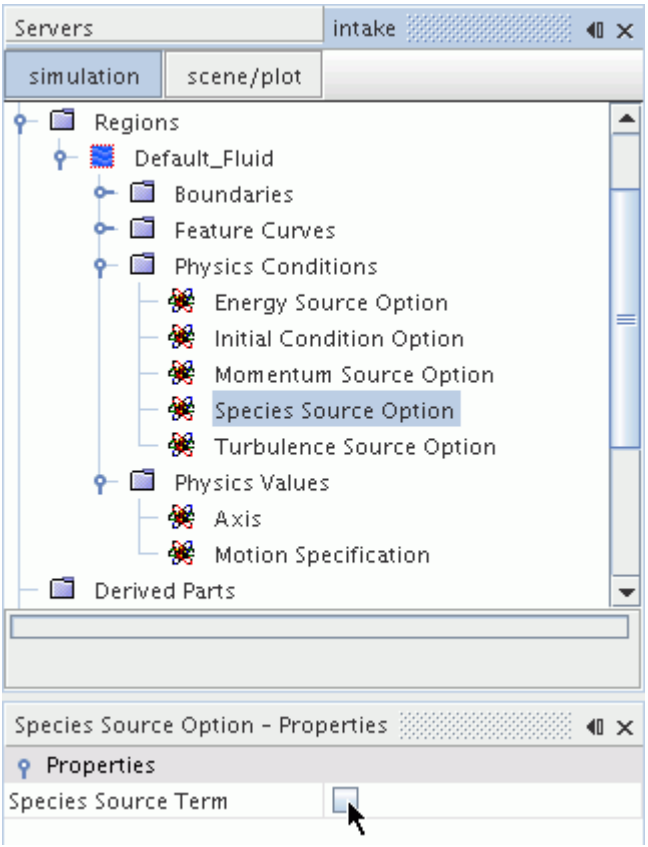
For information about properties in the *Expert* category, see Expert Properties.

Setting a Species Region Source

The Species Source is a region source that is activated by the Coupled Species or the Segregated Species model. It specifies the rate at which a unit mass of the species is added or removed per unit volume of the region (Mass/Time/Volume). In the Physics Conditions of a region, you can enable user-specified species sources by ticking the checkbox of the *Species Source Term* property.

To define a species source within a region:

- Select the **Species Source Option** node within the **Physics Conditions** folder for the region.
- Tick the *Species Source Term* property in the *Properties* window.

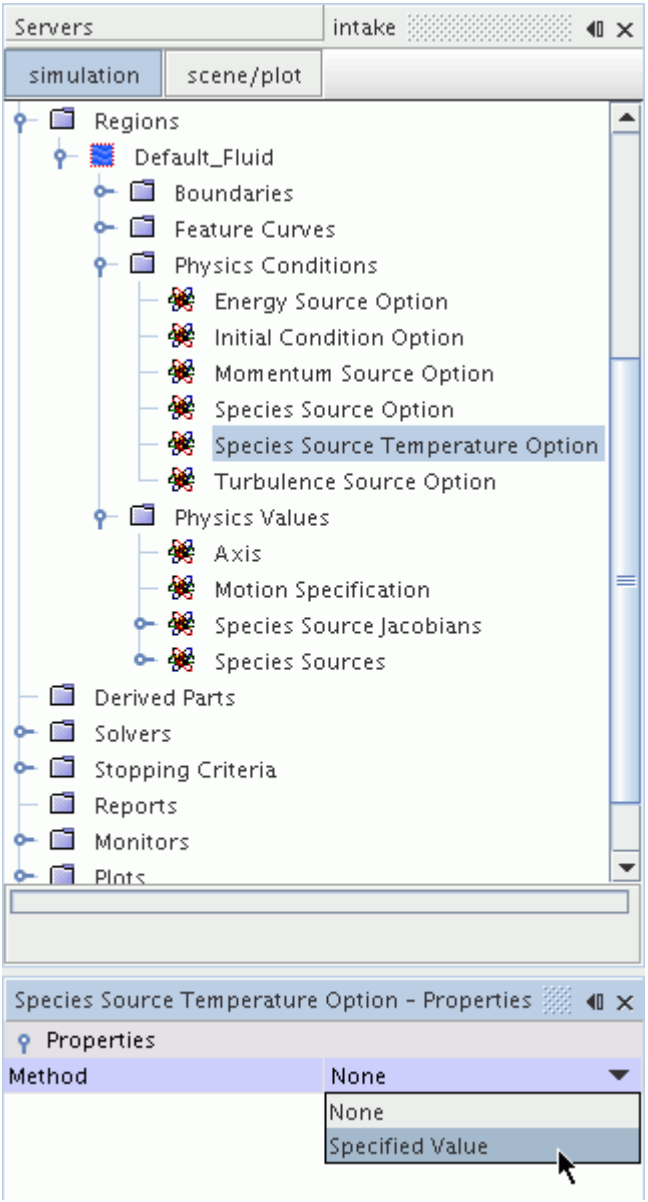


One additional node appears in the **Physics Conditions** folder for the region: the Species Source Temperature Option, and two nodes are added to the **Physics Values** folder for the region: the **Species Sources** and **Species Source Jacobians** nodes. Use these nodes to set the value and derivative of the source you wish to set.

Species Source Temperature Option

When you add a species source to a simulation, ensure that an equivalent source of energy is added or removed with the species mass. STAR-CCM+ can provide this energy source automatically by calculating the energy that is associated with the species source at a particular temperature. To have STAR-CCM+ add the equivalent energy automatically:

- Select the **Species Source Temperature Option** node. In the *Properties* window, set the *Method* property to **Specified Value**.



An additional node, the **Species Source Temperature**, is added to the **Physics Values** folder for the region, where the temperature of the user species source can be specified.

If you leave the *Method* property as **None**, you must supply your own energy source using the Energy Source Option on the region.

Species Sources

This node can be used to specify the species source, , of each of the defined species. They can be specified as constant or composite, since this value is entered as a scalar array profile.

Species Source Jacobians

This node can be used to specify the Jacobian, $d\left(S_{Y_i}\right)/d\left(Y_i\right)$, for each species. This value is entered as a scalar array profile.

Species Source Temperature

This node enables you to specify the temperature of the user species sources. STAR-CCM+ adds an amount of energy corresponding to the mass of the user species sources and this temperature. The default option uses a constant temperature value set in Kelvin. The alternative options are described in the scalar profile properties section.

Species Transport Formulation

The transport equation for the mass fraction Y_i of a species is given as:

$$\frac{\partial}{\partial t}\left(\int_{\widetilde{V}} \rho Y_i\right) d\widetilde{V} + \oint_A \rho Y_i \left(\mathbf{v} - \mathbf{v}_g\right) \cdot d\widetilde{\mathbf{a}} = \oint_A \left[J_i + \frac{\mu_t}{\sigma_t} \nabla Y_i\right] \cdot d\widetilde{\mathbf{a}} + \int_{\widetilde{V}} S_{Y_i} \, \mathrm{d}\, \widetilde{V} \tag{350}$$

In which the effect of turbulent diffusion is accounted for by the term $\frac{\mu_t}{\sigma_t}$ and the laminar diffusion is defined by J_i.

where:

- i is the component index.
- ρ is the overall density.
- $d\widetilde{V} = \alpha_i \chi dV$, where α_i is the volume fraction of phase *i* (set to 1 for single phase flows) and χ is the void fraction.
- $d\widetilde{\mathbf{a}} = \alpha_i \chi d\mathbf{a}$
- $\mathbf{v}$ is the velocity.
- $\mathbf{v}_g$ is the grid velocity.
- μ_t is the turbulent dynamic viscosity.
- σ_t is the turbulent Schmidt number – which is assumed to have a constant default value of 0.9. This value can be modified if necessary.
- S_{Y_i} is a user specified region source term for species i.
- J_i is the diffusive flux.

When multi-component diffusion is deactivated, Fick's Law is used to calculate the diffusive flux J_i, given by:

$$J_i = \rho D_{m,i} \nabla Y_i \tag{351}$$

where $D_{m,i}$ is the Molecular Diffusivity coefficient for component i.

When multi-component diffusion is activated, the scalar diffusivity of Fick's Law is replaced by a matrix, and the diffusive flux for component i becomes a function of the mass fractions for all components:

$$J_i = \rho \sum_{j=1}^{N} D_{i,j} \nabla Y_j \tag{352}$$

$D_{i,j}$ represents the multi-component diffusion coefficients which are calculated using the Maxwell-Stefan equations. These equations define the diffusive flux implicitly as a function of mole fraction gradients, which are given as:

$$\nabla x_i = \frac{M_w}{\rho} \sum_{j=1, j \neq i}^{N} \frac{x_i J_j}{M_j \mathfrak{D}_{i,j}} - \frac{x_j J_i}{M_i \mathfrak{D}_{i,j}} \tag{353}$$

$\mathfrak{D}_{i,j}$ represents the binary diffusion coefficient between component i and j. Writing these equations in matrix form gives:

$$[B]\nabla Y = [A]J \tag{354}$$

where $[B]$ represents the mapping from mass fraction gradients to mole fraction gradients and $[A]$ represents the Maxwell-Stefan equations. By inverting $[A]$ and multiplying by the matrix $[B]$, the multi-component diffusion coefficients are calculated.

$$D_{i,j} = \rho \sum_{k=1}^{N} A_{i,k}{}^{-1} B_{k,j} \tag{355}$$

For N components, STAR-CCM+ solves transport equations for all species and ensures that all mass fractions sum to 1.

When the Soret Effect option is activated, an additional term is added to the diffusive flux, J_i, for each component. In the case that multi-component diffusion is deactivated, the expression for the diffusive flux becomes [32]:

$$J_i = \rho D_{i,m} \nabla Y_i + \rho \frac{D_{i,t}}{T} \nabla T \tag{356}$$

and when multi-component diffusion is activated, the expression becomes [31]:

$$J_i = \rho \left(\sum_{j=1}^{N} D_{i,j} \nabla Y_j \right) + \rho \frac{D_{i,t}}{T} \nabla T \tag{357}$$

where $D_{i,t}$ is the Thermal Diffusion coefficient for component i.

For the Warnatz Model, the thermal diffusion coefficient is computed based on the thermal diffusion ratio. The thermal diffusion ratio between component i and component j is given as [33]:

$$K_{T,ij} = \frac{15}{2} \frac{\left(2A^*_{i,j} + 5\right)\left(6C^*_{i,j} - 5\right)}{A^*_{i,j}\left(16A^*_{i,j} - 12B^*_{i,j} + 55\right)} \frac{M_i - M_j}{M_i + M_j} X_i X_j \tag{358}$$

where:

$$\begin{aligned} A^*_{i,j} &= \frac{1}{2} \frac{\Omega^{(2,2)}_{i,j}}{\Omega^{(1,1)}_{i,j}} \\ B^*_{i,j} &= \frac{1}{3} \frac{5\Omega^{(1,2)}_{i,j} - \Omega^{(1,3)}_{i,j}}{\Omega^{(1,1)}_{i,j}} \\ C^*_{i,j} &= \frac{1}{3} \frac{\Omega^{(1,2)}_{i,j}}{\Omega^{(1,1)}_{i,j}} \end{aligned} \tag{359}$$

and $\Omega_{i,j}$ represent collision integrals between component i and component j. Using the thermal diffusion ratio, the thermal diffusion coefficient for component i is given as:

$$D_{i,T} = D_{i,m} \frac{M_i}{M_w} \sum_{j=1}^{N} K_{T,ij} \tag{360}$$

where $D_{i,m}$ is the effective molecular diffusivity of component i into the mixture, M_i is the molecular weight of component i, and M_w is the mean molecular weight of the mixture.

Species Transport in Porous Media

In porous media, the diffusive flux (resulting from species concentration or thermal gradients) is modified by the porosity χ and tortuosity τ of the media.

For Fick's Law:

$$J_i = \frac{\chi}{\tau}\rho D_{i,m}\nabla Y_i \tag{361}$$

For multi-component diffusion:

$$J_i = \rho\frac{\chi}{\tau}\sum_{j=1}^{N} D_{i,j}\nabla Y_j \tag{362}$$

For the Soret effect:

$$J_i = \rho\frac{\chi}{\tau}D_{i,m}\nabla Y_i + \rho\frac{\chi}{\tau}\frac{D_{i,t}}{T}\nabla T \tag{363}$$

These formulations apply equally to the porous region and porous media approaches.

Species Transport Bibliography

[31] *Warnatz, J., Maas, U., and Dibble, R.W. 2006. Combustion: Physical and Chemical Fundamentals, Modeling and Simulation, Experiments, Pollutant Formation, Springer, Berlin, Germany.*

[32] *Zel'dovich, Y.B., Raizer, Y.P. 2002. Physics of Shock Waves And High-Temperature Hydrodynamic Phenomena, Dover Publications, Mineola, NY.*

[33] *Kee, R. J., Dixon-Lewis, G., Warnatz, J., Coltrin, M.E., and Miller, J.A. 1986. "A Fortran Computer Code Package For the Evaluation of Gas-Phase, Multicomponent Transport Properties'", Sandia National Labs.*

Species Transport Field Functions Reference

The following field functions are made available when a Multi-Component Material has been specified.

Effective Mass Diffusivity of Species	The effective mass diffusivity, $D_{\text{eff}} = D_m + \mu_t/(\rho\sigma_t)$, where D_m is the molecular diffusivity, μ_t is the turbulent viscosity, and σ_t is the turbulent Schmidt number. For laminar flows, $\mu_t = 0$, and $D_{\text{eff}} = D_m$.
Mass Fraction of Species	The mass fraction, $Y_i = m_i/m_m$, of species i in mixture m.
Molar Concentration of Species	The molar concentration, $C_i = \rho_m Y_i/W_i$, of species i in a mixture m, where ρ_m is the mixture density,Y_i is the species mass fraction, and W_i is the species molecular weight.
Mole Fraction of Species	The mole fraction, $X_i = Y_i W_m/W_i$, of species i in a mixture m, where Y_i is the species mass fraction,W_m is the molecular weight of the mixture, and W_i is the molecular weight of the species.

Modeling Porous Media

STAR-CCM+ allows you to simulate the transport of a fluid or energy (e.g. heat, electrical charge) through porous materials using the concept that the action of the porous media can be represented using appropriate loss (or 'diffusion') coefficients.

Overview

What Is a Porous Medium?

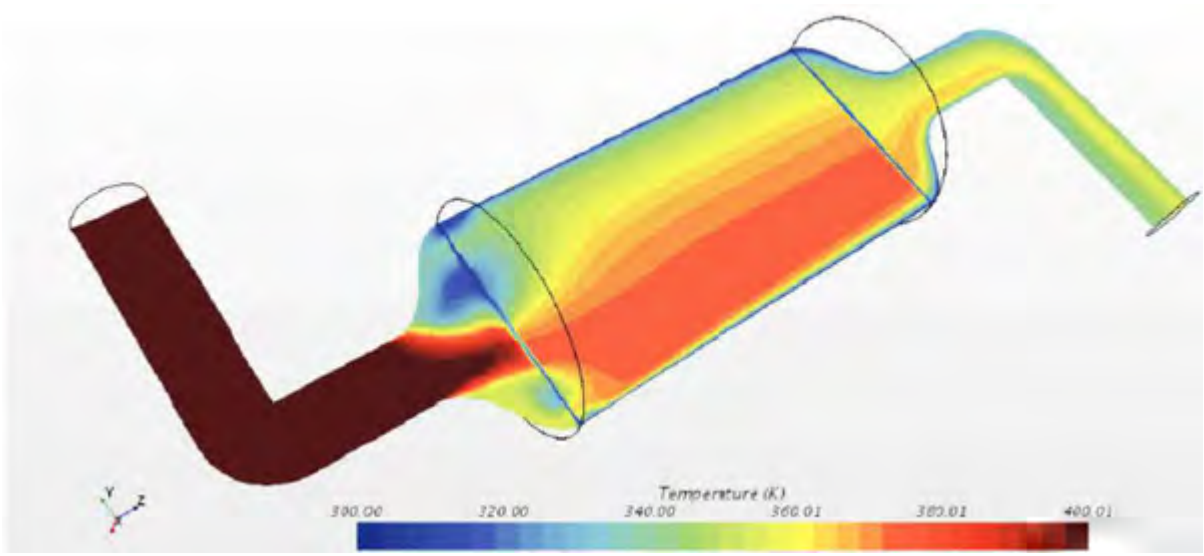
A porous medium is a solid material permeated by a number of pores filled by gas, liquid or a multiphase mixture. The pores may be relatively fine (as in porous rocks), or coarse in nature (honeycomb structures and wire meshes). The conducting phase can be the fluid, the solid material or a combination of both. The pores may permit fluid flow and diffusion of species; equally, heat, current, magnetic fields and acoustical energy (to name a few examples) may be transported through the fluid and/or the solid material.

In many applications, the porous structure does not have to be replicated in full three-dimensional realism, but can instead be approximated using loss coefficients—as detailed in the following sections and in the Porous Media Theory Guide for fluid flow and heat transfer.

Porous media can be incorporated within other physics models inSTAR-CCM+, such as electrochemical reactions or species diffusion, using modified diffusion coefficients within the relevant equations.

Applications

- Filtration and flow conditioning: honeycomb structures, meshes, sponges, and fibrous materials
- Exhaust components: mufflers, catalytic converter assemblies and baffles
- Fuel-cell components
- Lithium-ion battery design and optimization
- Packed-bed chemical reactors
- Representation of heat exchanger elements
- Representation of small components within much larger structures (e.g. flow around large structures—such as buildings and oil and gas facilities)



Definitions

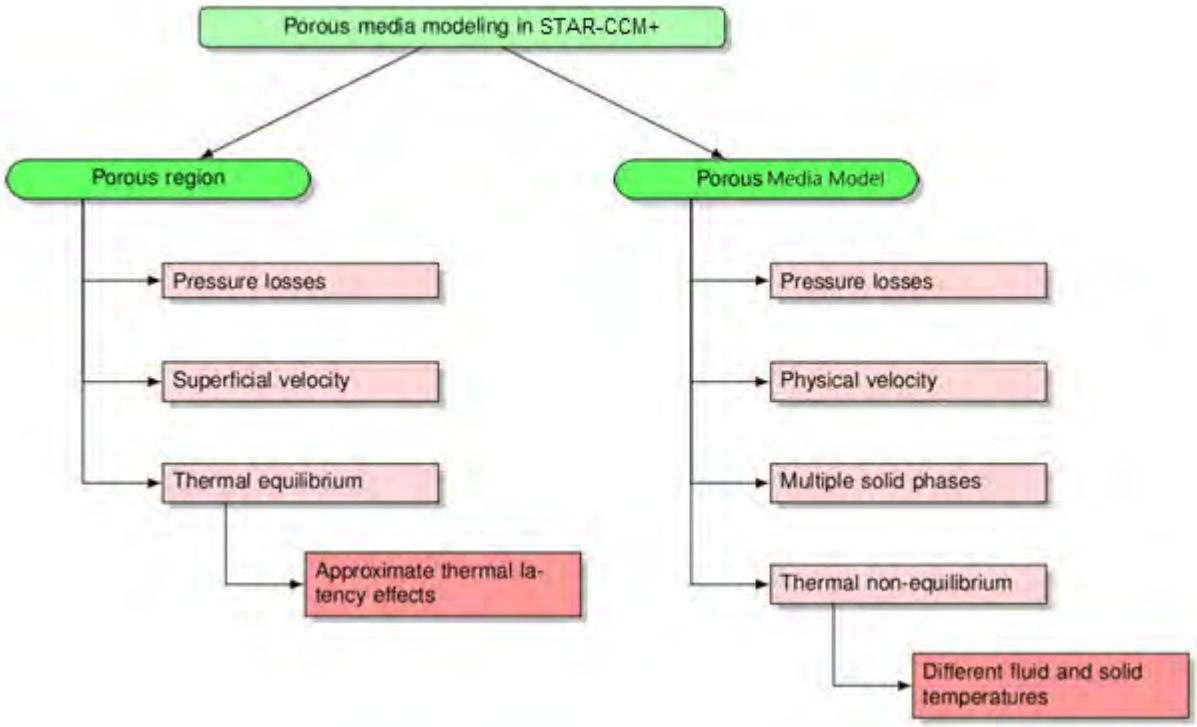
Porosity is defined as the ratio of open volume to total volume within the porous medium where fluid is transported.

Tortuosity is defined as the ratio between the actual path length through the porous medium (from point-to-point) and the straight-line distance between the same points.

Porous regions and porous phases (fluid flow application)

A porous medium can be represented in STAR-CCM+ in one of two ways:

1. Porous Region
2. Porous Media Model



The implementation of these two approaches is detailed in the Porous Media Theory Guide.

Porous Region

Represent porous media in your simulation using porous regions when:

- Working with the physical velocity of the fluid is not a requirement
- The solid and fluid phases of the porous media are in thermal equilibrium

Porous Media Model

In this case, the porous media are set up as 'frozen' Eulerian phases—within which the porosity of the material, resistance coefficients, temperature and other properties are set. Use the Porous Media model to simulate:

- Multiple solid phases
- Physical velocity inside a porous medium (required for transient heat transfer analyses)
- Heat transfer between the solid and fluid phases of the porous media
- Where other physics may be required to be modeled—such as chemical reactions
- Phase-specific electrodynamic potential

Regardless of the method used, you specify the porosity, resistance coefficients and tortuosity values within the Region.

Porous Baffles

In cases where the thickness of the porous medium is small compared with the other dimensions—for example: baffles, membranes, meshes and filters—"infinitely thin" porous baffles may be used, as described in Modeling Porous Baffles.

Note: Porous baffles can only be used with porous regions.

Mesh Requirements

Ideally, the mesh should be aligned with the principal flow direction. One appropriate method is the Directed Mesher; however, depending on the shape of the porous medium, the Generalized Cylinder, Thin Mesher, or Extruder can also be used.

Solution Method

Porous media can be solved using either the Segregated or Coupled solvers, depending on the application. The section Flow and Energy Workflow provides suggestions that can help with convergence using these solvers.

Contents:

Activating Porous Regions
Porous Media Models
Porous Media Model Workflow
Examples of Setting Porous Resistance Properties
Influence of Porous Resistance on Temperature for Incompressible Flow
Using Porous Media to Model Flow in Tubes

Porous Regions Workflow

To simulate a porous material, set the region type to **Porous Region**, then set the porosity and the inertial and viscous coefficients. These are required for the porous source term in the momentum equation. If a turbulence model is active, specify turbulence quantities downstream of the porous region. An option is also available to control the use of porous media flux.

1. Select the node **Regions** > **[porous region]**.
2. Set *Type* to **Porous Region**.
3. Under the selected region, select **Physics Values** > **Porosity** and set the value of the *Method* property. Do likewise for **Physics Values** > **Tortuosity**.

   Porosity χ and tortuosity τ are used to scale the diffusion coefficients for species and passive scalar transport. Porosity is used to calculate the effective thermal conductivity in the porous region.

   Porosity is defined as the ratio of open volume to total volume of the porous region, and is entered as a scalar profile. For unsteady flow calculations, the porosity is used to modify the fluid density ρ in the time derivative terms of the continuity, energy, and species equations; ρ is replaced by the product $\rho\chi$. This substitution reflects the fact that the fluid occupies only part of the volume. For effective thermal conductivity, the porosity is used to blend the thermal conductivities of the fluid and solid materials that are represented within the porous region. See Superficial Velocity Formulation.

   Tortuosity is defined as the ratio between actual (convoluted) path length between two points in the porous material and the length of the straight path connecting the same points. The higher the tortuosity of a porous region, the lower the rate of diffusion, since the convoluted paths slow down the spread of the diffusing material.
4. Specify porous resistance. The porous inertial resistance $\mathbf{P}_i$ and the porous viscous resistance $\mathbf{P}_v$ are defined as tensor profiles. See Eqn. (2199). You specify these properties for the region.
   - Select **[porous region]** > **Physics Values** > **Porous Inertial Resistance** and select the value of the *Method* property. Repeat for **Porous Viscous Resistance**.
   - For multi-phase fluids, for each phase, select **[porous region]** > **Phase Conditions** > **[phase]** > **Physics Values** > **Porous Inertial Resistance** and select the value of the *Method* property. Repeat for **Porous Viscous Resistance**.
5. If you are using the Segregated Flow model, decide whether to solve for momentum fluxes. By default, the momentum equation is solved with the addition of source terms for convection and viscous fluxes. See Eqn. (372). To omit these terms, edit the **Regions** > **[porous region]** > **Physics Conditions** > **Porous Media Flux Option** node, then activate the *Discount Momentum Fluxes* property.

   Omitting these terms reduces computational expense, but the flow solver becomes less robust. Therefore, it is best to use the default setting.
6. If the flow through the porous region is turbulent, select the **Turbulence Specification** node in the **Physics Conditions** manager. This node contains a *Method* property that provides several options for how turbulence scales are set within the porous region. Specify the appropriate turbulent quantities under the **Physics Values** > **Porous Viscous Resistance** node.

The turbulence transport equations are not solved for in the porous regions. Turbulence quantities in the fluid leaving the porous region are constructed from the user-defined values; they are not transported from the upstream side of the porous region.

7. If the simulation involves energy transport across porous regions, specify the thermal properties of the porous solid material and of the fluid passing through it. When one of the fluid energy models is active, the following values are provided in the **Physics Values** node of a porous region:
   - Solid Density
   - Solid Specific Heat
   - Solid Thermal Conductivity

   The fluid energy models are Coupled Energy, and the Segregated Fluid Enthalpy, Isothermal, and Temperature models.

Porous Media Models

The Porous Media model represents a porous material as a solid phase. Use it to model the physical velocity inside a porous medium instead of the superficial velocity, or when solid and fluid inside porous media are not in thermal equilibrium.

After selecting the Porous Media model, select either the Porous Media Thermal Equilibrium or the Porous Media Non-Equilibrium model. To model pressure losses, select the Porous Media Drag model.

Porous Media Model

Theory	See Porous Media in the Theory Guide.
Provided By	**[physics continuum]** > **Models** > *Optional Models*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Porous Media**
Requires	*Material*: **Gas** or **Liquid** *Flow*: **Coupled Flow** or **Segregated Flow**
Specific Right-Click Actions	Right-click **Porous Media** > **Porous Phases** and select **New** to create a sub-node under **Porous Phases** named **Phase *n***. See Porous Phases.

Limitations

The Porous Media model only supports internal interfaces between two fluid regions. If any interface type other than internal is necessary, the Porous Media Model cannot be used.

The Porous Media model is incompatible with the following models:

- Adjoint
- Dispersed Multiphase
- Eulerian Multiphase
- Harmonic Balance
- Lagrangian Multiphase
- Viscous Flow

Porous Phases

You can create multiple phases under the Porous Phases manager node, to define individual porous phases. Use the *Regions* property to select the regions to which the porous phase applies, specified through the custom editor ... . See Region Settings. Each phase has a **Models** subfolder. Right-click to select models for the phase.

Note: Models become available only when Porous Media Thermal Equilibrium or Porous Media Thermal Non-Equilibrium has been selected for the continuum.

After one of the above models has been selected, the following models become available for the individual phase:

- Solid
- Constant Density
- Polynomial Density
- User Defined EOS
- Segregated Solid Energy (available only with the Porous Energy Thermal Non-Equilibrium model)
- Coupled Solid Energy (available only with the Porous Energy Thermal Non-Equilibrium model)
- Electromagnetism (Electrodynamic Potential)

Porous Media Model Region Settings

Applies to fluid regions.

Porosity

χ in Eqn. (2179), the fraction of the porous medium occupied by fluid, specified by scalar profile methods. Physically meaningful values are in the range $0 < \chi \leq 1$.

Tortuosity

τ in Species Transport in Porous Media, the ratio between actual (convoluted) path length between two points in the porous medium and the length of the straight path connecting the same points. Physically meaningful values of τ are ≥ 1.

Porous Media Drag Model

The Porous Media Drag model calculates pressure losses inside the porous phase.

Theory	See Eqn. (2193).	
Provided By	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Porous Media Drag**	
Requires	*Optional Models*: **Porous Media**	
Activates	Physics Models	**Multiphase Interaction** > **Phase Interactions** > **[Phase Interaction n]** > *Optional Models*
	Region Inputs	**Porous Inertial Resistance** and **Porous Viscous Resistance**, available after regions are selected for Porous Phases. See Porous Media Drag Region Settings.

Porous Media Drag Model Region Settings

Applies to fluid regions.

Porous Inertial Resistance

The value of the coefficient α in Eqn. (81), non-negative. The default value is 0. This value is input as a scalar profile. See also Eqn. (2193).

Porous Viscous Resistance

The value of the coefficient β in Eqn. (81), non-negative. The default value is 0. This value is input as a scalar profile. See also Eqn. (2193).

Porous Media Thermal Equilibrium and Non-Equilibrium Models

The Porous Media Thermal Equilibrium model represents the case of thermal equilibrium between the fluid and solid phases in the porous medium and solves a single energy transport equation.

The Porous Media Thermal Non-Equilibrium model represents the case of thermal non-equilibrium between the fluid and solid phases. It solves one energy transport equation for fluid temperature in the porous region and another one for solid temperature.

The Porous Media Thermal Non-Equilibrium model is often more computationally expensive to run than Porous Regions or the Porous Media Thermal Equilibrium model because the combination solves extra energy transport equations.

Model Names	**Porous Media Thermal Equilibrium**
	Porous Media Thermal Non-Equilibrium
Theory	See Eqn. (2193).
Provided By	**[physics continuum]** > **Models** > *Porous Media Energy*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Porous Media Thermal Equilibrium**
Requires	*Optional Models*: **Porous Media** *Optional Models*: **Coupled Energy**, **Segregated Fluid Enthalpy**, **Segregated Fluid Isothermal**, or **Segregated Fluid Temperature**

Porous Phase-Physics Continuum Interaction Model

Use the Porous Phase-Physics Continuum Interaction model to calculate energy transfer between the solid and fluid phases of the porous region. Without this model, the energy is not exchanged between the solid and fluid phases. This model requires the use of the Porous Phase-Physics Continuum Energy Transfer model.

Provided By	**[physics continuum]** > **Models** > **Multiphase Interaction** > **Phase Interactions** > **[Phase Interaction n]** > **Models** > *Optional Models*	
Example Node Path	**Multiphase Interaction** > **Phase Interactions** > **[Phase Interaction n]** > **Models** > **Porous Phase-Physics Continuum Interaction**	
Requires	*Optional Models*: **Multiphase Interaction** *Porous Media Energy*: **Porous Media Thermal Non-Equilibrium**	
Properties	Key properties are: *Porous Phase* and *Fluid Phase*. See Porous Phase-Physics Continuum Interaction Properties.	
Activates	Physics Models	**Multiphase Interaction** > **Phase Interactions** > **[Phase Interaction n]** > *Optional Models*
	Region Inputs	See Region Settings.

Porous Phase-Physics Continuum Interaction Properties

Porous Phase

Select the solid porous phase from the dropdown menu.

Fluid Phase

The phase of the fluid region specified for Porous Media model, read only.

Porous Phase-Physics Continuum Energy Transfer Model

Select this model to enable calculation of energy transfer between the phases selected with the Porous Phase-Physics Continuum Interaction model.

Provided By	**[physics continuum]** > **Models** > **Multiphase Interaction** > **Phase Interactions** > **[Phase Interaction n]** > **Models** > *Optional Models*
Example Node Path	**Multiphase Interaction** > **Phase Interactions** > **[Phase Interaction n]** > **Models** > **Porous Phase-Physics Continuum Energy Transfer**
Requires	*Optional Models*: **Porous Phase-Physics Continuum Interaction**

Porous Media Model Workflow

Select the Porous Media model and associate it with the porous region. Next, select either the Porous Media Thermal Equilibrium or the Porous Media Thermal Non-Equilibrium model. Optionaly, select the Porous Media Drag model to represent pressure losses.

1. Select the Porous Media model from *Optional Models*.
2. Create a new porous phase (**Phase 1**) under Porous Media model.
3. Set the regions to be modeled as porous in the *Regions* property of **Phase 1**. Note that the region type is still **Fluid Region**.

   The Porosity and Tortuosity profile nodes appear under **Physics Values** node of each region.
4. To model pressure losses, select the Porous Media Drag model. Set the values of the porous resistance coefficients, which now appear under each specific **Physics Values** node of the regions.
5. If the solid and fluid inside porous region is not in thermal equilibrium, and this fact is relevant to the case, use the Porous Media Thermal Non-Equilibrium model:
   a. Select a fluid energy model from *Optional Models*:
      - Coupled Energy
      - Segregated Fluid Enthalpy
      - Segregated Fluid Isothermal
      - Segregated Fluid Temperature
   b. Select the Porous Media Thermal Non-Equilibrium model from *Porous Media Energy*.

      This model does **not** assume that fluid in the region is in thermal equilibrium with the solid, so it solves one energy transport equation for fluid temperature and another one for solid temperature in the region.
   c. Select the Multiphase Interactions model from *Optional Models*.
   d. Create a new phase interaction and select the appropriate Porous Phase-Physics Continuum models for it.
   e. Select the phase from the drop-down menu of the Porous Phase-Physics Continuum Interaction model.

      After this heat transfer coefficient and interaction area density profiles appear under Physics Values node of the phase interaction node in the region.
6. If the solid and fluid inside porous region **is** in thermal equilibrium and this fact is relevant to the case, use the Porous Media Thermal Equilibrium model:
   a. Select a fluid energy model from *Optional Models*:
      - Coupled Energy
      - Segregated Fluid Enthalpy
      - Segregated Fluid Isothermal
      - Segregated Fluid Temperature
   b. Select the Porous Media Thermal Equilibrium model from *Porous Media Energy*. This model assumes that the fluid in the region is in thermal equilibrium with the solid, so it solves a single energy transport equation.

c. Select appropriate models for the phase.

   If the solid phase is electrically conducting, select the **Electromagnetism** > **Electrodynamic Potential** model.

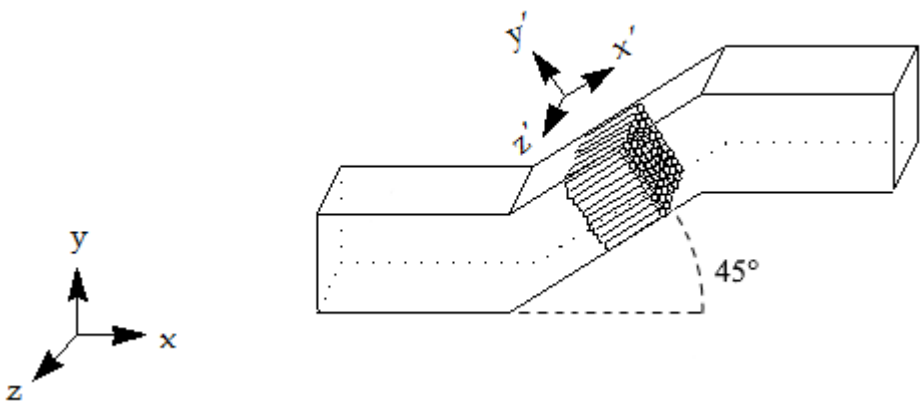
Examples of Setting Porous Resistance Properties

Porous resistance properties can be specified in several different ways depending on the characteristics of the porous media being modeled and the expected flow conditions through it.

By making it possible to specify porous resistances using field functions and expressions (as well as constant values), STAR-CCM+ provides a framework whereby you can define your own resistance models. This section provides some examples of how to do this.

Orthotropic Viscous Resistance

Consider a tube bundle of the type often used to condition flows in pipes and ducts. This porous medium is situated in a section of the duct that is oriented at 45 degrees to the x-axis as shown below.



Measurements undertaken in a wind tunnel (air is the working fluid) have shown that the tube bundle device imparts a pressure drop of 70 Pa per meter when the velocity is 1 m/s. In addition, for the range of velocities of interest, the pressure drop varies linearly with velocity.

Since the Reynolds number is low, only the viscous resistance tensor $\mathbf{P}_v$ is specified, so the porous resistance is a linear function of flow velocity. In the principal coordinates, the primary coefficient is 70. To restrict the flow in the directions perpendicular to the primary direction, choose a resistance coefficient that is 2-3 orders of magnitude larger than primary coefficient, for example 10000. Choosing a larger resistance does not affect the flow but can adversely affect the convergence characteristics. Therefore the viscous resistance tensor in principal coordinates is chosen to be:

$$\begin{bmatrix} 70 & 0 & 0 \\ 0 & 10000 & 0 \\ 0 & 0 & 10000 \end{bmatrix}$$

The ways that this tensor can be specified in STAR-CCM+ include:

- Using the principal tensor profile method:

The XX, YY and ZZ components are 70, 10000 and 10000 respectively. The **XX Axis** is the vector (1, 1, 0) relative to the Laboratory *Coordinate System*, representing a vector that is tilted at 45 degrees to the Laboratory x-axis. The YY Axis is then (-1, 1, 0), also relative to the Laboratory coordinate system.

- Using the composite symmetric tensor method:

  This method is simplified by creating a local Cartesian coordinate system with (1, 1, 0) as the *X Axis Direction* corresponding to x' and (-1, 1, 0) as the *Y Axis Direction* corresponding to y'. Using this coordinate system, the tensor components are all zero except for the XX, YY, and ZZ components, which are 70, 10000, and 10000 respectively.

Isotropic Resistance Using the Ergun Equation (Forchheimer for Packed Beds)

One of the classic equations describing flow through a porous medium is Darcy's law, which relates the flow velocity to the pressure gradient based on a measure of permeability. This law, which applies to creeping flow ($Re \ll 1$), is expressed using [397] as:

$$-\nabla p = \frac{\mu}{k_p}\mathbf{v}_s \tag{364}$$

where μ is the fluid molecular viscosity, k_p is the permeability (considered to be an intrinsic property of the porous medium) and v_s is the superficial velocity through the medium.

As the flow velocity increases, the relationship between velocity and pressure gradient becomes nonlinear. Dupuit and Forchheimer (as reported by [397]) proposed the addition of a quadratic term as follows:

$$-\nabla p = \frac{\mu}{k_p}\mathbf{v}_s + b\rho\left|\mathbf{v}_s\right|\mathbf{v}_s \tag{365}$$

Eqn. (365) is commonly known as the Forchheimer equation [397]. The factor b must be deduced depending on the particular flow and medium of interest and is commonly determined through experiment.

One example of the Forchheimer equation for a particular class of flow is the Ergun equation (as discussed in [196]). This equation is an empirical model for the pressure drop dp over a length L of fluid flowing through a packed bed:

$$-\frac{dp}{L} = P_v + P_i = \frac{150\mu\ (1-\chi)^2 v_s}{\chi^3 {D_p}^2} + \frac{1.75\rho\ (1-\chi)v_s^2}{\chi^3 D_p} \tag{366}$$

where:

- P_v and P_i are the viscous and internal terms.
- ρ is the fluid density.
- D_p is the mean diameter of particles in the porous medium.
- χ is the volume porosity.

Comparing Eqn. (366) with Eqn. (365) shows that the permeability and b factor are:

$$\frac{1}{k_p} = \frac{150\,(1-\chi)^2}{\chi^3 {D_p}^2} \tag{367}$$

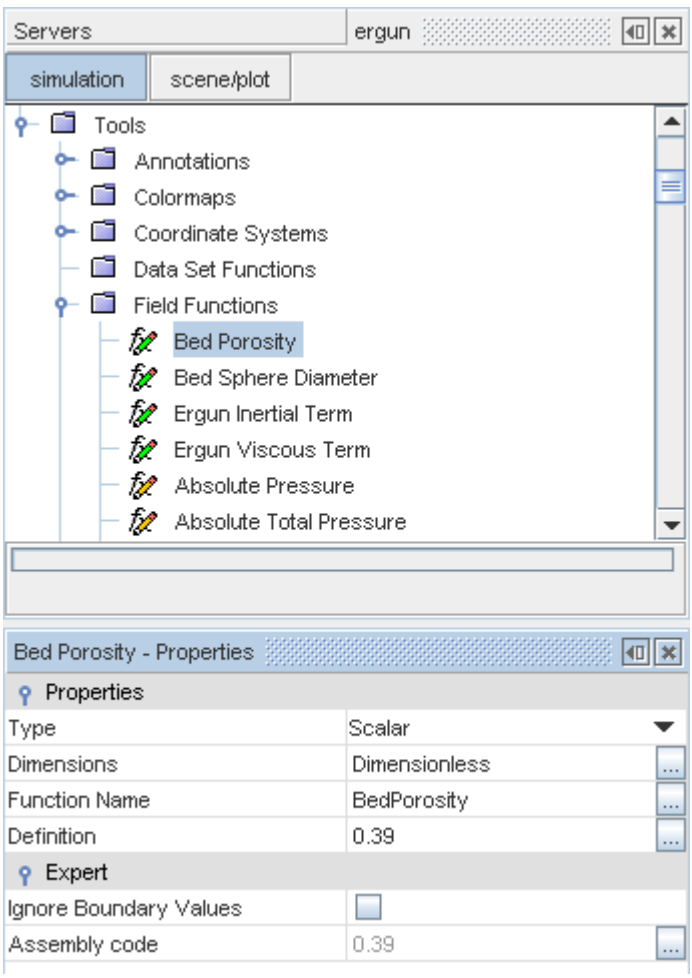
and

$$b = \frac{1.75\,(1-\chi)}{\chi^3 D_p} \tag{368}$$

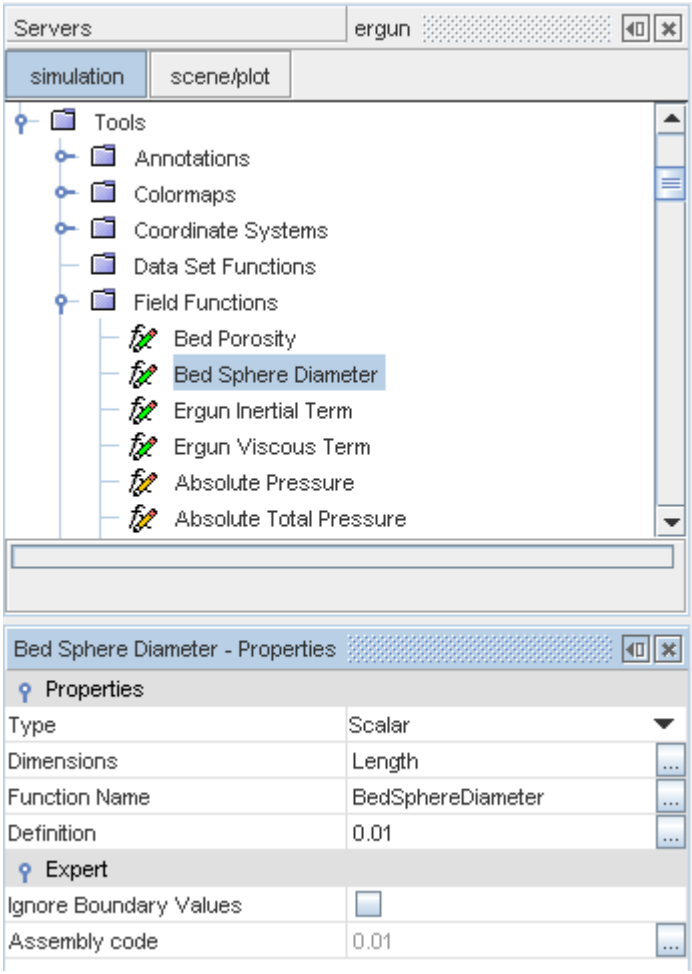
To implement this model in STAR-CCM+, it is necessary to create user field functions for each of these terms. Field functions are first defined for the volume porosity and mean particle diameter (which are set to constant values in this example). These values are then used in the field functions for the two terms.

Parameter Field Functions

The volume porosity is defined as a scalar field function with no dimensions, and whose *Definition* is set to the required value (0.39 in this example).



The particle diameter (0.01 m in this example) is defined in a similar way, except that the *Dimensions* property is set to `Length`.

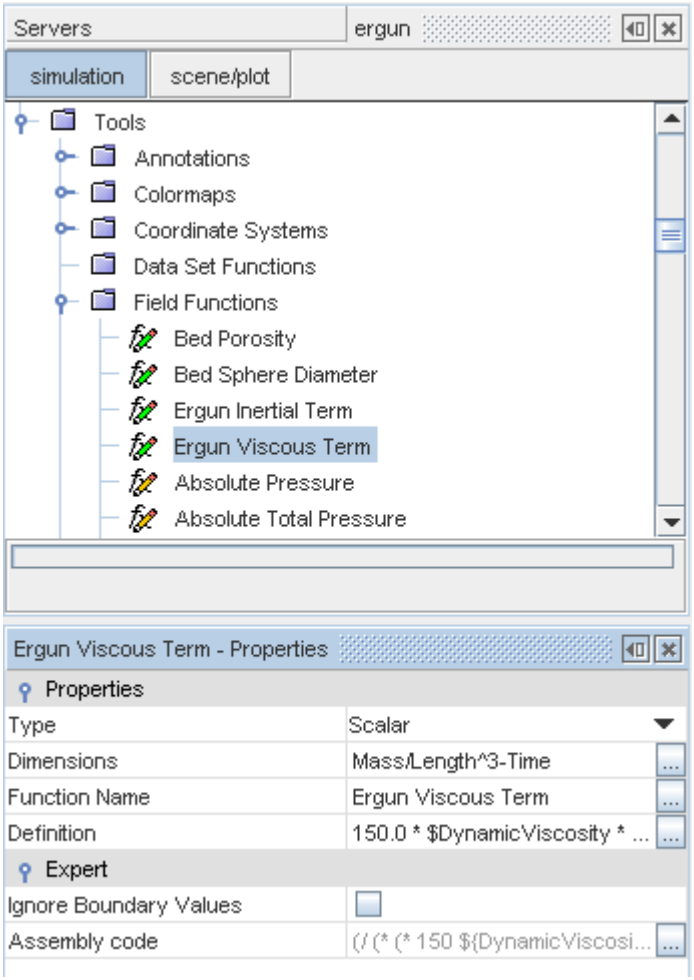


Viscous Term Field Function

The scalar field function for the viscous term is dimensioned as $Mass \cdot Length^{-3} \cdot Time^{-1}$, and the full *Definition* is:

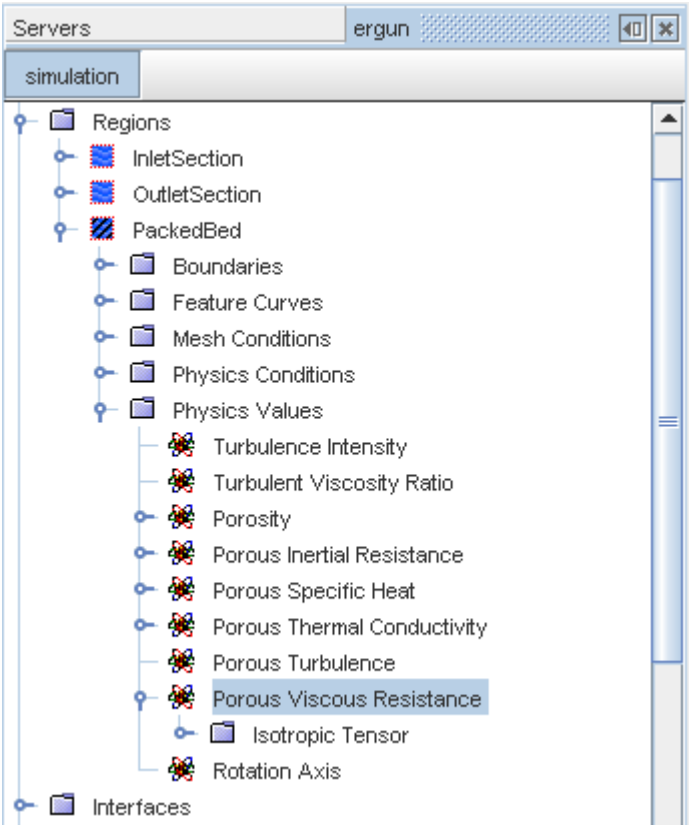
```
150.0*$DynamicViscosity*pow((1-$BedPorosity),2)/(pow($BedPorosity,
3)*pow($BedSphereDiameter,2))
```

The function uses the **Bed Porosity** and **Bed Sphere Diameter** functions that were defined previously.

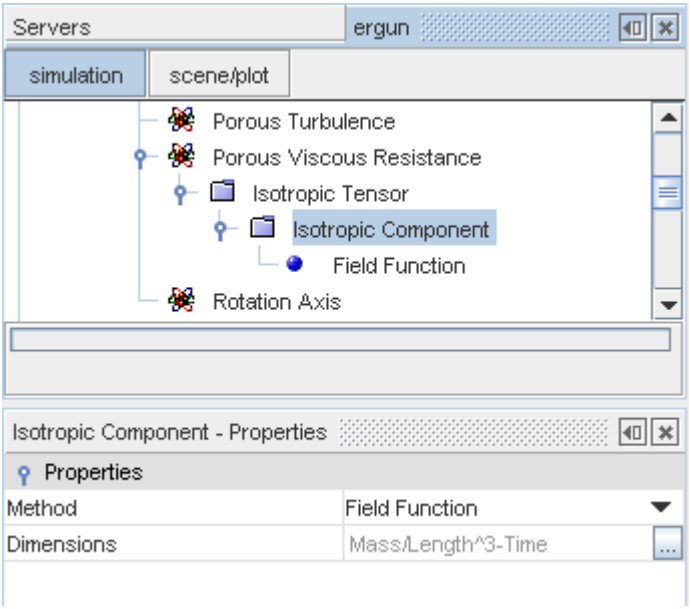


This field function is applied to the **Porous Viscous Resistance** node as an isotropic tensor.

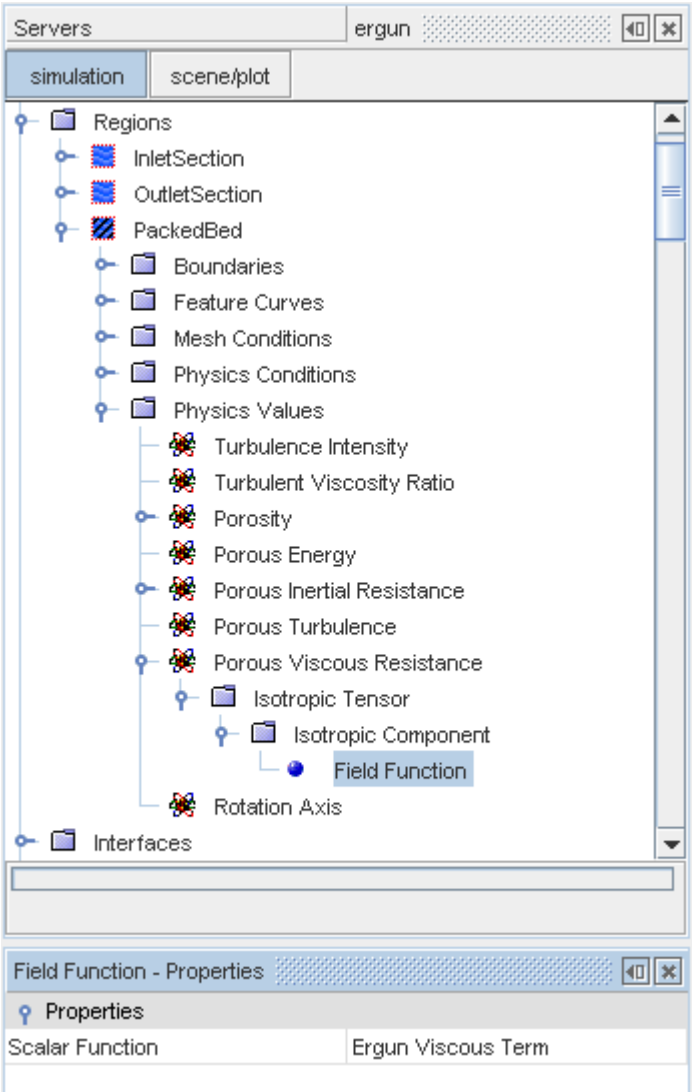
1. Select the **Physics Values** > **Porous Viscous Resistance** node on the porous region.



2. In the *Properties* window, set the *Method* to **Isotropic Tensor** using the drop-down menu. The **Isotropic Tensor** child node is shown underneath the **Porous Viscous Resistance** node.
3. Select the **Isotropic Tensor** > **Isotropic Component** node and, in the *Properties* window, set the *Method* to **Field Function**.



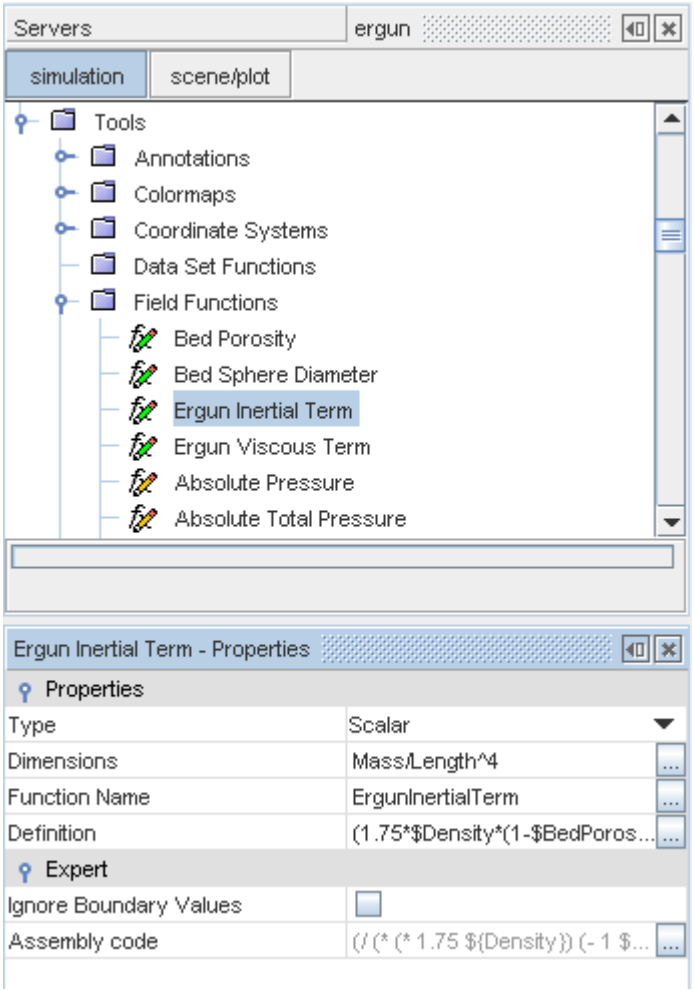
4. Finally, select the **Isotropic Component** > **Field Function** node and choose **Ergun Viscous Term** from the *Scalar Function* drop-down menu in the *Properties* window.



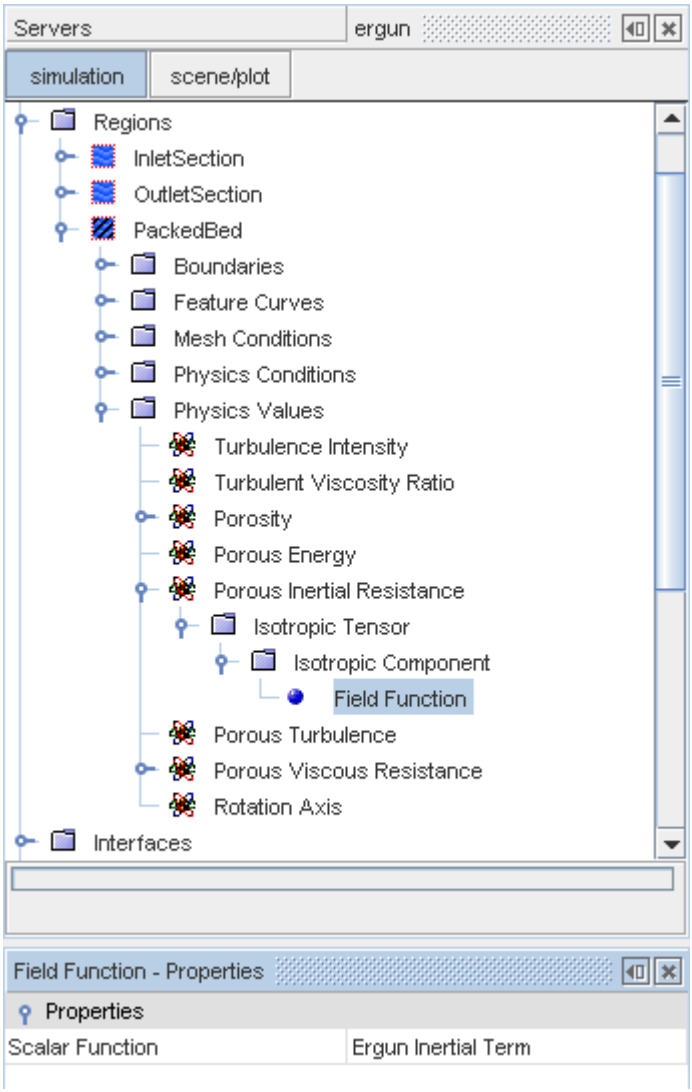
Inertial Term Field Function

The scalar field function for the inertial term is dimensioned as $Mass \cdot Length^{-4}$, and the *Definition* is:

```
(1.75*$Density*(1-$BedPorosity))/
(pow($BedPorosity,3)*$BedSphereDiameter)
```



The fluid density is included in this field function. Any variations in the fluid density therefore affect the inertial resistance. This function is applied to the **Porous Inertial Resistance** using the isotropic tensor method as before:



Region Porosity

For an unsteady calculation, the volume porosity χ can be set on the region using the **Bed Porosity** field function. To set the volume porosity:

- Select the **Physics Values** > **Porosity** node of the porous region (**PackedBed** in this example).
- In the *Properties* window, set the *Method* property to **Field Function** using the drop-down menu.
- Select the **Porosity** > **Field Function** node and, in the *Properties* window, set the *Scalar Function* property to **Bed Porosity**.

Influence of Porous Resistance on Temperature for Incompressible Flow

When simulating incompressible flow through a porous medium with the energy model activated, a change in temperature proportional to the user-specified porous resistance takes place. This section explains the reason for this temperature change.

Consider a one-dimensional steady incompressible flow of a fluid with constant c_v (specific heat at constant volume) through a porous medium with a constant cross-section. Mass conservation requires that the velocity u is constant. Momentum conservation reduces to a balance between the pressure gradient dp/dx and the porous resistance R:

$$R = \frac{dp}{dx} \tag{369}$$

The energy equation reduces to:

$$\frac{dH}{dx} = 0 \tag{370}$$

and since total enthalpy is $H = c_v T + p/\rho + u^2/2$, the equation becomes:

$$\frac{dT}{dx} = \frac{-1}{\rho c_v}\frac{dp}{dx} = \frac{-R}{\rho c_v} \tag{371}$$

Hence, it is normal to expect a temperature increase or decrease for an incompressible flow through a porous medium.

Using Porous Media to Model Flow in Tubes

In most porous media the additional source terms in the momentum equation, which are used to create the streamwise pressure gradient, dominate the convection and viscous fluxes. For this reason, the convection and viscous fluxes are not explicitly neglected in STAR-CCM+.

To omit explicitly the convection and viscous fluxes in porous media, the **Porous Media Flux Option** is available under the **Physics Conditions** node of porous regions. This option is deactivated by default. When it is activated, the fluxes are omitted. The flow solver is less robust with this option activated, since approximations are made in the linearization of the discretized equations. Therefore it is recommended that the default setting is used for truly porous media.

In some situations, you can model the flow in a tube without properly resolving the flow. In this case, if the porous media tensor coefficients are used to specify the pressure drop, they are unlikely to dominate the convection and viscous fluxes. The resulting pressure drop does not precisely balance the specified body force. By explicitly neglecting the convection and viscous fluxes, the momentum equation reduces to:

$$\frac{dp}{dx} = -\left(P_i v^2 + P_v v\right) \tag{372}$$

where P_i and P_v are the equivalent scalar coefficients of the inertial and viscous porous resistance tensors, respectively. See Eqn. (366), Eqn. (367), and Eqn. (368).

This relation is often erroneously termed “Darcy’s Law,” which in fact expresses a linear relationship between the flow rate and the pressure drop through porous media such as groundwater flow. Similar linear relations exist for fully developed, incompressible, laminar pipe flow, where the pressure drop per unit length can be expressed as a function of the mean velocity u, dynamic viscosity μ, and pipe radius r_0:

$$\frac{\Delta P}{\Delta L} = -\frac{8\mu u}{r_0^2} \tag{373}$$

($u = Q/A$, where Q is the volumetric flow rate and A is the cross-sectional area of the pipe.)

In this type of flow, the viscous fluxes dominate. These fluxes could be neglected by using a slip-wall boundary condition. The convective fluxes cancel each other out, unless there are significant entrance effects, in which case the pipe-flow relation can become non-linear in their presence. The convective fluxes also become more significant in the presence of varying fluid properties, such as in compressible flow.

Solving the Adjoint

The adjoint method is an efficient means to predict the influence of many design parameters and physical inputs on some engineering quantity of interest, that is, on the engineering objective of the simulation. In other words, it provides the sensitivity of the objective (output) with respect to the design variables (input).

Examples of the types of problems to which the adjoint method is applicable are:

- What effect does the shape of a duct (input variable) have on the pressure drop (objective)?
- What is the influence of inlet conditions (input variable) on flow uniformity at the outlet (objective)?
- What areas of the airfoil surface (input variable) have the biggest impact on lift and drag (objectives)?

The advantage of the adjoint method is that the computational cost for obtaining the sensitivities of an objective does not increase with an increasing number of design variables. The computational cost is essentially independent of the number of design variables because the adjoint method requires only a single flow solution and a single adjoint solution for any number of design variables.

The flow adjoint equations form a linear system that is typically solved by means of an iterative defect-correction algorithm. The cost of solving the linear system of equations is similar to solving the primal flow solution in terms of iterations and computational time.

The workflow for performing an adjoint analysis is typically the following:

- Set up, run and visualize the flow and/or heat transfer problem as usual.
- Choose the engineering objective for the problem at hand.
- Set up and run the adjoint analysis with regard to the chosen objective.
- Visualize the adjoint solution to obtain sensitivity information with regard to the chosen objective.

There are several different types of engineering objectives (cost functions) available.

Contents:

Setting up an Adjoint Flow Analysis
Enhancing the Coupled Flow Solver Convergence Behavior
Creating the Adjoint Cost Functions
Running the Adjoint Flow Solver
Computing Adjoint Error Estimates
Troubleshooting the Adjoint Flow Solver
Setting up a Shape Optimization
Creating Initial Control Point Positions
Specifying Mesh Deformation Boundary Conditions
Computing the Shape Sensitivity
Obtaining New Control Point Positions

Computing the Mesh Deformation
Obtaining Boundary Condition Value Sensitivity
Visualizing the Adjoint Flow Solution
Visualizing the Mesh Adjoint
Visualizing the Surface Sensitivity
Adjoint Cost Functions Reference
Point Set Reference
Adjoint Field Functions

Adjoint Analysis Workflow

An adjoint analysis is performed in two major steps. First, you simulate the steady problem using the coupled flow model, or both the coupled flow and coupled energy models. The result of this simulation is referred to as the "primal solution". The primal solution can involve isothermal fluid flow, fluid flow with energy, conduction in solids, or conjugate heat transfer between fluids and solids. In a following step, you set up and run the adjoint analysis starting from the previously obtained primal solution.

The robustness of the adjoint solution relies heavily on the fidelity of the primal flow and energy field. This dependency means that the primal solution must be converged to machine precision whenever possible. To enhance the convergence of the primal solution, create a mesh of good quality before you start the simulation.

You are advised to use a double precision version of STAR-CCM+ for the simulation.

To set up an adjoint analysis:

1. Create a suitable, high-quality volume mesh.

   Check the mesh quality and improve it if necessary.

   For more information, see Checking the Volume Mesh

   One measure that is useful for improving the quality of polyhedral meshes is the mesh optimizer. To improve the quality of a polyhedral mesh, increase the number of optimization cycles and increase the quality threshold. The mesh optimization leads to a longer meshing time.

   For more information, see Volume Mesher Controls and Values
2. Set up a steady-state flow and/or a heat transfer analysis using models and features compatible with the adjoint solver.
   a) To set up a coupled flow analysis, for the fluid physics continuum, select the following physics models:

Group Box	Model
Space	**Two Dimensional** or **Three Dimensional**
Time	**Steady**
Material	**Gas** or **Liquid**
Flow	**Coupled Flow** **Note:** As part of the coupled flow model, adjoint is also compatible with the AUSM+ flux vector splitting discretization scheme.
Equation of State	**Ideal Gas** or **Constant Density**
Viscous Regime	Any option.

   Adjoint analysis is only valid for an inviscid or laminar viscous regime. If the turbulent viscous regime is selected, the turbulent viscosity is treated as a passive quantity. Similarly, for porous regions, porosity and porous resistances are treated as passive quantities.

b) To set up a coupled solid energy analysis compatible with adjoint, for the solid physics continuum, select the following physics models:

Group Box	Model
Space	**Two Dimensional** or **Three Dimensional**
Time	**Steady**
Material	**Solid**
Optional Models	**Coupled Solid Energy**
Equation of State	**Constant Density**

3. Run the simulation until it converges, ideally down to machine precision.

   For general considerations regarding a coupled flow and energy analysis, Modeling Flow and Energy Using a Coupled Approach.

   If necessary, enhance the convergence of the simulation by modifying the coupled flow solver numerical parameters. See Enhancing the Coupled Flow Solver Convergence Behavior.

4. If you are running a flow solution, select the adjoint flow model:

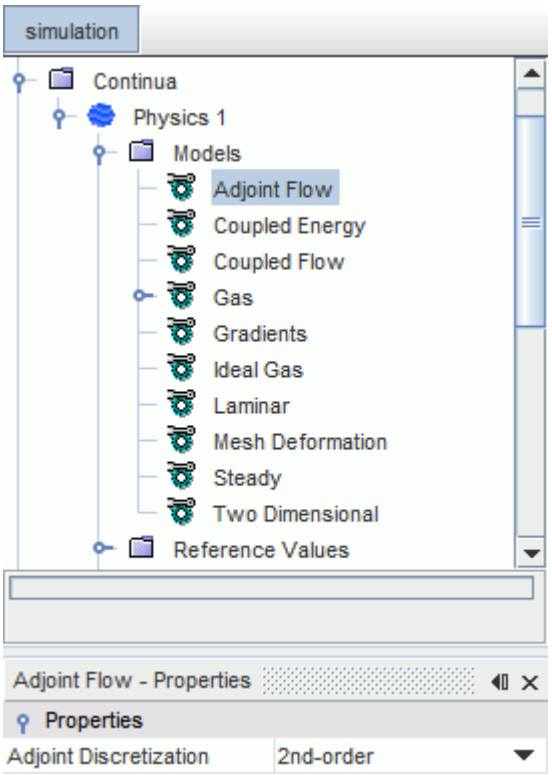
   a) Right-click the **Continua** > **Physics 1** > **Models** node and select **Select models...**

   b) In the *Physics Model Selection* dialog, select **Adjoint Flow** from the *Optional Models* group box.

      The **Mesh Deformation** model is automatically selected when the **Adjoint Flow** model is selected.

   c) Click **Close**.

      The **Adjoint Flow** node and the **Mesh Deformation** node appear under the **Models** node in the simulation tree.



d) In the *Adjoint Flow - Properties* window, set the discretization scheme to use for the adjoint flow model.

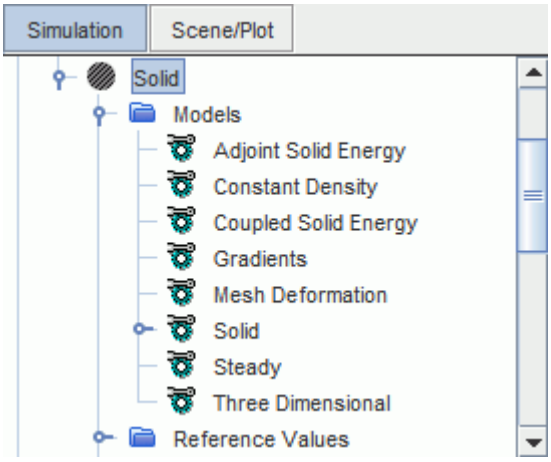
Adjoint Discretization	The discretization scheme that is used for the adjoint flow model.	
	2nd-order	This setting is the default. Use the second order discretization scheme for best accuracy of the adjoint solution.
	1st-order	Use the first order discretization scheme in the event the adjoint solver does not converge with the default second order setting. This setting makes the adjoint solver less accurate but more robust.

5. If you are running a heat conduction analysis in solids, select the adjoint solid energy model:

   a) For the solid physics continuum, **Continua** > **[Solid Continuum]**, select **Adjoint Solid Energy** from the *Optional Models* group box.

   The **Mesh Deformation** model is automatically selected.

   The **Adjoint Solid Energy** and the **Mesh Deformation** nodes appear in the simulation tree.



6. Create one or more adjoint cost functions.

   See Creating the Adjoint Cost Functions.

7. Run the adjoint flow analysis using the adjoint solver.

   Ideally, converge the adjoint simulation down to machine precision.

   See Running the Adjoint Solver

8. To visualize quickly the areas in the geometry that are most critical to the cost functions of interest, visualize the surface sensitivity.

9. Set up a shape optimization.

See Setting up a Shape Optimization.

10. Visualize the results using the adjoint field functions

    See Visualizing the Adjoint Solution and Visualizing the Mesh Adjoint.

For a complete setup of such cases, see the following tutorials:

- Adjoint Flow Solver: Flow through an S-Bend
- Adjoint Flow Solver: External Flow over a Dual Element Wing
- Adjoint Flow: Wing Shape Optimization

Enhancing the Coupled Flow Solver Convergence Behavior

For a standard flow analysis, the residuals of the flow variables are not required to converge to machine precision. However, to ensure that the adjoint solver converges in a robust manner, it is important to obtain a certain level of convergence in the primal solution.

With the default STAR-CCM+ numerical parameter settings, a satisfactory level of convergence can be reached for a standard flow analysis. For an adjoint flow analysis some additional effort is usually required.

There are some numerical measures that can be taken to enhance the convergence behavior of the coupled flow solver. These measures are not new and can be employed to improve the behavior of the primal flow analysis independent of any subsequent adjoint analysis.

- Grid Sequencing Initialization

  A reasonable way of improving the convergence behavior is to start the simulation with a good initialization of the flow field.

  For details on how to employ the Grid Sequencing Initialization, Grid Sequenzing and Expert Initialization and Setting Grid Sequencing Parameters.
- Continuity Convergence Accelerator (CCA)

  This feature can improve convergence significantly for both constant density and compressible flows.

  For details on how to employ the CCA, see Setting Convergence Accelerators.
- Expert Driver

  The Expert Driver is an automated convergence control for steady-state simulations. It can improve the robustness of the coupled flow solver and increase the speed of convergence.

  For more information, see Using the Expert Driver.
- Freezing the solution of the turbulence equations.

Contents:

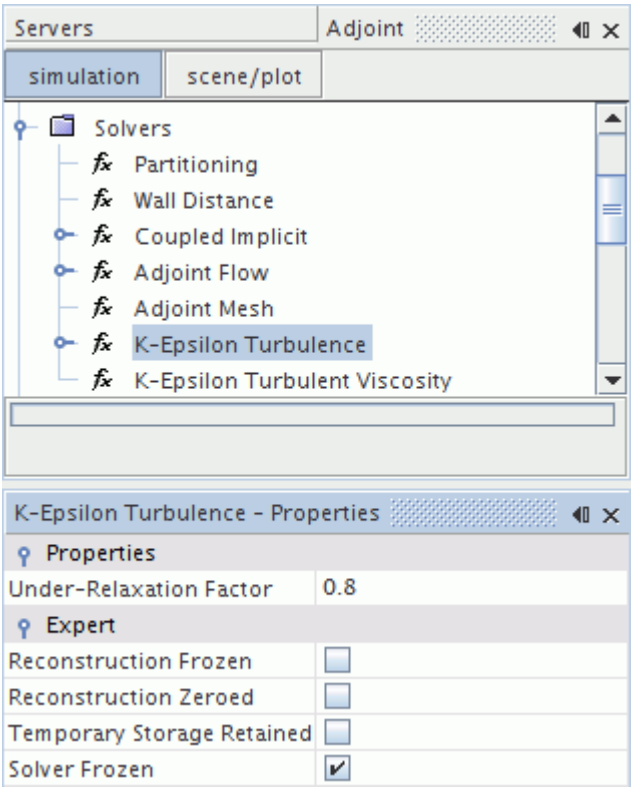
Freezing the Solution of the Turbulence Equations for Adjoint Flow Analysis

Freezing the Solution of the Turbulence Equations for Adjoint Flow Analysis

The solution of the turbulence equations can be frozen after a good level of convergence has been reached. This is a legitimate measure because sensitivity with respect to the turbulence is not being considered in the subsequent adjoint analysis.

To freeze the solution of the turbulence equations:

1. Click the **Solvers** node of the turbulence model that you have selected.
2. In the *Properties* window, activate *Solver Frozen*.



3. Continue the run to converge the residuals of the remaining variables as much as possible.

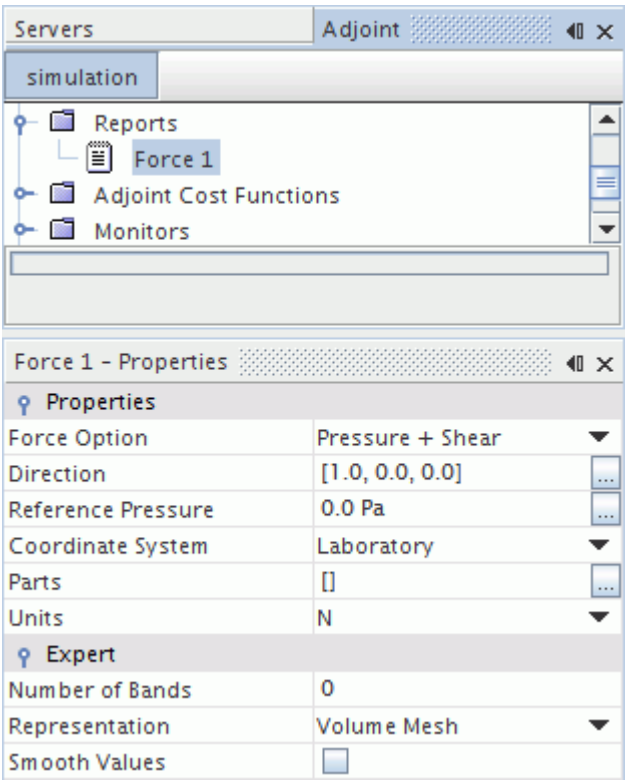
Creating the Adjoint Cost Functions

You define at least one cost function before you start the adjoint analysis. You can also create additional cost functions within the same run.

All cost functions require prior creation of respective reports. These reports serve as input for the cost functions.

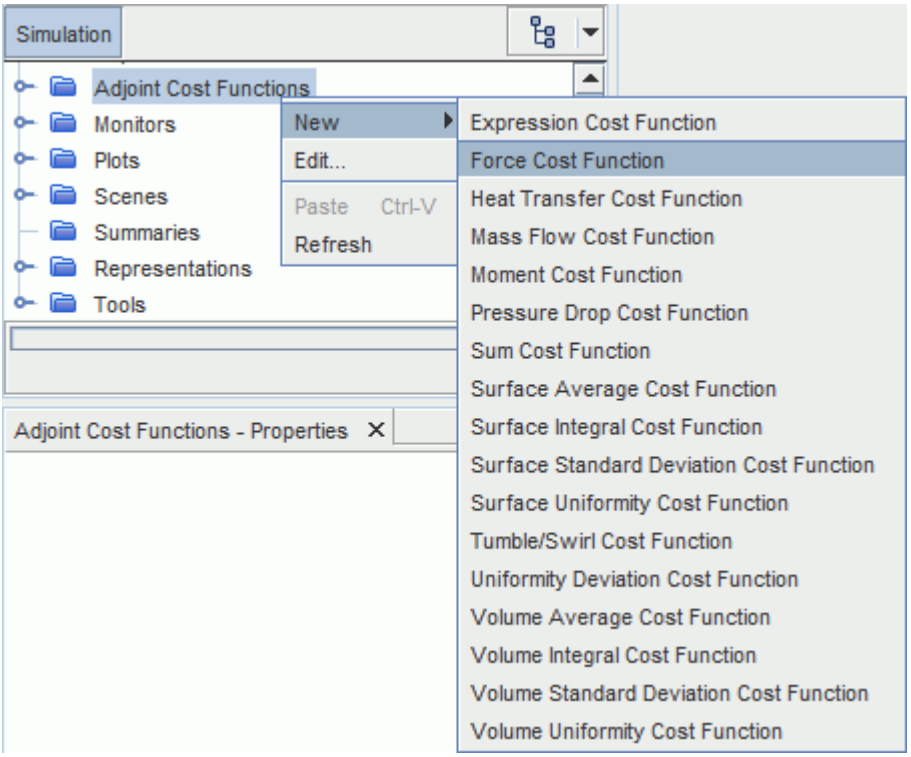
Before you create a cost function:

1. Create a report of the respective type. For example, for the force cost function you must first create a force report.



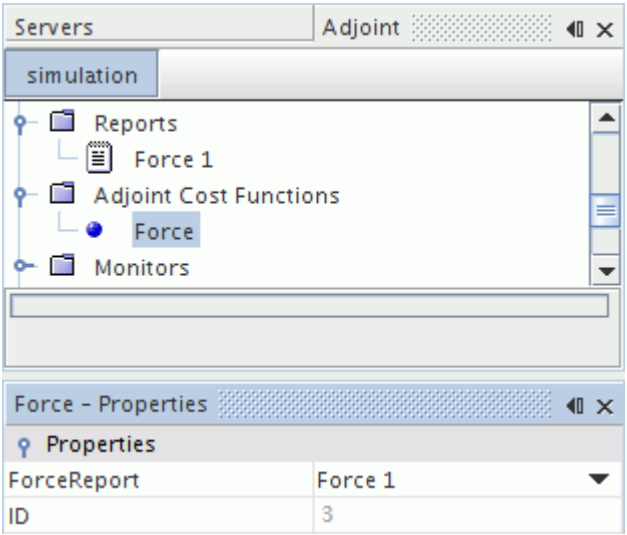
To create cost functions:

2. Right-click the **Adjoint Cost Functions** node and select **New** > **[Cost Function]**.



A new node for the cost function is created.

3. Select the cost function node and set *[Cost Function] Report* to the corresponding report.



See Also:

Adjoint Cost Functions Reference

Running the Adjoint Solver

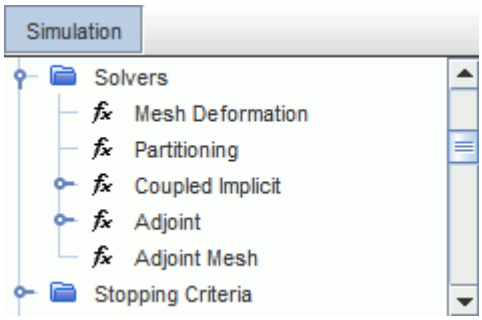
The adjoint solver solves the linear system of equations for the flow and the solid energy adjoint.

The adjoint equations are solved by means of an iterative defect-correction algorithm (see Eqn. (4093)). To enhance convergence, two GMRES solution drivers are available.

After you have activated the adjoint flow model and/or the adjoint solid energy model and set up the adjoint cost functions, you can run the adjoint solver. The adjoint solver is started and controlled separately from the coupled flow and the coupled solid energy solvers. It is not recommend that you run the coupled flow solver or the coupled solid energy solver while any of the adjoint models is active.

To run the adjoint solver:

1. Select the **Solvers** > **Adjoint** node and set the Adjoint Solver Properties.



2. The AMG linear solver controls the solution of the linear system of equations during the adjoint iterations. The **AMG Linear Solver** node is configured as for any standard flow analysis (Algebraic Multigrid). Usually, no changes to the default settings are required.
3. Right-click the **Adjoint** node and select **Step** or **Run**.

Contents:

GMRES Krylov Solver

Adjoint Solver Properties

The adjoint equations are solved with an iterative defect-correction algorithm.

Adjoint Acceleration Selection	Selects the solver to use to obtain the adjoint solution.	
	None	Selects the defect-correction algorithm. It is the default option.
	Restarted GMRES	Selects the defect-correction algorithm in combination with the Restarted generalized minimal residual GMRES solution driver for cases that are difficult to converge. The

		solution driver accelerates convergence of the adjoint flow algorithm. For more information, see GMRES Solution Driver Properties.
	Flexible GMRES	Selects an enhanced defect-correction algorithm in combination with the Flexible GMRES solution driver for cases that are difficult to converge. Use this option for adjoint problems where the default defect-correction algorithm or the Restarted GMRES solution driver do not provide acceptable rates of convergence. For more information, see GMRES Solution Driver Properties. The enhanced defect corrections procedure is based on the idea of minimizing the L2-norm of the Adjoint (linear) problem in the direction that is given by the defect correction update at each step. As such, this enhancement improves the rate of convergence in general by computing the appropriate step and by scaling the defect correction update to move the Adjoint solution to the local minimum. In some cases, where the computed defect correction direction is not accurate enough, this enhanced procedure can show a slowdown with respect to the standard defect-correction procedure or it can stall.
Courant Number	Adjust the Courant (CFL) number to under-relax and control convergence similarly to standard coupled flow analysis.	

GMRES Solution Driver Properties

For the solution of the linear system of equations for the flow adjoint, you can use the GMRES solver as an alternative to the default defect correction scheme. Using this solver adds to computation time and memory overhead.

If you select the **Restarted GMRES** or the **Flexible GMRES** option in the *Adjoint - Properties* window, the **Restarted GMRES(m) Driver** node or the **Flexible GMRES(m) Driver** node, respectively, is added to the simulation tree.

Verbose	Provides additional information on the GMRES solver in the *Output* window.
Krylov Space Dimension	Specifies the number of orthogonal Krylov space vectors to use. Increase this number to enhance GMRES-driver convergence and

	robustness. However, increasing this number increases the memory requirement.
Number of correction sweeps	Number of smoothing (enhanced) defect correction iterations. If the default value of 1 does not provide sufficient smoothing for the Krylov solver, you are advised to increase this number. A value of 3 smoothing defect corrections has proven efficient for cases with difficult convergence.
Relative Tolerance	Relative tolerance is used as an additional condition for the GMRES-driver inner-loop convergence and restart.

Computing Adjoint Error Estimates

Adjoint Error Estimates highlight locations in the flow where high numerical inaccuracy impacts the accuracy of the respective cost-function. You compute Adjoint Error Estimates for specific cost functions, (such as pressure drop in a channel or airfoil drag), after the Adjoint solution is complete. You access Adjoint Error Estimates through field functions.

Before you begin to compute Adjoint Error Estimates, the solution must have:

- A converged second-order flow solution
- Appropriate cost functions
- A converged second-order Adjoint solution with respect to the desired cost function

The primal solution is computed using a second-order discretization scheme. Using this second-order solution, STAR-CCM+ computes the Adjoint Error Estimates as follows:

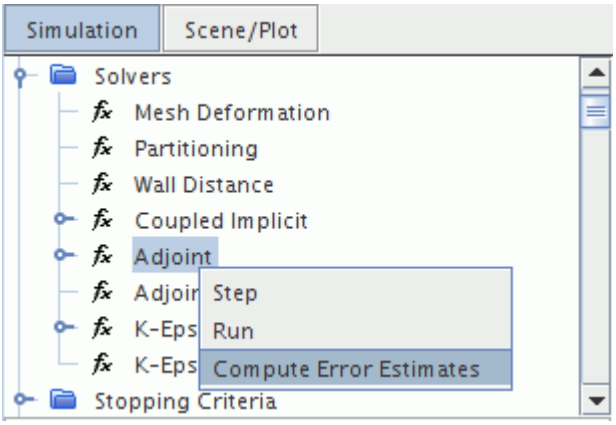
1. Recomputes the convective flux of the Navier-Stokes equations by using the MUSCL third-order/CD discretization scheme.
2. Subtracts the second-order computed cell-residual from the third-order MUSCL cell-residual.

   This difference is a simple estimate of the true numerical error that is present in the second-order solution. The true numerical error of the solution is theoretically calculated by using a higher-order discretization scheme and a fine mesh.
3. To obtain the Adjoint Error Estimate for each cell, the field of differences in cell-residual from the previous step is multiplied with the Adjoint solution for a given cost function.

Adjoint Error Estimates provide useful information for performing mesh adaption, that is, refinement or coarsening of the mesh. This information is related to a specific cost-function that is associated with that particular error estimate. The estimate can potentially provide an optimally adapted mesh with a minimum cell count and a maximum cost function accuracy.

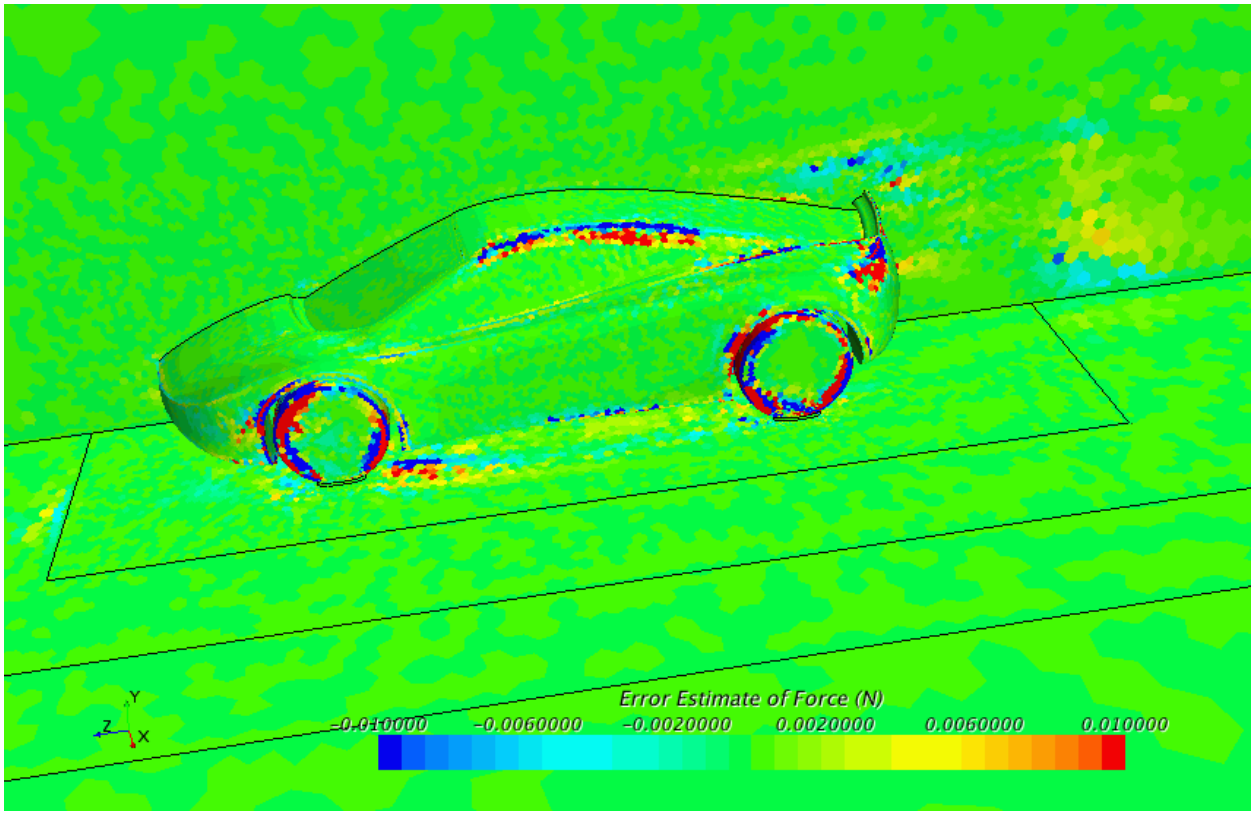
To compute Adjoint Error Estimates for all defined cost functions:

1. Right-click the **Solvers** > **Adjoint** and select **Compute Error Estimates**.



   For each defined cost function, an **Adjoint Error Estimates** > **Error Estimate of [Cost Function]** field function becomes available.

Here, you see a contour plot of the Adjoint Error Estimate of the cost function **Force** (here defined as force component in the Z direction corresponding to drag). This cost function is applied to the simulation of aerodynamics around a car.



In locations where the error estimate deviates from zero by a certain threshold, the mesh can be refined to improve the drag.

To analyze the total influence of the discretization error on a cost function:

2. Create a Sum Report and set *Scalar Field Function* to **Error Estimate of [Cost Function]**.

   This sum report gives an estimate of the cost function variation as a result of changing the flow convection order of accuracy from second order to MUSCL.

Troubleshooting the Adjoint Solver

The convergence rate of the linear adjoint solution is usually similar to that of the primal coupled solution. It takes roughly as many iterations to converge at a similar computational cost.

Similar to a standard flow simulation, it is possible that the iterative process does not converge or even diverges.

One reason for difficult convergence behavior can be the mesh quality. Even though the coupled primal solution converges, the mesh quality might not be sufficient for the adjoint solution to converge.

- If you suspect that this could be the case, then perform the mesh quality checks as described in Checking the Volume Mesh. If necessary, improve the cell quality of the volume mesh and re-run the primal solution.

There are some numerical measures that you can take to improve the adjoint solver convergence.

- Reduce the Courant number of the adjoint solver.
- Change the default *Adjoint Driver Selection* to **Restarted GMRES**. This solver increases the robustness of the adjoint algorithm at the cost of increased computational time and memory overhead. For more information, see GMRES Krylov Solver.
- Reduce the order of the adjoint flow model discretization scheme. This option is available for both the Defect Correction as well as the Restarted GMRES solver.

Setting Up a Shape Optimization

Shape sensitivity data is the product of the sensitivity of a specific cost function with respect to the mesh vertex position and the sensitivity of the mesh position with respect to either displacement boundaries or control point positions. This data represents to what degree a change in a boundary displacement vertex or control point position will affect a given cost function.

The calculation of the shape sensitivity is based on the previously obtained adjoint solution. By adding or subtracting a small fraction of the shape sensitivity to either the displacement boundary vertex positions or the control point final positions, the mesh can be deformed to increase or reduce the cost function, respectively. With the new mesh, you can rerun the primal and adjoint solutions in sequence to analyze the solution again.

To set up a shape optimization within the adjoint framework:

1. Specify the initial design point positions.

   See Creating Initial Control Point Positions.
2. Specify the mesh deformation boundary conditions.

   See Specifying Mesh Deformation Boundary Conditions
3. Compute the shape sensitivity with respect to the initial design points.

   See Computing the Shape Sensitivity.
4. Compute new design point positions that are based on the previously computed shape sensitivities.

   See Obtaining New Control Point Positions
5. Compute the mesh deformation and deform the mesh accordingly.

   See Computing the Mesh Deformation.
6. Rerun the primal and adjoint solutions.

   See Setting Up an Adjoint Flow Analysis.

Creating Initial Control Point Positions

Control points for the morpher are specified using a Point Set. You create the initial control point positions for the shape sensitivity and mesh deformation calculations by using the *Create Point Set* dialog.

Once you create the control points, the Point Set must be associated with the physics continuum that contains a mesh deformation model. When associated with a physics continuum, you specify the movement of the control points through the **Morpher: Displacement** profile under the **Point Sets** node.

Note: To compute the shape sensitivity and the mesh deformation directly for the vertices of the boundary, use the morpher boundary type **Displacement**. For more information, see Specifying Mesh Deformation Boundary Conditions.

When you position control points around a boundary to be deformed, locate them close enough to have a sufficiently strong influence on it, but far enough away from the boundary to ensure smooth deformations to it. For an airfoil, for example, a good starting point is to place 10 to 20 control points strategically at a distance of 20% of its chord length from the boundary.

To create the initial control point positions:

1. Right-click the **Point Sets** node and select **New Point Set**, then one of the following options:
   - **Table**...
   - **Lattice**...
   - **Line**...
   - **Part**...
2. In the respective *Create Point Set* dialog, make the appropriate point set settings, click **Create**, then **Close**.
3. Select the **[Table, Lattice, Line, Part] Point Set** node and set the *Physics Continuum* to the physics continuum for which the mesh deformation is active.

   If the point set is assigned to a continuum with a mesh deformation model, it will be included in the calculation of the mesh deformation solver.
4. Select the **Point Sets** > **[Table, Lattice, Line, Part] Point Set** > **Physics Values** > **Morpher: Displacement** node and set the appropriate profile method.

Specifying Mesh Deformation Boundary Conditions

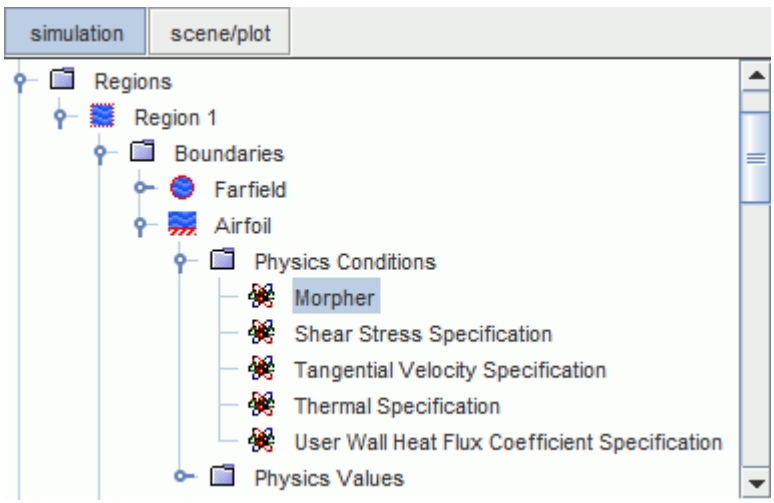
The morpher is used to update and deform the mesh according to the new control point positions (control point displacement) or the vertex positions at a displacement boundary that are computed based on the mesh adjoint.

For more details on how the morpher works, see Working with Morphing.

At each boundary of your computational domain, specify the conditions of how they are morphed. Consider, for example, the simulation of an airfoil within a wind tunnel. Only the airfoil is subject to adjoint shape optimization with the vertices at the boundary being morphed. The surrounding wind tunnel boundaries remain fixed.

To specify mesh deformation boundary conditions:

1. Select the **Regions** > **[Region]** > **Boundaries** > **[Boundary]** > **Physics Conditions** > **Morpher** node.



2. In the *Morpher - Properties* window, set the appropriate properties.

   See Mesh Deformation Morpher Properties

Contents:

Mesh Deformation Morpher Properties
Mesh Deformation Morpher: Thin Factor Properties

Mesh Deformation Morpher Properties

Method	Defines how the vertices at the boundary are morphed. For more details on the morpher boundary methods, see Morpher Boundary Conditions Reference.	
	Fixed	The boundary remains fixed. The points on this boundary are treated

		as design points with zero displacement. Selecting this option adds the **Morpher: Thin Factor** node to the **Physics Values** node of the boundary. Specify a value less than 1 for the *Vertex Thin Factor* to override auto-thinning from the mesh deformation solver.
	In Plane	Points on this boundary are allowed to slide in the plane of the boundary. Selecting this option adds the **Morpher Plane Options** node to the **Physics Values** node of the boundary.
	Floating	Points on this boundary move according to the computed displacement of the design points.
	Displacement	Vertices on this boundary are collectively viewed as design points. They move according to the computed displacement of the mesh deformation solver. Selecting this option adds the **Morpher: Thin Factor** node to the **Physics Values** node of the boundary. Specify a value less than 1 for the *Vertex Thin Factor* to override auto-thinning from the mesh deformation solver.

Mesh Deformation Morpher: Thin Factor Properties

Damping Factor	Displacements at the boundary are damped smoothly to zero starting at the specified distance. This factor is value L_p in Eqn. (143) . The default value of 0.0 means that the distances are calculated automatically.	
Continuity Level	Specifies which damping function to use in smoothing vertex displacements approaching the plane.	
	C0	The displacement is damped using Eqn. (141). Only this option is valid when evaluating the shape sensitivity.
	C2	The displacement is damped using Eqn. (142).

Ignore Factor	This factor is multiplied by the **Damping Factor** to calculate a finite distance. Control Points closer than the specified distance to this plane are given special treatment. A value of zero is acceptable.

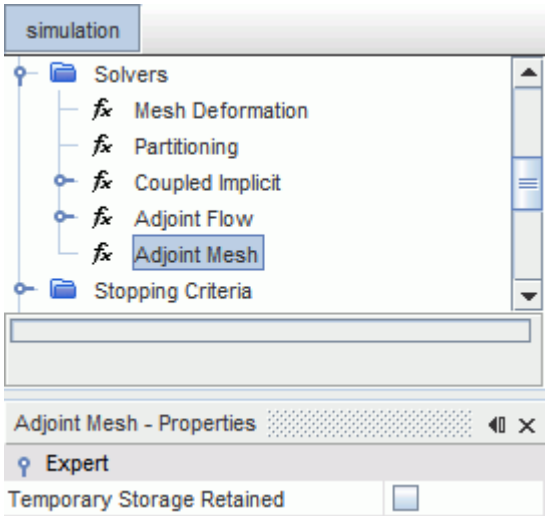
Computing the Shape Sensitivity

The shape sensitivity gives the sensitivity of the cost function with respect to the control points or displacement boundaries.

The evaluation of the shape sensitivity is a non-iterative two-step process. The mesh sensitivity is first computed and then, if the **Mesh Deformation** model is available, the shape sensitivity is solved.

To compute the shape sensitivity:

- Right-click the **Solvers** > **Adjoint Mesh** node and select **Compute Mesh Sensitivity**.



To visualize the results of the computation of the shape sensitivity:

- Visualize the mesh adjoint at displacement boundaries. For more information, see Visualizing the Mesh Adjoint.
- Visualize the mesh sensitivity at all the vertices. For more information, see Computing the Mesh Sensitivity. To make this option available, select the **Adjoint Mesh** node and activate **Temporary Storage Retained**.

If auto-thinning is disabled and the number of design points exceeds the maximum number of recommended design points as specified in the Mesh Deformation node, you receive a warning message.

To remedy this warning, you have the following options:

- Enable auto-thinning.

  See Computing the Mesh Deformation.
- Decrease the Thin Factor at the Fixed Morpher boundaries.

  See Specifying Mesh Deformation Boundary Conditions.
- Decrease the number of design points in the specified design point table.

  See Creating Initial Control Point Positions.

- Increase the maximum number of recommended design points and ignore the warning.

  See Computing the Mesh Deformation.

Obtaining New Control Point Positions

Based on the computed shape sensitivity at the control points, you compute the new positions of the control points.

This computation of the displacement of the control points from their initial position is typically performed by using an external (third-party) optimization program such as LBFGS. You provide tables containing the control point positions and the shape sensitivity at these control points to the external optimization program. You create an internal table that uses the previously created control points as inputs. For more information, see Creating Initial Control Point Positions.

To compute the new control point positions:

1. Right-click the **Tools** > **Tables** node and select **New Table** > **XYZ Internal Table.**
2. Select the **XYZ Internal Table** node:
   a. Set *Parts* to control points that you created previously.
   b. Set *Scalars* to **Adjoint of [Cost Function] w.r.t. Position**.
3. Right-click the **XYZ Internal Table** node and select **Extract**.
4. Right-click the **XYZ Internal Table** node and select **Export...**.

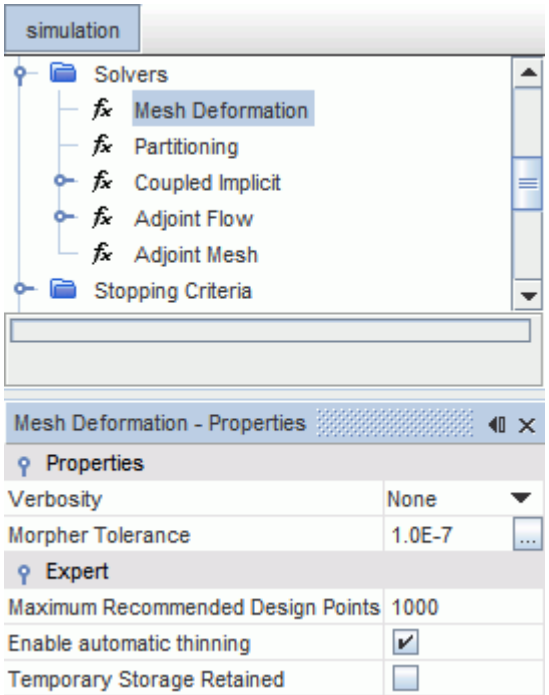
   You export the sensitivity data associated with the control point locations to a .csv file.
5. Import the shape sensitivity table into the optimization program of your choice.
6. With the information from the imported tables, compute the control point displacement using the optimization program.
7. From the optimization program, save the control point displacements to a [`Name of control point table].csv` file.
8. Right-click the **Tools** > **Tables** node and select **New Table** > **File Table** > **[Name of control point table].csv.**
9. Right-click the **Point Sets** > **[Point Set]** > **Physics Values** > **Morpher: Displacement** node and set *Method* to **Table(x,y,z).**
10. Select the **Morpher: Displacement Table (x,y,z)** node and set the table properties for the **[Name of control point table]**.

    The morpher uses the new control point positions to deform the mesh.

Computing the Mesh Deformation

To compute the mesh deformation:

1. Select the **Solvers** > **Mesh Deformation** node.



2. In the *Mesh Deformation - Properties* window, set the appropriate properties.

See Mesh Deformation Properties.

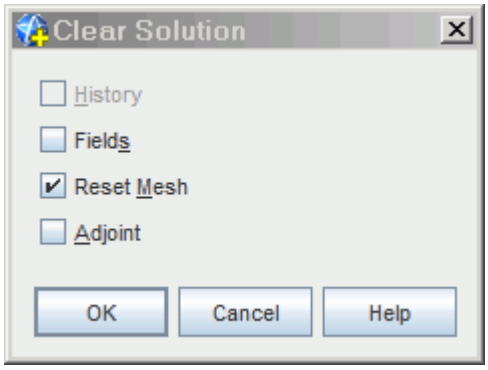
3. Right-click the **Solvers** > **Mesh Deformation** node and select **Deform Mesh**.

If the number of design points exceeds the maximum number of recommended design points as specified in the Mesh Deformation node, you receive a warning message. To remedy this warning, you have the following options.

You can visualize the mesh deformation in any suitable scene showing the volume mesh of the morphed geometry.

To undo the mesh deformation:

4. Select **Solution** > **Clear Solution...**
5. In the *Clear Solutions* dialog, select **Reset Mesh**.



6. Click **OK**.

Contents:

Mesh Deformation Properties

Mesh Deformation Properties

Verbosity	Provides more information on the mesh deformation operation to the Output window.	
	None	No additional output.
	Low	Additional information about deformation point positions.
	High	Additional information about numerical parameters of mesh deformation morphing operation.
Morpher Tolerance	Non-dimensional parameter that is used to specify acceptable accuracy in morphing. This parameter is multiplied by the longest edge length of the bounding box enclosing the morphing region.	
Maximum Recommended Design Points	The recommended upper limit for the number of fixed and displacement design points. The default is 1000. A recommended maximum number of design points is specified, because an excessive high number of design points can become expensive computationally. When the specified limit is exceeded, a warning is issued before the execution of a **Compute Mesh Sensitivity** and **Deform Mesh** operation if auto-thinning is disabled. For more information, see Computing the Shape Sensitivity and Computing the Mesh Deformation .	
Enable Automatic Thinning	Activated by default. Automatically thins out fixed and displacement boundary vertices for mesh deformation morphing.	

Obtaining Boundary Condition Value Sensitivity

You can obtain information about the sensitivity of the cost functions, for example pressure drop or drag, with respect to boundary input parameters such as free-stream Mach number, stagnation temperature, outflow pressure, and other imposed boundary conditions.

This information is available when you execute an adjoint-specific report where you specify the boundary parameter with respect to the cost functions for which the sensitivity is computed.

Before you begin to compute the boundary condition value sensitivity:

- Set up at least one cost function.

  See Creating the Adjoint Cost Functions.
- Complete at least one adjoint iteration step.

  See Running the Adjoint Flow Solver.

To obtain boundary condition value sensitivity:

1. Right-click the **Reports**node and select **New Report Boundary Parameter Sensitivity**.
2. Select the **Boundary Parameter Sensitivity 1** node.
3. In the *Boundary Parameter Sensitivity 1 - Properties* window, set the following parameters:

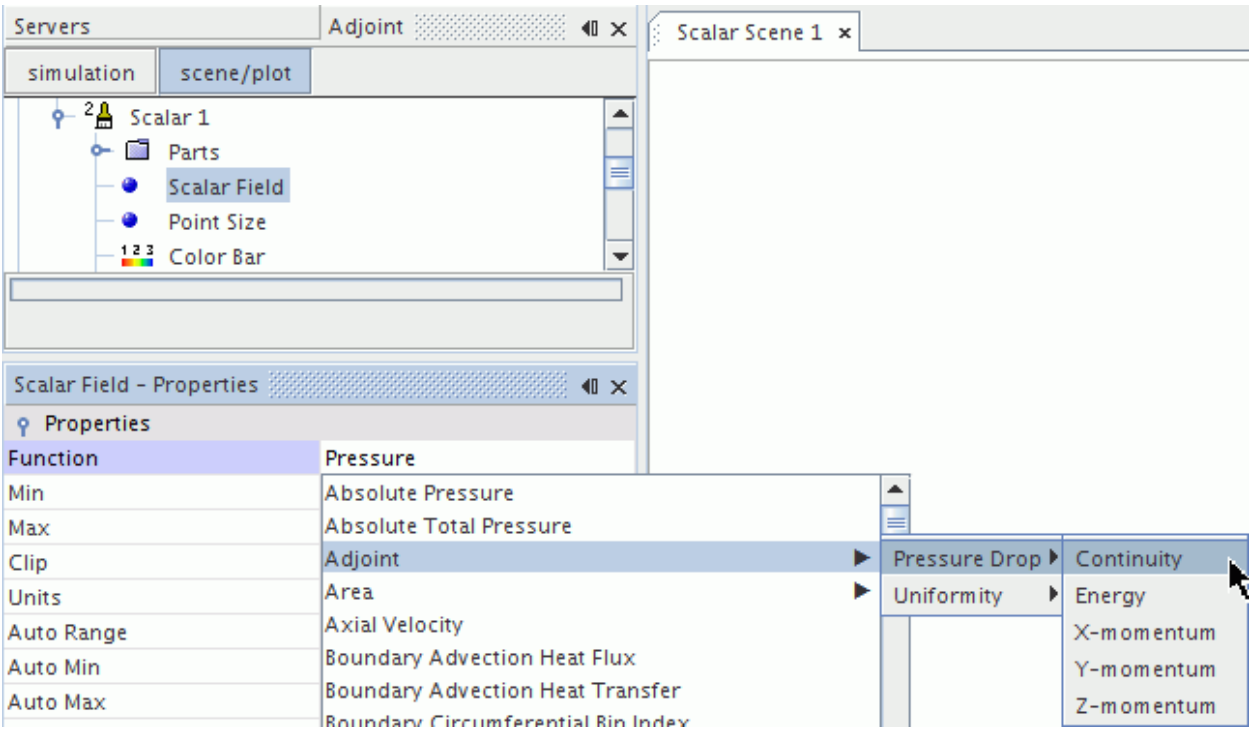
Adjoint Cost Function	Select the adjoint cost function for which you want to obtain the boundary condition value sensitivity.
Boundary Parameter	Select the boundary parameter with respect to which you want to obtain the sensitivity.
Parts	Select the input parts onto which you want to evaluate the boundary parameter sensitivity.
Units	Select the units of the previously selected *Adjoint Cost Function*.
Representation	Select a representation. Only the Volume Mesh representation is a valid choice.
Smooth Values	When you activate this property, vertex interpolated values are used.

Visualizing the Adjoint Solution

Specific adjoint field functions are available to visualize the adjoint solution.

To visualize the adjoint solution:

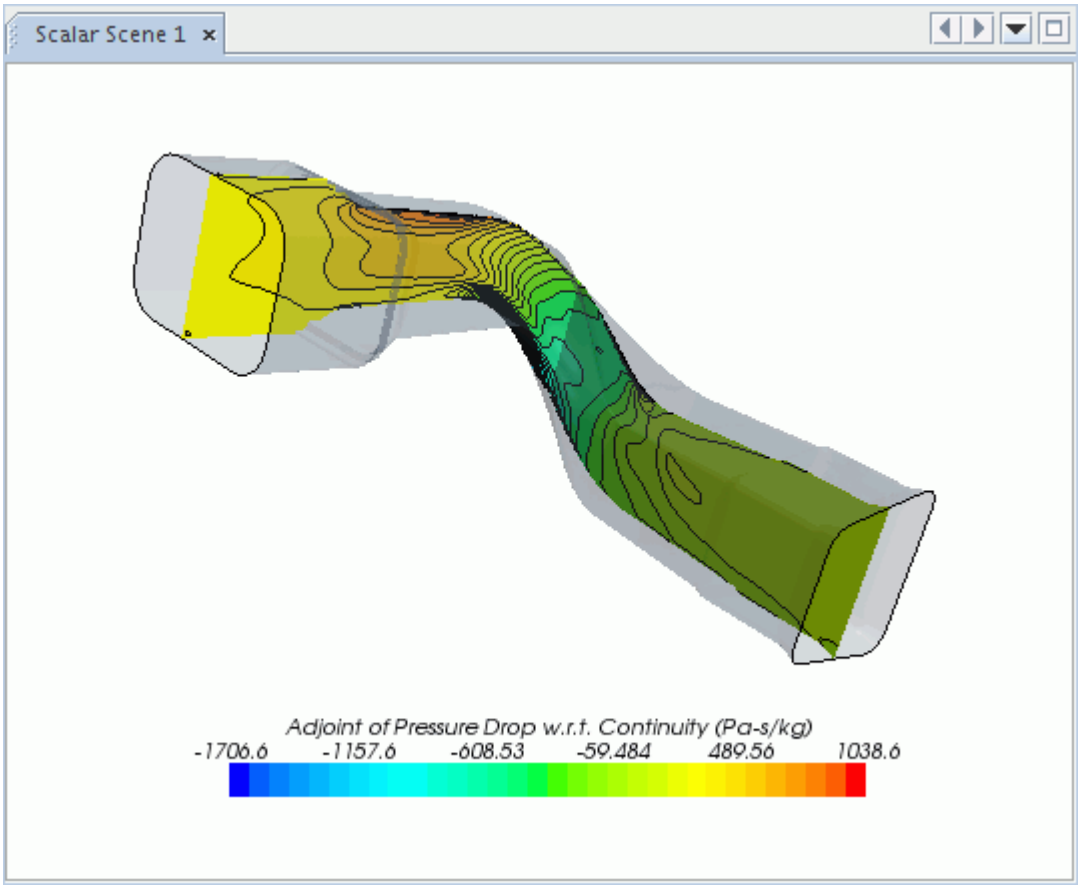
1. Create a scalar scene. See Creating a Scene.
2. Click the *scene/plot* button.
3. In the scalar scene, select the **Scalar 1** > **Scalar Field** node.
4. In the *Properties* window, click on the field next to *Function*. In the drop-down list, select **Adjoint**. For each of the previously specified cost functions, you can select a menu item.



The adjoint <cost function> field functions are available with regard to continuity, energy, X-, Y- and Z-momentum.

The adjoint solution is displayed in the currently open scalar scene in the *Graphics* window.

An example is shown below:



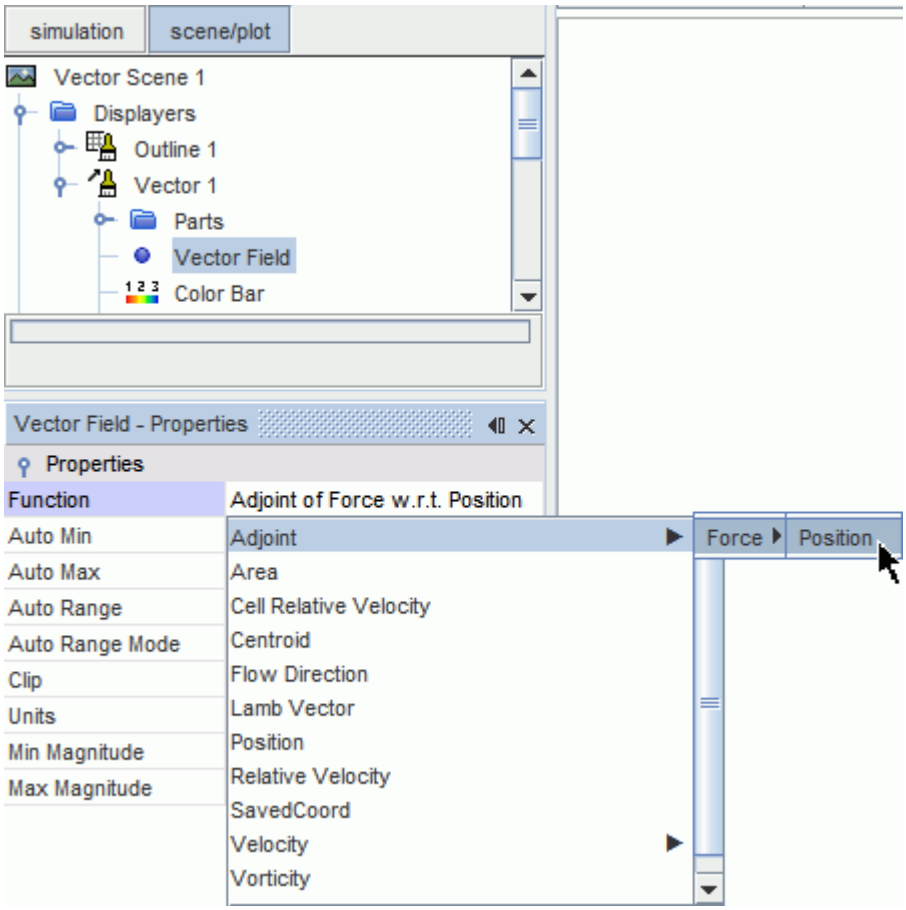
You can see the adjoint of the pressure drop with regard to continuity. This means that the plot shows you the sensitivity of the cost function, in this case the pressure drop, with regard to mass. (The continuity equation corresponds to the mass conservation equation.) According to Eqn. (4092), the flow adjoint represents the sensitivity of the cost function with regard to the flow residual. Consequently, the adjoint with regard to continuity residual indicates that directing mass away from areas with positive adjoint values decreases the cost function, in this case, the pressure drop. While directing mass towards areas with positive adjoint values, increases the cost function. Red areas in the plot indicate that directing mass away from these areas has the most effect on lessening the pressure drop. Blue areas in the adjoint plot also indicate a high sensitivity of the cost function with regard to mass. Directing mass away from areas with negative adjoint values increases the cost function. You can apply this interpretation of results to the other residuals, that is to the momentum and energy residuals, as well.

Visualizing the Mesh Adjoint

Specific sensitivity field functions are available to visualize the mesh adjoint.

To visualize the mesh adjoint:

1. Create a vector scene.
2. Click the *scene/plot* button.
3. In the vector scene, select the **Vector 1** > **Vector Field** node.
4. In the *Properties* window, set *Function* to the appropriate Adjoint cost function with respect to position.



The mesh adjoint is displayed in the currently open vector scene in the *Graphics* window.

Visualizing the Surface Sensitivity

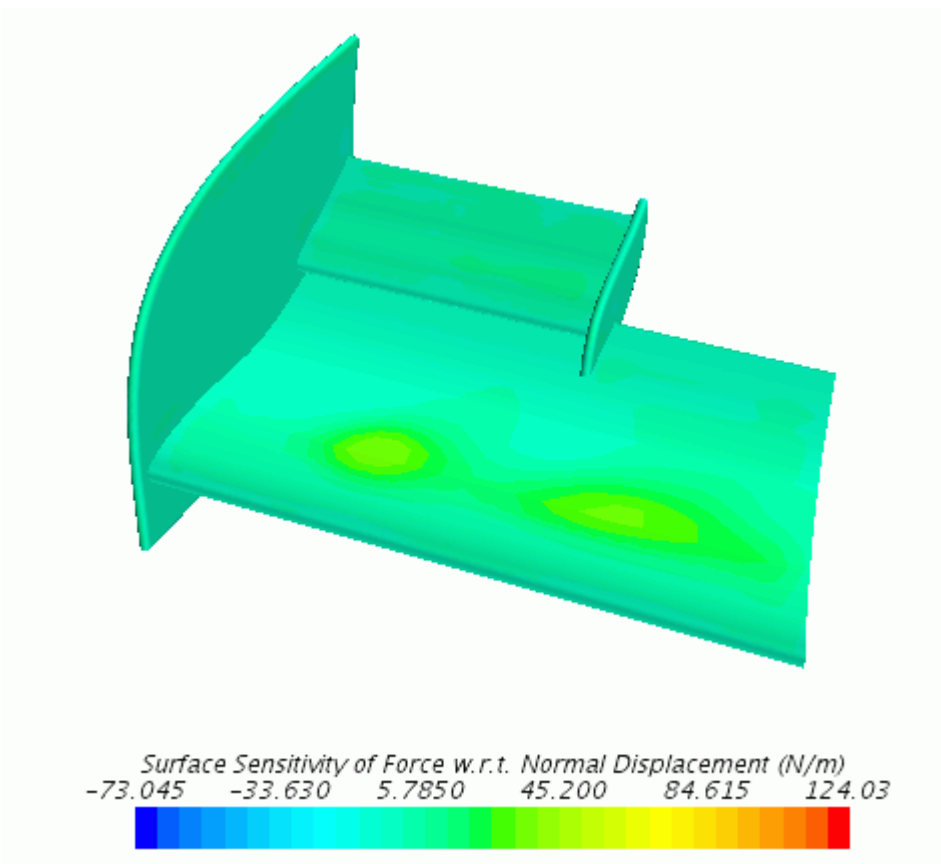
The surface sensitivity model allows you to identify which areas of geometry are most critical to the cost functions of interest without performing an actual shape optimization. To reduce computation time, the model computes approximate sensitivities with respect to surface position for the selected adjoint cost functions.

Before you begin, you must have the following:

- converged primal flow solution
- converged adjoint solution with respect to the defined cost functions

To visualize the surface sensitivity:

1. Right-click the **[Physics Continuum]** > **Models** node and select **Surface Sensitivity** from the *Optional Models* group box.
2. Right-click the **Solvers** > **Surface Sensitivity** node and select **Compute Surface Sensitivity**.
3. Create a scalar scene.
4. Select **Displayers** > **Scalar 1** > **Scalar Field** and set *Function* to **Surface Sensitivity of [Cost Function]**.
   The following example shows the surface sensitivity of the drag force on a wing:



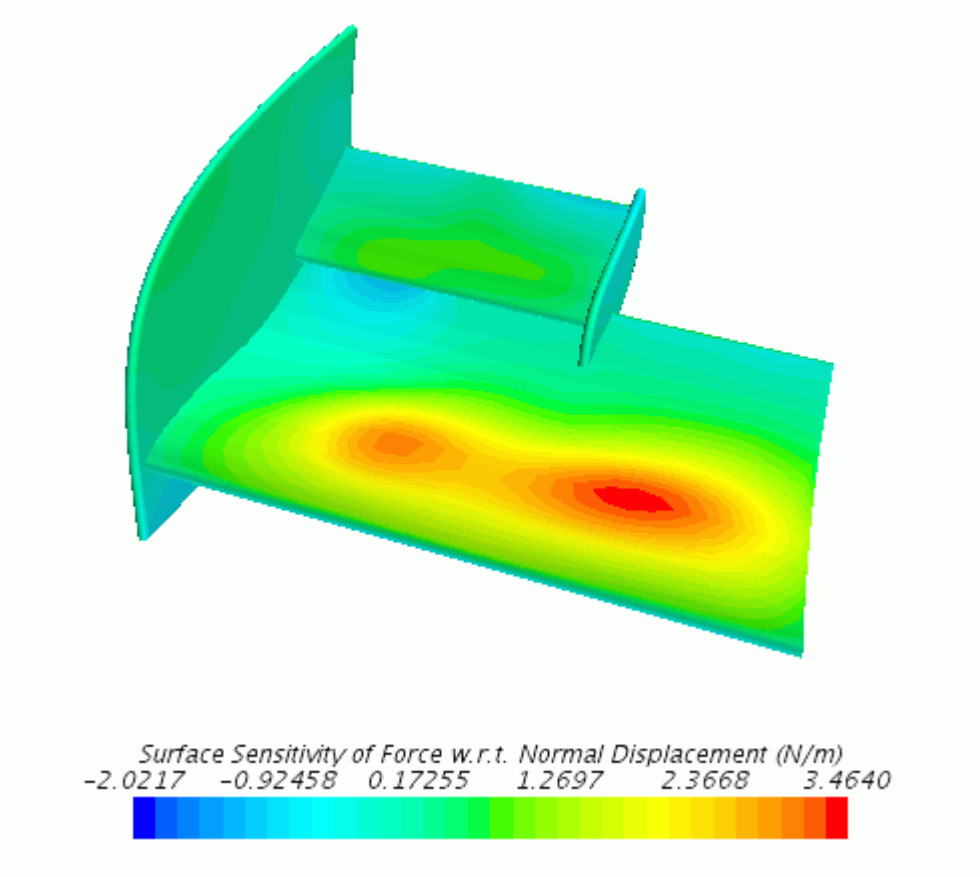
5. (Optional) If the surface sensitivity contour plot is noisy, select the **Solvers** > **Surface Sensitivity** node and increase the *Sensitivity Smoothing Factor*.

The default value is 0.

6. Right-click the **Solvers** > **Surface Sensitivity** node and compute the surface sensitivity again.

   To remove high-frequency noise while leaving the general solution intact, the smoothing step filters the raw surface sensitivities through a Poisson-type filter. The amount of smoothing to apply depends on the particular case.

   The following image shows the surface sensitivity of the drag force of the same wing as above with the *Sensitivity Smoothing Factor* set to 1.



Adjoint Cost Functions Reference

An adjoint cost function corresponds to the engineering objective that is examined during the adjoint flow analysis. One example of an engineering objective is to reduce the pressure drop in a piping system. Another example is the reduction of drag force around a car.

Cost functions are differentiated to form the right-hand side of the adjoint flow equations. The adjoint solver solves the system of differential equations with regard to the cost function specified.

All cost functions are associated with a report.

Adjoint Cost Functions	Associated Report
Force Cost Function Estimates the sensitivity of the force on a specified set of boundary surfaces in a specified direction.	Force
Moment Cost Function Estimates the sensitivity of the moment on a specified set of boundary surfaces with a specified direction and origin.	Moment
Pressure Drop Cost Function Estimates the sensitivity of the mass-flow averaged total pressure between two boundary surfaces or between two interfaces representing the high-pressure and low-pressure boundaries of the system.	Pressure Drop
Uniformity Deviation Cost Function Estimates the sensitivity of the deviation of the local normal velocity from the mass-flow averaged value on a specified set of boundary surfaces or on interfaces.	Uniformity Deviation Report
Tumble and Swirl Cost Function Estimates the sensitivity of tumble or swirl, usually for ICE applications.	Tumble/Swirl Report
Expression Cost Function Estimates the sensitivity of an expression that is defined through an expression report.	Expression Report
Mass Flow Cost Function Estimate the sensitivity of mass flow rate through boundaries.	Mass Flow
Heat Transfer Cost Function Estimates the sensitivity of heat transfer at boundaries.	Heat Transfer
Sum Cost Function Estimates the sensitivity of a summed up scalar quantity over surfaces or volumes.	Sum

Surface Average Cost Function Estimates the sensitivity of the surface average of a scalar quantity over surfaces.	Surface Average
Surface Integral Cost Function Estimates the sensitivity of the surface integral of a scalar or vector quantity.	Surface Integral
Surface Standard Deviation Cost Function Estimates the sensitivity of the standard deviation of a scalar quantity across a surface.	Surface Standard Deviation
Surface Uniformity Cost Function Estimates the sensitivity of the uniformity of a scalar quantity on a surface.	Surface Uniformity
Volume Average Cost Function Estimates the sensitivity of the volume average of a scalar quantity.	Volume Average
Volume Integral Cost Function Estimates the sensitivity of the volume integral of a scalar quantity.	Volume Integral
Volume Standard Deviation Cost Function Estimates the sensitivity of the standard deviation of a scalar quantity in a volume.	Volume Standard Deviation
Volume Uniformity Cost Function Estimates the sensitivity of the uniformity of a scalar quantity in a volume.	Volume Uniformity

There are certain limitations for reports that require the input of a field function and parts such as the sum or the surface uniformity report:

- When these reports are associated to cost functions, they must only reference model parts (that is, boundaries and regions); geometry parts and derived parts are not supported.
- The field functions supported for full differentiation are scalar or vector field functions which are either primitive, derived (vector magnitude, vector component), or user-defined, and are available on boundary faces or region cells. For user-defined field functions, only a subset of operations are differentiable—operations which do not involve neighbor stencil field data, tabular data, or coordinate transformations. Supported operations are: arithmetic, logical, and trigonometric operations. The complete list of user-defined field function operations that are supported is:
  - Index operator
  - Alternate Value operator
  - Conditional operator
  - Unary operators: `acos, asin, atan, cos, cosh, exp, abs, log, log10, sin, sinh, sqrt, cbrt, tan, tanh, mag, mag2, unit`
  - Binary operators: `'+', '-', '*', '/', pow, fmod, min, max, atan2, dot, cross`
  - The Field Function expert property *Inverse Distance Weight* is not supported.
  - The Report expert property *Smooth Values* is not supported.

- The following field functions are supported for adjoint:
  - `Volume`
  - `Area`
  - `Centroid`
  - `Grid Flux`
  - `Density`
  - `Temperature`
  - `Total Enthalpy`
  - `Boundary Heat Flux`
  - `Boundary Conduction Heat Flux`
  - `Pressure`
  - `Total Pressure`
  - `Absolute Total Pressure`
  - `Vorticity Vector`
  - `Helicity`
  - `Face Flux`
  - `Boundary Advection Heat Flux`
  - `Wall Shear Stress`
  - `Velocity`
  - `Relative Velocity`
  - `Mass Flux`
  - `Static Pressure`
  - `Total Temperature`
  - `Sound Speed`
  - `Mach Number`
- The following field functions are differentiable when temporary storage is retained:
  - Reconstructed gradients of
    - `Density`
    - `Pressure`
    - `Velocity`
    - `Temperature`
    - `Total Enthalpy`
  - Gradient of `Pressure`

See Also:

Creating the Adjoint Cost Functions

Point Set Reference

You define the different types of point sets through the point set dialogs.

Create Point Set from Table Dialog

Source Table	Specifies the table from which to extract the control point coordinates with respect to the laboratory coordinate system.
X Data	Associates each of these properties with their respective columns of coordinate data.
Y Data	
Z Data	

Create Point Set from Lattice Dialog

Origin	Specifies the origin of the three-dimensional lattice with respect to the laboratory coordinate system, in X, Y, Z coordinates.
Size	Specifies the width, length, and height of the lattice with respect to the origin. Define using dimensions in coordinate directions X, Y, and Z.
Rotation Axis	Specifies the axis of rotation in the laboratory coordinate system, in X, Y, and Z coordinates.
Rotation Angle	Specifies the angle by which to rotate the lattice around the Rotation Axis.
Number in X	For each direction, sets the number of controls points in the lattice in that direction.
Number in Y	
Number in Z	

Create Point Set along Line Dialog

Point 1	Defines X, Y, and Z coordinates of the line start point within the chosen *Coordinate System.*
Point 2	Defines X, Y, and Z coordinates of the line end point within the chosen *Coordinate System.*
Coordinate System	Selects the reference coordinate system in which the line start and end points are defined.
Number	The number of control points to create along the defined line.

Create Point Set from Part Dialog

Input Part	Specifies the input part relative to which a three-dimensional cloud of control points is created. The control points are created based on the vertices of that part surface.	
Representation	Specifies from which representation the control points are extracted. *Input Part* can be represented with different meshes or triangulations, so you must specify the appropriate representation from which to extract the control points. The default representation works for most cases. If you create control points along feature curves, you must specify a geometry representation to extract meaningful control points.	
Target Control Point Spacing	The approximate distance by which the control points are separated from each other. Adjusts the density of the control point cloud.	
Maximum Number of Points	Maximum number of control points to create in the control point region. If you specify a *Target Control Point Spacing* that yields too many points, a random mask is applied to the control points so that only the *Maximum Number of Points* is added to the newly created control point region.	
Offset from Part Surface	When activated, control points are displaced from the part surface by the given distance.	
	Surface Offset	The distance by which the control points are displaced along the normal of the input part surface.

Adjoint Field Functions

The following field functions are made available when computing the adjoint solution and the shape sensitivity.

Adjoint of [cost function] w.r.t.	Adjoint of the respective cost function. See $\Lambda_{\mathbf{Q}}$ in Eqn. (4092), whose components are the Adjoint w.r.t. Continuity, Adjoint w.r.t. X-, Y-, Z-momentum, and Adjoint w.r.t. Energy.	
	Continuity	The adjoint with respect to continuity. This value tells you the sensitivity of the cost function with respect to local changes in mass within the domain.
	X-, Y-, Z-momentum	The adjoint with respect to momentum. This value tells you the sensitivity of the cost function with respect to local changes in X-, Y-, and Z-momentum within the domain.
	Energy	The adjoint with regard to energy. This value tells you the sensitivity of the cost function with regard to local changes in energy within the domain.
	Position	The adjoint with respect to position. See $\frac{dL}{d\mathbf{D}}$ in Eqn. (4094). This value tells you the sensitivity of the cost function with respect to displacement boundary position.
Sensitivity of [cost function] w.r.t. Position	Sensitivity of the selected cost function with respect to position. See $\frac{dL}{d\mathbf{X}}$ in Eqn. (4091). The position refers to the coordinates of the vertices of the volume mesh. A high value (positive or negative) tells you that the respective cost function is particularly sensitive to that given vertex coordinate position.	
Adjoint Error Estimates of [cost function]	Adjoint error estimates of the respective cost function. See $AErr^L$ in Eqn. (4123).	
Surface Sensitivity of [cost function] w.r.t.	Sensitivity of the selected cost function with respect to surface position.	
	Direction	Displays the magnitude of the surface sensitivity or the selected component of the surface sensitivity vector.
	Normal Direction	Displays the dot product of the surface sensitivity vector with the inward surface normal.

Modeling Fans and Blowers

STAR-CCM+ provides models for axial and radial fans where classical fan laws apply.

STAR-CCM+ provides the following methods for modeling fans:

- The Fan Interface

  The fan interface is a simple model that represents the fan by a zero-thickness interface and imposes a pressure jump across the interface. This model can also add swirl to the downstream flow. The model is an implicit implementation that provides quicker convergence than the fan momentum source.
- The Blower Interface

  The Blower interface lets you model two types of radial fans: centrifugal fans and impeller fans.
- The Fan Momentum Source

  The fan momentum source is a more complex model than the fan interface. This model can add swirl to the downstream flow, and is applied to a region that represents the three-dimensional volume of the fan. The fan momentum source model uses an explicit implementation, and as such, is slower to converge.

The fan interface and the fan momentum source are simple models that trade accuracy for speed. To obtain more accurate results you can model the individual blades of the fan using either moving frames of reference (MRF) or mesh motion. See Advanced Methods for Modeling Fans.

Caution: The fan interface models in STAR-CCM+ are designed for electronic cooling applications that usually involve incompressible flow. Additionally, they can be used for ventilation systems involving low speed flows. Using these models for moderate to high Mach Number flows does not give the desired results.

For the theory that STAR-CCM+ uses in computing the fan interface and fan momentum sources, refer to the Fan Interface formulation and the Fan Momentum Source formulation.

If you are interested in modeling the acoustic effects of a fan, refer to recommended practices for modeling acoustic sources in axial fans.

The Fan Interface Model

The fan interface is a simple model that provides quicker convergence than the fan momentum source. It models the pressure rise through an axial fan as a function of local flow rates or velocity.

In axial fans, the outflow direction is the same as the inflow direction. However, in radial fans (or blowers), the outflow direction is not the same as the inflow direction. These types of fans are modeled using a blower interface.

The fan interface models the fan as a circular interface of zero thickness, although there is no restriction on the topology that you use. This interface type is used only between regions belonging to the same fluid continuum.

When you set up a fan interface, you supply the fan performance curve that corresponds to the physical fan that you are modeling. The fan performance curve specifies the pressure drop across the fan interface as a function of one of the following variables:

- Local velocity
- Mass average velocity
- Volumetric flow rate
- Mass flow rate.

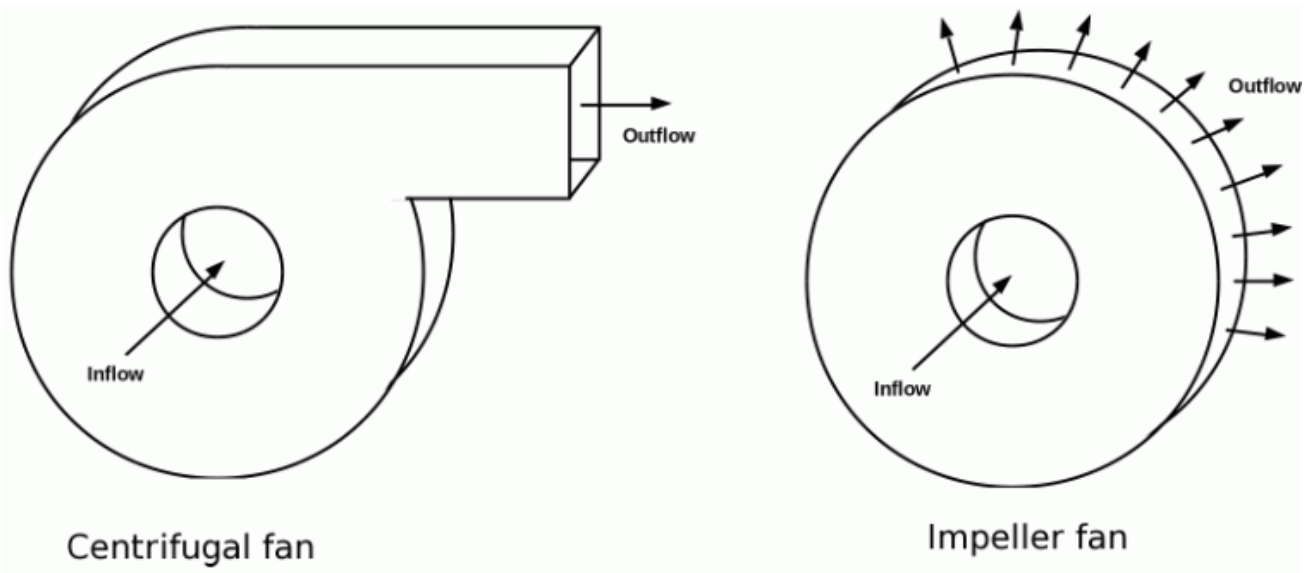
See Using the Fan Interface, Creating a Fan Interface, and Fan Interface Properties. For a worked example, see the tutorial Multi-Part Solid: Graphics Card Cooling.

The Blower Interface Model

The Blower interface lets you model two types of radial fans: centrifugal fans and impeller fans.

In axial fans, the outflow direction is the same as the inflow direction. These fans are modeled using a fan interface. However, in radial fans (or blowers), the outflow direction is not the same as the inflow direction. These types of fans are modeled using the blower interface.

The blower interface implementation in STAR-CCM+ is the same for both centrifugal fans and impeller fans, except that swirl at the outflow is allowed for impeller fans only.



For radial fans, the pressure rise is implemented across two non-contiguous boundaries (the blower inlet(s) and exit), so the Blower interface is an indirect interface.

Fan Momentum Source Model

The fan momentum source employs an “actuator disk” type of methodology.

With this methodology, the detailed geometry of the fan blades is not modeled. The circular fan disk is meshed as a separate region, and STAR-CCM+ identifies the cells that are in the volume swept out by the fan

blades. The appropriate momentum sources, including the swirl component, are applied to these cells. Analysis is steady, so blade passing effects are not included.

The fan momentum source model can simulate the most commonly designed axial fans. The model works well for adverse pressure gradients, where the fan is pushing against a back pressure. You can have more than one fan in the same simulation.

Since the momentum sources are based on theory, there is no guarantee that the resulting flow rate and pressure rise lies in the fan curve of a particular fan. You specify the appropriate fan performance curve. STAR-CCM+ automatically adjusts the momentum sources so that the operating point of the simulated fan falls on the fan curve.

The fan momentum source model uses an explicit implementation, and as such, can be slow to converge.

Advanced Methods for Modeling Fans

To model fans more accurately, you can model a region that contains the geometry of the fan blades. Use the following methods:

- Rotating Reference Frames

  A rotating reference frame is a rotating frame of reference that can be applied to regions to generate a constant grid flux.

  This approach gives a solution that represents the time-averaged behavior of the flow, rather than the time-accurate behavior. For more information, see Using Moving Reference Frames in Steady-State.
- Unsteady Sliding Mesh

  When the time-accurate behavior is required, STAR-CCM+ provides the option to move the mesh vertices of a region during a transient analysis. This mesh motion is specified using either a Rotation motion or a Rotation and Translation motion.

  Applying either of these motions results in mesh vertices being displaced in each time step. For more information, see Using Rigid Motion of Mesh Vertices. For a worked example, see the tutorial Moving Reference Frames: Rotating Fan.

Both the moving reference frame and unsteady sliding mesh approaches to fan modeling require that you create a separate region for the moving fan blades. The joins of the regions are interfaces. Conformal meshes are preferred but are not required.

You can use either trimmed cell or polyhedral meshes:

- For polyhedral meshes, all regions can use the same mesh continuum, and the mesh is conformal across the interface.
- For trimmed cell meshes, each region must have its own mesh continuum, and the mesh is not conformal across the interface.

Contents:

Overview of Fan Performance Curves
Fan Pressure Components
Fan Configurations
Modeling Fan Performance Curves

Using the Fan Interface
Creating a Fan Interface
Fan Interface Properties
Specifying the Fan Performance Curve Type
Specifying the Fan Performance Curve as a Table
Specifying the Fan Performance Curve as a Polynomial Function
Specifying the Fan Performance Curve Polynomial
Specifying the Fan Curve Temperature Scaling
Modeling the Fan Swirl
Orientation of Parts-Based Interfaces
Setting the Fan Interface Verbosity
Adjusting the Fan Interface Intersection Tolerance
Creating a Blower Interface
Blower Interface Properties
Modeling the Blower Swirl
Modeling Multiple Fans and Blowers in a Single Interface
Using the Fan Momentum Source
Importing the Fan Performance Data Table
Activating the Fan Momentum Source
Specifying the Fan Performance Curve
Fan Momentum Source Properties
Scaling the Fan Performance Curve
Working with the Fan Solver

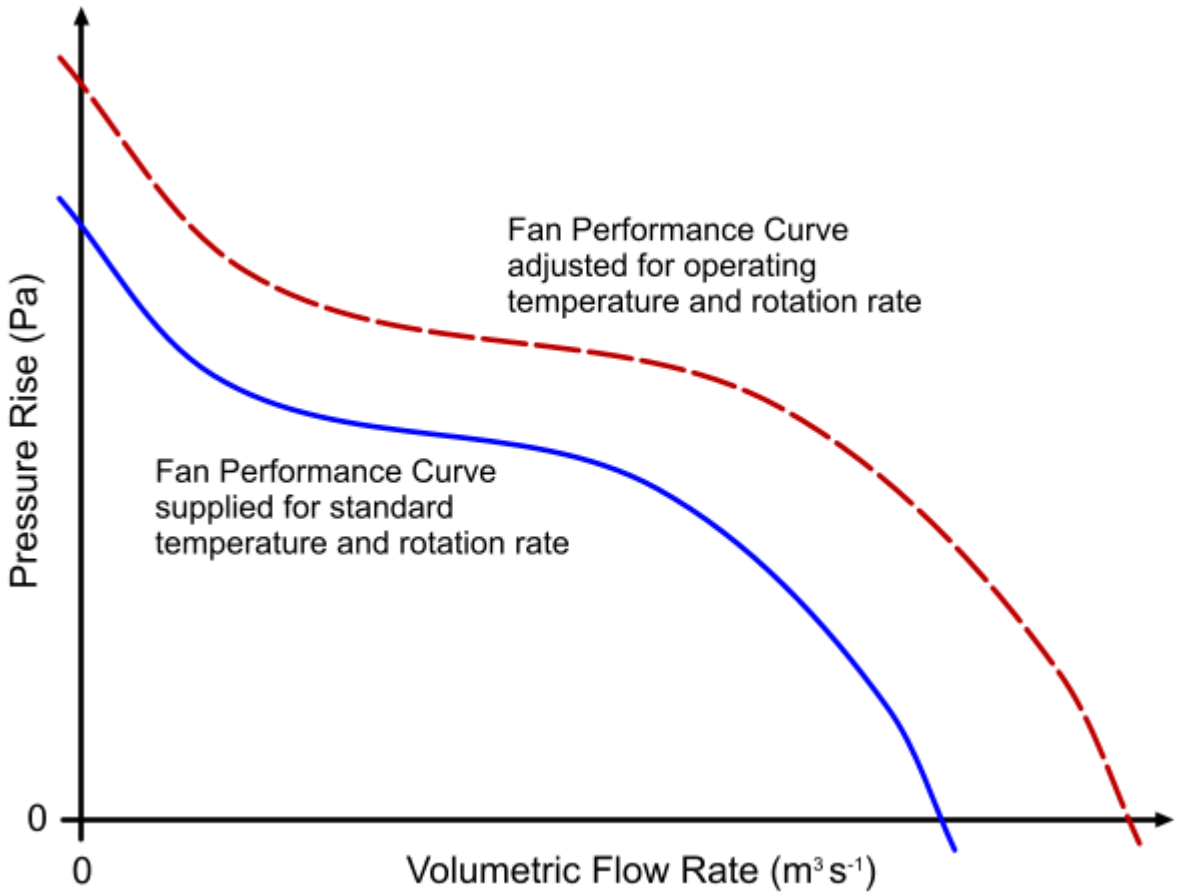
Overview of Fan Performance Curves

A fan performance curve describes the performance of a fan under specific operating conditions.

The fan performance curve typically specifies the pressure rise across the fan as a function of volumetric flow rate, for a standard fluid temperature and fan rotation rate. Fan manufacturers provide this information for each fan.

When you model a fan in STAR-CCM+, you specify the appropriate fan performance curve. To obtain the fan performance curve for a particular fan rotation rate and fluid temperature, you can make your own measurements of the fan under the appropriate conditions. Alternatively, STAR-CCM+ can derive it from the manufacturer-supplied fan performance curve: STAR-CCM+ adjusts the fan performance curve using classical fan laws to compensate for the different fan rotation rate and fluid temperature.

An example of a manufacturer-supplied fan performance curve and the same fan performance curve that is adjusted for particular operating conditions is shown below.



Fan Pressure Components

This section defines the fan pressure components that are used in deriving fan performance curves.

The fan pressure components are defined as follows:

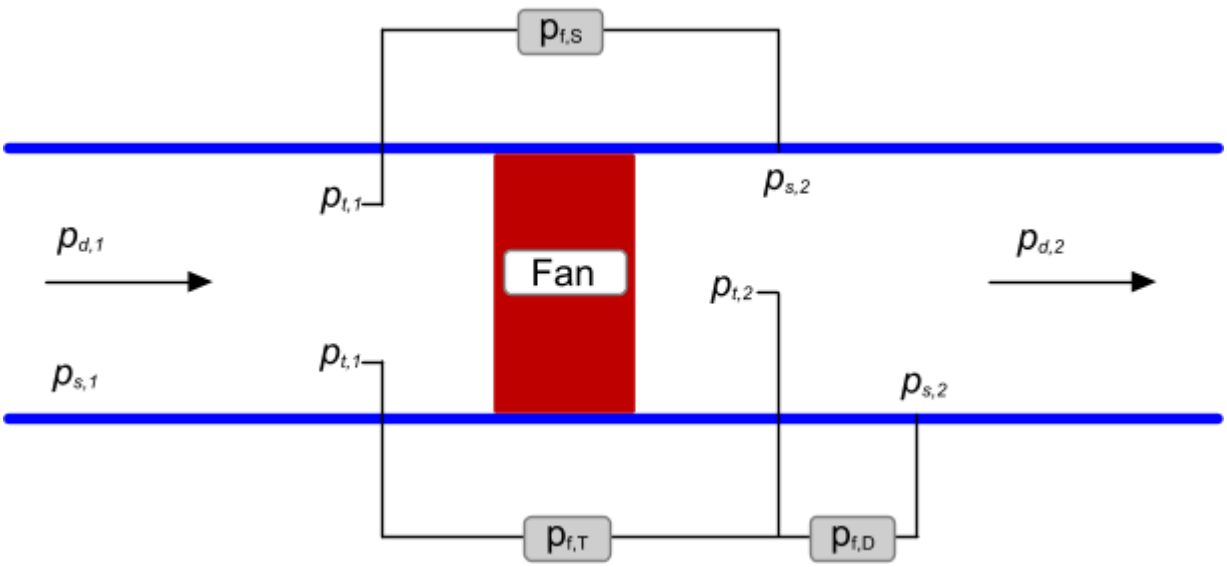
Fan Total Pressure ($p_{f,T}$)	The increase in total pressure (p_t) across the fan: $p_{f,T} = p_{t,2} - p_{t,1}$ The total pressure is the sum of the dynamic pressure (p_d) and the static pressure (p_s): $p_t = p_d + p_s$
Fan Dynamic Pressure ($p_{f,D}$)	The average dynamic pressure at the fan outlet: $p_{f,D} = p_{d,2} = p_{t,2} - p_{s,2}$
Fan Static Pressure ($p_{f,S}$)	The difference between the fan total pressure and the fan dynamic pressure: $p_{f,S} = p_{f,T} - p_{f,D}$ = $p_{t,2} - p_{t,1} - \left(p_{t,2} - p_{s,2}\right)$ = $p_{s,2} - p_{t,1}$

Fan Configurations

The three essential fan configurations are ducted fan, forcing fan, and exhausting fan.

Ducted Fan

The fan has both inlet and outlet ducts.



The dynamic pressure is non-zero at both the inlet and outlet.

In this configuration, the fan static pressure is:

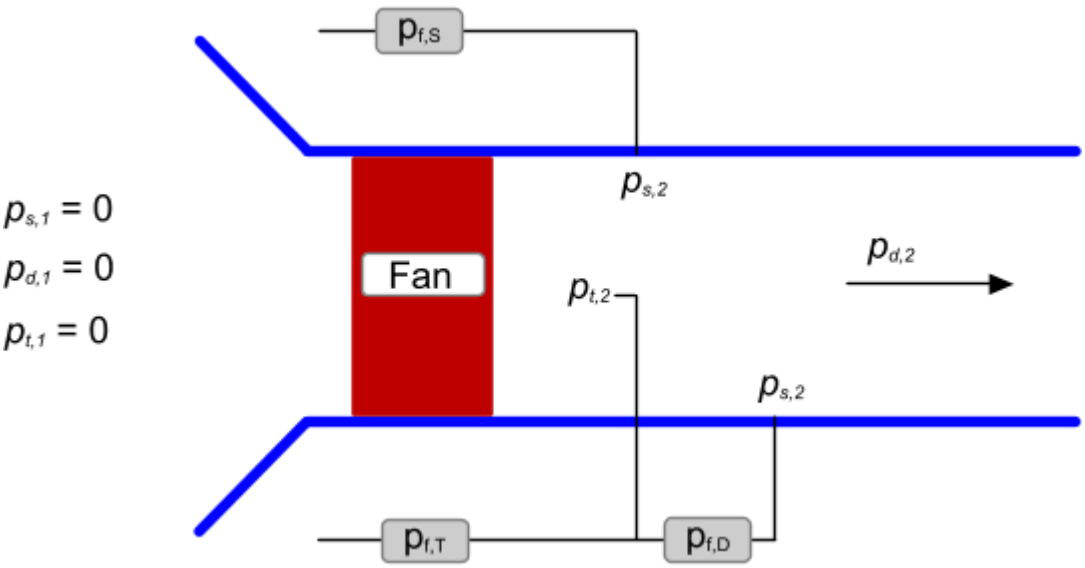
$p_{f,S} = p_{f,T} - p_{f,D}$

$= p_{t,2} - p_{t,1} - \left(p_{t,2} - p_{s,2}\right)$

$= p_{s,2} - p_{t,1}$

Forcing Fan

The fan upstream is a plenum and the downstream is a duct.



The dynamic pressure is zero at the upstream, but non-zero at the downstream.

In this configuration, the fan static pressure is:

$p_{f,S} = p_{f,T} - p_{f,D}$

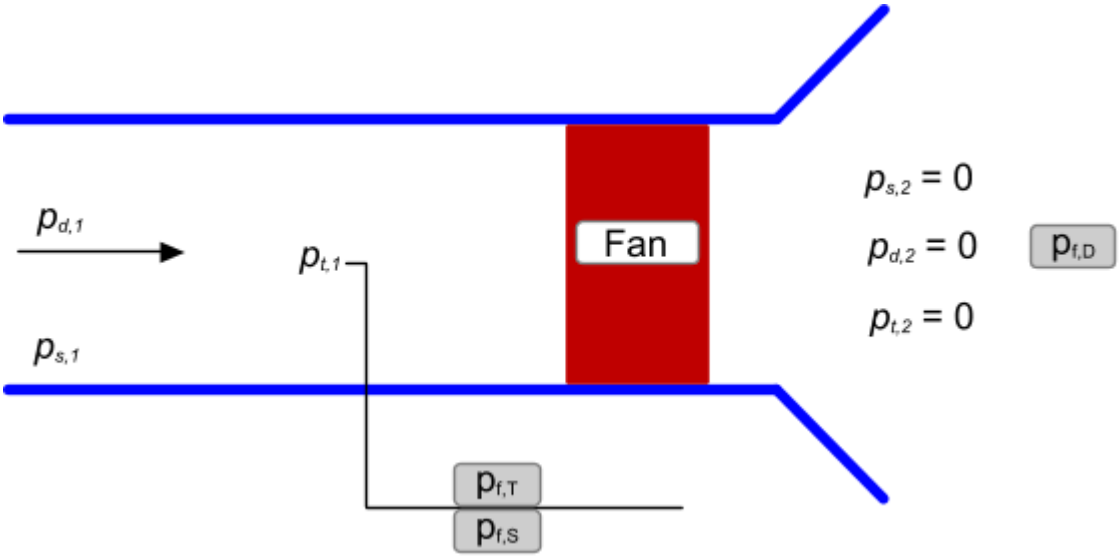
$= p_{t,2} - p_{t,1} - \left(p_{t,2} - p_{s,2}\right)$

$= p_{s,2} - p_{t,1}$

$= p_{s,2}$

Exhausting Fan

The fan upstream is a duct and the downstream is a plenum.



The dynamic pressure is non-zero at the upstream, but zero at the downstream.

In this configuration, the fan static pressure is:

$p_{f,S} = p_{f,T} - p_{f,D}$

$= p_{t,2} - p_{t,1} - \left(p_{t,2} - p_{s,2}\right)$

$= p_{s,2} - p_{t,1}$

$= -p_{t,1}$

Modeling Fan Performance Curves

The fan performance curves that are supplied by fan manufacturers plot the pressure rise across that fan against the flow rate.

The pressure rise is defined as the static pressure measured downstream of the fan minus the static pressure that is measured upstream of the fan ($p_{s,2} - p_{s,1}$), or as the static pressure measured downstream of the fan minus the total pressure that is measured upstream of the fan ($p_{s,2} - p_{t,1}$).

For more information on how fan manufacturers produce fan performance curves, refer to the appropriate industry standards:

- ANSI/AMCA Standard 210-07 | ANSI/ASHRAE 51-07

  Laboratory Methods of Testing Fans for Certified Aerodynamic Performance Rating
- BS EN ISO 5801:2008, BS 848-1:2007

  Industrial fans: Performance testing using standardized airways

STAR-CCM+ provides two options to define the pressure rise across the fan:

- *Standard*

  Defines the fan pressure rise as downstream static pressure minus upstream total pressure.
- *Static to Static*

  Defines the pressure rise as downstream static pressure minus upstream static pressure.

STAR-CCM+ allows for scaling the fan performance curves to fit new fan rotation rates and fluid temperatures. See:

- Scaling the Fan Performance Curve
- Specifying the Fan Curve Temperature Scaling

STAR-CCM+ assumes that the dynamic pressure is the same upstream and downstream so the *Static to Static* option can also be used as total pressure downstream minus total pressure upstream.

When you create a fan interface, choose an appropriate option as specified by the fan manufacturer. See Specifying the Fan Performance Curve Type.

Using the Fan Interface

This section describes how to implement a fan interface in your simulation.

Complete the following tasks:

1. Create the fan interface.

   See Creating a Fan Interface and Fan Interface Properties.
2. Specify the appropriate fan performance curve:

   See Specifying the Fan Performance Curve Type.

   - Import the fan performance curve table into the simulation.

     See Specifying the Fan Performance Curve as a Table.
   - Specify the polynomial function that represents the fan performance curve.

     See Specifying the Fan Performance Curve as a Polynomial Function.
3. If necessary, change the fan swirl, fan verbosity, and fan interface intersection tolerance settings.

   See Modeling the Fan Swirl.

   See Setting the Fan Interface Verbosity.

   See Adjusting the Fan Interface Intersection Tolerance.

Consider the following when you use a fan interface in your simulation:

- When analyzing results it is appropriate to report mass flow rates and pressures on the interface boundaries. This method is more representative than creating derived parts nearby.
- Radiation models require temperature, and an interface model of a fan has no temperature. Therefore, a Radiation model treats the space that the fan occupies as an ordinary opening.

Note: You can also use a fan performance curve to specify a pressure jump across a boundary. This option is available for Pressure Outlet and Stagnation Inlet boundaries only.

For a worked example, see the tutorial Multi-Part Solid: Graphics Card Cooling.

Creating a Fan Interface

Each interface has a direction from side 0 to side 1. When you create the interface, select side 0 first and then select side 1. For a fan interface, side 0 is upstream to side 1.

For a fan curve using a specified mass flow rate $\dot{m}$, the mass flow rate is taken as the maximum of $\dot{m}$ or 0. If you select the sides in the wrong order, the fan model computes the mass flow rate $\dot{m}$ as negative. Therefore, you get an erroneous pressure drop for the fan curve at the limit of $\dot{m} = 0$.

To create a fan interface:

1. Expand the **Regions** > **[Fluid Name]** > **Boundaries** node, select the upstream boundary (side 0), and then hold down the <Ctrl> key and select the downstream boundary (side 1).
2. Right-click either of the selected boundaries and select **Create Interface**.

   The new interface node is added under the **Interfaces** node.
3. Under the **Interfaces** node, select the new interface node, and set **Type** to **Fan Interface**.

   The appropriate fan nodes are added to the **Physics Conditions** and **Physics Values** nodes under the Fan Interface node.

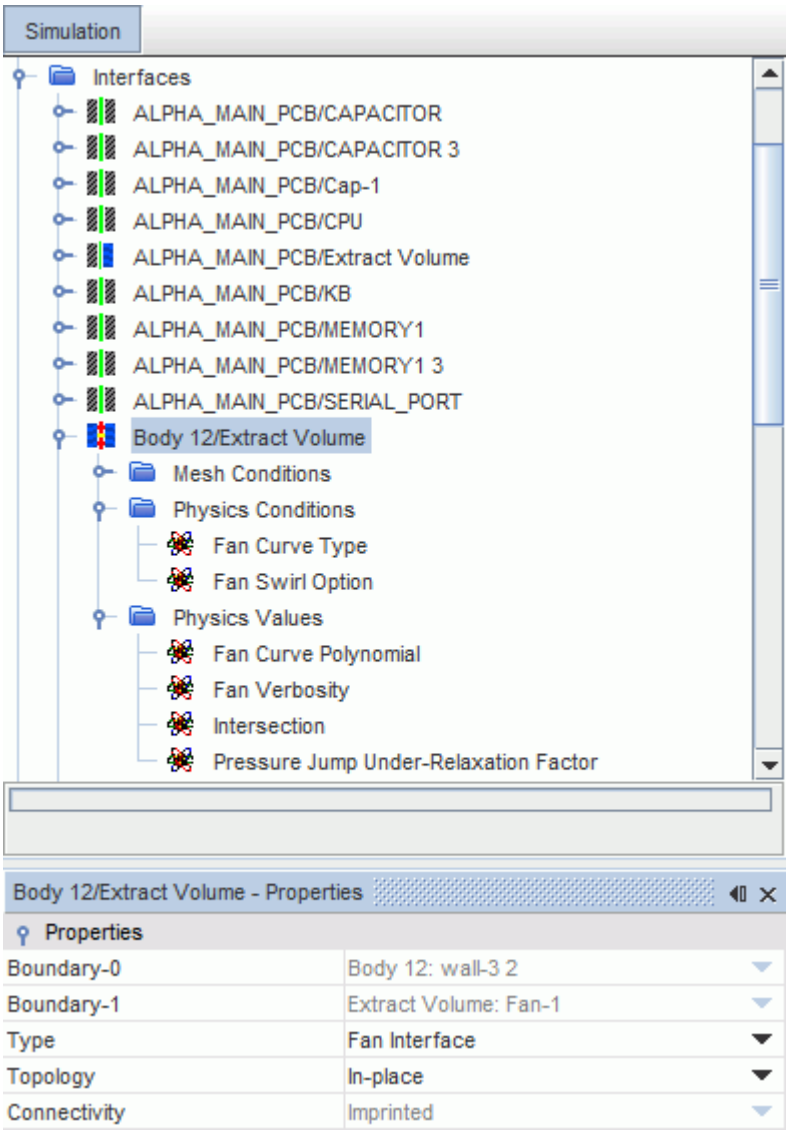
   The Pressure Jump Under-Relaxation Factor node is also added under the **Physics Values** node.

See Also:

Orientation of Parts-Based Interfaces

Fan Interface Properties

The **Fan Interface** node has properties, conditions, values, and a pop-up menu.



Fan Interface Properties

Boundary-0	Specifies the upstream boundary of the fan interface. (Read Only)
Boundary-1	Specifies the downstream boundary of the fan interface. (Read Only)
Topology	Defines the connection type between the interfaces.

	In-place	Uses the in-place topology.
	Periodic	Uses the periodic topology.
	Repeating	Uses the repeating topology.
Type	Specifies the type of interface: **Fan Interface**.	
Connectivity	Shows how the boundaries on two sides of the interface are connected. Only **Imprinted** connectivity is possible.	

Specifying the Fan Performance Curve Type

The fan performance curve specifies the pressure rise across the fan interface.

By default, you specify the fan performance curve as a table. Alternatively you can specify the fan performance curve as a polynomial function.

To specify the fan performance curve type:

1. Expand the **Interfaces** > **[Interface Name]** > **Physics Conditions** node, and click **Fan Curve Type**.
2. In the *Properties* window, set the appropriate **Curve Type**.

Table	The fan performance curve is supplied as a table. See Specifying the Fan Performance Curve as a Table.
Polynomial	The fan performance curve is supplied as a polynomial function. See Specifying the Fan Performance Curve as a Polynomial Function.

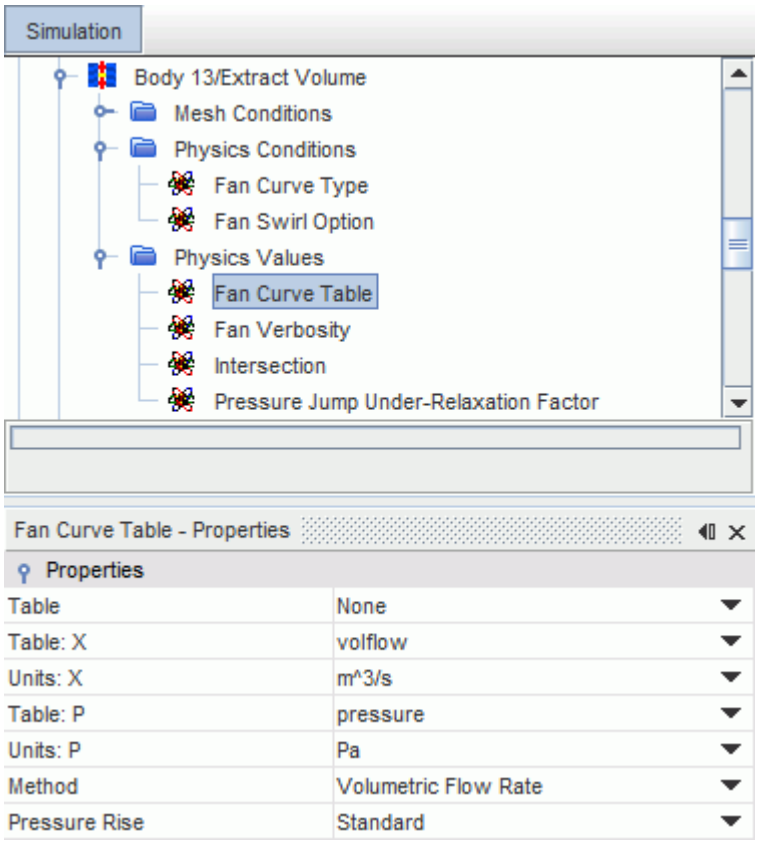
For a linear fan curve, both the table and the polynomial are equivalent if the parameters are set correctly.

Specifying the Fan Performance Curve as a Table

When you set up a fan interface, you can specify the fan performance curve as a table.

Use the following steps:

1. Import the appropriate fan performance curve table into the simulation.
2. Expand the **Interfaces** > **[Interface Name]** > **Physics Values** node, and select **Fan Curve Table**.



3. In the *Properties* window, set the following properties:

Table	Select the fan performance curve table that you imported into your simulation.
Table: X	Select the column that contains the input variable of the pressure rise function for the selected fan performance table.
Units: X	Select the appropriate input variable units.
Table: P	Select the column that contains the pressure data for the selected fan performance table.
Units: P	Select the appropriate pressure units.
Method	Select the input variable of the pressure rise function.

	When you choose the **Static to Static** option, ensure that the specified fan curve is a decreasing function of the input variable. When you choose the **Standard** option, ensure that [specified fan curve + $\frac{1}{2}pv^2$] is a decreasing function of input variable. Out of the four fan curve method options, only the **Mass Average Velocity** option is not linearized and is considered to be less robust. The other options apply a different pressure rise to each face at the fan interface. However, for some problems, it can be more meaningful to apply the same pressure rise to all of the faces on the fan interface. In that scenario, you can use the mass averaged velocity options.	
	Local Velocity	Sets the curve as a function of local velocity. This option is based on the velocity normal to the face. This option is the default.
	Mass Average Velocity	Sets the curve as a function of mass average velocity.
	Volumetric Flow Rate	Sets the curve as a function of volumetric flow rate.
	Mass Flow Rate	Sets the curve as a function of mass flow rate.
Pressure Rise	Specifies how the pressure rise across the fan interface is defined. See Modeling Fan Performance Curves.	
	Standard	The difference between the static pressure downstream and the total pressure upstream of the interface. This is the default option.
	Static to Static	The difference between the static pressure downstream and the static pressure upstream of the interface. This definition is the same as the difference between the total pressure downstream and the total pressure upstream of the interface.

The fan curve table computes the pressure rise by linear interpolation between the data points in the table. Outside the intervals, the nearest limit is used.

You can scale the fan curve in temperature as well as in rotation rate. See Specifying the Fan Curve Temperature Scaling.

Note: For Pressure Outlet and Stagnation Inlet boundaries, the fan does not add swirl velocity, and the operating and data rotation rates are only used for scaling the performance curve.

Specifying the Fan Performance Curve as a Polynomial Function

When you set up a fan interface, you can specify the fan performance curve as a polynomial function.

The fan curve polynomial computes the pressure rise as:

- Given N intervals $[x_0, x_1], [x_1, x_2], \ldots, [x_{N-1}, x_N]$ the polynomial in the ith interval is $y_i(x_i)$, evaluated using the coefficients for that interval.
- Outside the intervals, the nearest limit is used:

$$y(x) = \begin{cases} y(x_0) & x \le x_0 \\ y(x_N) & x > x_N \end{cases}$$

To specify the fan performance curve polynomial function:

1. Expand the **Interfaces** > **[Interface Name]** > **Physics Values** node, and then click **Fan Curve Polynomial**.
2. In the *Properties* window, set the appropriate values.

Polynomial	Specifies the fan performance curve polynomial function. See Specifying the Fan Performance Curve Polynomial.
Method	Specifies the input variable of the pressure rise function. For a full description of this property, see Specifying the Fan Performance Curve as a Table.
Pressure Rise	Specifies how the pressure rise across the fan interface is defined. For a full description of this property, see Specifying the Fan Performance Curve as a Table. For more information on fan performance curves, see Modeling Fan Performance Curves.
Operating Rotation Rate	Specifies the rotation rate used by the simulation. Swirl velocity is not added at the interface unless the Fan Swirl Option is activated. If this value is equal to the value of Data Rotation Rate, no scaling is done. See Scaling the Fan Performance Curve.

You can scale the fan curve in temperature as well as in rotation rate. See Specifying the Fan Curve Temperature Scaling.

Note: For Pressure Outlet and Stagnation Inlet boundaries, the fan does not add swirl velocity, and the operating and data rotation rates are only used for scaling the performance curve.

See Also:

Modeling the Fan Swirl

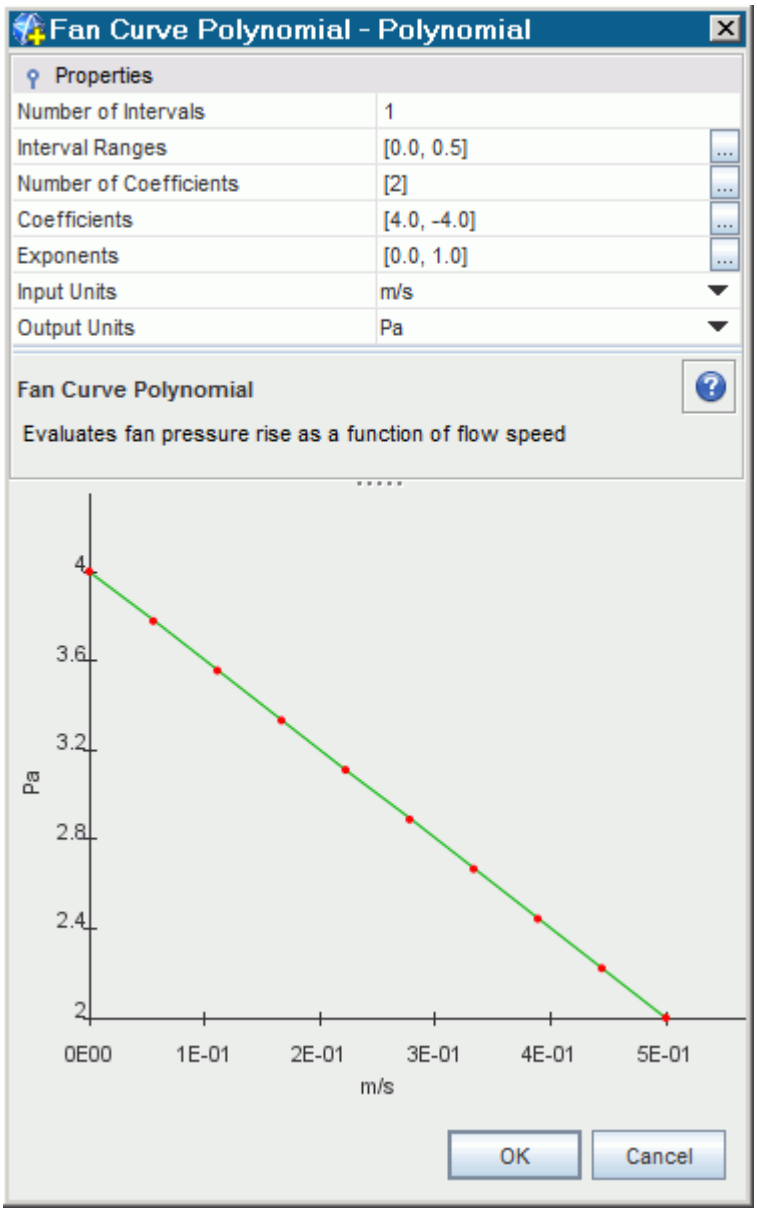
Pressure Outlet

Stagnation Inlet

Specifying the Fan Performance Curve Polynomial

The fan performance curve can be specified as a polynomial function.

The *Fan Curve Polynomial - Polynomial* dialog lets you specify the fan performance curve polynomial function.



In the *Properties* panel, set the properties to specify the polynomial function that represents the fan performance curve.

Number of Intervals	Set the number of intervals that define the fan performance curve. You can specify different equations for each interval of the fan curve.

Interval Ranges	Specify the minimum and maximum values of each interval. For example, to specify three intervals [A, B], [B, C], [C, D], enter `[A, B, C, D]`.
Number of Coefficients	Specify the number of coefficients per interval. For example, to specify the coefficients for the intervals [A, B], [B, C], [C, D], enter `[a, b, c]`. In this example, `a` is the number of coefficients for interval [A, B], `b` is the number for interval [B, C], and `c` is the number for interval [C, D].
Coefficients	Specify the coefficients for each of the terms in the polynomial equation. Enter the values for each interval, in order. For example, if a=3, b=2, and c=4, enter the values in the format: `[a1, a2, a3, b1, b2, c1, c2, c3, c4]`.
Exponents	Specify the exponents for each of the terms in the polynomial equation. Enter the values for each interval in order, in the same format as for the coefficients.
Input Units	Specify the units for the input variable (the X-axis) of the polynomial equation. The input variable is the flow rate that you set in the *Method* property of the **Fan Curve Polynomial**.
Output Units	Specify the units for the output variable (the Y-axis) of the polynomial equation. The output variable is the pressure drop across the fan interface.

Specifying the Fan Curve Temperature Scaling

You can modify the fan performance curve in response to temperature.

You specify the fan curve temperature scaling properties under the **Physics Values** > **Fan Curve Temperature Scaling** node.

See Scaling the Fan Performance Curve.

Fan Curve Temperature Scaling	Adjusts the fan performance curve according to the fluid temperature used in the simulation.	
	Activated	Scale in response to temperature.
	Deactivated	
Data Temperature	Specifies the temperature at which the fan curve table or polynomial was measured. This value is not used unless **Fan Curve Temperature Scaling** is activated.	

Modeling the Fan Swirl

You can model the fan swirl in your simulation.

You can use either the **Calculated** option, based on Eqn. (4062), or the **User-Specified** option to calculate the swirl downstream of the fan. By default, fan swirl is not included.

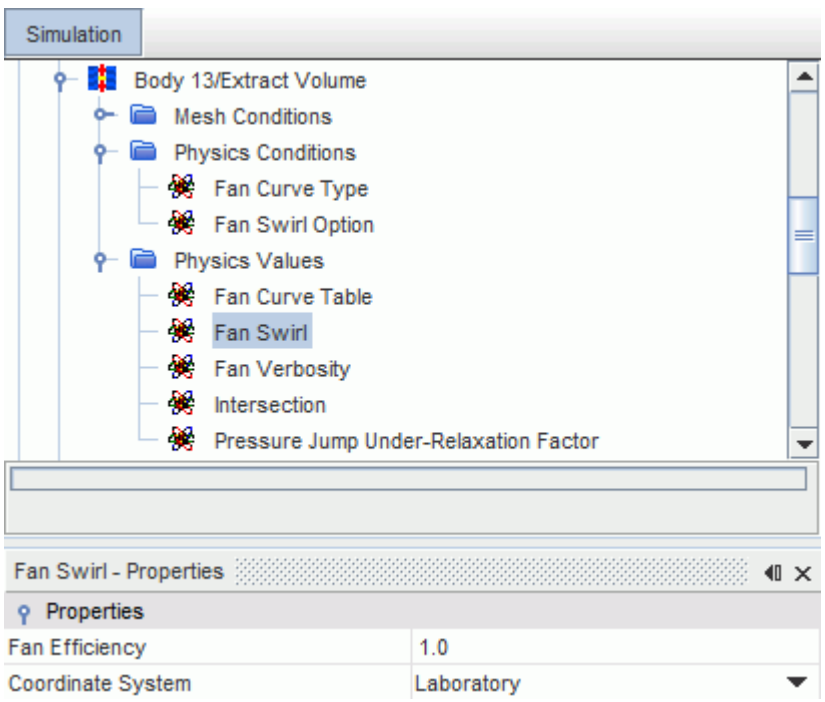
For more information, refer to the Fan Interface formulation.

To include the fan swirl in your simulation:

1. Expand the **Interfaces** > **[Interface Name]** > **Physics Conditions** node, and click **Fan Swirl Option**.
2. In the *Properties* window, select **Calculated** or **User-Specified** from the **Fan Swirl Option** drop-down menu.

   If you select **Calculated**, the **Fan Swirl** node is added to the **Physics Values**.

   If you select **User-Specified**, the **Specified Fan Swirl** node is added to the **Physics Values**.



3. In the *Properties* window, set the appropriate values.

Fan Swirl Properties

Fan Efficiency	Specify the fan efficiency, η in Eqn. (4059).
Coordinate System	Select the appropriate coordinate system. Types other than Laboratory are not available for selection until you create them.

	You can use any of the coordinate system types for fan modeling, but the following restrictions apply: • The origin of the coordinate axes must coincide with the center of rotation of the fan. • The z-axis must run along the fan axis and be oriented in the fan-flow direction.

Specified Fan Swirl Properties

Method	Selects the input for the fan swirl value as a vector profile.
Dimensions	The units of the fan swirl value (read-only).
Coordinate System	Select the appropriate coordinate system. Types other than Laboratory are not available for selection until you create them. You can use any of the coordinate system types for fan modeling, but the following restrictions apply: • The origin of the coordinate axes must coincide with the center of rotation of the fan. • The z-axis must run along the fan axis and be oriented in the fan-flow direction.

See Also:

Specifying the Fan Performance Curve as a Polynomial Function

Orientation of Parts-Based Interfaces

In cases where the correct orientation of an interface requires confirmation, STAR-CCM+ adds an **Orientation** node to parts-based interfaces. Use the right-click menu on this node to access the action for reversing the orientation. This confirmation is particularly required for interface types such as fan, blower, and fully-developed.

- The Orientation has two modes: Automatic and Manual.
  - Automatic:
    - Set direction for contacts based on regions associated with the part-surface pairs
    - Set direction for contacts based on parent boundary associated with the part-surface pairs
    - Set direction for contacts based on parts associated with the part-surface pairs
    - If mode is set to automatic, you can use the **Orient** command on the context menu to re-execute the ordering based on these rules.
  - Manual:
    - User can select contacts and reverse their direction
    - If the mode is set to Manual, you can use the Reverse Contacts... command on the context menu of the Orientation object to reverse selected contacts.
      - Invoking the command will pop-up the Reverse Contacts contact selector.
      - You can pick contacts to Reverse
      - The icon changes (small blue arrow) to indicate the "direction" of the orientation.
- Certain interfaces may have conditions that require consistent orientation, e.g. fan.

See Also:

Creating a Blower Interface

Creating a Fan Interface

Fully Developed Interface

Setting the Fan Interface Verbosity

When fan verbosity is activated, a report is printed to the *Output* window for every iteration. The report includes the calculated operating point of the fan: the pressure rise and flow rate value pair that lies on the fan curve at that iteration.

To activate or deactivate the fan verbosity:

1. Expand the **Interfaces** > **[Interface Name]** > **Physics Values** node, and then click **Fan Verbosity**.
2. In the *Properties* window, set the **Verbosity** property to the appropriate value: **On** or **Off**.

Adjusting the Fan Interface Intersection Tolerance

The intersection tolerance applies to all direct interfaces. The tolerance is the fraction of the length of the smallest edge that is attached to the vertex.

The default tolerance is 0.05. If the intersection is not 100% successful, you can increase the tolerance value to allow the unmatched faces to be matched.

To adjust the fan interface intersection tolerance:

1. Expand the **Interfaces** > **[Interface Name]** > **Physics Values** node, and then click **Intersection**.
2. In the *Properties* window, set the **Tolerance** property to the appropriate value.

Creating a Blower Interface

The blower interface in STAR-CCM+ allows you to model two types of radial fans: centrifugal fans and impeller fans.

Each interface has a direction from side 0 to side 1. When you create the interface, select side 0 first and then select side 1. For a blower interface, side 0 is upstream to side 1.

To create a blower interface:

1. Expand the **Regions** > **[Fluid Name]** > **Boundaries** node, select the upstream boundary (side 0), and then hold down the <Ctrl> key and select the downstream boundary (side 1).
2. Right-click either of the selected boundaries and select **Create Interface**.

   The new interface node is added under the **Interfaces** node.
3. Under the **Interfaces** node, select the new interface node, and set **Type** to **Blower Interface**.
4. Specify the appropriate values in the **Physics Conditions** and **Physics Values** nodes under the Blower Interface node.

See Also:

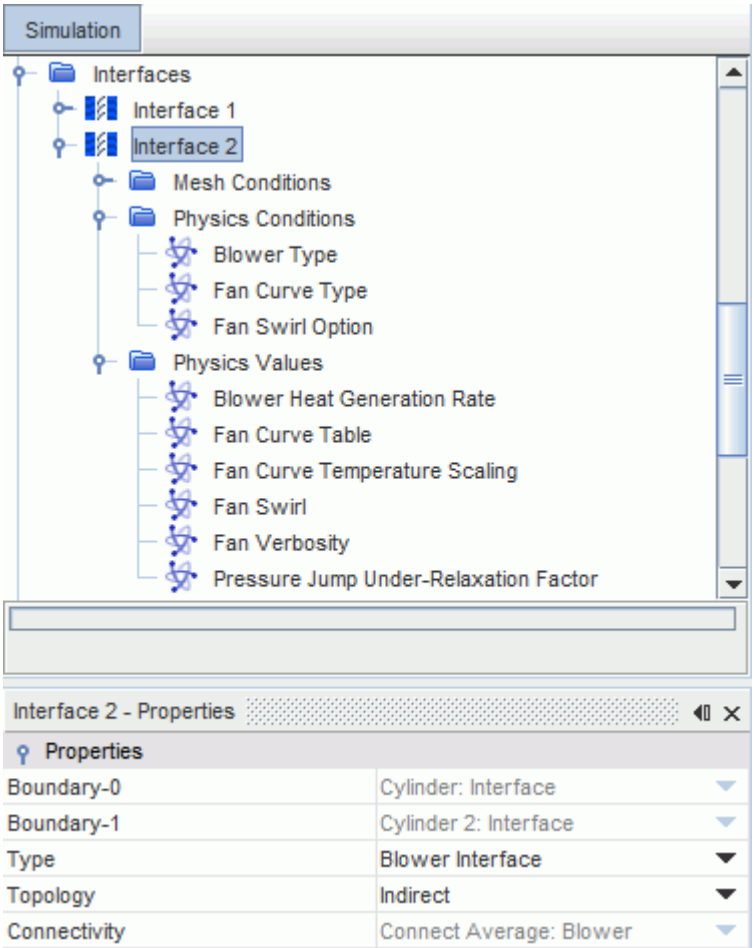
Orientation of Parts-Based Interfaces

Blower Interface Properties

Blower Interface Properties

This section describes the properties and physics values that you can specify for a blower interface.

The Blower Interface node has properties, conditions, values, and a pop-up menu.



Blower Interface Properties

The properties of a **Blower Interface** node are listed below. These properties cannot be modified.

Boundary-0	Specifies the upstream boundary of the blower interface (Read Only)
Boundary-1	Specifies the downstream boundary of the blower interface (Read Only)
Type	Specifies the type of interface: **Blower Interface**.
Topology	Defines the connection type between the interfaces. Only the **Indirect** type is possible.
Connectivity Option	Shows how the boundaries on two sides of the interface are connected. Only **Connect Average** connectivity is possible.

Blower Interface Conditions

You can specify the following physics conditions for a Blower interface:

Blower Type	Specifies the blower type according to the fan geometry.	
	Centrifugal	You set the fan curve type and specify the fan performance curve.
	Impeller	You set the fan curve type and specify the fan performance curve, and can optionally specify the fan swirl.
Fan Curve Type	The fan performance curve specifies the pressure rise across the blower interface.	
	Polynomial	The fan curve is defined as a polynomial function.
	Table	The fan curve is defined as a table of pressure versus flow rate values.
Fan Swirl Option	The option to add swirl to the flow is available only for blowers of the Impeller type.	
	Calculated	The Fan Swirl node is added to the Physics Values, where you specify the appropriate coordinate system and blade angle. The swirl is calculated accordingly and added to the flow.
	No Swirl	No swirl is added to the flow.

Blower Interface Values

You can specify the following physics values for a Blower interface:

Blower Heat Generation Rate	Specifies the heat that is generated by the blower per unit time. This heat is added to the flow that exits the blower. This option is available only when an energy model is activated in the physics continuum.
Fan Curve Table/Polynomial	Specifies the fan curve performance using a table or a polynomial, according to the Fan Curve Type that is specified in the physics conditions. See Specifying the Fan Performance Curve as a Polynomial Function. See Specifying the Fan Performance Curve as a Table.
Fan Curve Temperature Scaling	Scale the fan performance curve using data temperature. This option is available only when an energy model is activated in the physics continuum.

	See Specifying the Fan Curve Temperature Scaling.
Fan Swirl	Specifies the coordinate system and the blade angle that are used to compute the flow swirl for the impeller fan. See Blower Swirl Values.
Fan Verbosity	Activate or deactivate the reporting of fan operating point details in the Output window. See Setting the Fan Interface Verbosity.
Pressure Jump Under-Relaxation Factor	Specifies an under-relaxation factor between 0 and 1 to improve convergence. See Pressure Jump Under-Relaxation Factor.

Blower Swirl Values

You can specify the following values:

Coordinate System	The coordinate system that is used to specify the impeller orientation and its rotation direction. Types other than Laboratory are not available for selection until you create them. You can use any of the coordinate system types for impeller fan modeling, but the following restrictions apply: • The origin of the coordinate axes must coincide with the center of rotation of the impeller fan. • The z-axis must run along the axis of the impeller fan and be directed such that the angular velocity of the fan is along the axis.
Blade Angle	The blade angle (β) of the impeller fan: • β = 90 means radial blade. • β > 90 means forward-swept blade. • β < 90 means backward-swept blade.

See Also:

Creating a Blower Interface

Modeling the Blower Swirl

Modeling the Blower Swirl

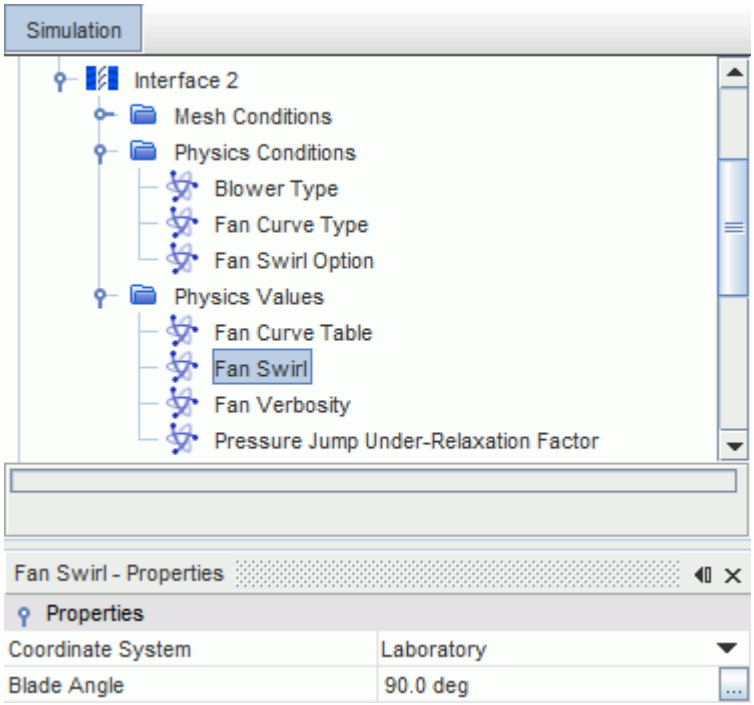
You can model the blower swirl in your simulation.

The option to add swirl to the flow is available only for blowers of the Impeller type.

To specify the blower swirl:

1. Expand the **Interfaces** > **[Interface Name]** > **Physics Conditions** node, and click **Fan Swirl Option**.
2. In the *Properties* window, set **Fan Swirl Option** to **Calculated**.

   The **Fan Swirl** node is added to the **Physics Values**.



3. In the *Properties* window, set the appropriate values.

See Also:

Blower Interface Properties

Modeling Multiple Fans and Blowers in a Single Interface

With per-contact values you can model any number of fans or blowers, with different fan curves, in a single parts-based interface.

Although it is possible to specify per-contact values for fan curve polynomials and fan curve tables, there are other parameters associated with fans/blowers that currently do not allow per-contact values (fan swirl, fan curve temperature scaling, and blower heat generation rate).

Since fan swirl does not allow parts-based specification, interfaces that use per-contact values for fan curve polynomials and fan curve tables cannot have swirl enabled — doing so leads to incorrect results. Fan swirl allows the specification of a coordinate system to identify the fan-axis. If there are multiple contacts in an interface, each representing an individual fan, each fan will have a different axis. In the absence of parts-based support for fan swirl, the same coordinate system will be use for all fans leading to wrong computation of swirl.

To model multiple fan curves under a single interface:

1. Activate per-contact values:
   a. Select the parts-based fan interface and set *Allow Per-Contact Values* to **True**.

      A **Contact Subgrouping** node is added to the fan interface.
2. Create subgroups for each individual fan curve that you want to model:
   a. Expand the **Contact Subgroupings** > **Subgrouping 1** node.

      By default, all part contacts are placed in the **Subgroup 1** subgroup.
   b. To create a new contact subgroup, right-click **Subgrouping 1** and select **New**.

      It is good practise to give each subgroup an appropriate name.
   c. Select the subgroup node and click ... next to *Objects*.
   d. In the *Objects* dialog, select the part contact that corresponds to the fan curve you want to model.
   e. Repeat these steps for each fan curve.
3. If required, create new contact subgroupings to help manage subgroups:
   a. Right-click the **Contact Subgroupings** node and select **New**.
4. Set up the fan curves:
   a. Select the **[Fan_Interface]** > **Physics Values** > **Fan Curve Polynomial / Fan Curve Table** node and set *Specify by Part Subgroup* to **True**.

      The **By Contact Subgroup** node appears under the **Fan Curve Polynomial / Fan Curve Table** node. This node contains all the subgroups.

      By default, the values from the interface **Fan Curve Polynomial / Fan Curve Table** node are copied to the **Fan Curve Polynomial / Fan Curve Table** subgroup nodes. If you want a different fan curve for any subgroup, you can expand and modify the corresponding **Fan Curve Polynomial / Fan Curve Table** subgroup node. See Specifying the Fan Performance Curve Polynomial.
   b. Repeat these steps for the remaining fan curves.

      If you want to model multiple fans that use the same fan curve, you are advised to create separate subgroups for each.

Using the Fan Momentum Source

The fan momentum source applies the effect of a rotating fan to the fluid cells within a region that you select. Using a fan momentum source implies that you already have a separate region that is located at the position of the fan.

For more information on the fan momentum source model formulation, see the Fan Momentum Source formulation.

To implement a fan momentum source in your simulation, complete the following tasks:

1. Ensure that you have selected a flow model.
2. Obtain the appropriate fan performance curve and import the data table into the simulation.

   For more information, see Importing the Fan Performance Data Table.
3. In the appropriate region, activate the fan momentum source.

   For more information, see Activating the Fan Momentum Source.
4. Configure the fan momentum source by setting the appropriate properties:
   a) Select the fan curve table that you want to use, and specify the **Voldot** and **Pressure** columns.
   b) Set the data rotation rate and the data temperature.

      For more information, see Specifying the Fan Performance Curve.
   c) Specify the upstream and downstream boundaries of the fan momentum source.
   d) Specify the operating rotation rate and the blade angle of the fan.

      For more information, see Fan Momentum Source Properties.
5. Set the appropriate fan solver properties.

   For more information, see Working with the Fan Solver.

To plot and monitor the flow rate through the fan, use the Surface Integral report with the velocity component normal to the fan face.

Importing the Fan Performance Data Table

STAR-CCM+ calibrates the fan momentum source in your simulation with performance data from a real fan.

To provide this performance data:

- Import the appropriate fan curve table into the simulation.

  Shown below is a simplified example of a fan performance data table. The two columns are volumetric flow rate, `Vol`, in cubic meters per second (m^3/s), and pressure rise, `Press`, in Pascals (pa):

Vol	Press
0.01	458.58
0.20	286.26
0.40	267.68
0.60	239.05
0.80	220.56
1.00	190.52
1.20	146.84
1.40	98.86
1.60	46.45
1.80	-27.10
2.00	-111.15
2.20	-204.01

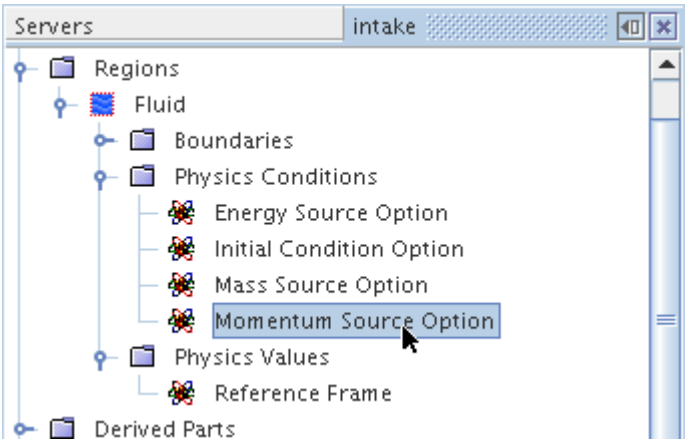
  Ensure that the pressure decreases with increasing flow rate across the fan for all data points in the table. If you use a fan curve that does not meet this requirement, the results are unstable.

Activating the Fan Momentum Source

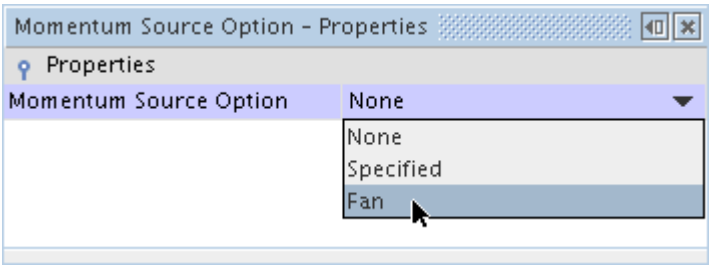
You activate the fan momentum source in the appropriate fluid region.

To activate the fan momentum source:

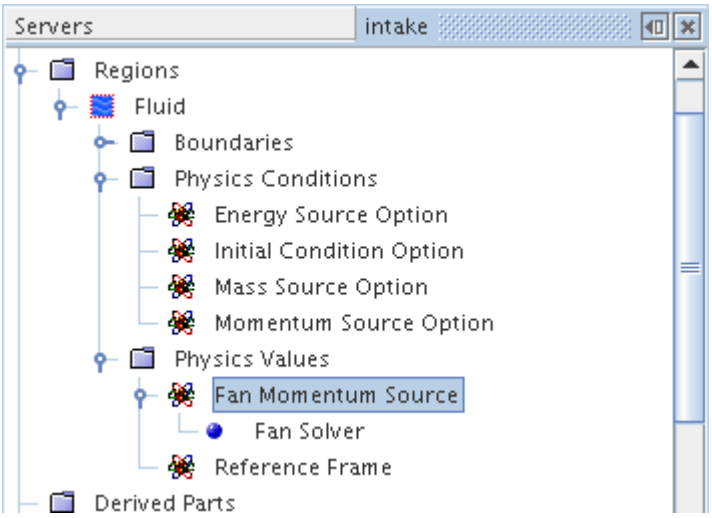
1. Expand the **Regions** > **[Region]** > **Boundaries** > **Physics Conditions** node and select **Momentum Source Option**.



2. In the *Properties* window, set *Momentum Source Option* to **Fan**.



The **Fan Momentum Source** node, and its child, the **Fan Solver** node, become accessible.



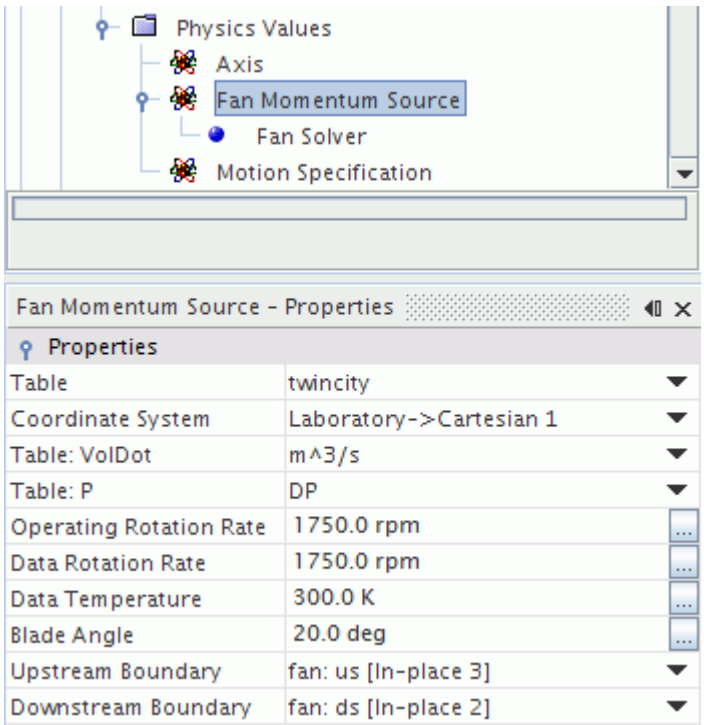
The **Fan Momentum Source** node represents fan properties and device data for an axial fan. It is one of the physics values of a region.

Specifying the Fan Performance Curve

After you have imported the fan performance curve table, set the appropriate **Fan Momentum Source** properties.

Set the **Fan Momentum Source** properties to specify the data columns to use in your simulation:

1. Expand the **Regions** > **[Region]** > **Boundaries** > **Physics Values** node and select **Fan Momentum Source**.



2. In the *Properties* window, set the following **Fan Momentum Source** properties:

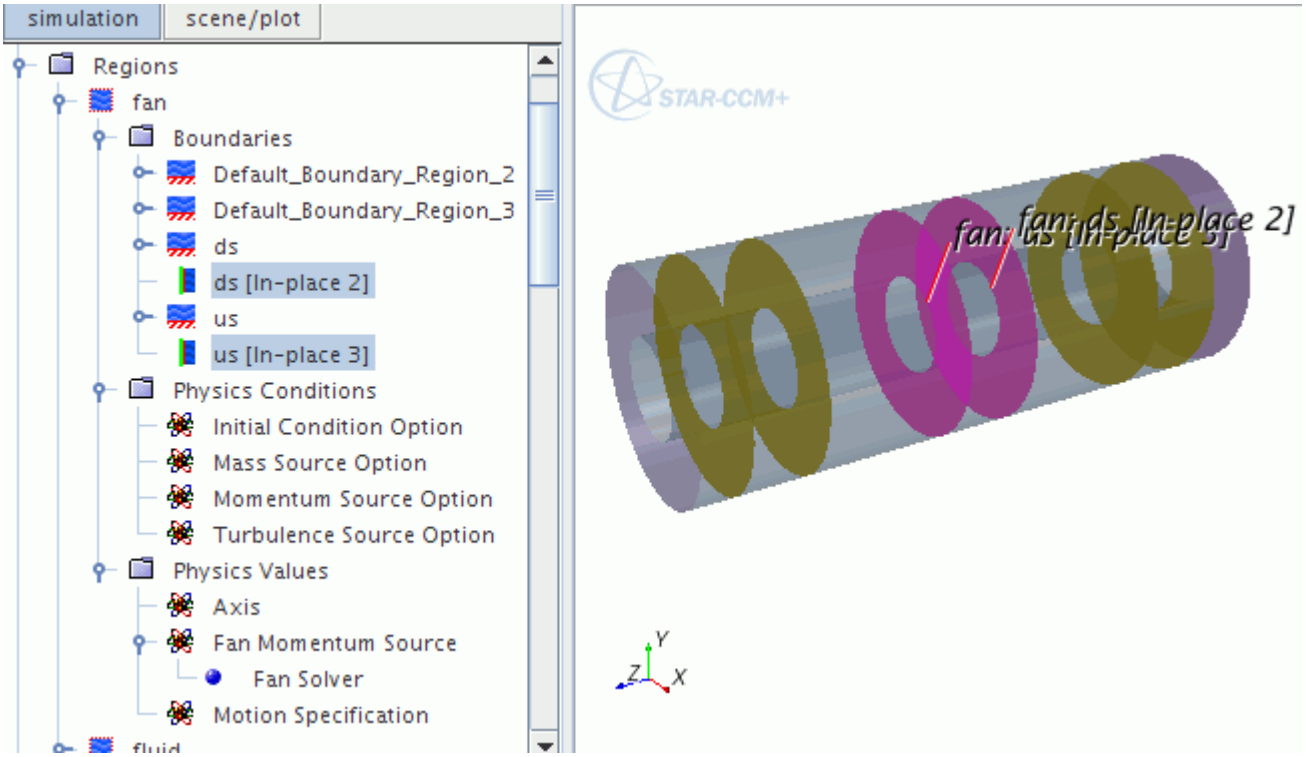
Table	Select the fan performance curve table that you imported into your simulation.
Table: VolDot	Select the column that contains the volumetric flow data for the selected fan performance table.
Table: P	Select the column that contains the pressure data for the selected fan performance table.
Data Rotation Rate	Specify the fan rotation rate at which the fan performance curve was measured. The available units are radians per second, revolutions per second, and revolutions per minute. If you want to use a different rotation rate in your simulation, you specify the appropriate rate in the Fan Momentum Source Operating

	Rotation Rate property. STAR-CCM+ automatically adjusts the fan curve data to the new rate by using classical fan laws.
Data Temperature	Specify the temperature at which the fan performance curve was measured. The temperature is taken as the mass flow average of the temperature coming into the fan. If you use a different temperature in your simulation, STAR-CCM+ automatically adjusts the fan curve data to the new temperature by using classical fan laws.

Fan Momentum Source Properties

You specify the upstream and downstream boundaries of the fan momentum source, and set the operating rotation rate and the blade angle of the fan.

The example below shows the upstream and downstream boundaries of the fan region. These boundaries must belong to interfaces where the fan region joins the fluid region.



If your simulation uses an operating rotation rate which is different to the rate at which the fan performance curve was measured, STAR-CCM+ automatically adjusts the fan curve data to the new rate. For more information, see Scaling the Fan Performance Curve.

You set the fan momentum source properties on the **Regions** > **[Region]** > **Boundaries** > **Physics Values** > **Fan Momentum Source** node.

Fan Momentum Source Properties

Table	Select the fan performance curve table that you imported into your simulation.
Table: VolDot	Select the column that contains the volumetric flow data for the selected fan performance table.
Table: P	Select the column that contains the pressure data for the selected fan performance table.

Operating Rotation Rate	By default, the fan speed for the simulation is the *Data Rotation Rate*, the fan rotation rate at which the fan performance curve was measured. If you want to use a different rotation rate in your simulation, specify the appropriate rate here. The available units are radians per second, revolutions per second, and revolutions per minute. STAR-CCM+ automatically adjusts the fan curve data to the new rate by using classical fan laws.
Data Rotation Rate	Specify the fan rotation rate at which the fan performance curve was measured. The available units are radians per second, revolutions per second, and revolutions per minute. If you want to use a different rotation rate in your simulation, you specify the appropriate rate in the Fan Momentum Source *Operating Rotation Rate* property. STAR-CCM+ automatically adjusts the fan curve data to the new rate by using classical fan laws.
Data Temperature	Specify the temperature at which the fan performance curve was measured. The temperature is taken as the mass flow average of the temperature coming into the fan. If you use a different temperature in your simulation, STAR-CCM+ automatically adjusts the fan curve data to the new temperature by using classical fan laws.
Blade Angle	Specify the angle that the fan blade makes with respect to the flow. The available units are degrees, radians, and revolutions.
Coordinate System	Select the appropriate coordinate system. Types other than Laboratory are not available for selection until you create them. You can use any of the coordinate system types for fan modeling, but the following restrictions apply: • The origin of the coordinate axes must coincide with the center of rotation of the fan. • The z-axis must run along the fan axis and be oriented in the fan-flow direction.
Upstream Boundary	Select the boundary that defines the upstream side of the fan. Select a boundary of the type in-place interface. If none appears in the object selector, create an in-place interface.
Downstream Boundary	Select the boundary that defines the downstream side of the fan. Select a boundary of the type in-place interface. If none appears in the object selector, create an in-place interface.

Scaling the Fan Performance Curve

A fan performance curve is for a specific fan rotation rate and fluid temperature. If your simulation uses a different fan rotation rate or a different fluid temperature from the fan performance curve that you specified in your simulation, the fan performance curve must be modified to suit those conditions.

STAR-CCM+ uses established fan laws to scale fan performance curves from one set of conditions to another, as per Eqn. (374) and Eqn. (375).

Fan Laws

$$\Delta P_2 = \Delta P_1 \times \left(\omega_2/\omega_1\right)^2 \times \left(T_1/T_2\right) \tag{374}$$

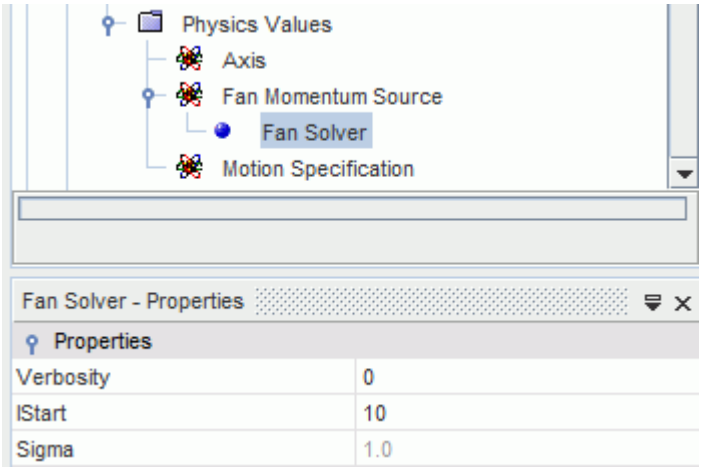
$$\psi_2 = \psi_1\left(\omega_2/\omega_1\right) \tag{375}$$

where ω is the fan speed (rpm), T is the fluid temperature (K), and ψ is the volumetric flow rate.

Working with the Fan Solver

The **Fan Solver** node contains properties that control how the fan momentum source is solved. You can specify the fan solver property values to suit your simulation.

You set the fan solver properties on the **Regions** > **[Region]** > **Boundaries** > **Physics Values** > **Fan Momentum Source** > **Fan Solver** node.



Fan Solver Properties

Verbosity	This setting controls the printing of diagnostic data to the *Output* window. The settings are:	
	0	Prints no information about the fan model.
	1	Prints deviation of the swirl velocity from the target, the pressure rise target for the current non-reversed volume flow rate through the fan, and the current pressure rise.
IStart	Specify the iteration number at which to start solving for the fan simulation. This setting lets you delay the start of the fan simulation while your simulation solves for other components. The default is 10, as the solution can be unstable for the first few iterations.	
Sigma	Specify the factor that is used to match the target for the swirl. This value is σ in Eqn. (4068).	

Virtual Disk Model

The virtual disk model is based upon the principle of representing propellers, turbines, rotors, fans, and so on, as an actuator disk. The actuator disk treatment is practical when information about the rotor/propeller behavior is available and its effect on the surrounding is desired.

The action of the actuator disk on the flow field enters the momentum equations in the form of a source term that is distributed over the virtual disk. Various distributions of different fidelity are possible to model the action of the actuator depending on the application area.

For the virtual disk model, there are currently four methods implemented:

- Body Force Propeller Method
- Blade Element Method
- 1D Momentum Method
- User Defined Method

Contents:

Selecting the Virtual Disk Model
Body Force Propeller Method
Blade Element Method
1D Momentum Method
User Defined Method
Reporting Virtual Disk Results
Virtual Disk Properties
Inflow Specification Properties Reference
Virtual Disk Reports Properties
Virtual Disk Field Functions

Selecting the Virtual Disk Model

Activate the virtual disk model as an optional physical model from the *Physics Model Selection* dialog of the relevant physics continuum.

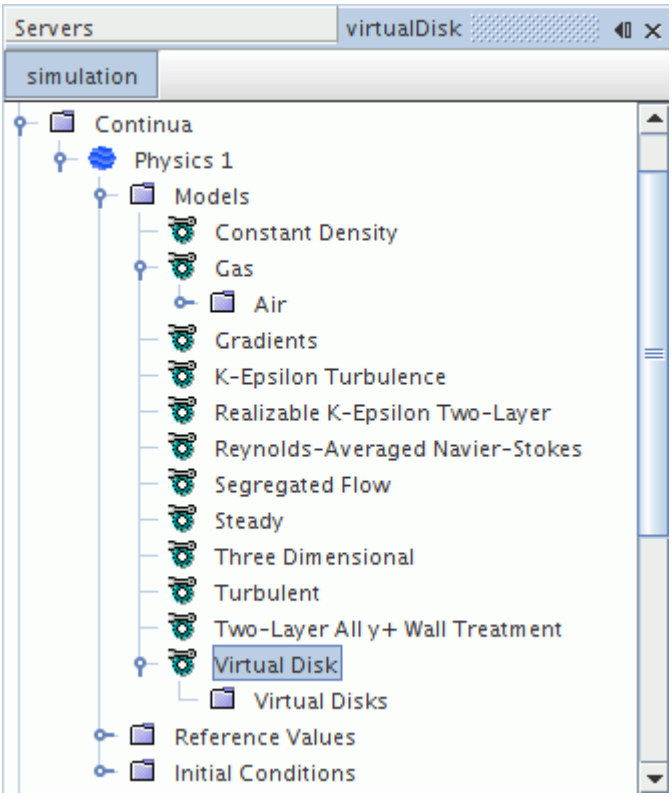
To select the virtual disk model:

1. In a physics continuum, activate models that are compatible with the virtual disk model.

Group Box	Model
Space	**Three Dimensional**
Time	Any, except **Harmonic Balance**
Material	All virtual disk model methods are compatible with any material model, except **Solid** and **Multi-Component Solid**.
Flow	Any, except **Viscous Flow**.
Equation of State	Any
Viscous Regime	Any

2. From the *Optional Models* group box, select **Virtual Disk**.
3. Click Close.

   The **Virtual Disk** node appears under the **Models** node in the simulation tree. It contains the **Virtual Disks** sub node that you use to actually create the virtual disk.


Servers
virtualDisk
simulation
Continua
Physics 1
Models
Constant Density
Gas
Air
Gradients
K-Epsilon Turbulence
Realizable K-Epsilon Two-Layer
Reynolds-Averaged Navier-Stokes
Segregated Flow
Steady
Three Dimensional
Turbulent
Two-Layer All y+ Wall Treatment
Virtual Disk
Virtual Disks
Reference Values
Initial Conditions

Body Force Propeller Method

As part of the virtual disk model, the body force propeller method is meant to model the effects of a marine propeller.

The method models the flow field interaction of the hull of a ship and the propeller. The flow that is induced by the propeller depends on the flow around the ship hull. Similarly, the hull flow is influenced by the propeller.

The body force propeller method can be used to model the propeller propulsion of a ship as part of a DFBI (Dynamic Fluid Body Interaction) simulation. It can be a clear advantage to use this method regarding mesh size and so reduced computational cost over performing a simulation employing the fully resolved geometry of the propeller. It is useful if you are not interested in analyzing the flow field around the propeller but more importantly need the correct propulsion specification.

The body force propeller method uses the following approach:

- Definition of a virtual disk regarding position and direction in which thrust is produced.
- Specification of a propeller performance curve.
- Specification of an operating point, for example, the rotational speed.
- Specification of an inflow velocity plane that yields an average velocity and density for propeller inflow.

As a result, the radially varying distribution of the axial and tangential forces of the modeled propeller acting on the flow is calculated. The integration of these forces over the disk gives the thrust and torque of the propeller, which are then available for coupling with a DFBI simulation.

Contents:

Setting up and using the Body Force Propeller Method
Observing Virtual Disk Meshing Requirements for the Body Force Propeller Method
Creating a Virtual Disk for the Body Force Propeller Method
Setting the Propeller Performance Curve for the Body Force Propeller Method
Specifying the Disk Geometry for the Body Force Propeller and the 1D Momentum Methods
Specifying the Disk Orientation
Specifying the Radial Thrust and Torque Distribution
Radial Thrust and Torque Distribution Properties Reference
Specifying the Inflow Velocity Plane
Specifying the Propeller Handedness for the Body Force Propeller Method
Propeller Handedness Option Properties
Propeller Handedness
Specifying the Operation Point for the Body Force Propeller Method

Setting Up and Using the Body Force Propeller Method

The setup of the virtual disk of the body force propeller method does not require the creation of a separate region to which the model is then applied. You create the shape of the virtual disk by specifying the relevant

parameters such as radius, thickness, and orientation to an existing mesh. To verify the dimensions and orientation of the virtual disk, you can visualize it by using virtual disk specific field functions.

To set up and run the body force propeller method:

1. Generate a mesh that is suitable for use with the virtual disk model.

   See Observing_Virtual_Disk_Meshing_Requirements_for_the_Body_Force_Propeller_Method
2. Select the virtual disk model.

   See Selecting the Virtual Disk Model
3. Create a virtual disk.

   See Creating a Virtual Disk for the Body Force Propeller Method
4. Visualize the virtual disk and the inflow velocity plane.

   See Visualizing the Virtual Disk and the Inflow Velocity Plane

Observing Virtual Disk Meshing Requirements for the Body Force Propeller Method

It is recommended that you refine the mesh in the zone where the virtual disk is located, particularly in the radial direction. This way, you obtain a good distribution of the axial and tangential body force components of the virtual disk.

For details on the body force computation, see Body Force Propeller Method.

You can achieve a suitable mesh refinement by use of a volumetric control that is based on a cylindrical part shape.

For more information, seeMesh Refinement In A Specified Volume.

The diameter of the cylindrical part shape should correspond to the diameter of the inflow velocity plane.

For more information, seeSpecifying the Inflow Velocity Plane.

If you want to analyze the flow field in the wake of the propeller in more detail, refine the wake zone to a length of approximately fives times the virtual disk diameter.

Note: If you set up a case from scratch, that is, the mesh is not yet available, it is recommended to first activate the virtual disk model. Then specify the geometry and location of the virtual disk. Create the required volumetric controls for mesh refinement and then proceed with the meshing.

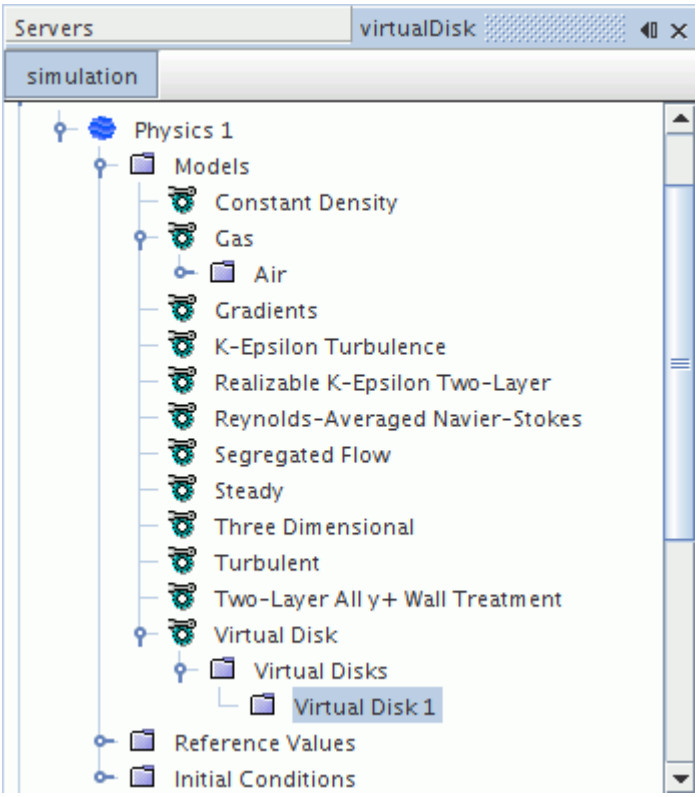
Creating a Virtual Disk for the Body Force Propeller Method

After having selected the virtual disk model, specify the details of the virtual disk.

To create a virtual disk:

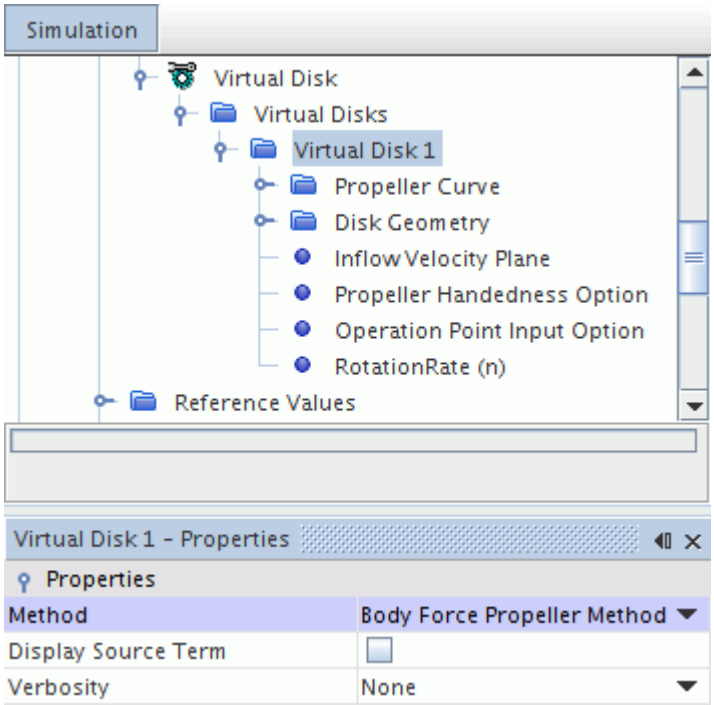
1. Right-click the **Virtual Disk** > **Virtual Disks** node and select **New**.

   A new child node that is called **Virtual Disk 1** is added to the simulation tree.



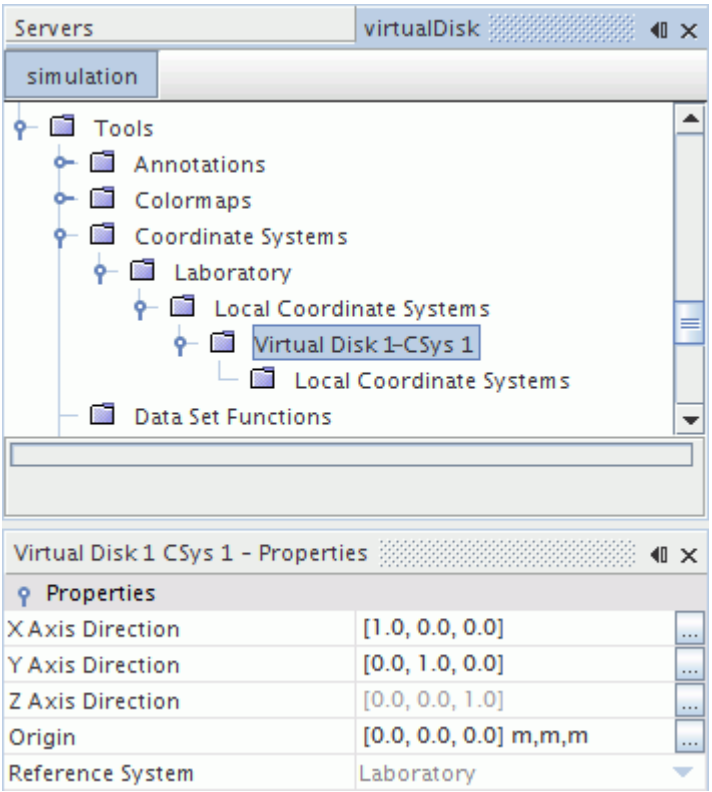
2. Select the **Virtual Disk 1** node and set *Method* to **Body Force Propeller Method**.

   The additional **Propeller Curve, Disk Geometry, Inflow Velocity Plane, Propeller Handedness Option, Operation Point Input Option,** and **RotationRate(n)** sub nodes are added to the simulation tree. The **RotationRate(n)** sub node can change depending on what you select as **Operation Point Input Option**



A new local coordinate system for the virtual disk is created automatically. You can find it under the **Tools** > **Coordinate Systems** > **Laboratory** > **Local Coordinate Systems** > **Virtual Disk 1-CSys 1** node. This coordinate system is oriented such that the z-axis is always normal to the plane of the disk and that it points in the direction of thrust that is produced by the virtual disk.

Do not modify the local coordinate system of the disk here in the **Local Coordinate Systems** node. Modify the virtual disk origin and orientation only under the **Virtual Disk** node in the relevant physics continuum.



For more information, Specifying the Disk Orientation.

3. If you want to output additional information on the body force propeller method during the simulation, set the Virtual Disk Properties.
4. Define the Virtual Disk by configuring the following nodes:
   - **Propeller Curve**

     Set the propeller performance curve as a function of advance ratio.

     See Setting_the_Propeller_Performance_Curve_for_the_Body_Force_Propeller_Method
   - **Disk Geometry**

     Define the disk geometry in terms of geometrical dimensions, location, and orientation.

     See Specifying_the_Disk_Geometry_for_the_Body_Force_Propeller_and_the_1D_Momentum_Methods
   - **Inflow Velocity Plane**

     The inflow velocity plane is located in front of the virtual disk. It is always defined in the direction of the disk normal.

     See Specifying the Inflow Velocity Plane
   - **Propeller Handedness Option**

     See Specifying_the_Propeller_Handedness_for_the_Body_Force_Propeller_Method
   - **Operation Point Input Option**

     See Specifying_the_Operation_Point_for_the_Body_Force_Propeller_Method

Setting the Propeller Performance Curve for the Body Force Propeller Method

Generally, the propeller performance is available from experiment in terms of thrust coefficient KT, torque coefficient KQ, and propeller efficiency η_0 as a function of advance ratio J.

(See Eqns. Eqn. (4011), Eqn. (4012), Eqn. (4013) for a definition of these values.)

To specify the propeller performance curve:

1. Select the **Virtual Disk 1** > **Propeller Curve**node and set *Method* to one of the following:

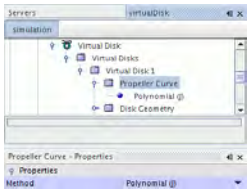
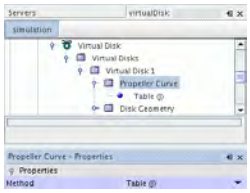
Option	Steps
Polynomial(J) 	1. Select the **Polynomial(J)** node and then click the ... (**custom editor**) next to the *Kt = f(J), Kq = f(J)* and *Eta = f(J)*. 2. In the *Polynomial(J)* dialog, enter the property values of the respective polynomial (Setting Values Using A Polynomial).
Table(J) 	1. Import the appropriate propeller performance curve table into the simulation (see Reading a Table Data File). 2. Select the **Table(J)** node. In the *Table (J) - Properties* window, set the following properties:

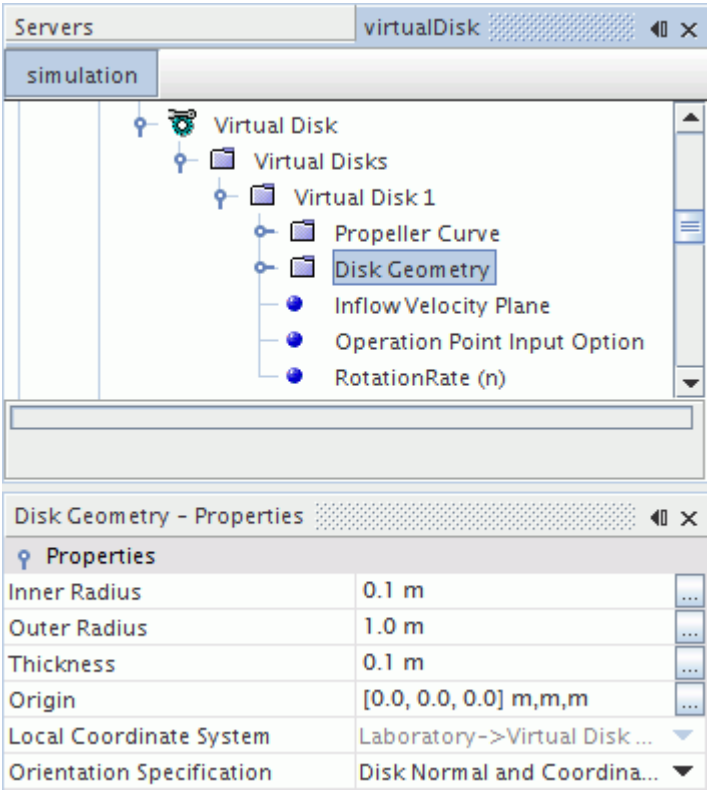
Table	Select the propeller performance curve table that you imported into your simulation.
Table: J	Select the column that contains the advance ratio for the selected propeller performance table.
Table: Kt	Select the column that contains the thrust coefficient for the selected propeller performance table.
Table: Kq	Select the column that contains the torque coefficient for the selected propeller performance table.
Table: Eta	Select the column that contains the propeller efficiency for the selected propeller performance table.

Specifying the Disk Geometry for the Body Force Propeller and the 1D Momentum Methods

The virtual disk is a simplified representation of a complex rotating machine such as a propeller or wind turbine. Only the geometrical dimensions of the disk are required. Details such as the number of blades, the chord, and twist variation are not necessary for this method.

To specify the virtual disk geometry:

1. Select the **Virtual Disk 1** > **Disk Geometry** node and set the Disk Geometry Properties.



Specifying the Disk Orientation

In addition to the geometric dimensions and the origin of the virtual disk, specify its orientation.

The way that the virtual disk is oriented, that is, in which direction the z-axis of the local disk coordinate system points, is the direction in which the virtual disk produces thrust. The positive z-axis points in the direction in which the disk would move if let free.

Before you begin to set the orientation of the virtual disk, it can be helpful to visualize the local disk coordinate system.

To visualize the local virtual disk coordinate system:

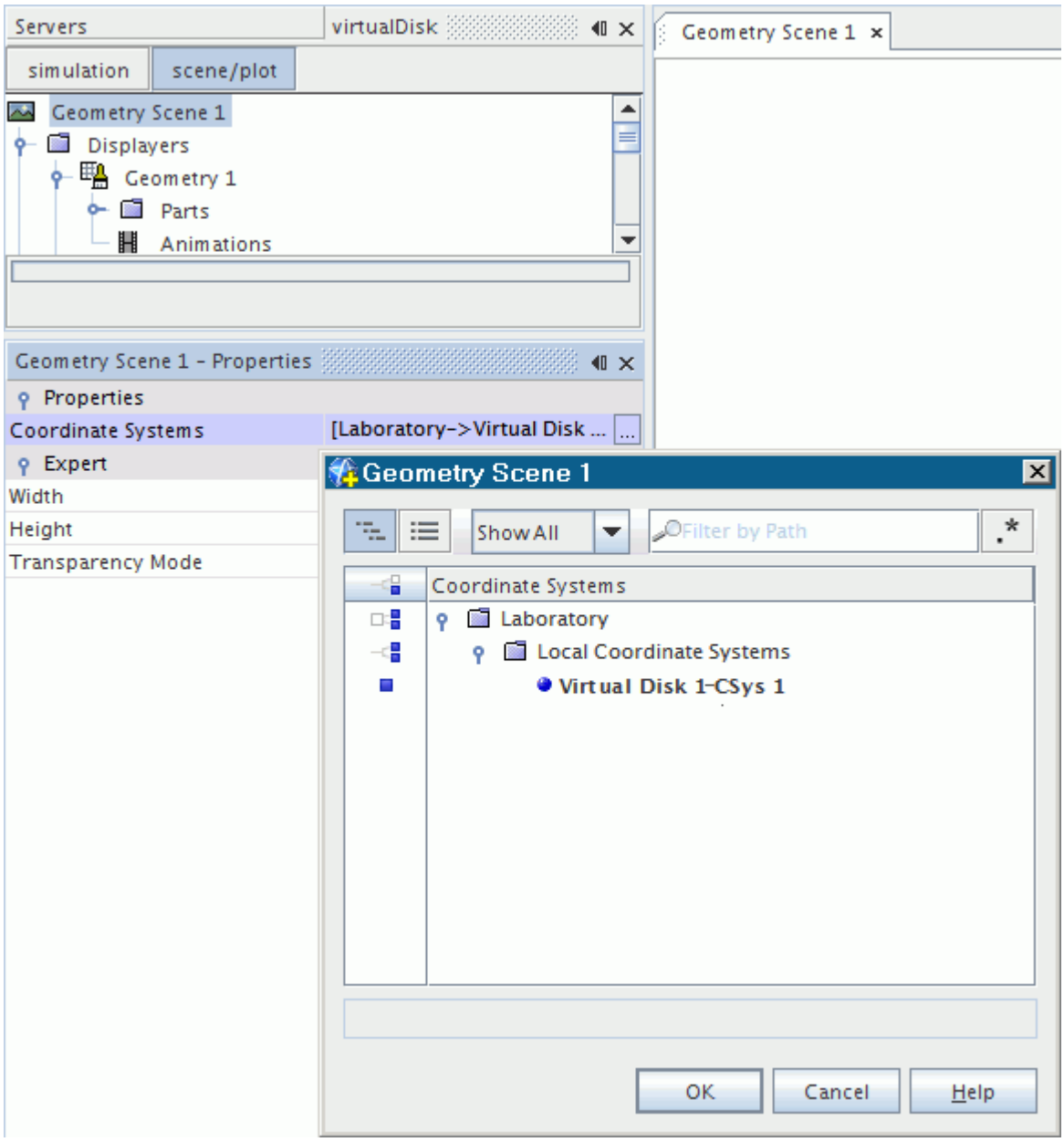
1. Open a scene, for example a geometry scene.

For more details, see Creating a Scene.

2. Click the **scene/plot** button.

3. In the scene node, for example the **Geometry Scene 1** node, click the ... (**custom editor)** and set the *Coordinate Systems property* to **[Laboratory-** > **Virtual Disk 1-CSys 1]**.

4. In the object selection dialog, select **Virtual Disk 1-CSys 1** and click **OK**.



You can now see the current orientation of the local coordinate system of the virtual disk in your currently open scene in the *Graphics* window. The orientation of this local coordinate system adapts in accordance with the input to the disk orientation.

You have two options available to specify the disk orientation:

- Disk orientation in terms of disk normal and coordinate system
- Disk orientation in terms of lateral and pitch angle

To specify the disk orientation in terms of disk normal and coordinate system:

5. Select the **Disk Geometry** node and set the *Orientation Specification* to **Disk Normal and Coordinate.**
6. Select the **Disk Geometry** > **Normal and Coordinate System** node.
7. In the *Normal And Coordinate System - Properties* window, set the following properties:

Coordinate System	Specify the coordinate system in regard to which the normal of the virtual disk is defined. You can also use a user-defined local cartesian coordinate system to define the disk normal.
Disk Normal	Specify the direction that is normal to the plane of the disk.

To specify the disk orientation in terms of disk lateral and pitch angle:

8. Select the **Disk Geometry** node and set the *Orientation Specification* to *Pitch Angle and Lateral Angle***.**
9. Select the **Disk Geometry** > **Lateral And Pitch Angle** node.
10. In the *Lateral and Pitch Angle - Properties* window, set the following properties:

Pitch Angle	To rotate the local disk coordinate system, you specify a pitch angle. The pitch angle rotates the x-z plane around the y-axis.
Lateral Angle	To rotate the local disk coordinate system, you specify a lateral angle. The lateral angle rotates the y-z plane around the x-axis.

Specifying the Radial Thrust and Torque Distribution

To create propulsion with the body force propeller method, thrust and torque vary in the radial direction, but not in the azimuthal direction. You specify the radial force distribution either by using the built-in Goldstein optimum distribution or by providing your own distribution. You can specify the same or a different distribution for thrust and torque.

To specify the radial thrust and torque distribution:

1. Select the **Virtual Disk** > **Radial Distribution Option** node and set *Thrust and Torque Distribution* to Radial Distribution Option Properties.
2. Select the corresponding thrust and torque specification nodes and set their Thrust and Torque Specification Properties.

Radial Thrust and Torque Distribution Properties Reference

Radial Distribution Option Properties

Thrust and Torque Distribution

Selects a common radial distribution for thrust and torque or different distributions for each.

- **Same Distribution**
  The distribution that you specify through the **Thrust and Torque Specification** node is the same for thrust and torque.
- **Different Distribution**

You specify thrust through the **Thrust Specification** node and torque through the **Torque Specification** node. Thrust and torque can be different and can also use different methods to specify the values.

Thrust and Torque Specification Properties

Depending on whether you select to have the same or a different distribution for thrust and torque, you use the following nodes to specify the properties:

- **Thrust and Torque Specification**
- **Thrust Specification**
- **Torque Specification**

All of these nodes have the same methods and properties. For all methods, the radial thrust and torque distributions must be specified as function of the dimensionless normalized disk span r^* as given by Eqn. (4008).

Method	Sub-node Properties
Goldstein's Optimum Distribution The radial thrust and torque distributions are calculated by using the Goldstein Optimum distribution. For more information, see Goldstein Optimum Distribution. The **Goldstein's Optimum Distribution** sub-node is created.	None
Field Function Distribution You specify the radial thrust and torque distributions by using a user-defined field function. The **Field Function Distribution** sub-node is created.	**Scalar Function** Make sure that you define the radial force distributions in your user-defined field function in terms of the field function that represents the normalized disk span r^* (see Virtual Disk Field Functions. For example, to specify the Goldstein distribution as a user-defined field function: `${VirtualDiskNormalizedSpanStar1} * sqrt(1.0 -${VirtualDiskNormalizedSpanStar1})`
Polynomial Distribution You specify the radial thrust and torque distributions by using a polynomial. The **Polynomial Distribution** sub-node is created.	**Polynomial** You specify the polynomial as a function of the normalized span r^*.
Tabular Distribution You specify the radial thrust and torque distributions by using a table. The *Tabular Distribution* and the **Table** (r^*) sub-nodes are created.	**Table: Data** Selects the column of the *Input Table* that contains the radial thrust and torque data. **Table: Normalized Span Star** Selects the column of the *Input Table* that contains the normalized disk span r^* values.

Method	Sub-node Properties
	Input Table Selects the name of the table that contains the radial thrust and torque data as a function of the normalized disk span r^* in the form of columns. **Columns** The column heading of the *Input Table* (read-only).

Setting the Inflow Specification

The body force propeller method and the 1D Momentum method require you to set the inflow specification of the virtual disk. As inflow specification, you set two quantities: an inflow velocity vector and an inflow density. The virtual disk uses these inflow quantities for the computation of the advance ratio in combination with the operation point of a propeller.

The body force propeller method uses the inflow velocity vector to compute the advance ratio J according to Eqn. (4017). J is then used to determine the operating point from the propeller characteristics curve yielding the thrust coefficient K_T and the torque coefficient K_Q. With these two coefficients, the thrust and the torque are evaluated (see Eqn. (4038) and Eqn. (4048)).

The 1D Momentum method uses the inflow velocity vector to calculate Eqn. (4044) to Eqn. (4055).

If the operation point is set to **Thrust (T)** or **Torque (Q)**, the inflow density value is required for the iterative calculation of the advance ratio according to Eqn. (4044) to Eqn. (4055).

To set the inflow specification:

1. Select the **Virtual Disk 1** > **Inflow Specification** node and set *Method* to the option that best suits your scenario.
2. Select the corresponding sub-nodes and set their properties. See Inflow Specification Properties Reference.

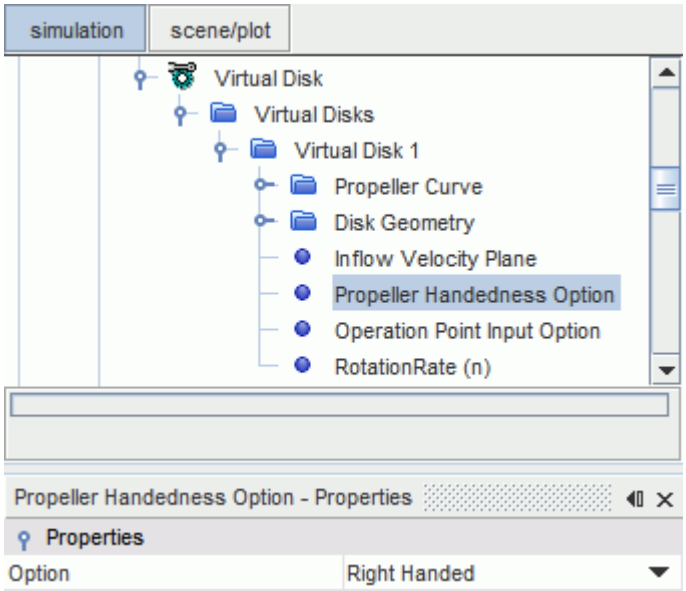
Specifying the Propeller Handedness for the Body Force Propeller Method

A propeller is characteristically classified in terms of handedness. You specify the handedness of the propeller to relate the direction of the rotation of the propeller to the direction of thrust of the vessel.

For more detailed information on how the propeller handedness is defined and how it interacts with the local coordinate system of the disk and the operation point, see Propeller Handedness.

To specify the propeller handedness:

1. Select the **Virtual Disk 1** > **Disk Geometry** > **Propeller Handedness Option** node.



2. In the *Propeller Handedness Option - Properties* window, select the Propeller Handedness Properties.

Propeller Handedness Option Properties

Option	Choose between two different propeller handedness options. For a definition of the propeller handedness and how it affects the directions of rotation, see Propeller Handedness .	
	Right Handed	Specifies the propeller is right-handed. A right-handed propeller always rotates in the counter-clockwise direction with respect to the positive z-axis of the local coordinate system of the virtual disk.
	Left Handed	Specifies the propeller is left-handed. A left-handed propeller always rotates in the clockwise direction with respect to the positive z-axis of the local coordinate system of the virtual disk.

Propeller Handedness

A propeller is - among other parameters - characterized by its handedness: it can be right-handed or left-handed. The handedness relates to the shape of the propeller, which in turn relates the propeller rotation direction to the direction of translation of the vessel.

In STAR-CCM+, the definition of the propeller handedness is modified compared to the traditional definition that is typically used by the marine community.

Propeller Handedness in the Marine Industry

The definition of the handedness requires a reference direction. Traditionally, the marine community defines the propeller handedness with respect to the stern of the vessel:

- A right-handed propeller rotates clockwise to produce forward thrust when viewed from the stern of the vessel.
- A left-handed propeller rotates counter-clockwise to produce forward thrust when viewed from the stern of the vessel.

The handedness of a propeller affects its shape. When you view a propeller from astern, the leading edges are farther away from you than the trailing edges. For example, the leading edges of a right-handed propeller are on the right and the propeller rotates clockwise when viewed from astern.

Propeller Handedness in STAR-CCM+

The selection of propeller handedness is available as an option as part of the Body Force Propeller method.

Generally, the design of a propeller is optimized to be used for forward motion. However in practice, a propeller can also be used for the reverse motion of a ship. Considering this fact, STAR-CCM+ does not use the stern of the vessel as the reference direction for the definition of left-/right-handedness as is the case for the traditional definition (see Propeller Handedness in the Marine Industry).

In STAR-CCM+, the local coordinate system of the virtual disk is used as the reference for defining the propeller handedness:

- A right-handed propeller rotates counter-clockwise when viewed from the positive direction of the z-axis of the local coordinate system of the virtual disk.
- A left-handed propeller rotates clockwise when viewed from the positive direction of the z-axis of the local coordinate system of the virtual disk.

The use of the local coordinate system of the virtual disk does not change the definition of handedness, only the reference axis for the direction of rotation is changed. This approach is consistent because the local coordinate system of the virtual disk changes according to the direction of propulsion of the vessel. The propulsion direction is specified in terms of the orientation specification of the virtual disk.

Rotation Direction

The Propeller Handedness Option is useful to prescribe or change the direction of rotation of the propeller - clockwise or counter-clockwise - in combination with the Operation Point Input Option (Specifying_the_Operation_Point_for_the_Body_Force_Propeller_Method) and the Orientation Specification of the virtual disk (Specifying the Disk Orientation).

To specify the operation point, there are three options available:

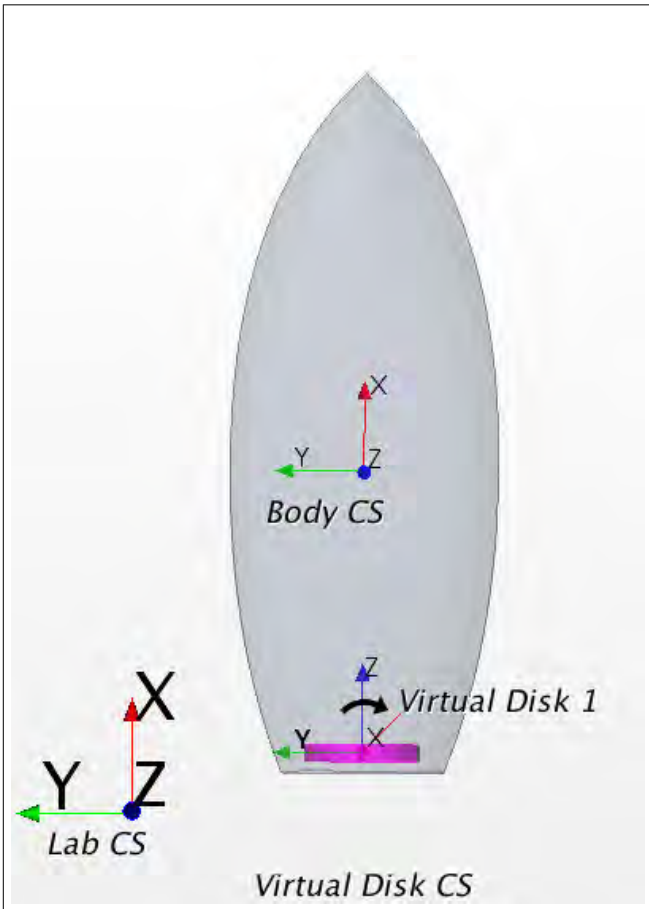
- Rotation rate with $\omega \geq 0$.
- Thrust with $T \geq 0$.
- Torque with $Q \geq 0$.

The direction of propulsion is specified in terms of the orientation specification of the virtual disk. Forward and reverse propulsion direction is defined in terms of the selected coordinate system and its orientation with respect to the nose and the stern of the vessel.

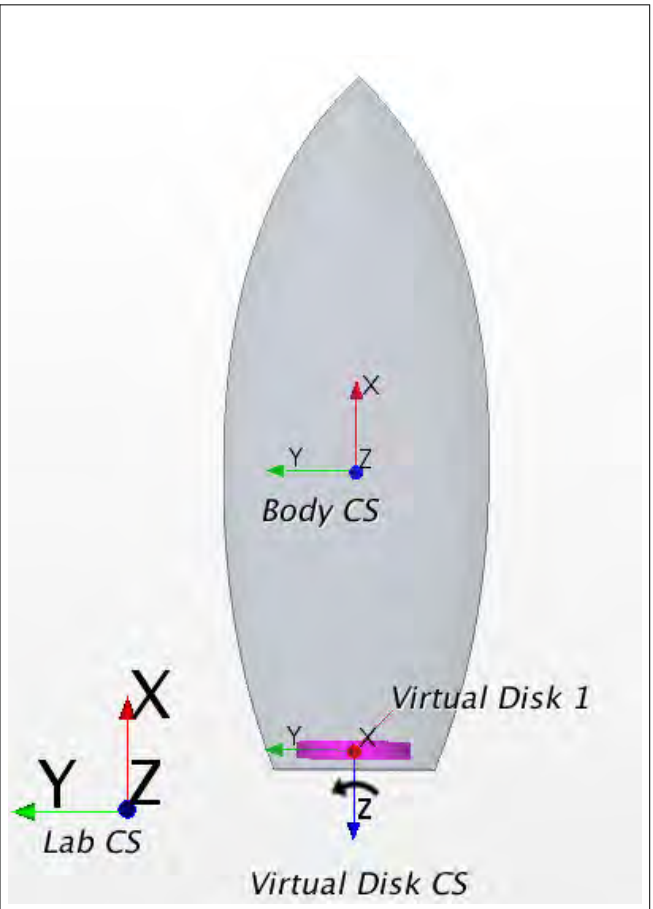
In the following table, it is assumed that the **Disk Normal and Coordinate System** option is used as *Orientation Specification*. The combination of propeller handedness and orientation of the local coordinate system of the virtual disk and its effect on the directions of propulsion and propeller rotation is summarized as follows:

Handedness	Normal	Propulsion Direction	Rotation Direction
Right	(1,0,0)	Forward	CCW
Right	(-1,0,0)	Backward	CCW
Left	(1,0,0)	Forward	CW
Left	(-1,0,0)	Backward	CW

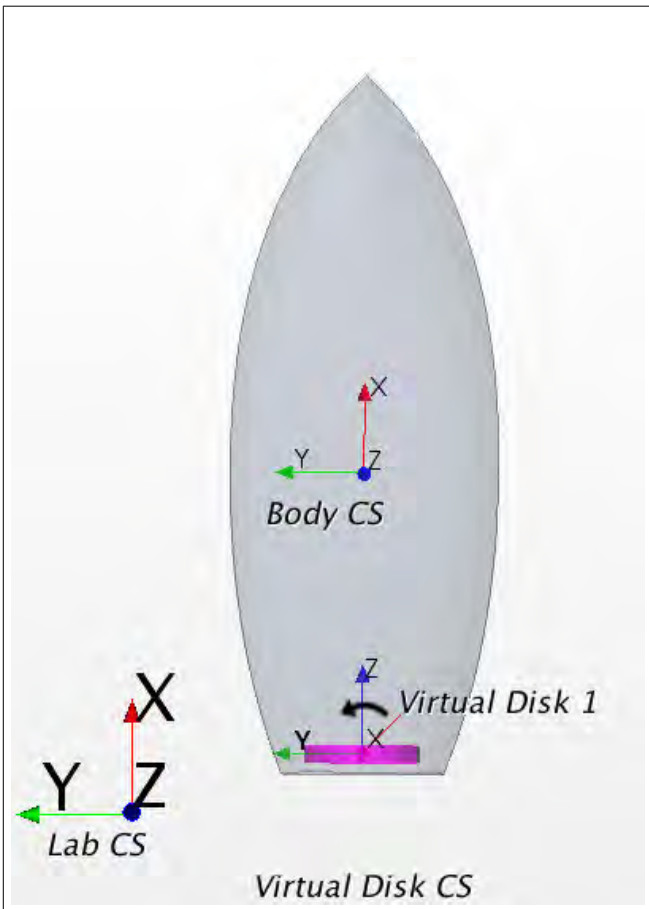
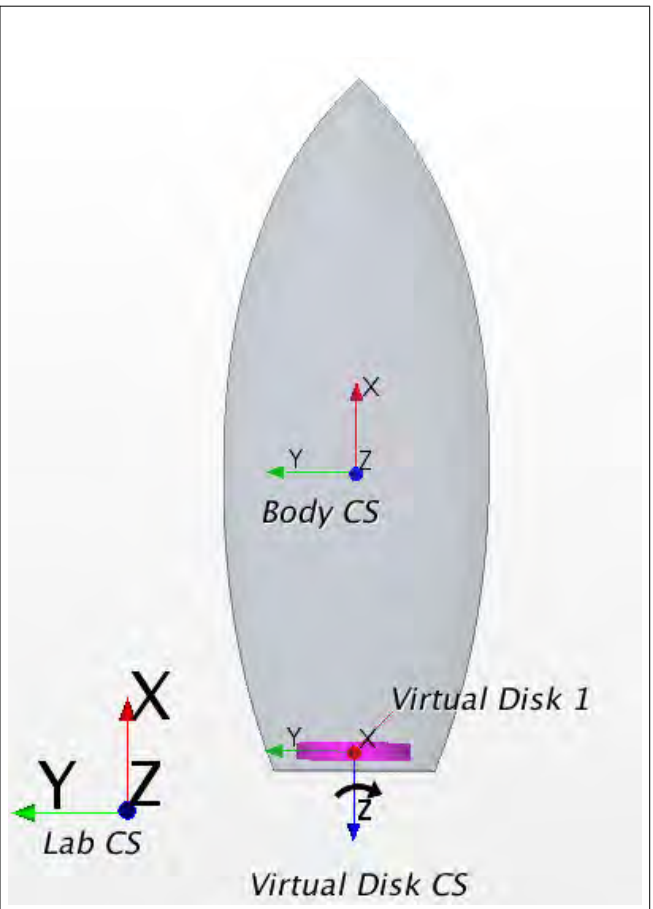
In the following figures, the orientation of the virtual disk is specified in terms of the Body coordinate system.The orientation of the local coordinate system of the virtual disk automatically adapts to the normal direction specified with respect to the Body coordinate system. The direction of propeller rotation for each type of handedness is shown:



Right-handed propeller in forward configuration



Right-handed propeller in backward configuration

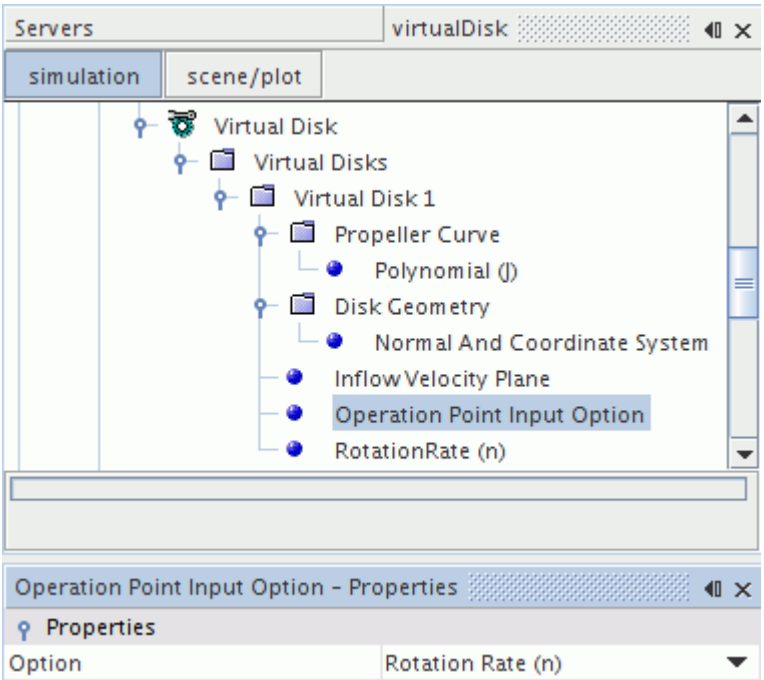
	
Left-handed propeller in forward configuration	Left-handed propeller in backward configuration

Specifying the Operation Point for the Body Force Propeller Method

The operation point defines where on the specified propeller performance curve you want to perform the simulation.

To specify the operation point:

1. Select the **Virtual Disk 1** > **Operation Point Input Option** node.



2. In the *Operation Point Input Option - Properties* window, select the following property:

Option	Choose between three different operation point options.	
	Rotation Rate (n)	Select the rotation rate as the operation point.
	Thrust (T)	Select the thrust as the operation point.
	Torque (Q)	Select the torque as the operation point.

Depending on your previous selection under the **Operation Point Input Option node**, either the **RotationRate (n)**, the **Thrust**, or the **Torque** sub node appears.

3. Specify the respective operating point value and units in this sub node.

Blade Element Method

The blade element method is primarily designed to analyze the complex flow field that is associated with helicopter rotors.

The complexity of the flow arises due to mutual aerodynamic interactions among the main rotor, the fuselage, and the tail rotor. The asymmetrical loading over the rotor disk complicates these interactions. A correct solution of the flow field around a rotor plays a crucial role in determining the aerodynamic characteristics and the performance of the helicopter.

The blade element method models the spinning rotor as a distribution of momentum sources. The strength of the source terms and their variations are interactively determined from the rotor geometry and the local velocity field. The number and the geometry of the blades in terms of chord and twist variation along the rotor radius are specified. The blade geometry is not explicitly resolved. The influence of the blade geometry on the flow field is described by the aerodynamic behavior of the blade in terms of lift and drag coefficients that are obtained for the corresponding two-dimensional cross-section of the blade. The aerodynamic characteristic of a blade can vary along the radius for an unlimited number of airfoil cross-sections. In forward flight, the rotor has an asymmetrical loading, which leads to undesired moments with respect to the rotor hub. Therefore, rotor trimming is necessary. The implemented method allows you to trim the rotor to the specified thrust and moments by using a non-linear iterative approach.

The blade element method has numerous advantages over a transient rigid body motion approach for this application. It eliminates the need for a body-fitted mesh along the blades, because the blades are not present physically. The wake structure is well captured and requires no extra model. The virtual disk approach allows the intersecting or overlapping of rotors. There is practically no restriction on the number of rotors. It is possible to study the main rotor and the tail rotor interaction of a single helicopter or numerous helicopters in a flying configuration. The method is not restricted to helicopter applications and can also be used to study automotive radiator fans.

The blade element method provides a computationally cost-effective and reliable method to obtain a flow field, which has enough details to make design decisions leading to an improved overall performance.

Contents:

Setting up and Using the Blade Element Method
Creating a Virtual Disk for the Blade Element Method
Specifying the Airfoil Sections
Airfoil Section Properties
Lift Coefficient Properties
Specifying the Chord Distribution
Chord Distribution Properties
Table (r/R) Properties
Specifying the Sweep Angle Distribution
Sweep Angle Distribution Properties
Specifying the Twist Distribution
Twist Distribution Properties
Specifying the Disk Geometry for the Blade Element Method

Disk Geometry Properties for Blade Element Method
Setting the Rotation Rate for the Blade Element Method
Ramping the Rotation Rate
Disk Stick Specification
Specifying the Disk Stick Specification
Setting the Disk Flap Specification
Disk Flap Specification Properties
Specifying the Virtual Disk Resolution
Virtual Disk Resolution Properties Reference
Tip Correction for the Blade Element Method
Setting the Source Convergence Control
Setting the Trim Convergence Control
Setting the Target Values for Trimming

Setting Up and Using the Blade Element Method

Similar to the body force propeller method, the setup of the virtual disk for the blade element method does not require the creation of a separate region to which the method is then applied. Instead, you create the shape of the virtual disk by specifying the relevant parameters such as radius, thickness, and orientation to an existing mesh.

To verify the dimensions and orientation of the virtual disk, you can visualize it by using virtual disk-specific field functions.

To set up and run the blade element method:

1. Select the virtual disk model.

   See Selecting the Virtual Disk Model
2. Create a virtual disk.

   See Creating a Virtual Disk for the Blade Element Method
3. Visualize the virtual disk.

   See Visualizing the Virtual Disk and the Inflow Velocity Plane

Creating a Virtual Disk for the Blade Element Method

After having selected the blade element method, specify the details of the virtual disk.

The following components are important for the blade element method:

Geometric definition of the blades

- Origin and orientation
- Outer radius and root cut-out radius
- Number of blades

Blade information

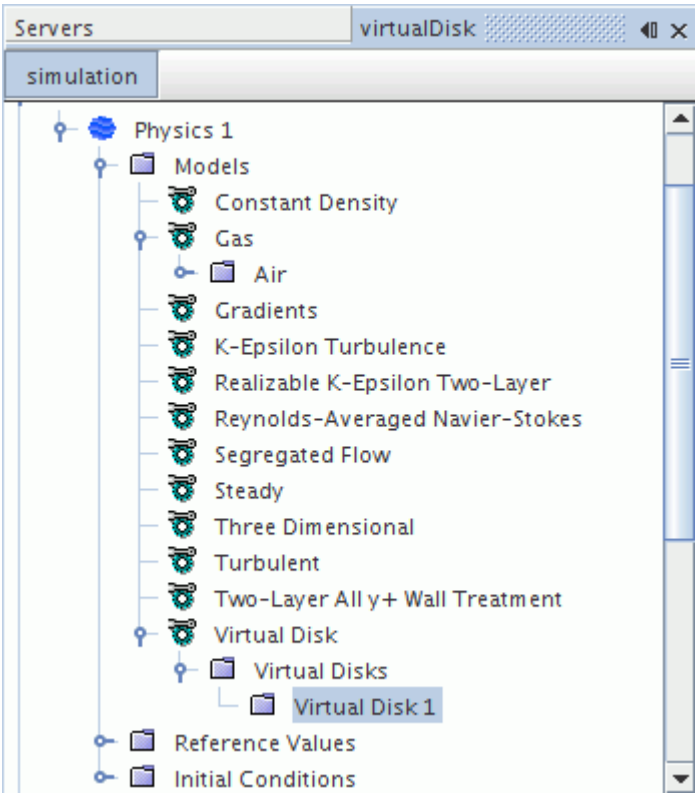
- Chord
- Twist
- Airfoil data (Drag and lift coefficients as a function of angle of attack and Mach/Reynolds number)

Rotor conditions

- Rotation rate
- Collective pitch
- Lateral and longitudinal cyclic pitch
- Trim Condition

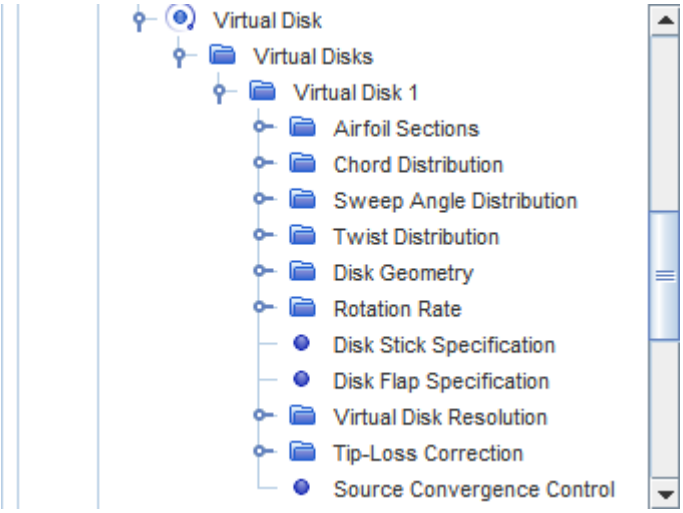
To create a virtual disk:

1. Right-click the **Virtual Disk** > **Virtual Disks** node and select **New**.



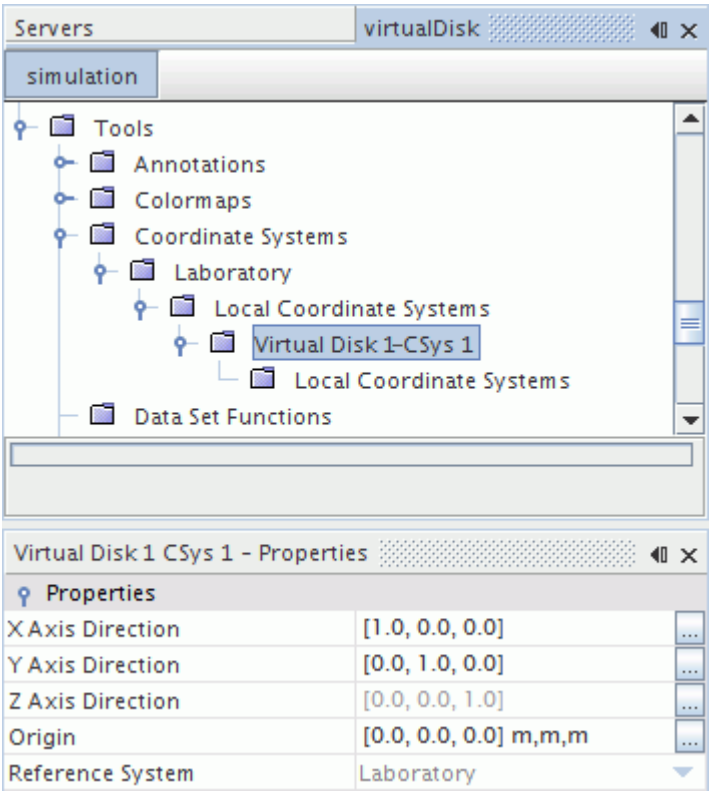
2. Select the **Virtual Disk 1** node and set *Method* to **Blade Element Method**

   The additional **Airfoil Sections, Chord Distribution, Twist Distribution, Disk Geometry, Rotation Rate, Disk Stick Specification, Virtual Disk Resolution, Tip-Loss Correction, and Source Convergence Control** sub nodes are added to the simulation tree.



A new local coordinate system for the virtual disk is created automatically. You can find it under the **Tools** > **Coordinate Systems** > **Laboratory** > **Local Coordinate Systems** > **Virtual Disk 1-CSys 1** node. This coordinate system is oriented such that the z-axis is always normal to the plane of the disk.

Do not modify the local coordinate system of the disk here in the **Local Coordinate Systems** node. Modify the virtual disk origin and orientation only under the **Virtual Disk** node in the relevant physics continuum.



For more information, see Specifying the Disk Orientation.

3. If you want to output additional information on the blade element method during the simulation, set Virtual Disk Properties.
4. Define the Virtual Disk by configuring the following nodes:
   - **Airfoil Sections**

     See Specifying the Airfoil Sections
   - **Chord Distribution**

     See Specifying the Chord Distribution
   - **Sweep Angle Distribution**

     See Specifying the Sweep Angle Distribution
   - **Twist Distribution**

     See Specifying the Twist Distribution
   - **Disk Geometry**

     See Specifying the Disk Geometry for the Blade Element Method
   - **Rotation Rate**

     See Setting the Rotation Rate for the Blade Element Method
   - **Disk Stick Specification**

     See Specifying the Disk Stick Specification
   - **Disk Flap Specification**

     Setting the Disk Flap Specification
   - **Virtual Disk Resolution**

     See Specifying the Virtual Disk Resolution
   - **Tip-Loss Correction**

     Tip-Loss Correction for the Blade Element Method
   - **Source Convergence Control**

     See Setting the Source Convergence Control

Specifying the Airfoil Sections

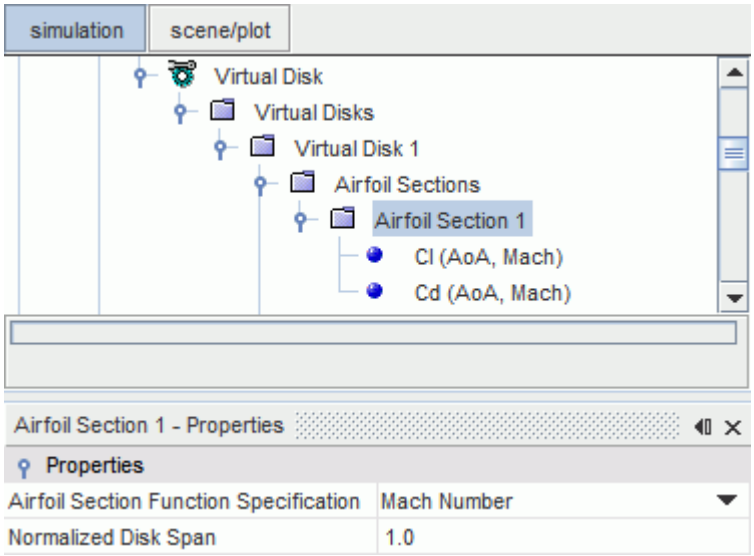
You can use the **Airfoil Sections** node to specify drag and lift coefficients as functions of Reynolds or Mach number and angle of attack. Providing this data brings the effect of local blade aerodynamics into the virtual disk simulation.

An airfoil section is a cross-section of the blade at a certain radial position. You can either specify one airfoil section for the entire blade or you can subdivide the blade into several cross-sections and provide local drag and lift coefficient data.

To specify an airfoil section:

1. Right-click **Virtual Disk 1** > **Airfoil Sections**and select **New**.

   An **Airfoil Section 1** node with two sub-nodes is added to the simulation tree.



2. In the *Airfoil Section 1 - Properties* window, set the appropriate properties.

   See Airfoil Section Properties

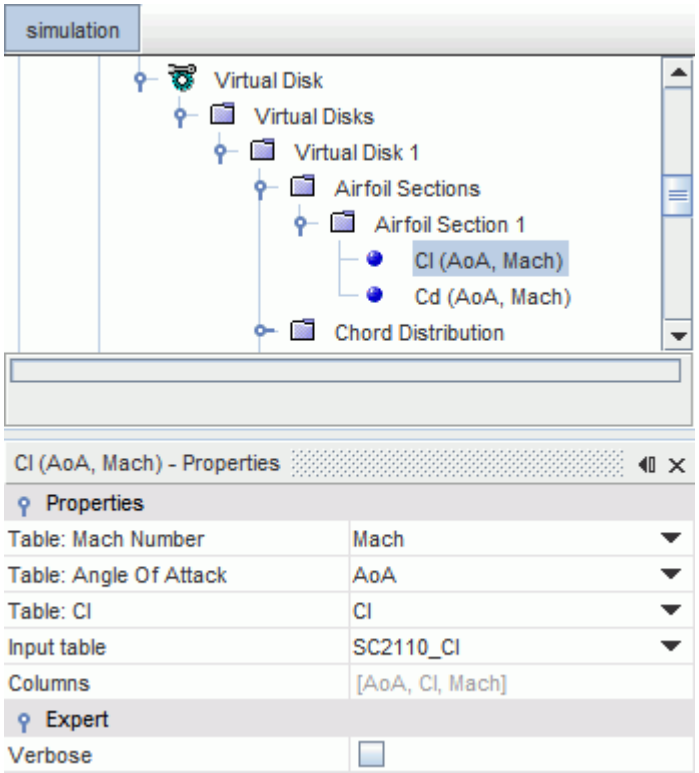
3. Import the airfoil data, that is the drag and lift coefficients, each as a table. The following procedure describes the setup for the lift coefficient as a function of Mach number. The procedure is the same for the drag coefficient and for a dependence on the Reynolds number.

   There are two table format options available:

   - File table
     1. Import the appropriate lift coefficient table into the simulation.

        See Reading a Table Data File

     2. Select the **Cl (AoA, Mach)** node.



3. In the *Cl (AoA, Mach) - Properties* window, set the appropriate properties.

   See Lift Coefficient Properties.

- C81 table
  1. Right-click the **Virtual Disk 1** > **Airfoil Sections** node and select **Import C81 Airfoil Section Data...**
  2. In the *Open* dialog, set the *Airfoil Data Function* to **(alpha, Mach)** and navigate to the C81 airfoil data file. The file contains the data for both the lift and the drag coefficient.
  3. Click **Open**.

     Two nodes for the lift and drag coefficients, respectively, are added to the **Tools** > **Tables** node.
  4. Select the **Virtual Disk 1** > **Airfoil Sections** > **Airfoil Section 1** > **Cl (AoA, Mach)** node and set the appropriate properties in the *Cl (AoA, Mach) - Properties* window.

     See Lift Coefficient Properties.

Airfoil Section Properties

Airfoil Section Function Specification	You specify the local aerodynamic coefficients of the rotor blade for this airfoil section at a certain radial position. Depending on the selection here, the **Cl** and **Cd** sub-nodes change.	
	Mach Number	Specify the aerodynamic coefficients as a function of angle of attack and the Mach number.

	Reynolds Number	Specify the aerodynamic coefficients as a function of angle of attack and the Reynolds number.
Normalized Disk span	Dimensionless radial coordinate along the blade with values from 0 through 1. It is defined as $\frac{r}{R}$ where r is the radial coordinate and R is the maximum blade radius. Here, you specify the radial position of the airfoil section.	

Lift Coefficient Properties

Table: Mach Number	Select the column that contains the Mach number for the selected input table.	
Table: Angle of Attack	Select the column that contains the angle of attack for the selected input table.	
Table: Cl	Select the column that contains the lift coefficient for the selected input table.	
Input table	Select the table that contains the airfoil section lift coefficient data.	
Columns	Lists the imported columns of the selected input table (read-only).	
Verbose	Activates additional output.	

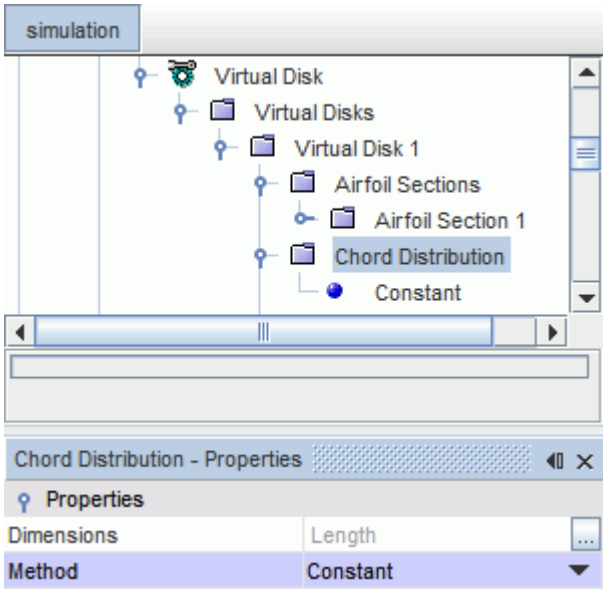
Specifying the Chord Distribution

The chord is a straight line connecting the leading and the trailing edge of a blade or airfoil. It describes the width of the blade.

For the blade element method, you specify the chord length either constant or as a function of the normalized disk span r/R.

To specify the chord distribution:

1. Select the **Virtual Disk 1** > **Chord Distribution** node.



2. In the *Chord Distribution - Properties* window, set the appropriate properties.

   See Chord Distribution Properties.

3. Corresponding to the *Method* that you specified, perform one of the following steps:
   - **Constant**
     1. Select the **Chord Distribution** > **Constant** node and specify a chord length *Value*.
   - **Polynomial in (r/R)**
     1. Select the **Chord Distribution** > **Polynomial in (r/R)**node and then click the ... (**custom editor**) next to *Polynomial*.
     2. In the *Polynomial in (r/R)* dialog, enter the property values of the respective polynomial see (Setting Values Using A Polynomial).
   - **Table (r/R)**
     1. Import the appropriate chord distribution table into the simulation see (Reading a Table Data File).
     2. Select the **Table (r/R)** node. In the *Table (r/R) - Properties* window, set the appropriate properties.

        See Table r R Properties.

Chord Distribution Properties

Dimensions	Gives the dimensions of the specified chord (read-only).	
Method	You specify the method how to provide the chord length data.	
	Constant	You set a constant value for the chord.

	Polynomial in (r/R)	You specify the chord distribution as a function of dimensionless disk span in the form of a polynomial.
	Table (r/R)	You specify the chord distribution as a function of dimensionless disk span in the form of a table.

Table (r/R) Properties

Table: Data	Select the column that contains the chord or twist distribution data.
Table: Normalized Span	Select the column that contains the position in the form of normalized span (r/R) for the selected chord or twist distribution.
Input table	Select the chord or twist distribution table that you imported into your simulation
Columns	Displays the extracted columns from the table (read-only).
Verbose	Activates additional output information.

Specifying the Sweep Angle Distribution

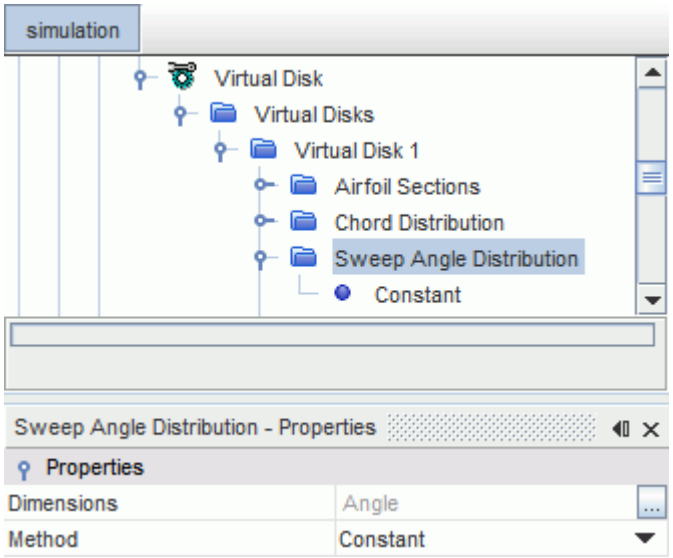
Similar to the wings of a subsonic aircraft, the rotor blades of a helicopter can experience the effects of wave drag at the tip of the blades. Wave drag occurs when the air flow reaches the speed of sound at the leading edge at the tip of the blade.

To avoid the occurrence of wave drag, the local velocity at the blade tip must be kept below the speed of sound. The local tangential velocity is given by the rotation rate times the blade radius ($v_\phi = r\omega$). Therefore, these wave drag effects restrict the achievable maximum rotation rates in combination with the blade radius to a certain limit.

To remedy this limitation, the wave drag effects can be reduced by introducing a sweep-back of the rotor blades. Sweeping the blade reduces the local velocity at the wing section by a factor of cosine of the angle of sweep. The sweeping thus increases the critical Mach number.

To specify the sweep angle distribution:

1. Select the **Virtual Disk 1** > **Sweep Angle Distribution** node.



2. In the *Sweep Angle Distribution - Properties* window, set the appropriate properties.

   See Sweep Angle Distribution Properties.

3. Corresponding to the *Method* that you specified in the previous step, perform one of the following steps:
   - **Constant**
     1. Select the **Sweep Angle Distribution** > **Constant** node and specify a sweep angle *Value*.
   - **Polynomial in (r/R)**
     1. Select the **Sweep Angle Distribution** > **Polynomial in (r/R)**node and then click ... (**custom editor**) next to *Polynomial*.
     2. In the *Polynomial in (r/R)* dialog, enter the property values of the respective polynomial (see Setting Values Using A Polynomial).
   - **Table (r/R)**
     1. Import the appropriate sweep angle distribution table into the simulation (see Reading a Table Data File).
     2. Select the **Table (r/R)** node. In the *Table (r/R) - Properties* window, set the appropriate properties.

        See Table (r/R) Properties.

Sweep Angle Distribution Properties

The **Sweep Angle Distribution** node requires the setting of several properties.

See the following table for descriptions of these properties:

Dimensions	Specifies the dimensions of the sweep angle data (read-only). It is set to **Angle**.
Method	Specifies how the sweep angle data is provided.

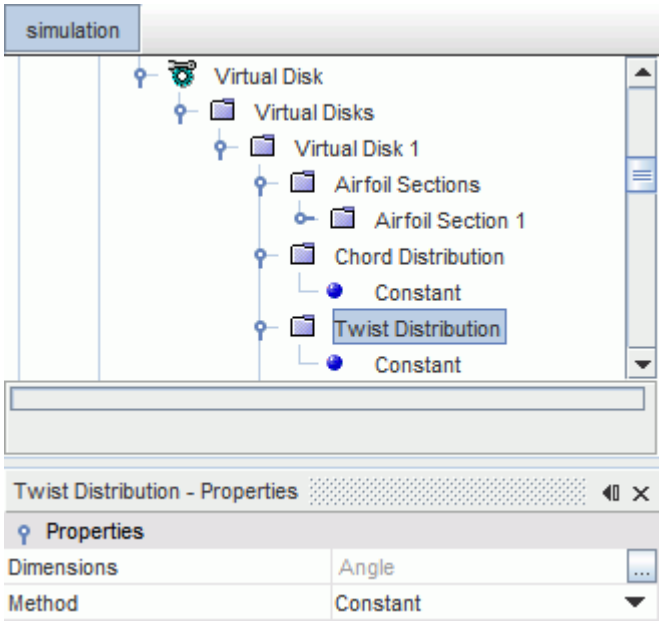
	Constant	Sets a constant value for the sweep angle.
	Polynomial in (r/R)	Specifies the sweep angle distribution as a function of dimensionless disk span in the form of a polynomial.
	Table (r/R)	Specifies the sweep angle distribution as a function of dimensionless disk span in the form of a table.

Specifying the Twist Distribution

A blade can be built with a twist to achieve different pitch angles between the hub and the tip. You can specify the twist distribution as constant or as a function of normalized disk span r/R.

To specify the twist distribution:

1. Select the **Virtual Disk 1** > **Twist Distribution** node.



2. In the *Twist Distribution - Properties* window, set the appropriate properties.

   See Twist Distribution Properties.

3. Corresponding to the *Method* that you specified, perform one of the following steps:
   - **Constant**
     1. Select the **Twist Distribution** > **Constant** node and specify a twist length *Value*.
   - **Polynomial in (r/R)**

1. Select the **Twist Distribution** > **Polynomial in (r/R)**node and then click the ... (**custom editor**) next to *Polynomial*.
2. In the *Polynomial in (r/R)* dialog, enter the property values of the respective polynomial (see Setting Values Using A Polynomial).

- **Table (r/R)**
  1. Import the appropriate twist distribution table into the simulation (see Reading a Table Data File).
  2. Select the **Table (r/R)** node. In the *Table (r/R) - Properties* window, set the appropriate properties.

     See Table (r/R) Properties.

Twist Distribution Properties

Dimensions	Gives the dimensions of the specified twist (read-only). It is set to **Angle**.	
Method	You specify the method how to provide the twist data.	
	Constant	You set a constant value for the twist.
	Polynomial in (r/R)	You specify the twist distribution as a function of dimensionless disk span in the form of a polynomial.
	Table (r/R)	You specify the twist distribution as a function of dimensionless disk span in the form of a table.

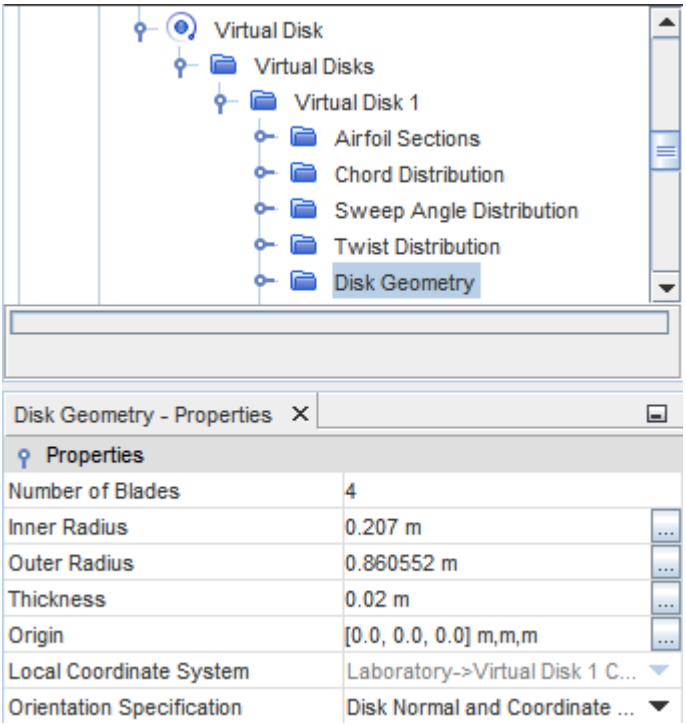
Specifying the Disk Geometry for the Blade Element Method

Here, you define the geometrical dimensions of the disk together with its location and orientation within the computational domain.

The disk for the blade element method is of cylindrical shape. Specify the thickness of the disk at least equal to or 10-15% greater than the cell size of the volume mesh where the disk is located. This way you ensure that the cells comprising the disk are continuous (that is, there is no hole in the disk) and that the disk is at least one cell layer thick. It is not recommended that you specify a larger thickness for the disk, because this larger thickness leads to a smearing of the local effect of the blades.

To describe the disk geometry:

1. Select the **Virtual Disks** > **Virtual Disk 1** > **Disk Geometry** node.



2. In the *Disk Geometry - Properties* window, set the appropriate properties.

   See Disk Geometry Properties for Blade Element Method

Disk Geometry Properties for Blade Element Method

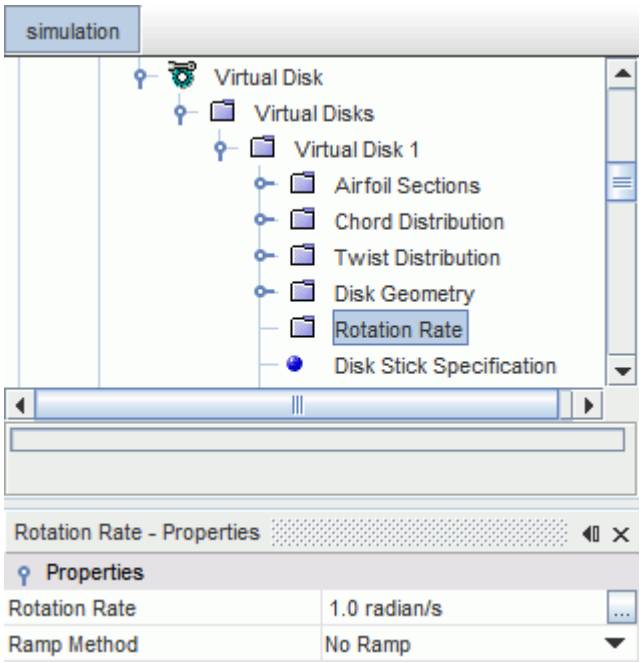
Number of Blades	You specify the number of blades of the rotor geometry that you want to model
Inner Radius	The inner radius corresponds to the root cut-out radius. The cut-out radius is typically 5% of the outer radius of the rotor.
Outer Radius	The outer radius corresponds to the tip radius of the rotor.
Thickness	You specify the thickness of the virtual disk for the blade element method. The thickness must be chosen such that the virtual disk is approximately one cell layer thick. Therefore, it is recommended that you set the thickness to a value that corresponds to the mesh cell size in the area of the virtual disk.
Origin	Specify the coordinates of the origin of the disk. The origin is defined in terms of the coordinates and the coordinate system with respect to which the coordinates are specified. Here, the choice of coordinate system is dictated by the selection you make for the **Orientation Specification** property: • For the option **Disk Normal and Coordinate System,**the coordinates of the origin are specified with respect to the coordinate system that is selected in that node.

	• For the option **Pitch Angle and Lateral Angle,**the coordinates of the origin are always with respect to the laboratory coordinate system.
Local Coordinate System	Reference to the local coordinate system of the disk (read-only).
Orientation Specification	You specify the disk orientation by selecting this option. The disk orientation defines in which direction the thrust is produced. For more information, see Specifying the Disk Orientation .

Setting the Rotation Rate for the Blade Element Method

To set the rotation rate for the blade element method:

1. Select the **Virtual Disks** > **Virtual Disk 1** > **Rotation Rate** node.



2. In the *Rotation Rate - Properties* window, set the following properties:

Rotation Rate	The rotation rate of the virtual disk.	
Ramp Method	Specify the ramping of the rotation rate. For high rotation rates, this ramping can improve convergence.	
	No Ramp	Ramping of the rotation rate is deactivated.
	Linear Ramp	The rotation rate is linearly ramped from a specified initial value to the final specified *Rotation Rate*. When you select

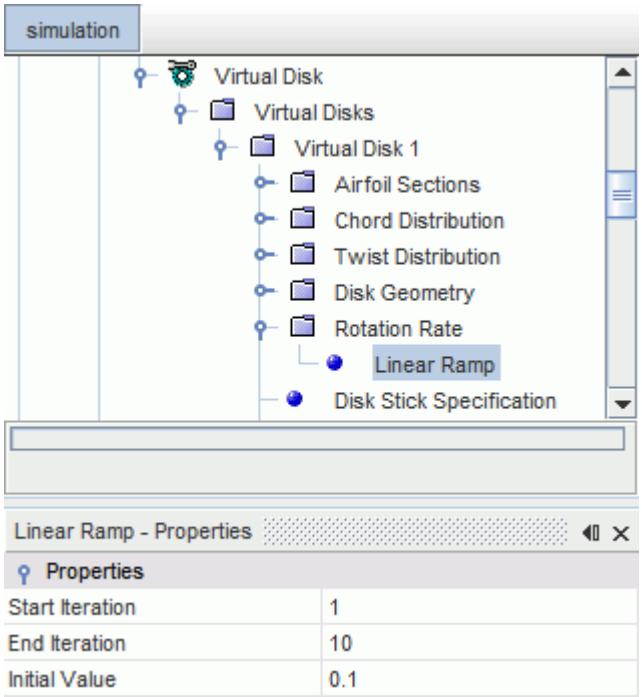
		this option, an additional Linear Ramp sub-node appears in the simulation tree. For more information, see Ramping the Rotation Rate .

Ramping the Rotation Rate

If you observe convergence problems that are related to a high rotation rate, you can linearly ramp the rotation rate.

To ramp the rotation rate:

1. Select the **Virtual Disks** > **Virtual Disk 1** > **Rotation Rate** node and set the *Ramp Method* to **Linear Ramp**.
2. Select the **Rotation Rate** > **Linear Ramp** node.



3. In the *Linear Ramp - Properties* window, set the following properties:

Start Iteration	You specify the iteration on which to begin the ramping.
End Iteration	You specify the iteration on which to end the ramping.
Initial Value	You specify the low rotation rate value with which to begin the ramping.

Disk Stick Specification

Particularly for aerospace applications, it is important to take the pitch variation of the blades into account.

For a helicopter, the pitch angle of the blade θ is important in establishing the angle of attack of the blade with respect to the rotor disk. Due to the asymmetrical nature of the rotor loading in forward flight, the pitch angle needs special treatment.

If the pitch angle of all blades were kept constant, the advancing side would have a higher angle of attack than the receding side.

To avoid such loading, the pitch angle consists of a collective part θ_{r0}, which is the same for each blade and the cyclic pitch component, which varies in the azimuthal direction. The complete blade pitch angle is then defined by the first harmonic term of a Fourier series:

$$\theta_r(\psi) = \theta_{r0} - \theta_{rc}\cos\psi - \theta_{rs}\sin\psi \tag{376}$$

where θ_{r0} is the collective pitch, θ_{rc} is the lateral cyclic pitch, θ_{rs} is the longitudinal cyclic pitch, and ψ is the azimuthal angle of the blade.

You can either specify all three components of the pitch angle (fixed rotor) or you can select the trim algorithm to calculate them. When using the trim routine to calculate these angles, provide an initial guess for the angles. The values of the desired thrust and moments are used to calculate iteratively the new values of the pitch angles.

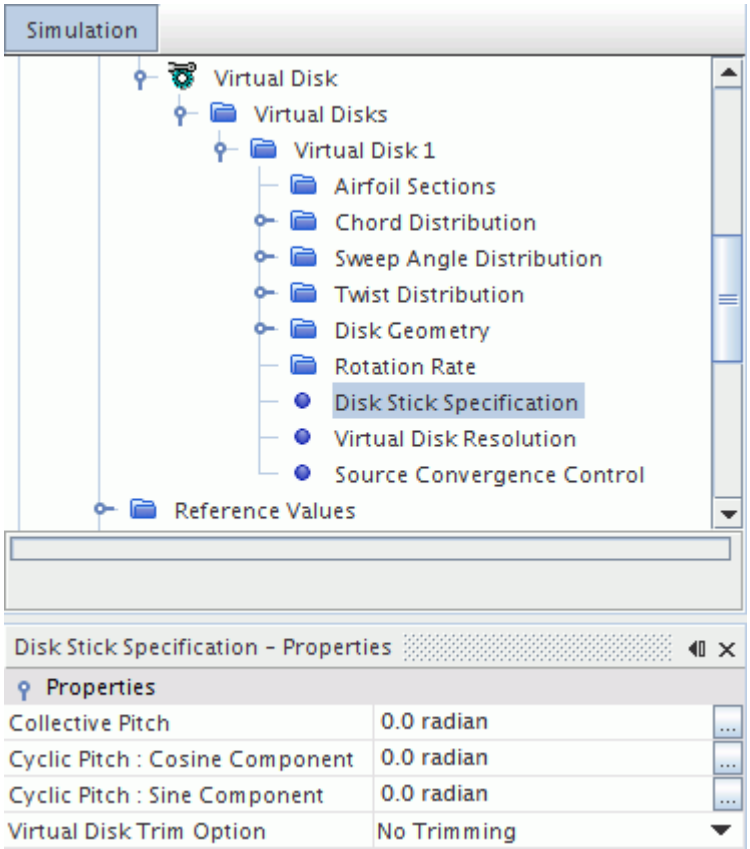
Setting the Disk Stick Specification

For a rotor, the pitch angle of the blade θ is important in establishing the angle of attack of the blade with respect to the rotor disk. Due to the asymmetrical nature of the rotor loading in forward flight, the pitch angle needs special treatment. You can either specify all three components of the pitch angle (fixed rotor) or you can select the trim algorithm to calculate them. When using the trim routine to calculate these angles, provide an initial guess for the angles. The values of the desired thrust and moments are used to calculate iteratively the new values of the pitch angles.

For more information on the blade pitch angles and the trim algorithm, see Numerical Trim Algorithm.

To specify the disk stick specification:

1. Select the **Virtual Disks** > **Virtual Disk 1** > **Disk Stick Specification** node.



2. In the *Disk Stick Specification - Properties* window, set the following properties:

Collective Pitch	The collective part of the pitch angle θ_0 (see Eqn. (4027)). The angle that you specify here is the same for all blades. When either **Trim Thrust Only** or **Trim Thrust and Moments** are selected for the *Virtual Disk Trim Option*, this property acts as the initial value for the iterative trim algorithm.
Cyclic Pitch: Cosine Component	The longitudinal cyclic part of the pitch angle θ_{1s} (see Eqn. (4027)). When **Trim Thrust and Moments** are selected for the *Virtual Disk Trim Option*, this property acts as the initial value for the iterative trim algorithm.
Cyclic Pitch: Sine Component	The lateral cyclic part of the pitch angle θ_{1c} (see Eqn. (4027)). When **Trim Thrust and Moments** are selected for the *Virtual Disk Trim Option*, this property acts as the initial value for the iterative trim algorithm.
Virtual Disk Trim Option	You specify the trimming type for the virtual disk trimming by selecting this option. For more background information, see Numerical Trim Algorithm. You can follow the iterative development of the pitch angle components by using the Virtual Disk Angle report. For more information, see Reporting Virtual Disk Results.

	No Trimming	No trimming is performed. All parts of the pitch angles are fixed and user-specified.
	Trim Thrust Only	The iterative trim algorithm adjusts the collective pitch angle only to achieve the desired thrust value. The three sub-nodes **Trim Convergence Control, Target Thrust**, and **Target Pitch and Roll Moment** are added to the simulation tree.
	Trim Thrust and Moments	The iterative trim algorithm adjusts all parts of the pitch angle (collective, longitudinal cyclic, and lateral cyclic pitch angle) to achieve the desired thrust and moment values. The three sub-nodes **Trim Convergence Control, Target Thrust**, and **Target Pitch and Roll Moment** are added to the simulation tree.

Setting the Disk Flap Specification

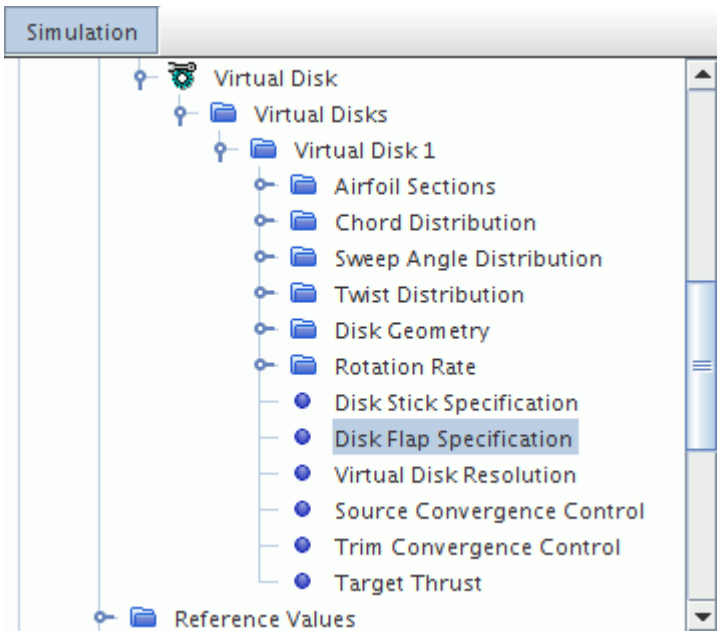
The phenomenon of rotor blade flapping is a result of the dynamic equilibrium between the aerodynamic lift force and the centrifugal force about the flap hinge acting on the blade. A rotor blade with a flap hinge flaps upwards due to the aerodynamic lift force. The centrifugal force acts as a restraining force directed outwards and keeps the blade in the rotor plane. As a result of these two forces, the blade assumes a conical path instead of remaining in the plane perpendicular to the mast.

In hover flight mode, the lift distribution is symmetrical with respect to the blade azimuth angle. This symmetrical distribution results in a constant flapping angle that is independent of azimuth. This constant angle is known as the coning angle. In forward flight mode, the lift distribution is asymmetrical due to the unequal amount of lift on the advancing and the retreating side of the rotor. The flapping response of the blades thus depends on the azimuth. On the advancing side, the blade flaps up to reduce the angle of attack and to generate less lift. On the retreating side, the blade flaps down to increase the angle of attack and to generate more lift. In addition, the blades experience asymmetry in velocity due to the constant component of the flap angle in the front and the back of the rotor. The blade responds to this asymmetry by flapping upwards as it passes the front part and by flapping downwards as it passes the back part.

STAR-CCM+ provides the functionality to specify the coning angle, the cyclic flapping components, and the flap hinge eccentricity. This functionality caters for a more realistic description of the rotor system being modeled.

To set the disk flap specification:

1. Select the **Virtual Disks** > **Virtual Disk 1** > **Disk Flap Specification** node and set the Disk Flap Specification Properties.



Disk Flap Specification Properties

Coning Angle	The angle that is formed between the span wise axis of a helicopter rotor blade, when the rotor system is producing lift, and a plane at right angles to the rotor mast. The coning angle is caused by the relationship between the centrifugal force acting on the rotor blade and the lift that is produced by the blades. This angle corresponds to δ_0 in Eqn. (4023).
Cyclic Flap: Cosine Component	This angle corresponds to δ_{1c} in Eqn. (4023).
Cyclic Flap: Sine Component	This angle corresponds to δ_{1s} in Eqn. (4023).
Flap Hinge Eccentricity	The distance between the horizontal hinge of a helicopter rotor system (the flapping hinge) and the hub. The hinge allows the blades to move up and down in a flapping motion that compensates for asymmetry in the lift force.

Specifying the Virtual Disk Resolution

The cylindrical shape of the virtual disk for the blade element method must be associated with the cells of the underlying volume mesh.

To perform this association, you specify the resolution for a two-dimensional interpolation grid in the radial and the azimuthal direction. To resolve flow effects near the tip efficiently without increasing the number of

elements, you can activate a clustering of interpolation grid lines in the radial direction towards the outer radius of the virtual disk. Each element of this virtual disk interpolation grid is called a *bucket*.

Set the virtual disk resolution such that each bucket is associated with at least one cell of the volume mesh. The virtual disk resolution must not be too fine. The resulting size of a bucket must approximately correspond to the cell size of the underlying volume mesh.

For a volume mesh with trimmed cells, you can approximate the required virtual disk resolution. Calculate the length of the virtual disk in the radial direction ($r_{out}-r_{in}$) and in the azimuthal direction along the inner radius ($2\pi r_{in}$). Divide the lengths by the specified resolution in the respective direction. The resulting value must approximately correspond to the volume mesh cell size in that area.

To specify the virtual disk resolution:

1. Expand the **Virtual Disks** > **Virtual Disk 1** > **Virtual Disk Resolution** node. Select the **Azimuthal Resolution** node and set the *Azimuthal Resolution*.
2. Similarly, select the **Radial Resolution** node and set the *Radial Resolution* and any distribution that you wish to apply. See Radial Resolution Properties.

If you specified a virtual disk resolution that is too high, STAR-CCM+ issues an error message. Reduce the grid resolution until the error message disappears. For more details, see Step 1: Cell marking and the allocation to the blade elements/buckets based on the interpolation grid..

Virtual Disk Resolution Properties Reference

Azimuthal Resolution Properties

Azimuthal Resolution

The resolution of the two-dimensional virtual disk interpolation grid in azimuthal direction.

Radial Resolution Properties

Radial Resolution

The resolution of the two-dimensional virtual disk interpolation grid in radial direction.

Radial Distribution Function

Selects by which method the grid lines of the interpolation grid are clustered towards the outer radius of the virtual disk. Determines how the grid lines are distributed radially.

- **Constant**
  Specifies an equidistant distribution of radial grid lines.
- **Geometric**
  Specifies a geometrical progression for the distribution of the radial grid lines. Distributes the bucket layers using simple geometric relationships. The stretch factor between adjacent bucket layers is always a constant value.
- **Hyperbolic Tan**
  Specifies a hyperbolic function for the distribution of the radial grid lines. Distributes the bucket layers according to a one-parameter hyperbolic tangent stretching function:

$$s(n_i) = 1 + \frac{\tanh[F_s(n_i/N - 1)]}{\tanh(F_s)}$$

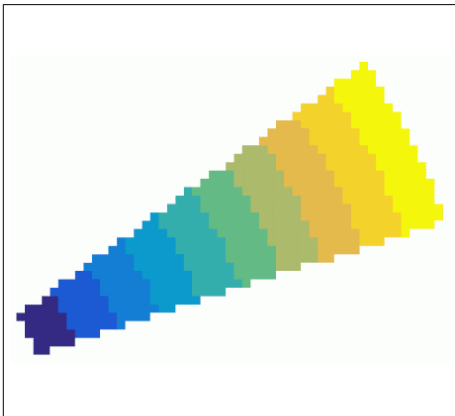
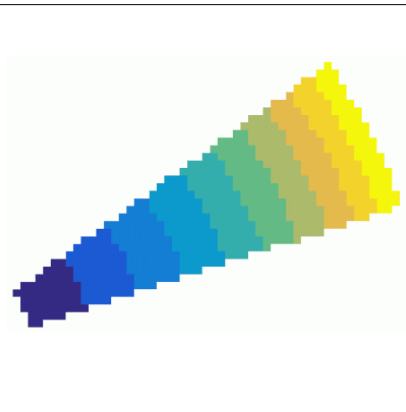
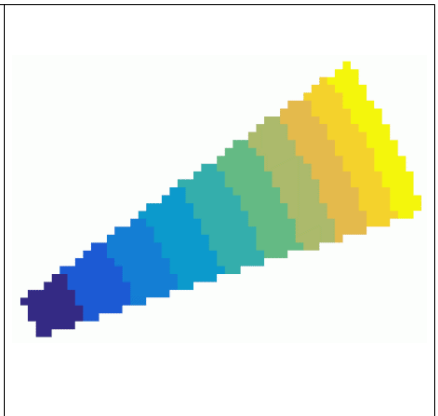
where N is the total number of layers, n_i is the current node, F_s is the overall stretch factor, and $s(n_i)$ is the distribution value for node n_i. Note that $0 \leq n_i \leq N$, so that for 5 layers, n_i takes the values 0, 1, 2, 3, 4, 5.

Radial Stretch Mode

Selects how the factor is defined that determines the density of the radial grid line distribution for the non-constant radial distribution functions.

- **Stretch Factor**
  Defined as the ratio of spacing at bucket layer n and layer n+1.
- **Ratio Factor**
  Ratio of spacing between last layer and the first layer of buckets.

Example: Interpolation Grid Sector with Different Radial Distribution Functions

		
Constant spacing of bucket layers.	Geometric progression with a stretch factor of 0.9.	Hyperbolic Tan with a stretch factor of 0.9.

Tip-Loss Correction for the Blade Element Method

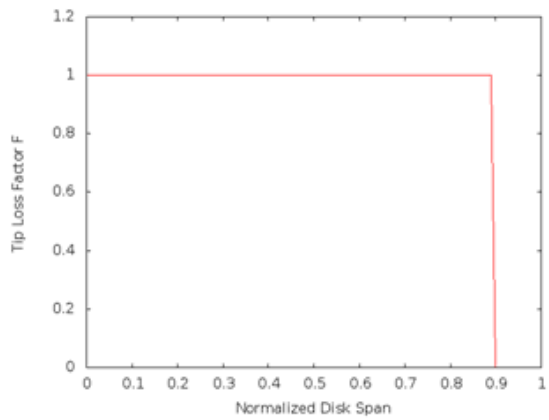
A trailing vortex forms at the tip of each rotor blade. This trailing vortex leads to a higher local inflow near the tip, which effectively reduces the lift generation capability of the blade in this region. This phenomenon is known as tip loss.

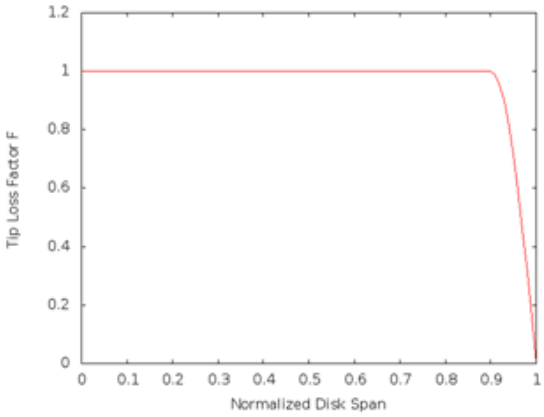
Tip-Loss Correction Properties

Tip-Loss Correction

The blade element method accounts for this effect by including the tip-loss correction. The tip-loss correction is a factor F within a range of 0 to 1 that is multiplied with the lift force to yield a corrected lift force towards the tip of the blade as given by Eqn. (4030). STAR-CCM+ provides several methods to specify the tip-loss correction factor. For all methods, this factor is specified as a function of the

dimensionless normalized disk span $r' = r/R$. The factor ranges from 0 to 1 where r is the radial position and R is the outer radius of the virtual disk (corresponding to the blade tip radius).

Method	Sub-node Properties
Constant Provides backward compatibility. On import of simulation files created prior to STAR-CCM + 12.04 and with tip correction active (0 < *Tip Correction* < 1), this method is set automatically. The tip-loss factor has the shape of a step function with a value of 1 until the radial starting point, and a value of 0 thereafter. 	**Constant** *Radial Start Point* defines the radial position on the virtual disk starting from which the tip-loss correction is applied. *Radial Start Point* takes the value of previously set *Tip Correction* from the restored simulation file. For example, a value of 0.9 for the *Radial Start Point* means that only 90% of the blade generates lift and therefore contributes to the generated thrust. The remaining 10% do not contribute to the lift, but only to drag. In other words, tip-loss correction takes effect at a normalized disk span of 0.9. At this point, the lift force is cut off.
Cosine From the radial starting point onwards, sets the tip-loss factor in the form of a half-cosine function: $$F = \begin{cases} 1 & , \ 0 \le r' < r'_{\text{start}} \\ \cos\left[\frac{\pi}{2}\left(\frac{r' - r'_{\text{start}}}{R - r'_{\text{start}}}\right)\right] & , \ r'_{\text{start}} \le r' \le R \end{cases} \quad (377)$$ This function provides a gradual decrease in lift force contribution to thrust from the radial starting point onwards to the outer radius of the virtual disk.	**Cosine** The *Radial Start Point* defines the radial position on the virtual disk from which the tip-loss correction is applied. Corresponds to r'_{start} in Eqn. (377).

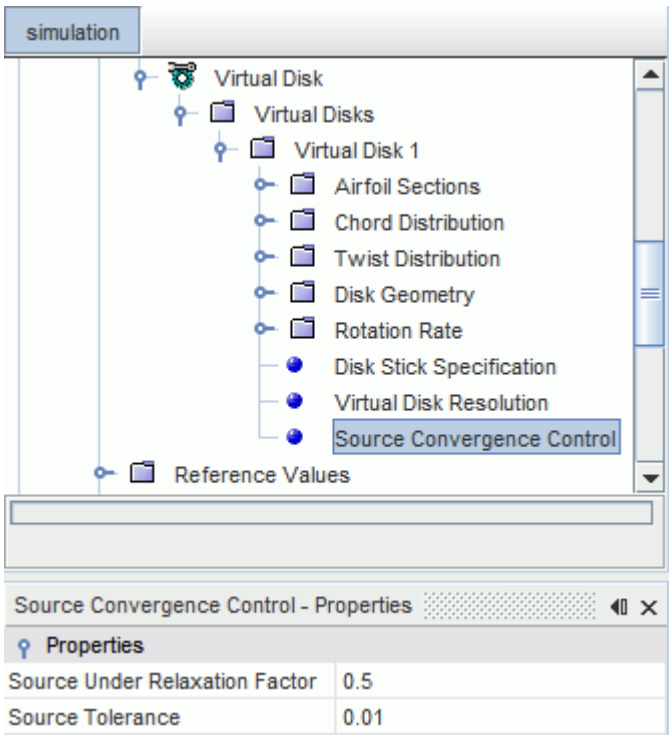
Method	Sub-node Properties
	

User-Defined
Sets the tip-loss factor in the form of user-defined field functions. Both the radial starting point and the user-defined tip-loss correction can vary in space.
	User-Defined
	The *Radial Start Point Function* defines the radial position on the virtual disk from which the tip-loss correction is applied. You can set this starting point using your own user-defined field function. In the field function definition, you can specify the radial starting point as a constant value or as a function of angular position on the virtual disk. To create a dependence on angular position, use the built-in field function **Cell Azimuth of [Virtual Disk]**.
	For example:
	`abs(0.15*sin(${VirtualDiskCellAzimuth1}))+0.8`
	Correction Function lets you specify a tip-loss correction function through a user-defined field function.
	For example:
	`cos(${pi}/2 * (${VirtualDiskNormalizedSpan1} - ${azimuthFunction})/(1.0 - ${azimuthFunction}))`
	In this example, the function refers to the field function `${azimuthFunction}`, which is the user-defined field function for *Radial Start Point Function*.

Setting the Source Convergence Control

To set the source convergence control:

1. Select the **Virtual Disks** > **Virtual Disk 1** > **Source Convergence Control** node.



2. In the *Source Convergence Control - Properties* window, set the following properties:

Source Under Relaxation Factor	The under-relaxation factor for the source term.
Source Tolerance	The convergence tolerance for the source term.

Setting the Trim Convergence Control

To enhance convergence of the numerical trimming algorithm, you set the convergence tolerance and the under-relaxation factor.

To set the trim convergence control:

1. Select the **Virtual Disks** > **[Virtual Disk]** > **Trim Convergence Control** node.
2. In the *Trim Convergence Control - Properties* window, set the following properties:

Trim Under Relaxation Factor	The trim under-relaxation factor for the trim algorithm. The default is 0.5. For more information, see Overview of the Source Term Update.
Trim Tolerance	The convergence tolerance for the trim algorithm. The default is 0.01. For more information, see Overview of the Source Term Update.

Setting the Target Values for Trimming

The trim algorithm is an iterative procedure that adjusts the various parts of the pitch angle to achieve the user-specified target values of thrust and moments. Here, you specify these target values.

To specify the target values for trimming:

1. Select the **Virtual Disks** > **[Virtual Disk]** > **Target Thrust** node.
2. In the *Target Thrust - Properties* window, set the following properties:

Target Rotor Thrust	The target value for the rotor thrust.

3. Select the **Target Pitch and Roll Moment** node.
4. In the *Target Pitch and Roll Moment - Properties* window, set the following properties:

Target Rotor Pitch Moment	The target value for the rotor pitch moment. If you selected the **Trim Thrust Only** option, this property has no influence on the trim algorithm. You can leave the default value.
Target Rotor Roll Moment	The target value for the rotor roll moment. If you selected the **Trim Thrust Only** option, this property has no influence on the trim algorithm. You can leave the default value.

1D Momentum Method

As part of the virtual disk model, the 1D Momentum method can be used to model the effects of a wind turbine.

Wind turbines convert the wind energy into mechanical energy of rotation, which is then used to generate power. Among the wide variety of types of wind turbine, the 1D Momentum method is restricted to the so-called Horizontal Axis Wind Turbine (HAWT). HAWT uses aerodynamic lift to drive a rotor mounted on a horizontal axis. The rotational energy drives an electrical generator.

The 1D Momentum method is based upon one-dimensional momentum theory with wake rotation. The method imitates the effect of the presence of a wind turbine by adding a source term to the momentum equations. The source term accounts for both axial and tangential effects. For more information on the formulation, see 1D Momentum Method.

The method requires the performance curves of a wind turbine, that is, the thrust coefficient and the power as functions of wind speed. An inflow velocity plane is used to compute the volume-averaged values for the velocity and the density of air. The inflow velocity plane is placed upstream of the wind turbine. This upstream velocity is used to extract the thrust coefficient and the power values from the performance curves and to calculate the source term.

The method can be used to study wake effects of a single turbine or of a complete wind park. A typical wind park design objective is to avoid placing a wind turbine in the wake of another. The 1D Momentum method is particularly useful for examining various wind park layouts in to order maximize energy production.

Contents:

Setting Up and Using the 1D Momentum Method
Creating a Virtual Disk for the 1D Momentum Method
Setting the Power Curve for the 1D Momentum Method
Wind Turbine Cut-in and Cut-out Speed
Specifying the Rotation Rate for the 1D Momentum Method
Visualizing the Virtual Disk and the Inflow Velocity Plane

Setting Up and Using the 1D Momentum Method

Similar to the other virtual disk methods available, the setup of the virtual disk for the 1D momentum method does not require the creation of a separate region to which the method is then applied. Instead, you create the shape of the virtual disk by specifying the relevant parameters such as radius, thickness, and orientation to an existing mesh.

To verify the dimensions and orientation of the virtual disk, you can visualize it by using virtual disk-specific field functions.

To set up and run the 1D momentum method:

1. Select the virtual disk model.

   See Selecting the Virtual Disk Model

2. Create a virtual disk.

   See Creating_a_Virtual_Disk_for_the_1D_Momentum_Method

3. Visualize the virtual disk.

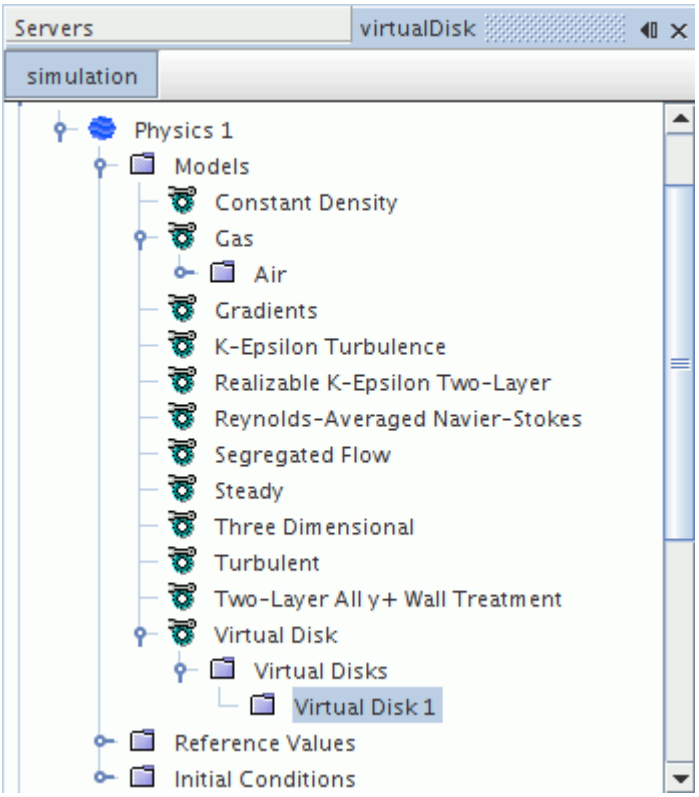
   See Visualizing the Virtual Disk and the Inflow Velocity Plane

Creating a Virtual Disk for the 1D Momentum Method

To create a virtual disk:

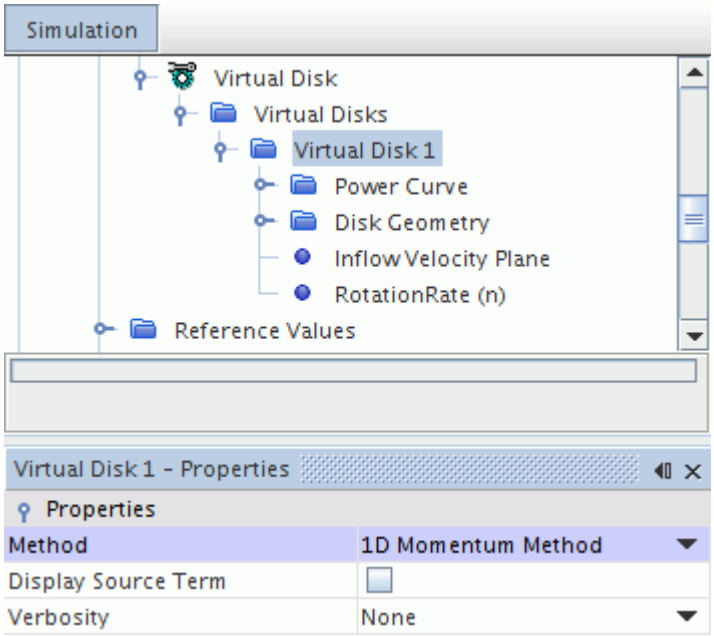
1. Right-click the **Virtual Disk** > **Virtual Disks** node and select **New**.

   A new child node called **Virtual Disk 1** is added to the simulation tree.



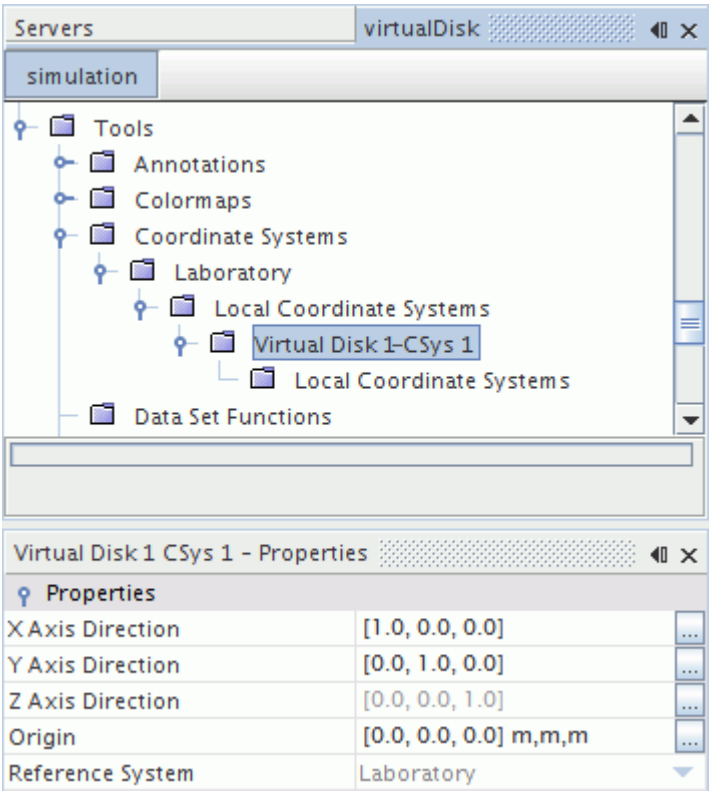
2. Select the **Virtual Disk 1** node and, in the *Properties* window, select the **1D Momentum Method** from the drop-down list.

   The additional **Power Curve, Disk Geometry, Inflow Velocity Plane,** and **RotationRate(n)** nodes are added to the simulation tree.



A new local coordinate system for the virtual disk is created automatically. You can find it under the **Tools** > **Coordinate Systems** > **Laboratory** > **Local Coordinate Systems** > **Virtual Disk 1-CSys 1** node. This coordinate system is oriented such that the positive z-axis is always normal to the plane of the disk.

Do not modify the local coordinate system of the disk here in the **Local Coordinate Systems** node. Modify the virtual disk origin and orientation only under the **Virtual Disk** node in the relevant physics continuum.



For more information, see Specifying the Disk Orientation.

3. If you want to output additional information on the 1D momentum method during the simulation, set the Virtual Disk Properties.
4. Define the Virtual Disk by configuring the following nodes:
   - **Power Curve**

     Set the wind turbine power curve as a function of wind speed.

     See Setting the Power Curve for the 1D Momentum Method
   - **Disk Geometry**

     Define the disk geometry in terms of geometrical dimensions, location, and orientation.

     See Specifying the Disk Geometry for the Body Force Propeller and the 1D Momentum Methods
   - **Inflow Velocity Plane**

     The inflow velocity plane is located upstream of the virtual disk. For the body force propeller method, it is always defined in the positive direction of the disk normal. For the 1D momentum method, it is always defined in the negative direction of the disk normal.

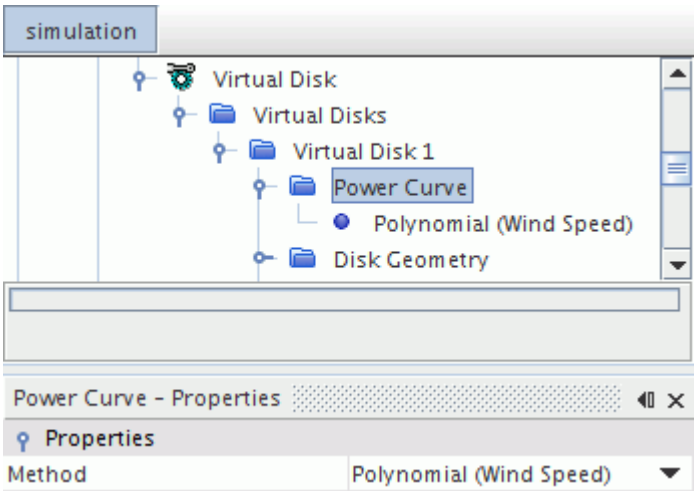
     See Specifying the Inflow Velocity Plane
   - **RotationRate(n)**

     See Specifying the Rotation Rate for the 1D Momentum Method

Setting the Power Curve for the 1D Momentum Method

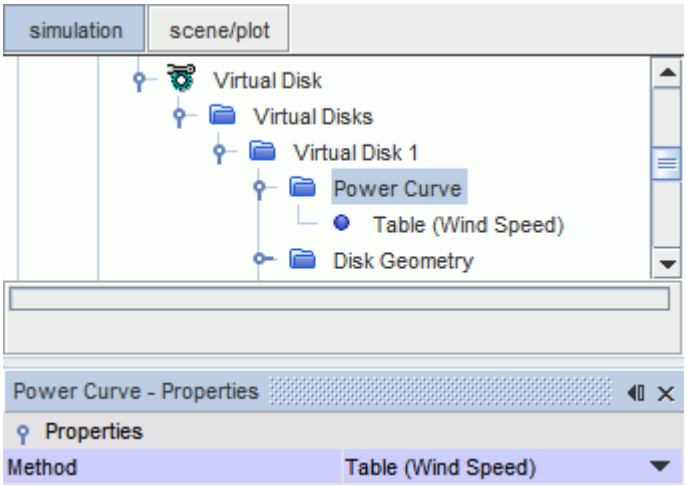
To specify the propeller performance curve:

1. Select the **Virtual Disk 1** > **Power Curve** node.
2. In the *Properties* window, select the *Method* that you would like to use to specify the propeller power curve. There are two options available:
   - **Polynomial (Wind Speed)**



1. Select the **Polynomial (Wind Speed)** node and then click the ... (**custom editor**) next to the *Power = f(Wind Speed) and Ct = f(Wind Speed)*.
2. In the *Polynomial(Windspeed)* dialog, enter the property values of the respective polynomial (see Setting Values Using A Polynomial).

   - **Table (Wind Speed)**



1. Import the appropriate propeller performance curve table into the simulation see ().

2. Select the **Table (Wind Speed)** node. In the *Table (Wind Speed) - Properties* window, set the following properties:

Table	Select the power performance curve table that you imported into your simulation.
Table: Wind Speed	Select the column that contains the wind speed in m/s for the selected wind turbine power table.
Table: Power	Select the column that contains the power in Watts for the selected wind turbine power table.
Table: Ct	Select the column that contains the thrust coefficient (dimensionless) for the selected wind turbine power table.

Wind Turbine Cut-in and Cut-out Speed

At very low wind speeds, the wind exerts insufficient torque on the wind turbine blades to make them rotate. With increasing wind speed, the turbine begins to rotate and to generate power. The speed at which the turbine first starts to rotate and to generate power is called the cut-in speed.

If the wind speed increases beyond a critical velocity, the forces acting on the turbine blades get too large and the turbine can be damaged. Therefore, when this happens, a braking system enforces the wind turbine to stand still. The speed limit where this braking happens is called the cut-out speed.

The 1D momentum method takes the effects of the cut-in speed and the cut-out speed into account in the following conditions:

- The averaged air velocity on the inflow velocity plane is lower than the first wind speed entry of the power curve table.
- The averaged air velocity on the inflow velocity plane is greater than the last wind speed entry of the power curve table.

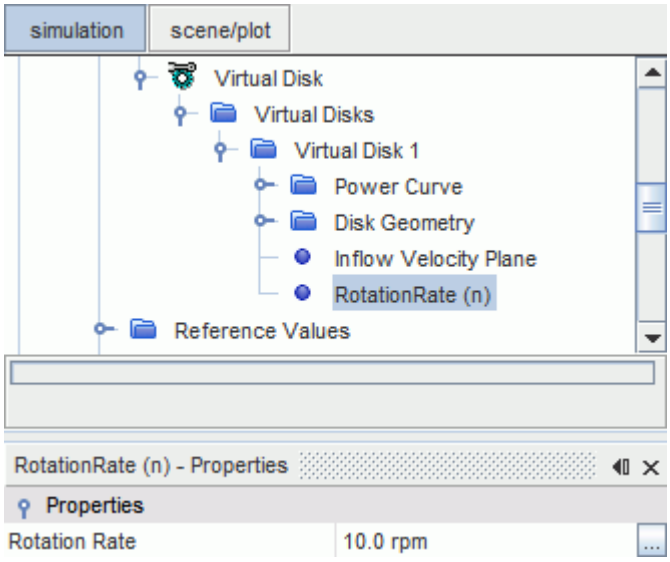
When either of the two cases occur, the 1D momentum source term is taken out of the computation of the momentum equations. This removal of the source term mimics the effect of the cut-in and the cut-out speed.

Specifying the Rotation Rate for the 1D Momentum Method

The rotation rate defines the operation point of the wind turbine.

To specify the rotation rate:

1. Select the **Virtual Disk 1** > **RotationRate (n)** node.
2. In the *RotationRate (n) - Properties* window, set the *Rotation Rate*.



Visualizing the Virtual Disk and the Inflow Velocity Plane

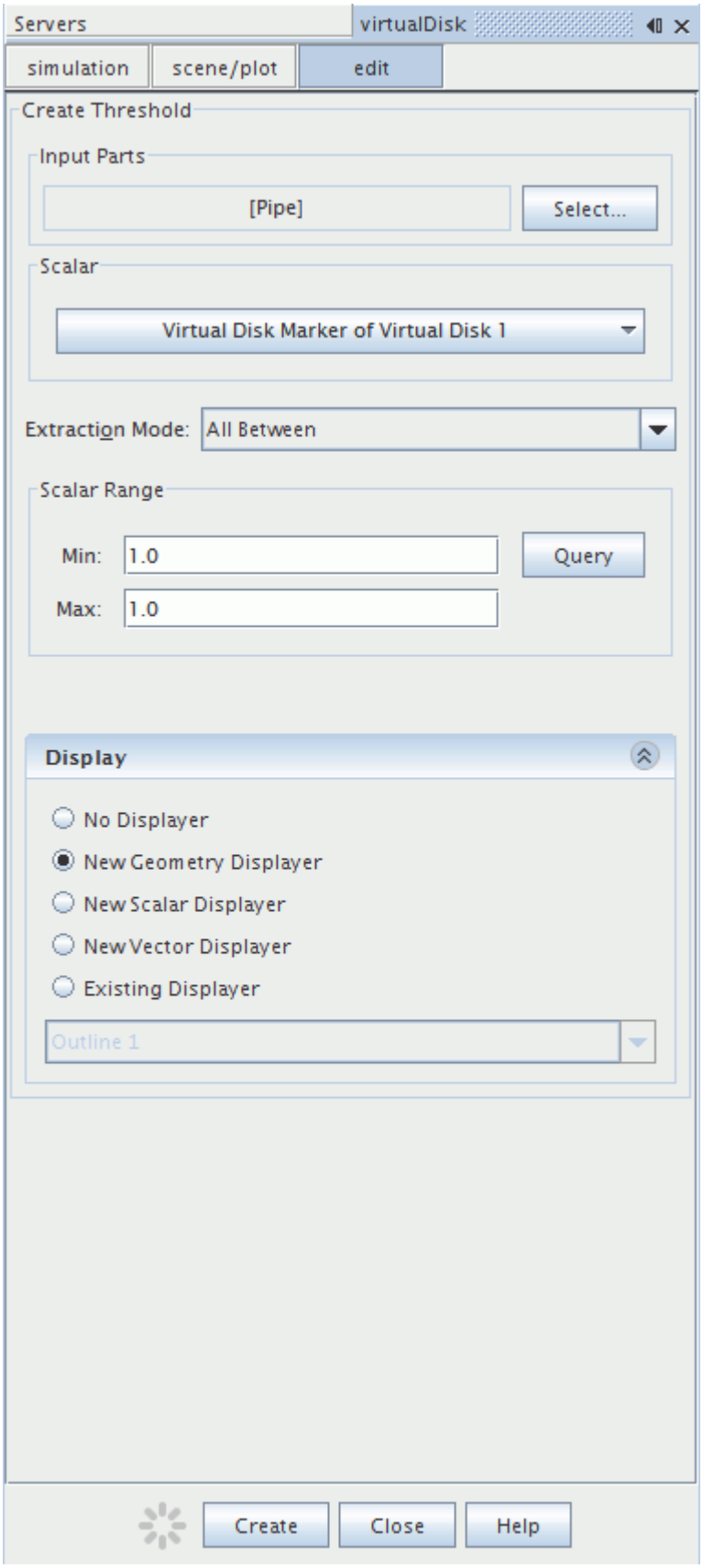
You can visualize the previously defined virtual disk by use of a field function that become available when the virtual disk model is activated: *Indicator of the virtual disk*. If you activated the Body Force Propeller method, you can visualize the inflow velocity plane in a similar fashion using the field function: *Inflow plane marker of virtual disk*. Each field function is then input into a derived part of the type threshold.

The following procedure is for visualizing the virtual disk, but the procedure for visualizing the inflow velocity plane is identical.

To visualize the virtual disk:

1. Create a scene of any type.

   For more details, see Working with Scenes.
2. To create a new derived part of the type threshold, right-click the **Derived Parts** node and select **New Part** > **Threshold...**
3. In the *Create Threshold* dialog:
   1. In the *Input Parts* box, select the region, in which the virtual disk resides.
   2. In the *Scalar* box, select **Indicator of Virtual Disk 1**.
   3. Set the *Extraction Mode* to **All Between**.
   4. In the *Scalar Range* box, set the *Min* and *Max* values both to `1`.
   5. In the *Display* box, select any display option.
   6. Click **Create** and **Close**.

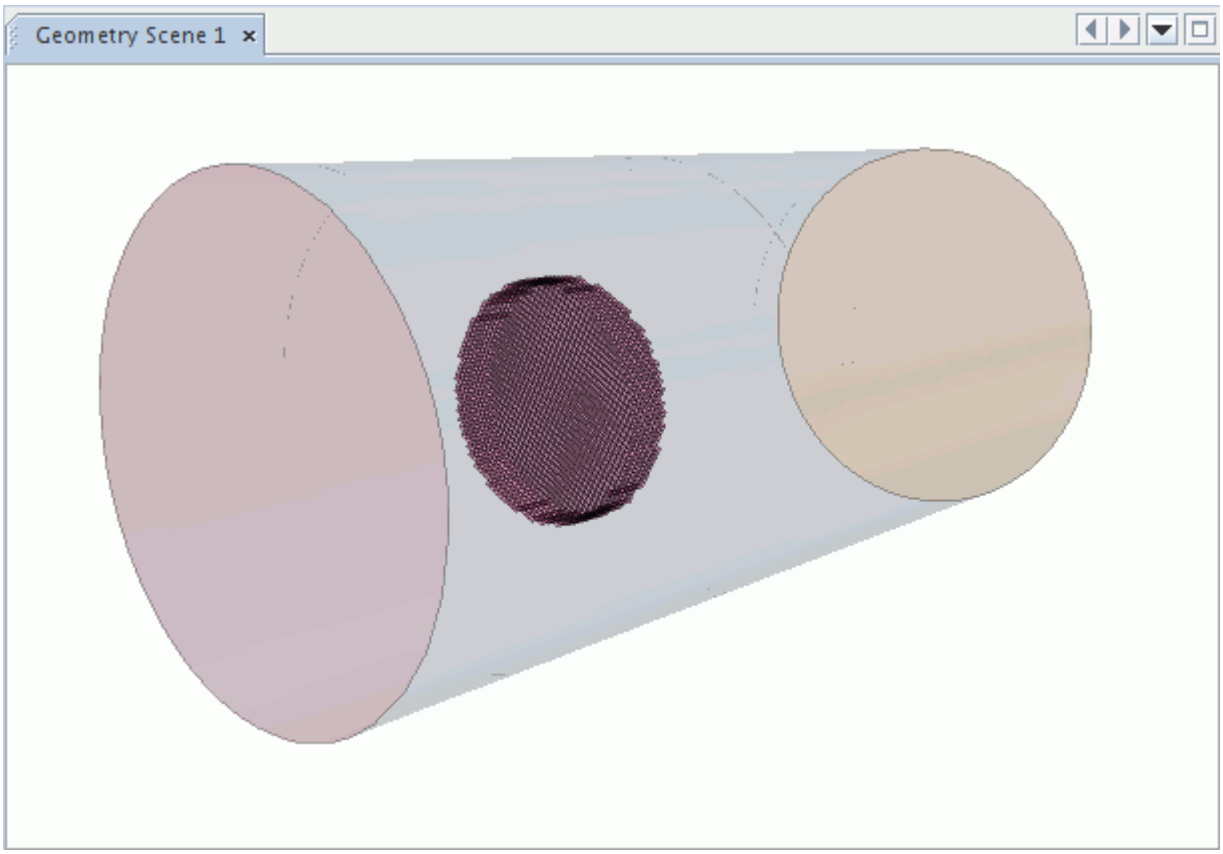


4. Add the newly created derived threshold part to the currently open scene in the *Graphics* window.

5. To view the cells that are part of the virtual disk, you can either initialize the solution manually or run the simulation.

For more information, see Initializing Manually and see Running the Simulation.

An example of a geometry scene showing the virtual disk with mesh lines displayed on it is shown below.



User Defined Method

The user defined method of the virtual disk model is designed to model the action of a rotating device by specifying a user-defined volume force on a three-dimensional cylindrical mesh.

The method uses the core component of the virtual disk framework, that is, the virtual disk geometry to specify the size, the location, and the orientation of the rotating device. For this disk geometry, you specify a resolution for the three-dimensional cylindrical mesh. With this resolution, the three-dimensional virtual disk mesh is subdivided into mesh elements or "buckets". The points on this virtual disk mesh correspond to the cell centers of the elements on the cylindrical mesh. An inflow velocity plane samples velocity and density information from the flow field to make it available on the virtual disk mesh. After preparing the virtual disk mesh, you specify a volumetric source term on the elements of that mesh. To specify the source term, several methods are available including user coding.

Contents:

Setting up the User Defined Method
User Defined Method Properties
Disk Resolution
Inflow Velocity Plane

Setting up the User Defined Method

You set up the user defined method similar to the other methods of the virtual disk model in that you define the geometry and its orientation. Additionally, the user defined method requires that you specify a virtual disk resolution and the location of the inflow velocity plane with respect to the location of the virtual disk. You must specify a user-defined volume source to model a rotating device. For this specification, different methods are available such as tables, field functions, or user coding.

To set up the user defined method:

1. Select the virtual disk model.
2. Create a virtual disk for the user defined method:
   a) Right-click **Continua** > **[Physics Continuum]** > **Models** > **Virtual Disk** > **Virtual Disks**.
   b) Select the **Virtual Disks** > **Virtual Disk 1** node and set *Method* to **User Defined Method**.
3. Configure the virtual disk for the user defined method by editing the following nodes:
   - Disk Geometry
   - Disk Resolution
   - Inflow Velocity Plane

Select how you provide the volumetric source term.

4. Edit the **User Source Term Profile** node and set *Method* to one of the following:
   - **Composite**
   - **Constant**

- **Field Function**
- **Table (Iteration)**
- **Table (time)**
- **Table (x,y,z)**
- **Table (xyz, time)**
- **User Code**

To define the volumetric source term through field functions or user coding, specific field functions for the user defined method are available. To access the points of the virtual disk mesh in user coding or field functions, you use the field function **Position**. It provides the disk mesh points coordinates with respect to the laboratory coordinate system. You can only use these field functions to define the volumetric source term.

5. If you set *Method* to **User Code**:
   a) Create the source code for the user coding in C or Fortran.
   An example coding for the user defined method is shown below::

```
#include "uclib.h"
#include <iostream>
#include <vector>
#include <math.h>
#include <cstdlib>

void userProp( Real (*result)[3],          // array for new values of the Source Term
               int size,                   // number of points on the disk mesh
               double *rho,                // inflow plane Density
               CoordReal (*inflowVel)[3])  // inflow plane Velocity
{
  /*
   * - source term is proportional to dynamic pressure
   * - only disk normal component is set
   *   using absolute value of disk normal velocity
   *   i.e z-component
   */
  for (int i=0; i!=size; i++)
  {
    result[i][0] = 0.0;
    result[i][1] = 0.0;
    result[i][2] = 0.5 * rho[i] * abs(inflowVel[i][2]) * abs(inflowVel[i][2]);
  }
}

void uclib()
{
  /* Register user functions here */
  ucfunc((void *)userProp, "RegionProfile", "User Propeller Profile");
  ucarg((void *)userProp, "Vertex", "$UserInflowPlaneDensity1", sizeof(double));
  ucarg((void *)userProp, "Vertex", "$$UserInflowPlaneVelocity1", sizeof(CoordReal[3]));
}
```

   This small example demonstrates how to use the specific field functions of the user defined method to access the inflow velocity vector and the density on the virtual disk mesh.
   b) Create a new user library.
   c) Load the user library into STAR-CCM+.
   d) Select the **User Source Term Profile** > **User Code** node and set *Function* to **[Your User Coding]**.
6. (Optional) Visualize the virtual disk and the inflow plane.

User Defined Method Properties

The user defined method of the virtual disk model has specific properties that are only available to this method.

Disk Resolution Properties

Set the virtual disk resolution such that each bucket is associated with at least one cell of the volume mesh. The virtual disk resolution must not be too fine. The resulting size of a bucket must approximately correspond to the cell size of the underlying volume mesh.

Radial Resolution	The resolution of the three-dimensional virtual disk interpolation grid in the radial direction.
Azimuthal Resolution	The resolution of the three-dimensional virtual disk interpolation grid in the azimuthal direction.
Normal Resolution	The resolution of the three-dimensional virtual disk interpolation grid in the normal direction.

For more information, see Disk Resolution.

Inflow Velocity Plane Properties

The inflow velocity plane is located upstream of the virtual disk and is oriented with respect to the direction of the virtual disk normal, that is, with respect to the Z direction of the disk local coordinate system. This plane is discretized using the same resolution as the radial and azimuthal virtual disk resolution.

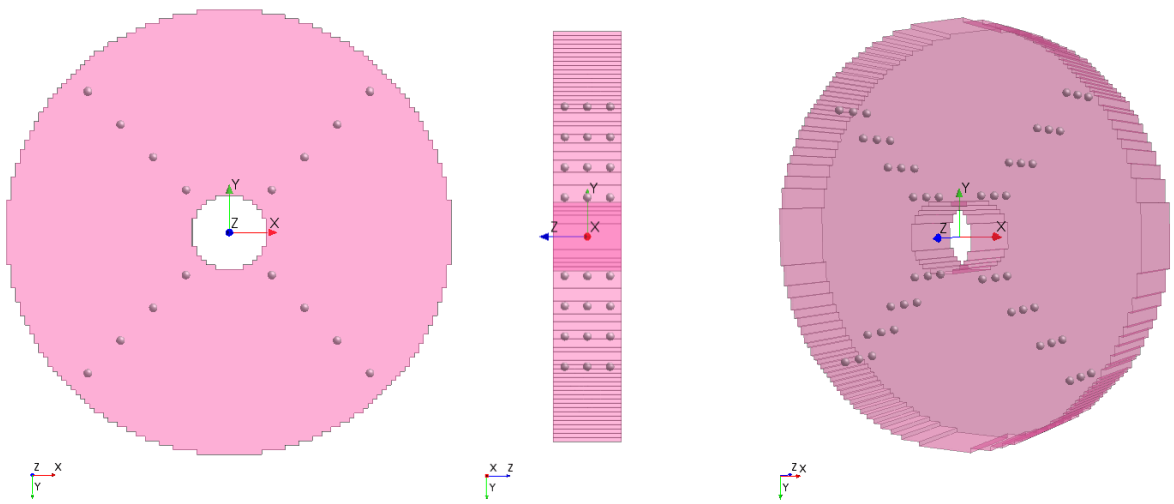
Inner Radius	(Read-only) The inner radius of the velocity plane. Corresponds to the inner radius of the virtual disk.
Outer Radius	(Read-only) The outer radius of the velocity plane. Corresponds to the outer radius of the virtual disk.
Velocity Plane Offset	The offset in the normal (z) direction of the virtual disk with respect to the virtual disk origin.
Local Coordinate System	(Read-only) Reference to the local virtual disk coordinate system.
Radial Resolution	(Read-only) The resolution of the two-dimensional velocity plane interpolation grid in the radial direction. Corresponds to the virtual disk radial resolution.
Azimuthal Resolution	(Read-only) The resolution of the three-dimensional velocity plane interpolation grid in the azimuthal direction. Corresponds to the virtual disk azimuthal resolution.

For more information, see Inflow Velocity Plane.

Disk Resolution

The user defined method is based on the interpolation methodology of the blade element method and extends it to three dimensions. The cylindrical virtual disk is associated with the cells of the underlying volume mesh. The disk resolution that you specify controls this association. Each element or bucket of this cylindrical three-dimensional virtual disk mesh is associated with a number of cells on the underlying volume mesh. A bucket is a cylindrical sector whose radial position, angular extent, and thickness are controlled by the user-specified disk resolution.

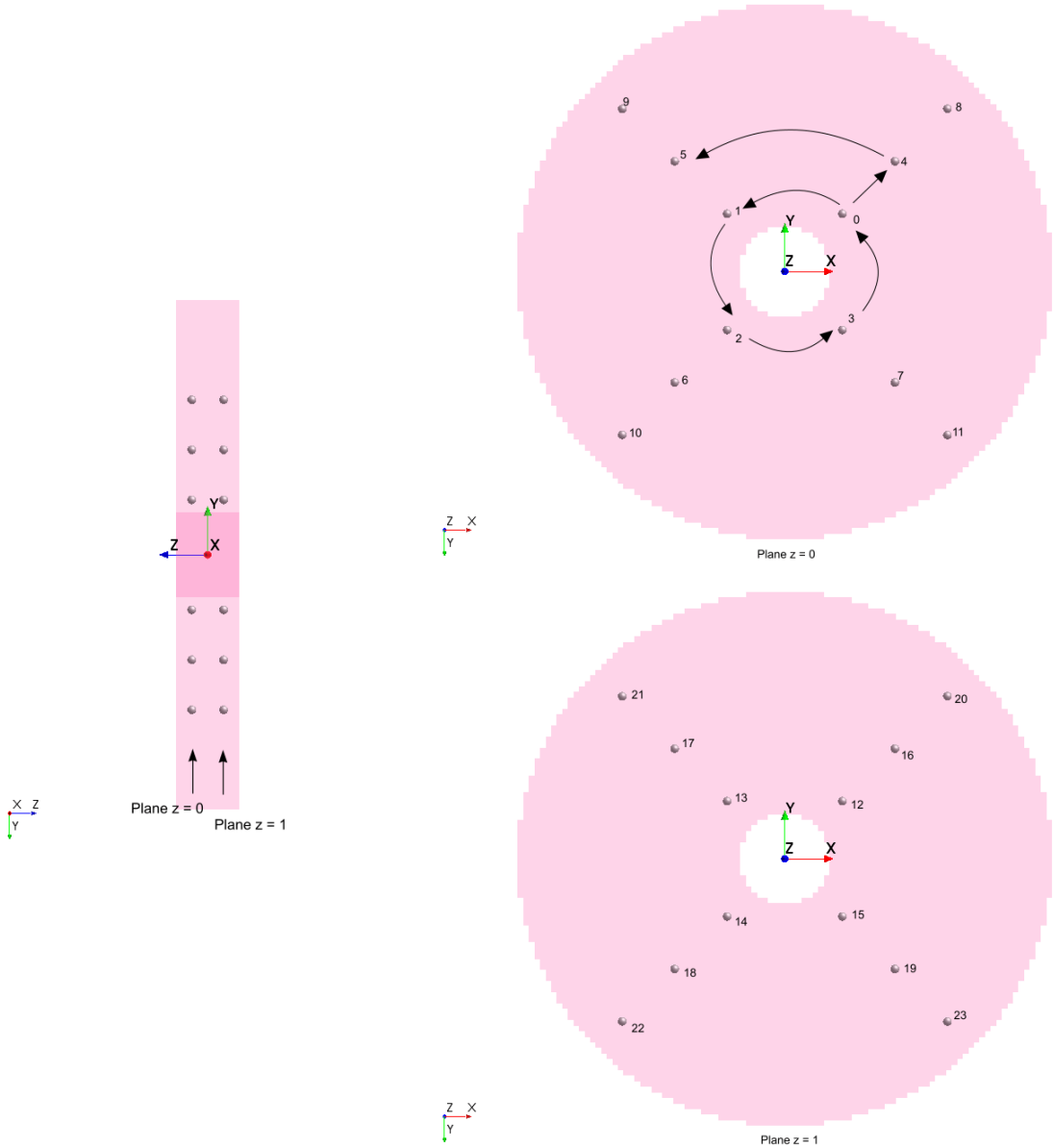
The following figure shows the points of the virtual disk mesh for a (4,4,3) resolution:



Based on the user-specified resolution `(nx, ntheta, nz)`, the total number of buckets is given by `n = nx * netheta * nz`, where `nx` is the number of cells in the radial direction, `ntheta` is the number of cells in the azimuthal direction, and `nz` is the number of cells in the disk normal direction. The buckets are arranged in a one-dimensional ordering starting from $bucket_{(0)}$, $bucket_{(1)}$, $bucket_{(2)}$,..., $bucket_{(n-1)}$. The mapping between the points on the cylindrical disk mesh and the bucket IDs is according to the following loop ordering:

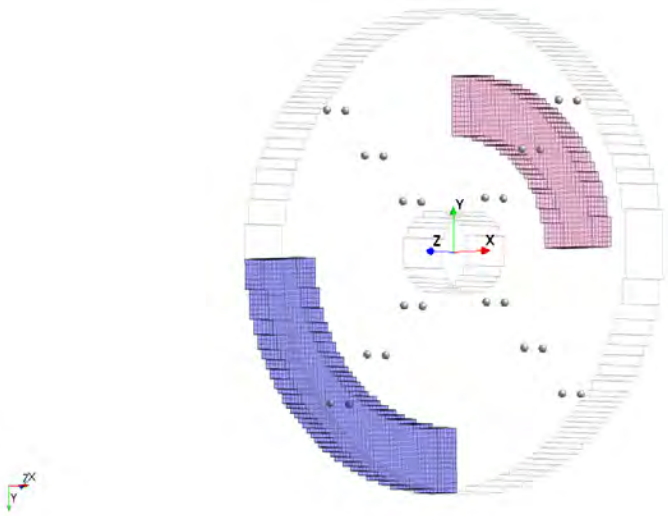
```
for (k=0:nz)
   for (i=0:nr)
      for (j=0:ntheta)
         bucketID = j + i * ntheta + k * ntheta * nr
```

where `k` is the index of the disk mesh points in Z direction, `i` is the index of the disk mesh points in radial directions, and `j` is the index of the disk mesh points in azimuthal direction. Therefore, the bucketID first runs through the azimuthal direction, then the radial direction, and finally in the disk normal direction. The following screenshot shows an example of this mapping procedure:



For this example, the disk resolution is `(3, 4, 2)` and the total number of buckets is `n = 3 * 4 * 2 = 24`.

The next diagram shows the buckets with bucketID = 4 corresponding to `(i = 1, j = 0, k = 0)` and bucketID = 22 corresponding to `(i = 2, j = 2, k = 1)`.



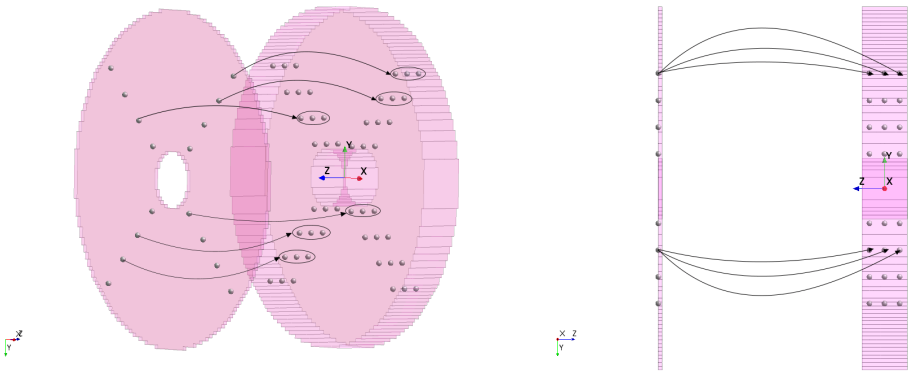
In above diagram, you see the buckets marked as points and the corresponding cells of the volume mesh that the volumetric source term affects. Because the disk mesh is cylindrical, the buckets that are located at the outer radius (blue cells) influence more cells of the volume mesh than the buckets that are located at an inner radius (red cells).

Inflow Velocity Plane

The inflow velocity plane is located upstream of the virtual disk and is oriented with respect to the direction of the virtual disk normal, that is, with respect to the Z direction of the disk local coordinate system. This plane is discretized using the same resolution as the radial and azimuthal virtual disk resolution.

The inner and outer radius of the velocity plane match to the inner and outer virtual disk radii, respectively. Based on the user-specified radial and azimuthal disk resolution, the velocity plane is resolved by a two-dimensional polar mesh that is divided into buckets similar to the virtual disk. Each bucket is associated with a number of cells on the volume mesh. The velocity and density of the flow field is volume-averaged over the bucket. The averaged velocity vector is then transformed into the local coordinate system of the virtual disk. The resulting volume-averaged velocity and density of a velocity plane bucket are interpolated to the corresponding bucket on the virtual disk mesh.

The following diagram demonstrates the interpolation procedure:



A bucket on the inflow plane with a particular (r, θ) coordinate is associated with the same (r, θ) coordinate on the virtual disk mesh. Thus, all buckets with the same (r, θ) coordinates, but varying z coordinates have the same density and velocity values. These values are available as field functions **User Defined Inflow Plane Density of [Virtual Disk]** and **User Defined Inflow Plane Velocity of [Virtual Disk]**. You cannot visualize these field functions on the underlying volume mesh. You use these field functions to access the flow field information to compute the volumetric source term using one of the methods of the **User Source Term Profile**.

Reporting Virtual Disk Results

The following reports are available when the virtual disk model is activated:

- Virtual disk force report
- Virtual disk moment report
- Virtual disk angle report
- Virtual disk rotation rate report
- Virtual disk advance ratio report

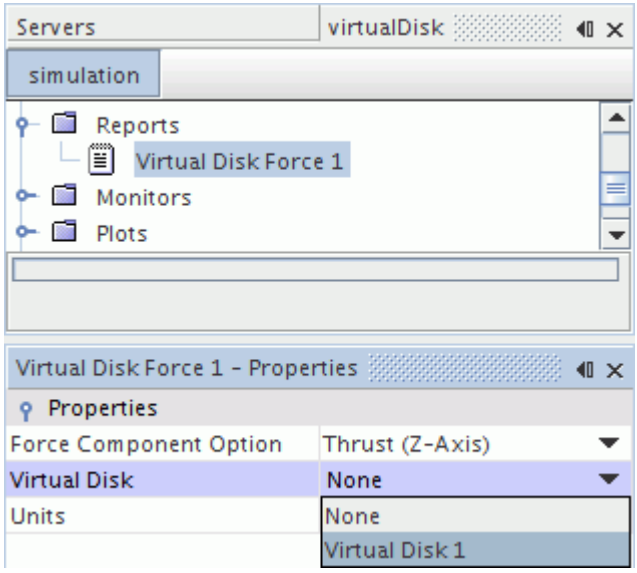
Some virtual disk reports are not available for all methods. The availability of the reports is shown below, where (x) indicates that the report is available and reports a valid value. (-) indicates that the report is not applicable for the corresponding method and reports a value of 0:

Virtual Disk Report	Body Force Propeller Method	Blade Element Method	1D-Momentum
Force	x	x	x
Moment	x	x	x
Angle	-	x	-
Rotation Rate	x	-	-
Advance Ratio	x	-	-

The following procedure is for the virtual disk force report. The same procedure is used for all of the virtual disk reports.

To create a virtual disk force report:

1. Right-click the **Reports** node and select **New Report** > **Virtual Disk Force**
2. Select **Reports Virtual Disk Force 1** node



3. In the *Properties* window, set the Virtual Disk Reports properties.
4. To display the result, create a monitor and a plot.

For more information, see Using Report-Based Monitors and see Working with Monitor Plots.

Virtual Disk Properties

Depending on the selection for the virtual disk method, the *Verbosity* outputs different parameters and values.

Display Source Term	Makes the field function **Virtual Disk Source Term of [Virtual Disk]** available (see Virtual Disk Field Functions).	
Verbosity	Activates the output of additional information on the respective method. All values are given at the current iteration.	
	None	No additional information is written to the output window.
Body Force Propeller Method		
	Low	Outputs the value of the velocity at the inflow plane. Can be interpreted as the advance velocity with respect to propeller terminology.
	High	Outputs the following values: • velocity at the inflow plane • selected operation point input option • advance ratio
Blade Element Method		
	Low	Outputs the following values: • rotation rate • user-specified rotation rate • rotation rate ramping status • residuals and convergence status of the source term computation • residuals and convergence status of the trimming algorithm
	High	Outputs the following values: • rotation rate • user-specified rotation rate • rotation rate ramping status • residuals and convergence status of the source term computation • residuals and convergence status of the trimming algorithm • perturbation used to calculate the Jacobian • number of iterations of the Newton-Raphson algorithm with convergence tolerace • under-relaxation factor for the trim angles update

		• disk stick state before and after the update
1-D Momentum Method		
	Low	Outputs the value of the velocity at the inflow plane.
	High	Outputs the following values: • velocity at the inflow plane • power curve limits (cut-in and cut-out velocity)

Disk Geometry Properties

Inner Radius	Specify the inner radius of the disk. This radius corresponds to the hub radius of the propeller or the wind turbine.	
Outer Radius	Specify the outer radius of the disk. This radius corresponds to the tip radius of the propeller or the wind turbine.	
Thickness	Specify the thickness of the propeller or the wind turbine.	
Origin	Specify the coordinates of the origin of the disk. The origin is defined in terms of the coordinates and the coordinate system with respect to which the coordinates are specified. Here, the choice of coordinate system is dictated by the selection you make for the **Orientation Specification** property: For the option **Disk Normal and Coordinate System,**the coordinates of the origin are specified with respect to the coordinate system that is selected in that node. For the option **Pitch Angle and Lateral Angle,**the coordinates of the origin are always with respect to the laboratory coordinate system.	
Local Coordinate System	Reference to the local coordinate system of the disk (read-only).	
Orientation Specification	You specify the disk orientation by selecting this option. The disk orientation defines in which direction the thrust is produced. For more information, Specifying the Disk Orientation .	
	Disk Normal and Coordinate System	Specify the disk orientation in terms of disk normal and coordinate system (default). When you select this option, a **Normal and Coordinate System** sub node appears in the simulation tree.
	Pitch Angle and Lateral Angle	Specify the disk orientation in terms of lateral angle and pitch angle. When you select this option, a **Lateral And Pitch Angle** sub node appears in the simulation tree.

Inflow Specification Properties Reference

Method	Sub-node Properties
Constant Inflow Values You specify the inflow velocity vector and the inflow density directly as single values. The inflow values remain constant throughout the simulation. This setup leads to a fixed advance ratio and corresponding fixed K_T and K_Q. Constant inflow values correspond to a fixed operating point on the user-specified propeller performance curve. This method is not compatible with motion, that is, with dynamic fluid body interaction, moving reference frames, and rigid-body motion.	*Inflow Velocity* Constant inflow velocity vector. *Inflow Density* Constant inflow density. *Coordinate System* The inflow velocity vector is specified with respect to this coordinate system.
Inflow Velocity Boundary The velocity and density values on the selected inflow velocity boundary are area-averaged over the boundary. The area-averaged velocity vector is then projected onto the local coordinate system of the virtual disk.	*Inflow Boundary* Boundary surface over which the boundary face velocities and boundary face densities are area-averaged to obtain a single value inflow velocity vector and an inflow density. The inflow velocity vector is projected onto the local coordinate system of the virtual disk.
Inflow Velocity Plane The inflow velocity plane is a plane that is located upstream of the virtual disk and is always oriented with respect to the direction of the virtual disk normal. The velocity components and the density of the fluid approaching the virtual disk are volume-averaged over this plane. The averaged velocity components are then projected onto the normal plane of the disk to yield one average velocity vector value. This setup entails the propeller-induced flow influencing the velocity on the inflow plane.	*Inflow Plane Radius* Radius of the inflow velocity plane. You are advised to use a value that is approximately 10% larger than the virtual disk radius. *Inflow Plane Offset* Offset in the normal (z) direction of the virtual disk. You specify the offset with respect to the virtual disk origin. For the body force propeller method, you are advised to you use an offset of 10% of the virtual disk diameter. For the 1D momentum method, there is no recommendation for the offset. The offset is in the negative z direction, because generally the thrust of a wind turbine, and therefore the positive z-axis, is directed downstream of the virtual disk.

Method	Sub-node Properties
	Local Coordinate System Reference to the local virtual disk coordinate system (read-only).

Virtual Disk Reports Properties

Virtual Disk Force Properties

Force Component Option	Specifies the force component to report. The force components are reported in the local coordinate system of the virtual disk.	
	Thrust (Z-Axis)	Specifies the force component in the direction of the z-axis. This direction is the direction in which thrust is produced.
	Pitch Force (X-Axis)	Specifies the force component in direction of the x-axis.
	Lateral Force (Y-Axis)	Specifies the force component in direction of the y-axis.
Virtual Disk	Specifies the virtual disk for which to report the force component.	
Units	Select the units of the force component.	

Virtual Disk Force Expert Properties

Force Type Option	Specifies the active or reactive force type to report.	
	Action of the Fluid on the Virtual Disk	The fluid dynamic forces acting on the virtual disk due to the action of the fluid. Usually selected to report the thrust produced by the virtual disk.
	Reaction of the Virtual Disk on the Fluid	The reaction of the virtual disk corresponds to the force added to the momentum equations in order to model the presence of the virtual disk. It is equal in magnitude, but opposite in sign to the *Action of the Fluid on the Virtual Disk*.

Virtual Disk Moment Properties

Moment Component Option	Specifies the moment component to report. The moment components are reported in the local coordinate system of the virtual disk.	
	Shaft Torque (Z-Axis)	Specifies the moment component in direction of the z-axis.

	Roll Moment (X-Axis)	Specifies the moment component in direction of the x-axis.
	Pitch Moment (Y-Axis)	Specifies the moment component in direction of the y-axis.
Virtual Disk	Specifies the virtual disk for which to report the moment component.	
Units	Select the units of the moment component.	

Virtual Disk Moment Expert Properties

Moment Type Option	Specifies the active or reactive moment type to report.	
	Action of the Fluid on the Virtual Disk	The moment acting on the virtual disk due to the action of the fluid.
	Reaction of the Virtual Disk on the Fluid	Is equal in magnitude, but opposite in sign to the *Action of the Fluid on the Virtual Disk*.

Virtual Disk Angle Properties

Angle Option	Specifies which angle to report.	
	Collective Pitch Angle	Specifies the collective pitch angle.
	Longitudinal Cyclic Pitch Angle	Specifies the longitudinal cyclic pitch angle.
	Lateral Cyclic Pitch Angle	Specifies the lateral cyclic pitch angle.
Virtual Disk	Specifies the virtual disk for which to report the angle.	
Units	Select the units of the angle.	

Virtual Disk Rotation Rate Properties

Use this report to obtain information about the rotation rate of the propeller when you select the **Thrust** or **Torque** option for *Rotation Rate*.

Virtual Disk	Select the virtual disk for which you want to report the rotation rate.	
Units	Select the units of the rotation rate.	

Virtual Disk Advance Ratio Properties

Use this report to obtain information about the advance ratio J of the propeller (see Eqn. (4013)).

Virtual Disk	Select the virtual disk for which you want to report the advance ratio.
Units	The advance ratio is dimensionless.

Virtual Disk Field Functions

The following field functions are made available when the virtual disk model is activated.

Inflow Plane Marker of [Virtual Disk]	Cell marker to indicate which volume cell is identified as part of the inflow velocity plane. A value of 1 indicates that the cell is part of the inflow velocity plane. A value of 0 indicates that the cell is not part of the inflow velocity plane. This field function can be used as input to a derived part of the type threshold.
Indicator of [Virtual Disk]	Cell marker to indicate which volume cell is identified as part of the virtual disk. A value of 1 indicates that the cell is part of the virtual disk. A value of 0 indicates that the cell is not part of the virtual disk. This field function can be used as input to a derived part of the type threshold.
Normalized Span of [Virtual Disk]	Normalized radius (values from 0 through 1 where 0 corresponds to the origin of the virtual disk and 1 corresponds to the outer radius of the virtual disk) of cells within the virtual disk with respect to the local coordinate system of the virtual disk.
Normalized Span Star of [Virtual Disk]	Normalized radius $r^*$of the virtual disk as given by Eqn. (4008). The values range from 0 through 1. A value of 0 corresponds to the inner radius (at the hub) and a value of 1 corresponds to the outer radius (at the tip) of the virtual disk. The radius is with respect to the local coordinate system of the virtual disk.
Cell Azimuth of [Virtual Disk]	Gives the angular position of a cell within the virtual disk with respect to the local coordinate system of the virtual disk.
Virtual Disk Source Term of Virtual Disk 1	Shows the distribution of the virtual disk source term per unit volume over the disk. This function is only available when *Display Source Term* is activated on the virtual disk.

User Defined Method Field Functions

These field functions are specific to the user defined method of the virtual disk model. They are made available to access the flow field values to define the volumetric source term through field functions or user coding. You cannot visualize these field functions on the volume mesh cells or on the virtual disk mesh points.

Normalized Thickness of [Virtual Disk]	Normalized thickness with values from 0 through 1 where 0 is at the outer surface of the virtual disk in the negative Z direction and 1 in the positive Z direction.
User Defined Inflow Plane Density of [Virtual Disk]	Volume-averaged density on the inflow plane for each bucket on the inflow plane mesh. For more information, see Inflow Velocity Plane.

User Defined Inflow Plane Velocity of [Virtual Disk]	Volume-averaged velocity on the inflow plane for each bucket on the inflow plane mesh. For more information, see Inflow Velocity Plane.

Modeling Turbulence

Most fluid flows of engineering interest are characterized by irregularly fluctuating flow quantities.

Often these fluctuations are at such small scales and high frequencies that resolving them in time and space comes at excessive computational costs. Instead of solving for the exact governing equations of turbulent flows (Direct Numerical Simulation), it is less expensive to solve for averaged or filtered quantities and approximate the impact of the small fluctuating structures. Turbulence models provide different approaches for modeling these structures.

The turbulence models that are implemented in STAR-CCM+ can be subdivided into two categories:

- Models that provide closure of the Reynolds-Averaged Navier-Stokes (RANS) equations.
- Scale-resolving simulations that solve for large scales of turbulence and model small-scale motions. These are Large Eddy Simulation (LES) and Detached Eddy Simulation (DES).

Most simulations will rely on the first approach. The second approach (LES and DES) is best used after carefully reviewing the applicable literature to gain confidence that the grid resolution requirements can be met and that the computational costs incurred by resolving the small time and length scales are indeed justified.

It is generally recognised that all existing turbulence models are approximate representations of the physical phenomena of turbulence. The degree of approximation in a given model depends on the nature of the flow to which it is applied. The characterization of the circumstances which give rise to 'good' and 'bad' performance must be based mainly on experience.

Contents:

Selecting a Turbulence Modeling Approach
Reynolds-Averaged Navier-Stokes (RANS) Turbulence Models
Scale-Resolving Simulations
Wall Treatment

Selecting a Turbulence Modeling Approach

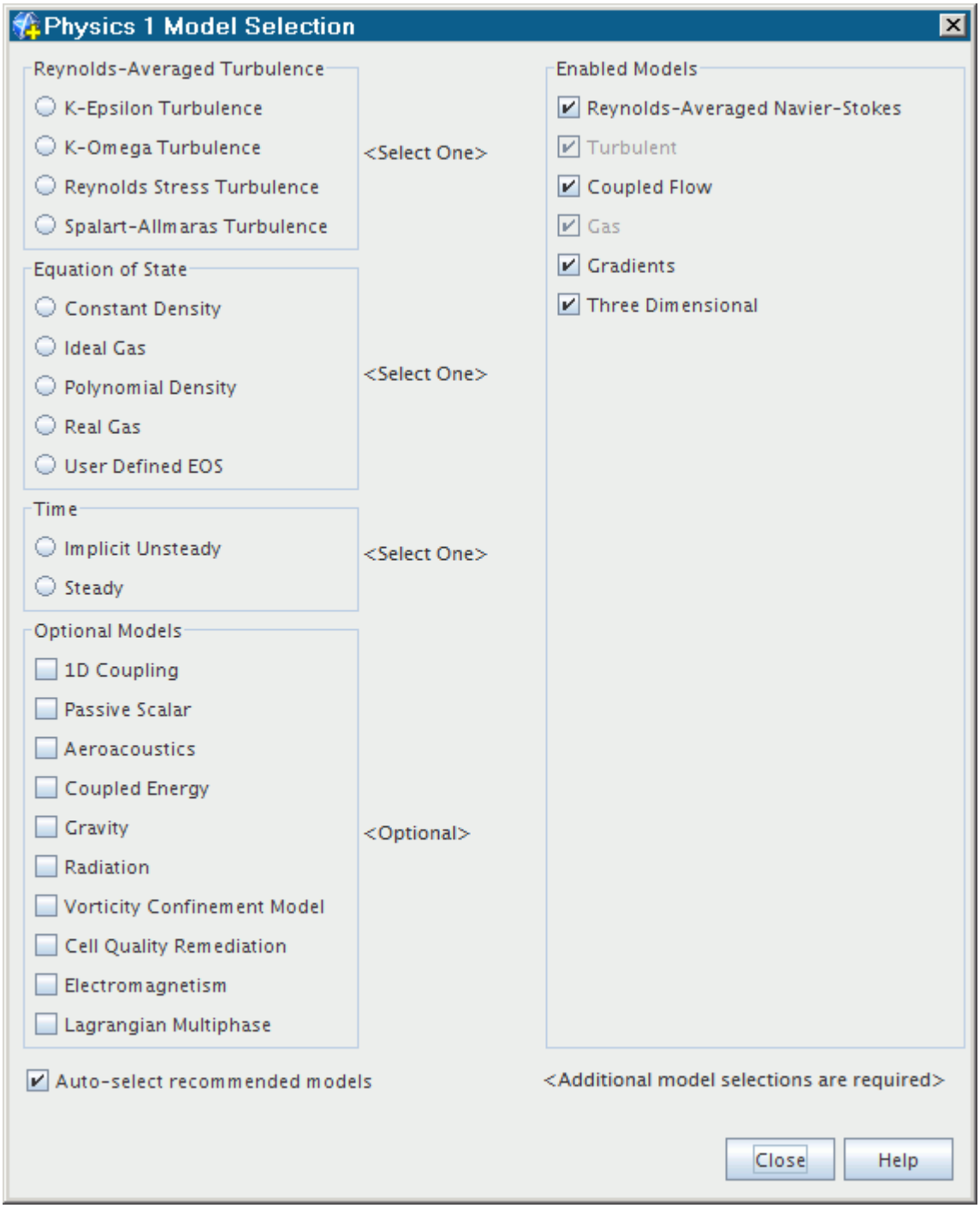
Selecting the turbulence modeling approach is possible with or without auto-select.

Using Auto-Select

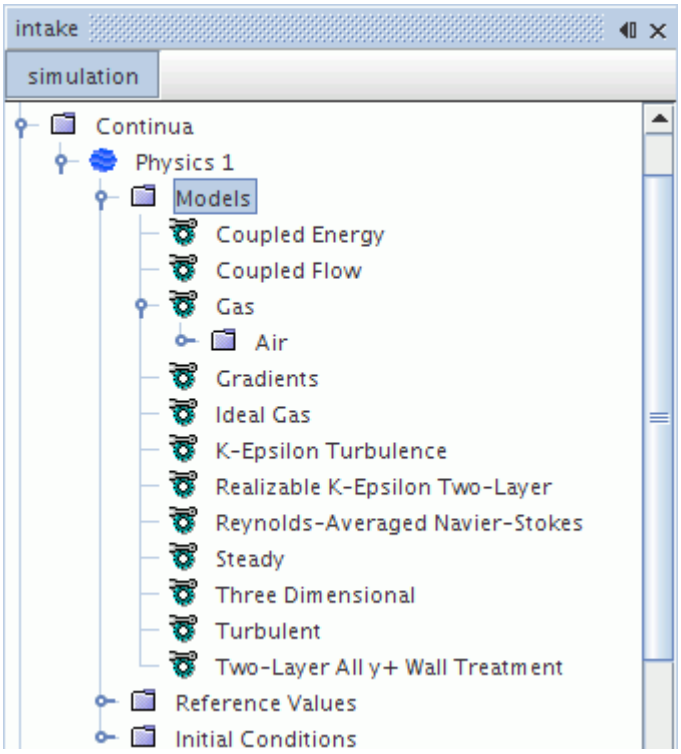
This section describes how to select a turbulence modeling approach using automatic selection of recommended models.

This is the default mode of selection, and is indicated by the **Auto-select recommended physics models** checkbox being ticked.

The selection of the **Turbulent** radio button in the Viscous Regime group box will automatically select turbulence models that provide closure of the Reynolds-Averaged Navier-Stokes (RANS) equations as a turbulence modeling approach. These modeling choices will be displayed in the *Reynolds-Averaged Turbulence* group box as shown below, although the choice of Enabled Models might differ.



While the end result will depend on the choice made above, a complete selection will be made, including the variant of the model and the wall treatment model. The result for the K-Epsilon model is shown below.



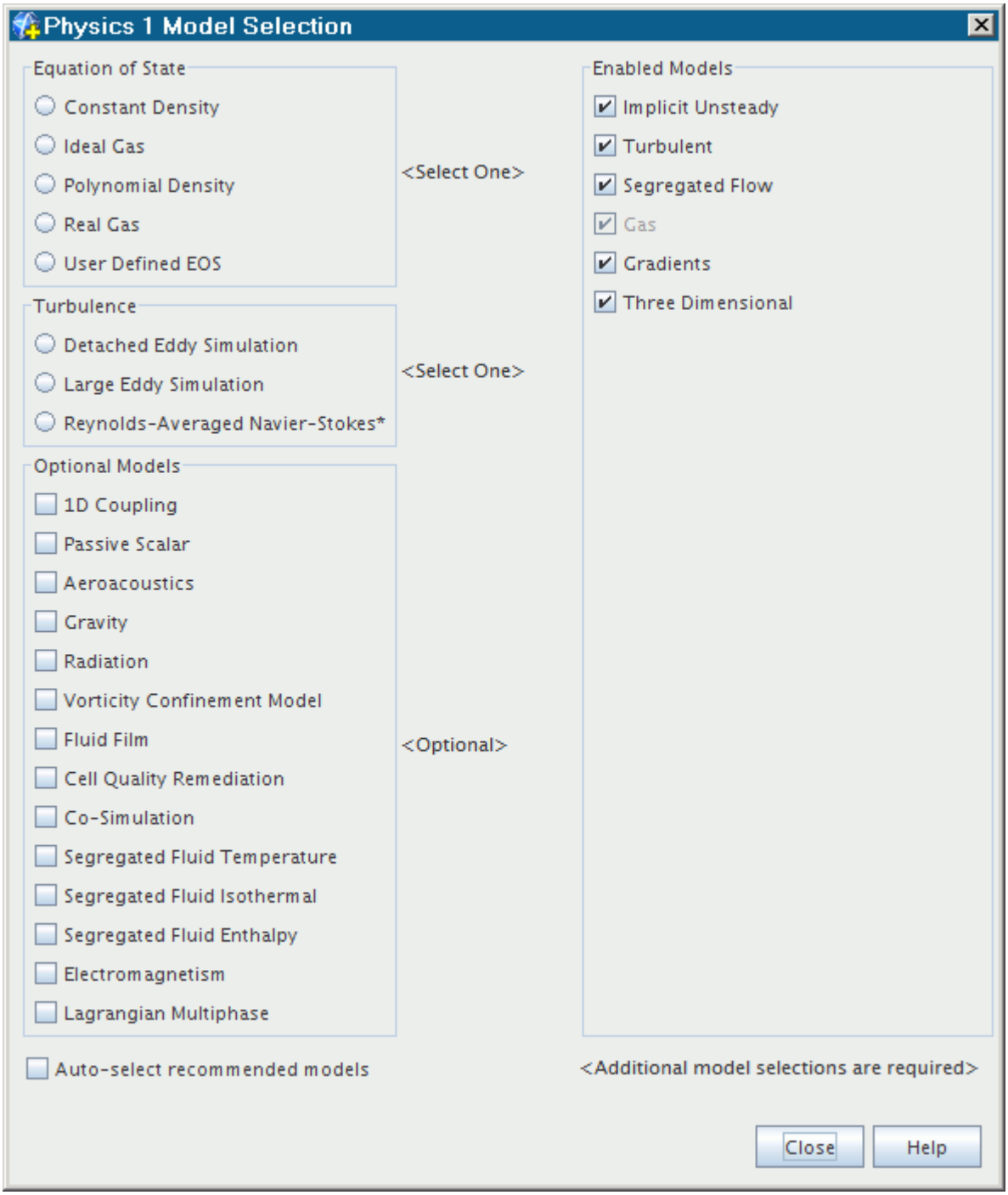
Without Using Auto-Select

This section describes how to select a turbulence model that supports DES without using the automatic selection of recommended models feature.

Use this selection method if large eddy simulation or detached eddy simulation is desired. Note too that these two turbulence modeling approaches are only available if the following prerequisite models are selected:

- the Three-Dimensional model;
- the Implicit Unsteady model;
- the Segregated Flow model.

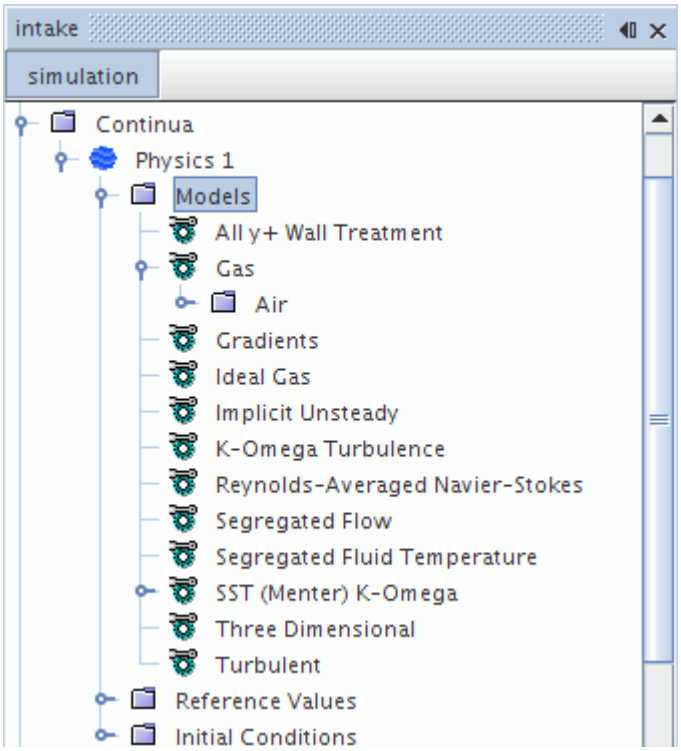
To select the turbulence modeling approach, first select the viscous regime. In the *Model Selection* dialog, make sure that the **Auto-select recommended physics models** checkbox is cleared. Then, in the *Turbulence* group box, select the radio button corresponding to the appropriate turbulence method. The availability of methods depends partly on your selection of a time model.



Depending on your selection, you will now be able to select:

- a (RANS) turbulence model;
- a subgrid scale model for large eddy simulation (LES); or
- a turbulence model for detached eddy simulation.

The result for the SST K-Omega model is shown below.


intake
simulation
Continua
Physics 1
Models
All y+ Wall Treatment
Gas
Air
Gradients
Ideal Gas
Implicit Unsteady
K-Omega Turbulence
Reynolds-Averaged Navier-Stokes
Segregated Flow
Segregated Fluid Temperature
SST (Menter) K-Omega
Three Dimensional
Turbulent
Reference Values
Initial Conditions

Reynolds-Averaged Navier-Stokes (RANS) Turbulence Models

RANS turbulence models provide closure relations for the Reynolds-Averaged Navier-Stokes equations, that govern the transport of the mean flow quantities.

Note: In this context, unless otherwise stated, it should be understood that the term “turbulence model” refers to RANS closure models.

To obtain the Reynolds-Averaged Navier-Stokes equations, each solution variable ϕ in the instantaneous Navier-Stokes equations is decomposed into its mean, or averaged, value $\overline{\phi}$ and its fluctuating component ϕ':

$$\phi = \overline{\phi} + \phi' \tag{378}$$

where ϕ represents velocity components, pressure, energy, or species concentration.

The averaging process may be thought of as time-averaging for steady-state situations and ensemble averaging for repeatable transient situations. The resulting equations for the mean quantities are essentially identical to the original equations, except that an additional term now appears in the momentum transport equation. This additional term is a tensor quantity, known as the Reynolds stress tensor, which has the following definition:

$$\mathbf{T}_t = -\rho \begin{pmatrix} \overline{u'u'} & \overline{u'v'} & \overline{u'w'} \\ \overline{u'v'} & \overline{v'v'} & \overline{v'w'} \\ \overline{u'w'} & \overline{v'w'} & \overline{w'w'} \end{pmatrix} \tag{379}$$

where:

- ρ is the density.
- u, v, and w are the velocity components.

The challenge is thus to model $\mathbf{T}_t$ in terms of the mean flow quantities, and hence provide closure of the governing equations. Two basic approaches are used in STAR-CCM+:

- Eddy viscosity models
- Reynolds stress transport models

Eddy Viscosity Models

Eddy viscosity models use the concept of a turbulent eddy viscosity μ_t to model the Reynolds stress tensor as a function of mean flow quantities.

The most common model is known as the Boussinesq approximation:

$$\mathbf{T}_t = 2\mu_t \mathbf{S} - \frac{2}{3}(\mu_t \nabla \cdot \overline{\mathbf{v}})\mathbf{I} \tag{380}$$

where:

- $\mathbf{S}$ is the mean strain rate tensor given by Eqn. (1745).
- $\overline{\mathbf{v}}$ is the mean velocity.
- $\mathbf{I}$ is the identity tensor.

While some simpler models rely on the concept of mixing length to model the turbulent viscosity in terms of mean flow quantities (similar to the Smagorinsky Subgrid Scale model used in Large Eddy Simulation (LES)), the eddy viscosity models in STAR-CCM+ solve additional transport equations for scalar quantities that enable the turbulent viscosity μ_t to be derived. These include the following turbulence models:

- Spalart-Allmaras models
- K-Epsilon models
- K-Omega models

The assumption that the Reynolds stress tensor is linearly proportional to the mean strain rate does not consider anisotropy of turbulence. In order to account for this turbulence anisotropy, some two-equation models are provided with an option to extend the linear approximation to include non-linear constitutive relations.

Reynolds Stress Transport (RST) Models

Reynolds Stress Transport (RST) models, also known as second-moment closure models, solve transport equations for each component of the Reynolds stress tensor.

Contents:

Deciding on a RANS Turbulence Model
Spalart-Allmaras Turbulence
K-Epsilon Turbulence
K-Omega Turbulence
Reynolds Stress Transport Turbulence
Initialization and Inflow for RANS Turbulence Models
Troubleshooting RANS Turbulence Solvers
Modeling Turbulent Heat Flux
Scaling Turbulent Viscosity

Deciding on a RANS Turbulence Model

There are four major classes of RANS turbulence models currently in STAR-CCM+. This section presents broad guidelines as to the applicability of each of these.

Further guidance on selecting the specific model variants can be found within the sections that provide details about the models.

- Spalart-Allmaras models are a good choice for applications in which the boundary layers are largely attached and separation is mild if it occurs. Typical examples would be flow over a wing, fuselage or other aerospace external-flow applications.

The Spalart-Allmaras models for RANS equations are not suited to flows that are dominated by free-shear layers, flows where complex recirculation occurs (particularly with heat transfer), or natural convection. This statement does not apply to the Spalart-Allmaras detached eddy model.

- K-Epsilon models provide a good compromise between robustness, computational cost and accuracy. They are generally well suited to industrial-type applications that contain complex recirculation, with or without heat transfer.
- K-Omega models are similar to K-Epsilon models in that two transport equations are solved, but differ in the choice of the second transported turbulence variable. The performance differences are likely to be a result of the subtle differences in the models, rather than a higher degree of complexity in the physics being captured. These models have seen most application in the aerospace industry. Therefore, they are recommended as an alternative to the Spalart-Allmaras models for similar types of applications.
- Reynolds Stress Transport models are the most complex and computationally expensive models offered in STAR-CCM+. They are recommended for situations in which the turbulence is strongly anisotropic, such as the swirling flow in a cyclone separator.

Spalart-Allmaras Turbulence

The Spalart-Allmaras turbulence model is a one-equation model that solves a transport equation for the modified diffusivity $\tilde{\nu}$ in order to determine the turbulent eddy viscosity.

This approach is in contrast to many of the early one-equation models that solved an equation for the transport of turbulent kinetic energy and required an algebraic prescription of a length scale. (See [314].)

The original model was developed primarily for the aerospace industry, and has the advantage of being readily implemented in an unstructured CFD solver (unlike the more traditional aerospace models, such as those of Baldwin-Lomax [310] and Johnson-King [312]). This advantage has resulted in its popularity increasing as the use of unstructured CFD methods has grown more widespread in the aerospace industry.

The authors of the original Spalart-Allmaras turbulence model presented results for attached boundary layers and flows with mild separation (such as flow past a wing). It is reasonable to expect that these cases are the types of flows for which the model yields the best results. Wilcox [315] presents free-shear flow spreading rates for the model. While acceptable results are obtained for wake, mixing layer and radial jet flows, the predicted spreading rates for plane and round jets are inaccurate. Therefore, Wilcox concludes that the model is not suited to applications involving jet-like free-shear regions. It is also likely to be less suited to flows involving complex recirculation and body forces (such as buoyancy) than two-equation models such as K-Epsilon and K-Omega or Reynolds Stress Transport.

Spalart-Allmaras Model Variants

The following variants of the Spalart-Allmaras model are implemented in STAR-CCM+:

Model Variants
Standard Spalart-Allmaras
High-Reynolds Number Spalart-Allmaras
Spalart-Allmaras Detached Eddy

In addition, separate versions of the first two variants exist for use with harmonic balance.

Standard Spalart-Allmaras

In its standard form, the Spalart-Allmaras model is a low Reynolds number model, meaning that it is applied without wall functions.

According to the formulation of the model, the entire turbulent boundary layer, including the viscous sublayer, can be accurately resolved and the model can be applied on fine meshes (small values of y^+). In STAR-CCM+, this model is also available with an all-y^+ wall treatment.

High-Reynolds Number Spalart-Allmaras

In the derivation of the model, the authors present a high Reynolds number version in which the viscous damping within the buffer layer and viscous sublayer is not included.

This form of the model, analogous to a K-Epsilon model with wall functions, is also available in STAR-CCM+. It is only suited to coarse, wall-function-type meshes where the y^+ values generally exceed about 30. The standard model with an all-y^+ wall treatment can give similar results to this model on coarse meshes.

Spalart-Allmaras Detached Eddy

See Spalart-Allmaras Detached Eddy Model.

Contents:

Deciding on a Spalart-Allmaras Model
Spalart-Allmaras Models Reference
Spalart-Allmaras Initial Conditions Reference
Spalart-Allmaras Boundaries Reference
Spalart-Allmaras Regions Reference
Spalart-Allmaras Solvers Reference
Spalart-Allmaras Field Functions Reference

Deciding on a Spalart-Allmaras Model

To choose the appropriate model and wall treatment for your application take into account the following recommendations:

- The Standard Spalart-Allmaras model, combined with the all-y^+ wall treatment, is recommended for most, if not all, applications.
- The Standard Spalart-Allmaras model, combined with the low-y^+ wall treatment, could be used in situations where the mesh is known to be fine enough to resolve the viscous sublayer everywhere. The model is designed to give substantially the same results as the all-y^+ wall treatment.
- The High-Reynolds Number Spalart-Allmaras model is provided as a check on the effect of the damping functions on the performance of the mode when the mesh is not coarse enough to be unaffected by the damping functions in the model ($20 < y^+ < 30$). It is not recommended for general-purpose applications.
- The Detached Eddy Spalart-Allmaras model is only available with unsteady simulations, since detached eddy simulation is an inherently unsteady phenomenon.

The available Spalart-Allmaras models and wall treatments available in STAR-CCM+ are:

Spalart-Allmaras Variant	Wall Treatment
Standard	low-y^+, all-y^+
High-Reynolds Number	high-y^+
Detached Eddy	low-y^+, all-y^+

Spalart-Allmaras Models Reference

The **Spalart-Allmaras Turbulence** models allow you to compute the modified diffusivity $\tilde{\nu}$ to provide closure to the Reynolds-Averaged Navier-Stokes equations.

Models Overview

Model Names and Abbreviations	**High-Reynolds Number Spalart-Allmaras**	**HRe SA**
	Standard Spalart-Allmaras*	**SSA**
Theory	See Theory Guide—Spalart Allmaras Model.	
Provided By	**[physics continuum]** > **Models** > *Rans Spalart-Allmaras Turbulence*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Standard Spalart-Allmaras**	
Requires	*Space*: **Axisymmetric**, **Two Dimensional**, or **Three Dimensional** *Time*: **Steady**, **Implicit Unsteady**, **PISO Unsteady**, or **Harmonic Balance** *Material*: **Gas**, **Liquid**, **Eulerian Multiphase**, **Multi-Component Gas**, or **Multi-Component Liquid** *Viscous Regime*: **Turbulent** *Turbulence*: **Reynolds-Averaged Navier-Stokes** *Reynolds-Averaged Turbulence*: **Spalart-Allmaras Turbulence**	
Properties	See Properties Lookup.	
Activates	Physics Models	*Wall Distance*: **Exact Wall Distance** and **PDE Wall Distance**
		Wall Treatment: **All y+ Wall Treatment** and **Low y+ Wall Treatment** (SSA), or **High y+ Wall Treatment** (HRe SA)
	Model Controls (child nodes)	**Curvature Correction Parameters** (optional, see Curvature Correction Option)
	Initial Conditions	See Spalart-Allmaras Initial Conditions Reference.
	Boundary Inputs	See Spalart-Allmaras Boundaries Reference.
	Region Inputs	See Spalart-Allmaras Regions Reference.

	Solvers	**Spalart-Allmaras Turbulence** **Spalart-Allmaras Turbulent Viscosity** See Spalart-Allmaras Solvers Reference.
	Monitors	**Sa_nut** (modified diffusivity)
	Field Functions	**Effective Viscosity** **Modified Diffusivity** **Turbulent Viscosity** **Turbulent Viscosity Ratio** See Spalart-Allmaras Field Functions Reference.

Properties Lookup

This table shows which properties are used by which **Spalart-Allmaras Turbulence** model. Use the abbreviations given in the Models Overview above.

	HRe SA	SSA
Cb1 The coefficient C_{b1} , Eqn. (1764) and Eqn. (1765).	✔	✔
Cb2 The coefficient C_{b2} , Eqn. (1763) and Eqn. (1765).	✔	✔
Convection Controls the convection scheme. • **1st-order**: Selects the first-order upwind convection scheme. • **2nd-order**: Selects the second-order upwind convection scheme.	✔	✔
Cprod The coefficient C_{prod} , Eqn. (1768).	✔	✔
Curvature Correction Option Controls whether to apply a curvature correction, that accounts for effects of strong streamline curvature and rapid frame-rotation. When **On**, a rotation function f_{r1} is applied, that alters the turbulent kinetic energy production term according to local rotation and vorticity rates (see Eqn. (1764)). Only use this option when you have a stable solution, to improve the results. **Curvature Correction Parameters** Child node that provides the parameters *Cr1*, *Cr2*, and *Cr3* to calculate f_{r1} using Eqn. (1911).	✔	✔

	HRe SA	SSA
Cv1 The coefficient C_{v1} , Eqn. (1769).		✔
Cw2 The coefficient C_{w2} , Eqn. (1765).	✔	✔
Cw3 The coefficient C_{w3} , Eqn. (1765).	✔	✔
Deformation Option Selects the modified or standard production term. • **Modified**: Selects the modified production term. This option is recommended for most simulations. See Eqn. (1768). • **Original**: Selects the standard production term. This option allows comparison with results from other CFD codes that lack the modified production term. See Eqn. (1767).	✔	✔
Kappa The von Karman constant κ , Eqn. (1765) and Eqn. (1766).	✔	✔
Secondary Gradients Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces. • **On**: Include both secondary gradients. • **Off**: Exclude both secondary gradients. • **Interior Only**: Include the interior secondary gradients only. • **Boundaries Only**: Include the boundary secondary gradients only.	✔	✔
Sigma The coefficient $\sigma_{\tilde{v}}$,Eqn. (1761).	✔	✔

Spalart-Allmaras Initial Conditions Reference

This section details the conditions and values that are required for initialization of a Spalart-Allmaras turbulence model.

Turbulence Specification

Controls how you define the turbulence profile for initialization.

The Spalart-Allmaras model requires the modified turbulent diffusivity $\tilde{\nu}$. You can enter the corresponding value directly or have it derived from other turbulence quantities.

Method	Corresponding Value Nodes
Modified Turbulent Diffusivity	**Modified Diffusivity** Scalar profile value to specify $\tilde{\nu}$ directly.
Intensity + Length Scale Calculation of $\tilde{\nu}$ from a specified coefficient C_μ, turbulence intensity I, length scale l, and velocity scale v using Eqn. (1955). Do not use this method if the velocity field is initialized to zero.	**Cmu coefficient** Scalar value to specify C_μ. **Turbulence Intensity** Scalar profile value to specify I. **Turbulent Length Scale** Scalar profile value to specify l. **Turbulent Velocity Scale** Scalar profile value to specify v. For an initial value, use a representative velocity or a representative velocity scale. For example, for a pipe flow use the inlet velocity.
Turbulent Viscosity Ratio Calculation of $\tilde{\nu}$ from a specified turbulent viscosity ratio μ_t/μ using Eqn. (1956).	**Turbulent Viscosity Ratio** Scalar profile value to specify the ratio of turbulent to laminar viscosity μ_t/μ.

Spalart-Allmaras Boundaries Reference

This section details the conditions that are required for each boundary type that is supported by the Spalart-Allmaras turbulence model. In addition, information is provided about the values for each boundary type.

Boundary conditions and values are elected by the boundary type, and/or by methods selected in corresponding nodes in the **Conditions** node.

Note: Boundary types that do not require setting any conditions or values are not listed.

Flow Boundaries

The following boundary conditions and values are equal to all flow boundaries of type:

- Free Stream
- Mass Flow Inlet
- Pressure Outlet
- Stagnation Inlet
- Velocity Inlet

Turbulence Specification

Controls how you define the turbulence profile.

The Spalart-Allmaras model requires the modified turbulent diffusivity $\tilde{\nu}$. You can enter the corresponding value directly or have it derived from other turbulence quantities.

Method	Corresponding Value Nodes
Modified Turbulent Diffusivity	**Modified Diffusivity** Scalar profile value to specify $\tilde{\nu}$ directly.
Intensity + Length Scale Calculation of $\tilde{\nu}$ from a specified coefficient C_μ, turbulence intensity I, and length scale l using Eqn. (1955). Do not use this method if the velocity field is initialized to zero.	**Cmu coefficient** Scalar value to specify C_μ. **Turbulence Intensity** Scalar profile value to specify I. **Turbulent Length Scale** Scalar profile value to specify l.
Turbulent Viscosity Ratio Calculation of $\tilde{\nu}$ from a specified turbulent viscosity ratio μ_t/μ using Eqn. (1956).	**Turbulent Viscosity Ratio** Scalar profile value to specify the ratio of turbulent to laminar viscosity μ_t/μ.

Spalart-Allmaras Regions Reference

This section details the conditions that are required for each region type that is supported by the Spalart-Allmaras turbulence model. In addition, information is provided about the values for each region type.

Region conditions and values are elected by the region type, and/or by methods selected in corresponding nodes in the **Conditions** node.

Note: Region types that do not require setting any conditions or values are not listed.

Fluid Region

Turbulence Source Option

Controls whether you want to use a turbulence source term.

Turbulence Source Option	Corresponding Value Nodes
Deactivated	None.
Activated	**Modified Diffusivity Source** Scalar profile value to specify a turbulence source for $\tilde{\nu}$ directly.

Porous Region

Turbulence Specification

Controls how you define the turbulence profile.

The Spalart-Allmaras model requires the modified turbulent diffusivity $\tilde{\nu}$. You can enter the corresponding value directly or have it derived from other turbulence quantities.

Method	Corresponding Value Nodes
Modified Turbulent Diffusivity	**Modified Diffusivity** Scalar profile value to specify $\tilde{\nu}$ directly.
Intensity + Length Scale Calculation of $\tilde{\nu}$ from a specified coefficient C_μ, turbulence intensity I, and length scale l using Eqn. (1955). Do not use this method if the velocity field is initialized to zero.	**Cmu coefficient** Scalar value to specify C_μ. **Turbulence Intensity** Scalar profile value to specify I. **Turbulent Length Scale** Scalar profile value to specify l.
Turbulent Viscosity Ratio Calculation of $\tilde{\nu}$ from a specified turbulent viscosity ratio μ_t/μ using Eqn. (1956).	**Turbulent Viscosity Ratio** Scalar profile value to specify the ratio of turbulent to laminar viscosity μ_t/μ.

Spalart-Allmaras Solvers Reference

This section details the solvers that are associated with the Spalart-Allmaras turbulence model: turbulence and turbulent viscosity.

Spalart-Allmaras Turbulence

This solver controls the solution of the turbulence transport equations in all the continua for which a Spalart-Allmaras model is activated.

For the transported variable $\tilde{\nu}$ the basic steps that are involved in the solution update are as follows:

1. Update boundary conditions.
2. Compute the reconstruction gradients and cell gradients.
3. Create the linear system of equations using discretization methods.
4. Compute the residual sum $R = \sum_{cells} |r|$ for monitoring convergence.
5. Solve the linear system.
6. Update the transported variable field.

Properties

Under-Relaxation Factor

At each iteration, governs the extent to which the newly computed solution supplants the old solution. The default value is 0.7. For the theoretical background, see Eqn. (1625).

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Under-Relaxation Factor Ramp

The **Spalart-Allmaras Turbulence** > **Under-Relaxation Factor Ramp** node allows you to set and control a linear ramp for under-relaxation. For details, see Under-Relaxation Factor Ramp Reference.

AMG Linear Solver

The **Spalart-Allmaras Turbulence** > **AMG Linear Solver** node allows you to set the algebraic multigrid parameters for the turbulence solver. For details, see Algebraic Multigrid and AMG Linear Solver Properties.

Spalart-Allmaras Turbulent Viscosity

This solver controls the update of the turbulent viscosity.

Let μ_t^n be the value of the previous iteration, and μ_t^{new} be the value that is computed during the current iteration. The update is controlled as follows:

$$\mu_t^{n+1} = \omega_\mu \mu_t^{\text{new}} + \left(1-\omega_\mu\right)\mu_t^n$$

Properties

Maximum Ratio	The maximum ratio of turbulent to laminar viscosity (μ_t/μ) permitted during the update process. This ratio prevents unphysically high values of turbulent viscosity from occurring on the path to convergence. See Troubleshooting RANS Solvers.

Under-Relaxation Factor	An under-relaxation factor ω_μ for the update of turbulent viscosity. The default value is 1.0.
Solver Frozen	When **On**, this prevents the solver from updating the solution while iterating. The property is **Off** by default.

Spalart-Allmaras Field Functions Reference

This section details the field functions that are made available to the simulation when a Spalart-Allmaras turbulence model is used.

Effective Viscosity

Scalar field that represents the sum of the laminar and turbulent viscosities $\mu + \mu_t$.

Modified Diffusivity

The transported variable $\tilde{\nu}$.

Turbulent Viscosity

Scalar field that represents the turbulent viscosity μ_t.

Turbulent Viscosity Ratio

Scalar field that represents the ratio of turbulent to laminar viscosity μ_t/μ.

K-Epsilon Turbulence

The K-Epsilon turbulence model is a two-equation model that solves transport equations for the turbulent kinetic energy k and the turbulent dissipation rate ε in order to determine the turbulent eddy viscosity.

Various forms of the K-Epsilon model have been in use for a number of decades, and it has become the most widely used model for industrial applications. Since the inception of the K-Epsilon model, there have been countless attempts to improve it. The most significant improvements have been incorporated into STAR-CCM+.

High Reynolds Number Approach

The original K-Epsilon turbulence model by Jones and Launder [319] was applied with wall functions. This high Reynolds number approach was later modified to take into account the blocking effects of the wall (viscous and buffer layer) using a low Reynolds number approach and a two-layer approach.

Low Reynolds Number Approach

The most common approach is to apply damping functions to some or all of the coefficients in the model (C_μ, $C_{\varepsilon 1}$, and $C_{\varepsilon 2}$). These damping functions modulate the coefficients as functions of a turbulence Reynolds number, often also incorporating the wall distance. Dozens of models incorporating damping functions have been proposed in the literature, and two have been incorporated into STAR-CCM+:

- Standard K-Epsilon Low-Re model

- AKN K-Epsilon Low-Re model

Another low Reynolds number K-Epsilon model implemented in STAR-CCM+ is the V2F Model. This model, however, does not use wall distance or damping functions; it is claimed to be valid throughout the flow domain while automatically modeling the region close to solid surfaces.

For details, see K-Epsilon Model Variants.

Two-Layer Approach

The two-layer approach, first suggested by Rodi [324], is an alternative to the low Reynolds number approach that allows the K-Epsilon model to be applied in the viscous-affected layer (including the viscous sub-layer and the buffer layer).

In this approach, the computation is divided into two layers. In the layer next to the wall, the turbulent dissipation rate ε and the turbulent viscosity μ_t are specified as functions of wall distance. The values of ε specified in the near-wall layer are blended smoothly with the values computed from solving the transport equation far from the wall. The equation for the turbulent kinetic energy is solved across the entire flow domain. This explicit specification of ε and μ_t is arguably no less empirical than the damping function approach, and the results are often as good or better.

Several types of two-layer formulations have been proposed, and three have been implemented in STAR-CCM+, two for shear-driven flows and one for buoyancy-driven flows:

- Shear Driven (Wolfstein)
- Shear Driven (Norris-Reynolds)
- Buoyancy Driven (Xu)

In STAR-CCM+, the two-layer formulations work with either low Reynolds number type meshes $y^+ \sim 1$ or wall-function type meshes $y^+ > 30$.

Of the two models for shear-driven flows, neither is clearly better. The Wolfstein model is activated by default, but the Norris-Reynolds model is offered as an option since some users find it preferable. Only use the Xu natural-convection model for buoyant flows.

K-Epsilon Model Variants

The following variants of the K-Epsilon model are implemented in STAR-CCM+:

Model Variants
Standard K-Epsilon
Standard K-Epsilon Two-Layer
Standard K-Epsilon Low-Re
AKN K-Epsilon Low-Re
Realizable K-Epsilon
Realizable K-Epsilon Two-layer
EB K-Epsilon
Lag EB K-Epsilon

Model Variants
EB K-Epsilon Detached Eddy
V2F K-Epsilon

Standard K-Epsilon

The Standard K-Epsilon model is a *de facto* standard version of the two-equation model that involves transport equations for the turbulent kinetic energy k and its dissipation rate ε. The transport equations are of the form suggested by Jones and Launder [319], with coefficients suggested by Launder and Sharma [321]. Some additional terms have been added to the model in STAR-CCM+ to account for effects such as buoyancy and compressibility. An optional non-linear constitutive relation is also provided.

Standard K-Epsilon Two-Layer

The Standard K-Epsilon Two-Layer model combines the Standard K-Epsilon model with the two-layer approach.

The coefficients in the models are identical, but the model gains the added flexibility of an all-y^+ wall treatment.

Standard K-Epsilon Low-Re

The Standard K-Epsilon Low-Reynolds Number model combines the Standard K-Epsilon model with the low Reynolds number approach.

The low Reynolds number model by Lien and others is dubbed the “Standard Low-Reynolds Number K-Epsilon Model” because it has identical coefficients to the Standard K-Epsilon model, but it provides more damping functions. These functions enable it to be applied in the viscous-affected regions near walls. This model is recommended for natural convection problems, particularly if the Yap correction is invoked. (See [322].)

AKN K-Epsilon Low-Re

The Abe-Kondoh-Nagano K-Epsilon Low-Reynolds Number model [316] has different coefficients than the Standard K-Epsilon model, and uses different damping functions than the Standard K-Epsilon Low-Reynolds Number model.

This model is known to work well for a wide range of complex flows. It is a good choice for applications such as compact heat exchangers, where the Reynolds numbers are low but the flow is relatively complex.

Realizable K-Epsilon

The Realizable K-Epsilon model contains a new transport equation for the turbulent dissipation rate ε [326]. Also, a critical coefficient of the model C_μ is expressed as a function of mean flow and turbulence properties, rather than assumed to be constant as in the standard model. This procedure lets the model satisfy certain mathematical constraints on the normal stresses consistent with the physics of turbulence

(realizability). The concept of a variable C_μ is also consistent with experimental observations in boundary layers.

This model is substantially better than the Standard K-Epsilon model for many applications, and can generally be relied upon to give answers that are at least as accurate. Both the standard and realizable models are available in STAR-CCM+ with the option of using a two-layer approach, which enables them to be used with fine meshes that resolve the viscous sublayer.

Realizable Two-Layer K-Epsilon

The Realizable Two-Layer K-Epsilon model combines the Realizable K-Epsilon model with the two-layer approach.

The coefficients in the models are identical, but the model gains the added flexibility of an all-y^+ wall treatment.

EB K-Epsilon

The concept of elliptic relaxation was proposed by Durbin [342] for Reynolds-stress models. The initial model required the solution of six additional transport equations, but this number was later reduced to a single additional equation. The model was later simplified by Manceau and Hanjalić [345] to make it more industry-friendly.

The elliptic relaxation model led to the development of some two-equation eddy viscosity models, starting with the $\overline{\vartheta^2} - f$ model. One of the most robust models, according to its authors, is the $BL{-}\overline{\vartheta^2}/k$ model of Billard and Laurence [341]. Important modifications to this model make it truly robust for complex flow geometries.

The benefits of using the Billard - Laurence model are:

- An improvement on the existing realizable $k{-}\varepsilon$ model in terms of accuracy, especially in the near-wall region.
- An improvement in terms of stability over the SST $k{-}\omega$ model.

Lag EB K-Epsilon

The Lag Elliptic Blending model combines the Standard Elliptic Blending model with the stress-strain lag concept originally proposed by Revell et al. [347].

In flow regions where non-equilibrium effects result in a misalignment of the principal components of the stress and strain-rate tensors, linear eddy viscosity models tend to over-predict the production of k. To overcome this effect, the Lag Elliptic Blending model incorporates the angle between these components. Additional terms model the effects of anisotropy—similar to non-linear constitutive relations—and curvatures and rotational effects—similar to curvature correction. These terms are directly embedded in the transport equation for the reduced stress function φ [346].

The Lag Elliptic Blending model provides a good predictive capability for separated or unsteady flow (such as vortex shedding), or when the flow is subject to rotation or strong streamline curvature.

EB K-Epsilon Detached Eddy

See EB K-Epsilon Detached Eddy Model.

V2F K-Epsilon

The V2F turbulence model solves transport equations for two more turbulence quantities in addition to k and ε, namely the wall-normal stress component $\overline{\vartheta^2} = \overline{\mathrm{v}'\mathrm{v}'}$ and the elliptic relaxation parameter f, in order to determine the turbulent eddy viscosity.

The V2F model ([348], [350], [349]) is designed to handle wall effects in turbulent boundary layers and to accommodate non-local effects. It is known to capture the near-wall turbulence effects more accurately, which is crucial for the accurate prediction of heat transfer, skin friction and flow separation.

Contents:

Deciding on a K-Epsilon Model
Controlling a K-Epsilon Model
K-Epsilon Models Reference
K-Epsilon Initial Conditions Reference
K-Epsilon Boundaries Reference
K-Epsilon Regions Reference
K-Epsilon Solvers Reference
K-Epsilon Field Functions Reference

Deciding on a K-Epsilon Model

To choose the appropriate model and wall treatment for your application take into account the following recommendations:

- The Standard K-Epsilon model and the Realizable K-Epsilon model are suitable for coarse meshes, where the wall-cell y^+ values are typically 30 and above. The Realizable model generally gives results at least as good as the Standard model, but typically better.
- The Standard Two-Layer K-Epsilon model and the Realizable Two-Layer K-Epsilon model offer the most mesh flexibility. They can be used with the same meshes as the high Reynolds number versions. They give good results on fine meshes (that is, low Reynolds number type or low-y^+ meshes), and also produce the least inaccuracies for intermediate meshes (that is, $1 < y^+ < 30$).
- The Standard Low-Reynolds Number model, Abe-Kondoh-Nagano Low-Reynolds number model, and V2F Low-Reynolds number model are recommended for truly low Reynolds number applications. The Standard Low-Reynolds number model is suitable for natural convection flows, for situations when it is desirable to have a low Reynolds number version of the Standard K-Epsilon model, or when a Non-linear Constitutive Model is needed with a low Reynolds number treatment. The Abe-Kondoh-Nagano model is appropriate for other complex low Reynolds number flows, where a linear constitutive model and damping-function approach is adequate. The V2F model is recommended for applications where it is crucial to capture the near-wall turbulence effects more accurately.
- The Elliptic Blending model accurately models near wall anisotropy and is valid both in the low Reynolds number boundary layer and the high Reynolds number bulk flow. The model is well suited for internal flows, heat transfer modeling, and other cases where accurate near wall modeling is important.

- The Lag Elliptic Blending model improves the sensitivity of the Elliptic Blending model to anisotropy of turbulence and streamline curvature and rotation. The Lag Elliptic Blending model is well suited for external aerodynamics simulations.

If there is uncertainty as to which turbulence model to use in a given situation, the Realizable Two-Layer K-Epsilon model would be a reasonable choice. If the mesh is coarse, it provides results that are quite close to the version without the two-layer formulation. If the mesh is fine enough to resolve the viscous sublayer, it produces results similar to a low Reynolds number model.

The K-Epsilon models and wall treatments available in STAR-CCM+ are:

K-Epsilon Model	Wall Treatment
Standard	high-y^+
Standard Two-Layer	all-y^+
Realizable	high-y^+
Realizable Two-Layer	all-y^+
Standard Low-Reynolds Number	low-y^+, all-y^+
Elliptic Blending	low-y^+, all-y^+
Lag Elliptic Blending	low-y^+, all-y^+
Abe-Kondoh-Nagano Low-Reynolds Number	low-y^+, all-y^+
V2F Low-Reynolds Number	low-y^+, all-y^+

Controlling a K-Epsilon Model

This section provides guidelines for controlling simulations with K-Epsilon turbulence models:

- Accounting for Strong Streamline Curvature and Frame-Rotation
- Overcoming an Unexpectedly Large Growth of K
- Accounting for Anisotropy of Turbulence

Accounting for Strong Streamline Curvature and Frame-Rotation

Curvature correction is a factor that alters the turbulent energy production term according to local rotation and vorticity rates.

The **Curvature Correction Parameters** node provides the properties with which you can control the curvature correction factor. It is activated as a subnode of any realizable K-Epsilon model node when the **Curvature Correction Option** property is enabled.

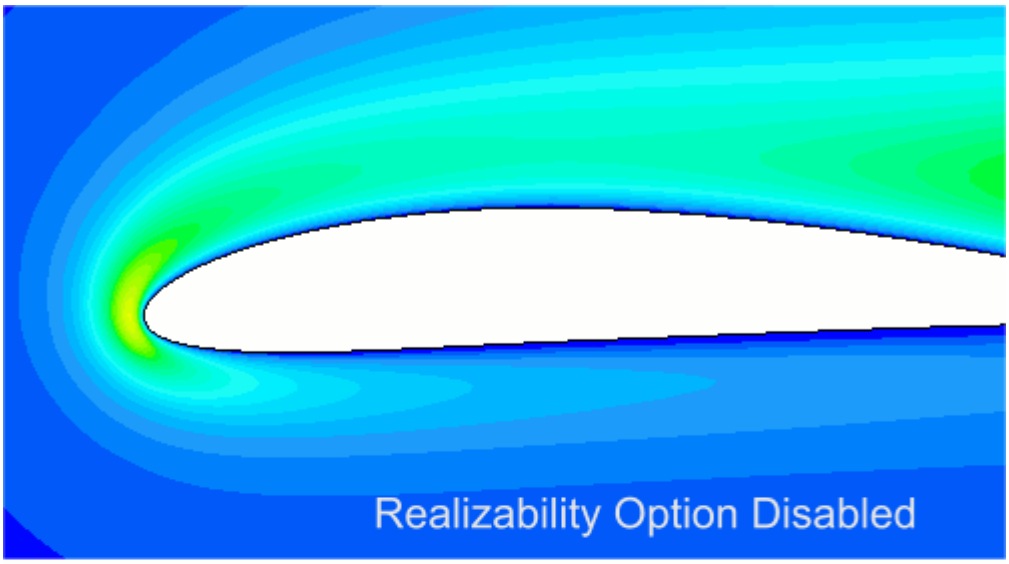
This feature is disabled by default. It is only necessary for cases that have strong streamline curvature or a rapidly rotating reference frame. In addition, we recommend you activate it only when you have a stable solution, to improve the results.

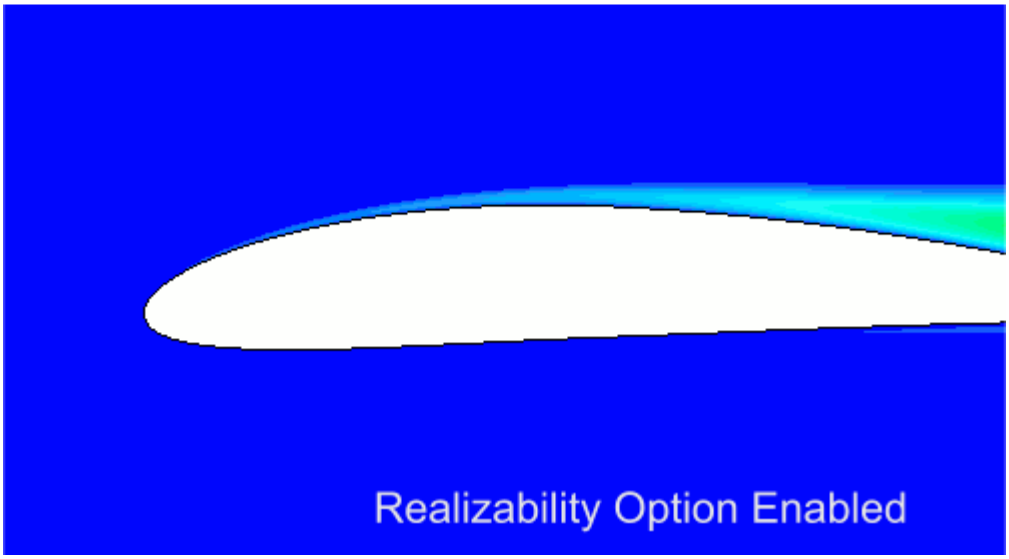
For more information, see Curvature Correction Option.

Overcoming an Unexpectedly Large Growth of K

Two-equation models predict an unexpectedly large growth of turbulent kinetic energy in stagnation point flows [332]. This growth can have an adverse effect on the rest of the flow solution. To overcome this effect, a lower limit on the turbulence time scale has been proposed that imposes the “realizability” constraint on the eddy-viscosity formula. To apply it, set the *Realizability Option* property of the parent model object to **Durbin Scale Limiter**.

The effect of this coefficient can be demonstrated by observing the difference between two solutions of flow over an airfoil. In the first solution that is shown below, the Realizability option has not been activated. The build-up of turbulence at the stagnation point is clearly observed. In the second solution, the Realizability option has been activated.





For more information, see Realizability Option.

Accounting for Anisotropy of Turbulence

You can account for anisotropy of turbulence by extending the Boussinesq approximation to include non-linear functions of the strain and rotation tensors. These non-linear constitutive relations improve turbulence prediction by including effects such as secondary flows, streamline curvature and rotation [322].

Relation coefficients appear as sub-nodes of the Standard K-Epsilon, Standard Two-Layer K-Epsilon, or Standard Low-Reynolds Number model node, depending on the *Constitutive Relation* property setting of that node. Two non-linear relations are available:

- **Quadratic**
- **Cubic**

The Quadratic relation improves the performance of the K-Epsilon model by predicting secondary flows better.

The Cubic relation extends the non-linear constitutive relation of the Quadratic relation by including cubic functions of the velocity gradients. These additional terms improve the sensitivity of the K-Epsilon Model to streamline curvature and rotation.

For more information, see Constitutive Relation.

K-Epsilon Models Reference

The **K-Epsilon Turbulence** models allow you to compute the turbulent kinetic energy k and the turbulent dissipation rate ϵ to provide closure to the Reynolds-Averaged Navier-Stokes equations.

Models Overview

Model Names and Abbreviations	**AKN K-Epsilon Low-Re** (Abe-Kondoh-Nagano K-Epsilon Low-Reynolds Number)	**AKN KE**
	EB K-Epsilon (Elliptic Blending K-Epsilon)	**EB KE**
	Lag EB K-Epsilon (Lag Elliptic Blending K-Epsilon)	**EBL KE**
	Realizable K-Epsilon	**RKE**
	Realizable K-Epsilon Two-Layer*	**RKE 2L**
	Standard K-Epsilon	**SKE**
	Standard K-Epsilon Low-Re (Standard K-Epsilon Low-Reynolds Number)	**SKE LRe**
	Standard K-Epsilon Two-Layer	**SKE 2L**
	V2F K-Epsilon (V2F K-Epsilon Low-Reynolds Number)	**V2F KE**
Theory	See Theory Guide—K-Epsilon Model (AKN KE, RKE, RKE 2L, SKE, SKE LRe, SKE 2L) Theory Guide—Elliptic Blending Model (EB KE, EBL KE) Theory Guide—V2F Model (V2F KE)	
Provided By	**[physics continuum]** > **Models** > *K-Epsilon Turbulence Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Realizable K-Epsilon Two-Layer**	

Requires	Physics Models	*Space*: **Axisymmetric**, **Two Dimensional**, or **Three Dimensional** *Time*: **Steady**, **Implicit Unsteady**, **PISO Unsteady** (all except AKN KE and V2F KE), or **Harmonic Balance** (all except EB KE and EBL KE) *Material*: **Gas**, **Liquid**, **Eulerian Multiphase**, **Multi-Component Gas**, or **Multi-Component Liquid** *Viscous Regime*: **Turbulent** *Turbulence*: **Reynolds-Averaged Navier-Stokes** *Reynolds-Averaged Turbulence*: **K-Epsilon Turbulence**
Properties	See Properties Lookup.	
Activates	Physics Models	*Wall Distance*: **Exact Wall Distance** and **PDE Wall Distance**
		Wall Treatment: **All y+ Wall Treatment** and **Low y+ Wall Treatment** (AKN KE, SKE LRe, EB KE, EBL KE, V2F), **High y+ Wall Treatment** (RKE, SKE), or **Two-Layer All y+ Wall Treatment** (RKE 2L, SKE 2L)
		Optional Models: **Turbulent Viscosity User Scaling** and **Temperature Flux Model** (SKE LRe)
	Model Controls (child nodes)	**C3e** (optional, see Buoyancy Production of Dissipation) **Curvature Correction Parameters** (optional, see Curvature Correction Option) **Non-Linear Cmu Coefficients** (optional, see Constitutive Relation) **Non-Linear Cubic Coefficients** (optional, see Constitutive Relation) **Non-Linear Quadratic Coefficients** (optional, see Constitutive Relation) **Realizability Coefficient** (optional, see Realizability Option)
	Initial Conditions	See K-Epsilon Initial Conditions Reference.
	Boundary Inputs	See K-Epsilon Boundaries Reference.
	Region Inputs	See K-Epsilon Regions Reference.
	Solvers	**K-Epsilon Turbulence** (all except EB KE, EBL KE, and V2F KE) **EB K-Epsilon Turbulence** (EB KE) **Lag K-Epsilon Turbulence** (EBL KE) **Elliptic K-Epsilon Turbulence** (V2F KE) **K-Epsilon Turbulent Viscosity** See K-Epsilon Solvers Reference.
	Monitors	**Tdr** (turbulent dissipation rate) **Tke** (turbulent kinetic energy) **Alpha** (elliptic blending function, EB KE and EBL KE) **Phi** (reduced stress function, EB KE and EBL KE) **F** (elliptic function, V2F KE) **V2** (normal stress function, V2F KE)
	Field Functions	**Effective Viscosity** **Elliptic Blending Function** (EB KE, EBL KE) **Elliptic Function** (V2F KE)

		Kolmogorov Length Scale **Kolmogorov Time Scale** **Normal Stress Function** (V2F KE) **Reduced Stress Function** (EB KE, EBL KE) **Strain Rate Tensor Modulus** **Taylor Micro Scale** **Turb Wall Distance Re** (AKN KE, RKE 2L, SKE LRe, SKE 2L) **Turbulent Dissipation Rate** **Turbulent Kinetic Energy** **Turbulent Viscosity** **Turbulent Viscosity Ratio** See K-Epsilon Field Functions Reference.

Properties Lookup

This table shows which properties are used by which **K-Epsilon Turbulence** model. Use the abbreviations given in the Models Overview above.

	SKE	SKELRe	SKE2L	RKE	RKE2L	EBKE	EBLKE	AKNKE	V2FKE
A The coefficient A in Eqn. (1905).									✔
Alpha Minimum The minimum value that the transported variable α is permitted to have. An appropriate value is a small number that is greater than the floating point minimum of the computer.						✔	✔		
Buoyancy Production of Dissipation Determines how the coefficient $C_{\varepsilon 3}$ in the production term P_ε is calculated (see Eqn. (1776), Eqn. (1876) (EB KE, EBL KE), and Eqn. (1898) (V2F KE)). • **None**: Sets $C_{\varepsilon 3}$ to zero. • **Boundary Layer Orientation**: Computes $C_{\varepsilon 3}$ according to Eqn. (1801). • **Thermal Stratification**: Computes $C_{\varepsilon 3}$ according to Eqn. (1802). • **Constant Coefficient**: Computes $C_{\varepsilon 3}$ as a constant coefficient. This option requires the specification of $C_{\varepsilon 3}$ in the corresponding child node **C3e** .	✔	✔	✔	✔	✔	✔	✔	✔	✔

	SKE	SKELRe	SKE2L	RKE	RKE2L	EBKE	EBLKE	AKNKE	V2FKE
C coefficient The coefficients C in Eqn. (1796).		✓							
C_T The coefficient C_T in Eqn. (1873).						✓	✓		
C1 The coefficient C_1 in Eqn. (1883) (EB KE) and Eqn. (1884) (EBL KE). For V2F KE, the coefficient C_1 is used in Eqn. (1906) and Eqn. (1907).						✓	✓		✓
C1e The coefficient $C_{\varepsilon 1}$ in the basic transport equations.	✓	✓	✓	✓	✓	✓	✓	✓	✓
C1s The coefficient C_1^* in Eqn. (1884).							✓		
C2 The coefficient C_2 in Eqn. (1883) (EB KE). For V2F KE, the coefficient C_2 is used in Eqn. (1906) and Eqn. (1907).						✓			✓
C2e The coefficient $C_{\varepsilon 2}$ in the basic transport equations.	✓	✓	✓	✓	✓	✓	✓	✓	✓
C3 The coefficient C_3 in Eqn. (1891).							✓		
C3e The coefficient $C_{\varepsilon 3}$ in Eqn. (1882).						✓	✓		
C3s The coefficient C_3^* in Eqn. (1884).							✓		

	SKE	SKELRe	SKE2L	RKE	RKE2L	EBKE	EBLKE	AKNKE	V2FKE
C4 The coefficient C_4 in Eqn. (1884).							✓		
C5 The coefficient C_5 in Eqn. (1884).							✓		
Cd0 coefficient The coefficient C_{d0} in Eqn. (1797).		✓							
Cd1 coefficient The coefficient C_{d1} in Eqn. (1797).		✓							
Cd2 coefficient The coefficient C_{d2} in Eqn. (1797).		✓							
Ceta The coefficient C_η used in the calculation of the turbulent length scale L.						✓	✓		✓
Ck The coefficient C_k in Eqn. (1888).						✓	✓		
Cl The coefficient used in the calculation of the turbulent length scale L.						✓	✓		✓
Cmu The coefficient C_μ in the calculation of the turbulent viscosity μ_t and in the basic transport equations.	✓	✓	✓	✓	✓	✓	✓	✓	✓
CmuV2 The coefficient $C_{\mu_{\vartheta^2}}$ used in Eqn. (1894), Eqn. (1896), Eqn. (1903), and Eqn. (1907).									✓

	SKE	SKELRe	SKE2L	RKE	RKE2L	EBKE	EBLKE	AKNKE	V2FKE
Constitutive Relation Controls the type of constitutive relation used. • **Linear**: Selects the linear constitutive relation as implied by the Boussinesq approximation (see Eqn. (1759). Use this selection for most simulations. • **Quadratic**: Selects the quadratic constitutive relation (see Eqn. (1813)). This relation requires the coefficients C_μ, C_1, C_2, and C_3. **Non-Linear Cmu Coefficients** Child node that provides the parameters *Ca0*, *Ca1*, *Ca2*, and *Ca3* to calculate C_μ. **Non-Linear Quadratic Coefficients** Child node that provides the parameters *Cnl1*, *Cnl2*, *Cnl3*, *Cnl6*, and *Cnl7* to calculate C_1, C_2, and C_3. • **Cubic:** Selects the cubic constitutive relation (see Eqn. (1814)). This relation requires the coefficients C_μ, C_1, C_2, C_3, C_4, and C_5. **Non-Linear Cmu Coefficients** As for **Quadratic**. **Non-Linear Quadratic Coefficients** As for **Quadratic**. **Non-Linear Cubic Coefficients** Child node that provides the parameters *Cnl4*, *Cnl5* to calculate C_4 and C_5. For guidelines when to use non-linear constitutive relations see Accounting for Anisotropy of Turbulence.	✓	✓	✓						
Convection Controls the convection scheme. • **1st-order**: Selects the first-order upwind convection scheme. • **2nd-order**: Selects the second-order upwind convection scheme.	✓	✓	✓	✓	✓	✓	✓	✓	✓

	SKE	SKELRe	SKE2L	RKE	RKE2L	EBKE	EBLKE	AKNKE	V2FKE
Ct The coefficient C_t used in the calculation of the turbulent time scale T.	✔	✔	✔	✔	✔	✔	✔	✔	✔
Curvature Correction Option Controls whether to apply a curvature correction, that supplies the effects of strong (streamline) curvature and frame-rotation. When **On**, a curvature correction factor f_c is applied, that alters the turbulent kinetic energy production term according to local rotation and vorticity rates (see Eqn. (1784) to Eqn. (1787)). Only use this option when you have a stable solution, to improve the results. **Curvature Correction Parameters** Child node that provides the parameters *Cr1*, *Cr2*, *Cct* and *Upper Limit* (C_{max}) to calculate f_c using Eqn. (1912). For guidelines when to apply a curvature correction, see Accounting for Strong Streamline Curvature and Frame-Rotation.				✔	✔				
D coefficient The coefficient D in Eqn. (1793).		✔							
E coefficient The coefficient E in Eqn. (1793).		✔							
F Minimuim The minimum value that the variable f is permitted to have. An appropriate value is a small number that is greater than the floating point minimum of the computer.									✔
Free-stream Option Controls the definition of $C^*_{\varepsilon 2}$ in the transport equation for ε (see Eqn. (1876)): • **Variable C2e Option**: Uses a variable $C^*_{\varepsilon 2}$ value (see Eqn. (1889)). Set by default. • **Off**: Uses a constant $C^*_{\varepsilon 2}$ value (see Eqn. (1890)).						✔			

	SKE	SKELRe	SKE2L	RKE	RKE2L	EBKE	EBLKE	AKNKE	V2FKE
For some external aerodynamic cases, the **Variable C2e Option** can cause instabilities in the free-stream. For such cases, set *Free-stream Option* to **Off**.									
Normal Stress Term This property is an explicit term that directly incorporates divergence and turbulent kinetic energy, $-2/3(\mu_t \nabla \cdot \mathbf{v} + \rho k)\mathbf{I}$, according to the full Boussinesq approximation. When **On**, the Reynolds stress tensor is modeled as $\mathbf{T}_t = 2\mu_t \mathbf{S} - 2/3(\mu_t \nabla \cdot \mathbf{v} + \rho k)\mathbf{I}$ and the turbulent production is modeled using $G_k = \mu_t S^2 - \frac{2}{3}\rho k \nabla \cdot \overline{\mathbf{v}} - \frac{2}{3}\mu_t (\nabla \cdot \overline{\mathbf{v}})^2$ When **Off**, the Reynolds stress tensor is modeled as $\mathbf{T}_t = 2\mu_t \mathbf{S}$ and the turbulent production is assumed to be $G_k = \mu_t S^2$ This property is off by default, in which case the quantity $-2/3(\mu_t \nabla \cdot \mathbf{v} + \rho k)\mathbf{I}$ is absorbed into the pressure, and causes the pressure to be slightly different. However, the results are the same even if this term is not explicitly included. In incompressible flow only the gradients of pressure matter, so this setting has no effect on the results. In compressible flow, however, the absolute value of pressure is used in the Ideal Gas Law (Eqn. (1466)). If k is poorly initialized, activating this option affects the solution convergence.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Phi Minimum The minimum value that the transported variable φ is permitted to have. An appropriate value is a small number that is greater than the floating point minimum of the computer.						✓	✓		

	SKE	SKELRe	SKE2L	RKE	RKE2L	EBKE	EBLKE	AKNKE	V2FKE
Realizability Option Controls whether to activate Durbin's realizability constraint on the turbulent time scale T. When set to **Durbin Scale Limiter**, a minimum constraint on the turbulent time scale is applied. See Eqn. (1774) and Eqn. (1896) (V2F KE). **Realizability Coefficient** Child node that provides the *Realizability Coefficient* C_T in Eqn. (1774). For guidelines when to activate Durbin's realizability constraint see Overcoming an Unexpectedly Large Growth of K.	✓	✓	✓					✓	✓
Sarkar The coefficient C_M in the compressibility modification Υ_y (see Eqn. (1794)). For EB KE and EBL KE, see Eqn. (1887). For V2F KE, see Eqn. (1910).	✓	✓	✓	✓	✓	✓	✓	✓	✓
Secondary Gradients Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces. • **On**: Include both secondary gradients. • **Off**: Exclude both secondary gradients. • **Interior Only**: Include the interior secondary gradients only. • **Boundaries Only**: Include the boundary secondary gradients only.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Sigma_e The coefficient σ_e in the basic transport equations.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Sigma_k The coefficient σ_k in the basic transport equations.	✓	✓	✓	✓	✓	✓	✓	✓	✓

	SKE	SKELRe	SKE2L	RKE	RKE2L	EBKE	EBLKE	AKNKE	V2FKE
Sigma_phi The coefficient σ_φ in Eqn. (1877).						✓	✓		
Tdr Minimum The minimum value that the transported variable ε is permitted to have. An appropriate value is a small number that is greater than the floating point minimum of the computer.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Tke Minimum The minimum value that the transported variable k is permitted to have. An appropriate value is a small number that is greater than the floating point minimum of the computer.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Two-Layer Delta ReY The value of $\Delta\ \mathrm{Re}_y$ in Eqn. (1805).			✓		✓				
*Two-Layer ReY** The value of $\mathrm{Re}_y{}^*$ in Eqn. (1804).			✓		✓				
Two-Layer Type Controls the type of two-layer formulation • **Shear Driven (Wolfstein)**: Selects the two-layer formulation of Wolfstein [328]. Use this selection for flows where buoyancy forces do not dominate. See Eqn. (1807) and Eqn. (1808). • **Buoyancy Driven (Xu)**: Selects the two-layer formulation of Xu [329]. Use this selection for flows where buoyancy forces dominate. See Eqn. (1811) and Eqn. (1812). • **Shear Driven (Norris-Reynolds)** : Selects the two-layer formulation of Norris and Reynolds [323]. Use this selection for flows where buoyancy forces do not dominate. See Eqn. (1809) and Eqn. (1810).			✓		✓				
V2 Minimum The minimum value that the transported variable $\overline{\vartheta^2}$ is permitted to have. An appropriate value is a small number that is greater than the floating point minimum of the computer.									✓

	SKE	SKELRe	SKE2L	RKE	RKE2L	EBKE	EBLKE	AKNKE	V2FKE
Yap Correction When **On**, includes the Yap correction Υ_y in the transport equation for ε, see Eqn. (1781)(SKE 2L) and Eqn. (1783) (SKE LRe). The default setting is **Off**.		✔	✔						
Yap Cl The coefficient C_l in the Yap correction (see Eqn. (1795)).		✔	✔						
Yap Cw The coefficient C_w in the Yap correction (see Eqn. (1795)).		✔	✔						

K-Epsilon Initial Conditions Reference

This section details the conditions and values that are required for initialization of a K-Epsilon turbulence model.

Turbulence Specification

Controls how you define the turbulence profile for initialization.

The K-Epsilon model requires the turbulent kinetic energy k and the turbulent dissipation rate ε. You can enter the corresponding values directly or have them derived from other turbulence quantities.

Method	Corresponding Value Nodes
K + Epsilon	**Turbulent Kinetic Energy** Scalar profile value to specify k directly. **Turbulent Dissipation Rate** Scalar profile value to specify ε directly. **Elliptic Function (V2F KE)** Scalar profile value to specify the elliptic function f directly. The V2F KE model solves an additional transport equation for this redistribution term in the $\overline{\vartheta^2}$ equation.

	Normal Stress Function (V2F KE) Scalar profile value to specify the normal stress function $\overline{\vartheta^2}$ directly. The V2F KE model solves an additional transport equation for this quantity. **Reduced Stress Function (EB KE, EBL KE)** Scalar profile value to specify the reduced stress function φ directly. The EB KE model solves an additional transport equation for this quantity.
Intensity + Length Scale Calculation of k and ε from a specified turbulence intensity I, length scale l, and velocity scale v using Eqn. (1957) and Eqn. (1959). Do not use this method if the velocity field is initialized to zero.	**Turbulence Intensity** Scalar profile value to specify I. **Turbulent Length Scale** Scalar profile value to specify l. **Turbulent Velocity Scale** Scalar profile value to specify v. For an initial value, use a representative velocity or a representative velocity scale. For example, for a pipe flow use the inlet velocity. **Reduced Stress Function (EB KE, EBL KE)** As for **K+Epsilon**.
Intensity + Viscosity Ratio Calculation of k and ε from a specified turbulence intensity I, viscosity ratio μ_t/μ, and velocity scale v using Eqn. (1958) and Eqn. (1960). Do not use this method if the velocity field is initialized to zero.	**Turbulence Intensity** As for **Intensity + Length Scale**. **Turbulent Viscosity Ratio** Scalar profile value to specify the ratio of turbulent to laminar viscosity μ_t/μ. **Turbulent Velocity Scale** As for **Intensity + Length Scale**. **Reduced Stress Function (EB KE, EBL KE)** As for **K+Epsilon**.

K-Epsilon Boundaries Reference

This section details the conditions that are required for each boundary type that is supported by the K-Epsilon turbulence model. In addition, information is provided about the values for each boundary type.

Boundary conditions and values are elected by the boundary type, and/or by methods selected in corresponding nodes in the **Conditions** node.

Note: Boundary types that do not require setting any conditions or values are not listed.

Flow Boundaries

The following boundary conditions and values are equal to all boundaries of type:

- Free Stream
- Mass Flow Inlet
- Pressure Outlet
- Stagnation Inlet
- Velocity Inlet

Turbulence Specification

Controls how you define the turbulence profile at flow boundaries.

The K-Epsilon model requires the turbulent kinetic energy k and the turbulent dissipation rate ε. You can enter the corresponding values directly or have them derived from other turbulence quantities.

Method	Corresponding Value Nodes
K + Epsilon	**Turbulent Kinetic Energy** Scalar profile value to specify k directly. **Turbulent Dissipation Rate** Scalar profile value to specify ε directly. **Elliptic Function (V2F KE)** Scalar profile value to specify the elliptic function f directly. The V2F KE model solves an additional transport equation for this redistribution term in the $\overline{\vartheta^2}$ equation. **Normal Stress Function (V2F KE)** Scalar profile value to specify the normal stress function $\overline{\vartheta^2}$ directly. The V2F KE model solves an additional transport equation for this quantity. **Reduced Stress Function (EB KE, EBL KE)** Scalar profile value to specify the reduced stress function φ directly. The EB KE model solves an additional transport equation for this quantity.
Intensity + Length Scale Calculation of k and ε from a specified turbulence intensity I and length scale l using Eqn. (1957) and Eqn. (1959). Do not use this method if the velocity field is initialized to zero.	**Turbulence Intensity** Scalar profile value to specify I. **Turbulent Length Scale** Scalar profile value to specify l. **Reduced Stress Function (EB KE, EBL KE)** As for **K+Epsilon**.

Intensity + Viscosity Ratio
Calculation of k and ε from a specified turbulence intensity I and viscosity ratio μ_t/μ using

Eqn. (1958) and Eqn. (1960).

Do not use this method if the velocity field is initialized to zero.

Turbulence Intensity

As for **Intensity + Length Scale**.

Turbulent Viscosity Ratio

Scalar profile value to specify the ratio of turbulent to laminar viscosity μ_t/μ.

Reduced Stress Function (EB KE, EBL KE)

As for **K+Epsilon**.

K-Epsilon Regions Reference

This section details the conditions that are required for each region type that is supported by the K-Epsilon turbulence model. In addition, information is provided about the values for each region type.

Region conditions and values are elected by the region type, and/or by methods selected in corresponding nodes in the **Conditions** node.

Note: Region types that do not require setting any conditions or values are not listed.

Fluid Region

Turbulence Source Option

Controls whether you want to use a turbulence source term, as well as what type.

Turbulence Source Option	Corresponding Value Nodes
None	None.
Specified	**Turbulent Dissipation Rate Source** Scalar profile value to specify a turbulence source for ε directly. **Turbulent Kinetic Energy Source** Scalar profile value to specify a turbulence source for k directly.
Ambient	**Ambient Turbulence Specification** Adds a turbulence source to counteract turbulence decay in external aero flows. The source terms are inferred from the specified **Inflow Boundary**. Provided that reasonable inflow turbulence is specified and that flow uniformly enters the domain at this boundary, the sources should not contaminate any boundary layers. The benefits of this feature are: • Easier fine-tuning of turbulence intensity without having to have excessively large turbulent viscosity ratios. This is particularly advantageous for the transition model.

- The ability to properly simulate a plane or car moving through background turbulence.

Porous Region

Turbulence Specification

Controls how you define the turbulence profile within a porous region.

The K-Epsilon model requires the turbulent kinetic energy k and the turbulent dissipation rate ε. You can enter the corresponding values directly or have them derived from other turbulence quantities.

Method	Corresponding Value Nodes
K + Epsilon	**Turbulent Kinetic Energy** Scalar profile value to specify k directly. **Turbulent Dissipation Rate** Scalar profile value to specify ε directly.
Intensity + Length Scale Calculation of k and ε from a specified turbulence intensity I and length scale l using Eqn. (1957) and Eqn. (1959). Do not use this method if the velocity field is initialized to zero.	**Turbulence Intensity** Scalar profile value to specify I. **Turbulent Length Scale** Scalar profile value to specify l.
Intensity + Viscosity Ratio Calculation of k and ε from a specified turbulence intensity I and viscosity ratio μ_t/μ using Eqn. (1958) and Eqn. (1960). Do not use this method if the velocity field is initialized to zero.	**Turbulence Intensity** As for **Intensity + Length Scale**. **Turbulent Viscosity Ratio** Scalar profile value to specify the ratio of turbulent to laminar viscosity μ_t/μ.

K-Epsilon Solvers Reference

This section details the solvers that are associated with a K-Epsilon turbulence model: turbulence and turbulent viscosity.

K-Epsilon Turbulence

This solver controls the solution of the turbulence transport equations in all the continua for which a K-Epsilon model is activated.

For each transported variable the basic steps that are involved in the solution update are as follows:

1. Update boundary conditions.
2. Compute the reconstruction gradients and cell gradients.
3. Create the linear system of equations using discretization methods.

4. Compute the residual sum $R = \sum_{cells} |r|$ for monitoring convergence.
5. Solve the linear system.
6. Update the transported variable field.

Properties

Under-Relaxation Factor

At each iteration, governs the extent to which the newly computed solution supplants the old solution. The default value is 0.8, for EB KE it is 0.6. For the theoretical background, see Eqn. (1625).

Boundary Layer Initialization

When **On**, this computes initial values for turbulent kinetic energy and dissipation rate automatically, taking account of the proximity of walls as well as of free-stream values. This speeds convergence in most cases. (See Boundary Layer Initialization.) The property is **Off** by default.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Under-Relaxation Factor Ramp

The **K-Epsilon Turbulence** > **Under-Relaxation Factor Ramp** node allows you to set and control a linear ramp for under-relaxation. For details, see Under-Relaxation Factor Ramp Refernce.

AMG Linear Solver

The **K-Epsilon Turbulence** > **AMG Linear Solver** node allows you to set the algebraic multigrid parameters for the turbulence solver. For details, see Algebraic Multigrid and AMG Linear Solver Properties.

K-Epsilon Turbulent Viscosity

This solver controls the update of the turbulent viscosity.

Let μ_t^n be the value of the previous iteration, and μ_t^{new} be the value that is computed during the current iteration. The update is controlled as follows:

$$\mu_t^{n+1} = \omega_\mu \mu_t^{\text{new}} + \left(1-\omega_\mu\right)\mu_t^n$$

Properties

Maximum Ratio	The maximum ratio of turbulent to laminar viscosity (μ_t/μ) permitted during the update process. This ratio prevents unphysically high values of turbulent viscosity from occurring on the path to convergence. See Troubleshooting RANS Solvers.
Under-Relaxation Factor	An under-relaxation factor ω_μ for the update of turbulent viscosity. The default value is 1.0.
Solver Frozen	When **On**, this prevents the solver from updating the solution while iterating. The property is **Off** by default.

K-Epsilon Field Functions Reference

This section details the field functions that are made available to the simulation when a K-Epsilon turbulence model is used.

Effective Viscosity

Scalar field that represents the sum of the laminar and turbulent viscosities $\mu + \mu_t$.

Elliptic Blending Function (EB KE, EBL KE)

Scalar field that represents the transported variable α.

Elliptic Function (V2F KE)

Scalar field that represents the transported variable f.

Kolmogorov Length Scale

Scalar field that represents the turbulent length scale η, as defined in Eqn. (2061).

This field is only available when the *Temporary Storage Retained* option is ticked for the **K-Epsilon Turbulence** solver.

Kolmogorov Time Scale

Scalar field that represents the turbulent time scale τ_η, as defined in Eqn. (2062).

This field is only available when the *Temporary Storage Retained* option is ticked for the **K-Epsilon Turbulence** solver.

Normal Stress Function (V2F KE)

Scalar field that represents the transported variable $\overline{\vartheta^2}$.

Reduced Stress Function (EB KE, EBL KE)

Scalar field that represents the transported variable φ.

Reynolds Stresses (SKE, SKE LRe, SKE 2L)

Scalar fields that represent the specific normal and shear stresses:

- **Reynolds Stress uu**
- **Reynolds Stress uv**
- **Reynolds Stress uw**
- **Reynolds Stress vv**
- **Reynolds Stress vw**
- **Reynolds Stress ww**

These fields are only available when a non-linear *Constitutive Relation* is used and when the *Temporary Storage Retained* property is ticked for the **K-Epsilon Turbulence** solver. If the simulation is two-dimensional, only the UU, VV, and UV stresses are exposed. All other stresses are equal to zero.

Strain Rate Tensor Modulus

Scalar field that represents the modulus of the mean strain rate tensor S, as defined in Eqn. (1744).

This field is only available when the **Temporary Storage Retained** option is ticked for the relevant turbulence solver.

Taylor Micro Scale

Scalar field that represents the turbulent length scale λ, as defined in Eqn. (2063).

This field is only available when the *Temporary Storage Retained* option is ticked for the **K-Epsilon Turbulence** solver.

Turb Wall Distance Re (AKN KE, RKE 2L, SKE LRe, SKE 2L)

Scalar field that represents the wall distance Reynolds number Re_d, as defined in Eqn. (1749).

Turbulent Dissipation Rate

Scalar field that represents the transported variable ε.

Turbulent Kinetic Energy

Scalar field that represents the transported variable k.

Turbulent Viscosity

Scalar field that represents the turbulent viscosity μ_t.

Turbulent Viscosity Ratio

Scalar field that represents the ratio of turbulent to laminar viscosity μ_t/μ.

K-Omega Turbulence

The K-Omega turbulence model is a two-equation model that solves transport equations for the turbulent kinetic energy k and the specific dissipation rate ω—the dissipation rate per unit turbulent kinetic energy ($\omega \propto \varepsilon/k$)—in order to determine the turbulent eddy viscosity.

The book by D.C. Wilcox [339] is the most comprehensive reference on the K-Omega model, discussing the origin of the model, comparing it to other models, and presenting the latest version of the model. As the originator of the K-Omega model, Wilcox claims the superiority of his model over the K-Epsilon model, and the superiority of the Omega transport equation over other scale equations.

One reported advantage of the K-Omega model over the K-Epsilon model is its improved performance for boundary layers under adverse pressure gradients. Perhaps the most significant advantage, however, is that it may be applied throughout the boundary layer, including the viscous-dominated region, without further modification. Furthermore, the standard K-Omega model can be used in this mode without requiring the computation of wall distance.

The biggest disadvantage of the K-Omega model, in its original form, is that boundary layer computations are sensitive to the values of ω in the free-stream. This translates into extreme sensitivity to inlet boundary conditions for internal flows, a problem that does not exist for the K-Epsilon models. The K-Omega model variants included in STAR-CCM+ have been modified in an attempt to address this shortcoming.

K-Omega Model Variants

The following variants of the K-Omega model are implemented in STAR-CCM+:

Model Variants
Standard K-Omega
SST K-Omega
SST K-Omega Detached Eddy

Standard K-Omega

Wilcox revised his original model in 1998, and then in 2006, to account for several perceived deficiencies in the original version (1988). These revisions include:

- A revised set of model coefficients
- Two corrections to account for sensitivity to free-stream/inlet conditions, both based on products of ∇k and $\nabla \omega$. The first model, introduced in 1998, is a modification of the turbulent kinetic energy equation. The more recent revision [340] is the introduction of a cross-diffusion term that is similar to that used in the SST K-Omega model.
- A correction to improve the free-shear-flow spreading rates of the model
- A compressibility correction
- Low Reynolds number corrections that allow the K-Omega model to be better applied in the prediction of low Reynolds number and transitional flows

However, the validation results published in Wilcox's book are typically for two-dimensional, primarily parabolic, flows. Until further validations for complex flows are widely published, the corrections should be used with caution. Therefore, each correction is included as an option in STAR-CCM+.

To regain the 1988 form of the standard K-Omega model, all these corrections should be omitted. To use the most recent updates of the Standard K-Omega model [340], select the Cross-diffusion Limiter in the Free-Shear Modification, activate the Vortex-Stretching Modification, and set the Realizability Option to Durbin Scale Limiter.

SST K-Omega Model

The problem of sensitivity to free-stream/inlet conditions was addressed by Menter [334], who recognized that the ε transport equation from the Standard K-Epsilon model could be transformed into an ω transport equation by variable substitution.

The transformed equation looks similar to the one in the Standard K-Omega model, but adds an additional non-conservative cross-diffusion term containing the dot product $\nabla k \cdot \nabla \omega$. Inclusion of this term in the ω transport equation potentially makes the K-Omega model give identical results to the K-Epsilon model. Menter suggested using a blending function (which includes functions of wall distance) that would include the cross-diffusion term far from walls, but not near the wall. This approach effectively blends a K-Epsilon model in the far-field with a K-Omega model near the wall. Purists may object that the blending function crossover location is arbitrary, and could obscure some critical feature of the turbulence. Nevertheless, the fact remains that this approach cures the biggest drawback to applying the K-Omega model to practical flow simulations.

Menter also introduced a modification to the linear constitutive equation and named the model containing this modification the SST (shear-stress transport) K-Omega model. However, the linear relation between the Reynolds stresses and the mean strain rate tends to strongly under predict the anisotropy of turbulence. Turbulence is anisotropic in most complex flows—for example in strong swirl, streamline curvature, shear layer, or boundary layer flows. The anisotropy of the Reynolds stresses not only affects the flow field but also the turbulent transport of scalars (temperature, concentration, passive scalar).

To account for anisotropy of turbulence, STAR-CCM+ offers the following non-linear constitutive relations:

- Quadratic—as suggested by Spalart [336].
- Cubic—as proposed by Wallin and Johansson [338] and further improved by Hellsten [333]

The cubic constitutive relation is derived from a Reynolds stress transport model and, thus, represents an Explicit Algebraic Reynolds Stress Model (EARSM). Reynolds stress transport models perform much better than eddy viscosity models in anisotropic turbulence, but are often unstable when used in complex flows. Therefore, the cubic model is a good compromise between the two.

The SST model has seen fairly wide application in the aerospace industry, where viscous flows are typically resolved, and turbulence models are applied throughout the boundary layer.

SST K-Omega Detached Eddy Model

See SST K-Omega Detached Eddy Model.

Contents:

Deciding on a K-Omega Model
Controlling a K-Omega Model
K-Omega Models Reference
K-Omega Initial Conditions Reference
K-Omega Boundaries Reference
K-Omega Regions Reference
K-Omega Solvers Reference
K-Omega Field Functions Reference

Deciding on a K-Omega Model

Decide about free-stream conditions when choosing a K-Omega model.

The Standard K-Omega model in its original form is sensitive to the levels of turbulence in the free stream. Wilcox has attempted to correct this and other limitations of the model by including several corrections in a recent revision. These corrections are incorporated into STAR-CCM+. However, they need to be used with caution until they are properly validated for complex flow situations.

Menter's SST K-Omega model uses the insensitivity to free-stream conditions of the K-Epsilon model in the far-field, while retaining the advantages of the K-Omega model near walls.

Both K-Omega turbulence models are available with all three wall treatments: high-y^+, low-y^+ and all-y^+. For most applications, the recommended choice of models is the SST K-Omega model combined with the all-y^+ wall treatment. It is recommended that only the default parameters be used with the SST K-Omega model.

Controlling a K-Omega Model

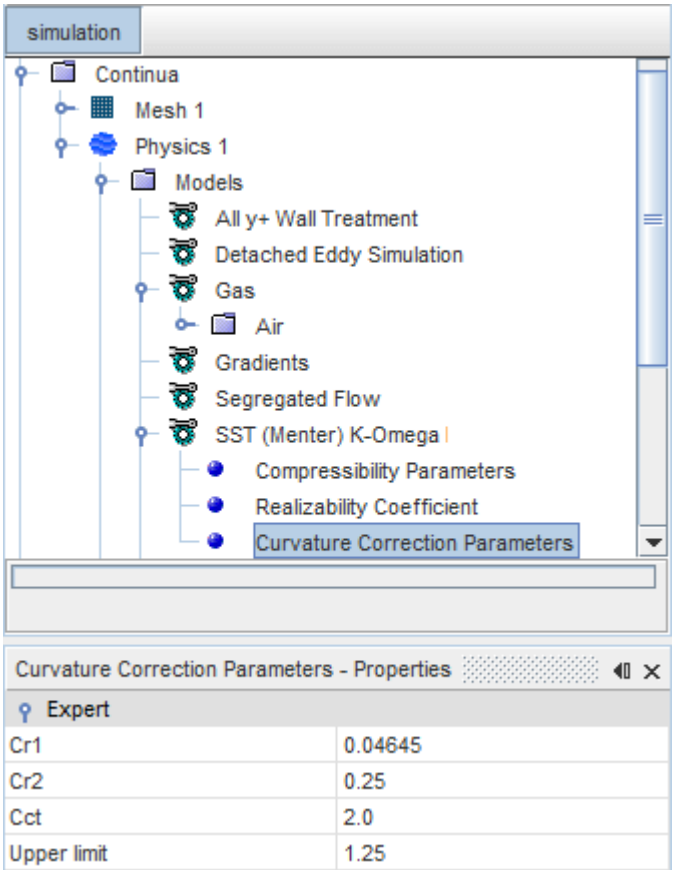
This section provides guidelines for controlling simulations with K-Omega turbulence models:

- Accounting for Strong Streamline Curvature and Frame-Rotation
- Overcoming an Unexpectedly Large Growth of K
- Accounting for Low Reynolds Number Effects
- Overcoming the Round/Plane-Jet Anomaly
- Accounting for Compressibility Effects
- Improving Predictions for Free-Shear Flows
- Accounting for Anisotropy of Turbulence

Accounting for Strong Streamline Curvature and Frame-Rotation

Curvature correction is a factor that alters the turbulent energy production term according to local rotation and vorticity rates.

The **Curvature Correction Parameters** node provides the properties with which you can control the curvature correction factor. It is activated as a sub-node of any K-Omega model node when the **Curvature Correction Option** property is activated.

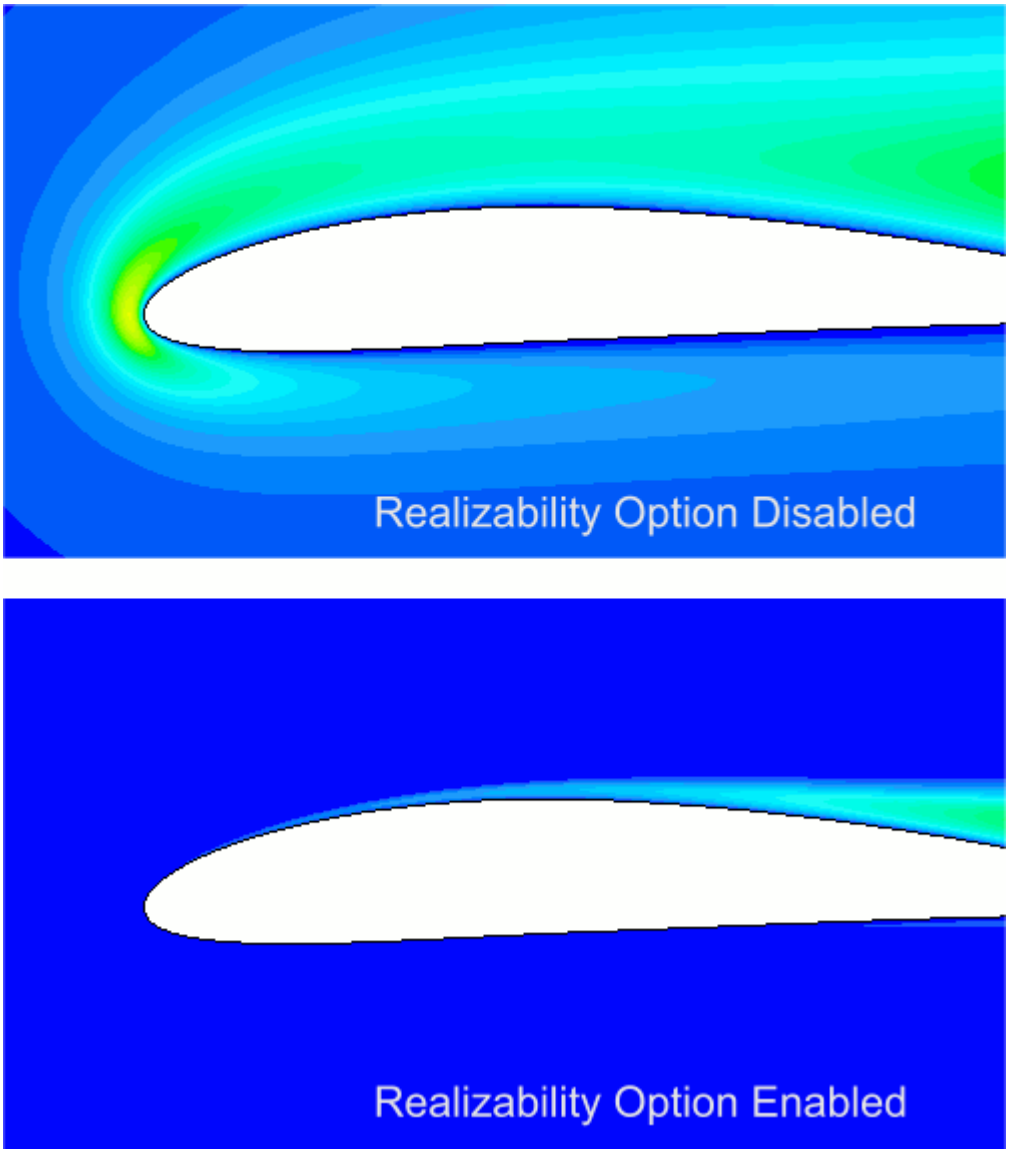


This feature is deactivated by default. It is only necessary for cases that have strong streamline curvature or a rapidly rotating reference frame. Only activate this feature to improve the results when you have a stable solution.

Overcoming an Unexpectedly Large Growth of K

It has been observed [332] that two-equation models predict an unexpectedly large growth of turbulent kinetic energy in stagnation point flows. This issue can have an adverse effect on the rest of the flow solution. To overcome this issue, a lower limit on the turbulence time scale has been proposed that imposes the “realizability” constraint on the eddy-viscosity formula. The *Realizability Option* property of the parent model object is set to **Durbin Scale Limiter**.

The effect of this coefficient can be demonstrated by observing the difference between two solutions of flow over an airfoil. In the first solution that is shown below, the realizability option has not been activated. The build-up of turbulence at the stagnation point is clearly observed. In the second solution, the realizability option has been activated.

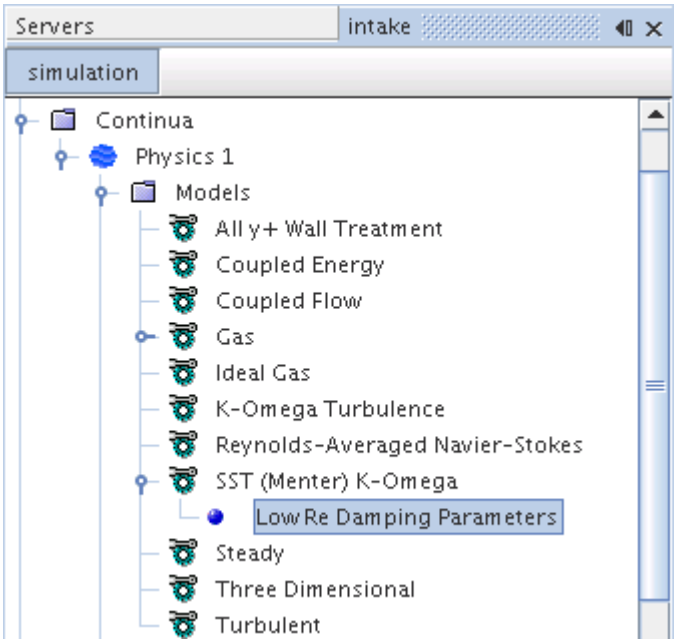


This time scale realizability constraint is applied by default to the K-Omega turbulence models in STAR-CCM+.

Accounting for Low Reynolds Number Effects

The *Low Re Damping Modification* property of a K-Omega model is needed for low Reynolds number flows, or when it is important to predict turbulent quantities close to surfaces.

The **Low Re Damping Parameters** sub-node provides the properties with which you can control the damping modification.



This feature is deactivated by default. It is not needed for low-y^+ wall treatment, only for “transitional flows”. Although further development is required to make it reliable as a transition model, it can handle laminar bubble transition to some degree. It is not compatible with the Gamma ReTheta Transition model.

Overcoming the Round/Plane-Jet Anomaly

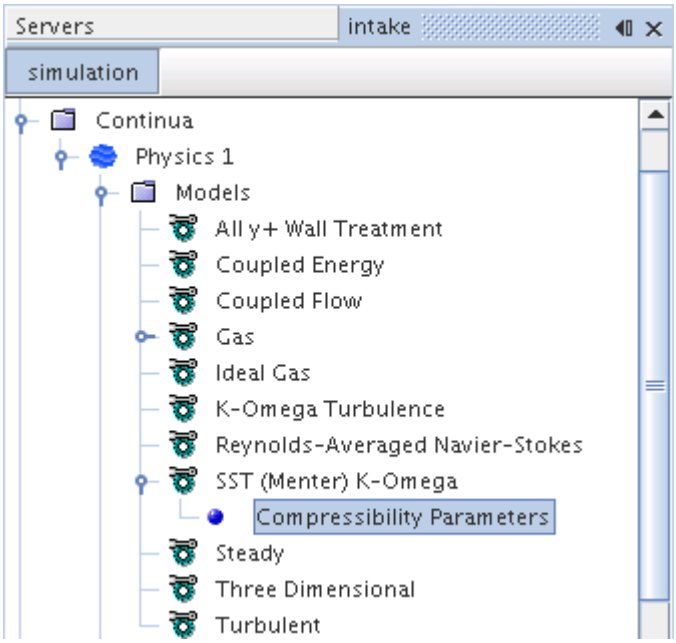
Almost all two-equation models tend to predict a larger spreading rate for round jet than for plane jets, while the opposite effect is observed in experiments. The *Vortex Stretching Modification* property of the Standard (Wilcox) K-Omega model allows you to remedy this problem. This model is suitable for most cases where flow separates from a surface.

This feature is deactivated by default.

Accounting for Compressibility Effects

The compressibility correction that is implemented for the K-Omega models allows you to account for dilatation dissipation and enhances the prediction of flows where compressible effects are dominant.

The **Compressibility Parameters** node provides the properties with which you can control the compressibility correction. It is activated as a sub-node of any K-Omega model node when the *Compressibility Correction* property is activated.



Improving Predictions for Free-Shear Flows

The free-shear modification improves predictions for flows that are dominated by free-shear layers (like jets and mixing layers) or when free-stream turbulence is high.

The **Free-Shear Modification** property is available for the Standard (Wilcox) K-Omega model and is deactivated by default. The following modification methods are available:

- **Dissipation Limiter**
- **Cross-diffusion Limiter**

Accounting for Anisotropy of Turbulence

You can account for anisotropy of turbulence by extending the Boussinesq approximation to include non-linear functions of the strain and vorticity-rate tensors. Using non-linear constitutive relations improves turbulence prediction for flows with strong secondary recirculation, streamline curvature, or where highly non-equilibrium conditions are prevalent.

The *Constitutive Option* property is available for the SST (Menter) K-Omega model. Two non-linear relations are available:

- **QCR** (Quadratic Constitutive Relation)
- **Cubic**

The Quadratic relation improves the performance of the SST (Menter) K-Omega model by predicting secondary flows better.

The Cubic relation extends the non-linear constitutive relation of the Quadratic relation by including cubic functions of the velocity gradients. These additional terms improve the sensitivity of the SST (Menter) K-Omega Model to streamline curvature.

K-Omega Models Reference

The **K-Omega Turbulence** models allow you to compute the turbulent kinetic energy k and the specific dissipation rate ω to provide closure to the Reynolds-Averaged Navier-Stokes equations.

Models Overview

Model Names and Abbreviations	**SST (Menter) K-Omega*** (Shear-Stress Transport (Menter) K-Omega)		**SST M KO**
	Standard (Wilcox) K-Omega		**ST W KO**
Theory	See Theory Guide—K-Omega Model.		
Provided By	**[physics continuum]** > **Models** > *K-Omega Turbulence Models*		
Example Node Path	**Continua** > **Physics 1** > **Models** > **SST (Menter) K-Omega**		
Requires	*Space*: **Axisymmetric**, **Two Dimensional**, or **Three Dimensional** *Time*: **Steady**, **Implicit Unsteady**, **PISO Unsteady**, or **Harmonic Balance** *Material*: **Gas**, **Liquid**, **Eulerian Multiphase**, **Multi-Component Gas**, or **Multi-Component Liquid** *Viscous Regime*: **Turbulent** *Turbulence*: **Reynolds-Averaged Navier-Stokes** *Reynolds-Averaged Turbulence*: **K-Omega Turbulence**		
Properties	See Properties Lookup.		
Activates	Physics Models	*Wall Distance*: **Exact Wall Distance** and **PDE Wall Distance**	
		K-Omega Wall Treatment: **All y+ Wall Treatment**, **High y+ Wall Treatment**, and **Low y+ Wall Treatment**	
		Optional Models: **Turbulent Viscosity User Scaling**	
	Model Controls (child nodes)	**Compressibility Parameters** (optional, see Compressibility Correction) **Realizability Coefficient** (optional, see Realizability Option) **Curvature Correction Parameters** (optional, see Curvature Correction Option) **Low Re Damping Parameters** (optional, see Low Re Damping Modification)	
	Initial Conditions	See K-Omega Initial Conditions Reference.	
	Boundary Inputs	See K-Omega Boundaries Reference.	
	Region Inputs	See K-Omega Regions Reference.	
	Solvers	**K-Omega Turbulence** **K-Omega Turbulent Viscosity** See K-Omega Solvers Reference.	
	Monitors	**Sdr** (specific dissipation rate)	

		Tke (turbulent kinetic energy)
	Field Functions	**Effective Viscosity** **Kolmogorov Length Scale** **Kolmogorov Time Scale** **Specific Dissipation Rate** **Strain Rate Tensor Modulus** **Taylor Micro Scale** **Turbulent Kinetic Energy** **Turbulent Viscosity** **Turbulent Viscosity Ratio** **Turbulent Dissipation Rate** See K-Omega Field Functions Reference.

Properties Lookup

This table shows which properties are used by which **K-Omega Turbulence** model. Use the abbreviations given in the Models Overview above.

	SST M KO	ST W KO
a1 The coefficient a_1 in Eqn. (1820) and Eqn. (1821).	✔	
Alpha The coefficient α in Eqn. (1828).		✔
Beta The coefficient β in Eqn. (1824).		✔
Beta1 The coefficient β_1 in Eqn. (1832) and Eqn. (1835).	✔	
Beta2 The coefficient β_2 in Eqn. (1832) and Eqn. (1836).	✔	
BetaStar The coefficient β^* in Eqn. (1822) and Eqn. (1823).	✔	✔
Compressibility Correction Controls whether to apply a compressibility correction to account for dilatation dissipation.	✔	✔

	SST M KO	ST W KO
This correction is irrelevant for incompressible flows. **Compressibility Parameters** Child node that provides the parameter *Zeta_Star* to modify the model coefficients in Eqn. (1870) and Eqn. (1871). For guidelines when to apply a compressibility correction, see Accounting for Compressibility Effects.		
Constitutive Option Controls the type of constitutive relation used. Consider applying a non-linear relation in the presence of anistropic turbulence. See Accounting for Anisotropy of Turbulence. • **Linear**: Selects the linear constitutive relation as implied by the Boussinesq approximation (see Eqn. (1759). • **QCR**: Selects the quadratic constitutive relation (Eqn. (1850)). • **Cubic:** Selects the cubic constitutive relation (see Eqn. (1851)).	✔	
Convection Controls the convection scheme. • **1st-order**: Selects the first-order upwind convection scheme. • **2nd-order**: Selects the second-order upwind convection scheme.	✔	✔
Curvature Correction Option Controls whether to apply a curvature correction, that supplies the effects of strong (streamline) curvature and frame-rotation. When **On**, a curvature correction factor f_c is applied, that alters the turbulent kinetic energy production term according to local rotation and vorticity rates (see Eqn. (1825)). Only use this option when you have a stable solution, to improve the results. **Curvature Correction Parameters** Child node that provides the parameters *Cr1*, *Cr2*, *Cct* and *Upper Limit* (C_{max}) to calculate f_c using Eqn. (1912). For guidelines when to apply a curvature correction, see Accounting for Strong Streamline Curvature and Frame-Rotation.	✔	✔
Free-Shear Modification Controls the method to use for the free-shear modification.		✔

	SST M KO	ST W KO
• **None**: The modification is not included. • **Dissipation Limiter**: The modifications detailed in Eqn. (1840). • **Cross-diffusion Limiter**: The modification detailed in Eqn. (1841). For guidelines when to apply a free-shear modification, see Improving Predictions for Free-Shear Flows.		
Kappa The coefficient κ , see Eqn. (1835) and Eqn. (1836).	✔	
Low Re Damping Modification When **On**, the low Reynolds number modification is incorporated. **LowRe Damping Parameters** Child node that provides the parameters *RBeta* (Re_{β}), *Rk* (Re_{k}), and *Rw* (Re_{ω}) to modify the model coefficients in Eqn. (1844) to Eqn. (1849). For guidelines when to apply a low Reynolds number modification, see Accounting for Low Reynolds Number Effects.	✔	✔
Normal Stress Term This property is an explicit term that directly incorporates divergence and turbulent kinetic energy, $-2/3(\mu_t \nabla \cdot \mathbf{v} + \rho k)\mathbf{I}$, according to the full Boussinesq approximation. When **On**, the Reynolds stress tensor is modeled as $\mathbf{T}_t = 2\mu_t \mathbf{S} - 2/3(\mu_t \nabla \cdot \mathbf{v} + \rho k)\mathbf{I}$ and the turbulent production is modeled using $G_k = \mu_t S^2 - \frac{2}{3}\rho k \nabla \cdot \overline{\mathbf{v}} - \frac{2}{3}\mu_t (\nabla \cdot \overline{\mathbf{v}})^2$ When **Off**, the Reynolds stress tensor is modeled as $\mathbf{T}_t = 2\mu_t \mathbf{S}$ and the turbulent production is assumed to be $G_k = \mu_t S^2$ This property is off by default, in which case the quantity $-2/3(\mu_t \nabla \cdot \mathbf{v} + \rho k)\mathbf{I}$ is absorbed into the pressure, and causes the pressure to be slightly different. However, the results are the same even if this term is not explicitly included. In incompressible flow only the gradients of pressure matter, so this setting has no effect on the results. In compressible flow, however, the absolute value of	✔	✔

	SST M KO	ST W KO
pressure is used in the Ideal Gas Law (Eqn. (1466)). Activating this option may affect the solution convergence if k happens to be poorly initialized.		
Realizability Option Controls whether to activate Durbin's realizability constraint on the turbulent time scale. When set to **Durbin Scale Limiter**, a minimum constraint on the turbulent time scale is applied. **Realizability Coefficient** Child node that provides the *Realizability Coefficient* (C_T) to calculate the turbulent time scale using Eqn. (1819) (ST W KO) and Eqn. (1821) (SST M KO). For guidelines when to activate Durbin's realizability constraint, see Overcoming an Unexpectedly Large Growth of K.	✔	✔
Secondary Gradients Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces. • **On**: Include both secondary gradients. • **Off**: Exclude both secondary gradients. • **Interior Only**: Include the interior secondary gradients only. • **Boundaries Only**: Include the boundary secondary gradients only.	✔	✔
Sdr Minimum The minimum value that the transported variable ω is permitted to have. An appropriate value is a small number that is greater than the floating point minimum of the computer.	✔	✔
Sigma_k The coefficient σ_k in Eqn. (1823).		✔
Sigma_k1 The coefficient σ_{k1} in Eqn. (1837).	✔	
Sigma_k2 The coefficient σ_{k2} in Eqn. (1837).	✔	

	SST M KO	ST W KO
Sigma_w The coefficient σ_ω in Eqn. (1824).		✔
Sigma_w1 The coefficient $\sigma_{\omega 1}$ in Eqn. (1838).	✔	
Sigma_w2 The coefficient $\sigma_{\omega 2}$ in Eqn. (1838) and Eqn. (1830).	✔	
TkeMinimum The minimum value that the transported variable k is permitted to have. An appropriate value is a small number that is greater than the floating point minimum of the computer.	✔	✔
Vortex Stretching Modification When **On**, the vortex modification as detailed in Eqn. (1842) through Eqn. (1843) is incorporated. For guidelines when to activate the vortex modification, see Overcoming the Round/Plane-Jet Anomaly.		✔

K-Omega Initial Conditions Reference

This section details the conditions and values that are required for initialization of a K-Omega turbulence model.

Turbulence Specification

Controls how you define the turbulence profile for initialization.

The K-Omega model requires the turbulent kinetic energy k and the specific dissipation rate ω. You can enter the corresponding values directly or have them derived from other turbulence quantities.

Method	Corresponding Value Nodes
K + Omega	**Turbulent Kinetic Energy** Scalar profile value to specify k directly. **Specific Dissipation Rate** Scalar profile value to specify ω directly.
Intensity + Length Scale Calculation of k and ω from a specified turbulence intensity I, length scale l, and	**Turbulence Intensity** Scalar profile value to specify I.

velocity scale v using Eqn. (1957) and Eqn. (1963). Do not use this method if the velocity field is initialized to zero.	**Turbulent Length Scale** Scalar profile value to specify l. **Turbulent Velocity Scale** Scalar profile value to specify v. For an initial value, use a representative velocity or a representative velocity scale. For example, for a pipe flow use the inlet velocity.
Intensity + Viscosity Ratio Calculation of k and ω from a specified turbulence intensity I, viscosity ratio μ_t/μ, and velocity scale v using Eqn. (1958) and Eqn. (1964). Do not use this method if the velocity field is initialized to zero.	**Turbulence Intensity** As for **Intensity + Length Scale**. **Turbulent Viscosity Ratio** Scalar profile value to specify the ratio of turbulent to laminar viscosity μ_t/μ. **Turbulent Velocity Scale** Scalar profile value to specify v. For an initial value, use a representative velocity or a representative velocity scale. For example, for a pipe flow use the inlet velocity.

K-Omega Boundaries Reference

This section details the conditions that are required for each boundary type that is supported by the K-Omega turbulence model. In addition, information is provided about the values for each boundary type.

Boundary conditions and values are elected by the boundary type, and/or by methods selected in corresponding nodes in the **Conditions** node.

Note: Boundary types that do not require setting any conditions or values are not listed.

Flow Boundaries

The following boundary conditions and values are equal to all flow boundaries of type:

- Free Stream
- Mass Flow Inlet
- Pressure Outlet
- Stagnation Inlet
- Velocity Inlet

Turbulence Specification

Controls how you define the turbulence profile.

The K-Omega model requires the turbulent kinetic energy k and the specific dissipation rate ω. You can enter the corresponding values directly or have them derived from other turbulence quantities.

Method	Corresponding Value Nodes
K + Omega	**Turbulent Kinetic Energy** Scalar profile value to specify k directly. **Specific Dissipation Rate** Scalar profile value to specify ω directly.
Intensity + Length Scale Calculation of k and ω from a specified turbulence intensity I and length scale l using Eqn. (1957) and Eqn. (1963). Do not use this method if the velocity field is initialized to zero.	**Turbulence Intensity** Scalar profile value to specify I. **Turbulent Length Scale** Scalar profile value to specify l.
Intensity + Viscosity Ratio Calculation of k and ω from a specified turbulence intensity I and viscosity ratio μ_t/μ using Eqn. (1958) and Eqn. (1964). Do not use this method if the velocity field is initialized to zero.	**Turbulence Intensity** As for **Intensity + Length Scale**. **Turbulent Viscosity Ratio** Scalar profile value to specify the ratio of turbulent to laminar viscosity μ_t/μ.

K-Omega Regions Reference

This section details the conditions that are required for each region type that is supported by the K-Omega turbulence model. In addition, information is provided about the values for each region type.

Region conditions and values are elected by the region type, and/or by methods selected in corresponding nodes in the **Conditions** node.

Note: Region types that do not require setting any conditions or values are not listed.

Fluid Region

Turbulence Source Option

Controls whether you want to use a turbulence source term, as well as what type.

Turbulence Source Option	Corresponding Value Nodes
None	None.
Specified	**Specific Dissipation Rate Source** Scalar profile value to specify a turbulence source for ω directly. **Turbulent Kinetic Energy Source** Scalar profile value to specify a turbulence source for k directly.

Ambient	**Ambient Turbulence Specification** Adds a turbulence source to counteract turbulence decay in external aero flows. The source terms are inferred from the specified **Inflow Boundary**. Provided that reasonable inflow turbulence is specified and that flow uniformly enters the domain at this boundary, the sources should not contaminate any boundary layers. The benefits of this feature are: • Easier fine-tuning of turbulence intensity without having to have excessively large turbulent viscosity ratios. This is particularly advantageous for the transition model. • The ability to properly simulate a plane or car moving through background turbulence.

Porous Region

Turbulence Specification

Controls how you define the turbulence profile.

The K-Omega model requires the turbulent kinetic energy k and the specific dissipation rate ω. You can enter the corresponding values directly or have them derived from other turbulence quantities.

Method	Corresponding Value Nodes
K + Omega	**Turbulent Kinetic Energy** Scalar profile value to specify k directly. **Specific Dissipation Rate** Scalar profile value to specify ω directly.
Intensity + Length Scale Calculation of k and ω from a specified turbulence intensity I and length scale l using Eqn. (1957) and Eqn. (1963). Do not use this method if the velocity field is initialized to zero.	**Turbulence Intensity** Scalar profile value to specify I. **Turbulent Length Scale** Scalar profile value to specify l.
Intensity + Viscosity Ratio Calculation of k and ω from a specified turbulence intensity I and viscosity ratio μ_t/μ using Eqn. (1958) and Eqn. (1964). Do not use this method if the velocity field is initialized to zero.	**Turbulence Intensity** As for **Intensity + Length Scale**. **Turbulent Viscosity Ratio** Scalar profile value to specify the ratio of turbulent to laminar viscosity μ_t/μ.

K-Omega Solvers Reference

This section details the solvers that are associated with the K-Omega turbulence models: turbulence and turbulent viscosity.

K-Omega Turbulence

This solver controls the solution of the turbulence transport equations in all the continua for which a K-Omega model is activated.

For each transported variable k and ω the basic steps that are involved in the solution update are as follows:

- Update boundary conditions.
- Compute the reconstruction gradients and cell gradients.
- Create the linear system of equations (see Eqn. (1625)) using the discretization methods that are described in the K-Omega model formulation.
- Compute the residual sum $R = \sum_{cells} |r|$ for monitoring convergence.
- Solve the linear system.
- Update the transported variable field.

Properties

Under-Relaxation Factor

At each iteration, governs the extent to which the old solution is supplanted by the newly computed solution. The default value is 0.8. For the theoretical background, see Eqn. (1625).

Boundary Layer Initialization

When **On**, this computes initial values for turbulent kinetic energy and specific dissipation rate automatically, taking account of the proximity of walls as well as free-stream values. This speeds convergence in most cases. (See Eqn. (1970) to Eqn. (1973)) The property is **Off** by default.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Under-Relaxation Factor Ramp

The **K-Omega Turbulence** > **Under-Relaxation Factor Ramp** node allows you to set and control a linear ramp for under-relaxation. For details, see Under-Relaxation Factor Ramp Reference.

AMG Linear Solver

The **K-Omega Turbulence** > **AMG Linear Solver** node allows you to set the algebraic multigrid parameters for the turbulence solver. For details, see Algebraic Multigrid and AMG Linear Solver Properties.

K-Omega Turbulent Viscosity

This solver controls the update of the turbulent viscosity.

Let μ_t^n be the value of the previous iteration, and μ_t^{new} be the value that is computed during the current iteration. The update is controlled as follows:

$$\mu_t^{n+1} = \omega_\mu \mu_t^{\text{new}} + \left(1-\omega_\mu\right)\mu_t^n$$

Properties

Maximum Ratio	The maximum ratio of turbulent to laminar viscosity (μ_t/μ) permitted during the update process. This is used to prevent unphysically high values of turbulent viscosity from occurring on the path to convergence. See Troubleshooting RANS Solvers.
Under-Relaxation Factor	An under-relaxation factor ω_μ for the update of turbulent viscosity. The default value is 1.0.
Solver Frozen	When **On**, this prevents the solver from updating the solution while iterating or responding to other solvers. The property is **Off** by default.

K-Omega Field Functions Reference

This section details the field functions that are made available to the simulation when a K-Omega turbulence model is used.

Effective Viscosity

Scalar field that represents the sum of the laminar and turbulent viscosities $\mu + \mu_t$.

Kolmogorov Length Scale

Scalar field that represents the turbulent length scale η, as defined in Eqn. (2066).

This field is only available when the *Temporary Storage Retained* option is ticked for the **K-Omega Turbulence** solver.

Kolmogorov Time Scale

Scalar field that represents the turbulent time scale τ_η, as defined in Eqn. (2067).

This field is only available when the *Temporary Storage Retained* option is ticked for the **K-Omega Turbulence** solver.

Reynolds Stresses (SST M KO)

Scalar fields that represent the specific normal and shear stresses:

- **Reynolds Stress uu**
- **Reynolds Stress uv**
- **Reynolds Stress uw**
- **Reynolds Stress vv**
- **Reynolds Stress vw**
- **Reynolds Stress ww**

These fields are only available when a non-linear *Constitutive Option* is used and when the *Temporary Storage Retained* property is activated for the **K-Omega Turbulence** solver. If the simulation is two-dimensional, only the UU, VV, UV and WW stresses are exposed. All other stresses are equal to zero.

Specific Dissipation Rate

Scalar field that represents the transported variable ω.

Strain Rate Tensor Modulus

Scalar field that represents the modulus of the mean strain rate tensor S, as defined in Eqn. (1744).

This field is only available when the **Temporary Storage Retained** option is ticked for the relevant turbulence solver.

Taylor Micro Scale

Scalar field that represents the turbulent length scale λ, as defined in Eqn. (2068).

This field is only available when the *Temporary Storage Retained* option is ticked for the **K-Omega Turbulence** solver.

Turbulent Dissipation Rate

Scalar field that represents the dissipation rate in the turbulent kinetic energy transport equation. It is defined as $\beta^* f_{\beta^*} \omega k$ as in Eqn. (1823).

This field is only available when the *Temporary Storage Retained* option is ticked for the relevant turbulence solver.

Turbulent Kinetic Energy

Scalar field that represents the transported variable k.

Turbulent Viscosity

Scalar field that represents the turbulent viscosity μ_t.

Turbulent Viscosity Ratio

Scalar field that represents the ratio of turbulent to laminar viscosity μ_t/μ.

Reynolds Stress Transport (RST) Turbulence

Reynolds Stress Transport (RST) models, also known as second-moment closure models, directly calculate the components of the specific Reynolds stress tensor $\mathbf{R} = -\mathbf{T}_t/\rho$ by solving their governing transport equations.

RST models have the potential to predict complex flows more accurately than eddy viscosity models because the transport equations for the Reynolds stresses naturally account for the effects of turbulence anisotropy, streamline curvature, swirl rotation and high strain rates.

The starting point for the development of an RST model is generally the exact differential transport equation for the Reynolds stresses, which is derived by multiplying the instantaneous Navier-Stokes equations by a fluctuating property and Reynolds-averaging their product (see [357] for this exact equation). In the resulting equations, the transient, convective and molecular diffusion terms do not require modeling. The terms remaining to be modeled are the turbulent diffusion term, the dissipation term and, perhaps the greatest challenge, the pressure-strain term. Appropriate models for these terms have received much attention during the past few decades, some of which are reviewed in [358].

Note: A RST model carries significant computational overhead. Seven equations must be solved (as opposed to the two equations of a K-Epsilon or a K-Omega model): six equations for the Reynolds stresses (symmetric tensor) and one equation for the isotropic turbulent dissipation ε. Apart from the additional memory and computational time required for these equations to be solved, there is also likely to be a penalty in the total number of iterations required to obtain a converged solution due to the numerical stiffness of the RST equations.

Pressure Strain Models

To model the pressure-strain term, the following approaches are implemented in STAR-CCM+:

- Linear Pressure Strain
- Quadratic Pressure Strain

The Linear Pressure-Strain model can be used with a high-y^+ wall treatment, but it is also available with a two-layer formulation, which makes it applicable right down to the wall.

The Quadratic Pressure-Strain model and its low Reynolds number variant, the Elliptic Blending model, are based on the most recent and precise formulation of the pressure-strain term. Thus, they are likely to be the models of choice in a majority of cases, the Elliptic Blending model being preferred as it also contains specific treatments for all-y^+ meshes. The Quadratic Pressure-Strain model can only be used with a high-y^+ wall treatment (that is, using wall functions) without resolving the viscous-affected near-wall region.

Linear Pressure Strain

The linear model of Gibson and Launder [352] for the pressure-strain term comprises five terms; these are the rapid term, the slow (return-to-isotropy) term, and their respective wall-reflection terms.

Linear Pressure Strain Two-Layer

The advantage of the linear pressure-strain model is that it lends itself to being incorporated into a two-layer formulation, which can be used to resolve the viscous sublayer for low Reynolds number type applications.

In the linear pressure-strain model approach, suggested by Rodi [324], the computation is divided into two layers. In the layer adjacent to the wall, the turbulent dissipation rate ε and the turbulent viscosity μ_t are specified as functions of wall distance. The values of ε specified in the near-wall layer are blended smoothly with the values computed from solving the transport equation far from the wall.

Quadratic Pressure Strain

The quadratic pressure-strain model of Sarkar, Speziale and Gatski [358] uses a higher-order truncation of the expansion for the pressure-strain term.

Elliptic Blending

The Elliptic Blending RST model of Manceau and Hanjalić [345] is a low Reynolds number model that is based on a inhomogeneous near-wall formulation of the quasi-linear quadratic pressure-strain term. The blending function is used to blend the viscous sub-layer with the log-layer formulation of the pressure-strain term. This approach requires the solution of an elliptic equation for the blending parameter α. The version of the model implemented in STAR-CCM+ was revised by Lardeau and Manceau [354].

Contents:

Reynolds Stress Models Reference
Reynolds Stress Initial Conditions Reference
Reynolds Stress Boundaries Reference
Reynolds Stress Regions Reference
Reynolds Stress Solvers Reference
Reynolds Stress Field Functions Reference

Reynolds Stress Models Reference

The **Reynolds Stress Turbulence** models allow you to compute the components of the specific Reynolds stress tensor $\mathbf{R}$ to provide closure to the Reynolds-Averaged Navier-Stokes equations.

Models Overview

Model Names and Abbreviations	**Elliptic Blending**		**EB**
	Linear Pressure Strain		**LPS**
	Linear Pressure Strain Two-Layer		**LPS 2L**
	Quadratic Pressure Strain		**QPS**
Theory	See Theory Guide—Reynolds Stress Transport Models.		
Provided By	**[physics continuum]** > **Models** > *Reynolds Stress Turbulence Models*		
Example Node Path	**Continua** > **Physics 1** > **Models** > **Linear Pressure Strain Two-Layer**		
Requires	Physics Models	*Space*: **Axisymmetric**, **Two Dimensional**, or **Three Dimensional** *Time*: **Steady** or **Implicit Unsteady** *Material*: **Gas**, **Liquid**, **Eulerian Multiphase**, **Multi-Component Gas**, or **Multi-Component Liquid**	

		Viscous Regime: **Turbulent** *Turbulence*: **Reynolds-Averaged Navier-Stokes** *Reynolds-Averaged Turbulence*: **Reynolds Stress Turbulence**
Properties	See Properties Lookup.	
Activates	Physics Models	*Wall Distance*: **Exact Wall Distance** and **PDE Wall Distance**
		Wall Treatment: **All y+ Wall Treatment** (EB), **High y+ Wall Treatment** (LPS, QPS), or **Two-Layer All y+ Wall Treatment** (LPS 2L)
	Model Controls (child nodes)	**C3e** (optional, see Buoyancy Production of Dissipation)
	Initial Conditions	See Reynolds Stress Initial Conditions Reference.
	Boundary Inputs	See Reynolds Stress Boundaries Reference.
	Region Inputs	See Reynolds Stress Region Reference.
	Solvers	**Reynolds Stress Turbulence** (LPS, LPS 2L, QPS) **Elliptic Reynolds Stress Turbulence** (EB) **Reynolds Stress Turbulent Viscosity** See Reynolds Stress Solvers Reference.
	Monitors	**Tdr** (turbulent dissipation rate) **uu-stress** (UU Reynolds stress component) **uv-stress** (UV Reynolds stress component) **uw-stress** (UW Reynolds stress component) **vv-stress** (VV Reynolds stress component) **vw-stress** (VW Reynolds stress component) **ww-stress** (WW Reynolds stress component)
	Field Functions	**Kolmogorov Length Scale** **Kolmogorov Time Scale** **Reynolds Stress uu** **Reynolds Stress uv** **Reynolds Stress uw** **Reynolds Stress vv** **Reynolds Stress vw** **Reynolds Stress ww** **Strain Rate Tensor Modulus** **Taylor Micro Scale** **Turbulent Dissipation Rate** **Turbulent Kinetic Energy** **Turbulent Viscosity** **Turbulent Viscosity Ratio** See Reynolds Stress Field Functions Reference.

Properties Lookup

This table shows which properties are used by which **Reynolds Stress Turbulence** model. Use the abbreviations that are given in the Models Overview above.

	EB	LPS	LPS 2L	QPS
A1 The coefficient A_1 in Eqn. (1952).	✔			
Alpha Minimum The minimum value that the transported variable α is permitted to have. An appropriate value is a small number that is greater than the floating point minimum of the computer.	✔			
Buoyancy Production of Dissipation Determines how the coefficient $C_{\varepsilon 3}$ in Eqn. (1928) is calculated. • **None**: Neglects the term $C_{\varepsilon 3}\mathrm{tr}(\mathbf{G})$. • **Boundary Layer Orientation**: Computes $C_{\varepsilon 3}$ according toEqn. (1801). • **Thermal Stratification**: Computes $C_{\varepsilon 3}$ according to Eqn. (1802). • **Constant Coefficient**: Computes $C_{\varepsilon 3}$ as a constant coefficient. This option requires the specification of $C_{\varepsilon 3}$ in the corresponding child node **C3e**.	✔	✔	✔	✔
C1 The coefficient C_1 in Eqn. (1944) (EB) and Eqn. (1931) (LPS, LPS 2L).	✔	✔	✔	
C1e The coefficient $C_{\varepsilon 1}$ in Eqn. (1928).	✔			
C1s The coefficient C_{1_s} in Eqn. (1944).	✔			
C1w The coefficient C_{1w}in Eqn. (1934).		✔	✔	
C2 The coefficient C_2 in Eqn. (1932).		✔	✔	

	EB	LPS	LPS 2L	QPS
C2e The coefficient $C_{\varepsilon 2}$ in Eqn. (1928).	✔	✔	✔	✔
C2w The coefficient C_{2w} in Eqn. (1936).		✔	✔	
C3 The coefficient C_3 in Eqn. (1933).	✔			
C3s The coefficient C_{3_s} in Eqn. (1944).	✔			
C4 The coefficient C_4 in Eqn. (1944).	✔			
C5 The coefficient C_5 in Eqn. (1944).	✔			
Ceta The coefficient C_η in Eqn. (1949).	✔			
Cl The coefficient C_l in Eqn. (1949) (EB) and Eqn. (1935) (LPS, LPS 2L).	✔	✔	✔	
Cmu The coefficient $C_\mu$in Eqn. (1922).	✔	✔	✔	✔
Convection Controls the convection scheme. • **1st-order**: Selects the first-order upwind convection scheme. • **2nd-order**: Selects the second-order upwind convection scheme.	✔	✔	✔	✔
Cr1 The coefficient C_{r1} in Eqn. (1941).				✔

	EB	LPS	LPS 2L	QPS
Cr2 The coefficient C_{r2} in Eqn. (1941).				✔
Cr3 The coefficient C_{r3} in Eqn. (1941).				✔
Cr3star The coefficient C_{r3}^{*} in Eqn. (1941).				✔
Cr4 The coefficient C_{r4} in Eqn. (1941).				✔
Cs The coefficient C_s in Eqn. (1924).	✔	✔	✔	✔
Cs1 The coefficient C_{s1} in Eqn. (1941).				✔
Cs2 The coefficient C_{s2} in Eqn. (1941).				✔
Ct The coefficient C_t in Eqn. (1951).	✔			
FwMax The coefficient $f_w^{\max}$ in Eqn. (1935).		✔	✔	
Sarkar The coefficient C_M in Eqn. (1929).	✔	✔	✔	✔
Sigma_e The coefficient σ_ε in Eqn. (1776) in K-Epsilon model formulation.	✔	✔	✔	✔
Sigma_k The coefficient σ_k in Eqn. (1921).	✔	✔	✔	✔

	EB	LPS	LPS 2L	QPS
Secondary Gradients Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces. • **On**: Include both secondary gradients. • **Off**: Exclude both secondary gradients. • **Interior Only**: Include the interior secondary gradients only. • **Boundaries Only**: Include the boundary secondary gradients only.	✔	✔	✔	✔
Tdr Minimum The minimum value that the transported variable ε is permitted to have. An appropriate value is a small number that is greater than the floating point minimum of the computer.	✔	✔	✔	✔
Tke Minimum The minimum value that the transported variable k is permitted to have. An appropriate value is a small number that is greater than the floating point minimum of the computer.	✔	✔	✔	✔
Two-Layer Delta ReY The value of $\Delta\ \mathrm{Re}_y$ used in Eqn. (1805) in two-layer models formulation.			✔	
*Two-Layer ReY** The value of $\mathrm{Re}_y{}^*$ used in Eqn. (1804) in two-layer models formulation.			✔	
Two-Layer Type Controls the type of two-layer formulation • **Shear Driven (Wolfstein)**: Selects the two-layer formulation of Wolfstein [328]. Use this selection for flows where buoyancy forces do not dominate. See Eqn. (1807) and Eqn. (1808). • **Buoyancy Driven (Xu)**: Selects the two-layer formulation of Xu [329]. Use this selection for flows where buoyancy forces dominate. See Eqn. (1811) and Eqn. (1812). • **Shear Driven (Norris-Reynolds)** : Selects the two-layer formulation of Norris and Reynolds [323]. Use this selection for flows where buoyancy forces do not dominate. See Eqn. (1809) and Eqn. (1810).			✔	
Use Boussinesq Approximation When **On**, instead of using the divergence of the computed Reynolds stress tensor, uses the Boussinesq approximation, Eqn. (1759).	✔	✔	✔	✔

	EB	LPS	LPS 2L	QPS
Use Daly-Harlow When **On**, instead of using the isotropic formulation, as in Eqn. (1921), uses the Generalized Gradient Diffusion model, Eqn. (1924).	✔	✔	✔	✔
Wall Reflection Term When **On**, includes the wall-reflection terms $\underline{\Phi}_{1w}$ and $\underline{\Phi}_{2w}$ (see Eqn. (1930))		✔	✔	

Reynolds Stress Initial Conditions Reference

This section details the conditions and values that are required for initialization of a Reynolds Stress turbulence model.

Turbulence Specification

Controls how you define the turbulence profile for initialization.

The Reynolds Stress model requires the Reynolds stress tensor $\mathbf{R}$ and the turbulent dissipation rate ε. The corresponding values are derived from other turbulence quantities.

Method	Corresponding Value Nodes
K + Epsilon Calculation of $\mathbf{R}$ from the turbulent kinetic energy k using Eqn. (1969).	**Turbulent Kinetic Energy** Scalar profile value to specify k. **Turbulent Dissipation Rate** Scalar profile value to specify ε directly.
Intensity + Length Scale Calculation of $\mathbf{R}$ and ε from a specified turbulence intensity I, length scale L, and velocity scale v using Eqn. (1965) and Eqn. (1967). Do not use this method if the velocity field is initialized to zero.	**Turbulence Intensity** Scalar profile value to specify I. **Turbulent Length Scale** Scalar profile value to specify l. **Turbulent Velocity Scale** Scalar profile value to specify v. For an initial value, use a representative velocity or a representative velocity scale. For example, for a pipe flow use the inlet velocity.
Intensity + Viscosity Ratio Calculation of $\mathbf{R}$ and ε from a specified turbulence intensity I, the ratio of turbulent to laminar viscosity μ_t/μ, and the turbulent velocity scale v using Eqn. (1966) and Eqn. (1968).	**Turbulence Intensity** As for **Intensity + Length Scale**.

Do not use this method if the velocity field is initialized to zero.

Turbulent Viscosity Ratio

Scalar profile value to specify the ratio of turbulent to laminar viscosity μ_t/μ.

Turbulent Velocity Scale

Scalar profile value to specify v.

For an initial value, use a representative velocity or a representative velocity scale. For example, for a pipe flow use the inlet velocity.

Reynolds Stress Boundaries Reference

This section details the conditions that are required for each boundary type that is supported by the Reynolds Stress turbulence model. In addition, information is provided about the values for each boundary type.

Boundary conditions and values are elected by the boundary type, and/or by methods selected in corresponding nodes in the **Conditions** node.

Note: Boundary types that do not require setting any conditions or values are not listed.

Flow Boundaries

The following boundary conditions and values are equal to all flow boundaries of type:

- Free Stream
- Mass Flow Inlet
- Pressure Outlet
- Stagnation Inlet
- Velocity Inlet

Turbulence Specification

Controls how you define the turbulence profile at flow boundaries.

The Reynolds Stress model requires the Reynolds stress tensor $\mathbf{R}$ and the turbulent dissipation rate ε. You can enter the corresponding values directly or have them derived from other turbulence quantities.

Method	Corresponding Value Nodes
K + Epsilon Calculation of $\mathbf{R}$ from the turbulent kinetic energy k using Eqn. (1969).	**Turbulent Kinetic Energy** Scalar profile value to specify k. **Turbulent Dissipation Rate** Scalar profile value to specify ε directly.
Intensity + Length Scale	**Turbulence Intensity** Scalar profile value to specify I.

Calculation of $\mathbf{R}$ and ε from a specified turbulence intensity I and length scale Lusing Eqn. (1965) and Eqn. (1967). Do not use this method if the velocity field is initialized to zero.	**Turbulent Length Scale** Scalar profile value to specify l.
Intensity + Viscosity Ratio Calculation of $\mathbf{R}$ and ε from a specified turbulence intensity I and the ratio of turbulent to laminar viscosity μ_t/μ using Eqn. (1966) and Eqn. (1968). Do not use this method if the velocity field is initialized to zero.	**Turbulence Intensity** As for **Intensity + Length Scale**. **Turbulent Viscosity Ratio** Scalar profile value to specify the ratio of turbulent to laminar viscosity μ_t/μ.
RS + Epsilon	**Turbulent Dissipation Rate** Scalar profile value to specify ε directly. **Reynolds Stresses** Scalar profile values to specify the components of the Reynolds Stress tensor directly: • **UU Reynolds Stress** • **UV Reynolds Stress** • **UW Reynolds Stress** • **VV Reynolds Stress** • **VW Reynolds Stress** • **WW Reynolds Stress**

Reynolds Stress Regions Reference

This section details the conditions that are required for each region type that is supported by the Reynolds Stress turbulence model. In addition, information is provided about the values for each region type.

Region conditions and values are elected by the region type, and/or by methods selected in corresponding nodes in the **Conditions** node.

Note: Region types that do not require setting any conditions or values are not listed.

Fluid Region

Turbulence Source Option

Controls whether you want to use a turbulence source term.

Turbulence Source Option	Corresponding Value Nodes
Deactivated	None
Activated	**Reynolds Stress User Sources** Scalar profile values to specify turbulence sources for the components of the Reynolds Stress tensor directly: • **UU Stress User Source** • **UV Stress User Source** • **UW Stress User Source** • **VV Stress User Source** • **VW Stress User Source** • **WW Stress User Source** **Reynolds Stress User Source Derivatives** Scalar profile values to specify the derivatives of the turbulence sources for the components of the Reynolds Stress tensor directly: • **UU Stress User Source Derivative** • **UV Stress User Source Derivative** • **UW Stress User Source Derivative** • **VV Stress User Source Derivative** • **VW Stress User Source Derivative** • **WW Stress User Source Derivative** **Turbulent Dissipation Rate Source** Scalar profile value to specify a turbulence source for ε directly. **Turbulent Dissipation Rate Source Derivative** Scalar profile value to specify the derivative of a turbulent dissipation rate source against dissipation rate directly.

Porous Region

Turbulence Specification

Controls how you define the turbulence profile.

The Reynolds Stress model requires the Reynolds stress tensor $\mathbf{R}$ and the turbulent dissipation rate ε. The corresponding values are derived from other turbulence quantities.

Method	Corresponding Value Nodes
K + Epsilon Calculation of $\mathbf{R}$ from the turbulent kinetic energy k using Eqn. (1969).	**Turbulent Kinetic Energy** Scalar profile value to specify k.

	Turbulent Dissipation Rate Scalar profile value to specify ε directly.
Intensity + Length Scale Calculation of $\mathbf{R}$ and ε from a specified turbulence intensity I and length scale L using Eqn. (1965) and Eqn. (1967). Do not use this method if the velocity field is initialized to zero.	**Turbulence Intensity** Scalar profile value to specify I. **Turbulent Length Scale** Scalar profile value to specify l.
Intensity + Viscosity Ratio Calculation of $\mathbf{R}$ and ε from a specified turbulence intensity I and the ratio of turbulent to laminar viscosity μ_t/μ using Eqn. (1966) and Eqn. (1968). Do not use this method if the velocity field is initialized to zero.	**Turbulence Intensity** As for **Intensity + Length Scale**. **Turbulent Viscosity Ratio** Scalar profile value to specify the ratio of turbulent to laminar viscosity μ_t/μ.

Reynolds Stress Solvers Reference

This section details the solvers that are associated with a Reynolds Stress turbulence model: turbulence and turbulent viscosity.

Reynolds Stress Turbulence

This solver controls the solution of the turbulence transport equations in all the continua for which a Reynolds Stress model is activated.

For each transported variable (components of $\mathbf{R}$ and the dissipation rate ε) the basic steps that are involved in the solution update are as follows:

1. Update boundary conditions.
2. Compute the reconstruction gradients and cell gradients.
3. Create the linear system of equations (see Eqn. (1615)) using the discretization methods that are described in the K-Omega model formulation.
4. Compute the residual sum $R = \sum_{\text{cells}} |r|$ for monitoring convergence.
5. Solve the linear system.
6. Update the transported variable field.

Properties

Under-Relaxation Factor

At each iteration, governs the extent to which the newly computed solution supplants the old solution. The default value is 0.6. For the theoretical background, see Eqn. (1625).

Boundary Layer Initialization

When **On**, this computes initial values for turbulent kinetic energy and dissipation rate automatically, taking account of the proximity of walls as well as of free-stream values. This speeds convergence in most cases. (See Eqn. (1970) to Eqn. (1973).) The property is **Off** by default.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Under-Relaxation Factor Ramp

The **Reynolds Stress Turbulence** > **Under-Relaxation Factor Ramp** node allows you to set and control a linear ramp for under-relaxation. For details, see Under-Relaxation Factor Ramp Reference.

AMG Linear Solver

The **Reynolds Stress Turbulence** > **AMG Linear Solver** node allows you to set the algebraic multigrid parameters for the turbulence solver. For details, see Algebraic Multigrid and AMG Linear Solver Properties.

Reynolds Stress Turbulent Viscosity

This solver controls the update of the turbulent viscosity.

Let μ_t^n be the value of the previous iteration, and μ_t^{new} be the value that is computed during the current iteration. The update is controlled as follows:

$$\mu_t^{n+1} = \omega_\mu \mu_t^{\text{new}} + \left(1-\omega_\mu\right)\mu_t^n$$

Properties

Maximum Ratio	The maximum ratio of turbulent to laminar viscosity (μ_t/μ) permitted during the update process. This ratio prevents unphysically high values of turbulent viscosity from occurring on the path to convergence. See Troubleshooting RANS Solvers.

Under-Relaxation Factor	An under-relaxation factor ω_μ for the update of turbulent viscosity. The default value is 1.0.
Solver Frozen	When **On**, this prevents the solver from updating the solution while iterating. The property is **Off** by default.

Reynolds Stress Field Functions Reference

This section details the field functions that are made available to the simulation when a Reynolds Stress turbulence model is used.

Kolmogorov Length Scale

Scalar field that represents the turbulent length scale η, as defined in Eqn. (2069).

This field is only available when the *Temporary Storage Retained* option is ticked for the relevant turbulence solver.

Kolmogorov Time Scale

Scalar field that represents the turbulent time scale τ_η, as defined in Eqn. (2070).

This field is only available when the *Temporary Storage Retained* option is ticked for the relevant turbulence solver.

Reynolds Stress uu

The specific normal stress $\overline{u'u'}$.

Reynolds Stress uv

The specific shear stress $\overline{u'v'}$.

Reynolds Stress uw

The specific shear stress $\overline{u'w'}$.

Reynolds Stress vv

The specific normal stress $\overline{v'v'}$.

Reynolds Stress vw

The specific shear stress $\overline{v'w'}$.

Reynolds Stress ww

The specific normal stress $\overline{w'w'}$.

Strain Rate Tensor Modulus

Scalar field that represents the modulus of the mean strain rate tensor S, as defined in Eqn. (1744).

This field is only available when the **Temporary Storage Retained** option is ticked for the relevant turbulence solver.

Taylor Micro Scale

Scalar field that represents the turbulent length scale λ, as defined in Eqn. (2071).

This field is only available when the *Temporary Storage Retained* option is ticked for the relevant turbulence solver.

Turbulent Dissipation Rate

The transported variable ε.

Turbulent Kinetic Energy

The variable k , computed from the trace of the specific Reynolds stress tensor.

Turbulent Viscosity

Scalar field that represents the turbulent viscosity μ_t.

Turbulent Viscosity Ratio

Scalar field that represents the ratio of turbulent to laminar viscosity μ_t/μ.

Initialization and Inflow for RANS Turbulence Models

The specification of realistic initial and inflow boundary values is important for good convergence and accurate simulation results.

For a one-equation turbulence model, only a single value is required, for two-equation turbulence models, two values are required, and so forth.

For many engineering applications, two-equation models are used. The k-epsilon models for example require two boundary conditions: one for k and one for ε. These values cannot be measured directly, but there are equations that calculate k and ε from values that you can measure or estimate such as turbulence intensity and length scale. In STAR-CCM+, three turbulence specification methods are typically available (depending on the turbulence model):

- Direct specification of the turbulence variables (such as turbulent kinetic energy and turbulent dissipation rate)
- Specification of turbulence intensity and turbulent length scale
- Specification of turbulence intensity and turbulent viscosity ratio

It does not matter which method you use to specify the boundary conditions, STAR-CCM+ converts these values to k and ε values automatically.

Turbulence intensity is a measure of the root mean square of the local velocity fluctuations relative to the mean velocity and is typically estimated as follows:

- High-turbulence case: High-speed flow inside complex geometries like heat-exchangers and flow inside rotating machinery (turbines and compressors). Typically the turbulence intensity is between 5% and 20%
- Medium-turbulence case: Flow in not-so-complex devices like large pipes or ventilation flows, or low speed flows (low Reynolds number). Typically the turbulence intensity is between 1% and 5%

- Low-turbulence case: Flow originating from a fluid that is stagnant, like external flow across cars, submarines and aircrafts. High-quality wind-tunnels can also reach very low turbulence levels. Typically the turbulence intensity is low, below 1%.

Turbulence length scale is a physical quantity describing the size of the large energy-containing eddies in a turbulent flow. It is typically set to be about 5% of the characteristic dimension of the inflow, so for a 30 mm diameter pipe it would be 1.5 mm.

To avoid divergence during the initial iterations, it also helps to have reasonable initial values for the turbulence quantities. Admittedly, it might be hard to find a single number that describes the turbulence adequately throughout the domain. However, the turbulence intensity is typically set to a value in the range $0.01 < I < 0.1$ and the turbulent viscosity ratio in the range $10 < \mu_t/\mu < 100$. These values mean that there is some additional viscosity over and above the laminar viscosity, and the dissipation scales are not unreasonable.

Troubleshooting RANS Turbulence Solvers

This section provides guidance for fixing problems with Reynolds-Averaged Navier-Stokes (RANS) turbulence solvers.

Solution of the transported turbulence variables diverges

The residual for any of the transported turbulence variables increases significantly after the startup transient, indicating that the solution of the turbulence field is diverging.

- Make sure the velocity field is not diverging and hence causing the turbulence model to diverge.
- Slightly reducing the turbulence solver under-relaxation factor (say to between 0.3 and 0.5) might stabilize the solution.
- In situations where mesh quality is impeding convergence, using the first-order convection scheme might help.

Solution of the transported turbulence variables converges slowly

The flow residuals are converging well, but the residuals for the transported turbulence variables are not decreasing as fast.

- You can increase the turbulence solver under-relaxation factor if you are confident that the solution is otherwise progressing well, but a value of more than 0.9 is not recommended, since it might result in solution divergence.

Unrealistically high turbulent viscosity impedes flow solution

During the first few iterations, unrealistically high turbulent viscosity values are impeding the flow solution.

- Reducing the turbulence solver under-relaxation factor will retard the update of turbulent viscosity.
- It is possible to make this factor quite small (say 0.01) to cause only a minimal update, but this would typically be done for a few iterations only.

Warnings of limited turbulent viscosity occur

Repeated warnings occur that the turbulent viscosity is being limited in some cells.

- This is both common and acceptable during the first few iterations due to startup transients.
- If the warnings continue to occur as the flow solution nears convergence, it is possible to increase the Maximum Ratio to prevent the warnings.
- Generally one would only increase the *Maximum Ratio* if the simulation has a very high Reynolds number and no errors seem to be present in the simulation setup. The turbulent viscosity ratio is proportional to a turbulent Reynolds number, where the velocity scale is typically given as $k^{1/2}$ and the length scale as $\left(k^{3/2}\right)/\varepsilon$ (for a model based on the eddy viscosity concept). If you expect very high turbulent Reynolds numbers (which occur when the mean flow Reynolds number is high, such as in a planetary boundary layer), you could try increasing the Maximum Ratio limit by one or two orders of magnitude. If you do not expect high turbulent Reynolds numbers, it is likely that another problem is present in your simulation, such as an incorrect boundary condition, or a mesh that hasn't been scaled correctly.

Modeling Turbulent Heat Flux

The **Temperature Flux Model** replaces the Boussinesq approximation in the energy equation by an algebraic formulation for the turbulent heat flux. In STAR-CCM+, this approach is available for the Standard K-Epsilon Low-Reynolds Number model.

The classical definition of heat flux in the energy equation is based on the Boussinesq approximation, where the heat flux is assumed to be proportional to the turbulent eddy-viscosity. This assumption fails either when buoyancy forces are dominant, or at locations very near the wall. A remedy proposed by Kenjeres and others [378] is to replace the Boussinesq approximation by an algebraic formulation for the turbulent heat flux itself. This formulation is a function of the Reynolds-stress anisotropy and the temperature variance, for which an additional transport equation is solved. The performance of the algebraic heat flux (or Temperature Flux) model is strongly linked to the correct approximation of the near-wall turbulent behavior, thus requiring a low-Reynolds number turbulence model.

Temperature Flux Model Reference

Theory	See Temperature Flux Formulation.
Provided By	**[physics continuum]** > **Models** > *Optional Models*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Temperature Flux Model**
Requires	*Space*: **Axisymmetric**, **Two Dimensional**, or **Three Dimensional** *Time*: **Implicit Unsteady**, **PISO Unsteady**, or **Steady** *Material*: **Gas** or **Liquid** *Flow*: **Segregated Flow** or **Coupled Flow** *Optional Models*: **Segregated Fluid Temperature**, **Segregated Fluid Enthalpy**, or **Coupled Energy** *Equation of State*: **Constant Density** *Viscous Regime*: **Turbulent** *Turbulence*: **Reynolds-Averaged Navier-Stokes** *Reynolds-Averaged Turbulence*: **K-Epsilon Turbulence** *K-Epsilon Turbulence Models*: **Standard K-Epsilon Low-Re** *Wall Treatment*: **All y+ Wall Treatment** or **Low y+ Wall Treatment**

	Optional Models: **Gravity** *Optional Models*: **Boussinesq Model**	
Properties	See Turbulent Heat Flux Properties.	
Activates	Initial Conditions	**Temperature Variance** See Initial Conditions.
	Boundary Inputs	**Temperature Variance** See Boundary Settings.
	Region Inputs	**Temperature Variance Source Option** See Region Settings.
	Solvers	**Temperature Flux Solver** See Temperature Flux Solver Properties.
	Field Functions	**Temperature Variance** See Field Functions.

Turbulent Heat Flux Properties

Convection

Controls the convection scheme.

- **1st-order**: Selects the first-order upwind convection scheme.
- **2nd-order**: Selects the second-order upwind convection scheme.

Ctu0

The coefficient C_{tu_0} , see Eqn. (382) and Eqn. (388).

Ctu1

The coefficient C_{tu_1} , see Eqn. (382) and Eqn. (388).

Ctu2

The coefficient C_{tu_2} , see Eqn. (382) and Eqn. (388).

Ctu3

The coefficient C_{tu_3} , see Eqn. (382) and Eqn. (388).

Ctu4

The coefficient C_{tu_4} , see Eqn. (382) and Eqn. (388).

R

The turbulent-to-thermal time-scale ratio R , see Eqn. (387) and Eqn. (388).

Secondary Gradients

Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces.

- **On**: Include both secondary gradients.
- **Off**: Exclude both secondary gradients.
- **Interior Only**: Include the interior secondary gradients only.
- **Boundaries Only**: Include the boundary secondary gradients only.

Sigma_Theta2

The coefficient σ_{θ^2} , see Eqn. (385) and Eqn. (388).

Theta2 Minimum

The minimum value that the transported variable $\overline{\theta^2}$ is permitted to have. An appropriate value is a small number that is greater than the floating point minimum of the computer.

Initial Conditions

The Temperature Flux model requires the temperature variance $\overline{\theta^2}$ as initial condition. You can enter the corresponding value directly.

Temperature Variance

Scalar profile value to specify $\overline{\theta^2}$ directly.

Boundary Settings

Inflow and Outflow Boundaries

Temperature Variance

Scalar profile value to specify $\overline{\theta^2}$ directly.

Region Settings

Applies to fluid regions.

Temperature Variance Source Option

Controls whether you want to apply a source for the temperature variance.

Method	Corresponding Physics Value Nodes
Deactivated	None.
Activated	**Temperature Variance Source** Scalar profile value to specify a source for $\overline{\theta^2}$ directly.

Temperature Flux Solver Properties

Under-Relaxation Factor

At each iteration, governs the extent to which the newly computed solution supplants the old solution. The default value is 0.8. For the theoretical background, see Eqn. (1625).

Under-Relaxation Factor Ramp

The **Temperature Flux Solver** > **Under-Relaxation Factor Ramp** node allows you to set and control a linear ramp for under-relaxation. For details, see Under-Relaxation Factor Ramp Reference.

AMG Linear Solver

The **Temperature Flux Solver** > **AMG Linear Solver** node allows you to set the algebraic multigrid parameters for the turbulence solver. For details, see Algebraic Multigrid and AMG Linear Solver Properties.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Field Functions

Temperature Variance

Scalar field that represents $\overline{\theta^2}$ in Eqn. (382), Eqn. (385), and Eqn. (387).

Contents:

Temperature Flux Formulation
Temperature Flux Nomenclature

Temperature Flux Formulation

The following is a mathematical overview of the Temperature Flux model.

The classical definition of the heat flux in the energy equation is based on a Boussinesq approximation:

$$\acute{\mathbf{q}}'' = -\left(k + \frac{\mu_t C_p}{\sigma_t}\right)\nabla T \tag{381}$$

where σ_t is a turbulent Prandtl Number, usually taken as a constant equal to 0.9. The Temperature Flux model is based on an algebraic formulation for the turbulent heat flux [378]:

$$\overline{\mathbf{v}\theta} = -C_{tu_0}\frac{k}{\varepsilon}\left(C_{tu_1}\mathbf{R}\cdot\nabla T + C_{tu_2}\nabla\mathbf{v}\cdot\overline{\mathbf{v}\theta} + C_{tu_3}\beta\overline{\theta^2}\mathbf{g}\right) + C_{tu_4}\mathbf{A}\cdot\overline{\mathbf{v}\theta} \tag{382}$$

where $\overline{\mathbf{v}\theta}$ replaces the Boussinesq approximation in the definition of the total heat flux:

$$\acute{\mathbf{q}}'' = -\kappa\nabla T - \rho C_p\overline{\mathbf{v}\theta} \tag{383}$$

The turbulent heat flux $\overline{\mathbf{v}\theta}$ is also used for the buoyancy-related source terms G_b in the transport equation for k and ε (Eqn. (1791)):

$$G_b = -\rho\beta\,\mathbf{g}\cdot\overline{\mathbf{v}\theta} \tag{384}$$

In Eqn. (382), the remaining unknown, the temperature variance $\overline{\theta^2}$, is computed by solving an additional transport equation:

$$\frac{d}{dt}\int_V \rho\overline{\theta^2}\,dV + \int_A \rho\overline{\theta^2}\left(\mathbf{v}-\mathbf{v}_g\right)\cdot d\mathbf{a} = \int_A\left(\mu + \frac{\mu_t}{\sigma_{\theta^2}}\right)\nabla\overline{\theta^2}\cdot d\mathbf{a} + \int_V\left(G_\theta - \rho\epsilon_\theta\right)dV \tag{385}$$

where the production of the temperature variance G_θ is:

$$G_\theta = -\rho\overline{\mathbf{v}\theta}\cdot\nabla T \tag{386}$$

The temperature variance dissipation rate ε_θ is obtained from the definition of the thermal time scale (equivalent for the temperature variance to the turbulence time scale for the K-Epsilon model):

$$\tau_\theta = \frac{\overline{\theta^2}}{2\varepsilon_\theta} \tag{387}$$

and assuming a constant turbulent-to-thermal time-scale ratio $R = \tau_\theta/\tau$:

$$\begin{gathered} C_{tu_0} = 0.12,\quad C_{tu_1} = 1,\quad C_{tu_2} = 0.6,\quad C_{tu_3} = 0.6 \\ C_{tu_4} = 1.5,\quad \sigma_{\theta^2} = 1,\quad R = 0.5 \end{gathered} \tag{388}$$

Temperature Flux Nomenclature

The following is a guide to the terms used in this section.

Scalar Quantities

Variable	Definition
C_p	Specific heat $[J/kg \cdot K]$
G_θ	Production of the temperature variance $\left[K^2/s\right]$
R	Time-scale ratio
T	Temperature $[K]$
$\overline{\theta^2}$	Temperature variance $\left[K^2\right]$
β	Thermal expansion coefficient $[1/K]$
ε_θ	Dissipation of the temperature variance $\left[K^2/s\right]$
κ	Thermal conductivity $[W/m \cdot K]$
μ	Dynamic viscosity $[Pa \cdot s]$
μ_t	Turbulent viscosity $[Pa \cdot s]$
ρ	Density $\left[kg/m^3\right]$
σ_t	Turbulent Prandtl number
τ	Turbulent time scale $[s]$
τ_θ	Thermal time scale $[s]$

Vectors

Variable	Definition
$\mathbf{g}$	Gravity $\left[m/s^2\right]$
$\mathbf{v}$	Velocity $[m/s]$
$\mathbf{v}_g$	Grid velocity $[m/s]$
$\overline{\mathrm{v}\theta}$	Turbulent heat flux $[mK/s]$
$\dot{\mathbf{q}}''$	Heat flux $\left[W/m^2\right]$

Tensors

Variable	Definition
$\mathbf{A}$	Reynolds-stress anisotropy
$\mathbf{R}$	Reynolds-stress $\left[m^2/s^2\right]$

Scaling Turbulent Viscosity

The **Turbulent Viscosity User Scaling** model allows you to apply a custom scaling factor to the turbulent viscosity of K-Epsilon and K-Omega turbulence models.

Use this model when the turbulent flow includes a range or mixture of densities.

Turbulent Viscosity User Scaling Model Reference

Theory	See Theory Guide—Turbulent Viscosity User Scaling.	
Provided by	**Continua** > **[physics continuum]** > **Models** > **Optional Models**	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Turbulent Viscosity User Scaling**	
Requires	*Space*: **Axisymmetric**, **Two Dimensional**, or **Three Dimensional** *Material* **Gas**, **Liquid**, **Eulerian Multiphase**, **Multi-Component Gas**, or **Multi-Component Liquid** *Viscous Regime*: **Turbulent** *Turbulence*: **Reynolds-Averaged Navier-Stokes** *Reynolds-Averaged Turbulence*: **K-Epsilon Turbulence** or **K-Omega Turbulence**	
Properties	None.	
Activates	Region Inputs	**User Turbulent Viscosity Scaling** See Region Settings.

Region Settings

Applies to fluid and porous regions.

User Turbulent Viscosity Scaling

Scalar profile value to specify the scaling factor α in Eqn. (1918).

Scale-Resolving Simulations

In contrast to RANS models, scale-resolving simulations resolve the large scales of turbulence and model small-scale motions.

Two approaches are available in STAR-CCM+:

- Large Eddy Simulation (LES)
- Detached Eddy Simulation (DES)

A Large Eddy Simulation resolves turbulent structures in space everywhere in the flow domain down to the grid limit, where subgrid models approximate the impact of the subgrid structures on the flow field. In order to resolve the crucial turbulent structures near the wall, this approach requires an excessively high mesh resolution in the wall boundary layer—not only in the direction normal to the wall but also in the flow direction. As a result of the high computational costs that go along with the high cell count, LES is used mainly for academic applications or for flows with low Reynolds numbers.

A Detached Eddy Simulation (DES) is a Hybrid LES-RANS approach that resolves turbulent structures in the core flow and covers the wall boundary layers by a RANS model. Thereby, DES avoids the expensive mesh requirements of LES.

Contents:

Large Eddy Simulation (LES)
Detached Eddy Simulation (DES)
Initialization and Inflow for LES and DES

Large Eddy Simulation (LES)

Large Eddy Simulation (LES) is an inherently transient technique in which the large scales of the turbulence are directly resolved everywhere in the flow domain, and the small-scale motions are modeled.

One justification for the LES technique is that by modeling “less” of the turbulence, and explicitly solving for more of it, the error in the turbulence modeling assumptions is not as consequential. Furthermore, it is hypothesized that the smaller eddies are self-similar and thus lend themselves to simpler and more universal models. The downside of the approach is the computational expense, which, although less than direct numerical simulation, is still nonetheless excessive.

In contrast to the RANS equations, the equations that are solved for LES are obtained by a spatial filtering rather than an averaging process. Each solution variable ϕ is decomposed into a filtered value $\widetilde{\phi}$ and a sub-filtered, or sub-grid, value ϕ':

$$\phi = \widetilde{\phi} + \phi' \tag{389}$$

where ϕ represents velocity components, pressure, energy, or species concentration.

Inserting the decomposed solution variables into the Navier-Stokes equations results in equations for the filtered quantities. The filtered equations are rearranged into a form that looks identical to the unsteady RANS equations. However, the turbulent stress tensor $\mathbf{T}_t$ now represents the subgrid scale stresses. These stresses

result from the interaction between the larger, resolved eddies and the smaller, unresolved eddies and are modeled using the Boussinesq approximation as follows:

$$\mathbf{T}_t = 2\mu_t \mathbf{S} - \frac{2}{3}\left(\mu_t \nabla \cdot \tilde{\mathbf{v}} + \rho k\right)\mathbf{I} \tag{390}$$

where:

- $\mathbf{S}$ is the mean strain rate tensor given by Eqn. (1745) and computed from the resolved velocity field.
- $\tilde{\mathbf{v}}$ is the filtered velocity.
- ρ is the density.
- k is the subgrid scale turbulent kinetic energy.

The subgrid scale turbulent viscosity μ_t must be described by a Subgrid Scale Model that accounts for the effects of small eddies on the resolved flow.

Mean-flow quantities must be obtained by gathering statistics either over a long physical time, and/or from a homogeneous spatial coordinate. Field function monitors are provided in STAR-CCM+ to monitor the field statistics.

Inflow Boundaries and Initialization

One of the difficulties in large eddy simulations is representing the flow upstream of the flow domain, as an inadequate amount of information introduces sources of error. In STAR-CCM+ the Synthetic Eddy Method is implemented to automatically provide turbulent eddies across inflow boundaries and provide an initial perturbated flow field.

Contents:

Subgrid Scale Turbulence Models
Selecting LES and a Subgrid Scale Turbulence Model
LES and Subgrid Scale Turbulence Models Reference
LES Guidelines
Troubleshooting LES

Subgrid Scale Turbulence Models

Large Eddy Simulation requires closure of the filtered Navier-Stokes equations by modeling the subgrid scale stress tensor.

A subgrid scale model provides a formula for the subgrid scale viscosity, μ_t, in the Boussinesq approximation of the subgrid scale stress tensor (see Eqn. (380)). Currently, three subgrid scale models are available in STAR-CCM+:

Smagorinsky Subgrid Scale

The Smagorinsky Subgrid Scale model ([364]) is the original and most basic subgrid scale model, upon which many more advanced models are based. This model uses a mixing length hypothesis to model the subgrid scale stresses.

One shortcoming of the Smagorinsky Subgrid Scale model is that it contains a model coefficient C_s that is not universal and depends on the local flow conditions. Additionally, a damping function is needed for proper results in wall-bounded flows. The Van Driest damping function is a non-local operation, requiring the use of a KD-tree data structure to communicate wall information to the interior. In simulations with very complex geometries, this KD-tree must be broadcast to each parallel node, resulting in a memory and communication overhead. An update policy can be used to improve the communication overhead, but not the memory overhead. Therefore, this model should be avoided for large-scale computations on complex geometries. It is, however, useful for reference simulations, since it is the original algebraic subgrid scale model formulation.

Dynamic Smagorinsky Subgrid Scale

The Dynamic Smagorinsky Subgrid Scale model ([361], [362]) has the same basic form as the Smagorinsky model, however instead of using a single user-defined C_s coefficient, the model computes a local time-varying coefficient by test-filtering the flow field at a length scale greater than the grid length scale. This dynamic variation of the constant gives the model its name, and allows it to compute the correct result for wall-bounded flows without the use of damping functions.

WALE Subgrid Scale

The WALE (Wall-Adapting Local-Eddy Viscosity) Subgrid Scale model ([363]) is a more modern subgrid scale model that uses a novel form of the velocity gradient tensor in its formulation.

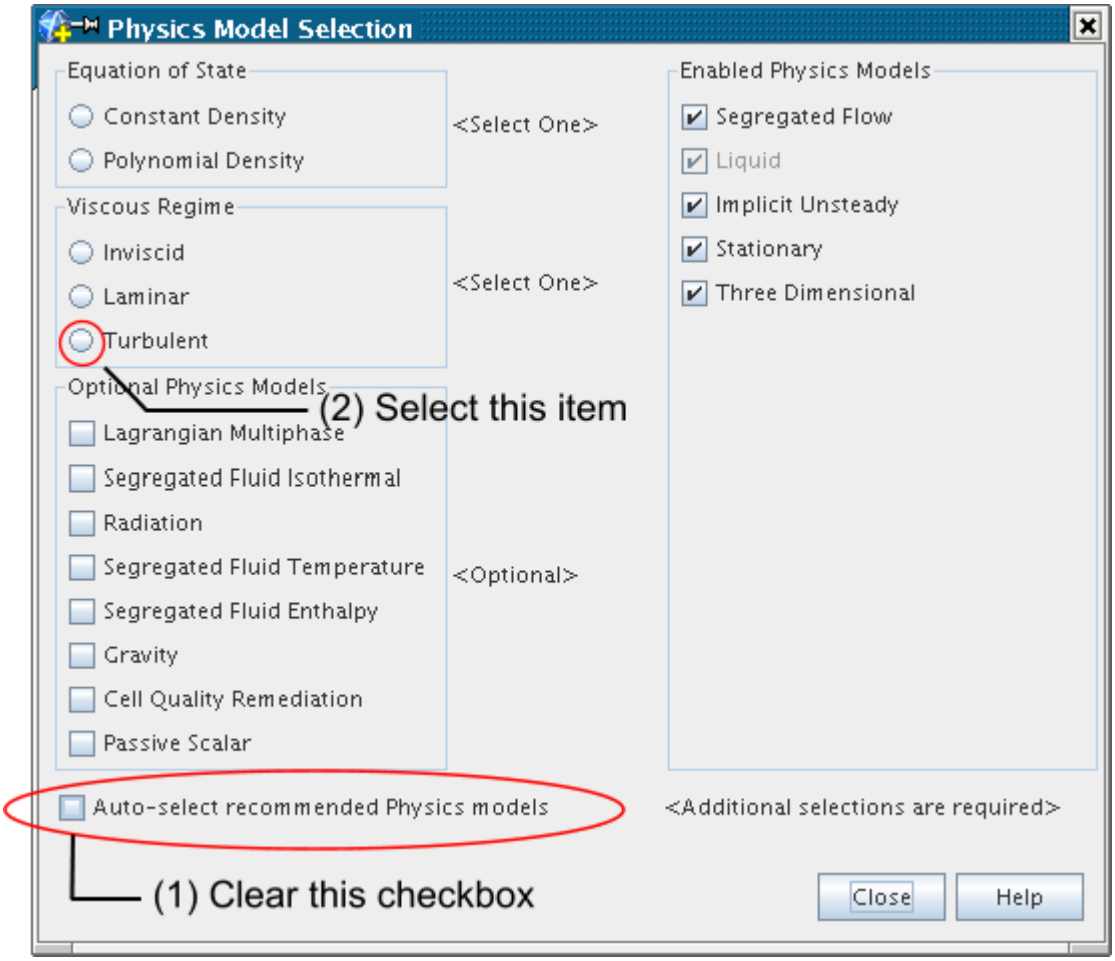
Similar to the Smagorinsky Subgrid Scale model, it suffers from the limitation that the model coefficient is not universal. But validations using STAR-CCM+ have shown that the WALE model is seemingly less sensitive to the value of this coefficient than the Smagorinsky model. Another advantage of the WALE model is that it does not require any form of near-wall damping—it automatically gives accurate scaling at walls.

Selecting LES and a Subgrid Scale Turbulence Model

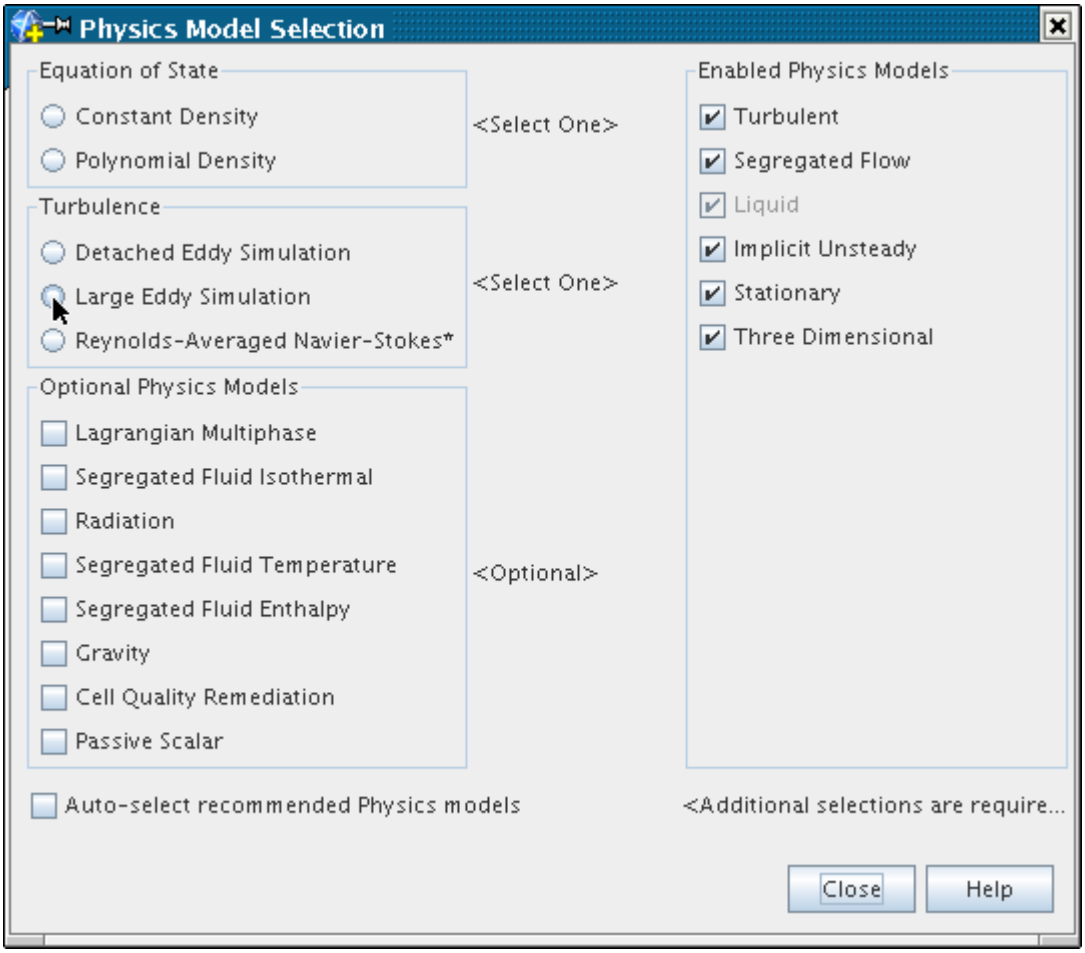
This section describes how to select the **Large Eddy Simulation** model and a **Subgrid Scale Turbulence** model with or without auto-select.

In order to select the **Large Eddy Simulation** model in the *Physics Model Selection* dialog for a physics continuum, it is first necessary to deactivate the **Auto-select recommended Physics models** checkbox before choosing the **Turbulent** option from the *Viscous Regime* group box.

Some model selections must be made before the *Viscous Regime* group box becomes visible. The following graphic illustrates this operation:

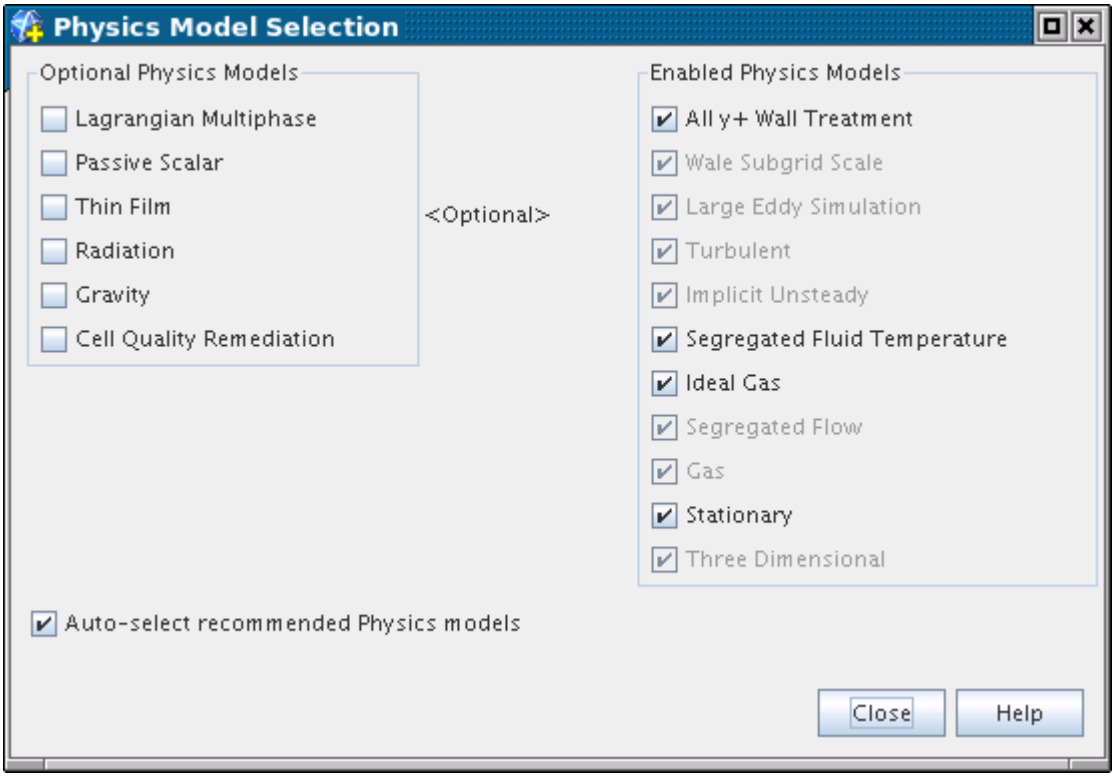


The *Physics Model Selection* dialog displays the *Turbulence* group box on completion.



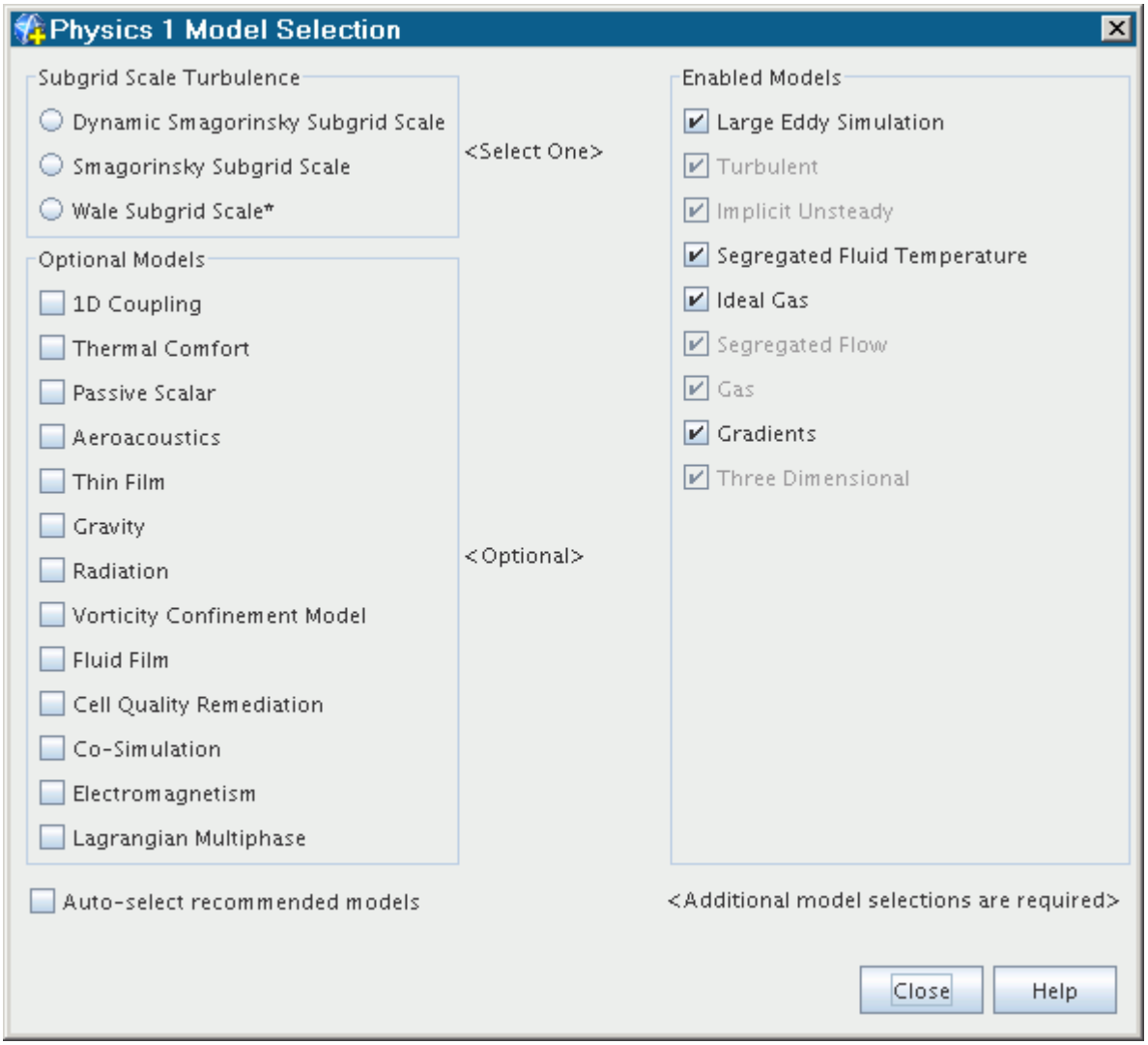
Using Auto-Select

Before choosing **Large Eddy Simulation** from the *Turbulence* group box, ensure that the **Auto-select recommended physics models** checkbox is ticked. This causes the **WALE Subgrid Scale** model and the **All y+ Wall Treatment** option to be selected, since it is the recommended subgrid scale modeling approach in STAR-CCM+.



Without Using Auto-Select

Before choosing **Large Eddy Simulation** from the *Turbulence* group box, ensure that the **Auto-select recommended physics models** checkbox is not ticked. You can now select the radio button corresponding to the **Smagorinsky Subgrid Scale** model, the **Dynamic Smagorinsky Subgrid Scale** model, or the **WALE Subgrid Scale** model.



Additionally, you can choose one of the options from the *LES Wall Treatment* group box.

LES and Subgrid Scale Turbulence Models Reference

The **Subgrid Scale Turbulence** models allow you to compute the subgrid scale viscosity μ_t to provide closure to the filtered Navier-Stokes equations.

Models Overview

Model Names and Abbreviations	**Dynamic Smagorinsky Subgrid Scale**	**DS SS**
	Smagorinsky Subgrid Scale	**S SS**
	WALE Subgrid Scale*	**W SS**
Theory	See Theory Guide—Large Eddy Simulation.	
Provided By	**[physics continuum]** > **Models** > *Subgrid Scale Turbulence*	

Example Node Path	**Continua** > **Physics 1** > **Models** > **Dynamic Smagorinsky Subgrid Scale**	
Requires	(Deactivate the **Auto-select recommended physics models** checkbox) *Space*: **Three Dimensional** *Time*: **Implicit Unsteady** or **PISO Unsteady** *Material*: one of **Gas**, **Liquid**, **Eulerian Multiphase**, **Multi-Component Gas**, **Multi-Component Liquid** *Flow*: any *Viscous Regime*: **Turbulent** *Turbulence*: **Large Eddy Simulation**	
Properties	See Properties Lookup.	
Activates	Physics Models	*Wall Distance*: **Exact Wall Distance** and **PDE Wall Distance**
		Wall Treatment: **All y+ Wall Treatment** and **Low y+ Wall Treatment**
	Methods	**Convection** See Methods.
	Reference Values	**Maximum Number of Eddies** See Reference Values.
	Initial Conditions	**Synthetic Turbulence Specification** See Initial Conditions.
	Boundary Inputs	**Synthetic Turbulence Specification** **Synthetic Turbulence Mass Flow Scaling Specification** See Boundary Settings.
	Solvers	**Dynamic Smagorinsky SGS Viscosity** (DS SS) **Smagorinsky SGS Viscosity** (S SS) **WALE SGS Viscosity** (W SS) See SGS Viscosity Solvers.
	Field Functions	**SGS Wall Damping Factor** (S SS) **Turbulent Viscosity** **Turbulent Viscosity Ratio** See Field Functions.

Properties Lookup

This table shows which properties are used by which **Subgrid Scale** model. Use the abbreviations that are given in the Models Overview above.

Unless you are thoroughly familiar with the theoretical aspects of this model and the discretization techniques used in STAR-CCM+, we recommend that you not make any changes within the *Expert* category. The values in that category reflect both the model's design and discretization approaches that have been optimized for accuracy and performance. Tampering with them may diminish the effectiveness of the model.

	DS SS	S SS	W SS
A Van Driest damping coefficient, Eqn. (1981).		✓	
Average Parameter When **On**, uses averaging in calculation of C_s^2 (see Eqn. (1988)).	✓		
Cs The coefficient C_s in Eqn. (1980).		✓	
Ct The coefficient C_t in Eqn. (2072).	✓	✓	✓
Cw The coefficient C_w in Eqn. (1990).			✓
Filter Width Ratio Ratio of test filter width to grid filter width $\frac{\hat{L}}{\tilde{L}}$ in Eqn. (1987).	✓		
Kappa The von Karman constant κ in Eqn. (1980) (S SS) and in Eqn. (1990) (W SS).		✓	✓
Limit Length Scale When **On**, the mixing length minimum is applied to the subgrid scale viscosity, see Eqn. (1980) (S SS) and Eqn. (1990) (W SS).		✓	✓
Minimum Cs^2 Value Clips values less than the specified value of C_s^2 (see Eqn. (1988)).	✓		
Maximum Cs^2 Value Clips values greater than the specified value of C_s^2 (see Eqn. (1988)).	✓		
Van Driest Damping Selects the Van Driest damping function, see Eqn. (1981): • **None**: No damping. • **Original**: Van Driest's original damping function.		✓	

	DS SS	S SS	W SS
• **Modified**: Van Driest's damping function modified according to Piomelli and co-workers [360].			

Methods

Convection

For the Segregated Flow Model, the following additional convection methods become available:

Method	Corresponding Method Node
Central This scheme is second-order accurate but is prone to dispersive error and stability problems. However, central-differencing is useful in large eddy simulations where upwind schemes accelerate the rate at which turbulent kinetic energy decays.	None.
Bounded-Central This scheme applies a boundedness criterion that makes the scheme more robust than central-differencing alone. Use this scheme for complex turbulent flows.	**Bounded Differencing** Provides the *Upwind Blending Factor*, which specifies the proportion of upwind differencing, ς_{ubf} related to Eqn. (1586). The default value provides the most robustness for the scheme. Reducing it would, in principle, increase accuracy. However, unless you are thoroughly familiar with the theoretical aspects of bounded differencing, do not change this property. The default value reflects optimization for accuracy and performance.

Reference Values

Maximum Number of Eddies

See Synthetic Eddy Method Reference.

Initial Conditions

Synthetic Turbulence Specification

See Synthetic Eddy Method Reference.

Boundary Settings

Note: Boundary types that do not require setting any conditions or values are not listed.

Flow Boundaries

The following boundary conditions are equal to all boundaries of type:

- Free Stream
- Mass Flow Inlet
- Velocity Inlet

Synthetic Turbulence Specification

See Synthetic Eddy Method Reference.

Synthetic Turbulence Mass Flow Scaling Specification

See Synthetic Eddy Method Reference.

SGS Viscosity Solver Properties

This section describes the solver that updates the subgrid scale viscosity μ_t in all the continua for which the subgrid scale model is activated. Since subgrid scale models do not contain any transport equations, there are no other solvers associated with the models.

Damping Factor Update Policy **(S SS)**

Controls the policy for updating the damping factor:

- **Time Step**: Updates will be triggered on a per time-step basis.
- **Iteration**: Updates will be triggered on a per iteration basis.

Damping Factor Update Frequency **(S SS)**

This parameter governs the frequency at which the damping factor f_v will be updated in the free stream.

The default is 1, which means that the update occurs every iteration. Increasing this parameter to a larger interval, say 10, might improve parallel performance in certain situations. This is because the free-stream values are stored in a KD tree that is broadcast to each parallel node. incurring a communication overhead.

Solver Frozen

When **On**, this prevents the solver from updating the solution while iterating. The property is **Off** by default.

Temporary Storage Retained

When **On**, retains temporary storage at the end of the iteration. Examples include the ap coefficients, the residual, the corrections, the reconstruction gradients and the cell gradients. These quantities will become available as field functions during subsequent iterations. This property if **Off** by default.

Field Functions

SGS Turbulent Kinetic Energy

Subgrid kinetic energy, k_{SGS} in Eqn. (2074).

This field function is made available to the simulation when you activate an aeroacoustics or combustion model along with a subgrid scale model.

SGS Wall Damping Factor (S SS)

Scalar field that represents the Van Driest damping function f_v defined in Eqn. (1981).

Turbulent Viscosity

Scalar field that represents the turbulent viscosity μ_t.

Turbulent Viscosity Ratio

Scalar field that represents the ratio of turbulent to laminar viscosity μ_t/μ.

LES Guidelines

This section provides guidelines for analysis of a combustor with spray, using Large Eddy Simulation and the Non-Adiabatic PPDF (Equilibrium or Flamelet) models.

Guidelines for LES differ from RANS guidelines in:

- Meshing
- Differencing Schemes
- Subgrid Scale Models
- Initialization
- Monitoring Solution
- Sampling

See LES Analysis Workflow.

Meshing

The fidelity of a scale resolving simulation is strongly related to the mesh: usually, the finer the mesh, the better the results. However, the available computational resources limit the number of cells.

To create a mesh that gives reasonably good results for an LES analysis, you can use one of the following approaches:

- Choose a mesh size that gives a cutoff wavenumber within the inertial subrange in the region of interest. Keep the ratio of turbulent kinetic energy to subgrid kinetic energy (k/ksgs) above 5 and, if possible, above 20. When possible, avoid stretching the mesh. To estimate k/ksgs, use the following steps:

  1. Estimate the turbulence kinetic energy from the field variance values of the three velocity components and define a field function using this estimate.
  2. Use the field function for the turbulent kinetic energy and SGS Turbulent Kinetic Energy (the field function for subgrid scale turbulent kinetic energy) to determine the ratio of the two kinetic energies.

  Carry out 3-D simulations if the scales in the Z direction are large and must be captured accurately for a reliable LES study.

- Run a preliminary RANS (K-Epsilon, K-Omega, or RST) simulation on an exploratory mesh [[365]. Using the RANS field functions Kolmogorov Length Scale and the Taylor Micro Scale, determine a reasonable cells size for the LES study.

  The Kolmogorov Length Scale denotes the smallest scales in the turbulent spectrum. At these smallest scales, LES results tend towards the results of a DNS. For this reason, the Kolmogorov Length Scale can be used to limit the local minimum cell size.

  The Taylor Micro Scale is an intermediate length scale, which lies at the dissipation region end of the inertial sub-range within the turbulence spectrum. Scales that are smaller than the Taylor Micro Scale are mainly viscous-driven. Addad and others [359] demonstrated that limiting the maximum cell size based on the Taylor Micro Scale can be used to construct a mesh that provides good results for scale resolving simulations.

  Therefore, using the Kolmogorov Length Scale η and the Taylor Micro Scale λ, you can determine a reasonable local cell size from $\eta < \Delta < \lambda$, where $\Delta = \left(\Delta_x \Delta_y \Delta_z\right)^{1/3}$.

See How Do I Conduct an Aeroacoustics Analysis? and Turbulent Time and Length Scales.

Discretization

To ensure temporal and spatial fidelity for good LES results, it is important to achieve a Convective Courant number (available as a field function) with a value close to 1 in the domain of interest. This number depends on the mesh size; if you have a large mesh size, reduce time-step size to improve the Convective Courant number value. Use a second-order temporal discretization and a time-step of 1E-5 s or less. For faster convergence per time-step, increase the under-relaxation factors. As a result, you can then reduce the number of inner iterations to 15 or less, for a faster turnaround time.

Use Bounded Central Differencing (BCD) with LES—with appropriate blending factor. A value of 0 for *Upwind Blending Factor* implies pure central differencing. A value of 1 implies pure upwinding. See Bounded Central-Differencing.

Subgrid Scale Model

Depending on the application, you can choose from:

- The Smagorinsky Subgrid Scale model. The default value for the Smagorinsky model constant is 0.1. For boundary layer flows and jets, consider using lower values. Higher values occur for homogeneous isotropic decaying turbulence. Take care to evaluate the value of the Smagorinsky model constant for each cases.
- The Dynamic Smagorinsky Subgrid Scale model
- The WALE Subgrid Scale model. The WALE model is computationally less expensive, more thoroughly validated, and more suitable for use near walls. Use the WALE model unless you require the Smagorinsky model for reasons of compatibility, standards, or legacy.

Initialization

A large proportion of computational time is spent in reaching a stage when it is possible to start collecting flow statistics for post-processing. The effects of initial conditions have to be eliminated before any time-averaging can begin. This process usually takes 2–5 flow-through cycles. To minimize the number of cycles needed:

- Use results from a RANS calculation to give the right mean fields.

- Get convergence on a coarse mesh and interpolate the results onto a fine mesh. This way, you can retain the larger turbulent scales, and the smaller scales can develop quickly.

Monitoring Solution

Always have several monitor points for quantities like velocity, to help you determine when to start averaging. Use time-steps as the triggers for these monitors, since the LES simulation is a transient simulation.

Monitor residuals to determine if the solution converges well within each time-step. Monitor other scalars of interest, (for example, Temperature or Velocity) versus inner iterations at different locations to determine if you have proper convergence within the specified inner iterations.

Sampling

Set Start Time:

Choose a time to start sampling. Typical start times are 2–5 flow-through times. Use one of the following methods to compute residence time:

- Track the residence time for a massless particle, starting with the converged RANS solution. See Modeling Residence Time.
- Divide combustor length by average axial velocity.

Define Field Monitors:

Define the following field monitors:

- Field Mean Velocity Magnitude
- Field Mean Temperature
- Field Mean Mixture Fraction
- Field Variance Velocity [i]
- Field Variance Velocity [j]
- Field Variance Velocity [k]

To define a field monitor:

1. Right-click the **Monitors** node, select **New Monitor**, then select **Field Mean** or **Field Variance**.
2. Select the monitor node. In the *Properties* window, select the **Field Function** and the **Parts** for the monitor.
3. Expand the monitor node and select the **Time-Step Frequency** node. In the *Properties* window, enter the *Start* time.
4. Rename the monitor node to indicate the selected function.

LES Analysis Workflow

Use the following steps:

1. Run the case using RANS until you reach a reasonable level of convergence. See RANS Simulation Settings.
2. Estimate the Lagrangian residence time.
3. Estimate the flow-through time.
4. Set up the LES models. See LES Simulation Settings.

5. Run the Lagrangian solver with all other solvers frozen. Use a time period of about twice the residence time of the spray. This practice ensures that the spray catches up with the flow field. See Lagrangian Update.
6. Unfreeze all the other solvers and run an LES analysis. See LES Analysis.

   If the time-step is small (about 1E-5 or lower), you can use a Lagrangian *Update Frequency* of once per time-step for a faster turnaround time.

RANS Simulation Settings

Select a RANS model. (See Reynolds-Averaged Navier-Stokes (RANS) Turbulence Models.)

For the **Non-Adiabatic PPDF Flamelet** combustion model:

- Select the following physics models:

Group Box	Model
Equation of state	**Ideal Gas**
Material	**Multi-Component Gas**
Time	**Steady**

- Under the **Multi-Component Gas** physics model node, set the following properties:

Node	Property	Setting
Mixture Properties		
└ **Molecular Weight**	*Method*	**PPDF Table**
└ **Specific Heat**	*Method*	**PPDF Table**

- Under the **PPDF Flamelet (Adiabatic)** physics model node, set the following properties:

Node	Property	Setting
PPDF Flamelet Table	*Identical Mixture Fraction Space*	Activated

For the **Non-Adiabatic PPDF Equilibrium** combustion model:

- Select the following physics models:

Group Box	Model
Equation of state	**Ideal Gas**
Material	**Multi-Component Gas**
Time	**Steady**

- Under the **Multi-Component Gas** physics model node, set the following properties:

Node	Property	Setting
Mixture Properties		
└ **Molecular Weight**	*Method*	**PPDF Table**

Node	Property	Setting
└ **Specific Heat**	*Method*	**PPDF Table**

- Under the **PPDF Flamelet (Adiabatic)** physics model node, set the following properties:

Node	Property	Setting
PPDF Equilibrium Table	*Identical Mixture Fraction Space*	Activated

For **solvers**, set the following properties:

Solver	Property	Setting
Lagrangian Multiphase	*Maximum Courant Number*	0.5
	Minimum Courant Number	0.3
└ **Two-Way Coupling**	*Under-Relaxation Factor*	0.75
└ **Steady** > **Update Frequency**	*Time-Step Update Frequency*	10
└ **Partitioning** > **Load balancing options**	*Rebalance Frequency*	200
Velocity	*Under-Relaxation Factor*	0.7
└ **Under-Relaxation Ramp**	*Ramp Method*	**Linear Ramp**
└ **Linear Ramp**	*Start Iteration*	1
	End Iteration	100
	Initial Value	0.5
Pressure	*Under-Relaxation Factor*	0.3
└ **Under-Relaxation Ramp**	*Ramp Method*	**Linear Ramp**
└ **Linear Ramp**	*Start Iteration*	1
	End Iteration	100
	Initial Value	0.1
Energy	*Under-Relaxation Factor*	0.9
└ **Under-Relaxation Ramp**	*Ramp Method*	**Linear Ramp**
└ **Linear Ramp**	*Start Iteration*	1
	End Iteration	100
	Initial Value	0.7
PPDF Combustion	*Under-Relaxation Factor*	0.9
└ **Under-Relaxation Ramp**	*Ramp Method*	**Linear Ramp**

Solver	Property	Setting
└ **Linear Ramp**	*Start Iteration*	1
	End Iteration	100
	Initial Value	0.7

LES Simulation Settings

Use the following steps:

1. Set the following models, in order, with **Auto-select recommended models** activated:

Group Box	Model	Action
Enabled Models	**Steady**	Deselect
Time	**Implicit Unsteady**	Select
Enabled Models	**Two-Layer All y+ Wall Treatment**	Deselect
Enabled Models	**Realizable Two-Layer K-Epsilon**	Deselect
Enabled Models	**K-Epsilon Turbulence**	Deselect
Enabled Models	**Reynolds-Averaged Navier-Stokes**	Deselect
Turbulence	**Large Eddy Simulation**	Select
Subgrid Scale Turbulence	**Wale Subgrid Scale**	Select
Enabled Models	**All y+ Wall Treatment**	(Selected automatically)

2. Set the following properties:

Node	Property	Setting
Segregated Flow model	*Convection*	**Bounded Central Differencing**
Non-Adiabatic PPDF model > **Mixture Fraction Variance**	*Method*	**Algebraic Relationship**

3. Set boundary conditions as required.
4. For **solvers**, set the following properties:

Solver	Property	Setting
Lagrangian Multiphase	*Maximum Courant Number*	0.5
	Minimum Courant Number	0.3
└ **Two-Way Coupling**	*Under-Relaxation Factor*	0.75
└ **Implicit Unsteady** > **Update Frequency**	*Time-Step Update Frequency*	Once per time-step

Solver	Property	Setting
└ **Partitioning** > **Load balancing options**	*Rebalance Frequency*	200
Velocity	*Under-Relaxation Factor*	0.7
Pressure	*Under-Relaxation Factor*	0.3
Energy	*Under-Relaxation Factor*	0.9
PPDF Combustion	*Under-Relaxation Factor*	0.9

5. Set **Stopping Criteria** > **Maximum Inner Iterations** to `15`. (You can reduce this number further if the residuals converge before 15 iterations and you set velocity URF to `0.8` and pressure URF to `0.4`.)
6. Set up all mean and variance monitors with appropriate collection start times (equal to the Lagrangian update time plus 4–5 flow-through times).
7. Set up appropriate reports and plots for post-processing.
8. Save the file.

Lagrangian Update

Next, run Lagrangian Update to synchronize the unsteady spray with the flow field:

1. Freeze all solvers except Implicit Unsteady and Lagrangian Multiphase.
2. Set **Stopping Criteria** > **Maximum Inner Iterations** to `1`.
3. Set **Stopping Criteria** > **Maximum Physical Time** to 2 × Lagrangian Residence Time.
4. Run Lagrangian Update for 2 Lagrangian residence times, to let the spray catch up with the flow.
5. Save the file.

LES Analysis

Finally, run LES with BCD:

1. Unfreeze all the solvers.
2. Set **Stopping Criteria** > **Maximum Inner Iterations** to `15`.
3. Set **Stopping Criteria** > **Maximum Physical Time** to 10 × flow-through time.
4. Run for ~5 flow-through times, or as required. Then switch on data sampling and continue running for another 5 flow-through times (so that total run time is 10 flow-through times).
5. Save the file.

Troubleshooting LES

The following troubleshooting quantities allow you to analyze the results of large eddy simulations:

- Subgrid Kinetic Energy
- Smagorinsky Subgrid Scale Wall Damping Function

Subgrid Kinetic Energy

The subgrid scale turbulent kinetic energy k_{SGS} (Eqn. (2074)) is a physical by-product of the STAR-CCM+ flow. LES always uses a subgrid scale model—Smagorinsky or WALE—which in turn directly computes turbulent viscosity. Within this computation is an implicit level of subgrid kinetic energy.

You may need to post-process the **SGS Turbulent Kinetic Energy** field function in certain kinds of STAR-CCM+ cases. For example, in Lagrangian simulations that include turbulent dispersion, subgrid kinetic energy is contained in turbulent fluctuations that the calculation has not taken into account. Therefore this field function can provide an approximate analysis of these unresolved fluctuations.

The values for subgrid kinetic energy are affected by the C_t parameter in the Smagorinsky expert properties or WALE expert properties.

Smagorinsky Subgrid Scale Wall Damping Function

The Smagorinsky subgrid scale model can give inaccurate scaling at walls, so the Van Driest damping function f_v (Eqn. (1981)) modifies the length scale closer to the wall. Visualizing this factor as a field function (**SGS Wall Damping Factor**) allows you to see the extent of the domain that is affected by this damping function.

Detached Eddy Simulation (DES)

Detached Eddy Simulation (DES) is a hybrid modeling approach that combines features of Reynolds-Averaged (RANS) simulation in some parts of the flow and Large Eddy Simulation (LES) in others.

The unsteady RANS equations are applicable to transient situations where the unsteadiness is either imposed, such as by a time-varying boundary condition, or is inherent, such as the vortex shedding in a massively separated flow. In the latter case, transient simulations often yield better results than attempting use a steady-state approach. However, successful unsteady RANS simulations require that the time scales of the turbulence be disparate from the mean-flow unsteadiness. Furthermore, the limitations of the turbulence model may preclude good unsteady results.

DES turbulence models are set up so that boundary layers and irrotational flow regions are solved using a base RANS closure model. However, the turbulence model is intrinsically modified so that, if the grid is fine enough, it will emulate a basic LES subgrid scale model in detached flow regions. In this way, one gets the best of both worlds: a RANS simulation in the boundary layers and an LES in the unsteady separated regions. For more background on DES, see [371].

STAR-CCM+ provides the DES modeling approach for three different RANS models:

- Spalart-Allmaras Detached Eddy Model
- Elliptic Blending K-Epsilon Detached Eddy Model
- SST K-Omega Detached Eddy Model

The following DES variants are implemented:

- Delayed Detached Eddy Simulation (DDES)
- Improved Delayed Detached Eddy Simulation (IDDES)

The DDES model incorporates a delay factor that enhances the ability of the model to distinguish between LES and RANS regions on meshes where spatial refinement could give rise to ambiguous behavior.

For IDDES, the subgrid length-scale includes a dependence on the wall distance. This approach allows RANS to be used in a much thinner near-wall region, in which the wall distance is much smaller than the boundary-layer thickness. IDDES was introduced in order to provide some WMLES (Wall-modelled LES) capabilities to the DES formulation.

While DES holds great promise for certain types of simulations, it must be cautioned that this technology is not the answer to all turbulence modeling problems. The creation of suitable grids is something of an art, and an unsteady simulation is required, with statistics gathered to obtain a mean field as in a LES.

Contents:

Selecting a DES Model
Spalart-Allmaras Detached Eddy Model Reference
Elliptic Blending K-Epsilon Detached Eddy Model Reference
SST K-Omega Detached Eddy Model Reference
Controlling a Detached Eddy Simulation

See Also:

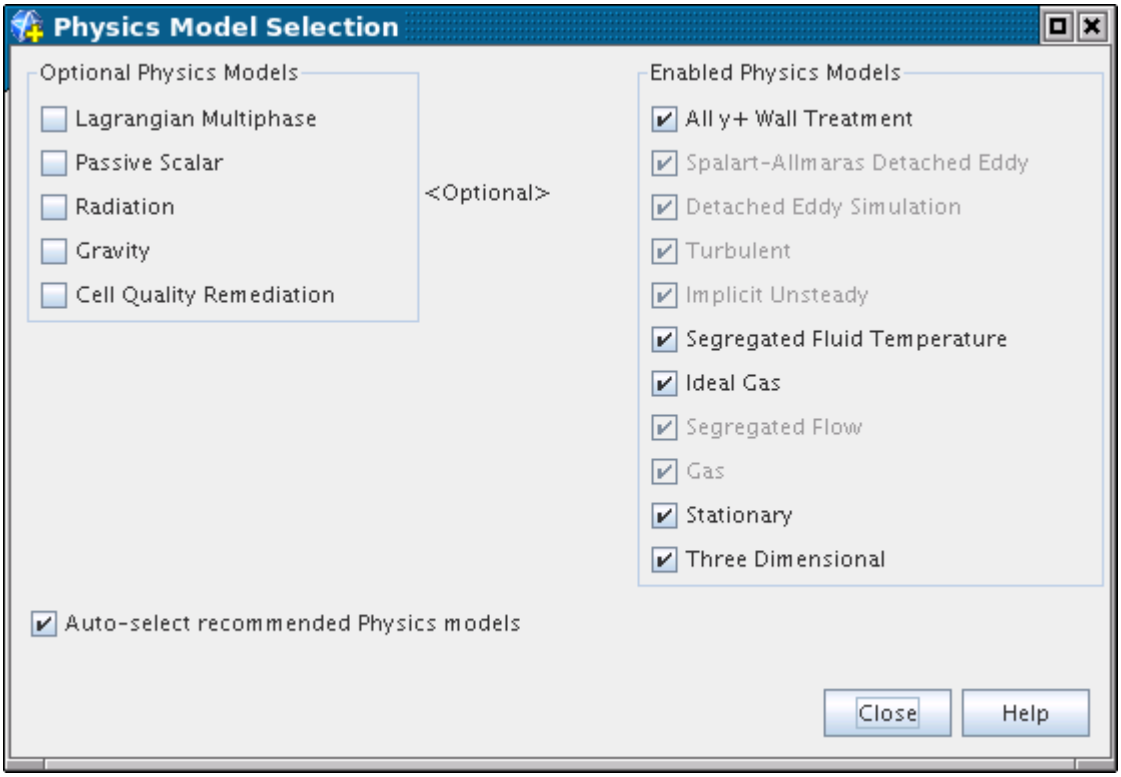
Bounded Differencing

Selecting a DES Model

This section describes how to select a Detached Eddy Simulation model either with or without auto-select.

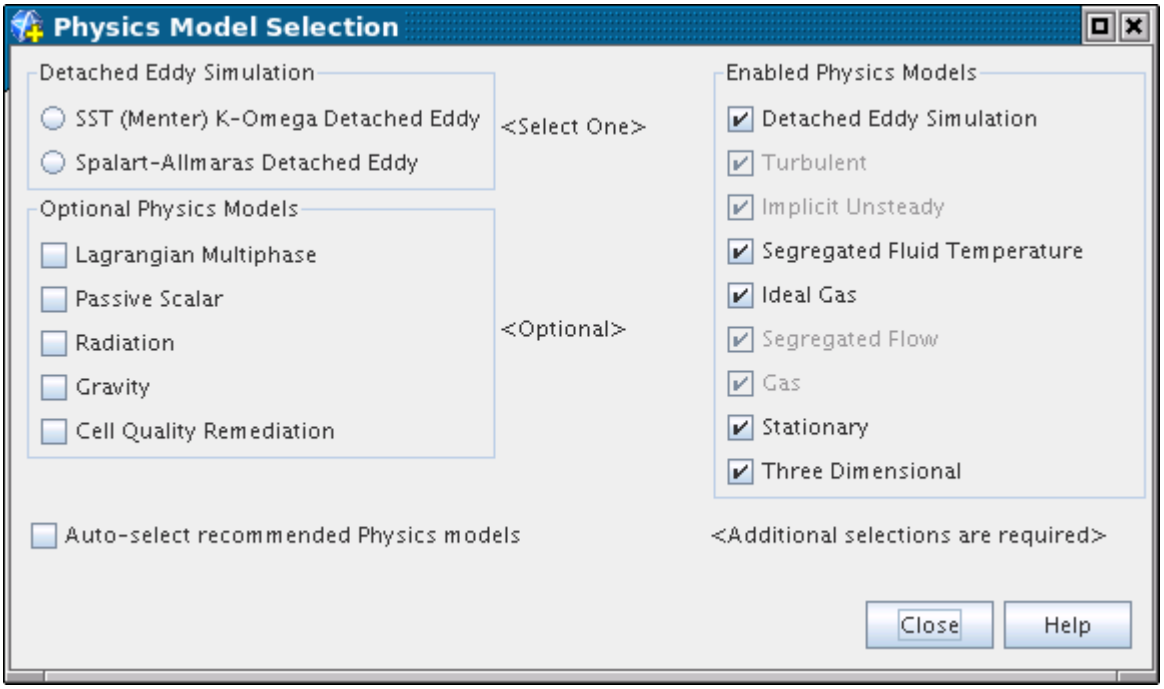
Using Auto-Select

This section describes how to select a turbulence model that supports DES using automatic selection of recommended models. When selecting a turbulence modeling approach, ensure that the **Auto-select recommended physics models** checkbox is ticked before choosing Detached Eddy Simulation. This will cause the Spalart-Allmaras DES model to be selected, since it is the recommended detached eddy simulation modeling option.



Without Using Auto-Select

This section describes how to select a turbulence model that supports DES without using automatic selection of recommended models. When selecting a turbulence modeling approach, ensure that the **Auto-select recommended physics models** checkbox is not ticked before choosing Detached Eddy Simulation. You can now select the radio button corresponding to the Spalart-Allmaras DES model or the SST (Menter) K-Omega Detached Eddy model.



Spalart-Allmaras Detached Eddy Model Reference

The **Spalart-Allmaras Detached Eddy** model combines features of the Standard Spalart-Allmaras RANS model in the boundary layers with a large eddy simulation (LES) in unsteady separated regions.

The DES version of the Spalart-Allmaras model contains modifications to the damping functions recommended by Squires and others [373]. These modifications do not affect the predictions of fully turbulent flows, but prevent spurious upstream propagation of the eddy viscosity into attached laminar regions.

For the Spalart-Allmaras model, the DDES approach is implemented in STAR-CCM+. In addition, the IDDES formulation is available as an option.

Theory	See Theory Guide—Spalart-Allmaras DES.	
Provided by	**[physics continuum]** > **Models** > *Detached Eddy Simulation*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Spalart-Allmaras Detached Eddy**	
Requires	(Deactivate the **Auto-select recommended physics models** checkbox.) *Space*: **Three Dimensional** *Time*: **Implicit Unsteady** or **PISO Unsteady** *Material*: one of **Gas**, **Liquid**, **Eulerian Multiphase**, **Multi-Component Gas**, **Multi-Component Liquid** *Viscous Regime*: **Turbulent** *Turbulence*: **Detached Eddy Simulation**	
Properties	See Properties.	
Activates	Physics Models	*Wall Distance*: **Exact Wall Distance** and **PDE Wall Distance**

		Wall Treatment: **All y+ Wall Treatment** and **Low y+ Wall Treatment**
	Model Controls (child nodes)	**Curvature Correction Parameters** (optional, see Curvature Correction Option) **IDDES Coefficients** (optional, see Formulation Option)
	Methods	**Convection** See Methods.
	Reference Values	**Maximum Number of Eddies** See Reference Values.
	Initial Conditions	**Turbulence Specification** **Synthetic Turbulence Specification** See Initial Conditions.
	Boundary Inputs	See Boundary Settings.
	Region Inputs	See Region Settings.
	Solvers	**Spalart-Allmaras Turbulence** **Spalart-Allmaras Turbulent Viscosity** See Spalart-Allmaras Solvers Reference .
	Monitors	**Sa_nut** (modified diffusivity)
	Field Functions	See Field Functions.

Properties

The **Spalart-Allmaras Detached Eddy** model owns the same properties as the **Standard Spalart-Allmaras** model, see Spalart-Allmaras Model Reference - Properties Lookup. The following properties are specific to the DES version of the model:

Cdes

The DES length scale coefficient C_{DES} in Eqn. (1999) and Eqn. (2003).

CdesTimeLim

The DES blended upwind scheme time scale coefficient C_{desT} in Eqn. (2039).

Cv2

The coefficient C_{v2} in Eqn. (2016).

Formulation Option

Selects the DES formulation to use.

- **DDES**: Selects the DDES formulation.
- **IDDES**: Selects the IDDES formulation. This formulation requires the coefficients C_t in Eqn. (2009), C_l in Eqn. (2010), and C_{dt} in Eqn. (2014).

IDDES Coefficients

Child node that provides the parameters *Ct*, *Cl*, and *Cdt*.

Low Re Correction

Controls whether to apply a low Reynolds number correction.

When **On**, the low Reynolds number correction factor ψ in Eqn. (1999) and Eqn. (2003) is computed as in Eqn. (2002). When **Off**, ψ is set to 1.

Methods

Convection

For the Segregated Flow Model, the following additional convection methods become available:

Method	Corresponding Method Node
Hybrid This scheme blends second-order upwind and central differencing.	None.
Bounded-Central This scheme applies a boundedness criterion that makes the scheme more robust than central-differencing alone. Use this scheme for complex turbulent flows.	**Bounded Differencing** Provides the *Upwind Blending Factor*, which specifies the proportion of upwind differencing, ς_{ubf} related to Eqn. (1586). The default value provides the most robustness for the scheme. Reducing it would, in principle, increase accuracy. However, unless you are thoroughly familiar with the theoretical aspects of bounded differencing, do not change this property. The default value reflects optimization for accuracy and performance.
Hybrid-BCD This scheme blends second-order upwind and bounded-central differencing. This scheme is the default scheme for detached eddy simulation (DES).	As for **Bounded-Central**.

Reference Values

Maximum Number of Eddies

See Synthetic Eddy Method Reference.

Initial Conditions

Turbulence Specification

Controls how you define the turbulence profile for initialization.

The available methods and corresponding value nodes are equal to the **Standard Spalart-Allmaras** model, see Spalart-Allmaras Initial Conditions Reference.

Synthetic Turbulence Specification

See Synthetic Eddy Method Reference.

Boundary Settings

Note: Boundary types that do not require setting any conditions or values are not listed.

Flow Boundaries

The following boundary conditions are equal to all boundaries of type:

- Free Stream
- Mass Flow Inlet
- Pressure Outlet
- Stagnation Inlet
- Velocity Inlet

Turbulence Specification

Controls how you define the turbulence profile at flow boundaries.

The available methods and corresponding value nodes are equal to the **Standard Spalart-Allmaras** model, see Spalart-Allmaras Boundaries Reference.

Synthetic Turbulence Specification (except Stagnation Inlet and Pressure Outlet)

See Synthetic Eddy Method Reference.

Synthetic Turbulence Mass Flow Scaling Specification (except Stagnation Inlet and Pressure Outlet)

See Synthetic Eddy Method Reference.

Region Settings

The conditions that are required for each region type that is supported by the **Spalart-Allmaras Detached Eddy** model are equal to the **Standard Spalart-Allmaras** model, see Spalart-Allmaras Regions Reference.

Field Functions

The **Spalart-Allmaras Detached Eddy** model owns the same field functions as the **Spalart-Allmaras** model, see Spalart-Allmaras Field Functions Reference. The following field functions are specific to the DES version of the model:

DES Low Re Correction Factor

Scalar field that represents the low Reynolds number correction function ψ defined in Eqn. (2002).

DES Upwind Blending Factor

Scalar field that represents the blending function for the hybrid upwind scheme.

Detached Eddy Length Scale

Scalar field that represents the cell length scale that is used for detached eddy simulation, see Δ in Eqn. (1999) and Eqn. (2015).

IDDES Blending Function

Scalar field that represents the blending function $\tilde{f}_d$ defined in Eqn. (2013) that is used for detached eddy simulation with the IDDES formulation option. This field is only available when you activate the *Temporary Storage Retained* property of the **Spalart-Allmaras Turbulence** solver.

IDDES Elevating Function

Scalar field that represents the elevating function f_e defined in Eqn. (2006) that is used for detached eddy simulation with the IDDES formulation option. This field is only available when you activate the *Temporary Storage Retained* property of the **Spalart-Allmaras Turbulence** solver.

IDDES Length Scale

Scalar field that represents the cell length scale that is used for detached eddy simulation with the IDDES formulation option Δ_{IDDES} , see Eqn. (2015).

Elliptic Blending K-Epsilon Detached Eddy Model Reference

The **EB K-Epsilon Detached Eddy** model combines features of the Elliptic Blending RANS model in the boundary layers with a large eddy simulation (LES) in unsteady separated regions.

The EB K-Epsilon model provides a better description of near-wall behavior than other $k-\epsilon$ models and makes the transition to turbulence more smoothly. For the EB K-Epsilon model, the DDES approach is implemented in STAR-CCM+.

Theory	See Theory Guide—Elliptic Blending DES.
Provided by	**[physics continuum]** > **Models** > *Detached Eddy Simulation*
Example Node Path	**Continua** > **Physics 1** > **Models** > **EB K-Epsilon Detached Eddy**
Requires	(Deactivate the **Auto-select recommended physics models** checkbox.) *Space*: **Three Dimensional** *Time*: **Implicit Unsteady** or **PISO Unsteady** *Material*: one of **Gas**, **Liquid**, **Eulerian Multiphase**, **Multi-Component Gas**, **Multi-Component Liquid** *Viscous Regime*: **Turbulent** *Turbulence*: **Detached Eddy Simulation**
Properties	See Properties.

Activates	Physics Models	*Wall Distance*: **Exact Wall Distance** and **PDE Wall Distance**
		Wall Treatment: **All y+ Wall Treatment** and **Low y+ Wall Treatment**
	Methods	**Convection** See Methods.
	Reference Values	**Maximum Number of Eddies** See Reference Values.
	Initial Conditions	**Turbulence Specification** **Synthetic Turbulence Specification** See Initial Conditions.
	Boundary Inputs	See Boundary Settings.
	Region Inputs	See Region Settings.
	Solvers	**EB K-Epsilon Turbulence** **K-Epsilon Turbulent Viscosity** See K-Epsilon Solvers Reference
	Monitors	**Alpha** **Phi** **Tdr** **Tke**
	Field Functions	See Field Functions.

Properties

The **EB K-Epsilon Detached Eddy** model owns the same properties as the **EB K-Epsilon** model, see K-Epsilon Models Reference - Properties Lookup. The following properties are specific to the DES version of the model:

Cdes

The DES length scale coefficient C_{DES} in Eqn. (2018).

CdesTimeLim

The DES blended upwind scheme time scale coefficient C_{desT} in Eqn. (2039).

Methods

Convection

For the Segregated Flow Model, the following additional convection methods become available:

Method	Corresponding Method Node
Hybrid	None.

This scheme blends second-order upwind and central differencing.	
Bounded-Central This scheme applies a boundedness criterion that makes the scheme more robust than central-differencing alone. Use this scheme for complex turbulent flows.	**Bounded Differencing** Provides the *Upwind Blending Factor*, which specifies the proportion of upwind differencing, ς_{ubf} related to Eqn. (1586). The default value provides the most robustness for the scheme. Reducing it would, in principle, increase accuracy. However, unless you are thoroughly familiar with the theoretical aspects of bounded differencing, do not change this property. The default value reflects optimization for accuracy and performance.
Hybrid-BCD This scheme blends second-order upwind and bounded-central differencing. This scheme is the default scheme for detached eddy simulation (DES).	As for **Bounded-Central**.

Reference Values

Maximum Number of Eddies

See Synthetic Eddy Method Reference.

Initial Conditions

Turbulence Specification

Controls how you define the turbulence profile for initialization.

The available methods and corresponding value nodes are equal to the **EB K-Epsilon** model, see K-Epsilon Initial Conditions Reference.

Synthetic Turbulence Specification

See Synthetic Eddy Method Reference.

Boundary Settings

Note: Boundary types that do not require setting any conditions or values are not listed.

Flow Boundaries

The following boundary conditions are equal to all boundaries of type:

- Free Stream

- Mass Flow Inlet
- Pressure Outlet
- Stagnation Inlet
- Velocity Inlet

Turbulence Specification

Controls how you define the turbulence profile at flow boundaries.

The available methods and corresponding value nodes are equal to the **EB K-Epsilon** model, see K-Epsilon Boundaries Reference.

Synthetic Turbulence Specification (except Stagnation Inlet and Pressure Outlet)

See Synthetic Eddy Method Reference.

Synthetic Turbulence Mass Flow Scaling Specification (except Stagnation Inlet and Pressure Outlet)

See Synthetic Eddy Method Reference.

Region Settings

The conditions that are required for each region type that is supported by the **EB K-Epsilon Detached Eddy** model are equal to the **EB K-Epsilon** model, see K-Epsilon Regions Reference.

Field Functions

The **EB K-Epsilon Detached Eddy** model owns the same field functions as the **EB K-Epsilon** model, see K-Epsilon Field Functions Reference. The following field functions are specific to the DES version of the model:

DES Upwind Blending Factor

Scalar field that represents the blending function for the hybrid upwind scheme, see Using the Hybrid Second-Order Upwind/Central Difference Scheme.

Detached Eddy Length Scale

Scalar field that represents the cell length scale that is used for detached eddy simulation, see Δ in Eqn. (2018).

SST K-Omega Detached Eddy Model Reference

The **SST (Menter) K-Omega Detached Eddy** model combines features of the SST K-Omega RANS model in the boundary layers with a large eddy simulation (LES) in unsteady separated regions.

For the SST K-Omega model, the DDES approach is implemented in STAR-CCM+. In addition, the IDDES formulation is available as an option.

Theory	See Theory Guide—SST K-Omega DES.
Provided by	**[physics continuum]** > **Models** > *Detached Eddy Simulation*
Example Node Path	**Continua** > **Physics 1** > **Models** > **SST (Menter) K-Omega Detached Eddy**

Requires	(Deactivate the **Auto-select recommended physics models** checkbox.) *Space*: **Three Dimensional** *Time*: **Implicit Unsteady** or **PISO Unsteady** *Material*: one of **Gas**, **Liquid**, **Eulerian Multiphase**, **Multi-Component Gas**, **Multi-Component Liquid** *Viscous Regime*: **Turbulent** *Turbulence*: **Detached Eddy Simulation**	
Properties	See Properties.	
Activates	Physics Models	*Wall Distance*: **Exact Wall Distance** and **PDE Wall Distance**
		Wall Treatment: **All y+ Wall Treatment**, **High y+ Wall Treatment**, and **Low y+ Wall Treatment**
	Model Controls (child nodes)	**Compressibility Parameters** (optional, see Compressibility Correction) **Realizability Coefficient** (optional, see Realizability Option) **Curvature Correction Parameters** (optional, see Curvature Correction Option) **Low Re Damping Parameters** (optional, see Low Re Damping Modification) **IDDES Coefficients** (optional, see Formulation Option)
	Methods	**Convection** See Methods.
	Reference Values	**Maximum Number of Eddies** See Reference Values.
	Initial Conditions	**Turbulence Specification** **Synthetic Turbulence Specification** See Initial Conditions.
	Boundary Inputs	See Boundary Settings.
	Region Inputs	See Region Settings.
	Solvers	**K-Omega Turbulence** **K-Omega Turbulent Viscosity** See K-Omega Solvers Reference
	Monitors	**Sdr** **Tke**
	Field Functions	See Field Functions.

Properties

The **SST (Menter) K-Omega Detached Eddy** model owns the same properties as the **SST (Menter) K-Omega** model, see K-Omega Models Reference - Properties Lookup. The following properties are specific to the DES version of the model:

Formulation Option

Selects the DES formulation to use.

- **DDES**: Selects the DDES formulation.
- **IDDES**: Selects the IDDES formulation. This formulation requires the coefficients C_t in Eqn. (2030), C_l in Eqn. (2031), and C_{dt} in Eqn. (2035).

IDDES Coefficients

Child node that provides the parameters *Ct*, *Cl*, and *Cdt*.

CdesKw

The DES length scale coefficient for the model's K-Omega branch, see $C_{\mathrm{DES},k\text{-}\omega}$ in Eqn. (2022).

CdesKe

The DES length scale coefficient for the model's K-Epsilon branch, see $C_{\mathrm{DES},\,k\text{-}\varepsilon}$ in Eqn. (2022).

CdesTimeLim

The DES blended upwind scheme time scale coefficient C_{desT} , calculated as for the Spalart-Allmaras turbulence model in Eqn. (2039).

Methods

Convection

For the Segregated Flow Model, the following additional convection methods become available:

Method	Corresponding Method Node
Hybrid This scheme blends second-order upwind and central differencing.	None.
Bounded-Central This scheme applies a boundedness criterion that makes the scheme more robust than central-differencing alone. Use this scheme for complex turbulent flows.	**Bounded Differencing** Provides the *Upwind Blending Factor*, which specifies the proportion of upwind differencing, ς_{ubf} related to Eqn. (1586). The default value provides the most robustness for the scheme. Reducing it would, in principle, increase accuracy. However, unless you are thoroughly familiar with the theoretical aspects of bounded differencing, do not change this property. The default value reflects optimization for accuracy and performance.
Hybrid-BCD This scheme blends second-order upwind and bounded-central differencing. This scheme is	As for **Bounded-Central**.

the default scheme for detached eddy simulation (DES).	

Reference Values

Maximum Number of Eddies

See Synthetic Eddy Method Reference.

Initial Conditions

Turbulence Specification

Controls how you define the turbulence profile for initialization.

The available methods and corresponding value nodes are equal to the **SST (Menter) K-Omega** model, see K-Omega Initial Conditions Reference.

Synthetic Turbulence Specification

See Synthetic Eddy Method Reference.

Boundary Settings

Note: Boundary types that do not require setting any conditions or values are not listed.

Flow Boundaries

The following boundary conditions are equal to all boundaries of type:

- Free Stream
- Mass Flow Inlet
- Pressure Outlet
- Stagnation Inlet
- Velocity Inlet

Turbulence Specification

Controls how you define the turbulence profile at flow boundaries.

The available methods and corresponding value nodes are equal to the **SST (Menter) K-Omega** model, see K-Omega Boundaries Reference.

Synthetic Turbulence Specification (except Stagnation Inlet and Pressure Outlet)

See Synthetic Eddy Method Reference.

Synthetic Turbulence Mass Flow Scaling Specification (except Stagnation Inlet and Pressure Outlet)

See Synthetic Eddy Method Reference.

Region Settings

The conditions that are required for each region type that is supported by the **SST (Menter) K-Omega Detached Eddy** model are equal to the **SST (Menter) K-Omega** model, see K-Omega Regions Reference.

Field Functions

The **SST (Menter) K-Omega Detached Eddy** model owns the same field functions as the **SST (Menter) K-Omega** model, see K-Omega Field Functions Reference. The following field functions are specific to the DES version of the model:

DDES Correction Factor

Scalar field that represents the correction factor ϕ defined in Eqn. (2021). This field is only available when you activate the *Temporary Storage Retained* property of the **K-Omega Turbulence** solver.

DES Upwind Blending Factor

Scalar field that represents the blending function for the hybrid upwind scheme.

Detached Eddy Length Scale

Scalar field that represents the cell length scale that is used for detached eddy simulation, expressed as Δ in Eqn. (2021) and Eqn. (2036).

IDDES Blending Function

Scalar field that represents the blending function $\tilde{f}_d$ defined in Eqn. (2034) that is used for detached eddy simulation with the IDDES formulation option. This field is only available when you activate the *Temporary Storage Retained* property of the **K-Omega Turbulence** solver.

IDDES Elevating Function

Scalar field that represents the elevating function f_e defined in Eqn. (2027) that is used for detached eddy simulation with the IDDES formulation option. This field is only available when you activate the *Temporary Storage Retained* property of the **K-Omega Turbulence** solver.

IDDES Length Scale

Scalar field that represents the cell length scale that is used for detached eddy simulation with the IDDES formulation option Δ_{IDDES} , see Eqn. (2036).

Controlling a Detached Eddy Simulation

A number of guidelines that concern the difference scheme are advised when setting up a Detached Eddy Simulation.

Using the Hybrid Second-Order Upwind/Central Difference Scheme

The second-order upwind is the most accurate scheme for steady-state calculations, while the kinetic energy-preserving properties of the central differencing scheme make it appropriate for large eddy simulation (LES).

When performing detached eddy simulation (DES), the Segregated Flow model makes available a hybrid second-order upwind/central scheme where the blending factor is computed according to the features of the flow field.

It should be noted that, like DES itself, this feature is not a mature development, and some experimentation might be required to tune it. It is suggested that the simulation be initiated using the second-order upwind scheme. Then the coefficient CdesTimeLim can be adjusted so that the field function quantity DES Upwind Blending Factor looks like it is indicating unity for areas such as attached shear layers and irrotational free-stream flow, and zero for areas where LES is required such as massively separated regions. Once the field function looks correct, the hybrid second-order upwind/central scheme can be activated.

Initialization and Inflow for LES and DES

The specification of realistic initial and inflow boundary conditions is of paramount importance in Large Eddy and Detached Eddy Simulations of spatially developing flows.

As an inadequate amount of information introduces sources of error, a number of methods have been proposed for the specification of initialization and inflow, with varying degrees of complexity and computational cost.

The simplest approach consists of superimposing Gaussian noise to the mean velocity. However, this approach has the disadvantage of producing a decorrelated velocity field that is incapable of sustaining the growth of coherent turbulent structures.

In STAR-CCM+, the Synthetic Eddy Method (SEM) is implemented, which automatically provides turbulent eddies across inflow boundaries and initializes the flow domain with a perturbated flow field. This method provides a simple, yet effective, approach for generating the instantaneous velocity field that is required.

Synthetic Eddy Method

The Synthetic Eddy Method, proposed by Jarrin et al. [377], retains the conceptual basis of the vortex method [375], but is more flexible and virtually mesh-independent.

Within the SEM framework, the turbulent flow field is seen as a superposition of spinning eddies, whose spin (clockwise or counterclockwise), and position, are drawn from a normalized uniform distribution. The size of the eddies is the characteristic scale of turbulence. As eddies must span at least two cells in order to produce a correlated signal, the minimum mesh-spacing imposes a limit on this turbulence scale.

A user-assigned Reynolds stress tensor provides the correlation function that the method requires. If this stress tensor is not known, the correlation function uses only normal components that are directly derived from a given turbulence intensity (assumption of isotropy).

The synthetic eddies generated at the inflow are convected and recycled in the computational domain with the mean inflow velocity.

The turbulent structures that SEM produces are only an approximation of real turbulence, and must be allowed to develop the proper correlations. The correlations develop naturally as the eddies are convected downstream. To allow this behavior to occur, it is important to leave sufficient distance between the inflow boundary and the region of interest. The necessary distance varies from case to case, but in channel flow simulations, a distance of at least ten half-heights is required.

Note: The Synthetic Eddy Method is incompatible with motion. If you attempt to combine SEM with a motion, an error message is displayed.

For more details, see Theory Guide—Synthetic Eddy Method.

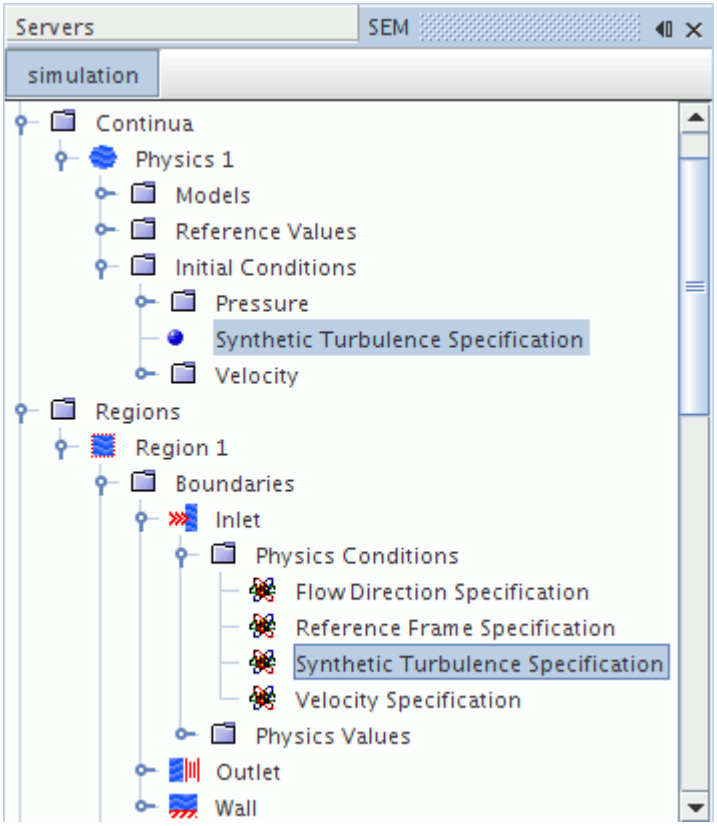
Contents:

Setting Up Synthetic Eddy Method
Synthetic Eddy Method Reference

Setting Up Synthetic Eddy Method

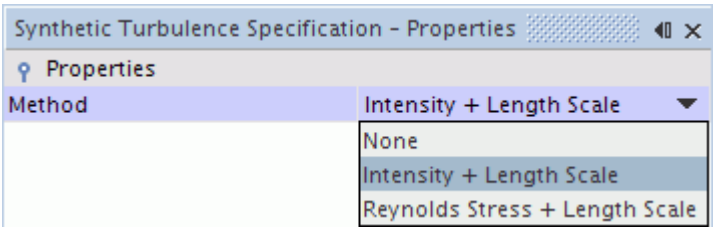
The Synthetic Eddy Method is activated when any of the LES or DES turbulence models are selected.

You can access its settings through the **Synthetic Turbulence Specification** node which is located under both the **Initial Conditions** node of a physics continuum and **Physics Conditions** node of an inflow boundary.



To activate SEM:

- Specify either Intensity + Length Scale or Reynolds Stress + Length Scale in the *Method* property of the *Properties* window.



- Provide either a value or distribution for the turbulence intensity or the six components of the (symmetric) Reynolds stress tensor.

The values for turbulence intensity or the Reynolds stress tensor may come from experimental data, from precursor RANS or LES simulations, or from some other approximation. If running a precursor simulation, Reynolds stress data can only be obtained by running an LES simulation, or a RANS simulation using a Reynolds Stress Transport turbulence model. For other turbulence models, the isotropic turbulent intensity can be computed from the turbulent kinetic energy.

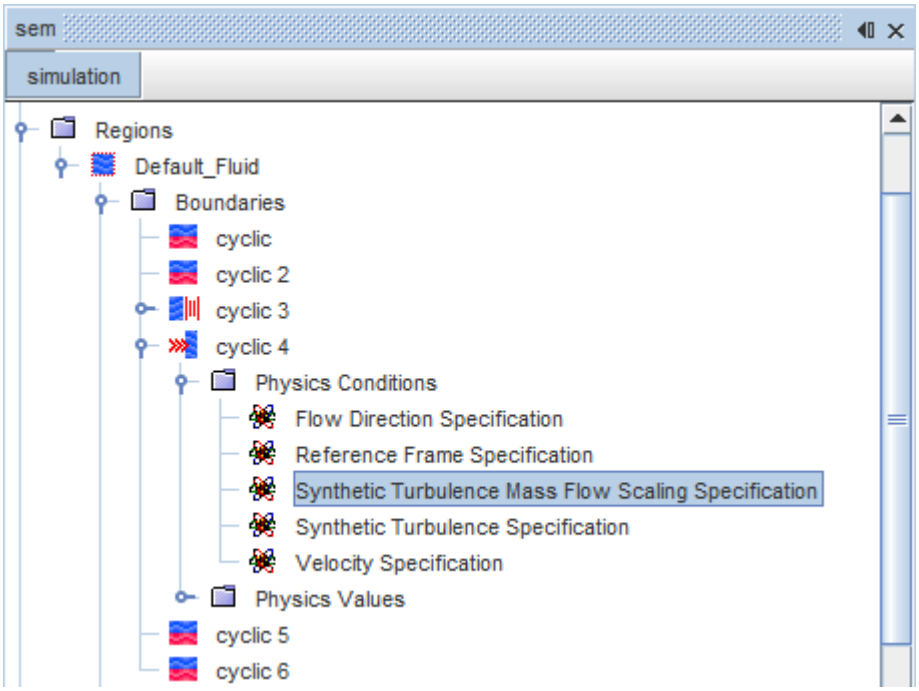
In many cases, a constant length scale, representative of eddy size, is sufficient for the Turbulent Length Scale. When using a RANS precursor simulation, the integral length scale can be calculated and used. For wall bounded flows, the wall distance may be a suitable approximation of the length scale when this information is not available from elsewhere. If the given length scale is inappropriate for the flow, the resolved turbulence may take much longer to develop, or disappear altogether.

When computing the SEM velocities, the number of eddies that are used in the calculation is inversely proportional to the cube of the length scale. To prevent the number of eddies becoming too large and affecting run-time and memory requirements, it is good practice to limit the minimum value of the length scale.

Synthetic Turbulence Mass Flow Scaling

If you use the Synthetic Eddy Method at inflow boundaries, there is no guarantee of constant mass flow. This can cause problems for the solver, especially for incompressible cases. In such cases, use the Synthetic Turbulence Mass Flow Scaling Specification to add a correction that maintains constant mass flow by scaling the computed fluctuations.

This specification is available whenever the Synthetic Turbulence Specification is available.



Synthetic Eddy Method Reference

This section details the reference values, initial conditions, and boundary settings for the Synthetic Eddy Method.

Reference Values

Maximum Number of Eddies

Sets an upper limit on the number of initial eddies that are generated within the flow domain by the Synthetic Eddy Method. If the maximum limit is reached, a warning is shown in the *Output* window, as this could potentially destroy the desired effect of the initial turbulence. It should not be set too low.

Initial Conditions

Synthetic Turbulence Specification

Controls how to calculate an isotropic field of velocity fluctuations that are superimposed on the specified mean velocity.

Method	Corresponding Physics Value Nodes
None Leaves the specified mean velocity unperturbed.	None
Intensity + Length Scale	**Turbulence Intensity** Scalar profile value to specify I.

	Turbulent Length Scale Scalar profile value to specify the eddy turbulent length scale σ_k.
Reynolds Stress + Length Scale	**Turbulent Length Scale** Scalar profile value to specify the eddy turbulent length scale σ_k. **Reynolds Stresses** Scalar profile values to specify the components of the Reynolds stress tensor: • **UU Reynolds Stress** • **UV Reynolds Stress** • **UW Reynolds Stress** • **VV Reynolds Stress** • **VW Reynolds Stress** • **WW Reynolds Stress**

Boundary Settings

Note: Boundary types that do not require setting any conditions or values are not listed.

Flow Boundaries

The following boundary conditions are available for all boundaries of type:

- Free Stream
- Mass Flow Inlet
- Velocity Inlet

Synthetic Turbulence Specification

Controls how to calculate an isotropic field of velocity fluctuations that are superimposed on the specified mean velocity at flow boundaries. For the available methods and corresponding value nodes, see Initial Conditions—Synthetic Turbulence Specification.

Synthetic Turbulence Mass Flow Scaling Specification

When **On**, scales fluctuations to maintain constant mass flow across boundaries.

Wall Treatment

Walls are a source of vorticity in most flow problems of practical importance. Therefore, an accurate prediction of flow and turbulence parameters across the wall boundary layer is essential.

The inner region of the boundary layer can be split up into three sublayers. In each of them the flow has different characteristics and can be modeled using different empirical approaches:

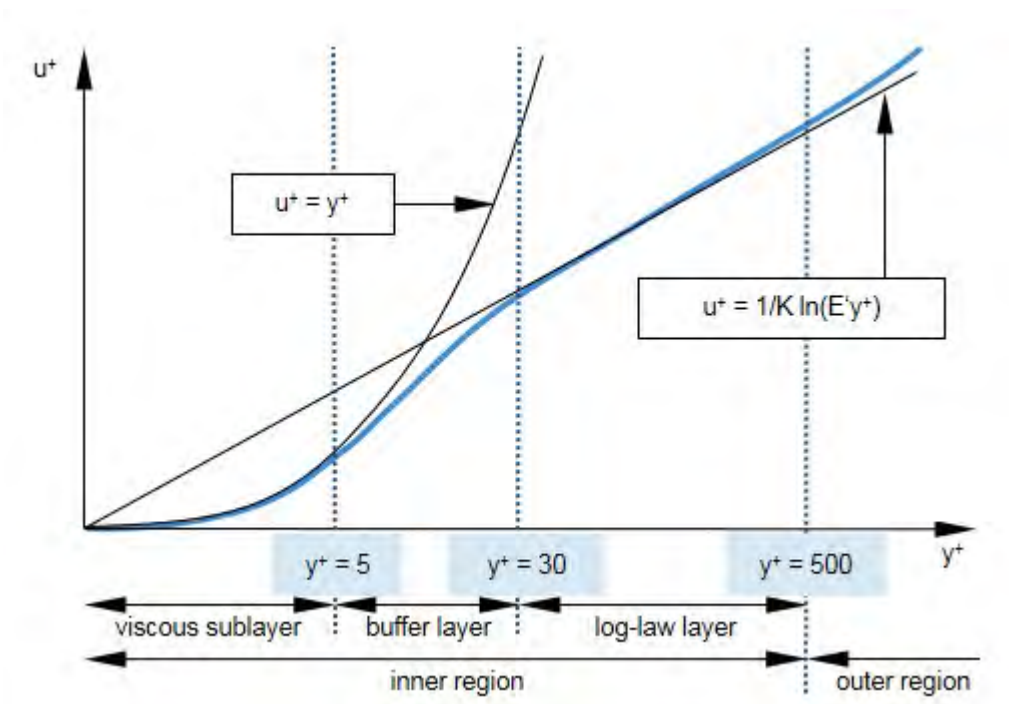
- Viscous sublayer

  The fluid layer in contact with the wall is dominated by viscous effects and is almost laminar. The mean flow velocity only depends on the fluid density, viscosity, distance from the wall, and the wall shear stress.
- Log-law layer

  The turbulent log-law layer is dominated equally by viscous and turbulent effects.
- Buffer layer

  The buffer layer is a transitional layer between the viscous sublayer and the log-law layer.

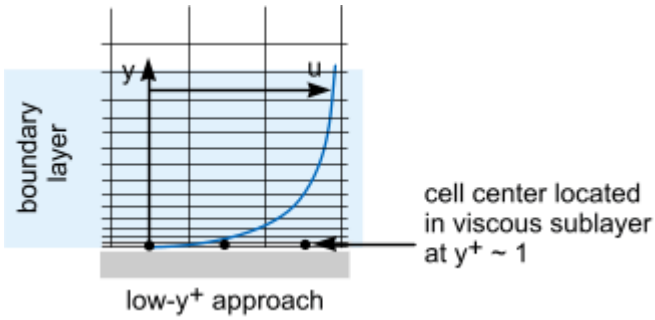
The non-dimensional wall distance y^+, see Eqn. (410), can be used to define the extents of the sublayers. The following plot shows the non-dimensional velocity u^+, see Eqn. (411), as a function of y^+ across the three sublayers:



STAR-CCM+ provides three types of wall treatment:

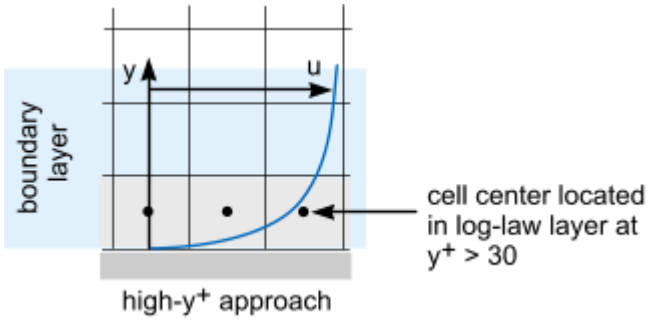
- The low-y^+ wall treatment resolves the viscous sublayer and needs little or no modeling to predict the flow across the wall boundary. The transport equations are solved all the way to the wall cell. The wall shear stress is computed as in laminar flows.

To resolve the viscous sublayer, these models require a sufficiently fine mesh with near-wall cells located at y^+ of around unity. The computational expense that is associated with this approach can be significant, particularly for large Reynolds number flows where the viscous sublayer can be very thin. Therefore this wall treatment is suitable only for low Reynolds number flows.



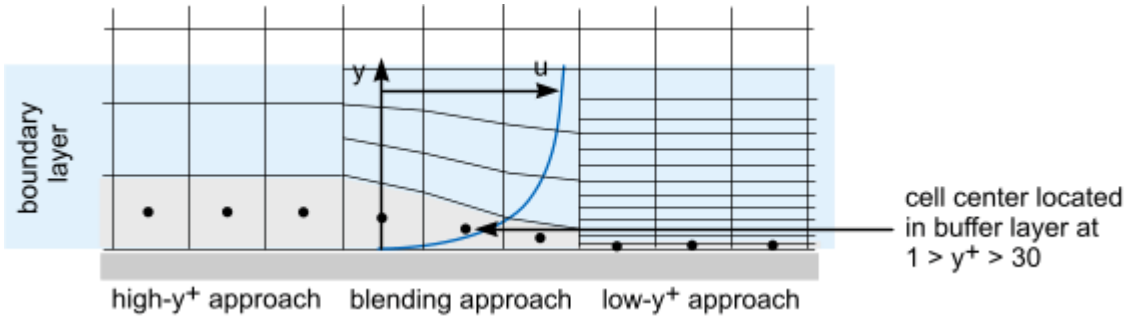
- The high-y^+ wall treatment does not resolve the viscous sublayer. Instead wall functions are used to obtain the boundary conditions for the continuum equations. Wall shear stress, turbulent production, and turbulent dissipation are derived from equilibrium turbulent boundary layer theory.

  This approach assumes that the near-wall cell lies within the log-law layer of the boundary layer at y^+ > 30. The main advantage of the high-y^+ wall treatment is therefore the significant savings in the number of near-wall cells.



- The all-y^+ wall treatment is a hybrid treatment that emulates the low-y^+ wall treatment for fine meshes, and the high-y^+ wall treatment for coarse meshes.

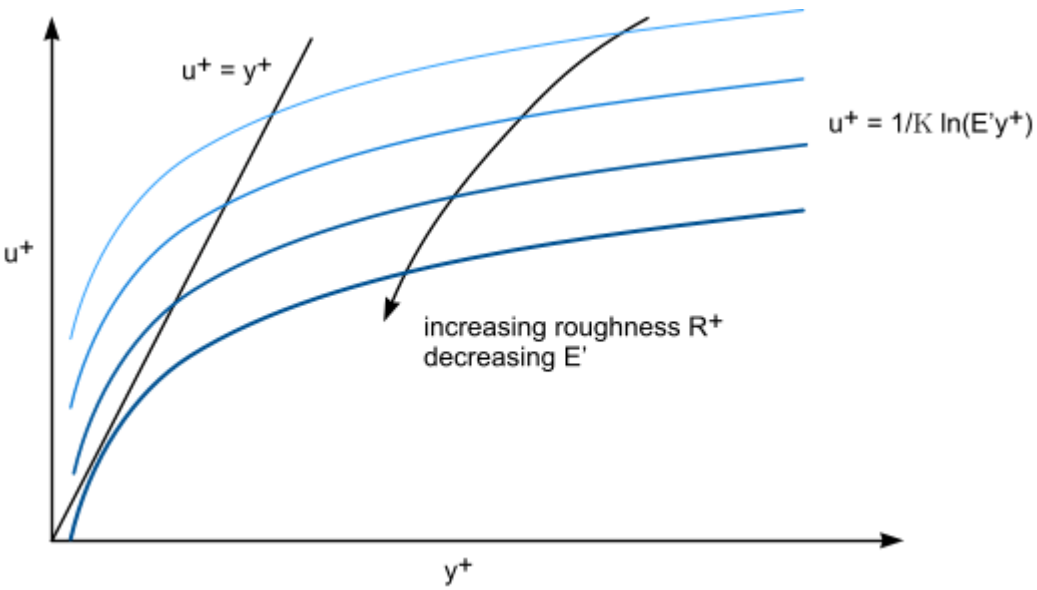
  It is also formulated with the desirable characteristic of producing reasonable answers for meshes of intermediate resolution, that is, when the wall-cell centroid falls within the buffer region of the boundary layer. A blending function is then used to calculate turbulence quantities such as dissipation, production, and stress tensor.



Wall Treatment for Rough Walls

Roughness effects are a critical issue in diverse engineering applications. Even small surface imperfections can lead to significant disturbances in the velocity field and change the flow behavior and thus the performance of a product. As roughness structures are in most cases too small for the mesh to resolve, wall models need to be employed.

In general, the effect of wall roughness is modeled by moving the logarithmic region of the boundary layer closer to the wall. Reducing the turbulent wall function coefficient E incorporates this effect (see Eqn. (417)):



As shown in the accompanying figure, for large values of roughness parameter R^+, the logarithmic profile can stop intersecting with the assumed linear profile in the sublayer. If this event occurs in the fully rough regime $R^+ > R^+_{\text{rough}}$, there is no problem, since the sublayer is irrelevant. If these curves do not intersect in the transitional roughness regime $R^+_{\text{smooth}} < R^+ < R^+_{\text{rough}}$, STAR-CCM+ uses the logarithmic profile, with the proviso that u^+ must never be less than zero.

By default, STAR-CCM+ limits the roughness height locally such that $R^+ = y^+$ when this condition is violated. You can remove this limit by deactivating the *Roughness Limiter* Wall Roughness Parameters expert property.

Wall Treatment for Reynolds-Averaged Navier-Stokes Turbulence Models

Not all wall treatments are available with every RANS model or model variant. Some high Reynolds number models do not contain the ability to attenuate the turbulence in the viscous-affected regions and therefore only include a high-y^+ wall treatment. Low Reynolds number models are offered only with a low-y^+ wall treatment and an all-y^+ wall treatment.

Each wall treatment is adapted to the underlying turbulence model, as assumptions specific to the models must be made for the turbulence wall boundary conditions.

The two-layer all-y^+ wall treatment for example, that is available for the two-layer turbulence models, uses an approach that is identical to the all-y^+ wall treatment, but contains a wall boundary condition for ϵ that is consistent with the two-layer formulation of the K-Epsilon and Reynolds Stress turbulence models.

The available wall treatments for the RANS turbulence models are listed in the Turbulence Models Lookup.

Wall Treatment for Scale-Resolving Simulations

Resolving the viscous sublayer for a Large Eddy Simulation is potentially expensive, since unlike RANS simulations, it is not sufficient to stretch the mesh only normal to the wall. The streaky structures that develop in the near-zero wall region also require adequate mesh definition in the spanwise and streamwise directions. Therefore, it may be desirable to avoid having to resolve the viscous sublayer. To facilitate this choice, a low-y^+ wall treatment and an all-y^+ wall treatment are available.

Detached Eddy Simulations circumvent the challenging near-wall mesh requirements of Large Eddy Simulations. They solve the boundary layer by using RANS closure models. The available wall treatments depend on the RANS model that is used for closure, see Turbulence Models Lookup.

Contents:

Deciding on a Wall Treatment
Setting up a Wall Treatment
Wall Treatment Models Reference
Wall Treatment Formulation
Wall Laws Formulation

Deciding on a Wall Treatment

To choose the appropriate wall treatment for your simulation take into account the following recommendations:

Wall Treatment	Choose for simulations where:
all-y^+	• the selected turbulence model allows this wall treatment. Choose this approach whenever available, especially when it is difficult to achieve a pure high-y^+ or low-y^+ mesh because of varying geometrical and velocity scales associated with the model.

Wall Treatment	Choose for simulations where:
high-y^+	• wall-bounded effects are not a priority. • separation is expected only due to sharp changes in the geometry. • wall roughness effects need to be included. • simulation time and thus cell count is a critical issue. • the selected turbulence model does not explicitly damp the turbulence in the near-wall region. Examples: Standard K-Epsilon model, Realizable K-Epsilon model.
low-y^+	• an accurate prediction of the boundary layer velocity and/or temperature profile is important, such as for pressure-drop or drag calculations. • the boundary layer tends to separate due to adverse pressure gradients. • simulation time and thus cell count is not a critical issue. • the selected turbulence model damps the turbulence in the near-wall region. Example: Standard K-Epsilon Low-Reynolds Number model.

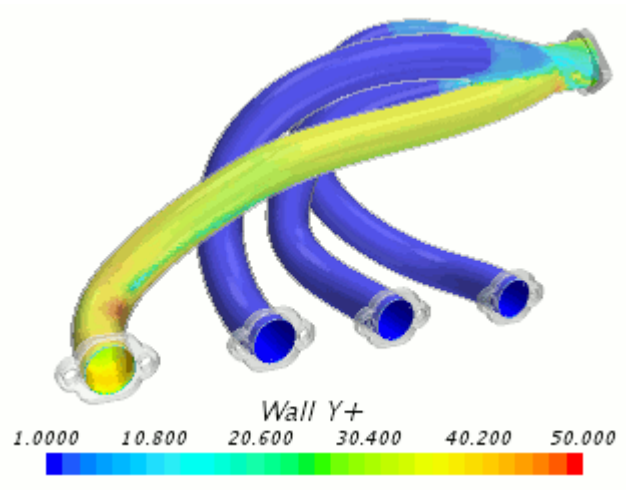
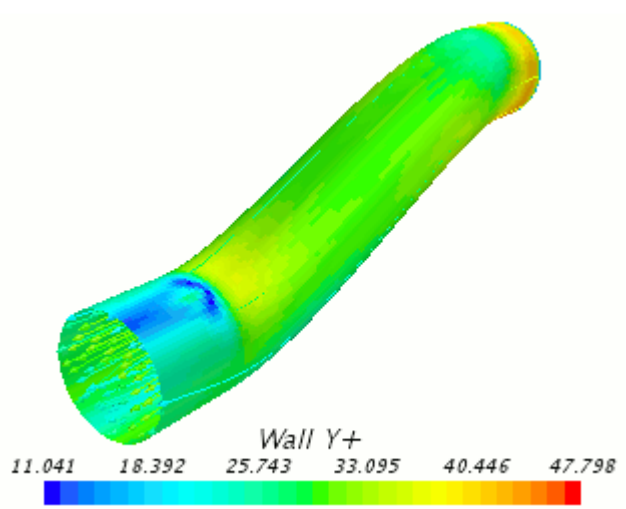
Setting up a Wall Treatment

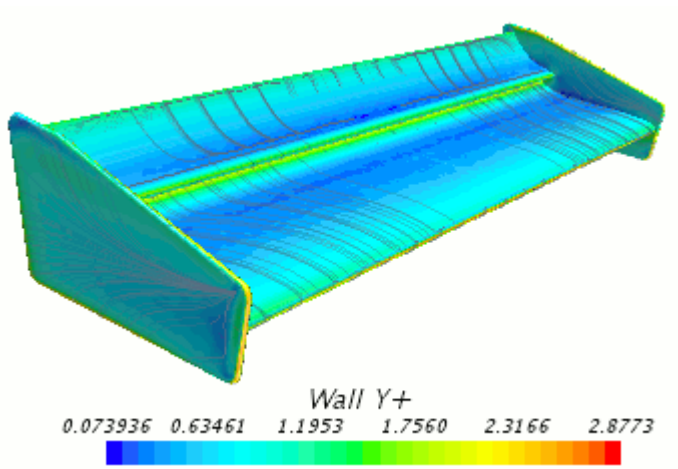
Each wall treatment requires a specific near-wall spacing of the mesh. If the near-wall mesh resolution is not consistent with the modeling assumptions, significant errors can result.

The non-dimensional wall distance y^+ determines whether the near-wall mesh resolution is appropriate for a specific wall treatment.

There is no automated way to create a mesh that fits the wall treatment of your choice. To set up a suitable combination of mesh and wall treatment:

1. Create an initial exploratory mesh.
2. Decide on a Wall Treatment.
3. Set up a physics continuum with a turbulent viscous regime and a turbulence model associated. From the *Wall Treatment* group box, select the Wall Treatment Model of your choice.
4. Run an initial exploratory simulation.
5. Create a scalar scene and plot the field function Wall Y+ on the wall boundaries that define the turbulent fluid domain.
6. Determine whether your mesh is appropriate for the selected wall treatment:

Wall Treatment	Appropriate Wall Y+ Values
all-y^+	**Wall Y+** < 1 or **Wall Y+** > 30 **Note:** For 1 < **Wall Y+** < 30, the blending approach creates reasonable solutions. Nevertheless, you are advised to keep the number of near-wall cells with centroids lying in the buffer layer low. Example: Exhaust Manifold 
high-y^+	**Wall Y+** > 30 **Note:** In some cases, a few cells have a small value of **Wall Y+**, especially near stagnation or separation. This situation is acceptable. Example: S-Bend 

low-y^+	**Wall Y+** < 1 **Note:** It is acceptable to have some boundary faces with higher **Wall Y+** values as long as the near-wall cells are still placed within the viscous sublayer, that is **Wall Y+** < 5. Example: Dual Element Wing 

7. If the **Wall Y+** values are not adequate for the wall treatment, refine or coarsen the mesh as required.

   The Wall Thickness distribution mode option of the Prism Layer Mesher allows you to directly control the thickness of the near-wall cell layer. The **Wall Y+** value is linearly dependent on the thickness of the wall adjacent cell layer. If you double the near-wall prism layer thickness, you approximately double the **Wall Y+** value.
8. Repeat steps 4 to 7 until the **Wall Y+** values fit the selected wall treatment.

Wall Treatment Models Reference

The **Wall Treatment** models allow you to set near-wall modeling assumptions for turbulence models.

Models Overview

Model Names and Abbreviations	**High y+ Wall Treatment**	**High y+**
	Low y+ Wall Treatment	**Low y+**
	All y+ Wall Treatment*	**All y+**
Theory	See: Spalart-Allmaras Turbulence: Evaluation of Reference Velocity u* K-Epsilon Turbulence: Wall Treatment K-Omega Turbulence: Wall Treatment Reynolds Stress Transport Turbulence: Wall Treatment and:	

	Wall Laws Formulation	
Provided By	**[physics continuum]** > **Models** > **Wall Treatment**	
Example Node Path	**Continua** > **Physics 1** > **Models** > **All y+ Wall Treatment**	
Requires	*Space*: **Axisymmetric**, **Two Dimensional**, or **Three Dimensional** *Material*: **Gas**, **Liquid**, **Eulerian Multiphase**, **Multi-Component Gas**, or **Multi-Component Liquid** *Viscous Regime*: **Turbulent** *Turbulence*: See Turbulence Models Lookup.	
Properties	See Properties Lookup.	
Activates	Boundary Inputs	See Boundary Settings.
	Field Functions	**Ustar** **Virtual Local Heat Transfer Coefficient** **Wall y+** See Field Functions.

Turbulence Models Lookup

			High y+	Low y+	All y+
Detached Eddy Simulation	**EB K-Epsilon Detached Eddy**			✓	✓
	SST (Menter) K-Omega Detached Eddy		✓	✓	✓
	Spalart-Allmaras Detached Eddy			✓	✓
Large Eddy Simulation	**Dynamic Smagorinsky Subgrid Scale**			✓	✓
	Smagorinsky Subgrid Scale			✓	✓
	WALE Subgrid Scale			✓	✓
Reynolds-Averaged Navier-Stokes	**K-Epsilon Turbulence**	**AKN K-Epsilon Low-Re**		✓	✓
		EB K-Epsilon		✓	✓
		Lag EB K-Epsilon		✓	✓
		Realizable K-Epsilon	✓		
		Realizable K-Epsilon Two-Layer			✓
		Standard K-Epsilon	✓		

			High y+	Low y+	All y+
		Standard K-Epsilon Low-Re		✓	✓
		Standard K-Epsilon Two-Layer			✓
		V2F K-Epsilon		✓	✓
	K-Omega Turbulence	**SST (Menter) K-Omega**	✓	✓	✓
		Standard (Wilcox) K-Omega	✓	✓	✓
	Reynolds Stress Turbulence	**Elliptic Blending**			✓
		Linear Pressure Strain	✓		
		Linear Pressure Strain Two-Layer			✓
		Quadratic Pressure Strain	✓		
	Spalart-Allmaras Turbulence	**High-Reynolds Number Spalart-Allmaras**	✓		
		Standard Spalart-Allmaras		✓	✓

Properties Lookup

	High y+	Low y+	All y+
Iterative Ustar When **On**, this property allows iterative computation of the reference velocity u^* from the wall law. When this property is **Off**, the reference velocity u^* is computed according to the specific turbulence model.			✓

Boundary Settings

Note: Boundary types that do not require setting any conditions or values are not listed.

Domain Boundaries

The following boundary conditions and values are equal to all boundaries of type:

- Wall
- Baffle Boundary
- Porous Baffle Boundary
- Contact Interface Boundary

- Mapped Contact Boundary

Blended Wall Function (All y+)

Added to the **Values** node on no-slip walls when an all-y^+ wall treatment is selected. You need to specify two scalar values:

- *E*: The log law offset.
- *Kappa*: The von Karman constant.

Standard Wall Function (High Y+)

Added to the **Values** node on no-slip walls when the high-y^+ wall treatment is selected. The values you need to specify are equal to the **Blended Wall Function**.

Wall Surface Specification (All Y+, High Y+)

Controls whether the boundary is smooth or rough.

Added to the **Conditions** node on no-slip walls when an all-y^+ or the high-y^+ wall treatment is selected.

Method	Corresponding Value Nodes
Smooth Regular wall treatment without modification for wall roughness.	None.
Rough Modification of wall treatment to incorporate roughness. Calculation of the roughness function f from a specified equivalent sand-grain roughness height r and specified wall roughness parameters using Eqn. (431) and Eqn. (432). f modifies the wall function coefficient E using Eqn. (417).	**Roughness Height** Scalar profile value to specify r. **Wall Roughness Parameters** Scalar values to specify: • *B*: Coefficient B in roughness function. • *C*: Coefficient C in roughness function. • *RplusSmooth*: Roughness parameter R^+_{smooth} in roughness function corresponding to a fully smooth surface. • *RplusRough*: Roughness parameter R^+_{rough} in roughness function corresponding to a fully rough surface. The *Roughness Limiter* controls whether to limit the roughness height locally, such that $R^+ = y^+$ when y^+ becomes less than R^+. This expert property is activated by default.

Field Functions

Ustar

Scalar field that represents the reference velocity $u*$, computed according to the selected wall treatment.

Virtual Local Heat Transfer Coefficient

Scalar field that represents an approximation to the true Local Heat Transfer Coefficient. It does not require an energy model to be activated and is defined as $h = \rho C_p u^* / T^+$ where T^+ is computed according to the selected wall treatment. Since thermal material properties are not available, the specific heat C_p and the molecular and turbulent Prandtl numbers are specified as reference values. The fluid density ρ is obtained locally from the simulation.

Wall Y+

Scalar field that represents the non-dimensional wall distance. It is defined as $y^+ = u^* y / \nu$ where u^* is the reference velocity, y is the normal distance from the centroid to the wall in wall-adjacent cells and ν is the kinematic viscosity.

Wall Treatment Formulation

This section contains the wall treatment formulation for the RANS turbulence models, Large Eddy Simulation (LES), and Detached Eddy Simulation (DES).

Contents:

Spalart Allmaras Wall Treatment
K-Epsilon Wall Treatment
Elliptic Blending Wall Treatment
K-Omega Wall Treatment
Reynolds Stress Transport Wall Treatment

Spalart-Allmaras Wall Treatment

Evaluation of Reference Velocity u*

Spalart and Allmaras devised the model such that $u^* = \tilde{\nu} / \left(\kappa \tilde{d}\right)$ would remain constant throughout the viscous sublayer as well as in the log region. However, they concede that "ν_t does not follow its ideal linear log-law behavior $\kappa y u_\tau$ far up..." This means that the assumption of a linear relationship causes inaccuracies in u^* at large y^+ values. In STAR-CCM+, a proprietary quadratic relation is used that yields more realistic values of u^* for coarse meshes.

K-Epsilon Wall Treatment

In the K-Epsilon models, the wall treatment performs three important functions:

- Specifies the reference velocity u^* to be used in the wall laws
- Computes a special value of turbulent production G_k in the wall cell
- Computes a special value of turbulent dissipation ε in the wall cell

The following equations present the wall-treatment formulations. In general, the high-y^+ formulation is equivalent to the traditional wall-function approach, and the low-y^+ formulation is equivalent to the traditional

low Reynolds number approach where no modeling beyond the assumption of laminar flow is necessary in the wall cells.

For the all-y^+ formulation, a blending function g is defined in terms of the wall-distance Reynolds number:

$$g = \exp\left(-\frac{Re_y}{11}\right) \tag{391}$$

and $Re_y = \sqrt{k}y/\nu$, a wall-distance-based Reynolds number.

Reference Velocity

$$u^* = \begin{cases} \sqrt{C_\mu^{1/2}k} & ;\text{high-}y^+ \\ \sqrt{g\nu u/y + (1-g)C_\mu{}^{1/2}k} & ;\text{low-}y^+ \\ \sqrt{g\nu u/y + (1-g)C_\mu{}^{1/2}k} & ;\text{all-}y^+ \end{cases} \tag{392}$$

The reference velocity u^* is not needed in the calculation for low-y^+ simulations (using a low Reynolds number model). However, it is still needed when computing the wall $y+$ for post-processing. Since the assumption is that $u^* \equiv \sqrt{\tau_w/\rho} \approx \sqrt{\nu u/y}$ gives an inaccurate result for coarse meshes, the hybrid approach is used for the low-y^+ wall treatment as well.

Wall-Cell Production

$$G_k = \tag{393}$$

$$\begin{cases} \frac{1}{\mu}\left(\rho u^* \frac{u}{u^+}\right)^2 \frac{\partial u^+}{\partial y^+} & ;\text{high-}y^+ \\ \mu_t S^2 & ;\text{low-}y^+ \\ g\mu_t S^2 + (1-g)\frac{1}{\mu}\left(\rho u^* \frac{u}{u^+}\right)^2 \frac{\partial u^+}{\partial y^+} & ;\text{all-}y^+ \end{cases}$$

Wall-Cell Dissipation

$$\varepsilon = \begin{cases} \frac{u^{*3}}{\kappa y} & ;\text{high-}y^+ \\ \frac{2\nu k}{y^2} & ;\text{low-}y^+ \\ \frac{u^{*3}}{\kappa y} & ;\text{all-}y^+ \\ g\frac{2\nu k}{y^2} + \frac{k^{3/2}}{l_\varepsilon} & ;\text{two-layer all-y+} \end{cases} \tag{394}$$

Elliptic Blending Wall Treatment

For cases of low-y^+ mesh but high wall roughness, STAR-CCM+ provides the elliptic blending wall treatment. These equations are used with the Elliptic Blending K-Epsilon model.

Velocity Conditions

The elliptic blending wall treatment uses a method that is called "displacement of origin." (See [343] and [344].) Under fully rough conditions, the log-layer velocity profile is of the form:

$$u^+ = \frac{1}{\kappa}\log\left(\frac{y^+ + y_0^+}{y_0^+}\right) \tag{395}$$

where y_0^+ is:

$$y_0^+ = \begin{cases} 0.56\left(r^+/20\right)^{2.5}; & r^+ < 20 \\ 0.63\ \zeta\ \left(r^+\right) + 0.028\ r^+\ ; & 20 \le r^+ \le 90 \\ 0.031\ r^+; & r^+ > 90 \end{cases}$$

where:

$$\zeta\left(r^+\right) = \sin\left[\pi\left(\frac{r^+ - 20}{70}\right)^{0.9}\right]$$

Displacement of origin can be combined with the blended wall treatment, in a modification of Reichardt's law:

$$u^+ = \frac{1}{\kappa}\log\left(1 + \kappa y^+\right) + F_r\left[1 - \exp\left(-\frac{y^+}{y_m^+}\right) - \frac{y^+}{y_m^+}\exp\left(-Dy^+\right)\right] \tag{396}$$

where:

- $F_r = \frac{1}{\kappa}\ln\left(\frac{1}{\kappa y_0^+}\right)$
- y_m^+ is the point at which the viscous sublayer meets the log-layer solution, which is determined by Newtonian iteration.
- $D = \frac{1}{2}\left(\kappa\frac{y_m^+}{F} + \frac{1}{y_m^+}\right)$

Temperature Conditions

The wall-treatment for temperature also uses displacement of origin.

$$T^+ = \exp\left(-\Gamma'\right)\sigma y^+ + \exp\left(-\frac{1}{\Gamma'}\right)\sigma_t\left[\frac{1}{\kappa}\ln\left(\frac{y^+ + y_0^+}{y_0^+}\right) + \mathcal{P}\right] \tag{397}$$

where:

- Γ' is a blending function. $\Gamma' = \frac{0.01\left[\sigma\left(y^+ + y_0^+\right)\right]^4}{1 + 5\,\sigma^3\left(y^+ + y_0^+\right)}$
- $\mathcal{P}$ governs the velocity at which the logarithmic and viscous regions of the thermal profiles intersect. $\mathcal{P} = 9.24\left[\left(\frac{\sigma}{\sigma_t}\right)^{3/4} - 1\right]\left[1 + 0.28\exp\left(-0.007\frac{\sigma}{\sigma_t}\right)\right]$

Turbulence Conditions

Near-wall dissipation ε is a function of roughness height:

$$\varepsilon = \Gamma\frac{\nu k}{y^2}Y_e + \left(1 - \Gamma\right)\frac{u^*}{\kappa y} \tag{398}$$

where:

$$Y_e = \frac{y^2}{\left(y/2.236 + y_0^{+3}\nu/u^*\right)^2}$$

K-Omega Wall Treatment

The equations in this section present the wall treatment formulations.

In the K-Omega models, the wall treatment performs three important functions:

- Specifies the reference velocity u^* to be used in the wall laws
- Computes a special value of turbulent production G_k in the wall cell
- Computes a special value of specific dissipation ω in the wall cell

In general, the high-y^+ formulation is equivalent to the traditional wall-function approach, and the low-y^+ formulation is equivalent to the traditional low Reynolds number approach where no modeling beyond the assumption of laminar flow is necessary in the wall cells.

For the all-y^+ formulation, a blending function g is defined in terms of the wall-distance Reynolds number:

$$g = \exp\left(-\frac{Re_y}{11}\right) \tag{399}$$

and $Re_y = \sqrt{k}y/\nu$, a wall-distance-based Reynolds number.

Reference Velocity

$$u^* = \begin{cases} \sqrt{\beta^{*1/2}k} & ;\text{high-}y^+ \\ \sqrt{g\nu u/y + \left(1-g\right)\beta^{*1/2}k} & ;\text{low-}y^+ \\ \sqrt{g\nu u/y + \left(1-g\right)\beta^{*1/2}k} & ;\text{all-}y^+ \end{cases} \tag{400}$$

The reference velocity u^* is not needed in the calculation for low-y^+ simulations. However, it is still needed when computing the wall $y+$ for post-processing. Since the assumption is that $u^* \equiv \sqrt{\tau_w/\rho} \approx \sqrt{\nu u/y}$ gives an inaccurate result for coarse meshes, the blended expression is used for the low-y^+ wall treatment as well.

Wall-Cell Production

$$G_k = \begin{cases} \frac{1}{\mu}\left(\rho u^* \frac{u}{u^+}\right)^2 \frac{\partial u^+}{\partial y^+} & ;\text{high-}y^+ \\ \mu_t S^2 & ;\text{low-}y^+ \\ g\mu_t S^2 + (1-g)\frac{1}{\mu}\left(\rho u^* \frac{u}{u^+}\right)^2 \frac{\partial u^+}{\partial y^+} & ;\text{all-}y^+ \end{cases} \tag{401}$$

Wall-Cell Specific Dissipation

$$\omega = \begin{cases} \frac{u^*}{\sqrt{\beta^*}\kappa y} & ;\text{high-}y^+ \\ \frac{6\nu}{\beta y^2} & ;\text{low-}y^+ \\ g\frac{6\nu}{\beta y^2} + \left(1+g\right)\frac{u^*}{\sqrt{\beta^*}\kappa y} & ;\text{all-}y^+ \end{cases} \tag{402}$$

Roughness Effects

Following Wilcox [339], the K-Omega model modifies specific dissipation ω as a function of wall roughness in the cases of low Reynolds number and the all y+ wall treatment.

Omega for Low Reynolds Numbers

In a low-Re treatment, the K-Omega model modifies specific dissipation ω in wall cells to be:

$$\omega = \frac{u_*^2}{\nu} S_r \tag{403}$$

with

$$S_r = \begin{cases} \left(\frac{50}{R^+}\right)^2 & \text{if } 5 \le R^+ \le 25 \\ \frac{100}{R^+} & \text{if } R^+ \ge 25 \end{cases} \tag{404}$$

where u_* is the friction velocity, ν is kinematic viscosity, and R+ is the roughness parameter.

Omega for the All y+ Wall Treatment

In the all y+ wall treatment, the K-Omega model modifies specific dissipation ω in wall cells to be:

$$\omega = g\frac{u_*^2}{\nu} S_r + (1-g)\frac{u_*}{\kappa y\sqrt{C_\mu}} \tag{405}$$

where g is the blending function described in Eqn. (399), with the additional requirement that g = 0 if $R^+ > R^+_{\text{rough}}$ where R^+_{rough} is the roughness parameter for a fully rough surface (i.e., a surface where roughness effects are so large, they leave no visible viscous flow behavior near the wall). In addition, u_* is the friction velocity, ν is kinematic viscosity, Sr is the function described above in Eqn. (404), κ is the von Karman constant, and C_μ (= 0.09) is a model coefficient.

Reynolds Stress Transport Wall Treatment

The following equations present the wall-treatment formulations.

For the Reynolds Stress Transport models, the wall treatment performs four important functions:

- Specifies the reference velocity u^* to be used in the wall laws
- Computes a special value of turbulent production G_k in the wall cell
- Computes a special value of turbulent dissipation ε in the wall cell
- Sets the specific Reynolds stress tensor $\mathbf{R}$ in the wall cell

In general, the high-y^+ formulation is equivalent to the traditional wall-function approach, and the all-y^+ and two-layer all-y^+ formulation.

For the all-y^+ formulation, a blending function g is defined in terms of the wall-distance Reynolds number:

$$g = \exp\left(-\frac{Re_y}{11}\right) \tag{406}$$

and $Re_y = \sqrt{k}y/\nu$, a wall-distance-based Reynolds number.

Reference Velocity

$$u^* = \begin{cases} \sqrt{C_\mu^{1/2}k} & ; \quad \text{high-}y^+ \\ \sqrt{g\nu u/y + (1-g)C_\mu{}^{1/2}k} & ; \quad \text{two-layer all-}y^+ \\ \sqrt{g\nu u/y + (1-g)C_\mu{}^{1/2}k} & ; \quad \text{all-}y^+ \end{cases} \tag{407}$$

Wall-Cell Normal Gradient

$$\frac{G_k}{\tau_w} = \begin{cases} \dfrac{u}{u^+\kappa y} & ; \quad \text{high-}y^+ \\ g\dfrac{u}{y} + \left(1-g\right)\dfrac{u}{u^+\kappa y} & ; \quad \text{two-layer all-}y^+ \\ 0.95\dfrac{u^2}{u^{+2}\nu}\dfrac{\nu_t^+}{\left(1+\nu_t^+\right)^2} & ; \quad \text{all-}y^+ \end{cases} \tag{408}$$

where $\nu_t^+ = \kappa y^+\left[1-\exp\left(\frac{-y^+}{26}\right)\right]^2$.

Wall-Cell Dissipation

$$\varepsilon = \begin{cases} \dfrac{u^{*3}}{\kappa y} & ; \quad \text{high-}y^+ \\ \dfrac{k^{3/2}}{l_\varepsilon} & ; \quad \text{two-layer all-}y^+ \\ g\dfrac{2\nu k}{y^2} + \left(1-g\right)\dfrac{u^{*3}}{\kappa y} & ; \quad \text{all-}y^+ \end{cases} \tag{409}$$

Wall Laws Formulation

This section presents details of the wall laws that are used in STAR-CCM+.

A wall law is a mathematical description of mean flow quantities, such as velocity, temperature and species concentration, in turbulent boundary layers. These formulations include non-dimensional quantities, asymptotic limits, and standard and blended wall laws. The most basic wall laws apply only to equilibrium situations, although there have been some attempts to extend wall laws to include non-equilibrium effects.

Two types of wall laws are used in STAR-CCM+:

- standard wall laws, which are slope-discontinuous between the laminar and turbulent profiles;
- blended wall laws, which include a buffer region that smoothly blends the laminar and turbulent profiles together.

You cannot select wall laws; they are chosen automatically based on the behavior of the turbulence model.

Turbulence models that contain damping functions (or behave as if they do) typically use blended wall laws in their all-y^+ wall treatment. The standard wall law is used with high-y^+ wall treatments.

Contents:

Non-Dimensional Quantities
Asymptotic Limits
Standard Wall Laws
Blended Wall Laws
Roughness Function
Wall Laws Nomenclature
Wall Laws Bibliography

Non-Dimensional Quantities

The following non-dimensional quantities are used:

$$y^+ = \frac{yu^*}{\nu} \tag{410}$$

$$u^+ = \frac{u}{u^*} \tag{411}$$

$$T^+ = \left(T - T_w\right)\frac{\rho\, C_p u^*}{\dot{q}''} \tag{412}$$

In these equations, the following nomenclature applies:

y is the normal distance from the wall to the wall-cell centroid.

u is the component of wall-cell velocity parallel to the wall.

T is the wall-cell temperature.

T_w is the wall-boundary temperature.

$\dot{q}''$ is the wall heat flux.

ν is the kinematic viscosity.

ρ is the density.

C_p is the specific heat.

u^* is a reference velocity.

Although the reference velocity is often related to the wall shear stress ($u^* = \sqrt{\tau_w/\rho}$), in actual practice the reference velocity is derived from a turbulence quantity specific to the particular turbulence model. For blended and two-layer models, the reference velocity u^* can be computed iteratively from the wall law.

The wall laws are set up to provide u^+ and T^+ (and their first derivatives) as a function of y^+ and other relevant quantities, such as molecular and turbulent Prandtl numbers.

Asymptotic Limits

The wall laws differ only in their treatment in the buffer region; the viscous-sublayer and log-layer behaviors are identical.

Viscous Sublayer

The velocity distribution is modeled as:

$$u^+_{\text{lam}} = y^+ \tag{413}$$

The temperature distribution (in the absence of viscous dissipation) is modeled as:

$$t^+_{\text{lam}} = \sigma u^+_{\text{lam}} \tag{414}$$

The non-dimensional quantity q^+_{lam} introduces the effect of viscous dissipation:

$$q^+_{\text{lam}} = \frac{\rho u^* u^2}{\dot{2}q''} \tag{415}$$

Logarithmic Layer

The velocity distribution is modeled as:

$$u^+_{\text{trb}} = \frac{1}{\kappa}\ln\left(E' y^+\right) \tag{416}$$

where:

$$E' = \frac{E}{f} \tag{417}$$

and the default values of the coefficients are κ = 0.42 and E = 9.0. The value of the Roughness Function f in Eqn. (417) is unity for smooth walls, and for rough walls is computed according to Eqn. (432).

The temperature distribution (in the absence of viscous dissipation) is modeled as:

$$t^+_{\text{trb}} = \sigma_t\left(u^+_{\text{trb}} + P\right) \tag{418}$$

where the function P, which governs the velocity at which the logarithmic and viscous regions of the thermal profiles intersect, is by Jayatilleke [35]:

$$P = 9.24\left[\left(\frac{\sigma}{\sigma_t}\right)^{3/4} - 1\right]\left[1 + 0.28\exp\left(\frac{-0.007\sigma}{\sigma_t}\right)\right] \tag{419}$$

and σ is the molecular Prandtl number, and σ_t is the turbulent Prandtl number.

The non-dimensional quantity q^+_{trb} introduces the effect of viscous dissipation:

$$q^+_{\mathrm{trb}} = \frac{\rho u^*}{\dot{2}q''}\left[u^2 - \left(\frac{\sigma}{\sigma_t} - 1\right)\left(u^+_c u^*\right)^2\right] \tag{420}$$

where u^+_c is a fictitious non-dimensional velocity that would occur at the intersection of the laminar and turbulent temperature profiles. It is computed from Eqn. (419) as:

$$u^+_c = \frac{P}{\sigma - \sigma_t} \tag{421}$$

Standard Wall Laws

The standard wall laws have slope discontinuities between the laminar and logarithmic regions.

They are given by:

$$u^+ = \begin{cases} u^+_{\mathrm{lam}} & \text{for } y^+ \le y^+_m \\ u^+_{\mathrm{trb}} & \text{for } y^+ > y^+_m \end{cases} \tag{422}$$

$$T^+ = \begin{cases} \sigma\left(u^+_{\mathrm{lam}} - q^+_{\mathrm{lam}}\right) & \text{for } y^+ \le y^+_t \\ \sigma_t\left(u^+_{\mathrm{trb}} + P + q^+_{\mathrm{trb}}\right) & \text{for } y^+ > y^+_t \end{cases} \tag{423}$$

For the momentum wall law, a Newton iteration determines the intersection of the viscous and fully turbulent regions y^+_m. A similar iterative process determines y^+_t, the intersection of the viscous and fully turbulent regions for the thermal wall law.

Blended Wall Laws

The blended wall laws are intended to represent the buffer layer by appropriately blending the viscous sublayer and logarithmic regions.

For momentum, Reichardt's law [38] is used:

$$u^+ = \frac{1}{\kappa}\ln\left(1 + \kappa y^+\right) + C\left[1 - \exp\left(-\frac{y^+}{D}\right) - \frac{y^+}{D}\exp\left(-by^+\right)\right] \tag{424}$$

where:

$$D = y^+_m \tag{425}$$

$$C = \frac{1}{\kappa}\ln\left(\frac{E'}{\kappa}\right) \tag{426}$$

$$b = \frac{1}{2}\left(\frac{D\kappa}{C} + \frac{1}{D}\right) \tag{427}$$

Reichardt's law is also used to compute the near-wall reference velocity u^* if the appropriate option is selected; see Eqn. (410). For temperature, Kader's law [36] modified slightly for the effects of wall roughness is used:

$$T^+ = \exp(-\Gamma)\left(T^+_{\text{lam}} - q^+_{\text{lam}}\right) + \exp\left(-\frac{1}{\Gamma}\right)\left(T^+_{\text{trb}} - q^+_{\text{trb}}\right) \tag{428}$$

where:

$$\Gamma = \frac{0.01c\left(\sigma y^+\right)^4}{1 + \frac{5}{c}\sigma^3 y^+} \tag{429}$$

and

$$c = e^{\left(\frac{1}{f} - 1\right)} \tag{430}$$

where f is the Roughness Function.

Roughness Function

The roughness function f that modifies the log-law coefficient E is a function of the roughness parameter.

This parameter is expressed as follows:

$$R^+ = \frac{ru^*}{\nu} \tag{431}$$

where r is the equivalent sand-grain roughness height, that you specify. You obtain the value of the equivalent sand-grain roughness height that is appropriate for your model either from the literature or empirically. To determine r from an experiment:

- Measure the friction factor for the Reynolds number of your case in the experiment.
- Plot your measurements on a Moody diagram.
- Deduce the equivalent sand-grain roughness height from the Moody diagram curve.

The function that is used is a slightly expanded version of the expression that was given in [34]:

$$f = \begin{cases} 1 & \text{for } R^+ \le R^+_{\text{smooth}} \\ \left[B\left(\dfrac{R^+ - R^+_{\text{smooth}}}{R^+_{\text{rough}} - R^+_{\text{smooth}}}\right) + CR^+\right]^a & \text{for } R^+_{\text{smooth}} < R^+ < R^+_{\text{rough}} \\ B + CR^+ & \text{for } R^+ > R^+_{\text{rough}} \end{cases} \tag{432}$$

where the exponent a is expressed as:

$$a = \sin\left[\frac{\pi}{2}\ \frac{\log\left(R^{+}/R^{+}_{\text{smooth}}\right)}{\log\left(R^{+}_{\text{rough}}/R^{+}_{\text{smooth}}\right)}\right] \tag{433}$$

and the default values of the coefficients are from [34]:

$B = 0 \quad C = 0.253 \quad R^{+}_{\text{smooth}} = 2.25 \quad R^{+}_{\text{rough}} = 90$

The coefficient B is not required for most correlations, but as the original formulation includes it, STAR-CCM+ provides it for completeness.

Wall Laws Nomenclature

The following is a guide to the terms used in this section.

Scalars

E	wall function coefficient
f	roughness function
u	component of wall-cell velocity parallel to the wall
u^*	reference velocity that is used in wall functions
u^+	wall-parallel velocity non-dimensionalized with u^*
R^+	roughness parameter
y^+	non-dimensional wall coordinate
y	normal distance from the wall to the wall-cell centroid
P	function of Prandtl numbers
q^+	dimensionless viscous dissipation
T^+	dimensionless temperature
T	wall-cell temperature
T_w	wall-boundary temperature
$\dot{q}''$	wall heat flux magnitude
C_p	specific heat
κ	von Karman constant
ρ	density
μ	dynamic viscosity
μ_t	turbulent viscosity
σ	Prandtl number
σ_t	turbulent Prandtl number

τ	wall shear stress magnitude
Γ	blending function in Kader's wall law

Subscripts

lam	laminar
turb	turbulent

Wall Laws Bibliography

[34] *Cebeci, T. and Bradshaw, P. 1977. Momentum Transfer in Boundary Layers, Hemisphere Publishing/McGraw-Hill, pp. 176-180.*

[35] *Jayatilleke, C.L. 1969. "The influence of Prandtl number and surface roughness on the resistance of the laminar sub-layer to momentum and heat transfer", Progress in Heat and Mass Transfer, 1, pp. 193-330.*

[36] *Kader, B.A. 1981. "Temperature and Concentration Profiles in Fully Turbulent Boundary Layers", Int. J. Heat Mass Transfer, 24, pp. 1541-1544.*

[37] *Launder, B.E. and Spalding, D.B. 1974. "The Numerical Computation of Turbulent Flows", Computer Methods in Applied Mechanics and Engineering, 3, pp. 269-289.*

[38] *Reichardt, H. 1951. "Vollstaendige Darstellung der turbulenten Geschwindigkeitsverteilung in glatten Leitungen", Z. Angew. Math. Mech, 31(7), pp. 208-219.*

[39] *Spalding, D. B. 1994. 'Calculation of turbulent heat transfer in cluttered spaces', Proc. 10th Int. Heat Transfer Conference, Brighton, UK.*

Modeling Transition

The term “transition” refers to the phenomenon of laminar to turbulence transition in boundary layers. A transition model in combination with a turbulence model predicts the onset of transition in a turbulent boundary layer.

Three primary modes of transition are typically involved:

1. Natural transition, in which a laminar boundary layer subjected to weak disturbances becomes linearly unstable beyond a critical Reynolds number at which Tollmien-Schlichting waves start to grow.
2. Bypass transition, the process of transition in response to large disturbances outside the boundary layer, typically free stream turbulence levels in excess of 1%.
3. Separation induced transition, in which separation of the laminar boundary layer induces transition. The laminar boundary layer often reattaches in response to the enhanced mixing caused by the turbulent flow, forming laminar separation bubble upstream of the transition location.

Transition models are necessary because turbulence models themselves cannot be relied upon to predict the onset of transition.

In STAR-CCM+, two approaches are currently available to account for transition:

1. Turbulence Suppression model—this approach mimics the effect of transition by suppressing the turbulence in a certain pre-defined region.
2. Gamma ReTheta Transition model and Gamma Transition model—these models are based on correlations and solve additional transport equations that are coupled with the SST K-Omega model.

The Turbulence Suppression model is a zero-equation model and is thus the fastest and least expensive one, but it requires that you already know the location of transition.

The Gamma ReTheta Transition model and the Gamma Transition model are more expensive, but provide a true predictive capability. The Gamma ReTheta Transition model solves for two additional transport equations, whereas the Gamma Transition model only solves for one equation—it is therefore faster and less expensive than the Gamma ReTheta Transition model. Both models offer an additional correlation for stationary cross-flow instabilities.

Contents:

Turbulence Suppression Model
Gamma ReTheta Transition Model
Gamma Transition Model
Transition Guidelines
Transition Formulation

Turbulence Suppression Model

The Turbulence Suppression model is a zero-equation model that allows the user to mimic the effect of transition by suppressing the turbulence in a certain pre-defined region. Since it does not have a predictive capability, it is arguably not a true transition model. However, if the location of transition is known a priori (for instance due to a boundary layer trip) this approach is computationally efficient.

Turbulence Suppression

Turbulence suppression is achieved in a region as follows:

1. Setting the turbulent viscosity and/or the Reynolds stresses to zero
2. Setting the production terms in the turbulence transport equations to zero
3. Not using wall functions to compute wall shear stress or heat flux, but using the laminar flow methods instead

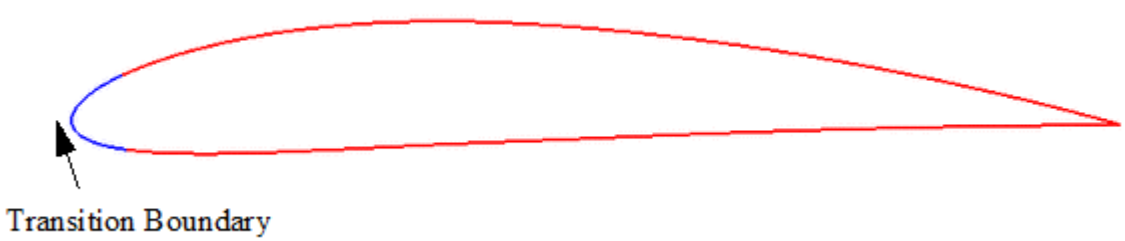
Turbulence Suppression Region

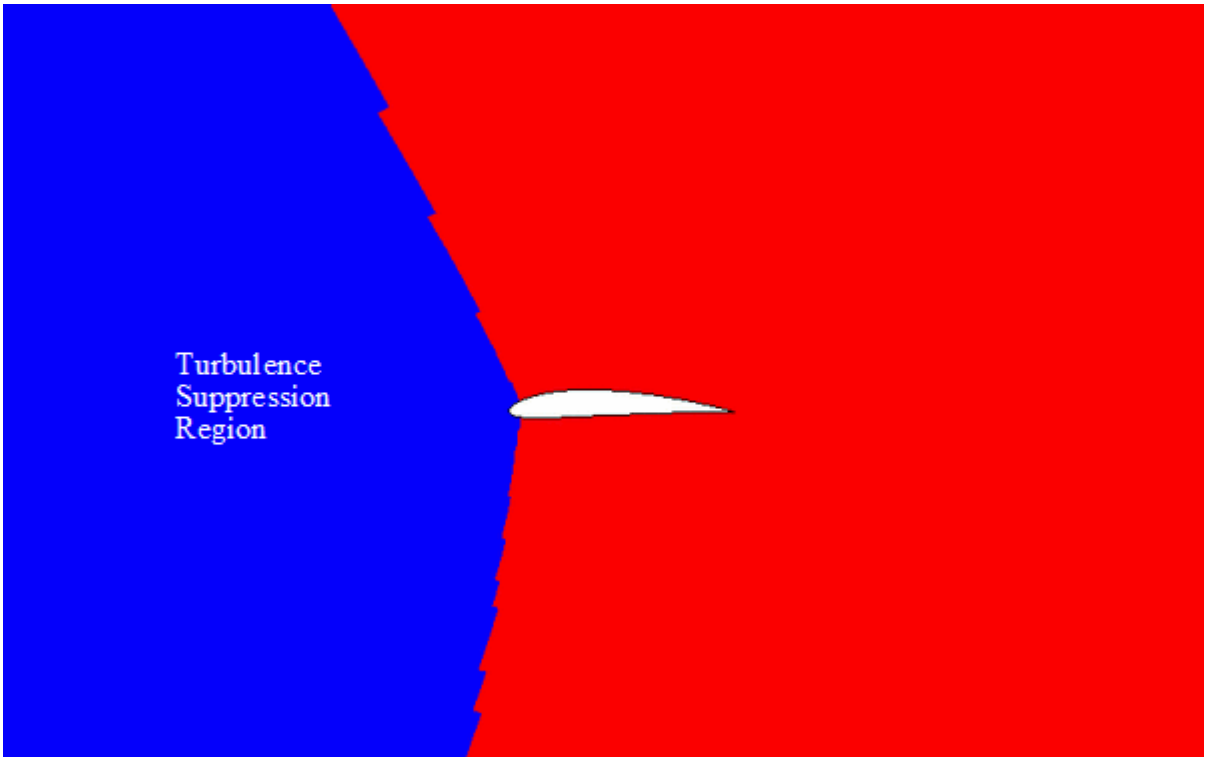
One possible approach to define a region in space in which turbulence is suppressed would be to identify an entire region in the volume mesh. However, for certain grid topologies it might impractical to create the appropriate region in the volume mesh.

Instead, the approach taken in STAR-CCM+ is to compute two distances for each field point in the interior:

- The wall distance, defined as the distance from a field point to the nearest point on a wall that is not a transition boundary.
- The transition boundary distance, defined as the distance from the field point to the nearest point on a transition boundary.

The region in space in which turbulence is suppressed is then defined as any point at which the transition boundary distance is less than the wall distance. The way in which a turbulence suppression region is created from a transition boundary is illustrated in the following figures.





Transition Boundary

A transition boundary is a wall boundary adjacent to which turbulence effects will be suppressed. A wall boundary in a turbulent flow may be turned into a transition boundary by first activating the Transition model in the continuum, and then selecting the turbulence suppression option for that wall. Note that the transition model will have no effect in a physics continuum unless the turbulence suppression option is selected on a wall boundary.

Transition Boundary Distance Calculation

Transition boundary distance is calculated in exactly the same manner as the wall distance, with the proviso that only walls that are transition boundaries are included in the calculation.

The transition boundary distance is computed during solution initialization, and updated only after the mesh or boundary conditions have changed.

Contents:

Selecting the Turbulence Suppression and Transition Boundary Distance Models
Running a Simulation with the Transition Model
Turbulence Suppression Model Reference
Transition Boundary Distance Model Reference

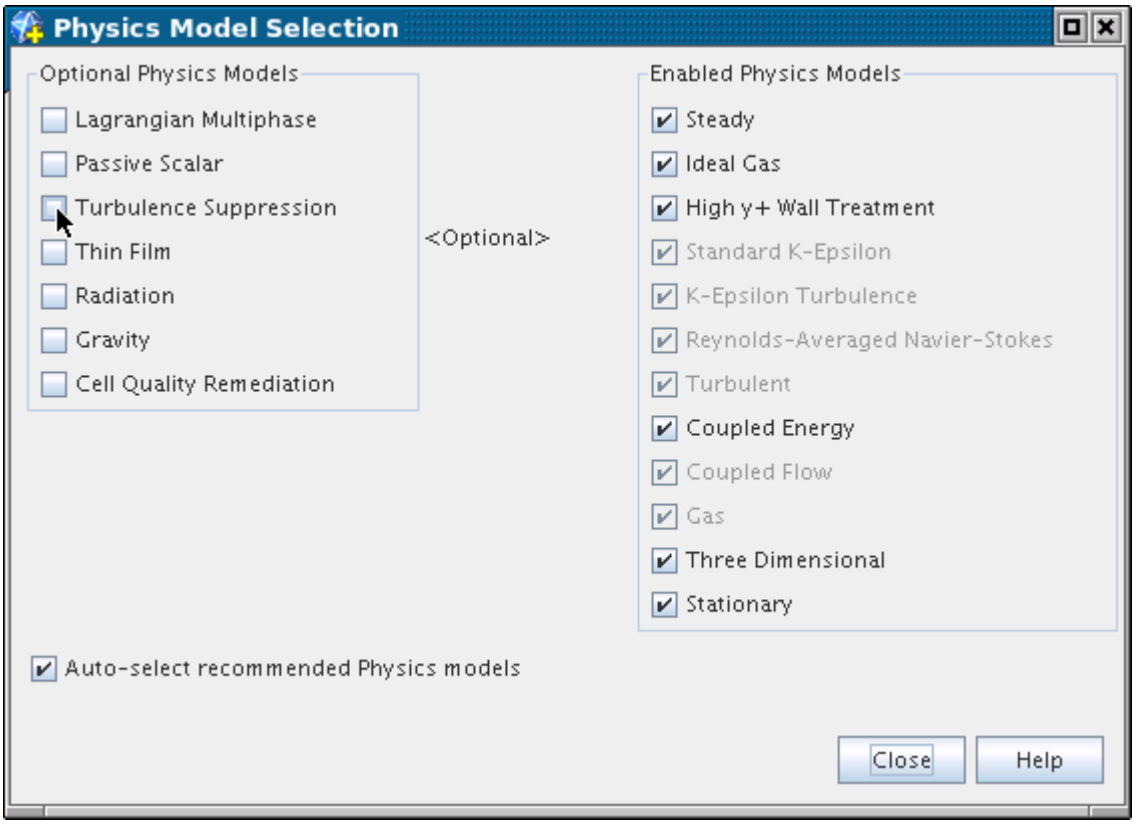
Selecting the Turbulence Suppression and Transition Boundary Distance Models

This section describes how to select Turbulence Suppression and Transition Boundary Distance models with or without auto-select.

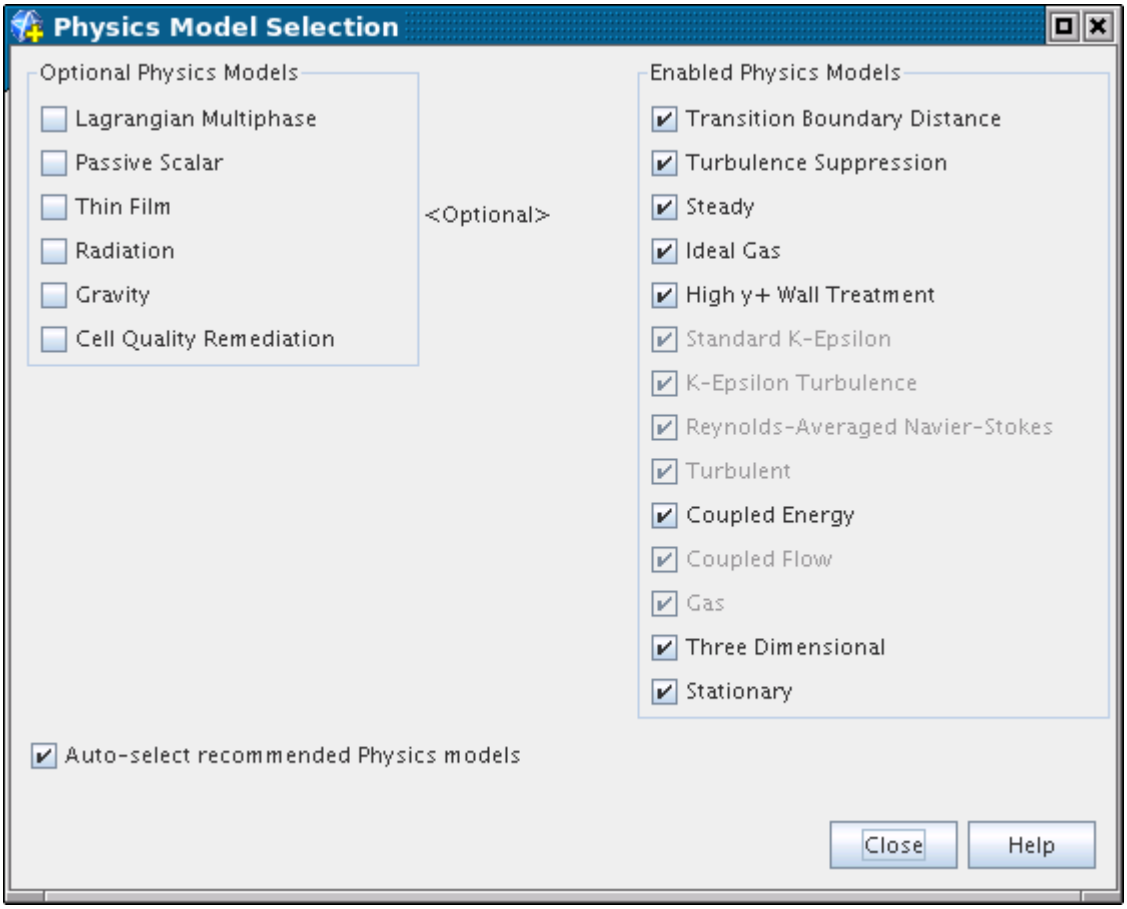
Using Auto-Select

This section describes how to select a model using automatic selection of recommended models. This is the default mode of selection, and is indicated by the **Auto-select recommended models** checkbox being ticked.

To select the Turbulence Suppression model, first select the turbulence model. Then in the *Optional Models* group box, select the **Turbulence Suppression** radio button.



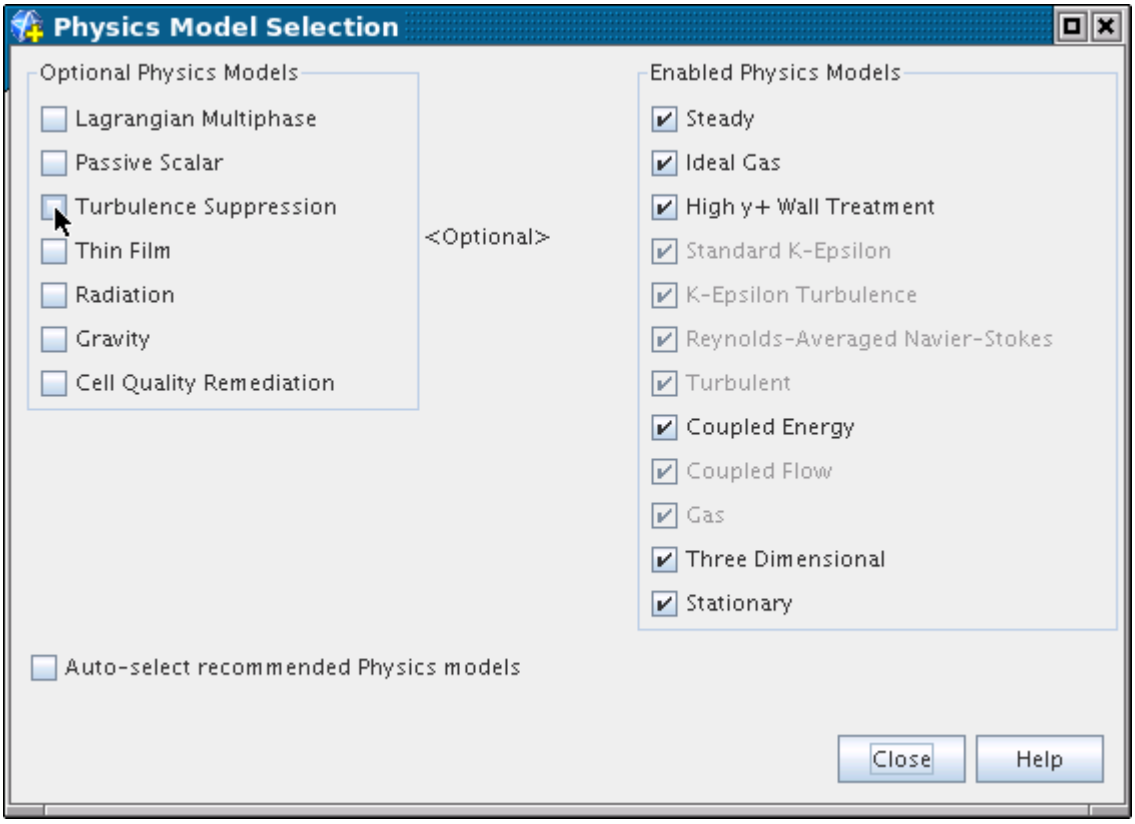
The Transition Boundary Distance model will be automatically selected.



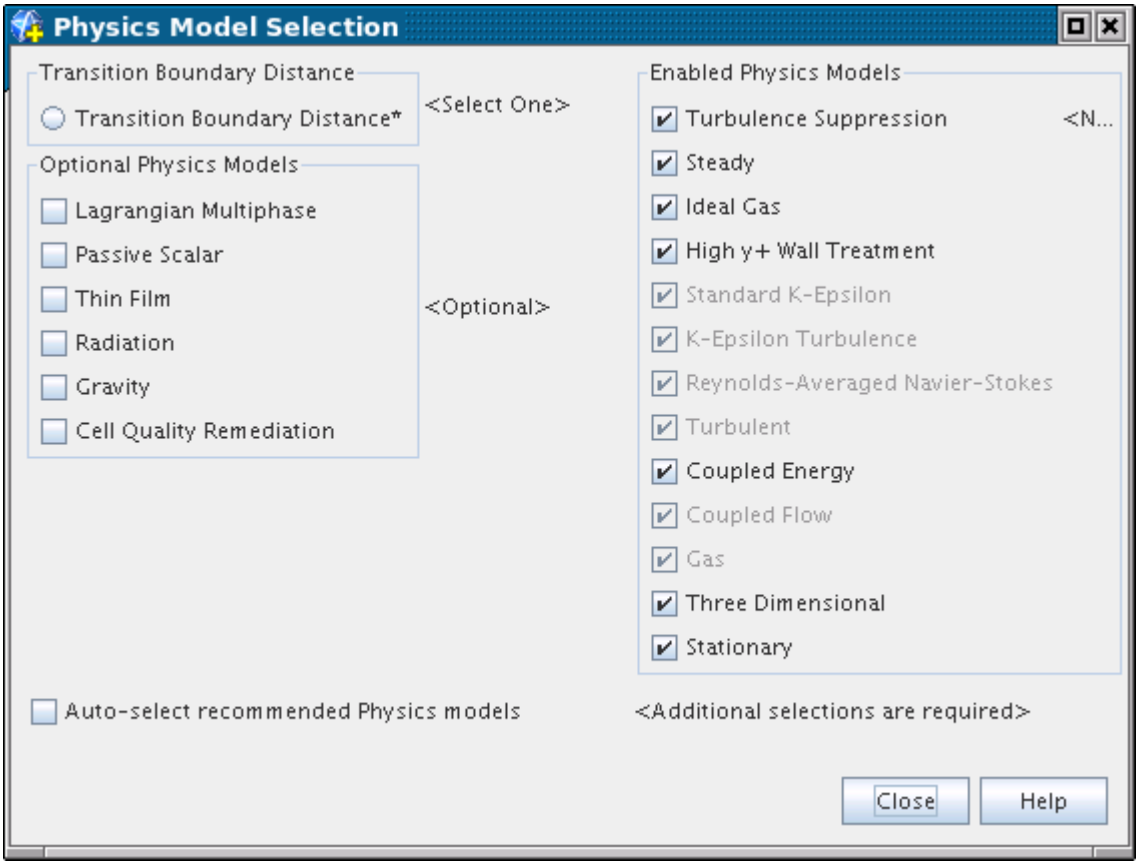
Without Using Auto-Select

This section describes how to select the Turbulence Suppression model without using automatic selection of recommended models. This means that the Transition Boundary Distance model will not be automatically selected.

To select the Turbulence Suppression model, first select the turbulence model. Then in the *Optional Models* group box, select the **Turbulence Suppression** radio button.



Even if the **Auto-select recommended models** checkbox is deactivated in the *Model Selection* dialog only one radio button, **Transition Boundary Distance**, will be presented in the *Transition Boundary Distance* group box. Therefore the selection procedure is identical toUsing Auto-Select.



Running a Simulation with the Turbulence Suppression Model

The activation of the Transition model and the selection of the turbulence suppression option on a wall boundary are clearly essential steps. However, several other issues are important.

These include:

- The choice of turbulence model and wall treatment
- The near-wall mesh resolution
- The inflow or free-stream turbulence boundary conditions

Choice of Turbulence Model and Wall Treatment

There is no restriction on the choice of turbulence model when the Transition model is activated. However, it makes the most sense to use models that are designed to resolve the laminar sublayer.

These include (but are not necessarily limited to):

- The Standard Spalart-Allmaras model
- The Realizable Two-Layer K-Epsilon model
- The Lien low-Reynolds number model
- The Abe-Kondoh-Nagano low-Reynolds number model
- The Standard K-Omega model

- The SST K-Omega model
- The Linear Pressure Strain Two-Layer model

In each case the low-y^+ or all-y^+ wall treatment model should be used.

Near-Wall Mesh Resolution

Since the wall shear stress is computed using laminar flow relationships rather than wall functions adjacent to a transition boundary, the mesh should be sufficiently fine for acceptable accuracy. A sound approach would be to ensure that the mesh is adequately refined in the turbulent region and that the near-wall mesh spacing is not discontinuous.

Inflow or Free-Stream Turbulence Boundary Conditions

Suppressing turbulence in one region of the flow does not guarantee that the flow will become rapidly turbulent where it is not suppressed. The onset of transition, though not accurately predicted by turbulence models, will be affected by the Reynolds number and the free-stream turbulence values. Some low-Reynolds number turbulence models can be quite sensitive to the free-stream turbulence values, so some artful tuning of the inflow turbulence conditions might be required to get the desired transition effect.

Turbulence Suppression Model Reference

The **Turbulence Suppression** model allows you to suppress turbulence effects adjacent to a specified transition boundary.

Provided By	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Turbulence Suppression**	
Requires	*Turbulence*: **Reynolds-Averaged Navier-Stokes** *Reynolds-Averaged Turbulence*: any *K-Epsilon Turbulence*, *K-Omega Turbulence*, *Reynolds Stress Turbulence*, *Spalart-Allmaras Turbulence*: any	
Activates	Physics Models	*Transition Boundary Distance*: **Transition Boundary Distance***
	Boundary Inputs	See Boundary Settings.

Boundary Settings

Note: Boundary types that do not require setting any conditions or values are not listed.

Wall

Turbulence Suppression Option

Controls whether the wall boundary is a transition boundary.

Turbulence Suppression Option	Corresponding Physics Value Nodes
On Suppresses turbulence near the wall boundary.	None.
Off Does not suppress turbulence.	None.

Transition Boundary Distance Model Reference

The **Transition Boundary Distance** model allows you to calculate the distance from the field point to the nearest point on a transition boundary.

Provided By	**[physics continuum]** > **Models** > *Transition Boundary Distance*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Transition Boundary Distance**	
Requires	*Turbulence*: **Reynolds-Averaged Navier-Stokes** *Reynolds-Averaged Turbulence*: any *K-Epsilon Turbulence*, *K-Omega Turbulence*, *Reynolds Stress Turbulence*, *Spalart-Allmaras Turbulence*: any *Optional Models*: **Turbulence Suppression**	
Activates	Solvers	**Transition Boundary Distance** See Transition Boundary Distance Solver Properties.
	Field Functions	**Transition Boundary Distance** **Transition Boundary Distance Factor** See Field Functions.

Transition Boundary Distance Solver Properties

The **Transition Boundary Distance** solver controls the transition boundary distance solution in all the continua for which transition boundary distance is calculated.

Solver Frozen

When **On** , prevents the solver from updating the solution while iterating. The property is **Off** by default.

Field Functions

Transition Boundary Distance

The transition boundary distance, which is defined as the distance from the field point to the nearest point on a transition boundary.

Transition Boundary Distance Factor

An integer number that is zero when the transition boundary distance is less than the wall distance, and unity when it is greater.

Gamma ReTheta Transition Model

The Gamma ReTheta transition model is a two-equation correlation-based transition model that provides a true predictive capability for the onset of transition in a turbulent boundary layer.

Correlation-based transition models, such as that by Abu-Ghannam and Shaw [41], required the evaluation of non-local quantities such as momentum thickness Reynolds number. These quantities cannot be evaluated on unstructured meshes, which precludes their practical application in modern unstructured CFD codes.

The Gamma ReTheta transition model [43], [45] is a correlation-based transition model that has been specifically formulated for unstructured CFD codes. The evaluation of momentum thickness Reynolds number is avoided by relating this quantity to vorticity-based Reynolds number. In addition, a correlation for transition onset momentum thickness Reynolds number defined in the free stream is propagated into the boundary layer by a transport equation. An intermittency transport equation is further used in such a way that the source terms attempt to mimic the behavior of algebraic engineering correlations.

The Gamma ReTheta transition model, as originally published, is incomplete, since two critical correlations were claimed to be proprietary and hence omitted. One justification for such an omission is that the model provides a “framework” for users to implement their own correlations. Therefore, the model’s implementation is done in such a way as to allow the custom specification of correlations via field functions. Recognizing, however, that the process of calibrating the needed correlations is complex and time consuming, the required correlations have been carefully calibrated and are provided by default within the model.

Furthermore, the original Gamma ReTheta transition model does not account for an important type of transition mechanism, the cross-flow instabilities. Langtry et al. [44] proposed a correlation, implemented as an additional source term for the transported momentum thickness Reynolds number, to account for such instabilities. This correlation is incorporated as optional in STAR-CCM+.

Contents:

Defining the Free-Stream Edge
Defining Correlations Using Field Functions
Gamma ReTheta Transition Model Reference

Defining the Free-Stream Edge

Although the Gamma ReTheta transition model purports to be defined in terms of local variables, it is really semi-local since it depends on the definition of the $Re_{\theta t}$ through a correlation defined in the free stream.

This free-stream value is transported through the boundary layer through the variable $\overline{Re_{\theta t}}$. In order to specify the value of $Re_{\theta t}$ using a correlation, the location of the free-stream edge must be defined.

Since there is no completely general way to define the free stream, it is up to the user to create a user field function that suitably defines the free stream in such a way that the field function takes the value of 1 in the free stream and 0 inside the boundary layer. A good starting point for this definition is in terms of Wall Distance. An example would be:

```
$WallDistance > 0.005?1:0
```

where it is estimated that the boundary layer is everywhere thinner than 5mm. The user field function, once defined and named, must then be assigned to the *Free Stream Edge* property of the Gamma ReTheta transition model. See Gamma ReTheta Transition Model Reference.

Defining Correlations Using Field Functions

The *Correlation Method* property of the Gamma ReTheta transition model allows you to use field functions to specify the calibrations used in the model.

Once you set *Correlation Method* to **User Field Function**, the following child nodes appear under the **Gamma-Re Theta Transition** node:

Transition onset Re Correlation

Defines the $Re_{\theta t}$ correlation. The following options are available:

- **Modified**
- **Modified NZPG**
- **Original**
- **Original NZPG**
- **Abu-Ghannam & Shaw**
- **Abu-Ghannam & Shaw NZPG**
- **User Field Function**

F_length Correlation

Field Function that defines the F_{length} correlation.

Re_crit Correlation

Field Function that defines the $Re_{\theta c}$ correlation.

Re_theta_t Correlation

Field Function that defines the $Re_{\theta t}$ correlation.

For more information, see Gamma ReTheta Transition Model Reference.

To emulate the internal (Suluksna-Juntasaro) calibration, you could set *Transition onset Re Correlation* to **Modified** and specify the field functions for *F_length correlation* and *Re_crit correlation*, repsectively, as follows:

```
min(300,exp(7.168 - 0.01173*$TransitionReTheta) + 0.5)
```

```
min($TransitionReTheta, $TransitionReTheta*0.615 + 61.5)
```

If the **User Field Function** option is to be used to specify the $Re_{\theta t}$ correlation, the following field function could be used to emulate the internal (**Modified** option) implementation:

```
(100*$TurbulenceIntensity > 1.3) ? 331.5*pow(100*$TurbulenceIntensity
- 0.5658, -0.671) : (1173.51 - 589.428*100*$TurbulenceIntensity+0.2196/
(10000*$TurbulenceIntensity*$TurbulenceIntensity))
```

Gamma ReTheta Transition Model Reference

The **Gamma-ReTheta Transition** model allows you to solve for the turbulence intermittency γ and the transition momentum thickness Reynolds number $\overline{Re_{\theta t}}$ to predict the onset of transition in a turbulent boundary layer.

Theory	See Gamma ReTheta Transition Formulation.	
Provided By	**[physics continuum]** > **Models** > *Transition*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Gamma-ReTheta Transition**	
Requires	*Turbulence*: **Reynolds Averaged Navier-Stokes*** *Reynolds-Averaged Turbulence*: **K-Omega Turbulence** *K-Omega Turbulence Models*: **SST (Menter) K-Omega**	
	Turbulence: **Detached Eddy Simulation** *Detached Eddy Simulation*: **SST (Menter) K-Omega Detached Eddy**	
Properties	See Properties.	
Activates	Model Controls (child nodes)	**Cross-flow parameters** (optional, see Cross-Flow term) **Transition onset Re correlation** (optional, see Correlation Method) **F_length correlation** (optional, see Correlation Method) **Re_crit correlation** (optional, see Correlation Method) **Re_theta_t_correlation** (optional, see Correlation Method)
	Initial Conditions	**Intermittency** See Initial Conditions.
	Solvers	**GammaReTheta Transition** See Solvers.
	Monitors	**Intermittency** **ReTheta_t** (transported transition momentum thickness Reynolds number)
	Field Functions	**Effective Intermittency** **Flength Correlation** **Intermittency** **ReThetaCrit Correlation** **Specific Dissipation Rate Reynolds Number** **Transition Momentum Thickness Reynolds Number** **Turbulence Intensity** See Field Functions.

Properties

Convection

Controls the convection scheme.

- **1st-order**: Selects the first-order upwind convection scheme.
- **2nd-order**: Selects the second-order upwind convection scheme.

Cross-Flow term

Controls whether to apply a correlation for stationary cross-flow instabilities. The correlation accounts for transition that is triggered by surface roughnesses whose height is in the order of micrometers.

When **On**, STAR-CCM+ includes the cross-flow term of Eqn. (464) in the equation for the transported transition momentum thickness Reynolds number $\overline{Re_{\theta t}}$. The default setting is **Off**.

Cross-flow parameters

Child node that provides the model constant *C_scf*, that is C_{SCF} in Eqn. (464), and the cross-flow inducing roughness height in meters *h(m)*, that is h in Eqn. (467).

Activating this option introduces a Reynolds number dependency to the Gamma ReTheta model.

Free Stream Edge

Field Function that defines the free-stream edge. See Free-Stream Edge Definition.

Sigma_f

The coefficient σ_f , see Eqn. (440).

ca2

The coefficient c_{a2} , see Eqn. (450).

ce2

The coefficient c_{e2} , see Eqn. (450).

Intermittency Minimum

The minimum value that the transported variable γ is permitted to have. An appropriate value is a small number that is greater than the floating point minimum of the computer.

Secondary Gradients

Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces.

- **On**: Include both secondary gradients.
- **Off**: Exclude both secondary gradients.
- **Interior Only**: Include the interior secondary gradients only.
- **Boundaries Only**: Include the boundary secondary gradients only.

Correlation Method

Choose how the correlations for $Re_{\theta c}$ and F_{length} are to be specified.

- **Suluksna-Juntasaro**: Use the internally-specified correlations (see Eqn. (481), Eqn. (500) and Eqn. (501)).
- **User Field Function**: Use correlations defined by user specified field function.

Transition onset Re Correlation

Child node that controls the correlation used for $Re_{\theta t}$.

- **Modified**: Selects the $Re_{\theta t}$ correlation of Eqn. (481).
- **Modified NZPG**: Selects the $Re_{\theta t}$ correlation of Eqn. (497).
- **Original**: Selects the $Re_{\theta t}$ correlation of Eqn. (480).
- **Original NZPG**: Selects the $Re_{\theta t}$ correlation of Eqn. (492).
- **Abu-Ghannam & Shaw**: Selects the $Re_{\theta t}$ correlation of Eqn. (479).
- **Abu-Ghannam & Shaw NZPG**: Selects the $Re_{\theta t}$ correlation of Eqn. (489).
- **User Field Function**: Allows specification of $Re_{\theta t}$ through a user field function. Activates the **Re_theta_t correlation** child node.

F-length correlation

Child node that defines the field function for the F_{length} correlation.

Re_crit correlation

Child node that defines the field function for the $Re_{\theta c}$ correlation.

Re_theta_t correlation

Child node that defines the field function for the $Re_{\theta t}$ correlation.

Sigma_ReTheta

The coefficient $\sigma_{\theta t}$, see Eqn. (454).

ca1

The coefficient c_{a1} , see Eqn. (441).

ce1

The coefficient c_{e1} , see Eqn. (441).

cTheta_t

The coefficient $c_{\theta t}$, see Eqn. (455).

s1

The coefficient s_1 , see Eqn. (478).

ReThetaTMinimum

The minimum value that the transported variable $\overline{Re_{\theta t}}$ is permitted to have. The default value is 20.

Initial Conditions

Intermittency

Scalar profile value to specify the transported variable γ.

Gamma ReTheta Transition Solver

This solver controls the solution of the Gamma ReTheta transition model in all the continua for which the model is activated.

For each transported variable, γ and $\overline{Re_{\theta t}}$, the basic steps that are involved in the solution update are as follows:

- Update boundary conditions.
- Compute the reconstruction gradients and cell gradients.
- Create the linear system of Eqn. (1615) using the discretization methods that are described in the K-Omega model formulation.
- Compute the residual sum for monitoring convergence:

$$R = \sum_{\text{cells}} |r|$$

- Solve the linear system.
- Update the transported variable field.

Under-Relaxation Factor

At each iteration, governs the extent to which the old solution is supplanted by the newly computed solution. The default value is 0.8. For the theoretical background, see Eqn. (1625) in Finite Volume Discretization.

Freestream Update Frequency

This parameter governs the frequency at which the correlation for $Re_{\theta t}$ will be updated in the free stream. The default is 1, which manes that the update occurs every iteration. Increasing this parameter to a larger interval, say 10, might improve parallel performance in certain situations. This is because the free-stream values are stored in a KD tree that is broadcast to each parallel node, incurring a communication overhead.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Field Functions

Effective Intermittency

The transported variable γ_{eff} , see Eqn. (477).

Flength Correlation

The specified correlation for F_{length} , see Eqn. (501).

Intermittency

The transported variable γ , see Eqn. (440).

ReThetaCrit Correlation

The specified correlation for $Re_{\theta c}$, see Eqn. (500).

Specific Dissipation Rate Reynolds Number

The specific dissipation rate Reynolds number, Re_{ω} , see Eqn. (462).

Transition Momentum Thickness Reynolds Number

The transported variable $\overline{Re_{\theta t}}$, see Eqn. (454).

Turbulence Intensity

The turbulence intensity, defined as $I = \sqrt{2/3k}/U$

Gamma Transition Model

The Gamma transition model is a one-equation correlation-based transition model that provides a true predictive capability for the onset of transition in a turbulent boundary layer.

The Gamma transition model [46] is a simplified version of the Gamma ReTheta Model. It avoids the transport equation for the momentum thickness Reynolds number and solves for turbulence intermittency only. Like the Gamma ReTheta model, it uses transition onset criteria in terms of the momentum thickness Reynolds number, but computes them algebraically using local variables. Furthermore, the complex formulations of the experimental correlations are simplified.

Like the Gamma ReTheta model, the Gamma transition model incorporates an optional correlation for cross-flow instabilities. However, for the Gamma Transition model, the correlation is implemented as a modification of the transition trigger function, which activates the production term for the transported intermittency.

Contents:

Gamma Transition Model Reference

Gamma Transition Model Reference

The **Gamma Transition** model allows you to solve for the turbulence intermittency γ to predict the onset of transition in a turbulent boundary layer.

Theory	See Gamma Transition Formulation.	
Provided By	**[physics continuum]** > **Models** > *Transition*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Gamma Transition**	
Requires	*Turbulence*: **Reynolds Averaged Navier-Stokes*** *Reynolds-Averaged Turbulence*: **K-Omega Turbulence** *K-Omega Turbulence Models*: **SST (Menter) K-Omega**	
	Turbulence: **Detached Eddy Simulation** *Detached Eddy Simulation*: **SST (Menter) K-Omega Detached Eddy**	
Properties	See Properties.	
Activates	Initial Conditions	**Intermittency** See Initial Conditions.
	Solvers	**Gamma Transition** See Solvers.
	Monitors	**Intermittency**
	Field Functions	**Intermittency** **Specific Dissipation Rate Reynolds Number** **Turbulence Intensity**

		See Field Functions.

Properties

Convection

Controls the convection scheme.

- **1st-order**: Selects the first-order upwind convection scheme.
- **2nd-order**: Selects the second-order upwind convection scheme.

Cross-Flow Term

Controls whether to apply a correlation for stationary cross-flow instabilities. The correlation accounts for transition that is triggered by surface roughnesses whose height is in the order of micrometers.

When **On**, modifies the transition trigger function F_{onset} (see Eqn. (520)), which activates the production term in the equation for the transported intermittency γ. The default setting is **Off**.

Cross-flow parameters

Child node that provides the model constant *C_scf*, that is C_{SCF} in Eqn. (522), and the cross-flow inducing roughness height in meters *h(m)*, that is $h(m)$ in Eqn. (531). Increasing the default values accelerates the cross-flow induced transition.

Sigma_f

The coefficient σ_f , see Eqn. (502).

ca2

The coefficient c_{a2} , see Eqn. (512).

ce2

The coefficient c_{e2} , see Eqn. (512).

Intermittency Minimum

The minimum value that the transported variable γ is permitted to have. An appropriate value is a small number that is greater than the floating point minimum of the computer.

Secondary Gradients

Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces.

- **On**: Include both secondary gradients.
- **Off**: Exclude both secondary gradients.
- **Interior Only**: Include the interior secondary gradients only.
- **Boundaries Only**: Include the boundary secondary gradients only.

Initial Conditions

Intermittency

Scalar profile value to specify the transported variable γ.

Gamma Transition Solver

This solver controls the solution of the Gamma transition model in all the continua for which the model is activated.

For the transported variable γ the basic steps that are involved in the solution update are as follows:

- Update boundary conditions.
- Compute the reconstruction gradients and cell gradients.
- Create the linear system of equations using the discretization methods that are described in the K-Omega model formulation.
- Compute the residual sum for monitoring convergence:

$$R = \sum_{\text{cells}} |r|$$

- Solve the linear system.
- Update the transported variable field.

Under-Relaxation Factor

At each iteration, governs the extent to which the old solution is supplanted by the newly computed solution. The default value is 0.8. For the theoretical background, see Eqn. (1625) in Finite Volume Discretization.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Field Functions

Intermittency

The transported variable γ , see Eqn. (502).

Specific Dissipation Rate Reynolds Number

The specific dissipation rate Reynolds number defined as $Re_{\omega} = \frac{\omega d^2}{\nu}$.

Turbulence Intensity

The turbulence intensity, defined as $I = \sqrt{2/3k}/U$.

Transition Guidelines

A number of guidelines are advised when setting up simulations that use the Gamma or the Gamma ReTheta transition model. These guidelines concern mesh creation, the turbulence specification on inflow boundaries, and convergence.

Mesh Creation

Validation studies show that the Gamma ReTheta and the Gamma transition models require a high-quality, refined low-Reynolds number mesh:

- Create a mesh with a wall-normal spacing that is fine enough to guarantee Wall Y+ values between 0.1 and 1.
- Make sure that the streamwise mesh spacing near the transition region is fine enough to accurately capture the onset of transition.
- Use either the all-y^+ or the low-y^+ wall treatment formulation.
- If transition is induced by laminar separation, then refine the mesh to resolve the curvature of the body where the separation exists, and to capture the laminar separation bubble.

Inflow Turbulence Specification

The Gamma ReTheta and the Gamma transition models attempt to account for the effect of the freestream turbulence levels on the transition location. Therefore, pay proper attention to the turbulence specification on inflow boundaries to encounter realistic turbulence intensities in the boundary layers within the transition zone.

Whenever possible, provide experimental data that describes the behavior of the freestream turbulence. With three or more data points, you can determine the dissipation rate and subsequently the inflow turbulence values through a regression analysis of the data.

The freestream turbulence decay laws, which are derived from the SST K-Omega model transport equation, are as follows (using the default coefficients):

$$k = k_{inlet}\left[1 + 0.0828\frac{x}{U_\infty}\omega_{inlet}\right]^{-1.087} \tag{434}$$

$$\omega = \omega_{inlet}\left[1 + 0.0828\frac{x}{U_\infty}\omega_{inlet}\right]^{-1} \tag{435}$$

where x is the streamwise distance and U_∞ is the (assumed constant) freestream velocity.

Using the following relations for turbulence intensity I and turbulent viscosity ratio ($TVR = \mu_t/\mu$):

$$I = \frac{\sqrt{(2/3)k}}{U} \tag{436}$$

$$TVR = \frac{k}{\omega\mu} \tag{437}$$

you can transform Eqn. (434) and Eqn. (435) into:

$$I = I_{inlet}\left[1 + \frac{0.1242 x \rho U_{\infty} I_{inlet}^{2}}{\mu\, TVR_{inlet}}\right]^{-0.5435} \tag{438}$$

$$TVR = TVR_{inlet}\left[1 + \frac{0.1242 x \rho U_{\infty} I_{inlet}^{2}}{\mu\, TVR_{inlet}}\right]^{-0.087} \tag{439}$$

Care must be taken with the specification of turbulent viscosity ratio at the inlet because large values of effective viscosity cause excessive diffusion. The exponent in Eqn. (439) is small, so that the turbulent viscosity does not decay significantly, allowing the inlet values to propagate far downstream. Unfortunately, the smaller the value of the turbulent viscosity ratio, the faster the turbulence will decay. In the absence of measured data, the proper turbulent inflow boundary conditions become a delicate balance between an acceptable turbulent decay rate and an acceptably low turbulent viscosity ratio.

Since the turbulent viscosity ratio is, in effect, a turbulent Reynolds number, it is impossible to specify what an acceptable value is for any given flow. (Higher bulk Reynolds numbers would suggest higher turbulent Reynolds numbers, and hence higher levels of acceptable turbulent viscosity ratio.) However, a rough guideline is that the freestream turbulent viscosity ratio should be significantly lower than the turbulent viscosity that occurs within the boundary layer. For natural transition problems and lower freestream turbulence bypass transition, a desirable goal would be to have a TVR from 1 through 10. This prevents the high levels of effective viscosity from contaminating the laminar boundary layer and affecting the laminar skin friction prediction.

In some situations, such as wind tunnels with very low background turbulence, it might be a challenge to obtain a suitable compromise between these two variables. This issue is often exacerbated when a far-field boundary is specified a large distance away from the body.

Convergence

Apart from the fact that the Gamma ReTheta and the Gamma transition models require the solution of additional transport equations, incurring an additional computational overhead as a result, an impact on solution covertness must be expected.

It has been noted that the interaction between flow, turbulence, and the transition model can increase the number of iterations to convergence significantly. Furthermore, the intermittency transport equation residuals are naturally noisy. This is attributed to the strong dependence of the source terms on higher powers of the mean flow strain rates, and the fact the source terms are not differentiable functions.

Transition Formulation

This section contains the formulation of the Gamma ReTheta transition model and the Gamma transition model.

Contents:

Gamma ReTheta Transition Formulation
Gamma Transition Formulation
Coupling with SST (Menter) K-Omega Model
Gamma ReTheta Transition Nomenclature
Gamma ReTheta Transition Bibliography

Gamma ReTheta Transition Formulation

The formulation of the Gamma ReTheta transition model includes the transport equations and ReTheta correlations.

Contents:

Intermittency Transport Equation
Transition Momentum Thickness Reynolds Number Transport Equation
Separation-Induced Transition Modification
Correlations for ReTheta_t
Correlations for ReTheta_c and F_length

Intermittency Transport Equation

The transport equation for the intermittency γ is formulated as:

$$\frac{d}{dt}\int_V \rho\gamma\, dV + \int_A \rho\gamma\left(\mathbf{v}-\mathbf{v}_g\right)\cdot d\mathbf{a} = \int_A \left(\mu + \frac{\mu_t}{\sigma_f}\right)\nabla\gamma\cdot d\mathbf{a} + \int_V \left[P_\gamma - E_\lambda\right] dV \tag{440}$$

For details on how γ enters the SST K-Omega transport equations see Coupling with SST (Menter) K-Omega Model.

Production Term

The transition source term is defined as:

$$P_\gamma = F_{length} c_{a1} \rho S \left[\gamma F_{onset}\right]^{1/2}\left(1 - c_{e1}\gamma\right) \tag{441}$$

where:

$$F_{onset1} = \frac{Re_v}{2.193 Re_{\theta c}} \tag{442}$$

$$F_{onset2} = \min\left[\max\left(\left(F_{onset1}^4, F_{onset1}\right), 2\right)\right] \tag{443}$$

$$F_{onset3} = \max\left[1 - \left(Re_T/2.5\right)^3, 0\right] \tag{444}$$

$$F_{onset} = \max\left(F_{onset2} - F_{onset3}, 0\right) \tag{445}$$

Re_v is the strain-rate Reynolds number defined as:

$$Re_v = \frac{\rho\, d^2 S}{\mu} \tag{446}$$

and S is the modulus of the mean strain rate tensor:

$$S = |\mathbf{S}| = \sqrt{2\mathbf{S}:\mathbf{S}^T} = \sqrt{2\mathbf{S}:\mathbf{S}} \tag{447}$$

where:

$$\mathbf{S} = \frac{1}{2}\left(\nabla\mathbf{v} + \nabla\mathbf{v}^T\right) \tag{448}$$

The turbulent Reynolds number R_T is defined as:

$$Re_T = \frac{k}{\nu\omega} \tag{449}$$

Note that the correlations for $Re_{\theta c}$ and F_{length} are not specified by Langtry and Menter.

Destruction Term

The destruction/relaminarization source is defined as:

$$E_\lambda = C_{a2}\rho W \gamma F_{turb}\left(c_{e2}\gamma - 1\right) \tag{450}$$

where:

$$F_{turb} = \exp\left[-\left(Re_T/4\right)^4\right] \tag{451}$$

and:

$$W = \sqrt{2\mathbf{W}:\mathbf{W}^T} \tag{452}$$

where the vorticity tensor $\mathbf{W}$ is defined:

$$\mathbf{W} = \frac{1}{2}\left(\nabla\mathbf{v} + \nabla\mathbf{v}^T\right) \tag{453}$$

Model coefficients

The model coefficients for the intermittency transport equation are defined as:

$C_{e1} = 1.0,\ C_{a1} = 2.0,\ C_{e2} = 50,\ C_{a2} = 0.06,\ \sigma_f = 1.0$

Transition Momentum Thickness Reynolds Number Transport Equation

The transition momentum thickness Reynolds number transport equation is formulated as:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_V \rho\overline{Re_{\theta t}}\,\mathrm{d}V + \int_A \rho\overline{Re_{\theta t}}\left(\mathbf{v}-\mathbf{v}_g\right)\cdot\mathrm{d}\mathbf{a} = \int_A \sigma_{\theta t}\left(\mu+\mu_t\right)\nabla\overline{Re_{\theta t}}\cdot\mathrm{d}\mathbf{a} + \int_V \left(P_{\theta t} + D_{SCF}\right)\mathrm{d}V \tag{454}$$

Source Term

The source term is defined as:

$$P_{\theta t} = C_{\theta t}\frac{\rho}{t}\left(Re_{\theta t} - \overline{Re_{\theta t}}\right)\left(1-F_{\theta t}\right) \tag{455}$$

where:

$$t = \frac{500\nu}{U^2} \tag{456}$$

and:

$$F_{\theta t} = \min\left\{\max\left[F_{wake}\exp\left(-\left(\frac{d}{\delta}\right)^4\right), 1-\left(\frac{C_{e2}\gamma-1}{C_{e2}-1}\right)^2\right], 1\right\} \tag{457}$$

Using:

$$\delta = 50W\frac{d}{U}\delta_{BL} \tag{458}$$

$$\delta_{BL} = \frac{15}{2}\theta_{BL} \tag{459}$$

$$\theta_{BL} = \frac{\overline{Re_{\theta t}}\nu}{U} \tag{460}$$

then:

$$\frac{d}{\delta} = \frac{U^2}{375W\nu\overline{Re_{\theta t}}} \tag{461}$$

And F_{wake} is given by:

$$Re_{\omega} = \frac{\omega d^2}{\nu} \tag{462}$$

$$F_{wake} = \exp\left[-\left(Re_{\omega}/10^{5}\right)^{2}\right] \tag{463}$$

Cross-Flow Term

The cross-flow term is defined according to [44] as:

$$D_{SCF} = C_{\theta t}\frac{\rho}{t}C_{SCF}\min\left(Re_{SCF} - \overline{Re_{\theta t}},\ 0\right)F_{\theta t2} \tag{464}$$

where:

$$t = \min\left(\frac{500\mu}{\rho U^{2}}, \frac{\rho L^{2}}{(\mu + \mu_{t})}\right) \tag{465}$$

and:

$$F_{\theta t2} = \min\left(F_{wake}e^{-\left(\frac{d}{\delta}\right)^{4}} - \overline{Re_{\theta t}},\ 1\right) \tag{466}$$

The empirical correlation for stationary cross-flow transition is defined as:

$$Re_{SCF} = \frac{\theta_{t}\rho\left(\frac{U}{0.82}\right)}{\mu} = -35.088ln\left(\frac{h}{\theta_{t}}\right) + 319.51 + f\left(+\Delta H_{SCF}\right) - f\left(-\Delta H_{SCF}\right) \tag{467}$$

where h is the cross-flow inducing roughness height. The cross-flow strength shift terms are given by:

$$f\left(+\Delta H_{SCF}\right) = 6200\left(+\Delta H_{SCF}\right) + 50000\left(+\Delta H_{SCF}\right)^{2} \tag{468}$$

$$f\left(-\Delta H_{SCF}\right) = 75\tanh\left(\frac{-\Delta H_{SCF}}{0.0125}\right) \tag{469}$$

using:

$$+\Delta H_{SCF} = \max\left(0.1066 - \Delta H_{SCF}, 0\right) \tag{470}$$

$$-\Delta H_{SCF} = \max\left[-1.0\left(0.1066 - \Delta H_{SCF}\right), 0\right] \tag{471}$$

and:

$$\Delta H_{SCF} = H_{SCF}\left[1.0 + \min\left(\frac{\mu_{t}}{\mu}, 0.4\right)\right] \tag{472}$$

The non-dimensional cross-flow strength is defined as:

$$H_{SCF} = \frac{d\mathrm{W}_{\mathrm{Streamwise}}}{U} \tag{473}$$

where $\mathrm{W}_{\mathrm{Streamwise}}$ is the stream-wise vorticity given by:

$$W_{\text{Streamwise}} = |\mathbf{u} \cdot \mathbf{w}| \tag{474}$$

$\mathbf{u}$ is the unit velocity vector and $\mathbf{w} = \nabla \times \mathbf{v}$ is the vorticity vector.

The correlation for stationary cross-flow Eqn. (467) is an implicit function that is solved iteratively using a Newton-Raphson method to derive the momentum thickness θ_t.

Model Coefficients

The model coefficients are defined as:

$C_{\theta t} = 0.03,\ C_{SCF} = 0.6,\ \sigma_{\theta t} = 2.0$

Separation-Induced Transition Modification

The modification for separation-induced transition is:

$$\gamma_{sep} = \min\left[s_1 \max\left(0, \frac{Re_v}{3.235 Re_{\theta c}} - 1\right) F_{reattach}, 2\right] \tag{475}$$

where:

$$F_{reattach} = \exp\left[\left(\frac{-Re_T}{15}\right)^4\right] \tag{476}$$

$$\gamma_{eff} = \max\left(\gamma, \gamma_{sep}\right) \tag{477}$$

and the coefficient s_1 has the default value:

$$s_1 = 2 \tag{478}$$

Correlations for ReTheta_t

A well-established correlation for $Re_{\theta t}$ was developed by Abu-Ghannam and Shaw [41]. In the case of a zero pressure gradient, this correlation is:

$$Re_{\theta t} = 163 + e^{(6.91 - Tu)} \tag{479}$$

where Tu is the turbulence intensity expressed as a percentage.

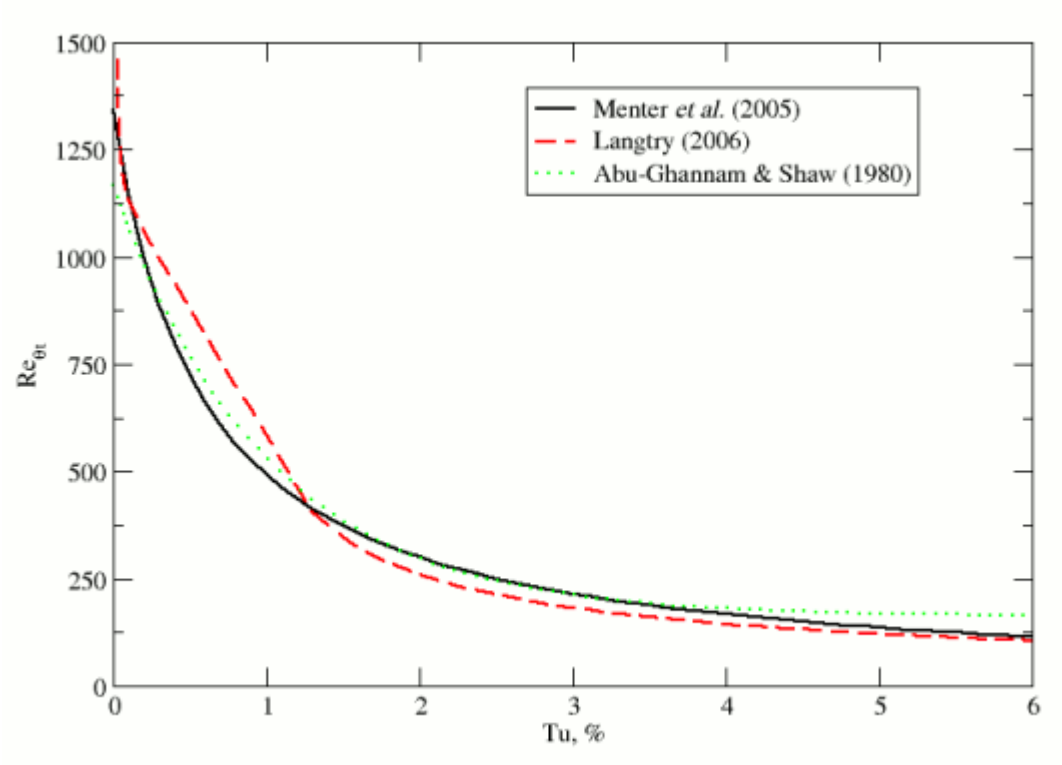
An alternative expression, developed for the Gamma ReTheta transition model by Menter et al. [45] is:

$$Re_{\theta t} = 803.73\left(Tu + 0.6067\right)^{-1.027} \tag{480}$$

and this was later revised by Langtry [43] to “improve the results at lower freestream turbulence intensities” as follows:

$$Re_{\theta t} = \begin{cases} \left(1173.51 - 589.428 Tu + 0.2196/Tu^2\right) \ ;\ Tu \le 1.3 \\ 331.5\left(Tu - 0.5658\right)^{-0.671};\ Tu > 1.3 \end{cases} \tag{481}$$

The three zero-pressure gradient correlations are plotted in the following figure. All three are provided in STAR-CCM+ to allow the user flexibility to customize the correlations, but Eqn. (481) is the default correlation used in the internal “Suluksna-Juntasaro” calibration.



$Re_{\theta t}$ vs. Free-Stream Turbulence Intensity

The three correlations presented for zero pressure gradient in Eqn. (479) through Eqn. (481) were originally presented in a form that include the effects of a pressure gradient. There is some doubt that the pressure gradient effects are indeed required in the $Re_{\theta t}$ correlation [49], since they arguably introduce a double accounting of the pressure gradient effects, but they are nevertheless optionally included in STAR-CCM+. In the presence of a pressure gradient, the correlations require the computation of the Thwaites parameter, λ_θ, and in the case of the Menter et al. [45] correlation, the acceleration parameter, K.

To avoid computing an integral parameter such as momentum thickness, the momentum thickness Reynolds number is estimated as:

$$Re_\theta = \frac{Re_\nu}{2.193} \tag{482}$$

Thwaite’s parameter, λ_θ, is defined:

$$\lambda_\theta = \frac{\theta^2}{\nu}\frac{\partial}{\partial s}U \tag{483}$$

with:

$$\theta = \frac{\nu}{U} R_{\theta t} \tag{484}$$

The parameter K is defined as:

$$K = \frac{\nu}{U^2} \frac{\partial}{\partial s} U \tag{485}$$

The turbulence intensity is obtained from the turbulent kinetic energy as follows:

$$Tu = 100 \frac{\sqrt{2k/3}}{U} \tag{486}$$

where:

$$U = |\mathbf{v}| \tag{487}$$

The streamwise velocity gradient is calculated as:

$$\frac{dU}{ds} = \frac{\left(\nabla \mathbf{v}^T \cdot \mathbf{v}\right) \cdot \mathbf{v}}{\mathbf{v} \cdot \mathbf{v}} \tag{488}$$

Since the momentum thickness θ must be derived from the momentum thickness Reynolds number Re_θ the correlation becomes an implicit function that is solved iteratively.

The non-zero pressure gradient (NZPG) correlation of Abu-Ghannam and Shaw [41] is:

$$Re_{\theta t} = 163 + e^{\left[F\left(\lambda_\theta\right)\left(1 - \frac{Tu}{6.91}\right)\right]} \tag{489}$$

where:

$$F\left(\lambda_\theta\right) = \begin{cases} 6.91 + 12.75\lambda_\theta + 63.64\lambda_\theta^2; \ \lambda_\theta \le 0 \\ 6.91 + 2.58\lambda_\theta - 12.27\lambda_\theta^2; \ \lambda_\theta > 0 \end{cases} \tag{490}$$

For numerical robustness, the following limits are applied:

$$\begin{gathered} -0.1 \le \lambda_\theta \le 0.1 \\ Re_{\theta t} \ge 20 \end{gathered} \tag{491}$$

The NZPG correlation of Menter et al. [45] is:

$$Re_{\theta t} = 803.73\left(Tu + 0.6067\right)^{-1.027} F\left(\lambda_\theta, K\right) \tag{492}$$

$$F\left(\lambda_\theta, K\right) = \begin{cases} 1 + F_\lambda \exp\left(-Tu/3\right); \ \lambda_\theta \le 0 \\ 1 + F_K\left[1 - \exp\left(-Tu/1.5\right)\right] + 0.556\left[1 - e^{-23.9\lambda_\theta}\right] e^{-Tu/1.5}; \ \lambda_\theta > 0 \end{cases} \tag{493}$$

where:

$$F_\lambda = 10.32\lambda_\theta + 89.47\lambda_\theta^2 + 265.51\lambda_\theta^3 \tag{494}$$

$$F_K = \left[0.0962\left(K \cdot 10^6\right) + 0.148\left(K \cdot 10^6\right)^2 + 0.0141\left(K \cdot 10^6\right)^3\right] \tag{495}$$

For numerical robustness, the following limits are applied:

$$\begin{gathered} -0.1 \le \lambda_\theta \le 0.1 \\ -3x10^{-6} \le K \le 3x10^{-6} \\ Re_{\theta t} \ge 20 \end{gathered} \tag{496}$$

The NZPG correlation of Langtry [43] is:

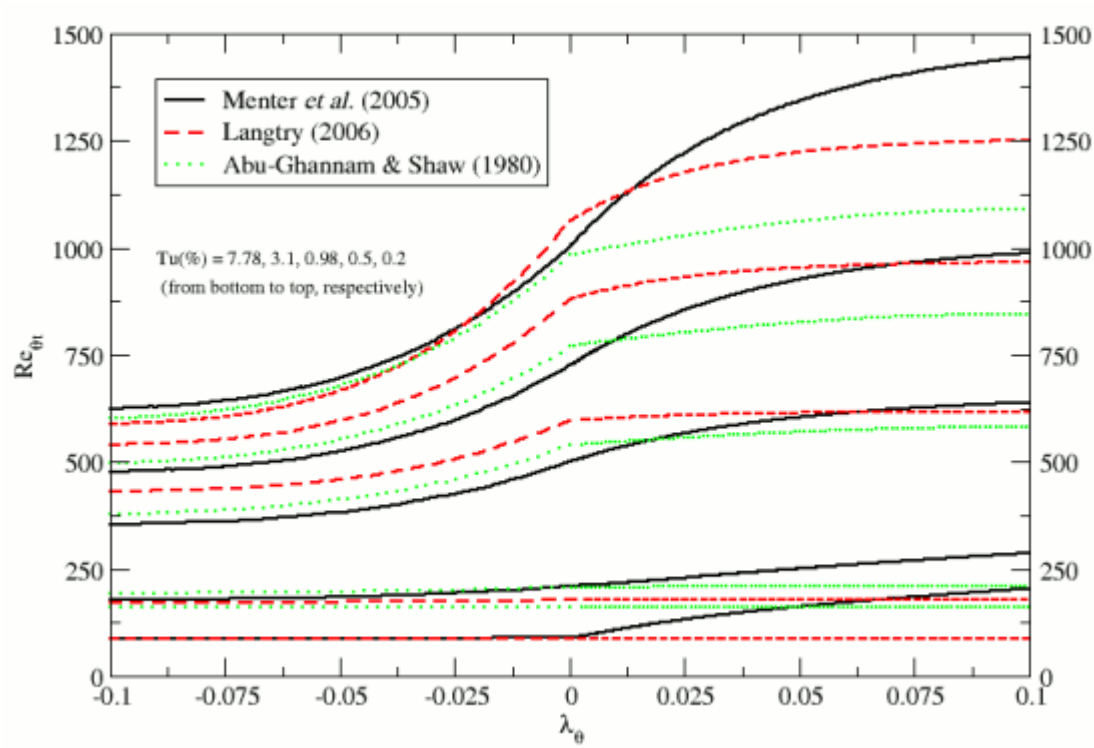
$$Re_{\theta t} = \begin{cases} 1173.51 - 589.428Tu + 0.2196/Tu^2 \quad F\ \left(\lambda_\theta\right)\ ;\ Tu \le 1.3 \\ 331.5\left(Tu - 0.5658\right)^{-0.671} F\left(\lambda_\theta\right);\ Tu > 1.3 \end{cases} \tag{497}$$

$$F\left(\lambda_\theta\right) = \begin{cases} 1 + 12.986\lambda_\theta + 123.66\lambda_\theta^2 + 405.689\lambda_\theta^3;\ \lambda_\theta \le 0 \\ 1 + 0.275\left[1 - \exp\left(-35\lambda_\theta\right)\right]\exp\left(-2Tu\right);\ \lambda_\theta > 0 \end{cases} \tag{498}$$

For numerical robustness, the following limits are applied:

$$\begin{gathered} -0.1 \le \lambda_\theta \le 0.1 \\ Tu \ge 0.027 \\ Re_{\theta t} \ge 20 \end{gathered} \tag{499}$$

A comparison of the three NZPG correlations is shown in the following figure for a range of free-stream turbulence intensities.



$Re_{\theta t}$ vs. λ_θ for various values of Free-Stream Turbulence Intensity

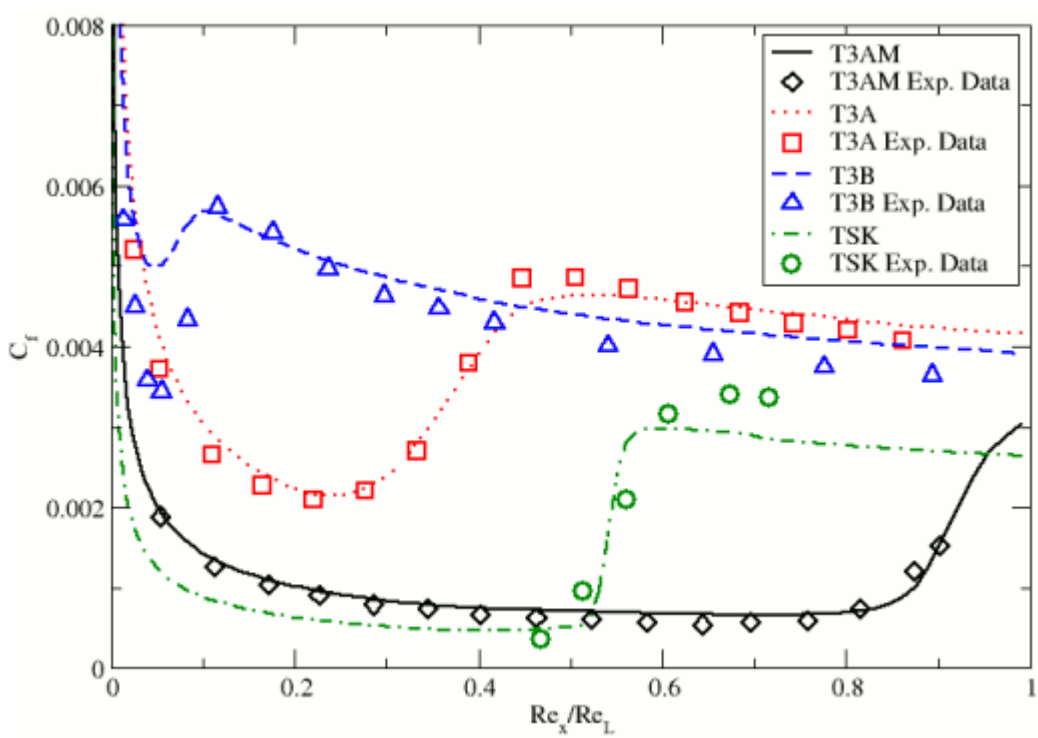
Correlations for ReTheta_c and F_length

The correlations for $Re_{\theta c}$ and F_{length} have been calibrated according to the forms suggested by Suluksna et al. [49]. The correlations are as follows:

$$Re_{\theta c} = \min\left(0.615\overline{Re_{\theta t}} + 61.5, \overline{Re_{\theta t}}\right) \tag{500}$$

$$F_{length} = \min\left[0.5 + \exp\left(7.168 - 0.01173\overline{Re_{\theta t}}\right), 300\right] \tag{501}$$

When combined with the correlation for $Re_{\theta t}$ in Eqn. (481), these correlations enable the Gamma ReTheta transition model to mimic the bypass transition cases T3A, T3B, T3AM [42] and the natural transition cases TSK [48], as shown in the following figure.



Skin friction coefficient predictions for calibrated correlations

Gamma Transition Formulation

The formulation of the Gamma transition model includes the intermittency transport equation and the correlation for the critical Reynolds number.

Contents:

Intermittency Transport Equation

Correlation for ReTheta_c
Cross-Flow Term

Intermittency Transport Equation

The transport equation for the intermittency γ is formulated as:

$$\frac{d}{dt}\int_V \rho\gamma\, dV + \int_A \rho\gamma\left(\mathbf{v}-\mathbf{v}_g\right)\cdot d\mathbf{a} = \int_A \left(\mu + \frac{\mu_t}{\sigma_f}\right)\nabla\gamma\cdot d\mathbf{a} + \int_V \left[P_\gamma - E_\lambda\right] dV \tag{502}$$

For details on how γ enters the SST K-Omega transport equations see Coupling with SST (Menter) K-Omega Model.

Production Term

The transition source term is defined as:

$$P_\gamma = 100\rho S\gamma\left(1-\gamma\right)F_{onset} \tag{503}$$

where:

$$F_{onset1} = \frac{Re_v}{2.2Re_{\theta c}} \tag{504}$$

$$F_{onset2} = \min\left(F_{onset1}, 2\right) \tag{505}$$

$$F_{onset3} = \max\left[1-\left(Re_T/3.5\right)^3, 0\right] \tag{506}$$

$$F_{onset} = \max\left(F_{onset2}-F_{onset3}, 0\right) \tag{507}$$

Re_v is the strain-rate Reynolds number defined as:

$$Re_v = \frac{\rho\, d^2 S}{\mu} \tag{508}$$

and S is the modulus of the mean strain rate tensor:

$$S = |\mathbf{S}| = \sqrt{2\mathbf{S}:\mathbf{S}^T} = \sqrt{2\mathbf{S}:\mathbf{S}} \tag{509}$$

where:

$$\mathbf{S} = \frac{1}{2}\left(\nabla\mathbf{v} + \nabla\mathbf{v}^T\right) \tag{510}$$

The turbulent Reynolds number Re_T is defined as:

$$Re_T = \frac{k}{\nu\omega} \tag{511}$$

Destruction Term

The destruction/relaminarization source is defined as:

$$E_{\lambda} = C_{a2}\rho W\gamma F_{turb}\left(c_{e2}\gamma - 1\right) \tag{512}$$

where:

$$F_{turb} = \exp\left[-\left(Re_T/2\right)^4\right] \tag{513}$$

and:

$$W = \sqrt{2\mathbf{W}:\mathbf{W}^T} \tag{514}$$

where the vorticity tensor $\mathbf{W}$ is defined as:

$$\mathbf{W} = \frac{1}{2}\left(\nabla\mathbf{v} + \nabla\mathbf{v}^T\right) \tag{515}$$

Model Coefficients

The model coefficients for the intermittency transport equation are defined as:

$C_{a2} = 0.06,\ \ C_{e2} = 50,\ \ \sigma_f = 1.0$

Correlation for ReTheta_c

The correlation for the critical Reynolds number $Re_{\theta c}$ is defined as follows:

$$Re_{\theta c} = 100 + 1000\exp\left[-Tu_L F_{PG}\left(\lambda_{\theta L}\right)\right] \tag{516}$$

where the arguments Tu_L and $\lambda_{\theta L}$ are approximated locally as:

$$Tu_L = \min\left(100\frac{\sqrt{2k/3}}{\omega d}, 100\right) \tag{517}$$

and

$$\lambda_{\theta L} = \min\left[\max\left(-7.57\cdot 10^{-3}\frac{d^2}{\nu}\nabla(\mathbf{n}\cdot\mathbf{v})\cdot\mathbf{n} + 0.0128,\ -1.0\right), 1.0\right] \tag{518}$$

where:

- k is the turbulent kinetic energy.
- ω is the specific dissipation rate.
- d is the distance to the nearest wall.
- $\mathbf{n}$ is the wall-normal unit vector.

The function $F_{PG}\left(\lambda_{\theta L}\right)$ is calculated as:

$$F_{PG}(\lambda_{\theta L}) = \begin{cases} \max[\min(1 + 14.68\lambda_{\theta L}, 1.5), 0] \ , & \lambda_{\theta L} \geq 0 \\ \max[\min(1 - 7.34\lambda_{\theta L} + , 3), 0] \ , & \lambda_{\theta L} < 0 \end{cases} \tag{519}$$

where a limiter is applied to avoid negative values.

Cross-Flow Term

The cross-flow term allows you to account for cross-flow effects in three dimensional boundary layer transition.

Activating the cross-flow term modifies the transition trigger function F_{onset}, which activates the production term in the intermittency transport equation given by Eqn. (502).

The modified transition trigger function is given by:

$$F_{onset} \equiv \max(F_{onset}, F_{onset,SCF}) \tag{520}$$

where F_{onset} is given by Eqn. (507) and $F_{onset,SCF}$ is defined as:

$$F_{onset,SCF} = \min[\max(0, f_{SCF} - 1), 1] \tag{521}$$

The cross-flow function f_{SCF} is defined as:

$$f_{SCF} = \frac{C_{SCF} C_r \Delta H_{SCF} \mathrm{Re}_{\theta_c}}{(\mathrm{Re}^*_{\delta 2t})_{tr}} \tag{522}$$

where:

- C_{SCF} is a Model Coefficient.
- C_r is the Surface Roughness Constant.
- Re_{θ_c} is given by Eqn. (516).
- $(\mathrm{Re}^*_{\delta 2t})_{tr}$ is the Critical Reynolds Number.

Critical Reynolds Number

According to Arnal's C1-criterion [40], cross-flow triggers transition as long as the following condition is satisfied:

$$\frac{\mathrm{Re}^*_{\delta 2t}}{(\mathrm{Re}^*_{\delta 2t})_{tr}} \geq 1 \tag{523}$$

where:

- $\mathrm{Re}^*_{\delta 2t}$ is the cross-flow displacement thickness Reynolds number.
- $(\mathrm{Re}^*_{\delta 2t})_{tr}$ is the critical Reynolds number, beyond which the cross-flow triggers transition in the boundary layer.

The critical Reynolds number is defined as a function of a boundary layer shape factor as:

$$\left(\mathrm{Re}^{*}_{\delta 2t}\right)_{tr} = \begin{cases} -\dfrac{300}{\pi}\arctan\left[\dfrac{0.106}{(H-2.3)^{2.05}}\right] & if \quad 2.3 \le H < 2.7 \\ 150 & if \quad H < 2.3 \end{cases} \tag{524}$$

where the shape factor H is derived from the pressure gradient parameter $\lambda_{\theta L}$ given by Eqn. (518) as [50]:

$$H = 2 + 4.14k - 83.5k^2 + 854k^3 - 3337k^4 + 4576k^5 \tag{525}$$

with:

$$k = 0.25 - \lambda_{\theta L} \tag{526}$$

The cross-flow displacement thickness Reynolds number is a non-local quantity that needs integration across the boundary layer. This term is proportional to the three-dimensionality of the boundary layer and approximated using helicity. Helicity as a cross-flow measure was reported by Langtry et al. [44] and Mueller et al. [47].

The streamwise helicity Ω_s is defined as:

$$\Omega_s = \left|\frac{\mathbf{v}}{|\mathbf{v}|} \cdot \mathbf{w}\right| \tag{527}$$

where:

- $\mathbf{v}$ is the velocity vector.
- $\mathbf{w} = \nabla \times \mathbf{v}$ is the vorticity vector.

Following Langtry et al. [44], a dimensionless cross-flow strength is defined from the streamwise helicity as:

$$H_{SCF} = \frac{\Omega_s d}{|\mathbf{v}|} \tag{528}$$

where:

- d is the wall distance.
- $|\mathbf{v}|$ is the magnitude of the local velocity vector.

The cross-flow displacement thickness Reynolds number is finally approximated as:

$$\mathrm{Re}^{*}_{\delta 2t} \approx C_{SCF}\Delta H_{SCF}\mathrm{Re}_{\theta_c} \tag{529}$$

where:

$$\Delta H_{SCF} = H_{SCF}\left[1 + \min\left(\frac{\nu_t}{\nu}, 0.4\right)\right] \tag{530}$$

Surface Roughness Constant

Langtry et al. [44] reported a log dependence between the stationary cross-flow Reynolds number and the surface roughness and supplemented the model to include this correlation. A similar approach is used to mimic the roughness effect into the cross-flow model by adding a roughness constant C_r, which is defined as:

$$C_r = 2 - 0.5^{\frac{h\,(m)}{0.25 \times 10^{-6}\,m}} \tag{531}$$

where $h\,(m)$ is the cross-flow inducing roughness height (in meters) and a Model Coefficient.

Model Coefficients

C_{SCF}	$h\,(m)$
1	3×10^{-6}

Coupling with SST (Menter) K-Omega Model

The Gamma ReTheta transition model and the Gamma transition model enter the transport equation for the turbulent kinetic energy k (see SST (Menter) K-Omega Model) as follows:

$$\frac{\partial}{\partial t}(\rho k) + \nabla\cdot\left[\rho k\left(\overline{\mathbf{v}} - \mathbf{v}_g\right)\right] = \nabla\cdot\left[\left(\mu + \sigma_k \mu_t\right)\nabla k\right] + P_k - \gamma'\rho\beta^* f_{\beta^*}\left(\omega k - \omega_0 k_0\right) + S_k \tag{532}$$

where the production term P_k is defined as:

$$P_k = \gamma_{trans}\left(G_k + G_{nl} + G_b\right) + G_k^{\lim} \tag{533}$$

and:

$$\gamma' = \min\left[\max\left(\gamma_{trans}, 0.1\right), 1\right] \tag{534}$$

where:

$$\gamma_{trans} = \begin{cases} \gamma_{eff} & \text{for the Gamma ReTheta transition model} \\ \gamma & \text{for the Gamma transition model} \end{cases} \tag{535}$$

γ_{eff} and γ are provided by the respective transition model (see Gamma ReTheta Transition Formulation and Gamma Transition Formulation).

γ_{trans} and γ' are unity if no transition model is activated.

The Gamma transition model introduces an additional production term $G_k^{\lim}$ to ensure proper generation of k at transition points for low values of free stream turbulence:

$$G_k^{\lim} = 5\max\left(\gamma - 0.2, 0\right)\left(1 - \gamma\right)F_{on}^{\lim}\max\left(3\mu - \mu_t, 0\right)SW \tag{536}$$

with:

$$F_{on}^{\lim} = \min\left(\max\left(\frac{Re_v}{2.2 \cdot 1100} - 1, 0\right), 3\right) \tag{537}$$

S is the modulus of the mean strain rate tensor that is defined in Eqn. (1744). W is the modulus of the rotation rate tensor given by Eqn. (1746):

Re_v is the strain-rate Reynolds number defined in Eqn. (508).

The additional production term is set to zero if the Gamma transition model is not activated.

Transition Formulation Nomenclature

The following is a guide to the terms used in this section.

Scalars

d	distance to the nearest wall
k	turbulent kinetic energy
t	time, time scale
P_γ	intermittency production
E_γ	intermittency destruction
F_{length}	function governing transition “length”
I	turbulence intensity (expressed as a fraction)
Tu	turbulence intensity (expressed as a percentage)
Re_θ	momentum thickness Reynolds number
$Re_{\theta c}$	critical momentum thickness Reynolds number
$Re_{\theta t}$	transition momentum thickness Reynolds number
$\overline{Re_{\theta t}}$	transported transition momentum thickness Reynolds number
Re_T	turbulent Reynolds number
Re_v	strain-rate Reynolds number
Re_ω	dissipation Reynolds number
K	acceleration parameter
L	local grid length
S	strain rate tensor modulus
W	vorticity tensor modulus
U	velocity magnitude
V	cell volume
γ	intermittency
γ_{eff}	effective intermittency
ρ	density

λ_θ	Thwaite's pressure gradient parameter
θ	boundary layer momentum thickness
μ	dynamic viscosity
μ_t	turbulent viscosity
ν	kinematic viscosity
σ_γ	turbulent Schmidt number
$\sigma_{\theta t}$	inverse turbulent Schmidt number
ω	specific dissipation rate

Tensors or Column Vectors

S	strain rate tensor
W	vorticity tensor

Cartesian Vectors

a	face area vector
n	wall-normal unit vector
u	unit velocity vector
v	velocity
$\mathbf{v}_g$	grid velocity
w	vorticity vector

Field Operators

- Gradient of a scalar quantity:

  $$\nabla\phi = \frac{\partial\phi}{\partial x}\,\mathbf{i} + \frac{\partial\phi}{\partial y}\,\mathbf{j} + \frac{\partial\phi}{\partial z}\,\mathbf{k}$$

- Divergence of vector quantity:

  $$\nabla\cdot\mathbf{v} = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z}$$

- Curl of vector quantity:

  $$\nabla\times\mathbf{v} = \left(\frac{\partial v_z}{\partial y} - \frac{\partial v_v}{\partial z}, \frac{\partial v_x}{\partial z} - \frac{\partial v_z}{\partial x}, \frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y}\right)$$

- Scalar (inner, dot) product of two vectors, whose result is a scalar:

  $$\mathbf{a}\cdot\mathbf{b} = a_i b_i$$

- Scalar (inner, double-dot) product of two tensors, whose result is a scalar:

  $$\mathbf{S}{:}\mathbf{T} = S_{ij}T_{ji}$$

Transition Bibliography

[40] *Arnal, D., Habiballah, M., and Coustols, C., "Laminar instability theory and transition criteria in two- and three-dimensional flows", La Recherche Aerospatiale, 1984-2, 1984.*

[41] *Abu-Ghannam, B. J. and Shaw, R. "Natural Transition of Boundary Layers --- The Effects of Turbulence, Pressure Gradient, and Flow History," J. Mech. Eng. Sci., Vol. 22, No. 5, pp213-228, 1980.*

[42] *ERCOFTAC (European Research Community on Flow, Turbulence, Combustion) Nexus. [online database], URL: http://ercoftac.mech.surrey.ac.uk/.*

[43] *Langtry, R. B. "A Correlation-Based Transition Model Using Local Variables for Unstructured Parallelized CFD Codes," Doctoral Thesis, University of Stuttgart, 2006.*

[44] *Langtry, R. B., Sengupta K., Yeh, D. T., and Dorgan, A. J. "Extending the* $\gamma - Re_{\theta t}$ *Local Correlation based Transition Model for Crossflow Effects," AIAA Fluid Dynamics Conference 2015, June 22-26, Dallas, TX, USA.*

[45] *Menter, F. R., Langtry, R. B., Likki, S. R., Suzen, Y. B., Huang, P. G., and Volker, S. 2004. "A Correlation-Based Transition Model Using Local Variables Part 1 --- Model Formulation," Proc. ASME Turbo Expo, June 14-17, Vienna, Austria.*

[46] *Menter, F. R., Smirnov, P. E., Liu, T., and Avancha, R. "A One-Equation Local Correlation-Based Transition Model," Flow, Turbulence and Combustion, DOI 10.1007/s10494-015-9622-4, Springer, published online July 2015*

[47] *Mueller, C. and Herbst, F. "Modelling of Crossfow-Induced Transition Based on Local Variables," 6th European Conference on Computational Fluid Dynamics (ECFD), 2014.*

[48] *Schubauer, G.B. and Klebanoff, P.S. "Contributions on the Mechanics of Boundary Layer Transition," NACA Technical Note 3489, 1955.*

[49] *Suluksna, K., Dechaumphai, P., and Juntasaro, E. "Correlations for Modeling Transitional Boundary Layers under Influences of Freestream Turbulence and Pressure Gradient," Int. J. Heat and Fluid Flow, in press 2008.*

[50] *White, F.M. "Viscous Fluid Flow", McGraw-Hill, 1991, Singapore*

Modeling Wall Distance

Wall distance is a parameter that represents the distance from a cell centroid to the nearest wall face with a non-slip boundary condition. Various physical models require this parameter to account for near-wall effects.

There are two models available in STAR-CCM+ to calculate the wall distance:

- **Exact Wall Distance**

  The Exact Wall Distance model makes an exact projection calculation in real space, which is based on a triangulation of the surface mesh. The use of K-D search trees or SFC/SIMD algorithms accelerates the calculation.
- **PDE Wall Distance**

  The PDE Wall Distance model makes a close estimate of the distance that is based on a Partial Differential Equation (PDE) calculation, the Poisson equation.

Note: The **Exact Wall Distance** model is selected on your behalf after the selection of a specific physical model that requires the wall distance parameter. To select the **PDE Wall Distance** model, you need to explicitly deselect the **Exact Wall Distance** model in the physics continuum. A *Wall Distance* group box is then presented for the selection.

The PDE wall distance calculation is faster, but solving for the exact wall distance is more accurate.

The wall distance calculation takes place during the initialization step of the simulation. Information on the progress of this calculation is shown in the progress bar and status bar beneath the STAR-CCM+ explorer window.

Since the wall distance is a mesh parameter, it does not clear automatically when the solution is cleared. To clear the wall distance, select **Reset Mesh** in the **Solutions** > **Clear Solutions** menu.

Contents:

Wall Distance Models Reference
Poisson Formulation

Wall Distance Models Reference

The **Wall Distance** models allow you to compute the wall distance d for the relevant turbulence and wall treatment models.

Models Overview

<table>
<tr><td rowspan="2">Model Names and Abbreviations</td><td colspan="4">Exact Wall Distance*</td><td>E WD</td></tr>
<tr><td colspan="4">PDE Wall Distance</td><td>PDE WD</td></tr>
<tr><td>Theory</td><td colspan="5">See Poisson Formulation (PDE WD).</td></tr>
<tr><td>Provided By</td><td colspan="5">[physics continuum] > Models > Wall Distance</td></tr>
<tr><td>Example Node Path</td><td colspan="5">Continua > Physics 1 > Models > Exact Wall Distance</td></tr>
<tr><td rowspan="9">Requires</td><td colspan="5">Material: one of Gas, Liquid, Eulerian Multiphase, Multi-Component Gas, Multi-Component Liquid
Flow: any
Viscous Regime: Turbulent</td></tr>
<tr><td rowspan="8">Turbulence:</td><td>Detached Eddy Simulation</td><td colspan="3">Detached Eddy Simulation: any</td></tr>
<tr><td>Large Eddy Simulation</td><td colspan="3">Subgrid Scale Turbulence: any</td></tr>
<tr><td rowspan="6">Reynolds-Averaged Navier-Stokes</td><td rowspan="6">Reynolds Averaged Turbulence:</td><td>K-Epsilon Turbulence</td><td>K-Epsilon Turbulence Models: any</td></tr>
<tr><td>K-Omega Turbulence</td><td>K-Omega Trubulence Models: any</td></tr>
<tr><td>Reynolds Stress Turbulence</td><td>Reynolds Stress Turbulence Models: any</td></tr>
<tr><td>Spalart-Allmaras Turbulence</td><td>Rans Spalart-Allmaras Turbulence: any</td></tr>
<tr><td rowspan="3">Activates</td><td>Reference Values</td><td colspan="4">Minimum Allowable Wall Distance. See Reference Values.</td></tr>
<tr><td>Solvers</td><td colspan="4">Wall Distance (E WD), see Wall Distance Solver Properties
PDE Wall Distance (PDE WD), see PDE Wall Distance Solver Properties</td></tr>
<tr><td>Field Functions</td><td colspan="4">Wall Distance
See Field Functions.</td></tr>
</table>

Reference Values

Minimum Allowable Wall Distance

The wall distance algorithm has been developed to return a positive definite value for almost all cell shapes. However it is possible that some degenerate cells, such as boundary sliver cells, could return a value that is close to zero. Since turbulence models and wall functions require a non-zero value, this could cause numerical problems. For this reason, the minimum allowable wall distance limits the value of the wall distance in the continuum. The default value is 1.0E-6 m.

If the wall distance is limited in one or more cells, a warning message is displayed. In this case, investigate the cause and attempt to remedy the situation.

Wall Distance Solver Properties

This solver controls the wall distance computation for the Exact Wall Distance model in all the continua for which the model is activated.

Minimum Tree Size Threshold	Controls the size of the search tree that is used by the wall distance solver. By default, the search tree can contain 500,000 data points. Increasing this number can speed up the wall distance calculation at the expense of consuming additional memory. For a parallel simulation, the maximum tree size is computed as the higher of the following numbers: • Number of cells per processor • Minimum tree size threshold By forcing the tree size to be at least the value set by the minimum tree size threshold, STAR-CCM+ limits the number of times that the wall distance solver rebuilds the search tree.
Solver Frozen	When **On**, prevents the solver from updating the solution while iterating or responding to other solvers. This feature is primarily a “debugging” tool and is not supported for normal simulation work. Deactivated by default.
Use SFC/SIMD	When activated, a Space-Filling Curve search algorithm is used for the turbulence wall distance calculation. This property is deactivated by default. Activating this option usually speeds up the wall distance calculation while maintaining a similar memory consumption. Be aware that for cases with low core count an increase in peak memory can occur.
Verbosity	Allows you to choose how much information STAR-CCM+ reports in the Output window while the simulation is running. This property is useful for debugging problems when they occur. The only valid entries are 0 (default), 1, and 2: • 0 = None: Leaves output at its default. • 1 = Low: Displays CPU time statistics across cores for calculating the wall distance. It reports the mean, maximum, and minimum elapsed times in seconds.

	• 2 = High: Displays CPU time statistics (see low verbosity) and memory usage statistics across cores. This entry allows you to evaluate the memory consumption of all server processes that are relevant to the wall distance calculation: ◦ max process peak—that is the maximum resident memory usage across cores. ◦ max host peak—that is the maximum memory that is used on a node. ◦ accumulated peak—that indicates the rough scalability in terms of required memory.

PDE Wall Distance Solver Properties

This solver controls the wall distance computation for the PDE Wall Distance model in all the continua for which the model is activated.

Maximum Iteration

Sets the maximum number of iterations the solver can use. The default is 10.

Solver Tolerance

Specifies the total relative solver residual. The default value is 0.01.

Under-Relaxation Factor

In order to promote convergence, this property is used to under-relax changes of the solution during the iterative process. If residuals show solution divergence and do not decrease, reduce the under-relaxation factor.

Verbosity

Allows you to choose how much information STAR-CCM+ reports in the Output window while the simulation is running. This property is useful for debugging problems when they occur. The only valid entries are 0 (default) and 1:

- 0 = None: Leaves output at its default.
- 1 = Low: Displays convergence history for wall distance and CPU time information.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Under-Relaxation Factor Ramp

The **PDE Wall Distance** > **Under-Relaxation Factor Ramp** node allows you to set and control a linear ramp for under-relaxation. For details, see Under-Relaxation Factor Ramp Reference.

AMG Linear Solver

The **PDE Wall Distance** > **AMG Linear Solver** node allows you to set the algebraic multigrid parameters for the turbulence solver. For details, see Algebraic Multigrid and AMG Linear Solver Properties.

Field Functions

Wall Distance

Scalar field that represents the normal distance y from the cell centroid to the nearest wall face with a non-slip boundary condition calculated by the Wall Distance solver.

Poisson Formulation

The PDE Wall Distance solver uses a Poisson-type pure diffusion equation for a distance-related variable ϕ.

This equation is a generalized 3D extension of an exact one-dimensional intuitive formulation proposed by Spalding [39]:

$$\nabla^2\phi = -1 \tag{538}$$

Boundary conditions:

- $\phi = 0$ at the wall boundary (Dirichlet boundary condition).
- $\nabla\phi \cdot \mathbf{n} = 0$ at all other boundaries, where $\mathbf{n}$ is the normal vector for the boundary (Neumann boundary condition).

The wall distance d is then computed from this equation:

$$d = -\left|\nabla\phi\right| + \sqrt{|\nabla\phi|^2 + 2\phi} \tag{539}$$

Modeling Radiation

The Radiation Model is the gateway, or entry point to all of the radiation modeling capabilities of STAR-CCM+. This section describes STAR-CCM+'s radiation modeling.

To solve a radiation problem:

- Select the Radiation model.
- Select one of the radiative transfer models.
- Select one of the radiation spectrum models.

Reference information is provided for field functions that the radiation models use.

Radiative Transfer Models

The radiative transfer model defines the overall solution method for the governing Radiative Transfer Equation (RTE), including how the RTE is discretized with respect to space and solid angle. Available transfer models are:

- Surface-to-Surface (S2S) radiative heat transfer for modeling non-participating media radiation.
- Participating Media Radiation (the Discrete Ordinates Method, or DOM).

Radiation Spectrum Models

The radiation spectrum model defines how the radiation wavelength spectrum is considered (and potentially discretized) within the context of the transfer model. The radiation spectrum models are:

- Gray Thermal Radiation for modeling wavelength-independent radiation properties.
- Multiband Thermal Radiation for modeling wavelength-dependent radiation properties.

For the Gray Thermal spectrum model, the full thermal wavelength domain is considered as a whole, and all radiative properties are considered invariant within this single spectrum. For the Multiband Thermal spectrum model, the full-thermal spectrum can be divided into subspectra or spectral bands. For each subspectrum, waveband-specific properties can be defined. During the radiation solution process, the transfer model is applied to each subspectrum. The total radiative transfer is the sum of the resulting solutions from the individual subspectra.

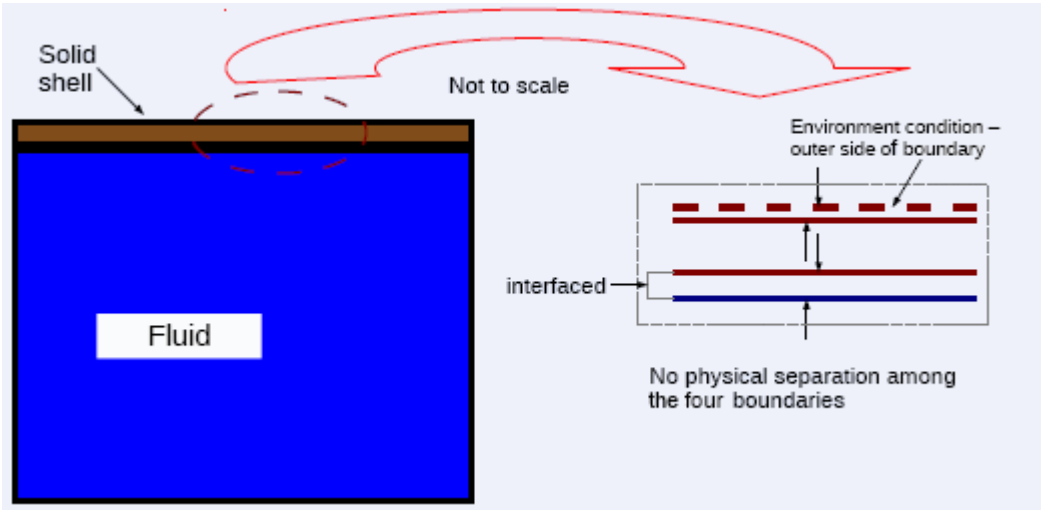
For the S2S transfer model, solar loads can additionally be considered with both the Gray and Multiband Thermal spectrum models by selecting the optional Solar Loads model.

Surface-to-surface modeling also requires view factors. These view factors are calculated using the View Factors model.

Radiation with Shell Regions

The model of radiation with solid shells is consistent with the general approach to radiation in solids, and is compatible with solar radiation, and with gray and multi-band radiation modeling. Each shell boundary has individually specified radiative properties (emissivity, reflectivity, transmissivity, and specularity).

The example given in the following figure shows a shell bounding a fluid, such as a thin-walled pipe containing liquid, applicable to engine and heating simulations. A shell can also stand alone, without containing fluid, for example as a mirror in a lighting simulation.



Contents:

Radiation Characterization
Radiation Model Workflow
Modeling Surface-to-Surface Radiation
Modeling Participating Media Radiation
The Gray Thermal Radiation Model
The Multiband Thermal Radiation Model
Setting Thermal Radiation Model Parameters
Modeling Solar Radiation Loads
Radiation Field Functions Reference

Radiation Characterization

Radiation is characterized in terms of the following:

- Radiation Spectrum
- Spatial Domain for Solution of Radiation Transfer
- Radiation Properties

Radiation Spectrum

Within the full electromagnetic spectrum, the thermal radiation spectrum - approximately 0.1 to 100 μm - is pertinent to heat transfer. This thermal spectrum includes a portion of the ultra-violet (UV) range and all of the visible and infra-red (IR) ranges. The thermal radiation that a given object emits generally has a wide distribution of wavelengths. Relatively hot objects, such as the sun, emit thermal radiation over the entire thermal spectrum. Relatively cooler objects (T < 800K) emit predominantly in the infra-red range.

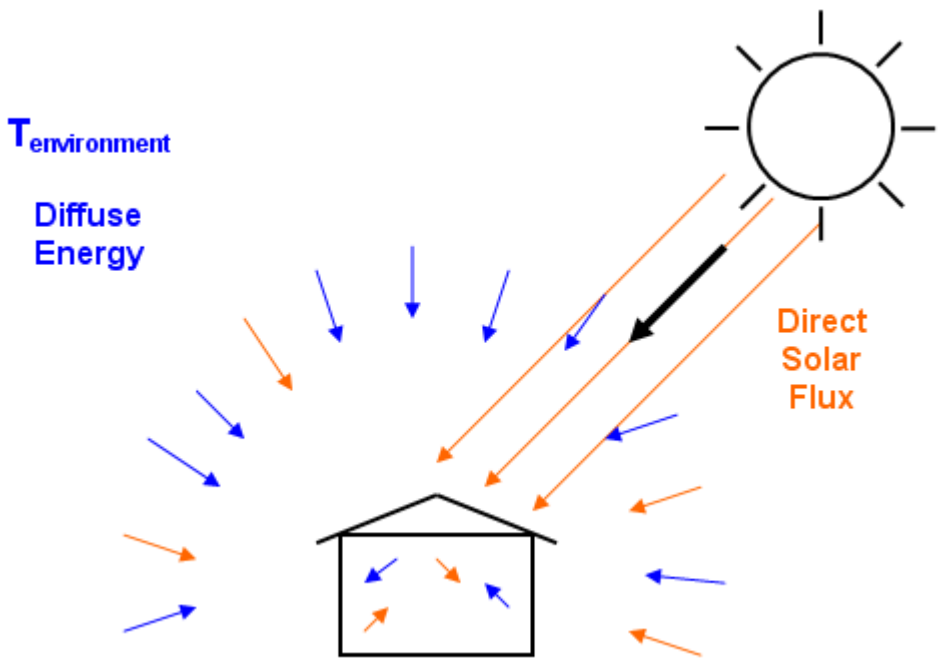
When radiation properties are invariant with wavelength, the radiation is said to be gray. Gray radiation is modeled using the Gray Thermal Radiation model. The radiation properties are taken to be same over the entire thermal spectrum, and only a single radiative transfer solution is necessary for the full thermal spectrum.

When the radiation properties vary with wavelength, the radiation is said to be spectral. Spectral radiation is modeled using the Multiband Thermal Radiation model. Different radiation properties can be specified for each individual spectral band, and radiative transfer is considered individually in each band. The total transfer is the sum of the contributions from each individual band.

Spatial Domain for Solution of Radiation Transfer

The effects of radiation are modeled within the regions in the simulation as well as between the regions. Radiation exchange between regions takes place whenever the interface between the regions is transparent or partially transparent to radiation.

In addition, radiation exchange with the surrounding environment can be modeled with the Surface-to-Surface transfer model (see schematic below). Both terrestrial and solar emissions can be considered. The Planck distribution governs the distributions of the solar and terrestrial emissions over the thermal spectrum. Solar emissions span the entire thermal radiation spectrum including the UV, visible, and IR ranges.



Schematic Representation of Radiation

Thermal radiative exchange with the terrestrial environment includes only diffuse effects. The terrestrial environment surrounding the regions emits diffusively as a blackbody at the environment radiation temperature. It is also possible to have thermal radiative exchange with the solar environment.

Radiation Properties

The definition of a radiation problem is completed by defining the properties of the media inside the regions and the surface properties of the boundaries. Non-participating media are transparent to radiation. The only requirement is that the surface properties of emissivity, reflectivity, and transmissivity are specified on the boundaries. In contrast, participating media can emit, absorb, and scatter radiation. These media additionally require you to specify the material absorption and scattering properties within the regions using the continuum material properties.

Radiation Model Workflow

Use the following steps to set up either of the Radiation models, S2S or Participating Media (DOM):

1. Select the S2S Radiation or Participating Media Radiation (DOM) model.
2. For the DOM model, select a single- or multi-component material model from the *Material* group.
3. Select the Gray Thermal Radiation or Multiband Thermal Radiation model from the *Radiation Spectrum* group.
   a) For the Multiband Thermal Radiation model, set up the Spectral Bands.
4. For the S2S model, identify the regions of your model which use radiation and deactivate radiation in a region if radiation transfer is not relevant in that region:
   a) In each region, select the **Physics Conditions** > **Radiation Transfer Option** node.
   b) Turn **off** the *Internal Radiation Transfer* property in regions that are opaque.
5. For the S2S model, to set up objects to interact through a meshless space, use the Environment condition.
6. For the DOM model, set radiative material properties of the participating medium:
   a) Set the order of the *Ordinate Sets* property to specify the desired degree of accuracy in simulating the participating media radiation effect. See Participating Media Radiation (DOM) Properties.
   b) Set the radiative properties of the medium lying between the radiating boundaries. See Participating Media Radiation (DOM) Material Properties.
   c) Set the radiative properties of the boundary surfaces. These properties are defined in the same way as for surface-to-surface models. See Setting Thermal Radiation Model Parameters.
7. Select a Flow model for the participating medium:
   - Select *Flow*: **Segregated Flow**, then select *Energy*: **Segregated Fluid Temperature**, **Segregated Fluid Isothermal**, or **Segregated Fluid Enthalpy**.

     or
   - Select *Flow*: **Coupled Flow**, then select *Optional Models*: **Coupled Energy**.
8. If two regions of same continuum are interfaced, place either a baffle or a porous baffle between them where both interface boundaries are actively participating in radiation. If the interface does not participate, use an internal interface.
9. To produce a semi-transparent interface, make sure that the same thermal model is used on both sides—both Gray or both Multiband. An interface boundary automatically becomes opaque when any one side has no radiation model or has the **Radiation Transfer Option** off.
10. Set thermal radiation parameters on boundaries:
    - Emissivity
    - Reflectivity
    - Transmissivity
    - Radiation temperature on open boundaries such as pressure outlets
    - Reflection specularity
11. Identify the boundaries of your model where diffuse radiation enters and activate diffuse radiation at boundaries where it is needed:

a) At each boundary, select the **Physics Conditions** > **Radiation Flux Option** node.

b) Set the *Radiation Flux Option* property to **Diffuse radiation flux**.

c) Select the **Physics Values** > **Diffuse Radiation Flux** node.

d) Set the *Method* and *Value* properties. The flux is specified as a scalar profile.

   If you are using the Multiband Thermal model and the **Diffuse Radiation Flux** node has *Method* set to **Composite**, there is a node for each band under **Diffuse Radiation Flux**. Set the physics values for each band.

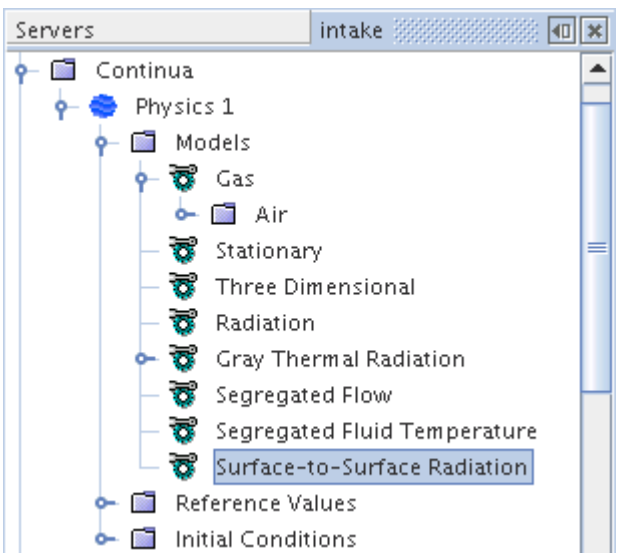
12. Set **Thermal Environment** > Radiation Temperature to a single value for all continua.

Modeling Surface-to-Surface Radiation

S2S (Surface-to-Surface) Radiation simulations concerns only the radiating and absorbing surfaces, not any intervening medium.

The medium that fills the space between the surfaces is non-participating. That is, it does not absorb, emit, or scatter radiation. Under these circumstances, the radiation properties and the thermal boundary conditions that are imposed on each surface uniquely define the amount of radiation that a surface receives and emits. The surface properties are quantified in terms of emissivity, reflectivity, transmissivity, and radiation temperature. These properties are not dependent on direction; however, they can be dependent on radiation wavelength with the Multiband Radiation model.

The corresponding node (**Surface-to-Surface Radiation** in the following screenshot) is activated when you select an S2S simulation.



Note: The S2S Radiation model can apply to a multi-part object. Inputs must be specified per-boundary.

Note: An environment boundary condition makes it possible to model both convection and radiation heat transfer between the external sides of domain boundaries and the environment.

Note: An S2S simulation can have regions connected by mixing plane interfaces. Mixing planes are non-participating boundaries and so, like transparent boundaries, do not take part in emission, absorption, or reflection. However, unlike the radiation passing through a fully transparent boundary or interface, radiation passing through a mixing plane gets radially averaged coming out on the other side. Use of the mixing plane interface is not compatible with the Solar Loads model or the Circumferential Flux Averaging model.

Patches and View Factors

The S2S Radiation model relies on:

- A spatial discretization of the boundary surfaces into patches.
- View factors that quantify, for each patch, the proportion of surface area that the other patches illuminate.

Note: View factors are calculated from non-diffuse surfaces to other partially diffuse surfaces, but not the other way round.

Patches and view factors are explained in further detail in the Theory Guide—S2S (Surface-to-Surface) Radiation section, and are generated using the View Factors model.

View Factor Use

The S2S Radiation model operates in two steps:

1. Calculate view factors using ray tracing.
2. Apply view factors to compute radiosity and irradiation fields on all surfaces.

In the view factor calculation step, deterministic ray tracing is applied to compute the factors. Rays are traced into the hemisphere above each patch using a fixed distribution of directions over this hemisphere. By default, a distribution of 1024 rays per patch is used.

The view factors must be calculated only when the geometry and radiative properties such as transmissivity and specular reflectivity change. Therefore, they generally can be calculated once and then reused during the combined-mode heat transfer solution to update the radiative fluxes from the surface emissive powers (that is temperatures).

An advantage of the two-step approach is computational efficiency gained from doing relative infrequent ray tracing. The trade-off is that view factors must be stored, often requiring considerable amount of memory.

Impact of Patch and Angular Resolution

View factors are considered computationally efficient for combined-mode problems. However, the number of degrees of freedom (that is, the number of patches) is often limited. The limitation is because the view factors count can be O(N2) in the limit, where N is the number of patches. The STAR-CCM+ approach provides flexibility through patch and angular resolution to help address the computation time and memory issues that are associated with view factors. This approach makes it is possible to run cases having a million patches or so.

With the deterministic ray tracing approach in STAR-CCM+, the number of rays per patch limits the view factor count. Therefore, with reciprocity you can get a maximum of 2L view factors per patch, where L is the number of rays per patch. Thus, 1 million patches with 1024 rays/patch could produce a maximum of about 2 billion view factors. This number is manageable in STAR-CCM+ in terms of both computation time and memory with parallel ray tracing and S2S algorithms. The resolution of the ray-tracing can be increased to increase accuracy, but at the expense of computation time and memory. There is thus a classic trade-off between accuracy and required computational resources (time and memory).

For some combined-mode problems, the resolution of the underlying mesh for the advection/diffusion transport can be prohibitively high when using its boundary faces directly as patches in the S2S calculations. For instance, some simulations can have tens of millions of boundary faces, making a 1-to-1 ratio between the S2S patches and the boundary faces intractable. Additionally, this fine patch resolution is not always needed with respect to accuracy, especially when the surface topology and surface temperatures do not change rapidly. STAR-CCM+ therefore provides the flexibility to use a patch resolution that is coarser than the boundary face resolution. Patches can be formed, for instance, from aggregating the boundary faces from the

underlying mesh. For high-resolution meshes, you can generally use 1 patch for every 10 or so faces from the underlying mesh. There is again a trade-off between accuracy and required computational resources. A finer patch resolution generally provides better accuracy, but also requires more computational resources. You can find a balance that provides the needed accuracy within the given time and hardware constraints. In the end, most methods for solving the radiative transfer equations are constrained in a similar trade-off between performance and accuracy due to the nature of the physics (transfer at a distance).

Comparison with the Monte Carlo Method

The Monte Carlo method, which is not implemented in STAR-CCM+, can be applied in two ways to solve the radiative transfer equations (RTE): to compute view factors in a two-step approach, or to solve the RTE equation directly.

When the Monte Carlo method is used to compute view factors, the directions for rays leaving a patch are selected randomly (non-deterministically). Because the Monte Carlo method is a statistical approach, it generally requires a high number of rays to obtain a statistically meaningful result. In contrast, STAR-CCM+ uses a deterministic approach where the directions are based on a fixed quadrature. Here, the number of traced rays determine the accuracy. The regular angular distribution of rays in the deterministic approach provides flexibility to vary angular resolution while still maintaining symmetry properties at modest resolutions. In the limit of an infinite number of rays, both methods give the same set of view factors.

When the Monte Carlo method is used to solve the RTE directly, it is invoked every time a solution of the RTE is required. For a combined-mode heat transfer analysis, the solution is normally required once every global iteration and is often prohibitively expensive for most practical problems. Therefore, the use of view factors that are calculated from either a deterministic or Monte Carlo approach is an advantage with respect to computation time: the ray tracing is done only once at the beginning of the simulation (assuming a static geometry).

Contents:

The S2S Radiation Model
The View Factors Calculator Model
Setting the View Factors Calculator Model Parameters
Generating Patches and View Factors
Working with View Factor Data
Post-Processing S2S Radiation

The S2S Radiation Model

The S2S (Surface-to-Surface) Radiation model makes it possible to simulate thermal radiation exchange between diffuse surfaces forming a closed set.

Theory	See Theory Guide—S2S (Surface-to-Surface) Radiation.
Provided By	**[physics continuum]** > **Models** > *Radiation*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Surface-to-Surface Radiation**
Requires	*Space*: **Three Dimensional**

	Optional Models: **Radiation**	
Activates	Physics Models	*View Factors* *Radiation Spectrum* *Solar Radiation*
	Boundary Inputs	See S2S Radiation Boundary Settings.
	Region Inputs	See S2S Radiation Region Settings.
	Solvers	See S2S Solvers.
	Field Functions	**Boundary Irradiation**, **Boundary Irradiation on External Side**, **Boundary Mean Radiant Temperature**, **Boundary Radiation Heat Flux**, **Boundary Radiation Heat Flux on External Side**, **Boundary Radiosity**, **Boundary Radiosity on External Side**, **Total Emissivity**. See Radiation Field Functions Reference.

S2S Radiation Boundary Settings

When a simulation uses standard overset mesh with coincident wall boundaries, these boundaries are thermally uncoupled and so can be inconsistent. Decide if the possible errors at the portion of wall with coincident overlapping wall boundaries contribute significantly to the problem of interest. In many cases, the possible errors have very little impact on the problem being studied (for instance on heat transfer toward the body contained within the overset region). If significant error is possible, match the boundary conditions by hand.

Inflow, Outflow, Free Stream, and Wall Boundaries

Radiation Flux Option

This option allows you to model diffuse radiation sources at boundaries according to your own specification. See Diffuse Radiation Flux.

Radiation Flux Option	Corresponding Physics Value Nodes
No flux	None
Diffuse radiation flux	**Diffuse Radiation Flux** The diffuse radiation flux is specified as a scalar profile. Requires the Multiband Thermal Radiation or Gray Thermal Radiation model.

Specularity Option

Models reflection specularity, the specular fraction of the surface reflectivity. See Specular Reflectivity.

Specularity Option	Corresponding Physics Value Nodes
Diffuse reflection	None

Adjustable reflection specularity	**Reflection Specularity** The specularity is specified as a scalar profile.

Inflow, Outflow, and Free Stream

Radiation Temperature

The temperature of an open boundary that has a total radiant emittance identical to that of a black body. See Radiation Temperature.

S2S Radiation Region Settings

Radiation Transfer Option

Specifies that a region uses S2S radiation. Either a baffle or porous baffle must be placed between any two regions where only one region has this option activated. This baffle ensures that rays do not pass into regions where S2S is deactivated from regions where it is activated.

Radiation Transfer Option Properties

Radiation Transfer Option

When **On**, activates S2S transfer in the region. This property is **On** by default.

S2S Surface Orientation (Deprecated)

The S2S Surface Orientation Option is a deprecated option for modeling radiation between two opaque solids. Use the Environment condition instead.

The S2S Surface option makes provision for the special situation where you want to model radiation exchange between two opaque *solids* that do not have a mesh in the void between. For such cases, you can set the *S2S Surface Orientation* property of the **S2S Surface Orientation Option** node to **Outward** on both bodies. For all other cases with mesh in between, leave the option at its default setting of **Inward**.

This option is only available for solid regions belonging to continua in which S2S Radiation is active.

S2S Surface Orientation Option Properties

S2S Surface Orientation

Surfaces are directed **Inward** (the default) or **Outward** for S2S calculation.

S2S Solvers

The S2S solver is activated by default, after **Surface-to-Surface Radiation** has been selected in the *Physics Model Selection* dialog and a case geometry has been imported.

The S2S solver has the S2S AMG Linear solver as a sub-solver. See AMG Linear Solver Properties.

Properties

Solver Frozen

When **On**, prevents the solver from updating the solution while iterating or responding to other solvers. This feature is primarily a “debugging” tool and is not supported for normal simulation work.

The View Factors Calculator Model

Selecting the View Factors Calculator model enables solvers for generating radiation patches and calculating view factors.

View Factors Calculator Reference

Theory	See Theory Guide—S2S (Surface-to-Surface) Radiation.	
Provided By	**[physics continuum]** > **Models** > *Viewfactors*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **View Factors Calculator**	
Requires	*Space*: **Three Dimensional** *Optional Models*: **Radiation** *Radiation*: **Surface-to-Surface (S2S) Radiation**	
Activates	Boundary Inputs	**Inflow, Outflow, and Wall Boundaries** **Custom Patch Angularity Specification** See Angularity. **Custom Patch Specification** See Setting the View Factors Calculator Model Parameters.
	Region Inputs	**Patch Angularity** See Angularity. **Patch Specification** See Patching Parameters.
	Solvers	**View Factors Calculator**. See Solver Properties: View Factors Calculator.
	Field Functions	**Radiation Patch Id**, **Radiation patch Id on External Side**, **Voxel partition**. See Radiation Field Functions Reference.

Solver Properties: View Factors Calculator

Patches and view factors are computed automatically at the beginning of the solution if they do not exist. However, they can also be generated manually using creation commands that are located in a pop-up menu that is accessed by right-clicking the **View Factors Calculator** node. The creation commands simply provide added flexibility to generate patches and view factors manually before initiating the solution.

The View Factors Calculator solver can work with overset mesh, but in the case of coincident uncoupled boundaries, both boundaries are patched and the behavior of the solver becomes inconsistent since rays can be terminated at either boundary without preference.

Verbosity

Provides additional output during the calculation of the view factors.

None

No output.

Low

Displays ray tracing progress and reciprocity convergence.

High

Displays additional progress and resource usage information.

Update Frequency

Controls the frequency with which the solver calculates view factors. Use this property only for cases with mesh motion that results from rigid body motion.

Number of Beams

The number of beams affects accuracy, view factor calculation time, and view factor storage requirements. A higher number of beams increases the accuracy of the view factors. However, increasing the number also increases the view factor calculation time and the memory requirements. The default number (1024) is usually a good balance between accuracy and resource use.

Maximum Reciprocity Iterations

Sets the maximum number of iterations in the reciprocity procedure. (Reciprocity is the constraint on view factors that the second part of Eqn. (2096) imposes.)

Reciprocity Tolerance

Specifies the criterion for convergence that is used in the reciprocity procedure. (Reciprocity is the constraint on view factors that the second part of Eqn. (2096) imposes.)

Maximum Polygons per Voxel

Controls the resolution of the voxel mesh that is used in ray tracing. Lowering the value increases the resolution of the voxel mesh. This property generally does not need to be changed from the default value of 50. Increasing the number slightly decreases memory requirements, but can also slightly increase the calculation time.

Solver Frozen

When **On**, prevents the solver from updating the solution while iterating or responding to other solvers. This feature is primarily a “debugging” tool and is not supported for normal simulation work.

View Factors Calculator Solver Pop-Up Menu

The following commands are accessed by right-clicking on the **View Factors Calculator** node.

Generate Radiation Patches

Adds radiation patches to the simulation.

Calculate View Factors

Generates the view factors. If patches do not exist, this command generates them.

Delete View Factors

Deletes all view factors in the simulation.

Display View Factor summary

Writes a summary of the view factor data in the *Output* window.

Export ASCII View Factors

Exports the view factor data to an ASCII file in your current working directory.

Controlling View Factor Recalculation for Rigid Body Motion

In cases where rigid body motion is present, view factors are recalculated every time-step by default. For cases where some reduction in accuracy is acceptable, you can save on computational time by recalculating the view factors less frequently. The property *Update Frequency*, provides this control.

This update frequency is designed to control view factor recalculation only for the case of mesh motion that results from rigid body motions. For all other cases (such as addition or removal of cells, reconstruction of interfaces, and morphing motion), the view factor solver acts as if the update frequency equaled 1. If mesh motion occurs due to rigid body motion, view factors are recalculated based on the input value of the update frequency. Within the update frequency interval, the S2S solver uses old view factors. The maximum allowable update frequency is 100.

The update frequency is related to the time-steps, not the inner iterations for the unsteady solver. This approach represents a compromise between accuracy and the computational cost of view factor recalculation.

Note: Indiscriminate use of this feature can lead to prominent loss of accuracy.

Setting the View Factors Calculator Model Parameters

Selecting the View Factors Calculator model enables both patch generation and view factors calculation.

Patching Parameters

The Patching feature provides you with two control parameters: the number of patches and the angularity.

The control parameters (options and values) are defined by default on a regionwise basis in each region where the View Factors Calculator model is active under the **Physics Conditions** and **Physics Values** nodes.

Patch Specification Properties

Patch Specification

Sets the number of patches specification method.

Patch Specification	Corresponding Value Nodes
Proportion of Faces	**Patch/Face Proportion** Specifies the target for the fraction of patch allocated to each boundary face. The target number patches on a boundary = (**Patch/Face Proportion** /100)* number of faces on boundary. A setting of `25` means that the target is to give each face 25% of a patch, so that 80 faces would be targeted to get 20 patches and each patch covers four faces. The default setting is 100, that is, each gets 100% of a patch—each boundary face has a patch. This setting is also the maximum.
Total	**Number of Patches** Specifies the target for total number of patches using **Number Of Patches** under **Physics Values**. If the value is entered at **Region** > **Physics Values**, the specification method then seeks to allocate patches from the total number in proportion to the number of discretized faces on each boundary (so that a boundary with twice as many faces gets twice as many patches). The total number of patches for a region cannot be less than the total number of boundaries or more than the total number of faces on all boundaries. The total number of patches for a boundary cannot be less than one and cannot be more than the total number of faces on that boundary.

Custom Patch Specification

Specifies whether the number of patches is set to **Use Region Values** or custom values. When you choose either **Proportion of Faces** or **Total**, the corresponding value node appears for the boundary.

The number of patches and patch angularity can be further refined or customized at individual boundaries under the **Physics Conditions** node. Alternatively, you can let these settings be handled automatically for two boundaries by creating an internal, periodic interface between them. It carries the full transmissivity of the originating patch from one parent boundary of the interface to the other.

Angularity

This parameter defines the maximum allowed angle between the normals to contiguous boundary faces for these faces to belong to the same patch. It provides a control over the patch planarity and is used to detect sharp angles in the geometry, such as would happen at a corner, and to prevent patches from bridging such features. The angularity is set at 150 degrees by default and would require changing only in cases where specific geometric features must be captured.

Custom Patch Angularity Specification Properties

Custom Patch Angularity Specification

For patch angularity, specifies whether to **Use Region Values** or custom values on the boundary.

View Factors Parameters

The view factors are calculated by shooting beams from each patch and computing the amount of energy that is deposited at intercepting patches. This amount is a function of the optical properties of the surface.

For diffuse reflection, the view factors calculator only keep track of irradiated and transmitted energy and would formally only require specification of transmissivity at the boundary. However, for completeness, and because the individual spectrum models enforce Kirchhoff's law, for Thermal spectral bands, emissivity, reflectivity, and transmissivity are provided at boundaries.

For transparent or partially transparent external walls, a black body, whose temperature is characterized by an environmental temperature, can absorb the beams which are transmitted out of the enclosure.

These radiative properties of the boundary surfaces, **Surface Emissivity**, **Surface Reflectivity** and **Surface Transmissivity**, are defined in the same way as for thermal models (see Setting Thermal Radiation Model Parameters).

Generating Patches and View Factors

Radiation patches and view factors can be generated automatically or manually.

Patches must be generated before view factors can be calculated. As the validity of the view factors depends on interfaces that were previously created, the **Calculate View Factors** pop-up menu action of the View Factors Calculator solver intersects the interfaces before the actual calculation begins.

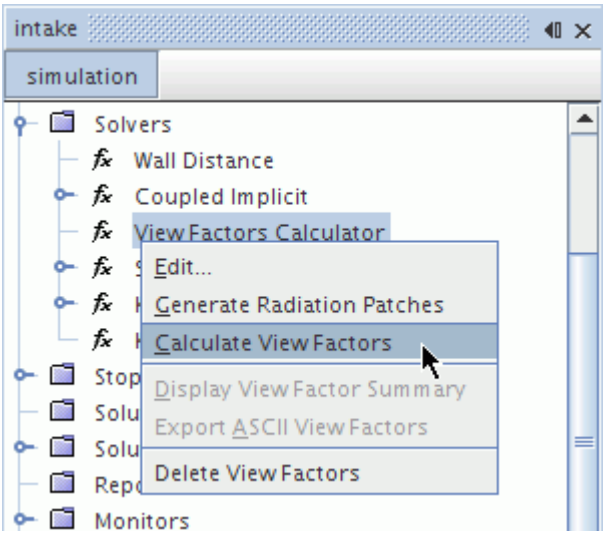
STAR-CCM+ generates radiation patches automatically in the following ways:

- As a result of the **Initialize Solution** command
- After a mesh motion (unless the frequency of recalculation has been changed; see Controlling View Factor Recalculation for Rigid Body Motion)
- After modifications of Patching parameters
- After a boundary type has been changed to or from symmetry or periodic
- After radiation has been activated or deactivated in a continuum

View factors are calculated automatically as a result of the following:

- In the first iteration during the solving process
- After a mesh motion (recalculating)
- After a boundary type has been changed to or from symmetry or periodic
- After radiation has been activated or deactivated in a continuum
- After modifications of transmissivities (recalculating)
- After modifications of the sun position for the Solar Radiation model (recalculating)
- After the number of wavebands has been changed (either when switching from gray to multiband or when changing the number of bands for multiband)

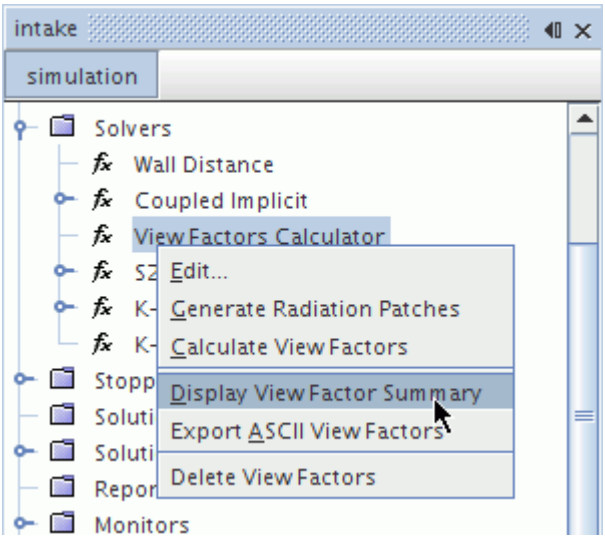
Manual creation is possible after the View Factors Calculator model is selected. The commands **Generate Radiation Patches** and **Calculate View Factors** are available by right-clicking on the View Factors Calculator solver.



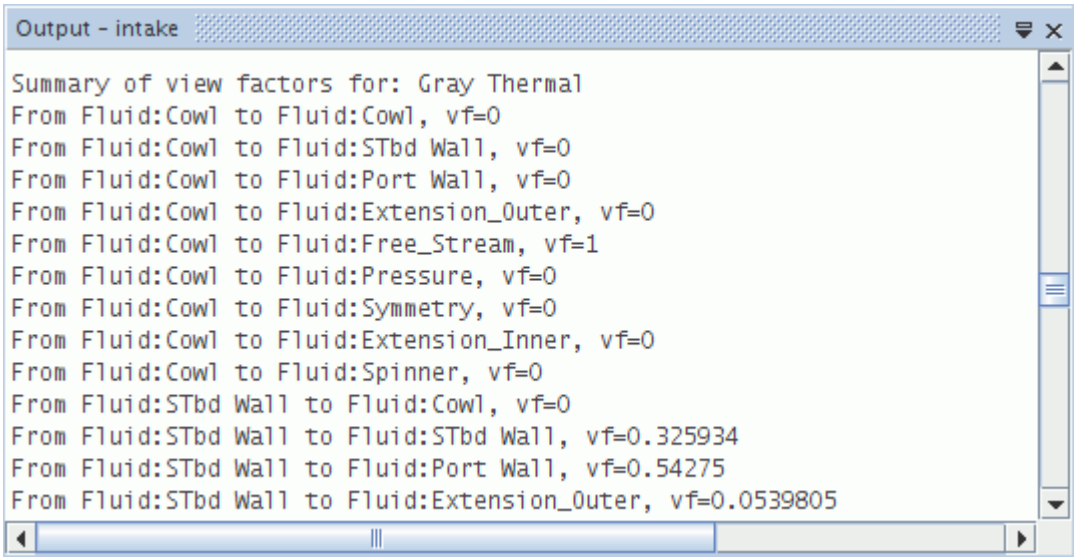
Working with View Factor Data

There are various options for working with view factor data, including displaying, exporting, and deleting.

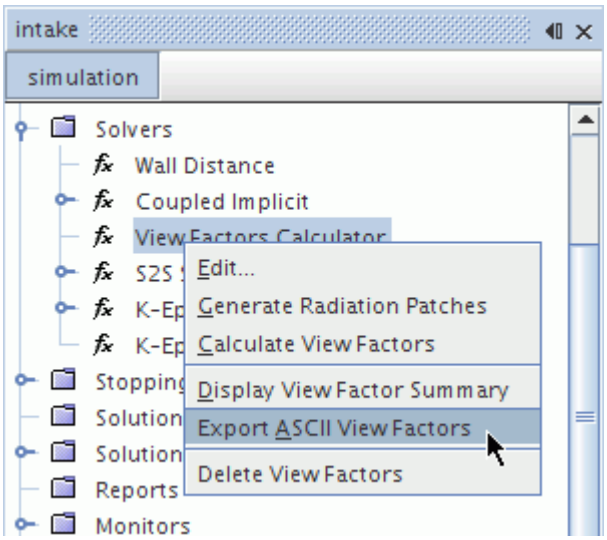
To display view factor data, right-click on the node of the View Factors Calculator solver and select **Display View Factor Summary**.



The data appears in the *Output* window.

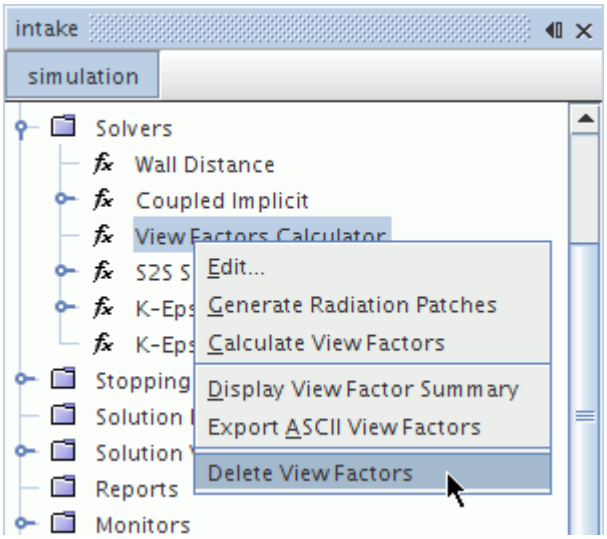


To export the data to ASCII format, select the respective option from the pop-up menu of the solver.



This option generates a file in your current working directory.

To delete the view factor data, select the **Delete View Factors** option.



Post-Processing S2S Radiation

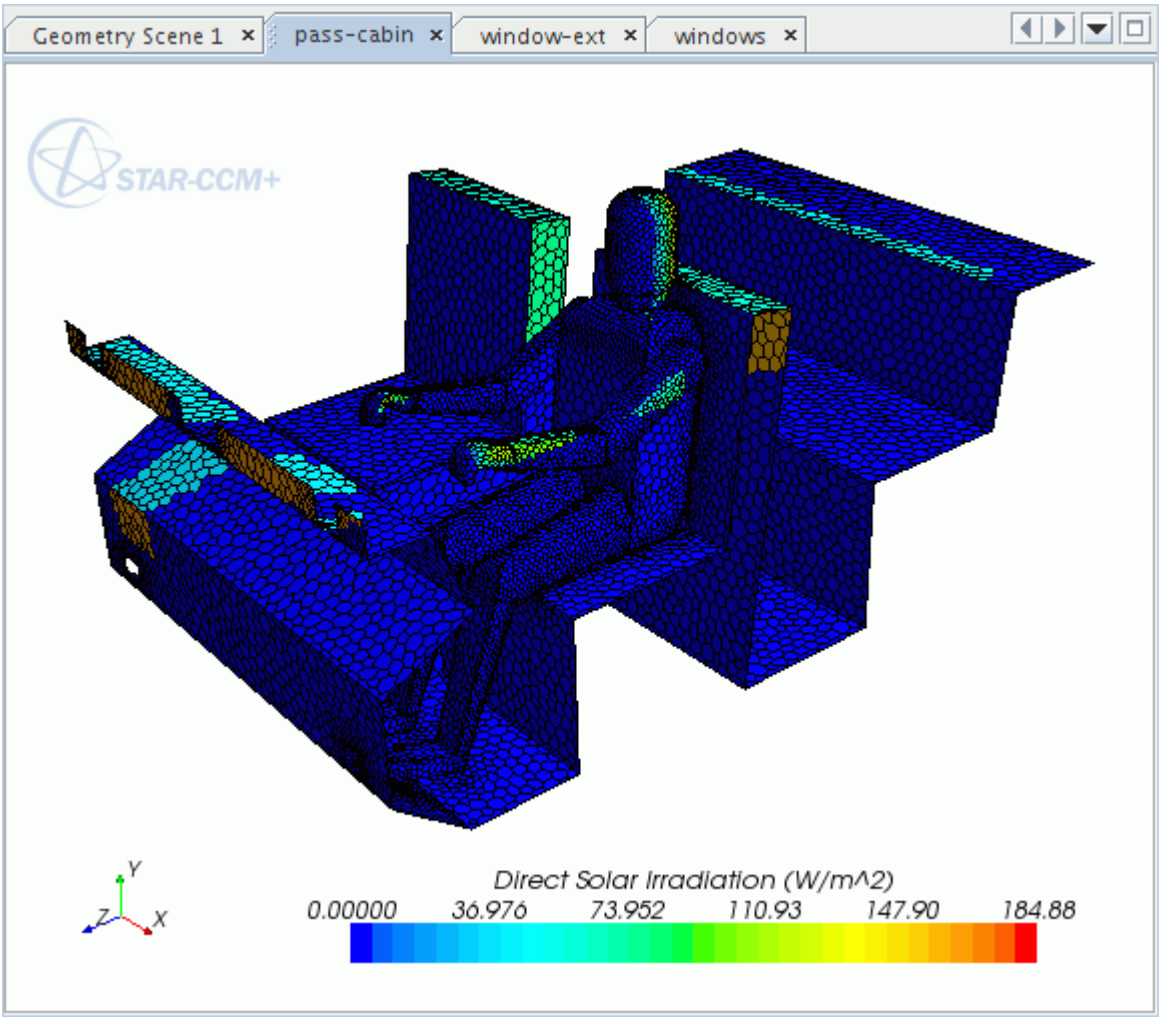
STAR-CCM+ allows you to analyze radiative quantities on transmissive and reflective boundaries.

For boundaries with transmissivity = 1 or specular reflectivity = 1, you can post-process the following quantities:

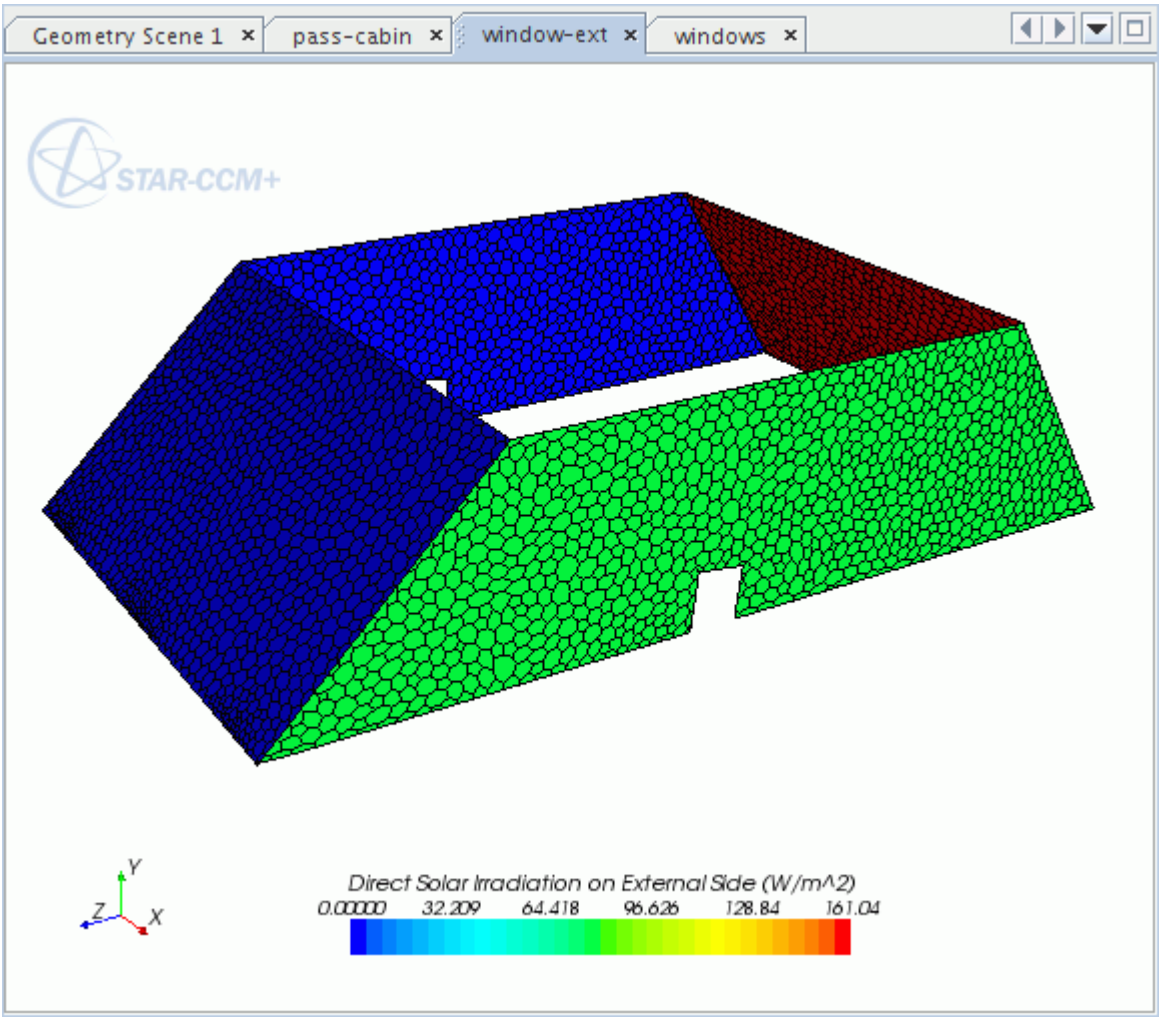
- Irradiation - the radiative heat flux incident on a surface
- Mean Radiant Temperature (MRT)
- Direct solar irradiation - the direct solar flux on the boundary.

For example, consider a thermal comfort case where a car driver experiences solar irradiation:

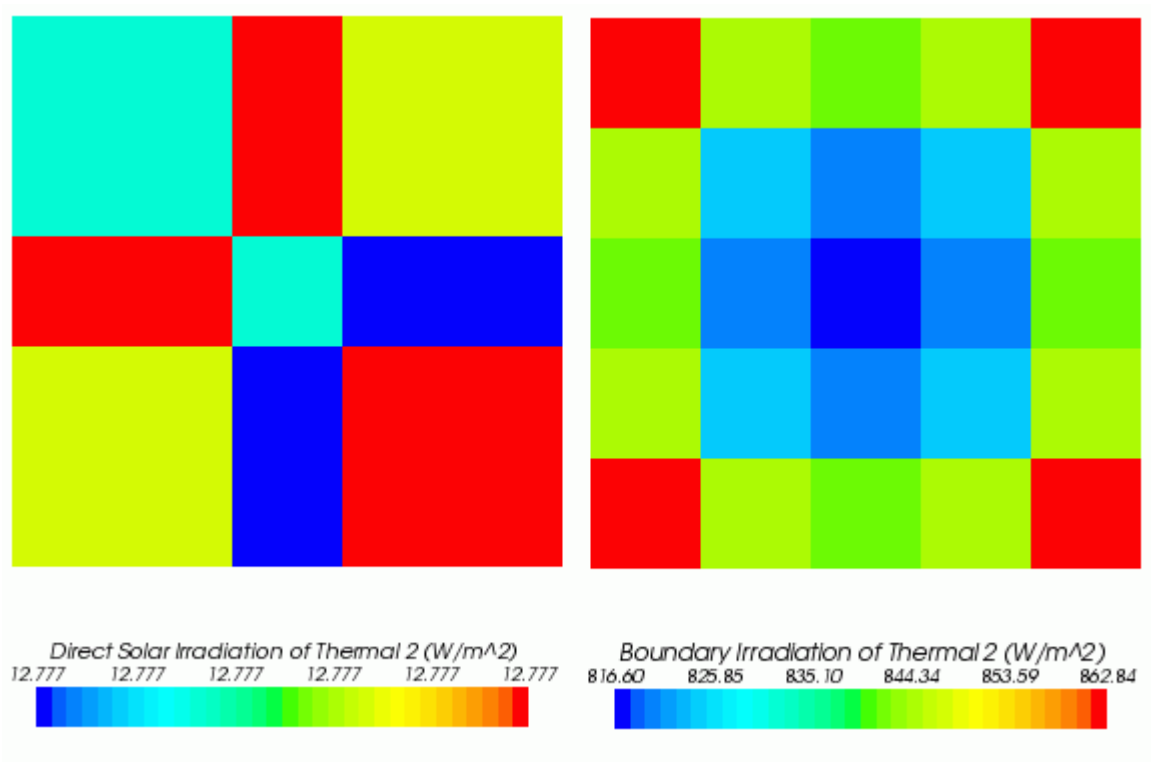
- You can view the solar irradiation falling on the driver.



- You can view the direct solar irradiation on the windows (transmissivity = 1).



Boundaries with specular reflectivity + transmissivity = 1 are called "fully specular boundaries." On fully specular boundaries, the total irradiation accounts for irradiation from other boundaries, solar radiation (direct and diffuse), and radiation from the environment, if any.



Fully specular boundaries do not participate in the solution of S2S and energy equations; zero net radiative flux exists across them.

You can generate a summary of detailed radiation exchange among boundaries, by right-clicking the S2S solver node and selecting **Display Detailed Radiation Exchange**. The summary is printed in the output window.

Example output:

```
Detailed radiation exchange among boundary pairs for Gray Thermal :
    Boundary 1    Boundary 2  Incoming Radiation (W)   Outgoing Emission (W)      Radiative Flux (W)
     fluid:Top
                     fluid:Top                     0                       0                       0
                  fluid:Bottom                 215.8                  2738.6                   -2566
                  fluid:zminus                662.77                   10656                  -10124
                   fluid:zplus                662.76                   10656                  -10126
                  fluid:xminus                662.78                   10655                  -10125
                   fluid:xplus                662.77                   10656                  -10124
                         Total                2866.9                   45360                  -43065
  fluid:Bottom
                     fluid:Top                2773.5                  171.16                  2047.7
                  fluid:Bottom                     0                       0                       0
                  fluid:zminus                217.58                  665.98                   -491.9
                   fluid:zplus                217.58                  665.97                  -491.91
                  fluid:xminus                217.58                  665.97                  -491.91
                   fluid:xplus                217.58                  665.98                  -491.91
                         Total                3643.9                    2835                    80.04
  fluid:zminus
                     fluid:Top                 10794                  86.345                    8548
                  fluid:Bottom                837.27                   86.34                  583.48
                  fluid:zminus                     0                       0                       0
                   fluid:zplus                1131.7                  279.19                  626.21
```

The calculation for this summary does not include boundaries with symmetric or periodic conditions, or fully specular boundaries. These boundaries do not actively participate in the surface-to-surface solution, so there is no radiation heat flux for them.

Note: If view factors or patches have not been calculated or the S2S solution does not exist, detailed exchange post-processing is automatically disabled in GUI.

Modeling Participating Media Radiation

Participating Media radiation, also called DOM because of the use of the Discrete Ordinary Method, passes through media that can absorb, emit, or scatter thermal radiation.

In some cases, thermal radiation can occur only as a surface phenomenon. The media separating the surfaces (such as dry air) are transparent to thermal radiation. Modeling of this type of heat transfer is called Surface-to-Surface (S2S) radiation in STAR-CCM+. However, other applications require consideration of participating media. These media can include particles belonging to a Lagrangian phase, provided the Particle Radiation model is active, or particles such as soot, using the soot emission models.

Modeling this type of heat transfer uses the Participating Media Radiation (DOM) model. The main parameter of the model is the accuracy of the angular representation of space around a given cell. The order of the ordinate set, as selected from the Ordinate Sets drop-down list in the corresponding *Properties* window specifies this accuracy. (See Discrete Ordinate Method Numerical Solution.)

STAR-CCM+ accounts for participating media effects by using the Discrete Ordinate Method (DOM). DOM is a numerical model for describing radiative heat transfer in participating media (see Participating Media Radiation). The method offers itself readily to parallelization in STAR-CCM+. Some of the basic assumptions of this model are shared with the S2S model as follows:

- They both simulate thermal radiation exchange between diffuse/specular surfaces forming a closed set.
- The surface radiative properties are quantified in terms of emissivity, specular and diffuse reflectivity, transmissivity, and radiation temperature. These properties do not depend on direction. However, for the Multiband spectrum model, the surface properties can depend on wavelength.

However, the medium that fills the space between the surfaces can also absorb, emit, or scatter radiation. Therefore, the amount of radiation that each surface receives and emits depends on this effect, as well as on the optical properties of the surface and the thermal boundary conditions that are imposed on it. The Participating Media Radiation model lets you control these phenomena by activating material properties with Participating Media Radiation (DOM) Material Properties.

Contents:

The Participating Media Radiation (DOM) Model
Participating Media Radiation (DOM) Material Properties
Modeling Refraction

The Participating Media Radiation (DOM) Model

Use this model in simulations that include media that can absorb, emit, or scatter thermal radiation.

Theory	Theory Guide—Participating Media Radiation
Provided By	**[physics continuum]** > **Models** > *Radiation*

Example Node Path	**Continua** > **Physics 1** > **Models** > **Participating Media Radiation (DOM)**	
Requires	*Space*: any *Optional Models*: **Radiation**	
Properties	*Ordinate Sets*. See Participating Media Radiation (DOM) Properties.	
Activates	Region Inputs	See Participating Media Radiation (DOM) Boundary Settings.
	Solvers	**DO Radiation** See The DO Radiation Solver.
	Field Functions	**Boundary Irradiation**, **Boundary Radiation Heat Flux**, **Boundary Radiosity**, **Incident Radiation**, **Radiative Absorption**, **Radiative Energy Source**, **Radiative Intensity n**, **Total Absorption Coefficient**, **Total Emissivity**. See Radiation Field Functions Reference.

Participating Media Radiation (DOM) Properties

Ordinate Sets

Sets the order of discrete ordinate method to **n**. For a given choice **Sn**, the number of ordinates is n*(n+2), so increasing the size of the discrete ordinate set significantly increases both accuracy and computational time. The default is **S4**.

Sn quadratures are implemented to provide angular discretization in the Discrete Ordinate Method. See [389]. Sn sets are implemented for n = {2,4,6,8,12,16}. S2 through S8 are based on odd-moment formulation. S12 and S16 are based on even-moment formulation (to avoid negative weights as occur for n > 10 in odd-moment formulation).

Participating Media Radiation (DOM) Boundary Settings

Inflow, Outflow, Free Stream, and Wall Boundaries

Radiation Flux Option

This option allows you to model diffuse radiation sources at boundaries according to your own specification. See Diffuse Radiation Flux.

Radiation Flux Option	Corresponding Physics Value Nodes
No flux	None
Diffuse radiation flux	**Diffuse Radiation Flux** The diffuse radiation flux is specified as a scalar profile. Requires the Multiband Thermal Radiation or Gray Thermal Radiation model.

Specularity Option

Models reflection specularity, the specular fraction of the surface reflectivity. See Specular Reflectivity.

Specularity Option	Corresponding Physics Value Nodes
Diffuse reflection	None
Adjustable reflection specularity	**Reflection Specularity** The specularity is specified as a scalar profile.

Inflow, Outflow, and Free Stream

Radiation Temperature

The temperature of an open boundary that has a total radiant emittance identical to that of a black body. See Radiation Temperature.

The DO Radiation Solver

The DO Radiation solver is activated by default, after **Participating Media Radiation** has been selected in the *Physics Model Selection* dialog and a case geometry has been imported.

To tune the performance of the solver, you can alter a number of the properties that are displayed in the *Properties* window after selecting the **DO Radiation** node.

If you are using overset mesh and the mesh is modified, for instance by mesh motion, the solver updates the DOM solution even if the current step is not at the iteration frequency update point according to the **Update Frequency** setting. Overset mesh in DOM requires no additional boundary set-up, but does require the same continuum on both sides, since it is a region-based indirect interface.

DO Radiation Solver Properties

Under-Relaxation Factor

Applies under-relaxation to the solution of the ordinate equations. The default value (1.0) means that no under-relaxation is used.

Maximum Iteration

Controls the maximum number of internal iterations that are required to solve the ordinate equations. The default value (50) is adequate in most scenarios.

Solver Tolerance

Defines the accuracy that is required in the iterative solution of the ordinate equations. The default value (1.0E-5) is adequate in most scenarios.

Verbosity

Provides extra output for this value while the simulation is running. This property can be useful for debugging problems when they occur.

none	Leaves output at its default.
low	Displays the convergence history for the DOM solution.

high	(Currently displays the same output as **low**.)

Solver Frozen

When **On**, prevents the solver from updating the solution while iterating or responding to other solvers. This feature is primarily a “debugging” tool and is not supported for normal simulation work. The default is **Off**.

Reconstruction Frozen

When **On**, uses reconstruction gradients from the last iteration in which they were updated. *Temporary Storage Retained* must also be on. The default is **Off**.

Reconstruction Zeroed

When **On**, sets reconstruction gradients to zero at the next iteration. Gradients remain zeroed until you turn it **Off** again. The default is **Off**.

Temporary Storage Retained

When **On**, retains temporary storage at the end of the iteration. Examples include the *ap* coefficients, the residual, the corrections, the reconstruction gradients, and the cell gradients. These quantities become available as field functions during subsequent iterations. The default is **Off**.

Setting Algebraic Multigrid Parameters

To set the algebraic multigrid parameters for the DO Radiation solver, open the **DO Radiation** node under the **Solvers** node, then select the node for the **AMG Linear Solver**. The algebraic multigrid parameters can be entered using the *Properties* window.

Participating Media Radiation (DOM) Material Properties

The following values require user input.

Absorption Coefficient

The absorption coefficient is the property of a medium that describes the amount of absorption of thermal radiation per unit path length within the medium (m^{-1}).

This value node (**Absorption Coefficient** in the preceding screenshot) is activated as soon as the Participating Media Radiation model is selected. For the Multiband spectrum model, values of this property can be specified for each spectral band.

The emission coefficient is equal to the absorption coefficient by Kirchhoff’s law.

Weighted Sum of Gray Gases (WSGG)

The **Weighted Sum of Gray Gases** node becomes available under **Absorption Coefficient** node when you have gray radiation as your thermal spectrum specification in a multi-component mixture.

This method calculates radiative properties from local gas species concentrations and local temperatures without you specifying them (see Weighted Sum of Gray Gases Methof (WSGGM)). You specify the path length in the *Optical Path Length* property.

Weighted Sum of Gray Gases Properties

Optical Path Length

Set the optical path length according to the specific case. Where material properties do not vary over distance, you can use Eqn. (2150) to get an estimate.

Scattering Coefficient

The scattering coefficient is the property of a medium that describes the amount of scattering of thermal radiation per unit path length within the medium [m^{-1}].

This value node (**Scattering Coefficient** in the preceding screenshot) is activated as soon as the Participating Media Radiation model is selected. For the Multiband spectrum model, values of this property can be specified for each spectral band.

Refractive Index

Refractive index is the property of a medium that indicates the velocity of light in that medium with respect to the velocity of light in some reference medium or in a vacuum. The "relative refractive index" uses velocity of light in a medium (for example, air) as the reference velocity. The "absolute refractive index" uses the velocity of light in a vacuum as the reference velocity. STAR-CCM+ uses absolute refractive indices. Direct consequences of refractive index include enhanced local emission and change in direction when light crosses the boundary between media with different refractive indices. See Refraction, Reflection, and Critical Angle.

This value node (**Refractive Index** in the preceding screenshot) is activated as soon as the Refraction (Gray) model or Refraction (Multiband) model is selected. For the Multiband spectrum model, values of this property can be specified for each spectral band. With gray radiation, there is a single value to set. With multiband radiation, there is a value for each band. The default value of the index of refraction is 1.0. The index of refraction for the environment is always 1.0. Since refractive index in STAR-CCM+ is always absolute refractive index, the value is strictly $\geq$ 1.0.

Note: Refractive index, absorption and scattering coefficients, boundary emissivity, reflectivity, and transmissivity are all related to each other through the complex index of refraction as used in the electromagnetic theory of radiation. In STAR-CCM+, these variables can be specified independently, so the values must be chosen carefully to obtain a physically meaningful solution.

Modeling Refraction

The Refraction models calculate optical refraction at boundaries between a continuum and the environment and at interfaces between continua.

There are two Refraction models, Refraction (Gray) and Refraction (Multiband), which require the Gray Thermal Radiation model or the Multiband Thermal Radiation model respectively. The thermal radiation models require the Participating Media Radiation (DOM) model.

The Refraction models do not calculate refraction between phase interfaces, nor do they apply to periodic or mapped interfaces.

Use a Refraction model in regions where one or more boundaries are transparent or semi-transparent. Any interface boundary where the other side does not use DOM acts as an external boundary and can be set as transparent or semi-transparent. A fluid film boundary is a possible example. For these transparent and semi-

transparent boundaries, the environment is modeled as a blackbody, an energy sink, into which radiation can escape.

Provided by	**[physics continuum]** > **Models** > *Refraction (Gray)* or *Refraction (Multiband)*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Refraction (Gray)** or **Refraction (Multiband)**	
Requires	*Space*: any selection *Optional Models*: **Radiation** *Radiation*: **Participating Media Radiation (DOM)** *Radiation Spectrum (Participating)*: **Gray Thermal Radiation** or **Multiband Thermal Radiation**	
Activates	Model Controls (child nodes)	**[material model]** > **[material model manager]** > **Material Properties** > **Refractive Index** Example: **Gas** > **Air** > **Material Properties** > **Refractive Index** See Refractive Index
	Field Functions	**Refractive Index** (for gray radiation) or **Refractive Index 0**, **1**, ... (for multiband radiation)

The Gray Thermal Radiation Model

The Gray Thermal radiation model makes it possible to simulate diffuse radiation independently of wavelength. With this model, radiation properties of the media and surrounding surfaces are considered the same for all wavelengths.

Gray Thermal Radiation Model Reference

Theory	See Theory Guide—Radiation Equilibrium.	
Provided By	**[physics continuum]** > **Models** > *Radiation Spectrum*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Gray Thermal Radiation**	
Requires	*Space*: **Three Dimensional** *Time*: any *Optional Models*: **Radiation** *Radiation*: **Surface-to-Surface Radiation** or **Participating Media Radiation (DOM)**	
Properties	*Kirchhoff Enabled* When **On**, enforces Kirchhoff's law (absorptivity equals emissivity) at surfaces. This property must be activated for radiative heat transfer, and can only be deactivated in the special case of using the Surface-to-Surface model to simulate molecular transport by physical analogy. Kirchhoff's law is enforced by requiring the sum of the emissivity, reflectivity, and transmissivity to be 1.0 at surfaces. The default is **On**.	
Activates	Model Controls (child nodes)	**Thermal Environments** > **Radiation Temperature**. See Setting the Thermal Environment.
	Boundary Inputs	See Boundary Settings.
	Field Functions	**Absorption Coefficient**, **Boundary Emissivity**, **Boundary Emissivity on External Side**, **Boundary Reflection Specularity**, **Boundary Reflection Specularity on External Side**, **Boundary Reflectivity**, **Boundary Reflectivity on External Side**, **Boundary Transmissivity**, **Scattering Coefficient**, **User-specified Diffuse Flux**. See Radiation Field Functions Reference.

Boundary Settings

Inflow, Outflow, Free Stream Boundaries, and Wall

Diffuse Radiation Flux

The diffuse radiation flux specified as a scalar profile. Requires the Multiband Thermal Radiation or Gray Thermal Radiation model. See Diffuse Radiation Flux.

Surface Emissivity

The ratio of the power that a body emits to the power it would emit as a black body at the same temperature. See Emissivity.

Surface Reflectivity

The ratio of reflected radiant energy over incident radiant energy at a given surface. See Reflectivity.

Surface Transmissivity

The ratio of transmitted radiant energy over incident radiant energy at a given surface. See Transmissivity.

Setting the Thermal Environment

When the Surface-to-Surface (S2S) Radiation model or the View Factors Calculator model is used, each continuum is required to have a thermal environment. This thermal environment is a simplified representation of the environment surrounding the continuum, from the standpoint of thermal radiation. The Thermal Environment is activated with the Surface-to-Surface (S2S) Radiation model or the View Factors Calculator model. The environment is fully defined by ascribing the desired value to the Radiation Temperature.

The thermal environment is modeled as a black body with unity emissivity, and hence can be characterized solely with the Radiation Temperature. This temperature defines the energy that is effectively radiated from the environment. However, since there has to be only one environment, the value of the Radiation Temperature in every continuum must be the same.

The *Radiation Temperature* is a scalar quantity which is selected by opening the **Thermal Environment** node within the Gray Thermal Radiation model. The temperature can then be set in the *Properties* window.

The Multiband Thermal Radiation Model

The Multiband Thermal radiation model makes it possible to simulate diffuse and specular radiation that is dependent on wavelength or frequency. With this model, wavelength bands specify radiation properties of the media and surrounding surfaces.

The number and extents of bands are chosen to describe property variations with respect to wavelength. During the solution process, the governing radiation equations are solved for each defined band using the specified properties. The total radiation solution is then obtained by integrating (summing) the solutions from each of the individual bands. Because the governing equations are solved for each band, the radiation solution time generally increases linearly with the number of bands. As a consequence, a trade-off often exists between accuracy and computational time.

Absorption is a phenomenon which describes energy being removed from the incident radiative energy by the material or boundary. Absorptivity is a property which is the ratio of the radiative energy that is absorbed to the total radiative energy incident on a boundary. For a surface, the absorptivity dictates absorption whereas for participating media, the absorption coefficient dictates absorption. From Kirchhoff's Law, the spectral directional emissivity is equal to the spectral directional absorptivity. In STAR-CCM+, directional properties are not used so Kirchhoff's Law is implemented as spectral emissivity is equal to the spectral absorptivity. So for each band in a multiband model the emissivity equals the absorptivity. For a gray model, the properties are independent of both direction and wavelength and emissivity equals absorptivity for the entire spectrum.

The bands for the Multiband model are defined using the Spectral Bands sub-node.

Multiband Thermal Radiation Model Reference

Theory	See Theory Guide—Radiation Equilibrium.	
Provided By	**[physics continuum]** > **Models** > *Radiation Spectrum*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Multiband Thermal Radiation**	
Requires	*Space*: **Three Dimensional** *Time*: any *Optional Models*: **Radiation** *Radiation*: **Surface-to-Surface Radiation** or **Participating Media Radiation (DOM)**	
Properties	*Kirchhoff Enabled* When **On**, enforces Kirchhoff's law (absorptivity equals emissivity) at surfaces. This property must be activated for radiative heat transfer, and can only be deactivated in the special case of using the Surface-to-Surface model to simulate molecular transport by physical analogy. Kirchhoff's law is enforced by requiring the sum of the emissivity, reflectivity, and transmissivity to be 1.0 at surfaces. The default is **On**.	
Activates	Model Controls (child nodes)	**Thermal Environments** > **Radiation Temperature**. See Setting the Thermal Environment. Spectral Bands

	Boundary Inputs	See Boundary Settings.

Boundary Settings

Inflow, Outflow, Free Stream Boundaries, and Wall

Diffuse Radiation Flux

The diffuse radiation flux specified as a scalar profile. Requires the Multiband Thermal Radiation or Gray Thermal Radiation model. See Diffuse Radiation Flux.

Surface Emissivity

The ratio of the power that a body emits to the power it would emit as a black body at the same temperature. See Emissivity.

Surface Reflectivity

The ratio of reflected radiant energy over incident radiant energy at a given surface. See Reflectivity.

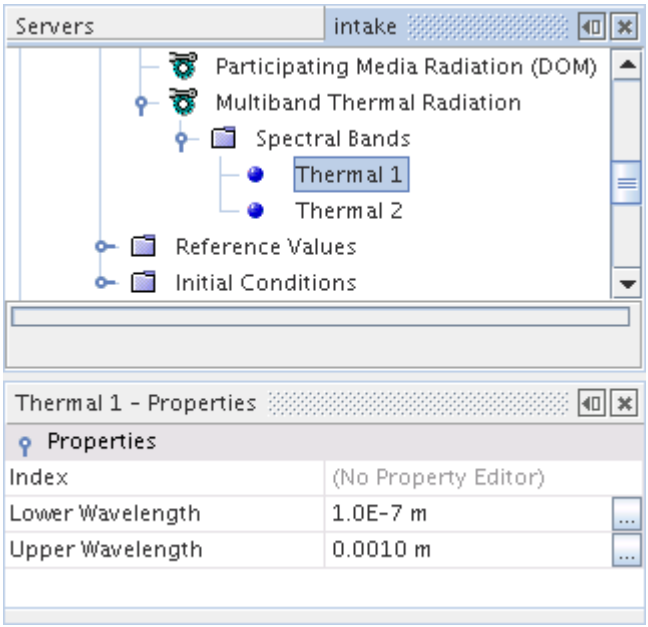
Surface Transmissivity

The ratio of transmitted radiant energy over incident radiant energy at a given surface. See Transmissivity.

Spectral Bands

When Multiband Thermal Radiation is activated, spectral bands are used to describe the discretization of the radiation spectrum. The spectrum is divided into a number of discrete bands, each defined by a lower and an upper wavelength. Surface and media properties can be different in each band. The number of bands and their extents must be chosen to represent variations in radiation properties with respect to wavelength. However, as mentioned before, radiation solution time increases linearly with the number of bands so a trade-off often exists between accuracy and runtime.

The **Spectral Band** node is located under the Multiband Thermal Radiation node. Individual bands are added to this manager node by right-clicking the node and then by selecting **New**. After a new band is created, its lower and upper wavelength extents can be specified under the band properties.



The rules for defining bands are:

- For an individual band, the lower wavelength must be smaller than the upper wavelength.
- Bands cannot overlap one another.

  In general, the bands must be contiguous and span the relevant part of the wavelength domain for the characteristic temperatures of the analysis. (See Planck's distribution in a standard radiation textbook.) The relevant domain for thermal radiation is typically located between 0.1-1000 micro meters, but it can be narrower depending on the characteristic temperatures.
- If radiation is activated in more than one continuum, the set of bands in each continuum must be the same. This rule can be met by defining the spectral bands in one continuum and then copying the bands to any other continua.

Radiation Properties on Boundaries

When solving fluid flows in the presence of thermal radiation, STAR-CCM+ requires additional input that characterizes the radiation properties of boundary surfaces.

Kirchhoff's law requires that radiation properties sum up to one on every surface where they are required. Kirchhoff's law is tied to the radiative energy conservation which is in turn tied to the overall energy conservation, so the law much be enforced for virtually all applications. The *Kirchhoff Enabled* setting of the Gray Thermal Radiation model is on by default.

The radiation properties are defined in the **Regions** node, for each boundary surface. Boundaries are classified either as *Closed*, *Open*, or *Inactive*. As summarized below, *Closed* boundaries require specification of emissivity, ε, reflectivity, ρ, and transmissivity, τ. In addition, *Open* boundaries require specification of the radiation temperature, T_{rad}. The radiation does not directly see *Inactive* boundaries, so they do not require any further specifications.

	Boundary Type			Specify			
Boundary Condition	Closed	Open	Inactive	ε	ρ	τ	T_{rad}
Inlet: mass flow, stagnation, velocity		X		X	X	X	X
Outlet: pressure, flow-split		X		X	X	X	X
Wall	X			X	X	X	
Symmetry plane			X				
Cyclic			X				
Baffle	X			X	X	X	

To specify ε, ρ, τ, and T_{rad}:

- Open the **Boundaries** node.

- For each of the boundary sub-nodes, specify appropriate **Physics Values** for the relevant optical properties as shown in the following screenshot. However, for the Discrete Ordinate Method (DOM), τ can only be specified on baffles.

The emissivity and reflectivity of the interfaces must be specified at each interface boundary, not the original boundary (for example **top-solid1 [In-place 1]**, not **top-solid1**). The latter becomes defunct when the interface is created. These properties must be set in regions on each side of the interface.

In addition, the interface transmissivity is specified on the interface itself instead of the two associated interface boundaries. While emissivity and reflectivity can be different on each side of the interface, the transmissivity must be the same. It is thus specified at one location, on the interface itself. In order to satisfy both the second law of thermodynamics and the reciprocity constraint, the transmissivity must be the same for both sides.

With the Participating Media (DOM) model, semi-transmissive external boundaries only interact with the surrounding thermal environment, and not with other objects separated by a space. (The S2S model handles interaction across separating spaces.) If a meshless region, however thin, separates two semi-transparent objects when Participating Media is used for radiation modeling, radiation exchange happens within each object and between each object and the environment, but not between the two objects through the meshless space.

At wall boundaries with the Surface-to-Surface (S2S) model, an environment condition can also be applied. This condition effectively acts as an interface with the environment and permits:

- Absorption, reflection, and emission from both sides.
- Radiative transmission and conduction through the wall.
- Separate wall temperatures for the inner and outer sides.

Radiative surface properties must be specified for both sides of the boundary when the environment condition is activated.

Emissivity

Emissivity is the ratio of the power that a body emits to the power it would emit as a black body at the same temperature (dimensionless).

The value node **Surface Emissivity** is activated as soon as a thermal radiation model is selected. It is entered as a scalar profile. For the Multiband spectrum model, values of this property can be specified for each spectral band.

Reflectivity

Reflectivity is the ratio of reflected radiant energy over incident radiant energy at a given surface (dimensionless).

The value node **Surface Reflectivity** is activated as soon as a radiation model is selected.

It can be entered as a scalar profile or calculated automatically using the Auto-Calculate method. The Auto-Calculate method is the easiest and preferred approach. When this method is selected, the reflectivity is calculated automatically from the specified emissivity and transmissivity using Kirchhoff's law (reflectivity = 1 - emissivity - transmissivity). Anytime afterwards, when you change the emissivity and/or transmissivity, the reflectivity is automatically recalculated. If the reflectivity is entered using a scalar profile, you are responsible

for ensuring Kirchhoff's law is satisfied. The only time the Auto-Calculate method must not be used is in the special situation where Kirchhoff's law is explicitly deactivated (see The Gray Thermal Radiation Model).

For the Multiband spectrum model, values of this property can be specified for each spectral band.

Transmissivity

Transmissivity is the ratio of transmitted radiant energy over incident radiant energy at a given surface (dimensionless).

The value node **Surface Transmissivity** is entered as a scalar profile. For the Multiband spectrum model, values of this property can be specified for each spectral band.

It is activated as soon as a radiation model is selected. Transmissivity can be specified at inlet, outlet, and wall boundaries.

Transmissivity must be same on both sides of an interface. As a result, transmissivity appears under the **Physics Values** node of each interface node instead of under the **Physics Values** nodes of the associated interface boundaries. For transmissivity to appear under the interface, continua on both sides of the interface must have the same radiation model and spectrum model (gray or multiband) activated.

If radiation or spectrum models do not match on both sides of an interface, that interface does not have any transmissivity input and is treated as opaque. The only exception is the boundary of a volume region when the Fluid Film and Surface-to-Surface Radiation models are selected. In that case, the boundary on the volume region side of the fluid film interface has transmissivity input instead of the interface. Radiation is not active on the fluid film region side.

Transmissivity becomes available in both interface boundaries when a Baffle or Porous Baffle interface is created, provided a radiative transfer model (Surface-to-Surface Radiation or Participating Media Radiation) is selected in the regions on both sides of the interface.

Radiation Temperature

Radiation temperature is the temperature of an open boundary that has a total radiant emittance identical to that of a black body radiator (K).

The value node **Radiation Temperature** is activated as soon as a radiation model is selected. It is entered as a scalar profile.

Specular Reflectivity

Surface reflectivity can be partially or fully specular (as opposed to diffuse).

Specular reflectivity is activated by selecting **Adjustable reflection specularity** in the *Specularity option* property of the **Physics Conditions** > **Specularity Option** node for the boundary. This setting adds the **Reflection Specularity** node to the physics values of the boundary.

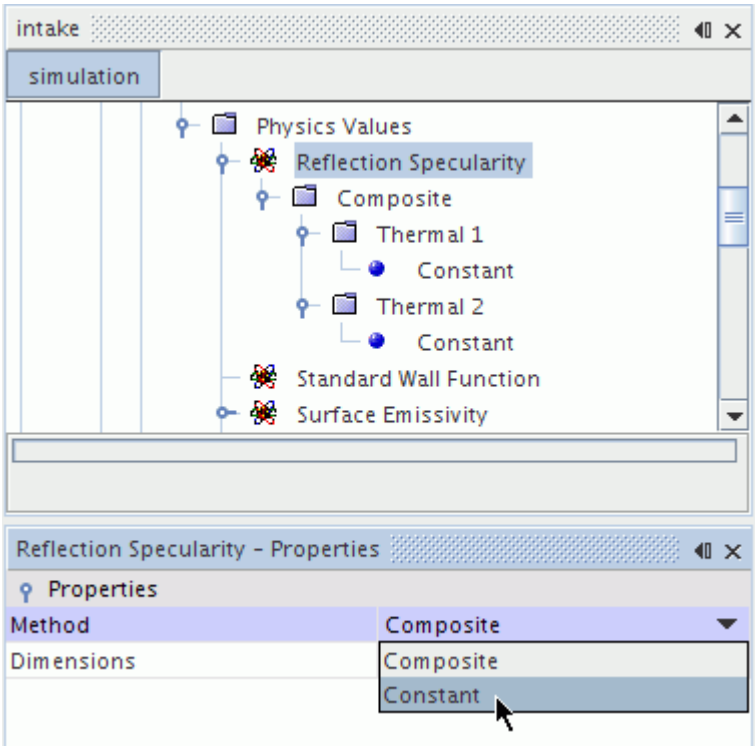
If the boundary is a wall and **Physics Conditions** > **Thermal Specifications** is set to **Environment**, the **Adjustable reflection specularity** setting adds the **External Reflection Specularity** node under **Physics Values**.

Reflection Specularity

Reflection specularity is the specular fraction of the surface reflectivity**:**

- A value of 0 denotes that all of the reflection is diffuse.
- A value of 1 denotes that all of the reflection is specular.
- A value in between denotes a combination of diffuse and specular reflection.

Reflection specularity is set in the **Physics Values** > **Reflection Specularity** node. For the Gray Thermal Radiation model, only one method can be specified for the entire model in the *Method* property. For the Multiband spectrum model, you can select either **Composite** or **Constant** in the *Method* property. Selecting the **Composite** option allows you to set a separate method for each spectral band within the **Composite** folder.



Specular reflectivity is then determined from the product of reflection specularity and surface reflectivity.

A special case is that of a mirrored surface, which is represented by setting the surface reflectivity and the reflection specularity to 1. This setting means that the surface is completely reflective, and all of the reflection is specular. In this case, the specular reflectivity is also equal to 1. Alternatively, specular reflective surfaces can also be partially transmissive (such as plate glass, particularly low-emissivity glass) and/or absorptive (such as high-gloss-painted surface). In such cases, the surface reflectivity is less than 1, with surface transmissivity and/or surface emissivity making up the difference to satisfy Kirchhoff's law. If part of the reflection is diffuse, the reflection specularity can be set to 1 or less. Specular reflectivity here would thus be less than 1.

Diffuse Radiation Flux

This is diffuse non-Planck radiation unrelated to the temperature of the boundary, although this source can affect the thermal behavior of the far field boundaries and the system being modeled. The source can be either uniform over the spectrum (Gray Thermal Radiation model) or spectrally variant (Multiband Thermal

Radiation model). See the theory section for more information. Examples include illumination from LED bulbs or from daylight.

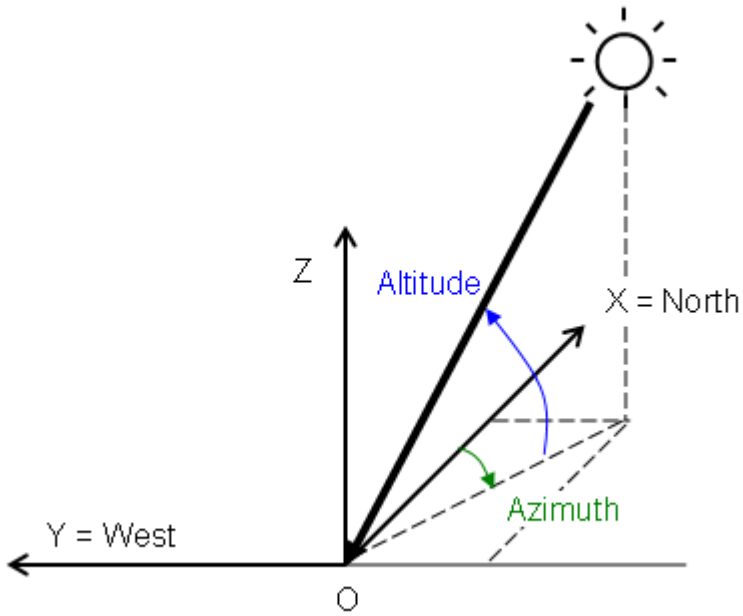
To activate diffuse radiation at a boundary, select **Physical Conditions** > **Radiation Flux Option** for the boundary and set the *Radiation Flux Option* to **Diffuse radiation flux**. This action adds the node **Physics Values** > **Diffuse Radiation Flux**. If you are using the Multiband Thermal Radiation model, this option creates a **Composite** node with a node for each thermal band under it. If you are using the Gray Thermal Radiation model, there is only one set of properties.Select the method for specifying the flux. The value must be positive.

Modeling Solar Radiation Loads

Thermal radiative exchange with the solar environment includes both direct and diffuse components.

The direct solar flux is the irradiation from the sun incident directly on the regions without interference from the atmosphere or other objects. Diffuse solar radiation results from the atmospheric scattering, and reflection and transmission from objects such as the ground.

The coordinate system for the orientation of the direct solar flux is defined in the following figure:



Solar Coordinate System

The X-axis is aligned with the local magnetic north direction and the Y-axis is aligned with the local west direction—that is, the XY plane is the local earth-surface plane. The Z-axis then points skyward since STAR-CCM+ employs right-handed coordinate systems.

Thermal radiative exchanges with the environment include only diffuse effects, where the environment surrounding the regions emits diffusively as a blackbody at the environment radiation temperature.

Solar radiative exchanges have a direct and a diffuse component. The direct solar flux is the irradiation from the sun impacting directly on the regions, without interference from the atmosphere or other objects. Diffuse solar radiation results from the atmospheric scattering, and reflection and transmission from objects such as the ground.

Contents:

The Solar Loads Model
What Is the Solar Calculator?

The Solar Loads Model

The Solar Loads model is used to model radiation loads from the sun — both direct and diffuse. This optional model is available with Surface-to-Surface Radiation and works with either the Gray or Multiband spectrum models.

The solar emissions are considered together with terrestrial emissions within the context of a given spectrum model. For the Gray spectrum model, the total solar loads are applied to the single transfer system representing the full-thermal spectrum. For the Multiband spectrum model, the solar loads are divided between spectral bands that are based on Planck's distribution for a blackbody at the effective surface temperature of the sun. These per-band solar emissions are then subject to the radiation surface properties for the associated band as are all other per-band emissions.

When the Solar Loads model is selected for a continuum, a **Solar Loads** node becomes visible under the **Models** node for that continuum.

Each continuum with radiation can include a Solar Loads model. However, the coordinate and the specification method (with its properties) for the Solar Loads model in every continuum must be the same. The assumption is that only one sun exists and all continua see this single sun.

Some changes to input parameters of the solar load specification method can result in a change in solar direction and fluxes. When this change happens, solar load distribution on patches is immediately recalculated.

Solar Loads Model Reference

Theory	See Theory Guide—Solar Radiation.	
Provided By	**[physics continuum]** > **Models** > *Solar Radiation*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Solar Loads**	
Requires	*Space*: **Three Dimensional** *Optional Models*: **Radiation** *Radiation*: **Surface-to-Surface (S2S) Radiation**	
Properties	Key properties are: *Coordinate System*, *Solar Specification Option*. See Solar Load Model Properties.	
Activates	Model Controls (child nodes)	Manual Specification
	Field Functions	**Diffuse Solar Irradiation**, **Diffuse Solar Irradiation on External Side**, **Direct Solar Irradiation**, **Direct Solar Irradiation on External Side**. See Radiation Field Functions Reference.

Solar Loads Model Properties

Coordinate System

Specifies the coordinate system (laboratory by default). The drop-down list of this property includes any local coordinate systems that exist in the simulation.

Solar Specification Option

Selects the method for specifying solar values.

Manual specification

Activates the Manual Specification sub-node, in which you can specify directions and fluxes.

Solar calculator

Activates the Solar Calculator sub-node, in which you can specify values such as date, time and geographical locations, and obtain calculations of directions and fluxes.

The solar emissions are subject to the radiative surface properties (see Radiation Properties on Boundaries) specified with the use of either the Gray or the Multiband Thermal Radiation models. No additional radiation property specification on boundaries is needed when the Solar Loads model is enabled.

Manual Specification

The Solar Loads model is fully closed by defining the following properties of the **Manual Specification** node: the sun direction vector in a specified coordinate system, the direct solar flux, and the diffuse solar flux. The properties can also be defined with field function syntax. For example, you can define the orientation of the sun as a function of time by defining the *Azimuth* and *Altitude* properties with expressions such as `sin(100*$Time)`.

If the definitions of the angles and/or the direct solar flux are a function of time, the solar loads are recalculated automatically during each time-step.

Manual Specification Properties

Azimuth

Sets the sun direction vector horizontal component.

Altitude

Sets the sun direction vector vertical or elevation component.

Direct Solar Flux

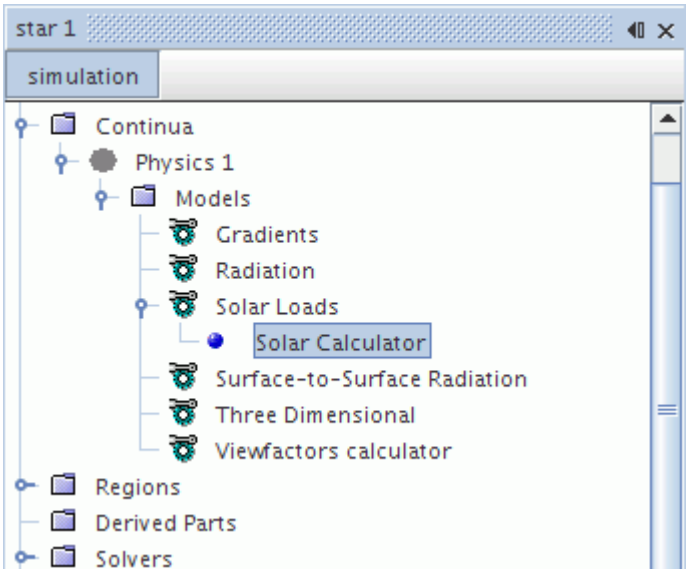
Sets the solar radiative energy directly from the sun along the specified direction vector.

Diffuse Solar Flux

Sets the solar radiative energy from the surrounding environment.

The Solar Calculator

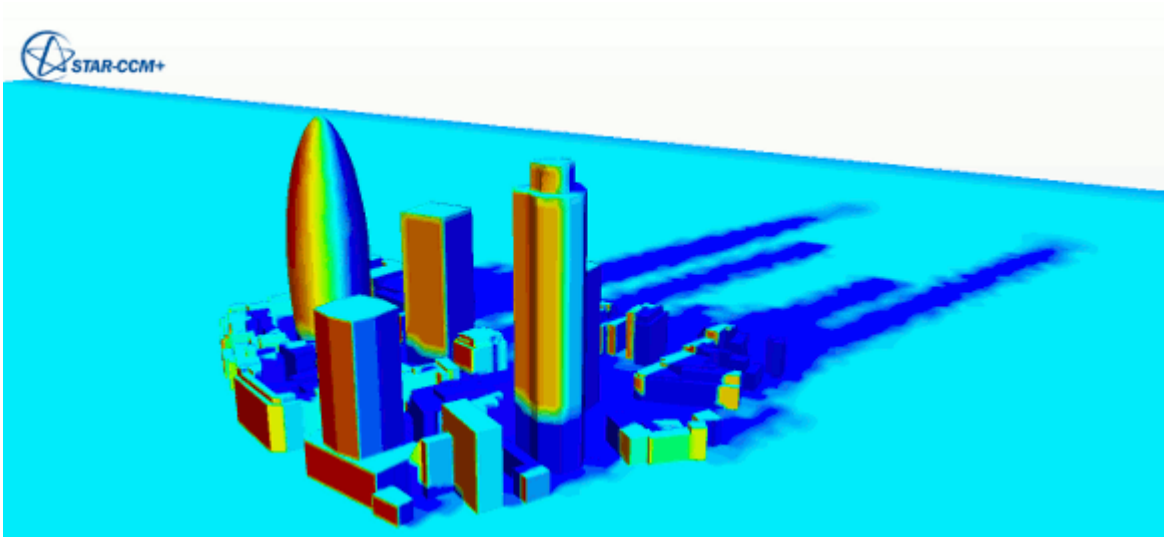
For a chosen date, time, and geographical location, the Solar Calculator computes sun altitude (elevation) and azimuthal angles, along with the corresponding direct and diffuse solar fluxes. Select the **Solar Calculator** node to access the properties that are required by the tool.

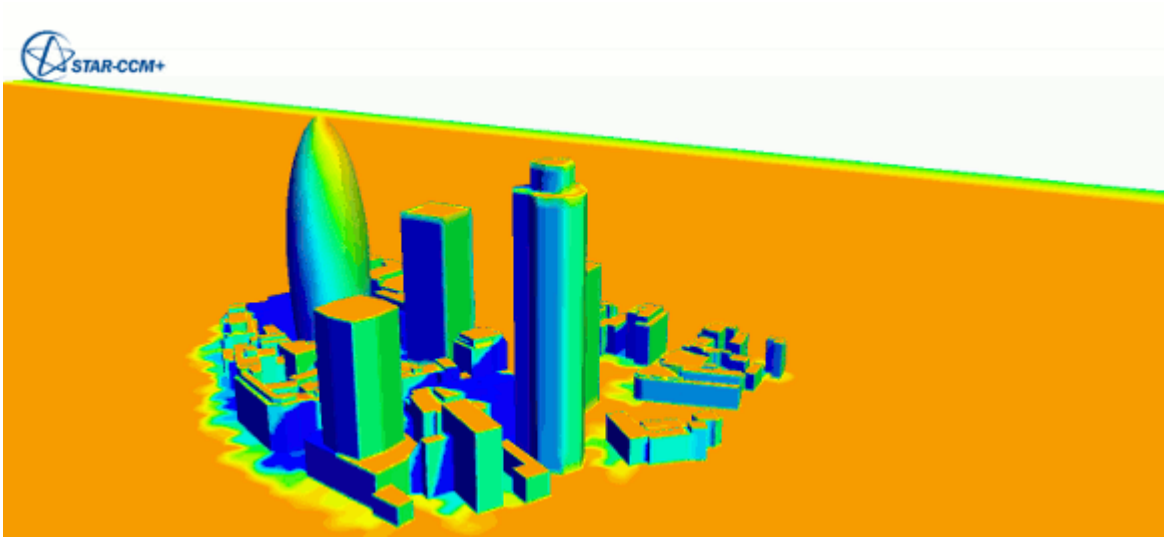


Additional inputs include a sunshine factor to accommodate the clouds and a diffuse fraction to consider a fraction of solar radiation as diffuse (the rest is direct flux). The solar calculator automatically includes a correction due to refraction of solar radiation through the atmosphere of the earth.

For unsteady simulations, the solar calculator automatically refreshes solar values at every time-step.

The following screenshots are examples of scalar displays using results from the solar calculator.





Solar Calculator Properties

Latitude

Latitude in degrees (north positive, south negative; default 51.47917, location of Greenwich, United Kingdom).

Longitude

Longitude in degrees from Greenwich, United Kingdom (east positive, west negative; default 0.0).

Time Zone

Time that is offset with respect to UTC (east positive, west negative; default 0.0 hr).

Year

Starting year in the Gregorian calendar (default 2000).

Month

Starting month of the year.

Day

Starting day of the month.

Hour

Starting local time (24-hour format; default 12:00 noon).

Minute

Starting minute.

Azimuth

Sun direction vector horizontal component (read-only).

Altitude

Sun direction vector vertical or elevation component (read-only).

Direct Solar Flux

Solar radiative energy directly from the sun along the specified direction vector (read-only).

Diffuse Solar Flux

Solar radiative energy from the surrounding environment (read-only).

Current Solar Time

Starting solar time plus elapsed solar time (read-only).

Diffuse Fraction

Fraction of total solar flux to be considered as diffuse (default 0.0); remainder is direct flux.

Sunshine Factor

Factor by which total solar flux is reduced to take into account a cloudy sky (default 1.0).

Radiation Field Functions Reference

The following primitive field functions are made available to the simulation when a radiation transfer model (S2S or DOM) is used.

Boundary Emissivity	The gray emissivity at the boundary. Only available for gray radiation.
Boundary Emissivity of <band-name>	The spectral emissivity at the boundary for band <band-name>. Only available for spectral radiation (multiband).
Boundary Irradiation	The irradiation at the boundary. The irradiation is the radiative heat flux incident on a surface. This property is a full-spectrum quantity. For spectral radiation (multiband), it includes contributions from all wavelengths (bands).
Boundary Irradiation of <band-name>	The irradiation for band <band-name>. Only available for spectral radiation (multiband).
Boundary Radiation Heat Flux	The net radiative heat flux to the boundary (absorption minus emission). A positive value indicates a net absorption by the boundary. This property is a full-spectrum quantity. For spectral radiation (multiband), it includes contributions from all wavelengths (bands).
Boundary Radiosity	The radiosity at the boundary. The radiosity is essentially the radiative heat flux leaving a surface. This property is a full-spectrum quantity. For spectral radiation (multiband), it includes contributions from all wavelengths (bands).
Boundary Radiosity of <band-name>	The radiosity for band <band-name>. Only available for spectral radiation (multiband).
Boundary Reflection Specularity	The fraction of total reflectivity which is specular in nature. The second one is on the external side when environment boundary condition is selected.
Boundary Reflection on External Side	The fraction of total reflectivity which is specular in nature, on the external side of a boundary. Only available when an environment boundary condition is selected.
Boundary Reflectivity	The gray reflectivity at the boundary. Only available for gray radiation.
Boundary Reflectivity of <band-name>	The spectral reflectivity at the boundary for band <band-name>. Only available for spectral radiation (multiband).

Boundary Transmissivity	The gray transmissivity at the boundary. Only available for gray radiation.
Boundary Transmissivity of <band-name>	The spectral transmissivity at the boundary for band <band-name>. Only available for spectral radiation (multiband).
Total Emissivity	The blackbody-emissive-power-weighted emissivity at the boundary. For gray radiation, this property is simply the gray surface emissivity. For spectral radiation, this property is defined as: $\varepsilon_{total} = \left(\sum_{i \in bands} n_i^2 \varepsilon_i E_{b,i} \right) / \left(\sum_{i \in bands} E_{b,i} \right)$ where n is the index of refraction, E_b is the blackbody emissive power and the subscript i denotes the band.
User-specified Diffuse Flux	The user-defined value of diffuse radiation flux specified on a boundary. This flux models non-thermal (non-Planck) emission coming out of a boundary. It does not affect the originating boundary thermally but can affect other boundaries thermally. For spectral radiation (multiband), it can be specified per band.

The following primitive field functions are additionally made available when the Surface-to-Surface (S2S) Radiation model is activated.

Boundary Irradiation on External Side	The irradiation of the external side of a boundary between the environment and an internal domain. Available only when an environment condition is applied.
Boundary Mean Radiant Temperature	The mean radiant temperature (MRT) at the boundary. See Mean Radiant Temperature (MRT).
Diffuse Solar Irradiation	The solar flux on the boundary from the surrounding environment. This quantity is only available for gray radiation when the Solar Loads model is activated.
Diffuse Solar Irradiation of <band-name>	The solar flux on the boundary from the surrounding environment for band <band-name>. This quantity is only available for spectral radiation (multiband) when the Solar Loads model is activated.
Diffuse Solar Irradiation on External Side	The solar flux from the surrounding environment on the external side of a boundary which has the **Thermal Specification** condition set to **Environment**. This quantity is only available for gray radiation when the Solar Loads model is activated.

Direct Solar Irradiation	The direct solar flux on the boundary. This quantity is only available for gray radiation when the Solar Loads model is activated.
Direct Solar Irradiation of <band-name>	The direct solar flux on the boundary for band <band-name>. This quantity is only available for spectral radiation (multiband) when the Solar Loads model is activated.
Radiosity on External Side	The radiosity of the external side of a boundary between the environment and an internal domain. Available only when an environment condition is applied.
Radiation Heat-Flux on External Side	The radiation heat-flux of the external side of a boundary between the environment and an internal domain. Available only when an environment condition is applied.
Reflectivity on External Side	The reflectivity of the external side of a boundary between the environment and an internal domain. Available only when an environment condition is applied.
Reflection Specularity on External Side	The reflection specularity of the external side of a boundary between the environment and an internal domain. Available only when an environment condition is applied.

The following primitive field functions are additionally made available when the Participating Media Radiation (DOM) model is activated.

Absorption Coefficient	The absorption coefficient for the media. For spectral radiation (multiband), each band has a separate variable which is differentiated by the band index, for example **Absorption Coefficient 0** for the first band.
Incident Radiation	The zeroth angular moment of radiative intensity ($G = \int_{4\pi} I \, \mathrm{d}\, \Omega$). This quantity is also proportionally related to the energy density ($u_e = G/c$ where c is the speed of light) and to the mean intensity ($\bar{I} = G/4\pi$). This property is a full-spectrum quantity. For spectral radiation (multiband), it includes contributions from all wavelengths (bands).
Particle Radiative Energy Source	The volumetric (integrated over dispersed media in a given cell) source due to radiative absorption and emission by dispersed media, such as particles (absorption by particles minus emission from particles). A positive value indicates net absorption by the dispersed media. This quantity is equal to the

	negative of the divergence of the radiative heat flux from dispersed media.
Radiative Absorption	The volumetric absorption by the media. This quantity is the absorption part of the Radiative Energy Source.
Radiative Energy Source	The volumetric source to the energy equation due to radiative absorption and emission (absorption minus emission). A positive value indicates net absorption by the media. This quantity is equal to the negative of the divergence of the radiative heat flux. This field function is available for continuous media only.
Radiative Intensity	The radiative intensity for the jth ordinate direction. For spectral radiation (multiband), this variable contains the intensity field for the last band that was solved (it is not a full-spectrum quantity).
Refractive Index	The refractive index for the media. This quantity is only available for gray radiation when the Refraction (Gray) model is activated.
Refractive Index i	The refractive index for the media, of the ith band. This quantity is only available for multiband thermal radiation when the Refraction (Multiband) model is activated.
Scattering Coefficient	The scattering coefficient for the media. For spectral radiation (multiband), each band has a separate variable which is differentiated by the band index, for example **Scattering Coefficient 0** for the first band.
Total Absorption Coefficient	The blackbody-emissive-power-weighted absorption coefficient within the domain. For gray radiation, this property is simply the gray absorption coefficient. For spectral radiation, it is defined as: $\kappa_{total} = \left(\sum_{i \in bands} n_i^2 \kappa_i E_{b,i} \right) / \left(\sum_{i \in bands} E_{b,i} \right)$ where n is the index of refraction, E_b is the blackbody emissive power and the subscript i denotes the band.

The following primitive field functions are additionally made available when the View Factors Calculator model is activated.

Radiation Patch Id	An ID number identifying the boundary face the patch belongs to.
Radiation patch Id on External Side	An ID number identifying the boundary face the patch belongs to. Available only for the external side of the boundary when an environment condition is applied.

Voxel partition	The partition ID of the voxel mesh, which is used for view factor ray tracing. The voxel mesh has different domain decomposition from the CFD mesh.

Modeling Aeroacoustics

STAR-CCM+ offers a choice of models for simulating acoustics in liquids and gases.

To date, the primary focus has been on aeroacoustics, the study of noise from flows of gases as they interact with surfaces. However, STAR-CCM+ also allows you to model hydroacoustics, the study of noise transmission in water.

The following general types of acoustics models are available in STAR-CCM+:

- Broadband Noise Source
- APE (Acoustic Perturbation Equations) based noise source
- Direct Noise Simulation
- Ffowcs Williams-Hawkings
- Lighthill Stress Tensor

Note: Use the double precision version of STAR-CCM+ for all aeroacoustic analyses. For more information, refer to the aeroacoustics guidelines section.

Contents:

What is Aeroacoustics
How Do I Conduct an Aeroacoustics Analysis?
Selecting an Aeroacoustics Model
Using Broadband Noise Source Models
Using the Acoustic Wave Model
Using Direct Noise Calculation
Using the Acoustic Suppression Zone Model
Using the Ffowcs Williams-Hawkings Aeroacoustics Models
Using the Lighthill Stress Tensor
Aeroacoustics Bibliography
Modeling Hydroacoustics

What Is Aeroacoustics?

Aeroacoustics investigates the aerodynamic generation of sound.

Modern aeroacoustics science was initiated in the 1950's by Sir James Lighthill who derived a theory for the estimation of the intensity of sound that radiates from a turbulent flow. Before this, flow-generated noise studies were focused on the relation between the frequency of the fluid fluctuations and the emitted sound.

The frequency of the aeolian tone was mathematically related to flow parameters by Strouhal who, as a result of quantitative observations, introduced the non-dimensional frequency - the Strouhal number. Lord Rayleigh related the Strouhal number to the flow Reynolds number and further recognized that the direction of most intense sound emission coincides with the direction of the fluctuating force on the wire.

In 1952, Lighthill established the theoretical background that is generally referred to when investigating aerodynamic noise [80]. Lighthill first introduced the concept of aeroacoustic analogy which consists of replacing the actual flow field responsible for generating noise with an equivalent system of noise sources. The noise sources act on a uniform stagnant fluid that is governed using standard acoustic propagation equations. The aerodynamic characterization of the sources then becomes the main issue in noise prediction.

In 1955, Curle [64] extended the concepts that Lighthill developed to include the effect of flow-body interaction on sound generation. In 1969, Ffowcs Williams and Hawkings [69] extended the Lighthill analogy to account for arbitrary surface motion. This formulation was used for noise prediction in rotor blades aerodynamics, such as in the helicopter and turbine industries [59] and [65].

The main mechanisms of sound generation in the presence of solid structures can be classified as [93]:

- Vortex shedding noise

  Vorticity that is released from a bluff body in a flow. The time varying circulation on the body due to vortex shedding induces a fluctuating force on the body itself which is transferred to the fluid and propagates as sound.
- Turbulence-structure interaction noise

  Vortical structures impinging on a solid surface generate local pressure fluctuations on the body surface which feed the acoustic far field.
- Trailing edge noise

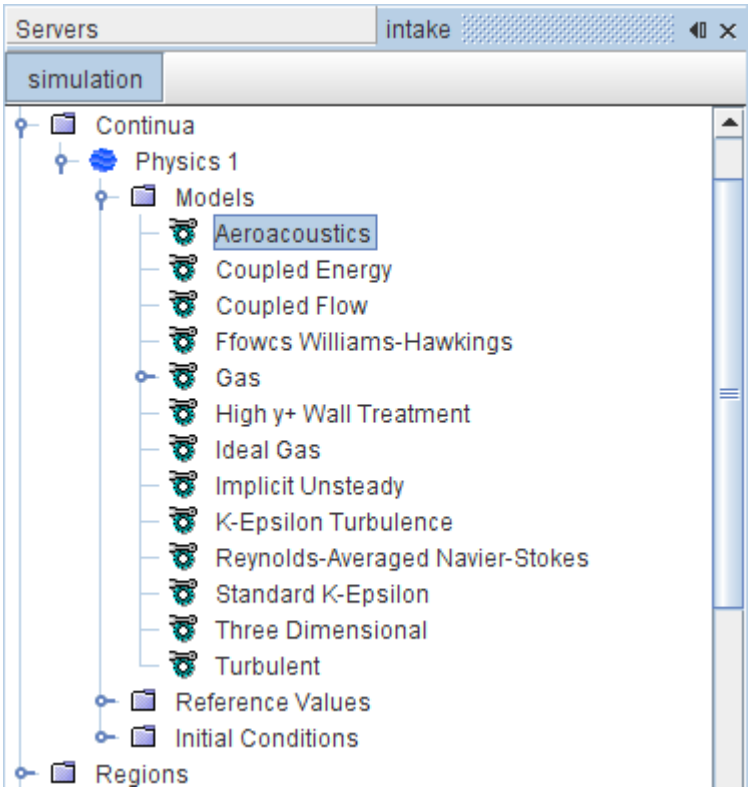
  This type of noise is important for all rotating blade technologies, due to the interaction of the boundary layers instabilities with the surface edges.

In current engineering practice, computations of sheared and free shear flows using the unsteady compressible Navier-Stokes equations are performed routinely. It remains challenging, however, to resolve the acoustic field of those simulations whose acoustic energy is many orders of magnitude smaller than the energy of the hydrodynamic field. There are four sources of difficulty in directly resolving the acoustic waves using Navier-Stokes computations (from [51]):

- The acoustic-field is larger than the flow-field.
- The acoustic-field has a smaller energy content than the flow field.
- The numerical discretization can act as a more significant source of sound than the simulated flow-field.
- The difficulty of imposing free space boundary conditions appropriate for acoustics in the far-field, at an artificial computational boundary that is positioned at a finite distance away from the source region.

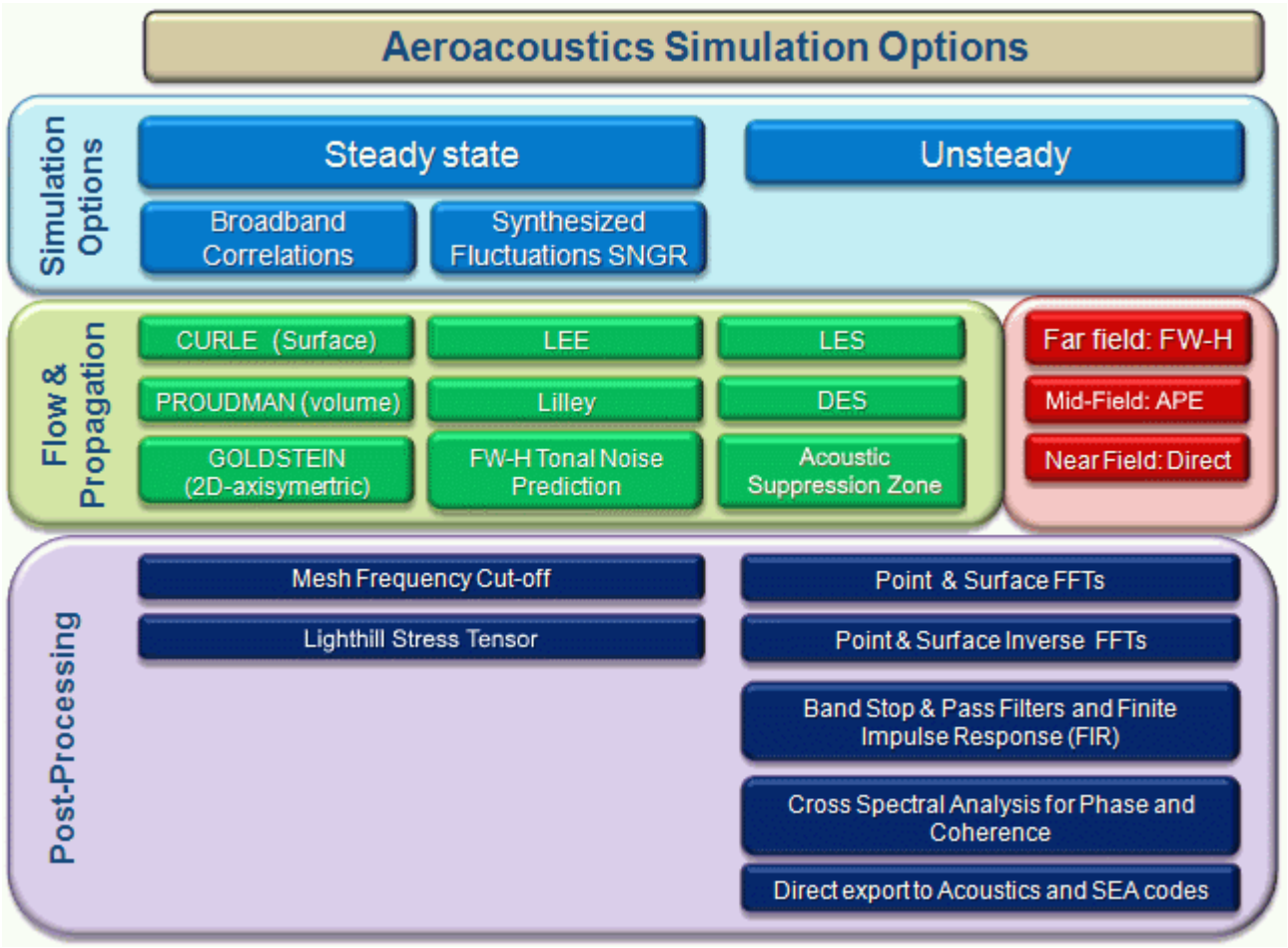
Artificial boundaries must be appropriate to ensure non-reflection of acoustic waves, but also in certain regions to provide inflow and outflow of the aerodynamic field.

The aeroacoustics models in STAR-CCM+ are responsible for providing aeroacoustics equations. The **Aeroacoustics** model node represents the selection of aeroacoustics models.



How Do I Conduct an Aeroacoustics Analysis?

STAR-CCM+ offers a hierarchy of models to conduct an aeroacoustics analysis.



The physics in aeroacoustics are inherently transient and ultimately, must be solved using transient calculations. The computational run times that are associated with such simulations are high. As such transient simulations take a long time to run, optimize the mesh and settings so that you only capture noise sources and frequencies that are relevant to your analysis. An initial steady state analysis provides you with the data for optimizing the mesh and settings.

In steady state analysis, there are two forms of aeroacoustic modeling available:

The first category relates to models that provide aeroacoustic source information through correlations. These models are based on the assumptions implicit to RANS turbulence modeling. They are termed Broadband Noise Source Models and can be attributed to the classical aeroacoustic categorization of dipoles and quadrupoles in shear and jet flows. They are strictly applied to turbulence-generated flow-noise--they do not apply to noise associated with large-scale flow features, such as vortex shedding. For turbulence generated noise, they will:

- Identify the location of flow generated aeroacoustic sources.

- Approximate the associated sound power or dB level.

They provide a reasonable basis on which to:

- Compare one component design with another.
- Decide where to apply mesh refinement ahead of a well-qualified transient calculation.

You can further qualify the suitability of the mesh ahead of a transient calculation by analyzing the frequency that the mesh can resolve. The suitability analysis determines where, and by how much, the mesh must be refined to capture up to the desired frequency. The resolvable frequency is deduced from the mesh frequency cutoff scalar function. If, for example, the function value shows 1 kHz and you want to resolve up to 2 kHz, then refine the mesh a factor of two.

The second category relates to synthesization of turbulent perturbations from the steady state using Synthesized Fluctuations SNGR. The modeled turbulence kinetic energy and length scales are used to create synthetic fluctuations in pressure and velocity through assumption of the turbulent energy spectrum and wave number decomposition. The fluctuations may then be output directly for use in acoustics propagation codes.

After you have optimized the steady state model for an aeroacoustics simulation, you can use it to initialize the transient simulation.

Recommended Analysis Procedure

Set up aeroacoustics simulations as three-dimensional cases. Use compressible flow rather than constant density. Then use the following steps:

1. Set up and run the case in steady state mode using a relatively coarse mesh to optimize the case for aeroacoustics. Use the broadband noise source models to identify where to refine the mesh. Use the mesh frequency cut-off scalar function to judge the required size of the cells in the source region.
2. Load the case set up in Step 1. Refine the mesh using information from the broadband noise models and the mesh frequency cut-off scalar function. Run the case again in steady state mode, restarting the solution from that of Step 1.
3. Load the case set up in Step 2 and modify it to be a transient simulation with an appropriate time-step. Choose LES-based or DES-based turbulence modeling and restart the solution from that of Step 2.

   To be able to post-process Lighthill Stress Tensor field functions, select the Lighthill Stress Tensor.

   For far field propagation analysis, use the Ffowcs Williams-Hawkings (FWH) model.
4. Run the simulation.

Guidelines

See Guidelines for Aeroacoustics Calculations for guidance on how best to apply the aeroacoustics models to your applications.

Selecting an Aeroacoustics Model

To select the Aeroacoustics model:

1. In the *Physics Model Selection* dialog, select a flow model from the *Flow* group (**Coupled Flow** or **Segregated Flow**). The **Aeroacoustics** radio button becomes available.
2. Select the **Aeroacoustics** model in the *Optional Models* group.

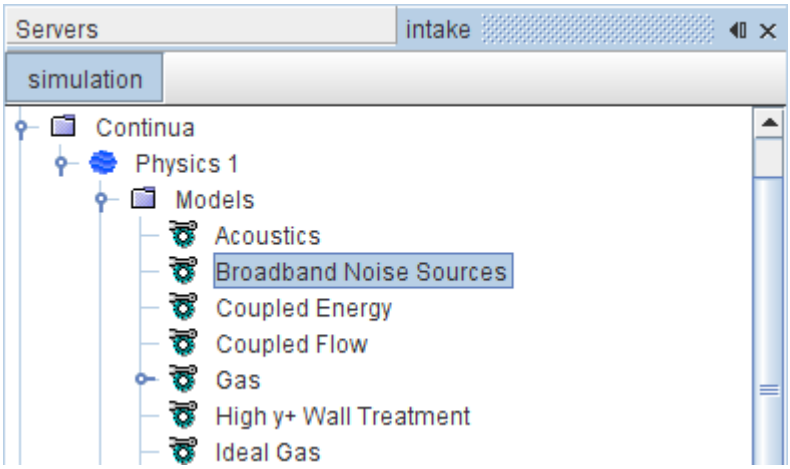
   Following its selection, the *Physics Model Selection* dialog shows the available aeroacoustics models.

- The Lighthill Stress Tensor model is available with all turbulence models.
- The Ffowcs Williams-Hawkings model is available with both steady and implicit unsteady models.
- The Broadband Noise Source models are available as follows:
  - Curle, Goldstein, and Proudman models can be used in steady and unsteady (transient) analyses. These models are compatible with all Reynolds-Averaged Navier-Stokes (RANS) models that can provide turbulence time and length scales.
  - Linearized Euler Equation (LEE) and Lillely Noise Source Model are available only for steady-state simulations. These models can be used with Reynolds-Averaged Navier-Stokes (RANS) cases that include turbulence time and length scales.
- The Acoustic Wave model is available with implicit unsteady models. The Acoustic Wave model is not compatible with Explicit, Two-Dimensional, or Axisymmetric models. The Acoustic Wave model is compatible with stationary cases only-the simulation cannot include any moving regions (such as fan blades).

Using Broadband Noise Source Models

In near-field noise prediction, use broadband noise source models to compute the location and strength of the main noise sources.

The noise sources consist of a volume distribution of quadrupole sources and a surface distribution of dipole sources. STAR-CCM+ offers a group of models capable of evaluating the broadband sources of noise. The selection of this group is represented in the object tree by the **Broadband Noise Sources** node.



In the frequency domain, a broadband noise has a continuous spectrum, where the acoustic energy is continuously distributed at all frequencies in a given range. These models can compute the location and strength of the main sources of noise that are generated aerodynamically. They use flow field quantities from Reynolds-Averaged Navier-Stokes (RANS) Turbulence equations. Because these broadband noise models use RANS simulations, the required computational time is small.

In STAR-CCM+, the following broadband noise source models estimate the local acoustic power:

For dipole-generated sound, generated per unit surface, use:

- Curle noise source model

For quadrupole-generated sound, generated per unit volume, use:

- Goldstein axisymmetric noise source model
- Proudman noise source model

For sound that the turbulent shear flow generates, use:

- Lilley noise source model
- Linearized Euler Equation (LEE) noise source model

For suppressing spurious reflections, use:

- Acoustic suppression zone model

You can use the numerically predicted loudness rankings to characterize and compare different designs.

Contents:

Selecting a Noise Source Model
Dipole and Quadrupole Broadband Noise Sources
Stochastic Shear and Self Noise Sources
Working with Mesh Frequency Cutoff
Noise Source Field Function Reference

Selecting a Noise Source Model

The Broadband Noise Sources model makes a range of noise sources available for simulation.

To select a noise source model, use the following steps:

1. Choose a Reynolds-Averaged Navier-Stokes (RANS) turbulence modeling approach, and select the models entailed.
2. Select **Broadband Noise Sources** from the *Aeroacoustic Models* box.
3. Select one of these models from the *Broadband Noise Source Models* box:
   - Goldstein:

     This model is available only if you have already selected **Axisymmetric** from the *Space* box.
   - LEE
   - Lilley
   - **Noise Source Models**:

     Selecting this opens the *Noise Source Models* box. This box contains the following models:

     - Curle
     - Proudman

     These models can be selected individually or together.

Dipole and Quadrupole Broadband Noise Sources

The Curle, Goldstein Axisymmetric, and Proudman noise source models compute total acoustic power per unit area or per unit volume.

The Curle Noise Source Model

The Curle Noise source model evaluates the noise from a turbulent boundary layer flow over a solid body at low Mach number.

The model represents dipole sources of noise, that is, noise coming from the fluctuating surface pressure from solid boundaries acting on the fluid. Most system noise is generated for low-Mach number flows. Specifically, this model computes the surface acoustic power to evaluate the local contribution to the total acoustic power per unit area of the body surface. By contrast, the Goldstein and Proudman models evaluate the acoustic power on the volume, with quadrupole-generated sound sources. This model assumes isotropic turbulence.

The Curle model has been used for air handling subsystem (AHS) applications, including rotating parts (such as blowers), heat exchangers, mufflers, distribution ducts and wall-enclosed spaces.

Like the Goldstein and Proudman models, this model can be used in steady and unsteady (transient) analyses. It is compatible with all Reynolds-Averaged Navier-Stokes (RANS) Turbulence models that can provide turbulence time and length scales. This model is preferable for steady-state solutions. It is available for coupled and segregated flow. The CPU time is comparable to the times of the Goldstein and Proudman models, and calculations are done at the end of each iteration (or time-step).

See also the Curle Field Functions.

The Goldstein Axisymmetric Noise Source Model

The Goldstein model is based on Lighthill's theory for predicting aerodynamic noise from a turbulent shear flow on an assumption of locally homogeneous and anisotropic turbulence.

Based on Goldstein's theory [74], this model is a generalization of the model Ribner [91] developed for axisymmetric turbulent flow. It eliminates the Ribner assumption of the space-time correlations of Reynolds stresses.

This model is used to compute the local noise contribution per unit volume for anisotropic turbulence in axisymmetric turbulent jets.

In STAR-CCM+, this model is available for all cells from a two-dimensional axisymmetric case. It can be used in either a steady or transient simulation, with either coupled or segregated flow. Although it is preferable for steady simulations, the Goldstein model could be used for an unsteady simulation in which the correct flow field for convergent steady-state simulations is not available. It is compatible with any of the Reynolds-Averaged Navier-Stokes (RANS) Turbulence models.

See also the Goldstein Field Functions.

The Proudman Noise Source Model

Like the Goldstein model, the Proudman noise source model evaluates acoustic power per unit volume, and the sound is from quadrupoles.

Specifically, this model computes the acoustic power to evaluate the local contribution to the total acoustic power per unit volume from the turbulent flow. The Proudman model assumes isotropic turbulence.

The Proudman model can be used for quadrupole sources of noise, such as the areas around blower blades. This model has been used for air handling subsystem (AHS) applications, including rotating parts (such as blowers), heat exchangers and distribution ducts.

Like the Curle and Goldstein models, this model can be used in steady and unsteady (transient) analyses. It is compatible with all Reynolds-Averaged Navier-Stokes (RANS) Turbulence models that can provide turbulence time and length scales. This model is preferable for steady-state solutions. It is available for coupled and segregated flow. The CPU time taken to run the Proudman model is comparable to time required for either the Curle and Goldstein models, and calculations are done at the end of each iteration (or time-step).

See also the Proudman Field Functions.

Curle, Goldstein, and Proudman Noise Source Properties

The Curle, Goldstein, and Proudman noise source models share the following properties, except where otherwise noted.

Alpha	The alpha coefficient, used by the Proudman model only.
Far-Field Density	The density of the far-field fluid. The default value is 1.225 kg/m^3.
Far-Field Sound Speed	The speed of sound in the far-field. See Eqn. (3885) The default value is 340 m/s.
Reference Acoustic Power	An input to a model-specific acoustic power equation: If the Curle model is selected, the acoustic power per unit surface area is calculated in decibels using Eqn. (3891). The default reference acoustic power is 1.0E-12 W/m^2. If the Goldstein model is selected, the acoustic power per unit volume is calculated in decibels using Eqn. (3903). The default reference acoustic power is 1.0E-12 W/m^3. If the Proudman model is selected, the acoustic power per unit volume is calculated in decibels using Eqn. (3908). The default reference acoustic power is 1.0E-12 W/m^3.

Stochastic Shear and Self Noise Sources

The Lilley and LEE noise source models compute the turbulent shear flow noise from the fluctuating velocity flow field.

The Lilley Noise Source Model

The Lilley Noise Source model computes three terms, as field functions:

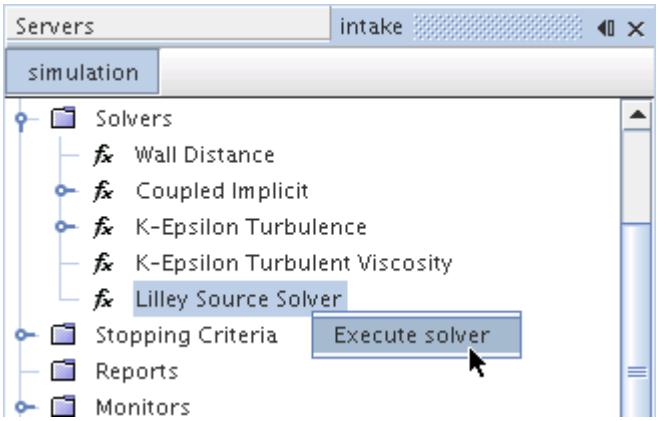
- Lilley Self-Noise Source
- Lilley Shear-Noise Source
- Lilley Total-Noise Source

Applications include problems where local shear effects dominate, such as flow around solid bodies and shear flows in channels. This model evaluates the location and strength of aeroacoustic sound sources.

This model is available only for steady-state simulations. Use it with Reynolds-Averaged Navier-Stokes (RANS) Turbulence cases that include turbulence time and length scales.

The Lilley Noise Source model is compatible with the coupled and segregated flow models. Computational costs (specifically CPU time) depend on input of two properties: *Number of Realizations* and *Number of Fourier Modes*.

The computation of the Lilley field functions involves a separate procedure with the Lilley Source solver after the simulation run is finished. Right-click the **Lilley Source Solver** node and select **Execute solver**.



The field function calculations are available for post-processing.

The Lilley Source Solver

The Lilley noise source model activates the Lilley source solver.

The purpose of this solver is to control the iterative update for the calculation. The Lilley Source solver first calls the Stochastic Noise Generation and Radiation (SNGR) solver, then it computes the Lilley Noise Source Terms. The SNGR solver computes the synthetic turbulent velocity field and gradients of that field using the SNGR method.

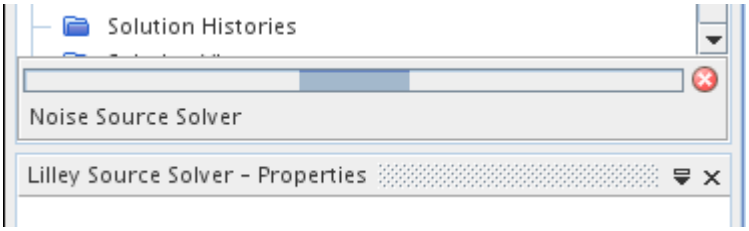
Although this solver has no properties, it provides a pop-up menu action to calculate the noise source values for the Lilley model.

Lilley Source Solver Pop-Up Menu

Execute solver	Controls the Lilley Source solver. This action calculates values for the Lilley field functions.

Stopping the Lilley Source Solver

To stop the solver immediately, without completing the set number of realization steps, click on the solver stop button located next to the progress bar. The solver stops and saves the partially computed field functions.



The Linearized Euler Equation (LEE) Noise Source

The LEE account for refraction and convection effects in any sheared mean flows. The equations can be used for a wider range of conditions than the Lilley equation, where the associated source term is a nonlinear function of the fluctuating velocity flow field. See [55].

The LEE Noise Source Model is compatible with both the coupled and segregated flow models. Computational costs (specifically CPU time) depend on the properties *Number of Realizations* and *Number of Fourier Modes*.

This model is available only for steady-state simulations. It activates the following field functions:

- LEE Self Noise Source
- LEE Shear Noise Source
- LEE Total Noise Source

The LEE Source Solver

The LEE noise source model activates the LEE Source solver. Its primary function is to control the iterative update for the calculation.

The Lee Source solver first calls the Stochastic Noise Generation and Radiation (SNGR) solver, then computes the LEE Noise Source Terms. The SNGR solver computes the synthetic turbulent velocity field and gradients of that field using the SNGR method. Although the LEE Source solver has no properties, it provides a pop-up menu from which to run the solver.

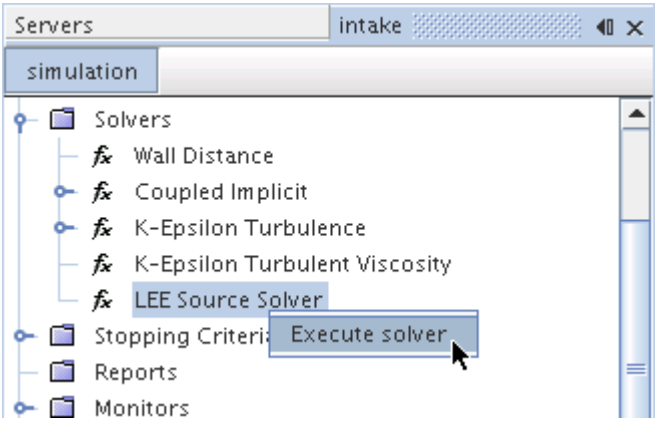
LEE Source Solver Pop-Up Menu

Execute solver	Controls the LEE Source solver. This action calculates values for the LEE noise source model's field functions.

Executing the LEE Source Solver

To run the LEE Source Solver and obtain the field functions:

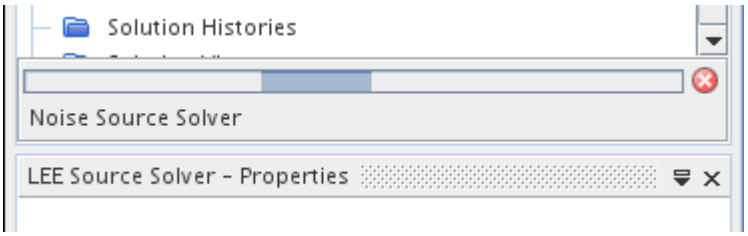
1. Obtain a converged steady state solution, or a converged transient time-step.
2. Right-click the **LEE Source Solver** node and select **Execute solver**.



The field function calculations are available for post-processing.

Stopping the LEE Source Solver

To stop the solver immediately, without completing the set number of realization steps, click on the solver stop button located next to the progress bar. The solver stops and saves the partially computed field functions.



Lilley and LEE Noise Source Properties

The Lilley and LEE noise source models share the following properties.

Number of Realizations	The number of samples used to compute the average of the synthetic turbulent velocity field based on the Stochastic Noise Generation and Radiation (SNGR) approach. The default value is 200. You can use up to 1000 realizations (see Karweit *et al.* [77]).
Number of Fourier Modes	The number of the Fourier modes used for the spectral resolution for isotropic turbulence given by a von Karman-Pao spectrum, when computing the synthetic turbulent velocity field (see n from Eqn. (3924)). The default value is 50. Use values between 30 and 50 Fourier modes.

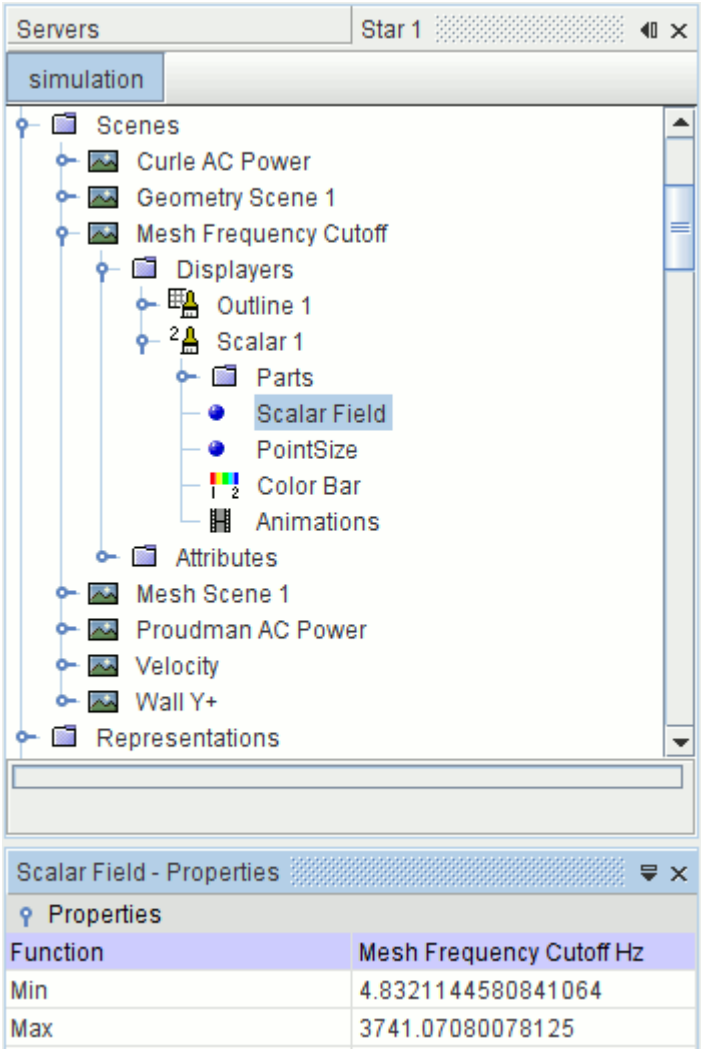
Working with Mesh Frequency Cutoff

The mesh frequency cutoff scalar function can be used to qualify the suitability of the mesh ahead of a transient calculation by analyzing the frequency that the mesh can resolve.

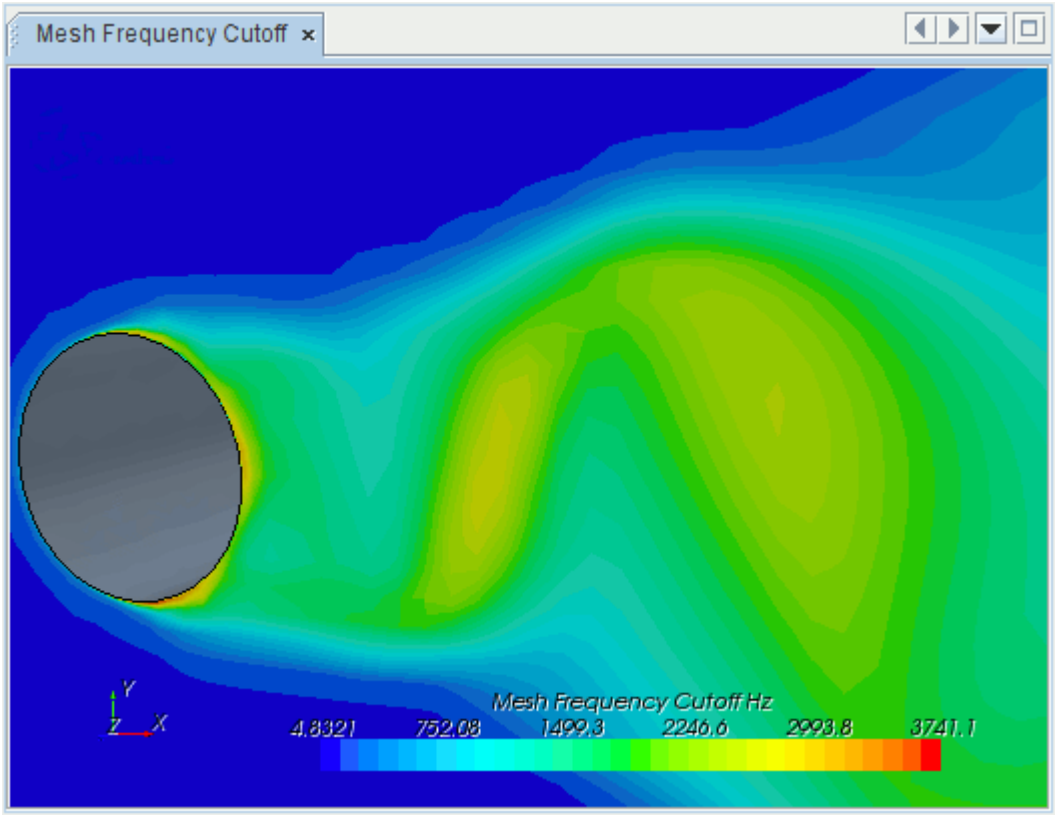
This analysis determines where and by how much to refine the mesh to capture up to the desired frequency. If, for example, the function value shows 1 kHz and you want to resolve up to 2 kHz, then refine the mesh by a factor of two.

The mesh frequency cutoff field function becomes available when the Curle, Proudman and Goldstein broadband noise source model are activated.

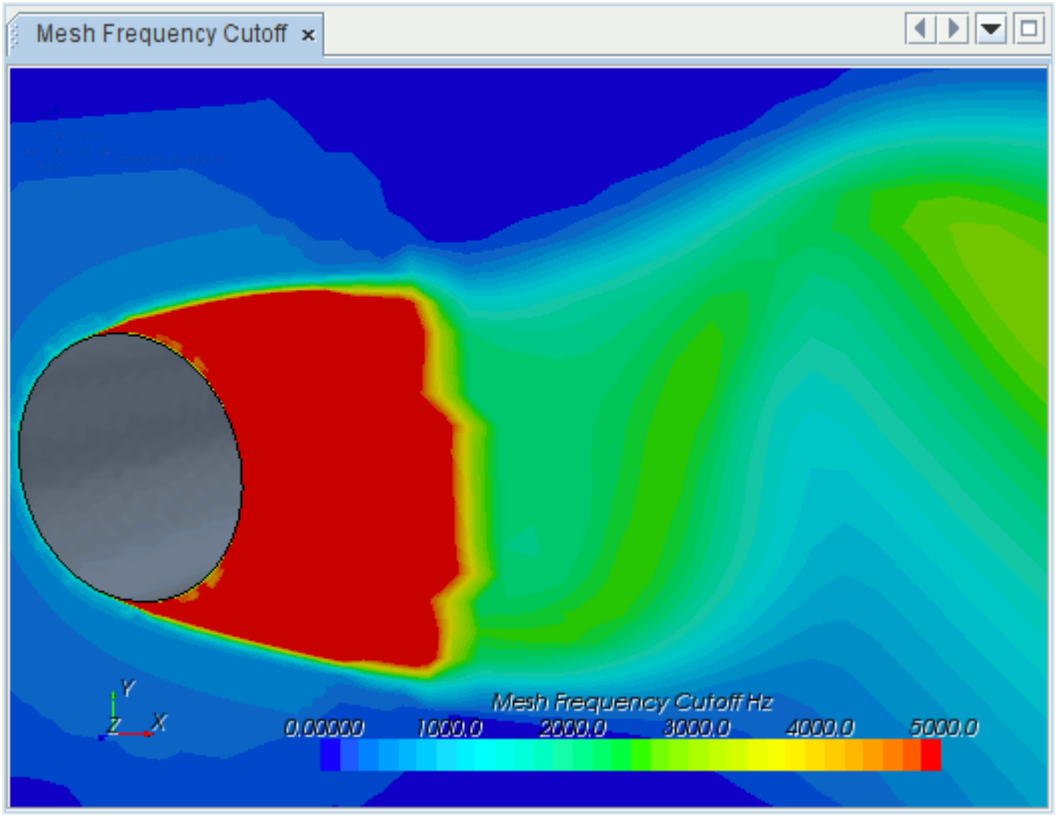
Create a scalar scene to display the mesh frequency cutoff contour at a desired location and select **Mesh Frequency Cutoff Hz** for the scalar field function.



The following screenshots provide an example of working with the mesh frequency cutoff scalar function. See the Flow Generated Noise Around a Cylinder tutorial for further details.



The above plot shows the mesh frequency cutoff contour for a coarse mesh. This plot indicates that the mesh is able to resolve up to 2 kHz (approximately) in the main volume source regions indicated by the Proudman correlation (in this example). Typically, however, the peak of the human hearing range is between 1-3 kHz. Therefore, refine the mesh further to increase the resolvable frequency. The plot that is shown in the following screenshot shows the same scalar function with the refined mesh.



From the above plot, it is clear that the fine mesh adequately resolves higher frequencies close to the cylinder.

Noise Source Field Function Reference

The following primitive field functions are made available to the simulation when the respective noise source models are used. All of them have the same base properties.

Curle Field Functions

Curle Surface Acoustic Power	The local surface acoustic power. Computed with Eqn. (3891).
Curle Surface Acoustic Power DB	The local surface acoustic power in dB units. Computed with Eqn. (3892).

Goldstein Field Functions

Goldstein Acoustic Power	The acoustic power per unit volume, which can be reported using dimensional units ($\mathrm{W/m^3}$), defined in Eqn. (3893).
	The acoustic power per unit volume, which can be reported using dB, defined in Eqn. (3903).

Goldstein Acoustic Power DB	

Proudman Field Functions

Proudman Acoustic Power	The local acoustic power that is computed with Eqn. (3906).
Proudman Acoustic Power DB	The local acoustic power in dB that is computed with Eqn. (3908).

Lilley Field Functions

Lilley Self Noise Source	The Lilley source term that computes the effect of noise of the turbulent velocity components, see Eqn. (3916).
Lilley Shear Noise Source	The source term which computes the noise from the interaction between the mean shear flow and turbulent velocity components, see Eqn. (3917).
Lilley Total Noise Source	The summation of the Lilley Self-Noise Source and Lilley Shear Noise Source.

LEE Field Functions

LEE Self Noise Source	The LEE source term that computes the sound generated by turbulent velocity components. See Eqn. (3922).
LEE Shear Noise Source	The LEE source term that computes the acoustic effects from interaction between the mean shear flow and turbulent velocity components. See Eqn. (3918).
LEE Total Noise Source	Summarizes the LEE Self Noise Source and LEE Shear Noise Source.

Using the Acoustic Wave Model

The Acoustic Wave model uses the theory of Acoustic Perturbation Equations (APE) to model noise that is generated by the flow of an incompressible fluid.

The Acoustic Wave solver is one-way coupled with the CFD solver through this APE-based noise source term. In this approach, STAR-CCM+ models the acoustic source field inside a specified zone of interest only. A key part of the setup for this model is that you define the zone within which noise is generated.

The Acoustic Wave model is intended for use in unsteady flow simulations, typically incompressible (constant density). By adopting a hybrid approach, the Acoustic Wave model produces better results (with fewer spurious effects) than a pure compressible flow simulation. Perform acoustic wave simulations in double precision.

The Acoustic Wave model uses the mean flow pressure and velocity fields to compute the convective terms of the wave equation. These fields must be obtained before starting the aeroacoustic simulation. See Preliminary Analysis.

For the formulation of the Acoustic Perturbation Equation, see Acoustic Wave Model theory.

Contents:

Preliminary Analysis
Acoustic Wave Analysis
Post Processing
Selecting the Acoustic Wave Model
Clearing Acoustic Data from the Simulation
Setting Up the Noise Source for the Acoustic Wave Simulation
Reducing Spurious Acoustic Reflections
Acoustic Wave Conditions and Values
Acoustic Wave Solver Properties
Guidelines for Using the Acoustic Wave Model
Acoustic Wave Solver Field Functions Reference

Preliminary Analysis

Before you begin the acoustic wave simulation, obtain a fully developed flow solution, then obtain mean flow pressure or velocity fields, according to the kind of simulation, and define filtering and damping.

1. For hybrid flow and aeroacoustics simulations with convection, set up a **Field Mean of Pressure** monitor:
   a) Right-click the **Monitors** node and then click **New Monitor** > **Field Mean**.
   b) Select the new **Field Mean** node and set the following properties:

Property	Value
Parts	[Flow Region]

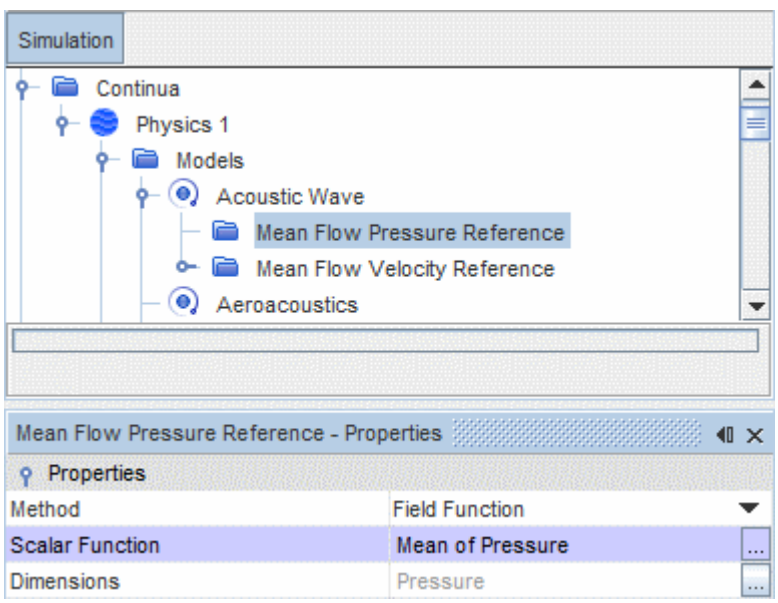
Property	Value
Field Function	**Pressure**

2. For aeroacoustic simulations using convective effects, set up **Field Mean** monitors for the X, Y, and Z components of the velocity:
   a) Right-click the **Monitors** node and then click **New Monitor** > **Field Mean**.
   b) Select the new **Field Mean** node and set the following properties:

Property	Value
Parts	[Flow Region]
Field Function	**Velocity** > **Lab Reference Frame** > **Laboratory** > **i**, **j**, or **k**

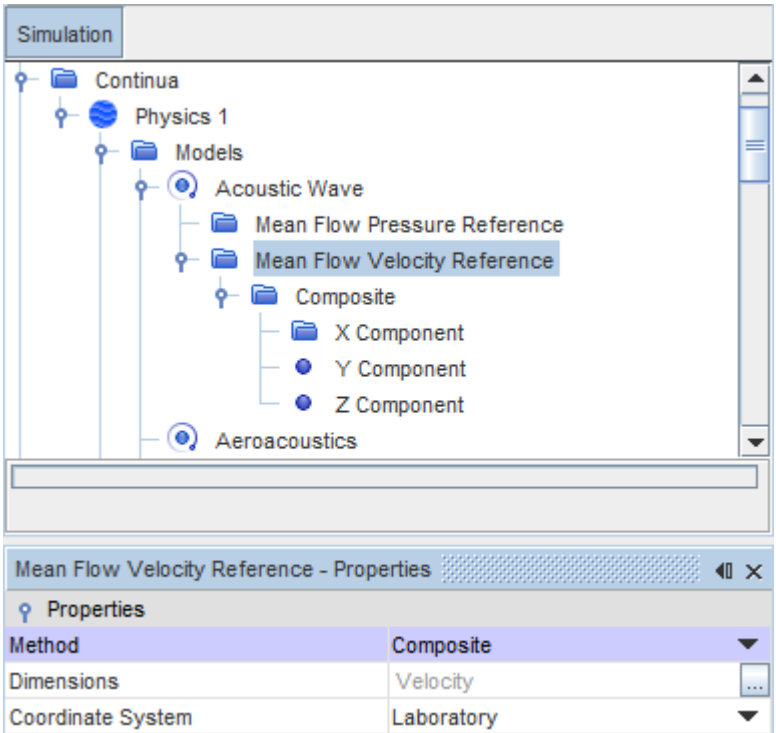
   c) Name the three resulting monitors, for example: `MeanVelocity X`, `MeanVelocity Y`, `MeanVelocity Z`, for **i**, **j**, and **k** respectively.
3. Run the flow simulation to obtain the fully developed solution.
4. Activate the Acoustic Wave model and, under the **Acoustic Wave** node, set the Mean Flow Reference property to **Specified**.

   This exposes the **Mean Flow Pressure Reference** and **Mean Flow Velocity Reference** child nodes.
5. To obtain the mean flow pressure field, set the **Mean Flow Pressure Reference** to use the **Mean of Pressure** field function.



6. To obtain the mean flow velocity field, use one of the following procedures:
   - Use the Composite method:
     1. Select the **Acoustic Wave** > **Mean Flow Velocity Reference** node and set the *Method* property to **Composite**.
     2. Expand the **Composite** node to expose the **X Component**, **Y Component**, and **Z Component** sub-nodes.

3. In each sub-node, using the *Method* property, specify the value for each component of the velocity.



- Define a field function combining the individual mean velocity components. This example is based on the monitors that were defined previously:

  ```
  [${MeanVelocityXMonitor},${MeanVelocityYMonitor},$
  {MeanVelocityZMonitor}]
  ```

Computing the Mean Flow Pressure and Velocity Fields without Monitors

In the absence of monitors, you can compute the mean pressure and velocity fields with the Acoustic Wave solver already set up to run.

To ensure stable mean fields, specify a Start Time delay for the Acoustic Wave solver. The time that is required to obtain stable mean pressure and velocity fields depends on the simulation parameters, particularly the geometry and the flow velocity. A reasonable guideline is to allow a delay time of ten flow-through periods.

To compute the mean pressure and velocity fields:

1. In the **Acoustic Wave** model, set the Mean Flow Reference property to **Compute**.
2. In the **Acoustic Wave** solver, set the appropriate Start Time.

Defining Filtering and Damping Functions

Create the appropriate field functions to define the noise source filtering and acoustic damping zones.

- The Acoustic Wave solver uses a field function ($0 \leq f_{source} \leq 1$) to specify the noise source or act as a filter to specify the region of interest in the flow.
- Physical damping is needed in the Acoustic Wave solver, since the CFD mesh typically becomes coarse away from region of interest. The Acoustic Wave solver requires a field function ($0 \leq f_{damp} \leq 1$) to define the damping region that is used to avoid spurious acoustic reflections from mesh transition.

See Setting Up an Acoustic Perturbation-Based Noise Source and Reducing Spurious Acoustic Reflections.

Acoustic Wave Analysis

This section describes the acoustic wave simulation workflow.

The acoustic wave simulation workflow is as follows:

1. Activate the Acoustic Wave model.

   See Selecting the Acoustic Wave Model.
2. Specify how the Mean Flow Pressure is obtained.

   See Preliminary Analysis.
3. Set up the noise source to use in the model.

   See Acoustic Wave Conditions and Values.
4. If necessary, set the boundary conditions in the flow region.

   The Acoustic Wave model supports the following boundary types: Wall, Symmetry Plane, Free Stream, Inlet, Outlet, and Periodic.

   See Acoustic Wave Conditions and Values.
5. Set up the Acoustic Wave solver.
   - Use implicit time marching with sufficient inner iterations to ensure convergence.
   - If you are computing the mean flow pressure value, you must specify the appropriate start time delay.
   - To suppress instabilities in the simulation due to spurious effects:
     - Set *Temporal scheme* to **Standard Newmark**, then select the **Newmark Delta Parameter** node and set the value to slightly higher than the default, for example 0.502. This can help suppress instability issues with spurious high-frequency content.
     - If using Standard Newmark does not work, change *Temporal scheme* to **Newmark Alpha**, then select the **Newmark Alpha Parameter** node and set the value slightly lower than the default of -0.05. (The allowed range is $-0.33 \leq \alpha \leq 0$.)

   See Acoustic Wave Solver Properties.
6. Set up the necessary stopping criteria, derived parts, and reports: the usual options to create probes for acoustic pressure monitoring in the region of interest.
7. Run the acoustic wave simulation (with the relevant flow and Acoustic Wave models activated) to obtain the acoustic pressure values throughout the flow region.

Post Processing

Output the acoustic data and compare to other simulation results.

Perform the following tasks:

1. Output the acoustic data in a `.trn` file.

   Use the **File** > **Auto Export** option to export the simulation results. For more information, see Exporting Surface Spectral Data.

2. Perform post-processing of the acoustic data to compare the simulation results against other data.

   STAR-CCM+ provides built-in data set functions, such as Fourier Transforms, that you can use to apply signal processing to the acoustic data.

   For more information, see Using Data Set Functions.

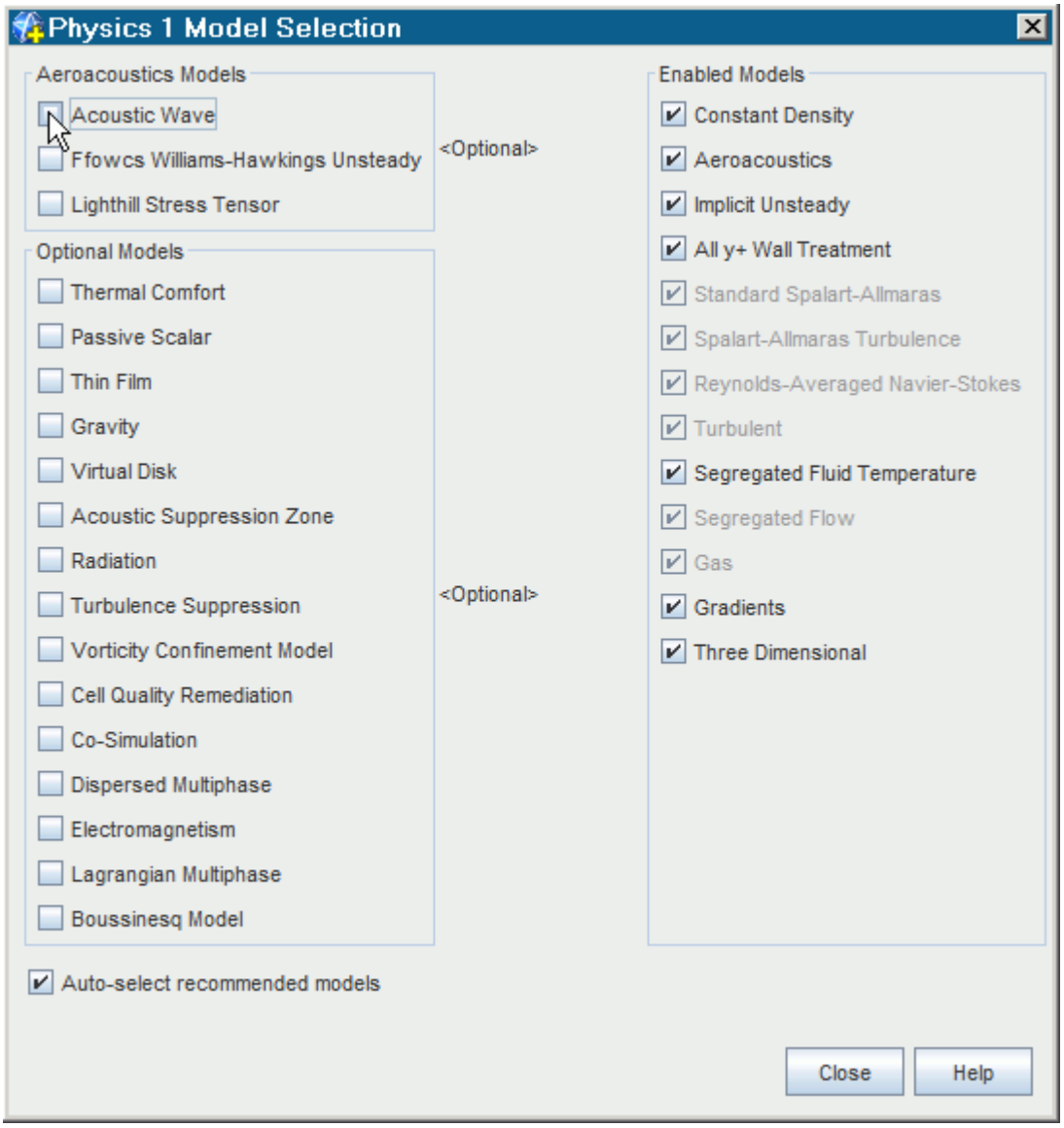
Selecting the Acoustic Wave Model

The Acoustic Wave model is available for three-dimensional unsteady simulations, and with any viscous regime.

The Acoustic Wave model is not compatible with Explicit, Two-Dimensional, or Axisymmetric models. The Acoustic Wave model is compatible with stationary cases only - the simulation cannot include any moving regions (such as fan blades).

Note: When you use the Acoustic Wave model, it is recommended that you also activate the Cell Quality Remediation model.

When you select the Aeroacoustics model from the *Optional Models* group box, the **Acoustic Wave** option becomes available in the *Aeroacoustics Models* group box.



Contents:

Acoustic Wave Model Properties

Acoustic Wave Model Properties

The **Acoustic Wave** node represents the model in the physics continuum.

Acoustic Wave Model Properties

Far-Field Sound Speed

The sound speed in the far-field. See Eqn. (3885). The default value is 340 m/s.

Noise Source

Defines the source of acoustic pressure. The specified source is used as the right-hand side of the Acoustic Perturbation Equation (APE) based noise source equation (Eqn. (3942)).

None

No noise source is defined in the flow region. You use this option when you model the acoustic waves that pressure fluctuations generate at a boundary wall.

Acoustic Perturbation Based Source

Models the acoustic waves that the flow pressure fluctuations generate in the fluid flow. This method is the recommended option for hybrid flow and aeroacoustics simulations. For more information, see Setting Up an Acoustic Perturbation-Based Noise Source.

User-Specified Source

Models a user-defined pressure field that is applied to a single cell and then propagates into the domain. This option is used for testing purposes. For more information, see Setting Up a User-Defined Noise Source.

High Order APE Sources

When **On**, this property activates fourth-order accurate noise source computation. When this property is **Off**, the default, accuracy is second-order.

Enable Convective Acoustic Effects

When **On**, this property includes convection in the simulation. See Acoustic Wave Model theory.

Low-pass Space Filtering

When **On**, the model uses a narrow-band, low-pass, space-filtered value for pressure fluctuations in individual cells. See Eqn. (3946). This option also adds the **Low-Pass Space Filter Coefficient** node under **Physics Values** for all regions. This coefficient defines the fraction of the filtered value that is used by the solver. The default value of the coefficient is 1.0.

Low-pass Time Filtering

Specifies the time filter that is used by the model. Each filter gives the model a low-pass, time-filtered value for the pressure perturbation term on the right-hand side of Eqn. (3941). Options other than **None** also add the **Low-Pass Time Filter Cutoff Frequency** node under **Physics Values** for all regions. The default value of the frequency is `1E20 /s`.

None

No time filter used.

1st-order IIR

This filter does the least signal damping of the three time filters, and introduces the least phase error. See Eqn. (3943).

2nd-order Butterworth

This filter produces a useful amount of signal damping and only moderate phase error. It is the recommended filter. See Eqn. (3944).

3rd-order Butterworth

This filter gives the most signal damping of the three time filters, but it also produces more phase error. See Eqn. (3945).

Mean Flow Reference

Specifies the method by which the mean pressure and velocity fields are obtained. For more information, see Preliminary Analysis.

Specified

Use the pressure field that is specified in the **Mean Flow Pressure Reference** node. When *Enable Convective Acoustic Effects* is **On**, use the velocity field that is specified in the Mean Flow Velocity Reference node. These values are typically the **Mean of Pressure** and **Mean of Velocity** field functions that are obtained from a fully developed flow solution. For more information, see Preliminary Analysis.

Compute

Compute the mean pressure field with the Acoustic Wave solver already set up to run. With this option, the mean flow calculation starts at time 0 regardless of the time that is specified by the *Start Time* property of the Acoustic Wave Solver. For more information, see Start Time.

Secondary Gradients

Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces. You can use this option to avoid solution divergence in cases where the Cell Quality Remediation model does not detect and treat bad cells.

On

Include both secondary gradients.

Off

Exclude both secondary gradients.

Interior Only

Include the interior secondary gradients only.

Boundaries Only

Include the boundary secondary gradients only.

Evaluate Total APE Fields For Post FW-H

When **On**, the model calculates the following additional fields and makes them available as field functions:

- **APE Total Density**, flow density + acoustic density. See Eqn. (3952).
- **APE Total Velocity**, flow velocity + acoustic velocity. See Eqn. (3953).

- **APE Total Pressure**, flow pressure + acoustic pressure. See Eqn. (3954).

Store these variables in the `.trn` file for post-processing with the Post FW-H model. You can use this option to combine the Acoustic Wave model with far-field sampling, without the high computational cost of a large dense mesh.

Clearing Acoustic Data from the Simulation

When you clear acoustic data, the acoustic pressure and all of the related variables that are calculated in the Acoustic Wave solver are removed from the simulation.

The mean flow pressure is reset to the field that was originally specified. The flow and pressure data that was calculated by the flow solver is not affected.

To clear the acoustic data from a simulation:

- Right-click the **Acoustic Wave** model and then click **Clear Solution**.

Setting Up the Noise Source for the Acoustic Wave Simulation

You can specify the noise source in the Acoustic Wave simulation in three ways.

You can specify the noise source in the Acoustic Wave simulation in the following ways:

- Acoustic perturbation-based noise source

  Models the acoustic waves that are generated by the fluid flow in the physics continuum.

  See Setting Up an Acoustic Perturbation-Based Noise Source.
- User-defined noise source

  Models the acoustic waves that are generated by a user-defined pressure field in a specified location in the flow region.

  See Setting Up a User-Defined Noise Source.
- Wall pressure fluctuation noise source

  Models the acoustic waves that are generated by user-defined pressure fluctuations at a boundary wall.

  See Acoustic Wave Conditions and Values.

Contents:

Setting Up an Acoustic Perturbation-Based Noise Source
Setting Up a User-Defined Noise Source

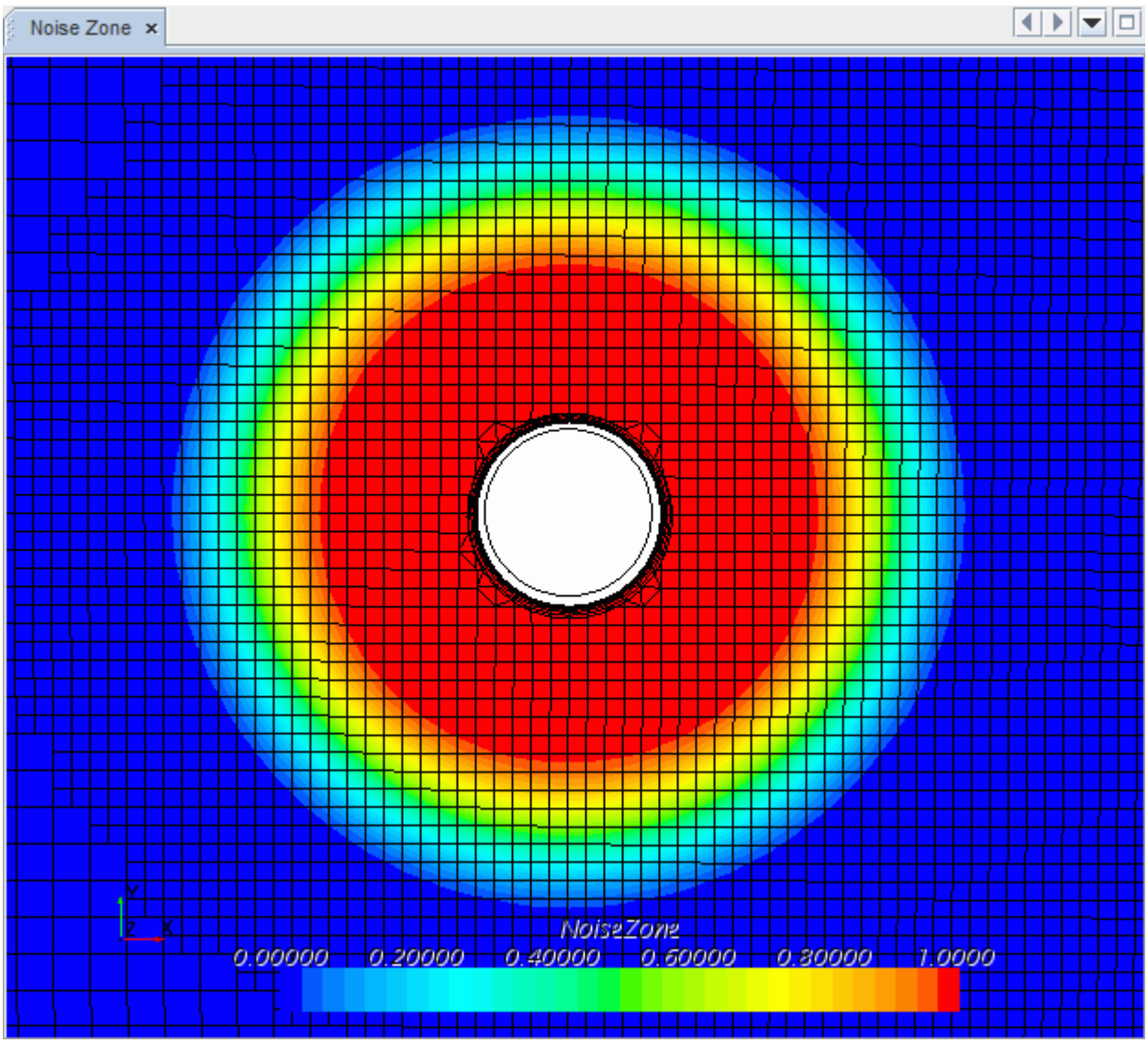
Setting Up an Acoustic Perturbation-Based Noise Source

In the Acoustic Wave method, you define a particular volume within which you expect most of the noise generation to occur. Outside the noise source volume, you do not expect the flow to generate any appreciable noise. You typically use a field function to specify the volume of interest and to filter out the quiescent region.

This field function for the noise source zone must return a value $0 \leq f_{source} \leq 1$. You can apply a transition modifier, such as a Hanning window function Eqn. (542), to provide a smooth transition between the value of 1 in the interior of the noise source zone, and 0 on the outside.

Note: Do not use the name APE Source as the name for the noise source field function. This name is used by an existing field function.

An example noise source field function is illustrated below.



The noise source is the region around the cylinder. In this example, there is a strong left-to-right airflow that generates acoustic waves by vortex shedding from the cylinder. Note the fine mesh throughout the noise source zone.

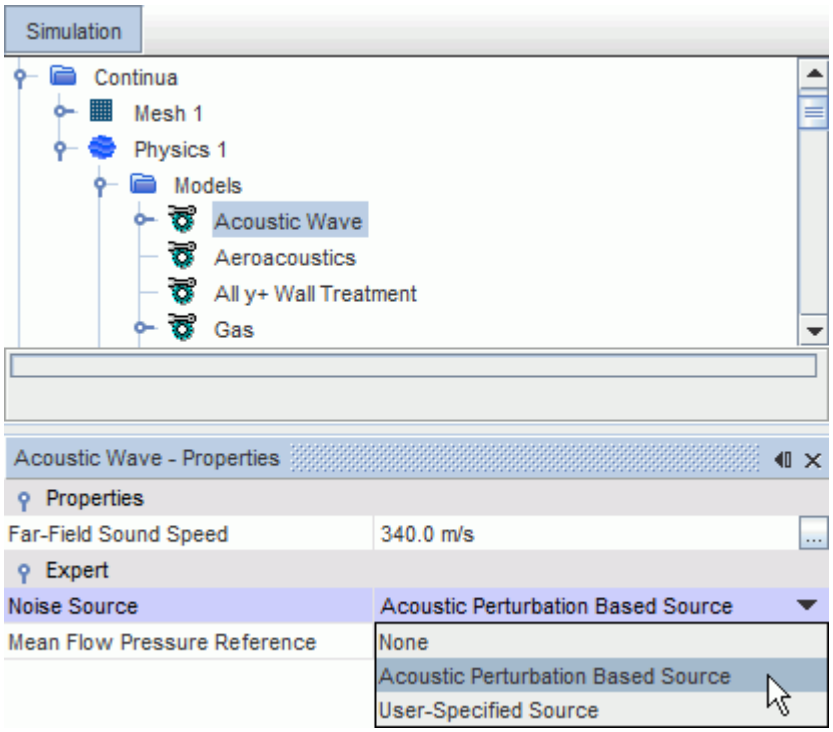
To set up an acoustic perturbation-based noise source:

1. Create the appropriate Noise Source Filter field function (f_{source}) to define the noise source enclosure space.

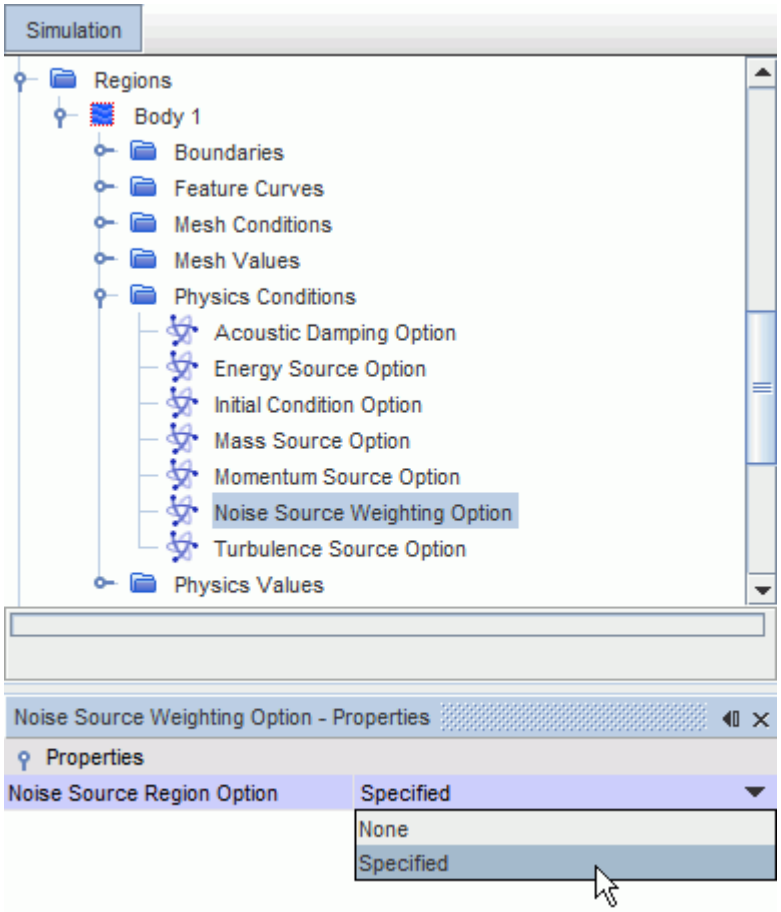
You can optionally create the appropriate Acoustic Damping field function (f_{damp}) to suppress acoustic reflections from outside the area of interest.

For more information, see Reducing Spurious Acoustic Reflections.

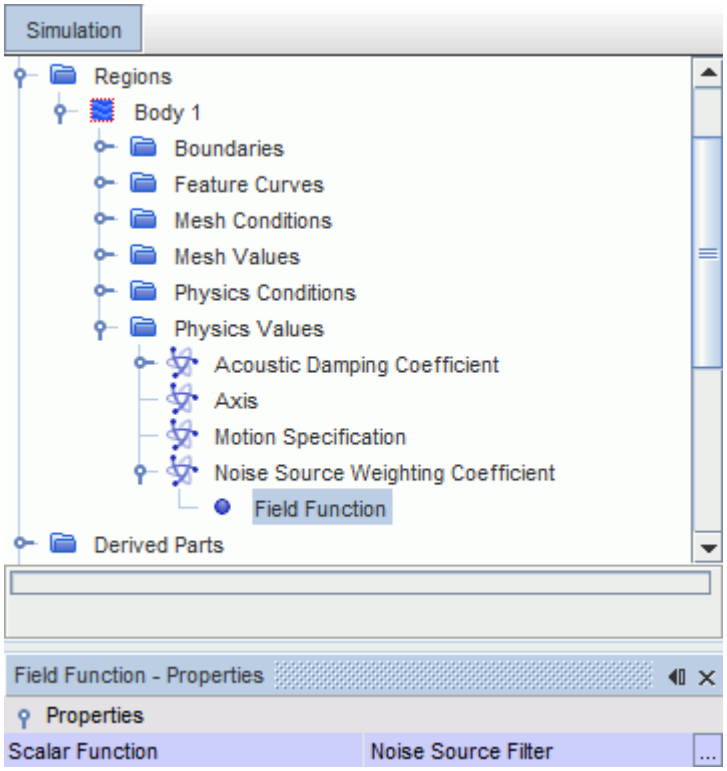
2. In the physics continuum, select the **Acoustic Wave** model and set the *Noise Source* property to **Acoustic Perturbation Based Source**.



3. Under the **Regions** node, expand the appropriate region. In the Physics Conditions for the region, select the **Noise Source Weighting Option** node and set the *Noise Source Region Option* property to **Specified**.

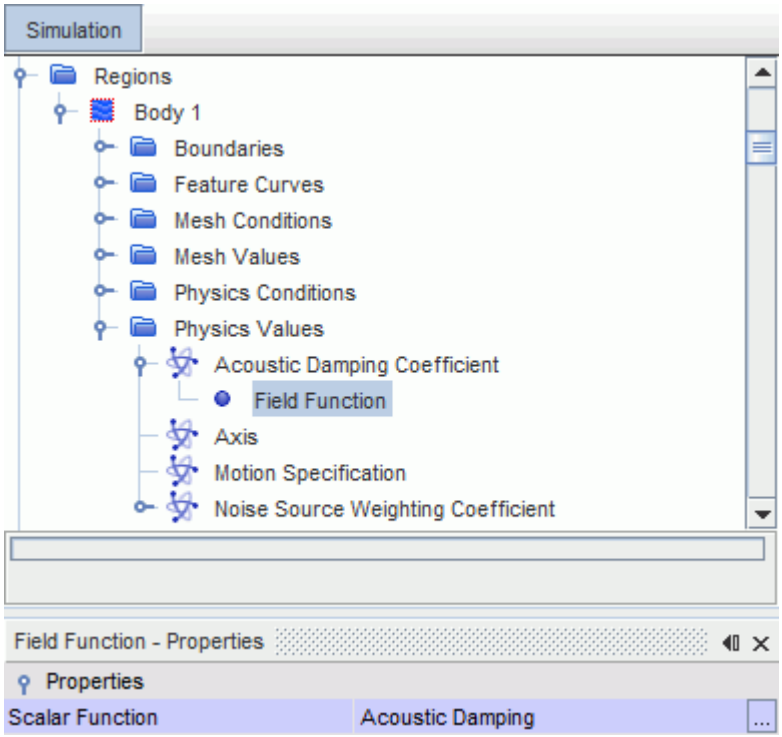


4. In the Physics Values for the region, select the **Noise Source Weighting Coefficient** node and set the *Method* property to **Field Function**.
5. Select the **Field Function** node and set the *Scalar Function* property to the *Noise Source Filter* field function that you created earlier.



If you want to apply damping to the region outside the area of interest, perform the following steps:

6. In the Physics Conditions for the region, select the **Acoustic Damping Option** node and set the *Acoustic Damping Option* property to **Specified**.
7. In the Physics Values for the region, select the **Acoustic Damping Coefficient** node and set the *Method* property to **Field Function**.
8. Select the **Field Function** node and set the *Scalar Function* property to the *Acoustic Damping* field function that you created earlier.



Setting Up a User-Defined Noise Source

You can create a user-defined noise source for testing or calibrating the model. You specify the source pressure field for a particular cell (or cells) in the flow mesh. The pressure field then propagates into the flow region.

Monopole (the pressure field is defined for a single cell), dipole (two cells), and quadrupole (four cells) noise sources are typical examples of user-defined noise sources.

For example, the analytical expression of the monopole source is:

$$P = A\frac{R}{r}\sin(\omega t - kr) \tag{540}$$

where:

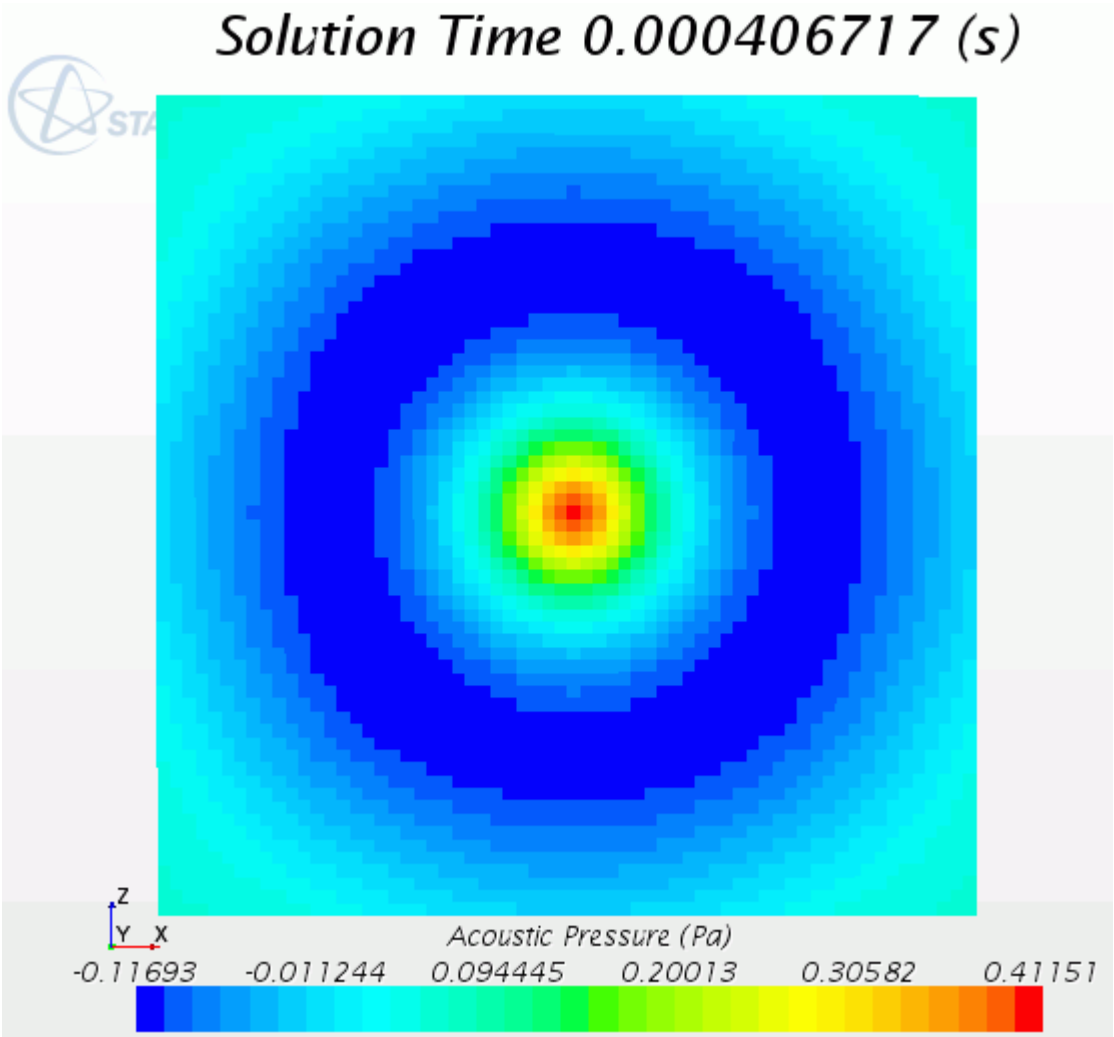
- A is the amplitude.
- R is the radius.
- $\omega = 2\pi f$ and f is the frequency of the source.
- r is the radial location in the domain.
- $k = \omega / c$ and c is the speed of sound.

The corresponding field function that you need to apply for the monopole source is:

$$\text{Input Pressure} = 4\pi AR\sin(\omega t) \tag{541}$$

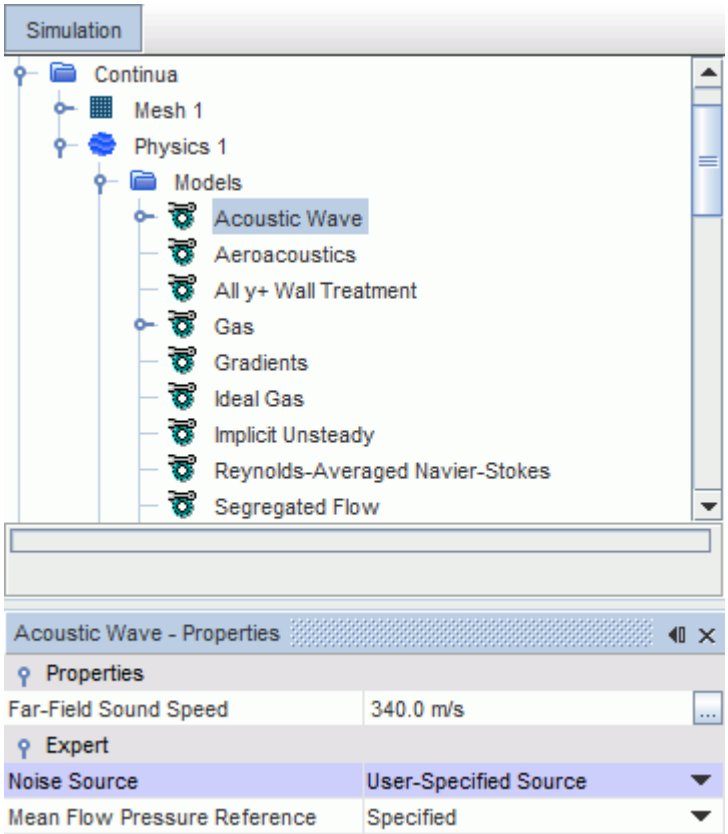
Ideally, the monopole source should be a sphere or a point source, but this is not possible with a finite volume mesh. For this reason, expect to see some difference between the analytical and computational.

The acoustic pressure field that is generated by the monopole source is illustrated below:

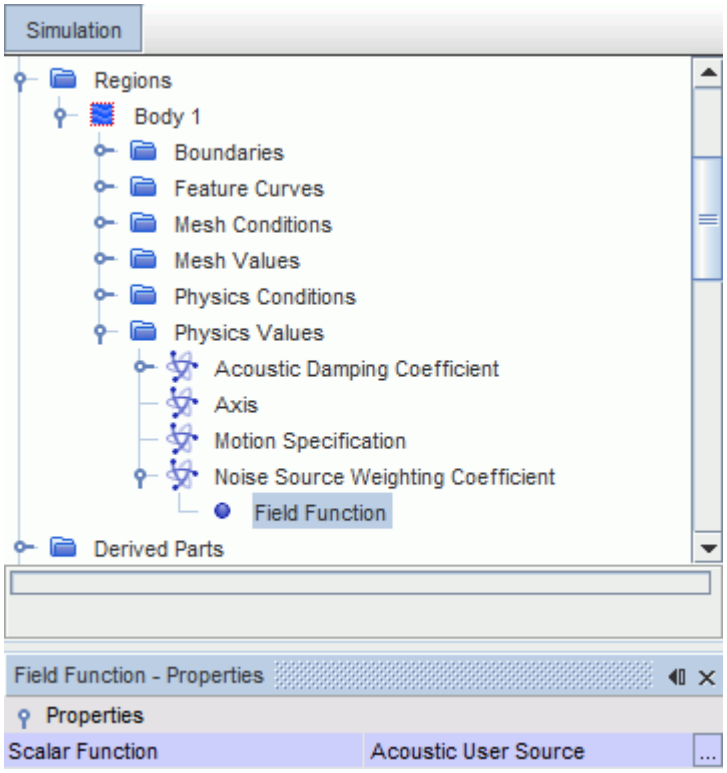


To set up a user-defined noise source:

1. Create the appropriate *Acoustic User Source* field function.
2. In the physics continuum, select the **Acoustic Wave** node and set *Noise Source* to **User-Specified Source**.



3. Select **Regions** > **[flow region]** > **Physics Conditions** > **Noise Source Weighting Option** and set *Noise Source Region Option* to **Specified**.
4. Within the **[flow region]** > **Physics** > **Values** node, select **Noise Source Weighting Coefficient** and set *Method* to **Field Function**.
5. In the **Field Function** node, set the *Scalar Function* to the *Acoustic User Source* field function.



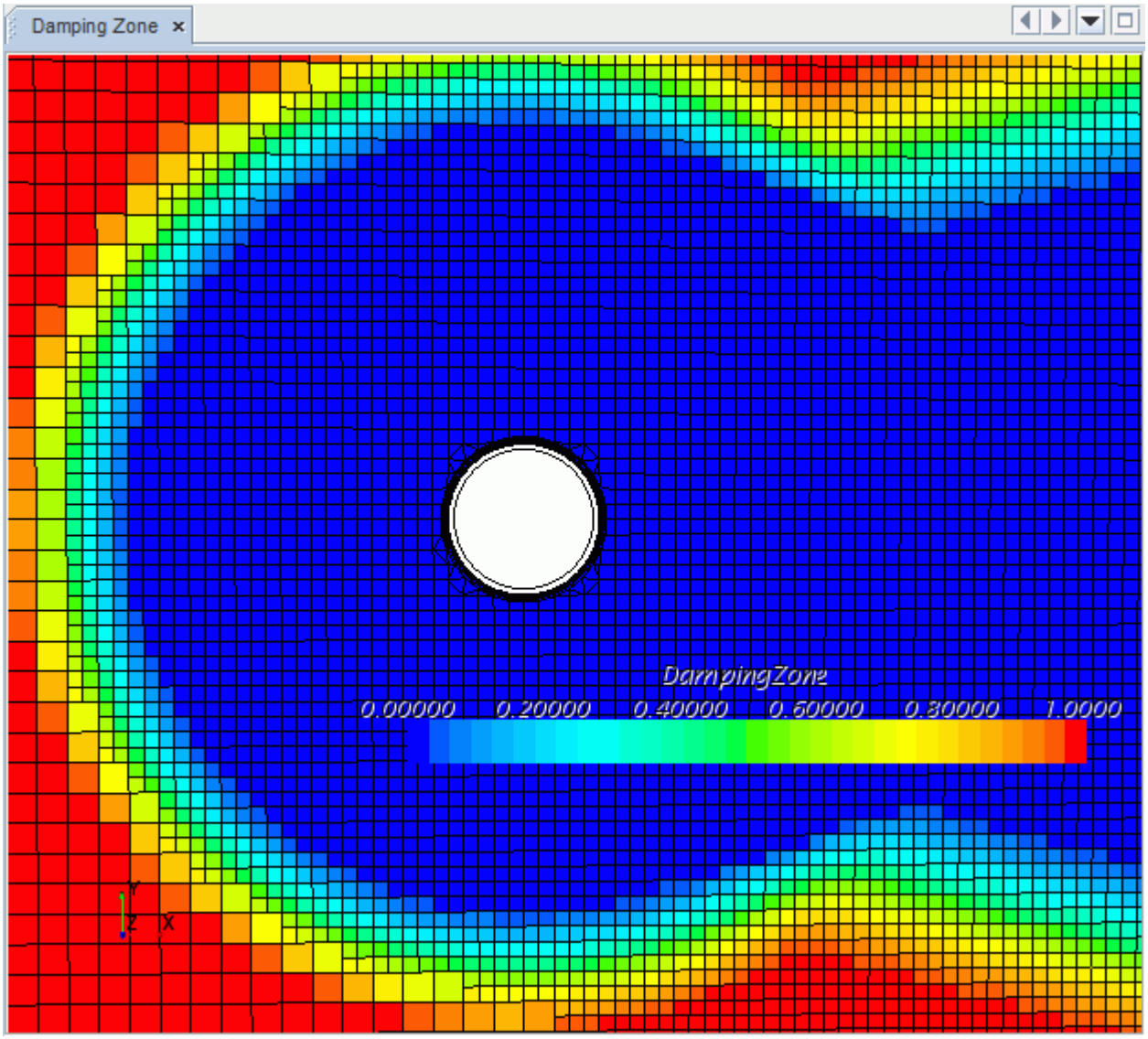
Reducing Spurious Acoustic Reflections

This section describes how to suppress acoustic reflections from outside the noise source zone.

For practical applications, the flow mesh is typically coarsened away from the region of interest. A strong vortical flow crossing through this mesh transition in the downstream region can generate spurious acoustic waves that propagate upstream. These spurious acoustic waves interfere with those inside the noise source zone, resulting in erroneous results.

You typically use a field function (f_{damp}) to suppress acoustic reflections from outside the noise source zone. This field function must return a value of 0 within the noise source zone, and a value of 1 at the boundaries of the flow domain. Within the transition from 0 to 1, you can apply a Hanning windowing modifier Eqn. (542) to give a smooth transition.

An example of a damping field is illustrated below.



In this example, there is a strong left-to-right air flow which generates acoustic waves by vortex shedding from the cylinder. The region of interest (in blue) is undamped and everywhere else is fully damped (in red). Note the fine mesh throughout the undamped zone and the damping that is applied at the mesh transitions.

The recommended points per wavelength (PPW) must be maintained inside the undamped zone.

For more information on the recommended PPW, see Guidelines for Using the Acoustic Wave Model.

Acoustic Wave Conditions and Values

If necessary, depending on your geometry, you can set acoustic boundary conditions at walls and symmetry planes.

Initial Conditions for the Acoustic Wave Model

When you activate the Acoustic Wave model, the following nodes are added under the Initial Conditions node in the physics continuum:

Acoustic Pressure	The value p^a that is used in the Acoustic Perturbation Equation (Eqn. (3942)).
Acoustic Pressure Second Time Derivative	The second time derivative of acoustic pressure, $\frac{\partial^2 p^a}{\partial t^2}$. This value is used for the Newmark solver.
Acoustic Pressure Time Derivative	The first time derivative of acoustic pressure, $\frac{\partial p^a}{\partial t}$. This value is used for the Newmark solver.

You do not need to set these initial conditions in a typical acoustic wave simulation. However, you can use particular settings for testing purposes.

For more information on the corresponding field functions, see Acoustic Wave Solver Field Functions Reference.

Boundary Conditions for the Acoustic Wave Model

When you activate the Acoustic Wave model, the following acoustic boundary conditions become available:

Wall Acoustic Pressure Specification	Applies to Wall boundaries only.	
	None	The wall boundary has no specific acoustic pressure but is a reflecting boundary.
	Specified	The acoustic pressure is specified in the **Physics Values** > **Prescribed Acoustic Pressure** node to set the value of the pressure at the wall directly.
	Non-reflecting	The wall is a non-reflecting boundary. Acoustic waves pass through the boundary.
	Partially Absorbing	The acoustic pressure is specified in the **Physics Values** > **Partial Wall Absorption Coefficient** node to set the fraction of the pressure absorbed at the wall to a value between 0 and 1.
Symmetry Acoustic Pressure Specification	Applies to Symmetry Plane boundaries only. By default the boundary is reflecting, but you can make it non-reflecting if necessary.	
	Non-reflective Boundary Condition Treatment	• Activated

		The symmetry plane boundary is non-reflecting. Acoustic waves pass through the boundary. • Deactivated The symmetry plane boundary is a reflecting boundary.

Acoustic Wave Solver Properties

The Acoustic Wave solver is activated when the Acoustic Wave model is selected. Similar to the flow solvers, the Acoustic Wave solver uses the AMG Linear Solver infrastructure that is provided in STAR-CCM+.

The **Acoustic Wave** node represents the Acoustic Wave solver in the **Solvers** tree.

Acoustic Wave Solver Properties

Under-Relaxation Factor

The under-relaxation factor is set to 1.0 and cannot be modified. It is not appropriate to use under-relaxation, as that would be equivalent to applying damping to the simulation.

Start Time

The elapsed simulation time before the solver begins calculating.

If the Mean Flow Pressure Reference property in the Acoustic Wave model is set to **Compute**, use the *Start Time* property to specify a start time delay, to minimize initial transient flow effects into the mean flow fields. The time that is required depends on the simulation parameters, particularly the geometry and the flow velocity. A reasonable guideline is to allow a delay time of ten flow-through periods. By default, the solver starts with a five time-step delay (if the start time is set to 0.0 s) in order to pre-compute the noise source time derivative.

Temporal scheme

Allows you to select the time scheme that best suppresses spurious frequencies and promotes stability.

Standard Newmark

The default. This selection exposes the Newmark Delta Parameter node, where you specify a parameter value for the Newmark-delta method of numerical integration that is used in the Acoustic Wave solver. The default value of 0.5 is suitable for most cases. However, to suppress spurious high frequency content and simulation instability due to mesh imperfections, you can set this value to slightly higher than the default. For example, 0.502 or higher, as required to stabilize the solution. Some trial and error can be required to determine the appropriate value.

Newmark Alpha

This selection exposes the Newmark Alpha Parameter node, where you specify a parameter value for the Newmark-alpha method of numerical integration that is used in the Acoustic Wave solver. The default

value is -0.05. Use this option to suppress spurious high frequencies if the **Standard Newmark** option does not eliminate spurious effects. Note, however, that this option sometimes produces instabilities when there are bad cells in the mesh.

Solver sub-time stepping

When **On**, the acoustic wave solver uses a smaller time-step size than the flow solver. This substepping allows the flow solver to run with a time-step higher than the numerical limit of the acoustic wave solver allows. The inner time-step for the acoustic wave solver is set by dividing the flow time-step by the *Number of sub-time steps* property. Noise sources from the flow are interpolated within the inner time-steps using either second-order or fourth-order accurate computations, depending on the setting of *High Order APE Sources* on the Acoustic Wave model.

Number of sub-time steps

The number of iterations that are used by the *Solver sub-time stepping* property.

Sub-time stepping convergence tolerance

Sub-time stepping allows the Acoustic Wave solver to perform time integration with smaller time-steps than those used by the flow solver. This improves the accuracy of the acoustic pressure solution. For the Wave Equation solver, the inner iterations convergence during sub-time stepping (the tolerance) needs to be enough to preserve the desired level of accuracy without causing an unnecessary increase in CPU cost. Typically, a value 3–4 orders of magnitude normalized to the Wave Equation solver residual per sub-time step is sufficient.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Guidelines for Using the Acoustic Wave Model

This section contains recommendations for using the Acoustic Wave model.

- When you use the Acoustic Wave model, you are advised to activate the Cell Quality Remediation model.
- Use a high-resolution turbulence model to ensure good noise source predictions.
- Use the appropriate mesh resolution in the zone of interest.

  The mesh resolution must be based on the recommended points per wavelength (PPW) to maintain the accuracy of acoustic predictions. PPW = (speed of sound/frequency) / grid spacing.

  You are advised to use PPW ≥ 30–40, and this value must be maintained throughout the zone of interest. The recommended Acoustic CFL depends on the PPW: for PPW of 40, you are advised to set Acoustic CFL ≤ 4 inside the zone of interest. However, for higher PPW values, a higher Acoustic CFL is acceptable.
- Apply damping to improve acoustic prediction. There are two damping methods that you can use to improve simulation stability:
  - Numerical damping that is applied to the Acoustic Wave solver

    The implementation of the Acoustic Wave model takes into account the solver robustness, but some instability can still arise due to mesh imperfections. Over time in a long simulation, localized hot spots of acoustic pressure can appear. These hot spots can lead to the accumulation and spreading of instability.

    To suppress simulation instability due to spurious effects, you can use the Temporal scheme property to adjust the Newmark parameters in the Acoustic Wave solver.
  - Physical damping to eliminate spurious acoustics from mesh transition or boundary effects

    In order to eliminate acoustic wave reflections into the area of interest, you can apply a damping function (f_{damp}) to the region outside the area of interest. The reflections are attenuated and their effects are minimized.

    For more information, see Reducing Spurious Acoustic Reflections.
- Apply APE source filtering to remove possible high frequency spurious content from the noise sources. The low-pass time filter is more effective for this purpose than space filtering. 1st-, 2nd-, or 3rd-order time filters can be used with increasing effect of numerical damping. However, higher-order filters introduce phase errors in the acoustic solution.
- When you specify field functions to define the noise source filtering (f_{source}) and acoustic damping (f_{damp}) zones, a Hanning windowing is recommended for the transition from a value of 0–1.

  The Hanning function has the general form:

$$\mathrm{H}(r) = 0.5\left(1-\cos\left(\pi\left(\frac{\mathbf{r}}{\mathbf{t}}\right)\right)\right) \tag{542}$$

  where:
  - $\mathbf{r}$ is the direction vector
  - $\mathbf{t}$ is the thickness
- A hybrid approach allows you to benefit from the acoustic wave solver and still propagate acoustics to a distant far-field observer. Achieve this goal by using the acoustic wave solver in the near field and propagating the acoustics using the Ffowcs Williams-Hawkings (FW-H) model. To do so:
  1. Activate the Acoustic Wave model property Enable Total APE Fields for Post FW-H. This makes the following field functions available:
     - **APE Total Density**, flow density + acoustic density. See Eqn. (3952).

- **APE Total Velocity**, flow velocity + acoustic velocity. See Eqn. (3953).
- **APE Total Pressure**, flow pressure + acoustic pressure. See Eqn. (3954).

2. Use **File** > **Auto Export** to export a selection of these functions to a `.trn` file.
3. Select receivers under **Post FW-H Receivers** and set the *Acoustic Data Source* property to **APE**.
4. Run the Post FW-H Solver using the `.trn` file and selected receivers.

Acoustic Wave Solver Field Functions Reference

When the Acoustic Wave Solver has the *Temporary Storage Retained* property activated, some field functions are made available to the simulation.

Acoustic CFL in Wave Equation	The Acoustic CFL ($c\frac{\Delta t}{\Delta x}$, where c is the speed of sound) is computed in the whole domain, based on the mesh size and time step.
Acoustic Pressure	The Acoustic Pressure is the variable being solved.
Acoustic Pressure Gradient	The computed gradient fields using STAR-CCM+ gradient options.
Acoustic Pressure Recon	
Acoustic Pressure Residual	The residual on the current solution for acoustic pressure p^a in Eqn. (3939).
Acoustic Pressure Second Time Derivative	The fields that are used for the Newmark solver.
Acoustic Pressure Time Derivative	
Mean Flow Pressure	The mean flow pressure that is obtained from the flow reference.

Activating the following functions also requires that the *Noise Source* property of the **Acoustic Wave** model be set to **Acoustic Perturbation Based Souce**.

APE Pressure	The fluctuating flow pressure field.
APE Sources	The acoustic perturbation-based source term that is generated in the Acoustic Wave equation.
Mean Flow Velocity	The mean flow velocity that is obtained from the flow reference.

Using Direct Noise Calculation

Direct noise calculation, or the Computational Aeroacoustics approach, requires solving the entire flow field for the full unsteady simulation.

To calculate the aerodynamically generated acoustics, record the time-varying static pressure on points (probes) or surfaces. It requires high numerical resolution to capture the spatial and transient effects of the acoustic pressure perturbations. For post-processing, the Fourier Transform (FT) tools can be used for spectral analysis. (See the section on data set functions for details.).

Predicting noise levels near to the source of sound generation can be done directly within a CFD model. Direct simulation can be too expensive to predict noise levels some distance from the source, however. For such far-field predictions, STAR-CCM+ provides the Ffowcs Williams-Hawkings (FW-H) acoustics integral formulation.

Using the Acoustic Suppression Zone Model

Use the Acoustic Suppression Zone model to suppress spurious reflective disturbances in unsteady aeroacoustic simulations.

Running unsteady flow simulations produces time-dependent variations in the flow. These flow disturbances convect with the flow (for vortical disturbances) or propagate at the speed of sound (for acoustic disturbances) outward to exterior boundaries.

If the applied exterior boundary types have reflective properties, these disturbances can be spuriously reflected back into the domain. This reflection affects the flow solution, causing problems for acoustic-type calculations.

To satisfy the requirements for good quality aeroacoustic calculations, the numerical algorithm must allow these pressure waves and vortices to travel through the boundary smoothly. In other words, the algorithm must give the boundary quasi-non-reflective properties.

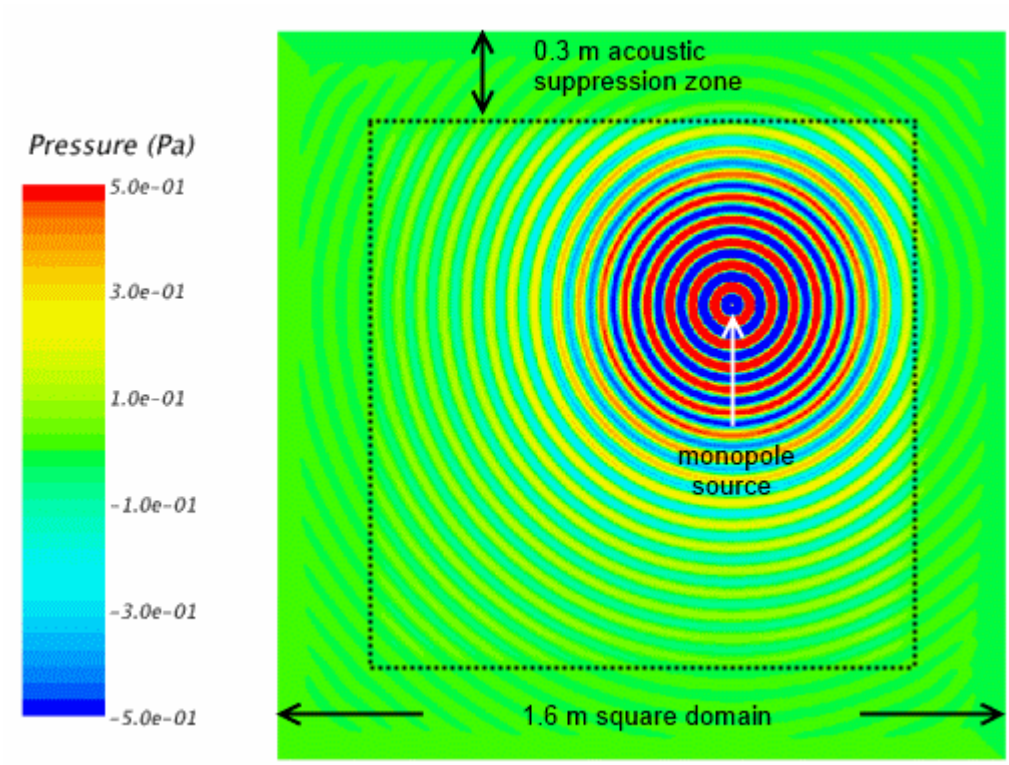
For example, a pressure outlet with non-reflecting properties allows the defined constant pressure to represent an average pressure at the outlet boundary. At the same time, it allows the flow pressure at the boundary to vary and disturbances to pass out of the domain without reflecting off the boundary.

A practical approach to constructing a non-reflective boundary treatment is to complement the respective characteristics-based boundary conditions with an Acoustics Suppression Zone treatment.

Use the Acoustic Suppression Zone when:

- Boundary disturbances are creating spurious acoustic effects.
- The spurious acoustic effects are propagating upstream.
- There is enough distance (at least several wavelengths) next to the boundary in which to suppress the spurious acoustic effects.

The following image shows a monopole source which is radiating an instantaneous pressure field at 6800 Hz in a 1.6 m square domain. The simulation ran using the coupled solver. An Acoustic Suppression Zone 0.3 m deep was applied to the boundaries of the domain to suppress echoes.



The Acoustic Suppression Zone is compatible with both the coupled flow solver and the segregated flow solver.

With experience, you can use the Acoustic Suppression Zone model to model complex far-field conditions (for example, the simulation of wind gust effects on an airfoil), beyond the boundary conditions that are documented here. See Kim et al. [78].

Selecting the Acoustic Suppression Zone Model

To select the Acoustic Suppression Zone model, first select a flow model and an unsteady model. This model is optional, and is compatible with compressible ideal gas and real gas models.

Acoustic Suppression Zone Model Properties

Verbose	Displays the operations that are performed to compute *Continuity*, *Momentum*, or *Energy source*.	
	Activated	Displays operations
	Deactivated	Does not display operations.
Continuity source	Specifies $-\sigma\left(\rho-\rho_{ref}\right)$ as a damping term. See Eqn. (3956).	
	Activated	Applies acoustic damping to the **Continuity source**.
	Deactivated	Does not use **Continuity source** damping term.

Momentum source	Specifies $-\sigma\left(\rho v - \rho_{ref} v_{ref}\right)$ as a damping term. See Eqn. (3956).	
	Activated	Uses the **Momentum source** damping term.
	Deactivated	Does not use **Momentum source** damping term.
Energy source	Specifies $-\sigma\left(\rho E - \rho_{ref} E_{ref}\right)$ as a damping term. See Eqn. (3956). Avoid using the **Energy source** with the segregated solver.	
	Activated	Uses the **Energy source** damping term.
	Deactivated	Does not use **Energy source** damping term.
Flow Conditions Option	Specifies the flow conditions within the Acoustic Suppression Zone.	
	Specified Per-Region	Adds three nodes under **Physics Values** for the region: **Acoustic Suppression Zone Pressure**, **Acoustic Suppression Zone Velocity**, and **Acoustic Suppression Zone Temperature**. In each of these nodes, you can specify the values by **Constant**, **Field Function**, **Table**, or **User Code**. This method is the default.
	Computed Average	The relaxation coefficient is defined as the average per boundary of pressure, temperature, or velocity, for all boundaries using Acoustic Suppression.
Relaxation Coefficient Option	Specifies the relaxation coefficient σ used in Eqn. (3956).	
	Use Boundary Distance	Computes the relaxation coefficient σ , based on the boundary distance from the selected boundaries and the maximum specified value for the relaxation coefficient. Adds a node under the **Acoustic Suppression Zone** model node, called **Computed Relaxation Coefficient**, which lets you specify **Maximum Relaxation Coefficient** (default value, 0) and **Acoustic**

		Suppression Zone Thickness (default value, 0).
	Specify Per-Region	Specifies the relaxation coefficient σ for any selection of regions, positioning the Acoustic Suppression Zone anywhere in the domain. Adds a node under **Physics Values** for each region, the **Acoustic Suppression Zone Relaxation Coefficient** node. In each of these nodes, you can specify the values by **Constant**, **Field Function**, **Table**, or **User Code**.

When *Flow Conditions Option* is set to **Specified Per-Region**, set the values for p, v, and T to the local flow mean values of the unsteady flow field. The pressure p and temperature T are used to compute the reference density ρ_{ref}. The model assumes that a statistically steady-state flow is achieved before starting to accumulate the flow-mean values.

The Acoustic Suppression Zone model is unlikely to work properly with ill-posed *Flow Conditions* values, or with aggressive or even discontinuous distributions (in space and time) of the relaxation coefficient σ. Set the Suppression Zone to at least 5–7 (representative) wave-lengths wide and to avoid numerical instabilities in the simulation, increase the *Maximum Relaxation Coefficient* gradually.

Acoustic Suppression Zone Guidelines

Determine the best values for pressure, velocity, temperature, relaxation coefficient, and maximum relaxation coefficient:

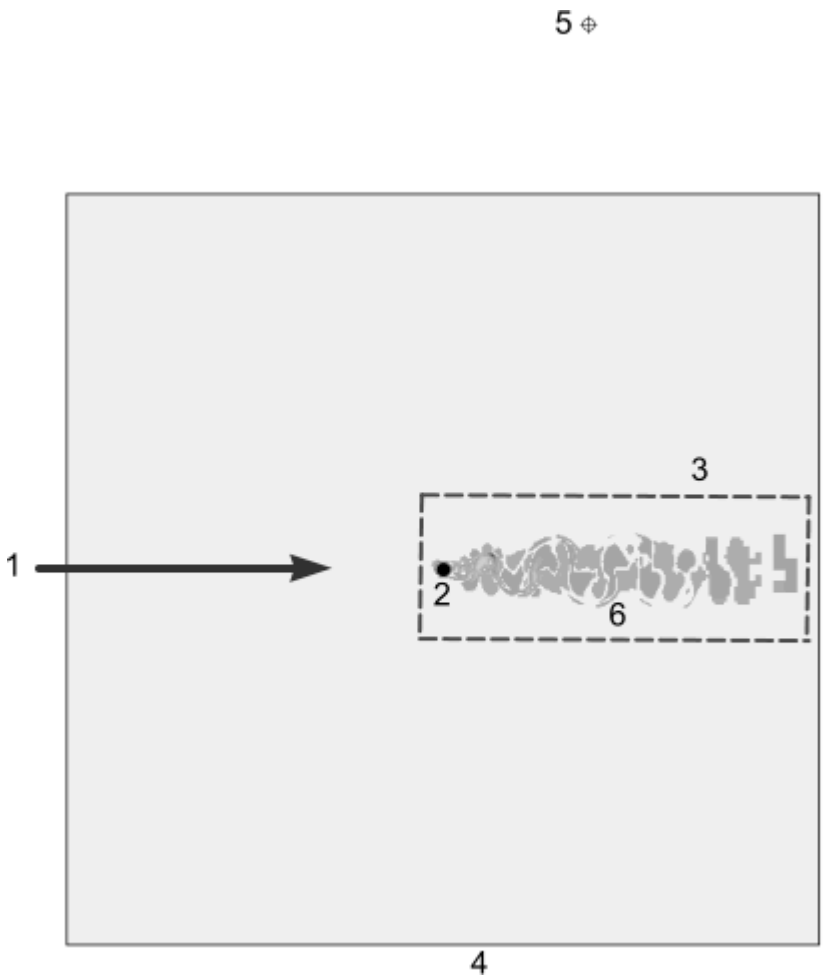
- Before running the simulation with the Acoustic Suppression Zone model, run it using the Steady model for time unless mean values are already known.
- Use mean values of pressure, velocity, and temperature from the steady time run to supply the values for **Physics Values** > **Acoustic Suppression Zone Pressure**, **Velocity**, and **Temperature**. See *Flow Conditions Option*.
- Make sure the relaxation coefficient σ (Eqn. (3956)) varies smoothly in space.
- Make sure the maximum relaxation coefficient σ_{max} ramps up smoothly in time.
- The best value for the maximum relaxation coefficient varies strongly depending on the simulation design, but values of several hundred often work well, together with values of Acoustic Suppression Zone Thickness in the range of 3 to 5 wavelengths.
- STAR-CCM+ recalculates Acoustic Suppression Zone distances immediately, whenever the values are changed. Since this calculation is time-consuming, change the values only when necessary. The distances are stored in the `.sim` file, so, if they are not changed, STAR-CCM+ makes no recalculation even if the simulation is re-started.

Using the Ffowcs Williams-Hawkings Aeroacoustics Models

Modern CFD adopts two strategies for prediction of far-field acoustics: one is based on the direct noise simulation of sound propagation and the other is based on integral formulations.

Both approaches are known Computational Aero-Acoustics (CAA) methods, and have been extensively used to study the sound generation and sound propagation in the fluid flow.

The Ffowcs Williams-Hawkings (FW-H) acoustics integral formulation is the preferred strategy for far-field noise prediction. This model is able to calculate the far-field sound signal that is radiated from near-field flow data from a CFD solution. The goal is to predict small amplitude acoustic pressure fluctuations at the locations of each receiver.



Index	Description
1	Flow direction
2	Cylinder surface / FW-H Impermeable source surface
3	FW-H Permeable source surface

Index	Description
4	Computational domain boundary / Limit of Direct Noise Simulation near-field solution
5	Working with FW-H Receivers in the far field
6	FW-H Volume Terms noise visualization

Contents:

The Ffowcs Williams-Hawkings Models
The Steady FW-H Workflow
The Ffowcs Williams-Hawkings Steady Aeroacoustics Model
The On-The-Fly FW-H Model
Working with FW-H Receivers
Working with FW-H Surfaces
Visualizing the FW-H Noise Sources
FW-H Field Functions Reference
The Post FW-H Model
Guidelines for Ffowcs Williams-Hawkings Modeling

The Ffowcs Williams-Hawkings Models

The Ffowcs Williams-Hawkings (FW-H) acoustic model is the preferred strategy for far-field noise prediction such as overhead aircraft noise.

This model is able to calculate the far-field sound signal that is radiated from near-field flow data from a CFD solution. The goal is to predict small amplitude acoustic pressure fluctuations at the location of each receiver.

The FW-H acoustic model for stationary motion can simulate wind tunnel aeroacoustic measurements with stationary sources such as aircraft landing gear or an airfoil trailing edge. The FW-H acoustic model is used only to predict the propagation of sound in free space. It does not include effects such as sound reflections, refraction, or material property change.

The Ffowcs Williams-Hawkings formulations are based on Farassat's Formulation 1A ([59], [66]) and Dunn Farassat Padula 1A ([76]), both for the non-convective form of FW-H. Therefore, numerical accuracy is not an issue of great concern compared to the full Computational Aeroacoustics approach, which explicitly resolves propagation of sound waves all the way to the far field. With the FW-H approach, a second order discretization scheme, both spatial and temporal, together with judicious choices for mesh resolution and time-step size, can give accurate far-field noise prediction.

For input, the FW-H model needs time-accurate data on the near field flow, corresponding to a fine mesh on the near field. The fine meshes used for accurate prediction of the boundary layer or other requirements for turbulence models (such as y+ for LES) are sufficient. The computational domain for this model does not need to include the location of the receivers. However, to use non-reflecting boundary conditions requires a larger domain so a buffer layer can be added to reduce and dissipate the acoustic waves.

The depth of the computational domain must be greater than, or equal to, the acoustic correlation length. This depth is sometimes different from the depth that is required for a good LES simulation. Currently, the FW-H

model uses the same solution domain for acoustic sources estimation. When three-dimensional CFD solutions with the FW-H model are periodic in the spanwise direction (for example an infinite cylinder), replicate the solution domain in the spanwise direction. For two-dimensional simulations, a correlation length can be used to overcome this issue.

The FW-H models are available to simulations with porous media. These models describe the propagation of sound in free space and are suited to simulations where porous regions do not affect sound propagation from FW-H surfaces to the receivers. Do not use FW-H models in simulations where porous regions lie between FW-H surfaces and the receivers. For simulations with both fluid and porous regions, the FW-H Surface filters allow you to select impermeable and permeable FW-H surfaces from fluid regions, but not from porous regions.

The following sections cover the three main Ffowcs Williams-Hawkings submodels:

- The Ffowcs Williams-Hawkings Steady model, for 3-D rotation cases, using the FW-H Steady Solver and moving reference frames, with impermeable FW-H surfaces.
- The On-The-Fly FW-H model, an unsteady time model for far-field noise prediction for transient cases, both stationary and rotating, using the FW-H Unsteady Solver.
- The Post FW-H model, an unsteady time model for collecting flow field data in `.trn` files, using the Post FW-H Solver, for the following transient cases:
  - Stationary cases with both permeable and impermeable FW-H surfaces
  - Rotating cases with rotating impermeable Post FW-H surfaces, for complete 360-degree domains, and for periodic domains

The following sections also cover the following:

- Working with FW-H Receivers
- Working with FW-H Surfaces
- Visualizing the FW-H Noise Sources
- FW-H Field Functions Reference
- Ffowcs Williams-Hawkings theory

The Steady FW-H Workflow

The Steady FW-H model is designed for rotating cases. Before using it, first solve until you get a convergent steady-state simulation with a rotating reference frame for noise sources in the near-field using the aeroacoustics recommended analysis procedure.

Then use the following steps:

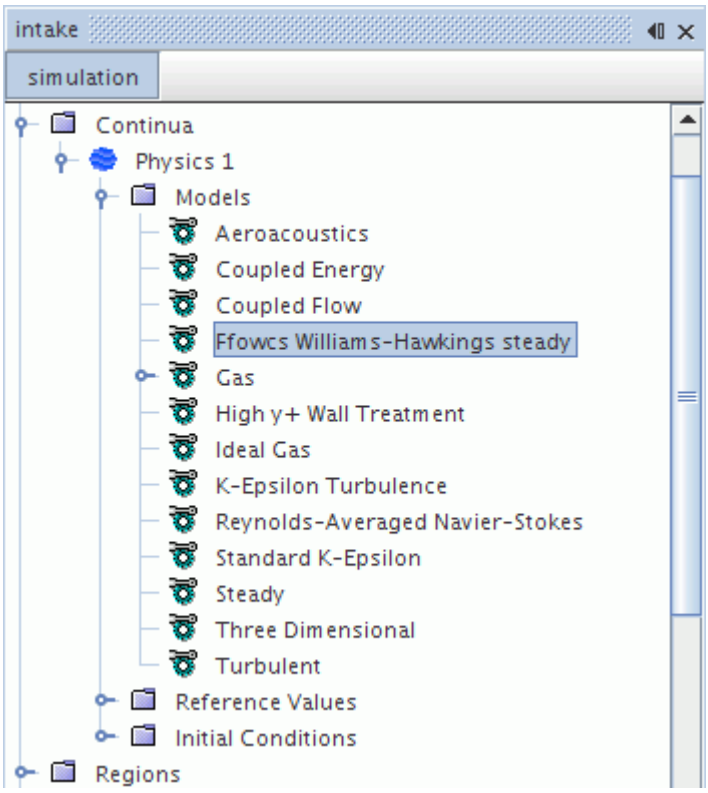
1. While selecting physics models for the fluid continuum, select the Ffowcs Williams-Hawkings Steady model. The FW-H Steady Solver is added automatically.
2. Create FW-H impermeable surfaces. (See Creating Surfaces.)
3. Create FW-H receivers and associate each with an FW-H surface. (See Creating Receivers.)
4. Select the **Solvers** > **FW-H Solver** node and set the solver properties, especially *Number of Time Steps per Revolution* and *Number of Revolutions*, as described in Using the FW-H Steady Solver.
5. Right-click the **FW-H Steady Solver** node and select **Execute Solver**.

6. When the FW-H Steady Solver has finished, export the receiver noise data, then re-import it as tables to visualize and post-process the data.

The Ffowcs Williams-Hawkings Steady Aeroacoustics Model

The FW-H Steady model is based on the convergent steady-state RANS approach for aerodynamic computation of propeller blade loads for a single or multiple rotating reference frames with the FW-H Acoustic Analogy Model. It is for three-dimensional simulations with impermeable FW-H surfaces. It cannot be used for non-rotating cases. (To see applications of this model refer to [59], [66], [70], [71], and [86].)

The **Ffowcs Williams-Hawkings steady** node represents the model.



This method deals with the case where an unsteady motion problem is converted to a steady-state RANS problem by imposing a rotating frame of reference on a static mesh. The **FW-H Steady** acoustics model (for convergent steady-state RANS simulations with MRF) uses a single rotating reference frame when all parts selected for an FW-H impermeable surface have an identical rotation rate value. In this case, even though all parts execute an identical rotating movement, different FW-H impermeable surfaces can have different rotation rates. When different parts rotate at different rotation rates, the FW-H Steady model implements multiple rotating reference frames.

Thickness surface noise is dependent only on the shape and motion of the blade. Conceptually, the displacement of the air by the rotor blade causes this noise. Loading noise is an aerodynamic adverse effect due to the acceleration of the force distribution on the air around the rotor blade due to the blade passing through it. Brentner and Farassat [59] listed various sound generation mechanisms. However, steady-loading

noise and blade-displacement noise are perceived as the major contributions in the absence of installation effects.

The thickness and loading noise from relations of the FW-H acoustics model are computed from steady-state computation of the pressure field, and the time derivative of the surface pressure is set to zero.

The steady formulation allows prediction of an unsteady pressure time signal from a steady simulation by defining an artificial unsteady time-step. Specify this artificial time-step in the *Number of Time Steps Per Revolution* and the *Number of Revolutions* properties of the FW-H Steady solver.

Note: When the flow direction is not aligned with the axis of rotation, it is not possible to use FWH for steady simulations because the loading on the blade is unsteady. Therefore, perform a transient rigid body motion simulation (with a transient sliding mesh) with the FW-H Unsteady model.

Previous aeroacoustic results for a few propeller cases show good agreement with experimental measurements for high rotational speed (over 2500 rpm) and larger under-prediction of the noise at low rotational speed. Thus it is not possible to predict vortex noise at low tip Mach numbers (low rotational speed). For low tip Mach numbers, it is suggested to choose accurate simulations such as a transient LES simulation with rotating motion, to simulate all transient features and broadband noise.

If the boundary is part of the region where motion is stationary and the reference frame is rotating, then the model uses the rotating reference frame properties to compute the rotating movement of the FW-H Impermeable Surface. The reference frame properties are: origin, direction, and rotation rate. To determine the position of the blade on rotating movement, the face centroid of the blade and the surface area of the blade are evaluated at each artificial unsteady time-step.

Periodic domain is considered in the same way as in the On-The-Fly FW-H model for rotating motion.

Ffowcs Williams-Hawkings Steady Acoustics Model Properties

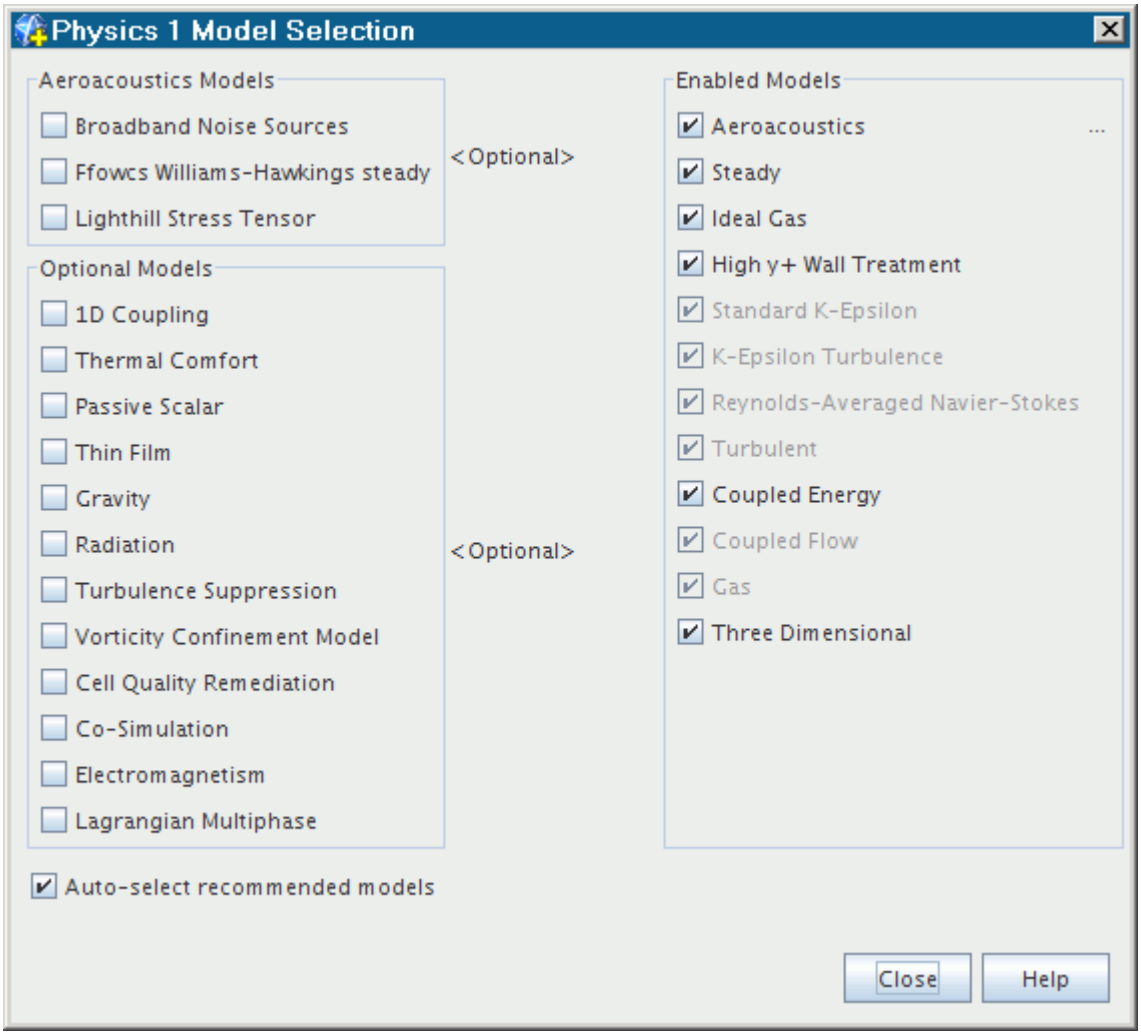
Far-Field Density	The density of the far-field fluid. The default value is 1.225 kg/m^3.
Far-Field Sound Speed	The sound speed in the far-field. See Eqn. (3885) The default value is 340 m/s.

Contents:

Selecting the Ffowcs Williams-Hawkings Steady Model
Using the FW-H Steady Solver

Selecting the Ffowcs Williams-Hawkings Steady Model

The Ffowcs Williams-Hawkings Steady aeroacoustics model becomes available in the *Physics Model Selection* dialog after you select the Steady time model.

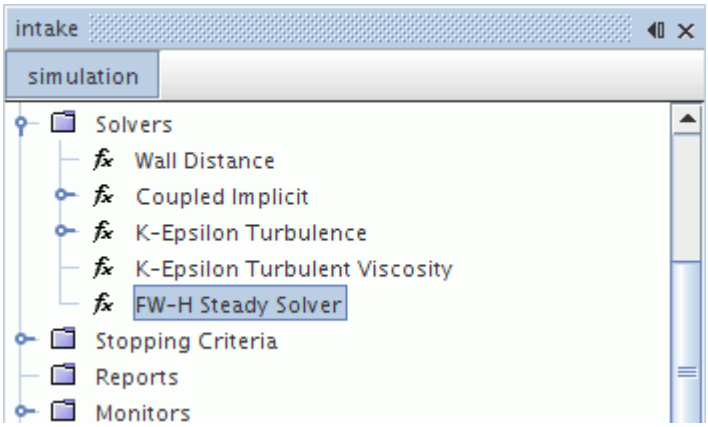


Using the FW-H Steady Solver

The solver provides an approximation of the results that you can expect from a steady state simulation.

On computation of the thickness and loading noise, the term corresponding to the time derivative of the surface pressure is set to zero.

The Ffowcs Williams-Hawkings Steady model activates the FW-H Steady solver. This solver implements the advanced time approach, to account for the time delay between the emission time and reception time. The FW-H Steady model is able to compute noise from propellers in the subsonic as well as transonic and low supersonic domain at low numerical cost. It is suited for propeller optimization because it is inexpensive and reliable, though it is an approximate method. The **FW-H Steady Solver** node represents the solver.



To run the solver:

1. Enter the *Number of Time Steps per Revolution* and *Number of Revolutions*, as described in the following properties table.
2. Right-click the **FW-H Steady Solver** node and select **Execute Solver**.

FW-H Steady Solver Properties

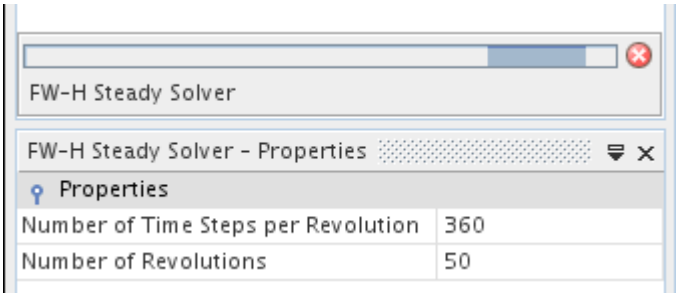
Number of Time Steps per Revolution	Sets the number of equivalent transient time-steps for the rotating region to complete one revolution.
Number of Revolutions	Specifies the number of revolutions that simulated by the FW-H Steady Model.

FW-H Pop-up Menu

Execute Solver	Runs the FW-H Steady Solver.

Stopping the FW-H Steady Solver

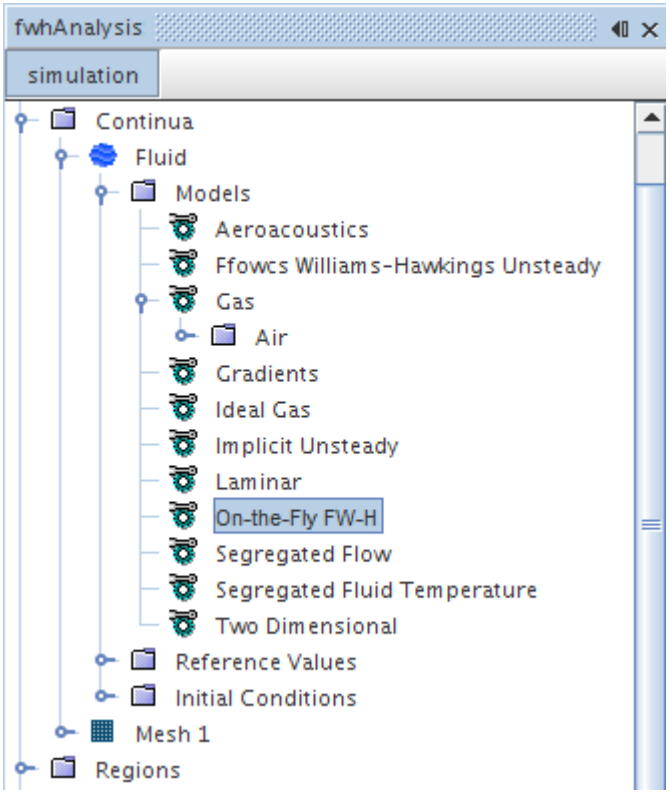
To stop the solver immediately, without completing the set number of time-steps, click on the solver stop button located next to the progress bar. The solver stops and saves the partial solution on the point receivers.



The On-the-Fly FW-H Model

The On-the-Fly FW-H acoustic model provides the aerodynamic noise prediction in parallel with the aerodynamic computation.

The **On-the-Fly FW-H** node becomes available when the Ffowcs Williams-Hawkings Unsteady model is active.



A transient simulation using the On-the-Fly FW-H model does the following:

- It computes the time-accurate data on the emission surfaces (sources of noise), which are the integration surfaces for the FW-H solver. Time-accurate solutions can be obtained from an unsteady turbulent or laminar simulation (DES/LES simulations are suggested).
- Time variation of the sound pressure at receiver locations is computed based on the FW-H solver.
- It can calculate sound pressure from quadrupole noise sources for all regions selected in all point receivers when *Quadrupole Noise* is **On**.

On-the-Fly FW-H Model Expert Properties

The expert properties below are common to both stationary simulations or simulations with rigid body motion:

Quadrupole Noise

Determines if the volumetric quadrupole radiated noise is calculated. When this property is **On**, the model calculates the quadrupole noise. The **Quadrupole Volume Term** field function is activated. An option to

gather quadrupole noise data at each FW-H receiver is activated. When this property is **Off** (which is the default) the quadrupole noise is not calculated.

Quadrupole Noise Calculations

The On-the-Fly model calculates quadrupole noise using the collapsing sphere formulation. This formulation is independent of the formulations used to compute the sound pressure for the selected FW-H surfaces, such as Farassat's Formulation 1A or Dunn-Farassat-Padula Formulation 1A.

The FW-H quadrupole calculations and integration on regions are performed independently from the calculations on surfaces.

Contents:

Using the On-The-Fly FW-H Model with Rigid Body Motion
What Is the On-The-Fly FW-H Workflow?
What Is the FW-H Unsteady Solver?

Using the On-The-Fly FW-H Model with Rigid Body Motion

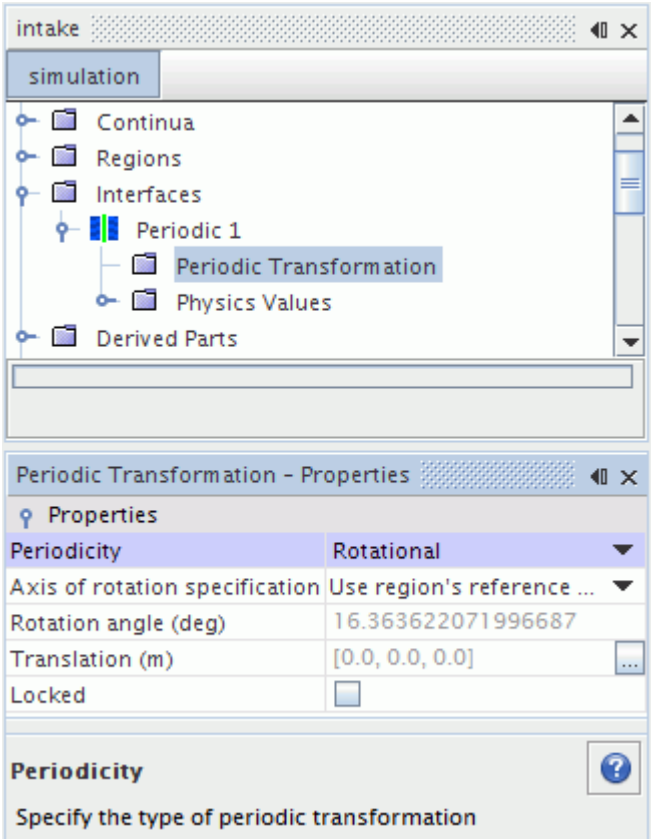
The FW-H Surface can have a rigid body motion, where it can be available for rotating motion as well as rotating and translating motion.

Using the On-The-Fly FW-H model with a rigid body motion (see [57], [59], [66]) involves actual displacement of mesh vertices in real time. The On-The-Fly FW-H model for rotating sliding mesh (a subset of rigid body motion) can be used in the prediction of noise from rotating blades as helicopter rotors and aircraft propellers.

For domains with rotational symmetry, you can model one sector of the domain using a periodic interface. The On-The-Fly FW-H model then reproduces noise sources for all sectors of the full domain. The model determines the number of sectors from the Rotation angle that you specify for the periodic transformation.

The periodic transformation must be purely rotational (that is, either *Periodicity* is set to **Rotational**, or *Periodicity* is set to **Rotational+Translational** with *Translation* equal to `[0, 0, 0]`.)

The *Axis of rotation specification* property can use either the method **Use region's reference axis** or the method **Specify axis.**



The On-The-Fly FW-H model computes the sound pressure signal for all boundary zones of the FW-H surface, where each boundary zone can be part of a region with different number of sectors. You can use the model with periodic domains by selecting *FW-H Impermeable*, but only for three-dimensional simulations.

What Is the On-the-Fly FW-H Workflow?

Before using the On-the-Fly FW-H model, first solve your unsteady simulation for noise sources in the near-field using the aeroacoustics recommended analysis procedure.

Then, do the following:

1. Add the On-the-Fly FW-H model and the FW-H Unsteady Solver to your analysis.

   Note: For quasi-steady rotating machines, use the Ffowcs Williams-Hawkings Steady model and the FW-H Steady Solver.

2. Create FW-H surfaces. See Creating FW-H Surfaces.
3. For each FW-H surface, specify the relevant source boundaries. See Impermeable FW-H Surfaces and Permeable FW-H Surfaces.

   For Permeable FW-H surfaces, make sure that you select as source boundary the side of the interface that has surface normal pointing inwards (that is, opposite the direction of sound propagation). To check the surface normal direction:

   a) In a vector scene, plot the **Area** field function on both boundaries of the internal interface.

   b) If the source boundary that you selected for the FW-H surface does not have the correct surface normal, choose the other side of the interface as source boundary.
4. Create FW-H receivers and associate each with an FW-H surface. Highlight the receivers and double-check the locations.
5. Set the FW-H Unsteady Solver properties, especially the time when receivers start collecting data.
6. If desired, set up quadrupole noise calculation:
   a) Select the On-the-Fly model and activate the *Quadrupole Noise* property.
   b) Select the volume sources of noise: select the *Region* property of each point receiver and select the list of regions generating the volumetric quadrupole noise. By default, no regions are selected.
   c) Export the sound pressure: for each point receiver, right-click and select the **Export Volume Quadruple...** action.

   When the simulation runs, the model calculates the quadrupole noise using the collapsing sphere formulation, based on the selected volume distribution of the noise sources.
7. When the solution has finished, export the receiver noise data and reimport as tables to visualize and post-process the data.

Using Volume Sources for Quadrupole Noise

The On-the-Fly model can compute quadrupole noise contributions for specified volumes. Use this feature when you want to:

- Compute the quadrupole noise contribution from specific region.
- Compare quadrupole noise contributions from different regions.
- Compute quadrupole noise from combinations of regions.
- Visualize the **Quadrupole Volume Term** field function to see the location of the quadrupole sources.

See Visualizing the FW-H Noise Sources.

What is the FW-H Unsteady Solver?

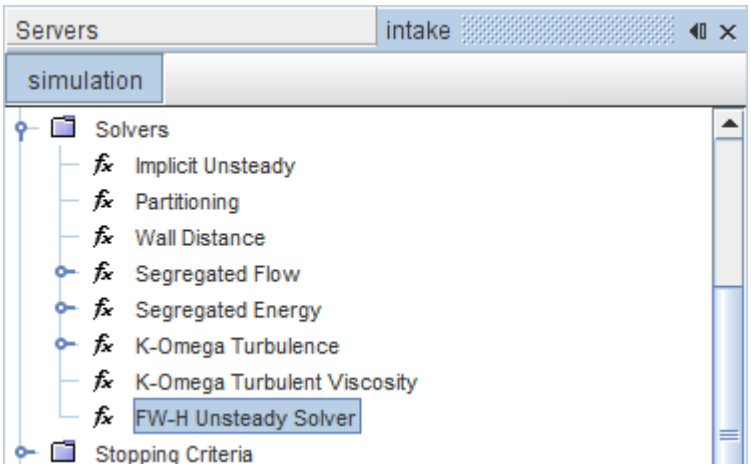
The On-The-Fly FW-H model activates the FW-H Unsteady Solver. This solver implements the advanced time approach for subsonic and transonic flows, to account for the time delay between the emission time and reception time.

The advanced time approach is also known in the literature as a “source-time-dominant” approach. It was developed by Brentner [58], based on Formulation 1A of the FW-H analogy by Farassat [68]. It was extended to permeable (or penetrable) integration surfaces by di Francescantonio [65] and by Brentner and Farassat [59].

This approach looks forward in time to see when the observer receives the currently generated sound waves. In the source-time-dominant algorithm, it fixes the source time and determines when the sound from each point on the FW-H Surface reached the observer. The arrival time of the acoustic signal is different for each point on the FW-H Surface. Then the sound pressure at the receiver is obtained by accumulating the arriving signals in time slots. The overall observer acoustic signal is obtained from the summation of the acoustic signal that is radiated from each source element of the FW-H Surface during the same source time.

Note: It is important to collect a time history of the sound pressure at the receiver location long enough to do spectral analysis using Fourier Transforms.

The **FW-H Unsteady Solver** node represents the model.



FW-H Unsteady Solver Properties

Start Time	The elapsed simulation time before the solver begins calculating.

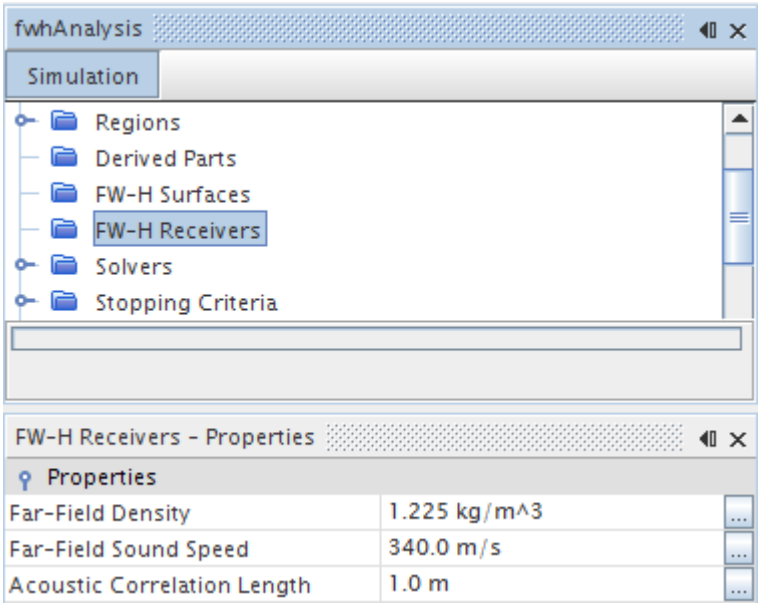
FW-H Unsteady Solver Expert Properties

Solver Frozen	When **On**, prevents the solver from updating the solution while iterating or responding to other solvers. This feature is primarily a “debugging” tool and is not supported for normal simulation work.

Working with FW-H Receivers

The **FW-H Receivers** node is available for the Ffowcs Williams-Hawkings steady model and the On-the-Fly FW-H model.

The **FW-H Receivers** node is the manager node for all point receivers in the STAR-CCM+ simulation tree.



FW-H Receivers Properties

Far-Field Density

The density of the far-field fluid. The default value is 1.225 kg/m^3.

Far-Field Sound Speed

The sound speed in the far-field. See Eqn. (3885)

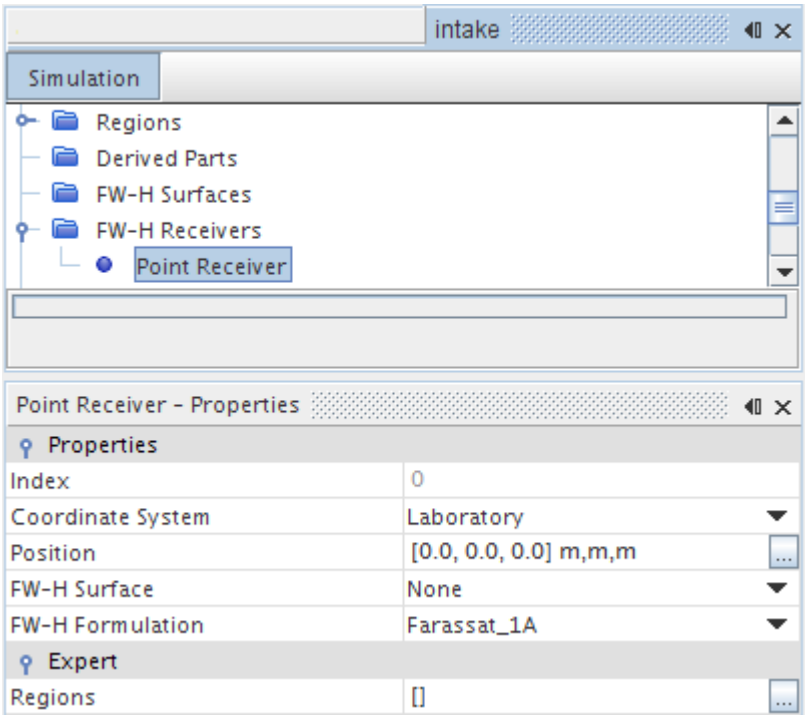
The default value is 340 m/s.

Acoustic Correlation Length

Sets the correlation length to include perfect spanwise correlation of the flow over a finite span, for two-dimensional simulations only.

What is a Point Receiver?

A point receiver is a point inside or outside the domain (located in the mid- to far-field), where the sound pressure is predicted. The total acoustic signal that is measured at a receiver is the sum of the acoustic signals that are radiated from each source element of the FW-H surfaces. A point receiver is represented by a node in the object tree.

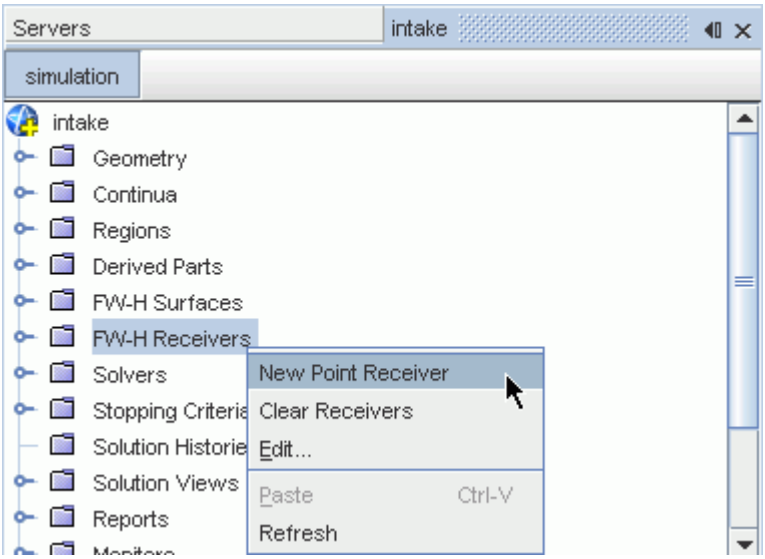


If more than one of these objects are added to the simulation, then the second and subsequent ones have numbers at the end of the labels. You can define many receivers, each with a specific FW-H surface that can be monitored.

Receivers can be renamed, copied, and deleted in the same way as other simulation objects.

Creating Point Receivers

To add a point receiver to the simulation, right-click the **FW-H Receivers** manager node and select **New Point Receiver**. Set the properties.



To display the point receiver, open the associated scene, then open the **FW-H Receivers** node and select the node for that point receiver.

Pop-Up Menu of the FW-H Receivers Manager

New Point Receiver

Adds a point receiver to the object tree.

Clear Receivers

Clears the acoustic recorded data of all the point receivers.

Point Receiver Properties

Index

Index reference for the point receiver (read-only).

Coordinate System

Combined with the setting for the *Position* property, your selection of a local coordinate system permits definition of the point location of the receiver.

Position

The position of the receiver.

FW-H Surface

Selects which FW-H surface to watch from a drop-down list of surfaces in the simulation. Only one of these surfaces (impermeable or permeable) can be selected at a time for each point receiver, though each surface can contain multiple boundaries.

FW-H Formulation

Selects the formulation to use for the surface terms.

Farassat_1A

Uses Farassat's Formulation 1A, Eqn. (3967) and Eqn. (3968). This is the default formulation.

Dunn_Farassat_Padula_1A

Uses the Dunn-Farassat-Padula Formulation 1A, Eqn. (3969) and Eqn. (3970).

Regions

Defines the regions or user-defined volumes to contribute to the computation of quadrupole noise. By default, no regions are selected.

Point Receiver Pop-Up Menu

Export Surface Total...

Exports the summation of the overall surface acoustic signal, $\mathrm{p}'_S(\vec{x}, t)$, that is radiated from each element of the FW-H Surface during the source time. See Eqn. (3966) for formulation.

Export Surface Thickness...

Exports the summation of the acoustic thickness signal, $p'_T(\vec{x},t)$, that is radiated from each element of the FW-H Surface during the source time. See Eqn. (3967) for formulation.

Export Surface Loading...

Exports the summation of the acoustic loading signal, $p'_L(\vec{x},t)$, that is radiated from each element of the FW-H Surface during the source time. See Eqn. (3968) for formulation.

Export Volume Quadrupole...

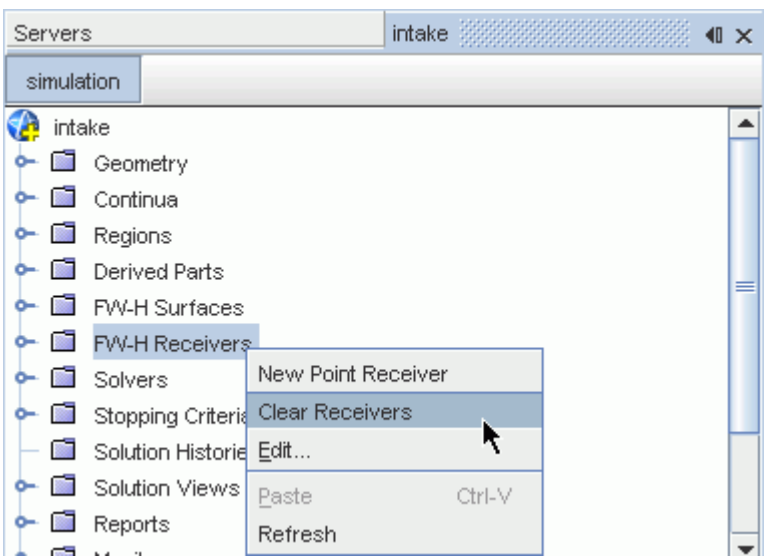
Exports the summation of the acoustic quadrupole signal, $p'_Q(\vec{x},t)$, that is radiated from each cell in the regions defined in the point receiver property during the source time. This action is independent of the **Export Surface...** actions. See Eqn. (3973) for formulation.

Clear Receiver

Clears the acoustic recorded data of the point receiver.

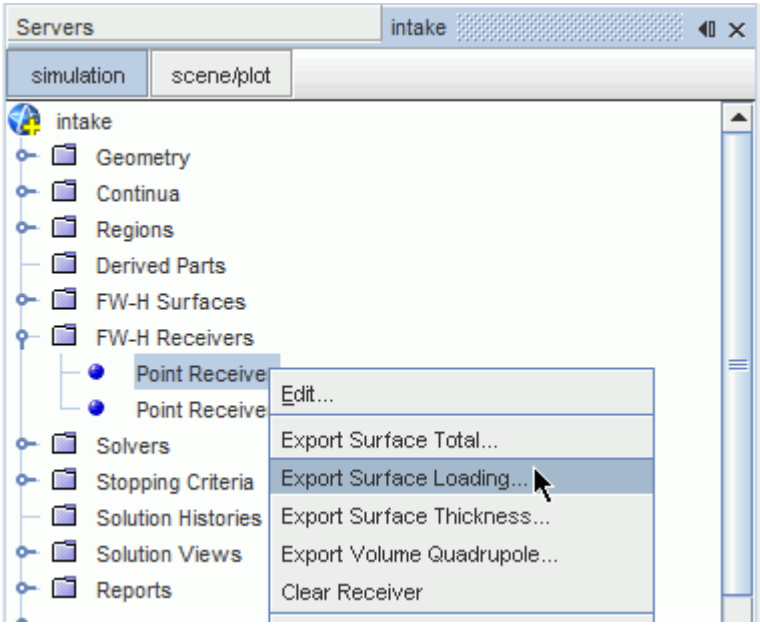
Clearing Point Receiver Data

To clear the solution data of the point receivers, right-click the **FW-H Receivers** manager node and select **Clear Receivers**.



Exporting Point Receiver Data

To export the solution data of a point receiver, right-click the node of that receiver and select one of the export options.



This action opens a standard dialog for you to save the data of the receiver as a `.csv` file in a location of your choice.

For each type of export, the sound pressure data is pruned before export, which effectively clips the receiver data at the beginning and end of the collection time to ensure that all faces on all the source surfaces were contributing to the received data. The pruning accounts for the fact that signals from source surface faces closest to the receiver arrive before signals from faces furthest from the receiver. To account for this pruning of exported data, run the case for a longer period.

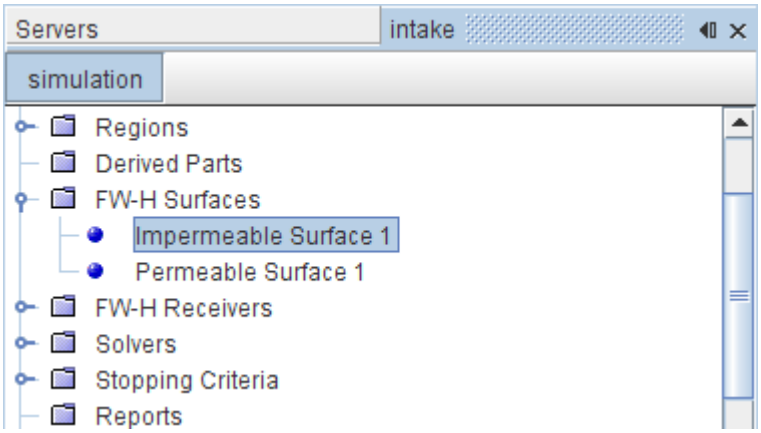
Working with FW-H Surfaces

FW-H Surfaces define the source locations of the noise.

FW-H surfaces can be thought of as being similar to radio speakers—they are the source of noise. Greschner and others [72] suggest that there are two types of FW-H Surfaces: impermeable and permeable. Surfaces can be renamed, copied, and deleted in the same way as other simulation objects.

Impermeable FW-H Surfaces

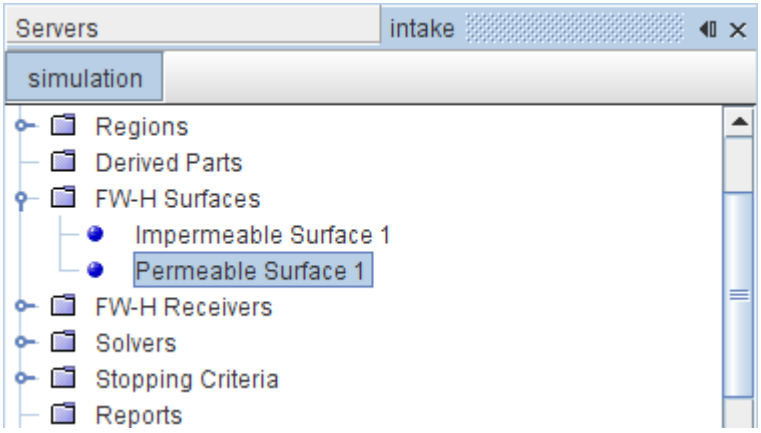
An impermeable FW-H surface acts as a filter of wall boundary conditions from which noise originates. Locate impermeable surfaces at one or more wall boundaries.



Note: For the Ffowcs Williams-Hawkings steady model, use only impermeable surfaces.

Permeable FW-H Surfaces

A permeable FW-H surface acts as a filter of internal interface boundaries. Locate permeable surfaces on a fine region of the mesh that surrounds all sources of noise: monopoles, dipoles, and quadrupoles.



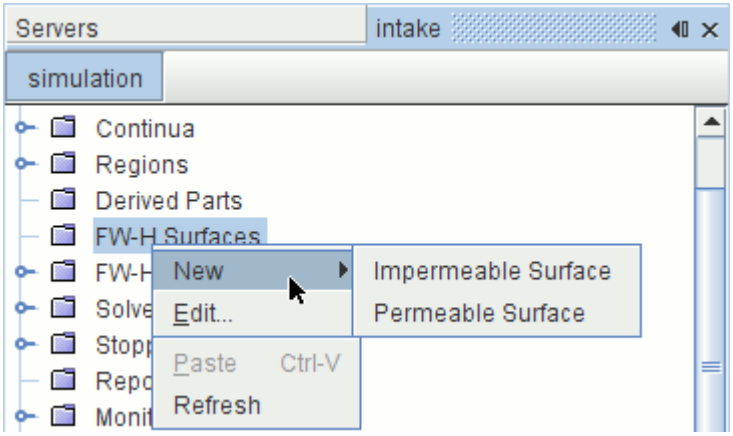
When you select a Permeable FW-H surface, the On-the-Fly and Post FWH models calculate the sound pressure at receiver locations using Farassat's Formulation 1A or Dunn-Farassat-Padula Formulation 1A.

The Permeable FW-H formulations include all sources of noise, distributed on the surface as monopole and dipole terms, and also the contribution from the volumetric sources inside by the integration surface. The formulations assume that the contribution to the total noise of volumetric sources outside the integration surface is negligible.

Since the evaluation of the FW-H surface sources is independent of FW-H volume sources, you do not need to select regions for each point receiver.

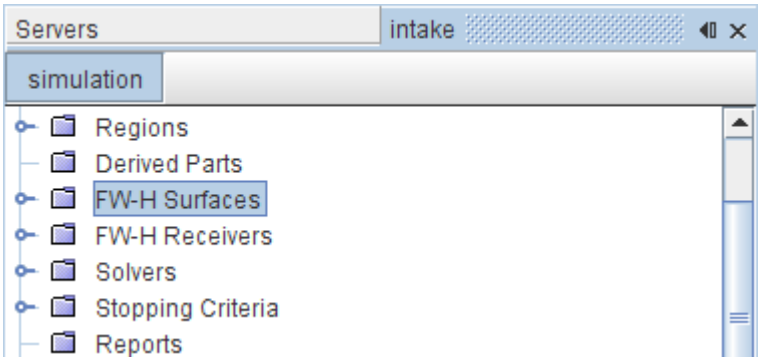
Creating Surfaces

To add a surface to the simulation, right-click the **FW-H Surfaces** manager node and select **Impermeable Surface** or **Permeable Surface** from the **New** submenu.



FW-H Surfaces Manger

The **FW-H Surfaces** manager node is available for the Ffowcs Williams-Hawkings steady model and the On-the-Fly FW-H model. The **FW-H Surfaces** node is the manager node for all FW-H surfaces (impermeable and permeable) in the STAR-CCM+ simulation tree.



This manager can hold any number of surfaces up to the memory limits of the computer.

Surface Properties

Boundaries	List of boundaries used for FW-H surfaces. Edit this property to add or remove boundaries with the in-place object selector.

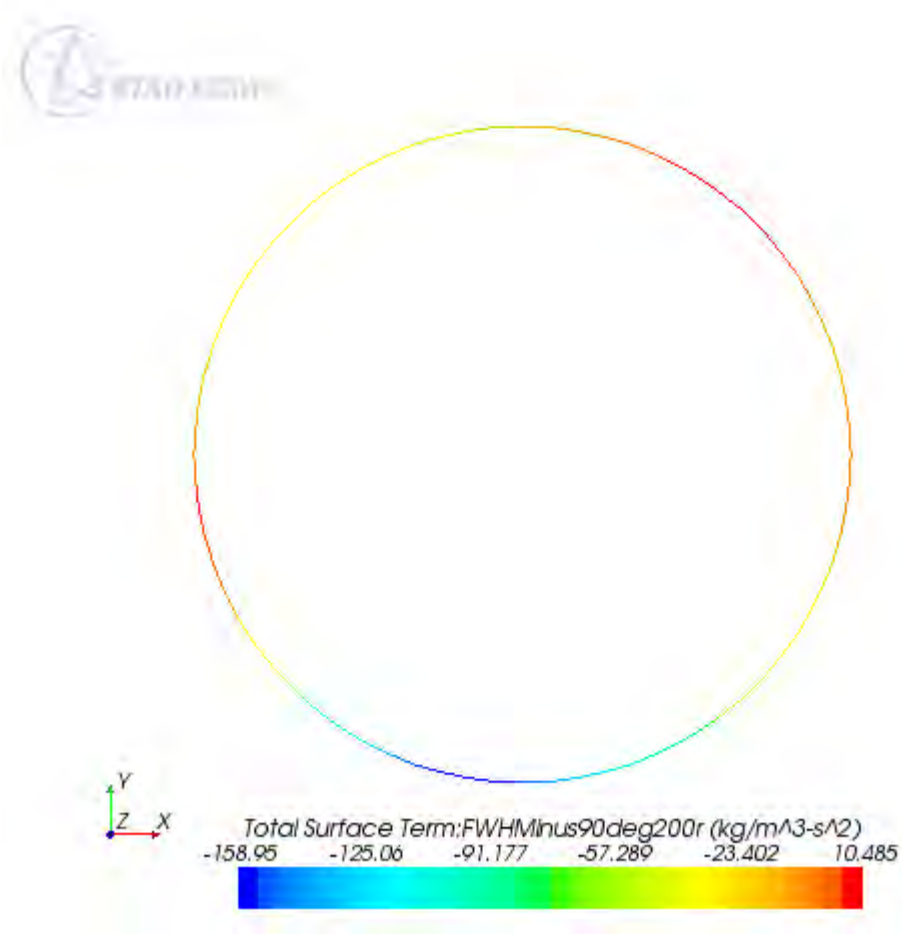
Pop-Up Menu of the FW-H Surfaces Manager

New	Adds a surface (impermeable or permeable) to the object tree, depending on your choice from the submenu.

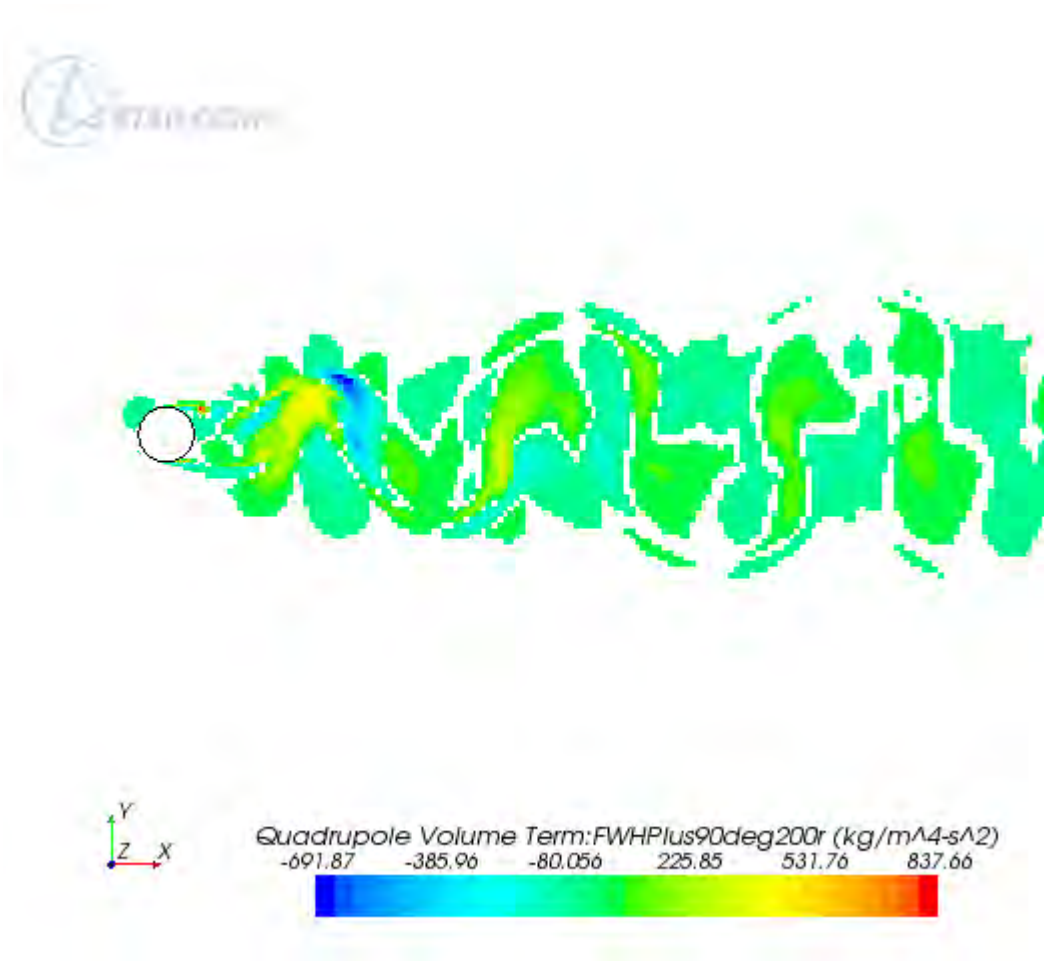
Visualizing the FW-H Noise Sources

To view the Loading and Thickness surface noise components, or the Quadrupole volume noise, select the scalar function relating to the correct part.

For a surface term, display the scalar function on the same parts you used to define the associated **FW-H Surface**. The image below shows the Total Surface Noise for a cylinder in a crossflow:



For a volume term, display the scalar function on the region you specified in the Point Receiver properties. The image below shows the quadrupole noise for the same cylinder as shown above:



Note: The FW-H Noise Sources field functions can be visualized after the FW-H solver begins calculating, provided that the physical time is larger than the start time of the FW-H solver. (The start time of the FW-H solver is set on the **Solvers** > **FW-H Solver** node.)

FW-H Field Functions Reference

The following field functions are made available when either the Ffowcs Williams-Hawkings steady model or the On-The-Fly FW-H model is used.

Total Surface Term [Receiver Name]	$p'_S(\vec{x},t)$ in Eqn. (3966). The summation of the *Loading Surface Term* and the *Thickness Surface Term* for a particular source surface generates the *Total Surface Term*.
Thickness Surface Term [Receiver Name]	$p'_T(\vec{x},t)$ in Eqn. (3967) for general flows and in Eqn. (3969) for flows with rigid body motion or moving reference frames. In each equation, the field function is normalized with face-area. The displacement of fluid as the body passes generates

	the Eqn. (3958) noise. For stationary bodies (such as side view mirrors and circular cylinders) the thickness surface term is zero.
Loading Surface Term [Receiver Name]	$p'_L(\vec{x},t)$ in Eqn. (3968) for general flows and in Eqn. (3970) for flows with rigid body motion or moving reference frames. In each equation, the field function is normalized with face-area. The unsteady motion of the force distribution on the body surface generates the Eqn. (3959) noise.
Quadrupole Volume Term [Receiver Name]	$p'_Q(\vec{x},t)$ in Eqn. (3973), normalized with cell volume. Non-linearities in the flow generate quadrupole noise. The field function is only enabled when you activate the *Quadrupole Noise* expert property.

The Post FW-H Model

The Post FW-H model collects data in `.trn` files for post-processing. The Post FW-H model calculates sound pressure at far-field receivers based on input data stored in `.trn` files, but does not use data from other sources such as monitors, field functions, or flow field data from the original simulation.

This model can be used in the following cases:

- Stationary cases with multiple permeable and impermeable Post FW-H surfaces.
- Transient rotating cases with multiple impermeable Post FW-H surfaces, for whole 360-degree domains and periodic domains.

Rotating cases use periodic domains when the coupled or segregated solver computes flow in one sector of a rotating body. (In such cases, the periodic domain models one sector of a symmetrical rotating body, such as one blade of a fan.)

The Post FW-H model exports pressure, density, and velocity flow field data from the selected Post FW-H surfaces in `.trn` file format as described in Working with Post FW-H Surfaces. For rotating cases, with periodic domains, the model also exports the rotation angle (in degrees) required to compute the number of sectors.

The Post FW-H model obtains the rotation angle by searching for a Periodic Interface configuration and exports the value of the angle from the **Periodic Transformation** properties. The rotation angle is available when the *Periodicity* property is set to **Rotational** or to **Rotational+Translational** with *Translation* set to zero, and the *Axis of rotation specification* property uses the method **Use region's reference axis** or the method **Specify axis**. See Using the On-the-Fly FW-H Model with Rigid Body Motion.

This model uses the Post FW-H Solver. This is an unsteady model.

This model uses the advanced time approach, as described for the FW-H Unsteady Solver. In the advanced time approach, the surface integrals are computed at source time, but signals are accumulated in discrete bins, at receiver time.

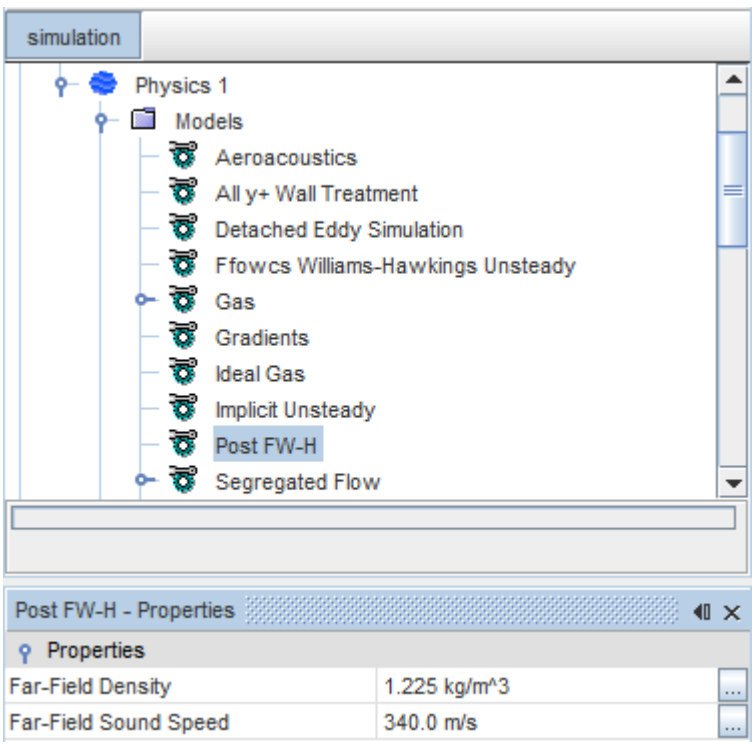
The model calculates the far-field noise contribution at each point receiver on request, when you use the **Execute Solver** action in the Post FW-H Solver. For each selected imported surface and each receiver, at each collected flow field time-step, the model:

- Computes the thickness and loading terms. It computes all surface integrals that are based on reading pressure surface data at the required time levels for accurate time derivatives.
- Accumulates the contribution of each individual integral in discrete “time bins,” each bin correspond with an actual reception time.

The noise level at each individual receiver, for the time bin of that receiver, is a superposition of the signals from sound sources on the selected acoustic surfaces, lagged in time.

The model can include convective effects. The simulation takes account of the moving medium by including the effective speed of sound in the surface terms. It makes the appropriate corrections for the receiver time by using the different propagation speed of the pressure waves. See the *Convective Acoustics Effects* property under Post Point Receiver Properties.

If the receiver is not visualized in the post-processing scene, use Steps 2 and 3 under Working with Post FW-H Point Receivers.



Contents:

What Is the Post FW-H Workflow?
What Is the Post FW-H Solver?
Working with Post FW-H Surfaces
Working with Post FW-H Point Receivers

What Is the Post FW-H Workflow?

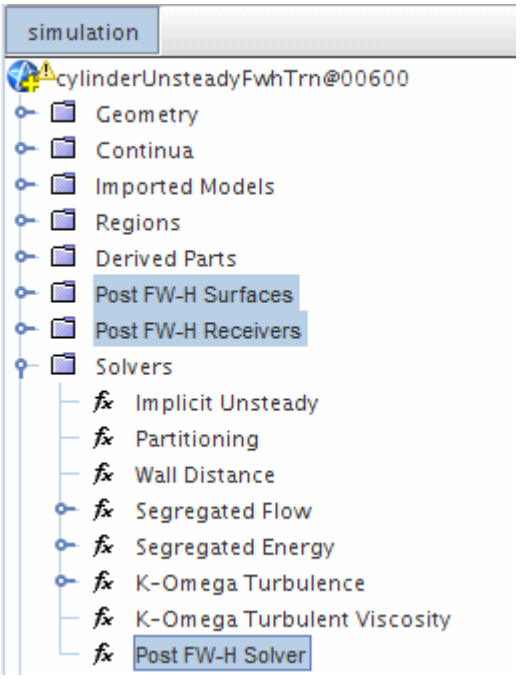
Before using the Post FW-H model, solve the unsteady simulation for noise sources in the near-field using the recommended aeroacoustics analysis procedure:

1. Add the Post FW-H model to your analysis.
2. Create all required Post FW-H Surfaces. Creating Surfaces.
3. For each FW-H surface, specify the relevant source boundaries. See Impermeable FW-H Surfaces and Permeable FW-H Surfaces.

   For Permeable FW-H surfaces, make sure that you select as source boundary the side of the interface that has surface normal pointing inwards (that is, opposite the direction of sound propagation). To check the surface normal direction:

   a) In a vector scene, plot the **Area** field function on both boundaries of the internal interface.

   b) If the source boundary that you selected for the FW-H surface does not have the correct surface normal, choose the other side of the interface as source boundary.
4. Set the Post FW-H Solver properties, especially the *Start Time Step* and *Collection Frequency*.
5. Run the simulation. The Post FW-H Solver collects the source data in `.trn` file format.
6. Import the `.trn` file format and create Post FW-H receivers, then set the properties *Imported FW-H Surfaces* and *Point Part*. Highlight the receivers. The point receiver can be on any imported surface. Use units of meters; see Importing a CAE Model.
7. Right-click on the **Post FW-H Solver** node and select **Execute Solver**.
8. When the Post FW-H Solver has finished, export the receiver noise data and reimport it as tables to visualize and post-process the data.



9. If desired, go back to step 5, alter the receivers for different output data (for instance, set the receivers at a different distance), and execute the solver again.

What Is the Post FW-H Solver?

The Post FW-H model activates the Post FW-H Solver.

The solver has two actions:

1. Define the method for collecting source data in `.trn` files, before the simulation starts.

   The solver collects source data for permeable and impermeable Post FW-H surfaces, in `.trn` file format, on a schedule based on the properties *Start Time Step* and *Collection Frequency*. The solver can export multiple surfaces at the same time.
2. Compute the FW-H sound pressure signals after the simulation is finished.

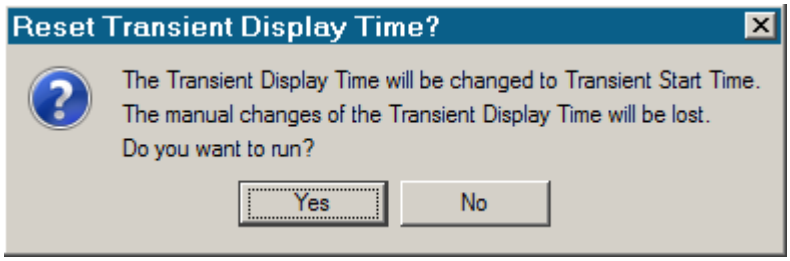
   The solver computes the sound pressure at the locations of the selected point receivers, for the selected imported surfaces.

The solver computes the sound pressure based on a user-chosen selection of imported FW-H surfaces, to capture all noise sources. In each run, the solver can compute values at multiple selected point receivers and imported FW-H surfaces.

Running the Post FW-H Solver

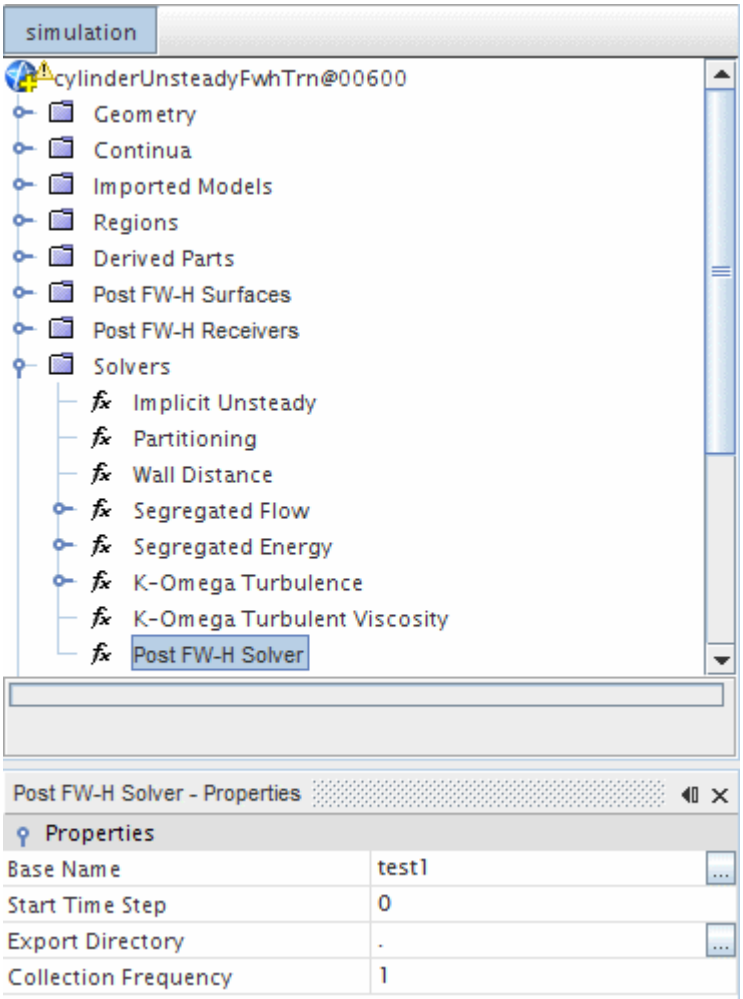
To run the solver:

1. Right-click on the **Post FW-H Solver** node.
2. Select **Execute Solver**.
3. Running the solver entails resetting the transient display time. STAR-CCM+ displays a dialog asking if you want to do this. Click **Yes**.



The sound pressure data is saved on each point receiver.

The export of noise source data in `.trn` file format is completely separate from the computation of the sound pressure signal at different receivers, at different noise sources.

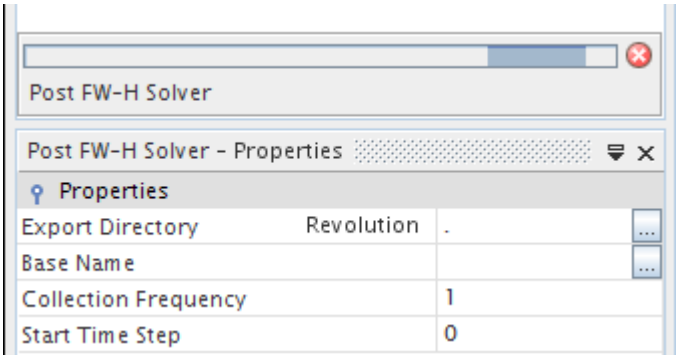


Post FW-H Solver Properties

Start Time Step	The timestep at which the *Collection Frequency* comes into effect. The value can range from 0 up. The default is 0.
Collection Frequency	The number of timesteps between updates. The value can range from 1 up. The default is 1.
Base Name	The base `.trn` file name. The default is the name of the `.sim` file.
Export Directory	The directory for exported `.trn` files. The default is the directory of the `.sim` file.

Stopping the Post FW-H Solver

To stop the solver immediately, without completing far-field computation for all time-steps in the `.trn` file, click the solver stop button located next to the progress bar. The solver stops and saves the partial solution on the post point receivers.



Working with Post FW-H Surfaces

Post FW-H surfaces define the boundaries from which field data is exported to the .trn file, so that it becomes available for a subsequent analysis using the Post FW-H solver. The surfaces define the source locations of noise for far-field noise prediction using the Post FW-H solver.

You can create permeable and impermeable forms of Post FW-H surfaces. For permeable surfaces, the Post FW-H model exports pressure, density, and velocity data. For impermeable surfaces:

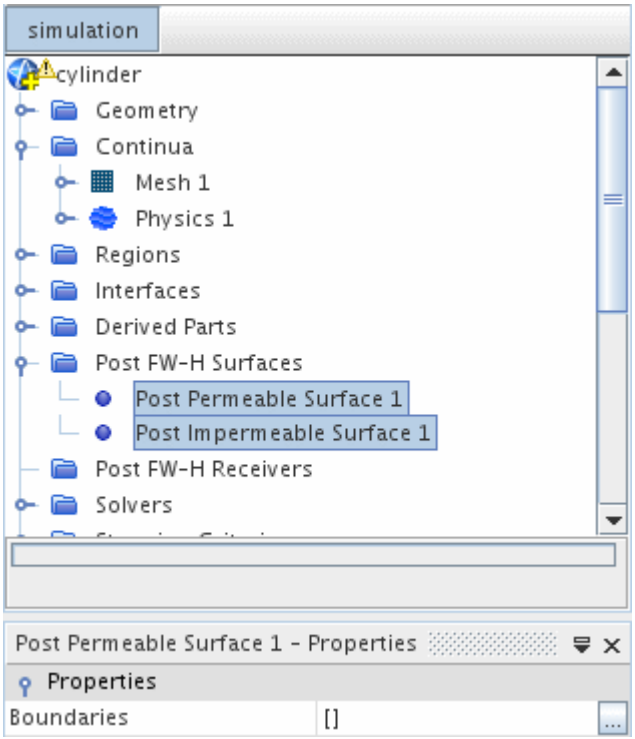
- The model exports pressure data from stationary regions.
- The model exports pressure, density, and velocity data from rotating regions (rigid body motion with zero translation).

See FW-H_Surface_Terms.

To create a post FW-H surface:

1. Right-click on the **Post FW-H Surfaces** node.
2. Select **New** > **Post Permeable Surface** or **New** > **Post Impermeable Surface**.

Post FW-H surfaces can be renamed, copied, and deleted in the same way as other simulation objects.



Post FW-H Surface Properties

Boundaries	Sets the boundaries defined as permeable or impermeable.

Working with Post FW-H Point Receivers

The post FW-H point receivers collect acoustic data. Use them to compute the far-field noise prediction at each receiver location.

You must select a point-derived part for each receiver and a list of imported FW-H surfaces from a single `.trn` file.

To set up a point receiver:

1. Import a `.trn` file containing all transient surface data for the FW-H surfaces generated during a preliminary transient run:
   a. Import the `.trn` file: select **File** > **Import** > **Import CAE Model...**
   b. Select the `.trn` file and click **Open**.
2. Create a point probe under **Derived Parts** . Locate the point at the far-field position where you want to receive acoustic data.
3. For the *Input Parts* property of the probe, select all the FW-H surfaces coming from the `.trn` file, located under **Imported Models** > **Imported Surfaces**. Use the Tree View when selecting surfaces.
4. To create the new post point receiver, right-click the **Post FW-H Receivers** node and select **New...**
5. For the new point receiver, set the *Point Part* property to the name of the point probe.

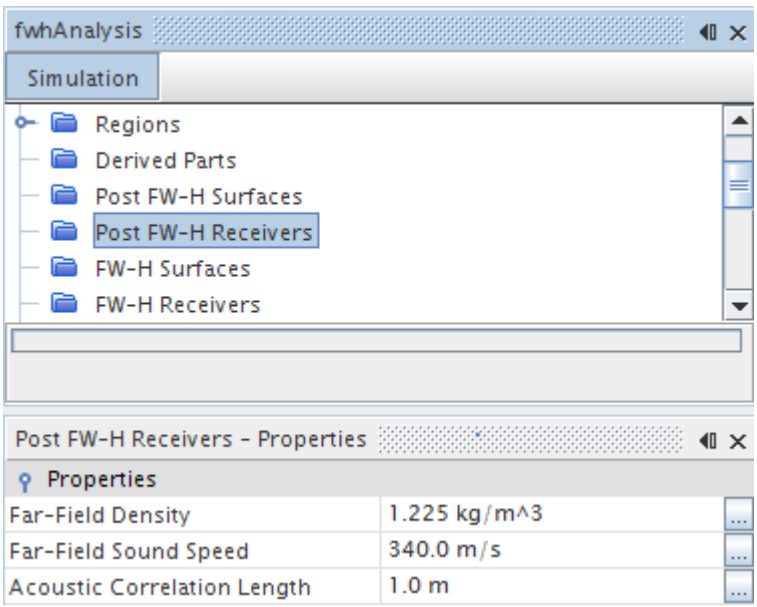
6. If you want to use the Post FW-H model at high flow speeds with compressible flow and account for convective acoustic effects, activate *Convective Acoustic Effects*. Select the **Convective Velocity** node and specify the convective velocity.
7. Set the *Imported FW-H Surfaces* property to the FW-H surface from which you want to capture sound for this receiver.

After you run the Post FW-H Solver, the data from the post-processing point receiver is available for export.

FW-H post-processing point receivers can be renamed, copied, and deleted in the same way as other simulation objects.

To display the point receiver, open the associated scene, then open the **Post FW-H Receivers** node and select the node for that Post FW-H point receiver.

Post FW-H Receivers Node Properties



Far-Field Density

The density of the far-field fluid. The default value is 1.225 kg/m^3.

Far-Field Sound Speed

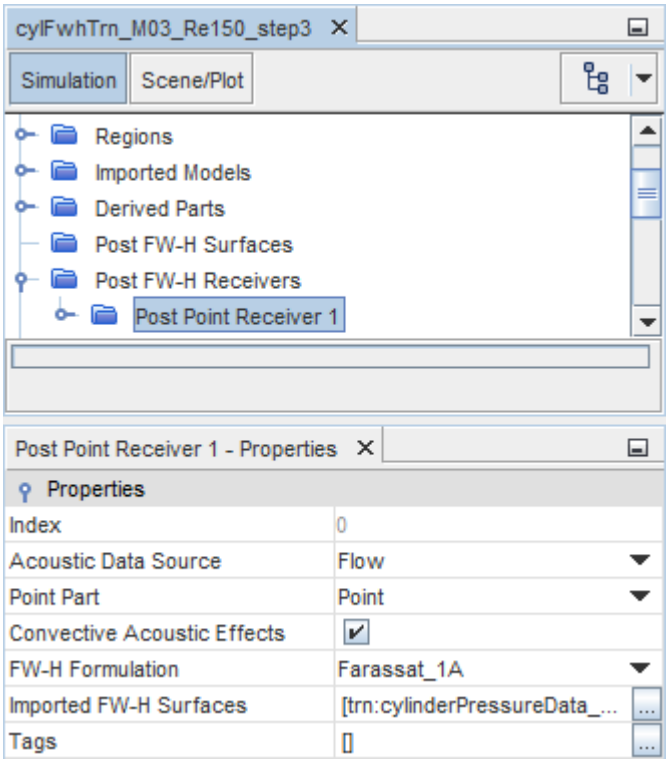
The sound speed in the far-field. See Eqn. (3885)

The default value is 340 m/s.

Acoustic Correlation Length

Sets the correlation length to include perfect spanwise correlation of the flow over a finite span, for two-dimensional simulations only.

Post Point Receiver Properties



Imported FW-H Surfaces

Selects the FW-H surfaces used as inputs.

Point Part

Selects the point probe used as receiver, representing a single microphone.

Convective Acoustic Effects

Reveals the **Convective Velocity** node. See Convective Velocity Node Properties.

Activating Convective Acoustic Effects allows you to use the Post FW-H model at high flow speeds, with compressible flow. The simulation takes moving medium propagation effects into account by including the effective speed of sound in all the FW-H surface terms. It makes the appropriate corrections for the receiver time by accounting for the different propagation speed of the pressure waves.

FW-H Formulation

Selects the formulation to use for the surface terms.

Farassat_1A

Uses Farassat's Formulation 1A, Eqn. (3967) and Eqn. (3968). This is the default formulation.

Dunn_Farassat_Padula_1A

Uses the Dunn-Farassat-Padula Formulation 1A, Eqn. (3969) and Eqn. (3970).

Convective Velocity Node Properties

The convective velocity is the component of velocity of the medium caused by convection. It applies to any Post Point receiver and is defined with respect to the imported FW-H surface.

Method

Selects methods for specifying the profile for the convective velocity from among **Constant**, **Composite**, **Field Function**, **Table**, and **User Code** methods.

Dimensions

Dimensions of the convective velocity. Read-only.

Coordinate System

Specifies the coordinate system for defining the convective velocity. The drop-down list of this property includes any local coordinate systems that exist in the simulation. The default is **Laboratory**.

Post Point Receiver Pop-Up Menu

Export...

Exports the sum of surface acoustic signals radiated from each element of the imported FW-H surface during the collected time in the `.trn` file. See Eqn. (3966) for the formulation.

Guidelines for Ffowcs Williams-Hawkings Modeling

Use the following guidelines with the Ffowcs Williams-Hawkings models.

- Use a mesh fine enough to resolve the noise sources and the acoustic waves. 20 cells per wavelength is best. See Automobile Aerodynamics.
- In cases of quasi-periodic flow, such as a fan blade in a wind tunnel, use spanwise periodic boundary conditions and set a span dimension for them. No value gives a perfect result because the true situation is not 2D or periodic. To get reasonably satisfactory results, make the depth of the computational domain large enough. In the cases of unsteady flow simulations (RANS, DES, or LES) and of noise prediction using the FW-H models, make the depth of the domain greater than or equal to the acoustic correlation length. This depth is not necessarily the same as the depth required for LES and DES simulation. See The Ffowcs Williams-Hawkings Models.
- Use compressible ideal gas LES or DES modeling, coupled with an FW-H model. The compressible ideal gas model correctly accounts for compressible gas effects and sound wave speed in Aeroacoustic simulations.
- For the far-field broadband component, use LES and DES modeling to resolve accurately all turbulent scales that are needed for sound generation.
- For noise sources and propagation in near- to far-field, use high-order discretization schemes, such as Bounded Central-Differencing, Hybrid 3rd Order MUSCL/CD, or other low-dissipation convection schemes.
- Use the **Unsteady Flux Dissipation Corrections** property for the Segregated Flow model, especially if there are abrupt changes in mesh size such as adapted trimmed mesh.

Note: This property adds numerical dissipation to prevent spurious oscillations at mesh interfaces. Do not apply corrections inside the region, where noise sources are present and where noise is transmitted to the permeable FW-H surfaces. You can specify the **Limiting Acoustic-CFL** property as a constant, field function, table, or user code. If you use non-constant values for **Limiting Acoustic-CFL**, make sure that these values vary smoothly in space.

- Use the Min-Mod limiter (Eqn. (1621)) as a less dissipative gradient limiting scheme. For slightly imperfect meshes, use the Modified Venkatakrishnan limiter (Eqn. (1622)).
- Use a time-step small enough to resolve the frequencies being used. Make sure that the Courant number is less than 1 throughout all, or almost all, of the domain.
- Use these guidelines to minimize spurious acoustic reflections when specifying far-field boundary conditions:
  - Set non-reflecting boundary conditions on the outer boundaries of the domain, to allow outgoing disturbances to exit the domain without spurious reflections.
  - Use the Acoustic Suppression Zone model to improve the effect of the freestream boundary conditions. Such boundary conditions are normally used for open domains.
  - To dissipate the vortical structures, stretch the mesh downstream with an acoustic suppression zone.
  - To avoid spurious acoustic reflections, use the correct free-stream boundary together with the Acoustic Suppression Zone model in addition to the mesh stretching.
- To tell if any visible pressure fluctuations are originating at mesh interfaces or numeric boundaries:
  1. Allow sufficient simulation time to compute a mean pressure field p_{mean}.
  2. Subtract the mean pressure field from the instantaneous pressure field $p(t)$, to get the pressure fluctuations $p' = p(t) - p_{mean}$.
  3. Perform surface spectral analysis on the pressure field (**Fft:Pressure**), based on the stored pressure field $p(t)$ on derived parts or sections through the computational domain (saved in `.trn` file format). Use a few fixed frequencies (such as 2000 Hz, 3000 Hz, and 5000 Hz) and a sufficiently long physical time (usually more than 0.1 s).
- To reach a statistically steady transient simulation, use long simulation times (3 to 4 times the flow-through time), so that first the initial transients can be eliminated.
- Activate the unsteady FW-H models in the unsteady simulation.
- Run the On-The-Fly or Post FW-H simulation to have sufficient far-field pressure data for FFT and correct frequency resolution.
- Set the permeable FW-H surface to be open downstream, to avoid the crossing of the wake through the integration surface and to avoid the generation of spurious waves. When you select the internal interface boundary for the permeable FW-H surface, make it surround all the turbulent noise sources, wrapping around both the body and the most energetic vortical structures of turbulence.

Using the Lighthill Stress Tensor

This section explains the following:

- What Is the Lighthill Stress Tensor?
- Selecting the Lighthill Stress Tensor

Also included is a reference section on related field functions.

See also Lighthill Stress Tensor theory.

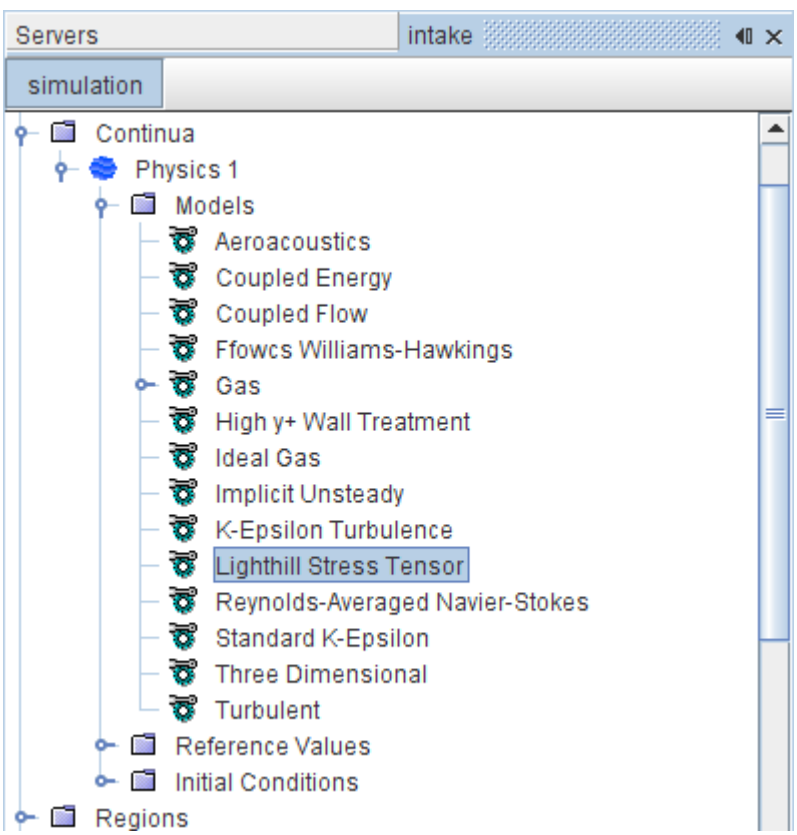
Contents:

What Is the Lighthill Stress Tensor?
Selecting the Lighthill Stress Tensor
Lighthill Stress Tensor Field Functions Reference

What Is the Lighthill Stress Tensor?

In STAR-CCM+ the Lighthill Stress Tensor is purely for post-processing.

Although it is represented by the **Lighthill Stress Tensor** node as a continuum model, it does not actually model acoustics in its own right.



This optional feature interrogates the flow field and compiles velocity derivatives which make up the Lighthill Stress Tensor. It is a scalar with vector components. By observing Lighthill results, it is possible to identify the location of flow-generated aeroacoustics sources.

Contents:

Lighthill Stress Tensor Model Properties

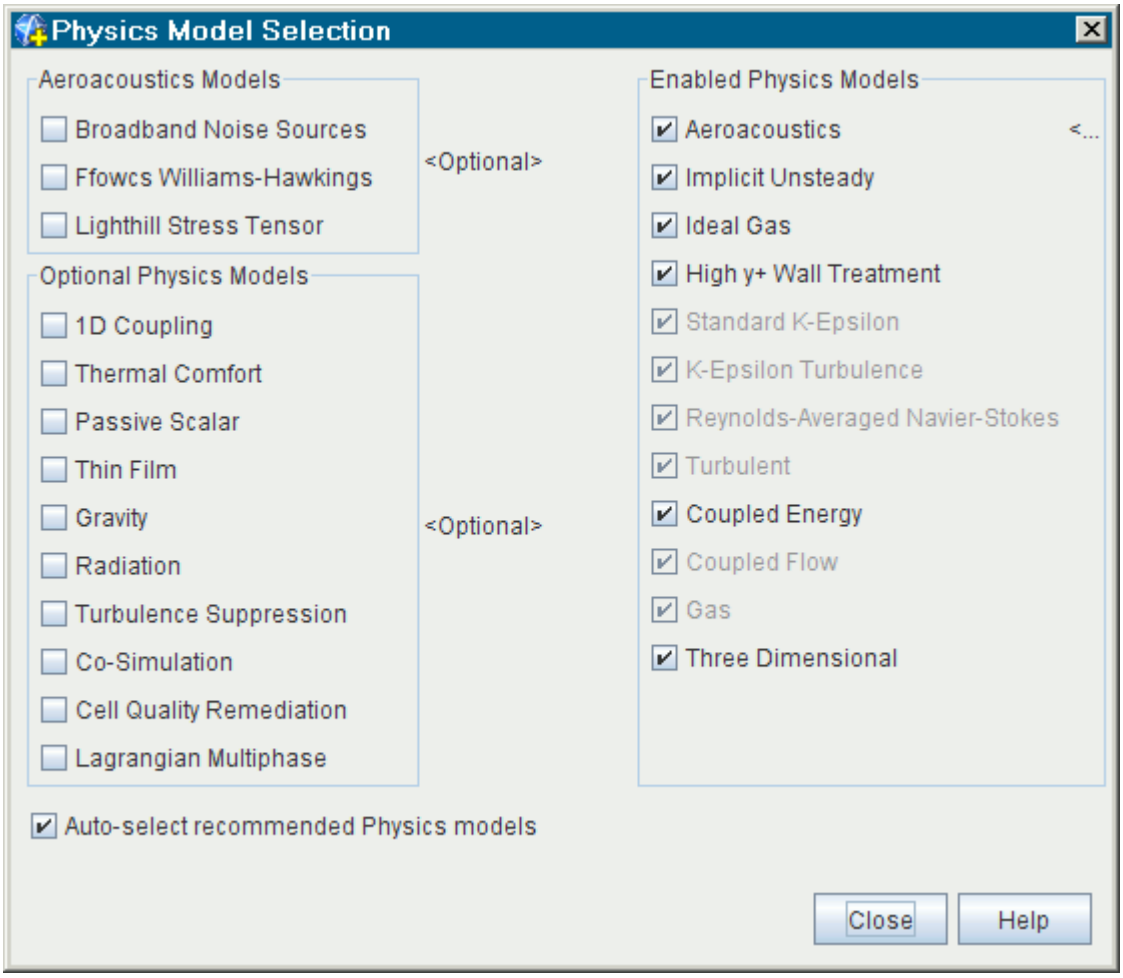
Lighthill Stress Tensor Model Properties

Far-Field Sound Speed	The speed of sound in the far-field: $a_0 = \sqrt{(\gamma R T_0)}$ The default value is 340 m/s.

Selecting the Lighthill Stress Tensor

The Lighthill Stress Tensor is available for steady and unsteady simulations, and with any viscous regime.

To make the **Lighthill Stress Tensor** option available, first select the Aeroacoustics model from among the *Optional Models* group.



Lighthill Stress Tensor Field Functions Reference

The following field functions are made available to the simulation when the Lighthill Stress Tensor is used.

Lighthill Stress Tensor XX	The sound the specific normal stress, shear, nonlinear acoustic generation processes, T_{xx}
Lighthill Stress Tensor XY	The sound the specific shear stress, shear, nonlinear acoustic generation processes, T_{xy}
Lighthill Stress Tensor XZ	The sound the specific shear stress, shear, nonlinear acoustic generation processes (activated only in three-dimensional simulations), T_{xz}
Lighthill Stress Tensor YY	The sound the specific normal stress, shear, nonlinear acoustic generation processes, T_{yy}
Lighthill Stress Tensor YZ	The sound the specific shear stress, shear, nonlinear acoustic generation processes (activated only in three-dimensional simulations), T_{yz}

Lighthill Stress Tensor ZZ	The sound the specific normal stress, shear, nonlinear acoustic generation processes (activated only in three-dimensional simulations), T_{zz}

Aeroacoustics Bibliography

[51] *Aybay, O., Caraeni, M. and Holst, S. 2011."Tandem Cylinder and Idealized Side Mirror Far-Field Noise Predictions Using DES and An Efficient Implementation of FW-H Equation", 17th AIAA/CEAS Aeroacoustics Conference, AIAA, Portland, Oregon (USA).*

[52] *Bailly, C. and Juve, D. 1999. "A Stochastic Approach to Compute Subsonic Noise using Linearized Euler's Equations", AIAA Journal, AIAA-99-1872.*

[53] *Baris, O., and Mendonca, F. 2011. "Automotive Turbocharger Compressor CFD and Extension Towards Incorporating Installation Effects", Proceedings of ASME Turbo Expo 2011: Power for Land, Sea, and Air. GT2011-46796.*

[54] *Bechara, W., Bailly, C., Lafon, P., and Candel, S. 1994. "Stochastic approach to Noise Modeling for Free Turbulent Flows", AIAA Journal, 32:3, pp. 455-463.*

[55] *Bechara, W., Lafon, P., Bailly, C., and Candel, S. 1994. "Application of a k-epsilon turbulence model to the prediction for simple and coaxial free jets", Journal Acoustics Soc. Am, 97:6, pp. 3518-3531.*

[56] *Bodony, D. J. 2006. "Analysis of Sponge Zones for Computational Fluid Mechanics", Journal of Comp. Physics, 212-2, pp. 681-702.*

[57] *Bres, G.A., Brentner, K.S., Perez, G. and Jones, H.E. 2004. "Maneuvering rotorcraft noise prediction", J Sound and Vibration, 275:3-5, pp. 719-738.*

[58] *Brentner, K.S. 1994. "A new algorithm for computing acoustic integrals", Proceedings of the 14th IMACS World Congress.*

[59] *Brentner, K.S. and Farassat, F. 1998. "Analytical comparison of the acoustic analogy and Kirchhoff formulation for moving surfaces", AIAA Journal, 36:8, pp. 1379-1386.*

[60] *Brentner, K.S. and Farassat, F. 2003. "Modeling aerodynamically generated sound of helicopter rotors", Progress in Aerospace Sciences, 39:2-3, pp. 83-120.*

[61] *Caridi, D. 2008. "Industrial CFD Simulation of Aerodynamic Noise", Doctoral Thesis, University of Naples Federico II.*

[62] *Casalino, D. 2003. "An advanced time approach for acoustic analogy predictions", J Sound and Vibration, 261:4, pp. 583-612.*

[63] *Colonius, T. 2004. "Modeling Artificial Boundary Conditions for Compressible Flow", Annual Review of Fluid Mechanics, 36, pp. 315-345.*

[64] *Curle, N. 1955. "The Influence of Solid Boundaries upon Aerodynamic Sound", Proceedings of the Royal Society of London, series A, Mathematical and Physical Sciences, 231, pp. 505-514.*

[65] *di Francescantonio, P. 1997. "A new boundary integral formulation for the prediction of sound radiation" J. Sound and Vibration, 202, 491-509.*

[66] *Farassat, F. 2007. "Derivation of formulations 1 and 1A of Farassat", Technical Memorandum TM-2007-214853, NASA, NASA Langley Research Center (USA).*

[67] *Farassat F., Padula S.L. and Dunn M.H. 1987. "Advanced Turboprop Noise Prediction Based on Recent Theoretical Results" J. Sound and Vibration, 119:1 pp. 53-79.*

[68] *Farassat F., Succi G. P. 1983. "The prediction of helicopter discrete frequency noise" Vertica, 7:4 pp. 309-320.*

[69] *Ffowcs Williams, J. and Hawkings, D. 1969. "Sound Generation by Turbulence and Surfaces in Arbitrary Motion", Philosophical Transactions of Royal Society of London, Series A, Mathematical and Physical Sciences (1934-1990), 264:1151, pp. 321-342.*

[70] *Frota, J., Lempereur, P. and Roger, M. 1998. "Computation of the noise of a subsonic propeller at an angle of attack", 4th AIAA/CEAS Aeroacoustic Conference, AIAA, Toulouse (France).*

[71] *Gennaro, M. and Caridi, D. 2010. "Ffowcs Williams-Hawkings Acoustic Analogy for Simulation of NASA SR2 Propeller Noise in Transonic Cruise Condition", ECCOMAS CFD 2010.*

[72] *Greschner, B., Thiele, F., Jacob, M.C., and Casalino, D., 2008. "Prediction of sound generated by a rod-airfoil configuration using EASM DES and the genralise Lighthill/FW-H analogy", Computers and Fluids, 37, pp. 402-413.*

[73] *Goldstein, M.E. 1976. Aeroacoustics, McGraw-Hill, New York.*

[74] *Goldstein, M.E. and Rosenbaum, B. 1973. "Effect of Anisotropic Turbulence on Aerodynamic Noise", J. Acoustical Society of America, 54, pp. 630-645.*

[75] *Hinze, J. O. 1975. Turbulence, McGraw-Hill, pp. 672-673.*

[76] *Hubbard, H.H. 1995. "Aeroacoustics of Flight Vehicles, Theory and Practice, Volume 1: Noise Sources" NASA Langley Research Center, Hampton, Virginia, pp. 13.*

[77] *Karweit, M., Blanc-Benon, P., Juve, D., and Comte-Bellot, G. 1991. "Simulation of the Propagation of an Acoustic Wave Through a Turbulent Velocity Field: A Study of Phase Variance", J. Acoustical Society of America, 89:1, pp. 52-62.*

[78] *Kim, J. W., Lau, A. S.H., Sandham, N.D. 2010. "Proposed Boundary Conditions for Gust-Airfoil Interaction Noise", AIAA Journal, 48-11.*

[79] *Kraichnan, R.H. 1970. "Diffusion by a Random Velocity Field", J. Computational Physics, 13:1, pp. 22-31.*

[80] *Lighthill, M. 1952. "On Sound Generated Aerodynamically. I. General Theory", Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences (1934-1190), 211:1107, pp. 564-587.*

[81] *Lighthill, M. 1954. "On Sound Generated Aerodynamically. I. Turbulence as a Source of Sound", Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences (1934-1190), 222:1148, pp. 1-32.*

[82] *Lilley, G.M. 1993. "The radiated noise from isotropic turbulence revisited", NASA Contract Report No. 93-75, NASA Langley Research Center.*

[83] *Lin, P.H. 2011. "Computational Aero- and Hydro-acoustics by Space-Time Conservation Element and Solution Element Method", Ph. D. Thesis, Ohio State University.*

[84] *Lockard, D. P. 2002. "A Comparison of Ffowcs Williams-Hawkings Solvers for Airframe Noise Applications", 8th AIAA-/CEAS Aeroacoustics Conference, AIAA-2002-2580.*

[85] *Mankbadi, R.R. 1993. "Computational aero- and hydro-acoustics" ASME United Engineering Center.*

[86] *Marinus, B. and Roger, M. 2009. "Truncated Method for Propeller Noise Prediction Up to Low Supersonic Helical Tip Mach Numbers", AIAA-2009-3330.*

[87] *Mendonca, F., Baris, O., and Capon, G. 2012 "Simulation of Radial Compressor Aeroacoustics Using CFD.", Effects", Proceedings of ASME Turbo Expo 2011: Power for Land, Sea, and Air. GT2012-70028.*

[88] *Mendonça, F., Read, A., Caro, S., Debatin, K. and Caruelle, B. 2005. "Aeroacoustic Simulation of Double Diaphragm Orifices in an Aircraft Climate Cooling System", AIAA-2005-2976.*

[89] *Pilon A. and Lyrintzis A. 1998. "Development of an Improved Kirchhoff Method for Jet Aeroacoustics", AIAA Journal, 36:5, pp.783-790.*

[90] *Proudman, I. 1952. "The Generation of Noise by Isotropic Turbulence", Proc. R. Soc. Lond. 214, pp. 119-132.*

[91] *Ribner, H.S. 1964. "The Generation of Sound by Turbulent Jets", in Advances in Applied Mathematics, Academic Press, New York.*

[92] *Robinson, A.R. and Lee, D. 1994. "Oceanography and acoustics: prediction and propagation models", American Institute of Physics.*

[93] *Roger, M. 2006 "Fundamentals of aeroacoustics. In Experimental Aeroacoustics", M. L. Riethmuller and M. R. Lema editors, LS 2007-01. von Karman Institute, Rhode-St-Genese.*

[94] *Sarkar, S., Hussaini, M.Y. 1993. "Computation of the Sound Generated by Isotropic Turbulence", NASA Report 191543, ICASE Report No. 93-74.*

[95] *Witkowska, A., and Juve, D. 1993. "Numerical Simulation of Noise Generated by Homogeneous and Isotropic Turbulence,' Proc. 13th Colloquium on Aero- and Hydro-Acoustics, Lyon.*

[96] *Wagner, Hüttl, Sagaut (2007). "LES for Acoustics", Cambridge University Press, ISBN-13: 9780521871440.*

[97] *Zorumski W.E. and Weir D.S. 1986. “Aircraft Noise Prediction Program Theoretical Manual-Propeller Aerodynamics and Noise” NASA TM-83199, 3.*

Modeling Hydroacoustics

Hydroacoustics is the study of sound generation and propagation in water.

See [92]. Sound comes from marine wildlife, by SONAR, and by ships [85]. A sound pressure wave moves through water at 1,484 m/s, over 4 times faster than the speed of sound in air. Also, due to slower dissipation sound can travel much longer distances underwater than is possible in air.

Setting Up a Hydroacoustics Analysis

Modeling hydroacoustics in STAR-CCM+ is similar to the process for modeling aeroacoustics. Both types of simulation require that you model the full unsteady flow surrounding the geometry from which sound propagates [83]. However, for your hydroacoustics analysis, use the double precision version of STAR-CCM+ to simulate the generation of acoustic waves in turbulent water flows. Also, in hydroacoustics, use a compressible liquid such as IAPWS-IF97 (Water) to model the transmission of sound.

To model sound wave propagation through water:

- Launch the double precision version of STAR-CCM+.
- Choose an equation of state that supports compressibility.

  An **IAPWS-IF97 (Water)** model or a **User Defined EOS** model with compressibility enabled are both appropriate for hydroacoustics analyses.
- Use the recommended analysis procedure for aeroacoustics to define a suitable mesh and time-step and solve the case in both the near-field and far-field.

Caution: When setting up a hydroacoustics simulation pay special attention to the default acoustics properties, many of which use values for air. For instance, the **Far-Field Sound Speed** FW-H property is set to 340 m/s by default.

Combustion and Other Reacting Flows

STAR-CCM+ provides a selection of models that you can use to simulate a wide range of reacting flow applications.

In STAR-CCM+, there are two specific categories of reacting flow models, Flamelet, and Reacting Species Transport. Whenever applicable, you use the flamelet models, otherwise, you use the reacting species transport models. To decide on a modeling approach, it is important to consider the relation between the rate at which species react and the rate at which species mix (the chemistry time-scales and turbulence time-scales).

You can create simple chemical mechanisms in STAR-CCM+ either manually or using the flamelet table generators, or you can import more complex chemical mechanisms from another source, such as DARS. A selection of ignitors are also available to use with most combustion models in STAR-CCM+.

Flamelet

In STAR-CCM+, the chemistry is calculated using detailed chemical mechanisms in simple 0D or 1D laminar flamelet geometries. The results are then tabulated and interpolated in a 3D simulation of a turbulent flame.

Flamelet Models	Description	Applications
Flamelet Generated Manifold	Mainly for premixed or partially-premixed flames where the flamelet assumption is valid.	• Gas Turbines • Furnaces • Burners
Steady Laminar Flamelet	Mainly for non-premixed combustion where the flamelet assumption is valid.	• Furnaces • Burners
Chemical Equilibrium	When only the temperature is of interest, or there is no chemical mechanism available.	• Fires • Coal Combustors

The following flame propogation models are also available:

- Turbulent Flame Closure
- Coherent Flame Model

See, Flamelet.

Reacting Species Transport

STAR-CCM+ solves transport equations for all species and applies a separate mechanism for predicting the chemistry. Possible mechanisms range from simple one or two step reactions to those that require complex chemistry.

Reacting Species Transport Models	Description	Applications
Complex Chemistry	For capturing transient phenomena and slowly forming species in gaseous combustion.	• CI ICE • Gas Turbines • Multifuel Combustion
Eddy Break-Up	For quick flame positioning in gaseous combustion.	• Gas Turbines • Furnaces • Burners
Thickened Flame Model	For premixed or partially premixed flames in Large Eddy Simulations (LES), when flamelet models cannot be used.	• Gas Turbines • Explosions
Eddy Contact Micromixing	For liquid-liquid chemistry, where the chemistry is fast compared to mixing.	Liquid-Liquid Reactors
Polymerization	For polymerization reactions.	Polymerization reactors

See, Reacting Species Transport.

Other Reacting Flow Models

You can also use other types of reacting flow models alongside the Flamelet, and Reacting Species Transport models:

- **Emissions**: STAR-CCM+ provides specific emissions models that you can use with the specific reacting flow models to simulate slowly forming pollutant species, such as Soot, NOx Thermal, NOx Prompt, and NOx Fuel.

  See, Emissions.
- **Surface Chemistry**: Surface chemistry describes the interaction between fluid molecules and a solid surface material. You can model surface chemistry independently or as an additional option alongside other categories of reaction models.

  See, Surface Chemistry.
- **Reacting Channels**: The Reacting Channels model is designed to simulate chemical processes that occur within long narrow tubes.

  See, Reacting Channels.
- **Interphase**: By using combustion models in combination with relevant multiphase models, you can simulate reacting solid particles or fluid droplets.

  See, Modeling Multiphase Flow.

Optional Reaction Models	Description	Applications
Emissions	For NOx and soot prediction.	• Gas Turbines • Furnaces • Burners
Particle Reactions (see Interphase)	For combustion of solid particles	• Coal Combustion • Biomass Combustion • Gasification
Surface Chemistry	For catalyses and surface deposition or etching.	• Aftertreatment • CVD
Reacting Channel	For fast calculation (1D) of chemical reactions in long channels.	• Steam Reforming • Ethylene Cracking
Interphase Reactions	For phase-changing reactions.	• Gas Oil Cracking • Fluidized Bed

Contents:

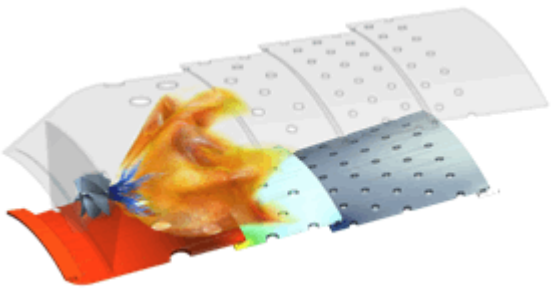
Reacting Flow Applications
Reacting Flow General Workflow
Reacting Flow General Reference
Reacting Species Transport
Flamelet
Emissions
Interphase
Surface Chemistry
Reacting Channels
Common Reacting Flow Actions

Reacting Flow Applications

It is possible to model many different types of reacting flow applications using combinations of the reacting flow models that are provided in STAR-CCM+.

The reacting flow general workflow describes how to begin modeling all types of reacting flow applications in STAR-CCM+. See Reacting Flow General Workflow. Each specific type of reacting flow application follows an additional workflow, which you decide based on the specific application. A selection of common reacting flow applications and guidelines for selecting a suitable workflow are described in the following sections.

Gas Turbines



You can follow either of the following workflows:

- Reacting Species Transport

  Best for gas turbines at part load.

  Use any of the following models:

  - Complex Chemistry (computationally expensive, so use only when necessary)
    - when there is a mixture of multiple fuels
    - to simulate flame ignition, quenching, or extinction
    - to simulate slowly forming pollutants
    - to simulate off-design load points
  - EBU

    Use if the reactions are predominantly limited by turbulence and the mechanism is small (one or two steps), otherwise, use Complex Chemisty.
  - TFM

    Useful for transient LES simulations. This model thickens flames artificially—which can help to reach a converged solution when the other flamelet models cannot.

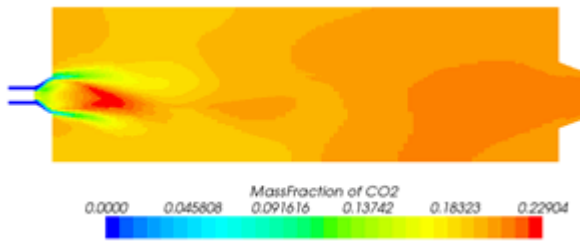
  See Reacting Species Transport.
- Flamelet

  Best for gas turbines at full load or in which statistically-steady turbulent combustion occurs.

Use the FGM model.

See Flamelet.

Coal Combustion



You can follow one of the following workflows:

- Interphase

  Best for modeling interactions between the solid coal particles and the background gas, such as in coal furnaces. You can select either of the following models:

  - Coal Combustion

    Best for simulating coal furnaces. You can also model NOx emissions that are designed specifically for coal fuel.
  - Particle Reaction
    - Particle Combustion: to simulate particle combustion or char oxidation.
    - Particle Devolatilization: to simulate particle devolatilization, gasification, or sublimation.
- Reacting Species Transport

  You can simulate multi-fuel combustion with the complex chemistry model.

  See Reacting Species Transport.
- Flamelet

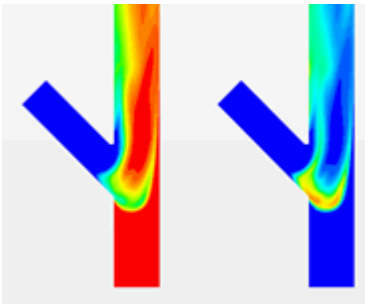
  You can use the chemical equilibrium flamelet model for simulating simple coal combustion or fires.

  See Flamelet.

Additionally, if required, you can model Emissions alongside the Flamelet or Reacting Species Transport workflows. The Interphase workflow provides the option to model specific NOx emissions.

Fluid Catalytic Cracking

Example application: gasoline production.



You can follow one of the following workflows:

- Interphase

  Best for modeling interactions between the liquid droplet particles and the background gas.
- Reacting Species Transport

  You can simulate multi-fuel combustion with the Complex Chemistry model.

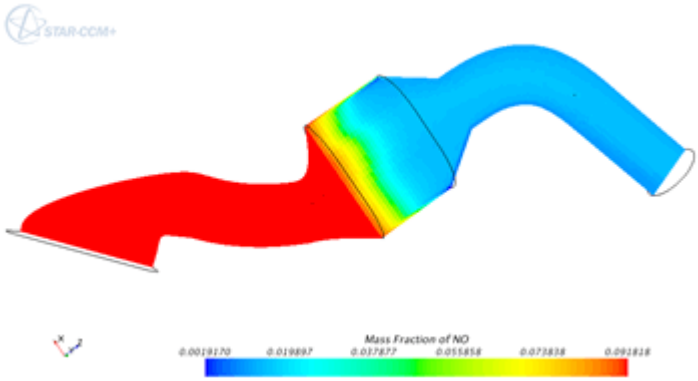
  See Reacting Species Transport.

Polymerization

Use the Polymerization workflow.

The Polymerization model allows you to simulate all relevant processes, including initiation, propagation, transfer, branching, scission, and termination.

Catalytic Convertors

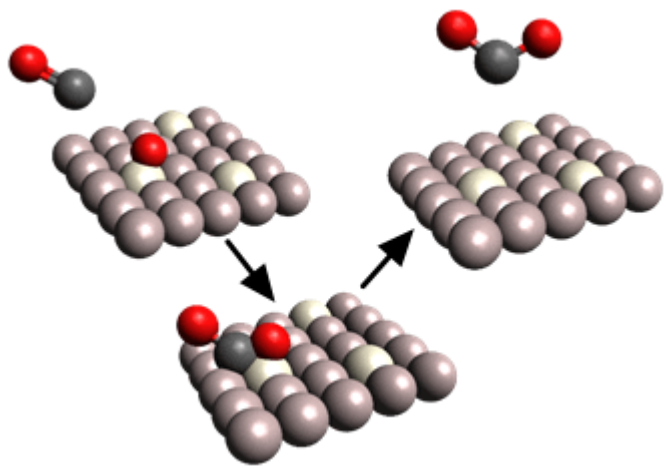


Use the Reacting Species Transport workflow—which allows you to model transport of all species, with the additional Surface Chemistry and Emissions workflows.

See Reacting Species Transport, Surface Chemistry, Emissions.

You can specify the reacting surfaces as walls within solid regions or surfaces in porous regions.

Reacting surfaces



Use the Reacting Species Transport workflow—which allows you to model transport of all species, with the additional Surface Chemistry workflow.

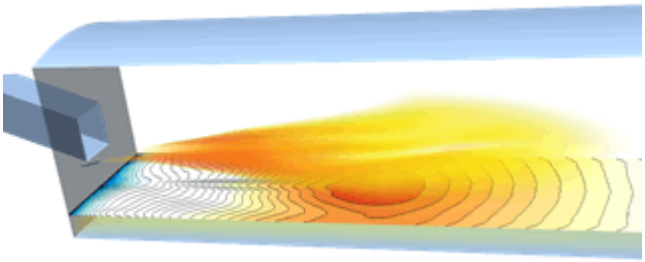
See Reacting Species Transport, Surface Chemistry.

If you want to monitor emissions, you can also use the Emissions workflow alongside the Reacting Species Transport workflow.

See Emissions.

The reacting surfaces can be walls within solid regions or surfaces in porous regions.

Glass Furnaces



You can follow one of the following workflows:

- Reacting Species Transport
  - Eddy Break-Up

Use if the reactions are predominantly limited by turbulence and the mechanism is small (one or two steps).

- Complex Chemisty

  Use with a Chemkin formatted mechanism.

See Reacting Species Transport.

- Flamelet

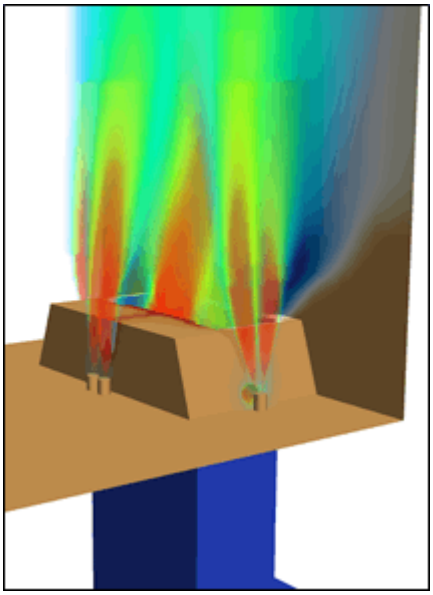
  You can use the Steady Laminar Flamelet model for steady state opposed flow diffusion flames with tabulated detailed kinetics.

  See Flamelet.

If you want to monitor the emission levels of particular species, you can additionally use the Emissions workflow.

See Emissions.

Furnaces and Burners



You can follow one of the following workflows:

- Flamelet (use with tabulated detailed kinetics)
  - Steady Laminar Flamelet

    for steady state opposed flow diffusion flames.
  - Flamelet Generated Manifold

    Suitable for all premixed or partially premixed flames where the flamelet assumption is valid (there are infinitely fast chemical reactions). Use if you want to capture ignition, extinction, or quenching.

  See Flamelet.

- Reacting Species Transport (Use if none of the flamelet models are suitable)
  - Complex Chemistry

    Use with a Chemkin formatted mechanism, for transient calculations, ignition, and extinction.
  - Eddy Break-Up

    Use with one or two simple reactions that you define in the GUI.

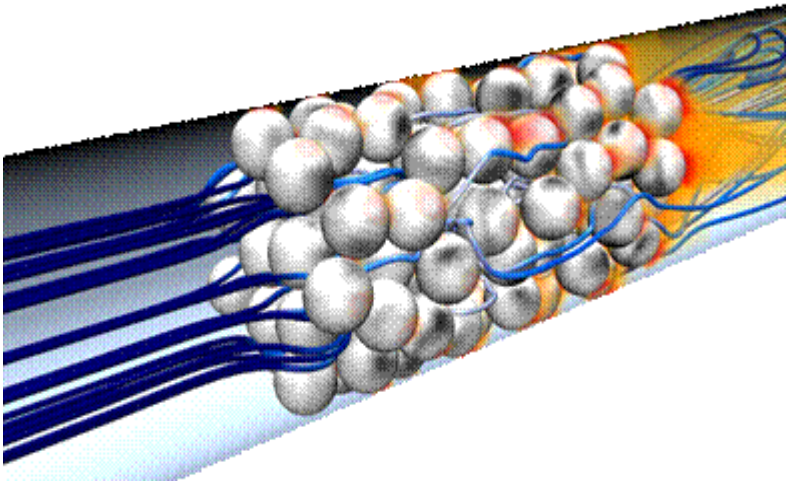
  See Reacting Species Transport.

If you want to monitor the emission levels of particular species, you can additionally use the Emissions workflow.

See Emissions.

Reacting Channels

Example application: Methane reformation

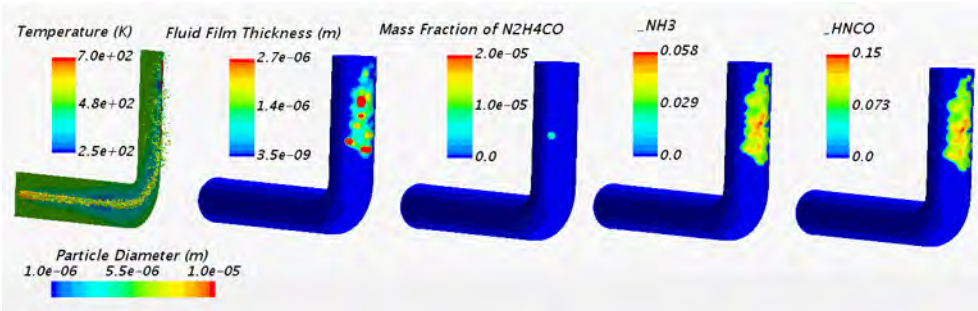


Use the Reacting Channels workflow which describes how to set up the reacting channels and the firebox volume that surrounds the channels.

See Reacting Channels.

Fluid Film Reactions

Example application: trickle bed reactors or selective catalytic reduction (SCR) reactions in urea film.



You can simulate reactions within a fluid film (intraphase) using the Reacting Species Transport workflow with the optional model, Fluid Film. You use the Multi-Component Gas model for the background fluid region.

See Reacting Species Transport.

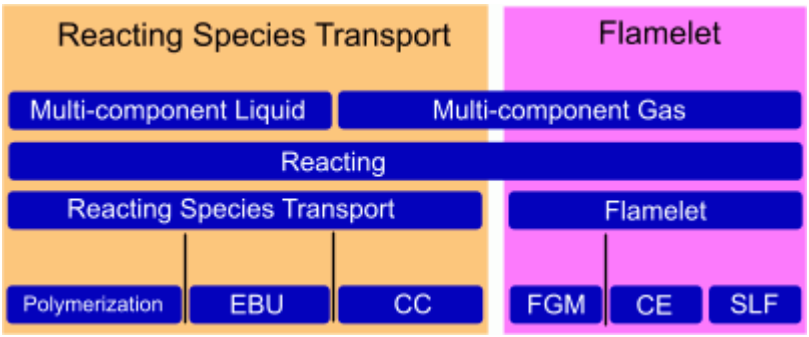
Reacting Flow General Workflow

STAR-CCM+ supports several types of combustion analysis, each with their own workflow. Use this section to help identify the choice of models that are appropriate for the type of analyses you intend.

Import or create geometry parts that represent the volume in which the reacting flow occurs. You can use STAR-CCM+ in serial or parallel mode.

Before selecting physics models and setting up physics parameters, it is important to have appropriate geometry and a suitably refined mesh in STAR-CCM+. The detailed workflow that you follow depends on the physics of the application that you intend to simulate. STAR-CCM+ provides a wide range of physics models that you can use in different combinations to suit different types of reacting flow processes. Follow the advice that is given in the introductory pages of the Reacting Flow section to help you determine which specific detailed workflow to use in addition to this workflow.

The following diagram provides an overview of the initial physics models that are required in the two most common detailed reacting flow workflows.

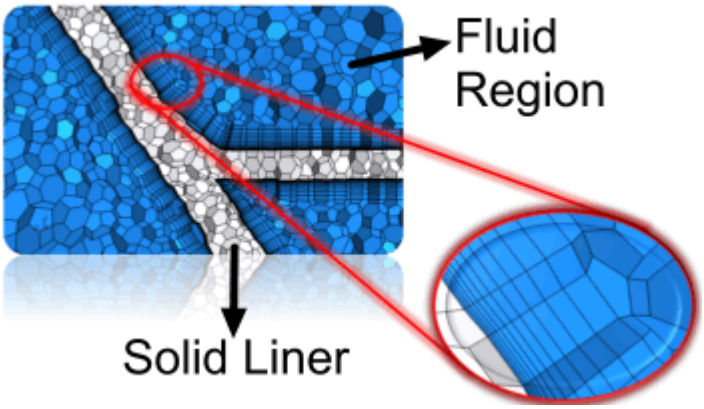


1. Generate a quality mesh to represent the reacting flow volume.

   A polyhedral mesh is suitable for modeling most combustion applications as polyhedral cells generally have a low skewness and can calculate a more accurate face flux than tetrahedral cells. Polyhedral cells also provide a smooth volumetric growth, which is important for capturing complex geometry effectively.

   It is important to consider refining the mesh by increasing the resolution in significant areas or volumes of the geometry. For example:

   - Refine mesh on and around fuel jets.
   - Create a higher resolution mesh in narrow volumes.
   - Refine mesh on impellers and baffles.
   - In simple geometries, where mixing occurs, use volumetric refinement.
   - In complex geometries, where mixing occurs, use solution based refinement.
   - When modeling combustion with conjugate heat transfer, create a conformal mesh at the interface. It is also necessary to capture the boundary layers by using the Prism Layer mesher to create orthogonal prismatic cells next to the walls.



2. Decide which detailed workflow is most suitable for the application that you are modeling.

   It is useful to consider how and where the reactions are happening in the application that you are simulating, as these details help to elicit the most appropriate workflow.

Consider the type of reacting flow models that are available to use within the specific detailed workflow that you have decided to follow. When you proceed to the specific detailed workflow, it is useful to have an idea about which models you intend to use.

3. Before following the steps within one of the specific detailed workflows, for the physics continuum that represents the reacting flow volume, start to select appropriate general physics models:

Group Box	Model
Material	For Flamelet workflows, select **Multi-Component Gas**. For Reacting Species Transport or Reacting Channels workflows, select **Multi-Component Gas** or **Multi-Component Liquid**. For reacting channels, you select the type of material that best represents the reacting flow within the fluid domain that surrounds the reacting channels (not the material within the reacting channels). For Lagrangian Multiphase Interphase workflows, select the material type that surrounds the reacting particles—not the material of the reacting particles: • **Multi-Component Gas** • **Multi-Component Liquid** • **Liquid** • **Gas** For Eulerian Multiphase Interphase workflows, select **Eulerian Multiphase**. For Polymerization workflows, select **Liquid**. For Surface Chemistry workflows, select **Multi-Component Gas**.
Space	• **Axisymmetric** • **Three Dimensional** • **Two Dimensional**

Group Box	Model
Time	• **Implicit Unsteady** (required for Lagrangian Multiphase Interphase workflows). • **PISO Unsteady** (Segregated Flow Only) • **Steady** When any Unsteady model is selected with a Flamelet reacting flow model, the Ideal Gas is compressible, and when Steady is selected, the Ideal Gas is incompressible.
Reaction Regime (for Multi-Component Gas or Multi-Component Liquid)	For the Flamelet, Reacting Species Transport, and Reacting Channels workflows, select **Reacting**. For the Lagrangian Multiphase Interphase workflow, select **Non-Reacting**. The reactions are defined in the Lagrangian phase. For the Surface Chemistry workflow: • If the gaseous mixture is only undergoing surface reactions and there are no other reactions, select **Non-Reacting**. • If reactions are occurring within the volume of the gaseous mixture in addition to the surface reactions, select **Reacting**.

4. Proceed to the specific reacting flow workflow that you have decided to follow—keeping in mind any specific models that you think are appropriate for the application that you are simulating.

   In addition to the reacting flow physics models that you select in the specific reacting flow workflow, as necessary, you also select the following general physics models:

Group Box	Model
Flow	Either: • **Coupled Flow** • **Segregated Flow** (recommended when using the NOx Emission model) Depending on the selection, further models are selected automatically.
Equation of State	• **Constant Density** • **Polynomial Density** • **User Defined EOS**
Viscous Regime	One of: • **Inviscid** • **Laminar** • **Turbulent** Depending on the selection, further models are required.

- Reacting Species Transport Workflow

- Polymerization Workflow
- Flamelet Workflow
- Interphase Workflow: Lagrangian Multiphase
- Interphase Workflow: Eulerian Multiphase
- Surface Chemistry Workflow
- Reacting Channels Workflow

After completing the steps in the specific reacting flow workflow, complete the set-up as follows:

5. For the continuum, define any necessary parameters for the **Reference Values** and **Initial Conditions**.
6. Define any necessary Physics Conditions and Values for the Region and Boundaries.

   For example:

   - For a wall boundary, you can define any flux at the boundary using the boundary physics condition, **Wall Combustion Scalar Option**.
   - For regions in which reacting flows are affected by a momentum source such as a fan, you can define the momentum source using the region physics condition, **Momentum Source Option**.
   - To modify the size of chemistry time-steps for steady-state simulations, modify the CFL Scaling Factor—available to specify as a physics value for regions which use the Complex Chemistry model and have the physics condition, *CFL Factor Option*, activated. See Complex Chemistry Model Reference.
   - If you are modeling the Fuel NOx for a purely gaseous fuel, it is important to specify values for the inlet composition which include the intermediate species, HCN and/or NH3.
7. Specify the required Solver settings and Stopping Criteria.

   a) Select the **Stopping Criteria** > **Maximum Steps** node and set the *Maximum Steps*.

      If you are unsure of what value to set, maintain the default. If the solution converges before the maximum number of steps, you can stop it manually and note the number of steps to use the next time you run the simulation. If the solution has not converged, you can increase the maximum number of steps and continue running the simulation.

   b) Expand the Solvers node and adjust any solver parameters that are necessary. If you are unsure, maintain the default values.

   Note: Make sure that the Under-Relaxation Factor property for the energy and species solvers are both set to the same values.

8. Set up any Reports, Monitors, Plots, and Scenes that are required for Post-Processing.

   For example:

   - To determine if a reacting flow simulation solution is converging, you can generate a heat balance plot from a report.
     1. Right-click the **Reports** node and select **New Report** > **Heat Transfer**.
     2. Select the Heat Transfer 1 node and set Parts to all of the inlet and outlet boundaries for the reacting flow region.
     3. Right-click the **Reports** > **Heat Transfer 1** node and select **Create Monitor and Plot from Report**.
     4. Open the **Plots** > **Heat Transfer 1 Monitor Plot**.

When you run the simulation and the plot settles at a value that is close to zero, this shows that the heat at the inlet and outlet boundaries is reconciled and the solution has converged.

- To plot a specific field function within the reacting flow region, or at a boundary, you can generate an XY Plot. For example, you can monitor the temperature on the surface of reacting channels, the mass fraction of CO emissions at an outlet, or the progress variable within a reacting flow region. For more information, see Plotting Results.
  1. Right-click the **Plots** node and select **New Plot** > **XY Plot**
  2. Expand the **Plots** > **XY Plot 1** node and set at least the following parameters:

Node	Property	Setting
XY Plot 1	*Parts*	Select the region or boundary on which you want to monitor the chosen field function.
⌊ **Y Types**		
Y Type 1 > **Scalar Function**	*Field Function*	Select a field function that you want to monitor.

- You can visualize physics values on or within the reacting flow geometry using scalar scenes and vector scenes. For Example, you can visualize the flame area density within a scalar scene, or the velocity of the reacting flow within a vector scene. For more information, see Working with Scenes.

9. Save the simulation.
10. Run the simulation.

Troubleshooting

Initialising an Unsteady Calculation Through a Steady Calculation

Steady calculations converge best with the incompressible option activated, while unsteady calculations converge best with the incompressible option deactivated. When initialising an unsteady calculation through a steady calculation, the following workflow is advised:

1. Converge the steady calculation with the incompressible option activated (default option when the Steady model is used with a Flamelet model).
2. Deactivate the incompressible option and re-converge.
3. Switch to using the Unsteady model—keep the incompressible option deactivated.

Warning messages in the output window indicate that there are issues with the setup. To resolve the issues, follow the advice given below then re-run the simulation.

Temperature Out Of Bound

When the temperature is out of bounds, a message similar to the following is displayed: `Temperature limited to [maximum/minimum] value [n] times on model-part "[part]"`

Under **Reference Values**, edit the value for Minimum Allowable Temperature or Maximum Allowable Temperature.

Heat Loss Ratio Out Of Bound

When the heat loss ratio (HLR) is out of bounds, a message similar to the following is displayed: `Heat loss ratio limited to [maximum/minimum] value [n] times on model-part "[part]"`. If this message only appears at the start of a computation, or only occasionally throughout the computation, take no action.

- When HLR hits the lower limit (-1), the heat gain in the domain is more than is specified in the combustion table. Increase the combustion table heat gain range by increasing the temperature of the oxidizer in the stream.
- When HLR hits the upper limit (1), too much heat loss is occurring from the domain. Within the combustion table, expand the **Table Generator** > **Parameters** > **Numerical Settings** node and lower the value that is set for the Reference Temperature.

Reacting Flow General Reference

Some reference material is common to different sub-sections within the reacting flow material.

Contents:

Reacting Model Reference
Ignitors
Spark Ignition Model Reference
Reaction Mechanism Formats

Reacting Model Reference

You select the Reacting model for physics continua in which reactions occur in the bulk of the fluid.

Reacting Model Reference

Theory	See Reacting Flow.	
Provided By	**[physics continuum]** > **Models** > *Reaction Regime*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Reacting**	
Requires	*Material*: **Multi-Component Gas** or **Multi-Component Liquid**	
Activates	Physics Models	*Reacting Flow Models*
	Model Controls (child nodes)	Reactions and Reaction System Properties

Reacting > Reactions

The Reactions node contains details of the reactions in the chemical mechanism. You can define these reactions manually, or they appear upon importing a chemical mechanism. See Importing Species and Reactions or Defining Chemical Reactions Manually.

[Reaction]

Each **Reactions** > **[Reaction]** node contains three sub-nodes: Reactants, Products, and Properties.

Properties

The Properties node contains further sub-nodes, Complex Reaction Properties, and Reaction Coefficient. When Third Bodies is activated under Arrhenius Coefficients, a Third Body Efficiencies sub node is also available. If pressure-dependent reactions are defined in the imported chemical mechanism,

another node also appears, named after the specific method that is used to specify the pressure-dependent parameters—either, Lindemann Fall-Off Formulation, Troe Fall-Off Formulation, or SRI Fall-Off Formulation. The values for Lindemann, Troe, or SRI, are set automatically from the imported chemical mechanism. However, if necessary, it is possible to set these values manually.

Complex Reaction Properties

Pressure Dependent

Allows you to specify that this reaction uses a pressure-dependent Lindemann mechanism. When activated, the Lindemann Fall-Off Formulation node appears.

Reaction Coefficient

The Arrhenius reaction parameters are used to calculate chemical reaction rates from finite-rate kinetics.

Method	Corresponding Sub-Nodes
Arrhenius Coefficients	**Pre-exponent** Defines the pre-exponential factor, A, in Eqn. (3301).

Arrhenius Coefficients: Properties

Temperature Exponent, Beta

Sets the temperature exponent, β, for this reaction in Eqn. (3301).

Activation Energy, Ea

Sets the activation energy, E_a, for this reaction in Eqn. (3301).

Reversible

When activated, the reaction is specified as reversible. In a reversible reaction, you specify the forward rate coefficients and select how the backward rate is calculated using the **Properties** > **Reverse Reaction Coefficient** node. When reversible reactions are read from a Chemkin mechanism, this property is activated and the reaction properties are set automatically. If you define a reversible reaction, make sure that you specify values for the enthalpy, entropy, and specific heat for each component of the mixture. For the specific heat, you choose either the thermodynamic polynomial or polynomial in T option.

Third Bodies

By default, each species in the mixture is assumed to contribute with equal efficiency as a third body. Hence all species are given a third body efficiency of unity. If one or more species contribute more or less as a third body, you can activate this property and then specify the efficiency coefficient of that species under the Third Body Efficiencies node. If the imported mechanism contains details of third body efficiencies, this property becomes activated and the values are imported automatically.

The Arrhenius Coefficients node also provides the **Pre-exponent** sub-node that defines the pre-exponential factor, A, in Eqn. (3301).

Reverse Reaction Coefficient

Available when *Reversible* is activated under the **Reaction Coefficient** > **Arrhenius Coefficients** node.

Method	Corresponding Sub-Nodes
Reverse Coefficient By Equilibrium Constant	**Reverse Coefficient By Equilibrium Constant** Uses the Equilibrium Constant to calculate the backward reaction rate coefficient. This calculation requires that enthalpy, entropy, and specific heat are defined in the reaction definition.
Arrhenius Coefficients For Reverse Reaction	**Arrhenius Coefficients For Reverse Reaction** *Temperature Exponent, Beta* Sets the temperature exponent, β, for the backward reaction in Eqn. (3301). *Activation Energy, Ea* Sets the activation energy, E_a, for the backward reaction in Eqn. (3301). **Arrhenius Coefficients For Reverse Reaction > Pre-exponent** Defines the pre-exponential factor, A, for the backward reaction in Eqn. (3301).

Third Body Efficiencies

If third body efficiencies are imported as part of the chemical mechanism, details of the third body species and their efficiency coefficients are displayed under this node. If necessary, you can define third body efficiencies manually by right-clicking this node and selecting the option, **Add Third Body Coefficient** > **[species (continuum)]**. Each species that you include appears as a sub-node for which you can specify the third body efficiency coefficient, α_{ij}, in Eqn. (3303).

Lindemann Fall-Off Formulation

Specifies values for the Arrhenius rate parameters for the high-pressure limits (A_∞, β_∞, and E_∞ in Eqn. (3309)) and low-pressure limits (A_0, β_0, and E_0 in Eqn. (3308)), to account for the pressure-dependence of the rate constants, A_j, β_j, and E_j, respectively, in Eqn. (3301). Terminology within the reaction mechanism determines if the values represent high or low pressure limits.

Troe Fall-Off Formulation

Specifies values for α, T, $T*$, $T**$, and also optionally $T***$, in Eqn. (3311), which are used to calculate the blending factor, F, in Eqn. (3310).

SRI Fall-Off Formulation

Specifies values for a, b, c, and also optionally d and e, in Eqn. (3312), which are used to calculate the blending factor, F, in Eqn. (3310).

Reacting > Reacting System Properties

Species Reaction Source Jacobian

When using user-coded reaction rates, it is strongly recommended to use the Numerical option for the Jacobian calculation. However, if you require the additional computational speed of an analytical Jacobian, STAR-CCM+ does allow the option of a user-code analytical Jacobian.

Method	Corresponding Sub-Nodes
Analytical An analytical Jacobian is used in the CVODE solver.	**Analytical** Allows you to specify the user function that defines the analytical Jacobian.
Numerical Jacobian The Jacobian is calculated numerically.	none

A Jacobian is a square matrix of derivatives of the reaction rate with respect to the species specific mole fractions (mass fraction divided by molecular weight), and temperature. See Eqn. (3347) and Eqn. (3348).

Species Reaction Sources

Method	Corresponding Sub-Nodes
Internal The STAR-CCM+ CVODE solver provides the reaction rate source terms for reacting species.	none
User-Defined Provides the **User-Defined Species Sources Specification** node which allows you to specify whether you are modifying the internally calculated reaction rate source terms, or defining entirely new reaction rate source terms.	**User-Defined** Allows you to specify a previously imported User Function file that defines the species reaction rate source terms. See Working With User Functions.

User-Defined Species Sources Specification

Available when the Species Reaction Sources Method is set to User-Defined.

Method	Corresponding Sub-Nodes
Calculate Species Sources	**Calculate Species Sources** Specifies that entirely new reaction rate source terms are to be defined with a previously imported User-Function library. In this case, the internal reaction rate sources are not calculated.

Modify Internal Species Sources	**Modify Internal Species Sources** Specifies that the existing reaction rate source terms are to be modified with a previously imported User-Function file—typically by multiplying the existing values with a constant.

You use the **Species Reaction Sources** > **User-Defined** node to select the User-Function file.

Contents:

User-Defined Reaction Rate Source Terms

User-Defined Reaction Rate Source Terms

You can either modify the internally calculated reaction rate source terms, or define entirely new reaction rate source terms.

There are two kinds of user-coded reaction rate functions.

- **Modified Reaction Source Terms**

  STAR-CCM+ initially evaluates the internal Arrhenius species sources, and copies this to the source array passed in the function argument list. In the example below, this user function is referred to as `wdotModify()`—although any name can be given in the user code. `wdotModify()` is appropriate when modifying the internal Arrhenius species sources by a factor, which is typically a constant.
- **Custom Reaction Source Terms**

  STAR-CCM+ does not initialize the source array passed in the argument to the function. Below, this user function is referred to as `wdotCalculate()`, although any name can be given in the user code. `wdotCalculate()` is appropriate when computing reaction species sources that are not simply a multiple of the internal Arrhenius rate sources.

The function signature for user-defined reaction rate source terms, for both *wdotModify()* and *wdotCalculate()*, is:

```
 void wdotModify(double *omega, double T, unsigned int nSpe, double
*zi, void *data)
```

where,

	Reactions in the Gas Phase	Surface Reactions
`nSpe` (type unsigned int)	The number of species in the mixture.	`nSpe = nGas + nSite` (where `nGas`, `nSite`, and are the number of gas site species, surface site species, and bulk species respectively).
`zi`	A pointer to an array (type double, size `nSpe`) of specific mole fractions = mass fractions of the i'th species divided by molecular	A vector of gas mass fractions, surface site fractions, and cell temperature.

	Reactions in the Gas Phase	Surface Reactions
	weight of the i'th species (units of molecular weight are kg/kmol).	
`omega`	A pointer to the output array (type double, size `nSpe`) of the species source terms, which are the rate of change of specific mole fractions (mass fractions of the i'th species divided by molecular weight of the i'th species)	A source vector of gas mass fraction and surface site fraction. In the case of porous chemistry, omega also has (as the last element) cell enthalpy source term (J/kg-s).
`T` (type double)	Temperature (K)	Wall temperature (K)
`data`	A pointer to the `UserAccessibleData struct` that is defined in the section, UserAccessibleData.	

UserAccessibleData

```
#ifndef _UserAccessibleData_h
#define _UserAccessibleData_h


struct UserAccessibleData {
   double _pressure;
   double _dt;
   double _volume;
   double _volumeFraction;
   double _turbLen;
   double _turbTime;


   struct SurfaceChemistryData {

     double _area;
     unsigned int _nGas;
     unsigned int _nBulk;
     unsigned int _nSite;
     unsigned int _isPorous;
     double *_bulkFraction;
   } _surfaceChemistryData;


   struct ReactionData {
     unsigned int _nReactions;

     unsigned int *_nGasReactants;
     unsigned int *_nGasProducts;
```

```
     unsigned int *_nSiteReactants;
     unsigned int *_nSiteProducts;
     unsigned int *_nBulkReactants;
     unsigned int *_nBulkProducts;

     double **_gasStoichiometryReactants;
     double **_bulkStoichiometryReactants;
     double **_siteStoichiometryReactants;
     double **_bulkReactantRateExponents;
     double **_siteReactantRateExponents;
     double **_gasReactantRateExponents;
     unsigned int **_gasReactantSpeciesIndex;
     unsigned int **_siteReactantSpeciesIndex;
     unsigned int **_bulkReactantSpeciesIndex;

     double **_gasStoichiometryProducts;
     double **_bulkStoichiometryProducts;
     double **_siteStoichiometryProducts;
     double **_bulkProductRateExponents;
     double **_siteProductRateExponents;
     double **_gasProductRateExponents;
     unsigned int **_gasProductSpeciesIndex;
     unsigned int **_siteProductSpeciesIndex;
     unsigned int **_bulkProductSpeciesIndex;
   } _reactionData;

};

#endif
```

Note: The values in `UserAccessibleData` have SI units.

Modifying Reaction Source Terms

The following text is an example of a user-coded reaction rate function which modifies the internal Arrhenius species sources to be double the value that is defined by STAR-CCM+.

```
    void wdotModify(double *omega, double T, unsigned int nSpe, double
*zi, void *data)
    {
            // modify the internal Arrhenius reaction rate to be p /
pref
            struct UserAccessibleData *udata = (struct
UserAccessibleData *)data;
         double const factor = data->_pressure / 101325.;
```

```
            for( unsigned int i=0; i<nSpe; i++ )
                zi[i] *= factor;
    }

    void wdotJacModify(double *jac, double T, unsigned int nSpe,
double *zi, void *data)
    {
         // modify the internal Arrhenius reaction rate to be p / pref
         struct UserAccessibleData *udata = (struct UserAccessibleData
*)data;
            double const factor = data->_pressure / 101325.;

         for( unsigned int i=0; i<(nSpe+1)*(nSpe+1); i++ )
                jac[i] *= factor;
    }

    void uclib()
    {

     ucfunc(wdotModify, "ReactionRate", "wdot");
     ucfunc(wdotJacModify, "ReactionRate", "wdotJac");
    }
```

Creating Custom Reaction Source Terms

When using user-coded reaction rates, it is strongly recommended to use the (default) Numerical option for Species Reaction Source Jacobian (under the **Reacting** > **Reacting System Properties** node). If you want the additional computational speed of an analytical Jacobian, STAR-CCM+ does allow the option of using a user-code analytical Jacobian. The functions `wdotJacCalculate()` and `wdotJacModify()` must be provided, with the following signature:

```
 void wdotJacModify(double *jac, double T, unsigned int nSpe, double
*zi, void *data)
```

The double array Jacobian, of size $(nSpe+1)^2$, is the Jacobian of the vector of the species source terms plus the temperature source term. The species source terms are the rate of change of specific mole fractions (mass fractions of the i'th species divided by molecular weight of the i'th species).

Ignitors

You can control the ignition of some combustion simulations in STAR-CCM+ using ignitors.

When combustion models are selected that allow you to control ignition, a **Continuum** > **Ignitors** node appears. The **Ignitors** node allows you to select a specific type of ignitor according to the combustion physics models that are selected. Mesh cells which are contained within the ignitor volume shapes have their properties changed as specified by the ignitor when it is on.

- **EBU Ignitor**: When the EBU Ignitor is on, for the cells which are contained within the EBU ignitor volume shapes, it forces reaction rates to be mixing limited without using products.

- **Fixed Temperature Ignitor**: When the Fixed Temperature Ignitor is on, it allows you to set a constant temperature for the cells that are contained within the ignitor volume shapes.
- **Flame Area Density Ignitor**: When the Flame Area Density ignitor is on, it sets the method by which STAR-CCM+ determines the final Flame Area Density (FAD) that the ignitor applies to its cells. When it is off, this ignitor does not affect the values of flame area density.
- **Progress Variable Ignitor**: When the Progress Variable Ignitor is on, it sets the value of the cells that are contained within the ignitor volume shapes to that of the Ignitor Variable Value property that is set for the ignitor. When it is off, this ignitor does not affect the values of the progress variable.
- **Subgrid Spark Ignitor**: The Subgrid Spark Ignitor is available when either the Complex Chemistry model or Flamelet Generated Manifold model are selected along with the Spark Ignition physics model and any Unsteady model. See Ignitors.

Model		EBU Ignitor	Fixed Temperature Ignitor	Flame Area Density ignitor	Progress Variable Ignitor	Subgrid Spark Ignitor
Eddy Break-Up	Standard EBU	**Yes**	No	No	No	No
	Hybrid	**Yes**	**Yes**	No	No	No
	Combined Time-Scale	**Yes**	**Yes**	No	No	No
	Kinetics Only	No	**Yes**	No	No	No
Complex Chemistry		No	**Yes**	No	No	**Yes***
Thickened Flame Model		No	**Yes**	No	No	No
Coherent Flame Model		No	No	**Yes**	No	No
Flamelet Generated Manifold		No	No	No	**Yes**	**Yes***
Turbulent Flame-Speed Closure		No	No	No	**Yes**	No

* When the Unsteady model is also used.

Ignitor Properties Lookup

Property	EBU Ignitor	Fixed Temperature Ignitor	Flame Area Density ignitor	Progress Variable Ignitor	Subgrid Spark Ignitor
Activator	✓	✓	✓	✓	See Spark Ignition Model Reference.
Ignitor Set Method			✓	✓	
Ignitor Variable Value			✓	✓	
Shapes	✓	✓	✓	✓	

Property	EBU Ignitor	Fixed Temperature Ignitor	Flame Area Density ignitor	Progress Variable Ignitor	Subgrid Spark Ignitor
Temperature	✔	✔			

Properties

Activator

Selects between options that define when the ignitor is active (on).

- **Always**: Indicates that the Ignitor is always on.
- **Pulse**: Applies the ignition condition for the duration of a pulse that you specify using the *Begin* and *End* properties of the Pulse child node. For steady-state simulations, the properties specify iterations, and for unsteady simulations, the properties specify physical time. The activator is on for iterations when Begin <= iteration <= End, and for physical time when Begin < time <= End.

Ignitor Set Method

Sets the method to determine the value of flame area density (FAD) in each cell.

- **Maximum of Cell FAD or Ignitor Constant**: Uses the greater of the cell FAD that is obtained from the transport equation and the ignitor constant that is set with the Ignitor Variable Value property.
- **Ignitor Constant**: Overrides the cell value of FAD using the Ignitor Variable Value. This option is helpful when you know the exact FAD value for one cell.

Ignitor Variable Value

A floating point value for the progress variable.

Shapes

Sets the volume to which the ignitor applies using one or more volume shapes. Create shapes under **Tools** > **Volume Shapes**.

Temperature

Sets the temperature that the ignitor applies to cells within the specified shape when the ignitor is active.

Spark Ignition Model Reference

The Spark Ignition model solves a time-accurate model for subgrid spark expansion until the flame can be resolved on the mesh.

When the Spark Ingition model is activated, a new model to specify the laminar flame speed appears.

For the flame speed models (FGM with CFM or TFC), this model is called Subgrid Spark Ignition, and for the kinetic models (FGM Kinetic rate or Complex Chemistry), this model is called Subgrid Spark Flame Speed Ignition.

Combustion Model		Subgrid Spark Flame Speed Ignition	Subgrid Spark Ignition
Complex Chemistry		✔	
Flamelet Generated Manifold (FGM)	Coherent Flame Model (CFM)		✔
	FGM Kinetic Rate	✔	
	Turbulent Flame Speed Closure (TFC)		✔

Spark Ignition Model Reference

Theory	See Ignitors.	
Provided By	**[physics continuum]** > **Models** > *Spark Ignition Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Spark Ignition**	
Requires	*Material*: **Multi-Component Gas** *Time*: any unsteady *Reaction Regime*: **Reacting** Then either of:	
	Reacting Flow Models: **Flamelet** *Flamelet Models*: **Flamelet Generated Manifold (FGM)**	*Reacting Flow Models*: **Reacting Species Transport** *Reacting Species Models*: **Complex Chemistry**

Subgrid Spark Flame Speed Ignition Model

Subgrid Spark Flame Speed Ignition Model Reference

Provided By	**[physics continuum]** > **Models** > *Spark Ignition Models*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Subgrid Spark Flame Speed Ignition**
Requires	*Optional Models*: **Spark Ignition**
Properties	Laminar Flame Speed Option.

Properties

Laminar Flame Speed Option

Allows you to specify the model for the freely propagating premixed laminar flame speed:

- **Gulder**—this approach provides the Gulder Laminar Flame Speed sub-node which allows you to select the type of fuel. See, Gulder.
- **Metghalchi**. See Metghalchi.
- **User Defined Profile**

Subgrid Spark Ignition Model

Subgrid Spark Ignition Model Reference

Provided By	**[physics continuum]** > **Models** > *Optional Models*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Spark Ignition**
Requires	*Optional Models*: **Spark Ignition**

Subgrid Spark Ignitor

When the Spark Ignition model is selected as a physics model, you can create a Subgrid Spark Ignitor using the right-click option of the **[Continuum]** > **Ignitors** node.

Properties

Spark Start Time

Time from which the spark is activated.

Spark Position

Coordinates of the flame kernel (spark).

Coordinate System

The coordinate system which is used to define the coordinates of the flame kernel.

Turbulent Flame Speed Coefficient A

Coefficient for the turbulent flame speed, A in Eqn. (3458).

Maximum Duration Time

Maximum time for which the spark remains activated.

Curvature Correction

When activated, allows STAR-CCM+ to make corrections to the flame speed due to the curvature of the kernel.

Reaction Mechanism Formats

Reaction data, thermodynamic properties, and transport properties are imported in the standard format.

- All species in gas phase and/or surface data files must have unique names.

- Lines in an input file that start with an exclamation mark (!) are regarded as comments.

Gas Chemistry Reaction Files

The gas phase data files state the chemical elements and species that are included in the mechanism. The elements and species are followed by a list of possible reactions that the species can participate in, and the corresponding variables, A, n (β in Eqn. (3301)), and E_a, that are used by the Arrhenius law Eqn. (3301) to calculate the rate constant (k in Eqn. (3294)) for each reaction.

```
ELEMENTS
  C  H  O  N AR
END
SPECIES
H                O                OH               HO2              H2O2
CH               HCO              CH2-P            CH2-S            CH2O
CH3              CH3O             CH2OH            CH3O2            CH4
CH3OH            CH3O2H           C2H              C2O              HCCO
C2H2             CH2CO            C2H3             CH2CHO           CH3CO
C                CO               CO2              H2               H2O
O2               AR               N2
END
REACTIONS
!-------------------------------------------------------------------------------
!Reaction                                          A            n         Ea
!-------------------------------------------------------------------------------
H+O2=O+OH                                          2.000E+14    0.00   1.679E+04
O+H2=H+OH                                          5.100E+04    2.67   6.282E+03
OH+H2=H+H2O                                        1.000E+08    1.60   3.296E+03
OH+OH=O+H2O                                        1.500E+09    1.14   1.003E+02
H+HO2=OH+OH                                        1.500E+14    0.00   1.003E+03
H+HO2=H2+O2                                        2.500E+13    0.00   6.927E+02
H+HO2=O+H2O                                        3.000E+13    0.00   1.720E+03
O+HO2=OH+O2                                        1.800E+13    0.00  -4.060E+02
...                                                ...          ...    ...
END
```

When third-body reactions are specified in a mechanism, the reaction is written with an additional species M as a reactant, a product, or both. Auxiliary information regarding enhanced third-body efficiencies of certain species is given on the line below the reaction. The names of the third-body species are followed by their enhancement factors.

```
O+CO+M<=>CO2+M              1.800E+10    0.00   8600.00
H2/2.00/ O2/6.00/ H2O/6.00/ CO/1.50/ CO2/3.50/ AR/0.50/
```

Gas / Surface Thermodynamic Properties Files

Thermodynamic properties are given in state function data files for either seven-coefficient or nine-coefficient NASA polynomial form for each species:

Seven-coefficient NASA polynomial format

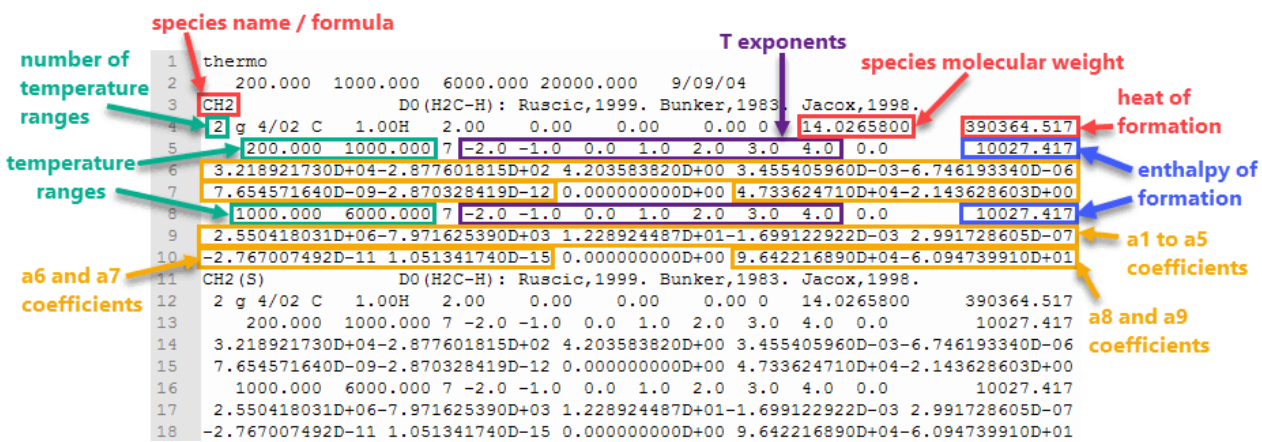
- Line 1: The name of the species, the elemental composition of the species, three temperatures which define the temperature ranges over which the polynomial fits to thermodynamic data are valid. For

example, the three temperatures (K) in the example below show the low temperature fit (298.000), the high temperature fit (5000.000), and the common temperature between these fits (1000.00).

- Line 2-4: Fourteen polynomial coefficients are provided (seven coefficients, a_1 through a_7, for two consecutive temperature ranges).

```
H                         H   1    0    0    0    298.000  5000.000 1000.00      1
 0.25000000D+01 0.00000000D+00 0.00000000D+00 0.00000000D+00 0.00000000D+00     2
 0.25471620D+05-0.46011760D+00 0.25000000D+01 0.00000000D+00 0.00000000D+00     3
 0.00000000D+00 0.00000000D+00 0.25471620D+05-0.46011760D+00                    4
O                         O   1    0    0    0    298.000  5000.000 1000.00      1
 0.25420590D+01-0.27550610D-04-0.31028030D-08 0.45510670D-11-0.43680510D-15     2
 0.29230800D+05 0.49203080D+01 0.29464280D+01-0.16381665D-02 0.24210310D-05     3
-0.16028431D-08 0.38906960D-12 0.29147640D+05 0.29639950D+01                    4
OH                        H   1O   1    0    0    298.000  5000.000 1000.00      1
 0.28827300D+01 0.10139743D-02-0.22768770D-06 0.21746830D-10-0.51263050D-15     2
 0.38868880D+04 0.55957120D+01 0.36372660D+01 0.18509100D-03-0.16761646D-05     3
 0.23872020D-08-0.84314420D-12 0.36067810D+04 0.13588605D+01                    4
HO2                       H   1O   2    0    0    298.000  5000.000 1000.00      1
 0.40721910D+01 0.21312960D-02-0.53081450D-06 0.61122690D-10-0.28411640D-14     2
-0.15797270D+03 0.34760290D+01 0.29799630D+01 0.49966970D-02-0.37909970D-05     3
 0.23541920D-08-0.80890240D-12 0.17622730D+03 0.92227240D+01                    4
```

Nine-coefficient NASA polynomial format



For more information, see Using Thermodynamic Polynomial Data.

Gas Transport Properties Files

Gas phase transport properties are given (optionally) by a molecular data file which provides seven variables for each species in the mechanism:

- chemical name
- indicator for the structure of the species (0 = atom, 1 = linear molecule, 2 = non-linear molecule)
- two parameters describing the shape of the Lennard-Jones potential well:
  - well depth divided by Boltzmann's constant [K]

- collision diameter [Å]—the (finite) distance at which the inter-particle potential is zero
- the bond dipole moment [Debye]
- the polarizability [Å3]
- the rotational collision number (ZROT) at 298 K

Since the molecular data is species-specific, you can use any file which contains data for all species in the gas or solid state phase. Molecular data files are not tied to a specific kinetic scheme.

```
H                    0 0.145D+03 0.205D+01 0.000D+00 0.000D+00 0.000D+00
O                    0 0.800D+02 0.275D+01 0.000D+00 0.000D+00 0.000D+00
OH                   1 0.800D+02 0.275D+01 0.000D+00 0.000D+00 0.000D+00
HO2                  2 0.107D+03 0.346D+01 0.000D+00 0.000D+00 0.100D+01
H2O2                 2 0.107D+03 0.346D+01 0.000D+00 0.000D+00 0.380D+01
CH                   1 0.800D+02 0.275D+01 0.000D+00 0.000D+00 0.000D+00
HCO                  2 0.498D+03 0.359D+01 0.000D+00 0.000D+00 0.000D+00
...                  . ...       ...       ...       ...       ...
```

Surface Chemistry Reaction Files

Surface data files contain relevant solid state data and look much the same as gas phase data files. In addition it contains data for the material of the surface, the surface sites, and the site density.

```
MATERIAL Pt1A1
SITE/PLATINUM    SDEN/3.28E-008/
  PT(S) O(S) NO2(S) NO(S)
END
REACTIONS
!-----------------------------------------------------------------------------
!Reaction                        A          n        Ea
!-----------------------------------------------------------------------------
O2+2PT(S)=2O(S)              4.900E+18    0.00   3.040E+01
   Rev                       3.100E+22    0.00   2.094E+02
NO+PT(S)=>NO(S)              6.300E+12    0.00  -2.350E+01
NO(S)=>NO+PT(S)              1.060E+16    0.00   1.155E+02
NO2+PT(S)=>NO2(S)            6.710E+12    0.00   0.210E+00
NO(S)+O(S)=NO2(S)+PT(S)     3.300E+20    0.00   1.273E+02
...                          ...           ...    ...
END
```

Keywords

All Input Files

The following keywords are accepted in all types of input files.

BULK

Precedes lines with surface bulk species in the mechanism. You can specify this keyword immediately before a custom name for the bulk data.

ELEMENTS / ELEMENT / ELEM

Precedes lines with elements of the species in the mechanism.

END

Required on the line below the data which follows each of the above keywords.

MATERIAL

Precedes the name of the material that the subsequent surface data corresponds to. You can state more than one material, where each material is followed by material-specific surface data.

REACTIONS / REAC

Precedes lines with reaction data for all reactions in the mechanism.

- To set the unit of E_a in the mechanism, follow this keyword with one of:
  - CAL/MOLE
  - KCAL/MOLE
  - JOULES/MOLE
  - KELVINS
  - EVOLTS
- To set the unit of A in the mechanism, follow this keyword with MOLES.

Units for E_a and A can be set on the same line. If no unit is specified, data is assumed to be given in cal/mole.

SDEN

Standard-state density for the site in units of moles/cm².

SITE

Precedes lines with surface species in the mechanism. SITE must be followed by the SDEN keyword.

You can specify this keyword immediately before a custom name for the site data.

SPECIES / SPEC

Precedes lines with species in the mechanism.

THERMO ALL

Same as THERMO but shows that all state functions data of the mechanism is found after the keyword and not in additional sections of the mechanism.

THERMO / THERM

Precedes lines with state function data in the mechanism.

Gas Input Files

The following keywords are only accepted in gas phase data input files.

DUPLICATE / DUP

If a reaction path can have multiple sets of reaction coefficients, follow each set by a line with this keyword.

FORD

In order to alter the forward rate of a specific reaction, you use this keyword on the line below the specific reaction, followed by a species name and reaction order. Specifying a reaction order in this way replaces existing reaction orders or adds new species and reaction orders to the reaction rate calculation.

HIGH

Placed on the line below a pressure-dependent, chemically activated bimolecular reaction, followed by reaction coefficients for the high-pressure limit: A_∞, β_∞, and E_∞ in Eqn. (3309).

LOW

Placed on the line below a pressure dependent fall-off reaction, followed by reaction coefficients for the low-pressure limit: A_0, β_0, and E_0 in Eqn. (3308).

PLOG

Given after a reaction, this keyword is followed (within forward slashes) by its corresponding pressure ranges, in units of bar, with pre-exponential factors, temperature exponents, and activation energies in Eqn. (3315).

REV

For one-way reactions, you can specify this keyword on the line below a reaction followed by reaction data for the reverse reaction.

RORD

In order to alter the reverse (backwards) rate of a specific reaction, you use this keyword on the line below the specific reaction, followed by a species name and reaction order. Specifying a reaction order in this way replaces existing reaction orders or adds new species and reaction orders to the reaction rate calculation.

SRI

For SRI pressure dependent reactions, this keyword is placed on a separate line after the keywords LOW and HIGH. This keyword is followed by five parameters: a, b, c, d, and e in Eqn. (3312).

TROE

For Troe pressure dependent reactions, this keyword is placed on a separate line after the keywords LOW and HIGH. This keyword is followed by four parameters: α, $T*$, $T**$, and $T***$ in Eqn. (3311).

Surface Input Files

The following keywords are only accepted in surface data input files.

COV

This keyword followed by a species and three coverage parameters, $\eta_{i,j}$, $\mu_{i,j}$, and $\varepsilon_{i,j}$, modifies the forward reaction rate constant according to Eqn. (3330).

DUPLICATE / DUP

If a reaction path can have multiple sets of reaction coefficients, precede each set by a line with this keyword.

FORD

In order to alter the forward rate of a specific reaction, you use this keyword on the line below the specific reaction, followed by a species name and reaction order. Specifying a reaction order in this way replaces existing reaction orders or adds new species and reaction orders to the reaction rate calculation.

REV

For one-way reactions, you can specify this keyword on the line below a reaction followed by reaction data for the reverse reaction.

RORD

In order to alter the reverse (backwards) rate of a specific reaction, you use this keyword on the line below the specific reaction, followed by a species name and reaction order. Specifying a reaction order in this way replaces existing reaction orders or adds new species and reaction orders to the reaction rate calculation.

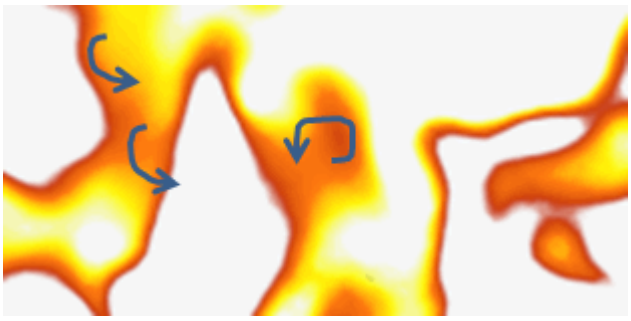
STICK

To use sticking coefficients instead of reaction coefficients, provide this keyword without parameters on the line below a reaction. Replace the reaction variables (A, β, and E_a in Eqn. (3301)) of the preceding reaction by sticking coefficients: a_i, b_i and c_i in Eqn. (3331).

Reacting Species Transport

When using Reacting Species Transport models, STAR-CCM+ solves transport equations for the mass fractions of all species that are involved in the chemical reactions. These species mass fractions are passed to a separate chemistry solver that returns the amount of product.

In the following diagram, the dark area shows a thick reaction front (where the reactions occur). Since some reactions happen slower than turbulent mixing, the turbulent mixing causes some slowly-reacting intermediate species to escape this thick reaction front and continue mixing further downstream.



The Reacting Species Transport models include:

- Complex Chemistry: for simulating reacting flow using detailed mechanisms within a turbulent flame in which kinetic phenomena are significant.
- Eddy Break-Up: for simulating reacting flow with a simple (one or two step) mechanism, where kinetic phenomena are not significant.
- Eddy Contact Micromixing
- Thickened Flame Model
- Polymerization

Use the Reacting Species Transport workflow for simulating reacting flow within a multi-component liquid or within a gas-phase turbulent flame.

Complex Chemistry

The Complex Chemistry model employs a stiff solver to integrate the chemistry over a time-step, and uses techniques, such as Clustering, Dynamic Mechanism Reduction, and In-Situ Adaptive Tabulation (ISAT), to mitigate this expense. Since the Complex Chemistry model uses a detailed mechanism, you can model kinetic phenomena—such as flame ignition, slowly forming pollutants, flame quenching, and extinction. Complex chemistry mechanisms are usually imported in Chemkin format.

This model offers the most general approach in modeling finite-rate kinetics. Use this model if other models do not capture the finite-rate effects that drive the process.

The Complex Chemistry model has two submodels for simulating turbulence-chemistry interactions—the Laminar Flame Concept (LFC), and the Eddy Dissipation Concept (EDC).

- Laminar Flame Concept (LFC)

  The LFC model evaluates the instantaneous reaction rate at the mean temperature, pressure, and species mass fraction—which, for steady-state, corresponds to integrating the chemistry with respect to the residence time in a cell. The LFC model is appropriate for premixed, partially premixed, and unsteady flames.
- Eddy Dissipation Concept (EDC)

  The EDC model integrates chemistry for a time-scale close to the Kolmogorov (smallest eddy) time-scale in a cell. The EDC model is appropriate for steady-state diffusion flames, which gives a lower reaction rate than with the LFC model.

These models account for turbulent-chemistry interaction through the enhanced diffusivity of the turbulence model. You can adjust the flame position by changing the turbulent Schmidt number.

Eddy-Break-Up (EBU) Model

The Eddy-Break-Up model assumes that reactions are limited by the rate that turbulence can mix the reactants and heat into the flame zone. The EBU model limits kinetic reaction rates by the large-scale turbulent mixing rate. Since kinetic rates are replaced by a single turbulent mixing rate, use only small mechanisms with one or two steps. The EBU model is a flame position model and is not intended for modeling kinetically dominated phenomena such as ignition and pollutants.

If kinetic data exist, you can choose between the combined time scale model and hybrid model. These models account for finite-rate kinetic effects—required for premixed flames where turbulence would otherwise ignite the mixture before the flame holder. However, for diffusion flames that are purely turbulence-controlled, the standard EBU model is adequate and efficient.

Eddy Contact Micromixing Model

The Eddy Contact Micromixing Model is a combustion model for liquids. In liquids, where species diffusivities are typically small and molecular mixing time-scales are long, the Eddy Contact Micromixing model accounts for diminished molecular diffusivity in the turbulent mixing rate by reducing the reaction rate. The Eddy Contact Micromixing model is a variant of the EBU model.

Thickened Flame Model (TFM)

The Thickened Flame Model is a Large Eddy Simulation (LES) model where the flame is artificially thickened, by increasing the local diffusivity, so that the flame front can be resolved on the mesh.

Polymerization Model

The polymerization model is a dedicated model for simulating liquid polymerization processes. It tracks the polymer size distribution for live and dead polymers.

Polymerization is the process of creating long-chain molecules by chemically combining large numbers of small monomer molecules. This process starts by mixing monomer and initiator molecules in a solvent. The process of polymerization is comprised of several chemical reactions, such as initiation, propagation, chain transfer, chain branching, scission, and termination, that involve radicals of different chain lengths. The polymerization model in STAR-CCM+ simulates free-radical polymerization and uses Method of Moments to describe the size distribution of the polymers.

Reacting Species Transport Model Ignitors

When the Complex Chemistry, EBU, or TFM models are selected, a **Continuum** > **Ignitors** node appears which allows you to control ignition.

The Complex Chemistry, EBU, or TFM models allow you to create a Fixed Temperature Ignitor. When active, a Fixed Temperature Ignitor applies a constant temperature for the cells that are contained within the ignitor volume shapes.

In addition to the Fixed Temperature Ignitor, the EBU model also provides the EBU Ignitor. When the EBU Ignitor is on, for the cells which are contained within the EBU ignitor volume shapes, the ignitor forces reaction rates to be mixing limited without using products.

The type of ignitor that is available for the EBU model depends upon the setting that you select for the EBU Reaction Control property.

EBU Reaction Control Property Setting	EBU Ignitor available?	Fixed Temperature Ignitor available?
Standard EBU	Yes	No
Hybrid	Yes	Yes
Combined Time-Scale	Yes	Yes
Kinetics Only	No	Yes

For more details, see the specific combustion model reference page.

Contents:

Chemistry Acceleration
Reacting Species Transport Workflow
Polymerization Workflow
Reacting Species Transport Reference

Chemistry Acceleration

You can use Clustering, ISAT, or Dynamic Mechanism Reduction methods to speed up Complex Chemistry calculations.

The following are chemistry acceleration options:

- Clustering

- In Situ Adaptive Tabulation (ISAT)
- Dynamic Mechanism Reduction

The Equilibrium Time Scale model is not an acceleration option, however, you can use it to provide a quick approximate solution for steady-state detailed chemistry computations. See Equilibrium Time Scale.

Clustering

The clustering method groups cells with similar thermal and chemical states before integrating the averaged state for the group. The reaction mapping is then interpolated back to the cells assuming the net reaction rate of all cells in the cluster is the same as the cluster average.

Clustering usually provides a substantial speed-up as the number of clusters is less than the number of cells in the simulation. Since the number of clusters increases much slower than the number of cells, clustering performance improves with mesh size.

The clustering algorithm uses a uniform N-dimensional grid, where N is the number of clustering variables. The cells in the simulation are then binned into each clustering grid. By default, STAR-CCM+ uses two clustering variables for unsteady simulations, namely equivalence ratio and temperature. For steady-state simulations, the log of the chemical time scale is used as an additional clustering variable.

STAR-CCM+ provides other user selectable clustering variables, including species mass fractions, mixture fraction, pressure, enthalpy, and entropy.

The default clustering variables of equivalence ratio and temperature represent the underlying trajectory of mixing followed by reaction. These default clustering variables are suitable for most hydrocarbon combustors. However, there are cases where careful selection of additional clustering variables can improve accuracy. For example, for slowly forming NO formation behind a premixed flame front, where the equivalence ratio and temperature are nearly uniform, you can add the NO species mass fraction to the clustering variables list. For supersonic combustion with shock-waves, you can include the pressure variable since the pressure changes substantially through the domain.

You can view the clusters that STAR-CCM+ generates by visualizing the Clustering Index. Each cluster has its own unique index, with the maximum value representing the total number of clusters. A Clustering Index of zero represents cells that are excluded from the reaction, such as cells with temperatures that are below the Minimum Temperature that is required for reactions. The Clustering Equivalence Ratio and Clustering Mixture Fraction are also available for post-processing.

Clustering is available when using either the DARS-CFD or the CVODE ODE (Ordinary Differential Equation) solver.

In Situ Adaptive Tabulation (ISAT)

ISAT is available when using the Complex Chemistry model with either the CVODE ODE solver or the DARS-CFD ODE solver.

In Situ Adaptive Tabulation (ISAT) is a tabulation method in which computationally expensive functions are tabulated and then interpolated at run time. In this case, the stiff chemistry ODEs are tabulated. ISAT can substantially speed up complex chemistry simulations for steady state problems (including unsteady LES of

steady combustion) in which the tabulated chemistry mappings are regularly retrieved and mechanisms have less than 40 species. Dr. Stephen B. Pope developed the ISAT method [662].

When using ISAT with the CVODE solver, the ISAT tables are created at a constant pressure and time-step. If the pressure or time-step in the domain varies (as occurs in steady simulations), the ISAT solver bins the pressure and time-step values into appropriate ranges automatically, creating one table per bin.

The ISAT method approximates the solution of the ODE for a given initial condition through:

- The thermodynamic variable (sensible enthalpy)
- The composition variable (specific molar number)
- The absolute pressure and time-step (depending on the type of simulation)

Note: The ISAT method is not available when both the Implicit Unsteady and Eddy Dissipation Concept models are activated.

You can use ISAT to retrieve approximate values for multi-dimensional functions that are expensive to compute. In reacting simulations, you can use ISAT to store and retrieve approximate reaction mapping solutions from a system of coupled ODEs. During the chemistry step, the ISAT solver computes the reaction mapping in a particular cell. ISAT searches the table for an existing point with which to approximate the reaction mapping within the specified error tolerance. If ISAT finds such a point, then the new reaction mapping is not calculated with the ODE solver, but rather interpolated based on the stored reaction mapping in the table. Otherwise, if ISAT cannot find a suitable point in the table, the ODE solver is called, and a reaction mapping for this point is calculated. This newly calculated point is stored in the table. The size of the table therefore grows as the simulation progresses. See Complex Chemistry.

Dynamic Mechanism Reduction

When using the Complex Chemistry model, you can choose to allow the CVODE solver in STAR-CCM+ to solve for a reduced number of species that are taken from the full chemical mechanism that is imported. The mechanism is reduced dynamically—in every cell at every time-step or iteration. Solving for fewer species reduces computational time, however, accuracy is also reduced. Dynamic Mechanism Reduction is based on the Directed Relation Graph (DRG) algorithm [652] where species that do not change substantially over the reaction step are eliminated from the mechanism. See Dynamic Mechanism Reduction.

You provide the inputs for the DRG algorithm—a tolerance and a small set of target species, which are retained in the mechanism.

Reacting Species Transport Workflow

Use this workflow for simulating multi-component liquid reactions or gas-phase turbulent flames which involve kinetic phenomena such as flame ignition, slowly forming pollutants, flame quenching, and extinction.

The steps in this workflow are intended to follow on from the initial steps in the Reacting Flow General Workflow.

Note: If you are modeling Reacting Channels, follow the Reacting Channels Workflow—even when using a reacting species transport model. See Reacting Channels Workflow.

1. For the physics continuum that represents the reacting flow, select the following models—in addition to the models that are previously selected, with **Auto-Select recommended models** activated:

Group Box	Model
Reacting Flow Models	**Reacting Species Transport**
Reacting Species Models	• For multi-component gas turbulent flames with a large mechanism: **Complex Chemistry** • For multi-component liquid-liquid reactions limited by finite-rate kinetics: **Complex Chemistry** • For multi-component liquid or gas reactions with a simple mechanism: **Eddy Break-Up** • For multi-component liquid or gas reactions which are slow: **Eddy Break-Up** with the Kinetics Only property activated. • For multi-component liquid reactions with a simple mechanism in which the chemical reactions are fast (molecular mixing time-scales are long): **Eddy Contact Micromixing**
Turbulence Chemistry Interactions (for Complex Chemistry)	• For steady-state diffusion flames: **Eddy Dissipation Concept** • For premixed, partially-premixed, and unsteady flames, or slow reactions: **Laminar Flame Concept**

2. Select any Optional models.

   For example:

   - NOx Emission or Soot Emissions—to model the formation of these pollutants.
   - Fluid Film—to model intraphase reactions such as in trickle bed reactors, or selective catalytic reduction (SCR) reactions in urea film. See Modeling Fluid Film.
   - The **Radiation** model is useful for modeling applications in which radiative heat transfer is important, as is the case for most combustion systems. The soot emission models influence the **Participating Media Radiation (DOM)** and **Gray Thermal Radiation** models by contributing to the absorption coefficient of the continuous phase (the absorption coefficient describing both absorption and emission).
   - VOF—Useful for modeling multi-phase reacting flow applications where two phases co-exist with a distinct interface, such as glass furnaces.
   - Gravity—Use when gravity forces significantly influence the solution, for example in fire simulations.
3. If using Emissions models or Surface Chemistry, follow the specific workflow before continuing with the following steps.
4. Define the parameters of the Reacting Species physics model:
   - For the Complex Chemistry model, define the Properties of the Complex Chemistry model and its sub-nodes.
     1. Select the **Models** > **Complex Chemistry** node and set the required Properties.

        The ODE Solver is set to CVODE by default—which is the recommended choice for most simulations. The CVODE solver calculates reaction rates and solves complex chemistry problems in STAR-CCM+. If required, you can set the ODE Solver to DARS-CFD. The DARS-

CFD solver translates the chemical mechanism to FORTRAN code, which can either be interpreted (ASCII) or compiled (Binary).

2. Select the **Complex Chemistry** > **Chemistry Acceleration** node and, if required, activate the necessary acceleration options:
   - *Clustering* (activated by default)
   - *Dynamic Mechanism Reduction*
   - *ISAT* (not compatible with Dynamic Mechanism Reduction and not advised with Clustering)

   Then define the acceleration option Properties, see Chemistry Acceleration Properties.
3. When using the DARS-CFD ODE Solver, you can relax the species to their equilibrium compositions over a set time-scale. To set an equilibrium time scale, select the **Complex Chemistry** > **Approximation Options** node and activate *Equilibrium Time Scale*. Select the **Approximation Options** > **Equilibrium Time Scale** node and set the Properties.

For more information, see Complex Chemistry Model Reference.

- For the EBU or ECM model, define the EBU or ECM Properties.
  - The EBU property, *Reaction Control*, is set to **Hybrid** by default—which is appropriate for most simulations. Hybrid specifies the reaction rate as the minimum of the rate predicted by turbulent mixing and the rate predicted by finite-rate chemical kinetics. However, if the rate is determined solely by the turbulent mixing scale, select **Standard EBU** instead.

  See Eddy Break-Up Model Reference or Eddy Contact Micromixing Model Reference.

5. Import or create the mechanism which represents the reacting flow reactions.
   - For Complex Chemistry, import the complex chemistry definition using the **Models** > **Complex Chemistry** node. It is possible to create the mechanism using the **Models** > **Reacting** > **Reactions** node—however, it is impractical, as most Complex Chemistry definitions are too large. You can find chemical mechanisms that are developed by academia on university websites worldwide.
   - For EBU or ECM, create or import reactions using the **Models** > **Reacting** > **Reactions** node.

   See Importing Species and Reactions.
6. Set the parameters of any other **[Continuum]** > **Models**, as required.

   When using the Eddy Dissipation Concept model or the Laminar Flame Concept model, you can adjust the flame position by changing the **Multi-Component Gas** > **Material Properties** > **Turbulent Schmidt Number**.
7. Return to the Reacting Flow General Workflow.

Polymerization Workflow

Polymerization processes involve various chemical reactions during which monomer molecules combine to form polymer chains. These processes require the reacting species transport physics model in STAR-CCM+.

The steps in this workflow are intended to follow on from the initial steps in the Reacting Flow General Workflow.

1. For the physics continuum that represents the reacting flow, select the following models—in addition to the models that are previously selected, with **Auto-select recommended models** activated:

Group Box	Model
Enabled Models	**Reacting Species Transport** (selected automatically)
Reacting Species Models	**Polymerization**

2. Select any Optional models that are also required.

The process of polymerization is comprised of several sub-reactions which include free radical polymer reactions. Free radicals are small, charged molecules that react with monomers and create charged monomers. The charged monomers then combine with other monomers and form a polymer chain. Throughout the polymerization process, free radical polymer reactions consume and create free radicals at different rates. In STAR-CCM+, you can specify which free radical polymer reactions occur and define the rates.

3. Set the free-radical polymer reaction properties.
   a) Expand the **[Continuum]** > **Models** > **Polymerization** > **Free-Radical Polymer Reactions** node.
   b) To define the reaction rate for a free-radical polymer reaction, select the required free-radical polymer reaction sub-node and set the **Method** that you require as a scalar profile.
   c) Define the *Unit System* property appropriate to the reaction.
   d) If you do not wish to specify the reaction rate for any of the free-radical polymer reactions, select the relevant sub-node and:
      - Set the **Method** property for the sub-node to **Constant**.
      - Set *Value* to 0.0.

The **Polymer Scalars** and **Polymer Moments** sub-nodes allow you to specify User Rate and User Rate Jacobian source terms directly to the Polymer Scalar and Polymer Moment equations.

4. If you intend to specify User Rate or User Rate Jacobian source terms:
   a) Select the **Polymerization** node and activate the property, *User Rates*.

      The **Polymer Scalars** and **Polymer Moments** sub-nodes become populated.
   b) Set the **User Rate** and / or **User Rate Jacobian** values for the **Polymer Scalars** and **Polymer Moments** as required.
5. Set the remaining Polymerization model Properties as required.

   See: Polymerization Model Reference.
6. Specify the Liquid Components and Material Properties of the Multi-Component Liquid.
   a) Right-click the **[Continuum]** > **Models** > **Multi-Component Liquid** > **Liquid Components** node and select one of the following options:
      - Import Species.

        In the dialog that appears, **Browse** to and select the file which contains the monomer species that you are importing.
      - Select Mixture Components.

        Browse the Standard Materials Database that appears and select the liquid components that represent the monomers. Click **Apply** then **Close**.

      For more information, see Managing Mixture Components.

b) Expand the **Multi-Component Liquid** > **Liquid Components** node and define the material properties as required.

For more information, see Setting Material Properties Methods.

7. Set the parameters of any other **[Continuum]** > **Models**, as required.
8. For the continuum, define any necessary parameters for the **Reference Values** and **Initial Conditons**.

   Set the Initial Condition concentrations for the **Polymer Initiator**, **Polymer Monomer**, **Polymer Radical**, **Polymer Solvent**, and**Polymer Modifier** scalars.
9. Define any necessary Physics Conditions and Values for the Region and Boundaries.

   For the Inlet boundaries, expand the **Physics Values** node and set concentrations for the **Polymer Initiator**, **Polymer Monomer**, **Polymer Radical**, **Polymer Solvent**, and**Polymer Modifier** using scalar array profiles.
10. If necessary, set up Injectors. For example, if any of the polymerization reaction components are injected into the domain.

    See: Working With Injectors.
11. Return to the Reacting Flow General Workflow.

Reacting Species Transport Reference

This section provides reference material, including properties, sub-nodes, field functions, and other model-specific settings, for all of the reacting species transport models.

Contents:

Complex Chemistry Model Reference
Approximation Options Properties
Chemistry Acceleration Properties
CVODE Properties
DARS-CFD Properties
Turbulence-Chemistry Interactions Model Reference
Eddy Break-Up Model Reference
Restoring the Premixed Eddy Break Up Model
Eddy Contact Micromixing Model Reference
Thickened Flame Model Reference
Polymerizaton Model Reference

Complex Chemistry Model Reference

The Complex Chemistry model requires gas-phase or liquid-phase chemical mechanisms which typically contain tens or hundreds of species, with hundreds or thousands of reactions. Since different species evolve at different rates, the kinetics are termed stiff, and integrating the reaction step is computationally expensive.

Theory	See Complex Chemistry
Provided By	**[physics continuum]** > **Models** > *[Combustion Models]*

Example Node Path	**Continua** > **Physics 1** > **Models** > **Complex Chemistry**	
Requires	*Material*: **Multi-Component Gas** or **Multi-Component Liquid** *Reaction Regime*: **Reacting** *Reactive Flow*: **Reacting Species Transport**	
Properties	Key properties are: [ODE Solver]. See Complex Chemistry properties.	
Specific Right-Click Actions	See Right-Click Actions.	
Activates	Model Controls (child nodes)	CVODE or DARS-CFD, Approximation Options, Chemistry Acceleration
	Region Settings	Active Reactions, CFL Factor Option
	Report Options	Complex Chemistry Quantity Maximum, Complex Chemistry Quantity Minimum, Complex Chemistry Quantity Sum. See Reports.
	Other Continuum Nodes	The **Ignitors** node provides the right-click option to create a **Fixed Temperature Ignitor**. See Ignitors.
	Field Functions	Chemistry Heat Release Rate, Chemistry Time Step, Clustering Equivalence Ratio, Clustering Index, Clustering Mixture Fraction, Number of DMR Retained Species. See Field Functions.

Complex Chemistry Model Properties and Expert Properties

Source Enabled Trigger

Allows you to deactivate chemistry at the beginning of the calculation—to let the flow field converge before starting to calculate the chemistry. You can choose when the chemistry is calculated, according to the following options:

- **Never**: The Source Enabled Trigger is deactivated, so chemical source terms are calculated from the first iteration/timestep.
- **Iteration**: The Source Enabled Trigger is activated, so the chemical source terms are activated and are calculated after the iteration that is specified as value of the *Begin* property. This option is available in steady-state cases.
- **Timestep**: The Source Enabled Trigger is activated, so the chemical source terms are activated and are calculated after the timestep that is specified as value of the *Begin* property. This option is available in unsteady-state cases.

Begin

Specifies the iteration or time-step after which the reactions are activated. During previous iterations or time-steps, reactions are deactivated. This feature is useful when modeling significant flow as you can deactivate reactions until a flow solution is found.

ODE Solver

- **DARS-CFD**: Reaction rates and their numerical derivative (Jacobian) are evaluated in DARS and integrated with the DARS ODE solver. You can evaluate the reaction rate routines either in ASCII (un-compiled code), or binary (Fortran compiled code).
- **CVODE**: Reaction rates and their analytical derivative (Jacobian) are evaluated in STAR-CCM+, and integrated with the Sundials CVODE solver [651].

Minimum Temperature For Reactions

Below this temperature, chemical reactions are deactivated.

Dynamic Load Balancing

When activated, dynamic load-balancing is turned on for complex chemistry calculations.

Right-Click Actions

Import Chemistry Definition (Chemkin format)

Activates a standard *Open* dialog that allows you to import a chemical mechanism file, thermodynamic data file, and optionally a transport properties file.

Delete Chemistry Definition

Removes the reactions, and optionally the species, that are used for the chemistry definition.

Region Settings

Applies to all regions:

Active Reactions Option

Allows you to activate or deactivate chemical reactions in this region.

Applies to fluid regions:

CFL Factor Option

Method	Corresponding Physics Value Nodes
Activated For steady-state simulations, an artificial time-step that is approximately equal to the cell residence time, is used to advance chemical reactions. This option allows you to modify the size of that chemistry time-step using the CFL Scaling Factor.	**CFL Scaling Factor** The chemistry time-step is equal to the local time-step multiplied by the CFL Scaling Factor, where the local time-step is based on the inverse of accumulated diffusion and convection fluxes of the species, as shown in Eqn. (3339).
Deactivated	None

Complex Chemistry Reports

These Complex Chemistry reports take a particular value and perform a global operation (maximum, minimum or sum) over all of the processors. For a serial run, all three types of reports report the same value. Several of the Complex Chemistry Variable options only produce reports with values when ISAT is used. If ISAT is not available, these reports have a zero value.

Complex Chemistry Quantity Maximum

Complex Chemistry Quantity Minimum

Complex Chemistry Quantity Sum

Complex Chemistry Reports Properties

Each of the reports have the following Properties available:

Units

The units that are used for specifying the quantity.

Complex Chemistry Variable

Allows you to choose the type of Complex Chemistry Variable to plot.

- **None**
- **CPU Time**: The time that the CPU takes with complex chemistry calculations.
- **Elapsed Time**: The time that has elapsed during complex chemistry calculations.
- **Adds**: The number of points that have been added to the table. (ISAT only).
- **Grows**: The number of points in the table that have increased their region of accuracy. (ISAT only).
- **Retrieves**: The number of points that are retrieved from the ISAT table; these retrieves avoid expensive direct ODE solver calculations (ISAT only).
- **Direct Evaluations**: The number of points that are calculated by calling the ODE solver. (ISAT only).
- **Table Size**: The ISAT table size.
- **Queries**: The total number of queries given to ISAT.
- **Direct ODE Queries**: The number of queries that result in direct ODE solver calculations. Low values in this report reflect a good setup of ISAT parameters.

Variable Accumulation Type

Format of displayed value for those Complex Chemistry Variables that report a value of time.

- **Cumulative**:
  The time taken for the complex chemistry solver calculations in all iterations.
- **Per Iteration** : The time taken for the complex chemistry solver calculation for each iteration.

Field Functions

Chemistry Heat Release Rate

The chemistry heat release shows the amount of heat that is released due to complex chemistry reactions.

Chemistry Time Step

For steady state simulations, the chemistry time step is the total time that is spent integrating the chemistry in the cell.

Clustering Equivalence Ratio

The clustering equivalence ratio is defined as the fuel to oxidizer ratio divided by the stoichiometric fuel to oxidizer ratio.

Clustering Index

Clustering index is an integer value showing the cluster that the cell belongs to. Values of zero represent cells which are not chemically active, such as cells with a temperature below the Minimum Temperature for Reactions.

Clustering Mixture Fraction

The clustering mixture fraction is defined as the atomic mass fraction that originates from the fuel stream. Since the fuel is assumed to be hydrocarbons, the clustering mixture fraction is the mass fraction of C and H in the cell.

Number of DMR Retained Species

Available when the chemistry acceleration property, *Dynamic Mechanism Reduction*, is activated. Displays the actual number of species from the full chemical mechanism that the CVODE solver retains and solves for.

Approximation Options Properties

The Approximations Options node only appears when the Complex Chemistry ODE Solver is set to DARS-CFD.

Approximation Options Properties

Equilibrium Time Scale

When this option is activated, the Chemistry Acceleration node becomes unavailable. An **Approximation Options** > **Equilibrium Time Scale** node appears in which you can specify the fuel components.

Equilibrium Time Scale

The Equilibrium Time Scale model relaxes the species field to its chemical equilibrium compositions over a set turbulent time-scale. Since no chemical kinetics are calculated, the model is an approximation that is usually used as an initial condition for steady-state finite-rate chemistry simulations. This model cannot predict ignition or slowly forming pollutants such as NOx.

Use the Equilibrium Time Scale option as an initial condition for steady-state finite-rate chemistry simulations. This method is designed to provide a quick approximate solution for steady-state detailed chemistry computations.

The chemistry time scale is computed based on the specified fuel species. See Complex Chemistry.

Equilibrium Time Scale Properties

ISAT Absolute Tolerance

Specifies the absolute tolerance for the ISAT approximation. The solver makes sure that the absolute value of the interpolation error is less than the value that is specified.

ISAT Table Size (MB)

Maximum Size for the ISAT Table.

Flow Time Constant

Time constant for the flow τ_{flow} that is used to calculate τ_{char} in Eqn. (3341).

Chemistry Time Constant

Time constant for the chemical reaction τ_{chem} that is used in Eqn. (3341).

Include Chemistry Time?

When activated, the chemistry time scale is included in the computation.

Equilibrium Time Scale Expert Properties

Restart ISAT Table

When activated, the simulation resumes from the ISAT table that is given in the work folder. The ISAT file name is `isat_10.tab`. When deactivated, the simulation is started from a blank table.

Save on Master Only

When activated, this option saves ISAT tables only on the master in a parallel server.

Equilibrium Time Scale Context Menu

Save Equilibrium ISAT Table

Saves the table in the work folder under the `isat_1.tab` file name.

Delete Equilibrium ISAT Table

Deletes the internal ISAT table and makes ISAT start from a blank table.

Fuel Components Properties

Fluid Stream Components

Provides a selection dialog from which you choose the fluid stream components.

Fuel Components Context Menu

Add Fuel Component

Provides a selection dialog from which you choose the fuel components.

[Fuel Component] Properties

Mass Fraction

Allows you to specify the mass fraction of the fuel component between 0.0 and 1.0. All fuel components that are in a fluid stream, must sum to 1.0.

Chemistry Acceleration Properties

The **Complex Chemistry** > **Chemistry Accelaration** node allows you to speed up complex chemistry calculations using In-Situ Adaptive Tabulation (ISAT), Clustering, or Dynamic Mechanism Reduction.

Chemistry Acceleration

When the In Situ Adaptive Tabulation and the Clustering properties are activated at the same time, accuracy can be reduced. Therefore, if In Situ Adaptive Tabulation is activated, deactivate Clustering. ISAT is not available when using Dynamic Mechanism Reduction.

Chemistry Acceleration Properties

In Situ Adaptive Tabulation

When this option is activated, a **Chemistry Accelaration** > **In Situ Adaptive Tabulation** node appears in which you can specify the ISAT tables.

Clustering

When this option is activated, a **Chemistry Accelaration** > **Clustering** node appears in which you can specify the clustering components.

Dynamic Mechanism Reduction

When this option is activated, a **Chemistry Accelaration** > **Dynamic Mechanism Reduction** node appears in which you can specify the parameters for reducing the number of species in the mechanism that are solved for by the CVODE solver.

In Situ Adaptive Tabulation

In Situ Adaptive Tabulation Properties

Restart

Only available when the Complex Chemistry ODE Solver is set to DARS-CFD. When activated, the simulation resumes from the ISAT table that is given in the work folder. The ISAT file name is `isat_1.tab`. When deactivated, the simulation is started from a blank table.

Absolute Tolerance

Absolute tolerance for the ISAT approximation.

Table Size (MB)

Maximum size for the ISAT table.

Pressure Specification

Only available when the Complex Chemistry ODE Solver is set to DARS-CFD. Allows you to select if pressure remains constant or variable.

Time Step Specification

Only available when the Complex Chemistry ODE Solver is set to DARS-CFD. In unsteady simulations, this property is Constant and in most steady simulations, variation of the time-step is not suitable for ISAT tabulation. Therefore, STAR-CCM+ uses a special Iterative tables technique to accommodate this variation.

In Situ Adaptive Tabulation Expert Properties

Save on Master Only

Only available when the Complex Chemistry ODE Solver is set to DARS-CFD. When activated, this option saves ISAT tables only on the master in a parallel server.

Sensitivity Absolute Tolerance

Only available when the Complex Chemistry ODE Solver is set to CVODE. Allows you to specify the absolute tolerance of the CVODE solver which computes the sensitivity matrix that is used in the ISAT approximation.

Sensitivity Relative Tolerance

Only available when the Complex Chemistry ODE Solver is set to CVODE. Allows you to specify the relative tolerance of the CVODE solver which computes the sensitivity matrix that is used in the ISAT approximation.

In Situ Adaptive Tabulation Context Menu

Save ISAT Table

Only available when the Complex Chemistry ODE Solver is set to DARS-CFD. Saves the table in the work folder under the `isat_1.tab` file name.

Delete ISAT Table

Only available when the Complex Chemistry ODE Solver is set to DARS-CFD. Deletes the internal ISAT table and makes ISAT start from a blank table.

ISAT Tables Context Menu

Only available when the Complex Chemistry ODE Solver is set to DARS-CFD.

New ISAT Table

Creates an ISAT Table for which you can specify the limits between which the time-steps are resolved for this table. You can add more ISAT tables, but be careful that the time-step domains of the multiple tables are consecutive and do not overlap. There are two special cases when the time-step is outside of the range of all the tables:

- The time-step is less than the lowest value of Time Lower Limit among the ISAT table nodes. In this case, the table with the smallest value for this property is called once.
- The time-step is greater than the highest value of Time Upper Limit. In this case, ISAT is not called and the chemistry step is calculated directly through DARS-CFD.

Clustering

Components

The default clustering components for steady-state simulations are:

- Chemistry Time Step
- Equivalence Ratio
- Temperature

For unsteady simulations, the chemistry time-step is the flow time-step—which is constant and therefore unavailable for clustering.

The following clustering components are also available:

- Chemistry Time Step
- Entropy
- Equivalence Ratio
- Mixture Fraction
- Pressure
- Progress Enthalpy
- Static Enthalpy
- Temperature

[Clustering Component] Properties

Tolerance

Allows you to specify the tolerance of the component for clustering.

Tolerance Type

There are two ways to specify the clustering table resolution for any clustering variable:

- **Relative**: The clustering bin size is calculated as the maximum of that variable in the computational domain minus the minimum, which is then multiplied by the Relative Tolerance, ε_{rel}.
- **Absolute**: The clustering bin size is the Absolute Tolerance, ε_{abs}.

Dynamic Mechanism Reduction

Dynamic Mechanism Reduction Properties

Error Tolerance

The lower the number, the closer the solution is to that of the full chemical mechanism—less species are removed from the mechanism. Higher values exclude a greater number of species from the solution, which is therefore less accurate.

Imposed Target Species

Species that you select from the full chemical mechanism that are target species for the Directed Relation Graph (DRG) algorithm Eqn. (3352). These species are always included in the reduced chemical mechanism.

Minimum Number of Target Species

The minimum number of target species (including the imposed target species). If this value is greater than the number of *Imposed Target Species*, the remaining target species are taken to be those with the fastest reaction rates.

CVODE Properties

When the Complex Chemistry model ODE Solver property is set to CVODE, the **Complex Chemistry** > **CVODE** node appears.

Absolute Tolerance

Specifies the absolute error tolerance for each ODE step in the CVODE ODE solver. The solver makes sure that the error is less than the relative tolerance multiplied by the quantity that is being integrated, plus the absolute tolerance.

Relative Tolerance

Specifies the relative error tolerance for each ODE step in the CVODE ODE solver. The solver makes sure that the error is less than the relative tolerance multiplied by the quantity that is being integrated, plus the absolute tolerance.

Error Codes

If you receive an error code when using the CVODE solver and building a combustion table, you can find the meaning of the error code value in Appendix B.2, *CVODE main solver module*, here: https://computation.llnl.gov/sites/default/files/public/cv_guide.pdf

DARS-CFD Properties

When the Complex Chemistry model ODE Solver property is set to DARS-CFD, the **Complex Chemistry** > **DARS-CFD** node appears.

DARS-CFD Properties

Binary Problem Description

When activated, a binary format library file is used.

Case Path

The name and path of the folder containing the complex chemistry description (either ASCII text files or a FORTRAN library).

Is Library Loaded

Indicates with a read-only checkbox whether the DARS-CFD library is loaded into the simulation. The library maps to the type of complex chemistry model that you select. It is loaded either when it is imported or when the solution is initialized.

Absolute Tolerance

Specifies the absolute tolerance for each ODE step in the DARS-CFD ODE solver. The solver makes sure that the error is less than the relative tolerance multiplied by the quantity that is being integrated, plus the absolute tolerance.

Relative Tolerance

Specifies the relative tolerance for each ODE step in the DARS-CFD ODE solver. The solver makes sure that the error is less than the relative tolerance multiplied by the quantity that is being integrated, plus the absolute tolerance.

Use Reference Values for Temperature

When activated, the low and high temperature limits for the DARS-CFD ODE solver are taken from the Minimum and Maximum Allowable Temperature of the Continuum. When deactivated, a separate, DARS-CFD Temperature Range node appears that allows you to set the low and high temperature limit to the DARS-CFD ODE solver.

DARS-CFD Context Menu

Import Complex Chemistry Definition (DARS-CFD Format)

Activates a standard *Open* dialog that imports a library file for the complex chemistry definition in DARS-CFD format.

Import Case From STAR-CCM+ Repository

Allows you to select from a list of numerical and analytical jacobian libraries that are stored in STAR-CCM+.

Reload DARS-CFD Case

Allows you to select a previously defined DARS-CFD case.

Save Complex Chemistry Definition Input

Allows you to save the currently defined DARS-CFD case which you can then reload in future simulations.

Temperature Range Properties

The Temperature Range node appears under the DARS-CFD node when the DARS-CFD property, Use Reference Values for Temperature, is deactivated.

Minimum Temperature

Low temperature limit for the DARS-CFD ODE solver. The default value, 250K, is the Minimum Allowable Temperature.

Maximum Temperature

High temperature limit for the DARS-CFD ODE solver. The default value 4000.0K is the Maximum Allowable Temperature.

Binary Run Options Properties

The Binary Run Options node appears as a sub-node to the DARS-CFD node.

User Coding

These user coding options are all available to select when the DARS-CFD property, Binary Problem Description, is activated.

- **None**: Does not use user coding.
- **Calculate Source Term**: Calculates the species source terms with the DARS-CFD FORTRAN file SetDARSGasSrcTerms.f
- **Modify Reaction Rates**: Modifies the species reaction rates with the DARS-CFD FORTRAN file ModDARSGasReacRates.f
- **Modify Source Term**: Modifies the species source terms with the DARS-CFD FORTRAN file ModDARSGasSrcTerms.f

Binary Run Options Expert Properties

Run-time Compile

When activated, the relevant files are loaded into the simulation and compiled during runtime. When deactivated, the pre-compiled library is loaded.

Turbulence-Chemistry Interactions Model Reference

Turbulence-chemistry interactions require additional modeling, as well as RANS or LES modeling, to allow chemical reactions in thin reaction zones to be solved.

Turbulence-Chemistry Interactions Reference

Model Names and Abbreviations	**Eddy Dissipation Concept**	**EDC**
	Laminar Flame Concept	**LFC**
Theory	See Complex Chemistry.	
Provided By	**[physics continuum]** > **Models** > *Turbulence Chemistry Interactions*	

Example Node Path	**Continua** > **Physics 1** > **Models** > **Eddy Dissipation Concept**	
Requires	*Material*: **Multi-Component Gas** or **Multi-Component Liquid** *Reaction Regime*: **Reacting** *Reactive Flow*: **Reacting Species Transport** *[Partially-Premixed] Combustion models*: **Complex Chemistry**	
Properties	none	
Activates	Model Controls (child nodes)	EDC only: • EDC Fine Structure Time Factor • EDC Fine Structure Length Factor When modeling Steady simulations, the **Eddy Dissipation Concept** > **Acceleration Factor** or **Laminar Flame Concept** > **Acceleration Factor** node appears. See Acceleration Factor Properties.
	Field Functions	EDC Fine Structure Length Fraction, EDC Fine Structure Time, EDC Mean Reaction Rate Ratio. (There are no LFC-related field-functions). See Field Functions.

EDC Fine Structure Time Factor

Allows you to specify the fine structure (flamelet) time factor, C_t in Eqn. (1773) as a scalar profile.

EDC Fine Structure Length Factor

Allows you to specify the fine structure (flamelet) length factor, C_l in Eqn. (3338) as a scalar profile.

Acceleration Factor Properties

Acceleration Factor

Setting a value of 1.0 indicates that the species source term is fully explicit which results in unstable, but fast, convergence. When a value of 0.0 is set, the species source term is taken as fully implicit which results in stable, but slow, convergence. In practice, you set a value between 0.0 and 1.0 which provides the best convergence when using default Under Relaxation Factor values.

Turbulence-Chemistry Interactions Field Functions

Extra field functions are made available to the simulation when the Eddy Dissipation Concept (EDC) model is used:

EDC Fine Structure Length Fraction

$C_l\left(\frac{\upsilon T}{L^2}\right)^{0.25}$ in Eqn. (3338).

EDC Fine Structure Time

The fine structure (flamelet) region residence time, τ in Eqn. (3339).

EDC Mean Reaction Rate Ratio

The ratio f that is calculated by Eqn. (3338) which is used to modify the chemical reaction source term in Eqn. (3337).

Eddy Break-Up Model Reference

Eddy Break-Up (EBU) combustion models solve individual transport equations for mean species concentrations on the computational grid.

The reaction rates that are used in the transport equations are calculated as functions of the mean species concentrations, turbulence characteristics and, depending on the specific model that is used, temperature. A mean enthalpy equation is solved in addition to the species transport equations. The mean temperature and density are then calculated knowing the mean enthalpy and species concentrations.

Four variants are available for this model, selectable from the *Reaction Control* property in the *Properties* window.

Eddy Break-Up Combustion Models

Theory	See Eddy Break-Up	
Provided By	**[physics continuum]** > **Models** > *[Non-Premixed] Combustion Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Eddy Break-Up**	
Requires	*Material*: either **Multi-Component Gas** or **Multi-Component Liquid** *Reaction Regime*: **Reacting** *Reactive Flow*: **Reacting Species Transport**	
Properties	Key properties are: **Reaction Control**, **Store Reaction Rates**, **Begin**, **Source Enabled Trigger**, and **Source Term Limiting**. See EBU properties.	
Specific Right-Click Actions	See Right-Click Actions.	
Activates	Region Settings	Active Regions Option. See Region Settings.
	Monitors	**FuelMassFr**: Fuel Mass Fraction.
	Other Continuum Nodes	The **Ignitors** node provides the right-click option to create a **Fixed Temperature Ignitor**. See Ignitors.
	Field Functions	Residence Time. See EBU Field Functions

EBU Model Properties

Reaction Control

Selects the variant of EBU combustion model. Apart from the Kinetics Only variant, these models are not compatible with the Spalart-Allmaras and Spalart-Allmaras-DES turbulence models.

- **Standard EBU**
  Selects Standard EBU—assumes that the reaction rate is dictated solely by the turbulent mixing time scale.
- **Hybrid**
  Selects Hybrid—reaction rate is the minimum of the rate predicted by turbulent mixing and the rate predicted by finite-rate chemical kinetics.
- **Combined Time Scale**
  Selects Combined Time Scale—reaction rate is given by the sum of the rates predicted by turbulent mixing and by chemical kinetics.
- **Kinetics Only**
  Selects Kinetics Only—assumes that the reaction rate is dictated by finite-rate chemical kinetics. This option is very similar to the Complex Chemistry option, however, without the stiff solver for solving several stiff reactions simultaneously. Therefore, it is suggested that you use the Complex Chemistry model instead of EBU with Kinetics Only.

Store Reaction Rates

Activates the capability to post-process reaction rates, forward and backward (where applicable).

- **Activated**
  Stores reaction rates during the simulation run and makes them available among scalar field functions.

  After the simulation run, each reaction rate is available in the scalar field function list under the item Reaction Rate Forward or Reaction Rate Backward.

  If you set the *Reaction Control* property of the EBU model to Standard EBU, you can only visualize forward reaction rates.
- **Deactivated**
  Does not store reaction rates (default).

Begin

Specifies the iteration or time-step after which the reactions are activated. During previous iterations or time-steps, reactions are deactivated. This feature is useful when modeling significant flow as you can deactivate reactions until a flow solution is found.

Source Enabled Trigger

Allows you to deactivate chemistry at the beginning of the calculation—to let the flow field converge before starting to calculate the chemistry. You can choose when the chemistry is calculated, according to the following options:

- **Never**: The Source Enabled Trigger is deactivated, so chemical source terms are calculated from the first iteration/timestep.
- **Iteration**: The Source Enabled Trigger is activated, so the chemical source terms are activated and are calculated after the iteration that is specified as value of the *Begin* property. This option is available in steady-state cases.

- **Timestep**: The Source Enabled Trigger is activated, so the chemical source terms are activated and are calculated after the timestep that is specified as value of the *Begin* property. This option is available in unsteady-state cases.

Source Term Limiting

For the EBU combustion model (particularly with the **Kinetics Only** option), the specified reaction rate can be large enough to drive the reactant concentration below zero. To keep all reactant concentrations in a physically correct range, activate this limit on the final reaction rate that is applied to the species transport equation.

For most applications, leave this option deactivated. Use it only when a reasonable convergence cannot be obtained and the species concentrations clearly show unphysical values.

- **Activated**
  Limits the final reaction rate that is applied to the species transport equation Eqn. (3361). Provides the field function, Residence Time (chemistry time step).
- **Deactivated**
  Does not limit the final reaction rate (default).

Right-Click Actions

Import Chemistry Definition (Chemkin format)

Activates a standard Open dialog that allows you to import a chemical mechanism file, thermodynamic data file, and optionally a transport properties file.

Delete Chemistry Definition

Removes the reactions, and optionally the species, that are used for the chemistry definition.

Region Settings

Applies any region:

Active Reactions Option

Allows you to activate or deactivate chemical reactions in this region.

EBU Field Functions

Residence Time

For steady state simulations, represents the time duration, t_r in Eqn. (3361), over which the chemistry is calculated (transient/unsteady simulations use the time step as the duration for calculating the chemistry). This field function is available when the property, *Source Term Limiting*, is activated.

Restoring the Premixed Eddy Break Up Model

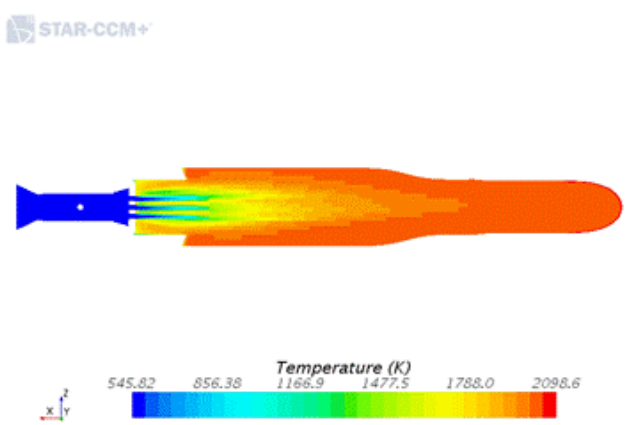
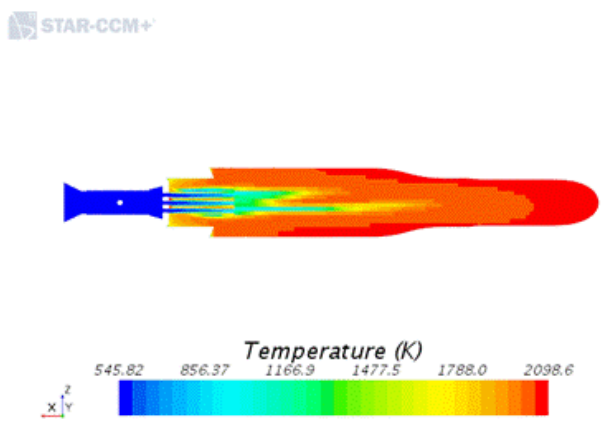
The Premixed Eddy Break-Up model in STAR-CCM+ v.11.04 behaves as a Partially Premixed version. You can restore the Premixed model settings as described in this section.

As part of a plan to improve and simplify the procedure for selecting combustion models, the Premixed version of the Eddy Break Up model is discontinued in STAR-CCM+ v.11.04.

The new Eddy Break Up Model is an improvement on the deprecated Premixed Eddy Break Up Model—the new Eddy Break Up model:

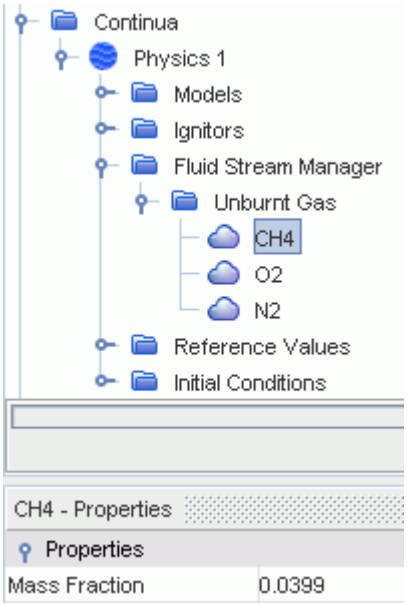
- Supports multiple reactions as well as hybrid and kinetics-based schemes. Therefore, you can perform calculations more accurately and over wider operation ranges.
- Accounts for oxidizer and product concentrations when calculating the source term.
- Allows you to switch between fully premixed and partially premixed calculations.

When loading a simulation in STAR-CCM+ v.11.04 or later that was saved in a version of STAR-CCM+ before v11.04, the Premixed Eddy Break Up model behaves as a Partially-Premixed Eddy Break Up model. Several settings are defined differently and must be set up again. The following images show a comparison of the temperatures on the cross section of a flame holder.

PEBU 11.02	PEBU 11.04 (Partially Pre-Mixed)
	

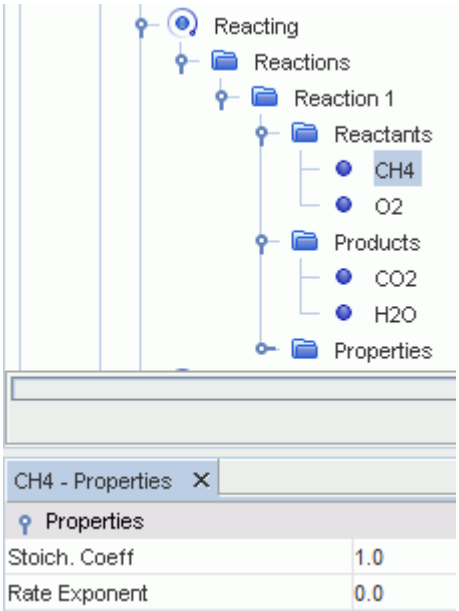
STAR-CCM+ v.11.04 provides the EBU model as an option for Premixed Combustion. However, this model is the same as that provided for Partially-Premixed combustion. Further steps are required to restore the premixed behaviour that was available in STAR-CCM+ v.11.02 and earlier. These steps are as follows:

In STAR-CCM+ v11.02, the reactants and corresponding mass fractions are defined under the **[Continuum]** > **Fluid Stream Manager** node.



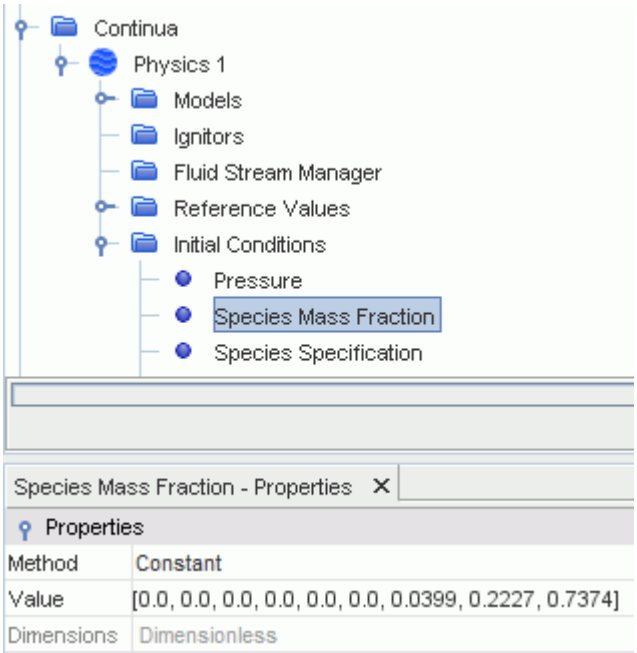
1. In STAR-CCM+ v11.02, you use the Fluid Stream Manager to set the species composition of a fuel and an oxidizer. In STAR-CCM+ v11.04, you define the reaction framework as follows:
   a) Define the reactants and products for the reaction under the **Models** > **Reacting** > **Reactions** node. Make sure that you create one reaction between a fuel and an oxidizer and specify the products of the complete combustion.

      See Defining Chemical Reactions Manually.
   b) Specify stoichiometric coefficients for each component to represent a balanced complete combustion reaction.

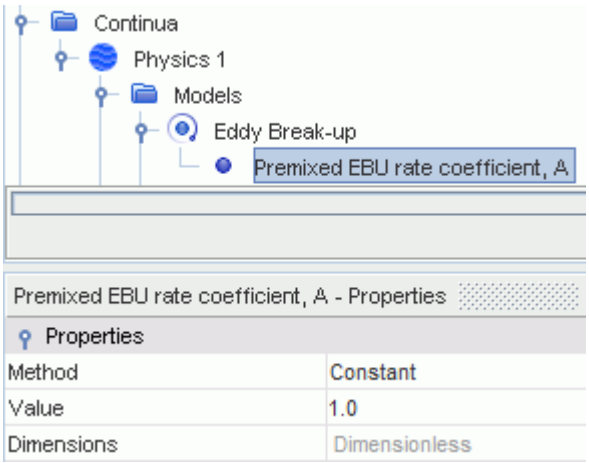


   c) To specify the mass fractions of each component, select the **[Continuum]** > **Initial Conditions** > **Species Mass Fraction** node.

d) For the *Value* property, specify the mass fractions of each species in the composition of the initial mixture.



In STAR-CCM+ v11.02, the Eddy Break-Up Coefficients are defined under the **Models** > **Eddy Break-Up** > **Premixed EBU rate coefficient, A** node.



In STAR-CCM+ v11.04 or later, the Eddy Break-Up Coefficients are defined in the general reactions framework.

2. Select the **Models** > **Eddy Break-Up** node and set *Reaction Control* to **Standard EBU**.
3. Define the EBU rate coefficient A to suit STAR-CCM+ v11.04 or later—this value is different from the value that is set for the Premixed EBU rate coefficient A in STAR-CCM+ v11.02.
   a) Expand the **Models** > **Reacting** > **Reactions** > **[Reaction]** > **Properties** node.
   b) Select the **Properties** > **EBU Coefficients** > **EBU Rate Coefficient, A** node and specify the *Value*.

4. To restore the Ignitor:
   a) Right-click the **[Continuum]** > **Ignitors** node and select **New** > **EBU Ignitor**.
   b) Select the **EBUIgnitor 1** node and specify the properties as they are set in STAR-CCM+ v11.02.
   c) If the **EBUIgnitor 1** node *Activator* is set to **Pulse**, select the **Pulse** sub-node and specify the necessary properties.
5. For all inlet and outlet boundaries within each physics continuum, select the **[Boundary]** > **Physics Values** > **Species Mass Fraction** node and specify the mass fractions of each species with the same values as are specified in the initial conditions.

Eddy Contact Micromixing Model Reference

This physics model allows you to model turbulent molecular-scale mixing and reaction of multi-component liquids, dominated by viscous-convective and viscous-diffusive processes.

The Eddy Contact Micromixing model [654] is numerically efficient, reasonably accurate, and easy to use for multiple reactions. It can be used for both steady and unsteady cases. This model is only available with the multicomponent liquid model, and requires turbulence modeling. It works with all turbulent models that are available for the EBU hybrid model.

Eddy Contact Micromixing Model Reference

Theory	See Eddy Contact Micromixing.	
Provided By	**[physics continuum]** > **Models** > *Reacting Species Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Eddy Contact Micromixing**	
Requires	*Material*: **Multi-Component Liquid** *Reaction Regime*: **Reacting** *Reacting Flow Models*: **Reacting Species Transport**	
Properties	Key properties are: Source Enabled Trigger, Begin, and Store Reaction Rates. See Eddy Contact Micromixing Properties.	
Activates	Model Controls (child nodes)	When you use the Eddy Contact Micromixing model, the **Models** > **Reacting** > **Reactions** > **[Reaction]** > **Properties** > **Micromixing Timescale** node is available among the objects for defining chemical reactions. See Micromixing Timescale Properties.
	Materials	Specify the liquid materials under the Multi-Component Liquid model.

Eddy Contact Micromixing Properties

Source Enabled Trigger

Allows you to deactivate chemistry at the beginning of the calculation—to let the flow field converge before starting to calculate the chemistry. You can choose when the chemistry is calculated, according to the following options:

- **Never**: The Source Enabled Trigger is deactivated, so chemical source terms are calculated from the first iteration/timestep.
- **Iteration**: The Source Enabled Trigger is activated, so the chemical source terms are activated and are calculated after the iteration that is specified as value of the *Begin* property. This option is available in steady-state cases.
- **Timestep**: The Source Enabled Trigger is activated, so the chemical source terms are activated and are calculated after the timestep that is specified as value of the *Begin* property. This option is available in unsteady-state cases.

Begin

Specifies the iteration or time-step after which the reactions are activated. During previous iterations or time-steps, reactions are deactivated. This feature is useful when modeling significant flow as you can deactivate reactions until a flow solution is found.

Store Reaction Rates

Provides the capability to post-process reaction rates, forward and backward (where applicable).

- **Activated** Stores reaction rates during the simulation run and makes them available among scalar functions.
- **Deactivated** Does not store reaction rates (default).

Micromixing Timescale Properties

Select method

Allows you to select the method for calculating the micromixing timescale.

- **Classical Scalar Dissipation** Selects the Classical Scalar Mixing method (considers the high-Schmidt number effect) [655].
- **Kolmogorov** Selects the Kolmogorov Method (corresponds to Engulfment type of mixing). See Source Term Definition.
- **No Micromixing (Kinetics Only)** Selects the assumption that the reaction rate is dictated solely by finite-rate chemical kinetics.
- **User Micromixing** Selects the User Defined method, allowing you to work with a user-defined field function.

Thickened Flame Model Reference

For transient large-eddy simulations (LES) of turbulent reactive flows, the Thickened Flame Model (TFM) provides a method for modeling flames on meshes that are too coarse to resolve the flame structure. This model artificially thickens flames in three-dimensional premixed or partially-premixed gas combustion scenarios.

The TFM is particularly useful for simulating applications such as explosions and gas-turbine combustors at part load.

When using imported reaction data with the TFM, the Global Chemistry Kinetics model is activated—which helps to control the import.

When more than one fluid stream is present, this model uses a solver to calculate mixture fraction, when applicable. Some of the calculations can be post-processed as field functions.

Thickened Flame Model Reference

Theory	See Thickened Flame.	
Provided By	**[physics continuum]** > **Models** > *Reacting Species Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Thickened Flame Model**	
Requires	*Material*: **Multi-Component Gas** *Reaction Regime*: **Reacting** *Reacting Flow Models*: **Reacting Species Transport** *Space*: **Three Dimensional** *Time*: any Unsteady	
Properties	None. Model parameters are controlled using child nodes.	
Activates	Physics Models	Model Groups: *Kinetics for Thickened Flame Model* The properties of the **Global Chemistry Kinetics** model control the timing of imported reaction data in the simulation. See Global Chemistry Kinetics Model Properties.
	Model Controls (child nodes)	• Efficiency Factor • Flame Thickening Factor • Laminar Flame Properties • Fluid Stream Manager
	Solvers	Thickened Flame Solver—appears when more than one fluid stream is defined. See Thickened Flame Solver Properties.
	Other Continuum Nodes	The **Ignitors** Node provides the right-click option to create a **Fixed Temperature Ignitor**. See Ignitors
	Field Functions	Efficiency Factor and Flame Thickening Factor (see Field Functions)

Global Chemistry Kinetics Model Properties

Source Enabled Trigger

Allows you to deactivate chemistry at the beginning of the calculation—to let the flow field converge before starting to calculate the chemistry. You can choose when the chemistry is calculated, according to the following options:

- **Never**: The Source Enabled Trigger is deactivated, so chemical source terms are calculated from the first iteration/timestep.

- **Iteration**: The Source Enabled Trigger is activated, so the chemical source terms are activated and are calculated after the iteration that is specified as value of the *Begin* property. This option is available in steady-state cases.
- **Timestep**: The Source Enabled Trigger is activated, so the chemical source terms are activated and are calculated after the timestep that is specified as value of the *Begin* property. This option is available in unsteady-state cases.

Begin

Specifies the iteration or time-step after which the reactions are activated. During previous iterations or time-steps, reactions are deactivated. This feature is useful when modeling significant flow as you can deactivate reactions until a flow solution is found.

Store Reaction Rates

When activated, stores reaction rates during the simulation run and makes them available among scalar field functions.

Source Term Limiting

When activated, limits the final reaction rate that is applied to the species transport equation Eqn. (3361).

For the Thickened Flame model, the specified reaction rate can be large enough to drive the reactant concentration below zero. To keep all reactant concentrations in a physically correct range, activate this limit on the final reaction rate that is applied to the species transport equation.

For most applications, leave this option deactivated. Use it only when you cannot obtain reasonable convergence and unphysical values appear for the species concentrations.

An example of unphysical values would be large variations of the N2 mass fraction in a premixed flame. The mass fraction of N2 is expected to be uniform in the domain, since N2 does not participate in the chemical reaction of a premixed flame.

Model Controls

Efficiency Factor

Sets the method by which STAR-CCM+ computes the efficiency factor. This factor increases the flame speed in order to correct underestimation of the flame front wrinkling.

Method	Corresponding Method Node
Power Law	**Power Law** These model constants control the sharpness of the transitions between asymptotic behaviors: • *Constant alpha*, exponent α, from Eqn. (3375)] • *Constant b*, exponent b, from Eqn. (3376) • *Constant Ck*, coefficient C_k, from Eqn. (3377) through Eqn. (3379)
Wrinkling Factor Ratio	**Wrinkling Factor Ratio** Constant for calibrating β in the model computation (Eqn. (3372)):

	• *Constant cms*, coefficient c_{ms}

Flame Thickening Factor

STAR-CCM+ multiplies the species and heat diffusion coefficients by this factor, and also uses it to decrease reaction rates. Although you can set the factor as a constant, STAR-CCM+ provides a choice of dynamic forms.

Method	Corresponding Method Node
Arrhenius	**Arrhenius** Applies to one-step chemistry schemes. Model parameters are: • *Thickening Factor Maximum*, F_{max}, from Eqn. (3365) • *Constant Beta*, coefficient β, from Eqn. (3365), Eqn. (3371), and Eqn. (3372) • *Constant Gamma*, coefficient Γ , from Eqn. (3364), Eqn. (3371), Eqn. (3373), Eqn. (3375), and Eqn. (3376)
Progress Variable	**Progress Variable** In this approach the only model parameter that you set is the *Thickening Factor Maximum* value, F_{max} in Eqn. (3366).
User-Defined	**User-Defined** Sets the dynamic thickening factor F in Eqn. (3362) using its **Flame Thickening Factor Profile** sub-node, which functions as a scalar profile.

Laminar Flame Properties

Laminar Flame Speed

The **Laminar Flame Speed** node provides options for controlling flame speed in its properties.

Method	Corresponding Method Node
Gulder Laminar Flame Speed	Selects the Gulder method. Activates the **Gulder Laminar Flame Speed** node which allows you to select a fuel using the *Fuel Name* property.
Metghalchi Laminar Flame Speed	Selects the Metghalchi method. Activates the **Metghalchi Laminar Flame Speed** node.
User Defined Laminar Flame Speed	Selects the User-Defined method. Activates the **User Defined Laminar Flame Speed** node.

Laminar Flame Thickness

The **Laminar Flame Thickness** node provides options for controlling flame thickness in its properties.

Method	Corresponding Method Node
Laminar Flame Speed Relation	**Laminar Flame Speed Relation** Exposes the following terms: • *Laminar Prandtl Number*, Prandtl number P_r, from Eqn. (3381) • *Constant mu1*, coefficient μ_1, from Eqn. (3382) • *Constant mu2*, coefficient μ_2, from Eqn. (3382)
User Defined Laminar Flame Thickness	**User Defined Laminar Flame Thickness** Sets the laminar flame thickness using its **Laminar Flame Thickness Profile** sub-node. This sub-node functions in a manner similar to a scalar profile, but it also includes a table with laminar flame thickness as a function of equivalence ratio.

Thickened Flame Solver Properties

Under-Relaxation Factor

In order to promote convergence, this property is used to under-relax changes of the solution during the iterative process. If residuals show solution divergence and do not decrease, reduce the under-relaxation factor for the relevant solvers. The default value is 0.9.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Field Functions

Efficiency Factor

Accounts for the effect of eddies that are too small to ressolve in the bulk model. Expressed as E in Eqn. (3362)], Eqn. (3369), Eqn. (3370), Eqn. (3374) and Eqn. (3375).

Flame Thickening Factor

F in Eqn. (3362), Eqn. (3363), Eqn. (3365), Eqn. (3366), Eqn. (3369) and Eqn. (3370).

Polymerizaton Model Reference

Polymerization is the process of creating long-chain molecules by chemically combining large numbers of small monomer molecules. This process starts by mixing monomer and initiator molecules in a solvent.

The process of polymerization is comprised of several chemical reactions, such as initiation, propagation, chain transfer, chain branching, scission, and termination, that involve radicals of different chain lengths.

The Polymerization model in STAR-CCM+ simulates free-radical polymerization and uses Method of Moments for size distribution calculations.

Polymerizaton Model Reference

Theory	See Polymerization.		
Provided By	**[physics continuum]** > **Models** > *Reacting Species Models*		
Example Node Path	**Continua** > **Physics 1** > **Models** > **Polymerizaton**		
Requires	*Material*: **Multi-Component Liquid** *Reaction Regime*: **Reacting** *Reacting Flow Models*: **Reacting Species Transport**		
Properties	A key property of this model is *User Rates*. See Polymerizaton Properties.		
Activates	Model Controls (child nodes)	The Polymerization model contains three sets of sub-nodes for polymer scalars, polymer moments, and free-radical polymer reactions. See Model Controls.	
	Initial Conditions	Polymer Dead Moment Polymer Initiator Polymer Live Moment Polymer Modifier	Polymer Monomer Polymer Radical Polymer Solvent See Initial Conditions.
	Boundary Inputs	Polymer Dead Moment Polymer Initiator	Polymer Monomer Polymer Radical

		Polymer Live Moment Polymer Modifier	Polymer Solvent See Boundary Settings.
	Solvers	Polymerizaton Solver. See Polymerizaton Solver Properties.	
	Monitors	DeadPolymerMoment0 DeadPolymerMoment1 DeadPolymerMoment2 InitiatorConcentration LivePolymerMoment0	LivePolymerMoment1 LivePolymerMoment2 MonomerConcentration RadicalConcentration
	Field Functions	Initiator Concentration Modifier Concentration Monomer Concentration Polydispersity Index Polymer MassAvg ChainLength Polymer Moment Dead 0 Polymer Moment Dead 1	Polymer Moment Dead 2 Polymer Moment Live 0 Polymer Moment Live 1 Polymer Moment Live 2 Polymer NumberAvg ChainLength Radical Concentration Solvent Concentration See Field Functions.

Polymerizaton Properties

Secondary Gradients

There are two sources of secondary gradients in STAR-CCM+ flow solvers:

- boundary secondary gradients for diffusion
- interior secondary gradients at cell faces

Use this property to control which gradients are included in the solver. **On** gives both gradients while **Off** excludes them. **Interior Only** and **Boundaries Only** select the corresponding gradients.

Convection

Sets the discretization scheme that STAR-CCM+ uses for computing the convection flux on a cell face in appropriate transport equations. More information is given in the related topic for the Convection Term:

- **1st-Order**: First-order upwind scheme. This scheme scales the transported quantity by the upstream or downstream mass flowrate depending on flow direction. Only use when a higher-order scheme fails to give convergence, or in order to obtain an initial solution before switching to a higher-order scheme.
- **2nd-Order**: Second-order upwind scheme. This scheme introduces linear interpolation of cell values on either side of the upstream or downstream face. Using this scheme can lead to poorer convergence properties, but gives accuracy as good as or better than the first-order scheme.

User Rates

Allows you to add user sources to Polymer Scalars and Polymer Moment equations, in addition to the built-in sources. Activating this property populates the **Polymer Scalars** and **Polymer Moments** sub-nodes. See User Rates Sub-Nodes.

Model Controls

Free-Radical Polymer Reactions Properties

- *Initiator Efficiency, f*: Allows you to specify the efficiency of the initiator of the chain in Eqn. (3589), Eqn. (3591), and Eqn. (3593).
- *Chain Transfer to Modifier*: Activating this property creates an extra **Free-Radical Polymer Reactions** sub-node that represents a chain transfer to modifier reaction.
- *Chain Transfer to Solvent*: Activating this property creates an extra **Free-Radical Polymer Reactions** sub-node that represents a chain transfer to solvent reaction (Eqn. (3581)).
- *Chain Transfer to Polymer*: Activating this property creates an extra **Free-Radical Polymer Reactions** sub-node that represents a chain transfer to polymer reaction (Eqn. (3583)).
- *Beta Scission*: Activating this property creates an extra **Free-Radical Polymer Reactions** sub-node that represents a Beta Scission reaction (Eqn. (3584) and Eqn. (3585)).
- *Heat of Propagation*: Allows you to specify a value for the heat that is released due to propagation reactions.
- *Heat of Initiation*: Allows you to specify a value for the heat that is released due to initiation reactions.

Free-Radical Polymer Reaction Sub-Nodes

Some of these sub-nodes are available by default while others require activation in the properties of the **Free-Radical Polymer Reactions** node.

- **Initiator Decomposition (I -> 2R')**
- **Radical Chain Initiation (R' + M -> R1)**
- **Propagation (Rn + M -> Rn+1)**
- **Chain Transfer to Monomer (Rn + M -> Pn + R1)**
- **Disproportionation (Rn + Rm -> Pn + Pm)**
- **Combination (Rn + Rm -> Pn+m)**
- **Chain Transfer to Modifier (Rn + X -> Pn + R1)**
- **Chain Transfer to Polymer (Rn + Pm -> Pn + Rm)**
- **Beta Scission (Rn + Pm -> Pm + Rp + Pm-p)**
- **Terminal Double Bond (Rn + Pm -> Rn+m)**

See Reacting Flow Nomenclature.

Free-Radical Polymer Reaction Sub-Node Properties

- *Method*: Allows you to specify the **Method** as a scalar profile.
- *Value*: Specifies the value with a text entry. You can also enter an expression, such as `10 * sin(3.14 * $Time)`, directly. This property is only available when *Method* is set to **Constant**.
- *Scalar Function*: Activates an object selector dialog from which you can choose a field function. This property is only available when **Method** is set to **Field Function**.
- *Unit System*: Allows you to define **Units** that are appropriate to the reaction:
  - **kmol, m, s, K**
  - **mol, cm, s, K**

User Rates Sub-Nodes

When the **User Rates** property is activated, the following sub-nodes of the **Polymer Scalars** and **Polymer Moments** nodes allow you to specify User Rate and User Rate Jacobian source terms directly to the Polymer Scalar and Polymer Moment equations. Each of these sub-nodes functions as a scalar profile.

Polymer Scalars

Initiator, **Modifier**, **Monomer**, **Radical**, and **Solvent**

Polymer Moments

Dead Moment and **Live Moment**

Initial Conditions

Polymer Dead Moment, Polymer Live Moment

Respectively, sets concentrations for dead moment (Eqn. (3594), Eqn. (3596), and Eqn. (3598)) and live moment (Eqn. (3588), Eqn. (3590), and Eqn. (3592)). Scalar array profile values.

Polymer Initiator, Polymer Modifier, Polymer Monomer, Polymer Radical, Polymer Solvent

Respectively, sets concentrations for initiator (Eqn. (3576)), modifier, monomer (Eqn. (3579)), radical (Eqn. (3577)), and solvent (Eqn. (3582)). Scalar profile values.

Boundary Settings

Polymer Dead Moment, Polymer Live Moment

Respectively, sets concentrations for dead moment (Eqn. (3594), Eqn. (3596), and Eqn. (3598)) and live moment (Eqn. (3588), Eqn. (3590), and Eqn. (3592)). Scalar array profile values.

Polymer Initiator, Polymer Modifier, Polymer Monomer, Polymer Radical, Polymer Solvent

Respectively, sets concentrations for initiator (Eqn. (3576)), modifier, monomer (Eqn. (3579)), radical (Eqn. (3577)), and solvent (Eqn. (3582)). Scalar profile values.

Flow Boundaries

The following boundary types have all of the above-listed physics values available by default.

- Free Stream
- Mass Flow Inlet
- Pressure Outlet
- Stagnation Inlet
- Velocity Inlet

Wall Boundary

Wall Combustion Scalar Option

Wall Combustion Scalar	Corresponding Physics Value Nodes
Zero Flux	None

Specified Value	All of the above-listed physics values

Polymerizaton Solver Properties

Under-Relaxation Factor

In order to promote convergence, this property is used to under-relax changes of the solution during the iterative process. If residuals show solution divergence and do not decrease, reduce the under-relaxation factor for the relevant solvers. The default value is 0.9.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Field Functions

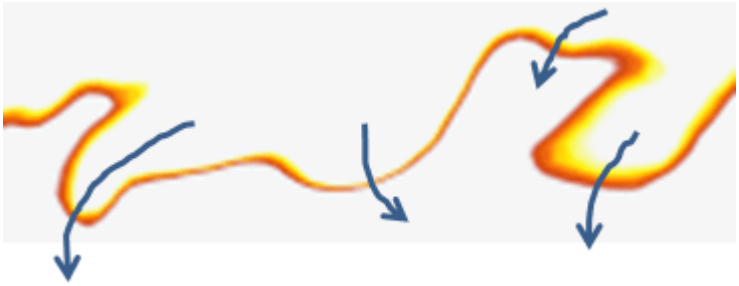
- **Initiator Concentration**
- **Modifier Concentration**
- **Monomer Concentration**
- **Polydispersity Index**
- **Polymer MassAvg ChainLength**
- **Polymer Moment Dead 0**
- **Polymer Moment Dead 1**
- **Polymer Moment Dead 2**
- **Polymer Moment Live 0**
- **Polymer Moment Live 1**
- **Polymer Moment Live 2**
- **Polymer NumberAvg ChainLength**
- **Radical Concentration**

- **Solvent Concentration**

Flamelet

Flamelet models avoid the computational expense of complex chemistry calculations by pre-calculating the chemistry in gas-phase laminar flames using simple 0D or 1D geometries with detailed chemical mechanisms. A reduced set of variables are used to parameterize and tabulate the species from these simple flames, which are then interpolated in the 3D simulation of turbulent flames in STAR-CCM+.

In the following diagram, the dark area shows a thin reaction front in which reactions happen faster than turbulent mixing. Intermediate species do not exist for long enough to be transported out of the reaction front by turbulent mixing.



All Flamelet models in STAR-CCM+ use mixture fraction as one parameter. The mixture fraction is defined as the atomic (or elemental) mass fraction that originated from the fuel stream. For combustors with one fuel and one oxidizer stream, if all species diffuse at the same rate—a good assumption for turbulent flows—you can define one unique mixture fraction. Since atomic elements are conserved in chemical reactions, the mixture fraction is conserved and therefore the corresponding transport equation does not have a source term.

STAR-CCM+ provides the following Flamelet models:

- Flamelet Generated Manifold (FGM): generally the most accurate flamelet model.
- Steady Laminar Flamelet (SLF): only better than FGM for some highly strained non-premixed flames.
- Chemical Equilibrium: least accurate flamelet model, but is straightforward to use when you do not have a mechanism, or when the combustor is large.

The Chemical Equilibrium and SLF models are best-suited for turbulent diffusion (non-premixed) flames. However, you can apply Flamelet models to partially premixed and even perfectly premixed flames by combining a Flamelet model with a flame position model. See Flame Propagation.

For guidance on which of the combustion models are best for a selection of applications, see Combustion and Other Reacting Flows in STAR-CCM+.

A few parameters, such as mixture fraction, reaction progress, scalar dissipation, and enthalpy, are used to parameterize the thermochemical state (species composition and temperature) of the system. Since the number of parameters are small, it is feasible to use presumed Probability Density Functions (PPDF) to account for fluctuations. A Beta function approximates the distribution of mixture fraction as it is a conserved scalar. The Beta function requires a variance, or second moment, which can be solved from a transport equation in RANS and an algebraic expression in LES. The Flamelet models use beta PPDF functions for mixture fraction, and delta functions for the other parameters such as enthalpy and scalar dissipation.

You use Flamelet models for gas-phase flames where the thermo-chemistry occurs near the flamelet manifold. For example, in statistically steady, constant pressure combustors, such as gas-turbines at full power.

Flamelet Generated Manifold Model

The Flamelet Generated Manifold (FGM) model parameterizes flamelet species by mixture fraction and reaction progress variable. To generate the flamelet manifold, the FGM implementation in STAR-CCM+ uses 0D ignition reactors. To account for non-adiabatic effects and heat loss or gain, species are also parameterized with enthalpy. The FGM model assumes that the chemical states in turbulent flames are similar to the chemical states in laminar flames. Unlike SLF, no assumption is made about flame structure. You can use the FGM model for modeling:

- High Damkohler-number zones, where reaction zones are thin.
- Low Damkohler-number zones, where reaction zones are diffuse. However, this model does not capture slowly forming pollutants.
- Premixed and partially premixed flames.

To initiate flame ignition, the FGM model provides the **Progress Variable Ignitor**. Access this ignitor by right-clicking the **Continuum** > **Ignitors** node. When active, a Progress Variable Ignitor applies its specified progress variable value to the cells that are contained within the ignitor volume shapes. When inactive, this ignitor does not affect the value of the progress variable.

Chemical Equilibrium Model

The Chemical Equilibrium model parameterizes species by the mixture fraction and enthalpy, or heat loss/gain. The chemistry is assumed to be infinitely fast so that thermo-chemical equilibrium is reached instantaneously. Since chemical equilibrium is a thermodynamic property and independent of the chemical mechanism, the Chemical Equilibrium model is appropriate for modeling:

- fuels for which a chemical mechanism is not available
- combustors where the residence time is large enough that the thermo-chemistry is near chemical equilibrium—for example, in fires

Steady Laminar Flamelet Model

The Steady Laminar Flamelet (SLF) model solves 1D opposed-flow diffusion flames and parameterizes species by mixture fraction and scalar dissipation, which corresponds to the strain (compression) of the opposed flow flame. The SLF model embeds these opposed-flow flames in the CFD turbulent flame brush and assumes that the local turbulent flow field compresses the flamelets. The chemistry is assumed to be fast and therefore responds to the local turbulent strain instantaneously. However, it is not possible to model extinction or ignition with the SLF model.

Since a non-premixed flame is resolved, the SLF model is suitable for modeling steady state diffusion combustion, such as in non-premixed furnaces and burners.

Flamelet Tables

You can create flamelet tables for each of the flamelet models within STAR-CCM+. If you run STAR-CCM+ in parallel, the flamelet table is generated in parallel. Each flamelet model table generator allows you to specify the chemistry definition, the composition of fluid streams, and several parameters, including the ability to generate the flamelet table using adaptive gridding. All flamelet variables that are plotted are given in SI units, which cannot be changed—this is regardless of any preferred unit choice.

Adaptive gridding places table points such that table interpolation is accurate to a tolerance that you specify. Adaptive gridding is useful for resolving flamelet tables where the variables that are stored in the table change non-linearly. Since adaptive tables generally provide more accurate as well as smaller, tables, adaptive gridding is recommended for all table dimensions except the FGM Progress Variable Variance. For FGM Progress Varaible Variance, a single point (delta function PDF) is recommended.

When using adaptive gridding, STAR-CCM+ starts to generate the flamelet table using an initial coarse mesh. STAR-CCM+ then inserts a new table point midway between these existing points. If the difference between linear interpolation of the two existing points and the newly inserted point is less than the normalized tolerance, the newly inserted point is discarded. If not, the process is repeated until the maximum number of grid points are reached. The normalized tolerance is the tolerance that you specify, multiplied by the maximum value of the variable in this table, minus the minimum value of the variable in the table. The algorithm is applied to temperature and all of the post-processing species that you specify.

See

- Chemical Equilibrium Table Reference
- FGM Table Reference
- Steady Laminar Flamelet Table Reference

Contents:

Flame Propagation
Flamelet Workflow
Flamelet Reference

Flame Propagation

STAR-CCM+ has two types of flame position models, which you can use with all flamelet models.

Each flamelet model describes how species and temperature is calculated in the flow field—it defines the species and temperature in the flame and after the flame. A flame position model is used to calculate the flame movement in space, for premixed and partially-premixed systems, through calculation of the turbulent flame speed.

You can select one of the following flame position models in STAR-CCM+:

- Coherent Flame Model (CFM)
- Turbulent Flame-speed Closure (TFC)

Both of the flame position models solve for the flame position by transporting a reaction progress variable. Normalized product species are used to define the reaction progress variable—so that it is zero for unburnt reactants and one for burnt products at chemical equilibrium.

Coherent Flame Model (CFM)

The CFM model solves an additional transport equation for the Flame Surface Density, which represents the extent of convolution of the premixed flame. The Flame Surface Density is then used to calculate the source term for the reaction progress variable. The CFM model is designed for simulating flames where the fuel and oxidizer are perfectly mixed before entering the computational domain.

To initiate combustion, the CFM model provides the **Flame Area Density ignitor**, accessible by right-clicking the **Continuum** > **Ignitors** node. While active, the Flame Area Density ignitor applies its specified value to the cells that are contained within the ignitor volume shapes. When inactive, this ignitor does not affect the values of flame area density.

Turbulent Flame-speed Closure (TFC)

The TFC model uses an algebraic expression, called the turbulent flame speed, to calculate the source term for the mean reaction progress variable equation.

To initiate combustion, the TFC model provides the **Progress Variable Ignitor** that you access by right-clicking the **Continuum** > **Ignitors** node. While active, the Progress Variable ignitor applies its specified progress variable to the cells that are contained within the ignitor volume shapes. When inactive, this ignitor does not affect the values of the progress variable.

Flamelet Workflow

Follow the steps in this workflow to simulate combustion using the flamelet models that are provided in STAR-CCM+.

The steps in this workflow are intended to follow on from the initial steps in the Reacting Flow General Workflow.

1. For the physics continuum that represents the reacting flow, select the following models—in addition to the models that are previously selected, with **Auto-Select recommended models** activated:

Group Box	Model
Reacting Flow Models	**Flamelet**
Flamelet Models	• For perfectly-premixed and partially-premixed flames, the **Flamelet Generated Manifold (FGM)** model is recommended. • For non-premixed flames, the **Steady Laminar Flamelet** model is recommended. • If no chemical mechanism is available, use the **Chemical Equilibrium** model.
Progress Variable Source (for FGM)	• **FGM Kinetic Rate** • **Coherent Flame Model (CFM)**. • **Turbulent Flame Speed Closure (TFC)**.

Group Box	Model
Flame Type (for Chemical Equilibrium or Steady Laminar Flamelet)	• **Non-Premixed Flame** • **Partially-Premixed Flame**
Flame Propagation (required for partially-premixed flames)	• **Coherent Flame Model (CFM)**. • **Turbulent Flame Speed Closure (TFC)**.

2. Select any Optional models.

   For example:

   - **NOx Emission** or **Soot Emissions**—to model the formation of these pollutants.
   - **Inert Stream**—allows the simulation to model an inert (chemically inactive) stream in addition to the flamelet fuel and oxidizer streams. This model allows you to define the inert stream species outside of the flamelet table, which avoids unnecessary computation of inactive species in the flamelet generation and tabulation. Useful when simulating exhaust gas recirculation (EGR) in internal combustion engines (ICE), and steam injection in gas-turbines.
   - The **Radiation** model is useful for modeling applications in which radiative heat transfer is important—as is the case for most combustion systems—such as in glass furnaces and in gas turbines. The soot emission models influence the **Participating Media Radiation (DOM)** and **Gray Thermal Radiation** models by contributing to the absorption coefficient of the continuous phase (the absorption coefficient describing both absorption and emission).
   - **Gravity**—if gravity forces significantly influence the solution, such as in fire simulations.

When the model selection process is complete, you then generate the flamelet table.

3. Import the mechanism which represents the reacting flow reactions.

   You can import the chemistry definition in Standard format—using ASCII files and then have STAR-CCM+ generate the flamelet tables directly. See Reaction Mechanism Formats. Alternatively, if you have a DARS FGM library (for FGM) or DARS Flamelet library (for SLF), you can import these DARS-generated flamelet library files into STAR-CCM+ and construct the table.

   - To import the chemistry definition in Standard format:
     1. Right-click the **Continuum** > **[Table Generator]** > **Chemistry Definition** node and select **Import Complex Chemistry Definition (Chemkin format)**.
     2. Select appropriate files for each of the required fields.
        - *Gas Chemistry Reaction File*: Select a file which contains the chemistry of the reactions.
        - *Gas Thermodynamic Properties File*: Select a file which contains details of the thermal properties.
        - *Import Gas Transport Properties File*: Importing this file is optional. If you select a gas transport properties file here, STAR-CCM+ calculates the mean molecular dynamic viscosity, thermal conductivity, and molecular diffusivity from the flamelet species and temperature. The **Multi-Component Gas** > **Material Properties** node then contains the option to set the *Method* to **Flamelet Table**.

- *Gas Transport Properties File*: If activated, select a file which contains details of the species transport properties.

- To read a DARS library file into STAR-CCM+:
  1. For the FGM table generator, right-click the **Continuum** > **[Table Generator]** node and select **Read DARS FGM Library and Construct Table**.
  2. For the SLF table generator, right-click the **Continuum** > **[Table Generator]** node and select **Read DARS Flamelet Library and Construct Table**.

4. Specify the composition and temperature of the fluid streams for flamelet generation. Make sure that the sum of each stream equals `1.0`.
   a) Select the **[Table Generator]** > **Fluid Streams** > **Fuel** node and specify the components of the fuel stream. If the fuel is premixed or partially-premixed with the oxidizer, include the correct proportions of fuel and oxidizer within this stream.
   b) Select the **[Table Generator]** > **Fluid Streams** > **Oxidizer** node and specify the components of the oxidizer stream. By default, the composition of the oxidizer stream is set to that of air.
5. Specify the table parameters—particularly, the table pressure.

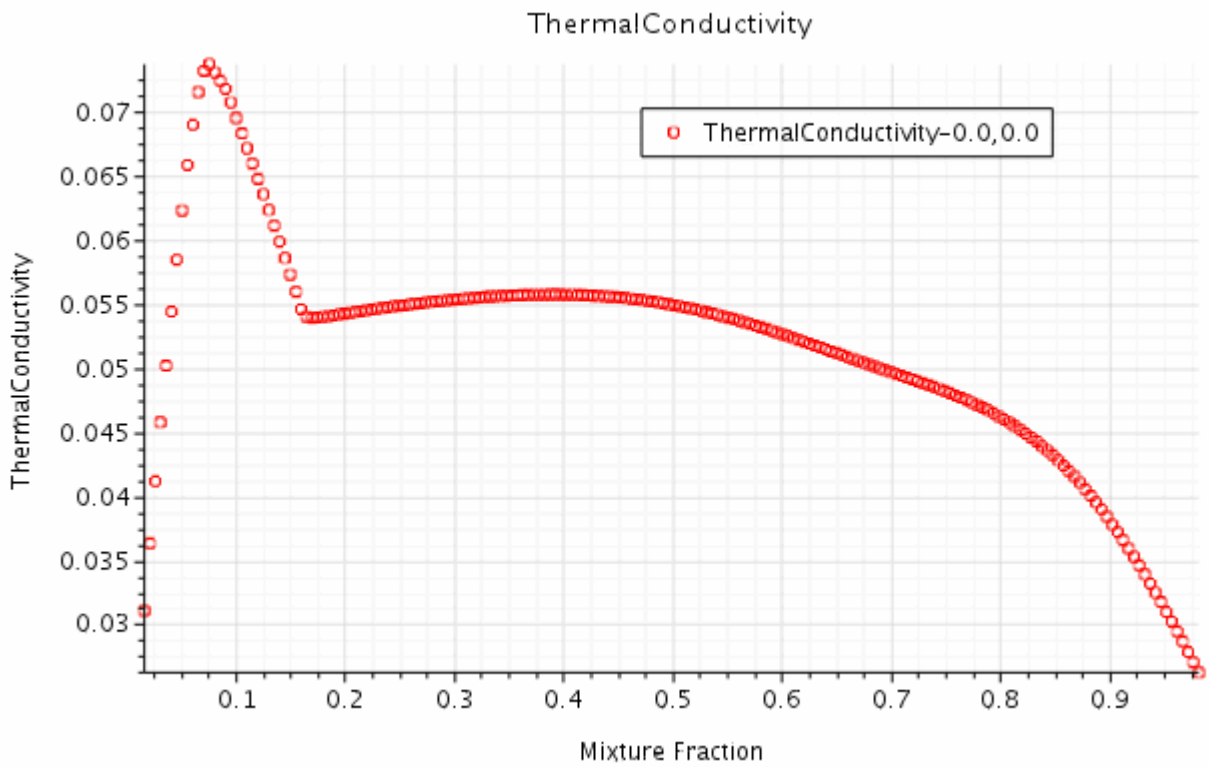
   By default, the specified fuel and oxidizer species, as well as CO2 and H2O, are defined for post-processing. However, make sure that you specify any more species that are to be included in the table. For example, you can add CO for monitoring emissions, or OH for monitoring the position of the flame front. See,

   - FGM Table Reference
   - Steady Laminar Flamelet Table Reference
   - Chemical Equilibrium Table Reference
6. Right-click the **[Table Generator]** > **[Table]** node and select **[Generate Library and Construct Table]**
7. To visualize values for variables that are stored in the flamelet table:
   a) Right-click the **[Table Generator]** > **[Table]** > **[XY Plots]** node, then select **New Plot** and the **[variable]** to plot.
   b) Right-click the **[XY Plots]** > **[variable]** node and select **Add New Table View for Plotting**.
   c) In the *Select Points for Plotting* dialog, enter the parameters that are required for the table and click **OK**.

      See Flamelet Table XY Plots Reference.
   d) To display the XY plot, double-click the **[XY Plots]** > **[variable]** node.

   If required, you can also change the *X-Axis Variable* property to display different grid resolution variables.

   For example, the following XY plot shows the thermal conductivity variable for the mixture fraction grid resolution variable:



When the flamelet table generation is complete, you then define any properties that are required for the physics models that you are using.

8. Set the properties of any **[Continuum]** > **Models** and model sub-nodes, as required.

   For example, if you are using the CFM or TFC models, make sure that all necessary reaction constants are specified.

   - If you are using the **Flamelet Generated Manifold (FGM)** model, select the **Flamelet Generated Manifold (FGM)** > **Progress Variable Variance** node and set the method for solving the progress variable variance. See: Flamelet Generated Manifold Model Reference.
   - If you are using the **Steady Laminar Flamelet** model, by default, the Steady Laminar Flamelet model selects the **Ideal Gas** model and the **Non-Adiabatic** model—which is necessary even for insignificant heat losses/gains.
   - For the CFM model, set the following properties under the **Coherent Flame Model (CFM)** node.
     - *CFM Constant, Alpha*, α in Eqn. (3426).
     - *CFM Constant, Beta*, β, in Eqn. (3426).
     - *CFM Constant, A*, the flame surface annihilation coefficient, A.
     - *CFM Constant, B*, the flame quenching coefficient by stretch, B.
   - For the TFC model, set the *TFC Rate Coefficient, A* constant, A in Eqn. (3420), under the **Turbulent Flame Speed Closure (TFC)** > **TFC Rate Coefficient, A** node.

9. Return to the Reacting Flow General Workflow.

Flamelet Reference

This section provides reference material, including properties, sub-nodes, field functions, and other model-specific settings, for all of the flamelet models.

Contents:

Chemical Equilibrium Model Reference
Chemical Equilibrium Table Reference
Steady Laminar Flamelet Model Reference
Steady Laminar Flamelet Table Reference
Flamelet Generated Manifold Model Reference
FGM Table Reference
FGM Reaction Model Reference
FGM Kinetic Rate Model Reference
Table Dimensions Reference
Flamelet Table XY Plots
Fluid Streams Reference
Restoring the Premixed FGM Model
Coherent Flame Model Reference
Restoring the Premixed Coherent Flame Model
Turbulent Flame-Speed Closure Model Reference
Restoring the Premixed Turbulent Flame-Speed Closure Model

Chemical Equilibrium Model Reference

The Chemical Equilibrium model assumes local instantaneous chemical equilibrium conditions.

In this model, a few parametric variables such as the mean mixture fraction (and in non-adiabatic cases, the mean enthalpy) are tracked on the grid. The mean species concentrations, temperature, and density are obtained as functions of the mean parametric variables, after an averaging process around a presumed probability distribution (PPDF) that is considered to represent turbulent fluctuations.

Chemical Equilibrium Model Reference

Theory	See Chemical Equilibrium.
Provided By	**[physics continuum]** > **Models** > *Flamelet Models*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Chemical Equilibrium**
Requires	*Material*: **Multi-Component Gas** *Reaction Regime*: **Reacting** *Reacting Flow Models*: **Flamelet**
Properties	Key properties are: Active Species, Dissipation Constant, Number of Streams. See Chemical Equilibrium Model Properties.

Activates	Physics Models	*Equation of State*: **Ideal Gas** *Energy Option*: **Non-Adiabatic PPDF**
	Reference Values	Minimum Allowable Temperature, Maximum Allowable Temperature. See Reference Values.
	Other Continuum Nodes	Chemical Equilibrium Table Generator. See Chemical Equilibrium Table Reference.
	Monitors	**MixFrac0**: Mixture Fraction 0 **MixFracVar0**: Mixture Fraction Variance 0
	Solvers	PPDF Combustion (only after a *Flow* model is also selected)
	Field Functions	Mixture Fraction 0, Mixture Fraction Variance 0. See Field Functions.

Chemical Equilibrium Model Properties

Active Species

Read only. Displays, from the species that are available for post-processing in the flamelet table, a sub-set of species that are accessed, used, or interpolated during the simulation. For example, species that are used in plots, reports, monitors, scenes, or custom field functions. The remaining post-processing species in the flamelet table that are not accessed during the simulation are inactive.

Dissipation Constant

Represents the ratio of velocity and chemical species fluctuations.

Number of Streams

You can choose to have either two or three streams.

Reference Values

Minimum Allowable Temperature

The smallest temperature value that is permitted anywhere in the continuum.

The Energy models (Coupled Energy, Coupled Solid Energy, Segregated Solid Energy, Segregated Fluid Enthalpy, Segregated Fluid Temperature) limit temperature corrections such that the corrected value does not go below this minimum. If this occurs, a message is printed to the *Output* window.

Maximum Allowable Temperature

The largest temperature value that is permitted anywhere in the continuum.

The Energy models (Coupled Energy, Coupled Solid Energy, Segregated Solid Energy, Segregated Fluid Enthalpy, Segregated Fluid Temperature) limit temperature corrections such that the corrected value does not exceed this maximum. If this occurs, a message is printed to the *Output* window.

PPDF Combustion Solver Properties

Under-Relaxation Factor

In order to promote convergence, this property is used to under-relax changes of the solution during the iterative process. If residuals show solution divergence and do not decrease, reduce the under-relaxation factor for the relevant solvers.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Field Functions

Mixture Fraction 0

Represents the stoichiometry of the fuel stream ratio in a combustion system consisting of three separate streams.

Mixture Fraction Variance 0

Represents the turbulent fluctuations in the fuel mixture fraction values.

Chemical Equilibrium Table Reference

The Chemical Equilibrium model uses a flamelet table called the Chemical Equilibrium Table.

Chemical Equilibrium Table Generator

The Chemical Equilibrium Table Generator node contains the following sub-nodes:

- Chemistry Definition
- Fluid Streams (see Fluid Streams Reference)
- Parameters

- Chemical Equilibrium Table

Chemistry Definition

The Chemistry Definition describes the chemical reaction mechanism, which is the collection of all species in a reaction—reactants, intermediates, and products—and the corresponding set of all intermediate reactions.

Properties

None

Right-Click Actions

Import Complex Chemistry Definition (Chemkin format)

Activates a standard *Open* dialog that imports a library file for the complex chemistry definition in Chemkin™ format. Upon importing a chemistry definition, the reactions for the definition appear as sub-nodes of the Chemistry Definition node.

Select Mixture Components

Opens a dialog which allows you to select the reaction components for the chemistry definition from the Material Databases that are available.

Delete Complex Chemistry Definition

Removes the library that is used for the complex chemistry definition.

Parameters

Properties

Mixture Fraction Grid Stretch Factor

Allows you to set the stretch factor for the z grid of the flamelet table.

Child Nodes

Absolute Pressure

Value

The pressure that is used when generating the flamelet table.

Species for Post Processing

Species

Allows you to select which species from the chemical mechanism are stored in the flamelet table. These species are then made available in a simulation, for example, to use for post-processing in contours or reports.

Emissions

Thermal NOx

When activated, the NOx source term coefficients are calculated and tabulated.

Numerical Settings

Reference Temperature

Specifies the minimum temperature for enthalpy normalization in the flamelet table.

Numerical Settings > Table Dimensions

Allows you to define the dimensions for the flamelet table. See Table Dimensions Reference.

Chemical Equilibrium Table

Properties

High Mixture Fraction Limit

For mixture fraction values above a certain limit, the mixture can fail to burn because it is too fuel-rich. STAR-CCM+ does not perform the chemical equilibrium calculation above this limit, but can obtain a composition for a non-reacting state that is based on pure mixing. To impose this non-reacting state, set this property at less than 1.0.

High Limit Window Size

Allows for a gradual change in flow properties from the reacting to the non-reacting state. Specifically, when you set High Mixture Fraction Limit to a value less than 1.0, the transition from the reacting chemical equilibrium state to the non-reacting pure mixing state can be abrupt as well as discontinuous. The greater the High Limit Window Size (from 0.0 through 1.0), the smoother the transition between the two states.

Right-Click Actions

Generate Chemical Equilibrium and Construct Table

Generates Chemical Equilibrium flamelet libraries and constructs flamelet table.

Export Table

Saves a table that is stored in STAR-CCM+'s memory to a file with extension `.tbl`.

Delete Table

Removes the complex chemistry definition.

Stop Calculations

Child Nodes

Final Table Grid Size

Contains sub-nodes for each of the table dimensions. Each table dimensions sub-node displays the read only, *Number of Grid Points*, that are used from the data for that parameter. The values are specified before building the table by defining settings under the **Table Dimensions** node. See Table Dimensions Reference.

Chemical Equilibrium XY Plots

To visualize the chemical equilibrium flamelet table, you can create an XY plot—right-click this node and select New Plot. See Flamelet Table XY Plot Reference.

Steady Laminar Flamelet Model Reference

The Steady Laminar Flamelet model accounts for chemical non-equilibrium and finite-rate chemistry effects in turbulent non-premixed flames.

In this model, a few parametric variables such as the mean mixture fraction (and in non-adiabatic cases, the mean enthalpy) are tracked on the grid. The mean species concentrations, temperature, and density are obtained as functions of the mean parametric variables, after an averaging process around a presumed probability distribution (PPDF) that is considered to represent turbulent fluctuations.

Steady Laminar Flamelet Model Reference

Theory	See Steady Laminar Flamelet.	
Provided By	**[physics continuum]** > **Models** > *Flamelet Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Steady Laminar Flamelet**	
Requires	*Material*: **Multi-Component Gas** *Reaction Regime*: **Reacting** *Reacting Flow Models*: **Flamelet**	
Properties	Key properties are: Active Species, Dissipation Constant, Number of Streams. See Steady Laminar Flamelet Model Properties.	
Activates	Physics Models	*Equation of State*: **Ideal Gas** *Energy Option*: **Non-Adiabatic PPDF**
	Reference Values	Minimum Allowable Temperature, Maximum Allowable Temperature. See Reference Values.
	Other Continuum Nodes	SLF Table Generator. See Steady Laminar Flamelet Table Reference .
	Solvers	PPDF Combustion (only after a *Flow* model is also selected)
	Field Functions	Mixture Fraction 0, Mixture Fraction Variance 0, Scalar Dissipation Rate. See Field Functions.

Steady Laminar Flamelet Model Properties

Active Species

Read only. Displays, from the species that are available for post-processing in the flamelet table, a sub-set of species that are accessed, used, or interpolated during the simulation. For example, species that are used in plots, reports, monitors, scenes, or custom field functions. The remaining post-processing species in the flamelet table that are not accessed during the simulation are inactive.

Dissipation Constant

Represents the ratio of velocity and chemical species fluctuation time scales.

Number of Streams

The Steady Laminar Flamelet model allows you to have two streams.

Reference Values

Minimum Allowable Temperature

The smallest temperature value that is permitted anywhere in the continuum.

The Energy models (Coupled Energy, Coupled Solid Energy, Segregated Solid Energy, Segregated Fluid Enthalpy, Segregated Fluid Temperature) limit temperature corrections such that the corrected value does not go below this minimum. If this occurs, a message is printed to the *Output* window.

Maximum Allowable Temperature

The largest temperature value that is permitted anywhere in the continuum.

The Energy models (Coupled Energy, Coupled Solid Energy, Segregated Solid Energy, Segregated Fluid Enthalpy, Segregated Fluid Temperature) limit temperature corrections such that the corrected value does not exceed this maximum. If this occurs, a message is printed to the *Output* window.

PPDF Combustion Solver Properties

Under-Relaxation Factor

In order to promote convergence, this property is used to under-relax changes of the solution during the iterative process. If residuals show solution divergence and do not decrease, reduce the under-relaxation factor for the relevant solvers.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Field Functions

Mixture Fraction 0

Represents the stoichiometry of the fuel stream ratio in a combustion system consisting of three separate streams.

Scalar Dissipation Rate

χ in Eqn. (3404).

Steady Laminar Flamelet Table Reference

This section describes the parameters that you set for the SLF Table which the Steady Laminar Flamelet model uses.

SLF Table Generator

The SLF Table Generator node contains the following sub-nodes:

- Chemistry Definition
- Fluid Streams (see Fluid Streams Reference)
- Parameters
- SLF Table

Chemistry Definition

The Chemistry Definition describes the chemical reaction mechanism, which is the collection of all species in a reaction—reactants, intermediates, and products—and the corresponding set of all intermediate reactions.

Properties

None

Right-Click Actions

Import Complex Chemistry Definition (Chemkin format)

Activates a standard *Open* dialog that imports a library file for the complex chemistry definition in Chemkin™ format. Upon importing a chemistry definition, the reactions for the definition appear as sub-nodes of the Chemistry Definition node.

Delete Complex Chemistry Definition

Removes the library that is used for the complex chemistry definition.

Parameters

Properties

Mixture Fraction Grid Stretch Factor

Allows you to set the stretch factor for the z grid of the flamelet table.

Child Nodes

Absolute Pressure

Value: The pressure that is used when generating the flamelet table.

Species for Post Processing

Species: Allows you to select which species from the chemical mechanism are stored in the flamelet table. These species are then made available in a simulation, for example, to use for post-processing in contours or reports.

Emissions

- *Thermal NOx*: When activated, the NOx source term coefficients are calculated and tabulated.
- *Soot Moment*: When activated, creates a **Soot Moments Options** sub-node which specifies the Nucleation Option as **C2H2-based** (currently the only option). See Soot Moments.

Numerical Settings

The Numerical Settings node allows you to define the following properties. It also contains the **Table Dimensions** sub-node which you can use to define the dimensions for the flamelet table. See Table Dimensions Reference.

- *Absolute Tolerance*: Specifies the absolute tolerance of the solver.
- *Relative Tolerance*: Specifies the relative tolerance of the solver.
- *Analytical Jacobian*: Uses analytical Jacobian.
- *Flamelet Relative Tolerance*: The flamelet equations are integrated in time using the ODE solver Absolute Tolerance and Relative Tolerance until steady-state. Steady state is determined when the average residual over all species and temperature and all grid points is less than the Flamelet Relative Tolerance.

- *Initial Scalar Dissipation (/s)*: Specifies the scalar dissipation rate, χ in Eqn. (3283), for the first steady laminar flamelet.
- *Maximum Scalar Dissipation (/s)*: Specifies the largest scalar dissipation, χ in Eqn. (3283), that is allowed. Usually, flamelets extinguish well before this value. See Scalar Dissipation Rate.
- *Scalar Dissipation Multiplier*: The sequence of flamelet scalar dissipation rates are calculated as follows: Starting from the initial scalar dissipation rate, the next scalar dissipation is multiplied by 10/s (if the current scalar dissipation is less than 1/s). Else it is multiplied by the scalar dissipation multiplier, until either the flamelet extinguishes or else the maximum scalar dissipation rate is exceeded. See Scalar Dissipation Rate.

SLF Table

Properties

None

Right-Click Actions

Generate Flamelet Library and Construct Table

Generates steady laminar flamelet libraries and constructs PPDF table.

Export Table

Saves a table that is stored in the memory of to a file with extension `.tbl`.

Delete Table

Removes the complex chemistry definition.

Stop Calculations

Child Nodes

Final Table Grid Size

Contains sub-nodes for each of the table dimensions. Each table dimensions sub-node displays the read only value, *Number of Grid Points*, that are used from the data for that parameter. The values are specified before building the table by defining settings under the **Table Dimensions** node. See Table Dimensions Reference.

SLF XY Plots

To visualize the steady laminar flamelet table, you can create an XY plot—right-click this node and select New Plot. See Flamelet Table XY Plots Reference.

Flamelet Generated Manifold Model Reference

The Flamelet Generated Manifold (FGM) model is designed to approximate complex chemical mechanisms with a reduced computational cost.

FGM Model

For simulations involving combustion, it is often important to include a detailed reaction mechanism, which has many reactions and many species. Inclusion of such a mechanism in a CFD simulation with realistic geometries, however, inevitably leads to a large computational cost.

In the FGM model, the effects of detailed chemistry on combustion are stored in a Flamelet table and the required values are retrieved during the CFD simulation. A progress variable is used to facilitate interactions between CFD computation and the Flamelet table.

FGM Model Reference

Theory	See Flamelet Generated Manifold.	
Provided By	**[physics continuum]** > **Models** > *Flamelet Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Flamelet Generated Manifold (FGM)**	
Requires	*Material*: **Multi-Component Gas** *Reaction Regime*: **Reacting** *Reacting Flow Models*: **Flamelet**	
Properties	Key properties are: **FGM Constant, Alpha**. See FGM Model Properties.	
Activates	Physics Models	**FGM Reaction** and **Ideal Gas Equation of State for FGM**
	Model Controls (child nodes)	Progress Variable Variance
	Materials	See Materials and Methods.
	Initial Conditions	**Mixture Fraction Profile** and **Mixture Fraction Variance Profile**. See Initial Conditions.
	Boundary Inputs	**Wall Combustion Scalar Option**. See Boundary Settings.
	Region Inputs	**Active Reactions Option**. See Region Settings.
	Other Continuum Nodes	• **FGM Table**. See FGM Table Reference. • The **Ignitors** node provides the right-click option to create a **Progress Variable Ignitor**. See Ignitors.
	Solvers	**FGM Combustion**. See Solvers.
	Field Functions	Combustion Scalar Diffusion Coefficient, Heat Loss Ratio, Mach Number, Mixture Fraction 0, Mixture Fraction Variance 0, Normalized Progress Variable Variance, Progress Variable, Progress Variable Variance, Unnormalized Progress Variable Variance. See Field Functions.

Flamelet Generated Manifold (FGM) Model Properties

FGM Constant, Alpha

A scaling factor to the table source term for the chemical enthalpy.

Secondary Gradients

Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces.

- **On**: Default value. Solves for interior and boundary types of secondary gradient.
- **Off**: Does not solve for either type of secondary gradient.
- **Interior Only**: Solves for the interior secondary gradients only.
- **Boundaries Only**: Solves for the boundary secondary gradients only.

Convection

In transport equations, you can choose from a range of schemes that calculate the convection term at a cell face. This calculation requires STAR-CCM+ to compute the face value of a quantity from the surrounding cell values. The method used for computing this face value has a profound effect on the stability and accuracy of the numerical scheme. For guidance on selecting a convection scheme, see Convective Flux.

- **1st-order**: First-order convection scheme.
- **2nd-order**: Second-order convection scheme.

Materials and Methods

Molecular Weight

The Molecular Weight Method is set to Flamelet Table by default. This setting specifies that all values for Molecular Weight are taken from the values in the FGM Flamelet Table.

Specific Heat

The Specific Heat Method is set to Flamelet Table by default. This setting specifies that all values for Specific Heat are taken from the values in the FGM Flamelet Table.

Initial Conditions

Mixture Fraction Profile

The mixture fraction is a measure of the local fuel/oxidizer ratio in a combustion system consisting of separate fuel and oxidizer streams. The Mixture Fraction Profile represents the reacting proportion of a fluid stream—the atomic mass fraction that originated from the fuel stream. 1.0 is equal to 100%. To initialize with pure oxidizer, maintain the default of 0.0. For inflows where the mixture is unburnt, the Mixture Fraction Profile is the mass fraction of the fuel.

Mixture Fraction Variance Profile

The mixture fraction variance is a measure of turbulent fluctuations in the local mixture fraction values. It is a scalar quantity whose initial and boundary condition values are entered in the Properties window as a scalar array profile. In simulations that use the Large Eddy Simulation (LES) model, you can select one of two methods for mixture fraction variance:

- **Transport Equation**: Uses standard STAR-CCM+ calculations.
- **Algebraic Relationship**: Activates the Algebraic Relationship node which allows you to set values for the model constant, C_v in Eqn. (3414), and the Internal Dissipation Constant, C_d in Eqn. (3415).

Boundary Settings

Wall

Wall Combustion Scalar

Allows you to select the scalars for the wall combustion calculation.

Method	Corresponding Value Nodes
Zero Flux	None.
Specified Value	When the Partially-Premixed Combustion or Non-Premixed Combustion models are selected, all of the following values are available to specify. When the Premixed Combustion model (available for the FGM model) is selected, only the Progress Variable value is available: **Mixture Fraction Profile** The mixture fraction is a measure of the local fuel/oxidizer ratio in a combustion system consisting of separate fuel and oxidizer streams. The Mixture Fraction Profile represents the reacting proportion of a fluid stream—the atomic mass fraction that originated from the fuel stream. 1.0 is equal to 100%. To initialize with pure oxidizer, maintain the default of 0.0. For inflows where the mixture is unburnt, the Mixture Fraction Profile is the mass fraction of the fuel. **Mixture Fraction Variance Profile** The mixture fraction variance is a measure of turbulent fluctuations in the local mixture fraction values. It is a scalar quantity whose initial and boundary condition values are entered in the Properties window as a scalar array profile. In simulations that use the Large Eddy Simulation (LES) model, you can select one of two methods for mixture fraction variance: • **Transport Equation**: Uses standard STAR-CCM+ calculations. • **Algebraic Relationship**: Activates the Algebraic Relationship node which allows you to set values for the model constant, C_v in Eqn. (3414), and the Internal Dissipation Constant, C_d in Eqn. (3415). **Progress Variable** The progress variable is a dimensionless parameter, Eqn. (3412), that shows the progress of the combustion. The progress variable is a scalar quantity, with Method set to Constant by default. The Constant value is 0 for the unburnt state and 1 for the fully burnt state. You can set a value for the Constant property of the Progress Variable node or select a scalar array profile for the Method property and define the scalar profile values appropriately.

	Mass Flux
Specified Flux	The following values are available to specify: **Mass Flux** **Mixture Fraction Profile** The mixture fraction is a measure of the local fuel/oxidizer ratio in a combustion system consisting of separate fuel and oxidizer streams. The Mixture Fraction Profile represents the reacting proportion of a fluid stream—the atomic mass fraction that originated from the fuel stream. 1.0 is equal to 100%. To initialize with pure oxidizer, maintain the default of 0.0. For inflows where the mixture is unburnt, the Mixture Fraction Profile is the mass fraction of the fuel. **Static Temperature**

Region Settings

Applies to any region:

Active Reactions Option

Allows you to activate or deactivate chemical reactions in this region.

FGM Combustion Solver Properties

Under-Relaxation Factor

In order to promote convergence, this property is used to under-relax changes of the solution during the iterative process. If residuals show solution divergence and do not decrease, reduce the under-relaxation factor for the relevant solvers.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Field Functions

Combustion Scalar Diffusion Coefficient

Progress Variable

Progress Variable Variance

Method	Corresponding Method sub-node
Transport Equation Computes the progress variable variance using a transport equation, Eqn. (3415).	
Algebraic Relationship Computes the progress variable variance using Eqn. (3414).	**Algebraic Relationship**—allows you to specify the Cv model constant, c_v, in Eqn. (3414).

FGM Table Reference

The parameters that you set for the FGM Table which the Flamelet Generated Manifold (FGM) model uses.

FGM Table Generator

The FGM Table Generator node contains the following sub-nodes:

- Chemistry Definition
- Fluid Streams (see Fluid Streams Reference)
- Parameters
- FGM Table

Chemistry Definition

The Chemistry Definition describes the chemical reaction mechanism, which is the collection of all species in a reaction—reactants, intermediates, and products—and the corresponding set of all intermediate reactions.

Properties

None

Right-Click Actions

Import Complex Chemistry Definition (Chemkin format)

Activates a standard *Open* dialog that imports a library file for the complex chemistry definition in Chemkin™ format. Upon importing a chemistry definition, the reactions for the definition appear as sub-nodes of the Chemistry Definition node. See Defining Chemical Reactions Manually.

If you import transport data along with the chemical mechanism and and thermal data, STAR-CCM+ tabulates the molecular transport properties for the Dynamic Viscosity, Molecular Diffusivity, and Thermal Conductivity material properties. You can then choose to use these tabulated molecular transport properties by selecting the `Flamelet Table` option for each of the Material Properties.

Delete Complex Chemistry Definition

Removes the library that is used for the complex chemistry definition.

Parameters

Properties

Mixture Fraction Grid Stretch Factor

Allows you to set the stretch factor for the z grid of the FGM table.

Progress Variable Definition

The un-normalized progress variable y is defined by Eqn. (3409).

- **Species Weights**: Adds a **Species Weights** sub-node which allows you to define the progress variable species weights W_k manually in Eqn. (3409).
- **Chemical Enthalpy**: The progress variable species weights, W_k in Eqn. (3409), are set to the chemical enthalpies (h) of each species. Values for h are taken from the thermodynamic properties file which is imported as part of the reaction mechanism. See Reaction Mechanism Formats: Gas / Surface Thermodynamic Properties Files.
- **Optimal Weights**: Calculates the optimal proportion of progress variable species weights W_k [661]. At the end of the optimization, the species weights are changed and are shown in the **Species Weights** sub-node. The optimal weights are calculated to minimize the gradients of the tabulated variables (ϕ) with respect to the progress variable $\frac{d\phi}{dy}$ and ensure that the progress variable increases monotonically through the flamelet.

Child Nodes

Absolute Pressure

Value

The pressure that is used when generating the FGM table.

Species for Post Processing

Species

Allows you to select which species are included in the FGM table for use in post processing.

Emissions

Thermal NOx

When activated, the NOx source-term coefficients are calculated and tabulated.

Numerical Settings

The **Numerical Settings** node allows you to define the following properties. It also contains the **Table Dimensions** sub-node which you can use to define the dimensions for the flamelet table. See Table Dimensions Reference.

Absolute Tolerance

Specifies the absolute tolerance of the solver.

Relative Tolerance

Specifies the relative tolerance of the solver.

Analytical Jacobian

When activated, an analytical Jacobian is used instead of the default numerical Jacobian.

Reference Temperature

Specifies the minimum temperature for enthalpy normalization in the flamelet table.

Species Weights

The following values for the progress variable species weights, W_k in Eqn. (3409), are set by default. However, these values can be edited manually prior to generating the table.

- If CO and CO2 are present in the mechanism, they are both set to 1.0 and all other species are set to 0.0.
- If neither CO or CO2 are present in the mechanism, but H2 and H2O are present (for example, during hydrogen combustion), H2O is set to 1.0 and all other species are set to 0.0.

After a table is generated, these values cannot be changed unless the table is deleted first.

FGM Table

The FGM Table node has the following options available from the context menu:

Properties

None.

Right-Click Actions

Generate FGM Library and Construct Table

This option allows you to generate the FGM library and construct the FGM table in STAR-CCM+ using the properties and table dimensions that are specified under the FGM table node. The progress is displayed in the Output window.

Export Table

Saves the FGM table that is stored in STAR-CCM+'s memory to a file with extension `.tbl`.

Delete Table

This option allows you to remove a stored table from memory.

Stop Calculations

This option allows you to stop the calculations during the FGM library generation.

Child Nodes

Final Table Grid Size

Contains sub-nodes for each of the table dimensions. Each table dimensions sub-node displays the (read only) *Number of Grid Points* that are used from the data for that parameter. The values are specified before building the table by defining settings under the **Table Dimensions** node. See Table Dimensions Reference.

FGM XY Plots

To visualize the flamelet generated manifold flamelet table, you can create an XY plot. Right-click this node and select New Plot. See Flamelet Table XY Plots Reference.

FGM Reaction Model Reference

The FGM Reaction model is selected automatically with the Flamelet Generated Manifold model.

FGM Reaction Model Reference

Theory	See Flamelet Generated Manifold.
Provided By	**[physics continuum]** > **Models** > *FGM Reaction*
Example Node Path	**Continua** > **Physics 1** > **Models** > **FGM Reaction**
Requires	*Material*: **Multi-Component Gas** *Reaction Regime*: **Reacting** *Reacting Flow Models*: **Flamelet** *Flamelet Models*: **Flamelet Generated Manifold (FGM)**
Properties	Key properties are: **Dissipation Constant** and **Number of Streams**. See FGM Reaction Model Properties.

Activates	Field Functions	Heat Loss Ratio, Mixture Fraction 0, Mixture Fraction Variance 0, Normalized Progress Variable Variance, Progress Variable Variance, Unnormalized Progress Variable Variance. See Field Functions.

FGM Reaction Model Properties

Dissipation Constant

Represents the ratio of velocity and chemical species fluctuation time scales.

Number of Streams

You can choose to have either two or three streams.

Field Functions

Heat Loss Ratio

γ in Eqn. (3399). When using the Inert Stream model, heat losses or gains from the fluid stream are also accounted for. See Inert Streams.

Mixture Fraction 0

Represents the stoichiometry of the fuel stream ratio in a combustion system consisting of three separate streams.

Mixture Fraction Variance 0

Represents the turbulent fluctuations in the fuel mixture fraction values.

Normalized Progress Variable Variance

c, in Eqn. (3412).

Progress Variable Variance

y_{var} in Eqn. (3413) or Eqn. (3415).

Unnormalized Progress Variable Variance

y in Eqn. (3410).

FGM Kinetic Rate Model Reference

The FGM Kinetic Rate model takes the progress variable source term from the FGM table.

The FGM Kinetic Rate model uses the chemical rate to integrate the source term, it doesn't include the effect of turbulence or the laminar flame speed. Therefore, this model is useful for combustion cases that are close to laminar, or when the chemistry occurs at a slower rate than the turbulence mixes.

FGM Kinetic Rate Model Reference

Theory	See Flamelet Generated Manifold.

Provided By	**[physics continuum]** > **Models** > *Progress Variable Source*
Example Node Path	**Continua** > **Physics 1** > **Models** > **FGM Kinetic Rate**
Requires	*Material*: **Multi-Component Gas** *Reaction Regime*: **Reacting** *Reacting Flow Models*: **Flamelet** *Flamelet Models*: **Flamelet Generated Manifold (FGM)** *FGM Reaction*: **FGM Rection**

Table Dimensions Reference

Different table dimensions (grid resolution) sub-nodes are available for each flamelet table.

You can specify the grid resolution for a flamelet table using the **[Table Generator]** > **Parameters** > **Numerical Settings** > **Table Dimensions** node.

When trying to reach a flamelet table grid independent solution, you can refine the grid resolution—for adaptive gridding tables you can decrease the tolerance, or for fixed tables you can increase the number of data points—then re-converge the solution until it stops changing.

Flamelet Table Type	Available Table Dimensions
Flamelet Generated Manifold (FGM)	**Heat Loss Ratio**, **Mixture Fraction**, **Mixture Fraction Variance**, **Progress Variable**, **Progress Variable Variance**
Steady Laminar Flamelet (SLF)	**Mixture Fraction**, **Mixture Fraction Variance**
Chemical Equilibrium (with two streams)*	**Mixture Fraction**, **Mixture Fraction Variance**
Chemical Equilibrium (with three streams)*	**Mixture Fraction**, **Mixture Fraction Variance**, **Mixture Fraction 0**, **Mixture Fraction 1**, **Mixture Fraction Variance 0**, **Mixture Fraction Variance 1**

*When using the Non-Adiabatic model, the **Heat Loss Ratio** table dimension is also available to specify.

- **Mixture Fraction**: Data points which represent the mixture fraction, Z. See Mixture Fraction.
- **Mixture Fraction Variance**: Data points which represent the turbulent fluctuations in the local mixture fraction values, Z_{var}.
- **Progress Variable**: Data points which represent the progress of the combustion, c.
- **Progress Variable Variance**: Data points which represent the fluctuations in the progress of the combustion, c_{var}.
- **Mixture Fraction 0**: Data points which represent the stoichiometry of the fuel stream ratio, Z_{fuel}.
- **Mixture Fraction 1**: Data points which represent the stoichiometry of the second stream ratio, Z_{ox}.
- **Mixture Fraction Variance 0**: Data points which represent the turbulent fluctuations in the mixture fraction values of the fuel stream, $Z_{fuel,\mathrm{var}}$.

- **Mixture Fraction Variance 1**: Data points which represent the turbulent fluctuations in the mixture fraction values of the second stream, $Z_{ox,\mathrm{var}}$.
- **Heat Loss Ratio**: Data points which represent the heat loss ratio (enthalpy), γ in Eqn. (3399). When using the Inert Stream model, heat losses or gains from the fluid stream are also accounted for. See Inert Streams.

Each of the Table Dimension sub-nodes has the following property:

Use Adaptive Number of Grid Points

Allows you to specify if adaptive gridding is used for this table dimension. For more details of how this feature adapts tables, see Flamelet Tables. If the table is populated, the property cannot be changed unless you delete the table.

- **Activated**: STAR-CCM+ creates a non-uniform grid for this table dimension by placing table points so that table interpolation is accurate to the tolerance that you specify. The **Grid Parameters** sub-node allows you to set the *Maximum Number of Grid Points*, and an interpolation *Tolerance* which is used to calculate a normalized tolerance.
- **Deactivated**: The **Grid Parameters** sub-node allows you to set the *Number of Grid Points* for the table dimension. STAR-CCM+ clusters grid points at stochiometric values for the mixture fraction grid, but uses equal spacing for the other dimensions of variance, reaction progress (Progress Variable) and enthalpy (Heat Loss Ratio).

Flamelet Table XY Plots

You can create XY Plots to visualize the values that are stored in a flamelet table for each variable.

[Flamelet Model] XY Plots

Properties

None.

Right-Click Actions

New Plot

Creates an XY Plot which allows you to visualize the variable that you select—appears as a **[Variable]** child node. The following XY plots are available to specify:

ACoeff

Thermal NOx coefficient A in Eqn. (3484).

BCoeff

Thermal NOx coefficient B in Eqn. (3484).

CCoeff

Thermal NOx coefficient C in Eqn. (3484).

DCoeff

Thermal NOx coefficient D in Eqn. (3484).

DYDT

The temporal derivative of the unnormalized progress variable source term ω_y in Eqn. (3410).

Density

DynamicViscosity

Enthalpy

MolecularDiffusivity

MolecularWeight

[SPECIES]

SensibleEnthalpy

SpecificHeat

Temperature

ThermalConductivity

DYSRCDY

The gradient of the source term, which is the Jacobian of the FGM kinetic source term ω_y in Eqn. (3410) (YSRC). That is the derivative of YSRC with respect to the unnormalized progress variable, y. This plot should be as smooth as possible—which you can ensure by setting the *Progress Variable Definition* to **Optimal Weights**.

Child Nodes

[Variable]

Properties

The flamelet table XY plots share some of the properties of the regular XY plots, with the addition of the properties that are described below. See XY Plot Properties

Variable Name

Displays the read-only name of the variable that was chosen to plot.

X-Axis Variable

Allows you to select from the grid resolution variables that are available which is used for the X-axis. For example, Heat Loss Ratio, Mixture Fraction, Mixture Fraction Variance, etc.

Right-Click Actions

The flamelet table XY plots share some of the right-click actions of the regular XY plots, with the addition of the right-click actions that are described below. See XY Plot Pop-Up Menu.

Add New Table View for Plotting

Activates a dialog in which you can choose to specify either values or table points to plot.

- In the *Select Values to Plot* tab, you can set specific values for the variables and the table is interpolated to these values. Since the points are interpolated, they appear equally spaced.
- In the *Select Table Points to Plot* tab, you can set the actual grid points to plot for the variables.

Child Nodes

Data Series

See Adding and Removing Data in a Plot.

Axes

See Customizing Axes.

Legend

See Customizing the Legend.

Fluid Streams Reference

You can define the fluid streams for a reacting flow within each of the flamelet table generators.

Fluid Streams

You define the number of fluid streams using the *Number of Streams* property of the flamelet model. When two streams are specified, the **[Flamelet Table Generator]** > **Fluid Streams** node contains the following sub-nodes:

- **Oxidizer**
- **Fuel**

When three streams are specified for chemical equilibrium, an additional sub-node appears, called **Second Stream**.

Note: It is not possible to specify a third stream when using the Steady Laminar Flamelet model.

All three flamelet models allow you to define an additional inert stream using the Inert Stream model. See Inert Streams.

The mass fractions of all components within a fluid stream must add up to 1.0. For example, at a boundary, the combined inert (mass) fraction and mixture (mass) fraction must be less than 1.0. The remaining mass fraction accounts for the oxidizer.

Each of the streams has the following Properties:

Fluid Stream Components

Allows you to select the fluid stream components for the selected fluid stream. You can select the fluid stream components from a list of the components that are already specified within the Multi-Component Gas or Multi-Component Liquid models.

Composition Specification

Allows you to specify the composition of the fluid stream using Mass fraction or Mole fraction.

Temperature

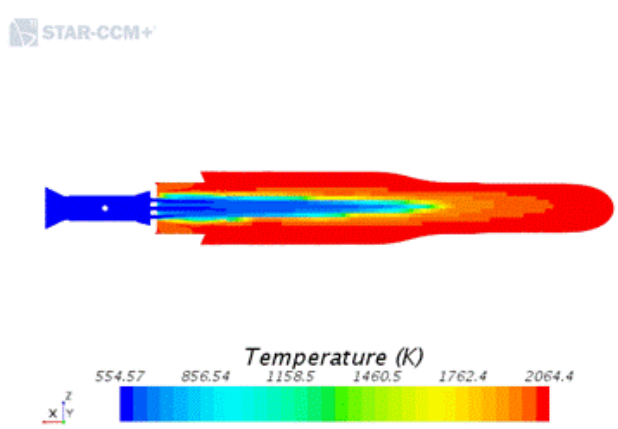
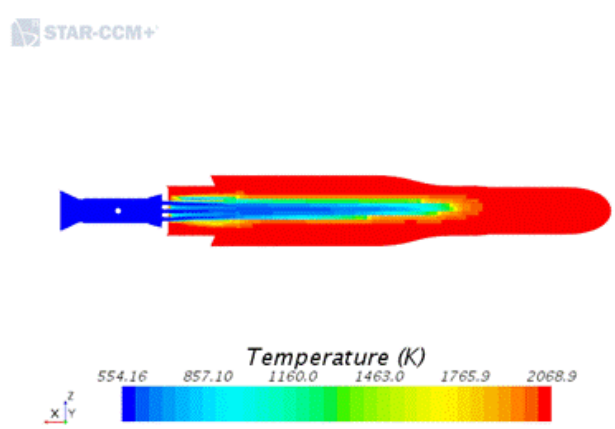
Allows you to set the temperature of the fluid stream.

Restoring the Premixed FGM Model

The Premixed Flamelet Generated Manifold (FGM) model in STAR-CCM+ v.11.04 is superceded by a Partially Premixed version. You can restore the Premixed model settings as described in this section.

As part of a plan to improve and simplify the procedure for selecting combustion models, the Premixed version of the FGM model is discontinued in STAR-CCM+ v.11.04.

When loading a simulation in STAR-CCM+ v.11.04 or later that was saved in a version of STAR-CCM+ before v11.04, the Premixed FGM model behaves as a Partially-Premixed FGM model. Several settings are defined differently and must be set up again. The following images show a comparison of the temperatures on the cross section of a flame holder.

Premixed FGM 11.02	Premixed FGM 11.04 (Partially Pre-Mixed)
	

STAR-CCM+ v.11.04 provides the FGM model as an option for Premixed Combustion. However, this model is the same as that provided for Partially-Premixed combustion. Further steps are required to restore the premixed behaviour that was available in STAR-CCM+ v.11.02 and earlier. These steps are as follows:

You can restore the settings of the deprecated Premixed combustion model, using the following procedure:

1. Redefine the reacting fluid stream to match the Premixed set-up.
   a) Select the **[Continuum]** > **Fluid Stream Manager** > **Fuel Stream** node and set the *Fluid Stream Components* as the Premixed mixture.
   b) Select the **Fuel Stream** node and set the *Temperature of the Stream* using the temperature from STAR-CCM+ v11.02 that is set for the inlet boundary of the fuel/air mixture.
   c) Select the **[Continuum]** > **Fluid Stream Manager** > **Oxidizer Stream** node and set the *Fluid Stream Components* as Air.
   d) Select the **Oxidizer Stream** node and set the *Temperature of the Stream* using the temperature from STAR-CCM+ v11.02 that is set for the inlet boundary of the fuel/air mixture.

Any FGM Table parameters that were set in the simulation before STAR-CCM+ v11.04, must be redefined.

2. Create an FGM table in DARS and construct the FGM table in STAR-CCM+. It is not possible to use a Premixed FGM table in STAR-CCM+ v.11.04 or later. See FGM Table.

   When creating the FGM table in DARS:
   - Set the Fuel Composition to the premixed mixture and the oxidizer to air.
   - Set the Fuel Temperature to the temperature that is set for the inlet boundary of the fuel/air mixture in STAR-CCM+ v11.02.
   - Set the Pressure to the absolute pressure that is specified in STAR-CCM+.
3. Select the **[Physics]** > **Initial Conditions** > **Mixture Fraction Profile** node and specify the *Value* as `1.0`.
4. Select the **Models** > **Flamelet Generated Manifold** > **Progress Variable Variance** node and set the *Method* that you require.
5. To restore the Ignitor:
   a) Right-click the **[Continuum]** > **Ignitors** node and select **New**.
   b) Select the **ProgressVariableIgnitor 1** node and specify the necessary properties.
   c) If the **ProgressVariableIgnitor 1** node *Activator* is set to **Pulse**, select the **Pulse** sub-node and specify the necessary properties.
6. For all inlet and outlet boundaries within each physics continuum, select the **[Boundary]** > **Physics Values** > **Mixture Fraction Profile** node and specify the *Value* as `1.0`.

Coherent Flame Model Reference

The Coherent Flame Model is a flame positioning model which you can use with any of the flamelet combustion models—Chemical Equilibrium, Steady Laminar Flamelet, or Flamelet Generated Manifold.

In this model, a flame area density and a progress variable on the grid are tracked through transport equations. The mean species concentrations come from the underlying combustion model—Chemical Equilibrium, Steady Laminar Flamelet, or Flamelet Generated Manifold. A mean enthalpy transport equation is also solved. The mean temperature, density, and viscosity are then calculated knowing the mean enthalpy and species concentrations.

Coherent Flame Model Reference

Theory	See Flame Propagation.	
Provided By	• When using the FGM model: **[physics continuum]** > **Models** > *Progress Variable Source* • When using the Chemical Equilibrium or SLF model: **[physics continuum]** > **Models** > *Flame Propagation*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Coherent Flame Model (CFM)**	
Requires	*Material*: **Multi-Component Gas** *Reaction Regime*: **Reacting** *Reacting Flow Models*: **Flamelet** then either: • *Flamelet Models*: **Flamelet Generated Manifold (FGM)** • *Flamelet Models*: **Chemical Equilibrium** or **Steady Laminar Flamelet** *Flame Type*: **Partially-Premixed Flame**	
Properties	Key properties are the coefficients α, β, a, and B. See Coherent Flame Model (CFM) Properties.	
Activates	Initial Conditions	Flame Area Density Progress Variable See Initial Conditions.
	Boundary Inputs	Flame Area Density Progress Variable See Boundary Settings.
	Region Settings	Active Reactions Option. See Region Settings.
	Other Continuum Nodes	The **Ignitors** node provides the right-click option to create a **Flame Area Density Ignitor**. See Ignitors.
	Solvers	CFM Combustion Solver. See CFM Combustion Solver Properties.
	Monitors	**Fad**: Flame Area Density **ProgVarCFM:** Progress Variable CFM
	Field Functions	Flame Area Density Progress Variable Progress Variable Variance Unnormalized Progress Variable CFM Unnormalized Progress Variable Variance See Field Functions.

Coherent Flame Model (CFM) Properties

CFM Constant, Alpha

Coefficient α, from Eqn. (3426).

CFM Constant, Beta

Coefficient β, from Eqn. (3426).

CFM Constant, A

Coefficient a, from Eqn. (3426).

CFM Constant, B

Coefficient B, from Eqn. (3433).

Laminar Flame Speed Control

- **Gulder Laminar Flame Speed**
  Selects the Gulder method. Activates the **Gulder Laminar Flame Speed** node.
- **Metghalchi Laminar Flame Speed**
  Selects the Metghalchi method. Activates the **Metghalchi Laminar Flame Speed** node.
- **User Defined Laminar Flame Speed**
  Selects the User-Defined method. Activates the **User Defined Laminar Flame Speed** node.

Secondary Gradients

There are two sources of secondary gradients in STAR-CCM+ flow solvers:

- boundary secondary gradients for diffusion
- interior secondary gradients at cell faces

Use this property to control which gradients are included in the solver. **On** gives both gradients while **Off** excludes them. **Interior Only** and **Boundaries Only** select the corresponding gradients.

Convection

Sets the discretization scheme that STAR-CCM+ uses for computing the convection flux on a cell face in appropriate transport equations. More information is given in the related topic for the Convection Term:

- **1st-Order**: First-order upwind scheme. This scheme scales the transported quantity by the upstream or downstream mass flowrate depending on flow direction. Only use when a higher-order scheme fails to give convergence, or in order to obtain an initial solution before switching to a higher-order scheme.
- **2nd-Order**: Second-order upwind scheme. This scheme introduces linear interpolation of cell values on either side of the upstream or downstream face. Using this scheme can lead to poorer convergence properties, but gives accuracy as good as or better than the first-order scheme.

Initial Conditions

Flame Area Density

Progress Variable

Scalar profile values.

Boundary Settings

Flow Boundaries

Flow boundaries except the Outlet have both of the following physics values available by default.

Flame Area Density

Progress Variable

Scalar profile values.

Wall Boundary

Wall Combustion Scalar Option

Wall Combustion Scalar	Corresponding Physics Value Nodes
Zero Flux	None
Specified Flux	Mixture fraction values
Specified Value	Both of the above-listed physics values in addition to the mixture fraction values

Region Settings

Applies to any region:

Active Reactions Option

Allows you to activate or deactivate chemical reactions in this region.

CFM Combustion Solver Properties

Under-Relaxation Factor

In order to promote convergence, this property is used to under-relax changes of the solution during the iterative process. If residuals show solution divergence and do not decrease, reduce the under-relaxation factor for the relevant solvers. The default value is 0.9.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Field Functions

Flame Area Density

Progress Variable

Progress Variable Variance

Unnormalized Progress Variable CFM

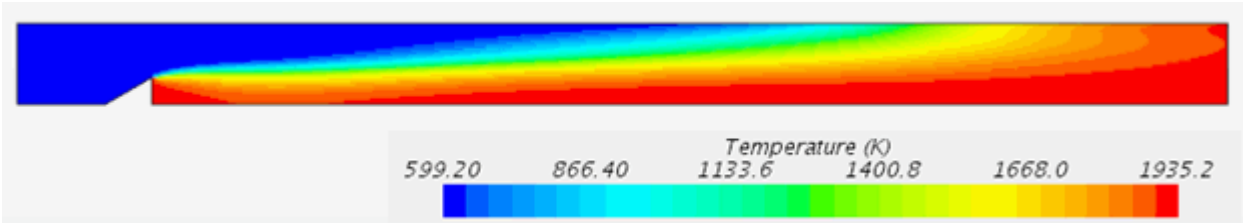
Unnormalized Progress Variable Variance

Restoring the Premixed Coherent Flame Model

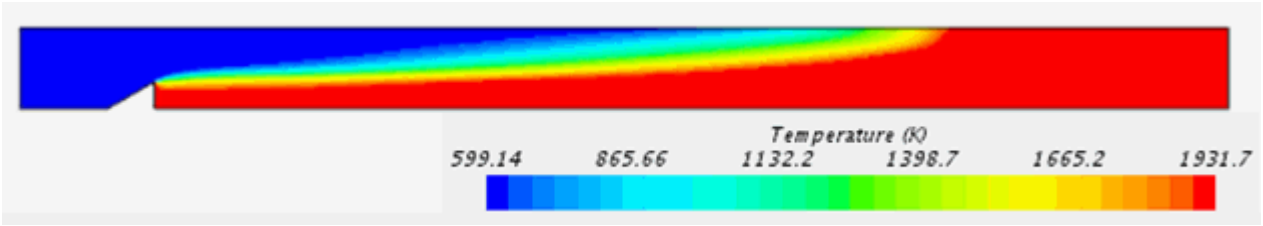
The Premixed Coherent Flame Model (CFM) in STAR-CCM+ v.11.06 is restored to the Flamelet Generated Manifold (FGM) model with the Coherent Flame Model (CFM) flame propagation model. You can restore the Premixed model settings as described in this section.

As part of a plan to improve and simplify the procedure for selecting combustion models, the Premixed CFM model is discontinued in STAR-CCM+ v.11.06.

When loading a simulation in STAR-CCM+ v.11.06 or later that was saved in a version of STAR-CCM+ before v11.06, the Premixed CFM model is restored to the Flamelet Generated Manifold (FGM) model with the Coherent Flame Model (CFM) flame propagation model. Several settings are defined differently and must be set up again. The following images show a comparison of the temperatures on the cross section of a flame holder.

Premixed CFM 11.04


Premixed FGM 11.06 (Partially Pre-Mixed)



STAR-CCM+ v.11.06 provides the FGM model as an option for Premixed Combustion. However, this model is the same as that provided for Partially-Premixed combustion. Further steps are required to restore the premixed behaviour that was available in STAR-CCM+ v.11.04 and earlier. These steps are as follows:

You can restore the settings of the deprecated Premixed combustion model, using the following procedure:

1. Redefine the reacting fluid stream to match the Premixed set-up.
   a) Select the **[Continuum]** > **FGM Table Generator** > **Fluid Streams** node and set the *Fluid Stream Components* as the Premixed mixture.
   b) Select the **Fluid Streams** node and set the *Temperature of the Stream* using the temperature from STAR-CCM+ v11.04 that is set for the inlet boundary of the fuel/air mixture.
   c) Select the **[Continuum]** > **FGM Table Generator** > **Fluid Streams** > **Oxidizer Stream** node and set the *Fluid Stream Components* as Air.
   d) Select the **Oxidizer Stream** node and set the *Temperature of the Stream* using the temperature from STAR-CCM+ v11.04 that is set for the inlet boundary of the fuel/air mixture.

Any FGM Table parameters that were set in the simulation before STAR-CCM+ v11.06, must be redefined.

2. Create an FGM table using **[Continuum]** > **FGM Table Generator**.
   - Set the Fuel Composition to the premixed mixture and the oxidizer to air.
   - Set the Fuel Temperature to the temperature that is set for the inlet boundary of the fuel/air mixture in STAR-CCM+ v11.04.
   - Set the Pressure to the absolute pressure that is specified in STAR-CCM+.
3. Select the **[Physics]** > **Initial Conditions** > **Mixture Fraction Profile** node and specify the *Value* as `1.0`.
4. Select the **Models** > **Flamelet Generated Manifold** > **Progress Variable Variance** node and set the *Method* that you require.
5. To restore the Ignitor:
   a) Right-click the **[Continuum]** > **Ignitors** node and select **New**.
   b) Select the **ProgressVariableIgnitor 1** node and specify the necessary properties.
   c) If the **ProgressVariableIgnitor 1** node *Activator* is set to **Pulse**, select the **Pulse** sub-node and specify the necessary properties.
6. For all inlet and outlet boundaries within each physics continuum, select the **[Boundary]** > **Physics Values** > **Mixture Fraction Profile** node and specify the *Value* as `1.0`.

Turbulent Flame-Speed Closure Model Reference

The Turbulent Flame Speed Closure (TFC) model is a flame positioning model which you can use with any of the flamelet combustion models—Chemical Equilibrium, Steady Laminar Flamelet, or Flamelet Generated Manifold.

This model provides a choice of correlations for turbulent flame speed:

- Zimont, the most widely used option for TFC type combustion simulations
- Peters, commonly for internal combustion engines
- User Defined, a scalar profile

Turbulent Flame-Speed Closure Model Reference

Theory	See Turbulent Flame Speed.	
Provided By	• When using the FGM model: **[physics continuum]** > **Models** > *Progress Variable Source* • When using the Chemical Equilibrium or SLF model: **[physics continuum]** > **Models** > *Flame Propagation*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Turbulent Flame Speed Closure (TFC)**	
Requires	*Material*: **Multi-Component Gas** *Reaction Regime*: **Reacting** *Reacting Flow Models*: **Flamelet** then either: • *Flamelet Models*: **Flamelet Generated Manifold (FGM)** • *Flamelet Models*: **Chemical Equilibrium** or **Steady Laminar Flamelet** *Flame Type*: **Partially-Premixed Flame**	
Properties	Key properties are: Turbulent Flame Speed Control. See Turbulent Flame-Speed Closure Model Properties.	
Activates	Model Controls (child nodes)	The TFC model contains sub-nodes for a rate coefficient, turbulent flame speed options, and laminar flame properties. See the following: • TFC Rate Coefficient, A • Zimont Turbulent Flame Speed • Peters Turbulent Flame Speed • User Defined Turbulent Flame Speed • Laminar Flame Properties
	Initial Conditions	Progress Variable See Initial Conditions.
	Boundary Inputs	Progress Variable See Boundary Settings.

	Region Settings	Active Reactions Option. See Region Settings.
	Other Continuum Nodes	The **Ignitors** node provides the right-click option to create a **Progress Variable Ignitor**. See Ignitors.
	Solvers	TFC Combustion Solver. See TFC Combustion Solver Properties.
	Monitors	**ProgVarTFC**: Progress Variable TFC
	Field Functions	Progress Variable, Progress Variable Variance, Unnormalized Progress Variable TFC, Unnormalized Progress Variable Variance. See Field Functions.

Turbulent Flame-Speed Closure (TFC) Model Properties

Turbulent Flame Speed Control

- **Zimont**
  Selects the Zimont method. Activates the **Zimont Turbulent Flame Speed** node.
- **Peters**
  Selects the Peters method. Activates the **Peters Turbulent Flame Speed** node.
- **User Defined Profile**
  Selects the User-Defined method. Activates the **User Defined Turbulent Flame Speed** node.

Secondary Gradients

There are two sources of secondary gradients in STAR-CCM+ flow solvers:

- boundary secondary gradients for diffusion
- interior secondary gradients at cell faces

Use this property to control which gradients are included in the solver. **On** gives both gradients while **Off** excludes them. **Interior Only** and **Boundaries Only** select the corresponding gradients.

Convection

Sets the discretization scheme that STAR-CCM+ uses for computing the convection flux on a cell face in appropriate transport equations. More information is given in the related topic for the Convection Term:

- **1st-Order**: First-order upwind scheme. This scheme scales the transported quantity by the upstream or downstream mass flowrate depending on flow direction. Only use when a higher-order scheme fails to give convergence, or in order to obtain an initial solution before switching to a higher-order scheme.
- **2nd-Order**: Second-order upwind scheme. This scheme introduces linear interpolation of cell values on either side of the upstream or downstream face. Using this scheme can lead to poorer convergence properties, but gives accuracy as good as or better than the first-order scheme.

Model Controls

TFC Rate Coefficient, A

Coefficient from Eqn. (3420). Scalar profile value.

Zimont Turbulent Flame Speed

Zimont Turbulent Flame Speed Properties

Wall Effect Constant

Uses wall effects to model the quenching of the flame at walls.

Select a setting from `0` (fully extinguished) to `1` (no effect).

See Turbulent Flame Speed.

Unburnt Thermal Diffusivity Control

- **Laminar Flame Thickness Option**
  Uses the laminar flame calculation technique (see Laminar Flame Thickness). Activates the **Laminar Flame Thickness Option** node.
- **User Defined Profile**
  Allows you to set this diffusivity as a scalar profile. Activates the **User Defined Unburnt Thermal Diffusivity** node.

Flame Stretch Effect

When activated, uses the flame stretch factor G (Eqn. (3437)), which takes the flame stretch effect into account by representing the probability of unquenched flamelets. Activates the **Flame Stretch Effect** node.

Zimont Turbulent Flame Speed Sub-Nodes

Laminar Flame Thickness Option

This node serves as a placeholder to indicate that the Unburnt Thermal Diffusivity Control is using the laminar flame calculation technique.

User Defined Unburnt Thermal Diffusivity

Use the **Unburnt Thermal Diffusivity Profile** sub-node of this node to set this diffusivity as a scalar profile.

Flame Stretch Effect

Flame Stretch Effect Properties

Constant, ustr

Coefficient μ_{str} from Eqn. (3438).

Critical Strain Rate Control

- **Chemical Time Scale Method**
  Activates the **Chemical Time Scale Method** node.

- **User Defined Profile**
  Activates the **User Defined Critical Strain Rate** node.

Flame Stretch Effect Sub-Nodes

Chemical Time Scale Method

Constant B from Eqn. (3440).

User Defined Critical Strain Rate

The **Critical Strain Rate Profile** sub-node of this node functions as a scalar profile.

Peters Turbulent Flame Speed

Peters Turbulent Flame Speed Properties

Wall Effect Constant

Uses wall effects to model the quenching of the flame at walls.

Select a setting from `0` (fully extinguished) to `1` (no effect).

Constant, A1

Coefficient A_1, from Eqn. (3442)

Constant, A4

Coefficient A_4, from Eqn. (3442)

Constant, B1

Coefficient B_1, from Eqn. (3442)

Constant, B3

Coefficient B_3, from Eqn. (3442)

Ewald's Corrector Constant

c_{ew}, from Eqn. (3442)

User Defined Turbulent Flame Speed

User Defined Turbulent Flame Speed Properties

Wall Effect Constant

Uses wall effects to model the quenching of the flame at walls.

Select a setting from `0` (fully extinguished) to `1` (no effect).

User Defined Turbulent Flame Speed Sub-Nodes

Turbulent Flame Speed Profile

Scalar profile value.

Laminar Flame Properties

Laminar Flame Speed

The **Laminar Flame Speed** node provides options for controlling flame speed in its properties.

Method

- **Gulder Laminar Flame Speed**
  Selects the Gulder method. Activates the **Gulder Laminar Flame Speed** node.
- **Metghalchi Laminar Flame Speed**
  Selects the Metghalchi method. Activates the **Metghalchi Laminar Flame Speed** node.
- **User Defined Laminar Flame Speed**
  Selects the User-Defined method. Activates the **User Defined Laminar Flame Speed** node.

Laminar Flame Thickness

The **Laminar Flame Thickness** node provides options for controlling flame thickness in its properties.

Method	Corresponding Method Node
Laminar Flame Speed Relation	**Laminar Flame Speed Relation** Exposes the following terms: • *Laminar Prandtl Number*, Prandtl number P_r, from Eqn. (3381) • *Constant mu1*, coefficient μ_1, from Eqn. (3382) • *Constant mu2*, coefficient μ_2, from Eqn. (3382)
User Defined Laminar Flame Thickness	**User Defined Laminar Flame Thickness** Sets the laminar flame thickness using its **Laminar Flame Thickness Profile** sub-node. This sub-node functions in a manner similar to a scalar profile, but it also includes a table with laminar flame thickness as a function of equivalence ratio.

Initial Conditions

Progress Variable

Scalar profile value.

Boundary Settings

Flow Boundaries

Flow boundaries except the Outlet have the **Progress Variable** physics value, a scalar profile, available by default.

Wall Boundary

Wall Combustion Scalar Option

Wall Combustion Scalar	Corresponding Physics Value Nodes
Zero Flux	None
Specified Flux	Mixture fraction values
Specified Value	**Progress Variable**, a scalar profile, in addition to the mixture fraction values

Region Settings

Applies to all regions:

Active Reactions Option

Allows you to activate or deactivate chemical reactions in this region.

TFC Combustion Solver Properties

Under-Relaxation Factor

In order to promote convergence, this property is used to under-relax changes of the solution during the iterative process. If residuals show solution divergence and do not decrease, reduce the under-relaxation factor for the relevant solvers. The default value is 0.8.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Field Functions

Progress Variable

Progress Variable Variance

Unnormalized Progress Variable CFM

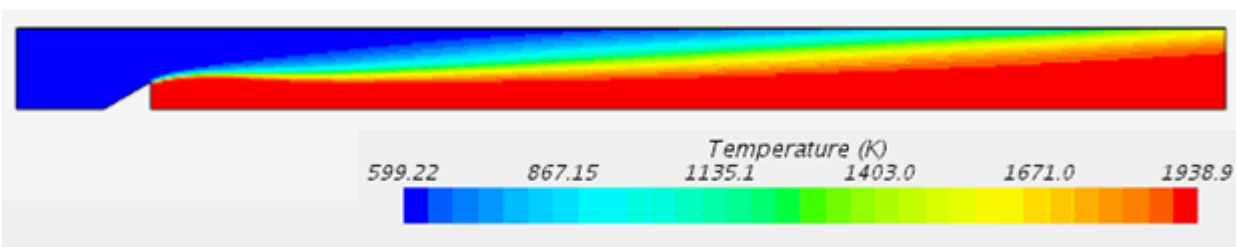
Unnormalized Progress Variable Variance

Restoring the Premixed Turbulent Flame-Speed Closure Model

The Premixed Turbulent Flame-Speed Closure (TFC) model in STAR-CCM+ v.11.06 is restored to the Flamelet Generated Manifold (FGM) model with the Turbulent Flame-Speed Closure (TFC) flame propagation model. You can restore the Premixed model settings as described in this section.

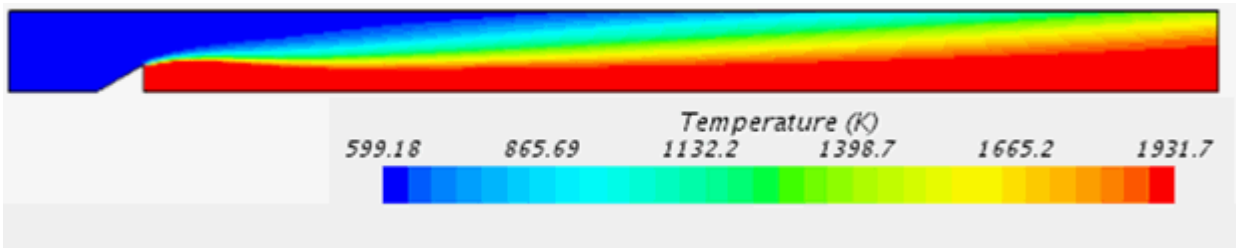
As part of a plan to improve and simplify the procedure for selecting combustion models, the Premixed TFC model is discontinued in STAR-CCM+ v.11.06.

When loading a simulation in STAR-CCM+ v.11.06 or later that was saved in a version of STAR-CCM+ before v11.06, the Premixed TFC model is restored to the Flamelet Generated Manifold (FGM) model with the Turbulent Flame-Speed Closure (TFC) flame propagation model. Several settings are defined differently and must be set up again. The following images show a comparison of the temperatures on the cross section of a flame holder.

Premixed TFC 11.04


Premixed FGM 11.06 (Partially Pre-Mixed)

Premixed FGM 11.06 (Partially Pre-Mixed)



STAR-CCM+ v.11.06 provides the FGM model as an option for Premixed Combustion. This model is the same as that provided for Partially-Premixed combustion. Further steps are required to restore the premixed behaviour that was available in STAR-CCM+ v.11.04 and earlier. These steps are as follows:

You can restore the settings of the deprecated Premixed combustion model, using the following procedure:

1. Redefine the reacting fluid stream to match the Premixed set-up.
   a) Select the **[Continuum]** > **FGM Table Generator** > **Fluid Streams** node and set the *Fluid Stream Components* as the Premixed mixture.
   b) Select the **Fluid Streams** node and set the *Temperature of the Stream* using the temperature from STAR-CCM+ v11.04 that is set for the inlet boundary of the fuel/air mixture.
   c) Select the **[Continuum]** > **FGM Table Generator** > **Fluid Streams** > **Oxidizer Stream** node and set the *Fluid Stream Components* as Air.
   d) Select the **Oxidizer Stream** node and set the *Temperature of the Stream* using the temperature from STAR-CCM+ v11.04 that is set for the inlet boundary of the fuel/air mixture.

Any FGM Table parameters that were set in the simulation before STAR-CCM+ v11.06, must be redefined.

2. Create an FGM table using the **[Continuum]** > **FGM Table Generator**.
   - Set the Fuel Composition to the premixed mixture and the oxidizer to air.
   - Set the Fuel Temperature to the temperature that is set for the inlet boundary of the fuel/air mixture in STAR-CCM+ v11.04.
   - Set the Pressure to the absolute pressure that is specified in STAR-CCM+.
3. Select the **[Physics]** > **Initial Conditions** > **Mixture Fraction Profile** node and specify the *Value* as `1.0`.
4. Select the **Models** > **Flamelet Generated Manifold** > **Progress Variable Variance** node and set the *Method* that you require.
5. To restore the Ignitor:
   a) Right-click the **[Continuum]** > **Ignitors** node and select **New**.
   b) Select the **ProgressVariableIgnitor 1** node and specify the necessary properties.
   c) If the **ProgressVariableIgnitor 1** node *Activator* is set to **Pulse**, select the **Pulse** sub-node and specify the necessary properties.
6. For all inlet and outlet boundaries within each physics continuum, select the **[Boundary]** > **Physics Values** > **Mixture Fraction Profile** node and specify the *Value* as `1.0`.

Emissions

Emissions are unwanted pollutant species that are by-products of reactions. There are two options to model pollutant species in STAR-CCM+. When using the Complex Chemistry model, you can include the pollutant species in the chemical mechanism. For all other models, in particular Flamelet models and the Eddy Break-Up (EBU) model, separate transport equations for the pollutant species are solved.

STAR-CCM+ has the following pollutant models:

- NOx Emission
  - NOx Fuel
  - NOx Thermal
    - NOx Zeldovich
  - NOx Prompt
- Soot Emissions
  - Soot Moments
  - Soot Two-Equation

NOx Emission Model

Although oxides of nitrogen (NOx) are created naturally from atmospheric nitrogen during thunderstorms, transportation and industries that burn fossil fuels create more significant NOx emissions. NOx emissions contribute significantly to the creation of acid rain. It is therefore important to develop technologies which produce lower levels of NOx emissions wherever possible.

When using any of the reacting flow models, you can activate the optional NOx Emission model. This model provides a tool to predict the NOx production from various sources and helps in the design of NOx control measures.

The NOx Emission model solves the transport equation for the concentration of NOx, using the Fuel NOx model and/or the NOx Thermal model to contribute to the NOx reactive source term. You can also additionally use the NOx Prompt model.

- The NOx Fuel model contributes the NOx emission arising from fuels such as heavy liquids and coal which release fuel-bound nitrogen during combustion. Therefore, when modeling emissions that are resultant from burning these fuels, use the Fuel NOx model in addition to the Thermal NOx model. In Lagrangian phases, you can use the Fuel NOx model for liquid fuel or coal. In addition to liquid fuels and coal, STAR-CCM+ allows you to model Fuel NOx for a purely gaseous fuel. For pure gas fuel, you can introduce the intermediate species, HCN and/or NH3, arising from the fuel directly at the inlet boundary.
- The NOx Thermal model contributes the NOx emission that is formed from N2 and O2 or OH under lean fuel conditions and at high temperatures, using the three-step extended Zeldovich reaction mechanism. Thermal NOx is dominant in hydrocarbon flames. For combustion set-ups that do not use the Complex Chemistry model, the concentrations of species are computed during the flamelet table generation and are taken directly from the flamelet table for each iteration. When using the Complex Chemistry model, the source terms are computed at each iteration.
- The NOx Prompt model contributes the NOx emission arising from processes that are not accounted for by the NOx Fuel or NOx Thermal models. For example, atmospheric nitrogen reacting with intermediate

species of combustion such as C, CH, and CH2 radicals. Such contributions are generated during low-temperature combustion.

NOx is considered to be a passive scalar, not influencing density calculations. Therefore, in steady-state simulations, use the NOx Emission model as a post-processing tool, that is, activate it after the flow field has converged. In transient simulations, to model the evolution of NOx accurately, activate this model from the beginning.

Soot Emissions Model

The formation and emission of carbonaceous particles is often observed during the combustion of hydrocarbons. These particulates, called soot, are identified in flames and fires as yellow luminescence. In gas turbines, internal combustion engines and other practical combustion devices, the formation of soot is mostly a product of incomplete combustion.

Apart from the resulting loss in combustion efficiency, a serious effect of soot formation is the health hazard. On the other hand, there are situations where the presence or generation of soot is required. For example, generation of carbon black is needed in the production of automobile tires. In furnaces for industrial application or in heat generators, the intermediate formation of soot is required to augment the heat transfer by radiation. The soot, however, must undergo oxidation before the exhaust is released into the environment.

It is widely accepted that the formation of soot is a complex process which consists of the following:

- Fuel pyrolysis and oxidation reactions
- Formation of polycyclic aromatic hydrocarbons (PAH)
- Inception of first particles
- Growth of soot particles due to reaction with gas-phase species
- Coagulation of particles
- Oxidation of soot particles and intermediates

Soot Moments Model

The method of moments is based on the fact that solving an infinite set of equations for the statistical moments of the PSDF is equivalent to the direct simulation of the PSDF. Additionally, it can be shown that often the accuracy using only a few moments is sufficiently high. Usually, a set of equations for the first few moments is applied, where the accuracy of the approach increases with the number of moments that are used. The main advantages are:

- Computational efficiency
- The ability to extract major features of the PSDF from the moments, such as:
  - Soot volume fraction
  - Mean number density
  - Total surface
  - Mean diameter

  The Soot Moments model in STAR-CCM+ can solve up to four moments: Moment 0, 1, 2, and 3. Soot is treated as a passive scalar. The model is compatible with the Complex Chemistry and Steady Laminar Flamelet (SLF) combustion models—it is not currently compatible with the Chemical Equilibrium and Flamelet Generated Manifold (FGM) combustion models.

The **Soot Moments** node in the continuum object tree represents this type of modeling. To access the properties of the **Soot Moments** model, select the **Soot Moments** node.

Soot Two-Equation Model

STAR-CCM+ provides a **Soot Two-Equation** model that is based on a technique that is called the Moss-Brookes-Hall (MBH) soot model.

The Soot Two-Equation model is a semi-empirical soot model that is based on four physical processes of soot formation:

- Nucleation / Inception
- Coagulation
- Surface growth
- Oxidation

The Soot Two-Equation model allows you to implement any two-equation soot model by setting the built-in source terms to zero, and then specifying new source terms for each of the four physical processes of soot formation.

When you activate the Soot Two-Equation model with the Eddy Break-up (EBU) model, the fluid stream manager becomes available. The fluid stream manager lets you define the fuel and oxidizer streams.

In STAR-CCM+ the Soot Two-Equation model offers a choice of nucleation options in its properties: Acetylene-based and PAH-based.

To access the properties of the **Soot Two-Equation** model, select the **Soot Two-Equation** node.

Contents:

Emissions Workflow
Emissions Reference

Emissions Workflow

Follow the steps in this workflow to simulate NOx and/or Soot emissions.

Before following this workflow, make sure that you have completed the model selection steps from the Reacting Flow General Workflow, and one of the following specific detailed workflows:

- Flamelet Workflow
- Reacting Species Transport Workflow
- Surface Chemistry Workflow

This emissions workflow is not suitable for use with the Interphase workflow. The Interphase workflow describes how to use emissions models that are specific to the Lagrangian phase.

The NOx Emission model requires species and enthalpy models to be activated, so a flow model must be selected unless one is already selected with the combustion model. **Segregated Flow** is the recommended choice when using the NOx Emission model.

1. For the physics continuum that represents the reacting flow, select the following models—in addition to the models that are previously selected, with **Auto-Select recommended models** activated:

Group Box	Model
Optional Models	You can select one or both of these models. • **NOx Emission**: provides a framework for modeling the NOx transport equation by providing a tool to predict the NOx production from various sources. • **Soot Emissions**
Specific NOx Models (available when NOx Emission is activated)	You can select one or more of the NOx models to calculate the reactive source term in the NOx transport equation. • **NOx Fuel**: contributes the NOx emission that arise from the fuel portion. You can use this model for a pure gas-phase combustion. The NOx Fuel model transports three passive scalars: NO, NH3, and HCN. When this model is selected, HCN and NH3 boundary conditions are available for you to specify the inlet composition. • **NOx Thermal**: contributes to the source term by using the three-step extended Zeldovich reaction mechanism. • **NOx Prompt**: contributes the NOx emission that arise from the reaction of hydrocarbon fragments and molecular nitrogen. Often produced in combustion environments with relatively low-temperature and fuel-rich conditions—as in staged combustion systems and gas turbines.
Thermal NOx Models (available when NOx Thermal is activated)	The **NOx Zeldovich** model, which is the only Thermal NOx model that is available, is selected automatically when NOx Thermal is selected. For combustion set-ups that do not use the Complex Chemistry model, the NOx Zeldovich model uses the coefficients—A, B, C, and D, in Eqn. (3483)—that are stored in the flamelet table, to compute the averaged reactive source term in the NOx transport equation. When using the Complex Chemistry model, the Complex Chemistry model computes the source terms at each iteration.
Soot Emissions Model (available when Soot Emissions is activated)	You can select only one Soot model. • **Soot Moments**: uses the Method of Moments technique to calculate the soot particle size distribution function (PSDF). Can solve up to four moments: Moment 0, 1, 2, and 3. Soot is treated as a passive scalar. (Available for all Non-Premixed combustion models). • **Soot Two-Equation**: solves transport equations for scaled number density and scaled mass density based on the Moss-Brookes-Hall (MBH) technique.

2. Return to the specific detailed workflow that you are following.

Emissions Reference

Most of the controls that you can set for emissions modeling are located in the physics continuum or within the Solvers node. Some models require that you set mass fractions on inlet boundaries.

Contents:

NOx Emission Model Reference

NOx Fuel Model Reference
NOx Prompt Model Reference
NOx Thermal Model Reference
NOx Zeldovich Model Reference
Soot Emissions Model Reference
Soot Moments Model Reference
Restoring the 11.06 Soot Moments Model
Soot Two-Equation Model Reference

NOx Emission Model Reference

The NOx Emission model provides a framework for modeling the NOx transport equation.

NOx Emission Model Reference

Theory	See NOx.	
Provided By	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **NOx Emission**	
Requires	*Material*: **Multi-Component Gas** *Reaction Regime*: **Reacting** *Reacting Flow Models*: **Reacting Species Transport** or **Flamelet**	
Properties	Key properties are: Source Enabled Trigger, Begin, Secondary Gradients, Convection. See NOx Emission Model Properties.	
Activates	Initial Conditions	Nitrogen Oxide. See Initial Conditions.
	Boundary Inputs	Wall Combustion Scalar Option. See Boundary Settings.
	Solvers	NOx. See NOx Emission Solvers.
	Monitors	NO
	Field Functions	Mass Fraction of Nitrogen Oxide Emission. See Field Functions.

NOx Emission Model Properties

Source Enabled Trigger

Allows you to deactivate chemistry at the beginning of the calculation—to let the flow field converge before starting to calculate the chemistry. You can choose when the chemistry is calculated, according to the following options:

- **Never**: The Source Enabled Trigger is deactivated, so chemical source terms are calculated from the first iteration/timestep.

- **Iteration**: The Source Enabled Trigger is activated, so the chemical source terms are activated and are calculated after the iteration that is specified as value of the *Begin* property. This option is available in steady-state cases.
- **Timestep**: The Source Enabled Trigger is activated, so the chemical source terms are activated and are calculated after the timestep that is specified as value of the *Begin* property. This option is available in unsteady-state cases.

Begin

Specifies the iteration or time-step after which the reactions are activated. During previous iterations or time-steps, reactions are deactivated. This feature is useful when modeling significant flow as you can deactivate reactions until a flow solution is found.

Secondary Gradients

Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces.

- **On**: Default value. Solves for interior and boundary types of secondary gradient.
- **Off**: Does not solve for either type of secondary gradient.
- **Interior Only**: Solves for the interior secondary gradients only.
- **Boundaries Only**: Solves for the boundary secondary gradients only.

Convection

In transport equations, you can choose from a range of schemes that calculate the convection term at a cell face. This calculation requires STAR-CCM+ to compute the face value of a quantity from the surrounding cell values. The method used for computing this face value has a profound effect on the stability and accuracy of the numerical scheme. For guidance on selecting a convection scheme, see Convective Flux.

- **1st-order**: First-order convection scheme.
- **2nd-order**: Second-order convection scheme.

Initial Conditions

Nitrogen Oxide

Allows you to set the profile for the mass fraction of the Nitrogen Oxide emission. For example, when simulating exhaust gas recirculation (EGR).

Boundary Settings

Any Boundary Type

Wall Combustion Scalar Option

Selects the scalars for the wall combustion calculation.

Method	Corresponding Physics Value Nodes
Zero Flux	None
Specified Value Allows you to specify the composition of species at a boundary.	**Nitrogen Oxide**

NOx Solver Properties

Under-Relaxation Factor

In order to promote convergence, this property is used to under-relax changes of the solution during the iterative process. If residuals show solution divergence and do not decrease, reduce the under-relaxation factor for the relevant solvers.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Field Functions

Mass Fraction of Nitrogen Oxide Emission

Represents the mass fraction of NO. Since it is considered that NOx emissions consist only of NO, this is the transported scalar of the NOx Emission model.

NOx Fuel Model Reference

The Fuel NOx Model contributes the NOx emission arising from the fuel portion.

The Fuel NOx model takes care of the reactions to produce NOx.

NOx Fuel Model Reference

Theory	See Fuel NOx	
Provided By	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **NOx Fuel**	
Requires	*Material*: **Multi-Component Gas** *Reaction Regime*: **Reacting** *Reacting Flow Models*: **Reacting Species Transport** or **Flamelet** *Optional Models*: **NOx Emission**	
Properties	No properties	
Activates	Initial Conditions	Ammonia, Hydrogen Cyanide. See Initial Conditions.
	Boundary Inputs	Wall Combustion Scalar Option. See Boundary Settings.
	Monitors	HCN, NH3
	Field Functions	Ammonia, Hydrogen Cyanide. See Field Functions.

Initial Conditions

Ammonia

Hydrogen Cyanide

Boundary Settings

Any Boundary Type

Wall Combustion Scalar Option

Selects the scalars for the wall combustion calculation.

Method	Corresponding Physics Value Nodes
Zero Flux	None
Specified Value Allows you to specify the composition of species at a boundary.	**Ammonia** **Hydrogen Cyanide**

Field Functions

Ammonia

Hydrogen Cyanide

NOx Prompt Model Reference

If the conditions are fuel-rich conditions in which you suspect that there are hydrocarbon fragments, then prompt NOx is important. In cases such as gas turbines and staged combustion, you might want to include the NOx Prompt model to get more accurate NOx predictions.

NOx Prompt Model Reference

Theory	See NOx Prompt.
Provided By	**[physics continuum]** > **Models** > *Optional Models*
Example Node Path	**Continua** > **Physics 1** > **Models** > **NOx Prompt**
Requires	*Material*: **Multi-Component Gas** *Reaction Regime*: **Reacting** *Reacting Flow Models*: **Reacting Species Transport** or **Flamelet** *Optional Models*: **NOx Emission**
Properties	Key properties are: Fuel Component, Pre-Exponent Correction Factor, Activation Energy Ea. See NOx Prompt Model Properties.

NOx Prompt Model Properties

Fuel Component

Selects a component from a drop-down list of materials added to the continuum.

Pre-exponent Correction Factor

If you have the kinetics data of a particular hydrocarbon, you can adjust this parameter for it, which is f in Eqn. (3486).

Activation Energy, Ea

E_A in Eqn. (3486).

NOx Thermal Model Reference

The NOx Thermal Model contributes to the thermal part of the NOx source term using a variant of the extended three-step Zeldovich mechanism.

NOx Thermal Model Reference

Theory	See NOx Thermal.	
Provided By	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **NOx Thermal**	
Requires	*Material*: **Multi-Component Gas** *Reaction Regime*: **Reacting** *Reacting Flow Models*: **Reacting Species Transport** or **Flamelet** *Optional Models*: **NOx Emission**	
Properties	None	
Activates	Physics Models	**NOx Zeldovich**. See NOx Zeldovich Model Reference.

NOx Zeldovich Model Reference

The NOx Zeldovich model is used for computation of the instantaneous reactive source term that is needed for assembling the NOx transport equation.

NOx Zeldovich Model Reference

Theory	See NOx Thermal.	
Provided By	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **NOx Thermal** > **NOx Zeldovich**	
Requires	*Material*: **Multi-Component Gas** *Reaction Regime*: **Reacting** *Reacting Flow Models*: **Reacting Species Transport** or **Flamelet** *Optional Models*: **NOx Emission** *Optional Models*: **NOx Thermal**	
Activates	Model Controls (child nodes)	O Concentration, OH Concentration
	Field Functions	Molar Concentration (O), Molar Concentration (OH). See Field Functions.

Field Functions

Molar Concentration (O)

The molar concentration of oxygen.

Molar Concentration (OH)

The molar concentration of the OH group.

O Concentration

Selects the approximation model for the concentration of oxygen.

Equilibrium

Selects the approach that is used in Eqn. (3474) and Eqn. (3475)].

Partial Equilibrium

Selects the approach that is used in Eqn. (3474) and Eqn. (3476) which considers a third-body reaction.

Species List

This option is recommended if the O species is present in the gas-phase reaction mechanism.

OH Concentration

Selects the approximation model for the concentration of the OH group.

Exclusion

Selects the approach that is used in Eqn. (3477), in which the third reaction in the extended Zeldovich mechanism is assumed to be negligible.

Partial Equilibrium

Selects the approach that is used in Eqn. (3478).

Species List

This option is recommended if the OH species is present in the gas-phase reaction mechanism.

Soot Emissions Model Reference

There are two soot emission models available to solve combustion problems, the Soot Moments model and the Soot Two-Equation model.

The soot emissions models influence the Participating Media Radiation (DOM) and Gray Thermal Radiation models by contributing to the absorption coefficient of the continuous phase (the absorption coefficient describing both absorption and emission). STAR-CCM+ models the soot as a gray medium, treating the soot as one component of the continuous phase. The total absorption coefficient of the gas includes the soot absorption coefficient as part of an algebraic sum. You can specify the value of the absorption coefficient as a constant, as a field function, or by the Planck Mean Absorption Coefficient method.

Soot Emission Model Reference

Theory	See Soot.
Provided By	**[physics continuum]** > **Models** > *Optional Models*

Example Node Path	**Continua** > **Physics 1** > **Models** > **Soot Emissions**
Requires	*Material*: **Multi-Component Gas** *Reaction Regime*: **Reacting** *Reacting Flow Models*: **Flamelet** or **Reacting Species Transport** Then either: • *Flamelet Models*: **Chemical Equilibrium**, or **Flamelet Generated Manifold**, or **Steady Laminar Flamelet** • *Reacting Species Models*: **Complex Chemistry** or **Eddy Break-Up** *Flow*: **Segregated Flow**
Properties	None

Soot Moments Model Reference

The Soot Moments Model is based on a technique that is called Method of Moments. This method is used to calculate the soot particle size distribution function (PSDF).

Currently, the nucleation model pathway Eqn. (3539) and Eqn. (3540) is used, which is based on the acetylene concentration.

Soot Moments Model Reference

Theory	See Soot Moments.	
Provided By	**[physics continuum]** > **Models** > *Soot Emissions Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Soot Moments**	
Requires	*Material*: **Multi-Component Gas** *Reaction Regime*: **Reacting** *Reacting Flow Models*: **Flamelet** or **Reacting Species Transport** Then either: • *Flamelet Models*: **Chemical Equilibrium**, or **Flamelet Generated Manifold**, or **Steady Laminar Flamelet** • *Reacting Species Models*: **Complex Chemistry** or **Eddy Break-Up** *Flow*: **Segregated Flow** Then: *Optional Models*: **Soot Emissions**	
Properties	Key properties are: Source Enabled Trigger, Begin, Secondary Gradients, Convection. See Soot Moments Model Properties.	
Activates		
	Model Controls (child nodes)	Soot, Soot Model Properties, PPDF Soot Flamelet Table

	Boundary Inputs	See Boundary Settings.
	Solvers	Soot. See Soot Moments Model Solvers.
	Monitors	SootMoment0, SootMoment1
	Field Functions	Soot Mass Density, Soot Mean Diameter, Soot Moment 0, Soot Moment 1, Soot Moment Source 0, Soot Moment Source 1, Soot Number Density, Soot Surface Density, Soot Volume Fraction. See Field Functions.

Soot Moments Model Properties

Source Enabled Trigger

Allows you to deactivate chemistry at the beginning of the calculation—to let the flow field converge before starting to calculate the chemistry. You can choose when the chemistry is calculated, according to the following options:

- **Never**: The Source Enabled Trigger is deactivated, so chemical source terms are calculated from the first iteration/timestep.
- **Iteration**: The Source Enabled Trigger is activated, so the chemical source terms are activated and are calculated after the iteration that is specified as value of the *Begin* property. This option is available in steady-state cases.
- **Timestep**: The Source Enabled Trigger is activated, so the chemical source terms are activated and are calculated after the timestep that is specified as value of the *Begin* property. This option is available in unsteady-state cases.

Begin

Specifies the iteration or time-step after which the reactions are activated. During previous iterations or time-steps, reactions are deactivated. This feature is useful when modeling significant flow as you can deactivate reactions until a flow solution is found.

Secondary Gradients

Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces.

- **On**: Default value. Solves for interior and boundary types of secondary gradient.
- **Off**: Does not solve for either type of secondary gradient.
- **Interior Only**: Solves for the interior secondary gradients only.
- **Boundaries Only**: Solves for the boundary secondary gradients only.

Convection

In transport equations, you can choose from a range of schemes that calculate the convection term at a cell face. This calculation requires STAR-CCM+ to compute the face value of a quantity from the surrounding cell values. The method used for computing this face value has a profound effect on the stability and accuracy of the numerical scheme. For guidance on selecting a convection scheme, see Convective Flux.

- **1st-order**: First-order convection scheme.

- **2nd-order**: Second-order convection scheme.

Boundary Settings

Inlet or Pressure Outlet Boundary

Soot Moment Profile

Density weighted soot moment profile.

Wall Boundary

Wall Combustion Scalar Option

Sets the scalars for the wall combustion calculation.

Method	Corresponding Physics Value Nodes
Zero Flux	None
Specified Value	**Soot Moment Profile** Density weighted soot moment profile.

Soot Solver Properties

Under-Relaxation Factor

In order to promote convergence, this property is used to under-relax changes of the solution during the iterative process. If residuals show solution divergence and do not decrease, reduce the under-relaxation factor for the relevant solvers.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Field Functions

Soot Mass Density

M in Eqn. (3534). The units are $\left[\mathrm{kg/m^3}\right]$.

Soot Mean Diameter

d in Eqn. (3512).

Soot Moment 0

M_0 in Eqn. (3509).

Soot Moment 1

M_1 in Eqn. (3510) and Eqn. (3511).

Soot Moment Source 0

$r = 0$ in Eqn. (3507).

Soot Moment Source 1

$r = 1$ in Eqn. (3474).

Soot Number Density

N in Eqn. (3509) and Eqn. (3533). The units are $\left[\mathrm{/m^3}\right]$.

Soot Surface Density

S in Eqn. (3513).

Soot Volume Fraction

f_v in Eqn. (3510).

Soot

The Soot sub-node allows you to specify specific material properties for the soot material.

Soot Model Properties

Number of Moments

Selects the number of moments to solve. Possible values range from 2 to 4.

Solving for 4 moments is more computationally expensive than 2. However, if the problem and mesh size are not too large, and you are interested in soot size dispersion, then run 4 moments.

Steric Factor Option

Allows you to specify the steric factor α in Eqn. (3532), that is, the fraction of reactive sites on the surface of the soot particle that are available for soot growth or oxidation reactions.

Method	Corresponding Sub Node
Constant	**Steric Factor** Allows you to define a constant value for Alpha (the steric factor α). The default value is set as 0.7, however, you can specify a value between 0 and 1.
Premixed Temperature Correlation Uses fitted correlation from Appel et al. [671] for α, where α is a function of the local temperature and the average soot particle size, quantified by the reduced soot moment $\mu 1$, see Eqn. (3532).	none
User Defined Profile	**User-Defined Steric Factor** Allows you to define the steric factor using a scalar profile.

Nucleation Option

- *C2H2*: C2H2 (acetylene) is used as a soot precursor for nucleation, see Eqn. (3516).
- *A4 (Pyrene)*: A4 (Pyrene) is used as a soot precursor for nucleation, see Eqn. (3514) and Eqn. (3515).

Surface-growth Scale

Scales surface growth (part of W_r in Eqn. (3508)).

Nucleation Scale

Scales nucleation (R_r in Eqn. (3508)).

Oxidation Scale

Scales oxidation (part of W_r in Eqn. (3508)).

Cold Temperature Limit

If the temperature in the cell is lower than this number, then cold flow is assumed—no soot source calculations are performed in that cell.

Two-Way Coupled Species

When activated, allows two-way coupling between soot-related species.

Restoring the 11.06 Soot Moments Model

In STAR-CCM+ v.12.02, the A3R5 Soot Moments model nucleation option (which was present in STAR-CCM+ v.11.06) is removed for all combustion models. Also, the Soot Moments model is no longer available to use

with the Chemical Equilibrium model. This section provides details of how to restore v.11.06 simulation files that use the Soot Moments model with the Steady Laminar Flamelet (SLF) model.

Soot moments libraries that are generated in DARS are no longer supported in STAR-CCM+ since v.12.02. Instead, the soot moment source terms are calculated in STAR-CCM+, wether you generate a flamelet table in STAR-CCM+ or import a DARS flamelet library.

When loading a sim file from v.11.06 which uses the SLF combustion model with the Soot Moments model, you must regenerate the flamelet table. Make sure that all of the species (H2, H, H2O, OH, O2, C2H2) that are required for soot calculations are present.

Note: The SLF soot moment library that is calculated in STAR-CCM+ v.12.02 does not include a model for condensation. Therefore, soot results will change from v.11.06 to v.12.02.

To restore v.11.06 simulation files that use the Soot Moments model with the SLF model, you can either import a DARS generated SLF library, or generate a SLF library in STAR-CCM+. To generate a SLF library in STAR-CCM+:

1. Open the v.11.06 simulation file in STAR-CCM+ v.12.02 or later.
2. Expand the **Continua** > **[Continuum]** > **SLF Table Generator** node.
3. Make sure that the chemistry definition contains all of the species (H2, H, H2O, OH, O2, C2H2) that are required for soot calculations. If not:
   a) If you do not have a chemical mechanism, or want to use a different chemical mechanism, right-click the **SLF Table Generator** > **Chemistry Definition** node and select **Delete Complex Chemistry Definition**.
   b) Import or create the complex chemistry definition.

      See Importing Species and Reactions.
   c) Redefine any other necessary settings. If necessary, you can compare the previous and current simulation trees to check that the settings are consistent. See Comparing Simulation Trees.
4. Select the **SLF Table Generator** > **Parameters** > **Emissions** node and activate *Soot Moment*.
5. To generate the flamelet library, right-click the **SLF Table** node and select **Generate Flamelet Library and Construct Table**.

Soot Two-Equation Model Reference

The Soot Two-Equation model is based on a technique that is called the Moss-Brookes-Hall (MBH) soot model.

Soot Two-Equation Model Reference

Theory	See Soot Two-Equation.
Provided By	**[physics continuum]** > **Models** > *Soot Emissions Model*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Soot Two-Equation**
Requires	*Material*: **Multi-Component Gas** *Reaction Regime*: **Reacting**

	Reacting Flow Models: **Flamelet** or **Reacting Species Transport** Then either: • *Flamelet Models*: **Chemical Equilibrium**, or **Flamelet Generated Manifold**, or **Steady Laminar Flamelet** • *Reacting Species Models*: **Complex Chemistry** or **Eddy Break-Up** *Flow*: **Segregated Flow** Then: *Optional Models*: **Soot Emissions**	
Properties	Key properties are: Source Enabled Trigger, Begin, Secondary Gradients, Convection. See Soot Two-Equation Properties.	
Activates	Model Controls (child nodes)	Soot, Soot Species Option, Soot Model Properties
	Boundary Inputs	See Boundary Settings.
	Solvers	Soot. See Solvers.
	Monitors	SootScaledMassDensity, SootScaledNumberDensity
	Field Functions	Soot Coagulation Rate, Soot Growth Rate, Soot Mass Density, Soot Mean Diameter, Soot Nucleation Rate For Scaled Mass Density, Soot Nucleation Rate For Scaled Number Density, Soot Number Density, Soot Oxidation Rate, Soot Scaled Mass Density, Soot Scaled Mass Density Rate, Soot Scaled Number Density, Soot Scaled Number Density Rate, Soot Surface Density, Soot Volume Fraction. See Field Functions.

Soot Two-Equation Model Properties

Source Enabled Trigger

Allows you to deactivate chemistry at the beginning of the calculation—to let the flow field converge before starting to calculate the chemistry. You can choose when the chemistry is calculated, according to the following options:

- **Never**: The Source Enabled Trigger is deactivated, so chemical source terms are calculated from the first iteration/timestep.
- **Iteration**: The Source Enabled Trigger is activated, so the chemical source terms are activated and are calculated after the iteration that is specified as value of the *Begin* property. This option is available in steady-state cases.
- **Timestep**: The Source Enabled Trigger is activated, so the chemical source terms are activated and are calculated after the timestep that is specified as value of the *Begin* property. This option is available in unsteady-state cases.

Begin

Specifies the iteration or time-step after which the reactions are activated. During previous iterations or time-steps, reactions are deactivated. This feature is useful when modeling significant flow as you can deactivate reactions until a flow solution is found.

Secondary Gradients

Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces.

- **On**: Default value. Solves for interior and boundary types of secondary gradient.
- **Off**: Does not solve for either type of secondary gradient.
- **Interior Only**: Solves for the interior secondary gradients only.
- **Boundaries Only**: Solves for the boundary secondary gradients only.

Convection

In transport equations, you can choose from a range of schemes that calculate the convection term at a cell face. This calculation requires STAR-CCM+ to compute the face value of a quantity from the surrounding cell values. The method used for computing this face value has a profound effect on the stability and accuracy of the numerical scheme. For guidance on selecting a convection scheme, see Convective Flux.

- **1st-order**: First-order convection scheme.
- **2nd-order**: Second-order convection scheme.

Boundary Settings

Inlet or Pressure Outlet Boundary

Soot Scaled Mass Density Profile

Soot Scaled Number Density Profile

Wall Boundary

Wall Combustion Scalar Option

Sets the scalars for the wall combustion calculation.

Method	Corresponding Physics Value Nodes
Zero Flux	None
Specified Value	**Soot Scaled Mass Density Profile** **Soot Scaled Number Density Profile**
Specified Flux	**Mass Flux** Specifies a species flux at the boundary. The energy that is associated with the wall species influx or outflux is appropriately added.

Soot Solver Properties

Under-Relaxation Factor

In order to promote convergence, this property is used to under-relax changes of the solution during the iterative process. If residuals show solution divergence and do not decrease, reduce the under-relaxation factor for the relevant solvers.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Field Functions

Soot Coagulation Rate

$\left(\frac{dN}{dt}\right)_{co}$ in Eqn. (3541).

Soot Growth Rate

$\left(\frac{dN}{dt}\right)_{sg}$ in Eqn. (3542).

Soot Mass Density

M in Eqn. (3534). The units are $\left[\mathrm{kg/m^3}\right]$.

Soot Mean Diameter

d in Eqn. (3512).

Soot Nucleation Rate For Scaled Mass Density

$\left(\frac{dM}{dt}\right)_{nu}$ in Eqn. (3540).

Soot Nucleation Rate For Scaled Number Density

$\left(\frac{dN}{dt}\right)_{nu}$ in Eqn. (3539).

Soot Number Density

N in Eqn. (3509) and Eqn. (3533). The units are $\left[/\mathrm{m}^3\right]$.

Soot Oxidation Rate

$\left(\frac{dM}{dt}\right)_{ox}$ in Eqn. (3543).

Soot Scaled Mass Density

$\varnothing_M$ Obtained from equation Eqn. (3535). There are no units as this entity is dimensionless.

Soot Scaled Mass Density Rate

Computed from equation Eqn. (3537). There are no units as this entity is dimensionless.

Soot Scaled Number Density

$\varnothing_N$ Obtained by equation Eqn. (3535).

The units are $\left(\frac{kmol}{kg}\right)$.

Soot Scaled Number Density Rate

Computed from equation Eqn. (3536).

The units are $\left(\frac{kmol}{m^3-s}\right)$.

Soot Surface Density

S in Eqn. (3513).

Soot Volume Fraction

f_v in Eqn. (3510).

Soot

The Soot sub-node allows you to specify specific material properties for the soot material.

Soot Species Option

O Concentration

Uses the **Species List** approximation model for the concentration of oxygen.

OH Concentration

Uses the **Species List** approximation model for the concentration of the OH species.

C2H2 Concentration

Uses the **Species List** approximation model for the concentration of the C2H2 species.

Soot Model Properties

Nucleation Option

Allows you to select one of two nucleation options.

- **PAH-based**: This nucleation model pathway, Eqn. (3538), is based on polycyclic aromatic hydrocarbons (PAH). This option is only available when the gas-phase mechanism contains the species: C6H6, C6H5, C2H2, and H2.

  PAH nucleation requires more species than acetylene nucleation, and is therefore more computationally expensive due to the large chemical mechanisms involved.
- **Acetylene-based**: This nucleation model pathway, Eqn. (3539) and Eqn. (3540), is based on the acetylene concentration.

Surface-growth Scale

Scales surface growth (W_r in Eqn. (3508)).

Coagulation Scale

Scales $\left(\frac{dN}{dt}\right)_{co}$ in Eqn. (3541).

Nucleation Scale

Scales nucleation (R_r in Eqn. (3508)).

Oxidation Scale

Scales oxidation (W_r in Eqn. (3508)).

Cold Temperature Limit

If the temperature in the cell is lower than this number, then cold flow is assumed—no soot source calculations are performed in that cell.

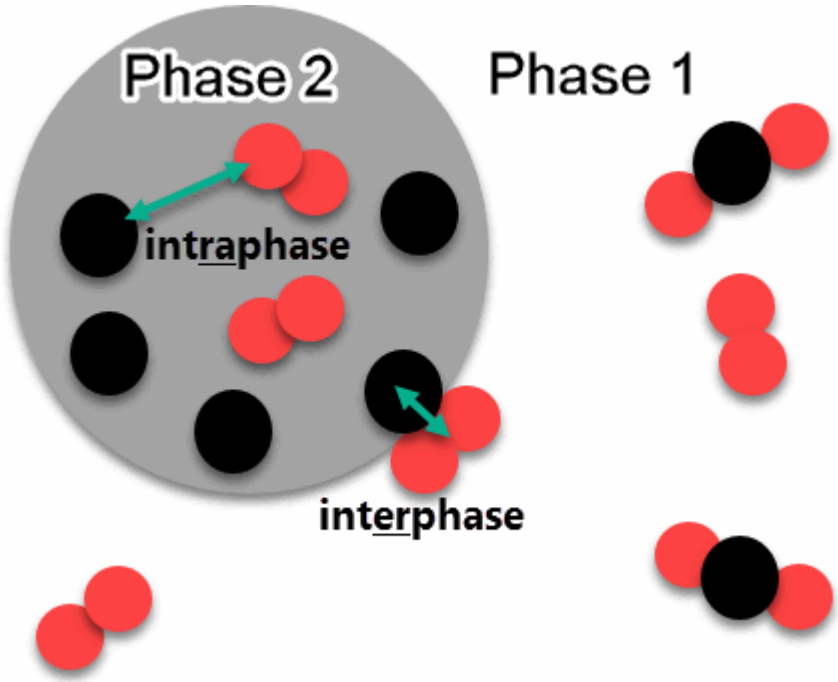
Soot User Rates

Allows you to include a custom soot rate.

Interphase

There are many applications that involve the transport of reacting particles, such as calcination and char oxidation. You can simulate the process of interphase reactions using Lagrangian Multiphase or Eulerian Multiphase models.

The following diagram explains the difference between interphase and intraphase reactions between species.



- Intraphase reactions occur between atoms/molecules in the same phase.
- Interphase reactions occur between atoms/molecules in different phases at the interface between the different phases.

Lagrangian Multiphase Interphase Reactions

You can simulate the process of reacting solid particles using one of the Lagrangian Multiphase models. The solid particles can interact with the continuous phase using the Two-Way Coupling model.

When simulating applications such as coal or biomass combustors, you can use the Lagrangian Particles model to simulate small solid particles that are traveling in a flow. Larger clumps are simulated as Discreet Element Method (DEM) particles. STAR-CCM+ also provides the Particle Porosity model which you can use to specify a factor by which solid particles shrink due to reactions.

Particle components can devolatilize or react with a material in a gas phase to form products in gas or solid phases.

To model the transport of reacting particles, the Material Particles model is required in the Lagrangian phase. Material particles have mass and volume, obey the physical conservation laws of mass and momentum, and

are assumed to be internally homogenous, without internal motion. The type of Material model that is selected influences the type of mass transfer model that is made available:

- When Multi-Component Particle is selected, you can select **Particle Reaction**.

  You can then model the following types of particle reaction (use these models together or separately):
  - **Particle Combustion**
  - **Particle Devolatilization**
- When Multi-Component Coal is selected, you can select **Coal Combustion**. You can use the Coal Combustion model to simulate coal furnaces and gasification.

Particle Combustion

You can use the Particle Combustion model to simulate various particle reactions such as char oxidation, biomass particle oxidation, and ferric oxide reduction. It is possible to simulate multiple reactions between solid particles reacting with a gas-phase species to form solid and/or gas-phase products.

Particle Devolatilization

The Particle Devolatilization model can simulate various particle mass transfer processes, such as gasification, sublimation, phase transformation, and biomass/coal devolatilization. Particle devolatilization is a gasification process where a solid particle is converted to volatile matter and/or a residue.

Coal Combustion

Coal is an important fuel that is used in electricity generation worldwide. It is pulverized to a fine powder before being blown into the combustion chambers of high-temperature furnaces.

The Coal Combustion model in STAR-CCM+ has pre-defined species and reactions that are designed specifically for simulating coal combustion. You can use this model to simulate the transport, gasification, and combustion of coal particles as they pass through such furnaces. This model also allows you to simulate coal fuel NOx emissions.

Coal is a composite material consisting of organic and inorganic substances that can also be moist. The coal particle is assumed to be initially coated with moisture so that a single-component evaporation treatment is appropriate. The driving force for evaporation of this layer is the departure from equilibrium of the liquid-vapor system. In coal combustion, the moisture evaporation is usually heat-transfer limited and takes place rapidly. The devolatilization and char oxidation reactions occur after the moisture has evaporated. In addition to an appropriate combustion model in the continuous phase, coal combustion requires the following Lagrangian phase submodels:

- **Coal Moisture Evaporation**
  Represents the Quasi-Steady Evaporation model for the moisture component (H2O) of the coal particle. This model allows the H2O component of the coal particle to lose its mass through a quasi-steady evaporation process.
- **Raw Coal Devolatilization**
  Defines how the combustible substances are made available for combustion, using either the Two-Step Devolatilization model or the User-Defined Devolatilization model. The Volatile Vapor Component

property allows you to select the species to which the raw coal devolatilizes. If no species is selected, an error message (`Error: no CoalVolatile vapor selected for Devolatilization model in coal`) is issued when the solution field is initialized. The benefit of this property is that it allows you to model two different coal types with two different types of CoalVolatiles.

- **Two-Step Devolatilization**
  Controls the process by which raw coal is converted to gaseous combustible material (volatile matter) and solid char residue. Experimental studies have shown that, under certain heating conditions, the volatile matter yield is higher than the value that is obtained from standard proximate analysis. This effect is accounted for by allowing the coal material to devolatilize by two temperature-dependent competing reactions—the Two-Step Devolatilization model [648].
- **User-Defined Devolatilization**
  This model allows you to define the devolatilization rate for the raw coal component of the coal. In addition, you can set the volatile yield in the properties of this model.

- **Char Oxidation**:
  There are three types of char oxidation models available. Each specific char oxidation model node contains three reaction nodes, one for each of the possible char oxidation processes.
  - Char Reaction With O2
  - Char Reaction With H2O
  - Char Reaction With CO2

  Only the first reaction, Char Reaction With O2, is active by default. If you require a more accurate simulation and have reaction rates for the other two reactions, you can activate the two further optional reactions.
  - **First-Order Char Oxidation**
    This model is used to predict the rate at which the char content of the coal burns. The time that is required for this char burnout is a significant part of the coal combustion process. The First-Order Char Oxidation model consists of one or more First-Order Combined Rate reactions that account for gas phase diffusion of the oxidizer as well as the first-order surface reaction kinetics.
  - **Half-Order Char Oxidation**
    The Half-Order Char Oxidation model consists of one or more Half-Order Combined Rate reactions that account for gas phase diffusion of the oxidizer as well as the half-order surface reaction kinetics.
  - **User-Defined Char Oxidation**
    This model provides an alternative to the two other char oxidation rates which you can use to set custom rates as a profile or as a constant.

When these Lagrangian phase models are used with a combustion model in the continuous phase—and if the Two-Way Coupling phase model is also active—the complete coal combustion process can be realized. It is also possible to model a non-reacting Lagrangian phase containing the multi-component coal material model only, without activating coal combustion.

Particle Porosity

STAR-CCM+ also provides the optional Lagrangian model, Particle Porosity, which allows you to define how the porosity of particles which burn internally changes through the course of a coal combustion reaction. The porosity of a particle determines how the diameter and surface area of the particle changes throughout a reaction. When the surface area of a particle changes, the overall reaction rate is affected.

Eulerian Multiphase Interphase Reactions

You can simulate reactions between two different Eulerian phases using the Interphase Reaction model. For example, between fluid and solid particles in a fluidized bed reactor.

Interphase Reaction

The Interphase Reaction model is available to use with the Multiphase Segregated Flow or Volume of Fluid (VOF) variants of the Eulerian Multiphase model.

The Interphase Reaction model allows you to specify the reactant and product species from either of the phases that are defined within a phase interaction. Phase interactions can define reactions between two phases of the same or different material types—Multi-component Gas, Multi-component Liquid, or Multi-component Particle. You can define specific species which are involved in the interphase reaction from either phase and also specify to which phase each product belongs.

Contents:

Interphase Workflow: Lagrangian Multiphase
Interphase Workflow: Eulerian Multiphase
Interphase Reference

Interphase Workflow: Lagrangian Multiphase

You can simulate reacting particles using the Lagrangian Multiphase Particle Reaction or Coal Combustion models.

The steps in this workflow are intended to follow on from the initial steps in the Reacting Flow General Workflow.

1. For the Physics Continuum that represents the background phase in which the particle reactions occur (the continuous phase), select the following models—in addition to the models that are previously selected, with **Auto-Select recommended models** activated:

Group Box	Model
Optional Models	**Lagrangian Multiphase**

2. Select any further Optional models as necessary.
   For example, if you are simulating reacting flow of solid granular particles such as powders or slurries, select the **Discrete Element Model (DEM)**. See Using the Discrete Element Method.

In Lagrangian phase particle reactions, you can define both reactants and products from the continuous phase as reaction components.

3. Define the gas or liquid components in the continuous phase.
   a) Expand the **Continua** > **Continuum** > **Models** > **[Multi-Component Gas]** node.
   b) Define the species and corresponding material properties.
4. Add at least one Lagrangian phase. Right-click the **Continuum** > **Models** > **Lagrangian Multiphase** > **Lagrangian Phases** node and select **New**.

If you are simulating only one type of reacting particle, only one Lagrangian phase is required. See: Creating One or More Lagrangian Phases.

5. For each Lagrangian Phase, select the following models, with **Auto-Select recommended models** activated:

Group Box	Model
Particle Type	• **Material Particles** • **DEM Particles** (when using DEM)
Material	• **Multi-Component Coal**: Designed specifically for simulating coal combustion. • **Multi-Component Particle**: For example, when modeling combustion or devolatilization of a multi-component solid.
Mass Transfer (when **Multi-Component Coal** or **Multi-Component Particle** are selected)	• When Multi-Component Particle is selected, select **Particle Reaction** • When Multi-Component Coal is selected, select **Coal Combustion**
Char Oxidation (when **Coal Combustion** is selected)	• **First-Order Char Oxidation** • **Half-Order Char Oxidation** • **User-Defined Char Oxidation**
Raw Coal Devolitilization (when **Coal Combustion** is selected)	• **Two-Step Devolitilization** • **User-Defined Devolitilization**
Particle Reaction Type (when **Particle Reaction** is selected)	Select one or both: • **Particle Combustion** • **Particle Devolitilization**
Particle Shape (when **DEM Particles** is selected when using DEM)	any (See DEM Particle Types)
Equation of State	• **Constant Density** • **Polynomial Density**

When **Auto-select recommended models** is activated, the following additional phase models are included:

- **Pressure Gradient Force**
- **Spherical Particles** (except with DEM)
- **Energy**
- **Species**
- **Two-Way Coupling**—when the reaction includes interaction with the continuous phase.

6. Select any additional Lagrangian phase models that are required. See: Lagrangian Phase Models.

   For example:

- **Track File**
- **Coal Fuel NOx**
- **Particle Radiation**

7. Define the components in the Lagrangian phases.
   a) Expand the **[Continuum]** > **Models** > **Lagrangian Multiphase** > **Lagrangian Phases** > **[Phase]** > **Models** node.
   b) Set the components as described below.
      - For the **Multi-Component Particle** model:

        Right-click the **Multi-Component Particle** > **Particle Components** node and import species or select mixture components from the material database.
      - For the **Multi-Component Coal** model:

        The appropriate species which represent the composition of coal particles are already defined under the **Multi-Component Coal** > **Coal Components** node:
        - **RawCoal**
        - **Char**
        - **Ash**
        - **H20**
   c) If necessary, you can adjust the **[Component]** > **Material Properties**.

      For each component (other than **RawCoal**, **Char**, and **Ash**—which are pre-defined specifically for coal combustion), the Molecular Weight material property method is set to **Elemental Composition** by default. The molecular weight depends on the settings within the Elemental Composition material property node—elemental composition is vital for checking the stoichiometry for a particle reaction.
   d) For DEM particle types, select the **[Particle Type]** node and specify the required settings.
      For more information on particle types, see:
      - The Spherical Particles Model
      - Coarse Grain Particle Models
      - The Composite Particles Model
      - The Particle Clumps Model
      - Working with Individually Composed Shapes
      - Working with Sphere-Filled Parts
      - The Cylindrical Particles Model
      - The Polyhedral Particle Model
8. Define the Particle Reactions.
   - For the **Multi-Component Particle** model:
     1. Right-click the **Particle Combustion** > **Particle Reactions** or **Particle Devolitilization** > **Particle Reactions** node and select **New Particle Reaction**.
     2. Expand the **Particle Reactions** > **Reaction** node and define the Solid and Gas Reactants and Products.
     3. Select the **Reaction** > **Reaction Rate** node and select the method that is required for calculating the reaction rate.

- For the **Multi-Component Coal** model, the necessary coal combustion reactions are already defined under the **[Char Oxidation]** node.

9. Set the parameters of other Lagrangian phase models as required.

   See: Lagrangian Phase Models.

10. Collapse the **Lagrangian Multiphase** model node.
11. Set the parameters of the remaining **[Continuum]** > **Models**, as required.
12. For the continuum, define any necessary parameters for the **Reference Values** and **Initial Conditons**.
    - For combustion/devolatilization reactions, make sure that particles reach a high enough temperature to devolatilize and form combustible gaseous fuel. Either set a high initial temperature (around 800K), or temporarily raise the inlet gas temperature or the injection temperature of the coal particles.
    - Make sure that you specify the species mass fraction or species mole fraction for the initial reaction components (in the continuous phase).
13. Define any necessary Physics Conditions and Values for the Region and Boundaries.

    See: Lagrangian Phase Boundary Conditions.

14. If necessary, set up Injectors. The injectors allow you to define where, in what direction, and with what frequency, the Lagrangian particles enter the continuous phase.

    See: Working With Injectors.

15. Return to the Reacting Flow General Workflow.

Interphase Workflow: Eulerian Multiphase

When defining reactions within an Eulerian phase, you can include species from the Particle or Gas models within a different Eulerian phase that is specified within the same phase interaction.

The steps in this workflow are intended to follow on from the initial steps in the Reacting Flow General Workflow.

1. For the Physics Continuum that represents the background phase in which the Eulerian phase reactions occur (the continuous phase), select the following models—in addition to the models that are previously selected, with **Auto-Select recommended models** activated:

Group Box	Model
Eulerian Multiphase Model	Select either: • **Multiphase Segregated Flow** • **Volume of Fluid (VOF)**
Optional Models	**Phase Coupled Fluid Energy**

2. Select any further Optional models as necessary.
3. Under the **Eulerian Multiphase** model node, create and define at least two Eulerian phases using the following guidance on model selection (along with any other models that are required).

Group Box	Model
Material	For Eulerian phases which participate in reactions, select one of:

Group Box	Model
	• **Multi-Component Gas** • **Multi-Component Liquid** • **Multi-Component Particle**
Energy	**Segregated Fluid Enthalpy**
Reaction Regime	• For phases in which intraphase reactions occur, select **Reacting** • For any other phases, select **Non-Reacting**
Reacting Species Models (when the Reacting model is selected)	• When using large reaction mechanisms that are imported in the standard format, select **Complex Chemistry** • When using small mechanisms that are defined in the GUI with four steps or less, select **Eddy Break-Up**
Equation of State (except when the Complex Chemistry model is selected)	any
Turbulence Chemistry Interactions (when the Complex Chemistry model is selected)	Select either: • **Eddy Dissipation Concept** • **Laminar Flame Concept**

See Defining Eulerian Phases.

Phase interactions can define reactions between any two phases of Multi-component Gas, Multi-component Liquid, or Multi-component Particle. You can specify any number of reactions within a phase interaction and any number of phase interactions. However, each phase interaction can only contain two phases.

Create and define the phase interactions.

4. Right-click the **Multiphase Interaction** > **Phase Interactions** node and select **New**.
5. Right-click the **Phase Interactions** > **[Phase Interaction]** > **Models** node and from the *Phase Interaction Topology* group box, select either of the following (with **Auto-Select recommended models** activated):
   - **Continuous Dispersed Phase Interaction**
   - **Multiple Flow Regime Phase Interaction**
6. Select any further Optional models for the phase as necessary (except **Interphase Mass Transfer**).
7. Depending on the Phase Interaction Topology that is specified, either:
   - Select the **Continuous Dispersed Phase Interaction** node and set the phases for the *Continuous Phase* and *Dispersed Phase*.
   - Select the **Multiple Flow Regime Phase Interaction** node and set the phases for the *Primary Phase* and *Secondary Phase*.
8. Right-click the **Phase Interactions** > **[Phase Interaction]** > **Models** node and select the following phase interaction models, with **Auto-Select recommended models** activated:

Group Box	Model
Optional Models	**Interphase Mass Transfer**
Interphase Mass Transfer Rate/ Interphase Mass Transfer Models	**Interphase Reaction**

9. Create and define the interphase reactions within each phase interaction.

   You can specify reactant and product species from either of the two phases that are specified for a phase interaction.

   See Setting Up Interphase Reactions.

10. Return to the Reacting Flow General Workflow.

Interphase Reference

This section provides reference material, including properties, sub-nodes, field functions, and other model-specific settings, for the specific reacting flow interphase models.

You can find more information about using Lagrangian Multiphase and Eulerian Multiphase models in the following sections of the User Guide:

- Using the Lagrangian Multiphase Model
- Using the Eulerian Multiphase Model

Contents:

Particle Reaction Model Reference
Particle Combustion Model Reference
Particle Devolatilization Model Reference
Particle Reactions Reference
Coal Combustion Model Reference
Coal Moisture Evaporation Model Reference
Interphase Reaction Model Reference

Particle Reaction Model Reference

There are many applications that involve the transport of reacting particles, such as calcination and char oxidation. The particle reaction models let you simulate the process of such reacting solid particles through Lagrangian Multiphase models.

The particle components can devolatilize or react with a material in a gas phase to form desired products in gas/solid phases.

The calculation of particle reaction requires the Gas or Multi-Component Gas model for the continuous phase, along with either of the following submodels:

- Particle Combustion
- Particle Devolatilization

Particle Reaction Model Reference

Theory	See Particle Reactions.	
Provided By	**[Lagrangian Phase]** > **Models** > *Mass Transfer*	
Example Node Path	**[physics continuum]** > **Models** > **Lagrangian Multiphase** > **Lagrangian Phases** > **[Phase 1]** > **Models** > **Particle Reaction**	
Requires	Either:	
	Material: **Gas** *Flow*: any *Optional Models*: **Lagrangian Multiphase**	*Material*: **Multi-Component Gas** *Reaction Regime*: **Non-reacting** *Flow*: any *Optional Models*: **Lagrangian Multiphase**
	Then, in a Lagrangian phase: *Particle Type*: **Material Particles** *Material*: **Multi-Component Particle**	
Activates	Physics Models	*Two-Way Coupling*: **Two-Way Coupling** *Energy*: **Energy**

Particle Combustion Model Reference

The Particle Combustion model allows you to simulate processes in which a solid component reacts with a gas-phase component to form solid and/or gas-phase products.

Particle Combustion Model Reference

Theory	See Particle Combustion.	
Provided By	**[Lagrangian Phase]** > **Models** > *Particle Reaction Type*	
Example Node Path	**[physics continuum]** > **Models** > **Lagrangian Multiphase** > **Lagrangian Phases** > **[Phase 1]** > **Models** > **Particle Combustion**	
Requires	Either:	
	Material: **Gas** *Flow*: any *Optional Models*: **Lagrangian Multiphase**	*Material*: **Multi-Component Gas** *Reaction Regime*: **Non-reacting** *Flow*: any *Optional Models*: **Lagrangian Multiphase**
	Then, in a Lagrangian phase: *Particle Type*: **Material Particles** *Material*: **Multi-Component Particle** *Mass Transfer*: **Particle Reaction**	
Specific Right-Click Actions	New Particle Reactions. See Right-Click Actions.	

Activates	Model Controls (child nodes)	Particle Reactions. See Particle Reactions Reference.

Right-Click Actions

New Particle Reactions

Creates a **Particle Reactions** sub-node. This new object contains sub-nodes for specifying reactants, products, and the reaction rate.

Particle Devolatilization Model Reference

The Particle Devolatilization model controls the process by which raw coal is converted to gaseous combustible material (volatile matter) and solid char residue.

This model supplies the rate of devolatilization for the raw coal component of the coal.

Particle Devolatilization Model Reference

Theory	See Particle Devolatilization.	
Provided By	**[Lagrangian Phase]** > **Models** > *Particle Reaction Type*	
Example Node Path	**[physics continuum]** > **Models** > **Lagrangian Multiphase** > **Lagrangian Phases** > **[Phase 1]** > **Models** > **Particle Devolatilization**	
Requires	Either:	
	Material: **Gas** *Flow*: any *Optional Models*: **Lagrangian Multiphase**	*Material*: **Multi-Component Gas** *Reaction Regime*: **Non-reacting** *Flow*: any *Optional Models*: **Lagrangian Multiphase**
	Then, in a Lagrangian phase: *Particle Type*: **Material Particles** *Material*: **Multi-Component Particle** *Mass Transfer*: **Particle Reaction**	
Specific Right-Click Actions	New Particle Reactions. See Right-Click Actions.	
Activates	Model Controls (child nodes)	Particle Reactions. See Particle Reactions Reference.

Right-Click Actions

New Particle Reactions

Creates a **Particle Reactions** sub-node. This new object contains sub-nodes for specifying reactants, products, and the reaction rate.

Particle Reactions Reference

You can define Particle Reactions under the Particle Combustion or Particle Devolatilization nodes.

Components from the continuous gas phase can be used as both reactants and as products in the particle reactions within the Lagrangian phase. Therefore, these components need to be defined in the multi-component gas phase mixture in the continuous phase.

Particle Reactions

The Particle Reactions node has no properties, however, the context menu has the following options:

New Particle Reaction

Creates a particle reaction. This new object contains sub-nodes for reactants, products, and the reaction rate.

Reactants / Products

The **Reactants** and **Products** sub-nodes both have a context menu with the following options:

Add Solid Reactant

Add Gas Reactant

Add Solid Product

Add Gas Product

Selecting one of these options opens a submenu of all materials of the same phase in the material database. When you select a material, a **[Reaction Component]** sub-node is created for that material. You can add as many reactant or product components as required.

[Reaction Component]

Each **[Reaction Component]** node has the following Properties:

Stoich. Coeff.

Allows you to set the stoichiometric coefficient for that reaction component in the reaction definition.

Reaction Rate

The **Reaction Rate** node, which controls the rate of the particle devolatilization or particle combustion reaction, has the following *Properties*:

Method

- **First-Order Combined Rate**: When using the Particle Combustion model, activates the **First Order Combined Rate** node.
- **Half-Order Combined Rate**: When using the Particle Combustion model, activates the **Half-Order Combined Rate** node.
- **First-Order Rate**: When using the Particle Devolitilization model, activates the **First-Order Rate** node.
- **User Reaction Rate**: Activates the **User Reaction Rate** node.

[Reaction Rate Method] Properties

The **First-Order Combined Rate**, **Half-Order Combined Rate**, and **First-Order Rate** sub-nodes have the following Properties:

Temperature Exponent, Beta

Sets the temperature exponent β for this reaction. See Eqn. (3550), Eqn. (3553), and Eqn. (3301).

Activation Energy, Ea

Sets the temperature exponent E_a for this reaction. See Eqn. (3550), Eqn. (3553), and Eqn. (3301).

First-Order Combined Rate

The first-order combined rate, available for the Particle Combustion model, calculates the effective particle reaction rate by accounting for both chemical kinetics at the particle surface and diffusion of gas-phase reactant species. The term first-order means that the kinetics reaction rate is first-order with respect to gas-phase reactant species concentration. The First-Order Combined Rate node uses its properties to control part of this calculation method, while its sub-nodes Pre-exponent and Diffusion Coefficient provide inputs as scalar profiles. See Eqn. (3553) and Eqn. (3554).

Half-Order Combined Rate

The half-order combined rate, available for the Particle Combustion model, calculates the effective particle reaction rate by accounting for both chemical kinetics at the particle surface and diffusion of gas-phase reactant species. The term half-order means that the kinetics reaction rate is half-order with respect to gas-phase reactant species concentration. The Half-Order Combined Rate node uses its properties to control part of this calculation method, while its sub-nodes Pre-exponent and Diffusion Coefficient provide inputs as scalar profiles. See Eqn. (3301) and Eqn. (3556).

First-Order Rate

The first-order rate, available for the Particle Devolatilization model, calculates the particle reaction rate where the chemical kinetics is first-order with respect to solid-particle reactant concentration. The First-Order Rate node uses its properties to control part of this calculation method, while its sub-node Pre-exponent provides inputs as a scalar profile. See Eqn. (3550) and Eqn. (3551).

User Reaction Rate

The user-reaction rate, available for both Particle Combustion and Particle Devolatilization models, allows you to specify custom reaction rates. The User Reaction Rate node functions as a scalar profile (see Eqn. (3557)).

Pre-exponent

The Pre-exponent node, which is a sub-node of reaction rate nodes other than the User Reaction Rate node, functions as a scalar profile. It lets you use the Arrhenius form. See Eqn. (3550), Eqn. (3553), and Eqn. (3301).

Diffusion Coefficient

The Diffusion Coefficient node is included in the First-Order Combined Rate and Half-Order Combined Rate nodes (Particle Combustion only). It is needed because this reaction includes a gas oxidizer, and you have to know the rate at which it diffuses. It represents D_m in Eqn. (3555), and functions as a scalar profile.

Coal Combustion Model Reference

There are several Lagrangian phase models available to select alongside the Coal Combustion model for simulating coal combustion in STAR-CCM+.

Coal Combustion Model Reference

Theory	See Coal Combustion.	
Provided By	**[Lagrangian Phase]** > **Models** > *Mass Transfer*	
Example Node Path	**[physics continuum]** > **Models** > **Lagrangian Multiphase** > **Lagrangian Phases** > **[Phase 1]** > **Models** > **Coal Combustion**	
Requires	In the continuous phase, either:	
	Material: **Gas** *Flow*: any *Optional Models*: **Lagrangian Multiphase**	*Material*: **Multi-Component Gas** *Reaction Regime*: **Non-reacting** *Flow*: any *Optional Models*: **Lagrangian Multiphase**
	Then, in a Lagrangian phase: *Particle Type*: **Material Particles** *Material*: **Multi-Component Coal**	
Activates	Physics Models	*Two-Way Coupling*: **Two-Way Coupling** *Energy*: **Energy** *Moisture Evaporation*: **Coal Moisture Evaporation**

Coal Moisture Evaporation Model Reference

The Coal Moisture Evaporation model allows the H2O component of the coal particle to lose its mass through a quasi-steady evaporation process.

The coal particle is assumed to be initially coated with moisture so that a single-component evaporation treatment is appropriate. The driving force for evaporation of this layer is the departure from equilibrium of the liquid-vapor system. In coal combustion, the moisture evaporation is usually heat-transfer limited and takes place rapidly. The devolatilization and char oxidation reactions occur after the moisture has evaporated.

The formulation is that of the Quasi-Steady Single-Component Droplet Evaporation model as applied to a water droplet, and the associated Sherwood and Nusselt numbers are calculated using the Ranz-Marshall correlation.

Coal Moisture Evaporation Model Reference

Theory	See Droplet Evaporation
Provided By	**[Lagrangian Phase]** > **Models** > *Moisture Evaporation*
Example Node Path	**[physics continuum]** > **Models** > **Lagrangian Multiphase** > **Lagrangian Phases** > **[Phase 1]** > **Models** > **Coal Moisture Evaporation**
Requires	In the continuous phase, either:

	Material: **Gas** *Flow*: any *Optional Models*: **Lagrangian Multiphase**	*Material*: **Multi-Component Gas** *Reaction Regime*: **Non-reacting** *Flow*: any *Optional Models*: **Lagrangian Multiphase**
	Then, in a Lagrangian phase: *Particle Type*: **Material Particles** *Material*: **Multi-Component Coal** *Mass Transfer*: **Coal Combustion**	
Activates	Field Functions	Coal Moisture Latent Heat of Vaporization (H2O), Coal Moisture Saturation Pressure (H2O). See Field Functions.

Field Functions

Coal Moisture Latent Heat of Vaporization (H2O)

Coal Moisture Saturation Pressure (H2O)

Interphase Reaction Model Reference

You can use the Interphase Reaction model to define reactions that result in mass transfer between different Eulerian phases.

Interphase Reaction Model Reference

Theory	
Provided By	**Multiphase Interaction** > **Phase Interactions** > **[Phase Interaction]** > **Models** > *Interphase Mass Transfer Rate*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Interphase Reaction**
Requires	• Main physics continuum: *Material*: **Eulerian Multiphase** *Eulerian Multiphase Model*: any *Optional Models*: **Multiphase Interaction** • **Eulerian Multiphase** > **Eulerian Phases** > **[Eulerian Phase]**: (Define a minimum of two Eulerian phases). For each Eulerian phase, select *Material*: **Multi-Component Gas**, **Multi-Component Liquid**, or **Multi-Component Particle** *Energy*: **Segregated Fluid Enthalpy** Define the multi-component materials. • **Multiphase Interaction** > **Phase Interactions** > **[Phase Interaction]**: For each phase interaction, select

	Phase Interaction Topology: **Continuous-Dispersed Phase Interaction** Define the continuous and dispersed phases within the **Continuous-Dispersed Phase Interaction** node. Select *Optional Models*: Interphase Mass Transfer	
Activates	Model Controls (child nodes)	Reactions. See Reacting Model Reference: Reactions.

Surface Chemistry

The Surface Chemistry model is used to represent surface reaction mechanisms and surface species on the reacting walls of solid regions, or surfaces in porous regions. You can use this model to simulate applications such as monolithic catalysts, chemical vapor deposition (CVD), and packed bed reactors.

Surface Chemistry Model

There are two options available for simulating surface chemistry reactions.

- If the simulation involves reacting flow reactions other than the surface reactions, you can: follow another reacting flow workflow, select Surface Chemistry as an optional model, and then follow the Surface Chemistry workflow.
- If the simulation only involves surface reactions and does not involve any other reacting flow reactions, you can follow the Surface Chemistry workflow.

You can simulate surface chemistry reactions on fluid regions with reacting surfaces when using the following models:

- The Complex Chemistry model.
- The Eddy Break-Up (EBU) model.
- The Non-reacting and Segregated Species models.

When using the Complex Chemistry model, it is also possible to simulate surface chemistry reactions on porous regions.

The **Surface Chemistry** model provides The Surface Mechanism Manager which allows you to define bulk species, site species, and reactions using the **Reacting Surface** and **Solid Composition** submodels.

Note: You can import surface mechanisms which contain multiple sites or bulk mixtures within the same surface mechanism.

The way in which the ODEs (Ordinary Differential Equations) that arise from the reaction mechanism are solved depends on the other physics models that are selected.

- When Surface Chemistry is selected with the Complex Chemistry model, depending on the complex chemistry settings, either DARS-CFD or CVODE resolves the reaction system.
- When Surface Chemistry is selected with the Eddy-Break Up model, or with the Non Reacting and Segregated Species models, the Surface-Gas Interaction model resolves the reaction system.

The Reacting Surface Model

The Reacting Surface model defines chemical reactions for the surface mechanism in the same manner that the Reacting model defines reactions for the continuum gas phase. It uses reaction definition objects. Use this node to define chemical reactions occurring between mixture components that were defined earlier, either by importing Chemkin-formatted data or manually (see Defining Chemical Reactions Manually).

The Solid Composition Model

The Solid Composition model is a multi-component material model that allows you to specify two distinct types of surface species:

- **Solid Composition** > **[Site Solid Mixture]** (surface species that occupy a surface site that is next to the gas)
- **Solid Composition** > **[Bulk Solid Mixture]** (species that are underneath the site layer)

  When the Complex Chemistry model is set to use the DARS-CFD or CVODE ODE solver to resolve surface chemistry reaction mechanisms, the **[Site Solid Mixture]** and **[Bulk Solid Mixture]** managers cannot be manipulated explicitly. No adding, removing, or renaming species is possible. However, when a complex chemistry definition is imported, these nodes are populated with the mechanisms and surface species that are involved in surface-gas reactions.

  When the Surface-Gas Interaction model is used, you can define the surface chemistry mechanisms, species, and reactions, either manually, or by importing the Chemkin-formatted surface chemistry definition.

The Surface-Gas Interaction Model

The Surface-Gas Interaction Model is activated when the Surface Chemistry Model is chosen together with either, the EBU model, or the Non-Reacting and Segregated Species models. The Surface-Gas Interaction Model solves the ODEs that arise from the surface reaction system using the CVODE solver from the Sundials kit. If you are using the Surface-Gas Interaction model with the Surface Chemistry model, first import a chemical library for the Surface Chemistry model.

Limitations Modeling Surface Chemistry

- Surface chemistry can only be applied to wall and baffle boundary conditions.
- No energy equations are solved for the reactive surface. Fixed temperature boundary conditions apply.
- Surface growth and shrinking are not taken into account.

You specify the surface reaction mechanism, which you can import (See Reaction Mechanism Formats). The surface reaction rate is defined per unit surface area—different from fluid-phase reactions, which are based on unit volume.

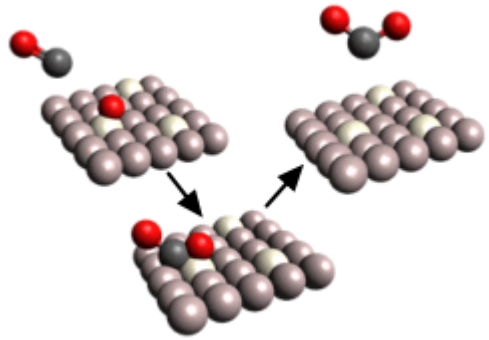
Contents:

Surface Chemistry Workflow
Surface Chemistry Reference

Surface Chemistry Workflow

Use the following procedure to simulate surface reactions which occur when a gaseous mixture contacts a reacting surface in a fluid or porous region.

The steps in this workflow are intended to follow on from the initial steps in the Reacting Flow General Workflow.



1. For the physics continuum which contains the reacting surface, select the following models in order—in addition to the models that are previously selected, with **Auto-select recommended models** activated.

Group Box	Model
Reacting Flow Models (when Reacting is selected)	**Reacting Species Transport**
Reacting Species Models (when Reacting is selected)	• If the reactions that are occurring within the gaseous mixture have simple mechanisms and the reacting surfaces are not on porous regions, select **Eddy Break-Up**. This model is not compatible with reacting surfaces on porous regions. • If the reactions that are occurring within the gaseous mixture have detailed mechanisms, or the reacting surfaces are on porous regions, select **Complex Chemistry**.
Turbulence Chemistry Interactions (when Complex Chemistry is selected)	• For steady-state diffusion flames, select **Eddy Dissipation Concept** • For premixed, partially-premixed, and unsteady flames, **Laminar Flame Concept**
Optional Models	**Surface Chemistry**
Enabled Models	When Non-Reacting is selected, or when Reacting and Eddy Break-Up are selected, **Surface-Gas Interaction** is selected automatically.

2. Select any Optional models that are necessary. For example:
   - **NOx Emission** or **Soot Emissions**—to model the formation of these pollutants.
   - The **Radiation** model is useful for modeling applications in which radiative heat transfer is important, such as in glass furnaces. The soot emission models influence the **Participating Media Radiation (DOM)** and **Gray Thermal Radiation** models by contributing to the absorption coefficient of the continuous phase (the absorption coefficient describing both absorption and emission).
3. Import the surface mechanism using one of the following procedures, depending on the models that are previously selected:
   - **Using the Complex Chemistry model with the CVODE solver**

1. To import the surface chemistry definition, right-click the **Complex Chemistry** node and select **Import Complex Chemistry Definition (Chemkin format)**.
2. In the *Import Chemkin Files* dialog, select appropriate files for the *Gas Mechanism* and the *Surface Mechanism*, then click **OK**.

   Upon import, the **Surface Chemistry** > **Surface Mechanism Manager** node becomes populated with the surface mechanism.
3. Set the CVODE properties.

- **Using the Complex Chemistry model with the DARS-CFD ODE solver**

  If you are using the Complex Chemistry model with the DARS-CFD ODE solver and the Surface Chemistry model, first import a chemical library for the DARS-CFD solver.

  1. To import a chemical library, right-click the **Complex Chemistry** > **DARS-CFD** node and select an appropriate option. See, DARS-CFD Properties and Importing the Complex Chemistry Definition.

     Upon import, the **Surface Chemistry** > **Surface Mechanism Manager** node becomes populated with the surface mechanism.
  2. Set the DARS-CFD properties.
  3. If necessary, you can modify the reaction species source terms or reaction rates.

- **Using the Surface-Gas Interaction model**

  The Surface-Gas Interaction model is required when the Non-Reacting model is selected, or when the Reacting model and the Eddy Break-Up model are selected.

  1. Check if the standard or extended material database contains all the species that are involved in a mechanism:
     - If it contains all the relevant species, skip to Step (3).
     - If not, create your own material database in Step (2).
  2. Create a material database.

     The Chemkin mechanism must contain, at a minimum, the description of the NASA polynomials for all the species that are involved in the mechanism. Put this information into the mechanism file itself or in separate files.

     There can be more than one file describing the NASA polynomials (usually one for gas and one for surface species). Import these files into the new material database.
  3. To define the gas species, follow the procedure for importing mixture components from a file. Then, in the *Select file and material database* dialog, select the material database that contains the data for the mechanism.
  4. Define the site species, bulk species, and (if applicable) reactions:
     - To define all of these phenomena at once, right-click the node of the surface mechanism and select **Import Species and Reactions**. Use this option if you need only one material database for the surface mechanism species and the reactions.
       - In the *Select file and material database* dialog, select the file that contains the surface species and reactions information.
       - Select the material database that contains the data for the mechanism.
     - Alternatively, to specify another material database for the reactions:

- Select **Import Species** from the same context menu.
- Proceed to Step 5 to define the reactions.

5. Define reactions:
   a. Right-click the node of the surface mechanism and select **Import Species and Reactions**.
   b. Choose again the file that contains the surface species information and the material database that is relevant for this case.

To have alternate surface chemistry definitions in the same simulation, you can create additional surface mechanisms when using the Surface-Gas Interaction model.

You can define the surface site and bulk species together with reactions manually. However, if the surface mechanism is involved, it is better to import it from Chemkin-formatted files.

If you choose to import surface site and bulk species manually, the database materials that are selected must satisfy the following requirements:

- For surface site materials, define the elemental composition and thermodynamic polynomial for specific heat.
- For surface bulk materials, define the elemental composition.

Note: Manual definition of chemistry can lead to reactions with unrealistic reaction rates, causing the ODE solver to fail. To identify why the solver failed, activate the checkbox of the *verbose on error* property. Details about the solver failure appear in the *Output* window during subsequent iterations.

Note: You can use open site formalism for surface chemistry mechanisms in STAR-CCM+. For the surface site species to be defined as open (have no elemental composition and molecular mass of 0), the open site species name must include `OPEN` in its name. For more details, please refer to the Chemkin manuals.

4. If the Eddy Break-Up model or Complex Chemistry model is selected, define the parameters of the Reacting Species physics model:
   - For the Complex Chemistry model, define the Properties of the Complex Chemistry model and its sub-nodes.
     1. Select the **Models** > **Complex Chemistry** node and set the required Properties.

        The ODE Solver is set to CVODE by default—which is appropriate for most simulations. The CVODE solver calculates reaction rates and solves complex chemistry problems in STAR-CCM+. If necessary, you can set the ODE Solver to DARS-CFD. The DARS-CFD solver translates the chemical mechanism to FORTRAN code, which can either be interpreted (ASCII) or compiled (Binary)
     2. Select the **Complex Chemistry** > **Chemistry Acceleration** node and if necessary, activate either *ISAT* or *Clustering*. Then define the Properties for ISAT or Clustering.
     3. When using the DARS-CFD ODE Solver, you can relax the species to their equilibrium compositions over a set time-scale. To set an equilibrium time scale, select the **Complex Chemistry** > **Approximation Options** node and activate *Equilibrium Time Scale*. Select the **Approximation Options** > **Equilibrium Time Scale** node and set the Properties.

     For more information, see Complex Chemistry Model Reference.
   - For the EBU model, define the EBU Properties.

- The EBU property, *Reaction Control*, is set to **Hybrid** by default—which is appropriate for most simulations. Hybrid specifies the reaction rate as the minimum of the rate predicted by turbulent mixing and the rate predicted by finite-rate chemical kinetics. However, if the turbulent mixing scale solely defines the rate, select **Standard EBU** instead.

See Eddy Break-Up Model Reference.

5. If the Eddy Break-Up model or Complex Chemistry model is selected, define the mechanism for the reactions that are occurring within the gaseous mixture.
   - To import larger reaction mechanisms, particularly when using the Complex Chemistry model, see Importing Species and Reactions.
   - To define species and reactions in STAR-CCM+ without importing a mechanism, see Defining Species Manually and Defining Chemical Reactions Manually.
6. Set the parameters of any other **[Continuum]** > **Models**, as required.
   - When using the Eddy Dissipation Concept model or the Laminar Flame Concept model, you can adjust the flame position by changing the **Multi-Component Gas** > **Material Properties** > **Turbulent Schmidt Number**.
7. For the continuum, define any necessary parameters for the **Reference Values** and **Initial Conditons**.
8. Define any necessary Physics Conditions and Values for the Region and Boundaries.
   - To modify the size of chemistry time-steps for steady-state simulations, modify the CFL Scaling Factor—available to specify as a physics value for regions which use the Complex Chemistry model and have the physics condition, *CFL Factor Option*, activated. See CFL Scaling Factor
9. Return to the Reacting Flow General Workflow.

Surface Chemistry Reference

This section provides reference material, including properties, sub-nodes, field functions, and other model-specific settings, for the Surface Chemistry models.

Contents:

Surface Chemistry Model Reference
Surface-Gas Interaction Model Reference
Reacting Surface Model Reference
Solid Composition Model Reference

Surface Chemistry Model Reference

You use the Surface Chemistry model to define surface reaction mechanisms and surface species on the reacting walls of solid regions, or surfaces in porous regions.

Surface Chemistry Model Reference

Theory	See Surface Chemistry Surface Chemistry
Provided By	**[physics continuum]** > **Models** > **Optional Models**

Example Node Path	**Continua** > **Physics 1** > **Models** > **Surface Chemistry**		
Requires	One of the following combinations of model selections:		
	Material: **Multi-Component Gas** *Reaction Regime*: **Reacting** *Reacting Flow Models*: **Reacting Species Transport** *Reacting Species Models*: **Complex Chemistry** or **Eddy Break-Up** *Flow*: **Segregated Flow** or **Coupled Flow**	*Material*: **Multi-Component Gas** *Reaction Regime*: **Non-Reacting** *Flow*: **Segregated Flow**	
Activates	Physics Models	After selecting the **Surface Chemistry** model when the **Eddy Break-Up** model or **Non-Reacting** model is selected, the **Surface-Gas Interaction** model is also selected. See Surface-Gas Interaction Model Reference.	
	Model Controls (child nodes)	**Surface Mechanism Manager** See Reacting Surface Model Reference.	
	Materials	Specify the gases under the **Multi-Component Gas** node and the reacting surface under the **[SurfaceMechanism]** > **Models** > **Solid Composition** node. See Materials and Methods.	
	Boundary Inputs	Only boundary regions that are designated as **Wall** can be reactive surfaces. See Boundary Settings.	
	Region Inputs	See Region Settings.	
	Field Functions	See Field Functions.	

Materials and Methods

Specify the gas components using the **Continua** > **Physics 1** > **Models** > **Multi-Component Gas** node. See Defining Species Manually.

Specify the reacting surface using the **Models** > **Surface Chemistry** > **Surface Mechanism Manager** > **[Mechanism]** > **Models** > **Solid Composition** node. See Solid Composition Model Reference.

Site Occupancy

Allows you to specify the number of surface sites, σ_j (in Eqn. (3332), Eqn. (3333), and Eqn. (3334)), that each of the adsorbed molecules (species) occupies. Available to set as a material property for each [Site Species] that is specified within the Solid Composition model.

Site Density

Allows you to specify the number of active sites, ρ_{site} (in Eqn. (3446)), that exist on the surface (kmol/m²). Available to set as a material property for the Site Solid Mixture within the Solid Composition model.

Bulk Density

Allows you to specify the number of surface sites on the solid surface that are occupied by bulk species, N_b in Eqn. (3324), (kmol/m²). Available to set as a material property for each [Bulk Species] that is specified within the Solid Composition model.

Boundary Settings

Wall

Surface Mechanism Option

To allow the wall boundary to become a reactive surface, select the **Physics Conditions** > **Surface Mechanism Option** sub-node of that boundary. Change the *Mechanism* property to one of the available mechanisms for your simulation. Mechanisms are available only after you import the complex chemistry definition or after you manually define the surface chemistry mechanism and surface species/reactions.

Mechanism	Corresponding Physics Value Nodes
[Surface Mechanism] Defines the wall boundary as a reacting surface.	**Surface Initial Bulk Activity Profiles** Allows you to set the surface molar fraction for species of the given mechanism on the reactive wall as a scalar array profile, B_i in Eqn. (3324). **Surface Initial Site Fraction Profiles** Allows you to set (as a scalar profile) the fraction of the total number of sites that a given species occupies, z_i in Eqn. (3446). This fraction only applies to species of the given mechanism on the interface between the reactive wall and gas phase. **Surface Washcoat Factor** Allows you to account for a porous washcoat layer which increases the active surface area of a solid catalyst surface. For example, on surfaces that are covered by a porous washcoat, you can specify a surface washcoat factor, γ_w in Eqn. (3444) and Eqn. (3445), greater than 1.0. A value of 1.0 represents the surface area before a washcoat is applied.
NONE Defines the wall boundary as a non-reacting surface.	

Region Settings

Porous

Surface Mechanism Option

To allow the internal surfaces within a porous region to become reactive surfaces, select the **Physics Conditions** > **Surface Mechanism Option** sub-node of that porous region. Change the *Mechanism* property to one of the available mechanisms for your simulation. Mechanisms are available only after you import the complex chemistry definition. On porous regions, surface chemistry is available when using Complex Chemistry with the DARS-CFD or CVODE ODE solver options.

Mechanism	Corresponding Physics Value Nodes
[Surface Mechanism] Defines the porous region as a reacting surface.	**Area to Volume Ratio** Specifies the surface area of the reacting/catalytic solid per unit volume. **Surface Initial Bulk Activity Profiles** The **Surface Initial Bulk Activity Profiles** > **[Bulk] Solid Mixture Activity** node allows you to set the surface molar fraction for species of the given mechanism on the reactive region as a scalar array profile. **Surface Initial Site Fraction Profiles** The **Surface Initial Site Fraction Profiles** > **[Site] Solid Mixture Fraction** node allows you to set (as a scalar profile) the fraction of the total number of sites that a given species occupies. This fraction only applies to species of the given mechanism on the interface between the reactive region and gas phase.
NONE Defines the porous region as a non-reacting.	

Field Functions

Surface Initial Bulk Fraction of [Species]

Surface Initial Bulk Growth Rate of [Species]

Surface Initial Bulk Source of [Species]

Surface Chemistry Energy Source

Surface Initial Site Fraction of [Species]

Surface Initial Site Source of [Species]

Surface Patch

Surface-Gas Interaction Model Reference

Surface-Gas Interaction Model Reference

Theory	See Surface Chemistry.	
Provided By	**[physics continuum]** > **Models** > *Surface Chemistry Reaction*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Surface-Gas Interaction**	
Requires	One of the following combinations:	
	Material: **Multi-Component Gas** *Flow*: any *Reaction Regime*: **Reacting** *Reacting Flow Models*: **Reacting Species Transport** *Reacting Species Models*: **Eddy Break-Up** *Flow*: any *Optional Models*: **Surface Chemistry**	*Material*: **Multi-Component Gas** *Reaction Regime*: **Non-Reacting** *Flow*: any *Optional Models*: **Surface Chemistry**
Properties	See Surface-Gas Interaction Model Properties.	

Surface-Gas Interaction Model Properties

Enabled Trigger

Never: The model is not activated. The *Begin* property is inactive if this option is selected.

Iteration: Activates the model in a steady-state case.

Timestep: Activates the model in an unsteady-state case.

Begin

Sets the number of iterations or time-steps when the model is activated.

verbose on error

Provides details about solver errors.

Activated: During iterations, this option generates maximum details in the *Output* window about solver errors.

Deactivated: Uses standard iteration output.

Absolute Tolerance

Absolute tolerance for the CVODE ODE solver.

Relative Tolerance

Relative tolerance for the CVODE ODE solver.

Reacting Surface Model Reference

Reacting Surface Model Reference

Theory	See Surface Chemistry Surface Chemistry	
Provided By	*Surface Mechanism Model Selection* dialog, *Enabled Models*	
Example Node Path	**Surface Chemistry** > **Surface Mechanism Manager** > **Surface Mechanism** > **Models** > **Reacting Surface**	
Requires	• Follow the instructions to select the Surface Chemistry Model. See Surface Chemistry Model Reference • Create a surface mechanism within the Surface Chemistry model.	
Properties	Key properties are: Non-Conservation On. See Reacting Surface Model Properties.	
Activates	Model Controls (child nodes)	**Reactions** See Importing Species and Reactions or Defining Chemical Reactions Manually.

Reacting Surface Model Properties

Non-Conservation On

When activated, STAR-CCM+ checks every reaction for the conservation of the number of sites, as well as stoichiometry. When deactivated, STAR-CCM+ checks every reaction for stoichiometry only.

Solid Composition Model Reference

Solid Composition Model Reference

Theory	See Surface Chemistry Surface Chemistry	
Provided By	*Surface Mechanism Model Selection* dialog, *Enabled Models*	
Example Node Path	**Surface Chemistry** > **Surface Mechanism Manager** > **Surface Mechanism** > **Models** > **Solid Composition**	
Requires	• Follow the instructions to select the Surface Chemistry Model. See Surface Chemistry Model Reference • Create a surface mechanism within the Surface Chemistry model.	
Activates	Model Controls (child nodes)	**Site Solid Mixture** and **Bulk Solid Mixture**

Site Solid Mixture

Provides the **Site Species** sub-node which allows you to specify surface species that occupy a surface site that is next to the gas. See Managing Mixture Components. You can also specify material properties that are specific to the Site Solid Mixture under the **Material Properties** sub-node.

Bulk Solid Mixture

Provides the **Bulk Species** node which allows you to specify species that are underneath the site layer. See Managing Mixture Components.

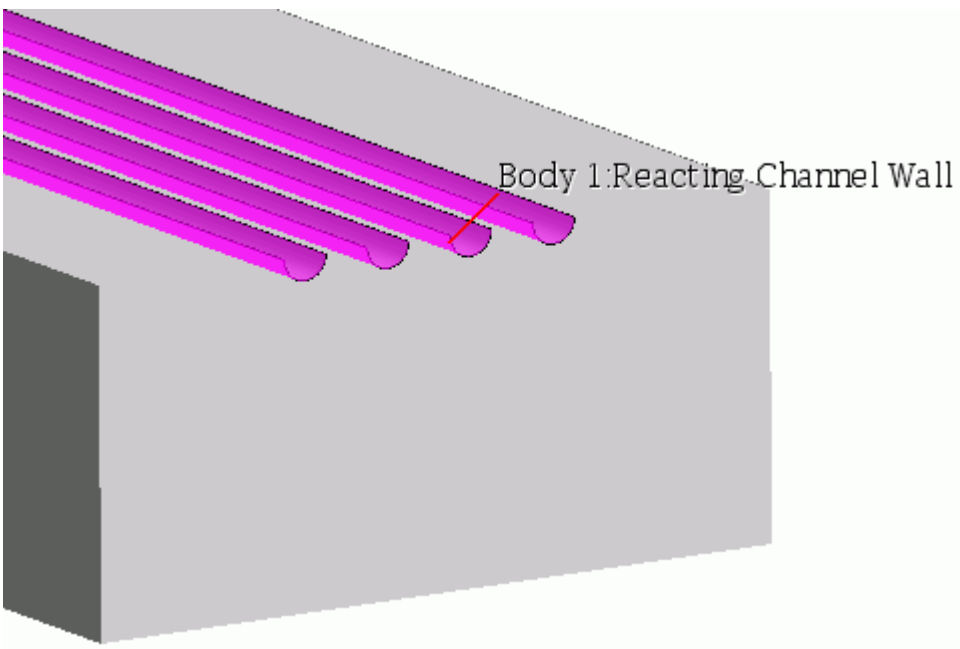
Reacting Channels

The Reacting Channel Coupling feature provides the ability for exchange of data between three-dimensional STAR-CCM+ and the one-dimensional Plug Flow Reactor in DARS-CFD. You can use the **Reacting Channel** co-simulation model in STAR-CCM+ to solve reacting flow simulations in tubular heat exchangers with channels that are long and thin.

When simulating Reacting Channels, you simulate combustion in the three-dimensional background phase which surrounds the reacting channels. The Reacting Species Transport model, Eddy Break-Up, is often the most appropriate model for simulating combustion in a firebox surrounding the reacting channels.

When the **Reacting Channel** model is selected along with the **Co-Simulation** model, a **Co-simulations** > **Reacting Channel Co-Simulation** node appears.

Several geometrical channels can represent one reacting channel as long as all of the geometrical channels have identical geometries of the same orientation. For geometrical channels that differ in length, diameter, or orientation, define each different group of geometrical channels as a separate reacting channel.



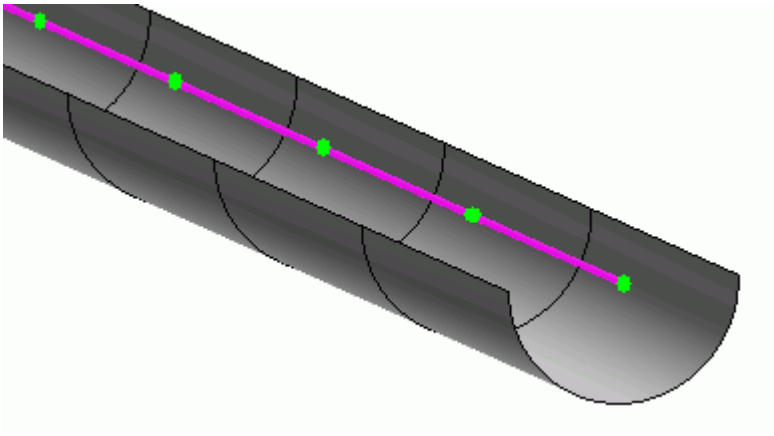
This functionality is useful for simulating chemical processes such as steam reforming and firebox combustion.

STAR-CCM+ meshes and solves the volume of the outer flow. The reacting channels are solved using a Plug Flow Reactor (PFR) approximation. The PFR is part of the DARS-CFD package. The reacting channels are not meshed and are considered 1D. However, the outer flow volume must account for the presence of the reacting channels in its geometry.

The coupling happens on the reacting channel wall boundary between the outer flow in STAR-CCM+ and the reacting channel in the PFR. The temperature is exchanged from the outer flow to the reacting channel and heat flux is returned back from the reacting channel to the outer flow. Although reacting channels appear as

3D features in a geometry, they become 1D representations when defined as reacting channels in the Reacting Channel Coupling model.

Since the reacting channels are one-dimensional and have no mesh, data is exchanged between axial points, whose frequency you specify, along a vector through the centre of the reacting channel and the surface boundary of the outer flow, as represented by the diagram below:



Note: It is imperative that all geometrical channels that are defined within a single reacting channel, are the same length and diameter with the same properties.

Reactions that occur in the reacting channel can consume or create heat energy. At specified time intervals, STAR-CCM+ sends the temperature along the geometrical channel wall to the Plug Flow Reactor (PFR). The PFR then solves for the reacting flow, determines the temperature within the reacting flow, and calculates the heat flux at each axial point in the reacting flow channel. The PFR then sends this data back to the outer flow in STAR-CCM+.

Contents:

Reacting Channels Workflow
Reacting Channel Co-Simulation Reference

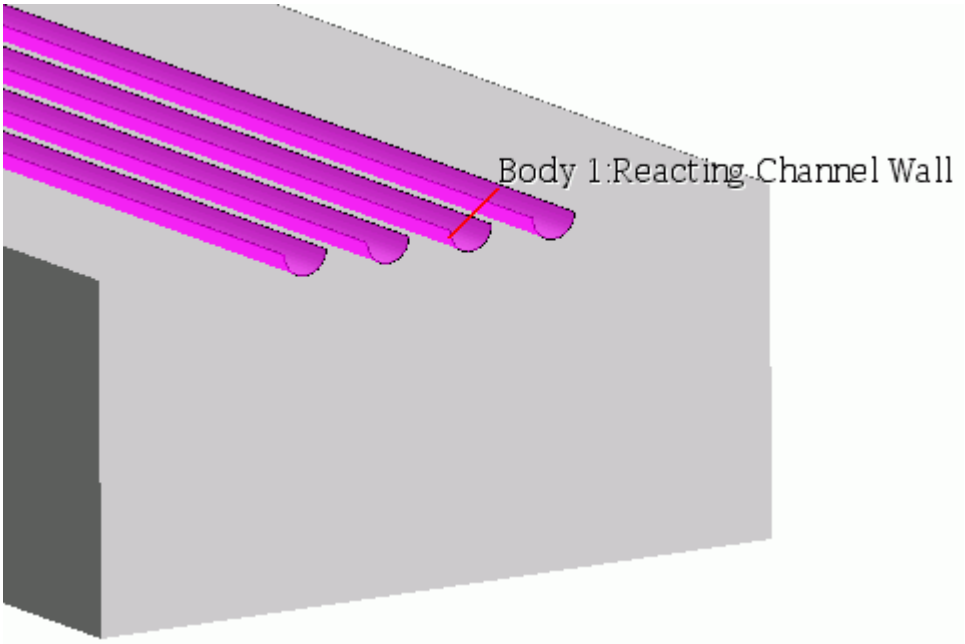
Reacting Channels Workflow

The **Reacting Channel** co-simulation model in STAR-CCM+ allows you to exchange data between reacting channels that use the DARS-CFD one-dimensional Plug Flow Reactor and three-dimensional STAR-CCM+.

Before setting up reacting channels, make sure that you have a fluid domain with a three-dimensional mesh that contains unmeshed voids that represent the reacting channels.

The steps in this workflow are intended to follow on from the initial steps in the Reacting Flow General Workflow.

When using the reacting channel coupling model, set up at least one reacting channel co-simulation zone. Each reacting channel can represent several geometrical channels.



1. For the physics continuum that represents the fluid domain that surrounds the reacting channels, select the following models—in addition to the models that are previously selected, with **Auto-Select recommended models** activated:

Group Box	Model
Reacting Flow Models	**Reacting Species Transport**
Reacting Species Models	• **Complex Chemistry** Only required for detailed reaction mechanisms and for situations in which you intend to model kinetic phenomena such as flame ignition, slowly forming pollutants, flame quenching, and extinction. • **Eddy Break Up** Suitable for simulating small reaction mechanisms for firebox-type regions which assume that reactions are limited by the rate that turbulence can mix the reactants and heat into the flame zone.

2. De-activate **Auto-Select recommended models** and select the following models:

Group Box	Model
Optional Models	**Co-Simulation**
Co-Simulation Models	**Reacting Channel**

3. Click **Close**.

   STAR-CCM+ adds the **Co-Simulations** node and a single **Reacting Channel Co-Simulation**.

Within one simulation, it is possible to create more than one reacting channel co-simulation. Separate reacting channel co-simulations are necessary for each reacting channel mechanism that is required. Each reacting channel co-simulation must contain at least one reacting channel zone. You can specify reaction conditions and values for each reacting channel co-simulation and also define further conditions and values that are specific to each reacting channel zone.

4. Create a **Reacting Channel Co-Simulation** node for each reacting channel mechanism that you want to use. To specify more than one reacting channel mechanism, right-click the **Co-Simulations** node and select **New**.
5. Specify the Conditions and Values for each Reacting Channel Co-Simulation.

   In particular, for each reacting channel co-simulation, import the reacting channel mechanism.

   a) Select the **Reacting Channel Co-Simulation** > **Conditions** > **PFR Library Input Option** node and specify the *Input Type* as **Binary** or **ASCII**, depending on the format of the input file.

   b) Expand the **Reacting Channel Co-Simulation** > **Values** > **PFR Library** node.

   c) In the *Properties* window, click the Custom Editor for *PFR Input* and in the dialog that appears, select the reaction mechanism file then click **OK**.

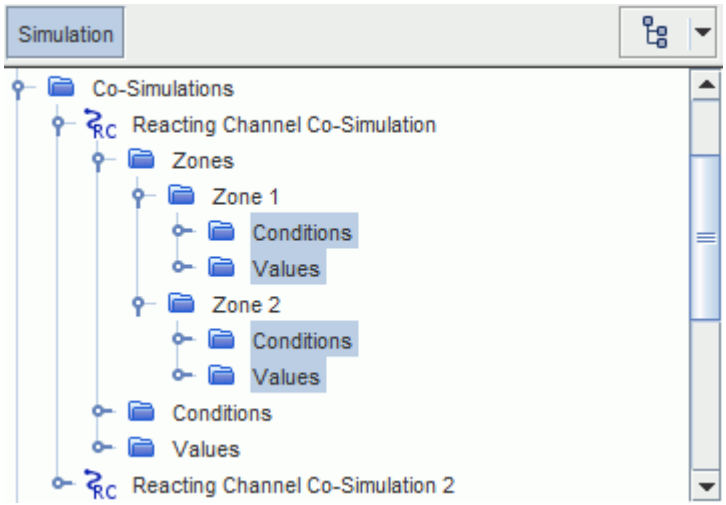
   d) Select the **Reacting Channel Co-Simulation** > **Values** > **Coupling Strategy** node and specify the parameters that are required for coupling.

   See: Reacting Channel Co-Simulation Reference.

The **Zones** node contains a **Zone 1** node by default.

6. To create more reacting channel zones, right-click the **Reacting Channel Co-Simulation** > **Zones** node and select **New**.

   A **Zone [2]** node appears.
7. Specify the Conditions and Values for each Zone within each Reacting Channel Co-Simulation.

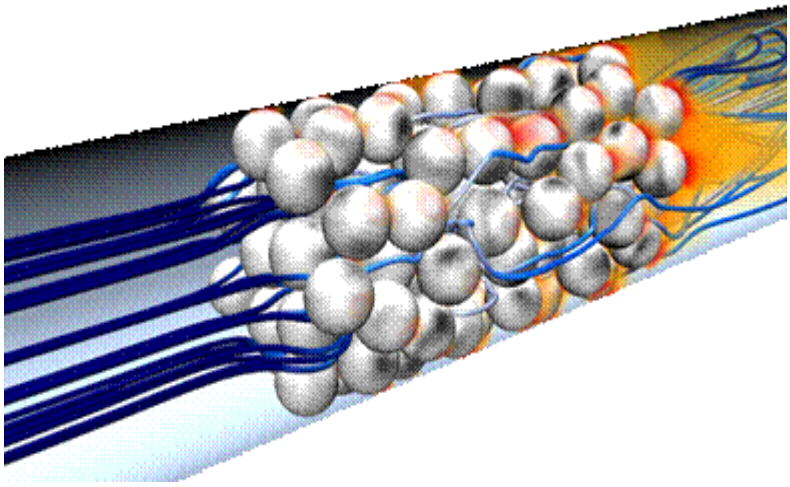


In particular:

a) Select the **[Zone]** > **Conditions** > **Reacting Channel Type** node and specify the *Correlation Type*.

- To simulate reacting channels that are void of any porous material, through which the reacting material will flow freely, select **Pipe**.

- To simulate reacting channels that are packed with porous material through which the reacting material will pass through, select **Packed Bed**. You can specify the parameters of the porous material.



b) Select the **[Zone]** > **Conditions** > **Heat Transfer Coefficient Correlation** node and choose the appropriate *Correlation Type*.

c) Set other **[Zone]** > **Conditions** as necessary.

Each zone represents one reacting channel and contains a set of conditions and values which are applied to the boundaries that you specify for that reacting channel zone.

8. To specify the boundaries for a reacting channel zone, group the boundaries for each reacting channel zone.
   a) Right-click the **Regions** > **[Region]** > **Boundaries** node and select **New Group**.
   b) Rename the **New Group** node appropriately.
   c) Expand the **Boundaries** node and multi-select the boundaries for one reacting channel zone.
   d) Drag the selected boundaries on to the **[New Group]** node.

   You can use the **[New Group]** node to edit the Properties, Conditions, and Values of all boundaries that it contains.
9. Right-click the **[New Group]** node and select **Edit**.
10. In the *Multiple Objects* dialog, expand the **Physics Conditions** > **External Code Coupling Specification** node and set *External Code Coupling Specification* to the Reacting Channel Co-Simulation Zone for the selected boundaries.
11. Group the remaining boundaries as necessary, and select an appropriate zone for the **External Code Coupling Specification** for each group of boundaries.

12. Set the parameters of any other **[Continuum]** > **Models**, as required.
13. For the continuum that represents the fluid domain which surrounds the reacting channels, define any necessary parameters for the **Reference Values** and **Initial Conditons**.

    Make sure that you specify the initial Species Mass Fraction.
14. Define any necessary Physics Conditions and Values for the Region and Boundaries.

- For the inlet boundaries, specify the Species Mass Fraction and the Mass Flow Rate.
- For the wall boundaries within the reacting flow region, specify the Surface Emissivity.

15. Return to the Reacting Flow General Workflow.

Reacting Channel Co-Simulation Reference

This co-simulation model is for reacting channels that use the DARS-CFD Plug Flow Reactor and STAR-CCM+.

You can use the Reacting Channel co-simulation model in STAR-CCM+ to solve reacting flow simulations in tubular heat exchangers with channels that are long and thin.

This functionality is useful for simulating chemical processes such as steam reforming and firebox combustion.

Reacting Channel Model Reference

Theory	See Reacting Channel Coupling Formulation.	
Provided By	**[physics continuum]** > **Models** > *Co-Simulation Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Reacting Channel**	
Requires	*Material*: **Multi-Component Gas** *Reaction Regime*: **Reacting** *Reacting Flow Models*: **Reacting Species Transport** *Reacting Species Models*: any *Space*: **Three Dimensional** *Time*: **Steady** *Flow*: any *Enthalpy*: any At this point deselect **Auto-select recommended models**. *Optional Models*: **Co-Simulation**	
Specific Right-Click Actions	See Specific Right-Click Actions.	
Activates	Physics Models	Model Groups: *Reacting Channel Coupling Models*
	Model Controls	When the Reacting Channel model is selected, a **Co-Simulations** > **Reacting Channel Co-Simulation** node is added to the simulation object tree. The **Reacting Channel Co-Simulation** node contains three sub-nodes which allow you to define the reacting channel zones, conditions, and values. See Zones, Conditions, and Values.
	Reference Values	Minimum Allowable Temperature, Maximum Allowable Temperature. See Reference Values.
	Boundary Inputs	External Code Coupling Specification. See Boundary Settings.

Zones

The Zones node contains a [Zone n] node for each zone that you specify. Each [Zone n] node contains two additional sub-nodes in which you specify the Conditions and Values of that specific zone.

Conditions

Coupled Model Parts Properties

Boundaries

Displays a read only value which contains the boundaries that are specified for that reacting channel zone.

Regions

Displays a read only value which contains the regions that are specified for that reacting channel zone.

Heat Transfer Coefficient Correlation Properties

Correlation Type

Allows you to specify the correlation type for the heat transfer coefficient:

- **Gnielinski** (Pipe)

  Determines the heat transfer coefficient using Eqn. (1392)
- **Leva/Grummer** (Packed Bed)

  Determines the heat transfer coefficient using Eqn. (1396)
- **Beek** (Packed Bed)

  Determines the heat transfer coefficient using Eqn. (1398)
- **DeWasch/Froment** (Packed Bed)

  Determines the heat transfer coefficient using Eqn. (1399)

Pipe Friction Correlation Properties

Correlation Type

Allows you to specify the correlation type for the pipe friction coefficient

- **Blasius** (Pipe)

  Determines the pipe friction coefficient using Eqn. (1400)
- **Filonenko** (Pipe)

  Determines the pipe friction coefficient using Eqn. (1401)
- **Ergun** (Packed Bed)

  Determines the pipe friction coefficient using Eqn. (1403) which is valid for simulations with low $\frac{\mathrm{Re}}{1-\chi}$ values.

- **Hicks** (Packed Bed)

  Determines the pipe friction coefficient using Eqn. (1402) which is valid for simulations with high $\frac{\mathrm{Re}}{1-\chi}$ values.

- **Ergun/Hicks** (Packed Bed)

  Determines the pipe friction coefficient using Ergun correlation (Eqn. (1403)) when $\frac{\mathrm{Re}}{1-\chi} < 300$, Hicks correlation (Eqn. (1402)) when $\frac{\mathrm{Re}}{1-\chi} > 500$, and linear interpolation in between.

Reacting Channel Heat Transfer Coefficient Option Properties

Reacting Channel Heat Transfer Coefficient Option

Allows you to specify the method for determining the Heat Transfer Coefficient, h_f which is used to determine the convective heat transfer source $\dot{Q}_C$ in Eqn. (1391).

- **Constant**
  Adds a **Values** > **Heat Transfer Coefficient** node which allows you to specify the Heat Transfer Coefficient, h_f as a constant array value.
- **Uniform**
  Allows you to specify the Heat Transfer Coefficient using values which are taken directly from the reacting channel Inlet values.
- **Non-Uniform**
  Determines the heat transfer coefficient using data from each axial point.
- **Tabular**
  Adds a **Values** > **Heat Transfer Coefficient** node which allows you to specify the Heat Transfer Coefficient using tabular data which you can import.

Reacting Channel Inlet Viscosity Option Properties

Option

Allows you to specify how the viscosity, μ, of the components entering the reacting channel is determined at the reacting channel inlet:

- **Constant**
  Adds a **Values** > **Inlet Viscosity** node which allows you to set a value for the gas Viscosity at the inlet.
- **Uniform**
  Allows you to specify the gas Viscosity at the inlet using values which are taken directly from the reacting channel Inlet values.

Reacting Channel Pipe Friction Factor Option Properties

Option

Allows you to specify a multiplying factor for the pipe friction, f which is used to calculate the Nusselt number Eqn. (1394) for cases with turbulent flow:

- **Constant**
  Allows you to specify a constant value for the **Values** > **Pipe Friction Factor** node.
- **Tabular**
  Allows you to specify the **Values** > **Pipe Friction Factor** using tabular data that you can import.

Reacting Channel Type Properties

Correlation Type

Allows you to specify the correlation type for the reacting channel:

- **Pipe**
  Specifies the pipes as empty reacting channels – which contain no material other than the gas that is in them.
- **Packed Bed**
  Specifies the pipes as fixed bed / packed bed reacting channels. This option is suitable for modeling applications such as steam cracking and methane steam reforming.

Reacting Channel Wall Temperature Option Properties

Option

Allows you specify how the Reacting Channel Wall Temperature, T_W, Eqn. (1391), is determined at the wall boundary:

- **Uniform**
  Determines the temperature using data that is averaged from all axial points throughout the reacting channel zone.
- **Non-Uniform**
  Determines the temperature using data from each axial point throughout the reacting channel zone.

Values

The following Values are available to specify for each reacting channel zone:

Heat Transfer Coefficient

Available when the reacting channel correlation type is set to **Pipe**. The convective heat transfer coefficient, h_f, is available to specify as:

- a constant value when **Reacting Channel Heat Transfer Coefficient Option** is set to **Constant**
- tabular data when **Reacting Channel Heat Transfer Coefficient Option** is set to **Tabular**

This value is used to determine the convective heat transfer source, $\dot{Q}_C$, in Eqn. (1391)

Inlet Area

Specifies the area of one geometrical channel in a reacting channel zone (each geometrical channel within a reacting channel zone is expected to be of equivalent dimensions). This value is used to calculate the channel diameter D_C which is used to determine the convective heat transfer coefficient of the reacting channel, h_f in Eqn. (1392).

Inlet Mass Fractions

Allows you to specify the initial mass fraction profile of the components entering the reacting channel.

Inlet Pressure

Allows you to specify the pressure at the reacting channel inlet.

Inlet Temperature

Allows you to specify the reacting channel bulk temperature, T.

Inlet Velocity

Allows you to specify the velocity of the components entering the reacting channel inlet.

Inlet Viscosity

Available to specify when the **Reacting Channel Inlet Viscosity Option** is set to **Constant**. Allows you to specify the viscosity, μ , in Eqn. (1395) of the components entering the reacting channel inlet.

Packed Bed Heat Transfer Factor

Available when the reacting channel correlation type is set to **Packed Bed**. Allows you to specify the heat transfer factor, f_{htg} , which is used to calculate the heat transfer correlation for the packed bed reacting channel in Eqn. (1396) and Eqn. (1398).

Packed Bed Particle Diameter

Available when the reacting channel correlation type is set to **Packed Bed**. Allows you to specify the equivalent packed bed particle diameter, D_p , which is used to calculate the heat transfer correlation for the packed bed reacting channel in Eqn. (1396) and Eqn. (1398).

Packed Bed Porosity

Available when the reacting channel correlation type is set to **Packed Bed**. Allows you to specify the porosity, χ , of the material within the packed bed reacting channel.

Pipe Friction Factor

The pipe friction factor, f, is available to specify as:

- a constant value when **Reacting Channel Pipe Friction Factor** is set to **Constant**.
- tabular data when **Reacting Channel Pipe Friction Factor** is set to **Tabular**.

This value is used to determine the Nusselt number for cases with turbulent flow, in Eqn. (1394), which is used to calculate the convective heat transfer coefficient, h_f, in Eqn. (1392).

Pipe Length

Allows you to specify the length of one geometrical channel in the reacting channel zone.

Solver Settings

Allows you to set the reacting channel solver settings for the zone.

Properties:

Absolute Tolerance

Allows you to specify an absolute tolerance for the solver.

Relative Tolerance

Allows you to specify a relative tolerance for the Solver.

Minimum Time Step Size

Sets the minimum time step size of the solver.

Maximum Time Step Size

Sets the maximum time step size of the solver.

Limit Rates

When activated, the rates within the solver are limited to the physical solution only.

Maximum BDF Order

Specifies the maximum order of the backward differentiation formula (BDF) method which is used to solve the stiff differential equations by numerically integrating the ordinary differential equations (ODE).

Maximum Newton Steps

Specifies the maximum number of Newton steps that are taken to solve the non-linear equations at each timestep when the implicit backward differentiation formula (BDF) method is used.

Under-Relaxation Factor

Allows you to set an under-relaxation factor, a, which governs the extent to which the old solution is supplanted by the newly computed solution. The default value (1.0) means that no under-relaxation is used.

Wall Discretization

Allows STAR-CCM+ to receive a different temperature or heat flux value from each axial division that you specify which provides a gradient along the reacting channel in the specified orientation.

Conditions

PFR Library Input Option Properties

Input Type

Allows you to specify the type of table that imports the reacting channel plug flow reactor data:

- **ASCII**
  Set *Input type* to **ASCII** when the data that you import is in ASCII format.
- **Binary**
  Set *Input type* to **Binary** when the data that you import is in Binary format. Creates an extra Conditions sub-node called **PFR User Coding Option**.

PFR User Coding Option Properties

User Coding

Only available to set when the Input *Type* is set to **Binary**. Allows you to modify the reaction rates or species source terms that are imported as binary data using the Dars-CFD FORTRAN files.

- **None**
  Does not use user coding.
- **Calculate Source Term**
  Calculates the species source terms with the Dars-CFD FORTRAN file: `SetDARSGasSrcTerms.f`
- **Modify Reaction Rates**
  Modifies the species reaction rates with the Dars-CFD FORTRAN file: `ModDARSGasReacRates.f`
- **Modify Source Term**
  Modifies the species source terms with the Dars-CFD FORTRAN file `ModDARSGasSrcTerms.f`

Reacting Channel Species Properties

PFR Species (read only)

Allows you to see all species that are involved in this mechanism in the reacting channel – reactants and products.

Values

Coupling Strategy Properties

Couple After Step

Specifies the time-step after which data is transferred from the Plug Flow Reactor (PFR)to STAR-CCM+. Delaying the data transfer until a specified time step allows the solver to reach a level of stability while receiving data from the PFR, before sending data back to the PFR . Set to 0 as default.

Coupling Frequency

Specifies the interval of time steps that occur between each transfer of data from the PFR to STAR-CCM+. Set to 5 by default.

PFR Library Properties

PFR Input

Specifies the location of the PFR files that are used for the *PFR Library Input Option* Condition.

Files Loaded

Appears as activated and read-only when PFR files are loaded.

Library Loaded

Appears as a read-only option which appears activated when a PFR library is loaded.

Specific Right-Click Actions

Right-clicking the **Reacting Channel Co-Simulation** > **Values** > **PFR Library** node provides the following options:

Export Complex Chemistry Definition Input

Allows you to save the complex chemistry definition for the reacting channel as a .txt file by default.

Delete Complex Chemistry Case

Deletes the complex chemistry definition from the reacting channel.

Reference Values

Minimum Allowable Temperature

The smallest temperature value that is permitted anywhere in the continuum.

The Energy models (Coupled Energy, Coupled Solid Energy, Segregated Solid Energy, Segregated Fluid Enthalpy, Segregated Fluid Temperature) limit temperature corrections such that the corrected value does not go below this minimum. If this occurs, a message is printed to the *Output* window.

Maximum Allowable Temperature

The largest temperature value that is permitted anywhere in the continuum.

The Energy models (Coupled Energy, Coupled Solid Energy, Segregated Solid Energy, Segregated Fluid Enthalpy, Segregated Fluid Temperature) limit temperature corrections such that the corrected value does not exceed this maximum. If this occurs, a message is printed to the *Output* window.

Boundary Settings

Domain Boundary

External Code Coupling Specification

Allows you to assign the boundary to a co-simulation zone for coupling. During the co-simulation, the STAR-CCM+ boundary exchanges data with the corresponding coupled surface in the partner simulation, according to the co-simulation zone specifications.

Common Reacting Flow Actions

This section contains short sets of instructions that describe how to perform some of the actions which are common to several of the reacting flow detailed workflows.

Contents:

Importing Species and Reactions
Defining Species Manually
Defining Chemical Reactions Manually

Importing Species and Reactions

STAR-CCM+ allows you to import mechanisms in Chemkin or DARS-CFD format. You can also import species and reactions separately in Chemkin format.

Objective	Steps
Importing Mixture Components from a Chemkin-formatted (`.chm`) File	1. Right-click the **[Continuum]** > **Models** > **Multi-Component [Gas]** > **[Gas] Components** node and select **Import Species**. 2. In the *Select file and material database* dialog that appears, click **Browse**. 3. In the *Open* dialog, navigate to and select the required file. Click **Open**. 4. In the *Select file and material database* dialog, click **OK**. The **[Gas] Components** node is populated with species data.
Importing Chemical Reactions from a Chemkin-formatted (`.chm`) File	1. Right-click the **[Continuum]** > **Models** > **Reactions** node and select **Import Reactions from a File**. 2. In the *Open* dialog, navigate to and select the required file. Click **Open**. The **Reactions** and **Gas Components** nodes are populated with reaction and species data, respectively.
Importing Mechanisms from a DARS-CFD-Formatted File	If *Run Time Compile* is activated, make sure that case-specific F90 files (*.f90 and `*.txt`) exist. If these files were created by STAR-CCM+, they are in a directory that is called `DarsInputFiles`. If *Run Time Compile* is deactivated, make sure that you select a folder that contains the actual case-specific pre-compiled library. 1. Right-click the **Complex Chemistry** > **DARS-CFD** node and select **Import Complex Chemistry Definition (DARS-CFD format)** 2. In the *Open* dialog, navigate to and select the `DarsInputFiles` directory and click **Open**. If the *Run Time Compile* expert property is activated, STAR-CCM+ will then: • Call the Fortran compiler to compile the case-specific F90 files and package them into a case-specific DARS-CFD library. • Read in the case-specific DARS-CFD library.

Objective	Steps
Importing Mechanisms from a Chemkin-Formatted File	When importing kinetic mechanisms in Chemkin format, it is necessary to import the Chemkin reaction mechanism file along with the corresponding NASA thermodynamic file. Importing the associated transport file is optional. See Using Thermodynamic Polynomial Data. For laminar flame simulations, the molecular transport properties are important and therefore a transport file is generally imported. However, for turbulent flames where turbulent transport dominates over molecular transport, it is not necessary to import the molecular transport properties—constant laminar transport properties are sufficient. 1. Right-click the **Complex Chemistry** node and select **Import Complex Chemistry Definition (Chemkin Format)** 2. In the *Import Chemkin Files* dialog that appears: a. Select the *Fluid Chemistry Reaction Files* and the *Fluid Thermodynamic Properties Files* to import. b. If you are simulating a laminar flame and the molecular transport properties are important: • Make sure that the *Import Fluid Transport Properties File* option is activated. • Select the *Fluid Transport Properties Files* to import. c. If you are simulating turbulent flames, deactivate the *Import Fluid Transport Properties File* option. d. Click **OK**. 3. Expand the **[Continuum]** > **Models** node. • The gas components appear under the **Multi-Component Gas** > **Gas Components** node. • The reactions appear under the **Reacting** > **Reactions** node. • The polynomial data appears under the **Material Properties** > **Specific Heat** > **Thermodynamic Polynomial Data** node for each species.

Restrictions on Importing Reactions

There should be no space between the stoichiometric coefficient and reactant/product.

For example, this format is invalid:

CH4 + 2 O2 => CO2 + 2 H2O

The correct format is:

CH4 + 2O2 => CO2 + 2H2O

Restrictions on Importing Species

- Thermodynamical polynomials and molecular data also must be present for the imported species, either in the supplied files or in the user-defined files.
- You can import either 7-coefficient or 9-coefficient thermodynamic polynomial data, however, you cannot use a mixture of these formats within the same file.

- The species names that are used in the thermodynamic polynomial data must match the species names that are used in the chemical mechanism.
- The species phase in the user-defined thermdat.dbs file must be G(gas), L(liquid), or S(solid).

Restrictions on Importing Surface Mechanisms

- Only one site and one bulk phase per surface mechanism is allowed.
- Species in the site phase must have the suffix (S).
- Species in the bulk phase must have the suffix (B).

Defining Species Manually

Before you can select a species as a component in a chemical reaction, the species must be defined as a component under the material model.

The Standard Material Database is provided in STAR-CCM+ by default. This database contains a wide selection of species components. However, if necessary, you can create and save custom material databases. See Using Material Databases.

1. Right-click the material model node, for example, the **[Physics 1]** > **Models** > **Multi-Component Gas** node, and choose **Select Mixture Components**.
2. In the *Select Mixture Components* dialog, expand the **Material Databases** > **Standard** > **[Gases]** node.
   a) Select all of the species components that are present in the reactions.
   b) Click **Apply**, then click **Close**.

   The Gas or Liquid species components appear under the Material model node, for example, the **Multi-Component Gas** > **Gas Components** node.
3. If necessary, you can replace a species component. Right-click the **Multi-Component Gas** > **Gas Components** > **[Component]** node.
   a) Select **Replace with**.
   b) In the *Replace Mixture Component* dialog, select the component that is required.
   c) Click **OK**.
4. The order in which the mixture components are added is not important as all species are solved for. However, if you wish, you can reorder species components. Right-click the **Multi-Component Gas** > **Gas Components** node.
   a) Select **Reorder Mixture Components**.
   b) In the *Reorder Mixture Components* dialog, drag and drop the species into the order that is required.
   c) Click **OK**.
5. For each species component, set the Material Properties.
   a) For a species component, expand the **[Gas Components]** > **[Component]** > **Material Properties** node.
   b) Select a Material Property node and set the Method that is required. See Material Property Node Properties.
   c) Select the Material Property sub-node and set any properties that are required.
   d) Repeat these steps until all of the necessary properties are set for all species components.

Defining Chemical Reactions Manually

The **Models** > **Reacting** > **Reactions** node and allows you to define reactions which occur within the reacting flow.

Before defining chemical reactions manually, make sure that species are defined under the relevant materials node. See Defining Species Manually or Importing Species and Reactions.

In STAR-CCM+, you can define simple reacting flow chemistry definitions manually. You can define reaction components as reactants or products, however, it is not possible to define any intermediate reaction components manually. If it is important to define specific intermediate reaction components in the chemistry definition, you can import an externally compiled reaction mechanism. See Importing Species and Reactions.

1. To set up a new reacting flow chemistry definition, right-click the **Reactions** node and select **New Reaction**.

   A **[Reaction n]** sub-node appears.
2. Define the reaction components:
   a) Right-click the **[Reaction n]** > **Reactants** node and select **Add Reactant**.
   b) Select a reactant from the submenu.

      The menu displays components that are already defined as part of the mixture and not already selected as a reactant in this reaction.
   c) Select any further reactants for the reaction.
   d) Right-click the **[Reaction n]** > **Products** node and select **Add Product**.
   e) Select products from the submenu in the same manner as for the reactants.
3. Specify the properties of each reaction component.
   a) Set stoichiometric coefficients, ν_{ij}, that are appropriate to the reaction.

      Make sure that the stoichiometric coefficients represent the correct ratio of reactants and products as are defined in the balanced chemical equation.
   b) For the EBU model and surface chemistry reactions, specify a rate exponent, p_{ij}.

      This parameter is used to calculate chemical reaction rates from finite-rate kinetics—not required if the **Standard EBU** model is used for **Eddy Break-up**.
4. Repeat the steps above until all reactions that are required to describe the reacting flow chemistry definition are fully defined.
5. If necessary, define the overall properties for the reacting flow chemistry definition. Expand the **[Reaction n]** > **Properties** node.
   - When using the Eddy Break-Up (EBU) model, expand the **Properties** > **EBU Coefficients** node and specify the EBU rate coefficients.
   - When using other reacting flow models, expand the **Properties** > **Reaction Coefficient** node and specify the Arrhenius coefficients or User-defined reaction coefficient.

Modeling Multiphase Flow

Multiphase flow is a term which refers to the flow and interaction of several phases within the same system where distinct interfaces exist between the phases. The term ‘phase’ usually refers to the thermodynamic state of the matter: solid, liquid, or gas.

In modeling terms, a phase is defined in broader terms, and can be defined as a quantity of matter within a system that has its own physical properties to distinguish it from other phases within the system. For example:

- Liquids of different density
- Bubbles of different size
- Particles of different shape

Multiphase flows are different from multi-component flows. In multi-component flows, the different species are mixed at the molecular level. These species have the same convection velocity. In multiphase flows, the different phases are mixed at the macroscopic scale. These phases have different convection velocity. Many flows are multiphase multi-component flows.

Multiphase flows can be classified into two categories:

- Dispersed flows, such as bubbly, droplet, and particle flows
- Stratified flows, such as free surface flows, or annular film flow in pipes.

A phase is considered dispersed if it occupies disconnected regions of space—otherwise it is continuous.

STAR-CCM+ provides the following distinct models to meet the requirements of these two categories of flow:

- The Lagrangian Multiphase model
- The Fluid Film model
- The Discrete Element Model (DEM)
- The Eulerian Multiphase Mixture model
- The Multiphase Segregated Flow model
- The Dispersed Multiphase model (DMP)
- The Volume of Fluid model (VOF)

In mathematical terms, both the Volume of Fluid model and the Segregated Flow model utilize a Eulerian (as opposed to a Lagrangian) framework in their formulation. Therefore, setting up simulations for both these approaches involve use of distinct Eulerian phases.

Both the Multiphase Segregated model and the Volume of Fluid model use distinct Eulerian phases, as opposed to a Lagrangian phase and a dispersed phase. Therefore, a common framework has been developed to specify the phases for both models. This framework is called the Eulerian Multiphase model in STAR-CCM+.

A common framework has also been developed to specify the models that define how pairs of phases interact with each other. These models are known as phase interactions.

Contents:

What Are Multiphase Flows?
Using the Lagrangian Multiphase Model
Modeling Fluid Film
Using the Discrete Element Method
Using the Eulerian Multiphase Model
Eulerian Multiphase Mixture Model
Using the Multiphase Segregated Flow Model
Dispersed Multiphase Model
Using the VOF Multiphase Model
Modeling VOF Waves

What Are Multiphase Flows?

Multiphase flow is a term which refers to the flow and interaction of several phases within the same system where distinct interfaces exist between the phases. The term ‘phase’ usually refers to the thermodynamic state of the matter: solid, liquid, or gas.

In modeling terms, a phase is defined in broader terms, and can be defined as a quantity of matter within a system that has its own physical properties to distinguish it from other phases within the system. For example:

- Liquids of different density
- Bubbles of different size
- Particles of different shape

Multiphase flows are different from multi-component flows. In multi-component flows, the different species are mixed at the molecular level. These species have the same convection velocity. In multiphase flows, the different phases are mixed at the macroscopic scale. These phases have different convection velocity. Many flows are multiphase multi-component flows.

Multiphase flows can be classified into two categories:

- Dispersed flows, such as bubbly, droplet, and particle flows
- Stratified flows, such as free surface flows, or annular film flow in pipes.

A phase is considered dispersed if it occupies disconnected regions of space—otherwise it is continuous.

STAR-CCM+ provides the following distinct models to meet the requirements of these two categories of flow:

- The Lagrangian Multiphase model: this model solves the equation of motion for representative parcels of the dispersed phase as they pass through the system. It is intended for systems that consist mainly of a single continuous phase carrying a relatively small volume of discrete particles, droplets, or bubbles. It is suited where the interaction of the discrete phase with physical boundaries is important.
- The Fluid Film model: this model predicts the dynamic characteristics of wall films using boundary layer approximations and assumed velocity and temperature profiles across the depth of the film. Film transport is predicted using thin shells that lie across the surface of solid walls on which the film is formed.
- The Discrete Element Model (DEM): this model is an extension of the Lagrangian Multiphase model, but where individual particles are modeled rather than representative parcels, and where inter-particle contact forces are explicitly accounted for.
- The Eulerian Multiphase Mixture model: this model is a simplified multiphase model that can be used to model suspension-like multiphase flows. In this model, the computational efforts are reduced by assuming the suspension to be a homogeneous single-phase system.
- The Multiphase Segregated Flow model: this model is commonly known as the Eulerian Multiphase model in the literature, but that term has been given a wider significance in STAR-CCM+. The Multiphase Segregated Fluid model solves conservation equations for mass, momentum, and energy for each phase. Phase interaction models are provided to define the influence that one phase exerts upon another across the interfacial area between them.
- The Dispersed Multiphase Model (DMP): this model simulates dispersed phases in a Eulerian manner. The Dispersed Multiphase model combines aspects of both the Lagrangian Multiphase (LMP) model and

the Segregated (Eulerian) Multiphase (EMP) models. The DMP model and the Volume of Fluid (VOF) model can be activated in the same simulation.

- The Volume of Fluid (VOF) model: this model is provided for systems containing two or more immiscible fluid phases, where each phase constitutes a large structure within the system (such as typical free surface flows). This approach captures the movement of the interface between the fluid phases, and is often used for marine applications.

In mathematical terms, both the Volume of Fluid model and the Segregated Flow model utilize a Eulerian (as opposed to a Lagrangian) framework in their formulation. Therefore, setting up simulations for both these approaches involve use of distinct Eulerian phases.

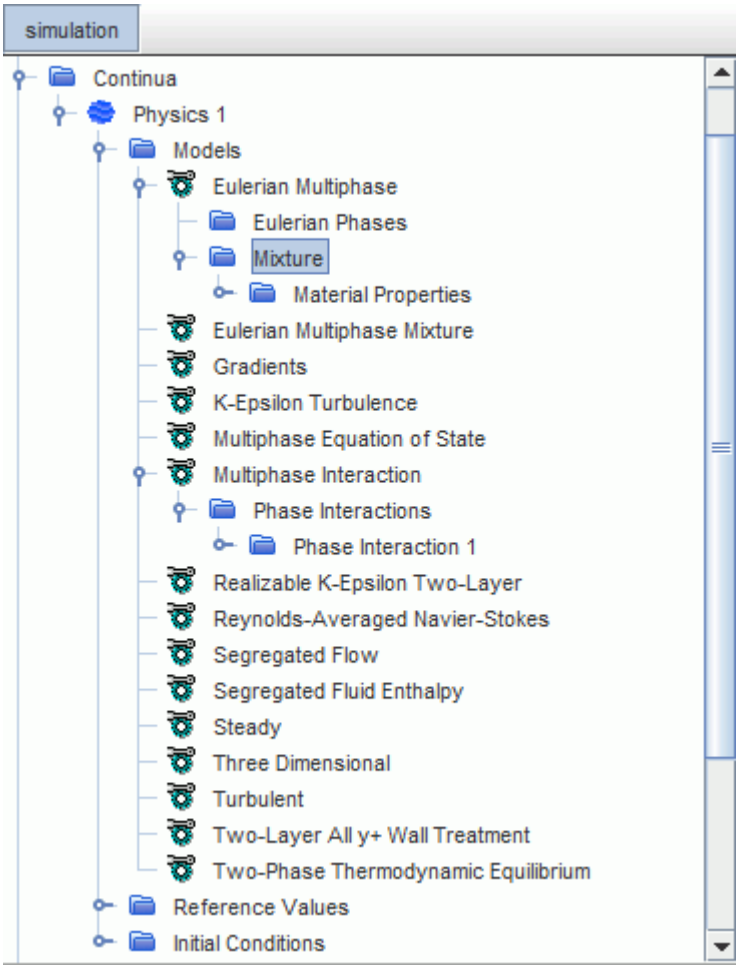
Contents:

What Is a Multiphase Mixture?
What Is a Phase?
Defining Phase Interactions

What Is a Multiphase Mixture?

A multiphase mixture is a fluid that is composed of multiple phases.

The corresponding node (**Eulerian Multiphase** > **Mixture** in the following screenshot) is activated as soon as the option **Eulerian Multiphase** is selected in the *Physics Model Selection* window.



The Multiphase Interaction model is added automatically in the *Physics Model Selection* dialog when the Eulerian Multiphase model is selected. The Eulerian Multiphase model is used to define phases and mixture properties, and the Multiphase Interaction model is used to define phase interactions.

Once these models are activated, you can choose one of three approaches for modeling the multiphase flow:

- The Volume of Fluid (VOF) homogenous multiphase model, which assumes a common velocity, pressure and temperature field for all phases.
- The Multiphase Segregated Flow model which solves separate conservation equations for each phase (apart from the pressure field, which is common).
- The Eulerian Multiphase Mixture model, which can be used with the Coupled Flow model or the Segregated Flow model.

If you select the Volume of Fluid or the Eulerian Multiphase Mixture model, the **Eulerian Multiphase** > **Mixture** node allows you to specify the bulk properties of the mixture. When using the Multiphase Segregated Flow approach, there are no settings to define under the **Eulerian Multiphase** > **Mixture** node.

Note: For Volume of Fluid simulations, the calculation method for key properties such as *Dynamic Viscosity* is set by default to **Volume Weighted Mixture** and you do not normally change it.

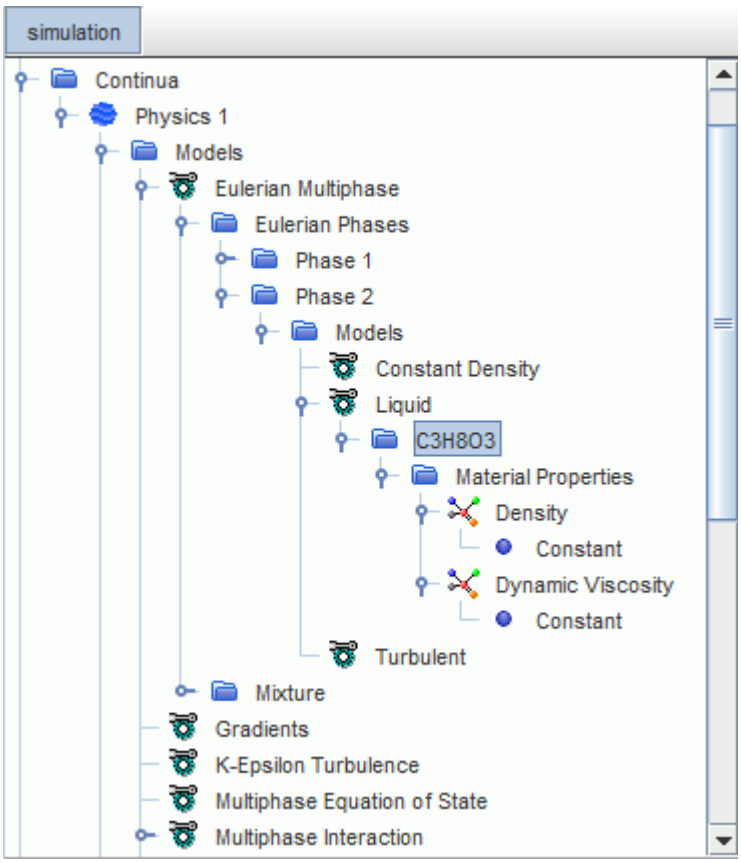
Overview of Phases and Phase Conditions

A Phase is matter whose properties continuously change in space. Some physics models expose their own region or boundary conditions for the phase to which they apply. These conditions are Phase Conditions.

What Is a Phase?

Typically the word “properties” brings to mind the physical properties, and this phase definition is relevant for the VOF multiphase model. However a generalization of the concept of phases also includes “properties” like velocity, temperature, or characteristic length scale of the matter. For example, if a problem involves spherical water droplets which are very close to each other and have different diameters, and/or move with different velocities, and/or have different temperatures, they could also be considered different phases.

STAR-CCM+ uses the Eulerian Multiphase model to manage materials and equation of state specifications in multiphase mixtures.

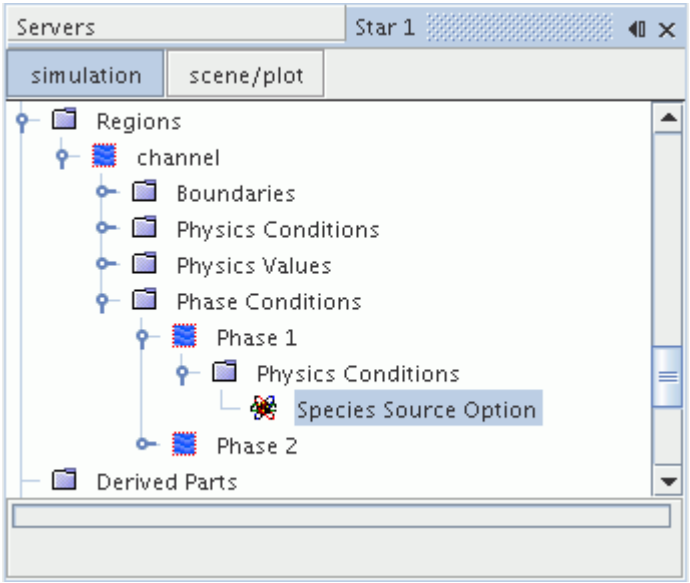


This phase is not the same as a Lagrangian phase, which is restricted to dispersed multiphase flows that are modeled using the Lagrangian Multiphase model.

What Are Phase Conditions?

Some physics models expose their own region or boundary conditions for the phase to which they apply. In this case, a **Phase Conditions** manager node is added to the regions and boundaries for which the phase is active.

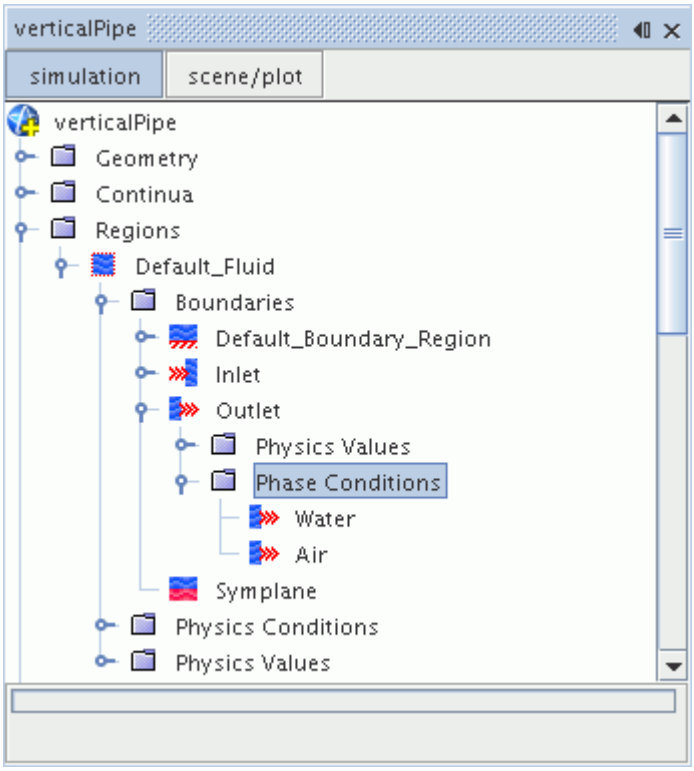
For example, when you select a multi-component liquid in a Eulerian phase, STAR-CCM+ adds the **Species Source Option** within a phase node for regions that are associated with the phase.



In some circumstances, you may need to define different boundary conditions for a specific phase. To set these conditions, phase conditions are applied at the boundary level and are used to define phase-specific attributes. For example, in a Multiphase Segregated Flow case you can specify that a wall boundary is permeable for a particular phase, while all other phases still see a wall. Similarly, in a Lagrangian Multiphase flow case, the settings that are used to define a Lagrangian phase in the physics continuum can be overridden at a particular boundary by specifying different settings at the boundary level.

For more information on boundary conditions, and how the conditions may differ between single-phase and multiphase flows, see Flow Boundaries Reference.

The **Phase Conditions** node has no properties and is subject to common menu items.



What Is a Free Surface?

The interface between two immiscible fluids is called a free surface.

The VOF multiphase model allows you to resolve the position and shape of the free surface, which is also important in evaluating surface tension effects.

Defining Phase Interactions

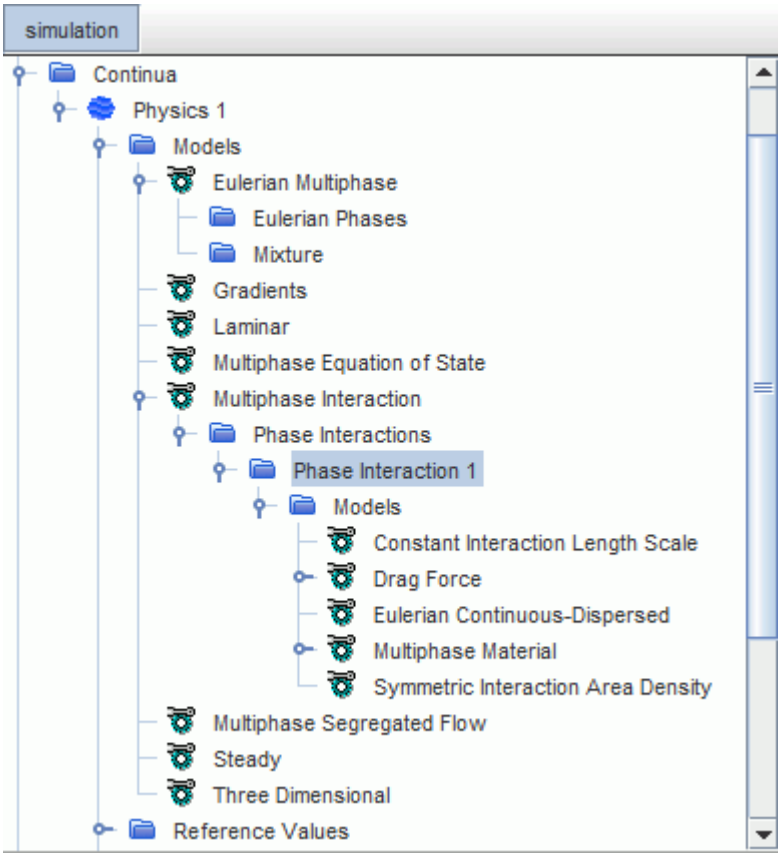
The phase interaction models available for selection depend on the physics models that are selected in the physics continuum. For further information on the models available, refer to the corresponding sections.

What Is a Phase Interaction?

A phase interaction is an object that defines the choice of models that are used to predict the influence of one phase upon another phase within a multiphase flow simulation. Define a phase interaction for every possible significant coupling.

Once created, phase interactions are stored under the **Phase Interactions** manager node which represents the process of how Eulerian phases interact with each other in STAR-CCM+. Each **Phase Interaction** node contains a **Models** manager node that is used to select phase interaction models for each phase pair. The phase interaction objects define the models that predict the interchange of conserved quantities between

phases in a multiphase simulation. The Multiphase Interaction model is selected automatically after the Eulerian Multiphase model.

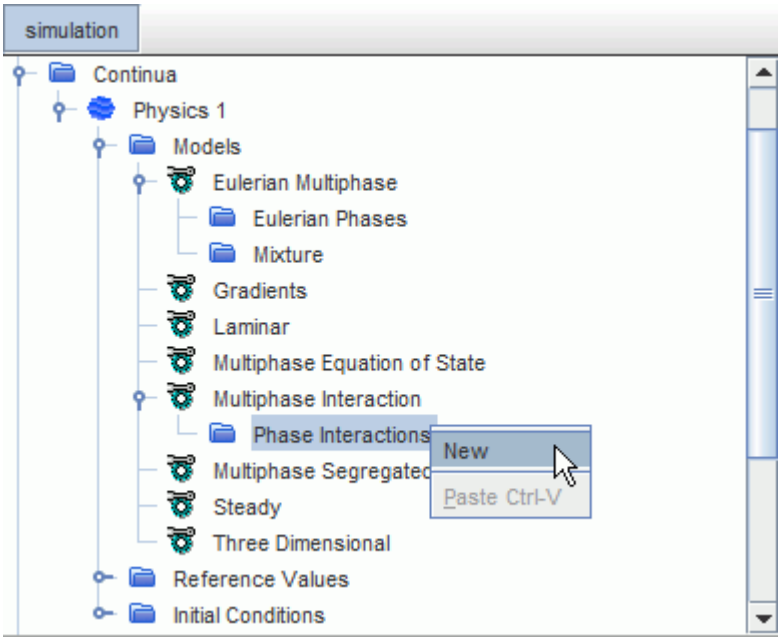


For Eulerian multiphase flows, these interactions effectively provide the closures that are required for the conservation equations in each phase. It is only necessary to find closures (and hence to define phase interactions) where significant phase interactions are expected. For example, it can be possible to neglect interactions between a number of phases that are in dilute concentration, while retaining interactions between each dilute phase and a dominant phase. Thus, for a simulation where all N phases are strongly interacting, it would be necessary to define all possible N(N-1)/2 pairs. For multiple dilute phases that are carried in one dominant continuous phase, however, it can be necessary to define only N-1 phase interactions.

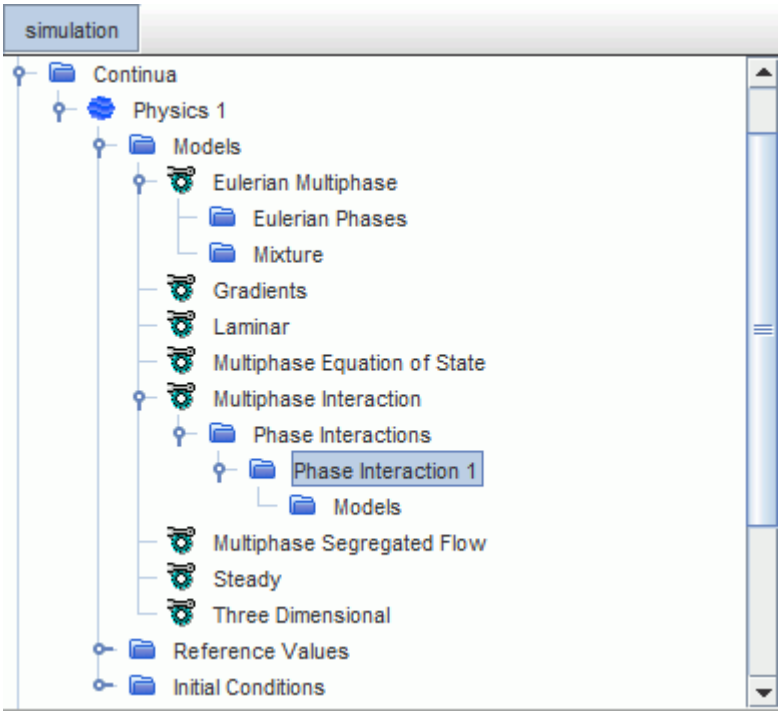
Standard phase interaction models are provided for the flow pattern where one dispersed phase (particles or bubbles) is carried in a continuous phase (liquid or gas). For modeling other kinds of multiphase flow patterns, you can define appropriate phase interaction models using field functions.

Creating a Phase Interaction

To create a phase interaction, right-click the **Phase Interactions** manager node and select **New**.



The new **Phase Interaction** node appears as a child of the **Phase Interactions** manager node, and contains a **Models** sub-node. It is given a default name that is based on the number of existing **Phase Interaction** nodes, if any.

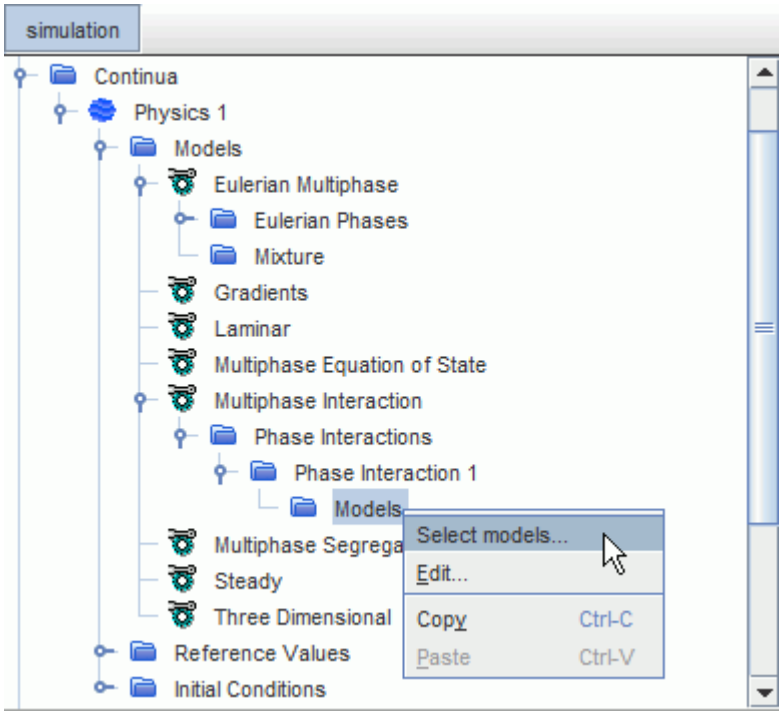


Selecting Phase Interaction Models

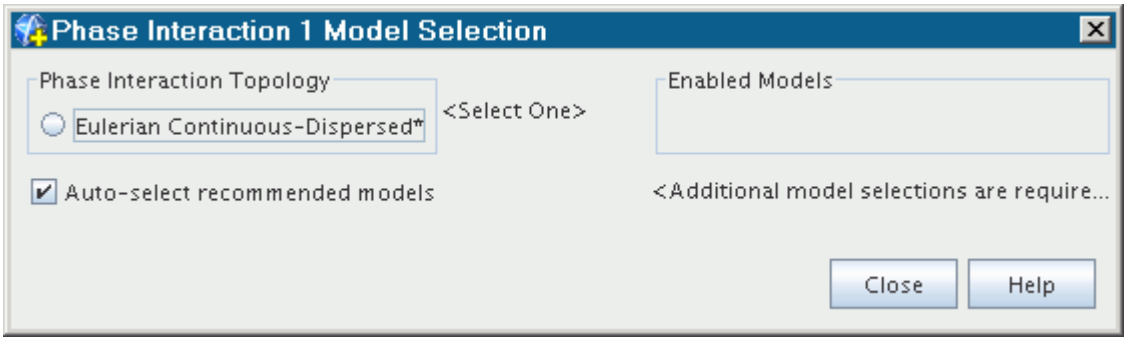
You select and manage phase interaction models for a simulation that uses the Multiphase Segregated Flow or the Volume of Fluid model.

To select models for a phase interaction, for example **Phase Interaction 1**:

1. Open the **Phase Interaction 1** node and right-click on the **Models** sub-node.
2. In the pop-up menu, click **Select Models...**



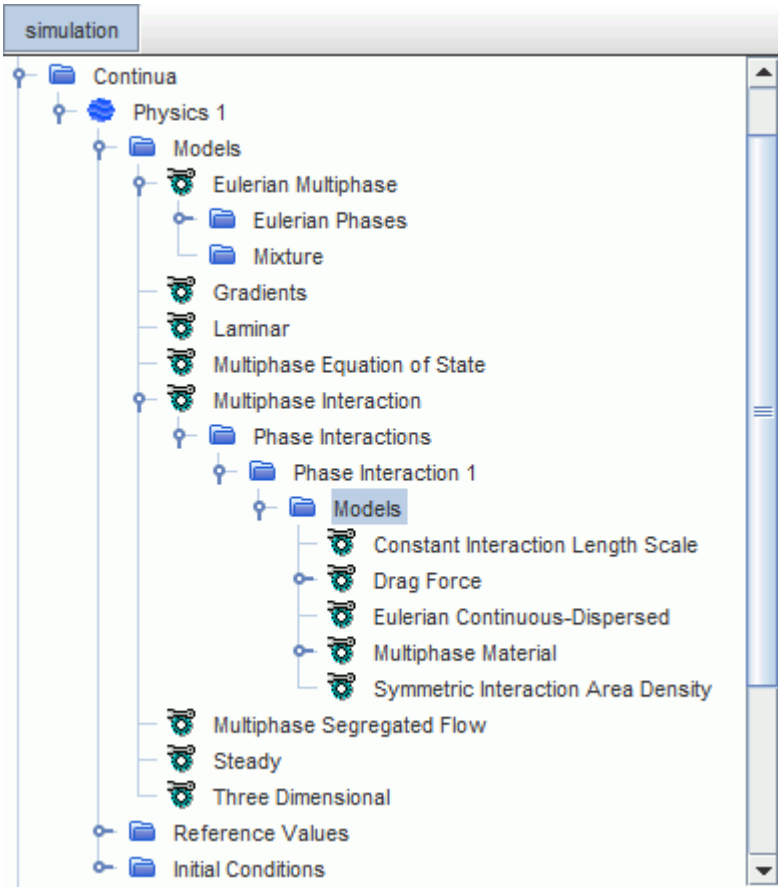
The appearance of the *Phase Interaction Model Selection* dialog when accessed for the first time depends on the type of multiphase flow you are modeling. For a multiphase segregated flow, it appears as follows:



Models that have been selected are shown in the Enabled Models list on the right. Models that are still available to be selected are in the various group boxes that appear on the left.

The dialog is “live” in that the server is updated each time that you change the state of a checkbox. If you have the relevant nodes open, the change is reflected in the tree. In the screenshot below, for example, you

have selected several models from the dialog, and their corresponding nodes are shown under the **Phase Interaction** parent node.



Using the Lagrangian Multiphase Model

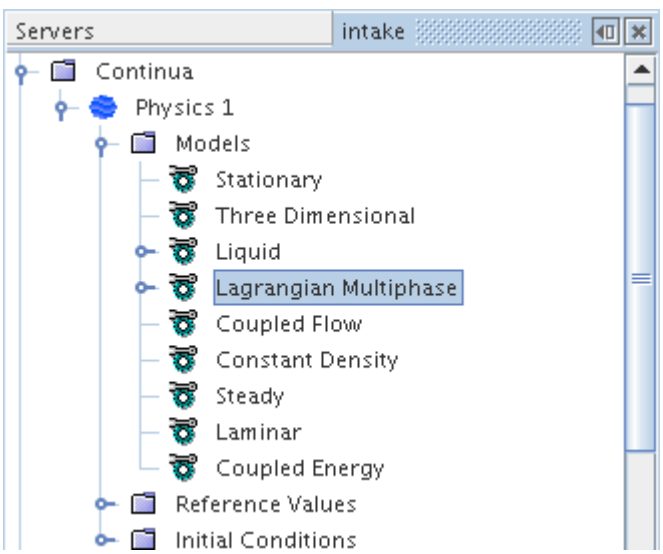
A wide variety of flow processes involve the transport of solid particles, liquid droplets, or gas bubbles—known as dispersed phases—by a gaseous or liquid continuous phase.

Examples include:

- Cyclone dust separators
- Vehicle soiling
- Spray coating
- Aerosol dispersion
- Spray cooling
- Liquid fuel combustion

The generic term particle is used in STAR-CCM+ to describe solid particles, liquid droplets, gas bubbles and massless (virtual) particles. If interactions of particles with the continuous phase, rather that with each other, dominate their motion, the dispersed phases are dilute. The Lagrangian Multiphase model is designed for such flows.

In STAR-CCM+, the **Lagrangian Multiphase** node is activated as soon as the model is selected in the *Physics Model Selection* dialog for a physics continuum. This node is shown below.



The documentation on the Lagrangian Multiphase model is divided into a number of sections.

Contents:

Introducing Lagrangian Multiphase Modeling
The Lagrangian Multiphase Workflow
Using the Lagrangian Multiphase Model with Time Models
Lagrangian Multiphase Model Reference

Lagrangian Phase Models
Working with Lagrangian Phase Boundary Conditions
Working with Primary Atomization
Working with Secondary Breakup
Working with Injectors
Working with Particle Tracks
Load Balancing for Lagrangian/Eulerian Flows
VOF-Lagrangian Impingement
Lagrangian/Eulerian Passive Scalar Interactions
Lagrangian Multiphase Field Functions Reference
Post-Processing Lagrangian Data

Introducing Lagrangian Multiphase Modeling

The physics continuum is a continuous phase whose governing equations are expressed in Eulerian form. The Lagrangian Multiphase model permits solving an arbitrary number of dispersed phases, each modeled in a Lagrangian framework.

In a Lagrangian framework [576], particle-like elements that are known as parcels are followed through the continuum. The state of each parcel is updated according to a selected set of models and can be optionally recorded as a track. Dispersed phases modeled using the Lagrangian Multiphase model are known as Lagrangian phases.

In the same way as models are selected for physics continua, the models that are provided to predict the state of each parcel can be selected per phase. Either (real) material particles or (virtual) massless particles can be modeled. Material models are available to represent single-component gases, single-, or multi-component liquids and single-component solids, together with constant density or polynomial density equations of state. The momentum conservation equation governs the flow of material particles and can optionally include drag, virtual mass, and user-defined forces. Similarly, an energy conservation equation can be solved, including convective heat transfer and an optional user-defined energy source. Specific models for droplets in a gaseous medium (“sprays”) include a mass conservation equation to account for the change in droplet mass from evaporation or condensation. They also include models for primary atomization and secondary breakup.

Boundary conditions define the interaction between particles and boundaries. Particle behavior at impermeable boundaries, in particular, requires modeling. You can control this behavior, depending on the active models. Particles can optionally be considered to cause erosion at these boundaries.

Injectors define the initial conditions of each particle, namely how and where they are introduced into the simulation.

In general, the state of the continuous phase influences the state of the dispersed phases. Two-way coupling can therefore be applicable, in which the state of the continuous phase depends on the dispersed phases, through inter-phase mass, momentum, and energy transfer effects.

If the flow of the continuous phase is laminar, a parcel that is released from a point at a given instant follows a smooth unique trajectory, that is, the motion is deterministic. On the other hand, parcels that are introduced into a turbulent carrier flow each have their own random path due to interaction with the fluctuating turbulent velocity field. A turbulent dispersion model is provided to account for this phenomenon.

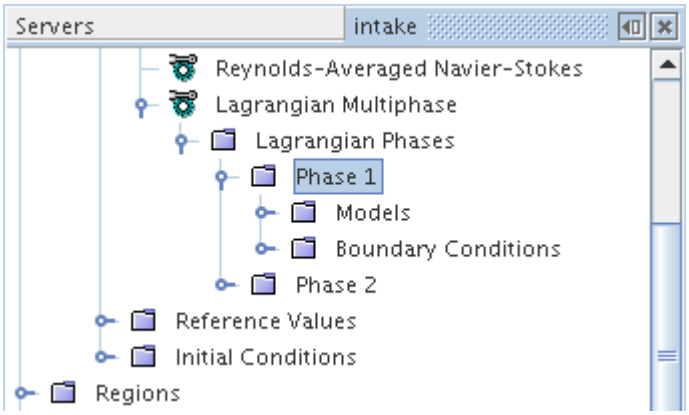
Contents:

What Is a Lagrangian Phase?
What Is a Parcel?
Setting Up a Lagrangian Multiphase Simulation

What Is a Lagrangian Phase?

A Lagrangian phase in STAR-CCM+ is a dispersed phase (droplets, bubbles, or particles) modeled in a Lagrangian framework.

The Lagrangian phase node, which has properties, represents this model. The Lagrangian phase node contains two manager nodes: a **Models** node that facilitates selection of phase models, and a **Boundary Conditions** node that facilitates setting of phase boundary conditions.



The definition of each Lagrangian phase includes the set of models applicable to its parcels. By definition, the same models govern particles in the same Lagrangian phase. Therefore, for example, particles of two different materials constitute two different Lagrangian phases.

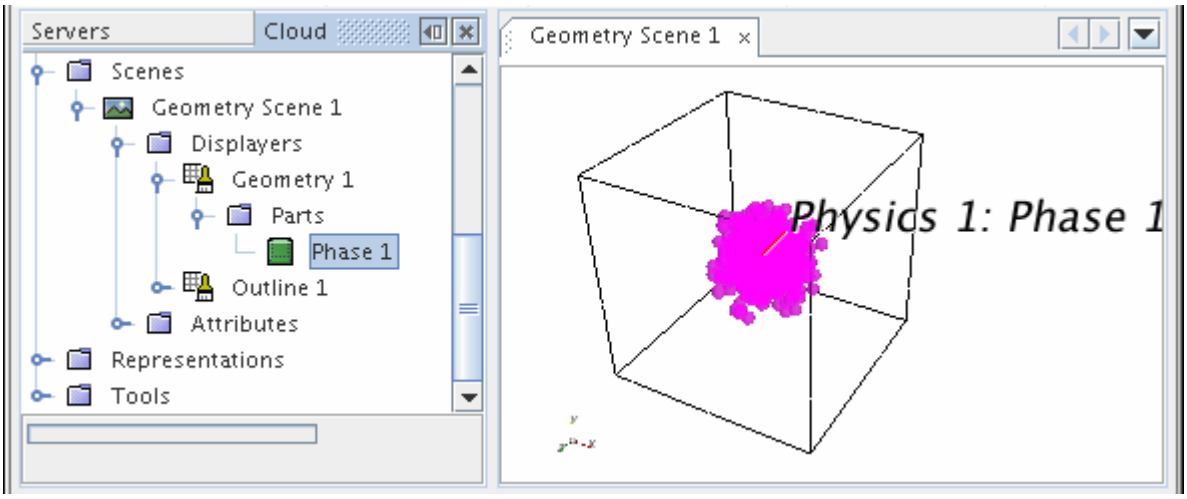
What Is a Parcel?

For flows involving a comparatively small number of dispersed phases, it is possible to formulate and solve governing equations for every droplet, bubble, or particle. However, if the number of dispersed phases is large, a statistical approach is more practical. In this approach, a smaller number of computational parcels represents the total population of dispersed phases.

Each parcel represents a localized group (cluster) of dispersed phases having the same properties. In effect, parcels are a discretization of the population of dispersed phases in the same way that cells are a discretization of continuous space. As with cells, the number of parcels is not arbitrary; it must be large enough so that the properties of the full population of dispersed phases are represented. This number can be assessed, in the absence of any other measures, by performing calculations with different numbers of parcels and comparing the results.

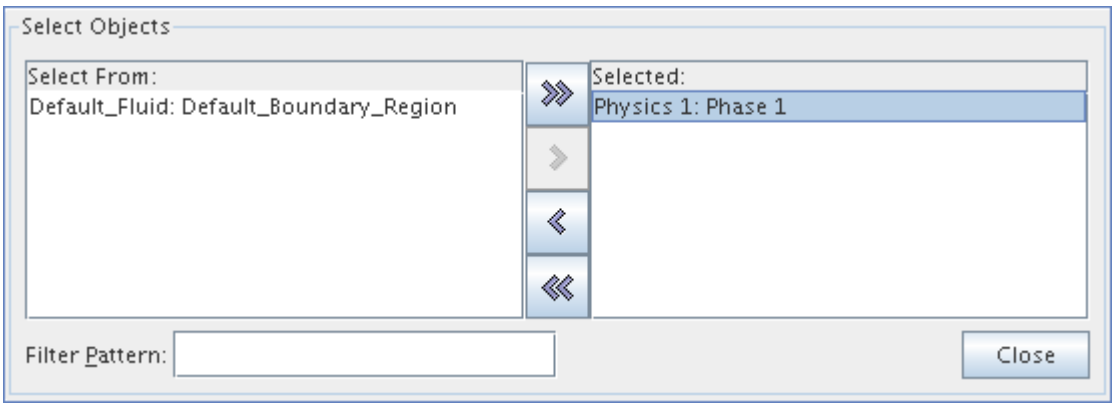
Visualizing Lagrangian Phase Parcels

It is possible to visualize the parcels of a Lagrangian phase in a scene.



Parcels are available to be visualized in this way provided they have not left the computational domain, which usually means only in unsteady simulations.

As with any other parts in a simulation, you can manage the parcels through the scene, or through an individual displayer as shown in the following screenshot.

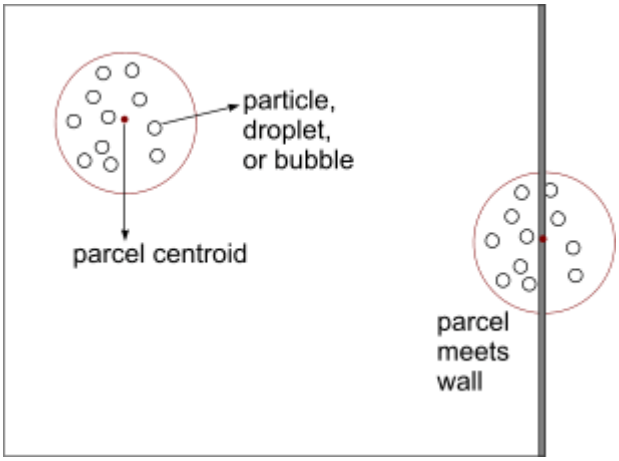


Using Lagrangian Phase Parcels

A parcel can also serve as an input part for certain derived parts: a cell surface, section, threshold, or iso surface. When a derived part has parcels that are specified as input parts, parcel fields can be visualized and analyzed on it, and parcel-specific reports can operate on it.

Parcel Interaction with Boundaries

A parcel centroid determines the intersection of a Lagrangian parcel with a boundary face. The parcel centroid is the center of mass of all the dispersed phases within the parcel.



STAR-CCM+ does not provide the distribution of the dispersed phases in a parcel.

If you require a higher level of resolution for dispersed phase, use DEM.

Setting Up a Lagrangian Multiphase Simulation

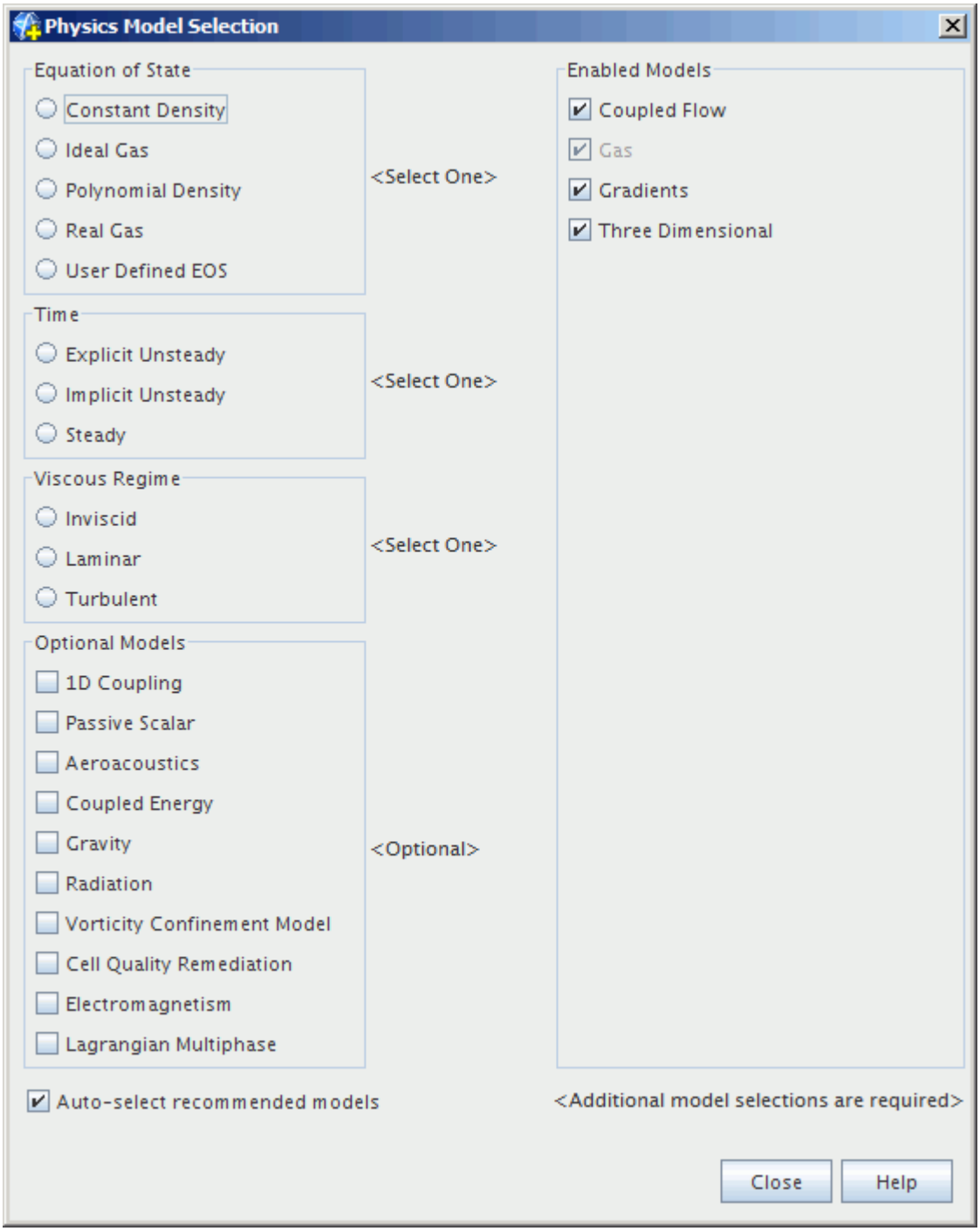
Several actions are required to set up a simulation that uses the Lagrangian Multiphase model.

These actions are outlined briefly in the following steps:

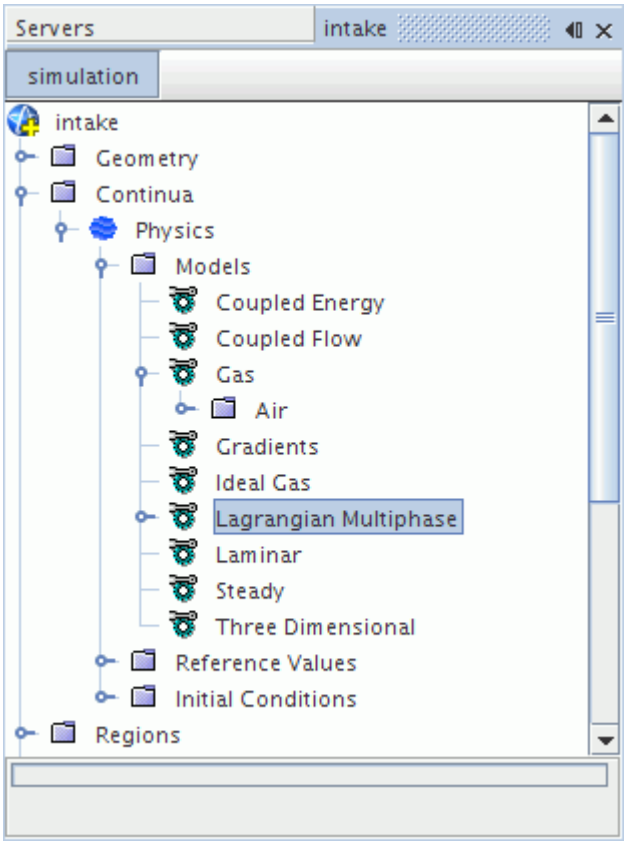
- Select the Lagrangian Multiphase Model
- Create one or more Lagrangian phases
- Select appropriate models for each Lagrangian phase
- Define material properties for each Lagrangian phase
- Define boundary conditions for each Lagrangian phase
- Create and define injectors
- Adjust parameters for the Lagrangian Multiphase solver

Selecting the Lagrangian Multiphase Model

The Lagrangian Multiphase model is activated from the *Physics Model Selection* dialog of the relevant physics continuum. It is necessary to first select a Liquid or Gas Material model in the dialog (which is accessed by right-clicking on the **Models** node of the physics continuum and choosing item **Select Models...**). Continue by selecting either the Coupled Flow model or the Segregated Flow model. The **Lagrangian Multiphase** option then appears in the *Optional Models* group box.



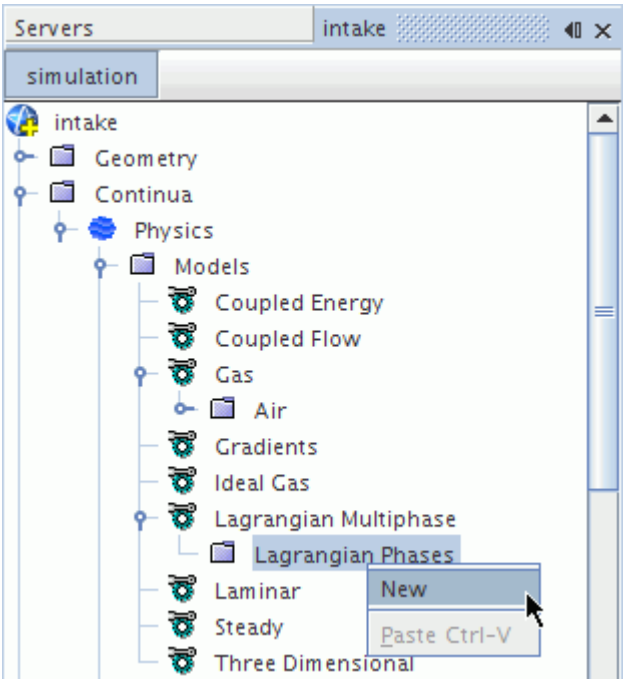
The remaining selections in this dialog depend on problem characteristics that are not related to dispersed phases. When the Lagrangian Multiphase option has been selected in the dialog, the corresponding model node becomes visible in the object tree.



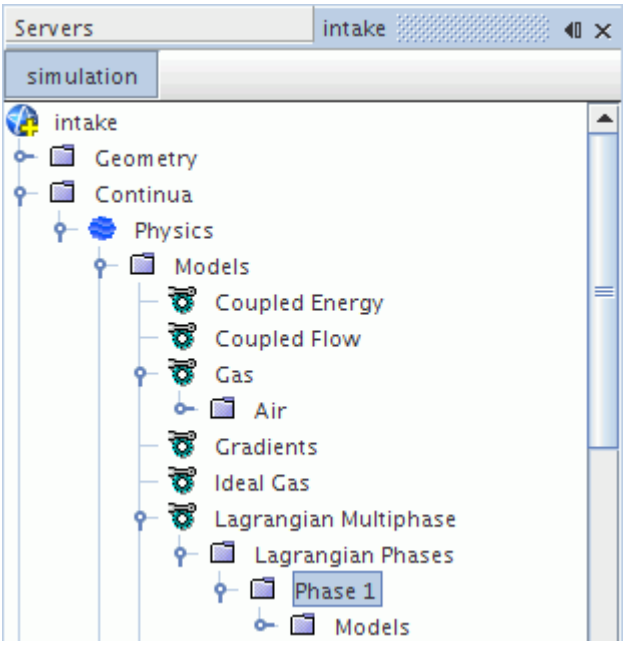
Creating One or More Lagrangian Phases

Having activated the Lagrangian Multiphase model itself, it is necessary to create one or more Lagrangian phases to represent each distinct dispersed phase within the simulation. To create a Lagrangian phase:

1. Open the **Models** > **Lagrangian Multiphase** model node. This action exposes the **Lagrangian Phases** manager node.
2. Right-click the **Lagrangian Phases** manager node and select **New** from the pop-up menu.



A new Lagrangian phase node appears with the default name, **Phase 1**.



The Lagrangian phase node can be renamed if desired. The Lagrangian phases manager can hold any number of phases up to the memory limits of the computer.

Lagrangian Phase Manager Pop-Up Menu

This pop-up menu item is available when you right-click the **Lagrangian Phases** manager node.

New	Creates a Lagrangian phase with a default name with a **Models** sub-node and a **Boundary Conditions** sub-node.

Copying a Lagrangian Phase

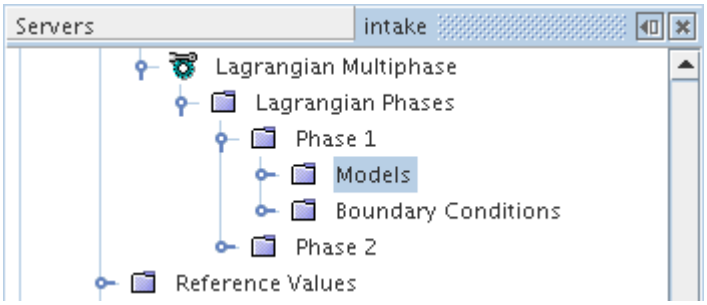
To copy the contents of a Lagrangian phase from one phase to another, use the clipboard or the drag-and-drop method. The copy action activates all the phase models from the source Lagrangian phase in the destination one, with the properties of each phase model also being copied.

This copy feature can be used between continua of the same simulation, provided the destination continuum already has the Lagrangian Multiphase model activated, with at least one Lagrangian phase created. Such copying is also possible between two different simulations, if both simulations have been opened in the same STAR-CCM+ session.

Additionally, you can right-click on the phase node and select **Copy**, and then right-click on the phase manager node and select **Paste**.

Selecting Appropriate Models for Each Lagrangian Phase

The **Models** manager node, which has its own pop-up menu, contains the nodes of all of the models in a Lagrangian phase.

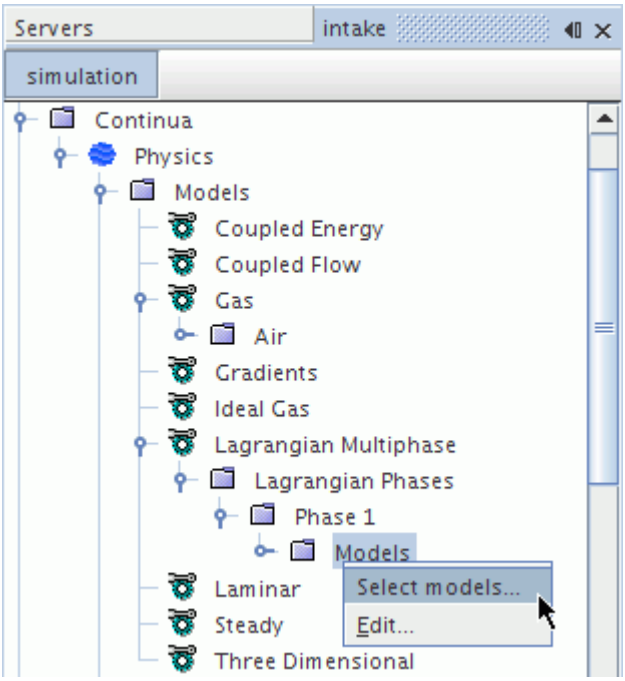


This section deals with selecting the models that are required to define a Lagrangian phase. The primary tool for this selection is the Phase Model Selection dialog, which guides you through the model selection process and assists with model recommendations.

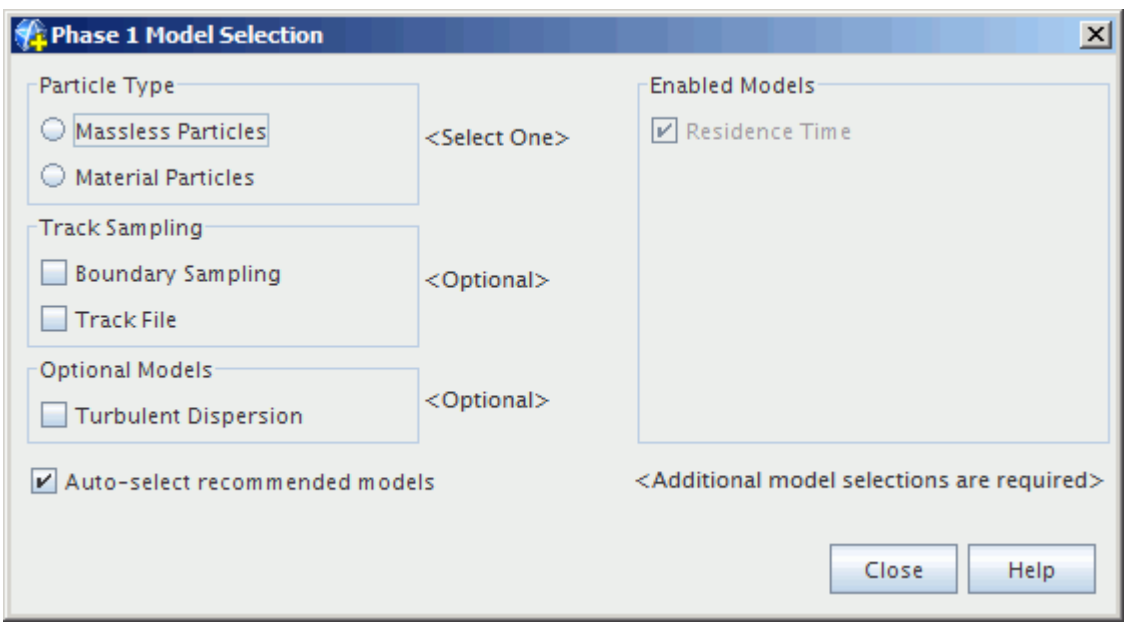
Lagrangian phase model selection is not identical to model selection for a continuum, since Lagrangian phases implicitly inherit certain model choices such as those for space, time and motion.

To select phase models for a Lagrangian phase:

- Right-click the **Models** node within the phase and choose **Select Models...** from the pop-up menu.



The *Phase Model Selection* dialog appears.



Lagrangian Phase Models Manager Pop-Up Menu

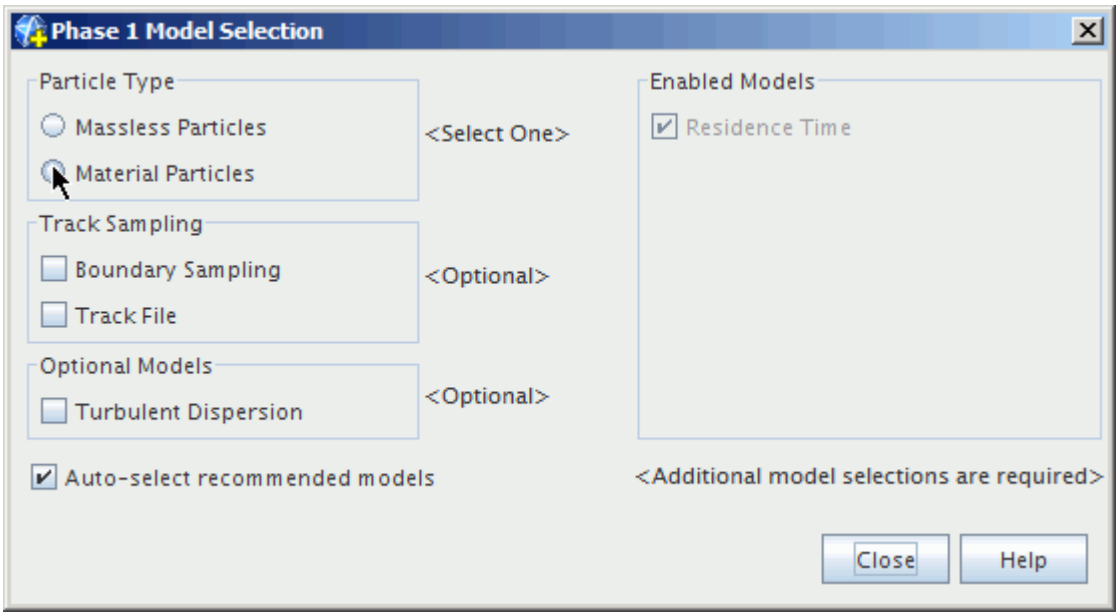
Select Models...	Activates selection of the Lagrangian phase models.

Using the Phase Model Selection Dialog

The *Phase Model Selection* dialog is used to select Lagrangian phase models, and change the selection in a simulation that uses Lagrangian Multiphase modeling. Lagrangian phase model selection and deletion can only be done through the dialog:

- This dialog is similar to a wizard in that it can assist you through a fairly complex process selection network.
- It is unlike a wizard in that it does not have the traditional **Next** and **Back** buttons.
- It does not restrict you to answering questions in a particular order.
- It has an auto select feature that adds recommended models automatically in response to a particular model selection by you. While this feature is on by default, you can deactivate it.

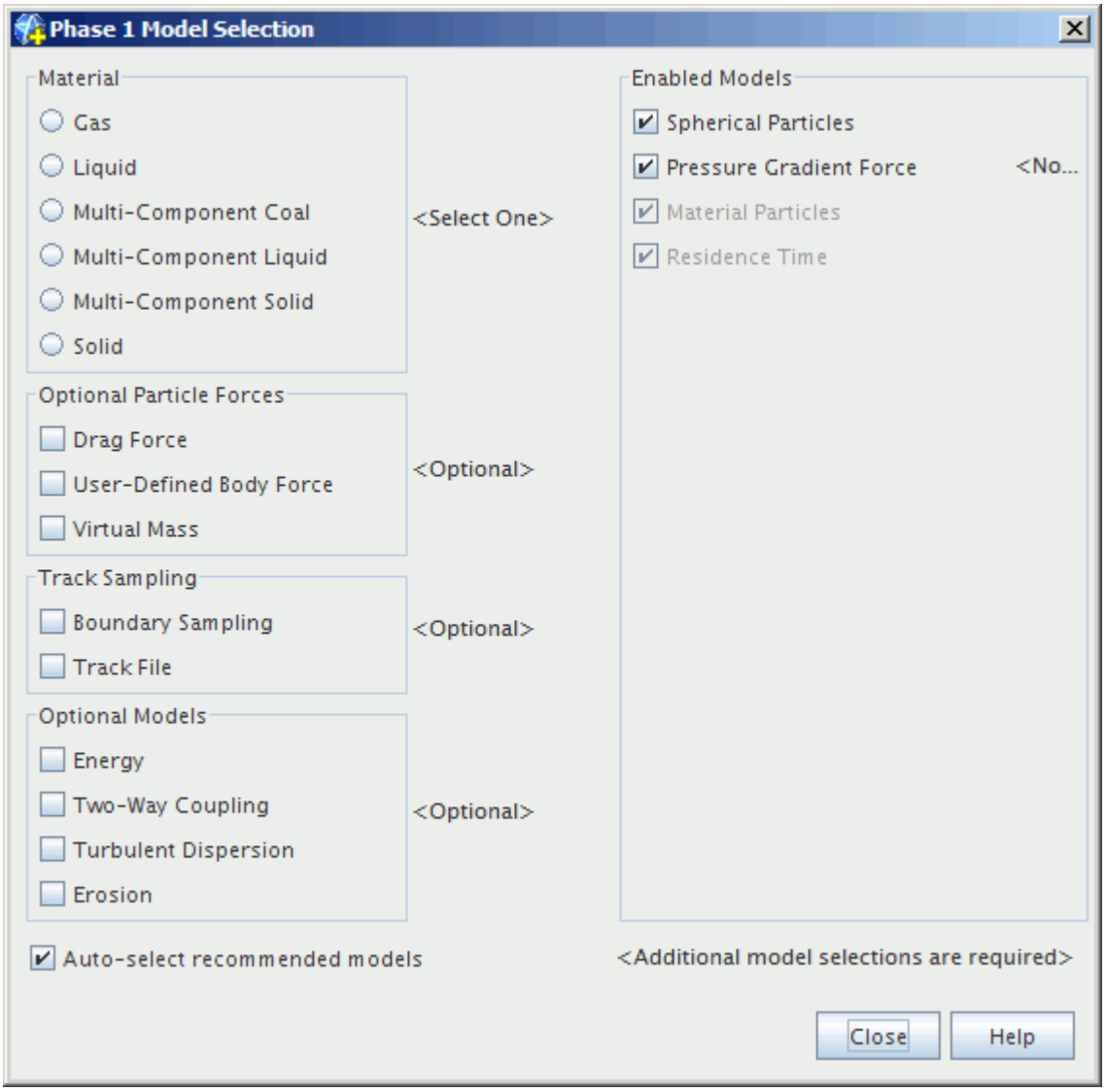
The *Phase Model Selection* dialog first appears as shown below for a new Lagrangian phase in a steady simulation.



For a steady simulation, **Residence Time** is selected automatically in the *Enabled Phase Models* group box. For an unsteady simulation, no initial selections are made.

Two types of model are available for selection: *Particle Type* and *Optional Models*. The <Optional> labels indicate that the selection of these items is not mandatory. The **Auto-select recommended Phase models** checkbox provides an option for the automatic selection of recommended models.

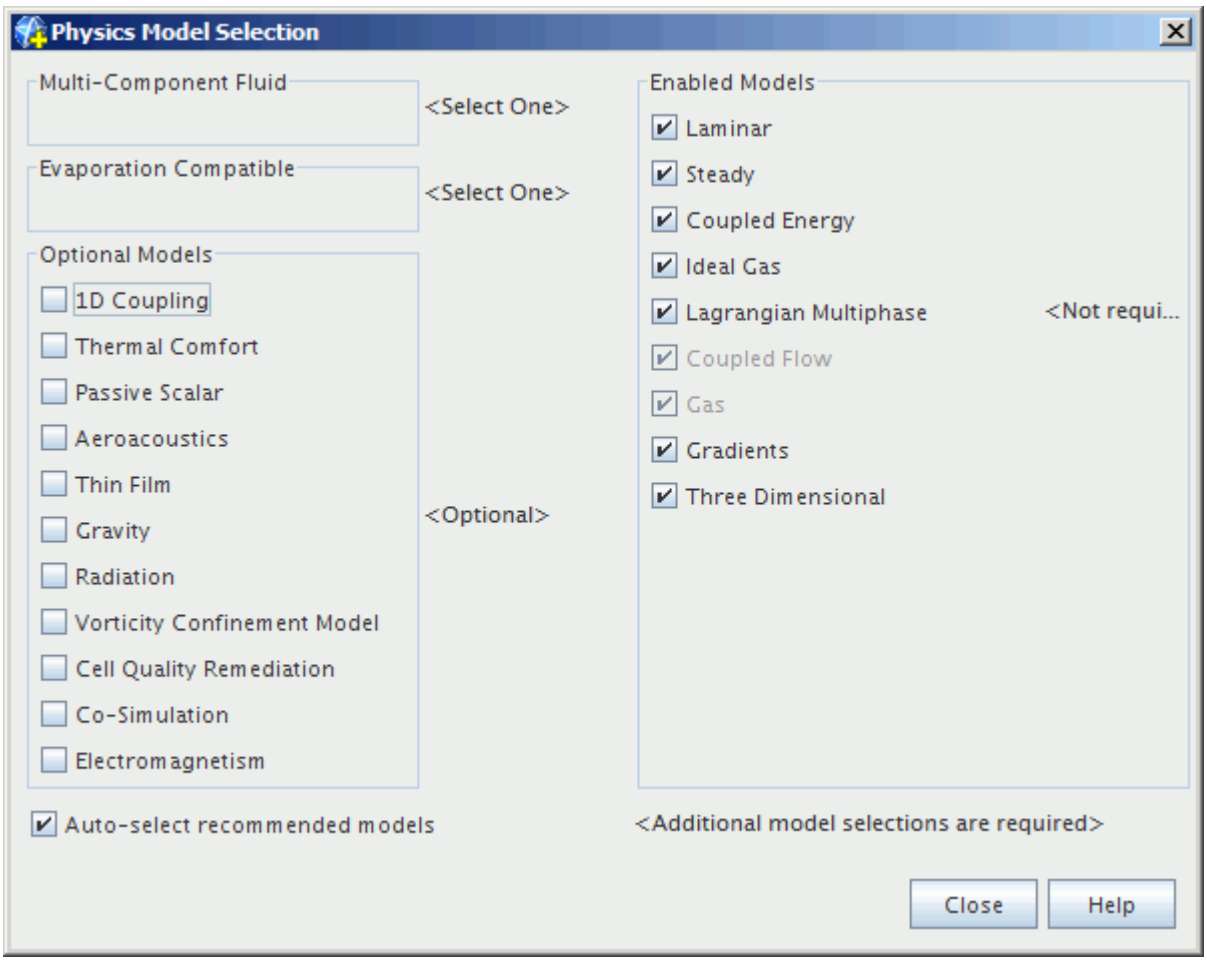
The *Phase Model Selection* dialog changes dynamically in response to previous selections. For instance if **Residence Time** and **Material Particles** are selected, it appears as shown below, where a selection from the *Material* group is required. The <Additional selections are required> label tells you that the model selection is incomplete. An incomplete model selection means that the Lagrangian phase cannot be initialized in its current state.



Selections that are made in this dialog can have an impact on the models that are required in the continuous phase. For example, selecting **Quasi-Steady Evaporation** for a **Liquid** particle in the *Phase Model Selection* dialog requires the presence of a multi-component gas in the continuous phase (that is, the physics continuum). If this selection has not been made in the continuous phase, the simulation fails to initialize, and an information dialog is displayed describing the continuous phase models that are missing.

You can check the compatibility of models that are chosen in the *Phase Model Selection* dialog with the models of the underlying continuum. Reopen the *Physics Model Selection* dialog for the physics continuum and look for any <Select One> indicators on the left-hand side. If these indicators are present, it is necessary to make further selections that satisfy the Lagrangian phase requirements. This process can involve deselecting existing models to expose appropriate group boxes on the left.

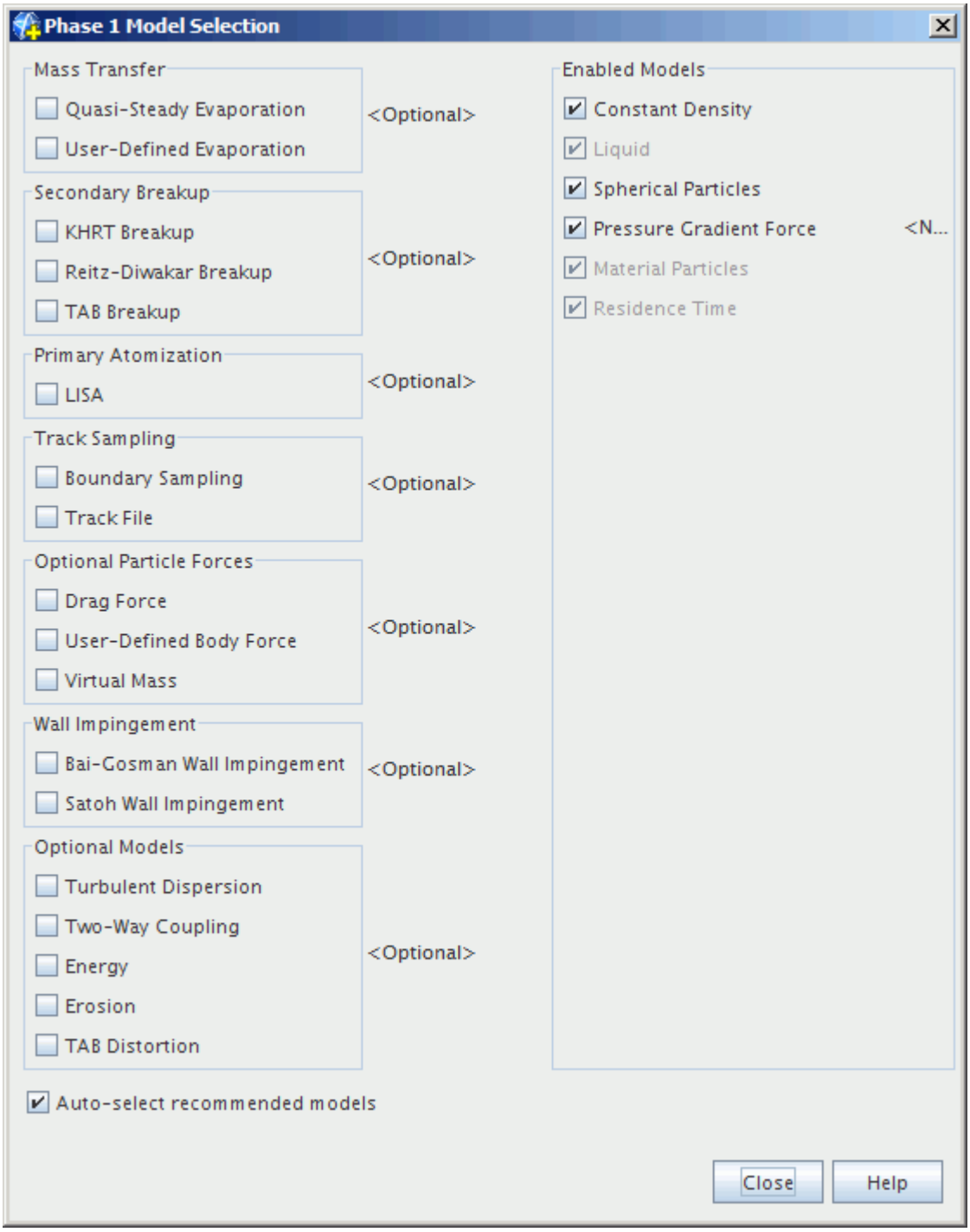
An example of a *Physics Model Selection* dialog with <Select One> indicators is shown below. In this example, **Quasi-Steady Evaporation** is selected for the Lagrangian phase without having first selected a multi-component gas in the physics continuum. To rectify this issue, it would be necessary to clear models to the point where the **Gas** model is on the left-hand side. You can then select the **Multi-Component Gas** model from the appropriate group box.



More details on the *Phase Model Selection* dialog, as well as descriptions of the individual models available, are given in section, Working with Lagrangian Phase Models.

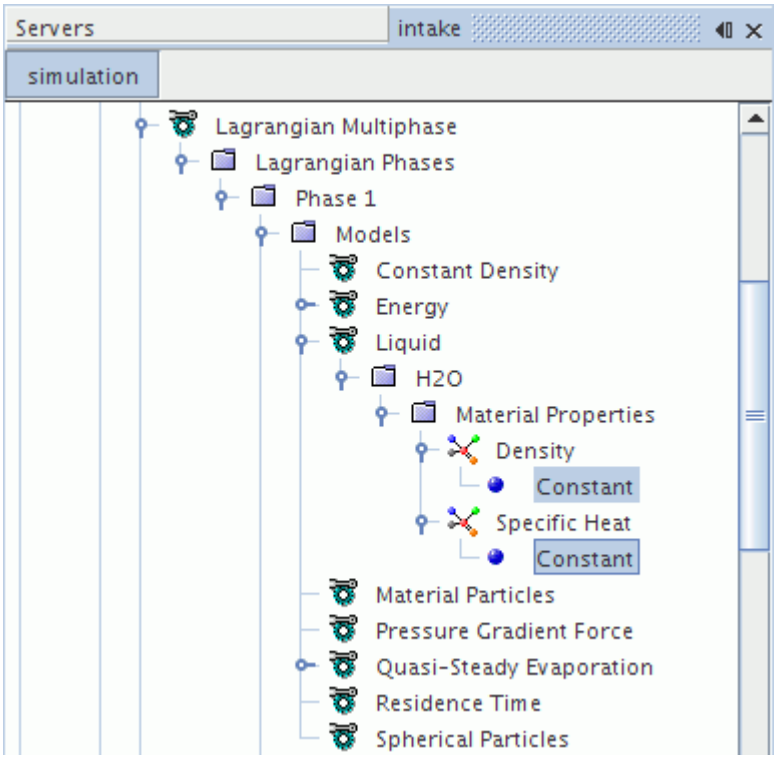
Using Auto-Select Recommended Phase Models

The **Auto-select recommended Phase models** checkbox is activated by default, which indicates that when there is a choice of models, the recommended ones are selected automatically.



Defining Material Properties for Each Lagrangian Phase

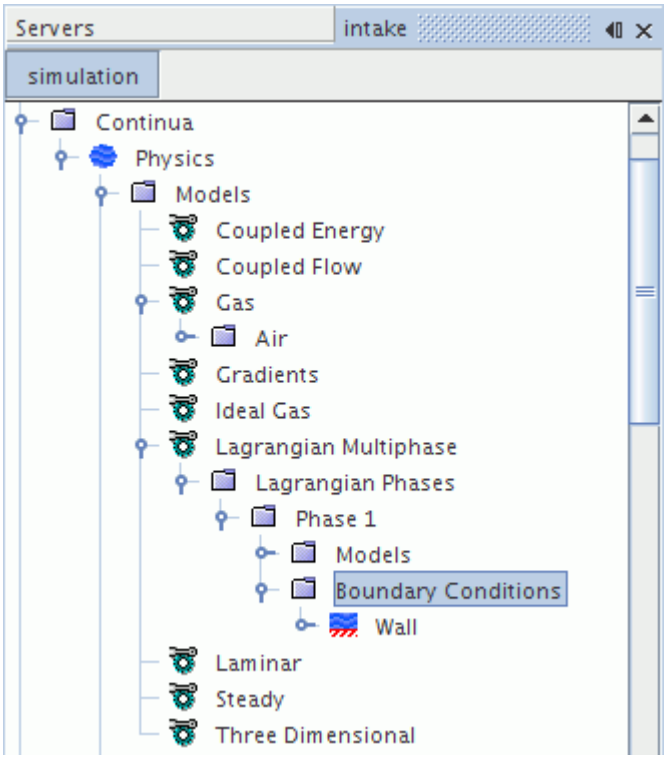
If a **Material Particle** model is chosen in the *Phase Model Selection* dialog, set the material properties, using the corresponding material node in the Lagrangian phase object tree. For example, the tree below has been expanded to show the **Density** and **Specific Heat** node for Lagrangian phase, **Phase 1**.



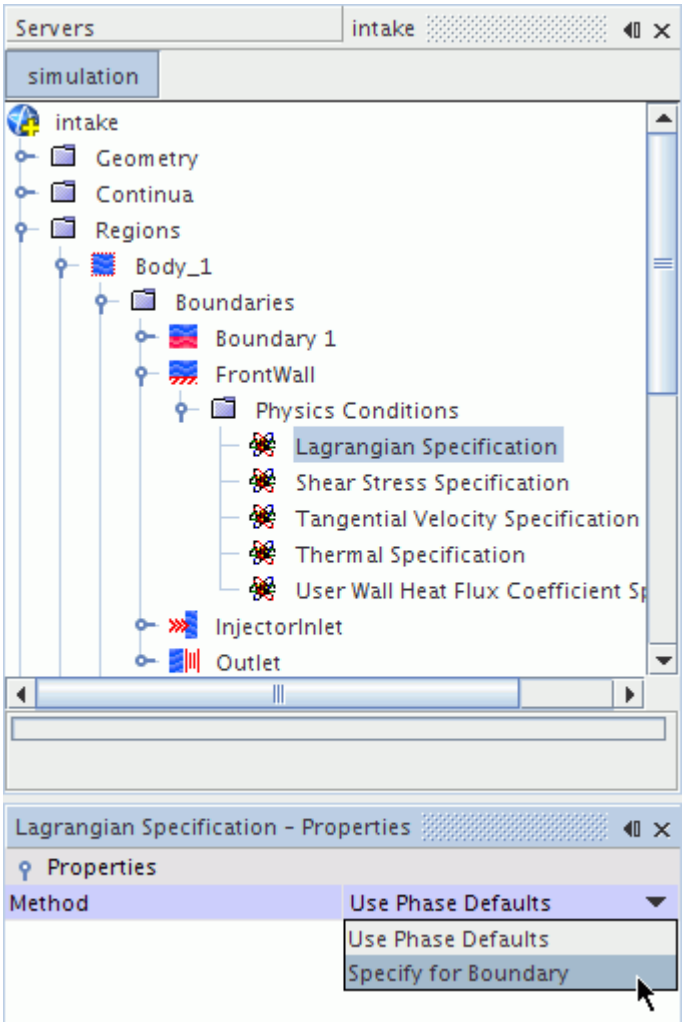
Further information on the material models available for a Lagrangian phase is given in section, Modeling Lagrangian Phase Materials.

Defining Boundary Conditions for Each Lagrangian Phase

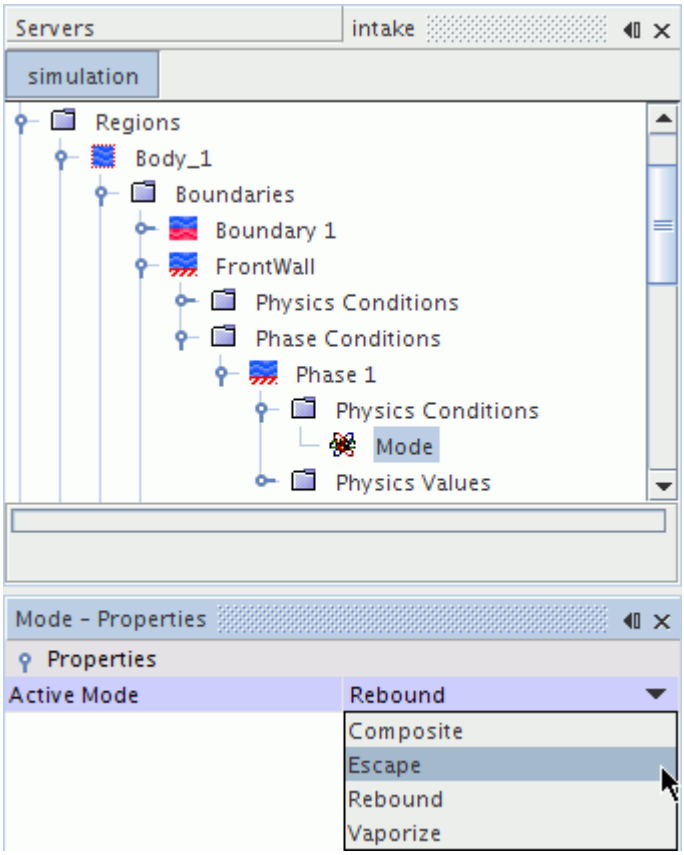
How particles respond on impact with a solid boundary greatly depends on the properties of the particle. For this reason, each Lagrangian phase contains a **Boundary Conditions** manager node that defines the default response at solid boundaries for particles belonging to that phase.



It is possible to override the default behavior on a particular solid boundary using the *Method* property of the **Physics Conditions** > **Lagrangian Specification** node belonging to the boundary. In the example that is shown below, the *Method* property is being set to **Specify for Boundary** instead of **Use Phase Defaults**.



This change results in a **Phase Conditions** manager appearing for the boundary. The **Phase Conditions** node contains sub-nodes for each Lagrangian phase, and these sub-nodes can be used to customize the particle impact behavior for the boundary. An example is shown below.



Further information on specifying impact behavior is given in Working with Lagrangian Phase Boundary Conditions.

Creating and Defining Injectors

The locations at which particles enter the fluid continuum, and the manner in which they enter, are defined using injectors. Each injector generates new parcels for a given Lagrangian phase. Any number of injectors can be associated with the same Lagrangian phase.

Full information on creating and defining injectors is given in section, Working with Injectors.

Adjusting Parameters for the Lagrangian Multiphase Solver

The default settings of the Lagrangian Multiphase solver are a good compromise between accuracy, CPU time, and stability. The section, Lagrangian Multiphase Solver Reference, provides more detail on how accuracy and stability can be increased when necessary.

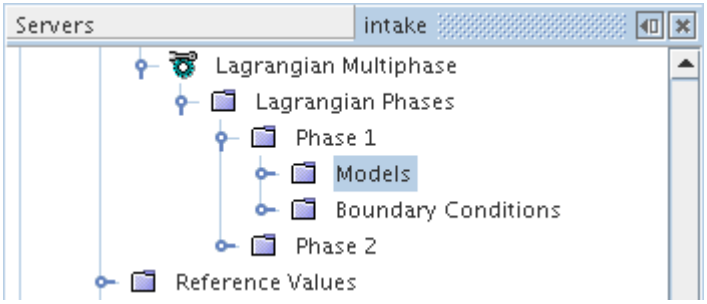
The Lagrangian Multiphase Workflow

To use the Lagrangian Multiphase model, define the properties of the continuous and dispersed phases, their boundary conditions, their interactions, and their mode of injection.

As a prerequisite, Lagrangian modeling requires a physics continuum for fluid flow, using either the coupled or the segregated flow solver. Then use the following steps:

1. Select **Lagrangian Multiphase** from the *Optional Models* group box.
2. Create one or more Lagrangian phases to represent each distinct dispersed phase (each kind of particle, droplet, or bubble) within the simulation:
   a. Expand the **Models** > **Lagrangian Multiphase** model node.
   b. Right-click the **Lagrangian Phases** manager node and select **New**.

      A new Lagrangian phase node appears with the default name **Phase 1**. Rename the node if desired.
   c. Repeat the previous step for each additional dispersed phase.
3. Select models that describe the physics in each Lagrangian phase. For each phase node:
   a. Expand the phase node, right-click the **Models** node, and choose **Select Models...**.



   b. Select models from *Phase Model Selection* dialog.

      For a steady simulation, **Residence Time** is selected automatically in the *Enabled Phase Models* group box. For an unsteady simulation, no initial selections are made.

      - If the Two-Way Coupling model is active, source terms that the Lagrangian phase models compute are stored for subsequent application in their respective transport equations.
      - If the Two-Way Coupling model is not active, it is possible to freeze the Lagrangian Multiphase Solver until convergence is reached, using its *Solver Frozen* property. Freezing can save significant CPU time. The Lagrangian Multiphase solution can then be obtained by unfreezing the solver and stepping the solution. Even when the Two-Way Coupling model is active, an economic route to a steady solution can be to update the Lagrangian Multiphase solution less frequently than the other solvers. Use the *Update Frequency* property of the Lagrangian Steady solver. However, do not use the Solver Frozen feature as your first choice in other situations, since it is primarily a “debugging” tool and not supported for normal simulation work.

      Selections that are made in this dialog can have an impact on the models that are required in the continuous phase. For example, selecting **Quasi-Steady Evaporation** for a **Liquid** particle in the *Phase Model Selection* dialog requires the presence of a multi-component gas in the continuous phase (that is, the physics continuum). If this selection has not been made in the continuous phase, the simulation fails to initialize, and an information dialog is displayed describing the continuous phase models that are missing.

      When you want to include the impact of particle interactions on the continuous phase, activate **Two-Way Coupling**.

      You can check the compatibility of models that are chosen in the *Phase Model Selection* dialog with the models of the underlying continuum. Reopen the *Physics Model Selection* dialog for the physics continuum and look for any `<Select One>` indicators on the left-hand side. If these indicators are

present, it is necessary to make further selections that satisfy the Lagrangian phase requirements. This process can involve deselecting existing models to expose appropriate group boxes on the left.

More details on the *Phase Model Selection* dialog, as well as descriptions of the individual models available, are given in section, Working with Lagrangian Phase Models.

4. Define material properties for each Lagrangian phase.

   If a Material Particle model is chosen in the *Phase Model Selection* dialog, set the material properties, using the corresponding material node in the Lagrangian phase object tree. Further information on the material models available for a Lagrangian phase is given in Modeling Lagrangian Phase Materials.

5. Define boundary conditions for each Lagrangian phase.

   How particles respond on impact with a solid boundary greatly depends on the properties of the particle. For this reason, each Lagrangian phase contains a **Boundary Conditions** manager node that defines the default response at solid boundaries for particles belonging to that phase. Further information on specifying impact behavior is given in Working with Lagrangian Phase Boundary Conditions.

   Additionally, Lagrangian particles can interact with the phases of other multiphase models in a simulation. Available interactions are:

   - Lagrangian particles (droplets) to fluid film. See Setting Up Lagrangian Impingement.
   - Lagrangian particles to VOF phase. See VOF-Lagrangian Impingement.

6. Create and define injectors.

   The locations at which particles enter the fluid continuum, and the manner in which they enter, are defined using injectors. Each injector generates new parcels for a given Lagrangian phase. Any number of injectors can be associated with the same Lagrangian phase. Full information on creating and defining injectors is given in Working with Injectors.

7. Set parameters for the Lagrangian Multiphase solver.

   The default settings of the solver are a good compromise between accuracy, CPU time, and stability.

   When you are modeling Two-Way Coupling on a fine mesh, set the **Source Smoothing Method** of the **Two-Way Coupling** solver to **Cell Cluster**. This ensures that two-way coupling assumptions are valid when particle sizes are comparable to cell sizes.

   Use the *Scale* property of the **Cell Cluster** node to control the size of the large-scale cells. The cells of the large-scale grid are not always convex, but are approximately cubical in most cases. See Source Smoothing Method.

   See the section Lagrangian Multiphase Solver Reference for more detail on how accuracy and stability can be increased when necessary.

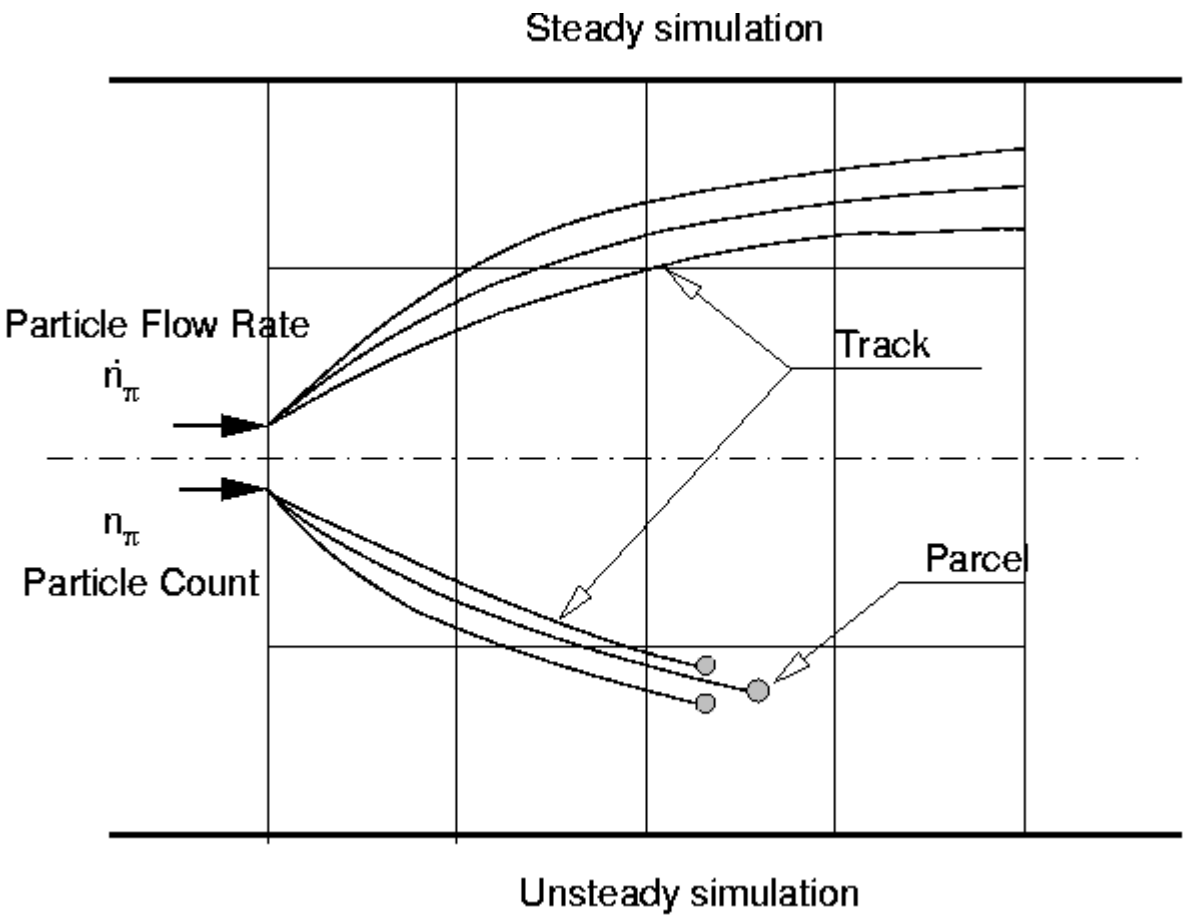
8. Prepare for post-processing. Define scenes, reports, monitors, and plots for output such as:

   - Direct data on boundaries
   - Local instantaneous data
   - Particle track data
   - Sampled data on boundaries
   - Phase data in solution histories

9. Run the simulation and view the results.

Using the Lagrangian Multiphase Model with Time Models

The Lagrangian Multiphase model is compatible with all Time models except Harmonic Balance.



Unsteady Simulation

In unsteady simulations, the approach can be categorized as a *discrete element method* [576]. In this method, the solution at a given instant in time is the state (position, velocity, temperature, and so on.) of each parcel. The number of nominally identical particles that a parcel represents, or particle count, is part of the state of the parcel. Optionally, the tracks of parcels can also be recorded in unsteady simulations: these tracks are the history of the parcel state, which can be visualized as a trajectory.

Steady Simulation

In steady simulations, the approach can be categorized as a trajectory method [576]. The solution is typically the tracks of the parcels, and the variation of state (velocity, temperature, and so on) along them. Each track records the complete lifetime of one parcel, from when it is injected to when it leaves the domain. The particle flow rate along each track is part of its state.

Parcels are indeed supposed to leave the domain in steady simulations. This approach is unsuitable for configurations that permanently carry particles in the continuous phase, whether through flow features (for

example, recirculation zones) or geometry (for example, mixing vessels). For these configurations, an unsteady simulation can be more appropriate.

Lagrangian Multiphase Model Reference

The Lagrangian Multiphase model node is a manager node containing sub-nodes describing each Lagrangian phase. It is required for all Lagrangian multiphase simulation.

Theory	See Lagrangian Multiphase Flow in the *Theory Guide*.	
Provided By	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Lagrangian Multiphase**	
Requires	*Material*: one of **Gas**, **Liquid**, **Eulerian Multiphase**, **Multi-Component Gas**, **Multi-Component Liquid** *Flow*: **Coupled** or **Segregated**	
Properties	None	
Specific Right-Click Actions	**New** Right-click **Lagrangian Multiphase** > **Lagrangian Phases** and select **New** to create a sub-node under **Lagrangian Phases** named **Phase *n***. See What is a Lagrangian Phase?	
Activates	Model Controls (child nodes)	**Lagrangian Phases**
	Solvers	Lagrangian Multiphase Solver. See Lagrangian Multiphase Solver Reference.

Contents:

Lagrangian Multiphase Solver Reference

Lagrangian Multiphase Solver Reference

The Lagrangian Multiphase solver controls the update of the Lagrangian Multiphase model. Sub-solvers are added beneath the Lagrangian Multiphase solver node depending on other selections for the physics continuum or Lagrangian phase.

Lagrangian Multiphase Solver Properties

Maximum Courant Number

The value of Co_{max} in Eqn. (3264) used to provide an upper bound for the local time-step. Reducing this value can increase accuracy, at the expense of CPU time.

Minimum Courant Number

The value of Co_{min} in Eqn. (3265) used to provide a lower bound for the local time-step. Reducing this value can increase accuracy, at the expense of CPU time.

Maximum Parcel Count

The maximum number of parcels per Lagrangian phase. If this limit is exceeded, the simulation throws an error. A value of 0 (the default) means unlimited.

Solver Frozen

When **On**, prevents the solver from updating the solution while iterating or responding to other solvers. This feature is primarily a “debugging” tool and is not supported for normal simulation work.

Temporary Storage Retained

When **On**, retains or deactivates temporary storage at the end of the iteration. Examples include the Lagrangian source terms that become active in the Two-Way Coupling model. These quantities become available as field functions during subsequent iterations.

Lagrangian Sub-Solvers

Additional solvers become available under the Lagrangian Multiphase solver as you select additional models.

Lagrangian Load-Balancing Solver

When *Activated*, this solver maintains and controls the Lagrangian component of the load-balancing framework for parallel servers. Other properties are as follows:

Parcel Weight

Controls the relative weight of the parcel versus the cell. This parameter allows you to control the balance of Lagrangian physics versus Eulerian physics. The weight can range from 1 (the default), as Lagrangian as possible, to 0, strictly Eulerian.

Rebalance Limit

Sets the point at which the load balance is even enough and the solver stops. The limit is the ratio of the loading between the most and least loaded domains. Choose a value of the limit that avoids rebalancing efforts beyond useful levels. The lower the limit, the less frequent the rebalancing. Setting the value close to 1 results in 'always rebalance' behavior. The default value is 0.8.

Lagrangian Steady Solver

Provides controls for when you run a Lagrangian simulation in steady state.

Update Frequency

Sets the number of updates of the continuous phase solution per update of the Lagrangian Multiphase solution. Increasing this value can save CPU time.

First Iteration

Specifies the first iteration in which the Lagrangian solver runs.

Maximum Sub-Steps

Sets the maximum number of sub-steps for which particles are tracked, to prevent particles being tracked indefinitely. Set this quantity high enough to avoid prematurely terminating tracks.

Verbosity

Controls the level of detail that the solver presents is in the Output window.

- **None**: No output.
- **Low**: If any tracks reached maximum residence time or maximum sub-steps, a single message is printed at the end of the Lagrangian solver stage. This setting is the default.
- **High**: Same as Low, plus an update every 1000 sub-steps of the number of active parcels and the minimum residence time observed among those parcels.

Sub-Step Reporting Frequency

Sets the number of sub-steps between each instance of reporting solver progress.

Maximum Residence Time

Sets the residence time beyond which particles are not tracked, to prevent particles being tracked indefinitely. Set this quantity high enough to avoid prematurely terminating tracks.

Active Parcel Fraction Cut-off

Sets the minimum fraction of active parcels required for Lagrangian tracking to continue in the current iteration. The fraction is based on the maximum number of active parcels seen previously. If the fraction falls below the entered value, Lagrangian tracking for the iteration ends with no further sub-steps. The default value is `0.0`.

Tracking Integration Method

Specifies the integration scheme that STAR-CCM+ uses to integrate the particles position and velocity. The available options are **1-st order** and **2-nd order**.

Lagrangian Unsteady Solvers: Implicit, Explicit, and PISO

The Implicit, Explicit, and PISO Unsteady solvers share properties.

Update Frequency

Selects the frequency of updates of the continuous phase solution per update of the Lagrangian Multiphase phase solution. In scenes containing Lagrangian data, do not set the scene update policy to Iteration when this Lagrangian solver is set to update after the last iteration. Otherwise, the scene updates before the Lagrangian solver does. Options are:

- **Recommended**: Runs the solver every iteration (implicit) or PISO corrector (PISO) when there are two-way coupled phases, or after the last iteration (or PISO corrector) in a time-step when there are only one-way coupled phases.

- **Once per Time-Step**: Runs the solver in the first iteration (implicit) or PISO predictor (PISO) when there are two-way coupled phases, or after the last iteration (or PISO corrector) when there are only one-way coupled phases.
- **Every Iteration**: Runs the solver in every iteration or PISO corrector.

This property is not available for Explicit Unsteady simulations.

Verbosity

Controls the level of detail that the solver presents in the *Output* window.

- **None**: No output.
- **Low**: If any tracks reached maximum residence time or maximum sub-steps, a single message is printed at the end of the Lagrangian solver stage. This setting is the default.
- **High**: Same as Low, plus an update every 100 sub-steps of the number of active parcels and the minimum residence time observed among those parcels.

Maximum Sub-Steps

Sets the maximum number of sub-steps for which particles are tracked, to prevent particles being tracked indefinitely. Set this quantity high enough to avoid prematurely terminating tracks.

Sub-Step Reporting Frequency

Sets the number of sub-steps between each instance of reporting solver progress.

Active Parcel Fraction Cut-off

Sets the minimum fraction of active parcels required for Lagrangian tracking to continue in the current time-step. The fraction is based on the maximum number of active parcels seen previously. If the fraction falls below the entered value, Lagrangian tracking for the time-step ends with no further sub-steps. The default value is `0.0`.

Tracking Integration Method

Specifies the integration scheme that STAR-CCM+ uses to integrate the particles position and velocity.

The **1-st order** integration scheme is generally sufficient. For particles with highly curved trajectories, the **2-nd order** integration scheme allows you to use a larger time-step, while retaining the same accuracy.

Two-Way Coupling Solver

Controls the under-relaxation, linearization, and smoothing of the sources appearing in the continuous phase equations due to the Lagrangian phases. See Two-Way Coupling with the Continuous Phase.

Under-Relaxation Factor

The fraction of the new source values which are used when updating source values. Lower values can stabilize simulations with strong coupling between the phases. See Eqn. (3034).

If you are also using the PISO Unsteady solver in a simulation of transient flow:

- When the under-relaxation factor of the Two-Way Coupling solver is 1, the PISO Unsteady solver stops after it converges for the first time, or when it reaches the maximum number of PISO corrector steps.

- When the under-relaxation factor of the Two-Way Coupling solver is < 1, the PISO Unsteady solver continues after convergence or after it reaches one step before the maximum number of PISO correctors. The PISO Unsteady solver uses sources that are not under-relaxed. It keeps the Langrangian phases frozen until it converges again or reaches the maximum number of PISO corrector steps. This approach conserves total mass, momentum, and energy.

Maximum Volume Fraction

Sets the maximum volume fraction used in the continuous phase to limit the source terms.

Linearize Sources

Indicates when, if possible, to split the sources into explicit and implicit components. Typically, splitting stabilizes simulations with strong coupling between the phases.

Limit Sources

When the current volume fraction exceeds the maximum volume fraction and this property is activated, the two-way coupling sources are limited proportionately to (maximum volume fraction)/(current volume fraction). This limitation can improve stability. There is no effect when the current volume fraction is less than the maximum.

Under-Relaxation Factor Ramp

When *Method* is set to **Linear Ramp**, the Two-Way Coupling solver progressively increases its under-relaxation factor from an *Initial Value* at the *Start Iteration* to the final value at the *End Iteration*.

Source Smoothing Method

When *Method* is set to **Cell Cluster**, STAR-CCM+ imposes a coarse grid for evaluating void fraction data and momentum exchanges. Activate this method when particle sizes are close to, or greater than, the smallest cell sizes in the simulation.

- Use the *Scale* property of the **Cell Cluster** child node to set the approximate size of the coarse-grid cells. The scale is defined as a multiple of the length of the largest cell in the original grid, and ranges from `1` to `10`. Default entry is `1`.

This method also activates the **Particle Smooth Source Group** and **Particle Smooth Source Weight** field functions. See Two-Grid Procedure and Lagrangian Multiphase Field Functions Reference.

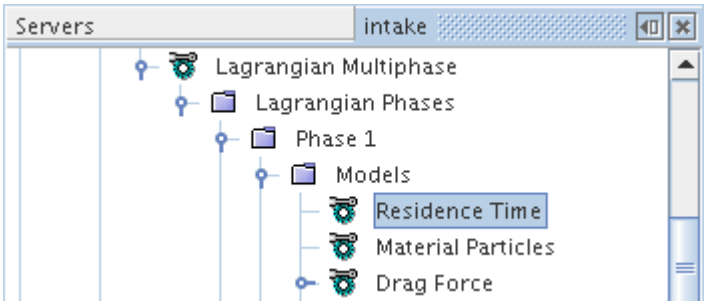
Lagrangian Passive Scalar Solver

Solver Frozen

When **On**, prevents the solver from updating the solution while iterating or responding to other solvers. This feature is primarily a “debugging” tool and is not supported for normal simulation work.

Lagrangian Phase Models

A combination of Lagrangian phase models defines a Lagrangian phase. Each individual model represents one of the characteristics of that phase.



A Lagrangian phase model plays the same role in a Lagrangian phase as a conventional model does in a continuum. It defines how a physical phenomenon in a phase is represented, what the primary variables of the simulation are and what mathematical formulation is used to generate the solution.

The phase models available in STAR-CCM+ are:

Group Box	Phase Model
Particle Types	Massless Particles
	Material Particles
Material	Gas
	Liquid
	Multi-Component Coal
	Multi-Component Liquid
	Multi-Component Particle
	Solid
Equations of State	Constant Density
	Polynomial Density
Optional particle forces	Drag Force
	Pressure Gradient Force
	User-Defined Body Force
	Virtual Mass
	Shear Lift Force
	Electrical Charge
Mass transfer	Quasi-Steady Single-Component Droplet Evaporation
	User-Defined Single-Component Droplet Evaporation
	Multi-Component Droplet Evaporation
	Coal Combustion
	Particle Reaction
Energy	Lagrangian Energy Model

Group Box	Phase Model
Secondary Breakup	KHRT Breakup
	Reitz-Diwakar Breakup
	TAB Breakup
	SSD Breakup
Primary Atomization	LISA Atomization
	Huh Atomization
Track Sampling	Track File
	Boundary Sampling
Wall Impingement	Bai-Gosman Wall Impingement
	Bai-ONERA Wall Impingement
	Satoh Wall Impingement
	Ice Accretion
Species	Lagrangian Species
Optional Models	Lagrangian Energy Model
	Particle Radiation
	Turbulent Dispersion
	Residence Time
	Coal Fuel NOx
	Particle Porosity
	Erosion
	Two-Way Coupling
	TAB Distortion
	Passive Scalars
Optional Energy Sources	User-Defined Energy Source
Collision Detection	NTC Collision Model
Moisture Evaporation	Coal Moisture Evaporation
Char Oxidation	First-Order Char Oxidation
	Half-Order Char Oxidation
Raw Coal Devolatilization	Two-Step Devolatilization
	User-Defined Devolatilization

Contents:

Particle Type Models

Lagrangian Phase Material Models
Lagrangian Phase Equation of State Model
Particle Force Models
Charged Particle Models
Lagrangian Energy Model
User-Defined Energy Source Model
NTC Collision Model
Droplet Evaporation Models
Lagrangian Species Model
Two-Way Coupling Model
Turbulent Dispersion Model
Erosion Model
Residence Time
Particle Radiation Model
TAB Distortion Model
Lagrangian Passive Scalars
Impingement Heat Transfer Model
Parcel Depletion Model
Particle Porosity

Particle Type Models

The Massless and Material Particles models are available in the *Particle Type* group box.

Massless Particles

The Massless Particles model represents massless virtual particles, sometimes known as fluid particles, which are not material objects. These massless particles move with the continuous phase, without influencing each other or the continuous phase.

It can be useful to include massless particles in a simulation to visualize or quantify:

- Turbulent diffusion of the continuous phase, with the Turbulent Dispersion model
- The residence time of fluid in the continuous phase, with the Residence Time model
- Mixing of fluid streams in the continuous phase, by associating a different massless Lagrangian phase with each fluid stream

Provided By	**[phase]** > **Models** > *Particle Type*	
Example Node Path	**[phase]** > **Models** > **Massless Particles**	
Requires	*Optional Models*: **Lagrangian Multiphase**	
Properties	None	
Activates	Boundary Inputs	See Massless Particle Boundary Settings.

	Field Functions	**Incident Particle Flux of Phase n**, **Injector Index**, **Number Density of Phase n**, **Parcel Centroid**, **Parcel Time Step**, **Particle Velocity**. See Lagrangian Multiphase Field Function Reference.

Massless Particle Boundary Settings

Wall

Mode

Selects the boundary interaction mode for the Lagrangian phase. Lagrangian boundary conditions set defaults that can then be modified for specific boundaries. See Setting Default Lagrangian Phase Boundary Conditions.

Active Mode	Corresponding Physics Value Nodes
Composite Sets the actual boundary interaction as a range of possibilities, each with their own probability.	**Mode Probabilities** A manager node. Available boundary interaction modes are listed under this manager and the probability of its occurrence can be specified for each.
Escape Escape mode means that the particle is removed from the simulation, effectively escaping the solution domain. The mass, momentum, and energy (where applicable) of the parcels are also removed.	None

Composite Default Mode

Sets the default interaction mode when the *Active Mode* is **Composite**. Choose the interaction that has the highest probability of occurring.

Material Particles

The **Material Particles** model represents real particles of a physical substance, such as sand, water droplets, or air bubbles. Material particles have mass and volume, and physical conservation laws govern their behavior. In the most basic setup, these laws are conservation of mass and momentum. The particles are assumed to be internally homogenous and without internal motion.

Theory	See Particle Mass Balance and Mass Transfer.
Provided By	**[phase]** > **Models** > *Particle Type*
Example Node Path	**Lagrangian Multiphase** > **Lagrangian Phases** > **[phase]** > **Models** > **Material Particles**
Requires	*Optional Models*: **Lagrangian Multiphase**

Properties	None	
Activates	Physics Models	*Material* *Optional Particle Forces*: **Pressure Gradient Force** *Particle Shape*: **Spherical Particles**
	Boundary Inputs	See Material Particles Boundary Settings.
	Field Functions	**Incident Particle Flux of Phase n**, **Mass Flow Rate of Phase n**, **Mass Flux of Phase n**, **Parcel Centroid**, **Parcel Mass Flow Rate**, **Parcel Size**, **Parcel Time Step**, **Parcel Volume Flow Rate**, **Particle Density**, **Particle Diameter**, **Particle Incidence Angle**, **Particle Mass**, **Particle Slip Velocity**, **Particle Surface Area**, **Particle Velocity**, **Particle Volume**, **Void Fraction**, **Volume Fraction of Phase n**. See Lagrangian Multiphase Field Function Reference.

Material Particles Boundary Settings

Wall

Normal Restitution Coefficient

See Boundary Values Reference.

Tangential Restitution Coefficient

See Boundary Values Reference.

Mode

Selects the boundary interaction mode for the Lagrangian phase. See Setting Default Lagrangian Phase Boundary Conditions.

Active Mode	Corresponding Physics Value Nodes
Escape Escape mode means that the particle is removed from the simulation, effectively escaping the solution domain. The mass, momentum, and energy (where applicable) of the parcels are also removed.	None
Rebound Rebounding particles remain active in the simulation, but with their velocity modified according to restitution relationships. For walls, baffles and contact interfaces, this mode is the default. When a two-way coupled particle rebounds off a Fluid Film, some of the energy of the particle can be transferred to the film.	None

Composite Default Mode

Sets the default interaction mode when the *Active Mode* is **Composite**. Choose the interaction that has the highest probability of occurring.

Lagrangian Phase Material Models

Any Lagrangian phase that includes material particles also has its own material model, since the phase represents a distinct physical substance. (Lagrangian phases that use massless particles do not have their own material models.)

There are currently six selectable material models for a Lagrangian phase:

- Gas, representing a single-component gas bubble
- Liquid, representing a single-component liquid droplet
- Solid, representing a single-component solid particle
- Multi-Component Coal, representing a coal particle
- Multi-Component Liquid, representing a multi-component liquid droplet
- Multi-Component Particle, representing a multi-component solid particle

Each of these material models behaves in an identical manner to their continuous phase counterpart.

Gas, Liquid, Solid, and Multi-Component Liquid Phases

Model Names	**Gas**	
	Liquid	
	Solid	
	Multi-Component Liquid	
Theory	See: • Gas: Ideal Gas and Real Gas in the *Theory Guide*. • Liquid and Solid: Lagrangian Multiphase Flow in the *Theory Guide*. • Multi-Component Liquid: Droplet Evaporation in the *Theory Guide*.	
Provided By	**[phase]** > **Models** > *Material*	
Example Node Path	**[phase]** > **Models** > **Gas** | **Liquid** | **Solid** | **Multi-Component Liquid**	
Requires	*Optional Models*: **Lagrangian Multiphase** *Particle Type*: **Material Particles**	
Activates	Physics Models	Gas: *Equation of State* Liquid: *Equation of State*, *Mass Transfer* Solid: *Equation of State* Multi-Component Liquid: *Equation of State*, *Species*
	Material Properties	**[gas]**: The default is **Air**. **[liquid]**: The default is **H2O**. **[solid]**: The default is **Al**.

		See Managing Single-Component Materials. **Droplet Components**

Multi-Component Coal

This material model node represents the Lagrangian phase material model for a coal particle. The particle is a mixture that is composed of four constituent components: raw coal, char, ash, and H2O. Raw coal, that is, the dry ash-free (daf) portion of the coal, consists of volatile matter (VM) and char. In coal combustion literature, char is also sometimes referred to as fixed carbon (Cs). You can either split the VM and char content present in raw coal and inject char separately, or VM and char can be injected together as raw coal. In the latter case, which is recommended, the char mass fraction that is specified at the injector would be zero and the char content in the domain would evolve due to the devolatilization process. The **Coal Components** node manages these four constituents. Default properties for each of these components are supplied from the property database. Modify the properties according to the type of coal particle being simulated.

The proportions in which components are present for a particular phase of coal particles are set using the Species Mass Fraction node on the corresponding injector. Using this node is equivalent to setting the proximate analysis of a given coal mixture. If a particular component is not present, set its mass fraction to zero.

Material properties of each component can be set using the property sub-nodes within the **Component Properties** nodes for each of the four components. Relevant properties of the overall mixture can be set using the property nodes within the **Mixture Properties** manager node.

See Coal Combustion for details on using coal particles in a coal combustion simulation.

Theory	See Coal Combustion.	
Provided By	**[phase]** > **Models** > *Material*	
Example Node Path	**[phase]** > **Models** > **Multi-Component Coal**	
Requires	*Optional Models*: **Lagrangian Multiphase** *Particle Type*: **Material Particles**	
Activates	Physics Models	*Equation of State* *Mass Transfer* *Species*
	Material Properties	**Coal Components** > **RawCoal**, **Char**, **Ash**, **H2O**
	Field Functions	**Particle Mass Fraction of Ash** **Particle Mass Fraction of Char** **Particle Mass Fraction of H2O** **Particle Mass Fraction of RawCoal** **Particle Molar Concentration of Ash** **Particle Molar Concentration of Char** **Particle Molar Concentration of H2O** **Particle Molar Concentration of RawCoal**

Multi-Component Particle

Provided By	**[phase]** > **Models** > *Material*	
Example Node Path	**[phase]** > **Models** > **Multi-Component Particle**	
Requires	*Optional Models*: **Lagrangian Multiphase** *Particle Type*: **Material Particles**	
Activates	Physics Models	*Species*
	Material Properties	**Particle Components**

Lagrangian Phase Equation of State Model

Any Lagrangian phase that includes material particles also requires an Equation of State model, to define how the density of the material is computed.

The Constant Density and Polynomial Density Lagrangian phase models are the same as the Constant and Polynomial Density models for the STAR-CCM+ simulation in general, except that they apply to the Lagrangian phase rather than throughout the continuum.

Model Names	**Constant Density**	
	Polynomial Density	
Theory	For the Polynomial Density model, see Polynomial Density Formulation.	
Provided By	**[phase]** > **Models** > *Equation of State*	
Example Node Path	**[phase]** > **Models** > **Constant Density | Polynomial Density**	
Requires	*Particle Type*: **Material Particles** *Material*: one of **Gas**, **Liquid**, **Multi-Component Coal**, **Multi-Component Liquid**, **Multi-Component Particle**, **Solid**	
Activates	Material Properties	Selecting the Constant or Polynomial Density model determines the *Method* property of the **Density** node for the model selected from the *Material* group. See Using Polynomial in T.

Particle Force Models

You can apply models for the forces of drag, shear lift, gradient presure, virtual mass, and electric charge to Lagrangian phases as needed.

Particle Force Models Reference

Model Names	Drag Force
	Shear Lift Force

	Pressure Gradient Force	
	Virtual Mass	
	User-Defined Body Force	
Theory	See: Drag Force Particle Shear Lift Force Pressure Gradient Force Virtual Mass Force User-Defined Body Force	
Provided By	**[phase]** > **Models** > *Optional Particle Forces*	
Example Node Path	**[phase]** > **Models** > **Drag Force**	
Requires	*Particle Type*: **Material Particles** Shear Lift Force also requires *Optional Particle Forces*: **Drag Force**	
Activates	Model Controls (child nodes)	Drag Force model: **Drag Coefficient**: see Drag Coefficient Properties. Shear Lift Force model: **Shear Lift Coefficient**: see Shear Lift Coefficient Properties. User-Defined Body Force model: User Body Force
	Field Functions	For the Particle Drag Force model: **Particle Drag Force**. See Lagrangian Multiphase Field Function Reference.

Drag Force

The Drag Force model calculates the force on a material particle due to its velocity relative to the continuous phase, Eqn. (2973).

Although the **Drag Force** node has no properties of its own, its sub-node **Drag Coefficient** does.

Drag Coefficient Properties

The Drag Coefficient is used to formulate the drag force, Eqn. (2973).

Dimensions

The dimensionality of the drag coefficient (read-only).

Method

Choose from the following options:

Field Function

Defines the drag coefficient using a field function (typically user defined). A field function method node is added as a child to this node.

Haider & Levenspiel

The Haider and Levenspiel method is suitable for non-spherical particles with $Re_p < 2.6 \times 10^5$.

Liu

The Liu correlation is only available when the TAB Distortion phase model has also been selected. It is suitable for liquid droplets.

Schiller-Naumann

The Schiller-Naumann correlation is available only when the continuous phase is viscous. It is suitable for spherical solid particles, liquid droplets, and small-diameter bubbles.

Shear Lift Force

The Shear Lift Force model calculates the force acting on a particle moving relative to a fluid where there is a velocity gradient in the fluid orthogonal to the relative motion. The Shear Lift Force model is available for both Lagrangian and DEM particles. STAR-CCM+ calculates the shear lift force using a shear lift coefficient. You can define the lift coefficient using a field function, the Saffman method, or the Sommerfeld method. See Eqn. (2998) and Eqn. (2999).

Shear Lift Coefficient Properties

Method

Choose from the following options:

Field function

Define your own field function to specify the shear lift force coefficient.

Saffman

Use the definition of Saffman [633] provided as Eqn. (2998). This method recovers the original Saffman asymptotic solution for low Reynolds numbers.

Sommerfeld

Use the definition of Sommerfeld [634] provided as Eqn. (2999). This definition covers a broader range of Reynolds numbers than the Saffman definition

Pressure Gradient Force

This model activates the pressure gradient forces acting on the surface of the particle, as given by Eqn. (2988).

Whether the **Auto-select recommended models** option is activated or deactivated, the Pressure Gradient Force model is selected automatically when you select the Material Particles phase model. You can deactivate the **Pressure Gradient Force** model from the *Enabled Models* group box.

Virtual Mass

The Virtual Mass model calculates the force that is exerted on a material particle as it accelerates the surrounding continuous phase, Eqn. (2989).

The name virtual mass comes from the fact that it is equivalent to increasing the particle mass; the terms apparent or added mass are sometimes also encountered. The virtual mass is proportional to the mass of the continuous phase the particle displaces. Therefore, virtual mass is more important when the continuous phase density is significant compared to the particle density. For example, it can be important for gas bubbles in a liquid.

User-Defined Body Force

This model allows you to customize the Eqn. (2967) by adding a term to it. The User-Defined Body Force model stipulates how the user-defined body force is defined; see Eqn. (3008). It is a vector profile, with a reduced set of method options.

User Body Force Properties

You defined the body force using the common properties of a vector profile. See Scalar and Vector Quantities.

Charged Particle Models

Particles can enter the domain bearing electrical charges or, under some circumstances, can acquire electric charges after entering, resulting in additional forces acting on the particles. Charged particles appear in many applications, such as electrostatic spray painting and electrostatic precipitation in air pollution control.

To model particles that are already charged when they enter the domain, STAR-CCM+ uses the Charged Particles model in combination with the Coulomb Force model.

To model particles that acquire a charge after entering the domain, STAR-CCM+ uses the Charged Particles model in combination with the Field Charging model.

Selecting Charge Models

To select one of the charged particle models:

- Observe the following prerequisites:
  - Make sure that no mass transfer models are activated.
  - Make sure that the Electrostatic Potential model is already selected in the physics continuum models.
  - Make sure that the Material Particles model is already selected in the phase models.
- To select the Coulomb Force model, select Coulomb Force from the Optional Particle Forces group. The Charged Particles model is selected automatically.
- To select the Field Charging model, first select Charged Particles from the Optional Models group, then select Field Charging from the Optional Models group.
- For cases where charged particles collide or combine with other particles, consider using the NTC Collision model.

Charged Particles Model

The Charged Particle model is the base model that must be selected in order to simulate charge on particles. This model initializes charge on the parcels and adds Particle Charge to the injector values. As the base model, this model is required by the other two models and does not need any electrical properties.

Provided By	**[phase]** > **Models** > *Optional Models*	
Example Node Path	**[phase]** > **Models** > **Charged Particles**	
Requires	Incompatible with mass transfer models. *Optional Models*: **Electromagnetism** *Electrostatic Potential*: **Electrostatic Potential** In the phase models, *Particle Type*: **Material Particles**	
Activates	Physics Models	*Optional Particle Forces*: **Coulomb Force** selected automatically.
	Injectors	**Injector Values** > **Particle Charge**. See Injector Values Reference.
	Field Functions	**Particle Charge**. See Lagrangian Multiphase Field Function Reference.

Once the particles have a charge, a force acts on them if they are in an electric field, using the Coulomb Force model. If the particle charge changes over time by field charging, use the Field Charging model.

The implementation of this model in the current version of STAR-CCM+ is subject to the following assumptions and restrictions:

- If the Two-Way Coupling model is not activated, the charge on the particles does not influence the electric field. This condition can be desirable to save CPU time, provided the charge on the particles makes a negligible contribution to the electric field distribution.

  The Two-Way Coupling model does not get selected automatically with Coulomb Force—it must be selected manually.
- If the Two-Way Coupling model is not activated, particles do not attract or repel each other based on their charge.
- Once initialized, the charge per unit mass of each particle is fixed unless the Field Charging model is selected. During droplet breakup, this value is conserved.

Coulomb Force Model

Theory	The Coulomb Force models the electrostatic force acting on charged particles. This force is modeled as $q\mathbf{E}$ where q is the particle charge and $\mathbf{E}$ is the electric field.	
Provided By	**[phase]** > **Models** > *Optional Particle Forces*	
Example Node Path	**Lagrangian Multiphase** > **Lagrangian Phases** > **[phase]** > **Models** > **Coulomb Force**	
Requires	*Optional Models*: **Charged Particles**	
Activates	Physics Models	*Optional Models*: **Charged Particles** selected automatically.

		If **Charged Particles** is deselected, **Coulomb Force** is deselected automatically.

Field Charging Model

Field Charging models the charging of particles by an electric field that comes from a unipolar charge distribution in the carrier phase. It is the dominant mechanism of particle charging for particles larger than 1 micron.

When the Field Charging model is active, it requires the electrical permittivity property ϵ. This property applies to the material used in the Lagrangian phase. The model also requires inputs from the user in the form of the ion mobility and space charge density.

Theory	See Field Charging.	
Provided By	**[phase]** > **Models** > *Optional Models*	
Example Node Path	**Lagrangian Multiphase** > **Lagrangian Phases** > **[phase]** > **Models** > **Charged Particles**	
Requires	*Optional Models*: **Charged Particles**	
Activates	Physics Models	*Optional Particle Forces*: **Coulomb Force** selected automatically.
	Model Controls (child nodes)	**Ion Mobility**. See Ion Mobility Properties **Ion Space Charge Density**. See Ion Space Charge Density Properties
	Injectors	**Injector Values** > **Particle Charge**. See Injector Values Reference.
	Field Functions	**Particle Permittivity**. See Lagrangian Multiphase Field Function Reference.

Ion Mobility Properties

Ion mobility K is the terminal velocity of ions through a given medium in response to a unit electric field. It is a factor in field charging. See Eqn. (3007). Set this property using a Method from the scalar profile options provided.

Dimensions

Dimensions of the modeled quantity (read-only).

Method

Selects the method that is used for specifying the ion mobility K, in Eqn. (3007), from a subset of scalar profile options.

Ion Space Charge Density Properties

Ion space charge density ρ is the electrical charge from ions, per unit volume. It is a factor in field charging. See Eqn. (3007).

Dimensions

Dimensions of the modeled quantity (read-only).

Method

Selects the method that is used for specifying the ion space charge density, ρ in Eqn. (3007), from a subset of scalar profile options.

Lagrangian Energy Model

The Lagrangian Energy model activates the energy conservation equation for the particles.

Theory	See Particle Energy Balance.	
Provided By	**[phase]** > **Models** > *Optional Models*	
Example Node Path	**[phase]** > **Models** > **Energy**	
Requires	*Particle Type*: **Material Particles**	
Activates	Physics Models	*Optional Energy Sources*
	Model Controls (child nodes)	**Heat Transfer Coefficient**. See Heat Transfer Coefficient Properties.
	Field Functions	**Particle Specific Heat**, **Particle Temperature**. See Lagrangian Multiphase Field Functions Reference.

Heat Transfer Coefficient Properties

The heat transfer coefficient is used to formulate the surface heat transfer, Eqn. (3025).

Dimensions

The dimensionality of the heat transfer coefficient (read-only).

Method

Choose from the following options:

Ranz-Marshall

Defines the heat transfer coefficient using the Ranz-Marshall correlation. This correlation is available only when the continuous phase is viscous. It is suitable for spherical particles up to a Reynolds number of ~5000.

Field Function

Defines the heat transfer coefficient using a field function (typically user defined).

User-Defined Energy Source Model

The User-Defined Energy Source model adds an optional user-defined source term to the particle energy equation.

See Eqn. (3033).

Theory	See Other Sources of Heat Transfer.	
Provided By	**[phase]** > **Models** > *Optional Energy Sources*	
Example Node Path	**[phase]** > **Models** > **User-Defined Energy Source**	
Requires	*Optional Models*: **Energy**	
Activates	Model Controls (child nodes)	**User Energy Source**. See User Energy Source Properties.

User-Defined Energy Source Properties

The **User Energy Source** stipulates how the user-defined energy source Q_s in Eqn. (3033) is defined. Set as a scalar profile using a reduced set of method options.

NTC Collision Model

The NTC (No Time Counter) collision detection model simulates droplet collisions in transient simulations.

Theory	See NTC Algorithm.	
Provided By	**[phase]** > **Models** > *Collision Detection*	
Example Node Path	**[phase]** > **Models** > **NTC Collision Model**	
Requires	In the Lagrangian phase: *Particle Type*: **Material Particles** *Particle Shape*: **Spherical Particles** (selected automatically) *Material*: **Liquid** or **Multi-Component Liquid**	
Activates	Materials	**Surface Tension** for **Liquid** under the Lagrangian phase.
	Field Functions	**Droplet Surface Tension**. See Lagrangian Multiphase Field Function Reference.

Droplet Evaporation Models

This section describes the Quasi-Steady Evaporation, User-Defined Evaporation, and Droplet Evaporation models. The first two apply to single-component liquids, while the last can apply to multi-component liquids.

Quasi-Steady and User-Defined Evaporation Models for Single-Component Droplets

The Quasi-Steady model adds the quasi-steady evaporation rate to the Eqn. (3015) equation for the droplets.

The User-Defined model supplies the rate of mass transfer in the Eqn. (3015) equation for the particles.

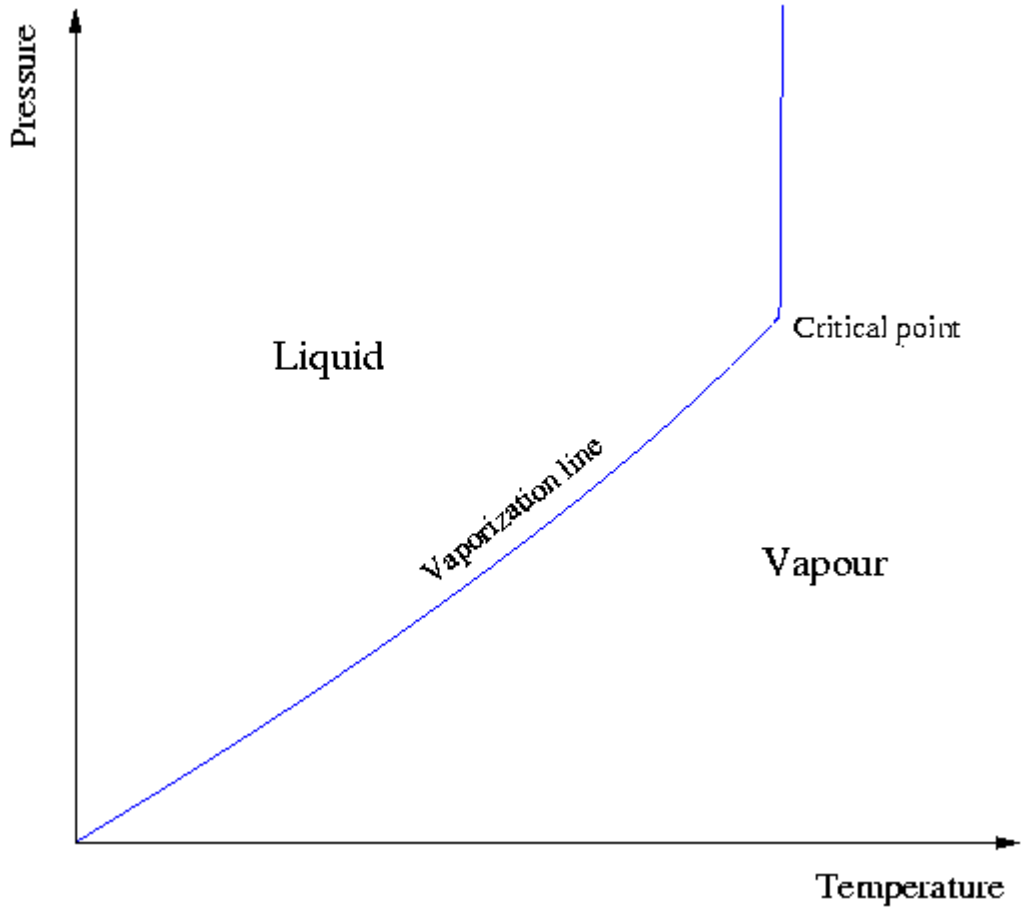
These models are only available when the particle material is a single-component liquid. These models are compatible with the Real Gas equation of state models.

Model Names	Quasi-Steady Evaporation	
	User-Defined Evaporation	
Theory	See Droplet Evaporation.	
Provided By	**[phase]** > **Models** > *Mass Transfer*	
Example Node Path	**[phase]** > **Models** > **Quasi-Steady Evaporation**	
Requires	In the Lagrangian phase: *Particle Type*: **Material Particles** *Particle Shape*: **Spherical Particles** (selected automatically) *Material*: **Liquid**	
Properties	Key properties for Quasi-Steady Evaporation are: *Vapor Component*, *Condensation*, *Heat-Transfer Limited Mode*. See Quasi-Steady Evaporation Properties. Key property for User-Defined Evaporation is: *Vapor Component*, the gas component in the continuous phase which represents the droplet vapor.	
Activates	Physics Models	*Energy*
	Model Controls (child nodes)	Quasi-Steady Evaporation: Sherwood Number User-Defined Evaporation: User Evaporation Rate
	Boundary Inputs	See Quasi-Steady Evaporation Boundary Settings.
	Field Functions	Quasi-Steady Evaporation: **Droplet Latent Heat of Vaporization**, **Droplet Saturation Pressure** User-Defined Evaporation: **Droplet Latent Heat of Vaporization**, **Particle Specific Heat**, **Particle Temperature** See Lagrangian Multiphase Field Functions Reference.

Quasi-Steady Evaporation

The Quasi-Steady Evaporation model allows liquid particles (that is, droplets) to lose mass through evaporation. It also optionally allows droplets to gain mass through condensation. The driving force for evaporation and condensation is the departure from equilibrium of the liquid-vapor system. This system consists of the droplet and the gas component of the continuous phase that is specified as the droplet vapor. This equilibrium is defined on an idealized phase diagram that is shown below. Two material properties

determine the vaporization line: the saturation pressure and the critical temperature, which are added to the material properties of the Lagrangian phases.



Besides the driving force, two other important parameters in determining the droplet evaporation rate are the molecular diffusivity of the gas component and the Sherwood Number of the droplet. The **Quasi-Steady Evaporation** node has a sub-node for the latter.

When this model is active, the enthalpy difference method for specifying latent heat of vaporization becomes available.

Quasi-Steady Evaporation Properties

Vapor Component

The gas component in the continuous phase which represents the droplet vapor.

Condensation

When **On**, specifies that condensation is permitted in addition to evaporation. When **Off**, only evaporation is permitted. The default is **On**.

Heat-Transfer Limited Mode

When **On**, specifies that the heat-transfer limited evaporation mode is permitted. This mode is used when the vapor at the droplet surface is saturated and subcritical. See the Droplet Evaporation. When **Off**, the vapor diffusion limited mode is applied (with the vapor mole fraction at the droplet surface fixed to 0.99999). The default is **On**.

Quasi-Steady Evaporation Boundary Settings

Wall

Vaporize

Selecting the Quasi-Steady Evaporation model adds **Vaporize** to the available set of modes. See Vaporize under "Understanding Boundary Interaction Modes."

User-Defined Evaporation

The User-Defined Evaporation model allows you to define the evaporation rate using a constant, a table, or a field function. Make the evaporation rate positive for evaporation, negative for condensation. Use units of kg/s. If the Two-Way Coupling model is active, the evaporated mass enters the continuous phase, provided the destination gas component has been set in the *Vapor Component* property.

When this model is active, the enthalpy difference method for specifying latent heat of vaporization becomes available.

User Evaporation Rate Properties

The evaporation rate that this node provides is used in Eqn. (3015) as the rate of mass transfer. It is entered as a scalar profile.

Method

Selects the method for entering the scalar value.

Dimensions

The dimensionality of the value (read-only).

Droplet Evaporation for Multi-Component Droplets

The Droplet Evaporation model adds the to the component mass conservation equations for the droplets. This model is only available when the droplet material is a multi-component liquid. This model is compatible with the Real Gas equation of state models. See What Is the Real Gas Model?

Theory	See Droplet Evaporation.
Provided By	**[phase]** > **Models** > *Optional Models*
Example Node Path	**Lagrangian Multiphase** > **Lagrangian Phases** > **[phase]** > **Models** > **Droplet Evaporation**
Requires	In the Lagrangian phase:

	Particle Type: **Material Particles** *Particle Shape*: **Spherical Particles** (selected automatically) *Material*: **Multi-Component Liquid** *Species*: **Species** (selected automatically)	
Properties	Key properties are: *Vapor Component*, *Condensation*, *Heat-Transfer Limited Mode*. See Droplet Evaporation Properties.	
Activates	Physics Models	*Energy*
	Model Controls (child nodes)	Sherwood Number
	Boundary Inputs	See Droplet Evaporation Boundary Settings.

The Droplet Evaporation model allows some or all of the components within the droplet to lose mass through evaporation. It also optionally allows droplets to gain mass through condensation. The driving force for evaporation and condensation is the departure from equilibrium of the liquid-vapor system. This system consists of the droplet and the gas components that are specified as exchanging mass. That is, gas components can be specified as representing the vapor of selected droplet components. The equilibrium state of each component is defined on an idealized phase diagram as for the equivalent single-component droplet evaporation model.

Besides the driving force, two other important parameters in determining the droplet evaporation rate are the Sherwood Number of the droplet and the molecular diffusivity of the gas. (The molecular diffusivity is assumed equal for all components.) The **Droplet Evaporation** node has a sub-node for the latter.

When this model is active, the enthalpy difference method for specifying latent heat of vaporization becomes available.

Sherwood Number Properties

The Sherwood number is used to formulate the quasi-steady evaporation rate for both single-component and multi-component droplets.

Method

Selects the method for specifying the Sherwood number.

Ranz-Marshall

Defines the Sherwood number using the Ranz-Marshall correlation. This correlation is available only when the continuous phase is viscous. It is suitable for spherical particles up to a Reynolds number of ~5000.

Field Function

Defines the Sherwood number using a field function (typically user defined). A field function method node is added as a child to this node.

Droplet Evaporation Properties

Vapor Component

The gas component in the continuous phase which represents the droplet vapor.

Condensation

When **On**, specifies that condensation is permitted in addition to evaporation. When **Off**, only evaporation is permitted. The default is **On**.

Heat-Transfer Limited Mode

When **On**, specifies that the heat-transfer limited evaporation mode is permitted. This mode is used when the vapor at the droplet surface is saturated and subcritical. See the Droplet Evaporation. When **Off**, the vapor diffusion limited mode is applied (with the vapor mole fraction at the droplet surface fixed to 0.99999). The default is **On**.

Droplet Evaporation Boundary Settings

Wall

Rebound

Selecting the Quasi-Steady Evaporation model adds **Rebound** to the available set of modes. See Rebound under "Understanding Boundary Interaction Modes."

Vaporize

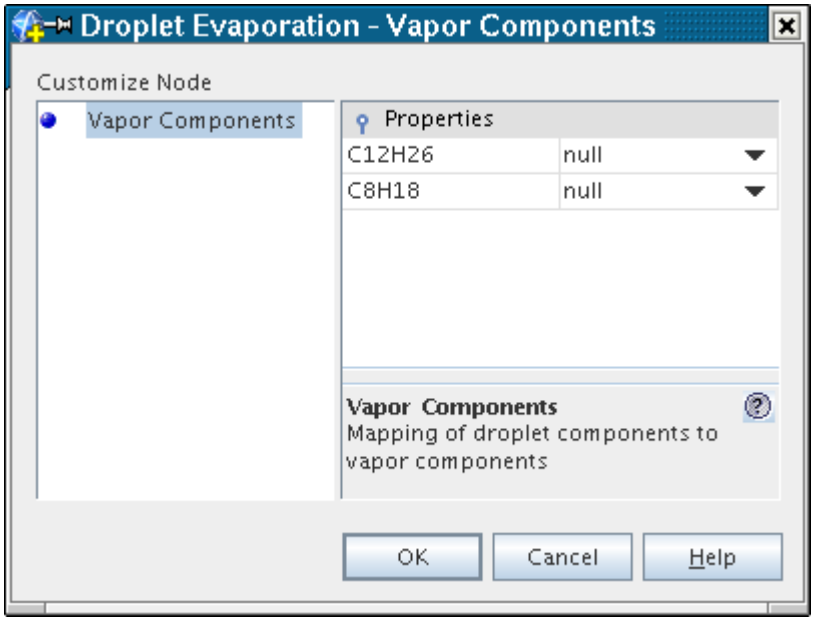
Selecting the Quasi-Steady Evaporation model adds **Vaporize** to the available set of modes. See Vaporize under "Understanding Boundary Interaction Modes."

Mapping Droplet Components to Gas Components

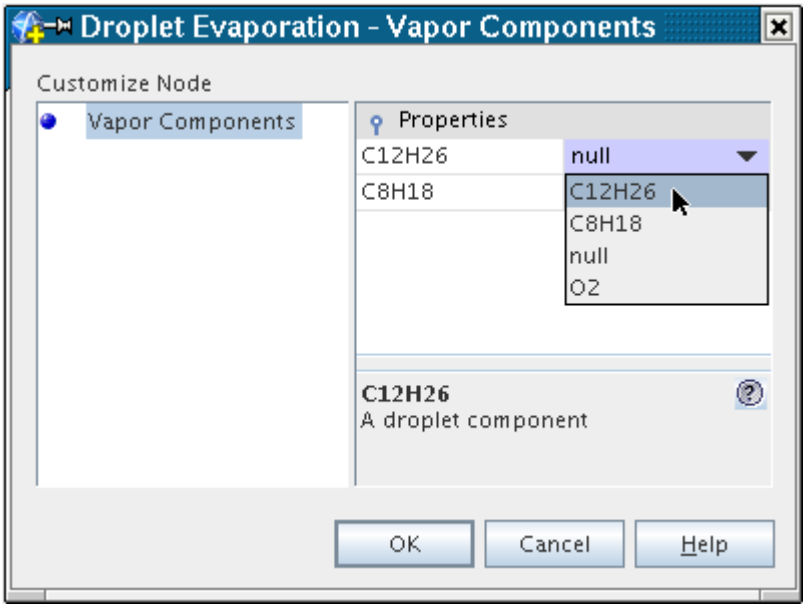
The *Vapor Components* property in the *Properties Window* for the **Droplet Evaporation** node defines how droplet components are mapped to gas components in the continuous phase. The mapping is defined as follows:

1. Click ... (**Custom Editor**) to the right of the *Vapor Components* property.

   The *Droplet Evaporation - Vapor Components* dialog appears.



The liquid components in the droplet are listed on the right side of the dialog. These components can be mapped to available components in the continuous gas phase by using the drop-down menus next to each droplet component as shown below.



2. Click **OK** to close the dialog when all components have been mapped.

Lagrangian Species Model

The Species model enables description the composition of multi-component particles by mass fraction.

The Lagrangian Species model is automatically activated when a multi-component material model is selected for the Lagrangian phase. It activates the interface elements for multi-component particles and droplets.

Theory	See Particle Mass Balance.	
Provided By	**[phase]** > **Models** > *Species*	
Example Node Path	**[phase]** > **Models** > **Species**	
Requires	In the Lagrangian phase: *Particle Type*: **Material Particles** *Material*: one of **Multi-Component Coal**, **Multi-Component Liquid**, or **Multi-Component Particle**	
Activates	Physics Models	*Equation of State*
	Injector Inputs	Species Specification in injector conditions Species Mass Fraction in injector values These become active after injectors are activated.
	Field Functions	**Particle Mass Fraction of [component]**, **Particle Molar Concentration of [component]**, **Particle Mole Fraction of [component]** These become active after the components of the material are specified. See Lagrangian Multiphase Field Functions Reference.

Two-Way Coupling Model

For Lagrangian and DEM simulations, the two-way coupling model allows the particle phase to exchange mass, momentum, and energy with the continuous phase.

Without the Two-Way Coupling model, the default “one-way coupling” means that the continuous phase influences the particles through terms such as drag in the momentum equation and heat transfer in the energy equation. However, there is no reverse effect. With the Two-Way Coupling model, the reverse effect is accounted for, and Lagrangian source terms appear in continuous phase equations. Typically, this effect becomes more important at higher particle loadings (see [576], for example).

Another use for the Two-Way Coupling model is with Coulomb Force. When both these models are activated, the electric charge on the particles influences the electric field.

Theory	See Two-Way Coupling with the Continuous Phase.	
Provided By	**[phase]** > **Models** > *Optional Models*	
Example Node Path	**[phase]** > **Models** > **Two-Way Coupling**	
Requires	*Particle Type*: **Material Particles**	
Activates	Solvers	Two-Way Coupling Solver.
	Field Functions	**Void Fraction**. See Lagrangian Multiphase Field Functions Reference .

Turbulent Dispersion Model

The Turbulent Dispersion model collaborates with the Reynolds-Averaged Navier Stokes turbulence model in the physics continuum.

This model employs a random walk technique to synthesize the fluctuating nature of the turbulent velocity field in the continuous phase. The introduction of a random element means that turbulent flows are likely to need a higher number of parcels than similar laminar flows, to provide statistically meaningful results.

Theory	See Turbulent Dispersion.
Provided By	**[phase]** > **Models** > *Optional Models*
Example Node Path	**[phase]** > **Models** > **Turbulent Dispersion**
Requires	*Particle Type*: **Material Particles**
Properties	*Exact Eddy Interaction Time* When this property is **On**, the model restricts substep durations so that they exactly sum to the eddy interaction time. Although this restriction improves accuracy, computational speed is reduced. Without this property, the model does not allow the substep durations to become less than the limit implied by the minimum Courant number. (See Lagrangian Multiphase Solver Reference.) However, the model still aims for the closest possible approximation to the exact sum. The default is **Off**.

Erosion Model

STAR-CCM+ provides erosion models for both Lagrangian and DEM frameworks. In the Lagrangian framework, STAR-CCM+ only models erosion due to impact; the DEM framework additionally models abrasion.

Theory	See Erosion.	
Provided By	**[phase]** > **Models** > *Optional Models*	
Example Node Path	**[phase]** > **Models** > **Erosion**	
Requires	*Particle Type*: **Material Particles**	
Activates	Boundary Inputs	See Erosion Boundary Settings.
	Field Functions	**Erosion Rate**. See Lagrangian Multiphase Field Functions Reference .

The erosion model predicts the rate of erosion from particle impact on solid walls. Erosion rate is defined as the mass of wall material removed per unit area per unit time. It is provided as a field function when the Erosion model is activated.

In calculating the erosion rate, STAR-CCM+ accumulates the damage that each particle impact causes. This calculation is done by selecting a correlation for the erosion ratio, that is, the mass of wall material eroded per

unit mass of impinging particles. Erosion damage depends on both the material of the walls and the material of the particles. Hence, the erosion ratio correlation is a boundary condition which can optionally be specified differently for each boundary. Further details are given in the formulation.

Any correlation for the erosion ratio is specific to a particular wall material, particle material, and carrier fluid. It can also be specific to a particular range of flow conditions. Correlations that are used outside their intended application can be inaccurate. For this reason, take care to ensure that a suitable correlation is chosen and its coefficients are specified accordingly. It is prudent to compare the predictions of a number of suitably chosen correlations, unless there is a strong reason to trust a particular one.

Erosion modeling does not necessarily predict material loss with a consistent degree of accuracy. If estimating component lifetime is important, then obtain additional experimental validation to assess the effectiveness of the erosion model under appropriate conditions.

Erosion Boundary Settings

Wall

Impact Wear

Specifies the method that is used to calculate the impact wear component of the erosion rate:

- Ahlert Correlation
- DNV Correlation
- Field Function
- Neilson-Gilchrist Correlation
- Oka Correlation

See Erosion Correlations.

See Also:

Modeling Erosion in the Discrete Element Method

Residence Time

Residence time is the length of time a particle has been in the simulation. Residence time data is available only on existing particles, currently in the simulation. The Residence Time model makes particle residence time available as a field function.

Provided By	**[phase]** > **Models** > *Optional Models*	
Example Node Path	**[phase]** > **Models** > **Residence Time**	
Requires	*Optional Models*: **Lagrangian Multiphase**	
Activates	Field Functions	**Particle Flow Rate**, **Particle Residence Time**. See Discrete Element Method Field Functions Reference.

Particle Radiation Model

The Particle Radiation model accounts for the scattering and absorption of thermal radiation due to particles when the Participating Media Radiation model is active in the continuous phase. It also calculates the radiation that is emitted from the particles themselves.

Theory	Radiative Heat Transfer, Participating Media Radiation
Provided By	**[phase]** > **Models** > *Optional Models*
Example Node Path	**[phase]** > **Models** > **Particle Radiation**
Requires	*Particle Type*: **Material Particles** *Optional Models*: **Energy**
Properties	Key properties are: *Qap* and *Qsp*. See Particle Radiation Properties.

Particle Radiation Properties

Two properties, the absorption efficiency and the scattering efficiency, govern absorption and scattering for each Lagrangian phase. These properties are specified without respect to wavelength, and so even if surface and continuum radiative properties are spectral, the particle properties are gray. Scattering is modeled as isotropic. You can use the particle radiative energy source field function to post-process radiative contribution from particle phases.

Qap

Particle absorption efficiency, Q_a in Eqn. (2132).

Qsp

Particle scattering efficiency, Q_s in Eqn. (2133).

TAB Distortion Model

The TAB Distortion model is used to calculate the distortion of liquid droplets under the action of aerodynamic forces.

This model uses the Taylor analogy [591], which represents a distorting droplet as a damped spring-mass system. It considers only the fundamental mode of oscillation of the droplet. The displacement and velocity of the mass in the spring-mass system correspond to representative distortion and rate of distortion for the droplet. These quantities are made available as the **TAB Distortion** and **TAB Distortion Rate** field functions. While these quantities are sometimes of interest themselves, the main purpose of the TAB Distortion model is to make these quantities available to other models, such as:

- The TAB Breakup model, which uses the **TAB Distortion** to determine when and how droplets breakup under the action of aerodynamic forces.
- The Liu drag coefficient method, which uses the **TAB Distortion** to modify the drag coefficient of distorted droplets consistent with their non-spherical form. If the drag model is active, activating the TAB distortion model also activates the Liu drag coefficient method.

Theory	See TAB Distortion and Breakup.	
Provided By	**[phase]** > **Models** > *Optional Models*	
Example Node Path	**[phase]** > **Models** > **TAB Distortion**	
Requires	*Particle Type*: **Material Particles** *Particle Shape*: **Spherical Particles** (selected automatically) *Material*: **Liquid**	
Properties	Key properties are: *WeCrit*, *Cd*, and *Ck*. See TAB Distortion Properties.	
Activates	Field Functions	**Droplet Dynamic Viscosity**, **Droplet Surface Tension**, **Droplet Weber Number**, **TAB Distortion**, and **TAB Distortion Rate**. See Lagrangian Multiphase Field Function Reference.

TAB Distortion Properties

WeCrit

Critical Weber number. See Eqn. (3102).

Cd

Damping coefficient. See Eqn. (3102).

Ck

Stiffness coefficient. See Eqn. (3102).

Lagrangian Passive Scalars

In the Lagrangian framework, passive scalars are user-defined variables of arbitrary value, assigned to fluid phases or individual particles. They are passive because they do not affect the physical properties of the simulation.

An intuitive way to think of passive scalars is as marker paint on a particle or parcel, but with numerical values instead of colors, and with no appreciable mass or volume.

More elaborate physical effects can be modeled with passive scalars on particles, such as change of composition through radioactive decay.

You can use more than one passive scalar in a simulation, for example when analyzing the mixing of multiple streams of particles with the same properties. Each particle can be tagged with a scalar indicating its source.

The value of a passive scalar on a particle is set on the injector from which the particle enters the simulation. As a result, particles of the same phase can be given different passive scalars at different injectors. See the Passive Scalar entry in the Injector Values Reference.

To change the value of a passive scalar during a simulation, apply a user source on the scalar. When selecting to add a source you must either choose the source as a mass flux (mass/volume/time), or as a scalar rate (time^{-1}) where the density will be inferred from the Lagrangian phase. For massless particles, only the scalar rate option is avalable.

Lagrangian passive scalars can also interact with Eulerian passive scalars, to allow transfer of scalar values from one to the other.

For uses outside the scope of Lagrangian modeling, see Modeling Passive Scalars.

Theory	See Passive Scalar Interactions.	
Provided By	**[phase]** > **Models** > *Optional Models*	
Example Node Path	**[phase]** > **Models** > **Passive Scalar**	
Requires	*Particle Type*: **Material Particles** or **Massless Particles**	
Properties	Key properties are: *User Source Option*. See Passive Scalar Properties.	
Activates	Model Controls (child nodes)	Passive Scalars manager node Passive Scalar Source node
	Solvers	Lagrangian Passive Scalar Solver

Passive Scalars Properties

User Source Option

Creates a new node, Passive Scalar Source, under the **Passive Scalar** model node, used to specify a source term.

No sources active

Use no source term. The default.

Mass Flux

Use a source term. See The Lagrangian Passive Scalar Source Node. This option is not available for massless particles.

Scalar rate with inferred density

Use a source term as above, and automatically use the density of the material composing the phase.

The Lagrangian Passive Scalars Manager Node

This node allows you to organize multiple passive scalars and their properties. An individual **Passive Scalar** node under the **Passive Scalars** manager node represents a specific passive scalar in the simulation tree. Individual passive scalar nodes can be renamed or deleted.

To create an individual passive scalar node:

1. Right-click the **Passive Scalar** > **Passive Scalars** manager node and choose **New**.
2. Set the properties of the newly created **Passive Scalar** node.

Individual Passive Scalar Properties

Number

A property that indicates the number of passive scalar materials preceding the current one. For example, for the first passive scalar material that you create, this property would equal zero. These values remain unchanged even when you delete one or more of these objects.

Maximum clip value

Sets the maximum scalar value used for clipping.

Minimum clip value

Sets the minimum scalar value used for clipping.

Clip

Clipping prevents scalar values from exceeding limits you set.

Off

No clipping applied.

Below Min, Above Max

Prevents scalar values from rising above the *Maximum clip value*, and falling below the *Minimum clip value*.

Below Min

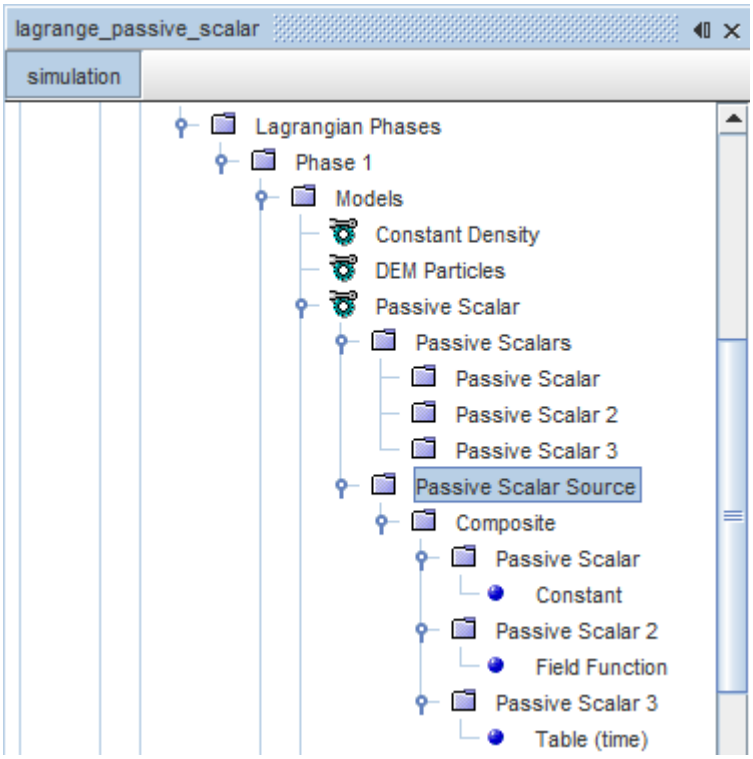
Prevents scalar values from falling below the *Minimum clip value*.

Above Max

Prevents scalar values from rising above the *Maximum clip value*.

The Lagrangian Passive Scalar Source Node

The optional **Passive Scalar Source** node is activated when the Passive Scalar property User Source Option is set to **Mass Flux** or **Scalar rate with inferred density**. The node specifies source terms for the scalars.



Properties

Method

Selects the method used for the source terms of the individual passive scalars.

Constant

Applies the constant vector of values to all individual passive scalars as their source term.

Composite

Allows you to determine a different type and vector of source term for each individual passive scalar.

Injection Table

Set the value of the passive scalars according to an injection table. The table specifies the location of injection points, the time of injection, and the value of each passive scalar separately. Used only with table injectors.

Dimensions

The dimensionality of the value, *Mass/Volume-Time* (read-only). The passive scalar itself is always dimensionless, but its source term has units referring to the density of the receiving phase over time.

Impingement Heat Transfer Model

The Impingement Heat Transfer model describes the heat transfer between impinging droplets and walls or fluid-solid interfaces. The model allows you to simulate the gradual cooling or heating of the wall, which can

affect behaviors such as reaction rates or crystallization within or beyond the wall. The cooling or warming effect of the droplets is slow and cumulative.

The Impingement Heat Transfer model accounts for heat transfer between droplets and walls using the Heat Penetration Coefficient multiplier b_w in Eqn. (3168). This value is represented in the simulation tree by the **HPC Multiplier** sub-node under the **Impingement Heat Transfer** node.

Theory	See Impingement Heat Transfer.	
Provided By	**[phase]** > **Models** > *Optional Energy Sources*	
Example Node Path	**[phase]** > **Models** > **Impingement Heat Transfer**	
Requires	*Particle Type*: **Material Particles** *Optional Models*: **Energy** *Material* : **Liquid** or **Multi-Component Liquid**	
Activates	Model Controls (child nodes)	HPC Multiplier
	Field Functions	**Droplet Dynamic Viscosity**. See Lagrangian Multiphase Field Function Reference.

HPC Multiplier Properties

Method

Selects the method that is used for specifying the Heat Penetration Coefficient multiplier b_w in Eqn. (3168) from a subset of scalar profile options.

Dimensions

Dimensions of the units that are associated with the HPC multiplier (read only).

Parcel Depletion Model

The Parcel Depletion model allows you to deplete (remove, delete) parcels you no longer wish to track.

Depletion is controlled by the **Depletion Criterion** property. While the Lagrangian solver is running, the depletion criterion is evaluated once per parcel per sub-step. Parcels for which the criterion evaluates to a non-zero value are depleted.

Note: Depletion is not conservative. The mass, momentum, energy, and so forth, of depleted parcels are removed from the simulation.

When the Track File model is also selected, the final states of depleted parcels are written to the track file.

Provided By	**[phase]** > **Models** > *Optional Models*
Example Node Path	**[phase]** > **Models** > **Parcel Depletion**

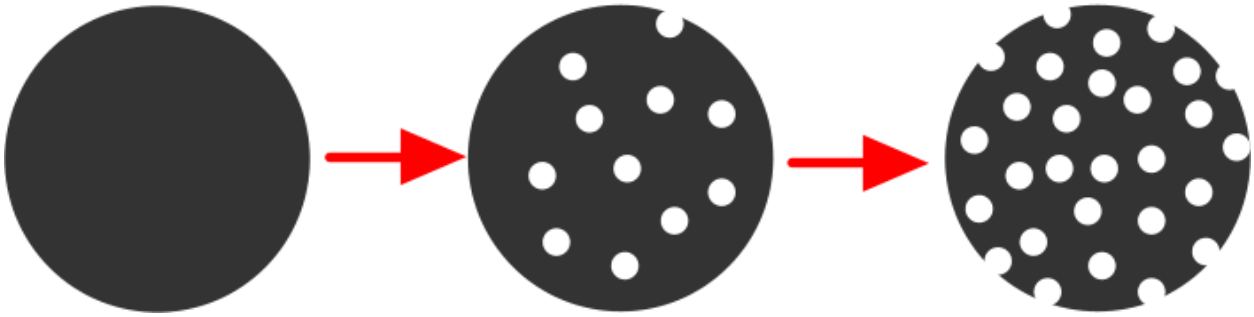
Requires	*Optional Models*: **Lagrangian Multiphase**
Properties	*Depletion Criterion* A user-set limit which determines whether to delete a parcel. Parcels for which this criterion returns a non-zero value are depleted. Example: `$ParticleDiameter < [value]` removes parcels for which the particle diameter is less than `[value]`.

Particle Porosity

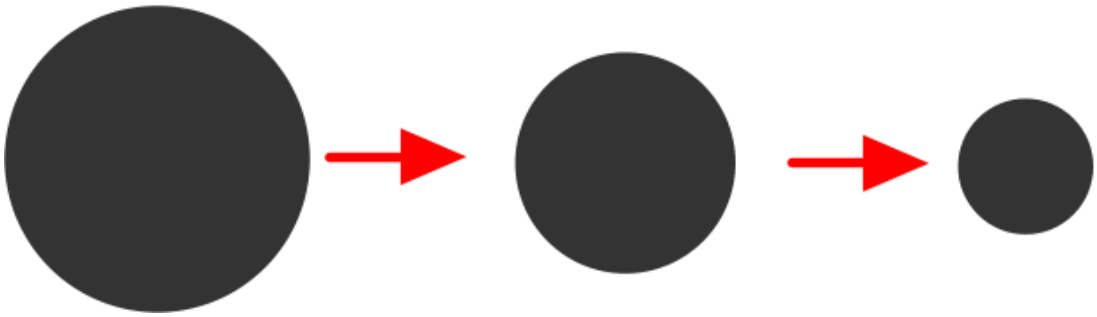
This optional Lagrangian phase model allows you to define how the porosity and diameter of a material particle changes in response to a coal combustion reaction or other particle reaction.

During most combustion reactions, particles lose mass. Loss of mass can occur by an increase in porosity or a decrease in the diameter of particles—or a combination of both. The Particle Porosity model allows you to specify how the porosity of particles changes throughout the course of combustion reactions and specify the maximum porosity above which you want to stop tracking particles.

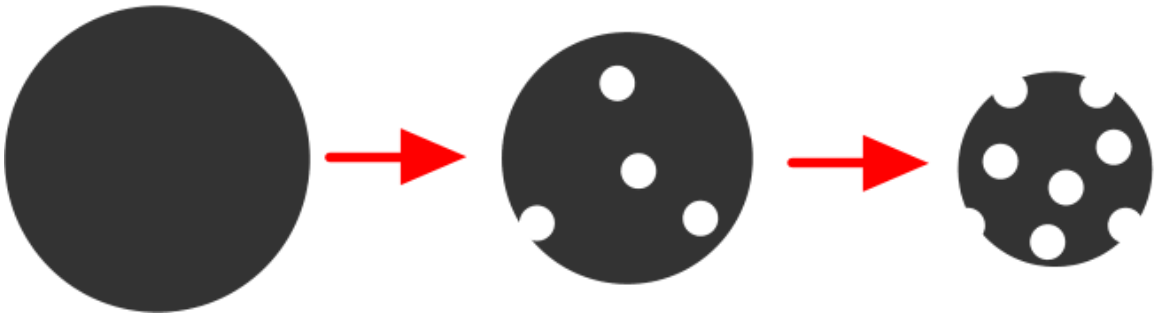
- Particles can lose mass by increasing porosity yet maintaining the original diameter ($\alpha = 0$):



- Particles can lose mass by decreasing in diameter yet maintaining zero porosity ($\alpha = 1$):



- Particles can lose mass by decreasing in diameter and increasing porosity ($0 < \alpha < 1$)



Theory	See Particle Porosity Formulation.	
Provided By	**[phase]** > **Models** > *Optional Models*	
Example Node Path	**[phase]** > **Models** > **Particle Porosity**	
Requires	*Material*: **Multi-Component Particle** or **Multi-Component Liquid** *Mass Transfer*: **Particle Reactions** or *Material*: **Multi-Component Coal** *Mass Transfer*: **Coal Combustion**	
Properties	*Maximum Porosity* Specifies the porosity above which particles are considered depleted.	
Activates	Model Controls (child nodes)	**Porosity Specification** > **Diameter-Based Exponent**. See Diameter-Based Exponent Properties.
	Field Functions	**Particle Porosity**. See Lagrangian Multiphase Field Functions Reference.

Diameter-Based Exponent Properties

Exponent, alpha

Specifies the value for the exponent α, which is used to calculate porosity θ in Eqn. (3182). A value of 1.0 implies that the particle diameter reduces proportional to the cube root of particle mass. However, a value of 0.0 implies that the particle diameter, and therefore volume, remains the same and particle porosity increases inversely to particle mass. For coal/char combustion setups, a value of 0.75 works well and for devolatilization reactions, a value of 0.0 works well.

Lagrangian Phase Boundary Conditions

The “boundary conditions” for a Lagrangian phase describe how particles behave when they impinge on a boundary.

This behavior varies according to the boundary interaction mode and whatever conditions and values are required at the boundary by the interaction mode and the Lagrangian phase models. For some boundary

types, this behavior follows obvious default modes with no additional user input. For other boundaries, however, some modeling is sometimes required, and so boundary conditions for each Lagrangian phase must be specified. You can choose to specify:

- Default boundary conditions which apply at each of these boundary types.
- Specific boundary conditions at a particular boundary.

Phase models that are associated with boundary interaction modes are:

- Bai-Gosman
- Bai-ONERA
- Droplet Film Transition
- Satoh
- Ice Accretion

When you work with the Discrete Element Method (DEM), a special DEM mode is available.

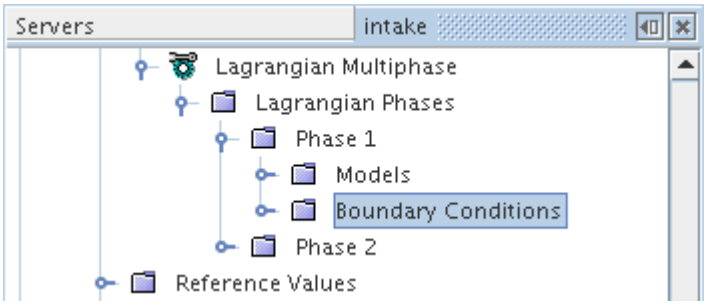
Contents:

Setting Default Lagrangian Phase Boundary Conditions
Setting Lagrangian Phase Boundary Conditions for a Specific Boundary
Setting a Phase Impermeable Boundary
Understanding Boundary Interaction Modes
Modeling Bai-Gosman Wall Impingement
Modeling Bai-ONERA Wall Impingement
Modeling Satoh Wall Impingement
Droplet Film Transition Model Reference
Modeling Ice Accretion
Boundary Values Reference

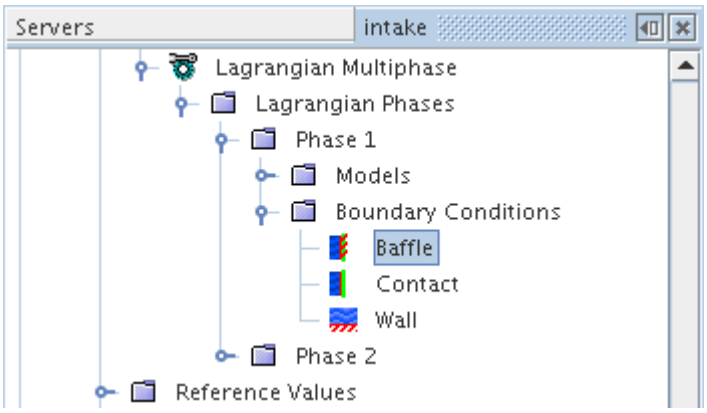
Setting Default Lagrangian Phase Boundary Conditions

This section describes how to set the default boundary conditions for a Lagrangian phase.

The **Boundary Conditions** of the Lagrangian phase are contained within the Boundary Conditions manager node for a Lagrangian phase.



It contains a sub-node for each of the boundary or interface types for which you can set the default boundary interaction mode. Boundary or interface types are only included if at least one boundary or interface of the type is present in the simulation.



The properties of a boundary type sub-node define the interaction mode of the Lagrangian phase for boundaries of that type, unless the mode is over-ridden at a specific boundary. Depending on the selected boundary interaction mode, additional sub-nodes can be added for boundary values.

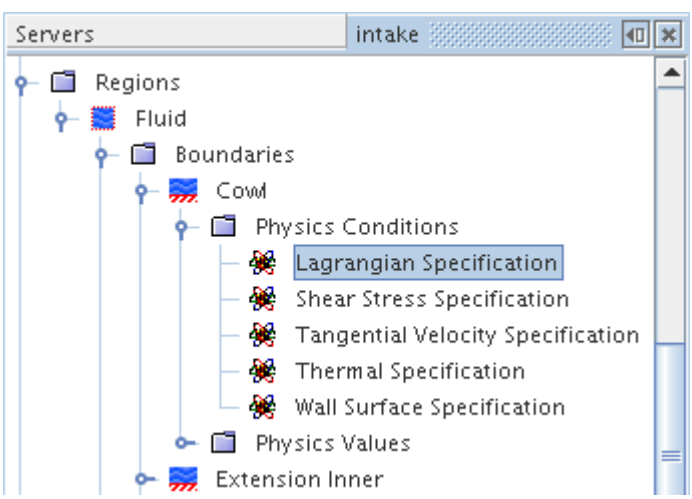
Lagrangian Phase Boundary Conditions Properties

Mode	Selects the boundary interaction mode for the Lagrangian phase at this boundary type. The actual list that you see depends on the compatibility with the boundary type and the selected Lagrangian phase models.

Setting Lagrangian Phase Boundary Conditions for a Specific Boundary

Lagrangian phase boundary conditions can be set at specific boundaries as well as for the entire phase.

The **Lagrangian Specification** node allows you to set these conditions. This node is a sub-node of the **Physics Conditions** node for the boundary of interest.



To set phase-specific conditions:

1. Select the **Lagrangian Specification** node.
2. Set *Method* to **Specify for Boundary**.

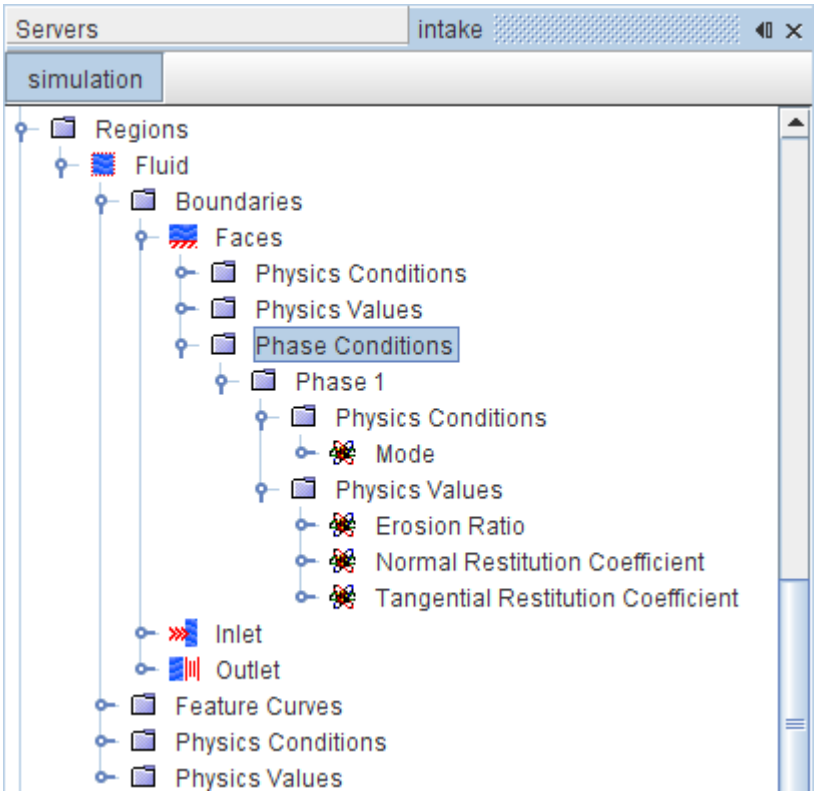
   The Phase Conditions manager node appears for the boundary.

Lagrangian Specification Properties

Method	Selects the method for the Lagrangian boundary condition specification.	
	Use Phase Defaults	Specifies that the default boundary conditions for all Lagrangian phases are used at this boundary.
	Specify for Boundary	Specifies that specific boundary conditions are provided for all Lagrangian phases at this boundary. A Phase Conditions sub-node is added as a child to the **Physics Values** node for this boundary.

What Is the Phase Conditions Manager?

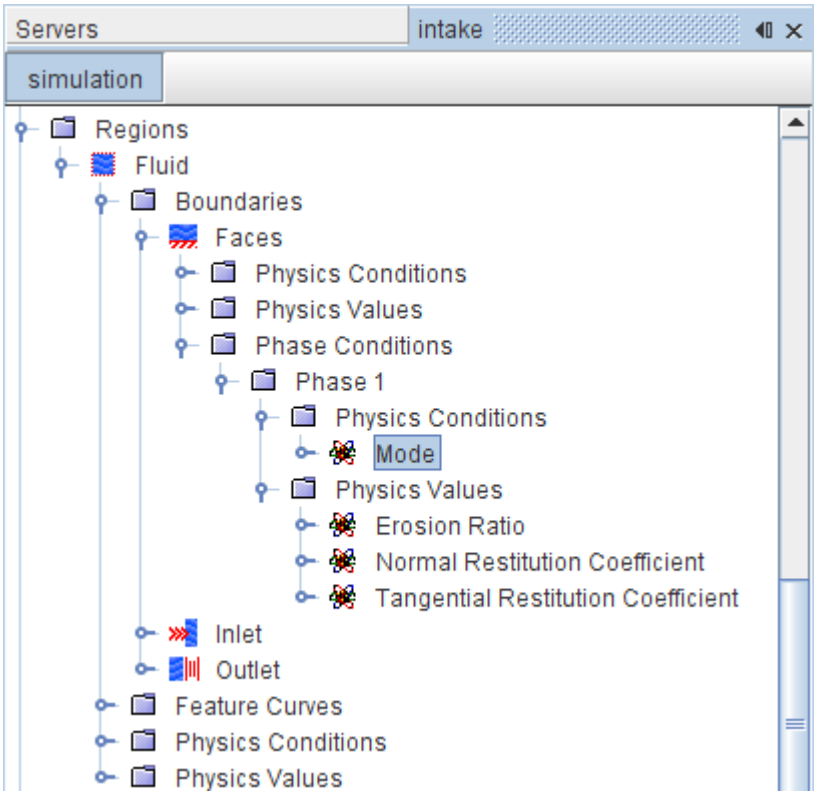
The **Phase Conditions** manager node is added to a **Physics Values** node of a boundary when you choose to specify Lagrangian phase boundary conditions specifically for that boundary.



There are sub-nodes of the **Phase Conditions** node representing each of the Lagrangian phases.

Selecting the Mode

The **Physics Conditions** > **Mode** node within each phase, which has properties, allows you to specify the boundary interaction mode at the boundary for that phase. Depending on the selected boundary interaction mode, additional sub-nodes can be added for boundary values.



Phase Boundary Interaction Mode Properties

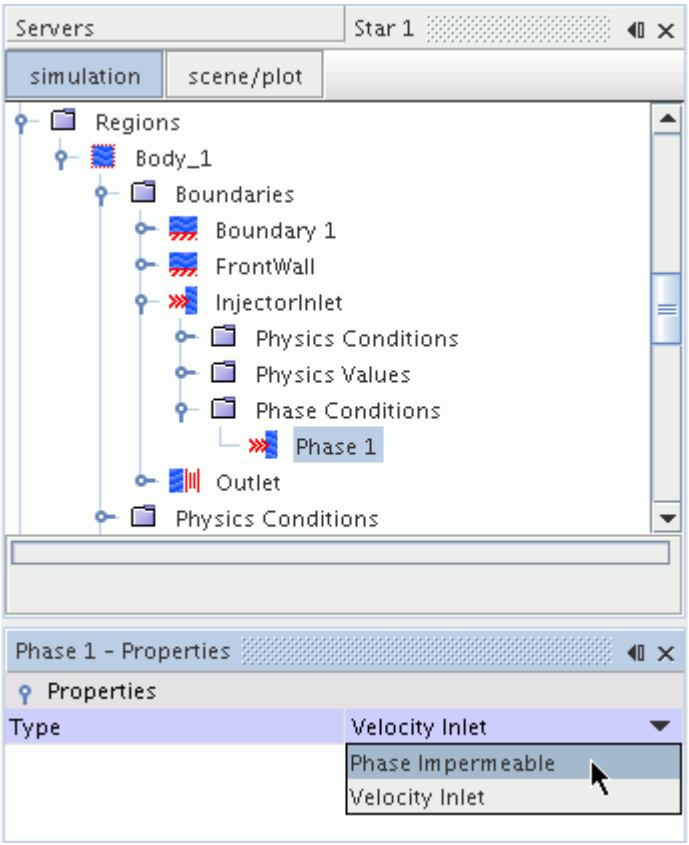
Active Mode	Selects the boundary interaction mode for the Lagrangian phase at this boundary. The actual list that you see depends on the compatibility with the boundary type and the selected Lagrangian phase models.

Setting a Phase Impermeable Boundary

Flow boundaries can be set to phase impermeable for Lagrangian multiphase flows. Phase impermeable means that particles do not pass through the boundary; instead they interact with the boundary as if it were a wall. Fluid in the continuous phase continues to pass through as normal.

To set a phase impermeable boundary:

1. On a flow boundary, select the phase representation of the boundary node within the **Phase Conditions** manager.
2. In the *Properties* window, set *Type* to **Phase Impermeable**.

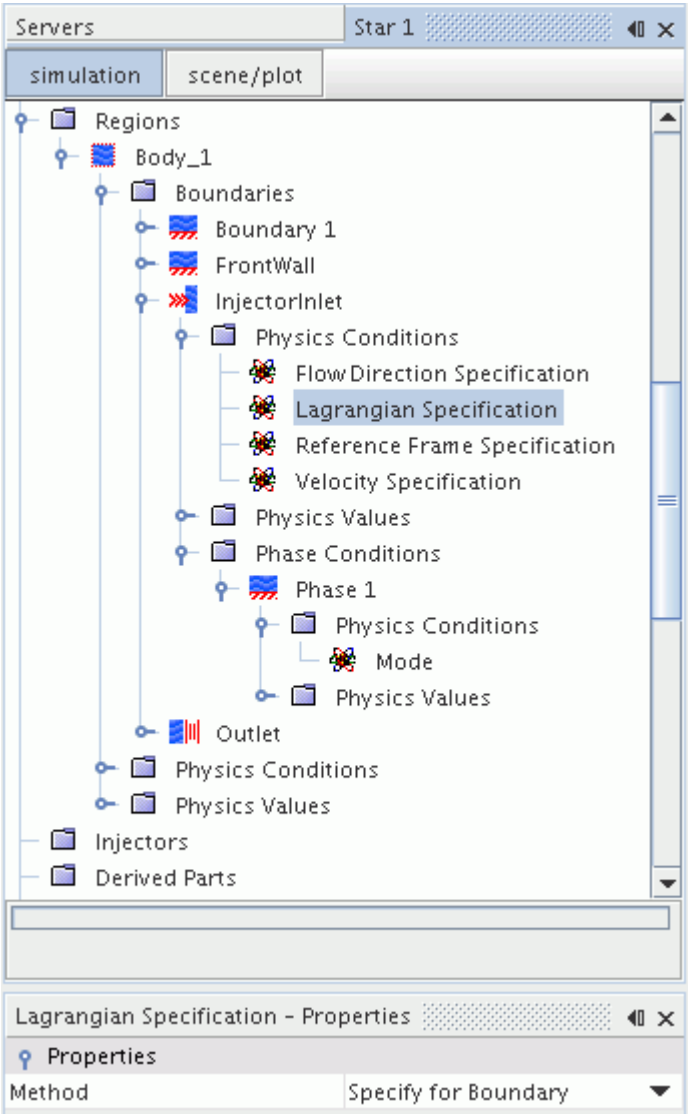


The boundary node within the **Phase Conditions** manager changes to a wall icon. The wall icon indicates that Lagrangian particles see the boundary as a wall.

To set specific wall interaction options on the impermeable boundary:

3. Select the **Physics Conditions** > **Lagrangian Specification** node of the parent boundary and set *Method* to **Specify for Boundary**.

   The **Physics Conditions** and **Physics Values** manager nodes appear beneath the impermeable boundary node for the phase.



Understanding Boundary Interaction Modes

The boundary interaction mode defines how particles behave when they impinge on a boundary.

The boundary interaction mode controls what conditions and values, if any, must be set to determine the behavior.

The availability of a boundary interaction mode for a Lagrangian phase at a boundary depends on both the boundary type and the Lagrangian phase models, as summarized in the table below.

The Phase Impermeable type only appears if you have selected the Phase Impermeable option for a Lagrangian Phase within the Phase Conditions node of a flow boundary object.

Boundary/Interface Type	Massless Particle Interaction Modes	Material Particle Interaction Modes	DEM Particle Interaction Modes

Axis	Escape	Reflect	N/A
Baffle	Composite Escape* Stick	Bai-Gosman Bai-ONERA Composite Escape Rebound* Satoh Stick Vaporize	Escape DEM*
Contact	Composite Escape* Stick	Bai-Gosman Bai-ONERA Composite Escape Ice Accretion Rebound* Satoh Stick Vaporize	Escape DEM*
Fluid Film	Composite Escape* Stick	Bai-Gosman Bai-ONERA Composite Droplet Film Transition Escape Ice Accretion Film Damping Fluid Film Rebound* Satoh Stick Vaporize	Escape DEM*
Flow-Split Outlet	Escape	Escape	Escape
Free-Stream	Escape	Escape	Escape
Fully Developed	Escape	Escape	Escape
Internal	Transmit	Transmit	Transmit
Mass Flow Inlet	Escape	Escape	Escape

Phase Impermeable	Composite Escape* Stick	Bai-Gosman Bai-ONERA Composite Escape Rebound* Satoh Stick Vaporize	Escape DEM*
Porous Baffle	Composite Escape Stick Transmit*	Composite Escape Rebound Stick Transmit*	Transmit DEM*
Pressure Outlet	Escape	Escape	Escape
Stagnation Inlet	Escape	Escape	Escape
Symmetry Plane	Escape	Reflect	N/A
Velocity Inlet	Escape	Escape	Escape
Wall	Composite Escape* Stick	Bai-Gosman Bai-ONERA Composite Escape Ice Accretion Rebound* Satoh Stick Vaporize VOF Conversion VOF-Lagrangian Impingement VOF-Lagrangian Phase Interaction	Escape DEM*

* default

Bai-Gosman

Bai-Gosman mode is available only with the Bai-Gosman Impingement model at walls, baffles, fluid film, phase impermeable, and contact interfaces. A procedure is adopted to determine some of the droplet post-impingement quantities depending on wall temperature, surface roughness and whether it is wet or dry.

Bai-ONERA

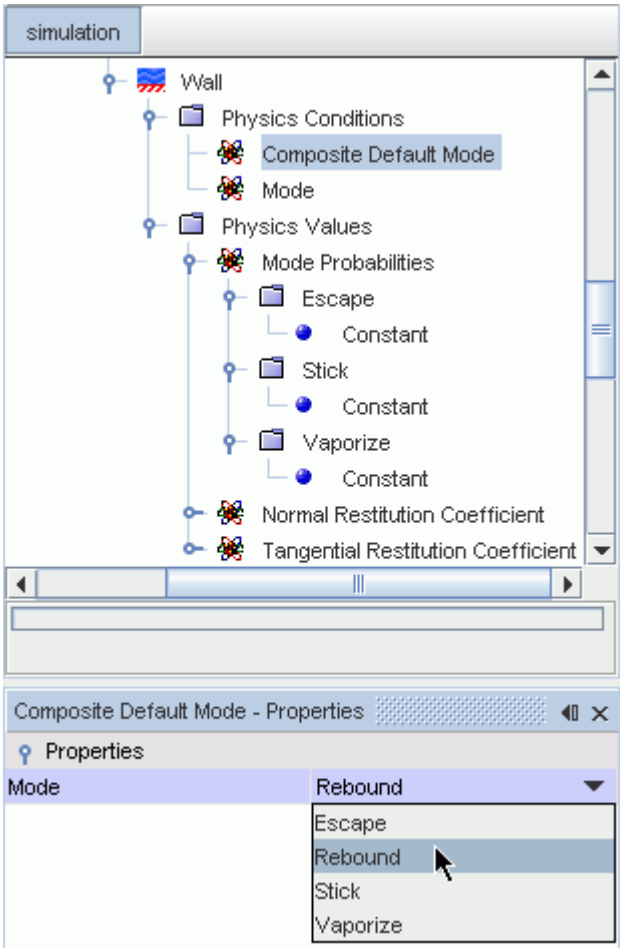
Bai-ONERA mode is available only with the Bai-ONERA Impingement model at walls, baffles, fluid film, phase impermeable, and contact interfaces. Like the Bai-Gosman mode, it determines some droplet post-impingement quantities, but also determines deposition and fluid film formation.

Composite

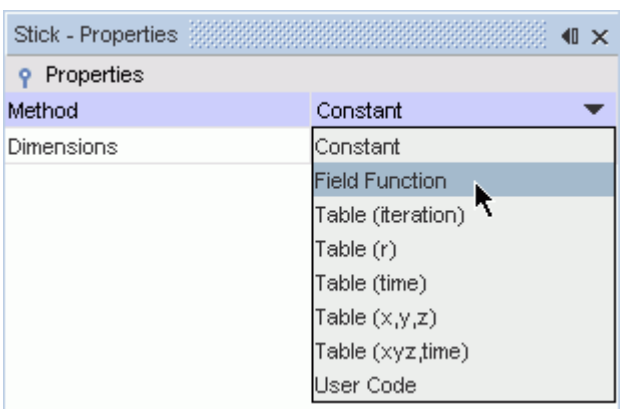
Composite mode is available at baffle, contact, fluid film, phase impermeable, porous baffle, and wall boundaries. It is available with massless and material particles but not with DEM. The aim of the mode is to allow the actual boundary interaction mode for each parcel to be selected probabilistically, based on local conditions.

Selecting **Composite** mode adds a physics condition **Composite Default Mode** that allows you to select the main boundary interaction mode.

It also adds a manager of **Mode Probabilities** under **Physics Values**. Available boundary interaction modes are listed under this manager and the probability of its occurrence can be specified for each.



By default, the probability of each mode under **Mode Probabilities** is set to be constant with a value of 0.0. Although you can use the usual options for a scalar profile to set the values, remember that the total sum of probability must be less than or equal to one. The probability of a certain event is 1, and the probability of an impossible event is 0. Choosing the Field Function method allows the probability to be expressed in terms of local conditions, such as impingement velocity and wall temperature.



The probability of the default mode is automatically calculated as one minus the sum of the probabilities of all other modes. For this reason, the mode with the highest chance of occurrence is usually the best choice to set as default.

DEM

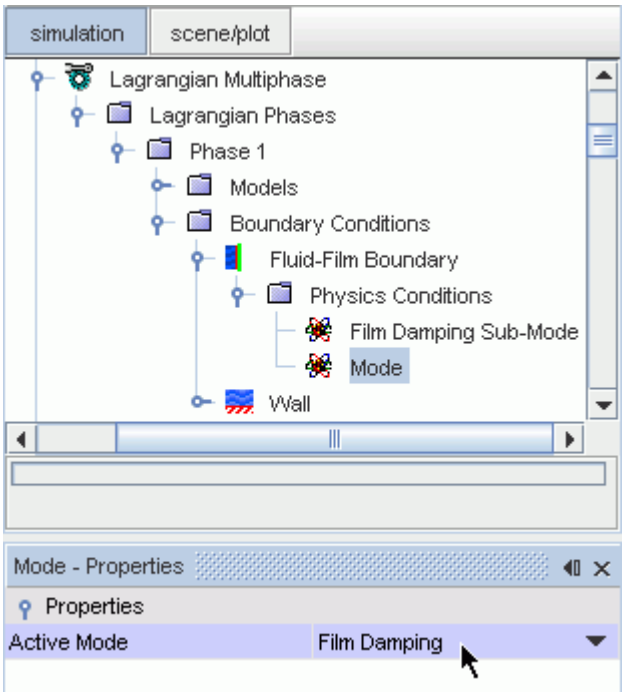
DEM mode is only available with the Discrete Element Method (DEM). This mode makes the material properties of the DEM particles available at the boundary.

DEM Trap

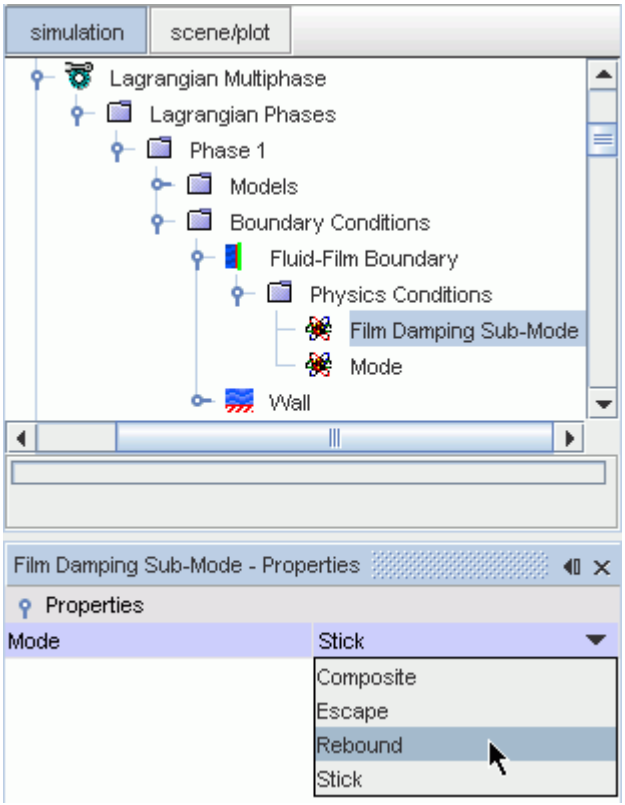
DEM mode is only available with the Discrete Element Method (DEM) when the DEM Trap Walls model is selected. This mode allows you to specify the rate at which DEM particles leave a boundary.

Film Damping

Film Damping mode allows the motion of solid particles to be damped immediately before and after wall impingement as they pass through a fluid film. This mode is only available for solid particles on fluid film boundaries.



Selecting **Film Damping** mode adds a physics condition **Film Damping Sub-Mode**. The behavior of the particle as it impinges on the wall having passed through the film can be specified as stick, rebound, escape, or composite.



Fluid Film

In this mode, droplets which impinge on the Fluid-Film interface are converted to fluid film: their mass, momentum, and energy are added to the conservation equations maintained by the Fluid Film model, ensuring global conservation.

Ice Accretion

Ice Accretion mode is available only with the Ice Accretion model at walls and contact interfaces. It means that the droplet is added to the ice layer instantaneously upon contact with the boundary.

Escape

Escape mode means that the particle is removed from the simulation, effectively escaping the solution domain. The mass, momentum, and energy (where applicable) of the parcels are also removed.

Rebound

Rebounding particles remain active in the simulation, but with their velocity modified according to restitution relationships. For walls, baffles and contact interfaces, this mode is the default. When a two-way coupled particle rebounds off a Fluid Film, some of the energy of the particle can be transferred to the film.

Reflect

Reflect mode means that the particle rebounds perfectly from the boundary, effectively equivalent to rebound with unit restitution coefficients.

Satoh

Satoh mode is available only with the Satoh Wall Impingement model at walls, baffles, fluid film, phase impermeable, and contact interfaces. A portion of particles encountering a boundary rebounds, while the residue either escapes, sticks, or forms a fluid film. The response depends on the particle normal velocity, diameter, and the boundary properties. Satoh mode can be used to model the impingement of oil droplets in oil-mist separators.

Stick

Stick mode is available only for unsteady flows. Stuck particles remain stuck to the same boundary face with the same coordinates until either they evaporate or the face is no longer a wall. In the latter case, they become active (free) particles again. Stuck particles get the velocity of the boundary face to which they are attached, but note that this velocity is ignored for tracking purposes: if the wall has a net velocity relative to the mesh, the parcel is not tracked along the wall. No net force is allowed to act on the particle, and there is no momentum transfer. Heat and mass transfer continue as for free particles. In the current version of STAR-CCM+, there is no heat transfer directly to the wall using this mode. When a two-way coupled particle sticks to a Fluid Film, the kinetic energy of the particle is transferred to the film.

Transmit

Transmit mode means that the particle is allowed to cross the interface, with its position and velocity transformed in a way consistent with the interface topology.

Vaporize

Vaporize mode is available only with the Droplet Evaporation model at walls, baffles, and contact interfaces. It means that the droplet evaporates instantaneously upon contact with the boundary. If the Two-Way Coupling model is active, the mass, momentum, and energy of the droplet is transferred to the continuous phase; otherwise this mode is functionally identical to Escape mode.

VOF Conversion

VOF Conversion mode is available on walls only when the VOF-Lagrangian Impingement model is active. When Lagrangian droplets impinge on a dry wall, the mass, tangential momentum, and energy are transferred to a VOF phase in cells on the wall boundary. Once there is sufficient volume fraction of the liquid VOF phase in a cell, VOF-Lagrangian phase interactions come into effect and the droplets are absorbed in the cell before they interact with the boundary.

Modeling Bai-Gosman Wall Impingement

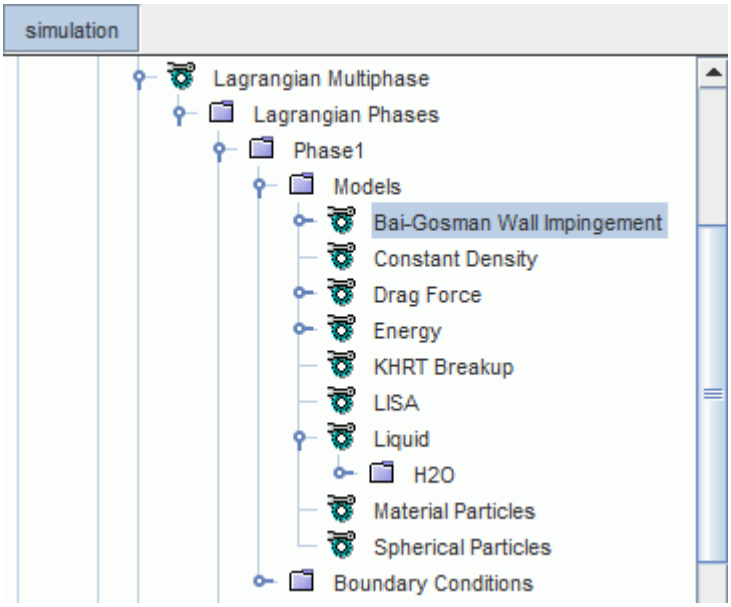
The Bai-Gosman wall impingement model is based on literature findings and mass, momentum, and energy conservation constraints.

To reflect the stochastic nature of the impingement process, the model uses a random procedure to determine some of the droplet post-impingement quantities. This randomization allows secondary droplets resulting from a primary droplet splash to have a distribution of sizes and velocities.

This section includes instructions on selecting this model.

The Bai-Gosman Wall Impingement Model

The Bai-Gosman wall impingement model ([571], [572], [573]) provides a methodology for modeling the behavior of droplets impacting on a wall. In particular, this model attempts to predict how and when droplets break up or stick to the wall. This model is used with impermeable boundaries (wall, contact, and baffle) as well as with fluid film. The **Bai-Gosman Wall Impingement** node represents this model in the Lagrangian phase model tree.



Bai-Gosman Wall Impingement Properties

Child Parcels	Number of child parcels to create during a splash/breakup event.	
WeT1	Minimum Weber number for breakup We_{T1} in temperature ranges 2 (Eqn. (3148) and Eqn. (3149)) and 3 (Eqn. (3153) and Eqn. (3154)).	
WeT2	Minimum Weber number for breakup and spread We_{T2} in temperature range 2, see Eqn. (3149) and Eqn. (3150).	
Wall State	This property defines whether an impermeable boundary (other than a Fluid Film boundary) is treated as wet or dry. In the case of a Fluid Film boundary, the wall is treated as wet when the film thickness is larger than zero. See Bai-Gosman Wall Impingement.	
	Wet	Specifies that the wall is wet; droplet impingement produces different outcomes, depending on Weber number and wall temperature: • At low Weber numbers, droplets adhere, rebound, or spread. • At high Weber numbers, droplets splash, breakup and spread, or break up and rebound.

	Dry	Specifies that the wall is dry; droplet impingement produces different outcomes, depending on Weber number and wall temperature: • At low Weber numbers, droplets spread or rebound. • At high Weber numbers, droplets splash, breakup and spread, or break up and rebound.
		Leidenfrost effect occurs when the temperature is high enough.

Lower and Upper Transition Temperature Properties

The **Lower** and **Upper Transition Temperature** nodes are subnodes of the **Bai-Gosman Wall Impingement** node.

Lower Transition Temperature	Temperature for transition between ranges 1 and 2 (T_{12} in Eqn. (3130) and Eqn. (3147)). This value is expected to be approximately the boiling temperature of the droplet.
Upper Transition Temperature	Temperature for transition between ranges 2 and 3 (T_{23} in Eqn. (3147) and Eqn. (3152)). This value is expected to be approximately the Leidenfrost temperature of the droplet.

Bai-Gosman Wall Impingement Expert Properties

Rosin-Rammler Exponent	Exponent in Rosin-Rammler size distribution, q in Eqn. (3145).
Cf	Wall friction coefficient for splash-generated droplets, see Eqn. (3146).
Aw	Coefficient governing the onset of the splash regime, see Eqn. (3136).
a0	Coefficient governing the number of splash-generated droplets, see Eqn. (3144).
Minimum Ejection Angle	Minimum ejection angle from the wall of splash-generated droplets.
Maximum Ejection Angle	Maximum ejection angle from the wall of splash-generated droplets.
WeRebound	Minimum Weber number for rebound in range 1. The default value is 2.
WeSpread	Minimum Weber number for spread in range 1. The default value is 20.

Selecting the Bai-Gosman Wall Impingement Model

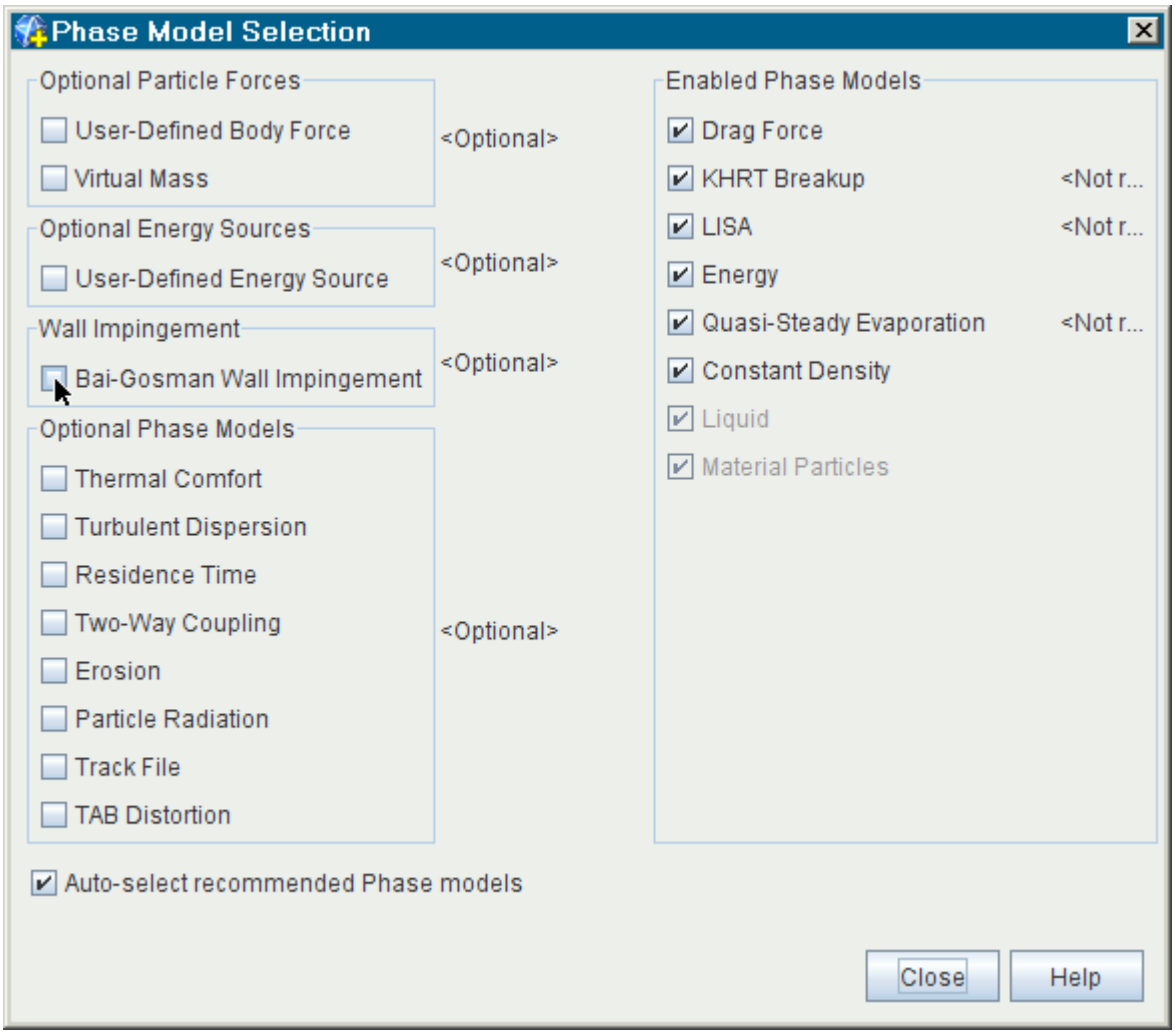
The Bai-Gosman wall impingement model is only used for liquid droplets (single- or multi-component). It can be used with a single-phase gas model or a VOF model in the physics continuum.

When the Bai-Gosman wall impingement model is used in a VOF region, it is possible that the droplet impingement occurs where the droplet is passing through a liquid. As the Bai-Gosman wall impingement model is strictly valid only for droplets in a gas, the impingement is treated as follows:

- When impingement in a liquid occurs, the assumed behavior is for the droplet to rebound (using the standard Bai rebound mode).
- When impingement in a gas occurs, the droplet behavior is identical to the single-phase gas case.

To determine whether the droplet is in a gas or liquid, STAR-CCM+ evaluates the total volume fraction of all of the gas phases in the local cell. When this value is above 0.5, the droplet is determined to be in a gas.

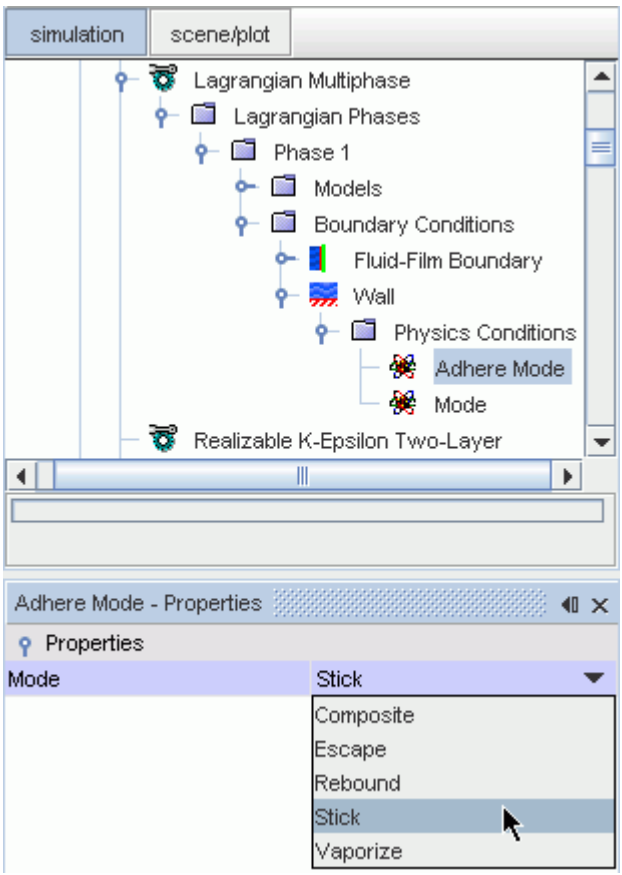
To activate the model, select Bai-Gosman Wall Impingement from the *Wall Impingement* group box of the *Phase Model Selection* dialog.



Choosing this option makes the Bai-Gosman boundary interaction mode available in the *Mode* property list for walls, baffles, contact interfaces, and fluid films. The additional **Adhere Mode** node appears for walls, baffles, and contact interfaces only.

Selecting the Bai-Gosman Adhere Mode

When the Bai-Gosman mode is activated (either directly through the **Mode** node or when **Composite** mode is selected), an additional node **Adhere Mode** appears in the **Physics Conditions**. The **Adhere Mode** allows you to select which specific boundary interaction mode is used when the impingement satisfies the Adhere conditions. The default mode for this condition is Stick.



See Also:

Using the Leidenfrost Temperature

Modeling Bai-ONERA Wall Impingement

The Bai-ONERA Wall Impingement model is a further development of the Bai-Gosman model, adding smoother transitions between impingement outcomes across transition temperatures.

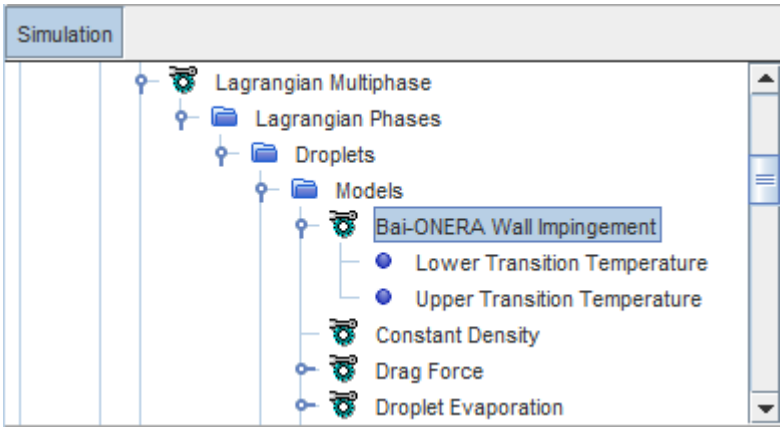
The original ONERA model, described in [602], was developed for a smooth and dry wall. Modifications made to it in the Bai-ONERA model are designed to:

- Allow for the effect of a rough or wet wall.
- Achieve a smooth transition to recover the Bai-Gosman model for situations where the wall temperature is lower than the droplet saturation temperature.

The Bai-ONERA Wall Impingement Model

Unlike the Bai-Gosman model, this model allows for droplet deposition and spreading even when the wall temperature is between the saturation temperature and the Leidenfrost temperature. Use the Bai-ONERA model when you are concerned with liquid accumulating on walls in high wall temperature applications.

Possible outcomes for a Bai-ONERA impingement are rebounding, spreading, and splashing. Compare to the Bai-Gosman model, where impingement can also result in droplet adhesion or break-up. Decide which outcomes you want to consider when choosing between these two models. The Bai-ONERA is compatible with impermeable boundaries (wall, contact, and baffle) and with fluid film. The **Bai-ONERA Wall Impingement** node represents this model in the Lagrangian phase model tree.



Bai-ONERA Wall Impingement Properties

Child Parcels	Number of child parcels to create during a splash/breakup event.	
Wall State	This property defines whether an impermeable boundary (other than a Fluid Film boundary) is treated as wet or dry. For a Fluid Film boundary, the wall is treated as wet when the film thickness is larger than zero. • At low K numbers, droplets spread or rebound. • At high K numbers, droplets splash, spread, or rebound. Leidenfrost effect occurs when the temperature is high enough. See Bai-ONERA Wall Impingement.	
	Wet	Specifies that the wall is wet; droplet impingement produces different outcomes, depending on K number (see Eqn. (3156)) and wall temperature. The transition point between rebound and spreading depends on the Laplace number of the droplet. See Wet Wall in Bai-ONERA Wall Impingement.
	Dry	Specifies that the wall is dry; droplet impingement produces different outcomes, depending on K number (see Eqn. (3156)) and wall temperature. The transition point between splashing and spreading depends on wall roughness. See Dry Wall in Bai-ONERA Wall Impingement.

Lower and Upper Transition Temperature Properties

The **Lower** and **Upper Transition Temperature** nodes are subnodes of the **Bai-ONERA Wall Impingement** node.

Lower Transition Temperature	Temperature for transition between ranges 1 and 2 (T_{12} in Bai-ONERA Wall Impingement). This value is expected to be approximately the boiling temperature of the droplet.
Upper Transition Temperature	Temperature for transition between ranges 2 and 3 (T_{23} in Bai-ONERA Wall Impingement). This value is expected to be approximately the Leidenfrost temperature of the droplet.

Bai-ONERA Wall Impingement Expert Properties

The Rosin-Rammler exponent q, the wall friction coefficient c_f, the onset coefficient A_w, and the splash number coefficient a_0 are used just as they are in the Bai-Gosman treatment.

Rosin-Rammler Exponent	Exponent in Rosin-Rammler size distribution, q in Eqn. (3145).
Cf	Wall friction coefficient for splash-generated droplets, see Eqn. (3146).
Aw	Coefficient governing the onset of the splash regime, see Eqn. (3136).
a0	Coefficient governing the number of splash-generated droplets, see Eqn. (3144).
WeRebound	Minimum Weber number for rebound in range 1. The default value is 2.
WeSpread	Minimum Weber number for spread in range 1. The default value is 20.
Minimum Ejection Angle	Minimum ejection angle from the wall of splash-generated droplets.
Maximum Ejection Angle	Maximum ejection angle from the wall of splash-generated droplets.
K0	(See Regime Transition Criteria in Bai-ONERA Wall Impingement.)
K1	(See Regime Transition Criteria in Bai-ONERA Wall Impingement.)
Gamma	The exponent γ in Eqn. (3158), describing the transition between spreading and splashing in the temperature range between the saturation point and the Leidenfrost point.
Smrf	Splash Mass Ratio Exponent, the exponent n in Eqn. (3160), describing the splash ratio between the saturation point and the Leidenfrost point.

Selecting the Bai-ONERA Wall Impingement Model

The Bai-ONERA wall impingement model is only used for liquid droplets (single- or multi-component). It can be used with a single-phase gas model or a VOF model in the physics continuum.

When the Bai-ONERA wall impingement model is used in a VOF region, it is possible that the droplet impingement occurs where the droplet is passing through a liquid. As the Bai-ONERA wall impingement model is strictly valid only for droplets in a gas, the impingement is treated as follows:

- When impingement in a liquid occurs, the assumed behavior is for the droplet to rebound (using the standard Bai rebound mode).
- When impingement in a gas occurs, the droplet behavior is identical to the single-phase gas case.

To determine whether the droplet is in a gas or liquid, STAR-CCM+ evaluates the total volume fraction of all of the gas phases in the local cell. When this value is above 0.5, the droplet is determined to be in a gas.

To activate the model, select Bai-ONERA Wall Impingement from the *Wall Impingement* group box of the *Droplets Model Selection* dialog.

Choosing this option makes the Bai-ONERA boundary interaction mode available in the *Mode* property list for walls, baffles, contact interfaces, and fluid films.

See Also:

Using the Leidenfrost Temperature

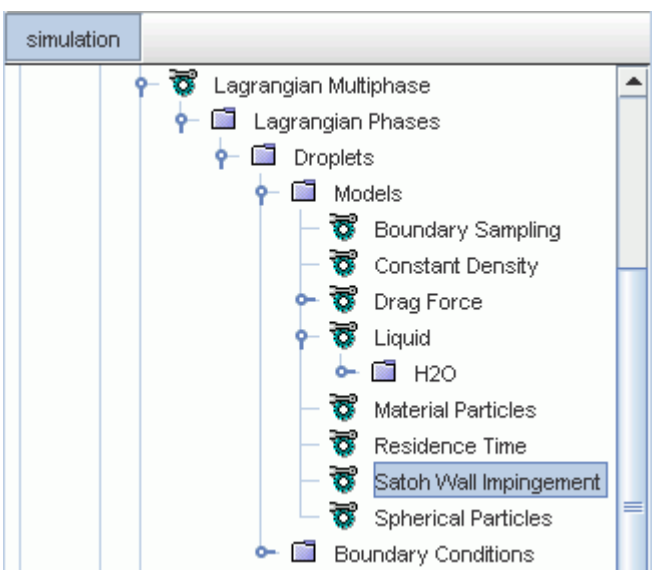
Modeling Satoh Wall Impingement

The Satoh Wall Impingement Model is a wall impingement feature for modeling oil droplets in oil-mist separators.

This section includes instructions on selecting this model.

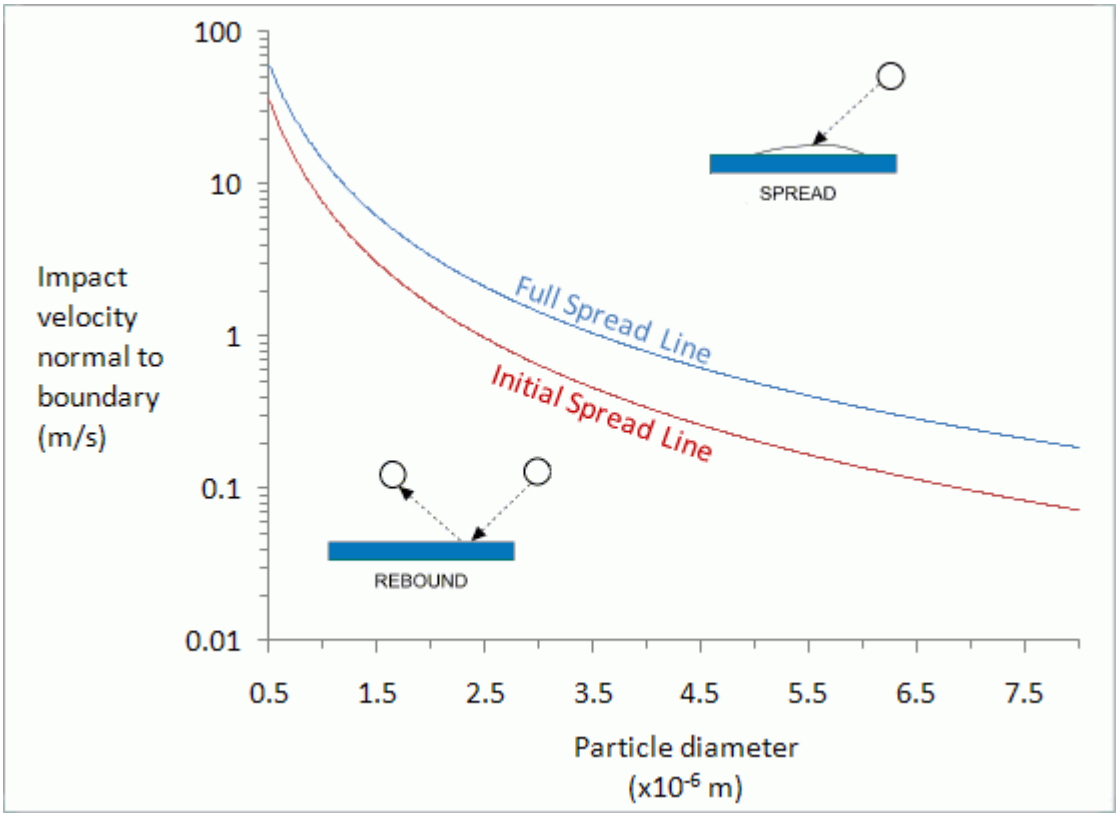
The Satoh Wall Impingement Model

The **Satoh** model node represents the Satoh Wall Impingement model. It provides a method for modeling the behavior of oil droplets in a “blow-by” flow as they encounter the walls, baffles, and porous plates of an oil-mist separator.



According to Satoh and others [603], the particle diameter and its velocity normal to the boundary determine the outcome of a collision between a particle and a boundary. The relationship between the two is assumed to follow two power law curves as depicted below. Below the lower curve, called the Initial Spread Line, all the

particles rebound. Above the upper curve, called the Full Spread Line, the particles spread over the boundary. In the area between the curves some particles rebound and the rest spread.



As per Satoh and others [603], the droplet diameter is in the range 0 m to 8E-6 m. The droplet impingement velocity normal to the wall is in the range 0.01 m/s to 13 m/s. In this range, small droplets with low normal velocity rebound from the wall whereas large droplets with high normal velocity tend to spread over the surface to form a film.

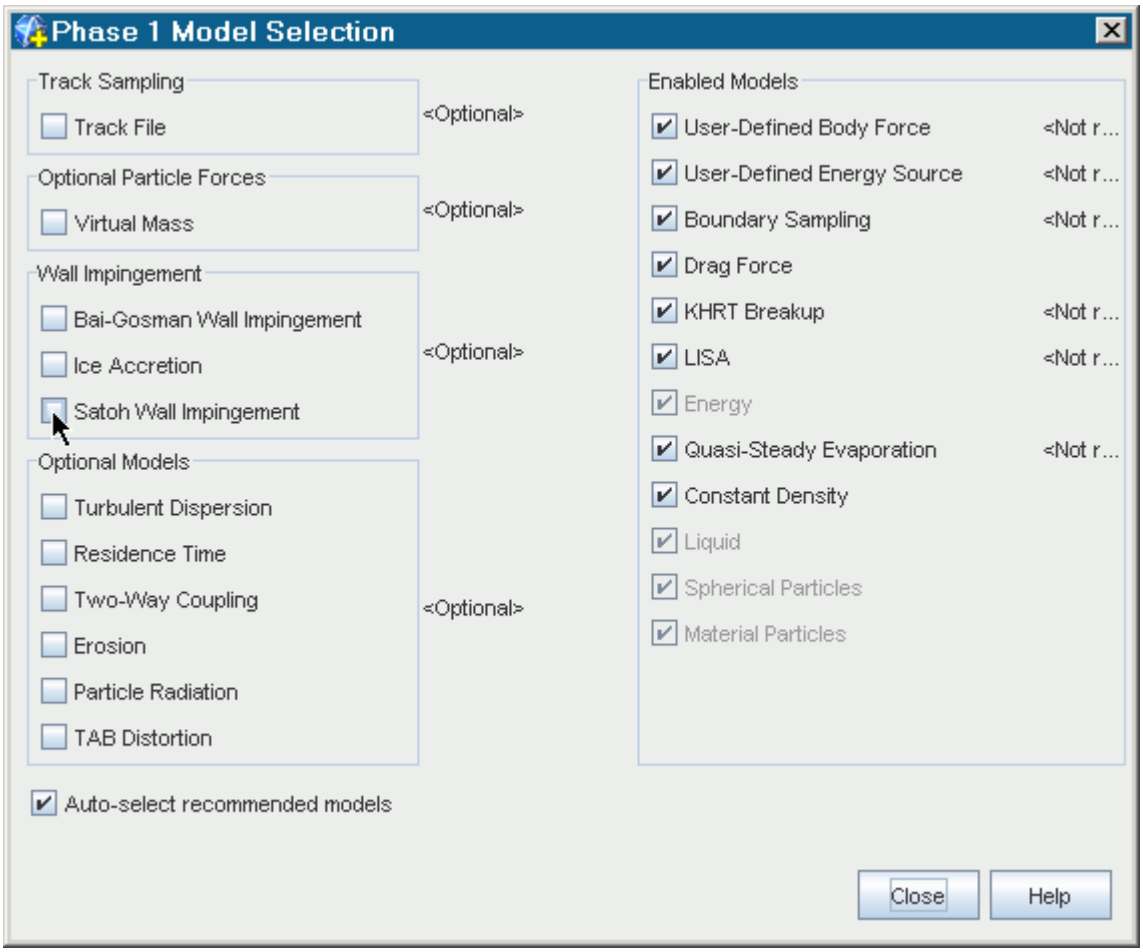
In STAR-CCM+ fluid film formation is dependent on both model selection and the boundary type. An order of precedence is used to model spreading behavior: if the Fluid Film model is not available, Stick mode is used. If neither mode is applicable on the incident boundary, the particle is allowed to Escape.

Satoh Wall Impingement Properties

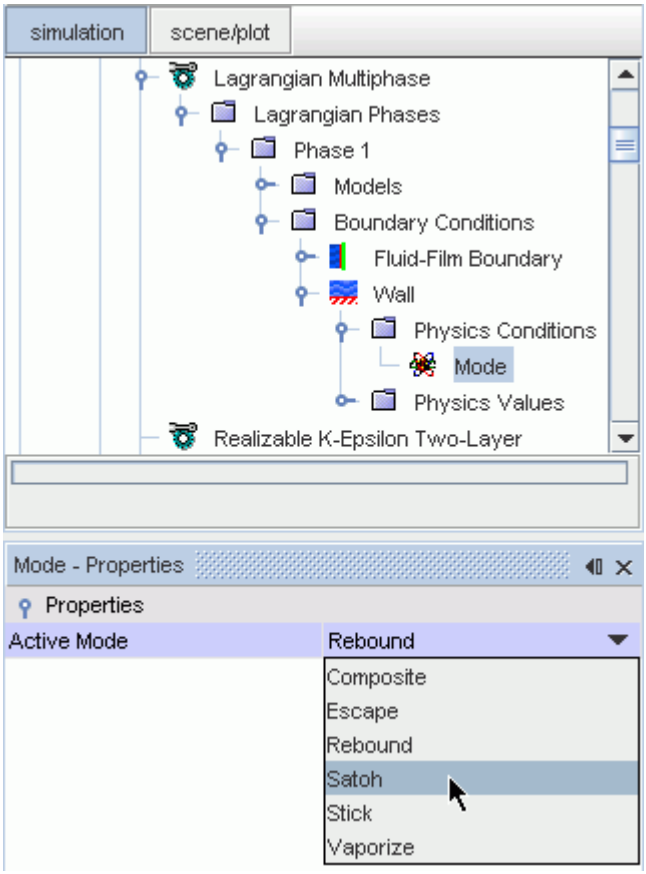
Reference Diameter	Reference particle diameter, D_{ref} , used to calculate the non-dimensional diameter for the initial Eqn. (3162) and full spread lines Eqn. (3163).
Initial Pre-exponential	Scalar multiplier, a_{ISL} , for initial spread line.
Initial Exponent	Exponent, n_{ISL} , for the dimensionless diameter for the initial spread line.
Full Pre-exponential	Scalar multiplier, a_{FSL} , for full spread line.
Full Exponent	Exponent, n_{FSL} , for the dimensionless diameter for the full spread line.

Selecting the Satoh Wall Impingement Model

The Satoh wall impingement model is only used for liquid droplets (single- or multi-component). It also requires a Gas model in the physics continuum. The Satoh wall impingement model is not compatible with the Eulerian Multiphase model. To activate the model, select Satoh Wall Impingement from the *Wall Impingement* group box of the *Phase Model Selection* dialog.



Choosing this option makes the boundary interaction mode available in the *Mode* property list for walls, baffles, and contact interfaces.



Droplet Film Transition Model Reference

The Droplet Film Transition model controls the details of the process by which droplets impinging on a surface form a liquid film.

Theory	See Droplet-to-Film Transition Criteria.
Provided By	**[Lagrangian phase]** > **Models** > *Optional Models*
Example Node Path	**[Lagrangian phase]** > **Models** > **Droplet Film Transition**
Requires	Phase models: *Particle Type*: **Material Particles** *Material*: **Liquid** *Optional Models*: **Two-Way Coupling**
Properties	*Droplet Transition Option* Determines the criterion by which droplets landing on a cell wall makes the transition from individual droplets to a liquid film on the wall. See the Droplet-to-Film Transition Criteria formulation.

	• **Coverage Ratio**: The transition happens when the droplets cover a specified fraction of the cell surface. The default fraction is `0.85`. • **Equivalent Thickness**: The transition happens when the total volume of liquid on the surface can form a film of a specified thickness. The default thickness is `5E-5` meters.	
Activates	Materials	**Surface Tension** for **Liquid** under the Lagrangian phase. The value is set as a scalar.
	Field Functions	**Film Transition Mark**. See Lagrangian Multiphase Field Functions Reference.

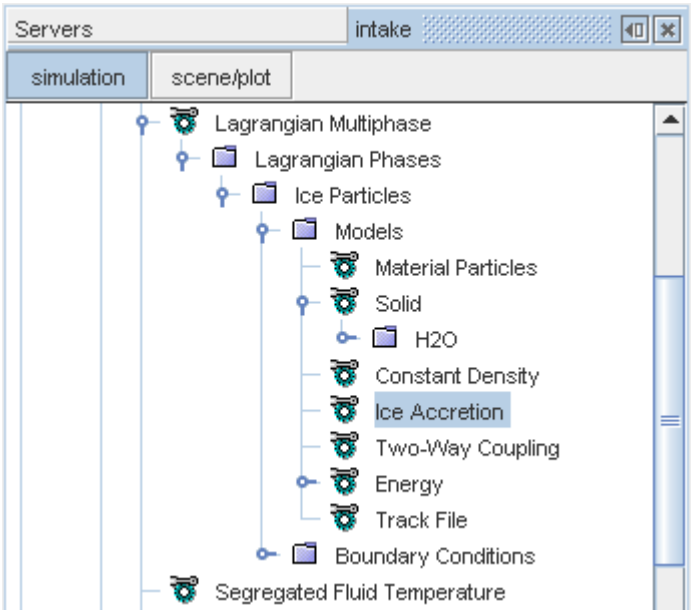
Modeling Ice Accretion

This section describes the Ice Accretion model and how to select it.

The Ice Accretion Model

The **Ice Accretion** model node represents the Ice Accretion model. When this model is active, particles impinging on a boundary ice layer are accounted for in the mass and energy balance equations from the Deicing model.

This model governs the transfer of particle mass to an ice layer on impact with an ice-layer boundary, to become part of the overall mass balance of the ice layer.



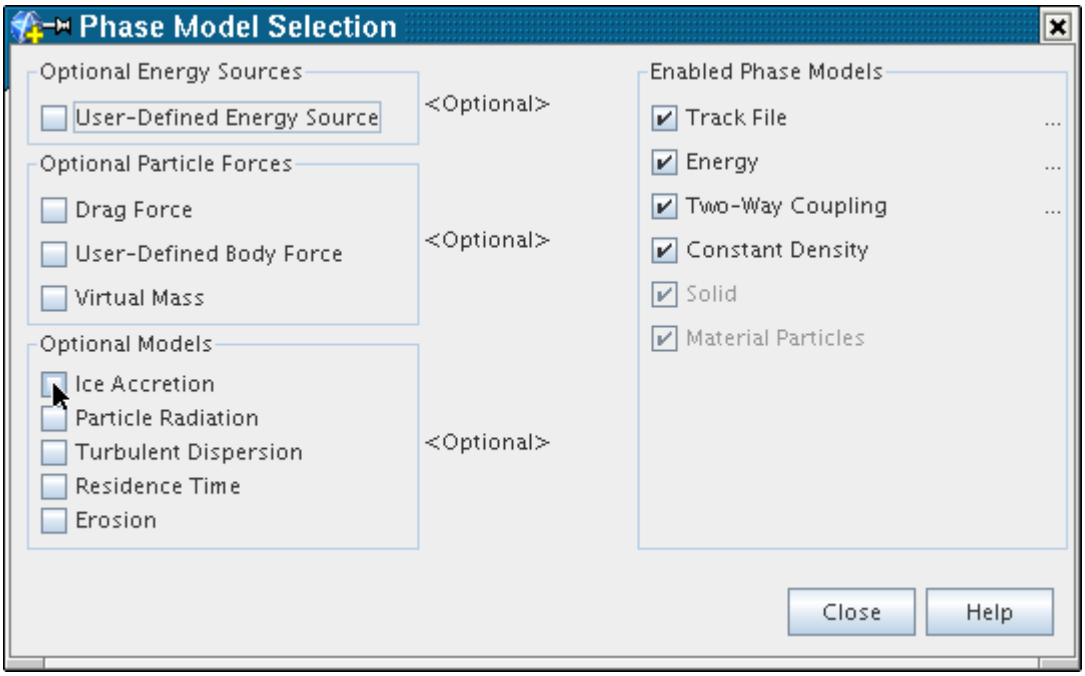
The Ice Accretion model has no properties.

Selecting the Ice Accretion Model

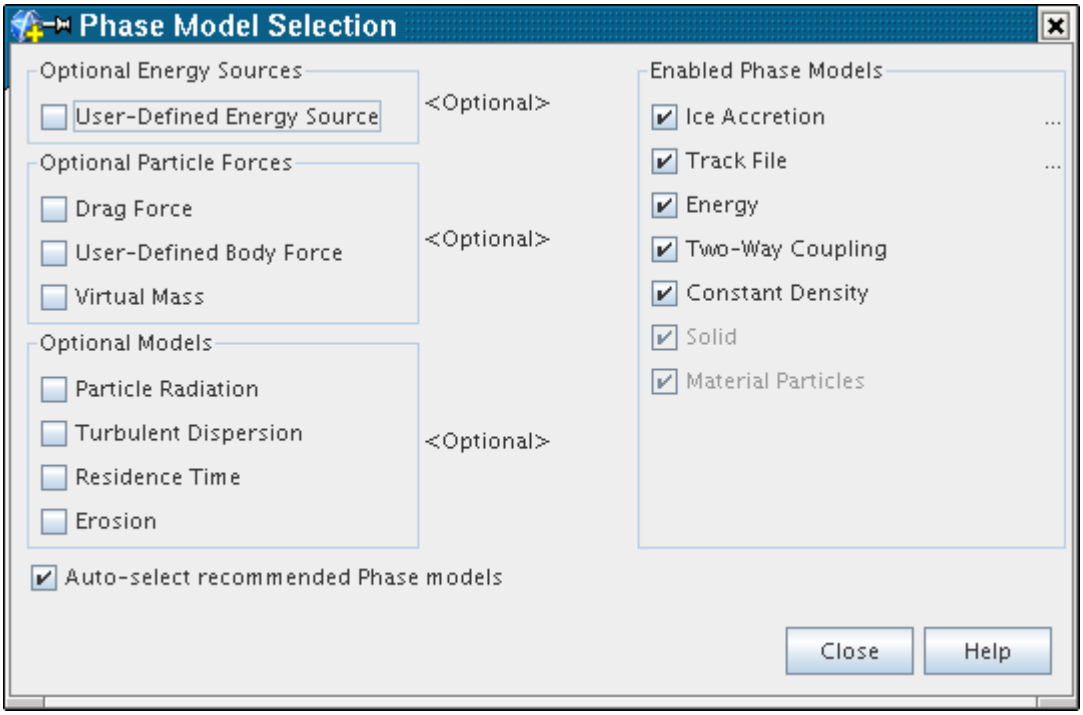
The Ice Accretion model becomes available only if you have the Deicing model selected in the physics continuum and these models selected under the **Lagrangian Phases** node:

- Lagrangian Energy
- Two-Way Coupling
- The single-component droplet model or the single-component particle model

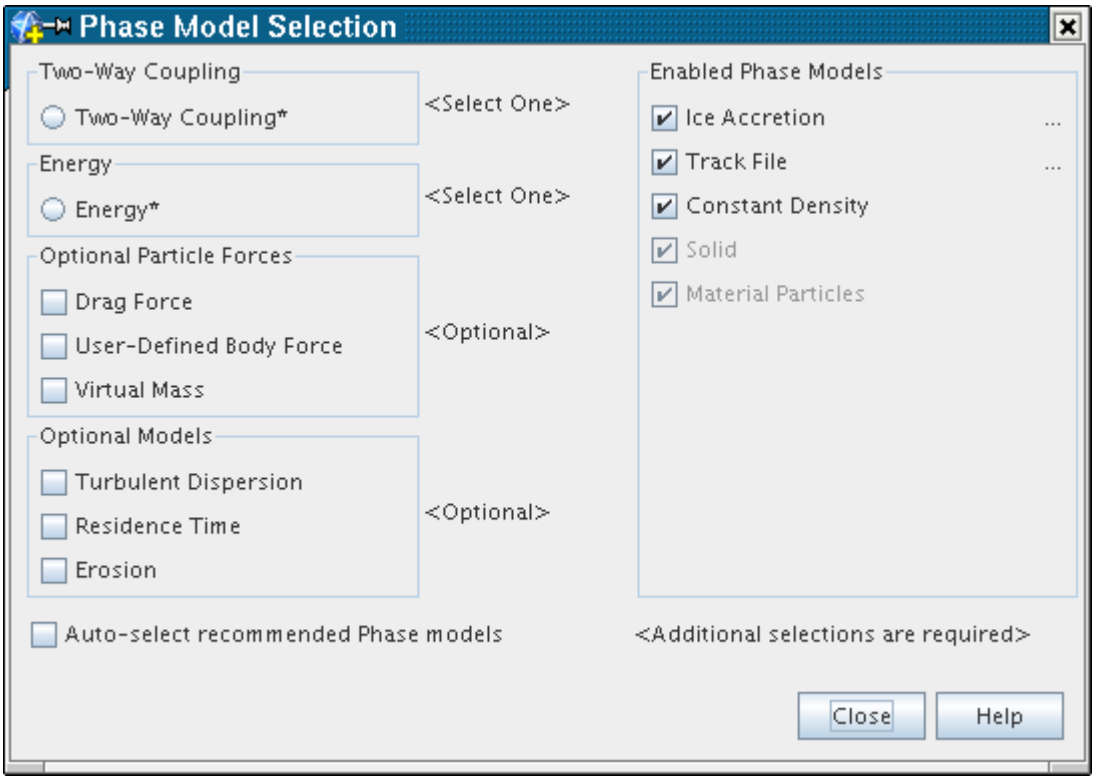
Select the **Ice Accretion** option from the *Optional Models* group as shown below.



The dialog shows the **Ice Accretion** model activated.



If the **Auto-select recommended Phase models** box has been checked, then selecting the **Ice Accretion** model causes the **Energy** and **Two-Way Coupling** models to be selected as well. If the box is not checked, and the models are not already selected, a warning message appears beneath the *Enabled Phase Models* group box to indicate that more selections are required.



Boundary Values Reference

This section describes the boundary values that you can set for a Lagrangian phase.

Depending on how the Lagrangian phase boundary conditions and models are set, you can control the interaction with boundaries using the following values:

- Normal Restitution Coefficient
- Tangential Restitution Coefficient
- Erosion Ratio

Normal Restitution Coefficient

Added to the tree when **Rebound** is chosen as the interaction mode at a boundary. This coefficient is the constant of proportionality e_n for the normal components of rebounding and impinging velocities. See Rebound.

Tangential Restitution Coefficient

Added to the tree when **Rebound** is chosen as the interaction mode at a boundary. This coefficient is the constant of proportionality e_t for the tangential components of rebounding and impinging velocities. See Rebound.

Erosion Ratio

Added to the tree when the Erosion model is activated in the Lagrangian phase. Depending on the method that is chosen for the erosion ratio, a sub-node is added to this node.

Erosion Ratio Properties

Method

Selects the method for calculating the erosion ratio.

Ahlert

Specifies that the erosion ratio is calculated using the Ahlert Correlation. An Ahlert node is added as a child to this node.

Field Function

Specifies that the erosion ratio is calculated using a field function. This method is typically user-defined and can involve only constants and particle-based functions. A field function method node is added as a child to this node.

DNV

Specifies that the erosion ratio is calculated using the DNV Correlation. A DNV node is added as a child to this node.

Neilson-Gilchrist

Specifies that the erosion ratio is calculated using the Neilson-Gilchrist Correlation. A Neilson-Gilchrist node is added as a child to this node.

Oka

Specifies that the erosion ratio is calculated using the Oka Correlation. An Oka node is added as a child to this node.

Dimensions

The dimensionality of the erosion ratio (read-only).

Ahlert

This sub-node is added to the Erosion Ratio node when **Ahlert** is chosen as the erosion ratio method. It represents the Ahlert Correlation. The default values for this method are for liquid-borne semi-rounded sand particles eroding aluminum, as given by McLaury *et al.* [586].

Ahlert Properties

Material Coefficient

A material-dependent constant. It is the quantity K in Eqn. (3247).

Shape Coefficient

A factor to account for the shape of the particles. It is the quantity F_s in Eqn. (3247). Suggested values are 1 for angular particles, 0.53 for semi-rounded particles and 0.2 for fully rounded particles.

Transition Angle

The angle α_0 which is the upper limit of validity of the Low Alpha Function and the lower limit of validity of Eqn. (3248).

Low Alpha Function

The polynomial for the angle function $f(\alpha)$ in Eqn. (3247) to be used below the transition angle α_0 . The polynomial is in terms of the incidence angle in radians.

Coefficient W, X, Y, Z

The coefficient w, x, y, or z in Eqn. (3248) for the angle function $f(\alpha)$ above the transition angle α_0. *Coefficient Z* is read-only.

Reference Velocity

The normalization factor for the particle velocity. It is the quantity u_{ref} in Eqn. (3247).

Velocity Exponent

The exponent for the normalized velocity. It is the quantity n in Eqn. (3247).

DNV

This sub-node is added to the Erosion Ratio node when **DNV** is chosen as the erosion ratio method. It represents the DNV Correlation. The default values for this method are for air-borne sand eroding carbon steel, and are taken from Haugen *et al.* [581].

DNV Properties

Material Coefficient

A material-dependent constant. It is the quantity K in Eqn. (3249). The default value is 2.0E-9.

Reference Velocity

The normalization factor for the particle velocity. It is the quantity u_{ref} in Eqn. (3249). The default value is 1.0 m/s.

Velocity Exponent

The exponent for the normalized velocity. It is the quantity n in Eqn. (3249). The default value is 2.6.

Angle Function

The polynomial for the angle function $f(\theta)$ in Eqn. (3249). It is evaluated in terms of the incidence angle in degrees. See Eqn. (3250).

Neilson-Gilchrist

This sub-node is added to the Erosion Ratio node when **Neilson-Gilchrist** is chosen as the erosion ratio method. It represents the Neilson-Gilchrist Correlation. The default values for this method are for liquid-borne sand eroding AISI 4130 steel, and are taken from Wallace *et al.* [614].

Neilson-Gilchrist Properties

Transition Angle

The angle at which cutting deformation changes behavior. It is the quantity α_0 in Eqn. (3252).

Deformation Cutoff Velocity

The normal velocity at which deformation wear is minimized. It is the quantity K in Eqn. (3253).

Cutting Coefficient

The material-dependent constant governing cutting erosion. It is the quantity ε_C in Eqn. (3252).

Deformation Coefficient

The material-dependent constant governing deformation erosion. It is the quantity ε_D in Eqn. (3253).

Oka

This sub-node is added to the Erosion Ratio node when **Oka** is chosen as the erosion ratio method. It represents the Oka Correlation. The default values for this method are for air-borne sand eroding 0.025%

carbon steel, and are taken from Oka *et al.* [592] and Oka and Yoshida [593], except for the *Reference Erosion Ratio*, which is derived from the DNV Correlation.

Oka Properties

Reference Velocity

The normalization factor for the particle velocity. It is the quantity u_{ref} in Eqn. (3254).

Velocity Exponent

The exponent for the normalized velocity. It is the quantity k_2 in Eqn. (3254).

Reference Erosion Ratio

The erosion ratio at reference conditions and normal incidence. It is the quantity e_{90} in Eqn. (3254).

Coefficient Hv

The coefficient H_v in the angle function Eqn. (3255). It is typically, but not necessarily, equal to the Vickers hardness of the eroded material in GPa.

Coefficient n1

The coefficient n_1 in the angle function Eqn. (3255).

Coefficient n2

The coefficient n_2 in the angle function Eqn. (3255).

Reference Diameter

The normalization factor for the particle diameter. It is the quantity D_{ref} in Eqn. (3254).

Diameter Exponent

The exponent for the normalized diameter. It is the quantity k_3 in Eqn. (3254).

Working with Primary Atomization

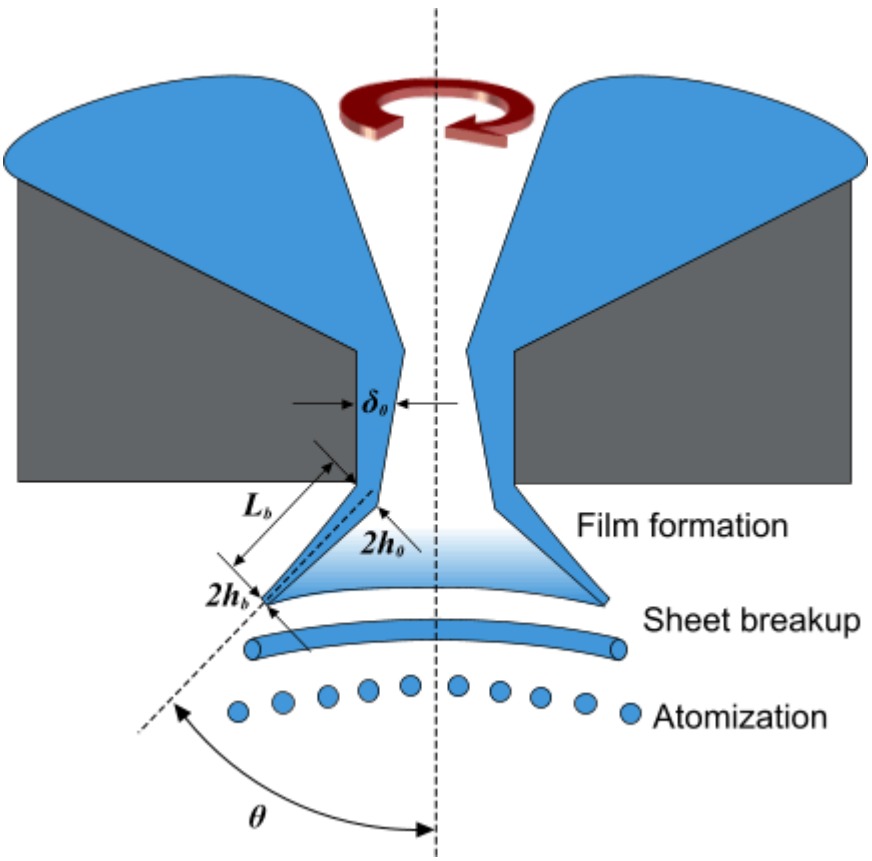
Primary atomization refers to the process of forcing liquid through a small orifice at a high pressure, resulting in a fine spray of liquid droplets.

Depending on the internal construction of the injector, the liquid at the exit of an injector nozzle can be in the form of a thin liquid sheet or a jet. STAR-CCM+ currently provides two models that simulate the atomization process of the liquid exiting the nozzle: the LISA model for the breakup of a thin liquid sheet that is produced with a pressure-swirl atomizer, and the Huh model for the breakup of a liquid jet created through a simple hole-type nozzle. Both types of injectors are grouped under the heading of "Nozzle Injector" that is defined by two basic parameter: orifice dimeter at nozzle exit and injector axis. Additional model-specific parameters are added according to the selected model.

What Is the Pressure-Swirl Atomizer?

The pressure-swirl atomizer, also commonly known as a simplex atomizer, is widely used for liquid-fuel injection in gas turbines, oil furnaces, and direct-injection spark-ignited automobile engines.

In this atomizer type, a swirling motion is imparted to the injected liquid through nozzles that are known as swirl ports. The liquid accelerates as it flows through the swirl ports into a central swirl chamber. Under the action of centrifugal force, it spreads out in the form of a conical sheet and develops a hollow air core. It emerges from the orifice as an unstable thinning liquid sheet, then breaking up into ligaments and then droplets. A sketch of a hollow-cone pressure swirl injector is shown in the figure below.



In the illustration above, three stages in the atomization process are identified: film formation, sheet breakup, and atomization. δ_o is the initial film thickness in the nozzle; $2h_0$ is the initial thickness of the film as it exits the nozzle, and $2h_b$ is the thickness of the film at the point of breakup. θ is the spray angle, which is defined as $0.5 \times (\text{inner cone angle} + \text{outer cone angle})$.

Contents:

Atomization Models

Atomization Models

The atomization models simulate the breakup of a jet or a thin liquid sheet as it emerges from a spray injector.

- The LISA Atomization model simulates a pressure-swirl atomizer.
- The Huh Atomization model simulates a simple hole injector.

The LISA Atomization Model

The Linearized Instability Sheet Atomization (LISA) model is a primary atomization model for the thin liquid sheet at nozzle exit created by the action of a pressure-swirl atomizer. This model corresponds to the LISA model node in the Lagrangian phase model tree, and selecting it activates the Nozzle Injector type for injectors. The model generates initial size and velocity values for droplets entering through the injectors active for the Lagrangian phase, as described in the formulation.

Theory	LISA Model	
Provided By	**[physics continuum]** > **Models** > **Lagrangian Multiphase** > **[phase]** > **Models** > *Primary Atomization*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Lagrangian Multiphase** > **Phase 1** > **Models** > **LISA Atomization**	
Requires	*Optional Models*: **Lagrangian Multiphase** From **Lagrangian Multiphase** > **Lagrangian Phases** > **[phase]** > **Models**: *Particle Type*: **Material Particles** *Material*: **Liquid** or **Multi-Component Liquid**	
Activates	Materials	See Materials and Methods.
	Field Functions	**Droplet Dynamic Viscosity** **Droplet Surface Tension**

The Huh Atomization Model

The Huh Atomization Model is a primary atomization model that simulates the disintegration process of a liquid jet exiting from a hole-type injector at high speed. This model estimates the initial perturbations from an analysis of the flow through the nozzle and then uses established wave growth theory, together with other hypotheses, to represent the atomization process. This model corresponds to the Huh Atomization model node in the Lagrangian phase model tree, and selecting it activates the Nozzle Injector type for injectors. The Huh model assumes an initial droplet size equal to the nozzle diameter; droplet velocity and subsequent breakup is modeled as described in the formulation.

Theory	Huh Model
Provided By	**[physics continuum]** > **Models** > **Lagrangian Multiphase** > **[phase]** > **Models** > *Primary Atomization*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Lagrangian Multiphase** > **Phase 1** > **Models** > **Huh Atomization**
Requires	*Optional Models*: **Lagrangian Multiphase** From **Lagrangian Multiphase** > **Lagrangian Phases** > **[phase]** > **Models**: *Particle Type*: **Material Particles**

	Material: **Liquid** or **Multi-Component Liquid**	
Activates	Materials	See Materials and Methods.
	Field Functions	**Droplet Dynamic Viscosity** **Droplet Surface Tension** **Droplet Weber Number** **Primary Atomization Status**
Properties	See below.	

Huh Atomization Model Properties

C1

Atomization length scale coefficient, C_1 used to define κ in Eqn. (3076). The default value is `2.0`.

C2

Wave length scale coefficient, C_2 used to define κ in Eqn. (3076). The default value is `0.5`.

C3

Spontaneous time scale coefficient, C_3 in Eqn. (3077). The default value is `1.0`.

C4

Exponential time scale coefficient, C_4 in Eqn. (3077). The default value is `1.5`.

CA1

Turbulence time scale coefficient, C_{a1} in Eqn. (3074) and Eqn. (3075). The default value is `0.92`.

CA2

Turbulence length scale coefficient, C_{a2} in Eqn. (3074). The default value is `0.4565`.

KA

Droplet breakup rate coefficient, K_A in Eqn. (3078). The default value is `0.1`.

Normal Velocity Coefficient

Controls the radial diffusion of the spray; K_3 in Eqn. (3109). The larger it is, the more rapid the radial diffusion of the spray. The default value is 0.

WeCrit

Critical Weber number, We_{cr} in Eqn. (3105). Below this number, the parcel bypasses the primary atomization process and proceeds directly to the secondary breakup process, if activated. The default value is 6.0.

Materials and Methods

The LISA and Huh Atomization models activate the following material properties under **Lagrangian Phases** > **[phase]** > **Models** > **Liquid** or **Multi-Component Liquid**:

Dynamic Viscosity

Dimensions

The dimensionality of the viscosity (read-only).

Method

Constant or **Field Function**.

Surface Tension

Dimensions

The dimensionality of the surface tension (read-only).

Method

Constant or **Field Function**.

Working with Secondary Breakup

Liquid droplets sometimes break up under the action of non-uniform surface forces, a phenomenon that is known as secondary breakup.

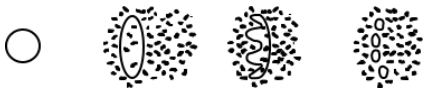
The response of droplets to non-uniform surface forces is invariably to deform, with the deformation resisted by surface tension and viscous forces inside the droplet. Characteristic measures of this behavior are the Weber and Ohnesorge numbers

$$We = \frac{\rho_g |\mathbf{v}_s|^2 D_p}{\sigma} \tag{543}$$

$$Oh = \frac{\mu_l}{\sqrt{\rho_l D_p \sigma}} \tag{544}$$

Depending on the magnitude of these forces, various breakup regimes have been identified, each characterized by the shape of the deforming droplets. An example categorization (after Stiesch [612]) for low Oh is given below.

Category		Weber Number
Vibrational breakup		~ 12
Bag breakup		< 20

Category		Weber Number
Bag/ streamer breakup		< 50
Stripping breakup		< 100
Catastrophic breakup		> 100

Other categorizations identify more or fewer regimes; the above is representative.

Detailed modeling of even one breakup regime is difficult. At most, the goal of secondary breakup models is to predict when breakup occurs and what diameters result from it. A selection of models is provided in STAR-CCM+:

- KHRT (stripping, catastrophic)
- Reitz-Diwakar (bag, stripping)
- TAB (vibrational)
- SSD (catastrophic)

The most appropriate regime for each model is noted in parentheses. However it is not unusual to find models that are applied outside these regimes. Where possible, results from more than one breakup model can be compared to add confidence.

These models become available when a gas model (single- or multi-component) is selected. It is not compatible with the Multiphase Mixture model.

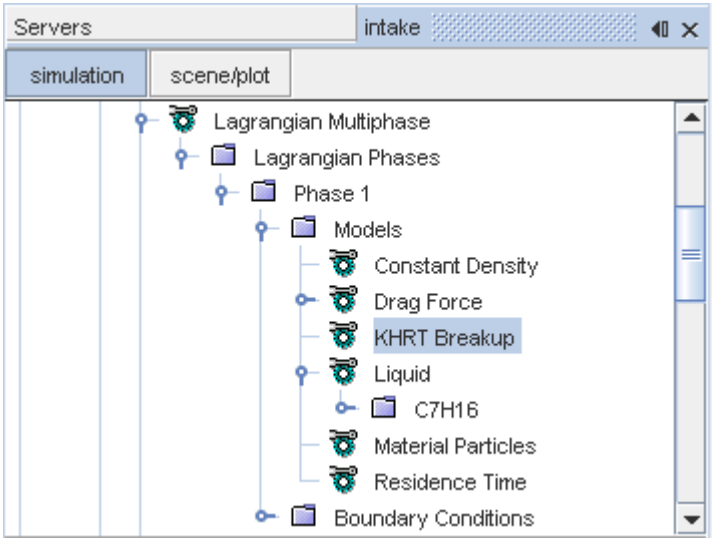
Contents:

What Is the KHRT Droplet Breakup Model?
What Is the Reitz-Diwakar Droplet Breakup Model?
What Is the TAB Droplet Breakup Model?
What is the SSD Breakup Model?

What Is the KHRT Droplet Breakup Model?

The KHRT Breakup model combines two submodels, one based on Kelvin-Helmholtz (KH) theory and one based on Rayleigh-Taylor (RT) theory.

The **KHRT Breakup** node represents this model in the Lagrangian phase model tree.



Both breakup submodels consider the growth of instabilities on a droplet and provide expressions for their wavelength and frequency, as discussed in the Theory section. Kelvin-Helmholtz instabilities are due to the slip velocity of the droplet, which eventually shears small child droplets off the parent [597], corresponding to the stripping regime. Rayleigh-Taylor instabilities are due to the acceleration of the droplet and tend to shatter the droplet completely [594], corresponding to the catastrophic regime. The KH and RT submodels compete: instabilities due to both can grow simultaneously; if they grow for long enough, breakup occurs due to the RT instabilities. Otherwise KH breakup occurs.

Because there is a tendency for the KH submodel to generate a bi-modal size distribution, the smaller child droplets are accumulated into a new parcel; the parent parcel retains the larger parent droplets. However, to prevent too many child parcels being created, these parcels are only released when their mass reaches a predefined proportion of the original mass of the parent. Child parcels are given a lateral velocity proportional to the kinetic energy of the droplet oscillation at the instant of breakup. This velocity tends to generate a spreading effect from an injector which, to some extent, makes a cone injector unnecessary.

The KHRT model was developed as a replacement for the TAB model [594]. Although the latter can still be preferred at low Weber numbers, the KHRT model is preferred at higher Weber numbers. The principal difficulty in using the model is tuning the coefficients to the application under consideration. This difficulty reflects the complexity of the physics and is not due to a flaw in the model.

KHRT Breakup Properties

Child Parcels	Number of child parcels to create during a breakup event.

KHRT Breakup Expert Properties

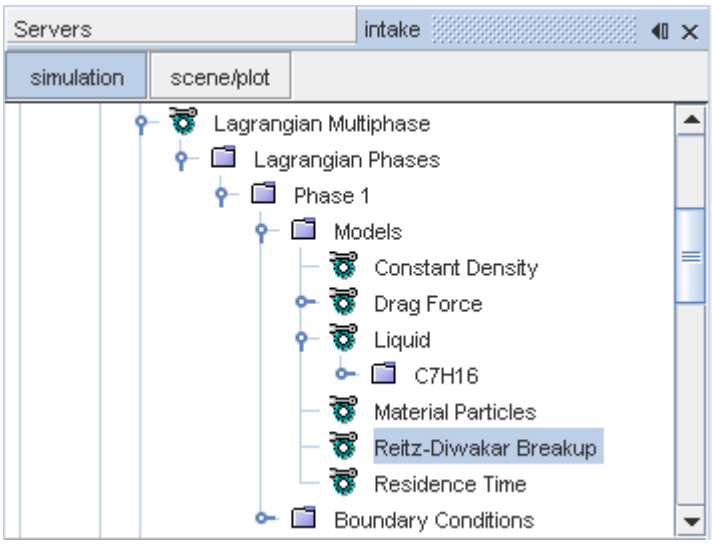
Normal Velocity Coefficient	Normal velocity coefficient for child parcels, A_1 in Eqn. (3092)
B0	KH length coefficient, Eqn. (3090)
B1	KH time coefficient, Eqn. (3089)

C3	RT length coefficient, Eqn. (3095)
Ctau	RT time coefficient, Eqn. (3094)
Maximum fractional shed mass	Maximum fraction of the parcel mass that is shed before the child parcels are created.

What Is the Reitz-Diwakar Droplet Breakup Model?

The Reitz-Diwakar breakup model is based on observed length- and time-scales of droplet breakup.

The **Reitz-Diwakar Breakup** node represents this model in the Lagrangian phase model tree.



As discussed in the Theory section, the Reitz-Diwakar breakup model, [598] and [599], assumes that breakup occurs in one of two possible modes

- “bag” breakup, in which the non-uniform pressure field around the droplet causes it to expand in the low-pressure wake region and eventually disintegrate when surface tension forces are overcome.
- “stripping” breakup, in which liquid is sheared or stripped from the droplet surface.

The model is simple and does not generate child parcels, hence it is relatively cheap to run. It can be used to estimate the importance of breakup before turning to a more complex model, or to provide support to conclusions obtained from such a model.

Reitz-Diwakar Properties

None

Reitz-Diwaker Expert Properties

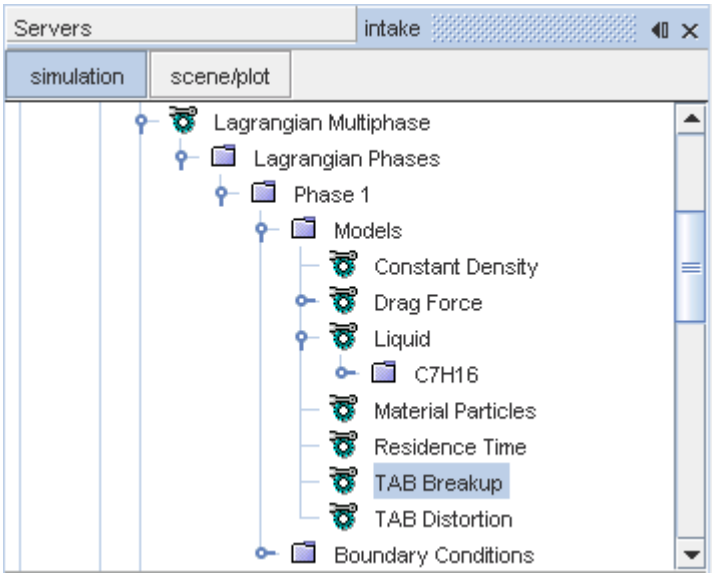
WeCrit	Minimum Weber number for breakup, see Eqn. (3097).
Cb2	Bag breakup time-scale coefficient, see Eqn. (3100).

Cs1	Coefficient prescribing the onset of stripping breakup, see Eqn. (3099).
Cs2	Stripping breakup time-scale coefficient, see Eqn. (3100).

What Is the TAB Droplet Breakup Model?

The TAB breakup model is an extension of the TAB distortion model which assumes that breakup occurs when the TAB distortion reaches a prescribed magnitude.

The **TAB Breakup** node represents this model in the Lagrangian phase model tree.



The TAB breakup model begins with the assumption that breakup occurs when the TAB distortion, as calculated by the TAB distortion model, exceeds unity. Breakup replaces the parent particles with child particles whose diameter is chosen from a Rosin-Rammler distribution (see the Theory section). By default, the breakup event creates no parcels—the original parcel is retained, but the particle diameter change to the new child value. New parcels can optionally be created however, in which case the droplet diameter in each is chosen independently. This feature can be useful in modeling the generation of a distribution of droplet sizes through breakup. Parcels that are involved in a breakup event are given a lateral velocity proportional to the kinetic energy of the droplet oscillation at the instant of breakup. This velocity tends to generate a spreading effect from an injector which, to some extent, makes a cone injector unnecessary.

Despite being based on a single mode of oscillation in the vibrational regime, the TAB model reproduces the same characteristic time-scales in low and high Weber number limits as the Reitz-Diwakar model. Typically, however, the TAB model is used at low Weber numbers; hollow-cone gasoline sprays are an example of a preferred application for this model [570]. Outside its range of validity, the model tends to underpredict droplet sizes.

TAB Breakup Properties

Child Parcels	Number of child parcels to create during a breakup event.

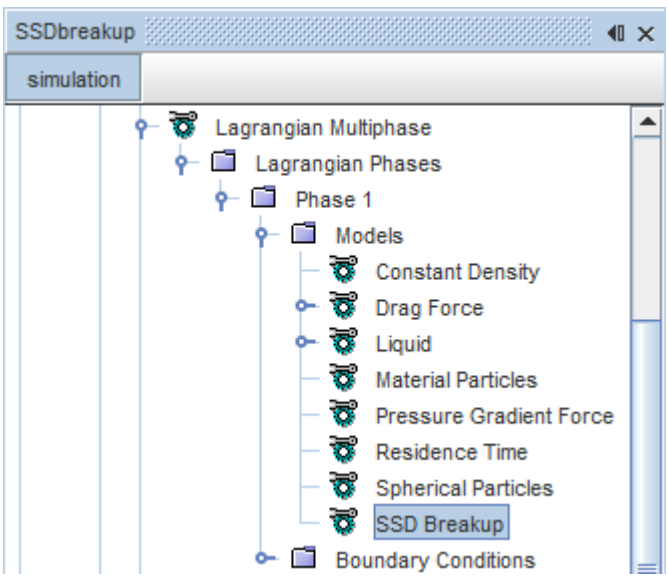
TAB Breakup Expert Properties

Cv	Normal velocity coefficient, see Eqn. (3104).
K	Energy ratio, see Eqn. (3103).
Rosin-Rammler Exponent	Exponent in Rosin-Rammler size distribution.
Cb	Critical distortion coefficient, see Eqn. (3101).

What is the SSD Breakup Model?

In the Stochastic Secondary Droplet (SSD) breakup model, breakup is a random process unrelated to original droplet size.

This results in a log normal distribution of droplet diameters in the long time limit, as discussed in the Theory Guide. The SSD Breakup node represents this model in the simulation tree.



The particles interact with the Eulerian fluid phase as they travel. Particles break up when three criteria are met:

- The particle radius is larger than the critical radius (calculated from local droplet conditions using Eqn. (3105)).
- The Weber number (calculated from local droplet conditions) is larger than a critical Weber number (set in the interface).
- The particle's local time (the accumulated residence time) is larger than the breakup time (Eqn. (3106)). The model tracks parcels as they cross the domain and compares the residence time with a locally evaluated breakup time. The residence time is only calculated on parcels with a radius larger than a critical radius and a Weber number larger than a critical Weber number.

When the particle breaks up, the local time is reset for the new parcels.

You can control the average number of particles per child parcel and the maximum number of child parcels formed in each breakup event using the *Target Count* and *Maximum Child Parcels* properties. The model uses these values when calculating the size, number of parcels, and particles per parcel (parcel count) of the child parcels. This gives you some control over the description of the spray.

Note that neither *Target Count* nor *Maximum Child Parcels* changes the physics of the breakup event, since these parameters do not change the diameter distribution functions. Rather, these parameters influence the total number of parcels in the calculation and thus the relative expense of the computation. A larger number of samples usually means an improved representation of the drop size distribution.

This model differs from other breakup models in three ways:

- The parent drop no longer exists after the breakup event. (The particle numbering system re-numbers after the breakup, so a particle before the event with the same number as a particle after the event is not the same particle.)
- Because the model works with a user-specified particle count, you seldom have very large particle counts that are difficult to handle numerically.
- The breakup distribution function, Φr in Eqn. (3107), is a function of local conditions and depends only weakly on the diameter of the parent. Though the distribution is limited to the range between the current radius and half the critical radius, the function depends more on the Weber number, which is a strong function of local flow.

SSD Breakup Properties

WeCrit	The critical Weber number, We_{cr} in Eqn. (3105). The larger this value is, the larger droplets can get before breaking up. The default value is 6.
Target Count	The average number of particles per child parcel, derived from Eqn. (3107). The default value is 1000.
B1	The breakup parameter, B_1 in Eqn. (3106). The larger this value is, the later breakups occur. The default value is $\sqrt{3}$.

SSD Breakup Expert Properties

K0	Controls the mean distribution of child diameters; K_0 , equal to $\langle \xi \rangle$ in Eqn. (3108). The default value is -0.1.
K1	Controls the breadth of the distribution of diameters; K_1 in Eqn. (3108). The default value is 0.1.
Maximum Child Parcels	The maximum number of parcels that can result from a breakup. Use this to limit the greater computational expense of a large number of child parcels. The default value is 10.
Normal Velocity Coefficient	Controls the radial diffusion of the spray; K_3 in Eqn. (3109). The larger it is, the more rapid the radial diffusion of the spray. Use a value of 0 if the spray angle is known and fixed at the injector. The default value is 0.1.

Working with Injectors

In STAR-CCM+, particles enter the fluid continuum through injectors. While the Lagrangian phase defines what particles enter the domain and how they behave, the injector defines where, in what direction, and with what frequency, the particles enter.

Injectors are contained within a top-level injectors manager node.

The following types of injector are currently provided in STAR-CCM+:

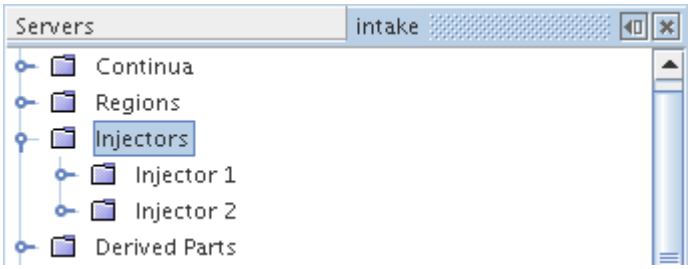
- The Part injector generates a cloud of injection points that are positioned according to the geometry of the parts you select.
- The Point injector injects particles from a single point in a direction you specify.
- The Hollow Cone injector injects particles from a single point, with inner, and outer angles you specify.
- The Solid Cone injector injects particles from a single point, with only the outer angle you specify.
- The Surface injector injects particles from the face centroids of a geometric surface you select.
- The Nozzle injector is used with the Linear Instability Sheet Atomization (LISA) model and Huh model to generate particles produced by atomization.
- The Lattice injector is used with the Discrete Element Method (DEM) to inject particles into the system with specified, mesh-independent spatial distribution.
- The Random injector is used with the Discrete Element Method (DEM) to inject particles into a region with a random point distribution. The Particle Packing Specification controls the number of particles that are injected.
- The Parcel Transfer injector is used to reinject parcels of one phase as parcels of another, according to criteria you specify.
- The Film Stripping injector is used with the Fluid Film model, to simulate droplets being formed and removed from the surface of a fluid film. Both wave stripping and edge stripping can contribute to this phenomenon.

These types are explained in greater detail in the section, Understanding Injector Types. Further documentation is provided in:

- Using injectors
- Using injector particle size distributions
- Injector Conditions Reference
- Injector Values Reference

Injectors Manager Node

Injector nodes represent injectors and are all contained within the **Injectors** manager node, which has properties and its own pop-up menu. The injector manager can hold any number of injectors up to the memory limits of the computer.



Injector Manager Expert Properties

Tree Threshold	Specifies the maximum estimated number of injection points per region before the algorithm changes to a tree search. There is a memory and CPU time penalty that is incurred in building the tree. If there are enough points, the algorithm repays this penalty by using the tree search to locate the points faster than the default brute-force algorithm.

Contents:

Understanding Injector Types
Using Injectors
Using Injector Particle Size Distributions
Using Injection Tables
Using Cone Angle Tables
Injector Conditions Reference
Injector Values Reference

Understanding Injector Types

Each type of injector has distinctive features that determine where and how particles are injected.

The following injector types are provided in STAR-CCM+:

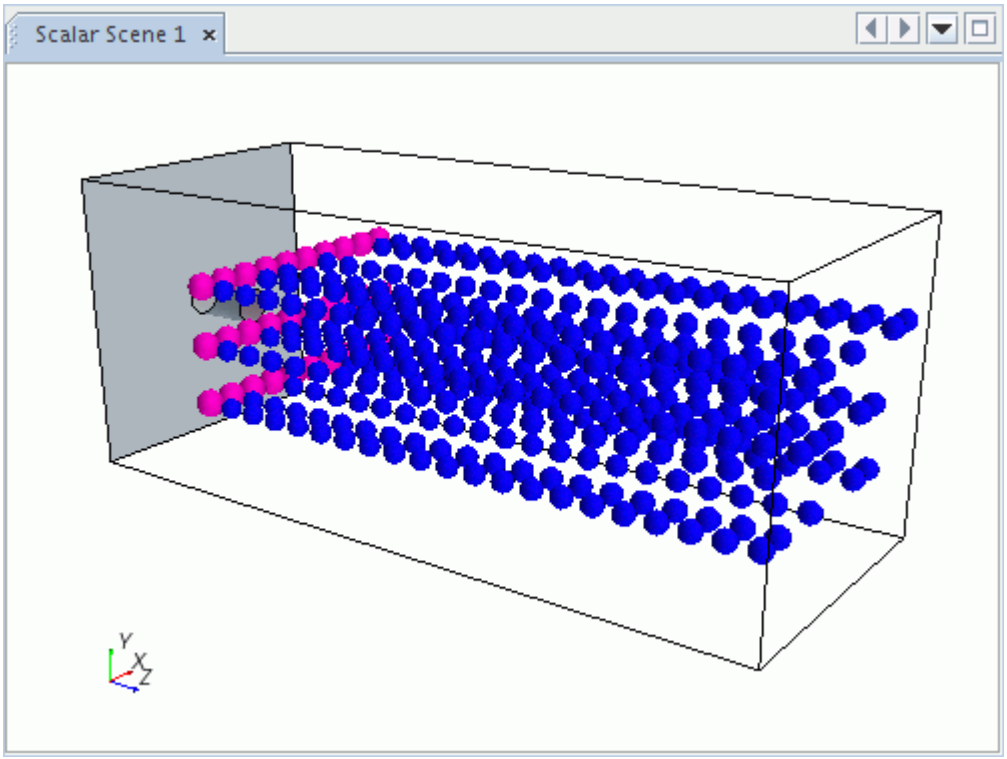
- Part injector
- Point injector
- Hollow Cone injector
- Solid Cone injector
- Surface injector
- Nozzle injector
- Table injector
- Parcel Transfer injector
- Lattice injector—used with the Discrete Element Method (DEM)
- Random injector—used with the Discrete Element Method (DEM)
- Film Stripping injector—used with the Fluid Film model

Each type has distinctive features that determine where and how particles are injected. A common property among all injector types is the injector Inputs property. As its use varies between injector types, it is important to understand how the property is used in each injector: this property is discussed in the individual descriptions below. The type of the injector can also collaborate with the Lagrangian phase models to determine which conditions and values are necessary to define the initial conditions completely.

A key role of the type of injector is to determine how the injection points are generated from the Inputs of the injector. Injection points are where the injector introduces particles—in other words the initial positions of the particles.

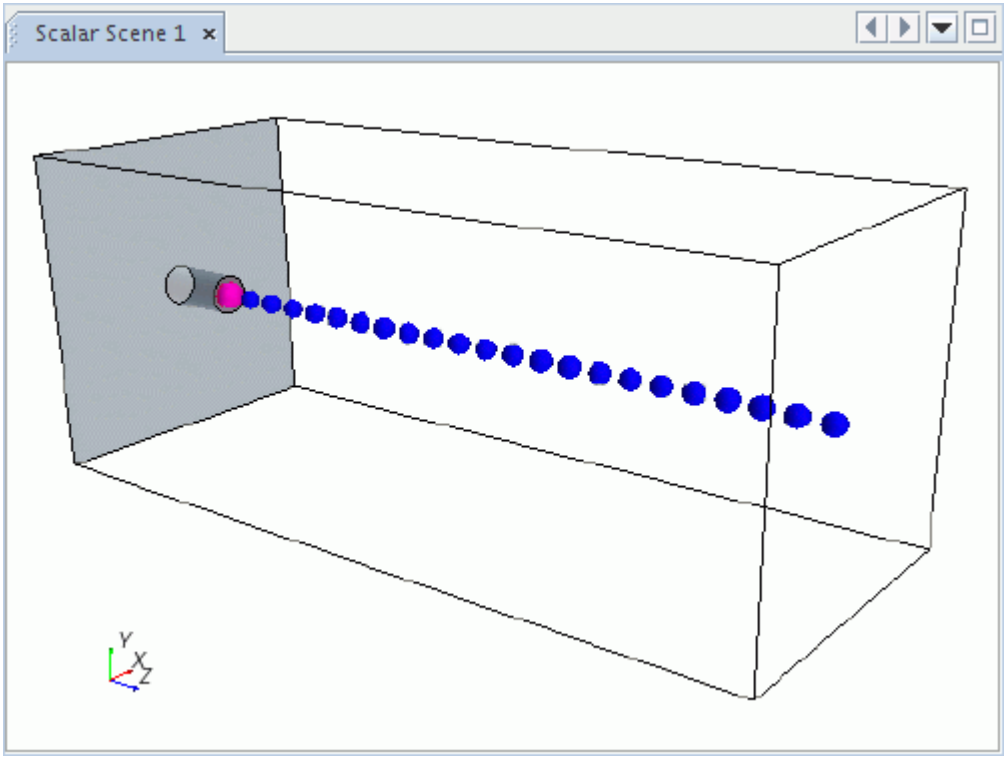
Part Injector

A part injector represents a collection of injection points that are extracted from the geometry of the parts of the injector. If desired, the points can be a subset of the points that are defined on the parts. The point inclusion probability property of the injector sets the probability that a particular point is used. An example of a part injector (defined by three line probes) is shown below.



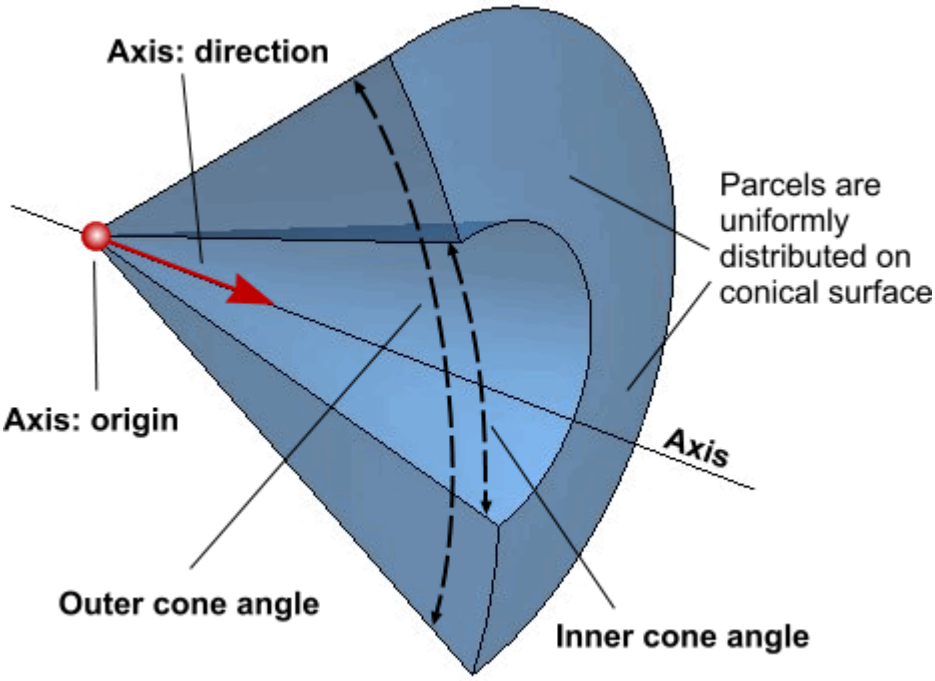
Point Injector

A point injector injects at a single user-specified point which is the position of the injector. The initial cell location for this point is found by searching the parts of the injector, which are limited to regions. Thus, if you can specify which region contains the point, the input parts can be used to accelerate the cell location process. You can also use input parts to resolve ambiguities. For example, suppose that the point lies on the interface between two regions but the intention is not to inject into one of them. An error is thrown if a cell location for the point injector cannot be found by searching its input parts. Typically this error means that the point is outside the specified region or regions.

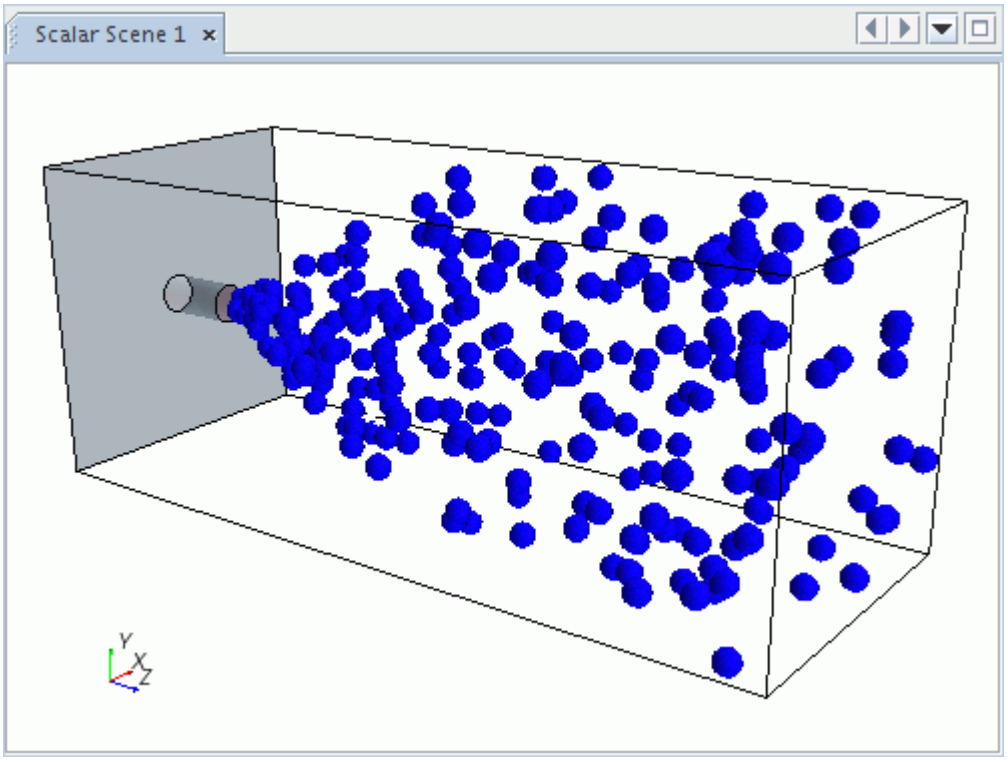


Hollow Cone Injector

A hollow cone injector injects parcels in a cone formation about an axis you specify. An injector diameter, an inner cone angle, and an outer cone angle define the cone, as shown in the diagram below. (The diagram shows only half the cone; the injector uses a whole cone. The example shown has an injector diameter of zero.)



The specified number of parcel streams are injected in random directions conforming to a uniform distribution on the surface of the cone. Given the direction, the velocity is obtained using the specified velocity magnitude. In steady simulations, each iteration uses the same stochastically chosen direction and size distribution. In unsteady simulations, the direction and size distribution are randomized again and newly sampled each time-step, if you activate the Cone Randomization property. For this injector, the Inputs property is used to define the region within which the origin of the axis is located.



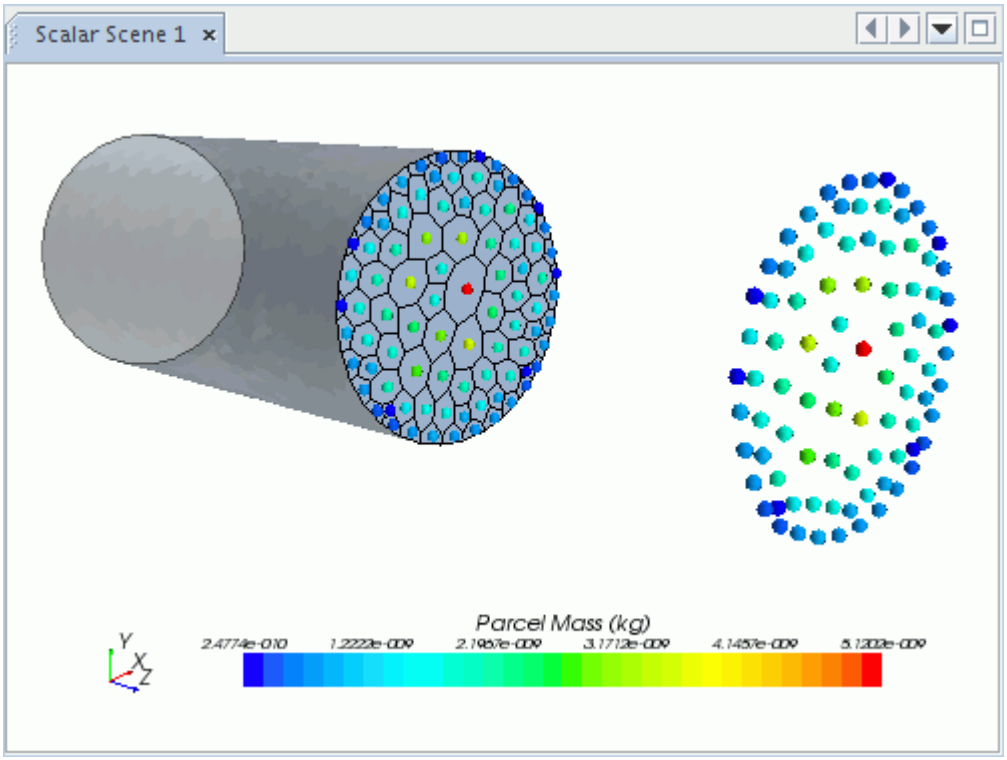
Solid Cone Injector

A solid cone injector is similar to the hollow cone type, except that there is no inner cone angle parameter (the angle is treated as zero).

Surface Injector

A surface injector injects parcels from each face of the surface comprising the parts of the injector. Parcels are injected at the centroid of each face in the direction of the face surface normal. For this type, parts are selected from those parts that belong to the model geometry.

A unique feature of the surface injector type is that the injection flow rate can be defined with a flux, that is, as a rate per unit area. The final flow rate is the product of the face area and the specified flux, whether this flux is a particle, mass or volume flux. In the following example, the initial location of each parcel coincides with the centroid of each face on the starting surface. The mass of each parcel is proportional to the area of the face from which it came (as a mass flux has been specified).



When the flow rate through a surface injector is defined directly, the flow rate through each face scales to the ratio of the face area to the overall surface area.

Nozzle Injector

A nozzle injector requires two basic parameters: the diameter of the injector at the nozzle exit and the origin and axis of the injector. The nozzle injector becomes available only if you choose of the primary atomization models, LISA or Huh. Additional model-specific parameters are then added, based on the selected model.

When used with the LISA Atomization model, this injector type represents a pressure-swirl atomizer. A thin liquid film is formed within the injector nozzle that then breaks up into droplets soon after entering the gas. A mathematical model represents the breakup process. This model derives from a stability analysis of the two-dimensional liquid film.

For this injector type, the Inputs property is used to define the region where the origin of the axis is located. The inner cone angle and outer cone angle are used to calculate the spray angle, which is defined as half the sum of the inner and outer cone angles. The model also requires an injector upstream pressure value to predict the exit velocity of the thin liquid film.

When used with the Huh atomization model, this injector represents a liquid jet breaking up into droplets as it enters the gas. The initial droplets have diameters equal to the nozzle diameter, then break up according to the Huh model.

Since Huh model estimates the initial turbulence-generated perturbations from an analysis of the flow through the nozzle, it requires additional nozzle-related parameters, as outlined in the formulation (see Huh Model).

Note: Only parcels injected from a nozzle injector can go through primary atomization. Parcels from other types of injector skip the primary atomization process.

These injectors are only available for use with the three-dimensional space model.

Table Injector

A table injector injects parcels from points that are specified in an input table. The table also specifies the injection times and the properties of the parcels, such as mass, diameter, and temperature.

Input parts for a table injector must be regions. The injection points can be anywhere within a specified region.

Parcel Transfer Injector

The Parcel Transfer injector replaces parcels of a selected input phase with parcels of a selected output phase, if they meet specified criteria.

Example: If particles modeled with DEM become smaller through breakup or reaction, or move into a region with large cells, the particles are then small compared to the distances between them. Collisions between particles then become insignificant. In such a case, non-DEM Lagrangian parcels can be a better model. You can use the Parcel Transfer injector to reinject the DEM particles as Lagrangian parcels, at the same position.

Input and output phases can be either simple Lagrangian parcels or DEM parcels (particles). However, the combination of Lagrangian input and DEM output is not allowed.

Input and output particles must be spherical, isothermal, and of a single component. The transfer preserves the total mass and velocity (and therefore total momentum and kinetic energy). Input and output particles have the same injector index.

The **Transfer Criterion** value node controls which parcels in the source phase the injector reinjects into the destination phase. Click ... (**Custom Editor**) and enter the criterion in the dialog window. Example:

```
$ParticleDiameter < 0.05
```

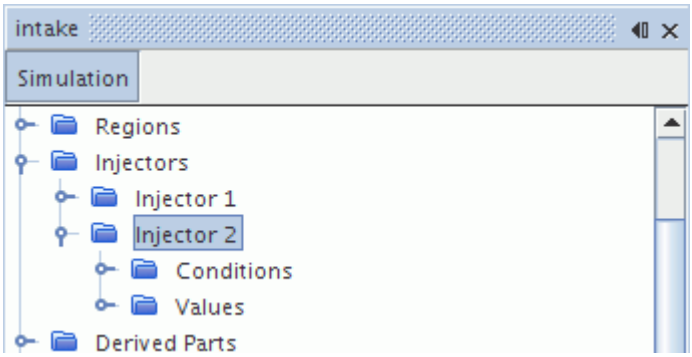
This criterion transfers particles in the source phase with a diameter below 0.05 m.

If you create more than one Parcel Transfer injector, the injectors act in order of their *Index* numbers. First, injector 1 reinjects all DEM particles that meet criterion 1, then injector 2 reinjects the remaining DEM particles meeting criterion 2, and so on.

Using Injectors

An injector has a type, which determines how it generates injection points. Particles are introduced at these injection points and are assigned to the user-specified Lagrangian phase.

Injector nodes represent injectors within the injector manager. They are created using the right-click menu on the injectors manager node. The injector node has its own properties. It contains condition and value objects that apply to the Lagrangian phase to which it is assigned.



The goal of an injector is to provide the initial state of particles, according to its user-specified conditions and values. The injector type and the models of the Lagrangian phases determine these conditions and values.

Injector Properties

Inputs	Selects the parts that are used to generate the injection points. How the parts are used and which parts can be selected depend on the injector type.	
Type	The injector type. The possible types are listed below.	
	Film Stripping	A film stripping injector.
	Lattice Injector	A lattice injector.
	Part Injector	A part injector.
	Point Injector	A point injector.
	Hollow Cone Injector	A hollow cone injector.
	Solid Cone Injector	A solid cone injector.
	Random Injector	A random injector.
	Surface Injector	A surface injector.
	Nozzle Injector	A nozzle injector.
	Parcel Transfer	A parcel transfer injector.
	Table Injector	A table injector.
Number of Points	The number of injection points this injector produces (read-only). The injection points are first generated when the solution is initialized.	
Index	A unique identifier for this injector (read-only)	
Lagrangian Phase	Selects the Lagrangian phase from a drop-down list.	

To set up an injector, do the following:

1. Create the injector
2. Assign a Lagrangian phase
3. Set the injector type

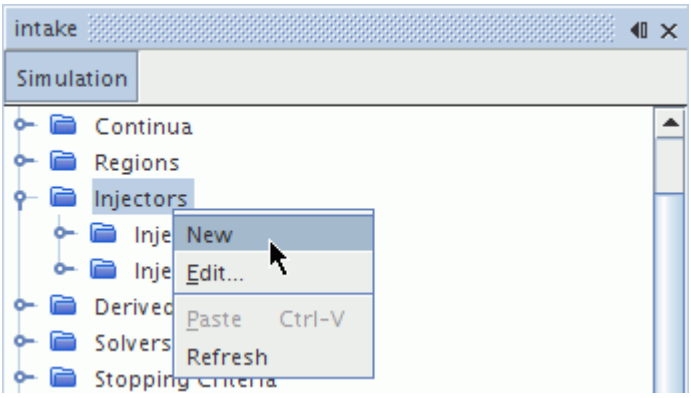
4. Assign the injector parts
5. Set conditions and values

Other operations that can be performed on an injector node are:

- Renaming an injector
- Deleting an injector

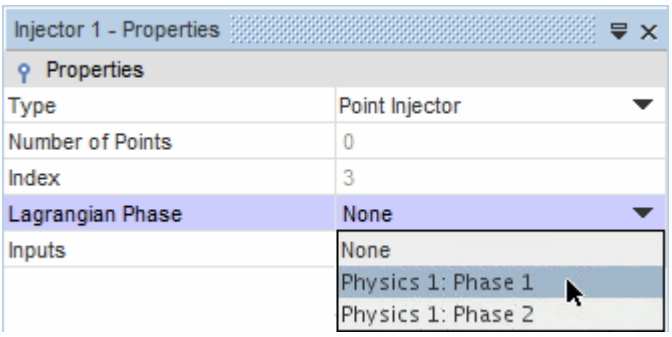
Creating an Injector

To create an injector, right-click the **Injectors** manager node and select **New**.



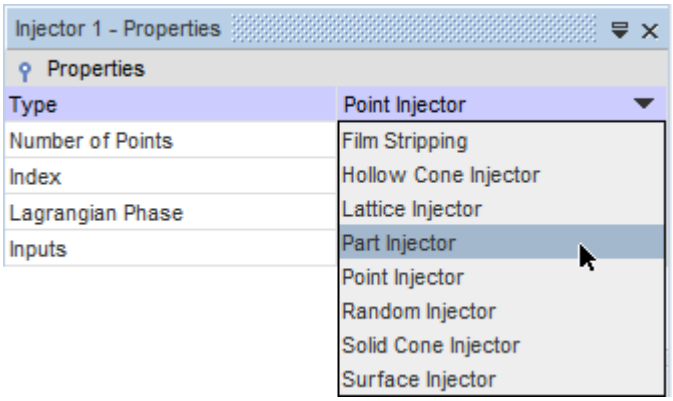
Assigning a Lagrangian Phase to the Injector

To assign a Lagrangian phase to an injector, select the injector node and select that phase from the drop-down list of the *Lagrangian Phase* property.



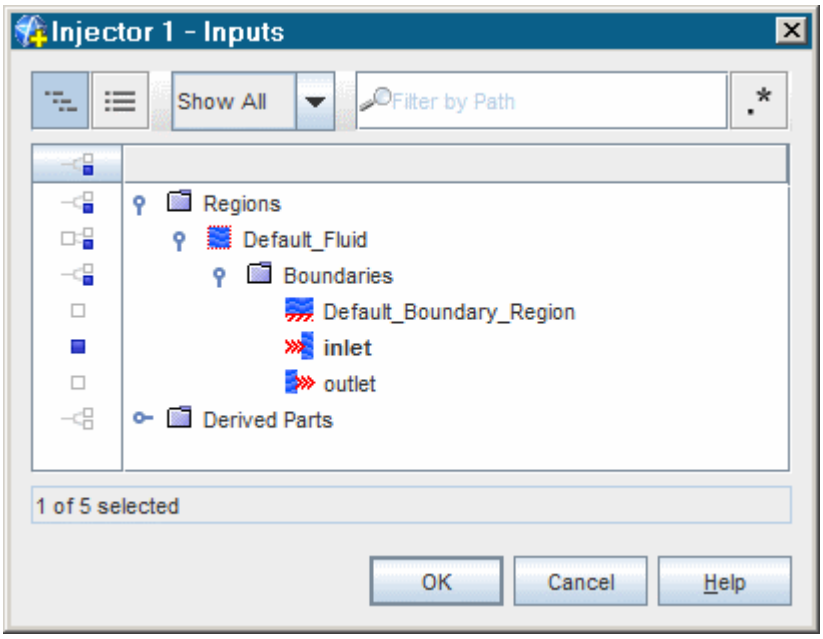
Setting the Injector Type

To set the injector type, select the injector node and select the required type from the drop-down list of the *Type* property.



Assigning the Injector Parts

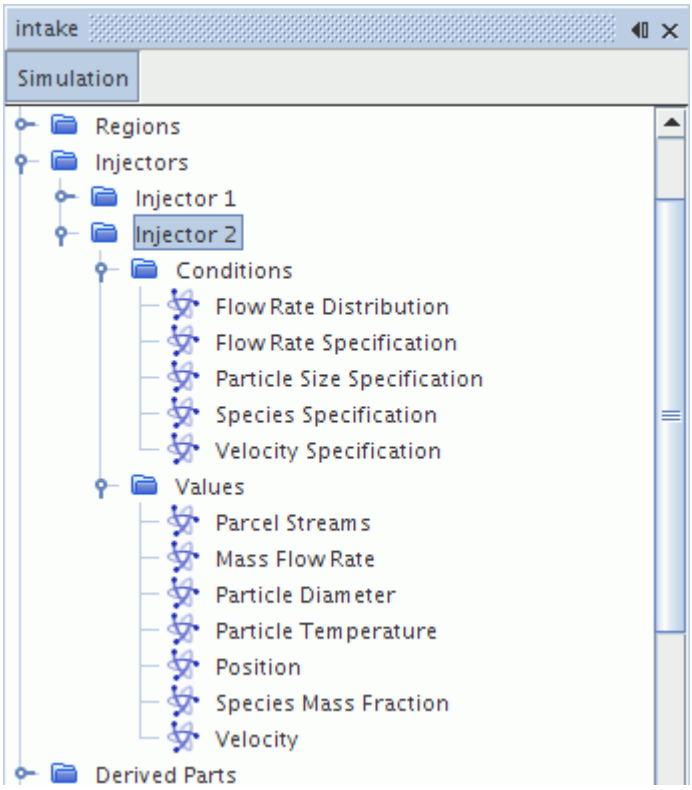
To assign the input parts of the injector, click the right half of the *Inputs* property of the injector node. This action activates the *Inputs* dialog.



The displayed selection contains parts appropriate to the injector type; not all parts are appropriate for every injector.

Setting Conditions and Values

Each injector contains sub-nodes that are managers of injector conditions and values, which work in ways similar to the conditions and values for a STAR-CCM+ simulation.



Renaming an Injector

Similar to other objects, an injector can be renamed within STAR-CCM+.

Deleting an Injector

Injectors can be deleted from an existing simulation.

Injector Manager Pop-Up Menu

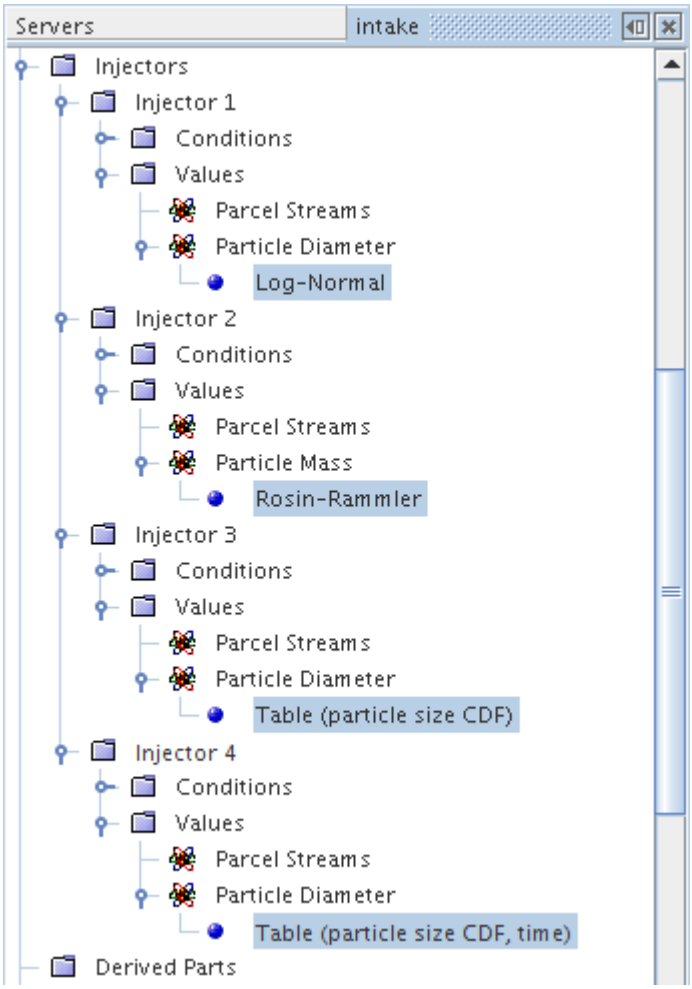
This pop-up menu item is available when you right-click the **Injectors** manager node.

New	Creates an injector with a default name.

Using Injector Particle Size Distributions

Particle size specification at an injector can be either by diameter or mass.

Among the methods available, four define the particle size using a particle size distribution, each represented by a node.



The available methods are:

- Constant
- Field Function
- Log-Normal (using a particle size distribution)
- Normal
- Nukiyama-Tanazawa (using a particle size distribution)
- Rosin-Rammler (using a particle size distribution)
- Table (iteration)
- Table (particle size CDF) (using a particle size distribution)
- Table (particle size CDF, time) (using a particle size distribution)
- Table (r)
- Table (time)
- Table (x,y,z)
- Table (xyz,time)
- User Code

In each case, a range of particle sizes is generated between a user-specified minimum and maximum, according to the distribution of the method.

What Is Log-Normal?

The Log-Normal size distribution is a smooth continuous distribution depending on four parameters: mean size, standard deviation, minimum size, and maximum size.

Log-Normal Size Distribution Properties

Mean	The mean size, $\overline{D}$ in Eqn. (3049).
Standard Deviation	The standard deviation, σ in Eqn. (3049).
Minimum	The minimum size that the distribution generates.
Maximum	The maximum size that the distribution generates.

What Is Normal?

The Normal size distribution is a smooth continuous distribution depending on four parameters: mean size, standard deviation, minimum size, and maximum size.

Normal Size Distribution Properties

Mean	The mean size, $\overline{D}$ in Eqn. (3050).
Standard Deviation	The standard deviation, σ in Eqn. (3050).
Minimum	The minimum size that the distribution generates.
Maximum	The maximum size that the distribution generates.

What is Nukiyama-Tanazawa?

The Nukiyama-Tanazawa size distribution is a smooth continuous distribution depending on three parameters: mean size and two model constants, α and β.

Nukiyama-Tanazawa Size Distribution Properties

Mean	The mean size, $\overline{D}$. See Eqn. (3051).
Alpha	A model constant α with a default value of 2. See Eqn. (3051).
Beta	A model constant β with a default value of 1. See Eqn. (3051).

What Is Rosin-Rammler?

The Rosin-Rammler size distribution is a smooth continuous distribution depending on four parameters: reference size, exponent, minimum size, and maximum size.

Rosin-Rammler Size Distribution Properties

Exponent	The exponent, q in Eqn. (3052).
Reference	The reference size, D_{ref} in Eqn. (3052).
Minimum	The minimum size that the distribution generates.
Maximum	The maximum size that the distribution generates.

What Is Table (particle size CDF)?

A table can be used to specify the particle size distribution using a cumulative distribution function (CDF). A particle size CDF table has a column of 'CDF' values and a column of 'data' values for the particle sizes.

The CDF values must be strictly increasing—duplicate values are not permitted.

Table (Particle Size CDF) Properties

Table: CDF	Looks for the column in the table with the heading CDF (read-only).
Table: Data	Identifies the selected column in the table file as the data column containing the particle sizes for the corresponding CDF values.
Table	Selects a table from among the table sub-nodes in the Tables node of the Tools node.

You must have read a table into STAR-CCM+ to specify these properties; otherwise the drop-down lists are empty and the *Table: Data* and *Table* properties show as Null. The other property, *Table: CDF*, is read-only since it simply indicates the title of the column that the tabular method expects.

What Is Table (particle size CDF, time)?

This table can be used to specify a particle distribution using a cumulative distribution function (CDF) that varies over time.

The first column of the table lists the sizes of the particles, which must increase monotonically. Subsequent columns have headers of the form `CDF [t=####]` where `####` is a time value. These time values must increase monotonically, without duplicates, as you go down the header row. Each column of the table corresponds to the starting time for use of the next CDF.

Under each time column, each entry is a CDF value; these must increase monotonically down the column, without duplicates, though the rate of increase is independent between columns.

Example:

```
diameter CDF [t=0.01] CDF [t=0.50] CDF [t=1.2] CDF [t=1.24]
0.01     0.000        0.000        0.00        0.000
0.03     0.002        0.030        0.05        0.004
0.04     0.004        0.040        0.12        0.078
0.06     0.007        0.050        0.89        0.512
0.11     1.000        1.000        1.00        1.000
```

Table (Particle Size CDF, Time) Properties

Table: Particle Size	The header of the column listing particle sizes. In the example, this is **diameter**.
Table: Starting Time Column	The header of the column for the starting simulation time, for example, **CDF [t=0.50]**. The starting time must be less than or equal to the starting time of the simulation.
Table	Selects a table from among the table sub-nodes in the Tables node of the Tools node.
Interpolation	Selects interpolation method, **Spline**, **Step**, or **Linear**.

You must have read a table into STAR-CCM+ to specify these properties; otherwise the drop-down lists are empty and the properties show as Null.

Using Injection Tables

Injection tables are the input method for table injectors. These tables specify the location of injection points and the time of injection. They can also specify some properties of the injected parcels.

There are three kinds of injection table:

- Injection Points table
- Unsteady Injection Points table
- Parcels table

Injection Points Table

Injection Points tables are available with both steady and unsteady Lagrangian Multiphase simulations. A table injector using an Injection Points table has the following features:

- Each row of the table represents an injection point.
- Any injection point that is not in the parts specified under the injector Inputs property is discarded.
- The injector uses the Parcel Streams value for the number of parcel streams per injection point.
- Parcels are injected according to the Update Frequency property of the solver.
- The injector can use any of the Flow Rate Distribution options, but only the **Per Injector Point** or **Per Parcel Stream** options allow you to specify the flow rate in the injection table.
- The table can include any properties of the parcels listed under the physics values node of the injector. Examples of such properties include particle diameter, velocity, temperature, passive scalar values, parcel mass fraction, and parcel mole fraction.

Example table:

X(cm)	Y(cm)	Z(cm)	Diameter	U	V	W	Mass Flow Rate	Temperature
0	0	6.0	1.e-05	0	0	0.001	0.001	350
0	0	2.0	2.e-05			0.002	0.001	350

Injection Points Table Properties

The table is represented by a node named **Injection Points Table**, located under the injector **Values** node. The table node has the following properties:

Table	The table name, selected by drop-down menu from the tables defined under **Tools** > **Tables**. The default is **None**.
X coordinate	The names specified in the table for the columns for the X, Y, and Z coordinates. For the example table in this section, the entries would be `X(cm)`, `Y(cm)`, and `Z(cm)`.
Y coordinate	
Z coordinate	
Coordinate System	The coordinate system in which the injection points are specified. This can be the laboratory system or a local coordinate system you have created. This coordinate system applies only to the injection points, not to any other related properties, such as velocity. The default is **Laboratory**.

Unsteady Injection Points Table

Unsteady Injection Points tables are available only with unsteady Lagrangian Multiphase simulations. A table injector using an Unsteady Injection Points table has the following features:

- Each row of the table represents an injection point at a specific time value.
- The total number of injection points is equal to the maximum index found in the table.
- Each injection point is identified in the table by a unique positive index.
- Numbering for the injection point indices starts from 1.
- For every injection point index, there is at least one row in the table.
- For every injection point index, the time values must be specified in ascending order.
- At any given time, for each injection point index, the locations are interpolated in time as follows:
  - If the current value of time is less than the minimum time value found for that injection point, parcels are injected from the location corresponding to the minimum time value.
  - If the current value of time is greater than the maximum time value found for that injection point, parcels are injected from the location corresponding to the maximum time value
- Any injection point that is not in the parts specified under the injector Inputs property is discarded.
- The injector uses the Parcel Streams value for the number of parcel streams per injection point.
- Parcels are injected according to the Update Frequency property of the solver.
- The injector can use any of the Flow Rate Distribution options, but only the **Per Injector Point** or **Per Parcel Stream** options allow you to specify the flow rate in the injection table.
- The table can include any properties of the parcels listed under the physics values node of the injector. Examples of such properties include particle diameter, velocity, temperature, passive scalar values, parcel mass fraction, and parcel mole fraction.

Example table, for a simulation with time-step of 0.001:

Index	X(cm)	Y(cm)	Z(cm)	Diameter	Time
1	0	0	1.0	1.e-05	0.0012

Index	X(cm)	Y(cm)	Z(cm)	Diameter	Time
2	0	0	2.0	2.e-05	0.0022
1	0	0	3.0	3e-05	0.0032
3	0	0	3.0	3e-05	0.0035
2	0	0	4.0	4e-05	0.0042

There are three injection points, indexed 1, 2, and 3. At every time-step, all three injection points inject parcels. At every time-step, the position of the injection point is interpolated, based on the values given in the table.

- For the time 0.001, corresponding to the first time-step, points inject from their earliest given position. For example, the earliest time specified for point 1 is 0.0012 s, which is greater than the current physical time, so point 1 injects from position 1 cm.

Injection Point	Position
1	1 cm
2	2 cm
3	3 cm

- At time 0.002, corresponding to the second time-step, there are no positions specified for points 1 or 3.
  - Point 1 injects from a position between 1 cm and 3 cm; the exact position depends on the interpolation option specified.
  - Point 3 injects from a position of 3 cm, because that is still its earliest specified position.
- At time 0.003, corresponding to the third time-step, there are no positions specified for points 2 or 3.
  - Point 2 injects from a position between 2 cm and 4 cm; the exact position depends on the interpolation option specified.
  - Point 3 injects from a position of 3 cm, because that is still its earliest specified position.
- After the value for time passes 0.004, all three injection points inject parcels from the position for the last, highest time value:

Injection Point	Position
1	3 cm
2	4 cm
3	3 cm

Unsteady Injection Points Table Properties

The table is represented by a node named **Unsteady Injection Points Table**, located under the injector **Values** node. The table node has the following properties:

Table	The table name, selected by drop-down menu from the tables defined under **Tools** > **Tables**. The default is **None**.

X coordinate	The names specified in the table for the columns for the X, Y, and Z coordinates. For the example table in this section, the entries would be X(cm), Y(cm), and Z(cm).
Y coordinate	
Z coordinate	
Coordinate System	The coordinate system in which the injection points are specified. This can be the laboratory system or a local coordinate system you have created. This coordinate system applies only to the injection points, not to any other related properties, such as velocity. The default is **Laboratory**.
Time	The name of the column listing time values (read-only). The column in the table must be named time, Time, or TIME.
Injection Point Index	The name of the column listing a unique positive index for each injection point (read-only). The column in the table must be named index, Index, or INDEX.

Expert Properties

Interpolation	Selects interpolation method, **Spline**, **Step**, or **Linear**.

Parcels Table

Parcels tables are available only with unsteady Lagrangian Multiphase simulations. A table injector using a parcels table has the following features:

- Each row of the table represents a parcel at a specific time value.
- Parcels are injected only at the times specified in the table. There is no interpolation.
- A set of rows with time values that lie between the current physical time and the current physical time plus the time-step are treated as a single block of parcels and are injected at the start of the most recent time-step.
- Any parcel that is not in the parts specified under the injector Inputs property is discarded.
- The injector uses the Particle Count value for the number of particles per parcel.

  This value can be constant or read from the injection table.

  This value is taken as 1 if the DEM model is active, or if the value is not set.
- The table can include any properties of the parcels listed under the physics values node of the injector. Examples of such properties include particle diameter, velocity, temperature, passive scalar values, parcel mass fraction, and parcel mole fraction.

Example table, for a simulation with time-step of 0.001:

X(cm)	Y(cm)	Z(cm)	Diameter	Time
0	0	1.0	1.e-05	0.0
0	0	2.0	2.e-05	0.0
0	0	3.0	3e-05	0.003
0	0	4.0	3e-05	0.0031

X(cm)	Y(cm)	Z(cm)	Diameter	Time
0	0	5.0	4e-05	0.0032

The table has two time blocks, corresponding to injection times 0.0 and 0.003 seconds.

- There are two parcels in the first time block, and three parcels in the second time block.
- In the first time step (corresponding to physical time = 0.001) two parcels are injected from 1.0 cm and 2.0 cm.
- In the second time step (corresponding to physical time = 0.002), no parcels are injected.
- In the third time-step (corresponding to physical time = 0.003), no parcels are injected.
- In the fourth time step (corresponding to physical time = 0.004) three parcels are injected from 3.0 cm, 4.0 cm and 5.0 cm.

Parcel Table Properties

The table is represented by a node named **Parcels Table**, located under the injector **Values** node. The table node has following properties:

Table	The table name, selected by drop-down menu from the tables defined under **Tools** > **Tables**. The default is **None**.
X coordinate	The names specified in the table for the columns for the X, Y, and Z coordinates. For the example table in this section, the entries would be X(cm), Y(cm), and Z(cm).
Y coordinate	
Z coordinate	
Coordinate System	The coordinate system in which the injection points are specified. This can be the laboratory system or a local coordinate system you have created. This coordinate system applies only to the injection points, not to any other related properties, such as velocity. The default is **Laboratory**.
Time	The name of the column listing time values. The column in the table must be named time, Time, or TIME.

Using Cone Angle Tables

Cone angle tables are available only with hollow or solid cone injectors. Using a cone angle table, you can specify the properties of the injected particles as a function of the angle of injection.

A cone angle table has the following features:

- Each row of the table represents the properties of the injected particles at a given injection angle.
- The injection angles are listed in a column entitled `ANGLE`, `Angle`, or `angle`.
- The other columns have the names of the injector values for the injected particles, for example `Mass flow rate` or `Particle diameter`.

Cone Angle Table Properties

Table	The table name, selected by drop-down menu from the tables defined under **Tools** > **Tables**. The default is **None**.
Table: Data	The name of the table column, selected by drop-down menu from the columns defined in the selected table, containing the data describing the particles injected at the specified angle.
Cone Angle	The name of the table column listing the injection angles (read-only). The column name must be `angle`, `Angle`, or `ANGLE`. If more than one of these names is used, `angle` is given the highest preference and `ANGLE` is given the lowest. If none of these names exists, the field reads `null`.

Injector Conditions Reference

This section describes the available injector conditions.

Injectors can be set up with the following conditions:

- Angular Velocity Specification
- Coarse Graining Option Specification
- Flow Rate Distribution
- Flow Rate Specification
- Injection Time Rate Specification
- Injection Volume Correction Specification
- Lattice Structure
- Orientation Specification
- Overlap Specification
- Particle Packing Specification
- Particle Size Specification
- Species Specification
- Time Randomization
- Velocity Specification

Angular Velocity Specification

DEM particles can be injected with an angular velocity specified using vector notation. Specifies the distribution of the flow rate.

Properties:

Method

Specifies the distribution of the flow rate.

Method	Corresponding Value Node
Components Enter the angular velocity as a vector with magnitude.	**Angular Momentum** Vector notation is used to specify the angular velocity of injected particles where the magnitude and direction of the vector are equivalent to the angular rotation rate and the axis of rotation respectively. Properties: • *Method*: Selects the method for entering the angular velocity. Common methods for a scalar profile can be used, as well as the **Composite** option, which sets the angular velocity of each component separately. A **Composite** array node is added as a child to this node. • *Dimensions*: The dimensionality of the volume flow rate (read-only). • *Value*: Specifies the x, y, and z values. These values are separated by commas in a text entry. You can also enter an expression, such as `10 * sin(3.14 * $Time)`, directly. The property customizer opens a dialog which contains separate fields for these values. This property is only available when *Method* is set to **Constant**. • *Vector Function*: Activates an object selector dialog from which you can choose a field function. This property is only available when *Method* is set to **Field Function**. • *Function*: Provides a drop-down list from which you can choose a user function. This property is only available when *Method* is set to **User Code**. • *Coordinate System*: The coordinate system in which the position is defined. Select from a drop-down list of available systems.

Coarse Graining Option Specification

This node is added to the tree when a Coarse Grain Particle model is activated. This condition defines the number of individual particles represented in each simulated parcel.

Properties:

Method

Specifies how the number of individual particles per coarse-grained parcel is determined.

Method	Corresponding Value Node
Constant Indicates that the number of individual particles per coarse-grained parcel is specified directly. This is the only option for random injectors. The default value is 1.	**Particle Count** For a table injector, the number of particles represented by a single parcel. It can be a constant or read from the injection table. The default value is 1. This node is not available with the DEM model, since each DEM parcel always represents one particle.

	For a random injector, the number of particles to inject in a single operation. Added to the tree under one of two conditions: • When *Injector Type* is set to **Table Injector** and *Table Type* is set to Parcels Table. • When *Injector Type* is set to **Random Injector** and the Particle Packing Specification is set to **Particle Count**.
Dynamic Indicates that STAR-CCM+ dynamically calculates the number of individual particles per coarse-grained particle. The number depends on other injector conditions and values, such **Flow Rate Specification**, **Particle Size Specification**, and **Mass Flow Rate**. The number need not be an integer, but is never less than 1.	None.

Flow Rate Distribution

This node is added to the tree when the injector is any type other than a point injector. There are three ways to distribute the flow rate. As an example, consider an injector consisting of 20 injection points. Each point injects 10 parcels every time step; that is to say, each injection point has 10 parcel streams. Let the time step be 1 s and the specified flow rate be 1 kg/s.

- If the flow rate distribution is **Per Injector**, the total mass per second is 1 kg.
- If the flow rate distribution is **Per Injection Point**, the total mass per second is 20 kg.
- If the flow rate distribution is **Per Parcel Stream**, the total mass per second is 200 kg.

Properties:

Distribution

Specifies the distribution of the flow rate.

Per Parcel Stream

Indicates that the specified flow rate is applied at each parcel stream of each injection point of this injector.

Per Injection Point

Indicates that the specified flow rate is applied at each injection point of this injector. (Not available for point injectors.)

Per Injector

Indicates that the specified flow rate is the total for all the injection points of this injector.

Flow Rate Specification

This condition specifies how the flow rate of particles at the injector is defined.

Properties:

Method

Selects the method for entering the flow rate.

Method	Corresponding Value Node
Mass Flow Rate Indicates that the rate at which the injector introduces particles is deduced from a mass flow rate. Hence this condition is not available with the Massless Particles model.	**Mass Flow Rate** **Applies to:** any injector. In most cases, specifying mass flow rate means that each parcel in the parcel streams has the same mass (unsteady) or mass flow rate (steady). For the surface injector type, A_f/A_s gives the scale for the mass or mass flow rate of each parcel. A_f is the area of the face through which the parcel enters and A_s is the total area of the surface injector. Properties: The following list applies to all injector types and distribution conditions except for those that are listed in the next list: • *Method*: Selects the method for entering the injector mass flow rate. ◦ Common methods for a scalar profile can be used, as well as **Table (Cone Angle)**, which specifies the table and table column containing volume flow rate as a function of angle. A **Table (Cone Angle)** node is added as a child to this node. ◦ If the flow rate is specified as series of time-dependent values in a table, STAR-CCM+ computes a time-averaged value of the flow rate in a given time step. This time-averaged value depends on the interpolation option that you specify for the table. • *Value*: Specifies the value with a text entry. You can also enter an expression, such as `10 * sin(3.14 * $Time)`, directly. This property is only available when **Method** is set to **Constant**. • *Function*: Provides a drop-down list from which you can choose a user function. This property is only available when **Method** is set to **User Code**. • *Dimensions*: The dimensionality of the mass flow rate (read-only).

	The following list applies to the surface and part injectors, for **Per Injector Point** and **Per Parcel Stream** distributions: • *Method*: Selects the method for entering the injector mass flow rate. Common method for a scalar profile can be used. • *Value*: Specifies the value with a text entry. You can also enter an expression, such as `10 * sin(3.14 * $Time)`, directly. This property is only available when **Method** is set to **Constant**. • *Scalar Function*: Activates an object selector dialog from which you can choose a field function. This property is only available when **Method** is set to **Field Function**. • *Function*: Provides a drop-down list from which you can choose a user function. This property is only available when **Method** is set to **User Code**. • *Dimensions*: The dimensionality of the mass flow rate (read-only).
Volume Flow Rate Indicates that the rate at which the injector introduces particles is deduced from a volume flow rate. Hence this condition is not available with the Massless Particles model.	**Volume Flow Rate** **Applies to:** the surface injector. Specifies volume flow rate means that each parcel in the parcel streams has the same volume (unsteady) or volume flow rate (steady). A_f/A_s gives the scale for the volume or volume flow rate of each parcel. A_f is the area of the face through which the parcel enters, and A_s is the total area of the surface injector. Properties: • *Method*: Selects the method for entering the volume flow rate. ◦ Common methods for a scalar profile can be used, as well as **Table (Cone Angle)**, which specifies the table and table column containing volume flow rate as a function of angle. A **Table (Cone Angle)** node is added as a child to this node. ◦ If the flow rate is specified as series of time-dependent values in a table, STAR-CCM+ computes a time-averaged value of the flow rate in a given time step. This time-averaged value depends on the interpolation option that you specify for the table. • *Value*: Specifies the value with a text entry. You can also enter an expression, such as `10 * sin(3.14 * $Time)`, directly. This property is only available when **Method** is set to **Constant**. • *Dimensions*: The dimensionality of the volume flow rate (read-only).

Particle Flow Rate Indicates that the rate at which the injector introduces particles is specified directly.	**Particle Flow Rate** **Applies to:** the surface injector. Specifies particle flow rate means that each parcel in the parcel streams has the same particle count (unsteady) or particle flow rate (steady). A_f/A_s gives the scale for the mass or mass flow rate of each parcel. A_f is the area of the face through which the parcel enters, and A_s is the total area of the surface injector. Properties: • *Method*: Selects the method for entering the particle flow rate. ◦ Common methods for a scalar profile can be used, as well as **Table (Cone Angle)**, which specifies the table and table column containing volume flow rate as a function of angle. A **Table (Cone Angle)** node is added as a child to this node. ◦ If the flow rate is specified as series of time-dependent values in a table, STAR-CCM+ computes a time-averaged value of the flow rate in a given time step. This time-averaged value depends on the interpolation option that you specify for the table. • *Dimensions*: The dimensionality of the particle flow rate (read-only).
Mass Flux Indicates that the rate at which the injector introduces particles is specified using a mass flux. The final mass flow rate depends on the surface area of the injector part. This method is only available for surface type injectors.	**Mass Flux** **Applies to:** the surface injector. Added to the tree when the injector has surface injector type and the **Mass Flux** option has been chosen for the **Flow Rate Specification**. It is entered as a scalar profile. The actual mass flow rate through each individual face of the injector surface is the product of the face surface area and the specified mass flux.
Volume Flux Indicates that the rate at which particles are introduced is specified using a volume flux. The final volume flow rate depends on the surface area of the injector part. This method is only available for surface type injectors.	**Volume Flux** **Applies to:** the surface injector. It is entered as a scalar profile. The actual volume flow rate through each individual face of the injector surface is the product of the face surface area and the specified volume flux.
Particle Flux Indicates that the rate at which particles are introduced is specified	**Particle Flux** **Applies to:** the surface injector.

using a particle flux. The final particle flow rate depends on the surface area of the injector part. This method is only available for surface type injectors.	It is entered as a scalar profile. The actual particle flow rate through each individual face of the injector surface is the product of the face surface area and the specified particle flux.
Dynamic Indicates that STAR-CCM+ dynamically calculates the rate at which particles are introduced. The particle flow rate depends on the Particle Packing Specification and number of seeds values. This method is only available for random type injectors.	None.

Injection Time Rate Specification

Unlike other Lagrangian particles, DEM particles interact with each other. Hence it is not always possible to inject the required mass simultaneously at the beginning of an iteration. This option allows you to change this default behavior to smooth the injection over the time-step.

Properties:

Time Rate

Selects if the DEM particles are injected simultaneously at the start of the time-step or over a series of sub steps.

Time-step

All particles are injected simultaneously at the start of the iteration. The default setting. Issues can occur where many particles are injected simultaneously

Substep

The particle injection occurs throughout the solution of the time-step. Using this method avoids issues where large numbers of particles are injected simultaneously but can cause the solvers to slow.

Injection Volume Correction Specification

This condition is available for DEM injectors. By default, DEM particles are injected into the simulation domain prior to any fluid being present. Fluid is also initialized at the same time with a volume fraction equal to 1 in the same location as the DEM particles. This inconsistency causes instability in the early stages of a DEM simulation.

When the Injection Volume Correction Specification is activated, STAR-CCM+ accounts for the volume occupied by DEM particles when it initializes the fluid volume fractions. Fluid volume fractions sum to (1 - DEM phase volume). The property is deactivated by default.

Lattice Structure

This node is added to the tree when the injector has lattice injector type. Allows injection of particles into the system with specified, mesh-independent spatial distribution. The main advantage of the lattice injector when compared to the other injector types is the ability to introduce higher particle loading rapidly and make the particle distribution stable with respect to remeshing.

Properties:

Lattice

Selects the distribution type of injected DEM particles.

Simple cubic

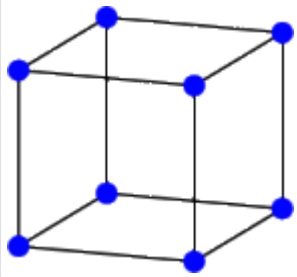
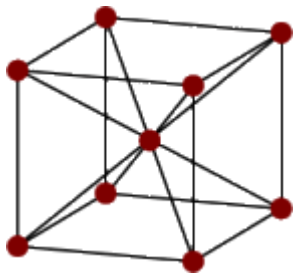
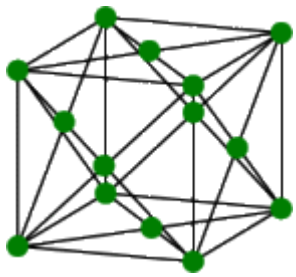
Particle injection points correspond to the corners of a cube.

Body centered cubic

Particle injection points correspond to the corners and centroid of a cube.

Face centered cubic

Particle injection points correspond to the corners and face centers of a cube.

		
Simple cubic	Body centered cubic	Face centered cubic

Orientation Specification

Added to the tree when a non-spherical DEM particle shape is selected. Used to specify the initial orientation of the particle as it is injected.

Properties:

Method

Selects the method for entering the particle orientation.

Method	Corresponding Value Node
Angles	None.

The orientation is expressed as the Euler angles of the principle moments of inertia relative to the global co-ordinate frame.	
Angles, Body Aligned The orientation is expressed as the Euler angles of the axes of the frame used to define the particle relative to the global co-ordinate frame.	None.

Overlap Specification

Added to the tree when DEM particles are being injected with a table injector. Activated to specify that overlapping particles can be injected.

Properties:

Allow Overlap

When **On**, specifies that overlapping particles can be injected.

Particle Packing Specification

This condition specifies the method for defining the quantity of particles injected. Added to the tree when the random injector type is chosen.

Properties:

Method

Selects the method for entering the particle amount.

Method	Corresponding Value Nodes
Porosity Limit This method is the default.	**Porosity Limit** Indicates that the packing of particles is specified by setting the volume fraction taken up by particles. The random injector porosity is defined as: $\text{Porosity} = \frac{\text{Total Volume of Injected Partices}}{\text{Volume of Regions Assigned to the Injector}}$ The porosity limit defines the maximum packing limit for particles that are introduced into the injected region. As the injector injects particles that are composed of spheres, the porosity is always less than 1. For example, when using single spheres the maximum value is 0.74. STAR-CCM+ stops injecting particles if the limit is exceeded, or if there is no space for further particles in the region. Properties: • *Value*: Specifies the porosity limit for the random injector, which acts as an injection stopping condition. The porosity is the ratio of the total injected particles volume to the region volume.

	Injection Cycle Limit This value sets the number of times the injector can try to insert the **Number of Seeds**. This value and the **Porosity Limit** work together to limit the random injector. **Number of Seeds** The number of seeds is the number of particles the random injector tries to inject at the same time. It does not define the total number of particles injected—the **Particle Count** sets that number. For most cases, set this value as one tenth of the total number of particles that can be packed inside the volume of the injector region. Calculate the optimal number of seeds for a case as follows: • Calculate the volume V_r of the injector region. • Calculate the volume V_p of a single particle. • Calculate the number of seeds as the number of particles that would fill 0.1 of the region volume as $0.1\,\frac{V_r}{V_p}$.
Particle Count	**Particle Count** Indicates that the quantity of particles is specified directly. For a table injector, the number of particles represented by a single parcel. It can be a constant or read from the injection table. The default value is 1. This node is not available with the DEM model, since each DEM parcel always represents one particle. For a random injector, the number of particles to inject in a single operation.
Maximum Packing	**Number of Particles** Indicates that a specified number of particles is grown from small starting size until maximum packing density is reached.

Particle Size Specification

This condition specifies the method for defining the size of particles from the injector.

Properties:

Method

Selects the method for entering the particle magnitude.

Method	Corresponding Value Node
Particle Size	**Particle Diameter** **Applies to:** any injector. Specifies the particle size directly.

Properties:

- *Method*: The standard methods for a scalar profile are available for this value, along with more methods shown in the properties list below.
  - **Log-Normal**: Specifies a range of particle diameters using a Log-normal particle size distribution. Log-Normal settings are added to these properties.
  - **Normal**: Specifies a range of particle diameters using a normal particle size distribution. Normal settings are added to these properties.
  - **Rosin-Rammler**: Specifies a range of particle diameters using a Rosin-Rammler particle size distribution. Rosin-Rammler settings are added to these properties.
  - **Table (Cone Angle)**: Specifies the table and table column containing particle diameter as a function of angle. A Table (Cone Angle) node is added as a child to this node.
  - **Table (particle size CDF)**: Specifies a range of particle diameters using a tabulated particle size distribution. A Table (particle size CDF) node is added as a child to this node.
- *Value*: Specifies the value with a text entry. You can also enter an expression, such as `10 * sin(3.14 * $Time)`, directly. This property is only available when **Method** is set to **Constant**.
- *Scalar Function*: Activates an object selector dialog from which you can choose a field function. This property is only available when **Method** is set to **Field Function**.
- *Function*: Provides a drop-down list from which you can choose a user function. This property is only available when **Method** is set to **User Code**.
- *Dimensions*: The dimensionality of the particle diameter (read-only).

Cylinder Height

Applies to: the Cylindrical Particles model.

The absolute value of the height and radius are determined from the particle mass and density.

Cylinder Radius

Applies to: the Cylindrical Particles model.

Properties:

- *Method*: Selects the method for specifying the dimensions of cylindrical particles. Common methods for a scalar profile can be used, as well as the additional options shown below.
  - **Log-Normal**: Specifies a range of particle dimensions using a Log-normal particle size distribution. Log-Normal settings are added to these properties.
  - **Normal**: Specifies a range of particle dimensions using a normal particle size distribution. Normal settings are added to these properties.

	◦ **Rosin-Rammler**: Specifies a range of particle dimensions using a Rosin-Rammler particle size distribution. Rosin-Rammler settings are added to these properties. ◦ **Table (particle size CDF)**: Specifies a range of particle dimensions using a tabulated particle size distribution. A Table (particle size CDF) node is added as a child to this node. ◦ **Table (particle size CDF, time)**: Specifies a range of particle dimensions using a tabulated particle size distribution. A Table (particle size CDF, time) node is added as a child to this node. • *Value*: Specifies the value with a text entry. You can also enter an expression, such as `10 * sin(3.14 * $Time)`, directly. This property is only available when **Method** is set to **Constant**. • *Scalar Function*: Activates an object selector dialog from which you can choose a field function. This property is only available when **Method** is set to **Field Function**. • *Function*: Provides a drop-down list from which you can choose a user function. This property is only available when **Method** is set to **User Code**. • *Dimensions*: The dimensionality of the particle height and radius (read-only).
Particle Mass	**Particle Mass** Indicates that the particle size is deduced from its mass. Properties: • *Method*: Selects the method for specifying the injector particle mass. Common methods for a scalar profile can be used, as well as the additional options shown below. ◦ **Log-Normal**: Specifies a range of particle masses using a Log-normal particle size distribution. Log-Normal settings are added to these properties. ◦ **Normal**: Specifies a range of particle masses using a normal particle size distribution. Normal settings are added to these properties. ◦ **Rosin-Rammler**: Specifies a range of particle masses using a Rosin-Rammler particle size distribution. Rosin-Rammler settings are added to these properties. ◦ **Table (Cone Angle)**: Specifies the table and table column containing particle mass as a function of angle. A Table (Cone Angle) node is added as a child to this node. ◦ **Table (particle size CDF)**: Specifies a range of particle masses using a tabulated particle size distribution. A Table (particle size CDF) node is added as a child to this node. • *Value*: Specifies the value with a text entry. You can also enter an expression, such as `10 * sin(3.14 * $Time)`, directly. This property is only available when **Method** is set to **Constant**. • *Scalar Function*: Activates an object selector dialog from which you can choose a field function. This property is only available when **Method** is set to **Field Function**.

- *Function*: Provides a drop-down list from which you can choose a user function. This property is only available when **Method** is set to **User Code**.
- *Dimensions*: The dimensionality of the particle diameter (read-only).

Cylinder Height/Radius Ratio

Applies to: the Cylindrical Particles model.

The absolute value of the height and radius are determined from the particle mass and density.

Species Specification

Added to the tree when the Lagrangian Species model is chosen.

Similar to the species specification for a continuum, this object gives you a choice of how to specify the proportions of components in the injected particle:

Properties:

Method

Selects the method for entering the fraction of a component in a particle.

Method	Corresponding Value Node
Mole Fraction The ratio of the molar amount of a component over the molar amount of the particle.	**Species Mole Fraction** The mole fraction of a component in the particle is the ratio of the molar amount of the component over the total molar amount of the particle. This object functions in a manner similar to the species mole fraction of a continuum. Properties: • *Method*: Selects the method for specifying component mole fractions through the injector. ◦ **Constant**: Set the mole fraction of each component in the order they appear in the mixture. A **Constant** array node is added as a child to this node. ◦ **Composite**: Set the mole fraction of each component separately. A Composite array node is added as a child to this node. If you use a table injector, individual species mole fractions can be initialized by an injection table. If you use a hollow or solid cone injector, individual species mole fractions can be initialized by a cone angle table. ◦ **Injection Table**: Set the mole fraction of each component according to an injection table. The table specifies the location of injection points, the time of injection, and the mole fraction of each component separately. Used only with table injectors. • *Dimensions*: The dimensions of the mole fraction (read-only).

Mass Fraction The ratio of the mass of a component over the total mass of the particle.	**Species Mass Fraction** The mass fraction of a component in the particle is the ratio of its mass over the total mass of the particle. This object functions in a manner similar to the species mass fraction of a continuum. Properties: • *Method*: Selects the method for specifying component mass fractions through the injector. ◦ **Constant**: Set the mass fraction of each component in the order they appear in the mixture. A **Constant** array node is added as a child to this node. ◦ **Composite**: Set the mass fraction of each component separately. A Composite array node is added as a child to this node. If you use a table injector, individual species mass fractions can be initialized by an injection table. If you use a hollow or solid cone injector, individual species mass fractions can be initialized by a cone angle table. ◦ **Injection Table**: Set the mass fraction of each component according to an injection table. The table specifies the location of injection points, the time of injection, and the mass fraction of each component separately. Used only with table injectors. • *Dimensions*: The dimensions of the mass fraction (read-only).

Time Randomization

This object is available for non-DEM injectors in unsteady simulations. When activated, it causes the model to choose parcel injection time randomly within the time-step, rather than the default injection of all parcels at the start. This randomization can, for example, improve the appearance of animations where otherwise groups of parcels would appear to “march forward” together.

Properties:

Random

When **On**, injection time for parcels is chosen randomly within the time-step. When **Off**, all parcels are injected at the start of the time-step.

Velocity Specification

The initial velocity of injected particles can be entered as a vector, or as a magnitude and direction, in which case the direction would be specified as a direction vector. You can also control the reference frame.

Properties:

Method

Specifies the method for entering velocity.

Method	Corresponding Value Node
Components Enter the velocity as a vector with magnitude.	**Velocity** **Applies to:** any injector. This value is entered as a vector profile. If the particle injection velocity is to correspond to the fluid velocity at the injection point, take the following steps: 1. In the *Properties* window for the **Values** > **Velocity** node, select **Field Function** for the *Method* property. The *Vector Function* property becomes available. 2. Select **Velocity** for the *Vector Function* property.
Magnitude + Direction Enter a direction vector and a magnitude.	**Injection Direction** **Applies to:** any injector. Entered as a vector profile. **Velocity Magnitude** **Applies to:** any injector. Entered as a scalar profile.

Reference Frame

The injection velocity can be specified relative to the mesh or in the absolute frame.

Relative to Mesh

The velocity is relative to the mesh.

Absolute

The velocity is absolute.

See Also:

The Cylindrical Particles Model

Injector Values Reference

This section describes the available injector values.

Depending on how the injector conditions are set, you can define injections using the following values:

- Axis

- Cone Randomization
- Injector Diameter
- Injector Upstream Pressure
- Inner Cone Angle
- Mean Injection Velocity
- Nozzle Discharge Coefficient
- Nozzle Diameter
- Nozzle Form Loss Coefficient
- Nozzle Length
- Nozzle Turbulence Dissipation Rate Coefficient
- Outer Cone Angle
- Parcel Streams
- Particle Charge
- Particle Count
- Particle Orientation
- Particle Temperature
- Passive Scalar
- Point Inclusion Probability
- Position
- Sheet Breakup Constant
- Spacing
- Swirl
- Swirl Angle
- Transfer Criterion
- Volume Flux

Axis

Applies to: hollow cone, solid cone, or nozzle injectors.

This value defines the central axis of the cone within which parcels are injected.

Properties:

Direction

The direction of the axis. Specified as a vector in the chosen coordinate system.

Origin

The origin of the axis (and therefore the point from which parcels are injected). Specified as a position vector in the chosen coordinate system.

Coordinate System

The coordinate system in which the axis is defined. Select from a drop-down list of available systems.

Cone Randomization

Applies to: hollow cone or nozzle injectors in unsteady simulations.

This determines whether injection points change position randomly from time-step to time-step, or remain in fixed positions.

Properties:

Rerandomize

When **On**, specifies that the injection points are randomized again for each time-step. When **Off**, re-uses the initial randomization. **Off** is the default.

Value

Proportion of the distance between initial positions over which injection points can vary between one time-step and the next, expressed as a value between 0 and 1. Effective only when *Rerandomize* is activated. The default value is 1.

Injector Diameter

Applies to: the hollow cone injector.

This value defines the diameter of the injector hole through which the droplet fluid flows.

Injector Upstream Pressure

Applies to: the nozzle injector.

This value sets the upstream pressure, which is used to calculate Δp in Eqn. (3054). Upstream pressure is the pressure head available to force liquid through the pressure swirl nozzle, and is used to predict the liquid sheet exit velocity.

Inner Cone Angle

Applies to: the hollow cone or nozzle injectors.

This value defines the internal angle of the cone about the central axis. A value of zero results in a solid cone formation.

Mean Injection Velocity

Applies to: the Huh model together with the nozzle injector, in unsteady simulations.

Specifies U in Eqn. (3071). The mean value is 1.0 m/s.

Nozzle Discharge Coefficient

Applies to: the Huh model together with the nozzle injector, in unsteady simulations.

A dimensionless proportionality constant for the rate of volume flow through the nozzle. Specifies c_d in Eqn. (3071). The default value is 0.6.

Nozzle Diameter

Applies to: the nozzle injector, in unsteady simulations.

The diameter of the circle through which parcels are injected. Specifies D in Eqn. (3071). The default value is 3.0E-4.

Nozzle Form Loss Coefficient

Applies to: the Huh model together with the nozzle injector, in unsteady simulations.

A dimensionless proportionality constant for the pressure loss associated with the sharpness of the nozzle entrance corner. Specifies K_c in Eqn. (3071). The default value is 0.45.

Nozzle Length

Applies to: the Huh model together with the nozzle injector, in unsteady simulations.

The distance between the point where primary atomization begins and the point where secondary breakup begins. Specifies L in Eqn. (3071). The default value is 0.0012 m.

Nozzle Turbulence Dissipation Rate Coefficient

Applies to: the Huh model together with the nozzle injector, in unsteady simulations.

Specifies ϵ_a in Eqn. (3071). The default value is 0.5.

Outer Cone Angle

Applies to: the hollow cone, solid cone, or nozzle injectors.

This value defines the outer angle of the cone about the central axis, within which parcels are injected.

Parcel Streams

Applies to: the film stripping injector.

This value sets the number of parcel streams per injection point (and per time-step in unsteady simulations). This quantity has no influence on the mass or volume flow rate from the injector, but instead can influence accuracy, depending on other selections. Specifically:

- With the Turbulent Dispersion model, increasing the number of parcel streams increases accuracy by creating a larger ensemble of tracks from each injection point, each representing a different realization of the particle trajectory. Typically 10–100 parcel streams are specified for this purpose.
- With particle size distributions, increasing the number of parcel streams increases the resolution of the particle size distribution. Typically 5-20 parcel streams are specified for this purpose.

The combined effect of both of the above, is multiplicative. If neither is relevant, use the default number of parcel streams (1). CPU time is directly proportional to this quantity.

Properties:

Value

- In **steady** simulations, the number of parcel tracks originating from each injection point.
- In **unsteady** simulations, the number of parcels introduced at each injection point at the start of each time-step.

The number of parcel streams can be a constant or derived from an expression. Expressions allow the value to vary with time or other conditions. Field functions are allowed.

All values, whether from expressions or constants, are rounded to the nearest integer. Values lower than 1 generate an error.

Particle Charge

Applies to: the Coulomb Force model.

This value is entered as a scalar profile.

Particle Count

Applies to: the table injectors or random injector.

Added to the tree under one of two conditions:

- When *Injector Type* is set to **Table Injector** and *Table Type* is set to **Parcels Table**.
- When *Injector Type* is set to **Random Injector** and the Particle Packing Specification is set to **Particle Count**.

Particle Orientation

Applies to: non-spherical DEM particle shapes.

Used to specify the initial orientation of the particle as it is injected.

Properties:

Method

Selects the method for entering the particle orientation. Common methods for a scalar profile can be used.

Value

Specifies the x, y, and z values. These values are separated by commas in a text entry. You can also enter an expression, such as `10 * sin(3.14 * $Time)`, directly. The property customizer opens a dialog which contains separate fields for these values. This property is only available when **Method** is set to **Constant**.

Angle Function

Activates an object selector dialog from which you can choose a field function. This property is only available when **Method** is set to **Field Function**.

Dimensions

The dimensionality of the particle orientation (read-only).

Particle Temperature

Applies to: the Lagrangian Energy model.

This value is entered as a scalar profile.

Passive Scalar

Applies to: the Passive Scalar model.

Properties:

Method

Selects the method used for setting the value of the scalar on the particle as the particle enters the simulation.

Constant

Set the values of all individual scalars as a vector of constants.

Composite

Set the value of each passive scalar separately (constant, field function, table or user code). If you use a table injector, individual passive scalars can be initialized by an injection table.

Injection Table

Set the value of the passive scalar for each component according to an injection table. The table specifies the location of injection points, the time of injection, and the passive scalar value of each component separately. Used only with table injectors.

Dimensions

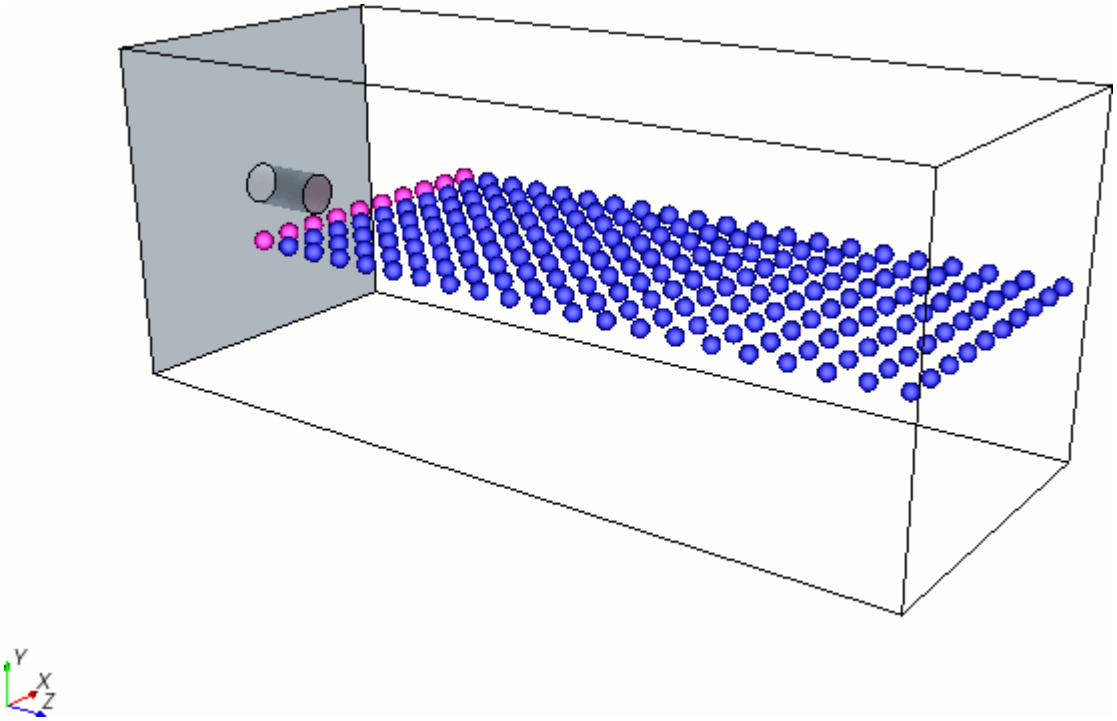
The dimensionality of the value, *Mass/Volume-Time* (read-only). The passive scalar itself is always dimensionless, but its source term has units referring to the density of the receiving phase over time.

Point Inclusion Probability

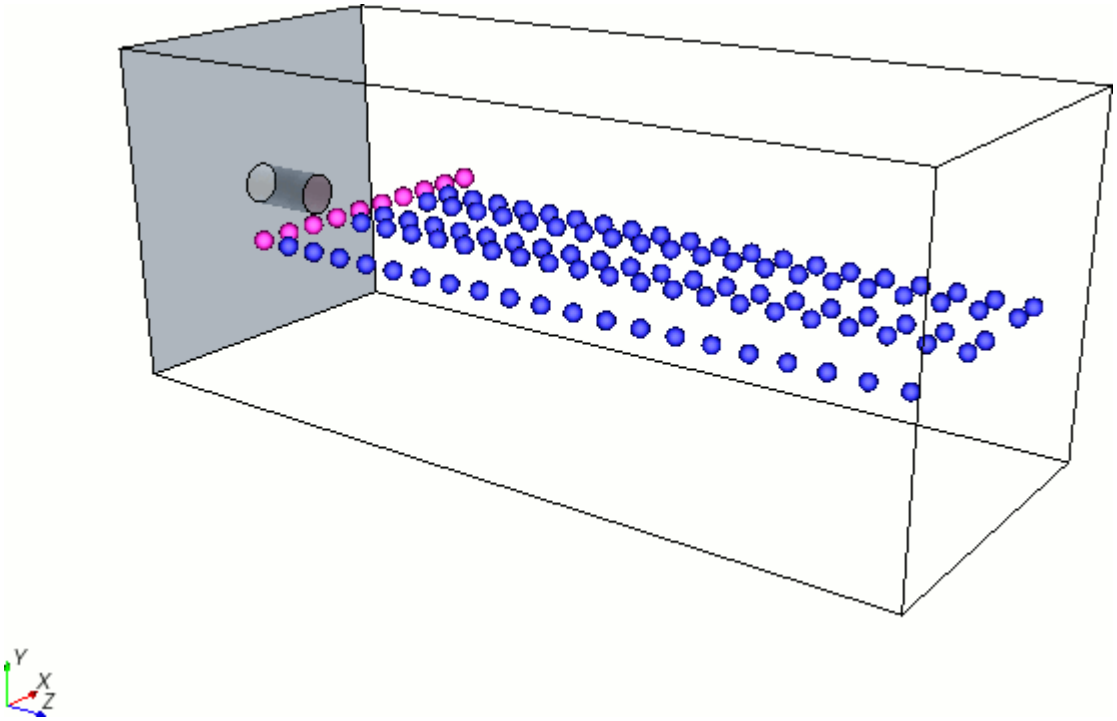
Applies to: the part injector.

This node sets the probability of a point being included in the set of points from which particles are injected. In the example that is shown below, a line probe derived part with a resolution of 11 is used as the input part for the injector. This configuration means that 10 points are available from which to inject particles. In the first screenshot, the point inclusion probability is set to 1.0, which means that the injector uses all points that are extracted from the part. In the second screenshot, the point inclusion probability is set to 0.4, meaning that only a subset of the points is used. In the third screenshot, the *Re-randomize* property is activated, regenerating the injection points at the start of each time-step.

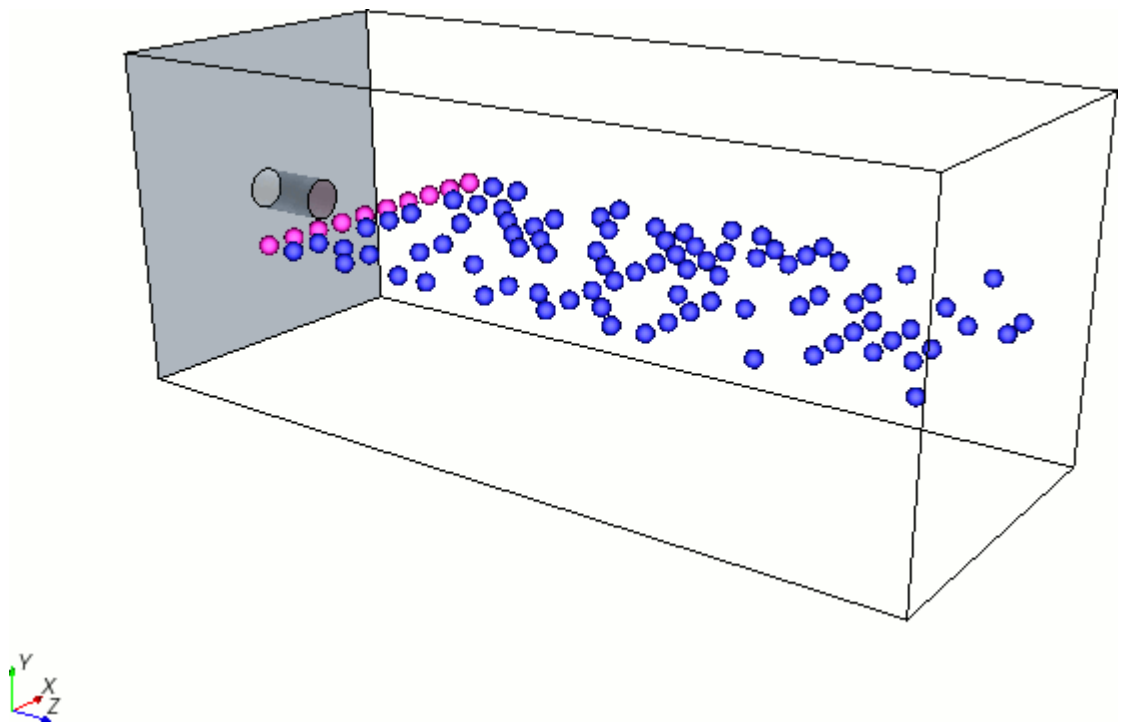
Point Inclusion Probability = 1.0



Point Inclusion Probability = 0.4



Point Inclusion Probability = 0.4, with re-randomizing option activated



Properties:

Value

The probability that a point in the injector is used as an injection point. The probability is applied to each part point.

Re-randomize

When **On**, regenerates the injection points at the start of each time-step. When **Off**, leaves injection points fixed after the start of the simulation (as in a steady analysis). Available for transient analyses only.

Polyhedron Elongation Factor

Applies to: the Polyhedral Particles model.

Properties:

Method

Selects the method for specifying the dimensions of polyhedral particles. Common methods for a scalar profile can be used.

Value

An ordered triplet of multipliers for the length of the polyhedron along three axes of the particle's principal moment of inertia, enabling the injection of polyhedra of varying proportion, based on the template described in the *Part* property of the Polyhedral Particle model. The default is `[1.0, 1.0, 1.0]` for no change. Use values greater than one to elongate along an axis, or less than one to shrink.

Dimensions

The units for the size of the template particle. Read-only.

Coordinate System

The coordinate system in which the template particle is defined. Select from a drop-down list of available systems.

Position

Applies to: the point injector.

Properties:

Coordinate System

The coordinate system in which the position is defined. Select from a drop-down list of available systems.

Value

The value and units of the position. Clicking the customizer executes the define value action. An expression is not accepted for this entry.

Sheet Breakup Constant

Applies to: the nozzle injector with the LISA model.

This value is an empirical constant, ξ in Eqn. (3061), that calibrates the ratio of the sheet surface disturbance at breakup to its initial value in the dispersion relation of the LISA model. It is set to 12 by default.

Spacing

Applies to: the lattice injector.

This value allows you to control the dimensions of the bounding box in the x, y, and z directions. Clicking the customizer executes the define value action. An expression is not accepted for this entry.

Species Mass Fraction

Applies to: the Species model.

This value allows you to control the mass fractions of multi-component particles and droplets using common properties of a scalar profile.

Swirl

Applies to: the hollow cone injector.

This value defines additional angular velocity about the axis, added to existing velocity. It is set to zero by default.

Swirl Angle

Applies to: the nozzle injector with the LISA model.

This value defines the angular/azimuthal velocity component about the injector axis of the atomized droplets. It is set to zero by default.

Transfer Criterion

Applies to: the parcel transfer injector.

The criterion is entered as an expression in the **Custom Editor** ( ).

If the transfer criterion for a parcel in the source phase evaluates as non-zero, the parcel is transferred to the destination phase.

Volume Flux

Applies to: the surface injector.

Added to the tree when **Volume Flux** is chosen as the Flow Rate Specification. It is entered as a scalar profile. The actual volume flow rate through each individual face of the injector surface is the product of the face surface area and the specified volume flux.

See Also:

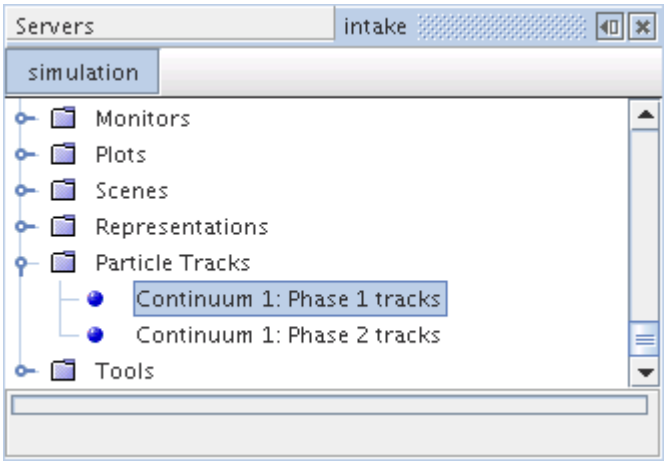
The Cylindrical Particles Model

Working with Particle Tracks

A track records the history of a parcel.

This history is the locus of parcel positions which can be visualized much like a streamline. STAR-CCM+ allows other data to be recorded too, such as particle velocity and temperature, which can be used for subsequent visualization and analysis. Because there is the potential to generate a large amount of data, tracks can be recorded in a dedicated file under your control.

The particle track represents the track of one Lagrangian phase that is taken from a track file. A node of the particle track (named **Continuum 1: Phase 1 tracks** in the following screenshot) has its own properties.



This object is created automatically when you create a track file, and cannot be deleted directly. It is deleted only when you delete the track file that is associated with it. You can, however, rename this object.

To record a snapshot of the complete simulation at a given time, use a Simulation History File.

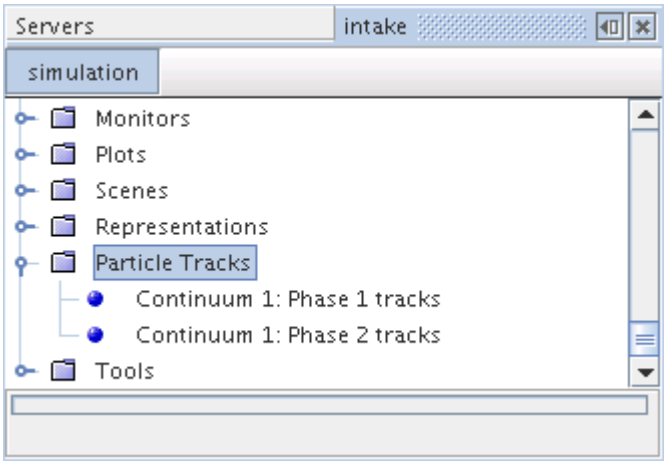
Particle Track Properties

Track File	The track file from which this part is derived (read-only).
Phase Id	The ID number of the Lagrangian phase with which the track file is associated (read-only).
Phase Name	The names of the continuum and Lagrangian phase with which the track file is associated when created (read-only).

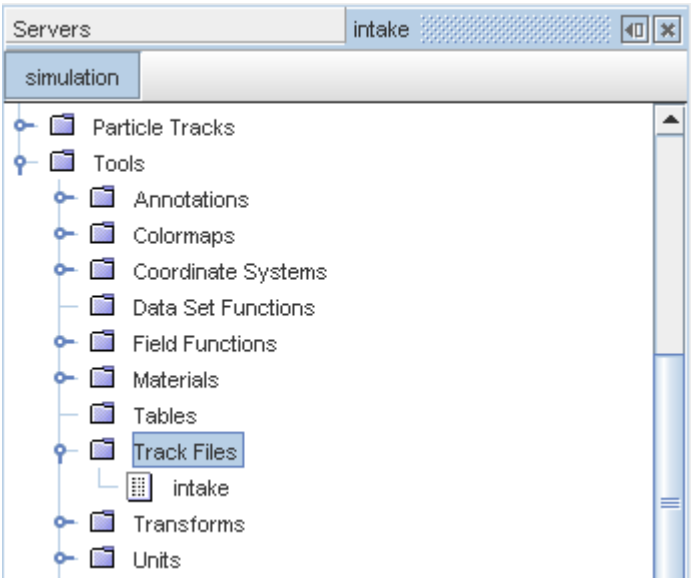
What Is the Particle Tracks Manager?

The **Particle Tracks** manager node contains all of the particle track nodes in the object tree.

The particle track manager can hold any number of particle tracks up to the memory limits of the computer.



Particle tracks get their data from track files. The **Track Files** node is the manager object for working with track files in the simulation tree. It is added to the tree once you select the Lagrangian Multiphase model. The track files manager can hold any number of track files up to the memory limits of the computer.



Contents:

Modeling with a Track File
Creating a Track File
Reading a Track File
Modeling with Boundary Sampling
Mapping Cell and Boundary Field Functions to Parcels
Using Particle Tracks in Analysis
Example of Using Particle Track Data
Animating Particle Tracks

Track Files for Storing Particle Data

In STAR-CCM+, you must select the Track File model before particle track data is stored during a simulation. Selecting this model adds the Track File solver and the Track Files manager to the simulation tree.

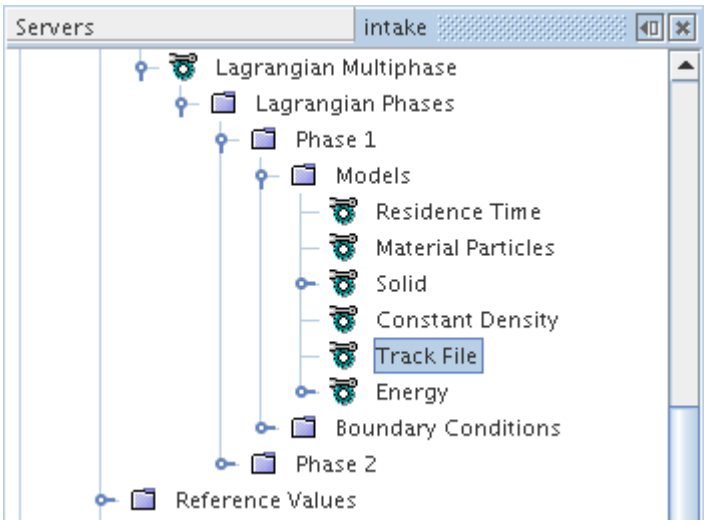
These related objects are described in this section:

- Track file model, located within the phase models for a Lagrangian phase
- Track file solver, which appears when the Track File model is active
- Track files manager, which holds track file objects

Track File Model

The **Track File** node, which has properties, represents the Track File model. This model indicates that the tracks of parcels are recorded in a track file and, if so, what variables are recorded.

A temporary track file only is created while the simulation runs. It is essential to save the simulation so that this temporary track file is moved to a final file having the same name as the simulation. Only the final file can be brought into STAR-CCM+ for analysis, as discussed in section, Creating a Track File.



Every time a track file entry is written, the values of all the user-selected variables are written, together with the following:

- **Parcel Centroid**, the instantaneous parcel centroid.
- **Parcel Index**.
- **Time** (unsteady) or **Particle Residence Time** (steady).
- **Particle Count** (unsteady) or **Particle Flow Rate** (steady).

Hence, you do not need to select the field functions corresponding to these variables. A parcel generates a track file entry at the following events:

- Injection.
- Crossing an internal face.
- Interacting with a boundary face.

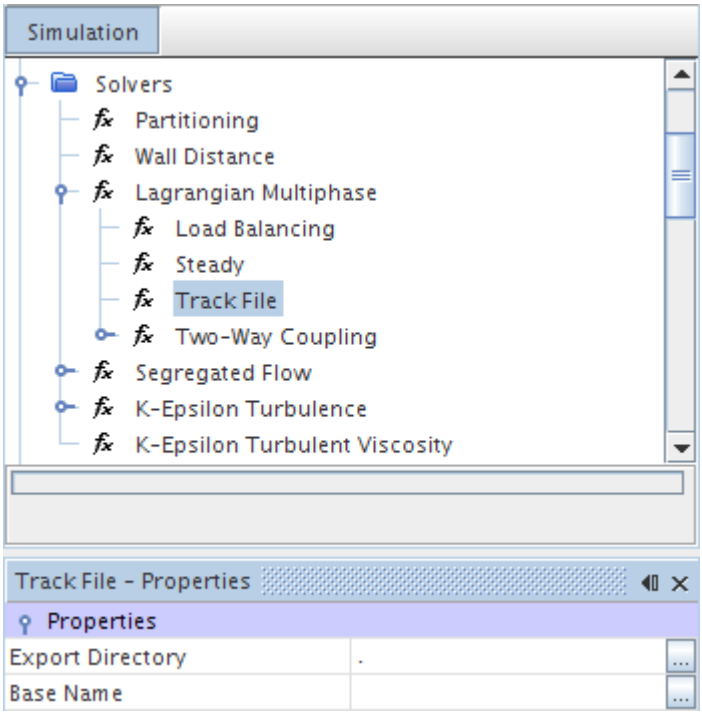
Track File Model Properties

Scalars	Selects a set of scalar field functions, with a Select Objects dialog, for recording on the track file. Three scalars that are automatically selected, **Parcel Index**, **Time** (or **Particle Residence Time** in steady simulations), and **Particle Count** (or **Particle Flow Rate** in steady simulations). They appear in the Selected list in grayed-out text, since they cannot be removed from the selector.
Vectors	Selects a set of vector field functions, with a Select Objects dialog, for recording on the track file.

	One vector that is automatically selected, **Parcel Centroid** appears in the Selected list in grayed-out text, since it cannot be removed from the selector.

Track File Solver

The Track File solver is activated when the Track File model is chosen. It controls the name and location of the track file.



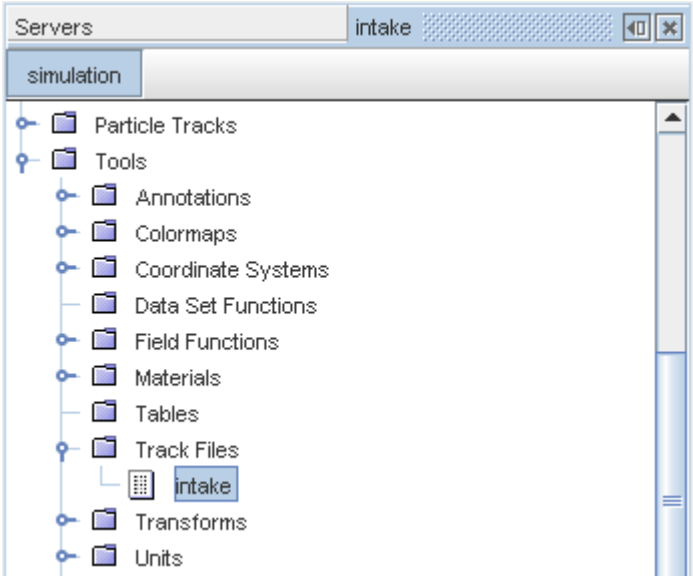
Track File Solver Properties

Export Directory	The directory to which the track file is exported. The default value is "`.`" and means the same directory as the `.sim` file.
Base Name	The name of the track file, without the `.trk` extension. The default is empty, in which case the track file has the same name as the `.sim` file.

Track Files Manager

The track file records the tracks of parcels when the Track File model is active for the Lagrangian phase.

STAR-CCM+ uses an external file to record particle track data because there is the potential to generate—and regenerate—a large amount of data during the simulation. This data is saved for subsequent visualization and analysis. It is represented in the object tree by a track file node (**intake** in the following screenshot), which has properties and a pop-up menu and can be renamed or deleted, located under the **Track Files** manager node.



Track Files Manager Node Properties

Auto-Load	Controls whether or not the track file manager automatically reloads the track file when the file is written to by the track file solver.	
	Activated	Automatically reload the track file.
	Deactivated	Do not automatically reload. This is the default.

Track Files Manager Pop-Up Menu

Track File	Executes the read track file action, and so adds a track file to the **Track Files** node.

Track File Properties

Path	The location of the external data source of the track file. This path is relative to the location of the simulation file. Edit to change to a different track file.	
Parallel Processing	Controls whether the track file data is shared among all the server processes. In the case of large files, this sharing reduces the memory burden on any individual process.	
	Activated	The track file data is shared by all processes.
	Deactivated	The track file data is stored only on the master process. (The default.)

Track File Pop-Up Menu

Reload File...	Rereads the track file data from the external file.

Creating a Track File

This section describes how to create a track file.

To create an external track file (`.trk`), do the following:

1. Under **Lagrangian Multiphase** > **Lagrangian Phases** > **Phase n** > **Models**, select the **Track File** option from the *Track Sampling* group.
2. Select **Solvers** > **Lagrangian Multiphase** > **Track File** and set the **Export Directory** and **Base Name** properties for the location and name of the track file.
3. Step the solution at least one iteration or time-step. Stepping creates a temporary track file. The extension of this file is the first available in the series `.trk.tmp`, `.trk.tmp2`, `.trk.tmp3`, and so on.
4. Save the simulation file. This action replaces the temporary file with a permanent track file (`.trk`).

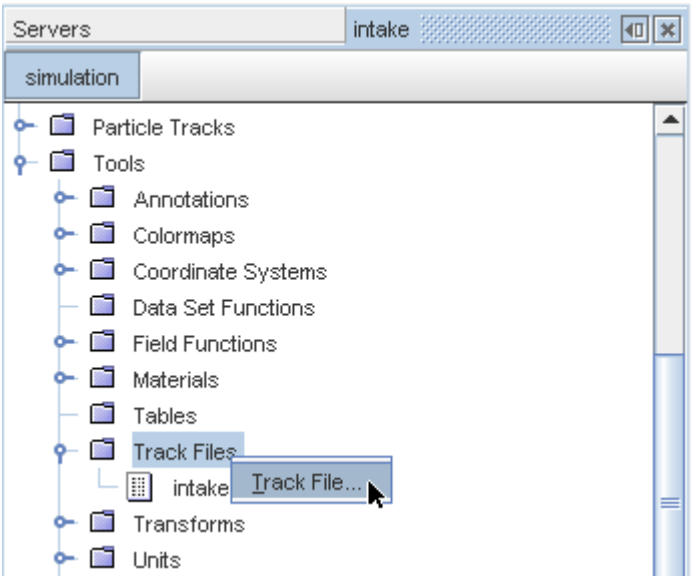
By default, the external track file has the same base file name as the simulation file. If this name is already in use, for example by another server, then the base file name is appended with a unique number. The format of the track file is unique to STAR-CCM+.

Reading a Track File

This section describes how to read a track file.

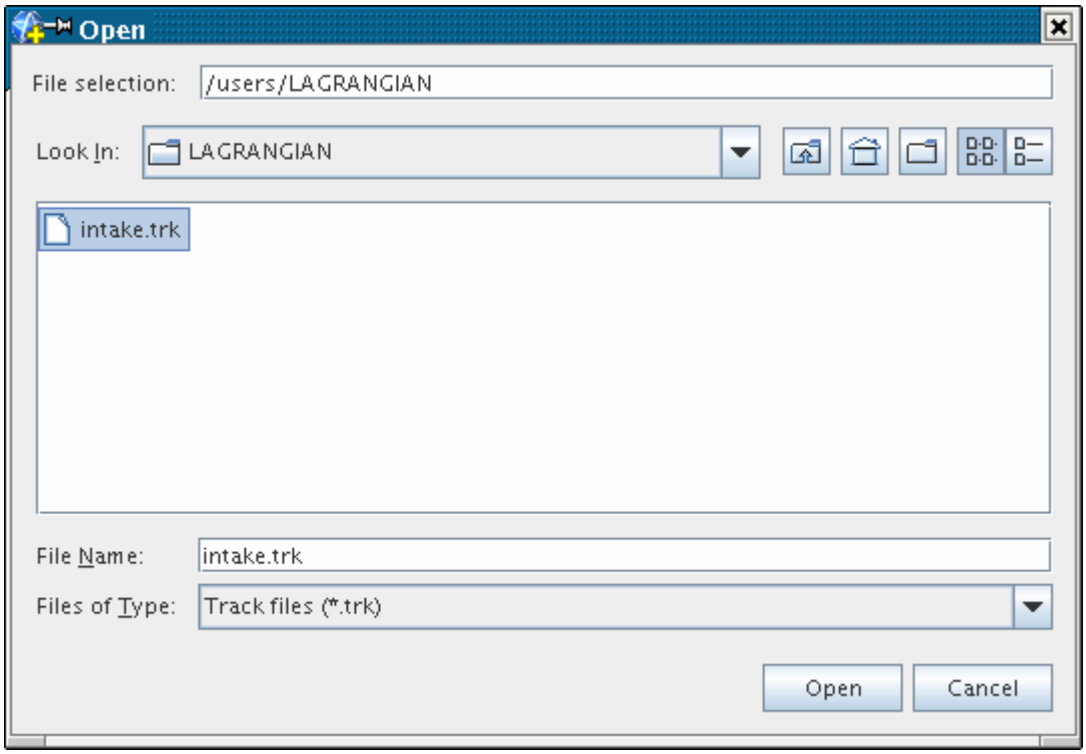
To read a saved track file:

1. Right-click the **Track Files** manager node and select **Track File...**.



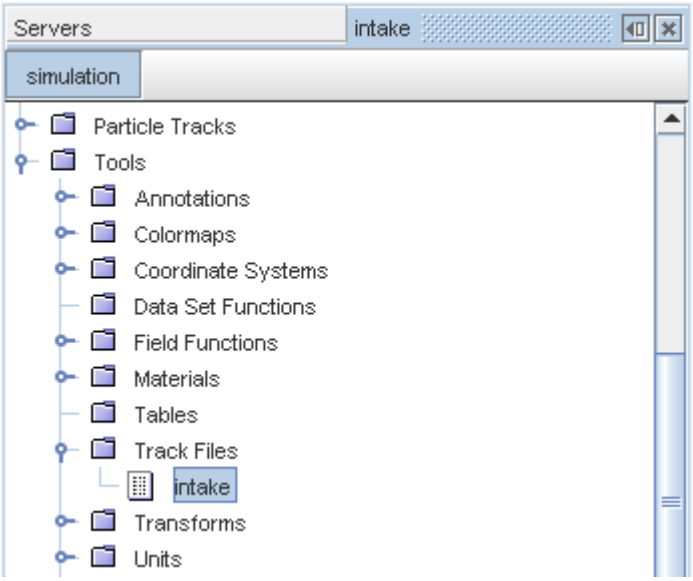
2. Select an available track file from the *Open* dialog, then click **Open**.

   This action adds a track file to the object tree using the data from the specified file.



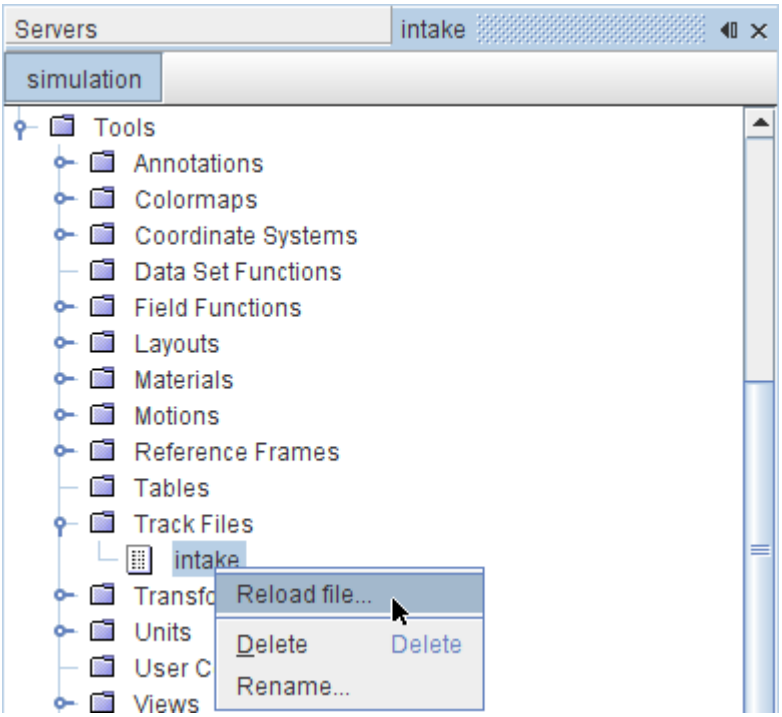
The path to the track file is relative to the location of the simulation file.

The new track file appears in the **Track Files** node. It is initially assigned a name corresponding to the file name (without the `.trk` extension), but can be renamed.



If the external file has been modified since you read it into STAR-CCM+, it is easy to update the track file object:

3. Simply right-click the node and select **Reload file...**.

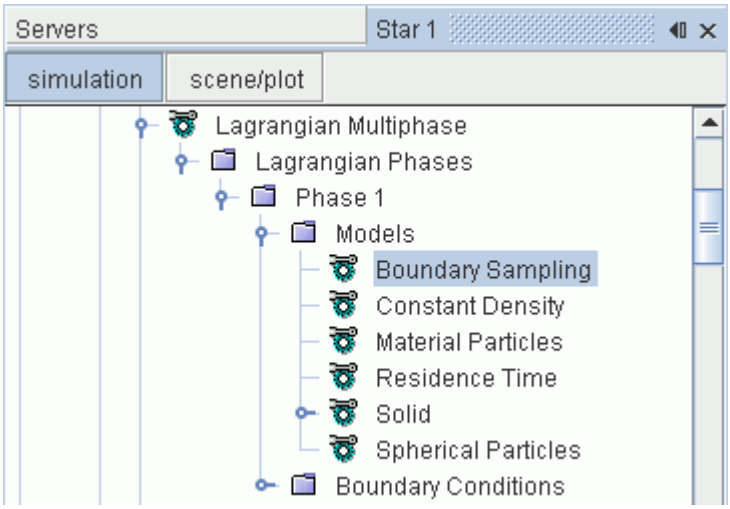


Modeling with Boundary Sampling

This section describes the Boundary Sampling model and how to select it.

What Is the Boundary Sampling Model?

The **Boundary Sampling** node, which has properties, represents the Boundary Sampling model. This model records parcel states when they impact particular boundaries.



Every time a parcel impacts one of the designated boundaries, the values of all the user-selected variables are stored, together with the following:

- **Parcel Centroid**, the instantaneous parcel centroid.
- **Parcel Index**.
- **Time** (unsteady) or **Particle Residence Time** (steady).
- **Particle Count** (unsteady) or **Particle Flow Rate** (steady).

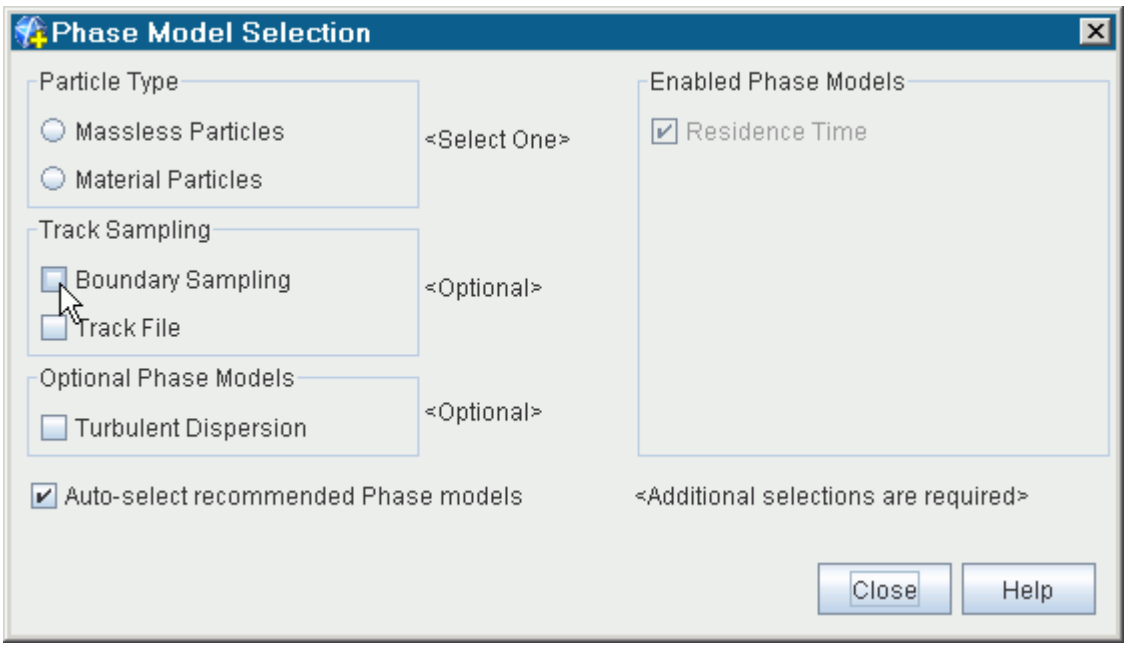
The Boundary Sampling model differs from the Track File model in that it does not have a file that is associated with it. Sampled quantities are stored in memory, and are accessed through particle track parts that are added to the **Particle Tracks** folder when boundaries are added to the Boundaries property of the model node. One particle track part is added for each boundary that is selected to participate in sampling.

Boundary Sampling Model Properties

Boundaries	The set of boundaries for which particle sampling is active.
Scalars	Selects a set of scalar field functions, with a Select Objects dialog, for recording on the boundary. Three scalars that are automatically selected, **Parcel Index**, **Time** (or **Particle Residence Time** in steady simulations), and **Particle Count** (or **Particle Flow Rate** in steady simulations) appear in the Selected list. They are in grayed-out text, since they cannot be removed from the selector.
Vectors	Selects a set of vector field functions, with a Select Objects dialog, for recording on the boundary. One vector that is automatically selected, **Parcel Centroid** appears in the Selected list in grayed-out text, since it cannot be removed from the selector.

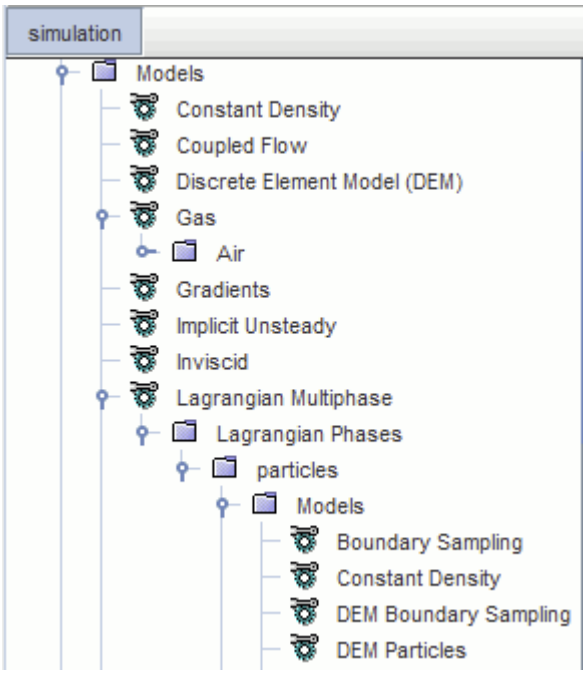
Selecting the Boundary Sampling Model

Select the **Boundary Sampling** option from the *Track Sampling* group, as shown below.



Selecting the DEM Boundary Sampling Model

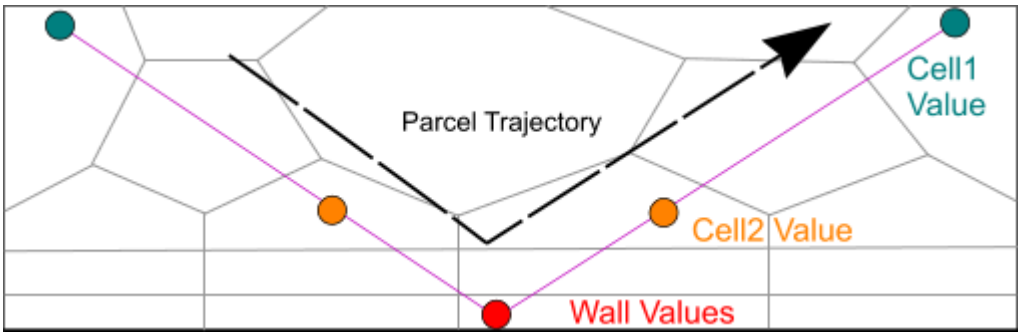
If you are using the Discrete Element Method (DEM) and the Boundary Sampling Model, the DEM Boundary Sampling Model becomes available. This model makes the **Impact Velocity** field function available and tracks particle velocity at the initial impact point for the selected surfaces.



Mapping Cell and Boundary Field Functions to Parcels

Both cell and boundary field function data can be mapped to parcels any time a field function is required on parcels.

An example is shown below.



Track files and boundary sampling facilitate recording field functions of cells and incident boundaries in particle tracks. However, field function mapping can also be used:

- to evaluate user-defined source terms,
- to evaluate composite boundary interaction,
- when displaying scenes or evaluating reports, and so on.

Mapping Cell Data to Parcels

When a cell field function is mapped to parcels, each parcel gets the value from its host cell.

When a vertex field function is mapped to parcels, each parcel gets the arithmetic average of the vertex values of the vertices belonging to its host cell.

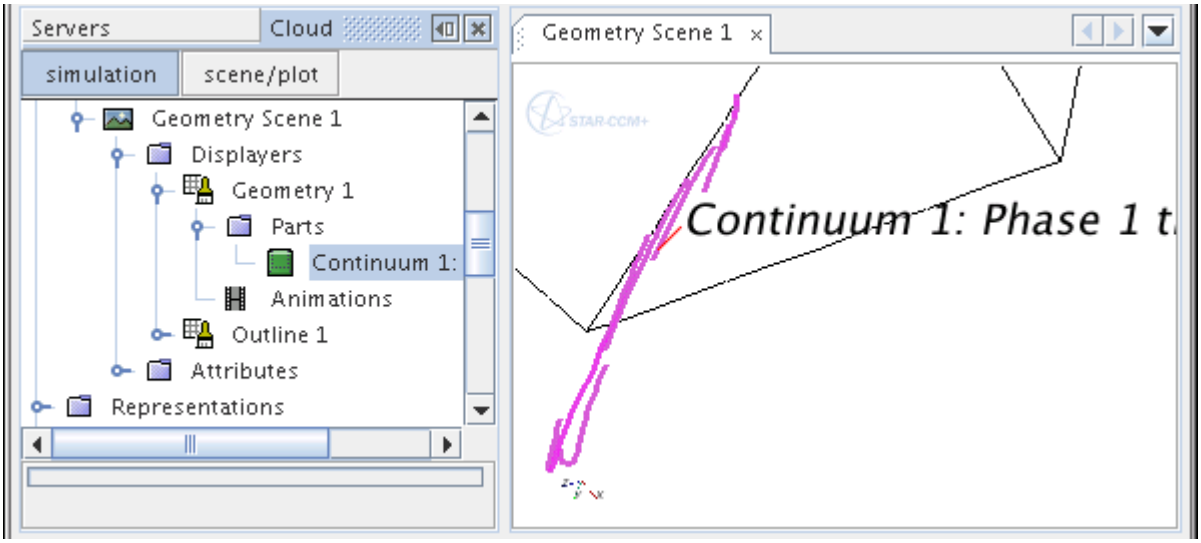
Mapping Boundary Data to Parcels

When a boundary field function is mapped to parcels, each parcel just gets the value of its host boundary face, but only if the latter is defined. Parcels which are intersecting a boundary have a host face (the face they are intersecting), whereas parcels which are in the interior do not have a host face. For the latter, a null value is mapped to the parcels for the boundary field function.

Using Particle Tracks in Analysis

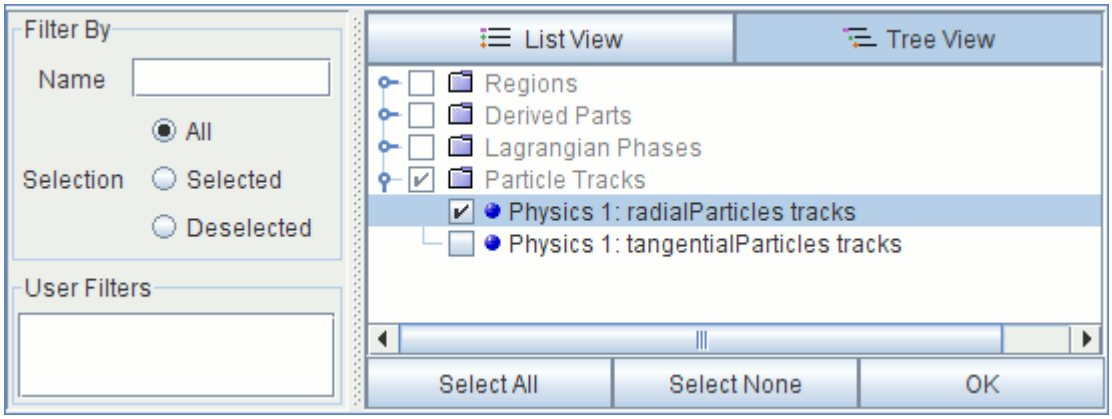
It is possible to visualize a particle track in a scene.

An example is shown below:



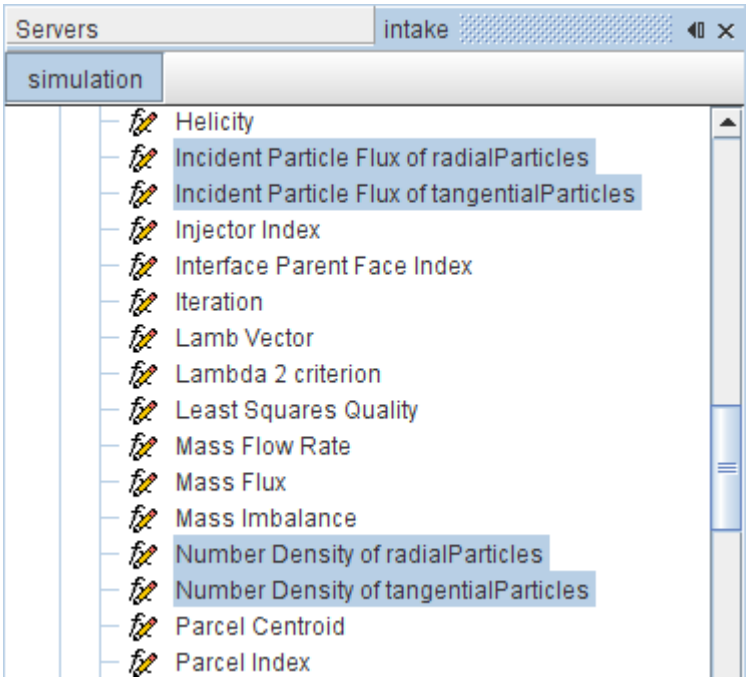
Particle tracks function similar to any other parts in a simulation in the following ways:

- You can manage the particle tracks through the scene, or through an individual displayer as shown in the following screenshot.



- A particle track can also serve as an input part for certain derived parts: a cell surface, section, threshold, or iso surface.
- It is possible to select particle tracks when specifying the parts for an xyz internal table.

When you specify the scalar for an xyz internal table, you can also access specific field functions that correspond to the particle tracks.

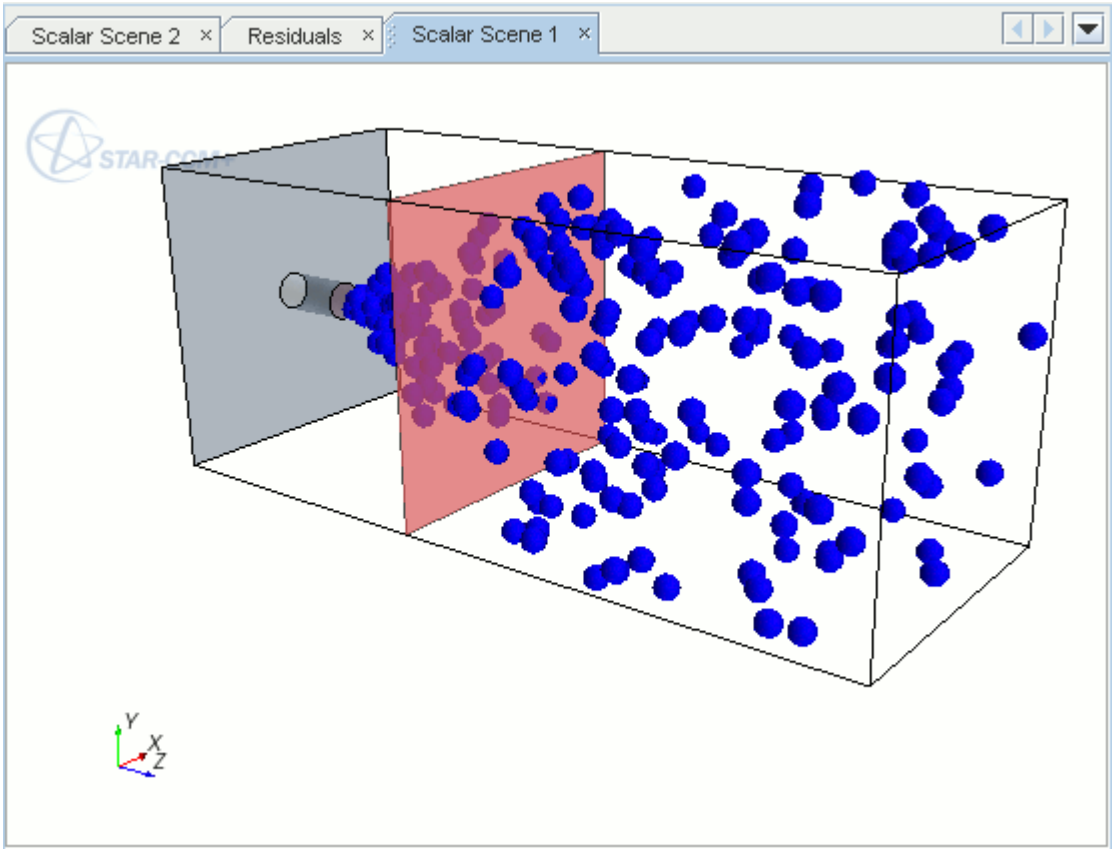


Example of Using Particle Track Data

When a track file is read into a simulation, STAR-CCM+ creates field functions for each of the scalar and vector variables that was chosen in the Track File model.

For example, if scalar variables **Parcel Index**, **Particle Count**, and **Particle Surface Area** were selected for the *Scalars* property of the node of the **Track File** model, the corresponding field functions **Track: Parcel Index**, **Track: Particle Count** and **Track: Particle Surface Area** would appear when the track file is created. These functions become available in appropriate scalar and vector function selectors throughout the simulation.

As an example of using particle track data, suppose that it was necessary to plot the distribution of parcels crossing the plane section that is shown below.



The steps that are required to obtain such a plot are as follows:

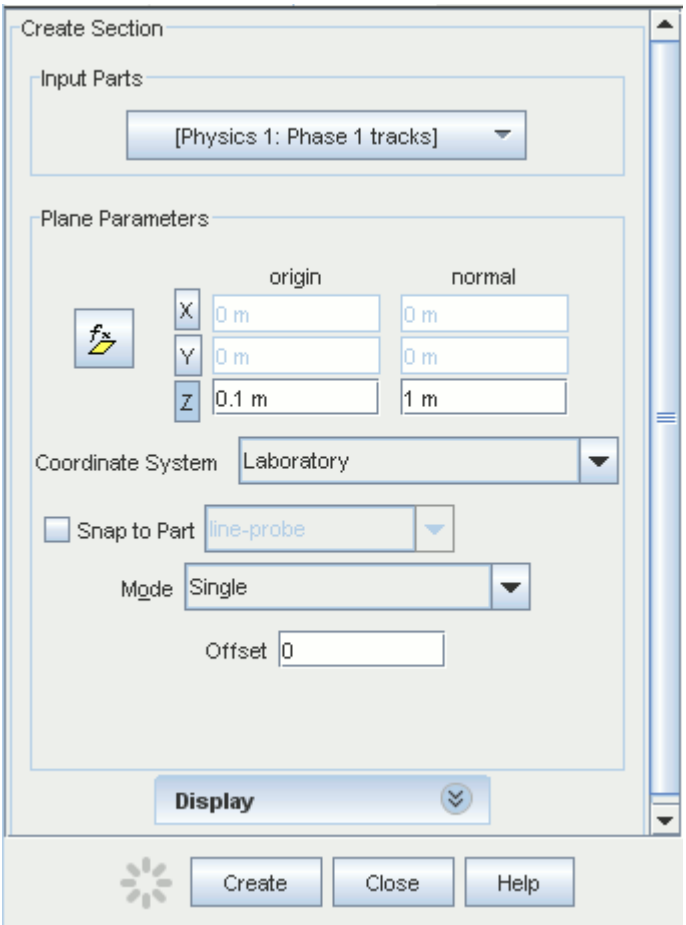
- Define a derived part that is based on the particle tracks in to obtain a discrete set of points.
- Create an XY Plot and set its Part property to the newly created derived part.
- Choose appropriate particle track variables for the X and Y scalar functions.

Defining a Derived Part Based on Particle Tracks

In this example, the steps for creating a plane section are:

1. Make an appropriate scene visible in the *Graphics* window. Right-click the **Derived Parts** node and choose **New Part** > **Section** > **Plane...**.
2. In the *Create Section* dialog, remove the default region from the selector in the *Input Parts* group box, and replace it with the particle tracks object belonging to a Lagrangian phase. More than one particle tracks object could be selected if available.
3. Set the other properties of the plane section as desired.

   A completed dialog is shown below.

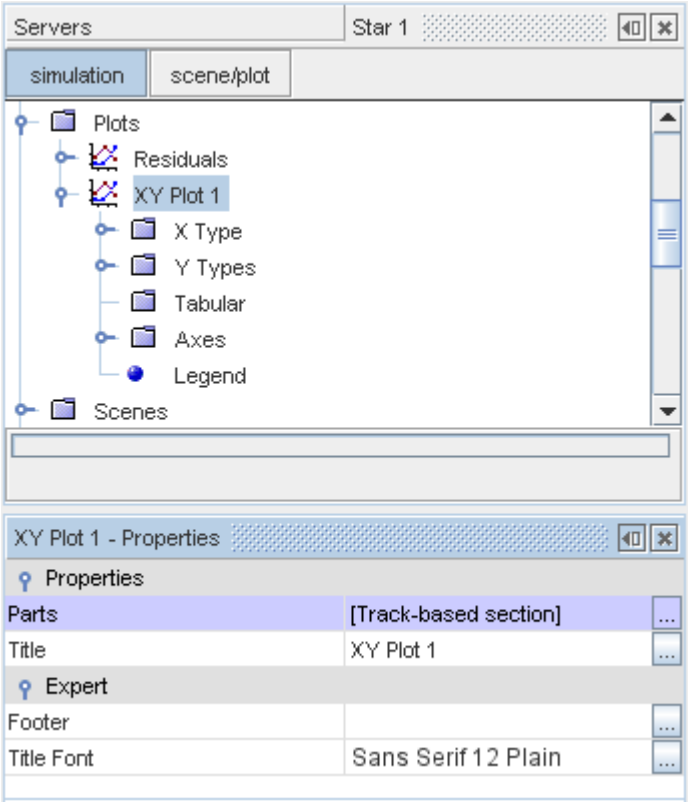


4. Click **Create** to create the plane section.
5. Rename the newly created derived part node if desired. In this example, the node is called **Track-based section**.

Creating an XY Plot with a Part Based on Particle Tracks

1. Right-click the **Plots** node and choose **New Plot** > **XY Plot** from the pop-up menu.
2. Select the newly created plot node (**XY Plot 1** in this example) and, in the *Properties* window, click to the right of the *Parts* property so that the *Select Objects* dialog appears.
3. In the *Select Objects* dialog, move the plane section that was created previously to the Selected list and click **OK**.

   The properties of the **XY Plot 1** node shows the selection in the *Parts* property.

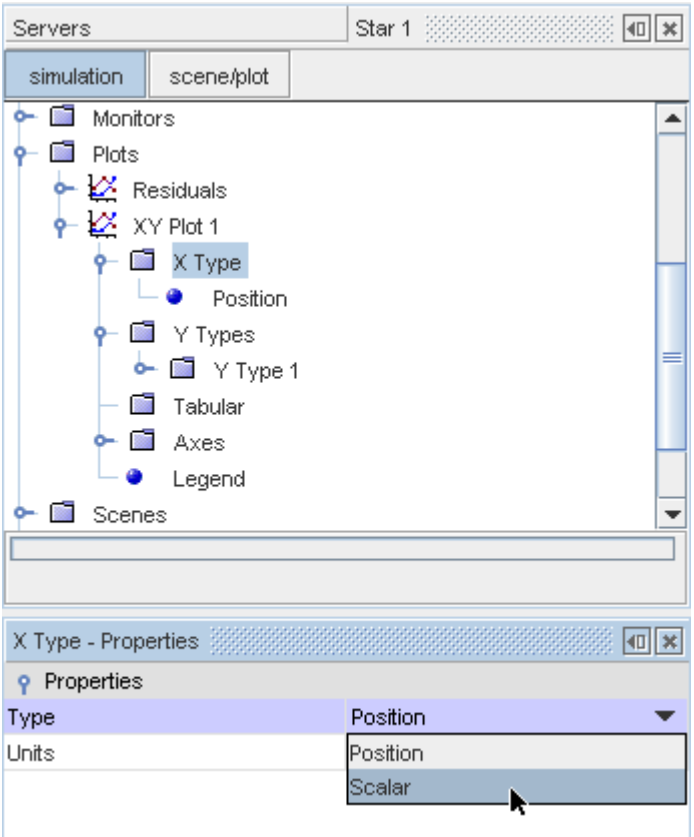


Selecting X and Y Functions Based on Particle Tracks

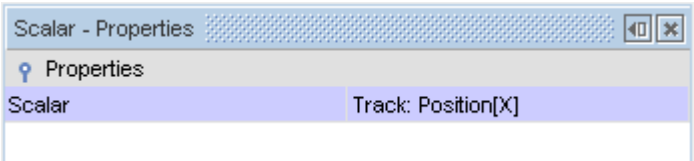
The X axis of the plot is set to show the X coordinates of particles crossing through the plane, and the Y axis of the plot shows the Y coordinates. In this way, a cross-sectional distribution of particle positions is obtained.

1. Select the **Plots** > **XY Plot 1** > **X Type** node and, in the *Properties* window, set the *Type* property to **Scalar**.

   The child node is changed to **Scalar** instead of **Position**.

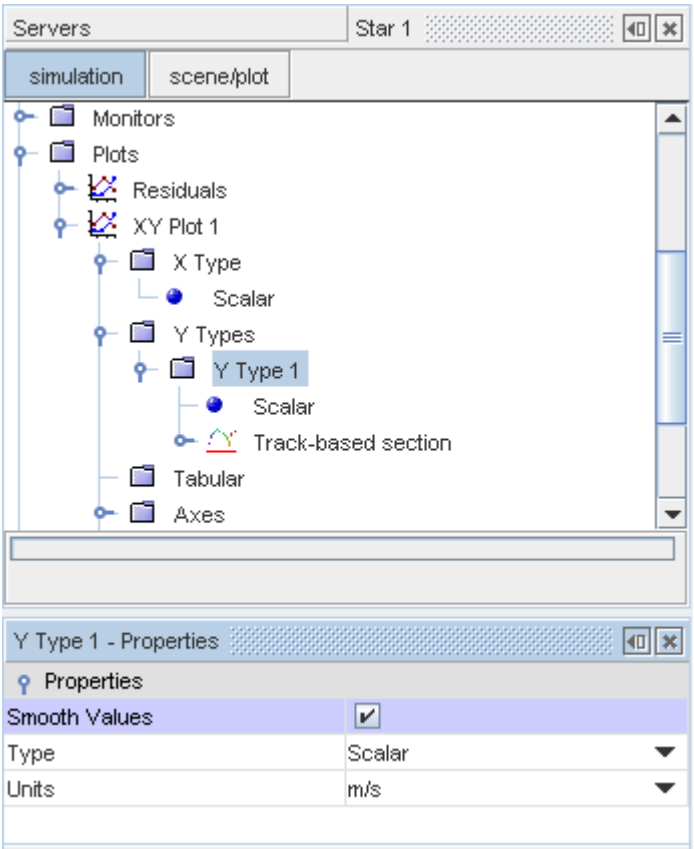


2. Select the **X Type** > **Scalar** node and, in the *Properties* window, set the *Scalar* property to **Track** > **Position** > **Laboratory** > **X**.

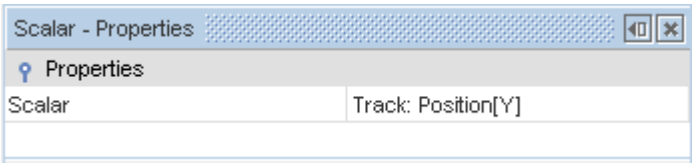


3. Select the **Plots** > **XY Plot 1** > **Y Types** node and, in the *Properties* window, activate *Smooth Values*.

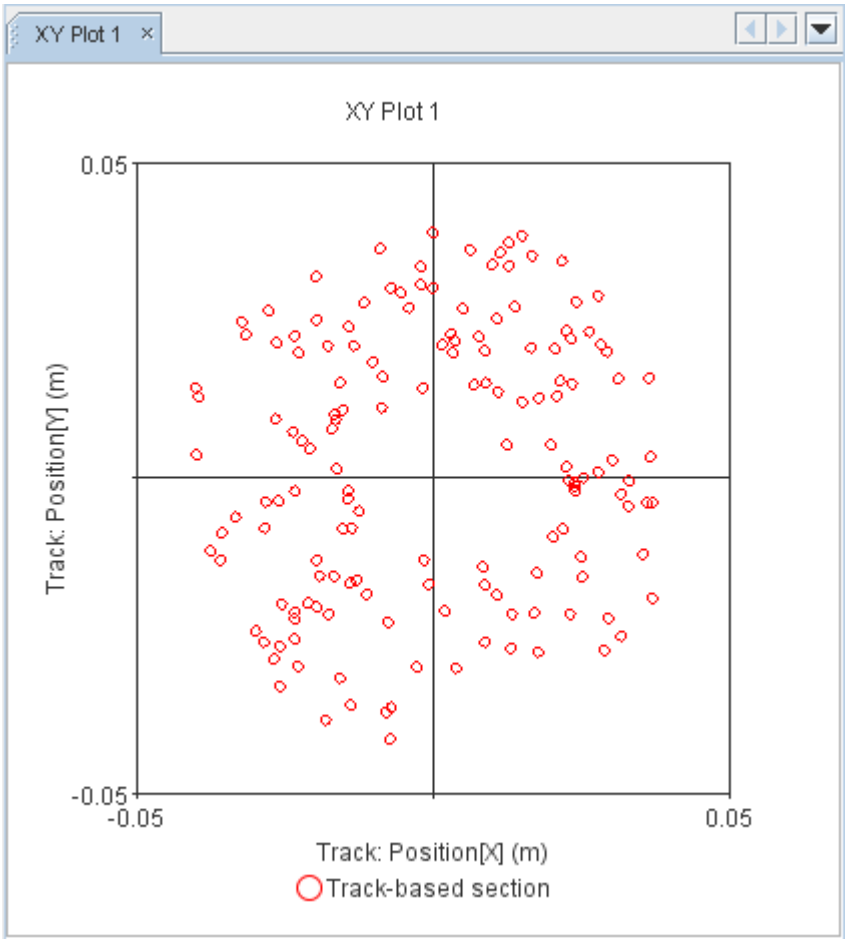
   This step is necessary, since particle tracks do not store cell center values.



4. Select the **Y Types** > **Scalar** node and, in the *Properties* window, set *Scalar* to **Track** > **Position** > **Laboratory** > **Y**.



The final plot is shown below. Chart properties have been set so that both axes have the same dimensions and scale.



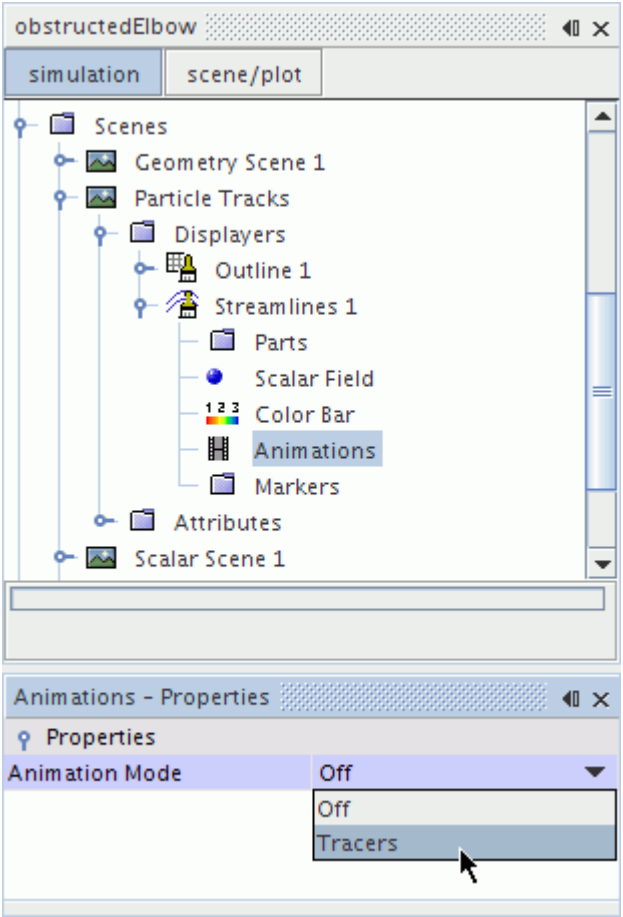
Other parcel and particle quantities could be plotted in a similar way, provided they are available from the track file.

Animating Particle Tracks

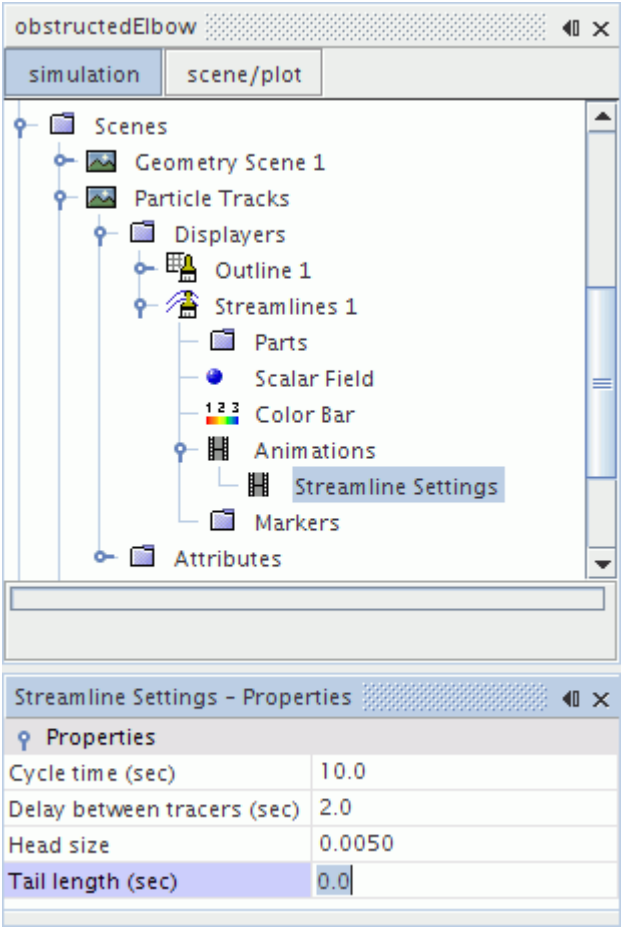
Particle tracks can be animated in a scene just like streamlines.

To animate particle tracks:

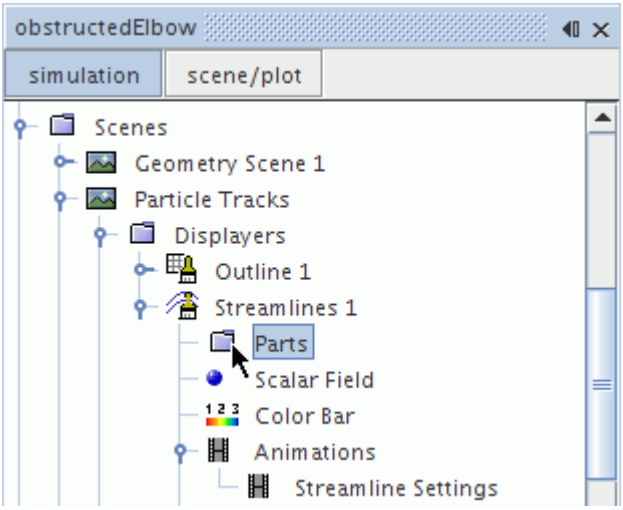
1. First, create a streamline displayer in a scene. Open its node and select the **Animations** sub-node. Set the *Animation Mode* property to **Tracers**.



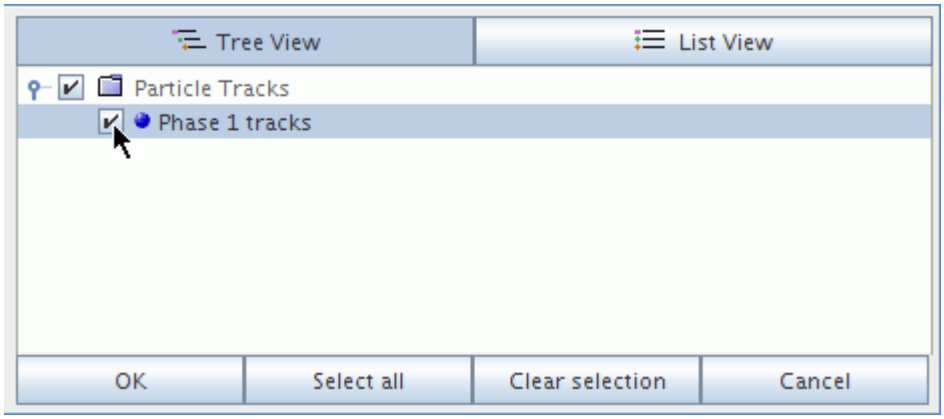
This action adds the **Streamline Settings** sub-node to the **Animations** node. This sub-node allows you to control the appearance of the tracers.



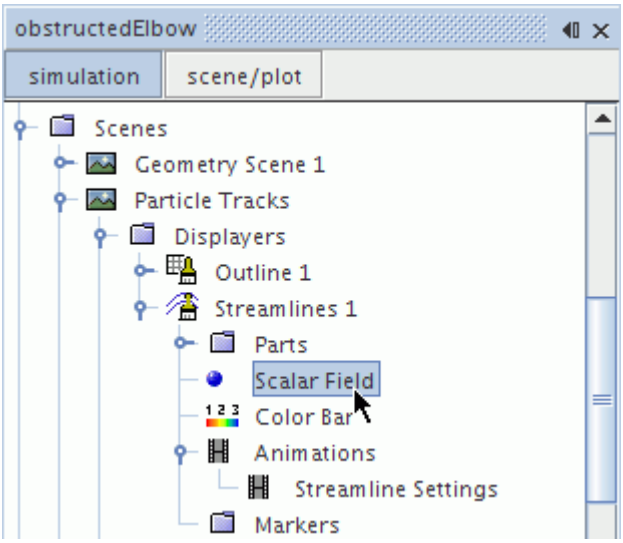
2. As you choose your settings, set the *Tail Length* property to zero to make this displayer a particle animation.
3. Once your animation settings are ready, make your streamline displayer access the particle track data. Select the **Parts** sub-node of the streamline displayer node.



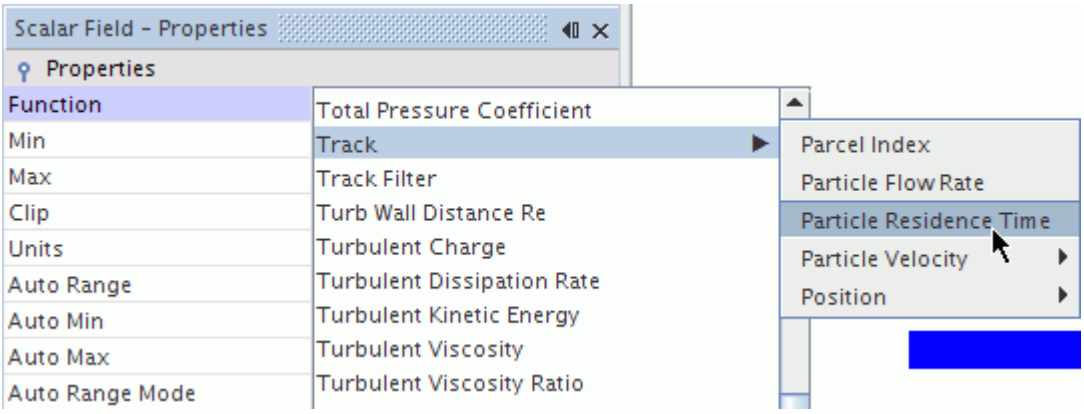
4. In the *Properties* window, activate the in-place object selector by clicking the right column of the *Parts* property. Select the particle tracks and click **OK**.



5. Once you have selected a particle track, specify what scalar function to display. Select the **Scalar Field** sub-node of the scalar displayer node.

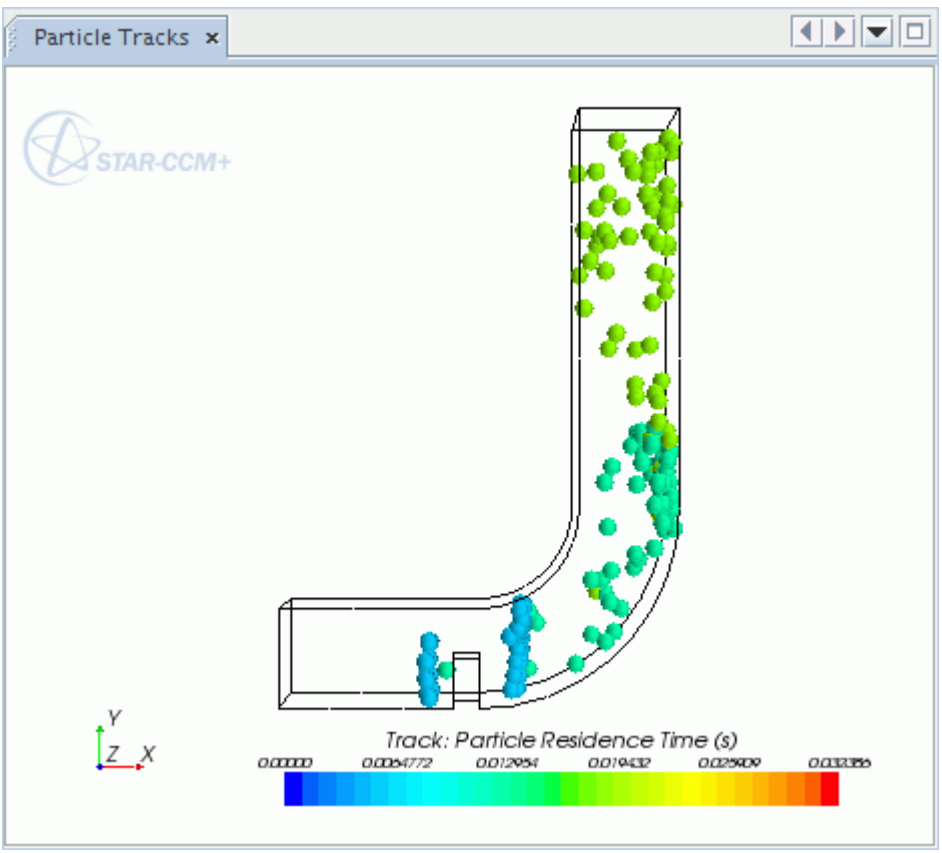


6. Set the *Function* property to **Track** > **Particle Residence Time**.



7. With the function selected, make sure that the scene display is open and active, and then click the (**Play/Pause Animation**) toolbar button to start the animation.

   The toolbar contains a dynamic set of buttons for controlling the movement.

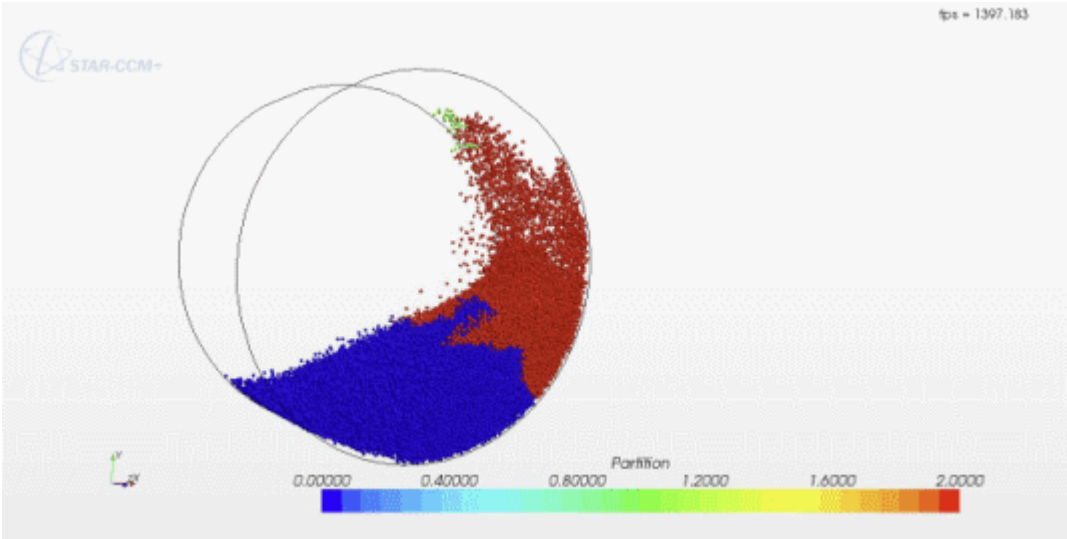


Once you have finished, you can record the animation onto separate files.

Load Balancing for Lagrangian/Eulerian Flows

For parallel processing, the traditional approach to decomposing the volume mesh is to give each processor approximately the same number of cells. For cases where the same Eulerian flow equations are solved in each cell, this approach generally gives good performance.

For cases containing particles, this traditional form of decomposition can lead to imbalanced CPU loads across the parallel cores. If one or two partitions contain most of the particles, then those partitions require significantly more compute time than partitions with few particles. The traditional decomposition does not account for this difference. See the example in the following figure, where Eulerian mesh-based domain decomposition leads to poor CPU load and low parallel efficiency.



Dynamic Load Balancing for Lagrangian Simulations

STAR-CCM+ provides an alternative approach to domain decomposition that accounts for the presence of particles.

There are two key components involved:

- The partitioning solver:

  The main load-balancing solver. This solver rebalances the domain decomposition in response to information provided by physics solvers. It also rebalances the decomposition in response to mesh topology changes, such as those caused by interface intersections.
- The Lagrangian load-balancing solver:

  This solver provides the partitioning solver with information about the computational load that is required to solve particle physics in each Eulerian cell. See Lagrangian Load-Balancing Solver.

These two solvers operate together. The Lagrangian solver collects the information that is required to rebalance the domain and passes it to the partitioning solver. The partitioning solver coordinates the final load balancing, accounting for all load-balancing criteria as it does so. This approach optimizes the domain decomposition in response to changing particle distributions. The computational load across all cores remains balanced as particles migrate throughout the domain.

Running with Load Balancing Activated

To run with load balancing, use the following steps:

1. Load the simulation into a parallel server. See Working with Parallel Servers.
2. Activate the Lagrangian load-balancing solver. See Lagrangian Load-Balancing Solver.
3. Start the run.

The load-balancing framework must complete at least one cycle before the load-balancing information can be collected and used to rebalance mesh. The first iteration or time step therefore runs with the original domain decomposition based on cell count.

VOF-Lagrangian Impingement

Lagrangian models track individual packets of finely dispersed droplets through the domain, while VOF Multiphase models track the volume of two or more interpenetrating phases. When using both modeling techniques (for example, water droplets spraying through the air into a large volume of water), you can link the two forms of modeling with a VOF-Lagrangian Impingement model which is a type of Phase Interaction model.

The workflow for combining these models is as follows:

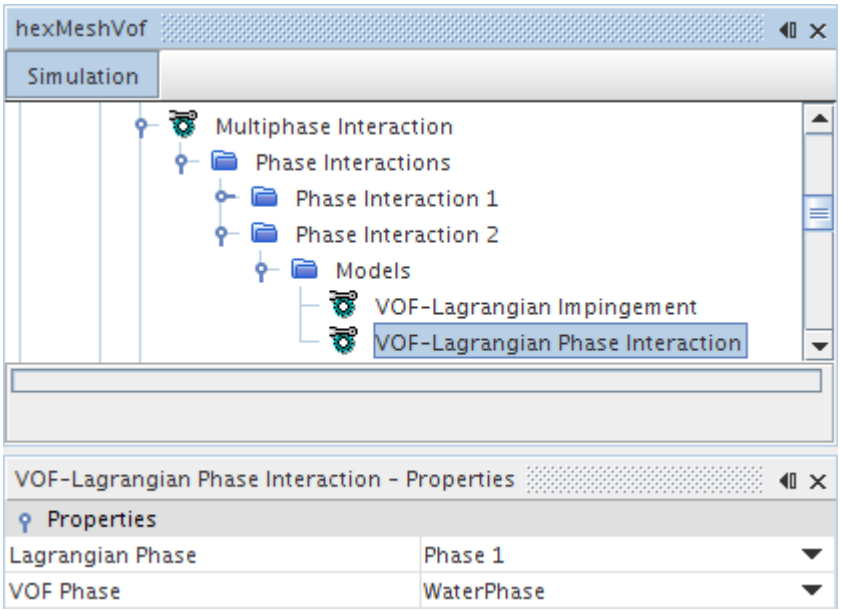
1. Start with a simulation that contains both Eulerian and Lagrangian phases, and has VOF multiphase activated for the Eulerian phase.
2. Add the following models, in order:

Node	Group Box	Model
Lagrangian Multiphase > **Lagrangian Phases** > **Phase n** > **Models**	*Optional Models*	**Two-Way Coupling**
Multiphase Interaction > **Phase Interactions** > **Phase Interaction n** > **Models**	*Phase Interaction Topology*	**VOF-Lagrangian Phase Interaction**
	Optional Models	**VOF-Lagrangian Impingement**

3. In the *Properties* window of the **VOF-Lagrangian Phase Interaction** node, select the Lagrangian and VOF phases to interact.
4. Create an injector to introduce the Lagrangian particles.

One-way coupling between VOF and Lagrangian models is appropriate when the loading from the Lagrangian phase is small and can be neglected in the VOF solution. When the Two-Way Coupling model is selected in the Lagrangian phase models, the mass, momentum, and energy of parcels that impinge on the VOF phase are accumulated as part of the solution.

When the simulation runs, the injected Lagrangian particles are deleted after impinging on any cells that are used by the VOF model. Parcels that are absorbed after impingement transfer their mass, momentum, and energy to the VOF phase you select in the interface.



The VOF-Lagrangian Impingement Model

The VOF-Lagrangian Impingement model allows you to simulate Lagrangian droplets impinging on a VOF free surface phase. The model conserves fluid volume.

The model has no properties.

The VOF-Lagrangian Phase Interaction Model

The VOF-Lagrangian Phase Interaction model allows you to select the phases involved in the interaction when Lagrangian droplets impinge on a VOF surface.

VOF-Lagrangian Phase Interaction Properties

Lagrangian Phase	Select the phase from the drop-down menu.
VOF Phase	Select the phase from the drop-down menu.

Lagrangian/Eulerian Passive Scalar Interactions

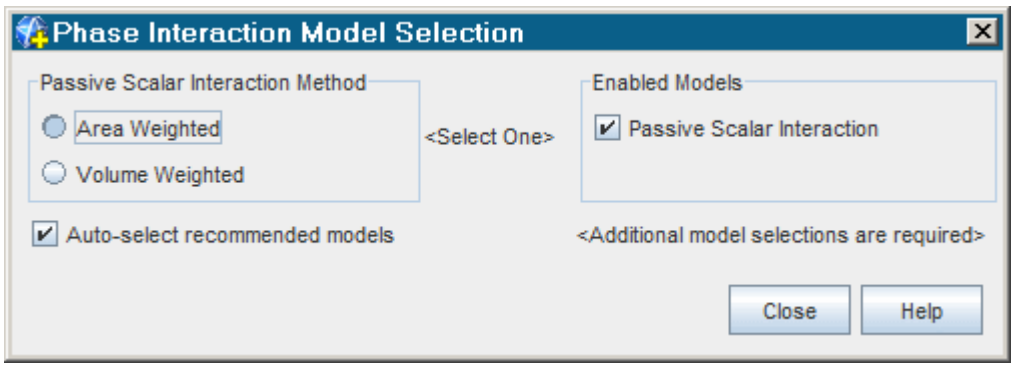
Lagrangian phases can interact both with other Lagrangian phases and with phases of other types using Multiphase Interaction models. Both Lagrangian and Eulerian frameworks give the option of using passive scalars as user-defined variables of arbitrary value, assigned to fluid phases or individual particles.

They are passive because they do not affect the physical properties of the simulation. An intuitive way to think of passive scalars is as marker paint on a particle or parcel, or marker dye in a fluid, but with numerical values instead of colors, and with no appreciable mass or volume.

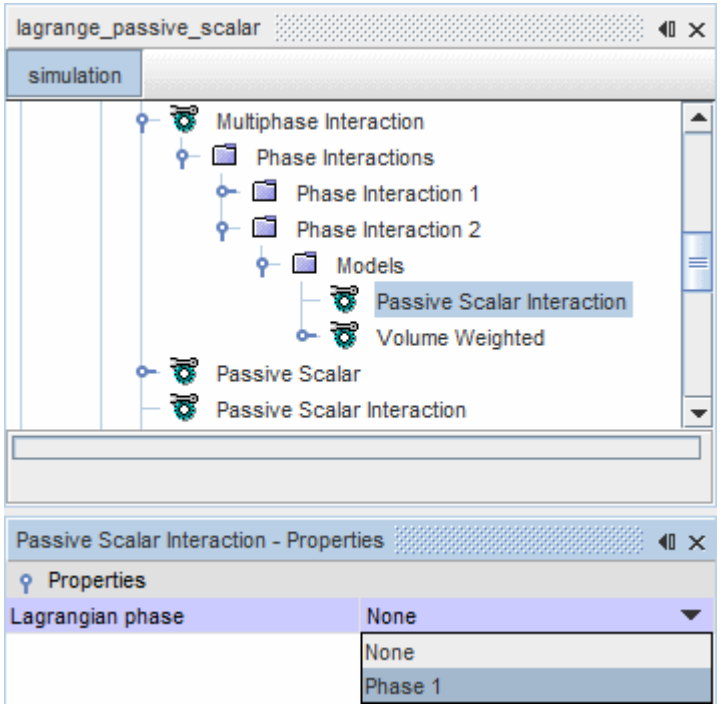
The Passive Scalar Interaction model allows transfer of values between a Lagrangian and an Eulerian passive scalar. It becomes available when a Lagrangian and an Eulerian phase both have passive scalars defined. When you select it, the **Multiphase Interaction** node also appears, if it was not already present. The Passive Scalar Interaction model has no properties. It is not available for massless particles.

To set up an interaction between Lagrangian and Eulerian passive scalars:

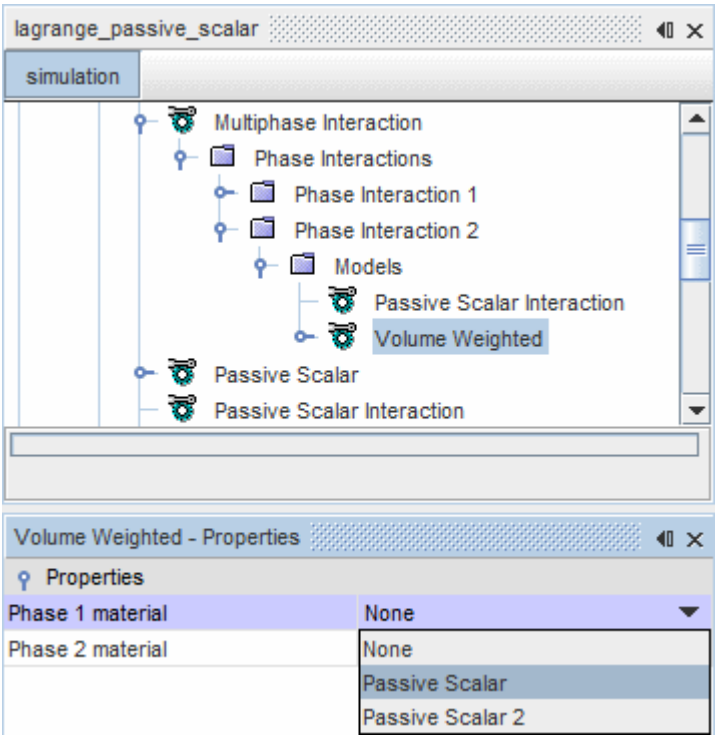
1. Right-click on the **Phase Interactions** node under **Multiphase Interaction** and select **New**. This adds a **Phase Interaction** node, with a **Models** node beneath it.
2. Right-click on the **Models** node and choose **Select models...**.
3. The *Phase Interaction Model Selection* dialog opens. Select **Passive Scalars Interaction** and either **Area Weighted** or **Volume Weighted**. Nodes for these appear under the **Models** node. See the Theory section for a description of area and volume weighting.



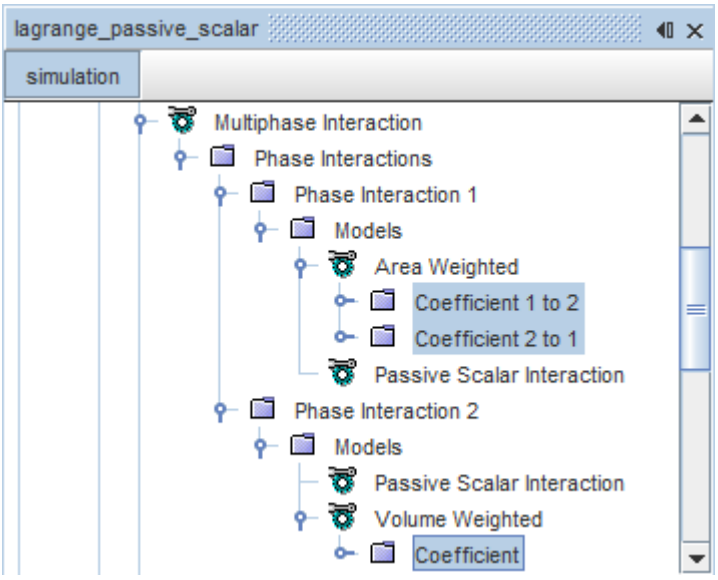
4. Open the pull-down menu of the Passive Scalar Interaction node, and select the Lagrangian phase to be involved in the interaction.



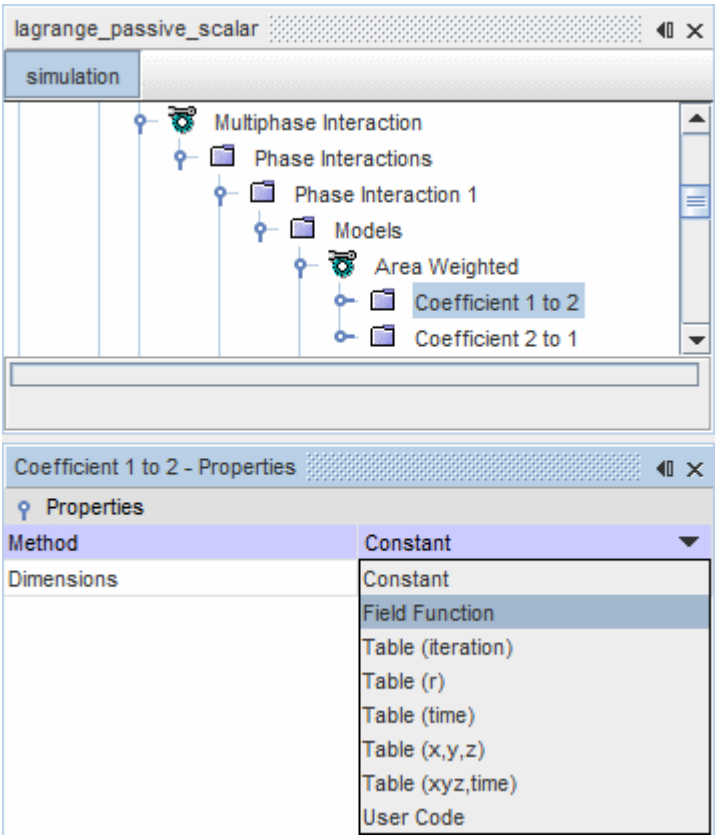
5. The properties of the **Area Weighted** or **Volume Weighted** nodes list the two interacting phases. Use the pull-down menus for each phase to assign a passive scalar to the phase.



6. The **Area Weighted** node has the nodes **Coefficient 1 to 2** and **Coefficient 2 to 1** under it. The **Volume Weighted** node as a single **Coefficient** node under it.



Set the *Method* for determining the value of each coefficient, using the pull-down menu in the coefficient properties.



Lagrangian Multiphase Field Functions Reference

This section describes the field functions that are made available to the simulation when the Lagrangian Multiphase model is used.

Field functions can be used to provide data for:

- Scenes
- Plots
- Reports
- Monitors
- Data tables

In many cases, a field function requires one or more specific Lagrangian phase models before it is available. These models are listed in brackets after the description. Some field functions also require that you activate the Lagrangian Multiphase solver expert property *Temporary Storage Retained*. These cases have "(Temporary)" after the description.

Accretion Ratio	The ratio of the rate at which ice melts to the rate at which droplets impinge, according to $-\dot{m}_{melt}/\dot{m}_d$.
Aborted Parcel Count	The number of parcels that are aborted in each cell during tracking. (Temporary)

Droplet Dynamic Viscosity	The dynamic viscosity of the droplet material, μ_l. (TAB Distortion, KHRT breakup, Primary Atomization)
Droplet Latent Heat of Vaporization	The latent heat of vaporization of the droplet material, L. (Droplet Evaporation)
Droplet Saturation Pressure	The absolute saturation pressure of the droplet material, p_{sat}. (Droplet Evaporation)
Droplet Surface Tension	The surface tension of the droplet material, σ. (TAB Distortion, Secondary breakup, Primary Atomization)
Droplet Weber Number	The droplet Weber number, Eqn. (3084). (Secondary breakup)
Eddy Velocity Fluctuation	The difference between the instantaneous and Reynolds-Averaged velocity in eddies of the particles. It is the quantity $\mathbf{v}'$ in Eqn. (3010). (Turbulent Dispersion) (Temporary)
Erosion Rate	The sum of **Abrasive Wear Rate** and **Impact Wear Rate**. It is the quantity E_f in Eqn. (3245). (Erosion)
Film Transition Mark	Returns a value of `1` at every point on the cell face where the Droplet Film Transition model is active. Returns a value of `0` where a droplet has made the transition to film, deactivating the model. (Droplet Film Transition)
Incident Mass Flux of <Phase>	The mass flux of material particles of <phase> impinging on a boundary. (Material Particles)
Incident Mass Flux of <Component> of <Mixture>	The mass flux of <component> of a liquid <mixture> impinging on a boundary. (Material Particles)
Incident Mass Flux of <Component> of <Phase>	The mass flux of <component> of the material particles of <phase> impinging on a boundary. (Material Particles)
Incident Particle Flux of <Phase>	The rate of massless particles of <phase> impinging on a boundary. (Massless Particles)
Injector Index	The index of the injector which created each parcel.
Lagrangian Charge Density	The total particle charge per unit cell volume. (Coulomb Force, Two-Way Coupling.)
Lagrangian Energy Source	The rate of energy transfer to the continuous phase from all Lagrangian phases with a Two-Way Coupling model. It is the quantity S_E in Eqn. (3036) and Eqn. (3038). (Material Particles, Two-Way Coupling, Energy) (Temporary)

Lagrangian Mass Source	The rate of mass transfer to the continuous phase from all Lagrangian phases with a Two-Way Coupling model. It is the quantity S_m in Eqn. (3040) and Eqn. (3042). (Material Particles, Two-Way Coupling, Mass Transfer) (Temporary)
Lagrangian Momentum Source	The rate of momentum transfer to the continuous phase from all Lagrangian phases with a Two-Way Coupling model. It is the quantity $\mathbf{S_v}$ in Eqn. (3036) and Eqn. (3038). (Material Particles, Two-Way Coupling) (Temporary)
Lagrangian Species Source (<Species>)	The rate of mass transfer of <species> to the continuous phase from all Lagrangian phases with a Two-Way Coupling model. (Material Particles, Two-Way Coupling, Mass Transfer, Species) (Temporary)
Mass Flow Rate	The rate at which mass flows through a boundary.
Mass Flow Rate of <Phase>	The rate at which mass of <phase> flows through a boundary. (Eqn. (1269))
Mass Flux	The rate per unit area at which mass flows through a boundary.
Mass Flux of <Phase>	The rate per unit area at which mass of <phase> flows through a boundary.
Number Density of <Phase>	The number of massless particles of <phase> per unit volume of space. (Massless Particles)
Number of Collisions	The number of collisions in a cell for a particular time-step. (No Time Counter) (Temporary)
Parcel Centroid	The position of parcels, r_π.
Parcel Index	A unique identifier for each parcel. The index is unique within a Lagrangian phase.
Parcel Mass	The mass of parcels, $m_\pi = n_\pi m_p$. (Material Particles). (Unsteady)
Parcel Mass Flow Rate	The mass of parcels, $\dot{m}_\pi = \dot{n}_\pi m_p$. (Material Particles). (Steady)
Parcel Size	Radius R_π of an equivalent sphere of mass and density equal to that of the parcel $R_\pi = 0.5 D_p \sqrt[3]{n_\pi}$, where n_π is the `Particle Count`. This field function sets the display size of parcels in scenes when the *Point Scale Mode* of a scalar displayer is set to **Particle Model**. This function is also the default function for the *Point Size Scale Field* when

	Point Scale Mode is set to **Scalar**. In steady simulations, **Parcel Size** evaluates to zero. Do not use **Parcel Size** for scaling parcel displays in steady simulations.
Parcel Time Step	The local time-step, δt_p in Eqn. (3264) and Eqn. (3265).
Parcel Volume	The volume of parcels, $V_\pi = n_\pi V_p$. (read-only). (Unsteady)
Parcel Volume Flow Rate	The volume of parcels, $\dot{V}_\pi = \dot{n}_\pi V_p$. (Material Particles). (Steady)
Particle Charge	The particle charge q. (Coulomb Force).
Particle Count	The number of particles that are represented by a parcel, n_π. (Unsteady)
Particle Density	The density of the particle material, ρ_p. (Material Particles)
Particle Diameter	The diameter of particles, D_p. (Material Particles)
Particle Drag Force	The drag force on a particle, $\mathbf{F}_d$. (Drag Force)
Particle Flow Rate	The rate at which particles flow along parcel tracks, $\dot{n}_\pi$. (Steady)
Particle Incidence Angle	The angle between the particle velocity vector and the plane of face on which it is incident, if any, $\theta = \mathrm{asin}\left(\mathbf{v}_p \cdot \mathbf{n} / \left\|\mathbf{v}_p\right\|\right)$. This function is provided for use in boundary condition expressions, it is not available for visualization, reporting, or recording on the track file. (Material Particles)
Particle Mass	The mass of single particles, m_p. Note, not the mass of parcels. (Material Particles)
Particle Mass Fraction of <Species>	The mass fraction, $Y_{pi} = m_{pi}/m_p$, of species *i* in a single particle.
Particle Molar Concentration of <Species>	The molar concentration, $C_{pi} = \rho_p Y_{pi}/W_{pi}$, of species *i* in a single particle, where ρ_p is the particle mixture density, Y_{pi} is the species mass fraction, and W_{pi} is the species molecular weight.
Particle Mole Fraction of <Species>	The mole fraction, $X_{pi} = Y_{pi} W_p / W_i$, of species *i* in a single particle, where Y_{pi} is the species mass fraction, W_p is the molecular weight of the particle

	mixture, and W_{pi} is the molecular weight of the species.
Particle Nusselt Number	The non-dimensional heat transfer coefficient of particles, Nu_p, defined in Eqn. (3030). (Lagrangian Energy)
Particle Permittivity	The relativty permittivity of the particle ϵ_r. (Field Charging).
Particle Porosity	The porosity of particles within the Lagrangian phase. (Particle Porosity)
Particle Residence Time	The “age” of particles as measured from time of injection.
Particle Reynolds Number	The quantity Re_p in Eqn. (2977). (Material Particles)
Particle Sherwood Number	The non-dimensional mass transfer conductance, Sh_p, defined in [eqnlink]. (Mass Transfer)
Particle Slip Velocity	The velocity deficit of particles relative to the continuous phase, $\mathbf{v}_s = \mathbf{v} - \mathbf{v}_p$. (Material Particles)
Particle Smooth Source Group	Unsigned integer ID for the cluster to which a cell belongs when the cell cluster method is active for two-way coupling. (Two-Way Coupling and Two-Way Coupling Solver)
Particle Smooth Source Weight	The volume fraction of a cell relative to the total volume of the cluster in which it belongs. The fluid source terms generated by particles are summed over the cluster, then distributed back to the cell using this factor. The result is that the source contribution per unit volume is uniform across each cluster. (Two-Way Coupling and Two-Way Coupling Solver)
Particle Specific Heat	The specific heat of particles. (Lagrangian Energy)
Particle Surface Area	The surface area of single particles, A_s . Note, not the surface area of parcels. (Material Particles)
Particle Temperature	The temperature of particles, T_p . (Lagrangian Energy)
Particle Velocity	The absolute velocity of particles, $\mathbf{v}_p$.
Particle Volume	The volume of single particles, $V_p = m_p/\rho_p$. Note, not the volume of parcels. (Material Particles)

Stuck Mark	A particle stuck to a cell face has a **Stuck Mark** value of `1`. Otherwise, the value is `0`. See Stick Mode. (Bai-ONERA, Bai-Gosman, Droplet Film Transition)
TAB Distortion	The normalized distortion of droplets, y in Eqn. (3101). (TAB Distortion) (Temporary)
TAB Distortion Rate	The rate of change of **TAB Distortion**, $\dot{y}$. (TAB Distortion) (Temporary)
Void Fraction	The fraction of the cell volume available to the fluid, unoccupied by a solid or dispersed phase.
Volume Fraction of <Phase>	The volume of material particles of <phase> per unit volume of space. (Material Particles)

Post-Processing Lagrangian Data

STAR-CCM+ allows you to analyze the flow of Lagrangian phases through the containing fluid, and also the interaction of these phases with boundaries.

Four sources of data are available to post-process Lagrangian analyses:

- Direct data on boundaries
- Local instantaneous data
- Particle track data
- Sampled data on boundaries

Direct Data on Boundaries

The following field functions are available on boundaries:

- Incident mass flux (particle flux for massless particle): computes the instantaneous mass flux (or particle flux) of one Lagrangian phase interacting with a boundary (hitting it or escaping through it).
- Erosion rate: computes the estimated loss of mass material per second due to particle impact. This field function is available only if erosion is activated in Lagrangian models. See Modeling Erosion.

See Lagrangian Multiphase Field Functions Reference.

Local Instantaneous Data

To analyze the particle field at an instant, the Lagrangian phase is available as a part for the scalar or geometry displayer. The data available on this field are Particle and Parcel field functions. The parcels are shown as small spheres, whose size and color can both be field function dependent.

As an example of how you can use local instantaneous data, consider the following steps for counting the number of particles in a particular set of regions:

1. Open the **Regions** node and select the regions for which you want the parcel counts.
2. In the properties for each region, note the value of the *Index* property, an integer *n*.
3. Right-click **Derived Parts** and select **New Part** > **Threshold...** to create a **Threshold** node under **Derived Parts**.

4. Select the new threshold node, set the following values:

Property	Value
Parts	**Lagrangian Phases**
Scalar Field	**Region Index**
Mode	**Between** In the special case of *Index* = 0, use **Below Min**.
Range	Enter `[a, b]`, a range that includes the indices of the regions for which you want the parcel count. For example, for parcel counts of regions 2, 3, and 4, enter `[2, 4]`.

   The threshold now passes all and only the parcels with **Region Index** values in the specified range.

5. Right-click **Reports** and select **New Report** > **Element Count** to create an **Element Count** node under **Reports**.
6. In the **Element Count** node properties, edit the *Parts* property, open **Derived Parts**, and select the threshold node that was created earlier.
7. Right-click the **Element Count** node and select **Run Report**. The number of parcels in each region is displayed in the *Output* window.

The above steps are based on a set of regions with consecutive indices. Regions with non-consecutive indices require multiple reports.

Particle Track Data

The Lagrangian track file is an object which stores the field function values for Lagrangian particles as the calculation proceeds. It can be used as a post-processing part and data source. Post-processing is set up as follows:

1. Activate the Track File in the Lagrangian models selection box. Select the desired field function: the values available are particle and parcels relative field functions.
2. Run the calculation. The track data is stored in a temporary file that is named `<simname>.trk.tmp`, where `<simname>` is the simulation name.
3. Save the simulation. This action appends the temporary track file to the existing one, or creates a permanent track file with name `<simname>.trk`. (Check the output.)
4. Load the track file in the simulation by right-clicking on the node **Tools** > **Track Files**. This action generates a new object in the **Particle Track** node.
5. You can now use the particle track as a part in a Geometry, Scalar, or Streamline displayer. The data available from the field functions with names beginning with “Track.” See Using Displayers.

Boundary Sampling

The Boundary Sampling model, available in the *Lagrangian* selection box, allows the storage of field function values on a boundary during a calculation. This feature is similar to the Track Files feature but with these differences:

- Data is not stored in an external file, so you do not need to save the simulation, nor to load a track file to use the tracks from the Boundary Sampling.
- The stored data is for one Lagrangian step, so it is necessary to monitor the desired values during the calculation. However, unlike the track file, you can monitor the evolution of this data during the calculation, as with every Eulerian scalar or vector field.
- This data is only available at a local position, so if Boundary Sampling tracks are used in a Streamline displayer, it is not possible to see trajectories.

Use the following workflow:

1. Select the Boundary Sampling physics model.
2. In the **Boundary Sampling** node properties:
   a. Edit the *Scalars* property and select the field functions that you want to track on the parcels.
   b. Edit the *Boundaries* property and select the boundaries that you want to sample.
3. Create a sum report and give it an appropriate name. (See Sum.)
4. In the sum report properties:
   a. Edit the *Scalar Field Function* property and select the Track functions for the functions that are selected for the Boundary Sampling physics model.
   b. Edit the *Parts* property and select the Particle Tracks you want to sample.
5. Run the simulation, then right-click the report and select **Run report**.

See Modeling with Boundary Sampling.

Simulation History Phase Data

Use simulation history files to save scalar and vector data on selected Lagrangian phases. See Configuring the Simulation History File.

By default, simulation history files record data from the following field functions when one of their inputs is a Lagrangian phase:

- **Parcel Centroid**
- **Parcel Orientation**
- **Parcel Size**

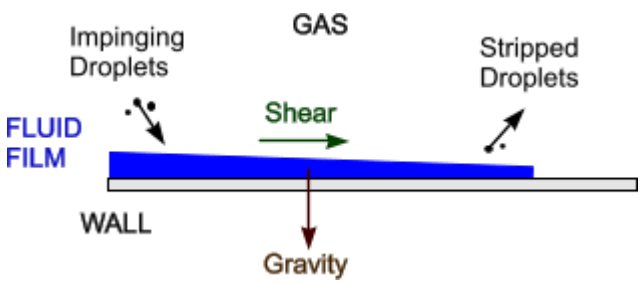
When the *Export for reporting* property is **On** and one of the inputs is a Lagrangian phase, simulation history files record data from the following field functions by default:

- **Particle Count** (for the Unsteady model)
- **Particle Diameter**
- **Particle Flow Rate** (for the Steady model)
- **Particle Mass**
- **Particle Surface Area**
- **Particle Volume**

Modeling Fluid Film

Problems where a thin film of fluid exists on solid boundaries are common in engineering practice. Examples include fluid film on the internal surfaces of automotive engines, gas turbines, spray-cooling systems and ink-jet printers.

Depending on the conditions, complex interactions can occur between the fluid film and the surrounding environment. These interactions require further mathematical modeling to capture the details of the developed flows. The Fluid Film model in STAR-CCM+ provides a mathematical description of the behavior of such films. Examples of physical processes that can be modeled with fluid films include gravity driven flow, shear forces on the film interface, external forcing through user-defined momentum source terms and evaporative cooling.



There are different methods by which a film can be formed and removed:

- You can specify the properties of an initial layer of fluid film on a boundary surface.
- A fluid film can flow from a fluid film inlet, and can be removed at an outlet.
- You can specify both a fluid film mass source and a fluid film mass sink to form or remove a fluid film.
- A film can be formed as droplets impact boundary surfaces. For this method, you define injectors for fluid particles with the Lagrangian Multiphase model and a Film-Lagrangian Phase Interaction model with impingement. You can also model the impingement of a dispersed Eulerian phase on a surface to form a fluid film.

  It is also possible to model the stripping of droplets from the surface of a fluid film due to wave-induced instabilities or sharp-angled edges. Droplets that are formed in this way are later tracked using the Lagrangian Multiphase model.
- A film can be formed as a gas, or gas component, condenses onto a surface. The film can be removed as liquid evaporates from the surface.

  For this method, define a Film-Eulerian Phase Interaction (for Multiphase Segregated Flow simulations) or a Film-Physics Continuum Interaction (for all other simulations) together with the Film Evaporation and Condensation model. The Film Boiling model is available as an extension of the Film Evaporation and Condensation model.

You can model fluid film melting and solidification. STAR-CCM+ also supports multi-component fluid film models: you can model a mixture of different species components within the film. It is also possible to simulate reactions within the fluid film, see Reacting Species Transport Workflow.

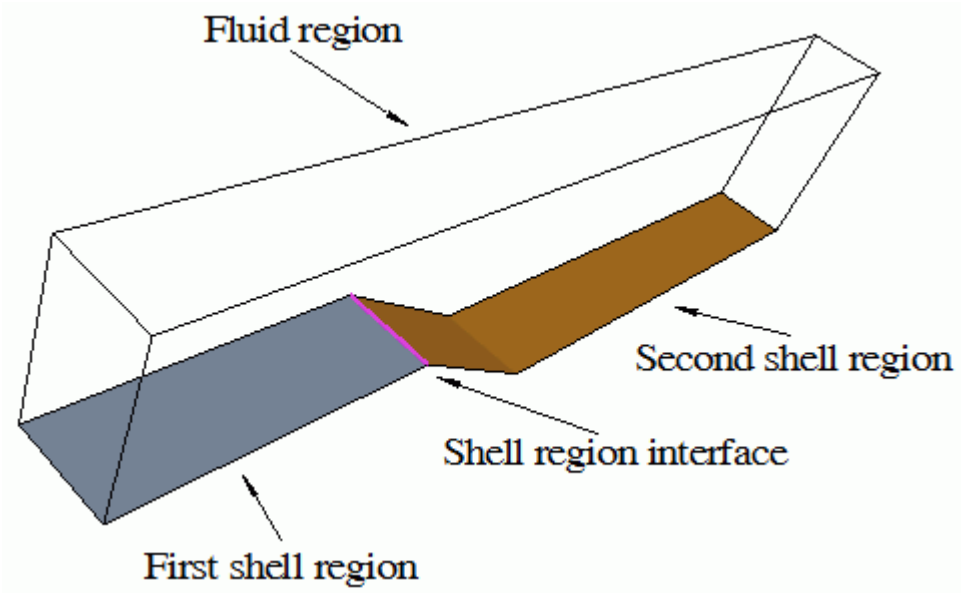
Shell Regions

A fluid film is not modeled directly within a regular STAR-CCM+ region. Instead, STAR-CCM+ uses a shell region that you create as the space within which the fluid film flows.

A shell region is a surface domain in space. The region is effectively two-dimensional—only one cell thick—and its boundaries are edges. Shell regions can be created from existing region boundaries.

One shell region is joined to a region or another shell region using an interface. When you create a shell region, an interface between the original fluid region and the shell region is created automatically at the boundary that you selected for creating the shell region.

When the fluid film is expected to flow over multiple shell regions, an interface must exist between them. These different shell regions are created from boundaries belonging to different regions. Similarly, if shell regions have been created separately within a region, create an interface manually between them.



STAR-CCM+ supports motions and moving reference frames for fluid film simulations.

Contents:

Setting Up a Fluid Film Simulation
Fluid Film Model Reference
Fluid Film Phase Models
Fluid Film Phase Interactions
Modeling Impingement
Modeling Film Stripping
Modeling Fluid Film Evaporation and Condensation
Modeling Fluid Film Boiling
Modeling Fluid Film Melting and Solidification
Modeling Fluid Film Surface Tension
Modeling a Fluid Film with Different Characteristic Length Scales

Fluid Film Formulation

Setting Up a Fluid Film Simulation

STAR-CCM+ assumes that you are modeling a fluid film on the surfaces of a volume that contains a background fluid. While the background fluid occupies a regular region, the fluid film must flow in a three dimensional shell region.

Before setting up a fluid film analysis, you must already have a physics continuum defined with a single or multiphase flow model and associated fluid materials. You can model fluid films in a steady or an unsteady simulation.

For steady simulations, the following conditions apply:

- Impingement, wave stripping, and Film-VOF phase interactions are not supported.
- The Stabilised Film Thickness Equation must be activated in the Fluid Film Flow model.

  A steady fluid film setup sometimes has a zero coefficient matrix for the film thickness equation. This issue can occur when the film thickness, as well as the film velocity, are initially set to zero. When this issue occurs, the **Stabilized Fluid Film Thickness Equation** option must be activated in the Fluid Film Flow model in order to get the simulation to run.

A Fluid Film simulation can include effects such as impingement, film stripping, evaporation, condensation, and boiling.

To set up a Fluid Film simulation:

1. Open the model selection dialog for the physics continuum that provides the background fluid. From the **Optional Models** group, select the Fluid Film model.
2. Within the **[physics continuum]** > **Models** node, right-click the **Fluid Film** > **Fluid Film Phases** node and select **New**.

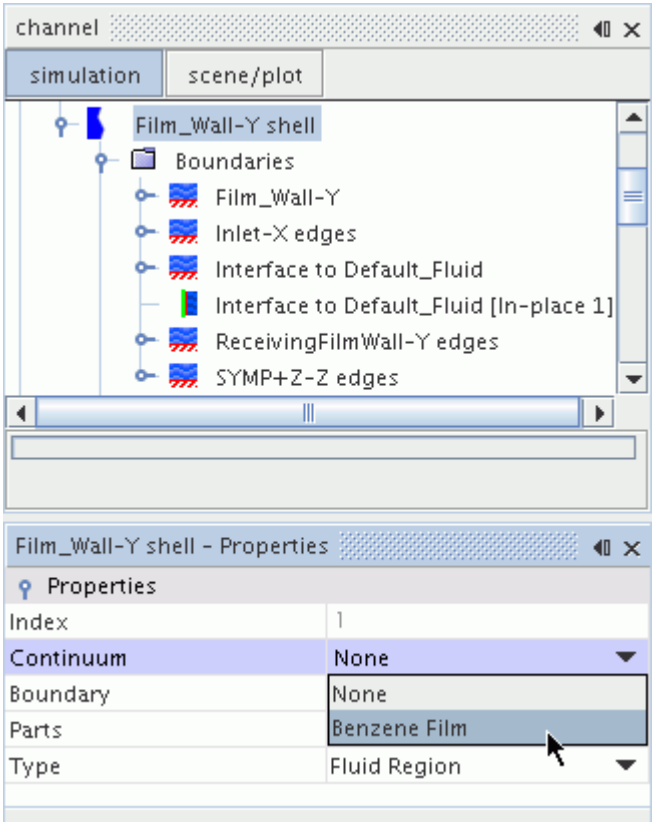
   STAR-CCM+ adds a film phase.
3. Define the film phase:
   a) Right-click the **[film phase]** > **Models** node and choose **Select Models...**
   b) Choose the film phase models appropriate to your analysis.

      See Fluid Film Phase Models.
   c) Expand the nodes under the fluid film phase and modify the material properties to match the film you wish to model.
   d) Select the **[film phase]** > **Initial Conditions** > **Fluid Film Thickness** node and provide a value or function for the initial thickness.

      The tutorial, Fluid Film: Liquid Film Flow, demonstrates how you can use a field function to define an initial thickness according to spatial location.
4. Create the fluid film region on the surfaces of the containing fluid region:
   a) Expand the **Regions** > **[containing region]** > **Boundaries** node. Multi-select boundaries on which you expect the film to flow. Include boundaries that have no initial film, but which receive film during the simulation, for example due to impingement or condensation.
   b) Right-click one of the selected boundaries and select **Create Shell Region**.

STAR-CCM+ adds a new shell region to the simulation tree. In addition, STAR-CCM+ creates an interface between the fluid region and the shell region, based on the original boundaries of the fluid region.

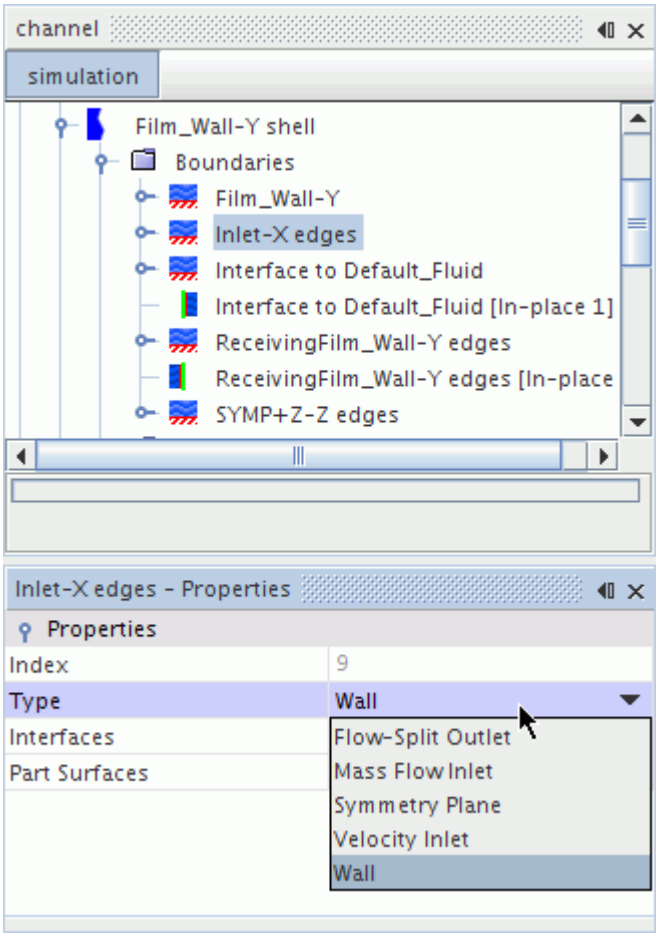
c) Set the shell region continuum to specify the fluid film phase that is associated with the shell region.



Once the shell region is connected to a fluid film phase, it becomes an active part of the simulation.

d) Set up the shell region boundaries, interfaces, and physics conditions for the fluid film flow in the same way as for any other flow in STAR-CCM+.

The boundary type that you select applies the conditions appropriate to that type to the fluid film.



Boundaries within the same fluid region belong to the same shell region.

e) If fluid film passes from one shell region to another, join the two regions using an in-place interface.

f) In the **Physics Conditions** > **Initial Condition Option** node, define the initial film conditions.

By default, the shell region uses the global settings of the continuum. If you want to set different initial conditions for the shell region, select **Specify Region Values**, and then set the appropriate values under the **Initial Conditions** node.

g) If you want to model a moving shell region, assign the appropriate motion or moving reference frame to the volume region from which the shell was created.

The specified motion and reference frame are set in the **Physics Values** > **Motion Specification** node in the associated shell region. These settings are read-only in the shell region. The *Lead Region* property indicates the volume region that the shell region follows. This region is the volume region from which the shell region was created.

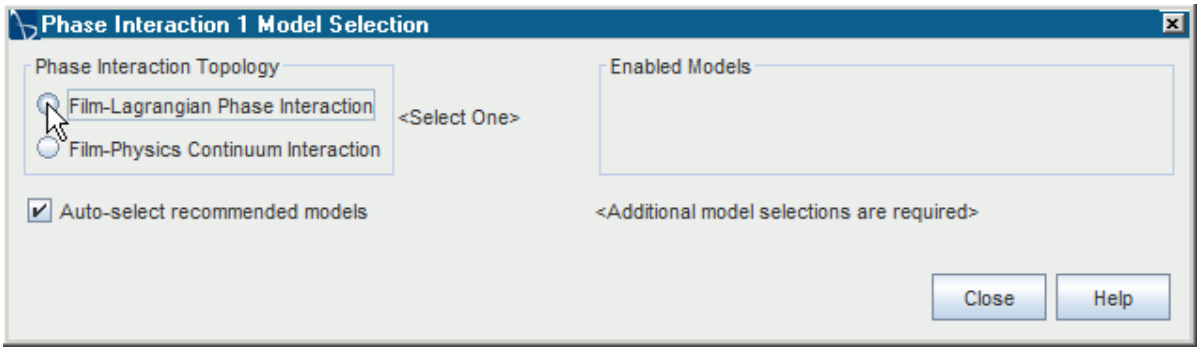
The fluid film model supports additional options that you can add to the simulation:

- Films formed by vapor condensation and reduced by evaporation.
- Films formed by droplet impingement and reduced by film stripping.

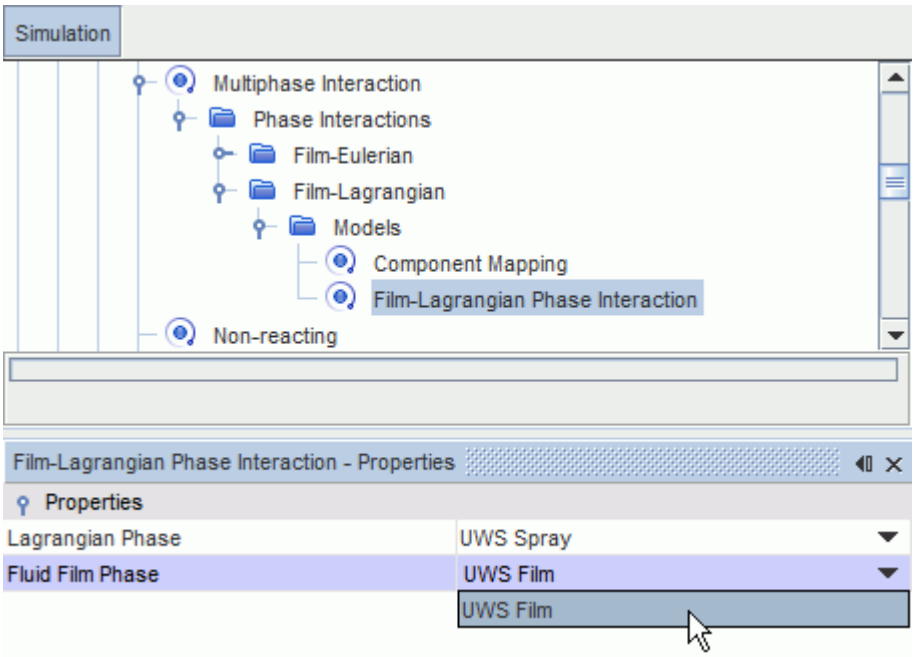
5. Select the appropriate optional models in the physics continuum:
   - To model fluid film stripping or impingement, select the Multiphase Interaction and the Lagrangian Multiphase models. See Modeling Impingement or Modeling Film Stripping.

- To model heat transfer or energy through the fluid film, select Coupled Energy or one of the Segregated Fluid Energy models.
- To model film evaporation and condensation, select the Multiphase Interaction model.

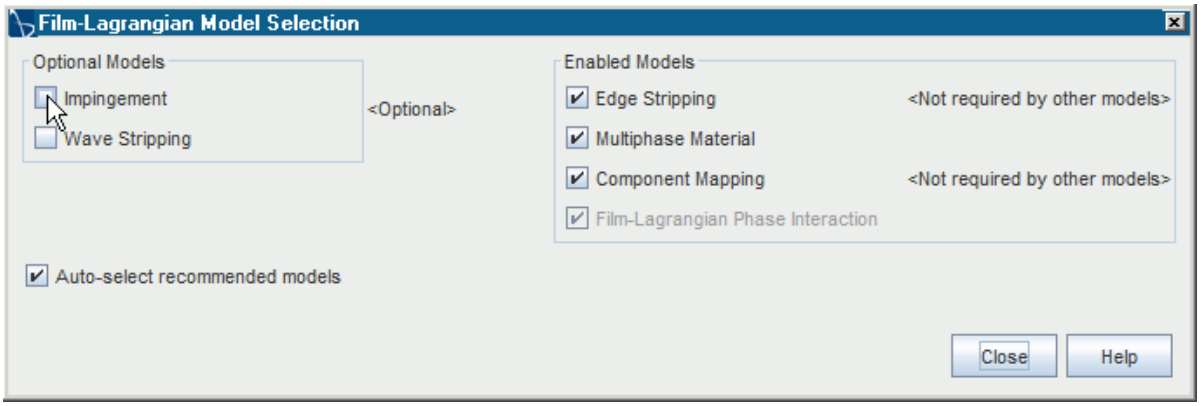
6. Under the Multiphase Interaction node, set up the appropriate phase interactions:
   a) In the *Phase Interaction Model Selection* dialog, specify the Phase Interaction Topology. For example, **Film-Lagrangian Phase Interaction**.



   b) Select the new phase interaction and specify the two phases.



   c) In the *Phase Interaction Model Selection* dialog, select the appropriate models. For example, **Edge Stripping** and **Impingement**.



For information on the phase interaction types and the optional models that are available for each, see Fluid Film Phase Interactions and Film Phase Interaction Model Family Reference.

7. Prepare for film analysis:
   a) To view film thickness on a wall surface, create a scalar scene containing the wall boundaries of the film shell region and set the scalar field function to **Fluid Film Thickness**.

      For an example, see the **Tutorial Guide > Multiphase Flow > Fluid Film: Liquid Film Flow > Visualizing the Solution**.
   b) To view the film thickness profile along a line, first create a plane section derived part on the film shell region. This derived part is a line as the film shell region has zero thickness. Use the derived part in an X-Y plot whose field function is set to **Fluid Film Thickness**.

Fluid Film Model Reference

The Fluid Film model is used in problems where a thin film of fluid exists on solid boundaries. Example applications include fluid film on the internal surfaces of automotive engines, gas turbines, spray-cooling systems and ink-jet printers.

The Fluid Film model is subject to certain assumptions, as described in the Fluid Film Formulation.

Fluid Film Model Reference

Provided By	**[physics continuum]** > **Models** > *Optional Models*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Fluid Film**
Requires	*Space*: **Three Dimensional** *Time*: **Implicit Unsteady**, **PISO Unsteady**, or **Steady** *Material*: one of **Gas**, **Multi-Component Gas**, **Eulerian Multiphase** (For Multi-Component Gas) *Reaction Regime*: **Non-reacting** or **Reacting** (For Gas and Multi-Component Gas) *Flow*: **Coupled Flow** or **Segregated Flow** (For Eulerian Multiphase) *Eulerian Multiphase Model*: **Multiphase Segregated Flow** or **Volume of Fluid (VOF)**

Activates	Model Controls (child nodes)	**Fluid Film Phases**
	Solvers	**Fluid Film**
	Field Functions	See Fluid Film Model Field Functions.

Fluid Film Model Field Functions

When you activate the Fluid Film model, the following field function is made available to the simulation:

Fluid Film Thickness	h_f in Eqn. (546).

Contents:

Fluid Film Solver Reference

Fluid Film Solver Reference

The Fluid Film solver lets you customize solver settings for segregated flow, film velocity, and film thickness. The fluid film solvers generally use the same techniques as the Segregated Flow solvers. Differences arise from the fact that these solvers work with film equations on surfaces, and film control volumes depend on the film distribution.

The Fluid Film solver incorporates the following solvers:

- Fluid Film Segregated Flow Solver
- Film Velocity Solver
- Film Thickness Solver

Fluid Film Segregated Flow Solver Properties

The Fluid Film Segregated Flow solver lets you adjust the Segregated Flow solver parameters specifically for the fluid film calculations.

Temporary Storage Retained

When activated, retains data from temporary storage at the end of the iteration. Examples include the ap coefficients, the residual, the corrections, the reconstruction gradients, and the cell gradients. These quantities become available as field functions during subsequent iterations.

Solver Frozen

When activated, prevents the solver from updating the solution while iterating or responding to other solvers. In general, to freeze flow or turbulence, activate **Temporary Storage Retained** for both flow and turbulence for one iteration after freezing any solver.

Film Velocity Solver Properties

The Film Velocity solver uses a form of the Segregated Flow Velocity solver for the fluid film. It solves the momentum conservation equation, Eqn. (547).

Under-Relaxation Factor

At each iteration, this property governs the extent to which the newly computed film velocity supplants the old one. The default value is 0.7.

Film Thickness Solver Properties

The Film Thickness solver calculates the mass conservation equation (Eqn. (546)) which is expressed in terms of film thickness corrections. This solver is equivalent to the Segregated Flow Pressure solver which calculates the mass conservation equation that is expressed in terms of pressure corrections.

Under-Relaxation Factor

At each iteration, this property governs the extent to which the newly computed film thickness field supplants the old one. The default value is 0.7.

Fluid Film Phase Models

You create the necessary fluid film phases and specify the appropriate models. STAR-CCM+ provides several models that you can select for each fluid film phase.

The following fluid film phase models are available:

- Space: Shell Three-Dimensional
- Material: Liquid and Multi-Component Liquid
- Fluid Film Flow: Flow Model
- Energy: Segregated Fluid Film Temperature or Coupled Temperature
- Viscous Regime: Laminar
- Equation of State: Constant Density and Polynomial Density
- Species: Segregated Species

Shell Three-Dimensional Space Model

The Shell Three Dimensional space model is similar to the Three Dimensional space model for a physics continuum. Activating this model lets you create a shell region from a region boundary.

Liquid and Multi-Component Liquid Material Models

The fluid film Liquid model and Multi-Component Liquid model are similar to the corresponding models for a physics continuum.

Segregated Fluid Film Temperature and Coupled Temperature Energy Models

These energy models solve the total energy equation for the fluid film phase, and also account for the energy that is transferred due to droplets being added or removed.

The Segregated Fluid Film Temperature model is similar to the Segregated Temperature model for a physics continuum. To make this model available in the fluid film phase, the following models must be activated in the physics continuum: Segregated Flow and one of Segregated Fluid Temperature or Segregated Fluid Enthalpy.

The Coupled Temperature model is similar to the Coupled Energy model for a physics continuum. To make this model available in the fluid film phase, the following models must be activated in the physics continuum: Coupled Flow and Coupled Energy.

Both of these energy models have the same properties.

Energy Model Properties

Secondary Gradients	Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces.	
	On	Include both secondary gradients.
	Off	Exclude both secondary gradients.
	Interior Only	Include the interior secondary gradients only.
	Boundaries Only	Include the boundary secondary gradients only.

Laminar Viscous Regime Model

The Laminar viscous regime model defines the flow within the film phase as being laminar, as opposed to being inviscid, or turbulent.

Equation of State Models

The equation of state for a fluid film phase can be constant density or polynomial density. See General Equation of State Models.

Segregated Species Model

The Segregated Species model solves the species continuity equations for a multi-component fluid mixture. This model is identical to the Segregated Species model in a physics continuum.

Contents:

Fluid Film Flow Model Reference

Fluid Film Flow Model Reference

The Fluid Film Flow model solves conservation equations for mass and momentum.

Film thickness comes from the mass equation. These equations are solved in a segregated manner, that is, one after the other. The energy equation, however, can either be solved segregated or as part of the coupled energy equation. A parabolic velocity profile is assumed in the direction normal to the wall.

Fluid Film Flow Model Reference

Provided By	**[film phase]** > **Models** > *Fluid Film Flow*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Fluid Film** > **Fluid Film Phases** > **[phase]** > **Models** > **Flow Model**	
Requires	A Fluid Film phase.	
Activates	Physics Models	Equation of State Viscous Regime Fluid Film Segregated Energy
	Initial Conditions	Fluid Film Thickness Velocity
Properties	See Flow Model Properties.	

Flow Model Properties

Maximum Film Thickness

Sets the maximum film thickness (h_f in Eqn. (546)) for the fluid film phase.

In some cases the film thickness can become very large. This behavior is typically physical (for example, the liquid film is pushed into a corner and it cannot escape). A very large film thickness (much larger than the thickness of the neighbor volume cell) can cause numerical instabilities in the simulation. Setting this property ensures that the film thickness does not exceed an appropriate maximum value at any location. The default value is 1.0 meters.

When the specified thickness is reached, any extra fluid is removed from the film and is lost from the simulation. In this scenario, mass conservation is not explicitly satisfied in the simulation.

Maximum Volume Fraction

Sets the maximum volume fraction ($\phi_{film,max}$ in Eqn. (555)) for the fluid film phase. The default is 0.

Setting this property allows you to include the effect of the liquid film on the void fraction. The volume that the film takes up is subtracted from the gas-filled cells next to the shell region.

To ensure stability, set the maximum volume fraction lower than 1 (for example, 0.75).

To deactivate the volume fraction effect, set this value to 0.

If the film forms because of drop impingement, set this value to 0.

Stabilised Film Thickness Equation

Activating the Stabilised Film Thickness Equation improves the ability of the Fluid Film solver to handle high mass impingement upon very small fluid film cells. In cases where the impinging droplets are large (that is, a few times bigger than the cell size), the sudden addition of fluid affects adjacent cells. The residuals can increase dramatically and cause convergence problems. Using this option prevents the

residuals from increasing out of control, keeping the simulation stable and allowing the solution to converge.

This option must be activated for steady simulations.

Fluid Film Phase Interactions

You specify the Fluid Film phase interaction model to control the interaction between the fluid film phase and another phase.

When the Fluid Film model is activated in the physics continuum, the phase interaction models that are appropriate for the physics continuum become available.

- **Film-Dispersed Phase Interaction**

  Controls the interaction between the fluid film and a dispersed phase, such as fluid droplets impinging on a surface film. This model is available for Dispersed Multiphase simulations only.

  The dispersed phase interacts with the fluid film phase by impingement of droplets from the dispersed phase into the fluid film and stripping of droplets from the fluid film into the dispersed phase. Only single-component fluid film is supported, as DMP uses a single-component dispersed phase.

  The Dispersed Multiphase model can be used as a lightweight replacement for Lagrangian particles in simulations where the LMP model is too computationally expensive due to the large number of particles. Wave and edge stripping transfers droplets from the fluid film phase into the dispersed phase of a DMP simulation. Using DMP particles for icing and soiling simulations is significantly quicker than using Lagrangian particles.
- **Film-Eulerian Phase Interaction**

  Controls the interaction of the fluid film with a specified Eulerian phase, such as a gas condensing onto or evaporating from a surface. This model applies to Multiphase Segregated Flow simulations only.

  This model supports mass transfer from a dispersed Eulerian phase into the fluid film domain. However, the reverse (film stripping, converting fluid film mass to the dispersed Eulerian phase) is not supported. Both single-component and multi-component Eulerian phases are supported. This model is used when the physics continuum contains multiple phases (typically a continuous phase and one or more dispersed phases), and allows you to specify the particular Eulerian phase with which the fluid film interacts.
- **Film-Lagrangian Phase Interaction**

  Controls the interaction between the fluid film and a Lagrangian phase, and is used for modeling droplet impingement or film stripping in a Lagrangian Multiphase simulation. This model is available for Lagrangian Multiphase simulations only.
- **Film-Physics Continuum Interaction**

  Controls the interaction between the fluid film and a single background phase, such as a gas condensing onto or evaporating from a surface. Both single-component and multi-component Fluid Film phases are supported. This model is available in most multiphase simulations. The exception is a Multiphase Segregated Flow simulation, in which you use the Film-Eulerian Phase Interaction model instead.
- **Film-VOF Phase Interaction**

Controls the interaction between the fluid film phase and a VOF phase. This model automatically uses the phase model (fluid film or VOF) that is most suitable for the local flow regime and the computer resources that are available. This model is available for Volume of Fluid (VOF) simulations only.

This phase interaction is not supported for **Steady** simulations.

The primary application of this phase interaction model is in simulations when a fluid film can accumulate in particular areas of the geometry to form pools. In those areas, the accumulated fluid is modeled as a VOF phase rather than as a thin fluid film.

Both single-component and multi-component fluid film phases are supported. For a multi-component material, you map the component pairs between the phases. Note that only a complete mixture mapping is supported.

See Resolved Fluid Film Model Reference.

Contents:

Film Phase Interaction Model Family Reference
Component Mapping Model Reference

Film Phase Interaction Model Family Reference

The Fluid Film phase interaction models control the interaction between the fluid film phase and another phase.

Film Phase Interaction Model Family Reference

Model Names	**Film-Dispersed Phase Interaction**	DMP
	Film-Eulerian Phase Interaction	EMP
	Film-Lagrangian Phase Interaction	LMP
	Film-Physics Continuum Interaction	PhysCon
	Film-VOF Phase Interaction	VOF
Provided By	**[phase interaction]** > **Models** > *Phase Interaction Topology*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Multiphase Interaction** > **Phase Interactions** > **[phase interaction]** > **Models** > **Film-VOF Phase Interaction**	
Requires	The appropriate type of Multiphase simulation (Dispersed, Segregated Flow, Lagrangian, or VOF) with the Fluid Film model activated in the physics continuum. A phase interaction must exist. For the Film-Physics Continuum Interaction, any type of Multiphase simulation, except Segregated Flow or VOF, is required.	
Properties	See Film Phase Interaction Model Family Properties.	
Activates	Phase Interaction Models	• Film-VOF Phase Interaction *Optional Models*: **Evaporation/Condensation**, **Resolved Fluid Film**, **Surface Tension**

		See Evaporation-Condensation Model Reference, Resolved Fluid Film Model Reference and Surface Tension Model Reference. • Film-Physics Continuum Interaction *Optional Models*: **Evaporation/Condensation**, **Boiling**, **Surface Tension** • Film-Lagrangian Phase Interaction and Film-Dispersed Phase Interaction *Optional Models*: **Impingement**, **Edge Stripping**, **Wave Stripping** • Film-Eulerian Phase Interaction *Optional Models*: **Inertial Impingement**
	Field Functions	Film-VOF Mass Transfer Rate of [Film-VOF Phase Interaction] Available when the Resolved Fluid Film model is activated in a Film-VOF Phase Interaction. See Resolved Fluid Film Field Functions.

Film Phase Interaction Model Family Properties

	DMP	EMP	LMP	PhysCon	VOF
Fluid Film Phase	✓	✓	✓	✓	✓
Dispersed Phase	✓				
Eulerian Phase		✓			
Lagrangian Phase			✓		
Physics Continuum				✓	
VOF Phase					✓

Fluid Film Phase

The fluid film continuum.

Dispersed Phase

The dispersed phase in the phase interaction.

Eulerian Phase

The Eulerian dispersed phase in the phase interaction.

Lagrangian Phase

The Lagrangian phase interacting with the fluid film phase.

Physics Continuum

The physics continuum that contains the phase interaction. (Read-only)

VOF Phase

The Volume of Fluid (VOF) phase in the phase interaction.

Component Mapping Model Reference

For phase interactions between multi-component phases, every component in the fluid film continuum must be mapped to its corresponding component in the Lagrangian, Eulerian, or VOF phase.

Component Mapping Model Reference

Provided By	**[phase interaction]** > **Models** > *Optional Models*
Example Node Path	**Multiphase Interaction** > **Phase Interactions** > **[phase interaction]** > **Models** > **Component Mapping**
Requires	A Multiphase simulation with the Fluid Film model activated. A Phase Interaction model defined with both the fluid film phase and the interacting phase being multi-component phases. A phase interaction model, such as Impingement, Wave Stripping, or Resolved Fluid Film, activated in the Phase Interaction.
Properties	See Component Mapping Model Properties.

Component Mapping Model Properties

Connectivity

Matches the fluid film components to their corresponding Lagrangian, VOF, or Eulerian dispersed phase components.

The following conditions apply:

- A complete mixture mapping is required. Every component in the fluid film must be mapped to the corresponding component in the liquid Lagrangian, VOF, or Eulerian phase.
- Both the fluid film and Lagrangian, VOF, or Eulerian phases must consist of the same material components. However, the ratios of the components can be different.

Modeling Impingement

STAR-CCM+ provides impingement models for Lagrangian phases, Dispersed Multiphase phases and Eulerian phases. Eulerian Phase Impingement is supported for **Implicit Unsteady** or **PISO Unsteady** simulations only; Lagrangian and Dispersed Multiphase impingement is additionally supported for **Steady** simulations.

Impingement of a Lagrangian Phase

For simulations where the Lagrangian Multiphase model is selected and fluid droplets are present, fluid films can be formed. These films form when the fluid droplets impinge the interface between a fluid film shell region and the fluid region.

For consistency, use the same materials for the fluid film and Lagrangian phases. Although differences in material properties are allowed between the phases, avoid these differences. They would lead to improper conservation of mass or enthalpy. If the densities of the phases differ, volume is not conserved.

Impingement of a Dispersed Phase

The Dispersed Multiphase model can be used as a lightweight replacement for Lagrangian particles in simulations where the Lagrangian Multiphase model is too computationally expensive due to the large number of particles. Impingement transfers droplets from the dispersed phase of a Dispersed Multiphase simulation into the fluid film phase. Using Dispersed Multiphase particles for icing and soiling simulations is significantly quicker than using Lagrangian particles. However, only single-component liquid phases are supported.

Impingement of a Eulerian Phase

For simulations where the Eulerian Multiphase model is selected and a liquid Eulerian phase is present, the Film-Eulerian Phase Interaction model supports droplet impingement from the Eulerian phase into a fluid film domain. Both single-component and multi-component liquid phases are supported.

You can specify the impingement efficiency, that is, the fraction of the total mass impinging on the fluid film boundary that is transferred from the Eulerian phase into the fluid film.

For more information on the Inertial Impingement model, see the Inertial Impingement Model Formulation.

Contents:

Setting Up Lagrangian Impingement
Setting Up Dispersed Multiphase Impingement
Impingement Model Reference
Setting Up Eulerian Impingement
Inertial Impingement Model Reference

Setting Up Lagrangian Impingement

To model droplet impingement in a Lagrangian Multiphase simulation, you set up a Film-Lagrangian phase interaction, activate the Impingement model, and specify the appropriate fluid-film boundary mode.

A Lagrangian Multiphase simulation with the Fluid Film model activated. Lagrangian phases and Fluid Film phases defined.

To model droplet impingement in a Lagrangian Multiphase simulation:

1. Set up the Lagrangian phases so that the Lagrangian droplets can exchange mass, momentum, and energy with the fluid film.
    - Activate the Two-Way Coupling model.

- Optionally, to model interactions between droplets and walls in greater detail, activate the Bai-Gosman or Satoh wall impingement models.

2. Create a multiphase interaction and select the Film-Lagrangian Phase Interaction model. On the model node, assign the correct Lagrangian phase and Fluid Film phase. Also activate the Impingement model.
3. If you want to model the transition from droplet to film in detail:
   a) On the Lagrangian phase, activate the Droplet Film Transition model.
   b) Set the *Droplet Transition Option* property to **Coverage Ratio** or **Equivalent Thickness**.
   c) Set the related value, (**Coverage Ratio** or **Equivalent Film Thickness**), under **[Lagrangian phase]** > **Boundary Conditions** > **Fluid-Film Boundary** > **Physics Values**.
4. Set up the Film-Lagrangian Phase Interaction:
   a) Specify the Lagrangian Phase and the Fluid Film Phase.
   b) If you are modeling a multi-component fluid film and a multi-component droplet, map the components between the two phases.

Set the Fluid-Film Boundary mode for each Lagrangian phase.

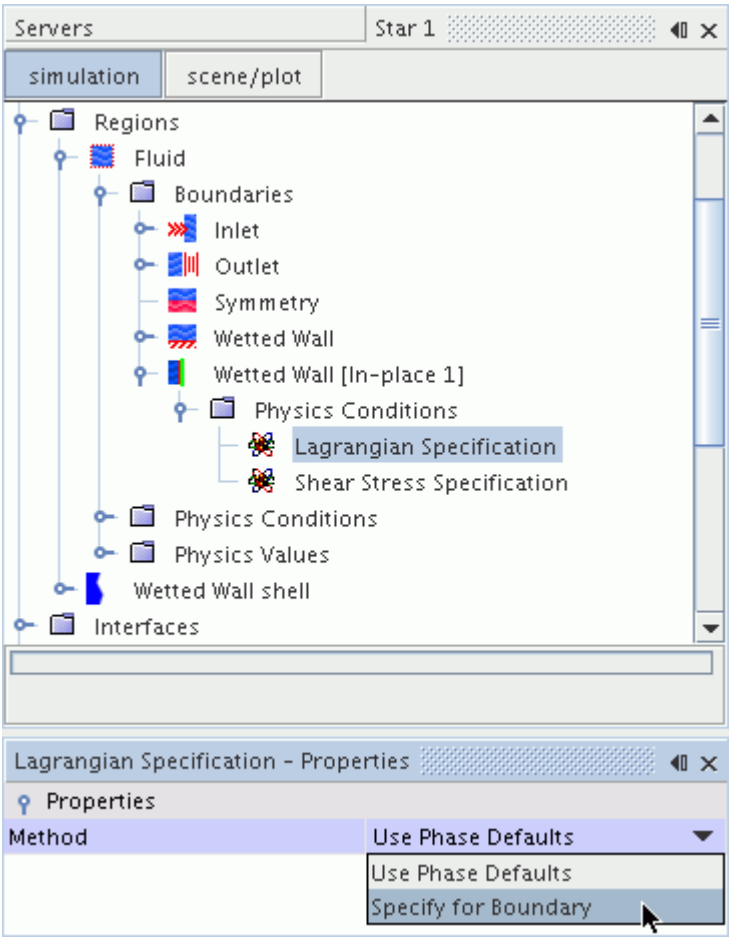
Within each participating Lagrangian droplet phase, set the boundary interaction mode for Fluid-Film boundaries to Fluid Film. This option transfers all Lagrangian droplets to the fluid film. Alternatively, you can use the Bai-Gosman or Satoh wall impingement models and set the boundary interaction mode accordingly. This method models the impingement in more detail by calculating which droplets rebound or stick.

To set the boundary interaction mode globally for all fluid film boundaries:

5. In the Lagrangian phase, select the **Boundary Conditions** > **Fluid-Film Boundary** > **Physics Conditions** > **Mode** node and set *Active Mode* to the appropriate option.
   - To transfer all of the Lagrangian droplets to the fluid film, select **Fluid Film**.
   - If you are using the Bai-Gosman or Satoh wall impingement models, select the **Bai-Gosman** or **Satoh** options respectively.

To set the boundary interaction mode locally for specific boundaries:

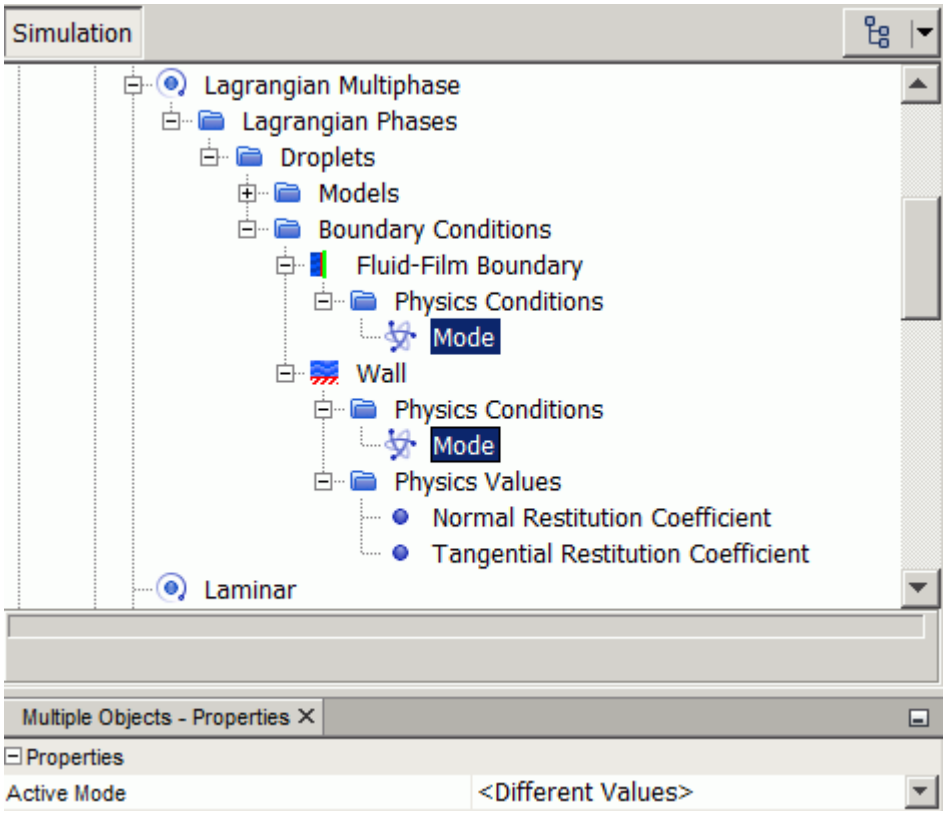
6. For the boundary, select the **Physics Conditions** > **Lagrangian Specification** node and set *Method* to **Specify for Boundary**.



7. Select the **Phase Conditions** > **[Phase]** > **Physics Conditions** > **Mode** node and set *Active Mode* to **Fluid Film** (or **Bai-Gosman** / **Satoh**, as appropriate).

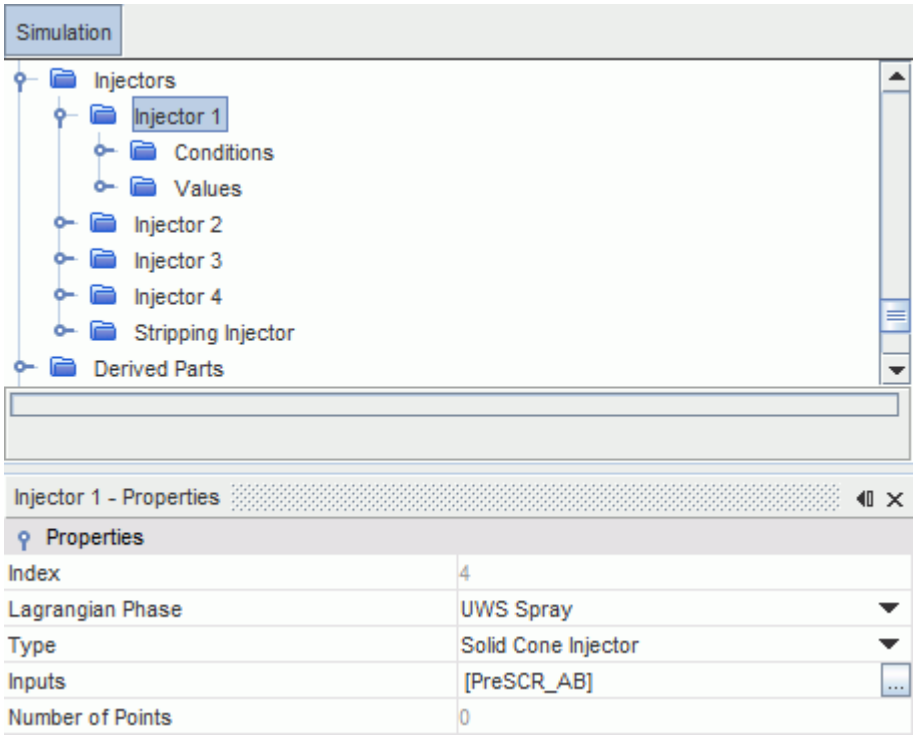
   The **Fluid Film**, **Bai-Gosman**, or **Satoh** options are available only when the Film-Lagrangian Phase Interaction is valid. Removing or adding models to the film phase or Lagrangian phase after setting a Lagrangian boundary condition can cause the Film-Lagrangian Phase Interaction to become temporarily invalid. In that case, the *Active Mode* defaults to **Rebound** without warning. Always check if the boundary mode is still set to **Fluid Film** when changing the phase models.

   All of the wall boundaries where fluid film is present use the mode that is specified for the **Fluid-Film Boundary** under the **Lagrangian Multiphase** > **Lagrangian Phases** > **[Droplet Phase]** > **Boundary Conditions** node. Any wall boundaries where no fluid film is present use the mode that is specified for the **Wall** under the same node.



8. Define the appropriate Lagrangian injectors. For each injector that you require:
   a) Create the injector.
   b) Specify the injector type.
   c) Specify the Lagrangian phase.
   d) Assign the input parts of the injector.
   e) Set the appropriate conditions and values.

   For more information, see Using Injectors.



Setting Up Dispersed Multiphase Impingement

To model droplet impingement in a Dispersed Multiphase simulation, you set up a Film-Dispersed phase interaction and activate the Impingement model.

A Dispersed Multiphase simulation with the Fluid Film model activated. DMP phases and Fluid Film phases defined.

To model droplet impingement in a Dispersed Multiphase simulation:

1. Set up a multiphase interaction and select the Film-Dispersed Phase Interaction model.
2. Set up the Film-Dispersed Phase Interaction:
   - Specify the Dispersed Phase and the Fluid Film Phase.
   - Activate the Impingement model.

Impingement Model Reference

The Impingement model is used to model the impingement of a Lagrangian phase or a dispersed DMP phase on to an interface to form a fluid film.

Impingement Model Reference

Provided By	**[phase interaction]** > **Models** > *Optional Models*

Example Node Path	**Continua** > **Physics 1** > **Models** > **Multiphase Interaction** > **Phase Interactions** > **[phase interaction]** > **Models** > **Impingement**	
Requires	For a Lagrangian Multiphase simulation, the Film-Lagrangian Phase Interaction model must be defined. For a Dispersed Multiphase simulation, the Film-Dispersed Phase Interaction model must be defined. The *Time* model can be **Implicit Unsteady**, **PISO Unsteady**, or **Steady**.	
Activates	Boundary Inputs	For a Lagrangian Multiphase simulation: • In the Lagrangian phase, **Boundary Conditions** > **Fluid-Film Boundary** > **Physics Conditions** > **Mode**. • For each boundary, **Physics Conditions** > **Lagrangian Specification**.

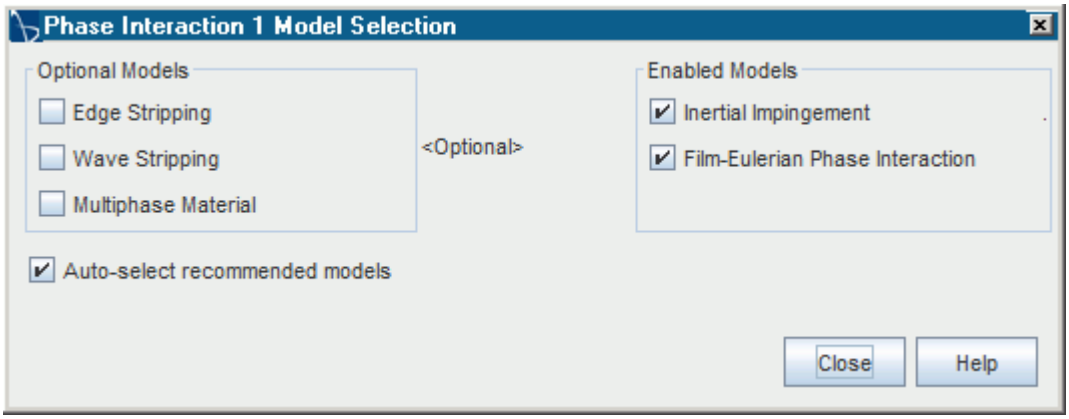
Setting Up Eulerian Impingement

To model droplet impingement in a Eulerian Multiphase simulation, you set up a Film-Eulerian phase interaction or a Film-Physics Continuum phase interaction. You then activate the Inertial Impingement model and specify the impingement efficiency at the appropriate boundary.

A Multiphase Segregated Flow simulation with the Fluid Film model activated. The relevant Eulerian phases and fluid film phases defined.

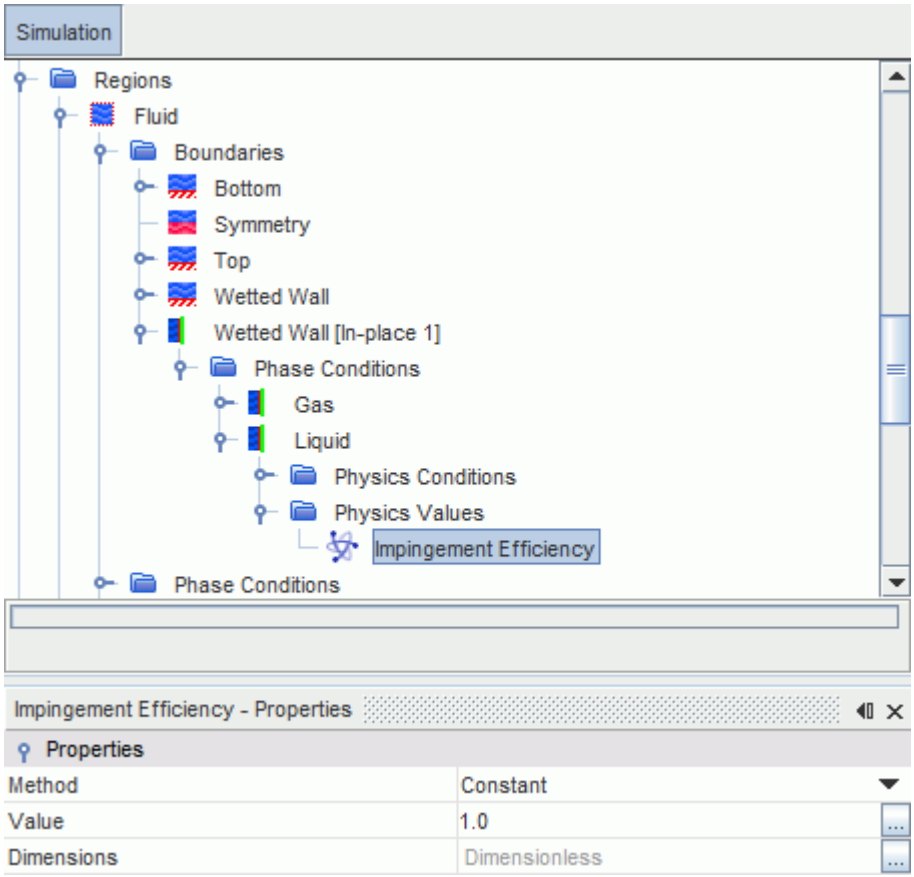
To model mass transfer due to inertial impingement on a surface:

1. Set up a multiphase interaction and select the Film-Eulerian Phase Interaction model.
2. Set up the Film-Eulerian Phase Interaction:
   a) Specify the Eulerian Phase and the Fluid Film Phase.
   b) Activate the Inertial Impingement model.



   c) If modeling a multi-component fluid film and a multi-component droplet, map the components between the two phases.
3. Specify the impingement efficiency of the liquid phase at the fluid-film interface.

You set the impingement efficiency at the **Boundaries** > **[fluid-film interface]** > **Phase Conditions** > **[eulerian phase]** > **Physics Values** > **Impingement Efficiency** node.



The default impingement efficiency is a constant value of 1.0, meaning all of the impinging liquid phase is added to the fluid film. However, if you have some understanding of the impingement efficiency in your specific case, you can use a field function to provide an appropriate expression. For example, you can use this setting to model different impingement conditions without changing the physics setup.

Inertial Impingement Model Reference

The Inertial Impingement model is used to model the impingement of a dispersed Eulerian phase on to an interface to form a fluid film.

Inertial Impingement Model Reference

Provided By	**[phase interaction]** > **Models** > **Optional Models**
Example Node Path	**[phase interaction]** > **Models** > **Inertial Impingement**

Requires	A Multiphase Segregated Flow simulation with the Film-Eulerian Phase Interaction model defined.	
Activates	Boundary Inputs	You set the impingement efficiency at the **Boundaries** > **[fluid-film interface]** > **Phase Conditions** > **[Eulerian phase]** > **Physics Values** > **Impingement Efficiency** node. See Boundary Values.

Boundary Values

Impingement Efficiency

The impingement efficiency represents the fraction of the liquid phase that hits the wall that is incorporated into the film. The default impingement efficiency is a constant value of 1.0 (all of the impinging liquid phase is added to the fluid film).

Modeling Film Stripping

In STAR-CCM+, you can model fluid being removed from a film due to wave stripping across the surface, or edge stripping on sharp corners. Both wave stripping and edge stripping are supported for **Implicit Unsteady**, **PISO Unsteady**, and **Steady** simulations.

Waves generate instabilities on the surface, and these instabilities form droplets which are stripped off the film. These droplets are entrained in the bulk fluid flow. Similarly, droplets are also stripped from the film when the fluid film flows over a sharp edge.

You can model film stripping in both Lagrangian Multiphase simulations and Dispersed Multiphase simulations. A particular fluid film phase can have only one edge stripping phase interaction and one wave stripping phase interaction.

Film Stripping in Lagrangian Multiphase Simulations

Droplets that enter the bulk flow are tracked using the Lagrangian Multiphase model in STAR-CCM+.

Setting up a film stripping simulation includes the same basic technique of setting up a fluid film shell region. Lagrangian features are added with the selection of Lagrangian Multiphase in the continuum, activating wave and/or edge stripping, and using film stripping injectors.

It is possible for droplets to form on the surface of a fluid film, and to be immediately forced back into the film on the subsequent time step.

Film Stripping in Dispersed Multiphase Simulations

The Dispersed Multiphase model can be used as a lightweight replacement for Lagrangian particles in simulations where the Lagrangian Multiphase model is too computationally expensive due to the large number of particles. Wave and edge stripping transfers droplets from the fluid film phase into the dispersed phase of a Dispersed Multiphase simulation. Using Dispersed Multiphase particles for icing and soiling simulations is significantly quicker than using Lagrangian particles.

The Wave Stripping and Edge Stripping models do not compute the size of the stripped droplets that are added to the dispersed phase: all of the droplets produced have the same user-defined diameter. Each

stripping model contributes to one dispersed phase only. However, it is possible to have edge stripping to one dispersed phase and wave stripping to a different dispersed phase.

Only single-component fluid film is supported, as Dispersed Multiphase uses a single-component dispersed phase.

Contents:

Setting Up Film Stripping to a Lagrangian Phase
Setting Up Film Stripping to a Dispersed Phase
Wave Stripping Model Reference
Edge Stripping Model Reference

Setting Up Film Stripping to a Lagrangian Phase

To model film stripping in a Lagrangian Multiphase simulation, you set up a Film-Lagrangian phase interaction, activate the Wave Stripping model or Edge Stripping model, and specify the film stripping injector.

Make sure that you have already defined:

- Implicit Unsteady, PISO Unsteady, or Steady time model
- A Lagrangian Multiphase simulation with the Fluid Film model activated.
- Lagrangian phases and Fluid Film phases.

When you set up Lagrangian stripping, you create a Lagrangian injector and (for impingement) specify two-way coupling.

To model film stripping in a Lagrangian Multiphase simulation:

1. Activate the Two-Way Coupling model so that the Lagrangian droplets can exchange mass, momentum, and energy with the fluid film.
2. Set up a multiphase interaction and select the Film-Lagrangian Phase Interaction model.
3. Set up the Film-Lagrangian Phase Interaction:
   - Specify the Lagrangian Phase and the Fluid Film Phase.
   - Activate either or both the Wave Stripping model and the Edge Stripping model.
   - If you are modeling a multi-component fluid film and a multi-component droplet, map the components between the two phases.
4. Set the Wave Stripping and Edge Stripping property values according to your simulation.

A Film Stripping Injector is used to identify a film surface from which droplets leave the film and are injected into the bulk flow.

5. Create a Lagrangian injector, and set the following properties:

Property	Value
Type	**Film Stripping**
Inputs	Specify the entities which are used to generate inject points. In this case, select the Fluid-Film Interface boundary on the original fluid region.

Property	Value
Lagrangian Phase	The appropriate Lagrangian phase.

6. Set the injector Time Randomization condition and the injector Parcel Streams value.

Note: If a film stripping injector is not activated for the Lagrangian phase, the film stripping models are still active. In this scenario, a warning message similar to the following is displayed:

```
Warning: Stripping injector [Film Stripping Injector] is inactive
as no Lagrangian Phase has been selected. The Film Stripping
model for [Film] will strip into void.
```

The stripping on the fluid film side is calculated but the stripped mass is not transferred into the Lagrangian phase. This feature can be useful in cases where you expect the stripped mass to make a negligible contribution to the relevant transport processes.

Setting Up Film Stripping to a Dispersed Phase

Dispersed Multiphase wave and edge stripping can be used in simulations where using a dispersed phase with a single droplet diameter is a good enough approximation to resolve the particle flow.

Make sure that you have defined:

- An Implicit Unsteady, PISO Unsteady, or Steady simulation.
- A Dispersed Multiphase simulation with the Fluid Film model activated.
- The dispersed phase and fluid film phase defined. Both phases must be a single-component liquid: multi-component phases are not supported.

Dispersed Multiphase stripping is intended as a simple replacement for Lagrangian stripping, so the property settings are nearly identical. The only difference is that the settings for computing the size of Lagrangian particles are not used for Dispersed Multiphase. The computation of mass, momentum, and energy stripped from the fluid film phase to the Dispersed Multiphase phase is exactly the same as in the Lagrangian case.

The Wave Stripping and Edge Stripping models do not compute the size of the stripped droplets that are added to the dispersed phase: all of the droplets produced have the same user-defined diameter. Each stripping model contributes to one dispersed phase only. However, it is possible to have edge stripping to one dispersed phase and wave stripping to a different dispersed phase.

Note: When you set up Lagrangian stripping, you create a Lagrangian injector and specify two-way coupling. Dispersed Multiphase stripping does not have any corresponding procedures.

1. Set up a multiphase interaction and select the Film-Dispersed Phase Interaction model.
2. Set up the Film-Dispersed Phase Interaction:
   - Specify the Dispersed Phase and the Fluid Film Phase.
   - Activate either or both the Wave Stripping model and the Edge Stripping model.
3. Set the Wave Stripping and Edge Stripping property values according to your simulation.

   Ensure that you specify an appropriate value for the Maximum Volume Fraction in the Dispersed Phase Volume Fraction Solver. If the value is lower than the stripping rate, the stripped mass is truncated and mass is lost from the simulation.

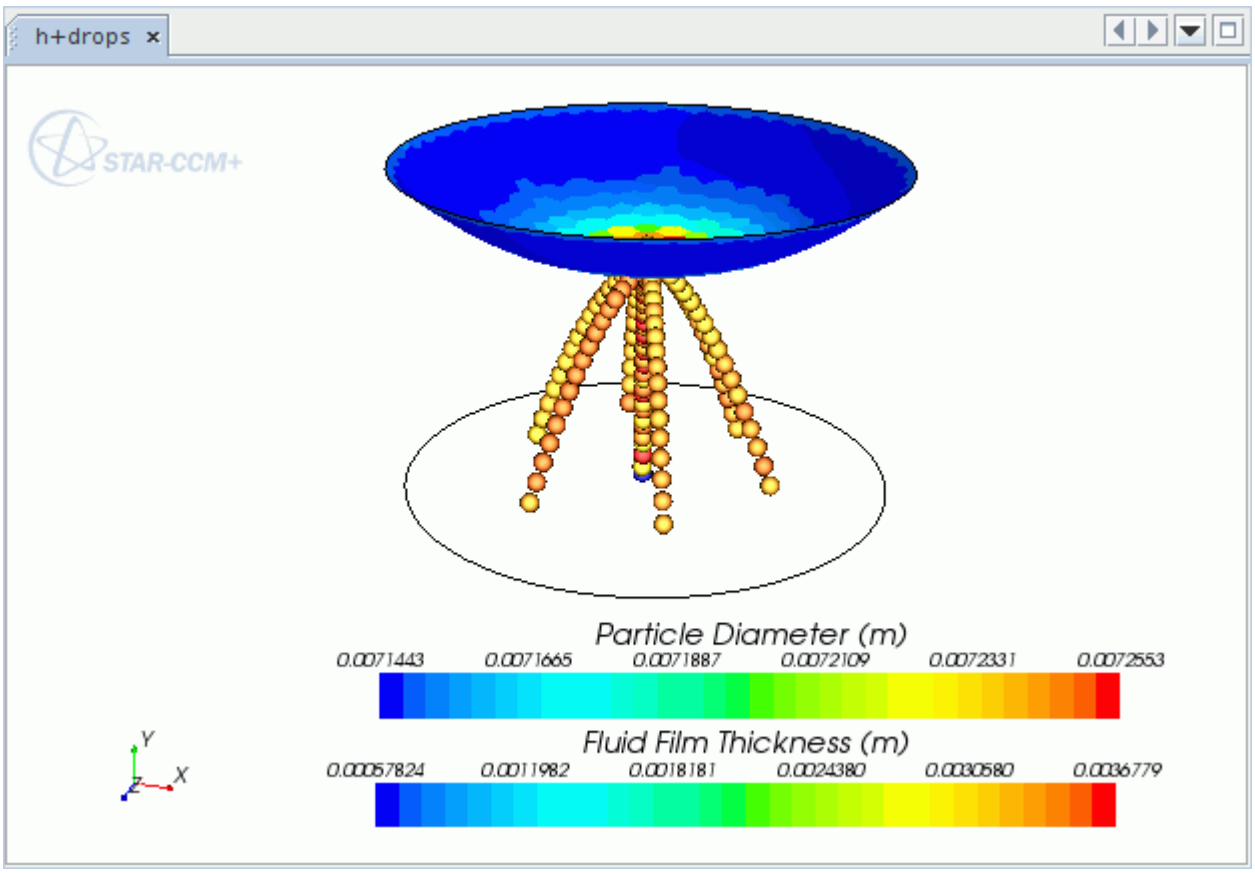
Wave Stripping Model Reference

Wave film stripping models the ejection of droplets from the fluid film as a result of wave-induced instabilities.

The theory and methodology for wave film stripping is provided in detail in the Formulation section.

The Wave Stripping model assumes that sinusoidal perturbations disturb the flat film surface, and that the most unstable wavelength grows linearly. In some cases, these assumptions are not satisfied: for example, when a film drips from underneath a horizontal surface or ceiling.

Consider the following example: a case with a “hanging film” building up on a ceiling. One would intuitively expect the stripping speed to increase as the film grows thicker, eventually reaching some equilibrium thickness.



To obtain correct results, you can tune the model with a validation study. You can set the following parameters:

- c_H to control the thickness threshold at which stripping occurs
- c_D to control the size of the droplets that result from stripping
- c_B to control the rate at which film mass is stripped

Wave Stripping Model Reference

Provided By	**[phase interaction]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Multiphase Interaction** > **Phase Interactions** > **[phase interaction]** > **Models** > **Wave Stripping**	
Requires	A Lagrangian Multiphase or Dispersed Multiphase simulation with the Fluid Film model activated. A Film-Lagrangian Phase Interaction model or Film-Dispersed Phase Interaction model defined with both the fluid film phase and the interacting phase specified. The *Time* model can be **Implicit Unsteady**, **PISO Unsteady**, or **Steady**. A fluid film phase can have only one wave stripping phase interaction.	
Activates	Field Functions	See Wave Stripping Model Field Functions.
Properties	See Wave Stripping Model Properties.	

Wave Stripping Model Properties

cH

The coefficient (c_H in Eqn. (562)) for minimum film thickness to initiate stripping. The default setting is 1.0. Lowering this value makes wave stripping occur at a lower film thickness threshold.

This property applies to both Lagrangian Multiphase and Dispersed Multiphase edge stripping.

cD

The droplet diameter coefficient (c_D in Eqn. (564)). The default setting is 3.78. Increasing this number leads to bigger particles (but the total amount of mass that is stripped remains the same).

This property applies to Lagrangian Multiphase edge stripping only. In Dispersed Multiphase wave stripping, the droplet diameter is specified in the dispersed phase rather than being calculated.

cB

The breakup rate scaling coefficient (c_B in Eqn. (566)). The default setting is 1.0. Increasing this number increases the rate of wave stripping (more particles per time-step).

This property applies to both Lagrangian Multiphase and Dispersed Multiphase edge stripping.

Gas Velocity Scaling Factor

The constant that empirically relates the shear velocity to the free stream velocity ($K_{*,\infty}$ in Eqn. (568)). The default setting is 10.0.

This property applies to both Lagrangian Multiphase and Dispersed Multiphase edge stripping.

Wave Stripping Model Field Functions

The following primitive field functions are made available when the *Temporary Storage Retained* expert property of the Fluid Film Segregated Flow solver is activated. They allow you to see whether any stripping occurs and how fast.

Film Wave Stripping Minimum Thickness	The minimum thickness that allows stripping, $h_{\min}$ in Eqn. (562).
Film Wave Stripping Time Scale	The time scale (t_b in Eqn. (565)) of wave stripping, where the film thickness exceeds $h_{\min}$.

Edge Stripping Model Reference

Edge film stripping models the ejection of droplets from the fluid film when it flows over a sharp edge.

The theory and methodology for edge film stripping is provided in the Edge Stripping Formulation.

Edge Stripping Model Reference

Provided By	**[phase interaction]** > **Models** > *Optional Models*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Multiphase Interaction** > **Phase Interactions** > **[phase interaction]** > **Models** > **Edge Stripping**
Requires	A Lagrangian Multiphase or Dispersed Multiphase simulation with the Fluid Film model activated. A Film-Lagrangian Phase Interaction model or Film-Dispersed Phase Interaction model defined with both the fluid film phase and the interacting phase specified. The *Time* model can be **Implicit Unsteady**, **PISO Unsteady**, or **Steady**. A fluid film phase can have only one edge stripping phase interaction.
Properties	See Edge Stripping Model Properties.

Edge Stripping Model Properties

In Dispersed Multiphase edge stripping, the droplet diameter is specified in the dispersed phase rather than being calculated. The properties that relate to droplet size calculations apply to Lagrangian Multiphase edge stripping only.

Minimum Corner Angle

Defines the minimum sharp edge angle to which the edge film stripping model is applied. This angle allows you to define which edges cause stripping, and which edges are due to mesh discretization.

This property applies to both Lagrangian Multiphase and Dispersed Multiphase edge stripping.

q

A shape factor that is used in the Rosin-Rammler droplet size distribution (see Eqn. (579)). Reducing the value of *q* generates statistically smaller droplets.

This property applies to Lagrangian Multiphase edge stripping only.

Critical FR

The force ratio FR that must be exceeded to initiate edge stripping. This value is essentially the ratio of the film momentum flux to the surface tension and gravity force and is used in Eqn. (569) and Eqn. (575). Reducing the critical force ratio increases the occurrence of edge stripping.

This property applies to both Lagrangian Multiphase and Dispersed Multiphase edge stripping.

c1

The droplet diameter scale factor, which is used to scale the parent droplet size (see Eqn. (576)). Reducing this value reduces the droplet size.

This property applies to Lagrangian Multiphase edge stripping only.

Limit Parent Droplet Diameter

Specifies whether the maximum parent droplet diameter is limited.

This property applies to Lagrangian Multiphase edge stripping only.

Maximum Diameter

Sets the maximum parent droplet diameter. This limit is only applied if the *Limit parent droplet diameter* property is activated.

This property applies to Lagrangian Multiphase edge stripping only.

Randomized Droplet Diameter

Specifies whether the droplet diameter is calculated or randomized. Randomized droplet diameters use the Rosin-Rammler size distribution (see Eqn. (579)). Calculated droplet diameters are the average of the size distribution (see Eqn. (581)).

This property applies to Lagrangian Multiphase edge stripping only.

Modeling Fluid Film Evaporation and Condensation

STAR-CCM+ provides an evaporation and condensation model for fluid films. This model is a phase interaction model between a multi-component liquid film and a multi-component gas.

The fluid film evaporation and condensation model is similar to the VOF evaporation and condensation model. This model is available in a Film-Eulerian Phase Interaction (for a Multiphase Segregated Flow simulation), in a Film-VOF Phase Interaction (for a hybrid Film-VOF simulation), and in a Film-Physics Continuum Interaction (for all other simulation types).

Contents:

Setting Up Fluid Film Evaporation and Condensation
Evaporation-Condensation Model Reference

Setting Up Fluid Film Evaporation and Condensation

To model fluid film evaporation and condensation, you set up the appropriate phase interaction model, activate the fluid film Evaporation-Condensation model, and set the property values according to your simulation.

Before you begin, you need the following:

- A simulation with the Fluid Film model activated.
- The fluid film phase defined as a multi-component liquid.
- The corresponding Eulerian phase (for the Multiphase Segregated Flow model), VOF phase, or the physics continuum defined as a multi-component gas.

To model fluid film evaporation and condensation:

1. Select the physics models that are required for a fluid film simulation. Include the following models:
   - Multi-Component Gas material model in the physics continuum.
   - Multiphase Interaction model in the physics continuum.
   - If you want to model the effect of evaporation and condensation on the heat balance, select either Coupled Energy or one of the Segregated Fluid Energy models.
2. Set up a multiphase interaction and select one of the following phase interaction models:
   - For a Multiphase Segregated Flow simulation, you use the Film-Eulerian Phase Interaction model.
   - For a hybrid Film-VOF simulation, you use the Film-VOF Phase Interaction.
   - For all other simulations, you use the Film-Physics Continuum Interaction model.
3. Set up the phase interaction:
   - Specify the Fluid Film Phase and the corresponding Eulerian Phase or VOF Phase.
   - Activate the Component Mapping model.
   - Activate the Evaporation-Condensation model.
4. In the Component Mapping node, map the components of the two phases.
5. In the Evaporation-Condensation node, set the property values according to your simulation.

Evaporation-Condensation Model Reference

In the Fluid Film model, the evaporation and condensation occurs on the interface between the fluid film and the bulk continuum.

This model is available in a Film-Eulerian Phase Interaction (for a Multiphase Segregated Flow simulation), in a Film-VOF Phase Interaction (for a hybrid Film-VOF simulation), and in a Film-Physics Continuum Interaction (for all other simulation types).

The Fluid Film Evaporation and Condensation model is implemented as a diffusion limited model. For that reason, it is not directly applicable to regimes where energy transport dominates the phase exchange. For example, during condensation of pure vapor or when evaporation takes place close to boiling point. In such saturated regimes, you can switch to using the thermally limited evaporation and condensation model.

Evaporation-Condensation Model Reference

Theory	See Evaporation and Condensation	
Provided By	**[phase interaction]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Multiphase Interaction** > **Phase Interactions** > **[phase interaction]** > **Models** > **Evaporation-Condensation**	
Requires	Any Multiphase simulation with the Fluid Film model activated. A Film-Eulerian Phase Interaction model, a Film-VOF Phase Interaction model, or a Film-Physics Continuum Interaction model defined with both the fluid film phase and the corresponding Eulerian phase, VOF phase, or physics continuum specified.	
Activates	Field Functions	See Evaporation-Condensation Model Field Functions.
Properties	See Evaporation-Condensation Model Properties.	

Evaporation-Condensation Model Properties

Implicit Coupling

When this property is deactivated, STAR-CCM+ uses the hydrodynamically limited evaporation and condensation model. This model is the default, and is used for non-saturated regimes.

When this property is activated, STAR-CCM+ uses the thermally limited implementation of the evaporation and condensation model. This model is used for saturated regimes, for example, during condensation of pure vapor or when evaporation takes place close to boiling point.

For more information, see Evaporation and Condensation Solution Procedure.

Under-Relaxation Factor

At each iteration, this property governs the extent to which the newly computed solution supplants the old solution.

Saturation Threshold

The saturation ratio that is used in Eqn. (607) to determine whether saturation occurs. This setting lets you adjust when the fluid film evaporation and condensation model switches into saturation mode.

N_seeds

The nucleation density for condensation droplets. This value is N in Eqn. (611).

For more information, see Steam Condensation at a Dry Wall.

R_seeds

The minimum radius of condensation droplets. This value is $R_{\min}$ in Eqn. (611).

For more information, see Steam Condensation at a Dry Wall.

Linearize Film Energy

When activated, the effect of phase exchange is linearized in the film energy equation.

Linearize Gas Energy

When activated, the effect of phase exchange is linearized in the gas energy equation.

Condensation

When activated, condensation is included in the phase interaction.

Evaporation

When activated, evaporation is included in the phase interaction.

Evaporation-Condensation Model Field Functions

The following primitive field functions are made available when the Fluid Film Evaporation and Condensation model is activated. They allow you to monitor how fast evaporation or condensation occurs.

Film Evaporation Latent Heat Flux	The latent heat due to evaporation (W/m^2) that is injected at the interface.
Film Evaporation Rate	The rate of evaporation in kg/m^2-s for every species.
Film Evaporation Mass Fraction	The saturated vapor mass fraction on the gas side of the interface, for every interacting component.

Modeling Fluid Film Boiling

The fluid film boiling model assumes that boiling takes place either on hot film walls or inside film cells. Boiling at hot walls increases the heat flux between the film and the wall.

The mechanisms causing this effect are:

- Evaporation (latent heat) transport

  Heat is supplied near the wall to bubbles which then move away into the liquid film. In subcooled boiling, simultaneous evaporation at the base of a bubble and condensation at its tip can occur; see Using the Leidenfrost Temperature.
- Micro-convection

  Growth and collapse of bubbles cause random liquid motion.
- Vapor-liquid exchange

  A Reynolds analogy model in which bubble growth causes an exchange of liquid between the wall and bulk regions.
- Surface quenching

  A variation of the vapor-liquid exchange, assuming transient conduction to the cold liquid in contact with the wall after bubble departure.
- Wake flow

  Liquid motion behind a departing bubble causes convection from the wall.
- Thermo-capillary flow

Small variations in surface tension due to temperature differences between the base and tip of a bubble cause a jet of hot liquid to flow away from the wall.

Troubleshooting Fluid Film Boiling Simulations

The following error message can appear:

Error: FilmEvaporationModel: Detect negative latent heat of boiling

The latent heat is the difference between the total enthalpy of the liquid (film) and gas phase. When this error message appears, you can evaluate the enthalpy of the film and gas phase with a contour plot to check for errors.

In addition, this error message can occur when you have incorrectly defined the Heat of Formation or Specific Heat material properties for the film or gas phases.

Contents:

Setting Up Fluid Film Boiling
Fluid Film Boiling Model Reference

Setting Up Fluid Film Boiling

To model fluid film boiling, you set up the appropriate phase interaction model, activate the fluid film Boiling model, and set the property values according to your simulation.

A Multiphase simulation with the Fluid Film model activated.

To set up a Fluid Film Boiling simulation:

1. Select the physics models that are required for a fluid film simulation. Include the following models:
   - Multi-Component Gas material model in the physics continuum.
   - Multiphase Interaction model in the physics continuum.
   - If you want to model the effect of boiling on the heat balance, select either Coupled Energy or one of the Segregated Fluid Energy models.
2. Set up a multiphase interaction and select the appropriate phase interaction model.

   For a Multiphase Segregated Flow simulation, you use the Film-Eulerian Phase Interaction model. For all other simulations, you use the Film-Physics Continuum Interaction model.
3. Set up the Film-Eulerian Phase Interaction or Film-Physics Continuum Interaction:
   - Specify the Fluid Film phase and, if necessary, the Eulerian phase.
   - Activate the Boiling model.
   - If you are modeling a multi-component fluid film and a multi-component droplet, map the components between the two phases.
4. Set the Boiling property values according to your simulation.

Fluid Film Boiling Model Reference

The multi-component liquid film boiling model models vaporization above boiling temperature. This model assumes that boiling takes place either on hot film walls or inside film cells. Boiling at hot walls increases the heat flux between the film and the wall.

For the mathematical formulation of the fluid film boiling model, see Boiling.

Fluid Film Boiling Model Reference

Provided By	**[phase interaction]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Multiphase Interaction** > **Phase Interactions** > **[phase interaction]** > **Models** > **Boiling**	
Requires	Any Multiphase simulation with the Fluid Film model activated. A Film-Eulerian Phase Interaction model or Film-Physics Continuum Interaction model defined with both the fluid film phase and the Eulerian phase or physics continuum specified.	
Activates	Model Controls (child nodes)	HTCxArea
	Field Functions	See Fluid Film Boiling Model Field Functions.
Properties	See Fluid Film Boiling Model Properties.	

Fluid Film Boiling Model Properties

Boiling Model Parameter c_sf

Characterizes the fluid wall interface. c_{sf} in Eqn. (615).

n

The Prandtl Number Exponent. n in Eqn. (615).

Boiling Model Parameter c_s

Parameter that is used to compute the wall heat flux. c_s in Eqn. (621).

Constant of Critical Heat Flux

c_{max} in Eqn. (616).

Constant of Minimal Heat Flux

c_{min} in Eqn. (617).

Wall Boiling Evaporation Factor c_e

The wall boiling evaporation c_e factor that determines how much of the boiling heat flux generates vapor. See Eqn. (623).

Fluid Film Boiling Model Field Functions

The following primitive field functions are made available when the Fluid Film Boiling model is activated.

Film Boiling Latent Heat	The latent heat of phase transfer due to boiling (J/kg).
Film Boiling Latent Heat Flux	The total power (W/m^2) that is spent during the phase change due to wall boiling. This field function is available on both gas and liquid sides.
Film Boiling Rate	The evaporation rate due to wall boiling (kg/m^2-s) for every interacting component. This field function is available on both gas and liquid sides.
Film Boiling Temperature	The boiling temperature (K) of the liquid mixture.
Film Bulk Boiling Latent Heat Flux	The total power (W/m^2) that is spent during the phase change due to boiling in film cells. This field function is available on both gas and liquid sides.
Film Bulk Boiling Rate	The evaporation rate due to boiling inside the film cell (kg/m^2-s) for every interacting component. This field function is available on both gas and liquid sides.
Film Boiling Flag	Indicates if the conditions for boiling are met in the film $T > T_{sat}$

HTCxArea

The HTCxArea node represents the heat transfer coefficient (W/m^2K) between the vapor bubbles and the surrounding liquid, which is multiplied by the specific contact area (contact area per unit volume, m^2/m^3) between the two.

Modeling Fluid Film Melting and Solidification

Melting is the process which changes the state of matter from solid to liquid. It takes place when a solid substance is heated to the melting temperature. The opposite process is called solidification.

In STAR-CCM+ you do not need to specify a solid material for this process. You choose a liquid material, and then specify the temperature values at which the material melts and solidifies. The Relative Solid Volume Fraction field function indicates the state of the fluid film.

A major application of the Melting-Solidification model for fluid films is airfoil icing. An important aspect of airfoil icing is run-back: the flow of liquid water over the layer of recently solidified ice. In airfoil icing simulations, it is important to represent the geometry of the airfoil as a combination of the airfoil and the ice layer, as the geometry has a significant effect on the flow field. Accurately modeling the flow field is essential for predicting aerodynamic characteristics such as lift and drag, and also for modeling the self-reinforcing impact on the ice accretion rate itself.

To ensure that the geometry is accurately represented, you can use the Morpher to adjust the surface of the solidified volume when the solidified mass is removed.

Contents:

Setting Up Fluid Film Melting and Solidification
Setting up Solidified Mass Removal
Melting-Solidification Model Reference

Setting Up Fluid Film Melting and Solidification

To model fluid film melting and solidification, you activate the Fluid Film model in the physics continuum, activate the Melting-Solidification model in the fluid film phase, and then set the Melting-Solidification model property values according to your simulation. You can also configure the model so that the solidified mass is removed from the fluid film.

To set up a fluid film melting and solidification simulation:

1. In the physics continuum, activate the Fluid Film model.
2. Activate the appropriate energy model:
   - For a Coupled Flow simulation, **Coupled Energy**.
   - For a Segregated Flow simulation, either **Segregated Fluid Temperature**, **Segregated Fluid Isothermal**, or **Segregated Fluid Enthalpy**.
3. In the **Models** > **Fluid Film** > **Fluid Film Phases** node, create the fluid film phase.

   The liquid film can be a single-component liquid or a multi-component liquid.
4. In the new fluid film phase, activate the Melting-Solidification model.
5. If necessary, configure the simulation so that the solidified mass is removed from the fluid film.

Setting up Solidified Mass Removal

By default, the Melting-Solidification model treats the fluid film as a homogeneous mixture of solidified mass and liquid. A Relative Solid Volume Fraction is used to represent the amount of solid. You can configure the model so that the solidified mass is removed from the fluid film: all of the mass that remains in the fluid film is liquid.

A major application of the Melting-Solidification model for fluid films is airfoil icing. An important aspect of airfoil icing is run-back: the flow of liquid water over the layer of recently solidified ice. In airfoil icing simulations, it is important to represent the geometry of the airfoil as a combination of the airfoil and the ice layer, as the geometry has a significant effect on the flow field. Accurately modeling the flow field is essential for predicting aerodynamic characteristics such as lift and drag, and also for modeling the self-reinforcing impact on the ice accretion rate itself.

To ensure that the geometry is accurately represented, you can use the Morpher to adjust the surface of the solidified volume when the solidified mass is removed. If the Morpher is activated and configured, the wall morphs to model the surface of the solidified volume. You can specify the density of the solid; STAR-CCM+ uses the specified density to compute the morphing that is applied to the surface of the solidified mass. If the

Morpher is not activated, or is not correctly configured, the solidified mass is removed with no effect on the geometry of the wall.

To set up solidified mass removal:

1. In the *Melting-Solidification Properties* window, set the *Solidified Film Option* to **Remove Solidified Film**.

   The **Solid Density** node appears under the **Melting-Solidification** node. If you are using the Morpher to model the surface of the solidified mass, you can specify the density of the solid.
2. Select the **Solid Density** node and select the Method to use for specifying the density of the solid.
3. Specify the appropriate constant value, or select the appropriate field function.

To activate and configure the Morpher:

4. Activate the Morpher.

   To set up a Morphing motion, right-click the **Tools** > **Motions** node and click **New** > **Morphing**.
5. Apply the Morphing motion to the fluid region.

   Under the **Regions** > **[Fluid]** > **Physics Values** > **Motion Specification** node, set the *Motion* to **Morphing**.
6. At the interface between the fluid film and the gas (on the gas side), configure the Morpher to allow STAR-CCM+ internal models to add to the displacement.

   Under the **Regions** > **[Fluid]** > **Boundaries** > **[Boundary Name]** > **Physics Conditions** > **Morpher** node, set the *Method* to **Internal**.
7. In the *Melting-Solidification Properties* window, set the *Smoothing Steps* property to the appropriate value.

   This property affects the smoothing of the morphed surface that is used to model the surface of the solidified mass. At each iteration, the smoothing process is repeated the specified number of times.

Melting-Solidification Model Reference

STAR-CCM+ provides a melting and solidification model for fluid films. The fluid film melting and solidification model is similar to the VOF melting and solidification model.

For the mathematical formulation of the fluid film melting and solidification model, see Melting and Solidification.

Melting-Solidification Model Reference

Provided By	**[fluid film phase]** > **Models** > *Optional Models*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Fluid Film** > **[fluid film phase]** > **Models** > **Melting-Solidification**
Requires	The Fluid Film model activated in the physics continuum. The appropriate energy model activated: • For a Coupled Flow simulation, **Coupled Energy**.

	• For a Segregated Flow simulation, either **Segregated Fluid Temperature**, **Segregated Fluid Isothermal**, or **Segregated Fluid Enthalpy**. In the fluid film *Phase Model Selection* dialog: *Space*: **Shell Three Dimensional** *Fluid Film Flow*: **Flow Model** *Material*: **Liquid** or **Multi-Component Liquid** *Fluid Film Energy*: For a Coupled Flow simulation, **Coupled Temperature**; For a Segregated Flow simulation, **Segregated Fluid Film Temperature** *Optional Models*: **Melting-Solidification**	
Activates	Model Controls (child nodes)	Solid Density This node appears when the *Solidified Film Option* property is set to **Remove Solidified Film**.
	Material Properties	• Fraction Solid Curve • Latent Heat of Fusion • Liquidus Temperature • Solidus Temperature It is good practice to use a field function for the viscosity and thermal conductivity of the film, since the values of these properties differ for liquid and solid states. Make these properties a function of either relative solid volume fraction or temperature.
	Field Functions	See Melting-Solidification Model Field Functions.
Properties	See Melting-Solidification Model Properties.	

Melting-Solidification Model Properties

Solidified Film Option

Specifies how the solidified mass is treated in the simulation. When **Remove Solidified Film** is selected, the Solid Density node is activated and the solidified mass is removed. All of the mass that remains in the fluid film is liquid. Alternatively, the fluid film is treated as a homogeneous mixture of solidified mass and liquid.

Smoothing Steps

If the Morpher is enabled, this property affects the smoothing of the morphed surface that is used to model the surface of the solidified mass. At each iteration, the smoothing process is repeated the specified number of times.

Solid Time-step Factor

If the Morpher is enabled, this property scales the morphing speed of the morphed surface that is used to model the surface of the solidified mass. In practical solidification simulations, the morphing of the solidified mass front is typically so small that it does not noticeably influence the solution at each time step. This setting lets you speed up the calculation by the specified factor. For example, by setting this factor to 10, and reducing the Maximum Physical Time by the same factor, you can obtain (to good approximation) the same solidified mass shape, but approximately 10 times faster.

Linearize Melting

When activated, this property linearizes the influence of latent heat.

Under-Relaxation Factor

At each iteration, this property governs the extent to which the newly computed solution supplants the old solution.

Melting-Solidification Model Field Functions

The following primitive field function is made available when the Fluid Film Melting-Solidification model is activated.

Relative Solid Volume Fraction	The volume fraction of the fluid film that is in the solid state.

Solid Density Properties

Method

Specifies the density of the solid. The default is **Liquid**, which sets the density of the solid to be identical to the density of the liquid. Alternatively, you can specify the solid density as constant or field function.

Modeling Fluid Film Surface Tension

At the interface between a fluid film and the gas phase, surface tension is caused by the greater attraction of fluid molecules to each other than to the gas molecules. The net effect is an inward force at the interface that causes the fluid film to behave as if its surface were covered with a stretched elastic membrane: that is, the surface is under tension. This tension is expressed using the experimentally determined surface tension coefficient σ.

The magnitude of the surface tension between a given fluid and a given gas depends on the nature of the fluid and gas that are in contact and on the temperature. The surface tension coefficient is defined for each phase interaction.

To model surface tension in a fluid film simulation:

1. Set up a Multiphase Segregated Flow simulation, and activate the Fluid Film model.

   See Setting up a Fluid Film Simulation.
2. Set up a Film-Physics Continuum phase interaction connecting the Film phase with the physics continuum, and activate the Surface Tension model.

   See Surface Tension Model Reference.
3. Set the minimum film thickness and the appropriate tuning parameters.

   See Surface Tension Model Properties.
4. Specify the surface tension contact angle at each wall boundary in the fluid film shell region.

   See Modeling a Dynamic Contact Angle.

Contents:

Modeling Surface Tension in a Combined Fluid Film-VOF Simulation
Surface Tension Model Reference

Modeling Surface Tension in a Combined Fluid Film-VOF Simulation

In a VOF multiphase simulation, you can use the Fluid Film model to represent a film of variable thickness. In such a hybrid Film-VOF simulation, applications such as rainwater management, defogging, and painting/coating analyses require you to model surface tension effects in both the film domain and the VOF domain.

A Film-VOF phase interaction lets you model surface tension seamlessly across both the VOF phase and the film phase.

To model surface tension in a combined Fluid Film-VOF simulation:

1. Set up a VOF Multiphase simulation, and activate the Fluid Film model.

   See Setting up a VOF Multiphase Simulation.

2. Set up the three phase interaction models that you require.

   You require two surface tension interactions (in the VOF and Fluid Film domains) and a Resolved Fluid Film interaction to connect the fluid film and VOF liquid phases.

   For example, to model surface tension effects in a hybrid Film-VOF multiphase simulation with two VOF phases (*Air* and *Water*) and a liquid film phase (*WaterFilm*), you set up the following phase interaction models:

   - A VOF-VOF phase interaction connecting the *Water* and *Air* phases, with the Surface Tension Force model activated.

     See VOF Surface Tension Force Model.

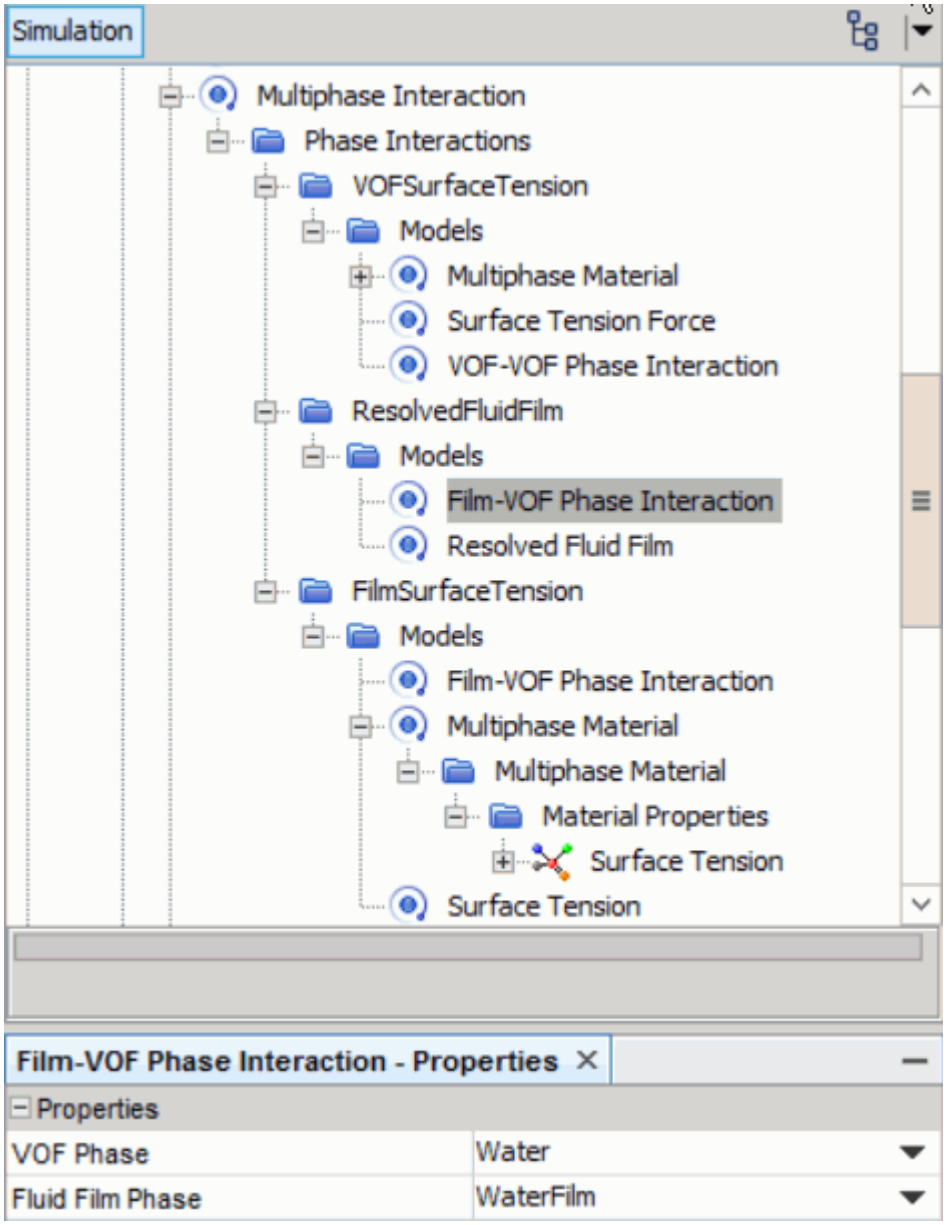
   - A Film-VOF phase interaction connecting the *Water* and *WaterFilm* phases, with the Resolved Fluid Film model activated.

     See Resolved Fluid Film Model Reference.

   - A Film-VOF phase interaction connecting the *Air* and *WaterFilm* phases, with the Surface Tension model activated.

     See Surface Tension Model Reference.

   The Film-VOF Phase Interaction controls the interaction between the fluid film phase and a VOF phase. This model automatically uses the phase model (fluid film or VOF) that is the most suitable for the local flow regime and the computer resources that are available.



3. Specify the surface tension coefficient σ as a material property in the Film-VOF phase interaction.

   See Surface Tension Material Properties.

4. You also specify the surface tension contact angle at each wall boundary.

   A **Contact Angle** node is added to the **Physics Values** of the phase interaction for all closed (wall) boundaries. This node lets you specify the capillarity effect at each of these boundaries.

   See Modeling a Dynamic Contact Angle.

5. You can specify the transition volume fraction α_{TRANS} at each interface between the VOF and the Fluid Film region.

   See Resolved Fluid Film Model Reference.

Surface Tension Model Reference

The Surface Tension model is available in Fluid Film simulations. This model can be used in Film-Physics Continuum and Film-VOF phase interactions.

Surface Tension Model Reference

Theory	See Fluid Film Surface Tension.	
Provided By	**[phase interaction]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Multiphase Interaction** > **Phase Interactions** > **[phase interaction]** > **Models** > **Surface Tension**	
Requires	A Multiphase Segregated Flow simulation with the Fluid Film model activated. A Film-Physics Continuum phase interaction or a Film-VOF phase interaction defined.	
Properties	See Surface Tension Model Properties.	
Activates	Materials	Surface Tension See Surface Tension Material Properties.
	Boundary Inputs	For all wall boundaries: **Boundaries** > **[wall boundary]** > **Phase Conditions** > **[phase interaction]** > **Physics Values** > **Contact Angle** See Modeling a Dynamic Contact Angle.
	Field Functions	See Surface Tension Model Field Functions.

Surface Tension Model Properties

Minimum Film Thickness

The minimum film thickness $\delta_{\min}$ that is used to determine the value of w in Eqn. (559). The default minimum film thickness is 0.01 mm.

Alpha Tuning Parameter

The tuning parameter that is used in the capillary pressure calculation. This value is a in Eqn. (558).

Beta Tuning Parameter

The parameter that is used to calibrate the contact line force calculation with respect to experimental results. This value is b in Eqn. (559).

Surface Tension Material Properties

Surface Tension

Each phase interaction is assigned its own surface tension coefficient (σ in Eqn. (558)). This coefficient is used to calculate the surface tension force between each of the defined phases in the phase interaction. It is entered as a scalar profile.

Surface Tension Model Field Functions

The following field functions are made available to the simulation when the Surface Tension model is activated in a phase interaction.

Capillary Number of [phase interaction]	This scalar field function is defined at wall boundaries and can be used in the definition of a user field function to specify your own user-defined dynamic contact angle correlation.
Contact Angle of [phase interaction]	The surface tension contact angle. This scalar field function is defined at wall boundaries.

Modeling a Fluid Film with Different Characteristic Length Scales

You can use the Resolved Fluid Film model to simulate free-surface flows in cases where the fluid film model is not adequate in all places or over the entire simulation time, and where using the VOF model requires fine meshes and unacceptably large computing times.

The two modeling approaches are combined so that the fluid film approach is used in regions where a thin liquid film is present and where the numerical mesh is not sufficient to resolve the flow and distribution of phases with the VOF model. As the fluid film gets thicker, or as the film enters regions with finer meshes, the flow is modeled using the VOF model. Similarly, if a VOF liquid phase enters a region with a coarse mesh, or if the liquid thickness gets smaller, the flow of liquid is modeled using the fluid film model.

The transition between the fluid film and VOF phase models is based on the volume fraction of liquid (the total of the fluid film phase and the VOF phase combined) at the interface and the transition volume fraction α_{TRANS} that is specified for the interface:

- Fluid film transitions into the VOF phase when the total volume fraction in the neighbor volume cell is greater than the specified transition volume fraction α_{TRANS}.
- The VOF phase transitions into fluid film when the total volume fraction is less than the specified transition volume fraction α_{TRANS}.

To ensure stability, mass is transferred at a finite rate: the transition of mass from one phase to the other takes place over a few iterations rather than in a single time-step. For more information, see the Resolved Fluid Film formulation.

The Resolved Fluid Film model can be used with both single-component and multi-component fluids.

However, the following conditions apply:

- A complete mixture mapping is required. Every component in the fluid film must be mapped to the corresponding component in the liquid VOF phase.
- Both the fluid film and VOF phases must consist of the same material components. However, the mass fractions of the components can be different in the initial material composition of the two phases.
- A fluid film or VOF phase can use only a single instance of the Resolved Fluid Film model.

  A particular phase cannot be included in two different Film-VOF phase interactions. For example, you cannot model two fluid film phases that are flowing into a single VOF phase.

Contents:

Resolved Fluid Film Model Reference

Resolved Fluid Film Model Reference

The Resolved Fluid Film model controls the transition between the specified fluid film phase and VOF phase. This model uses the phase model (fluid film or VOF) that is most suitable for the local flow regime and the computer resources that are available.

Resolved Fluid Film Model Reference

Theory	See Resolved Fluid Film.	
Provided By	**[phase interaction]** > **Models** > *Optional Models*	
Example Node Path	**[phase interaction]** > **Models** > **Resolved Fluid Film**	
Requires	A VOF Multiphase simulation with the **Fluid Film** model activated. A phase interaction with the **Film-VOF Phase Interaction** model activated. In the *Phase Interaction Model Selection* dialog: *Optional Models*: **Resolved Fluid Film**	
Properties	None.	
Activates	Interface Inputs	See Interface Settings.
	Field Functions	**Film-VOF Mass Transfer Rate of [Film-VOF Phase Interaction]** See Field Functions.

Interface Settings

Under the **Phase Conditions** > **[phase interaction]** > **Physics Values** node:

Resolved Fluid Film Volume Fraction

Specifies the value of α_{TRANS} for the interface. This value can be specified independently for each interface, and can be specified as a constant or as a field function. By default, α_{TRANS} has a constant value of 0.5.

For more information, see Specifying the Resolved Fluid Film Volume Fraction as a Field Function.

In some cases, using a particular value of α_{TRANS} for all boundaries does not produce the appropriate results. For example, if the mesh resolution is set for other purposes, such as to resolve the boundary layer. In these cases, you can ensure that certain boundaries are modeled as fluid film only, and other boundaries are modeled as VOF only, irrespective of the mesh that is used.

If you specify a value of $\alpha_{TRANS} = 0$, the fluid phase is modeled with VOF only. If you specify a value of $\alpha_{TRANS} = 1$, the fluid film model is used.

Specifying the Resolved Fluid Film Volume Fraction as a Field Function

When you specify α_{TRANS} as a field function, remember that the profile is evaluated on the interface. The Fluid Film phase exists on one side of the interface only, so a field function such as **FluidFilmThickness** is undefined on the VOF phase side of the interface.

To specify a field function profile that is defined on both sides of the interface, you can use one of the techniques described below:

- Wrap the **FluidFilmThickness** field function into the **alternateValue** field function, for example:

  ```
  alternateValue((${FluidFilmThickness} <0.0005) ? 2:0,0)
  ```

  The interpolation across the interface returns the arithmetic mean.
- Map the fluid film thickness to both the liquid side and the gas side of the interface at each iteration. As shown in the following example, the **MappedFluidFilmThickness** variable is available on both sides of the interface:

  ```
  (${MappedFluidFilmThickness} <0.0005) ? 1:0
  ```

The solutions of these two examples are equivalent. However, using the mapper adds more computational cost.

Resolved Fluid Film Field Functions

The following field function is made available to the simulation when the Resolved Fluid Film model is selected in a Film-VOF Phase Interaction. To activate this field function, the **Temporary Storage Retained** option must be activated in the Segregated Flow solver.

Film-VOF Mass Transfer Rate of [Film-VOF Phase Interaction]

The rate of mass transfer is evaluated at the interface between the fluid film and the VOF domain. The mass flow rate is provided in kg/s, with the sign convention corresponding to the direction of mass transport across the interface. You can evaluate this field function on either the fluid film shell region or the surrounding fluid region. For an interface boundary of a particular region, a positive value indicates mass transfer across the boundary into the region.

Fluid Film Formulation

The fluid film model accounts for transport of conserved quantities within the film and its interaction with surroundings.

STAR-CCM+ uses calculations that include:

- Mass Conservation Equation
- Momentum Conservation Equation
- Energy Conservation Equation
- Species Conservation Equation
- Grid Flux
- Volume Fraction
- Resolved Fluid Film
- Wave Stripping
- Edge Stripping
- Inertial Impingement
- Evaporation and Condensation
- Boiling
- Melting and Solidification

Major assumptions that are made in the basic model formulation are:

- The film is thin enough for laminar boundary layer approximation to apply. If turbulence occurs within the film, the assumption is violated.
- The film stays attached to the boundary unless an internal or separate model predicts film separation.
- There is a parabolic velocity profile across the film.
- The analysis is transient.

Using these assumptions, the film conservation equations that are given in the following sections are integrated and discretized for each film cell.

If wall velocity is $\mathbf{v}_w$, average film velocity is $\overline{\mathrm{v}}_f$, shear force at the film free surface is τ_{fs}, and ξ is the local coordinate normal to the wall, the assumed velocity profile across the film is:

$$\mathbf{v}(\xi) = \mathbf{v}_w + \left(3\frac{\overline{\mathrm{v}}_f - \mathbf{v}_w}{h_f} - \frac{\tau_{fs}}{2\mu}\right)\xi - \left(3\frac{\overline{\mathrm{v}}_f - \mathbf{v}_w}{2h_f^2} - \frac{3\tau_{fs}}{4\mu h_f}\right)\xi^2 \tag{545}$$

Numerical Implementation

The film solution domain is made of shell regions. The mesh of shell regions is created using the boundary faces of the volume mesh. A film cell is made from two-dimensional boundary faces. The edge length and the film thickness define the area of an edge face. The volume of the film cell is defined with the boundary area and the film thickness.

The film governing equations are integrated over the volume of fluid film in each film cell to obtain a set of algebraic equations.

The discretized equations of mass and momentum are solved in a segregated manner. The energy equation can either be solved segregated or as part of the coupled energy equation.

Contents:

Fluid Film Conservation Equations
Volume Fraction

Resolved Fluid Film
Fluid Film Surface Tension
Wave Stripping
Edge Stripping
Inertial Impingement
Evaporation and Condensation
Evaporation and Condensation Solution Procedure
Boiling
Melting and Solidification
Fluid Film Nomenclature

Fluid Film Conservation Equations

The conservation equations of mass, momentum, energy, and species, govern the behavior of fluid film. Each equation accounts for the impact of mesh motion.

The equation that governs the film mass conservation is as follows:

$$\frac{d}{dt}\int_V \rho_f dV + \int_A \rho_f \left(\mathbf{v}_f - \mathbf{v}_g\right) \cdot d\mathbf{a} = \int_V \frac{s_m}{h_f} dV \tag{546}$$

This equation is used to calculate the film thickness h_f. It is specific in the sense that the volume V and the surface A are functions of the film thickness and its spatial distribution.

Here, ρ_f is the film density, $\mathbf{v}_f$ is the film velocity, $\mathbf{v}_g$ is the grid velocity, and the subscript f denotes the fluid film values.

The quantity s_m is the mass source/sink per unit area. Contributions to the mass source or sink come from droplet impingement, film stripping, mass transfer, or user-defined sources.

The momentum conservation equation is:

$$\frac{d}{dt}\int_V \rho_f \mathbf{v}_f dV + \int_A \rho_f \mathbf{v}_f \left(\mathbf{v}_f - \mathbf{v}_g\right) \cdot d\mathbf{a} = \int_A \mathbf{T}_f \cdot d\mathbf{a} - \int_A p_f d\mathbf{a} + \int_V \left(\rho_f \mathbf{f}_b + \frac{\mathbf{s}_m}{h_f}\right) dV \tag{547}$$

where $\mathbf{v}_g$ is the grid velocity, $\mathbf{s}_m$ is the momentum source corresponding to the mass source s_m, p_f is the pressure, $\mathbf{f}_b$ is the body force (for example, gravity, or the pseudo-force in a moving reference frame), and $\mathbf{T}_f$ is the viscous stress tensor within the film. The kinematic and dynamic conditions at the interface between the film and the surrounding fluid (film free surface) are satisfied:

$$\left(\mathbf{v}_f\right)_{\text{int}} = \left(\mathbf{v}\right)_{\text{int}} \tag{548}$$

$$\left(\mathbf{T}_f \cdot d\mathbf{a} + p_f d\mathbf{a}\right)_{\text{int}} = \left(\mathbf{T} \cdot d\mathbf{a} + p d\mathbf{a}\right)_{\text{int}} \tag{549}$$

The quantities without the subscript f are associated with the surrounding fluid. Assuming that the normal components of the viscous and convective terms are negligible, the pressure distribution within the fluid film is obtained from Eqn. (547) as:

$$p_f(\xi) = p_{\text{int}} - \mathbf{s}_m \cdot \mathbf{n} - \rho_f \mathbf{f}_b \cdot \mathbf{n}\left(h_f - \xi\right) + \int_{\xi}^{h_f} \frac{d}{dt}\left(\rho_f \mathbf{v}_f \cdot \mathbf{n}\right) \mathrm{d}\,\xi \tag{550}$$

where $\mathbf{n}$ is the wall surface unit vector pointing towards the film, and ξ is the local coordinate normal to the wall. Eqn. (550) assumes that the force $\mathbf{s}_m$ is applied at the film free surface.

The energy conservation equation is:

$$\frac{d}{dt}\int_V \rho_f E_f \,\mathrm{d}\,V + \int_A \left[\rho_f H_f\left(\mathbf{v}_f - \mathbf{v}_g\right) + \mathbf{v}_g p_f\right] \cdot d\mathbf{a} = \int_A \mathbf{q}''_f \cdot d\mathbf{a} + \int_A \mathbf{T}_f \cdot \mathbf{v}_f d\mathbf{a} + \int_V \mathbf{f}_b \cdot \mathbf{v}_f dV + \int_V \frac{s_e}{h_f} dV \tag{551}$$

where E_f is the film total energy, H_f is the film total enthalpy, $\mathbf{q}''_f$ is the film heat flux, $\mathbf{T}_f$ is the fluid viscous stress tensor, and $\mathbf{f}_b$ is the body force. s_e is the energy source/sink term per unit film area. Contributions to the energy source or sink come from droplet impingement, film stripping, or user-defined sources.

The species mass conservation equation of species i is:

$$\frac{d}{dt}\int_V \rho_f Y_{i,f} dV + \int_A \rho_f\left(\mathbf{v}_f - \mathbf{v}_g\right) Y_{i,f} \cdot d\mathbf{a} = \int_A \left(\frac{\mu}{\sigma}\right) \nabla Y_{i,f} \cdot d\mathbf{a} + \int_V \frac{s_{m,i}}{h_f} dV \tag{552}$$

where $Y_{i,f}$ is the mass fraction of species i, and σ is the molecular Schmidt number. The quantity $s_{m,i}$ is the mass source or since for species i.

The grid flux at a face, as used to compute the face mass flux, is calculated as follows.

For a moving reference frame:

$$G_f = \mathbf{v}_g \cdot \mathbf{a}_f \tag{553}$$

where $\mathbf{v}_g$ is the grid velocity and $\mathbf{a}$ is the face area.

For a moving mesh:

$$G_f = \frac{\mathbf{a}_s \cdot h}{\Delta\, t} \tag{554}$$

where $\mathbf{a}_s$ is the area that is swept by the movement of the two vertices that make up the edge of the shell face, h is a vector with the magnitude of the film thickness in the wall surface-normal direction, and $\Delta\, t$ is the time step.

Stabilized Film Thickness Equation

Convergence problems can appear when the standard film mass conservation equation Eqn. (546) is used in applications in which the film thickness undergoes locally large changes. For example, high mass impingement where the impinging particles have a much larger diameter than the film cells. The convergence issues arise because the film fluxes depend on two dependent variables: the film thickness and the film velocity. That is, the mass conservation is influenced by both the velocity and the film thickness, and the simulation can develop oscillations of both.

To resolve this issue, a modified film thickness equation can be used instead of Eqn. (546). This equation is activated by the expert property, *Stabilized Film Thickness Equation*, in the Fluid Film Flow model.

The film thickness equation has two stabilization mechanisms added:

- Capillary Gravity Advection

  To damp velocity-thickness-oscillations, a linearization of an artificial capillary-gravity advection term is added to the film thickness equation. For converged simulations, this extra term has no effect on the results.
- Mass Conservation Residual

  To further improve convergence, the mass conservation residual is added to the film thickness equation.

Volume Fraction

The volume fraction of the film in the gas phase cell is calculated.

The volume fraction for the film, in the neighboring gas cell, is computed as:

$$\phi_{film} = min\left(\frac{V_{film}}{V_{gas}}, \phi_{film,max}\right) \tag{555}$$

where V_{film} is the film volume (the area multiplied by the film height) and V_{gas} is the volume of the neighboring cell.

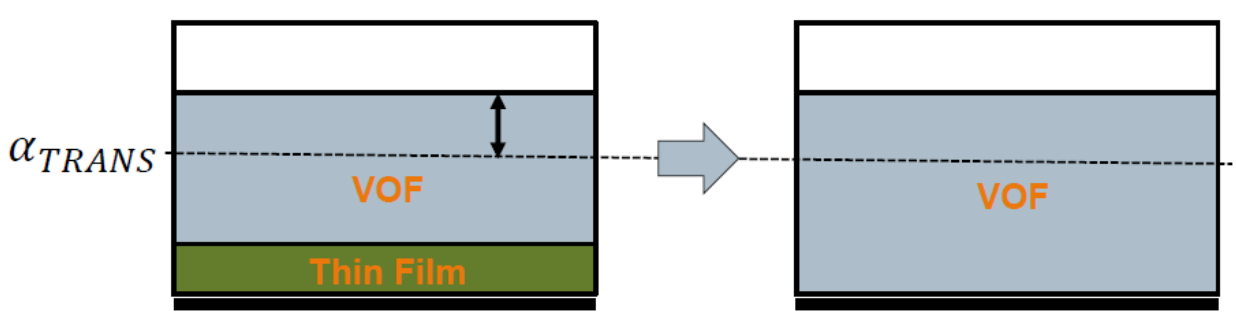
You can set $\phi_{film,max}$ in the Maximum Volume Fraction property of the fluid film phase.

Resolved Fluid Film

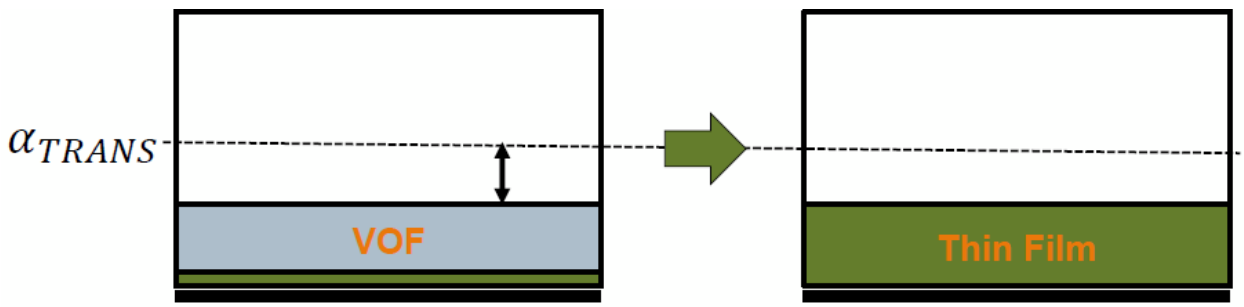
The Resolved Fluid Film model automatically uses the phase model (fluid film or VOF) that is most suitable for the local flow regime and the computer resources that are available.

The transition between the fluid film and VOF phase models is based on the volume fraction of liquid (the total of the fluid film phase and the VOF phase combined) at an interface and the transition volume fraction α_{TRANS} that is specified for the interface:

- Fluid film transitions into the VOF phase when the total volume fraction in the neighbor volume cell is greater than the specified transition volume fraction α_{TRANS}.



- The VOF phase transitions into fluid film when the total volume fraction is less than the specified transition volume fraction α_{TRANS}.



At every iteration, STAR-CCM+ determines the most appropriate physics model by considering the volume fraction:

$$\alpha_h = \frac{V_{VOF} + V_{film}}{V_{cell}} \tag{556}$$

where:

- V_{VOF} is the volume of the VOF liquid phase in the cell.
- V_{film} is the volume of fluid film in the cell.
- V_{cell} is the volume of the cell.

Values of α_{TRANS} from 0 through 1 give the expected mesh-dependent behavior. If $\alpha_h > \alpha_{TRANS}$, the fluid film is modeled as VOF; otherwise, it is modeled as fluid film. By default, $\alpha_{TRANS} = 0.5$.

If $\alpha_{TRANS} \leq 0$, the liquid is modeled as VOF only. If $\alpha_{TRANS} \geq 1$, the fluid film model is used.

The transfer of mass between the fluid film and the VOF liquid phase is:

$$\dot{m} = \left(\alpha_h - \alpha_{TRANS}\right) \cdot \frac{\widetilde{m}}{\tau} \tag{557}$$

where:

- $\dot{m}$ is the mass transfer rate in kg/s. This value is positive for fluid film → VOF mass transfer.
- $\widetilde{m}$ is the mass of liquid that is available for transfer.
- τ is the time scale. This value is of the order of the time step size.

Fluid Film Surface Tension

The surface tension model for fluid film is used to model surface tension between the film and the surrounding fluid. It is also used to model surface tension across both the VOF phase and the fluid film phase in combined Fluid Film-VOF simulations.

The surface tension formulation that is employed in the fluid film phase is based on the model that is introduced in [535]. There, the surface tension effects in the fluid film momentum transport are separated into two terms: capillary pressure and contact line force.

The capillary pressure, p_σ, is expressed as:

$$p_\sigma = -a\sigma\left(\nabla_s^2\delta\right) \tag{558}$$

where:

- σ is the surface tension coefficient.
- $\nabla_s^2\delta$ approximates the curvature of the liquid surface with δ being the fluid film thickness.
- a is a non-dimensional scale factor for the user to specify.

This force term has units of N/m^2 and acts orthogonal to the bottom surface. This relationship is valid only for surfaces with slight curvature [534].

The contact line force is calculated as:

$$\tau_\sigma = b\sigma(1-\cos(\theta))\nabla w \tag{559}$$

where:

- θ describes the contact angle as a fundamental parameter of the wetting behavior of the film.
- w is defined as 1 wherever $\delta > \delta_{\min}$ (a user-specified minimum value) and 0 elsewhere.
- b is an empirical parameter that is used to calibrate the model with respect to experimental results.

This force term has units of N/m^2 and acts tangential to the bottom surface.

The resulting surface tension force that is contributed to the momentum transport equation of the liquid film is:

$$\mathbf{F}_\sigma = p_\sigma + \tau_\sigma \tag{560}$$

Wave Stripping

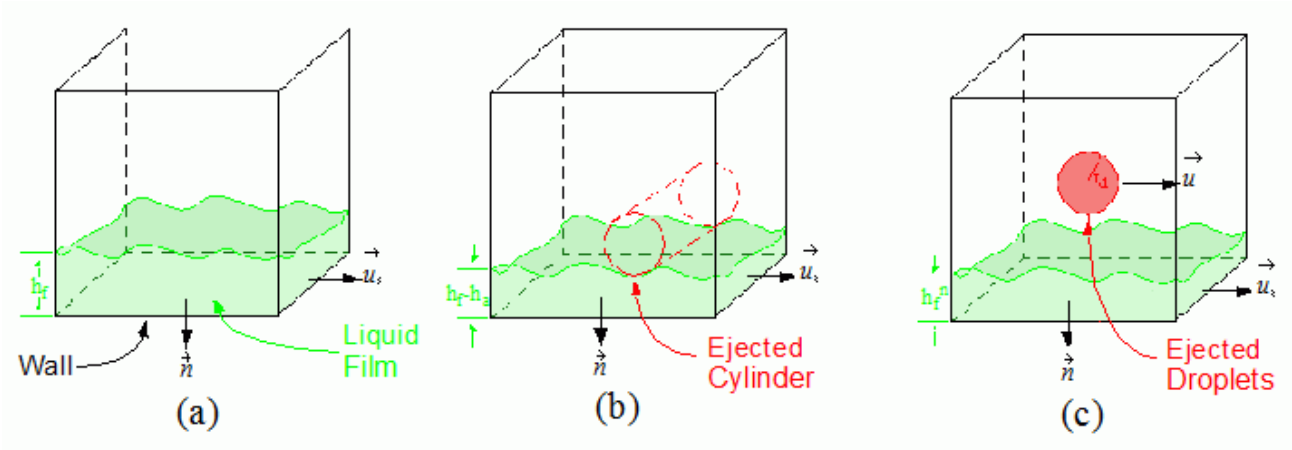
Depending on flow conditions and the geometry of the wall on which the film resides, the film can form droplets and this process is called film stripping.

Two stripping mechanisms are modeled:

- Stripping due to wave instability generated by an adjacent fluid flow
- Stripping due to body-force induced instability (gravity, film acceleration)

Adjacent Flow and Body-Force Induced Instability

Due to external forces such as gravity or shearing at the interface between a fluid film and its surrounding fluid, the film surface can develop instabilities and start shedding droplets. The model described here is based on the original documentation of Hervé Foucart [556]. It postulates that stripping at the surface occurs in a three-stage process, as represented schematically in the following illustration of the liquid film stripping mechanism.



First, waves develop at the liquid-gas interface. As these waves grow, they become unstable and form cylinders of liquid which are ejected into the surrounding gas. The cylinders then break up further forming spherical droplets. The model is closed by computing the radius of the child droplets as a function of the most unstable wavelength characterizing the surface instability.

It is accepted that for surface stripping:

- Ejection occurs when critical conditions are satisfied, that is a minimum height is required for droplet ejection.
- The ejected fluid volume forms a cylinder, as a result of the Kelvin-Helmholtz instability.
- The cylinder then breaks into droplets due to Rayleigh-Taylor instabilities.

Having characterized surface stripping as a function of wavelength, the stripping model relationships are now closed by computing the wavelength for which surface break-up is most likely to occur. This wavelength is the most unstable, and is calculated as the resonance wavelength using the dispersion equation. Solving the dispersion equation for surface waves, the resonance wavelength is obtained as:

$$\lambda_r = \frac{2\pi}{\rho_f \mathbf{f}_b \cdot \mathbf{n}} \left(\frac{1}{3}\rho \tilde{\mathbf{v}}_r^2 - \sqrt{\left(\frac{1}{3}\rho \tilde{\mathbf{v}}_r^2\right)^2 - \rho_f \mathbf{f}_b \cdot \mathbf{n}\sigma} \right) \tag{561}$$

where $\tilde{\mathbf{v}}_r^2$ is the square of relative velocity between the film and surrounding fluid $\left(\mathbf{v}_f - \mathbf{v}\right) \cdot \left(\mathbf{v}_f - \mathbf{v}\right)$, σ is the surface tension, and $\mathbf{f}_b$ is the body force acting on film which also includes the inertial force. The minimum height necessary for droplet ejection is then:

$$h_{\min} = c_H \frac{\lambda_r}{2\pi} \tag{562}$$

The film height to strip, that is the surface wave amplitude, is:

$$h_a = \left[\frac{3}{4} \left(\frac{2}{3.78} \right)^3 \right]^2 \pi \lambda_r \tag{563}$$

The resulting droplet diameter is:

$$D_d = c_D\sqrt{\frac{\lambda_r h_a}{\pi}} \tag{564}$$

Note: For wave stripping into a dispersed phase, Eqn. (564) is not used. The droplet diameter is specified manually rather than being computed.

The default setting for c_D is 3.78. The number 3.78 in Eqn. (563) arises from Rayleigh theory. It means that the diameter of each droplet that is generated from the cylinder is 3.78 times the diameter of that cylinder.

All droplets initially have the same velocity as the film and are placed at a point between the cell center and the boundary face center, depending on the radius of the ejected droplet.

Although no consideration of time scale is included in the original stripping model, the time scale can be defined in terms of a linear growth rate as:

$$t_b = \frac{\lambda_r\left(\rho_f + \rho\right)}{|\mathbf{v}_r|\sqrt{(1 - \mathrm{We})\rho_f\rho}} \tag{565}$$

where We is the Weber number and the most unstable mode occurs when $\mathrm{We} = 2/3$.

Using this time scale, stripping is expected to occur n_b times during a time-step Δt:

$$n_b = c_B\frac{\Delta t}{t_b} \tag{566}$$

Free-Stream Velocity

The Wave Stripping model requires a measure of the free-stream velocity ($\mathbf{v}_r$ as used in Eqn. (561)).

STAR-CCM+ calculates two measures of the free-stream velocity and uses the highest.

- One measure is the gas velocity in the nearest gas cell. The disadvantage with this measure is that the value depends on the mesh size, and the approximation becomes less valid as the mesh size decreases.
- A second measure is a mesh-independent estimate of the free-stream velocity based on the shear velocity. This second measure is derived below.

At the free surface, on the gas side, the shear (or frictional) velocity is defined as:

$$\mathbf{u}_* = -\sqrt{\frac{\tau}{\rho}}\mathbf{n}_\tau \tag{567}$$

where $\mathbf{n}_\tau$ is the unit vector in the direction of the shear stress τ.

It is assumed that the free-stream gas velocity can be reasonably approximated as:

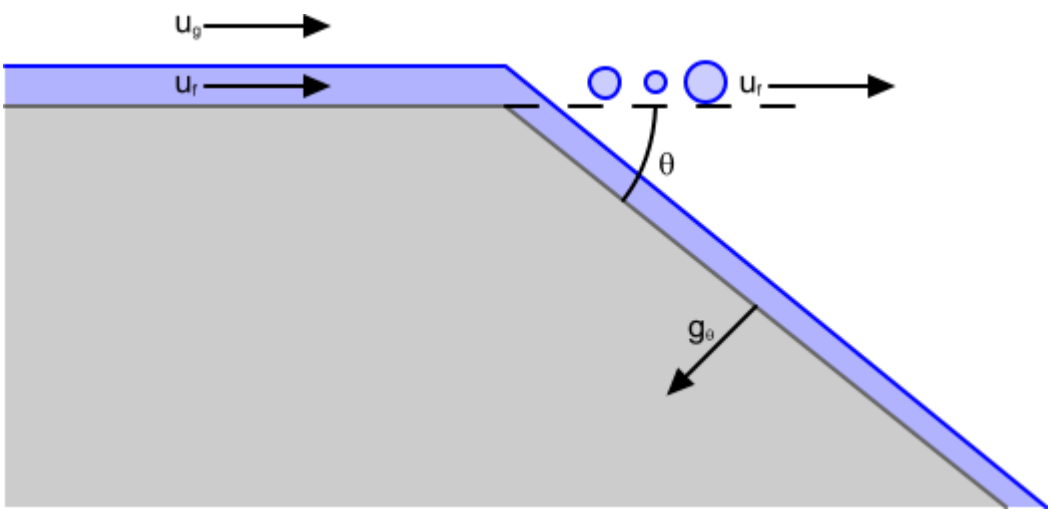
$$\mathbf{v}_r \approx \mathbf{v}_{l,g} + K_{*,\infty}\mathbf{u}_* = \mathbf{u}_{*,\infty} \tag{568}$$

where $\mathbf{v}_{l,g}$ is the interfacial velocity and $K_{*,\infty}$ is a constant that relates the shear velocity to the free-stream velocity. This constant is specified as the *Gas Velocity Scaling Factor* in the Wave Stripping model properties.

Edge Stripping

Edge stripping models the break-up of the liquid film over a sharp edge.

Maroteaux et al. [561] proposed a model for the break-up of liquid films over a sharp edge. This model is based on the Rayleigh-Taylor instability and contains the break-up criterion and resulting droplet diameter distribution. However, Gubaidullin [558] has raised questions about the validity of the breakup criterion. Therefore, an alternative criterion is adopted, proposed by Friedrich et al. [557], that is based on a force balance. The droplet diameter distribution still follows Maroteaux.



The model is only applied at edges where the corner angle (θ in the diagram above) is greater than the user-defined minimum corner angle, θ_{min}. If a boundary cell has multiple sharp edges, stripping occurs over all of them in a weighted manner.

The force ratio FR, which is essentially the ratio of the film momentum flux to the surface tension and gravity force, is:

$$FR = \frac{We_f}{1 + \frac{1}{\sin\theta} + Bo_f \frac{L_b}{h_f \sin\theta}} \tag{569}$$

where We_f is the film Weber number (see Eqn. (570)), Bo_f is the Bond number (see Eqn. (571)), L_b is the break-up length (see Eqn. (572)) and h_f is the film thickness.

The film Weber number is:

$$We_f = \frac{\rho_f {u_f}^2 h_f}{\sigma} \tag{570}$$

where ρ_f is film density, u_f is film velocity projected orthogonal to the stripping edge, h_f is film thickness and σ is film surface tension.

The film Bond number is:

$$Bo_f = \frac{\rho_f g_\theta {h_f}^2}{\sigma} \tag{571}$$

where g_θ is the component of acceleration normal to the downstream wall.

Arai and Hashimoto [554] give the break-up length as:

$$L_b = 0.0388 {h_f}^{0.5} {Re_f}^{0.6} {We_{rel}}^{-0.5} \tag{572}$$

where Re_f is the film Reynolds number (see Eqn. (573)) and We_{rel} is the relative Weber number (see Eqn. (574)).

The film Reynolds number is:

$$Re_f = \frac{\rho_f u_f h_f}{\mu_f} \tag{573}$$

where μ_f is the film viscosity.

The relative Weber number is:

$$We_{rel} = \frac{h_f \rho_g \left(u_g - u_f\right)^2}{2\sigma} \tag{574}$$

where ρ_g is the gas density and u_g is the gas velocity projected orthogonally to the stripping edge.

The break-up is deemed to occur when $FR > FR_C$, where FR_C is the user-defined critical force ratio, which is proposed in [557]. The critical force ratio default value is 1. Of the fluid crossing the edge, only a fraction x_S separates from the film. This fraction is approximated using the formula below:

$$x_S = \begin{cases} 0 & FR \leq FR_C \\ 0.44\left(FR - FR_C\right) & FR_c < FR \leq \left(FR_C + 1.6\right) \\ 0.057\left(FR - FR_C - 1.6\right) + 0.704 & \left(FR_C + 1.6\right) < FR \leq \left(FR_C' + 6.792\right) \\ 1 & \left(FR_C + 6.792\right) < FR \end{cases} \tag{575}$$

which is based on the experimental data in [557].

Note: For edge stripping into a dispersed phase, Eqn. (576) to Eqn. (581) are not used. The droplet diameter is specified manually rather than being computed.

For droplet formation, the model that is presented in [561] is adopted, which estimates the parent droplet diameter D_d as:

$$D_d = c_1 \sqrt{\frac{\lambda_m h_f}{\pi}} \tag{576}$$

where c_1 is a user-defined droplet diameter scale factor. This factor is set to 3.78 by default. The wave length λ_m is calculated from the most unstable wavenumber k_m, which maximizes the growth rate:

$$\omega = -\left(\frac{\sigma-\left(\rho_f-\rho_g\right)a/k^2}{2\mu_f h_f}\right)\left(\frac{\left(kh_f\right)\sinh\left(kh_f\right)\cosh\left(kh_f\right)-k^2h_f^2}{\cosh^2\left(kh_f\right)+k^2h_f^2}\right) \tag{577}$$

where the acceleration a is computed from:

$$a = \frac{u_f^2\theta}{h_f(\pi+\theta)} \tag{578}$$

and λ_m is related to k_m by $\lambda_m = 2\pi/k_m$. You can choose to provide an upper limit for the parent droplet diameter D_d by activating the Limit parent droplet diameter property and supplying a value for the Maximum diameter.

According to the model, the cumulative droplet size distribution $F(D)$ is a Rosin-Rammler distribution:

$$F(D) = 1-e^{-(D/X)^q} \tag{579}$$

where q is a user-defined parameter, which is set to 1.5 by default, and:

$$X = \frac{D_d}{(3\ln 10)^{1/q}} \tag{580}$$

There are two options for determining the droplet diameter:

- Generate droplets with a deterministic diameter, namely the average of the size distribution that is:

$$D = X\Gamma\left(1+\frac{1}{q}\right) \tag{581}$$

where $\Gamma(x)$ represents the Gamma function.

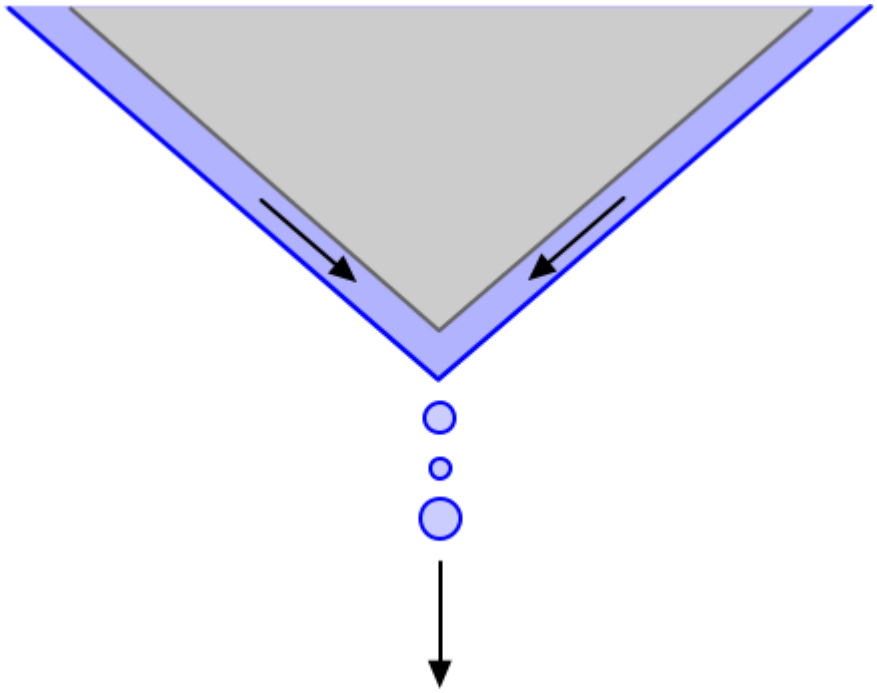
- Generate droplets with a randomized diameter D according to Eqn. (579).

You can select these options by either activating or deactivating the Randomize droplet diameter property.

The droplets are ejected at the film velocity speeds in terms of both magnitude and direction.

Edge Stripping with Flow from Both Sides

For a case where flow comes toward an edge from both sides, the flows are combined into one stream of Lagrangian droplets that conserves mass, momentum, species, and energy. For example, if both flows have a different temperature, the droplets get a temperature somewhere in between. For opposing flows, the stripping fraction x_S (Eqn. (575)) is also set to 1 for both sides (which is full stripping). Droplet diameter is computed based on the film properties of the side with the biggest flux. The following diagram illustrates opposing-flow edge stripping.



Inertial Impingement

The Inertial Impingement model uses the inertia of the dispersed droplets that approach the wall to compute the mass flux of the droplets that impinge on the wall to form a fluid film. Impingement occurs on the walls at which a shell interface has been created.

The mass flux due to impingement is:

$$m_i = \gamma \rho_i \mathbf{U}_i \cdot \mathbf{A} \alpha_i \tag{582}$$

where:

- m_i is the mass flux leaving the system.
- γ is the impingement efficiency.
- ρ_i is the cell-centered density.
- $\mathbf{U}_i$ is the cell-centered velocity.
- $\mathbf{A}$ is the cell wall area vector (normal to the wall in the outward facing direction).
- α_i is the cell-centered volume fraction.

The impingement efficiency specifies the fraction of the flux incident on the surface that impinges on the fluid film. The impingement efficiency can be a value from 0 through 1.

The kinematic and dynamic conditions at the interface between the fluid film and the surrounding multiphase fluid (film free surface) are satisfied. The impinging mass flux that is calculated at the interface cell is used in all of the other transport equations.

The liquid phase impinging on the fluid-film interface must have the exact same material components as the fluid-film phase.

Evaporation and Condensation

STAR-CCM+ models evaporation of the fluid film into a gas phase and condensation from a gas phase into the fluid film.

The species mass flux for every component is conserved at the interface between the gas and the fluid film, such that:

$$\rho_{g_s} Y_{g_s,i}\left(v_{g_s} - \dot{h}\right) - \rho_{g_s} D_{g,i} \frac{\mathrm{d}\, Y_i}{\mathrm{d}\, y}\bigg|_{g_s} = \rho_{l_s} Y_{l_s,i}\left(v_{l_s} - \dot{h}\right) - \rho_{l_s} D_{l,i} \frac{\mathrm{d}\, Y_i}{\mathrm{d}\, y}\bigg|_{l_s} \tag{583}$$

where, evaluated at surface s:

- ρ_{g_s} and ρ_{l_s} are the gas and liquid film densities
- $Y_{g_s,i}$ and $Y_{l_s,i}$ are the mass fractions for the gas and liquid film
- v_{g_s} and v_{l_s} are the normal velocity components for the gas and liquid film
- $D_{g,i}$ and $D_{l,i}$ are the gas and liquid film molecular diffusion coefficients
- $\dot{h}$ is the rate of change of film thickness

The total mass flux is also conserved:

$$\rho_{g_s}\left(v_{g_s} - \dot{h}\right) = \rho_{l_s}\left(v_{l_s} - \dot{h}\right) \tag{584}$$

Combining Eqn. (583) and Eqn. (584):

$$\rho_{l_s}\left(v_{l_s} - \dot{h}\right)\left(Y_{g_s,i} - Y_{l_s,i}\right) + \rho_{l_s} D_{l,i} \frac{\mathrm{d}\, Y_i}{\mathrm{d}\, y}\bigg|_{l_s} - \rho_{g_s} D_{g,i} \frac{\mathrm{d}\, Y_i}{\mathrm{d}\, y}\bigg|_{g_s} = 0 \tag{585}$$

The evaporation rate is defined as:

$$\dot{m}_v = -\rho_{l_s} \dot{h} \tag{586}$$

with v_{l_s} considered to be zero from here on. The summation of Eqn. (585) over all liquid film component N_L becomes:

$$\left[1 - \sum_j^{N_L} Y_{g_s,j}\right] \dot{m}_v = -\sum_j^{N_L} \rho_{g_s} D_{g,i} \frac{\mathrm{d}\, Y_i}{\mathrm{d}\, y}\bigg|_{g_s} \tag{587}$$

and since

$$-Y_{g_s,j} \dot{m}_v = -\rho_{g_s} D_{g,i} \frac{\mathrm{d}\, Y_i}{\mathrm{d}\, y}\bigg|_{g_s} \tag{588}$$

for all $N_{P,L}$ species i in the film that are inert, the summation can be limited to the N_v interacting components. The total evaporation rate can therefore be expressed as:

$$\dot{m}_v = -\frac{\sum_j^{N_v} \rho_{g_s} D_{g,i} \left.\frac{\mathrm{d}\, Y_i}{\mathrm{d}\, y}\right|_{g_s}}{1 - \sum_j^{N_v} Y_{g_{s'}j}} \tag{589}$$

Note that Eqn. (589) is valid only below saturation conditions. That is:

$$\sum_j^{N_v} Y_{g_{s'}j} < 1 \tag{590}$$

At saturation conditions, the algebra that is used to derive Eqn. (589) is not correct.

The normal derivative is treated through the species transfer coefficients $s_{t,i}$ and the Spalding transfer number B such that:

$$\dot{m}_v = -\frac{\sum_j^{N_v} s_{t,j}\left(Y_{g_{c'}j} - Y_{g_{s'}j}\right)}{1 - \sum_j^{N_v} Y_{g_{s'}j}} \cdot \frac{\ln(1+B)}{B} \tag{591}$$

where the subscript c indicates a cell value and B is defined as:

$$B = \frac{\sum_j^{N_v} Y_{g_{s'}j} - \sum_j^{N_v} Y_{g_{s'}\infty}}{1 - \sum_j^{N_v} Y_{g_{s'}j}} \tag{592}$$

where it is assumed that $Y_{g_{s'}\infty} \approx Y_{g_{s'}c}$

When the gas phase is a VOF phase, $\dot{m}_v$ is multiplied by the volume fraction of the VOF phase.

The interfacial gas mass fraction $Y_{g_{s'}j}$ is required to evaluate Eqn. (591). $Y_{g_{s'}j}$ is related to $Y_{l_{s'}j}$ through Raoult's law, as follows:

Consider the vapor pressure p_i of a component:

$$p_i = a_i p^*_i \tag{593}$$

where p^*_i is the vapor pressure of the pure component i and a_i is the activity, which is dependent on the liquid mixture. This expression can be written as:

$$p_i = \gamma_i X_{l_{s'}i} p^*_i\left(T_s\right) \tag{594}$$

Raoult's law states that $\gamma_i \approx 1$, an approximation that is followed here. Given the components vapor pressures, their molar fractions at the interface can be derived from:

$$X_{g_{s'}i} = \frac{p_i}{p} \tag{595}$$

where p is the ambient pressure. The conversion to interface mass fraction is obtained using:

$$Y_{g_{s'},i} = \frac{X_{g_{s'},i} W_i}{\sum_j^{N_v} X_{g_{s'},j} W_j + \sum_j^{N_{P,G}} X_{g_{s'},j} W_j} \tag{596}$$

While $X_{g_{s'},i}$ is known for the evaporation components, it is unknown for the inert species. A good approximation is the introduction of a background molar weight, which is calculated at a small distance away from the interface:

$$W_{bg} = \frac{\sum_j^{N_{P,G}} X_j W_j}{\sum_j^{N_{P,G}} X_j} \tag{597}$$

The interfacial background molar fraction is:

$$X_{g_{s'},bg} = 1 - \sum_j^{N_v} X_{g_{s'},j} \tag{598}$$

The interfacial vapor mass fraction can then be approximated as:

$$Y_{g_{s'},i} \approx \frac{X_{g_{s'},i} W_i}{X_{g_{s'},bg} W_{bg} + \sum_j^{N_v} X_{g_{s'},j} W_j} \tag{599}$$

Eqn. (594) assumes knowledge of $X_{l_{s'},i}$. Presently, the normal species profile in the film is approximated as a constant, such that the interfacial molar fractions follow from the associated cell center values:

$$X_{l_{s'},i} = X_{l_{c'},i} \tag{600}$$

The component evaporation rate can be expressed as:

$$\dot{m}_{v,\,i} = Y_{g_{s'},\,i} \dot{m}_v - \rho_{g_s} D_{g,\,i} \left.\frac{\mathrm{d}\,Y_i}{\mathrm{d}\,y}\right|_{g_s} \tag{601}$$

which is valid under all conditions.

The interfacial heat flux conservation is expressed as:

$$k_g \left.\frac{\mathrm{d}\,T}{\mathrm{d}\,y}\right|_{g_s} - k_l \left.\frac{\mathrm{d}\,T}{\mathrm{d}\,y}\right|_{l_s} - \dot{Q}_v = 0 \tag{602}$$

where k denotes thermal conductivity and

$$\dot{Q}_v = \sum_i^{N_v} \Delta H_i^{vap} \dot{m}_{v,i} \tag{603}$$

which is a condition for the interfacial surface temperature T_s.

Note that a combination of Eqn. (601) and Eqn. (603) leads to an expression for the total evaporation rate which is valid at all conditions:

$$\dot{m}_v = \frac{\dot{Q}_v + \sum_i^{N_v} \Delta H_i^{vap} \rho_{g_s} D_{g,\,i} \left.\frac{\mathrm{d}\,Y_i}{\mathrm{d}\,y}\right|_{g_s}}{\sum_i^{N_v} \Delta H_i^{vap} Y_{g_s,\,i}} \tag{604}$$

Evaporation and Condensation Solution Procedure

STAR-CCM+ calculates evaporation and condensation in both non-saturated and saturated regimes.

Below Saturation

Below the saturation temperature (also referred to as boiling temperature), evaporation and condensation rates are controlled by the following:

- Hydrodynamic effects - vapor needs to be delivered to or carried away from the surface, which happens at a finite rate.
- Thermal effects - heat needs to be conducted towards or away from the interface to balance the latent heat of evaporation.

STAR-CCM+ attempts to satisfy Eqn. (605), in which $\dot{Q}_v$ is given by Eqn. (603), which in turn relies on Eqn. (601), which requires Eqn. (589). Despite the apparent complexity, this condition is simply finding the interfacial temperature T_s such that Eqn. (605) is satisfied.

This problem can be approached in many ways. The present implementation treats it by iteratively looking for the root of the function:

$$f(T_s) = k_g \left.\frac{\mathrm{d}\,T}{\mathrm{d}\,y}\right|_{g_s} - k_l \left.\frac{\mathrm{d}\,T}{\mathrm{d}\,y}\right|_{l_s} - \dot{Q}_v(T_s) \tag{605}$$

through the secant method.

STAR-CCM+ uses the following process:

1. Start with two initial values, $T_{s,\,0}$ and $T_{s,\,1}$, and evaluate f_0 (shorthand for $f(T_{s,\,0})$) and f_1.
2. Compute $T_{s,\,n} = T_{s,\,n-1} - f_{n-1}\left(\frac{T_{s,\,n-1} - T_{s,\,n-2}}{f_{n-1} - f_{n-2}}\right)$ for $n = 3..N_{it}$.

At present, the first initial value is taken to be the actual interfacial temperature upon entering the optimizer, and the second initial value is taken 5 percent lower than the first. Note that the solution of T_s is limited from above by T_{sat}. At this point it is assumed that the exclusively thermally limited rate (essentially stating that vapor is infinitely available) is valid.

At Saturation

Saturation can be detected by either, or both of, the following conditions:

$$\sum_i^{N_V} Y_{g_{s,\,j}} = 1 \tag{606}$$

$$\sum_{i}^{N_V} Y_{g_{\infty,j}} = 1 \tag{607}$$

The first condition (Eqn. (606)) is obvious, as it is essentially stating that the temperature at the interface has reached the boiling temperature. The second condition (Eqn. (607)) states that a pure vapor (without the presence of non-condensable gases) can only be in quasi-steady equilibrium at the interface if its temperature is the saturation temperature. If either of these conditions are detected, the hydrodynamic procedure described above is cut short and the interfacial temperature is set to T_{sat}.

Final Application of Thermal Equilibrium

Once T_s is found (either below the boiling temperature for hydrodynamically limited conditions or exactly at the boiling temperature for thermally limited conditions), STAR-CCM+ finalizes the calculation by using the following steps:

1. Compute $\dot{Q}_v$ via enforcing the thermal equilibrium of Eqn. (605).
2. Compute $\dot{m}_v$ using Eqn. (604).
3. Compute $\dot{m}_{v,\,i}$ using Eqn. (601).

Note that if $T_s < T_{sat}$, these final steps are not necessary: they are fully consistent with the quantities that are computed during the iterative process, and would therefore be already available. However, these final steps are presented to illustrate that, once T_s has been found, evaporation rates can be calculated in an identical fashion for both the non-saturated and the saturated regimes.

Steam Condensation at a Dry Wall

The method that is outlined above runs into issues at perfectly dry cold walls that are in contact with a pure vapor (for example, at the start of the simulation).

The heat transfer coefficient in the film is generally given by:

$$h_f = \frac{2k_f}{h} \tag{608}$$

so the following approximation (based on the assumption of piecewise linear temperature profile) holds:

$$k_l \frac{\mathrm{d}\,T}{\mathrm{d}\,y}\bigg|_{l_s} \approx \frac{2k_f}{h}\left(T_{l,c} - T_s\right) \tag{609}$$

For vanishing film thickness and $T_s > T_{l,c}$, Eqn. (609) tends to infinity, expressing that at zero film thickness the heat is carried towards the wall infinitely fast. At saturation conditions, the thermally limited model (which is assumed here) effectively predicts infinite condensation rates, which causes the simulation to diverge. (The heat transfer coefficient that is calculated by STAR-CCM+ does not actually go to infinity, as a certain minimal thickness (currently 10^{-8}m) is used. However, the calculated value is still extremely high.)

Aside from molecular considerations (such as kinetic theory and interfacial non-equilibrium), the main physics that is missing is in how the condensation is modeled. Condensation at a dry wall does not happen in the form of a flat film, but rather as droplets growing at nuclei. The growth of these droplets is restricted by surface

tension effects, and the interfacial contact area between vapor and liquid is very small, so it is likely that these become limiting factors at the early stages of condensation.

To implement a model that behaves in the same way that the purely hydrodynamically limited model behaves (that is, models the condensation as a fully-wetting film), but remains stable, a simplified drop-wise approach is used. The impact of the model is only on the interfacial area.

For the derivation, consider a film cell with area A and thickness h. Assume that there is a nucleation density of N (m^{-2}) and that the nuclei have a minimum radius of $R_{\min}$. It is assumed that the droplets form with a 90 degree contact angle (a temporary simplification), and that they are all identical, with a radius R. For a given film thickness, the radius of the drops (assuming hemispheres) can be expressed as:

$$R = \left(\frac{3h}{2\pi N}\right)^{1/3} \tag{610}$$

It is assumed that the evaporation rates (as calculated before) still hold and that the only effect is due to the smaller total interfacial area between the vapor and the film. This ratio can be described as:

$$f_A = 2\pi N R_{eff}^2 \tag{611}$$

which behaves as $N^{1/3}h^{2/3}$.

When this factor f_A is smaller than 1, it is used as a multiplier for the evaporation rates. The model requires two additional parameters: the nucleation density N and the minimum radius $R_{\min}$. These parameters are specified as properties of the evaporation and condensation model. N is specified as **N_seeds** and $R_{\min}$ is specified as **R_seeds**. For more information, see Evaporation-Condensation Model Properties.

This model stabilizes the solution without affecting the longer-term result (that is, once a fully-wetting film has formed and grown for a while).

Boiling

Boiling is considered when the temperature of the fluid film is above the boiling temperature and the boiling property is activated. Below the boiling temperature, the same equilibrium conditions as for evaporation are assumed, but the vapor mass rates are determined differently.

Bulk Boiling

The heat transfer to the interface between the phases is used to compute the mass rate of evaporation when film cells are boiling. It is assumed that:

- The vapor bubbles are at the saturation temperature.
- The temperature of the liquid is approximately the temperature of the mixture.
- All the heat flux from the liquid to the interface is used in mass transfer due to evaporation.

$$\dot{m}_v = \frac{\mathrm{HTCxArea}\left(T - T_{sat}\right)}{\Delta H^{vap}} \tag{612}$$

$\mathrm{HTCxArea}$ is the product of the heat transfer coefficient between the vapor bubbles and the surrounding liquid, and the specific contact area (contact area per unit volume) between the two. ΔH^{vap} is the latent heat of vaporization of the mixture and computes as:

$$\Delta H^{vap} = \sum_i Y_{g,i} \Delta H_i^{vap} \tag{613}$$

where $Y_{g,i}$ is the mass fraction and ΔH_i^{vap} is the latent heat of pure component i. This latent heat is the difference between the total enthalpy of its liquid phase $H_{i,l}$ and its gas phase $H_{i,g}$.

$$\Delta H_i^{vap} = H_{i,g} - H_{i,l} \tag{614}$$

Wall Boiling

The empirical standard pool boiling correlation [564] is used to model liquid film boiling at a hot wall. The film starts to boil when the wall temperature exceeds the boiling temperature of the liquid. The heat flux passing from the wall to the film increases in proportion to the cubic power of $\Delta T_W = T_W - T_{sat}$ as:

$$q_{boil} = \frac{\mu_l \Delta H^{vap}}{\mathrm{Pr}_l^n} \sqrt{\frac{g\left(\rho_l - \rho_g\right)}{\sigma}} \left(\frac{C_{p,l} \Delta T_W}{c_{sf} \Delta H^{vap}} \right)^3 \tag{615}$$

where:

- the viscosity μ_l, the Prandtl number Pr_l^n, and the specific heat $C_{p,l}$ are evaluated for the liquid phase.
- ρ_l and ρ_g denote the densities of the liquid and gas phases.
- g and σ denote the gravity acceleration and surface tension coefficient respectively.
- c_{sf} and n characterize the fluid wall interface.

  For example, for water-brass contact, the values are $c_{sf} = 0.006$ and $n = 3$.
- T_{sat} represents the saturation temperature of the mixture and is computed from the temperature-dependent saturation pressure of its components by applying Dalton's law.

The heat flux increases as T_W increases but does not exceed a critical heat flux that is:

$$q_{max} = c_{max} \rho_g^{1/2} \Delta H^{vap} \left[g\left(\rho_l - \rho_g\right)\sigma \right]^{1/4} \tag{616}$$

where $c_{max} = 0.15$. After q_{max} has been attained, the heat flux decreases to its minimum at the Leidenfrost temperature T_L:

$$q_{min} = c_{min} \rho_g \Delta H^{vap} \left[g\left(\rho_l - \rho_g\right)\sigma \right]^{1/4} \left(\rho_l + \rho_g\right)^{-1/2} \tag{617}$$

where $c_{min} = 0.09$.

The wall temperature of maximal heat flux $T_S = T_{sat} + \Delta T_S$ can be obtained from equations for q_{max} and q_{boil} resulting in

$$\Delta T_S = \frac{c_{sf} c_{max}^{1/3} \mathrm{Pr}_l^{n/3} \sigma^{1/4} \rho_g^{1/6} \Delta H^{vap}}{C_{p,l} \mu_l^{1/3} g^{1/12} \left(\rho_l - \rho_g\right)^{1/12}} \tag{618}$$

Based on this correlation, the heat flux from the wall is computed as follows:

If $\Delta T_W < \Delta T_S$, then

$$q_W = q_{boil} \tag{619}$$

Else, if $\Delta T_W < c_s \Delta T_S$, then

$$q_W = q_{max} \tag{620}$$

Else, if $\Delta T_W < \Delta T_L$, then

$$q_W = q_{min} + \frac{\Delta T_L - \Delta T_W}{\Delta T_L - c_s \Delta T_S}(q_{max} - q_{min}) \tag{621}$$

Else,

$$q_W = q_{min} \frac{\Delta T_W}{\Delta T_L} \tag{622}$$

$\Delta T_L = T_L - T_{sat}$ and c_s is assumed to be 1.2.

The component mass rates of vaporization are then computed as

$$\dot{m}_{v,i} = c_e Y_{g,i} \frac{q_W}{\Delta H^{vap}} \tag{623}$$

where c_e is the wall boiling evaporation factor that determines how much of the boiling heat flux generates vapor. A value smaller than one means that a portion of the boiling heat flux does not generate vapor but heats up the film, for example:

- when heat is supplied near the wall to bubbles which then move away into the liquid film and condensate again inside the film
- when evaporation takes place at the base of a bubble and condensation at its tip.

Melting and Solidification

STAR-CCM+ models melting and solidification of the fluid film.

Melting-Solidification Enthalpy Formulation

The Melting-Solidification model does not track the liquid-solid interface explicitly. Instead, the model uses an enthalpy formulation to determine the distribution of the solidified portion of the liquid-solid film.

For basic melting and solidification, the enthalpy of the liquid-solid film $h_{\mathrm{ls}}{}^*$ includes the latent heat of fusion h_{fusion}:

$$h_{\mathrm{ls}}{}^* = h_{\mathrm{ls}} + \left(1 - \alpha^*{}_s\right)_h \tag{624}$$

where h_{ls} is the sensible enthalpy. The relative solid volume fraction $\alpha^*{}_s$ is defined as the portion of the volume of the liquid-solid film which the solid state occupies. In the enthalpy model, the relative solid volume fraction $\alpha^*{}_s$ is a function of temperature:

$$\alpha^*{}_s = \begin{cases} 1 & \text{if} \quad T^* < 0 \\ f(T^*) & \text{if} \quad 0 < T^* < 1 \\ 0 & \text{if} \quad 1 < T^* \end{cases} \tag{625}$$

where T^* is the normalized temperature that is defined as:

$$T^* = \frac{T - T_{\text{solidus}}}{T_{\text{liquidus}} - T_{\text{solidus}}} \tag{626}$$

The function $f(T^*)$ is called the fraction solid curve. For a linear dependence between $\alpha^*{}_s$ and T^*, the solidification path is defined as:

$$f(T^*) = 1 - T^* \tag{627}$$

The solidification path can also be specified as a table, in which case linear dependencies are assumed between the points in the table. The values of temperature in the table have to increase (decrease) monotonically as the values of the relative solid fraction decrease (increase) monotonically.

If melting and solidification take place at one temperature (that is, $T_{\text{solidus}} = T_{\text{liquidus}}$), a linear solidification path is assumed and a small temperature interval of 0.002 K is automatically introduced.

The physical properties of the liquid-solid phase can be constant, or can be a function of temperature and the relative solid volume fraction.

Removal of Solidified Mass

When the *Solidified Film Option* property is set to *Remove Solidified Film*, and the Morpher is activated, the Melting-Solidification model tracks the liquid-solid interface.

During every time step, a Solid Film Thickness Increment $\Delta\, h_s$ is computed, updated at every inner iteration during the segregated process. This value represents the increase in solidified thickness during the time step. $\Delta\, h_s$ is then used to construct the sources and sinks into the various transport equations.

The updates (at every inner iteration) are computed as:

$$\Delta\, h_s{}' = \begin{cases} -\dfrac{\Delta\, H(T) - \Delta\, H_{liq}}{\Delta\, H_{fus}} \left(h + \Delta\, h_s\right) & \text{if} \quad T \geq T_{liq} \\ h \cdot \alpha s & \text{otherwise} \end{cases} \tag{628}$$

with $\Delta\, h_s{}'$ the correction to $\Delta\, h_s$. This expression implies that (within each time step) the amount of solidified mass increases when the temperature is below the liquidus temperature, but is reduced if the temperature is above the liquidus temperature. This behavior is in accordance with intuition. However, the present implementation has a lower limit:

$\Delta\, h_s \geq 0$

and therefore this feature only allows mass to solidify; it does not allow the reverse process of melting. Once the solidified mass has been removed, it leaves the system and cannot be retrieved.

The solid film thickness increment can be related to a mass sink $\frac{kg}{m^2 s}$:

$$\dot{m} = \frac{\rho_{liq}\,\Delta\,h_s}{\Delta\,t} \tag{629}$$

which is used as a base for corresponding sources in the continuity, momentum, energy, and species transport equations in the film.

Morphing - Vertex Displacement

As described above, when the *Solidified Film Option* property is set to *Remove Solidified Film*, the solidified mass is removed from the film, leaving only liquid behind.

For each time step, the solidification is computed as follows:

- A mass sink is applied to the liquid film, corresponding to how much liquid is turning into solid mass.
- This mass sink is used to calculate the velocity of the solid front (that is, the velocity that the interface between solid and liquid is moving), also taking into account the density of the solid mass.
- This velocity is used to calculate the solid thickness increment $\Delta\,h_s$ for that time step; simply $\Delta\,h_s = V \times\ \mathrm{d}\,t$.

Typically, this solidified mass accumulates on the wall: it remains in place (it does not flow any further) and deforms the surface as seen by the gas and the remaining film. This effect is modeled by feeding the displacement $\Delta\,h_s$ into the Morpher.

For each time step, the displacement of every face is:

$$\overline{d}_{face} = \Delta h_s \mathbf{A} \tag{630}$$

where $\mathbf{A}$ is a unit vector, normal to the film surface and pointing into the gas (that is, the growth is assumed to happen in the direction normal to the surface). The displacement $\overline{d}_{face}$ is then interpolated to displacements $\overline{d}_{vert}$ on the vertices.

The Solid Time-step Factor value (K) acts as a multiplier for the solid front velocity (that is, $V = K \times V$). Consequently, within each time step, the solid thickness increment is increased by a factor K. This can be interpreted as a form of solid multiplier (that is, each 1kg/s of solidifying liquid is turned into K kg/s of solid). This could be considered as a mass source, although there are no transport equations solved for the solid: the solid is removed from the simulation and its effect is included only through the morphing of the boundary. The total physical time should be reduced by a factor K to get the same physical result.

Morphing - Smoothing

When the vertices are displaced according to the algorithm that is described above, the displacement can lead to roughness in the deformation, forming a zig-zag pattern. This roughness can cause instability issues in the fluid film transport.

To resolve this issue, an optional smoother is available. The smoother recomputes the displacement as a weighted average, including the displacements at the nearest neighbors. This smoothing can be repeated multiple times: set the Smoothing Steps property to the appropriate value.

Fluid Film Nomenclature

This section contains a guide to the terms that are used in fluid film modeling.

The subscript f denotes the fluid film values. The subscript g denotes a multi-component gas value and the superscript s denotes a surface value. The quantities without the subscript f are associated with the surrounding fluid.

ΔH^{vap}	enthalpy of vaporization
$\alpha^*{}_s$	relative solid volume fraction
ξ	local coordinate normal to the wall
λ_r	resonance wavelength
μ_f	film viscosity
σ	surface tension coefficient
ρ_f	film density
τ_{fs}	shear force at the film free surface
$v_{g,n}{}^s$	normal component of gas velocity at surface s
$v_{f,n}{}^s$	normal component of film velocity at surface s
B	Spalding transfer number
Bo_f	film Bond number
c_H	Minimum film thickness coefficient
c_D	Droplet diameter coefficient
c_B	Breakup rate scaling coefficient
c_e	wall boiling evaporation factor
D_g	gas molecular diffusion coefficient
D_f	film molecular diffusion coefficient
FR	force ratio
$\mathbf{f}_b$	body force
h_a	film height that is stripped (surface wave amplitude)
h_f	film thickness
h_{fusion}	latent heat of fusion
k_f	film thermal conductivity
k_g	gas thermal conductivity
L_b	break-up length
$\overline{m}_v$	evaporation rate

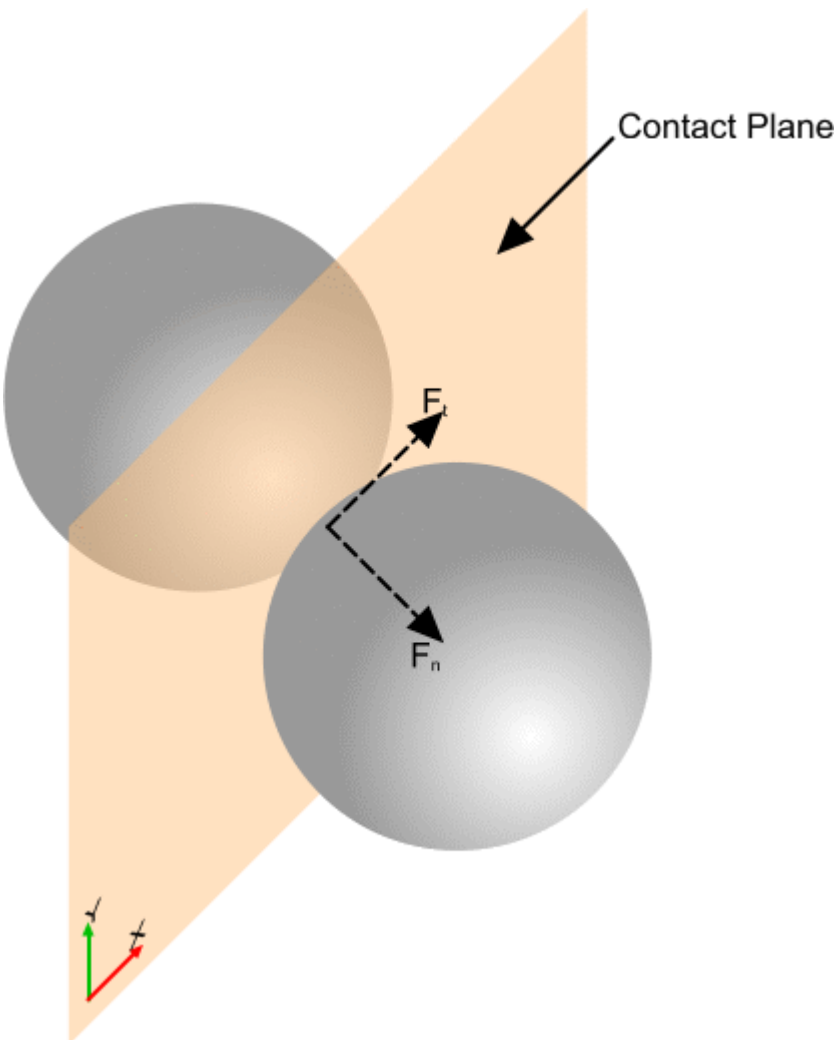
N_v	number of interacting components
N_f	number of components in fluid film
N_g	number of components in gas
$N_{P,f}$	number of inert components in fluid
$N_{P,g}$	number of inert components in gas
$\mathbf{n}$	wall surface unit vector
p_f	pressure
p^*	saturation pressure
Pr	Prandtl number
Re_f	film Reynolds number
r_d	droplet radius
s_m	mass source/sink per unit area
$\mathbf{s}_m$	momentum source
s_t	species transfer coefficient
$\mathbf{T}_f$	viscous stress tensor
T	temperature
T_L	Leidenfrost temperature
T_{sat}	saturation temperature
t_b	time scale
$\mathbf{v}_f$	film velocity
$\mathbf{v}_w$	wall velocity
$\overline{\mathrm{v}}_f$	average film velocity
We_f	film Weber number
We_{rel}	relative Weber number
${X_g}^s$	gas molar fraction
${X_f}^s$	fluid film molar fraction
x_S	stripping fraction
${Y_g}^s$	gas mass fraction
${Y_f}^s$	fluid film mass fraction

Using the Discrete Element Method

The Discrete Element Method (DEM) is an engineering numerical method to simulate motion of many interacting discrete objects that are typically solid particles. Although DEM modeling demands significant computing power, it provides detailed resolution that other methods cannot achieve.

Established by Cundall and Strack [103], the DEM model is an extension of the Lagrangian modeling methodology to include dense particle flows. The distinct characteristic of DEM is that inter-particle contact forces are included in the equations of motion. These forces cannot be ignored for highly loaded flows, that is, many interacting particles.

STAR-CCM+ uses a classical mechanics method to model DEM, and is based on soft-particle formulation where particles are allowed to develop an overlap. The calculated contact force is proportional to the overlap, as well as to the particle material and geometric properties.



Applications include modeling the granular flow of:

- Sand
- Food particles
- Powders
- Capsules
- Slurries

DEM in STAR-CCM+ is implemented within the Lagrangian framework and uses Lagrangian phase models to define the particles. In addition, the DEM model uses its own phase and phase interaction models for specific DEM particle properties and their behavior during contact with each other or wall boundaries.

The DEM model also uses Lagrangian injectors and its own lattice injector and random injector to introduce particles into the fluid region.

After running the solver, you can clear DEM particles from the region by activating the **Lagrangian DEM Solution Fields** option in the Clear Solution dialog.

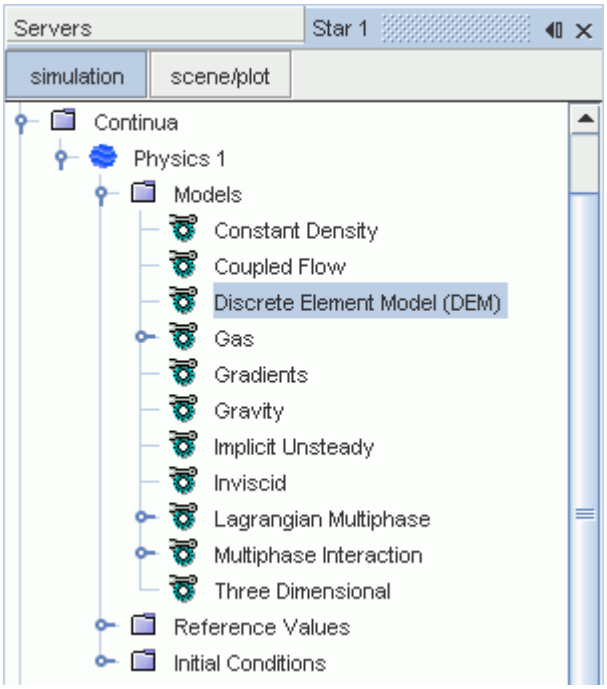
Regarding mesh motion, the DEM model is limited to motions where STAR-CCM+ can predict the position of the moving body with substep precision. In practice, this means rigid body motion only.

Considerations for the Discrete Element Method

When setting up a simulation that uses DEM, take the following points into consideration:

- Use a coarser mesh that typical CFD applications. If a particle is in a small cell and occupies most of its volume, it can cause high void fractions and thus solver instability.
- Typically, the most robust approach to solving the continuous phase with DEM is to use the coupled solver with incompressible, ideal gas.
- To reduce computation times, reduce the Young's Modulus of the particles. However, this change in physical properties compromise the accuracy and validity of your results.

The DEM model is represented in the STAR-CCM+ physics continuum by the **Discrete Element Model (DEM)** node.



Discrete Element Model Properties

None.

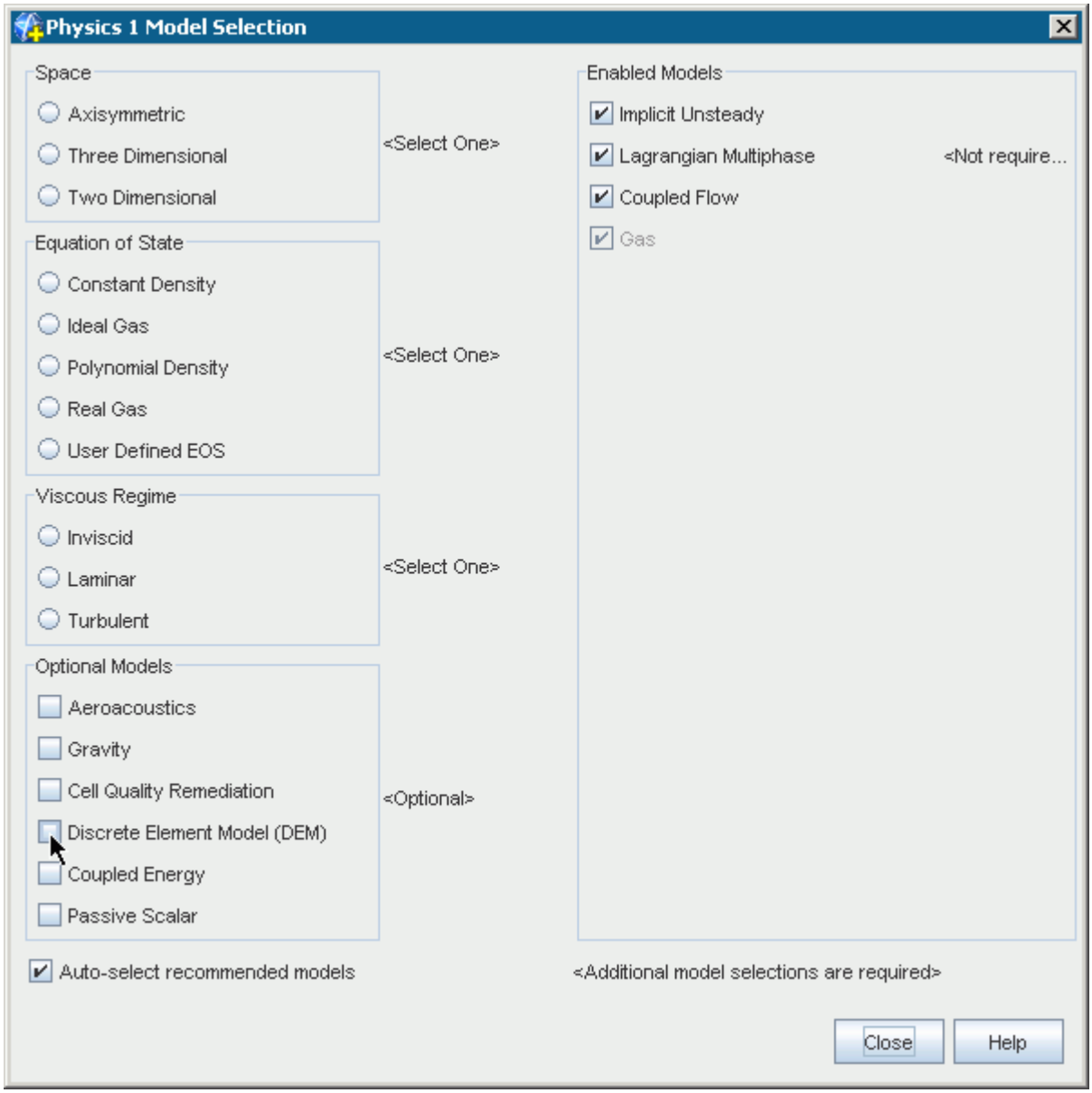
Selecting the Discrete Element Model

As a prerequisite, select the following continuum models:

- In the *Physics Model Selection* dialog, select the Lagrangian Multiphase model.
- Select the Implicit Unsteady model.

Then, to select the DEM model:

- In the *Optional Models* group box, select the **Discrete Element Model (DEM)** option.



If you set up Lagrangian phases with DEM particles, clearing the DEM model automatically deletes the Lagrangian phases, because they no longer serve any purpose in the simulation. To prevent the loss of those phases, clear the DEM Particles phase model in each Lagrangian phase before clearing the DEM continuum model.

Contents:

DEM Workflow
Modeling Discrete Element Method Particles
Setting Phase Boundary Conditions
Modeling Drag Force in the Discrete Element Method
Modeling Drag Torque in the Discrete Element Method
Modeling Lift Force in the Discrete Element Method
Modeling Erosion in the Discrete Element Method

Modeling Phase Interactions in the Discrete Element Method
Modeling Artificial Viscosity
Defining Injectors in the Discrete Element Method
What Is the Discrete Element Method Solver?
Guidelines for DEM
Discrete Element Model Formulation
Discrete Element Method Bibliography
Discrete Element Method Field Functions Reference
Migrating from EDEM
Post-Processing DEM Data

DEM Workflow

Set up a DEM simulation as you would other Lagrangian simulations, including DEM particles of suitable shape and source.

Use these steps:

1. Within the physics continuum, select the Lagrangian Multiphase model and the pre-requisite models for it in the following order:

Group Box	Model
Material	Any material except solids.
Flow	**Coupled Flow** or **Segregated Flow** **Gradients** (selected automatically)
Optional Models	**Lagrangian Multiphase**
Time	**Implicit Unsteady**
Optional Models	**Discrete Element Model (DEM)** **Multiphase Interaction** (selected automatically)

2. Create as many Lagrangian phases as you require for the DEM simulation:
   a) Open the **Lagrangian Multiphase** node to expose the **Lagrangian Phases** manager node.
   b) Right-click the **Lagrangian Phases** manager node and select **New**.
      A new Lagrangian phase node appears with the default name `Phase 1`.
   c) Rename the node if desired.
3. For each Lagrangian phase, define the particle physics and material as follows:
   a) Right-click the **[phase]** > **Models** node and click **Select Models...**
   b) In the phase model selection dialog, within *Particle Type*, select **DEM Particles**.
   c) Continue by selecting a DEM Particle type. See DEM Particle Types for a list of choices.
   d) Choose further phase models as necessary to simulate the physics present in the scenario you are modeling. Include an appropriate material model.
   e) When done with selecting models, close the dialog, and review the phase model nodes in the tree. Some models require further selections in their properties.

   f) Set the phase material properties.

4. Create phase interaction nodes for interactions between all particle phases including each phase with itself and each phase with each kind of boundary. For example, for particle phases A and B confined by a wall, create interactions for A-B, A-A, B-B, A-wall, and B-wall.
   a) Open the **Multiphase Interaction** node to expose the **Phase Interactions** manager node.
   b) Right-click the **Phase Interactions** manager node and select **New**.

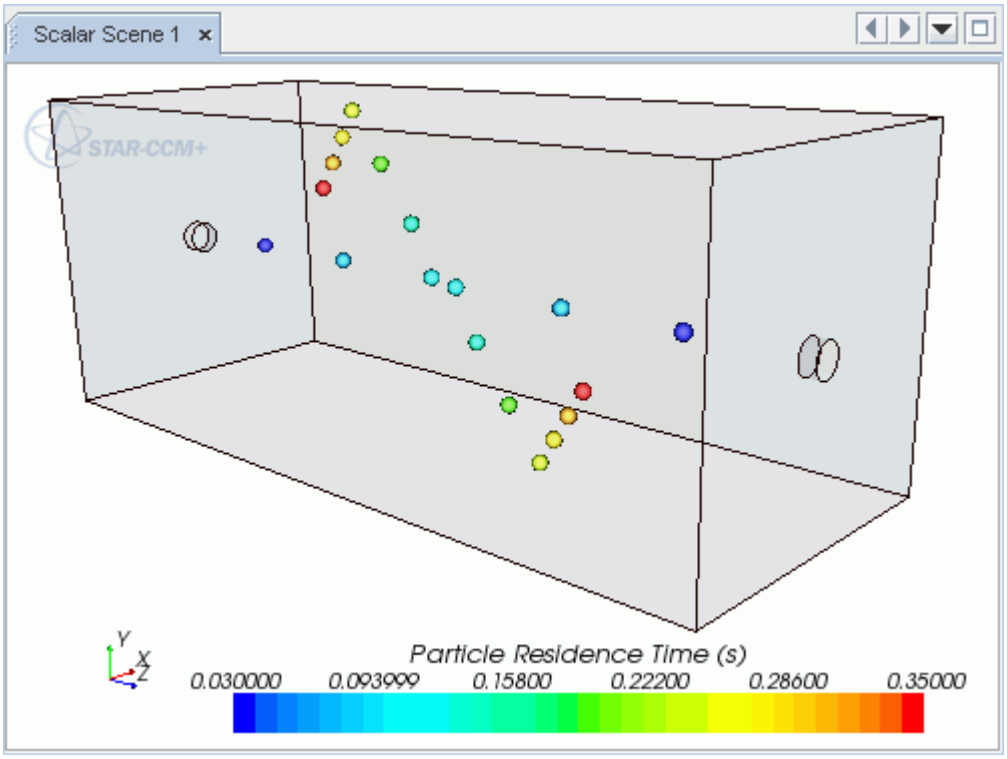
      A new interaction node appears with the default name `Phase Interaction 1`.
   c) Rename the node.
5. For each phase interaction, define the contact model:
   a) Select **[phase interaction]** > **Models** > **DEM Phase Interaction** and set the *First phase* and *Second phase* properties of the interaction. Boundaries can be used as second phases. See Modeling Phase Interactions in the Discrete Element Method.
   b) Right-click the **[phase interaction]** > **Models** node and click **Select Models...**
   c) In the phase model selection dialog, within *Phase Interaction Topology*, select **DEM Phase Interaction**.
   d) Select a model within *DEM Contact Model*.
   e) Choose further interaction models as necessary to simulate the physics present in the scenario you are modeling.
6. Define the source of DEM phases within the simulation: create injectors and set their properties. See Working with Injectors. In most cases, the Random Injector is the best choice.

   The following injector conditions are specific to DEM:
   - Angular Velocity Specification
   - Injection Time Rate Specification
   - Injection Volume Correction Specification
   - Lattice Structure
   - Orientation Specification
   - Overlap Specification
7. Adjust solver parameters to improve performance and scalability. See What Is the Discrete Element Method Solver? with particular attention to the *Reordering Frequency* and *Skin* properties.
8. Prepare for post-processing. DEM post-processing includes all features of Lagrangian post-processing, but also allows reporting on features of individual particles, such as contact between particles, particle shape, and rotation. See Simulation History Phase Data and Discrete Element Method Field Function Reference.
9. Run the simulation.
10. Review the results. These can include instantaneous data, particle track data, and boundary sampling, as with Lagrangian post-processing, but can also include data on particle contacts, angular momentum and kinetic energy, collision history, and changes in particle shape.

Modeling Discrete Element Method Particles

The DEM Particles model specifies that the Lagrangian phase represents solid particles and the Discrete Element Method (DEM) models the particle behavior within the simulation.

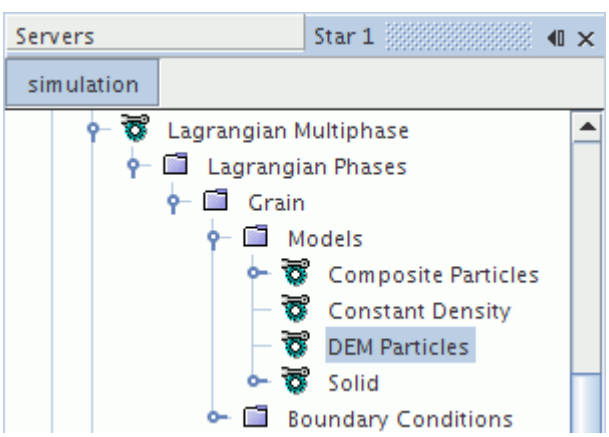
The momentum balance formulation determines how the particles interact with each other and with solid boundaries. The image below shows the effect of particle-particle collisions and particle-wall collisions.



If a particle spans 3 or more mesh partitions (due to small cells and a high number of computational processes), the solution can miss particle-particle or particle-wall collisions, or record them late. To avoid this error, either use a coarser mesh, or increase the size of mesh partitions by reducing the number of processes.

As the DEM model simulates solid particles, the only available Lagrangian phase material models are solid, multi-component coal, and multi-component solid. Similarly, the only available Lagrangian phase equation of state models are constant density and polynomial density.

The DEM Particles model is represented in the Lagrangian phase by the **DEM Particles** node.



DEM Particle Types

DEM Lagrangian phases include phase models that allow you to select the type of the particles as:

- Spherical particles—Single spherical particles.
- Coarse Grain particles—Larger spherical parcels each representing multiple smaller particles. This model can reduce computational cost.
- Composite particles—Non-spherical particles composed of spheres joined by fixed bonds. See also Sphere-Filled Parts.
- Particle clumps—Non-spherical particles composed of spheres joined by breakable bonds to model fragmentation.
- Cylindrical particles—Exact cylinders that are computationally cheaper and can be more precise than approximations of composite spheres.

You can specify the structure of particle clumps or composite particles by either of two methods:

- Individually Composed particles—You specify the structure using the particle shapes editor.
- Sphere-Filled Parts—You specify the number of spherical particles to use and STAR-CCM+ uses them to fill out a specified part.

While the Particle Clumps model defines clusters of particles that are injected with the inter-particle bonds already formed, the Parallel Bonds model forms bonded clusters from colliding particles after injection. For both these models, subsequent particle interaction and cluster breakup behaves in the same way.

DEM Particles Properties

None.

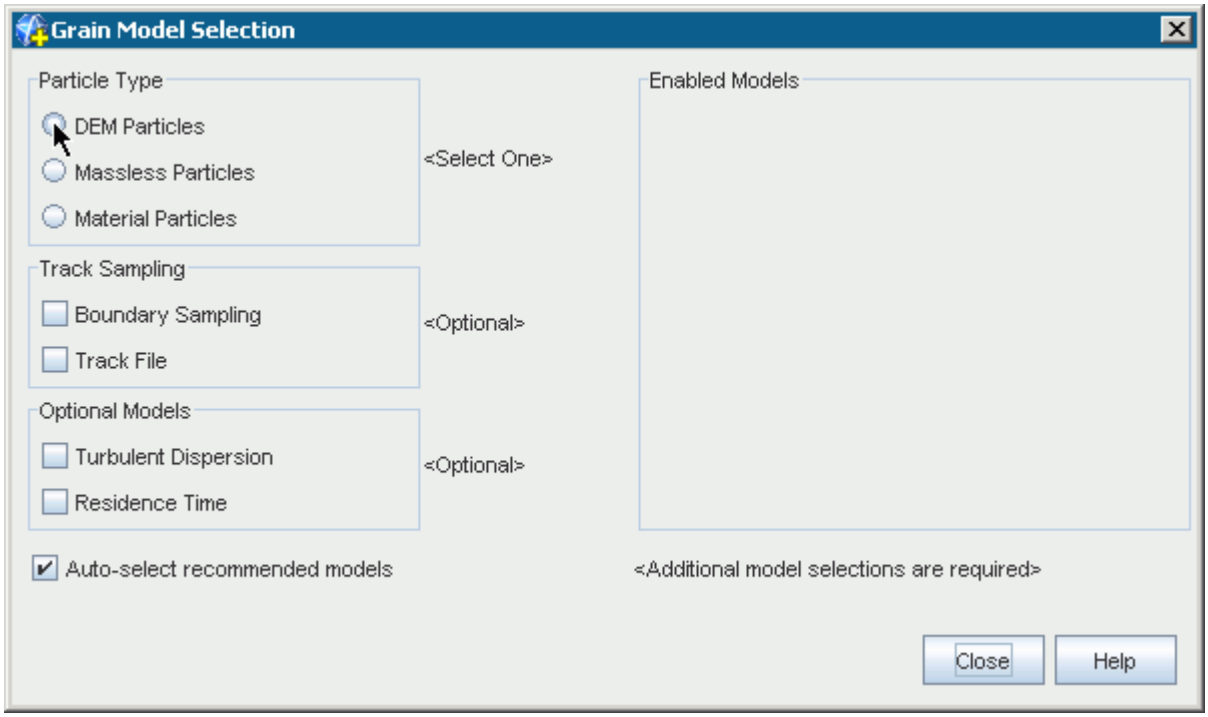
Selecting the DEM Particles Model

As a prerequisite, select the following continuum models:

- In the *Physics Model Selection* dialog, select the Discrete Element Model (DEM) model.

Then, to select the **DEM Particles** phase model for a newly created Lagrangian phase:

- In the *Phase Model Selection* dialog, select the **DEM Particles** model from the *Particle Type* group box.



Contents:

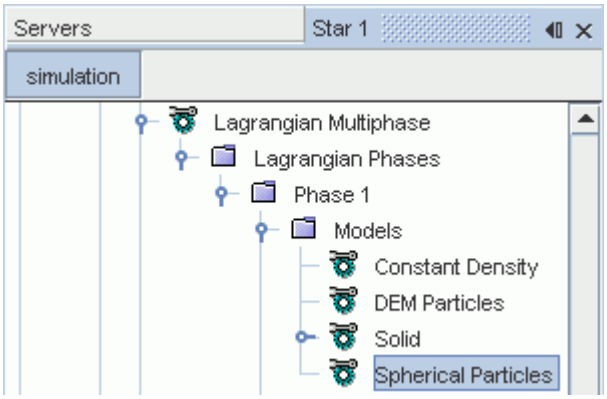
What Is the Spherical Particles Model?
Coarse Grain Particle Models
What Is the Composite Particles Model?
What Is the Particle Clumps Model?
Working with Individually Composed Shapes
Working with Sphere-Filled Parts
The Cylindrical Particles Model
The Capsule Particles Model
The Polyhedral Particles Model

The Spherical Particles Model

The Spherical Particles model uses a single sphere to represent the particle.

If you wish to model non-spherical particles, select the Composite Particles or Particle Clumps.

The Spherical Particles model is represented in the Lagrangian phase by the **Spherical Particles** node.



Spherical Particle Model Properties

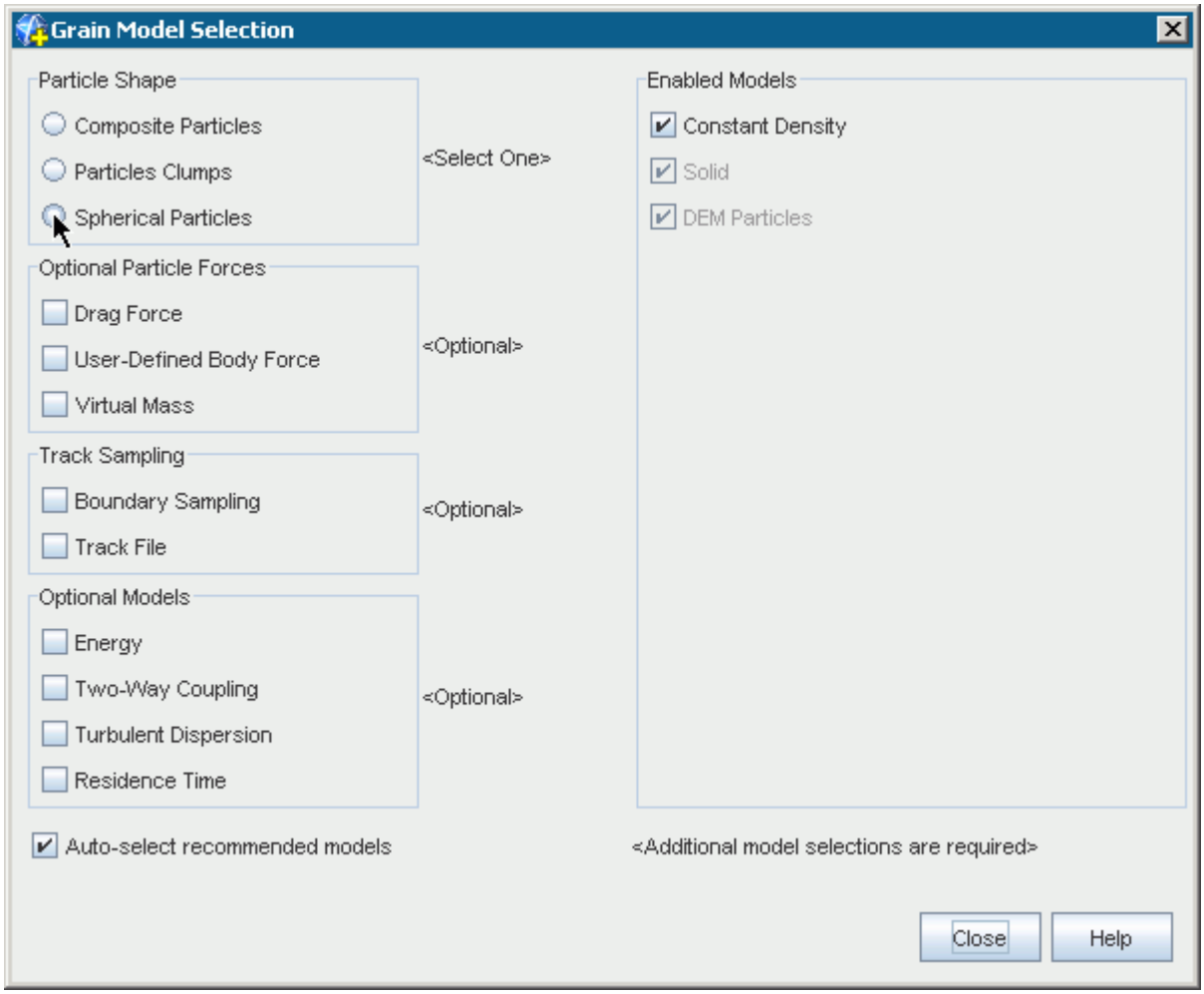
Moment of Inertia Scaling	Provides a scaling factor for each axis of the moment of inertia of the particle. This can be used to simulate effects such as uneven distribution of mass within the particle, or the dynamical response of a non-spherical particle. The scaling factor has an allowed range on each axis of 1E-2 to 1E5. The default value is `[1.0, 1.0, 1.0]`.

Selecting the Spherical Particles Model

- In the *Physics Model Selection* dialog, select the Discrete Element Model (DEM) model.
- In the *Phase Model Selection* dialog, select the DEM Particles model.

Then, to select the **Spherical Particles** phase model:

- In the *Phase Model Selection* dialog, select the **Spherical Particles** model from the *Particle Shape* group box.



Coarse Grain Particle Models

A common problem with using the Discrete Element Method is the high computational cost of simulating a realistic number of particles. One way to reduce this cost is to use one of the Coarse Grain Particle models.

In these models, each simulated particle is a parcel that represents multiple smaller identical particles. Both models calculate fluid-particle interactions on a single representative particle and apply the results to all of the smaller particles in the parcel. For particle-particle interactions:

- The Coarse Grain Particle model calculates the contact forces on a single representative particle pair, one from each interacting parcel, then uses these forces in a number of contact events equal to the number of particles in the parcel with the most particles.
- The Parcel Contact Coarse Grain model treats each parcel as a single large particle and calculates the contact between them. This method uses a larger time-step and so gives a higher computational speed at the cost of a small loss of fidelity in describing the details of individual contacts.

The number of individual particles per coarse-grained particle depends on injector conditions and values, set by *Flow Rate Specification*, *Particle Size Specification*, and *Mass Flow Rate*. The number does not have to be an integer.

The Parcel Coarse Grain Particle model is best used for fluidized bed applications, where accuracy of particle-fluid interactions is more important than accuracy of particle-particle interactions.

Do not use the Coarse Grain Particle models when the mesh size is smaller than the size of the coarse-grained particle.

Model Names	**Coarse Grain Particle**	
	Parcel Contact Coarse Grain	
Theory	See Contact Force Modeling.	
Provided By	**[phase]** > **Models** > *Particle Shape*	
Example Node Path	**[phase]** > **Models** > **Coarse Grain Particle** or **Parcel Contact Coarse Grain**	
Requires	Basic DEM setup In **[phase]** > **Models**: *Particle Type*: **DEM Particles** *Particle Shape*: **Coarse Grain Particle** or **Parcel Contact Coarse Grain**	
Activates	Field Functions	**Particle Diameter**. See Lagrange Multiphase Field Function Reference.

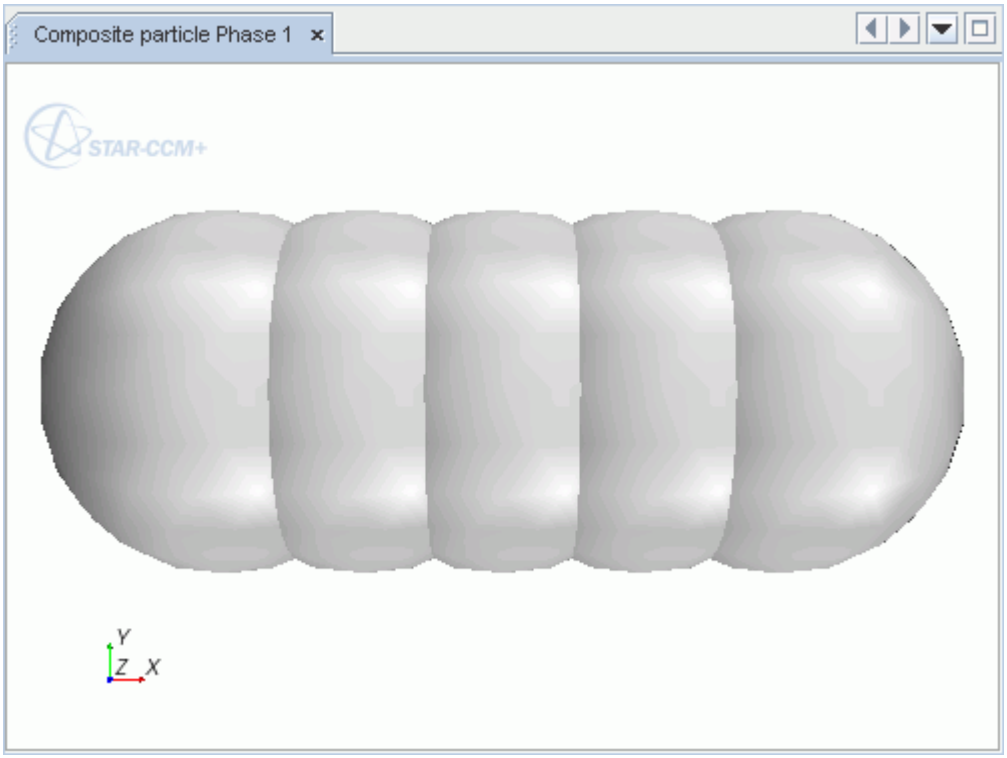
The Composite Particles Model

The Composite Particles model uses multiple shapes (spheres) to represent non-spherical particles. The shapes are fixed together and do not separate during the simulation.

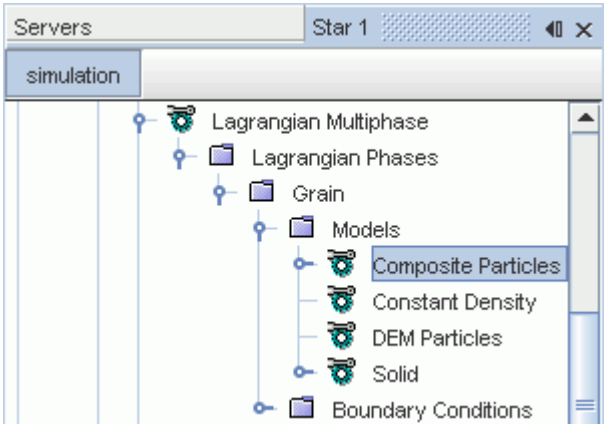
If you wish to model particle bonds that can break, select the Particle Clumps model.

You define the arrangement of spheres within a composite particle using the particle shapes editor to compose the shape yourself, or using the Update from Part option on the **Composite Particles** node to have STAR-CCM+ do it.

The image below shows an example of a composite particle that uses five spheres to represent its shape.



The Composite Particles model is represented in the STAR-CCM+ Lagrangian phase by the **Composite Particles** node. Select the node to access the properties of the model. Right-click the node to access the pop-up menu of the Composite Properties model.



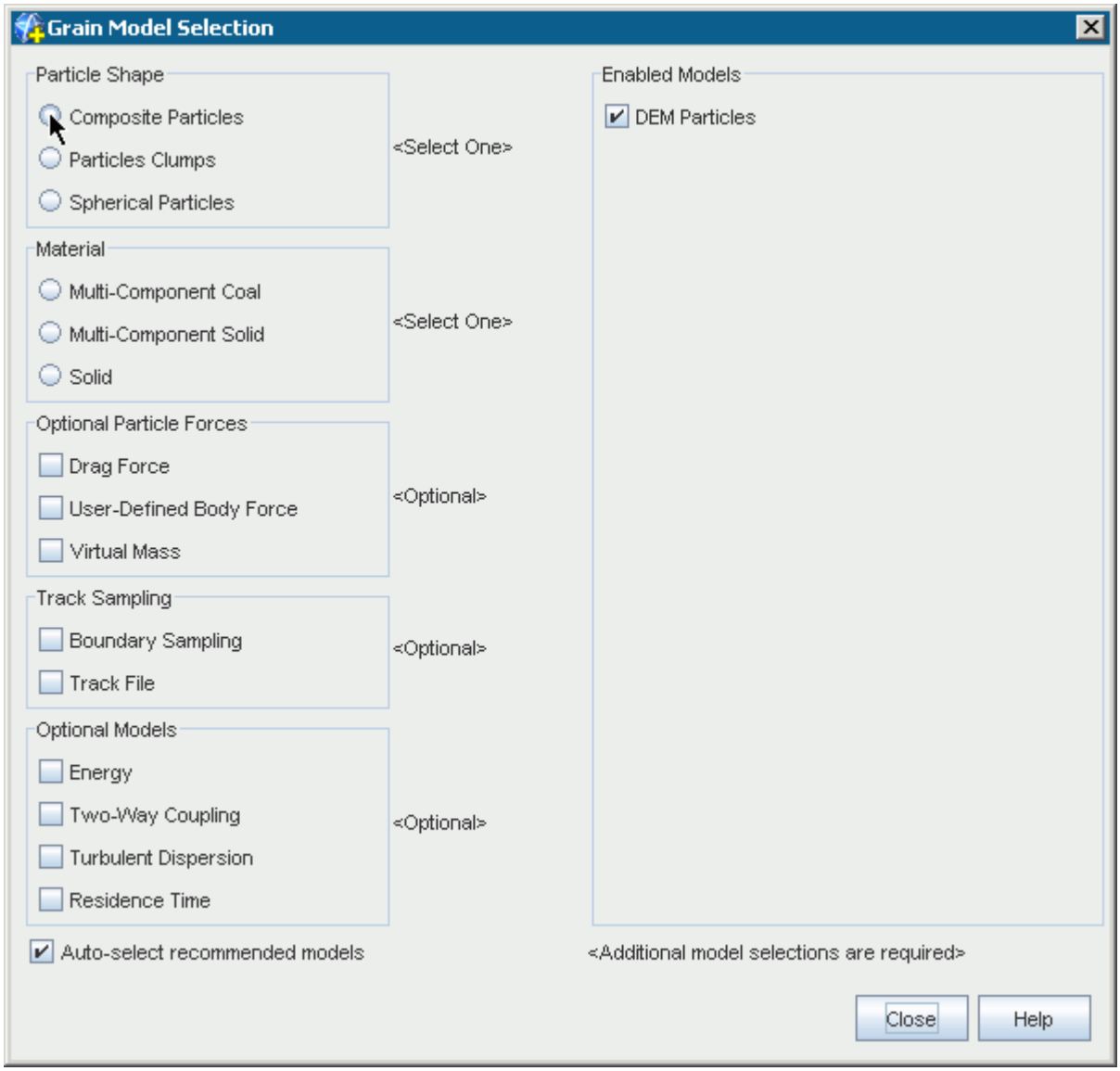
Selecting the Composite Particles Model

As a prerequisite, select the following continuum and phase models:

- In the *Physics Model Selection* dialog, select the Discrete Element Method (DEM) model.
- In the *Phase Model Selection* dialog, select the DEM Particles model.

Then, to select the **Composite Particles** phase model:

- In the *Phase Model Selection* dialog, select the **Composite Particles** model from the *Particle Shape* group box.



Thereafter, you can use the particle shape editor or the Composite Particles Part property to create shapes.

The Particle Clumps Model

The Particle Clumps model uses multiple shapes (spheres) to represent non-spherical particles. You can use the particle shape editor to define the initial particle assembly.

The join between these spheres is modeled using the Bonded Particles phase interaction model. If you wish to model non-spherical, fixed particles, select the composite particles model.

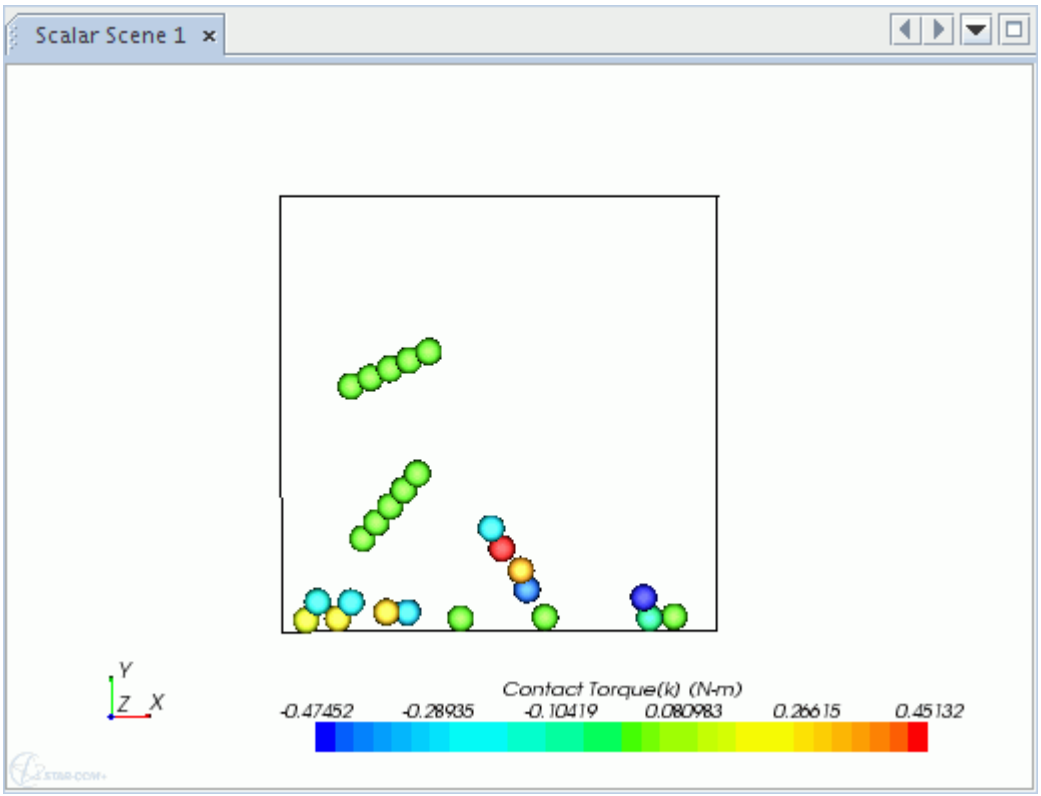
In contrast to the composite particles model, the particle clumps model defines particles as assemblies of spheres that are connected through parallel bonds. The composite particles model only treats the individual

spheres of a particle as separate entities for contact detection, where contacts between particles must be resolved separately. In the remainder of the calculation, the particle is a single object. With the particle clumps model, each sphere in the assembly is treated as an individual particle, with a separate position, velocity, angular velocity, and orientation that are maintained by the system.

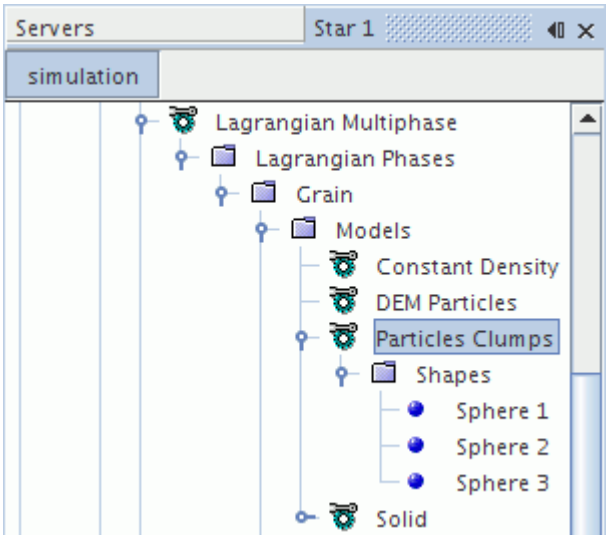
When the injector injects a particle, bonds immediately form between particles in contact. The initial configuration of the particle clump is in a state of equilibrium; that is, all the contact forces are balanced.

To make the bonds between particles breakable, select either the Simple Failure model or the Constant Rate Damage model. When the bond breaks, the particles interact according to the Hertz-Mindlin law. The bonding and grain forces do not act in parallel in this case. This fact allows STAR-CCM+ to define the initial particle shape without the excessive limitations that the small contact overlap of the soft particle model imposes.

The image below shows an example of particle clumps breaking up as they strike a wall boundary.



The Particle Clumps model is represented in the STAR-CCM+ Lagrangian phase by the **Particle Clumps** node. Select the node to access the properties of the model. Right-click the node to access the pop-up menu of the Particle Clumps model.



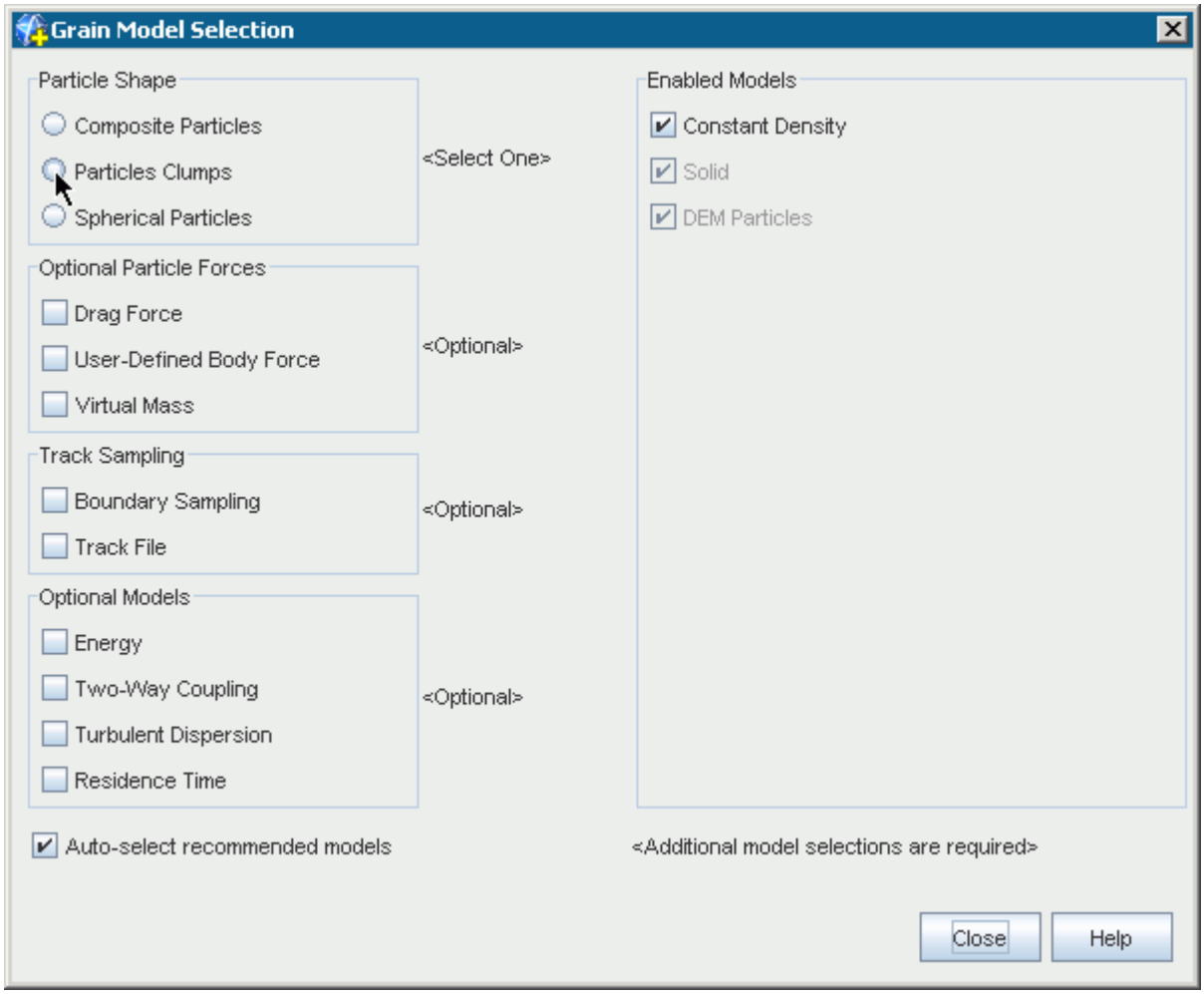
Selecting the Particle Clumps Model

As a prerequisite, select the following continuum and phase models:

- In the *Physics Model Selection* dialog, select the Discrete Element Model (DEM) model.
- In the *Phase Model Selection* dialog, select the DEM Particles model.

Then, to select the **Particle Clumps** phase model:

- In the *Phase Model Selection* dialog, select the **Particle Clumps** model from the *Particle Shape* group box.



Working with Individually Composed Shapes

STAR-CCM+ allows you to define composite particles or particle clumps as an arrangement of spheres. You define the relative dimensions and positions of the spheres in an arbitrary unit system, as STAR-CCM+ scales the final particle that is injected into the system according to the injector properties. The scale factor is:

$$\left(\frac{V_{injector}}{V_{particle}}\right)^{\frac{1}{3}}$$

where:

- $V_{particle}$ is the particle volume computed using Monte Carlo integration
- $V_{injector}$ is the injector volume

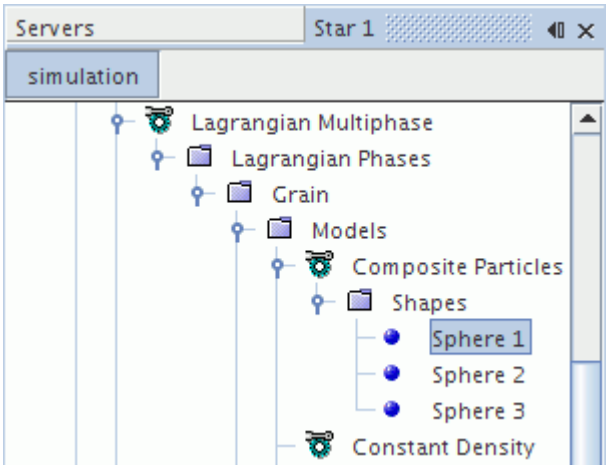
The injector volume is the volume of a sphere with the diameter specified at the injector:

$$V_{injector} = \pi\left(\frac{D^3_{injector}}{6}\right)$$

where:

- $D_{injector}$ is the injector diameter

Each sphere is represented in the STAR-CCM+ Lagrangian phase by a **Sphere** node, under the **Shapes** manager. Select a node to access the properties of the corresponding sphere.



Properties of Composite Particles and Particle Clumps

Number of Spheres	Read-only property reporting the number of spheres in the composite particle or particle clump.
Principal Moment of Inertia	Read-only property reporting the principal moment of inertia of the particle, which is composed of user-defined shapes. This property is used when calculating the momentum balance (see Eqn. (2971)).
Volume	Read-only property reporting the volume of the particle.
Area	Read-only property reporting the area of the particle.
Sphericity	Read-only property reporting the sphericity of the particle. This property is used when calculating the Haider and Levenspiel drag coefficient (see Eqn. (2987)).

Properties of Composite Particles

The following properties are available with composite particles but not with particle clumps.

Part	The part selected to be filled with spheres. See Working with Sphere-Filled Parts.
Number of Spheres to Generate	The number of spheres to use when filling a part. See Creating Sphere-Filled Parts.
Moment of Inertia Scaling	Provides a scaling factor for each axis of the moment of inertia of the particle. This can be used to simulate effects such as uneven distribution of mass within the particle, or the dynamical response of a non-spherical particle. The scaling factor has an allowed range on each axis of 1E-2 to 1E5. The default value is `[1.0, 1.0, 1.0]`.

Principal Moments of Inertia as Scaled	Read-only property reporting the product of **Principal Moment of Inertia** and **Moment of Inertia Scaling**. Not available with Particle Clumps.
Projected property resolution	Specifies the degree of resolution for particle cross-section and length relative to the direction of flow in the particle's frame of reference, expressed as an integer ranging from 0 to 5. The default value is 0. This is an expert property. Raise the value above the default of 0 when fluid drag is important. Computational cost increases directly with the value. In most cases, a value of 1 or 2 is sufficient.

Composite Particles and Particle Clumps Pop-Up Menu

Update from Part	For composite particles only. Creates (or re-creates) a composite particle that is based on the selected shape, using the number of spheres currently specified by the **Number of Sphere to Generate** property.
Update Properties	Updates the particle properties that are based upon the current geometry, that is, the user-defined shapes.
Show model	Displays a representation of the particle in the *Graphics* window.

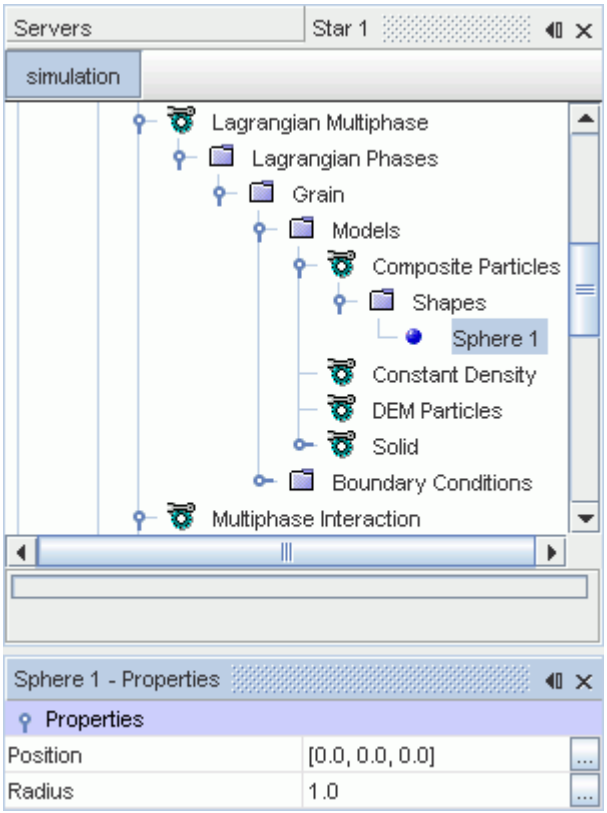
Sphere Properties

Position	Specifies the center-point of the sphere using Cartesian coordinates that are relative to an arbitrary coordinate system. This property is used to locate the spheres relative to one another.
Radius	Specifies the radius of the sphere with an arbitrary unit. The actual size of the sphere is not specified here, as the entire composite particle or clump is scaled to match the injector.

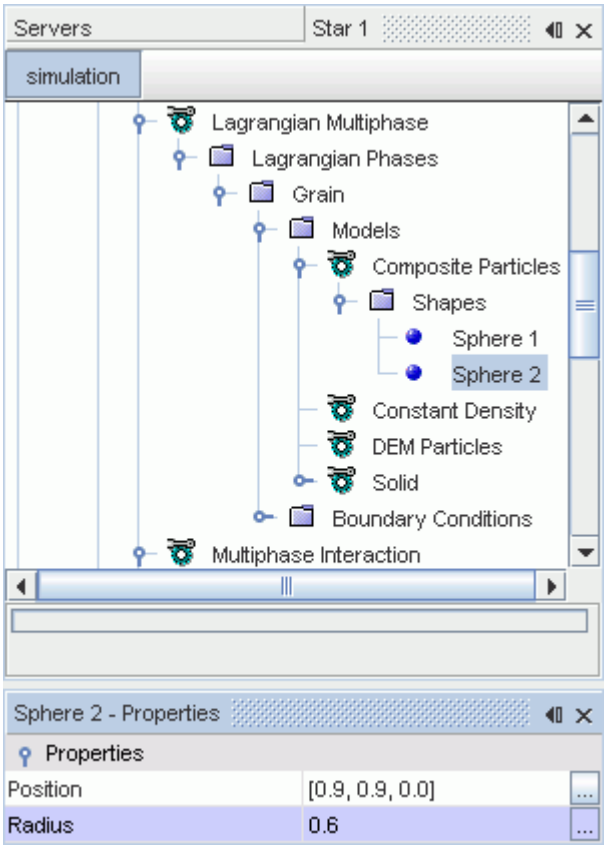
Example of Creating a Particle Using Shapes

The following example uses shapes to create a composite particle with a non-spherical shape. A similar procedure can also be used to create a particle clump.

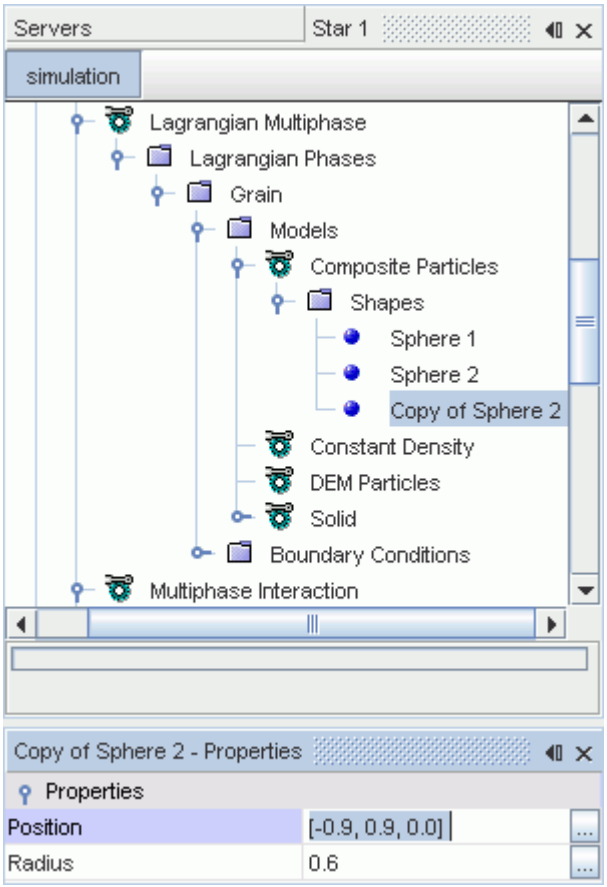
First, right-click the **Phase** > **Models** > **Composite Particles** > **Shapes** manager and select **New** to create a sphere. In the *Properties* window, set the *Radius* and *Position* properties. As the position and radius are arbitrary, with the spheres defined relative to one another, you can typically define the first sphere with the default values.



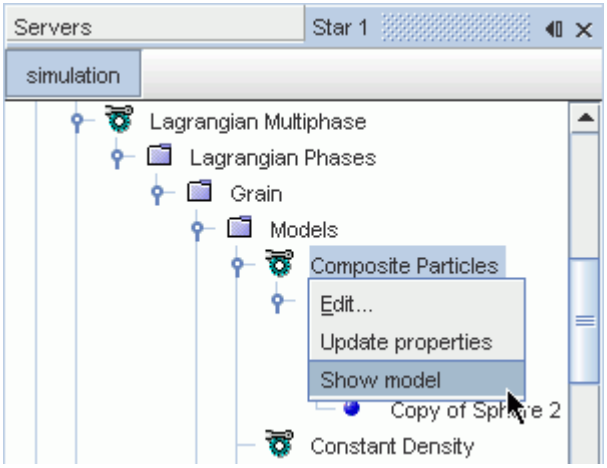
You can continue to create numerous spheres to represent the particle. Define each particle relative to the first.



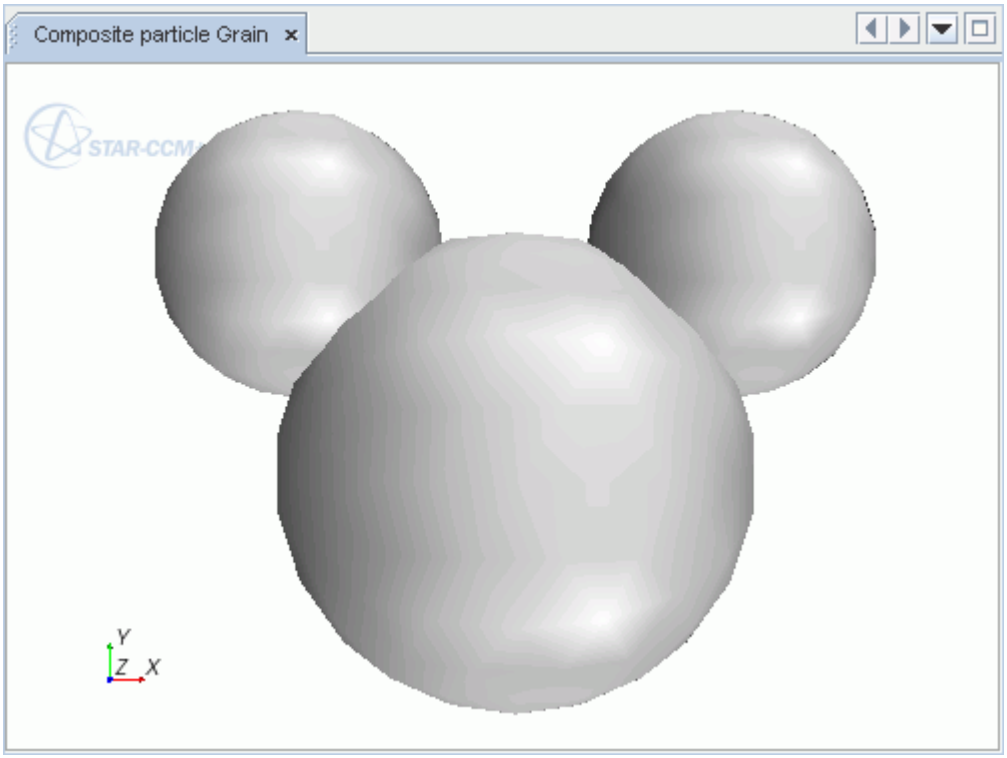
If two particles have similar properties, you can copy and paste the node to create an identical sphere and then change its properties.



To visualize the particle, right-click the **Composite Particles** node and select **Show model**.



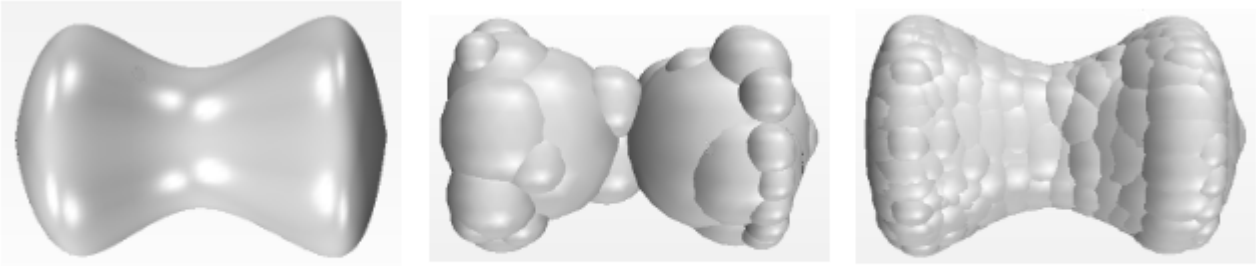
The *Graphics* window then displays the particle as a *Composite particle* scene.



Working with Sphere-Filled Parts

The composite particle method simulates particles with complex shapes by arrangements of spheres that are kept in fixed relationships to each other.

You can create a composite particle by selecting a previously defined part and having STAR-CCM+ fill the part with a specified number of spheres.



Original part. Composite particle, small number of spheres. Composite particle, large number of spheres.

This method works well for largely convex objects such as rocks or grains, but is more likely to fail for complex concave objects.

Simulating sharp edges and corners requires small spheres, but small spheres produce a small time-step in a DEM simulation—and hence higher computation time. To avoid small time-steps, you can edit individual spheres to the largest size acceptable to you, after STAR-CCM+ creates the initial composite particle.

Every sphere that is created by this method has some part of the surface exposed. As a result, the interior of the particle is hollow in some cases. Spheres that are completely embedded in the particle have no effect on simulation.

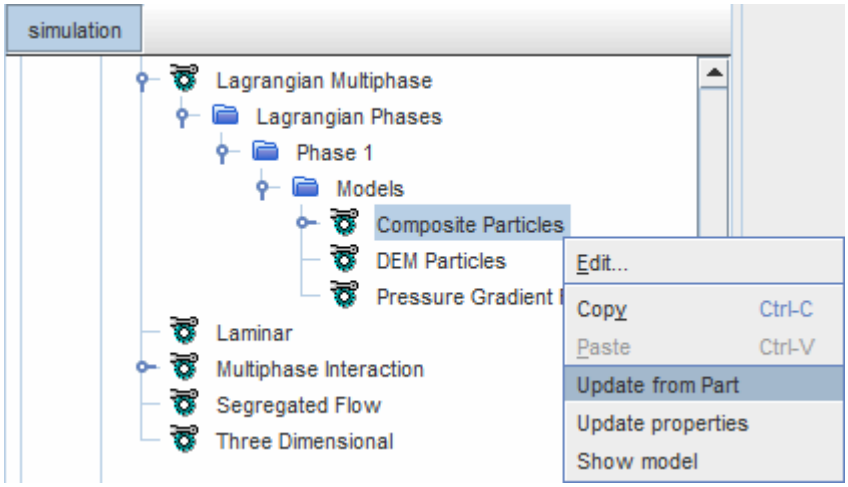
The method does not always produce a symmetrical composite particle when filling a symmetrical part. If symmetry considerations are important in the simulation, try a different number of spheres and see if the result is more symmetrical, or use the particle shapes editor instead.

A good practice is to start with a small number of spheres and see what the resulting representation looks like. If the representation is not satisfactory, then increase the number of spheres.

Creating Sphere-Filled Parts

To create a composite particle by filling a part with spheres:

1. Select the **Composite Particles** node under the **Models** node for the **Lagrangian Phases**.
2. In the *Composite Particles - Properties* window, edit **Part**.
3. In the *Edit* dialog, select the part that you want to use as a composite particle, then click **OK**.
4. In the *Composite Particles - Properties* window, set the *Number of Spheres to Generate* property.
5. Right-click the **Composite Particles** node and select **Update from Part**. STAR-CCM+ creates a composite particle that is based on the selected shape, using the specified number of spheres.



6. To visualize the particle, right-click the **Composite Particles** node and select **Show model**.

The Cylindrical Particles Model

The Cylindrical Particles model allows you to use cylindrical particles of specified size and proportion rather than particles composed of combinations of spheres.

Using cylindrical particles to model tablets, pellets, and similar particle shapes can be simpler, computationally less expensive, and more accurate than using combinations of spheres, depending on the individual case.

Provided By	**[phase]** > **Models** > *Particle Shape*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Lagrangian Multiphase** > **Lagrangian Phases** > **[phase]** > **Models** > **Cylindrical Particles**	
Requires	In **[physics continuum]** > **Models**: *Optional Models*: **Discrete Element Method (DEM)** In **[phase]** > **Models**: *Particle Type*: **DEM Particles**	
Properties	*Moment of Inertia Scaling* Provides a scaling factor for each axis of the moment of inertia of the particle. This can be used to simulate effects such as uneven distribution of mass within the particle, or the dynamical response of a non-spherical particle. The scaling factor has an allowed range on each axis of 1E-2 to 1E5. The default value is `[1.0, 1.0, 1.0]`.	
Activates	Region Inputs	**Physics Conditions** > **Energy Source Option** **Physics Values** > **Axis** and **Motion Specification**
	Injector Inputs	See Cylindrical Particles Injector Inputs.
	Field Functions	Cylinder Height Cylinder Radius Particle Projected Area Particle Projected Length

Cylindrical Particles Injector Inputs

The Cylindrical Particles model activates the following injector condition:

Orientation Specification

Sets the initial orientation of the particle as it is injected.

The Cylindrical Particles model activates the following injector values:

Cylinder Height

Sets the cylinder heights using a selection of methods.

Cylinder Height/Radius Ratio

Sets the ratio of cylinder height to radius using a selection of methods.

Cylinder Radius

Sets the cylinder radii using a selection of methods.

Particle Orientation

Sets the initial orientation of the particle as it is injected.

The Cylindrical Particles model is incompatible with the Random Injector using the **Maximum Packing** option.

Contact Force Modeling for Cylindrical Particles

Use Linear Spring to model contact force, and set the **Spring Stiffness** methods to **Constant**. The default value for stiffness, `100000.0 N/m`, works for most situations. However, for cylinders with very small aspect ratios, use higher values to avoid large overlap between cylinders, especially cylinders penetrating the boundary.

Compatibility with Other Models

The following models are incompatible with the Cylindrical Particles model. If any of them are selected when you select Cylindrical Particles, STAR-CCM+ de-selects the incompatible models and posts the message `Removing incompatable model(s) in` *[Phase Interaction]*.

- Artificial Viscosity
- Hertz Mindlin
- Linear Cohesion
- Linear Spring, with either *Normal Spring Stiffness Method* or *Tangential Spring Stiffness Method* set to **Particle Material Based**.
- Parallel Bond
- Rolling Resistance, using the **Displacement Damping** or **Constant Torque** method
- Walton Braun

See Also:

Injector Conditions Reference

Injector Values Reference

The Capsule Particles Model

The Capsule Particles model allows you to use capsule-shaped particles of specified size and proportion rather than polyhedral particles or particles composed of combinations of spheres.

Using capsule particles to model the shape of a cylinder with hemispherical caps can be simpler, computationally less expensive, and more accurate than using a polyhedral approximation or a combinations of spheres, depending on the individual case.

Provided By	**[phase]** > **Models** > *Particle Shape*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Lagrangian Multiphase** > **Lagrangian Phases** > **[phase]** > **Models** > **Capsule Particles**
Requires	In **[physics continuum]** > **Models**: *Optional Models*: **Discrete Element Method (DEM)** In **[phase]** > **Models**: *Particle Type*: **DEM Particles**

Properties	*Moment of Inertia Scaling* Provides a scaling factor for each axis of the moment of inertia of the particle. This can be used to simulate effects such as uneven distribution of mass within the particle, or the dynamical response of a non-spherical particle. The scaling factor has an allowed range on each axis of 1E-2 to 1E5. The default value is `[1.0, 1.0, 1.0]`.	
Activates	Region Inputs	**Physics Conditions** > **Energy Source Option** **Physics Values** > **Axis** and **Motion Specification**
	Injector Inputs	See Capsule Particles Injector Inputs.
	Field Functions	Cylinder Height Cylinder Radius Particle Projected Area Particle Projected Length

Capsule Particles Injector Inputs

The Capsule Particles model activates the following injector condition:

Orientation Specification

Sets the initial orientation of the particle as it is injected.

The Capsule Particles model activates the following injector values:

Cylinder Height

Sets the heights of the cylindrical sections using a selection of methods.

Cylinder Height/Radius Ratio

Sets the ratio of height to radius of the cylindrical sections using a selection of methods.

Cylinder Radius

Sets the radii of the cylindrical sections using a selection of methods.

Particle Orientation

Sets the initial orientation of the particle as it is injected.

The Capsule Particles model is incompatible with the Random Injector using the **Maximum Packing** option.

Contact Force Modeling for Capsule Particles

Use Linear Spring to model contact force, and set the **Spring Stiffness** methods to **Constant**. The default value for stiffness, `100000.0 N/m`, works for most situations. However, for capsules with very small aspect ratios, use higher values to avoid large overlap between capsules, especially capsules penetrating the boundary.

Compatibility with Other Models

The following models are incompatible with the Capsule Particles model. If any of them are selected when you select Capsule Particles, STAR-CCM+ de-selects the incompatible models and posts the message `Removing incompatable model(s) in` *[Phase Interaction]*.

- Artificial Viscosity
- Hertz Mindlin
- Linear Cohesion
- Linear Spring, with either *Normal Spring Stiffness Method* or *Tangential Spring Stiffness Method* set to **Particle Material Based**.
- Parallel Bond
- Rolling Resistance, using the **Displacement Damping** or **Constant Torque** method
- Walton Braun

The Polyhedral Particles Model

The Polyhedral Particles model allows you to use particles of any convex polyhedral shape rather than particles composed of combinations of spheres.

Provided By	**[phase]** > **Models** > *Particle Shape*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Lagrangian Multiphase** > **Lagrangian Phases** > **[phase]** > **Models** > **Polyhedral Particles**	
Requires	In **[physics continuum]** > **Models**: *Optional Models*: **Discrete Element Method (DEM)** In **[phase]** > **Models**: *Particle Type*: **DEM Particles**	
Properties	*Moment of Inertia Scaling* and *Part*. See Polyhedral Particle Model Properties.	
Specific Right-Click Actions	See Polyhedral Particle Model Right-Click Actions.	
Activates	Injector Inputs	**Physics Conditions** > **Orientation Specification** **Physics Values** > **Particle Diameter**, **Particle Orientation** and **Polyhedron Elongation Factor** See Polyhedral Particles Injector Inputs.
	Field Functions	**Particle Diameter**, **Particle Projected Area**, **Particle Projected Length**, **Polyhedron Elongation Factor** See Polyhedral Particle Model Field Functions.

The computational cost of this model is proportional to the number of non-coplanar faces representing the surface of the polyhedron. Before running a simulation, right-click the Polyhedral Particle model node and

select **Show Part** to display the surface representation of the template polyhedron and check the number of faces.

Polyhedral Particle Model Properties

Principal Moment of Inertia

Read-only property reporting the principal moment of inertia of the particle, which is composed of user-defined shapes. This property is used when calculating the momentum balance (see Eqn. (2971)).

Area

Read-only property reporting the area of the particle.

Volume

Read-only property reporting the volume of the particle.

Sphericity

Read-only property reporting the sphericity of the particle. This property is used when calculating the Haider and Levenspiel drag coefficient (see Eqn. (2987)).

Moment of Inertia Scaling

Provides a scaling factor for each axis of the moment of inertia of the particle. This can be used to simulate effects such as uneven distribution of mass within the particle, or the dynamical response of a non-spherical particle. The scaling factor has an allowed range on each axis of 1E-2 to 1E5. The default value is `[1.0, 1.0, 1.0]`.

Part

The part selected to serve as a template for the polyhedral particles, a convex geometry part created from a 3D-CAD model, or a simple shape part, or an imported mesh part.

Polyhedral Particle Model Right-Click Actions

Show Part

Opens a geometry scene showing the template polyhedral part.

Update Properties

Calculates the area, volume, moment of inertia, and sphericity of the template polyhedral part.

Polyhedral Particle Model Field Functions

Particle Diameter

The particle diameter as that of a sphere of equal volume. See Particle Diameter.

Particle Projected Area

The cross-sectional area of the particle orthogonal to the direction of fluid flow.

Particle Projected Length

The average length of the particle parallel to the direction of fluid flow.

Polyhedron Elongation Factor

An ordered triple of multipliers defining stretch factors (>1) or shrink factors (<1) for the template polyhedron. See Polyhedron Elongation Factor.

Polyhedral Particles Injector Inputs

The Polyhedral Particles model activates the following injector conditions:

Orientation Specification

Sets the initial orientation of the particle as it is injected.

Particle Diameter

Specifies the particle diameter as that of a sphere of equal volume.

The Polyhedral Particles model activates the following injector values:

Particle Orientation

Sets the initial orientation of the particle as it is injected.

Polyhedron Elongation Factor

An ordered triple of multipliers defining stretch factors (>1) or shrink factors (<1) for the template polyhedron along each axis of the moment of inertia of the particle. As such, the injected Lagrangian phase can have different shapes and sizes of polyhedral particles of the same topology, based on the template described in the *Part* property of the Polyhedral Particle model.

The Polyhedral Particles model is incompatible with the Random Injector using the **Maximum Packing** option.

Compatibility with Other Models

The following models are incompatible with the Polyhedral Particles model. If any of them are selected when you select Polyhedral Particles, STAR-CCM+ de-selects the incompatible models and posts the message `Removing incompatable model(s) in` *[Phase Interaction]*.

- Artificial Viscosity
- Hertz Mindlin
- Linear Cohesion
- Linear Spring, with either *Normal Spring Stiffness Method* or *Tangential Spring Stiffness Method* set to **Particle Material Based**.
- Parallel Bond
- Rolling Resistance, using the **Displacement Damping** or **Constant Torque** method
- Walton Braun

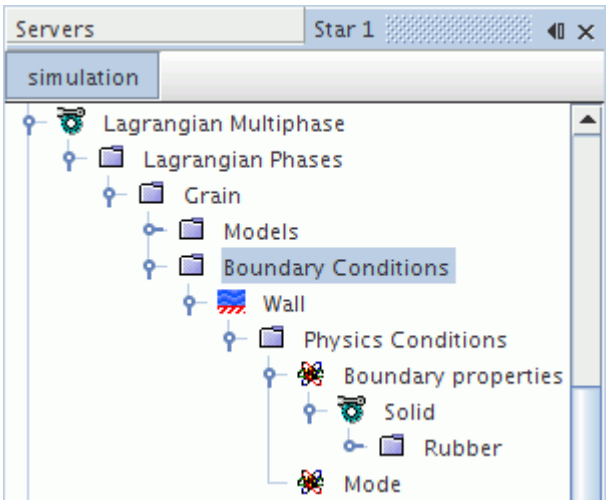
Setting Phase Boundary Conditions

The Lagrangian phase boundary conditions specify how the particles interact with wall boundaries.

As the Discrete Element Method (DEM) is implemented within the Lagrangian framework, you can find further reading on the interaction of particles with boundaries in the Understanding Boundary Interaction Modes section.

For contacts between particle and boundary, the model ignores the tangential component of the contact force when the viscous regime is set to Inviscid.

The Lagrangian phase boundary conditions are represented in the STAR-CCM+ Lagrangian phase by the **Lagrangian Phase** > **Boundary Conditions** node.



Mode

The mode defines how the particle behaves when it interacts with the boundary. Depending on the boundary type, the following boundary modes are available when using DEM:

- Escape—Particles can leave the region through the boundary.
- Transmit—Particles can cross an interface, from one region to another.
- DEM mode—Particles contact and rebound off the boundary. Selecting this mode activates the **Boundary Properties** node, which defines the boundary surface material properties.
- DEM trap mode—Particles leave the region through the boundary at a rate you specify.

Boundary Properties

The boundary properties specify the surface material properties at the boundary and are only available with the DEM mode and DEM trap mode. These properties influence how the particles behave when they contact the boundary. For example, a wood boundary surface would absorb more of the particle contact than an aluminum surface.

You can specify the material properties by either setting the material properties manually, or using a materials database.

Flow Rate Specification

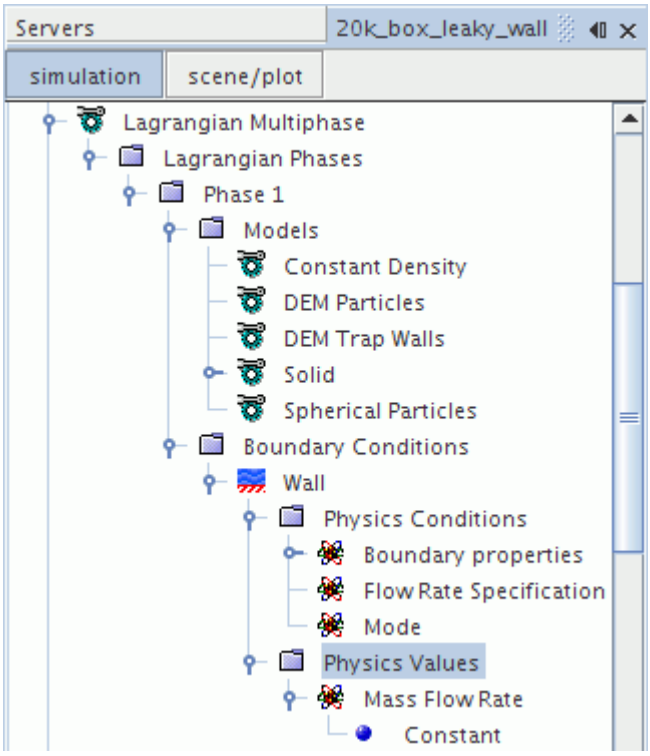
The flow rate specification is available with the DEM trap mode only. This condition specifies how the flow-rate of particles at the boundary is defined.

Flow Rate Specification Properties

Method	Selects the method for entering the flow rate.	
	Mass Flow Rate	Indicates that the rate at which particles leave the boundary as a mass flow rate. Hence this condition is not available with the Massless Particles model. A Mass Flow Rate node is added to the **Values** node.
	Volume Flow Rate	Indicates that the rate at which particles leave the boundary as a volume flow rate. Hence this condition is not available with the Massless Particles model. A Volume Flow Rate node is added to the **Values** node.
	Particle Flow Rate	Indicates that the rate at which particles leave the boundary is specified directly. A Particle Flow Rate node is added to the **Values** node.

Physics Values

You can specify the rate at which DEM particles leave a boundary in the Physics Values node. The Mass Flow Rate, Volume Flow Rate, or Particle Flow Rate are specified as scalar profiles.



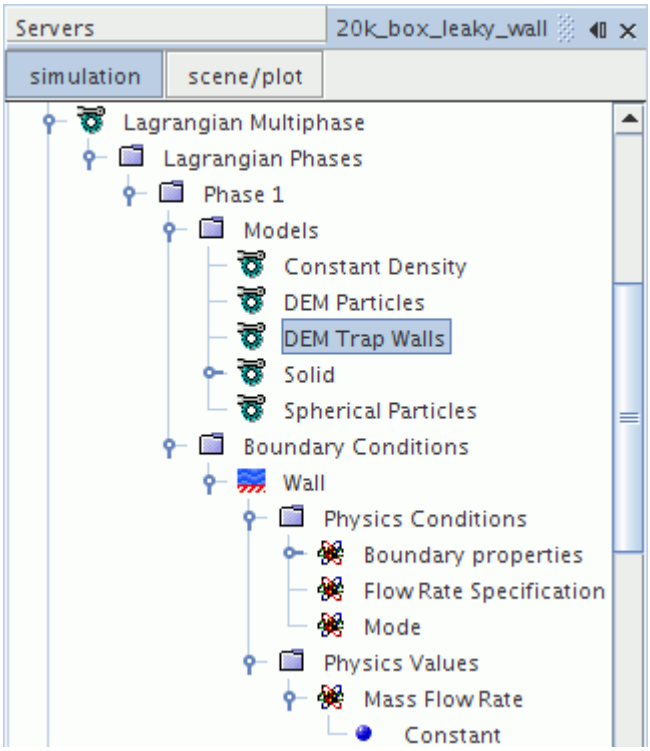
Contents:

Modeling Particle Discharge on Wall Boundaries
Modeling Specific Material Properties for Specific Walls
Boundary Values Reference

Modeling Particle Discharge on Wall Boundaries

A fluidized bed is a long geometry with a small orifice through which particles are discharged at one end. To save simulation time and avoid meshing the small features of the orifice, use the DEM Trap Walls model to approximate a discharge boundary condition through the wall of such geometries.

The DEM Trap Walls model allows you to specify the rate at which DEM particles leave a boundary. The **DEM Trap Walls** node represents this model in the simulation tree.



To specify a discharge boundary condition:

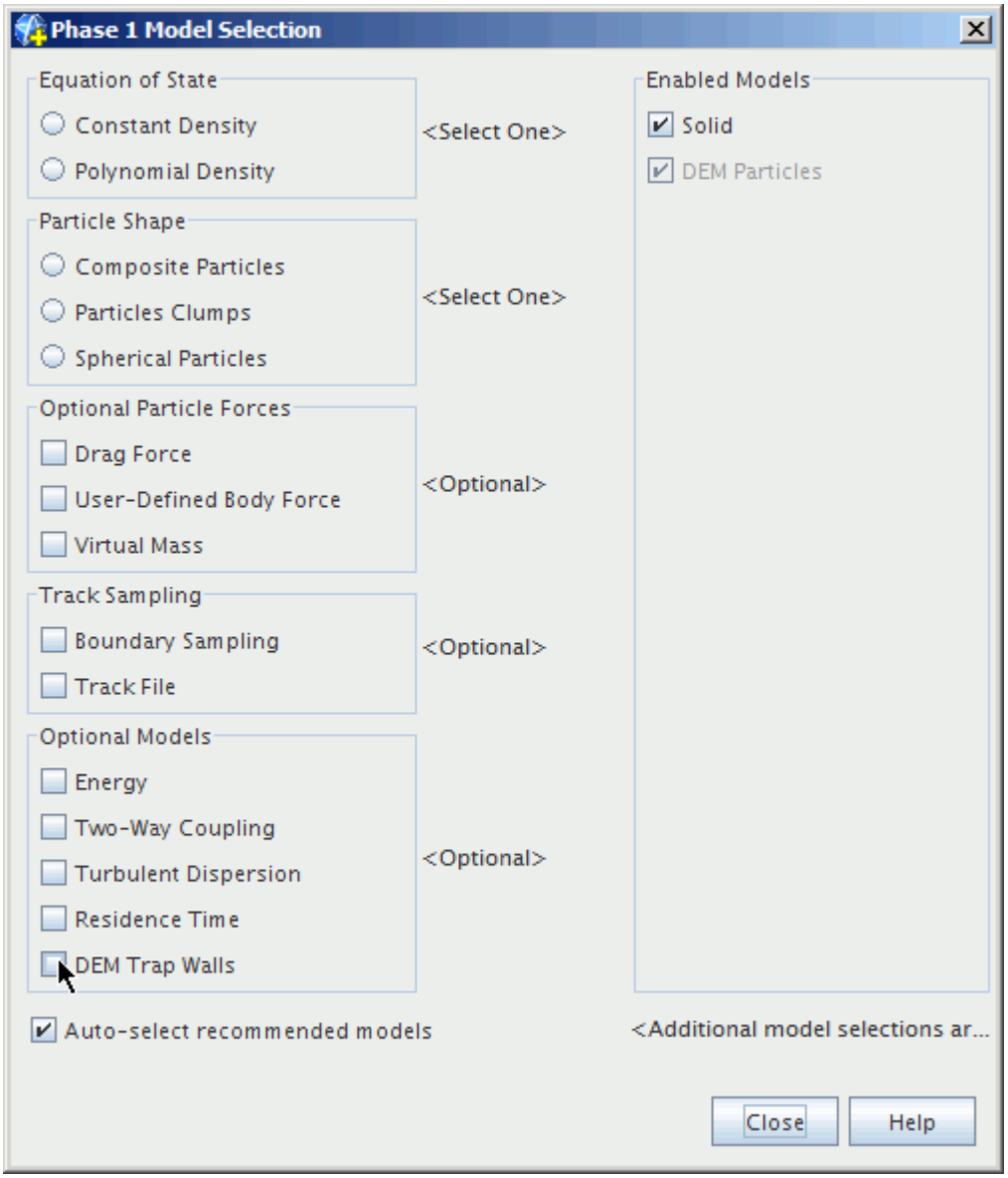
1. Select the DEM Trap Walls model.
2. Set the Mode of the boundary to DEM trap mode.

   The **Mode** node is contained within the **Phase Conditions** node of wall boundaries when DEM is active in the region. Setting the **Mode** to **DEM trap mode** adds a **Flow Rate Specification** node to the **Physics Conditions** for the boundary, and a corresponding value flow rate node depending on the selection for **Flow Rate Specification**.
3. Set the Flow Rate Specification for the boundary.
4. Set the value of the flow rate.

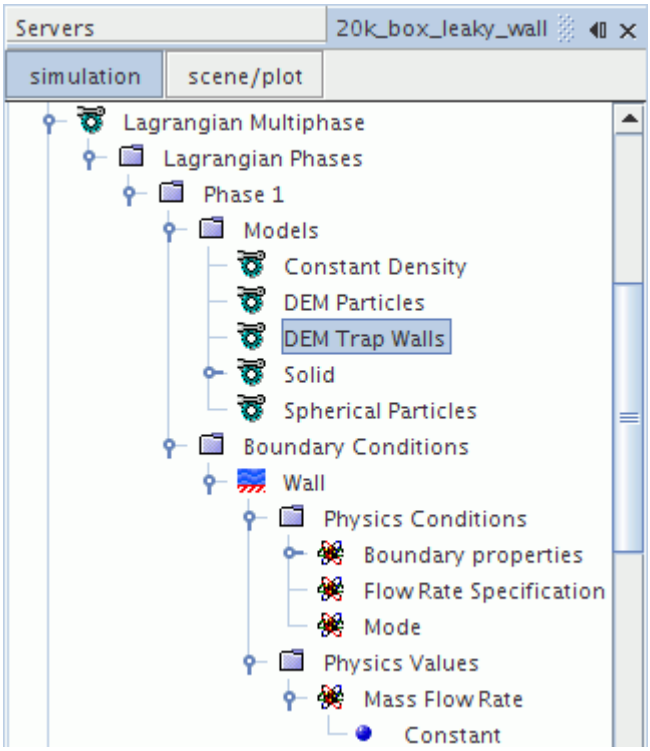
Unlike a fluid, particles do not form a continuum. Therefore, the outflow boundary conditions cannot be guaranteed. The DEM Trap Walls model only ensures that the specified maximum outflow is not exceeded. If the flow rate is set to an unrealistically high value, this model does not force particles to exit the domain.

Selecting the DEM Trap Walls Model

As a pre-requisite, select the **DEM Particles** model from the *Particle Type* group-box in the *Phase Model Selection* dialog. Then select **DEM Trap Walls** from the *Optional Models* group box.



The **DEM Traps Walls** node appears in the simulation tree.



Modeling Specific Material Properties for Specific Walls

You can specify different materials for different wall boundaries of a region while using DEM.

The procedure is as follows:

1. When you create the region, use the option **Create a Boundary for Each Part Surface**.
2. For each boundary where you want to customize the material properties:
   a) In the Regions node, select one of the relevant boundaries, go to **Physics Conditions** > **Lagrangian Specification**, and change the **Method** property to **Specify for Boundary**.
   b) Open up the newly created **Phase Conditions** node, right-click the **Solid** > **Al** node, and select **Replace With...**
   c) Under **Physics Models** > **Multiphase Interaction** > **Phase Interactions**, define a new Phase Interaction pair between the DEM particle phase and the newly customized boundary.

      See Modeling Phase Interactions in the Discrete Element Method.

   Repeat these steps for each customized wall boundary.

Boundary Values Reference

This section describes the boundary values that you can set for a DEM phase.

Abrasive Wear

Added to the tree when the Abrasive Wear model is activated in the Lagrangian phase. Depending on the method that is chosen for the erosion ratio, a sub-node is added to this node.

Abrasive Wear Properties

Method	Selects the method for calculating the Abrasive Wear erosion ratio.	
	Archard	Specifies that the erosion ratio is calculated using the Archard Correlation. An Archard node is added as a child to this node.
	Field Function	Specifies that the erosion ratio is calculated using a field function. This method is typically user-defined and can involve only constants and contact-based functions. A field function method node is added as a child to this node.

Archard

This sub-node is added to the Abrasive Wear boundary value node when **Archard** is chosen as the erosion ratio method. It represents the Abrasive Wear erosion ratio. The default value of 1E-2 is the high end of the recommended range of values, 1E-8 to 1E-2, mild wear to severe wear.

Archard Properties

Abrasive Wear Coefficient	The proportionality constant a in the Archard equation Eqn. (3256). The default value is 0.01 kg/J.

Impact Wear

Added to the tree when the Impact Wear model is activated in the Lagrangian phase. Depending on the method that is chosen for the erosion ratio, a sub-node is added to this node.

Impact Wear Properties

Method	Selects the method for calculating the Impact Wear erosion ratio.	
	Ahlert	Specifies that the erosion ratio is calculated using the Ahlert Correlation. An Ahlert node is added as a child to this node.
	Field Function	Specifies that the erosion ratio is calculated using a field function. This method is typically user-defined and can involve only constants and contact-based functions. A field function method

		node is added as a child to this node.
	DNV	Specifies that the erosion ratio is calculated using the DNV Correlation. A DNV node is added as a child to this node.
	Neilson-Gilchrist	Specifies that the erosion ratio is calculated using the Neilson-Gilchrist Correlation. A Neilson-Gilchrist node is added as a child to this node.
	Oka	Specifies that the erosion ratio is calculated using the Oka Correlation. An Oka node is added as a child to this node.
Dimensions	The dimensionality of the erosion ratio (read-only).	

Modeling Drag Force in the Discrete Element Method

The drag force model calculates the force acting on a particle due to viscosity-induced fluid resistance.

Theory	See Drag Force.	
Provided By	**[phase]** > **Models** > *Optional Particle Forces*	
Example Node Path	**Lagrangian Multiphase** > **Lagrangian Phases** > **[phase]** > **Models** > **Drag Force**	
Requires	In the *Phase Model Selection* dialog, *Particle Type*: **DEM Particles**	
Activates	Model Controls (child nodes)	**Drag Coefficient**: see Drag Coefficient Properties.
	Field Functions	**Particle Drag Force**. See Lagrangian Multiphase Field Function Reference.

Drag Coefficient Properties

STAR-CCM+ calculates the drag force using drag coefficients. It is impractical to determine these drag coefficients spatially, so STAR-CCM+ uses experimental and theoretical correlations to obtain suitable values.

Dimensions

The dimensionality of the drag coefficient (read-only).

Method

Selects the method for specifying the drag coefficient.

Method	**Corresponding Method Node**
Clift Defines the drag coefficient using the Clift correlation. Use this method for cylindrical particles. This option is only available with the Cylindrical Particles model.	**Clift**
Di Felice Defines the drag coefficient using the Di Felice correlation. Use this method for dense particulate flow. This option is only available with the Two-Way Coupling Lagrangian phase model.	**Di Felice**
Field Function Defines the drag coefficient using a field function, which is typically user-defined. All other drag coefficient methods are also available as field functions. As a result, you can combine the STAR-CCM+ built-in drag coefficients with your own calibration factors in a user-defined field function. For example: `0.5*$ParticleGidaspowDragCoefficient`	**Field Function**
Gidaspow This method can be used to simulate flow through fluidized beds where the initial packing of particles transitions to low-volume fractions. Use for diluted flows, where other drag models can underpredict the drag in these flow regimes. The Gidaspow drag coefficient method [106] is a combination of the Wen Yu and Ergun methods. A cutoff void fraction determines the point at which one method switches to the other. Eqn. (2984) (Wen Yu) and Eqn. (2985) (Ergun) in the Formulation section are the relevant method equations.	**Gidaspow** **Gidaspow Drag Coefficient Method Properties** *Exponent* The Wen-Yu power exponent (Eqn. (2985)). *Cutoff Void Fraction* The void fraction cutoff, v_{min} in Eqn. (2984), for the transition between the Wen-Yu model and the Ergun equation.
Haider & Levenspiel Defines the drag coefficient using the Haider & Levenspielt correlation. Use this method for non-spherical particles with $\mathrm{Re} < 2.6 \times 10^5$.	**Haider & Levenspiel**

Schiller-Naumann

Defines the drag coefficient using the Schiller-Naumann correlation, which is one of the standard Lagrangian drag coefficient methods.

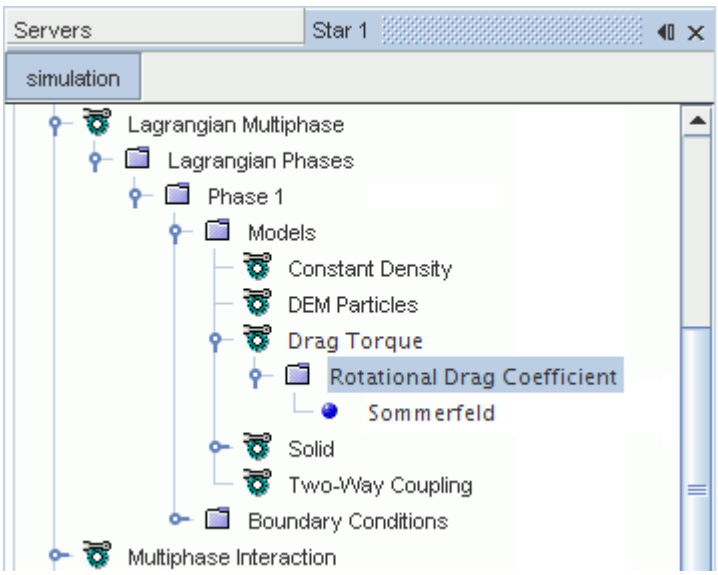
Schiller-Naumann

Modeling Drag Torque in the Discrete Element Method

The Drag Torque model calculates the rotational drag exerted on a particle as it spins in a viscous fluid.

The Drag Torque model calculates the rotational drag according to Eqn. (2990). This model is available in the *Optional Particle Forces* group box for a phase defined as **DEM Particles** in the **Lagrangian Multiphase** model.

The Drag Torque model is represented in the Lagrangian phase by the **Drag Torque** node. Use the **Rotational Drag Coefficient** node to set the method that STAR-CCM+ uses to predict drag torque



Note: The Drag Torque model applies only to spherical particles.

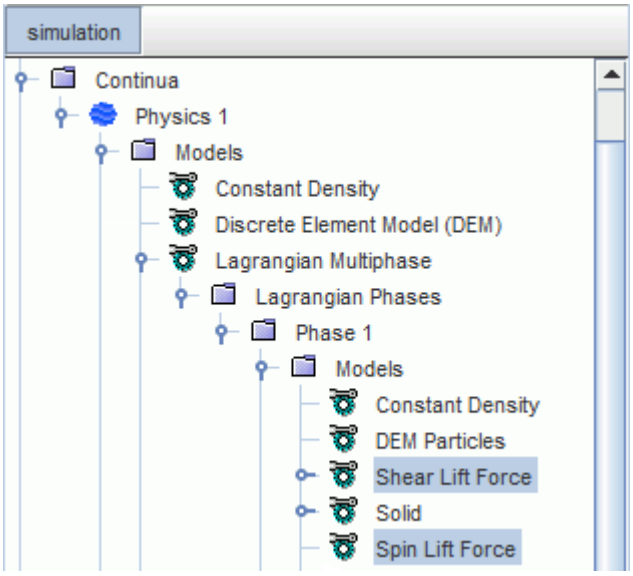
Rotational Drag Coefficient Properties

Dimensions	The dimensionality of the drag coefficient (read-only).	
Method	Selects the method for specifying the drag coefficient.	
	Sommerfeld	Use the definition supplied in Eqn. (2992)
	Field Function	Define your own field function, to specify the rotational drag coefficient.

Modeling Lift Force in the Discrete Element Method

STAR-CCM+ DEM provides two lift force models, one for spin lift force, and another for shear lift force.

In this context, lift forces are forces normal to the particle velocity; they are not necessarily forces in the upward direction.



Contents:

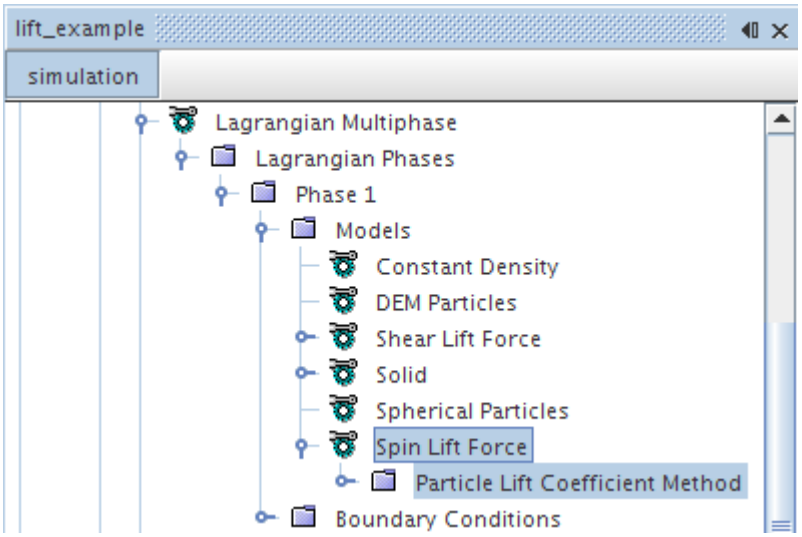
Modeling the Spin Lift Force
Modeling the Shear Lift Force

Modeling the Spin Lift Force

The Spin Lift Force model calculates the force acting on a spinning particle moving in a fluid.

STAR-CCM+ calculates the spin lift force using a particle lift coefficient. You can define the lift coefficient using a field function or the Sommerfeld method. See Eqn. (2995).

The Spin Lift Force model and the Particle Lift Coefficient method are represented in the Lagrangian phase by the **Spin Lift Force** node and the **Particle Lift Coefficient Method** node respectively. No properties are provided on the **Spin Lift Force** node; select the **Particle Lift Coefficient Method** node to access the properties of the appropriate method.

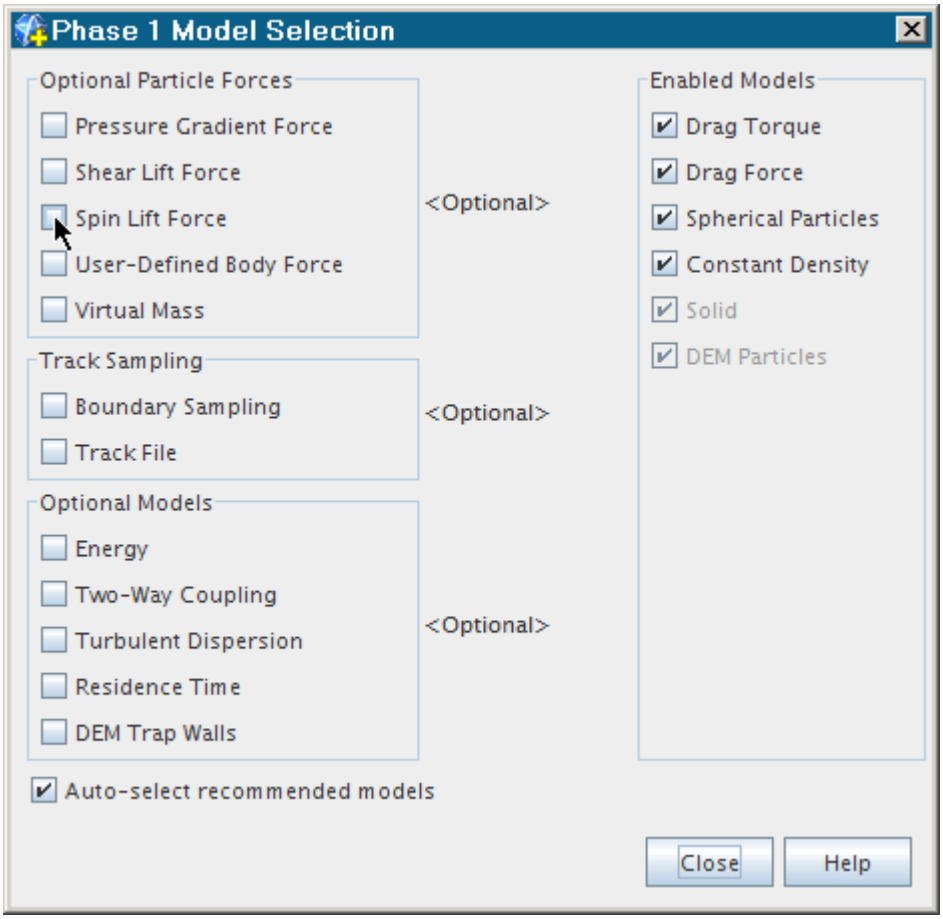


Spin Lift Force Coefficient Properties

Method	Selects the method for specifying the spin lift coefficient.	
	Field function	Define your own field function to specify the spin lift force coefficient.
	Sommerfeld	Calculates the spin lift force using the model derived by Sommerfeld [119], given as Eqn. (2995) in the formulation.

Selecting the Spin Lift Force Model

This model is available in the *Optional Particle Forces* group box for a phase defined as **DEM Particles** in the **Lagrangian Multiphase** model.



Modeling the Shear Lift Force

The Shear Lift Force model calculates the force acting on a particle moving relative to a fluid where there is a velocity gradient in the fluid orthogonal to the relative motion.

STAR-CCM+ calculates the shear lift force using a shear lift coefficient. You can define the lift coefficient using a field function, the Saffman method, or the Sommerfeld method. See Eqn. (2998) and Eqn. (2999).

The Shear Lift Force model and the Shear Lift Coefficient method are represented in the Lagrangian phase by the **Shear Lift Force** node and the **Shear Lift Coefficient Method** node respectively. No properties are provided on the **Shear Lift Force** node; select the **Shear Lift Coefficient Method** node to access the properties of the appropriate method.

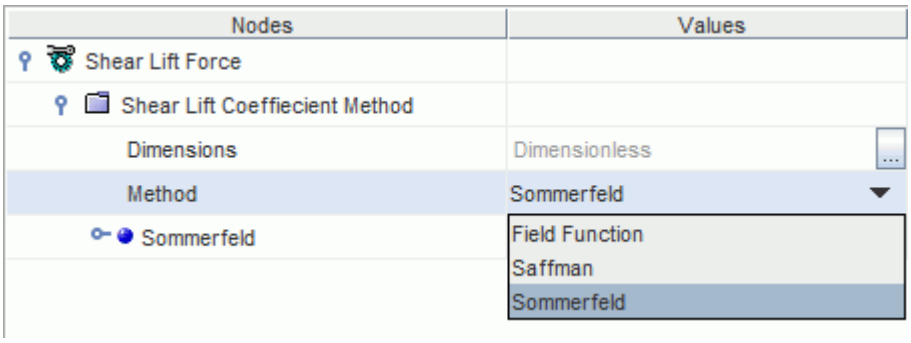
The Shear Lift Force model is available within the Lagrangian Multiphase model as well as within the Discrete Element Model.

Shear Lift Force Coefficient Properties

Method	Selects the method for specifying the shear lift coefficient.	
	Field function	Define your own field function to specify the shear lift force coefficient.

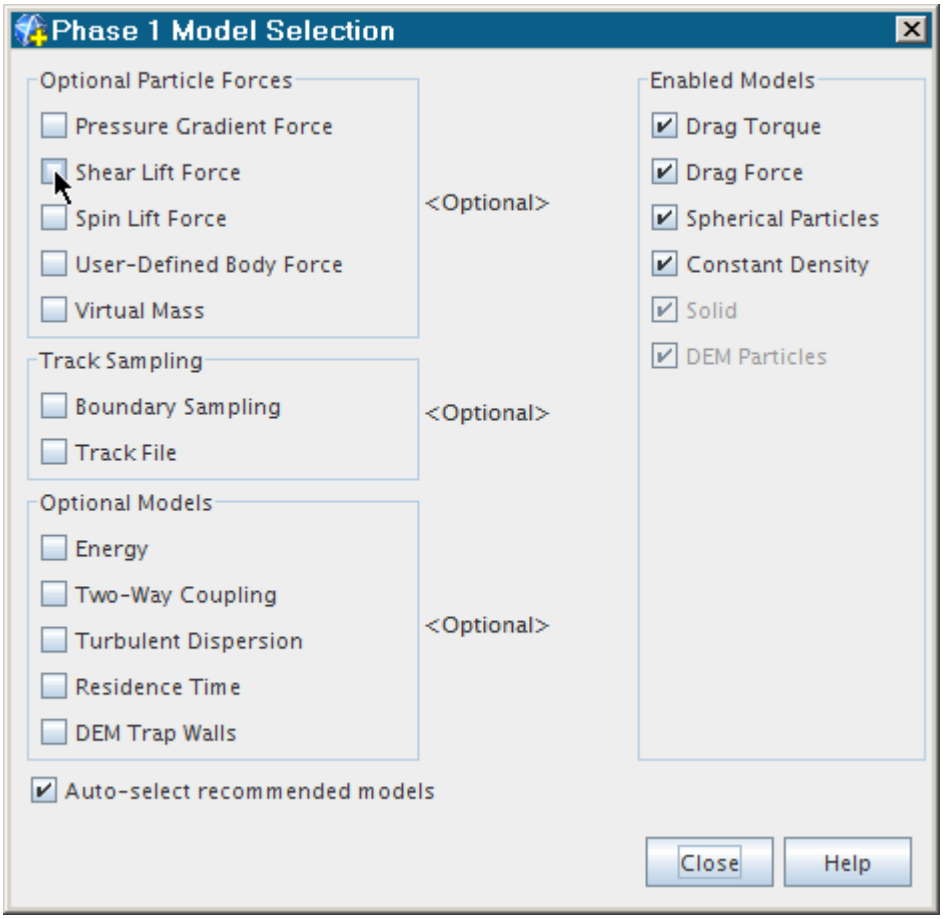
	Saffman	Use the definition of Saffman [118] provided as Eqn. (2998). This method recovers the original Saffman asymptotic solution for low Reynolds numbers.
	Sommerfeld	Use the definition of Sommerfeld [119] provided as Eqn. (2999). This definition covers a broader range of Reynolds numbers than the Saffman definition.

See Particle Shear Lift Force.



Selecting the Shear Lift Force Model

This model is available in the *Optional Particle Forces* group box for a phase defined as **DEM Particles** in the **Lagrangian Multiphase** model.



Modeling Erosion in the Discrete Element Method

Erosion models are provided in both the Lagrangian and DEM frameworks. For DEM, STAR-CCM+ can model abrasive erosion in addition to impact erosion.

What Are the DEM Erosion Models?

The Abrasive Wear model predicts the rate of erosion due to scouring, where particles scrape tangentially along walls or strike at low angles. The Impact Wear model predicts the rate of erosion due to direct impact of particles on the eroded surface. For impact erosion, STAR-CCM+ exposes the same correlations in the DEM framework as are available within the Lagrangian framework.

For each model, the erosion rate is defined as the mass of wall material removed per unit area per unit time, and is provided as a field function when the model is activated. Total erosion rate is the sum of the erosion rates for each model. (See Discrete Element Method Field Functions Reference.)

To calculate the erosion rate, it is necessary to accumulate the damage that each particle impact causes. This calculation is done by selecting a correlation for the erosion ratio, that is, the mass of wall material eroded per unit mass of impinging particles. (See Boundary Values Reference.) Erosion damage depends on both the material of the walls and the material of the particles. Hence, the erosion ratio correlation is a boundary

condition which can optionally be specified differently for each boundary. Further details are given in the Erosion theory.

The same stipulations that apply to erosion modeling with the Lagrangian framework also apply here. See Modeling Erosion (Lagrangian).

Properties

None for the erosion models. For properties of the boundary value methods, see Boundary Values Reference.

Selecting the DEM Erosion Models

To expose the DEM erosion models, activate the DEM Particles model under **Lagrangian Multiphase**. The Impact Wear and Abrasive Wear models then become available in the *Erosion* box. They can be selected together or individually.

To select the erosion ratio correlations:

1. Go to **Models** > **Lagrangian Multiphase** > **Lagrangian Phases**.
2. Open the node for the specific phase, then open **Boundary Conditions** > **Wall** > **Physics Values** node.
3. Select the node for **Abrasive** or **Impact Wear** and set the *Method* property.

See Also:

Erosion Model

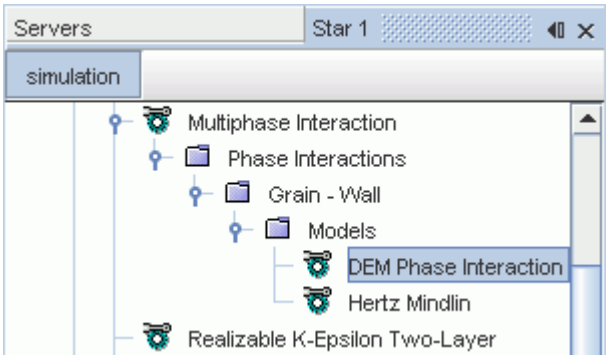
Modeling Phase Interactions in the Discrete Element Method

The DEM Phase Interaction model determines how particles behave when they come into contact with each other or solid boundaries.

The DEM Phase Interaction model provides access to other models that define particle interaction behavior.

- Hertz Mindlin model
- Walton Braun model
- Linear Spring model
- Conduction Heat Transfer model
- Parallel Bonds model
- Linear Cohesion model
- Bonded Particles model
- Simple Failure model
- Constant Rate Damage model
- Rolling Resistance model

The DEM Phase Interaction model is represented in the STAR-CCM+ phase interaction continuum by the **DEM Phase Interaction** node. Select the node to access the properties of the model



DEM Phase Interaction Model Properties

First phase	Specifies the first interacting DEM phase. Only DEM phases that are defined under the Lagrangian Multiphase model are available.
Second phase	Specifies the second interacting DEM phase. Only DEM phases that are defined under the Lagrangian Multiphase model are available.

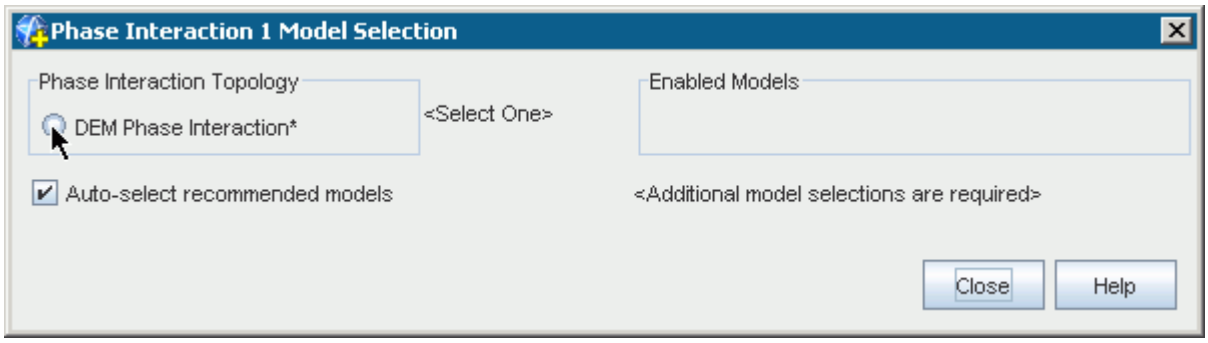
Selecting the DEM Phase Interaction Model

As a prerequisite, select the following physics model:

- In the *Physics Model Selection* dialog, select the Discrete Element Model (DEM).

Then, to select the **DEM Phase Interaction** model:

- In the *Phase Interaction Model Selection* dialog, select **DEM Phase Interaction** from the *Phase Interaction Topology* group box.



Contents:

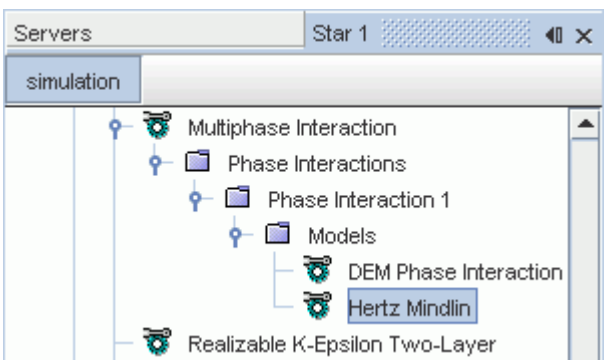
What Is the Hertz Mindlin Model?
What Is the Walton Braun Model?
The Linear Spring Model
User-Defined Contact Force Model
What Is the Conduction Heat Transfer Model?
What Is the Rolling Resistance Model?

What Is the Linear Cohesion Model?
What Is the Parallel Bonds Model?
What Is the Bonded Particles Model?
The Simple Failure Model
The Constant Rate Damage Model
The Impact Heat Model
The Passive Scalar Transfer Model
Visualizing and Reporting Contact

The Hertz Mindlin Model

The Hertz Mindlin no-slip contact model is the standard model that is used for modeling contact between particles.

The Hertz Mindlin model is represented in the STAR-CCM+ phase interaction continuum by the **Hertz Mindlin** node. Select the node to access the properties of the model.



Hertz Mindlin Properties

Static Friction Coefficient	C_{fs} (see Eqn. (3195)).
Rolling Friction Coefficient	$c_{\text{rolling friction}}$ (see Eqn. (634)).
Normal Restitution Coefficient	See Eqn. (645).
Tangential Restitution Coefficient	See Eqn. (646).

Selecting the Hertz Mindlin Model

To select the **Hertz Mindlin** model:

- In the *Phase Interaction Model Selection* dialog, select **Hertz Mindlin** from the *DEM Contact Model* group box.

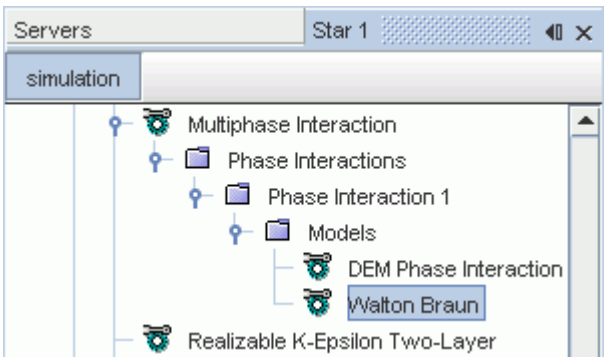
The Walton Braun Model

The Walton Braun model is an elastic-plastic contact model, which is useful in situations where a collision between two bodies includes plastic deformation.

This situation is typical for materials like grain, soybeans, or coal where the collisions lead to material deformation and the impact energy is dissipated during the collision.

In order to simulate inelastic material, STAR-CCM+ provides a hysteretic plastic-elastic contact model that is based on the formulation of Walton and Braun ([124], [125]).

The Walton Braun model is represented in the STAR-CCM+ phase interaction continuum by the **Walton Braun** node. Select the node to access the properties and expert properties of the model.



Walton Braun Properties

These property settings are available for you to calibrate when comparing results in STAR-CCM+ to your experimental results.

Static Friction Coefficient	C_{fs} (see Eqn. (3195)).
Rolling Friction Coefficient	μ_r (see Eqn. (3215)).
Normal Restitution Coefficient	$C_{n\ rest}$ See Eqn. (3198).
Tangential Restitution Coefficient	$C_{t\ rest}$ See Eqn. (3199).
Stiffness Ratio	Controls the ratio of release and accumulation of energies during loading and/or unloading, therefore this property must be a non-negative, non-zero value. Setting this property to 0 means the material is perfectly plastic and it only deforms. The Walton Braun model is not suitable for simulating such physics.
Yield Strength Fraction	Also known as yield stress fraction (see Eqn. (3212)).

Walton Braun Expert Properties

Freeze Tangential Force	Controls whether to include tangential force in the calculations.

	Activated	Does not include tangential force.
	Deactivated	Includes tangential force.
Freeze Torque	Controls whether to include torque in the calculations.	
	Activated	Does not include torque.
	Deactivated	Includes torque.

Selecting the Walton Braun Model

To select the **Walton Braun** model:

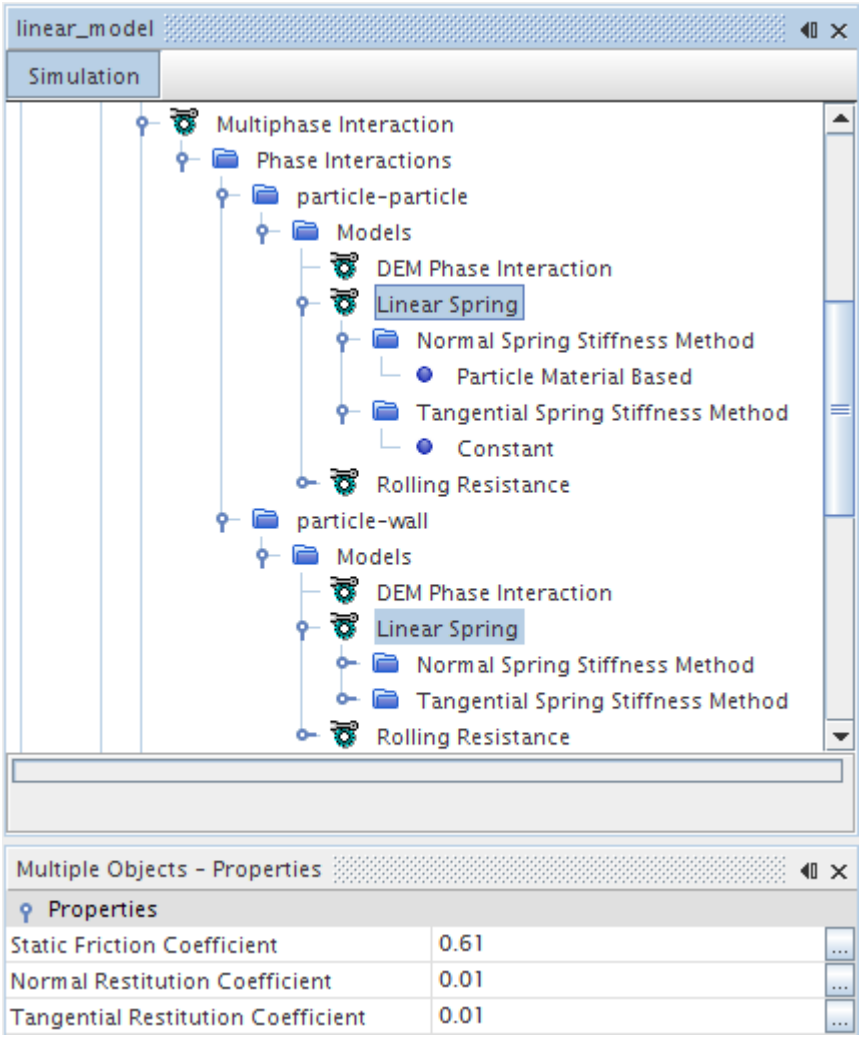
- In the *Phase Interaction Model Selection* dialog, select **Walton Braun** from the *DEM Contact Model* group box.

The Linear Spring Model

The Linear Spring model calculates contact between particles or between particles and a wall. The relationship between particle compression and contact forces is based on the relationship found in linear springs.

The model uses constants for normal and tangential spring stiffness that are either set by you directly or are based on properties of the particle: the radius, maximum overlap, and Young's modulus (see Contact Force Modeling).

This model is represented under the phase interactions node by the Linear Spring node. Select the node to set the properties.



Linear Spring Properties

Static Friction Coefficient	The friction coefficient C_{fs}. See Eqn. (3195).	
Normal Restitution Coefficient	$C_{n\ \mathrm{rest}}$. See Eqn. (3198).	
Tangential Restitution Coefficient	$C_{t\ \mathrm{rest}}$. See Eqn. (3199).	
Normal Spring Stiffness Method	Specifies the method for assigning normal spring stiffness K_n to the bodies in the interaction. See Eqn. (3192).	
	Constant	Stiffness is a constant set by the user.
	Particle Material Based	Stiffness is calculated based on the material composing the particle.

Tangential Spring Stiffness Method	Specifies the method for tangential normal spring stiffness K_t to the bodies in the interaction. See Eqn. (3195).	
	Constant	Stiffness is a constant set by the user.
	Particle Material Based	Stiffness is calculated based on the material composing the particle.

Particle Material Based Properties

Maximum Overlap/Radius Ratio	A constant set by the user, based on the overlap distance and the particle radius. The default is 0.1.

User-Defined Contact Force Model

The User-Defined Contact Force model allows you to use a contact force you have defined in a field function.

Define a vector field function that returns a vector for the contact force between two particles or particle and a wall. For example, you can define a linear spring model with added cohesion or dissipation. These additions can be made velocity-dependent, or given other conditions.

Provided By	**[physics continuum]** > **Models** > **Multiphase Interaction** > **Phase Interactions** > **[interaction]** > **Models** > *DEM Contact Model*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Multiphase Interaction** > **Phase Interactions** > **Phase Interaction 1** > **Models** > **User-Defined Contact Force**
Requires	*Optional Models*: **Discrete Element Model (DEM)**, **Multiphase Interaction** (selected automatically) From **Multiphase Interaction** > **Phase Interactions** > **[interaction]** > **Models**: *Phase Interaction Model Topology*: **DEM Phase Interaction**
Properties	See below

Properties

Dimensions

Read only. The dimensions for the field function that is selected for the model, usually `Force`.

Function

The field function that is selected for the model. To define a customized field function, write a user function. You can reference any of the following primitive field functions in your own field function:

Contact Centroid
Contact Contact Mass 0 | 1

Contact Contact Radius 0 | 1
Contact Damping Force
Contact Elastic Force
Contact Force
Contact Normal
Contact Normal Overlap
Contact Overlap
Contact Physics Radius 0 | 1
Contact Poisson Ratio 0 | 1
Contact Relative Velocity
Contact Surface Velocity 0 | 1
Contact Tangential Overlap
Contact Torque
Contact Youngs Modulus 0 | 1

See the Discrete Element Method Field Function Reference and Creating a User Field Function.

Example Field Functions

Example 1

```
$$ContactNormalOverlap * -100000
```

The contact force is set equal to `-100000` times the overlap vector. The minus sign is necessary to set the correct force direction.

Example 2

```
$${FCnormal}-$${FCtang}
```

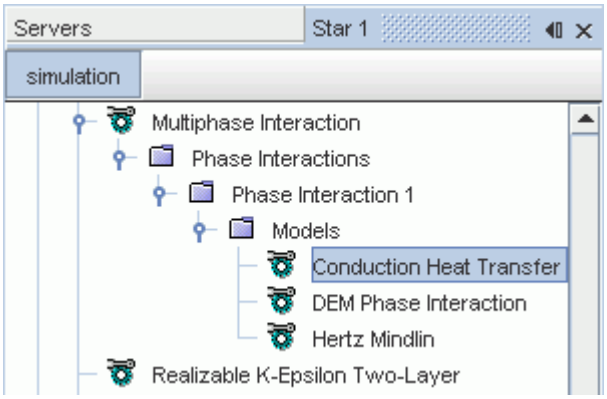
The contact force is the vector difference between `FCnormal` and `FCtang`, which are defined in terms of **Contact Normal Overlap** and **Contact Tangential Overlap**:

```
FCnormal = -1.72E7*$${ContactNormalOverlap}
FCtang = (mag(1.48E7*$${ContactTangentialOverlap}) < mag(1.72*$$
{ContactNormalOverlap})*0.092) ? 1.48*$$
{ContactTangentialOverlap}:mag(1.72E7*$$
{ContactNormalOverlap})*0.092*unit($${ContactTangentialOverlap})
```

The Conduction Heat Transfer Model

The Conduction Heat Transfer model calculates conduction heat transfer between two particles or a particle and a wall.

This model is represented in the STAR-CCM+ phase interaction continuum by the **Conduction Heat Transfer** node. This node has no properties.



The energy state of the continuous phase is not affected by particle-particle or particle-wall heat conduction unless you also activate the Two-Way Coupling Model.

When the Two-Way Coupling model is active in conjunction with Conduction Heat Transfer, the wall heat flux coefficients are modified according to Eqn. (665) and Eqn. (666). See the Conduction Heat Transfer formulation section.

Selecting the Conduction Heat Transfer Model

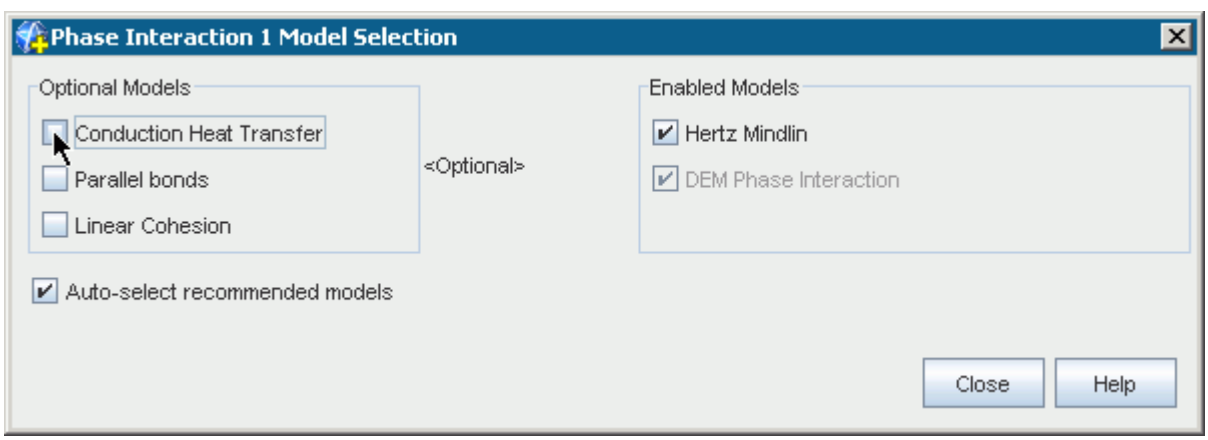
The Conduction Heat Transfer Model is available for phase interactions involving two particles or a particle and a wall. Therefore, set the First phase property of the **DEM Phase Interaction** model to the DEM Lagrangian phase and the Second phase property to either the DEM Lagrangian phase or a wall.

As a prerequisite, select the following phase interaction model:

- In the *Phase Interaction Model Selection* dialog, select either the Hertz Mindlin or Walton Braun models.

Then, to select the **Conduction Heat Transfer** model:

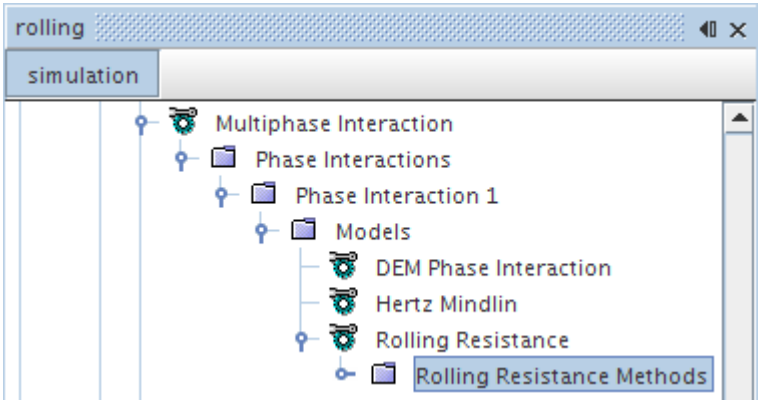
- In the *Phase Interaction Model Selection* dialog, select **Conduction Heat Transfer** from the *Optional Models* group box.



The Rolling Resistance Model

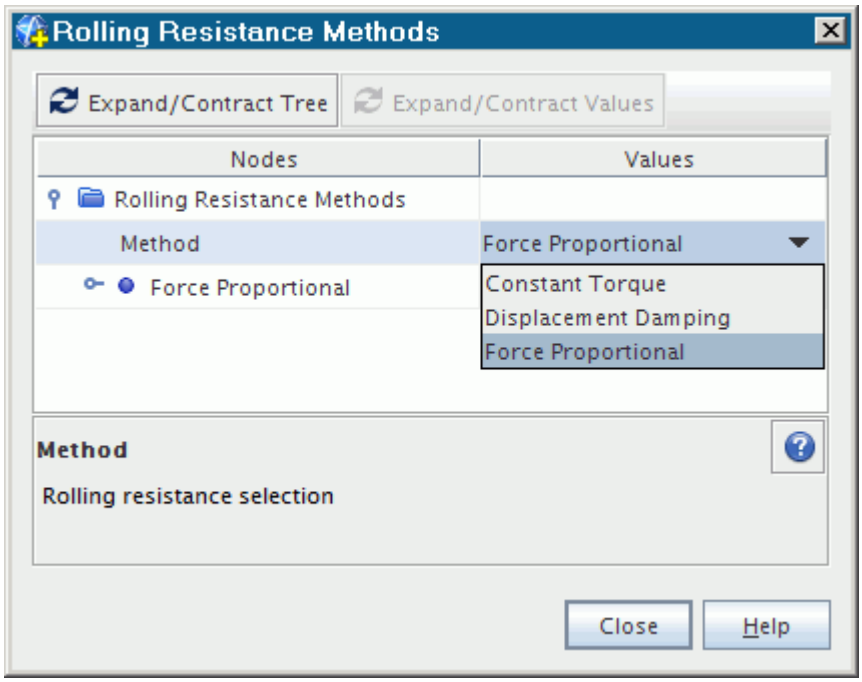
With the DEM model active in a physics continuum, rolling resistance is an optional model in a phase interaction.

The **Rolling Resistance** node is shown below:



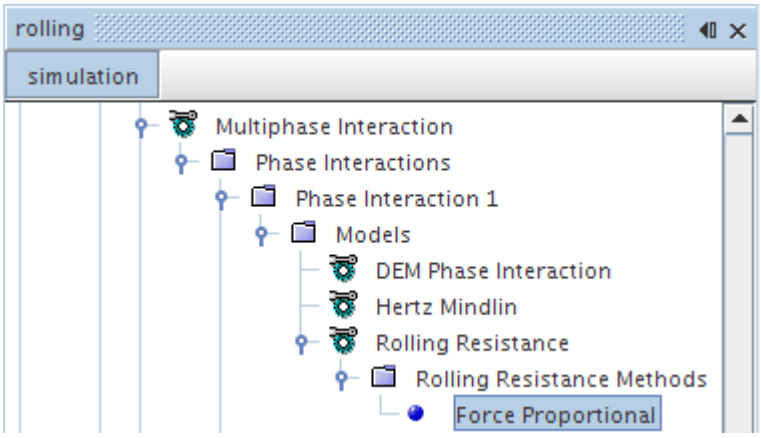
To use the rolling resistance model:

- Expand the **Rolling Resistance** node.
- Right-click on the **Rolling Resistance Methods** node and select **Edit...** to access the dialog for selecting the method.
- Select one of these methods from the *Method* drop-down menu: **Force Proportional**, **Constant Torque**, or **Displacement Damping**.



Force Proportional Properties

This method is the default version of Rolling Resistance. The Force Proportional method is represented in the STAR-CCM+ phase interaction continuum by the **Force Proportional** node.



Coefficient of Rolling Resistance	μ_r (see Force Proportional Method). It can take on any non-negative real value. The default value is 0.0010.

Constant Torque Properties

The Constant Torque method is represented in the STAR-CCM+ phase interaction continuum by the **Constant Torque** node.

Coefficient of Rolling Resistance	μ_r (see Eqn. (3215)). It can take on any non-negative real value. The default value is 0.1.	
Moment Limiter Factor	Sets the limit used by the *Use Limiter* property. It can take on any non-negative real value. It operates only when *Use Limiter* is active. The default value is 0.08.	
Use Limiter	Sets an upper limit on the angular momentum of the particles around the axes of moment. The value of the limit is set in the *Moment Limiter Factor* property.	
	Activated	Activates moment limiting.
	Deactivated	Leaves moment limiting inactive (the default).

Displacement Damping Properties

The Displacement Damping method is represented in the STAR-CCM+ phase interaction continuum by the **Displacement Damping** node.

Coefficient of Rolling Resistance	μ_r (see Eqn. (3215)). It can take on any non-negative real value. The default value is 0.1.

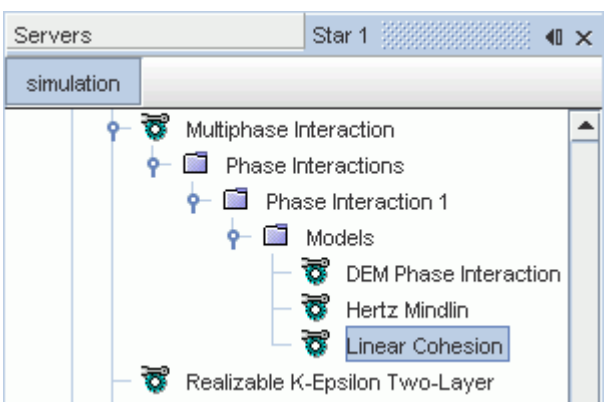
Theta Maximum	θ_r^{max} , the maximum angular displacement before rolling begins. It is available only when *Use Damper* is active. The default value is 0.035 radian.	
Viscous Damping Factor	ν , sets the factor used by the *Use Damper* property. It can take on any value from 0 to 1. It is available only when *Use Damper* is active. The default value is 0.1.	
Viscous Damping f Multiplier	f , provides a multiplier for the *Viscous Damping Factor* property. It can take on any value from 0 to 1. It is available only when *Use Damper* is active. The default value is 1.	
Use Damper	Sets an upper limit on the angular momenta of particles around their axes of moment. The value of the limit is set in the *Viscous Damping Factor* property.	
	Activated	Activates viscous damping.
	Deactivated	Leaves viscous damping inactive (the default).

The Linear Cohesion Model

Cohesion modeling facilitates simulation of inter-molecular attraction forces (Van der Waals forces) between particle surfaces. For some classes of simulations, such as dry powders, these forces significantly affect the outcome and cannot be ignored.

STAR-CCM+ implements a variant of the cohesion model that is known as the Linear Cohesion model, which allows you to choose between two major models by making an entry for F in the properties of the **Linear Cohesion** node (see the formulation) . The cohesion model acts in tandem with the contact model, so when particles attach both contact and cohesion work simultaneously until the particles detach again.

The Linear Cohesion model is represented in the STAR-CCM+ phase interaction continuum by the Linear Particles node. Select the node to access the properties of the model.



Linear Cohesion Properties

Work of Cohesion	W in Eqn. (3214).
Factor	The multiplication model blending factor F , with values varying between JKR and DMT models (see Eqn. (3214)): For the JKR model [110], enter 1.5. For the DMT model [99], enter 2.0.

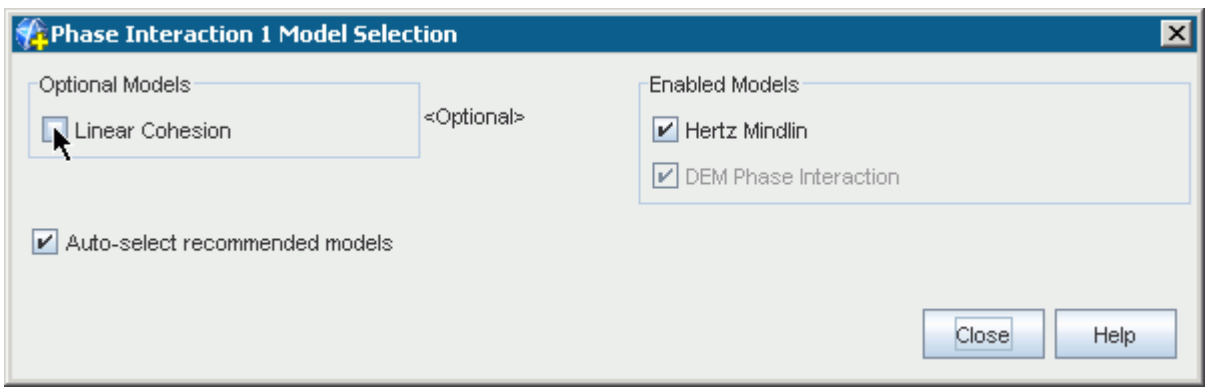
Selecting the Linear Cohesion Model

As a prerequisite, select the following physics and phase interaction models:

- In the *Physics Model Selection* dialog, select the Discrete Element Model (DEM).
- In the *Phase Interaction Model Selection* dialog, select the DEM Phase Interaction and either the Hertz Mindlin or Walton Braun models.

Then, to select the **Linear Cohesion** model:

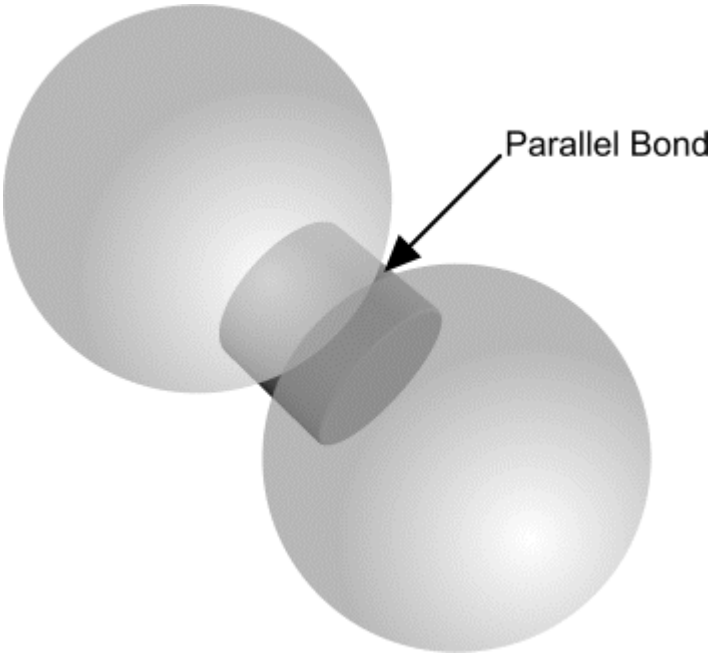
- In the *Phase Interaction Model Selection* dialog, select **Linear Cohesion** from the *Optional Models* group box.



The Parallel Bonds Model

The Parallel Bonds model introduces attractive inter-particle forces to the particle system. During the specified formation time period, the Parallel Bonds model acts to form clusters from particles that collide.

The parallel bonds model uses the concept of a massless bar connecting a pair of bonded particles. The bar can transmit force and torque between particles and it is also subject to cracking under load.

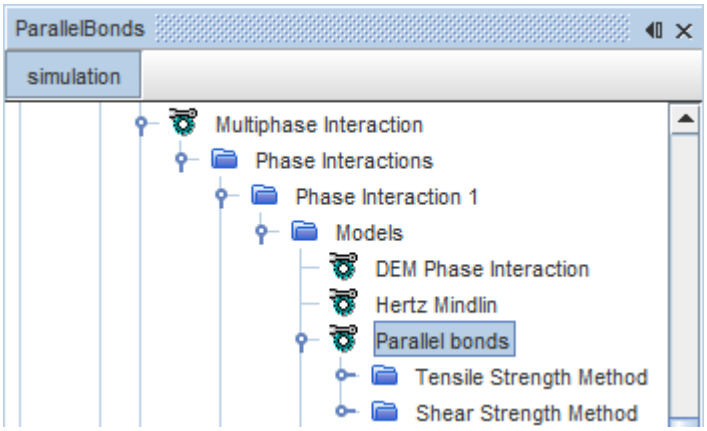


The inter-particle force consists of the traditional grain-grain interaction (represented by the specified contact model, Hertz-Mindlin or Walton-Braun) and the parallel bond forces. When the Parallel Bonds model is selected, the Simple Failure and Constant Rate Damage models become available. If either of these models is selected and determines that the bond breaks, the particles separate. Thereafter, the Hertz Mindlin model determines the particle-particle interaction upon collision.

Potential applications of this particular model include:

- Pharmaceutical and biomedical—the tableting process, the breakage of particles, and deforming cells in blood flow
- Nuclear—checking the effects of certain obstructions to the flow of nuclear waste material; heat transfer studies
- Chemical and metallurgical—particle breakage and dust formation during handling of solid raw materials; also phase change
- Shipping—ice breakup, specifically ice breakup upon collision with dams or shorelines, as well as upon collision with shipping vessels
- Petroleum—hydrates and asphaltenes. This application can include specifying how particles bond with each other based on various parameters.

The Parallel Bonds model is represented in the STAR-CCM+ phase interaction continuum by the **Parallel Bonds** node. Select the node to access the properties and expert properties of the model.



Controlling Bond Formation

Parallel bond formation depends on the expert properties in the model, *Bond Start Time* and *Bond End Time*. When the simulation time is in the interval between these times, any particles that collide establishes a parallel bond in addition to the regular contact force being calculated.

Parallel Bonds Properties

Bond Start Time	Specifies the time at which the bonds begin to be created. The default is 0.
Bond End Time	Specifies the time at which the bond creation process stops. The default is 0.

Parallel Bonds Expert Properties

Bond Radius Multiplier	Scales the radius of the particle bond. The default multiplier is 1.

Selecting the Parallel Bonds Model

The Parallel Bonds Model is only available for a phase interaction involving two particles. It is not available for use on a phase already using the Bonded Particles model, such as Particle Clumps. To select the Parallel Bonds model, set the First phase and Second phase properties of the **DEM Phase Interaction** model to DEM Lagrangian phases.

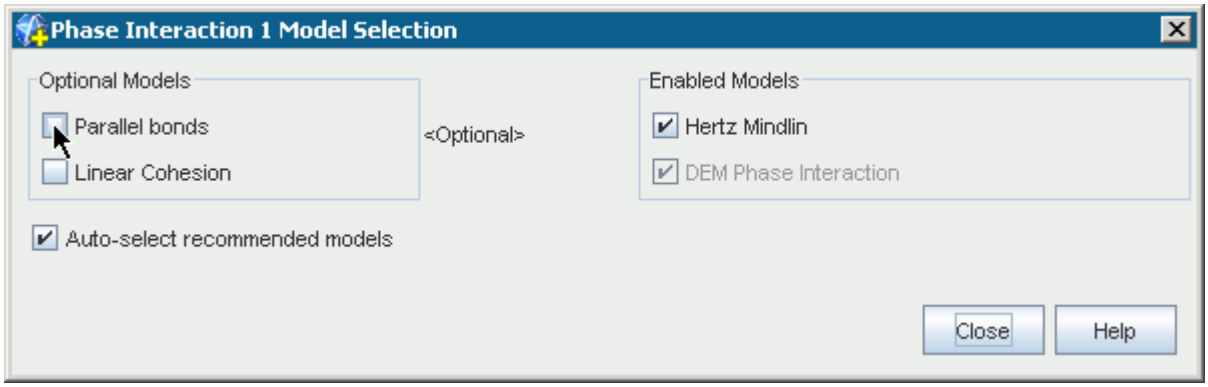
As a prerequisite, select the following physics, phase, and phase interaction models:

- In the *Physics Model Selection* dialog, select the Discrete Element Model (DEM).
- In the *Phase Model Selection* dialog, select the DEM Particles model.
- In the *Phase Interaction Model Selection* dialog, select the DEM Phase Interaction and either the Hertz Mindlin or Walton Braun models.

Then, to select the **Parallel Bonds** model:

- Close the model selection window, select **DEM Phase Interaction** node, and in the properties window set the first and second phase to the DEM phase to be bonded.

- In the *Phase Interaction Model Selection* dialog, select **Parallel Bonds** from the *Optional Models* group box.

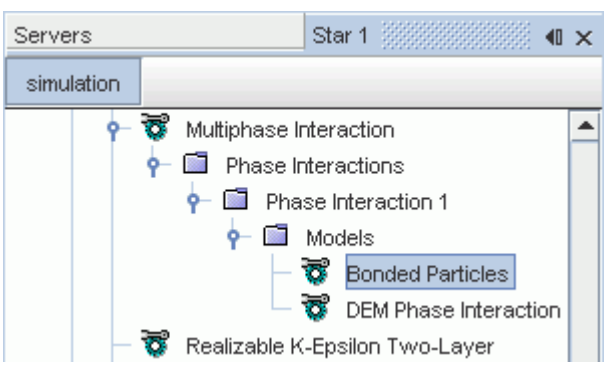


The Bonded Particles Model

The bonded particle model is a combination of the Hertz Mindlin and Parallel Bonds model, and is activated by default with the Particle Clumps model. When the particles are attached, the Parallel Bonds model determines the connection and inter-particle forces.

When the Bonded Particles model is selected, the Simple Failure and Constant Rate Damage models become available. If either of these models is selected and determines that the bond breaks, the particles separate. Thereafter, the Hertz Mindlin model determines the particle-particle interaction upon collision.

The Bonded Particles model is represented in the STAR-CCM+ phase interaction continuum by the **Bonded Particles** node. Select the node to access the properties of the model.



Bonded Particles Properties

Static Friction Coefficient	C_{fs} (see Eqn. (3195)).
Rolling Friction Coefficient	$c_{\text{rolling friction}}$ (see Eqn. (3215)).
Normal Restitution Coefficient	See Eqn. (3198).

Tangential Restitution Coefficient	See Eqn. (3198).

Bonded Particles Expert Properties

Bond Radius Multiplier	Scales the radius of the particle bond.

Selecting the Bonded Particles Model

Selecting the Particle Clumps phase model automatically selects the **Bonded Particles** model. This model is not available by any other method.

The Simple Failure Model

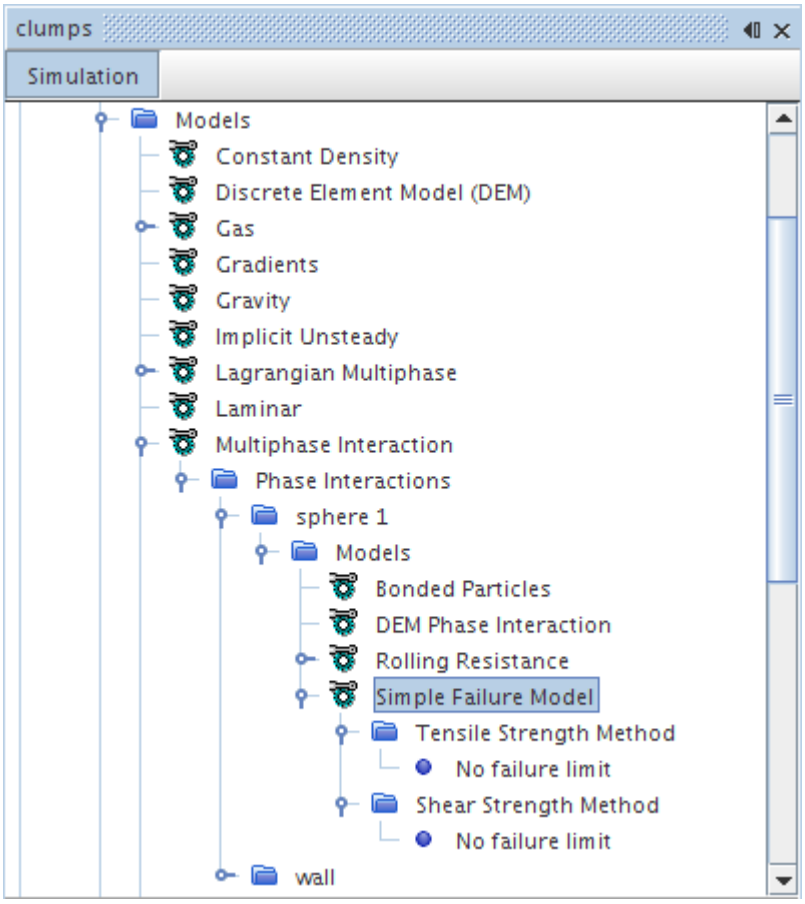
The Simple Failure model emulates bonds that fail instantaneously under load.

This model has applications such as:

- Oil and gas drilling, where the rock surface is represented by particle clumps and repeated passes of the drill cause cumulative damage.
- Pharmaceutical tablet manufacture, where tablets are formed from compressed powder and the strength of internal bonds, predicting breakage resistance, can be predicted from force and time of compression.

If the tensile or shear stresses between particles exceeds the maximum limits, the bond breaks and the particles separate. Thereafter, the other selected Phase Interaction models determine the particle-particle interaction upon collision.

The Simple Failure model is represented in the STAR-CCM+ phase interaction continuum by the Simple Failure Model node. Select the node to access the properties of the model.



Simple Failure Properties

Tensile Strength Method	Specifies the method for assigning tensile strength to the bond. The bond fails if tensile stress exceeds δ_{max} in Eqn. (3231).	
	No Failure Limit	The method does not allow the bond to break from tensile stress.
	Constant	Bond strength is a constant set by the user.
	Uniform Random	The method assigns a strength to each bond, selected from a uniform distribution with upper and lower limits set by the user.
	Normal Random	The method assigns a strength to each bond, selected from a normal distribution with mean, standard deviation, and upper and lower limits set by the user.
Shear Strength Method	Specifies the method for assigning shear strength to the bond. The bond fails if shear stress exceeds σ_{max} in Eqn. (3232).	

	No Failure Limit	The method does not allow the bond to break from shear stress.
	Constant	Bond strength is a constant set by the user.
	Uniform Random	The method assigns a strength to each bond, selected from a uniform distribution with upper and lower limits set by the user.
	Normal Random	The method assigns a strength to each bond, selected from a normal distribution with mean, standard deviation, and upper and lower limits set by the user.

Selecting the Simple Failure Model

This model and the Constant Damage Rate model become available when either the Parallel Bonds or the Bonded Particles model is selected. These, in turn, depend on the selection of the Particle Clumps model. If neither Simple Failure nor Constant Rate Damage is selected, the bonds within the particle clumps are treated as unbreakable.

The Constant Rate Damage Model

The Constant Rate Damage model emulates the breaking of bonds in brittle fracture growth. It can also be used to predict bond strengths.

This model has applications such as:

- Oil and gas drilling, where the rock surface is represented by particle clumps and repeated passes of the drill cause cumulative damage.
- Pharmaceutical tablet manufacture, where tablets are formed from compressed powder and the strength of internal bonds, predicting breakage resistance, can be predicted from force and time of compression.

This model represents damage to bonds on a scale from 0 (undamaged) to 1 (broken). The amount of energy needed to break a bond is proportional to a dimensionless scalar, the softening rate, k_r in Eqn. (3235).

The Constant Rate Damage model is represented in the STAR-CCM+ phase interaction continuum by the **Constant Rate Damage Model** node. Select the node to access the properties of the model.

Constant Rate Damage Model Properties

Softening Rate	The higher the value, the more work (energy) needed to completely break the bonds, or equivalently, the more strain for a given bond strength. The rate is a dimensionless scalar, k_r in Eqn. (3235). Set the value higher than 1 to achieve stability. The default value is 2.

Tensile Strength Method	Specifies the method for assigning tensile strength to the bond. An undamaged bond begins to fail if tensile stress exceeds $\delta_{\max}$ in Eqn. (3231).	
	No Failure Limit	The method does not allow the bond to break. This is the default.
	Constant	Bond strength is a constant set by the user.
	Uniform Random	The method assigns a strength to each bond, selected from a uniform distribution with upper and lower limits set by the user.
	Normal Random	The method assigns a strength to each bond, selected from a normal distribution with mean, standard deviation, and upper and lower limits set by the user.
Initial Damage Method	Specifies the method by which the initial bond damage is calculated.	
	No Initial Damage	The initial bond damage value is zero. This is almost always the best choice. This is the default.
	Uniform	The method assigns a damage to each bond, selected from a uniform distribution between upper and lower limits selected by the user. The default range is [0, 1], and values outside that range are changed to the nearer of 0 or 1.
	Weibull	The method assigns a strength to each bond, selected from a normal distribution with mean (*Lambda*, default value 0.5), standard deviation (*k*, default value 1), and upper and lower limits selected by the user. The default range is [0, 1], and values outside that range are changed to the nearer of 0 or 1.

Selecting the Constant Rate Damage Model

This model and the Simple Failure model become available when either the Parallel Bonds or the Bonded Particles model is selected. These, in turn, depend on the selection of the Particle Clumps model. If neither Constant Rate Damage nor Simple Failure is selected, the bonds within the particle clumps are treated as unbreakable.

The Impact Heat Model

The Impact Heat Model simulates the heating of DEM particles from friction and viscous damping.

This model calculates the rate of heat generation during particle-particle and particle-boundary interactions at each DEM time-step. The heat generated by impact is shared equally between the contacting bodies, independently of any other heat transfer effects in the simulation.

Provided by	**[Phase Interaction] >** > **Models** > *Optional Models*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Multiphase Interaction** > **Phase Interactions** > **Phase Interaction 1** > **Models** > **Impact Heat**
Requires	One or more defined regions Under **Lagrangian Multiphase** models, for at least one phase: • *Particle Type*: **DEM Particles** • *Optional Models*: **Energy** In phases under the **Multiphase Interaction** model, for at least two phases: • *Phase Interaction Topology*: **DEM Phase Interaction** Select the **DEM Phase Interaction** model and, in the properties, select phases. • *DEM Contact Model*: any • *Optional Models*: **Impact Heat**
Adds	**Impact Heat Rate** node under the **Impact Heat** model node
Properties	None, but see Impact Heat Rate Properties following.
Actions	None

Impact Heat Rate Properties

The **Impact Heat Rate** node appears directly under the **Impact Heat** model node.

Method	Selects the method for calculating q_r (see Eqn. (3244)). Available methods are **Linear** and **Field Function**. A node for the selected method appears under the **Impact Heat Rate** node.	
	Linear	See Linear Properties below.
	Field Function	Defines the impact heat rate using a field function (typically user-defined). The defined field function can only use constants and contact-based field functions. A

		field function node is added as a child to the **Impact Heat Rate** node.

Linear Properties

Frictional Work Fraction	The fraction of the frictional work that is converted to heat. See c_t in Eqn. (3244). The default value is `0.1`.
Damping Work Fraction	The fraction of the damping work that is converted to heat. See c_n in Eqn. (3244). The default value is `0.1`.

The Passive Scalar Transfer Model

The Passive Scalar Transfer model transfers conserved quantities of passive scalar from one particle to another when the particles are in contact.

In this model, when particles make contact, a quantity of passive scalar transfers from one particle to the other. For example, using this model, you can simulate a thin coating of negligible mass transferring from one particle to another as they scrape against each other.

Theory	See Passive Scalar Transfer for DEM Particles.	
Provided By	**[phase interaction]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Multiphase Interaction** > **Phase Interactions** > **[phase interaction]** > **Models** > **Passive Scalar Transfer**	
Requires	Basic DEM setup In **Lagrangian Multiphase** > **Lagrangian Phases** > **[phase]** > **Models**: *Particle Type*: **DEM Particles** *Particle Shape*: any shape *Optional Models*: **Passive Scalar** In **Multiphase Interaction** > **Phase Interactions** > **[phase interaction]** > **Models**: *Phase Interaction Topology*: **DEM Phase Interaction** (after setting properties, *First Phase* and *Second Phase* *Optional Models*: **Passive Scalar Transfer**	
Activates	Model Controls (child nodes)	**Contact Passive Scalar Transfers**

Contact Passive Scalar Transfers Node

This node holds nodes for each passive scalar interaction you want to model. These interactions can include contact between phases, contact of a phase with itself, and contact of a phase with a wall.

Before you can create interaction nodes, you must have at least one node for a passive scalar under the **Passive Scalars** model node, under **Lagrangian Multiphase**. See Creating One or More Lagrangian Phases.

To create an interaction node:

1. Right-click the **Contact Passive Scalar Transfers** node and select **New Interaction**. A menu of interactions displays.
2. From the menu, select **Area Weighted**, **Volume Weighted**, or **Increment**. A node for the new interaction appears under the **Contact Passive Scalar Transfers** node.
3. Set the properties for the new interaction node.

You can create an arbitrary number of interaction nodes.

Interaction Node Properties

Phase 1 Material **and** *Phase 2 Material*

Phase 1 Material is the passive scalar selected for the phase listed as *First Phase* under **DEM Phase Interaction**, and *Phase 2 Material* for the *Second Phase*. The abstract scalar "materials" are transferred between the first and second phases. Note that *Phase 1 Material* and *Phase 2 Material* can refer to the same material; that is, the two properties can have the same value.

Number

Read only. An index number for the interaction node.

Coefficient

For Area Weighted and Volume Weighted transfers, this is the coefficient k in Eqn. (3257), acting as the equivalent of a diffusion constant. The higher the value, the faster the transfer rate.

For Increment transfer, k is the amount of passive scalar transferred on each contact.

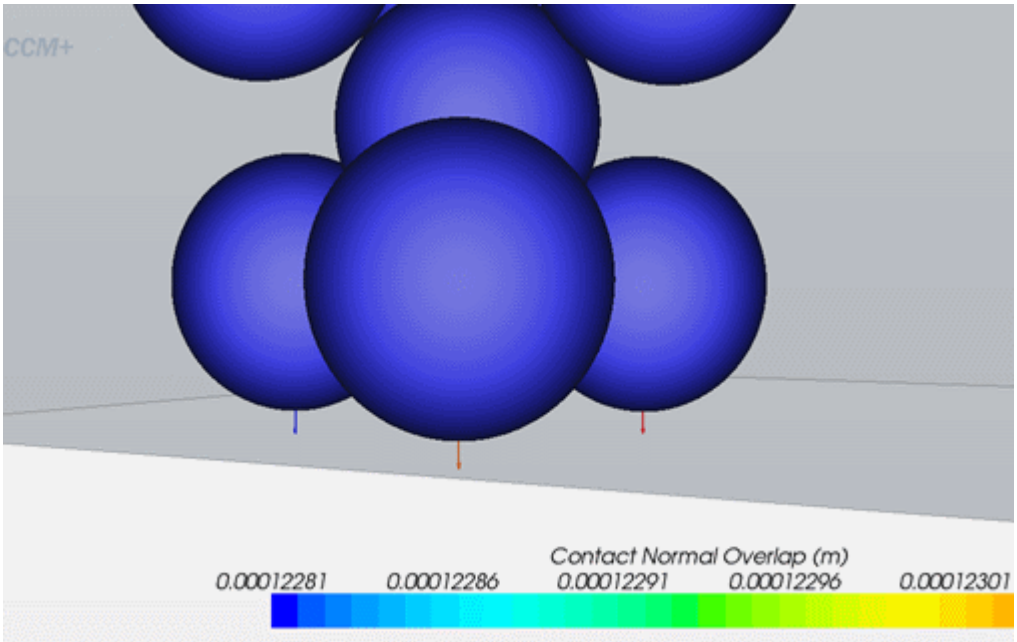
Flow Cutoff

Area Weighted and Volume Weighted transfers no longer take place when the transfer rate falls below this value. The default value is `1.0E-8`. Increment transfers do not have this property.

Visualizing and Reporting Contact

In general, collisions involving DEM particles are not the same as those involving Lagrangian parcels. With DEM particles, there can be more than one collision between particles or between a particle and a wall, if particles are modeled as spherical composites.

In the following illustration, arrows mark multiple collisions between a composite particle and the wall.



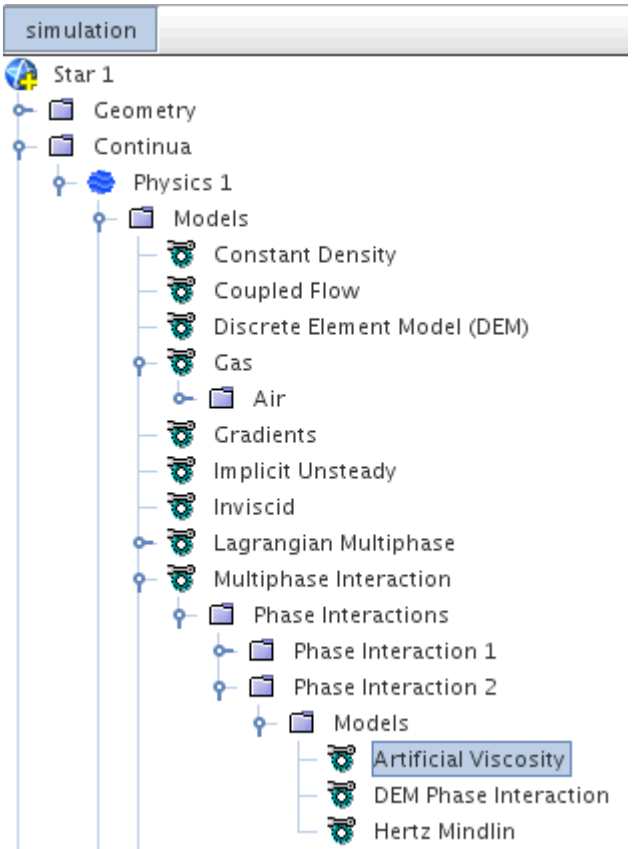
STAR-CCM+ provides discretized collision information for visualization and reporting, through the following field functions:

- **Contact Damping Force**
- **Contact Elastic Force**
- **Contact Normal**
- **Contact Normal Overlap**
- **Contact Relative Velocity**
- **Contact Tangential Overlap**

Modeling Artificial Viscosity

When two particles collide, or when a particle collides with a wall, the DEM module uses the impact velocity to predict the outcome. Impact velocities can increase to the same order of magnitude as the speed of sound in the particle material, which can cause the numerical models to over-predict the overlap between the colliding objects. The large overlap results in inaccurate results or unphysical ones, such as particles passing through walls. You can avoid or reduce these problems by applying the artificial viscosity model.

With the DEM model active in a physics continuum, artificial viscosity is an **Optional Model** under **Models** in a **Phase Interaction**.



The energy that artificial viscosity dissipates represents unresolved processes of damage, deformation, and entropy. Do not use the artificial viscosity model if you want to study these or similar collision effects in detail. Instead, use models that address those effects; for example, to study particle breakage, use the Bonded Particles model.

Artificial Viscosity Properties

Linear Coefficient	The linear coefficient Q_1 in Eqn. (3189), which must be nonnegative. The default value is 0.3.
Quadratic Coefficient	The quadratic coefficient Q_2 in Eqn. (3189), which must be nonnegative. The default value is 1.
Upper ramp limit	The upper ramp limit, p_{upper} in Eqn. (3190). The value must lie between zero and one, and p_{upper} must be larger than p_{lower}. The default value is 0.2.
Lower ramp limit	The lower ramp limit, p_{lower} in Eqn. (3190). The value must lie between zero and one, and p_{lower} must be smaller than p_{upper}. The default value is 0.1.

All these values are dimensionless.

The default value of Q_1, 0.3, is high, to enhance stopping at bounding surfaces. A value of 0.06 is more typical of shock problems.

The default value of Q_2, 1, is the value for an elastic fluid. Values typically range from 1.5 to 0.6.

The default values for p_{upper} and p_{lower}, 0.2 and 0.1, are high. Contact force models are valid only for values lower than 0.05, but the high values create a default situation where artificial viscosity comes into play even in borderline cases. Bring values down for stiff particles and up for soft ones.

Defining Injectors in the Discrete Element Method

An injector setup for DEM particles follows the Lagrangian methodology. Some of the Lagrangian injector types are not allowed for DEM particles. The main difference between Lagrangian and DEM particles with respect to the injector is the concept of a particle as opposed to a parcel.

For a Lagrangian phase (Material and Massless particle types) the parcel can represent many particles at once or a fraction of a particle. For DEM particles, the parcel always represents a single particle, except in the case of the Coarse Grain Particle model.

When setting up the injector, ensure that the injector injects particles into the system stress-free and does not violate the basic small overlap assumptions of the contact model: particles cannot overlap with other particles or with the wall. Particles that overlap are eliminated during injection. This condition imposes a limitation for injector flow rates when injecting particles too close to each other in space or time. As a result, the injector flow rate has a finite maximum value that depends on particle diameter, injection velocity, and distribution, as well as the unsteady iteration time.

The following special injector types are available for DEM particles only:

- Lattice Injector
- Random Injector

Contents:

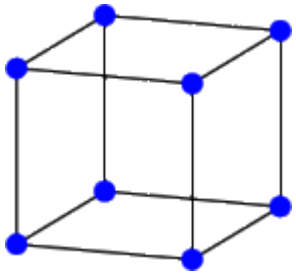
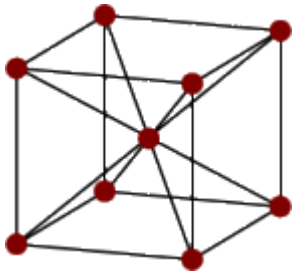
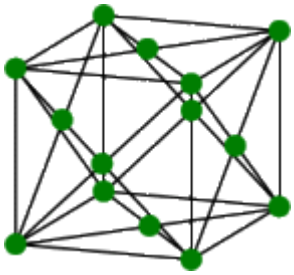
Lattice Injector
Random Injector

Lattice Injector

When used with the Discrete Element Method (DEM), the lattice injector allows injection of particles into the system with specified, mesh-independent spatial distributions.

You can use other injector types as usual. However, the main advantage when compared to the other injector types is the ability to introduce higher particle loading rapidly and make the particle distribution stable with respect to remeshing.

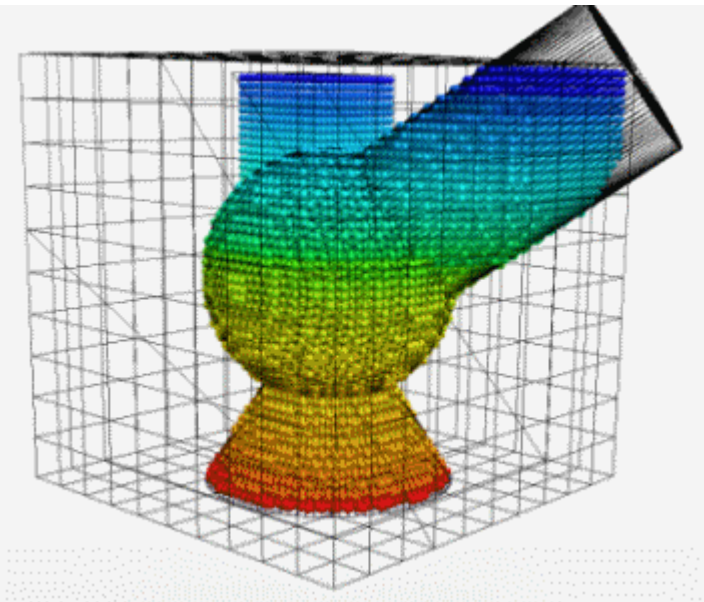
This injector places injection points at nodes of a regularly spaced lattice structure. Currently, the supported lattice types, which are selected in the Lattice property of the Lattice Structure injector condition, are:

		
Simple cubic	Body-centered cubic	Face-centered cubic

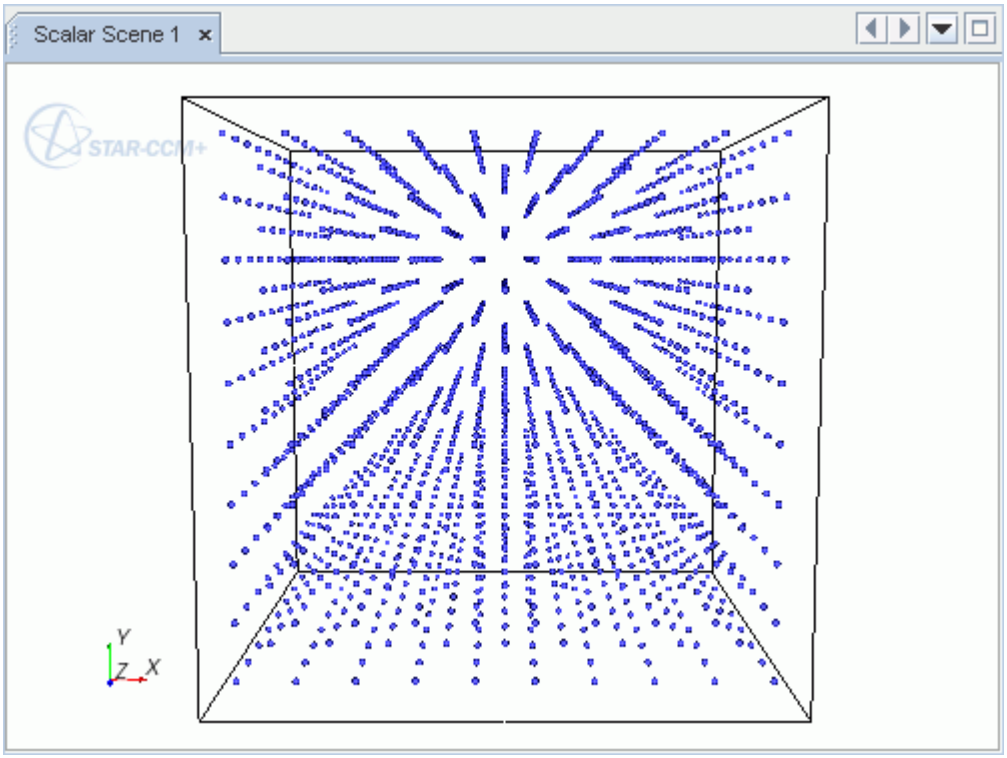
The Spacing injector value allows you to control the dimensions of the bounding box in the x, y, and z directions. The absolute placement of the lattice structure depends on a bounding box of all generating regions.

When you create a lattice injector, you can assign parts from the **Geometry** > **Parts** node or from the **Regions** node. If you assign a geometry part from the Parts node, first set the *Region* property of the part to match the fluid region that contains the lattice. The part does not need a mesh.

The only parts selectable from the **Parts** node are those parts from the New Shape Part menu: **block**, **cone**, **cylinder**, and **sphere**, shown combined in this example:



The following image displays particles in a body-centered cubic structure.



Random Injector

A random injector injects particles into a region with a random point distribution. This injector is suitable if you wish to inject particles in a non-uniform, arbitrary manner.

When you create a random injector, you can assign parts from the **Parts** node under the **Geometry** node, as well as from the **Regions** node. However, the only parts available from the **Parts** node are the parts that you create from the New Shape Part menu: **block**, **cone**, **cylinder**, and **sphere**. If you choose geometry parts, set the *Region* property of each part to match the region that contains the injector.

To speed up the injection process, assign non-zero initial velocities. Particles with actual velocities pack faster and make room for new particles to be injected.

The **Particle Packing Specification** controls the number of injected particles, using the *Porosity Limit*, *Particle Count*, or *Maximum Packing* method. *Porosity Limit* is the default.

- If the *Porosity Limit* method is chosen, the injector works as follows:
  1. Check if the Injection cycle limit has been reached. If not, continue; otherwise stop.
  2. Randomly inject a number of particles. The Number of Seeds property sets this quantity.
  3. Eliminate any overlapping particles.
  4. Calculate the quantity of particles per unit volume -- also known as the porosity.
  5. If the calculated porosity is less than the Porosity Limit, repeat steps 1–4. STAR-CCM+ continues injecting particles until the Porosity Limit is reached. The porosity limit for a fully packed volume is about 0.6.
- If the *Particle Count* method is chosen, the injector injects the specified number of non-overlapping particles, up to the limits of the volume.

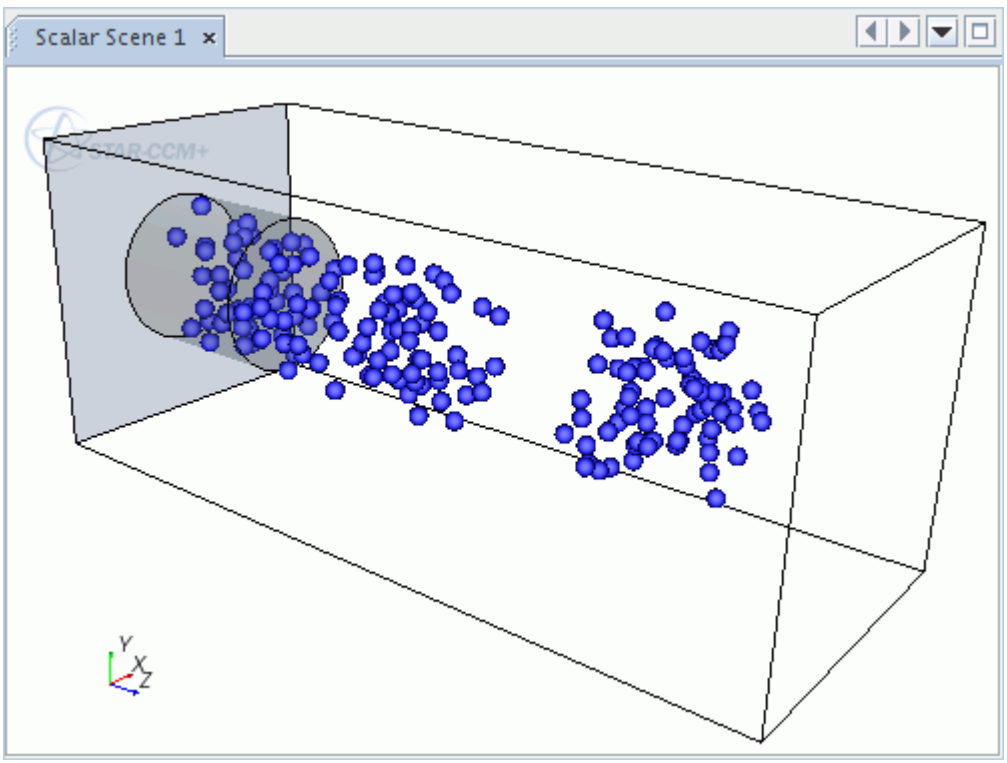
- If the *Maximum Packing* method is chosen, the injector works as follows:
  1. The injector injects a specified number of particle seeds of small starting volume.
  2. In the first time-step, spherical particles grow from the seeds until there is no room that is left for growth.

  Maximum Packing sets the Initial Porosity Limit value to 0.3. This value works for most cases. Do not change it except in special cases. If there is a diverged packing or an undesired packing configuration, decrease this value to reduce the scale-up factors.

  For Maximum Packing, observe the following requirements:
  - The particles must be Spherical or Coarse Grain.
  - Assign a single region to the injector *Inputs*, not a geometry part.
  - Only one injector in a region can use the Maximum Packing method. You can have multiple random injectors in a region, but only one can use the Maximum Packing method.
  - The boundary mode for the injector region walls must be set to DEM mode. See Setting Phase Boundary Conditions, Mode.

User-specified settings for contact models are not taken into account during injection. They are saved before injection, then replaced by model values, and finally restored after the particles are generated.

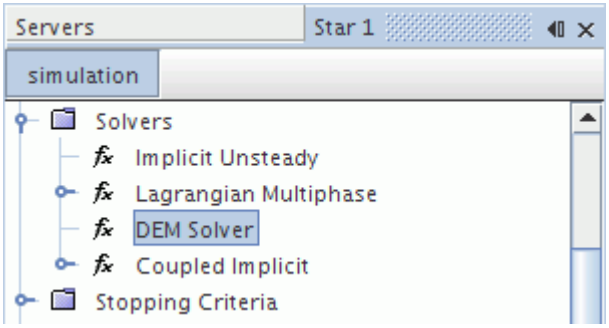
In the example below, the injector injects particles into the cylindrical region, which is connected to the main volume using internal interfaces.



What Is the Discrete Element Method Solver?

The DEM solver controls the solution update for the Discrete Element Method (DEM).

The DEM solver is represented in STAR-CCM+ by the **DEM Solver** node. Select the node to access the expert properties of the model.



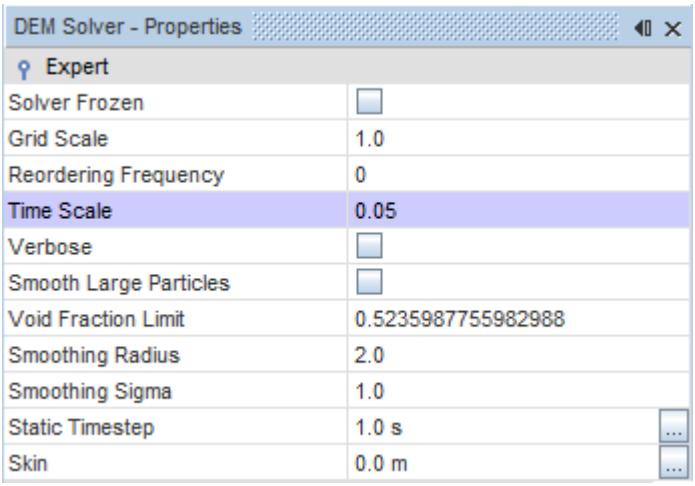
Controlling the DEM Time-Step

By default, the duration of the Rayleigh wave (Eqn. (3186)) across the surface of the sphere limits the DEM time-step. Furthermore, a minimum of 10 time-steps are necessary in order to resolve particle collisions correctly (Eqn. (3187)). The distance traveled by particles within a single time-step is constrained (Eqn. (3188)) to account for contact among particles and between particles and walls.

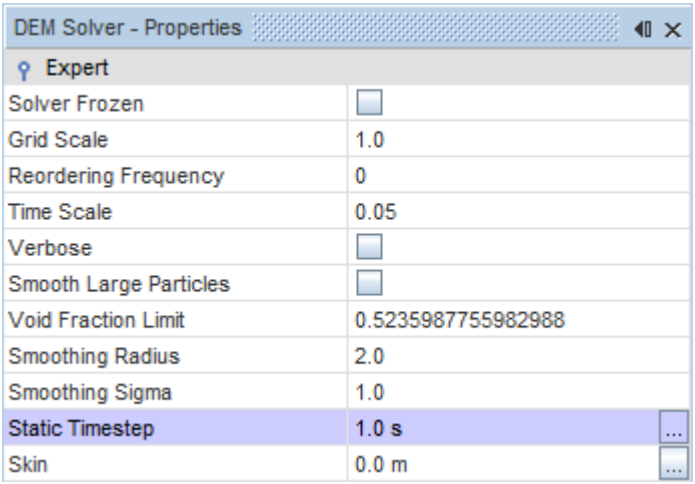
To account for the particle movements with precision, STAR-CCM+ uses substeps to assess physics such as flow and gravity (if applicable) as well as DEM. This results in dynamic time-step calculations by default.

To control the time-step, you can control the dynamic calculations of the solver in one of two ways:

1. Change the **Time Scale** property to reduce the allowed fraction of the Rayleigh wave limit.



2. Alternatively, enter the desired time-step in the *Static Timestep* property. This causes STAR-CCM+ to use the value of the time-step that you enter. Do not make this option your first choice, as it can cause a failure to converge.



To restore dynamic calculations, simply enter the default value of 0.0 in this property.

Discrete Element Method Solver Expert Properties

Grid Scale	Sets the scale of the grid cell to the particle diameter.	
Reordering Frequency	Sets the number of iterations until STAR-CCM+ reorders the sequence in which DEM particles are accessed in memory. The default value is 0 (no reordering).	
Time Scale	Sets the allowed fraction of the Rayleigh wave limit (Eqn. (3186)).	
Verbose	Provides more output for this value while the simulation is running. This property can be useful for debugging problems when they occur.	
	Activated	Activates verbose output.
	Deactivated	Leaves output at its default.
Smooth Large Particles	Smooths large particles on the Eulerian mesh for two-way coupling.	
Solver Frozen	Prevents or allows the solver from updating the solution while iterating. Do not make activating this feature your first choice in most situations, as it is primarily a “debugging” tool and not supported for normal simulation work.	
	Activated	The solver does not update the solution.
	Deactivated	The solver updates the solution.
Void Fraction Limit	Sets the void fraction limit for particle smoothing.	
Smoothing Radius	Sets the smoothing radius (normalized to the particle radius).	
Smoothing Sigma	Sets the smoothing distribution sigma (normalized to the particle radius).	

Static Timestep	Instructs STAR-CCM+ to use the value of the time-step that you enter. This option should not be your first choice, as it can cause a failure to converge. The default setting of 0.0 leaves the dynamic calculation uninterrupted.
Skin	Sets a distance around each particle, the “skin”. STAR-CCM+ runs contact detection only when the particle moves through a distance equal to the skin thickness. The default value is 0.

Guidelines for DEM

DEM gives high resolution, but at a high cost in computational power. Guidelines for DEM mostly consist of finding ways to reduce the computational cost.

Use the following guidelines:

- Maximize the number of processors. However, do not exceed about 1000 particles per processor. If you continue to add processors, speed goes down as processors spend more time communicating with each other. (So, for three million particles, use about three thousand processors, not more.)
- In composite particles, minimize the number of spheres in a particle.
- Maximize particle size or use the Coarse Grain Model to maximize parcel size.

  One DEM particle can represent several objective particles. Using larger particles reduces particle count and increases the DEM time-step, which reduces both the time that is needed and the number of processors needed. Consider the example of a simulation with particles in a rotating drum. Enlarging the particles from 2 mm to 3 mm produced the same results in half the time, using 5 processors instead of 18.

  Using the Coarse Grain model reduces time that is spent on contact physics while maintaining full accuracy for particle-fluid interactions. Coarse Grain is best used for fluidized beds applications, where accuracy of particle-fluid interactions is more important than accuracy of particle-particle interactions.

  Note: The change in size in the example simulation also results in slightly different bulk properties (such as angle of repose) that depend on contact physics. You can often compensate for similar changes in bulk properties by slightly adjusting the parameters of the DEM Contact Model, such as friction.

  Note: To maintain the accuracy of the DEM model, scaled-up particles or coarse-grain parcels must still be smaller than the smallest significant geometrical feature of the surroundings. In the example simulation, the parcels had to be smaller than the smallest important length scale in the geometry.

- Minimize Young's modulus. This practice increases the DEM time-step, resulting in faster computation. For applications with low-confinement shear flow regimes, you can often reduce Young’s modulus by several orders of magnitude and still get accurate results. For example, in one simulation, going from 1000 MPa to 100 MPa reduced time to one third, using 5 processors instead of 18.
- Make the mesh as coarse as practical.
- Reduce the modeled physical time as much as practical:
  - When possible, optimize injector settings to introduce the maximum number of particles in the minimum physical time.

  - Sometimes, you can reduce time by increasing physical speeds of moving geometrical boundaries, such as using higher rpm on a rotating boundary.
  - If you only need to establish the direction of a trend, you can use a short physical time.
- Use double precision in simulations that integrate quantities over a large number of time-steps or that deal in a large number of small energy exchanges. For example:
  - Simulations where particles recirculate or settle within the simulation domain
  - Predicted particle temperature increases in each sub-step that are negligible compared to the ambient temperature
  - Chemical reactions
  - Mass transfers between particles
- To speed memory access, use the DEM solver property **Reordering Frequency**. On reordering, STAR-CCM+ resets the sequence in which DEM particles are accessed in memory. For quasi-stationary cases, it is enough to reorder once, after the particles are injected and settle. For rapidly moving particles, reorder at least once every 20 iterations. The benefit of reordering decreases with repetition.
- To reduce time that is spent running contact detection, use the DEM solver property **Skin**. The **Skin** property defines a volume around each DEM particle, a “skin.” The solver runs contact detection only when the particle moves through a distance equal to the skin thickness. As a rule of thumb, set skin thickness to $10v_p t$ where v_p is a characteristic particle velocity and t is length of time-step.

Discrete Element Model Formulation

This section provides the formulations for models available under the DEM model.

Contents:

Momentum Balance for Discrete Element Method Particle
Discrete Element Method Particle Times Scale
Contact Force Modeling
Conduction Heat Transfer Model
Particle Bonding Model
Simple Failure Model
Constant Rate Damage Model
Erosion
Linear Cohesion Model
Rolling Resistance Model
Discrete Element Method Drag Coefficient
Drag Torque Model
Discrete Element Method Lift Forces
Artificial Viscosity Model
Impact Heat Model
Passive Scalar Transfer Formulation

Momentum Balance for Discrete Element Method Particle

The momentum balance for a DEM particle is derived from the momentum balance of the material particle.

The momentum balance for a material particle is computed as Eqn. (2969). The DEM modeling uses body force representing inter-particle interaction due to particle contacts with other particles and with mesh boundaries:

$$\mathbf{F}_b = \mathbf{F}_g + \mathbf{F}_u + \mathbf{F}_c \tag{631}$$

where:

$$\mathbf{F}_c = \sum\nolimits_{\text{neighbor particles}} \mathbf{F}_{\text{contact}} + \sum\nolimits_{\text{neighbor boundaries}} \mathbf{F}_{\text{contact}} \tag{632}$$

A user-specified contact model controls the exact formulation of contact force.

Besides the standard Lagrangian linear momentum equation, the DEM particle equations of motion incorporate angular momentum conservation equations:

$$\frac{d}{dt}\mathbf{L}_p = \frac{d}{dt}\left(\mathbf{I}_p \omega_p\right) = \sum\nolimits_{\text{neighbor particles}} \mathbf{T}_{\text{contact}} + \sum\nolimits_{\text{neighbor boundaries}} \mathbf{T}_{\text{contact}} \tag{633}$$

where contact torque is computed as:

$$\mathbf{T}_{\mathrm{c}} = \mathbf{r}_{\mathrm{c}} \times \mathbf{F}_{\mathrm{c}} - \mu_{\mathrm{r}} |\mathbf{r}_{\mathrm{c}}| |\mathbf{F}_{\mathrm{c}}| \frac{\boldsymbol{\omega}_p}{|\boldsymbol{\omega}_p|} \tag{634}$$

and where:

- $\mathbf{L}_p$ is the particle angular momentum.
- $\mathbf{I}_p$ is the particle moment of inertia.
- $\mathbf{T}_{\mathrm{c}}$ represents the torque on an individual particle due to contact force acting on the particle at a point other than the particle center of gravity.
- $\mathbf{r}_{\mathrm{c}}$ is a vector from the particle center of gravity to the contact point.
- $\mathbf{F}_{\mathrm{c}}$ is the contact force acting on the particle.
- μ_r is the coefficient of rolling friction.
- $\boldsymbol{\omega}_p$ is the particle angular velocity.

The detailed torque formulation is supplied by the inter-particle contact model.

Discrete Element Method Particle Times Scale

The maximum time-step that is allowed for a DEM particle is constrained by the assumption that the force acting on a particle is only affected by the particle's immediate neighbors during a single time-step duration.

The time-step is therefore limited by the time it takes the Rayleigh wave to propagate across the surface of the sphere to the opposite pole [110]:

$$\tau_1 = \pi \frac{R_{min}}{V_{Rayleigh}} \tag{635}$$

where R_{min} is the minimal sphere radius.

The Rayleigh wave velocity depends on material properties, and the exact value is obtained as a solution to a secular equation [116]. STAR-CCM+ uses an approximation of the solution that provides sufficient precision without incurring high computational cost [113], [114].

Besides the wave propagation, more time-step limiting criteria are applied to moving particles. The duration of impact of two perfectly elastic spheres assuming the Hertz contact theory was derived by Timoshenko [120] to be:

$$\tau_2 = 2.94\left(\frac{5\sqrt{2}\pi\rho}{4}\frac{1-\nu^2}{E}\right)^{\frac{2}{5}}\frac{R}{\sqrt[5]{v_{impact}}} \tag{636}$$

In order to resolve the collisions a minimum of 10 time-steps is required for collision resolution.

The final restriction on a DEM particle time-step is geometric. It is based on the assumption that particles must not move too far within the time-step. This condition prevents missing contacts between DEM particles, as well as particles and walls. Therefore each particle is constrained such that it takes at least 10 time-steps for the particle to move the full length of the radius.

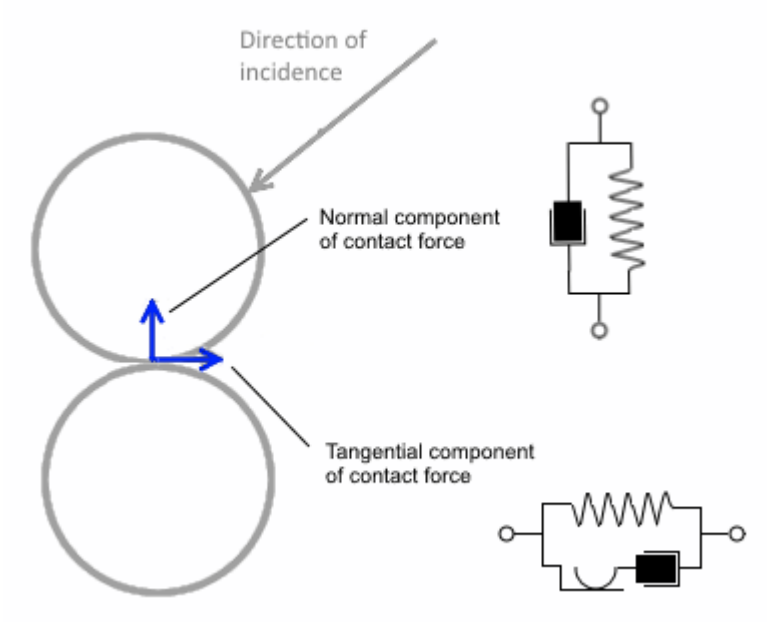
$$\tau_3 = \left(\frac{R}{v_{particle}}\right) \tag{637}$$

The final particle time-step is determined as a minimum of τ_1, τ_2, τ_3. In practice τ_1 is typically the limiting factor while τ_2 and τ_3 only constrain particles that are moving fast, or the Young's modulus of the material is set low in order to accelerate the simulation.

Contact Force Modeling

Contact force formulation in DEM is typically a variant of the spring-dashpot model. The spring generates repulsive force pushing particles apart and the dashpot represents viscous damping and allows simulation of collision types other than perfectly elastic.

The forces at the point of contact are modeled as a pair of spring-dashpot oscillators. A parallel linear spring-dashpot model represents the normal force and a parallel linear spring-dashpot in series with a slider represents the tangential direction of force with, respect to the contact plane normal vector. In both, the spring accounts for the elastic part of the response and the dashpot accounts for energy dissipation during collision.



STAR-CCM+ provides three contact models:

- Hertz Mindlin
- Linear Spring
- Walton Braun

Hertz-Mindlin No-Slip Contact Model

The Hertz-Mindlin contact model is a variant of the non-linear spring-dashpot contact model that is based on the Hertz-Mindlin contact theory [110], [105]. The forces between two spheres, A and B, are described by the following set of equations.

$$\mathbf{F}_{\text{contact}} = \mathbf{F}_n + \mathbf{F}_t \tag{638}$$

where $\mathbf{F}_n$ is the normal and $\mathbf{F}_t$ is a tangential force component.

The normal direction is defined by the following:

- Normal force:

$$\mathbf{F}_n = -K_n d_n - N_n v_n \tag{639}$$

- Normal spring stiffness:

$$K_n = \frac{4}{3} E_{eq} \sqrt{d_n R_{eq}} \tag{640}$$

- Normal damping:

$$N_n = \sqrt{\left(5 K_n M_{eq}\right)} N_{n\ \text{damp}} \tag{641}$$

  where $N_{n\ \text{damp}}$ is the normal damping coefficient as given in Eqn. (645).

The tangential direction is defined by the following:

- The tangential force is defined by $-K_t d_t - N_t v_t$ if $|K_t d_t| < |K_n d_n| C_{fs}$, where C_{fs} is a static friction coefficient. Otherwise:

$$\mathbf{F}_t = \frac{|K_n d_n| C_{fs} d_t}{|d_t|} \tag{642}$$

- Tangential spring stiffness:

$$K_t = 8 G_{eq} \sqrt{d_n R_{eq}} \tag{643}$$

- Tangential damping:

$$N_t = \sqrt{\left(5 K_t M_{eq}\right)} N_{t\,\mathrm{damp}} \tag{644}$$

where $N_{t\,\mathrm{damp}}$ is the tangential damping coefficient as given in Eqn. (646).

$$N_{n\ \mathrm{damp}} = \frac{-\ln\left(C_{n\,\mathrm{rest}}\right)}{\sqrt{\pi^2 + \ln\left(C_{n\,\mathrm{rest}}\right)^2}} \tag{645}$$

$$N_{t\ \mathrm{damp}} = \frac{-\ln\left(C_{t\ \mathrm{rest}}\right)}{\sqrt{\pi^2 + \ln\left(C_{t\ \mathrm{rest}}\right)^2}} \tag{646}$$

Here, $C_{n\ \mathrm{rest}}$ and $C_{t\ \mathrm{rest}}$ are the normal and tangential coefficients of restitution, model parameters set by the user. If $C_{n\ \mathrm{rest}} = 0$, then $N_{n\,\mathrm{damp}} = 1$, and if $C_{t\ \mathrm{rest}} = 0$, then $N_{t\,\mathrm{damp}} = 1$.

The equivalent radius is defined as:

$$R_{eq} = \frac{1}{\frac{1}{R_A} + \frac{1}{R_B}} \tag{647}$$

The equivalent particle mass is:

$$M_{eq} = \frac{1}{\frac{1}{M_A} + \frac{1}{M_B}} \tag{648}$$

The equivalent Young's modulus is expressed as:

$$E_{eq} = \frac{1}{\frac{1-\nu_A^2}{E_A} + \frac{1-\nu_B^2}{E_B}} \tag{649}$$

The equivalent shear modulus is:

$$G_{eq} = \frac{1}{\frac{2\left(2-\nu_A\right)\left(1+\nu_A\right)}{E_A} + \frac{2\left(2-\nu_B\right)\left(1+\nu_B\right)}{E_B}} \tag{650}$$

where:

- M_A and M_B are masses of sphere A and sphere B.
- d_n and d_t are overlaps in the normal and tangential directions at the contact point.

- R_A and R_B are the radii of the spheres.
- E_A and E_B are the Young's modulus of the spheres.
- ν_A and ν_B are the Poisson's ratios.
- v_n and v_t are the normal and tangential velocity components of the relative sphere surface velocity at the contact point.

For particle-wall collisions, the formulas stay the same, but the wall radius and mass are assumed to be $R_{wall} = \infty$ and $M_{wall} = \infty$, so the equivalent radius is reduced to $R_{eq} = R_{particle}$ and equivalent mass to $M_{wall} = M_{particle}$.

Di Renzo and Di Maio [105] proposed several formulations for detailed treatment of tangential micro-slip and tangential force computation. STAR-CCM+ uses the formulation that Tsuji proposed [122], where the tangential force is assumed to be non-linear but the detail micro-slip tracking is replaced by analytical expression. The resulting code is computationally efficient while still matching experimental data.

Linear Spring Contact Model

This model is based on the work of Cundall and Strack [103]. The normal and tangential forces $\mathbf{F}_n$ and $\mathbf{F}_t$ are defined as they are in Eqn. (639) and Eqn. (642), but parameters K_n, K_t, N_n, and N_t have the following definitions:

- K_n is the normal spring constant.
- K_t is the tangential spring constant.
- $N_n = 2N_{n\ \mathrm{damp}}\sqrt{K_n M_{eq}}$ where:
  - $N_{n\ \mathrm{damp}}$ is the normal damping coefficient as in Eqn. (645).
  - M_{eq} is the equivalent particle mass as in Eqn. (648).
- $N_t = 2N_{t\ \mathrm{damp}}\sqrt{K_t M_{eq}}$ where $N_{t\ \mathrm{damp}}$ is the tangential damping coefficient as in Eqn. (646).

When the Linear Spring model uses the Particle Material Based method, the spring stiffness is estimated based on the assumption that the linear model is a linearized Hertz model, in which the maximum normal contact force is:

$$K_n \delta_{max} = \frac{4}{3} E_{eq} \sqrt{R_{eq} \delta_{max}} \delta_{max} \tag{651}$$

where:

- δ_{max} is the maximum overlap.
- E_{eq} is the equivalent Young's modulus defined in Eqn. (649).
- R_{eq} is the equivalent radius defined in Eqn. (647).

To calculate the spring constants, let:

$$\lambda_{max} = \frac{\delta_{max}}{R_{eq}} \tag{652}$$

This gives the normal spring constant K_n as:

$$K_n = \frac{4}{3}\sqrt{\lambda_{max}}E_{eq}R_{eq} \tag{653}$$

The tangential spring constant K_t is:

$$K_t = 8\sqrt{\lambda_{max}}G_{eq}R_{eq} \tag{654}$$

where G_{eq} is the equivalent shear modulus defined in Eqn. (650).

Walton Braun Hysteretic Contact Model

The Walton Braun hysteretic plastic-elastic contact system is characterized by the following relations:

$$\mathbf{F}_n = -K_1\delta\mathbf{n} \tag{655}$$

when the contact is loading, and:

$$\mathbf{F}_n = -K_2\left(\delta - \delta_{\text{deformation}}\right)\mathbf{n} \tag{656}$$

during unloading, where:

$$K_1 = 1.6\pi RY_0 \tag{657}$$

$$K_2 = \frac{1}{E_f}K_1 \tag{658}$$

$$Y_0 = E \cdot \text{YieldStressFraction} \tag{659}$$

$$\delta_{\text{deformation}} = \delta_{\max}\left(1 - E_f\right) \tag{660}$$

E is the material Young modulus, δ is the contact normal overlap, $\mathbf{n}$ is the contact normal vector, and Yield Stress Fraction is a user-controllable model property defining the onset of plastic deformation. E_f is a user-controllable energy fraction that defines the amount of energy that is recovered during the unloading and $\delta_{\max}$ is the maximum overlap that is reached during the contact loading.

Contact Cycle

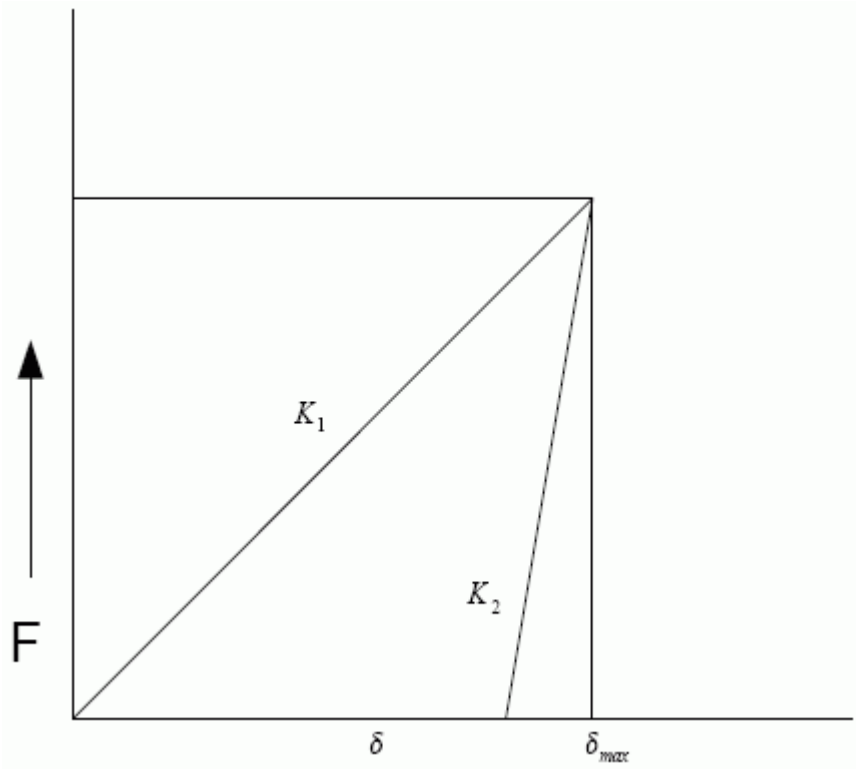
When particle contact is formed, the state of the contact is tracked at each time-step as follows:

- Contact is considered loading when $\mathbf{v}_r \cdot \mathbf{n} \geq 0$ and $\delta > \delta_{\max}$, where $\mathbf{v}_r$ is relative surface velocity.

  When contact is loading, the force is calculated using K_1 stiffness and $\delta_{\max}$ is updated to a value of δ.
- If the contact is not loading, it is either unloading ($\mathbf{v}_r \cdot \mathbf{n} < 0$) or reloading ($\delta \leq \delta_{\max}$). During unloading, the K_2 stiffness is used to compute the normal force and the value of $\delta_{\max}$ is not updated.
- If the particles separate completely, the contact tracking information is removed and subsequent contact is treated as new.

Geometric Interpretation

The loading/unloading cycle can be geometrically represented as follows:



Conduction Heat Transfer Model

Two bodies are assumed to exchange heat through conduction when they make physical contact. The two bodies can be two particles or a particle and a wall.

There are four cases:

- Particle-particle (see [101]):

  The heat exchange q_{ij} from particle i to particle j is:

  $$q_{ij} = 4r_c k\left(T_j - T_i\right) \tag{661}$$

  where:

  - r_c is the contact area radius.
  - k is the equivalent thermal conductivity of the two particles, where

    $\frac{1}{k} = \frac{1}{k_i} + \frac{1}{k_j}$

    where k_i and k_j are the thermal conductivities for particles i and j.
  - T_i and T_j are the temperatures of particle i and particle j.
- Particle-wall, adiabatic wall: heat exchange is zero.
- Particle-wall, fixed wall temperature:

The heat exchange q_{iw} from the wall to particle i is:

$$q_{iw} = 4r_c k\left(T_w - T_i\right) \tag{662}$$

where:

- k is the particle thermal conductivity.
- T_i and T_w are the temperatures of particle i and the wall.

- Particle-wall, fixed wall heat flux:

  The heat exchange q_{iw} from the wall to particle i is:

$$q_{iw} = A_c q_w \tag{663}$$

  where:

  - A_c is the contact area between the particle and the wall.
  - q_w is the heat flux density of the wall.

Two-Way Coupling Model

If the simulation includes a continuous phase, its energy state is unaffected by particle-particle and particle-wall heat conduction unless you activate the Two-Way Coupling model. With this model active, the Lagrangian energy source term is added to the energy equation of the continuous phase.

The linearized wall heat flux, q_w, is expressed as:

$$q_w = a + bT_c + cT_w + dT_w{}^4 \tag{664}$$

where:

- T_c and T_w are the cell and wall temperatures, respectively.
- a, b, c, and d are wall heat flux coefficients.

The contribution of DEM particle-wall conductive heat flux to the wall heat flux can be made by modifying wall heat flux coefficients a and c as follows:

For fixed wall heat flux conditions, only a needs to be modified:

$$a = \sum \frac{A_c}{A} q_w \tag{665}$$

where:

- A_c is the particle-wall contact area.
- A the boundary face area.

For fixed wall temperature conditions, a and c are modified as:

$$\begin{aligned} a &= \sum 4r_c k T_p \\ c &= \sum 4r_c k \end{aligned} \tag{666}$$

where:

- k is the particle thermal conductivity.
- r_c the contact radius.
- T_p the particle temperature.

Particle Bonding Model

This section describes the governing equations of the Particle Bonding model.

The Particle Bonding model is used by both the Parallel Bonds model and the Bonded Particle (clumps) model. The Parallel Bonds model uses Particle Bonding to describe interactions between the constituent particles, but also models elastic collisions between particles, based on the elastic contact models chosen by the user. The Bonded Particle (clumps) model uses only the Particle Bonding model.

The Particle Bonding model simulates cumulative damage to bonds. When an interaction producing some damage but falls short of failure, the amount of that damage is reflected in the reduced strength of the bond.

The parallel bond forces are represented in the following governing equations [115].

Force and torque on a particle due to parallel bonds is:

$$\mathbf{F}_i = F_n \mathbf{n}_i + F_s \mathbf{t}_i \tag{667}$$

$$\mathbf{M}_i = M_n \mathbf{n}_i + M_s \mathbf{t}_i \tag{668}$$

where the terms F_n, M_n denote the normal and F_s, M_s denote the shear components of forces and torques with respect to the contact plane, and:

$$\Delta F_n = k_n A \Delta U_n \tag{669}$$

$$\Delta F_s = -k_s A \Delta U_s \tag{670}$$

$$\Delta M_n = -k_s J \Delta \Omega_n \tag{671}$$

$$\Delta M_s = -k_n I \Delta \Omega_s \tag{672}$$

where:

$$A = \pi R^2 \tag{673}$$

$$I = \frac{1}{4}\pi R^4 \tag{674}$$

$$J = \frac{1}{2}\pi R^4 \tag{675}$$

$$k_n = \frac{E}{R_a + R_b} \tag{676}$$

$$k_s = 6\left(\frac{G}{R_a + R_b}\right) \tag{677}$$

$$\Delta U = V_{\text{relative}} \Delta t \tag{678}$$

where ΔU_n is the normal component of ΔU, and ΔU_s is the tangential component of ΔU.

$$\Delta\Omega = \omega_{\text{relative}}\Delta t \tag{679}$$

where $\Delta\Omega_n$ is the normal component of $\Delta\Omega$, and $\Delta\Omega_s$ is the tangential component of $\Delta\Omega$.

The parallel bond can be treated as brittle and disintegrative when maximum shear and tensile stresses exceed prescribed limit values. The stress limit values are calculated based on beam theory to be:

$$\delta_{\max} = \frac{-F_n}{A} + \frac{|M_s|R}{I} \tag{680}$$

$$\tau_{\max} = \frac{|F_s|}{A} + \frac{|M_n|R}{J} \tag{681}$$

If either value exceeds the predefined maximum, the bond is removed and the forces are further calculated only based on the granular interaction of particles.

Simple Failure Model

The Simple Failure model breaks a bond in the simulation if tensile stress or shear stress on the bond exceed given maximum values.

If $\delta_{\max} > \sigma_m$ or $\tau_{\max} > \sigma_s$, the bond fails. See Eqn. (680) and Eqn. (681) under Particle Bonding Model.

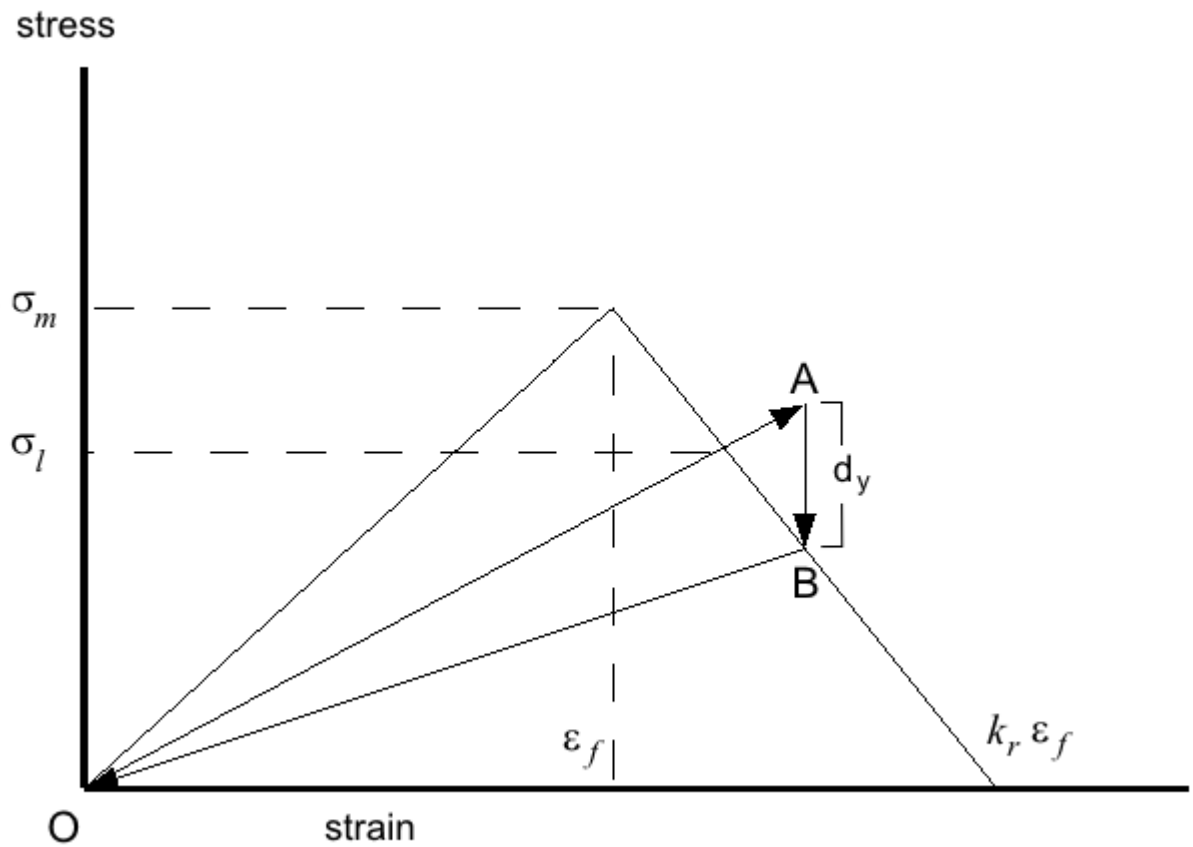
- $\delta_{\max}$ is the maximum tensile stress.
- σ_m is the current shear stress, determined by the **Tensile Strength Method** property of the Simple Failure model.
- $\tau_{\max}$ is the maximum shear stress.
- σ_s is the current shear stress, determined by the **Shear Strength Method** property of the Simple Failure model.

See What Is the Simple Failure Model?

Constant Rate Damage Model

The Constant Rate Damage model simulates the weakening and breaking of bonds between particles in a clump, resulting from collisions.

The Constant Rate Damage model follows the stress-strain curve shown in the following figure.



The point (ε_f, σ_m) is the point of strain ε_f at failure and maximum stress σ_m induced by that strain. Once conditions reach the point (ε_f, σ_m), the model calculates the damage required to return the stress to the failure surface, defined by the line from (ε_f, σ_m) to ($k_r\,\varepsilon_f$, 0), where k_r is the fracture softening modulus, a model parameter with default value 2 at constant strain. For stability, k_r should be greater than 1. [111]

If d_i is the damage at time-step i, the damage at time-step $i+1$ is:

$$d_{i+1} = \min\left(d_y,\ d_v,\ 1\right) \tag{682}$$

where:

d_v is the amount of damage produced by the fastest possible speed of crack growth:

$$d_v = d_i + 2\sqrt{d_i}\frac{C_r}{r_b}\Delta t + \left(\frac{C_r}{r_b}\right)^2 \Delta t^2 \tag{683}$$

- C_r is the Rayleigh wave speed.
- r_b is the bond radius.
- Δt is the size of the time-step.

$$d_y = \frac{k_r}{k_r - 1}\left(1 - \left(1 - d_i\right)\frac{\sigma_m}{\sigma_o}\right) \tag{684}$$

- k_r is the fracture softening modulus, a model parameter, default value 2.
- σ_m is the current shear stress, determined by the **Tensile Strength Method** property of the Constant Rate Damage model.
- σ_o is the maximum tensile stress for the bond. [100], [121]

$$\sigma_o = \sqrt{\max\left(0,\ \delta_{\max}\right)^2 + \tau_{\max}} \tag{685}$$

where:

- $\delta_{\max}$ is the maximum tensile stress.
- $\tau_{\max}$ is the maximum shear stress.

If $\sigma_o > \sigma_l$, damage is evolving.

$$\sigma_l = \sigma_m \frac{k_r\left(1 - d_i\right)}{\left(k_r - 1\right)\left(1 - d_i\right) + 1} \tag{686}$$

See What Is the Constant Rate Damage Model?

Erosion

Erosion in DEM is the sum of impact erosion and abrasive erosion.

Impact Erosion

Impact erosion is simulated in DEM by the Impact Wear model and the Impact Wear Rate field function, but is based on the same range of formulations as the Lagrangian Erosion model. See Lagrangian Erosion. Impact erosion is erosion due to direct impact of particles on the eroded surface.

The impact erosion rate is defined as the mass of wall material eroded per unit area per unit time. It is calculated on wall, baffle, and contact boundary faces by accumulating the damage that each particle impact does on the face:

$$E_f = \frac{1}{A_f}\Sigma_{\pi(f)}\dot{m}_\pi e_r \tag{687}$$

where:

- A_f is the area of the face.
- $\dot{m}_\pi$ is the mass flow rate of particles in parcel π impacting on the face.
- e_r is the erosion ratio.

The summation is over all parcels which strike the face in this iteration or time-step. The erosion rate therefore depends on the flow (whether and how particles impact on the wall) and the chosen method for the erosion ratio.

Abrasive Erosion

Abrasive erosion is simulated by the Abrasive Wear model and the Abrasive Wear Rate field function, and is available in DEM and EMP. Abrasive erosion is due to scouring, where particles scrape tangentially along walls or strike at low angles.

$$E_f = \frac{1}{A_f dt} \sum_{\pi(f)} e_r \tag{688}$$

where:

- dt is the fluid time-step.
- $e_r = aFs$ is the erosion ratio that is calculated with the Archard correlation, described in the following section, or with the field function method.

Archard Correlation

The Archard correlation [98] for the Abrasive Rate erosion ratio is:

$$e_r = aFs \tag{689}$$

where:

- a is the Abrasive Wear Coefficient, with a default value of 0.01 kg/J. The default value of 1E-2 is the high end of the recommended range of values, 1E-8 to 1E-2, mild wear to severe wear. (See Archard Properties.)
- F is the normal force.
- s is the sliding distance.

Linear Cohesion Model

The Linear Cohesion model describes the cohesion force between two spherical particles.

This variant of cohesion modeling can use either of two models that are described in published literature: the Johnson-Kendall-Roberts (JKR) model [110] or the Derjaguin-Muller-Toporov (DMT) model [99].

Cohesion force between two spherical particles is expressed in both cases as:

$$F_{cohesion} = R_{min} W \pi F \tag{690}$$

where R_{min} is the minimal radius of surfaces in contact, W is the work of cohesion $\left[J/m^2\right]$, and F is a multiplication model blending factor with values as follows:

- $\frac{3}{2}$ for the JKR model
- 2 for the DMT model

The principal difference between JKR and DMT model formulation is the surface area where the cohesion force is acting. In the JKR model, this area is limited to direct contact, while in the DMT model it also incorporates the “neck” area in immediate contact proximity.

Rolling Resistance Model

This section covers the rolling resistance methods applicable to DEM particles.

The following rolling resistance methods are applicable to DEM particles:

- Force Proportional Method
- Constant Torque Method
- Displacement Damping Method

If you are using the Hertz-Mindlin, Walton-Braun, or Bonded Particle contact models together with the Rolling Resistance model, the Rolling Resistance model applies a rolling friction coefficient to the contact models.

Force Proportional Method

This method only supplies a coefficient of rolling resistance, μ_r, also called the coefficient of rolling friction. It can take on any positive real value. The default value is 0.001.

Constant Torque Method

The Constant Torque method applies a torque to resist rolling, calculated as:

$$M_r = \frac{-\omega_{rel}}{|\omega_{rel}|}\mu_r R_{eq} F_n \tag{691}$$

where:

- ω_{rel} is the relative rotation, given as $\omega_{rel} = \omega_i - \omega_j$, where:
  - ω_i is the rotation of the particle for which the resistance moment M_r is being calculated.
  - ω_j is the rotation of the other particle in the contact.
- μ_r is the coefficient of rolling friction, also called coefficient of rolling resistance. The default value is 0.1.
- R_{eq} is the particle equivalent radius, that is, the harmonic mean of the radii of the two particles.
- F_n is the magnitude of the contact force.

This method also uses a “Moment Limiter.” This function limits the size of the angular momentum of the particle, measured around the axis of moment. This maximum is 0.08 of the moment that negates the lesser of the two rotations ω_i and ω_j in the next time-step.

This limit can control specious rotations in particle compacts. These rotations can arise from a large number of contacts or from high compact pressure.

Displacement Damping Method

The Displacement Damping method combines a viscous damping resistance with a spring-like resisting torque generated by elastic energy stored in the particle as it tries to turn around a finite contact area.

The spring resistance torque at time t is:

$$M^k{}_{r,t+\delta t} = M^k{}_{r,t} - k_r \omega_{rel} \Delta t \tag{692}$$

where $\left|M^k{}_{r,t}\right| \leq k_r \theta_r{}^{max}$ and $k_r = \left(\mu_r R_{eq} F_n\right) / \theta_r{}^{max}$, and where:

- $\theta_r{}^{max}$ is the maximum angular displacement before rolling.
- ω_{rel} is the relative rotation, given as $\omega_{rel} = \omega_i - \omega_j$, where:
  - ω_i is the rotation of the particle for which the resistance moment M_r is being calculated.
  - ω_j is the rotation of the other particle in the contact.
- μ_r is the coefficient of rolling friction, also called coefficient of rolling resistance. The default value is 0.1.
- R_{eq} is the particle equivalent radius, that is, the harmonic mean of the radii of the two particles.
- F_n is the magnitude of the contact force.

You can activate the viscous damping torque or leave it off. When active, it is:

$$M_r{}^d = \begin{cases} -\nu C^{\text{crit}} \omega_{\text{rel}} & \text{if } \left(\left|M^k{}_{r,t}\right| \leq k_r \theta_r{}^{max}\right) \\ -f \nu C^{\text{crit}} \omega_{\text{rel}} & \text{if } \left(\left|M^k{}_{r,t}\right| > k_r \theta_r{}^{max}\right) \end{cases} \tag{693}$$

where:

- ν is the damping factor parameter.
- f is a damping factor multiplier
- C^{crit} is the critical viscous damping constant, computed as:

  $C_r{}^{\text{crit}} = 2\sqrt{I_r k_r}$

  where I_r is the equivalent moment of inertia for the relative rotational vibration mode about the contact point.

Discrete Element Method Drag Coefficient

This section describes the drag force coefficients available with the DEM drag force model.

The following drag force coefficients are available with the DEM drag force model:

- Clift
- Di Felice
- Gidaspow
- Haider and Levenspiel

Clift Drag Coefficient Method

This method is available only with cylindrical particles. The drag coefficient C_d depends on the particle Reynolds number and the aspect ratio of the cylinder. The Clift drag coefficient is given as:

- For long cylindrical particles (aspect ratio ≥ 1):

$$\begin{aligned}
\mathrm{Re}_p \le 5&: \ C_d = 9.689\ \mathrm{Re}_p{}^{-0.78}\left(1 + 0.147\ \mathrm{Re}_p{}^{0.82}\right) \\
5 < \mathrm{Re}_p \le 40&: \ C_d = 9.689\ \mathrm{Re}_p{}^{-0.78}\left(1 + 0.227\ \mathrm{Re}_p{}^{0.55}\right) \\
40 < \mathrm{Re}_p \le 400&: \ C_d = 9.689\ \mathrm{Re}_p{}^{-0.78}\left(1 + 0.0.838\ \mathrm{Re}_p{}^{0.82}\right) \\
400 < \mathrm{Re}_p&: \ C_d = 0.99\ \gamma^{-.012} E^{-0.08}
\end{aligned} \tag{694}$$

where $\gamma = \rho_p / \rho_f$ is the ratio of particle density to fluid density and $E = h/d$ is the aspect ratio.

- For flattened cylindrical particles (aspect ratio < 1):

$$\begin{aligned}
\mathrm{Re}_p \le 0.01&: \ C_d = \frac{64}{\pi\ \mathrm{Re}_p}\left(1 + \frac{\mathrm{Re}_p}{2\pi}\right) \\
0.01 < \mathrm{Re}_p \le 1.5&: \ C_d = \frac{64}{\pi\ \mathrm{Re}_p}\left(1 + 10^{x}\right) \\
\left(x = -0.883 + 0.906\ \log\left(\mathrm{Re}_p\right) - 0.025\left(\log \mathrm{Re}_p\right)^2\right)& \\
1.5 < \mathrm{Re}_p \le 133&: \ C_d = \frac{64}{\pi\ \mathrm{Re}_p}\left(1 + 0.138\ \mathrm{Re}_p^{0.792}\right) \\
133 < \mathrm{Re}_p&: \ C_d = 1.17
\end{aligned} \tag{695}$$

Di Felice Drag Coefficient Method

This drag method introduces an extra term in the fluid drag force expression (Eqn. (2973)) to simulate the effect of the presence of other particles around a particle. The Di Felice drag coefficient is given as:

$$C_d = \left[0.63 + \frac{4.80}{\sqrt{\mathrm{Re}_p}}\right]^2 \varepsilon_i^{-\xi} \tag{696}$$

where Re_p defines the particle Reynolds Number, ε_i is void fraction around a particle, and:

$$\xi = 3.7 - 0.65 \exp\left[-0.5\left(1.5 - \log_{10} \mathrm{Re}_p\right)^2\right] \tag{697}$$

The term $\varepsilon_i^{-\xi}$ represents the effect of enhanced drag on a particle, due to the presence of other particles around it.

Gidaspow Drag Coefficient Method

STAR-CCM+ incorporates the equations in drag coefficient form as follows:

$$C_d = \frac{4}{3}\left(150 \frac{1 - \nu_f}{\nu_f \mathrm{Re}_p} + 1.75\right) \qquad \text{if } \nu_f < \nu_{min} \tag{698}$$

Otherwise:

$$C_d = \frac{\left(24 + 3.6 \cdot \mathrm{Re}_p^{0.687}\right)}{\mathrm{Re}_p} \nu_f^{-3.65} \tag{699}$$

where ν_f is the void fraction, ν_{min} is the cutoff void fraction (user-defined, with a default setting of 0.8), and Re_p is the particle Reynolds number. The exponent of -3.65 can also be user-defined in the *Exponent* property of the **Gidaspow** node.

Haider and Levenspiel Drag Coefficient Method

The drag coefficient C_d depends on the particle Reynolds number and the particle sphericity:

$$C_d = \frac{24}{\mathrm{Re}_p}\left(1 + A\ \mathrm{Re}_p^B\right) + \frac{C}{\left(1 + \frac{D}{\mathrm{Re}_p}\right)} \tag{700}$$

and:

$$\begin{aligned} A &= 8.1716e^{-4.0665\phi} \\ B &= 0.0964 + 0.5565\phi \\ C &= 73.690e^{-5.0746\phi} \\ D &= 5.3780e^{6.2122\phi} \end{aligned} \tag{701}$$

where ϕ is the particle sphericity and Re_p is the particle Reynolds number.

Drag Torque Model

Drag torque reduces the difference in the rotational velocities between a particle and the fluid in which it is immersed.

The drag is a torque $\mathbf{T}$ applied to a DEM particle (Sommerfeld [119]):

$$\mathbf{T} = \frac{\rho}{2}\left(\frac{D}{2}\right)^5 C_R|\mathbf{\Omega}|\mathbf{\Omega} \tag{702}$$

where:

- ρ is the fluid density
- D is the particle diameter.
- C_R is the rotational drag coefficient.
- $\mathbf{\Omega}$ is the relative angular velocity of the particle to the fluid, or slip-rotation, given by:

$$\mathbf{\Omega} = \frac{1}{2}\nabla \times \mathbf{u} - \boldsymbol{\omega}_p \tag{703}$$

  where:

  $\mathbf{u}$ is the fluid velocity

  $\boldsymbol{\omega}_p$ is the angular velocity of the particle.

The rotational drag coefficient C_R used here is defined as [119]:

$$C_R = \begin{cases} \dfrac{12.9}{\mathrm{Re}_R{}^{0.5}} + \dfrac{128.4}{\mathrm{Re}_R}, & 32 \le \mathrm{Re}_R < 1000 \\ \dfrac{64\pi}{\mathrm{Re}_R}, & \mathrm{Re}_R < 32 \end{cases} \tag{704}$$

Re_R is the rotational Reynolds number, defined as:

$$\mathrm{Re}_R = \frac{\rho D^2 |\mathbf{\Omega}|}{\mu} \tag{705}$$

where μ is the dynamic viscosity.

Although Sommerfeld restricts the rotational Reynolds number to values below 1000, the model places no upper limit on it.

Discrete Element Method Lift Forces

Lift forces in DEM simulations can arise from particle spin, particle shear, or both. In this context, lift forces are taken to mean forces normal to the particle velocity; they are not necessarily forces in the upward direction.

Particle Spin Lift Force

This force applies to a spinning particle moving relative to a fluid. The force is given by:

$$\mathbf{F}_{LR} = \frac{\rho\pi}{8} D^2 C_{LR} |\mathbf{u}_s| \frac{\mathbf{\Omega} \times \mathbf{u}_s}{|\mathbf{\Omega}|} \tag{706}$$

where:

- ρ is the fluid density.
- D is the particle diameter.
- C_{LR} is the coefficient of rotational lift.
- $\mathbf{u}_S$ is the fluid velocity relative to the moving particle, known as the slip velocity.
- $\mathbf{\Omega}$ is the relative angular velocity of the particle to the fluid, known as the slip rotation.

Sommerfeld [119] gives the coefficient of rotational lift C_{LR} as:

$$C_{LR} = 0.45 + \left(\frac{\mathrm{Re}_R}{\mathrm{Re}_p} - 0.45\right) e^{-0.5684 \mathrm{Re}_R{}^{0.4} \mathrm{Re}_p{}^{0.3}} \tag{707}$$

where:

- Re_R is the rotational Reynolds number.
- Re_p is the particle Reynolds number.

Sommerfeld reports this formula for C_{LR} as valid for particle Reynolds numbers below 140.

You can also define C_{LR} for yourself as a field function.

The rotational Reynolds number Re_R is given as:

$$\mathrm{Re}_R = \frac{\rho D^2 |\mathbf{\Omega}|}{\mu} \tag{708}$$

where:

- ρ is the fluid density.
- D is the particle diameter.
- $\mathbf{\Omega}$ is the relative angular velocity of the particle to the fluid, known as the slip rotation.
- μ is the dynamic viscosity of the fluid.

The slip rotation $\mathbf{\Omega}$ is given by:

$$\mathbf{\Omega} = \frac{1}{2}\nabla \times \mathbf{u} - \boldsymbol{\omega}_p \tag{709}$$

where:

- $\mathbf{u}$ is the fluid velocity.
- $\boldsymbol{\omega}_p$ is the angular velocity of the particle.

Particle Shear Lift Force

This force applies to a particle moving relative to a fluid where there is a velocity gradient in the fluid orthogonal to the relative motion.

Saffman [118] gives the shear lift force as:

$$\mathbf{F}_{LS} = 1.615 D^2 (\rho\mu)^{0.5} \left| \frac{\delta \mathbf{u}}{\delta y} \right| \mathbf{u}_S \tag{710}$$

where:

- D is the particle diameter.
- ρ is the fluid density.
- μ is the dynamic viscosity of the fluid.
- $\mathbf{u}$ is the fluid velocity.
- $\mathbf{u}_S$ is the fluid velocity relative to the moving particle, known as the slip velocity.
- The y direction is the direction of the velocity gradient.

The three-dimensional version of this equation is:

$$\mathbf{F}_{LS} = C_{LS} \frac{\rho\pi}{8} D^3 (\mathbf{u}_S \times \boldsymbol{\omega}) \tag{711}$$

where:

- $\boldsymbol{\omega} = \nabla \times \mathbf{u}$, the curl of the fluid velocity.
- C_{LS} is the shear lift coefficient.

You can define C_{LS} as a field function or use either of two published definitions:

- Saffman [118] provides a definition that recovers the original Saffman asymptotic solution for low Reynolds numbers:

$$C_{LS} = \frac{4.1126}{\mathrm{Re}_S^{0.5}} \tag{712}$$

- Sommerfeld [119] provides a definition for a broader range of Reynolds numbers:

$$C_{LS} = \frac{4.1126}{\mathrm{Re}_S^{0.5}} f\mathrm{Re}_p, \mathrm{Re}_S \tag{713}$$

where:

$$f\mathrm{Re}_p, \mathrm{Re}_S = \begin{cases} \left(1-0.3314\beta^{0.5}\right)e^{-0.1\mathrm{Re}_p} + 0.3314\beta^{0.5}, & \left(\mathrm{Re}_p \le 40\right) \\ 0.0524\left(\beta\mathrm{Re}_p\right)^{0.5}, & \left(\mathrm{Re}_p > 40\right) \end{cases} \tag{714}$$

and

$$\beta = 0.5\frac{\mathrm{Re}_S}{\mathrm{Re}_p} \tag{715}$$

where:

Re_S is the Reynolds number for shear flow, $\mathrm{Re}_S = \frac{\rho D^2|\boldsymbol{\omega}|}{\mu}$.

Re_p is the particle Reynolds number.

Artificial Viscosity Model

The Artificial Viscosity model is used to correct the results in cases where high impact velocities cause the numerical models to over-predict the overlap between the colliding objects. The energy that is dissipated by artificial viscosity represents unresolved processes of damage, deformation, and entropy.

The name "Artificial Viscosity" refers only to the similarity between the formulation of the Artificial Viscosity model and the formulations for viscosity. It does not refer to any physical effect resembling real viscosity.

This model introduces a force during compression (while the particles are overlapping and their centers are moving nearer) based on the von Neumann-Richtmyer artificial viscosity [123]:

$$F_v = -f\left(Q_2|v_n| - Q_1 c\right)\frac{m_{eq}}{R_{eq}}v_n \tag{716}$$

where:

- F_v is the artificial viscosity force.
- v_n is the component of the velocity in the contact normal direction.
- m_{eq} is the equivalent (harmonic mean) mass of the particles.
- R_{eq} is the equivalent (harmonic mean) radius of the particles.
- c is the speed of sound in the material. It is estimated from the equivalent mass and radius, and from the contact stiffness for the specific contact model in use.
- Q_1 is the linear coefficient.

- Q_2 is the quadratic coefficient.

The factor f is a linear ramp function that is scaled to the particle sizes as:

$$f = \begin{cases} 0, & r/R_{eq} < p_{\text{lower}} \\ \frac{r/R_{eq} - p_{\text{lower}}}{p_{\text{upper}} - p_{\text{lower}}}, & p_{\text{upper}} \geq r/R_{eq} \geq p_{\text{lower}} \\ 1, & r/R_{eq} > p_{\text{upper}} \end{cases} \tag{717}$$

where:

- r is the penetration, the overlap distance between particles or between particle and wall.
- p_{upper} and p_{lower} are the upper and lower ramp limits. They are defined in terms of the ratio of the penetration r to the equivalent particle radius R_{eq}.

This ramp function f allows you to use artificial viscosity without affecting typical low penetration contacts.

Impact Heat Model

The Impact Heat model is a widely used approximate model. It calculates the rate of heat production that results from friction and damping in DEM particles, using a linear formulation or a customized field function. This section describes the linear formulation.

The rate of heat generation is:

$$q_r = c_t F_t v_t + c_n F_n v_n \tag{718}$$

where:

- c_t and c_n are, respectively, the fractions of frictional work and damping work that are converted to heat.
- F_t and F_n are, respectively, the frictional (tangential) forces and damping (normal) forces on the particle.
- v_t and v_n are, respectively, the relative tangential and normal impact velocities.

The values for c_t and c_n must be estimated for the individual setup being simulated. The other factors are derived from other inputs in the DEM models. For examples, see Iwasaki [109] and Rojek [reflink].

Passive Scalar Transfer Formulation

The Passive Scalar Transfer model describes how a passive scalar rubs off one DEM particle onto another. It is primarily intended for describing how a coating of negligible mass is shared among colliding particles.

The Area Weighted and Volume Weighted transfer methods are both described by this equation:

$$\dot{m} = k\left(C_1 \rho_1 n_1 - C_2 \rho_2 n_2\right) A_c \tag{719}$$

where:

- $\dot{m}$ is the flow of passive scalar between phases 1 and 2, regarded as analogous to a mass flux.
- k is the value of the user-set property *Coefficient* in the **Area Weighted** or **Volume Weighted** node.

- C_1 and C_2 are the values of the passive scalar on phase 1 and phase 2.
- ρ_1 and ρ_2 are the particle densities of phase 1 and phase 2.
- n_1 and n_2 are the parcel particle counts for phase 1 and phase 2.
- A_c is the area that is computed from the harmonic mean of the radii of the two particles.

In Volume Weighted transfer, the flow is divided by the volume of the respective particles, while in Area Weighted transfer, the flow is divided by the surface areas. Compare these methods to the Passive Scalar Interactions under Lagrangian Multiphase Flow theory.

In the Increment transfer method, a fixed amount of passive scalar (the increment, which is given by *Coefficient*) is added to each particle every time they make contact. This method does not model transfer of any physical property, but can be used in cases such as counting the number of contacts a particle undergoes.

Discrete Element Method Bibliography

[98] *Archard, J. F., and Hirst, W. 1956. "The wear of metals under unlubricated conditions", Proceedings of the Royals Society of London. Series A, Mathematical and Physical Sciences. 236(1206), pp 397-410.*

[99] *Butt, H-J., Graf, K. and Kappl, M. 2006. Physics and Chemistry of Interfaces, Wiley-VCH.*

[100] *Camacho, G.T. and M. Ortiz, 1996, "Computation Modelling of Impact Damage in Brittle Materials," Int'l J. Solids Structures, 33(20-22) pp 2899-2938.*

[101] *Cheng, G. J., Yu, A. B. and Zulli, P. 1999. "Evaluation of effective thermal conductivity from the structure of packed bed", Chemical Engineering Science, 54, pp. 4199-4209.*

[102] *Clift, R., Grace, J. R., Weber, M. E. 1978. Bubbles, Drops and Particles, Academic Press, New York, chapters 4 & 6.*

[103] *Cundall, P. A., Strack, O. D. L. 1979. "A discrete numerical model for granular assemblies", Geotechnique, 29, pp. 47-65.*

[104] *Di Felice, R. 1994. "The voidage function for fluid-particle interaction systems", Int. J. Multiphase Flow, 20, pp. 153-159.*

[105] *Di Renzo, Alberto and Di Maio, Francesco Paolo. 2004. "Comparison of contact-force models for the simulation of collisions in DEM-based granular flow codes", Chemical Engineering Science 59 pp. 525-541.*

[106] *Gidaspow, D. 1994. Multiphase Flow and Fluidization-Continuum and Kinetic Theory Descriptions, Academic Press.*

[107] *Haider, A. and Levenspiel, O. 1989. "Drag coefficient and terminal velocity of spherical and non-spherical particles", Powder Technology, 77, pp. 143-152.*

[108] *Han, K., Feng, Y. T., and Owen, D. R. J, 2006. "Performance comparisons of tree-based and cell-based contact detection algorithms," Engineering Computations: International Journal for Computer-Aided Engineering and Software, 24:2, 165-181.*

[109] *Iwasaki, T., Munetake, S., Takaomi, K.. 2001. "Analysis of collision energy of bead media in a high-speed elliptical rotor-type poweder mixer using the discrete element method," Powder Technology 121 pp. 239-248.*

[110] *Johnson, K.L. 1987. Contact Mechanics, Cambridge University Press, Cambridge.*

[111] *Kosteski, L., R.B. D'Ambra and I. Iturrioz, 2012. "Crack propagation in elastic solids using the truss-like discrete element method," Int. J. Fract., 174, pp. 139-161.*

[112] *Perkins, Eric, and Williams, John, 2001. "CGrid: Neighbor Searching for Many Body Simulation", ICADD-4, pp 427-438.*

[113] *Pham, Chi Vinh, Ogden, R.W. 2004. “On formulas for the Rayleigh wave speed”, Wave Motion, 39 pp. 191-197.*

[114] *Pichugin, A.V. 2008. “Approximation of the Rayleigh wave speed”.*

[115] *Potyondy, D.O, and Cundall, P.A. 2004. “A bonded-particle model for rock”, Int. J. Rock Mechanics & Mining Sciences 41 pp. 1329–1364.*

[116] *Rayleigh, L. 1885. “On waves propagated along the plane surface of an elastic solid”, Proc. Lond. Math. Soc. 17.*

[117] *Rojek, J. 2014. "Discrete element thermomechanical modelling of rock cutting with valuation of tool wear," Comp. Part. Mech. 1:71-84.*

[118] *Saffman, P.G. 1965. "The lift on a small sphere in a slow shear flow," J. Fluid Mech. 22 385-400. (See also Saffman, P.G. 1968. Corrigendum to: “The lift on a small sphere in a slow shear flow,“ J. Fluid Mech. 31 624.)*

[119] *Sommerfeld, M. 2000. “Theoretical and Experimental Modelling of Particulate Flows,” Technical Report Lecture Series 2000-06, von Karman Institute for Fluid Dynamics pp 20-23.*

[120] *Timoshenko, S. 1951. Theory of Elasticity, McGraw-Hill Book Company.*

[121] *Trent, B.C., L.G. Margolin, P.A. Cundall, and Gaffney, E.D., 1986. "The Micro-Mechanics of Cemented Granular Material," Second Int'l. Conf. On Constitutive Laws for Engineering Materials, Tucson, Arizona.*

[122] *Tsuji, Y., Tanaka, T., Ishida, T. 1992. “Lagrangian numerical simulation of plug flow of cohesionless particles in horizontal pipe.” Powder Technology 71:3 pp. 239-250.*

[123] *Von Neumann, J., and Richtmyer, R. D. 1950. "A method for the numerical calculation of hydrodynamic shocks", J. Appl. Phys. 21, 232–237.*

[124] *Walton, O.R. 1993. "Numerical simulation of inclined chute flows of monodisperse inelastic, frictional spheres", Mech. Mater. 16 239–247.*

[125] *Walton, O.R. 1993. "Numerical simulation of inelastic, frictional particle-particle interactions", in Particulate Two-Phase Flow, M.C. Roco, Ed., Butterworth–Heinemann, Stoneham, MA, pp. 884–911.*

[126] *Walton, O.R. and Braun, R.L. 1986. "Viscosity, granular-temperature, and stress calculations for shearing assemblies of inelastic, frictional disks", J. Rheology 30:5, pp. 949–980.*

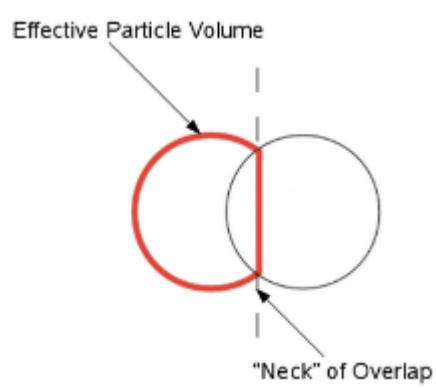
Discrete Element Method Field Functions Reference

This section describes the primitive field functions that are available in the simulation when using the DEM model.

These objects all have the same base properties.

Abrasive Wear Rate	The rate at which material is being eroded from a boundary by abrasion. To access this field function, select the Abrasive Wear model.
Bond Damage	The damage to a bond, on a scale from 0 (undamaged) to 1 (broken). See k_r in Eqn. (3237). To access this field function, select the Simple Failure or Constant Rate Damage model.
Bond Shear Strength	The shear strength of the bond, specified for each bonded contact, based on values assigned in the Simple Failure model. See $\sigma_{\max}$ in Eqn. (3232).
Bond State	Indicates whether any given particle contact is bonded (value 1) or not (value 0). To access this field function, select the Bonded Particles or Parallel Bonds model.
Bond Tensile Strength	The tensile strength of the bond, specified for each bonded contact, based on values assigned in the Simple Failure or Constant Rate Damage models. See $\delta_{\max}$ in Eqn. (3231).
Bonded Component Size	The number of spheres in the bonded particle cluster to which a given particle belongs. Compare with the **Particle Connected Component Size** field function.
Bonded Components	The labels (integers) of bonded particle clusters. These clusters are groups of particles that form a single connected network through particle-particle bonds. Compare with the **Particle Connected Components** field function.

Contact Centroid	The coordinates of the contact point between colliding particles.
Contact Contact Mass 0|1	The mass of one of the individual component particles (0 or 1) making contact when composite particles or particle clumps collide.
Contact Contact Radius 0|1	The radius of one of the individual component particles (0 or 1) making contact when composite particles or particle clumps collide.
Contact Damping Force	The non-elastic component of the contact force between two particles, that is, the contact force minus the elastic component. To access this field function, select the DEM Particles, DEM Phase Interaction, and either the Hertz Mindlin or Walton Braun models.
Contact Elastic Force	The elastic component of the contact force between two particles, that is, the contact force minus the damping component. To access this field function, select the DEM Particles, DEM Phase Interaction, and either the Hertz Mindlin or Walton Braun models.
Contact Force	The total contact force between two particles, that is, the damping force plus the elastic force. See $\mathbf{F}_c$ in Eqn. (2970).
Contact Normal	A unit length vector perpendicular (orthogonal) to the contact plane. To access this field function, select the DEM Particles, DEM Phase Interaction, and either the Hertz Mindlin or Walton Braun models.
Contact Normal Overlap	The contact normal vector that is multiplied by the overlap, giving the direction and magnitude of the particle overlap, normal to the contact plane. To access this field function, select the DEM Particles, DEM Phase Interaction, and either the Hertz Mindlin or Walton Braun models.
Contact Overlap	The amount of overlap between two particles or a particle and wall boundary upon contact. This overlap occurs as the particles are modeled using soft-particle formulation, which permits a small overlap. To access this field function, select the DEM Particles, DEM Phase Interaction, and either the Hertz Mindlin or Walton Braun models.
Contact Physics Radius 0|1	The radius of one of the two particles (0 or 1) in a collision.
Contact Poisson Ratio 0|1	The Poisson ratio for one of the two particles (0 or 1) in a collision.

Contact Relative Velocity	The relative overlap between two contacting particles. To access this field function, select the DEM Particles, DEM Phase Interaction, and either the Hertz Mindlin or Walton Braun models.
Contact Surface Velocity 0|1	The surface velocity vector $\mathbf{v}_s$ at the point of contact for one of the two particles (0 or 1) in a collision. $\mathbf{v}_s = \mathbf{v}_p + \boldsymbol{\omega}_p \times \left(\mathbf{x}_c - \mathbf{x}_p\right)$, where $\mathbf{v}_p$ and $\boldsymbol{\omega}_p$ and the linear and angular velocities of the particle, $\mathbf{x}_c$ is the position of the contact, and $\mathbf{x}_p$ is the position of the particle.
Contact Tangential Overlap	The magnitude and direction of the particle overlap, tangential to the contact. To access this field function, select the DEM Particles, DEM Phase Interaction, and either the Hertz Mindlin or Walton Braun models.
Contact Torque	Moment $\mathbf{M}_c$ in Eqn. (2972).
Contact Youngs Modulus 0|1	The Young's modulus for one of the two particles (0 or 1) in a collision.
Compression Force	Sum of values for $\mathbf{F}_c$.
Cylinder Height	The height of a cylindrical DEM particle. To access this field function, select the Cylindrical Particles model.
Cylinder Radius	The radius of a cylindrical DEM particle. To access this field function, select the Cylindrical Particles model.
Effective Particle Volume	The volume of a particle minus the overlapping volume beyond the plane(s) of intersection with any other particle(s). 
Erosion Rate	The sum of **Abrasive Wear Rate** and **Impact Wear Rate**. It is the quantity E_f in Eqn. (3245).

Impact Wear Rate	The rate at which material is being eroded from a boundary by direct impact. To access this field function, select the Impact Wear model.
Normal Bond Stress	The normal stress in a bond, used in damage and failure modeling. See $\delta_{\max}$ in Eqn. (3231). To access this field function, select the Bonded Particles or Parallel Bonds model.
Parcel DEM Timestep	See Eqn. (3186) through Eqn. (3188).
Parcel Halo Count	Used in debugging.
Parcel Halo Marker	Used in debugging.
Parcel Orientation	The orientation vector of the parcel, expressed as Euler angles.
Particle Angular Momentum	$\mathbf{I}_p\omega_p$. See Eqn. (2972).
Particle Angular Velocity	ω_p in Eqn. (2972).
Particle Connected Component Size	The number of spheres in the particle cluster to which the given particle belongs. Compare with the **Bonded Component Size** field function.
Particle Connected Components	The labels (integers) of particle clusters. These clusters are groups of particles that form a single connected network through particle-particle contact. Compare with the **Bonded Components** field function.
Particle Flow Rate	The rate at which particles flow along parcel tracks, $\dot{n}_\pi$. (Steady)
Particle Moment Inertia	$\mathbf{I}_p$ in Eqn. (2971). This quantity can be used for validation.
Particle Residence Time	The “age” of particles as measured from time of injection.
Particle Rotation Kinetic Energy	Allows post-processing of the kinetic energy of the rotating particle.
Particle Volume Overlap	How much of the particle is obscured by the contact area.
Tangential Bond Stress	The shear stress in the contact plane of a bond, used in damage and failure modeling. See $\sigma_{\max}$ in Eqn. (3232). To access this field function, select the Bonded Particles or Parallel Bonds model.
Void Fraction	The fraction of the cell volume available to the fluid, unoccupied by a solid or dispersed phase.

Migrating from EDEM

This section provides information to help you convert an EDEM simulation to run under STAR-CCM+.

To convert an EDEM simulation to run under STAR-CCM+, remember:

- Plots of time-dependent quantities should be set before simulation in STAR-CCM+ and after simulation in EDEM.
- Geometry meshing requirements are different in STAR-CCM+ compared to EDEM (concepts of volume mesh, regions).
- Cell size should be at least four times particle size, and as much larger as practical.
- Optimal number of CPUs: STAR-CCM+ readily scales up to at least 64 CPUs. These CPUs can be distributed through a cluster.
- The Hertz-Mindlin model in STAR-CCM+ uses two coefficients of restitution, one for energy dissipation due to damping of normal component of contact force, another for energy dissipation due to damping of tangential component of contact force. The Hertz-Mindlin model in EDEM uses only one parameter for coefficient of restitution.
- Differences between EDEM and STAR-CCM+ terminology include the following:

	EDEM	STAR-CCM+
	Write-out interval	Auto-save time
	Factory	Injector
	Rayleigh timestep	Time scaling factor
	Simulated time	Time stopping criteria parameter

Post-Processing DEM Data

STAR-CCM+ allows you to analyze the flow of DEM phases over boundaries.

Since DEM phases are a form of Lagrangian phases, Lagrangian post-processing methods also apply to them. See Post-Processing Lagrangian Data.

Cylindrical Particles in Scalar Scenes

When creating scenes for simulations using cylindrical particles, set the **Displayers** > **Scalar 1** property *Point Scale Mode* to **Particle Model**. With this setting, the scene takes size information directly from the particle model.

Direct Data on Boundaries

The following field functions are available on boundaries:

- Incident mass flux: computes the instantaneous mass flux of one DEM phase escaping through a boundary. This field function is valid only for boundaries not using DEM mode.
- Abrasive Wear Rate: Computes the estimated loss of mass material per second due to abrasion. This field function is available only if the Abrasive Erosion model is activated. See Modeling Erosion in the Discrete Element Method.

- Impact Wear Rate: Computes the estimated loss of mass material per second due to particle impact. This field function is available only if the Impact Wear Rate model is activated. See Modeling Erosion in the Discrete Element Method.
- Erosion Rate: The sum of Abrasive Wear Rate and Impact Wear Rate. This field function is available only if the Abrasive or Impact Wear Rate models are activated.

See Discrete Element Method Field Functions Reference.

Simulation History Phase Data

Use simulation history files to save scalar and vector data on selected DEM phases. See Configuring the Simulation History File.

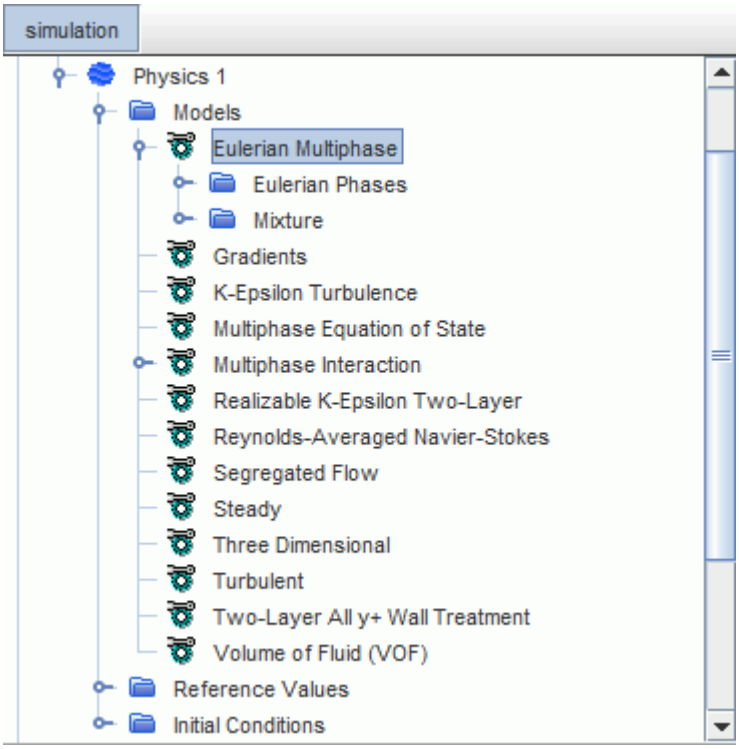
By default, simulation history files record data from the following field functions:

- When the Cylindrical Particles model is active:
  - **Cylinder Height**
  - **Cylinder Radius**
- When the DEM Phase Interaction model is active:
  - **Contact Centroid**

Using the Eulerian Multiphase Model

For those multiphase models that require multiple Eulerian phases, the Eulerian Multiphase model is used to create and manage those phases.

The corresponding node (**Eulerian Multiphase** in the following screenshot), is activated as soon as the Eulerian Multiphase model is selected. Right-click this node and choose **New** from the pop-up menu to create a Eulerian phase.



The Multiphase Segregated Flow, Eulerian Multiphase Mixture, and Volume of Fluid (VOF) models require you to select the Eulerian Multiphase model before they become available. The process of defining new phases is common to all three models, as is the selection of phase materials, viscous regime and phase equations of state. The relevant field functions are made available for each phase, such as Mass Flux, Mass Imbalance, Velocity, and Vorticity.

While the concept of volume fraction is different for these models, how you set the volume fraction for boundaries and initial conditions is the same.

Contents:

Selecting the Eulerian Multiphase Model
Defining Eulerian Phases
Setting Volume Fraction at Flow Boundaries
Modeling the Eulerian Phase Material
Modeling the Eulerian Phase Equation of State

Modeling the Eulerian Phase Viscous Regime
Modeling User-Defined Potential Forces

Selecting the Eulerian Multiphase Model

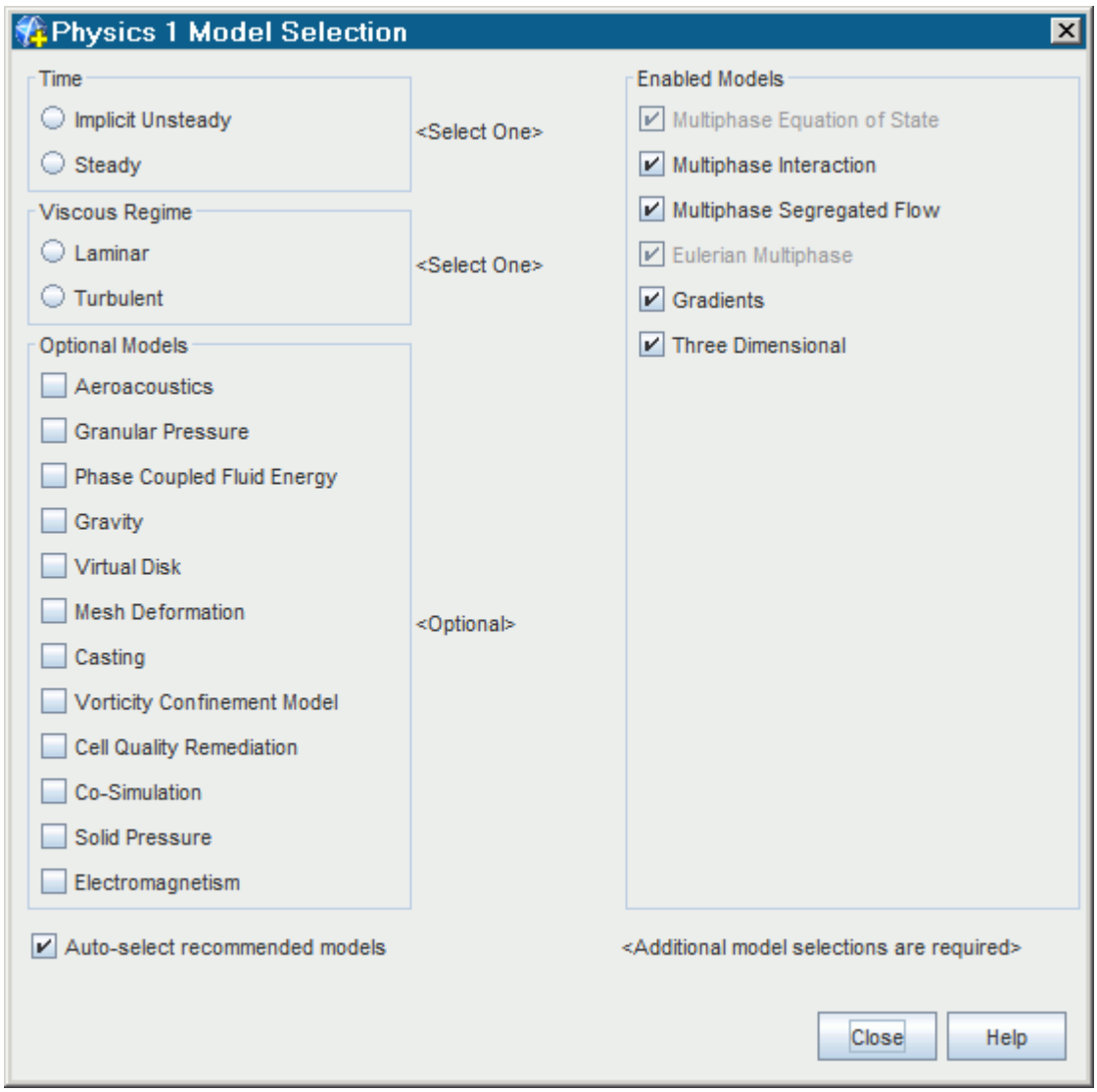
This section outlines the Eulerian Multiphase model selection process.

To solve problems using the Eulerian Multiphase model, first select the **Eulerian Multiphase** material model in the *Physics Model Selection* dialog. The **Multiphase Interaction** model is selected automatically.

1. Choose **Eulerian Multiphase Mixture**, **Multiphase Segregated Flow**, or **Volume of Fluid (VOF)** from the *Eulerian Multiphase Model* group box.

   The **Multiphase Equation of State** model is selected automatically, along with any other related models.

   An example of selections for Multiphase Segregated Flow is shown below.



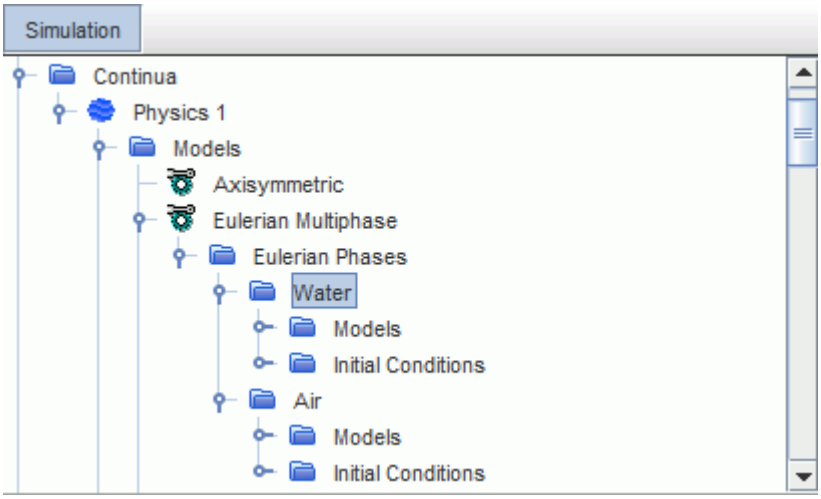
The remaining selections in this dialog depend on problem characteristics that are not related to Eulerian phases.

Defining Eulerian Phases

A Eulerian phase in STAR-CCM+ is a phase that is modeled in a Eulerian framework.

The definition of each Eulerian phase includes the set of models applicable to its material; two materials constitute two different Eulerian phases.

Eulerian phases are stored under the **Eulerian Phases** node, which has a pop-up menu. Each Eulerian phase node contains a **Models** node, on which you select phase models, and an **Initial Conditions** node, on which you set the initial volume fraction and other appropriate initial conditions. As Eulerian phase nodes are subject to common menu items, you can copy and paste the properties of one phase onto another.



Initially, the **Eulerian Phases** node is empty. The process for creating and defining Eulerian phases is outlined below:

1. Create a Eulerian Phase
2. Rename the Phase
3. Select Phase Models
4. Replace the Phase Material
5. Customize Material Properties
6. Define Volume Fraction in Eulerian Multiphase Flows

Repeat Steps 1 - 6 for all phases.

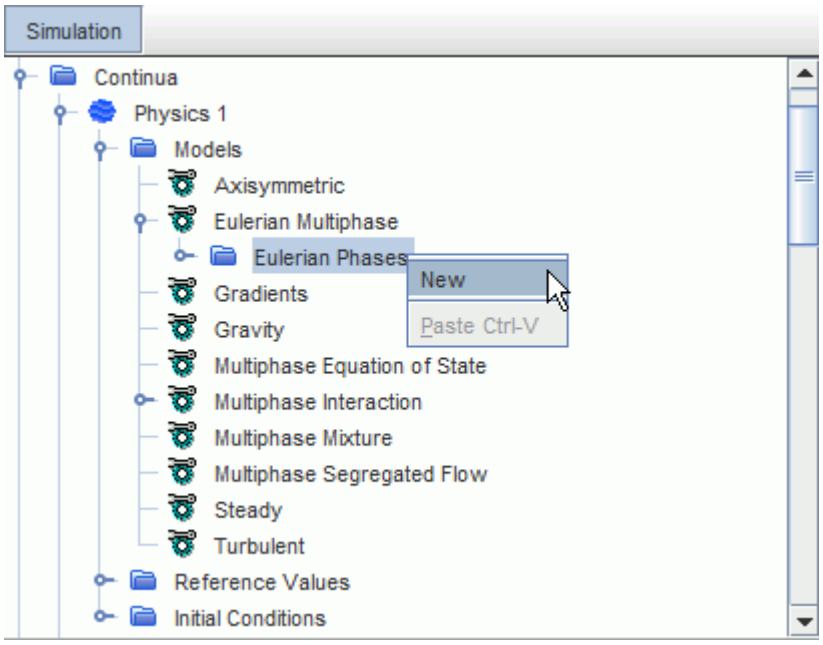
Repeat these steps until all Eulerian phases have been specified. When all Eulerian phases have been defined, define phase interactions for pairs of phases that have significant contact with each other.

Creating a Eulerian Phase

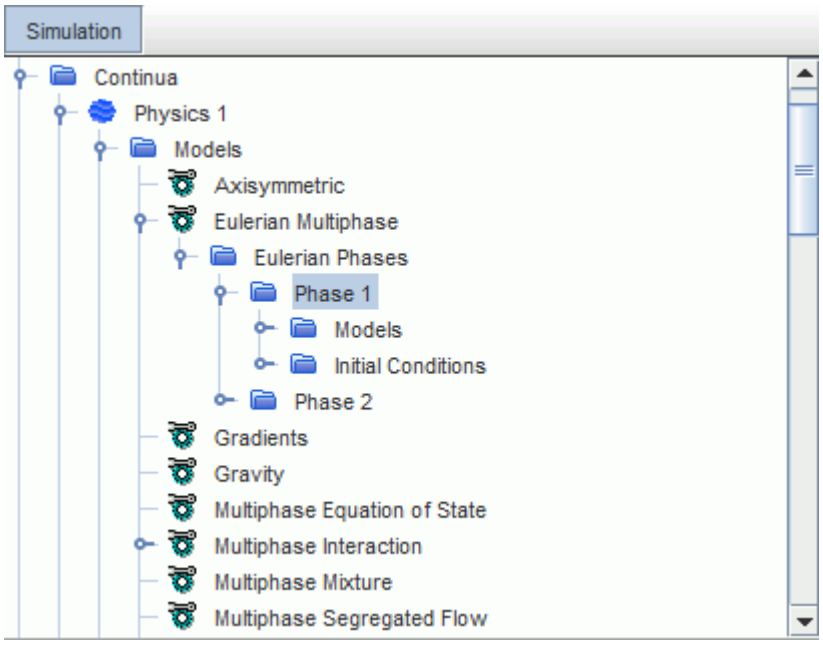
The **Eulerian Phases** node (which is under the **Eulerian Multiphase** model node) is initially unpopulated.

To create a Eulerian phase:

1. Right-click the **Eulerian Phases** node and select **New** from the pop-up menu.



A new Eulerian phase node appears beneath the **Eulerian Phases** node, with its own **Models** and **Initial Conditions** sub-nodes.

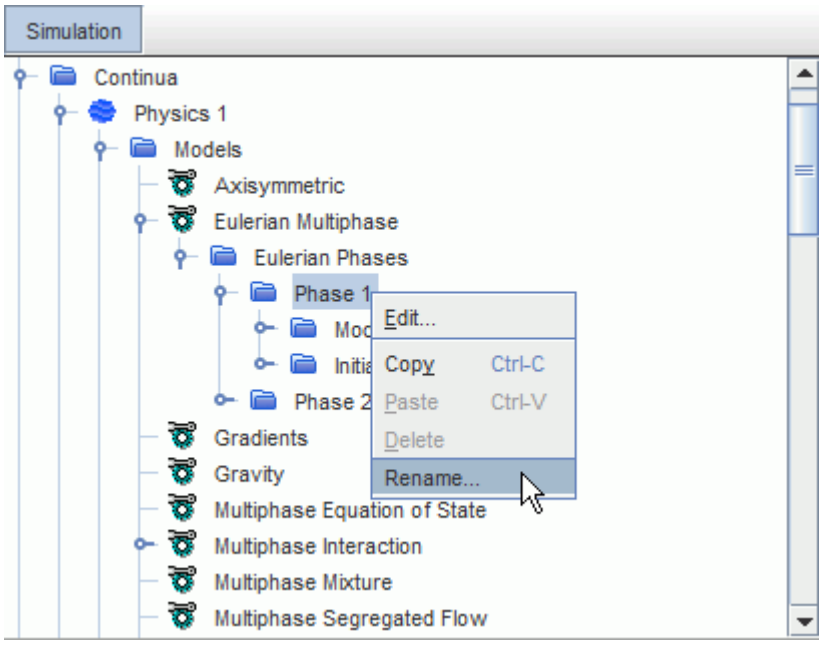


Renaming the Phase

The newly created phase can be renamed if desired.

To rename the phase:

1. Right-click the phase node and select **Rename** as shown below.



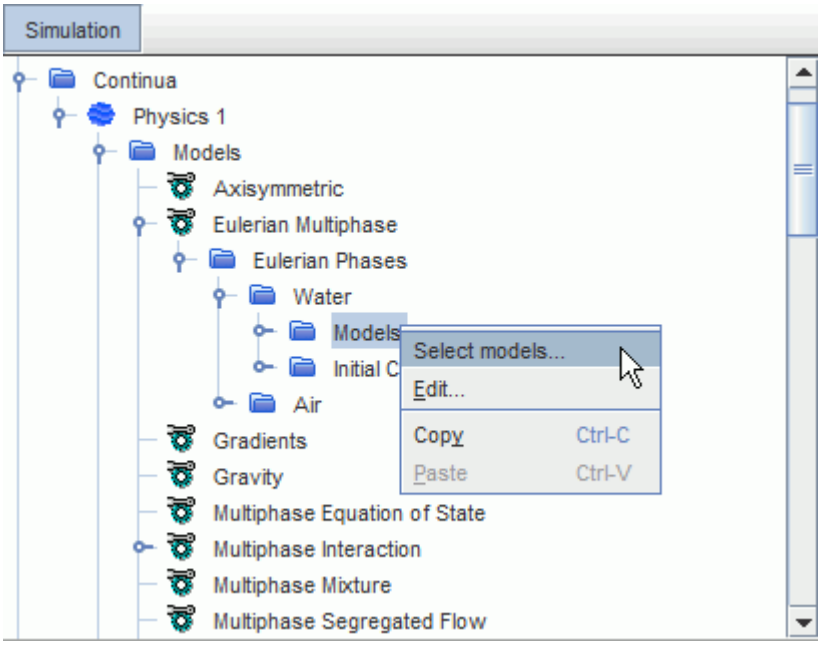
2. Enter the new name into the inline editor and press <Enter>.

Selecting Phase Models

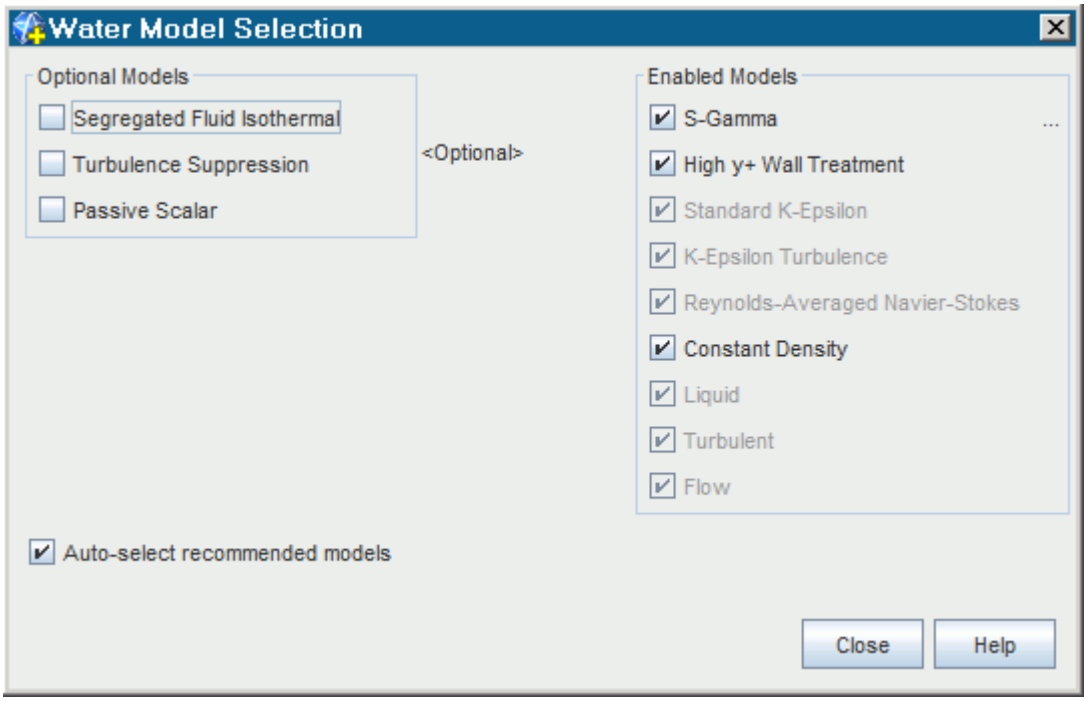
Each Eulerian phase has phase models that are associated with it. A Eulerian phase model plays the same role in a Eulerian phase as a conventional model does in a continuum. It defines how a physical phenomenon in a phase is represented, what the primary variables of the simulation are and what mathematical formulation is used to generate the solution.

Eulerian phase model selection is not identical to model selection for a continuum, since Eulerian phases implicitly inherit certain model choices such as those for space, time, and motion.

Eulerian phase models are stored under the **Models** manager node for the phase. They are selected using the *Phase Model Selection* dialog, which is accessed by right-clicking the **Models** node and choosing **Select models...**

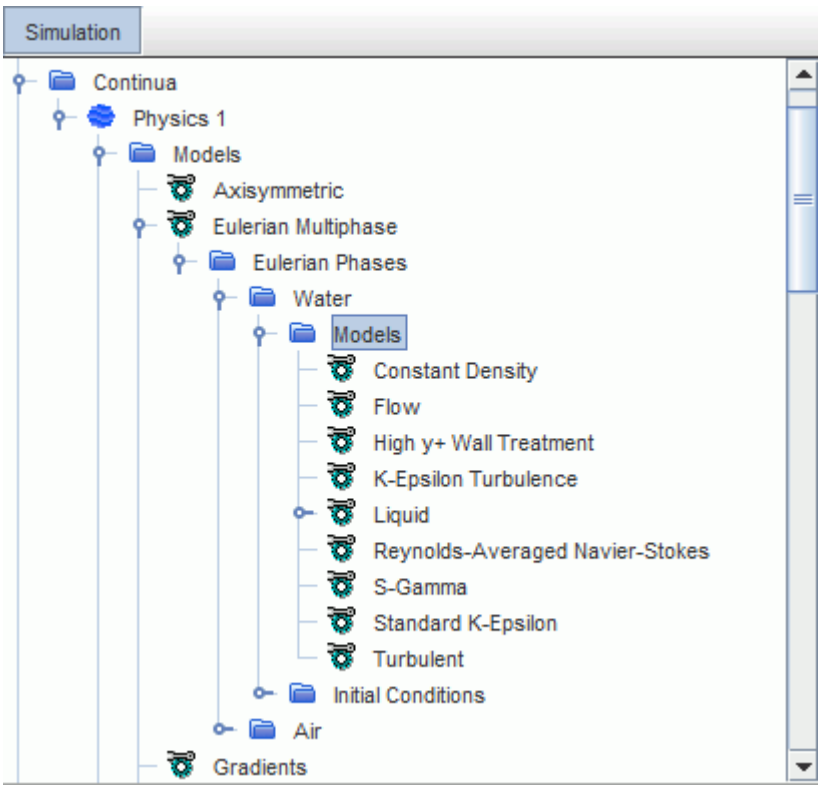


The *Phase Model Selection* dialog is used in the same way as the *Physics Model Selection* dialog. That is, it allows you to select and review phase models, change which models have been selected, and supports auto-selection of recommended models. The choice of material and equation of state models available in the dialog depends on the physics models you selected at the continuum level (either Volume of Fluid (VOF) or Multiphase Segregated Flow). An example *Phase Model Selection* dialog is given below:



Once you have selected the phase models, click **Close**.

Nodes corresponding to the chosen models appear under the **Models** node for the Eulerian phase.

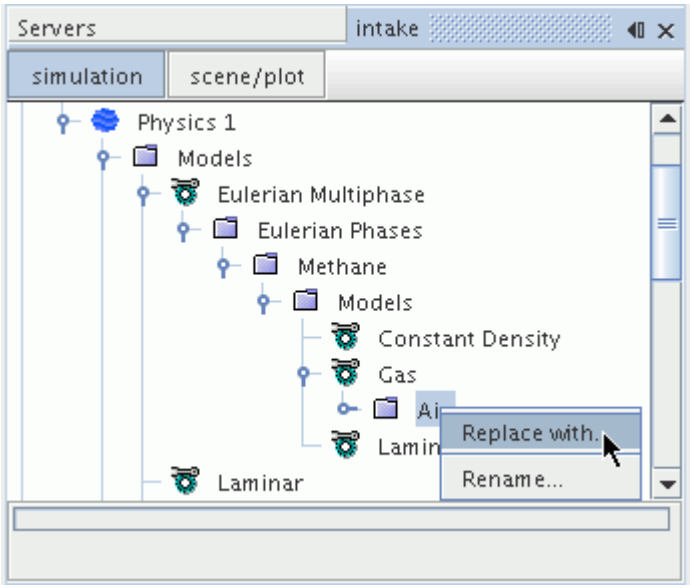


Replacing the Phase Material

The default material model appears within the **Models** node for the phase.

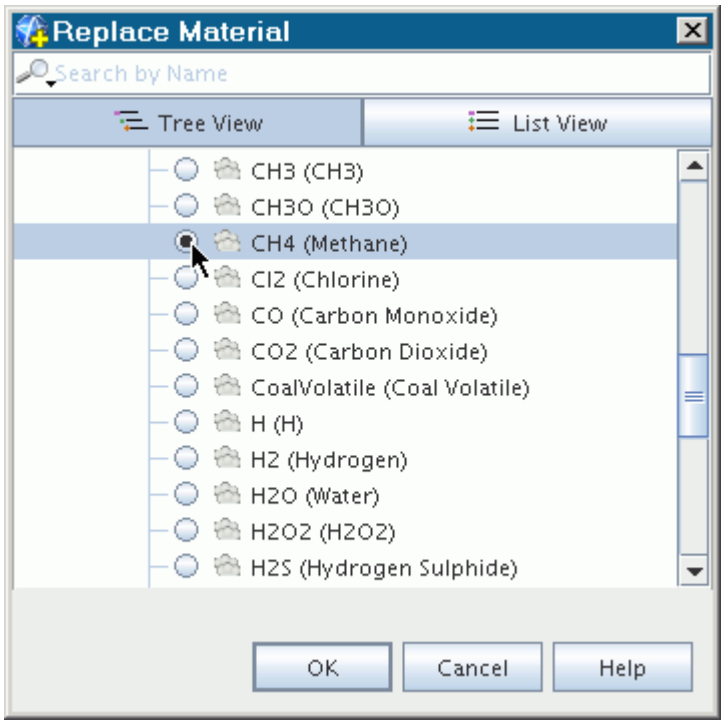
To replace the default material with something more appropriate:

1. Right-click the material node and selecting **Replace with**.



The *Replace Material* dialog appears.

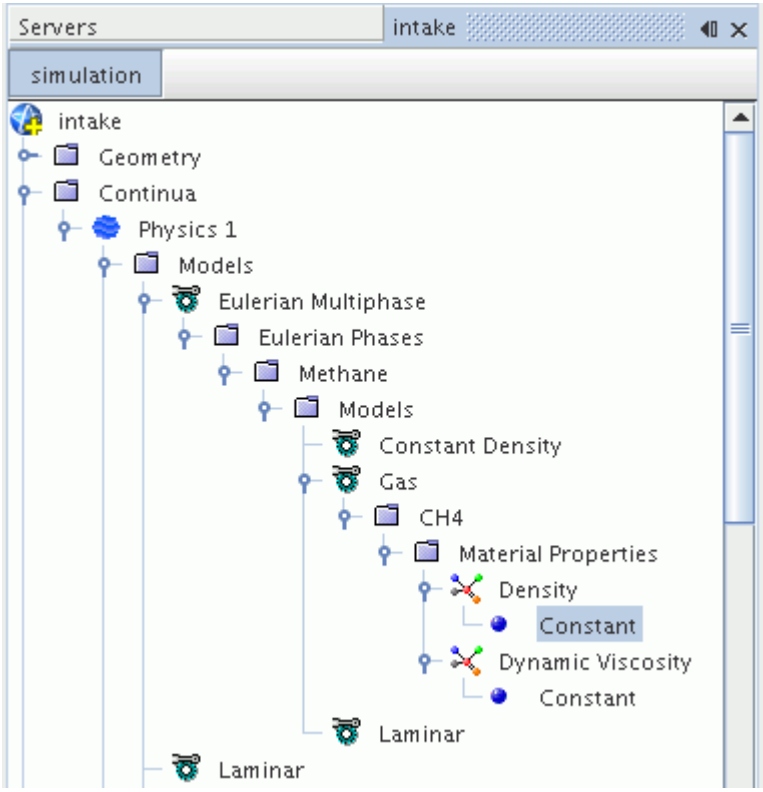
2. Navigate to the required material, select its radio button, and click **OK** to confirm the selection.



Customizing Material Properties

Further changes to material properties can be made by accessing the individual material property nodes within the material node.

An example is shown below:



If the flow is laminar, the dynamic viscosity of a liquid can be set to use the Non-Newtonian Generalized Power Law.

Defining Volume Fraction in Eulerian Multiphase Flows

The volume fraction of a phase is the ratio of the volume that the phase occupies to the computational cell volume. Define the initial and boundary conditions of this scalar before running the simulation. You specify the volume fraction for each phase under the corresponding phase node.

When you specify initial and boundary values, there is no restriction on the values that you can set for the volume fractions of the phases. However, STAR-CCM+ uses normalized values for each phase volume fraction to ensure that the total volume fraction is 1.0.

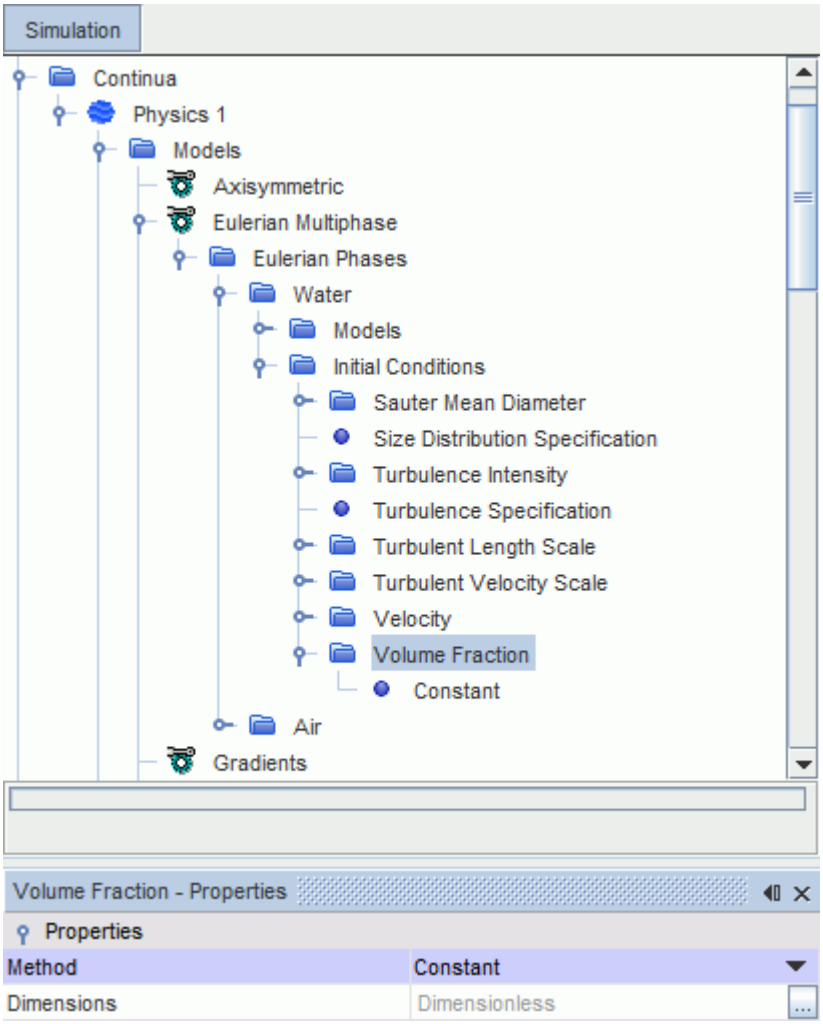
If you are using the Volume of Fluid (VOF) model with only two phases, the volume fraction of the second-defined phase is calculated by subtracting the volume fraction value of the first-defined phase from 1.0. Any value that you specify is ignored.

This section applies to flows using either the Multiphase Segregated Flow model or the Volume of Fluid (VOF) model.

Setting Initial Volume Fraction

Following the phase specification, the **Initial Conditions** node of each phase contains a **Volume Fraction** sub-node.

Use this node to define the initial spatial distribution of the phase.

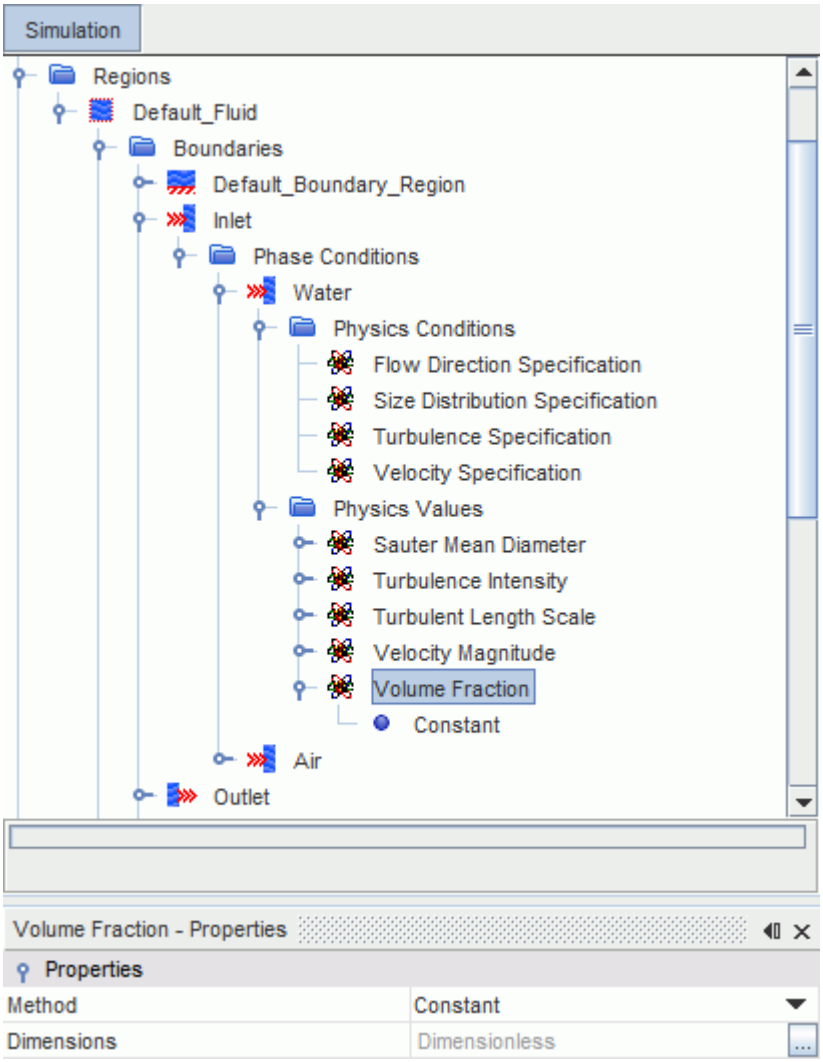


The volume fraction of a phase is entered as a scalar profile.

Setting Volume Fraction at Flow Boundaries

Following the Eulerian phase specification, the node of every open boundary in the simulation contains a **Phase Conditions** node that lets you specify the physics conditions and values for each phase.

For each phase, use the **Physics Values** > **Volume Fraction** node to define the boundary condition on the phase volume fractions. The volume fraction of a phase is entered as a scalar profile. When you define an outlet pressure boundary, you must set a volume fraction to cover the case of backflow. Otherwise, this volume fraction does not influence the solution domain.



A zero-flux condition is imposed at closed (wall) boundaries. Therefore, no user specification is needed there.

Modeling the Eulerian Phase Material

The material of each Eulerian phase is defined by selecting one of the material models in the *Phase Model Selection* dialog.

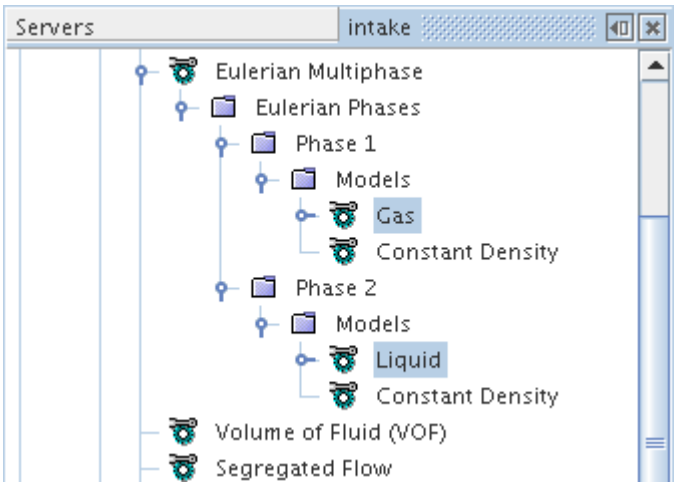
For the Multiphase Segregated Flow model, you can choose from:

- Gas
- Particle
- Liquid
- Multi-component gas
- Multi-component liquid

For the VOF Multiphase flow model, the Particle option is deactivated.

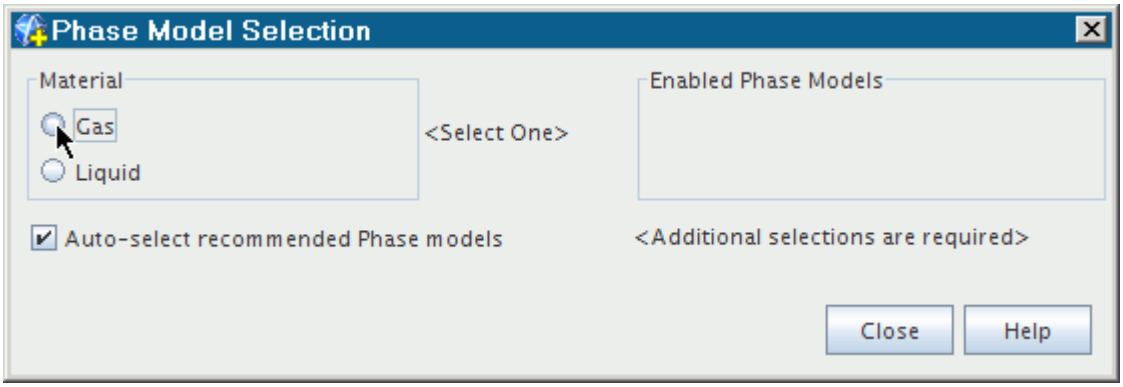
Additional documentation on working with single-component materials can be found in the section, Managing single-component materials. Documentation on working with multi-component materials can be found in the section, Managing multi-component materials. This latter section includes details on adding individual mixture components.

Upon selecting a material model, the corresponding node is displayed under the Models manager node for the phase. The material model node (**Gas** and **Liquid** in the following screenshot), which is subject to common menu items, represents the Eulerian phase material model. Each Eulerian phase has its own material model, since the phase represents a distinct physical substance.



In Multiphase Segregated Flow cases, the options for the phase material include Particle. Selecting this option allows you to define a dispersed particle phase that is modeled using the Eulerian Multiphase approach. To model the effect of particle-particle interaction within the phase, use the Solid Pressure Force model.

To define the phase material, select the appropriate material model from the *Material* group box, as shown below.



Modeling the Eulerian Phase Equation of State

A Eulerian phase requires its own Equation of State model to define how the density of the material is computed. This section explains the equation of state models available for Eulerian phases.

The particular models available depend on the underlying Multiphase Model for the continuum.

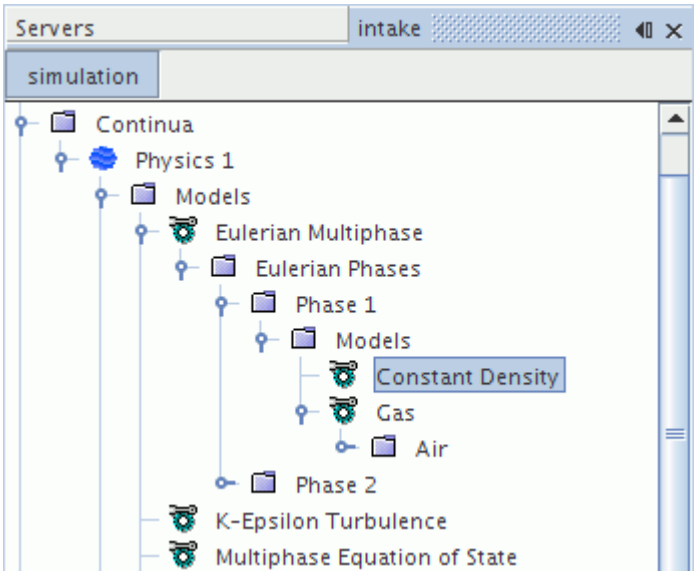
Contents:

What Is the Constant Density Eulerian Phase Model?
What Is the Ideal Gas Eulerian Phase Model?
What Is the Polynomial Density Eulerian Phase Model?
What Is the User Defined EOS Eulerian Phase Model?

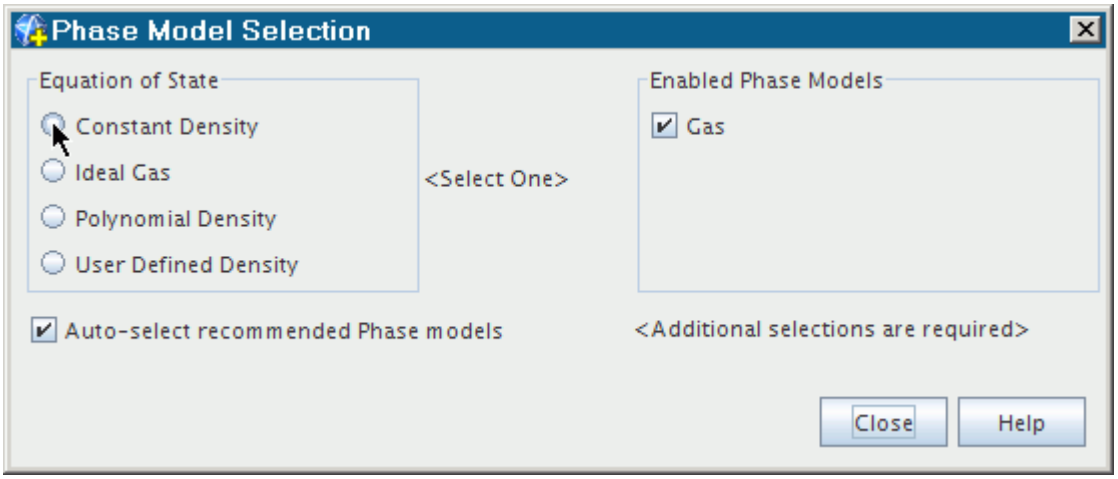
What Is the Constant Density Eulerian Phase Model?

The Constant Density Eulerian Phase model is the same as the Constant Density model for the STAR-CCM+ simulation in general, except that it applies to the Eulerian phase rather than throughout the continuum.

The **Constant Density** node, which is subject to common menu items, represents the Constant Density Eulerian phase model.



Select the **Constant Density** option from the *Equation of State* group, as shown below.



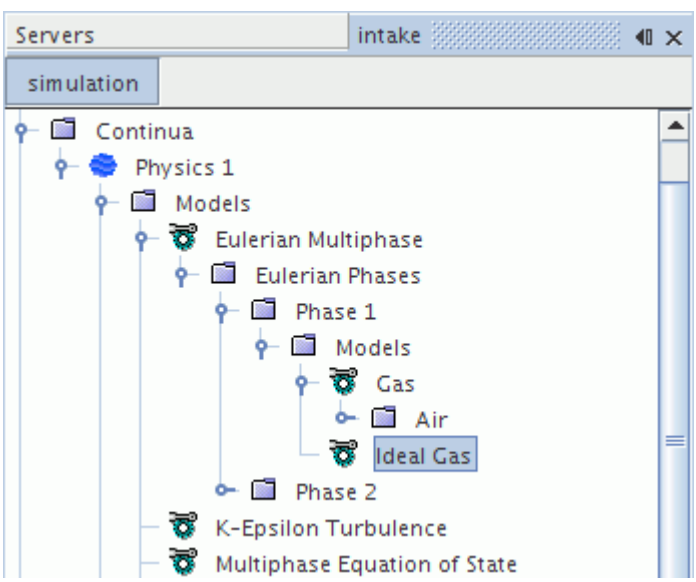
Properties

None

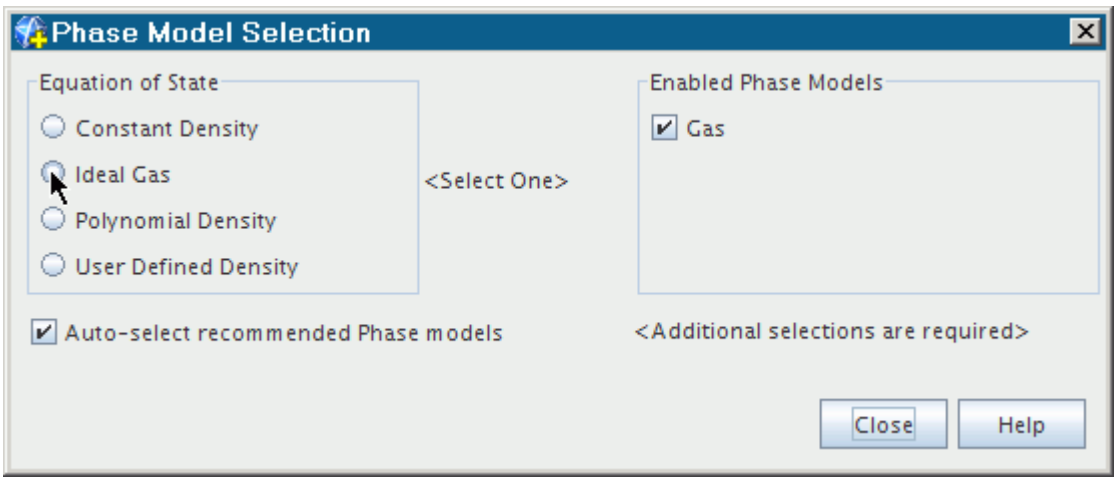
What Is the Ideal Gas Eulerian Phase Model?

The Ideal Gas Eulerian Phase model is the same as the Ideal Gas model for the STAR-CCM+ simulation in general, except that it applies to the Eulerian phase rather than throughout the continuum. In the context of Eulerian Multiphase, this model is intended for use in low Mach number flows (typically for Mach numbers less than 0.3).

The **Ideal Gas** node represents the Ideal Gas Eulerian phase model.



Activate the Gas Eulerian phase model to make the **Ideal Gas** option available. Select it from the *Equation of State* group, as shown below.



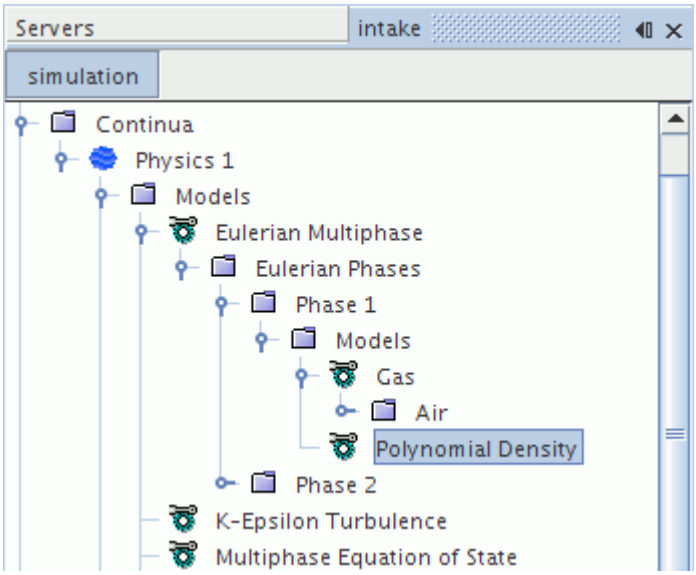
Ideal Gas Properties

Incompressible	Controls whether the ideal gas is compressible (that is, dependent on pressure).	
	Activated	Removes dependency on pressure.
	Deactivated	Allows dependency on pressure.

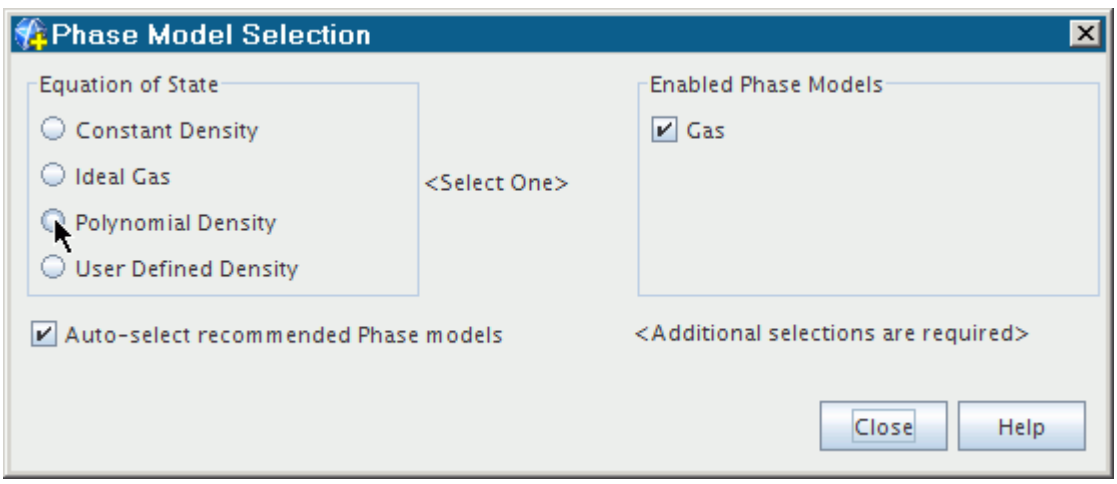
What Is the Polynomial Density Eulerian Phase Model?

The Polynomial Density Eulerian Phase model is the same as the Polynomial Density model for the STAR-CCM+ simulation in general, except that it applies to the Eulerian phase rather than throughout the continuum.

The **Polynomial Density** node represents the Polynomial Density Eulerian phase model.



Select the **Polynomial Density** option from the *Equation of State* group, as shown below.



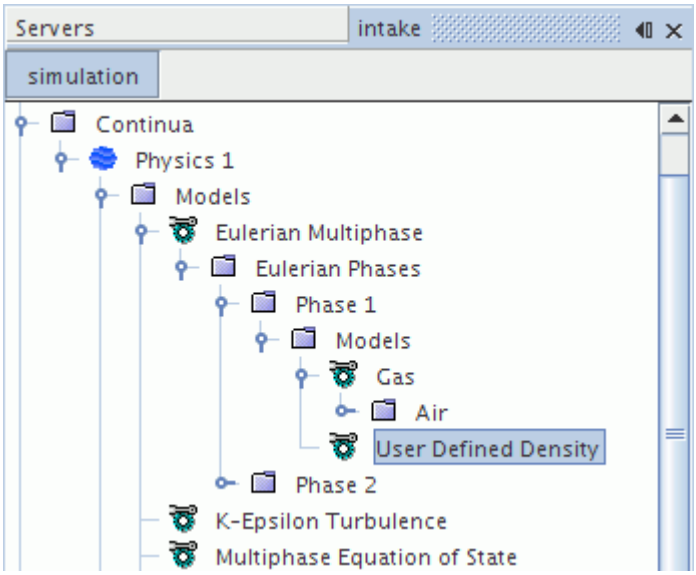
Properties

None

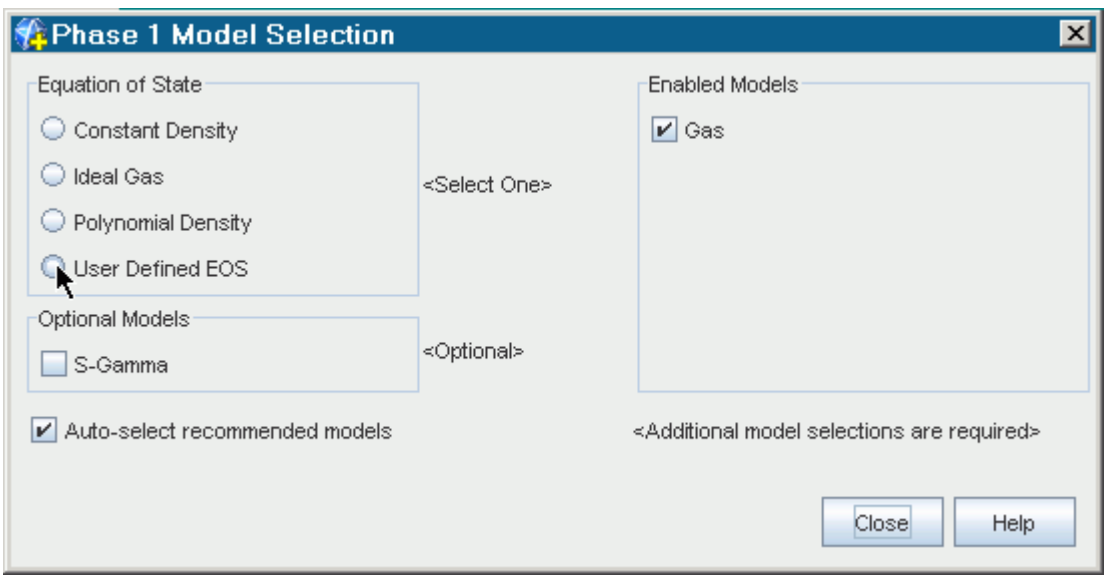
What Is the User Defined EOS Eulerian Phase Model?

The User Defined EOS Eulerian Phase model is the same as the User Defined EOS model for the STAR-CCM+ simulation in general, except that it applies to the Eulerian phase rather than throughout the continuum.

The **User-Defined EOS** node represents the User-Defined Equation of State Eulerian phase model.



Select the **User-Defined EOS** option from the *Equation of State* group, as shown below.



User Defined EOS Properties

Compressible	Controls whether the fluid is compressible (that is, dependent on pressure).	
	Activated	Allows dependency on pressure.
	Deactivated	Removes dependency on pressure.

Modeling the Eulerian Phase Viscous Regime

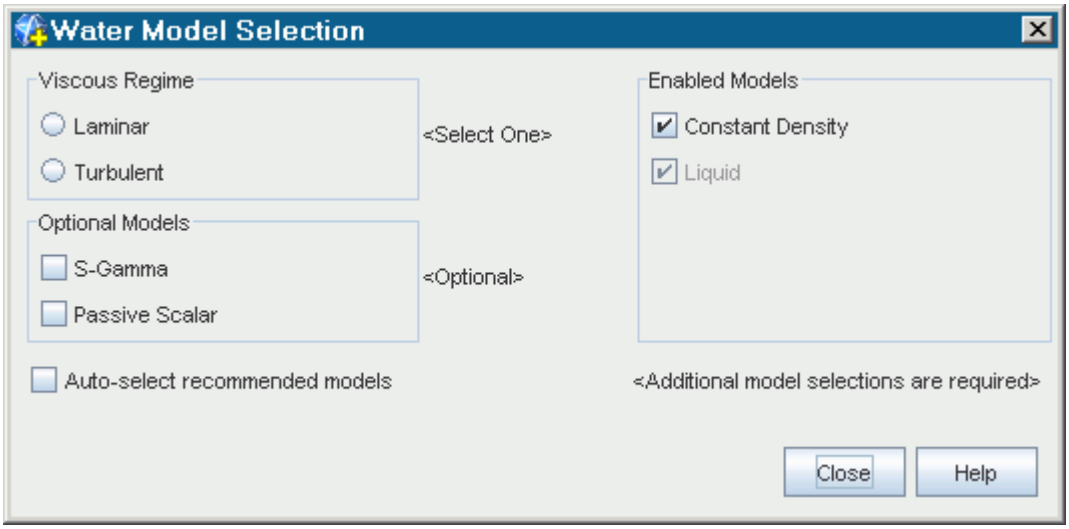
A Eulerian phase requires its own Viscous Regime model. The model depends on the viscous regime for the continuum.

If you select the Laminar or Inviscid models, all Eulerian phases are automatically set to match the continuum and cannot be changed. If you select the Turbulent model in the continuum, then phases can be set to use either the Laminar or Turbulent models (although Turbulent is activated by default).

To activate a laminar viscous regime in a turbulent continuum:

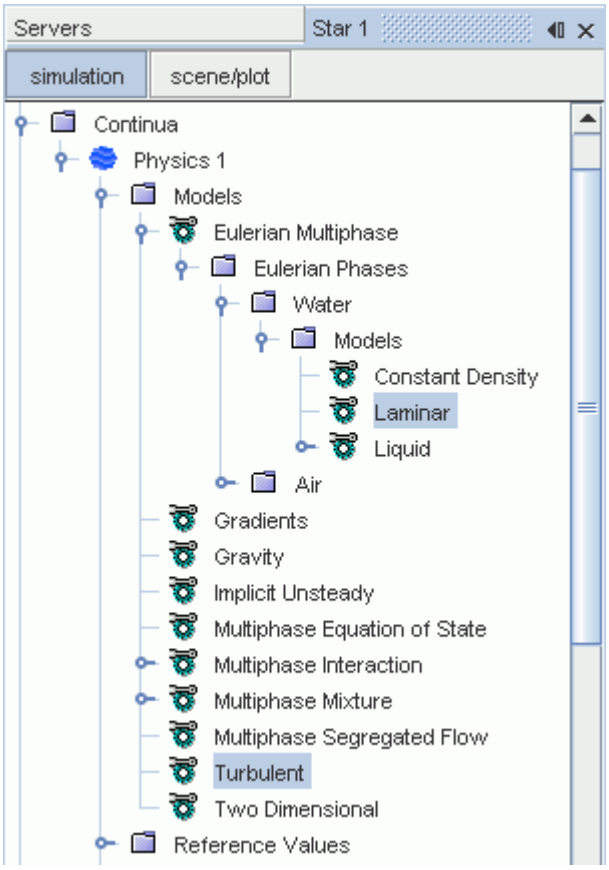
1. Right-click on the **Models** node of the phase to be changed.
2. In the *Model Selection* dialog, do the following:
   a) Deselect **Turbulent** in the *Enabled Models* box.

      The *Viscous Regime* box appears as below:



   b) Select **Laminar** from the *Viscous Regime* box.
3. Click **Close**.

The selected models are appears as below:



Modeling User-Defined Potential Forces

The User-Defined Potential Force models a potential Φ that applies to a Eulerian phase for a particular physical application. Different physical applications have different potential forces: capillary pressure and osmotic pressure are two examples of this force.

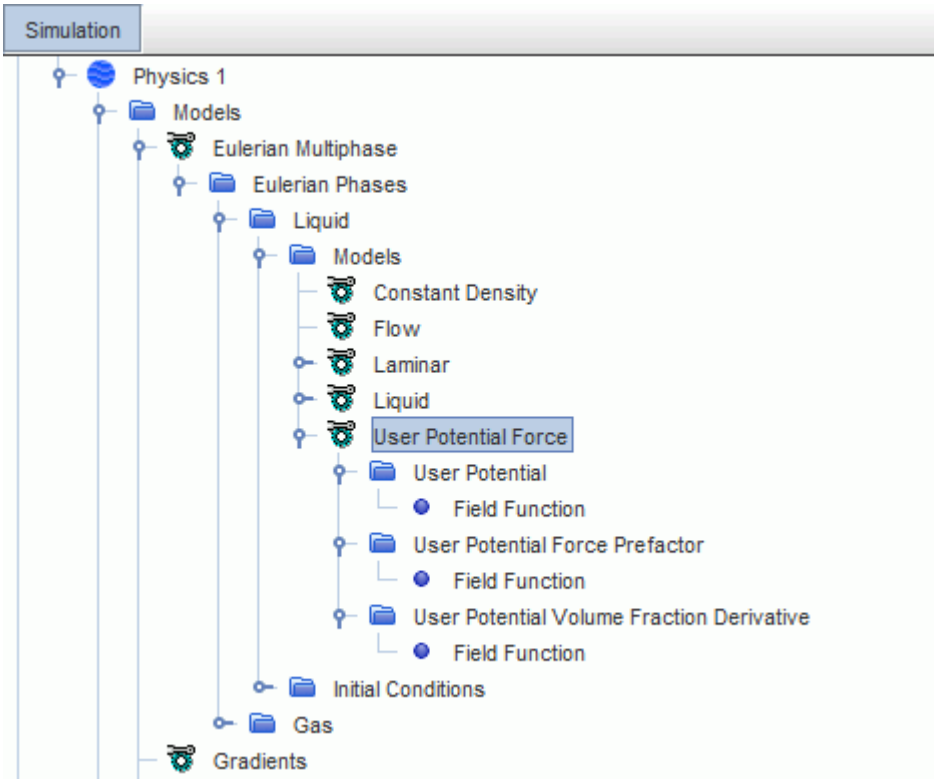
The User Potential Force model is a general model. For illustrations of how the model can be used in different applications, refer to the following examples:

- Foam Drainage
- Flow in a Porous Medium

This model is intended for use with open regions (such as for foams and settling emulsions) and with porous regions (such as capillary pressure models in oil reservoirs).

Example applications where you can use this model are the imbibition of wetting liquids into porous media, foam drainage, and the flow of dense emulsions and suspensions.

The **User Potential Force** node represents the User Potential Force model in a liquid phase.



Selecting the User Potential Force Model

To make the User Potential Force model available, you require a two-phase laminar Eulerian Multiphase simulation with one gas phase and one liquid phase.

See Selecting the Eulerian Multiphase Model and Defining Eulerian Phases.

When you set the phase models, the **User Potential Force** model is available as an optional model for the liquid phase.

You set the following properties:

- User Potential
- User Potential Force Prefactor
- User Potential Volume Fraction Derivative

User Potential Properties

The User Potential specifies the potential Φ . The potential can be a function of the volume fraction α and/or other scalars.

User Potential Force Prefactor Properties

The User Potential Force Prefactor specifies the potential force prefactor κ . The potential force prefactor can be a function of the volume fraction α and/or other scalars.

User Potential Volume Fraction Derivative Properties

The User Potential Volume Fraction Derivative specifies an approximation for the derivative of the potential with respect to the volume fraction of the phase .

This value is used for linearization only so there is no requirement for an exact expression. Excessive linearization can slow down the convergence rate, so you are recommended to use the default value $\frac{\partial \Phi}{\partial \alpha} = 0$. However, if the driving force is strong compared to other forces, it can lead to checker-boarding and instability. In these cases, the linearization stabilizes the calculations.

Contents:

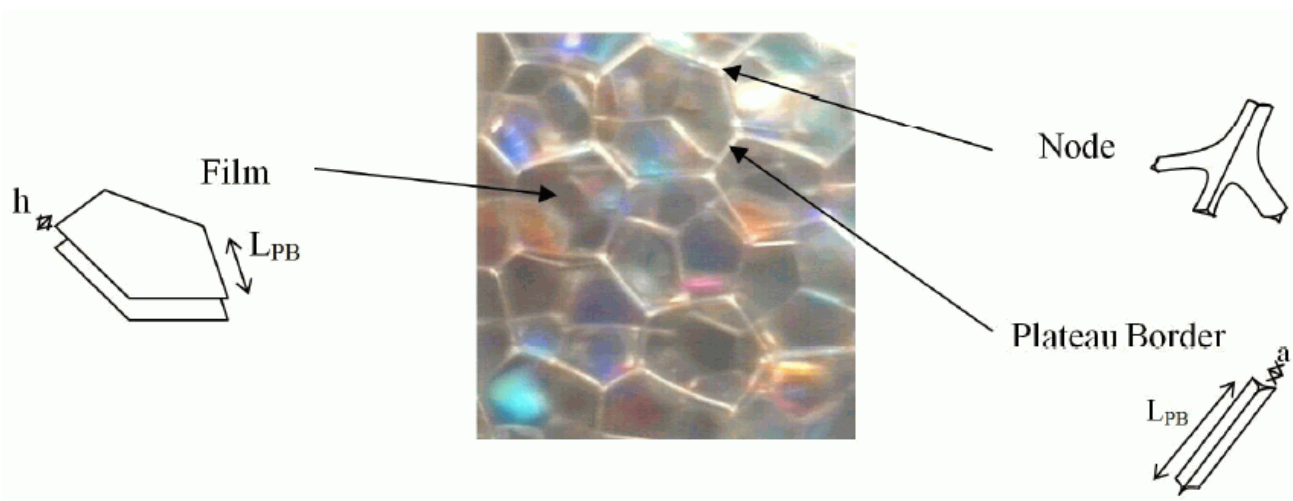
User-Defined Potential Force: Foam Drainage
User-Defined Potential Force: Flow in a Porous Medium

User-Defined Potential Force: Foam Drainage

An example application of the User-Defined Potential Force is modeling foam drainage.

The Foam Model

The structure of a foam is shown below:



The thickness of the film is negligible, and most of the liquid in a dry foam is captured in the Plateau borders. Due to the curvature of the interface, there is a negative capillary pressure P_c in the liquid phase, which supports the liquid against gravity.

This example uses the model that was proposed by Koehler [449]. In a dry foam, the radius of curvature of the Plateau borders is:

$$r \approx \delta_{\epsilon}^{-1/2} L \alpha_l^{1/2} \tag{720}$$

where:

- $\delta_\epsilon = 0.171$
- L is the characteristic length of a Plateau border.
- α_l is the volume fraction of the liquid.

The capillary pressure is:

$$P_c = -\frac{\sigma}{r} = -\frac{\sigma}{L}\frac{\delta_\epsilon^{1/2}}{\sqrt{\alpha_l}} \tag{721}$$

where σ is the surface tension.

The momentum balance for the gas and liquid phases is:

$$0 = \frac{\mu_l}{\kappa}\left(\mathbf{u}_l - \mathbf{u}_g\right) - \alpha_g \nabla P - \alpha_g \rho_g \mathbf{G} \tag{722}$$

$$0 = \frac{\mu_l}{\kappa}\left(\mathbf{u}_g - \mathbf{u}_l\right) - \alpha_l \nabla\left(P + P_c\right) - \alpha_l \rho_l \mathbf{G} \tag{723}$$

where:

- ρ is the density of the phase
- $\mathbf{u}$ is the velocity of the phase
- μ_l is the viscosity of the liquid
- P is the pressure
- $\mathbf{G}$ is the acceleration of gravity.

The permeability of the foam is given in the following formula (Koehler [449]):

$$\kappa\left(\alpha_l\right) = K_{1/2} L^2 \alpha_l^{1/2} \tag{724}$$

where $K_{1/2} \approx 2.3 \times 10^{-3}$.

Implementation in STAR-CCM+

Eqn. (721) and Eqn. (724) provide a complete model of foam drainage. To implement this model in STAR-CCM+, do the following:

1. Set up a two-phase laminar Eulerian Multiphase simulation.
2. Activate the User Potential Force model.
3. Specify the following User Potential Force properties:
   - User Potential

     In this example, this value is the capillary pressure as given in Eqn. (721).
   - User Potential Force Prefactor

     In this example, this value is the volume fraction of the liquid, α_l.
   - User Potential Volume Fraction Derivative

     In this example, this value is approximately equal to:

$$\frac{\partial P_c}{\partial \alpha} = \frac{\sigma}{2L}\frac{\delta_\epsilon^{1/2}}{\alpha_l^{3/2}}$$

4. In the phase interaction, set the Drag Force model to use the **Field Function (Linearized Coefficient)** method.

   Provide Eqn. (724) as the field function.

Simulation Results

This example uses water as the continuous phase and air as the dispersed phase. The surface tension is taken as equal to the water-air interface ($\sigma = 73 \times 10^{-3} kg/s^2$), and the length of the Plateau border is $L \approx 1\ mm$. The simulation starts with a uniform distribution of α_l. As the liquid flows down under action of the gravity, it creates a layer of pure liquid at the bottom, while some liquid is trapped by the capillary pressure. After some time, the flow stops and the foam takes its equilibrium shape.

Let z denote a vertical coordinate and z_0 denote the thickness of the liquid layer; for $z > z_0$ the two phases coexist. From Eqn. (722), $\rho_g \ll \rho_l$ so it follows that $\nabla P \approx 0$.

Therefore the force balance is:

$$\frac{\partial P_c(\alpha_l)}{\partial z} = -\rho_l G \tag{725}$$

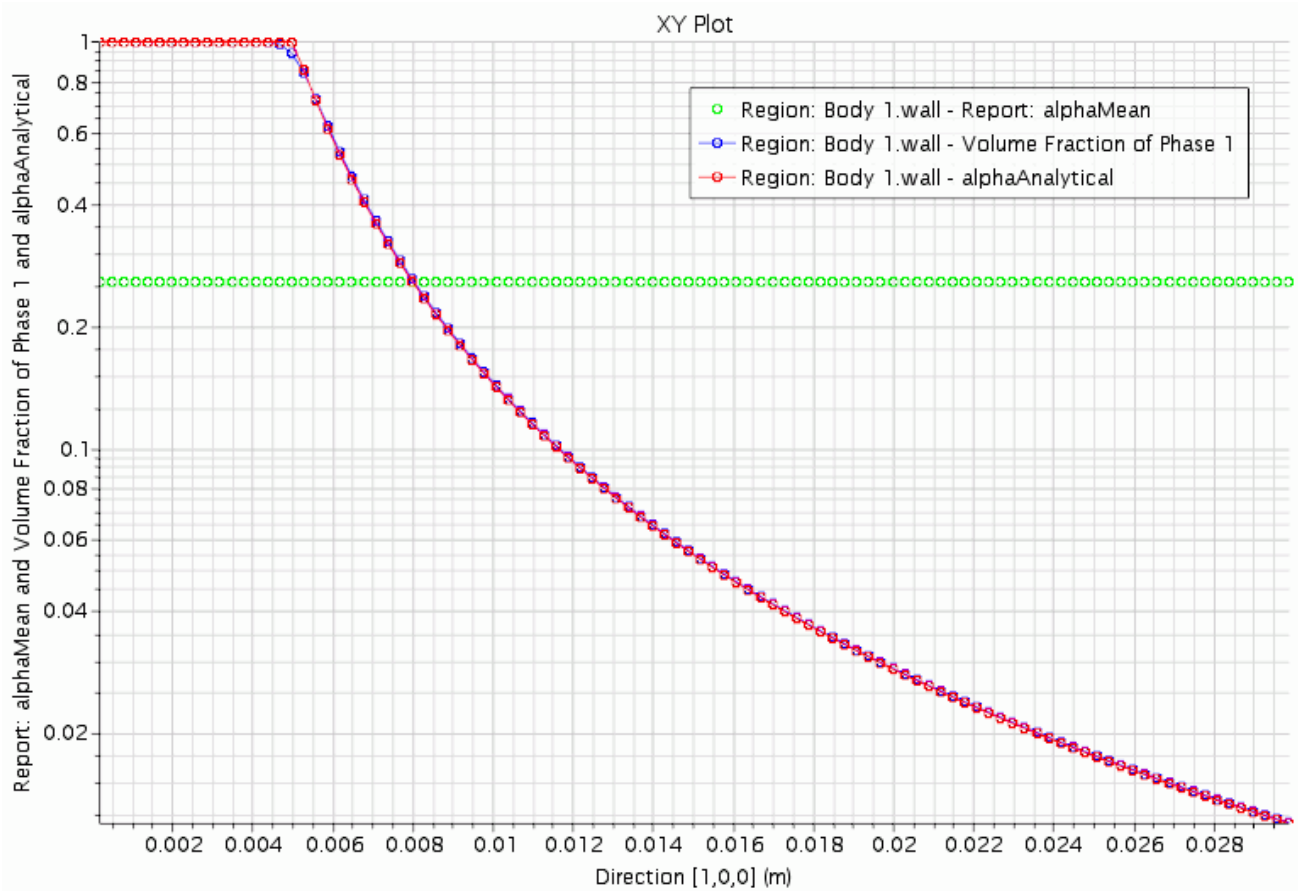
Integration of Eqn. (725) yields:

$$P_c(1) - P_c(\alpha_l) = \rho_l G(z - z_0) \tag{726}$$

Solving this equation with respect to α_l obtains the formula for the vertical distribution of the liquid:

$$\alpha_l = \left(1 + \frac{\rho_l G L}{\sigma \delta_\epsilon^{1/2}} \max(0, z - z_0)\right)^{-\frac{1}{2}} \tag{727}$$

An example of the simulation results is shown below: a plot of the vertical distribution of liquid in a static foam. The initial volume fraction $\alpha_l = 0.257$ is plotted in green. As the liquid settles down, a region with $\alpha_l = 1$ forms at the bottom, but some of the liquid remains captured by the gas phase. After approximately five minutes, the flow stops. The final vertical distribution of the liquid phase (plotted in blue) is in good agreement with the analytical solution (plotted in red).



User-Defined Potential Force: Flow in a Porous Medium

An example application of the User-Defined Potential Force is modeling flow in a porous medium.

Flow in a Porous Medium

In the absence of gravity, the momentum balance for the gas-liquid flow in a porous medium is:

$$0 = -\gamma_g \alpha_g \mathbf{u}_g - \alpha_g \nabla P \tag{728}$$

$$0 = -\gamma_l \alpha_l \mathbf{u}_l - \alpha_l \nabla \left(P + P_c\right) \tag{729}$$

where $\gamma(\alpha)$ is hydraulic resistance of a phase.

If the resistance of the gas phase is neglected, $\nabla P = 0$ and:

$$\alpha_l \mathbf{u}_l = -\frac{\alpha_l}{\gamma_l} \nabla P_c \tag{730}$$

Substitution of Eqn. (730) into the mass balance equation $\partial_t \alpha_l = -\nabla \cdot \alpha_l \mathbf{u}_l$ yields the Richards equation ([482]):

$$\frac{\partial \alpha_l}{\partial t} = \nabla \cdot \left(\frac{\alpha_l}{\gamma_l} \nabla P_c \right) \tag{731}$$

The Richards equation is a non-linear partial differential equation and only a few analytical solutions are available in the literature. However, if the resistance and capillary pressure satisfy the following equation:

$$\frac{\alpha_l}{\gamma_l} \frac{dP_c(\alpha_l)}{d\alpha_l} = C \tag{732}$$

Eqn. (731) can then be expressed as:

$$\frac{\partial \alpha_l}{\partial t} = C \nabla^2 \alpha_l \tag{733}$$

In this example, the following model is used:

$$P_c = -G_0 \frac{C}{\alpha_l^m} \quad \gamma_l = G_0 \frac{m}{\alpha_l^m} \tag{734}$$

where $m > 0$ and G_0 are arbitrary constants. To make inertia and viscosity effects negligible, G_0 is kept high ($G_0 = 10^3$).

The solution of the one-dimensional equation (Eqn. (733)) with the following initial condition:

$$\alpha_{l,0} = \frac{1}{2} + A\cos\left(C \frac{\pi}{L} x\right) \tag{735}$$

is:

$$\alpha_l = \frac{1}{2} + A \ \exp\left(-C \left(\frac{\pi}{L}\right)^2 t \right) \ \cos\left(C \frac{\pi}{L} x\right) \tag{736}$$

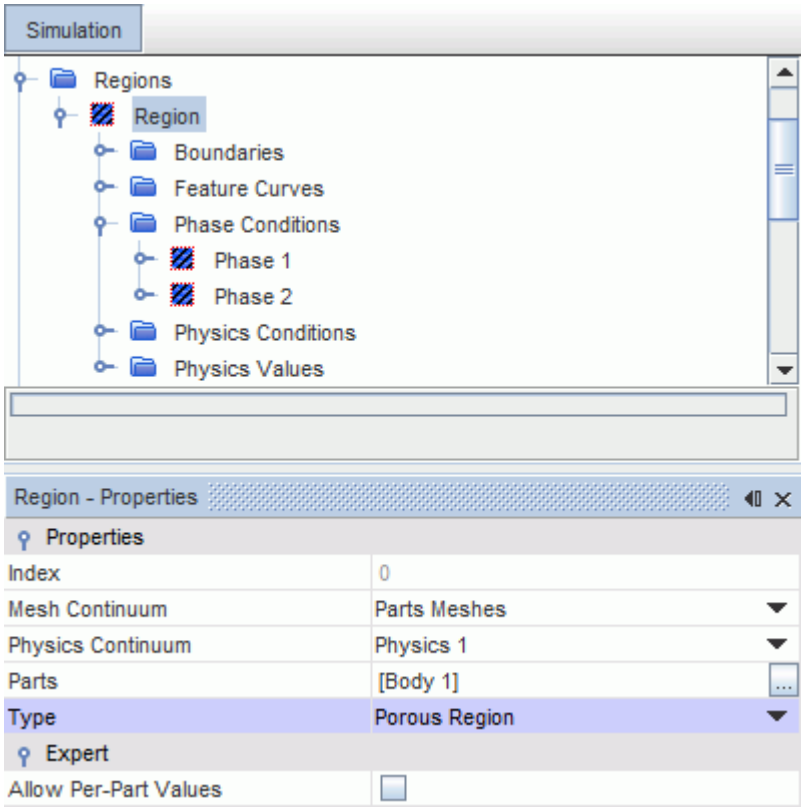
Implementation in STAR-CCM+

To implement this model in STAR-CCM+, do the following:

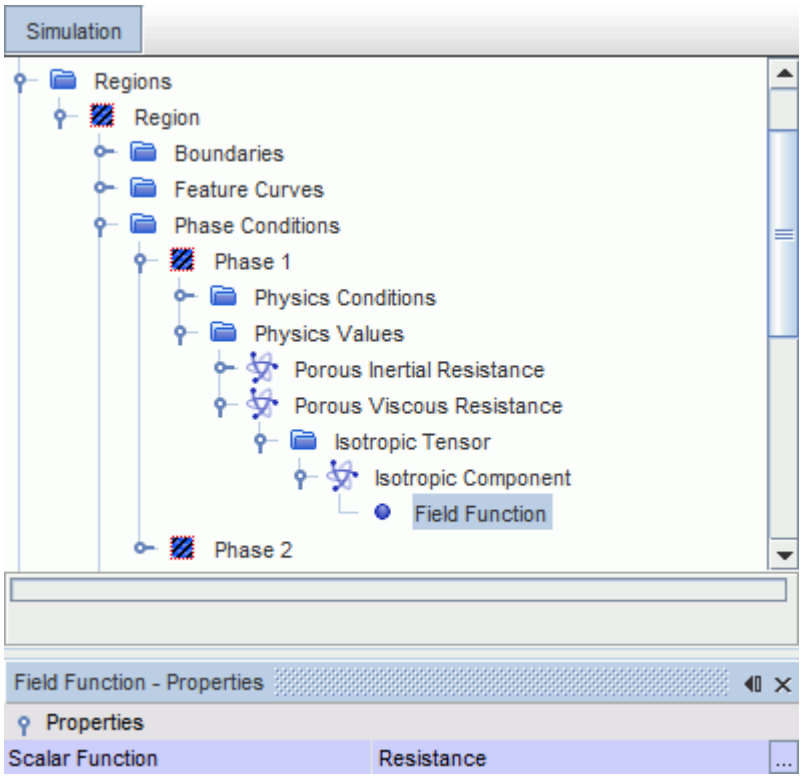
1. Set up a two-phase laminar Eulerian Multiphase simulation.
2. Activate the User Potential Force model.
3. Specify the following User Potential Force properties:
   - User Potential

     In this example, this value is the capillary pressure as given in Eqn. (734).
   - User Potential Force Prefactor

     In this example, this value is the volume fraction of the liquid, α_l.
   - User Potential Volume Fraction Derivative, $\frac{\partial P_c}{\partial \alpha}$
4. Set the initial volume fractions for the liquid phase and the gas phase ($\alpha_{l,0}$ and $(1 - \alpha_{l,0})$, respectively).
5. Set the **Region** *Type* to **Porous Region**.

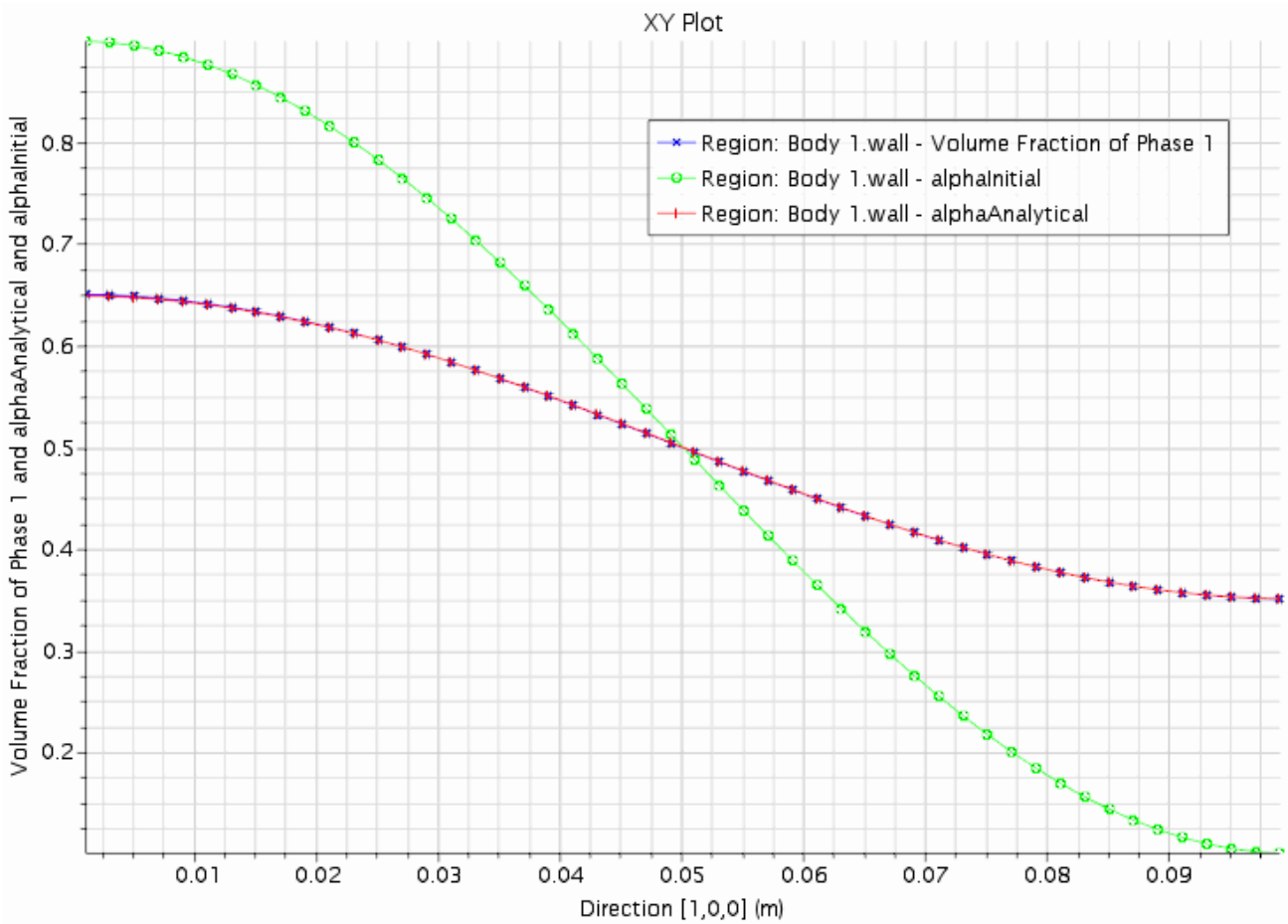


6. Specify the **Porous Viscous Resistance** as **Isotropic** and set the porous resistance γ_l (Eqn. (734)) as a field function.



Simulation Results

An example of the simulation results is shown below: a plot of the volume fraction after 10 seconds. The initial conditions are plotted in green, the analytical values from Eqn. (736) are plotted in red, and the numerical results are plotted in blue.



You can vary m and G_0 (from Eqn. (734)) as these constants have no effect on the solution.

Eulerian Multiphase Mixture Model

In the Eulerian Multiphase Mixture model, mass, momentum, and energy are treated as mixture quantities rather than phase quantities. STAR-CCM+ solves transport equations for the mixture as a whole, and not for each phase separately. The model is computationally more efficient than models that simulate each phase separately.

The Eulerian Multiphase Mixture (MMP) model is similar to the Volume of Fluid (VOF) multiphase model. Indeed, VOF can be considered a special case of MMP, with a specialized convection scheme and no diffusion. The MMP model is intended for use as a replacement for the more computationally expensive Eulerian Multiphase (EMP) model. However, in some cases, large variations between phase quantities are not well-resolved.

The MMP model is not suitable for use when the mixture properties are not good approximations of the properties of the phases. In this case, the EMP model is the better choice. The MMP model is intended for use in cases where the grid is too coarse to resolve the interface between the phases. Consequently, the MMP model cannot be expected to resolve a sharp interface, even on a fine grid. If you expect a sharp interface, use the VOF multiphase model instead.

You can apply the MMP model to cases where you can use weighted physical properties to represent a mixture of phases. A single set of equations for momentum, mass and energy transport is solved for the mixture, together with transport equations for phase volume fractions. The slip velocity between phases can also be modeled.

Examples of industrial applications of the MMP model include steam generators, engine cooling systems, and fuel cells.

The MMP model has two implementations:

- N-Phase Mixture

  The complete Eulerian Multiphase Mixture implementation, which can be used for an arbitrary combination of phases where the phase interactions can be of any kind.

  The following limitations apply:

  - The coupled solver is not supported.
  - Although inviscid flows are supported with this model, the interesting phase interactions (evaporation/condensation and drag-based phase slip) are not compatible with inviscid flow. These phase interactions require dynamic viscosity.
- Two-Phase Thermodynamic Equilibrium

  A simplified Multiphase Mixture implementation, which is designed to model applications that involve two phases of the same substance, typically water and steam, that are in thermodynamic equilibrium. The only supported phase interaction is Boiling.

  In this model, the computational efforts are reduced by assuming the suspension to be a homogeneous single-phase system. The distribution of static enthalpy and the assumption of thermodynamic equilibrium are used to calculate the distribution of phases.

  This model covers three different levels of approximation:

  - Zero Slip model (default)

The multiphase mixture is treated as a homogeneous single-phase system, with the following assumptions ([127], [133]):

- The mixture of phases behaves as a single fluid.

  A single set of transport equations for mass, momentum, and energy is solved for the mixture as a whole. You set the properties of the individual phases, and specify the phase volume fractions. The density, viscosity, and other fluid properties are calculated from the phase properties and phase volume fractions, based on the assumption of homogeneity.
- The phases are in thermal equilibrium.

  The two phases have the same temperature and the phase distribution is computed based on heat balance.
- There is no relative/slip velocity between the phases.
- The phases flow through a porous medium with porosity χ.

◦ Algebraic Slip model

This model uses empirical correlations for the slip ratio between phase velocities to calculate the void fraction. However, the kinematic effect of relative motion between the phases is not accounted for.

This approach provides a better estimate of the distribution of void fraction and mixture density than the homogenous model.

◦ Drift Flux model.

This model accounts for the kinematic effects of relative motion between the phases.

The relative motion is defined using kinematic constitutive equations for the particular flow regime and thermal-hydraulic conditions.

Contents:

Eulerian Multiphase Mixture Workflow
N-Phase Mixture Model
Two-Phase Thermodynamic Equilibrium Model
Eulerian Multiphase Mixture Model Formulation
Multiphase Mixture Bibliography

Eulerian Multiphase Mixture Workflow

The process for setting up a Eulerian Multiphase Mixture simulation is similar to the process for setting up a VOF Multiphase simulation.

(For the Two-Phase Thermodynamic Equilibrium model only) When planning the simulation domain, set the inflow boundary in a location where a pure phase enters the domain—not in a location where phase transfer occurs.

To set up a Eulerian Multiphase Mixture simulation for a physics continuum (assuming that **Auto-select recommended models** is activated):

1. In the physics model selection dialog, select an appropriate *Space* and *Time* model.
2. From the *Material* group box, select **Eulerian Multiphase**.

3. From *Eulerian Multiphase Model* group box, select **Eulerian Multiphase Mixture** and then choose the appropriate Multiphase Mixture model:
   - **N-Phase Mixture**

     See N-Phase Mixture Model.
   - **Two-Phase Thermodynamic Equilibrium**

     See Two-Phase Thermodynamic Equilibrium Model.
4. (For the Two-Phase Thermodynamic Equilibrium model only) Choose an appropriate *Flow* model.
5. Define the Eulerian phases and select the appropriate phase models.

   A typical case would have one liquid phase and one vapor phase.
   - To use the **Spalding Evaporation/Condensation** model, set up all of the phases as multi-component phases. Single-component phases are not supported. However, to model single-component phases, you can use multi-component phases that have only one component.
6. Define phase interactions and select phase interaction models.

   The Phase Interaction Topology selection is made automatically according to which Multiphase Mixture model is activated in the physics continuum: **MMP-MMP Phase Interaction** or **Two-Phase Equilibrium Interaction**.

   For N-Phase Mixture simulations, the following phase interaction models become available when the **MMP-MMP Phase Interaction** model is activated:
   - **Interaction Length Scale**

     See Modeling Interaction Length Scale.
   - **Slip Velocity**

     See Slip Velocity Model.
   - **Spalding Evaporation/Condensation**

     This model is available for multi-component phases only.

     See Spalding Evaporation/Condensation Model.

   For Two-Phase Thermodynamic Equilibrium simulations, the following phase interaction models become available when the **Two-Phase Equilibrium Interaction** model is activated:
   - **Multiphase Material** (activated automatically)
   - **Algebraic Slip** (available for segregated flow only)
   - **Drift Flux** (available for both coupled flow and segregated flow)
   - **Wall Boiling** (the Gravity model must be activated in the physics continuum)

     When the **Wall Boiling** model is activated in the *Phase Interaction Model Selection* dialog, the Rohsenow Boiling and Transition Boiling models become available for selection in the *Wall Boiling Models* group.

     Set the appropriate Rohsenow Boiling properties or Transition Boiling properties.
7. For any mass transfer models (such as **Spalding Evaporation/Condensation** or **Wall Boiling**) that have multi-component phases, set the *Connectivity* property to map the components in the liquid phase to their corresponding components in the gas phase.

N-Phase Mixture Model

The N-Phase Mixture model is the complete Multiphase Mixture implementation. This model can be used for an arbitrary combination of phases where the phase interactions can be of any kind.

N-Phase Mixture Model Reference

Theory	See Eulerian Multiphase Mixture Model Formulation.	
Provided By	**[physics continuum]** > **Models** > *Eulerian Multiphase Mixture Model*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **N-Phase Mixture**	
Requires	Pre-requisite selections: *Material*: **Eulerian Multiphase** *Eulerian Multiphase Model*: **Eulerian Multiphase Mixture**	
Properties	Key properties are: *Convection* and *Compressibility Enhancement*. See N-Phase Mixture Properties.	
Activates	Physics Models	Automatically selected models: **Multiphase Interaction** (activated by selecting the **Eulerian Multiphase** model) **Multiphase Equation of State** **Segregated Flow** **Gradients** Phase interaction models: **MMP-MMP Phase Interaction** **Spalding Evaporation/Condensation** **Drag-Based Slip Velocity** **User-Specified Slip Velocity** **Interaction Length Scale** See Models.
	Materials	**Dynamic Viscosity** **Turbulent Schmidt Number** See Materials and Methods.
	Boundary Inputs	See Boundary Settings.
	Region Settings	Physics Conditions: • Initial Condition Option • Momentum Source Option • Volume Fraction Source Option

		See Region Settings.
	Solvers	Segregated Flow Solver Segregated N-Phase Mixture Solver
	Field Functions	*Mass Flow Rate of [phase]* *Mass Flux of [phase]* *Mass Imbalance of [phase]* See Field Functions.

N-Phase Mixture Properties

Convection

Sets the discretization scheme that STAR-CCM+ uses for computing the convection flux on a cell face in appropriate transport equations:

- **1st-Order**: First-order upwind scheme. Only use when a higher-order scheme fails to give convergence, or in order to obtain an initial solution before switching to a higher-order scheme.
- **2nd-Order**: Second-order upwind scheme. Using this scheme can lead to poorer convergence properties, but gives accuracy as good as or better than the first-order scheme.

Compressibility Enhancement

Enhances the simulation of compressible MMP simulations (that is, where the Mach number exceeds about 0.3).

This method is an alternative to using the arithmetic mean value of the density of both adjacent cell centers. Computing the reconstruction gradient is computationally expensive, so this option is deactivated by default. Many MMP simulations do not require this more sophisticated approach.

Secondary Gradients

There are two sources of secondary gradients in STAR-CCM+ flow solvers:

- boundary secondary gradients for diffusion
- interior secondary gradients for diffusion at cell faces

Use this property to control which gradients are included in the solver. **On** gives both gradients while **Off** excludes them. **Interior Only** and **Boundaries Only** select the corresponding gradients.

Flow Boundary Diffusion

When activated, this property includes the flow-boundary diffusion fluxes (or viscous fluxes for flow models) as given by [eqnlink]. This property is activated by default.

Models

When the N-Phase Mixture model is activated, the following models become available for phase interactions:

Drag-Based Slip Velocity

The slip velocity is based on body forces. You can specify the drag using a Schiller-Naumann correlation or a Symmetric correlation.

See Slip Velocity Model.

User-Specified Slip Velocity

Use this model if you know the relative velocity between the two phases in the phase interaction.

See Slip Velocity Model.

Spalding Evaporation/Condensation

This model is used in applications where phases change their state from liquid to gas (evaporation) or from gas to liquid (condensation). The Spalding Evaporation/Condensation model requires all phases to be multi-component phases. Single-component phases are not supported, but you can use multi-component phases that have only one component. For a multi-component phase, not all of the components are required to change state.

See Spalding Evaporation/Condensation Model.

Interaction Length Scale

This model is activated automatically when Drag-Based Slip Velocity is selected.

See Modeling Interaction Length Scale.

Materials and Methods

Dynamic Viscosity

The dynamic (shear) viscosity of the fluid mixture is a measure of its resistance to shearing flows. The value that you set here is μ_m in the formulation.

Computed using the Volume-Weighted Mixture method.

Activated when the Laminar model or Turbulent model is selected in the Viscous Regime.

Turbulent Schmidt Number

The Turbulent Schmidt Number is used to characterize turbulent fluid flows in which there are simultaneous momentum and mass diffusion convection processes. The value that you set here is σ_t in Eqn. (737).

Activated when the Turbulent model is selected in the Viscous Regime.

Boundary Settings

Inflow Boundaries

Velocity

When the Slip Velocity model is activated, the phase velocities at the inlet boundary are extrapolated consistently from the interior domain. In this scenario, there is no direct control over the velocity of the individual phases.

Volume Fraction

The ratio of the volume that each phase occupies to the computational cell volume.

When you specify boundary values, there is no restriction on the values that you can set for the volume fractions of the phases. However, STAR-CCM+ uses normalized values for each phase volume fraction to ensure that the total volume fraction is 1.0.

Pressure Outlet

Volume Fraction

The ratio of the volume that each phase occupies to the computational cell volume.

- This setting is relevant only if backflow occurs. If you do not have backflow, ignore this setting.
- If you have backflow, and only one phase is present at the pressure outlet, set the volume fraction of that phase to 1.
- If you have backflow, and a mixture of phases is present at the outlet, it is unlikely that the pressure outlet correctly captures your physics. (A pressure outlet assumes a constant pressure across the outflow interface.) You are advised to move the outlet further downstream and bring any outlet pipes back to the horizontal orientation so that backflow no longer occurs.

Region Settings

Initial Condition Option

Lets you customize initial conditions for an individual region.

See Setting Initial Conditions for a Particular Region.

Momentum Source Option

Lets you define a momentum source term.

See Momentum Source Option.

Volume Fraction Source Option

Allows you to specify phase sources. When activated, the following nodes are added to the Physics Values:

- **Volume Fraction Sources**
- **Volume Fraction Sources Pressure Derivative**
- **Volume Fraction Sources Volume Fraction Derivative**

Segregated N-Phase Mixture Solver Properties

The Segregated N-Phase Mixture solver controls the solution update for the phase volume fractions. More specifically, it solves the discretized volume-fraction conservation equation for each phase present in the flow.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

The Single-Step solver (under the Segregated N-Phase Mixture solver) has the following property:

Under-Relaxation Factor

At each iteration, this property governs the extent to which the newly computed solution supplants the old solution. The default value is 0.9.

Field Functions

When the N-Phase Mixture model is activated, the following field functions become available:

Mass Flow Rate of [phase]

The mass of a particular phase per unit time kg/s flowing through the boundary face. The convention is such that positive mass flow rate means outflow.

Mass Flux of [phase]

The mass of a particular phase per time per area flowing through the boundary face. This value is oriented such that positive mass flux means inflow.

Mass Imbalance of [phase]

The imbalance of mass in any cell, which is defined as $\sum_f \dot{m}_f$ for each cell.

Velocity Divergence Imbalance

The imbalance of volume fluxes, equivalent to the mass imbalance for mass-flux based models.

Contents:

Slip Velocity Model
Spalding Evaporation/Condensation Model

Slip Velocity Model

To model differences in velocity between two phases in a phase interaction, use the Slip Velocity model.

In a Mixture Multiphase simulation, by default, all of the phases move at the same velocity. However, the different densities, and the size and shape of the secondary phase particles, can cause the forces acting on them to differ. Consequently, the phases can move at different velocities, which can be both different magnitudes and different directions.

The Slip Velocity model can be useful in the following applications:

- Dispersed multiphase flow in pipelines
- Bubble column reactor
- Segregation of phases under the influence of gravity
- Fuel cells

Note: The Slip Velocity model is not compatible with a porous baffle interface as a porous baffle cannot be set up with phase-specific resistance values.

Slip Velocity Model Reference

Model Names	**Drag-Based Slip Velocity**	
	User-Specified Slip Velocity	
Theory	See Slip Velocity Formulation.	
Provided By	**Phase Interactions** > **[phase interaction]** > **Models** > *Slip Velocity Models*	
Example Node Path	**[phase interaction]** > **Models** > **Drag-Based Slip Velocity**	
Requires	A Multiphase Mixture simulation with the **N-Phase Mixture** model activated. An **MMP-MMP Phase Interaction**.	
Properties	None.	
Activates	Physics Models	Phase interaction models: **Drag-Based Slip Velocity** **User-Specified Slip Velocity** Automatically selected models: **Interaction Length Scale** (activated automatically when Drag-Based Slip Velocity is selected). See Modeling Interaction Length Scale.
	Model Controls (child nodes)	**Drag Coefficient** (for Drag-Based Slip Velocity) **Slip Velocity** (for User-Specified Slip Velocity) See Slip Velocity Models.
	Solvers	Phase Slip Velocity Solver
	Field Functions	*Velocity of [primary phase]* *Velocity of [secondary phase]* See Field Functions.

Slip Velocity Models

Drag-Based Slip Velocity

The slip velocity is based on body forces. The drag coefficient can be specified using a Schiller-Naumann correlation, a Symmetric correlation, or a user-defined field function.

Method	Description
Schiller-Naumann	This drag coefficient assumes that the primary phase is continuous and the secondary phase is dispersed. See Schiller-Naumann Drag Coefficient
Symmetric	Multiple regimes are possible in this drag coefficient model, as the dispersed phase and the continuous phase are automatically selected per cell. See Symmetric Drag Coefficient

User-Specified Slip Velocity

The relative velocity between the two phases is specified as a constant, a field function, or with user-code.

Phase Slip Velocity Solver Properties

The Phase Slip Velocity solver controls the solution update for the phase slip velocity.

Temporary Storage Retained

Activates additional slip velocity based field functions. See Field Functions.

The **Neglect Local Time Derivative** child node appears only for an unsteady simulation that uses the Drag-Based Slip Velocity model.

This node has the following property:

Neglect Local Time Derivative

Stabilizes the computation of the inertial forces by neglecting the local time derivative in the force computation.

Field Functions

The following field functions are available immediately:

Velocity of [primary phase]

This value is v_p in Eqn. (739).

Velocity of [secondary phase]

This value is v_s in Eqn. (739).

When *Temporary Storage Retained* is activated in the Phase Slip Velocity solver, the following field functions become available:

Phase Slip Velocity of [Phase Interaction]

This value is v_{ps} in Eqn. (739).

Phase-Pair Reynolds Number of [Phase Interaction]

This value is Re_{ps} in Eqn. (754).

Applies to Drag-Based Slip Velocity only.

Resultant Specific Body Force

This value is $\mathbf{a}$ in Eqn. (750).

Applies to Drag-Based Slip Velocity only.

Spalding Evaporation/Condensation Model

The Spalding Evaporation/Condensation model is used in applications where phases change their state from liquid to gas (evaporation) or from gas to liquid (condensation).

The model is enabled only for multi-component phases. Single-component phases are not supported, but you can use multi-component phases that have only one component. For a multi-component phase, not all of the components are required to change state.

The Spalding Evaporation/Condensation model has no known incompatibilities. However, the following limitations apply:

- Different energy and velocity in phases is not resolved. There can be some discrepancy in the simulation results when compared to EMP and Lagrangian evaporation/condensation.
- Single-component phases are not supported. However, you can use multi-component phases that have only one component to model single-component phases.

Spalding Evaporation/Condensation Model Reference

Theory	See Evaporation/Condensation Formulation.	
Provided By	**Phase Interactions** > **[phase interaction]** > **Models** > *Optional Models*	
Example Node Path	**[phase interaction]** > **Models** > **Spalding Evaporation/Condensation**	
Requires	A Multiphase Mixture simulation with the **N-Phase Mixture** model activated. The appropriate liquid phase and corresponding vapor phase defined. An **MMP-MMP Phase Interaction** with the *Primary Phase* set to the liquid phase and the *Secondary Phase* set to the vapor phase.	
Properties	See Spalding Evaporation/Condensation Properties.	
Activates	Physics Models	Automatically selected models: **Interaction Length Scale** See Modeling Interaction Length Scale.

	Model Controls (child nodes)	See Spalding Evaporation/Condensation Child Nodes.
	Materials	**Critical Temperature** **Heat of Formation** **Saturation Pressure** **Saturation Temperature** **Standard State Temperature** See Materials and Methods.
	Field Functions	**Saturation Pressure of [liquid component]** **Saturation Temperature of [liquid component]** **Evaporation Rate of [vapor component]** See Field Functions.

Spalding Evaporation/Condensation Properties

The Spalding Evaporation/Condensation model has the following properties:

Connectivity

Maps the components in the liquid phase to their corresponding components in the gas phase.

Heat-Transfer Limited Mode

When activated, heat-transfer limited evaporation mode is used when applicable, that is, when the vapor at the liquid surface is saturated and subcritical. The condition for a saturated vapor is that the vapor mole fraction at the liquid surface exceeds unity. The vapor mole fraction at the liquid surface is evaluated as the ratio of the saturation pressure at the liquid surface temperature and the pressure in the surrounding gas.

When deactivated, mass diffusion limited evaporation mode is applied instead, and the vapor mole fraction at the liquid surface is fixed at 0.99999. Use this setting with care, as the diffusion-limited evaporation mass transfer rate can cause numerical instability if applied close to saturation conditions.

Spalding Evaporation/Condensation Child Nodes

The Spalding Evaporation/Condensation model has the following child nodes:

Nusselt Number

Sets the method for specifying the Nusselt number Nu_p that is used for calculating the mass transfer conductance g^* for heat transfer limited evaporation and condensation.

In addition to the usual methods, you can use the Armenante-Kirwan correlation that is given in Eqn. (759). The values that you can set are:

- **Alpha**

  By default, α = 0.6.
- **Beta**

By default, β = 0.5.

- **Gamma**

  By default, γ = 0.333333.

Sherwood Number

Sets the method for specifying the Sherwood number Sh_p that is used for calculating the mass transfer conductance g^* for diffusion limited evaporation and condensation.

In addition to the usual methods, you can use the Armenante-Kirwan correlation that is given in Eqn. (760). The values that you can set are the same as for the Nusselt number, above.

Interaction Length Scale

Used to define non-dimensional parameters for the phase interaction, and also to compute an interaction area density.

This value is equivalent to the droplet diameter.

See Modeling Interaction Length Scale.

Materials and Methods

The Spalding Evaporation/Condensation model activates the following material properties:

Critical Temperature

The temperature above which a gas cannot be liquefied by pressure alone. This value is $T_{c,i}$ for component i in the Evaporation/Condensation formulation.

Heat of Formation

The heat that is evolved when 1 kilogram of the substance is formed from its elements in their respective standard states [J/kg].

See Using the Heat of Formation.

Saturation Pressure

The pressure of a vapor when it is in equilibrium with its liquid at a given temperature. This value is p^i_{sat} for component i in the Evaporation/Condensation formulation.

The following methods are available:

- **Antoine Equation**

  The Antoine equation for saturation pressure is based on the logarithm of the ratio P_{sat}/P_{atm}.

  See Using the Antoine Equation.

- **Wagner Equation**

  The Wagner equation provides a good fit to experimental data, but is more complex than the Antoine equation. It expresses reduced vapor pressure as a function of reduced temperature.

  See Using the Wagner Equation.

Saturation Temperature

The temperature at which, for any given pressure, the gas is saturated; any further temperature reduction at fixed pressure results in condensation of the vapor to liquid. This value is T^{i}_{sat} for component i in the Evaporation/Condensation formulation.

In addition to the standard Constant and Field Function methods, the **Iterative** method is available. This method computes the saturation temperature automatically from the definition of Saturation Pressure. However, this method is available only if the Saturation Pressure changes with temperature, as is the case for the Antoine and Wagner methods.

Standard State Temperature

A reference point that is used to calculate the material properties under different conditions. The default value is 298.15 K (25.00 C).

See Using the Standard State Temperature.

Field Functions

The following field functions are available immediately:

Saturation Pressure of [liquid component]

Saturation Temperature of [liquid component]

When the Temporary Storage Retained flag is activated in the Segregated N-Phase Mixture solver, the following field function also becomes available:

Evaporation Rate of [vapor component]

Two-Phase Thermodynamic Equilibrium Model

The Two-Phase Thermodynamic Equilibrium model is a simplified Multiphase Mixture implementation. This model is designed for applications that involve two phases of the same substance, typically water and steam, which are in thermodynamic equilibrium.

In this model, the computational effort is reduced by assuming the suspension to be a homogeneous single-phase system. The distribution of static enthalpy, and the assumption of thermodynamic equilibrium, are used to calculate the distribution of phases.

Two-Phase Thermodynamic Equilibrium Model Reference

Theory	See Two-Phase Thermodynamic Equilibrium Model Formulation.
Provided By	**[physics continuum]** > **Models** > *Eulerian Multiphase Mixture Model*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Two-Phase Thermodynamic Equilibrium**
Requires	Pre-requisite selections: *Material*: **Eulerian Multiphase**

	Eulerian Multiphase Model: **Eulerian Multiphase Mixture**	
Activates	Physics Models	Automatically selected models: **Multiphase Interaction** **Multiphase Equation of State** Phase interaction models: **Algebraic Slip** (available for segregated flow only) See Algebraic Slip Model Properties. **Drift Flux** (available for both coupled flow and segregated flow) See Drift Flux Model Properties. **Wall Boiling** (the Gravity model must be activated in the physics continuum) See Wall Boiling Models.
	Materials	**Dynamic Viscosity** **Specific Heat** **Thermal Conductivity** **Turbulent Prandtl Number** See Materials and Methods.
	Boundary Inputs	See Boundary Settings.
	Region Settings	Physics Conditions: • Initial Condition Option • Momentum Source Option • Volume Fraction Source Option See Region Settings.
	Solvers	Segregated Flow Segregated Energy Coupled Implicit
	Field Functions	*Mass Fraction of [phase]* *Specific Heat of [phase]* *Thermal Conductivity of [phase]* *Static Enthalpy at Saturation Temperature of [phase]* See Field Functions.

Algebraic Slip Model Properties

Velocity Ratio

Specifies the slip velocity ratio. This value is S in Eqn. (775).

This value can be defined as a constant or as a field function (typically user-defined).

Drift Flux Model Properties

Distribution Parameter

Specifies the distribution factor that reflects non-uniformity in the void fraction and velocity distribution. This value is C_0 in Eqn. (784).

This value can be defined as a constant – a single three-part (or two-part for two-dimensional), comma-separated, number – or as a field function (typically user-defined).

Drift Velocity

Specifies the volume-weighted mean drift velocity. This value is $\mathbf{v}_{vj}$ in Eqn. (784).

This value can be defined as a constant, a field function (typically user-defined), or as a composite. A composite value has three components (for example, X, Y, Z) for three-dimensional problems, or two components (for example, X, Y) for two-dimensional problems.

Wall Boiling Models

The following wall boiling models are available:

Rohsenow Boiling

See Rohsenow Boiling properties.

Transition Boiling

See Transition Boiling properties.

Materials and Methods

Dynamic Viscosity

The dynamic (shear) viscosity of the fluid mixture is a measure of its resistance to shearing flows. This value is μ_m in the formulation. Computed using the Volume-Weighted Mixture method.

Specific Heat

The specific heat of the fluid mixture is the ratio of the heat added to the mixture and the resulting temperature change. This value is C_p in the formulation. Computed using the Mass-Weighted Mixture method.

Thermal Conductivity

The thermal conductivity of the fluid mixture is a measure of the ability of the mixture to conduct heat. This value is λ in the formulation. Computed using the Volume-Weighted Mixture method.

Turbulent Prandtl Number

The Turbulent Prandtl Number is the ratio of the momentum eddy diffusivity and the heat transfer eddy diffusivity in the fluid mixture. It is used in modeling heat transfer in turbulent boundary layer flows.

Activated when the Turbulent model is selected in the Viscous Regime.

Boundary Settings

No drift flux contributions are accounted for at inflow boundaries. An inflow boundary condition that includes a drift flux would imply that phase transfer (such as boiling or condensation) is taking place at the boundary. When using the Two-Phase Thermodynamic Equilibrium model, set the inflow boundary in a location where a homogeneous phase mixture enters the domain—not in a location where phase transfer occurs.

Inflow Boundaries

Volume Fraction

The ratio of the volume that each phase occupies to the computational cell volume.

When you specify boundary values, there is no restriction on the values that you can set for the volume fractions of the phases. However, STAR-CCM+ uses normalized values for each phase volume fraction to ensure that the total volume fraction is 1.0.

Pressure Outlet

Volume Fraction

The ratio of the volume that each phase occupies to the computational cell volume.

- This setting is relevant only when you have backflow. If you do not have backflow, ignore this setting.
- If you have backflow, and only one phase is present at the pressure outlet, set the volume fraction of that phase to 1.
- If you have backflow and a mixture of phases is present at the outlet, it is unlikely that the pressure outlet correctly captures your physics. (A pressure outlet assumes a constant pressure across the outflow interface.) You are advised to make your domain larger so that backflow no longer occurs.

Region Settings

Initial Condition Option

Lets you customize initial conditions for an individual region.

See Setting Initial Conditions for a Particular Region.

Momentum Source Option

Lets you define a momentum source term.

See Momentum Source Option.

Volume Fraction Source Option

Allows you to specify phase sources. When activated, the following nodes are added to the Physics Values:

- **Volume Fraction Sources**
- **Volume Fraction Sources Pressure Derivative**
- **Volume Fraction Sources Volume Fraction Derivative**

Field Functions

Mass Fraction of [phase]

The ratio of the mass of the particular phase to the total mass in the cell.

Specific Heat of [phase]

Thermal Conductivity of [phase]

Static Enthalpy at Saturation Temperature of [phase]

Eulerian Multiphase Mixture Model Formulation

The Eulerian Multiphase Mixture model describes a mixture of n phases. Both single-component and multi-component phases are supported. Mass, momentum, and energy conservation are considered for the whole mixture, rather than for the individual components.

Given volume fractions α_i of phases i, the following (volume-weighted and mass-weighted) mixture quantities are computed:

Density	$\rho_m = \sum_{i=1}^{n} \rho_i \alpha_i$
Viscosity	$\mu_m = \sum_{i=1}^{n} \mu_i \alpha_i$
Velocity	$\mathbf{v}_m = \frac{1}{\rho_m} \sum_{i=1}^{n} \mathbf{v}_i \alpha_i \rho_i$
Thermal Conductivity	$k_m = \sum_{i=1}^{n} k_i \alpha_i$
Enthalpy	$H_m = \frac{1}{\rho_m} \sum_{i=1}^{n} h_i \alpha_i \rho_i + \frac{\lvert \mathbf{v}_m \rvert^2}{2}$

The volume fractions are transported according to the following conservation equation:

$$\frac{d}{dt}\int_V \alpha_i dV + \int_A \alpha_i \left(\mathbf{v}_m - \mathbf{v}_g\right) \cdot da = \int_V \left(s_{\alpha_i} - \frac{\alpha_i}{\rho_i}\frac{D\rho_i}{Dt}\right) dV + \int_A \frac{\mu_t}{\sigma_t \rho_m} \nabla \alpha_i \cdot da \tag{737}$$

Where s_{α_i} is the source (or sink) for phase i, μ_t is the turbulent dynamic viscosity, and σ_t is the turbulent Schmidt number.

The last term is set to zero for laminar flow cases.

To clarify the turbulent diffusion term, the exact term is:

$$\frac{\mu_t}{\rho_i \sigma_t} \nabla y_i = \frac{\mu_t}{\rho_i \sigma_t} \nabla \left(\frac{\rho_i}{\rho_m} \alpha_i\right) = \frac{\mu_t}{\rho_i \sigma_t} \left(\frac{\rho_i}{\rho_m} \nabla \alpha_i + \alpha_i \nabla \left(\frac{\rho_i}{\rho_m}\right)\right)$$

However, the second component can be omitted, as typically:

$$\left|\frac{\rho_i}{\rho_m} \nabla \alpha_i\right| \gg \left|\alpha_i \nabla \left(\frac{\rho_i}{\rho_m}\right)\right|$$

Omitting the second component results in the turbulent diffusion term that is given in the formulation above, Eqn. (737).

Mass, momentum, and energy conservation are modeled for the mixture in the same way as they are for single-phase flow. See Eqn. (1461), Eqn. (1462), and Eqn. (1463), and replace all single-phase quantities with their weighted counterparts:

$$\begin{aligned} \rho &= \rho_m \\ \mathbf{v} &= \mathbf{v}_m \\ k &= k_m \end{aligned} \tag{738}$$

In the absence of any further phase interaction models, the simplest velocity model, homogeneous velocity, is assumed. That is, $\mathbf{v}_i = \mathbf{v}_m$ for all phases i.

Contents:

Slip Velocity Formulation
Evaporation and Condensation
Two-Phase Thermodynamic Equilibrium Model Formulation

Slip Velocity Formulation

For flow of nonhomogeneous multiphase mixtures, the slip between the phases introduces additional terms in the volume fraction transport equation as well as the mass, momentum, and energy conservation.

In this formulation, the subscript p is used for the primary phase, the subscript s is used for the secondary phase, and the subscript i is used when an expression is valid for any phase.

Denote the relative velocity or slip velocity between two phases p and s as:

$$\mathbf{v}_{ps} = \mathbf{v}_s - \mathbf{v}_p \tag{739}$$

The diffusion velocity is defined as:

$$v_{dr,i} = v_i - v_m \tag{740}$$

The diffusion and slip velocity are connected by the following formula:

$$v_{dr,p} = -v_{ps} - \sum_i y_i v_{si} \tag{741}$$

where the mass fractions are:

$$y_i = \frac{\alpha_i \rho_i}{\rho_m} \tag{742}$$

Assuming that all velocities except for $v_{dr,p}$ and $v_{dr,s}$ are known:

$$v_{dr,p} = \frac{-v_0 - y_s v_{ps}}{y_p + y_s}$$
$$v_{dr,s} = \frac{-v_0 + y_p v_{ps}}{y_p + y_s} \tag{743}$$

where

$$v_0 = \sum_{i \notin \{p,s\}} y_i v_{dr,i} \tag{744}$$

Navier-Stokes Equations

The Navier-Stokes equations can be rewritten in terms of the mixture velocity and the diffusion velocities.

The continuity equation is unchanged from the homogeneous case (see Two-Phase Thermodynamic Equilibrium Model Formulation), that is:

$$\frac{d}{dt}\int_V \rho_m \, dV + \int_A \rho_m \left(v_m - v_g\right) \cdot da = \int_V s_v \, dV \tag{745}$$

The momentum equation is:

$$\frac{d}{dt}\int_V \rho_m v_m \, dV + \int_A \rho_m v_m \otimes \left(v_m - v_g\right) \cdot da = -\int_A pI \cdot da + \int_A T_m \cdot da + \int_V f \, dV - \sum_i \int_A \alpha_i \rho_i v_{dr,i} \otimes v_{dr,i} \, da \tag{746}$$

The energy equation has the following terms:

$$\frac{d}{dt}\int_V \rho_m E_m \, dV + \int_A \left(\rho_m H_m \left(v_m - v_g\right) + v_g p\right) + \sum_i \alpha_i \rho_i H_i v_{dr,i} \cdot da = -\int_A q_m \cdot da + \int_A (T_m v_m) \cdot da + \int_V f_m \cdot v_m \, dV + \int_V s_E \, dV \tag{747}$$

The volume fraction transport equation is:

$$\frac{d}{dt}\int_V \alpha_i \, dV + \int_A \alpha_i \left(v_m - v_g\right) \cdot da = \int_V \left(s_{\alpha_i} - \frac{\alpha_i}{\rho_i}\frac{D\rho_i}{Dt}\right) dV - \int_V \frac{1}{\rho_i} \nabla \cdot \left(\alpha_i \rho_i v_{dr,i}\right) dV + \int_A \frac{\mu_t}{\sigma_t \rho_m} \nabla \alpha_i \cdot da \tag{748}$$

Slip Velocities

The slip velocities [132] can be specified (using the User-Specified Slip Velocity model) or modeled as follows:

$$\mathbf{v}_{ps} = c_d \mathbf{a} \tag{749}$$

where the body forces $\mathbf{a}$ are defined as:

$$\mathbf{a} = \mathbf{g} - \left(\mathbf{v}_m \cdot \nabla\right)\mathbf{v}_m - \frac{d\mathbf{v}_m}{dt} \tag{750}$$

c_d can be specified with the Schiller-Naumann drag coefficient or the Symmetric drag coefficient.

Schiller-Naumann Drag Coefficient Model

The Schiller-Naumann drag coefficient assumes that the primary phase is continuous and the secondary phase is dispersed. c_d is defined as:

$$c_d = c_d^{ps} = \frac{\tau_s}{f_{\text{drag}}^{ps}} \frac{(\rho_s - \rho_m)}{\rho_s} \tag{751}$$

where τ_s is the particle relaxation time.

$$\tau_s = \frac{\rho_s d_s^2}{18\mu_p} \tag{752}$$

and

$$f_{\text{drag}}^{ps} = \begin{cases} 1 + 0.15 Re_{ps}^{0.687} & if \quad Re_{ps} \leq 1000 \\ 0.0183 Re_{ps} & else \end{cases} \tag{753}$$

where Re_{ps} is the phase pair Reynolds number (also known as the relative Reynolds number). Re_{ps} is defined as:

$$Re_{ps} = \frac{\rho_p \left| \vec{v}_{ps} \right| d_s}{\mu_p} \tag{754}$$

where ρ_p is the density and μ_p is the viscosity of the continuous phase.

Symmetric Drag Coefficient Model

In the Symmetric drag coefficient model, the dispersed phase and the continuous phase are automatically selected per cell, so multiple regimes are possible. c_d is defined as:

$$c_d = \chi_{primary} c_d^{ps} + \chi_{secondary} c_d^{sp} \tag{755}$$

where c_d^{ps} and c_d^{sp} are computed as above, and the indicator functions are defined as:

$$\begin{aligned} \chi_{primary} &= \frac{1}{2} \tanh\left(\frac{\frac{\alpha_p}{\alpha_p + \alpha_s} - \frac{1}{2}}{c_{\text{smooth}}} \right) \\ \chi_{secondary} &= 1 - \chi_{primary} \end{aligned} \tag{756}$$

with a smoothing constant $c_{\text{smooth}} = 0.01$

Evaporation and Condensation

The Evaporation and Condensation phase interaction model is used in applications where phases change their state from liquid to gas (evaporation) or from gas to liquid (condensation).

The Evaporation and Condensation model assumes that the phases are liquid droplets dispersed in a gas. The following physical assumptions are made:

- The droplets are internally homogeneous.
- The droplets consist of an ideal mixture of liquid/gaseous components, some of which are transferred to the gas phase.
- Inert components can exist in the droplets and/or gas.

The following conditions can be identified for transferred components i (where the sum over transferred components is written as $\sum_T$):

Condition	Evaporation Computation
$T_m \geq \max T_{c,i}$	Super-critical evaporation
$\sum_T X_{is} \geq 1$ or $\sum_T Y_i = 1$	Heat-limited evaporation
All other conditions	Vapor diffusion-limited evaporation

Where:

- T_m is the mixture temperature
- $T_{c,i}$ is the critical temperature of the transferred component i
- Y_i are the mass fractions
- X_{is} are the equilibrium mole fractions at the droplet surface:

$$X_{is} = \frac{p^i_{\text{sat}}(T_m)}{p} X_{ip} \tag{757}$$

Where X_{ip} is the component mole fraction in the droplet, and $p^i_{\text{sat}}(T)$ is the saturation pressure of transferred component i at temperature T.

For super-critical evaporation, all transferred components vaporize immediately. Otherwise, the rate of change of each transferred component i due to quasi-steady evaporation $\dot{m}_{pi}$ is:

$$\dot{m}_{pi} = -\varepsilon_i g^* A_s \log(1 + B_i) \tag{758}$$

Where ε_i, B_i, g^* are determined as follows:

	Evaporation Type	
	Heat Transfer Limited	Diffusion Limited
$\varepsilon_i =$	$\frac{Y_{ip}}{\sum_T Y_{ip}}$	$\frac{Y_{is}(1 + B_i) - Y_i}{B_i}$
$B_i =$	$\frac{c_p(T_m - T^i_{\text{sat}})}{\sum_T \varepsilon_i L_i}$	$\frac{\sum_T Y_{is} - \sum_T Y_i}{1 - \sum_T Y_{is}}$
$g^* =$	$\frac{kNu_p}{c_p D_p}$	$\frac{\rho_m D_p Sh_p}{D_p}$

$Y_{is} = X_{is}\frac{W_i}{W_s}$ is the equilibrium mass fraction of transferred component i at the droplet surface.

The Nusselt number and Sherwood number can be specified as a constant, a field function, or with user code. Alternatively, they can be evaluated automatically with the Armenante-Kirwan correlation:

$$Nu_p = 2.0 + \alpha \mathrm{Re}_T{}^{\beta}\mathrm{Pr}^{\gamma} \tag{759}$$

$$Sh_p = 2.0 + \alpha \mathrm{Re}_T{}^{\beta}\mathrm{Sc}^{\gamma} \tag{760}$$

where:

$$\mathrm{Re}_T = \frac{\epsilon^{\frac{1}{3}} D_p^{\frac{4}{3}}}{\nu} \tag{761}$$

The turbulent dissipation rate ϵ is computed from the turbulent length and time scales for all models that provide it (all turbulence models except for the Spalart-Allmaras model). For the Spalart-Allmaras model, and in the laminar and inviscid case, this correlation falls back to the constant 2.0. Default values for the parameters α, β, γ are:

$\alpha = 0.6,\ \beta = \frac{1}{2},\ \gamma = \frac{1}{3}$

The symbols that are used in this formulation are defined below:

Symbol	Definition
A_s	droplet surface area $\left[m^2\right]$
B_i	Spalding transfer number
c_p	specific heat capacity $\left[\frac{J}{kgK}\right]$
D_p	droplet diameter $[m]$
D_v	molecular diffusivity of the gas phase $\left[\frac{m^2}{s}\right]$
ε_i	fractional mass transfer rate of component i
g^*	mass transfer conductance in the case $B \rightarrow 0$ $\left[\frac{kg}{m^2 s}\right]$
k	thermal conductivity $\left[\frac{W}{mK}\right]$
L_i	latent heat of vaporization of component i $\left[\frac{J}{kg}\right]$
$\dot{m}_{pi}$	rate of change of transferred component i due to quasi-steady evaporation $\left[\frac{kg}{s}\right]$
Nu_p	Particle Nusselt Number
p	pressure (outside of the droplet) $[Pa]$

Symbol	Definition
$p^i_{\text{sat}}(T)$	saturation pressure of transferred component i at temperature T $[Pa]$
ρ_m	mixture density $\left[\frac{kg}{m^3}\right]$
Sh_p	Particle Sherwood Number
$T_{c,i}$	critical temperature of transferred component i $[K]$
T_m	mixture temperature $[K]$
$T^i_{\text{sat}}(p)$	saturation temperature of transferred component i at pressure p $[K]$
W_i	molecular weight of transferred component i
W_s	molecular weight of the mixture at the droplet surface
X_{ip}	component mole fraction in the droplet
X_{is}	equilibrium mole fraction of transferred component i at the droplet surface
Y_i	mass fraction of component i in gas phase
Y_{ip}	mass fraction of component i in the particle (liquid phase)
Y_{is}	equilibrium mass fraction of transferred component i at the droplet surface

Two-Phase Thermodynamic Equilibrium Model Formulation

The Two-Phase Thermodynamic Equilibrium multiphase mixture model is a simplified formulation that models a multiphase flow where the secondary phase is suspended in the primary phase (for example, bubbles in a liquid flow). This model is intended for applications that involve two phases of the same substance (such as water and steam), that are in thermodynamic equilibrium.

To reduce the computational effort, the suspension is assumed to be a homogeneous single-phase system. The distribution of phases is calculated from the distribution of static enthalpy and the assumption of thermodynamic equilibrium.

Governing Equations

The following equations describe mass, momentum, and energy conservation of the mixture of phases in porous media:

$$\frac{d}{dt}\int_V \chi\rho_m \,\mathrm{d}\,V + \int_A \chi\rho_m\left(\mathbf{v}_m - \mathbf{v}_g\right)\cdot\,\mathrm{d}\,a = \int_V \chi s_m \,\mathrm{d}\,V \tag{762}$$

$$\frac{d}{dt}\int_V \chi\rho_m\mathbf{v}_m \,\mathrm{d}\,V + \int_A \chi\rho_m\mathbf{v}_m\left(\mathbf{v}_m - \mathbf{v}_g\right)\cdot\,\mathrm{d}\,a = -\int_A \chi p\mathbf{I}\cdot\,\mathrm{d}\,a + \int_A \chi\mathbf{S}\cdot\,\mathrm{d}\,a + \int_V \chi\left(\mathbf{f}_g + \mathbf{f}_p + \mathbf{s}_m\right)\mathrm{d}\,V \tag{763}$$

$$\frac{d}{dt}\int_V \left(\chi\left(\rho_m E_m\right)_{\text{fluid}} + \left(1-\chi\right)\left(\rho C_p\right)_{\text{solid}} T\right)\mathrm{d}\,V + \int_A \chi\left[\rho_m H_m\left(\mathbf{v}_m - \mathbf{v}_g\right) + \mathbf{v}_g p\right]\cdot\,\mathrm{d}\,a = -\int_A \dot{\mathbf{q}}\cdot\,\mathrm{d}\,a + \int_A \chi\left(\mathbf{S}\cdot\mathbf{v}_m\right)\cdot\,\mathrm{d}\,a + \int_V \chi\left(\mathbf{f}_g\cdot\mathbf{v}_m + s_h\right)\mathrm{d}\,V \tag{764}$$

The mixture density ρ_m is defined as:

$$\rho_m = \sum_{i=1}^{n} \alpha_i\rho_i \tag{765}$$

The mass-weighted mixture velocity $\mathbf{v}_m$ is defined as:

$$\mathbf{v}_m = \frac{\sum_{i=1}^{n} \alpha_i\rho_i\mathbf{v}_i}{\rho_m} \tag{766}$$

Where:

- s_m, $\mathbf{s}_m$, and s_h are user-defined sources of mass, momentum, and energy.
- $\mathbf{f}_g$ and $\mathbf{f}_p$ are the forces due to gravity and porous resistance, respectively.

The porous resistance force is approximated as:

$$\mathbf{f}_p = \mathbf{P}\cdot\mathbf{v}_{m_{sup}} \tag{767}$$

Where $\mathbf{P}$ is the porous resistance tensor that is composed of the viscous $\mathbf{P}_\mu$ and inertial resistance tensor $\mathbf{P}_i$:

$$\mathbf{P} = \mathbf{P}_\mu + \mathbf{P}_i\left|\mathbf{v}_{m_{sup}}\right| \tag{768}$$

and $\mathbf{v}_{m_{sup}}$ is the mixture superficial velocity, which is defined as $\mathbf{v}_{m_{sup}} = \chi\mathbf{v}_m$.

The stress tensor, $\mathbf{S}$, is estimated as:

$$\mathbf{S} = \mu_{eff}\left[\left(\nabla\mathbf{v}_m + \nabla\mathbf{v}_m^T\right) - \frac{2}{3}\left(\nabla\cdot\mathbf{v}_m\right)\mathbf{I}\right] \tag{769}$$

where μ_{eff} is the effective viscosity.

Turbulence quantities in the porous medium are either prescribed directly or by specifying the turbulence intensity along with a length scale or viscosity ratio.

Total energy E_m and total enthalpy H_m of the mixture are defined as:

$$E_m = H_m - \frac{p}{\rho_m} \tag{770}$$

$$H_m = h_m + \frac{\left|\mathbf{v}_m\right|^2}{2} \tag{771}$$

where h_m is the mixture-specific static enthalpy.

The heat flux $\dot{\mathbf{q}}$ in a porous medium is approximated as:

$$\dot{\mathbf{q}} = -\left[\chi \mathbf{K}_{\text{eff}}^{fluid} + \left(1-\chi\right)\mathbf{K}_{\text{eff}}^{solid}\right] \cdot \nabla T \tag{772}$$

where $\mathbf{K}_{\text{eff}}^{fluid}$ and $\mathbf{K}_{\text{eff}}^{solid}$ are the effective thermal conductivities in fluid and solid.

Modeling Phase Change

Assuming that the liquid and vapor phase are in thermodynamic equilibrium, steam mass quality (vapor mass fraction) x is a function of the static thermal enthalpy h ([127], [133]):

$$x = \max\left[0, \min\left(1, \frac{h_m - h_{ls}}{h_{vs} - h_{ls}}\right)\right] \tag{773}$$

Where h_{ls} and h_{vs} are the enthalpies of liquid and vapor at saturation temperature T_{sat}.

The vapor volume fraction α is calculated using the expression:

$$\alpha = \frac{x}{x + (1-x)\frac{\rho_{vs}}{\rho_{ls}}} \tag{774}$$

Where ρ_{vs} and ρ_{ls} are the densities of vapor and liquid at saturation temperature.

Contents:

Algebraic Slip Model
Drift Flux Model

Algebraic Slip Model

The Algebraic Slip model uses empirical correlations for the slip velocity ratio to obtain a better estimate of the distribution of void fraction and mixture density than the Two-Phase Thermodynamic Equilibrium multiphase mixture model.

The Algebraic Slip model uses the same set of transport equations as the Two-Phase Thermodynamic Equilibrium model (Eqn. (762), Eqn. (763), and Eqn. (764)). No kinematic effects of relative motion between the two phases are considered in the governing equations.

The Algebraic Slip model uses empirical correlations for the slip velocity ratio S:

$$S = \frac{\mathbf{v}_v \cdot \mathbf{g}}{\mathbf{v}_l \cdot \mathbf{g}} \tag{775}$$

where $\mathbf{g}$ is the gravity vector.

In the Algebraic Slip model, the expression for the vapor volume fraction (Eqn. (774)) is replaced with:

$$\alpha = \frac{x}{x + (1-x)\frac{\rho_{vs}}{\rho_{ls}}S} \tag{776}$$

where the slip velocity ratio S is calculated using an appropriate correlation.

Drift Flux Model

The Drift Flux model accounts for the kinematic effects of relative motion between the phases. This model has the advantages of the Two-Phase Thermodynamic Equilibrium model of being easy to formulate and apply, and it includes the influences of relative motion between the phases in the governing equations.

Relative motion for the Drift Flux model is defined using kinematic constitutive equations for the particular flow regime and thermal-hydraulic conditions.

The main assumptions of the Drift Flux model are ([127], [128], [133]):

- The mixture of phases behaves as a single fluid.
- The phases are in thermal equilibrium.
- The phases can move relative to each other.
- The phases flow through a porous medium with porosity χ.
- Inflow boundaries are at points where a pure phase enters the domain. STAR-CCM+ does not account for drift flux contributions at inflow boundaries.

Governing Equations

The mass, momentum, and energy conservation of a mixture of phases are described with the following set of transport equations ([127], [128], [133]):

$$\frac{d}{dt}\int_V \chi\rho_m \,\mathrm{d}\,V + \int_A \chi\rho_m\left(\mathbf{v}_m - \mathbf{v}_g\right)\cdot\,\mathrm{d}\,a = \int_V \chi s_m \,\mathrm{d}\,V \tag{777}$$

$$\begin{aligned}\frac{d}{dt}\int_V \chi\rho_m\mathbf{v}_m \,\mathrm{d}\,V + \int_A \chi\rho_m\mathbf{v}_m\left(\mathbf{v}_m - \mathbf{v}_g\right)\cdot\,\mathrm{d}\,a = -\int_A \chi p\mathbf{I}\cdot\,\mathrm{d}\,a + \int_A \chi\mathbf{S}\cdot\,\mathrm{d}\,a \\ + \int_V \chi\left(\mathbf{f}_g + \mathbf{f}_p + \mathbf{s}_m\right)\mathrm{d}\,V - \int_A \chi\frac{\alpha}{1-\alpha}\frac{\rho_v\rho_l}{\rho_m}\overline{\mathbf{v}}_{vj}\overline{\mathbf{v}}_{vj}\cdot\,\mathrm{d}\,a\end{aligned} \tag{778}$$

$$\begin{aligned}\frac{d}{dt}\int_V \chi\left(\left(\rho_m E_m\right)_{\text{fluid}} + \left(1-\chi\right)\left(\rho C_p\right)_{\text{solid}}T\right)\mathrm{d}\,V + \int_A \chi\left[\rho_m H_m\left(\mathbf{v}_m - \mathbf{v}_g\right) + \mathbf{v}_g p\right]\cdot\,\mathrm{d}\,a = -\int_A \dot{\mathbf{q}}\cdot\,\mathrm{d}\,a \\ + \int_A \chi\left(\mathbf{S}\cdot\mathbf{v}_m\right)\cdot\,\mathrm{d}\,a - \int_A \chi\alpha\frac{\rho_v\rho_l}{\rho_m}\left(h_{vs} - h_{ls}\right)\overline{\mathbf{v}}_{vj}\cdot\,\mathrm{d}\,a + \int_V \chi\left(\mathbf{f}_g\cdot\mathbf{v}_m + s_h\right)\mathrm{d}\,V\end{aligned} \tag{779}$$

Here $\overline{\mathbf{v}}_{vj}$ is the mean drift velocity of the vapor phase, which is defined as:

$$\overline{\mathbf{v}}_{vj} = \mathbf{v}_v - \mathbf{j} \tag{780}$$

Where $\mathbf{v}_v$ is the velocity of the vapor phase and $\mathbf{j}$ is the volume-weighted mixture velocity:

$$\mathbf{j} = \alpha\mathbf{v}_v + \left(1-\alpha\right)\mathbf{v}_l \tag{781}$$

The drift velocity can be specified using kinematic constitutive equations [130], which can be written as:

$$\overline{\mathbf{v}}_{vj} = \overline{\mathbf{v}}_{vj}\left(\alpha, \sigma, g, \mathbf{v}_m, p_m, \text{etc.}\right) \tag{782}$$

Knowing the drift velocity, the void-fraction-weighted mean velocity of individual phases can be estimated:

$$\begin{aligned} \mathbf{v}_v &= \mathbf{v}_m + \frac{\rho_l}{\rho_m}\overline{\mathbf{v}}_{vj} \\ \mathbf{v}_l &= \mathbf{v}_m - \frac{\alpha\rho_v}{(1-\alpha)_\rho}\overline{\mathbf{v}}_{vj} \end{aligned} \tag{783}$$

The mean drift velocity $\overline{\mathbf{v}}_{vj}$ can also be expressed as:

$$\overline{\mathbf{v}}_{vj} = \frac{\left[\left(C_0-1\right)\mathbf{v}_m + \mathbf{v}_{vj}\right]\rho_m}{C_0\alpha\rho_v + \left(1-C_0\alpha\right)\rho_l} \tag{784}$$

Where:

- C_0 is a distribution factor that reflects non-uniformity in the void fraction and velocity distribution
- $\mathbf{v}_{vj}$ is the volume-weighted mean drift velocity which could be estimated using appropriate constitutive equations.

Auxiliary Variables

The auxiliary variables that are associated with the Drift Flux model are:

- Mean drift velocity $\overline{\mathbf{v}}_{vj}$.
- Volume-weighted mean drift velocity $\mathbf{v}_{vj}$ and the distribution parameter C_0, as defined by appropriate correlations.

  See Lellouche-Zolotar Correlation.
- Volume-weighted mean velocities of vapor $\mathbf{v}_v$ and liquid $\mathbf{v}_l$ phase (Eqn. (783)).
- Velocity or slip ratio S between the phases, which can be calculated as:

$$S = \frac{\mathbf{v}_m\rho_m + \overline{\mathbf{v}}_{vj}\rho_l}{\mathbf{v}_m\rho_m + \overline{\mathbf{v}}_{vj}\rho_v\frac{\alpha}{1-\alpha}} \tag{785}$$

Lellouche-Zolotar Correlation

In a bubbly, or a churn turbulent bubbly, flow regime, Lellouche and Zolotar [131] recommended the following correlations. These correlations were derived for tubular heat exchangers.

For $\mathbf{v}_{vj}$:

$$\mathbf{v}_{vj} = C_2\left[\frac{\left(\rho_l-\rho_g\right)}{{\rho_l}^2}\cdot\sigma g\right]^{1/4}\frac{(1-\alpha)^{1/2}}{(1+\alpha)} \tag{786}$$

where:

- $C_2 = 1.41$ in the tube bundle region.
- $C_2 = 4.5$ above the tube bundle.
- g is gravity.

For C_0:

$$C_0 = \frac{L}{\left(\kappa_l + \left(1-\kappa_l\right)\alpha^{\gamma}\right)} \tag{787}$$

where:

$$\kappa_l = \kappa_0 + \left(1-\kappa_0\right)\left(\rho_g/\rho_l\right)^{1/4} \tag{788}$$

$$\kappa_0 = \left[1 + e^{-Re/10^5}\right]^{-1} \tag{789}$$

$$L = \frac{1-e^{-c_l\alpha}}{1-e^{-c_l}} \tag{790}$$

$$c_l = \frac{4.0{p_{cr}}^2}{p\left(p_{cr}-p\right)} \tag{791}$$

$$\gamma = \left[\frac{1 + 1.57\frac{\rho_g}{\rho_l}}{1-\kappa_0}\right] \tag{792}$$

and:

- p_{cr} is the critical pressure, which has a value of $221.14 \times 10^5 \mathrm{N/m^2}$ for steam and water.

Also, as recommended in [131], the value of κ_0 is bounded such that:

$\kappa_0 = \min\left(0.8, \kappa_0\right)$

or

$\mathrm{Re} = \min\left(\mathrm{Re}, 1.3863 \times 10^5\right)$

The Reynolds number Re is calculated as:

$$\mathrm{Re} = \frac{\mathrm{Gd}}{\mu} \tag{793}$$

where:

- d is either the equivalent hydraulic diameter inside the tube bundle or the inner shell diameter above the tubes.
- G is the mass flux entering the secondary-side of the generator.

Multiphase Mixture Bibliography

[127] *Collier, John G., and Thome, John R.,1996. “Convective Boiling and Condensation” (3rd ed. in paperback), Clarendon Press, Oxford,.*

[128] *Hibiki, T. and Ishii, M., 2003. “One-dimensional drift-flux model and constitutive equations for relative motion between phases in various two-phase flow regimes”, Int. Journal of Heat and Mass Transfer, 46, pp. 4935-4948.*

[129] *Hibiki, T. and Ishii, M., 2011. “Thermo-Fluid Dynamics of Two-Phase Flow”, Springer, ISBN 978-1-4419-7985-1*

[130] *Ishii, M., 1977. “One-dimensional Drift-Flux Model and Constitutive Equations for Relative Motion Between Phases in Various Two-Phase Flow Regimes”, ANL-77-47, USA.*

[131] *Lellouche, G.S. and Zolotar, B.A., February 1982. “Mechanistic Model for Predicting Two-Phase Void Fraction for Water in Vertical Tubes, Channels, and Rod Bundles”, Special Report, ELECTRIC POWER RESEARCH INSTITUTE.*

[132] *Manninen, M. and others, 1996. "On the mixture model for multiphase flow", VTT Publications 288, Technical Research Centre of Finland.*

[133] *Thome, John. R., 2010. “Engineering Data Book III”, Wolverine Tube, Inc.*

Using the Multiphase Segregated Flow Model

The Multiphase Segregated Flow model in STAR-CCM+ is based on an Eulerian-Eulerian formulation where each distinct phase has its own set of conservation equations.

Phases are considered to be mixed on length scales smaller than the length scales to resolve, and coexist everywhere in the flow domain. The volume fraction is the portion of a volume that a phase occupies.

This concept of coexisting phases is called “interpenetrating continua”. The concept assumes that you are interested in the time averaged behavior of the flow, rather than the instantaneous behavior. This assumption has important implications for understanding volume fraction in the context of the Multiphase Segregated Flow model.

Carry out the following steps when setting up a multiphase segregated flow simulation:

- Select the Multiphase Segregated Flow model and other physics models.
- Define Eulerian phases and select phase models.
- Define phase interactions and select phase interaction models.
- Define volume fractions for inflow boundaries and initial conditions.
- Specify solver settings.

Contents:

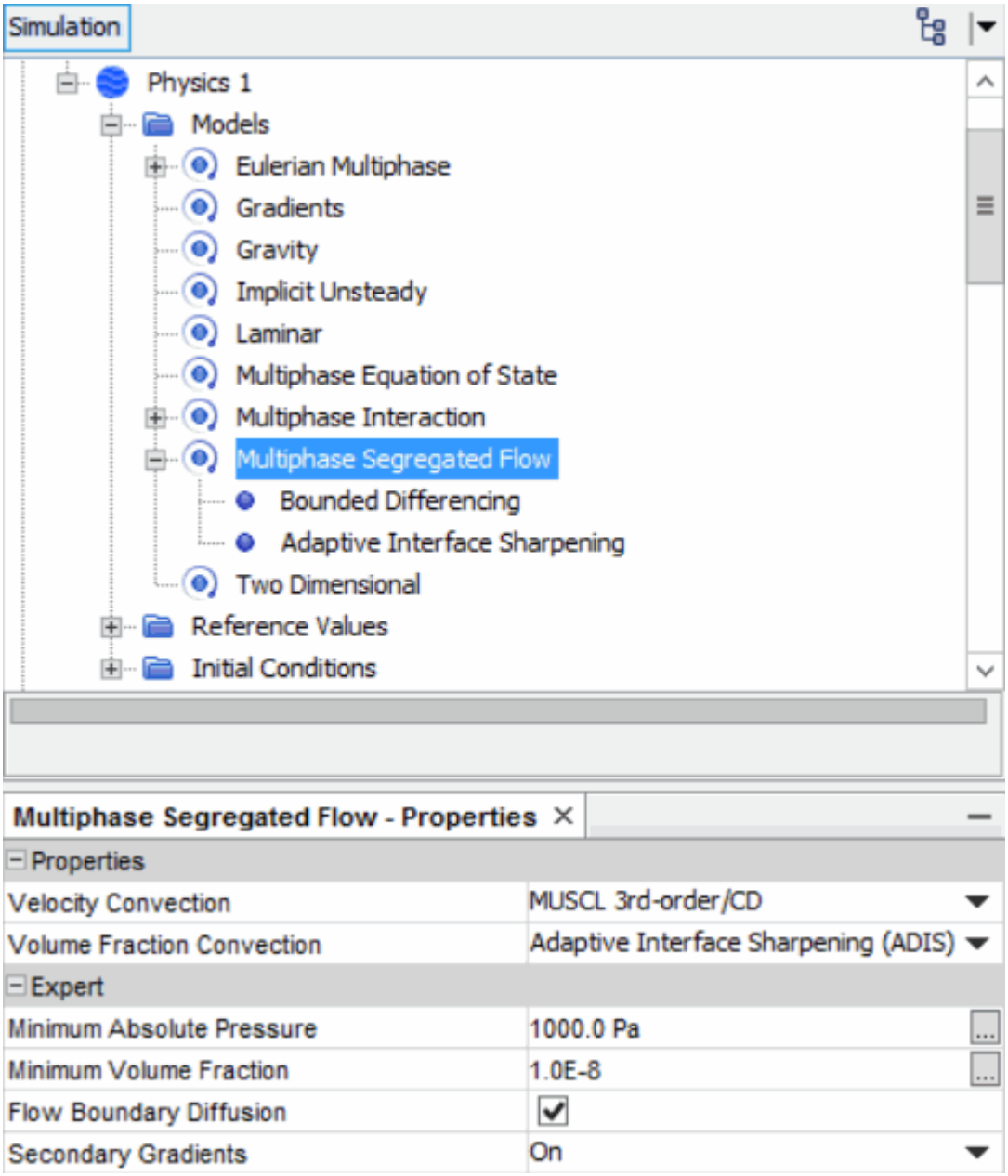
What Is the Multiphase Segregated Flow Model?
Selecting the Multiphase Segregated Flow Model
Modeling Phase Interactions in Multiphase Segregated Flow
Modeling Multiple Flow Regimes
Modeling Suspensions and Emulsions
Modeling Interphase Mass Transfer
Modeling Interphase Energy Transfer
Modeling Interphase Reactions
Working with Segregated Multiphase Flow Solvers
Modeling Turbulence in Multiphase Segregated Flow
Modeling Energy in Multiphase Segregated Flow
Modeling Combustion in Multiphase Segregated Flow
Modeling the Boiling Process
Modeling Particle-Phase Flows
Modeling Particle Size Distribution
Modeling Boundary Permeability
Using Mass Source Terms in Eulerian Multiphase Flows

What Is the Multiphase Segregated Flow Model?

The Multiphase Segregated Flow model is used to model Eulerian Multiphase cases (also referred to as Eulerian-Eulerian, Euler-Euler, EMP). The Multiphase Segregated Flow model solves a set of conservation equations for each Eulerian phase present in the simulation.

The pressure is assumed to be the same in all phases. The volume fraction gives the share of the flow domain that each phase occupies. Each phase has its own velocity and physical properties. The term “segregated” refers to the fact that the solution algorithm uses a SIMPLE-type approach, which has separate pressure and velocity solvers.

The **Multiphase Segregated Flow** node is activated as soon as the model is selected.



Interfaces separate the multiple phases present in the simulation. The motions of the phases influence the interface between each pair of phases. The phases can be any kind of fluid in the sense of moving gas, liquid or solid particles. In general, the phases are not in equilibrium and each phase has its own velocity, energy, and other variables and its own physical properties.

The conservation equations for each phase variable require closure by the definition of phase interactions at each phase interface. This definition consists of suitable models for the interfacial area, and for the rates of interphase transfer of mass, momentum and energy. These closures characterize and vary between different multiphase flow patterns, and are often not known.

Multiphase Segregated Flow Properties

Velocity Convection	Selects the convection scheme for flow velocity.	
	1st-order	Selects the first-order upwind convection scheme.
	2nd-order	Selects the second-order upwind convection scheme.
	MUSCL 3rd-order/CD	Selects the MUSCL 3rd-order/CD convection scheme. This scheme is a blend between a MUSCL 3rd-order upwind scheme and the 3rd-order central-differencing reconstruction scheme. See Hybrid MUSCL 3rd-Order/CD and Bounded Differencing.
Volume Fraction Convection	Selects the convection scheme for volume fraction.	
	1st-order	Selects the first-order upwind convection scheme.
	2nd-order	Selects the second-order upwind convection scheme. If you use this option, set the *Volume Fraction Minimum* expert property to have a value greater than the default setting of 0.0.
	Adaptive Interface Sharpening (ADIS)	Selects the Adaptive Interface Sharpening (ADIS) scheme. This scheme adaptively blends an interface sharpening scheme (the High Resolution Interface Capturing (HRIC) scheme) and a Total Variation Diminishing (TVD) scheme (the first order or second order scheme).

		See Adaptive Interface Sharpening (ADIS) Scheme for Volume Fraction. The **Adaptive Interface Sharpening** sub-node is added, allowing you to set the Adaptive Interface Sharpening parameters. This option is available only when the Large Scale Interface Detection model is enabled.

Multiphase Segregated Flow Expert Properties

Unless you are thoroughly familiar with the theoretical aspects of this model and the STAR-CCM+ discretization techniques, do not change the *Expert* settings. The values in that category have been optimized for accuracy and performance.

Minimum Absolute Pressure	Minimum allowable absolute pressure (for compressible flows).	
Flow Boundary Diffusion	Activate/deactivate the flow-boundary diffusion fluxes (or viscous fluxes for flow models)—see Eqn. (1593). This property is activated by default.	
	Activated	Activate flow-boundary diffusion.
	Deactivated	Deactivate flow-boundary diffusion.
Secondary Gradients	Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces.	
	On	Include both secondary gradients.
	Off	Exclude both secondary gradients.
	Interior Only	Include the interior secondary gradients only.
	Boundaries Only	Include the boundary secondary gradients only.
Minimum Volume Fraction	Sets a lower limit on the value of volume fraction in a cell for any phase. The default value is 0.0. If you have the *Volume Fraction Convection* set to 2nd-order, enter a value greater than zero. When instability issues arise, set to 1.0E-25.	

Adaptive Interface Sharpening Properties

The **Adaptive Interface Sharpening** node allows you to set the following properties:

Angle Factor	The angle factor in the HRIC convection discretization scheme for interface tracking. This value is C_θ in Eqn. (830).
CFL_l	The lower Courant number limit in the HRIC convection discretization scheme. This value is Cu_l in Eqn. (829).

CFL_u	The upper Courant number limit in the HRIC convection discretization scheme. This value is Cu_u in Eqn. (829).
Use Second Order TVD	When this option is deactivated, the first order scheme is used as the Total Variation Diminishing (TVD) scheme in blending.
Number of Cell Layers for Smoothing	Specifies the number of cells across which the blending function is smoothed from the Large Interface Marker Band. For more information, see Adaptive Interface Sharpening (ADIS) Scheme for Volume Fraction.

Adaptive Interface Sharpening Field Functions

The following field function is available only when **Temporary Storage Retained** is activated in the Multiphase Segregated Flow Solver.

Interface Sharpening Blending Function of [phase]

The blending function that is used for blending the HRIC and TVD schemes. This is $f_{blending}$ in Eqn. (2557).

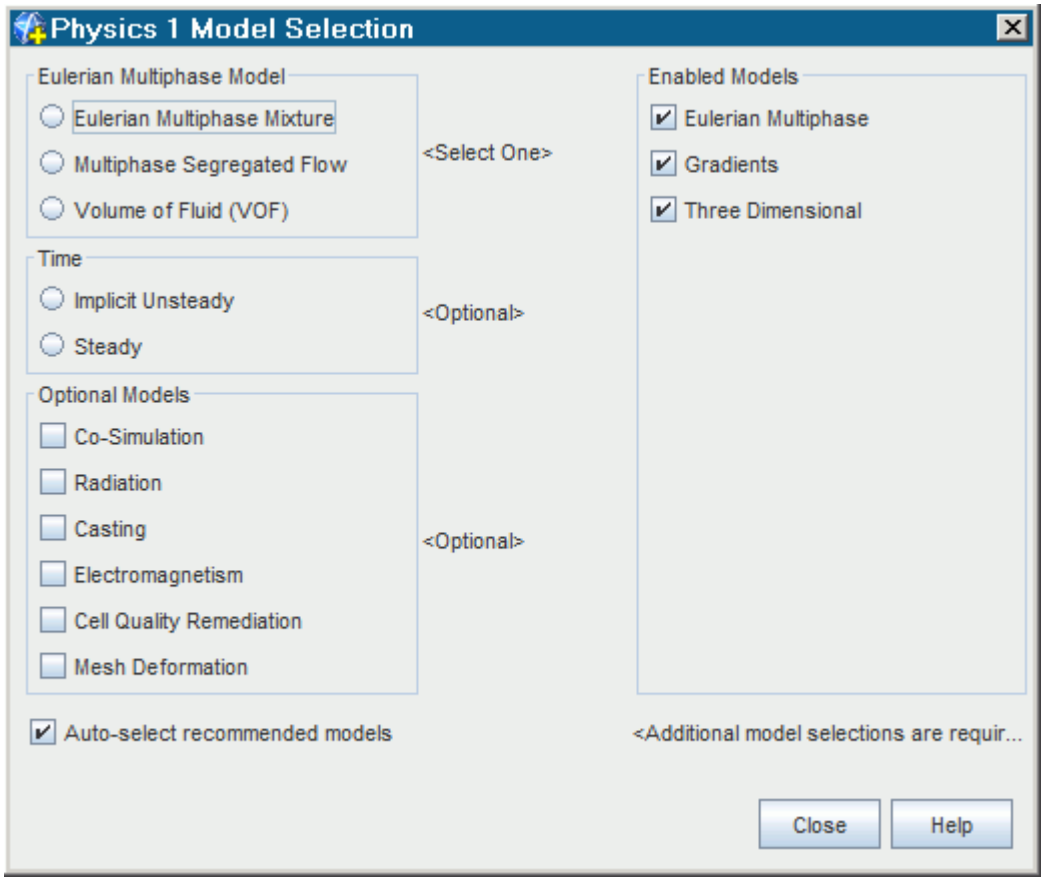
Selecting the Multiphase Segregated Flow Model

This section describes how to select and use the Multiphase Segregated Flow Model.

To select the Multiphase Segregated Flow Model:

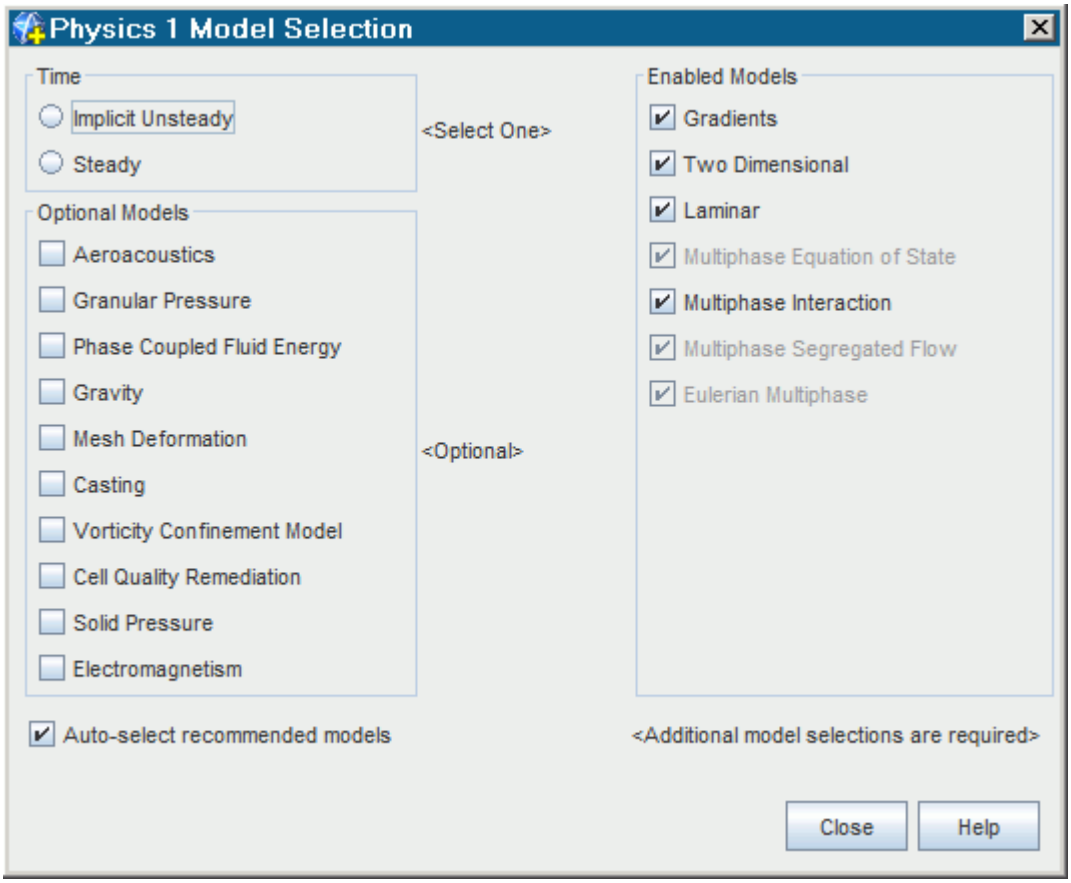
1. In the *Physics Model Selection* dialog, select option **Eulerian Multiphase** in the *Material* group box.

   The *Eulerian Multiphase Model* group box appears as shown below.



2. Select option **Multiphase Segregated Flow** from the *Eulerian Multiphase Model* group box. The *Viscous Regime* group box appears.
3. Select **Laminar** or **Turbulent** from the *Viscous Regime* group box as desired.

The **Laminar** option has been selected in the dialog that is shown below.



For Time models, use either **Steady** or **Implicit Unsteady**.

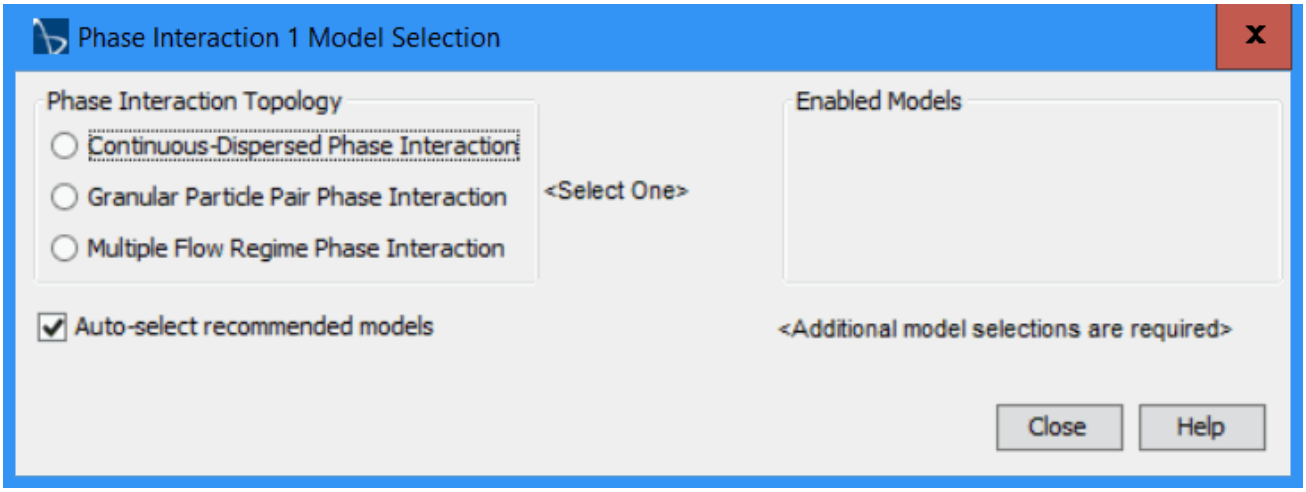
Modeling Phase Interactions in Multiphase Segregated Flow

In Multiphase Segregated Flow simulations, the phase interaction model defines the topology and physics of the interface that separates any two interacting phases.

For multiphase segregated flows, the following types of phase interactions are available:

- Continuous-Dispersed Phase Interaction
- Granular Particle Pair Phase Interaction
- Multiple Flow Regime Phase Interaction

You select the appropriate model from the *Phase Interaction Topology* group box in the *Phase Interaction Model Selection* dialog.



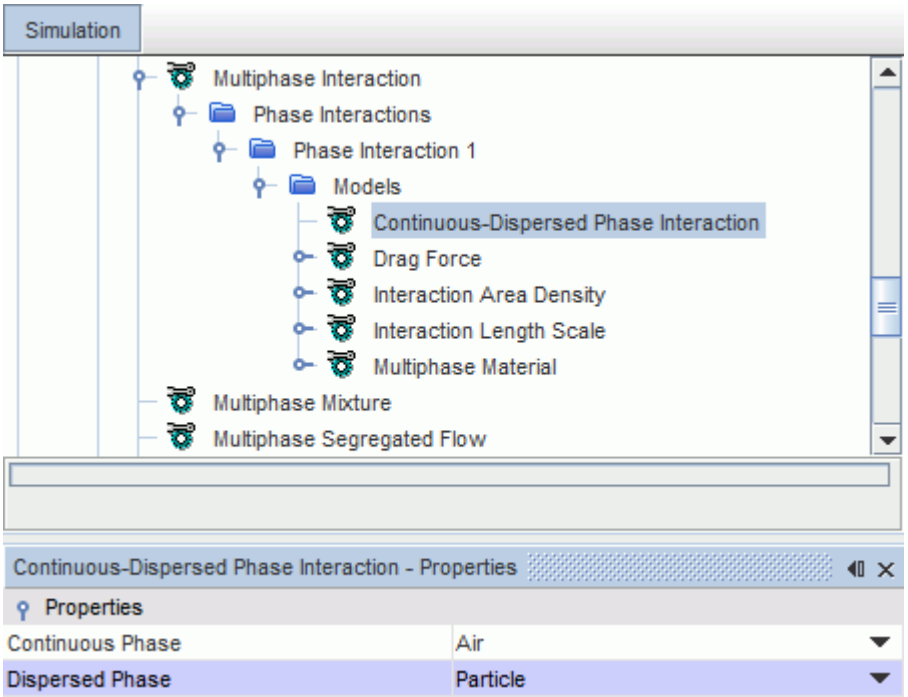
Contents:

What Is the Continuous-Dispersed Phase Interaction Model?
Modeling Interaction Length Scale
Modeling Interaction Area Density
Modeling Drag Force
Modeling Lift Force
Modeling Wall Lubrication Force
Modeling Turbulent Dispersion Force
Modeling Interface Turbulence Damping
Modeling Virtual Mass Force
What is the Virtual Mass Coefficient?
Modeling Virtual Mass Stress
Providing Additional Material Properties
What Is the Granular Particle Pair Phase Interaction Model?
Modeling Granular Drag Force

What Is the Continuous-Dispersed Phase Interaction Model?

For each Eulerian multiphase phase interaction that is not between particles in the Granular Pressure model, it is necessary to define which phase is continuous and which is dispersed.

This setting is defined on the properties of the **Continuous-Dispersed Phase Interaction** model node.



Continuous-Dispersed Phase Interaction Properties

Continuous Phase	Choose the continuous phase of the interacting pair from the drop-down list of previously defined Eulerian phases.
Dispersed Phase	Choose the dispersed phase of the interacting pair from the drop-down list of previously defined Eulerian phases.

When you choose the Continuous-Dispersed Phase Interaction model, the following models are selected automatically:

- Drag Force
- Multiphase Material
- Interaction Length Scale
- Interaction Area Density

The **Multiphase Material** node in the phase interaction **Models** manager lets you set any additional material properties that are required by some of the phase interaction models.

The following interphase transfer models are available for multiphase segregated flows:

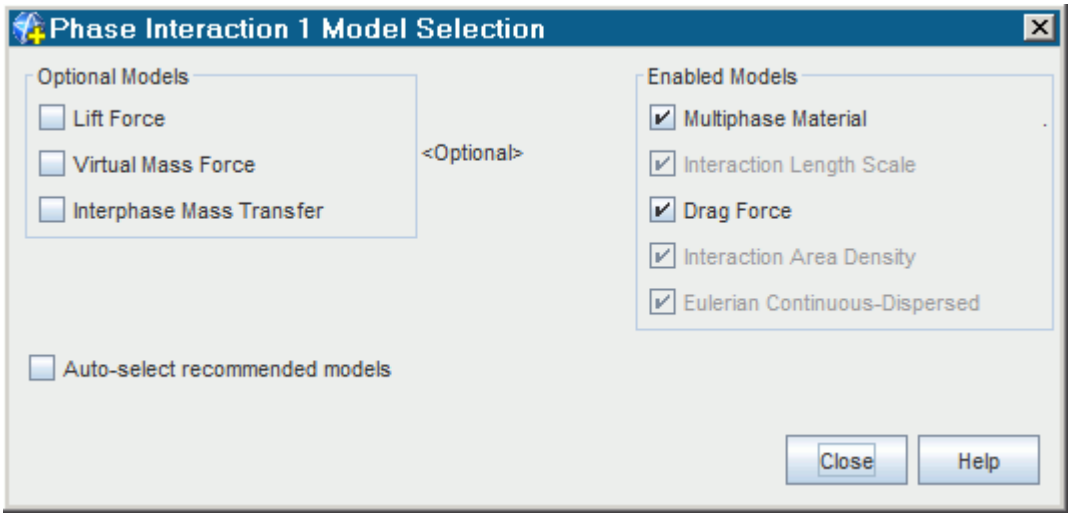
- Lift force model
- Wall Lubrication force model
- Turbulent dispersion force model
- Virtual mass force model
- Interphase Energy Transfer

If the continuous phase is a liquid and the dispersed phase is a particle, you can also select the Suspension Rheology model.

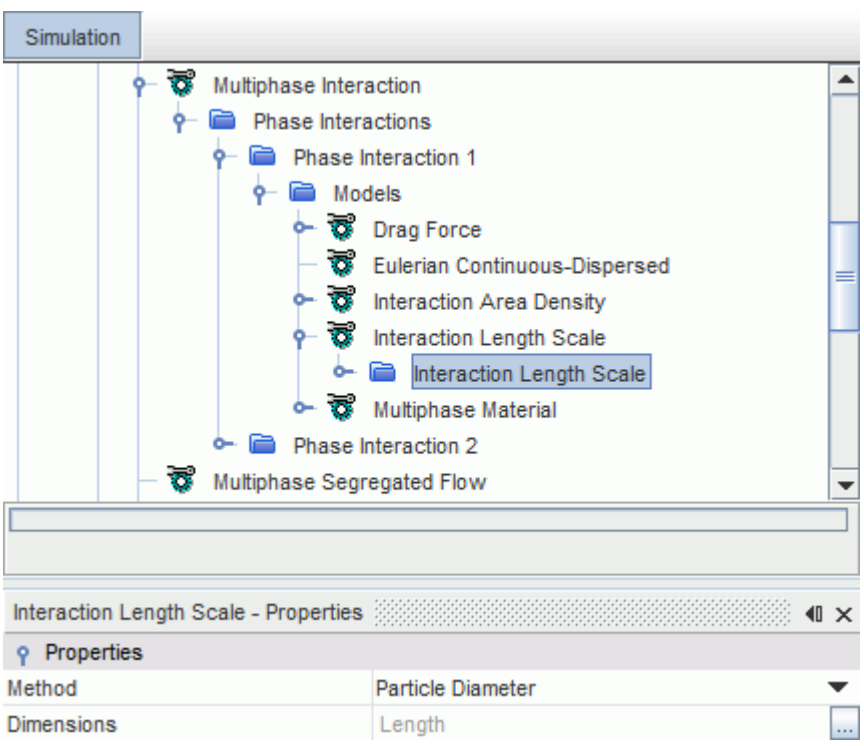
Modeling Interaction Length Scale

The interaction length scale is used to define non-dimensional parameters such as the Reynolds number for a phase interaction, and also to compute an interaction area density. The interaction length scale for a continuous-dispersed interaction is chosen to be the mean particle size.

The Interaction Length Scale model is selected automatically when you select the Drag Force model.



The **Interaction Length Scale** node is added to the phase interaction models.



In the *Properties* window, you set the method for defining the interaction length scale.

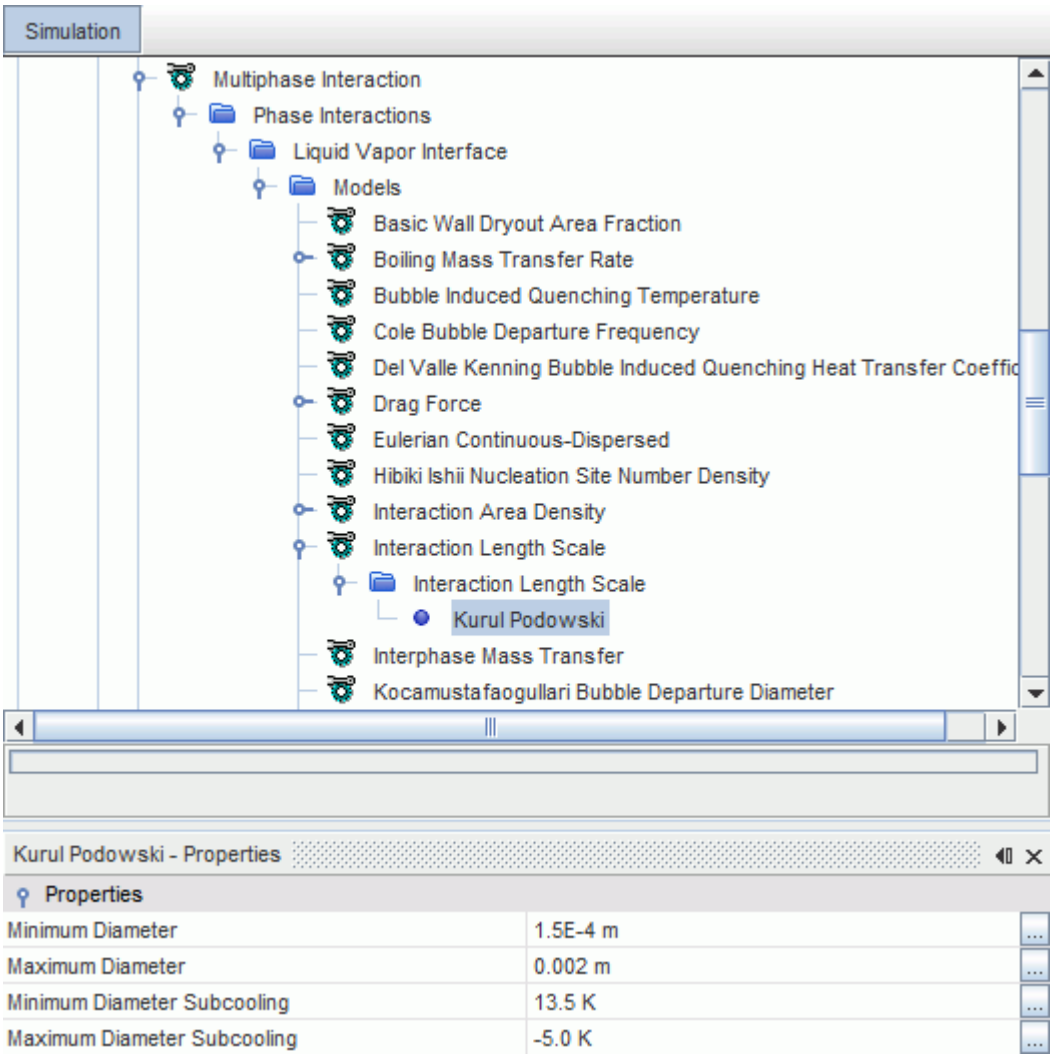
Interaction Length Scale Model Properties

Method	Specifies the method for defining the interaction length scale.	
	Constant	Specifies a constant value that represents the mean particle size for the phase interaction.
	Field Function	Specifies a field function that represents the mean particle size for the phase interaction. For example, in a flow with a constant number density of particles, particle size could be specified to be proportional to the phase volume fraction.
	Particle Diameter	Uses the particle diameter of the particle phase, which is set as the dispersed phase. This method is available when the Granular Pressure model is selected in the physics continuum. You can specify the particle diameter in the **Material Properties** node in the particle phase definition.
	Sauter Mean Diameter	Uses the Sauter Mean Diameter of the dispersed phase. This method is made available when the S-gamma model is activated for the dispersed phase.
	Kurul Podowski	Specifies bubble size as a function of local subcooling. The Kurul Podowski interaction length scale is an empirical relation that is used in boiling flows. It is available only when the boiling mass transfer rate has been selected with interphase mass transfer.
	Adaptive Multi Size Group Diameter	The usual and default method for Adaptive Multi Size Group (AMUSIG) simulations. When this option is selected in a Multi-Speed simulation, each size group automatically uses its own

		group diameter for the drag force and for other interaction models. In a Single-Speed simulation, the Sauter or surface-mean diameter is automatically used instead. See Adaptive Multi Size Group Model Reference.

Kurul Podowski Interaction Length Scale

The parameters that are required when calculating this length scale are set as properties of the **Kurul Podowski** node.



Kurul Podowski Interaction Length Scale Properties

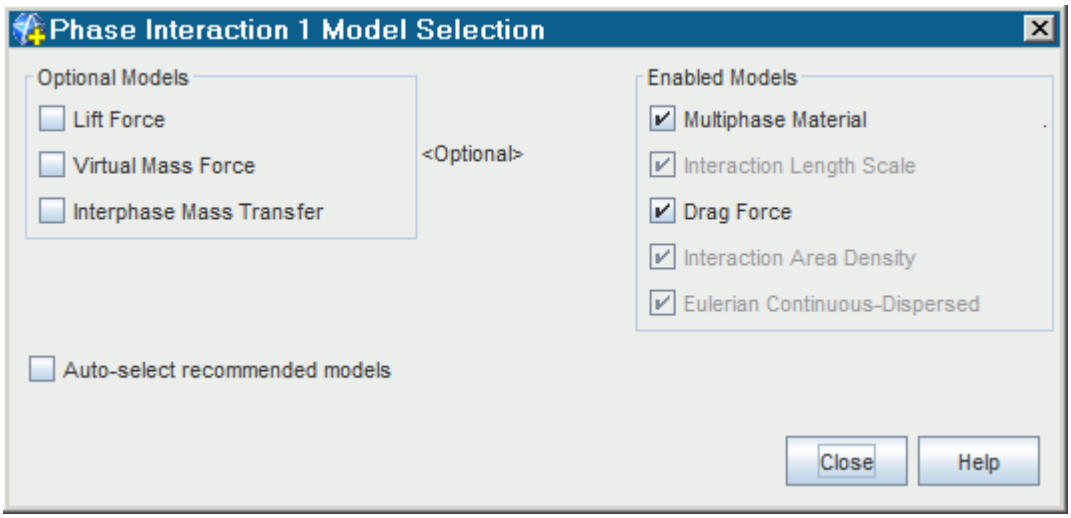
Minimum Diameter	Minimum bubble diameter

Maximum Diameter	Maximum bubble diameter
Minimum Diameter Subcooling	Liquid subcooling temperature corresponding to minimum bubble diameter
Maximum Diameter Subcooling	Liquid subcooling temperature corresponding to maximum bubble diameter

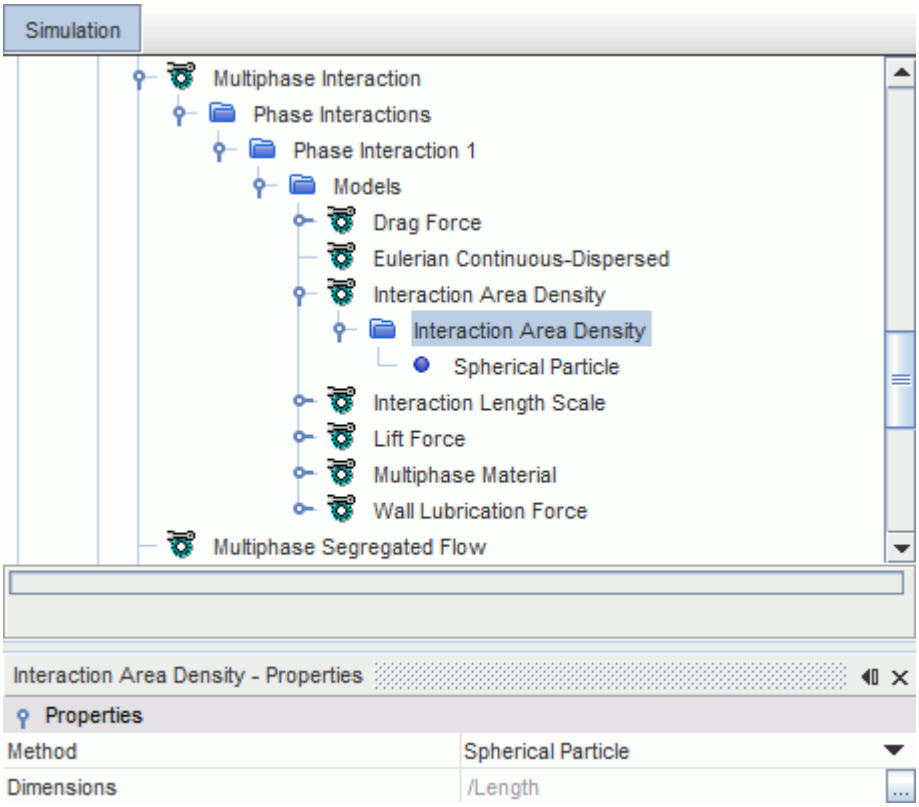
Modeling Interaction Area Density

The interaction area density specifies the interfacial area available for drag, heat, and mass transfer between each pair of phases in an interaction.

The Interaction Area Density model is selected automatically when you select the Drag Force model.



Use the *Method* property of the **Interaction Area Density** node to choose how you want to define the value.



Interaction Area Density Properties

Method	Specifies the method to use for modeling the interaction area density.	
	Spherical Particle	Defines the interaction area density using the surface area of a spherical particle, from Eqn. (2242). This model is not applicable at high particle volume fractions.
	Symmetric	Uses the availability of the continuous phase to modify the interaction area density, from Eqn. (2243). This model can be used for a wide range of volume fractions.
	Field Function	Choose a field function in the **Field Function** child node.

Modeling Drag Force

This section describes the Drag Force model and how to select it.

This section explains the following about the Drag Force model that is used in phase interactions:

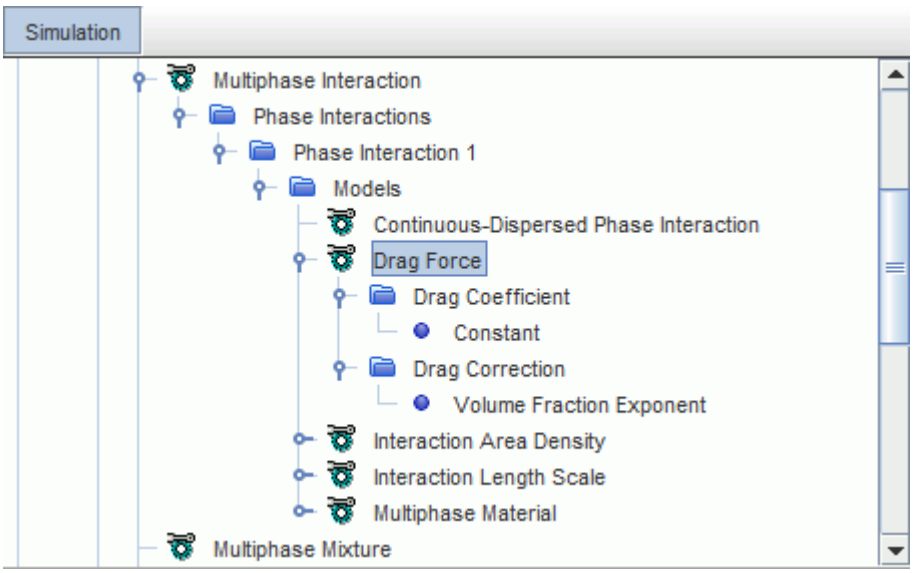
- What is the Drag Force Model?
- Selecting the Drag Force Model
- What is the Drag Coefficient?
- What is the Drag Correction?

What Is the Drag Force Model?

In STAR-CCM+, the Drag Force is modeled as a linear multiplier, A_D in Eqn. (2246), of the relative velocity between two phases.

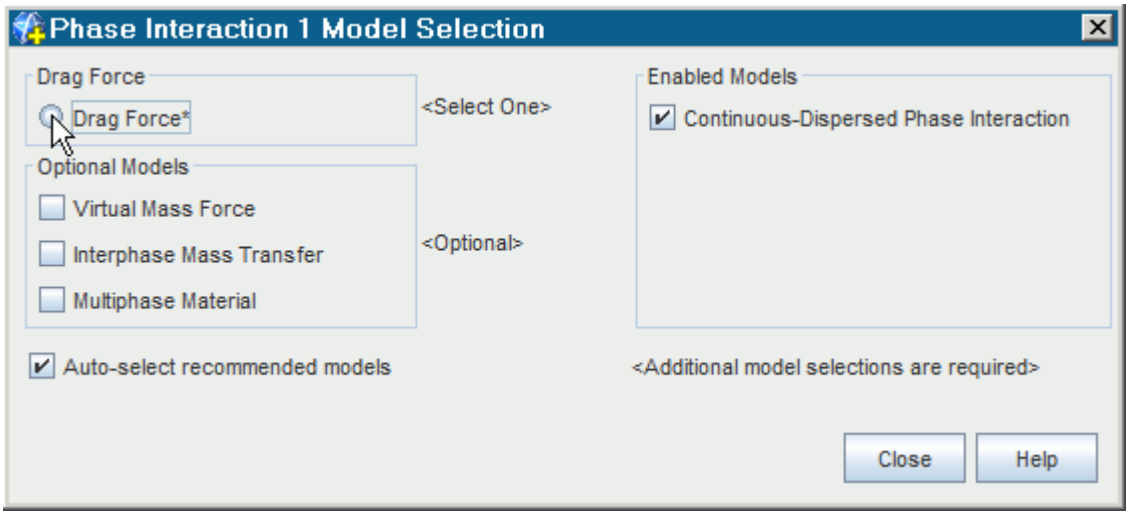
The Drag Force model calculates the force on a particle in the dispersed phase due to its velocity relative to the continuous phase, as from Eqn. (2246).

The **Drag Force** node represents the Drag Force model.



Selecting the Drag Force Model

The **Drag Force** model is selected by default in the *Phase Interaction Model Selection* dialog when you select the phase interaction from the *Phase Interaction Topology* group box. If it has been deselected, however, it can be selected from the *Drag Force* group box as shown.



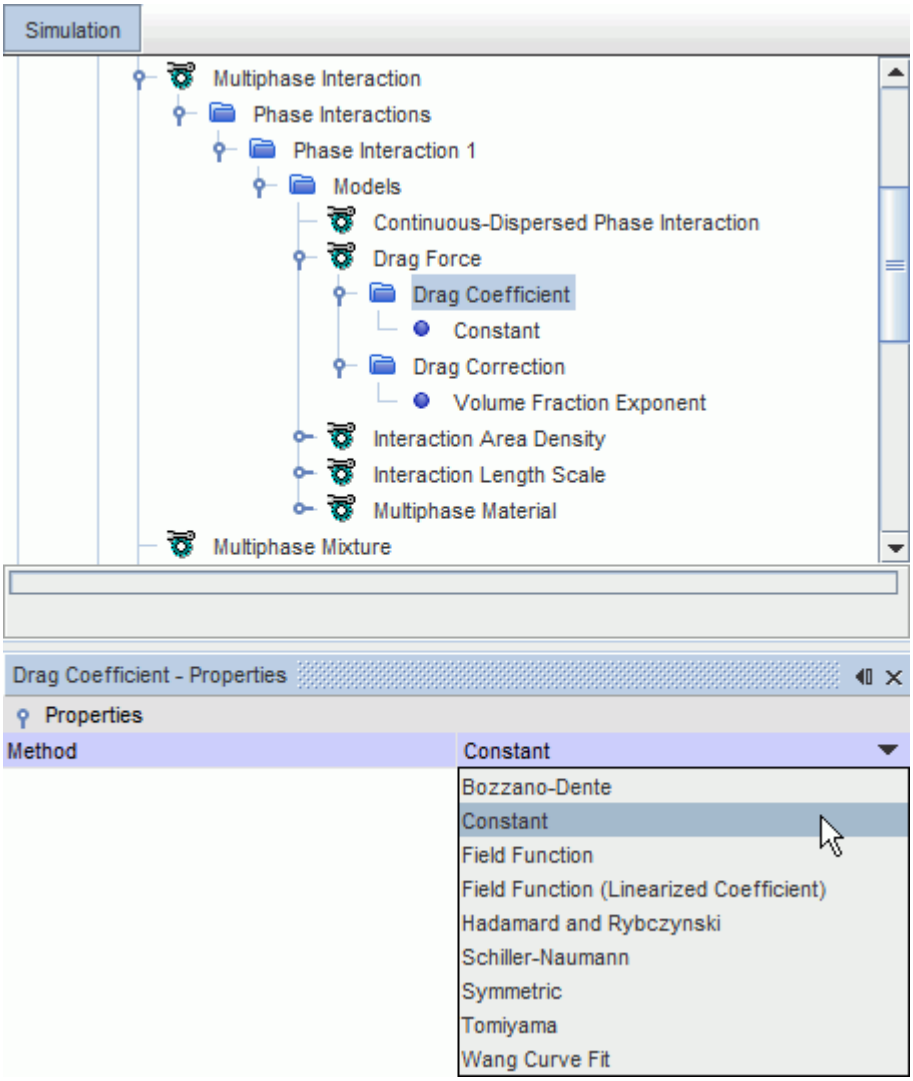
Contents:

What is the Drag Coefficient?
What is the Drag Correction?

What is the Drag Coefficient?

Standard drag coefficients are used for drag force models that are composed of a Standard Drag Coefficient for a single particle together with a Drag Correction factor to account for multi-particle effects. Linearized drag coefficients are used for integrated models for the interphase drag or friction force.

The **Drag Coefficient** node, which has properties and is subject to common menu items, is a sub-node of the **Drag Force** node.



Drag Coefficient Properties

Method	Selects the method to use for specifying the drag coefficient.	
	Arastoopour	Suitable for solid dispersed-phase applications such as fluidized beds. The Arastoopour drag coefficient method node is added. Only available when the dispersed phase is a solid.
	Blended	Applicable only to the interface regime in a Multiple Flow Regime phase interaction. The Blended drag coefficient method node is added.

	Bozzano-Dente	Suitable for a wide range of liquid-gas bubble systems. The Bozzano-Dente drag coefficient method node is added. Only available when the dispersed phase is a gas.
	Constant	Defines the standard drag coefficient using a constant value. A constant method node is added.
	Field Function	Defines the standard drag coefficient ${C_{ij}}^D$ in Eqn. (2250). A field function method node is added as a child to this node.
	Field Function (Linearized Coefficient)	Defines the linearized drag coefficient ${A_{ij}}^D$ in Eqn. (2249). A field function method node is added as a child to this node.
	Gidaspow	Suitable for solid dispersed-phase applications such as fluidized beds. The Gidaspow drag coefficient method node is added. Only available when the dispersed phase is a solid.
	Hamard and Rybczynski	Suitable for spherical liquid droplets. The Hamard and Rybczynski drag coefficient method node is added. Only available when the continuous phase is viscous.
	Linearized Emulsion Topology Inversion	The Linearized Emulsion Topology Inversion drag coefficient method node is added. Only available when the Emulsion Rheology model is selected.
	Rusche-Issa	Suitable for when the dispersed phase volume fraction becomes high. The Rusche-Issa drag coefficient method node is added.

	Schiller-Naumann	Suitable for spherical solid particles, liquid droplets, and small-diameter (spherical) bubbles. The Schiller-Naumann drag coefficient method node is added. Only available when the continuous phase is viscous. However, the continuous phase can be either Newtonian or non-Newtonian.
	Suspension	The Suspension drag coefficient method node is added. Only available when the Suspension Rheology model is selected.
	Syamlal-O'Brien	The Syamlal O'Brien drag coefficient method node is added. Only available when the dispersed phase is a solid.
	Symmetric	Recommended for flows where phase topology inversion occurs. The Symmetric drag coefficient node is added.
	Tomiyama	Suitable for liquid-gas bubble systems with a low Reynolds number and low Morton number flow, for example, small bubbles in an air-water system. In such cases, the Bozzano-Dente model over-predicts the drag coefficient. The Tomiyama drag coefficient node is added. Only available when the dispersed phase is a gas.
	Wang Curve Fit	Suitable for water-air bubble systems or similar systems at normal pressure and temperature. The Wang Curve Fit drag coefficient node is added.

Arastoopour Drag Coefficient

The Arastoopour drag coefficient method is provided for solid dispersed-phase applications such as fluidized beds, where particle loadings are high. See Eqn. (2564).

Arastoopour Drag Coefficient Properties

Exponent

The exponent n in Eqn. (2564).

Blended

Calculates the interface regime drag by blending the drags of the first regime and the second regime on either side of the interface (see Eqn. (2296)).

Bozzano-Dente Drag Coefficient

The Bozzano-Dente drag coefficient method defines the standard drag coefficient using the Bozzano and Dente correlation. See Eqn. (2277).

The Bozzano-Dente correlation is suitable for a wide range of liquid-gas bubble systems including water at low and high pressure, and highly viscous liquids such as some oils, glycerol, or molten glass. For small bubble diameters in low viscosity systems (for example, water), the Bozzano-Dente method gives a different asymptote to other standard drag correlations.

The Bozzano-Dente correlation requires that gravity is activated for the physics continuum.

The surface tension value must be specified in the **Multiphase Material** > **Material Properties** > **Surface Tension** node for the parent phase interaction. The **Surface Tension** node appears automatically when the Bozzano-Dente correlation is selected.

Linearized Emulsion Topology Inversion Drag Coefficient

The Linearized Emulsion Topology Inversion drag coefficient method allows the phases to invert when the volume fraction of the dispersed phase reaches the specified **Inversion Volume Fraction**. When the dispersed volume fraction becomes larger than the specified value, the drag, the relative viscosity, and the normal relative viscosities are swapped between the phases. After inversion, the relative viscosities are calculated using a new value for the maximum packing in the relative viscosity models.

Linearized Emulsion Topology Inversion Drag Coefficient Properties

Inversion Volume Fraction

The volume fraction of the dispersed phase at which the phase properties are inverted: the drag, the relative viscosity, and the normal relative viscosities are swapped between the phases.

Inversion Maximum Packing

The maximum packing value that is used in the relative viscosity models after phase inversion.

Gidaspow Drag Coefficient

The Gidaspow drag coefficient method is appropriate for solid dispersed-phase applications, such as fluidized beds, that have high particle loading.

Gidaspow Drag Coefficient Properties

Exponent

The hindered settling exponent n in Eqn. (2565).

Transition Volume Fraction

When the particle volume fraction is below this value a modified form of Stoke's law is used to compute the drag, as shown in Eqn. (2565). The Ergun equation is used for loadings above this value.

The Gidaspow drag coefficient method reduces to the Wen and Yu correlation when the transition volume fraction is set to 1.0.

Hamard and Rybczynski Drag Coefficient

The Hamard and Rybczynski drag coefficient method is used for viscous Newtonian fluid droplets dispersed in a second immiscible viscous Newtonian fluid. See Eqn. (2283).

The Hamard and Rybczynski drag model is not valid for high Reynolds numbers, where the Schiller-Naumann model is expected to be more accurate.

Rusche-Issa Drag Coefficient

The Rusche-Issa drag coefficient method is appropriate when the dispersed phase volume fraction becomes high. This model accounts for the modification of the drag force due to the presence of other particles when the particles are densely-packed. For example, you would use this model when predicting the separation of oil and water in an emulsion settling experiment. See Eqn. (2273).

STAR-CCM+ includes two implementations of this model. The particular implementation that STAR-CCM+ uses depends on the two phases:

- Both phases liquid:

  STAR-CCM+ applies the Inversion Rusche-Issa drag model. This model allows you to model immiscible liquid (such as oil-water) separation without turning on all the detailed physics of the Emulsion rheology model.

  The **Inversion Rusche-Issa** node and its properties represent this method.
- One liquid phase and one gas or particle phase:

  STAR-CCM+ applies the standard Rusche-Issa drag model. This drag model allows hindered settling in a similar way to the Richardson-Zaki drag correction.

  The **Rusche-Issa** node and its properties represent this method.

The Rusche-Issa drag coefficient has a value that ranges from one at zero dispersed phase volume fraction and increases exponentially [485].

Inversion Rusche-Issa Drag Coefficient Properties

Parameter Options

Specifies the choice of parameters that are used in the Rusche-Issa drag coefficient, Eqn. (2273).

STAR-CCM+ detects the type of dispersed phase and sets the default option accordingly to one of the following:

- **Gas Bubble**
- **Liquid Droplet**
- **Solid Particle**

The alternative option, **User Parameters**, lets you specify your own values of **K1** and **K2**.

Start Volume Fraction (mixture regime), End Volume Fraction (mixture regime)

The start and end values specify the range across which the drag limits are blended.

Inversion Length Scale

Specifies the size of the dispersed liquid drops after inversion.

Rusche-Issa Drag Coefficient Properties

Parameter Options

Specifies the choice of parameters that are used in the Rusche-Issa drag coefficient, Eqn. (2273).

STAR-CCM+ detects the type of dispersed phase and sets the default option accordingly to one of the following:

- **Gas Bubble**
- **Liquid Droplet**
- **Solid Particle**

The alternative option, **User Parameters**, lets you specify your own values of **K1** and **K2**.

Rusche-Issa Drag Coefficient User Parameters

K1, K2

The parameter values (K_1 and K_2 in Eqn. (2273)) are set automatically for the default dispersed phase type. These values are based on the averages that were obtained by Rusche [486].

These parameters are editable only when the **Parameters Options** property setting is **User Parameters**.

Schiller-Naumann Drag Coefficient

The Schiller-Naumann drag coefficient method is used for spherical, rigid particles. See Eqn. (2265).

This model has no properties.

Suspension Drag Coefficient

The Suspension drag coefficient method is a linearized form of the Schiller-Neumann drag model which allows A_D to become infinitely large when maximum packing is reached. The maximum-packed structure prevents slip from occurring between the phases.

This model has no properties.

Syamlal-O'Brien Drag Coefficient

The Syamlal-O'Brien drag coefficient method is based on measurements of terminal velocity that is made in settling beds.

The Syamlal-O'Brien drag coefficient properties should be adjusted according to the parameters of the simulation. For more information and an example case, see the technical report Study of Gas-Solid Fluidized Bed - Tuning Syamlal-O'Brien Drag Model in the Technical Library on the Steve Portal.

Syamlal-O'Brien Drag Coefficient Properties

Transition Volume Fraction

The cut-off volume fraction α_{tr} in Eqn. (2575).

C1

Value that is used as the exponent of ϵ above the cut-off volume fraction in Eqn. (2575).

C2

Value that is used as the multiplier of ϵ below the cut-off volume fraction in Eqn. (2575).

Symmetric Drag Coefficient

The Symmetric drag coefficient method is recommended for flows where phase topology inversion occurs, that is, the dispersed phase in one region of the domain becomes the continuous phase in another. See Eqn. (2274).

This model has no properties, but you specify the appropriate method and value in the **Inverted Topology Length Scale** node.

Tomiyama Drag Coefficient

The Tomiyama drag coefficient method is available only when the continuous phase is viscous, and when the dispersed phase is a gas. It is not available for solid particles. The Tomiyama drag coefficient determines the drag force depending on how contaminated the bulk flow is by impurities, such as surfactants. Surfactants reduce the liquid surface tension and prevent internal circulation in bubbles, hence increasing drag.

The Tomiyama correlation requires that gravity is activated for the physics continuum.

The surface tension value must be specified in the **Multiphase Material** > **Material Properties** > **Surface Tension** node for the parent phase interaction. The **Surface Tension** node appears automatically when the Tomiyama correlation is selected.

Tomiyama Drag Coefficient Properties

Contamination State

Selects the method to use for specifying contamination of the bulk flow by impurities, such as surfactants. Surfactants reduce the liquid surface tension and prevent internal circulation in bubbles, hence increasing drag. The **Contaminated** water option is the default option since it applies to most industrial cases.

- **Pure**
  Uses the drag coefficient that is described by Eqn. (2287).
- **Moderate**
  Specifies a higher drag coefficient than the **Pure** option, before bubble deformation effects become dominant. See Eqn. (2288).
- **Contaminated**
  Specifies a higher drag coefficient than the **Moderate** and **Pure** options to account for the presence of surfactants. See Eqn. (2289).

Wang Curve Fit Drag Coefficient

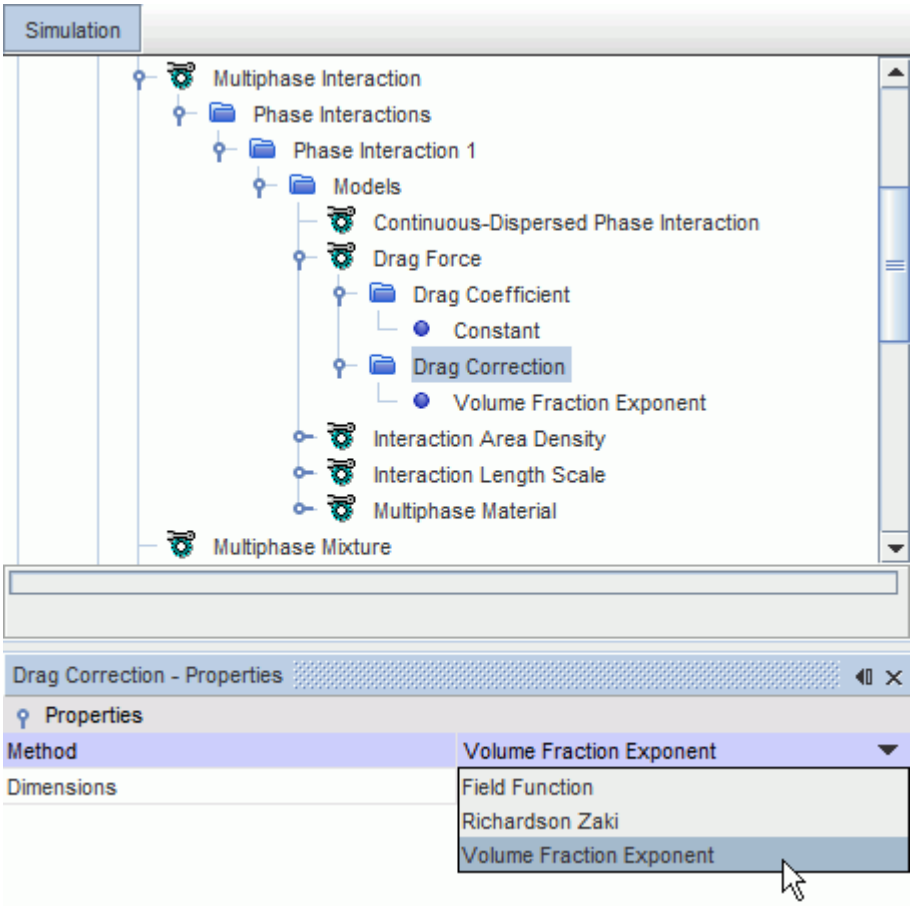
The Wang Curve Fit drag coefficient method is available only when the continuous phase is viscous, and when the dispersed phase is a fluid. It is not available for solid particles. This correlation is used for concentrations of air bubbles in water at near atmospheric pressure, and should be used with care in any other conditions. See Eqn. (2292).

This model has no properties.

What is the Drag Correction?

The Drag Correction model describes how concentration modifies the single-particle Drag Coefficient model in a multiparticle system.

The **Drag Correction** node, which has properties and is subject to common menu items, is a sub-node of the **Drag Force** node.



Drag Correction Properties

Dimensions	The dimensionality of the drag correction (read-only).	
Method	Selects the method to use for specifying the drag correction.	
	Field Function	Defines the drag correction using a field function. A field function method node is added.
	Lockett Kirkpatrick	This method is used for small ellipsoidal air/water bubbles with a size range in the order of 5 mm. The Lockett Kirkpatrick drag correction node is added.
	Richardson Zaki	This method is the recommended option for undeformable spherical particles. The Richardson Zaki drag correction node is added.

	Simonnet	This method is used for air / water bubbles in the 7-10 mm size range. The Simonnet drag correction node is added.
	Volume Fraction Exponent	This method is suitable for spherical solid particles, liquid droplets, and small-diameter (spherical) bubbles. The Volume Fraction Exponent drag correction node is added. This correlation is available only when the continuous phase is viscous.

Lockett Kirkpatrick

The Lockett Kirkpatrick drag correction is a method for small ellipsoidal bubbles. It applies to air/water systems with bubble size of order 5 mm (Eotvos number around 3.3). The effect of increasing concentration is hindering.

Lockett Kirkpatrick Properties

Maximum Packing

This setting limits the maximum packing, $\alpha_{d,max}$, in Eqn. (2303).

Richardson Zaki

The Richardson Zaki drag correction is a method for spherical particles, droplets, or bubbles. According to this model, the effect of increased concentration on spherical particles is that of hindering. The terminal velocity decreases faster than would be expected purely from the reduction of buoyancy force of the two-phase mixture on the particles.

This model can also cover droplets that are small enough to retain their spherical shape and whose internal viscosity is high enough for internal circulation to be negligible. This model also applies to bubbles whose surfaces are immobile due to surface contamination, as in the case of air bubbles in regular tap water.

Richardson Zaki Properties

Maximum Packing

This setting limits the maximum packing, $\alpha_{d,max}$, in Eqn. (2298).

Calibration Constant A, B, C, D, N1, N2

The calibration constants A, B, C, D, n_1, and n_2 in Eqn. (2299).

Simonnet

The Simonnet drag correction is used for air / water bubbles in the 7-10 mm size range (Eotvos number 6.6-13.4) and 0.0-0.3 volume fraction of dispersed/secondary phase range. Effects are hindering in the range 0.0-0.15 and swarming in the range 0.15-0.3.

This method has no properties.

Volume Fraction Exponent

Defines the drag correction using the Volume Fraction Exponent.

Volume Fraction Exponent Properties

Volume Fraction Exponent (continuous)

Sets the value of n_v. It is included to make sure that the drag tends to zero as the amount of the continuous phase material tends to zero, given by Eqn. (2311).

Maximum Packing

This setting limits the maximum packing, $\alpha_{d,max}$, given by Eqn. (2312).

Modeling Lift Force

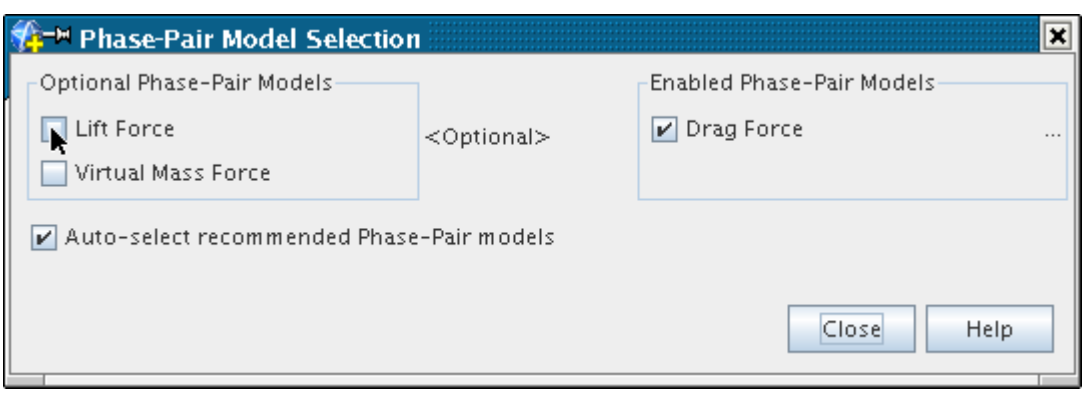
This section describes how to select the Lift Force model and set the appropriate properties.

This section explains the following about the Lift Force phase interaction model:

- Selecting the Lift Force Model
- What Is the Lift Force?
- What Is the Lift Coefficient?
- What Is the Lift Correction?

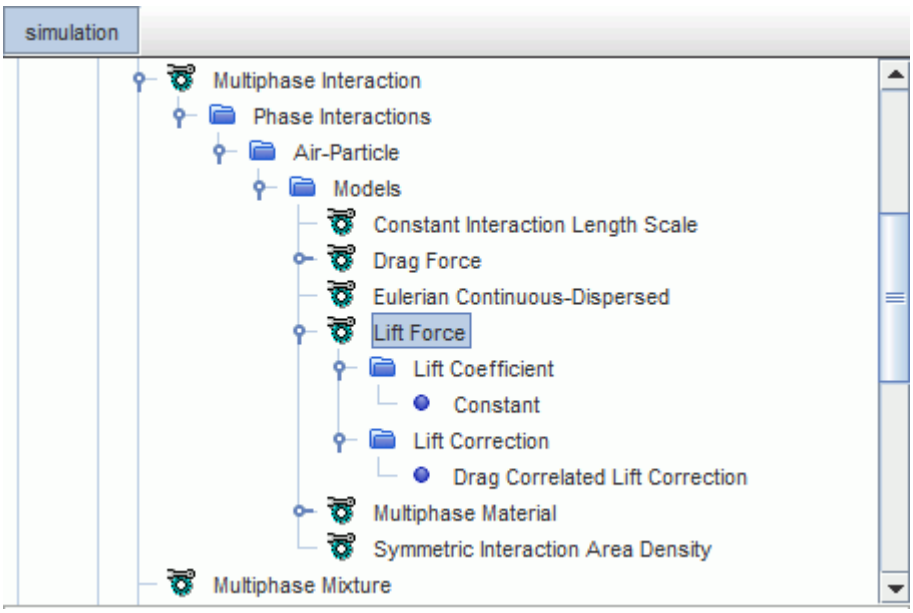
Selecting the Lift Force Model

Select **Lift Force** from the *Optional Phase Interaction Models* group box in the *Phase Interaction Model Selection* dialog as shown.



What Is the Lift Force?

The **Lift Force** node, which is subject to common menu items, represents the Lift Force model for calculating interphase momentum transfer. This models the perpendicular lift force that a particle can experience in a non-uniform or swirling flow field.



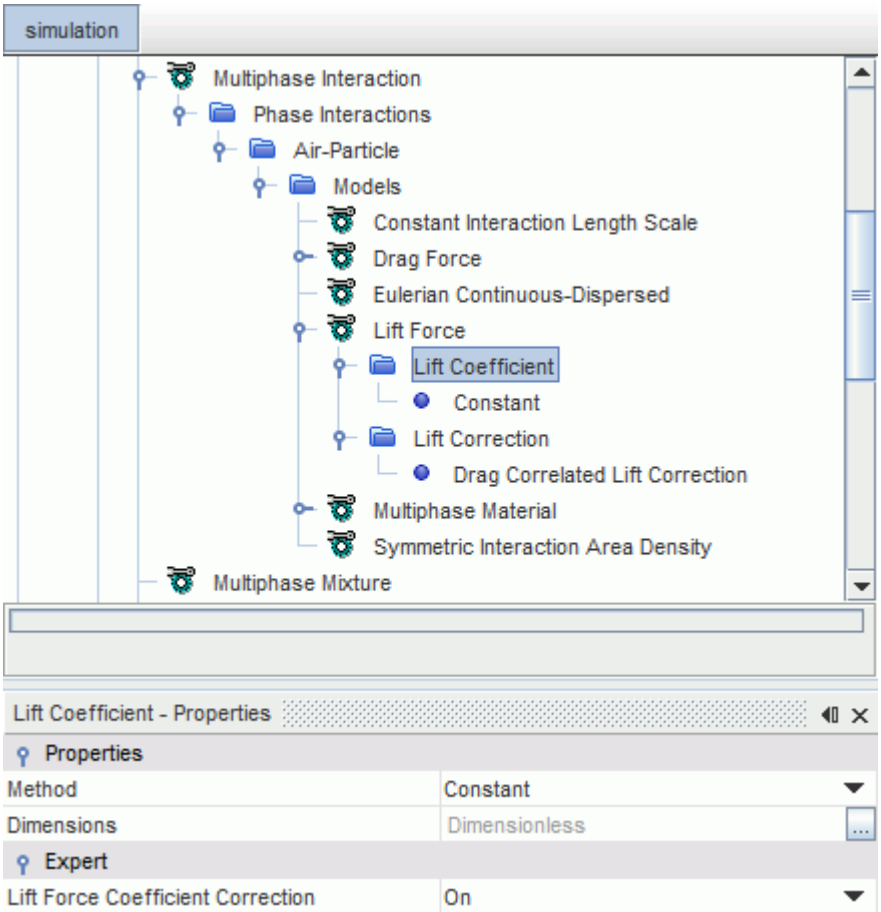
The Lift Force model requires a Lift Force Coefficient.

Lift Force Properties

None.

What Is the Lift Coefficient?

The **Lift Coefficient** node, which has properties and is subject to common menu items, is a sub-node of the **Lift Force** node. This value supplies the lift coefficient in Eqn. (2334).



The lift coefficient can be set as a constant value, or by a field function, or by Tomiyama method.

Lift Coefficient Properties

Dimensions	Dimensionless.	
Method	Selects the method to use for specifying the lift coefficient.	
	Constant	Defines the lift coefficient using a constant value. A constant method node is added as a child to this node.
	Field Function	Defines the lift coefficient using a field function. A field function method node is added as a child to this node.
	Tomiyama	Defines the standard lift coefficient using the Tomiyama correlation. It is only available when the continuous phase is viscous, and when the dispersed phase is a gas. Additionally, the correlation

		requires that gravity is activated for the physics continuum. Specify the surface tension value in the **Multiphase Material** > **Material Properties** > **Surface Tension** node for the parent phase interaction. The **Surface Tension** node appears automatically when the correlation is selected.

Lift Coefficient Expert Properties

Lift Force Coefficient Correction	**On**	Activates the lift correction. For more information, see What Is the Lift Correction?.
	Off	Deactivates the lift correction.

If the Drag Force Coefficient Type is set to Standard (the default setting), the Lift Force Coefficient Correction option is activated by default when you select the Lift Force model. However, if the Drag Force Coefficient Type is set to Linearized, the Lift Force Coefficient Correction property is deactivated and cannot be set.

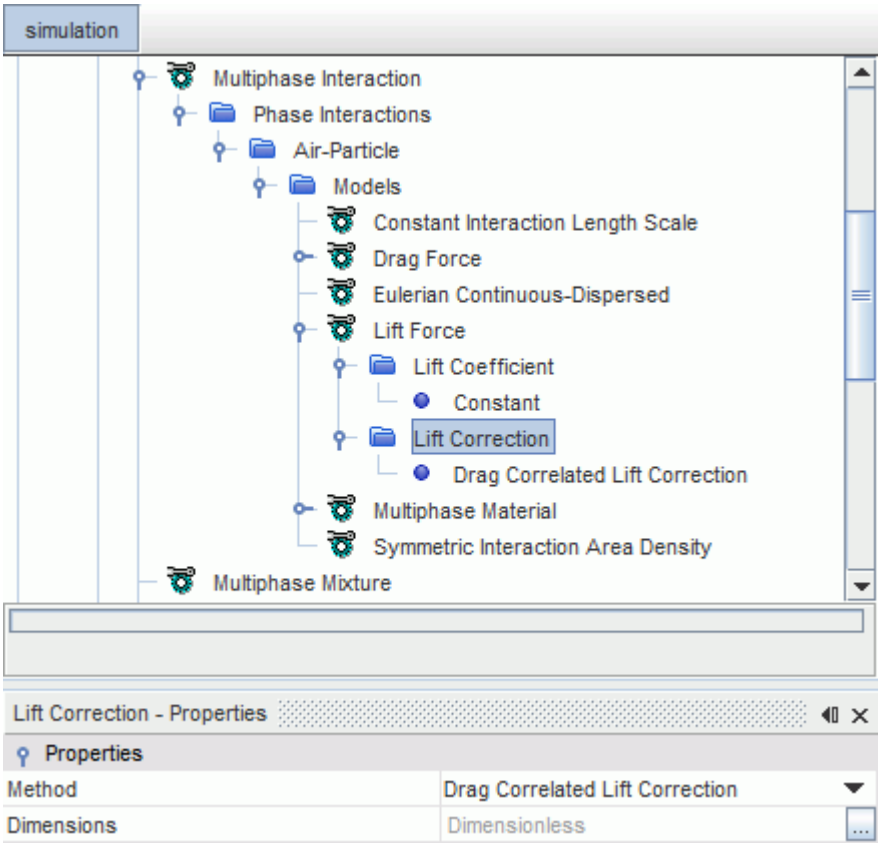
What Is the Lift Correction?

The Lift Correction is used in the Lift Force model to improve numerical stability in the simulation.

When you use drag correction in a simulation where the Lift Force model is also used, the reduction of drag force (see Eqn. (2246) and Eqn. (2249)) can render the simulation numerically unstable. To resolve this, the lift correction is used to account for multi-particle effects by modifying the single-particle Lift Coefficients.

The lift correction can be approximated by the drag correction, or specified by a field function. For more information, see Lift Force.

The **Lift Correction** node, which has properties and is subject to common menu items, is a sub-node of the **Lift Force** node.



Lift Correction Properties

Dimensions	Dimensionless.	
Method	Selects the method to use for specifying the lift correction.	
	Drag Correlated Lift Correction	Uses the drag coefficient correction to approximate the lift coefficient correction. This approximation is useful for cases when swarming is modeled.
	Field Function	Defines the lift correction using a field function. A field function method node is added as a child to this node.

Modeling Wall Lubrication Force

This section describes how to select the Wall Lubrication Force model and set the appropriate properties.

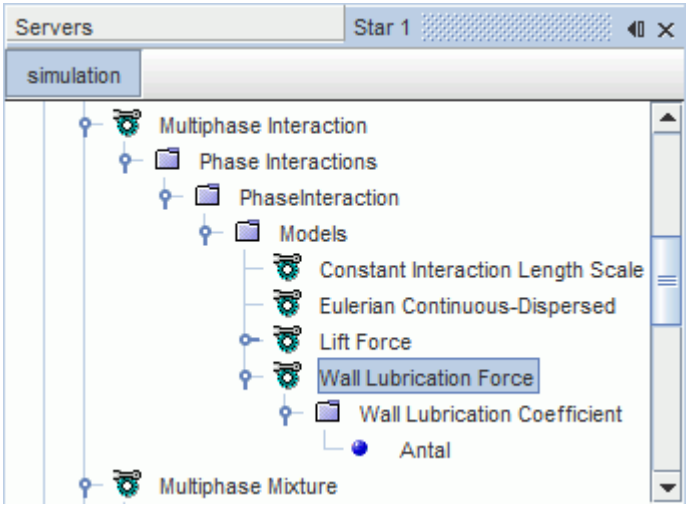
This section explains the following about the Lift Force phase interaction model:

- What Is the Wall Lubrication Force?
- What Is the Wall Lubrication Coefficient?

- Selecting the Wall Lubrication Force

What Is the Wall Lubrication Force?

The **Wall Lubrication Force** node represents the Wall Lubrication model for calculating interphase momentum transfer. This models the force that a rising bubble experiences that prevents it from touching the wall.

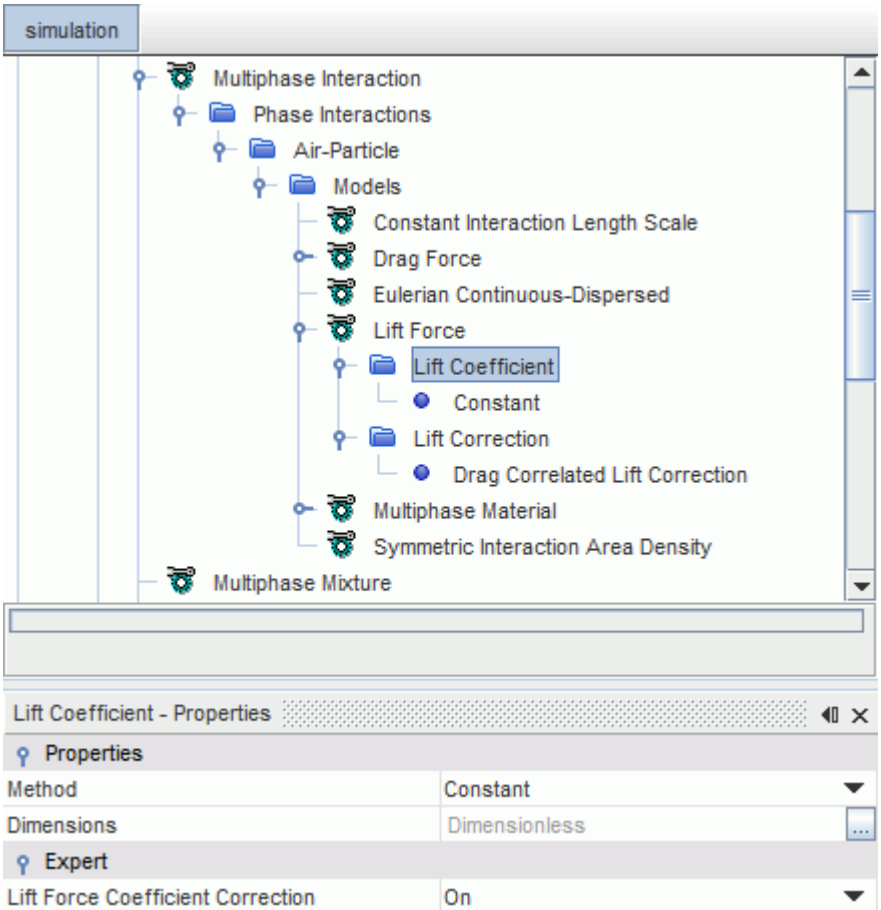


Wall Lubrication Force Properties

None.

What Is the Wall Lubrication Coefficient?

The **Wall Lubrication Coefficient** node, which has properties and is subject to common menu items, is a sub-node of the **Wall Lubrication Force** node. This supplies the wall lubrication coefficient in Eqn. (2348).



The wall lubrication coefficient can be set using the Antal equation or by using a field function.

Wall Lubrication Properties

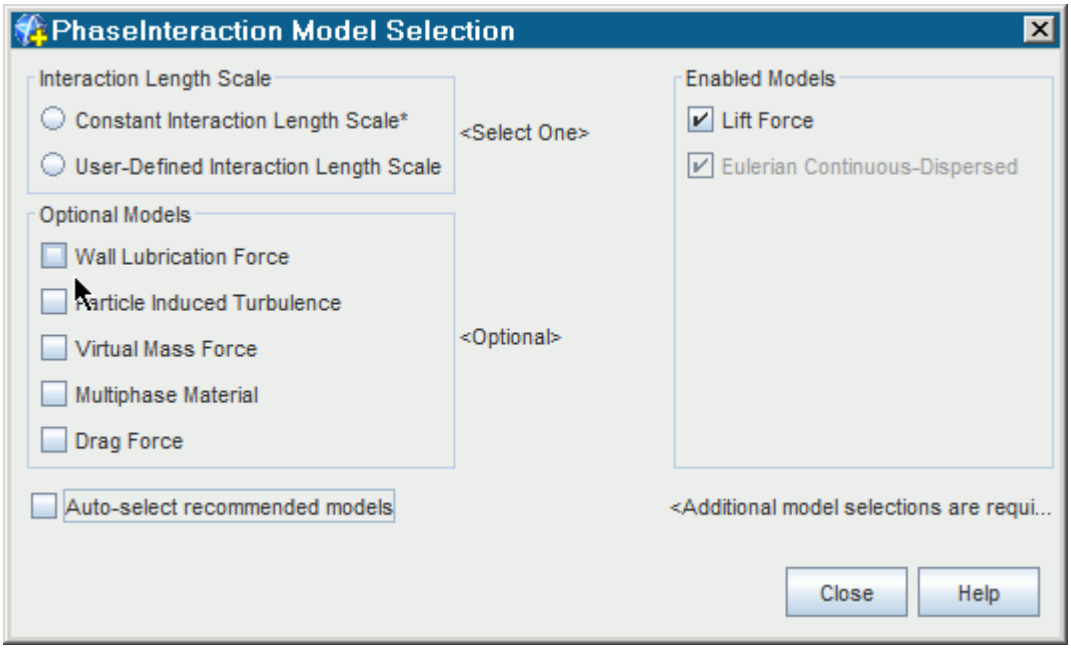
Dimensions	The dimensionality of the wall lubrication coefficient (read-only).	
Method	Selects the method to use for specifying the wall lubrication coefficient.	
	Antal	Defines the standard lift coefficient using the Antal correlation. An **Antal** node is added as a child to this node where you can specify the coefficients for C_{w1} and C_{w2} in Eqn. (2350). This is the default setting.
	Field Function	Defines the lift coefficient using a field function. A field function method node is added as a child to this node where you can add a function for C_{WL} in Eqn. (2348).

Selecting the Wall Lubrication Force

To select the Wall Lubrication Force model:

- Select **Lift Force** from the *Optional Phase Interaction Models* group box in the *PhaseInteraction Model Selection* dialog.

The **Wall Lubrication Force** becomes available in the *Optional Phase Interaction Models* group box as shown.

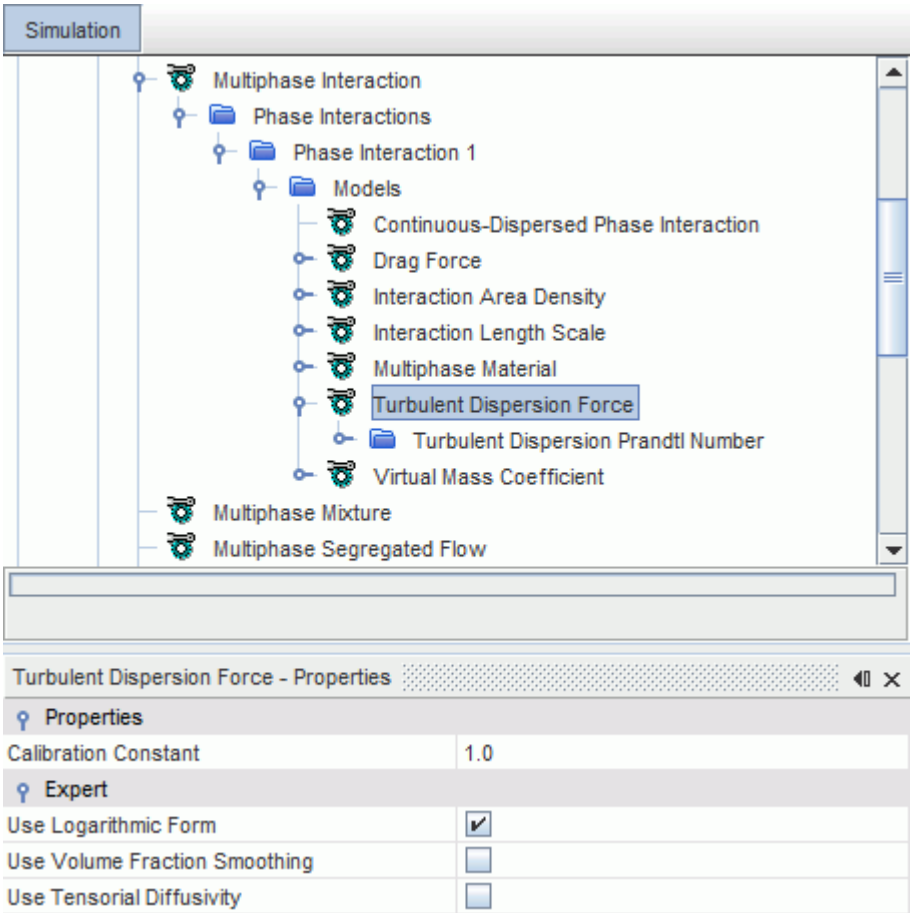


Modeling Turbulent Dispersion Force

Particles in a turbulent flow experience a greater level of dispersion than in a laminar flow. The Turbulent Dispersion Force model accounts for this interaction.

What is the Turbulent Dispersion Force?

The Turbulent Dispersion Force node represents the Turbulent Dispersion Force model that accounts for the interaction between the dispersed phase and the surrounding turbulent eddies. A turbulence model must be selected before you can apply this option.



Turbulent Dispersion Force Properties

Calibration Constant	C_0 in Eqn. (2341), with a default value of 1.

Turbulent Dispersion Force Expert Properties

Use Logarithmic Form	Analytically, the logarithmic form (Eqn. (2340)) is equivalent to the basic form (Eqn. (2339)), but it has a much simpler discretization on a collocated grid. This simplicity makes it more reliable whenever extreme volume fractions occur in neighboring cells -- for example, due to condensation or separation processes. In such applications, this reliability in turn improves convergence history.	
	Activated	Use Eqn. (2340), the default.
	Deactivated	Use Eqn. (2339).
Use Volume Fraction Smoothing	This option reduces numerical checkerboarding in the presence of dominant turbulent dispersion forces and hence improves convergence. It does not affect the converged solution.	
	Activated	Use smoothing.
	Deactivated	No smoothing, the default.

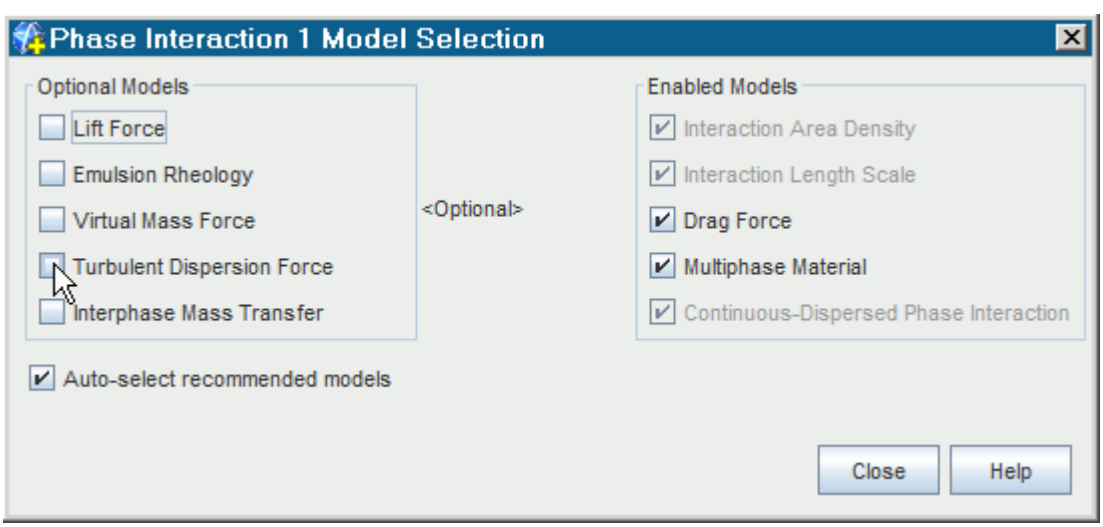
Use Tensorial Diffusivity	Where a Reynolds Stress model has been selected for the continuous phase, activating this option makes the simulations use an anisotropic model for turbulent dispersion (Eqn. (2342)). If this option is not active, or if a turbulence model based on the eddy viscosity hypothesis is selected for the continuous phase, then the simulation uses the isotropic model (Eqn. (2341)) instead.	
	Activated	Use Eqn. (2342).
	Deactivated	Use Eqn. (2341), the default.

Turbulent Dispersion Force Prandtl Number

Dimensions	The dimensionality of the Prandtl number (read only).	
Method	Selects the method to use for specifying the Prandtl number, σ_α in Eqn. (2344).	
	Constant	Takes a default value of 1, as for a passive scalar.
	Field Function	Defines the Prandtl number using a field function. A field function method node is added as a child to this node.
	Inertial Correction	Recommended for large particles or bubbles.
	Tchen Correction	Recommended for small particles or bubbles.

Selecting the Turbulent Dispersion Force

Select **Turbulent Dispersion Force** from the *Optional Models* group box in the *Phase Interaction Model Selection* dialog as shown.

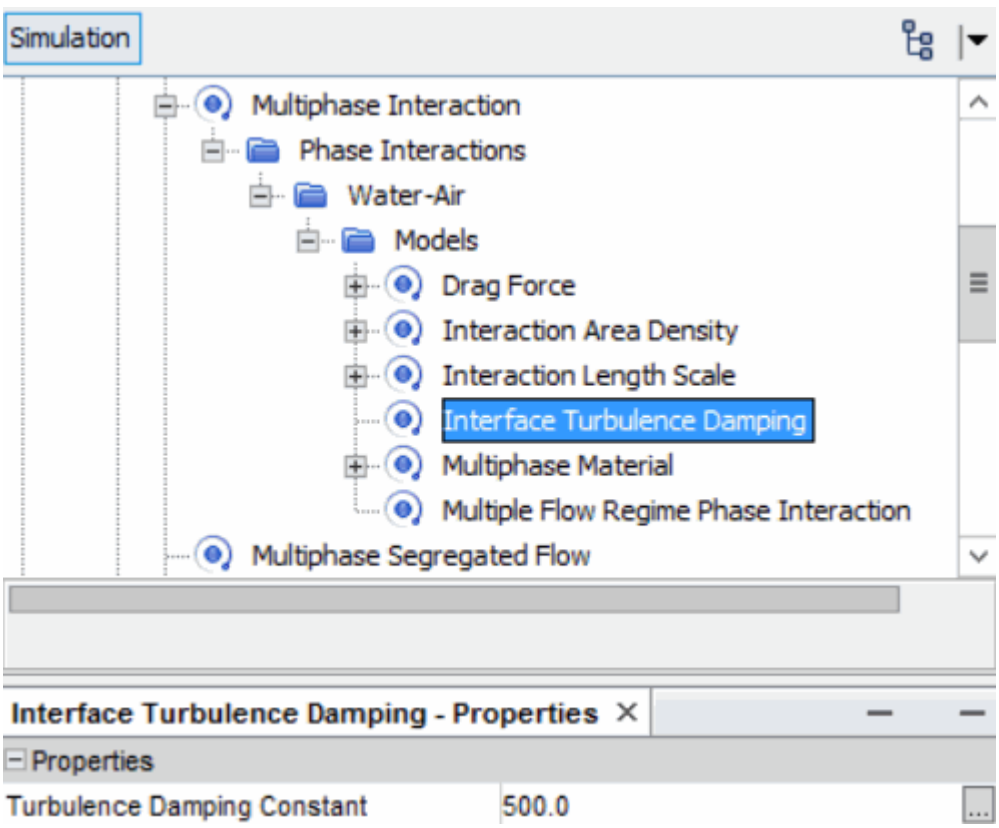


If **Auto-select Recommended Models** is activated, the Virtual Mass Coefficient model is also selected. This model is required when you use the Turbulent Dispersion Force model because the virtual mass coefficient is used in the Inertial and Tchen Correction methods for Turbulent Dispersion Force Prandtl Number. This model is also required so that the virtual mass coefficient becomes available for user modeling with the Field Function method.

Modeling Interface Turbulence Damping

The Interface Turbulence Damping model is currently compatible with the K-Omega turbulence model only. In the vicinity of large scale interface, a boundary layer develops and the interface must be treated as a moving wall. The Interface Turbulence Damping model achieves this by using a low Re treatment to estimate Omega (the specific dissipation rate) near the large scale interface.

The Interface Turbulence Damping model is selected automatically if a turbulence model is activated in the physics continuum. You specify the appropriate turbulence damping constant.



Interface Turbulence Damping Properties

Turbulence Damping Constant	Specifies the value B in Eqn. (2558). The default value is 500.

Modeling Virtual Mass Force

The Virtual Mass Force model accounts for the additional resistance that is experienced by a particle accelerating through a fluid. This resistance can be significant when the particle has a density that is comparable to, or much smaller than, the surrounding fluid. Example applications include short bubble columns or applications with swirl-promoting devices.

Virtual Mass Force Model Reference

Theory	See Virtual Mass	
Provided By	**[phase interaction]** > **Models** > *Optional Models*	
Example Node Path	**[phase interaction]** > **Models** > **Virtual Mass Force**	
Requires	Physics continuum selections: *Material*: **Eulerian Multiphase** *Eulerian Multiphase Model*: **Multiphase Segregated Flow** Automatically activates: **Multiphase Interaction**, **Multiphase Equation of State**, **Gradients**. A phase interaction is required. Phase interaction selections: *Phase Interaction Topology*: **Continuous-Dispersed Phase Interaction** *Optional Models*: **Virtual Mass Force** Also required selection in phase interaction: **Virtual Mass Coefficient**.	
Activates	Field Functions	Virtual Mass Force of [phase interaction] This field function becomes available when the **Interaction Force Storage Retained** property is activated in the Multiphase Segregated Flow solver.
Properties	None.	
Actions	None.	

What is the Virtual Mass Coefficient?

The virtual mass coefficient scales the virtual mass force in the particle equation of motion.

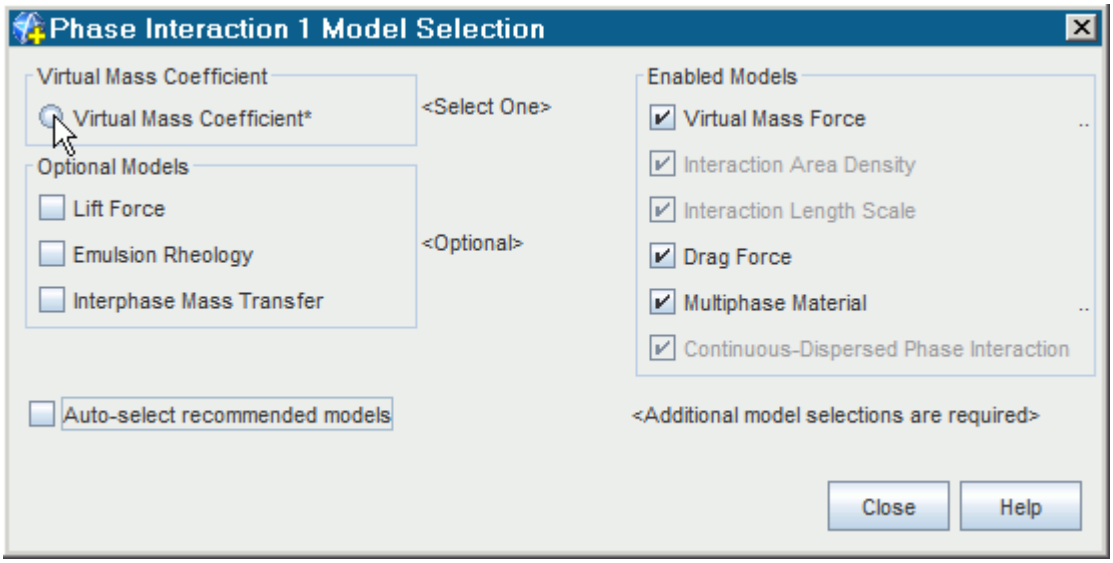
The Virtual Mass Coefficient model defines the virtual mass coefficient that appears in Eqn. (2315) for the Virtual Mass Force, relevant for accelerating dispersed flows. The same coefficient also influences the interaction of particles with turbulence, even when the mean flow is not accelerating.

The Virtual Mass Coefficient model is required for the following models:

- Virtual Mass Force
- Turbulent Dispersion Force
- Troshko and Hassan Particle Induced Turbulence
- Tchen Particle Induced Turbulence

- Tchen Turbulent Response

The Virtual Mass Coefficient model is selected automatically when any of these models is selected, except for the Tchen Turbulent Response since that is a phase model. If the Virtual Mass Coefficient model is deselected, you can select it again from the *Virtual Mass Coefficient* group box.



Virtual Mass Coefficient Properties

Dimensions	The dimensionality of the virtual mass coefficient (read-only).	
Method	Selects the method to use for specifying the virtual mass coefficient.	
	Constant	Defines the virtual mass coefficient using a constant value.
	Field Function	Defines the virtual mass coefficient using a field function.
	Spherical Particle	Defines the virtual mass coefficient using the Spherical Particle method. This option is the default method.
	Zuber	Defines the virtual mass coefficient using the Zuber method.

Spherical Particle Properties

Free Stream Minimum Fraction	Defines the freestream fraction that appears in Eqn. (2321). The default value is 0.0

Zuber Properties

Free Stream Minimum Fraction	Defines the freestream fraction that appears in Eqn. (2325). The default value is -10.0

Modeling Virtual Mass Stress

The Virtual Mass Stress model models the effect of turbophoresis in bubbly flows. This model complements the Virtual Mass Force model, which covers only the effect of mean flow acceleration.

The Virtual Mass Stress ($\mathbf{F}_{vms}$) is the term which arises from Reynolds averaging of the instantaneous virtual mass force (see Eqn. (2331)). The main application of the Virtual Mass Stress model is in turbulent bubbly flows simulated with full Reynolds Stress turbulence models, where it moderates the sensitivity of large bubbles to turbulent fluctuations.

The Virtual Mass Stress model can also be used to improve the behavior of bubbly flows simulated with K-epsilon or K-omega models with the *Normal Stress Term* activated. However, if you find that the effect of the *Normal Stress Term* is significant, then a full Reynolds Stress approach is likely to be more realistic and is recommended.

Virtual Mass Stress Model Reference

Provided By	**[phase interaction]** > **Models** > *Optional Models*	
Example Node Path	**[phase interaction]** > **Models** > **Virtual Mass Stress**	
Requires	A phase interaction is required. Phase interaction selections: *Phase Interaction Topology*: **Continuous-Dispersed Phase Interaction** *Optional Models*: **Virtual Mass Stress**	
Activates	Physics Models	Automatically activates **Virtual Mass Coefficient** and **Virtual Mass Force** in the phase interaction.
Properties	None.	
Actions	None.	

Recommendations

The Virtual Mass Stress model is normally used together with one of the Reynolds Stress turbulence models.

If a Reynolds Stress Turbulence model is selected for the continuous phase, and a Turbulence Response model is selected for the dispersed phase, then the *Normal Stress Term* property should be activated on the Turbulence Response model.

The Virtual Mass Stress model can also be activated together with K-epsilon or K-omega turbulence models, although this is not recommended. For reasons of balance, whichever combination of turbulence models you use with Virtual Mass Stress, you should activate the *Normal Stress Term* property either on all of the models

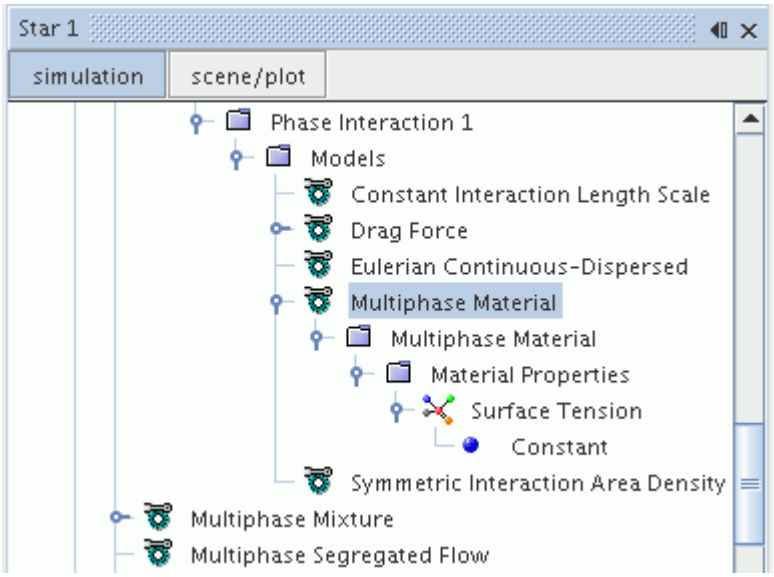
or on none of the models. In this context, full Reynolds Stress models always include normal stress terms in the momentum equation (unless the *Use Boussinesq Approximation* property is activated for initialization purposes or for diagnostic reasons).

Providing Additional Material Properties

The **Multiphase Material** node is used to hold any additional material property nodes that are required by the phase interaction models.

Currently, two more property nodes can be added, as follows:

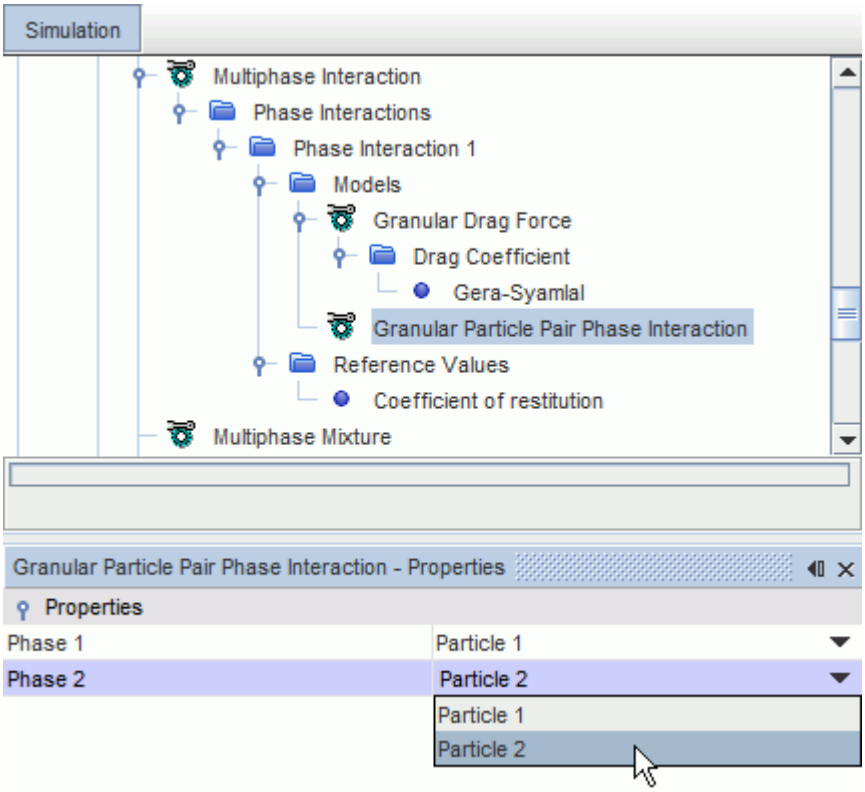
- **Surface Tension** – required when the Bozzano-Dente drag law, Tomiyama drag law, or Tomiyama lift law have been selected.
- **Latent Heat** – required when the bulk boiling or wall boiling models have been selected.



What Is the Granular Particle Pair Phase Interaction Model?

If you have activated the Granular Pressure model and have created more than one particle, it is necessary to define an interaction between the two particle phases.

This interaction is defined on the properties of the **Granular Particle Pair Phase Interaction** model node.



Granular Particle Pair Phase Interaction Properties

Phase 1	Choose a particle phase of the interacting pair from the drop-down list of previously defined Eulerian phases. The list contains the particle phases only: you cannot select another type of phase.
Phase 2	Choose a different particle phase of the interacting pair than you defined in **Phase 1**.

Granular Particle Pair Phase Interaction Reference Value

Coefficient of restitution	Defines the coefficient of restitution for collisions between particles in the phase interaction. This value, e, is used in the kinetic regime formulation. The default value of e = 0.9.

Once the Granular Particle Pair Phase Interaction model has been chosen, select the models for:

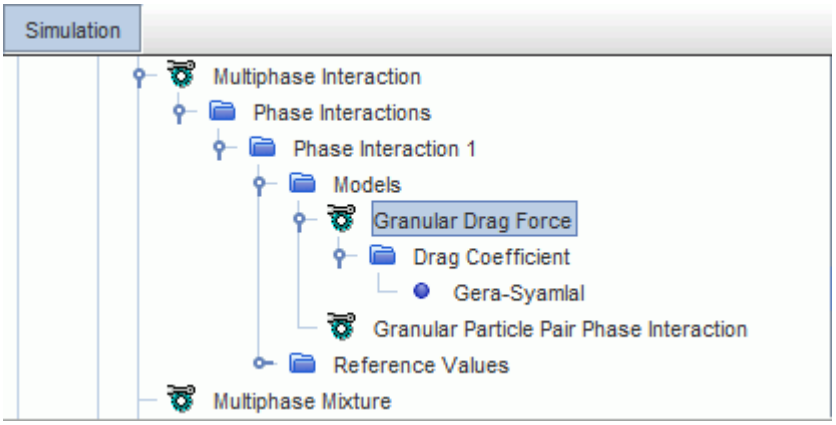
- Granular Drag Force

Modeling Granular Drag Force

This section describes the Granular Drag Force model that is used in particle-particle phase interactions and explains how to select it.

What is the Granular Drag Force?

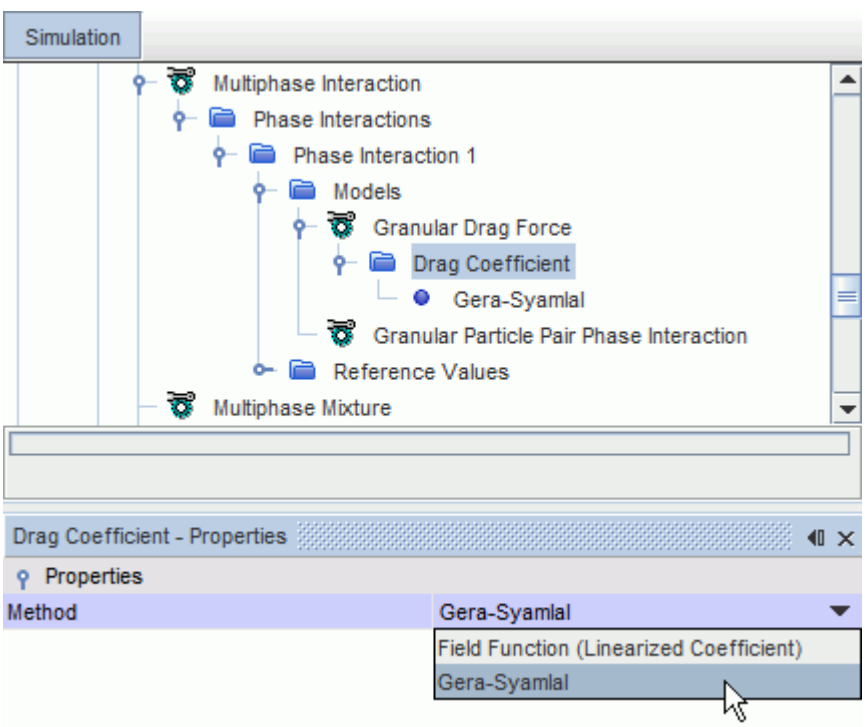
The **Granular Drag Force** node represents the Granular Drag Force model. It is related to the Drag Force model for continuous-dispersed phases.



In STAR-CCM+, the Granular Drag Force is modeled as a linear multiplier, A_D in Eqn. (2246), of the relative velocity between two phases.

What Is the Drag Coefficient?

The drag coefficient is used as a direct multiplier of the relative velocity between phases in Eqn. (2246). The **Drag Coefficient** node, which has properties and is subject to common menu items, is a sub-node of the **Granular Drag Force** node.



Drag Coefficient Properties

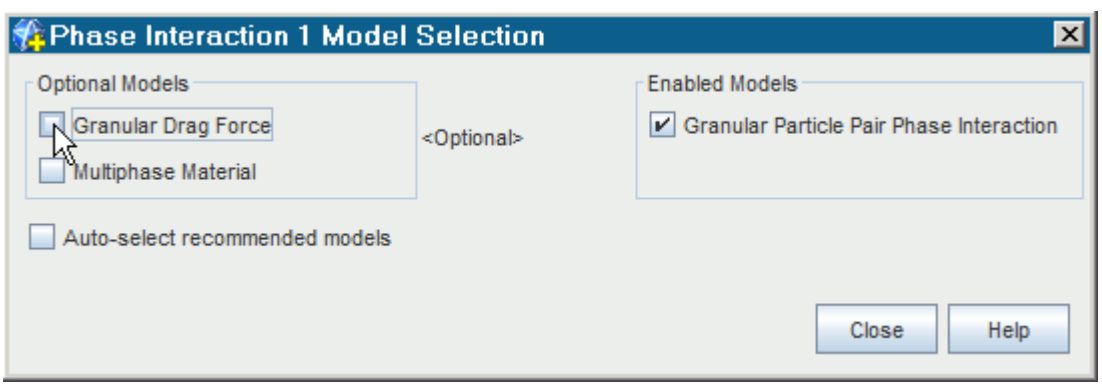
Method	Selects the method to use for specifying the drag coefficient.	
	Field Function (Linearized Coefficient)	Defines the linearized drag coefficient using a field function. A field function method node is added as a child to this node.
	Gera-Syamlal	Defines the linearized drag coefficient using the Gera-Syamlal granular drag force model. The Gera-Syamlal node is added. This node lets you define the coefficient of friction for the phase interaction.

Gera-Symlal Properties

Coefficient of Friction	Defines the coefficient of friction between particles for the phase interaction. This value, c_f , is used in Eqn. (2623).

Selecting the Granular Drag Force

The **Granular Drag Force** is selected automatically when the **Granular Particle Pair Phase Interaction** model is selected. If the Granular Drag Force model has been deselected, it can be selected from the *Optional Models* group box as shown.



Modeling Multiple Flow Regimes

The Multiple Flow Regime Phase Interaction model is used for cases where both segregated and dispersed two-phase flows exist in the same domain.

An example scenario is a flow of a liquid and gas under undulating motion. In this scenario, it is possible to have three concurrent regions of two-phase flow:

- The first regime: A region where the gas is dispersed in the liquid.
- The interface regime: A region where, of the two phases, neither phase is dispersed in the other phase.
- The second regime: A region where the liquid is dispersed in the gas.

In the interface regime, the interface between the two phases can be modeled as separated or blended. If the two phases are clearly separated by a sharp interface, the **Large Scale Interface Detection** model is used to determine the interface. If there is no clear separation of the two phases in the interface regime, a blended interface is modeled.

The following interphase transfer models are available for multiple flow regime phase interactions:

- Drag Force
- Lift Force
- Surface Tension Force
- Turbulent Dispersion Force
- Interface Turbulence Damping
- Interphase Energy Transfer
- Large Scale Interface Detection

The process for setting up a Multiple Flow Regime phase interaction for a basic simulation is as follows:

1. Set up an Eulerian Multiphase simulation with the appropriate gas or liquid phases.
2. Create a phase interaction and activate the **Multiple Flow Regime Phase Interaction** model.
3. In the *Phase Interaction Model Selection* dialog, activate the appropriate models from the *Optional Models* group box.
4. Select the **Multiple Flow Regime Phase Interaction** node and set the required properties:
   a) Define the primary and secondary phases of the phase interaction.
   b) Define the first regime terminus and the second regime onset.
5. Select the **Drag Force** node and set the appropriate drag coefficients:
   a) Define the type of drag for the first regime and second regime.
   b) Define the type of drag for the interface regime.

      If the two phases are to be modeled as separated in the interface regime, you are advised to use the Strubelj and Tiselj interface drag method. You also set the appropriate relaxation time factor for this interface drag method. If there is no clear separation of the two phases in the interface regime, you are advised to use the Blended interface drag method.
6. Select the **Interaction Length Scale** node and set the interaction length scale values for the first regime and the second regime.
7. Select the **Interaction Area Density** node and set the interaction area density methods for the first regime and the second regime.
8. If applicable, select the **Interphase Energy Transfer** node and set the Nusselt number for each regime.

The following steps are optional, and apply only if a clear separation of the two phases in the interface regime is modeled. That is, the Blended interface drag method is not selected.

9. Select the **Large Scale Interface Detection** node and set the large scale interface detection properties.

10. Select the **Interface Turbulence Damping** node and set the turbulence damping constant.
11. Select the **Surface Tension Force** node and set the surface tension coefficient.

Contents:

Multiple Flow Regime Phase Interaction Reference
Modeling Surface Tension in Multiple Flow Regimes

Multiple Flow Regime Phase Interaction Reference

The Multiple Flow Regime phase interaction model allows you to model dispersed and segregated two-phase flows within a single framework.

Multiple Flow Regime Phase Interaction Reference

Theory	See Multiple Flow Regimes.	
Provided By	**[phase interaction]** > **Models** > *Phase Interaction Topology*	
Example Node Path	**[phase interaction]** > **Models** > **Multiple Flow Regime Phase Interaction**	
Requires	A Eulerian Multiphase simulation with the following models activated: *Material*: **Eulerian Multiphase** *Eulerian Multiphase Model*: **Multiphase Segregated Flow** A minimum of two phases: • For each phase: *Material*: **Gas**, **Liquid**, **Multi-Component Gas**, or **Multi-Component Liquid** A new phase interaction.	
Properties	Key properties are: *Primary Phase*, *Secondary Phase*, *First Regime Terminus*, *Second Regime Onset*. See Multiple Flow Regime Phase Interaction Properties.	
Activates	Physics Models	The following models are activated automatically: Drag Force Interaction Length Scale Interaction Area Density Multiphase Material Interface Turbulence Damping (for a Turbulent viscous regime only) The following models are available in the *Optional Models* group box: Interphase Energy Transfer (when an energy model is activated in the physics continuum) Interphase Mass Transfer (when an energy model is activated in the physics continuum) Large Scale Interface Detection Lift Force Surface Tension Force

		Turbulent Dispersion Force (for a Turbulent viscous regime only)

Multiple Flow Regime Phase Interaction Properties

Primary Phase

The primary phase of the interacting pair of Eulerian phases.

Secondary Phase

The secondary phase of the interacting pair of Eulerian phases.

First Regime Terminus

The first regime terminus (α_{fr}) is the value of the volume fraction of the secondary phase, α_s, at which the first regime transits to the interface regime. The default value is 0.3.

In the first regime, the primary phase is treated as the continuous phase.

Second Regime Onset

The second regime onset (α_{sr}) is the value of α_s at which the interface regime transits to the second regime. The default value is 0.7.

In the second regime, the secondary phase is treated as the continuous phase.

Drag Force

When you use a Multiple Flow Regime phase interaction, you set the appropriate Drag Coefficient method for each phase interaction regime.

For the first regime and the second regime, the available methods are:

Schiller-Naumann

The drag coefficient is computed using the Schiller and Naumann correlation.

Tomiyama

The drag coefficient is computed using the Tomiyama correlation.

Available for the first regime when the secondary phase is a gas and available for the second regime when the primary phase is a gas.

For the interface regime, the available methods are:

Blended

The interface regime drag is calculated by blending the drags of the first regime and the second regime on either side of the interface (see Eqn. (2296)).

This method is not compatible with the **Large Scale Interface Detection** model, the **Surface Tension** model, or the **Interface Damping** model.

Field Function (Linearized Coefficient)

The drag coefficient is specified as a field function.

Strubelj-Tiselj

The drag coefficient is computed using the Strubelj and Tiselj interface drag method [496].

Strubelj-Tiselj property:

- *Relaxation Time Scale*
  A low value of the relaxation time scale helps in instantaneous equalizing of velocities of both of the phases. Increasing this value leads to reduced interface drag and slower equalizing of velocities of both of the phases. This value is t_{sc} in Eqn. (2295).

  For air-water flows, t_{sc} value of 0.01s is sufficient.

Interaction Length Scale

You specify the Interaction Length Scale for the first regime and the second regime.

Interaction Area Density

You specify the Interaction Area Density for the first regime and the second regime.

Large Scale Interface Detection

This model enables the detection of large interface cells and the calculation of the interfacial distance in those cells. This feature allows the implementation of special treatments and transfer terms between the phases including, for example, interphase heat transfer, mass transfer, and turbulence damping. The detection of a large interface also enables the use of a compressive scheme for volume-fraction in the vicinity of the large interface.

Large Scale Interface Detection properties:

Primary Criteria of Interface Detection

The primary criteria C_1, in Eqn. (2555).

The thickness of the raw interface can be adjusted by modifying this value.

Secondary Criteria of Interface Detection

The secondary criteria C_2, in Eqn. (2556).

This value is used to reduce the raw interface to a one-cell-thick interface.

Number of Cell Layers for Interface Band

Specifies the thickness of the Large Interface Marker Band which represents the large scale interface.

Large Scale Interface Detection field functions:

Large Interface Marker of [phase interaction]

A single-cell-thick layer that represents the Large Scale Interface in the phase interaction.

Large Interface Marker Band of [phase interaction]

A marker band that is created by including the neighbouring cells of the Large Interface Marker. The thickness of this band is governed by the value that is specified in the *Number of Cell Layers for Interface Band* property.

Lift Force

You specify the Lift Coefficient for the first regime and the second regime.

For more information, see the Lift Force formulation.

Turbulent Dispersion Force

You specify the Turbulent Dispersion Prandtl Number for the first regime and the second regime.

For more information, see the Turbulent Dispersion Force formulation.

Interphase Energy Transfer

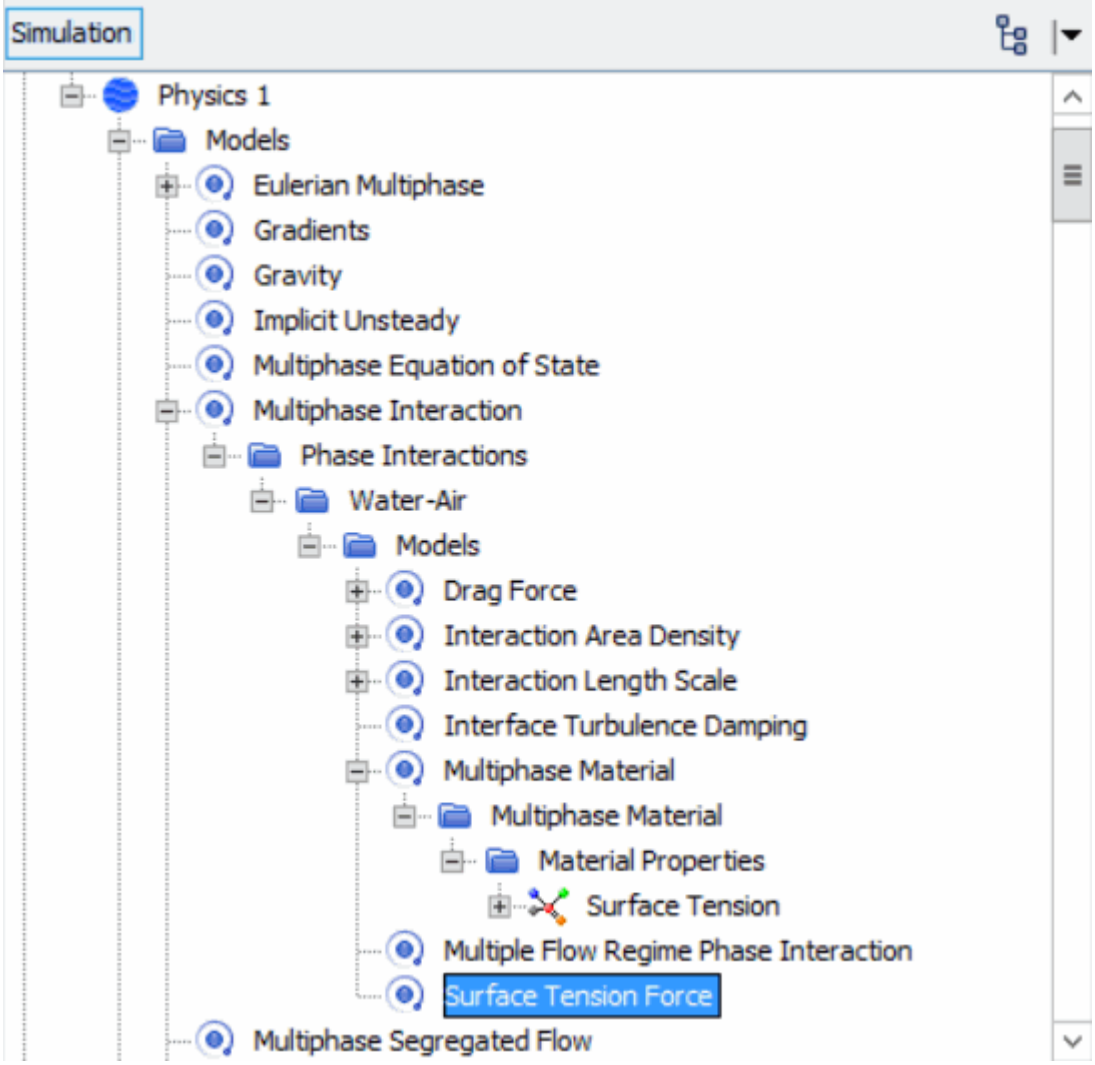
You specify the Nusselt number independently for the first regime, the interface regime, and the second regime.

For more information, see the Interphase Energy Transfer formulation.

Modeling Surface Tension in Multiple Flow Regimes

The surface tension force is a tensile force tangential to the interface separating two fluids. It works to keep the fluid molecules at the free surface in contact with the rest of the fluid.

σ is defined as the amount of work necessary to create a unit area of free surface. Surface tension effects are included in a Multiple Flow Regime phase interaction when the **Surface Tension Force** model is chosen in the *Phase Interaction Model Selection* dialog.



Surface tension always exists between a given pair of fluids. Its magnitude depends on the nature of the fluids in contact as well as on temperature. The experimentally determined surface tension coefficient σ expresses the ease with which the fluids can be mixed. For immiscible fluids, the value is always positive. The corresponding value node (**Surface Tension** in the above screenshot) appears as a material property for the phase interaction. It is entered as a scalar profile.

Surface tension is modeled in Multiple Flow Regime phase interactions according to the formulation proposed by Strubelj and others [495].

In the current implementation, each phase interaction is assigned its own surface tension coefficient. This coefficient is used to calculate the surface tension force between each of the phases in the phase interaction.

This implementation implies that, in general:

- The model works correctly in multiphase flows (more than two phases can be present).
- Different surface tension coefficients can be assigned to each phase interaction.

The calculation of the free surface curvature is sensitive to mesh quality. Therefore, to get good results even on coarse grids, you are recommended to use Cartesian meshes.

When the Surface Tension Force model is activated in a Multiple Flow Regime phase interaction, a **Contact Angle** node is added to the **Physics Values** of the phase interaction for all closed (wall) boundaries. This node lets you specify the capillarity effect at each of these boundaries.

See Modeling a Dynamic Contact Angle.

Modeling Suspensions and Emulsions

The presence of solid particles that are suspended in a Newtonian liquid is known to lead to non-Newtonian behavior, as the mixture viscosity becomes dependent on the volume fraction of the particle phase. In a Eulerian multiphase flow, STAR-CCM+ models this behavior as a suspension or an emulsion.

The Suspension Rheology model is used when the continuous phase is a liquid and the dispersed phase is a particle. The Emulsion Rheology model is used when both the continuous phase and the dispersed phase are liquids.

The relative viscosity is a strong function of volume fraction, so it is sensitive to small variations of volume fraction in the simulation. You are advised to run the Suspension Rheology model and Emulsion Rheology model as unsteady with a small time-step, and use an initial volume fraction that is below the maximum packing.

When you use the Suspension Rheology model or the Emulsion Rheology model, you can set the appropriate material properties to define the behavior of the multiphase material.

Contents:

Setting Up a Suspension Simulation
Setting Up an Emulsion Simulation
Suspension Rheology Model Reference
Emulsion Rheology Model Reference

Setting Up a Suspension Simulation

Suspensions of particles in liquids are common in industrial applications. The Suspension Rheology model describes the non-Newtonian viscosity of a fluid that consists of solid particles suspended in a liquid.

To model suspensions:

1. Set up a Eulerian Multiphase simulation.

   See Selecting the Eulerian Multiphase Model.
2. In the physics continuum, open the model selection dialog and, in the *Optional Models* group box, select **Suspension Rheology**.
3. Define the liquid phase and the particle phases.

   The Suspension Rheology model supports only one liquid phase. However, you can model multiple particle phases by creating multiple phase interactions and specifying a different particle phase as the dispersed phase in each phase interaction.

   See Defining Eulerian Phases.

4. For each particle phase, create a Continuous-Dispersed phase interaction, and specify the continuous phase as the liquid phase and the dispersed phase as the particle phase.

   See What Is the Continuous-Dispersed Phase Interaction Model?

For each particle phase, specify the maximum solid fraction and the initial volume fraction.

5. Expand the **Eulerian Multiphase** > **Eulerian Phases** > **[Particle Phase]** > **Models** node, select the **Solid Particle Rheology** node, and specify the *Maximum Solid Fraction*.

   See Suspension Rheology Properties.
6. Expand the **Eulerian Multiphase** > **Eulerian Phases** > **[Particle Phase]** > **Initial Conditions** node, and specify the initial **Volume Fraction**.
7. Expand the **Eulerian Multiphase** > **Eulerian Phases** > **Mixture** > **Material Properties** node and set the following material properties to define the behavior of the multiphase material:
   - Normal Relative Viscosity
   - Osmotic Pressure

     The *Hard Sphere Osmotic Pressure* option is temperature-dependent. To make this option available, select an energy model for the individual phases or assign the Segregated Fluid Isothermal model to each phase.
   - Relative Viscosity
   - Suspension Anisotropy

Setting Up an Emulsion Simulation

An emulsion is a dispersion of small droplets of one liquid in another liquid, where the droplets are not soluble or miscible. To model the non-Newtonian viscosity of such a fluid, use the Emulsion Rheology model.

Before you can use the Emulsion Rheology model, set up a Eulerian Multiphase simulation and define the appropriate liquid phases.

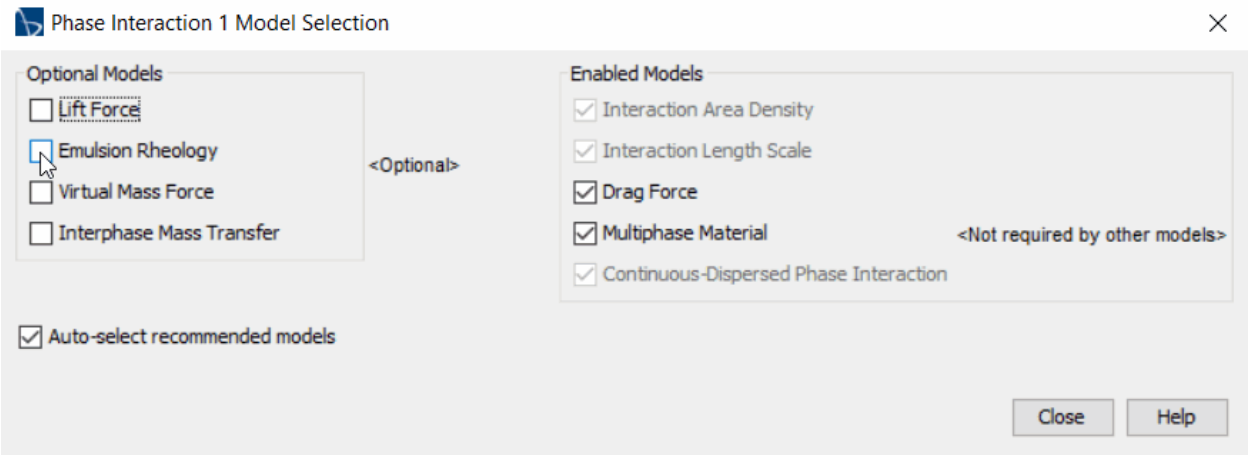
See Selecting the Eulerian Multiphase Model and Defining Eulerian Phases.

To model emulsions:

1. Create a Continuous-Dispersed phase interaction, and specify the continuous phase and the dispersed phase.

   The Emulsion Rheology model is available only when the Continuous-Dispersed phase interaction model is selected and both the continuous phase and the dispersed phase are specified as liquid phases.

   See What Is the Continuous-Dispersed Phase Interaction Model?
2. Reopen the *Phase Interaction Model Selection* dialog and, in the *Optional Models* group, select the **Emulsion Rheology** model.



3. Select the **Emulsion Rheology** node and set the *Maximum Packing* property value.

   See Emulsion Rheology Properties.

4. Select the **Calibration Prefactor** child node and specify the calibration value.

   See Calibration Prefactor.

5. Select the **Anisotropy Tensor** child node and specify the tensor.

   See Anisotropy Tensor.

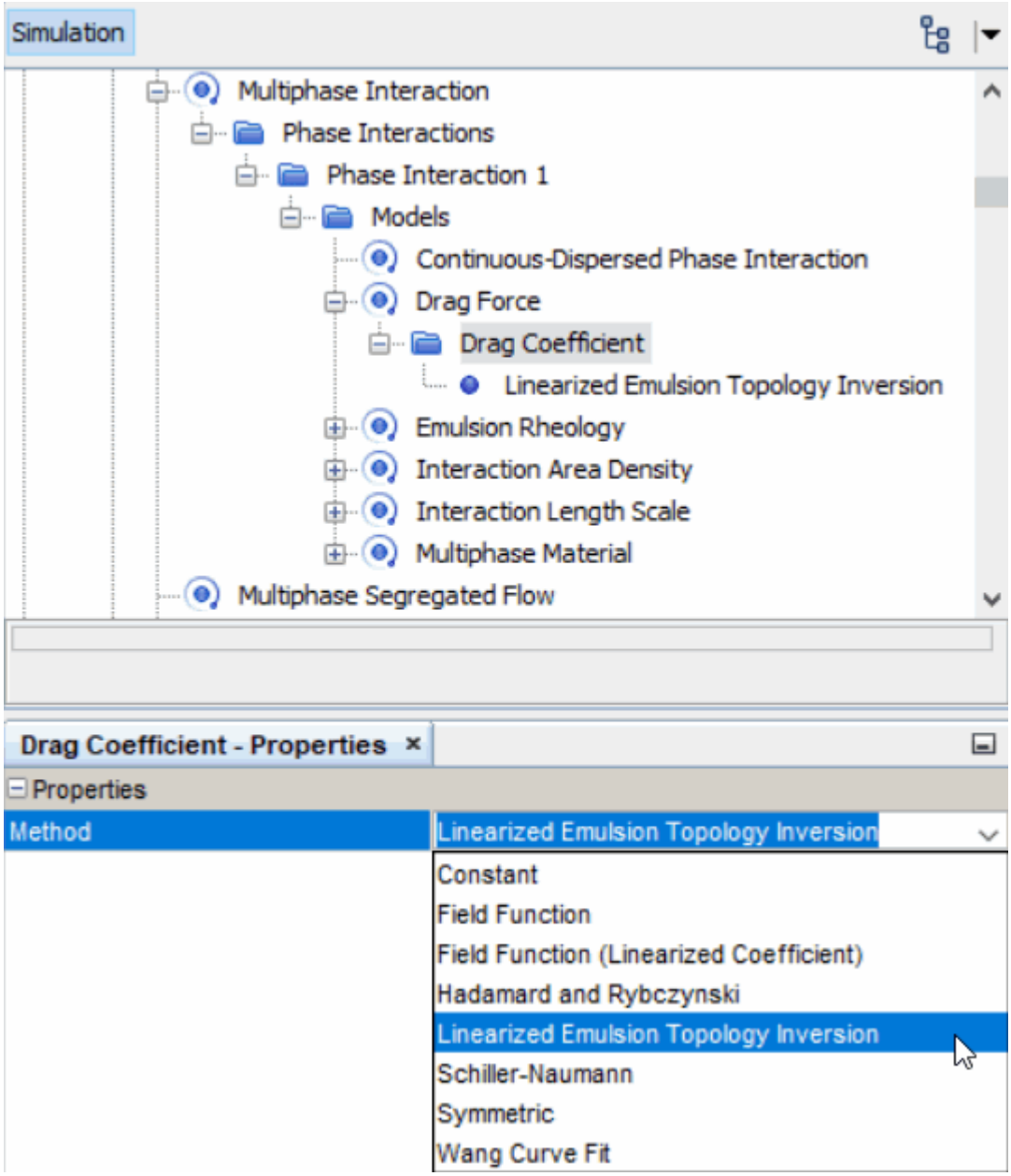
6. Expand the **Multiphase Material** > **Multiphase Material** > **Material Properties** node and set the following material properties:

   - Normal Relative Viscosity
   - Osmotic Pressure

     The *Hard Sphere Osmotic Pressure* option is temperature-dependent. To make this option available, select an energy model for the individual phases or assign the Segregated Fluid Isothermal model to each phase.
   - Relative Viscosity
   - Relaxation Time
   - Surface Tension
   - Yield Relative Viscosity
   - Yield Stress

7. Expand the **Drag Force** > **Drag Coefficient** node and select the appropriate Drag Force model.

   The Emulsion Rheology model uses the Linearized Emulsion Topology Inversion Drag Force model.



Suspension Rheology Model Reference

The Suspension Rheology model is used for fluids that consist of solid particles that are suspended in a liquid. The mixture viscosity depends on the volume fraction of the dispersed phase, and the fluid exhibits non-Newtonian behavior.

Suspension Rheology Model Reference

Theory	See Rheology of Emulsions and Suspensions.
Provided By	**[physics continuum]** > **Models** > *Optional Models*

Example Node Path	**[physics continuum]** > **Models** > **Suspension Rheology**	
Requires	An Eulerian Multiphase simulation with the following models activated: *Material*: **Eulerian Multiphase** *Eulerian Multiphase Model*: **Multiphase Segregated Flow** *Optional Models*: **Suspension Rheology** One liquid phase and one or more solid particle phases: • Phase 1, the continuous phase: *Material*: **Liquid** or **Multi-Component Liquid** • Phase 2, the dispersed phase: *Material*: **Particle** or **Multi-Component Particle** A phase interaction with the **Continuous-Dispersed Phase Interaction** model activated. The **Continuous-Dispersed Phase Interaction** model properties set: *Continuous Phase*: Phase 1 *Dispersed Phase*: Phase 2	
Activates	Models	Under the **Eulerian Multiphase** > **Eulerian Phases** > **[phase]** node: For Liquid phases: **Continuous Liquid Rheology** For Particle phases: **Solid Particle Rheology**
	Properties	Key property of **Solid Particle Rheology** is: *Maximum Solid Fraction*. See Solid Particle Rheology Properties.
	Materials	Under the **Eulerian Multiphase** > **Mixture** > **Material Properties** node: **Normal Relative Viscosity** **Osmotic Pressure** **Relative Viscosity** **Suspension Anisotropy** See Materials and Methods.
	Field Functions	See Field Functions.

Solid Particle Rheology Properties

Maximum Solid Fraction

The maximum solid fraction, assuming random close packing, is 0.645 for hard spheres. This value is a theoretical limit; the actual limit is lower for most practical cases. The value is lower again for non-spherical particles.

Materials and Methods

Normal Relative Viscosity

Describes the normal relative shear viscosity of the mixture. This value controls the shear-induced migration of the particles. If you set the Relative Viscosity property to use the Krieger-Dougherty model [450], this value should be set to constant zero.

Method	Corresponding Method Node
Morris and Boulay Model The normal relative viscosity is defined using the Morris and Boulay model (see Eqn. (2662)).	**Morris and Boulay Model** The Morris and Boulay Model has the following properties: • *Maximum Viscosity* Specifies the maximum value of the normal relative viscosity that is allowed in the simulation. In Eqn. (2662), the normal relative viscosity η_n tends to infinity as the dispersed phase reaches the limit of maximum packing. • *Contact Contribution* The contact contribution, K_n in Eqn. (2662). This value controls the strength of the shear-induced migration of particles.

Osmotic Pressure

The osmotic pressure is used to stop particles from exceeding their maximum packing volume fraction. This setting is useful for situations where the dispersed volume fraction could become large, such as in particle settling experiments.

The osmotic pressure is added to the particle momentum source in a similar way to the normal relative viscosity. The osmotic pressure is calculated using Eqn. (2671).

Method	Corresponding Method Node
Hard Sphere Osmotic Pressure Specifies Π using the Hard Sphere Osmotic Pressure formulation. This option is temperature-dependent. It is available when an energy model is selected for the individual phases, or the Segregated Fluid Isothermal model is assigned to each phase.	**Hard Sphere Osmotic Pressure** The Hard Sphere Osmotic Pressure model has the following properties: • *Maximum Packing* The maximum critical packing fraction (assuming random close-packing); ϕ_m in Eqn. (2672). This value is 0.645 for hard spheres. This value is a theoretical limit and is lower in most practical cases. The value is lower again for non-spherical particles. • *Maximum Pressure* Specifies the maximum value of the osmotic pressure Π. When the particles start to overlap, the osmotic pressure takes this maximum value, representing the infinite hard sphere potential. When the dispersed phase volume fraction is above the specified maximum packing value, Π is given this value to prevent the volume fraction from increasing much above the maximum packing.

When the dispersed phase volume fraction is below the specified maximum packing value, the osmotic pressure is calculated using Eqn. (2671).

Relative Viscosity

This dimensionless value describes the viscosity of the multiphase mixture. The relative viscosity (see Eqn. (2652)) tends to infinity as the dispersed phase reaches the limit of maximum packing.

Method	Corresponding Method Node
Krieger-Dougherty Model Describes non-Newtonian flow behavior in rigid sphere suspensions (see Eqn. (2658)). This model takes into account the interactions between neighboring spherical particles. See Krieger-Dougherty Model.	**Krieger-Dougherty Model** The Krieger-Dougherty model has the following properties: • *Maximum Viscosity* Specifies the maximum value of the relative viscosity that is allowed in the simulation. In Eqn. (2658), the relative viscosity η_r tends to infinity as the dispersed phase reaches the limit of maximum packing. • *Intrinsic Viscosity Exponent* The intrinsic viscosity; $[\eta]$ in Eqn. (2658). This value is 2.5 for spherical particles.
Morris and Boulay Model Describes the migration phenomenon that is observed in curvilinear flows of concentrated suspensions ([472]). This model uses shear-induced normal stresses to provide the driving force for migration (see Eqn. (2659)).	**Morris and Boulay Model** The Morris and Boulay model has the following properties: • *Maximum Viscosity* Specifies the maximum value of the relative viscosity that is allowed in the simulation. In Eqn. (2659), the relative viscosity η_r tends to infinity as the dispersed phase reaches the limit of maximum packing. • *Contact Contribution* The shear contact contribution, K_s in Eqn. (2659).

Suspension Anisotropy

Specifies the material property to be anisotropic. This property is a tensor profile, although it has limited options to only diagonal form. This property is handled in a similar way to anisotropic conductivity, that is, the tensor appears in the region. The only available method is **Anisotropic**, although the tensor can be made isotropic by adjusting the Physics Values in the region to Isotropic Tensor.

Field Functions

The following field functions are available:

Mixture Viscosity

Normal Relative Viscosity

Osmotic Pressure

Phase Particle Pressure of [phase]

Phase-Pair Eotvos Number of [phase interaction]

Phase-Pair Reynolds Number of [phase interaction]

Phase-Pair Single-Particle Reynolds Number of [phase interaction]

Relative Viscosity of [phase interaction]

Relaxation Time of [phase interaction]

Slip Viscosity of [phase interaction]

Solid Viscosity of [solid phase]

Emulsion Rheology Model Reference

The Emulsion Rheology model is used for fluids that consist of liquid droplets that are suspended in another liquid. The mixture viscosity depends on the volume fraction of the dispersed phase, and the fluid exhibits non-Newtonian behavior.

At a critical volume fraction, the phases can invert, and the dynamic viscosities of the continuous and dispersed phases swap.

Emulsion Rheology Model Reference

Theory	See Rheology of Emulsions and Suspensions.
Provided By	**[phase interaction]** > **Models** > *Optional Models*
Example Node Path	**[phase interaction]** > **Models** > **Emulsion Rheology**
Requires	A Eulerian Multiphase simulation with the following models activated: *Material*: **Eulerian Multiphase** *Eulerian Multiphase Model*: **Multiphase Segregated Flow** Two **Liquid** phases. A phase interaction with the **Continuous-Dispersed Phase Interaction** model activated. The **Continuous-Dispersed Phase Interaction** model properties *Continuous Phase* and *Dispersed Phase* set to liquid phases. The following phase interaction model activated: • *Optional Models*: **Emulsion Rheology**
Properties	Key properties are: *Maximum Packing*. See Emulsion Rheology Properties.

Activates	Model Controls (child nodes)	**Anisotropy Tensor** **Calibration Prefactor** See Emulsion Rheology Child Nodes
	Materials	Under the **Multiphase Material** node in the phase interaction: **Normal Relative Viscosity** **Osmotic Pressure** **Relative Viscosity** **Relaxation Time** **Surface Tension** **Yield Relative Viscosity** **Yield Stress** See Materials and Methods.
	Field Functions	See Field Functions.

Emulsion Rheology Properties

Maximum Packing

The maximum packing fraction is defined in a similar way as for suspensions of particles.

The specified value is multiplied by the calibration prefactor C to produce ϕ_m, which is used in Eqn. (2659) and Eqn. (2662) (for the Morris and Boulay model), and Eqn. (2658) (for the Krieger-Dougherty model).

A calibration prefactor of 1 reproduces the physics of hard particles, but for soft deformable particles the value is greater than 1.

Emulsion Rheology Child Nodes

Anisotropy Tensor

The Morris and Boulay model was derived from experiments in pipes and Couette devices where the flow becomes aligned in a particular direction that depends on the geometry. The normal stress was shown to be anisotropic and, at different strengths, lead to anisotropic normal stress differences.

Method	Corresponding Method Node
Isotropic Normal Stress Tensor	Sets the Anisotropy tensor equal to the identity tensor Eqn. (2661), and gives equal weighting to all directions. This method is the default and should be used for 3D flow.
X-Direction Aligned Anisotropic Normal Stress Tensor	Used for shear flow in the x-direction. The tensor is given in Eqn. (2660).

Calibration Prefactor

This factor C modifies the specified value of maximum packing fraction to account for changes that are due to shear-rate, temperature, or deformation of droplets. For example, in some emulsions, the deformation of droplets means that the maximum packing value can approach 1. You can use a constant or a field function to specify this factor.

See Maximum Packing.

Materials and Methods

Normal Relative Viscosity

Describes the normal relative viscosity of the mixture, which is similar to a particle pressure. If you set the Relative Viscosity property to use the Krieger-Dougherty model [450], this value defaults to zero.

Method	Corresponding Method Node
Morris and Boulay Model The normal relative viscosity is defined using the Morris and Boulay model (see Eqn. (2662)).	**Morris and Boulay Model** The Morris and Boulay Model has the following properties: • *Maximum Viscosity* Specifies the maximum value of the relative viscosity that is allowed in the simulation. In Eqn. (2662), the normal relative viscosity η_n tends to infinity as the dispersed phase reaches the limit of maximum packing. • *Contact Contribution* The contact contribution, K_n in Eqn. (2662).
Shear Thinning Model The normal relative viscosity depends on the shear strain-rate of the flow (see Eqn. (2665)).	**Shear Thinning Model** The Shear Thinning Model has the following properties: • *Maximum Viscosity* Sets the maximum viscosity cut-off for the zero strain rate Morris and Boulay normal viscosity. This value is $\eta_{n0}(\phi)$ in Eqn. (2663). • *Maximum Viscosity Infinity* Sets the maximum viscosity cut-off for the infinite strain rate Morris and Boulay normal viscosity. This value is $\eta_{n\infty}(\phi)$ in Eqn. (2664). • *Maximum Packing at Zero Shear Rate* Sets the asymptote for the zero-shear normal viscosity curves. This value is ϕ_{m0} in Eqn. (2663). • *Maximum Packing at Infinite Shear Rate* Sets the asymptote for the infinite-shear normal viscosity curves. This value is $\phi_{m\infty}$ in Eqn. (2664). • *Normal Zero-Strain Contact Parameter* Sets the strength of the particle interaction for the zero-shear normal viscosity curve. This value is K_{n0} in Eqn. (2663).

- *Normal Infinite-Strain Contact Parameter*
  Sets the strength of the particle interaction for the infinite-shear normal viscosity curve. This value is $K_{n\infty}$ in Eqn. (2664).
- *Use Normal Yield Stress*
  Activates the use of the Yield Stress material property.

Osmotic Pressure

The osmotic pressure is used to stop particles from exceeding their maximum packing volume fraction. This setting is useful for situations where the dispersed volume fraction could become large, such as in particle settling experiments.

The osmotic pressure is added to the particle momentum source in a similar way to the normal relative viscosity. The osmotic pressure is calculated using Eqn. (2671).

Method	Corresponding Method Node
Hard Sphere Osmotic Pressure Specifies Π using the Hard Sphere Osmotic Pressure formulation.	**Hard Sphere Osmotic Pressure** The Hard Sphere Osmotic Pressure model has the following properties: • *Maximum Packing* The maximum critical packing fraction (assuming random close-packing); ϕ_m in Eqn. (2672). This value is 0.645 for hard spheres. This value is a theoretical limit and is lower in most practical cases. The value is lower again for non-spherical particles. • *Maximum Pressure* Specifies the maximum value of the osmotic pressure Π. When the particles start to overlap, the osmotic pressure takes this maximum value, representing the infinite hard sphere potential. When the dispersed phase volume fraction is above the specified maximum packing value, Π is given this value to prevent the volume fraction from increasing much above the maximum packing.

When the dispersed phase volume fraction is below the specified maximum packing value, the osmotic pressure is calculated using Eqn. (2671).

Relative Viscosity

This dimensionless value describes the viscosity of the multiphase mixture. The relative viscosity (see Eqn. (2652)) tends to infinity as the dispersed phase reaches the limit of maximum packing.

Method	Corresponding Method Node
Krieger-Dougherty Model Describes non-Newtonian flow behavior in rigid sphere suspensions (see Eqn. (2658)). This model takes into account the	**Krieger-Dougherty Model** The Krieger-Dougherty model has the following properties: • *Maximum Viscosity* Specifies the maximum value of the relative viscosity that is allowed in the simulation.

interactions between neighboring spherical particles. See Krieger-Dougherty Model.	In Eqn. (2658), the relative viscosity η_r tends to infinity as the dispersed phase reaches the limit of maximum packing. • *Intrinsic Viscosity Exponent* The intrinsic viscosity; $[\eta]$ in Eqn. (2658). This value is 2.5 for spherical particles.
Morris and Boulay Model Describes the migration phenomenon that is observed in curvilinear flows of concentrated suspensions ([472]). This model uses shear-induced normal stresses to provide the driving force for migration (see Eqn. (2659)).	**Morris and Boulay Model** The Morris and Boulay model has the following properties: • *Maximum Viscosity* Specifies the maximum value of the relative viscosity that is allowed in the simulation. In Eqn. (2659), the relative viscosity η_r tends to infinity as the dispersed phase reaches the limit of maximum packing. • *Contact Contribution* The contact contribution, K_s in Eqn. (2659).
Shear Thinning Model Blends a zero-shear and infinite shear Morris and Boulay relative viscosity model [472] with a Carreau Generalized Newtonian model (see Eqn. (2665)).	**Shear Thinning Model** The Shear Thinning model has the following properties: • *Maximum Relative Viscosity 0* Sets the maximum viscosity cut-off for the zero strain rate Morris and Boulay viscosity. This value is $\eta_{r0}(\phi)$ in Eqn. (2663). • *Maximum Relative Viscosity Infinity* Sets the maximum viscosity cut-off for the infinite strain rate Morris and Boulay viscosity. This value is $\eta_{r\infty}(\phi)$ in Eqn. (2664). • *Maximum Packing at Zero Shear Rate* Sets the asymptote for the zero-shear viscosity curves. This value is ϕ_{m0} in Eqn. (2663). • *Maximum Packing at Infinite Shear Rate* Sets the asymptote for the infinite-shear viscosity curves. This value is $\phi_{m\infty}$ in Eqn. (2664). • *Zero-Shear Contact Parameter* Sets the strength of the particle interaction for the zero-shear curve. This value is K_{s0} in Eqn. (2663). • *Infinite-Shear Contact Parameter* Sets the strength of the particle interaction for the infinite-shear curve. This value is $K_{s\infty}$ in Eqn. (2664). • *Use Yield Stress* Activates the use of the Yield Stress material property.

Relaxation Time

Sets the strain-rate at which shear-thinning occurs. This property is equivalent to the relaxation time λ in the Carreau-Yasuda non-Newtonian model (see Eqn. (1516)).

The relaxation time is typically a constant or a function of volume fraction. It is entered as a scalar profile.

Note: Available only when the *Relative Viscosity* material property uses the *Shear Thinning Model*.

Surface Tension

The surface tension coefficient, σ, expresses the ease with which the two fluid phases can be mixed and is defined as the amount of work necessary to create a unit area of free surface. Its magnitude depends on the nature of the fluids in contact: for immiscible fluids, the value is always positive. The surface tension coefficient is entered as a scalar profile.

Yield Relative Viscosity

The yield relative viscosity is analogous to the yield viscosity μ_0 in the Herschel-Bulkley non-Newtonian model. This quantity is typically a large value that describes the material when it is solid-like (below the critical shear-strain rate).

The yield relative viscosity is typically a constant. It is entered as a scalar profile.

Note: Available only when the *Relative Viscosity* material property uses the *Shear Thinning Model*.

Yield Stress

The yield stress is analogous to the yield stress τ_0 in the Herschel-Bulkley non-Newtonian model.

Note: Available only when the *Relative Viscosity* material property uses the *Shear Thinning Model*.

Method	Corresponding Method Node
Emulsion Yield Stress The stress above which the emulsion behaves as a fluid. If the applied stress is less than the emulsion yield stress, the emulsion behaves as a solid. The emulsion yield stress τ_y is calculated in Eqn. (2669).	**Emulsion Yield Stress** The Emulsion Yield Stress model has the following properties: • *Maximum Packing* The maximum critical packing fraction; ϕ_m in Eqn. (2669). Emulsions can have a much larger maximum packing fraction than particle suspensions due to the deformability of liquid droplets. • *Maximum Yield Stress* Sets the maximum value of the Emulsion Yield Stress, τ_y, that is used in Eqn. (2668). • *Prefactor* A constant prefactor; τ_{y0} in Eqn. (2669). The default value is 0.5, as used in [468].

Field Functions

The following field functions are available:

Mixture Viscosity of [phase interaction]

Normal Relative Viscosity of [phase interaction]

Osmotic Pressure of [phase interaction]

Relative Viscosity of [phase interaction]

Relaxation Time of [phase interaction]

Slip Viscosity of [phase interaction]

Yield Relative Viscosity of [phase interaction]

Yield Stress of [phase interaction]

Modeling Interphase Mass Transfer

You can model mass transfer between Eulerian phases. The **Interphase Mass Transfer** model makes available the relevant mass transfer models. These models include boiling, evaporation, condensation, diffusion, and crystalization.

The following options are provided for modeling mass transfer between Eulerian phases:

- Bulk boiling and condensation, which is modeled with the Boiling Mass Transfer Rate model for a Continuous-Dispersed Phase Interaction and modeled with the Boiling/Condensation model for a Multiple Flow Regime Phase Interaction.
- Diffusion mass transfer, which is modeled with the Dissolution Mass Transfer Rate model.
- Evaporation and condensation, which is modeled with the Single Component and Multicomponent Droplet Evaporation Mass Transfer Rate models.
- Crystalization, which is modeled with the Single Component Crystal Growth models.
- Interphase reactions, which are modeled with the Interphase Reaction model.
- User-defined mass transfer, which is modeled with the User-Defined Mass Flux model.

Further information on bulk boiling is provided within the section, Modeling the boiling process.

Contents:

Modeling Dissolution Mass Transfer
Modeling Droplet Evaporation
Modeling Crystallization
Single Component Crystal Growth Model Reference
Nucleation Model Family Reference

Setting the Sherwood Number
What is the Equilibrium Coefficient?
What Is the User-Defined Interphase Mass Flux?

Modeling Dissolution Mass Transfer

For Eulerian multi-component multiphase flows, transfer between the components of each phase can occur. The Dissolution Mass Transfer Rate model includes the case where one phase is liquid and one phase is gaseous, and the case where both phases are liquid.

In STAR-CCM+, the rate of transfer of species from one phase to another is governed by Eqn. (2460).

To model dissolution mass transfer:

1. Set up a Multiphase Segregated Flow simulation.
2. Create two Eulerian phases, usually one liquid and one gas phase. Both of the phases must be multi-component.
3. Create a phase interaction and activate the Continuous-Dispersed Phase Interaction model.
4. On the *[phase Interaction]* node, set the *Continuous Phase* as the liquid phase and the *Dispersed Phase* as the gas phase.
5. In the *Phase Interaction Model Selection* dialog, in the *Optional Models* group box, select **Interphase Mass Transfer**.
6. In the *Interphase Mass Transfer Rate* group box, select **Dissolution Mass Transfer Rate**.
7. If the Phase Coupled Fluid Energy model is already activated, select the Interphase Energy Transfer phase interaction model to make sure that the phase temperatures reach sensible values.

   The **Dissolution Mass Transfer Rate** model assumes that the effect of interphase mass transfers on the phase temperatures is negligible.
8. Select the **Dissolution Mass Transfer Rate** node and set the *Phase 0 Components* property to specify the components to transfer.

   See Dissolution Mass Transfer Rate Properties.
9. In the **Continuous Phase Sherwood Number** and **Dispersed Phase Sherwood Number** nodes, specify the appropriate settings.

   See Setting the Sherwood Number.
10. In the **Equilibrium Coefficient** > **Composite** node, specify the appropriate equilibrium coefficients.

    See Setting the Equilibrium Coefficient.

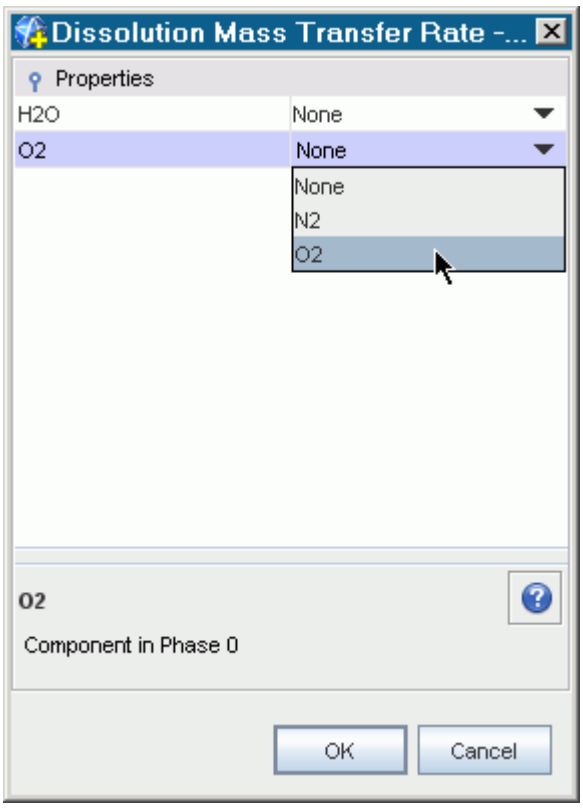
Dissolution Mass Transfer Rate Properties

The **Dissolution Mass Transfer Rate** model has the following property:

Phase 0 Components

Specifies the relationship between the components that are being transferred between phases.

The *Dissolution Mass Transfer Rate - Phase 0 Components* dialog lets you specify the components to transfer. The components of the continuous phase are activated on the left-hand column. The components of the dispersed phase can be matched in the right-hand column as shown:



For each transfer activated, the continuous phase component name appears under the **Equilibrium Coefficient** > **Composite** node.

Modeling Droplet Evaporation

The **Single Component Droplet Evaporation Rate** model and the **Multicomponent Droplet Evaporation Mass Transfer Rate** model account for mass transfer due to evaporation and condensation.

Mass transfer occurs when the system departs from the equilibrium condition (Eqn. (2359)). Evaporation occurs when the gas bulk mole fraction of the evaporating component is below the equilibrium value; otherwise condensation occurs.

The **Single Component Droplet Evaporation Rate** model is used for single component (pure fluid) droplets. For multiple component droplets, the **Multicomponent Droplet Evaporation Mass Transfer Rate** model is used. This model assumes that the liquid drops are internally homogeneous and that the liquid behaves like an ideal mixture. The presence of inert components is allowed in both the gas and the liquid.

To model droplet evaporation:

1. Set up a Multiphase Segregated Flow simulation with an energy model activated in the physics continuum.
2. Create two Eulerian phases: one multi-component gas phase and one liquid phase.

   The liquid phase can be a single component or a multi-component liquid.
3. Create a phase interaction and activate the Continuous-Dispersed Phase Interaction model.

4. On the *[phase Interaction]* node, set the *Continuous Phase* as the gas phase and the *Dispersed Phase* as the liquid phase.
5. In the *Phase Interaction Model Selection* dialog, in the *Optional Models* group box, select **Interphase Mass Transfer**.

   The appropriate Droplet Evaporation Mass Transfer Rate model becomes available.
6. In the *Interphase Mass Transfer Rate* group box, select **Single Component Droplet Evaporation Rate** or **Multicomponent Droplet Evaporation Mass Transfer Rate**, whichever is applicable.
7. Select the **Droplet Evaporation Mass Transfer Rate** node and specify the relevant property settings.

   See Droplet Evaporation Mass Transfer Rate Properties.
8. In the **Continuous Phase Sherwood Number** and **Continuous Phase Nusselt Number** nodes, specify the appropriate settings.

   See Setting the Sherwood Number and Setting the Nusselt Number.
9. In the **Equilibrium Coefficient** > **Composite** node, specify the appropriate equilibrium coefficients.

   See Setting the Equilibrium Coefficient.

Droplet Evaporation Mass Transfer Rate Properties

The **Droplet Evaporation Mass Transfer Rate** models have the following properties:

Vapor Component

The evaporating/condensing component of the gas. The molecular weight of the specified vapor component must be the same as the molecular weight of the liquid phase.

Applies to the **Single Component Droplet Evaporation Mass Transfer Rate** model only.

Phase 0 Components

Specifies the map between the species in the gas and the species in the liquid drops. Inert species are allowed in both phases. The molecular weight of the species inside each mapped pair is required to be the same.

Applies to the **Multicomponent Droplet Evaporation Mass Transfer Rate** model only.

Condensation Option

When **Activated**, both condensation and evaporation are allowed. This setting is the default. When **Deactivated**, only evaporation is allowed.

Heat Transfer Limited Mode

Specifies whether the heat-transfer limited evaporation mode is permitted.

This mode is used when the vapor at the droplet surface is saturated and sub-critical. The condition for a saturated vapor is that the vapor mole fraction at the droplet surface exceeds unity. The vapor mole fraction at the droplet surface is evaluated as the ratio of the saturation pressure at the droplet surface temperature and the pressure in the surrounding gas.

When **Activated**, the heat transfer limited evaporation mode is allowed. This setting is the default.

When **Deactivated**, the heat-transfer limited evaporation mode is not permitted.

When the conditions for the heat-transfer limited evaporation mode are satisfied (that is, the ratio of the saturation pressure at the droplet surface temperature to the pressure in the surrounding gas exceeds 0.99999), the mode is not used. Instead, the vapor mole fraction at the droplet surface is fixed at 0.99999 and the mass diffusion limited mode is applied.

Use this setting with care, as the diffusion limited evaporation mass transfer rate can cause numerical instability if applied close to saturation conditions.

Applies to the **Multicomponent Droplet Evaporation Mass Transfer Rate** model only.

Modeling Crystallization

You can model crystal growth in two ways: a concentration-driven (solute crystallization) model and a temperature-driven (melt crystallization) model. The crystal growth model also uses a corresponding nucleation model.

Solute crystallization is a popular technological process that is used, for example, in the pharmaceutical industry (Woo, [518]). In this process, a compound (the solute) is dissolved in another liquid (the solvent). When the temperature drops, or an anti-solvent is added, the concentration of the solute becomes higher than the saturation concentration - at which point crystallization starts. The surface growth rate of the crystal is a function of the supersaturation of concentration.

Melt crystallization can occur in a single-component liquid (for example, ice formation in water). Admixtures that are dissolved in the liquid are also important. Firstly, admixtures lower the saturation temperature. For example, the freezing temperature of sea water is lower than the freezing temperature of fresh water. Secondly, the composition of the crystals is different from the composition of the melt. For example, a melt crystallization process can be used for water desalination.

To model crystal growth:

1. Set up a Multiphase Segregated Flow simulation with the **Phase Coupled Fluid Energy** model activated.
2. Create an Eulerian phase for the solvent/solute or melt and do the following:
   a) In the *Phase Model Selection* dialog, activate the following phase models:

      Material: **Multi-Component Liquid**

      Energy: **Segregated Fluid Temperature**

      Reaction Regime: **Non-reacting**

      Equation of State: **Constant Density**

      Optional Models: **Exact Wall Distance**

   b) Specify the liquid material components.

      For solute crystallization applications, specify the solvent and solute components. For melt crystallization applications, specify the liquid component and the admixture component.

   c) Specify the appropriate material properties for each component.
3. Create a Eulerian phase for the crystals and do the following:
   a) In the *Phase Model Selection* dialog, activate the following phase models:

      Material: **Particle**

      Energy: **Segregated Fluid Temperature**

Optional Models: **Exact Wall Distance**

b) In the *Phase Model Selection* dialog, select the particle size distribution to use for the crystal nucleation.

Optional Models: **Particle Size Distribution**

Particle Size Distribution: **Adaptive Multiple Size-Group** or **S-Gamma**.

c) For the **Adaptive Multiple Size-Group** model only, ensure that the *Velocity Option* is set to **One-Speed** (the default setting).

The alternative option, **Multi-Speed**, is not compatible with mass transfer models.

See Adaptive Multiple Size-Group Model.

d) Specify the particle material component.

For solute crystallization applications, the particle material is the solute component. For melt crystallization applications, the particle material is the liquid component.

e) Specify the appropriate material properties.

The *Heat of Formation* is an important material property. The specified value must be consistent with the value that is specified in the corresponding component of the liquid phase.

4. Create a phase interaction and activate the **Continuous-Dispersed Phase Interaction** model.
5. On the *[phase Interaction]* node, set the *Continuous Phase* as the liquid phase and the *Dispersed Phase* as the particle phase.
6. In the phase interaction model selection dialog, do the following:

a) In the *Optional Models* group box, activate **Interphase Mass Transfer**.

b) In the *Interphase Mass Transfer Rate* group box, activate the appropriate crystal growth model:

- **Single Component Crystal Growth**
- **Single Component Crystal Growth From Melt**

c) In the *Optional Models* group box, activate the particle size distribution nucleation model: **MuSiG Nucleation** or **S-Gamma Nucleation**.

The available model corresponds to the particle size distribution model that is activated in the particle phase.

d) In the *Nucleation* group box, activate the mass transfer nucleation model: **Crystal Nucleation**.

The available model corresponds to the interphase mass transfer model that is activated in the phase interaction, in this case, crystal growth.

The required settings for melt crystallization are similar to the required settings for solute crystallization, with two important differences:

- *Equilibrium Temperature* is provided instead of *Equilibrium Concentration*.
- The *Continuous Phase Sherwood Number* and the *Surface Under-Relaxation Factor* are not required.

Any difference in solute concentration in the bulk liquid and at the crystal surface is neglected.

7. Specify the *Single Component Crystal Growth* model properties:

- Set the *Solute Component* to specify the liquid component that crystallizes.
- For the *Single Component Crystal Growth* model only, set the *Surface Under-Relaxation Factor*.
- For the *Single Component Crystal Growth From Melt* model only, set the *Reference Temperature*.

See Single Component Crystal Growth Model Properties.

8. In the **Single Component Crystal Growth** model child nodes, set the following:
   - **Continuous Phase Nusselt Number**

     See Setting the Nusselt Number.
   - **Continuous Phase Sherwood Number**.

     This setting applies to the **Single Component Crystal Growth** model only.

     See Setting the Sherwood Number.
   - **Equilibrium Concentration** (for the **Single Component Crystal Growth** model)

     Equilibrium Temperature (for the **Single Component Crystal Growth From Melt** model)
   - **Growth Exponent**, **Growth Prefactor**, **Dissolution Exponent**, and **Dissolution Prefactor**.

     See Model Controls.
9. Specify the nucleation model parameters.

   The *Nucleation Rate* and the *Nucleus Diameter* are required for all nucleation models.

   Crystal nucleation also uses the *Nucleation Rate Concentration Derivative* and the *Nucleation Rate Temperature Derivative* terms for linearization purposes.

   See Nucleation Model Controls.

Single Component Crystal Growth Model Reference

Crystal growth is modeled as a mass transfer between a liquid phase and a solid (particle) phase. STAR-CCM+ provides two crystal growth models: a concentration-driven (solute crystallization) model and a temperature-driven (melt crystallization) model.

Single Component Crystal Growth Model Reference

Model Names	**Single Component Crystal Growth**
	Single Component Crystal Growth From Melt
Theory	See Crystallization Formulation.
Provided By	**[phase interaction]** > **Models** > *Interphase Mass Transfer Rate*
Example Node Path	**Multiphase Interaction** > **Phase Interactions** > **[phase interaction]** > **Models** > **Single Component Crystal Growth**
Requires	An Eulerian Multiphase simulation with the following models activated: *Material*: **Eulerian Multiphase**, **Multiphase Interaction** (Selected automatically) *Eulerian Multiphase Model*: **Multiphase Segregated Flow**, **Multiphase Equation of State** (Selected automatically), **Gradients** (Selected automatically) *Optional Models*: **Phase Coupled Fluid Energy** *Viscous Regime*: **Laminar** or **Turbulent** *Space*: **Three Dimensional**, **Axisymmetric**, or **Two Dimensional** *Time*: **Steady** or **Implicit Unsteady** A minimum of two Eulerian phases: one liquid phase (the solvent/solute or melt) and one particle phase (the crystals).

	In the liquid phase, the following phase models activated: *Material*: **Multi-Component Liquid** *Energy*: **Segregated Fluid Temperature** *Reaction Regime*: **Non-reacting** *Equation of State*: **Constant Density** *Optional Models*: **Exact Wall Distance** In the particle phase, the following models activated: *Material*: **Particle** *Energy*: **Segregated Fluid Temperature** *Optional Models*: **Exact Wall Distance** *Optional Models*: **Particle Size Distribution** *Particle Size Distribution*: **Adaptive Multiple Size-Group** or **S-Gamma**. A Continuous-Dispersed phase interaction with the *Continuous Phase* set to the liquid phase and the *Dispersed Phase* set to the particle phase. In the phase interaction, the following models activated: *Optional Models*: **Interphase Mass Transfer** *Interphase Mass Transfer Rate*: **Single Component Crystal Growth** or **Single Component Crystal Growth From Melt**	
Properties	Key properties are: *Solute Component*, *Surface Under-Relaxation Factor*, *Reference Temperature*. See Single Component Crystal Growth Model Properties.	
Activates	Physics Models	In the phase interaction model selection dialog: *Optional Models*: **MuSiG Nucleation** or **S-Gamma Nucleation**. The available model corresponds to the particle size distribution model that is activated in the particle phase. See Crystal Nucleation Model Reference.
	Model Controls (child nodes)	The available options depend on the crystal growth model that is activated. **Continuous Phase Sherwood Number** **Equilibrium Temperature** **Equilibrium Concentration** **Continuous Phase Nusselt Number** **Growth Exponent** **Growth Prefactor** **Dissolution Exponent** **Dissolution Prefactor** See Model Controls.
	Materials	*Surface Tension* See Materials and Methods.

	Field Functions	See Field Functions.

Single Component Crystal Growth Model Properties

Solute Component

Specifies the liquid component that crystallizes.

Surface Under-Relaxation Factor

The under-relaxation factor for the surface concentration (Eqn. (2644)).

Applies to the **Single Component Crystal Growth** model only.

Reference Temperature

This value is used to calculate ΔT that is used in Eqn. (2646).

Applies to the **Single Component Crystal Growth From Melt** model only.

Model Controls

The following child nodes are provided:

Continuous Phase Sherwood Number

Applies to the **Single Component Crystal Growth** model only.

See Setting the Sherwood Number.

Equilibrium Temperature

This value is T_{sat} in Eqn. (2646).

Applies to the **Single Component Crystal Growth From Melt** model only.

Equilibrium Concentration

This value is C_{sat} in Eqn. (2642) and Eqn. (2643).

Applies to the **Single Component Crystal Growth** model only.

Continuous Phase Nusselt Number

See Setting the Nusselt Number.

Growth Exponent

For Single Component Crystal Growth, this value is m_g^+ in Eqn. (2643), which is the exponent m_g in Eqn. (2641).

For Single Component Crystal Growth From Melt, this value is m_g^+ in Eqn. (2646).

Growth Prefactor

For Single Component Crystal Growth, this value is k_g^+ in Eqn. (2642), which is the prefactor k_g in Eqn. (2641).

For Single Component Crystal Growth From Melt, this value is k_g^+ in Eqn. (2646).

Dissolution Exponent

For Single Component Crystal Growth, this value is m_g^- in Eqn. (2643), which is the exponent m_g in Eqn. (2641).

For Single Component Crystal Growth From Melt, this value is m_g^- in Eqn. (2646).

Dissolution Prefactor

For Single Component Crystal Growth, this value is k_g^- in Eqn. (2642), which is the prefactor k_g in Eqn. (2641).

For Single Component Crystal Growth From Melt, this value is k_g^- in Eqn. (2646).

Materials and Methods

Surface Tension

Set the value of the surface tension coefficient, σ.

See Surface Tension Formulation.

Applies to the **Single Component Crystal Growth** model only.

Field Functions

Interphase Mass Transfer Rate of [phase interaction]

Mass Fraction of [crystal material] of [phase interaction]

Nucleation Model Family Reference

Nucleation is the creation of new particles from the continuous phase. STAR-CCM+ models nucleation as a combination of a population balance nucleation model and a mass transfer nucleation model.

Nucleation Model Family Reference

Model Names	**MuSiG Nucleation**
	S-Gamma Nucleation
	Crystal Nucleation
	User-Defined Nucleation
Theory	See Crystallization Formulation.
Provided By	**[phase interaction]** > **Models** > *Nucleation*
Example Node Path	**Multiphase Interaction** > **Phase Interactions** > **[phase interaction]** > **Models** > **Crystal Nucleation**

Requires	An Eulerian Multiphase simulation with a Continuous-Dispersed phase interaction. The appropriate phase interaction model activated: • For Crystal Nucleation The **Single Component Crystal Growth** or **Single Component Crystal Growth From Melt** model. • For User-Defined Nucleation The **User-Defined Interphase Mass Flux** model. In the phase interaction model selection dialog: • The population balance nucleation model activated: *Optional Models*: **MuSiG Nucleation** or **S-Gamma Nucleation**. The available model corresponds to the particle size distribution model that is activated in the particle phase. • The mass transfer nucleation model activated: *Nucleation*: **Crystal Nucleation** (activated automatically when a crystal growth model is active) or **User-Defined Nucleation** (activated automatically when the **User-Defined Interphase Mass Flux** model is active) The available model corresponds to the interphase mass transfer model that is activated in the phase interaction.	
Properties	None.	
Activates	Physics Models	None.
	Model Controls (child nodes)	**Nucleation Rate** **Nucleation Rate Concentration Derivative** **Nucleation Rate Temperature Derivative** **Nucleus Diameter** See Model Controls.
	Field Functions	None.

Nucleation Model Controls

The following child nodes are available:

Nucleation Rate

This value is $\dot{N}_0$ in Eqn. (2636).

Nucleation Rate Concentration Derivative

The derivative of the nucleation rate with respect to the concentration of the solute. STAR-CCM+ uses this value for linearization purposes.

Applies to **Crystal Nucleation** only.

Nucleation Rate Temperature Derivative

The derivative of the nucleation rate with respect to the temperature of the solute. STAR-CCM+ uses this value for linearization purposes.

Applies to **Crystal Nucleation** only.

Nucleus Diameter

This value is L_0 in Eqn. (2636).

Setting the Sherwood Number

The Sherwood Number is a dimensionless mass transfer coefficient. It represents the ratio of the convective mass transfer to the rate of diffusive mass transport across the boundary between two fluid phases.

The *Continuous Phase Sherwood Number* is used in the calculation of the mass transfer rate from the continuous phase to the dispersed phase. For the reverse mass transfer, from the dispersed phase to the continuous phase, the *Dispersed Phase Sherwood Number* is used.

The Sherwood Number is used in the following calculations:

- Dissolution (**Dissolution Mass Transfer Rate** model)

  See Eqn. (2373) (continuous phase) and Eqn. (2373) (dispersed phase).
- Droplet Evaporation (**Single Component Droplet Evaporation Rate** and **Multicomponent Droplet Evaporation Mass Transfer Rate** models)

  See Eqn. (2366) (continuous phase).
- Solute Crystallization (**Single Component Crystal Growth** model)

  See Eqn. (2644) (continuous phase).

Sherwood Number Properties

The Sherwood Number has the following properties:

Dimensions

Dimensionless.

Method

The method to use for specifying the Sherwood number that is used in the mass transfer rate calculation.

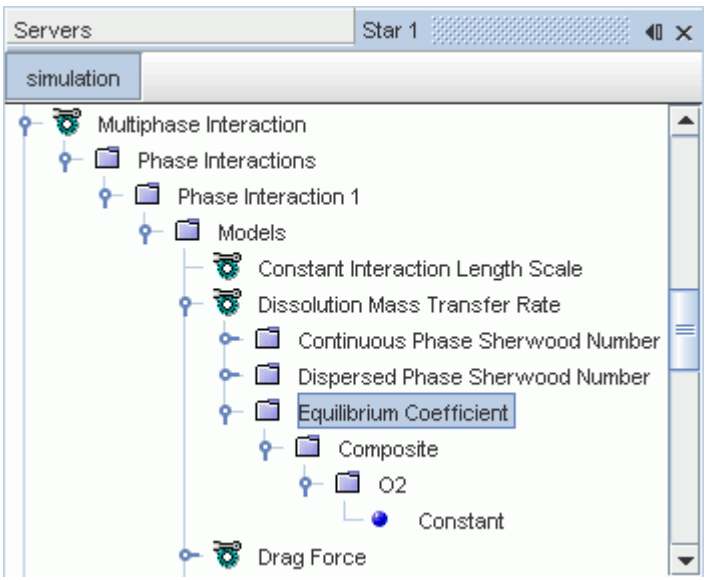
Method	Corresponding Method Node
Ranz-Marshall The Sherwood number is computed using the Ranz-Marshall correlation (Eqn. (2374)).	**Ranz-Marshall**
Armenante-Kirwan The model for mass transfer to micro-particles that was proposed by Armenante and Kirwan ([400]).	**Armenante-Kirwan** The **Armenante-Kirwan** node has the following properties:

This method is available only for turbulent flow regimes.

- *Turbulent Reynolds Number Exponent*
  This value is β in Eqn. (2650).
- *Prandtl Number Exponent*
  This value is γ in Eqn. (2650).
- *Relative Density Difference Exponent*
  This value is δ in Eqn. (2650).
- *Calibration Factor*
  This value is α in Eqn. (2650).

Setting the Equilibrium Coefficient

You specify the appropriate equilibrium coefficient for a mass transfer phase interaction.



The **Equilibrium Coefficient** node lets you specify the appropriate equilibrium coefficient:

- For gas-liquid interactions, the ratio of gas to liquid mole fractions, which is Eqn. (2456).
- For multicomponent interactions where you map the pairs of species involved, you specify the equilibrium coefficient for each species pair.
- For liquid-liquid interactions, the ratio of continuous to dispersed mole fractions, which is Eqn. (2459).

Equilibrium Coefficient Properties

Dimensions

The dimensionality of the Equilibrium Coefficient (read-only).

Method

Read-only manager for the equilibrium coefficients that are activated for each component transfer. The only option is **Composite**.

The **Composite** node does not have properties.

Equilibrium Coefficient for Individual Component Transfer Properties

Dimensions

The dimensionality of the Equilibrium Coefficient method (read-only).

Method

Selects the method:

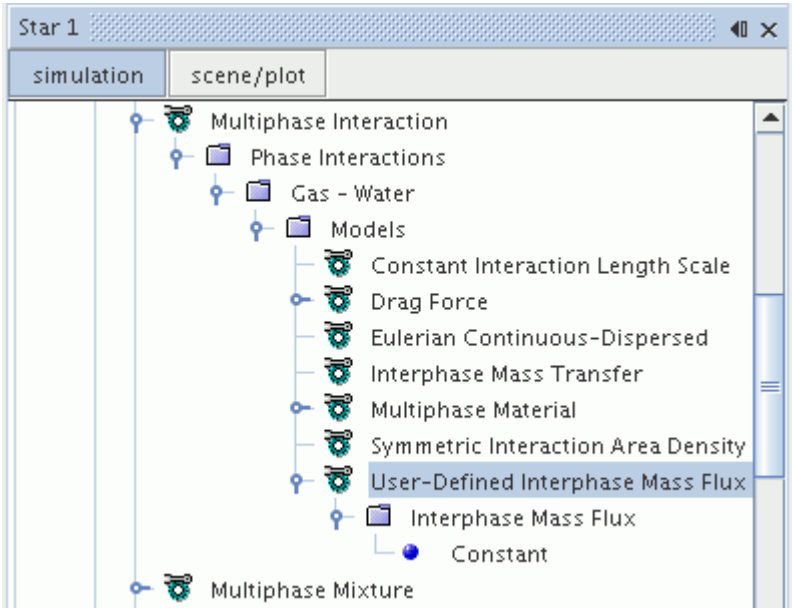
- **Henry's Law**
  Set a value for the Henry's Law coefficient, Eqn. (2458).
- **Raoult's Law**
  A **Saturation Pressure** node is added to the liquid phase. This node can be used as an input to Eqn. (2457).

For liquid-liquid interactions, the equilibrium coefficient specification is limited to **Constant** or **Field Function**.

Modeling a User-Defined Interphase Mass Flux

The **User-Defined Interphase Mass Flux** phase interaction model is a basic model for interphase mass transfer that is not governed by energy or species transfer.

The **User-Defined Interphase Mass Flux** phase interaction model is the only model that is available when the Phase Coupled Fluid Energy model is not active. An example of where this model can be useful is a simulation of bubble dissolution into de-aerated water. When selected, the **User-Defined Interphase Mass Flux** node represents the model, which has the **Interphase Mass Flux** as its child node. The Method property of this node is used to specify how the transfer rate is supplied.



If you select the **User-Defined Interphase Mass Flux** phase interaction model, and the Phase Coupled Fluid Energy model is already activated, then select the Interphase Energy Transfer phase interaction model to make sure that the phase temperatures reach sensible values. The **User-Defined Interphase Mass Flux** model assumes that the effect of interphase mass transfers on the phase temperatures is negligible.

User-Defined Interphase Mass Flux Properties

The **User-Defined Interphase Mass Flux** > **Interphase Mass Flux** node has the following property:

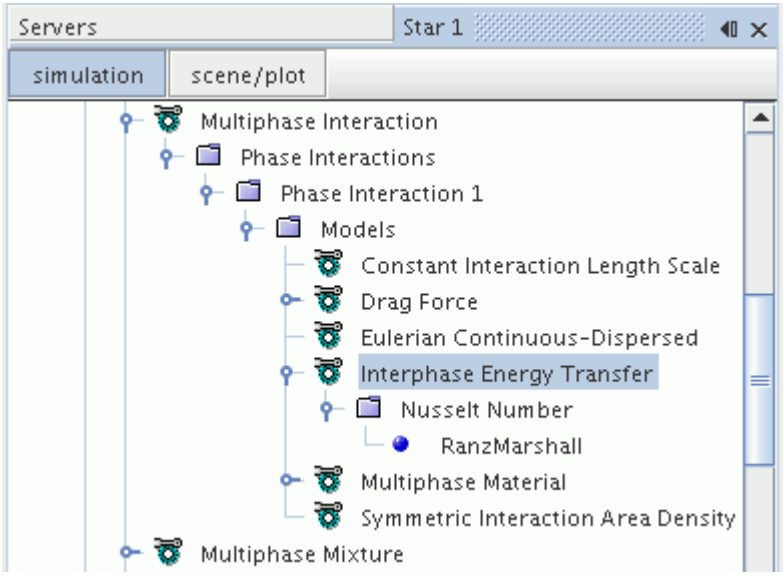
Method

Specifies the method to use in providing the value of mass flux.

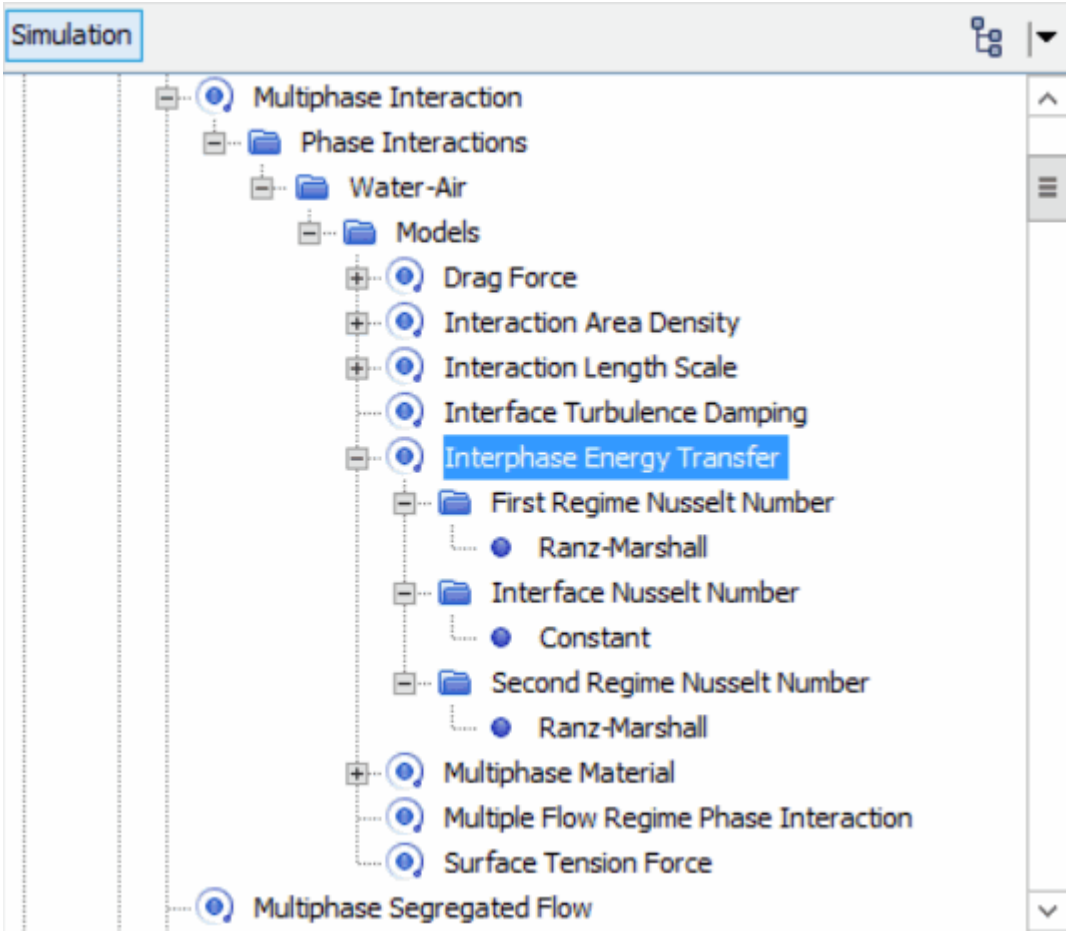
Modeling Interphase Energy Transfer

When the **Phase Coupled Fluid Energy** model is active, you can choose to model energy transfer between Eulerian phases. Activating the **Interphase Energy Transfer** model for a phase interaction permits this energy transfer between the two phases that are selected on the interaction.

For a Eulerian Continuous-Dispersed phase interaction, the **Interphase Energy Transfer** node has a single sub-node for the Nusselt number.



For a Multiple Flow Regime phase interaction, the **Interphase Energy Transfer** node has sub-nodes for each regime. You set the Nusselt number independently for the first regime, the interface regime, and the second regime.



Contents:

Setting the Nusselt Number

Setting the Nusselt Number

The Nusselt number (Nu) is the ratio of convective to conductive heat transfer across the boundary between two fluid phases (or a solid and a fluid phase). In laminar flow, the heat transfer from convection and conduction are of similar magnitude, so the Nusselt number is close to unity. The Nusselt number increases with more active convection. Typical values in turbulent flow are 100–1000, depending on the Reynolds number and other properties, such as the Prandtl number, Pr .

The Nusselt Number is used in the following calculations:

- Interphase Energy Transfer using the **Interphase Energy Transfer** model of a Continuous-Dispersed phase interaction.

  The Nusselt Number is Nu in Eqn. (2353).
- Interphase Energy Transfer using the **Interphase Energy Transfer** model of a Multiple Flow Regime phase interaction.

The First Regime Nusselt Number is Nu_{fr} in Eqn. (2356), the Second Regime Nusselt Number is Nu_{sr} in Eqn. (2357), and the Interface Nusselt Number is Nu_{ir} in Eqn. (2358).

The *First Regime Nusselt Number* and the *Second Regime Nusselt Number* correspond to the *Continuous Phase Nusselt Number* and the *Dispersed Phase Nusselt Number* below. The *Interface Nusselt Number* is specified for both the liquid side and the vapor side.

- Boiling and Condensation using the **Boiling Mass Transfer Rate** model in a Continuous-Dispersed phase interaction.

  The Continuous Phase Nusselt Number (Nu_i in Eqn. (2377)) controls the heat transfer rate from the continuous phase to the dispersed phase (see Eqn. (2375)).

  The Disperse Phase Nusselt Number (Nu_j in Eqn. (2378)) controls the heat transfer rate from the dispersed phase to the continuous phase (see Eqn. (2376)).
- Boiling and Condensation using the **Boiling/Condensation** model in a Multiple Flow Regime phase interaction.

  A total of six Nusselt Numbers are specified: Nu values on both the liquid side and the vapor side for the First Regime, the Interface Regime, and the Second Regime (see Eqn. (2381)).
- Droplet Evaporation (**Droplet Evaporation Mass Transfer Rate** and **Multicomponent Droplet Evaporation Mass Transfer Rate** models)

  See Evaporation and Condensation.
- Solute Crystallization (**Single Component Crystal Growth** model)

  See Eqn. (2645) (continuous phase).
- Melt Crystallization (**Single Component Crystal Growth From Melt** model)

  See Eqn. (2648) (continuous phase).

Nusselt Number Properties

The Nusselt Number has the following properties:

Dimensions

Dimensionless.

Method

The method to use for specifying the Nusselt number that is used in the heat transfer rate calculation.

Method	Corresponding Method Node
Field Function (Heat Transfer Coefficient) Specifies the interphase heat transfer coefficient between liquid and vapor (W/m^2-K).	**Field Function (Heat Transfer Coefficient)**
Ranz-Marshall The Nusselt number is computed using the Ranz-Marshall correlation (Eqn. (2354)).	**Ranz-Marshall**

This method is not available for the Interface regime of a Multiple Flow Regime phase interaction.	
Hughes-Duffey This method is available only in a Multiple Flow Regime phase interaction, and applies only to the liquid side of the interface regime. This method is available only when the primary phase has a turbulence model activated.	**Hughes-Duffey** The liquid side heat transfer coefficient in the interface regime is calculated using the surface renewal theory that was introduced by Hughes and Duffey ([439]), Eqn. (2381). The **Hughes-Duffey** node has the following property: • *Calibration Factor* A calibration factor to adjust the Hughes-Duffey heat transfer coefficient. Increase this factor to increase the heat transfer coefficient value.
Armenante-Kirwan The model for mass transfer to micro-particles that was proposed by Armenante and Kirwan ([400]). This method is available only for turbulent flow regimes.	**Armenante-Kirwan** The **Armenante-Kirwan** node has the following properties: • *Turbulent Reynolds Number Exponent* This value is β in Eqn. (2649). • *Prandtl Number Exponent* This value is γ in Eqn. (2649). • *Relative Density Difference Exponent* This value is δ in Eqn. (2649). • *Calibration Factor* This value is α in Eqn. (2649).

Modeling Interphase Reactions

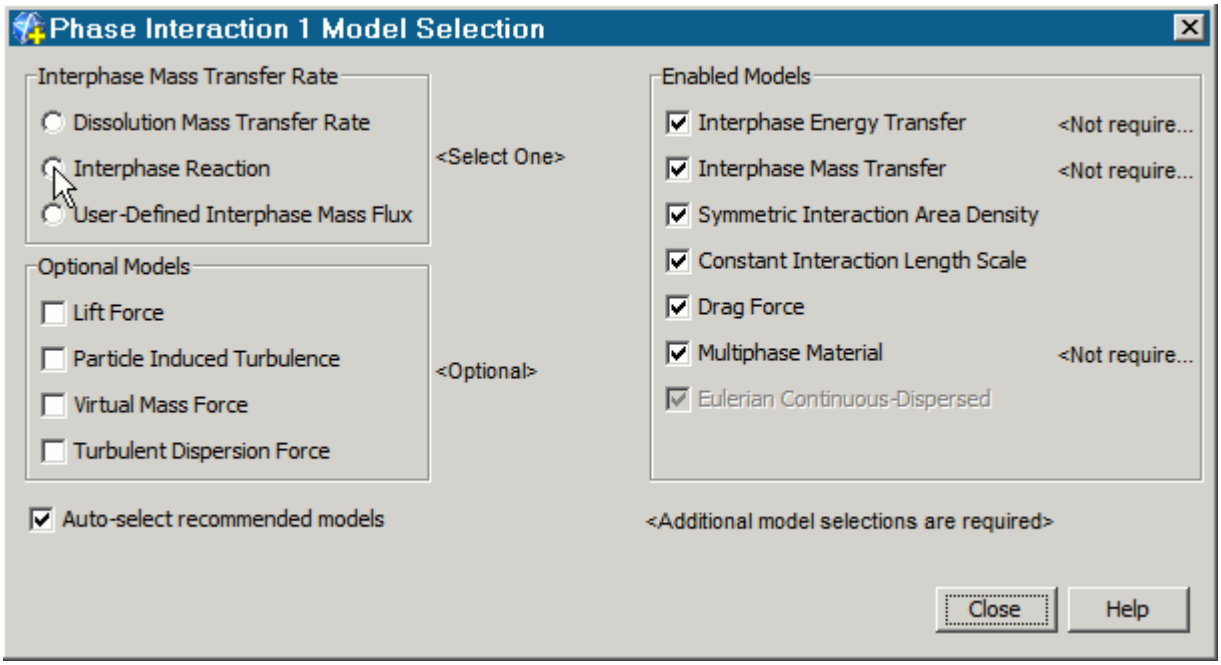
You can model mass transfer between Eulerian phases as a reaction between the phases.

To make the **Interphase Reaction** model available:

1. Define both of the phases in the interaction.

   The phases must have more than one component--usually one multi-component liquid and one multi-component gas.
2. Specify the continuous and dispersed phases.
3. Select the Interphase Mass Transfer model.

The **Interphase Reaction** model can be selected from the *Interphase Mass Transfer Rate* group box.



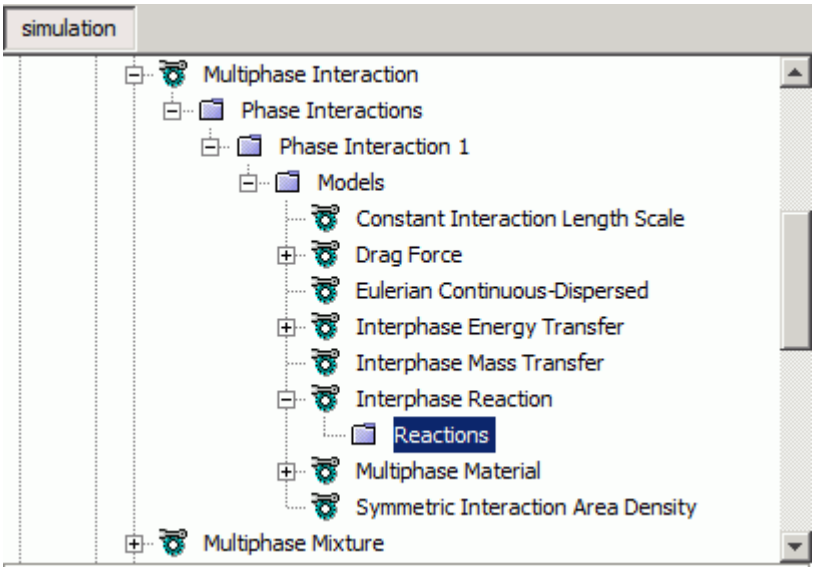
Contents:

Setting Up Interphase Reactions
Specifying the Reaction Component Parameters

Setting Up Interphase Reactions

The **Interphase Reaction** node contains a **Reactions** sub-node. Rather than using properties, this sub-node has a pop-up menu with which you can create objects for setting up interphase reactions.

The **Reactions** sub-node for the Interphase Reaction model is shown below.



Reactions Pop-Up Menu

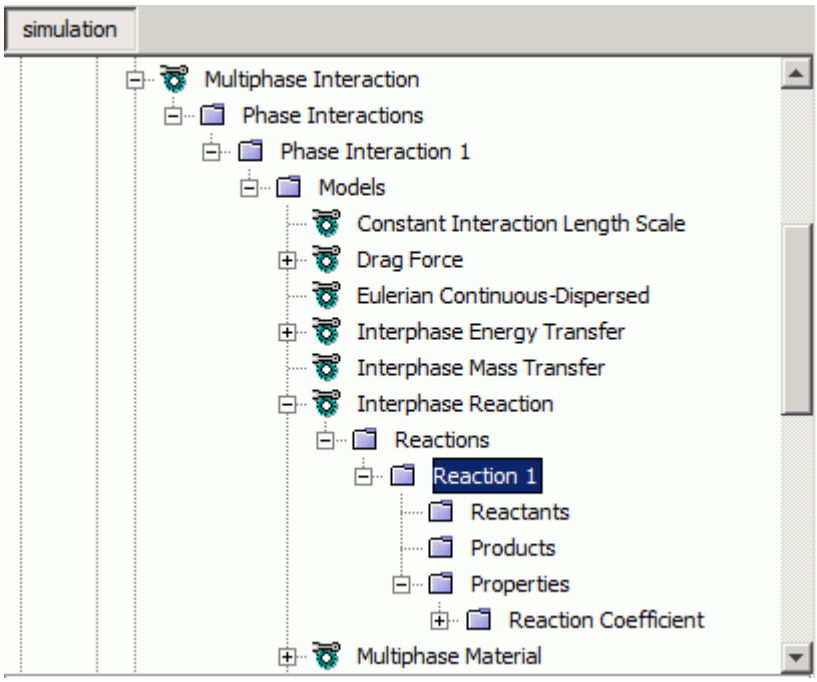
New Reaction	Creates an interphase reaction. This new object contains sub-nodes for reactants, products, and the reaction coefficient.
Import Reactions	Lets you import a reaction from the Material Database.

You use the **Reactions** node to define the reactions that occur between the phase components.

To set up an interphase reaction:

1. Right-click the **Reactions** node and select **New Reaction**.

   A new reaction definition is added.



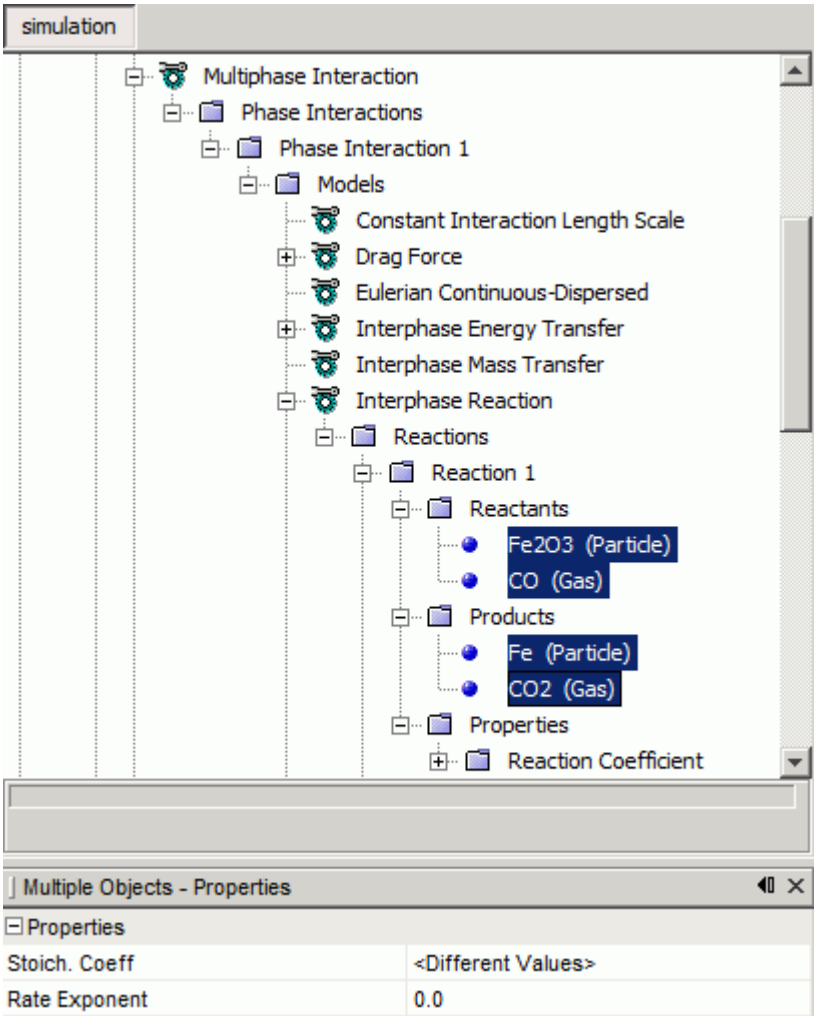
2. In the appropriate nodes, specify the reactants and products.

   This procedure is similar to the corresponding procedure for Lagrangian particle reactions.
3. Specify the properties of the reaction components.
4. In the **Properties** > **Reaction Coefficient** node, specify the appropriate reaction rate properties.

   This procedure is similar to the corresponding procedure for Lagrangian particle reactions.

Specifying the Reaction Component Parameters

A reaction component is any constituent of a multi-component mixture that has already been defined as a reactant or product in an interphase reaction equation.



To specify the required parameters for a particular reaction component:

1. Select the component.
2. In the *Properties* window, specify the appropriate values:

Stoich. Coeff.	Set the stoichiometric coefficient that is appropriate to the current reaction.
Rate Exponent	Set the rate exponent that is appropriate to the current reaction.

Working with Segregated Multiphase Flow Solvers

When setting up a multiphase flow case it is often necessary to adjust the solver settings.

Two solvers are invoked when the Multiphase Segregated Flow model is selected:

- Multiphase Segregated Flow solver
- Volume Fraction solver

Recommended settings are discussed in the section Setting Up Solvers for Multiphase Flow Cases.

Contents:

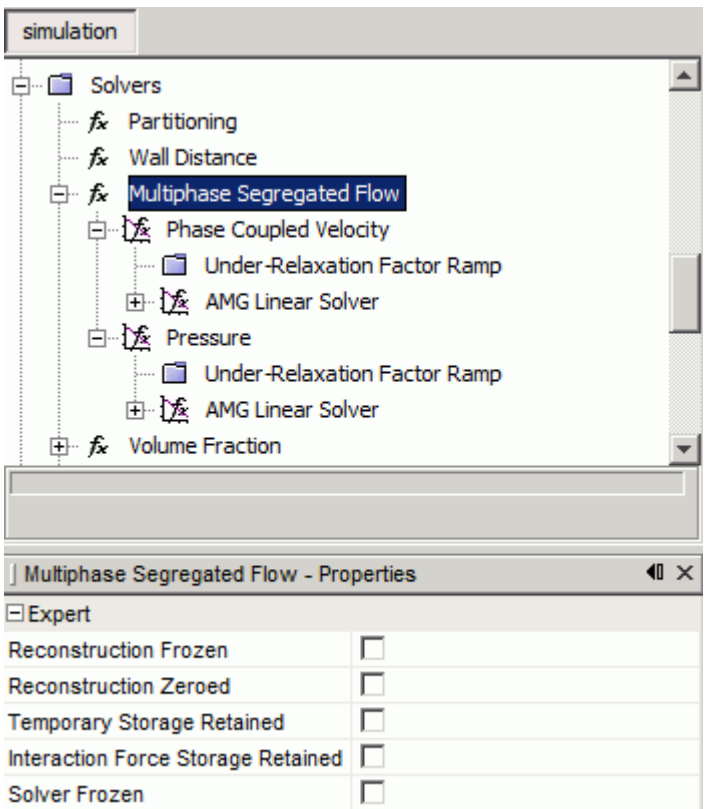
What Is the Multiphase Segregated Flow Solver?
What Is the Phase Coupled Velocity Solver for Multiphase Flow?
What Is the Pressure Solver for Multiphase Flow?
Segregated Energy Solver for Multiphase
What Is the Volume Fraction Solver for Multiphase?
Setting Up Solvers for Multiphase Flow Cases

What Is the Multiphase Segregated Flow Solver?

The Multiphase Segregated Flow solver controls the solution update for the Multiphase Segregated Flow model. It is responsible for solving the velocity field for each individual phase and the overall pressure solution.

The Multiphase Segregated Flow solver provides two more solvers:

- Phase Coupled Velocity
- Pressure



Multiphase Segregated Flow Solver Properties

The properties in the *Expert* category are for a temporary debug situation, at the expense of simulation accuracy and higher memory usage. Do not change any of these properties unless you are thoroughly familiar with the discretization techniques used in STAR-CCM+.

Interaction Force Storage Retained

Allows you to retain the information that is stored for each phase interaction force model at the end of the iteration. These quantities become available as field functions during subsequent iterations.

You can use this option when you want to visualize the phase interaction forces, such as drag, turbulent drag, lift, virtual mass, and wall lubrication.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

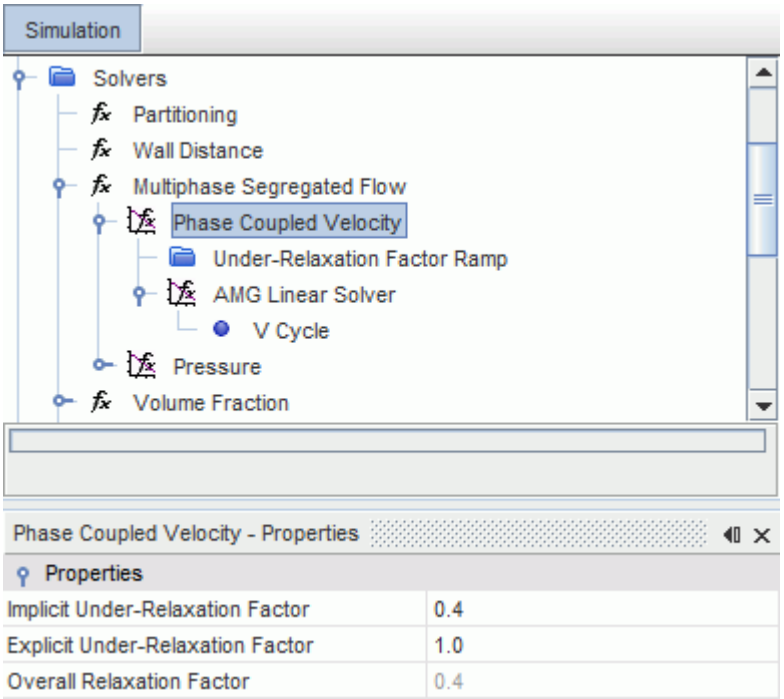
Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

What Is the Phase Coupled Velocity Solver for Multiphase Flow?

The phase coupled velocity solver solves the discrete conservation equations to obtain the intermediate velocity field for all the phases.

The corresponding **Phase Coupled Velocity Solver** node controls the under-relaxation factor and algebraic multigrid parameters for the Eulerian phase momentum equations.



The equations for velocity component correction are solved for all phases simultaneously. One advantage of this solver is that it permits greater robustness in solving multiphase flows with high levels of phase interaction.

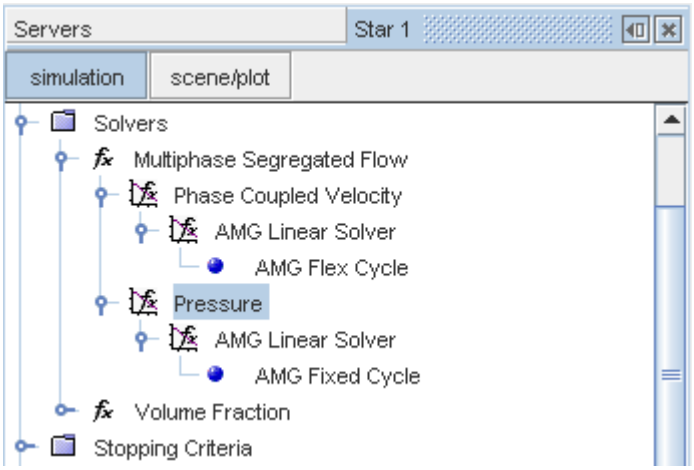
Phase Coupled Velocity Solver Properties

Implicit Under-Relaxation Factor	Improves stability and convergence of the linear system by using the relaxation factor to increase diagonal dominance of the matrix. See Setting Up Solvers for Multiphase Flow Cases. The default value is 0.7.
Explicit Under-Relaxation Factor	This property specifies the multiplier that is applied to the provisional increment of the solution. Sharing the relaxation between implicit and explicit factors can be used to improve performance. See Setting Up Solvers for Multiphase Flow Cases. The default value is 1.0.
Overall Relaxation Factor	The product of the two under-relaxation factors (that is, **Implicit Under-Relaxation Factor** times **Explicit Under-Relaxation Factor**). This (read-only) value is a guide to the combined effect of the relaxation. See Setting Up Solvers for Multiphase Flow Cases.

What Is the Pressure Solver for Multiphase Flow?

The Pressure solver controls the under-relaxation factor and algebraic multigrid parameters for the pressure correction equation.

The Pressure solver is controlled by the Multiphase Segregated Flow solver.



Pressure Solver Properties

Under-Relaxation Factor	At each iteration, this property governs the extent to which the newly computed solution supplants the old solution. The default value is 0.3.

Segregated Energy Solver for Multiphase

The Segregated Energy solver controls the solution update for the Segregated Fluid Energy model. It is used to set the under-relaxation factor and algebraic multigrid parameters for the energy equation.

Segregated Energy Solver Properties

Fluid Under-Relaxation Factor

At each iteration, this property governs the extent to which the newly computed solution supplants the old solution. The default value is 0.9.

Solid Under-Relaxation Factor

At each iteration, this property governs the extent to which the newly computed solution supplants the old solution. The default value is 0.99.

Enable High-Accuracy Temporal Discretization

Activates the High-Accuracy Temporal Discretization options for the solver.

For more information, see Setting High-Accuracy Temporal Discretization.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

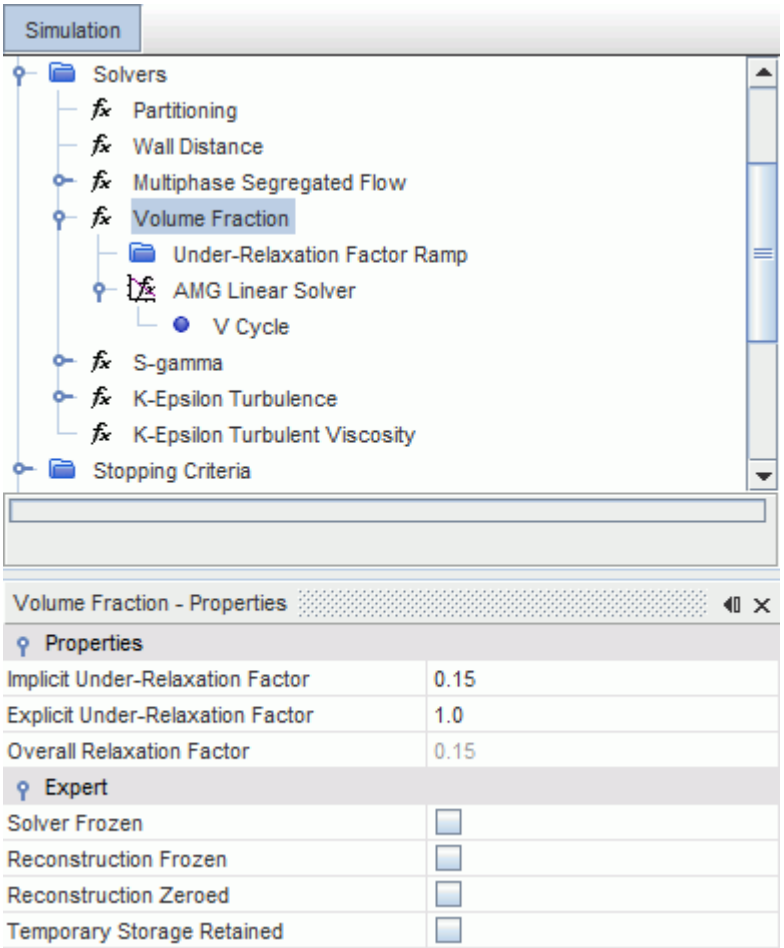
Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

What Is the Volume Fraction Solver for Multiphase?

The Volume Fraction solver controls the solution update for the volume fractions of all the phases.

The discretized form of the continuity equation (Eqn. (2230)), is solved for each phase.



To make sure that Eqn. (2229) is satisfied after solving the volume fraction equations for all phases, the volume fraction of each phase in a cell is normalized by the sum of the volume fractions of all the phases in that cell.

Volume Fraction Solver Properties

Implicit Under-Relaxation Factor	Improves stability and convergence of the linear system by using the relaxation factor to increase diagonal dominance of the matrix. See Setting Up Solvers for Multiphase Flow Cases. The default value is 0.5.
Explicit Under-Relaxation Factor	This property specifies the multiplier that is applied to the provisional increment of the solution. Sharing the relaxation between implicit and explicit factors can be used to improve performance. See Setting Up Solvers for Multiphase Flow Cases. The default value is 1.0.

Overall Relaxation Factor	The product of the two under-relaxation factors (that is, **Implicit Under-Relaxation Factor** times **Explicit Under-Relaxation Factor**). This (read-only) value is a guide to the combined effect of the relaxation. See Setting Up Solvers for Multiphase Flow Cases.

The default settings use only the implicit under-relaxation factor for robustness. In some scenarios, such as a fine mesh, it can take a while for the Volume Fraction to propagate through the system. In these cases, you can split the under-relaxation property value between the two factors. For example, to specify an overall relaxation factor = 0.15, you can specify the combination **Implicit Under-Relaxation Factor** = 0.6, **Explicit Under-Relaxation Factor** = 0.25. This combination can help the volume fraction to converge faster without upsetting the overall convergence of the simulation.

The same approach can be used for S-Gamma under-relaxation. In general, it is good practice to use the same under-relaxation factors for the Volume Fraction and S-Gamma equations since their ratio is a key parameter in interphase interactions.

Volume Fraction Solver Expert Properties

The properties in the *Expert* category are for a temporary debug situation, at the expense of simulation accuracy and higher memory usage. Do not change these properties unless you are thoroughly familiar with the STAR-CCM+ discretization techniques.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

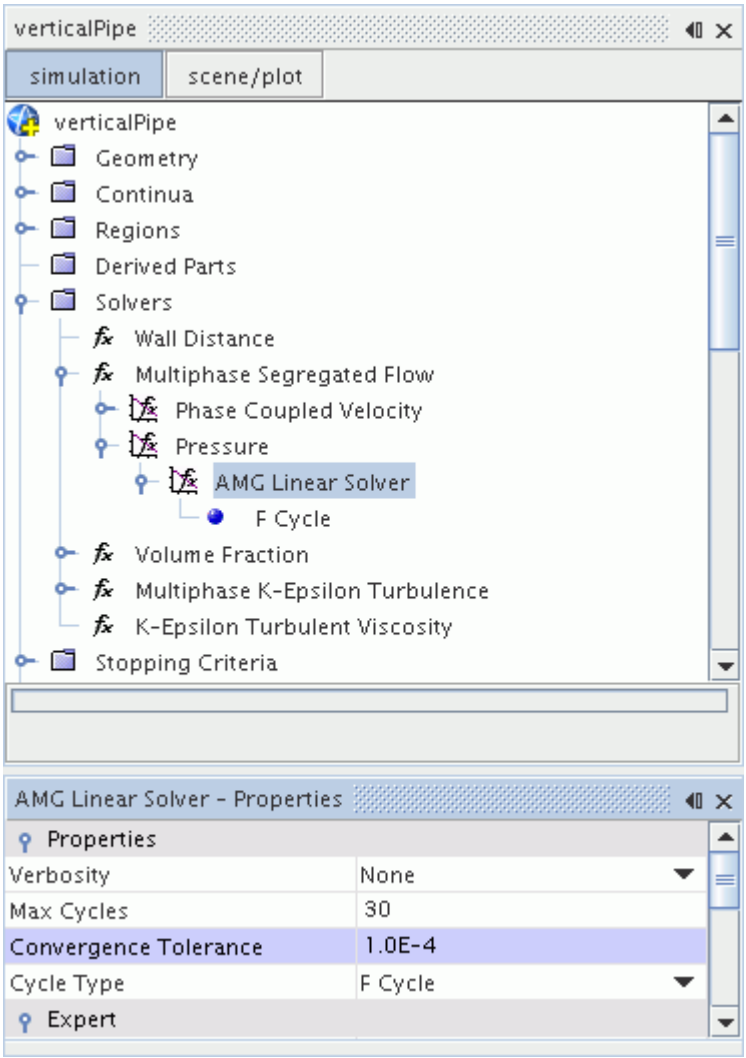
Setting Up Solvers for Multiphase Flow Cases

The default relaxation factors are chosen to give reliable convergence for common multiphase flow cases.

If the default values are insufficient for convergence of steady state cases, for example due to strong force or source terms, you can reduce the relaxation factors. The relaxation factors can be reduced as required towards 0.1 for each solver in turn, starting with the velocity or volume fraction solver. You can use the response of the residuals as a guide to the effectiveness of each adjustment. For relaxation factors below 0.1, convergence can be slow. In these cases, it can be more effective to approach steady state using time-stepping, especially if a physical timescale can be resolved for the term that is preventing convergence or causing unsteadiness. Reducing the time-step size is the first thing that you can try when attempting to resolve difficulties in unsteady simulations.

Alternatively, if the rate of convergence is too slow, you can improve performance by transferring some of the relaxation from the Implicit Under-Relaxation Factor (see Eqn. (1625)) to the Explicit Under-Relaxation Factor. An Explicit Under-Relaxation Factor is available for the Phase Coupled Velocity, Volume Fraction, and S-Gamma solvers.

Unusually, some cases require you to decrease the linear solver convergence tolerance. This setting is a property of the **AMG Linear Solver** component of each solver (the following example shows where this property is located for the **Pressure** solver). The default value for the *Convergence Tolerance* is 0.1, but this value can be reduced to between 1.0E-3 and 1.0E-4 in most cases. You can investigate the justification for, and the effect of, this adjustment by changing the *Verbosity* property from **None** to **Low** and then examining the diagnostics output from each solver over a single iteration.



Explicit Under-Relaxation for Multiphase Solvers

If a multiphase study requires extra under-relaxation for convergence, you can improve performance by using the Explicit Under Relaxation factor for the additional amount.

One strategy is to retain the Implicit Under-Relaxation Factor at its default value, or up to 0.8, and reduce the Explicit Under-Relaxation Factor. The product of the two factors then gives the overall relaxation required.

You are best to apply Explicit Under-Relaxation to both Velocity and Volume Fraction Solvers. Where the S-Gamma model is active, use the same relaxation scheme for the S-Gamma solver as for the Volume Fraction solver.

Steady-state flows that are specified with inlet boundary conditions benefit most from Explicit Under-Relaxation. In these cases, you can save up to 30% in the number of iterations that are required for convergence.

Modeling Turbulence in Multiphase Segregated Flow

STAR-CCM+ supports turbulence modeling in multiphase segregated flows.

The presence of a dispersed phase can modify the turbulence of a continuous phase in a number of ways:

- Shear production is modified through the effect of buoyancy, drag, and other forces on velocity profiles.
- The passage of the dispersed phase particles generates velocity fluctuations in the continuous phase. The wake of the dispersed phase particles is a source of turbulence.
- The interaction of velocity fluctuations and volume fraction fluctuations can generate further turbulence enhancement or damping terms.

STAR-CCM+ allows you to model the effects of turbulence on continuous and dispersed phases in several ways:

- Continuous and dispersed phases can each be modeled with their own set of equations for the turbulence energy and dissipation. You can model phases independently as laminar, turbulent or with different turbulence models.

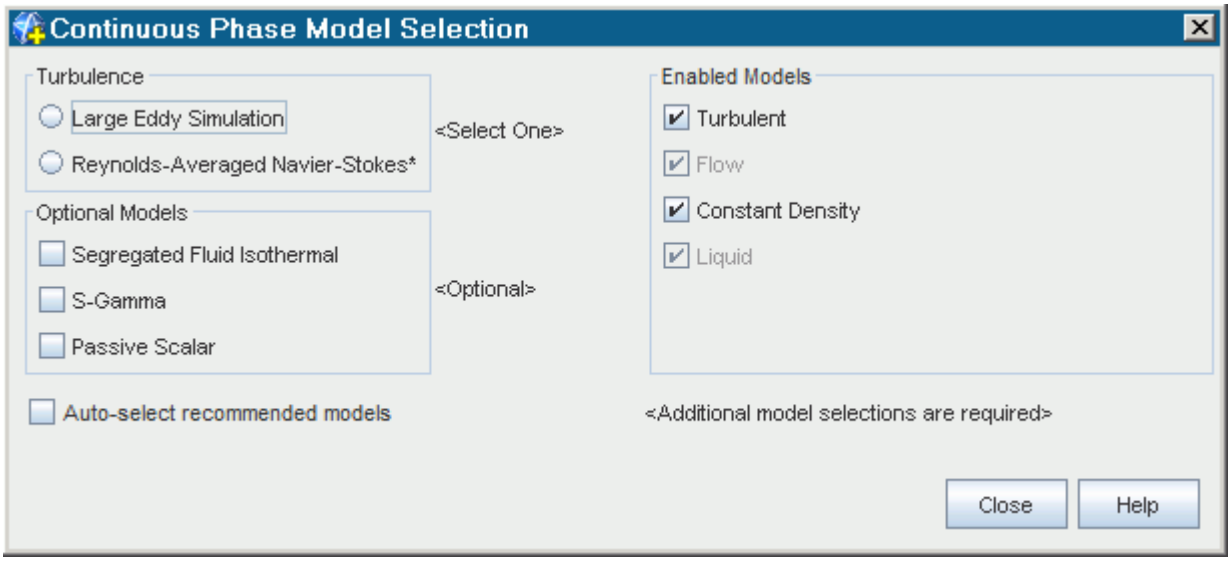
  The turbulence equations are identical to the single phase formulation, except each phase is scaled with a factor of the volume fraction of that phase. The same phase volume fraction also modifies convection and diffusion fluxes of phase-turbulence quantities.
- The turbulence of the dispersed phase can be calculated from the turbulence of the continuous phase using the Turbulence Response Model. This algebraic model reduces the computations that are required as only the continuous phase equations are solved. However, the model couples the turbulence of dispersed phases to the continuous ones.
- The phases can also be coupled through bubble/particle induced turbulence of the dispersed phase which adds more source terms to the continuous phase turbulence energy and dissipation equations.
- In addition to its contribution to the model turbulence, particle-generated turbulence can also be dissipated locally at the particle scale. This effect can be thought as a local enhancement of the mixing and is accounted for by the Particle Induced Mixing Model.
- The transfer of turbulence from one phase to another is accounted for by the Interphase Turbulence Transfer model.

STAR-CCM+ supports Large Eddy Simulation (LES) turbulence in multiphase flows. Large Eddy Simulation is a technique intermediate between the direct numerical simulation of turbulent flows and the solution of the Reynolds-Averaged Navier-Stokes (RANS) equations. In LES the contribution of the large, energy-carrying structures to momentum and energy transfer is computed exactly, and only the effect of the smallest scales of turbulence is modeled. Since the small scales tend to be more homogeneous, isotropic, and universal, and less affected by the boundary conditions than the large scales, the LES models can be simpler and require fewer adjustments when applied to different flows than similar models for the RANS equations. A good application for LES in a multiphase flow is a bubbly flow.

STAR-CCM+ supports the following Sub-Grid Scale models for LES:

- Smagorinsky Sub-Grid Scale
- Dynamic Smagorinsky Sub-Grid Scale
- WALE (Wall Adapting Local Eddy-Viscosity) Sub-Grid Scale

You can choose between Reynolds-Averaged Navier-Stokes turbulence and Large Eddy Simulation turbulence for both the continuous phase and the dispersed phase.



Continuous Phase Turbulence Models

The models that are available for modeling Reynolds-Averaged Navier-Stokes turbulence in the continuous phase are as follows:

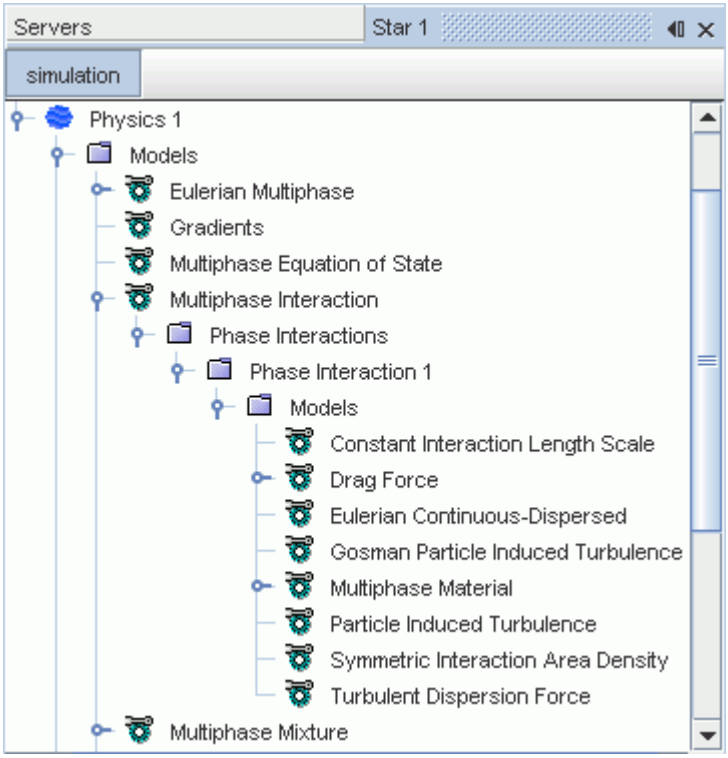
- K-Epsilon Turbulence options:
  - EB K-Epsilon Model, with All y+ Wall Treatment
  - EB K-Epsilon Model, with Low y+ Wall Treatment
  - Realizable K-Epsilon Model, with High y+ Wall Treatment
  - Realizable K-Epsilon Two-Layer Model, with All y+ Wall Treatment
  - Standard K-Epsilon Model, with High y+ Wall Treatment
  - Standard K-Epsilon Low-Re Model, with All y+ Wall Treatment
  - Standard K-Epsilon Low-Re Model, with Low y+ Wall Treatment
  - Standard K-Epsilon Two-Layer Model, with All y+ Wall Treatment
- K-Omega Turbulence options:
  - SST (Menter) K-Omega, with All y+ Wall Treatment
  - SST (Menter) K-Omega, with High y+ Wall Treatment
  - SST (Menter) K-Omega, with Low y+ Wall Treatment
  - Standard (Wilcox) K-Omega, with All y+ Wall Treatment
  - Standard (Wilcox) K-Omega, with High y+ Wall Treatment
  - Standard (Wilcox) K-Omega, with Low y+ Wall Treatment
- Reynolds Stress Turbulence options:
  - Elliptic Blending Model, with All-y+ Wall Treatment
  - Linear Pressure Strain Model, with High-y+ Wall Treatment
  - Linear Pressure Strain Two-Layer Model, with Two-Layer All y+ Wall Treatment
  - Quadratic Pressure Strain Model, with High-y+ Wall Treatment

The models that are available for modeling Large Eddy Simulation turbulence in the continuous phase are:

- Dynamic Smagorinsky Subgrid Scale, with All y+ Wall Treatment
- Dynamic Smagorinsky Subgrid Scale, with Low y+ Wall Treatment
- Smagorinsky Subgrid Scale, with All y+ Wall Treatment
- Smagorinsky Subgrid Scale, with Low y+ Wall Treatment
- WALE Subgrid Scale, with All y+ Wall Treatment
- WALE Subgrid Scale, with Low y+ Wall Treatment

You can expand the phase-based turbulence models to use extra bubble/particle induced turbulence models that exist at the phase interaction level:

- Gosman
- Troshko Hassan
- Tchen
- LES Particle Induced Turbulence model
- The Particle Induced Mixing Model contributes to the effective viscosity of the continuous phase. Two alternatives can be selected at the phase interaction level:
  - Sato. This is a recognised model for bubble induced mixing.
  - User-defined function



Dispersed Phase Turbulence Models

The dispersed phase can use the Laminar model instead of a turbulence model. However, if you expect the dispersed phase to carry a significant proportion of the wall stress or wall heat transfer, do not use the Laminar model.

The models that are available for modeling Reynolds-Averaged Navier-Stokes turbulence in the dispersed phase are:

- K-Epsilon Turbulence options:
  - EB K-Epsilon Model, with All y+ Wall Treatment
  - EB K-Epsilon Model, with Low y+ Wall Treatment
  - Realizable K-Epsilon Model, with High y+ Wall Treatment
  - Realizable K-Epsilon Two-Layer Model, with All y+ Wall Treatment
  - Standard K-Epsilon Model, with High y+ Wall Treatment
  - Standard K-Epsilon Low-Re Model, with All y+ Wall Treatment
  - Standard K-Epsilon Low-Re Model, with Low y+ Wall Treatment
  - Standard K-Epsilon Two-Layer Model, with All y+ Wall Treatment
- K-Omega Turbulence options:
  - SST (Menter) K-Omega, with All y+ Wall Treatment
  - SST (Menter) K-Omega, with High y+ Wall Treatment
  - SST (Menter) K-Omega, with Low y+ Wall Treatment
  - Standard (Wilcox) K-Omega, with All y+ Wall Treatment
  - Standard (Wilcox) K-Omega, with High y+ Wall Treatment
  - Standard (Wilcox) K-Omega, with Low y+ Wall Treatment
- Reynolds Stress Turbulence options:
  - Elliptic Blending Model, with All-y+ Wall Treatment
  - Linear Pressure Strain Model, with High-y+ Wall Treatment
  - Linear Pressure Strain Two-Layer Model, with Two-Layer All y+ Wall Treatment
  - Quadratic Pressure Strain Model, with High-y+ Wall Treatment
- Turbulence Response Model
  - Issa Turbulence Response
  - Tchen Turbulence Response
  - User-defined Turbulence Response

The models that are available for modeling Large Eddy Simulation turbulence in the dispersed phase are listed below:

- Dynamic Smagorinsky Subgrid Scale, with All y+ Wall Treatment
- Dynamic Smagorinsky Subgrid Scale, with Low y+ Wall Treatment
- Smagorinsky Subgrid Scale, with All y+ Wall Treatment
- Smagorinsky Subgrid Scale, with Low y+ Wall Treatment
- WALE Subgrid Scale, with All y+ Wall Treatment
- WALE Subgrid Scale, with Low y+ Wall Treatment

These models were derived for two-phase flow, but are also applicable for dilute mixtures of a number of dispersed phases.

Contents:

Turbulence Response Model
Particle Induced Turbulence Models
LES Particle Induced Turbulence
Particle Induced Mixing Models
Interphase Turbulence Transfer Model

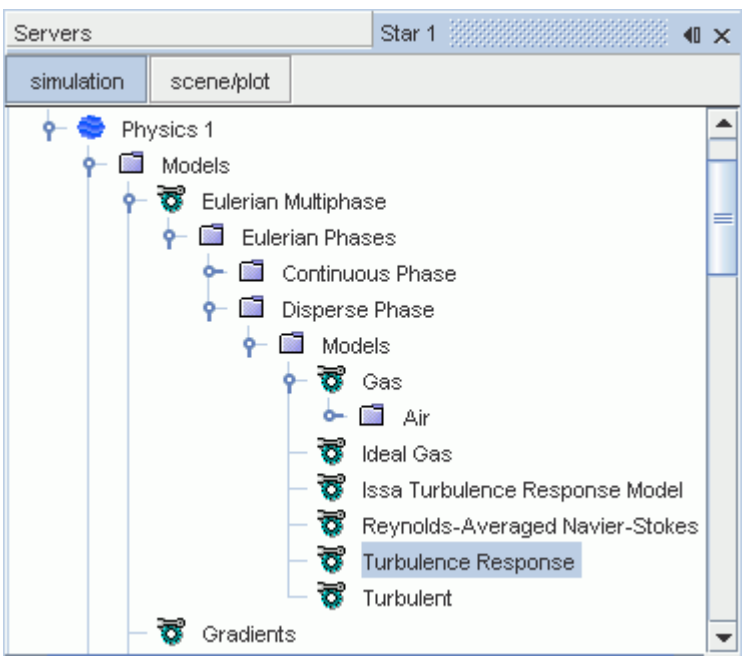
See Also:

K-Epsilon Turbulence
Reynolds Stress Transport (RST) Turbulence
Large Eddy Simulation (LES)

Turbulence Response Model

The Turbulence Response Model predicts velocity fluctuations in the dispersed phase using algebraic correlations to the velocity fluctuations in the continuous phase

For more information, see Eqn. (2676).



Selecting Turbulence Response

To setup a simulation with turbulence response:

1. Complete the basic setup for Multiphase Segregated Flow, with one or more Eulerian phases.

2. For each Eulerian phase in which you wish to activate a turbulence response model:
   a. Right-click the **[Phase]** > **[Models]** node and choose **Select Models**.
   b. In the *Phase Model Selection* dialog, select the following models:

Group Box	Model
Viscous Regime	**Turbulent**
Turbulence	**Reynolds-Averaged Navier-Stokes** (selected automatically)
Reynolds-Averaged Turbulence	**Turbulence Response**
Turbulence Response Models	One of the following models: • **Issa Turbulence Response Model** • **Tchen Turbulence Response Model** • **User-Defined Turbulence Response Model**

   c. When done with selecting phase models, select the node for the chosen turbulence response model and set its properties.

      For details, see the individual properties tables below.

Once you have selected one of the turbulence response models, a Response Coefficient Turbulent Viscosity node is added to the **Solvers** node.

Issa Turbulence Response Model

The Issa Turbulence Response model is defined as a correlation for the turbulence response coefficient with a volume fraction correction.

Issa Turbulence Response Model Properties

Phase Interaction	The drop-down list contains all of the phase interactions that use this phase as a dispersed phase. Select the phase interaction that owns the continuous phase turbulence model that you want to use. The Issa Turbulence Response model uses the appropriate turbulence model to calculate the turbulent quantities.	
Normal Stress Term	An explicit term that directly incorporates divergence and turbulent kinetic energy into the stress tensor according to the full Boussinesq approximation. This term often has a negligible effect in incompressible single-phase flows, but use it with care in Eulerian multiphase flows where it can cause a strong turbophoretic effect.	
	Activated	Use the term.
	Deactivated	Do not use the term (the default).

User-Defined Turbulence Response Model

The User-Defined Turbulence Response model allows you to define the turbulence response coefficient. When the turbulence response coefficient is equal to 1, the dispersed phase fluctuates in tandem with the continuous phase.

User-Defined Turbulence Response Model Properties

Phase Interaction	The drop-down list contains all of the phase interactions that use this phase as a dispersed phase. Select the phase interaction that owns the continuous phase turbulence model that you want to use. The User-Defined Turbulence Response model uses the appropriate turbulence model to calculate the turbulent quantities.	
Normal Stress Term	An explicit term that directly incorporates divergence and turbulent kinetic energy into the stress tensor according to the full Boussinesq approximation. This term often has a negligible effect in incompressible single-phase flows, but use it with care in Eulerian multiphase flows where it can cause a strong turbophoretic effect.	
	Activated	Use the term.
	Deactivated	Do not use the term (the default).

Turbulence Response Model Properties

Method	Selects the method to use in specifying the turbulence response coefficient.

Tchen Turbulence Response Model

The Tchen Turbulence Response model provides a dispersed phase turbulent diffusivity for gas flows laden with heavy particles (see Eqn. (2686)).

When the Tchen Turbulence Response model is selected for a phase, you must ensure that Virtual Mass Coefficient model is selected under the corresponding phase interaction. This model is usually selected automatically in a turbulent flow, since the Virtual Mass Coefficient is also a required model for the Turbulent Dispersion Force model.

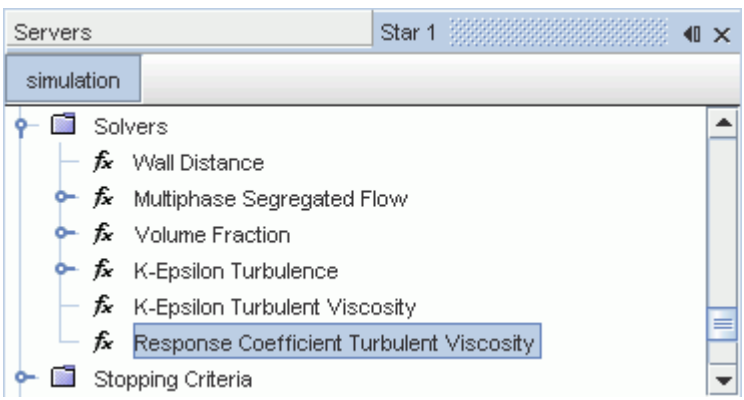
Tchen Turbulence Response Model Properties

Phase Interaction	The drop-down list contains all of the phase interactions that use this phase as a dispersed phase. Select the phase interaction that owns the continuous phase turbulence model that you want to use. The Tchen Turbulence Response model uses the appropriate turbulence model to calculate the turbulent quantities.

Turbulent Prandtl Number	The coefficient σ_0 in Eqn. (2745), with a default value of 1.0, representing basic passive diffusivity.
Crossing Trajectories Coefficient	The calibration coefficient C_β in Eqn. (2745), with a default value of 1.8.
Normal Stress Term	An explicit term that directly incorporates divergence and turbulent kinetic energy into the stress tensor according to the full Boussinesq approximation. This term often has a negligible effect in incompressible single-phase flows, but use it with care in Eulerian multiphase flows where it can cause a strong turbophoretic effect.

Response Coefficient Turbulent Viscosity Solver

The **Response Coefficient Turbulent Viscosity** node is added to the **Solvers** node when a turbulence response model is selected. It governs the update of the turbulent viscosity resulting from the presence of a dispersed phase.



Response Coefficient Turbulent Viscosity Solver Properties

Under-Relaxation Factor	Governs the portion of the newly computed turbulent viscosity field that is used to update the turbulent viscosity. The default value of 1 implies that the turbulent viscosity is entirely computed from the new field.
Maximum Ratio	Sets the maximum allowable ratio of turbulent to laminar viscosity. By default it is set to 100,000.
Solver Frozen	Prevents or allows the solver from updating the solution while iterating.

Particle Induced Turbulence Models

The particle induced turbulence models add more source terms to the continuous phase turbulence kinetic energy and dissipation equations. The models represent modification of turbulence due to the presence of a dispersed phase.

The available models are:

- Gosman model
- Troshko and Hassan model
- Tchen model
- LES Particle Induced Turbulence model

Selecting Particle Induced Turbulence

To select a Particle Induced Turbulence model:

1. Complete the setup for Multiphase Segregated Flow, with a minimum of two phases. One phase must be a continuous phase, and another phase must be a dispersed phase.
2. For each Eulerian phase, select an appropriate turbulence model.
3. In the **Multiphase Interaction** > **Phase Interactions** node, create a phase interaction.
4. Right-click the **[Phase Interaction]** node and click **Select Models**.
5. In the *Phase Interaction Model Selection* dialog, select the **Continuous-Dispersed Phase Interaction** model, and then close the dialog.
6. Select the **Phase Interactions** > **[Phase Interaction]** > **Continuous-Dispersed Phase Interaction** node and match the *Continuous Phase* and *Dispersed Phase* properties with the appropriate Eulerian phase.
7. Right-click the **[Phase Interaction]** node and click **Select Models**.
8. In the *Phase Interaction Model Selection* dialog, in the *Optional Models* group box, select **Particle Induced Turbulence**.
9. Optionally, within the *Optional Models* group box, select **Turbulent Dispersion Force** (required by the Gosman Particle Induced Turbulence model).
10. In the *Particle Induced Turbulence Models* group box, select one of the available models.

Gosman Particle Induced Turbulence Model

The *Gosman Particle Induced Turbulence* model adds source terms to the turbulent kinetic energy equation (see Eqn. (2695)) and to the turbulent dissipation equation (see Eqn. (2697)).

Troshko and Hassan Particle Induced Turbulence Model

The **Troshko and Hassan Particle Induced Turbulence** model describes bubble induced turbulence effects. It adds a source term to the turbulent kinetic energy equation (see Eqn. (2688)) and another to the dissipation equation (see Eqn. (2690)). The model uses the Virtual Mass phase interaction model but if the model is not selected the virtual mass coefficients default to the value of a spherical bubble, which is given by Eqn. (2687).

Properties of the Troshko and Hassan Particle Induced Turbulence Model

Troshko and Hassan C3 coefficient	Selects the calibration constant C_3 for the dissipation term in Eqn. (2690). By default this constant is set to 0.45.
Damping coefficient	This parameter is not part of the model derivation, but defaults to 1 and can be used by the user to regulate the proportion of the source term in Eqn. (2689) that is used in the calculation.

Tchen Particle Induced Turbulence Model

The **Tchen Particle Induced Turbulence** model provides source terms to the continuous turbulent kinetic energy equation due to the interaction with particles, based on Tchen's theory.

Properties of the Tchen Particle Induced Turbulence Model

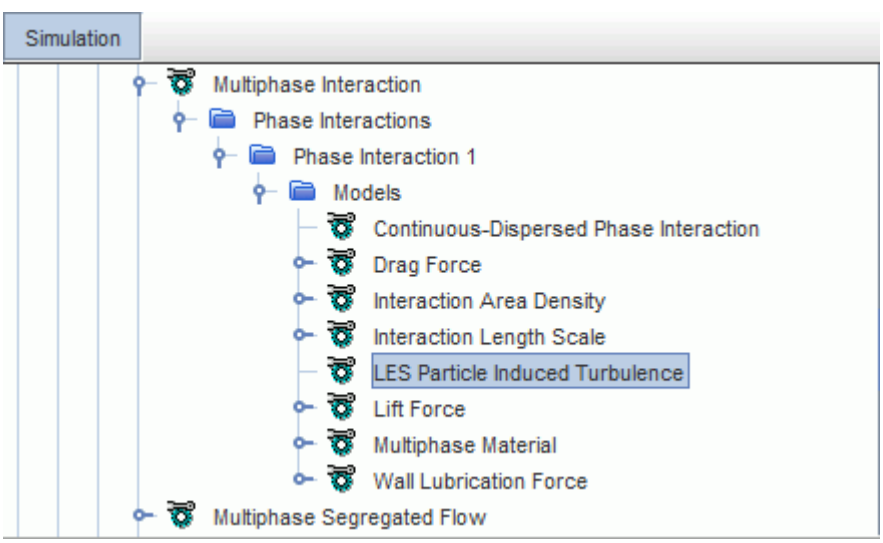
Production Calibration Coefficient	The coefficient C_0 in Eqn. (2701) defining the source term for turbulence energy or for Reynolds stress equations. This parameter is not part of the model derivation, but defaults to 1 and you can adjust it to test the sensitivity of the solution to this term.
Dissipation Calibration Coefficient	The coefficient $C_{\varepsilon 3}$ in Eqn. (2703) defining the corresponding source term for the turbulent dissipation equation. Parameter $C_{\varepsilon 3}$ has a default value of 1.44, and you can adjust it independently of constants for other terms.
Turbulent Prandtl Number	The coefficient σ_0 in Eqn. (2745), with a default value of 1.0, representing basic passive diffusivity.
Crossing Trajectories Coefficient	The calibration coefficient C_β in Eqn. (2745), with a default value of 1.8.

LES Particle Induced Turbulence

The LES Particle Induced Turbulence model represents modification of turbulence due to the presence of a dispersed phase.

The LES Particle Induced Turbulence model is available only when the Large Eddy Simulation model is activated in the physics continuum.

The **LES Particle Induced Turbulence** node has no properties.

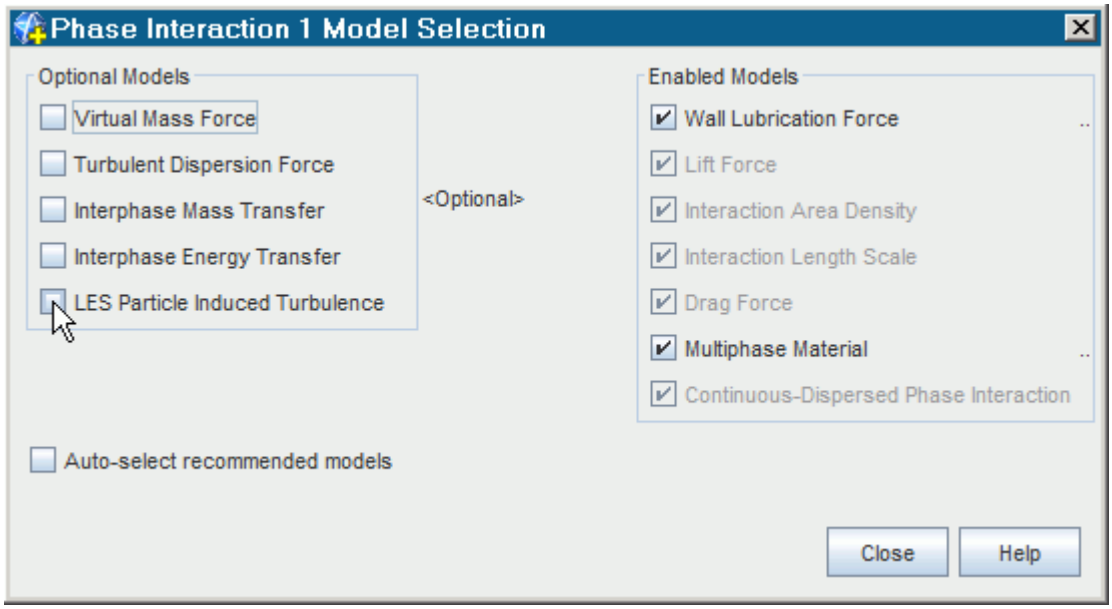


Selecting the LES Particle Induced Turbulence Model

To select the LES Particle Induced Turbulence model:

1. In the Physics continuum, activate the following models:
   - **Three Dimensional**
   - **Eulerian Multiphase**
   - **Implicit Unsteady**
   - **Multiphase Segregated Flow**

     Gradients and **Multiphase Equation of State** are selected automatically.
   - **Turbulent**
2. Under the **Eulerian Multiphase** > **Eulerian Phases** node, create the continuous phase, deactivate **Auto-select recommended models**, and activate the following models:
   - **Turbulent** (selected automatically)
   - **Large Eddy Simulation** from the *Turbulence* group box.
   - The appropriate option from the *Subgrid Scale Turbulence* group box.
   - The appropriate option from the *LES Wall Treatment* group box.
   - The appropriate *Material* option, which must be gas or liquid for the continuous phase.
   - The *Equation of State* option.
3. Create the dispersed phase and activate the same models as for the continuous phase. The *Material* can be a particle, as well as gas or liquid.
4. Under the **Multiphase Interaction** > **Phase Interactions** node, create a phase interaction, open the *Phase Interaction Model Selection* dialog, and activate **Continuous-Dispersed Phase Interaction**. Other required models are selected automatically.
5. Close the *Phase Interaction Model Selection* dialog, and then set the continuous and dispersed phases in the **Continuous-Dispersed Phase Interaction** node.
6. Open the *Phase Interaction Model Selection* dialog again. The **LES Particle Induced Turbulence** model is available for selection.



Particle Induced Mixing Models

The Particle Induced Mixing model contributes to the effective viscosity of the continuous phase.

The Sato model is the simplest and earliest model for particle-induced mixing, and is a robust alternative to later source-based models for particle induced turbulence. This model is applicable when modeling a dilute bubbly flow. For other flow scenarios, the field function method can be used.

In a dilute bubbly flow, the shear-induced turbulence and bubble-induced fluctuations operate at different scales. Figure 11 in Lance and Bataille (1991) [454] gives an example of grid turbulence where bubble-induced fluctuations are dominant but the decay rate of the shear-induced part remains unaffected. This situation is not well-described by the more advanced models that solve transport equations for the combined turbulence energy with one length scale equation. However, the Sato model assumes the shear-induced part is totally independent while the bubble-induced part is always in local equilibrium and does not need a transport equation.

Particle Induced Mixing Model Reference

Theory	See Particle Induced Mixing Models.
Provided By	**[phase interaction]** > **Models** > *Optional Models*
Example Node Path	**Multiphase Interaction** > **[Phase Interaction]** > **Models** > **Particle Induced Mixing**
Requires	A Multiphase Segregated Flow simulation, with a minimum of two phases. One phase must be a continuous phase, and another phase must be a dispersed phase. An appropriate turbulence model activated for each Eulerian phase. A Continuous-Dispersed Phase Interaction with the *Continuous Phase* and *Dispersed Phase* matched with the appropriate Eulerian phases.

Compatible With	• S-gamma size distribution model • k-epsilon turbulence models • k-omega turbulence models • Reynolds Stress models Note that the Particle Induced Mixing model is incompatible with Heat Transfer and the following continuous phase models: • Laminar flow • Multi-component phases • LES • Passive scalars	
Properties	None.	
Activates	Physics Models	None.
	Model Controls (child nodes)	**Particle Induced Viscosity** See Particle Induced Viscosity Properties.
	Field Functions	*Particle Induced Viscosity of [phase interaction]* See Field Functions.

Particle Induced Viscosity Properties

Method

Specifies the particle-induced viscosity method to use. The options are:

- **Constant**
- **Field Function**
- **Sato Bubble Induced Viscosity**
  See Sato Bubble Induced Viscosity Properties.

Sato Bubble Induced Viscosity Properties

Calibration Constant

A prefactor to the Sato bubble-induced viscosity. This value is k in Eqn. (2707).

A+ Constant

A dimensionless constant in the van Driest damping factor that is used in the Sato bubble-induced viscosity calculation. This value is A^+ in Eqn. (2708).

Field Functions

Particle Induced Viscosity of [phase interaction]

The bubble-induced viscosity for the specific phase interaction. For the Sato model, this value is ν_{Sato} as given by Eqn. (2707).

Interphase Turbulence Transfer Model

The Interphase Turbulence Transfer model accounts for the contributions to the fluid and particle turbulent kinetic energies due to drag. The unclosed or modeled term (within the source terms) is the cross-correlation between fluid and particle fluctuating velocities evaluated along the particle trajectory.

Once activated, the Interphase Turbulence Transfer model produces additional sources for kinetic energy of turbulence, turbulent dissipation rate, specific dissipation rate, and Reynolds stresses.

This model is applicable for gas-particle flows carrying fine particles (that is, Geldart A type particles, having a size between 20 and 100 μm, and a density typically less than 1.4g/cm^3). A typical application for this model is a Fluidized Catalytic Cracking (FCC) system.

Interphase Turbulence Transfer Model Reference

Theory	See Turbulence Transfer between Phases.	
Provided By	**[phase interaction]** > **Models** > *Optional Models*	
Example Node Path	**Multiphase Interaction** > **[Phase Interaction]** > **Models** > **Interphase Turbulence Transfer**	
Requires	A Multiphase Segregated Flow simulation, with the Viscous Regime set to **Turbulent**. A minimum of two phases: one phase must be a gas or liquid continuous phase, and another phase must be a particle dispersed phase. For each Eulerian phase, the Viscous Regime set to **Turbulent**. Both the continuous phase and the dispersed phase require the same type of turbulence model activated (for example, both K-Epsilon, or both K-Omega, or both RSM). A Continuous-Dispersed Phase Interaction with the *Continuous Phase* and *Dispersed Phase* matched with the appropriate Eulerian phases.	
Compatible With	• k-epsilon turbulence models Note that the Interphase Turbulence Transfer model is incompatible with the following models: • Tchen and Gosman particle induced turbulence models • Sato Particle Induced Mixing model • Turbulence Response model	
Properties	See Interphase Turbulence Transfer Model Properties.	
Activates	Physics Models	None.

	Model Controls (child nodes)	**Velocity Cross-Correlation Coefficient** See Velocity Cross-Correlation Coefficient Properties.
	Field Functions	*Velocity Cross Correlation Coefficient of [phase interaction]* See Field Functions.

Interphase Turbulence Transfer Model Properties

Calibration Constant

The coefficient C_0 in Eqn. (2723) to Eqn. (2726) defining the source terms for the turbulent kinetic energy and turbulence dissipation rate equations. This parameter is not part of the model derivation, but defaults to 1 and you can adjust it to test the sensitivity of the solution to this term.

C3e

The coefficient $C_{\varepsilon 3}$ in Eqn. (2725) and Eqn. (2726) defining the corresponding source term for the turbulence dissipation rate equations. This parameter has a default value of 1.54, and you can adjust it independently of constants for other terms.

Velocity Cross-Correlation Coefficient Properties

Method

Specifies the method to use to set the correlation coefficient of the turbulent velocity fluctuations of the phases. This value is β_{cd} in Eqn. (2723) to Eqn. (2726).

The options are:

- **Constant**
  The default method, with a default value of 0.7.
- **Field Function**
- **Zaichik**
  See Zaichik Model Properties.

Zaichik Model Properties

See Zaichik Method.

Crossing Trajectories Coefficient

C_β in the Zaichik formulation ([519]) is a calibration coefficient for the crossing-trajectories effect. By default this constant is set to 0.45 (Deutsch and Simonin [419]).

Field Functions

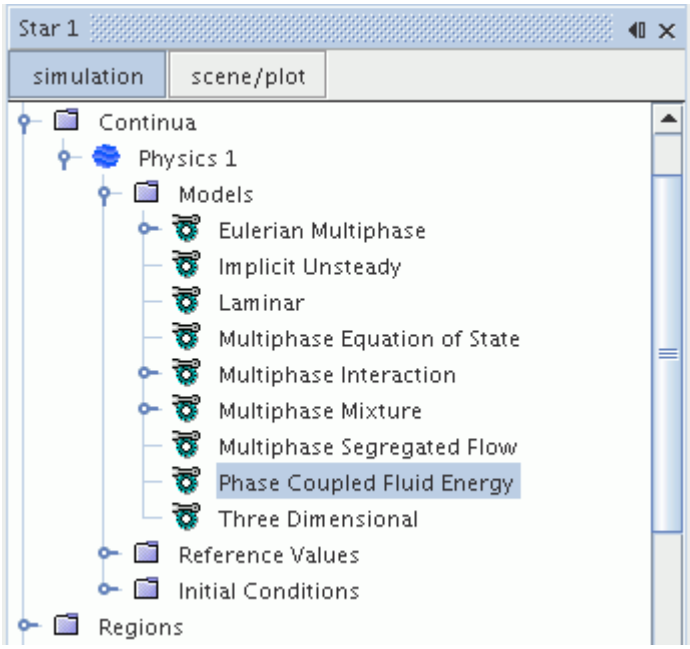
Velocity Cross Correlation Coefficient of [phase interaction]

β_{cd} in Eqn. (2723) to Eqn. (2726).

Modeling Energy in Multiphase Segregated Flow

Energy conservation within a multiphase flow can be modeled using the Phase Coupled Fluid Energy model.

This model solves the energy conservation equation for each Eulerian phase within the continuum.



When you select this model for the physics continuum, also choose the particular form of energy equation that is required in each Eulerian phase.

Solving for Enthalpy

The enthalpy field can be of interest in some applications, especially those involving phase change. In such applications, it is recommended that you also set the Heat of Formation and Standard State Temperature properties for each phase material. These two parameters, along with the specific heat of the phase, allow the enthalpy for a particular phase to be fully defined. Consistent definitions of phase enthalpies (that is with phase reference enthalpies defined against a reference temperature that is common for all phases) also correctly defines the magnitude and sign of enthalpy of phase.

These reference condition inputs have been named Heat of Formation and Standard State Temperature as commonly used in reacting flows. However, any other consistent set of reference data can be used instead.

For phase change calculations, a typical usage is to specify phase enthalpies at the saturation temperature corresponding to the system pressure. Example values for two-phase boiling water systems operating at 101,325 Pa are:

- Enthalpy reference temperature: 373.12 K
- Enthalpy reference value (liquid): 417436 J/kg
- Enthalpy reference value (vapor): 2674950 J/kg

The enthalpy results that are computed for this system can then be compared directly with steam table values.

Contents:

Selecting the Phase Coupled Fluid Energy Model
Multiphase User Wall Heat Flux Coefficient Specification

See Also:

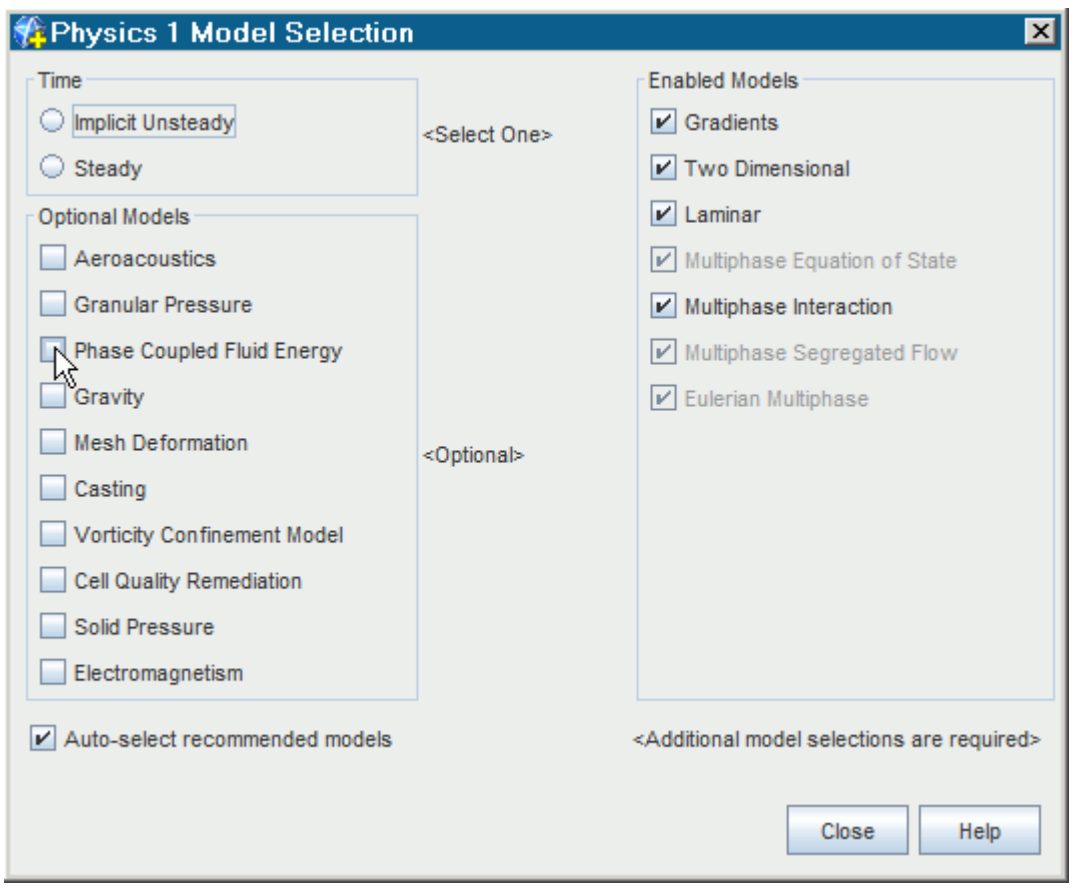
Using the Standard State Temperature

Selecting the Phase Coupled Fluid Energy Model

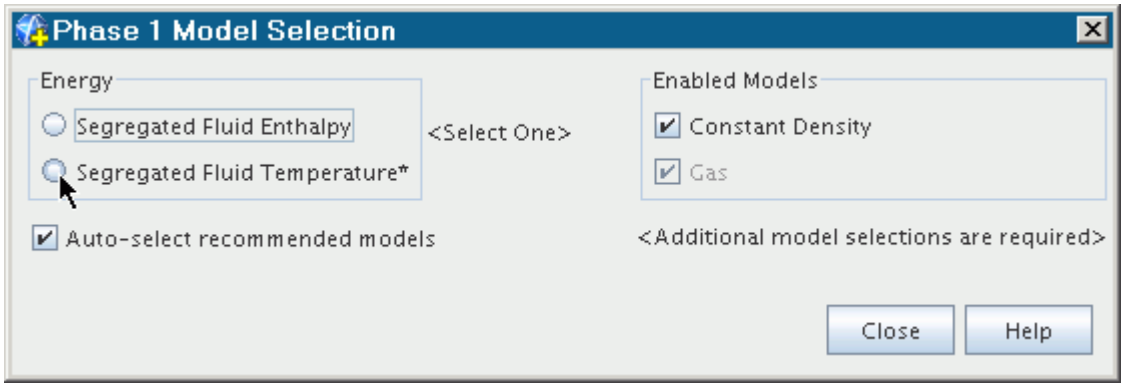
The **Phase Coupled Fluid Energy** option becomes available within the *Optional Models* group box of the *Physics Model Selection* dialog after you have chosen the **Multiphase Segregated Flow** model.

To select the **Phase Coupled Fluid Energy** model:

1. Click this option as shown below.



An additional *Energy* option appears within the *Phase Model Selection* dialog for each Eulerian phase in the continuum. An example of this dialog is shown below.



2. Choose between **Segregated Fluid Enthalpy** and **Segregated Fluid Temperature**.

Multiphase User Wall Heat Flux Coefficient Specification

The User Wall Heat Flux Coefficient Specification node allows you to control the heat flux relationship at walls and interfaces.

The User Wall Heat Flux Coefficient Specification in a multiphase simulation has the following differences from the single-phase usage:

- The **Specified Net Coefficients** method is available in Multiphase Segregated Flow simulations only.

  This method sets the net boundary heat flux coefficients equal to the user-specified heat flux coefficients, so that the net coefficients = user-defined coefficients.
- You can implement your own wall contact models for the energy equation specifying user wall heat flux coefficients from internal wall heat flux coefficients.

  The internal wall heat flux coefficients are defined per phase-wall contact area and the Net wall heat flux coefficients are defined against full geometric wall face area. For Multiphase Segregated Flow simulations, this allows you to implement an alternative to the default model, using the volume fraction for the phase-wall contact area. You can implement this independently for the energy equation.

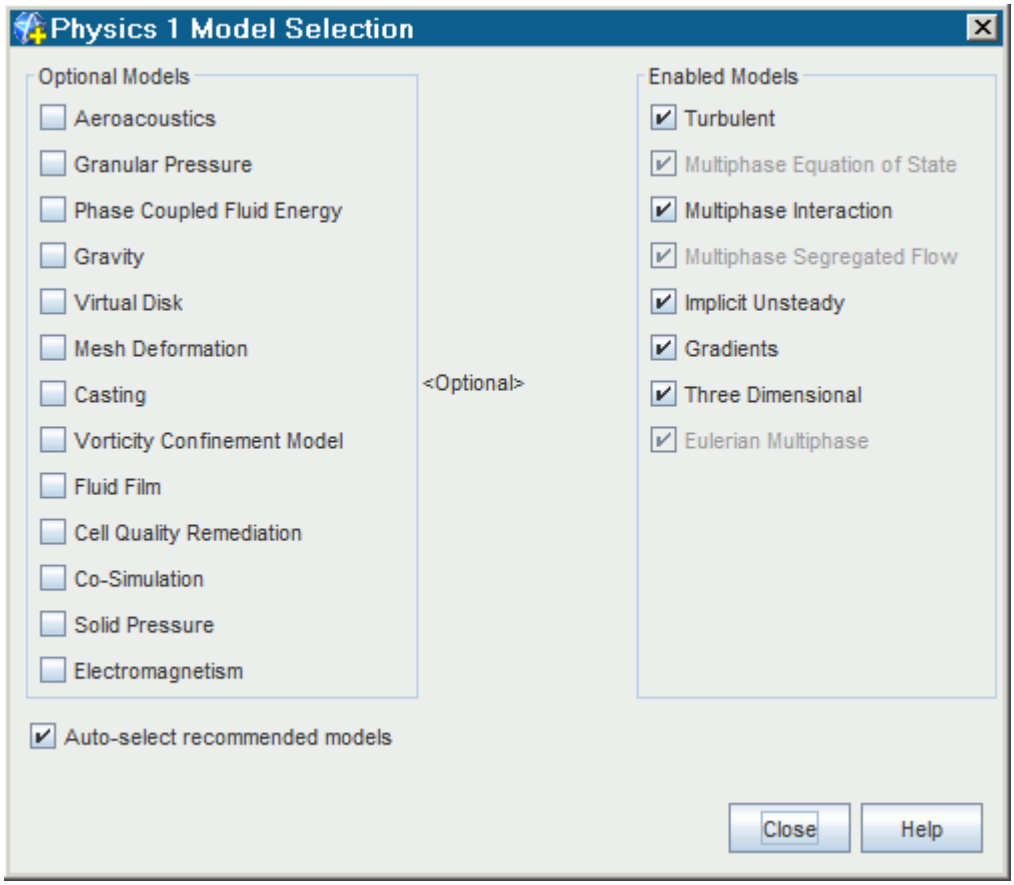
  For more information, see Phase Heat Flux.

Modeling Combustion in Multiphase Segregated Flow

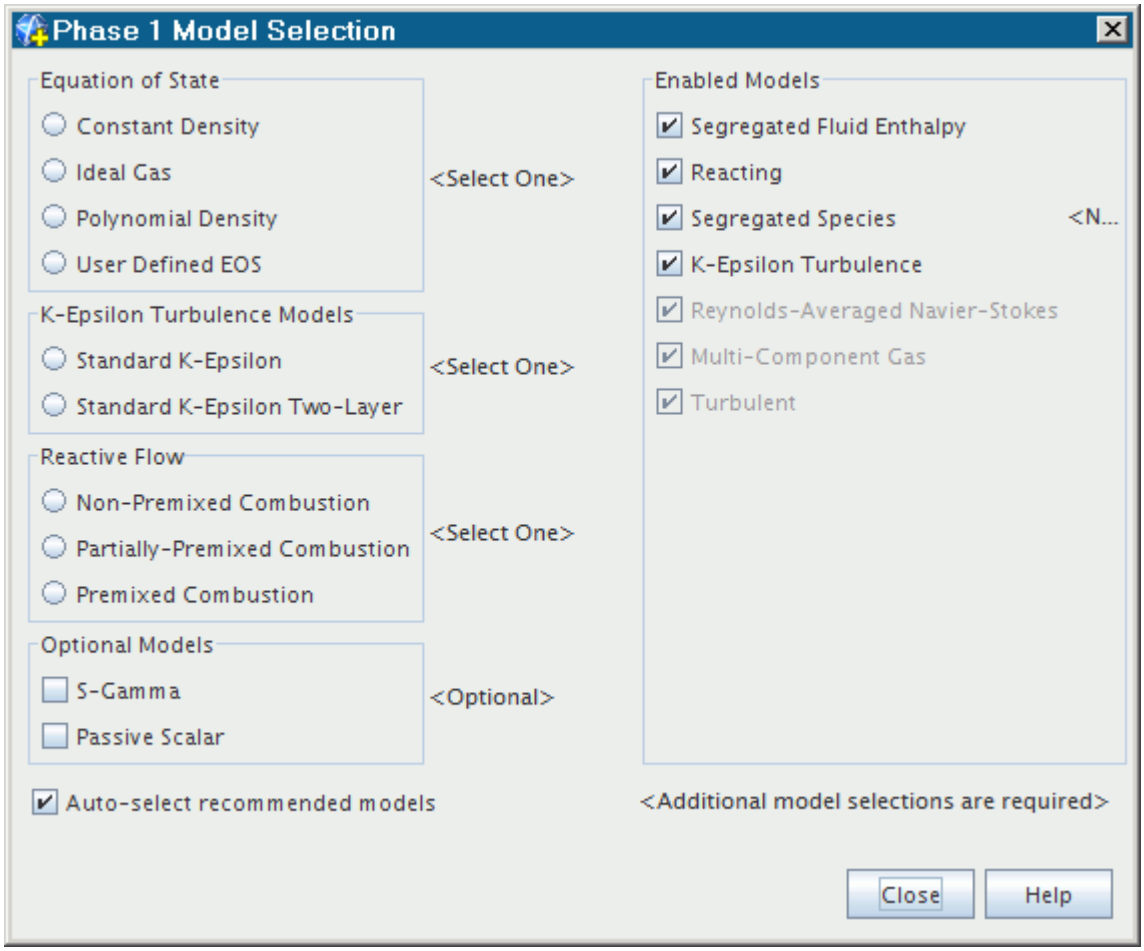
Combustion in a multiphase flow has applications that include fluid catalytic cracking, fluidized beds, and riser flow.

To use the STAR-CCM+ facilities for solving combustion problems within a multiphase flow:

1. Before you select phase models, activate the Phase Coupled Fluid Energy model in the single-phase continuum.



2. For the phase, in the *Phase 1 Model Selection* dialog, select either a Multi-Component Gas or a Multi-Component Liquid material model from the *Material* group.
3. Select **Segregated Fluid Enthalpy** from the *Energy* group. This model is required for each phase in which you want to use combustion.
4. Specify that this phase requires chemical reaction simulation by selecting **Reacting** in the *Reaction Regime* group.
5. Specify whether your combustion is premixed, non-premixed or partially premixed by selecting the corresponding radio button in the *Reactive Flow* group.



The availability of specific combustion modeling options depends on this choice and on the material model that you chose earlier.

Accessing Combustion Models

After selecting a combustion flame type, you are presented with a choice of combustion models valid for use with the flame type. If the combustion model you want is not available for the flame type, deactivate the flame type from the *Enabled Models* section and choose the correct alternative. You can switch between combustion models that are presented for a particular flame type without this step. For example, when you choose **Complex Chemistry**, you can switch among the three flame types even if you do not deactivate the combustion model. The relationships among material models, combustion flame types, and specific combustion modeling options are outlined in the Combustion section of the User Guide.

Most of the other features that are available with gas phase reactions, such as the species source and specific heat with NASA polynomials, are also available with liquid phase reactions.

Modeling the Boiling Process

The Wall Boiling and Bulk Boiling models within the Eulerian Multiphase framework of STAR-CCM+ are aimed at forced-convection, subcooled, boiling applications.

In subcooled boiling, local boiling occurs on a heated surface despite the mean temperature of the liquid being less than the saturation temperature. The degree of liquid subcooling is expressed either as a temperature difference or an enthalpy difference below the saturation value.

Provided the flow regime near the wall remains close to bubbly flow, a standard subcooled boiling model can be pushed towards zero subcooling conditions (that is, saturated boiling). The model can also be pushed towards higher heat fluxes approaching Departure from Nucleate Boiling (DNB) conditions.

A successful wall boiling model predicts the amount of wall superheat, which is the amount the wall temperature exceeds the saturation temperature, as well as the axial and radial distribution of vapor. The Eulerian Multiphase framework is appropriate for modeling this kind of wall boiling process for several reasons:

- By definition, the phases in subcooled boiling are not in thermal equilibrium and each phase requires its own energy equation.
- The phases in subcooled boiling require their own equation of motion, as the bubble moves away from the boiling surface while cool liquid moves to take its place.
- The condensation rate of vapor in the bulk subcooled liquid is a part of the wall boiling process and decides:
  - when net vapor generation first begins on the wall, and
  - how the vapor is distributed downstream of this point.

  Both of these non-equilibrium flow features can be different in reality from predictions using equilibrium models.

The Bulk Boiling model in STAR-CCM+ is an essential part of subcooled boiling. Boiling and condensation is modeled in the interior of the flow, provided the rate of interphase mass transfer only depends on the following:

- The heat transfer rate from the liquid and vapor to a saturated interface between these two phases.
- The interfacial area density that is provided by a well-defined characterization of bubbles by their size and number density.

This means that the bulk boiling model is suitable for modeling vapor bubbles whose initial size and concentration:

- are already known as an inlet condition (or initial condition in the case of transient simulations)
- are estimated by a size distribution model (such as S-gamma or AMUSIG)
- are generated by the Wall Boiling model.

This thermally-driven model is not suitable for describing boiling without bubbles (homogeneous nucleation), evaporation (concentration-driven multi-component mass transfer), nor cavitation (inertially-limited mass transfer). The thermally-driven model is also unsuitable when the resulting interphase mass transfer rate occurs on time scales that are much shorter than that of the flow. This scenario is unlikely to be solvable with a segregated solver algorithm.

Contents:

Modeling Boiling
What Is Bulk Boiling?
What Is the Boiling/Condensation Model?
What Is the Boiling Mass Transfer Rate?

What is the Multi-component Boiling Mass Transfer Rate Model?
Setting Up a Wall Boiling Simulation
Wall Boiling Model Reference

Modeling Boiling

The boiling models can be used for forced-flow, subcooled boiling applications.

The Setting Up a Wall Boiling Simulation section provides guidelines on the wall boiling model for both new and expert users.

To set up a boiling simulation:

1. Set up a Multiphase Segregated Flow simulation with the Phase Coupled Fluid Energy physics model activated.
2. Create two Eulerian phases: a single component (or multi-component) liquid phase and a gas phase defined as the corresponding vapor phase.
3. In the *Phase Model Selection* dialog for each phase, select the following:

   Material: **Gas**, **Liquid**, **Multi-Component Gas**, or **Multi-Component Liquid**.

   Energy: **Segregated Fluid Temperature** or **Segregated Fluid Enthalpy**

   Reaction Regime: **Non-reacting** (applies to multi-component phases only)
4. Create a phase interaction and activate the **Continuous-Dispersed Phase Interaction** model or the **Multiple Flow Regime Phase Interaction** model, whichever is appropriate to your simulation.
5. Under the **[Phase Interaction]** > **Models** node, do one of the following:
   - Select the **Continuous-Dispersed Phase Interaction** node and set the *Continuous Phase* as the liquid phase and the *Dispersed Phase* as the vapor phase.
   - Select the **Multiple Flow Regime Phase Interaction** node and set the *Primary Phase* as the liquid phase and the *Secondary Phase* as the vapor phase.
6. Reopen the *Phase Interaction Model Selection* dialog and activate the Interphase Mass Transfer model.
7. In the *Interphase Mass Transfer Rate* group box, do one of the following:
   - Select **Boiling Mass Transfer Rate**.
   - Select **Boiling/Condensation**.
8. Select the **[Phase Interaction]** > **Models** node and do one of the following:
   - Select the **Boiling Mass Transfer Rate** node and set the following:
     - **Continuous Phase Nusselt Number**
     - **Disperse Phase Nusselt Number**

       See Setting the Nusselt Number.
   - Select the **Boiling/Condensation** node and set the following:
     - **First Regime Nusselt Number**
     - **Second Regime Nusselt Number**
     - **Interface Nusselt Number**

       See What Is the Boiling/Condensation Model?

9. Select the **Interaction Length Scale** node and do one of the following:
   - For a **Continuous-Dispersed Phase Interaction**, select the appropriate method for the **Interaction Length Scale**.
   - For a **Multiple Flow Regime Phase Interaction**, select the appropriate methods for the **First Regime Interaction Length Scale** and the **Second Regime Interaction Length Scale**.

   See Modeling Interaction Length Scale.

To model wall boiling:

10. Reopen the *Phase Interaction Model Selection* dialog and, in the *Optional Models* group box, select **Wall Boiling**.

    See Wall Boiling Model Reference.

    The **Wall Bubble Nucleation** and **Wall Transient Conduction** models are selected automatically.
11. Select the **[Phase Interaction]** > **Models** > **Wall Boiling** node, and set the appropriate properties.
    - **Wall Dryout Area Fraction**

      Specifies the amount of heat flux applied at the wall that goes towards vapor convection, as opposed to liquid convection and evaporation.

      See Wall Dryout Area Fraction Properties.
12. Select the **Wall Bubble Nucleation** node, and set the appropriate properties.
    - **Nucleation Site Number Density**

      The nucleation site number density determines the number of locations on the heated surface where bubbles form, per unit area.

      See Nucleation Site Number Density Properties.
    - **Bubble Departure Diameter**

      The bubble departure diameter determines the diameter of the bubble at the instant it leaves the nucleation site.

      See Bubble Departure Diameter Properties.
    - **Bubble Departure Frequency**

      The bubble departure frequency determines how many bubbles leave a nucleation site per second.

      See Bubble Departure Frequency Properties.
13. Select the **Wall Transient Conduction** node, and set the appropriate properties.

    This model corrects the Bubble Induced Quenching Heat Flux so that it uses the temperature of the liquid brought to the wall by the action of the departing bubble.

    See Wall Transient Conduction Properties.
    - **Bubble Influence Wall Area Fraction**

      The bubble influence wall area fraction estimates the fraction of the wall area that is affected by the sweep of liquid inflow beneath a departing bubble.

      See Bubble Influence Wall Area Fraction Properties.
    - **Bubble Induced Quenching Heat Transfer Coefficient**

The quenching heat transfer coefficient is used to calculate the quenching heat flux.

See Bubble Induced Quenching Heat Transfer Coefficient Properties.

14. Specify the boundaries on which wall boiling is allowed. For each wall boundary, do the following:
    a) Select the **Regions** > **[Region Name]** > **Boundaries** > **[Wall Boundary Name]** > **Physics Conditions** > **Wall Interphase Mass Transfer Option** node and set *Method* to **Active**.

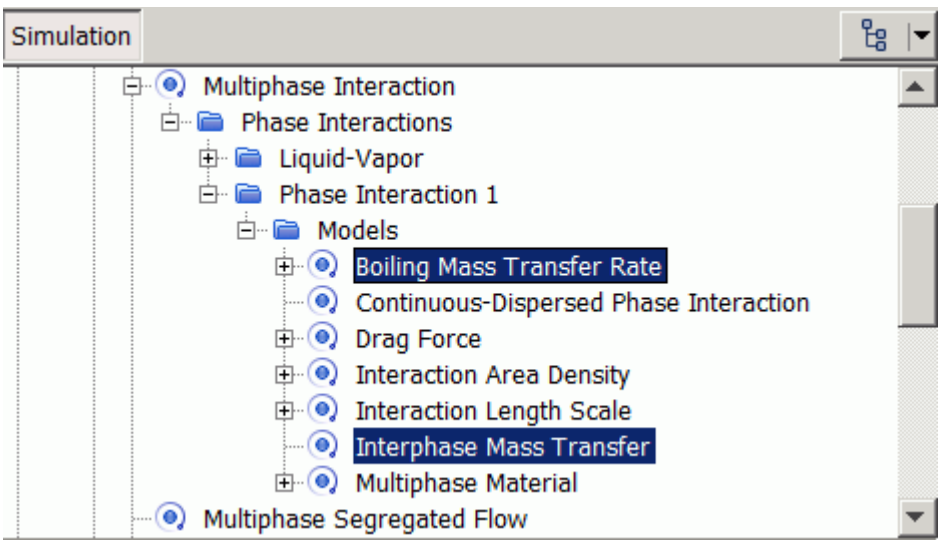
       To prevent wall boiling from occurring on this boundary, set *Method* to **None**. For example, use this option when a boundary is adiabatic.
    b) Select the **[Wall Boundary Name]** > **Phase Conditions** node and for each phase in the phase interaction, set the User Wall Heat Flux Coefficient Specification.

What Is Bulk Boiling?

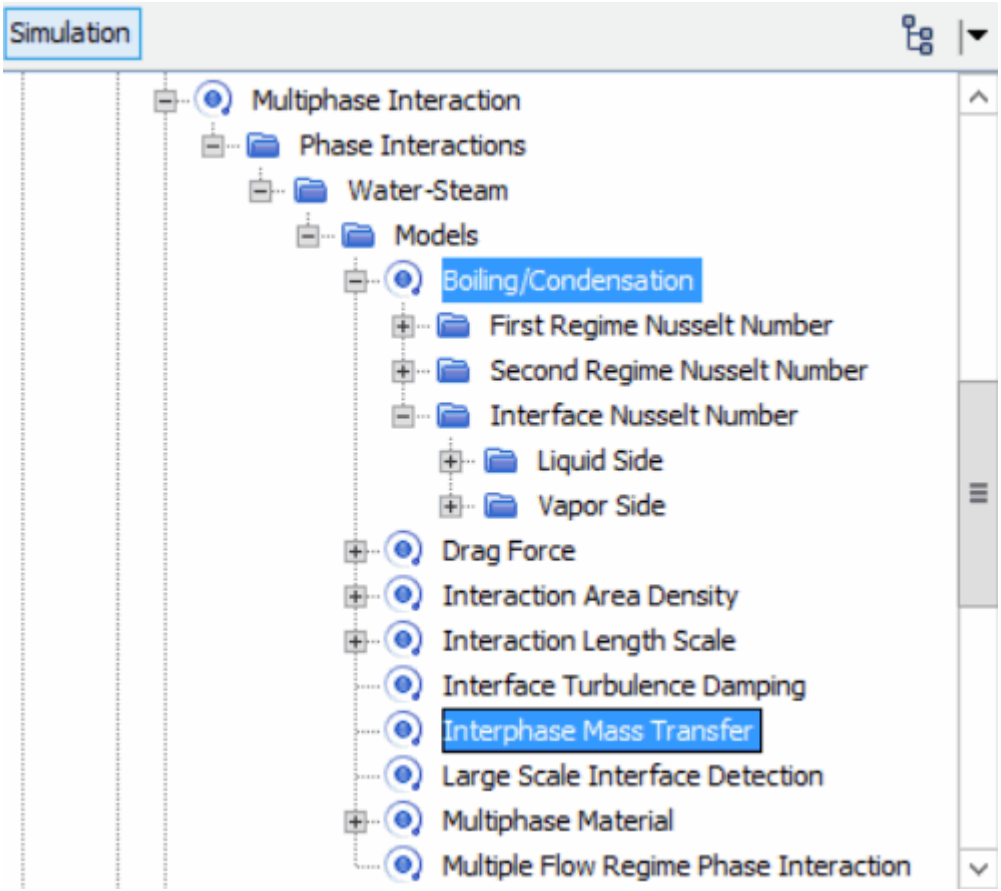
The bulk boiling model assumes that the phase temperatures are not in equilibrium, except at the interface where they are both equal to the saturation temperature.

Bulk boiling/condensation is activated by selecting **Interphase Mass Transfer** phase interaction models.

To model bulk boiling/condensation with a Continuous-Dispersed phase interaction, select the **Boiling Mass Transfer Rate** model.



To model bulk boiling/condensation with a Multiple Flow Regime phase interaction, select the **Boiling/ Condensation** model.



STAR-CCM+ can solve for bulk boiling or condensation at a rate comparable to flow time-scales. To solve these problems, use the interphase mass transfer models. The following models are provided:

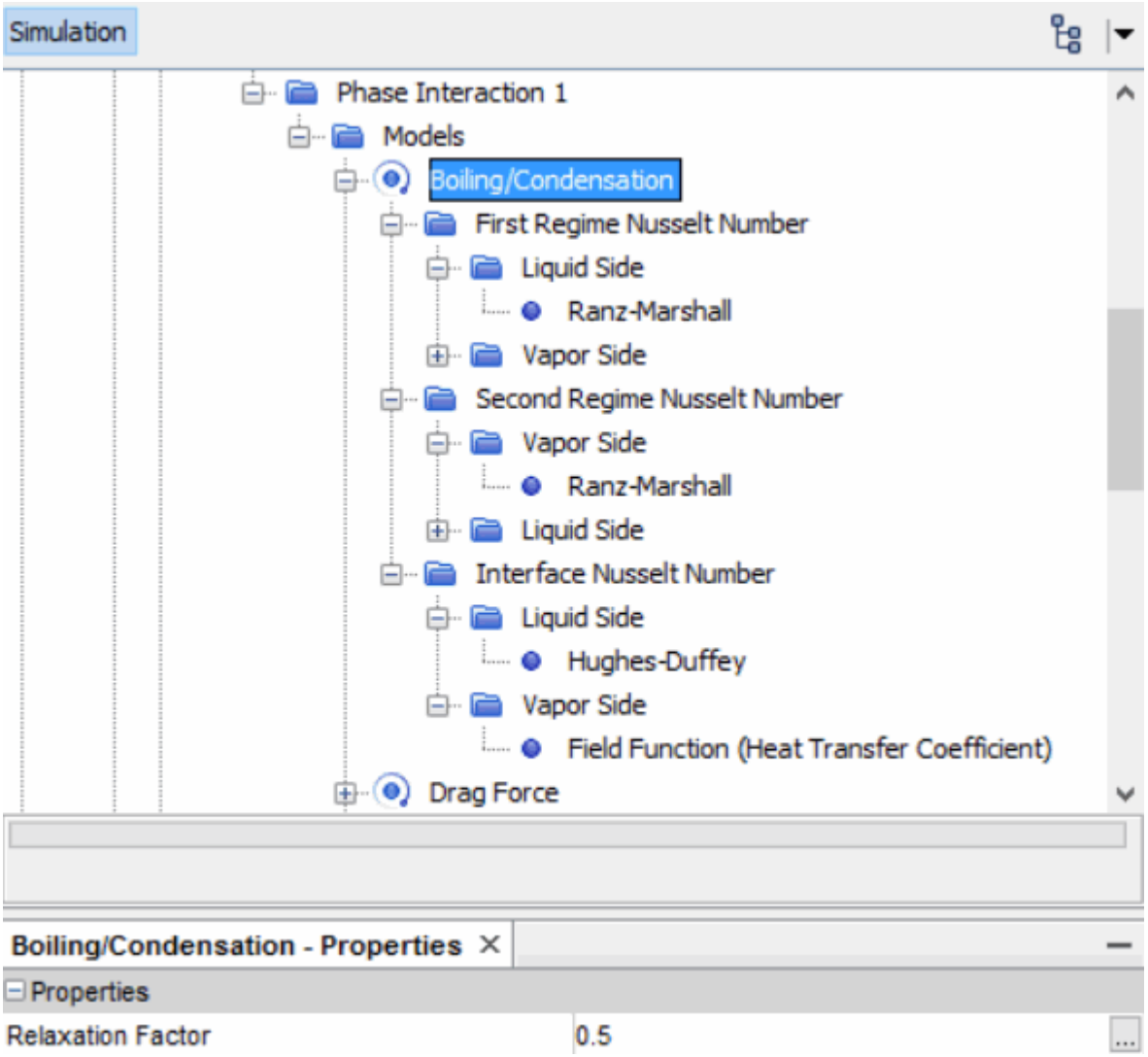
- Continuous-Dispersed Phase Interaction
  - Boiling Mass Transfer Rate
  - Multi-component Boiling Mass Transfer Rate
  - User-Defined Interphase Mass Flux
- Multiple Flow Regime Phase Interaction
  - Boiling/Condensation

What Is the Boiling/Condensation Model?

The Boiling/Condensation model is used to model the rate of bulk boiling or condensation between phases in a Multiple Flow Regime phase interaction.

This model is only available when the Phase Coupled Fluid Energy model is active in the physics continuum. The Multiple Flow Regime Phase Interaction must specify the primary phase as a liquid and the secondary phase as the vapor of that liquid.

The properties of three child nodes, the First Regime Nusselt Number, the Second Regime Nusselt Number, and the Interface Nusselt Number are used to define the parameters that the model equation requires.



Boiling/Condensation Properties

Relaxation Factor	Adjusts the model sensitivity. Reduce it to improve convergence.

Properties of the First Regime Nusselt Number

Controls the heat transfer rate to the boiling interface from the liquid side and the vapor side in the first regime (see Eqn. (2375) and Eqn. (2376)).

The Liquid Side and Vapor Side nodes have the following properties:

Method	**Constant**	Adds a **Constant** child node on which to set a constant value.
	Field Function	Adds a **Field Function** child node on which to select a field function.
	Field Function (Heat Transfer Coefficient)	Adds a **Field Function (Heat Transfer Coefficient)** child node.

		Specifies the interphase heat transfer coefficient between liquid and vapor (W/m^2-K).
	Ranz Marshall	Available on the **Liquid Side** node only. This option is available only for external heat transfer, that is, the liquid side in the first regime. Adds the **Ranz Marshall** child node that represents use of the Ranz-Marshall correlation that is given in Eqn. (2354).

Properties of the Second Regime Nusselt Number

This property controls the heat transfer rate to the boiling interface from the liquid side and the vapor side in the second regime (see Eqn. (2375) and Eqn. (2376)).

The Liquid Side and Vapor Side nodes have the following properties:

Method	**Constant**	Set a constant value.
	Field Function	Select a field function.
	Field Function (Heat Transfer Coefficient)	Adds a **Field Function (Heat Transfer Coefficient)** child node. Specifies the interphase heat transfer coefficient between liquid and vapor (W/m^2-K).
	Ranz Marshall	Available on the **Vapor Side** node only. This option is available only for external heat transfer, that is, the vapor side in the second regime. Adds the **Ranz Marshall** child node that represents use of the Ranz-Marshall correlation that is given in Eqn. (2354).

Properties of the Interface Nusselt Number

The Liquid Side and Vapor Side nodes have the following properties:

Method	**Constant**	Set a constant value.
	Field Function	Select a field function.
	Field Function (Heat Transfer Coefficient)	Adds a **Field Function (Heat Transfer Coefficient)** child node.

		Specifies the interphase heat transfer coefficient between liquid and vapor (W/m^2-K).
	Hughes-Duffey	The liquid side heat transfer coefficient in the interface regime is calculated using the surface renewal theory introduced by Hughes and Duffey ([439]), Eqn. (2381). This method is available only when the primary phase has a turbulence model activated.

Hughes-Duffey Properties

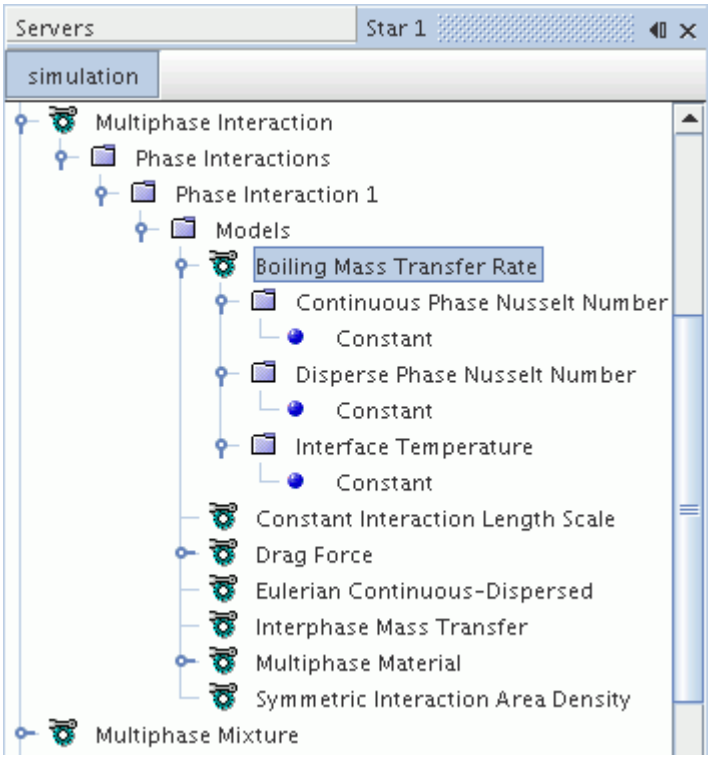
Calibration Factor	A calibration factor to adjust the Hughes-Duffey heat transfer coefficient. Increase this factor to increase the heat transfer coefficient value.

What Is the Boiling Mass Transfer Rate?

The Boiling Mass Transfer Rate model is used to model the rate of bulk boiling or condensation between phases for a Continuous-Dispersed phase interaction.

This model is only available when the Phase Coupled Fluid Energy model is active in the physics continuum. The properties of three child nodes, the Continuous Phase Nusselt Number, the Disperse Phase Nusselt Number and the Interface Temperature, are used to define the parameters that are required by the model equation.

For details of the Continuous Phase Nusselt Number and the Disperse Phase Nusselt Number, see Setting the Nusselt Number.



Boiling Mass Transfer Rate Properties

Relaxation Factor	Adjusts the model sensitivity. Reduce it to improve convergence.

Properties of the Interface Temperature

The interface temperature is often defined as a constant saturation temperature for a particular system pressure. In STAR-CCM+, a field function can optionally be used for the interface temperature, so that depth through the interface layer can be accounted for, or overpressure due to surface tension in micron-sized bubbles.

Method	Select the method to use in specifying the interface temperature T_{ij} in Eqn. (2375) and Eqn. (2376).	
	Constant	Adds a node on which to set the constant value.
	Field Function	Adds a node on which to select a field function.

What is the Multicomponent Boiling Mass Transfer Rate Model?

The Multicomponent Boiling Mass Transfer Rate model is used to model the rate of bulk boiling or condensation between multi-component phases. The model calculates the mass, energy, and momentum transfer between a continuous and a dispersed multi-component phase.

The two phases can be either a gas and a liquid, or two liquids that are divided by an interface.

STAR-CCM+ implements the full multi-component boiling model when the Maxwell-Stefan diffusion model is active. However, when the Maxwell-Stefan diffusion model is not active, the dispersion is assumed to be dilute and a reduced multi-component boiling model is used.

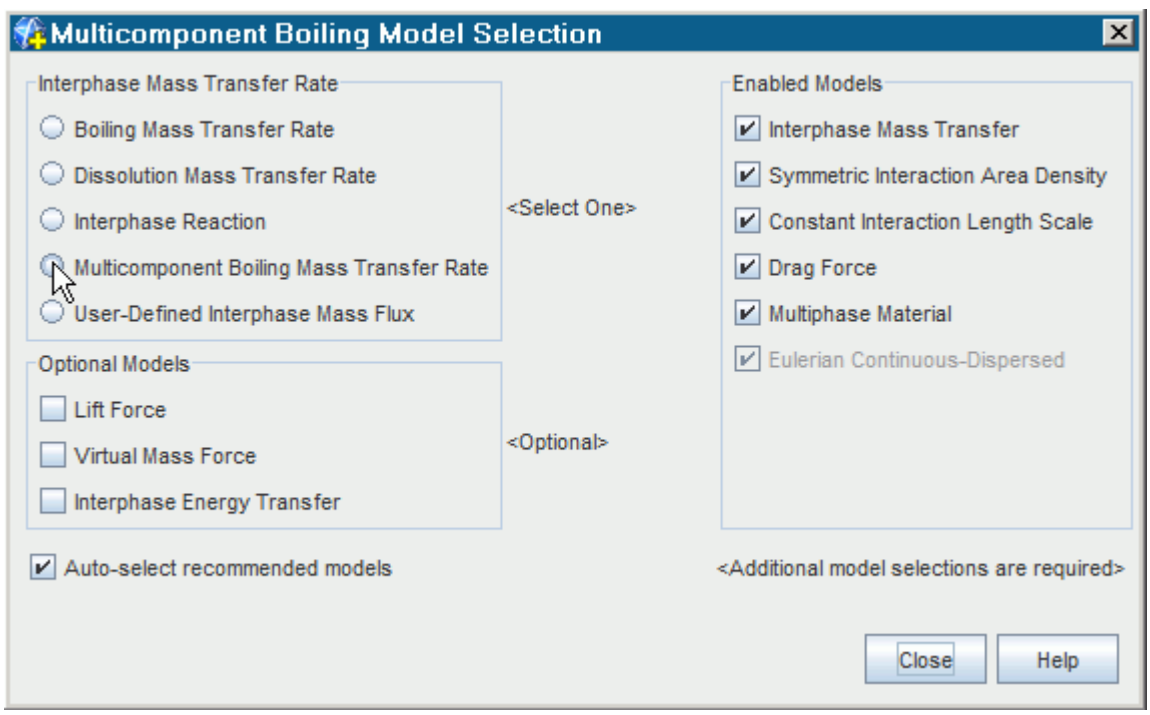
For more information on the reduced multi-component boiling model, see Reduced Model for Dilute Dispersions in the Multi-component Bulk Boiling formulation.

This model is only available when the Phase Coupled Fluid Energy model is active in the physics continuum.

To select the Multicomponent Boiling Mass Transfer Rate model:

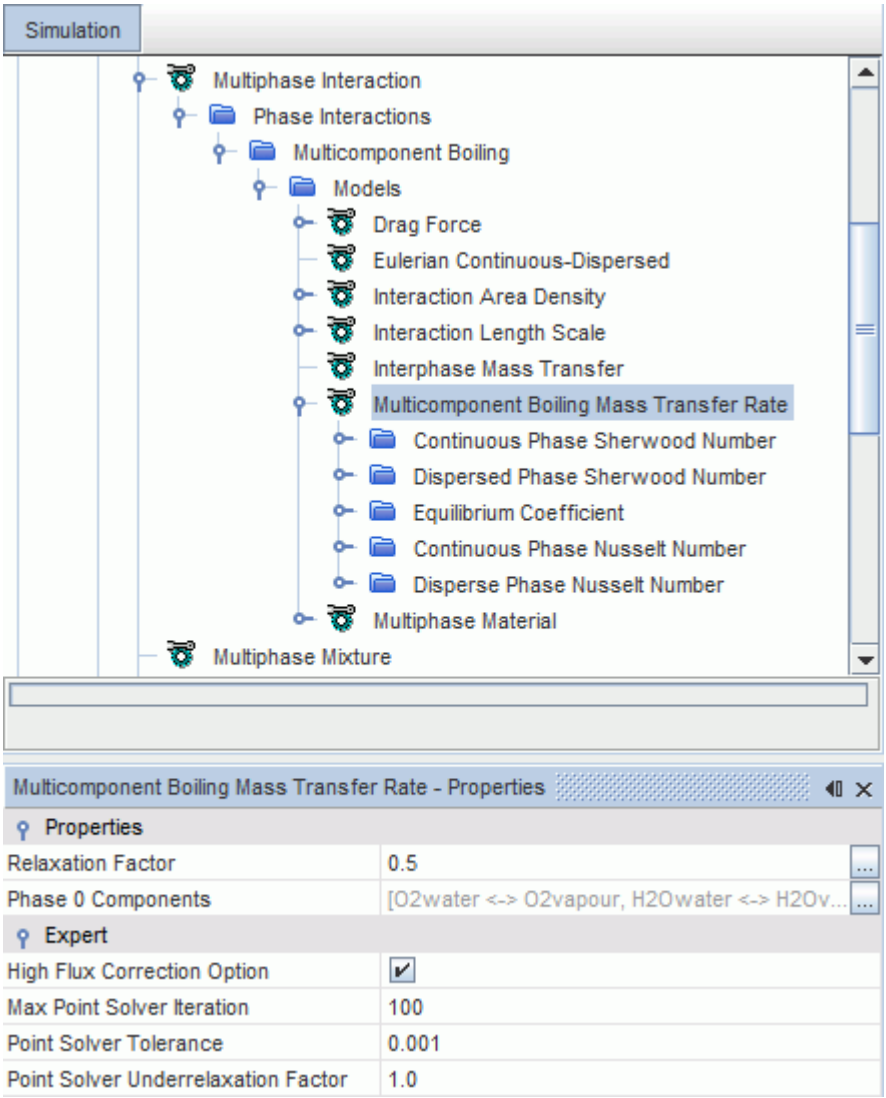
1. Define both of the phases in the interaction to have more than one component—usually one multi-component liquid and one multi-component gas.
2. Specify the continuous and dispersed phases.
3. Select the Interphase Mass Transfer model.

   The *Interphase Mass Transfer Rate* options become available.
4. Select the **Multicomponent Boiling Mass Transfer Rate** model from the *Interphase Mass Transfer Rate* group box.



5. Click **Close**.

The **Multicomponent Boiling Mass Transfer Rate** node is shown below:



Multicomponent Boiling Mass Transfer Rate Properties

Relaxation Factor	Adjusts the model sensitivity. Reduce it to improve convergence. The default value is 0.5, but lower values may be required for high mass transfer rates.
Phase 0 Components	Lets you specify the relationship between the components that are transferred between phases.

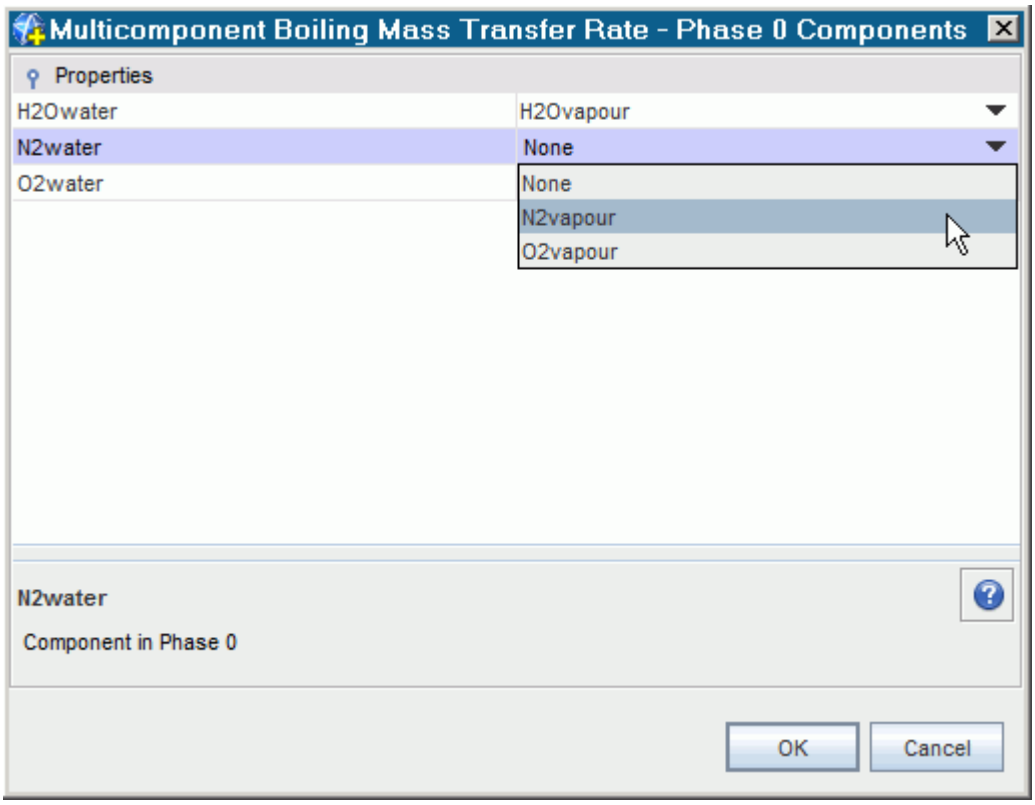
Multicomponent Boiling Mass Transfer Rate Expert Properties

High Flux Correction Option	When activated, the high mass transfer rate coefficients are calculated. You can deactivate this option to skip the high mass transfer rate coefficients calculation, in which case the coefficients are set to unity. This can save computing time for low mass transfer rates.

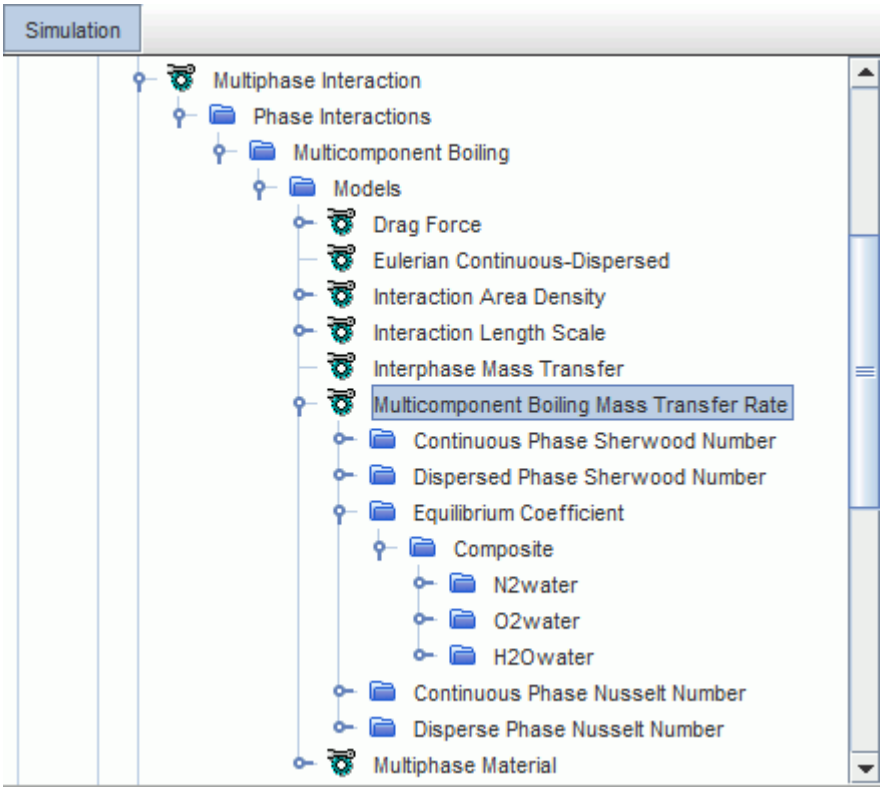
Max Point Solver Iteration	Specifies the maximum number of iterations performed by the Point Solver. The default value is 100.
Point Solver Tolerance	Specifies the maximum value of the Point Solver residual. The default value is 0.001.
Point Solver Underrelaxation Factor	Adjusts the sensitivity of the Point Solver. Reducing this value, or the *Point Solver Tolerance* value (above), can increase the computing time significantly.

Specifying the Relationship Between the Phase Components

The *Multicomponent Boiling Mass Transfer Rate - Phase 0 Components* dialog lets you specify the components to transfer. You match the components of the continuous phase (in the left-hand column) with the components of the dispersed phase in the right-hand column:



For each transfer activated, the continuous phase component name appears under the **Equilibrium Coefficient** > **Composite** node:



Multicomponent Boiling Mass Transfer Rate Model Nodes

The properties of child nodes are used to define the parameters that are required by the model equation.

- **Continuous Phase Nusselt Number** and **Dispersed Phase Nusselt Number**

  See Setting the Nusselt Number.
- **Continuous Phase Sherwood Number** and **Dispersed Phase Sherwood Number**

  See Setting the Sherwood Number.
- **Equilibrium Coefficient**

  See What is the Equilibrium Coefficient

Setting Up a Wall Boiling Simulation

This section provides recommendations and considerations for setting up a wall boiling simulation for both new and expert users.

Basic Setup

To set up a basic simulation:

- First, verify that your application is suitable for modeling as a subcooled boiling process. Base your model on real experimental or industrial conditions, or at least on an energy balance confirming that subcooled boiling is expected.
- Then, start from the default model selections and default model calibrations.

Condensation is a part of subcooled boiling, so you have to activate the Boiling Mass Transfer Rate model before activating the Wall Boiling model.

Selection of the Wall Boiling model requires that you set valid and accurate data for the following simulation properties:

- For **Reference Values** > **Gravity**, make sure that acceleration due to gravity is acting in the correct direction.
- For **Reference Values** > **Reference Pressure**, set the value to the expected outlet pressure, if setting zero relative pressure conditions at the outlet. Otherwise, set the value to the design operating pressure.
- For **Phase Interaction** > **Multiphase Material** > **Surface Tension**, set the surface tension coefficient, at saturation temperature, for the liquid-vapor interface.
- For **Eulerian Phases** > **Gas** > **Molecular Weight**, set an appropriate material property for the vapor phase.

Advanced Setup

The wall boiling model is intended to cover forced-flow, subcooled boiling up to pressures of around 155 bar, with as little calibration of model constants as possible. Therefore, it is built up mechanistically from many simpler submodels:

- Wall Dryout Area Fraction
- Nucleation Site Number Density
- Bubble Departure Diameter
- Bubble Departure Frequency
- Wall Transient Conduction
- Bubble Influence Wall Area Fraction
- Bubble Induced Quenching Heat Transfer Coefficient

The output from each of these submodels can be inspected graphically at run time.

If suitable experimental data is available, these submodels can also be calibrated or replaced with user-defined relationships.

Working Fluid

The wall boiling submodels are implemented to use all of the phase and interface properties you specify (such as density, saturation enthalpy, surface tension).

Note: Some of the original coefficients are based upon water as the working fluid. Therefore, some adjustments are required when using a working fluid other than water.

The wall contact angle for the Hibiki Ishii Nucleation Site Number Density and Kocamustafaogullari Bubble Departure Diameter models is specific to the combination of working fluid and boiling surface.

Note: The wall contact angle is a nominal value that is based on room temperature, rather than a value measured under boiling conditions.

Adjust the following options, from older correlations for use with other working fluids:

- Kurul Podowski Interaction Length Scale

- Lemmert Chawla Nucleation Site Number Density
- Tolubinsky Kostanchuk Bubble Departure Diameter

Convergence

High heat flux wall boiling cases can sometimes fail to converge for two reasons:

- Interphase mass transfer rates due to bulk condensation are determined from instantaneous values of interfacial area and interfacial energy balance. However, in a segregated solver algorithm, these mass transfer rates are not directly coupled to the hydrodynamic equations, so they are only limited by mass balances after full convergence.
- The most sensitive part of the wall boiling model is the nucleation site number density. The number of active sites depends on wall superheat, perhaps to the power of 1.8–5 or more. In a segregated solution algorithm, the resulting evaporation rate can exceed the liquid supply to the wall until full convergence is achieved.

Rather than switching on all of the options immediately in a new computation, and then using a high degree of under-relaxation, a more natural strategy is given below:

- Obtain a converged flow and heat transfer solution at inlet temperature, but without any boiling.
- Restart the simulation with bulk condensation activated and obtain a converged flow.
- Activate wall boiling and gradually increase the temperature from just below the value that induces wall boiling, to the actual (system) temperature over a number of iterations, time-steps, or restarts.

If you experience difficulty in achieving convergence, adjust the relaxation factor that is provided for the Boiling Mass Transfer Rate (the terms that cover bulk condensation). Typical values are 0.5, but this value can be reduced to make the residuals decrease monotonically without oscillation.

Another relaxation factor is provided for the Wall Boiling model. It is applied to the phase energy equations at the wall and is intended to account for the sensitivity of evaporation rate to wall temperature. The recommended value is 0.5 when the default Lemmert Chawla model is selected for Nucleation Site Number Density. Reduce the relaxation factor to about 0.1 when the Hibiki Ishii model is used.

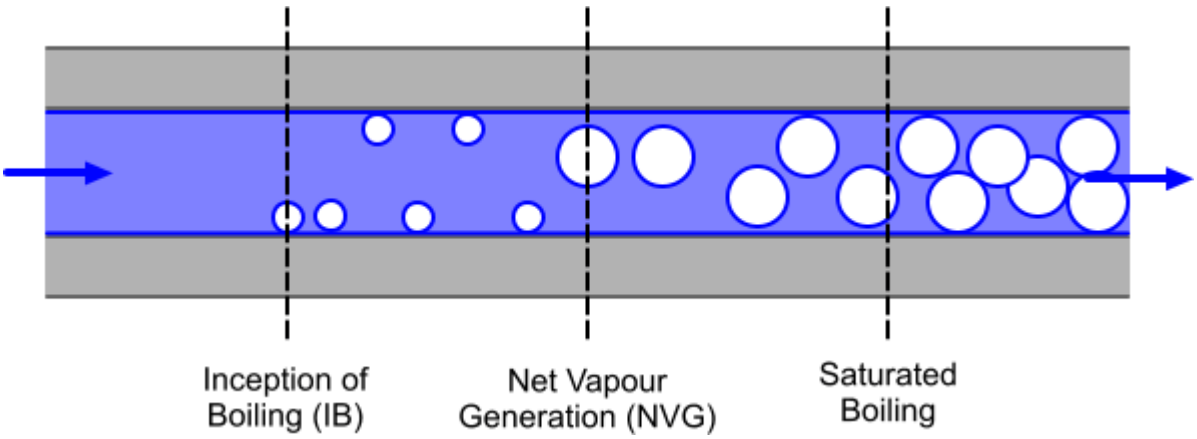
Sometimes, inappropriate turbulence profiles at inlets can cause unnecessary difficulties. The symptoms are that the turbulent viscosity levels collapse on entry with possible consequences for both nearby wall heat transfer coefficients and for global residual behavior. For long pipe and channel geometries, you can use the same model, but with boiling deactivated, to compute and save fully developed outlet velocity and turbulent profiles in a file. These boundary conditions can then be used as inlet conditions for a boiling simulation.

Wall Boiling Model Reference

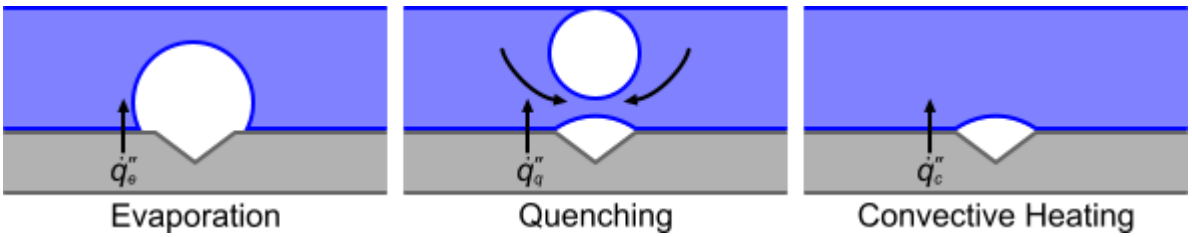
In a heated channel, where the flow at the entrance is subcooled, boiling begins when the wall temperature exceeds the saturation temperature of the liquid.

Small bubbles start to appear on the wall at the point of Inception of Boiling (IB). In this example, the degree of subcooling downstream of IB becomes smaller as the liquid heats up and the void fraction starts to grow from the point of Net Vapor Generation (NVG). Downstream of the point of NVG, the void fraction grows rapidly. Bubbles detach from the wall and move into the subcooled liquid. Some of these bubbles condense in

the subcooled liquid, enhancing heat transfer to the liquid. Further downstream, the bulk liquid reaches the saturation temperature and saturated boiling begins.



The wall boiling process can be broken down into three stages that are based upon the different types of heat transfer, as illustrated in the diagram below. The dark blue line shows the surface available for heat transfer to occur across vapor/liquid (bubble) and solid/liquid interfaces.



The method for calculating each type of heat transfer is described below:

- Evaporation – Modeling of heat transfer at the wall begins with a prediction of the nucleation site number density. This value determines the number of bubbles per unit area, which is based on the amount of superheating in the fluid adjacent to the wall. This number density is used, along with the bubble departure frequency and bubble departure diameter to model the evaporative heat flux across the nucleation sites.
- Quenching – This occurs when buoyancy causes a bubble to leave the surface, liquid fills the space that the bubble occupied, enhancing heat transfer. The resulting heat flux is modeled using the quenching heat transfer coefficient. The area over which quenching takes place is the bubble influence wall area fraction.
- Convective Heating – After a bubble has detached and moved away from the wall, the remaining area is available for convective heat transfer. The wall dryout area fraction specifies how much convective heat transfers to the liquid phase and how much of it transfers to the vapor phase.

The evaporation and quenching mechanisms can support much higher heat transfer rates than single-phase, forced convection.

The wall boiling model comprises several submodels to capture various aspects of the boiling process. It is recommended that you do not change the default properties for these submodels, unless you have supporting experimental data.

For guidelines on the wall boiling model for both new and expert users, see Setting Up a Wall Boiling Simulation. For instructions on setting up a wall boiling simulation, see Modeling Boiling.

Wall Boiling Model Reference

Theory	See Wall Boiling.
Provided By	*[Phase Interaction] Model Selection* > *Optional Models*
Example Node Path	**[Phase Interaction]** > **Models** > **Wall Boiling**
Requires	A Eulerian Multiphase simulation with the following models activated: *Material*: **Eulerian Multiphase**, **Multiphase Interaction** (Selected automatically) *Eulerian Multiphase Model*: **Multiphase Segregated Flow**, **Multiphase Equation of State** (Selected automatically), **Gradients** (Selected automatically) *Optional Models*: **Phase Coupled Fluid Energy** *Viscous Regime*: **Laminar** or **Turbulent** *Space*: **Three Dimensional**, **Axisymmetric**, or **Two Dimensional** *Time*: **Steady** or **Implicit Unsteady** A minimum of two Eulerian phases: one liquid phase and one gas phase. In the liquid phase, the following phase models activated: *Material*: **Liquid** or **Multi-Component Liquid** *Energy*: **Segregated Fluid Temperature** or **Segregated Fluid Enthalpy** *Reaction Regime*: **Non-reacting** (applies to multi-component phases only) *Equation of State*: **Constant Density** *Optional Models*: **Exact Wall Distance** In the gas phase, the following models activated: *Material*: **Gas** or **Multi-Component Gas** *Energy*: **Segregated Fluid Temperature** or **Segregated Fluid Enthalpy** *Reaction Regime*: **Non-reacting** (applies to multi-component phases only) *Equation of State*: **Constant Density** *Optional Models*: **Exact Wall Distance** A Continuous-Dispersed phase interaction or a Multiple Flow Regime phase interaction. In the phase interaction, the following models activated: *Optional Models*: **Interphase Mass Transfer** *Interphase Mass Transfer Rate*: **Boiling Mass Transfer Rate** (for a Continuous-Dispersed phase interaction) or **Boiling/Condensation** (for a Multiple Flow Regime phase interaction) *Optional Models*: **Wall Boiling**
Properties	Key properties are: *Relaxation Factor*. See Wall Boiling Properties.

Specific Right-Click Actions	None.	
Activates	Physics Models	**Wall Bubble Nucleation** See Wall Bubble Nucleation Properties. **Wall Transient Conduction** See Wall Transient Conduction Properties.
	Model Controls (child nodes)	**Wall Dryout Area Fraction** See Wall Dryout Area Fraction Properties.
	Boundary Inputs	See Boundary Settings.
	Field Functions	See Field Functions.

Wall Boiling Model Properties

Relaxation Factor

Adjusts the model sensitivity. Reduce it to improve convergence.

Wall Boiling Model Controls

The following child nodes are provided:

Wall Dryout Area Fraction

The wall dryout area fraction specifies how much of the heat flux that is applied at the wall is directed towards the vapor convection, as opposed to liquid convection and evaporation. It can be used to improve robustness during initial convergence, or for indicating the onset of Departure from Nucleate Boiling (DNB) conditions in converged solutions.

For more information, see Wall Dryout Area Fraction.

Wall Dryout Area Fraction Properties

Method

Basic Wall Dryout Area Fraction

Defines the area fraction as zero until the volume fraction of the vapor near the wall exceeds a specified value. It then transitions the area fraction to unity as the vapor volume fraction approaches unity.

To reduce grid-dependency, the near-wall volume fraction that is used for the area fraction calculation can optimally be defined as an average over a prescribed thickness of a notional bubbly layer next to the wall.

This method is the default.

Constant, Field Function, Table, and User Code

Standard methods provided for user-specified values.

Wall Dryout Breakpoint

The vapor volume fraction at which point heat transfer to the vapor phase begins (see Eqn. (2434)). The default value is 0.9.

This property is available for the **Basic Wall Dryout Area Fraction** method only.

Bubbly Layer Relaxation Factor

Controls the update of the Bubbly Layer Volume Fraction that is used by the dryout model. It helps convergence behavior once the dryout criterion has been exceeded. The default value is 0.5.

Bubbly Layer Option

Defines the thickness of the layer that is used for averaging volume fraction in the dryout criterion.

- *Wall Cell*
  The value that is computed at the cell center next to the wall represents the entire bubbly layer. This means the cell-center volume fraction is used in the dryout criterion. This is the default setting.
- *Fixed Number of Diameters*
  The bubbly layer thickness is defined by the local bubble departure diameter multiplied by this value.
- *Fixed Yplus*
  The bubbly layer thickness is defined by the wall turbulence length scale for the vapor phase multiplied by this value.

Wall Bubble Nucleation Properties

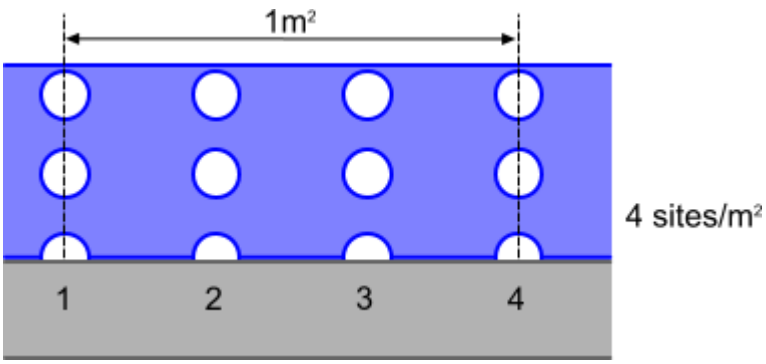
The following child nodes are provided:

- **Nucleation Site Number Density**
- **Bubble Departure Diameter**
- **Bubble Departure Frequency**

Nucleation Site Number Density Properties

The nucleation site number density determines the number of locations on the heated surface where bubbles form, per unit area.

This is the leading factor determining the evaporation rate in a mechanistic model of subcooled boiling (see Eqn. (2421)). The diagram below illustrates this parameter:



Method

The options for calculating the nucleation site number density.

- **Lemmert-Chawla**
- **Hibiki-Ishii**
- User-Defined: **Constant**, **Field Function**, **Table**, or **User Code**.

Lemmert-Chawla Properties

The Lemmert Chawla model ([455]) is a correlation with wall superheat and is the default model, recommended for initial studies on the grounds of robustness. The default settings assume that the number of active nucleation sites varies with the wall superheat to the power of 1.805. This value is used within the Tolubinsky and Kostanchuk model for calculating bubble departure diameter, for example for modeling forced convection, subcooled boiling of water at 45 bar. An alternative approach is discussed in the formulation section.

Calibration Constant A

A in Eqn. (2439). The default value of 1.805 is chosen according to Kurul and Podowski ([451]).

Calibration Constant B

B in Eqn. (2439). The default value is 0.0.

Maximum Superheat

ΔT_{max} in Eqn. (2438). Sets the upper limit for the wall superheat value that is passed to this submodel, which prevents overflow during initial convergence. Make sure that this limit does not affect the value of wall superheat predicted in the converged solution. The default value is 25.0 delta K.

Reference Nucleation Site Number Density

n_0'' in Eqn. (2439). The default value is 12366.44783 / m^2.

Reference Wall Superheat

ΔT_0 in Eqn. (2439). The default value is 1.0 K.

Hibiki-Ishii Properties

The Hibiki Ishii model is a more advanced nucleation site number density model. It is applicable for pressures up to 198 bar. Use the Hibiki Ishii model with the Kocamustafaogullari model for bubble departure diameter.

The model has the following characteristics:

- It considers a boundary condition for wall superheating.
- It is validated against numerous sets of experimental data.
- It has a wide range of applicability in terms of mass flow, pressure, and contact angle.

If you use a working fluid other than water, adjust the *Wall Contact Angle*. If you know average cavity density and the cavity length scale values for the working surface, adjust these also.

Maximum Superheat

Sets the upper limit for the wall superheat value that is passed to this submodel (see Eqn. (2440)), which prevents overflow during initial convergence. Make sure that this limit does not affect the value of wall superheat predicted in the converged solution.

Wall Contact Angle

Data that you define for your working fluid. A nominal value at room temperature is required. The default of 0.722 radians (41.37^{o}) is an approximate value for water systems (see Eqn. (2443)).

Average Cavity Density

The average number of nucleation sites per unit area, which is used when calculating the nucleation site number density (see Eqn. (2443)). This can be tuned to suit a particular surface finish.

Wall Contact Angle Scale

Reference value used in the model to account for dependence on *Wall Contact Angle* (see Eqn. (2443)). The standard value for most applications is 0.722 radians (41.37^{o}).

Cavity Length Scale

Value that is used when calculating the nucleation site number density (see Eqn. (2443)). This can be tuned to suit a particular surface finish.

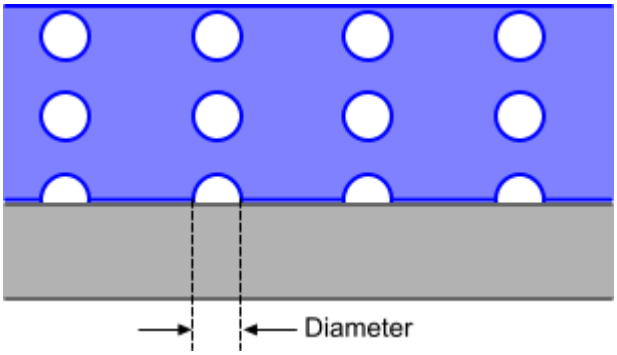
Density Function Constant C0, C1, C2, and C3

Values used in the model to account for the pressure, which is used when calculating nucleation site number density (see Eqn. (2445)). The standard values for most applications are -0.1064, 0.48246, -0.22712, and 0.05468.

Bubble Departure Diameter Properties

The bubble departure diameter determines the diameter of the bubble at the instant it leaves the nucleation site.

This is the second of the three factors used to determine the evaporation rate in subcooled boiling (see Eqn. (2421)). The diagram below illustrates this parameter:



Method

The options for calculating the bubble departure diameter.

- **Tolubinsky Kostanchuk**
- **Kocamustafaogullari**
- User-Defined: **Constant**, **Field Function**, **Table**, or **User Code**.

Tolubinsky Kostanchuk Properties

The Tolubinsky Kostanchuk model is a correlation against liquid subcooling and is the default model that is recommended for initial studies on the grounds of robustness. It matches the default Lemmert Chawla model for nucleation site number density.

Reference Diameter

Reference diameter that is used when calculating the bubble departure diameter (see Eqn. (2449)), as determined by experiment for a given subcooling of 45K.

Reference Diameter Subcooling

Reference diameter subcooling (that is, the difference between the saturation temperature and liquid temperature for the reference diameter) used when calculating the bubble departure diameter at the given value of subcooling (see Eqn. (2449)).

Minimum Diameter

Sets the lower limit for the bubble departure diameter. This setting makes sure that the result is within plausible limits.

Maximum Diameter

Sets the upper limit for the bubble departure diameter. This setting makes sure that the result is within plausible limits.

Kocamustafaogullari Properties

The Kocamustafaogullari model is more recent, more general and based on a force balance with an adjustment for pressure dependence. This model must be used whenever the Hibiki Ishii model is selected for nucleation site number density.

Calibration Constant

Calibration constant (see Eqn. (2451)). The standard value is 1.5126×10^{-3} m/radian (2.64×10^{-5} m/degree).

Wall Contact Angle

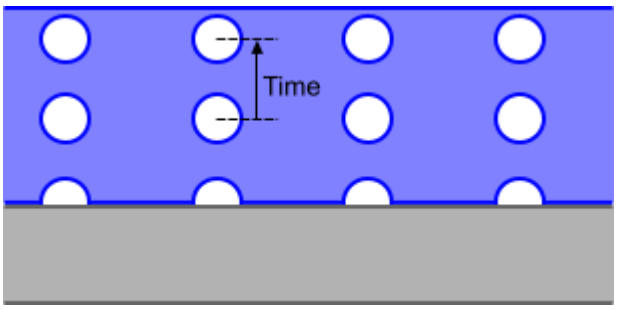
Data that you specify for your working fluid. A nominal value at room temperature is required. The default of 0.722 radians (41.37^{o}) is an approximate value for water systems (see Eqn. (2451)).

If you use a working fluid other than water, you can adjust the *Wall Contact Angle* as required.

Bubble Departure Frequency Properties

The bubble departure frequency determines how many bubbles leave a nucleation site per second.

This is the last of three key factors determining the evaporation rate in subcooled boiling (see Eqn. (2421)). The diagram below illustrates this parameter:



Method

The options for calculating the bubble departure frequency.

- **Cole**: This frequency is equivalent to a terminal velocity scale over bubble departure diameter, and the overall evaporation rate is calibrated around this assumption (see Eqn. (2452)). There are no adjustable parameters, but you can define an alternative model.
- User-Defined: **Constant**, **Field Function**, **Table**, or **User Code**.

Wall Transient Conduction Properties

This model corrects the Bubble Induced Quenching Heat Flux so that it uses the temperature of the liquid brought to the wall by the action of the departing bubble. This temperature can be different to the liquid temperature computed at the cell center next to the wall. The correction is taken from a location on the undisturbed liquid phase temperature profile away from the departure site.

Quenching Temperature Option

Defines the location for sampling the liquid temperature profile.

- *Wall Cell*
  The liquid temperature at the cell center next to the wall is used as the quenching temperature. This is the default setting.

- *Fixed Number of Diameters*
  The liquid temperature is estimated at a location that is specified as a fixed number of the local bubble departure diameter size. You set this in the **Fixed Number of Diameters** child node.
- *Fixed Yplus*
  The liquid temperature is estimated at a location that is specified as a fixed number of wall turbulence length scales for the liquid phase. You set this in the **Fixed Y+** child node.

The following child nodes are provided:

- **Bubble Influence Wall Area Fraction**
- **Bubble Induced Quenching Heat Transfer Coefficient**

Bubble Influence Wall Area Fraction Properties

The bubble influence wall area fraction estimates the fraction of the wall area that is affected by the sweep of liquid inflow beneath a departing bubble.

Method

The options for calculating the bubble influence wall area fraction.

- **Kurul Podowski**
- User-Defined: **Constant**, **Field Function**, **Table**, or **User Code**.

Kurul Podowski Properties

The Kurul Podowski model assumes that the wall area influenced by bubble-induced quenching is larger than the nucleation site area, by a specified factor (see Eqn. (2455)).

Area Coefficient

Ratio of nucleation site area density and bubble-influenced wall area density, which is given by (Eqn. (2455)). The default value is 2.0.

Bubble Induced Quenching Heat Transfer Coefficient Properties

When a bubble leaves the heated surface, cooler liquid fills the space that it occupied. The heat transfer during this process is known as quenching heat transfer. The Quenching Heat Transfer Coefficient is used to calculate the quenching heat flux.

Method

The options for calculating the bubble induced quenching heat transfer coefficient.

- **Del Valle Kenning**
- User-Defined: **Constant**, **Field Function**, **Table**, or **User Code**.

Del Valle Kenning Properties

This model calculates the bubble induced quenching heat transfer coefficient using Eqn. (2453).

Wait Coefficient

Proportion of bubble departure cycle period between departure of one bubble and nucleation of the next (see Eqn. (2454)).

Boundary Settings

Wall Boundaries

Wall Interphase Mass Transfer Option

Activates interphase mass transfer interaction on the wall boundary.

Method	Corresponding Physics Value Nodes
Active Allows wall boiling to occur on this boundary. Under the **[Wall Boundary Name]** > **Phase Conditions** node, for each phase in the phase interaction, set the User Wall Heat Flux Coefficient Specification.	None
None Prevents wall boiling from occurring on this boundary. For example, use this option when a boundary is adiabatic.	None

Field Functions

BubblyLayerVolumeFraction

α_δ in Eqn. (2433).

BubblyLayerVolumeFractionGradient

α'_g in Eqn. (2433).

BubbleDepartureDiameter

d_w in Eqn. (2449) for the **Tolubinsky Kostanchuk** model or Eqn. (2451) for the **Kocamustafaogullari** model.

BubbleDepartureFrequency

f in Eqn. (2452).

BubbleInfluenceWallAreaFraction

K_{quench} in Eqn. (2455).

BubbleInducedQuenchingHeatTransferCoefficient

h_{quench} in Eqn. (2453).

BubbleInducedQuenchingTemperatureFactor

s_{quench} in Eqn. (2428).

BubbleInducedQuenchingTemperatureDistance

y_{quench} in Eqn. (2428).

NucleationSiteNumberDensity

n'' in Eqn. (2439) for the **Lemmert Chawla** model or Eqn. (2443) for the **Hibiki-Ishii** model.

WallInterfaceMassFlux

$\dot{m}''$ in Eqn. (2426).

BoundaryHeatFlux<Liquid Phase Name>

q''_{wl} in Eqn. (2430).

BoundaryHeatFlux

q''_{wlg} in Eqn. (2431).

BoundaryHeatFlux<Vapor Phase Name>

q''_{wg} in Eqn. (2432).

InterphaseMassTransferRate

The mass transfer between phases due to a single reversible process, such as Bulk Boiling or Species Dissolution. This value is positive when the mass transfer is towards the first phase in the interaction. For continuous-disperse interactions, the first phase is the continuous phase. If the continuous phase is the liquid phase in a boiling problem, a positive rate signifies condensation and a negative rate signifies evaporation.

Modeling Particle-Phase Flows

Particle-particle interactions can be significant in solid particle flows and therefore require their own set of models.

STAR-CCM+ provides the Granular Pressure model for gas-solid flows, such as fluidized beds. The model introduces the concept of the “packing limit.” Below this limit, particles are loosely packed and inter-particle collisions determine their motion; these collisions are handled as part of a Granular Temperature model that becomes available when Granular Pressure is selected. Above the packing limit, friction between particles dominates their motion and a frictional stress model is used.

A granular phase can be modeled using either the Laminar or the Turbulent viscous regime. When the Turbulent viscous regime is used, the Standard Two-Layer K-Epsilon model, with Two-Layer Wall Treatment, is available. It is possible to work with multiple particle phases (that is, particles of different sizes) when using the Granular Pressure model.

The following sections describe the Granular Pressure model:

- What Is the Granular Pressure Model?
- What is the Granular Temperature Model?
- Setting Granular Pressure in a Particle Phase
- Setting Granular Pressure at Boundaries

In STAR-CCM+, the Solid Pressure Force model is provided for dispersed particle phases to account for the forces that arise during particle-particle interaction when the particles are packed closely together. It is a way of limiting the maximum particle volume fractions. The following sections are provided:

- What Is the Solid Pressure Force Model?

STAR-CCM+ also provides an Erosion model that simulates both impact erosion and abrasive wear on a wall. See the following sections:

- Modeling Erosion
- Erosion Formulation

Contents:

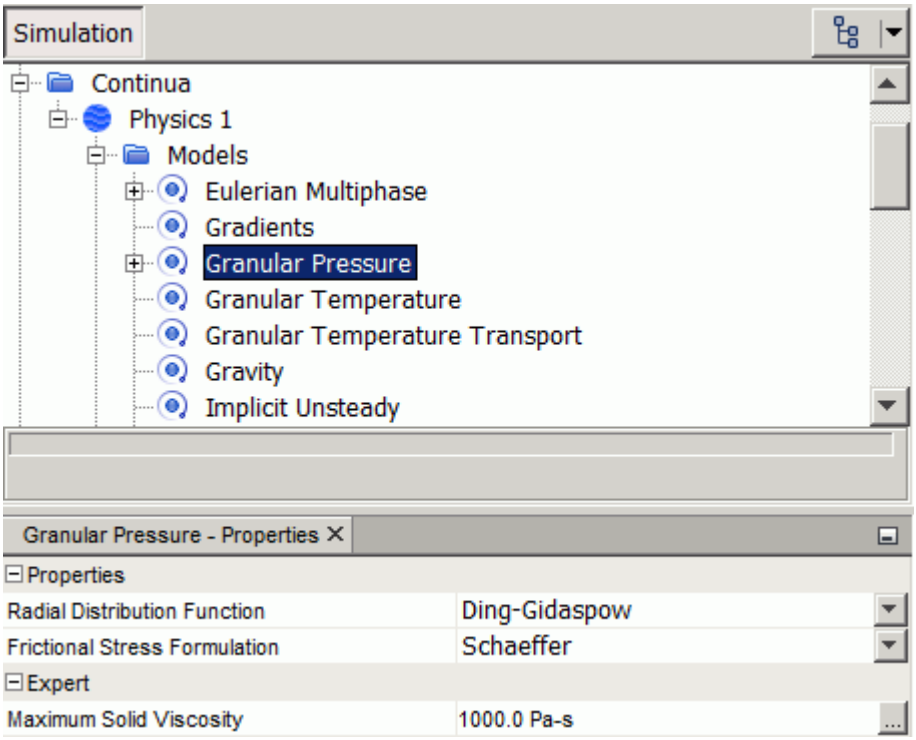
What Is the Granular Pressure Model?
What is the Granular Temperature Model?
What is the Granular Energy Transfer Model?
Setting Granular Pressure in a Particle Phase
Setting Granular Pressure at Boundaries
What Is the Solid Pressure Force Model?
Modeling Erosion

What Is the Granular Pressure Model?

The Granular Pressure model estimates the stresses of particle phases in a gas-solid fluidized bed.

Fluidized bed reactors are widely used in combustion, catalytic cracking and various other chemical and metallurgical processes. Granular pressures are also encountered in many industrial processes.

The closeness of particles to one another, or “packing,” determines the type of particle motion. Where the particles are closely packed, the motion is dominated by contact and a frictional stress model is used. When the particles are below the packing limit, then collisions and kinetic energy determine the motion, handled as part of the Granular Temperature model.



Granular Pressure Model Properties

Determines the regimes that are modeled.

Radial Distribution Function

- *Ding-Gidaspow*
  The **Ding-Gidaspow** radial distribution function uses Eqn. (2584) for packing below the packing limit and Eqn. (2585) for packing close to the packing limit.

Frictional Stress Formulation

Specifies the motion behavior of the particles when they approach the maximum packing limit.

- *Schaeffer*
  Uses the Schaeffer ([490]) frictional regime equations.
- *Modified Johnson*
  Uses the Modified Johnson ([444]) frictional regime equations.
- *Kinetic-Based*
  Uses the collisional solid pressure to limit the particle volume fraction near maximum packing. The collisional solid pressure relies on the correction to the particle radial distribution function near maximum packing (Eqn. (2585)).

Maximum Solid Viscosity

Specifies the maximum limit for solid viscosity, which is the summation of kinetic, collisional, and frictional contributions. This property is relevant for modeling contact-dominated granular flows. The default value is 1000 Pa-s.

This value is μ_p in Eqn. (2614).

Schaeffer Properties

Specifies the motion behavior of the particles. This behavior is based on the work by Schaeffer ([490]). Only available when the **Frictional Stress Formulation** property is set to **Schaeffer**.

Angle of Internal Friction	Used to determine the effective granular viscosity, which is given by Eqn. (2597). It is set to 25° by default.

Modified Johnson Properties

Specifies the motion behavior of the particles. This behavior is based on the work by Johnson and Jackson [444]. Only available when the **Frictional Stress Formulation** property is set to **Modified Johnson**.

Critical Volume Fraction	Solid volume fraction at which the frictional stress is activated. Set to 0.5 by default.
Angle of Internal Friction	Used to determine the effective granular viscosity, which is given by Eqn. (2597). Set to 28.5^{o} by default.

Modified Johnson Expert Properties

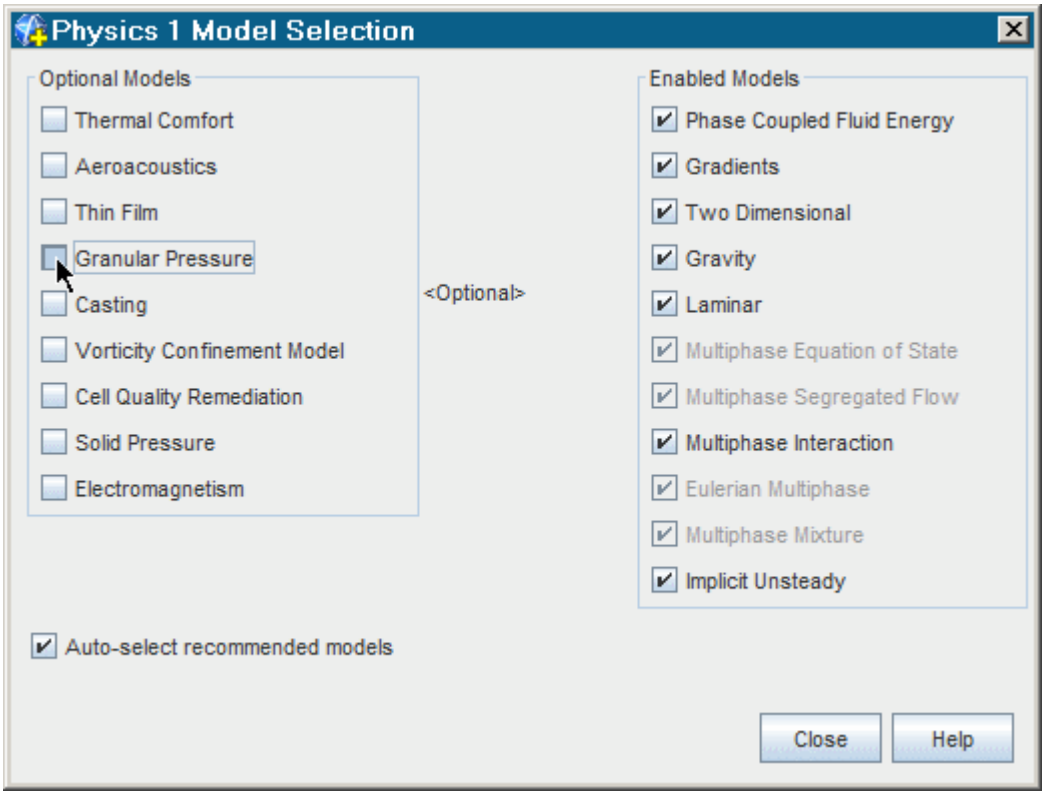
Multiplier Fr	Multiplication factor in Eqn. (2593). Set to 0.05 by default.
Exponent r	Numerator exponent in Eqn. (2593). Set to 2.0 by default.
Exponent s	Denominator exponent in Eqn. (2593). Set to 5.0 by default.

Selecting the Granular Pressure Model

This option becomes available after selecting the Multiphase Segregated Flow model.

To select the Granular Pressure Model:

In the *Physics Model Selection* dialog, select **Granular Pressure** in the *Optional Models* group box.



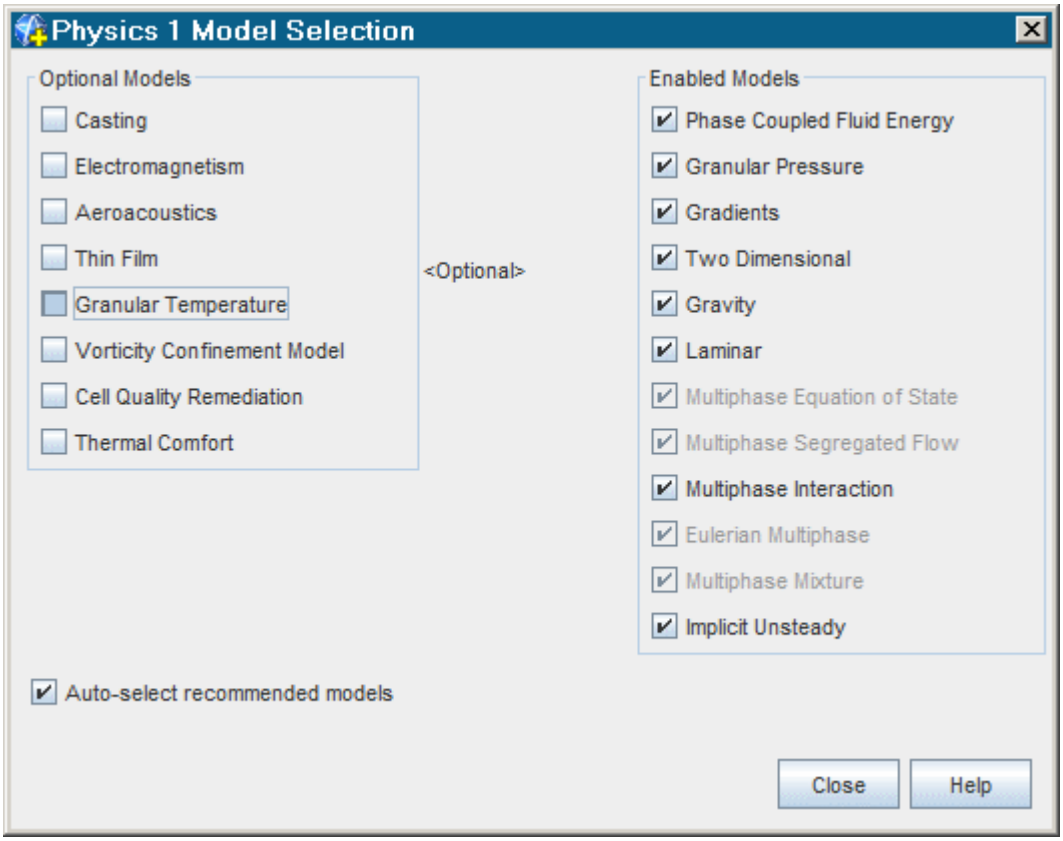
Once Granular Pressure has been activated for the continuum, you can specify relevant settings in the particle phase.

What is the Granular Temperature Model?

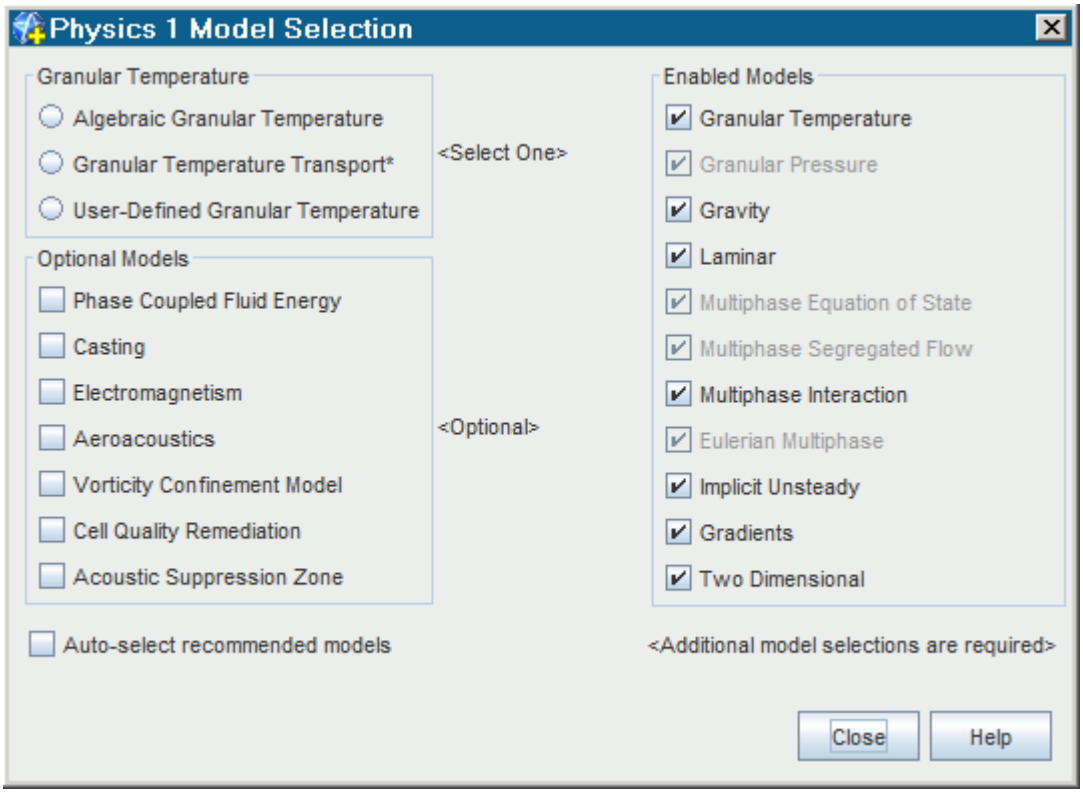
The Granular Temperature model calculates the collisions and motions of particles below the packing limit.

Select this model only if you wish to model the collision-dominated regime of flow. Do not select this model if you only wish to model the frictional regime.

The Granular Temperature model is selected automatically when you select the Granular Pressure model. You can deactivate the Granular Temperature model if you do not require it: the model then becomes available in the *Optional Models* section of the *Physics Model Selection* dialog.



When you select the Granular Temperature model, the Granular Temperature Transport model is selected automatically. However, if you deactivate Auto-select recommended models before you select the Granular Temperature model, three models are available in the *Granular Temperature* section of the *Physics Model Selection* dialog.



The Granular Temperature models are:

- Algebraic Granular Temperature

  This model applies to cases where the local equilibrium assumption holds. These cases include bubbling fluidized beds and applications involving packed beds of particles. This model is less computationally expensive than the Granular Temperature Transport model.

- Granular Temperature Transport

  This model applies to any granular flow case. It takes into account convection, production, diffusion, and dissipation of granular energy. It also accounts for dissipation of granular energy due to inter-phase drag. If you select this model, you must select the Granular Energy Transfer model for phase interactions involving the particle dispersed phase.

- User-Defined Granular Temperature

  This model lets you specify the minimum granular temperature directly.

Algebraic Granular Temperature Properties

This model uses Eqn. (2604) to solve for the granular temperature.

Granular Temperature Minimum	Sets a minimum value for the continuum. Set to 1.0E-10 m^2/s^2 by default.
Particle Kinetic Viscosity	Specifies the motion behavior of the particles below the maximum packing limit. Use the corresponding Linearized Drag Law with the chosen Kinetic viscosity.

	Gidaspow	Use the Gidaspow Kinetic Viscosity model (see Eqn. (2616)). Selected by default.
	Syamlal	Use the Syamlal Kinetic Viscosity model (see Eqn. (2617)).

Granular Temperature Transport Properties

When you select this model, STAR-CCM+ adds the Granular Temperature solver to the list of solvers.

Granular Temperature Minimum	Sets a minimum value for the continuum. Set to 1.0E-10 m^2/s^2 by default.	
Particle Kinetic Viscosity	Specifies the motion behavior of the particles below the maximum packing limit. Use the corresponding Linearized Drag Law with the chosen Kinetic viscosity.	
	Gidaspow	Use the Gidaspow Kinetic Viscosity model (see Eqn. (2616)). Selected by default.
	Syamlal	Use the Syamlal Kinetic Viscosity model (see Eqn. (2617)).
Granular Diffusion Coefficient	Specifies the method for calculating the diffusion coefficient in the Granular transport equation.	
	Gidaspow	Use the Gidaspow method (see Eqn. (2600)). Selected by default.
	Syamlal	Use the Syamlal method (see Eqn. (2601)).

User-Defined Granular Temperature Properties

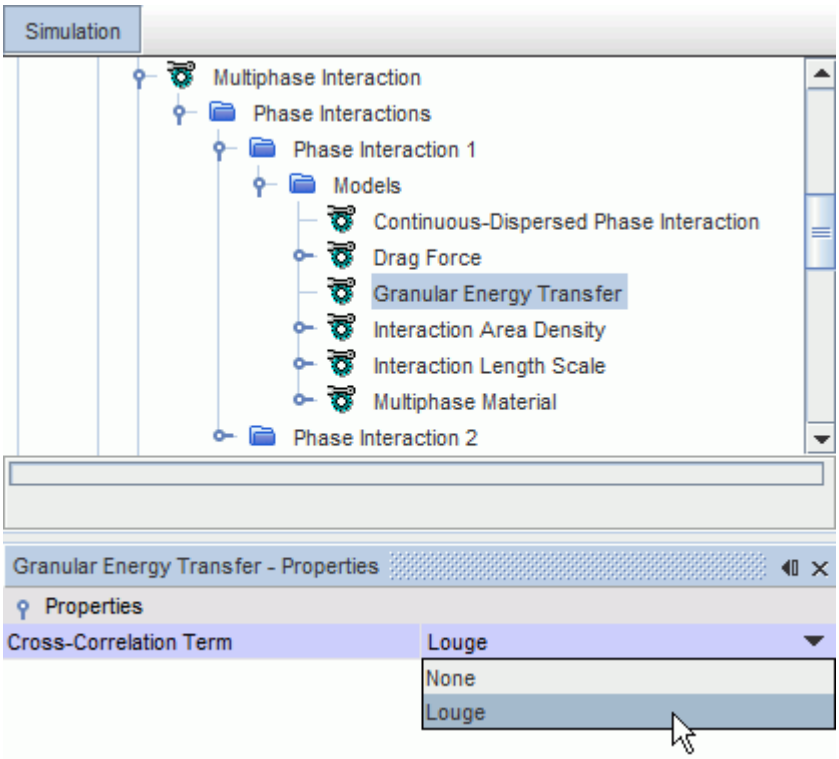
Granular Temperature Minimum	Sets a minimum value for the continuum. Set to 1.0E-10 m^2/s^2 by default.	
Particle Kinetic Viscosity	Specifies the motion behavior of the particles below the maximum packing limit. Use the corresponding Linearized Drag Law with the chosen Kinetic viscosity.	
	Gidaspow	Use the Gidaspow Kinetic Viscosity model (see Eqn. (2616)). Selected by default.
	Syamlal	Use the Syamlal Kinetic Viscosity model (see Eqn. (2617)).

What is the Granular Energy Transfer Model?

The Granular Energy Transfer model accounts for the dissipation of granular energy due to the fluctuating force exerted by the gas through the fluctuating velocity of the particles.

This model is selected automatically in a phase interaction involving a granular dispersed phase.

The Granular Energy Transfer model contributes the J_p term to the granular energy equation (Eqn. (2598)).

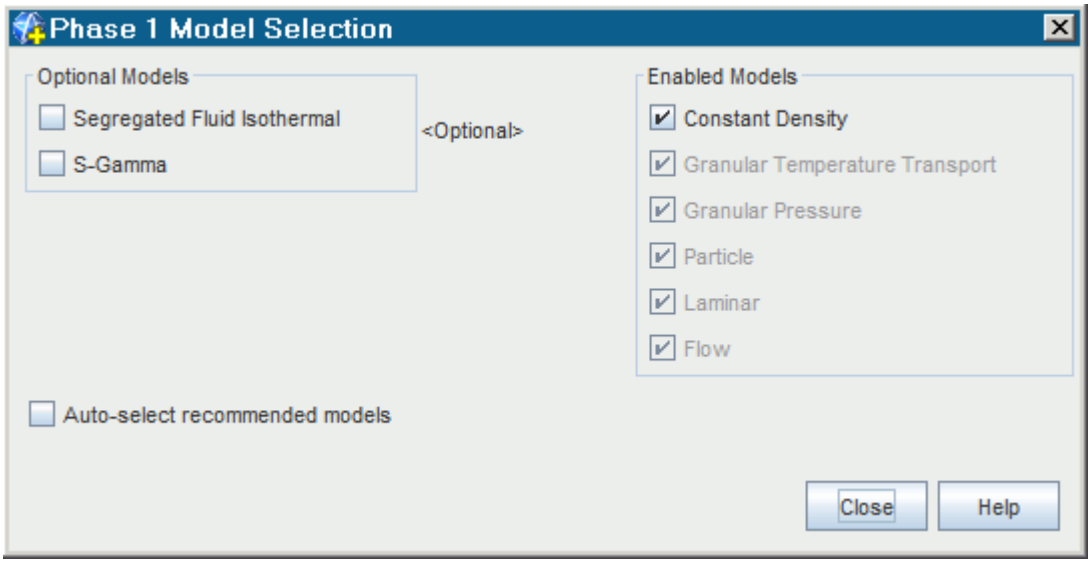


The properties of the Granular Energy Transfer model are as follows:

Cross-Correlation Term	Defines the contribution to the granular energy transport that is due to the correlation between the velocity fluctuations of the gas and the particles. This term is an optional source term. However, the additional term is expected to be significant for vigorously bubbling or circulating fluidized beds.	
	None	The granular transport model neglects this contribution.
	Louge	The granular transport model uses the source term proposed by Louge [464] in Eqn. (2603). For more information on the dissipation and generation of granular energy, see Granular Temperature.

Setting Granular Pressure in a Particle Phase

If you choose the **Granular Pressure** and **Granular Temperature** models for the continuum, the **Granular Pressure** model and one of the **Granular Temperature** models are automatically selected when you specify the phase type as **Particle**.



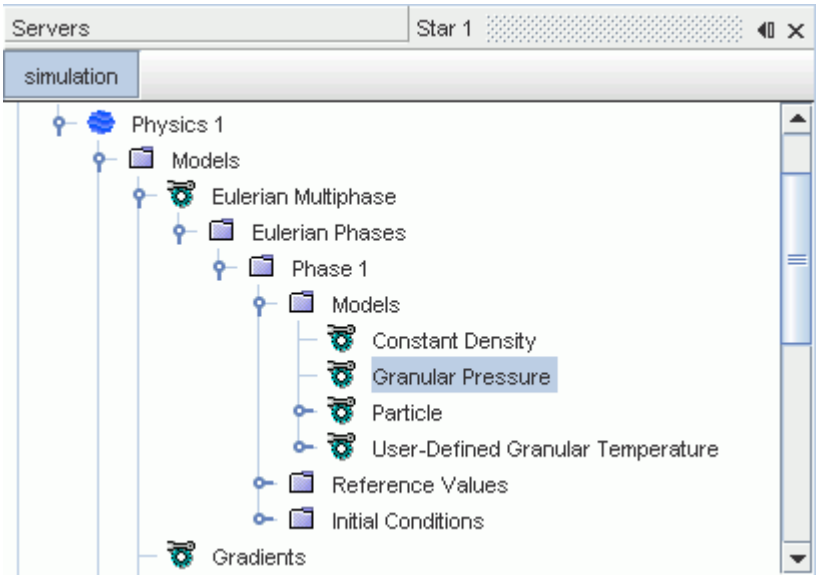
A granular phase can be modeled using either the Laminar or the Turbulent viscous regime. When the Turbulent viscous regime is used, the Standard Two-Layer K-Epsilon model, with Two-Layer Wall Treatment, is available.

Setting Phase Model Properties

This section describes the properties of granular pressure particle phase models.

Granular Pressure Properties

Sets the maximum packing limit for that phase.

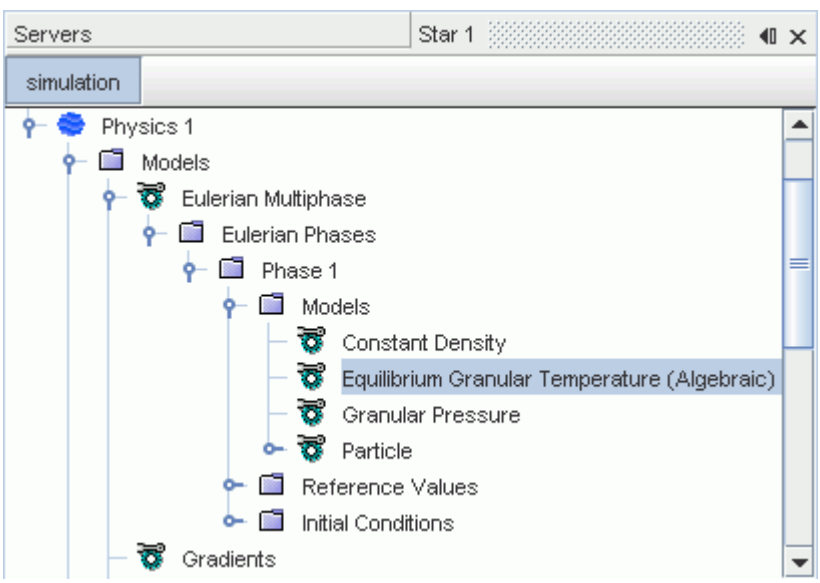


Maximum Solid Fraction	Set the upper packing limit of the particles. It is set to 0.624, the limit for rigid spherical particles, by default.

Algebraic Granular Temperature Properties

This option is available only when the Algebraic Granular Temperature model is selected in the physics continuum.

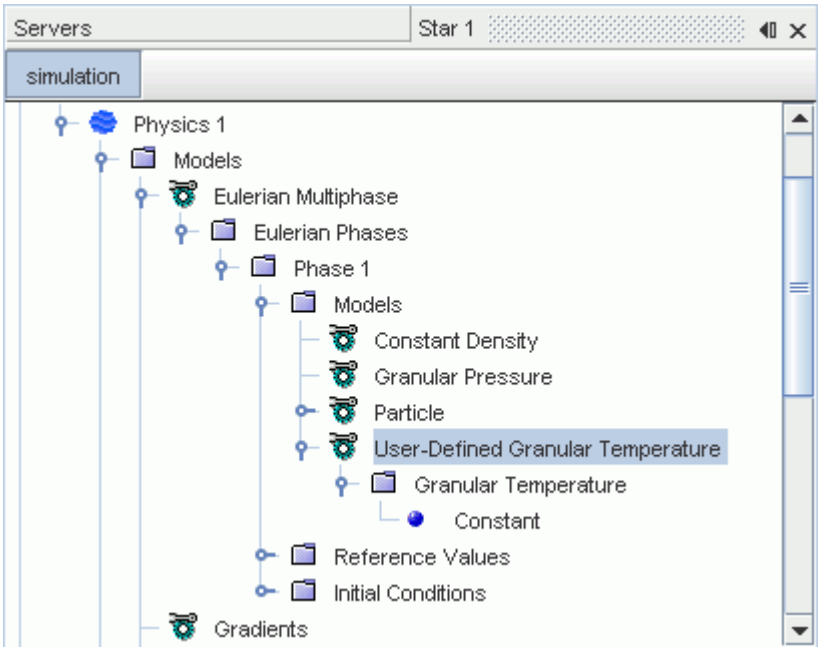
Uses Eqn. (2604) to solve for the granular temperature.



User-Defined Granular Temperature Properties

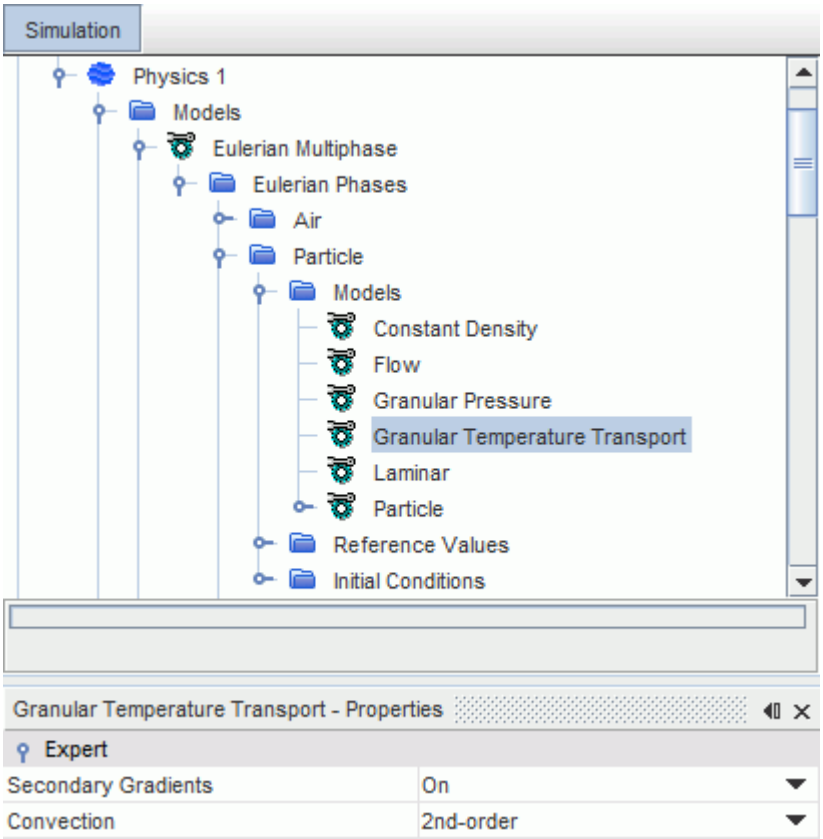
This option is available only when the User-Defined Granular Temperature model is selected in the physics continuum.

You can input your own values for granular temperature using a method that is specified in the **Granular Temperature** sub-node. A **Constant** value of 1.0E-6 m^2/s^2 is set by default.



Granular Temperature Transport Properties

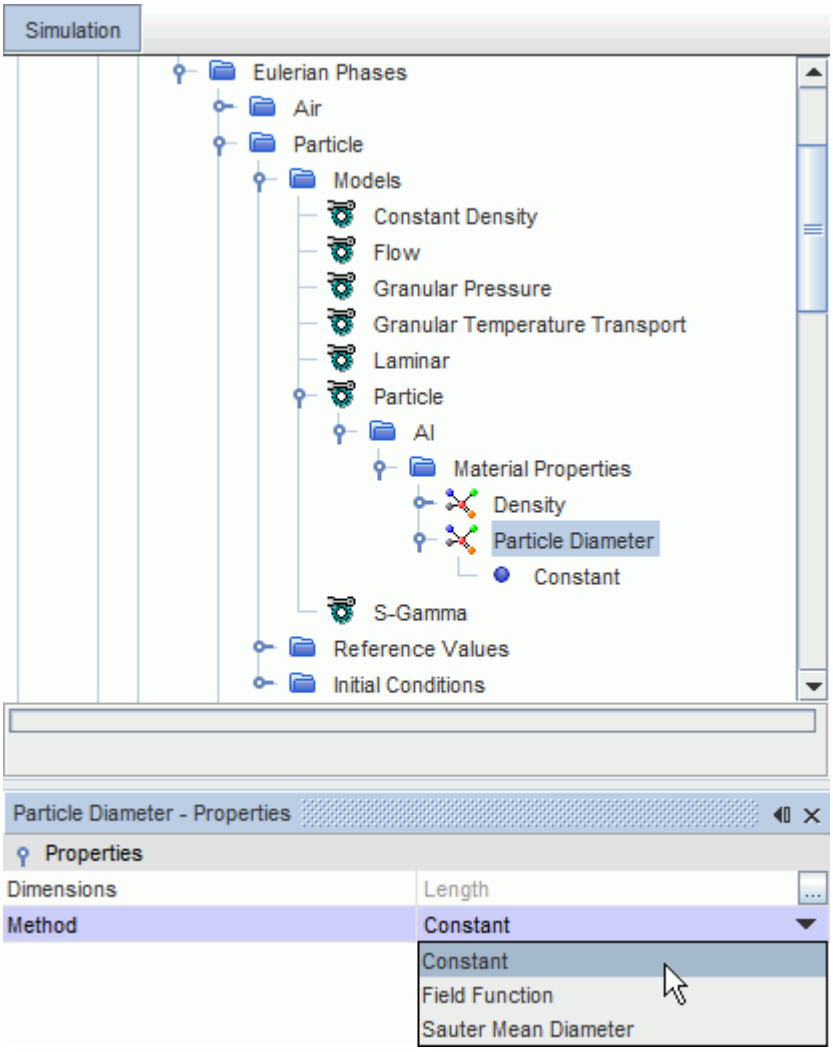
This option is available only when the Granular Temperature Transport model is selected in the physics continuum.



Secondary Gradients	Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces.	
	On	Include both secondary gradients.
	Off	Exclude both secondary gradients.
	Interior Only	Include the interior secondary gradients only.
	Boundaries Only	Include the boundary secondary gradients only.
Convection	Select the convection scheme for flow velocity.	
	1st-order	Selects the first-order upwind convection scheme.
	2nd-order	Selects the second-order upwind convection scheme.

Setting the Particle Diameter

You set the particle diameter in the **Material Properties** node in the particle phase definition.



Particle Diameter Properties

Method	Specifies the method for defining the particle diameter.	
	Constant	Specify a constant value that represents the mean particle diameter.
	Field Function	Specify a field function that represents the mean particle diameter.
	Sauter Mean Diameter	This method is made available when the S-gamma model is activated for the dispersed phase.

Setting Phase Reference Values

This section describes the properties of granular pressure particle phase reference values.

The phase **Reference Values** have the following properties:

Coefficient of Restitution Properties

Value	Represents the ratio of speeds of the particles before and after an impact. By default it is set to 0.9. A value of 1 indicates a perfectly elastic collision.

Setting Phase Initial Conditions

This section describes the properties of granular pressure particle phase initial conditions.

The phase **Initial Conditions** nodes are available only when any of the Granular Temperature submodels have been selected when selecting the Physics models.

Granular Temperature

You can input your own values for the initial granular temperature in a sub-node that is based on the method you select.

A **Constant** value of 1.0E-6 m^2/s^2 is set by default.

Velocity

You can specify the initial velocity as a vector. It is entered as a vector profile.

Volume Fraction

You can input your own values for the initial volume fraction in a sub-node that is based on the method you select.

Setting Granular Pressure at Boundaries

This section describes how to set the granular pressure at boundaries.

To set the granular pressure at boundaries:

- Choose the wall shear stress specification method.
- Choose the granular temperature specification method.
- Specify the appropriate values under the **Physics Values** node:

  Granular Temperature

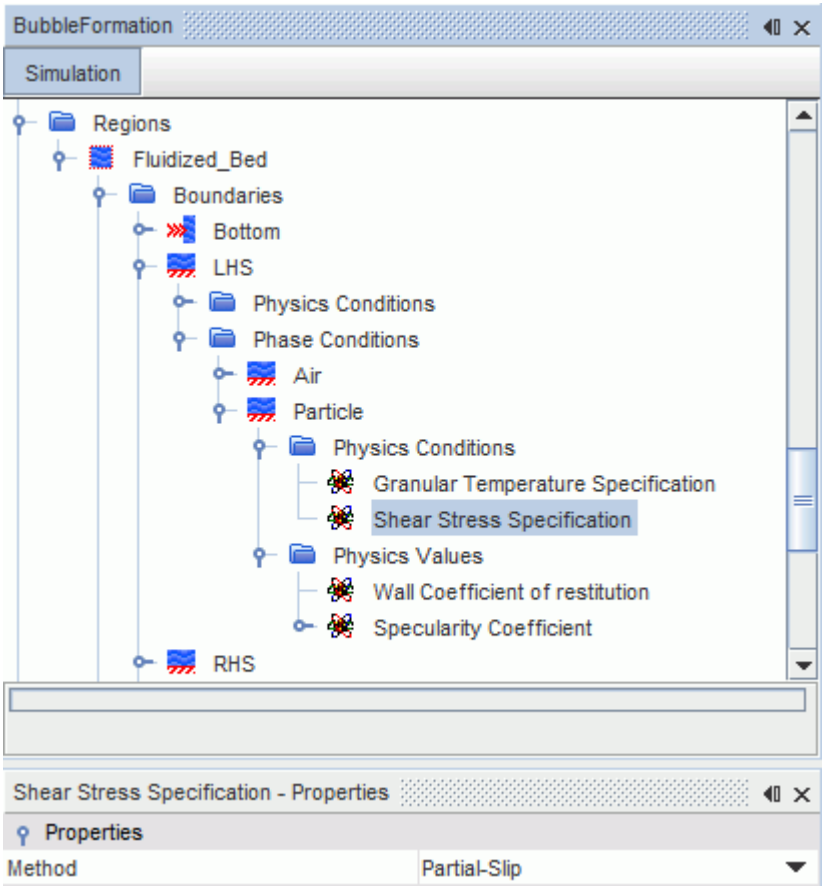
  Wall Coefficient of Restitution

  Specularity Coefficient

Setting the Wall Shear Stress Specification Method

The **Physics Conditions** > **Shear Stress Specification** node lets you choose the wall shear stress specification method.

You are advised to set the **Shear Stress Specification** for the particle phase to Slip at wall boundaries. If an estimate of the Specularity Coefficient is available, set this option to **Partial-Slip** and specify the appropriate value for the **Specularity Coefficient**.



Shear Stress Specification Properties

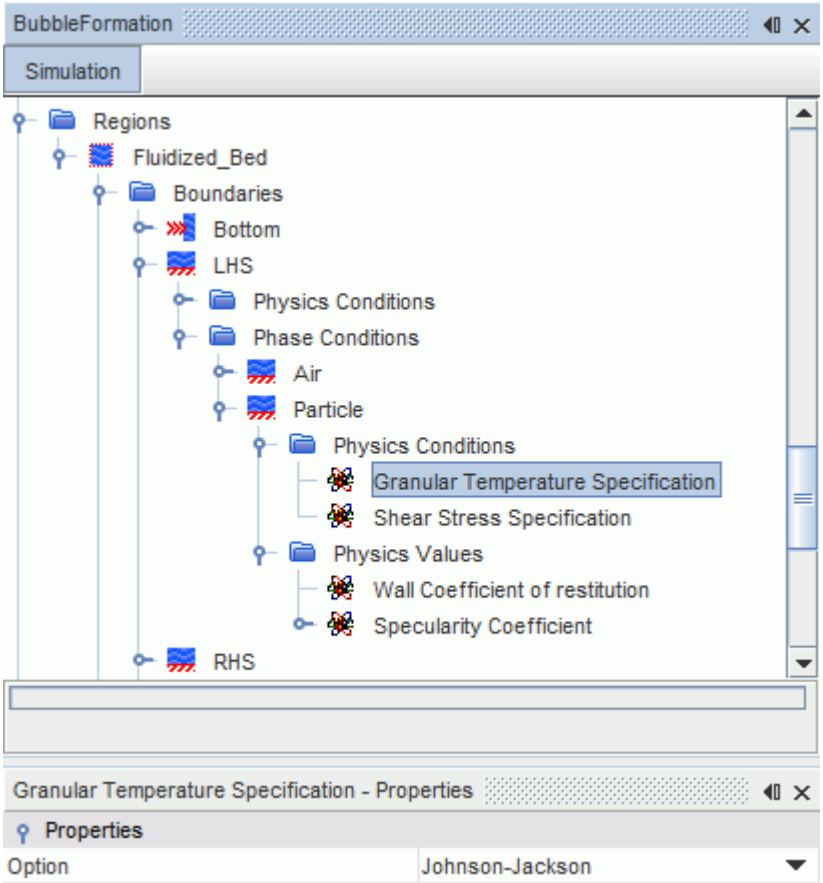
Method	Defines the wall shear stress specification method.	
	No-Slip	The tangential velocity is set to 0.
	Partial-Slip	Adds the Specularity Coefficient in a **Physics Values** node.
	Slip	No shear stress at the boundary.

Setting the Granular Temperature Specification Method

The **Physics Conditions** > **Granular Temperature Specification** node lets you choose the granular temperature specification method.

To estimate the granular temperature at boundaries, either extrapolate the granular temperature at the face cell, use the Johnson-Jackson method, or use a user-defined value.

The granular temperature value that is specified is used to calculate the solid pressure at the boundaries, which has a significant influence on results.



Granular Temperature Specification Properties

The *Option* property determines the method of inputting the granular temperature for the particle phase at your chosen boundary.

Extrapolation	Extrapolate from a face cell to the boundary. This option is not available if the wall shear stress specification method is set to *Partial-Slip*.
User	Specify your own boundary values for the Granular Temperature in a **Physics Values** node.
Johnson-Jackson	Use the Johnson-Jackson model for wall shear stress for particle phases in granular flows. This model is a more physically sound method of estimating granular temperature near the walls. This model can be used only with the Partial-Slip wall shear stress specification method. This option is available only when the Granular Temperature Transport model is selected.

	Specify the *Wall Coefficient of Restitution* and the *Specularity Coefficient* in the relevant **Physics Values** nodes. These values are e_w and ϕ in Eqn. (2619).

Granular Temperature Properties

Method	Sets the method to use for specifying your own values for granular temperature. A **Constant** value of 1.0E-6 m^2/s^2 is set by default.

Wall Coefficient of Restitution Properties

Value	The coefficient of restitution for collisions between particles and the wall. This value represents the ratio of speeds of the particles before and after an impact. By default the coefficient of restitution is set to 0.9. A value of 1 indicates a perfectly elastic collision.

Specularity Coefficient Properties

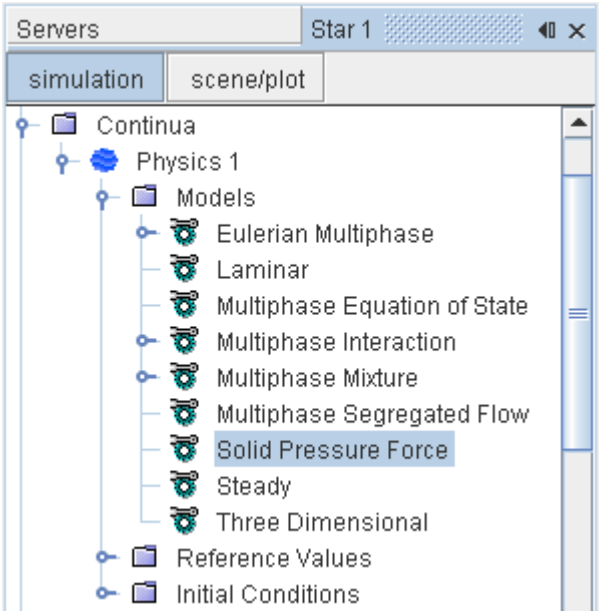
Method	Sets the method to use for specifying your own values for the specularity coefficient. A value of 0 corresponds to *Slip*, while a value of 1 corresponds to *No-Slip*.

You set the specularity coefficient only when the wall *Shear Stress Specification* method is set to *Partial-Slip*. The specularity coefficient does not depend on the granular temperature specification.

What Is the Solid Pressure Force Model?

The Solid Pressure Force model adds a compaction or ‘solid pressure’ force to the momentum equations of all the dispersed particle phases. This force becomes significant as the specified maximum packing limit is reached, and acts to repel particles and prevent them from exceeding the maximum volume fraction.

The **Solid Pressure Force** node and its properties represent the Solid Pressure Force model.



Solid Pressure Force Properties

Maximum Solid Fraction	The value of the maximum packing limit $\alpha_{p,max}$ in Eqn. (2580), which is set to 0.624 by default (representative of rigid spherical particles).

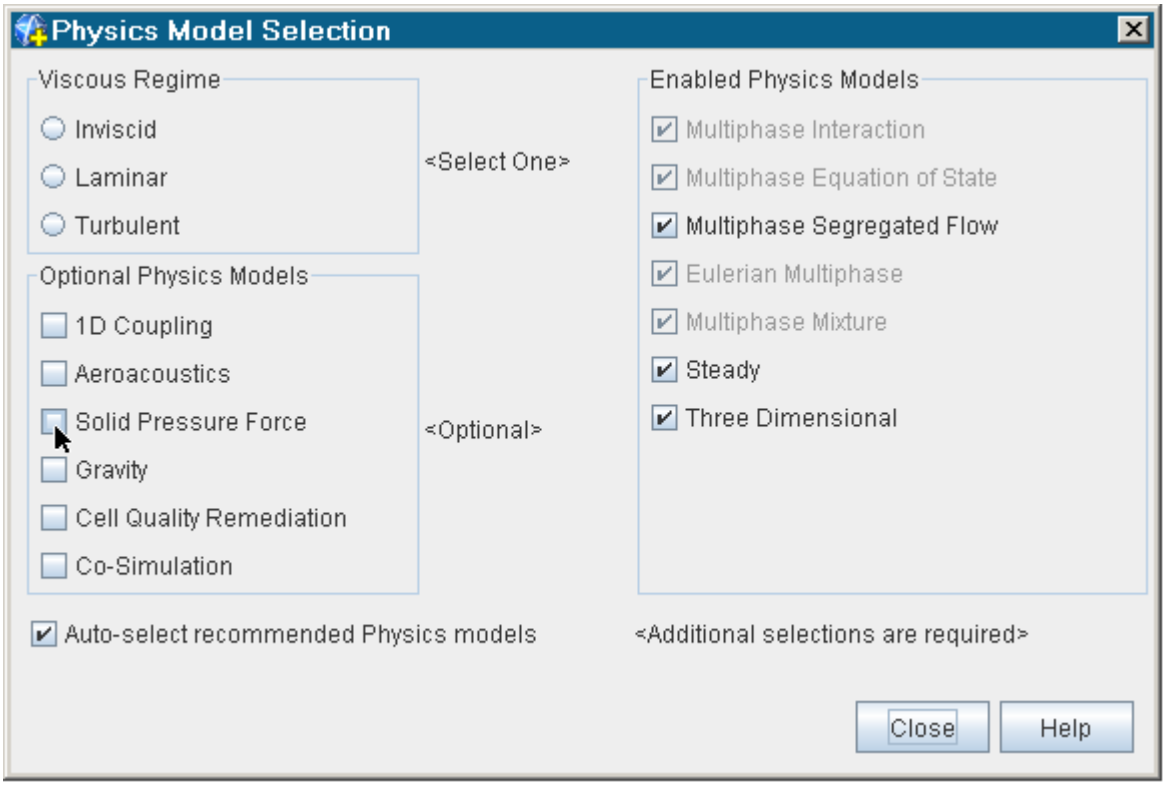
Expert Properties

Solid Pressure Force Constant	The model parameter A in Eqn. (2580), which is set to -600 by default.

Selecting the Solid Pressure Force Model

The Solid Pressure Force Model is selected through the **Solid Pressure Force** option of the *Optional Models* section of the *Physics Model Selection* dialog.

The *Physics Model Selection* dialog is shown below. This option becomes available after selecting the Multiphase Segregated Flow model.



Modeling Erosion

Particle flows can erode wall boundaries. The Erosion model accounts for both impact erosion and abrasive erosion.

Theory	See Erosion.	
Provided By	**[Phase]** > **Models** > *Optional Models*	
Example Node Path	**Eulerian Multiphase** > **Eulerian Phases** > **Phase 1** > **Models** > **Erosion**	
Requires	[physics continuum] *Material*: **Eulerian Multiphase** *Eulerian Multiphase Model*: **Multiphase Segregated Flow** *Optional Models*: **Granular Pressure** [phase] *Material*: **Particle**	
Properties	None.	
Activates	Boundary Inputs	See Boundary Settings.
	Field Functions	See Field Functions.

Boundary Settings

Erosion occurs only at wall boundaries.

Wall Boundary

Erosion Specification

For each particle phase, the **Erosion Specification** physics condition specifies the type of erosion that occurs at the wall boundary.

Method	Corresponding Physics Value Nodes
Impact Impact erosion only.	**Impact Wear** Specify the Impact Wear method: • Ahlert • DNV • Field Function • Neilson-Gilchrist • Oka Set the appropriate properties.
Abrasive Abrasive erosion only.	**Abrasive Wear** Specify the Abrasive Wear method: • Archard Set the appropriate **Abrasive Wear Coefficient**. • Field Function Specify the Scalar Function.
Impact + Abrasive Both Impact and Abrasive erosion.	**Abrasive Wear** As above. **Impact Wear** As above.

Field Functions

Abrasive Wear Rate

The abrasive wear rate, E_w, is the mass of wall material that is eroded due to abrasion per unit area per unit time. See Eqn. (2635). When multiple particle phases are specified, each phase contributes to the abrasion rate according to its granular temperature, density, and velocity (which are phase-specific quantities in the equation). This field function provides the sum of the contributions from each phase.

Available when Abrasive Wear erosion is activated.

Erosion Rate

The cumulative erosion rate, $E_w + E_f$.

Impact Wear Rate

The impact wear rate, E_f, is the mass of wall material that is eroded due to particle impact per unit area per unit time. See Eqn. (2631). When multiple particle phases are specified, this field function provides the sum of the contributions from each phase.

Available when Impact Wear erosion is activated.

Incidence Angle of [phase]

The impact angle for each particle phase. This angle is α in the erosion ratio formulation.

Modeling Particle Size Distribution

Particle size is a vital parameter for calculating particle hydrodynamics in dispersed multiphase flows.

In flows where the particles in the dispersed phase are bubbles, for example, the size of the bubbles can change continuously due to breakup and coalescence. As interfacial transfer terms are related to the surface area of the dispersed phase, it is essential to account for the particle size and its distribution to correctly simulate the multiphase flow behavior.

STAR-CCM+ provides the following particle size distribution models:

- Adaptive Multiple Size-Group (AMUSIG) model

  The AMUSIG model predicts the size distribution of particles (droplets or bubbles) in a dispersed flow regime of a multiphase flow. The set of particles is modeled by a number of groups of representative computational particles. Each group consists of identical particles, but each group has its own size and can have its own velocity. The method is locally adaptive so that each size group carries a meaningful share of the particle concentration. This model has a Multi-Speed option that allows you to study flow-size interactions and separation by size. This implementation is based on the work by Vikhansky and Splawski [510].

  See Adaptive Multiple Size-Group Model.

- S-Gamma model

  The S-Gamma model is an established model for predicting particle size distribution, with one-moment or two-moment options. This model is efficient to run and is compatible with a wide range of Eulerian multiphase models. This model assumes a log-normal distribution of particle sizes and is based on predicting the transport of the moments of the particle size distribution. This implementation is based on the work by Lo and Rao [460], and Lo and Zhang [462].

  See S-Gamma Model.

Contents:

Adaptive Multiple Size-Group Model
S-Gamma Model
Modeling Particle Size Distribution Workflow

Adaptive Multiple Size-Group (AMUSIG) Model Reference
AMUSIG Phase Interaction Model Family Reference
S-Gamma Model Reference
S-Gamma Phase Interaction Model Reference

Adaptive Multiple Size-Group Model

The Adaptive Multiple Size-Group (AMUSIG) model predicts the size distribution of droplets or bubbles in a dispersed flow regime of a multiphase flow.

The collection of droplets or bubbles is modeled using a number of groups of representative computational particles. Each group consists of identical particles, has its own size, and optionally can have its own velocity. A population balance is solved for each size-group in each cell. Breakup and coalescence models provide the principal contributions to the birth and death source terms for each balance. Other relevant effects include particle diffusion, acceleration, separation, and phase density changes. The solution to the balance equation gives the local particle number density for the group. The particle diameter is computed from the number density and the volume fraction for the group.

The method is locally adaptive so that each size-group carries approximately the same share of the phase mass or volume. This method is efficient and benefits fully from the computational expense of multi-speed modeling. Such modeling is necessary when analysing strong interactions between the size distribution and the flow. However, adaptation also means that the particle size that is associated with a group can vary gradually from cell to cell. Take care when you are comparing or integrating group results from different parts of the domain.

The Adaptive Multiple Size-Group model has two versions:

- One-Speed, for modeling flow-size interactions in the mean.

  Mass and momentum equations are solved at the phase level, and all of the particle size groups are assumed to be convected at phase velocity. This option allows STAR-CCM+ to model the effect of the flow and turbulence on the size distribution. The predicted mean particle size is passed back to the flow calculation.
- Multi-Speed, for modeling flow-size interactions in full.

  Mass and momentum equations are solved individually for each particle size-group. This option is more expensive, but is justified where there is strong interaction between the flow and the local size distribution. For example, this option is relevant for flows that separate particles by size. It is also relevant for flows where the interaction of buoyancy, turbulence, coalescence, and breakup cause different secondary flow patterns for different particle sizes, and therefore further promote overall turbulence and mixing.

Breakup and Coalescence Models

For laminar flows, coalescence and breakup are driven by the shear rate of the continuous phase. For turbulent flows, coalescence and breakup are driven by the turbulence scales of the continuous phase. The breakup and coalescence rates have some sensitivity to the turbulence models that are selected for the continuous phase and for particle-induced turbulence. To obtain comparable results with different turbulence models, you may need to recalibrate the models.

The STAR-CCM+ breakup and coalescence models for AMUSIG are not suitable for particulate flow. However, you can use AMUSIG to model the size distribution of solid particle phases in a particulate flow: use your own understanding and knowledge of coalescence and breakup phenomena for the application of interest to calibrate power law coalescence and breakup models as appropriate.

A set of models is available for describing breakup and coalescence processes for different applications:

- The model of Luo ([466]) is provided for turbulent coalescence.
- The model of Tsouris and Tavlarides ([467]), the model of Martinez-Bazan ([467]), and the model of Coulaloglou and Eskin ([427]) are provided for break-up in turbulent liquid/liquid and bubbly flows.
- Shear-induced breakup and coalescence models are provided for laminar flows.
- A generic film drainage coalescence model is provided for laminar and turbulent flows.
- Constant and power-law models are also implemented for testing and demonstration purposes, and to provide some user-specified modeling.

The Coalescence Rate model has submodels for Collision Rate and Coalescence Efficiency. The Breakup Rate model has submodels for Breakage Rate and Fragments Particle Size Distribution, which defines a removal term for particles of a given size. All of the rate models have an overall calibration constant which you can adjust to suit your requirements. The Critical Weber number is an adjustable parameter that is relevant for the breakup rate models.

A single size-group is sufficient for modeling hydrostatic expansion of dilute bubbles rising in a deep tank or column. However, for a meaningful collision/breakup study, a minimum of three groups is recommended. The adaptivity of the AMUSIG model allows you to investigate significant and rapid changes of particle size with a few groups. Increasing the number of groups should lead to resolution-independent results, in a manner similar to mesh refinement. Computational cost sets the practical upper limit on the number of size-groups.

For an overview of the main breakup models, see Breakup Models. For further details, see [467].

For an overview of the main coalescence models, see Coalescence Models. For further details, see [466].

Limitations and Restrictions

The Adaptive Multiple Size-Group model has the following general restrictions:

- AMUSIG is not available with Large Scale Interface (LSI) models, Fluid Film models, or Granular models.
- AMUSIG is a dispersed phase model, so it does not apply to phases that have a non-Newtonian rheology.
- The AMUSIG model is not intended to run through porous regions or porous baffles. Any results that are obtained would not be meaningful.
- The AMUSIG model is not intended to run with Large Eddy Simulation. This combination is likely to be expensive and not very accurate.
- Mass and momentum sources are available under one-speed AMUSIG. This model combination is of limited value since there is no corresponding source term for number density equations. However, this model combination can be useful to regulate system mass in systems that have closed or periodic boundaries, if the source term goes to zero when the system has converged.
- AMUSIG is not available for boiling or for other forms of interphase mass transfer.
- AMUSIG does not support a Volume Fraction Extrapolation boundary condition for backflow at Pressure Boundaries.
- AMUSIG does not support Mixing Plane interfaces.

- Particle flows can run with One-Speed AMUSIG only. However, only the power-law forms for breakup and coalescence are provided to explore modeling of particle breakup and coagulation.
- Passive Scalar modeling at the phase level is available for the dispersed phase, under both One-Speed and Multi-Speed AMUSIG.

The following restrictions apply to multi-speed AMUSIG only:

- Only Turbulent Response models can be used for modeling disperse phase turbulence.
- Mass and momentum sources are not available.
- The User Potential Force model is not available.
- The phase energy equation is not available in the disperse phase.
- Multi-component models are not available in the disperse phase.
- Some advanced field functions (for example, Vorticity) are not available.

S-Gamma Model

The S-Gamma model was developed by Lo and Rao, and Lo and Zhang, and assumes a log-normal distribution of particle sizes.

The model (see Lo and Rao [460], and Lo and Zhang [462]) is based on predicting the transport of the moments of the particle size distribution:

- Zeroth moment: the particle number density, n.
- Second moment: related to the interfacial area density.
- Third moment: related to the dispersed phase volume fraction.

One transport equation is solved for each moment. As the third moment is based on the dispersed phase volume fraction, it can be derived from the volume fraction equation that is solved by the Multiphase Segregated Flow solver. You can choose whether to solve for just the second moment, or for the zeroth and second moments, by selecting the required option in the properties of the S-Gamma model. More details are provided in the formulation section for this model.

Particles change size due to breakup and coalescence as they interact with each other. Additional models are provided to account for this interaction between phases. These models are selected on the multiphase interaction node for a pair of Eulerian phases.

The log-normal size distribution is defined by a mean particle diameter and its variance. The mean particle diameter is always updated in the calculation. However, the variance is updated only when the zeroth moment is solved in addition to the second moment, and breakup and/or coalescence is active. If the zeroth moment is not solved, the variance of the distribution remains at the initial value that you provide. This initial value could be zero.

Extensions to the S-Gamma model can be supplied using the source terms for the zeroth and second moments.

S-Gamma Breakup Model

For breakup, the S-Gamma model considers the balance between disruptive forces (due to shear and turbulence) and restoring forces (due to surface tension) on the particle (such as a droplet). In laminar flows,

the viscous effects dominate hence this regime is named “viscous breakup”. In turbulent flows, the interactions with turbulence eddies dominate and this regime is named “inertial breakup”.

S-Gamma Coalescence Model

For coalescence, the S-Gamma model considers the probability of collisions of the particles (such as droplets), the contact time of two colliding particles and the drainage time of liquid film between the particles. Similar to the breakup model, there is the “viscous collision” (or viscous coalescence) regime and the “inertial collision” (or inertial coalescence) regime. The drainage time is a function of the state of the particle surface, whether it is fully or partially mobile or immobile. The model therefore considers the breakup and coalescence processes in great detail.

It is possible to post-process certain results through field functions that these models activate.

Modeling Particle Size Distribution Workflow

To correctly simulate the flow behavior in dispersed multiphase flows, you must account for the particle size distribution.

The workflow for modeling the particle size distribution is:

1. Set up a Multiphase Segregated Flow simulation that includes the following:
   - A viscous regime activated: either **Laminar** or **Turbulent**.
   - At least two phases: a continuous phase and dispersed phase.
2. For the dispersed phase (under **Eulerian Multiphase** > **Eulerian Phases**), in the phase model selection dialog, activate the **Particle Size Distribution** model.
3. In the *Particle Size Distribution* group box, choose the model to use:
   - **Adaptive Multiple Size-Group**

     See Using the Adaptive Multiple Size-Group Model.
   - **S-Gamma**

     See Using the S-Gamma Model.

Using the Adaptive Multiple Size-Group Model

The usual range of turbulence models are available in the continuous phase. To get the best results with population balance modeling, the models in order of preference are Reynolds Stress, k-omega, and k-epsilon. A Turbulence Response model is used in the dispersed phase. In a dispersed multiphase flow, the AMUSIG model works with the Turbulence Response model for the dispersed phase, and can be run with either the One-Speed or Multi-Speed options.

The workflow for using the Adaptive Multi Size Group model is as follows:

1. In the dispersed phase, set the **Adaptive Multiple Size-Group** model properties:
   - **Number of Size Groups**

     At least three groups are required for most practical purposes.

     If you change the number of size groups, all the fields for all the size groups are automatically reinitialized.

- **Velocity Option**

  Select **One-Speed** or **Multi-Speed**. This selection depends on the scenario that you are modeling:

  - The Multi-Speed option is relevant when different size groups within the local size distribution have velocities or directions that are significantly different from the motion of the mean particle size.
  - For steady-state calculations, it can be useful to converge a large calculation first in One-Speed before switching over to a more expensive Multi-Speed calculation. For true transient simulations, switching between One-Speed and Multi-Speed may be less meaningful.
  - Always complete a whole time-step before modifying any of the models.

See Adaptive Multiple Size-Group Properties.

2. Set the initial conditions in the dispersed phase: select the **Particle Size Distribution** option, and then set the relevant initial condition nodes.

   See Initial Conditions.

3. Create a phase interaction and activate the **Continuous-Dispersed Phase Interaction** model.
4. On the *[phase interaction]* node, specify the continuous and dispersed phases.
5. In the phase interaction model selection dialog, activate the appropriate Breakup and Coalescence models, and set the necessary Breakup and Coalescence model properties.

   There are a number of models available, some of which can be used interchangeably. By default, a suitable combination of models is activated automatically when you select a Breakup or Coalescence model group.

   See Breakup Models and Breakage Rate Models.

   See Coalescence Models and Collision Rate Models.

6. Set the particle size distribution boundary condition for the inlets and pressure boundaries:

   In the Physics Conditions of the dispersed phase, set the **Particle Size Distribution** method, and then set the relevant Physics Values. The process is similar to setting the initial conditions of the dispersed phase.

   See Boundary Settings.

7. Set the Multiple Size-Group solver properties to suit your requirements. The default relaxation factors give reliable convergence in One-Speed cases.

   Coalescence rates increase rapidly with particle concentration. Higher concentrations can require additional relaxation, as well as confirmation that the concentration is being modeled realistically with suitable models for turbulence, for turbulence dispersion, and for breakup.

   See Multiple Size-Group Solver Properties.

Using the S-Gamma Model

The workflow for using the S-Gamma model is as follows:

1. In the dispersed phase, set the S-Gamma properties.

   See S-Gamma Model Properties.

2. Set the initial conditions in the dispersed phase: select the **Size Distribution Specification** method, and then set the relevant condition nodes: **Sauter Mean Diameter** and **Particle Size Variance**.

See Initial Conditions.

3. Create a phase interaction and activate the **Continuous-Dispersed Phase Interaction** model.
4. Specify the continuous and dispersed phases.
5. In the phase interaction model selection dialog, activate one or both of the **S-Gamma Breakup** and **S-Gamma Coalescence** models.

   See S-Gamma Phase Interaction Model Reference.
6. Set the necessary properties and values in child nodes. These properties and values can be different for different viscous regimes.

   In laminar flows, the viscous effects dominate, so this regime is named “viscous breakup”. In turbulent flows, the interactions with turbulence eddies dominate, so this regime is named “inertial breakup”.

   See S-Gamma Breakup Model Properties and S-Gamma Coalescence Model Properties.
7. Set the particle size distribution boundary condition for the inlets and pressure boundaries. As the S-Gamma model assumes a log-normal particle size distribution, set the mean particle size and its variance to define the size distribution:

   In the Physics Conditions of the dispersed phase, set the **Size Distribution Specification** method, and then set the relevant Physics Values. The process is similar to setting the initial conditions of the dispersed phase.

   Setting the variance to zero results in a mono-dispersed phase (single particle size). If the one-equation option is selected, the variance at flow boundaries is ignored, and the initial value that is set for the continuum is used.

   See Boundary Settings.
8. If you want to specify source values for the S-Gamma size distribution equations, in the Phase Conditions of the dispersed phase, activate the **S-Gamma Source Option** physics condition. In the Physics Values, set the appropriate **S-Gamma Source** values.

   See Region Settings.
9. If necessary, change the interaction length scale.

   By default, the Interaction Length Scale for the S-Gamma model is Sauter Mean Diameter. If you need to modify this setting for diagnostics purposes, change the setting on the **Interaction Length Scale** node in the appropriate phase interaction.

Adaptive Multiple Size-Group (AMUSIG) Model Reference

The Adaptive Multiple Size-Group (AMUSIG) model is used for the particle size distribution in a dispersed flow.

Adaptive Multiple Size-Group Model Reference

Theory	See Adaptive Multiple Size-Group (AMUSIG) Model.
Provided By	**[phase]** > **Models** > *Particle Size Distribution*
Example Node Path	**[phase]** > **Models** > **Adaptive Multiple Size-Group**

Requires	An Eulerian Multiphase simulation with the following models activated: *Material*: **Eulerian Multiphase**, **Multiphase Interaction** (Selected automatically) *Eulerian Multiphase Model*: **Multiphase Segregated Flow**, **Multiphase Equation of State** (Selected automatically), **Gradients** (Selected automatically) *Viscous Regime*: **Laminar** or **Turbulent** A minimum of two phases: one continuous phase and one dispersed phase. This model is used only in dispersed phases. In the dispersed phase, the following phase models activated: *Optional Models*: **Particle Size Distribution** *Particle Size Distribution*: **Adaptive Multiple Size-Group** The Adaptive Multiple Size-Group model has some limitations and restrictions that you must consider when setting up your simulation. See Limitations and Restrictions.	
Properties	Key properties are: • **Number Of Size-Groups** • **Velocity Option** See Adaptive Multiple Size-Group Properties.	
Activates	Physics Models	When the dispersed phase in a Continuous-Dispersed Phase Interaction has the **Adaptive Multiple Size-Group** model activated, the following phase interaction models become available. For a laminar viscous regime: • Laminar Breakup • Laminar Coalescence For a turbulent viscous regime: • Turbulent Breakup • Turbulent Coalescence See AMUSIG Phase Interaction Model Family Reference.
	Model Controls (child nodes)	**Number Density Turbulent Prandtl Number** See Model Controls.
	Initial Conditions	For the dispersed phase only: **Particle Size Distribution**. See Initial Conditions.
	Boundary Inputs	For inlets and pressure boundaries: **Particle Size Distribution**. See Boundary Settings.
	Solvers	**Multiple Size-Group** solver. See Multiple Size-Group Solver Properties.
	Field Functions	See Adaptive Multiple Size-Group Model Field Functions.

Adaptive Multiple Size-Group Model Properties

Number Of Size-Groups

One size-group can be sufficient for basic studies, such as the effect of depth on mean bubble size. Breakup and coalescence interactions require at least two size-groups for a meaningful analysis. However, the recommended minimum is three size-groups. Breakup and coalescence simulations are not allowed with one size-group.

The number of groups cannot be changed during iterations or time steps. If you change the number of size-groups, all of the fields for all of the size-groups are automatically reinitialized.

Velocity Option

Specifies whether the particle size-groups travel at a single shared velocity or individual velocities for each group.

- **One-Speed**
  Used where only the mean size of the particle influences the flow.
- **Multi-Speed**
  Allows each size group to move with a different velocity and direction. This option is used to study separation by particle size, or secondary currents that are induced by break-up and coalescence in a poly-disperse flow.

Minimum Particle Diameter

Maximum Particle Diameter

The minimum and maximum values control clipping when the Group Diameter is updated. These limits state what is physically plausible, but normally they are not invoked and do not require adjustment.

Size Distribution Diffusion Limiter

Controls the spurious dissipation rate when treating diffusion under high gradients of particle diameters on a coarse grid. Its value varies from 0 (which gives robustness at the expense of maximum spurious dissipation) to a high value (>> 0.5) (improving accuracy) to ∞ for maximum accuracy. The default value is 0.5. This setting becomes irrelevant on a fine mesh where the gradient is well-resolved.

If there are different results for different values (for example, $\lambda_D = 1.0$ and $\lambda_D = 2.0$), it means that a particle undergoes an intensive breakup/coalescence process when it passes through a single cell. To resolve the equations properly, use a finer mesh.

For more information on spurious dissipation, see [431].

Inter-Group Redistribution Rate

Inter-group redistribution is used to maintain approximately equal volume fractions of each of the different size-groups in each cell. The redistribution algorithm has a strong impact on the performance of the AMUSIG method. If the redistribution rate is too high, the values of the size-group volume fractions, and their accompanying residuals, can become oscillatory.

The default inter-group redistribution rate of 0.1 is sufficient for most of applications, although smaller values can be required in some cases. In the most severe situations, an inter-group redistribution rate of 0.01 is recommended. In such severe cases, use an under-relaxation factor ramp of 100 iterations for the Volume Fraction solver.

In a fully-converged case, the results are independent of the inter-group redistribution rate.

Model Controls

The following child node is available for turbulent viscous regimes only.

Number Density Turbulent Prandtl Number

Sets the ratio of kinematic turbulent viscosity and coefficient of turbulent diffusion of the Number Density.

Initial Conditions

STAR-CCM+ samples the initial conditions once, and then attempts to approximate the cumulative distribution for each method with a few groups. You set the number of groups with the *Number Of Size-Groups* property.

Particle Size Distribution

Sets the initial particle size distribution in the dispersed phase.

Method	Corresponding Initial Conditions Nodes
Uniform distribution A top-hat function, uniform within the specified diameter range and zero outside the range.	**Maximum Diameter** The parameter $d_{\max}$ in Eqn. (2542). **Minimum Diameter** The parameter $d_{\min}$ in Eqn. (2542).
Rosin-Rammler distribution The cumulative distribution function for the Weibull or Rosin-Rammler distribution.	**Scale Parameter** The parameter λ in Eqn. (2545). **Shape Parameter** The parameter k in Eqn. (2545). The larger the value of k the narrower the distribution.
Log-normal distribution The cumulative distribution function for the log-normal distribution.	**Mean Log Diameter** The parameter μ in Eqn. (2547). **Standard Deviation Log Diameter** The parameter σ in Eqn. (2547).

Boundary Settings

STAR-CCM+ resamples the inflow conditions at every iteration, and can contribute to different size-groups depending on the conditions inside the converging flow next to the boundary.

Inlets and Pressure Boundaries

Particle Size Distribution

Sets the particle size distribution at the boundary.

Method	Corresponding Physics Value Nodes
Uniform distribution A top-hat function, uniform within the specified diameter range and zero outside the range.	**Maximum Diameter** The parameter $d_{\max}$ in Eqn. (2542). **Minimum Diameter** The parameter $d_{\min}$ in Eqn. (2542).
Rosin-Rammler distribution The cumulative distribution function for the Weibull or Rosin-Rammler distribution.	**Scale Parameter** The parameter λ in Eqn. (2545). **Shape Parameter** The parameter k in Eqn. (2545). The larger the value of k the narrower the distribution.
Log-normal distribution The cumulative distribution function for the log-normal distribution.	**Mean Log Diameter** The parameter μ in Eqn. (2547). **Standard Deviation Log Diameter** The parameter σ in Eqn. (2547).

Multiple Size-Group Solver Properties

The Multiple Size-Group solver relaxation factors can be varied independently of the Volume Fraction solver relaxation factors.

Coalescence rates increase rapidly with particle concentration. Higher concentrations can require additional relaxation, as well as confirmation that the concentration is being modeled realistically with suitable models for turbulence, for turbulence dispersion, and for breakup.

Implicit Under-Relaxation Factor

The default value of 0.5 gives reliable convergence in One-Speed cases.

Keeping the Implicit Under-Relaxation Factor high (but no more than 0.8) helps to keep down the overall number of iterations.

Explicit Under-Relaxation Factor

The default value of 1.0 gives reliable convergence in One-Speed cases.

Some Multi-Speed cases where each size-group can travel in a different direction can require additional relaxation. For example, the Explicit Under-Relaxation Factor might be reduced to a factor of 0.25 for cases with intense breakup.

Particle Size Frozen

When activated, the Group Diameter is not updated, but the Number Density continues to track changes in Volume Fraction so that these three fields remain physically consistent. If adjusting the relaxation factors does not resolve any convergence difficulty, this option can help you to identify the equation that is responsible for the difficulty.

This option is deactivated by default.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Adaptive Multiple Size-Group Model Field Functions

The naming convention is that the size-groups are indexed in order of increasing particle size. However, this naming convention is not always true in all cells until the calculation is well converged.

GroupDiameter of [size group]

$\sqrt[3]{\frac{6\alpha}{\pi n}}$ where α is the Volume Fraction and n is the Number Density.

Number Density of [size group]

Number Density of [phase]

Interval Volume Fraction of [phase]

The volume fraction of the phase that is present in size-groups with diameter between the specified minimum and maximum particle size.

Mean Diameter of [phase]

The generic mean diameter that is calculated using Eqn. (2536). You can specify the appropriate values of p and q. The default is d_{10} or the number mean.

Surface Mean Diameter of [phase]

The Surface Mean Diameter that is calculated using Eqn. (2537).

Volume Mean Diameter of [phase]

The Volume Mean Diameter that is calculated using Eqn. (2538).

Under the Multi-Speed option, AMUSIG shows a number of flow and turbulence variables that are computed on size-groups. Some examples are:

Velocity of [size-group]

Volume Fraction of [size-group]

Turbulent Viscosity of [size-group]

Turbulent Viscosity Ratio of [size-group]

Effective Viscosity of [size-group]

Ustar of [size-group]

Wall Shear Stress of [size-group]

Wall Y+ of [size-group]

AMUSIG Phase Interaction Model Family Reference

You model breakup and coalescence by activating the appropriate optional models in a phase interaction.

AMUSIG Phase Interaction Model Family Reference

Model Names	**Interaction Length Scale**
	Laminar Breakup
	Laminar Coalescence
	Turbulent Breakup
	Turbulent Coalescence
Theory	See AMUSIG Breakup and Coalescence.
Provided By	**[phase interaction]** > **Models** > *Optional Models*
Example Node Path	**[phase interaction]** > **Models** > **Turbulent Coalescence**
Requires	These model groups are available only for a Continuous-Dispersed Phase Interaction. The dispersed phase must have the **Adaptive Multiple Size-Group** model activated. Laminar groups of coalescence and breakup models require the Laminar viscous regime activated in the Physics continuum. Turbulent groups of coalescence and breakup models require the Turbulent viscous regime activated in the Physics continuum. Breakup and coalescence interactions require at least two size-groups for a meaningful analysis. However, the recommended minimum is three size-groups. The number of size-groups is specified as a property of the **Adaptive Multiple Size-Group** model. Most models also require Surface Tension to be defined under Multiphase Material in the phase interaction.
Properties	Key property for breakup models: **Number of Abscissas**

	See Breakup Model Properties. The coalescence models have no properties.	
Activates	Physics Models	**Laminar Breakup** **Turbulent Breakup** See Breakage Rate Models. **Laminar Coalescence** **Turbulent Coalescence** See Coalescence Rate Models.
	Materials	**Surface Tension** See Material Properties.
	Field Functions	See Field Functions.

Interaction Length Scale Model Properties

Specifies the length scale that is used for interaction area and transfers. Three methods are available:

Adaptive Multiple Size-Group Diameter

The usual and default method for AMUSIG runs. When this option is selected in a multi-speed run, each size-group automatically uses its own group diameter for the drag force and for other interaction models. In a one-speed run, the Sauter or surface-mean diameter is automatically used instead.

Constant **and** *Field Function*

These options apply the same scale to all size-groups. They are of little modeling value in a multi-speed run, but they can be used for passive testing of the AMUSIG model, without using the size distribution prediction in other phase interaction models.

Breakup Model Properties

When the **Laminar Breakup** model or the **Turbulent Breakup** model is selected, the **Multiple Size-Group Breakup** model is selected automatically. This model has the following property:

Number of Abscissas

A discretization parameter that is common to the Fragments Particle Size Distribution submodels that are used in any break up calculation. The number of abscissas is independent from the number of groups. The default value is 3. This value can be increased to match the number of groups, but it does not have a strong effect on results or performance.

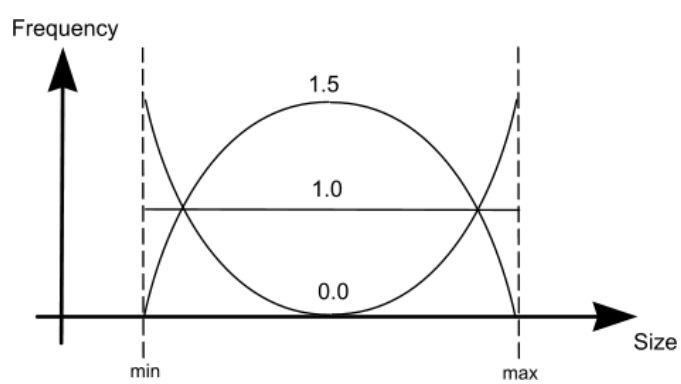
Breakage Rate Models

The breakup process is modeled in two parts, a Breakage Rate and a Fragments Particle Size Distribution. The preferred model combinations are:

- Laminar breakup: **Shear-Induced Breakage Rate** with the **Fragments and Satellites Daughter Particle Size Distribution**.

- Turbulent breakup: the **Martinez-Bazan** or **Tsouris and Tavlarides** models with their corresponding **Fragments Particle Size Distribution** submodel. For emulsion formation (water in oil), the **Coulaloglou and Eskin** model is recommended.

The preferred model combination is activated by default. However, alternative submodel combinations can be used, as described in the following table:

Breakage Rate Model	Available Fragments Particle Size Distribution Submodels
Shear-Induced Breakage Rate Applies to Laminar breakup only. Model properties: *Calibration Constant C1* The default is 0.285. *Calibration Constant C2* The default is 0.62.	**Fragments and Satellites Daughter Particle Size Distribution** This model is based on the physical observations of the breakup process: the mother droplet is extended by the flow to form a dumbbell shape, then the neck is ruptured into small satellites. This model assumes that the mother droplet splits into two large fragment droplets of the same size and a number of small satellites. Model properties: • *Fragments Volume Ratio* The ratio of volume of the fragment to volume of the mother droplet. The satellites have uniform size distribution. The default value is: 0.9. **Parabolic Fragments Particle Size Distribution** An empirical distribution that can be used with any breakup model. Model properties: • *Shape Parameter* The default value is 1.5, which corresponds to the smallest standard deviation of particle sizes. A value of 1.0 provides a uniform distribution, while a value of 0.0 gives the highest possible standard deviation of particle sizes. This distribution is illustrated below: 

Coulaloglou and Eskin Breakage Rate Applies to Turbulent breakup only. This model predicts a broader size distribution than the other models, and is suitable for modeling emulsion formation (water in oil). Model properties: *Calibration Constant Cg* This value is C_g in Eqn. (2527). The default is 1.0. *Critical Weber Number* This value is We_{cr} in Eqn. (2527). The default is 0.5.	**Parabolic Fragments Particle Size Distribution** See the description above.
Martinez-Bazan Breakage Rate Applies to Turbulent breakup only. Model properties: *Calibration Constant Cg* This value is C_g in Eqn. (2527). The default is 0.25. *Critical Weber Number* This value is We_{cr} in Eqn. (2527). The default is 1.0.	**Martinez-Bazan Fragments Particle Size Distribution** This model assumes that the particle size distribution depends on the turbulent stress that is applied to the droplet or bubble ([467]). **Parabolic Fragments Particle Size Distribution** See the description above.
Tsouris and Tavlarides Breakage Rate Applies to Turbulent breakup only. The model properties are the same as for the **Martinez-Bazan Breakage Rate** model.	**Tsouris and Tavlarides Daughter Particle Size Distribution** This model assumes that the energy of the particle size distribution is minimized ([506]). The resulting particle size distribution is bi-modal. Model properties: • *Diameter Ratio* The default value is: 0.2154434621334076. **Parabolic Fragments Particle Size Distribution** See the description above.
Power Law Breakage Rate Power Law Breakage Rate is a generic model with adjustable parameters for the breakage rate multiplier K_B of number density at some particle size d scaled by a characteristic diameter d_0.	**Parabolic Fragments Particle Size Distribution** See the description above.

Model properties:

Characteristic diameter

This parameter is d_0 in Eqn. (2525). The default value is 0.001 m.

Model child nodes:

Breakage Prefactor

This parameter is C in Eqn. (2525). The default is a constant value of 0.0 /s.

Breakage Exponent

This parameter is a in Eqn. (2525). The default is a constant value of 1.0.

Coalescence Models

When the **Turbulent Coalescence** model or the **Laminar Coalescence** model is selected, the **Multiple Size-Group Coalescence** model is selected automatically. This model has no properties.

The coalescence process is modeled in two parts, a Collision Rate and a Coalescence Efficiency. The preferred model combinations are:

- Laminar coalescence: **Laminar Collision Rate** with **Vinckier-Moldenaers Coalescence Efficiency**.
- Turbulent coalescence: **Turbulent Collision Rate** with **Luo Coalescence Efficiency**.

Collision Rate Models

The preferred model combination is activated by default. However, alternative submodel combinations can be used, as described in the following table:

Collision Rate Model	Available Coalescence Efficiency Submodels
Laminar Collision Rate Applies to Laminar coalescence only.	**Film Drainage Coalescence Efficiency** This model is provided for expert users. It is a generic coalescence efficiency model that is compatible with any collision rate model. This model has the following properties: • **Weber Number Exponent** Accounts for surface tension. This value is m in Eqn. (2531). The default value is 1.0. • **Reynolds Number Exponent** Accounts for viscosity. This value is n in Eqn. (2531). The default value is 0.5. You can adjust these property values to fit your own experimental data and model the appropriate physics. This model also has the following child node:

	• **Coalescence Efficiency Prefactor** You can provide the appropriate expression. The default value is 1, meaning that every collision event in the collision rate ends in a successful coalescence of the two particles into one. This value is C in Eqn. (2531). **Vinckier-Moldenaers Coalescence Efficiency** To calculate the coalescence efficiency, this model compares the following times: • the drainage time, that is, the time that is required to squeeze the liquid film between two droplets up to a critical thickness that is controlled by *Hc*. • the contact time, which depends on the shear rate. This model has the following property: • **Calibration Constant Hc** The default value is 1.5 E-8 m. **Uniform Coalescence Efficiency** The coalescence efficiency is identical for all groups – the probability of coalescence does not depend on the sizes of two colliding particles. This option is used mainly for debugging purposes. This model has no properties, but has the following child node: • **Coalescence Efficiency** You can provide the appropriate expression. The default value is 1, meaning that every collision event in the collision rate ends in a successful coalescence of the two particles into one.
Turbulent Collision Rate Applies to Turbulent coalescence only.	**Film Drainage Coalescence Efficiency** See the description above. **Luo Coalescence Efficiency** The contact time due to the turbulent fluctuations is compared to the deformation time of the bubble. This model assumes that a high contact time and a short deformation time (that is, high surface tension) make the coalescence more probable. This model has the following property: • **Calibration Constant C1** The probability of coalescence once two particles have collided. A higher value reduces the probability

	of coalescence. This value is C in Eqn. (2530). The default value is 1. **Uniform Coalescence Efficiency** See the description above.
Enhanced Power Law Collision Rate Allows you to investigate alternative coagulation kernels K between particles of two sizes d_1 and d_2, scaled by characteristic diameter d_0. This model has the following property: **Characteristic Diameter** The characteristic diameter of the power law breakage rate. This parameter is d_0 in Eqn. (2535). The default value is 0.001 m. This model has the following child nodes: • Collision Prefactor The calibration constant C in Eqn. (2535). This value has dimensions of metres-cubed per second. • Collision Exponents A, B, C, and D The exponents a, b, c, and d in Eqn. (2535).	**Film Drainage Coalescence Efficiency** See the description above. **Uniform Coalescence Efficiency** See the description above.
Uniform Collision Rate This model, and the corresponding submodel **Uniform Coalescence Efficiency** are simplified models that allow analytical verification solutions. This model has the following child node: **Collision Rate** The default value is zero because it is arbitrary. This value has units of m^3/s.	**Film Drainage Coalescence Efficiency** See the description above. **Uniform Coalescence Efficiency** See the description above.

Material Properties

Surface Tension

See Surface Tension Material Properties.

Field Functions

Breakup Rate of [phase interaction]

The total number of breakup events per second per unit volume. This field function can be used to highlight the flow features that are associated with breakage.

Coalescence Rate of [phase interaction]

The total number of coalescence events per second per unit volume. This field function can be used to highlight the flow features that are associated with coalescence.

Under Multi-Speed AMUSIG, the phase interaction models show a number of additional field functions, such as:

Interaction Length Scale of [phase interaction_size-group]

Interaction Area Density of [phase interaction_size-group]

Eulerian Drag Coefficient of [phase interaction_size-group]

S-Gamma Model Reference

The S-Gamma model is a phase model for particle size distribution.

If you wish to include the effects of breakup and coalescence on the predicted size distribution, you can select the S-Gamma Breakup and S-Gamma Coalescence models in a multiphase interaction that includes the dispersed phase.

S-Gamma Model Reference

Theory	See S-Gamma Model for Particle Size Distribution.
Provided By	**[phase]** > **Models** > *Particle Size Distribution*
Example Node Path	**[phase]** > **Models** > **S-Gamma**
Requires	*Material*: **Eulerian Multiphase**, **Multiphase Interaction** (Selected automatically) *Eulerian Multiphase Model*: **Multiphase Segregated Flow**, **Multiphase Equation of State** (Selected automatically), **Gradients** (Selected automatically) Create two phases. This model is used only in the dispersed phase. Phase model selection: *Optional Models*: **Particle Size Distribution** *Particle Size Distribution*: **S-Gamma**
Properties	Key properties are: *We Crit* and *Number of S-Gamma Equations*. See S-Gamma Model Properties.

Activates	Physics Models	**S-Gamma Breakup** and **S-Gamma Coalescence** are activated as optional models in a Continuous-Dispersed Phase Interaction when the dispersed phase has the **S-Gamma** model activated. See S-Gamma Phase Interaction Model Reference.
	Model Controls (child nodes)	**S-Gamma Turbulent Prandtl Number** See Model Controls.
	Initial Conditions	**Size Distribution Specification** See Initial Conditions.
	Boundary Inputs	**Size Distribution Specification** See Boundary Settings.
	Region Inputs	**S-Gamma Source Option** See Region Settings.
	Solvers	**S-Gamma solver** See S-Gamma Solver Properties.
	Field Functions	See S-Gamma Model Field Functions.

S-Gamma Model Properties

Minimum Diameter

Sets the minimum allowed Sauter mean diameter.

Maximum Diameter

Sets the maximum allowed Sauter mean diameter.

We Crit

Specifies the critical Weber number We_{cr} for breakup (Eqn. (2481)) and coalescence (Eqn. (2497)).

Number of S-Gamma Equations

Selects the number of extra equations to solve in addition to the volume fraction.

- **One Equation**
  Solves for the transport of interfacial area density (the second moment, S_2 in Eqn. (2507)).
- **Two Equation**
  Solves for both the transport of particle number density and the interfacial area density (zeroth and second moments, S_0 in Eqn. (2502) and S_2 in Eqn. (2507)).

Model Controls

The following child node is available for turbulent viscous regimes only.

S-Gamma Turbulent Prandtl Number

Sets the ratio of kinematic turbulent viscosity and coefficient of turbulent diffusion of the S-Gamma moments. This value is Pr in Eqn. (2511).

Initial Conditions

Size Distribution Specification

Sets the initial particle size distribution in the dispersed phase and controls whether to include particle size variance in the calculations.

Method	Corresponding Initial Conditions Nodes
Sauter Mean Diameter Set the mean diameter only.	**Sauter Mean Diameter** Specifies the Sauter mean diameter profile. (See Eqn. (2499)).
Sauter Mean and Variance of Diameter Set the mean diameter and particle size variance.	**Particle Size Variance** Specifies the particle size variance profile. (See Eqn. (2504)). **Sauter Mean Diameter** Specifies the Sauter mean diameter profile. (See Eqn. (2499)).

Boundary Settings

Inlets and Pressure Boundaries

Size Distribution Specification

Sets the initial particle size distribution in the dispersed phase and controls whether to include particle size variance in the calculations.

Method	Corresponding Physics Value Nodes
Sauter Mean Diameter Set the mean diameter only.	**Sauter Mean Diameter** Specifies the Sauter mean diameter profile. (See Eqn. (2499)).
Sauter Mean and Variance of Diameter Set the mean diameter and particle size variance.	**Particle Size Variance** Specifies the particle size variance profile. (See Eqn. (2504)). **Sauter Mean Diameter** Specifies the Sauter mean diameter profile. (See Eqn. (2499)).

Region Settings

Applies to the dispersed phase in a fluid region.

S-Gamma Source Option

Provides access to the S_0 and S_2 source terms.

Method	Corresponding Physics Value Nodes
S-Gamma Source Term When activated, you can specify source terms as scalar profiles. When deactivated, the source term calculations use their internal defaults.	**S-Gamma S0 Source** S_0 in Eqn. (2469). **S-Gamma S0 Source Derivative** The S_0 derivative in Eqn. (2507). **S-Gamma S2 Source** S_2 in Eqn. (2501) and Eqn. (2503). **S-Gamma S2 Source Derivative** The S_2 derivative in Eqn. (2515).

S-Gamma Solver Properties

The S-Gamma solver controls the solution update for the S-Gamma model.

You are advised to use the same relaxation factor settings for this solver and the Volume Fraction solver. See Setting Up Solvers for Multiphase Flow Cases.

The properties in the *Expert* category are for a temporary debug situation, at the expense of simulation accuracy and higher memory usage. Do not change these properties unless you are thoroughly familiar with the STAR-CCM+ discretization techniques.

Implicit Under-Relaxation Factor

Improves stability and convergence of the linear system by using the relaxation factor to increase diagonal dominance of the matrix.

The default value is 0.15.

Explicit Under-Relaxation Factor

Specifies the multiplier that is applied to the provisional increment of the solution. Sharing the relaxation between implicit and explicit factors can be used to improve performance.

The default value is 1.0.

Overall Relaxation Factor

The product of the two under-relaxation factors (that is, **Implicit Under-Relaxation Factor** times **Explicit Under-Relaxation Factor**). This (read-only) value is a guide to the combined effect of the relaxation.

Interaction Source Storage Retained

When activated, interaction source storage is retained at the end of the iteration. The breakup and coalescence source terms are made available as field functions.

To understand how breakup and coalescence are working in a particular application, you can examine their contributions to the S_0 and S_2 transport equations (Eqn. (2507) and Eqn. (2515) respectively).

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

S-Gamma Model Field Functions

The following field functions are made available to the simulation when the S-Gamma model is used.

ParticleSizeVariance of [phase]

σ in Eqn. (2504).

SauterMeanDiameter of [phase]

d_{32} in Eqn. (2503), Eqn. (2504), and Eqn. (2505).

SecondSizeDistributionMoment of [phase]

S_2 in Eqn. (2501), Eqn. (2505), and Eqn. (2515).

ThreeZeroDiameter of [phase]

d_{30} in Eqn. (2503), Eqn. (2504), and Eqn. (2505).

ZerothSizeDistributionMoment of [phase]

S_0 in Eqn. (2469), Eqn. (2468), Eqn. (2505), and Eqn. (2507).

S-Gamma Phase Interaction Model Reference

The S-Gamma Breakup and S-Gamma Coalescence models account for the effects of breakup and coalescence on the predicted particle size distribution in a multiphase Continuous-Dispersed phase interaction.

S-Gamma Phase Interaction Model Reference

Model Names	**S-Gamma Breakup**	
	S-Gamma Coalescence	
Theory	See S-Gamma Breakup and Coalescence.	
Provided By	**[phase interaction]** > **Models** > *Optional Models*	
Example Node Path	**[phase interaction]** > **Models** > **S-Gamma Coalescence**	
Requires	These models are available only for a Continuous-Dispersed Phase Interaction. The dispersed phase must have the **S-Gamma** model activated. See S-Gamma Model Reference.	
Properties	Key properties are: • Breakup Factor See S-Gamma Breakup Model Properties. • Coalescence Factor See S-Gamma Coalescence Model Properties.	
Activates	Field Functions	See S-Gamma Breakup Field Functions and S-Gamma Coalescence Field Functions.

S-Gamma Breakup Model Properties

Breakup Factor

A multiplier for breakup terms.

Viscous Breakup Properties

Lets you control the specific characteristics of the viscous breakup process.

Viscous Breakup Timescale

Sets how the breakup timescale is calculated.

Properties:

- **Standard**
  Uses the default timescale from Eqn. (2480).
- **User Defined**
  Activates the **User-Defined Viscous Breakup Timescale** method node, which has the following property:

- **Diameter Exponent**
  Sets the exponent a for the particle diameter in the timescale expression (Eqn. (2476)). The default setting is 1.0.

Child node:

- **Viscous Breakup Timescale**
  Sets the multiplier τ' in the timescale expression (Eqn. (2476)).

Viscous Breakup Fragments

Sets the number of fragments that form during breakup.

Properties:

- **Standard**
  Uses the default fragmentation number in Eqn. (2479).
- **User Defined**
  Activates the **User-Defined Viscous Breakup Fragments** method node, which has the following property:
  - **Diameter Exponent**
    Sets the exponent b for the particle diameter in the fragment number expression (Eqn. (2475)). The default setting is 3.0.

  Child node:
  - **Viscous Breakup Fragments**
    Sets the multiplier N' for the number of fragments in Eqn. (2475).

Viscous Critical Diameter

Sets how the critical diameter is calculated.

- **Standard**
  Uses the default critical diameter from Eqn. (2478).
- **User Defined**
  Activates the **User-Defined Viscous Critical Diameter** method node, which has the following child node:
  - **Viscous Breakup Dcrit**
    Sets the viscous critical diameter, d_{cr} in Eqn. (2478).

Inertial Breakup Properties

Lets you control the specific characteristics of the inertial breakup process.

For setting user-defined values for the breakup timescale and fragments, use the function of the form $N(d) = \mathrm{N}'d^b$. You set N' as a constant or field function and set b as a constant in the *Diameter Exponent* property (see Eqn. (2475)).

Concentration Correction Factor

This value is C_α in Eqn. (2481). This value is used in the calculation of the critical droplet diameter, since the presence of nearby droplets dampens the disruptive power of the inertial forces.

Inertial Breakup Timescale Coefficient

Time coefficient k_{br} in Eqn. (2482).

Inertial Breakup Timescale

Sets how the breakup timescale is calculated.

- **Standard**
  Uses the default timescale from Eqn. (2482).
- **User Defined**
  Activates the **User-Defined Inertial Breakup Timescale** method node, which has the following property:
  - **Diameter Exponent**
    Sets the exponent b for the particle diameter in the timescale expression (Eqn. (2476)). The default setting is 1.5.

  Tthe **User-Defined Inertial Breakup Timescale** method node also has the following child node:
  - **Inertial Breakup Timescale**
    Sets the multiplier τ' in the timescale expression (Eqn. (2476)).

Inertial Breakup Fragments

Sets how the breakup fragmentation is calculated.

- **Standard**
  Uses the default fragmentation number in Eqn. (2479).
- **User Defined**
  Activates the **User-Defined Inertial Breakup Fragments** method node, which has the following property:
  - **Diameter Exponent**
    Sets the exponent b for the particle diameter in the fragment number expression (Eqn. (2475)). The default setting is 0.0.

  The **User-Defined Inertial Breakup Fragments** method node also has the following child node:
  - **Inertial Breakup Fragments**
    Sets the multiplier N' for the number of fragments in Eqn. (2475).

Inertial Critical Diameter

Sets how the critical diameter is calculated.

- **Standard**
  Uses the default critical diameter from Eqn. (2481).
- **User Defined**
  Activates the **User-Defined Inertial Critical Diameter** method node, which has the following child node:
  - **Inertial Breakup Dcrit**
    Sets the inertial critical diameter, d_{cr} in Eqn. (2481).

    You can specify a constant or field function.

S-Gamma Coalescence Model Properties

Coalescence Factor

The calibration coefficient, F_{cl} in Eqn. (2484).

Hamaker Constant

The value A_H in Eqn. (2493), which is used in the calculation of critical film thickness.

Viscous Coalescence Properties

Lets you control the specific characteristics of the viscous coalescence process.

Viscous Collision Rate

Sets how the collision rate is calculated.

- **Standard**
  Calculates the collision rate using Eqn. (2487).
- **User Defined**
  Activates the **User-Defined Viscous Coalescence Collision Rate** method node, which has the following child node:
  - **Viscous Coalescence Collision Rate**
    Sets the value for K_{coll} in Eqn. (2487).

Viscous Coalescence Probability

Sets the mode of calculating the probability of a collision resulting in coalescence. The choice of this mode can affect the particle size, since increased mobility enhances coalescence, possibly leading to predictions of excessively large particle sizes (typically for bubbles or drops).

- **Fully Immobile Interface**
  Coalescence is blocked by contaminants in the fluid (Eqn. (2489)).
- **Partially Mobile (Short Collision Time)**
  Coalescence is partially blocked by contaminants in the fluid (Eqn. (2490)).
- **Partially Mobile (Quasi-Steady Flow in Film)**
  Coalescence takes place faster due to less blockage (Eqn. (2491)).
- **Fully Mobile Interface**
  Coalescence takes place as quickly as possible – no material is blocking it (Eqn. (2492)).
- **User Defined**
  Activates the User-Defined Viscous Coalescence Probability method node, where you specify a value for the coalescence probability, P_{cl} in Eqn. (2488).

Inertial Coalescence Properties

Lets you control the specific characteristics of the inertial coalescence process.

Probability Factor

This value is $k_{cl,2}$ in Eqn. (2495), and is used in the calculations for inertial collision probability.

Inertial Collision Rate

Sets how the collision rate is calculated.

- **Standard**
  Uses the default collision rate calculation from Eqn. (2494).
- **User Defined**
  Activates the **User-Defined Inertial Coalescence Collision Rate** method node, which has the following child node:
  - **Inertial Coalescence Collision Rate**
    Sets the value for the inertial coalescence collision rate, $K_{coll,i}$ in Eqn. (2494).

Inertial Coalescence Probability

Sets the method of calculating the probability of a collision resulting in coalescence.

- **Standard**
  Uses the default coalescence probability calculation from Eqn. (2495).
- **User Defined**
  Activates the **User-Defined Inertial Coalescence Probability** method node, which has the following child node:
  - **Inertial Coalescence Probability**
    Sets the value for the inertial coalescence probability, $P\left(d_{eq}\right)$ in Eqn. (2495).

S-Gamma Breakup Field Functions

The following field function is made available to the simulation when the S-Gamma Breakup model is used:

Inertial Breakup Dcrit of [phase interaction]

The critical diameter d_{cr} in Eqn. (2481).

The following field functions are made available to the simulation when the S-Gamma Breakup model is used and the **Interaction Source Storage Retained** property is activated in the S-Gamma solver:

S0 Inertial Breakup Source of [phase interaction]

The inertial contribution to s_{br} in the S_0 transport equation (Eqn. (2507)).

S2 Inertial Breakup Source of [phase interaction]

The inertial contribution to $s_{a,br}$ in the S_2 transport equation (Eqn. (2515)).

S0 Viscous Breakup Source of [phase interaction]

The viscous contribution to s_{br} in the S_0 transport equation (Eqn. (2507)).

S2 Viscous Breakup Source of [phase interaction]

The viscous contribution to $s_{a,br}$ in the S_2 transport equation (Eqn. (2515)).

S-Gamma Coalescence Field Functions

The following field functions are made available to the simulation when the S-Gamma Coalescence model is used:

Inertial Coalescence Collision Rate of [phase interaction]

The inertial collision rate $K_{coll,i}$ in Eqn. (2494).

Inertial Coalescence Probability of [phase interaction]

The inertial probability $P\left(d_{eq}\right)$ in Eqn. (2495).

Viscous Coalescence Collision Rate of [phase interaction]

The viscous collision rate $K_{coll,v}$ in Eqn. (2487).

Viscous Coalescence Probability of [phase interaction]

The viscous probability $P_v\left(d_{eq}\right)$ in Eqn. (2488).

The following field functions are made available to the simulation when the S-Gamma Coalescence model is used and the **Interaction Source Storage Retained** property is activated in the S-Gamma solver:

S0 Inertial Coalescence Source of [phase interaction]

The inertial contribution to s_{cl} in the S_0 transport equation (Eqn. (2507)).

S2 Inertial Coalescence Source of [phase interaction]

The inertial contribution to $s_{a,cl}$ in the S_2 transport equation (Eqn. (2515)).

S0 Viscous Coalescence Source of [phase interaction]

The viscous contribution to s_{cl} in the S_0 transport equation (Eqn. (2507)).

S2 Viscous Coalescence Source of [phase interaction]

The viscous contribution to $s_{a,cl}$ in the S_2 transport equation (Eqn. (2515)).

Modeling Boundary Permeability

You can define a boundary as being permeable or impermeable for a particular phase.

This setting can be applied to the following boundary types:

- Outlet

  You can specify that for a particular phase this boundary behaves like a wall (it is Impermeable).
- Wall

  You can specify that for a particular phase this boundary behaves like an outlet (it is Permeable). You can also set this boundary to be semi-permeable by specifying the proportion of the phase that is removed at the boundary.

These options are specified in the properties of the phases under the **Phase Conditions** node for the boundary.

An example where boundary permeability settings would be advantageous is a case modeling a bubble column containing air and water. In this case, you could specify a degassing boundary at the outlet, which would allow air to exit the fluid domain but not water. This would be achieved by selecting the *Phase Impermeable* option for the water phase at the outlet boundary. With such an approach, the mass of water within the column is conserved, while allowing the air bubbles to escape. The code ensures that the mass flow of air at the inlet and outlet of the domain is balanced. The volume fractions of each phase throughout the domain are then computed according to continuity.

An example involving a permeable wall phase condition is a case that involves droplet deposition on a wall. The permeable wall approach assumes that when droplets impinge on a surface they stick there and hence are removed from the flow field. The permeable wall phase condition says that the phase leaves the flow domain at the rate that it approaches the surface. It is also assumed that the deposition layer is sufficiently thin so as not to affect the flow field. Where not all the droplets stick to the wall, the wall permeability factor could be adjusted to limit the mass flux that is removed at the wall.

Phase Condition Properties

Type	The phase-specific boundary option.	
	Outlet	Normal outlet behavior (default option for outlets).
	Phase Impermeable	Specifies that the outlet behaves like a wall for the selected phase. This option is available on outlets.
	Phase Permeable	Specifies that the wall behaves like an outlet for the selected phase. This option is available on wall boundaries. Also allows you to define a semi-permeable wall using the Wall Permeability setting.
	Wall	Normal wall behavior (default option for wall boundaries).

Contents:

Defining an Impermeable Outlet
Defining a Permeable Wall

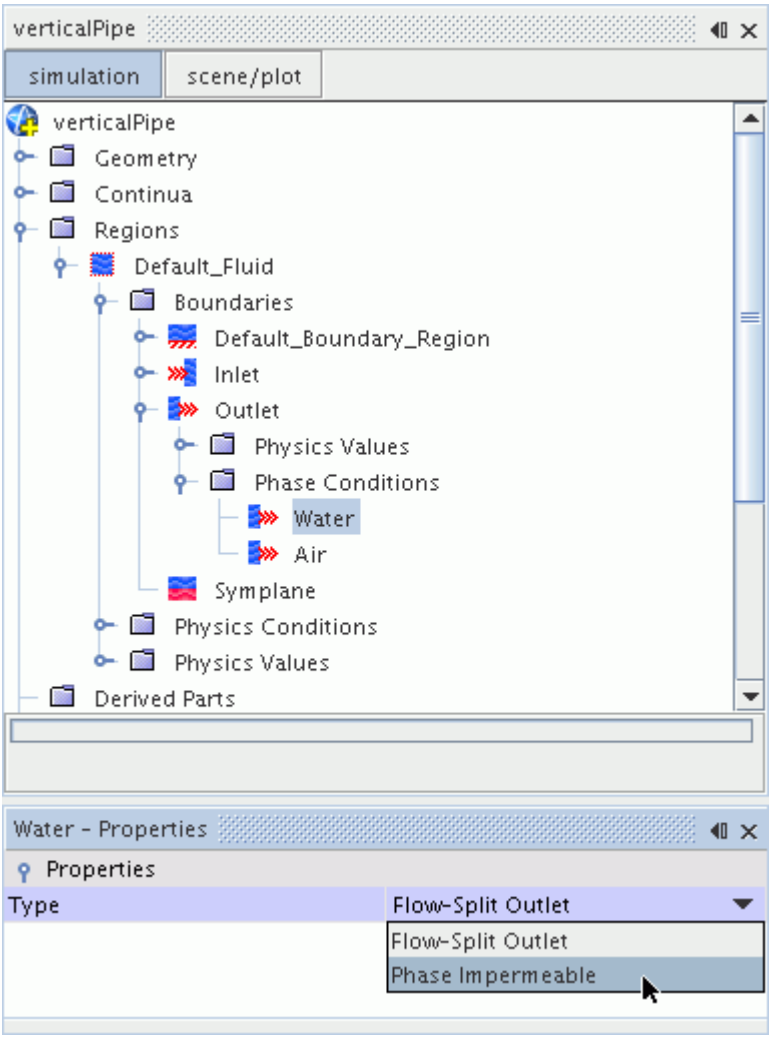
Defining an Impermeable Outlet

When an outlet is defined as impermeable for a particular phase, it behaves like a wall for that phase.

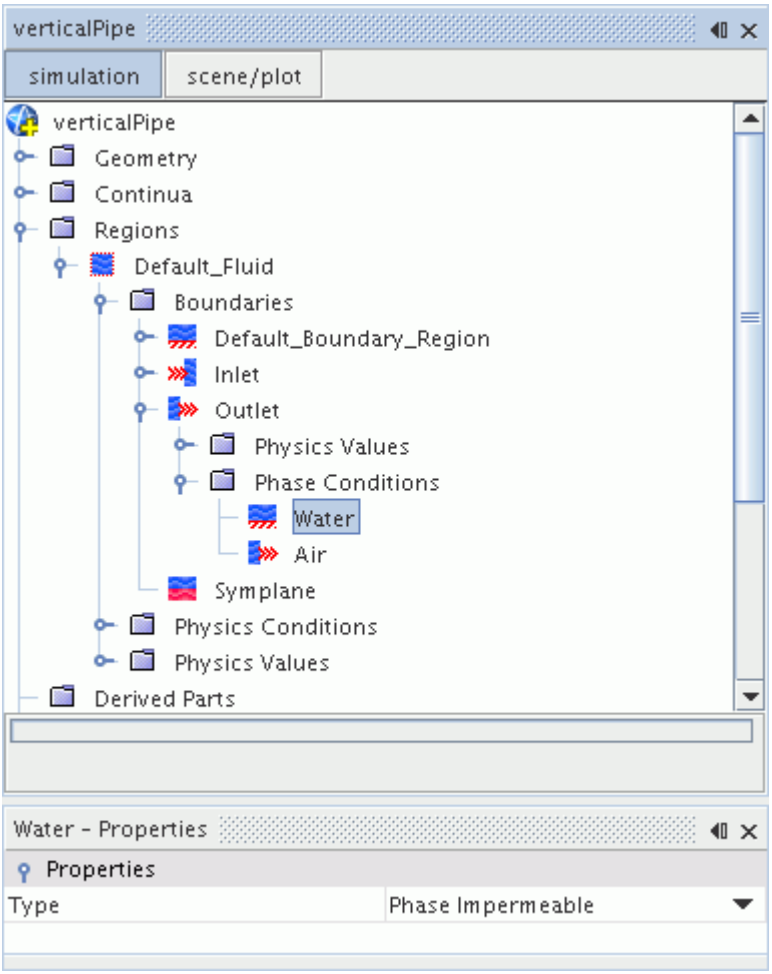
To define an impermeable outlet phase condition:

1. Set up your physics continuum, phases, regions, and boundaries.
   Make sure that the outlet boundary type is set to Outlet.

2. Expand the **Phase Conditions** node for the outlet boundary and select the node for the phase that is associated with the Impermeable condition.
3. In the *Properties* window, change the *Type* to **Phase Impermeable**.



The icon for the phase changes from to , indicating that the outlet behaves like a wall for that phase.



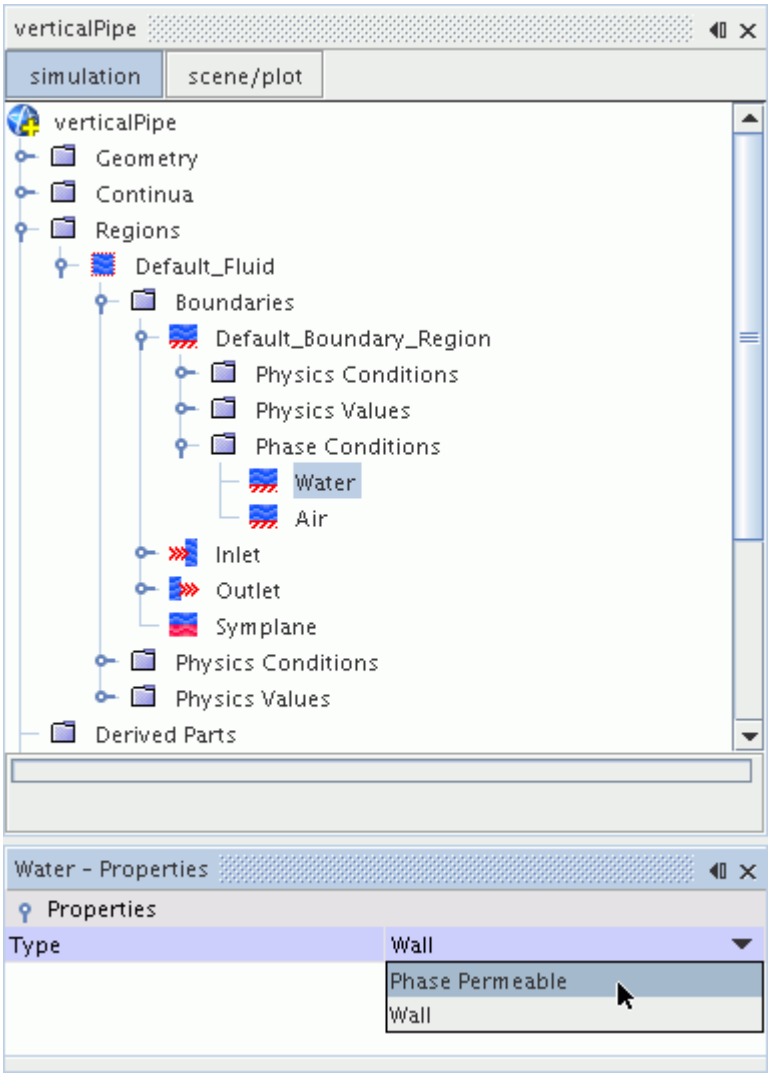
Defining a Permeable Wall

The permeable wall phase condition allows the selected phase to permeate though the wall, thus the wall behaves like an outlet for that phase. Other phases in the domain still treat the boundary as a wall.

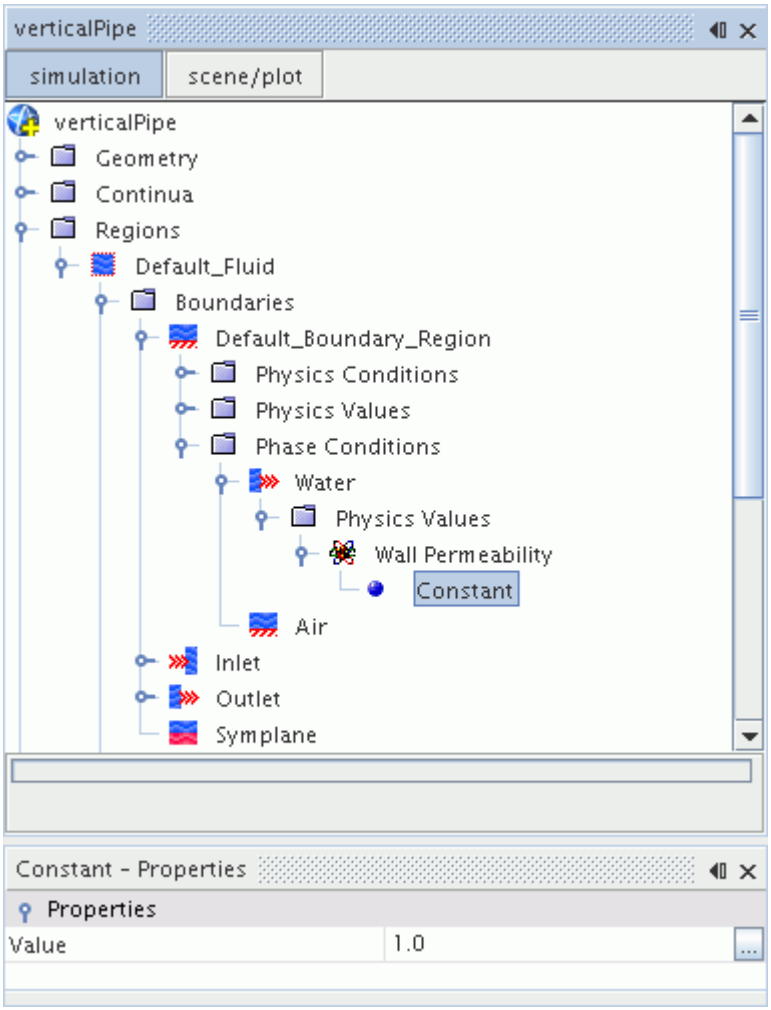
This phase condition can be used to model the deposition of a phase on a surface, or to model a gas leaving a liquid at a free surface (degassing), for example. At a phase permeable wall, any component of phase velocity parallel to the wall is treated according to the wall boundary conditions. For example, zero for a non-slip wall in a stationary geometry case.

You can define the proportion of mass flux that is removed from the phase at the permeable wall by defining the Wall Permeability factor. When the Wall Permeability factor is set to 1.0, the wall behaves as a pure outlet for that phase. Setting the Wall Permeability to a value from 0 through 1 results in a semi-permeable wall.

To define a permeable wall phase condition, first set up your physics continuum, phases, regions, and boundaries. Next, expand the **Phase Conditions** node for the wall boundary. Select the node for the phase that is associated with the Permeable condition. In the *Properties* window, change the *Type* to **Phase Permeable**.



When a phase is associated with a Permeable wall condition, a **Physics Values** > **Wall Permeability** node becomes available for the phase. You can adjust the proportion of mass flux that is removed from the phase at the wall boundary by entering a value for the **Wall Permeability** factor. A value of 1.0 is a pure outlet, and a value of 0.0 is a wall.

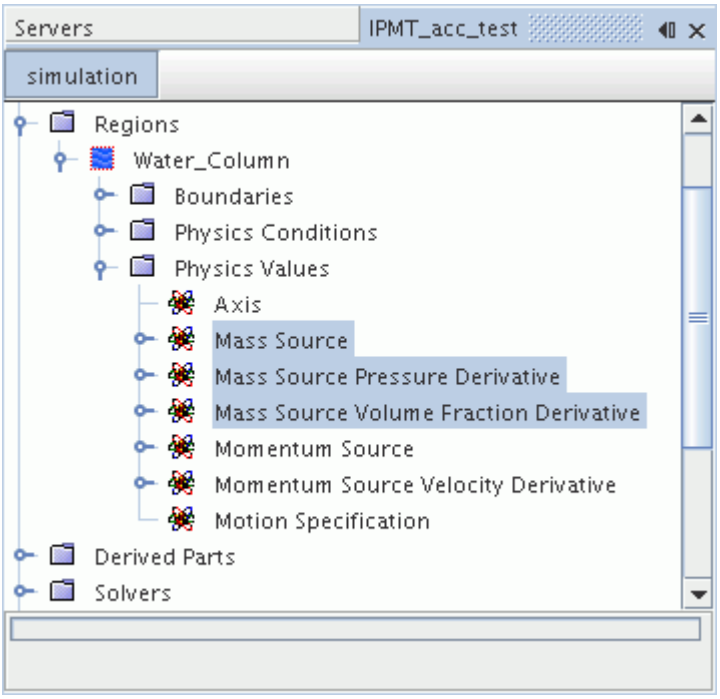


Using Mass Source Terms in Eulerian Multiphase Flows

Mass source terms can be used for Eulerian multiphase flows.

To activate mass source terms in a region select the **Mass Source Option** in the region Physics Conditions, and tick the checkbox in the *Properties* window. This reveals three nodes in the Physics Values for the region:

- Mass Source
- Mass Source Pressure Derivative
- Mass Source Volume Fraction Derivative



Mass Source for Eulerian Multiphase Flows

This node is added to the region physics values when the Mass Source Option region condition is activated. The values for the mass source can be entered as constant or composite values. The dimensions of the mass source term are mass per volume per time.

When you create a mass source for a particular phase, create source terms for other transport equations. The source terms describe the inflow or outflow of other variables that are carried on the mass source, for example, momentum, turbulence and energy.

Mass Source Pressure Derivative for Eulerian Multiphase Flows

This node represents linearization of the phase Mass Source with respect to pressure. The values for the mass source pressure derivative can be entered as constant or composite values. The derivative of the source term is used to linearize the overall volume correction equation. The default value is zero. Providing a value for the derivative helps to stabilize the solution.

Mass Source Volume Fraction Derivative for Eulerian Multiphase Flows

This node represents linearization of the phase Mass Source with respect to volume fraction. The values for the mass source volume fraction derivative can be entered as constant or composite values. The derivative of the source term is used to linearize the phase mass balance equations. The default value is zero. Providing a value for the derivative helps to stabilize the solution.

Contents:

Defining Mass Source Terms Using a Constant or Composite Method

Defining Mass Source Terms Using a Constant or Composite Method

In Eulerian multiphase simulations, two techniques are available for defining mass source terms.

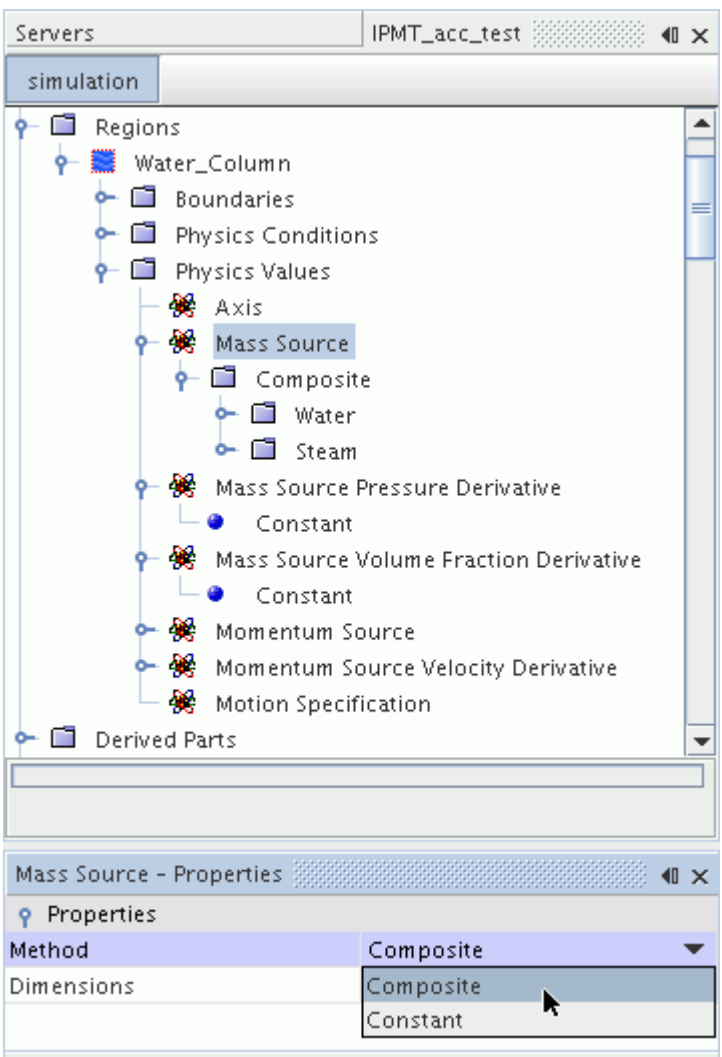
The two techniques for defining mass source terms are:

- Constant
- Composite

The constant method allows you to define constant values for all phases within a single property (set using an array profile). The composite method provides separate nodes for each phase, allowing you to define the source terms using different techniques. For example, the composite method allows you to define the source term for each phase using a different field function.

To select either the constant or composite method, first activate the mass source option for the corresponding region. Next, select the **Mass Source**, **Mass Source Pressure Derivative**, or **Mass Source Volume Fraction** physics value node. In the *Properties* window, set the *Method* property to the desired option.

In the screenshot below, the composite method is selected for the **Mass Source**, and the constant method has been selected for the two mass source derivatives.



Defining Constant Values for All Phases

The Constant method allows you to specify constant mass source values for all phases within a single property. It is also possible to define constant values using the composite technique, but it is necessary to set these values under separate nodes in the simulation tree.

To define values using the constant method, make sure that the method for the corresponding mass source node is set to **Constant**. Select the **Constant** node and enter values in the profile array.

To check which array value corresponds to which phase, click ... for the *Value* property. This opens the *Constant - Value* dialog, which lists each phase separately.

Defining Values Using a Composite Method

The Composite method allows you to specify mass source values for each phases independently. It provides a separate node for each phase, which can be used to enter values using field functions, for example.

To define values using the constant method, make sure that the method for the corresponding mass source node is set to **Composite**. Expand the nodes for the appropriate phase, and define the value or field function in the usual way.

Dispersed Multiphase Model

The Dispersed Multiphase (DMP) model simulates dispersed phases in a Eulerian manner. The Dispersed Multiphase model combines aspects of both the Lagrangian Multiphase (LMP) model and the Segregated (Eulerian) Multiphase (EMP) models.

- Like the EMP model, Dispersed Multiphase solves for the dispersed phases in a Eulerian manner.
- When used in the default one-way coupled mode, Dispersed Multiphase does not calculate the flow in the background medium (that is, the Physics Continuum). This behavior is like the LMP model.

The main difference between DMP and EMP is that the Eulerian phases in the DMP model represent only the phases that are dispersed. The continuous background phase is solved with typical single-phase models.

By default, DMP is one-way coupled: the background flow can influence the dispersed phases (for example, through drag force and heat transfer) but not the reverse. The existence of the dispersed phases has no effect on the background phase. The only way in which the dispersed phases can influence the physics continuum is through impingement onto a liquid film. Two-way coupling is optional in a DMP simulation. When two-way coupling is activated, the dispersed phases can influence the physics continuum. Drag force and heat transfer contribute to all phases, and the volume of the dispersed phases is subtracted from the available volume for the background phase.

The DMP model and the VOF model can be activated in the same simulation. The DMP model uses the VOF mixture as its background flow, and has no knowledge of the individual phases. The DMP model treats the mixture as a single phase and uses only the properties of the mixture.

Advantages of Using the DMP Model

The DMP model is useful for airfoil icing applications, for which it is more appropriate than either the EMP model or the LMP model. Using the LMP model for these applications would be time-consuming due to the large number of parcels that are required. The DMP model can also be useful in soiling applications.

For an example of an airfoil icing simulation, see the *Dispersed Multiphase: Airfoil Icing* tutorial in the *Tutorial Guide*.

The EMP model does not support impingement onto a liquid film, or a coupled solution of background flow and energy, which are often required in these applications.

Disadvantages of Using the DMP Model

The DMP model cannot use many of the complex physics models that are available with the EMP model or the LMP model, such as turbulent dispersion. The DMP model cannot model evaporation or other interactions with the background fluid.

The DMP model does not model interactions between droplets of the dispersed phase, or complicated wall interactions (such as splashing). The LMP framework provides models for these effects.

Contents:

Dispersed Multiphase Workflow
Dispersed Multiphase Model Reference

Dispersed Multiphase Solvers
Dispersed Multiphase Formulation

Dispersed Multiphase Workflow

Dispersed Multiphase (DMP) is an alternative to the Lagrangian Multiphase (LMP) model and the Segregated (Eulerian) Multiphase (EMP) models, rather than a replacement. Due to its superposing behavior, DMP is compatible with many existing single-phase models. The dispersed phase particles can exist in a background fluid that is a single- or multi-component gas or liquid. The particles themselves can be single-component gases or liquids.

The Dispersed Multiphase model can be activated at the end of a simulation without the need to deactivate and reactivate the appropriate set of models. As DMP does not solve for the background flow, it is automatically compatible with both segregated and coupled solution of the background flow and energy.

DMP can be used as a post-processing tool: you can clear the Dispersed Multiphase solution, leaving the background flow intact, and then recalculate the transport of the dispersed phases only. You can obtain a steady solution even when the entire simulation is Unsteady.

This feature can help you to perform a quick analysis of particle behavior, such as collection efficiencies. For example, if you are using a set of dispersed phases to model a range of particle diameters, you can repeat the calculation for each dispersed phase and sum the results.

To set up a simulation that uses the Dispersed Multiphase model, perform the following tasks:

1. Select the Dispersed Multiphase model.

   See Dispersed Multiphase Model Reference.
2. Define the dispersed phases.

   Each dispersed phase is a Eulerian phase for which you specify the relevant models, such as a flow model, a heat transport model, and a material model. You can also specify whether each phase is two-way coupled with the background phase. You can use a number of dispersed phases to model particles of different materials, or particles that are of a single material but of different sizes. Each dispersed phase is independent: the phases do not interact.

   See Setting Up the Dispersed Phases.
3. Set up the phase interactions.

   You can set up interactions between the dispersed phase and the background (continuous phase), between the dispersed phase and a fluid film, and between the dispersed phase and a VOF phase.

   See Setting Up Phase Interactions.
4. Set boundary and region conditions.

   The boundary conditions are typically set as phase conditions of the dispersed phase. The available methods to specify the velocity, volume fraction, and temperature depend on the type of boundary.

   See Dispersed Multiphase Model Reference.
5. Set the Dispersed Multiphase solver parameters.

   The Dispersed Multiphase solvers generally use the same techniques as the Segregated Flow solvers.

   See Dispersed Multiphase Solvers.

Contents:

Setting Up the Dispersed Phases
Setting Up Phase Interactions

Setting Up the Dispersed Phases

The **Dispersed Multiphase** model node contains a Eulerian Phase manager which allows you to define a number of dispersed phases.

Each dispersed phase is a Eulerian phase, similar to the phases that are used in the Segregated Multiphase model. For each phase, you specify the relevant models such as a flow model, a heat transport model, and a material model.

By default, each dispersed phase is one-way coupled: the influence of the dispersed phase on the continuous background phase is assumed to be insignificant and is neglected. However, if you expect a dispersed phase to have a significant effect on the continuous phase, you can activate two-way coupling. When two-way coupling is activated for a dispersed phase, the total momentum and energy of the dispersed phase - continuous background phase system are conserved. Heat transfer and drag force (velocity) are the two models that are currently included in the phase interaction. For example, if you have one very dilute dispersed phase and another dispersed phase that has a more significant concentration, you can model the interaction of the first phase with the continuous phase as one-way coupled and model the interaction of the second dispersed phase as two-way coupled.

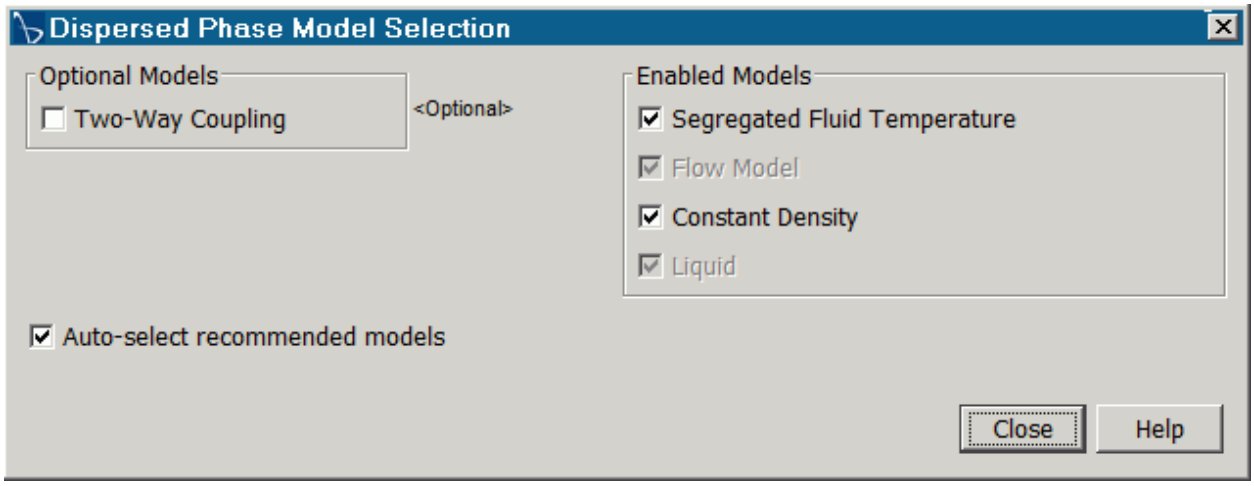
You can use a number of dispersed phases to model particles of different materials, or particles that are of a single material but of different sizes. Each dispersed phase is independent: the phases do not interact.

To specify a dispersed phase:

1. Under the **Dispersed Multiphase** > **Eulerian Phases** node, create a phase.
2. In the new **Phase** node, open the *Phase Model Selection* dialog and select the appropriate material model: Gas or Liquid.
3. Select the appropriate optional models.

   If the background fluid has an energy model, the dispersed phase also requires one. The only energy model option currently available is **Segregated Fluid Temperature**.

   If you expect the dispersed phase to have a significant effect on the continuous phase, activate **Two-Way Coupling**.



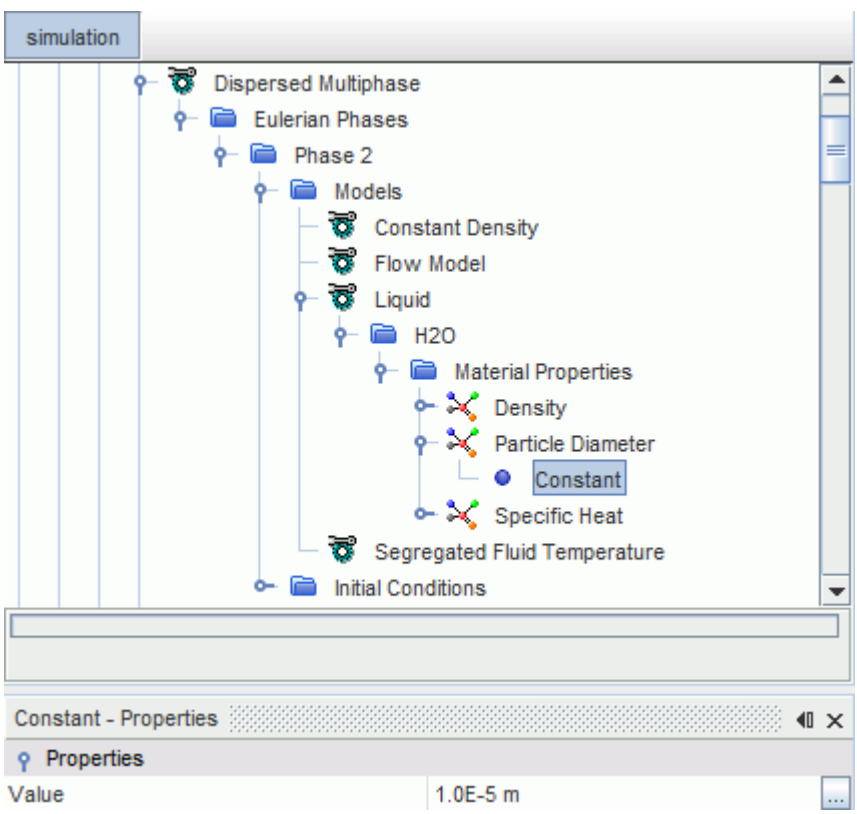
4. Click **Close**.

   The selected models are added under the **Phase** node.

The particle diameter is a constant for each phase and is specified as a material property.

To specify the particle diameter for a dispersed phase:

5. Under the **Liquid** or **Gas** node (the phase material model), select the **Material Properties** > **Particle Diameter** > **Constant** node.
6. In the *Properties* window, set the *Value* to the appropriate value.



Setting Up Phase Interactions

The interactions between the dispersed phase and the background (continuous phase), between the dispersed phase and a VOF phase, and between the dispersed phase and a fluid film, are set up as phase interactions. You set up these phase interactions under the Multiphase Interaction node.

In a typical scenario, with one-way coupling, the continuous phase interacts with the dispersed phase through drag force and heat transfer. The dispersed phase has no influence on the continuous phase. However, when two-way coupling is activated, heat is transferred between the dispersed phase and the continuous phase with overall heat conservation. Likewise, the drag force applies in both directions.

The dispersed phase interacts with the fluid film phase by impingement of droplets from the dispersed phase into the fluid film and stripping of droplets from the fluid film into the dispersed phase.

Dispersed Phase - Continuous Phase Interaction

The continuous phase can be a VOF phase or the physics continuum. If the dispersed phase interacts with two or more VOF phases, you specify a separate DMP-VOF phase interaction for each VOF phase.

To specify the interaction between the dispersed phase and a continuous phase:

1. Under the **Multiphase Interaction** > **Phase Interactions** node, create a phase interaction.
2. In the new **Phase Interaction** > **Models** node, open the *Phase Interaction Model Selection* dialog and activate the appropriate model:
   - **DMP-Physics Continuum Interaction**

     Uses the mixture properties (such as density, dynamic viscosity, and specific heat) when calculating source contributions. Available for all material models except **Volume of Fluid (VOF)**.
   - **DMP-VOF Phase Interaction**

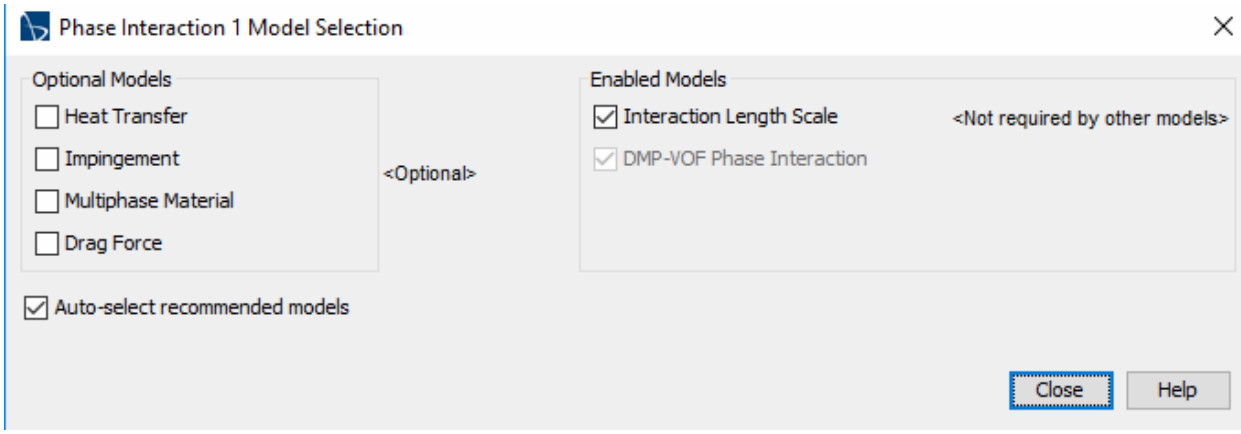
     Uses the phase properties when calculating source contributions for each VOF phase. The phase source contributions (such as drag and heat transfer) are weighted by the corresponding VOF volume fraction. Available for the **Volume of Fluid (VOF)** material model only.
3. Click **Close**.

   The selected phase interaction model is added under the **Phase Interaction** > **Models** node.
4. Select the phase interaction node (for example, the **DMP-VOF Phase Interaction** node) and specify the appropriate phases:
   - **Dispersed Phase**
   - **Continuous Phase**

     Available in a DMP-Physics Continuum Interaction.
   - **VOF Phase**

     Available in a DMP-VOF Phase Interaction.
5. Select the appropriate optional models.

   The example below shows the **DMP-VOF Phase Interaction** model. The optional models for the **DMP-Physics Continuum Interaction** model are the same, except that the Impingement model is not applicable. The optional models **Heat Transfer** and **Drag Force** are available only when the **Interaction Length Scale** model is selected.



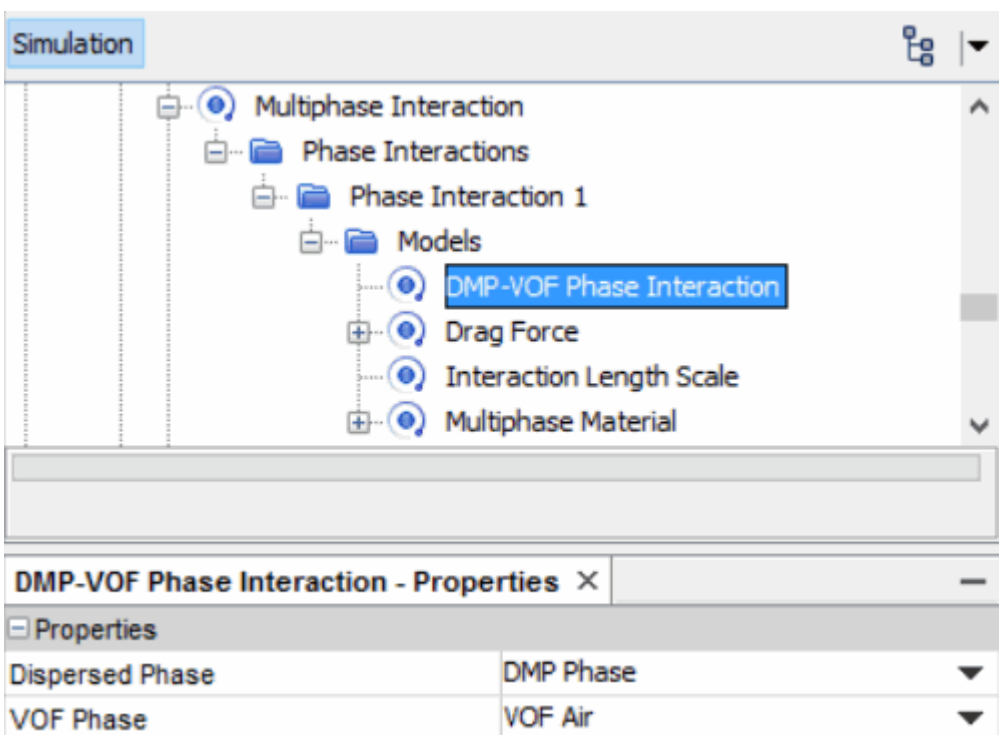
The available optional models are:

Heat Transfer	See Modeling Heat Transfer.
Impingement	See DMP-VOF Impingement.
Multiphase Material	See Providing Additional Material Properties.
Drag Force	See Modeling Drag Force.

6. Click **Close**.

   The selected models are added under the **Phase Interaction** > **Models** node.

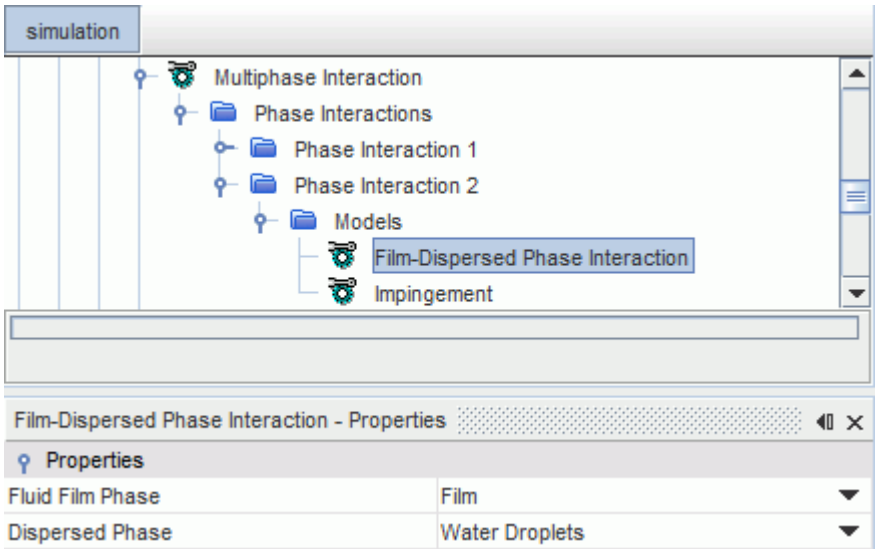
   The example below is for the **DMP-VOF Phase Interaction** model.



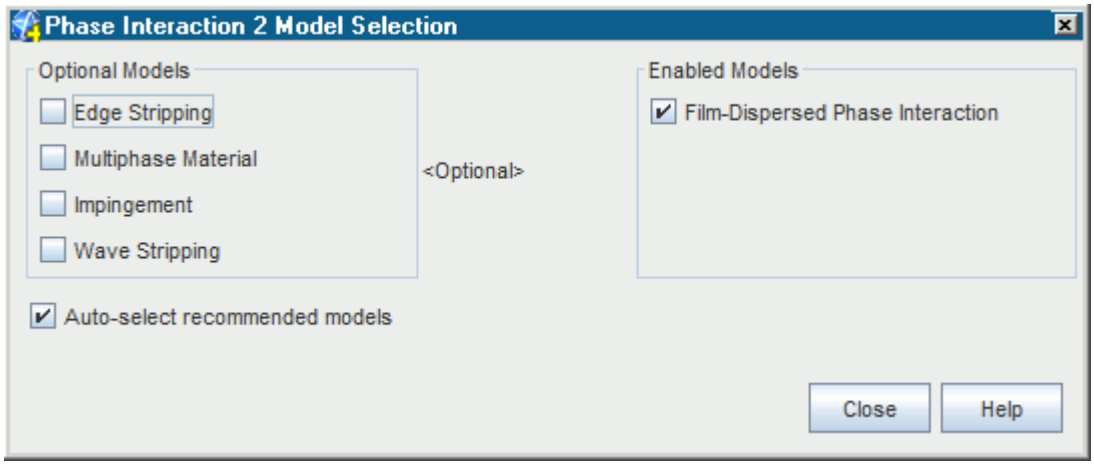
Dispersed Phase - Fluid Film Phase Interaction

To specify the interaction between the dispersed phase and the fluid film phase:

7. Under the **Multiphase Interaction** > **Phase Interactions** node, create a phase interaction.
8. In the new **Phase Interaction** > **Models** node, open the *Phase Interaction Model Selection* dialog and activate the **Film-Dispersed Phase Interaction** model.
9. Click **Close**.
10. Select the **Film-Dispersed Phase Interaction** node and set the *Fluid Film Phase* and the *Dispersed Phase* properties to specify the appropriate phases.



11. Reopen the *Phase Interaction Model Selection* dialog and select the appropriate optional models.



The available optional models are:

Edge Stripping	See Edge Stripping Model Reference.
Multiphase Material	See Providing Additional Material Properties.

Impingement	See Impingement Model Reference.
Wave Stripping	See Wave Stripping Model Reference.

12. Click **Close**.

    The selected models are added under the **Phase Interaction** > **Models** node.

Dispersed Multiphase Model Reference

The Dispersed Multiphase (DMP) model simulates dispersed phases in a Eulerian manner. The Dispersed Multiphase model combines aspects of both the Lagrangian Multiphase (LMP) model and the Segregated (Eulerian) Multiphase (EMP) models.

Dispersed Multiphase Model Reference

Theory	See Dispersed Multiphase Formulation.	
Provided By	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Dispersed Multiphase**	
Requires	*Material*: one of **Gas**, **Liquid**, **Multi-Component Gas** > **Non-reacting**, **Multi-Component Liquid** > **Non-reacting**, **Eulerian Multiphase** > **Volume of Fluid (VOF)**, or **Eulerian Multiphase** > **Eulerian Multiphase Mixture** > **[any Eulerian Multiphase Mixture Model]** *Flow*: **Coupled Flow** or **Segregated Flow** *Optional Models*: **Dispersed Multiphase**	
Activates	Model Controls (child nodes)	**Eulerian Phases**. See Setting up the Dispersed Phases.
	Boundary Inputs	See Boundary Settings.
	Region Inputs	See Region Settings.
	Solvers	See Dispersed Multiphase Solvers.

Boundary Settings

Applies to Velocity Inlet and Mass Flow Inlet only.

The Background method is useful in cases where the dispersed phase and the continuous background phase move at a similar velocity.

Velocity Specification

Controls how you define velocity on the boundary.

Method	Corresponding Value Nodes
Components Define all the components of the velocity vector.	**Velocity** **Relative Velocity** Sets the velocity vector in the chosen *Coordinate System*. When relative, the vector is defined within the reference frame applied to the boundary.
Magnitude + Direction Define the magnitude of the velocity vector, and take the direction based on the **Flow Direction Specification**.	**Velocity Magnitude** **Relative Velocity Magnitude** Sets the velocity magnitude by the selected *Method*. When relative, the magnitude is defined within the reference frame applied to the boundary.
Background This method is available in the Dispersed Multiphase model only. The velocity of the dispersed phase is taken from the background flow.	None

Region Settings

Applies to fluid regions only.

Energy Source Option

Provides energy source options for the region.

Method	Corresponding Physics Value Nodes
None No energy source term is defined.	None
Volumetric Heat Source Specifies a user-defined volumetric heat source in W/m^3.	**Volumetric Heat Source** **Energy Source Temperature Derivative**
Total Heat Source Specifies a user-defined total heat source in W.	**Heat Source** **Energy Source Temperature Derivative**
Specific Heat Source Specifies a user-defined specific heat source in W/kg.	**Specific Heat Source** **Energy Source Temperature Derivative**

Mass Source Option

Allows a mass source term to be defined.

Method	Corresponding Physics Value Nodes
Activated	**Mass Source**
Deactivated No mass source term is defined.	None

Momentum Source Option

Allows a momentum source term to be defined.

Method	Corresponding Physics Value Nodes
None No momentum source term is defined.	None
Specified	**Momentum Source** **Momentum Source Velocity Derivative**

Contents:

DMP-VOF Impingement

DMP-VOF Impingement

The DMP-VOF Impingement model allows you to simulate DMP particles (droplets or bubbles) impinging on a VOF free surface phase or a wall. DMP particles are converted to the VOF phase. The Impingement model conserves fluid volume.

The Dispersed Multiphase model can be used as a lightweight replacement for Lagrangian particles in simulations for situations where the Lagrangian Multiphase model is too computationally expensive due to the large number of particles. Impingement transfers droplets (or bubbles) from the dispersed phase into the VOF phase.

Using Dispersed Multiphase particles for these simulations is significantly quicker than using Lagrangian particles. However, only single-component phases are supported.

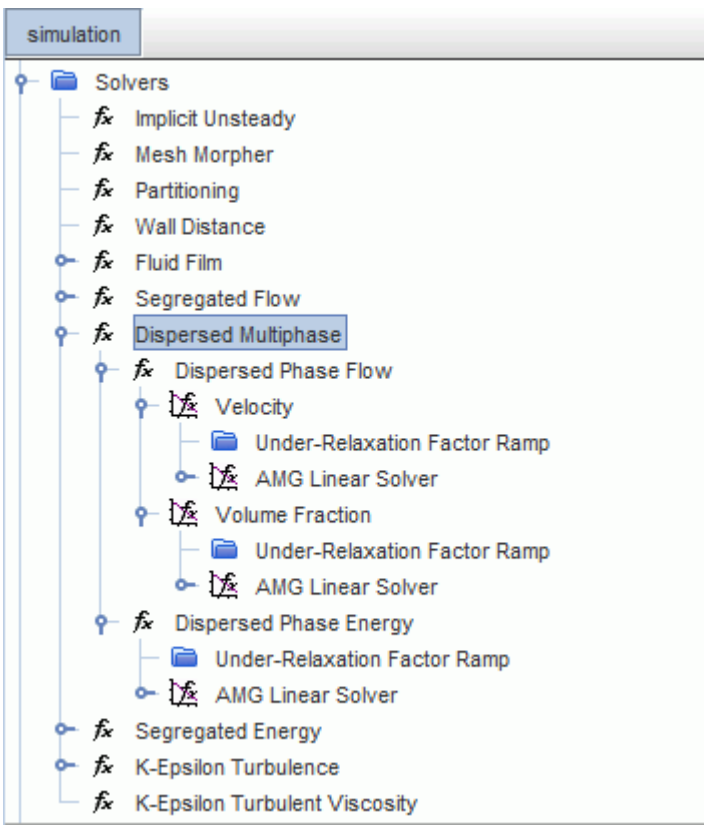
Impingement Model Reference

Theory	See VOF Phase Impingement
Provided By	**[phase interaction]** > **Models** > *Optional Models*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Multiphase Interaction** > **Phase Interactions** > **[phase interaction]** > **Models** > **Impingement**
Requires	The DMP-VOF Phase Interaction model must be defined.

	The *Time* model can be **Implicit Unsteady**, **PISO Unsteady**, or **Steady**.	
Activates	Boundary Settings	For Wall boundaries and Phase Impermeable boundaries: **[Boundary name]** > **Phase Conditions** > **[Phase Interaction]** > **Physics Conditions** > **Impingement Option** Set the **Enable Impingement** property to activate impingement at the boundary.
	Interface Settings	For the following interface types: Baffle, Contact, and Mapped: **[Interface name]** > **Phase Conditions** > **[Phase Interaction]** > **Physics Conditions** > **Impingement Option** Set the **Enable Impingement** property to activate impingement at the interface.

Dispersed Multiphase Solvers

When the Dispersed Multiphase model is selected, the Dispersed Multiphase solvers become available under the **Dispersed Multiphase** node.



The **Dispersed Multiphase** solver node contains sub-nodes that let you customize solver settings for the following:

- Dispersed Phase Flow

The Dispersed Phase Flow solver solves for the momentum and volume fraction:

- Dispersed Phase Velocity solver

  This solver uses a form of the Segregated Flow Velocity solver for the fluid film, and solves the momentum conservation equation, Eqn. (795).

- Dispersed Phase Volume Fraction solver

  This solver controls the solution update for the volume fractions of all the phases. The discretized form of the continuity equation Eqn. (794), is solved for each dispersed phase.

- Dispersed Phase Energy

  The Dispersed Phase Energy solver calculates the energy transport.

Dispersed Multiphase Solver Properties

The Dispersed Multiphase solvers generally use the same techniques as the Segregated Flow solvers. The dispersed multiphase solution is updated on every inner iteration.

Solver Frozen

When activated, prevents the solver from updating the solution while iterating.

For complex simulations, it can be useful to first let the flow converge before you activate the Dispersed Multiphase model. If the background flow is erratic and unconverged, the dispersed multiphase phases are also, and the simulation can diverge. To resolve this problem, you can freeze the Dispersed Phase Flow solver until the flow is stable.

Applies to the Dispersed Phase Flow solver and the Dispersed Phase Energy solver.

Temporary Storage Retained

When activated, retains the data that is in temporary storage at the end of the iteration.

Applies to the Dispersed Phase Flow solver and the Dispersed Phase Energy solver.

Under-Relaxation Factor

Governs the extent to which the newly computed solution supplants the old solution at each iteration. The default value is 0.7.

Applies to the Dispersed Phase Velocity solver, the Dispersed Phase Volume Fraction solver, and the Dispersed Phase Energy solver.

Minimum Volume Fraction

Sets a lower limit on the value of volume fraction in a cell for each dispersed phase.

Applies to the Dispersed Phase Volume Fraction solver only.

Maximum Volume Fraction

Sets an upper limit on the value of volume fraction in a cell for each dispersed phase.

Applies to the Dispersed Phase Volume Fraction solver only.

Dispersed Multiphase Formulation

The Dispersed Multiphase (DMP) model simulates dispersed phases in a Eulerian manner. The Dispersed Multiphase model combines aspects of both the Lagrangian Multiphase (LMP) model and the Segregated (Eulerian) Multiphase (EMP) models.

By default, DMP uses one-way coupling: the continuous phase influences the dispersed phase particles through terms such as drag in the momentum equation and heat transfer in the energy equation, but there is no reverse effect. When the optional Two-Way Coupling model is activated in DMP, the reverse effect is accounted for, and dispersed phase source terms appear in continuous phase equations.

In two-way coupled cases, drag force and heat transfer contribute to both the continuous phase and the dispersed phase, but with different signs. For example, the drag force between the two phases is an action-reaction force that satisfies Newton's third law. Activating two-way coupling ensures that the net drag force and heat transfer flux is zero in the continuous phase - dispersed phase system. Two-way coupling also affects the available volume from the point of view of the background phase: the volume that the dispersed phase occupies is accounted for, hence the available volume for the background phase is decreased. Typically, this effect becomes more important at higher particle loadings.

Transport Equations

For each dispersed phase, STAR-CCM+ solves transport equations for the conservation of mass and momentum.

STAR-CCM+ can also solve for energy transport if necessary.

The continuity equation (for a phase i) is given as:

$$\frac{\partial}{\partial t}\left(\alpha_i \rho_i\right) + \nabla\left(\alpha_i \rho_i \mathbf{v}_i\right) = \mathrm{s}_i^v \tag{794}$$

and the momentum equation is given as:

$$\frac{\partial}{\partial t}\left(\alpha_i \rho_i \mathbf{v}_i\right) + \nabla \cdot \left(\alpha_i \rho_i \mathbf{v}_i \otimes \mathbf{v}_i\right) = \mathrm{s}_i^v \tag{795}$$

Eqn. (794) is solved for as a volume fraction correction equation, and Eqn. (795) is solved as a velocity correction equation. These equations are essentially the same equations as are solved in Eulerian Multiphase (EMP) cases.

However, by default, the Dispersed Multiphase (DMP) model is one-way coupled: it solves only for the dispersed phases, while Eulerian Multiphase solves for all phases. In the DMP method, the background phase is handled with a single phase flow solver. Therefore, if you set up a particular simulation in both EMP and DMP, the EMP case has one more Eulerian phase than the DMP case. This extra phase is the background phase in DMP. In the EMP case, the volume fractions sum to unity. In the DMP case, the sum of volume fractions of the dispersed phases does not sum to unity, as the volume fraction of the background phase is not accounted for. When the optional Two-Way Coupling model is activated in DMP, these restrictions do not apply and the volume fractions sum to unity.

Phase Interactions

A phase interaction is defined as any process involving the dispersed phase that requires knowledge of the background continuous phase or a fluid film. For example, the gravity force $\rho\vec{g}$ is not a phase interaction, but the effect of buoyancy (included through the pressure gradient force) is a phase interaction.

STAR-CCM+ provides the following phase interaction models. The pressure gradient force is included automatically in a DMP phase interaction, but the other models are optional:

- Pressure Gradient Force
- Drag Force
- Heat Transfer
- Fluid Film Impingement
- VOF Phase Impingement

Pressure Gradient Force

The pressure gradient force acting on the dispersed phase is defined as:

$$\mathbf{F}_p = -\alpha_i \nabla p_{static} \tag{796}$$

This term contains the buoyancy force $-\rho_c g$, through the definition of the hydrostatic pressure as $p_{hs} = -\rho_c g h$.

The pressure gradient force is included in the momentum equation directly, as:

$$\frac{\partial}{\partial t}\left(\alpha_i \rho_i \mathbf{v}_i\right) + \nabla \cdot \left(\alpha_i \rho_i \mathbf{v}_i \otimes \mathbf{v}_i\right) = \mathrm{s}_i^v - \alpha_i \nabla p_{static} \tag{797}$$

Therefore the dispersed phase always experiences the pressure gradient force from the background flow.

Drag Force

Consider a DMP-VOF phase interaction. A cell c is in the VOF-VOF interfacial region with a volume fraction α_1 of VOF phase V_1 and a volume fraction α_2 of VOF phase V_2, and $\alpha_1 + \alpha_2 = 1$. The volume fraction of the dispersed phase in the cell is α_D. The density, dynamic viscosity, and specific heat of each VOF phase V_i is ρ_i, μ_i, and $c_{p,i}$. The dispersed phase properties are ρ_D, μ_D, and $c_{p,D}$. The characteristic particle diameter of the dispersed phase is D. The unique velocity field of the background flow is $\mathbf{v}(x,t)$, and the dispersed phase velocity field is $\mathbf{v}_D(x,t)$. The phase slip velocity between the dispersed phase and the background flow is $\Delta\mathbf{v} = \mathbf{v}_D - \mathbf{v}$.

For a DMP-Physics Continuum Interaction, the continuous drag force (in N/m^3) is:

$$f_d(x,t) = \frac{1}{2} C_d \left(\frac{6\alpha_D}{4D}\right) \rho_D \left\|\Delta\mathbf{v}\right\|_2 \Delta\mathbf{v} \tag{798}$$

Three methods are available for the calculation of the drag coefficient C_d:

- Constant
- Field Function
- Schiller-Naumann correlation, which is defined as:

$$C_d = \begin{cases} \frac{24}{Re_p}\left(1 + \frac{1}{6}Re_p^{2/3}\right), & Re_p \leq 1000 \\ 0.424, & Re_p > 1000 \end{cases} \tag{799}$$

with the particle Reynolds number defined as:

$$Re_p = \frac{\overline{\rho}\left\|\Delta\mathbf{v}\right\|_2 D}{\overline{\mu}} \tag{800}$$

The mixture density is volume-weighted, that is:

$$\overline{\rho} = \alpha_1\rho_1 + \alpha_2\rho_2 \tag{801}$$

Three methods are available for the mixture dynamic viscosity $\overline{\mu}$: Constant, Field Function, and Volume-weighted.

For a DMP-VOF Phase Interaction, the continuous drag force is weighted by the volume fraction:

$$f_{d,i}(x,t) = \frac{1}{2}\alpha_i C_{d,i}\left(\frac{6\alpha_D}{4D}\right)\rho_D\left\|\Delta\mathbf{v}\right\|_2\Delta\mathbf{v} \tag{802}$$

Given that $\alpha_1 + \alpha_2 = 1$, when the same constants or field function methods are specified for the drag coefficient in both VOF phases, the same drag force is obtained for both the DMP-VOF Phase Interaction and the DMP-Physics Continuum Interaction.

However, if the Schiller-Naumann correlation is used for the calculation of the drag coefficient, the results are different for the two phase interactions. The phase particle Reynolds number that is used in the calculation of $C_{d,i}$ is defined as:

$$Re_{p,i} = \frac{\rho_i\left\|\Delta\mathbf{v}\right\|_2 D}{\mu_i} \tag{803}$$

The phase properties ρ_i and μ_i are used instead of the mixture properties $\overline{\rho}$ and $\overline{\mu}$. C_d is continuous across the interface but $C_{d,i}$ is discontinuous across the interface. A linear combination $\alpha_1 C_{d,1} + \alpha_2 C_{d,2}$ is used for the drag coefficient in the interfacial region, which is different from C_d.

By default, the drag is one-way coupled: the background fluid does not feel the reacting force that the particles exert. The drag acts as a contribution to $\overline{s}_i^v$ in Eqn. (795). In two-way coupled cases, the drag force contributes to both the continuous phase and the dispersed phase, but with different signs.

Heat Transfer

Heat transfer causes the particles to heat or cool according to the temperature of their environment. For a single particle, the heat transfer (in W/m^2s) is defined as:

$$Q_c = h_c A\left(T_p - T_c\right) \tag{804}$$

where h_c is the heat transfer coefficient, A is the area of the particle, and T_p and T_c are the particle and background temperatures, respectively. In a continuous approach, the heat transfer can be expressed as:

$$\overline{Q}_{c,i} = \overline{h}_{c,i}\left(\frac{6\alpha_i}{D_i}\right)\left(T_i - T_c\right) \tag{805}$$

which is defined as positive for heat flowing from the particles to the background.

There are three methods available for the heat transfer coefficient:

- Constant
- Field Function
- Ranz-Marshall correlation, using the usual formulation of the heat transfer in terms of the Nusselt number:

$$h = \frac{kNu_p}{D} \tag{806}$$

and correlating the Nusselt number as:

$$Nu_p = 2\left(1 + 0.3{Re_p}^{1/2}Pr^{1/3}\right) \tag{807}$$

By default, the heat transfer is one-way coupled: the background fluid does not change temperature, even when the particles do. In two-way coupled cases, the heat transfer contributes to both the continuous phase and the dispersed phase, but with different signs so the net heat transfer flux is zero.

Fluid Film Impingement

Particles that reach the free-surface between the background fluid and a liquid film can be set to impinge onto the liquid film. All of the particle mass, energy, and momentum are transferred to the liquid film.

The mass source (kg/m^2s) due to impingement of a phase i is:

$$\dot{m}_{imp} = \rho_i \mathbf{v}_i \cdot \widehat{A}_{fs} \tag{808}$$

which is essentially the outgoing mass flux. Corresponding sources are also applied to the momentum and energy transport equations.

VOF Phase Impingement

When two-way coupling is activated, particles can impinge onto the surface of a VOF phase or a wall. A specified fraction of the impinging DMP phase is transferred to the VOF phase.

For a VOF free surface case, the mass source due to impingement of a DMP phase is a fraction of the incoming flux. The fraction equals to the volume fraction of VOF phase α_V in the cell as it is a measure of the collision probability between the DMP phase and the VOF phase.

Wall impingement is included at a wall boundary when the **Impingement Option** is activated for the phase interaction. See DMP-VOF Impingement.

For a wall impingement case, the mass source is a fraction of the incoming DMP flux. This fraction is equal to $1.0 - \alpha_V$, which takes into account that VOF phase exists in the cell and eliminates double-accounting of the impingement source. The incoming DMP flux on the wall is calculated by taking only the normal component of the DMP velocity to the wall.

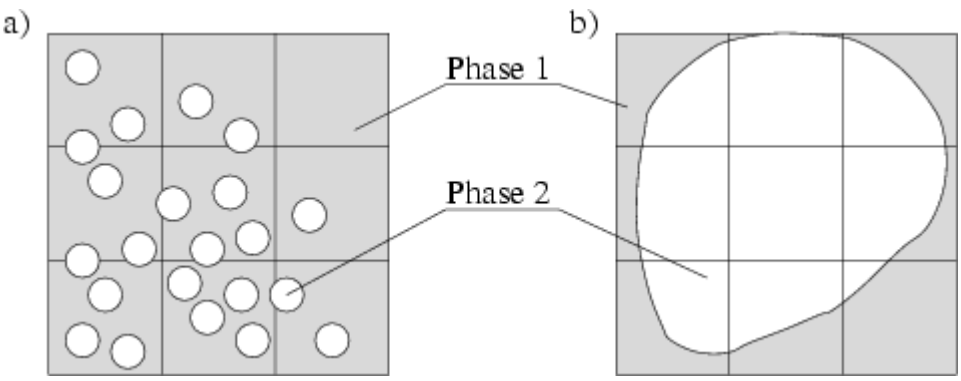
Corresponding sources are also applied to the momentum and energy transport equations.

Using the Volume Of Fluid (VOF) Multiphase Model

The Volume Of Fluid (VOF) Multiphase model is a simple multiphase model. It is suited to simulating flows of several immiscible fluids on numerical grids capable of resolving the interface between the phases of the mixture.

The VOF Multiphase model is used to solve problems involving immiscible fluid mixtures, free surfaces, and phase contact time. In such cases, there is no need for extra modeling of inter-phase interaction, and the model assumption that all phases share velocity, pressure, and temperature fields becomes a discretization error.

The figure below provides an illustration of unsuitable grids (left) and suitable grid (right) for two-phase flows using the VOF model.



Due to its numerical efficiency, the model is suited for simulations of flows where each phase constitutes a large structure, with a relatively small total contact area between phases. A good example of this type of flow is sloshing flow in a water tank, where the free surface always remains smooth. If the tank movement becomes pronounced, this results in breaking waves, large numbers of air bubbles in the water, and water droplets in the air. The method would then require a fine mesh (at least three cells across each droplet/bubble) to produce small modeling errors.

The spatial distribution of each phase at a given time is defined in terms of a variable that is called the volume fraction. A method of calculating such distributions is to solve a transport equation for the phase volume fraction (see Basic VOF Model Equations). The method uses the STAR-CCM+ Segregated Flow model.

In order to obtain sharp interfaces between the phases, use the 2nd-order discretization scheme and set the appropriate value of the Sharpening Factor parameter. For more information, see The HRIC Convection Discretization Scheme.

Thermal effects can also be included with the Segregated Multi-Phase Temperature model.

The VOF Multiphase model can be used to model change of state. Melting of ice and boiling of water are everyday examples of transitions between solid, liquid, and gaseous states of matter. STAR-CCM+ provides models to describe boiling, cavitation, and melting and solidification.

Contents:

Setting Up a VOF Multiphase Simulation
What Is the VOF Multiphase Model?
What Is the Segregated Multi-Phase Temperature Model?
Replacing VOF Phases
What Is Surface Tension?
Using the Contact Time Model
Using the Interface Momentum Dissipation Model
Mass Error Rate Model Reference
Working with Wall Porosity
Modeling Evaporation/Condensation
Modeling Boiling
Modeling VOF with Radiation
Modeling Gas Dissolution
Modeling Cavitation
Modeling Melting and Solidification
VOF Guidelines
VOF Formulation

Setting Up a VOF Multiphase Simulation

The Volume of Fluid (VOF) Multiphase model is used to solve problems involving immiscible fluid mixtures, free surfaces, and phase contact time.

To set up a VOF multiphase simulation:

1. Set up the Volume of Fluid (VOF) model in the physics continuum.

   See Setting Up the Volume of Fluid (VOF) Model.
2. Define the composition of the multiphase mixture.

   The Eulerian Multiphase model controls the definition of Eulerian phases within the physics continuum. You specify the bulk properties and phase composition of the fluid mixture.

   See Defining the Eulerian Phases.
3. Specify the material components for each phase, and set the material properties.

   You set the material properties for each individual phase and set the material properties for the mixture. For a multi-component phase, you also set the material properties for each individual component of the phase.

   See Specifying the Material Components.
4. Define the phase interactions.

   See Controlling Phase Interactions in VOF Simulations.
5. Set the appropriate initial conditions for each phase.

   See Setting Initial Conditions.
6. Set the Phase Source Option for each fluid region.

See Setting Region Conditions.

7. Set the boundary conditions for the flow.

   See Setting Boundary Conditions.

8. Set the appropriate solver parameters.

   The Segregated Volume Fraction Solver solves the discretized volume-fraction conservation equation for each phase present in the flow.

   See Segregated Volume Fraction Solver Properties.

9. Set the appropriate stopping criteria.

   You can optimize the computational time of your VOF simulation by automatically adjusting the number of inner iterations per time-step.

   See Setting Up Automatic Inner Iterations.

Contents:

Setting Up the Volume of Fluid (VOF) Model
Defining the Eulerian Phases
Specifying the Material Components
Controlling Phase Interactions in VOF Simulations
Setting Initial Conditions
Setting Region Conditions
Setting Boundary Conditions
Setting Up Automatic Inner Iterations

Setting Up the Volume of Fluid (VOF) Model

The Volume of Fluid (VOF) model is used for solving immiscible fluid problems.

To set up the Volume of Fluid (VOF) model:

1. In the physics continuum, open the *Physics Model Selection* dialog.
2. Select **Implicit Unsteady** from the *Time* group box.
3. Specify the material model as a multiphase fluid by selecting **Eulerian Multiphase** from the *Material* group box.

   The **Multiphase Interaction** model is selected automatically.

4. Select **Volume of Fluid (VOF)** from the *Eulerian Multiphase Model* group box.

   The **Multiphase Equation of State**, **Gradients**, and **Segregated Volume Flux Based Flow** models are selected automatically.

   See Segregated Flow Model Family Reference.

5. Make the appropriate selections from the *Viscous Regime* group box and from the *Space* group box.
6. Make the appropriate selections from the *Optional Models* group box:
   - If gravity forces significantly influence the solution, select **Gravity**.
   - To include thermal effects in your simulation, select **Segregated Multiphase Temperature**.
   - To model surface gravity waves, select **VOF Waves**.

7. Click **Close**.

Defining the Eulerian Phases

The Eulerian Multiphase model controls the definition of Eulerian phases within the physics continuum. You define the bulk properties and phase composition of the fluid mixture in two separate stages.

When you activate the Eulerian Multiphase model in the physics continuum, STAR-CCM+ automatically selects the multiphase fluid model and sets up a default immiscible fluid mixture. It contains no Eulerian phases in its initial state.

Note the following:

- The most common situation is a two-phase, liquid-gas mixture.
- Default names, material properties, and (**Constant**) property values are assigned to materials retrieved from the database.

  If none of these materials correspond to the phases present in your problem, do the following:

  - Rename the appropriate materials by right-clicking its node in the tree.
  - Open the corresponding **Properties** node and specify new settings as necessary.
- The order in which phases are defined is not important in principle.

  However, if you want to model surface tension in the simulation at some point, you are advised to define the liquid phase as the first component, as shown above.
- For laminar flows, it is possible to select the Non-Newtonian Generalized Power Law for the dynamic viscosity of liquid materials in a VOF simulation.
- STAR-CCM+ does not support multi-component VOF phases in combination with a mixing-plane interface. Single-component VOF phases only are supported.

Create the necessary phases and select the appropriate phase models.

The number of phases required depends on what you are modeling. Melting and Solidification require the liquid phase only - the solid phase is assumed to be the solid form of the liquid.

For each phase, do the following:

1. Create a Eulerian phase and then open the *Phase Model Selection* dialog.
2. From the *Material* group box, select the appropriate model:

Material	Notes
Gas	
Multi-Component Gas	
Liquid	
Multi-Component Liquid	Boiling is not supported.

3. For multi-component materials only, from the *Reaction Regime* box, select the appropriate option:

Reaction Regime	Notes
Non-reacting	The Species model is selected automatically.

Reaction Regime	Notes
Reacting	See Reacting Species Transport.

4. From the *Equation of State* group box, select the appropriate option:

Equation of State	Notes
Constant Density	
Polynomial Density	
User Defined EOS	
Ideal Gas	Gas materials only. Energy model is selected automatically.
Real Gas	Gas materials only. IAPWS-IF97 (Steam) model is selected automatically for single-component gas.
IAPWS-IF97 (Water)	Liquid materials only.

5. From the *Optional Models* group box, select the appropriate options:

Optional Model	Notes
Wall Porosity	Not available for multi-component liquids. See Wall Porosity.
Exact Wall Distance **PDE Wall Distance**	Used to account for near-wall effects. See Modeling Wall Distance.
Mass Error Rate	Used for dealing with mass conservation errors. See Mass Error Rate Model Reference.
Passive Scalar	Used for tracking the phase material within the simulation. See Modeling Passive Scalars.
Melting-Solidification	Available for liquid materials only. Adds the following models to the *Optional Models* group box: • **Slurry Viscosity** See Slurry Viscosity Model Reference. • **Mushy Zone Permeability** See Mushy Zone Permeability Model Reference.

Specifying the Material Components

Specify the material components for each phase, and set the material properties.

You set the material properties for each individual phase and set the material properties for the mixture. For a multi-component phase, you also set the material properties for each individual component of the phase.

1. Specify the material components for each phase.

   The example steps below are for the liquid phase.

   For a single-component liquid:

   a) Right-click the **Models** > **Liquid** > **H2O** node, and then click **Replace with...**.

   b) In the *Replace Material* dialog, select the appropriate liquid.

   For a multi-component liquid:

   a) Right-click the **Models** > **Multi-Component Liquid** > **Liquid Components** node, and then click **Select Mixture Components**.

   b) In the *Select Mixture Components* dialog, select the appropriate components.

   The steps for the gas phase are identical. However, in some cases, the gaseous form of the solvent is not available in the STAR-CCM+ material database. To specify the gaseous component, you can add the appropriate material to the material database.

2. Specify the material properties for the mixture, for each phase, and for each component (if applicable).

   The example steps below are for a liquid phase.

   a) Specify the following mixture properties under the **[physics continuum]** > **Models** > **Eulerian Multiphase** > **Mixture** > **Material Properties** node:

      - **Dynamic Viscosity**

   b) Specify the following phase properties under the **[physics continuum]** > **Models** > **Eulerian Multiphase** > **Eulerian Phases** > **[liquid phase]** > **Models** > **Multi-Component Liquid** > **Material Properties** node:

      - **Density**
      - **Dynamic Viscosity**
      - **Molecular Diffusivity**
      - **Molecular Weight**
      - **Saturation Pressure** (applies to single-component phases only)
      - **Turbulent Schmidt Number** (applies to turbulent flows only)

   c) Specify the following component properties under the **[liquid phase]** > **Models** > **Multi-Component Liquid** > **Liquid Components** > **[component]** > **Material Properties** node:

      - **Density**
      - **Dynamic Viscosity**
      - **Molecular Weight**

Controlling Phase Interactions in VOF Simulations

You create and define the appropriate VOF phase interactions.

The procedure for creating a phase interaction is described in Creating a Phase Interaction.

The following phase interaction models are available in VOF simulations:

- VOF-VOF Phase Interaction model, which controls the interaction between two VOF phases.

- Film-VOF Phase Interaction model, which controls the interaction between the fluid film phase and a VOF phase.

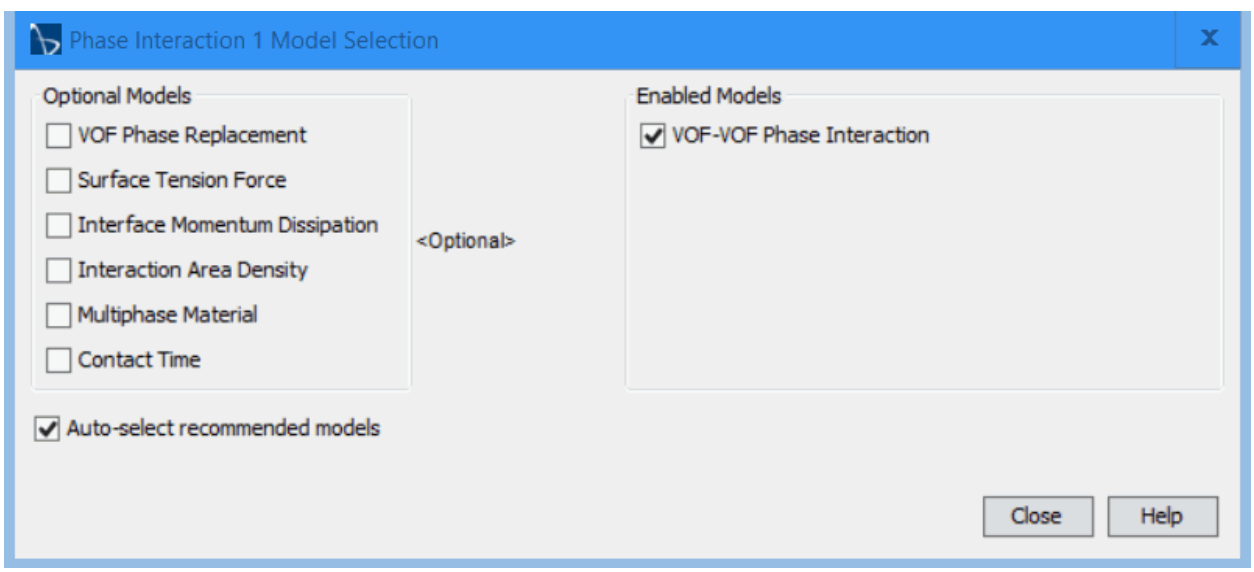
  This model is available only when the Fluid Film model is activated in the physics continuum.

Phase Combination Modeled	Physical Phenomena Modeled
Gas-Liquid	Evaporation Condensation Boiling Cavitation Gas Dissolution
Liquid-Liquid	Surface Waves
Liquid-Solid	Melting Solidification
Gas-Solid	

To select the VOF-VOF phase interaction model:

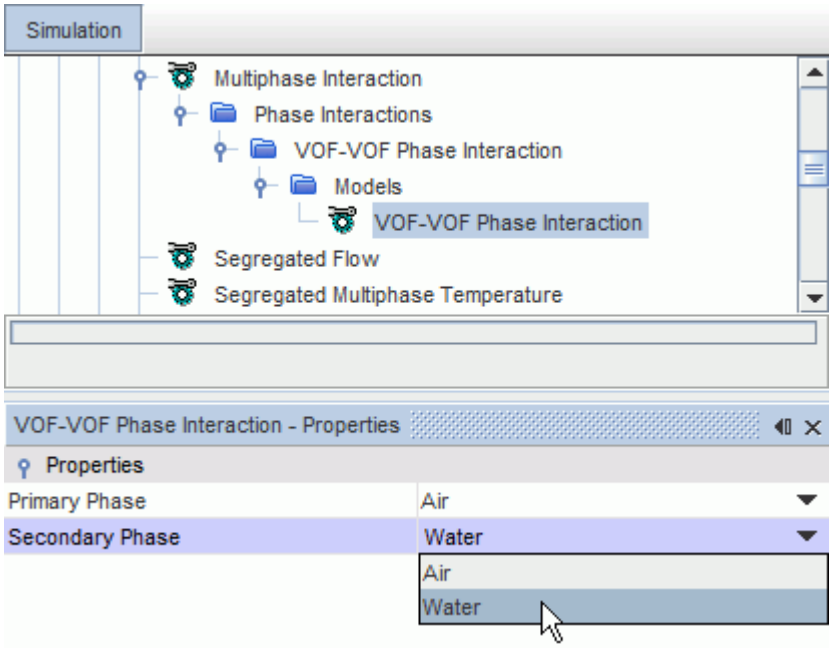
1. Select **VOF-VOF Phase Interaction** from the *Phase Interaction Topology* group box.

   Further VOF models are available in the *Optional Models* group, as shown below.



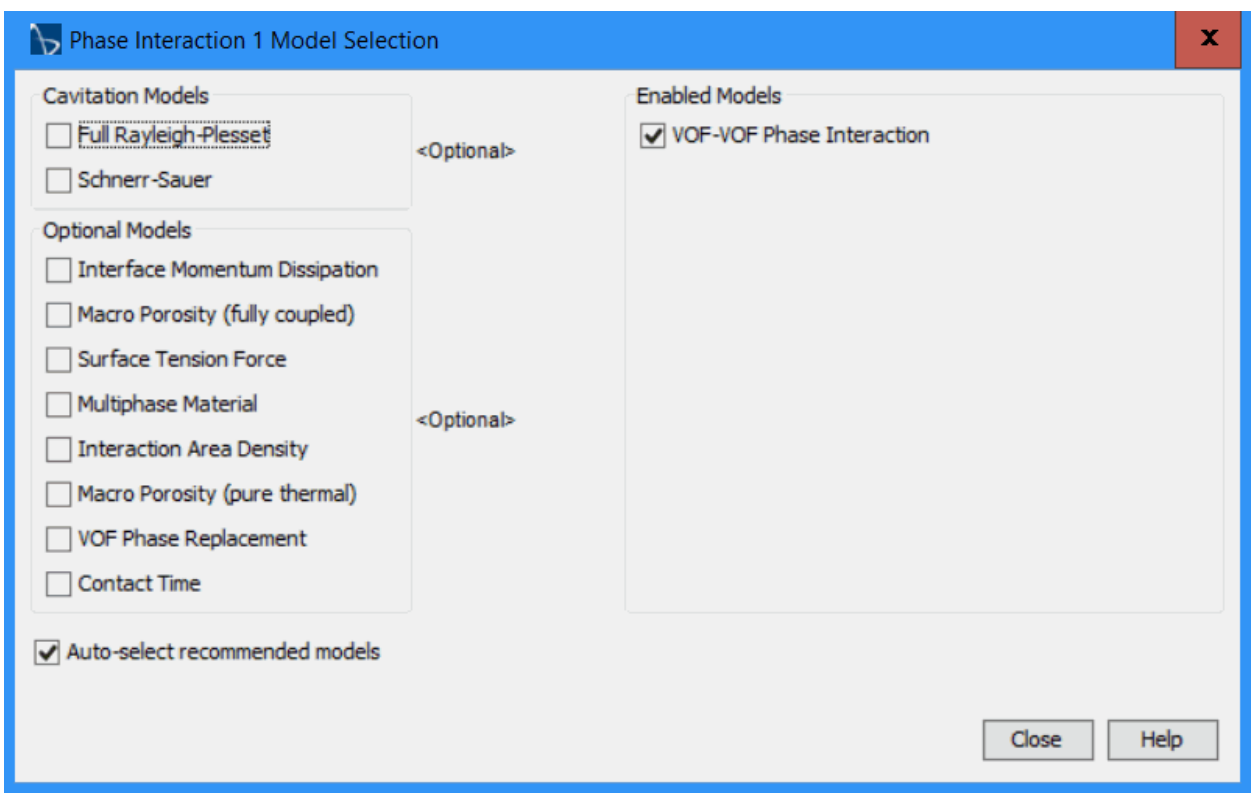
Set the primary and secondary phases in the phase interaction. These settings affect the optional models that are made available.

2. Select the **[phase interaction]** > **Models** > **VOF-VOF Phase Interaction** node and set the primary and secondary phases:



Define the phase interaction.

3. Reopen the *Phase Interaction Model Selection* dialog and select the appropriate optional models for the phase interaction.



Mass Transfer Models:

Model	Relevant information
Evaporation/Condensation	See Modeling Evaporation and Condensation.
Boiling	Two models are available: **Rohsenow Boiling** and **Transition Boiling** See Modeling Boiling.
Cavitation	Two models are available: **Full Rayleight-Plesset** and **Schnerr-Sauer** See Modeling Cavitation.
Gas Dissolution	See Modeling Gas Dissolution.
Melting and Solidification	See Modeling Melting and Solidification.

Other models:

Model	Relevant information
Contact Time	See Using the Contact Time Model.
Interaction Area Density	See Interaction Area Density.
Interface Momentum Dissipation	See Using the Interface Momentum Dissipation Model.
VOF Phase Replacement	See Replacing VOF Phases.
Surface Tension Force	See Modeling Surface Tension in VOF Simulations.
Macro Porosity (fully coupled) and **Macro Porosity (pure thermal)**	See Detecting Shrinkage-Related Defects.

Setting Initial Conditions

As the VOF model uses multiple Eulerian phases, it is necessary to set the initial volume fraction of each phase.

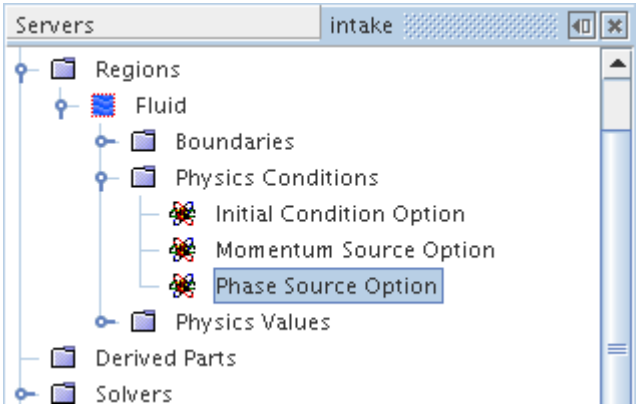
You define the initial distribution of each phase, including the gas (if present), as a volume fraction under the Initial Conditions of the physics continuum. STAR-CCM+ obtains the volume fraction of the last-defined phase automatically, by subtracting the sum of the volume fractions of the other phases from 1.0. For multi-component phases, you specify the initial conditions for the mass fraction or molar fraction of each component of the phase.

This procedure is described in Setting Initial Volume Fraction.

Setting Region Conditions

This section describes how to set the Phase Source Option for the region.

Following the selection of the VOF model, the node of every region that accesses this continuum contains an additional **Phase Source Option** sub-node.



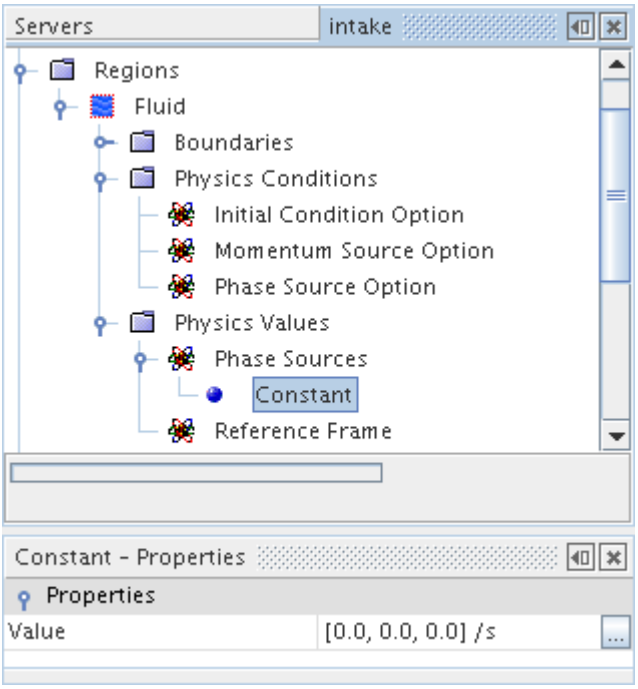
The Phase Source is a region source. It specifies the rate at which a unit volume of the phase is added or removed per unit volume of the region (Volume/Time/Volume). In the Physics Conditions of a region, you can set up user-specified phase sources by activating the *Phase Source Term* property.

Setting Phase Sources

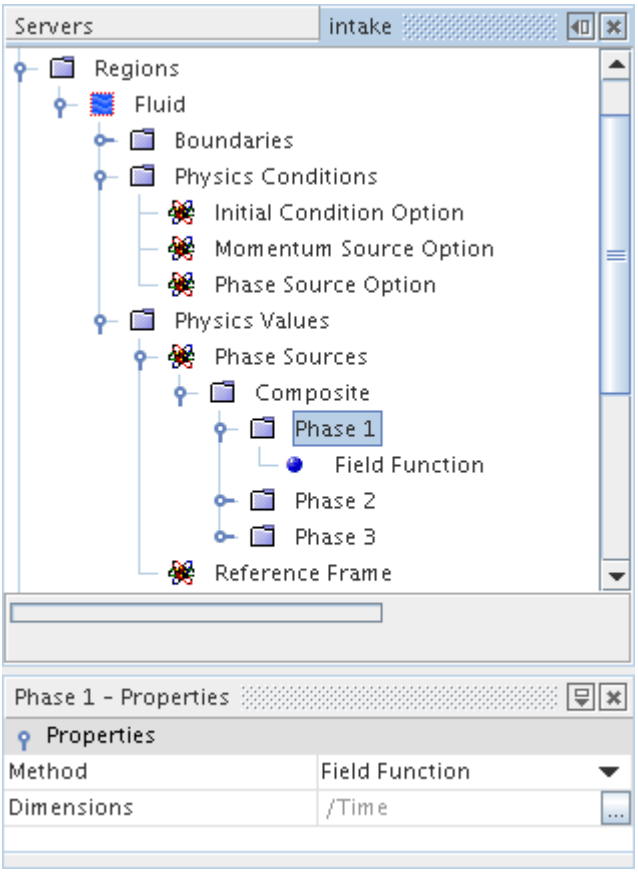
Once the Phase Source Option is activated, you can specify Phase Sources of each of the defined phases with the options of this scalar array profile.

To specify the Phase Sources:

- Option **Constant** defines sources uniformly for all phases.



- Option **Composite** defines the source on each phase separately.



Setting Boundary Conditions

As the VOF model uses multiple Eulerian phases, it is necessary to set the volume fraction of each phase at appropriate boundaries.

This procedure is described in the section, Setting Volume Fraction at Flow Boundaries.

The following boundary conditions apply to specific VOF applications:

- If surface tension effects are included in the simulation, all closed (wall) boundaries also contain an additional Contact Angle sub-node. This node lets you specify the contact angle at each of these boundaries.
- If a stagnation inlet is used with a VOF mixture for supersonic inflow, only a single phase of the VOF mixture is allowed to enter the domain through each inlet boundary.

Setting Up Automatic Inner Iterations

You can optimize the computational time of your VOF simulation by automatically adjusting the number of inner iterations per time-step. You achieve this optimization by creating a stopping criterion for the inner iterations of the activated solvers based on the relative mass error rate. Activating this stopping criterion lets STAR-CCM+ automatically determine the number of inner iterations to reach a minimum accuracy limit for mass conservation.

Before you begin, select the mass error rate model.

For a definition of the relative mass error rate, see Eqn. (809).

To set up automatic inner iterations:

1. Right-click the **Reports** node and select **New Report** > **Maximum**.
2. Select the newly created **Maximum 1** node and set *Scalar Field Function* to **Relative Mass Error Rate** > **Relative Mass Error Rate of [Phase]**.
3. Right-click the **Maximum 1** node and select **Create Monitor from Report**.
4. Select the **Monitors** > **Maximum 1 Monitor** node and set *Trigger* to **Iteration**.
5. Right-click the **Stopping Criteria** node and select **Create New Criterion** > **From Monitor...** > **Maximum 1 Monitor**.
6. Right-click the **Stopping Criteria** node and select **Create New Criterion** > **Minimum Inner Iterations**.
7. Edit the **Stopping Criteria** node and set the following the properties:

Node	Property	Setting
Maximum 1 Monitor Criterion	*Logical Rule*	**And**
└ **Minimum Limit**	*Minimum Value*	Provide a minimum limit for the relative mass error rate. You choose the minimum limit, for example, by considering the variation in weight of a casted part. If the weight varies, for instance, by 3% and the total physical simulation time is, for example, 8 s, a value like 3%/8s = 0.375 %/s is appropriate for this example.
Minimum Inner Iterations	*Minimum Inner Iterations*	Specify the minimum number of inner iterations for each unsteady time-step.
	Logical Rule	**And**

Volume Of Fluid (VOF) Multiphase Model Reference

Volume Of Fluid (VOF) is a simple multiphase model. It is suited to simulating flows of several immiscible fluids on numerical grids capable of resolving the interface between the phases of the mixture.

Volume Of Fluid (VOF) Multiphase Model Reference

Theory	See VOF Formulation.
Provided By	**[physics continuum]** > **Models** > *Eulerian Multiphase Model*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Volume Of Fluid (VOF)**

Requires	*Material*: **Eulerian Multiphase** *Eulerian Multiphase Model*: **Volume Of Fluid (VOF)**	
Properties	Key properties are: Convection, Compressibility Enhancement See Volume Of Fluid (VOF) Properties.	
Activates	Physics Models	*Viscous Regime* See Modeling Flow and Energy.
	Initial Conditions	See Initial Conditions.
	Region Inputs	See Region Settings.
	Solvers	Segregated Volume Fraction See Segregated Volume Fraction Solver Properties.

Volume of Fluid (VOF) Properties

Convection

Specifies a convection.

- **1st-order**
  Selects the first-order convection.
- **2nd-order**
  Selects the second-order convection.

Compressibility Enhancement

Enhances the simulation of highly-compressible (that is, where the Mach number exceeds about 0.3) VOF simulations.

When this option is activated, the gas density at internal faces is approximated by using the density reconstruction gradient of the upwind cell, instead of using the arithmetic mean value of the density of both adjacent cell centers. Including the linearization for convective fluxes to the pressure correction equation is necessary only in variable density cases.

Computing the reconstruction gradient is computationally expensive, so this option is deactivated by default. Many VOF simulations do not require this more sophisticated approach.

CFL_l

The lower Courant number limit in the HRIC convection discretization scheme. This value is Cu_l in Eqn. (829).

CFL_u

The upper Courant number limit in the HRIC convection discretization scheme. This value is Cu_u in Eqn. (829).

Sharpening Factor

This factor is used to reduce numerical diffusion in the simulation. The valid values are 0.0 through 1.0.

When the sharpening factor is set to 0.0, there is no reduction in numerical diffusion. This value is the default because, in most cases, the recommended 2nd-order discretization scheme is sufficient to achieve a sharp interface between the two phases.

When the sharpening factor is set to 1.0, there is no numerical diffusion and, therefore, a very sharp interface is obtained.

For some flows, such as surface-tension dominated flows or tank sloshing flows, it is recommended that you increase the sharpening factor to improve the resolution of the interface between the phases. However, this additional sharpening can result in a non-physical alignment of the free surface with the grid lines.

If a non-zero value is specified, an additional term (Eqn. (819)) is added to the VOF transport equation (Eqn. (818)). This value is C_α in Eqn. (819)

Angle Factor

The angle factor in the HRIC convection discretization scheme for interface tracking. This value is C_θ in Eqn. (830).

Initial Conditions

Volume Fraction

Region Settings

Applies to fluid regions.

Volume Fraction Source Option

Method	Corresponding Physics Value Nodes
Phase Source Term When activated, user-specified volume fraction sources are enabled.	**Volume Fraction Sources** **Volume Fraction Sources Pressure Derivative** **Volume Fraction Sources Volume Fraction Derivative**

Segregated Volume Fraction Solver Properties

The Segregated Volume Fraction Solver controls the solution update for the phase volume fractions. More specifically, it solves the discretized volume-fraction conservation equation for each phase present in the flow.

The properties in the *Expert* category are used in a temporary debug situation, at the expense of simulation accuracy and memory usage. It is recommended that you do not change any of these properties unless you are thoroughly familiar with the discretization techniques used in STAR-CCM+.

Under-Relaxation Factor

At each iteration, this property governs the extent to which the newly computed solution supplants the old solution. The default value is 0.7.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Segregated Multiphase Temperature Model Reference

The Segregated Multiphase Temperature model lets you control thermal effects in your simulation.

The Segregated Multiphase Temperature model solves the total energy equation with temperature as the solved variable. Enthalpy is then computed from temperature according to the equation of state.

This model is appropriate for simulations that do not involve combustion.

Segregated Multiphase Temperature Model Reference

Provided By	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Segregated Multiphase Temperature**	
Requires	*Material*: **Eulerian Multiphase** *Eulerian Multiphase Model*: **Volume Of Fluid (VOF)** *Optional Models*: **Segregated Multiphase Temperature**	
Properties	Key properties are: *Convection*. See Segregated Multiphase Temperature Properties.	
Activates	Model Controls (child nodes)	**Bounded Differencing**, when *Convection* is set to **MUSCL 3rd-order/CD**. See Bounded Differencing.
	Reference Values	See Reference Values.

	Initial Conditions	See Initial Conditions.
	Boundary Inputs	See Boundary Settings.
	Region Inputs	See Region Settings.
	Solvers	Segregated Energy solver. See Segregated Energy Solver for Multiphase.
	Report Options	See Reports.
	Field Functions	See Field Functions.

Segregated Multiphase Temperature Properties

Convection

For guidance on selecting a convection, see Finite Volume Discretization.

- **1st-order**
  Selects the first-order convection.
- **2nd-order**
  Selects the second-order convection.
- **MUSCL 3rd-order/CD**
  Selects the hybrid MUSCL 3rd-order/CD convection scheme. Adds a **Bounded Differencing** sub-node to the **Segregated Multiphase Temperature** node.

  See Bounded Differencing.

Flow Boundary Diffusion

Activate or deactivate the flow-boundary diffusion fluxes that are given in Eqn. (1593). A realistic temperature value must be set on flow boundaries when this option is activated. This option is activated by default.

Secondary Gradients

Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces

- **On**
  Include both secondary gradients.
- **Off**
  Exclude both secondary gradients.
- **Interior Only**
  Include the interior secondary gradients only.
- **Boundaries Only**
  Include the boundary secondary gradients only.

Reference Values

Minimum Allowable Temperature

The smallest temperature value that is permitted anywhere in the continuum. The default value is 100 K.

The Segregated Multiphase Temperature model limits temperature corrections such that the corrected value does not go below this minimum. If this occurs, a message is printed to the *Output* window.

Maximum Allowable Temperature

The largest temperature value that is permitted anywhere in the continuum. The default value is 5000 K.

The Segregated Multiphase Temperature model limits temperature corrections such that the corrected value does not exceed this maximum. If this occurs, a message is printed to the *Output* window.

Initial Conditions

Static Temperature

Boundary Settings

Stagnation Inlet, Mass Flow Inlet

Total Temperature

Velocity Inlet, Pressure Outlet

Static Temperature

Region Settings

Applies to fluid regions.

Energy Source Option

See Energy Source Option.

Reports

Heat Exchanger (Dual Stream)

See Heat Exchanger.

Heat Exchanger (Single Stream)

See Heat Exchanger.

Heat Transfer

See Heat Transfer.

Isentropic Efficiency

See Isentropic Efficiency.

Field Functions

Relative Total Enthalpy

Relative Total Temperature

The temperature obtained by isentropically bringing the flow to rest in the relative frame of motion.

Rothalpy

Specific Heat

The specific heat of the model material.

Temperature

The temperature field.

Temperature Coefficient

See Temperature Coefficient Field Function Properties.

Total Enthaply

The total enthalpy H is the sum of static enthalpy plus kinetic energy, $H = h + \frac{v^2}{2}$.

Total Temperature

The total temperature is the temperature that is obtained by isentropically bringing the flow to rest. For an ideal gas with constant specific heat, it is defined as:

$$T_t = T\left(1 + \frac{(\gamma-1)}{2}M^2\right)$$

where T , M , and γ are the static temperature, Mach number, and ratio of specific heats, respectively.

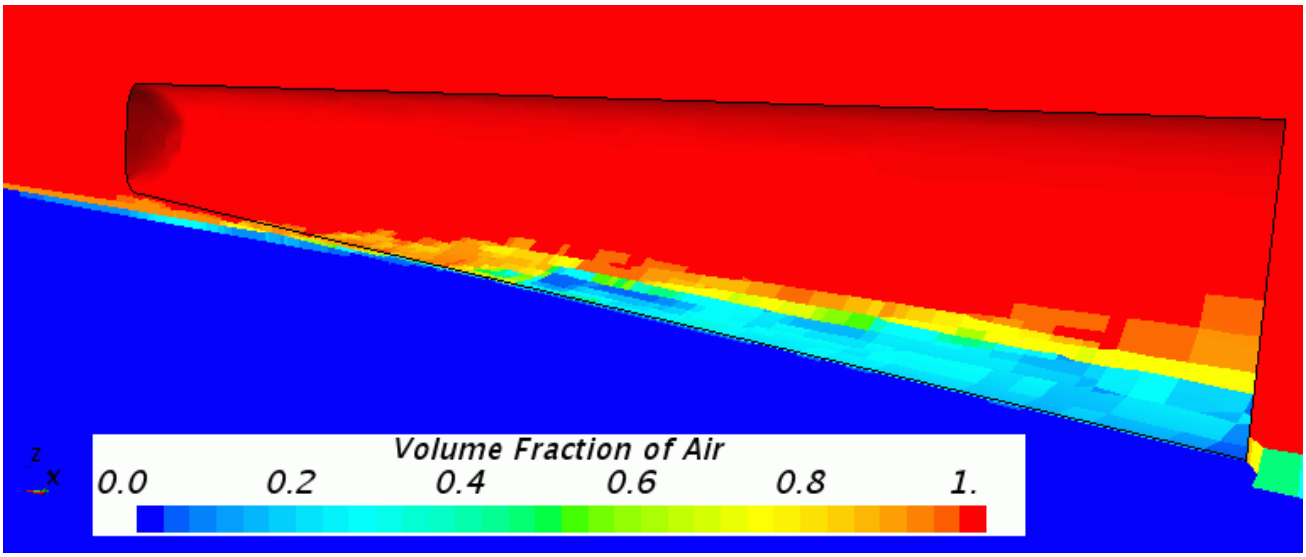
Replacing VOF Phases

In some VOF simulations, under-resolution of the mesh, time-step, or other numerical issues can cause numerical ventilation to occur. The VOF Phase Replacement model lets you reduce or eliminate numerical ventilation by replacing the unwanted phase (typically gas) with the appropriate phase (typically liquid) in specified regions. If appropriate, you can specify a maximum volume fraction in the phase replacement.

Ventilation and Numerical Ventilation

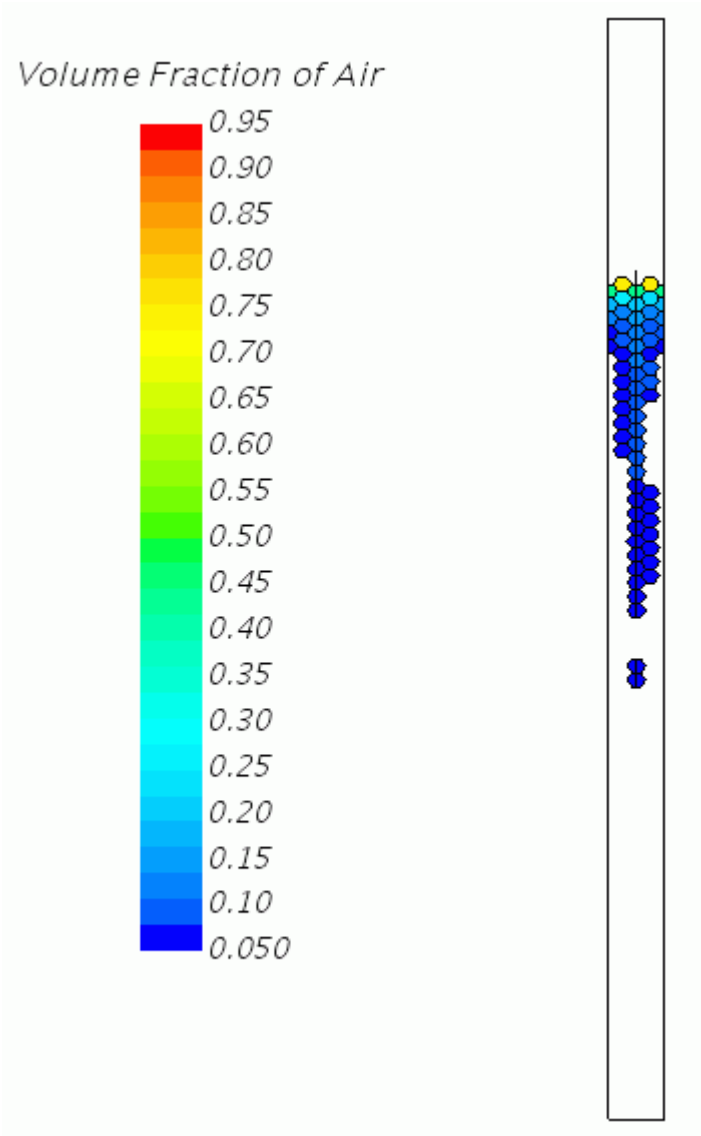
Ventilation in marine applications is the entrainment of air under the hull of a vessel. Numerical ventilation is where this phenomenon occurs erroneously in a simulation due to under-resolution of the mesh, time-step or other numerical issues.

An example is shown below: the entire wetted hull surface is covered by an air layer that has a thickness of approximately 0.3 of the first cell layer height. As air has a lower viscosity than water, the simulation results would show a reduced drag on the hull.



Ventilation is not common in casting applications, but numerical ventilation can occur in a simulation and can have significant effects on the results. For example, in a mold filling simulation, an air layer of one cell thickness isolates the hot melt from the cold mold. This isolation severely affects the heat transfer from the melt to the mold, and therefore affects the solidification time and the occurrence of cold shuts.

An example is shown below: a counter-gravity casting process where the melt flows upstream through a vertical channel has entrained air below the melt surface level. This air layer strongly affects boundary heat transfer.



Using the VOF Phase Replacement Model

The VOF Phase Replacement model lets you specify a limit for every cell (the Phase Replacement Volume Fraction Limit) up to which to replace the unwanted gas phase with the appropriate liquid phase. The VOF Phase Replacement model can also be used to eliminate very thin liquid phases from a wall in gas bulk flow simulations (such as external aero applications).

If appropriate, you can specify a maximum volume fraction in the phase replacement. When you activate the VOF Phase Replacement model, the *Phase Volumetric Replacement Volume Fraction Limit of <phase>* field function becomes available.

This model cannot distinguish between real ventilation and numerical ventilation in a simulation. The model attempts to eliminate a specified VOF phase, replacing it with another specified VOF phase, when the layer thickness decreases below a specified threshold.

The phase replacement is not implemented with source terms to transport equations; instead, some fields are overwritten before the first transport equation is solved within each iteration. For each cell, the VOF Phase Replacement model executes the following actions before all the old variables are updated:

- If the value of the Phase Replacement Volume Fraction Limit is non-zero, the volume fraction of the primary phase is added to the secondary phase up to the limit specified for that cell by the Phase Replacement Volume Fraction Limit.
- The mixture density is updated to match the new volume fraction field.
- The total enthalpy is updated for the primary phase and the secondary phase as well as for the mixture.

This process is important; otherwise the phase replacement would result in contributions to unsteady terms.

The VOF Phase Replacement model is incompatible with the following:

- 2nd order discretization in time

Contents:

VOF Phase Replacement Workflow

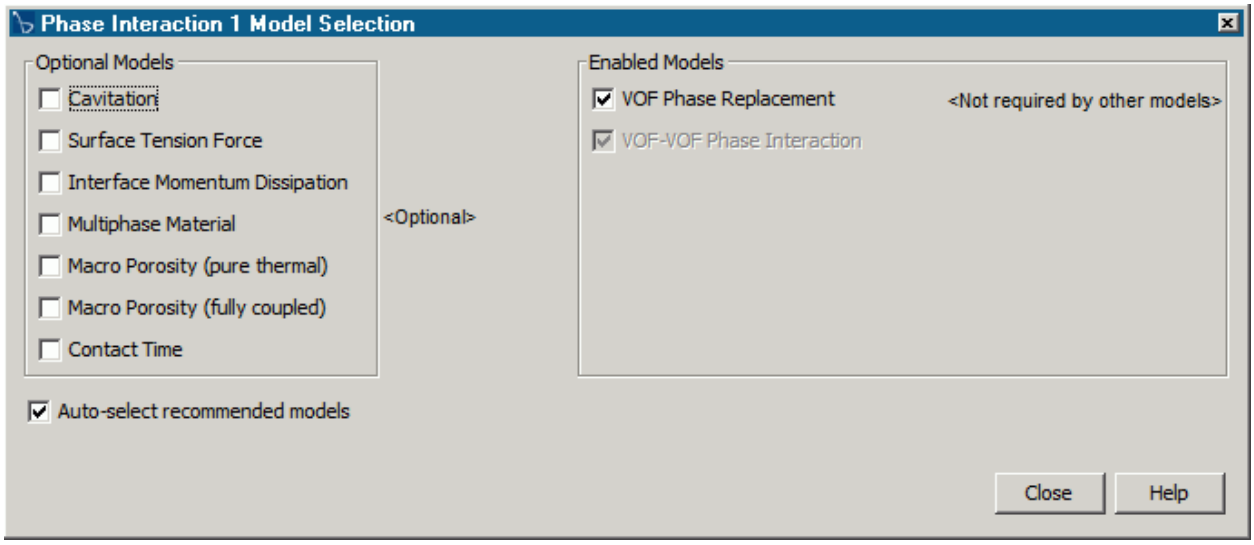
VOF Phase Replacement Workflow

The workflow for setting up VOF phase replacement includes setting up the appropriate phase replacement volume fraction limit, selecting the VOF Phase Replacement model, and specifying the appropriate phases.

The VOF Phase Replacement model lets you specify a limit for every cell (the Phase Replacement Volume Fraction Limit) up to which to replace the unwanted primary phase (usually gas) with the appropriate secondary phase (usually liquid). You specify the phase replacement with a user field function. In some situations, such as bubbles located in the interface between liquid and gas, replacing a bubble (liquid volume fraction of 0) fully with liquid (volume fraction of 1) may not be appropriate. In such scenarios you can specify a maximum volume fraction in the phase replacement field function.

To use the VOF Phase Replacement model in a VOF Multiphase simulation:

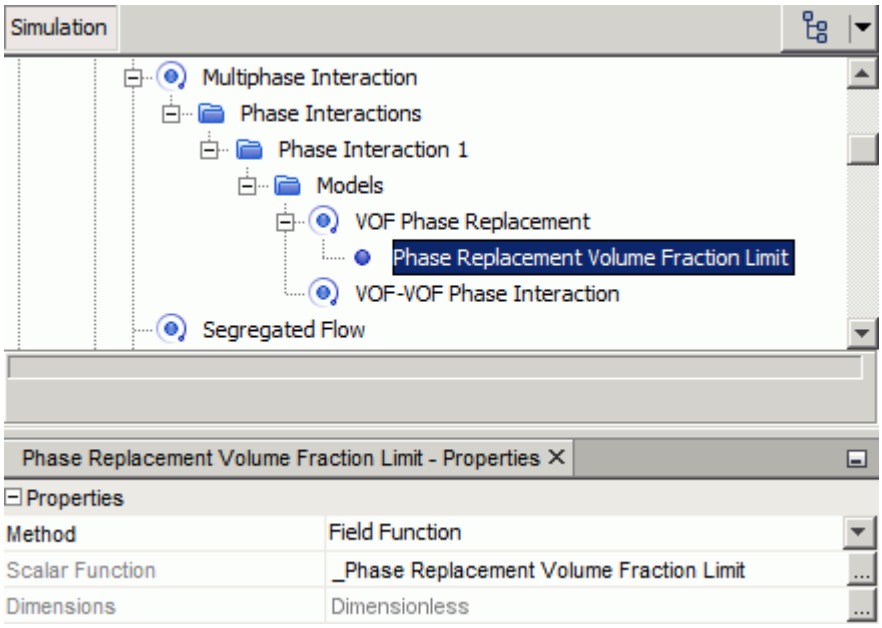
1. Create a new phase interaction.
2. Select the **VOF-VOF Phase Interaction** model and then select the optional **VOF Phase Replacement** model.



3. Select the **VOF-VOF Phase Interaction** node and set the primary phase and the secondary phase.
4. Set up the phase replacement field function.

   The non-zero values of the field function identify the cells in which phase replacement can occur. The volume fraction of phase that is replaced per time step is limited to the specified value.
5. Select the **VOF Phase Replacement** > **Phase Replacement Volume Fraction Limit** node and specify the phase replacement field function.

   Choose the **Field Function** method and set the **Scalar Function** to the user field function that you have set up to specify the appropriate cells. The VOF Phase Replacement model replaces the primary phase by the secondary phase in the relevant cells at the rate specified by the *Phase Replacement Volume Fraction Limit*.



Modeling Surface Tension in VOF Simulations

Immiscibility of two fluids is a result of strong cohesion forces between their molecules and depends on the nature of the fluids. Surface tension always exists between a given pair of fluids. The experimentally determined surface tension coefficient σ expresses the ease with which the fluids can be mixed.

Surface tension plays a fundamental role in many natural and industrial phenomena. Examples can be found in the studies of capillarity, low-gravity fluid flow, hydrodynamic stability, surfactant behavior, cavitation, and droplet dynamics in clouds and in fuel sprays used in internal combustion engines [522].

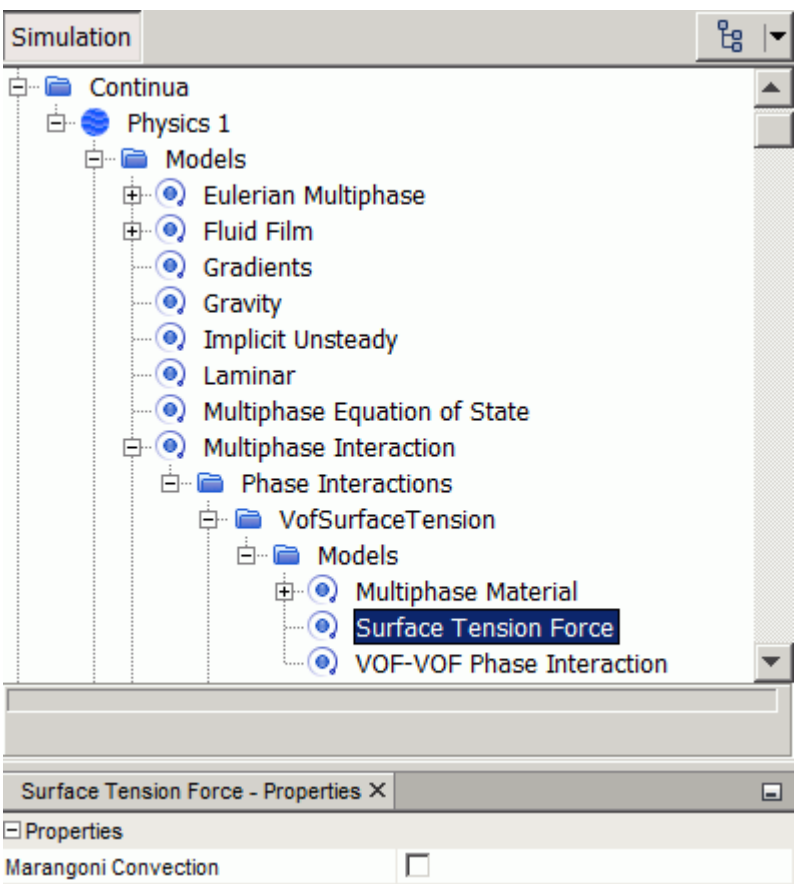
Consider the wetting and spreading phenomena of droplets impacting on a substrate. Inertial, viscous, capillary, and contact-line forces govern the contact line (triple line) dynamics at the free surface between two fluids and a solid ([548]).

The surface tension coefficient σ is defined as the amount of work necessary to create a unit area of free surface.

Calculation of the free surface curvature is sensitive to mesh quality. Therefore, to get good results even on coarse grids, the use of Cartesian meshes is recommended.

VOF Surface Tension Force Model

Surface tension effects are included in a VOF simulation when the Surface Tension Force model is activated in a phase interaction.



You specify the surface tension coefficient σ for each phase interaction as a material property. You also specify the surface tension contact angle at each wall boundary.

By default, the Surface Tension Force model uses a static contact angle to calculate the free surface curvature in the vicinity of the triple line. However, the use of a static contact angle does not satisfy the dynamics of many flows and causes unrealistic free surface shapes in many applications.

In practice, the contact angle is considered a function of the triple line velocity. The triple line is the line where the wall and both fluid phases are in mutual contact. Some simple models are based on a step function that uses a single advancing and receding angle, while other models use a smoothed step function. These models work well for inertia-dominated flows because the dynamic contact angle does not depend strongly on the velocity. For more information, see Modeling a Dynamic Contact Angle.

Other models, such as the Kistler correlation, work well for capillary-dominated flows. Capillarity is a phenomenon in which the free surface is observed to be elevated or depressed when it comes into contact with a solid wall. Capillarity can be explained by considering the effects of adhesion (the attractive or repulsive force between the fluid and solid wall molecules) and cohesion (the attractive force between the fluid molecules). These models attempt to account for the dependency of the advancing and receding contact angles on the triple line velocity. Both static advancing and receding contact angles have been found to increase or decrease further during spreading and recession, respectively. In most cases, the static advancing and receding contact angles can be determined from the contact angle hysteresis: the minimum and maximum contact angle that is measured for a stationary triple line.

Surface Tension Force Properties

Marangoni Convection	Marangoni convection develops along a free surface when a concentration or temperature gradient exists across that interface. These gradients cause the surface tension coefficient to vary across the free surface. These effects can be neglected if the spatial variation of surface tension coefficients is small. If constant surface tension coefficients are set for the phase interaction, activating this property has no effect.	
	Activated	Includes the effect of spatial variations in the surface tension coefficient. The tangential surface tension force Eqn. (838) is accounted for.
	Deactivate	Excludes the effect of spatial variations in the surface tension coefficient.

Surface Tension Material Properties

Surface Tension

Each phase interaction is assigned its own surface tension coefficient (σ in Eqn. (558)). This coefficient is used to calculate the surface tension force between each of the defined phases in the phase interaction. It is entered as a scalar profile.

Surface Tension Model Field Functions

The following field functions are made available to the simulation when the Surface Tension model is activated in a phase interaction.

Capillary Number of [phase interaction]	This scalar field function is defined at wall boundaries and can be used in the definition of a user field function to specify your own user-defined dynamic contact angle correlation.
Contact Angle of [phase interaction]	The surface tension contact angle. This scalar field function is defined at wall boundaries.

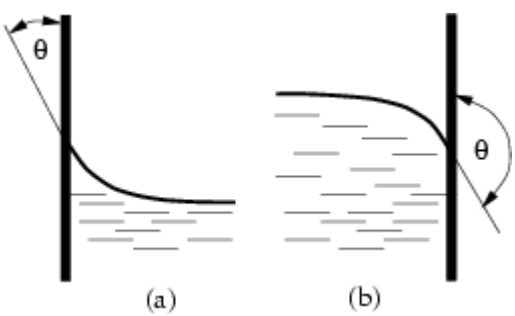
Contents:

What Is a Contact Angle?

Modeling a Dynamic Contact Angle

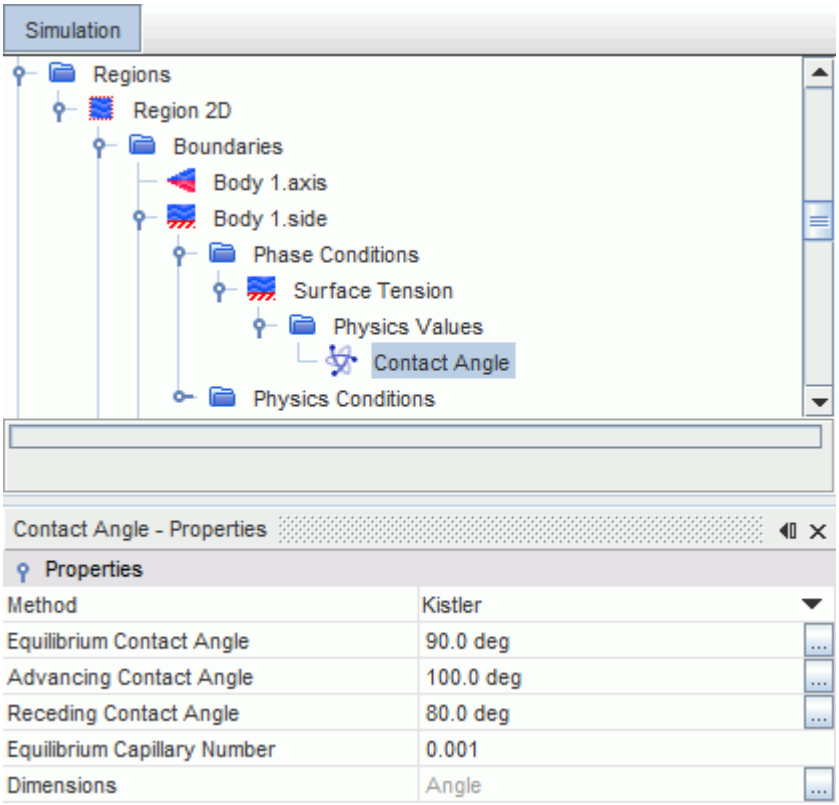
The contact angle θ describes the influence of a solid wall on the free surface. By default, STAR-CCM+ uses a static value of contact angle for calculating the surface tension. In some applications, using a static contact angle can produce unrealistic free surface shapes. For these applications, you can define a dynamic contact angle.

The magnitude of the contact angle for a phase interaction depends on the pair of fluids and the solid that are in contact and on the temperature. The contact angle is measured at the triple line, which is the line where the wall and both fluid phases are in mutual contact. It is assumed that the specified value is with respect to the **Primary Phase** of the phase interaction. Values of contact angle smaller than $90°$ mean that the phase is wetting the wall, as illustrated in part (a) of the following figure:



In order to comply with the usual convention on contact angle values, that are shown above, define a liquid phase as the primary phase of the interaction.

You specify the surface tension contact angle at each wall boundary. The corresponding value node (**Contact Angle**) is activated when the **Surface Tension** option is selected in the *Phase Interaction Model Selection* dialog and after the Phase Interaction has been defined. You specify the contact angle as a scalar profile.



If a static value of contact angle is used for the calculation of surface tension, the contact angle does not change when the flow is advancing or receding. The contact angle is not dependent on the velocity of the contact line. For applications that require a dynamic contact angle, you can use the **Kistler** method to define the contact angle. Alternatively, you can define your own dynamic contact angle correlation with a user field function.

Contact Angle Properties

Method	The method that is used to specify the contact angle.	
	Kistler	Use the **Kistler** method to define a dynamic contact angle. See Dynamic Contact Angle.
	Constant **Field Function** **Table** **User Code**	Use the selected method to define the contact angle. You can use the **Capillary Number** field function in the definition of a user field function to specify your own user-defined dynamic contact angle correlation.
Equilibrium Contact Angle	This value is θ_e in Eqn. (843).	
Advancing Contact Angle	This value is θ_s in Eqn. (841) when the Capillary Number is positive.	

Receding Contact Angle	This value is θ_s in Eqn. (841) when the Capillary Number is negative.
Equilibrium Capillary Number	The range for the equilibrium Capillary number. This range is used when performing the weighted averaging of the dynamic contact angle. This value is Ca_{eq} in Eqn. (844).
Dimensions	Angle (Read-only)

Recommendations and Limitations

The Kistler correlation is suitable for both inertia- and capillary-dominated flows. However, you cannot use a Capillary number-driven dynamic contact angle method at slip walls. The triple line velocity is always evaluated to be 0 at slip walls, and therefore there is no variation in the contact angle. The same restriction applies to modeling the contact angle hysteresis. These simulations are not possible: no variation is observed in the contact angle as, by the definition of this phenomenon, the triple line is not moving.

In some scenarios, oscillations can develop in the interfacial flow field for capillary-dominated flows. These oscillations can be damped or eliminated if a blending function is used in the dynamic contact angle specification. The Kistler method uses a blending function, see Eqn. (843). In addition, you are recommended to activate the Interface Momentum Dissipation model in combination with a dynamic contact angle method in order to remove the parasitic currents at the interface. Specify a value of Interface Artificial Viscosity large enough to reduce parasitic currents, but keep the value as small as possible to avoid affecting the simulation results.

If the interface shows much smearing, and other treatments (such as reducing the time-step size) do not improve the results, you can adjust the Sharpening Factor setting in the Volume of Fluid (VOF) model. Increasing this value from its default of 0 reduces the numerical diffusion at the interface. For surface-tension dominated flows, setting a higher value can improve the resolution of the interface.

Note: Treat both the Interface Artificial Viscosity and Sharpening Factor settings with care, as the additional artificial viscosity and sharpening can result in non-physical behavior.

Contact Time Model Reference

The Contact Time phase interaction model estimates how long two VOF phases have been in contact with each other. A typical application is estimating the probability of surface reactions such as oxidation. The transported quantity is the contact area multiplied by the contact time.

Contact Time Model Reference

Provided By	**[phase interaction]** > **Models** > *Optional Models*
Example Node Path	**[phase interaction]** > **Models** > **Contact Time**
Requires	A VOF Multiphase simulation. A phase interaction with the **VOF-VOF Phase Interaction** model activated. In the *Phase Interaction Model Selection* dialog: *Optional Models*: **Contact Time**

Properties	Key properties are: *Convection*, *Min Volume Fraction*, *Min Interface Fraction*. See Contact Time Model Properties.	
Activates	Model Controls (child nodes)	See Contact Time Condition.
	Field Functions	See Field Functions.

Contact Time Model Properties

Convection

Specifies a convection.

- **1st-order**
  Selects the first-order convection.
- **2nd-order**
  Selects the second-order convection.

Min. Volume Fraction

Time is accumulated only if the volume fraction of the host phase exceeds the specified threshold.

Min. Interface Fraction

Contact time is computed only when interfacial area exceeds the specified fraction of the cube root of the cell volume.

Contact Time Condition

You specify the contact time condition as a scalar profile. Contact time is accumulated where both phases are in contact and the selected profile method returns a nonzero value.

For example, if you want to determine the contact time between an air and a melt phase when the melt phase is in the liquid state, you can use a field function defined as:

```
($SolidVolumeFractionMelt < 1) 1 : 0.
```

Once the solid volume fraction of the melt reaches unity in the simulation, the field function returns zero and the contact time area stops accumulating.

A transport equation is solved to accumulate Contact Time Area, which is the Contact Area multiplied with the Contact Time of the phase.

Field Functions

Contact Area of Phase

The contact area of the phase.

Contact Time Area of Phase

The contact area multiplied by the contact time of the phase.

Contact Time of Phase

The contact time of the phase, which equals Contact Time Area divided by Contact Area.

This is only an approximation of the actual contact time; its accuracy depends on the simulation. For example, consider a fluid rising in a vertical tube that has a decreasing cross-sectional area. When the area is initially large, the contact area multiplied by contact time is also large. At the end, the free surface of the fluid reaches the height with a small cross-sectional area. If the accumulated value of contact area multiplied by contact time is divided by the recent smaller cross-sectional area, the calculated contact time can exceed the simulation time.

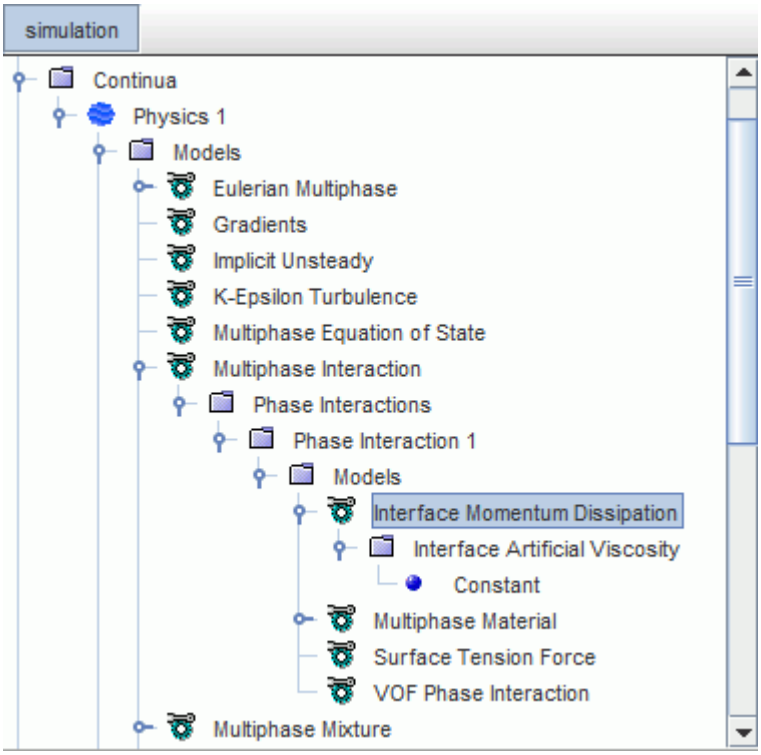
Using the Interface Momentum Dissipation Model

The Interface Momentum Dissipation model adds extra momentum dissipation in the proximity of the free surface to dissipate parasitic currents. The momentum dissipation is proportional to the velocity gradients and the interface artificial viscosity.

Parasitic currents are caused by discretization errors that arise due to solution discontinuities across the free-surface interface. An interface between phases represents a discontinuity in the solution domain. At this interface, all material properties, (for example, density, viscosity, and thermal conductivity) are discontinuous. All gradients of dependent variables, (for example, gradients of velocity, pressure, and temperature), are also discontinuous. In the case of flows that are dominated by surface tension, pressure, too, is discontinuous across the interface.

Discontinuities in solution fields can cause discretization errors which do not scale with the mesh size. In the case of free-surface flows, these errors can cause some spurious or parasitic currents. In many cases these currents are negligible relative to the other transport processes that are present and they do not affect the solution. However, as the parasitic currents can affect the solution in some cases, the Interface Momentum Dissipation model can be used to reduce them.

You access the Interface Momentum Dissipation Model from the *Phase Interaction Model Selection* dialog for the relevant pair of phases.



The Interface Artificial Viscosity node provides a choice between constant or field function for the method that defines the artificial viscosity.

Larger values of interface artificial viscosity are better at removing parasitic currents, but they can affect the physics of the problem on a particular mesh. Ensure that you specify values of interface artificial viscosity which are sufficient to remove or reduce parasitic currents, but are not large enough to affect the quality of the results.

Mass Error Rate Model Reference

The Mass Error Rate model triggers the creation of the **Relative Mass Error Rate** field function. You can use this field function to create a stopping criterion for inner iterations.

You activate the Mass Error Rate model to obtain information about the VOF phase mass conservation per time-step. Numerical mass conservation is related to the convergence of the inner iterations per time-step. If the inner iterations are well converged within a time-step, mass is conserved. However, if convergence of the inner iterations is not fulfilled for the time-step, mass might not be conserved and a mass error occurs.

See Setting Up Automatic Inner Iterations.

The relative mass error is computed as:

$$\epsilon = \left| \frac{m_{Imbalance}}{m_{tot}} \cdot 100 \right| \tag{809}$$

where $m_{Imbalance}$ is the sum of phase mass imbalance in the entire computational domain and m_{tot} is the total phase mass. The mass imbalance is computed when solving the transport equations. For more information, see General Transport Equation.

Mass Error Rate Model Reference

Provided By	**[phase]** > **Models** > *Optional Models*	
Example Node Path	**[phase]** > **Models** > **Mass Error Rate**	
Requires	A VOF Multiphase simulation. A Eulerian phase. In the *Phase Model Selection* dialog: *Optional Models*: **Mass Error Rate**	
Properties	None.	
Activates	Field Functions	*Relative Mass Error Rate of [Phase]* See Field Functions.

Field Functions

Relative Mass Error Rate of [Phase]

The relative mass error rate of the phase as given by Eqn. (809).

Wall Porosity

The wall porosity model lets you define phase-specific porous boundaries within a VOF Multiphase simulation. That is, it permits material that belongs to single-component phases to pass through selected wall boundaries.

A typical example for application of the wall porosity model is an investment casting or sand casting simulation. For these types of applications, liquid melt is injected into the domain that is initially filled with air. While the liquid melt is injected producing high pressures, air leaves the domain through the porous mold. When the melt solidifies, shrinkage of the solidified melt can produce low pressures in the domain. These low pressures can lead to air being sucked back into the domain. In STAR-CCM+, you model this behavior using the wall porosity model, which allows the selected phase, typically a gas phase, to leave and to enter the domain based on pressure distribution. For each phase, you can control whether it leaves through a particular boundary or not.

Use the wall porosity model only:

- if the mold is modeled as a solid region (and not as a porous fluid region).
- if the flow through the porous boundary can be considered a creeping flow.

  For fluid entering the domain, turbulent kinetic energy and turbulent dissipation rate are not taken into account. Also, the fluid enters the domain at the temperature of the phase at the boundary.

The velocity across the porous boundary is calculated from the known pressure drop across it.

This pressure drop is given as:

$$\Delta p = -\rho\left(\alpha\left|v_n\right| + \beta\right)v_n \tag{810}$$

where ρ is the density of the phase, α and β are user supplied constants and v_n is the normal velocity across the porous boundary. STAR-CCM+ calculates the pressure drop from a specified external ambient pressure on the region and the pressure in the cell next to the porous wall. Eqn. (848) is then applied to calculate the (virtual) normal outflow velocity. This normal velocity is multiplied with phase density, the volume fraction, and the area of the boundary face to predict the phase mass flux leaving through the boundary face within a given time-step.

If a phase leaves the domain, the volume fraction is known. Under inflow conditions for multiple phases, the volume fraction is unknown as it is not solved for in the solid region. Therefore, the volume fraction must be assumed when fluid enters the domain through a porous wall boundary. It is assumed that phases are equally distributed: the volume fraction is equal to 1 for one phase, it is equal to 0.5 for two phases, it is equal to 0.333 for three phases, and so on.

Contents:

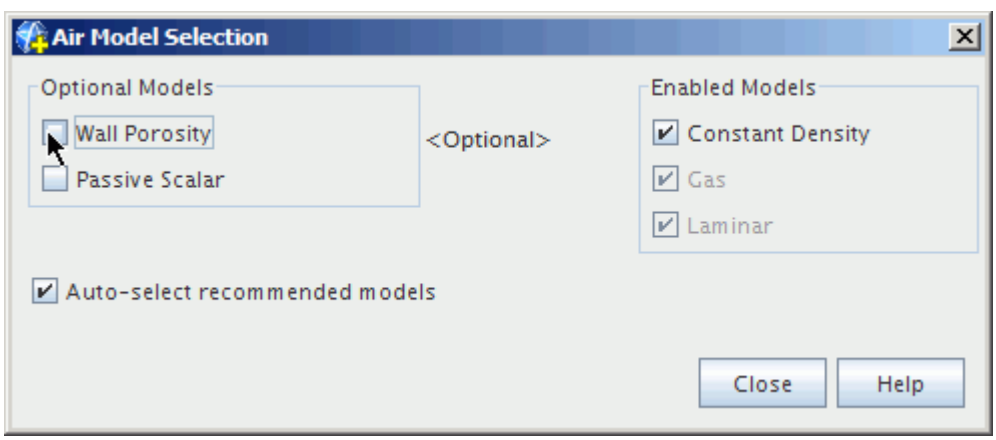
Selecting the Wall Porosity Model
Wall Porosity Properties
Setting up the Wall Porosity Model
Setting Wall Porosity Coefficients at Walls and Contact Interfaces
Setting Wall Porosity Coefficients at Porous Baffle Interfaces
Wall Porosity Field Function Reference

Selecting the Wall Porosity Model

This section describes how to solve problems that involve VOF phase-specific wall porosity.

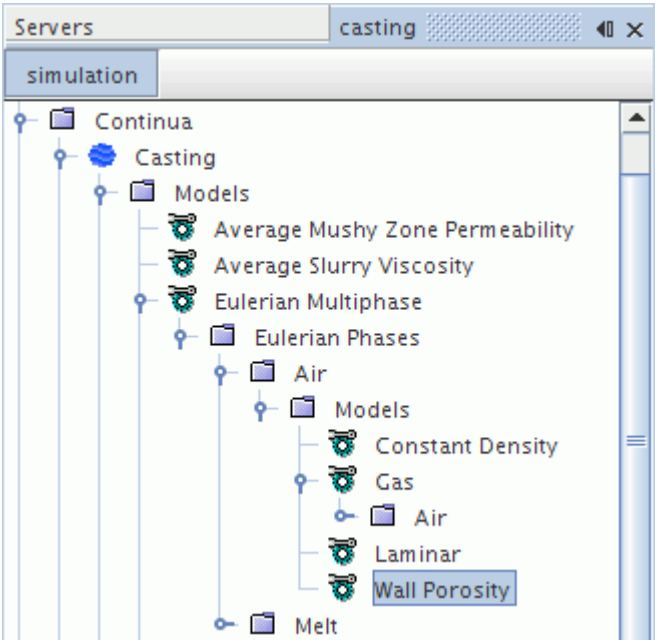
To select the Wall Porosity model:

1. Select the physics models that are required for a VOF multiphase simulation.
2. Set up a Eulerian phase.
3. In the *Phase Model Selection* dialog, choose **Wall Porosity** in the *Optional Models* group box.



4. Click **Close**.

   The **Wall Porosity** node appears in the simulation tree.



5. If necessary, select the **Wall Porosity** node and set the appropriate Max. Mass Extraction Factor.

   Ensure that at least one phase of the multiphase mixture is compressible when using the wall porosity model. Otherwise, mass extraction in a completely enclosed region (for example enclosed by wall boundaries or flow-stopped cells) leads to infinite negative absolute pressures causing a floating point exception.

Wall Porosity Properties

Wall Porosity Expert Property

Max. Mass Extraction Factor	Specifies the maximum mass fraction that can be removed from a cell. The default is 0.5. This property prevents the model from extracting more mass from a cell than is available in that particular cell. You are advised that you do not change the value of this property unless you are thoroughly familiar with the discretization techniques used in STAR-CCM+.

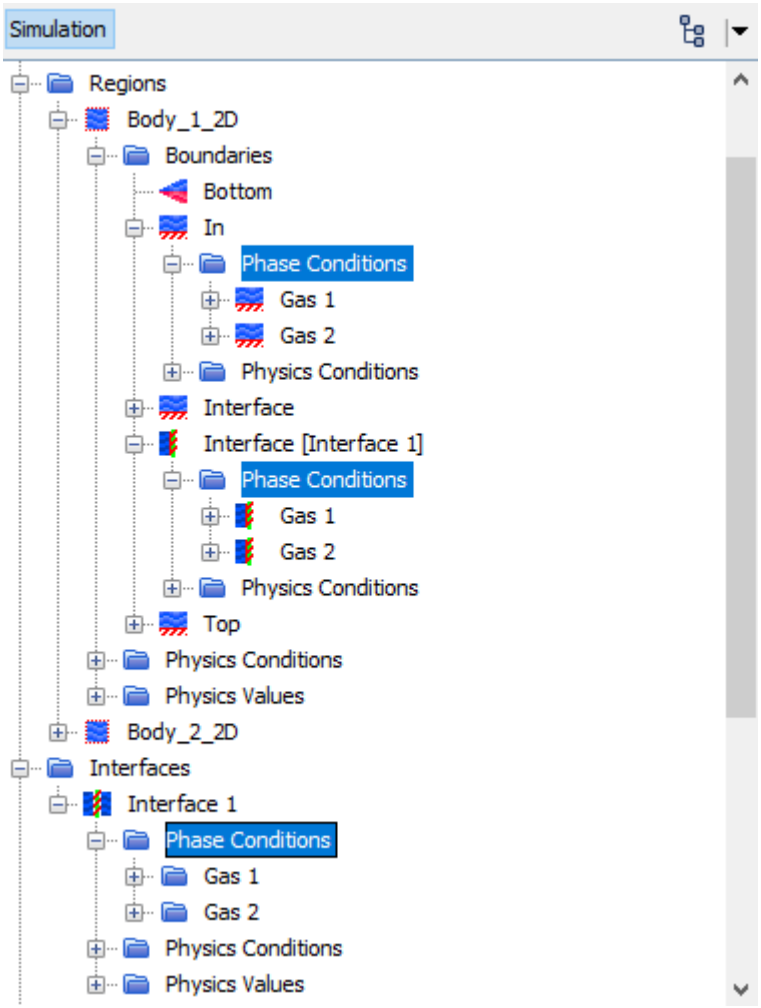
Setting up the Wall Porosity Model

To define the phase-specific wall porosity, several parameters must be set.

The parameters are:

- Ambient pressure
- Porous inertial resistance coefficient
- Viscous inertial resistance coefficient

Once the wall porosity model has been selected, **Phase Conditions** nodes appear within the boundaries that are associated with the physics continuum. If a baffle interface exists in the model, a **Phase Conditions** node is also added underneath the Interfaces node.



Do the following:

- To specify the ambient pressure and the porous inertial and viscous resistance coefficients for the porous wall boundaries and contact interfaces, use the boundary **Phase Conditions** node.
- To specify the ambient pressure and the porous inertial and viscous resistance coefficients for the porous baffle interface, use the **Interfaces** > **Phase Conditions** node.

The porous inertial resistance coefficient α and the porous viscous coefficient β are added to all the wall boundaries in regions for which the wall porosity model is active. The wall porosity model has no effect in either of the following scenarios:

- The porous inertial and viscous resistance coefficients are both zero (the default values).
- The ambient pressure is set to the **Pressure** field function, hence forcing Δp to be always zero.

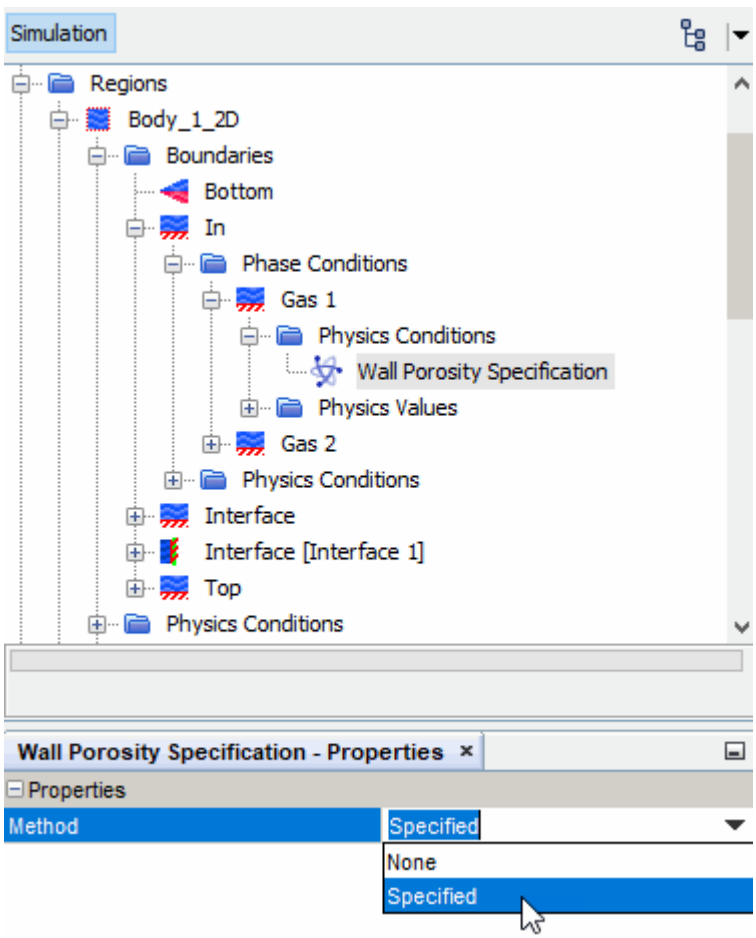
Setting Wall Porosity Coefficients at Walls and Contact Interfaces

The porous inertial resistance coefficient α and the porous viscous coefficient β determine the pressure drop across the porous boundary. The ambient pressure is also used to calculate the pressure drop Δp from which the phase extraction mass is calculated.

For more information on calculating the pressure drop, see Eqn. (848).

To set the ambient pressure and the wall porosity coefficient values at a phase-specific porous wall boundary or contact interface:

1. Select the **Regions** > **[Region]** > **Boundaries** > **[Boundary]** > **Phase Conditions** > **[Phase]** > **Physics Conditions** > **Wall Porosity Specification** node.
2. In the *Wall Porosity Specification - Properties* window, set the *Method* to **Specified**.



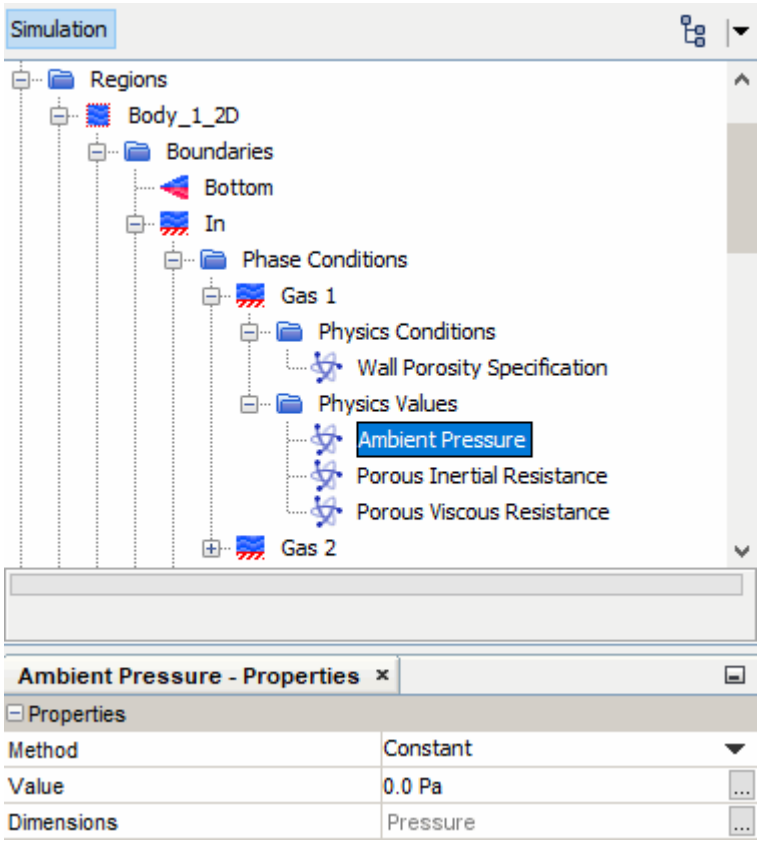
A **Physics Values** node is added to the simulation tree.

3. Open the **Physics Values** node and set the scalar profile method and the corresponding value for the following:

Ambient Pressure	The ambient pressure is specified relative to the reference pressure. It is not an absolute value.

Porous Inertial Resistance	The porous inertial resistance coefficient α in Eqn. (848).
Porous Viscous Resistance	The porous viscous coefficient β in Eqn. (848).

The example below is the **Ambient Pressure** node.



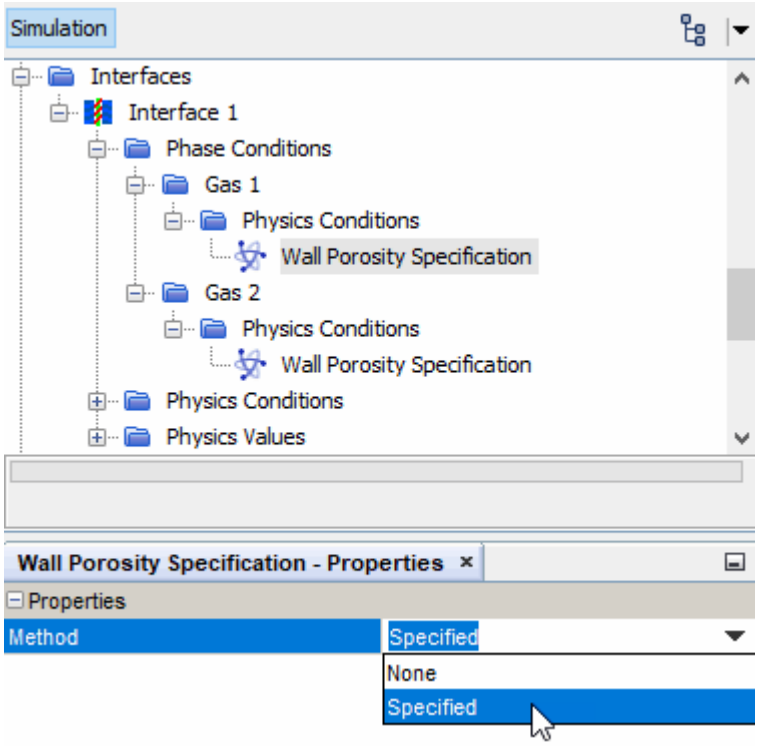
Setting Wall Porosity Coefficients at Porous Baffle Interfaces

The porous inertial resistance coefficient α and the porous viscous coefficient β determine the pressure drop across the porous baffle. The ambient pressure is also used to calculate the pressure drop Δp from which the phase extraction mass is calculated.

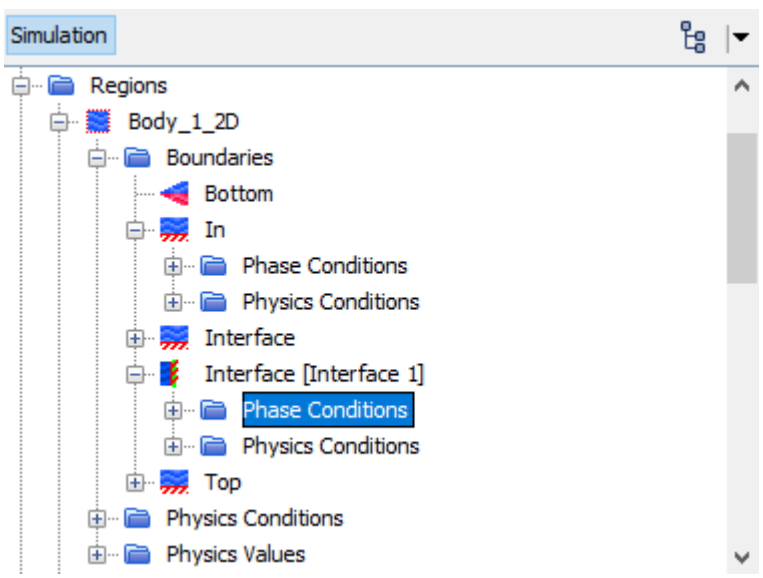
For more information on calculating the pressure drop, see Eqn. (848).

To set the wall porosity coefficient values at a phase-specific porous baffle interface:

1. Select the **Interfaces** > **[Interface]** > **Phase Conditions** > **[Phase]** > **Physics Conditions** > **Wall Porosity Specification** node.
2. In the *Wall Porosity Specification - Properties* window, set the *Method* to **Specified.**



A **Phase Conditions** node is added under **Regions** > **[Region]** > **Boundaries** > **[Interface]**.

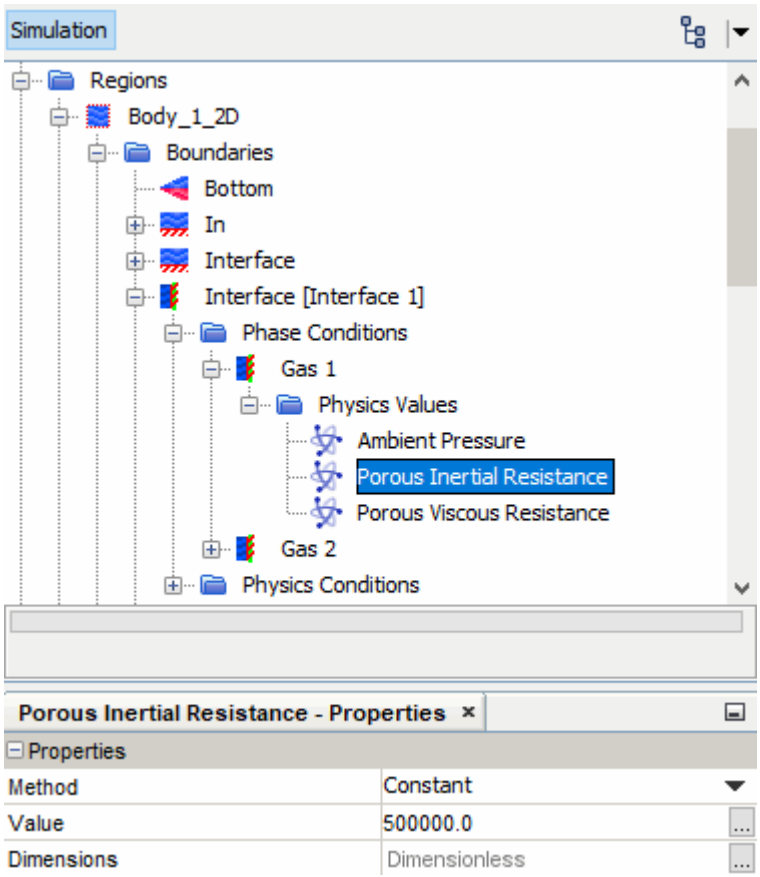


3. Open the **Physics Values** node and set the scalar profile method and the corresponding value for the following:

Ambient Pressure	The ambient pressure is specified relative to the reference pressure. It is not an absolute value.
Porous Inertial Resistance	The porous inertial resistance coefficient α in Eqn. (848).

Porous Viscous Resistance	The porous viscous coefficient β in Eqn. (848).

The example below is the **Porous Inertial Resistance** node.



Wall Porosity Field Function Reference

This section describes the field functions that are available for every Eulerian single-component gas phase with the wall porosity model selected.

These field functions all share base properties.

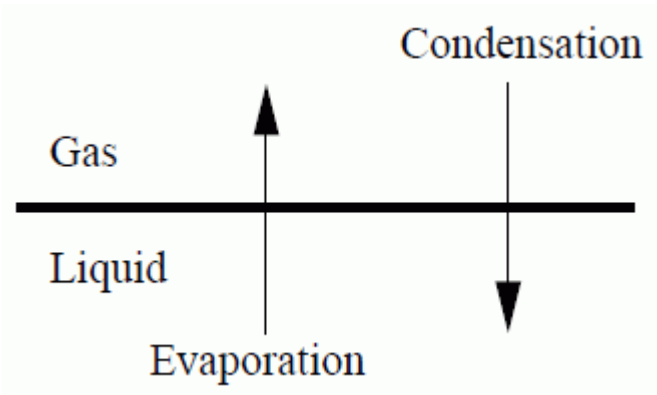
VOF Porous Wall Flux	The actual mass flow per unit area through the porous wall (kg/m^{2}s).
VOF Porous Wall Target Normal Velocity	The target velocity normal to the porous wall v_n calculated according to (Eqn. (848)) and the pressure drop Δp . This function is not the actual normal velocity, which is equal to **VOF Porous Wall Flux** divided by the phase density.

Modeling Evaporation and Condensation

The particular phase-change type that is described in this section concerns evaporation and condensation.

What are Evaporation and Condensation?

Evaporation is a form of vaporization, the phase change from liquid to gas, which occurs when liquid molecules or atoms with high enough kinetic energy escape through the interface between the two phases. Condensation is the reverse process of vaporization and is the phase change from gas to liquid.



The **Evaporation/Condensation** model deals with evaporation (without boiling) and condensation taking place at the free surface interface between a liquid and gas phase within the context of the Volume Of Fluid method. Both phases are modeled as multi-component mixtures that can include non-interacting (inert) species [547], [546]. Evaporation and condensation are treated as hydrodynamically limited, meaning that:

- The phases are considered to be in equilibrium at the interface, and
- The driving force is the diffusion of species.

The grid must be sufficiently fine to model the transport of species away from and toward the interface accurately.

Raoult's law is used to describe the phase equilibrium.

Evaporation/Condensation Properties

Connectivity	Identifies the evaporating/condensing pairs in the liquid and the gas phases.

Contents:

Setting Up a Simulation with Evaporation and Condensation
VOF Evaporation/Condensation Field Functions Reference

Setting Up a Simulation with Evaporation and Condensation

This section describes how to set up a simulation with evaporation and condensation.

To use STAR-CCM+ for solving problems that involve evaporation and condensation:

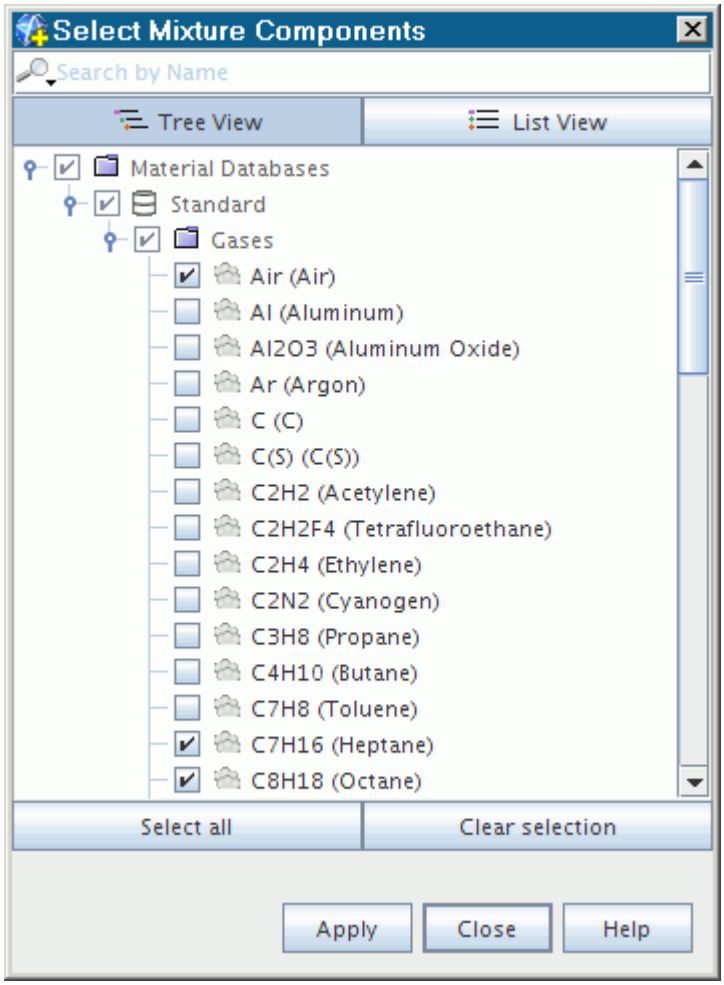
1. Select the physics models that are required for a VOF multiphase simulation.
2. Define the multi-component Eulerian phases.
3. Select the Evaporation/Condensation model in a phase interaction.

Defining Evaporation and Condensation Mixture Components

For a simulation with evaporation and condensation in STAR-CCM+, create two Eulerian Phases. The primary Eulerian Phase must be a liquid, and the other phase must be a gas. Both Eulerian Phases must be multi-component phases.

4. Create a **Eulerian Phase**, and select **Multi-Component Gas** from the *Material* group.
5. Select **Segregated Species** from the *Optional Models* group and then select an appropriate model from the *Equation of State* group.
6. Right-click the **Multi-Component Gas** > **Gas Components** node and choose **Select Mixture Components...**
7. In the *Select Mixture Components* dialog which appears, select the gas components required.

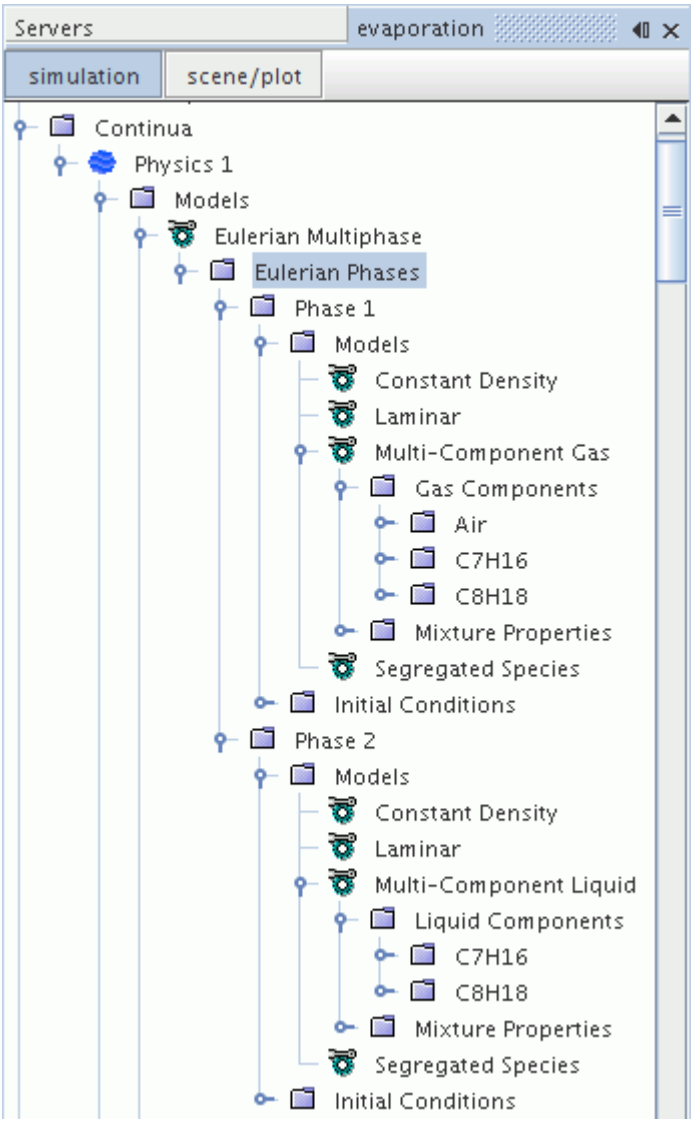
   The Evaporation/Condensation model in STAR-CCM+ requires the gas mixture to include at least one inert component. However, the mass fraction of this inert phase can be very small; typically 1e-3 or 1e-4, but a mass fraction of 0 is still valid. In the example below, Air, Heptane, and Octane are selected, and Air is the inert component.



8. Repeat the creation steps for a second phase where evaporation is likely to occur, but select **Multi-Component Liquid** as the material.
9. Select only the liquid components in the *Select Mixture Components* dialog.

   In this example, Heptane and Octane are selected.

   The simulation tree appears as shown below.



The saturation pressure is available from the **Material Properties** of each liquid component. The latent heat of vaporization is incorporated automatically through the heat of formation property, available for both gas and liquid components.

The final step in setting up an evaporation/condensation case is to define the evaporation/condensation phase interaction.

Selecting the Evaporation/Condensation Model from a Phase Interaction

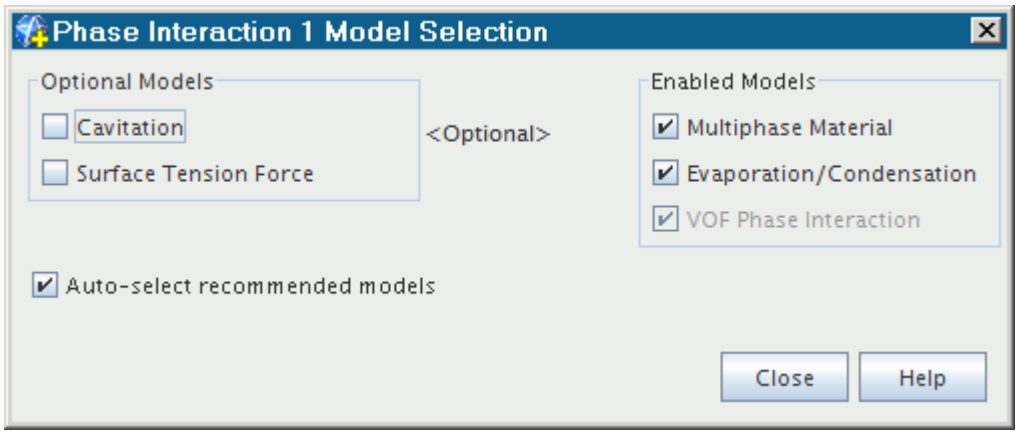
The Evaporation and Condensation model can be applied only to a single phase interaction where you define all of the liquid and vapor phase components. You cannot define the liquid and vapor phase components for the Evaporation and Condensation model in multiple phase interactions.

To select the Evaporation and Condensation model:

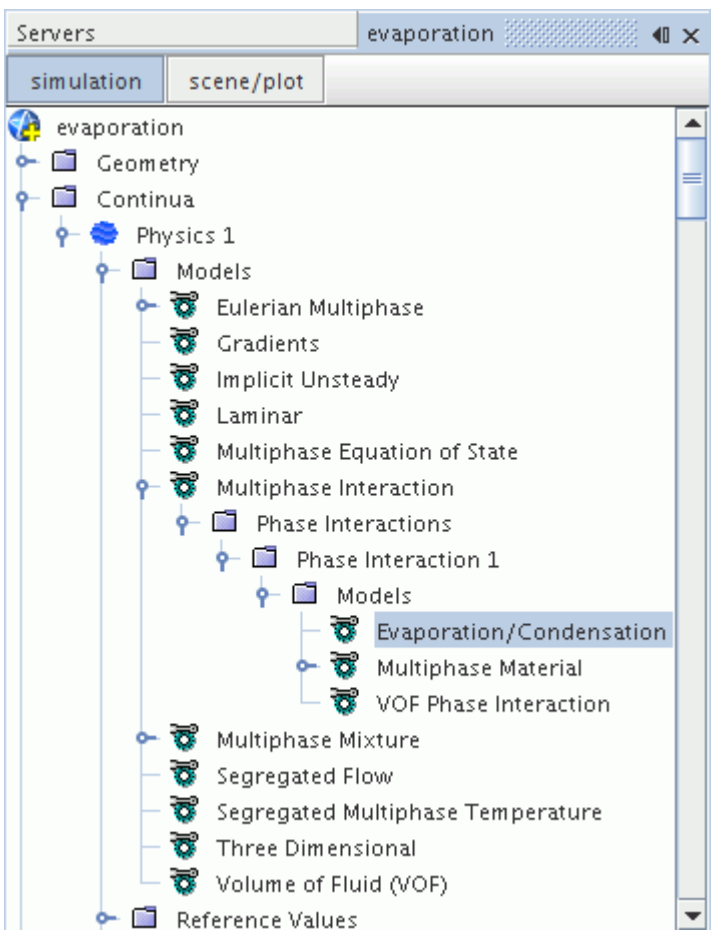
10. Create a phase interaction involving both Eulerian phases.
11. Select the VOF Phase Interaction model in the *Phase Interaction Model Selection* dialog.
12. Set the properties of the **VOF Phase Interaction** node.

The *Primary Phase* must be the multi-component liquid and the *Secondary Phase* must be the multi-component gas. The phases must be assigned before the **Evaporation/Condensation** model becomes available in STAR-CCM+.

13. Reopen the *Phase Interaction Model Selection* dialog.
14. Choose the **Evaporation/Condensation** model and click **Close**.

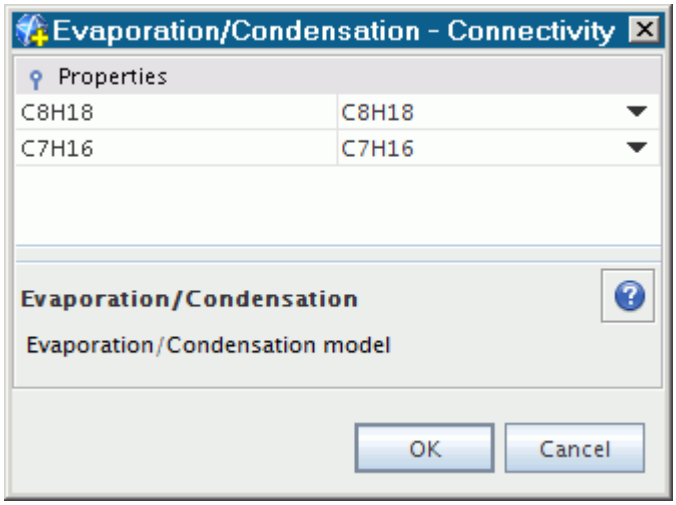


The **Evaporation/Condensation** node appears in the simulation tree.



Set Component pairs using the properties of the **Evaporation/Condensation** model node:

15. Select the **Evaporation/Condensation** model node.
16. Click the ellipsis to the right of the *Connectivity* property in the *Properties* window.
17. In the *Connectivity* dialog that appears, match the liquid and vapor phase components using the drop-down menus provided.



VOF Evaporation/Condensation Field Functions Reference

This section describes the field functions that are available for every interacting gas components: the evaporation rate, the evaporation latent heat, and the evaporation mass fraction.

These objects all share base properties.

Evaporation Rate	The rate of evaporation in kg/s for every cell.
Evaporation Latent Heat	The latent heat of vaporization in J/kg if evaporation/condensation takes place in the diagnosed cell. Available when *Temporary Storage Retained* is selected for the **Segregated Volume Fraction** solver.
Evaporation Mass Fraction	The saturated vapor mass fraction when evaporation/condensation takes place in the diagnosed cell. Available when *Temporary Storage Retained* is selected for the **Segregated Volume Fraction** solver.

Modeling Boiling

The particular phase-change type that is described in this section concerns boiling.

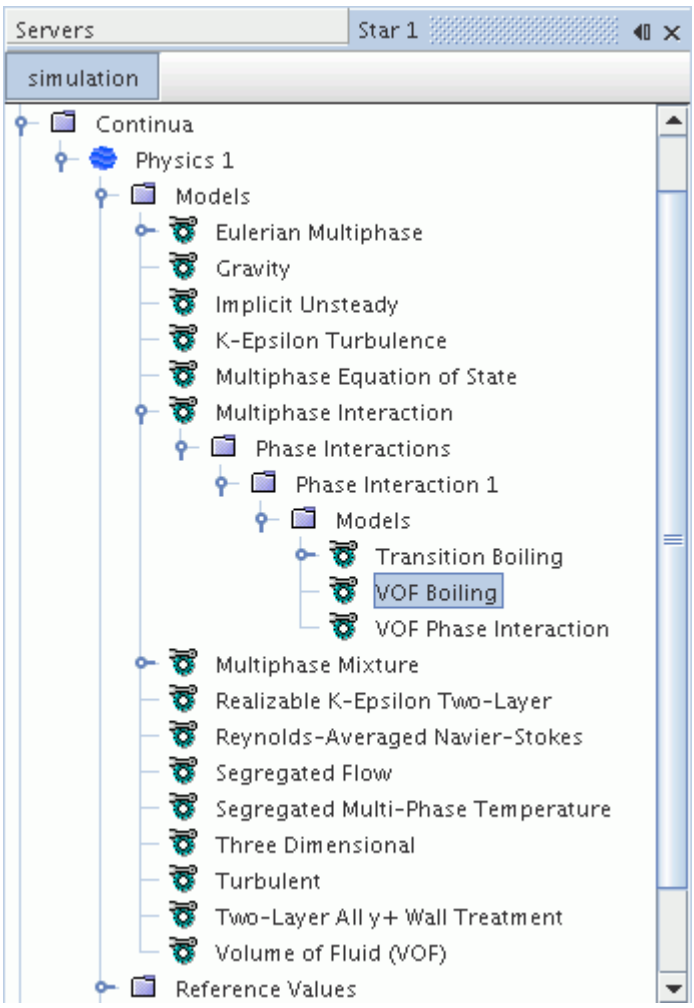
What Is Boiling?

Boiling is a rapid vaporization of a liquid. It typically takes place when a liquid is heated to the boiling point (saturation temperature of the liquid T_{sat}). Its saturation vapor pressure then becomes equal to or larger than the pressure of the surrounding liquid.

Vapor can form from a liquid in the following locations:

- At a vapor-liquid interface away from surfaces
- At a solid surface with pre-existing vapor or gas pockets
- In the bulk of the liquid due to density fluctuations

The VOF Boiling model is the gateway, or entry point to the boiling modeling capabilities of STAR-CCM+. The **VOF Boiling** node represents this model and is subject to common menu items.



If you are only interested in modeling the improved heat transfer due to boiling, you can simulate boiling with a single-component liquid.

Contents:

Selecting a Simulation with Boiling

Defining Boiling Mixture Components
Selecting the Boiling Models and Defining Phase Interactions
What Is the Rohsenow Boiling Model?
What Is the Transition Boiling Model?
What is the HTCxArea?
Setting Boundary Conditions
What Is the Film Boiling Vapor Volume Fraction?

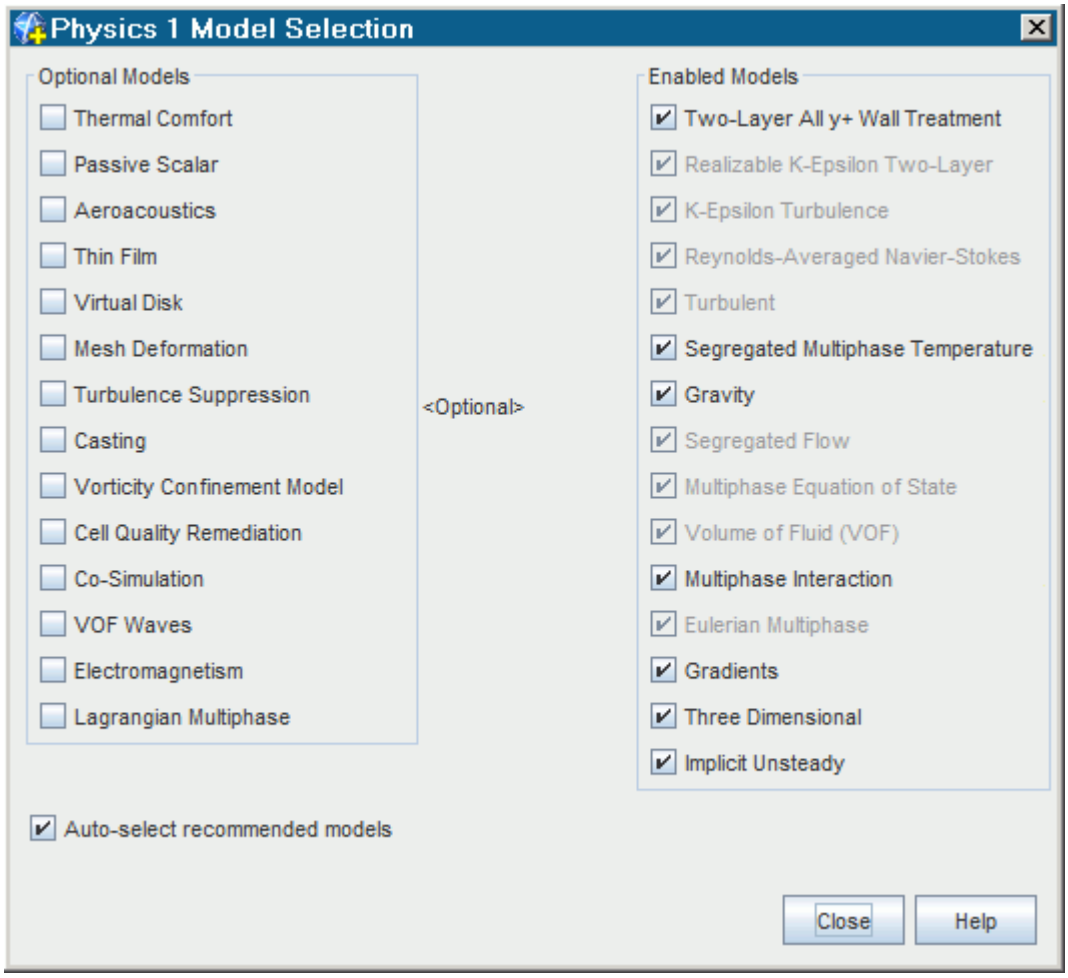
Selecting a Simulation with Boiling

This section describes the models that you select to use STAR-CCM+ for solving problems that involve boiling.

You can then select the boiling models in the **Multiphase Interaction** > **Phase Interactions** node:

1. In the *Physics Model Selection* dialog, specify the material model as a multiphase fluid by selecting **Eulerian Multiphase** in the *Material* group box.
2. Select **Volume of Fluid (VOF)** from the *Eulerian Multiphase Model* group box.
3. Select the Implicit Unsteady solver option from the *Time* group box.
4. Make an appropriate selection from the *Viscous Regime* group box.
5. In the *Optional Models* group box, select **Gravity** and **Segregated Multiphase Temperature**.

   The *Physics Model Selection* dialog then looks as shown below.



6. Make any further appropriate selections from the other optional physics model groups.

The next step is to define the boiling mixture components.

Defining Boiling Mixture Components

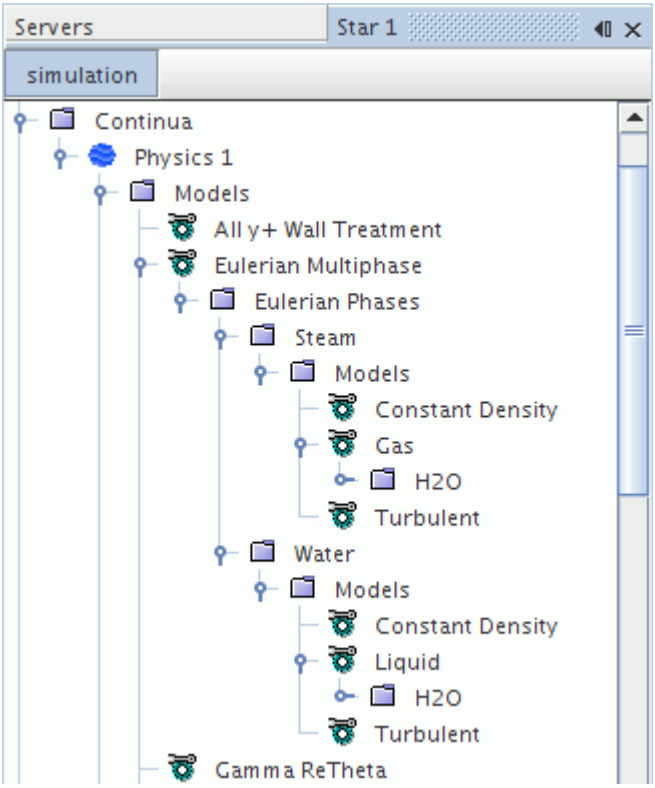
In a simulation with boiling, STAR-CCM+ automatically sets up a default immiscible fluid mixture. You define the bulk properties and phase composition of the mixture.

Each phase must be single-component (that is, pure liquid and pure vapor). The VOF Boiling model does not support multi-component phases.

To define the boiling mixture components:

1. Create a Eulerian phase and select the liquid material model.
2. If necessary, replace the material, such as H2O in the example below.

   STAR-CCM+ reads the material properties from its material database.
3. Create a second Eulerian phase and select the gas material.
4. Replace the default material so that the second Eulerian phase represents the gaseous form of the liquid phase.



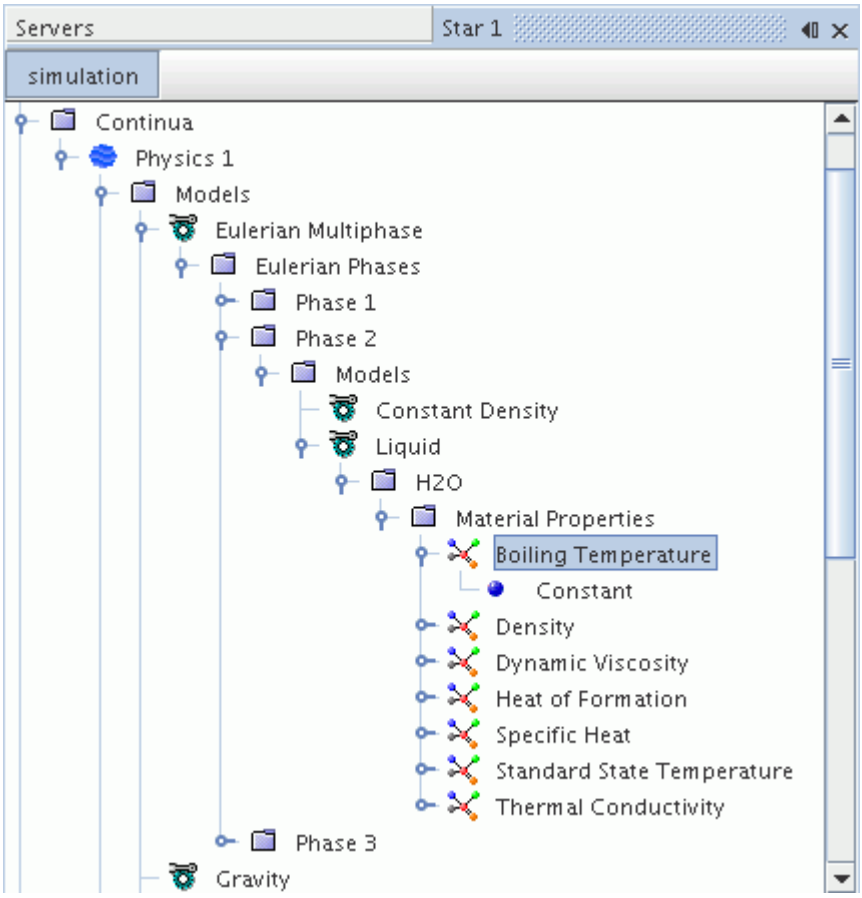
The continuum must consist of at least two phases, that is the boiling liquid and its vapor. You can add any number of additional phases to complete the physics set-up.

The boiling model has to know the boiling temperature for the liquid phase and the latent heat of vaporization. The latent heat of vaporization is the difference between the enthalpies of the vapor and the liquid at the saturation temperature.

The final step in setting up a boiling case is to define the boiling phase interactions.

What Is Boiling Temperature?

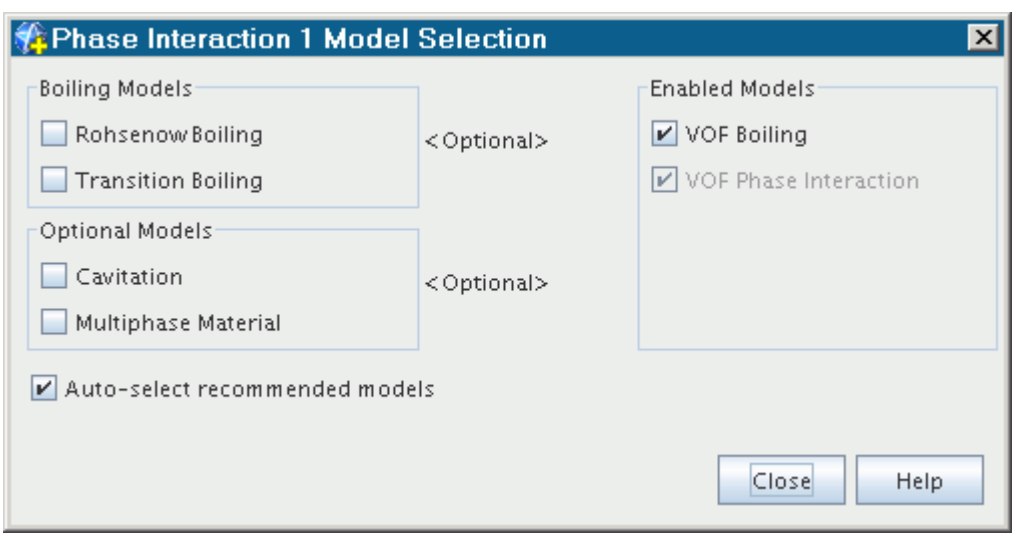
The boiling or saturation temperature T_{sat} of a substance is the temperature at which it can change its state from a liquid to a gas throughout the bulk of the liquid at a given pressure. It is only available once you have set up a boiling phase interaction.



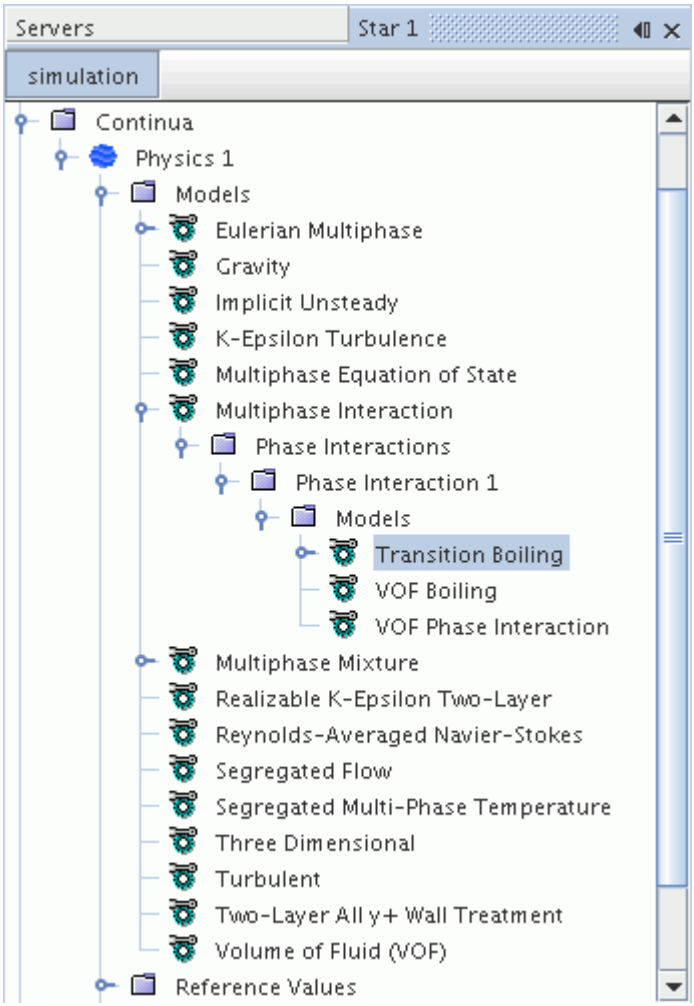
Selecting the Boiling Models and Defining Phase Interactions

The *Phase Interaction Model Selection* dialog lets you select the boiling models: the **VOF Phase Interaction** model and the **VOF Boiling** model.

The Rohsenow Boiling and Transition Boiling models are available for selection in the *Boiling Models* group.



The boiling models appear in the **Multiphase Interaction** > **Phase Interactions** > **Phase Interaction 1** > **Models** node. The Transition Boiling model is shown below.

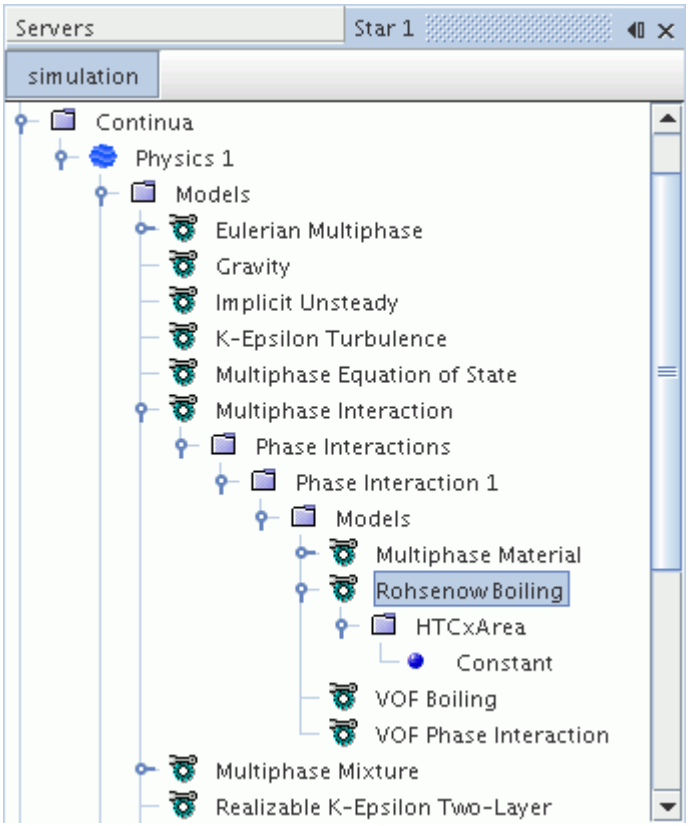


The phase interaction is defined in the *Properties* window of the **VOF Phase Interaction** node. The liquid phase corresponds to the *Primary Phase* and the gas phase corresponds to the *Secondary Phase*.

What Is the Rohsenow Boiling Model?

When Rohsenow Boiling is selected in a simulation, STAR-CCM+ creates the **Rohsenow Boiling** node in the simulation tree.

An example is shown below:



To specify the Rohsenow Boiling model properties, use the *Properties* window of this node.

Rohsenow Boiling Model Properties

C_qw	The empirical coefficient C_{qw} in the Rohsenow expression for the wall heat flux, Eqn. (313). Varies with the liquid-surface combination.
C_ew	The fraction C_{ew} of the wall heat flux due to boiling that is used in creation of vapor bubbles. Used in Eqn. (314).
Sc_t	The turbulent Schmidt number that is associated with the turbulent diffusion of vapor bubbles. Used in Eqn. (350).

Rohsenow Boiling Model Expert Properties

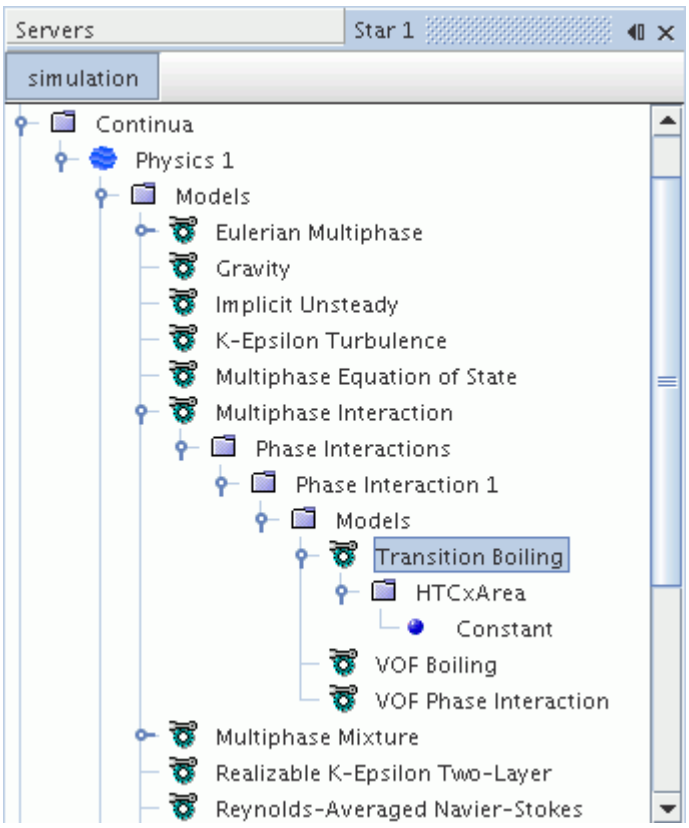
Resolve liquid-vapor interface	The boiling model assumes that the vapor bubbles are smaller than the mesh size and so the interface between the liquid and the vapor phase is not resolved. If, however, large clouds of vapor are expected, and the mesh is sufficiently fine that the interface between the liquid and the vapor phase can be resolved, activate this flag.	
	Activated	The liquid-vapor interface is resolved.
	Deactivated	The liquid-vapor interface is not resolved.

n_p	The Prandtl number exponent n_p (1.73 by default) in Eqn. (313).
URF_whf	Under-relaxation factor for heat flux due to boiling, at walls with prescribed heat flux.

What Is the Transition Boiling Model?

When Transition Boiling is selected in a simulation, STAR-CCM+ creates a **Transition Boiling** node in the simulation tree.

An example is shown below:



To specify the Transition Boiling model properties, use the *Properties* window of this node.

Transition Boiling Model Properties

C_ew	The fraction C_{ew} of the wall heat flux due to boiling that is used in creation of vapor bubbles. Used in Eqn. (314).
Sc_t	The turbulent Schmidt number that is associated with the turbulent diffusion of vapor bubbles.
q_max	The positive empirical constant $q_{\max}$ in the Transition Boiling model, Eqn. (862) to Eqn. (864).

k1	The positive empirical constant K_1 in the Transition Boiling model, Eqn. (862) to Eqn. (864).
delT1	The positive empirical constant ΔT_1 in the Transition Boiling model, Eqn. (862) to Eqn. (864).
k2	The positive empirical constant K_2 in the Transition Boiling model, Eqn. (862) to Eqn. (864).
delT2	The positive empirical constant ΔT_2 in the Transition Boiling model, Eqn. (862) to Eqn. (864).
phi	The constant ϕ in the Transition Boiling model, Eqn. (862) to Eqn. (864).

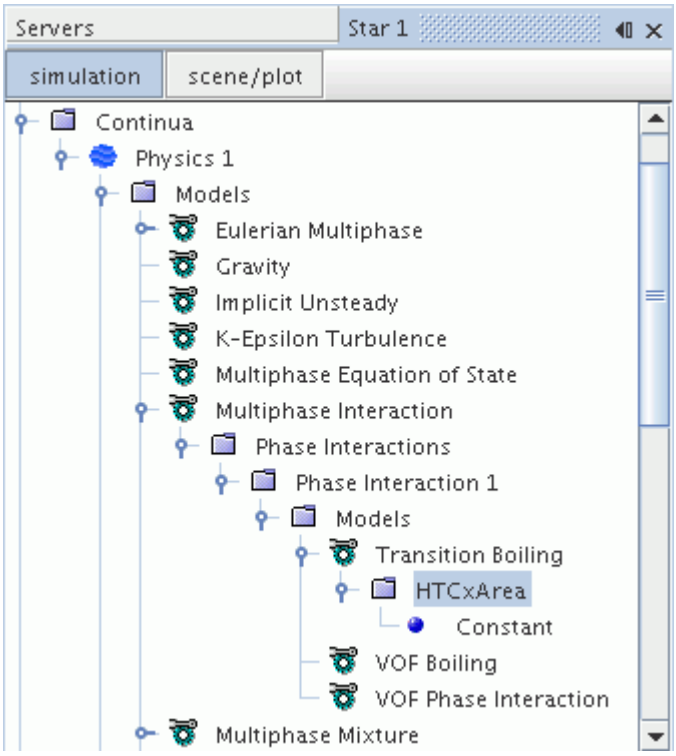
Transition Boiling Model Expert Properties

Resolve liquid-vapor interface	The boiling model assumes that the vapor bubbles are smaller than the mesh size and so the interface between the liquid and the vapor phase is not resolved. However, if large clouds of vapor are expected, and the mesh is sufficiently fine that the interface between the liquid and the vapor phase can be resolved, activate this flag.	
	Activated	The liquid-vapor interface is resolved.
	Deactivated	The liquid-vapor interface is not resolved.
URF_whf	Under-relaxation factor for heat flux due to boiling, at walls with prescribed heat flux.	

What is the HTCxArea?

The **HTCxArea** node represents the heat transfer coefficient (W/m^2K) between the vapor bubbles and the surrounding liquid, which is multiplied by the specific contact area (contact area per unit volume, m^2/m^3) between the two.

The **HTCxArea** node has properties and appears as a child of the **Rohsenow Boiling**, **Transition Boiling** or Fluid Film **Boiling** model node.



HTCxArea Properties

Method	Selects the method that is used to enter the HTCxArea coefficient.	
	Constant	The coefficient is entered as a constant in W/m^3K . A Constant node is added as a child to this node.
	Field Function	The coefficient is entered as a field function that returns a value in W/m^3K . A field function method node is added as a child to this node.

Setting Boundary Conditions

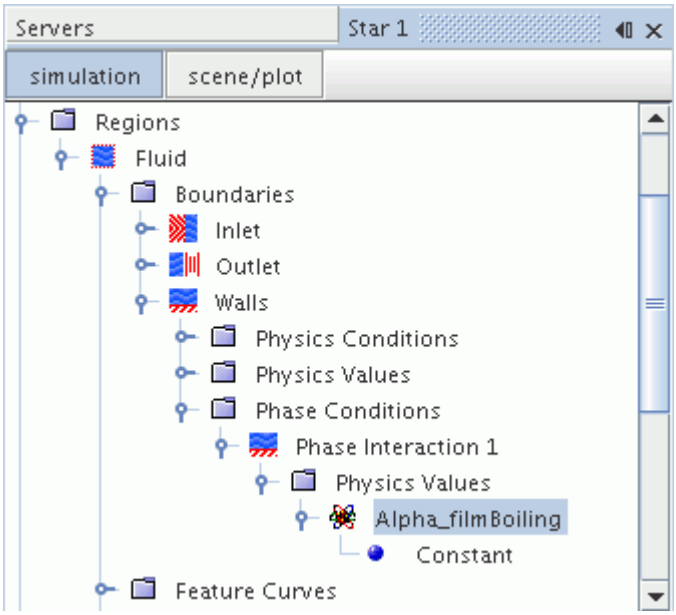
This section describes the boundary conditions that you set for the boiling model.

Either of the following objects becomes available when you work with a boiling model:

- Alpha film boiling—activated by the Rohsenow Boiling model
- Boiling scale factor—activated by the Transition Boiling model

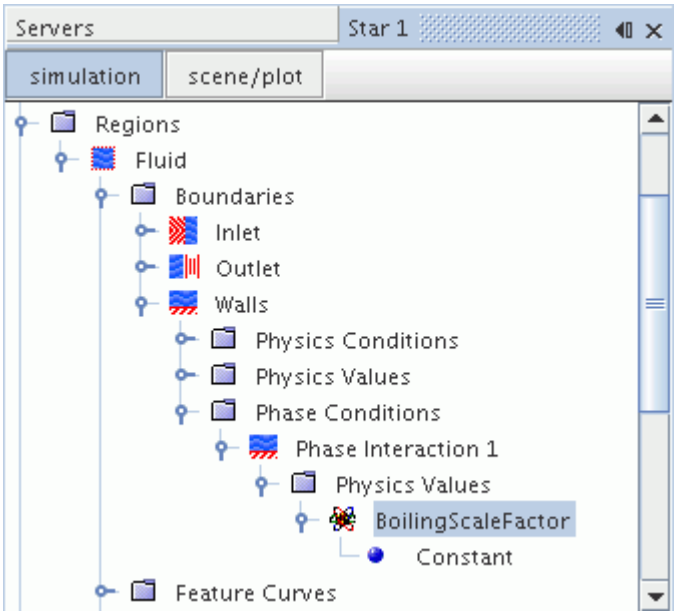
What Is Alpha Film Boiling?

Alpha Film Boiling controls the vapor volume fraction that is associated with the start of film boiling. This value node (**Alpha_filmBoiling** in the following screenshot) is activated as soon as the **Rohsenow Boiling** option is selected in the *Phase Interaction Model Selection* dialog. It is entered as a scalar profile.



What Is the Boiling Scale Factor?

The scale factor S in Transition Boiling allows you to scale $q_{\max}$ up or down for a given boundary. This value node (**BoilingScaleFactor** in the following screenshot) is activated as soon as the **Transition Boiling** option is selected in the *Phase Interaction Model Selection* dialog. It is entered as a scalar profile.



What Is the Film Boiling Vapor Volume Fraction?

The film-boiling vapor volume fraction $\alpha_{filmBoiling}$ is the value of volume fraction of the vapor phase which indicates full transition to film boiling.

If the VOF boiling model is active, you can specify $\alpha_{filmBoiling}$ for each wall boundary. A value of $\alpha_{filmBoiling}$ larger than 1 implies that there is no film boiling along these wall boundaries.

For values of volume fraction less than $\alpha_{filmBoiling}$, a transition function is:

$$f\left(\alpha_{vapor}\right) = 1 - \min\left(1, \alpha_{vapor}/\alpha_{filmBoiling}\right)^8 \tag{811}$$

This function controls the fraction of boiling at the wall that is considered to be nucleate boiling. Here α_{vapor} is the vapor volume fraction at the wall.

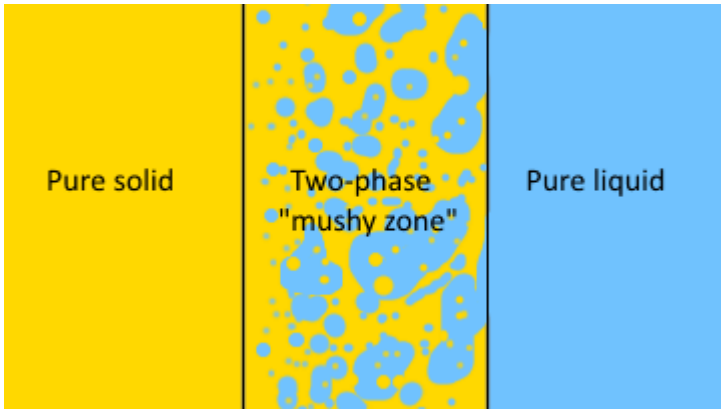
Modeling VOF with Radiation

You can use radiation and VOF models together in many cases involving solidification, boiling, evaporation, and melting, wherever the effects of radiative heat transfer are significant—for example, the case of glass melting in a furnace.

When you select the VOF model, the Radiation model becomes available in the *Optional Models* box. If you select Radiation, the Participating Media Radiation (DOM) model is selected automatically if Auto-select is active. If Auto-select is not active, Participating Media Radiation (DOM) becomes available in the *Radiation* box. In cases where you also want to model the effects of a combusting heat source, such as in a glass furnace, you can also select reacting flow models with the VOF and radiation models. See Combustion and Other Reacting Flows.

After you select the Participating Media Radiation (DOM) model:

- The Gray Thermal Radiation and Multiband Thermal Radiation models become available.
- The radiative properties for participating media appear under the **Material Properties** node of an Eulerian phase if a single component material model is active for that Eulerian phases. For an Eulerian phase with multi-component material models, these radiative properties appear under the **Mixture Properties** node, not under the **Material Properties** nodes of the constituent components.
- These properties can accept constants or field functions as input.
- Radiation is always treated as a single-phase phenomenon. If the simulation has multiple Eulerian phases, they are treated as a single-phase mixture phase for purposes of radiation, not individually. Radiative properties for the mixture are calculated by mass-fraction weighted radiative properties for each Eulerian phase.
- The interface between two phases is modeled using the "mushy zone" concept outline by Chan [526]. A "mushy zone" interface is treated as a very thin finite-volume zone of two phases, rather than as a sharp, discontinuous separation between two phases.



- Radiative properties at boundaries are the same as in all other cases using participating media.
- All other existing features of Participating Media Radiation (DOM) model, the associated solver, and the post-processing capabilities are unchanged.

Surface to Surface (S2S) and Weighted Sum of Gray Gases (WSGG) radiation models are not available.

See The Participating Media Radiation (DOM) Model, The Gray Thermal Radiation Model, The Multiband Thermal Radiation Model.

Modeling Gas Dissolution

This section describes the particular phase-change type that concerns gas dissolution.

What is Gas Dissolution?

Liquid can dissolve a gas up to a certain maximal concentration (saturation).

When there are changes in the pressure or temperature (or both), the liquid can become over-saturated or under-saturated. In the over-saturated case, gas bubbles will form and grow in the liquid; in the under-saturated case the gas bubbles will shrink and disappear. For a multi-component gas phase, the bubbles of free gas are a mixture of those components that come out of the solution.

The presence of dissolved and free gas in a liquid affects other properties, such as the rates of cavitation.

Contents:

Setting up a Gas Dissolution Simulation
Setting Gas Dissolution Model Parameters

Setting up a Gas Dissolution Simulation

This section describes how to use STAR-CCM+ for solving gas dissolution problems.

To set up a gas dissolution simulation:

1. Set up a VOF Multiphase simulation.

   For more information, see Setting Up a VOF Multiphase Simulation.

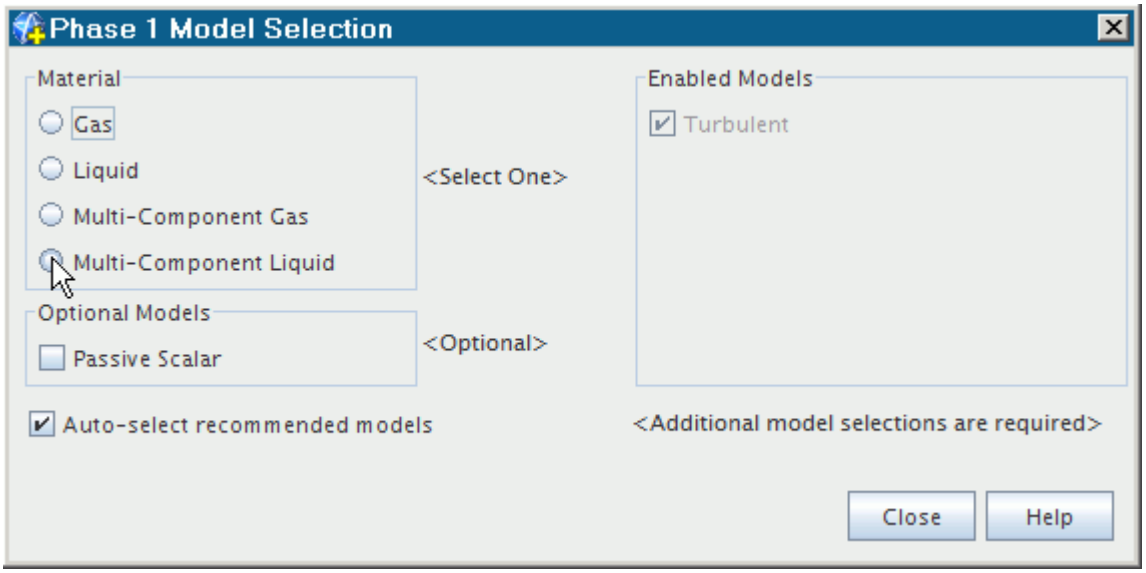
2. Define the bulk properties and phase composition of the mixture.

   For more information, see Defining the Eulerian Phases.

3. Create a Eulerian phase, and then do the following:

   a) Under the new phase node, right click the **Models** node and click **Select models...**

   b) In the *Phase 1 Model Selection* dialog, select **Multi-Component Liquid** from the *Material* group box.

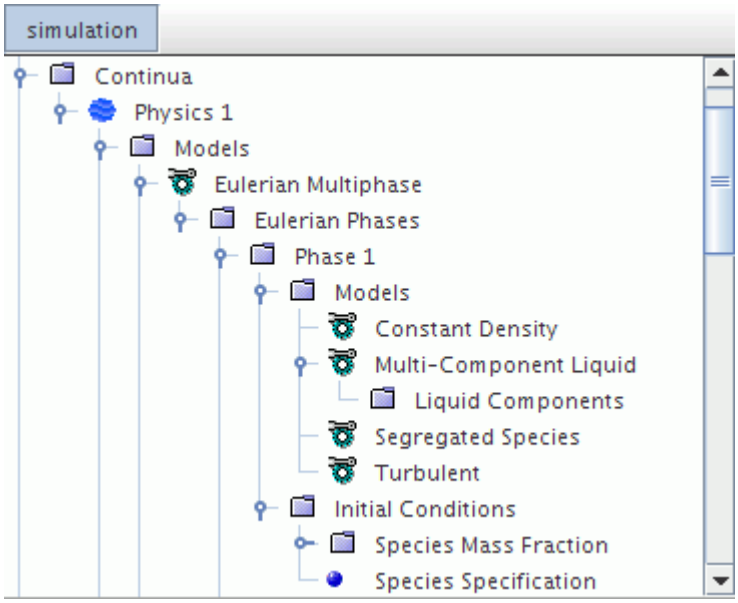


   c) Select **Segregated Species** from the *Optional Models* group box.

   d) Select **Constant Density** from the *Equation of State* group box.

   e) Click **Close**.

   The selected models are added to the phase.



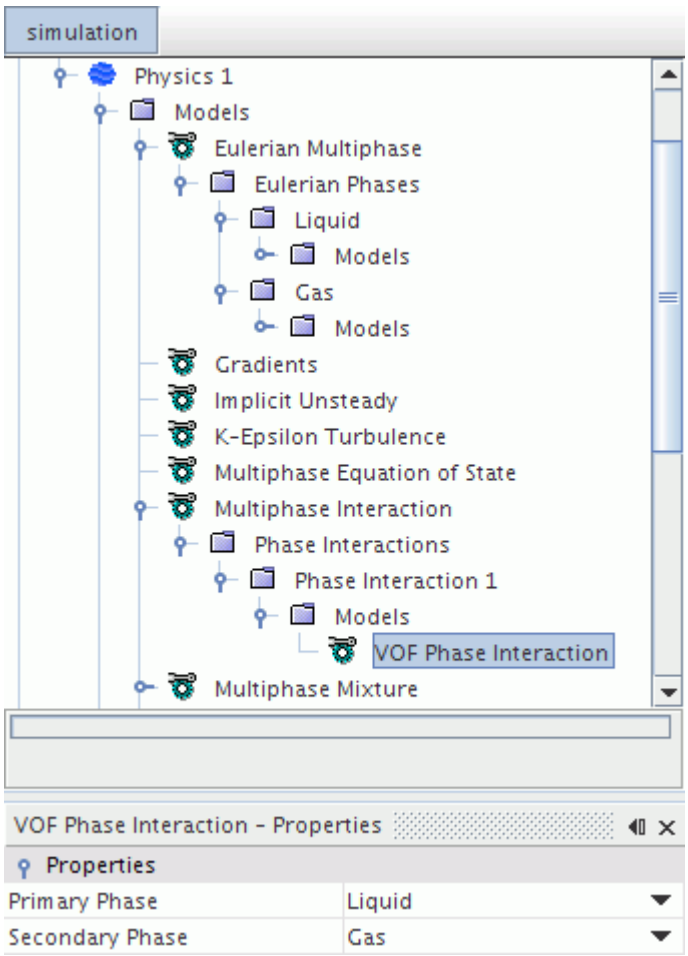
4. Repeat the previous step for a second phase but, in the *Phase 2 Model Selection* dialog, select **Multi-Component Gas** from the *Material* group box.
5. Right-click the **Phase 1** > **Models** > **Multi-Component Liquid** > **Liquid Components** node, and then click **Select Mixture Components**.
6. In the *Select Mixture Components* dialog, select the appropriate components.

   The liquids that are available in the STAR-CCM+ material database do not include dissolved gasses. To specify a dissolved gas as a liquid component, you add the appropriate material to the material database.
7. Repeat the previous step for the gas phase.
8. Set the appropriate material properties relevant to the simulation.
9. Define the appropriate phase interactions.

   For more information, see Controlling Phase Interactions in VOF Simulations.

Before you can select the Gas Dissolution model, you set the primary and secondary phases for the phase interaction.

10. Select the new **VOF Phase Interaction** model node.



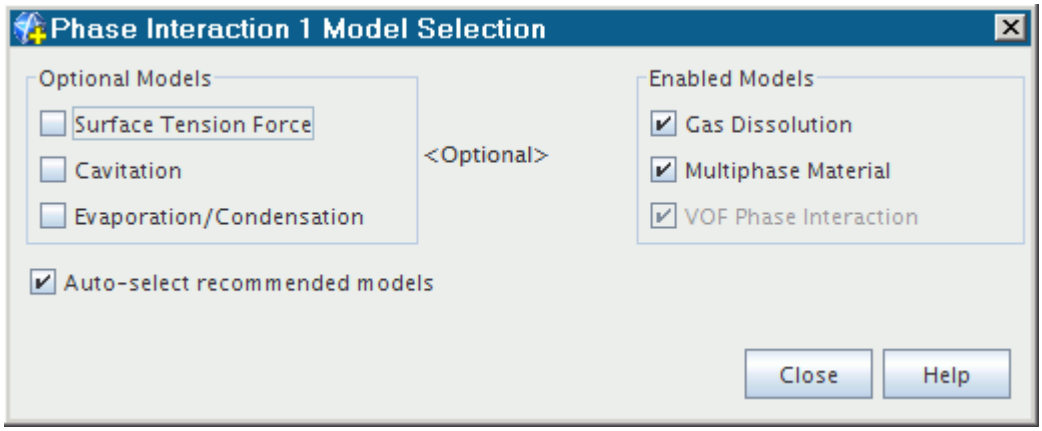
11. In the *Properties* window, do the following:
    - Set the *Primary Phase* to the appropriate liquid phase.

- Set the *Secondary Phase* to the appropriate gas phase.

After you have set the primary and secondary phases for the phase interaction, you can select the Gas Dissolution model.

12. Right-click the **Phase Interaction 1** > **Models** node and choose **Select models...**
13. In the *Phase Interaction 1 Model Selection* dialog, select **Gas Dissolution** from the *Optional Models* group box.

    The *Phase Interaction 1 Model Selection* dialog appears as shown below.



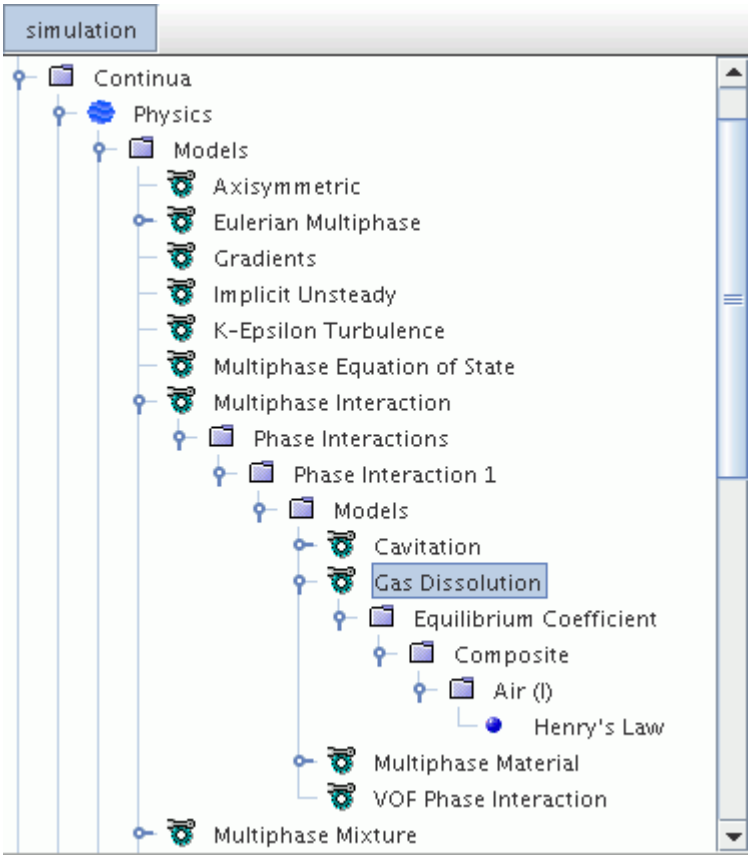
14. Click **Close**.

    The Gas Dissolution model appears under the **Multiphase Interaction** > **Phase Interactions** > **Phase Interaction 1** > **Models** node.

Setting Gas Dissolution Model Parameters

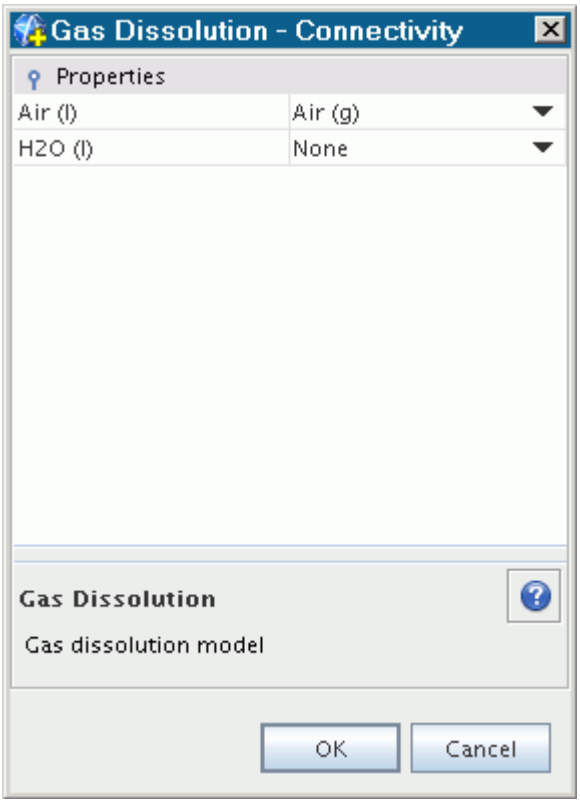
The Gas Dissolution model has one property, *Connectivity*. This property identifies the gas component that corresponds to each dissolved gas component in the liquid mixture.

When a gas dissolution model is selected, STAR-CCM+ creates a node that is called **Gas Dissolution** in the simulation tree, as shown below.



To specify the connections between the liquid and gas phases for each component:

1. Select the **Multiphase Interaction** > **Phase Interactions** > **Phase Interaction 1** > **Models** > **Gas Dissolution** node.
2. In the *Properties* window, click to the right of the *Connectivity* property.
3. In the *Gas Dissolution - Connectivity* dialog, for each liquid component of dissolved gas, select the corresponding gas component.



For gas dissolution models, you ignore the gas components of the solvent liquids in the mixture. To ignore a particular component of the liquid mixture, set the corresponding gas component to **None**.

4. Click **OK**.

Setting the Gas Dissolution Equilibrium Coefficient Method

For each liquid component of dissolved gas, you can specify the method by which the equilibrium coefficient is found. The default for gas components that are dissolved in liquids is Henry's Law.

5. Expand the **Models** > **Gas Dissolution** > **Equilibrium Coefficient** > **Composite** node, and click the component that you want to modify.
6. In the *Properties* window, set the *Method* property to the appropriate option:
   - **Henry's Law**

     This option applies when the dissolved fraction is small, so it is the default for gas components that are dissolved in liquids.
   - **Raoult's Law**

     This option applies when the component is a large fraction of the liquid mixture. It does not usually apply to dissolved gas components.
7. If you use Henry's Law, you can specify the appropriate coefficient value in the Henry's Law Coefficient property.

   The default value is 4.26 E9 Pa, which is the value for oxygen that is dissolved in water at 298.15 K.

Modeling Cavitation

When the static pressure at a particular location within a liquid falls below the saturation vapor pressure of the liquid, the liquid undergoes a phase change called cavitation. This phase change creates cavitation bubbles that are filled with the vapor of the liquid.

STAR-CCM+ provides three cavitation models:

- **Full Rayleigh–Plesset** (for single-component phases) or **Multi-component Full Rayleigh–Plesset** (for multi-component phases)

  The full cavitation model implements the full Rayleigh–Plesset equation and includes the influence of bubble growth acceleration, as well as viscous and surface tension effects.
- **Schnerr-Sauer** (for single-component phases) or **Multi-component Schnerr-Sauer** (for multi-component phases)

  This model is based on a reduced Rayleigh–Plesset equation, and neglects the influence of bubble growth acceleration, viscous effects, and surface tension effects.
- **Homogeneous Relaxation** (for single-component phases) or **Multi-component Homogeneous Relaxation** (for multi-component phases)

  This model is based on a finite rate equation with an empirical time scale formulation, and is suitable for modeling phenomena which occur in thermal non-equilibrium, such as flash boiling.

Contents:

Cavitation Workflow
Setting Up the Cavitating Phases
Setting Up a Cavitation Phase Interaction
Cavitation Model Reference
Homogeneous Relaxation Model Reference

Cavitation Workflow

To set up cavitation in a VOF multiphase simulation, create the liquid and vapor phases, and select the appropriate cavitation model. Then specify the required material properties for the phase interaction, and set the initial conditions and boundary conditions of each phase.

To use STAR-CCM+ for solving cavitation problems:

1. Set up a VOF Multiphase simulation.

   The unsteady (transient) approach is the recommended choice for a cavitation analysis. Cavitation is an intensive process, containing large sources in the momentum and continuity equations, which would require low under-relaxation parameters in a steady analysis.

   See Setting Up a VOF Multiphase Simulation.
2. Create the liquid and vapor phases.

   See Setting Up the Cavitating Phases.
3. Create a phase interaction, select the cavitation model that you want to use, set the model properties, and set the material properties for the phase interaction.

See Setting Up a Cavitation Phase Interaction.

Set the phase material properties that are relevant to the cavitation model. These material properties are available only when a cavitation model is activated in a phase interaction. All of the cavitation models require the saturation pressure to be set in the liquid phase.

4. Under the material properties of the liquid phase, set the saturation pressure for each component of the liquid phase.

   For a single-component liquid, you set the saturation pressure as an absolute value for the liquid. For a multi-component liquid, you set the saturation pressure as a material property for each component of the solvent. The saturation pressure is not relevant to the dissolved gas components.

   See Cavitation Model Material Properties.

The Homogeneous Relaxation model requires further settings.

5. Under the material properties of the liquid phase, set the following for each component of the liquid phase:
   - **Critical Pressure**
   - **Critical Temperature**
   - **Saturation Temperature**

   See Homogeneous Relaxation Model Material Properties.

Setting Up the Cavitating Phases

Define the bulk properties and phase composition of the mixture, including the vapor phase that is created when cavitation occurs.

The liquid and gas phases must be either of the following: both single-component materials or both multi-component materials. You cannot have one phase as a multi-component material and the other phase as a single-component material.

To set up the cavitating phases:

1. Create a Eulerian phase, and select the following models:
   - The appropriate liquid model from the *Material* group box: **Liquid** for a single-component model, or **Multi-Component Liquid** for a multi-component model.
   - **Constant Density** from the *Equation of State* group box.
2. Create a second Eulerian phase, and select the following models:
   - The appropriate gas model from the *Material* group box: **Gas** for a single-component model, or **Multi-Component Gas** for a multi-component model.
   - **Constant Density** from the *Equation of State* group box.
3. Specify the material components for each phase. The example steps below are for the liquid phase:
   - For a single-component liquid:
     1. Right-click the **Models** > **Liquid** > **H2O** node, and then click **Replace with...**.
     2. In the *Replace Material* dialog, select the appropriate liquid.
   - For a multi-component liquid:

1. Right-click the **Models** > **Multi-Component Liquid** > **Liquid Components** node, and then click **Select Mixture Components**.
2. In the *Select Mixture Components* dialog, select the appropriate components.

The steps for the gas phase are identical. However, in some cases, the gaseous form of the solvent is not available in the STAR-CCM+ material database. To specify the gaseous component, you can add the appropriate material to the material database.

Setting Up a Cavitation Phase Interaction

Create a phase interaction, select a cavitation model, and specify the appropriate model properties and material properties.

To set up a cavitation phase interaction:

1. Create a VOF-VOF Phase Interaction.

   See Controlling Phase Interactions in VOF Simulations.

Before you can select a cavitation model, you set the primary and secondary phases for the phase interaction.

2. Select the **VOF-VOF Phase Interaction** model node and do the following:
   - Set the *Primary Phase* to the appropriate liquid phase.
   - Set the *Secondary Phase* to the appropriate gas phase.
3. In the *Phase Interaction Model Selection* dialog, in the *Cavitation Models* group box, select the cavitation model to use:
   - **Full Rayleigh–Plesset** (for single-component phases) or **Multi-component Full Rayleigh–Plesset** (for multi-component phases)

     Implements the full Rayleigh–Plesset equation, which includes the influence of bubble growth acceleration, as well as viscous and surface tension effects.
   - **Schnerr-Sauer** (for single-component phases) or **Multi-component Schnerr-Sauer** (for multi-component phases)

     Implements a reduced Rayleigh–Plesset equation, which neglects the influence of bubble growth acceleration, viscous effects, and surface tension effects.
   - **Homogeneous Relaxation** (for single-component phases) or **Multi-component Homogeneous Relaxation** (for multi-component phases)

     Implements a finite rate equation with an empirical time scale formulation, and is suitable for modeling phenomena which occur in thermal non-equilibrium.

   The selected cavitation model appears under the **Multiphase Interaction** > **Phase Interactions** > **[phase interaction]** > **Models** node.

The Schnerr-Sauer cavitation model is an approximation that is derived for idealized conditions. However, the specific conditions of your simulation can be less than ideal (for example, insufficient mesh and time resolution, unresolved bubble to bubble interactions and flow details).

If you want to move the simulation results for the Schnerr-Sauer cavitation model in a direction that matches observed experimental results, you can scale the bubble growth rate by specifying the appropriate scaling factors for positive and negative bubble growth rates. Using the positive scaling factors lets you increase the cavitation rate if, for example, the mesh is not fine enough to resolve sharp corners that consequently cause

STAR-CCM+ to underpredict the minimum pressure. Negative scaling factors also let you delay bubble collapse if, for example, the bubbles travel inside turbulent eddies where the local pressure is smaller than the average pressure that STAR-CCM+ calculates. In this scenario, the bubble collapse is slower than predicted by the Schnerr-Sauer cavitation model.

4. To scale the positive and negative bubble growth rates for the Schnerr-Sauer cavitation model, expand the **Schnerr-Sauer** node or the **Multi-component Schnerr-Sauer** node and specify the scaling values in the **ScalingFac+** and **ScalingFac-** child nodes.

   The scaling factors are always positive and are independent: you can set either or both, as appropriate to your simulation.

   See Cavitation Model Child Nodes.

For multi-component phases, the *Connectivity* property of the cavitation model lets you select the gas component that corresponds to each liquid component.

5. Select the **Multi-component Schnerr-Sauer** node, the **Multi-component Full Rayleigh-Plasset** node, or the **Multi-component Homogeneous Relaxation** node and set the *Connectivity* property to map the components in the liquid phase to their corresponding components in the gas phase.

   For cavitation models, you can ignore the dissolved gas components of the liquid mixture. To ignore a particular component of the liquid mixture, set the corresponding gas component to **None**.

   See Cavitation Model Family Properties.

   The equilibrium coefficient for liquid components in a cavitation simulation is found using Raoult's Law. There are no *Equilibrium Coefficient* values to set.

The Homogeneous Relaxation model requires further property settings.

6. Select the **Homogeneous Relaxation** node or the **Multi-component Homogeneous Relaxation** node and set the following properties:
   - *Time Scale Modeling Constant*
   - *Void Fraction Exponent*
   - *Dimensionless Pressure Exponent*

   See Homogeneous Relaxation Model Properties.

Many phase change processes can be described as bulk processes that start from certain seeds (imperfections that act as nuclei). STAR-CCM+ uses a homogeneous approach, which is based on the presence of seeds and their subsequent growth and collapse as bubbles. This assumption is used to calculate an appropriate mass transfer rate that is based on the material and thermodynamic properties of the entire control volume [545].

The seed density and seed diameter are specified for each phase interaction, and not per rate model.

This step applies to the Schnerr-Sauer cavitation model and the Full Rayleigh-Plasset cavitation model only.

7. Under the **[phase interaction]** > **Models** > **Multiphase Material** > **Multiphase Material** > **Material Properties** node, set the *Seed Density* and *Seed Diameter* values.

   If multiple rate models use the same phase pair, for example simultaneous cavitation and gas dissolution between a liquid and a gas, the results are consistent. That is, the local radius of the bubbles cannot be different depending on which rate model is used.

If you use two different phase interactions, between the same two phases, for modeling cavitation and dissolution, you can specify the seed properties individually for each phase interaction. This option allows you to tune the models, but in a manner that is inherently inconsistent.

See Cavitation Model Material Properties.

Cavitation Model Reference

The cavitation models let you simulate the growth and collapse of bubbles in cavitating flows.

When cavitation bubbles collapse, they release large amounts of energy into relatively small volumes. The collapsing bubbles create spots of high temperature and emit shock waves which are sources of noise. The collapsing bubbles can produce extensive erosion of the solid surfaces which are in contact with the cavitating regions.

Cavitation is a complex process that is influenced by many factors such as fluid static pressure, saturation temperature and pressure, surface tension, viscosity, density, thermal conductivity, fluid velocity, and distribution of cavitation seeds.

Cavitating flows are typically considered to be isothermal. However, if heat transfer becomes important, cavitation can be considered to behave in a manner similar to boiling. To model this scenario, specify the latent heat that is released or consumed to support the cavitation correctly.

An indicator for the likelihood of cavitation occurrence is the cavitation number N_{cav}. This value is the difference between the static pressure p and the saturation pressure p_{sat}, divided by the dynamic pressure of the incoming flow:

$$N_{cav} = \frac{p - p_{sat}}{\frac{1}{2}\rho_l U^2} \tag{812}$$

In order for cavitation to occur, the cavitation bubbles generally need a surface on which to nucleate. Impurities in the liquid can provide this surface. These pre-existing bubbles are called seeds and start to grow when the conditions for cavitation are met.

These models use the inertia controlled growth model for bubble formation. For more information on bubble growth rates, see Seed-Based Mass Transfer.

Cavitation Model Family Reference

Model Names	**Full Rayleigh-Plesset**
	Multi-component Full Rayleigh-Plesset
	Schnerr-Sauer
	Multi-component Schnerr-Sauer
Theory	See Gas Dissolution and Cavitation Formulation.
Provided By	**[phase interaction]** > **Models** > *Cavitation Models*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Multiphase Interaction** > **Phase Interactions** > **[phase interaction]** > **Models** > **Full Rayleigh-Plesset**

Requires	A VOF Multiphase simulation with the appropriate liquid phase and corresponding vapor phase defined. The liquid and vapor phases must be both single-component materials or both multi-component materials. A VOF-VOF Phase Interaction with the primary phase set to the liquid phase and the secondary phase set to the vapor phase. In the *Phase Interaction Model Selection* dialog: For single-component phases: *Cavitation Models*: **Full Rayleigh-Plesset** or **Schnerr-Sauer** For multi-component phases: *Cavitation Models*: **Multi-component Full Rayleigh-Plesset** or **Multi-component Schnerr-Sauer**	
Properties	*Connectivity* *Bubble Radius Velocity URF* See Cavitation Model Family Properties.	
Activates	Model Controls (child nodes)	**Equilibrium Coefficient** **ScalingFac+** **ScalingFac-** See Cavitation Model Child Nodes.
	Materials	Phase Interaction Material Properties: *Seed Density* *Seed Diameter* *Surface Tension* Phase Material Properties: *Saturation Pressure* (applies to liquid phases only) See Cavitation Model Material Properties.
	Field Functions	Cavitation Rate of <phase interaction> See Field Functions.

Cavitation Model Family Properties

Connectivity

Maps the components in the liquid phase to their corresponding components in the gas phase. For multi-component phases, you select the gas component that corresponds to each liquid component. For cavitation models, you can ignore the dissolved gas components of the liquid mixture.

Applies to multi-component phases only.

Bubble Radius Velocity URF

The under-relaxation factor for the bubble radius velocity updates.

Applies to the Full Rayleigh-Plesset model and Multi-component Full Rayleigh-Plesset model only.

Cavitation Model Child Nodes

Equilibrium Coefficient

Specifies the method by which the equilibrium coefficient is found for each pair of liquid-gas components. By default, Raoult's Law is used, as this option applies when the component is a significant fraction of the liquid mixture.

Applies to multi-component phases only.

ScalingFac+

Scales the positive bubble growth rate (see Eqn. (881)). Specify the scaling value as a constant or a field function.

Applies to the Schnerr-Sauer model and the Multi-component Schnerr-Sauer model only.

ScalingFac-

Scales the negative bubble growth rate (see Eqn. (881)). Specify the scaling value as a constant or a field function.

Applies to the Schnerr-Sauer model and the Multi-component Schnerr-Sauer model only.

Cavitation Model Material Properties

Phase Interaction Material Properties:

Seed Density

The seed density (n_0 in Eqn. (876)) is a liquid-dependent constant and is defined as the number of bubble seeds per unit volume of liquid.

Seed Diameter

The average seed diameter (2 R_0) is used to approximate the spectral seed distribution in a liquid (see Eqn. (875)).

Surface Tension

Each phase interaction is assigned its own surface tension coefficient (σ in Eqn. (879)). This coefficient is used to calculate the surface tension force between each of the defined phases in the phase interaction.

This property applies to the Full Rayleigh-Plesset model and Multi-component Full Rayleigh-Plesset model only. Surface tension effects are neglected in the Schnerr-Sauer model.

Phase Material Properties:

Set under **[Phase]** > **Models** > [**Material** (for example, **Multi-Component Liquid**) or **Component** (for example, **Liquid Components** > **H2O**)] > **Material Properties** when a cavitation model is activated.

Saturation Pressure

The pressure of each vapor component when in equilibrium with the corresponding liquid component. This value is required for each liquid component.

The saturation pressure p_{sat} is a key material property and is required by the Rayleigh-Plesset model (see Eqn. (879) and Eqn. (882)).

Field Functions

The following field function is available when the *Temporary Storage Retained* flag is set on the Segregated Volume Fraction solver.

Cavitation Rate of [phase interaction]

The volumetric rate of evaporation: the amount of vapor (in cubic meters) that is created per cubic meter per second.

Homogeneous Relaxation Model Reference

The Homogeneous Relaxation Model (HRM) is used for modeling vaporization mechanisms ranging from cavitation to flash boiling.

Flash boiling is a phase change in a high temperature liquid that is depressurized below its vapor pressure. This phenomenon occurs in thermal non-equilibrium and therefore requires a finite rate model. The Homogeneous Relaxation Model (HRM) is based on a finite rate equation with an empirical time-scale formulation, so is suited to modeling flash boiling.

The Homogeneous Relaxation model may be used in the following applications:

- Diesel and Gasoline Direct Injection (GDI) fuel injectors

  GDI fuel injectors operate at high temperatures, which makes them susceptible to flash boiling. Flash boiling (or cavitation) in the nozzle greatly influences the flow in the injector and the subsequent spray characteristics of the injector.

  The time scale depends on flow conditions such as pressure, void fraction, fuel vapor pressure, and the saturated liquid and vapor densities. For standard diesel injectors, the time-scale value is small and the vaporization is pressure-driven (that is, cavitation). However, for GDI injectors, the flow conditions and fuel properties make the time-scale value relatively large such that thermal non-equilibrium effects must be modeled.
- Pressurized Water Reactors (PWR)

  Flash boiling of superheated water into steam

Homogeneous Relaxation Model Reference

Model Names	**Homogeneous Relaxation**
	Multi-component Homogeneous Relaxation
Theory	See Homogeneous Relaxation Model Formulation.
Provided By	**[phase interaction]** > **Models** > *Cavitation Models*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Multiphase Interaction** > **Phase Interactions** > **[phase interaction]** > **Models** > **Homogeneous Relaxation**
Requires	A VOF Multiphase simulation with the Segregated Multiphase Temperature model activated. The appropriate liquid phase and corresponding vapor phase defined. The liquid and vapor phases must be both single-component materials or both multi-component materials.

	A VOF-VOF Phase Interaction with the primary phase set to the liquid phase and the secondary phase set to the vapor phase. In the *Phase Interaction Model Selection* dialog: For single-component phases: *Cavitation Models*: **Homogeneous Relaxation** For multi-component phases: *Cavitation Models*: **Multi-component Homogeneous Relaxation**	
Properties	Key properties are: *Time Scale Modeling Constant* *Void Fraction Exponent* *Dimensionless Pressure Exponent* *Flash Boiling Rate Under-Relaxation Factor* See Homogeneous Relaxation Model Properties.	
Activates	Model Controls (child nodes)	**Equilibrium Coefficient** See Multi-component Homogeneous Relaxation Model Child Nodes.
	Materials	Phase Material Properties: *Critical Pressure* (applies to liquid phases only) *Critical Temperature* (applies to liquid phases only) *Heat of Formation* *Saturation Pressure* (applies to liquid phases only) *Saturation Temperature* (applies to liquid phases only) *Standard State Temperature* See Homogeneous Relaxation Model Material Properties.
	Field Functions	**Cavitation Saturation Pressure of [Phase Interaction]** **Cavitation Saturation Temperature of [Phase Interaction]** See Homogeneous Relaxation Field Functions.

Homogeneous Relaxation Model Properties

Time Scale Modeling Constant

This value is θ_0 in Eqn. (893). The default value of this modeling constant is 3.84×10^{-7}.

Void Fraction Exponent

The exponent that is applied to the vapor void fraction, α, in Eqn. (893). The default value is -0.54.

Dimensionless Pressure Exponent

The exponent that is applied to the non-dimensional pressure, ψ, in Eqn. (893). The default value is -1.76.

Flash Boiling Rate Under-Relaxation Factor

The under relaxation factor that is applied to the Homogeneous Relaxation model source term. Under relaxation is recommended in cases where pressure oscillations are observed in the solution.

Connectivity

Maps the components in the liquid phase to their corresponding components in the gas phase. For multi-component phases, you select the gas component that corresponds to each liquid component. For the Homogeneous Relaxation model, you can ignore the dissolved gas components of the liquid mixture.

Applies to multi-component phases only.

Multi-component Homogeneous Relaxation Model Child Nodes

Equilibrium Coefficient

Specifies the method by which the equilibrium coefficient is found for each pair of liquid-gas components. By default, Raoult's Law is used, as this option applies when the component is a significant fraction of the liquid mixture.

Applies to multi-component phases only.

Homogeneous Relaxation Model Material Properties

The following properties are set for single-component phases or for each component of multi-component phases.

These properties are set under **[Phase]** > **Models** > [**Material** (for example, **Multi-Component Liquid**) or **Component** (for example, **Liquid Components** > **H2O**)] > **Material Properties** when the Homogeneous Relaxation model is activated.

Critical Pressure

The pressure that is required to liquefy a material at its critical temperature, given by p_{crit} in Eqn. (894).

This value is required for each liquid component.

Critical Temperature

The temperature above which the material cannot be liquefied, regardless of the pressure that is applied. This value is required for each liquid component.

Heat of Formation

The heat that is evolved when 1 kilogram of the material is formed from its elements in their respective standard states [J/kg]. See Using the Heat of Formation.

Saturation Pressure

The saturation pressure p_{sat} is a key material property (see Eqn. (894)). It is the pressure of each vapor component when in equilibrium with the corresponding liquid component. This value is required for each liquid component.

For multicomponent mixtures, the saturation pressure must be a function of the saturation temperature and cannot be constant.

Saturation Temperature

The temperature for a corresponding saturation pressure at which the liquid material boils to form its vapor phase.

This property is not available for multicomponent mixtures, as STAR-CCM+ automatically calculates the saturation temperature using an iterative method.

Standard State Temperature

The temperature at which the standard state of the material is defined. See Using the Standard State Temperature.

Homogeneous Relaxation Field Functions

For both single component phases and multi-component phases:

Cavitation Saturation Pressure of [Phase Interaction]

Cavitation Saturation Temperature of [Phase Interaction]

For multi-component phases only:

Effective Mass Diffusivity of [component] in [multi-component material]

Enthalpy of [component] in [multi-component material]

Mass Fraction of [component] in [multi-component material]

Molar Concentration of [component] in [multi-component material]

Mole Fraction of [component] in [multi-component material]

Molecular Weight of [component] in [multi-component material]

Specific Heat of [component] in [multi-component material]

Modeling Melting and Solidification

Melting is the process which changes the state of matter from solid to liquid. It takes place when a solid substance is heated to the melting temperature. The opposite process is called solidification.

For pure substances, for example water, the melting and solidification usually take place at one temperature, that is, the liquidus and the solidus are at the same temperature. Melting and solidification take place instantaneously so, depending on the temperature, there is either liquid or solid in the computational domain.

For impure substances, for example alloys, the melting and solidification can take place over a temperature range. In this case, the liquidus and solidus are not at the same temperature. In the temperature interval between them, the liquid is partly solidified; the solid and liquid phases exist simultaneously (like a slurry). This temperature interval can have a finite width up to several 100 degrees Kelvin. The melting and

solidification process is not instantaneous but takes place over a period of time. During this time, the phase mixture can influence the flow field.

Flow Resistance Due to Solidification

The Melting and Solidification model simulates only the effect of phase change on the energy equation; it does not modify the momentum equations. A further model is required to account for the change in material properties from the liquid state to the solid state.

Depending on the value of the relative solid fraction (specifying to which extent the liquid has solidified), there are three optional models available:

- Slurry Viscosity Model

  For low values of solid fraction the mixture is considered as a slurry, where solid grains are suspended in the melt nearly without interacting with each other. Their effect on the flow is often modeled as a relative solid fraction-dependent viscosity.

  The Metzner method is implemented to increase the viscosity (slurry viscosity) up to the critical solid fraction (approximately 0.3) (see Eqn. (903)), which models the solidified grains flowing with the fluid without much interaction. However, using a range of viscosity larger than about two orders of magnitude can be unstable.

- Mushy Zone Permeability Model

  When the solid volume fraction increases, equiaxed grains grow and start to agglomerate and to form dendritic regions. Dendrites also grow from cooled solid surfaces or the solidification front into the melt. A partially solidified stationary region permeated with dendrites is called a “mushy zone”. The flow resistance in a mushy zone can be modeled similar to an isotropic porous medium, using the Carman-Kozeny Mushy Zone Permeability method (see Eqn. (908)). This model is used for relative solid volume fractions above the specified critical solid fraction (approximately 0.3).

- Flow Stop Functionality

  To stop the flow on solidification completely, you can use the flow stop functionality. The Melting-Solidification Flow Stop model is activated above the specified flow stop solid fraction (approximately from 0.4 through 1). The resistance that the Slurry Viscosity and Mushy Zone Permeability models add to moving fluids depends on the flow velocity. When the velocity is zero, the effect of both models is also zero. In the presence of pressure gradients, or body forces such as gravity, a creeping flow develops independently of the magnitude of slurry viscosity or mushy zone permeability.

Applicability of the Flow Resistance Models

The Slurry Viscosity model and the Mushy Zone Permeability model affect the flow properties of a partially solidified liquid only. You need these models only when you are simulating impure substances such as alloys. You do not need them, for example, when you simulate ice melting in a stationary container.

If you simulate the melting and solidification process of a pure substance, use the Melting and Solidification model only. Partial solidification is not possible for pure substances. To prevent creeping flows, you can additionally activate the flow stop functionality.

When simulating melting and solidification of impure substances, use the following table to determine which flow resistance models to apply.

Flow Velocities	Solidus/Liquidus temperature interval	Simulation Time	Flow Resistance Models
Low (< 1 m/s)	Small	Short (< 1 s)	Slurry Viscosity
Low (< 1 m/s)	Small	Long (> 3 s)	Slurry Viscosity Flow Stop
Low (< 1 m/s)	Large	Short (< 1 s)	Slurry Viscosity Mushy Zone Permeability
Low (< 1 m/s)	Large	Long (> 3 s)	Slurry Viscosity Mushy Zone Permeability Flow Stop
High (> 5 m/s)	Small	Short (< 1 s)	Slurry Viscosity (Flow Stop)
High (> 5 m/s)	Small	Long (> 3 s)	Slurry Viscosity Flow Stop
High (> 5 m/s)	Large	Short (< 1 s)	Slurry Viscosity Mushy Zone Permeability (Flow Stop)
High (> 5 m/s)	Large	Long (> 3 s)	Slurry Viscosity Mushy Zone Permeability Flow Stop

Contents:

Selecting a Simulation with Melting and Solidification
Melting-Solidification Model Reference
Slurry Viscosity Model Reference
Mushy Zone Permeability Model Reference
Stopping the Flow
Setting up the Relative Velocity field for Mushy Zone Permeability Model

Melting and Solidification Workflow

You can simulate the melting and solidification process of both pure substances and alloys.

To solve problems that involve melting and solidification:

1. Set up a VOF Multiphase simulation.

   The Melting-Solidification model requires a non-isothermal energy model, so activate **Segregated Multiphase Temperature** from the *Optional Models* group box.

   See Setting Up a VOF Multiphase Simulation.

2. Specify the melting or solidifying phases.

   The Melting-Solidification model is available for single-component liquid Eulerian phases only. An impure substance (an alloy) is modeled as a single liquid phase with material properties that represent the mixture. Multiple substances can be specified as discrete liquid phases, each with their own material properties.

   For each phase, do the following:

   a) Select **Liquid** from the *Materials* group box.
   b) Select **Melting-Solidification** from the *Optional Models* group box.

      You do not need to specify the corresponding solid material for this process.

   The Melting-Solidification Flow Stop model is added to the physics continuum automatically when you activate the Melting-Solidification model for a phase. It is deactivated by default, but is activated when you set the appropriate property values.

3. Select the optional flow resistance models that are appropriate for your simulation.

   See Applicability of the Flow Resistance Models.

   The optional flow resistance models are:

   - Slurry Viscosity
   - Mushy Zone Permeability

   Partial solidification is not possible for pure substances, so the optional flow resistance models are not relevant when you simulate the melting or solidification of a pure substance.

4. Set the *Minimum Allowable Temperature* and *Maximum Allowable Temperature* reference values for the physics continuum.

   See Reference Values.

5. Set the material properties of the liquid phase:

   For impure substances (alloys), you specify the material properties that represent the mixture. For a mixture, you specify a fraction solid curve, and different values for the liquidus and solidus temperatures. For pure substances, no fraction solid curve exists and the liquidus and solidus temperatures are identical.

   a) Specify the liquid material.
   b) Specify the temperature values that cause the material to melt and/or to solidify.

      See Material Properties.

   The Relative Solid Volume Fraction of [phase] field function indicates the state of the fluid.

6. Set the thermal conductivity.

   It is good practice to use a field function for the Thermal Conductivity material property of the phase that melts and solidifies. The values of this property differ for liquid and solid states. The thermal conductivity of most materials varies with phase and temperature.

7. If relevant to your simulation, set up the Slurry Viscosity model.

   The use of the switching function for the Slurry Viscosity and Mushy Zone Permeability models is optional. You are advised to activate the switching function for both models when you use the Metzner method and the Carman-Kozeny method in combination. The Metzner slurry viscosity model can produce high slurry viscosities when used above the critical solid fraction. If the switching function is not used, the

slurry viscosity is not reduced to zero above the critical solid volume fraction and serves as a multiplicative factor to the effect of the mushy zone permeability.

If you combine the Metzner slurry viscosity model with a user-defined mushy zone permeability model, you can deactivate the switching function. Similarly, if you combine the Carman-Kozeny mushy zone permeability model with a user-defined slurry viscosity model, you can deactivate the switching function. Ensure that the effect of each model in its respective applicability range is accounted for within the user-defined field function.

a) In the liquid phase, set the *Slurry Viscosity* material property. If you use the **Metzner** method, set the appropriate property values.

   See Material Properties.

b) In the physics continuum, set the **Average Slurry Viscosity** property: the *Slurry Viscosity Under-Relaxation Factor*.

   See Average Slurry Viscosity Property.

8. If relevant to your simulation, set up the Mushy Zone Permeability model.

   a) In the liquid phase, set the *Mushy Zone Permeability* material property. If you use the **Carmen-Kozeny** method, set the appropriate property values.

      See Material Properties.

   b) In the physics continuum, set the **Average Mushy Zone Permeability** properties.

      See Average Mushy Zone Permeability Properties.

   c) Specify the relative velocity field.

      See Setting up the Relative Velocity field for the Mushy Zone Permeability Model.

9. If relevant to your simulation, use the flow stop functionality to prevent creeping flows: activate the Melting-Solidification Flow Stop model.

   You can set the relative solid fraction at which the flow stops in a cell and the under-relaxation factor for melting and solidification. The flow stop is active only when the respective Flow Stop Solid Fraction is also specified for the Melting-Solidification Flow Stop model.

   The flow stop functionality can be used only in simulations where the cell volume remains constant. Examples include simulations with rigid body motion and moving reference frames, but exclude simulations with morphing or user-defined vertex motion. For cases where the cell volume changes, use a field function to increase the dynamic viscosity with respect to the relative solid volume fraction by several orders of magnitude. Avoid sudden changes of viscosity, as these changes can destabilize the simulation.

   See Stopping the Flow.

Melting-Solidification Model Reference

The Melting-Solidification model simulates the melting and solidification process of both pure substances and alloys. You can include optional flow resistance models in your simulation, and can stop the flow when the solid volume fraction exceeds a specified value.

Melting-Solidification Model Reference

Theory	See Melting-Solidification Formulation.

Provided By	**[phase name]** > **Models** > *Optional Models*	
Example Node Path	**[phase name]** > **Models** > **Melting-Solidification**	
Requires	In the physics continuum: *Material*: **Eulerian Multiphase** *Eulerian Multiphase Model*: **Volume of Fluid (VOF)** *Optional Models*: **Segregated Multiphase Temperature** In the Eulerian phase: *Material*: **Liquid** *Optional Models*: **Melting-Solidification**	
Properties	Key property is: *Flow Stop Relative Solid Fraction*. See Melting-Solidification Properties.	
Activates	Physics Models	Activated in the physics continuum: **Melting-Solidification Flow Stop**. Made available in the *Phase Model Selection* dialog, *Optional Models* group: **Slurry Viscosity**, **Mushy Zone Permeability**.
	Materials	*Fraction Solid Curve* *Latent Heat of Fusion* *Liquidus Temperature* *Solidus Temperature* See Material Properties.
	Reference Values	*Minimum Allowable Temperature* *Maximum Allowable Temperature* See Reference Values.
	Field Functions	See Field Functions.

Melting-Solidification Properties

Flow Stop Relative Solid Fraction

Defines the cell solid volume fraction at which the fluid loses its ability to flow. This threshold depends only on the material. However, the flow is stopped in the cell only when the total amount of solidified fluid exceeds the *Flow Stop Solid Fraction* that is specified for the **Melting-Solidification Flow Stop** model. This property is deactivated for values greater than 1. The default value is 1.1.

See Stopping the Flow.

Linearize Melting

When activated, the source term due to latent heat is linearized. This option is deactivated by default.

Under-Relaxation Factor

Governs the extent to which the newly computed solution supplants the old solution for each iteration.

Melting-Solidification Flow Stop Properties

Flow Stop Solid Fraction

Specifies the solid volume fraction at which the fluid flow is stopped in a cell. This value is a purely geometrical property. Valid values are from 0 through 1. A cell is blocked for flow when the accumulated volume of solid phases exceeds the specified fraction. A value greater than 1 deactivates the flow stop functionality.

This setting is a property of the continuum, so it depends on all of the phases that are present.

Flow Stop Mass Compensation

Applies appropriate mass and energy sources to compensate for the unphysical mass change in stopped cells and maintain the total mass of each phase.

When the flow stop functionality is used in combination with a compressible equation of state (Polynomial Density or User Defined EOS), fluxes through the cell faces cannot compensate for the change in density in stopped cells. This effect causes the fluid mass in the respective cell to change proportional to the density change. Over the simulation time, this change can have a significant effect on the total mass of the solidifying phase.

To prevent a flow deflection in a non-stopped cell next to a stopped cell, the face area weighted average static pressure of all non-stopped cells next to a stopped cell is applied to the stopped cell. To maintain the correct heat fluxes through the stopped cells, the temperature cannot be adjusted. The density, which is a function of pressure and temperature, determines the mass in a stopped cell that has a constant volume.

Material Properties

The following phase material properties are available:

Fraction Solid Curve

The fraction solid curve defines the relative solid volume fraction variation with temperature (and absolute pressure for compressible simulations).

The available methods are **Polynomial in T**, **Table(T)**, **Table(T,P)**, and **Linear**. The *Table(T,P)* method applies only to compressible cases.

When you use the *Polynomial in T*, *Table(T)*, or *Table(T,P)* method, you specify absolute temperature values. The solidus temperature and liquidus temperature are then assumed to be the minimum and maximum specified values. Both temperatures must exceed the *Minimum Allowable Temperature* that you set within the Reference Values for the physics continuum. The solid volume fraction is 1 for the solidus temperature, and 0 for the liquidus temperature.

When you use the *Linear* method, the solid volume fraction changes linearly with temperature between the solidus temperature and liquidus temperature, as specified in Eqn. (902).

Latent Heat of Fusion

The specific latent heat of fusion h_{fusion} is the amount of energy in the form of heat that a substance releases or absorbs during a change of state.

Liquidus Temperature

Solidus Temperature

Melting and solidification take place when the material temperature is between the solidus temperature and liquidus temperature. When the material temperature is above the liquidus temperature, the material is entirely in the liquid state. When the material temperature is below the solidus temperature, the material is entirely in the solid state. The solidus temperature and the liquidus temperature can have the same value for pure substances.

These material properties are available only when the *Fraction Solid Curve* material property uses the **Linear** method. When other methods are used, the solidus temperature and liquidus temperature are automatically computed as the lower and upper limits of the input that is provided for the respective method.

The liquidus temperature and solidus temperature must be within the *Minimum Allowable Temperature* and *Maximum Allowable Temperature* that you set within the Reference Values for the physics continuum.

When using the linear fraction solid curve method, the liquidus temperature must be greater or equal to the solidus temperature. A minimum difference of 0.002 K between both is automatically applied. For other methods, the liquidus temperature must be at least 0.002 K higher than the solidus temperature.

These values are specified as constants.

Reference Values

The Reference Values for the physics continuum are:

Minimum Allowable Temperature

Maximum Allowable Temperature

Field Functions

The following field functions become available:

Flow Stop Flag

Lets you visualize the flow stopped cells.

Flow Stop Flag of [phase]

Lets you visualize the flow stopped cells.

The *Flow Stop Flag* for a phase gets a non-zero value whenever the relative solid volume fraction of that phase exceeds the threshold value that is specified for the *Flow Stop Relative Solid Fraction* property.

See Melting-Solidification Properties.

Relative Solid Volume Fraction of [phase]

Indicates how much of the phase is in the solid state, as defined by Eqn. (902).

When the *Temporary Storage Retained* option is selected for the Segregated Flow solver, a number of field functions become available. Some examples are:

Ap Velocity, **Apparent Pressure Gradient**, **Body Force**, **Density Limiter**, **Density Recon**, **d(rho)/d(p)**, **Flow Work Energy Source**, **Pressure Limiter**, **Pressure Recon**, **Pressure-Correction**, **Rhie-Chow**

Unsteady Scale, **Slip Velocity**, **Strain Rate Tensor Modulus**, **Velocity Divergence Imbalance**, **Velocity-X Correction**, **Vorticity Tensor Modulus**, **X-Momentum Residual**.

Slurry Viscosity Model Reference

For values of solid fraction lower than a critical solid volume fraction, the mixture is considered as a slurry, where solid grains are suspended in the melt with almost no interaction between them. Their effect on the flow is modeled as an additional viscosity using the Slurry Viscosity model.

Slurry Viscosity Model Reference

Theory	See The Slurry Viscosity Model.	
Provided By	**[phase name]** > **Models** > *Optional Models*	
Example Node Path	**[phase name]** > **Models** > **Slurry Viscosity**	
Requires	The **Melting-Solidification** model activated in the Eulerian phase.	
Properties	None.	
Activates	Physics Models	**Average Slurry Viscosity** (in the physics continuum) See Average Slurry Viscosity Properties.
	Materials	*Slurry Viscosity* See Materials Properties.
	Field Functions	See Field Functions.

Average Slurry Viscosity Properties

A volume-weighted average slurry viscosity is computed based on the viscosities of each phase when melting or solidification is in progress.

Slurry Viscosity Under-Relaxation Factor

At each iteration, this value governs the extent to which the newly computed solution supplants the old solution. The default value is 0.7.

Material Properties

Slurry Viscosity (Phase)

The available methods are **Field Function** and **Metzner**.

The **Metzner** method has the following properties:

- *Use Switching Function*
  Activates the switching function for the combined use of the Metzner slurry viscosity model and the Carman-Kozeny mushy zone model.

See Switching Function for Metzner Slurry Viscosity and Carman-Kozeny Mushy Zone Permeability Model.

- *Switching Function: Critical Solid Fraction*
  The critical solid volume fraction α^*_{cr} (see Eqn. (909)). The default value is 0.27 (for Aluminum alloy A201).
- *Crystal Constant A*
  The crystal constant A which depends on the aspect ratio and surface roughness of the crystal (see Eqn. (903)). The default value is 0.3 (for Aluminum alloy A201).
- *Switching Function: Constant s*
  The non-dimensional constant s (see Eqn. (909)). The default value is 100 because the shape is rather insensitive for values from 80 through 120.

Slurry Viscosity (Mixture)

For simulations with two or more liquid phases, the slurry viscosity for one cell is taken as a volume-weighted average for the whole mixture.

Field Functions

Average Slurry Viscosity

The volume-weighted slurry viscosity of the mixture, which is calculated from the slurry viscosities of the component phases.

Slurry Viscosity of [phase]

The additional bulk viscosity due to the partial solidification of the phase.

Mushy Zone Permeability Model Reference

When the solid volume fraction increases above a critical solid volume fraction, a mushy zone permeated with dendrites is formed. The flow resistance in a mushy zone is modeled as an isotropic porous medium.

Mushy Zone Permeability Model Reference

Theory	See The Mushy Zone Permeability Model.	
Provided By	**[phase name]** > **Models** > *Optional Models*	
Example Node Path	**[phase name]** > **Models** > **Mushy Zone Permeability**	
Requires	The **Melting-Solidification** model activated in the Eulerian phase.	
Properties	None.	
Activates	Physics Models	**Average Mushy Zone Permeability** (in the physics continuum) See Average Mushy Zone Permeability Properties.
	Materials	*Mushy Zone Permeability* See Material Properties.

	Field Functions	See Field Functions.

Average Mushy Zone Permeability Properties

When the **Mushy Zone Permeability** model is activated in a phase, the **Average Mushy Zone Permeability** model node appears in the parent physics continuum. The contributions to flow resistance from each phase where melting and solidification occur are volume-weighted for the computed cell. These contributions are then added to the flow resistance source term of the momentum equation.

Use Forchheimer's Term

Adds the contribution of the Forchheimer's term to the flow resistance of the Carman-Kozeny mushy zone model.

See Eqn. (908) in The Mushy Zone Permeability Model.

Ergun Coefficient

Sets the Ergun coefficient in Eqn. (908). The default value is 0.55.

Max. Porous Viscous Resistance

Sets a maximum level of resistance. This setting applies only to the cells with a solid volume fraction greater than zero, in all regions to which the respective physics continuum is assigned.

Mushy Zone Permeability Under-Relaxation Factor

Governs the extent to which the newly computed solution supplants the old solution at each iteration. The default value is 0.7.

Material Properties

The phase has the following material property:

Mushy Zone Permeability

The available methods are **Field Function** and **Carman-Kozeny**.

See The Mushy Zone Permeability Model.

The **Carman-Kozeny** method has the following properties:

- *Use Switching Function*
  Use the switching function for the combined use of the Metzner slurry viscosity model and the Carman-Kozeny mushy zone model.

  See Switching Function for Metzner Slurry Viscosity and Carman-Kozeny Mushy Zone Permeability Model.
- *Switching Function: Critical Solid Fraction*
  The critical solid volume fraction α^*_{cr} (see Eqn. (910)). The default value is 0.27 (for Aluminum alloy A201).
- *Secondary Dendrite Arm Spacing*
  The secondary dendrite arm spacing is used for the calculation of the Carman-Kozeny Shape Factor.
- *Switching Function: Constant s*

The non-dimensional constant s (see Eqn. (910)). The default value is 100 because the shape is rather insensitive for values from 80 through 120.

- *Carman-Kozeny Shape Factor c*
  The shape factor c_s is a function of the dendrite dimension (typically secondary dendrite arm spacing d) and a shape constant c (see Eqn. (905)). The default value for c is 180.

The mixture has the following material property:

Mushy Zone Permeability

For simulations with different phases, the mushy zone permeability for one cell is taken as a volume-weighted average for the whole mixture.

Field Functions

Average Mushy Zone Permeability

The volume-weighted mushy zone permeability of a mixture, which is calculated from the mushy zone permeability of the component phases.

Average Mushy Zone Porous Viscous Resistance

The dynamic viscosity of the mixture divided by Average Mushy Zone Permeability. This field function is provided because the effect of a resistance on the flow is easier to understand than the effect of a permeability.

Mushy Zone Permeability of [phase]

The permeability due to obstruction of the flow by a dendritic mesh resulting from partial solidification of the specific phase. The effect of the dendritic mesh is modeled as a porous medium with spatially varying properties.

Mushy Zone Velocity

The momentum source terms of the **Mushy Zone Permeability** model enforce a flow velocity equal to the Mushy Zone Velocity.

Stopping the Flow

You can use the melting-solidification flow stop functionality when you want to prevent all flow (not allowing slow creeping flow) or when the solid needs to withstand high pressure gradients.

When the **Melting-Solidification** model is activated in a phase, the **Melting-Solidification Flow Stop** model is added to the physics continuum automatically. The **Melting-Solidification Flow Stop** model specifies the solid volume fraction at which the fluid flow is stopped in a cell and optionally applies appropriate mass and energy sources to compensate for the unphysical mass change in stopped cells. This model is deactivated by default.

Flow stopped cells have zero velocity (relative to the cell center velocity) and a pressure equal to the face-weighted average pressure of all adjacent non-stopped cells. However, if a flow stopped cell is only next to other flow stopped cells, the static pressure is equal to the reference pressure, as this model does not calculate stresses in the solid.

When you use the flow stop functionality, the following limitations apply:

- The flow stop relative solid fraction does not apply to regions that are subject to motion that affects the mesh volume (for example, Morpher or User-Defined Vertex Motion).

  In morphed regions, the flow stop flag is set to zero for every cell (that is, there are no flow-stopped cells at any time).
- The face between a stopped and a non-stopped cell is not a wall.

  No wall boundary conditions for flow (such as slip/no-slip, or wall function) or turbulence are applied to faces between stopped cells and non-stopped cells.

To stop the flow:

1. For each relevant phase, select the **Melting-Solidification** node and set the *Flow Stop Relative Solid Fraction* property to specify when the particular phase loses its ability to flow.

   The value that you specify depends on the properties of the material. Valid values are from 0 through 1.
2. In the physics continuum, select the **Melting-Solidification Flow Stop** node and set the following properties:

Property	Setting
Flow Stop Solid Fraction Specifies when a cell is blocked for the flow. This setting is a property of the continuum, so it depends on all of the phases that are present.	Specify the solid volume fraction at which the fluid flow is stopped in a cell. The fluid velocity is set to zero for cells that have a solid volume fraction above the specified flow stop value. Valid values are from 0 through 1. You normally set a high value (for example, 0.99) because the simulation already accounts for the effect of slurry viscosity and mushy zone, which decelerate the flow for lower solid volume fractions. However, you can set a lower value when there is a large interval between the solidus and liquidus temperatures. If you want to deactivate this functionality, specify a value greater than 1.
Flow Stop Mass Compensation Applies appropriate mass and energy sources to compensate for the unphysical mass change in stopped cells and maintain the total mass of each phase. This feature applies when a simulation uses a compressible equation of state (Polynomial Density or User Defined EOS).	If applicable to your simulation, activate this functionality. When this option is activated, the density of all VOF phases in stopped cells is kept constant at the value that it had at the time when the cell was stopped. This value is maintained independently of what the equation of state returns for the VOF phase. The constant density is required to keep the mass in the stopped cells constant when cell volume is kept constant and the dependency of density on stresses is omitted. The density in not-stopped cells is not modified. Also, the separate material properties *Density Temperature Derivative* and *Density Pressure Derivative* are not adjusted accordingly — these values are set to zero for stopped cells.

Setting Up the Relative Velocity Field for the Mushy Zone Permeability Model

Generally, the Mushy Zone Permeability model applies a momentum source term to stop the flow of the solidified liquid. The flow is stopped by reducing the convective velocity to zero with respect to the laboratory coordinate system.

For some applications, it can be useful to specify a non-zero velocity field for the solidified liquid. For example:

- Centrifugal casting

  Liquid melt is injected through a nozzle splashing onto the inner wall of a rotating cylinder. When solidification takes place, the solidified liquid must rotate at the same angular velocity as the rotating cylinder so the solidified liquid has zero velocity relative to the rotating wall.
- Continuous/Strand casting

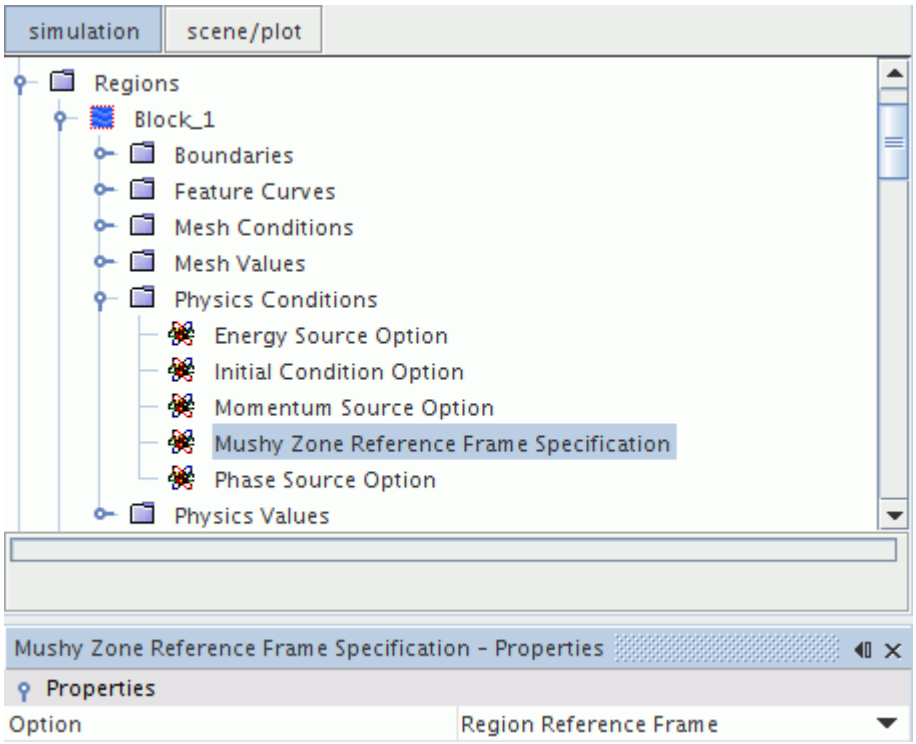
  Liquid melt is injected at the top of an open-base mold. In the mold, a thin sheet of metal next to the mold wall solidifies. The solidified liquid has to move at the same velocity as the strand is pulled away. The velocity with which the mold wall and the solidified liquid move (with zero relative velocity between them) is called the pull velocity.

To set up a relative velocity field for the Mushy Zone Permeability model:

1. Set up the appropriate stationary moving reference frame motion or rigid body motion model.

   You can specify any motion except motions that affect the cell volume (such as the morpher).

   See Working with Motions.
2. In the appropriate region, select the **Physics Conditions** > **Mushy Zone Reference Frame Specification** node.



3. In the *Mushy Zone Reference Frame Specification - Properties* window, select the appropriate reference frame option:
   - **Region Reference Frame**

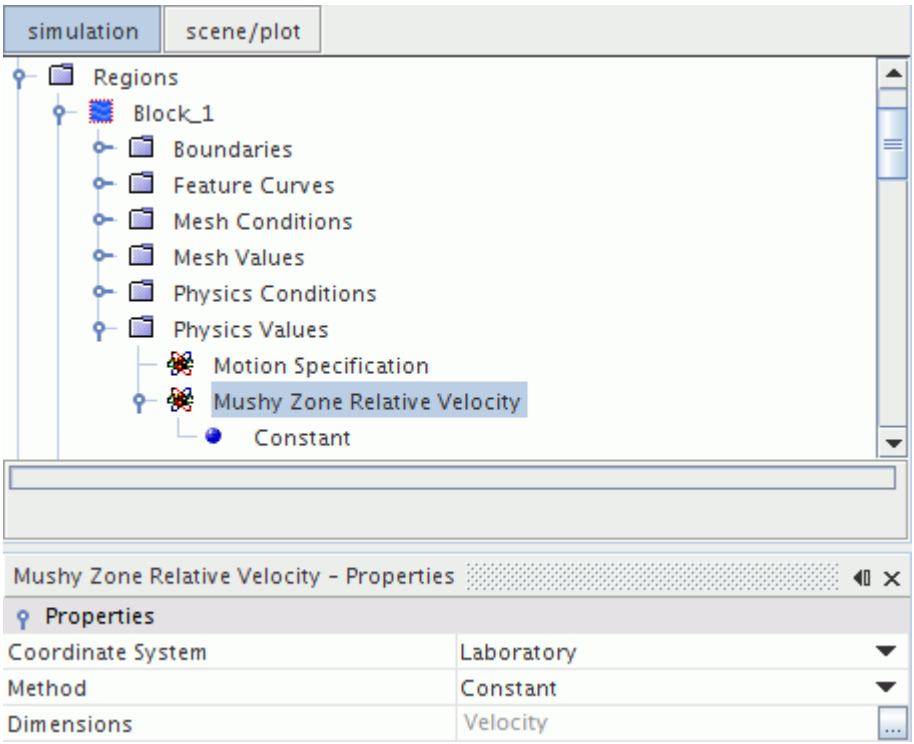
     The relative velocity is specified with respect to the region reference frame. The region reference frame is the motion reference frame, which is selected in **Physics Values** > **Motion Specification** > **Reference Frame**.
   - **Lab Frame**

     The relative velocity is specified with respect to the laboratory reference frame.
   - **Local Reference Frame**

     The relative velocity is specified with respect to a local reference frame with which the region is associated.

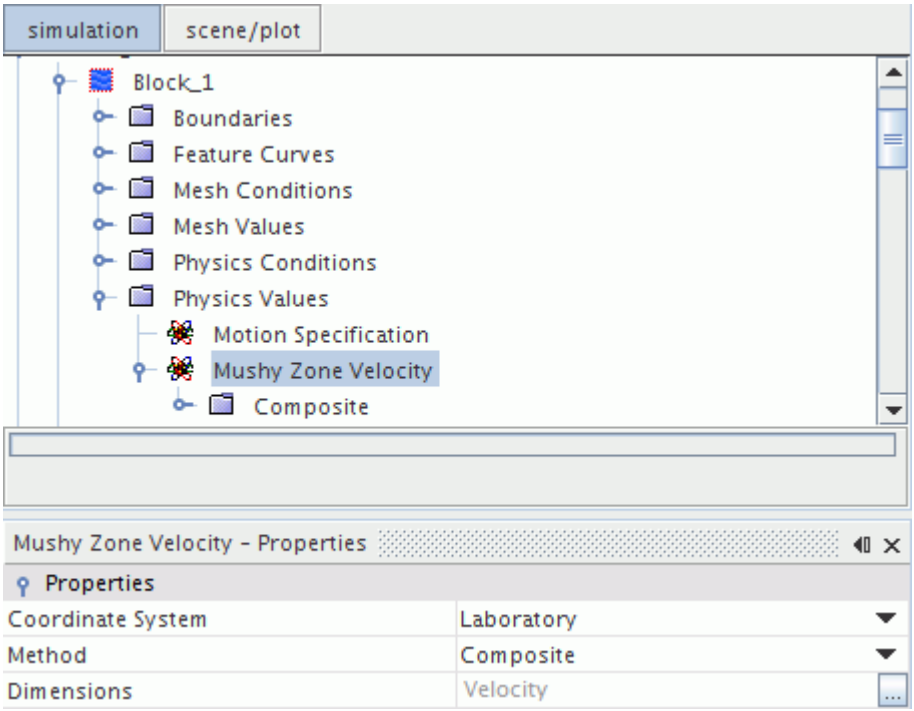
   The corresponding nodes are added to the **Physics Values** node of the region.
4. Specify the relative velocity field with respect to your selected reference frame:
   - For a region reference frame:

     Select the **Physics Values** > **Mushy Zone Relative Velocity** node and set its scalar profile method and value.

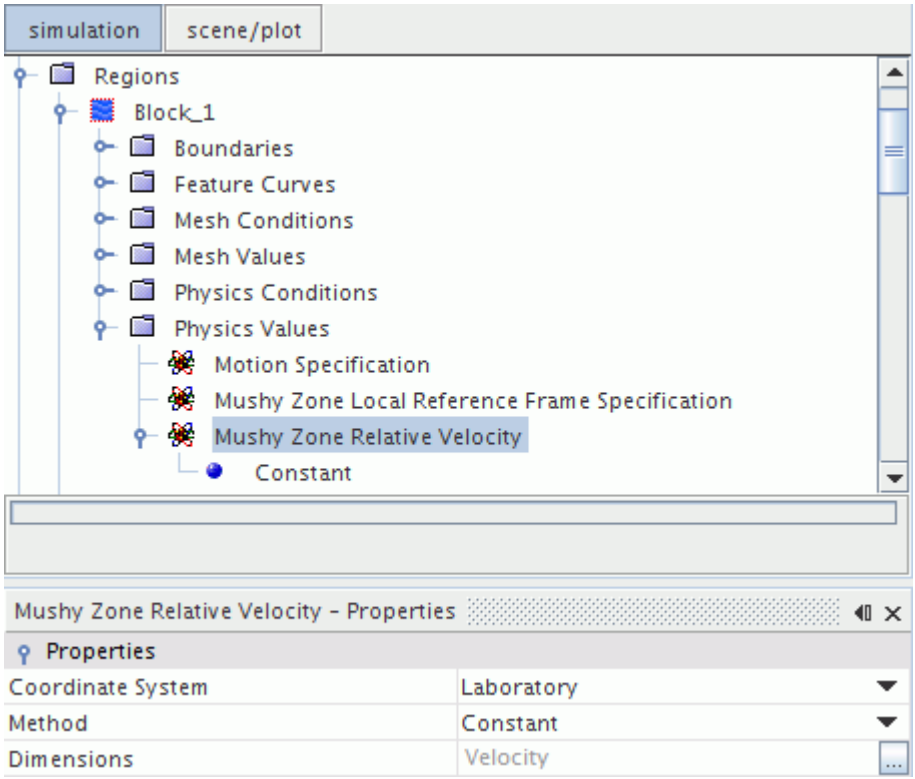


   - For the laboratory reference frame:

     Select the **Physics Values** > **Mushy Zone Velocity** node and set its scalar profile method and value.



- For the local reference frame:
  1. Select the **Physics Values** > **Mushy Zone Local Reference Frame Specification** node.
  2. In the *Properties* window, select the local reference frame that resides in **Tools** > **Reference Frames** to describe the motion.
  3. Select the **Physics Values** > **Mushy Zone Relative Velocity** node and set its scalar profile method and value.



When you specify the relative velocity value, observe the following difference between a stationary moving reference motion and a rigid body motion:

- For a rigid body motion, the mesh actually moves in the **Reference Frame for Rotation**. Therefore, you specify a relative velocity value of zero with respect to the **Coordinate System** in the *Mushy Zone Relative Velocity - Properties* window. The solidified liquid moves with the velocity specified for the rigid body motion.
- For a moving reference frame motion, the simulation is stationary and the mesh does not actually move. Therefore, you specify a non-zero relative velocity value with respect to the **Coordinate System** in the *Mushy Zone Relative Velocity - Properties* window. The solidified liquid moves with the velocity specified for the moving reference frame motion.

VOF Guidelines

This section describes the set-up and analysis procedures applicable to Volume Of Fluid (VOF) simulations.

See VOF Waves Guidelines for recommendations specific to wave simulations.

Meshing Considerations

Use the following meshing guidelines, where applicable:

- Make the mesh fine enough to resolve all geometry of interest and all important flow features.
- Refine the mesh locally to resolve flow separation and shed vortices.
- Either use a trimmed cell mesh and align the mesh with the free surface, or use polyhedral cells and inspect the results to make sure that they resolve the free surface properly.

- Set mesh size to meet the requirements for any wall treatment models.

To estimate the size that is needed for near-wall cells, calculate the value for y+ as:

$$y^+ = \frac{u^* y}{v} \tag{813}$$

where y is the distance from the wall, v is the fluid velocity and u^* is the reference velocity:

$$u^* \approx \sqrt{\frac{\tau_w}{\rho}}$$

where ρ is the fluid density and τ_w is the wall shear stress. τ_w is related to the skin friction coefficient $\overline{C}_f$:

$$\tau_w = \frac{1}{2}\overline{C}_f \rho U_\infty^2$$

where U_∞ is the free stream velocity.

Physics Models

When selecting physics models, use the following guidelines where applicable:

- Remember to select the Gravity model.
- Remember that the Cavitation and Surface Tension models are available with VOF.
- Define Eulerian phases for the fluids in the system.
- For internal flows, use K-Epsilon models.
- For external flows with little or no separation, use Spalart-Allmaras models, or K-Omega models as a second choice.
- For external flows with more severe separation, use K-Omega models.
- For flows with anisotropic turbulence, use Reynolds Stress Transport models.
- Use the All-y+ Wall Treatment whenever it is compatible with the selected model.
- Use the Interface Momentum Dissipation model in cases of:
  - Low-speed flows that are dominated by surface tension.
  - Wave models with unrealistically high velocities in the gas phase near free surface.
- Set reference densities to zero or to the density of the lightest phase. See Variable Density Flows.

Analysis Controls

Use the following analysis guidelines, where applicable:

- Use double precision if the single-precision results are unsatisfactory, or if the difference in density between phases is large.
- Set stopping criteria using key variables rather than a specific number of iterations.
- In most cases, run the analysis as transient, even if the final solution must be steady-state.
- For steady solutions, set the lower and upper Courant numbers *CFL_l* and *CFL_u* to large values, for example, *CFL_l* = `50`, *CFL_u* = `100`.

  This practice ensures that the HRIC scheme is always used at free surfaces.

  See Volume of Fluid (VOF) Properties.

- Use the default second-order space differencing scheme unless poor grid quality or other reasons require the use of a less accurate but more robust first-order discretization.
- Use the default first-order time differencing scheme when expecting a steady-state or weakly transient solution. Use the second-order scheme when accurate wave propagation or similar phenomena must be simulated.
- To estimate time-step, monitor the Courant number at the free surface. Target a Courant-number less than 0.7 for the first-order time discretization. Target a Courant-number less than 0.5 for the second-order time discretization. These targets apply for any cell face in the free-surface zone:

  $C = \frac{U\Delta t}{\Delta x} < 0.7$ (1st-order) or < 0.5 (2nd-order)

  where U is a characteristic velocity with one of these definitions:

  - For non-naval applications, U is the group velocity of a wave:

    $$U = \frac{1}{2}\sqrt{\frac{g\lambda}{2\pi}}$$

    where λ is the wavelength.
  - For naval applications, see VOF Waves Guidelines.

Convergence

To promote convergence, use the following guidelines where applicable:

- For naval applications, allow enough time for the flow to cross 5–10 vessel lengths.
- For transient cases, make sure that the solution is converging to reasonable tolerances at each time-step.
- Monitor overall quantities of interest, such as resistance, lift, sinkage, and trim. For steady simulations, these quantities converge to constant values when converged.
- Under normal conditions, you need not use the sharpening factor for free surfaces; the HRIC convection discretization scheme produces sharp interfaces by itself (typically within one cell). Use the sharpening factor only when the interface becomes smeared by splashing against walls or by wave overturning (such as in severe sloshing). Interface smearing occurs when the mesh is sufficiently fine to resolve all characteristic length scales of the free surface. Examples are when small droplets are produced by splashing or when the time-step is too large and the simulation cannot properly resolve the motion of the free surface. Typical values of sharpening factor for such cases are around 0.2.

VOF Formulation

The VOF model description assumes that all immiscible fluid phases present in a control volume share velocity, pressure, and temperature fields. Therefore, the same set of basic governing equations describing momentum, mass, and energy transport in a single-phase flow is solved.

Basic VOF Model Equations

The equations are solved for an equivalent fluid whose physical properties are calculated as functions of the physical properties of its constituent phases and their volume fractions.

The equations are:

$$\rho = \sum_i \rho_i \alpha_i \tag{815}$$

$$\mu = \sum_i \mu_i \alpha_i \tag{816}$$

$$c_p = \sum_i \frac{\left(c_p\right)_i \rho_i}{\rho} \alpha_i \tag{817}$$

where:

$\alpha_i = \frac{V_i}{V}$

is the volume fraction and ρ_i, μ_i and $\left(c_p\right)_i$ are the density, molecular viscosity and specific heat of the i th phase.

The conservation equation that describes the transport of volume fractions α_i is:

$$\frac{d}{dt} \int_V \alpha_i dV + \int_S \alpha_i \left(\mathbf{v} - \mathbf{v}_g\right) \cdot d\mathbf{a} = \int_V \left(s_{\alpha_i} - \frac{\alpha_i}{\rho_i} \frac{D\rho_i}{Dt}\right) dV \tag{818}$$

where s_{α_i} is the source or sink of the i th phase, and $D\rho_i/Dt$ is the material or Lagrangian derivative of the phase densities ρ_i.

If a non-zero sharpening factor is specified, an additional term is added to the VOF transport equation:

$$\nabla \cdot \left(\mathbf{v}_{c_i} \alpha_i \left(1 - \alpha_i\right)\right) \tag{819}$$

where:

- α_i is the volume fraction of phase i
- $\mathbf{v}_{c_i}$ is defined as follows:

  $\mathbf{v}_{c_i} = C_\alpha \times |\mathbf{v}| \frac{\nabla \alpha_i}{\left|\nabla \alpha_i\right|}$
- C_α is the sharpening factor that is specified in the **Volume of Fluid (VOF)** node properties.
- $\mathbf{v}$ is the fluid velocity

If there is a large time variation of phase volume fractions α_i, there is a large time variation of the mixture density ρ which features in the continuity equation, Eqn. (1461). Since this unsteady term cannot be linearized in terms of pressure and velocity, it acts as a large source term which can be “unpleasant” for a numerical treatment within a segregated solution algorithm. Therefore, Eqn. (1461) is rearranged in the following, non-conservative form:

$$\int_A \left(\mathbf{v} - \mathbf{v}_g\right) \cdot d\mathbf{a} = \sum_i \int_V \left(s_{\alpha_i} - \frac{\alpha_i}{\rho_i} \frac{D\rho_i}{Dt}\right) dV \tag{820}$$

In the case when phases have constant densities and have no sources, the continuity equation reduces to $\nabla \cdot \mathbf{v} = 0$.

Contents:

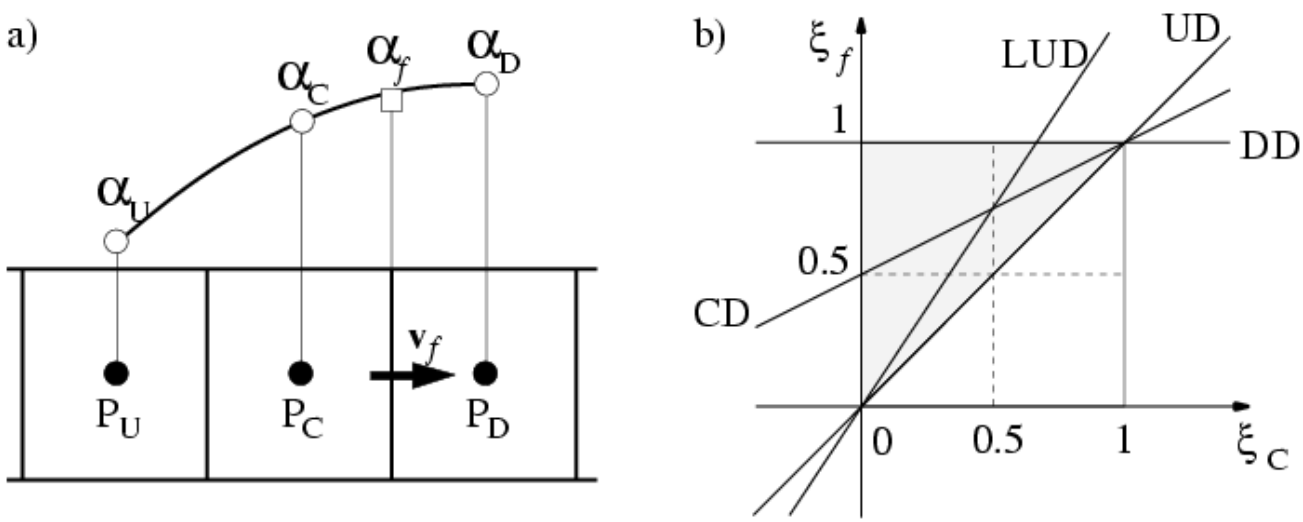
The Normalized Variable Diagram
The HRIC Convection Discretization Scheme
Surface Tension Formulation
Latent Heat of Vaporization
Wall Porosity Formulation
Evaporation-Condensation Formulation
Boiling Formulation
Gas Dissolution and Cavitation Formulation
Melting-Solidification Formulation

The Normalized Variable Diagram

The normalized variable diagram (NVD) is useful for analyzing boundedness properties of convective discretization schemes.

Part (a) of the figure below shows three cells in the vicinity of a cell face f, across which the velocity $\mathbf{v}_f$ is known. The nodal variable values are labeled α_D, α_C and α_U, representing the downwind, central, and upwind positions relative to each other.

The diagram below shows the upwind, downwind, and central cells that are used in the analysis (left) and convection boundedness criterion in the NVD diagram (right):



The normalized variable $\xi(\mathbf{r}, t)$ in the vicinity of face f is defined as:

$$\xi(\mathbf{r}, t) = \frac{\alpha(\mathbf{r}, t) - \alpha_U}{\alpha_D - \alpha_U} \tag{821}$$

The normalized face value:

$$\xi_f = \frac{\alpha_f - \alpha_U}{\alpha_D - \alpha_U} \tag{822}$$

calculated by any differencing scheme that uses only nodal values of α at points U, C, and D, can be written in the form:

$$\xi_f = \xi_f(\xi_C) \tag{823}$$

where:

$$\xi_C = \frac{\alpha_C - \alpha_U}{\alpha_D - \alpha_U} \tag{824}$$

To avoid non-physical oscillations in the solution, α_C (and consequently α_f) has to be locally bounded between α_U and α_D, meaning:

$$\alpha_U \leq \alpha_C \leq \alpha_D \tag{825}$$

or

$$\alpha_D \leq \alpha_C \leq \alpha_U \tag{826}$$

If this criterion is satisfied at every point in the domain, the entire solution is free of non-physical oscillations.

The boundedness criterion for convection differencing schemes can be presented in the NVD diagram, showing ξ_f as a function of ξ_C, as the hatched region in part (b) of the above Figure (including the line $\xi_f = \xi_C$). It can also be expressed through the following conditions:

- For $0 \leq \xi_C \leq 1$ the bounded region lies above the line $\xi_f = \xi_C$ and below $\xi_f = 1$.
- For $\xi_C < 0$ and $\xi_C > 1$, ξ_f is equal to ξ_C.

NVD is concerned with convective transport alone. If sources or sinks are present, the conditions that are given in Eqn. (825) and Eqn. (826) can be violated. The importance of the boundedness criterion is especially clear in the case of variables which have physical bounds; for example, the phase volume fraction cannot become negative, or larger than unity.

The HRIC Convection Discretization Scheme

An important quality of an immiscible phase mixture (for example, air and water) is that the fluid components always remain separated by a sharp interface. The High-Resolution Interface Capturing (HRIC) scheme is designed to mimic the convective transport of immiscible fluid components, resulting in a scheme that is suited for tracking sharp interfaces.

A simple higher-order scheme (for example, CD or LUD) would fail in approximating large spatial variations of phase volume fractions, which are best represented by the Heaviside unit step function.

The scheme is based on the NVD that was shown earlier. The normalized face value ξ_f is calculated as:

$$\xi_f = \begin{cases} \xi_C & \text{if} \quad \xi_C < 0 \\ 2\xi_C & \text{if} \quad 0 \le \xi_C < 0.5 \\ 1 & \text{if} \quad 0.5 \le \xi_C < 1 \\ \xi_C & \text{if} \quad 1 \le \xi_C \end{cases} \tag{827}$$

The calculated value of ξ_f is further corrected according to the local Courant number:

$$Cu = \frac{\mathbf{v} \cdot \mathbf{a}_f}{V_{P_C}} \delta t \tag{828}$$

This correction is to take into account the availability criterion. This criterion says that the amount of one fluid that is convected across a cell face during a time-step is always less than or equal to the amount available in the donor cell. This correction is made according to the expressions:

$$\xi_f{}^* = \begin{cases} \xi_f & \text{if} \quad Cu < Cu_l \\ \xi_C + \left(\xi_f - \xi_C\right)\frac{Cu_u - Cu}{Cu_u - Cu_l} & \text{if} \quad Cu_l \le Cu < Cu_u \\ \xi_C & \text{if} \quad Cu_u \le Cu \end{cases} \tag{829}$$

and it plays a role only in transient simulations. Default values for Cu_l and Cu_u are 0.5 and 1. They are introduced to control blending of HRIC and UD schemes depending on the Courant number. For values of $Cu < Cu_l$ HRIC is used, for $Cu_l < Cu < Cu_u$ a blend of HRIC and UD is used, and for $Cu_u < Cu$ UD is used.

The blending is introduced in the case when a large time variation of the free surface shape is present, and the time-step is too large to resolve details of it. It brings stability and robustness to the scheme, and is especially important when you use the second-order discretization in time. In this case, smaller values of Cu_u and Cu_l help to promote convergence. Smaller values activate UD sooner, and the calculation is more stable. At the same time, the interface is more “smudged” in regions with large Cu numbers.

For problems which have a steady-state solution, the values of Cu_u and Cu_l must be large, such that HRIC is used irrespective of the time-step selected. Any number larger than the maximum CFL is appropriate.

The front sharpening aspect of the downwind scheme is just what is needed when the interface is perpendicular to the flow. However, when the interface is parallel to the flow direction, the downwind scheme tends to wrinkle it. The final correction of ξ_f is based on the angle θ between the normal to the interface $\mathbf{n}_i$ and the cell-face surface vector $\mathbf{a}_f$, as shown below:

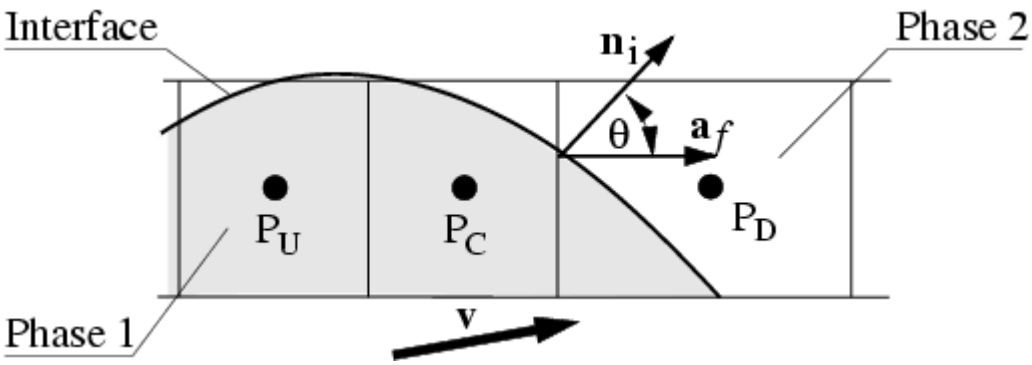
$$\xi_f{}^{**} = \xi_f{}^*\left(\cos\theta\right)^{C_\theta} + \xi_C\left(1 - \left(\cos\theta\right)^{C_\theta}\right) \tag{830}$$

The default value of the angle factor C_θ is 0.05. If the free surface is not smooth and not following the grid lines, increase its value.

The cell-face value ξ_f is now calculated as:

$$\alpha_f^{HRIC} = \xi_f{}^{**}\left(\alpha_D - \alpha_U\right) + \alpha_U \tag{831}$$

The following diagram shows the interface between two fluids and the notation used.



Surface Tension Formulation

The surface tension force is a tensile force tangential to the interface separating two fluids. It works to keep the fluid molecules at the free surface in contact with the rest of the fluid.

The surface tension force is an interfacial force, which is modeled as a volumetric force using the Continuum Surface Force (CSF) approach of Brackbill and others [522].

The magnitude of the surface tension force depends mainly on the nature of the fluid pair and on temperature. For a curved interface, the surface tension force $\mathbf{f}_\sigma$ can be resolved into two components:

$$\mathbf{f}_\sigma = \mathbf{f}_{\sigma,n} + \mathbf{f}_{\sigma,t} \tag{832}$$

Where:

$$\mathbf{f}_{\sigma,n} = \sigma K \mathbf{n} \tag{833}$$

and:

$$\mathbf{f}_{\sigma,t} = \frac{\partial \sigma}{\partial t} t \tag{834}$$

where:

- σ is the surface tension coefficient.
- $\mathbf{n}$ is the unit vector normal to the free surface and directed from liquid to gas.
- t is the unit vector in the tangential direction to the free surface.
- K is the mean curvature of the free surface.

For a constant surface tension coefficient σ, the tangential force is zero and the surface tension results in a force normal to the interface ($\mathbf{f}_\sigma = \mathbf{f}_{\sigma,n}$).

When the surface tension coefficient varies along the surface, which can be due to temperature differences, the tangential part does not vanish. In this case, Marangoni or Bénard convection can develop tangential to the free surface.

In the VOF multiphase model case, the surface tension force is calculated according to the continuum surface force (CSF) model. That is, it uses the smooth field of the phase volume fraction α_i to calculate a vector normal to the interface:

$$\mathbf{n} = \nabla\alpha_i \tag{835}$$

The curvature of the interface can therefore be expressed in terms of the divergence of the unit normal vector n, as follows:

$$\kappa = -\nabla\cdot\frac{\nabla\alpha_i}{|\nabla\alpha_i|} \tag{836}$$

Now the normal component of the surface tension force $\mathbf{f}_{\sigma,n}$ can be expressed as:

$$\mathbf{f}_{\sigma,n} = -\sigma\,\nabla\cdot\left(\frac{\nabla\alpha_i}{|\nabla\alpha_i|}\right)\nabla\alpha_i \tag{837}$$

When the surface tension coefficient σ is constant, $\mathbf{f}_\sigma = \mathbf{f}_{\sigma,n}$.

For a variable surface tension coefficient, the tangential force is evaluated as:

$$\mathbf{f}_{\sigma,t} = (\nabla\sigma)_t|\nabla\alpha_i| \tag{838}$$

where $(\nabla\sigma)_t$ is the gradient of the surface tension coefficient in the tangential direction.

Dynamic Contact Angle

The common quantity of many different dynamic contact angle correlations is the triple line velocity. The triple line is the line where the wall and both fluid phases are in mutual contact. This quantity is related to the dimensionless capillary number Ca as:

$$Ca = \frac{V\mu}{\sigma} \tag{839}$$

where:

- V is the triple line characteristic velocity
- μ is the dynamic viscosity of the primary phase (usually a liquid phase)
- σ is the surface tension force

The triple line velocity is defined as:

$$V = -\langle\mathbf{V}, \hat{n}_t\rangle \tag{840}$$

where:

- $\mathbf{V}$ is the relative velocity of the fluid and the corresponding wall at the triple line.
- $\hat{n}_t$ is the normalized wall tangent pointing in the same direction as the volume fraction gradient of the primary phase ($\nabla\alpha$)

Kistler Correlation

STAR-CCM+ implements the Kistler correlation [559], which is an empirical dynamic contact angle correlation based on the capillary number Ca and utilizing the Hoffman function. The Kistler contact angle is defined as:

$$\theta_k = f_{Hoff}\left(Ca + f_{Hoff}^{-1}(\theta_s)\right) \tag{841}$$

where f_{Hoff} is the Hoffman function:

$$f_{Hoff}(x) = \cos^{-1}\left(1 - 2\tanh\left(5.16\left(\frac{x}{1 + 1.31x^{0.99}}\right)^{0.706}\right)\right) \tag{842}$$

with f_{Hoff}^{-1} being its inverse.

Note: STAR-CCM+ uses a slightly modified definition of the Hoffman function to achieve a closed form for its inverse function. In the modified Hoffman function, the $x^{0.99}$ term in the denominator is changed to a simple x. The closed form of the inverse function is valid for contact angle values in the range 0–176 degrees. Static advancing or receding contact angles that are larger than the upper bound are limited to 176 degrees.

The static contact angle θ_s is inserted into Eqn. (841) as either the static advancing or receding contact angle, depending on the sign of the capillary number. θ_s is specified as the *Advancing Contact Angle* or *Receding Contact Angle* in the Contact Angle Properties.

To enhance the stability of the method that is implemented in STAR-CCM+, a range for the equilibrium capillary number (Ca_{eq}) can be defined. Within the specified range $-Ca_{eq} < Ca < Ca_{eq}$, the resulting dynamic contact angle is blended with the equilibrium contact angle (θ_e) as a weighted average:

$$\theta_d = f\theta_e + (1 - f)\theta_k \tag{843}$$

While θ_e is a user-specified value, the factor f is determined within the Ca_{eq} range as:

$$f = 0.5 + 0.5\cos\left(\frac{Ca}{Ca_{eq}}\pi\right) \tag{844}$$

Latent Heat of Vaporization

The specific latent heat of vaporization h_{lat} is the amount of heat that is required to convert unit mass of a liquid into the vapor without a change in temperature.

The temperature at which vaporization takes place is called saturation temperature T_{sat}.

The latent heat is also the difference between the enthalpies of the vapor $h_v(T_{sat})$ and the liquid $h_l(T_{sat})$ at the temperature of saturation:

$$h_{lat} = h_v(T_{sat}) - h_l(T_{sat}) \tag{845}$$

$$h_v(T_{sat}) = h_{0_v} + C_{p_v}\left(T_{sat} - T_{0_v}\right) \tag{846}$$

$$h_l(T_{sat}) = h_{0_l} + C_{p_l}\left(T_{sat} - T_{0_l}\right) \tag{847}$$

In problems with boiling, the heats of formation h_{0_l} and h_{0_v}, and the standard state temperatures T_{0_l} and T_{0_v} of the liquid and the vapor phase must be specified such that the above expression is satisfied. The latent heat is specified in this way. For stability reasons, set either h_{0_v} or h_{0_l} to zero, and the other to an appropriate value.

See Also:

Using the Standard State Temperature

Wall Porosity Formulation

The implementation of the VOF phase-specific wall porosity model is analogous to that of a porous baffle.

The velocity across the porous boundary is calculated from the known pressure drop across it.

This pressure drop is given as:

$$\Delta p = -\rho\left(\alpha\left|v_n\right| + \beta\right)v_n \tag{848}$$

where ρ is the density of the phase, α and β are user supplied constants and v_n is the normal velocity across the porous boundary. STAR-CCM+ calculates the pressure drop from a specified external ambient pressure on the region. Eqn. (848) is then applied to calculate the normal velocity. This normal velocity is used to predict the amount of mass leaving through the boundary face within a given time-step.

Evaporation-Condensation Formulation

This section describes the Evaporation-Condensation model formulation.

The evaporation rate (kg/s m^2) for a particular component i is:

$$\dot{m}_i = -\frac{\rho_G D_{G,i} \left.\frac{\partial Y_{G,i}}{\partial n}\right|_s}{1 - \sum_{j=1}^{N_v} Y_{G,j}^s} \tag{849}$$

The denominator expresses the influence of the Stefan flow and Y^s indicates the component mass fraction at the surface. N_v is the total number of interacting components, that is, components undergoing evaporation or condensation. The G subscript denotes a gas phase variable.

The vapor pressure p_i of a component is:

$$p_i = a_i p_i^* \tag{850}$$

where p_i^* is the vapor pressure of the pure component i and a_i the activity, which is dependent on the liquid mixture. This expression can be written as:

$$p_i = \gamma_i X_{L,i} p_i^* \tag{851}$$

Raoult's law states that $\gamma_i \approx 1$, an approximation that is used here. Given the vapor pressures of the components, their molar fractions at the interface can be derived from:

$$X_{G,i}^s = \frac{p_i}{p} \tag{852}$$

where p is the ambient pressure. The conversion to interface vapor mass fraction is done through:

$$Y_{G,i}^s = \frac{X_i^s W_i}{\sum_{j=1}^{N_v} X_j^s W_j + \sum_{j=1}^{N_{G,p}} X_j^s W_j} \tag{853}$$

where N_G is the total number of components in the gas phase. While the molar fraction of the evaporation components is known, it is unknown for the inert species (which do not undergo evaporation or condensation). A good approximation is the introduction of a background molar weight calculated a short distance away from the interface:

$$W_{bg} = \frac{\sum_{j=1}^{N_{G,p}} X_j W_j}{\sum_{j=1}^{N_{G,p}} X_j} \tag{854}$$

The interfacial background molar fraction is:

$$X_{bg}^s = 1 - \sum_{j=1}^{N_v} X_j^s \tag{855}$$

Therefore the interfacial vapor mass fraction can be approximated as:

$$Y_{G,i}^s \approx \frac{X_i^s W_i}{\sum_{j=1}^{N_v} X_j^s W_j + X_{bg}^s W_{bg}} \tag{856}$$

The numerical solution of Eqn. (849) begins from the definition of the following scalar function ϕ_i, for every interacting component:

$$\phi_i = \alpha_L Y_{G,i}^s + \alpha_G Y_{G,i} \tag{857}$$

where α_L is the liquid volume fraction and α_G is the gas volume fraction. The approximated evaporation rate $\dot{M}'_{i,c}$ for one cell can be obtained as:

$$\dot{M}'_{i,c} \approx -\frac{\rho_G D_{G,i} \nabla \phi_i \nabla \alpha_L V_c}{1 - \sum_{j=1}^{N_v} Y_{G,j}^s} \tag{858}$$

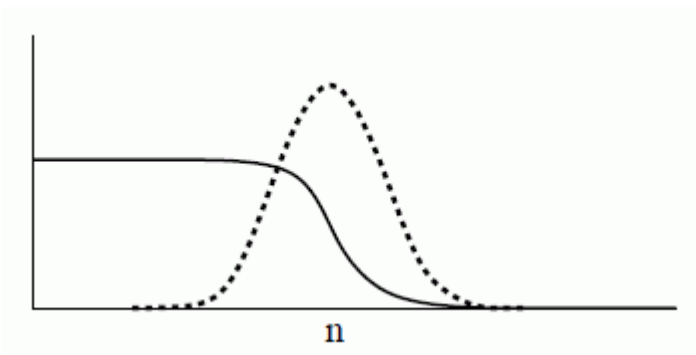
using the fact that the integral of $\nabla \alpha_L$, the gradient of liquid fraction, over a certain volume expresses the area of the interface it contains [522], [533].

Comparing Eqn. (849) and Eqn. (858), it is seen that $\nabla\phi_i$ expresses the effect of $\nabla Y_{G,i}$. A lowest order analysis shows that the relative error of the evaporation rate, in quasi-steady state, behaves as:

$$\varepsilon_{rel} \approx \frac{1}{H|\nabla\alpha|} \tag{859}$$

where H expresses the thickness of the boundary layer above the interface.

As such, the accuracy depends on how well the location of the interface is determined in comparison with the boundary layer. When the interface is not smeared, $|\nabla\alpha|^{-1}$, the reciprocal of the magnitude of the gradient of volume fraction, is a measure of the mesh size. This condition highlights the need for a sufficiently fine grid, to model the transport of species away from and towards the interface accurately. Similar considerations are valid for gradients of the temperature and pressure near the interface.



The sources for species, phases, and continuity are, based on Eqn. (858), distributed over those cells that make up the interface, that is with non-vanishing gradient of the volume fraction, as shown in the above graph depicting the typical behavior of α_L (solid line) and $-|\nabla\alpha_L|$ (dashed line) at a free surface interface, along the surface normal coordinate n.

In order to avoid undershoot or overshoot of these sources, $\dot{M}'_{i,c}$ is weighted along the interface according to:

$$\dot{M}'_{i,c} = \begin{cases} 2\alpha_L \dot{M}''_{i,c} & if \quad M'_{i,c} > 0 \\ 2\alpha_G \dot{M}''_{i,c} & if \quad M'_{i,c} < 0 \end{cases} \tag{860}$$

Boiling Formulation

This section contains the formulation of the boiling models.

This section describes:

- Condensation and boiling at a vapor-liquid interface away from surfaces
- Boiling at a solid surface
- The Rohsenow Boiling model
- The Transition Boiling model

Condensation and Boiling at a Vapor-Liquid Interface Away from Surfaces

The heat transfer to the interface between the phases is used to compute the rate of evaporation or condensation $\dot{m}_{ec}$. The following are assumed:

- The vapor bubbles are on the saturation temperature T_{sat}
- The temperature of the liquid T_l is approximately the temperature of the mixture T
- All the heat flux from the liquid to the interface is used in mass transfer (that is, evaporation or condensation):

$$\dot{m}_{ec} = \frac{\mathrm{HTCxArea}\left(T - T_{sat}\right)}{h_{lat}} \tag{861}$$

  where h_{lat} is the latent heat of vaporization. $\mathrm{HTCxArea}$ is the heat transfer coefficient between the vapor bubbles and the surrounding liquid, which is multiplied by the specific contact area (contact area per unit volume) between the two.

Boiling at a Solid Surface

When a liquid is in contact with a solid surface maintained at temperatures T_{wall} above the saturation temperature of the liquid, boiling eventually occurs at that liquid-solid interface. In this case, boiling occurs in three characteristic stages:

- Nucleate boiling involves a creation and growth of vapor bubbles on a heated surface, which rise from discrete points on a surface. The temperature of the surface is only slightly above the saturation temperature of the liquid. In general, the number of nucleation sites increases with an increasing surface temperature. An increased surface roughness can create more nucleation sites, while an exceptionally smooth surface can result in superheating.
- Film boiling occurs when the critical heat flux is exceeded and a continuous vapor film covers the heated surface. The vapor layer has a lower thermal conductivity so the vapor layer typically “insulates” the surface.
- Transition boiling occurs at surface temperatures between the maximum attainable in nucleate and the minimum attainable in film boiling.

There are two distinct choices for modeling boiling in STAR-CCM+, namely the Rohsenow and Transition Boiling models. The former uses the Rohsenow nucleate boiling model applicable for boiling at relatively low solid temperatures. A film boiling model is also integrated into the Rohsenow implementation in STAR-CCM+ to facilitate its use with boiling at high solid temperatures. The Transition Boiling model has expressions for nucleate and transition boiling. The boiling model that used to be available in previous versions of STAR-CCM+ has now been formally named the Rohsenow Boiling Model. Once VOF Boiling is chosen in the continuum model selection, you can choose between the two models.

The Rohsenow Boiling Model

This model works with nucleate and film boiling.

The Nucleate Boiling Model

See Single-Phase Boiling Formulation.

The Film Boiling Model

At a sufficiently high wall temperature, a layer of vapor is created above the heated wall. As a result the liquid is no longer in contact with the heated wall. The evaporation occurs at the vapor liquid interface instead of at the heated wall. The vapor film that is created between the liquid and the heated wall acts as an insulator, which slows down the heat transfer process considerably. This process is known as film boiling.

The film boiling model, which is used in a combination with the VOF multiphase model, assumes that the film thickness is resolved in one or more cells. That is, the vapor occupies one or more cells close to the heated wall. In this case, the expressions for the wall heat flux are the same as in the case of single-phase flows. If it is not possible to have a mesh which resolves the film thickness, you specify the value of the vapor volume fraction $\alpha_{filmBoiling}$, which indicates when the film boil starts. For values of the vapor volume fraction smaller than $\alpha_{filmBoiling}$ the nucleate boiling is assumed, and for larger values it is assumed that the film boiling takes place. If the film is resolved and the cell next to the heated wall is filled with vapor, the results do not depend on values of $\alpha_{filmBoiling}$ from 0 through 1.

The Transition Boiling Model

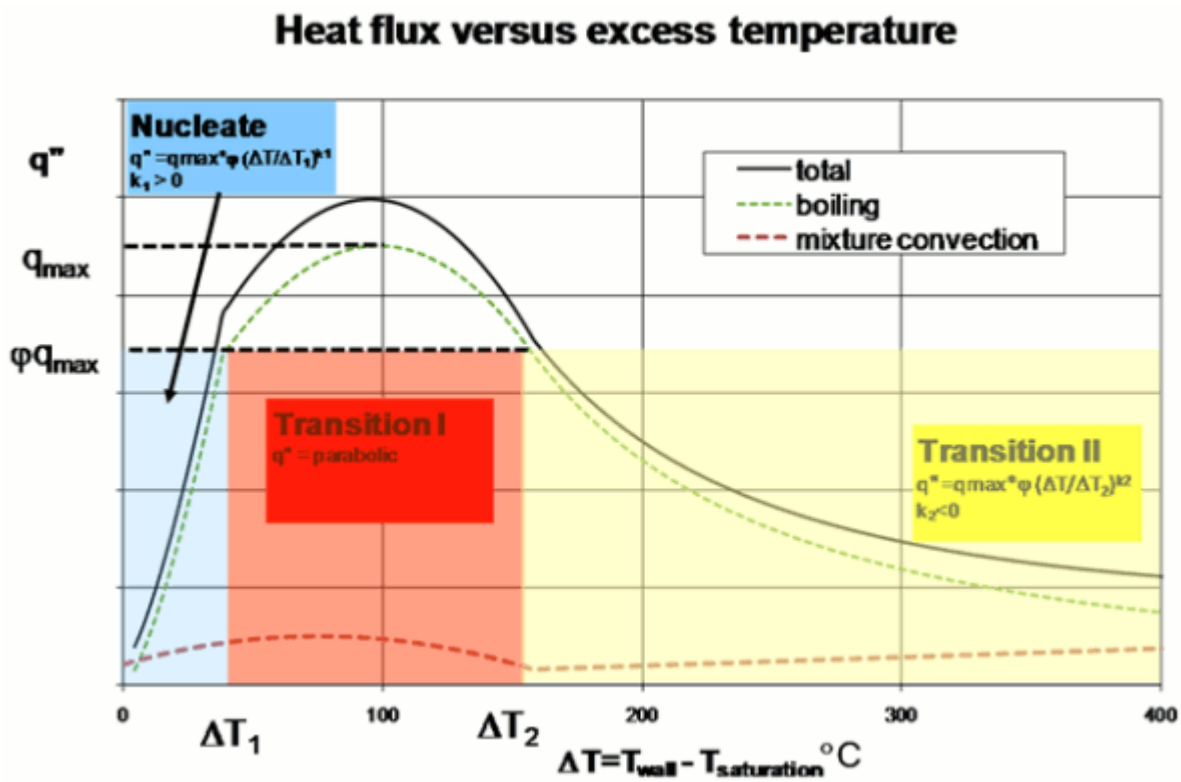
You can specify the constants of this boiling model. The equations are as follows:

$$q_{boiling}(\Delta T) = q_{\max} S \phi \left(\frac{\Delta T}{\Delta T_1}\right)^{K_1} \qquad 0 \le \Delta T \le \Delta T_1 \tag{862}$$

$$q_{boiling}(\Delta T) = q_{\max} S \left[1 - 4(1-\phi)\left(\frac{\Delta T - \Delta T_{\max}}{\Delta T_2 - \Delta T_1}\right)^2\right] \qquad \Delta T_1 \le \Delta T \le \Delta T_2 \tag{863}$$

$$q_{boiling}(\Delta T) = q_{\max} S \phi \left(\frac{\Delta T}{\Delta T_2}\right)^{-K_2} \qquad \Delta T_2 \le \Delta T \tag{864}$$

The first of the above equations simulates nucleate boiling, while the remaining two simulate transition I and transition II boiling. The three regimes are shown graphically in the following figure.



Different regimes of the new boiling transition boiling model

There are five empirical constants in the above model, namely, K_1, K_2, ΔT_1, ΔT_2, and $q_{\max}$, all of which are positive. ΔT_{max} is $\frac{1}{2}(\Delta T_1 + \Delta T_2)$.The constant ϕ is set equal to 0.75. Further, ΔT_2 must be greater than ΔT_1. (In the above expressions K_2 is positive even though it is a negative exponent because the sign is accounted for in the implementation.) The scale factor S in the above expressions is not treated as an independent sixth constant, as it allows you to scale $q_{\max}$ up or down for a given boundary. All the constants are likely to be application-specific. Hence, it is recommended that default values for the constants be tuned with credible measurements. The default values are from fits of boiling experiments of Ellion [529] using an apparatus with a particular geometry.

As mentioned in the previous section, $\alpha_{filmBoiling}$ (that you specify in the input) is the metric used to toggle between the nucleate and film boiling regimes in the Rohsenow Boiling Model. However, in the Transition Boiling Model, $\alpha_{filmBoiling}$ is not used; instead ΔT_1 and ΔT_2 are the metrics that switch between the three expressions (or regimes).

The total heat flux at the wall is the sum of the heat fluxes due to boiling, convection of the liquid-vapor mixture and thermal radiation. In the above equations, $q_{\max}$ is the maximum of the boiling heat flux component; it is not the critical heat flux which is a maximum of the total heat flux. The point on the curve corresponding to the minimum of the total heat flux is the Leidenfrost point and it occurs in the Transition II boiling regime. At the Leidenfrost point, vapor completely covers the surface and heat transfer is primarily driven by mixture convection and radiation with a negligible boiling contribution.

In the CFD implementation of the above boiling model, the boiling heat flux contribution to the total heat flux is considered to be zero when there is no liquid to boil in any cell that is next to the wall. Further, if the fluid

temperature T_f is above the saturation temperature T_{sat} the boiling heat flux reduces by the factor $\left(T_w - T_f\right)/\left(T_w - T_{sat}\right)$.

Gas Dissolution and Cavitation Formulation

STAR-CCM+ uses the same homogeneous seed-based approach for both cavitation and gas dissolution models. Although these zero-dimensional approaches do not capture all of the physical phenomena present, they are proven methods for use in industrial design studies.

STAR-CCM+ provides three cavitation models:

- **Full Rayleigh–Plesset**

  The full cavitation model implements the full Rayleigh–Plesset equation and includes the influence of bubble growth acceleration, as well as viscous and surface tension effects.

  See Seed-Based Mass Transfer and Full Rayleigh–Plesset Cavitation Model.
- **Schnerr-Sauer**

  This model is based on a reduced Rayleigh–Plesset equation, and neglects the influence of bubble growth acceleration, viscous effects, and surface tension effects.

  See Seed-Based Mass Transfer and Schnerr-Sauer Cavitation Model.
- **Homogeneous Relaxation**

  This model is based on a finite rate equation with an empirical time scale formulation, and is suitable for modeling phenomena which occur in thermal non-equilibrium, such as flash boiling.

  See Homogeneous Relaxation Model Formulation.

STAR-CCM+ also provides a gas dissolution model that is based on the Rayleigh–Plesset equation.

See Seed-Based Mass Transfer and Bubble Growth Rate Model for Gas Dissolution.

Overview of Mass Transfer Models

This VOF method handles a particular class of physical models: models that describe phase change, and corresponding mass transfer, between two physical phases (for example, a liquid and gaseous phase).

In general, there are two ways to handle phase change:

- Interface resolution

  This method requires the capturing of fine details, such as the micro-bubbles in a fluid, with multiple cells. Therefore it is numerically expensive.

  The Evaporation/Condensation model uses an interface resolving method.
- Homogeneous approach

  This method makes certain assumptions regarding the interface between the phases, on a subcell level. This method is numerically more efficient, but has limitations regarding the physics that can be resolved.

  The Boiling, Gas Dissolution, and Cavitation models use a homogeneous approach.

Seed-Based Mass Transfer

The cavitation models and the gas dissolution model in STAR-CCM+ are based on the Rayleigh–Plesset equation.

In order to be able to account for the effects of liquid purity (that is, the content of solid particles and dissolved gas), it is assumed that cavitation nuclei (seeds) are present in liquid. The following modeling assumptions are made:

- Seed bubbles are spherical and uniformly distributed in liquid, as characterized by the number n_0 (the number of bubbles per unit volume of liquid).
- All seeds initially have the same radius.

It is assumed that the number of bubbles in a control volume is proportional to the amount of liquid, so there is no need to follow each bubble and track its motion in space.

Under the above assumptions, the number of bubbles within a control volume at any time is:

$$N = n_0 c_l V \tag{865}$$

where c_l is the volume fraction of liquid within a control volume of volume V.

The total vapor volume inside the control volume equals:

$$V_v = N V_b \tag{866}$$

where V_b is the volume of one bubble, and:

$$V_b = \frac{4}{3}\pi R^3 \tag{867}$$

with R being the local bubble radius.

The vapor volume fraction c_v can now be expressed in terms of n_0, R, and volume fraction of cavitating liquid, c_l, following the definition:

$$c_v = \frac{V_v}{V} = \frac{N V_b}{V} = \frac{4}{3}\pi R^3 n_0 c_l \tag{868}$$

When the volume fraction of vapor is known, the local radius of bubbles can be computed from this expression:

$$R = \left(\frac{3c_v}{4\pi n_0 c_l}\right)^{1/3} \tag{869}$$

The modeling framework that is outlined above is used to define the rate of vapor production or consumption in the mass conservation equation. The rate of bubble radius change is the key parameter, as vapor production results in bubble growth and vapor consumption results in bubble collapse. Another parameter is the amount of liquid that can cavitate in the control volume. Obviously, any increase of the vapor mass is accompanied by a corresponding decrease in the liquid mass.

The vapor bubbles are moving with the flow. Therefore, the rate at which vapor is created at any particular time is approximately the rate at which the volume of bubbles present in the control volume at that time is changing. The change of volume of an individual bubble is:

$$\frac{\mathrm{d}V_b}{\mathrm{d}t} = 4\pi R^2 \frac{\mathrm{d}R}{\mathrm{d}t} = 4\pi R^2 v_r \tag{870}$$

where $v_r = \mathrm{d}R/\mathrm{d}t$ denotes the bubble growth velocity.

The rate of growth (or collapse) of all bubbles present in a control volume represents the source term in the equation for vapor volume fraction, that is:

$$Q_\mathrm{v} = N\frac{\mathrm{d}V_b}{\mathrm{d}t} = 4\pi n_0\left(1 - c_\mathrm{v}\right)VR^r v_r \tag{871}$$

The above source term represents the contribution from cavitation only; vapor can also be produced by boiling. For example, the same flow can have vapor generation at a hot wall due to boiling and vapor generation at a sharp corner due to locally low pressure and cavitation. In such cases, the source terms that result from different models must be added up.

Seed-based mass transfer models are used for gas dissolution and cavitation. The seed-based models are based on [545]. The two interacting phases are denoted as l and g, with mass transfer positive from l to g.

The volume fraction of the gas can be expressed as:

$$\alpha_g = \frac{V_g}{V} = \frac{V_g}{V_g + V_l} \tag{872}$$

where V is the control volume, and V_g and V_l denote the respective parts of the volume that the gas and the liquid phases occupy.

Within the control volume, the gas phase is assumed to be present in the form of bubbles. Each bubble has the same radius R. The number density of seeds is defined as n_0, which corresponds to the number of bubbles per unit volume.

Therefore the following relationship holds between the gas and liquid phases:

$$V_g = n_0 V_l \frac{4}{3}\pi R^3 \tag{873}$$

The volume fraction of gas, α_g, becomes:

$$\alpha_g = \frac{n_0 \frac{4}{3}\pi R^3}{1 + n_0 \frac{4}{3}\pi R^3} \tag{874}$$

The gas bubble radius can be expressed as:

$$R^3 = \frac{\alpha_g}{n_0 \frac{4}{3}\pi\left(1 - \alpha_g\right)} \tag{875}$$

The seed diameter $D_0 = 2R_0$ (which is user-specified) provides a lower limit R_{min} for the bubble size.

The mass transfer rate per unit volume $kg/m^3 s$ can be written as:

$$\dot{m} = n_0 \alpha_l 4\pi \rho_g R^2 \frac{DR}{Dt} \tag{876}$$

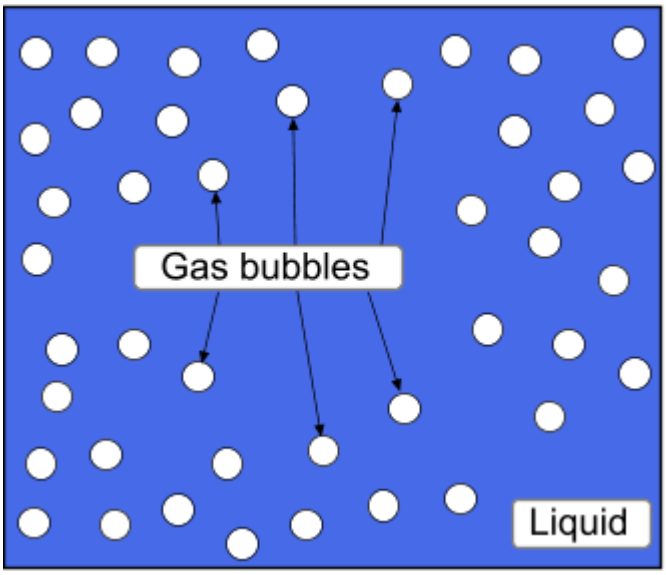
such that liquid disappears with a volume rate m^3/m^3s:

$$\dot{V}_l = -\frac{\dot{m}}{\rho_l} \tag{877}$$

and gas is generated with a volume rate:

$$\dot{V}_g = \frac{\dot{m}}{\rho_g} \tag{878}$$

Gas bubbles form in the liquid, and increase in radius as more liquid changes phase:



The remaining unknown factor is $\frac{DR}{Dt}$, giving the increase in bubble radius as the bubble flows.

That factor cannot be determined in a general fashion and requires input from a physical model. The model estimates the bubble radius change rate, which is based on properties of the flow and the materials, within the control volume.

The following models are provided:

- The **Full Rayleigh–Plesset** cavitation model

  See Full Rayleigh–Plesset Cavitation Model.
- The **Schnerr-Sauer** cavitation model

  See Schnerr-Sauer Cavitation Model.
- The **Gas Dissolution** model

  See Bubble Growth Rate Model for Gas Dissolution.

If the phases are multi-component, the rate model breaks down the global mass transfer rate $\dot{m}$ in rates $\dot{m}_i$ for each individual component. These component mass transfer rates appear as source terms in the species transport equations.

See Multi-Component Material.

Full Rayleigh–Plesset Cavitation Model

The Full Rayleigh–Plesset model includes the influence of bubble growth acceleration, as well as viscous and surface tension effects.

The bubble growth velocity, v_r, is determined using the Rayleigh-Plesset equation:

$$R\frac{\mathrm{d}v_r}{\mathrm{d}t} + \frac{3}{2}v_r^2 = \frac{p_{sat} - p}{\rho_l} - \frac{2\sigma}{\rho_l R} - 4\frac{\mu_l}{\rho_l R}v_r \tag{879}$$

where:

- $\frac{\mathrm{d}v_r}{\mathrm{d}t}$ is the substantial derivative of the bubble growth rate
- p_{sat} is the saturation pressure for given temperature
- p is the local pressure in the surrounding liquid
- σ is the surface tension
- ρ_l is the liquid density.

The bubble-growth velocity that is needed for the source term of the vapor volume fraction equation can be computed without neglecting any of the terms in the Rayleigh-Plesset equation.

For multi-component cavitation, the term p_{sat} in the formulation is replaced with $p_{sat,m}$. See Multi-Component Material.

Multi-Component Material

The Rayleigh–Plesset equation can be used for a mixture of multiple components.

In the context of mixture saturation pressure, the term p_{sat} in the formulation is replaced with $p_{sat,m}$, where:

$$p_{sat,m} = \sum_i^{N_c} X_{i,l,s} p^*_{sat,i} \tag{880}$$

This expression is Raoult's Law, with $p^*_{sat,i}$ indicating the saturation pressure of the pure component. The interfacial molar fraction of the liquid components is approximately their value in the bulk, so $X_{i,l,s} \approx X_{i,l,\infty}$.

This formulation neglects the effect of other gases that are present in the bubbles (for example, as a consequence of gas dissociation) and which have their own partial pressure in the bubble. Homogeneous models compute only the pressure of the entire phase mixture, rather than computing a pressure for each individual phase. This single pressure puts restrictions on including the effect of cavitation-inert gases on the cavitation bubble growth rate. The implicit assumption is that cavitation, once it occurs, is so fast that it overwhelms other effects. Important deviations can be expected near the end of the collapse process, as the presence of 'slow gases' dampens the collapse [523]. However, those deviations are probably no larger than the already omitted effects of thermal effects, surface tension, and viscosity. This limitation seems reasonable for most practical applications.

The overall cavitation rate is broken down into rates for the individual components. Unresolved interfacial properties also affect this overall rate. A reasonable assumption is to use the following:

$$\dot{m}_{cav,i} = \dot{m}_{cav}\left(\left(X_{i,l,\infty} p^*_{sat,i}\right) / \left(\sum_i^{N_c} X_{i,l,\infty} p^*_{sat,i}\right)\right) \tag{881}$$

This expression represents a weighting that is based on the individual fugacity of each component, in the context of ideal gases and ideal liquids.

Schnerr-Sauer Cavitation Model

The Schnerr-Sauer cavitation model is based on a reduced Rayleigh–Plesset (RP) equation and neglects the influence of bubble growth acceleration, viscous effects, and surface tension effects.

The Schnerr-Sauer model can be inadequate in modeling cavitation in flows where the cavitation bubbles experience a rapid change of pressure, and consequently have a rapid change of bubble growth velocity. These situations are not unusual, particularly in fast flows where the bubbles move quickly through regions of low pressure and high pressure.

The Schnerr-Sauer cavitation model lets you scale the bubble growth rate and collapse rate for both single-component materials and multi-component materials. You specify the appropriate scaling factors for positive and negative bubble growth rates in the Cavitation model settings.

For more information, see Scaling Bubble Growth Rates.

The single-component cavitation bubble growth rate is estimated using the inertia controlled growth model [545]:

$$\left(\frac{dR}{dt}\right)^2 = \frac{2}{3}\left(\frac{p_{sat} - p_\infty}{\rho_l}\right) \tag{882}$$

where p_{sat} is the saturation pressure corresponding to the temperature at the bubble surface, p_∞ is the pressure of the surrounding liquid and ρ_l is the liquid density. Eqn. (882) is a simplification of the more general Rayleigh-Plesset equation which accounts for the inertia, viscous effects, and surface tension effects.

Research results suggest that for most practical applications, it is not necessary to account for the viscous and surface tension effects [523].

For multi-component cavitation, the term p_{sat} in the formulation is replaced with $p_{sat,m}$. See Multi-Component Material.

Bubble Growth Rate Model for Gas Dissolution

For single-component bubbles, the approximated Epstein-Plesset [530], [538], [541] formulation is assumed to hold:

$$\frac{dR}{dt} = \frac{D_{i,l}\left(c_{i,l,\infty} - c_{i,l,s}\right)}{\rho_g R} \tag{883}$$

Due to the small fraction of $c_{i,l}$ present, this formulation can be written as:

$$\frac{dR}{dt} = \frac{D_{i,l}\rho_l\left(Y_{i,l,\infty} - Y_{i,l,s}\right)}{\rho_g R} \tag{884}$$

To determine the saturation mass fraction, Henry's law is used, in the following form:

$$X_{i,g,s} = \frac{H_i}{p^{abs}} X_{i,l,s} \tag{885}$$

or

$$X_{i,l,s} = \frac{p^{abs}}{H_i} X_{i,g,s} \tag{886}$$

where H_i is the Henry's law constant.

This expression is similar to Raoult's law, but is valid in the limit of $X_{i,l,s} \to 0$, whereas Raoult's law holds in the opposite limit.

In single-component gas dissolution, $X_{i,g,s} = 1$.

To convert molar fraction to mass fraction, use the assumption:

$$\sum_j^N X_{j,l} W_j \approx \sum_j^{N_{sv}} X_{j,l} W_j \tag{887}$$

with:

$$\sum_j^{N_{sv}} X_j \approx 1 \tag{888}$$

where N_{sv} is the number of solvent components. The dissolved species barely contribute to the molar mass of the liquid mixture, due to their low concentration. Therefore, a good approximation is:

$$Y_{i,l,s} \approx \left(X_{i,l,s} W_i\right) / \left(\sum_j^N X_{j,l,\infty} W_j\right) \tag{889}$$

Eqn. (884) determines the bubble radius change rate. In the single-component case, the partial pressure of this component is equal to the total gas pressure. The extension to multi-component is readily achieved by assuming ideal gas behavior. Each component contributes to the pressure according to its molar fraction in the free gas, and the species distribution within the bubble maintains a constant profile. In that case, $X_{i,g,s}$ becomes a transported quantity that is influenced by the gas dissolution rates.

The bubble radius rate change is computed as:

$$\frac{dR}{dt} = \sum_i^{N_{sl}} \frac{D_{i,l} \rho_l \left(Y_{i,l,\infty} - Y_{i,l,s}\right)}{\rho_g R} = \sum_i^{N_{sl}} \left(\frac{dR}{dt}\right)_i \tag{890}$$

Modeling Change of Enthalpy

Due to the lack of interface resolution in the cavitation model, the effects of latent heat on the mass transfer rate are only partially modeled.

The enthalpy is described using a Heat of Formation for all components, as follows:

$$\Delta H_i = \Delta H_{for,i} + \int_{T_{ref}}^{T} C_{p,i} \, \mathrm{d}\, T \tag{891}$$

From this equation, you can compute an enthalpy of the phase, and then compute the enthalpy of the entire mixture. The latent heat of phase change for a specific component is then incorporated into the difference in component enthalpies at that particular temperature.

Rapid evaporation (for example, as a consequence of cavitation or gas dissociation) cools the interface between the phases. Homogeneous models do not resolve the interface, so no energy balance at the interface is incorporated: the energy balance is considered only on the level of the control volume. Therefore, conditions under which the thermal effects become important are unlikely to be simulated accurately.

Note: The cooling (or heating) effect is accurately included for conditions that are not thermally limited: the entire phase mixture does cool down as a consequence of net evaporation. The inaccuracies lie only in the calculation of the rate as a function of temperature.

Formulation Notation

N_{sv}	number of solvent components
N_{sl}	number of solute components
N_d	number of dissolving components
N_c	number of cavitating components
D	diffusion coefficient (m^2/s)
ρ	density (kg/m^3)
c	concentration (kg/m^3)
Y	mass fraction
X	molar fraction
W	molar weight (kg/mol)
α	volume fraction
n_0	seed density (m^{-3})
D_0	seed diameter (m)
R	bubble radius (m)
g	gas
l	liquid
s	surface
i	component index
L	interaction length scale (m)
∞	bulk
p_{sat}	saturation pressure (Pa)
*	pure component

$\dot{m}$	mass transfer rate per unit volume (kg/m^3 -s)
C_p	heat capacity at constant pressure (J/K)

Contents:

Homogeneous Relaxation Model Formulation

Homogeneous Relaxation Model Formulation

The Homogeneous Relaxation model (HRM) is based on a finite rate equation for the rate of change of the vapor mass fraction.

For more information, see [532] and [540].

The rate of change of vapor mass fraction is:

$$\frac{Dx}{Dt} = \frac{\overline{x} - x}{\theta} \tag{892}$$

Where x is the non-equilibrium mass fraction, $\overline{x}$ is the equilibrium mass fraction, and θ is the relaxation time scale.

θ is calculated from an empirical correlation. The most widely used one of the two empirical correlations is:

$$\theta = \theta_0 \alpha^{-0.54} \psi^{-1.76} \tag{893}$$

where θ_0 is a modelling constant with a value equal to 3.84×10^{-7}, α is the vapor void fraction, and ψ is a non-dimensional pressure.

ψ is defined as:

$$\psi = \frac{P_{sat} - P}{P_{crit} - P_{sat}} \tag{894}$$

Homogeneous Relaxation Model Implementation

The Homogeneous Relaxation Model is included in the source term of the vapor volume fraction (α_{vap}) equation as follows:

$$\frac{d\alpha_{vap}}{dt} + \nabla \cdot \left(\alpha_{vap} V\right) = C\frac{Dx}{Dt} + unsteady\ source \tag{895}$$

Similarly, the liquid volume fraction (α_{liq}) equation becomes:

$$\frac{d\alpha_{liq}}{dt} + \nabla \cdot \left(\alpha_{liq} V\right) = -C\frac{\rho_{vap}}{\rho_{liq}}\frac{Dx}{Dt} + unsteady\ source \tag{896}$$

The corresponding continuity equation source term becomes:

$$C\rho_{vap}\frac{Dx}{Dt}\left[\frac{1}{\rho_{vap}}-\frac{1}{\rho_{liq}}\right] \tag{897}$$

The Homogeneous Relaxation model gives the rate of change of the vapor mass fraction. The term C is used to convert the vapor mass fraction into the vapor void fraction, which is the variable that STAR-CCM+ solves for. This correction term is calculated as:

$$C=\frac{\rho_{liq}\rho_{vap}}{\left[\rho_{vap}\left(1-x_{vap}\right)+\rho_{liq}x_{vap}\right]^2} \tag{898}$$

where:

- x is the mass fraction
- ρ is the density
- the subscripts vap and liq represent vapor and liquid respectively.

Melting-Solidification Formulation

The Melting-Solidification model that is implemented in STAR-CCM+ does not track the liquid-solid interface explicitly. Instead, the model uses an enthalpy formulation to determine the distribution of the solidified portion of the liquid-solid phase.

For basic melting-solidification, the enthalpy of the liquid-solid phase ${h_{\mathrm{ls}}}^*$ includes the latent heat of fusion h_{fusion}:

$${h_{\mathrm{ls}}}^*=h_{\mathrm{ls}}+\left(1-{\alpha^*}_s\right)h_{fusion} \tag{899}$$

where h_{ls} is the sensible enthalpy.

The relative solid volume fraction ${\alpha^*}_s$ is defined as the portion of the volume of the liquid-solid phase that is in the solid state. In the enthalpy model, the relative solid volume fraction ${\alpha^*}_s$ is a function of temperature:

$${\alpha^*}_s=\begin{cases}1 & if \quad T^*<0\\ f(T^*) & if \quad 0<T^*<1\\ 0 & if \quad 1<T^*\end{cases} \tag{900}$$

where T^* is the normalized temperature that is defined as:

$$T^*=\frac{T-T_{\mathrm{solidus}}}{T_{\mathrm{liquidus}}-T_{\mathrm{solidus}}} \tag{901}$$

The function $f(T^*)$ is called the fraction solid curve. For a linear dependence between ${\alpha^*}_s$ and T^*, the solidification path is defined as:

$$f\left(T^*\right)=1-T^* \tag{902}$$

The solidification path can also be specified as a table, in which case linear dependencies are assumed between the points in the table. The values of temperature in the table have to increase monotonically as the values of the relative solid fraction decrease monotonically.

If melting and solidification take place at one temperature (that is, $T_{\text{solidus}} = T_{\text{liquidus}}$), a linear solidification path is assumed and a small temperature interval of 0.002 K is introduced automatically. The physical properties of the liquid-solid phase can be constant, or can be a function of temperature and the relative solid volume fraction.

Methods exist for preventing fluid movement in solidified regions: the Slurry Viscosity model, the Mushy Zone Permeability model, and the flow stop functionality.

Flow Stop Mass Compensation

When Flow Stop Mass Compensation is activated, the density of each of the phases in the stopped cells is kept constant at the value that it had when the cell was stopped. This density value is independent of what the equation of state returns for each particular phase. This constant density is required to keep the mass in the stopped cells constant if the cell volume is kept constant and the dependency of density on stresses is omitted. The density in unstopped cells is not modified. Also, the separate material properties *Density Temperature Derivative* and *Density Pressure Derivative* are not adjusted accordingly (these values are zero for stopped cells). This approach covers shrinkage as well as expansion.

The Slurry Viscosity Model

In the early stages of solidification of an alloy, small crystals nucleate and grow in the melt. These crystals move with the liquid but do not interact with each other. The mixture of the solid and liquid phases is called a slurry. However, the presence of the solid crystals in the liquid increases the viscosity of the two-phase mixture, which increases the flow resistance. This increased resistance is simulated using the Slurry Viscosity model.

The Slurry Viscosity model is used for low values of the solid volume fraction, $\alpha^*{}_s \leq \alpha_{cr}$, where the critical solid fraction α_{cr} defines the upper limit of applicability.

The method of Metzner ([536]) expresses the variation in viscosity of the mixture as a function of the solid volume fraction $\alpha^*{}_s$:

$$\mu^* = \mu_l \left[1 - \left(\frac{\alpha^* \cdot F_\mu(\alpha^*{}_s)}{A} \right) \right]^{-2} \tag{903}$$

where:

- μ^* is the effective viscosity of the mixture
- μ_l is the dynamic viscosity of the liquid
- F_μ is a switching function that is given by Eqn. (909)

  See Switching Function for Metzner Slurry Viscosity and Carman-Kozeny Mushy Zone Permeability Model.
- A is a crystal constant which depends on the aspect ratio and surface roughness of the crystal.

The Mushy Zone Permeability Model

For high solid volume fractions, equiaxed grains grow and start to agglomerate and to form dendritic regions. Dendrites also grow from cooled solid surfaces or the solidification front into the melt. A partially solidified stationary region permeated with dendrites is called a “mushy zone”.

A common engineering application for this model is high-pressure die casting where melt is pressed through ducts with wall temperatures below solidus temperature at high velocities. The melt solidifies at the walls and dendrites grow from the solidification front into the melt. This partially solidified region causes an additional pressure loss in the duct that influences the melt mass flow. This pressure loss is simulated using the Mushy Zone Permeability model.

The *Carman-Kozeny Mushy Zone Permeability* method provides a mushy zone for the flow resistance, eliminating the need to set up porous media. In addition, the method updates the velocity field in the melting and solidification process. The flow resistance is active only in cells that have a solid volume fraction greater than zero and is applied to all regions in the physics continuum.

The flow resistance in a mushy zone can be modeled similar to an isotropic porous medium. Therefore, to quantify the permeability, the Carman-Kozeny equation [525] for flow through a porous medium is used:

$$K = \frac{\left(1-\alpha_s^*\right)^3}{\alpha_s^{*2} F_K c_s} \tag{904}$$

where:

- $K\left[m^2\right]$ is the permeability
- $\alpha^*{}_s$ is the non-dimensional solid volume fraction
- F_K is a non-dimensional switching function that is given by Eqn. (910).

  See Switching Function for Metzner Slurry Viscosity and Carman-Kozeny Mushy Zone Permeability Model.
- c_s is the shape factor:

$$c_s = c/d^2\left[m^{-2}\right] \tag{905}$$

where c is a shape constant, with a default value of 180, and $d[m]$ is the secondary dendrite arm spacing.

The momentum equation in continuous integral form is given by Eqn. (2231).

The contribution of the mushy zone permeability to the momentum equation goes into $\mathbf{f}_p$:

$$\mathbf{f}_p = -\mathbf{P}\cdot\mathbf{v} \tag{906}$$

where $\mathbf{P}$ is the porous resistance tensor, which is given by:

$$\mathbf{P} = \mathbf{P}_\mathrm{v} + \mathbf{P}_\mathrm{i}\left\|\mathbf{v}\right\| \tag{907}$$

$\mathbf{P}_\mathrm{v}$ and $\mathbf{P}_\mathrm{i}$ are the viscous (linear) and the inertial (quadratic) resistance tensors, respectively.

The mushy zone permeability contributes to $\mathbf{P}_\mathrm{v}$ through:

$$\mathbf{P}_{\mathrm{v,\ Mushy\ Zone}} = \left(\frac{\mu}{K} + \frac{C_F \rho_l \alpha^*{}_l}{\sqrt{K}}\left\|\mathbf{v}\right\|\right) \tag{908}$$

where:

- $\frac{\mu}{K}$ is Darcy's term

- $\frac{C_F \rho_l \alpha^*{}_l}{\sqrt{K}} \left\| \mathbf{v} \right\|$ is Forchheimer's term

$\mathbf{P}_\mathrm{v}$ can be considered a drag coefficient. The drag coefficient is given by the Darcy's and Forchheimer's terms, which are often referred to as the "viscous-drag" and "form-drag" terms.

- μ is the dynamic viscosity
- C_F is the Ergun coefficient
- ρ_l is the liquid phase density
- $\alpha^*{}_l$ is the liquid fraction that is given by $\alpha^*{}_l = \left(1 - \alpha^*{}_s\right)$.

The Forchheimer's term is optional and its contribution to the mushy zone permeability can be deactivated. The Forchheimer's term must be considered when relatively high interdendritic fluid-flow velocities are expected. Such high velocities can be important during the intensification phase in die or squeeze casting.

Switching Function for Metzner Slurry Viscosity and Carman-Kozeny Mushy Zone Permeability Model

The Metzner slurry viscosity model and the Carman-Kozeny mushy zone permeability model are typically used in combination. Both models cover different ranges of solidification state. The applicability of each model depends on the extent to which the liquid has solidified.

The state of solidification is expressed using the solid volume fraction $\alpha^*{}_s$. The critical solid fraction α_{cr} separates the applicability ranges of the two models.

$\alpha^*{}_s \leq \alpha_{cr}$	Metzner Slurry Viscosity Model
$\alpha_{cr} < \alpha^*{}_s$	Carman-Kozeny Mushy Zone Permeability Model

For the Metzner slurry viscosity model and the Carman-Kozeny mushy zone model, a switching function ensures that each model is applied within its appropriate range of solid volume fraction. The switching function is applied as a multiplicative factor to the respective model equation (see Eqn. (903) and Eqn. (904)).

The switching function for the slurry viscosity model is:

$$F_\mu\left(\alpha^*{}_s\right) = 0.5 - \frac{\arctan\left[s\left(\alpha^*{}_s - \alpha^*{}_{cr}\right)\right]}{\pi} \tag{909}$$

where s is a switching function constant.

The switching function for the mushy zone permeability model is the inverse of Eqn. (909):

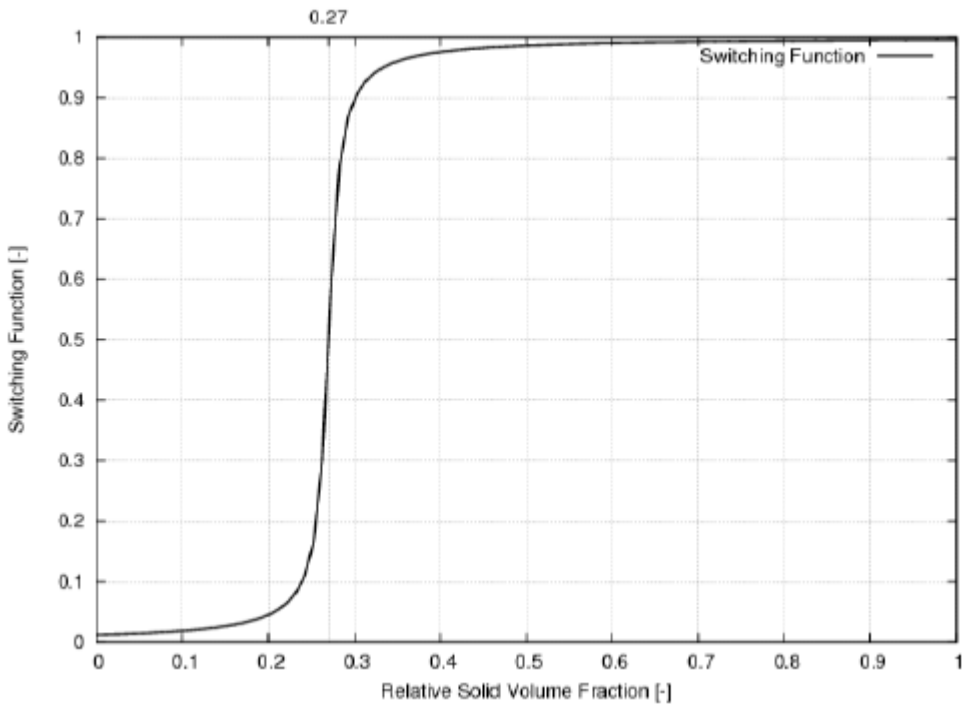
$$F_K\left(\alpha^*{}_s\right) = 1 - F_\mu\left(\alpha^*{}_s\right) = 0.5 + \frac{\arctan\left[s\left(\alpha^*{}_s - \alpha^*{}_{cr}\right)\right]}{\pi} \tag{910}$$

The switching function has a smooth transition near the non-dimensional critical solid volume fraction $\alpha^*{}_{cr}$(see the figure below) where solid grains start to agglomerate to a significant extent:

The default value of the non-dimensional constant s is 100 because the shape is rather insensitive for values from 80 through 120. The default value for the critical solid volume fraction $\alpha^*{}_{cr}$ is 0.27 according to Baeckerud and others [521] for Aluminum alloy A201.

Oldenburger and Spera [542] initially suggested this switching function. The switching function that is implemented in STAR-CCM+ is a slightly modified formulation that is based on the model of Chang and Stefanescu [527].

The following figure depicts the switching function F_K for critical solid volume fraction $\alpha_{cr}^* = 0.27$ (default value).



Modeling VOF Waves

The VOF Waves model is used to simulate surface gravity waves on the interface between a light fluid and a heavy fluid. This model is typically used with the 6-DOF Motion model for marine applications.

The concept of a steadily-progressing periodic wave train is a convenient model that is used in coastal and ocean engineering applications to give fluid velocities, pressures, and surface elevations caused by waves. Assuming that the waves are propagating steadily without change (the steady wave problem), the wave train can be uniquely specified and solved in terms of three physical length scales: water depth, wave length, and wave height.

The main theories that have been used for the steady wave problem are:

- Stokes theory

  An explicit theory that is based on an assumption that the waves are not very steep. This theory is best suited to modeling waves in deeper water.
- Cnoidal theory

  An explicit theory for modeling waves in shallower water, where the wavelength is long compared to the water depth.

STAR-CCM+ provides the following VOF wave models:

- Flat Wave

  A flat wave represents a calm plane of water.
- First Order Wave

  A first order wave is modeled with a first order approximation to the Stokes theory of waves. This approximation generates waves that have a regular periodic sinusoidal profile.
- Fifth Order Wave

  A fifth order wave is modeled with a fifth order approximation to the Stokes theory of waves. This wave more closely resembles a real wave than one generated by the first order method. The wave profile and the wave phase velocity depend on the water depth, wave height, and current.
- Superposition Wave

  A superposition wave is a linear superposition of different *partial* first order waves. It can be used to simulate more complex wave phenomena, such as a cross sea or spectral waves. A cross sea is a sea state with two wave systems traveling at oblique angles. This state can occur when water waves from one weather system continue despite a shift in wind.
- Cnoidal Wave

  A cnoidal wave is used for shallow water applications, where the wavelength is long compared to the water depth (that is, the wavelength is greater than ten times the water depth). The Stokes theory that models first order and fifth order waves is not suitable for these applications.
- Irregular Wave

  An irregular wave can be used to describe a short-term sea state by a wave spectrum, that is, the power spectral density function of the vertical sea surface displacement.

See VOF Wave Type Reference.

VOF waves can be forced to a specific solution at a boundary. Typically, the solution of the 3D Navier-Stokes equations is forced towards a solution that is based on a simplified theory. The forcing source terms adapt the solution to the simplified solution that is imposed at the reduced domain boundary. A VOF wave can also be damped in the vicinity of selected boundaries to reduce wave oscillation near those boundaries.

See Wave Forcing and Wave Damping.

Contents:

VOF Waves Workflow
VOF Waves Model Reference
VOF Wave Type Reference
VOF Wave Damping and Forcing
VOF Waves Guidelines
VOF Waves Formulation

VOF Waves Workflow

A VOF Waves simulation is a special case of a VOF multiphase simulation.

The procedure for setting up a simulation using the VOF Waves model is:

1. Set up the VOF multiphase simulation and select the **VOF Waves** model.

   You can model VOF waves in a single physics continuum only.

   See VOF Wave Model Reference.
2. Create the light fluid and heavy fluid phases.
3. In the **VOF Waves** > **Waves** node, use the right-click menu to create the waves of the appropriate type.

   See VOF Wave Type Reference.

   When created, VOF waves appear as their own node under the **VOF Waves** > **Waves** node.

   Set up the waves as appropriate for the wave type.

   For first order waves and fifth order waves, you specify the following:

   - Wave Height
   - Water Depth
   - Specification Type (Wave Length or Wave Period) and the corresponding value.

   For cnoidal waves, you specify the same properties as for first order waves and fifth order waves, and also the following:

   - Order

   For superposition waves, you create the necessary first order partial waves and set the following for each wave:

   - Amplitude
   - Phase
   - Specification Type (Wave Length or Wave Period) and the corresponding value.

For irregular waves, you specify the following:

- Significant Wave Height
- Peak Wave Period
- Spectrum Type (Pierson-Moskowitz or JONSWAP).

  The JONSWAP specification lets you set peak shapes and spectral widths for the random wave.

See Property Lookup for VOF Wave Types.

4. Set the initial conditions for Volume Fraction, Velocity, and Pressure.
5. Set the boundary conditions for the inlet boundary and outlet boundary.
6. Set up wave forcing or wave damping, if appropriate.

   See VOF Wave Forcing and Wave Damping.

VOF Waves Model Reference

STAR-CCM+ provides the VOF Waves model to simulate surface gravity waves on a light fluid-heavy fluid interface. This model is typically used with the 6-DOF Motion model for marine applications.

VOF waves can be modeled in a single physics continuum only.

VOF Waves Model Reference

Theory	See VOF Waves Formulation.	
Provided By	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **VOF Waves**	
Requires	*Space*: **Three Dimensional** *Material*: **Eulerian Multiphase** *Eulerian Multiphase Model*: **Volume of Fluid (VOF)** *Optional Models*: **VOF Waves**	
Properties	See VOF Waves Model Properties.	
Specific Right-Click Actions	See Right-Click Actions.	
Activates	Model Controls (child nodes)	**Waves** The VOF wave manager node. See VOF Wave Type Reference.
	Reference Values	**Minimum Allowable Wall Distance** See Reference Values.
	Region Inputs	See Region Settings.
	Boundary Inputs	See Boundary Settings.

	Solvers	VOF Wave Zone Distance Solver See VOF Wave Zone Distance Solver Settings.

VOF Waves Model Properties

Damping Constant 1

Specifies f_1 in Eqn. (925), used in VOF wave damping.

Damping Constant 2

Specifies f_2 in Eqn. (925), used in VOF wave damping.

Damping Exponent

Specifies n_d in Eqn. (926), used in VOF wave damping.

Enable Theory Comparison

When this option is activated, the VOF Waves field functions are also updated in the inner fluid domain. When this option is deactivated, the VOF Waves field functions are updated only on the boundaries of the fluid domain.

Hence, it is possible to plot the entire scalar field of the VOF Waves field functions. This plot shows the solution of Stokes wave theory of the selected wave with respective order. You can compare this theoretical solution to a plot of the volume fraction that is obtained through the solution of the transport equation.

Forcing Constant

Specifies the forcing coefficient, γ in Eqn. (927), used in VOF wave forcing.

Right-Click Actions

The following right-click actions become available on the **VOF Waves** > **Waves** node.

New > [wave type]

where **[wave type]** is one of **Flat**, **First Order**, **Fifth Order**, **Cnoidal**, **Superposition**, or **Irregular**.

For information on each of the wave types, see Modeling VOF Waves and VOF Wave Type Reference.

Reference Values

Minimum Allowable Wall Distance

The wall distance algorithm has been developed to return a positive definite value for almost all cell shapes. However it is possible that some degenerate cells, such as boundary sliver cells, could return a value that is close to zero. Since turbulence models and wall functions require a non-zero value, this could cause numerical problems. For this reason, the minimum allowable wall distance limits the value of the wall distance in the continuum. The default value is 1.0E-6 m.

If the wall distance is limited in one or more cells, a warning message is displayed. In this case, investigate the cause and attempt to remedy the situation.

Region Settings

Applies to fluid regions.

VOF Wave Zone Option

Specifies the type of VOF wave treatment that is applied to the specified zone in the region.

Method	Corresponding Physics Value Nodes
None No damping or forcing is applied.	None
Damping Damping is activated. A **VOF Wave Damping Boundary Option** node is added to the physics conditions for each boundary in the region.	**VOF Wave Damping Length** The distance from the VOF wave damping-enabled boundary at which the damping starts. The specified value must be at least one wavelength. See Eqn. (926), the starting point for wave damping. According to Eqn. (925), there is a resistance term added to the equation for w-velocity [134]. In order to be consistent with all effects, the effect of the damping is included in the wave profiles that are used for initialization.
Forcing Forcing is activated. A **Forcing Length Specification** node is added to the physics conditions for the region. A **VOF Wave Forcing Boundary Option** node is added to the physics conditions for each boundary in the region.	**Forcing VOF Wave Specification** This setting defines the VOF wave to which the simulation solution is forced at the boundary. This value is ϕ^* in Eqn. (927). • **VOF Wave** Select the appropriate wave from the list of VOF waves that are currently set up in the physics continuum.

Forcing Length Specification

Specifies whether the forcing length is set at the region level or set for individual boundaries. This setting is available when the **VOF Wave Zone Option** is set to **Forcing**.

Method	Corresponding Physics Value Nodes
Region The forcing length is set at the region level.	**VOF Wave Forcing Length** Specifies the distance from the VOF wave forcing-enabled boundary at which the forcing starts. The specified value must be at least one wavelength. The optimal forcing zone length is around two wavelengths. You should make the solution domain longer than two wavelengths since the forcing zone cannot start close to the body of interest.

Boundary The forcing length is set for individual boundaries.	None. A **Forcing Length** node is added to the physics values for each boundary in the region that has the **VOF Wave Forcing Boundary Option** physics condition set to **Yes**.

Boundary Settings

This setting is available when the **VOF Wave Zone Option** in the physics conditions of the region is set to **Damping**. It is available only in velocity inlet, pressure outlet, and symmetry boundaries.

VOF Wave Damping Boundary Option

Specifies whether VOF damping is used for the boundary.

Method	Corresponding Physics Value Nodes
No Damping is off.	None
Yes Damping is activated for the boundary.	None

This setting is available when the **VOF Wave Zone Option** in the physics conditions of the region is set to **Forcing**. It is available only in velocity inlet, pressure outlet, and symmetry boundaries.

VOF Wave Forcing Boundary Option

Specifies whether forcing to the specified VOF wave is used for the boundary.

Method	Corresponding Physics Value Nodes
No No forcing is applied.	None
Yes Forcing is applied in the forcing zone before the boundary.	**Forcing Length** The distance from the boundary at which the forcing starts. The specified value must be at least one wavelength. The optimal forcing zone length is around two wavelengths. You should make the solution domain longer than two wavelengths since the forcing zone cannot start close to the body of interest. This setting is available only when the **Forcing Length Specification** region condition is set to **Boundary**.

VOF Wave Zone Distance Solver Settings

This solver controls the wave zone distance computation for the VOF Waves model. All of the solver properties are expert properties, and you would not normally need to change them.

Solver Frozen

When activated, prevents the solver from updating the solution while iterating or responding to other solvers. This feature is primarily a “debugging” tool and is not supported for normal simulation work. This property is deactivated by default.

Use SFC/SIMD

When activated, a Space-Filling Curve search algorithm is used for the wave zone distance calculation. This property is deactivated by default.

Activating this option usually speeds up the wave zone distance calculation while maintaining a similar memory consumption. For cases with low core count an increase in peak memory can occur.

Verbosity

Allows you to choose how much information STAR-CCM+ reports in the Output window while the simulation is running. This property is useful for debugging problems when they occur. The only valid entries are 0 (default), 1, and 2:

- 0 = None: Leaves output at its default.
- 1 = Low: Displays CPU time statistics across cores for calculating the wave zone distance. It reports the mean, maximum, and minimum elapsed times in seconds.
- 2 = High: Displays CPU time statistics (see low verbosity) and memory usage statistics across cores. This entry allows you to evaluate the memory consumption of all server processes that are relevant to the wave zone distance calculation:
  - Max process peak—that is the maximum resident memory usage across cores.
  - Max host peak—that is the maximum memory that is used on a node.
  - Accumulated peak—that indicates the rough scalability in terms of required memory.

Minimum Tree Size Threshold

Controls the size of the search tree that is used by the wave zone distance solver. By default, the search tree can contain 500,000 data points. Increasing this number can speed up the wave zone distance calculation at the expense of consuming additional memory. For a parallel simulation, the maximum tree size is computed as the higher of the following numbers:

- Number of cells per processor
- Minimum tree size threshold

By forcing the tree size to be at least the value set by the minimum tree size threshold, STAR-CCM+ limits the number of times that the wave zone distance solver rebuilds the search tree.

VOF Wave Type Reference

The VOF Waves model supports the following VOF wave types: Flat, First Order, Fifth Order, Superposition, Cnoidal, and Irregular.

VOF Wave Type Family Reference

Model Names and Abbreviations	**Flat Wave**	**Flat**
	First Order Wave	**First**
	Fifth Order Wave	**Fifth**
	Cnoidal Wave	**Cnoid**
	Superposition Wave	**Supp**
	First Order Superposing Wave	**FirstSupp**
	Irregular Wave	**Irreg**
Theory	See VOF Waves Formulation.	
Provided By	**[physics continuum]** > **Models** > **VOF Waves** > **Waves** (Right-click menu: **New** > **[wave type]**)	
Example Node Path	**Continua** > **Physics 1** > **Models** > **VOF Waves** > **Waves** > **[wave type]**	
Requires	The VOF Waves model activated.	
Properties	See Property Lookup for VOF Wave Types.	
Activates	Model Controls (child nodes)	• Wave Length Specification • Wave Period Specification • SubWaves • JONSWAP Specification See Child Nodes.
	Field Functions	• Hydrostatic Pressure of [VofWave Name] • Relative Zone Distance • Velocity of [VofWave Name] • VofWave Damping Factor • Volume Fraction of Heavy Fluid of [VofWave Name] • Volume Fraction of Light Fluid of [VofWave Name] See Field Functions.

Property Lookup for VOF Wave Types

	Flat	First	Fifth	Supp	FirstSupp	Cnoid	Irreg
Point on Water Level The mean water level when the	✓	✓	✓	✓		✓	✓

	Flat	First	Fifth	Supp	FirstSupp	Cnoid	Irreg
simulation is initialized. If **First Wave Front Setting** is activated, this setting also defines the position of the leading front of the wave. For a flat VOF wave, this property defines the position of the water surface (that is, the water level).							
Vertical Direction The vector normal to the mean water surface.	✓	✓	✓	✓		✓	✓
Current Velocity of the water or other heavy fluid.	✓	✓	✓	✓		✓	✓
Wind Velocity of the air or other light fluid.	✓	✓	✓	✓		✓	✓

	Flat	First	Fifth	Supp	FirstSupp	Cnoid	Irreg
Light Fluid Density This property is required for the hydrostatic field function that is created automatically with the wave. The default value is for air. If you are simulating any other substance, change this value manually.	✓	✓	✓	✓		✓	
Heavy Fluid Density This property is required for the hydrostatic field function that is created automatically with the wave. The default value is for water. If you are simulating any other substance, change this value manually.	✓	✓	✓	✓		✓	

	Flat	First	Fifth	Supp	FirstSupp	Cnoid	Irreg
Advancing Direction Direction traveled by the wave.		✓	✓		✓	✓	✓
Amplitude Height of the wave crest.		✓			✓		
Specification Type The quantity to use in defining the length or duration of the wave. • **Wave Length** The wave is defined by its wavelength. A Wave Length Specification node is added as a child node. • **Wave Period** The wave is defined by its time period.		✓	✓		✓	✓	

	Flat	First	Fifth	Supp	FirstSupp	Cnoid	Irreg
A Wave Period Specification node is added as a child node.							
Deep Water Approximation When this option is activated (default), the dispersion relation for waves is independent of the water (or other heavy fluid) depth (see Eqn. (917)). When this option is deactivated, Eqn. (916) applies.		✓		✓			
First Wave Front Setting When this option is activated (default), the **Point on Water Level** setting defines both the mean water level and the start		✓	✓	✓		✓	

	Flat	First	Fifth	Supp	FirstSupp	Cnoid	Irreg
of the wave. There is only flat water in the **Advancing Direction** from the specified point. When this option is deactivated, the **Point on Water Level** setting defines only the mean water level. Waves are present in the entire domain at initialization.							
Additional Velocity This velocity is added to the combined velocity value that the current, wind, and wave predict. When solving for the flow around a moving vessel in the reference frame of the			✓			✓	

	Flat	First	Fifth	Supp	FirstSupp	Cnoid	Irreg
vessel, set this value to the inverse of the velocity of the vessel.							
Wave Height Vertical distance from crest to trough.			✓			✓	
Water Depth Depth of the body of water being simulated.			✓			✓	
Phase The initial wave phase at t = 0s for the particular partial wave. This property is used to describe the phase difference between the different partial waves. The wave phase is the fraction of a wave cycle that is measured in radians.					✓		

	Flat	First	Fifth	Supp	FirstSupp	Cnoid	Irreg
Enabled When activated, the effects of this wave are included in the simulation.					✓		
Order Specifies the order of the cnoidal wave solution. You can choose to include up to five terms in the calculation, by selecting the correspondin g value (First order – Fifth order). Terms higher than the specified order of approximatio n are neglected. Alternatively, you can use Aitken transformatio ns to improve convergence for the full series of terms in the calculation.						✓	

	Flat	First	Fifth	Supp	FirstSupp	Cnoid	Irreg
Significant Wave Height Defined as the average height (trough to crest) of the highest one-third waves.							✓
Peak Wave Period The wave period that is determined by the inverse of the frequency at which a wave energy spectrum has its maximum value.							✓
Spectrum Type The quantity to use in defining the length or duration of the wave: • **Pierson-Moskowitz** The Pierson-Moskowitz spectrum was							✓

	Flat	First	Fifth	Supp	FirstSupp	Cnoid	Irreg
proposed originally for a fully developed sea. • **JONSWAP** The JONSWAP spectrum extends the Pierson-Moskowitz spectrum to include fetch limited seas, describing developing sea states. A JONSWAP Specification child node is added.							
Number of Frequencies Number of partial waves that are used							✓

	Flat	First	Fifth	Supp	FirstSupp	Cnoid	Irreg
to create the irregular wave. The wave amplitudes of the partial wave are computed according to the wave spectrum.							
Test Option When this property is activated, the random number generator uses a fixed seed. This allows you to perform repeated runs with the same wave pattern.							✓

Child Nodes

The individual wave nodes can have the following child nodes:

Wave Length Specification

Specifies the wavelength. The corresponding wave period is displayed when the simulation has been initialized.

The Wave Length Specification properties are:

- **WaveLength**
  The wavelength λ in Eqn. (915).
- **WavePeriod**
  The wave period T in Eqn. (914) (read only).

This node appears when the *Specification Type* property of the wave is set to **Wave Length**. Applies to First Order, Fifth Order, and Cnoidal wave types only.

Wave Period Specification

Specifies the wave period. The corresponding wavelength is displayed when the simulation has been initialized.

The Wave Period Specification properties are:

- **WavePeriod**
  The wave period T in Eqn. (914).
- **WaveLength**
  The wavelength λ in Eqn. (915) (read only).

This node appears when the *Specification Type* property of the wave is set to **Wave Period**. Applies to First Order, Fifth Order, and Cnoidal wave types only.

SubWaves

This node is the manager node for the first order superposition waves that comprise a superposition wave.

The **New** option on the right-click menu lets you create the necessary first order superposition waves.

JONSWAP Specification

The JONSWAP specification lets you set peak shapes and spectral widths for the random wave. This node appears under an irregular wave node when the *Spectrum Type* property of the irregular wave is set to **JONSWAP**.

See JONSWAP Specification Properties.

Applies to Irregular wave types only.

JONSWAP Specification Properties

The JONSWAP spectrum extends the Pierson-Moskowitz spectrum to include fetch limited seas, describing developing sea states. This spectrum is expected to be a reasonable model for the following (in SI units):

$$3.6 < \frac{\text{Peak Wave Period}}{\sqrt{\text{Significant Wave Height}}} < 5$$

Peak Shape Parameter

Sets the peak shape, γ in Eqn. (924). Average value is γ = 3.3.

Spectral Width A

Sets σ_a (see Eqn. (924)). Average value is 0.07.

Spectral Width B

Sets σ_b (see Eqn. (924)). Average value is 0.09.

Field Functions

These field functions are added when any VOF wave is created. These field functions define initial conditions and boundary conditions for objects that are floating in the heavy fluid (typically water).

Hydrostatic Pressure of [VofWave Name]

Hydrostatic pressure of a particular VOF wave. This setting can be used to initialize the pressure field and on pressure boundaries.

Relative Zone Distance

The value of x^* from Eqn. (928).

Velocity of [VofWave Name]

The velocity vector field of a particular VOF wave. This setting can be used for initial conditions and for the velocity specification on boundaries.

VofWave Damping Factor

The damping factor S_z^{d} calculated in Eqn. (925).

Volume Fraction of Heavy Fluid of [VofWave Name]

Volume fraction of heavy fluid (such as water) of a particular VOF wave. This setting can be used for the initial volume fraction and for the volume fraction specification at boundaries.

Volume Fraction of Light Fluid of [VofWave Name]

Volume fraction of light fluid (such as air) of a particular VOF wave. This setting can be used for the initial volume fraction and for the volume fraction specification at boundaries.

VOF Wave Forcing and Wave Damping

The VOF Waves model includes both wave forcing and wave damping capabilities.

In many VOF Wave applications, you need to simulate transient flows around bodies over a longer period. Although the flow usually takes place in a large domain, you are normally interested only in the solution in the immediate vicinity of the body. STAR-CCM+ can reduce the size of the computational domain and thus reduce the computing effort while not compromising the accuracy and reliability of the solution. To achieve this, the solution of the 3D Navier-Stokes equations is forced towards a solution that is based on a simplified theory (or a 2D solution in a larger domain) over some distance around the body. Momentum sources only are used—no Phase sources or Turbulence sources are applied. The forcing source terms adapt the solution to the simplified solution that is imposed at the reduced domain boundary. This forcing resolves the problem of specifying boundary conditions on the reduced solution domain boundaries.

Forcing can be used to damp the upstream-propagating waves before they reach the inlet boundary, allowing long-lasting simulations to be performed on a reduced domain. Disturbances by reflections from boundaries must be eliminated, particularly when the incoming waves are irregular and possible extreme events need to be captured reliably.

See Wave Forcing.

A VOF wave can also be damped in the vicinity of selected boundaries to reduce wave oscillation near those boundaries. The damping introduces vertical resistance to the vertical motion. For example, waves that are generated by floating bodies can be damped to avoid their reflection at boundaries ([134]). However, this damping approach cannot be applied at the inlet since the incoming waves would be damped as well.

See Wave Damping.

Setting Up VOF Wave Forcing

To set up VOF wave forcing in a particular region or for specific boundaries in your simulation:

1. In the Physics Conditions of the region, set the **VOF Wave Zone Option** to **Forcing**.

   See VOF Wave Zone Option.
2. In the Physics Values of the region, set the **Forcing VOF Wave Specification**.

   See Forcing VOF Wave Specification.
3. In the Physics Conditions of the region, set the **Forcing Length Specification** to **Region** or **Boundary**, whichever is appropriate.

   See Forcing Length Specification.
4. If the forcing length is applied to the entire region, in the Physics Values of the region, set the **VOF Wave Forcing Length**.

   See VOF Wave Forcing Length.
5. In the Physics Conditions of each boundary, set the **VOF Wave Forcing Boundary Option** to specify whether to apply forcing at that particular boundary.

   See VOF Wave Forcing Boundary Option.
6. If the forcing length is applied to individual boundaries, in the Physics Values of each boundary in which wave forcing is applied, set the **Forcing Length**.

   See Forcing Length.

Setting Up VOF Wave Damping

To set up VOF wave damping in a particular region of your simulation:

1. In the Physics Conditions of the region, set the **VOF Wave Zone Option** to **Damping**.

   See VOF Wave Zone Option.
2. In the Physics Values of the region, set the **VOF Wave Damping Length**.

   See VOF Wave Damping Length.
3. In the Physics Conditions of each boundary, set the **VOF Wave Damping Boundary Option** to specify whether to apply damping in the zone before that particular boundary.

   See VOF Wave Damping Boundary Option.

VOF Waves Guidelines

This section describes the set-up and analysis procedures applicable to Volume Of Fluid (VOF) wave simulations.

See VOF Guidelines for general recommendations on VOF simulations.

Meshing Considerations

Use the following meshing guidelines, where applicable:

- Mesh near free surfaces must be able to resolve both amplitude (wave height) and wavelength. This objective requires a cell size appropriate for the wave under consideration.
- You can increase grid size with distance from the free surface, since wave effects decay exponentially with depth.
- As ships and similar bodies generate a wide spectrum of waves, it is not feasible to resolve the smallest wavelengths. Choose the minimum wavelength that you want to resolve.
- If you are using irregular or superposed waves, you must expect to use a fine mesh to resolve small waves.
- If you are using flat wave, set the mesh to capture the waves that are generated in the wake. Also, adjust the mesh such that the flat free surface coincides with grid surface (that is, avoid partially filled cells at the undisturbed free surface).
- Use the wave model appropriate for the given wave steepness and water depth. See Modeling VOF Waves for the available types waves.
- In most cases, the best size of time-step depends on speed of wave propagation. However, small time-steps can be required for breaking waves, or for cases where the mesh is fine in the vertical direction. In these cases, vertical motion of the free surface is more important than wave velocity.

Note: Simulations with second-order discretization diverge if the free surface moves half a cell or more per time-step.

- The best time-step size for ships and offshore structures depends on wave properties and whether additional detailed flow features must be resolved. For example, when modeling wave impacts on an offshore platform, the simulation can require smaller time steps so that it captures maximal pressures or forces.
- If you know the amplitude and wavelength or period of the VOF waves, use the following steps:
  1. In the Mesh continuum, select the Trimmer model.
  2. In the **Trimmer** node model properties, activate *Do Mesh Alignment*.
  3. Select **Reference Values** > **Mesh Alignment Location**.
  4. Set the *Location* property to the waterline at the front of the body in the simulation.
  5. Select **Geometry** > **Parts** > **New Shape Part** > **Block** and create a block that covers the entire domain in the X and Y directions. Also extend the block far enough in the Z direction to enclose all the waves, with 10% additional space.
  6. Create a Volumetric Control for the block.
  7. In the Mesh continuum, under **Volumetric Controls** > **Volumetric Control n** > **Mesh Values**, set the following properties, in order:

Node	Property	Value
Mesh Conditions > **Trimmer**	*Customize Anisotropic Size*	Activated
Mesh Values > **Trimmer Anisotropic Size**	*Relative/Absolute*	**Absolute**
	Custom X Size	Activated
	Custom Y Size	Activated
	Custom Z Size	Activated

Node	Property	Value
Absolute X Size	*Value*	1/80th to 1/100th of the wavelength
Absolute Y Size	*Value*	1/80th to 1/100th of the wavelength
Absolute Z Size	*Value*	About 1/20th of the wave height

The X and Y axes lie on the free surface, with the Z axis normal to it.

The mesh resolution along the wave does not have to be as high as in the wave propagation direction (except for short-crested waves).

8. Set the time-step for the Implicit Unsteady Solver according to this equation:

   time-step = $\frac{P}{2.4n}$

   where P is the wave period and n is the number by which you divided the wavelength to get **Absolute X Size**. For instance, if you used 1/80 of the VOF wavelength, n = 80.

The above steps apply best to first- and fifth-order waves.

Boundary Conditions

Use the following guidelines, where applicable:

- Use the inflow boundary condition whenever possible. Make sure current and wind velocities vectors point from the inlet to the outlet.
- Generally, use the no-slip condition on side boundaries.
- For open water or side boundaries, slip-wall or symmetry conditions are appropriate for side boundaries.
- For bottom boundaries, inlet, no-slip, or slip wall boundaries are appropriate, but never symmetry boundaries.
- Eliminate spurious wave reflections. These reflections most commonly happen at boundaries and sudden mesh transitions.
  - To avoid internal reflections, avoid changes in grid resolution in the vertical direction within the wave zone.
  - To avoid wave reflections at boundaries, use a large damping zone, initially, then reduce it to about half the volume after 5–10 time-steps. See VOF Wave Forcing and Wave Damping.
- For each wave, in the Physics continuum, under **Initial Conditions**, set the following values:

Node	Property	Value
Pressure	*Method*	**Field Function**
	Scalar Function	**Hydrostatic Pressure of [wave]**
Velocity	*Method*	**Field Function**
	Vector Function	**Velocity of [wave]**

Node	Property	Value
Volume Fraction	*Method*	**Composite**
└ **Composite** > **H2O** (or other heavy fluid)	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Heavy Fluid of [wave]**
└ **Composite** > **Air** (or other light fluid)	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Light Fluid of [wave]**

- For all inlet boundaries under **Regions** > **[region]** > **Boundaries**, set the following values:

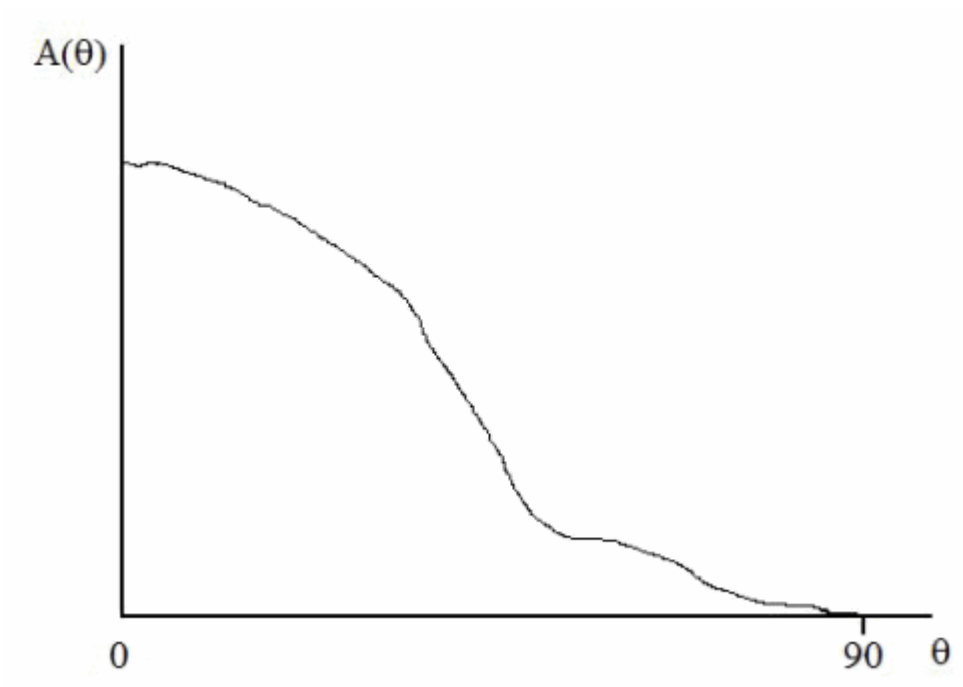
Node	Property	Value
[any velocity inlet] > **Physics Conditions** > **Velocity Specification**	*Method*	**Magnitude + Direction**
[any velocity inlet] > **Physics Values** > **Velocity Magnitude**	*Method*	**Field Function**
	Scalar Function	**Velocity of [wave]**
[any inlet] > **Physics Values** > **Volume Fraction**	*Method*	**Composite**
└ **Composite** > **H2O** (or other heavy fluid)	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Heavy Fluid of [wave]**
└ **Composite** > **Air** (or other light fluid)	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Light Fluid of [wave]**
[any outlet] > **Physics Values** > **Volume Fraction**	*Method*	**Composite** At the outlet, specify a pressure boundary. Set the pressure and volume fraction to the hydrostatic pressure and the volume fraction of the flat wave.
└ **Composite** > **H2O** (or other heavy fluid)	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Heavy Fluid of [wave]**
└ **Composite** > **Air** (or other light fluid)	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Light Fluid of [wave]**
[any pressure outlet] > **Physics Values** > **Presure**	*Method*	**Field Function**
	Scalar Function	**Hydrostatic Pressure of [wave]**

The Physics of Wakes

When setting up simulations using VOF waves, bear in mind the following features of wave physics:

- The energy of a wave is proportional to the square of the wave amplitude. If possible, obtain the amplitude spectrum for the ship. This profile relates the amplitude of waves $A(\theta)$ to the angle θ between the path of the ship and the path of the wave. Use this information to decide which waves must be resolved and which can be ignored.

  Example amplitude spectrum:



- The wavelength of wake waves is also related to θ.

  For a ship in steady forward motion:

  $$\lambda = \frac{2\pi U^2}{g}\cos^2\theta$$

  where U is the speed of the ship.
- Under **Models** > **VOF Waves** > **Waves**, in the properties for the individual wave nodes, set *Light Fluid Density* and *Heavy Fluid Density* to the correct values. Do not assume that the default values are correct for your simulation.
- In simulations that last many wave periods, to get the best wave propagation result, increase the *Angle Factor* property of the Volume of Fluid (VOF) model from the default of `0.05` to `0.15`.

VOF Waves Formulation

STAR-CCM+ provides the VOF Waves model to simulate surface gravity waves on the interface between a light fluid and a heavy fluid.

The VOF Waves model supports the following:

- First Order Waves
- Fifth Order Waves
- Cnoidal Waves
- Irregular Waves
- Superposition Waves

  A superposition wave is a linear superposition of two or more first order waves.
- Wave Damping
- Wave Forcing

First Order VOF Waves

A first order wave is modeled with a first order approximation to the Stokes theory of waves. This approximation generates waves that have a regular periodic sinusoidal profile.

The equation for horizontal velocity is:

$$\upsilon_h = a\omega\cos\left(\mathbf{K}\cdot\mathbf{x}-\omega t\right)e^{Kz} \tag{911}$$

The equation for vertical velocity is:

$$\upsilon_v = a\omega\sin\left(\mathbf{K}\cdot\mathbf{x}-\omega t\right)e^{Kz} \tag{912}$$

The equation for surface elevation is:

$$\eta = a\cos\left(\mathbf{K}\cdot\mathbf{x}-\omega t\right) \tag{913}$$

where:

- a is the wave amplitude
- ω is the wave frequency
- $\mathbf{K}$ is the wave vector
- K is the magnitude of the wave vector
- z is the vertical distance from the mean water level.

The wave period T is defined as:

$$T = \frac{2\pi}{\omega} \tag{914}$$

The wavelength λ is defined as:

$$\lambda = \frac{2\pi}{K} \tag{915}$$

The dispersion relation (between wave period T and wave length λ) for first order waves in finite water depth d is:

$$T = \left[\frac{g}{2\pi\lambda}\tanh\left(\frac{2\pi d}{\lambda}\right)\right]^{-1/2} \tag{916}$$

Whereas for infinite water depth, the dispersion relation is:

$$\lambda = \frac{gT^2}{2\pi} \tag{917}$$

The wave shape is independent of depth.

Fifth Order VOF Waves

A fifth order wave is modeled with a fifth order approximation to the Stokes theory of waves. This wave more closely resembles a real wave than a wave that is generated by the first order method. The wave profile and the wave phase velocity depend on the water depth, wave height, and current.

The fifth order VOF waves are based on work by Fenton [138].

The Ursell number U_R is defined as [135]:

$$U_R = \frac{H\lambda^2}{d^3} \tag{918}$$

where H, is the wave height λ is the wavelength and d is the depth of the water.

This wave theory is valid only for Ursell numbers less than 30.

Cnoidal Waves

A cnoidal wave is a nonlinear and exact periodic wave solution of the Korteweg–de Vries equation, which describes the propagation of waves over a flat bed. Korteweg & de Vries [142] obtained periodic solutions which they termed ”cnoidal” because the surface elevation is proportional to the square of the Jacobian elliptic function $\mathrm{cn}()$. Cnoidal waves are used to describe surface gravity waves that have a long wavelength when compared to the fluid depth. The cnoidal solution displays the long flat troughs and narrow crests that are observed in natural shallow water waves. As the wavelength tends to infinity, the cnoidal solution describes a solitary wave.

The cnoidal solution presents results in terms of expansions in wave height. The parameter that is used is the wave height relative to the trough depth of fluid, H/h, which is denoted by ϵ.

The series is expressed as a power series in ϵ/m rather than ϵ, [137]. As m can be less than 1, it is better to monitor the magnitude of ϵ/m than to have a power series in ϵ with coefficients which are polynomials in $1/m$.

The full solution to third order is presented. This solution is more applicable to shorter and reduced-amplitude waves, where the parameter m is less than 0.96.

The symbol cn is used to denote $\mathrm{cn}(\alpha X/h|m) = \mathrm{cn}(\alpha(x-ct)/h|m)$.

Surface elevation calculation

$$\frac{\eta}{h} = 1 + \left(\frac{\epsilon}{m}\right) m\,\mathrm{cn}^2 + \left(\frac{\epsilon}{m}\right)^2\left(-\frac{3}{4}m^2\mathrm{cn}^2 + \frac{3}{4}m^2\mathrm{cn}^4\right) + \left(\frac{\epsilon}{m}\right)^3\left(\left(-\frac{61}{80}m^2 + \frac{111}{80}m^3\right)\mathrm{cn}^2 + \left(\frac{61}{80}m^2 - \frac{53}{20}m^3\right)\mathrm{cn}^4 + \frac{101}{80}m^3\mathrm{cn}^6\right) \tag{919}$$

Horizontal fluid velocity in the frame of the wave calculation

Variation in the x-direction is assumed to be relatively slow and can be expressed in terms of a scaled dimensionless variable $\alpha x/h$. α is a small quantity which expresses the relative slowness of variation in the x-direction, and h is the trough depth.

The series are expressed in terms of α^2 (specifically $\delta = 4\alpha^2/3$) [139]. The results are accurate even for high waves.

$$\begin{aligned}\frac{U}{\sqrt{gh}} = & -1 + \delta\left(\frac{1}{2} - m + m\mathrm{cn}^2\right) \\ & + \delta^2\left(-\frac{19}{40} + \frac{79}{40}m - \frac{79}{40}m^2 + \mathrm{cn}^2\left(-\frac{3}{2}m + 3m^2\right) - m^2\mathrm{cn}^4\right. \\ & \left. + \left(\frac{Y}{h}\right)^2\left(-\frac{3}{4}m + \frac{3}{4}m^2 + \mathrm{cn}^2\left(\frac{3}{2}m - 3m^2\right) + \frac{9}{4}m^2\mathrm{cn}^4\right)\right) \\ & + \delta^3\left(\frac{55}{112} - \frac{3471}{1120}m + \frac{7113}{1120}m^2 - \frac{2371}{560}m^3 + \mathrm{cn}^2\left(\frac{71}{40}m - \frac{339}{40}m^2 + \frac{339}{40}m^3\right)\right. \\ & + \mathrm{cn}^4\left(\frac{27}{10}m^2 - \frac{27}{5}m^3\right) + \frac{6}{5}m^4\mathrm{cn}^6 + \left(\frac{Y}{h}\right)^2\left(\frac{9}{8}m - \frac{27}{8}m^2 + \frac{9}{4}m^3\right. \\ & \left. + \mathrm{cn}^2\left(-\frac{9}{4}m + \frac{27}{2}m^2 - \frac{27}{2}m^3\right) + \mathrm{cn}^4\left(-\frac{75}{8}m^2 + \frac{75}{4}m^3\right) - \frac{15}{2}m^3\mathrm{cn}^6\right) \\ & + \left(\frac{Y}{h}\right)^4\left(-\frac{3}{16}m + \frac{9}{16}m^2 - \frac{3}{8}m^3 + \mathrm{cn}^2\left(\frac{3}{8}m - \frac{51}{16}m^2 + \frac{51}{16}m^3\right)\right. \\ & \left.\left. + \mathrm{cn}^4\left(\frac{45}{16}m^2 - \frac{45}{8}m^3\right) + \frac{45}{16}m^3\mathrm{cn}^6\right)\right)\end{aligned} \tag{920}$$

The leading term -1 arises because the wave is considered to be traveling in the positive x-direction. The fluid is flowing under the wave in the negative x-direction, relative to the wave, with velocities of the order of the wave speed.

Vertical fluid velocity calculation

The vertical fluid velocity can be obtained from Eqn. (920) by using the mass conservation equation $\partial U/\partial X + \partial V/\partial Y = 0$, and $d(\mathrm{cn}(\theta|m))/d\theta = -\mathrm{sn}(\theta|m)\mathrm{dn}(\theta|m)$. Each term that contains $(Y/h)^i\mathrm{cn}^j(\alpha X/h|m)$, for $j > 0$, is replaced by $\alpha\mathrm{sn}()\mathrm{dn}()\left(\frac{j}{i+1}\right) \times (Y/h)^{i+1}\mathrm{cn}^{j-1}()$. Hence, if Eqn. (920) is written as:

$$\frac{U}{\sqrt{gh}} = -1 + \sum_{i=1}^{5}\delta^i\sum_{j=0}^{i-1}\left(\frac{Y}{h}\right)^{2j}\sum_{k=0}^{i}\mathrm{cn}^{2k}()\Phi_{ijk} \tag{921}$$

where each coefficient Φ_{ijk} is a polynomial of degree i in the parameter m, then the vertical velocity component is:

$$\frac{V}{\sqrt{gh}} = 2\alpha\mathrm{cn}()\mathrm{sn}()\mathrm{dn}()\sum_{i=1}^{5}\delta^i\sum_{j=0}^{i-1}\left(\frac{Y}{h}\right)^{2j+1}\sum_{k=1}^{i}\mathrm{cn}^{2(k-1)}()\frac{k}{2j+1}\Phi_{ijk} \tag{922}$$

Irregular VOF Waves

An irregular wave models wind seas (a short-term sea state) with a wave spectrum, that is, the power spectral density function of the vertical sea surface displacement. An irregular wave can use either the Pierson-Moskowitz spectrum or the JONSWAP spectrum [136]. Both spectra describe wind sea conditions that often occur for the most severe sea states.

The Pierson-Moskowitz spectrum $S_{PM}(\omega)$ is defined as:

$$S_{PM}(\omega) = \frac{5}{16}\left(H_S^2\omega_p^4\right)\omega^{-5}\exp\left(-\frac{5}{4}\left(\frac{\omega}{\omega_p}\right)^{-4}\right) \tag{923}$$

where $\omega_p = (2\pi)/\left(T_p\right)$ represents the angular spectral peak frequency.

The JONSWAP spectrum $S_J(\omega)$ is expressed as:

$$S_J(\omega) = A_\gamma S_{PM}(\omega)\gamma^{\exp\left(-0.5\left(\frac{\omega-\omega_p}{\sigma\omega_p}\right)^2\right)} \tag{924}$$

where:

- $S_{PM}(\omega)$ is the Pierson-Moskowitz spectrum
- γ is the non-dimensional peak shape parameter
- $A_\gamma = 1-0.287\ln(\gamma)$ is a normalizing factor
- σ is the spectral width parameter:

  σ_a for $\omega \leq \omega_p$

  σ_b for $\omega > \omega_p$.

Wave Damping

Damping of waves is possible by introducing resistance to vertical motion. STAR-CCM+ implements the method by Choi and Yoon [134], adding a resistance term to the equation for w-velocity:

$$S_z^{\mathrm{d}} = \rho\left(f_1 + f_2|w|\right)\frac{e^\kappa-1}{e^1-1}w \tag{925}$$

with:

$$\kappa = \left(\frac{x-x_{\mathrm{sd}}}{x_{\mathrm{ed}}-x_{\mathrm{sd}}}\right)^{n_d} \tag{926}$$

where:

- x_{sd} is the starting point for wave damping (propagation in the x-direction)
- x_{ed} is the end point for wave damping (boundary)
- f_1, f_2 and n_d are parameters of the damping model
- w is the vertical velocity component.

Wave Forcing

Following the approach of Kim and others ([141]), STAR-CCM+ allows coupling of the 3D flow simulation with a theoretical solution that is specified by VOF waves.

Forcing the solution of the discretized Navier-Stokes equations towards another solution (such as a theoretical solution or simplified numerical solution) over a specified distance reduces the computing effort by using a reduced-size solution domain. This forcing also eliminates problems that are associated with reflections of surface waves at boundaries, owing to the damping feature of the gradual forcing.

Wave forcing applies only to Momentum. No Phase sources or Turbulence sources are added. Wave forcing is achieved by adding a source term to the transport (momentum) equations of the form:

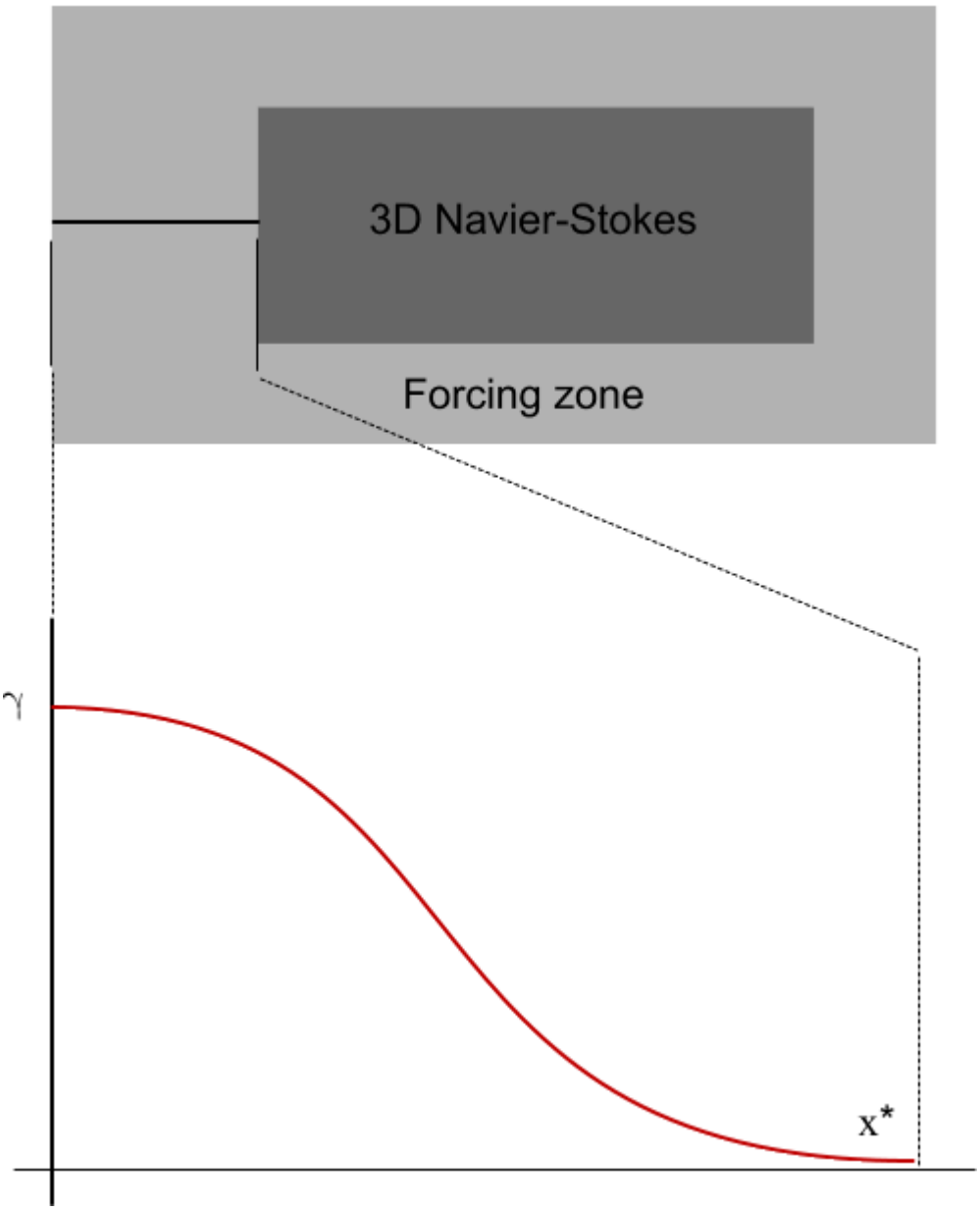
$$q_{\phi} = -\gamma\rho(\phi - \phi^*) \tag{927}$$

where:

- γ is the forcing coefficient
- ρ is the fluid density
- ϕ is the current solution of the transport equation
- ϕ^* is the value towards which the solution is forced.

When the solution needs to be fixed to a particular value of ϕ^*, a large number is used for γ. The remaining parts of the discretized equation then become negligible.

The source term from Eqn. (927) is applied with a variable forcing coefficient over a specified forcing zone. For example, the figure below is a schematic view of two overlapping solution domains. The 3D Navier-Stokes equations are solved in one domain, within the shaded zones. No forcing is applied within the inner zone (3D Navier-Stokes), but within the outer zone (Forcing zone) the forcing source term is activated along the solution domain boundaries. The width of the forcing zone can be different for different boundaries. The optimal width of the forcing zone depends on the problem that you are modeling.



The forcing coefficient varies smoothly from zero at the inner edge of the forcing zone to the maximum value γ at the boundary (the outer edge of the forcing zone).

$$\gamma = -\gamma_0 \cos^2(\pi x^*/2) \tag{928}$$

The volume coupling can also be either one-way or two-way, as the forcing zones usually do not coincide.

Contents:

VOF Waves Bibliography

VOF Waves Bibliography

[134] *Choi, Junwoo and Sung, Bum Yoon. 2009. “Numerical simulations using momentum source wave-maker applied to RANS equation model”, Coastal Engineering, 56 (10), pp. 1043 - 1060.*

[135] *Det Norske Veritas. 2007. “Environmental Conditions and Environmental Loads”, Recommended Practice DNV-RP-C205, April 2007, pp. 27.*

[136] *Det Norske Veritas. 2007. "Environmental Conditions and Environmental Loads", Recommended Practice DNV-RP-C205, April 2007 pp. 33.*

[137] *Fenton, John D. 1979. "A high-order cnoidal wave theory", J. Fluid Mech. 94, pp. 129–161.*

[138] *Fenton, John D. 1985. “A Fifth-Order Stokes Theory for Steady Waves”, J. Waterway, Port, Coastal and Ocean Eng., 111 (2), pp. 216-234.*

[139] *Fenton, John D. 1990. "Nonlinear wave theories", The Sea - Ocean Engineering Science, Part A, B. Le Méhauté & D. M. Hanes (eds), Vol. 9, Wiley, New York, pp. 3–25.*

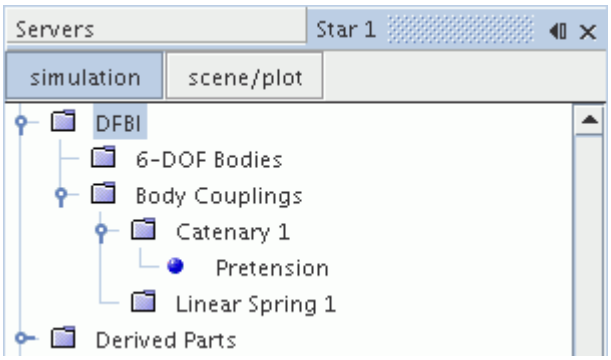
[140] *Fenton, John D. 1998. "The Cnoidal Theory of Water Waves", Ch. 2 of "Developments in Offshore Engineering", Ed. J. B. Herbich, Gulf, Houston*

[141] *Kim, J. W., O’Sullivan, J., & Read, A., 2012. "Ringing Analysis on a Vertical Cylinder by Euler Overlay Method", OMAE2012-84091.*

[142] *Korteweg, D. J. and de Vries, G., 1895. "On the Change of Form of Long Waves Advancing in a Rectangular Canal, and on a New Type of Long Stationary Waves", Philosophical Magazine 39 (240): 422–443.*

Modeling Dynamic Fluid Body Interaction

Use the DFBI (Dynamic Fluid Body Interaction) module to simulate the motion of a rigid body in response to pressure and shear forces the fluid exerts, and to additional forces you define. STAR-CCM+ calculates the resultant force and moment acting on the body due to all influences, and solves the governing equations of rigid body motion to find the new position of the rigid body.



Multiple rigid bodies can be modeled, with optional body couplings between them. The available couplings are:

- linear spring coupling
- catenary coupling
- contact coupling
- spherical joint coupling
- revolute joint coupling

The DFBI manager node only appears when one of the following DFBI motions has been created and applied to a region:

- DFBI Rotation and Translation
- DFBI Embedded Rotation
- DFBI Morphing
- DFBI Superposed Rotation

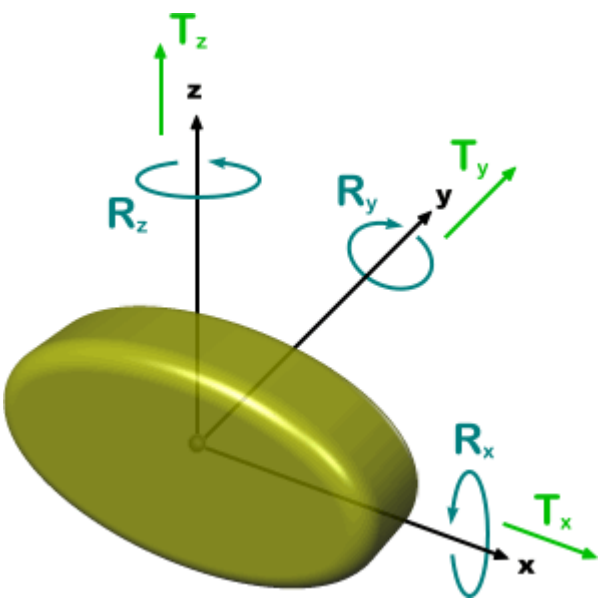
Contents:

What Is a Rigid Body?
What Is a DFBI Rotation and Translation Motion?
What Is a DFBI Embedded Rotation Motion?

What Is a DFBI Morphing Motion?
Setting up Superposing Motions
What Is the Six DOF Solver?
What Is the 6-DOF Motion Solver?
Working with 6-DOF Bodies
Working with 6-DOF Body Couplings
Setting up a Two-Dimensional DFBI Simulation
DFBI Formulation
DFBI Bibliography

What Is a Rigid Body?

A rigid body (defined using a 6-DOF Body in STAR-CCM+) can be defined as an object in which the relative distance between internal points does not change. Using the standard Cartesian coordinate system as a reference frame, it is possible for a rigid body to move (translate) along each of the three axes, and to rotate about the axes.



In the figure above:

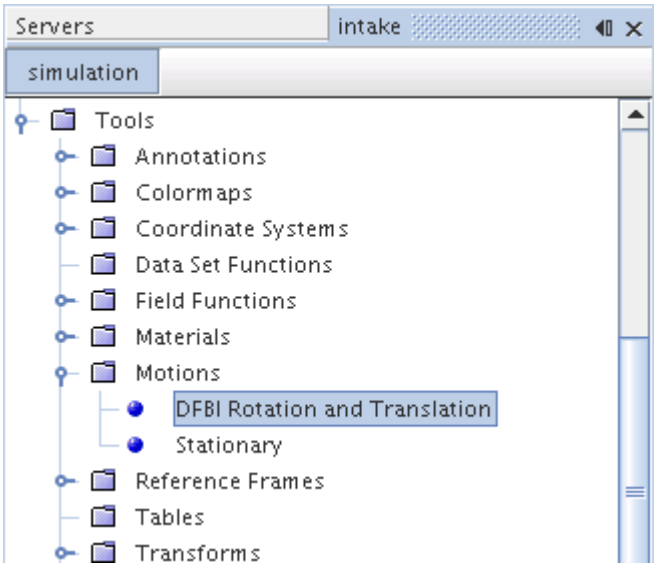
- Tx, Ty, and Tz are the translations in the x, y, and z directions respectively.
- Rx, Ry, and Rz are the rotations about the x, y, and z axes respectively.

For rigid bodies, it is sufficient to model the motion of the center of mass of the body alone. The relative motion of any other part of the body can be extrapolated from this center of mass. It is necessary to know the moments of inertia of the body about a fixed reference point (which is normally the center of mass) before the rotational motion can be known.

What Is a DFBI Rotation and Translation Motion?

This motion solves the equations of rigid body motion for all 6-DOF Bodies. When it is active, the **6-DOF Solver** and the **6-DOF Motion** nodes are added to the **Solvers** manager.

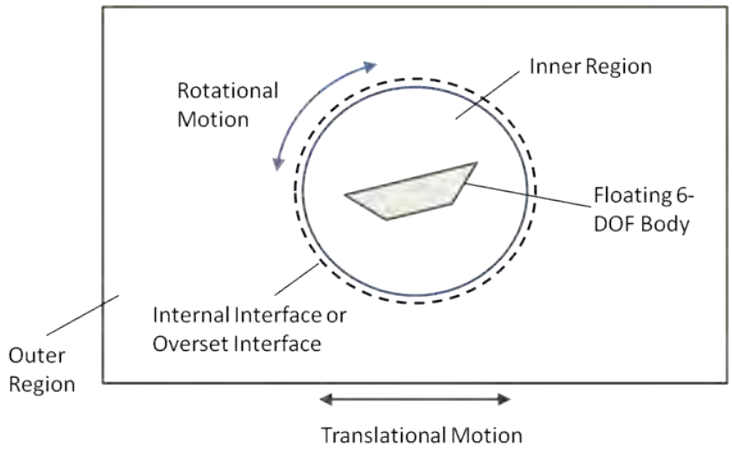
The **DFBI Rotation and Translation** motion node, which has no properties, is subject to common menu items.

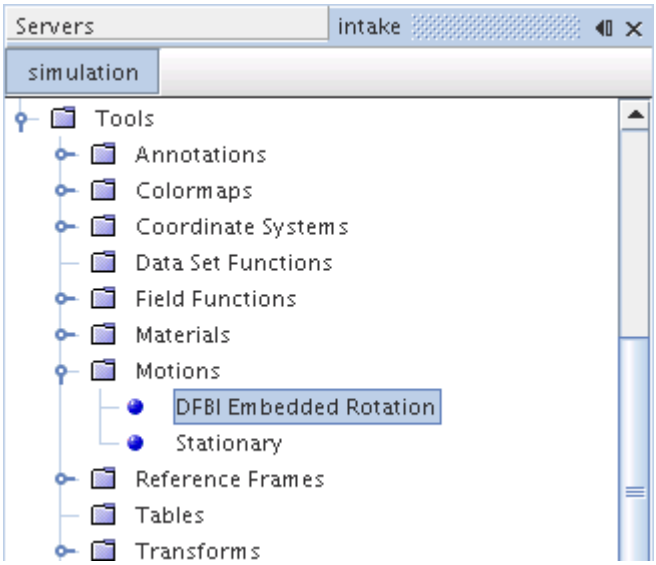


What Is a DFBI Embedded Rotation Motion?

The **DFBI Embedded Rotation** motion models a floating body within an inner region, which in turn is embedded in an outer region.

The following diagram shows the schematic setup of embedded rotation motion:





The inner and the outer region are connected through interfaces. There are two interface types available:

- Internal interface—the inner region must have the shape of a sphere or a cylinder. The internal interface act as a sliding plane along which the inner region rotates.
- Overset interface—the inner region can be an arbitrary shape.

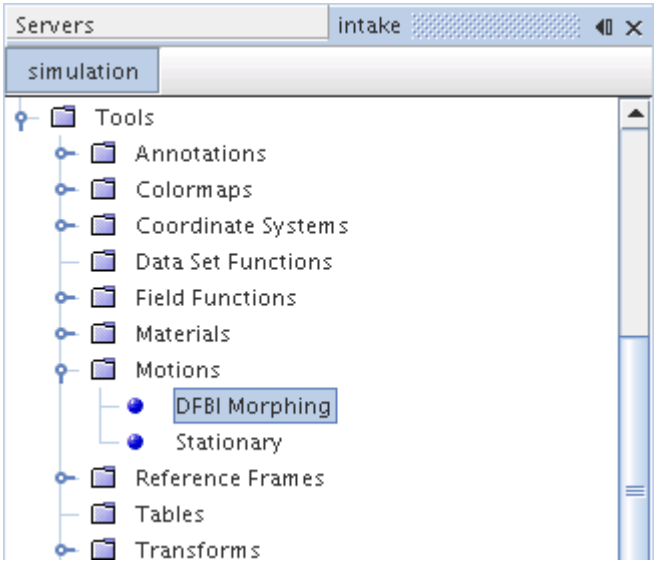
The fluid forces acting on the 6-DOF body can in general induce a motion that is composed of a rotational and a translational component. Within the inner region, STAR-CCM+ computes the full range of motion for the 6-

DOF body. When applicable, the translational motion component is transferred to the outer region and both inner and outer region translate. The embedded rotation motion approach is particularly beneficial for DFBI simulations in which the floating body rotates through large angles. To resolve the fluid flow due to the rotational motion accurately, you create a fine mesh within the inner region only. It is not required to extend the fine mesh resolution beyond the inner region. Choose the extents of the inner region such that it captures the zone of influence of the rotating floating body. Since the outer region performs the translational motion component, the inner region does not translate relative to the outer region.

What Is a DFBI Morphing Motion?

The DFBI Morphing motion is a variant of the Morphing motion that is used for DFBI (Dynamic Fluid Body Interaction). It activates customized boundary conditions so that you can use morphing in DFBI without making time-consuming adjustments.

The **DFBI Morphing** motion node is subject to common menu items.



Contents:

Setting up DFBI Morphing
Six DOF Morphing Properties

Setting up DFBI Morphing

You can use morphing motions together with DFBI by using customized morpher boundary conditions.

To set up DFBI morphing:

1. Right-click the **Tools** > **Motions** node and select **New** > **DFBI Morphing**.
2. Select the **Regions** > **[Region]** > **Physics Values** > **Motion Specificication** node and set *Motion* to **DFBI Morphing**.
3. Select the **[Region]** > **Boundaries** > **Physics Conditions** > **Six DOF morphing** node and set the Six DOF Morphing Properties.

Six DOF Morphing Properties

For the DFBI morphing motion, you set morphing conditions at the boundaries. The same boundary conditions as for the morphing motion are available. In addition to these standard morphing boundary conditions, you can set specific DFBI morphing boundary conditions.

For an overview of the standard morpher boundary conditions, see Morpher Boundary Condition.

Method	Sets the morphing condition.	
	Six Dof Body	Vertices on this boundary move according to the motion of a 6-DOF body. During initialisation, this morphing condition set automatically for boundaries who are selected as *Body Surface* to define the surface of the 6-DOF body (see What Is a 6-DOF Body). Can also be used for boundaries that do not define a 6-DOF body. In such cases, this morphing method is not applied automatically and must be set explicitly. Can also be used at interface boundaries that connect, for example, a DFBI morphing region with a DFBI rigid body motion region. Vertices on the interface boundary of the DFBI morphing region move according to the vertex motion at the interface boundary of the DFBI rigid body motion region. Adds the **Displacement Projection Option** node to the **Physics Conditions** node.
	Six Dof Body plus Displacement	Vertices on this boundary move according to the motion of a 6-DOF body where an additional specified displacement is superposed. Adds the **Morpher: Linear Displacement** node to the **Physics Values** node (see Morpher: LinearDisplacement).

Displacement Projection Option Properties

Project Displacement	Projects the **Six Dof Body** morpher displacement onto the provided direction. The vertices on this boundary are only allowed to move in the specified direction. Increases the flexibility of the **Six Dof Body** option in that only specified parts of the **Six Dof Body** displacement are allowed to move.	
	Activated	Enables **Six Dof Body** morpher displacement projection. Adds the **Displacement Projection** node to the **Physics Values** node.

	Deactivated	Disables **Six Dof Body** morpher displacement projection. This is the default.

Displacement Projection Properties

Projection	Specifies the direction of the displacement projection.
Coordinate System	Specifies the coordinate system in which the displacement projection is specified.

Setting up Superposing Motions

You can superimpose additional fixed body motions (rotational and/or translational) onto the DFBI Rotation and Translation Motion of a 6-DOF body. You can define multiple levels of nested rotational motions. These superposing motions let you model applications such as Cyclorotors or Azimuth thrusters attached to marine vessels. These propeller types combine a rotation of the propeller axis with a rotation of the attached blades. Other examples of complex motions can comprise a combination of rotational and translating motions, such as: daggerboards, rudders and retractable propellers on boats, floating wind-turbines and swinging crane motions on floating ships.

Before you begin:

- Set up a DFBI Rotation and Translation Motion
- Define a 6-DOF body
- Assign the DFBI Rotation and Translation Motion to the region around the 6-DOF body
- Create a region around the body whose motion you want to superpose

Note: **Superposing Motions** for DFBI is an extension of the previous **DFBI Superposed Rotation** motion. Simulations with a **DFBI Superposed Rotation** (see also What Is a DFBI Superposed Rotation Motion) setup are converted automatically to a **Superposing Motions** setup when only one 6-DOF body is defined.

To set up superposing motions:

1. Right-click the **DFBI** > **6-DOF Bodies** > **[Body 1]** node and select **Create Body Motion**.

   The **Tools** > **Motions** > **Motion of [Body 1]** node is created.

2. Right-click the **Tools** > **Motions** > **Motion of [Body 1]** > **Superposing Motions** node, select **New** from the pop-up menu and choose the option **Rotation** or **Translation**, as appropriate.

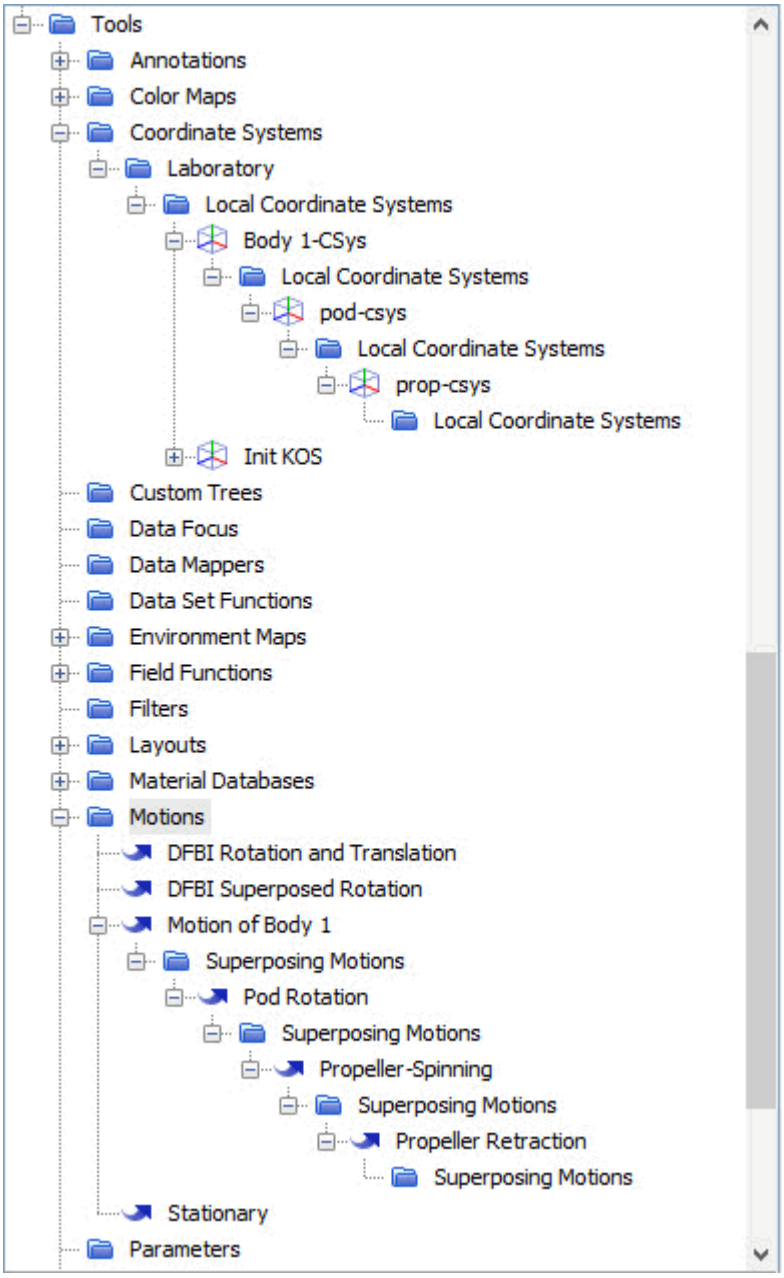
   The **Superposing Rotation** or **Superposing Translation** child node is created, along with a nested **Superposing Motions** child node.

3. To create nested superposed motions:

   a) Right-click the nested **Superposing Motions** node and create another set of superposing motions nodes.

   b) To create a local coordinate system as child coordinate system of the parent body's local coordinate system, right-click **Tools** > **Coordinate Systems** > **Laboratory** > **Body 1-Csys** > **Local Coordinate Systems** and select **New**.

      The nested superposed motion must be defined with respect to the body that has the parent superposed motion. The local coordinate system tracks the changing position of the parent body. The local coordinate systems for nested superposed motions must follow the same nested structure as the superposed motions. See also Superposing Additional Rigid Rotations.



In this example, **Pod Rotation** is a rotation that is superposed on the parent motion of **Body 1**. **Propeller-Spinning** is a rotation that is superposed on the parent **Pod Rotation** and **Propeller Retraction** is a translation that is superimposed on its parent **Propeller-Spining**

The structure of the coordinate systems corresponds to the structure of the superposing motions. The highest-level coordinate system is **Body 1-Csys**, which STAR-CCM+ creates automatically for **Body 1**. **pod-csys** is created relative to **Body 1-Csys** and **prop-csys** is nested underneath the parent **pod-csys**.

Repeating the above steps, you can create up to eight nested superposed motions of bodies that rotate and/or translate with respect to each other.

4. Select the **Motion of [Body 1]** > **Superposing Motions** node and define the superposing rotating motion by setting appropriate properties:

a) Specify the *Axis Direction* and the *Axis Direction* of the axis around which the region rotates.

b) Select the *Coordinate System* in respect to which the axis direction and origin are given. This coordinate system is managed by the parent motion.

c) Specify the magnitude of angular velocity in terms of *Rotation Rate*.

d) To model a nested superposed rotation, set *Managed Coordinate Systems* to the local coordinate system of the next lower level of nested superposed motion.

   In this example, the superposed motion **Pod Rotation** manages the coordinate system **prop-csys** and it is that system that is selected.

5. The process is similar for superposing translating motion, except that there is no requirement to set an axis and you follow these steps:

   a) Select the *Coordinate System*. This coordinate system is managed by the parent motion.

   b) Specify the magnitude of translational velocity in the *X*, *Y* and *Z* coordinate directions.

   c) To model a nested superposed translation, set *Managed Coordinate Systems* to the local coordinate system of the next lower level of nested superposed motion.

      In this example, the superposed motion **Propeller Retraction** manages the coordinate system **pod-csys** and it is that system that is selected.

6. Select **Regions** > **[Region that contains body with superposed motion]** > **Physics Values** > **Motion Specification** and set *Motion* to **[Body 1]-Motion->Superposing Rotation**.

   To model a nested superposing motion, assign the lowest-level motion to the region. In this example, **Superposing Motions** > **Propeller Retraction** is assigned to the translating region. You must not assign **[Body 1]-Motion** to the region, but always the **Superposing Rotation** that you defined.

Contents:

Superposing Motions Reference
What Is a DFBI Superposed Rotation Motion?

Superposing Motions Reference

Superposing Motions Right-Click Actions

New

Creates a new **Superposing Rotation** node that lets you define a superposing rotating motion. In addition, a **Superposing Motions** child node is created. Right-clicking this child node lets you define a further new superposing rotation.

Superposing Rotation Properties

You define a superposing rotating motion that rotates about a specified axis. The superposing rotating motion is associated with a region.

Axis Direction

The vector that specifies the direction of the axis.

Axis Origin

The vector that defines the origin of the axis.

Rotation Rate

The rate of rotation with which the region rotates.

Coordinate System

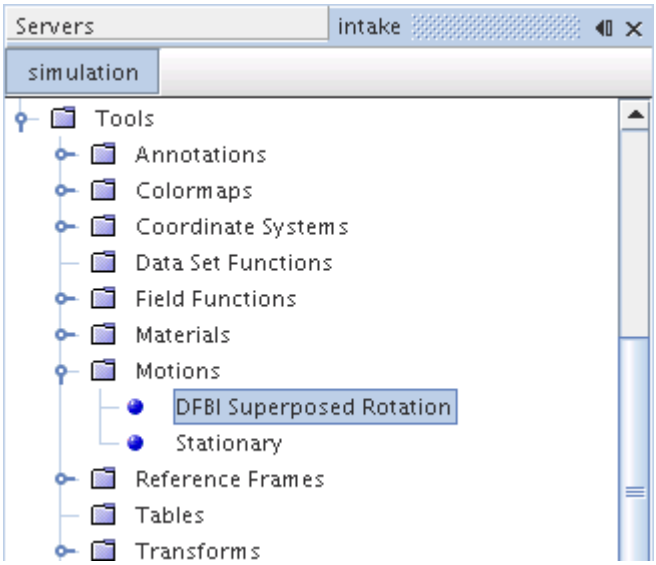
The coordinate system in respect to which *Axis Direction* and *Axis Origin* are defined.

Managed Coordinate Systems

User-defined local coordinate system that moves according to the user-defined superposed motion. Can be used to define a nested superposed rotation that uses this coordinate system as a reference for the parent motion.

What Is a DFBI Superposed Rotation Motion?

The DFBI Superposed Rotation option superimposes an additional fixed body rotation in addition to the DFBI motion. This option makes it possible to model propellers attached to marine vessels, for example.



Contents:

Setting up DFBI Superposed Rotation
DFBI Superposed Rotation Motion Properties

Setting up DFBI Superposed Rotation

You use DFBI superposed rotation together with a DFBI rotation and translation motion. You can create one or several DFBI superposed rotations. Each of these motions must be set up with a separate region. The individual regions are connected via internal interfaces.

Before you begin to set up a DFBI superposed rotation:

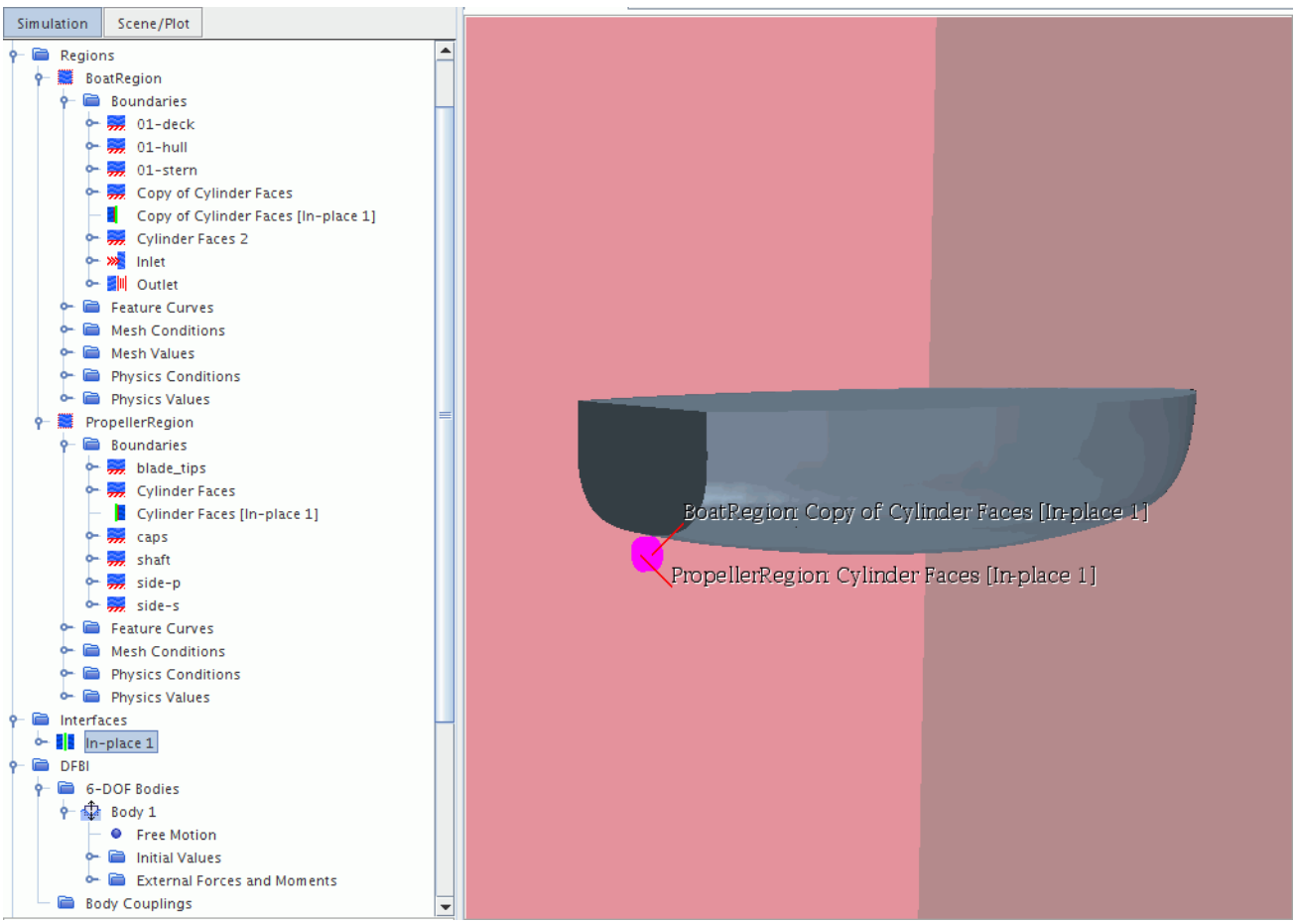
- Set up a DFBI rotation and translation motion.

  See What is DFBI Rotation and Translation.
- Create a region around the body whose motion you want to superpose.
- Create an internal interface between the surrounding region boundaries containing the superposing DFBI body and the region boundaries containing the DFBI body.

To set up a DFBI superposed rotation:

1. Right-click the **Tools** > **Motions** node and select **New** > **DFBI Superposed Rotation**.
2. Select the **Motions** > **DFBI Superposed Rotation** node and set the DFBI Superposed Rotation Motion Properties.
3. Select the **Regions** > **[Region]** > **Physics Values** > **Motion Specification** node and set *Motion* to **DFBI Superposed Rotation**.

The following screenshot shows an example setup of a DFBI superposed rotation simulation.



The boat is defined as a 6-DOF body. The red surfaces are the boundaries of the **BoatRegion**. The rotating propeller is placed inside a cylindrical region, here called **PropellerRegion**. The boundaries of the **PropellerRegion** and the **BoatRegion** are connected through internal in-place interfaces. The

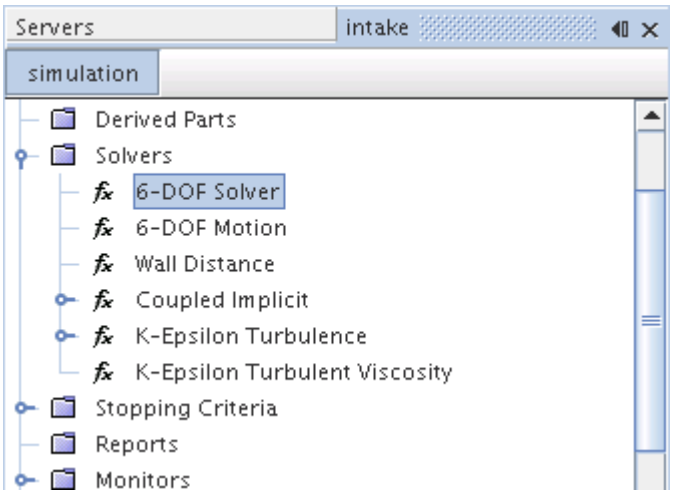
PropellerRegion is associated with the **DFBI Superposed Rotation** motion and the **BoatRegion** is associated with the **DFBI Rotation and Translation** motion.

DFBI Superposed Rotation Motion Properties

Direction	The direction of the axis. Specified in the Body Local Coordinate system.
Origin	The origin of the axis. Specified in the Body Local Coordinate system.
Rotation Rate	The angular velocity of the rotation about the axis.

What Is the Six DOF Solver?

The **6-DOF Solver** computes fluid forces, moments, and gravitational forces on a 6-DOF body. Pressure and shear forces are integrated over the surfaces of the 6-DOF bodies. More forces and moments can be added.



The forces and moments acting on the 6-DOF body are used to compute the translational motion of the center of mass of the body and the angular motion of the orientation of the body.

For time integration, the 6-DOF solver employs a trapezoidal scheme of second order accuracy. This order of accuracy is independent of the order of accuracy of the implicit unsteady solver.

Contents:

Six DOF Solver Properties

Six DOF Solver Properties

Verbosity	Provides additional output in the *Output* window.	
	None	No additional output.
	Low	Available only for multi-body motion. The output is the normalized drift from the constraint equations Eqn. (972) for each joint that is defined as part of a multi-body simulation. For more information, see Constraint Stabilization. • Spherical Joint: Drift in position. • Revolute Joint: Drift in position and the angular drift.

		Provides output per iteration.
	High	Available only for multi-body motion. The output is the normalized drift from the constraint equations Eqn. (972) for each joint that is defined as part of a multi-body simulation. For more information, see Constraint Stabilization. • Spherical Joint: Drift in position. • Revolute Joint: Drift in position and the angular drift. Provides output per sub-step.
	Diagnostics	Upon initialization, provides output about which regions and boundaries are *driven* by the 6-DOF body motion. In this context, *driven* means that the listed regions and boundaries are associated with the 6-DOF body and move with it.
Maximum Number of Iterations	Maximum number of iterations within a time-step.	

Six DOF Solver Expert Properties

Solver Frozen	Prevents or allows the solver from updating the solution while iterating. Activating this feature should not be your first choice, as it is primarily a “debugging” tool and not supported for normal simulation work.	
	Activated	The solver does not update the solution.
	Deactivated	The solver updates the solution.
Sub-Stepping Enabled	Activates or deactivates the application of a smaller time-step for the solution of external forces on a 6-DOF body. This option can be activated to overcome convergence problems due to strong non-linear external forces such as catenary forces. This option must not be used generally, but only to enhance stability for these special cases. With sub-stepping activated, the fluid mesh and the fluid forces are updated in each inner iteration and not sub-step-wise. Sub-stepping is presently only available for the *Body Motion Option* **Free Motion** (see 6-DOF Body Properties).	
	Activated	Sub-stepping is activated. A **Sub-Stepping** sub-node is created where you specify the *Number of Sub-Steps* during a time-step. If you specify a large number of sub-steps, you are advised to increase the number of inner iterations to

		ensure that the sub-steps converge.
	Deactivated	Sub-stepping is not activated.
Multi-Body Constraint Stabilization	When **On**, activates the Baumgarte constraint stabilization. The constraint stabilization is required to stabilize the kinematic constraints that are imposed by a joint between two 6-DOF bodies. **Multi-Body Constraint Stabilization** is always activated by default. This option is used only for multi-body motion simulations with multi-body joint couplings. For other simulation types, this option is ignored. For more information, see Constraint Stabilization. You specify the coefficients of the Baumgarte constraint stabilization in the **Baumgarte Constraint Stabilization** sub-node.	
Legacy Force Update	When **activated**, STAR-CCM+ reverts to the update scheme that was used before STAR-CCM+ 12.02. In this previous update scheme, forces, moments, couplings, and constraints were updated at the end of each inner iteration, but not additionally at the end of the time-step, except for the case that the number of inner iterations exceeded the maximum number of 6-DOF iterations. When **deactivated**, forces, moments, couplings, and constraints are always updated at the end of the time-step, in addition to the updates during the inner iterations.	

Baumgarte Constraint Stabilization Properties

Alpha, *Beta*

α and β parameters of the Baumgarte constraint stabilization method in Eqn. (979). The parameter α has a damping effect whereas the parameter β corresponds to a stiffness. Typical values for both parameters range between 1–30. If you set the parameters to overly large values, the simulation can become unstable, especially if combined with a large time-step.

Setting $\alpha = \beta$ corresponds to critical damping which means that constraint deviations decay most rapidly. However, large values of α can lead to energy losses due to damping. Therefore, the default values use a small α in combination with a larger β to stabilize the constraints effectively and to improve the energy conservation.

Multi-Body Constraint Solver Properties

The **Multi-Body Constraint Solver** node appears when at least one 6-DOF body is associated with multi-body motion. The expert properties that you adjust here are parameters of an SOR (Successive Over-Relaxation) Gauss-Seidel algorithm. The SOR Gauss-Seidel algorithm solves the constraint equations that are necessary for freezing motion components when setting Multi-Body Motion Properties. The solver is not required for the numerical solution of the constraint equations for the mechanical joints, which have their own specialized solver.

Verbosity

Provides additional output in the *Output* window to monitor convergence of the multi-body constraint solver (SOR Gauss-Seidel algorithm).

Maximum Number of Iterations

For each inner iteration of the 6-DOF solver, specifies the maximum number of iterations allowed for the SOR Gauss-Seidel algorithm.

Residual Tolerance

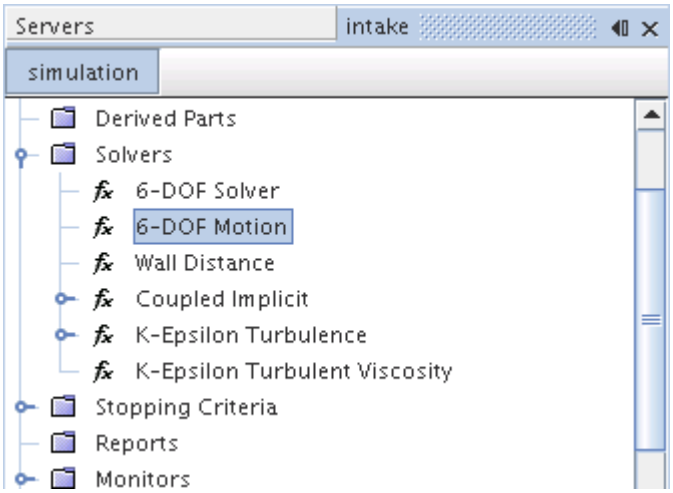
The SOR Gauss-Seidel algorithm stops iterating when either the specified *Residual Tolerance* or the *Maximum Number of Iterations* are reached.

Successive Overrelaxation Factor

Factor to enhance convergence of the SOR Gauss-Seidel algorithm. Assign values between 1 and 2 (1.5 is the default). Reduce below 1 when you must stabilize the numerical solution.

What Is the 6-DOF Motion Solver?

The 6-DOF Motion Solver moves the vertices of the grid according to motion of the body that is calculated by the 6-DOF solver.



Contents:

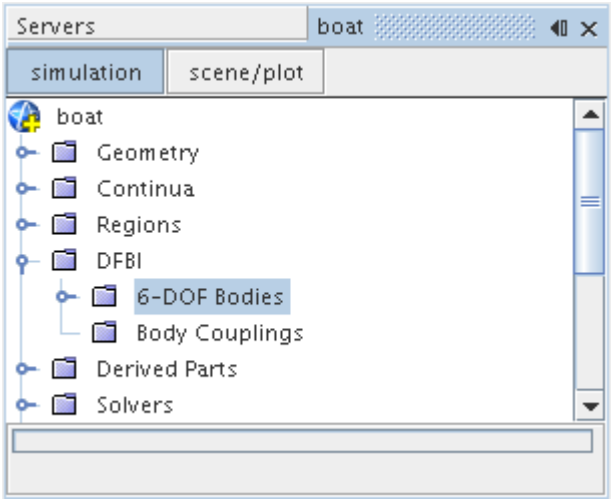
6-DOF Motion Solver Expert Properties

6-DOF Motion Solver Expert Properties

Solver Frozen	Prevents or allows the solver from updating the solution while iterating. Activating this feature should not be your first choice, as it is primarily a “debugging” tool and not supported for normal simulation work.	
	Activated	The solver does not update the solution.
	Deactivated	The solver updates the solution.

Working with 6-DOF Bodies

The **6-DOF Bodies** manager node is the manager object for working with 6-DOF bodies in the simulation tree. It is added to the tree once you select one of the DFBI motions for a region.

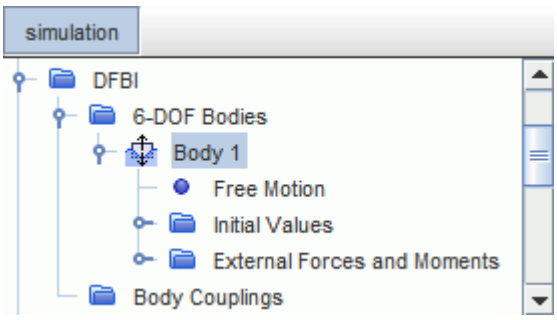


Contents:

What Is a 6-DOF Body?
Creating 6-DOF Bodies
What are Initial Values?
What Is a Free Motion?
What Is One-DOF Rotating Motion?
What Is One-DOF Translating Motion?
What is Four-DOF Maneuvering Motion?
What is Planar Motion Carriage?
Equilibrium
Setting up Multi-Body Motion
What Are External Forces and Moments?
6-DOF Coordinate Systems

What Is a 6-DOF Body?

In STAR-CCM+, the 6-DOF body defines the surface of the rigid body for the calculations of the 6-DOF solver. To compute the interaction between body and fluid flow, modeling of the body's interior is not required. If you want to include heat transfer between the rigid body and the fluid in conjunction with DFBI, STAR-CCM+ permits the rigid body to contain a solid region. The 6-DOF body is created with a DFBI motion.



6-DOF bodies can be renamed, copied, and deleted in the same way as other simulation objects. However, when a 6-DOF body is copied, any existing linear spring forces or catenary forces in its External Force and Moment manager are ignored. This is because these forces are managed by their respective coupling elements which exist outside of all 6-DOF body nodes. Other forces and moments are copied.

6-DOF Body Properties

Body Surface	A list of boundaries that represent the surface of the body. Opens the in-place part selector for editing. (Integration of shear force and pressure is over the body surface.) • Select the wall boundaries that compose the body surface. • If the 6-DOF body contains a solid region, select the fluid side boundaries and the corresponding boundaries for the interface between the fluid and the solid region. The associated interfaces can be of type contact or mapped.	
Body Mass	Mass of the object.	
Release Time	Time before calculation of body motion begins. It is best to allow some time for the fluid flow to initialize. The number of seconds depends on the time model you are using and the mesh size. In an unsteady model, the typical range would be 10 to 50 time-steps. If a time-step size is equivalent to 0.01 seconds, 50 time-steps would amount to 0.5 seconds.	
Ramp Time	Facilitates a more robust solution by reducing oscillation. At release time, force and moments are applied to the body and can cause a shock effect. Applying a ramp time applies the forces and moments proportionally across the interval to reduce the shock effect. The value of this property is typically 10 times the release time.	
Body Motion Option	Selects the type of body motion from a drop-down list. **Free Motion** is the default motion type.	
	Free Motion	Selects the Free Motion.
	Equilibrium	Selects the Equilibrium Motion.
	One-DOF Rotating Motion	Selects the One-DOF Rotating Motion.

	One-DOF Translating Motion	Selects the One-DOF Translating Motion.
	Four-DOF Maneuvering Motion	Selects the Four-DOF Maneuvering Motion.
	Planar Motion Carriage	Selects the Planar Motion Carriage.
	Multi-Body Motion	Selects the Multi-Body Motion.
Three Dimensional	(Read-only) When on, the body is three-dimensional. When off, it is two-dimensional.	
Index	(Read-only) Unique number to identify the body.	

Contents:

Pop-Up Menu of the 6-DOF Bodies Manager

Pop-Up Menu of the 6-DOF Bodies Manager

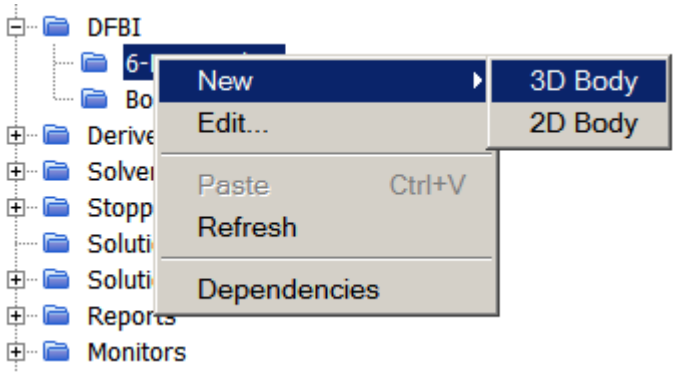
New	Adds a 6-DOF body to the **6-DOF Bodies** node.
Edit	Executes the customize object action to allow editing of the selected object and its children using a separate modeless stay-on-top dialog.
Paste	Executes the paste action to replace the properties of the current object with properties from the clipboard.
Refresh	Updates the selected object or display to reflect the latest changes to the simulation.

Creating 6-DOF Bodies

You can create two-dimensional or three-dimensional 6-DOF bodies.

To create a 6-DOF body:

1. Right-click the **DFBI** > **6-DOF Bodies** node and do one of the following:
   - To create a three-dimensional 6-DOF body, select **New** > **3D Body**.
   - To create a two-dimensional 6-DOF body, select **New** > **2D Body**.



This action adds a 6-DOF body object to the simulation tree.

What are Initial Values?

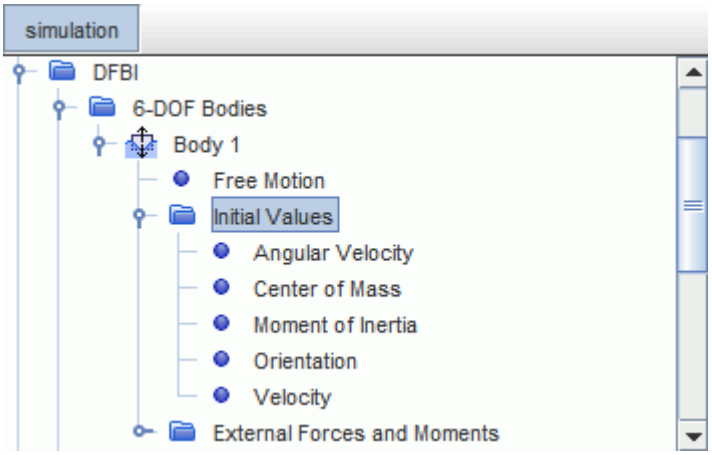
You must specify initial values for the 6-DOF body, which are used to define the body at the start of the simulation.

These initial values comprise:

- Angular Velocity
- Center of Mass
- Moment of Inertia
- Orientation
- Velocity
- Attachment Point

You set the initial values of the 6-DOF body in the subnodes of the **Initial Values** node. Not all of the listed initial values must be set for each motion type. Depending on the selection of the *Body Motion Option* in the **Body 1** node, the subnodes adjust dynamically.

The following screenshot shows the subnodes when a **Free Motion***Body Motion Option* is selected in the **Body 1** node.



You specify all of the initial values of the 6-DOF body with respect to an arbitrary coordinate system.

Angular Velocity Properties

Value	Specifies the initial angular velocity of the 6-DOF body. The initial angular velocity is specified as a vector with respect to **Coordinate System**.
Coordinate System	Selects the coordinate system with respect to which the initial angular velocity is specified.

Center of Mass Properties

Position	Specify the initial position of the center of mass of the 6-DOF body.
Coordinate System	Selects the coordinate system with respect to which the initial center of mass is specified.

Moment of Inertia Properties

Diagonal Components	The diagonal components of the initial moments of inertia tensor (Eqn. (931)).	
Coordinate System	Selects the coordinate system with respect to which the components of the Moments of Inertia tensor are specified. The coordinate system must be Cartesian.	
Off Diagonal Components	The off-diagonal components of the initial moments of inertia tensor (Eqn. (931)).	
Use Center of Mass	**Activated**	The Moments of Inertia tensor is given with respect to the initial center of mass of the body.
	Deactivated	The Moments of Inertia tensor is given with respect to the origin of the selected *Coordinate System*.

Orientation Properties

Coordinate System	Selects the coordinate system which defines the initial orientation of the 6-DOF body local coordinate system. The coordinate system must be Cartesian.

Velocity Properties

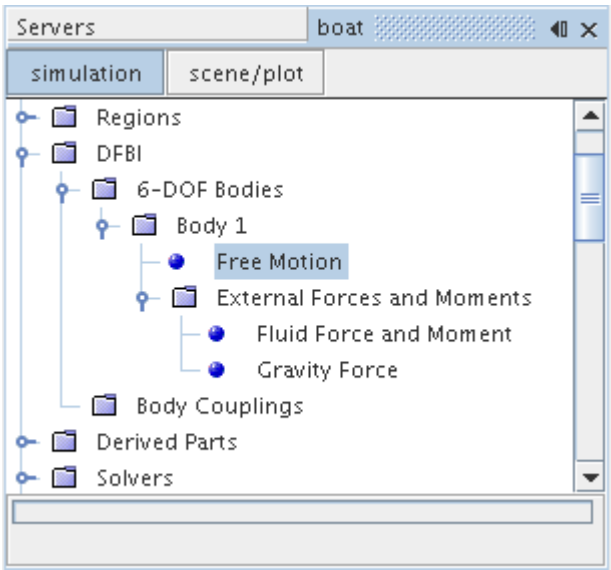
Value	Specify the initial velocity of the 6-DOF body.
Coordinate System	Selects the coordinate system with respect to which the initial velocity is specified.

Attachment Point Properties

Value	Specify the point where the 6-DOF body is initially attached to the planar motion carriage.
Coordinate System	Selects the coordinate system with respect to which the attachment point is specified.

What Is a Free Motion?

A free motion object lets you control which direction of motion is calculated, and which ones to leave frozen.



These motions are specified with respect to the laboratory coordinate system. Rotations are defined as positive according to the right-hand convention around each coordinate axis. For the purposes of rotation, the laboratory coordinate system can be considered as though its origin were moved to the center of mass of the 6-DOF body.

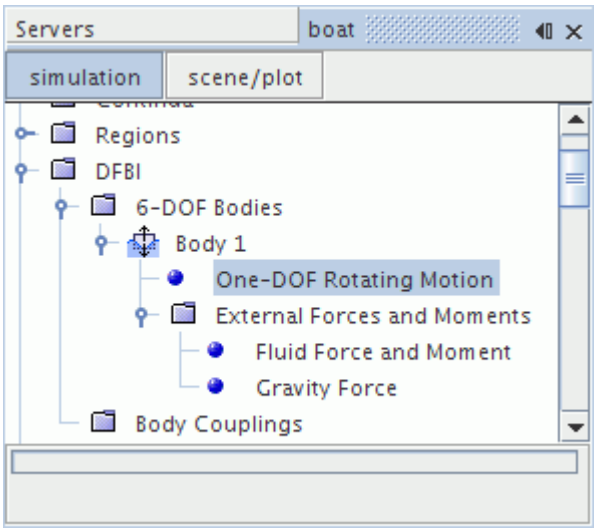
Free Motion Properties

X Motion	Calculate motion along the X-axis of the laboratory coordinate system.
Y Motion	Calculate motion along the Y-axis of the laboratory coordinate system.
Z Motion	Calculate motion along the Z-axis of the laboratory coordinate system.
X Rotation	Calculate rotation about the X-axis of the laboratory coordinate system (as transposed to the center of mass of the 6-DOF body).
Y Rotation	Calculate rotation about the Y-axis of the laboratory coordinate system (as transposed to the center of mass of the 6-DOF body).

Z Rotation	Calculate rotation about the Z-axis of the laboratory coordinate system (as transposed to the center of mass of the 6-DOF body).

What Is One-DOF Rotating Motion?

A One-DOF Rotating Motion object lets you rotate a body around a specified axis.



Each one-DOF rotating body can have its own axis of rotation; it is therefore possible to simulate several bodies having different rotation axes (for example, wind turbines in a wind farm).

Optionally, you can use a motion limiter to constrain the angular motion of the body between a minimum and a maximum angle (for example, for valve applications). The body is stopped when a motion limit is reached: the kinetic energy of the body is then destroyed. This physical situation is extreme and can have a significant influence on flow field residuals and convergence. Therefore, if a motion limiter is used, preferably use the default option: a damped motion limiter with which the body stops more smoothly.

You can also specify the angular damping length over which the body is stopped. The angular damping length is effective if the time step is small enough for the body to make several steps over the damping length.

The axis of rotation must be constant in time. Otherwise, the body has more than one degree of freedom, which is beyond the scope of the One-DOF Rotating Motion. Only use steady coordinate systems in the *Coordinate System* property; unsteady coordinate systems lead to unsteady axes of rotation. Body coordinate systems cannot be used, since these coordinate systems also change with time.

To report the rotation angle of the body, use the Rigid Body 1-DOF Angle report. Use the rotation angle report only with bodies undergoing **One-DOF Rotating Motion** since it reports zero for all other motions. For these other cases, use the 6-DOF Body Orientation report.

One-DOF Rotating Motion Properties

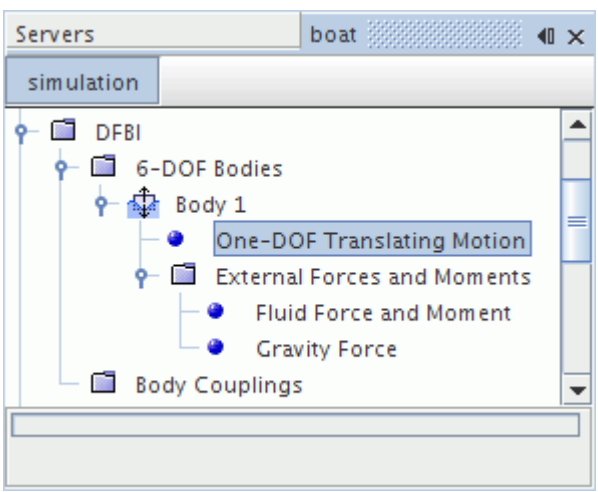
Rotation Axis Origin	The origin of the rotation axis.
Rotation Axis Direction	The direction of the rotation axis.

One-DOF Rotating Motion Expert Properties

Coordinate System	The coordinate system in which the origin and the direction of the rotation axis are specified.	
Motion Limits	Whether to activate the motion limiter (Eqn. (934) or Eqn. (935)).	
	Activated	Activates the motion limiter.
	Deactivated	Deactivates the motion limiter.
Minimum Angle	The minimum angle α_{min} at which the body motion is stopped if the motion limiter is enabled. The angle is measured with respect to the initial orientation of the body. The initial angle is 0. See Eqn. (934).	
Maximum Angle	The maximum angle α_{max} at which the body motion is stopped if the motion limiter is enabled. The angle is measured with respect to the initial orientation of the body. The initial angle is 0. See Eqn. (935).	
Angular Damping Length	The angular length λ (before the minimum and maximum angle) over which the stopping process takes place if the Motion Limiter for One-DOF Rotating Motion is enabled.	

What Is One-DOF Translating Motion?

A One-DOF Translating Motion object lets you translate a body along a specified direction.



Each one-DOF translating body can have its own direction of motion; it is therefore possible to simulate several bodies having different translation directions.

Optionally, you can use a motion limiter to constrain the translating motion of the body between a minimum and a maximum displacement (for example, for valve applications). The body is stopped when a motion limit is reached: the kinetic energy of the body is then destroyed. This physical situation is extreme and can have a significant influence on flow field residuals and convergence. Therefore, if a motion limiter is used, preferably use the default option: a damped motion limiter with which the body stops more smoothly.

You can also specify the damping length over which the body is stopped. The damping length is effective if the time step is small enough for the body to make several steps over the damping length.

The direction of motion must be constant in time. Otherwise, the body has more than one degree of freedom, which is beyond the scope of the One-DOF Translating Motion. Only use steady coordinate systems in the Coordinate System property; unsteady coordinate systems lead to unsteady directions of motion. Body coordinate systems cannot be used, since these coordinate systems also change with time.

One-DOF Translating Motion Properties

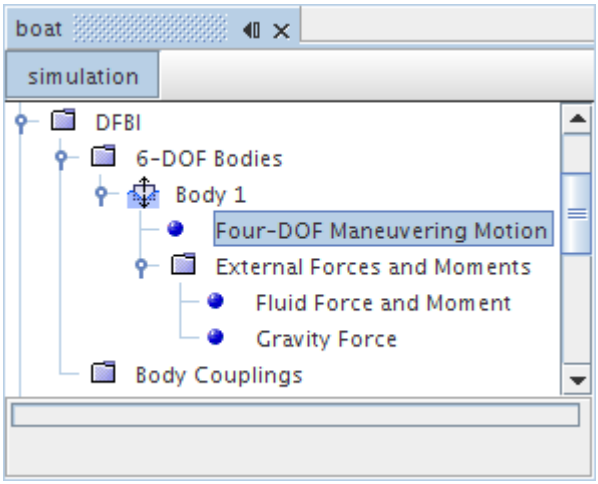
Translation Origin	The origin of the translational motion. This point is treated as the initial body position, which can be different from the initial center of mass.
Translation Direction	The direction of the translational motion.

One-DOF Translating Motion Expert Properties

Coordinate System	The coordinate system in which the origin and the direction of the translational motion are specified.	
Motion Limits	Whether to activate the motion limiter. (Eqn. (936) and Eqn. (937))	
	Activated	Activates the motion limiter.
	Deactivated	Deactivates the motion limiter.
Minimum Position	The minimum displacement d_{min} at which the body motion is stopped if the motion limiter is enabled. The displacement is measured with respect to the initial position of the body. The initial displacement is 0.	
Maximum Position	The maximum displacement d_{max} at which the body motion is stopped if the motion limiter is enabled. The displacement is measured with respect to the initial position of the body. The initial displacement is 0.	
Damping Length	The length λ (before the minimum and maximum position) over which the stopping process takes place if the motion limiter is enabled.	

What is Four-DOF Maneuvering Motion?

A Four-DOF Maneuvering Motion object lets you specify the speed and direction of a body in the X-Y plane, while leaving the body free to maneuver in the other four degrees of freedom.



Each Four-DOF maneuvering body can have its own speed and direction of motion; it is therefore possible to simulate several bodies having different velocities. For each body, the speed and direction in the X-Y plane is constant in time. The option uses the laboratory coordinate system throughout.

When you run an Orientation report for Four-DOF Maneuvering Motion, use the **Rotation X-Y-Z** option for the *Euler Angle Convention* property.

Four-DOF Maneuvering Motion Properties

Course Velocity	The speed of the body in the X-Y plane. The default is zero.
Course Angle	The direction of motion of the body in the X-Y plane, given as the counter-clockwise angle from the X-axis. The default is zero.

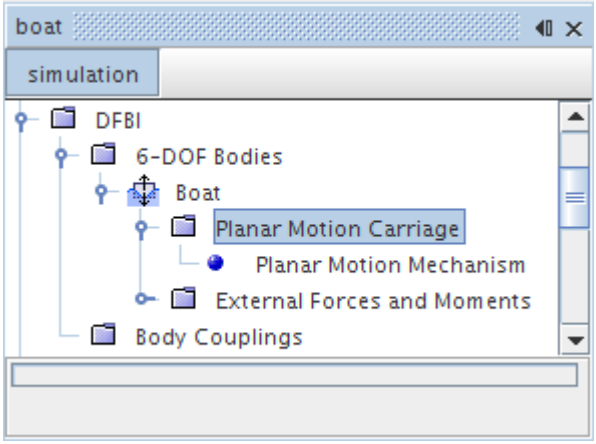
What is Planar Motion Carriage?

The Planar Motion Carriage option simulates a captive motion in the X-Y plane of the laboratory coordinate system. This mechanism drives the body along a prescribed trajectory in the X-Y plane, while the body is optionally allowed to move freely in the directions of heave, pitch, and roll.

There are three Planar Motion methods available:

- Planar Motion Mechanism (PMM)
- Rotating Arm Motion
- General Planar Motion

The **Planar Motion Carriage** node has properties and is subject to common menu items.



The option uses the laboratory coordinate system throughout. The Initial Orientation Coordinate System of the body must be aligned with the laboratory system.

When you run an Orientation report for the Planar Motion Carriage, use the **Rotation X-Y-Z** option for the *Euler Angle Convention* property.

Planar Motion Carriage Properties

Planar Motion Option	The type of planar motion that you want your rigid body to perform.	
	Planar Motion Mechanism	Select the Planar Motion Mechanism (PMM). An additional **Planar Motion Mechanism**sub-node is added to the simulation tree where you can define the properties for this planar motion option. For more information, see What is Planar Motion Mechanism?
	Rotating Arm	Select the Rotating Arm Motion. An additional **Rotating Arm**sub-node is added to the simulation tree where you can define the properties for this planar motion option. For more information, see What_is_Rotating_Arm_Motion_
	General Planar Motion	Select the General Planar Motion. An additional **General Planar Motion**sub-node is added to the simulation tree where you can define the properties for this planar motion option. For more information, see What is General Planar Motion?
Free Heave	When activated, this option allows the body to move along the Z-axis.	
	Activated	The body is free to move on the Z-axis. This is the default.
	Deactivated	No Z-axis motion allowed.
Free Pitch	When activated, this option allows the body to rotate around its own Y-axis.	
	Activated	The body is free to rotate around its Y-axis. This is the default.

	Deactivated	No Y-axis rotation allowed.
Free Roll	When activated, this option allows the body to rotate around its own X-axis.	
	Activated	The body is free to rotate around its X-axis. This is the default.
	Deactivated	No X-axis rotation allowed.

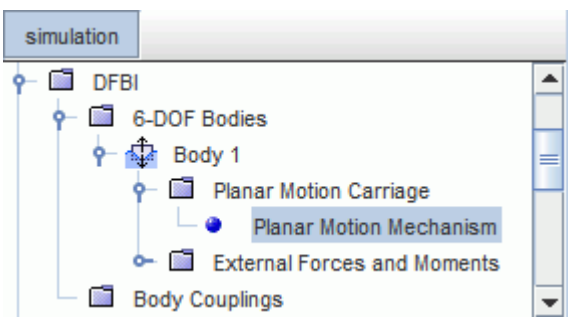
Contents:

What is Planar Motion Mechanism?
What is Rotating Arm Motion?
What is General Planar Motion?

What is Planar Motion Mechanism?

As part of the Planar Motion Carriage, the Planar Motion Mechanism (or PMM) simulates the motion of a rigid body in the X-Y plane of the laboratory coordinate system along a sinusoidal path.

The body moves with a constant velocity in X direction and performs an oscillatory motion in Y direction [143].



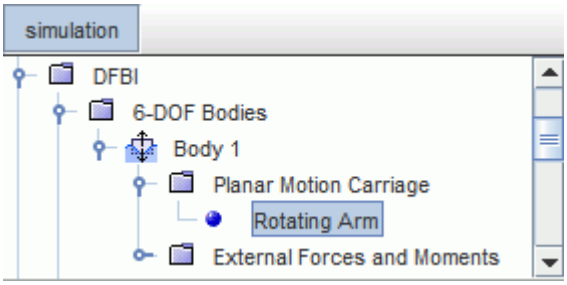
Planar Motion Mechanism Properties

Velocity X	The magnitude of the velocity along the X-axis. You must specify a constant value. The default is 1.0 m/s.	
Amplitude Y	The maximum displacement of the body along the Y-axis. The default is zero. This corresponds to the sway amplitude y_0 in Eqn. (964).	
Oscillation Frequency	The number of cycles per second. The default is zero. This corresponds to the sway frequency f in Eqn. (964).	
Additional Drift Angle	You can specify an additional drift angle of the body. If specified, this value is added to the yaw angle. The drift angle must be a constant value.	
Pure Yaw	With **Pure Yaw**, the body rotates back and forth in the X-Y plane (yaw) so that the same face (the bow, for a boat) always leads along the sinusoidal path. The *Additional Drift Angle* is added if specified. The yaw angle is given by Eqn. (965).	
	Activated	Body moves with pure yaw.

	Deactivated	Body does not change orientation as it sways. This is the default.

What is Rotating Arm Motion?

As part of the Planar Motion Carriage, the Rotating Arm motion simulates the motion of a rigid body in the X-Y plane of the laboratory coordinate system in a rotating action. The result is a circular path of the body in the X-Y plane.



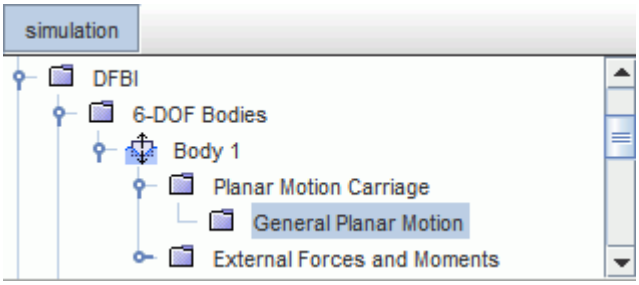
Rotating Arm Properties

Center of Rotation	The coordinates of the point, $\mathbf{r}_c$, around which the rigid body rotates with respect to the *Coordinate System* (see Eqn. (967)).
Drift Angle	The drift angle of the body. The drift angle must be a constant value. The drift angle is added to the yaw angle, which is due to the rotating arm motion.
Angular Velocity	The angular velocity, ω , of the rotating arm (see Eqn. (967)).
Coordinate System	The coordinate system in respect to which the center of rotation is specified.

What is General Planar Motion?

As part of the Planar Motion Carriage, the General Planar Motion simulates the motion of a rigid body in the X-Y plane of the laboratory coordinate system in a user-defined action.

You prescribe a trajectory for the motion in the X-Y plane and a yaw angle. The trajectory and the yaw angle are functions of time. The corresponding heave and pitch motions are computed by STAR-CCM+.



General Planar Motion Properties

Translation X	The translational motion of the rigid body in the X direction with respect to the laboratory coordinate system. You specify the translation as a function of time. To do this, you enter the function as an Expression.	
Translation Y	The translational motion of the rigid body in the Y direction with respect to the laboratory coordinate system. You specify the translation as a function of time. To do this, you enter the function as an Expression.	
Rotation Z	The rotational motion of the rigid body around the Z axis with respect to the laboratory coordinate system. You specify the rotation angle as a function of time. To do this, you enter the function as an Expression.	
User Defined Derivatives	To calculate the motion of a body defined by the General Planar Motion mechanism, the first and second order time derivatives of the prescribed trajectory are required. By default, STAR-CCM+ computes these derivatives numerically. If the accuracy of this numerical computation is considered insufficient, you can specify the first and second order derivatives explicitly.	
	Activated	Specifies that the first and second-order time derivatives of the user-defined trajectory are also user-defined. A new **Planar Motion Derivatives** sub-node is added to the simulation tree.
	De-activated	The first and second order time derivatives of the user-defined General Planar Motion trajectory are computed numerically.

Planar Motion Derivatives Properties

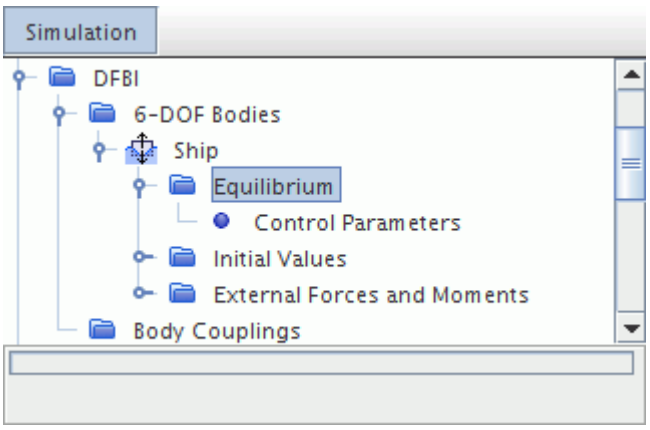
Velocity X	Velocity of the rigid body in X direction of the laboratory coordinate system.
Velocity Y	Velocity of the rigid body in the Y direction of the laboratory coordinate system.
Angular Velocity Z	Angular velocity of the rigid body around the Z axis with respect to the laboratory coordinate system.
Acceleration X	Acceleration of the rigid body in X direction of the laboratory coordinate system.
Acceleration Y	Acceleration of the rigid body in Y direction of the laboratory coordinate system.

Angular Acceleration Z	Angular acceleration of the rigid body around the Z axis of the laboratory coordinate system.

Equilibrium

You can use the equilibrium body motion option to help you quickly achieve a quasi steady-state equilibrium position of a rigid body that is subject to fluid forces or other external forces. A marine "sink and trim" simulation is an example of such a case.

Use this option to position and orient the rigid body in the flow so that the forces and moments acting on it are balanced in the chosen directions. The equilibrium body motion option has up to 6 degrees of freedom. The body moves within the specified directions according to a numerical iterative procedure until the forces and moments in these directions are balanced.



The numerical procedure to move the rigid body is purely mathematical, that is, there is no true physical motion of the rigid body. During the procedure, STAR-CCM+ moves the body in a stepwise manner, but does not solve an equation of motion. The body mass only plays a role for the gravity force which might need to be balanced to zero in the equilibrium state.

The numerical procedure for finding the equilibrium position operates iteratively:

1. At a time-step, the numerical procedure calculates a translation and/or rotation step size.
2. The rigid body moves (translates and/or rotates) by the calculated translation and/or rotation step size as determined by the numerical procedure. The flow must then adapt to this change in position.
3. In this new position, *inner* time-steps are performed that calculate the forces and moments until STAR-CCM+ reaches either the *Force and Moment Tolerance* or the *Maximum Inner Time-Steps*. (Inner time-steps are regular time-steps. They are called inner time-steps because they apply between incremental movements of the body.) See Equilibrium Properties

   At this point, STAR-CCM+ triggers any DFBI Equilibrium Forces Converged events that are associated with the body. These events in turn cause an update to any plots that depend on them.
4. The rigid body moves again by a certain distance as determined by the numerical procedure.

This process is repeated until the total of forces and the total of moments in the specified directions become zero. To reach good convergence of the procedure, forces and moments must converge well during the inner

time-steps. You can influence the convergence behavior by changing the control parameters of the equilibrium body motion option.

In order for you to plot results such as forces and moments over time, STAR-CCM+ provides the DFBI Equilibrium Forces Converged event. This update event only triggers when forces and moments are converged within a motion step. Plots that update in response to this event should show a smooth profile throughout the solution time history. For more information, see DFBI Equilibrium Forces Event.

You are advised to use the equilibrium body motion option only if the following requirements are fulfilled:

- For your DFBI simulation, you aim to reach an equilibrium position of the rigid body. The simulation itself can be dynamic and time-dependent, but your objective is to obtain a quasi steady-state solution.
- The forces and moments that act on the body depend on spatial position. This dependency is essential for the numerical procedure to work properly. A marine "sink and trim" simulation is an example of such a case.
- User-defined forces and moments acting on the body do not explicitly depend on time. For example, when modeling springs, forces can implicitly depend on time through position, but not explicitly by defining a direct time-dependency.

The equilibrium body motion option is designed for use in marine "sink and trim" simulations. You can also apply the option to other simulation scenarios as long as the above requirements are met. The numerical method works best for displacement hulls; the method is not recommended for planing hulls. Because of the dependency on spatial position, the numerical method works well for simulations involving moorings or bodies that are connected through springs.

Since you aim to reach a quasi steady-state solution at the end of the simulation, you can increase the time-step size without losing robustness in comparison to other DFBI simulations. You are advised to increase the time-step size to reduce the overall runtime of the simulation.

Contents:

Equilibrium Properties
DFBI Equilibrium Forces Converged Event

Equilibrium Properties

X Motion	For each chosen direction, the numerical procedure translates the body in the direction to achieve an equilibrium position. Directions are set with respect to the laboratory coordinate system.
Y Motion	
Z Motion	
X Rotation	For each chosen axis, the numerical procedure rotates the body around the axis to achieve an equilibrium position. Axis directions are set with respect to the laboratory coordinate system.
Y Rotation	
Z Rotation	

Control Parameters Properties

Maximum Translation Step	Maximum translation step size during one time-step.

	• If the algorithm calculates a translation step size that is smaller than the specified *Maximum Translation Step*, the calculated translation step size is applied to move the body. • If the algorithm calculates a translation step size that is greater than the specified *Maximum Translation Step*, the calculated translation step size is split over a number of consecutive time-steps. In each of these time-steps, at most the *Maximum Translation Step* is applied. This is done until the entire predicted translation step size is reached.
Maximum Rotation Step	Maximum rotation step size during one time-step. • If the algorithm calculates a rotation step size smaller than the specified *Maximum Rotation Step*, the calculated rotation step size is applied to rotate the body. • If the algorithm calculates a rotation step size greater than the specified *Maximum Rotation Step*, the calculated rotation step size is split over a number of consecutive time-steps. In each of these time-steps, at most the *Maximum Rotation Step* is applied. This is done until the entire predicted rotation step size is reached.
Expert Properties	
URF Translation	Under-relaxation factor for the translational motion of the rigid body to achieve equilibrium. Specifies which portion of the internally calculated translation step size is actually applied to the body. Reduce this factor if you encounter convergence problems.
URF Rotation	Under-relaxation factor for the rotational motion of the rigid body to achieve equilibrium. Specifies which portion of the internally calculated rotation step size is actually applied to the body. Reduce this factor if you encounter convergence problems.
Maximum Translation Growth Rate	To increase stability of the simulation, the translation step size of the current motion step is limited by the translation step size of the previous motion step times the *Maximum Translation Growth Rate*. Therefore, the translation step size cannot increase arbitrarily from one motion step to the next, but its growth is limited by the maximum growth rate factor. Reduce this value if you encounter convergence problems.
Maximum Rotation Growth Rate	To increase stability of the simulation, the rotation step size of the current motion step is limited by the rotation step size of the previous motion step times the *Maximum Rotation Growth Rate*. Therefore, the rotation step size cannot increase arbitrarily from one motion step to the next, but its growth is limited by the maximum growth rate factor. Reduce this value if you encounter convergence problems.

Force and Moment Tolerance	Convergence criterion for the forces and moments. Determines how much the forces and moments can vary before the next motion step is performed.
Maximum Inner Time-Steps	Maximum number of time-steps between two motion steps. • If the *Force and Moment Tolerance* convergence criterion is met during a time-step number that is lower than the *Maximum Inner Time-Steps*, the next motion step is performed. • If the *Force and Moment Tolerance* convergence criterion is not met during a time-step number that is lower than the *Maximum Inner Time-Steps*, a motion step is performed after the *Maximum Inner Time-Steps* are reached.

DFBI Equilibrium Forces Converged Event

The **DFBI Equilibrium Forces Converged** event is available only when a 6-DOF body is associated with **Equilibrium** motion. The update event is created under **Tools** > **Events**. This event enables you to monitor the converged force or moment values that act on a rigid body. The event triggers an update when the forces and moments are converged after a motion step. Typically, a new motion step is performed afterwards.

DFBI Equilibrium Forces Converged Properties

Enabled	Specifies whether the update event is active or not.
Satisfied	Indicates whether the update event has been satisfied. This property is read only.
Iteration	Displays the iteration at which the event was last evaluated. This property is read only.
Body	Selects the rigid body that has the **Equilibrium** motion option set.

See Also:

Equilibrium

Setting up Multi-Body Motion

Multi-body motion allows you to simulate a system of two or more rigid bodies that are connected through mechanical joints. The entire system can move and the rigid bodies can move relative to each other. The joints impose kinematic constraints on the motions of the bodies which limit the degrees of freedom of the entire multi-body system.

Consider the following:

- You can create mechanical joints between rigid bodies for which multi-body motion is activated.
- Coupling of a rigid body to the environment using mechanical joints is also possible.

- You can connect bodies with free motion and multi-body motion using couplings such as springs, catenaries, and contacts.
- You can connect bodies that are coupled using mechanical joints with other couplings such as springs, catenaries, and contacts.
- When coupling several rigid bodies using joints, the creation of closed loops of bodies is currently not supported. For example, you couple Body 1 to Body 2, then Body 2 to Body 3. Do not couple Body 3 back to Body 1 as this coupling creates a closed loop.
- You can create tree-like structures of bodies that are connected through mechanical joints.

Before you begin, set up two or more 6-DOF bodies to couple as a multi-body motion.

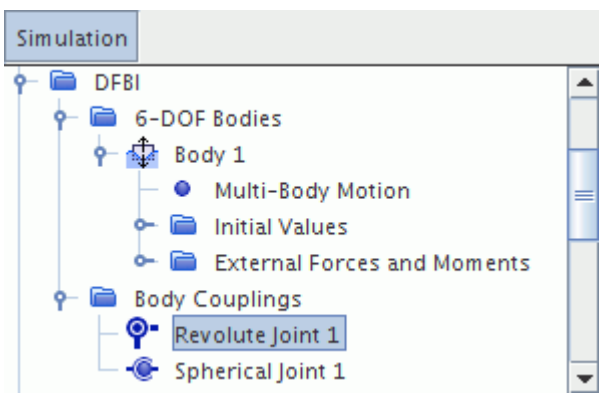
To set up multi-body motion:

1. Navigate to the **DFBI** > **6-DOF Bodies** node.
2. Multi-edit the **[Body 1]** and **[Body 2]** nodes and set *Body Motion Option* to **Multi-Body Motion**.

   The **Multi-Body Motion** node appears in the simulation tree.
3. Edit the **Multi-Body Motion** node and set its Multi-Body Motion Properties.

In addition to activating multi-body motion for rigid bodies, you define a spherical joint or a revolute joint coupling:

4. Right-click the **6-DOF Bodies** > **Body Couplings** node and create one of the following:
   - **Spherical Joint**
   - **Revolute Joint**



5. Depending on the type of coupling that you create, select the appropriate sub-node and set its properties.
6. Right-click the **Reports** node and select **New Report** > **6-DOF Constraint Deviation**.

   For more information, see 6-DOF Constraint Deviation Report.
7. Run the multi-body motion simulation.

   If the constraint deviation that you monitor is too large, see Multi-Body Motion Guidelines

Contents:

Multi-Body Motion Guidelines
Multi-Body Motion Properties

See Also:

Spherical Joint Coupling
Revolute Joint Coupling
Governing Equations for Multi-Body Motion
6-DOF Constraint Deviation
Constraint Force
DFBI_JointForceNode

Multi-Body Motion Guidelines

If you encounter large constraint deviations for the multi-body simulation, you can improve the simulation setup.

- Reduce the time-step size: Select the **Solvers** > **Implicit Unsteady** node and set *Time-Step* to an appropriate value.
- Alternatively, or in addition to reducing the time-step size, activate **6-DOF Solver** > *Sub-Stepping Enabled*.
- Select the **Solvers** > **6-DOF Solver** > **Baumgarte Constraint Stabilization** node and adjust the Baumgarte Constraint Stabilization properties.

Multi-Body Motion Properties

X Motion, Y Motion, Z Motion

When **On**, translation motion is calculated along the X-, Y-, Z-axis of the selected *Coordinate System*. When **Off**, the degree of freedom in the selected direction is frozen.

X Rotation, Y Rotation, Z Rotation

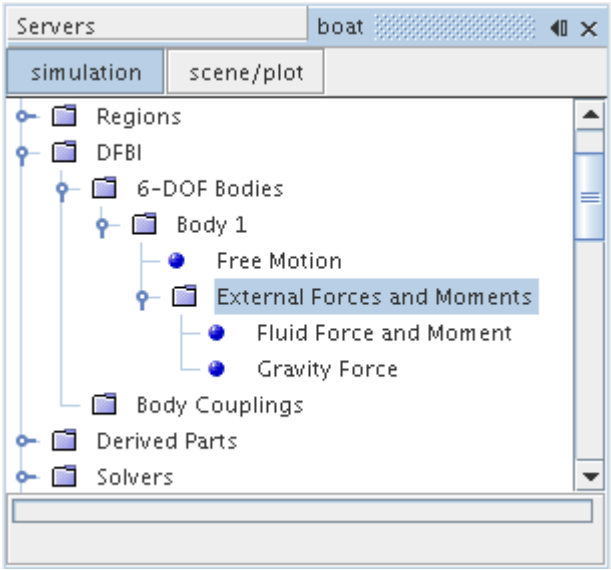
When **On**, rotation around the X-, Y-, Z-axis is calculated with respect to the selected *Coordinate System*. When **Off**, the degree of freedom in the selected direction is frozen.

Coordinate System

Coordinate system in which *X Motion, Y Motion, Z Motion* and *X Rotation, Y Rotation, Z Rotation* are specified.

What Are External Forces and Moments?

The **External Forces and Moments** node, manages all forces and moments that act on the 6-DOF body. These forces and moments include the force due to gravity, and the pressure and shear forces from the fluid.



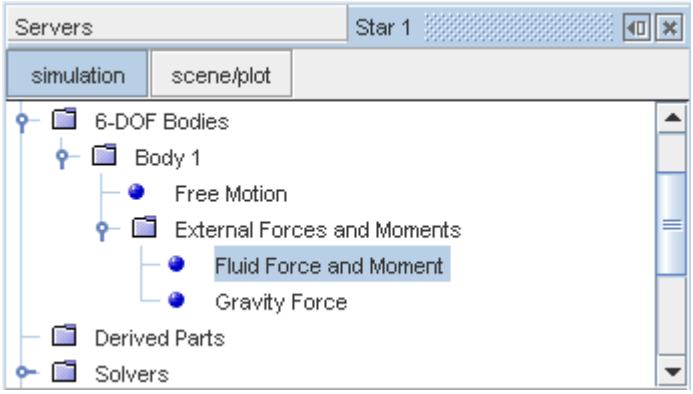
The Fluid Force and Moment and Gravity Force nodes are present by default. Additional forces and moments can be added to these using the manager node.

Contents:

Fluid Force and Moment
Gravity Force
Adding an External Force or Moment
Pop-Up Menu of the External Forces and Moments Manager
External Force Acting on the Center of Mass
External Moment
External Force
External Propulsion Force
Torsional Spring Moment
Damping Moment
Damping Force
Catenary Force
Linear Spring Force
Joint Force
Constraint Force

Fluid Force and Moment

The **Fluid Force and Moment** node is added by default to the **External Forces and Moments** node. This node represents the forces and moments that act on the 6-DOF body due to the fluid surrounding it.



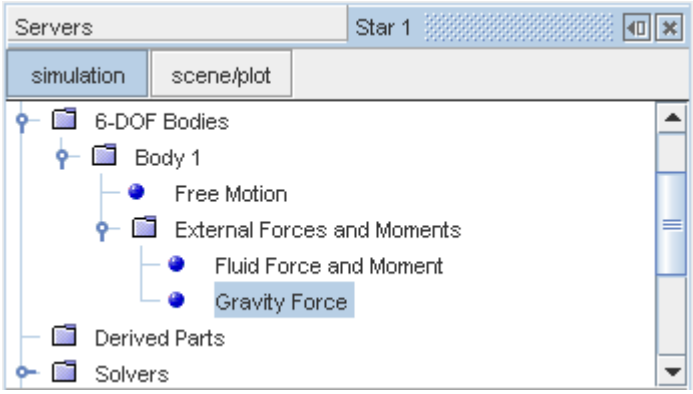
An option is provided to smooth the fluid forces and moments before they are applied to the body. A moving weighted average is used as a low pass filter; you set the number of time steps that should be included in this moving average.

Properties

Enable Smoothing	This option causes fluid forces and moments to be averaged over a specified time interval before they are applied to the body.	
	Activated	Forces and moments are smoothed.
	Deactivated	Forces and moments are applied directly.
Number of smoothing time steps	Sets the number of time steps that are used in the moving average filter.	
Use Static Pressure	Controls the type of pressure to use.	
	Activated	Switches to the use of static pressure.
	Deactivated	Leaves the choice at the default working pressure.
Enabled	Enables the action of fluid forces and moments on the 6-DOF body.	
	Activated	Fluid forces and moments are applied.
	Deactivated	Fluid forces and moments are ignored.

Gravity Force

The **Gravity Force** node is added by default to the External Forces and Moments node. This node represents the force due to gravity that acts on the 6-DOF body.



Properties

Enabled	Enables the action of gravity on the 6-DOF body. The gravity model also needs to be selected in the physics continuum—otherwise enabling this option has no effect.	
	Activated	Gravity force acts on the 6-DOF body.
	Deactivated	Gravity force is ignored.

Adding an External Force or Moment

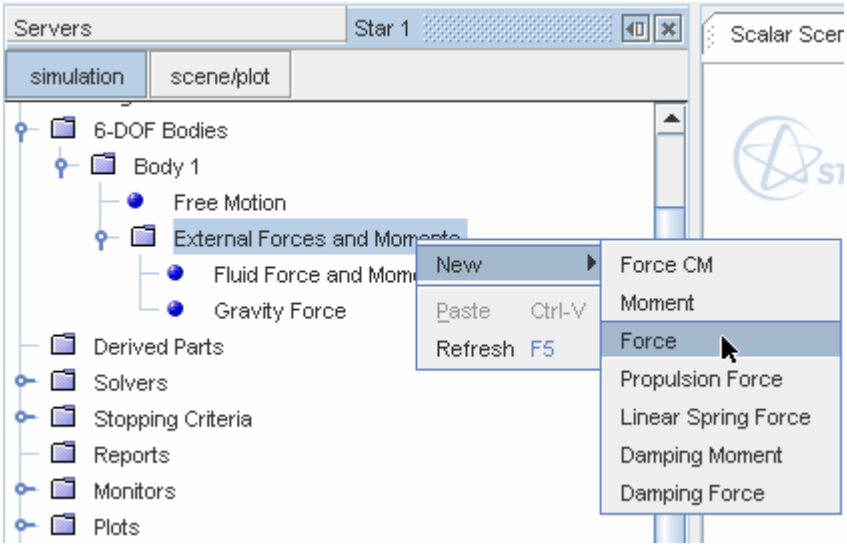
There are several types of external influence that can be added to the 6-DOF body using this manager node:

- External Force Acting on the Center of Mass
- External Moment
- External Force
- External Propulsion Force
- Torsional Spring Moment
- Damping_Moment
- Damping_Force

There are also two forces that are added in response to body couplings created using the Body Couplings manager node:

- Catenary Force
- Linear Spring Force

To add an external force or moment, right-click the **External Forces and Moments** node and select **New** followed by the particular type of action to create:



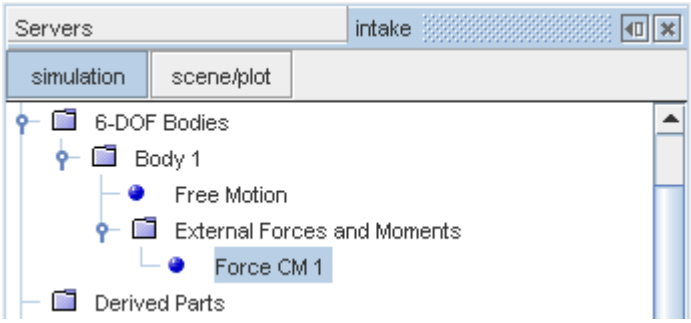
The appropriate sub-node for the external force or moment is added to the manager. These subnodes have their own properties that should be set in the *Properties* window.

Pop-Up Menu of the External Forces and Moments Manager

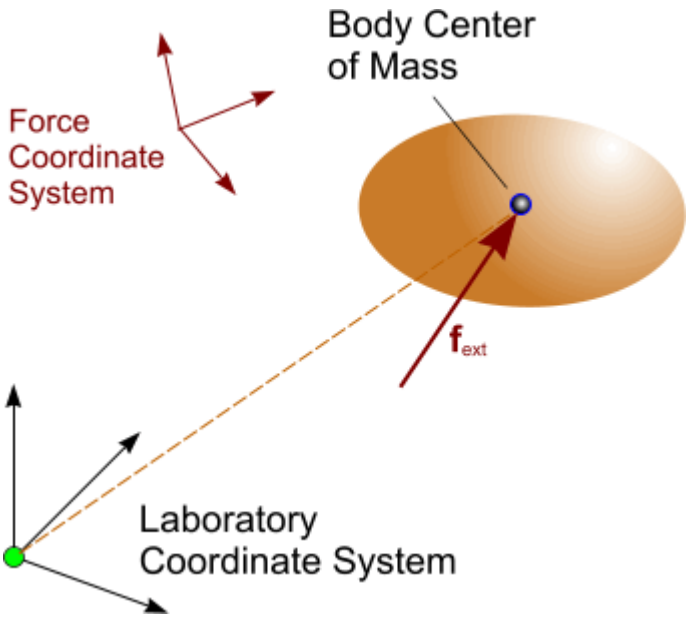
New	Adds a force or moment of the selected type.	
	Force CM	Adds a Force CM node.
	Moment	Adds a Moment node.
	Force	Adds a Force node.
	Propulsion Force	Adds a Propulsion Force node.
	Damping Moment	Adds a Damping_Moment node.
	Damping Force	Adds a Damping_Force node.

External Force Acting on the Center of Mass

This node is added when you choose **New** > **Force CM** from the **External Forces and Moments** manager pop-up menu. It represents a user-defined force acting on the 6-DOF body at its center of mass.



The application of this force is illustrated in the diagram below. It does not generate any moment about the center of mass.



Force CM Properties

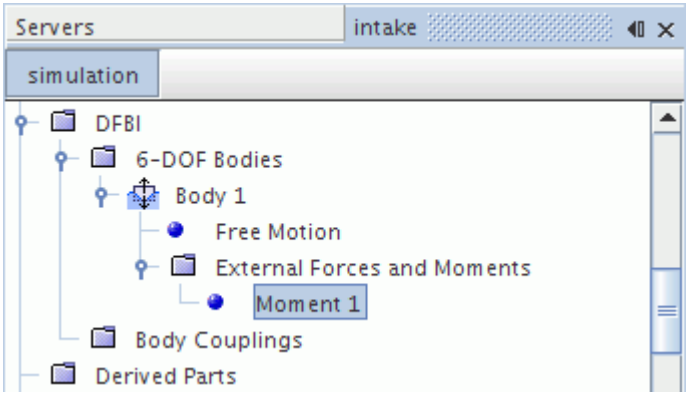
Force	The external force entered as a vector quantity, along with the appropriate unit of force.	
Enabled	Enables the action of the force on the 6-DOF body.	
	Activated	The force is active.
	Deactivated	The force is not active.

Force CM Expert Properties

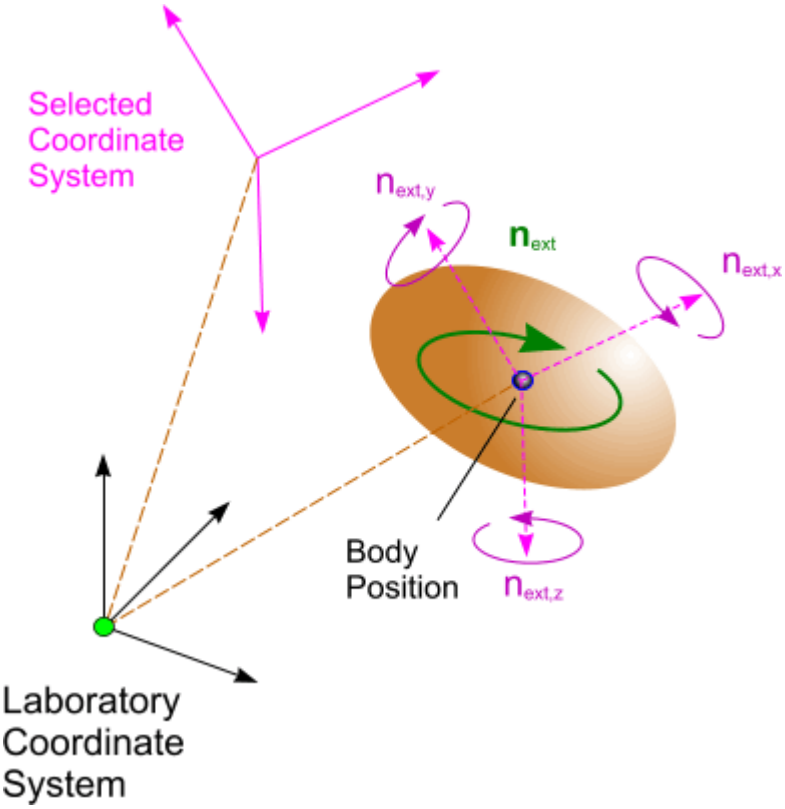
Coordinate System	The coordinate system in which the vector of the force is specified, chosen from the drop-down list.

External Moment

This node represents an external moment acting on the 6-DOF body with respect to the body position. It is added when you select **New** > **Moment** from the **External Forces and Moments** manager pop-up menu.



The moments are specified around each axis of the selected coordinate system. However, they act with respect to the position of the body, as illustrated in the following diagram:



Moment Properties

Moment	Components of the external moment entered as a vector quantity, along with the appropriate unit of moment.	
Enabled	Enables the action of the moment on the 6-DOF body.	
	Activated	The moment is active.
	Deactivated	The moment is not active.

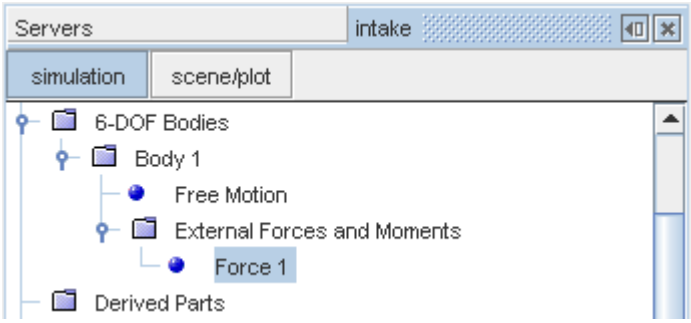
Moment Expert Properties

Coordinate System	The coordinate system in which the vector of the moment is specified, chosen from the drop-down list.

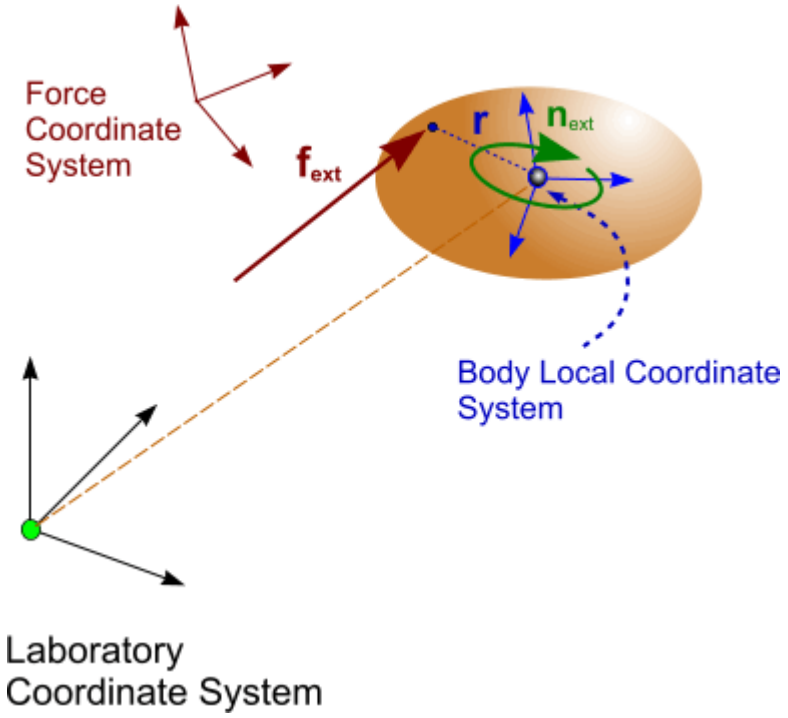
External Force

This node represents a general external force acting on the 6-DOF body at a specified location.

It is added when you select **New** > **Force** from the **External Forces and Moments** manager pop-up menu.



This force results in a moment acting around the body center of mass, as shown in the following diagram:



External Force Properties

Force	The external force entered as a vector quantity, along with the appropriate unit of force.	
Position	The point at which the force acts on the 6-DOF body.	
Enabled	Enables the action of the force on the 6-DOF body.	
	Ticked	The force is active.
	Cleared	The force is not active.

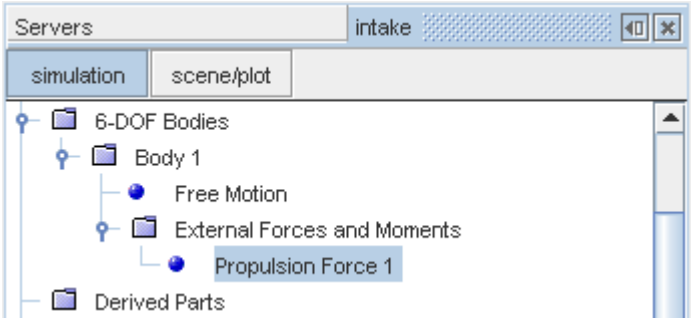
External Force Expert Properties

Coordinate System1	The coordinate system in which the *Position* coordinates are specified, chosen from the drop-down list.
Coordinate System 2	The coordinate system in which the *Force* vector is specified, chosen from the drop-down list.

External Propulsion Force

This node represents an external force acting on the body, in the same way as the External Force. The difference is that the external propulsion force is specified using a force magnitude and direction rather than specifying the force components directly.

This node is added when you select **New** > **Propulsion Force** from the **External Forces and Moments** manager pop-up menu



Propulsion Force Properties

Position	The point at which the force acts on the 6-DOF body.	
Thrust	The magnitude of the force as a constant scalar value with appropriate unit of force.	
Direction	Direction in which the force acts. Given as a vector.	
Enabled	Enables the action of the force on the 6-DOF body.	
	Activated	The force is active.
	Deactivated	The force is not active.

Propulsion Force Expert Properties

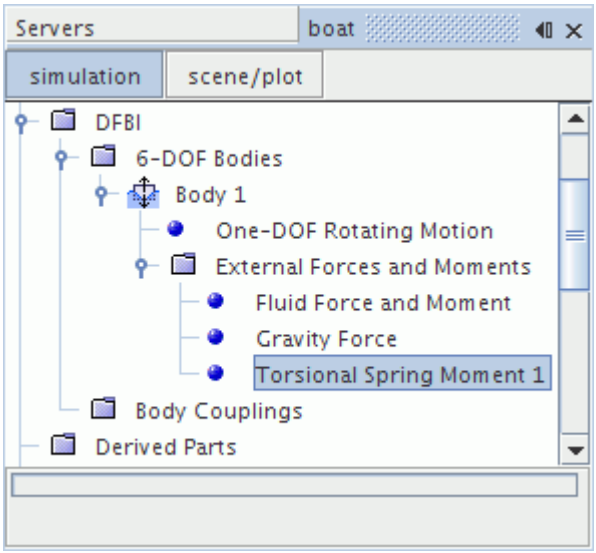
Coordinate System1	The coordinate system in which the *Position* coordinates are specified, chosen from the drop-down list.
Coordinate System2	The coordinate system in which the *Direction* vector is specified, chosen from the drop-down list.

Torsional Spring Moment

The **Torsional Spring Moment** node represents the effect of a torsional spring acting on a rotating body. The torsional moment always acts to rotate the body into the angular position that is given by the relaxation angle.

The magnitude of this moment is proportional to the difference between the current angular position of the body and the relaxation angle (as shown in Eqn. (983)).

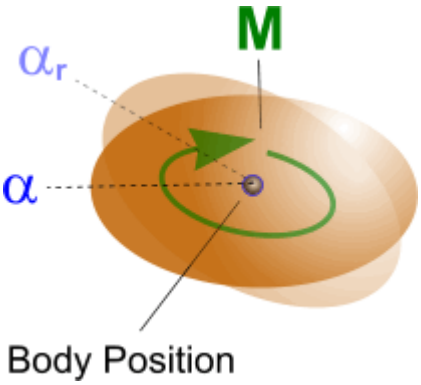
The **Torsional Spring Moment** node is added when you select **New** > **Torsional Spring Moment** from the **External Forces and Moments** manager pop-up menu.



The torsional spring moment can currently be added to bodies with one rotational degree of freedom only. Such bodies include:

- One-DOF Rotating bodies, and
- Bodies undergoing Free Motion with only one rotational degree of freedom selected (and all other degrees of freedom cleared).

The action of the torsion spring moment is illustrated in the following diagram.



The axis of rotation depends on the selected motion type. For a Free Motion, the axis of rotation is the axis of the laboratory coordinate system around which the body is allowed to rotate. For the One-DOF Rotating Motion, you specify the axis of rotation in the node of this motion type.

Torsional Spring Moment Properties

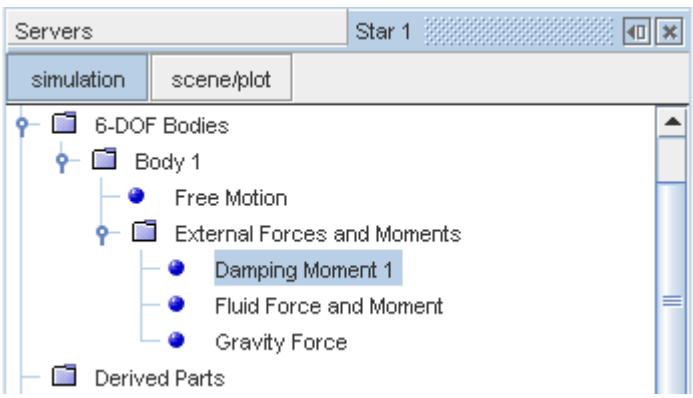
Spring Constant	The spring constant, k in Eqn. (983), of the torsional spring.
Relaxation Angle	The angle, α_r in Eqn. (983), at which the spring moment becomes zero. The angle is measured with respect to the initial orientation of the body. The initial angle is 0.

Enabled	Enables the effect of the torsional spring moment on the one-DOF rotating body.	
	Activated	The torsional spring moment is active.
	Deactivated	The torsional spring moment is not active.

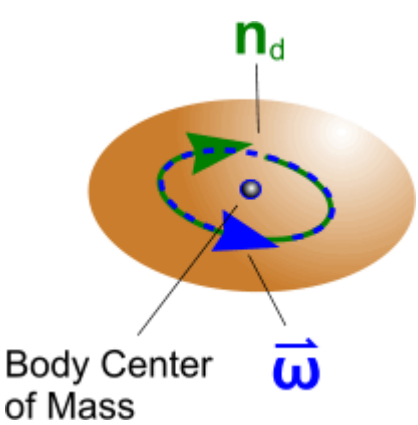
Damping Moment

The **Damping Moment** node represents the effect of a moment acting around the center of mass of the body in opposition to the angular velocity of the body.

The magnitude of the moment, which always acts to decelerate the body, is proportional to the angular velocity (as shown in Eqn. (984)). This node is added when you select **New** > **Damping Moment** from the **External Forces and Moments** manager pop-up menu.



The action of the damping moment is illustrated in the following diagram.



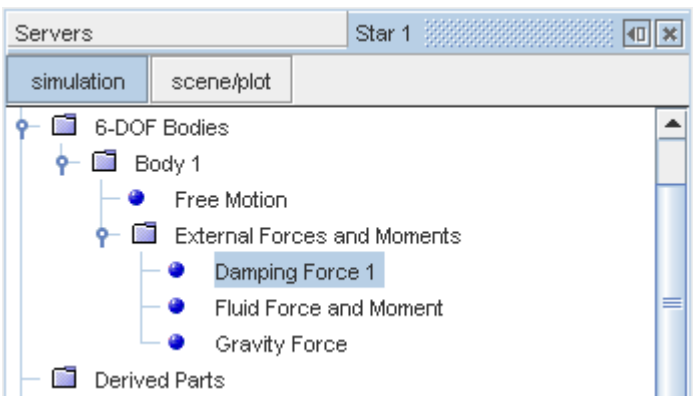
Damping Moment Properties

Damping Constant	Constant $a_{d,n}$ in Eqn. (984). See note on use of damping constants .	
Enabled	Enables the effect of the damping moment on the 6-DOF body.	
	Activated	The moment is active.
	Deactivated	The moment is not active.

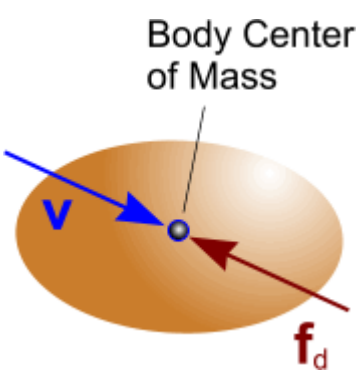
Damping Force

The **Damping Force** node represents a force acting on the center of mass of the body in the opposite direction to the body's velocity vector. It therefore always acts to decelerate the body.

A linear damping function, that is given by Eqn. (985), is used to define this force. The force is added when you select **New** > **Damping Force** from the **External Forces and Moments** manager pop-up menu.



The following diagram illustrates the action of the damping force.



Damping Force Properties

Damping Constant	Constant $a_{d,f}$ in Eqn. (985). See note on use of damping constants .	

Enabled	Enables the effect of the damping force on the 6-DOF body.	
	Activated	The force is active.
	Deactivated	The force is not active.

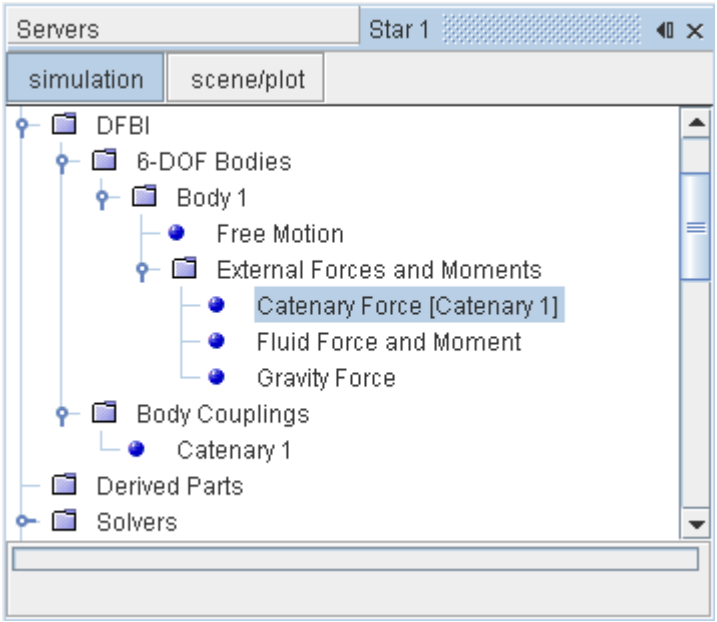
Use of Damping Constants

Choosing too large a value for the damping constant might lead to numerical instability of the 6-DOF Body motion. If numerical instability is encountered, reduce either the damping constant or the time step.

Catenary Force

This force is automatically created when a catenary coupling is created from the **Body Couplings** manager node. The name of the coupling is shown in square brackets next to the name of the catenary force.

The catenary force cannot be created, deleted, copied, or renamed within the **External Forces and Moments** manager node. It is managed entirely by its corresponding catenary coupling.



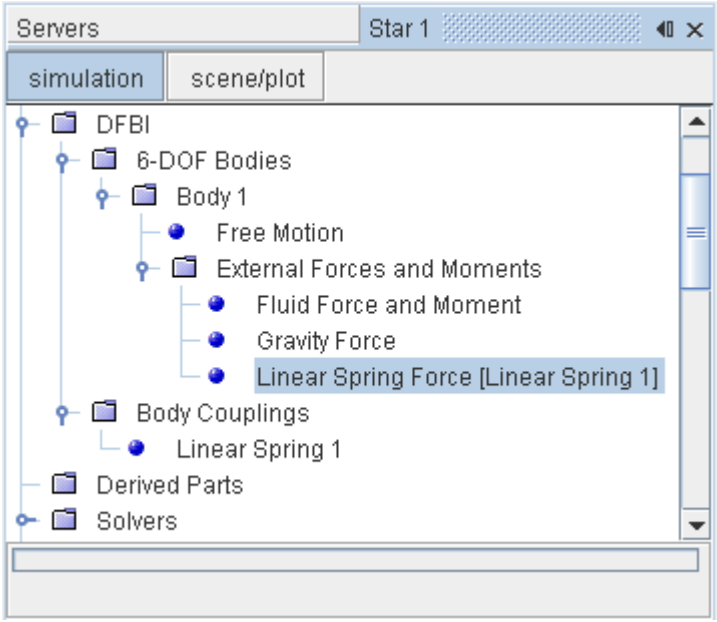
Catenary Force Properties

Coupling Element	The name of the corresponding catenary coupling element in the **Body Couplings** manager node (read-only).

Linear Spring Force

The linear spring force node is automatically created when a linear spring coupling is created from the **Body Couplings** manager node.

The name of the associated coupling is shown in square brackets next to the name of the linear spring force node. Unlike earlier versions of STAR-CCM+, the linear spring force cannot be created, deleted, copied, or renamed within the **External Forces and Moments** manager. It is managed entirely by its corresponding linear spring coupling.



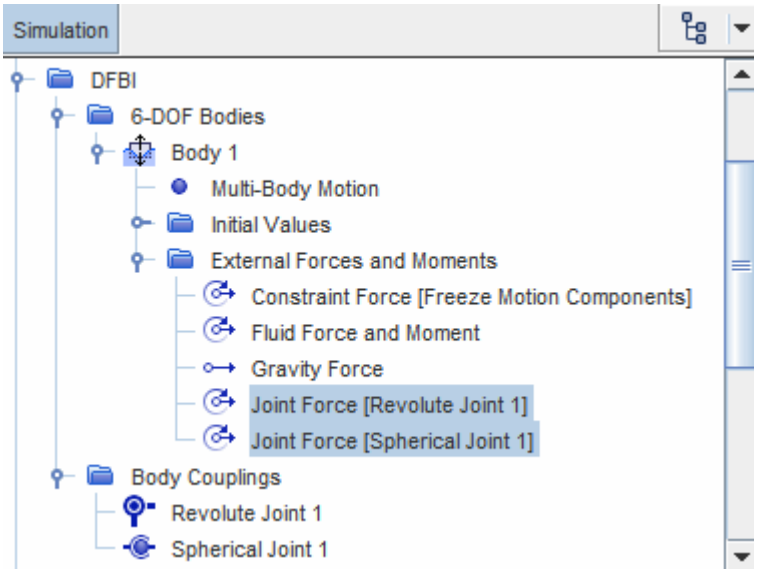
Linear Spring Force Properties

Coupling Element	The name of the corresponding linear spring coupling element in the **Body Couplings** manager node (read-only).

Joint Force

The joint force node is automatically created under the **External Forces and Moments** node when a spherical joint or a revolute joint coupling is created from the **Body Couplings** manager node.

The name of the associated coupling is shown in square brackets next to the name of the joint force node.



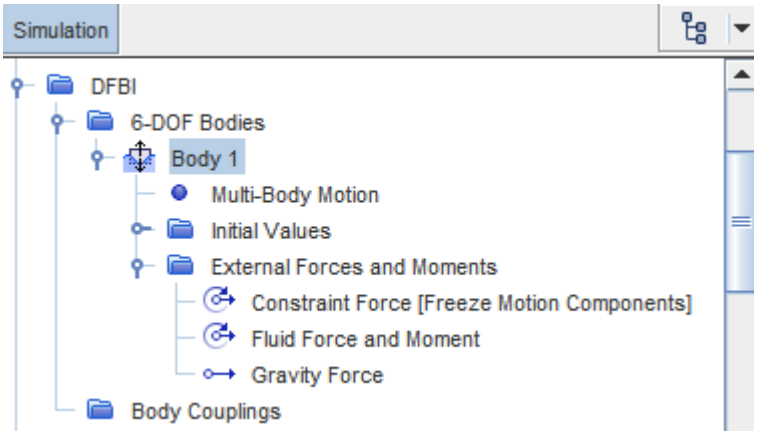
Linear Spring Force Properties

Coupling Element	The name of the corresponding joint coupling element in the **Body Couplings** manager node (read-only).

Constraint Force

The **Constraint Force [Freeze Motion Components]** node appears automatically when you activate multi-body motion.

This node represents the additional forces that STAR-CCM+ calculates to prevent motion of these degrees of freedom that you deactivate under Multi-Body Motion Properties. You can report this force by using a **6-DOF Body Force** or a **6-DOF Body Moment** report.



See Also:

Setting up Multi-Body Motion

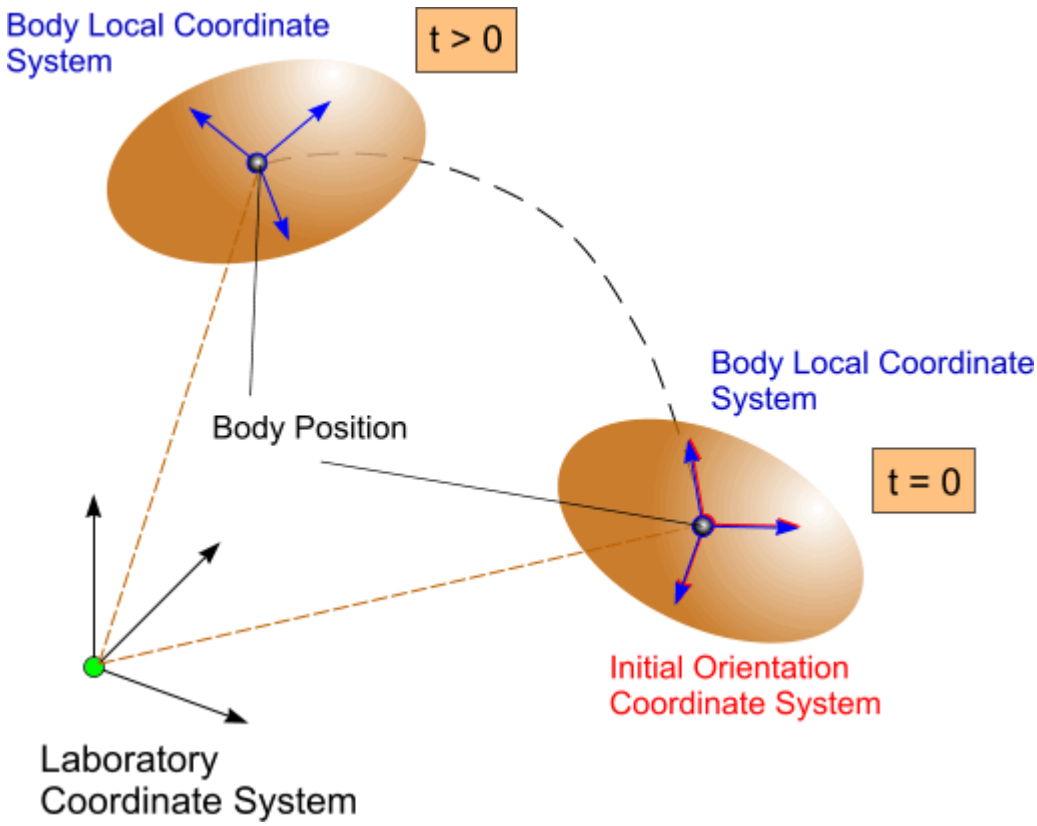
6-DOF Coordinate Systems

STAR-CCM+ automatically creates a new Cartesian coordinate system with the default name **Body n-CSys** when a new 6-DOF body with the name **Body n** is created. This coordinate system (which can be referred to as the Body Local Coordinate System) shows the current position and orientation of the 6-DOF body throughout the simulation.

The Initial Orientation Coordinate System is used to define the initial orientation of the 6-DOF Body Local Coordinate System.

The origin of the Initial Orientation Coordinate System need not necessarily coincide with the initial position of the 6-DOF body. The Initial Orientation Coordinate System can be set to any appropriate coordinate system defining the initial orientation of the body, including the Laboratory system. Upon initialization, the origin of the Body Local Coordinate System is automatically placed at the initial position of the body.

These coordinate systems are illustrated in the diagram below.



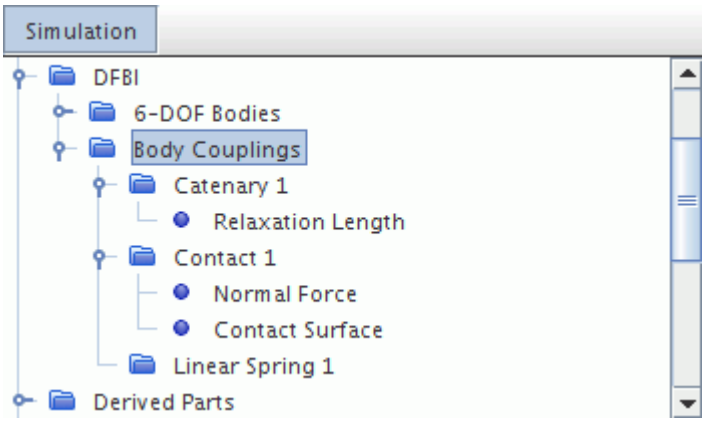
The rigid body motion is decomposed into a translation of a point, and a rotation about this point. The body position is the position of this point. The position of the rigid body depends on the selected body motion type:

Body Motion Type	Body Position
Free Motion	Body center of mass
Equilibrium	
One-DOF Translating Motion	

Body Motion Type	Body Position
Four-DOF Maneuvering Motion	
Multi-Body Motion	
One-DOF Rotating Motion	Origin of the rotation axis
Planar Motion Carriage	Attachment point

Working with 6-DOF Body Couplings

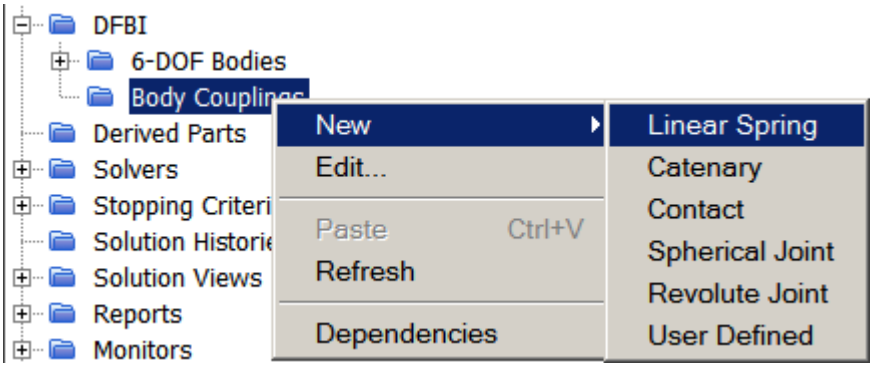
The **Body Couplings** manager node is used to create and contain pre-defined types of body-to-body and body-to-environment coupling elements.



You can create the following types of coupling:

- Linear Spring
- Catenary
- Contact
- Spherical Joint
- Revolute Joint
- User Defined

You create these couplings by right-clicking the **Body Couplings** manager node and choosing **New** followed by the desired type.



Contents:

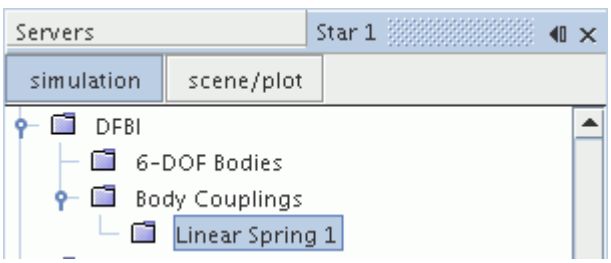
What Is a Linear Spring Coupling?

What Is a Catenary Coupling?
Setting Up Contact Couplings
Spherical Joint Coupling
Revolute Joint Coupling
Setting up User-Defined Couplings
Body Couplings Field Functions

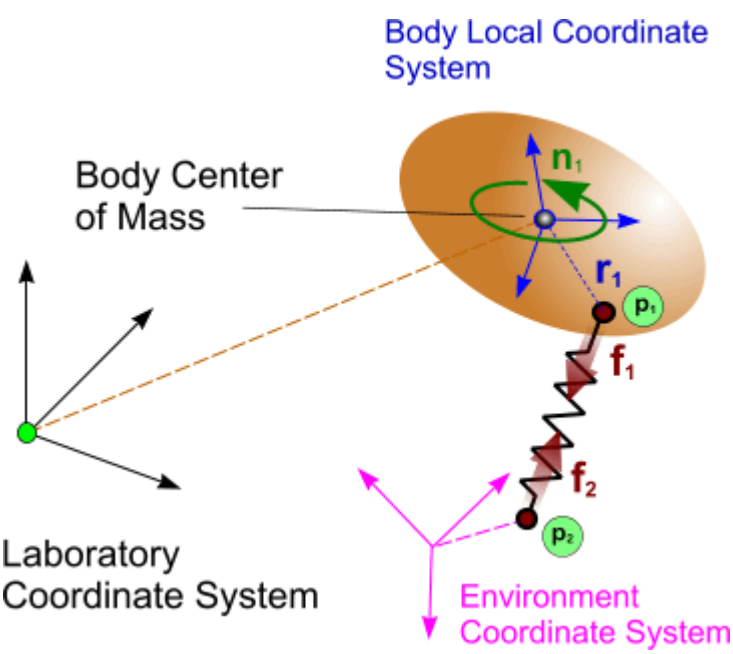
What Is a Linear Spring Coupling?

This coupling element models a linear (that is, non-torsional) spring between two bodies or one body and the environment. It carries the main functionality of springs in DFBI.

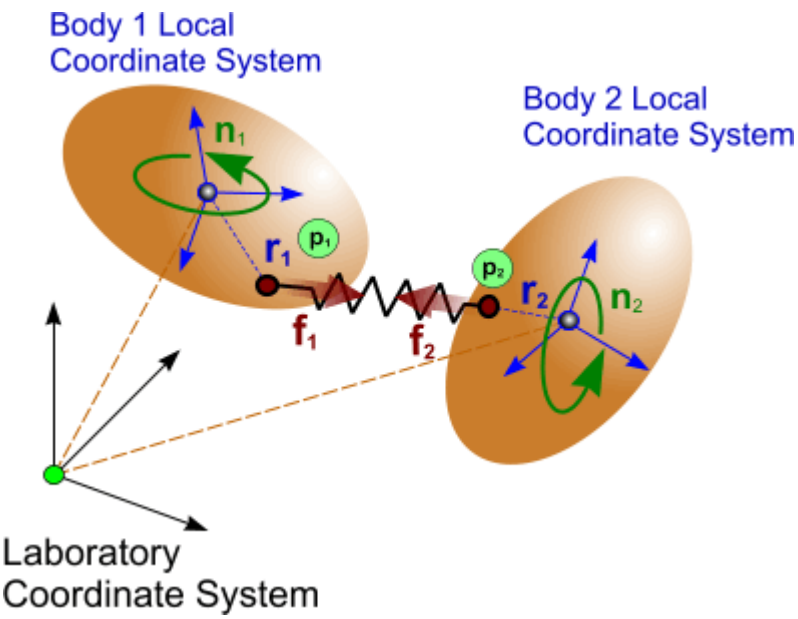
The **Linear Spring** node, which has properties and is subject to common menu items, represents this linear spring. It is added when you select **New** > **Linear Spring** from the **Body Couplings** manager pop-up menu.



The action of the linear spring coupling between a body and a point in the environment is illustrated in the following diagram; the formula is given in the formulation section.



A linear spring coupling between two bodies is also permitted; this system is depicted below.



Linear Spring Coupling Properties

Relaxation Length	Constant scalar value l_{eq} in Eqn. (988). No spring force is exerted when the distance between *Position 1* and *Position 2* is equal to this length.	
Spring Constant	Constant scalar value k_s in Eqn. (986). This property relates the distance by which the spring is stretched or compressed to the resulting force.	
Enabled	Sets whether the coupling is active or not.	
	Activated	The coupling is accounted for in the solution procedure.
	Deactivated	The coupling is not considered.
Object 1	Sets the type of anchor to which the first end of the spring is attached. This should be different from *Object 2*.	
	Environment	Attaches the first end point to a location in the environment that is defined by *Position 1*.
	<Body>	Attaches the first end point to the 6-DOF Body selected from this list. *Position 1* defines the location to which the spring is attached.
Object 2	Sets the type of anchor to which the second end of the spring is attached. This should be different from *Object 1*.	

	Environment	Attaches the second end point to a location in the environment that is defined by *Position 2*.
	<Body>	Attaches the second end point to the 6-DOF Body selected from this list. *Position 2* defines the location on the body to which the spring is attached.
Position 1	The position vector of the point at which the spring is attached to *Object 1*, specified in *Coordinate System 1*.	
Position 2	The position vector of the point at which the spring is attached to *Object 2*, specified in *Coordinate System 2*.	
Index	(Read-only) Unique number to identify the coupling.	

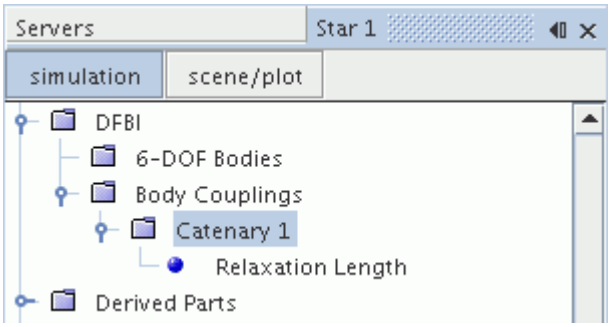
Linear Spring Coupling Expert Properties

Coordinate System 1	The coordinate system in which *Position 1* is specified, chosen from the drop-down list. Only coordinate systems that are related to *Object 1* can be chosen.	
Coordinate System 2	The coordinate system in which *Position 2* is specified, chosen from the drop-down list. Only coordinate systems that are related to *Object 2* can be chosen.	
No Repelling Force	Determines if repelling forces between *Position 1* and *Position 2* are considered. If the force would be repelling, and this option is ticked, the force is set to zero.	
	Activated	Only attracting forces between *Position 1* and *Position 2* are considered.
	Deactivated	Both attracting and repelling forces are considered.

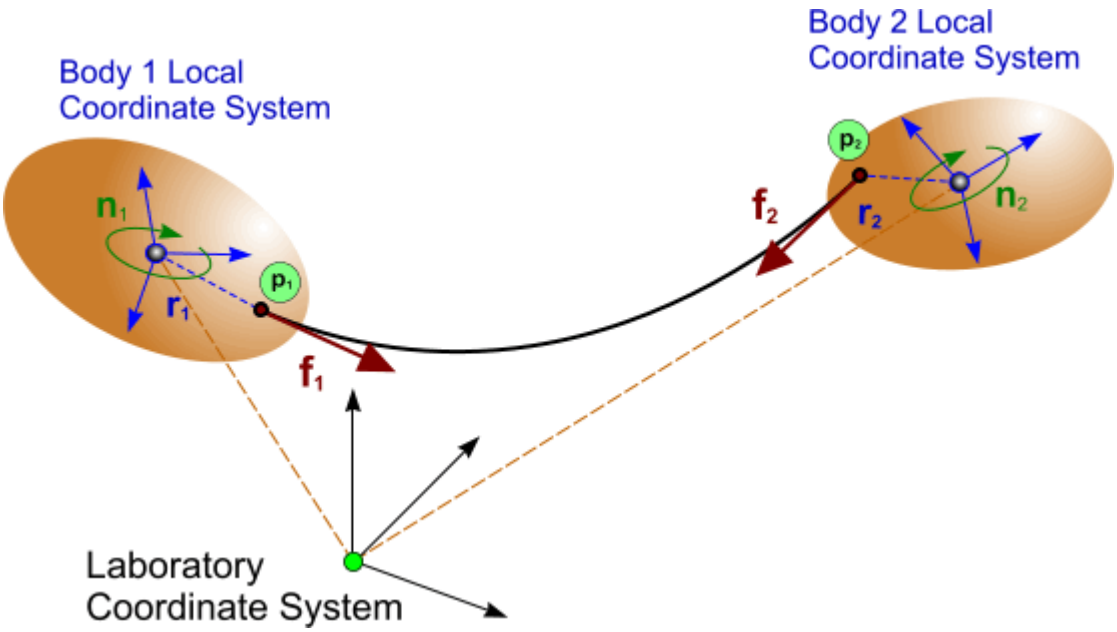
What Is a Catenary Coupling?

This coupling element, which a catenary coupling node represent, models a quasi-stationary, elastic catenary between a pair of bodies, or between one body and the environment.

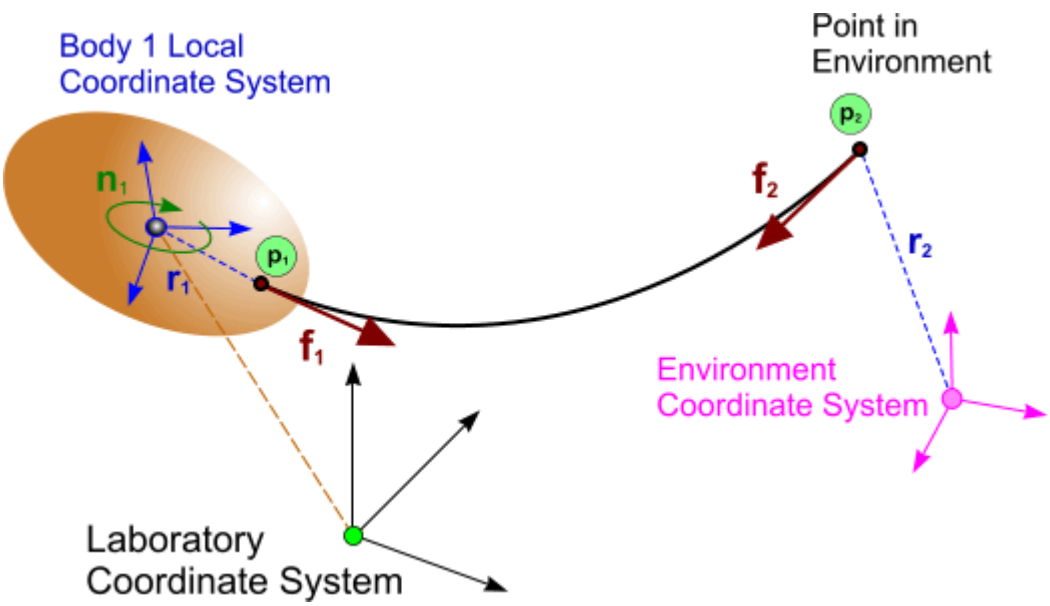
The formulation provides the underlying equations.



Examples of where this element could be used are in modeling a tugboat and its payload, or a moored vessel and its associated tether. Quasi-stationary means that the catenary has its steady state shape at each instant of time. This model is appropriate for body motions which are sufficiently slow compared to the wave velocity in the catenary. The catenary coupling between two bodies is depicted in the following diagram:



$\mathbf{f}_1$ and $\mathbf{f}_2$ are tangential to the catenary curve; the horizontal components are equal, but with opposite sign. For a coupling between a body and the environment, the system becomes as follows:



There is a known limitation of the catenary coupling method: the end points of the coupling must not become vertically aligned, as this results in a singularity in the underlying catenary equation. It is therefore necessary to ensure that the end points of the catenary do not coincide in the vertical direction during the simulation. Violation of this requirement results in an error message. In case of vertical alignment failure, the expert property *Alignment Failure Offset* can be adjusted to prevent such failures.

The catenary coupling facility has two methods for specifying the catenary relaxation length:

- Relaxation Length—The catenary relaxation length is user-defined.
- Pretension—The catenary relaxation length is calculated using a pretension force and estimated relaxation length.

Catenary Coupling Properties

Catenary Mode	Specifies the method to define the catenary mode.	
	Relaxation Length	Selects the Relaxation Length mode, which has its own properties.
	Pretension	Selects the Pretension mode, which has its own properties.
Mass Per Unit Length	The mass per unit length of the catenary without loading.	
Stiffness	The force per unit change of length of the elastic catenary.	
Enabled	Sets whether the coupling is active or not.	
	Activated	The coupling is accounted for in the solution procedure.
	Deactivated	The coupling is not considered.
Object 1	Sets the type of anchor to which the first end of the catenary is attached. This should be different from *Object 2*.	

	Environment	Attaches the first end point to a location in the environment as defined by *Position 1*.
	<Body>	Attaches the first end point to the 6-DOF Body selected from this list. *Position 1* defines the location to which the catenary is attached.
Object 2	Sets the type of anchor to which the second end of the catenary is attached. This should be different from *Object 1*.	
	Environment	Attaches the second end point to a location in the environment as defined by *Position 2*.
	<Body>	Attaches the second end point to the 6-DOF Body selected from this list. *Position 2* defines the location on the body to which the catenary is attached.
Position 1	Anchoring position of the first end point of the catenary on *Object 1*.	
Position 2	Anchoring position of the second end point of the catenary on *Object 2*.	
Index	(Read-only) Unique number to identify the coupling.	

Catenary Coupling Expert Properties

Coordinate System 1	The coordinate system in which *Position 1* is specified, chosen from the drop-down list. Only coordinate systems that are related to *Object 1* can be chosen.
Coordinate System 2	The coordinate system in which *Position 2* is specified, chosen from the drop-down list. Only coordinate systems that are related to *Object 2* can be chosen.
Alignment Failure Offset	Parameter that is used to prevent vertical alignment failure of the catenary. Whenever the horizontal distance between the two catenary end points gets smaller than this value, one of the end points is shifted by a small horizontal offset so that the total horizontal distance is equal to the Alignment Failure Offset. It is a relative value that is measured against the relaxation length of the catenary.

What Is the Relaxation Length Catenary Mode?

This mode assumes that you specify the catenary relaxation length explicitly.

Relaxation Length Mode Properties

Relaxation Length	Specifies the catenary length without loading.

What Is the Pretension Catenary Mode?

When using the pretension catenary mode, the catenary relaxation length is calculated so that the pretension force at position 1 equals the user-defined pretension. However, there can be more than one solution to this calculation, or possibly no solution. Therefore, a user-defined, estimated relaxation length is used to determine which of these multiple solutions is closest to the expected length. If the Estimated Relaxation Length property is set to 0, the shortest catenary length is used.

In some cases, the specified pretension may result in no possible solution. If this occurs, adjust the catenary properties until a result that meets all input parameters is found.

Pretension Catenary Mode Properties

Pretension Force	Specifies the pretension force that is applied at *Position 1* of the catenary.
Estimated Relaxation Length	Specifies the estimated catenary relaxation length. This property is used to determine the calculated relaxation length, as there may be more than one solution to the catenary calculation. A value of 0 uses the shortest calculated catenary relaxation length.
Calculated Relaxation Length	The calculated catenary relaxation length (read-only)

Setting Up Contact Couplings

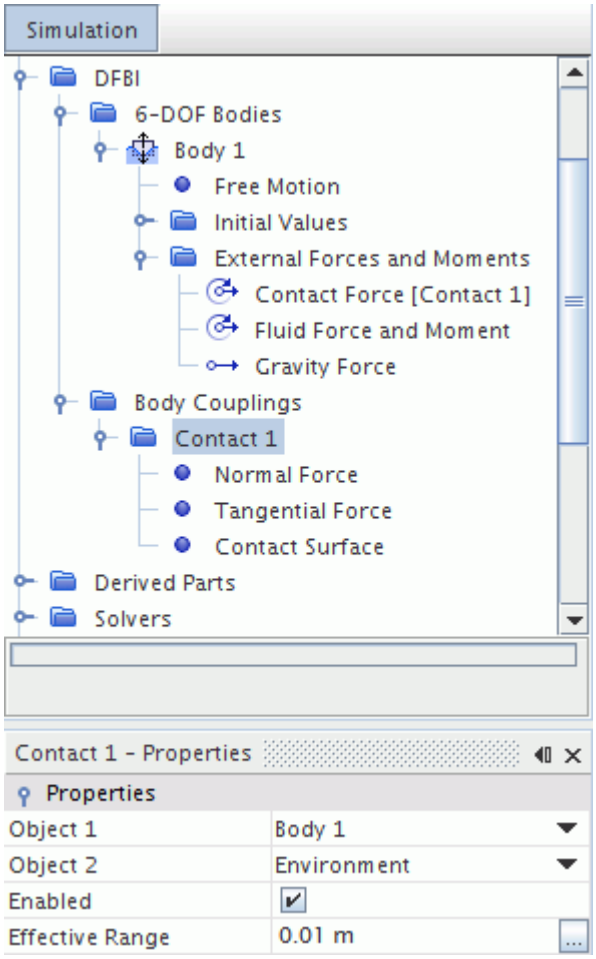
A 6-DOF body contact coupling represents a contact prevention force between a 6-DOF body and a boundary or between a 6-DOF body and another 6-DOF body. This contact prevention force acts on the 6-DOF body in a direction normal to the boundary and prevents the 6-DOF body from contacting the boundary or another 6-DOF body. This force is only active when the distance between the body and the boundary or another body falls below a user-defined effective range.

To set up a contact coupling:

1. Right-click the **DFBI** > **6-DOF Bodies** > **Body Couplings** node and select **New** > **Contact.**

   The **Body Couplings** > **Contact 1** node and the **6-DOF Bodies** > **Body 1** > **External Forces and Moments** > **Contact Force [Contact 1]** node are added to the simulation tree.
2. Select the **Body Couplings** > **Contact 1** node and set the appropriate properties.

   See Body_Couplings_Contact_Properties



3. Select each of the child nodes in turn and set their properties.
   - Normal Force Properties
   - Tangential Force Properties
   - Contact Surface Properties

To activate the Minimum Contact Distance field function:

4. Select the **Solver** > **6-DOF Solver** node and activate *Temporary Storage Retained*.

   For more information, see Contact Coupling Field Functions.

Contents:

Body Couplings Contact Properties

Body Couplings Contact Properties

Object 1	Specifies the 6-DOF body to which the contact prevention force is applied.
Object 2	(Read-only) Specifies the type of object *Object 1* couples to.

	Environment	Applies the contact prevention force to a set of boundaries. The boundaries are specified using the **Contact 1** > **Contact Surface** node.
	<Body>	Applies the contact prevention force to another body.
Enabled	Determines if the contact coupling is applied between *Object 1* and *Object 2.*	
	Activated	Contact prevention force is activated.
	Deactivated	Contact prevention force is deactivated.
Effective Range	Specifies the distance at which the contact prevention force starts to act on the 6-DOF body. The contact prevention force is only active when the body falls below the effective range.	
Index	(Read-only) Unique number to identify the coupling.	

Normal Force Properties

Elastic Coefficient	k_1 in Eqn. (993). To estimate this coefficient, see Eqn. (995).
Damping Coefficient	k_2 in Eqn. (993). To estimate this coefficient, see Eqn. (998).

Tangential Force Properties

Friction Coefficient	μ in Eqn. (1000).
Tanh Coefficient	k_t in Eqn. (1000).

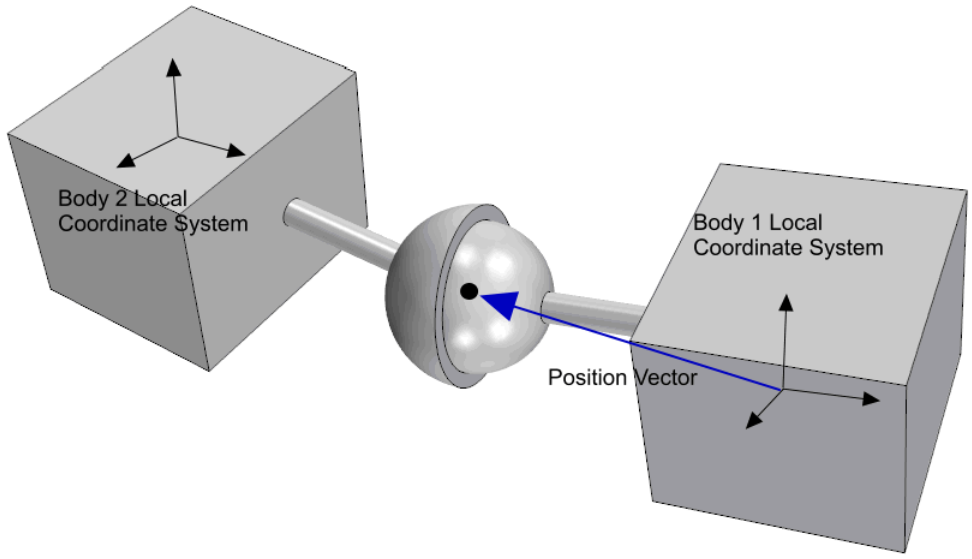
Contact Surface Properties

Inputs	Specifies the boundaries of the environment to which contact is to be prevented. These **Inputs** must not be set to the boundaries of a rigid body. If you want to prevent contact between two rigid bodies, you set *Object 1* and *Object 2* under the **Body Couplings > [Contact 1]** node each to a rigid body (see Body Couplings Contact Properties).

Spherical Joint Coupling

A spherical joint—often called ball-and-socket joint— allows you to couple two 6-DOF bodies that move relative to each other as part of a multi-body motion. The spherical joint restricts the relative motion of the two 6-DOF bodies to a pure rotation about the joint position. A relative translation of the two rigid bodies is not allowed.

The following diagram illustrates the concept of how a spherical joint is modeled in STAR-CCM+:



The spherical joint connects two 6-DOF bodies, which each has its own local body coordinate system. With respect to the respective local coordinate system, each body has a different view on where the spherical joint is located, that is, on the position of the spherical joint. The position in each of the local body coordinate systems remains constant while the bodies move. In the laboratory coordinate system, the position of each body must coincide. In this case, the constraint condition Eqn. (973) is fulfilled. However, the absolute position of the spherical joint with respect to the laboratory coordinate system can vary over time.

Spherical Joint Properties

Object 1

Sets the type of anchor to which the first end point of the spherical joint is attached. Must be different from *Object 2*.

- **Environment**: Attaches the first end point to a location in the environment.
- **Body**: Attaches the first end point to the selected 6-DOF body.

Object 2

Sets the type of anchor to which the second end point of the spherical joint is attached. Must be different from *Object 1*.

- **Environment**: Attaches the second end point to a location in the environment.
- **Body**: Attaches the second end point to the selected 6-DOF body.

Enabled

When on, the coupling is active.

Position

The coordinates of the spherical joint with respect to *Coordinate System*.

Coordinate System

The coordinate system in which *Position* is specified. Only local body coordinate systems of *Object 1* or *Object 2* are available.

Index

(Read-only) Unique number to identify the coupling.

See Also:

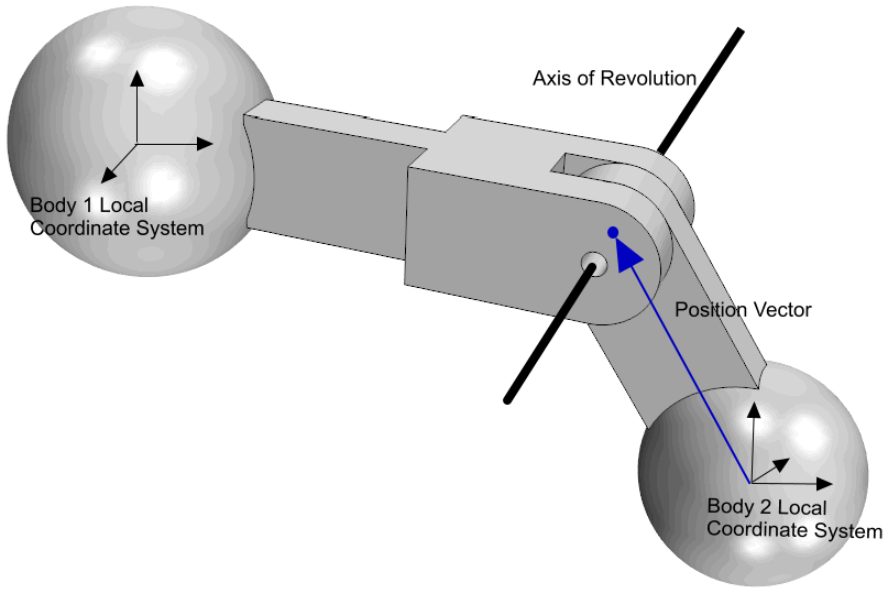
Setting up Multi-Body Motion

Governing Equations for Multi-Body Motion

Revolute Joint Coupling

A revolute joint—sometimes also called a pin or hinge—lets you couple two 6-DOF bodies that move relative to each other as part of a multi-body motion. The revolute joint allows a single relative rotation of the two 6-DOF bodies around the axis of revolution. All other relative rotations and translations are restricted.

The following diagram illustrates the concept of how a revolute joint is modeled in STAR-CCM+:



The revolute joint connects two 6-DOF bodies, which each has its own local body coordinate system. With respect to the respective local coordinate system, each body has a different view on where the revolute joint is located, that is, on the position and the axis direction of the revolute joint. In the laboratory coordinate system, the position and the axis direction of each body must coincide. In this case, the constraint condition Eqn. (973) is fulfilled. However, the absolute position and the axis of the revolute joint with respect to the laboratory reference frame can vary over time.

Revolute Joint Properties

Object 1

Sets the type of anchor to which the first end point of the revolute joint is attached. Must be different from *Object 2*.

- **Environment**: Attaches the first end point to a location in the environment.
- **Body**: Attaches the first end point to the selected 6-DOF body.

Object 2

Sets the type of anchor to which the second end point of the revolute joint is attached. Must be different from *Object 1*.

- **Environment**: Attaches the second end point to a location in the environment.
- **Body**: Attaches the second end point to the selected 6-DOF body.

Enabled

When on, the coupling is active.

Axis

The direction of the axis of revolution of the joint with respect to *Coordinate System*.

Position

The coordinates of the revolute joint with respect to *Coordinate System*.

Coordinate System

The coordinate system in which *Position* and *Axis* are specified. Only local body coordinate systems of *Object 1* or *Object 2* are available.

Index

(Read-only) Unique number to identify the coupling.

See Also:

Setting up Multi-Body Motion

Governing Equations for Multi-Body Motion

Setting up User-Defined Couplings

This coupling type lets you couple a rigid body to another rigid body or to the environment in a user-defined way. The user-defined coupling consists of two end points. The forces and moments acting at these end

points are defined through field functions, tables, or user-code. You can use this functionality, for example, to couple your STAR-CCM+ DFBI simulation to a third-party mooring dynamics API.

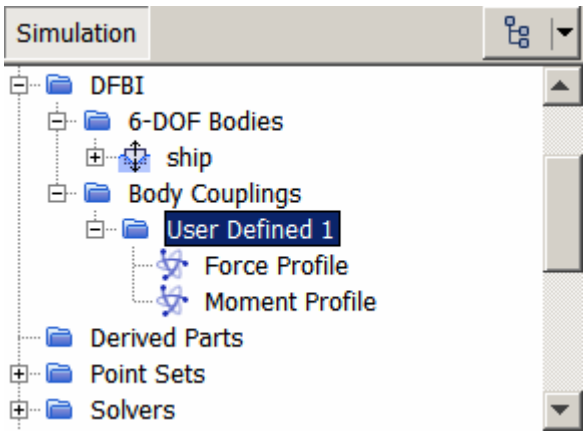
To set up user-defined couplings:

1. Right-click the **DFBI** > **6-DOF Bodies** > **Body Couplings** node and select **New** > **User Defined.**

   The following nodes are added to the simulation tree:

   - **Body Couplings** > **User Defined 1**
   - **6-DOF Bodies** > **[Body 1]** > **External Forces and Moments** > **User Defined Coupling Force [User Defined 1]**
   - **Point Sets** > **DFBI Point Set [User Defined Coupling 1]**

   The user-defined coupling automatically creates a point set containing two points. These two points represent the end points of the coupling. You can use the point set to access data that is defined at the end points or to visualize the end points in a scene.



2. Select the **Body Couplings** > **User Defined 1** node and specify the objects to be coupled:

   a) Set *Object 1* and *Object 2* to either a previously defined rigid body or to the **Environment**.

   b) To specify the positions of the attachment points of the coupling, enter coordinates for *Position 1* and *Position 2*.

   For a description of all properties, see User Defined Coupling Properties.

Next you define the force acting at the end points of the coupling and optionally an additional moment acting about the body position.

3. Select the **User Defined 1** > **Force Profile** node and set *Method* to one of the common options.

To define the force profile through field functions or user coding, specific field functions for user defined coupling are available.

4. If you set *Method* to **User Code**:

   a) Create the source code for the user coding in C, C++, or Fortran.
   An example coding for the user-defined method is shown below. The first file `uclib.c` defines two functions for the calculation of forces and moments that are called from STAR-CCM+:

```
#include "uclib.h"

void calculateForce(Real*, int, CoordReal*, unsigned int*, CoordReal*);
void calculateMoment(Real*, int, CoordReal* unsigned int*, CoordReal*);

void
USERFUNCTION_EXPORT uclib()
{
  /* Register user functions here */
  ucfunc(calculateForce, "PointSetProfile", "Calculate Force");
  ucarg(calculateForce, "Point", "Coord", sizeof(CoordReal[3]));
  ucarg(calculateForce, "Point", "BodyId", sizeof(unsigned int));
  ucarg(calculateForce, "Point", "BodyPosition", sizeof(CoordReal[3]));

  ucfunc(calculateMoment, "PointSetProfile", "Calculate Moment");
  ucarg(calculateMoment, "Point", "Coord", sizeof(CoordReal[3]));
  ucarg(calculateMoment, "Point", "BodyId", sizeof(unsigned int));
  ucarg(calculateMoment, "Point", "BodyPosition", sizeof(CoordReal[3]));
}
```

The following user code models a spring acting between the end points of the coupling. One end point is attached to a body and the other end point is attached to the environment or another body.

```
#include "uclib.h"

void USERFUNCTION_EXPORT
calculateForce(Real (*result)[3], int size, CoordReal (*position)[3],
  unsigned int *bodyId, CoordReal (*bodyPosition)[3])
{
  CoordReal delta[3];
  Real stiffness = 20.;     /* Spring stiffnes */
  int i, j;

  if (size != 2)    /* If the number of end points is unequal 2, do not execute the following coding. */
                    /* This if-block is essential for parallel simulations. The DFBI point */
                    /* set is filled with the two end points on the master node only. */

    return;

  for (j = 0; j < 3; ++j)   /* Compute force vectors at the end points */
  {
    delta[j] = position[1][j] - position[0][j];
    result[1][j] = -stiffness*delta[j]; /* Force at second end point */
    result[0][j] = -result[1][j];   /* Force at first end point */
  }
}

void USERFUNCTION_EXPORT
calculateMoment(Real (*result)[3], int size, CoordReal (*position)[3],
  unsigned int *bodyId, CoordReal (*bodyPosition)[3])
{
  int i, j;

  for (i = 0; i < size; ++i)
    for (j = 0; j < 3; ++j)
      result[i][j] = 0.;    /* Calculate no additional moment. */
}
```

b) Create a new user library.

c) Load the user library into STAR-CCM+.

d) Select the **Force Profile** node and set *Function* to **[your user coding]**.
For this example: Select the **Force Profile** node and set *Function* to **Calculate Force**.

For more information, see also Force Profile Properties.

5. (Optional) To define an additional moment, repeat the previous step, but edit the *Moment Profile* node instead.

   For more information, see also Moment Profile Properties.

6. To visualize user-defined couplings in a scene, add **Point Sets** > **DFBI Point Set [User Defined Coupling 1]** to a displayer.

   In the scene, the coupling is represented by its end points only. STAR-CCM+ is not aware of the actual coupling in between, which, by definition, is user-defined. You can access several specific field functions that are collected in the presenter group **User Defined DFBI Couplings** and apply them to scalar or vector displayers in a scene, and also to reports or expressions.

Contents:

User Defined Coupling Properties

User Defined Coupling Properties

Object 1

Sets the type of anchor to which the first end point of the user defined coupling is attached. Must be different from *Object 2*.

- **Environment**: Attaches the first end point to a location in the environment.
- **Body**: Attaches the first end point to the selected 6-DOF body.

Object 2

Sets the type of anchor to which the second end point of the user defined coupling is attached. Must be different from *Object 1*.

- **Environment**: Attaches the second end point to a location in the environment.
- **Body**: Attaches the second end point to the selected 6-DOF body.

Enabled

When on, the coupling is active.

Position 1

Anchoring position of the first end point of the user defined coupling on *Object 1*.

Position 2

Anchoring position of the second end point of the user defined coupling on *Object 2*.

Index

(Read-only) Unique number to identify the coupling. You can use it to identify different couplings in user-code or field functions.

User Defined Coupling Expert Properties

Coordinate System 1

The coordinate system in which *Position 1* is specified.

Coordinate System 2

The coordinate system in which *Position 2* is specified.

Compute Moments

When on, calculates the moments that are created by the forces acting at the end points of the user defined coupling. When off, you must provide all moments in the **Moment Profile**.

Force Profile

Method

Specify the force that acts at the end points of the user defined coupling as a vector profile. The force that you specify creates a moment. STAR-CCM+ calculates this moment automatically only if you activate *Compute Moments*.

Coordinate System

The coordinate system in respect to which the force vector is defined.

Moment Profile

Method

The moment that you specify here acts in addition to the moment that STAR-CCM+ calculates from the specified force when *Compute Moments* is active. This additional moment acts about a point of rotation, which is the body position. You can use it, for example, to define a user defined catenary coupling with additional torsional moments.

Coordinate System

The coordinate system in respect to which the moment vector is defined.

Body Couplings Field Functions

Contact Couplings

The following field functions are made available when activating contact coupling:

Minimum Contact Distance of [Contact]

Obtains the minimum distance between each face of the body surface and the nearest face on the opposing boundary, for the last iteration. If sub-stepping is used, the minimum distance is the global

minimum of all sub-steps of the last iteration. The minimum distance is projected onto the boundary normal. There is one Minimum Contact Distance available for each contact coupling in the simulation, so that results can be evaluated for each coupling separately. For body-body couplings, the Minimum Contact Distance is only available for the first of the two bodies.

To access this field function, select the **Solvers** > **6-DOF Solver** node and activate *Temporary Storage Retained*. Perform at least one iteration to populate the field. When you close the simulation and then restore it, the Minimum Distance field does not contain any values. You must perform at least one iteration to populate it again.

User-Defined Couplings

The following field functions are made available when activating user-defined coupling:

Body Acceleration

Accelerations of the connected bodies.

Body Angular Acceleration

Angular accelerations of the connected bodies.

Body Angular Velocity

Angular velocities of the connected bodies.

Body Id

Indices of the bodies that are connected to the end points.

Body Position

Positions of the connected bodies.

Body Velocity

Velocities of the connected bodies.

Coupling Acceleration

Accelerations of the end points of the coupling.

Coupling Id

Index of the coupling.

Coupling Velocity

Velocities of the end points of the coupling.

Setting up a Two-Dimensional DFBI Simulation

STAR-CCM+ allows you to create a two-dimensional rigid body. Currently, the body can move with a two-dimensional (2D) DFBI rotating motion with one degree of freedom. For example, you can use this capability for simulating 2D rotating electric machines.

To set up a 2D DFBI simulation:

1. Right-click the **DFBI** > **6-DOF Bodies** node and select **New** > **2D Body**.
2. Select the **[Body 2D 1]** node and set the appropriate properties to define the body and the body motion:
   a) Select the boundaries that compose the *Body Surface*.

      2D bodies can only be composed of boundaries of 2D regions. You must not mix 2D and 3D boundaries within the same body surface.

   b) Provide the *Body Mass* of the 2D body.

      The body mass that you specify is interpreted as value for one unit length of the body in z-direction.

   c) Set *Body Motion Option* to **One-DOF Rotating Motion**.

      This is currently the only option available.

   For a description of all properties, see Body 2D Properties
3. Select the **One-DOF Rotating Motion** node and specify the coordinates of the *Rotation Axis Origin* along with the *Coordinate System* in which it is defined.

   In STAR-CCM+, coordinates have generally three components even if the simulation is 2D. For the One-DOF rotating motion, the z-coordinate is not relevant.

   For more information on the properties, see One-DOF Rotating Motion Properties.
4. To define initial values such as **Angular Velocity**, **Center of Mass**, **Moment of Inertia**, and **Orientation**, expand the **Initial Values** node, select the sub-nodes in turn, and set their properties.

   Note that all values are interpreted as values for one unit length of the body in z-direction. For a full description of properties, see What are Initial Values.

Contents:

Two-Dimensional DFBI Reference

Two-Dimensional DFBI Reference

Body 2D Properties

Body Surface

A list of boundaries that represent the surface of the body. Opens the in-place part selector for editing. (Integration of shear force and pressure is over the body surface.) Select only boundaries of 2D regions. You must not mix 2D and 3D boundaries within the same body surface.

Body Mass

Mass of the body. Is interpreted as value for one unit length of the body in z-direction

Release Time

Time before calculation of body motion begins. It is best to allow some time for the fluid flow to initialize. The number of seconds depends on the time model you are using and the mesh size. In an unsteady model, the typical range would be 10 to 50 time-steps. If a time-step size is equivalent to 0.01 seconds, 50 time-steps would amount to 0.5 seconds.

Ramp Time

Facilitates a more robust solution by reducing oscillation. At release time, force and moments are applied to the body and can cause a shock effect. Applying a ramp time applies the forces and moments proportionally across the interval to reduce the shock effect. The value of this property is typically 10 times the release time.

Body Motion Option

Selects the type of body motion.

- **One-DOF Rotating Motion**— lets you rotate a body around a specified axis.

Three Dimensional

(Read-only) Indicates if the body is three-dimensional or two-dimensional. For a 2D body, it is deactivated by default.

Index

(Read-only) Unique number to identify the body.

One-DOF Rotating Motion

Rotation Axis Origin

The origin of the rotation axis. For the two-dimensional one-DOF rotating motion, the z-coordinate is not relevant.

Coordinate System

Coordinate system in respect to which the origin of the rotation axis is specified.

DFBI Formulation

This section provides the following formulation:

Contents:

Governing Equations of Free Motion
Governing Equation of One-DOF Rotating Motion
Governing Equation of One-DOF Translating Motion
Motion Limiter for One-DOF Rotating Motion
Motion Limiter for One-DOF Translating Motion
Governing Equations for Constrained Motions
Governing Equations for Multi-Body Motion
Resultant Force and Moment
Ramp Time
Torsional Spring Moment
Damping Moment
Damping Force
Linear Spring Coupling
Catenary Coupling
Contact Coupling

Governing Equations of Free Motion

The equation for the translation of the center of mass of the body is noted in the global inertial coordinate system:

$$m\frac{d\mathbf{v}}{dt} = \mathbf{f} \tag{929}$$

where m represents the mass of the body, $\mathbf{f}$ is the resultant force acting on the body and $\mathbf{v}$ is the velocity of the center of mass. The equation of rotation of the body is formulated in the Body Local Coordinate System with the origin in the center of mass of the body:

$$\mathbf{M}\frac{d\vec{\omega}}{dt} + \vec{\omega} \times \mathbf{M}\vec{\omega} = \mathbf{n} \tag{930}$$

where $\mathbf{M}$ is the tensor of the moments of inertia, $\vec{\omega}$ is the angular velocity of the rigid body and $\mathbf{n}$ is the resultant moment acting on the body. The tensor of the moments of inertia is expanded as:

$$\mathbf{M} = \begin{pmatrix} M_{xx} & M_{xy} & M_{xz} \\ M_{xy} & M_{yy} & M_{yz} \\ M_{xz} & M_{yz} & M_{zz} \end{pmatrix} \tag{931}$$

As this tensor is a symmetric, it can be defined by two vectors: one specifying the principal components, $\left(M_{xx}, M_{yy}, M_{zz}\right)$ along the diagonal, and another specifying the off-diagonal components, $\left(M_{xy}, M_{xz}, M_{yz}\right)$.

Governing Equation of One-DOF Rotating Motion

The equation of motion governing the rotation of a one-DOF rotating body is

$$M\frac{\partial}{\partial t}\omega = n \tag{932}$$

where n is the overall moment acting on the body with respect to the axis of rotation. M is the moment of inertia with respect to the rotation axis. ω is the rotation angular velocity with respect to the rotation axis.

Governing Equation of One-DOF Translating Motion

The equation of motion governing the displacement of a one-DOF translating body is

$$m\frac{dv}{dt} = f \tag{933}$$

where m represents the mass of the body, f is the resultant force acting on the body and v is the velocity of the center of mass.

Motion Limiter for One-DOF Rotating Motion

The motion limiter destroys the rotational energy of a one-DOF rotating body when the specified maximum angle α_{max} or minimum angle α_{min} is reached.

The stopping process starts when the rotation angle α is in the interval $\left[\alpha_{min}, \alpha_{min} + \lambda\right]$ or $\left[\alpha_{max} - \lambda, \alpha_{max}\right]$, where λ is the damping length.

A damping moment $n_{damping}$ is applied to the body and destroys the rotational energy of the body until the limiting angle is reached.

For the minimum angle,

$$n_{damping} = \frac{1}{2}\frac{M\omega^2}{\alpha - \alpha_{min}} \tag{934}$$

where M is the moment of inertia with respect to the axis of rotation. ω is the angular velocity of the body.

For the maximum angle,

$$n_{damping} = -\frac{1}{2}\frac{M\omega^2}{\alpha_{max} - \alpha} \tag{935}$$

To yield a more realistic stopping process, the full damping moment is not applied to the body. A smaller deceleration is applied at the beginning and a higher deceleration at the end.

Motion Limiter for One-DOF Translating Motion

The motion limiter destroys the kinetic energy of a one-DOF translating body when the specified maximum displacement d_{max} or minimum displacement d_{min} is reached.

The stopping process starts when the displacement d is in the interval $[d_{min}, d_{min} + \lambda]$ or $[d_{max} - \lambda, d_{max}]$, where λ is the damping length.

A damping force $f_{damping}$ is applied to the body and destroys the kinetic energy of the body until the limiting position is reached.

For the minimum displacement,

$$f_{damping} = \frac{1}{2}\frac{mv^2}{d - d_{min}} \tag{936}$$

where m is the mass of the body and v is the magnitude of the velocity of the body.

For the maximum displacement,

$$f_{damping} = \frac{1}{2}\frac{mv^2}{d - d_{max}} \tag{937}$$

To yield a more realistic stopping process, the full damping force is not applied to the body. A smaller deceleration is applied at the beginning and a higher deceleration at the end.

Governing Equations for Constrained Motions

It is sometimes useful to consider kinematic constraints for the simulation of body motion. These constraints are physical constraints to allow certain degrees of freedom and to restrict others.

The analytical expression of the holonomic constraint equations for a rigid body can generally be written as:

$$\Phi_k(\mathbf{d}, t) = 0 \tag{938}$$

where t is the time, $\mathbf{d}$ is the dependent coordinate vector, and p is the total number of constraints:

$$\mathbf{d} = \begin{pmatrix} r_x \\ r_y \\ r_z \\ \phi \\ \theta \\ \psi \end{pmatrix}$$

The number of degrees of freedom of the rigid body is therefore reduced to:

$$f = 6 - p \tag{939}$$

f is the number of degrees of freedom. f' denotes the minimum number of variables (independent variables) which would have described the motion of the rigid body completely. Only $f = 6-p$ equations are obtained using the independent coordinates. The method used here is based on the Jacobian matrix.

Eqn. (938) can be re-arranged to define the body position (dependent coordinates) $\mathbf{r}^{\mathrm{g}}$ and the orientation matrix $\mathbf{T}^{\mathrm{g}}$ in the laboratory coordinate system as a function of independent coordinates and time:

$\mathbf{r}^{\mathrm{g}} = \mathbf{r}(\mathbf{q}, t)$, $\mathbf{T}^{\mathrm{g}} = \mathbf{T}(\mathbf{q}, t)$

where $\mathbf{q}$ is the vector of the selected independent coordinates:

$$\mathbf{q} = \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_f \end{pmatrix}$$

Differentiating the above equations, gives:

$$\mathbf{v}^{\mathrm{g}} = \frac{\partial}{\partial \mathbf{q}}\mathbf{r}\dot{\mathbf{q}} + \frac{\partial}{\partial t}\mathbf{r} = \mathbf{J}_{\mathrm{T}}\dot{\mathbf{q}} + \mathbf{v}' \tag{940}$$

and

$$\boldsymbol{\omega} = \frac{\partial}{\partial \mathbf{q}}\mathbf{t}\dot{\mathbf{q}} + \frac{\partial}{\partial t}\mathbf{t} = \mathbf{J}_{\mathrm{R}}\dot{\mathbf{q}} + \boldsymbol{\omega}' \tag{941}$$

The $3 \times f$ Jacobian matrix for translation motion is:

$$\mathbf{J}_{\mathrm{T}} = \frac{\partial}{\partial \mathbf{q}}\mathbf{r} = \begin{pmatrix} \frac{\partial r_x}{\partial q_1} & \frac{\partial r_x}{\partial q_2} & \cdots & \frac{\partial r_x}{\partial q_f} \\ \frac{\partial r_y}{\partial q_1} & \frac{\partial r_y}{\partial q_2} & \cdots & \frac{\partial r_y}{\partial q_f} \\ \frac{\partial r_z}{\partial q_1} & \frac{\partial r_z}{\partial q_2} & \cdots & \frac{\partial r_z}{\partial q_f} \end{pmatrix} \tag{942}$$

and the $3 \times f$ Jacobian matrix for rotation motion is:

$$\mathbf{J}_{\mathrm{R}} = \frac{\partial}{\partial \mathbf{q}}\mathbf{t} = \begin{pmatrix} \frac{\partial t_x}{\partial q_1} & \frac{\partial t_x}{\partial q_2} & \cdots & \frac{\partial t_x}{\partial q_f} \\ \frac{\partial t_y}{\partial q_1} & \frac{\partial t_y}{\partial q_2} & \cdots & \frac{\partial t_y}{\partial q_f} \\ \frac{\partial t_z}{\partial q_1} & \frac{\partial t_z}{\partial q_2} & \cdots & \frac{\partial t_z}{\partial q_f} \end{pmatrix} \tag{943}$$

Combining Eqn. (942) with Eqn. (943) yields:

$$\mathbf{J} = \begin{pmatrix} \mathbf{J}_{\mathrm{T}} \\ \mathbf{J}_{\mathrm{R}} \end{pmatrix} \tag{944}$$

$\partial \mathbf{t} = \left(\partial t_x \ \partial t_y \ \partial t_z\right)^{\mathrm{T}}$ is an instant vector about which the rigid body rotates. The following tensor gives the components of this vector:

$$\partial \mathbf{T} \cdot \mathbf{T}^{\mathrm{T}} = \begin{pmatrix} 0 & -\partial t_z & \partial t_y \\ \partial t_z & 0 & -\partial t_x \\ -\partial t_y & \partial t_x & 0 \end{pmatrix} \tag{945}$$

The matrix elements of Eqn. (943) can be calculated as follows:

$$\frac{\partial}{\partial q_1} \mathbf{T} \cdot \mathbf{T}^{\mathrm{T}} = \begin{pmatrix} 0 & -\frac{\partial t_z}{\partial q_1} & \frac{\partial t_y}{\partial q_1} \\ \frac{\partial t_z}{\partial q_1} & 0 & -\frac{\partial t_x}{\partial q_1} \\ -\frac{\partial t_y}{\partial q_1} & \frac{\partial t_x}{\partial q_1} & 0 \end{pmatrix} \tag{946}$$

Differentiating Eqns. Eqn. (940) and Eqn. (941) yields the accelerations of the rigid body:

$$\dot{\mathbf{v}}^{\mathrm{g}} = \mathbf{J}_{\mathrm{T}} \ddot{\mathbf{q}} + \mathbf{L}_{\mathrm{T}} + \dot{\mathbf{v}}' \tag{947}$$

and

$$\boldsymbol{\omega}^{\mathrm{g}} = \mathbf{J}_{\mathrm{R}} \ddot{\mathbf{q}} + \mathbf{L}_{\mathrm{R}} + \boldsymbol{\omega}' \tag{948}$$

Here

$$\mathbf{L}_{\mathrm{T}} = \frac{\partial}{\partial \mathbf{q}} \left(\mathbf{J}_{\mathrm{T}} \dot{\mathbf{q}} \right) + 2 \frac{\partial}{\partial \mathbf{q}} \mathbf{v}' \tag{949}$$

and

$$\mathbf{L}_{\mathrm{R}} = \frac{\partial}{\partial \mathbf{q}} \left(\mathbf{J}_{\mathrm{R}} \dot{\mathbf{q}} \right) + 2 \frac{\partial}{\partial \mathbf{q}} \mathbf{v}' + \frac{\partial}{\partial t} \mathbf{J}_{\mathrm{R}} \tag{950}$$

with

$$\dot{\mathbf{v}}' = \frac{\partial^2 \mathbf{r}}{\partial t^2} \tag{951}$$

The governing equation under constraint conditions is given by:

$$\overline{\mathbf{M}} \ddot{\mathbf{q}} + \overline{\mathbf{N}} = \mathbf{F}^{\mathrm{g}} \tag{952}$$

where

$$\overline{\mathbf{M}} = \begin{pmatrix} m \mathbf{J}_T \\ \mathbf{M}^g \mathbf{J}_R \end{pmatrix},$$

$$\overline{\mathbf{N}} = \begin{pmatrix} m \cdot \mathbf{L}_{\mathrm{T}} \dot{\mathbf{q}} + m \cdot \dot{\mathbf{v}}' + m \left[\left(\mathbf{J}_{\mathrm{R}} \ddot{\mathbf{q}} + \mathbf{L}_{\mathrm{R}} \dot{\mathbf{q}} + \boldsymbol{\omega}' \right) \times \mathbf{r}_{PC} + \left(\mathbf{J}_{\mathrm{R}} \dot{\mathbf{q}} + \boldsymbol{\omega}' \right) \times \left(\left(\mathbf{J}_{\mathrm{R}} \dot{\mathbf{q}} + \boldsymbol{\omega}' \right) \times \mathbf{r}_{PC} \right) \right] \\ \mathbf{M}^{\mathrm{g}} \cdot \mathbf{L}_{\mathrm{R}} \dot{\mathbf{q}} \cdot \mathbf{M}^{\mathrm{g}} \cdot \boldsymbol{\omega}' + \left(\mathbf{J}_{\mathrm{R}} \dot{\mathbf{q}} + \boldsymbol{\omega}' \right) \times \mathbf{M}^{\mathrm{g}} \cdot \left(\mathbf{J}_{\mathrm{R}} \dot{\mathbf{q}} + \boldsymbol{\omega}' \right) \end{pmatrix},$$

where m is the mass of the body, $\mathbf{M}^{\mathrm{g}}$ is the tensor of the moments of inertia in the laboratory coordinate system, and $\mathbf{r}_{PC}$ is a position vector to describe the distance between the center of rotation and the center of mass. Only for the planar motion carriage mechanism, $\mathbf{r}_{PC} \neq 0$. For all other motions, $\mathbf{r}_{PC} = 0$.

The vector of forces and moments $\mathbf{F}^{\mathrm{g}}$ is defined as:

$$\mathbf{F}^{\mathrm{g}} = \begin{pmatrix} \mathbf{f}^{\mathrm{g}} \\ \mathbf{n}^{\mathrm{g}} \end{pmatrix} \tag{953}$$

$\mathbf{F}^{\mathrm{g}}$ can be split into two parts:

$$\mathbf{F}^{\mathrm{g}} = \mathbf{F}^{\mathrm{i,g}} + \mathbf{F}^{\mathrm{q,g}} \tag{954}$$

$\mathbf{F}^{\mathrm{i,g}}$ stands for the impressed forces (or moments) acting on the body. These forces can either be obtained from the fluid flow or any external sources. $\mathbf{F}^{\mathrm{q,g}}$ stands for the forces (or moments) which maintain the given constraints.

The forces (or moments) $\mathbf{F}^{\mathrm{q,g}}$ are still unknown. $\mathbf{F}^{\mathrm{q,g}}$ can be eliminated by substituting Eqn. (954) into Eqn. (952), multiplying the matrix $\mathbf{J}^{\mathrm{T}}$ to both sides of Eqn. (952), and by using:

$$\mathbf{J}^{\mathrm{T}}\mathbf{F}^{\mathrm{q}} = 0 \tag{955}$$

As a result, you get:

$$\left(\mathbf{J}^{\mathrm{T}}\overline{\mathbf{M}}\right)\ddot{\mathbf{q}} + \mathbf{J}^{\mathrm{T}}\overline{\mathbf{N}} = \mathbf{J}^{\mathrm{T}}\mathbf{F}^{\mathrm{i,g}} \tag{956}$$

This equation has $f = 6-p$ variables.

Contents:

Governing Equations for Four-DOF Maneuvering Motion
Governing Equations for Planar Motion Carriage
Governing Equations for Planar Motion Mechanism (PMM)
Governing Equations for Rotating Arm Motion
Governing Equations for General Planar Motion

Governing Equations for Four-DOF Maneuvering Motion

Consider a solid body, for example a ship, moving with constant velocity V and a constant course angle ψ. Four degrees of freedom are free, for the simulation of a ship this can be sway, heave, roll, and pitch motion. They can be selected as independent variables.

The independent coordinate can be written as:

$$\mathbf{q} = \begin{pmatrix} y \\ z \\ \phi \\ \theta \end{pmatrix} \tag{957}$$

The solid body position $\mathbf{r}$and the orientation $\mathbf{T}$can be expressed by:

$$\mathbf{r} = \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix} = \begin{pmatrix} -y \cdot \sin\psi + V \cdot t \cdot \cos\psi \\ y \cdot \cos\psi + V \cdot t \cdot \sin\psi \\ z \end{pmatrix},$$

$$\mathbf{T} = \begin{pmatrix} \cos\theta\cos\psi & \sin\varphi\sin\theta\cos\psi & \cos\varphi\sin\theta\cos\psi + \sin\varphi\sin\psi \\ \cos\theta\sin\psi & \sin\varphi\sin\theta\cos\psi + \cos\varphi\cos\psi & \cos\varphi\sin\theta\sin\psi - \sin\varphi\cos\psi \\ -\sin\theta & \sin\varphi\cos\theta & \cos\varphi\cos\theta \end{pmatrix} \tag{958}$$

According to Eqns. Eqn. (942) and Eqn. (943), the Jacobian matrices for translation and rotation motions are for Four-DOF motions:

$$\mathbf{J}_T = \begin{pmatrix} -\sin\psi & 0 & 0 & 0 \\ \cos\psi & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{959}$$

$$\mathbf{J}_R = \begin{pmatrix} 0 & 0 & \cos\psi\cos\theta & -\sin\psi \\ 0 & 0 & \sin\psi\cos\theta & \cos\psi \\ 0 & 0 & -\sin\theta & 0 \end{pmatrix} \tag{960}$$

According to Eqns. Eqn. (949), Eqn. (950), Eqn. (947), Eqn. (948), Eqn. (951), the other matrices and vectors are for Four-DOF motions:

$\mathbf{L}_T = 0,$

$$L_R = \begin{pmatrix} 0 & 0 & 0 & -\dot{\varphi}\cos\psi\sin\theta \\ 0 & 0 & 0 & -\dot{\varphi}\sin\psi\sin\theta \\ 0 & 0 & 0 & -\dot{\varphi}\cos\theta \end{pmatrix},$$

$$\mathbf{v}' = \begin{pmatrix} V \cdot \cos\psi \\ V \cdot \sin\psi \\ 0 \end{pmatrix}, \mathbf{\omega}' = 0$$

$\dot{\mathbf{v}}' = 0, \dot{\mathbf{\omega}}' = 0$

Governing Equations for Planar Motion Carriage

The planar motion carriage has only three free degrees of freedom. The other three are restricted by the prescribed motion of the planar motion carriage.

The independent coordinate is given by:

$$\mathbf{q} = \begin{pmatrix} z \\ \theta \\ \phi \end{pmatrix}, f = 3$$

The solid body position $\mathbf{r}$ can be written as:

$$\mathbf{r} = \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix} = \begin{pmatrix} r_x(t) \\ r_y(t) \\ z(t) \end{pmatrix} \tag{961}$$

Eqn. (961) describes the position vector in a general form. The orientation is described by Eqn. (958) with the yaw angle $\psi = \psi(t)$. Depending on the planar motion carriage method that has been selected, the position vector and the orientation matrix can vary. The first planar motion carriage method that has been implemented is the Planar Motion Mechanism (PMM).

For more details on the formulation of this method, see Governing_Equations_for_Planar_Motion_Mechanism__PMM_.

For the planar motion carriage, the Jacobian matrices can be written as:

$$\mathbf{J}_{\mathrm{T}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \tag{962}$$

$$\mathbf{J}_{\mathrm{R}} = \begin{pmatrix} 0 & -\sin\psi & \cos\psi \cdot \cos\theta \\ 0 & \cos\psi & \sin\psi \cdot \cos\theta \\ 0 & 0 & -\sin\theta \end{pmatrix} \tag{963}$$

According to Eqns. Eqn. (949), Eqn. (950), Eqn. (947), Eqn. (948), Eqn. (951), the other matrices and vectors can be expressed as:

$\mathbf{L}_{\mathrm{T}} = 0,$

$$\mathbf{L}_{\mathrm{R}} = \begin{pmatrix} 0 & -\cos\psi\frac{\partial\psi}{\partial t} - \cos\psi \cdot \sin\theta \cdot \dot{\phi} & -\sin\psi \cdot \cos\theta \cdot \frac{\partial\psi}{\partial t} \\ 0 & -\sin\psi\frac{\partial\psi}{\partial t} - \sin\psi \cdot \sin\theta \cdot \dot{\phi} & \cos\psi \cdot \cos\theta \cdot \frac{\partial\psi}{\partial t} \\ 0 & -\cos\theta \cdot \dot{\phi} & 0 \end{pmatrix},$$

$$\mathbf{v}' = \begin{pmatrix} \frac{\partial r_x}{\partial t} \\ \frac{\partial r_y}{\partial t} \\ 0 \end{pmatrix}, \ \boldsymbol{\omega}' = \begin{pmatrix} 0 \\ 0 \\ \frac{\partial\psi}{\partial t} \end{pmatrix},$$

$$\dot{\mathbf{v}}' = \begin{pmatrix} \frac{\partial^2 r_x}{\partial t^2} \\ \frac{\partial^2 r_y}{\partial t^2} \\ 0 \end{pmatrix}, \ \dot{\boldsymbol{\omega}}' = \begin{pmatrix} 0 \\ 0 \\ \frac{\partial^2\psi}{\partial t^2} \end{pmatrix}$$

Governing Equations for Planar Motion Mechanism (PMM)

The Planar Motion Mechanism is a method of the Planar Motion Carriage that prescribes the motion in the X-Y plane in the form of a sinusoidal sway motion.

The solid body position $\mathbf{r}$ can be written as:

$$\mathbf{r} = \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix} = \begin{pmatrix} V_0 \cdot t \\ y_0 \cdot \sin(2\pi f \cdot t) \\ z \end{pmatrix} \tag{964}$$

where V_0 is the forward velocity of the body, y_0 is the sway amplitude, and f' is the sway frequency.

The orientation $\mathbf{T}$ corresponds to Eqn. (958) with the roll angle $\phi = 0$.

If pure yawing is activated, the yawing angle is given by:

$$\psi = \psi_0 \cdot \cos\left(2\pi f \cdot t\right) \tag{965}$$

with

$$\Psi_0 = \arctan\left(\frac{y_0 \cdot 2\pi f}{V_0}\right) \tag{966}$$

else $\psi = 0$.

Governing Equations for Rotating Arm Motion

The Rotating Arm Motion mechanism is a method of the Planar Motion Carriage that prescribes the motion in the X-Y plane in the form of a rotation. The result is a circular path of the body in the X-Y plane.

The solid body position $\mathbf{r}$ can be written as:

$$\mathbf{r} = \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix} = \begin{pmatrix} r_{c,x} + \left(r_{0,x} - r_{c,x}\right)\cos\omega t - \left(r_{0,y} - r_{c,y}\right)\sin\omega t \\ r_{c,y} + \left(r_{0,x} - r_{c,x}\right)\sin\omega t - \left(r_{0,y} + r_{c,y}\right)\cos\omega t \\ z \end{pmatrix} \tag{967}$$

where:

$\mathbf{r}_c = \begin{pmatrix} r_{c,x} \\ r_{c,y} \\ r_{c,z} \end{pmatrix}$: Center of rotation of the rotating arm

$\mathbf{r}_0 = \begin{pmatrix} r_{0,x} \\ r_{0,y} \\ r_{0,z} \end{pmatrix}$: Initial body position

The initial body position corresponds to the location where the body is initially attached to the rotating arm.

The yaw angle is given by:

$$\psi = \omega \cdot t \tag{968}$$

Governing Equations for General Planar Motion

The General Planar Motion mechanism is a method of the Planar Motion Carriage that prescribes the motion in the X-Y plane in the form of a user-defined trajectory.

The solid body position $\mathbf{r}$ can be written as:

$$\mathbf{r} = \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix} = \begin{pmatrix} x(t) \\ y(t) \\ z \end{pmatrix} \tag{969}$$

The yaw angle is given by:

$$\psi = \psi(t) \tag{970}$$

where x, y, ψ are user-defined functions of time.

Governing Equations for Multi-Body Motion

Multi-body motion describes the dynamic behaviour of multiple interconnected rigid bodies. The rigid bodies can translate and rotate relative to each other whereby they are coupled by using different types of joints. The dynamic behaviour of the bodies is described by the equations of motion of the bodies and the constraint equations at the joints.

Two or more bodies are connected by one or more joints. A joint is defined by certain kinematic constraints. The position of the bodies relative to each other is therefore restricted by the kinematic constraint condition.

The equation of motion describes the dynamic behaviour of a multi-body system:

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{f} \tag{971}$$

where $\mathbf{q}$ are generalized coordinates, $\mathbf{f}$ is a generalized force vector acting on the bodies, and $\mathbf{M}$ is the block-diagonal matrix of the inertia matrices of the rigid bodies:

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{M}_n \end{pmatrix}$$

The equation of motion is combined with a constraint condition that restricts the kinematic degrees of freedom of the bodies:

$$\Phi(\mathbf{q}, t) = 0 \tag{972}$$

A constraint is expressed as a linear condition on the accelerations of the bodies. Forming the second derivative of Eqn. (972) yields:

$$\mathbf{J}\ddot{\mathbf{q}} = \mathbf{Q} \tag{973}$$

where $\mathbf{J}$ is the Jacobian matrix of Φ, $\ddot{\mathbf{q}}$ is the acceleration of the bodies, and $\mathbf{Q}$ is an inhomogeneity.

To enforce the acceleration conditions of the constraints, a constraint force is added to the system. By introducing the Lagrangian multiplier $\boldsymbol{\lambda}$ of all constraints, the workless constraint force is given by:

$$\mathbf{f}^c = \mathbf{J}^T \cdot \boldsymbol{\lambda} \tag{974}$$

A vector $\boldsymbol{\lambda}$ is sought such that the constraint force $\mathbf{f}^c$ in combination with any external force $\mathbf{f}^{ext}$ such as gravity produces a motion that satisfies the constraints as given by Eqn. (973).

Re-arranging Eqn. (971) under consideration of Eqn. (974) gives:

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{J}^T\boldsymbol{\lambda} + \mathbf{f}^{ext} \tag{975}$$

Solving for $\ddot{\mathbf{q}}$ gives:

$$\ddot{\mathbf{q}} = \mathbf{M}^{-1}\mathbf{J}^T\boldsymbol{\lambda} + \mathbf{M}^{-1}\mathbf{f}^{ext} \tag{976}$$

Inserting Eqn. (976) into Eqn. (973) yields the following linear system of equations:

$$\mathbf{A}\boldsymbol{\lambda} = \mathbf{b} \tag{977}$$

where matrix $\mathbf{A}$ and vector $\mathbf{b}$ are defined as:

$$\mathbf{A} = \mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T$$

and

$$\mathbf{b} = -\mathbf{J}\mathbf{M}^{-1}\mathbf{f}^{ext} + \mathbf{Q}$$

$\boldsymbol{\lambda}$ is obtained from Eqn. (977). With $\boldsymbol{\lambda}$ known, Eqn. (976) is integrated twice to yield the generalized coordinates vector.

Constraint Stabilization

The integration to obtain the coordinates vector $\mathbf{q}$ is performed numerically. In STAR-CCM+, the constraints are imposed in an acceleration form as given by Eqn. (973). For such a method, it is natural that some constraint drift occurs due to round-off and numerical integration errors. The goal of constraint stabilization is to ensure that this drift does not accumulate over time and that it remains bounded.

Constraint stabilization is applied by modifying Eqn. (973) according to Baumgarte [144]:

$$\mathbf{J}\ddot{\mathbf{q}} = \mathbf{Q}' \tag{978}$$

with:

$$\mathbf{Q}' = \mathbf{Q} - 2\alpha\left(\mathbf{J}\dot{\mathbf{q}} + \frac{\partial\Phi}{\partial t}\right) - \beta^2\Phi \tag{979}$$

where α and β are parameters of the Baumgarte constraint stabilization method.

See Also:

Spherical Joint Coupling

Revolute Joint Coupling

Setting up Multi-Body Motion

Resultant Force and Moment

The resultant force and moment acting on the rigid body are obtained from the fluid pressure and shear forces acting on each boundary face of the body.

Pressure forces:	$\mathbf{f}_p = \sum_f p_f \mathbf{a}_f$

Pressure moments:	$\mathbf{n}_p = \sum_f \left[\mathbf{r}_f \times \left(p_f \mathbf{a}_f\right)\right]$
Shear forces:	$\mathbf{f}_\tau = -\sum_f \boldsymbol{\tau}_f a_f$
Shear moments:	$\mathbf{n}_\tau = -\sum_f \left[\mathbf{r}_f \times \left(\boldsymbol{\tau}_f a_f\right)\right]$
Gravity force:	$\mathbf{f}_g = m\mathbf{g}$

In the above equations, p_f is the pressure acting on face f, $\mathbf{a}_f$ is the area vector of face f, $\mathbf{r}_f$ is the distance vector from the body center of mass to the center of face f, $\boldsymbol{\tau}_f$ is the shear stress acting on face f, and g is the gravity vector.

The overall resultant force and moment acting on the body are therefore:

$$\mathbf{f} = f_r\left(\mathbf{f}_p + \mathbf{f}_\tau + \mathbf{f}_g + \sum \mathbf{f}_{ext}\right) \tag{980}$$

$$\mathbf{n} = f_r\left(\mathbf{n}_p + \mathbf{n}_\tau + \sum \mathbf{n}_{ext}\right) \tag{981}$$

where f_r is the ramping factor that is defined in Eqn. (982) below. $\mathbf{f}_{ext}$ represents an external user-defined force acting on the body, and $\mathbf{n}_{ext}$ represents an external moment (which you can define directly or as the result of an external user-defined force).

Ramp Time

For 6-DOF bodies, with the values of release time t_s and ramp time t_r a ramping function is defined as follows:

$$f_r = \begin{cases} 0 & : \quad t < t_s \\ \dfrac{t - t_s}{t_r} & : \; t_s \le t < t_s + t_r \\ 1 & : \quad t \ge t_s + t_r \end{cases} \tag{982}$$

This ramping function $f_r(t)$ is applied to external forces and moments on the body, forces, and moments that are induced by the fluid flow on the body and to the gravity force.

Torsional Spring Moment

The torsional spring moment M acting on a rotating body is defined as:

$$M = -k\left(\alpha - \alpha_r\right) \tag{983}$$

where k is the spring constant, α is the rotation angle of the body and α_r is the relaxation angle at which the moment vanishes.

Damping Moment

The damping moment acts with respect to the center of mass of the body in opposition to the angular velocity vector.

It defined as:

$$\mathbf{n}_d = -a_{d,n}\vec{\omega} \tag{984}$$

where $a_{d,n}$ is the angular damping constant.

Damping Force

The damping force acts with respect to the center of mass of the body in opposition to the velocity vector.

It is defined as:

$$\mathbf{f}_d = -a_{d,f}\mathrm{v} \tag{985}$$

where $a_{d,f}$ is the linear damping constant.

Linear Spring Coupling

The linear spring coupling provides a restoring force, which means that it acts in the direction that tends to restore the spring to a state of equilibrium.

The forces acting at the two end points of the spring are given according to Hooke's law, and are defined as:

$$\begin{gathered}\mathbf{f}_1 = -k_s\left(\mathbf{r}_s - \mathbf{r}_{eq}\right) \\ \mathbf{f}_2 = -\mathbf{f}_1\end{gathered} \tag{986}$$

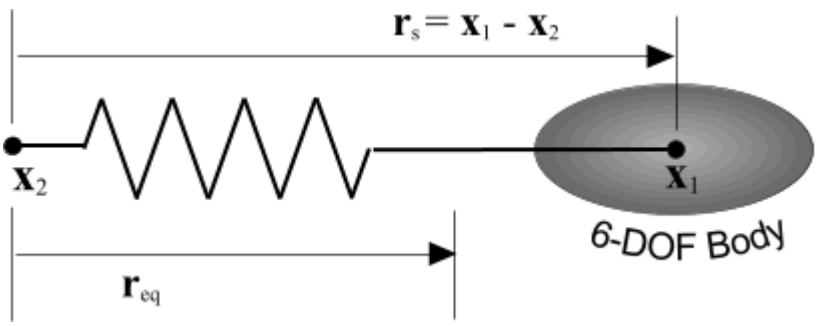
where

$$\mathbf{r}_s = \mathbf{x}_1 - \mathbf{x}_2 \tag{987}$$

and

$$\mathbf{r}_{eq} = l_{eq}\mathbf{r}_s / \left|\mathbf{r}_s\right| \tag{988}$$

In these equations, $\mathbf{x}_1$ is the location to which one end of the spring is attached, $\mathbf{x}_2$ is the opposite end of the spring, $\mathbf{f}_1$ is the force acting at $\mathbf{x}_1$, $\mathbf{f}_2$ is the reaction force acting at $\mathbf{x}_2$, and l_{eq} is the relaxation or equilibrium length of the spring. When $l_{eq} \equiv \left|\mathbf{r}_s\right|$, the spring force is zero. Some of these parameters are illustrated in the following diagram:



In this diagram, one end of the spring, $\mathbf{x}_1$, is attached to a 6-DOF body and the other end, $\mathbf{x}_2$, is attached to the environment. This is only one of several possible configurations. It is also possible to attach $\mathbf{x}_2$ to a 6-DOF body and $\mathbf{x}_1$ to the environment, or to attach $\mathbf{x}_1$ to one 6-DOF body and $\mathbf{x}_2$ to another.

Catenary Coupling

The catenary coupling models an elastic, quasi-stationary catenary (such as a chain or towing rope), which is hanging between two end points, being subject to its own weight in the gravity field.

In a local cartesian coordinate system, the shape of the catenary is given by the following set of parametric equations:

$$\begin{aligned} x &= au + b\sinh(u) + \alpha \\ y &= a\cosh(u) + \frac{b}{2}\sinh^2(u) + \beta \\ &\text{for } u_1 \le u \le u_2 \end{aligned} \tag{989}$$

where

$a = \frac{c}{\lambda_0 g}$

$b = \frac{ca}{DL_{eq}}$

$c = \frac{\lambda_0 L_{eq} g}{\sinh(u_2) - \sinh(u_1)}$

In these equations, g is the gravitational acceleration, λ_0 and L_{eq} are the mass per unit length and the relaxation length of the catenary, respectively, under force-free conditions. D is the stiffness of the catenary, and α and β are integration constants depending on the position of the two end points and the total mass of the catenary. The curve parameter u is related to the inclination angle ϕ of the catenary curve by the following equation:

$$\tan\phi = \sinh(u) \tag{990}$$

The parameter values u_1 and u_2 represent the positions of the catenary's end points p_1 and p_2 in parameter space.

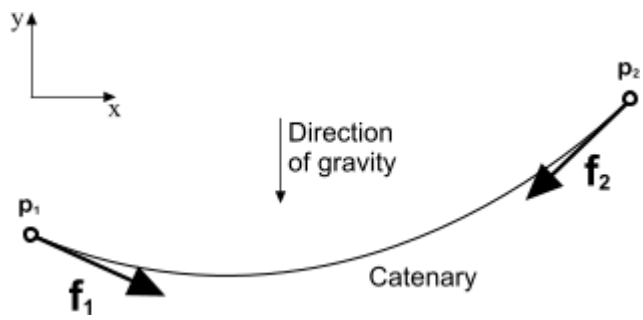
The forces $\mathbf{f}_1$ and $\mathbf{f}_2$ acting at the two end points of the catenary are directed along the local tangent vectors of the catenary curve at the parameter values u_1 and u_2, respectively. They are given by the following expressions:

$$\begin{aligned} \mathbf{f}_{1,x} &= c \\ \mathbf{f}_{1,y} &= c\sinh\left(u_1\right) \end{aligned} \tag{991}$$

and

$$\begin{aligned} \mathbf{f}_{2,x} &= -c \\ \mathbf{f}_{2,y} &= -c\sinh\left(u_2\right) \end{aligned} \tag{992}$$

These forces, and the resulting catenary curve, are illustrated in the diagram below.



Contact Coupling

The DFBI contact coupling prevents contact of a DFBI body with a boundary. This boundary can either be a boundary of the surrounding computational domain (Body-Environment coupling) or the surface of another DFBI body (Body-Body coupling). To achieve contact prevention, the model calculates an external force normal to the boundary that acts on the DFBI body.

The contact force depends on the distance between the DFBI body surface and the opposing boundary. The contact coupling is a near-field coupling, that is, the force decreases to zero when the distance between the body and the boundary is larger than a user-specified effective range. If the distance is smaller than the effective range, a repulsive contact force between the DFBI body and the boundary is created. The contact force is distributed per face over the body surface and has two force components:

- Normal component, which is normal to the opposing boundary. This force component prevents contact.
- Tangential component, which is tangential to the opposing boundary. This component models the friction of the DFBI body sliding along the boundary.

The normal component of the contact force on face f of the body surface is given by:

$$\mathbf{f}_n\left(d_f\right) = a_f\left[k_1\left(d_0 - d_f\right) - k_2\dot{d}_f\right]\mathbf{n}_{bf} \tag{993}$$

where

k_1	Elastic coefficient

k_2	Damping coefficient
a_f	Face area
d_f	Distance between face centroid and boundary
$\dot{d}_f$	Rate of change of the distance between face centroid and boundary
d_0	Effective range
$\mathbf{n}_{bf}$	Normal of the boundary face next to face f

The contact force has an elastic term and a damping term that are specified by the elastic coefficient k_1 and the damping coefficient k_2, respectively.

The elastic coefficient k_1 is essential to stop the DFBI body well before contact is made. To estimate a good value for this coefficient, you can apply the following relations:

- Body-Body couplings:

$$k_1 = \frac{m_1 m_2}{m_1 + m_2} \cdot \frac{1}{A} \cdot \frac{v_{n,rel}^2}{(d_0 - d_{min})^2} \tag{994}$$

- Body-Environment couplings:

$$k_1 = \frac{m_1}{A} \cdot \frac{v_{n,rel}^2}{(d_0 - d_{min})^2} \tag{995}$$

where

m_1, m_2	Body mass of Body 1 and Body 2
A	Estimated area of contact
$v_{n,rel}$	Normal relative impact velocity of the two coupled objects
d_{min}	Minimum distance. Body should stop before this distance is reached.

You can estimate the time that is needed to stop the DFBI body by applying the contact force as follows:

- Body-Body couplings:

$$\tau = \frac{\pi}{2}\sqrt{\frac{m_1 \cdot m_2}{(m_1 + m_2)Ak_1}} \tag{996}$$

- Body-Environment couplings:

$$\tau = \frac{\pi}{2}\sqrt{\frac{m_1}{Ak_1}} \tag{997}$$

To properly resolve the process of stopping, you must set the time step of the simulation sufficiently smaller than τ. If the simulation time step is too large to resolve the process, you can enable 6-DOF Solver sub-

stepping. Sub-stepping allows STAR-CCM+ to resolve the stopping process in DFBI. The above relations are only valid for weak damping.

To estimate a value for the damping coefficient, you can use the following relations:

- Body-Body coupling:

$$k_2 = 2\varsigma\sqrt{\frac{(m_1 \cdot m_2)k_1}{(m_1 + m_2)A}} \tag{998}$$

- Body-Environment coupling:

$$k_2 = 2\varsigma\sqrt{\frac{m_1 k_1}{A}} \tag{999}$$

ς is a constant factor describing the amount of damping. For $\varsigma = 1$, damping is critical. Note that very large values of ς in combination with a large time step can cause numerical instability of the damping force. To avoid numerical instability, use a sufficiently small value of ς. When in doubt, use $\varsigma \ll 1$.

The tangential component of the contact force on face f of the body surface is given by:

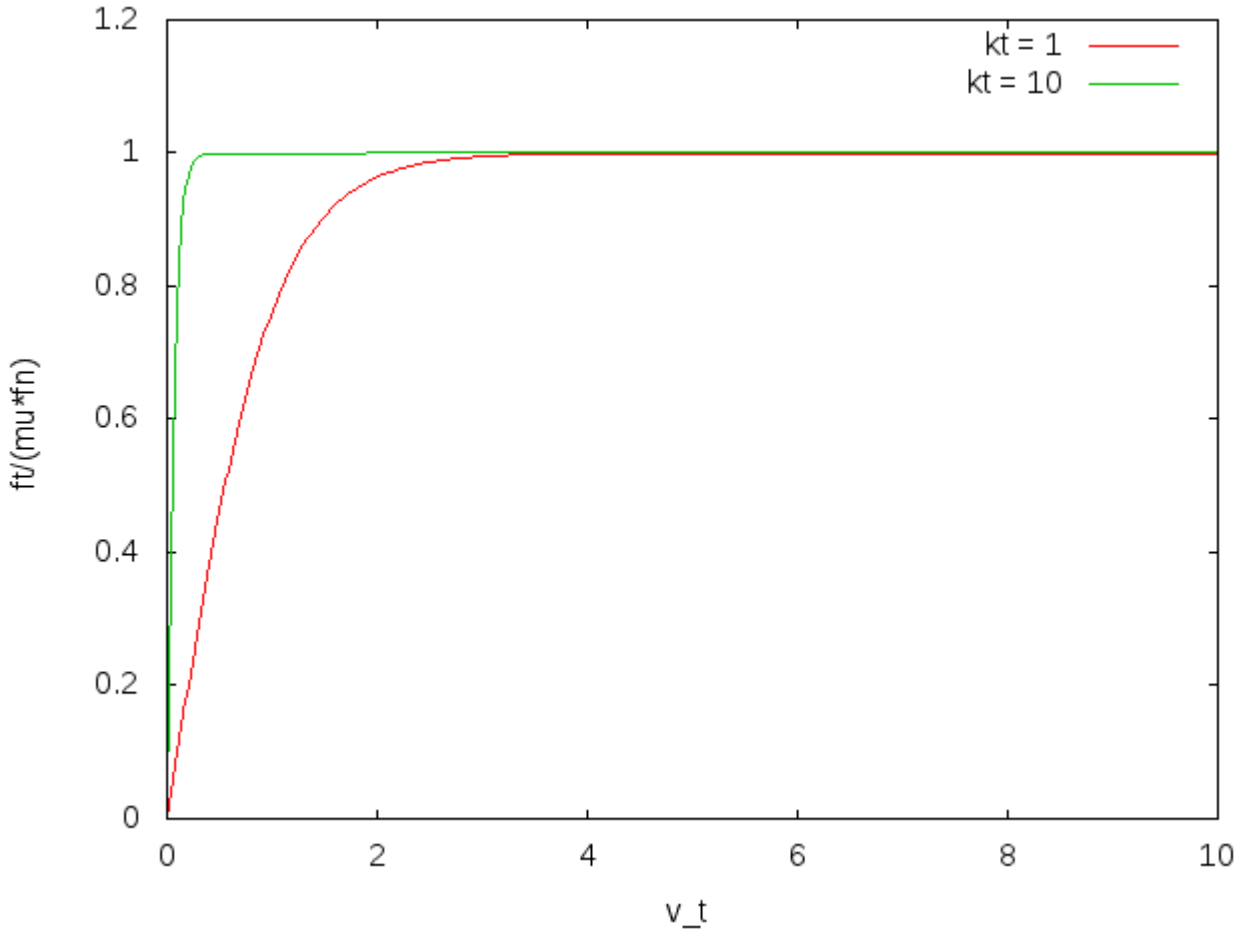
$$\mathbf{f}_t = -\mu|\mathbf{f}_n|\tanh(k_t\mathbf{v}_t) \tag{1000}$$

where

μ	Friction coefficient
k_t	tanh coefficient
$\mathbf{v}_t$	Relative velocity component of face f tangential to the boundary

This tangential force component models the kinetic friction between the body and the boundary. Static friction is not accounted for. The tanh function makes the friction force stable for small relative sliding velocities.

For sliding velocities $\mathbf{v}_t \gg 1/k_t$, the friction force depends only on the friction coefficient, the normal force, and the direction of the relative velocity. Depending on k_t, the friction force remains constant with increasing sliding velocity. For smaller sliding velocities, the friction force additionally depends on the velocity magnitude.



The total contact force on face f is given by:

$$\mathbf{f}_f = \mathbf{f}_n + \mathbf{f}_t \tag{1001}$$

The total force creates a moment $\mathbf{m}_f$ around the current body position $\mathbf{r}_b$:

$$\mathbf{m}_f = \left(\mathbf{r}_f - \mathbf{r}_b\right) \times \mathbf{f}_f \tag{1002}$$

where $\mathbf{r}_f$ is the location of the face centroid of face f. The total force and the total moment on the body are each the sum of all contributions of all faces of the body surface.

DFBI Bibliography

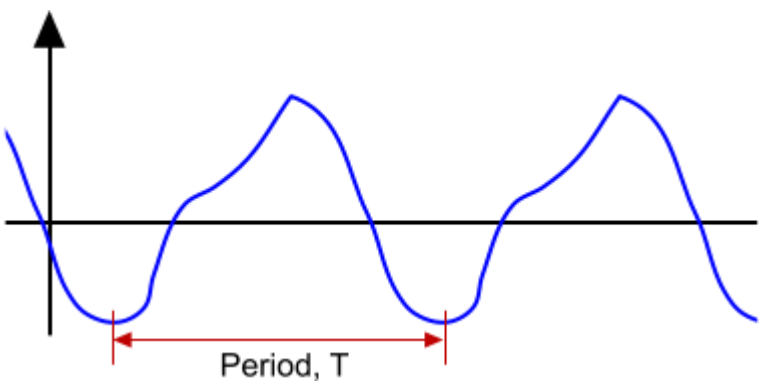
[143] *Ohmori, T. 1998. “Finite-volume simulation of flows around a ship in maneuving motion”, J. Mar. Sci. Tchnol., 3, pp. 82-93.*

[144] *Shabana, A. A. 2001. " Computational Dynamics" , Second Edition, John Wiley & Sons.*

Modeling with Harmonic Balance

Some unsteady flows have a regularly repeating flow pattern, that is, they are time-periodic. Consider the flow from a fan blade passing across the entrance to a duct. Measurements of the instantaneous flow just within the duct would show a regularly repeating pattern. If the flow disturbances are sufficiently large, and propagate to the end of the duct, measurements of the unsteady flow at any point within the duct show repeating patterns. Such time-periodic patterns can be expressed using Fourier series.

Fourier series are series of cosine and sine terms that when taken together, reproduce a periodic function.



This periodic function, which is also known as a trigonometric series, can be written in simple form as:

$$a_0 + \sum_{n=1}^{\infty} \left(a_n \cos\left((nx) + b_n \sin(nx)\right)\right)$$

The STAR-CCM+ harmonic balance method is a mixed frequency/time domain technique for computational fluid dynamics. It enables the efficient calculation of time periodic flows. The unsteady solution is stored at a small number of fixed time levels (solution snapshots) over one time period of the unsteady flow in a single blade passage in each blade row; thus the solution is periodic by construction. The individual time levels are coupled to one another through a spectral operator representing the time derivative term in the Navier-Stokes equation, and at the boundaries of the computational domain through the application of periodic and nonreflecting boundary conditions.

Applications

The harmonic balance method allows for the rapid computation of unsteady flow solutions that are periodic in time. As a result, this method is useful for a wide variety of applications, including:

- Turbomachinery
- Tonal aeroacoustics
- Flutter and forced response

- Rotorcraft

In addition to quickly computing the periodic unsteady solution, the harmonic balance method can also account for inter-blade phase angle effects. This additional benefit means the harmonic balance method is well-suited for turbomachinery simulation. With the harmonic balance method, it is possible to represent how the solution varies from one blade to the next within a given row of a turbomachine while only meshing a single blade in that row. The need to mesh only one blade in each row dramatically reduces computational cost.

Benefits

The main benefit of the Harmonic Balance method is that you can solve unsteady flow problems using steady flow solver techniques. This means that the method is often faster than conventional time-accurate time-marching approaches.

Limitations

The method is designed to work with temporally periodic flows, or even some “nearly” periodic flows. It is not suited for transient problems that are non-periodic in time.

Currently, K-Epsilon, K-Omega, and Spalart-Allmaras turbulence models are available for use with Harmonic Balance. However, the setup and use of these turbulence models are identical to how the models would be used with conventional unsteady calculations.

The porous baffle, fan, and fully developed interfaces are not available with Harmonic Balance.

Working with the Harmonic Balance Method in STAR-CCM+

The following sections provide information on the Harmonic Balance method in STAR-CCM+.

Contents:

Setting Up a Harmonic Balance Simulation
What Is the Harmonic Balance Model?
What Is the Harmonic Balance Flow and Energy Model?
What Are the Harmonic Balance Turbulence Models?
Defining a Wake
What Is the Harmonic Balance Flutter Motion?
What Is the Implicit Harmonic Balance Solver?
Post-Processing Harmonic Balance Solution Data
What Is the HB Solution View?
Filtered Mode Field Functions
Harmonic Balance Guidelines
Harmonic Balance Formulation
Harmonic Balance Field Functions Reference

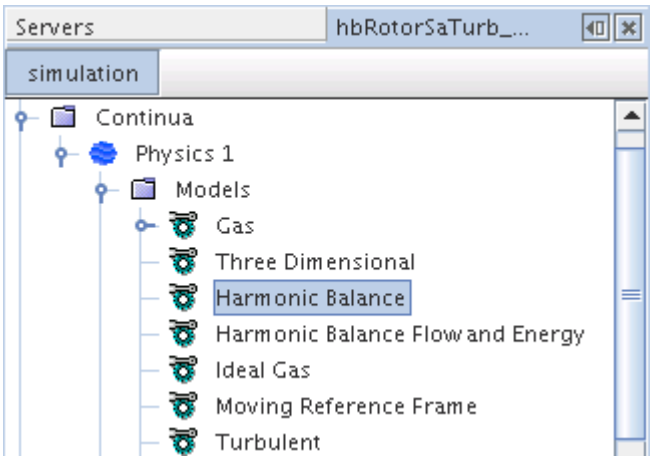
Setting Up a Harmonic Balance Simulation

The following steps outline the actions that are required to set up a simulation with the Harmonic Balance model:

- Activating the harmonic balance model
- Activating the harmonic balance flow and energy model
  - Working with Blade Rows
  - Working with Mode Tables
- Activating the Harmonic Balance Turbulence Models
- Defining Harmonic Balance Boundary Conditions
- Setting Up a Simulation with Harmonic Balance Flutter Motion
- Post-Processing Harmonic Balance Solution Data

What Is the Harmonic Balance Model?

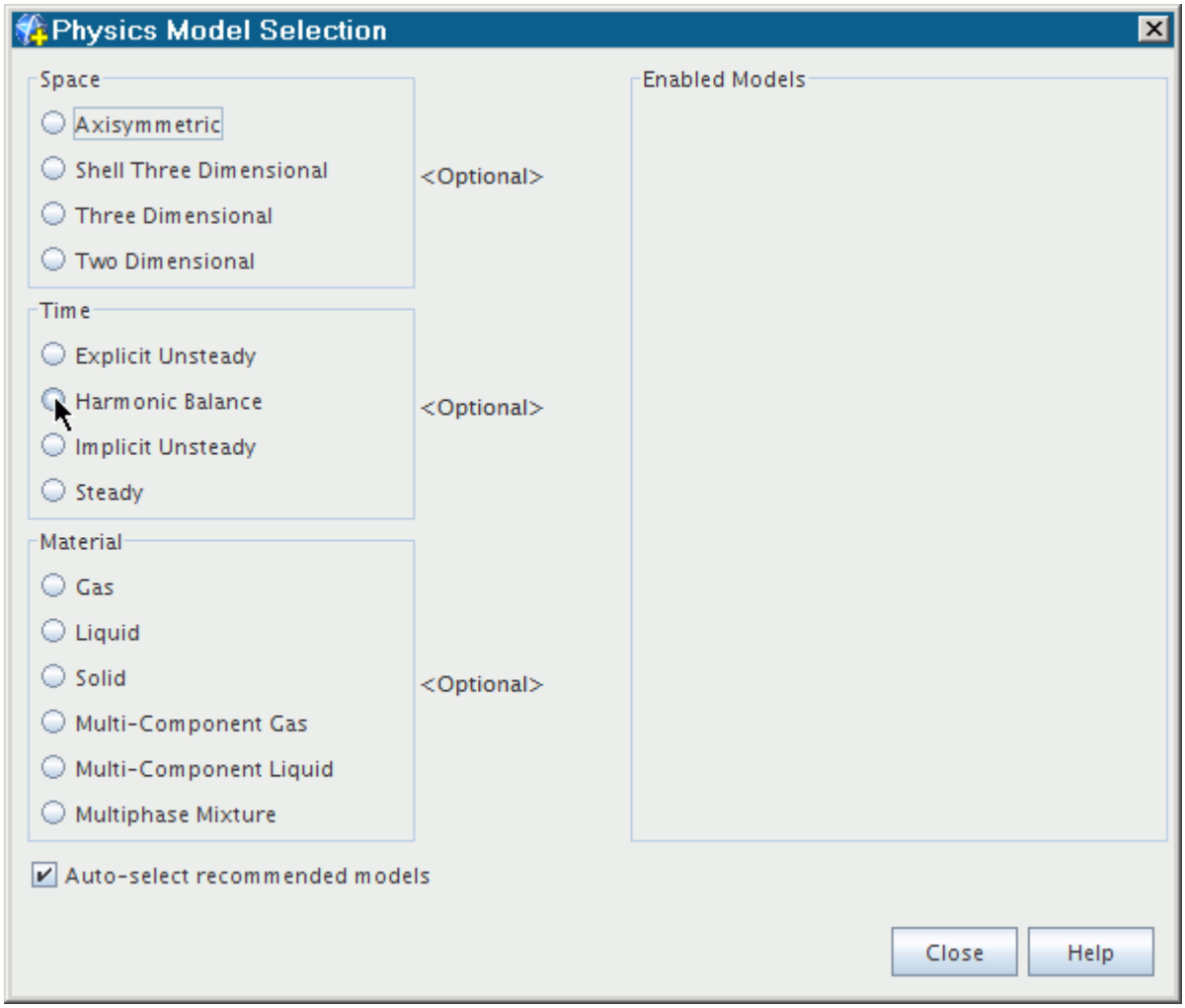
The Harmonic Balance model is represented in the object tree by the **Harmonic Balance** node, and is activated in the physics models selection dialog.



Activating the Harmonic Balance Model

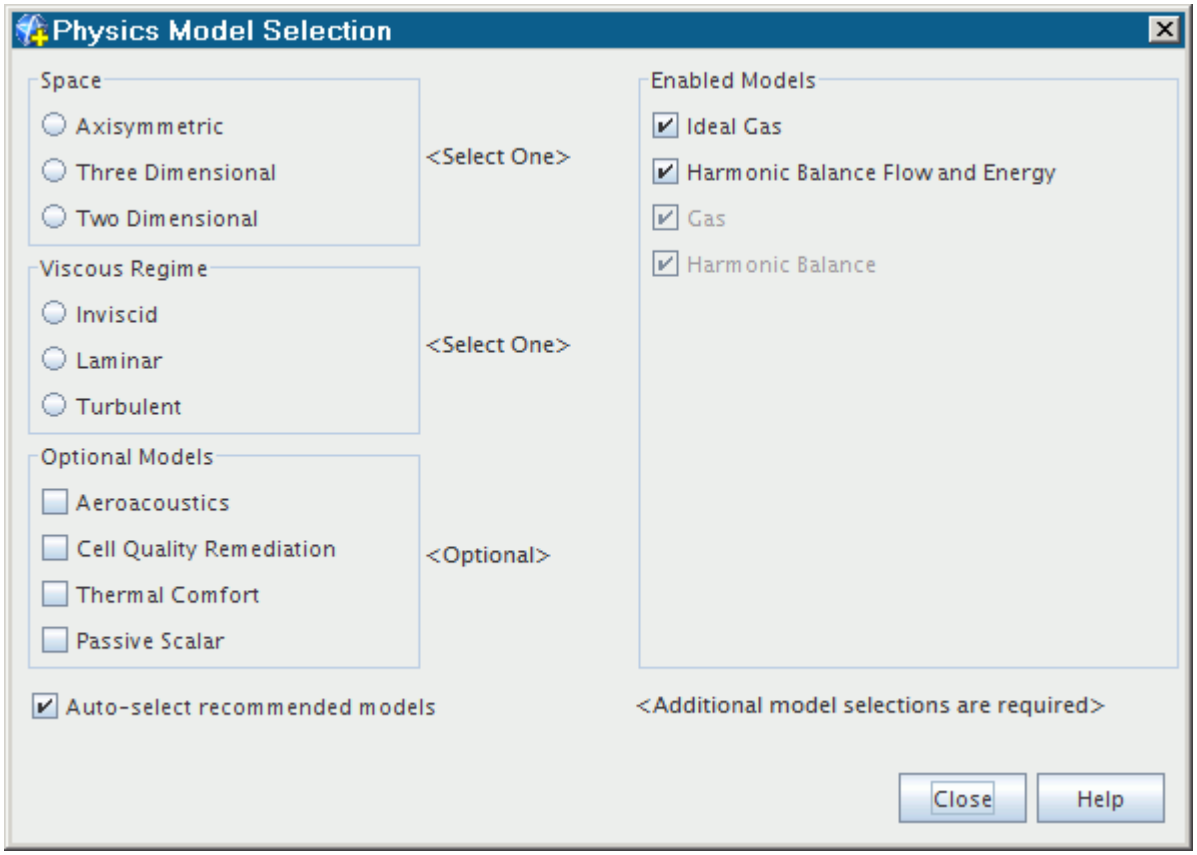
To activate the Harmonic Balance model:

- In the *Physics Model Selection* dialog, select the **Harmonic Balance** model in the *Time* group.



If the **Auto-select recommended models** checkbox is ticked, the Harmonic Balance Flow and Energy model is selected automatically.

When the Harmonic Balance model is selected, the standard Coupled Flow or Segregated Flow models cannot be selected.



The Implicit Harmonic Balance Solver is a coupled solver that couples the conservation of mass, momentum, and energy at all the time levels. Time levels are points in time, which are chosen by the solver, when a solution is computed.

The Harmonic Balance model works only with the following models within their respective categories:

- The Harmonic Balance *Time* model
- The Gas *Material* model
- The Ideal Gas *Equation of State* model
- The RANS K-Epsilon, K-Omega, and Spalart-Allmaras *Reynolds-Averaged Turbulence* models

From the other model groups, the Harmonic Balance method works with:

- Two-dimensional or three-dimensional models
- Stationary motion
- Reference Frame objects
- Inviscid, Laminar, or Turbulent viscous regimes
- The Passive Scalar model

  When STAR-CCM+ uses the Passive Scalar model with Harmonic Balance, it treats the passive scalars as steady and uses the time-mean harmonic balance solution (that is, quantities such as time-mean density, time-mean velocities, or time-mean turbulent viscosity) to solve the passive scalar transport equations. In other words, the passive scalars are transported within the solution domain using the time-mean flow field. Using the time-mean also implies that the passive scalar solution is unchanging over time.

You can also simulate blade vibration by selecting the Harmonic Balance Flutter motion instead of Stationary.

See Also:

What Is the Harmonic Balance Flow and Energy Model?

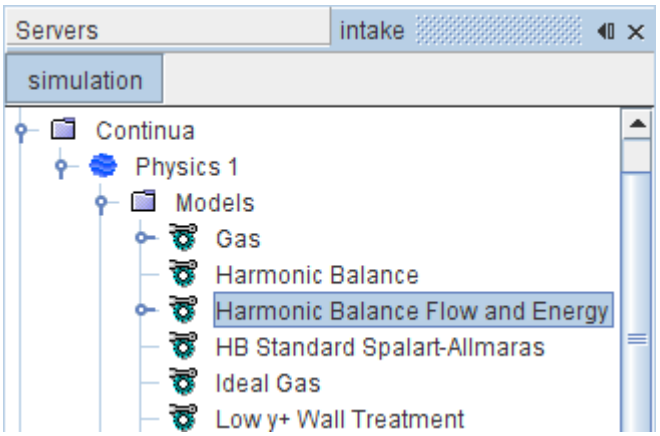
What Is the Implicit Harmonic Balance Solver?

What Is the Harmonic Balance Flow and Energy Model?

The Harmonic Balance Flow and Energy model operates on similar principles as the Coupled Flow and Coupled Energy models, and uses the Implicit Harmonic Balance Solver solver.

The Harmonic Balance Flow and Energy model is controlled by properties similar to the Coupled Flow model properties, including discretization schemes and secondary gradients.

The **Harmonic Balance Flow and Energy** node represents this model in the object tree. This node includes the objects for working with blade rows, and working with mode tables.



Once you have activated this model, you can create blade rows. Using multiple blade rows in a simulation requires the use of user-specified or default filtered Mode Tables.

A blade row represents a region or a set of regions that models one or more blade passages, such as that of a stator or a rotor of a turbomachine, with a fixed rotation rate and number of blades.

After creating the blade rows, and initializing the solution, the model computes a mode table. This table tabulates all the unsteady frequencies and nodal diameters to be considered in the simulation for every unsteady mode.

Harmonic Balance Flow and Energy Properties

Discretization	Selects the discretization scheme that is used.	
	1st-order	Selects the first-order upwind discretization scheme
	2nd-order	Selects the second-order upwind discretization scheme.

Expert Properties

Unless you are thoroughly familiar with the theoretical aspects of this model, and the discretization techniques that are used in STAR-CCM+, we recommend that you do not change any properties within the *Expert*

category. The values in that category reflect both the model's design and discretization approaches that have been optimized for accuracy and performance.

Secondary Gradients	Neglect or include the boundary secondary gradients for diffusion and/or the interior sec ondary gradients at mesh faces.	
	On	Include both secondary gradients.
	Off	Exclude both secondary gradients.
	Interior Only	Include the interior secondary gradients only.
	Boundaries Only	Include the boundary secondary gradients only.
Positivity Rate Limit	Maximum rate at which temperature is allowed to decrease.	
Minimum Absolute Pressure	Minimum allowable absolute pressure.	
Explicit Relaxation	A scaling factor that is used to relax all corrections explicitly before they are applied to the variables.	
Preconditioning Enabled	Enable/disable preconditioning.	
	Activated	Enable preconditioning.
	Deactivated	Disable preconditioning.
Flow Boundary Diffusion	Enable/disable the flow-boundary diffusion fluxes (or viscous fluxes for flow models)—see Eqn. (1593). This property is enabled by default.	
	Activated	Enable flow-boundary diffusion.
	Deactivated	Disable flow-boundary diffusion.

Activating the Harmonic Balance Flow and Energy Model

The **Harmonic Balance Flow and Energy** model is selected automatically when activating the harmonic balance (time) model.

Contents:

Working with Blade Rows
Working with Mode Tables
Time Varying Option

See Also:

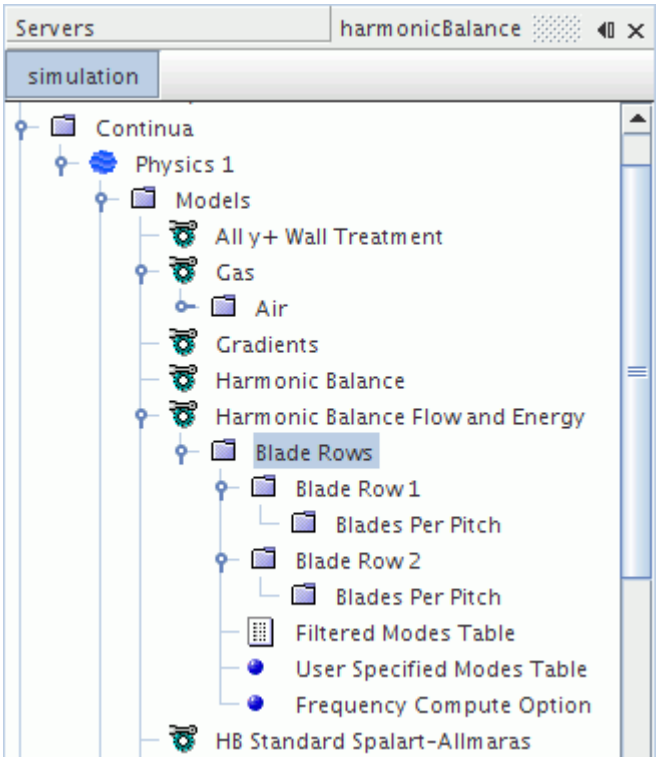
Modeling Flow and Energy
What Is the Harmonic Balance Model?
Creating Blade Rows
What Is the Harmonic Balance Model?
Defining a Wake
What Is the Wake Specification?

Harmonic Balance Field Functions Reference

Working with Blade Rows

You can model one or more blade rows in a Harmonic Balance simulation. First, you define blade rows in the **Blade Rows** manager that is located within the **Harmonic Balance Flow and Energy** node.

After this step, you associate each blade row with its corresponding region in the simulation. The **Blade Row Specification** property on a region is used to make this connection.



Contents:

What Is a Blade Row?
Creating Blade Rows
Setting Blades and Pitches
Defining the Region for a Blade Row
Choosing a Fictitious Rotation Rate Factor

What Is a Blade Row?

A blade row represents a region or a set of regions that models a fixed number of blades within a specified pitch.

The **Blade Rows** manager node contains all the individual blade row nodes, the **Filtered Modes Table** node, and the **User Specified Modes Table** node. This node is controlled with properties, and supports node actions.

Blade Rows Manager Pop-Up Menu

Use this pop-up menu to add blade row objects to the simulation.

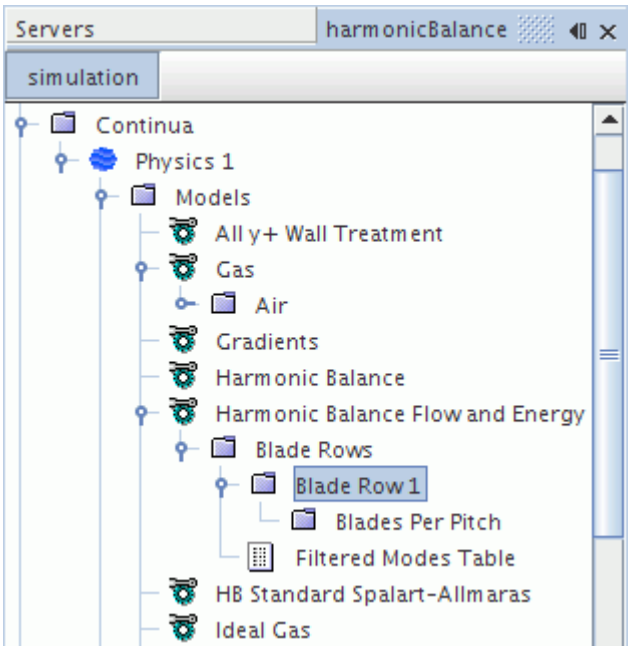
New	Adds a blade row to the simulation.

Blade Rows Manager Expert Properties

Frequency Compute Option	Specifies whether blade row frequencies are computed using disturbance from neighboring blade rows, or from all blade rows. See Blade Row Frequencies Computation. See also User Specified Mode Tables.	
	Consider Only Neighbors	Consider disturbances from neighboring rows only.
	Consider All Blade Rows	Consider disturbances from all blade rows. The relative rotation rate between any two blade rows must be non-zero. This is insured by the **Fictitious Rotation Rate Factor**.
Modes Table Filter Option	Specifies how to compute the maxmum allowable frequency $\omega_{\max}$ for ω^j Eqn. (1023), and the maximum allowable nodal diameter $N_{\max}$ for N^j Eqn. (1025), across all blade rows.	
	Cut-Off	Inserts a **Cut-Off Filter** node under the **Blade Rows** manager node. Use the *Cut-Off Factor* property of this node to set the cut-off factor γ for the maxima $\omega_{\max}$ Eqn. (1024) and $N_{\max}$ Eqn. (1026).
	Maximum Limits	Inserts a **Maximum Limits Filter** node under the **Blade Rows** manager node. Use the *Maximum Nodal Diameter* and *Maximum Frequency* properties of this node to set the maxima $\omega_{\max}$ and $N_{\max}$.
Fictitious Rotation Rate Factor	Specifies the proportionality constant for the offset applied to blade frequencies. The default value is 0.001. See Choosing a Fictitious Rotation Rate Factor and Fictitious Rotation Rate Computation.	

Blade Row Nodes

A **Blade Row** node represents a separate blade row in the simulation. This node is controlled with properties, and is subject to common menu items. Blade row objects can be created, renamed, and deleted.



In order to specify the number of blades, the Blades Per Pitch child node is provided under the **Blade Row** node.

Blade Row Properties

Index	The sequential number of the addition of this blade row to the simulation, beginning at 1 (read-only).
Modes	The maximum value that a modal index can take in this blade row, in the expressions for frequencies (Eqn. (1023)) and nodal diameter (Eqn. (1025)). This value subsequently determines how many unsteady frequencies are considered in this blade row. When you change the number of modes in any of the blade rows in the simulation, STAR-CCM+ reinitializes the new set of time-level solutions in all the blade rows. This reinitialization retains harmonic solutions at all the existing nodes.
User Specified Modes	A table column number. If a user-specified table is used as the modes table, then this field allows the user to select the column in the table that corresponds to the modal index of the current blade row. If there are multiple blade rows in the simulation, this property must be defined for all blade rows, otherwise the default filtered modes table is used.
Time-levels	The number of unique time instants that are used to sample the solution in the blade row (read-only).

Number of Frequencies	The number of unique frequencies in the blade row (read-only).
Regions	The regions in the current blade row (read-only).
Interfaces	The interfaces in the current blade row (read-only).
Unique Positive Frequences	A list of the unique computed frequencies for the blade row (read only).

Blade Row Expert Properties

Maximum Nodal Diameter	This property sets the maximum allowable nodal diameter, which limits the modes included in the default filtered modes table.
Maximum Frequency	The maximum allowable frequency for the current blade row, which limits the modes included in the filtered modes table.

See Also:

Fictitious Rotation Rate Computation

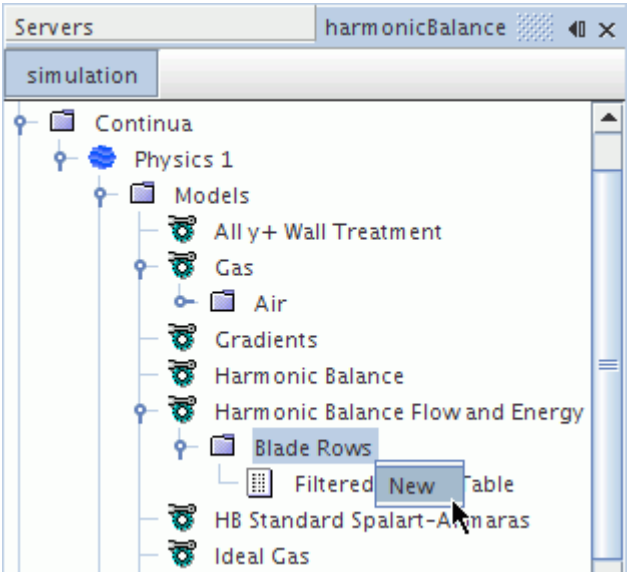
Blade Row Frequencies Computation

What Is the HB Solution View?

Creating Blade Rows

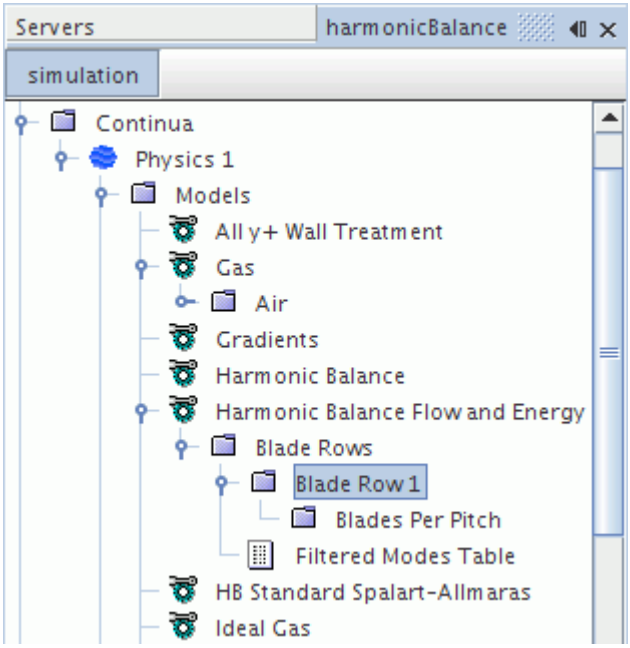
To create a blade row:

- Right-click the **Harmonic Balance Flow and Energy** > **Blade Rows** node, and select **New** from the pop-up menu.



A new blade row appears under the **Blade Rows** node. The default name **Blade Row** is concatenated with a number to ensure that the name is unique.

A child node **Blades Per Pitch** is created under the **Blade Row** node.



The properties of the blade row object include the ability to set the number of modes.

See Also:

[What Is the Harmonic Balance Flow and Energy Model?]

Setting Blades and Pitches

The properties of the **Blades Per Pitch** node are used to specify the *Pitch* P_{user} and *Number of Blades* B_{user}.

With these inputs, STAR-CCM+ computes the number of blades:

$$B = \frac{2\pi B_{\text{user}}}{P_{\text{user}}} \tag{1003}$$

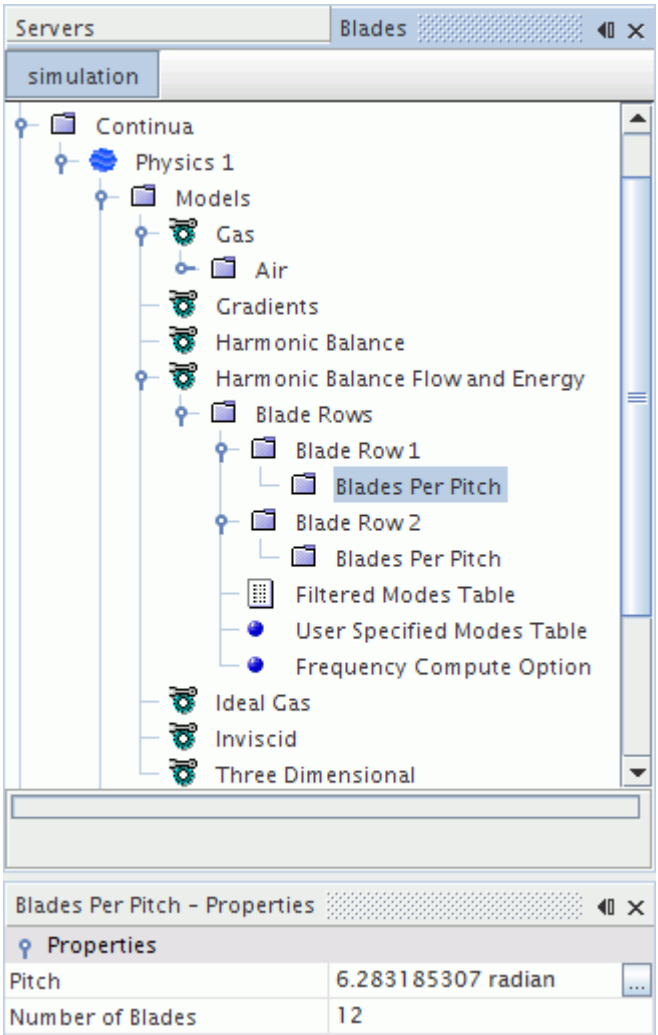
This number is used in the computation of the nodal diameters, and frequencies that occur in the blade row.

The *Number of Blades* B_{user} refers to the number of blades that are contained within the *Pitch* P_{user}.

Three-Dimensions

In three-dimensional problems, the *Pitch* P_{user} is best left at its default value of 2π radians, and the *Number of Blades* B_{user} is specified as the real number of blades on the full wheel.

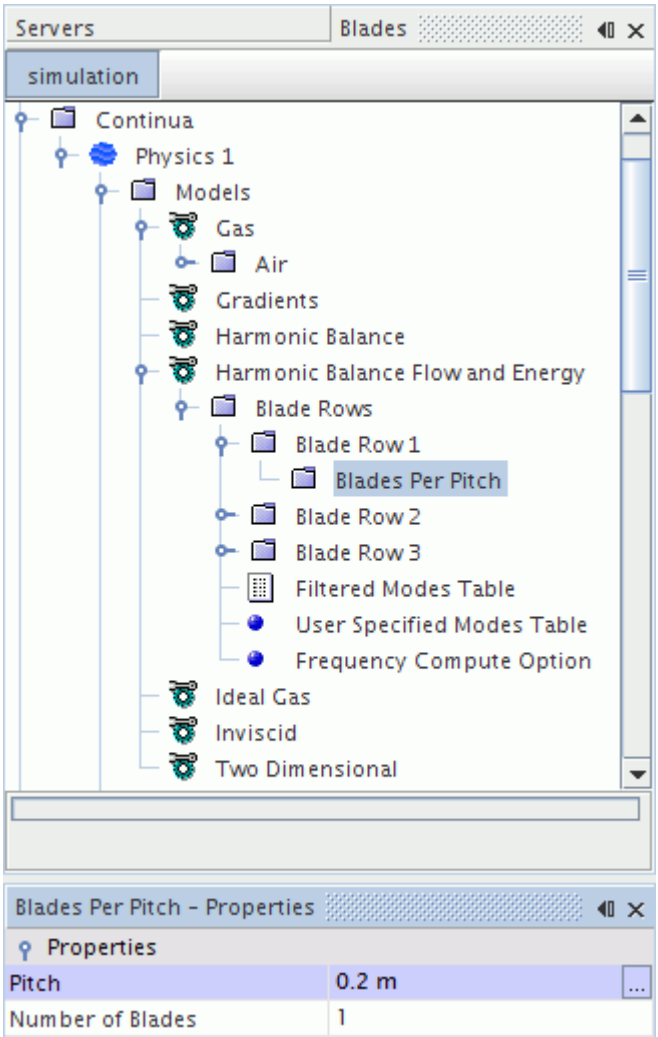
For example, if there are 12 blades on the full wheel, specify the *Pitch* as 2π radians, and the *Number of Blades* as 12 so that $B = 12$. If the *Pitch* is specified as 2π radians, the *Number of Blades* should always be equal to the real number of blades on the wheel, irrespective of the number of blade passages that are modeled in the geometry.



Two-Dimensions

In two-dimensions, the *Number of Blades* B_{user} is best left at its default value of 1. The *Pitch* P_{user} is taken as the distance between one blade and its neighboring blade, or the distance between the upper and lower periodic boundaries. This value is specified in units of length. This distance could be inferred from the **Periodic Transformation** node of the periodic interface.

If more than one blade is modeled within the geometry, the *Number of Blades* is equal to the number of blade passages modeled.



For example, if only one blade passage is modeled with a pitch of 0.2 m (which could be the value that is shown in the **Periodic Transformation** node), you would specify the *Pitch* as 0.2 m, and the *Number of Blades* as 1.

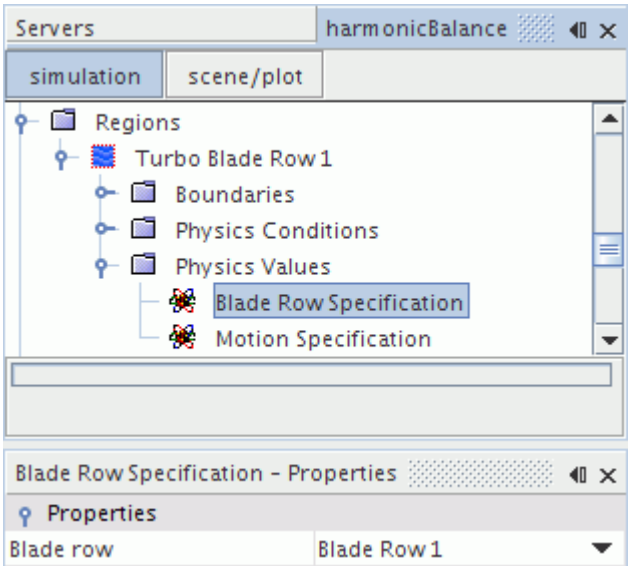
However, if two blade passages are modeled in the geometry with the same pitch, then you would specify the *Pitch* as 0.2 m, and the *Number of Blades* as 2.

Blades Per Pitch Properties

Pitch	Specifies the pitch in angular units in three-dimensions and in length units in two-dimensions (linear length in Y direction).
Number of Blades	Specifies the number of blades within the given pitch. The number of blades should be an integer.

Defining the Region for a Blade Row

When you are done with defining the blade rows in the **Blade Rows** manager of the physics continuum, you must associate these rows with the regions that represent them. The **Blade Row Specification node**, that is located within the **Physics Values** folder for the region, is used to make this connection.



Blade Row Specification Properties

Blade row	Assigns a blade row to a region selected from the drop-down menu.

Choosing a Fictitious Rotation Rate Factor

When you use **Consider All Blade Rows** as your *Frequency Compute Option*, every blade row must be assigned a different frequency for the purposes of computation.

This rule remains true even if some blade rows have identical frequencies in the **User Specified Mode Table**. Every blade is assigned an offset f proportional to the *Fictitious Rotation Rate Factor* ε and to the maximum user-specified rotation rate $\Omega_{\max}$.

The computation works up the blade rows, from last to first. The first time it comes to a blade that has the same frequency as a previous blade row, it adds an offset of f. (The last row is compared to the first.) The next time that it comes to a blade that repeats a former frequency, it adds $2f$, and the next time $3f$, and so on. This addition guarantees that no two blades have identical frequencies.

For example, consider the blade rows described in the following table:

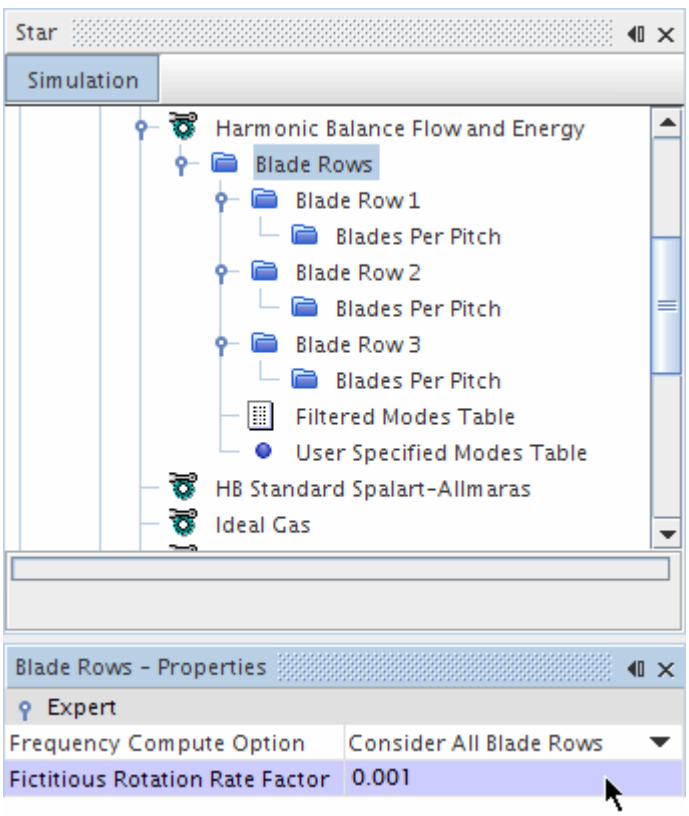
	1st Stator (1st Blade Row)	1st Rotor (2nd Blade Row)	2nd Stator (3rd Blade Row)	2nd Rotor (4th Blade Row)	3rd Stator (5th Blade Row)
User-Specified Rotation Rate	0	Ω	0	Ω	0
New Rotation Rate	0	Ω	$3f$	$\Omega + 2f$	f

The computation begins from the last blade row (that is, the downstream stator, the third) and works backwards, adding a frequency offset if desired. The computation first adds an offset of f to the downstream stator blade row (that is, the fifth blade row) because the relative rotation rate between this stator and the first stator (and also the third stator) is zero.

When the computation reaches the second rotor (fourth blade row), it adds an offset of $2f$ because the relative rotation rate between this rotor and the first rotor (the second blade row) is zero.

When the computation reaches the middle stator (the third row), it adds an offset of $3f$ because the relative rotation rate between this stator and the first stator is zero.

To set the **Fictitious Rotation Rate** factor, select the **Blade Rows** node, and set the value for the *Fictitious Rotation Rate* property.



The default value of f is 0.001. It is best to make the value small, so the departure from the original value stays small. See Fictitious Rotation Rate Computation.

The offset f is used only for entries in the modes table, not in computing the flow solution. The use of f is required when using the **Consider All Blade Rows** option under the **Frequency Compute Option**. Using f ensures that the unsteady solution at higher harmonics is exchanged between any two blade rows with the same rotation rate. The **Fictitious Rotation Rate** factor has no influence on the solution when the **Consider Only Neighbors** option is used.

Working with Mode Tables

For a single blade row, modes consist of the fundamental frequency and the harmonics, that is, integral multiples of the fundamental frequency.

For simulations with multiple blade rows, modes consist of the fundamental frequencies of each blade row interacting with neighboring blade rows, and the harmonics that result.

Mode tables are used to filter the modes included in the simulation when obtaining a solution. There are two methods of defining a mode table:

- the default filtered modes table that is calculated by the code, and
- a user-specified mode table.

Contents:

Filtered Modes Table
User Specified Mode Tables

See Also:

Fluttering Motion of a Rotor

Filtered Modes Table

The filtered modes table is created by default when the solution is initialized, and defines the modes that are included in the simulation.

The *Maximum Nodal Diameter* and *Maximum Frequency* expert properties define the inclusion of modes in the table. The table consists of the following types of columns:

- m1, m2... : Modal indices for blade rows
- w1 [Row *n*], w2 [Row *n*]... : Modal indices for the wake blade row that is attached to modeled blade row *n*.

  A wake blade row indicates a wake boundary condition that is applied to a stagnation inlet or a pressure outlet boundary. This boundary condition represents an external spatial disturbance that is applied to a modeled blade row.
- N: Nodal diameter
- omega1, omega2... : Frequencies for blade rows
- omegaw1 [Row *n*], omegaw2 [Row *n*]... : Frequencies for the wake blade row that is attached to the modeled blade row *n.*

Using the tabulating feature, the information in the table can be displayed. For example:

Tabular Data

	m1	m2	N	omega1	omega2
0	-1	-1	-84	960	-720
1	-1	0	-36	0	-720
2	-1	1	12	-960	-720
3	0	-1	-48	960	0
4	0	0	0	0	0
5	0	1	48	-960	0
6	1	-1	-12	960	720
7	1	0	36	0	720
8	1	1	84	-960	720

Export... Close

In the above example, there are two blade rows and a wake blade row with the following properties:

	Wake Blade Row	Blade Row 1	Blade Row 2
Number of blades	10	12	18
Rotation rate (radians/s)	0	10	0
Number of modes	1	1	1

The nodal diameter is:

$$N = \sum_{i=1}^{n} m_i B_i \tag{1004}$$

where n is the number of blade rows, m_i are the nodal indices, and B_i the number of blades. STAR-CCM+ calculates the frequencies from the rotation rate for each blade, the effects of rotation from neighboring blades, and the disturbances coming from upstream and downstream.

The following illustrates how the *Maximum Nodal Diameter* properties can affect the filtered nodes table. Using the example above, if the *Maximum Nodal Diameter* is set to 25, the following table is created.

Tabular Data

	m1	m2	N	omega1	omega2
0	-1	0	-36	0	-720
1	-1	1	12	-960	-720
2	0	-1	-48	960	0
3	0	0	0	0	0
4	0	1	48	-960	0
5	1	-1	-12	960	720
6	1	0	36	0	720

Export... Close

All modes where N > 25 have been removed from the table. Definition of the *Maximum Frequency* property has a similar effect, with respect to *omega1* and *omega2*.

User Specified Mode Tables

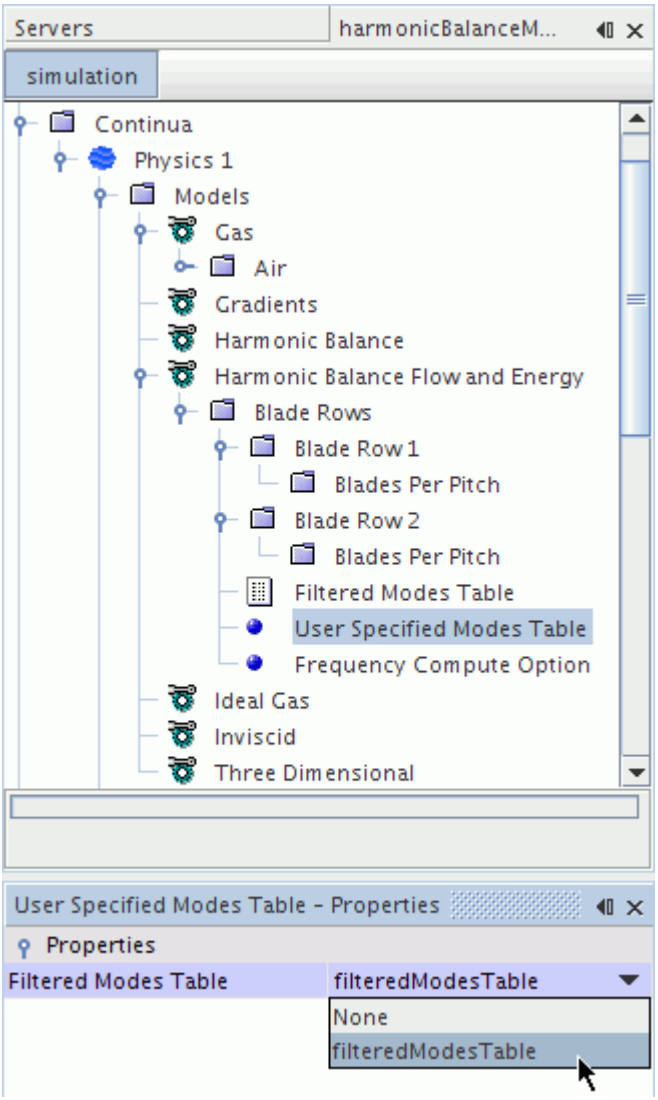
This feature allows you to define a custom mode table for use within the simulation. Implementing a user-specified mode table overrides the information in the default filtered mode table.

Use the user specified mode table option to retain only a specific set of modes needed to capture the significant unsteady effects. To do this, first exporting the default filtered modes table and then removing the entries (the rows) that are not desired.

The user-specified mode table should only contain information on blade row modal indices (*m1* and *m2*) as the nodal diameter (*N*), and blade row frequencies (*omega1* and *omega2*) are defined by existing simulation properties.

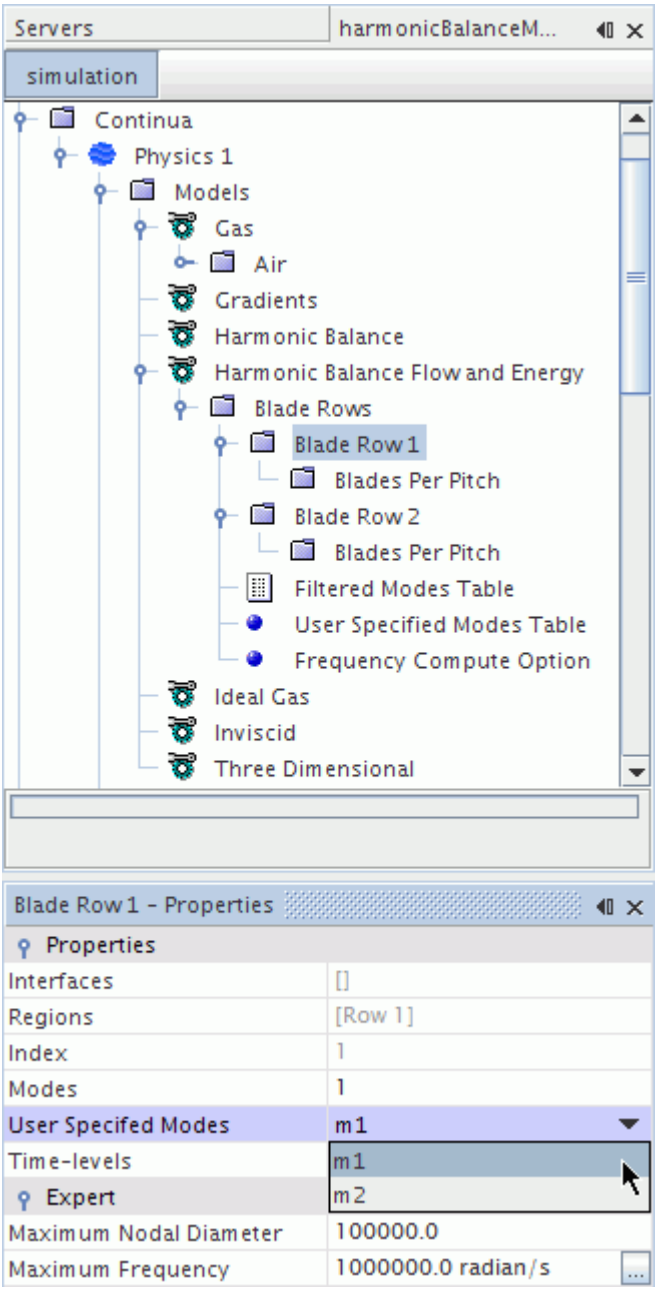
To define a user-specified mode table, first import the table into STAR-CCM+, and then apply it to the physics model as follows:

1. Select the **Physics** > **Models** > **Harmonic Balance Flow and Energy** > **Blade Rows** > **User Specified Mode Table** node.
2. In the *Properties* window, set the *Filtered Modes Table* property to the required table in the drop-down menu.



Then, match the table column containing the modal indices to each blade row:

3. Select the **Blade Row** node under the **Blade Rows** manager.
4. In the *Properties* window, set the *User Specified Modes* property to the corresponding column in the table.



Time Varying Option

On stagnation inlet and pressure outlet boundaries, you can specify field functions that define time-dependent flow quantities.

For a stagnation inlet boundary, you can specify field functions using `$Time` to vary the quantities Total Pressure, Total Temperature, and Flow Angles. For a pressure outlet boundary, you can vary Static Pressure with time using field functions.

This option is available only for boundaries of type **Stagnation Inlet** or **Pressure Outlet**. The **Time Varying Option Parameters** node appears under **Physical Values** when either of these boundary types is selected.

Time Varying Option Properties

Frequency	The fundamental frequency of variation in the boundary conditions, in radians/s. The default is 1.0 radians/s. **Note:** If there is a discrepancy between this value and the actual frequency with which the boundary condition varies over time, then no unsteady component is computed from the boundary condition.
Modes	The number of modes (harmonics) to be computed from the time varying boundary condition. The default is 1.
Phase Angle	The phase lag, in radians, of the time varying boundary condition with respect to the blade row for that boundary condition. The default is 0.

What Are the Harmonic Balance Turbulence Models?

These variants differ from their standard counterparts only in that they use the technique of the coupled Implicit Harmonic Balance Solver.

Harmonic Balance Turbulence Model		Standard Model
	K-Epsilon Models	
HB AKN K-Epsilon Low-Re		AKN K-Epsilon Low-Re
HB Realizable K-Epsilon		Realizable K-Epsilon
HB Realizable Two-Layer K-Epsilon		Realizable Two-Layer K-Epsilon
HB Standard K-Epsilon		Standard K-Epsilon
HB Standard K-Epsilon Low-Re		Standard K-Epsilon Low-Re
HB Standard K-Epsilon Two-Layer		Standard K-Epsilon Two-Layer
HB V2F K-Epsilon		V2F K-Epsilon
	K-Omega Models	
HB Standard K-Omega		Standard K-Omega
HB SST (Menter) K-Omega		SST K-Omega
	Spalart-Allmaras Models	
HB Standard Spalart-Allmaras		Standard Spalart-Allmaras
HB High-Reynolds Number Spalart-Allmaras		High-Reynolds Number Spalart-Allmaras

Contents:

Activating the Harmonic Balance Turbulence Models

See Also:

What Is the Implicit Harmonic Balance Solver?

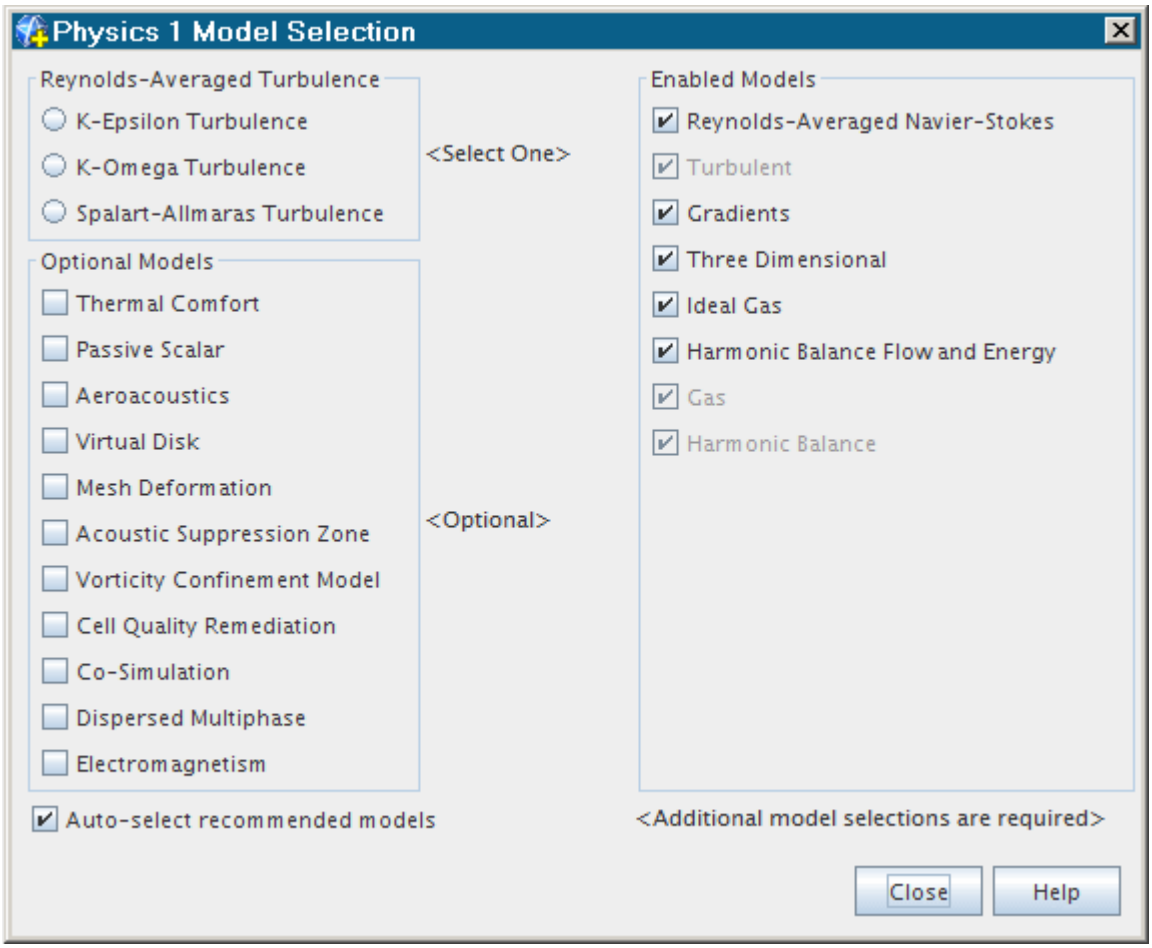
Activating the Harmonic Balance Turbulence Models

To use turbulence with the Harmonic Balance method:

1. Right-click the **Continua** > **Physics 1** > **Models** node and choose **Select models...**
2. Select the appropriate models. See Activating the Harmonic Balance Model.
3. Select **Turbulent** in the *Viscous Regime* group.

   Three types of turbulence, K-Epsilon, K-Omega, and Spalart-Allmaras, become available in the model selection dialog (*Physics 1 Model Selection* in the following screenshot), if you have the **Auto-select recommended models** checkbox activated.

Choose between the K-Epsilon, K-Omega, and Spalart-Allmaras models. Variants of each type of turbulence exist for use with Harmonic Balance.



- To use Realizable Two-Layer K-Epsilon:
  - Select **K-Epsilon Turbulence** in the *Reynolds-Averaged Turbulence* group.

    The HB Realizable Two-Layer K-Epsilon model is selected automatically.
- To use any of the other K-Epsilon models:
  - Deactivate the **Auto-select recommended models** checkbox.
  - Select **K-Epsilon Turbulence** in the *Reynolds-Averaged Turbulence* group.

    The following K-Epsilon models become available:

    - HB AKN K-Epsilon Low-Re
    - HB Realizable K-Epsilon
    - HB Realizable K-Epsilon Two-Layer
    - HB Standard K-Epsilon
    - HB Standard K-Epsilon Low-Re
    - HB Standard K-Epsilon Two-Layer

        - HB V2F K-Epsilon
- To use either of the K-Omega models:
    - Deactivate the **Auto-select recommended models** checkbox.
    - Select **K-Omega Turbulence** in the *Reynolds-Averaged Turbulence* group.

      The following K-Omega models become available:

        - HB Standard (Wilcox) K-Omega
        - HB SST (Menter) K-Omega
- To use Standard Spalart-Allmaras:
    - Select **Spalart-Allmaras** in the *Reynolds-Averaged Turbulence* group.
    - The Harmonic Balance Standard Spalart-Allmaras turbulence model is selected automatically.
- To use High-Reynolds Number Spalart-Allmaras:
    - Deactivate the **Auto-select recommended models** checkbox.
    - Select **Spalart-Allmaras** in the *Reynolds-Averaged Turbulence* group.

      Both Harmonic Balance Spalart-Allmaras models, High-Reynolds Number and Standard, become available for selection.

Defining a Wake

The wake is an unsteady boundary condition that is specified on a stagnation inlet or a pressure outlet boundary. The steady state values for pressure, temperature, and velocity are specified separately by specifying the total pressure, total temperature, and flow angle for the stagnation inlet boundary, and the static pressure and static temperature for the pressure outlet boundary.

The wake specification adds the unsteady components of these variables, based on pre-specified, or user-specified profile functions. STAR-CCM+ automatically computes and extracts the unsteady part of the user-specified wake data and discards the mean values.

The following boundary conditions are available for the Harmonic Balance Flow and Energy Model:

- Non-Reflecting Option
- Wake Specification

These boundary conditions are in addition to the ones for the steady part of the solution, which you control by specifying flow direction, total pressure and total temperature.

Contents:

What Is the Non-Reflecting Option?
What Is the Wake Specification?

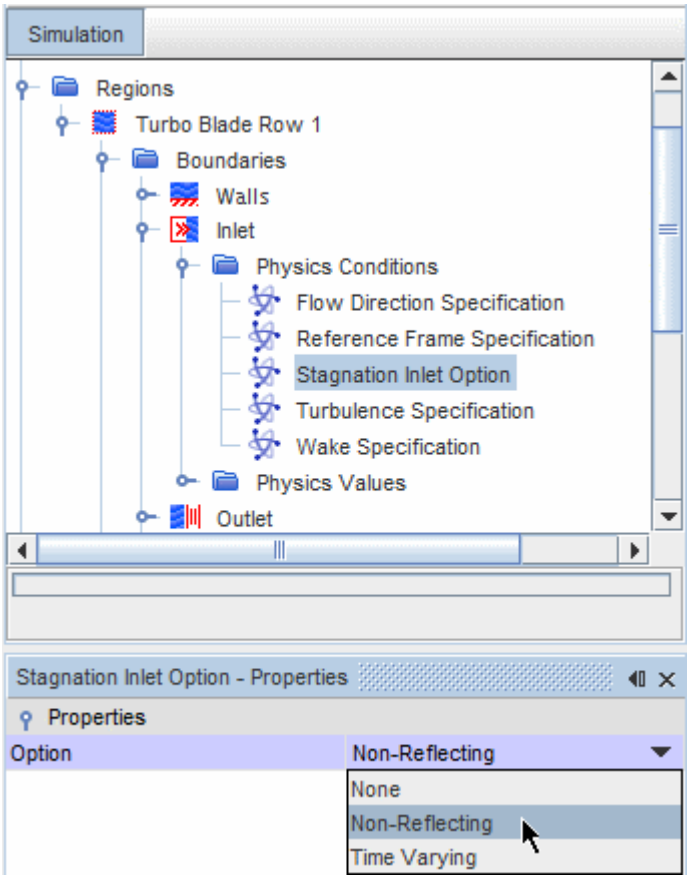
See Also:

What Is the Harmonic Balance Flow and Energy Model?

What Is the Non-Reflecting Option?

This boundary condition is the unsteady version of the Non-Reflecting Option for the Coupled Flow model.

Selecting this option makes the Wake Specification boundary condition available, which has seven different options for specifying a wake.



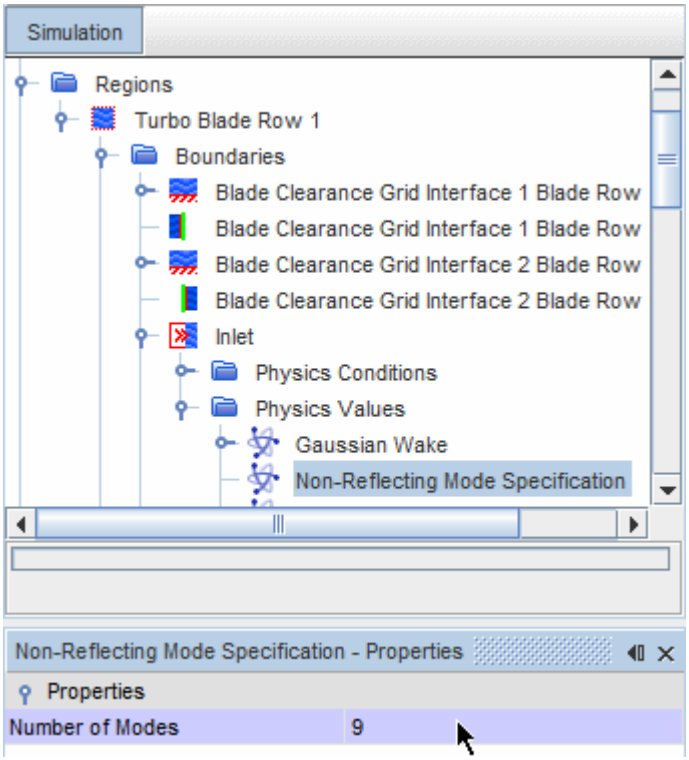
Non-Reflecting Option Properties

Non-Reflecting Option	Specifies whether to use the non-reflecting option.	
	Activated	Use the non-reflecting option.
	Deactivated	Do not use the non-reflecting option.

What Is the Non-Reflecting Mode Specification?

This boundary value is activated automatically when the Non-Reflecting Option boundary condition is activated and is the multiple frequency version of the Non-Reflecting Boundary Condition for the Coupled Flow model. This boundary condition prevents spurious numerical reflection of the solution into the solution domain.

You can specify the number of modes to retain using the *Number of modes* property. Specify a number of modes less than the number of cells in the circumferential direction.



Properties

Number of modes	Specifies number of modes to retain for the non-reflecting boundary condition. Specify a number of modes less than the number of cell faces in the circumferential direction.

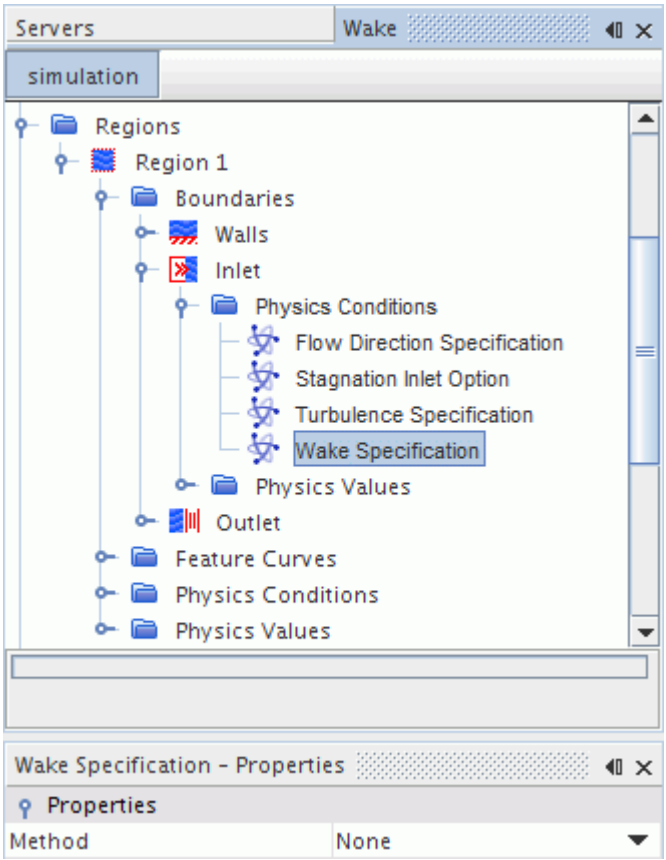
See Also:

What Is the Wake Specification?

What Is the Wake Specification?

This node allows you to determine how the wake for a boundary is specified.

It is activated when the Harmonic Balance Flow and Energy Model model and the Non-Reflecting Option boundary condition are enabled.



The Sine, Gaussian, and Hodson options are standard function shapes that are chiefly for academic use, while the other options are used more commonly for applications.

The wake can only be specified at the inlet boundary or the outlet boundary. However, the STAR-CCM+ GUI does not prevent you from entering values for the wake at both the inlet and outlet boundaries.

Properties

Method	Selects the method for specifying the wake, either one of the standard shapes, or one based on user table data.	
	None	The option is turned off for the boundary. No nodes are added to the **Physics Values** node.
	Sine	This option uses a standard sine curve for unsteadiness. Selecting it adds a Sine Wake node to the **Physics Values** node.
	Gaussian	This option uses a standard Gaussian shape for unsteadiness. Selecting it adds a Gaussian Wake node to the **Physics Values** node.

	Hodson	This option uses a standard Hodson shape for unsteadiness. Selecting it adds a Hodson Wake node to the **Physics Values** node.
	Fourier Data	This option lets you use Fourier data to add unsteady pressure, temperature, and velocity. Selecting it adds a Wake Fourier Coefficients node to the **Physics Values** node.
	Spatial Annular Data	This option works in cylindrical space (r, θ) , or (r, θ, z) for three-dimensional cases, to add unsteady pressure, temperature, and velocity. Selecting it adds a Spatial Annular Wake Data node to the **Physics Values** node.
	Spatial Linear Data	This option works in Cartesian space (x, y) , or (x, y, z) for three-dimensional cases, to add unsteady pressure, temperature, and velocity. Selecting it adds a Spatial Linear Wake Data node to the **Physics Values** node.

Contents:

Understanding Wake Options

See Also:

What Is the Harmonic Balance Flow and Energy Model?
What Is the Non-Reflecting Option?

Understanding Wake Options

The following boundary values are available for the Harmonic Balance Flow and Energy ModelFlow and Energy model:

- Gaussian wake
- Hodson wake
- Sine wake
- Spatial annular wake data
- Spatial linear wake data
- Wake Fourier coefficients

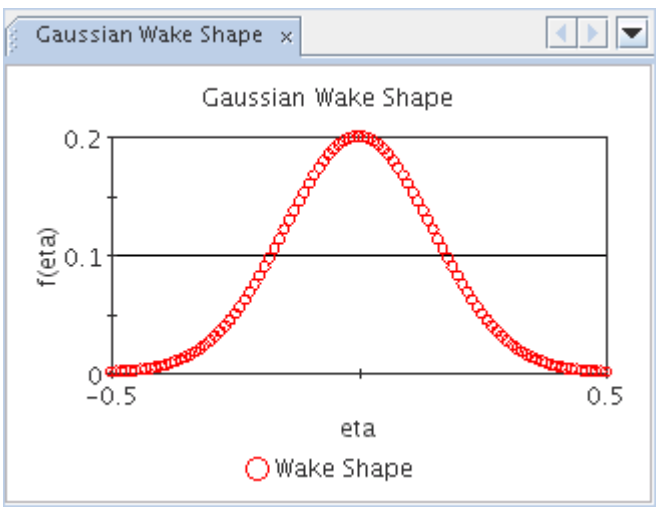
You must set the *Pitch* and the *Number of Blades* in the **Blades Per Pitch** child node of the **Sine**, **Gaussian**, **Hodson**, **Fourier Data**, **Spatial Annular Data**, or **Spatial Linear Data** wakes. These values are required for computing blade row frequencies.

Blades Per Pitch Properties

Pitch	Specifies the pitch in angular units in three-dimensions and in length units in two-dimensions (linear length in Y direction).
Number of blades	This property defines number of blades in the blade row corresponding to the wake. If the wake is defined at an inlet, the number of blades on the upstream blade row is required. Similarly, if the wake is defined at an outlet, the number of blades on the downstream blade row is required.

Gaussian Wake

The Gaussian wake function is defined in Eqn. (1019).



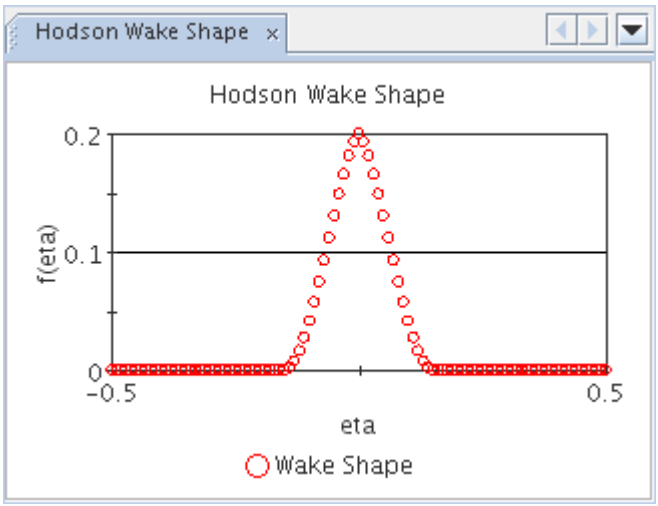
Properties

Width	The wake width, W . This property is a fractional wake width and is a non-dimensional number. For example, a width of $W = 0.2$ would mean that the width of the unsteady disturbance (as defined in the Wake Functions) would be 20% of the pitch.
Defect	The wake defect, D . This property is the amplitude of the unsteady wake function, which defines a velocity defect distribution along the circumferential direction. For instance, specifying the defect D as 0.2 would mean that the amplitude of the unsteady wake function would be equal to $0.2V$, where V refers to the magnitude of the absolute velocity averaged over all the cell faces in the circumferential direction.

Rotation Rate	This property defines angular velocity of blades in the blade row corresponding to the wake. If the wake is defined at an inlet, the angular velocity on the upstream blade row is required. Similarly, if the wake is defined at an outlet, the angular velocity on the downstream blade row is required.

Hodson Wake

The Hodson wake function is defined in Eqn. (1020).

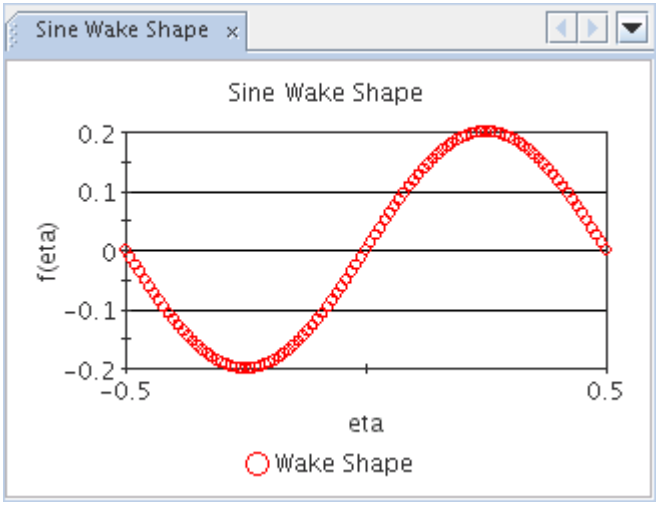


Properties

Width	The wake width, W . This property is a fractional wake width and is a non-dimensional number. For example, a width of $W = 0.2$ would mean that the width of the unsteady disturbance (as defined in the Wake Functions) would be 20% of the pitch.
Defect	The wake defect, D . This property is the amplitude of the unsteady wake function, which defines a velocity defect distribution along the circumferential direction. For instance, specifying the defect D as 0.2 would mean that the amplitude of the unsteady wake function would be equal to $0.2V$, where V refers to the magnitude of the absolute velocity averaged over all the cell faces in the circumferential direction.
Rotation Rate	This property defines angular velocity of blades in the blade row corresponding to the wake. If the wake is defined at an inlet, the angular velocity on the upstream blade row is required. Similarly, if the wake is defined at an outlet, the angular velocity on the downstream blade row is required.

Sine Wake

The Sine wake function is defined in Eqn. (1021).



Properties

Defect	The wake defect, D . This property is the amplitude of the unsteady wake function, which defines a velocity defect distribution along the circumferential direction. For instance, specifying the defect D as 0.2 would mean that the amplitude of the unsteady wake function would be equal to $0.2V$, where V refers to the magnitude of the absolute velocity averaged over all the cell faces in the circumferential direction.
Rotation Rate	This property defines angular velocity of blades in the blade row corresponding to the wake. If the wake is defined at an inlet, the angular velocity on the upstream blade row is required. Similarly, if the wake is defined at an outlet, the angular velocity on the downstream blade row is required.

Spatial Annular Wake Data

The table that you select must specify the cylindrical components of the velocity field for the wake, along with pressure and temperature. For three-dimensional cases, the table must provide:

- Tangential coordinate
- Radial coordinate
- Normal velocity component
- Tangential velocity component
- Radial velocity component
- Temperature
- Pressure

The column headings in the table you provide are mapped to the required quantities using the properties of the node. In two-dimensional cases, the radial coordinate and radial velocity component are not required.

Properties of Spatial Annular Wake Data

Annular Table: Theta	Selects the table column containing the tangential coordinate.
Annular Table: P	Selects the table column containing the pressure data.
Annular Table: Vn	Selects the table column containing the normal component of velocity.
Annular Table: Vt	Selects the table column containing the tangential component of velocity.
Annular Table: Vr	Selects the table column containing the radial component of velocity.
Annular Table: T	Selects the table column containing the temperature data.
Annular Table: R	Selects the table column containing the radial coordinate.
Spatial Wake Data Table	Selects the table to use.
Number of curves	Specifies the number of curves (radial or axial strips) in the table.
Number of data points per curve	Specifies the number of points in the circumferential direction for each curve in the table.
Number of modes to extract	Specifies the number of Fourier modes to compute from the table data.
Rotation Rate	This property defines angular velocity of blades in the blade row corresponding to the wake. If the wake is defined at an inlet, the angular velocity on the upstream blade row is required. Similarly, if the wake is defined at an outlet, the angular velocity on the downstream blade row is required.

Expert Properties

Wake Data Position Option	Determines whether wake data is positioned on the boundary, or at an offset to it.	
	At the boundary	The wake data is positioned at the boundary.
	At a specified location	The wake data is offset by an amount that is given by the *Wake Data Position*, which is applied in the direction normal to the wake boundary
Wake Data Position	The distance, positive, or negative, by which the wake data is offset from the boundary in the direction normal to the boundary.	

Spatial Linear Wake Data

The table that you select must specify the cartesian components of the velocity field for the wake, along with pressure and temperature. For three-dimensional cases, the table should provide:

- X coordinate
- Y coordinate

- Z coordinate
- X component of velocity
- Y component of velocity
- Z component of velocity
- Temperature
- Pressure

The column headings in the table you provide are mapped to the required quantities using the properties of the node. In two-dimensional cases, the Z coordinate and Z component of velocity are not required.

Properties of Spatial Linear Wake Data

Linear Table: P	Selects the table column containing the pressure data.
Linear Table: T	Selects the table column containing the temperature data.
Linear Table: Vx	Selects the table column containing the X component of velocity.
Linear Table: Vy	Selects the table column containing the Y component of velocity.
Linear Table: Vz	Selects the table column containing the Z component of velocity.
Linear Table: X	Selects the table column containing the X coordinate.
Linear Table: Y	Selects the table column containing the Y coordinate.
Linear Table: Z	Selects the table column containing the Z coordinate.
Spatial Wake Data Table	Selects the table to use.
Number of curves	Specifies the number of curves (radial or axial strips) in the table.
Number of data points per curve	Specifies the number of points in the circumferential direction for each curve in the table.
Number of modes to extract	Specifies the number of Fourier modes to compute from the table data.
Rotation Rate	This property defines angular velocity of blades in the blade row corresponding to the wake. If the wake is defined at an inlet, the angular velocity on the upstream blade row is required. Similarly, if the wake is defined at an outlet, the angular velocity on the downstream blade row is required.

Expert Properties

Wake Data Position Option	Determines whether wake data is positioned on the boundary, or at an offset to it.	
	At the boundary	The wake data is positioned at the boundary.
	At a specified location	The wake data is offset by an amount that is given by the *Wake Data Position*, which is applied in

		the direction normal to the wake boundary
Wake Data Position	The distance, positive, or negative, by which the wake data is offset from the boundary in the direction normal to the boundary.	

Wake Fourier Coefficients

The table that you select needs a special format. It specifies the non-zero Fourier modes to control the wake, and the column headers must be as follows:

`'Pnr','VNnr','VTnr','VRnr','Tnr',`

`'Pni','VNni','VTni','VRni','Tni'`

where `n` in each of the above column headers corresponds to the non-zero modes, that is, `n` can take values 1, 2, 3, and so on. In the above, `P`, `VN`, `VT`, `VR`, and `T` refer to pressure, velocity components (normal, tangential, radial), and temperature. The entries `r` and `i` refer to the real and imaginary components of these modes. For two-dimensional cases, `VR` would not be required. Also, the order in which the columns appear does not matter, as long as they are named as described above.

For example, if you wish to specify two modes, the table contains the following data:

`'P1r','VN1r','VT1r','VR1r','T1r',`

`'P1i','VN1i','VT1i','VR1i','T1i',`

`'P2r','VN2r','VT2r','VR2r','T2r',`

`'P2i','VN2i','VT2i','VR2i','T2i'`

While specifying the number of modes to use from this table, STAR-CCM+ allows you to specify an integral number from 0 to the number of modes present in the file. For the two-mode table example shown above, you could choose the number of modes as 0 (implying “use no wake data from file table”), 1 (“use only the first mode from the table”), or 2 (“use both the modes from the table”).

Properties

Fourier Coefficients Table	Selects the table to use.
Number of modes to use	Specifies the number of modes to use from the table data.
Rotation Rate	This property defines angular velocity of blades in the blade row corresponding to the wake. If the wake is defined at an inlet, the angular velocity on the upstream blade row is required. Similarly, if the wake is defined at an outlet, the angular velocity on the downstream blade row is required.

Expert Properties

Wake Data Position	Specifies the spatial location of the wake table data.
Wake Data Position Option	Determines where the wake data are specified.

	At the boundary	The wake data are at the boundary.
	At a specified location	The wake data are at a spatial location other than the boundary.

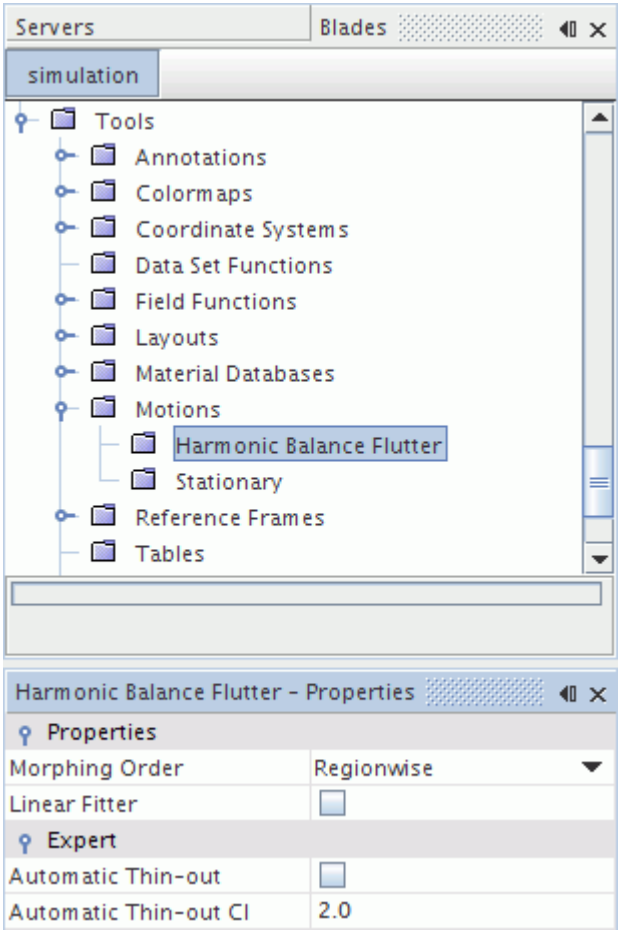
See Also:

Blade Row Frequencies Computation

Wake Functions

What Is the Harmonic Balance Flutter Motion?

Harmonic Balance Flutter motion is used in simulations that involve the Harmonic Balance method with blade vibration.



Harmonic Balance Flutter Properties

Morphing Order	The morpher can operate in two different modes: one region at a time, or one motion at a time. This property selects the mode to use.	
	Regionwise	Each region that is associated with this motion is morphed independently, one region at a time. This option is used, for example, in a fluid-solid case, where the solid region has a solid displacement motion and the fluid region has a morphing motion.

	Motionwise	All regions that are associated with this motion are morphed at the same time. This option is recommended for use in cases where internal inter faces are present between regions, if the floating morpher boundary method is applied to the boundaries on either side of the interface, and both regions are set to use the same morphing motion.
Linear Fitter	This option extracts a linear transformation from the prescribed motion, and uses this transformation to move vertices outside of the main morpher. Any remaining motion is treated using the multiquadric procedure as usual. If the linear part constitutes most the motion and the deformation part is small, the main morpher is expected to perform more robustly if the Morph From Zero property is enabled on the Morpher solver.	
	Activated	Linear motion is treated separately from multiquadric interpolation.
	Deactivated	All motion is treated in multiquadric interpolation.

Harmonic Balance Flutter Expert Properties

Automatic Thin-out	Applies an additional stage of control vertex thinning after the user-defined thin factors have been applied. Tick the Morph from Zero property on the Morpher solver when using this option.	
	Activated	Automated thinning is applied.
	Deactivated	No automated thinning is applied. Only user-defined thin factors are accounted for.
Automatic Thin-out CI	This property controls the target spacing between control points. If the value is smaller, the spacing is smaller resulting in denser control point distribution. If the value is larger, it coarser control point distribution.	

Contents:

Setting Up a Simulation with Harmonic Balance Flutter Motion
Specifying Flutter Parameters
What Is the Blade Flutter Solver?
Morphing Option
Harmonic Balance Flutter Boundary Values

See Also:

What Is the Blade Flutter Solver?

Setting Up a Simulation with Harmonic Balance Flutter Motion

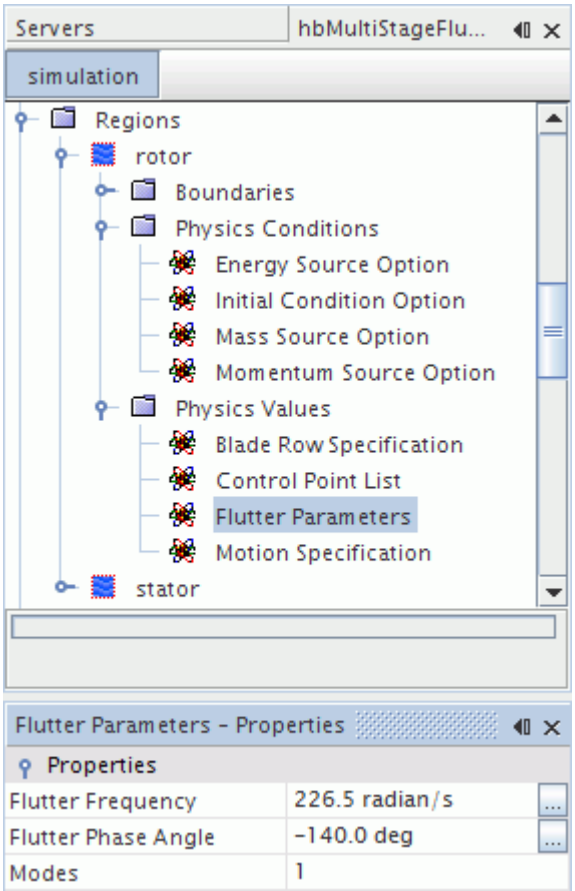
To use blade vibration in a simulation, follow the steps below:

1. Activate the Harmonic Balance Flow and Energy Model continuum model.
2. Create a Harmonic Balance Flutter Motion.
3. Apply the **Harmonic Balance Flutter** motion to a region using the Motion Specification region value.
4. Specify the Flutter Parameters and, if necessary, control points and their associated displacements.
5. Specify the properties of the Blade Flutter Solver node added to the **Solvers** manager.
6. Set up the properties for the Morphing Option boundary condition.
7. Finally, specify the Harmonic Balance Flutter Boundary Values.

To monitor the vibratory force, or the real or imaginary part of that force, use the Generalized Vibratory Force report.

Specifying Flutter Parameters

You specify flutter parameters in the properties of the **Physics Values** > **Flutter Parameters** node of a region.



Before setting the flutter parameters:

- Set the Motion Specification of the region to flutter motion.

  You can select flutter motion for different blade rows. There is no limit to the number of blade rows that can be modeled with flutter. However, in most practical cases, you use the flutter model only in one row, even in a multi-blade row case.
- Associate the region to a blade row.

To set the flutter parameters:

- Select the **Regions** > **[Region]** > **Physics Values** > **Flutter Parameters** node, and set the *Flutter Frequency*, *Flutter Phase Angle*, and *Modes* properties.

Flutter Parameters Properties

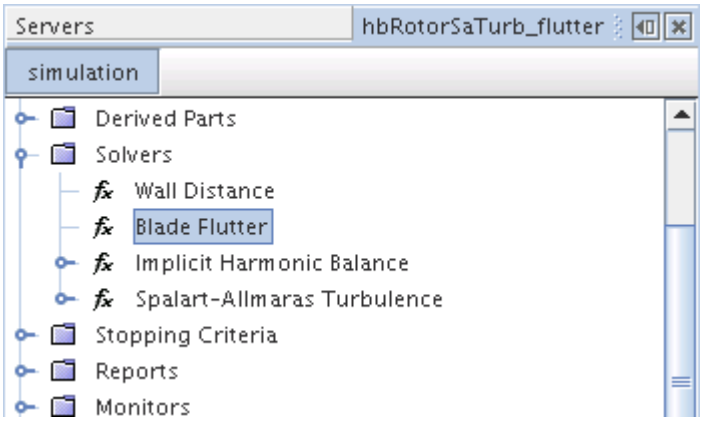
Flutter Frequency	The frequency ω of unsteadiness. See Eqn. (1028).
Flutter Phase Angle	The inter-blade phase angle σ . See Eqn. (1028).
Modes	The number of modes to retain for the blade flutter.

What Is the Blade Flutter Solver?

The Blade Flutter solver displaces the computational mesh at each of the time levels, based on user-specified blade displacements, and stores the mesh metrics (cell volumes and centroids, face areas, and centroids) at each time level for use by the Implicit Harmonic Balance solver.

These metrics are computed and stored only once, during the first iteration. They are recomputed when the flutter boundary or region conditions change. You can force the recomputation of the mesh metrics by right-clicking the **Blade Flutter** solver node, and selecting **Reset Solver**.

The Blade Flutter solver is activated when you select the Harmonic Balance Flutter motion. The **Blade Flutter** solver node represents this solver, and has Blade Flutter Solver Properties.



Contents:

Blade Flutter Solver Properties
Blade Flutter Solver Pop-Up Menu

See Also:

What Is the Harmonic Balance Flutter Motion?

Blade Flutter Solver Properties

Mode Shape Scale Factor	Scales the user-specified blade displacement magnitudes. The default value of this property is 1, indicating that no scaling is applied to the input mode shape. This scaling is required as the mode shapes that are extracted from third-party codes are normalized in a way that gives a non-physical amplitude, which can be orders of magnitude too large in most cases.

Expert Properties

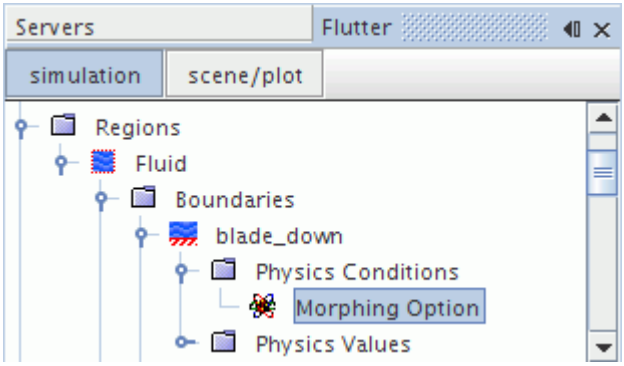
Verbosity	Provides additional output for this value while the simulation is running. This property may be useful for debugging problems when they occur.	
	None	Leaves output at its default.
	Low	Adds messages as to which region is morphed.
	High	Generates precise equation information about the morphing process as the solver converges.
Maximum Iterations	The number of iterations the morpher solver is allowed to take to calculate the vertex displacements to the specified tolerance (set by the *Morpher Tolerance* property).	
Skip Thinned Vertex Corrector	Whether to disable the Thinned Vertex Corrector. This option is provided for backward compatibility only. Keep this property deactivated at all times.	
	Activated	Disables the Thinned Vertex Corrector
	Deactivated	Enables the Thinned Vertex Corrector
Morpher Tolerance	Non-dimensional parameter that is used to specify acceptable accuracy in morphing. This value is multiplied by the longest edge length of the bounding box enclosing the morphing region. Morpher tolerance is the non-dimensional magnitude of the noise that enters in the displacement field. If your grid size is O(10)m, the noise level for the default setting of 1e-7 is 1e-6 m. This noise value must be well below your smallest grid size (typically layer thickness). Larger tolerance values allow the morpher to run faster.	
Solver Frozen	Prevents or allows the solver from updating the solution while iterating. Do not activate this feature as your first choice: it is primarily a debugging tool and not supported for normal simulation work.	
	Activated	The solver does not update the solution.
	Deactivated	The solver updates the solution.

Blade Flutter Solver Pop-Up Menu

Reset Solver	Initializes the Blade Flutter solver. This action has the same effect as initializing and clearing the solution combined, except the only solution data that are cleared are that of the Blade Flutter solver.

Morphing Option

This node allows you to determine how the morphing for a boundary is specified. It is activated when the Harmonic Balance Flutter motion is enabled.



Properties

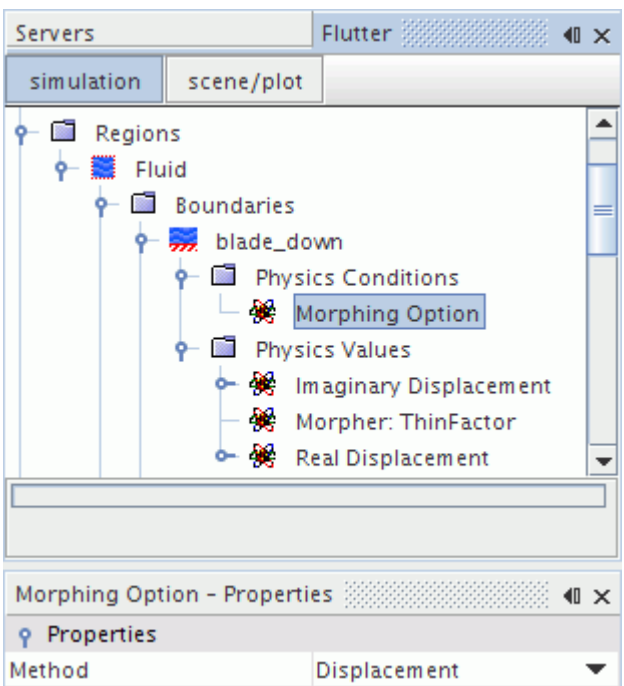
Method	Sets the morphing condition.	
	Fixed	This option adds the Morpher: ThinFactor physics value node. The Harmonic Balance Flutter motion and the Morpher motion share this node. Points on this boundary are treated as control points with zero displacement.
	In Plane	This option adds the Morpher Plane Options physics value node. The Harmonic Balance Flutter motion and the Morpher motion share this node. Points on this planar boundary are allowed to slide in the plane. This is implemented by damping the displacements, and is much less expensive than using control points.
	Displacement	Points on this boundary have specified displacement. This option, which is available when the Harmonic Balance Flutter motion is enabled, adds the following nodes to the **Physics Values** node: • Imaginary Displacement

		• Morpher: ThinFactor (also used by the Morpher motion) • Real Displacement
	Floating	The option is turned off for the boundary. No nodes are added to the **Physics Values** node. Points on this boundary have no boundary condition; they move wherever the other control points dictate.

Harmonic Balance Flutter Boundary Values

The following boundary values are available for the Harmonic Balance Flutter motion when the **Morphing Option** method is set to **Displacement**:

- Morpher: Imaginary Displacement
- Morpher: Real Displacement



The physical displacements are calculated according to Eqn. (1022) at each time level. The real and imaginary displacements specify the amplitude of the periodic function in Eqn. (1022).

In addition, the Harmonic Balance Flutter motion and the Morpher motion share the following boundary values:

- Morpher Thinout Factor

- Morpher Plane Options

See Morpher Boundary Condition Reference.

Contents:

What Is the Imaginary Displacement?
What Is the Real Displacement?

See Also:

Working with Morphing

What Is the Imaginary Displacement?

This boundary value is activated when the Morphing Option boundary condition is set to Displacement.

It is input as a vector profile and is used to specify the amplitude of physical displacements, cf Eqn. (1022).

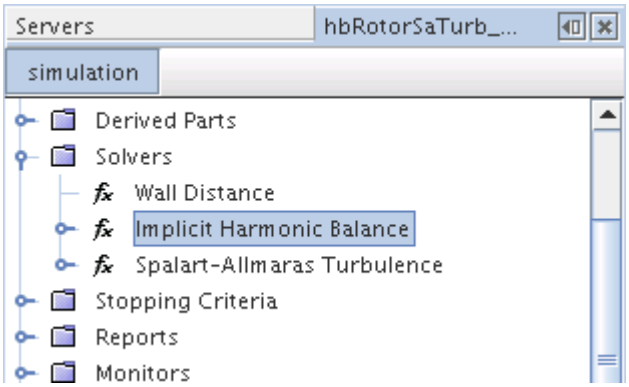
What Is the Real Displacement?

This boundary value is activated when the Morphing Option boundary condition is set to Displacement.

It is input as a vector profile and is used to specify the amplitude of physical displacements, cf Eqn. (1022).

What Is the Implicit Harmonic Balance Solver?

The Implicit Harmonic Balance solver controls the solution update for the Harmonic Balance Flow and Energy Model. It is used for implicit integration, using a coupled algebraic multi-grid method.



Implicit Harmonic Balance Solver Properties

Courant Number	The value of CFL in Eqn. (1670) that controls the local time-step at each iteration. The default Courant number is 5. In general, and depending on the initial conditions, it might be necessary to start at a lower Courant number, and then increase it.

Expert Properties

Reduced Frequency Limiter	Governs stability for convergence, especially for large far-field cells. The entry specifies ω_r in the ratio ω_r/ω_{max} , which limits the magnitude of the time-step.

For information about properties in the *Expert* category, see Expert Properties.

See Also:

What Is the Harmonic Balance Model?

What Are the Harmonic Balance Turbulence Models?

Post-Processing Harmonic Balance Solution Data

You can reconstruct the evolution of a flow variable over time at a given point or on a surface. You can also differentiate between the zeroth harmonic and higher harmonics when plotting results.

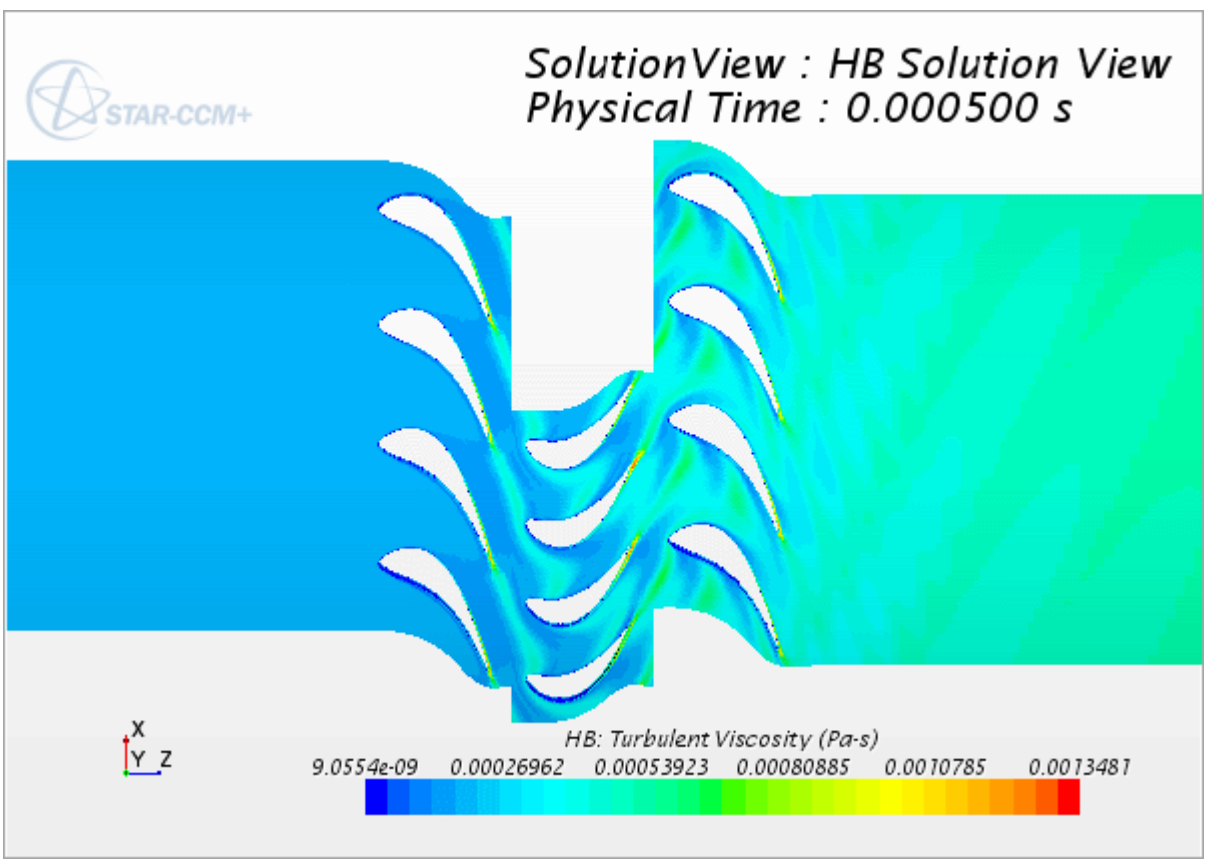
Post-processing for harmonic balance solution data offers the following features:

- The solution is known on every blade in the row, not just on the meshed blades.
- The solution is known for all times, so you can derive views of blade positions and field values for any time during the run.
- The solution can be represented and viewed in the frequency domain. This can be helpful for understanding the unsteady flow features.

To examine and analyze the results of a harmonic balance simulation, create a harmonic balance solution view and use it as the basis for creating scenes, reports, and plots. See Applying Recorded Solution Views to Scenes and subsequent sections. Field functions and an annotation specific to harmonic balance are available.

Example Post-Processing Procedure

The following screen shot shows a scene of three blade rows, with four passages for each.



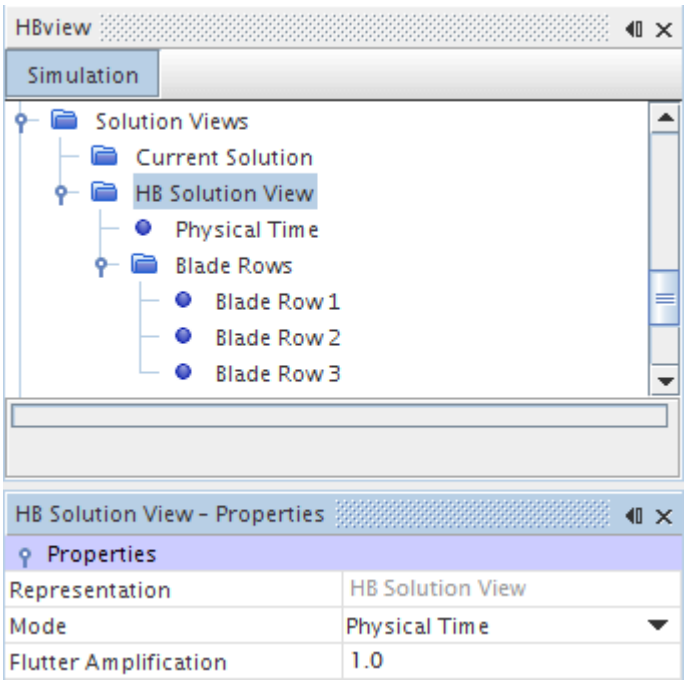
This scene was produced using the following procedure:

1. Right-click **Continua** > **Physics** > **Models** > **Harmonic Balance Flow and Energy** and select **New HB Solution View**. This creates an **HB Solution View** node under the **Solution Views** node.
2. Open the scalar scene, right-click on it, and select **Apply Representation** > **HB Solution View**.
3. Right-click on the color bar and select **Harmonic Balance** > **HB: Turbulent Viscosity** > **Time Function**.
4. Select **Solution Views** > **HB Solution View** > **Blade Row** and, in each **Blade Row**sub-node, set the *Blade Passage Selection* property to `[-2,1]`.
5. Select **Solution Views** > **HB Solution View** > **Physical Time** and set the *Physical Time* property to `5.0E-4 s`.

By varying the settings of the **HB Solution View** node, you can derive scenes for other physical times, or in time level, Fourier, or filtered modes. You can also derive plots and reports, as with other solution views.

What Is the HB Solution View?

The **HB Solution View** node controls the properties of harmonic balance solution view for all selected blade passages of all blade rows in the simulation.



To create a harmonic balance solution view:

1. Right-click **Continua** > **Physics** > **Models** > **Harmonic Balance Flow and Energy** and select **New HB Solution View**. This creates an **HB Solution View** node under the **Solution Views** node.
2. Select the **HB Solution Views node** and select a view mode from the *Mode* property drop-down menu. Set a new value for the *Flutter Amplification* property at this time, if necessary.
3. Select the **HB Solution Views** > **Blade Rows** > **Blade Row**n nodes, one by one, and set their properties.

The blade passages selected for all blade rows are shown in a *Scalar* scene under **Scenes** > **Scalar Scene** n.

HB Solution View Properties

Representation	The name of the **HB Solution View** node. (Read only.)	
Mode	Specifies the mode in which to visualize the fields selected for the individual blade rows.	
	Physical Time	The physical time at which to visualize the blade row fields. This is the default mode.

	Time Level	The time level at which to visualize the blade row fields.
	Fourier Mode	The number of the mode (harmonic) at which to visualized the blade row fields.
	Filtered Mode	The number of the filtered mode (harmonic) at which to visualize the blade row fields.
Flutter Amplification	Magnification factor for blade vibration. The default is 1 (no amplification).	

See Harmonic Balance Solution View Field Functions for a listing of the available fields.

Contents:

Physical Time Mode
Time Level Mode
Fourier Mode
Filtered Mode
Blade Row

See Also:

Harmonic Balance Solution View Field Functions
What Is a Blade Row?

Physical Time Mode

In this mode, the solution view shows the blade row fields at a given physical time.

Cycle Time is displayed to help you select the point in the rotation for visualization.

Physical Time Mode Properties

Physical Time	The physical time at which to visualize the blade row fields.
Cycle Time	The time taken to complete one rotation of the rotor. (Read only.)

Time Level Mode

In this mode, the solution view shows the blade row fields at a given time level—that is, at a given point in the repeating periodic flow.

Time Level Mode Properties

Time Level	The time level at which to visualize the blade row fields.

Fourier Mode

In this mode, the solution view shows the blade row fields for a given harmonic when the periodic flow is analyzed into a Fourier series.

Fourier Mode Properties

Mode	The number of the mode (harmonic) at which to visualized the blade row fields.	
Output Option	The feature of the Fourier mode being visualized.	
	Real	The real part of the Fourier mode.
	Imaginary	The imaginary part of the Fourier mode.
	Magnitude	The magnitude of the Fourier mode.
	Phase	The phase of the Fourier mode.

Filtered Mode

In this mode, the solution view shows the blade row fields for a given mode taken from the Filtered Modes Table under the Harmonic Balance Flow and Energy node.

Filtered Mode Properties

Mode	The number of the filtered mode (harmonic) at which to visualize the blade row fields.	
Output Option	The feature of the filtered mode being visualized.	
	Real	The real part of the filtered mode.
	Imaginary	The imaginary part of the filtered mode.
	Magnitude	The magnitude of the filtered mode.
	Phase	The phase of the filtered mode.

Blade Row

One **Blade Row** node appears under the **HB Solution Views** > **Blade Rows** node for each blade row. Enter the selection as an ordered pair in square brackets, with a negative integer for the number of lagging blades and a positive integer for the number of leading blades.

Examples:

- `[0, 0]` is the original blade alone. This is the default.

- `[-9, 8]` is nine lagging blades through eight leading blades.

Blade Row Properties

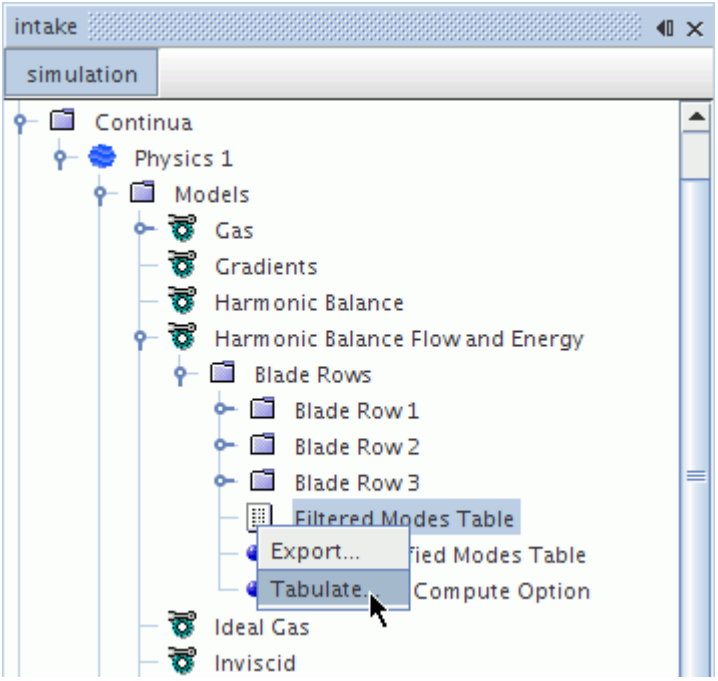
Blade Passage Selection	The blades in the passage selected for visualization.
Blade Passage Range	The number of blades available in the blade row. See Eqn. (1003). (Read only.)

Filtered Mode Field Functions

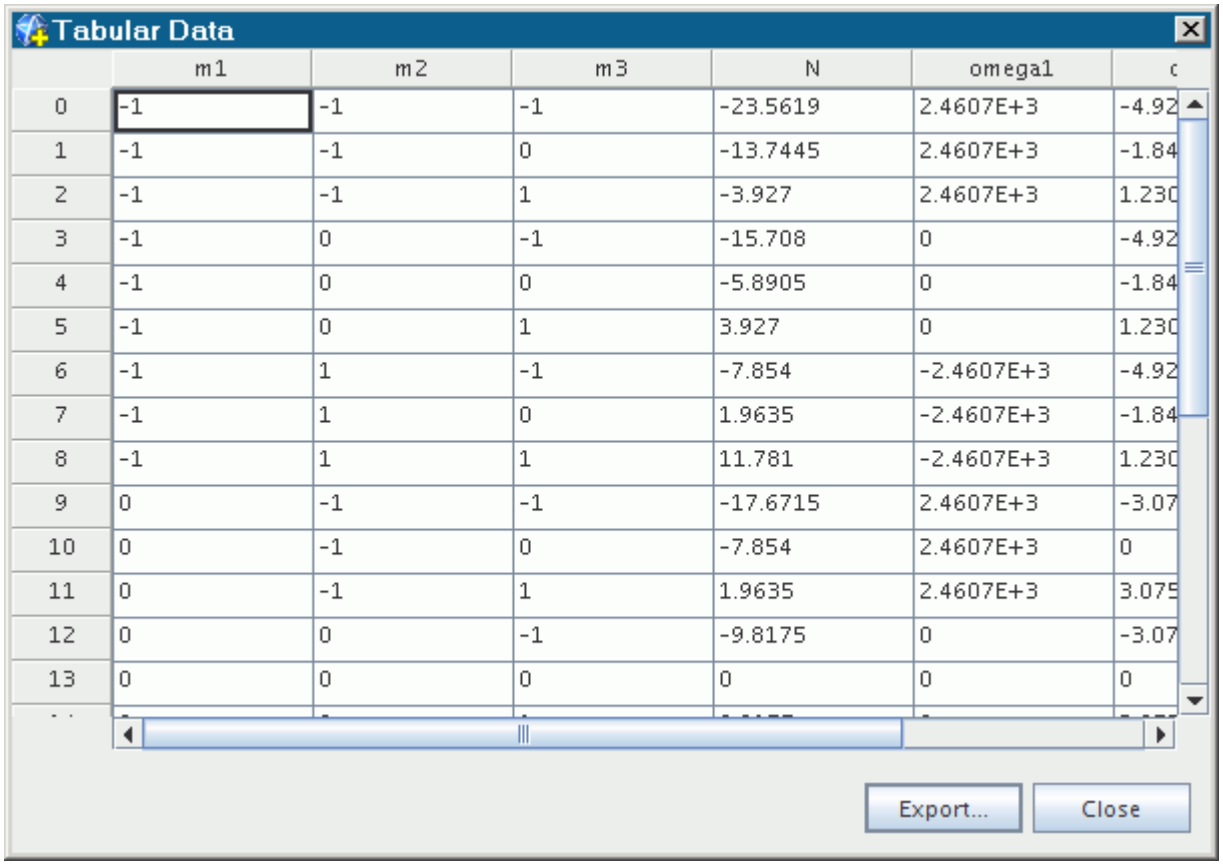
Filtered mode field functions allow you to visualize the Fourier coefficients corresponding to each mode (or harmonic) for multi-row harmonic balance cases. As the total number of modes that are made available in a multi-stage simulation could be a large number, you have the option to selectively choose the modes of interest.

To control the number of such field functions to be made available for visualization:

1. Right-click the **Continua** > **Physics 1** > **Models** > **Harmonic Balance Flow and Energy** > **Blade Rows** > **Filtered Modes Table** node and select **Tabulate**.



2. In the table that appears, review the values and note the numbers of the mode rows that you would like to post-process.

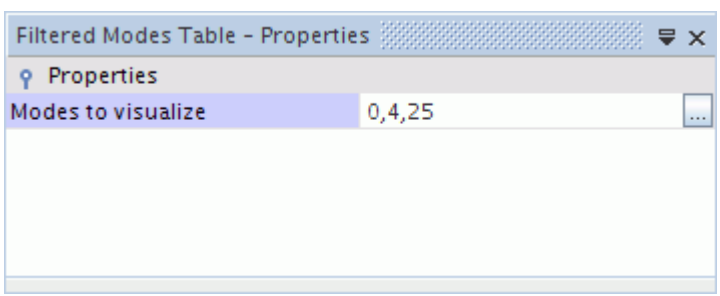


Tabular Data

	m1	m2	m3	N	omega1	c
0	-1	-1	-1	-23.5619	2.4607E+3	-4.92
1	-1	-1	0	-13.7445	2.4607E+3	-1.84
2	-1	-1	1	-3.927	2.4607E+3	1.230
3	-1	0	-1	-15.708	0	-4.92
4	-1	0	0	-5.8905	0	-1.84
5	-1	0	1	3.927	0	1.230
6	-1	1	-1	-7.854	-2.4607E+3	-4.92
7	-1	1	0	1.9635	-2.4607E+3	-1.84
8	-1	1	1	11.781	-2.4607E+3	1.230
9	0	-1	-1	-17.6715	2.4607E+3	-3.07
10	0	-1	0	-7.854	2.4607E+3	0
11	0	-1	1	1.9635	2.4607E+3	3.075
12	0	0	-1	-9.8175	0	-3.07
13	0	0	0	0	0	0

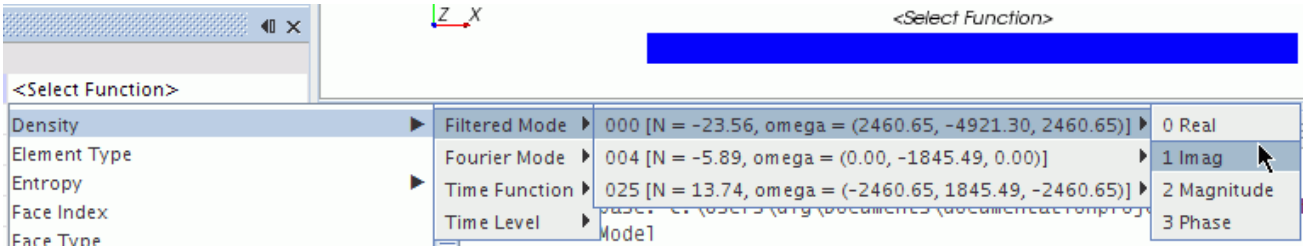
Export... Close

3. To close the table, click **Close**.
4. With the **Filtered Modes Table** node still selected, enter the row numbers in the *Modes to visualize* property.



After you enter this list, a new set of field functions, with **Filtered Mode** in their names, gets generated for the variables of interest such as density, velocity, temperature, and pressure. These field functions can be used to visualize the modal content of a variable corresponding to the chosen mode.

In the following example, the chosen modes are 0, 4, and 25. The variable **Density** has a subset called **Filtered Mode**. Each filtered mode (being a complex number) has four components: real, imaginary, magnitude, and phase. These components correspond to the real and imaginary parts of the complex number, the complex magnitude, and the phase, respectively.


<Select Function>
<Select Function>
Density
Element Type
Entropy
Face Index
Face Type
Filtered Mode
Fourier Mode
Time Function
Time Level
000 [N = -23.56, omega = (2460.65, -4921.30, 2460.65)]
004 [N = -5.89, omega = (0.00, -1845.49, 0.00)]
025 [N = 13.74, omega = (-2460.65, 1845.49, -2460.65)]
0 Real
1 Imag
2 Magnitude
3 Phase

Harmonic Balance Guidelines

This section describes the set-up and solution procedures generally applicable to harmonic balance simulations.

Wakes

The inlet boundary should be a stagnation inlet. Unsteady boundary conditions are prescribed by specifying an unsteady “Wake” at the inlet boundary.

This unsteady “Wake” can be specified either:

- By choosing a “standard” wake model, such as Sine, Gaussian and Hodson (as described in the journal paper: [reflink]), or
- By specifying a user table that contains the spatial variation of pressure, velocity, and temperature or their Fourier coefficients.

Without the wake, STAR-CCM+ computes a steady flow.

Memory Requirements

The required memory scales approximately with the number of time levels that are used in the calculation. The number of time levels N is given by N = 2M + 1, where M is the number of user-specified modes. For most purposes, 1 to 3 modes are enough, with 3 a typical value. The larger the unsteadiness in the flow, the more harmonics you require.

Domain and Boundary Conditions

The physical domain depends on the physical geometry. Use the following guidelines for setting boundary conditions:

- You do not need to extend the domain to mitigate reflections. Instead, use non-reflecting boundary treatments on the inlet, outlet, and interfaces. Typically, five to ten non-reflecting modes are sufficient. It is often best to first converge a case with the non-reflecting treatment disabled.
- Specify boundary conditions as inlet total pressure and exit static pressure. Verify boundary conditions first with the steady solver. If a particular mass flow is desired, first use the steady solver in conjunction with a target mass-flow pressure outlet to determine the proper boundary pressure.
- Use mixing plane interfaces between blade regions. See Mixing Plane Interface.
- Only a single blade passage in each row is required, regardless of the relative blade count in each row.

Mesh

Use the following guidelines for setting up the mesh:

- The harmonic balance solver can solve on an arbitrary unstructured mesh. A structured mesh at the interface is not required.
- It is best to use either polyhedral grids or structured grids (see Turbo Wizard).
- Mesh resolution on inter-row interfaces must be uniform and relatively fine.
- The two sides of an inter-row interface must have equivalent cell sizing.

- Use a low y+ mesh whenever possible. Take special care to ensure a low y+ mesh on blades and vanes where separation occurs.

Physics

Use the following guidelines for setting up the physics:

- Use the following physics models:
  - Three Dimensional
  - Harmonic Balance
  - Ideal Gas
  - Turbulent
  - All y+ Wall Treatment
- The harmonic balance solver is currently compatible with most K-Epsilon, K-Omega, and Spalar-Allmaras turbulence models.
- Initiate the harmonic balance simulation from a converged steady simulation.
- Per-region partitioning is the most efficient partitioning method for most harmonic balance simulations.
- In the Blade Row Manager expert properties, set the *Frequency Compute Option* to **Consider Only Neighbors** for most cases. The **Consider All Blade Rows** option is best for cases where unsteady flow features persist more than one blade row upstream or downstream, but not for cases with more than four blade rows. See Blade Rows Manager Expert Properties.
- Set the Courant number to 5, to begin with. Cases with higher harmonic content may require a lower Courant number.
- To investigate high turbulence residuals, turn residual normalization off to determine if the absolute value of turbulence residuals is low. When starting from a converged steady solution, turbulence residual values may already be low.

Solution Procedure

It is usually best to begin a harmonic balance simulation from a converged steady solution. While it is sometimes possible to begin a harmonic balance simulation from a uniform initial condition, this often takes longer than first converging a steady solution, then reconverging the harmonic balance solver.

Use the following steps:

1. Generate a steady physics continuum.
   - Use the same models to be used with the harmonic balance model in the last step, such as these:
     - Three Dimensional
     - Steady
     - Ideal Gas
     - Turbulent
     - Spalart-Allmaras Turbulence
     - All y+ Wall Treatment
   - Use the same reference values as the harmonic balance physics continuum.
   - Set all regions to the steady physics continuum.
   - Set the initial conditions as follows:

    - Initial velocity: a non-zero value in the axial direction.
    - Initial temperature: an estimate of the inlet static temperature.
    - Initial pressure: the greater of inlet pressure and exit pressure
  - Set the Courant number to 20.
2. Disable non-reflecting treatment. Implicit (reflecting) interfaces can be more stable for initialization.
  - Set all indirect interfaces to implicit. See Mixing Plane Interface.
  - Turn off non-reflecting treatment on the inlet and outlet.
3. Initialize the grid to provide a good initial flow field for turbines and moderate pressure ratio compressors. Perform grid sequencing initialization with the following settings:
  - Maximum grid levels: 10
  - Maximum iterations per level: 200
  - Convergence tolerance per level: 0.005
  - CFL number: 20
4. Run the steady solver. At convergence, residuals should drop about four orders of magnitude and integrated quantities should approach a constant value.
5. Re-set values for the non-reflecting case:
  - Enable non-reflecting treatment on the inlet and outlet. Set the number of non-reflecting modes to between 5 and 10.
  - Change all indirect interfaces to explicit and enable the non-reflecting treatment. (See Mixing Plane Interface.) Set the number of modes on all interfaces to between 5 and 10.
    - Set the number of non-reflecting modes on both sides of every interface.
    - Do not set non-reflecting treatment on the boundary entity that makes up the interface. The boundary entity should be empty upon initialization.
6. Run the steady solver to convergence again.
7. Change all regions to the harmonic balance physics continuum and run the Harmonic Balance solver to convergence.

Harmonic Balance Formulation

This section describes the following formulation for the Harmonic Balance method.

Contents:

Trigonometric Series
Time Derivative
Harmonic Balance Equations
Harmonic Time Functions
Wake Functions
Real and Imaginary Displacements
Blade Row Frequencies Computation
Blade Row Nodal Diameter Computation
Fictitious Rotation Rate Computation
Fluttering Motion of a Rotor
Harmonic Balance Bibliography

Trigonometric Series

The starting point for the Harmonic Balance method is the representation of a real, time-periodic function $w(\mathbf{x}, t)$ with period T by a trigonometric series in m harmonics:

$$w(\mathbf{x}, t) = w_0(\mathbf{x}) + \sum_{n=1}^{m} w_{2n-1}(\mathbf{x})\cos(\omega n t) + w_{2n}(\mathbf{x})\sin(\omega n t) \tag{1005}$$

where:

m is the number of user-specified modes or harmonics,

$w(\mathbf{x}, t)$ is the time domain or physical solution at the spatial location $\mathbf{x} = (x, y, z)$ and time t,

$N = 2m + 1$ is the number of time levels,

w_0 is the mean solution at position $\mathbf{x}$, and,

w_n are the Fourier coefficients.

ω is the user-specified excitation frequency, and

$$T = \frac{2\pi}{\omega} \tag{1006}$$

This can also be represented in complex form as:

$$w(\mathbf{x}, t) = \sum_{n=-m}^{m} w_n(\mathbf{x}) e^{j\omega n t} \tag{1007}$$

where the summation goes from $-m$ to m, and $w_n(\mathbf{x})$ and $w_{-n}(\mathbf{x})$ are complex conjugate pairs.

At each point in space (x, y, z) Eqn. (1005) contains N coefficients w_n that can be uniquely determined by N time-samples of the continuous function, $w_k^* = w(t = t_k)$, as follows:

$$w_0 = \frac{1}{N}\sum_{k=0}^{N-1} w_k^* \tag{1008}$$

$$w_{2n-1} = \frac{2}{N}\sum_{k=0}^{N} w_k^* \cos \omega n t_k;\ n = 1 \ldots m \tag{1009}$$

$$w_{2n} = \frac{2}{N}\sum_{k=0}^{N} X_i w_k^* \sin \omega n t_k;\ n = 1 \ldots m \tag{1010}$$

where the k-th discrete time level is evaluated at time

$$t_k = \frac{kT}{N} \tag{1011}$$

The N vector of functional values, w_k^* at the discrete time levels can be recovered from the N vector of coefficients w_k by use of Equations Eqn. (1005), Eqn. (1006), and Eqn. (1011) as follows:

$$w_k^* = w_0 + \sum_{n=1}^{m}\left[w_{2n-1}\cos\left(2\pi\frac{nk}{N}\right) + w_{2n}\sin\left(2\pi\frac{nk}{N}\right)\right. \quad ; \quad k = 0, \ldots, 1 \tag{1012}$$

Note that it is really $w_k{}^*$ that is stored in STAR-CCM+. The values w_n are computed when needed internally, or for reporting and plotting purposes when you request them.

Time Derivative

From Equations Eqn. (1006), Eqn. (1011) and Eqn. (1012), the time derivative of the discrete solution at each time level is given by:

$$\frac{\partial w_k^*}{\partial t} = \omega \sum_{n=1}^{m} n\left(w_{2n}\cos 2\pi\frac{nk}{N} - w_{2n-1}\sin 2\pi\frac{nk}{N}\right) \tag{1013}$$

Substituting Equations Eqn. (1009) and Eqn. (1010) into Eqn. (1013) yields

$$\frac{\partial w_k^*}{\partial t} = \frac{2\omega}{N}\sum_{n=1}^{m} n\left(\cos 2\pi\frac{nk}{N}\sum_{j=0}^{N-1} w_k{}^* \sin 2\pi\frac{nj}{N} - \sin 2\pi\frac{nk}{N}\sum_{j=0}^{N-1} w_k{}^* \cos 2\pi\frac{nj}{N}\right) \tag{1014}$$

We can represent the N vector of time-derivatives given by Eqn. (1014) as

$$\frac{\partial w^*}{\partial t} = [\mathbf{D}]w^* \tag{1015}$$

where the transform operator $[\mathbf{D}]$ has elements D_{ij} given by:

$$D_{ij} = \frac{2\omega}{N}\sum_{n=1}^{m} n\left(\cos 2\pi\frac{ni}{N}\sin 2\pi\frac{nj}{N} - \sin 2\pi\frac{ni}{N}\cos 2\pi\frac{nj}{N}\right) \tag{1016}$$

Diagonal elements D_{ii} are zero.

Harmonic Balance Equations

The transform operator is used to provide a time derivative source term for the conservation equations at each time level.

Applying the initial governing equation for the coupled solver, Eqn. (1649), to each time level and using the time derivative operator gives:

$$\oint \left[\mathbf{F}^* - \mathbf{G}^*\right] \cdot d\mathbf{a} = \int_V \mathbf{H}^* \, \mathrm{d}\, V - \int_V \mathbf{S}^* \, \mathrm{d}\, V \tag{1017}$$

where $\mathbf{S}^* = \mathbf{D}W^*$ is the time derivative source term. Eqn. (1017) is a steady flow equation which can be solved using a pseudo-time marching technique to converge to steady state.

Harmonic Time Functions

The value of any variable Q at a user-specified time t is given by:

$$Q(t) = \sum_{m=-NF}^{NF} Q_m e^{i\left(\omega_m t + sN_m \mathrm{d}\Theta\right)} \tag{1018}$$

where:

- NF is the number of frequencies (or harmonics) in the blade row in question.
- Q_m is the Fourier coefficient associated with the mth harmonic.
- ω_m is the frequency associated with the mth harmonic in Eqn. (1023).
- N_m is the nodal diameter for the mth harmonic in Eqn. (1025).
- $\mathrm{d}\,\Theta$ is the pitch of the modeled blade passage.
- s is 0 for the modeled passage, +1 for the phase-lagged up passage, and -1 for the phase-lagged down passage.

When the **Mean Flow Reconstruction** option is turned on, the only terms retained while computing $Q(t)$ in Eqn. (1018) are the terms corresponding to the mean mode (that is, m = 0) and to any mode that is artificially introduced because of the Consider All Blade Rows option.

Wake Functions

The Gaussian, Hodson, and Sine wake functions are defined as follows, respectively:

$$f(\eta) = D \exp\left(\frac{-\eta^2}{2W^2}\right) \tag{1019}$$

$$f(\eta) = D\left[\max\left(0, \left[1 - \left|\frac{\eta}{W}\right|^{\frac{3}{2}}\right]\right)\right]^2 \tag{1020}$$

$$f(\eta) = D\sin\left(2\pi\eta\right) \tag{1021}$$

where η is a normalized pitch that varies between -0.5 to 0.5, D is the defect and W is the width.

See Also:

Understanding Wake Options

Real and Imaginary Displacements

If v_r and v_i are the real and imaginary displacements that you specify, then the physical displacements dx at every time level k are computed as:

$$\begin{aligned} dx(t_k) &= \left(v_r, v_i\right)\exp\left(\frac{i(2\pi k)}{N}\right) + \left(v_r, -v_i\right)\exp\left(\frac{-i(2\pi k)}{N}\right) \\ &= 2v_r\left(\cos\frac{(2\pi k)}{N}\right) - 2v_i\left(\sin\frac{(2\pi k)}{N}\right) \end{aligned} \tag{1022}$$

where t_k represents the physical time at time level k (where k varies from 0 to $N-1$), N is the number of time levels, and $\left(v_r, v_i\right)$ represents a complex number with v_r and v_i being the real and imaginary parts of the complex number.

Blade Row Frequencies Computation

For a case with N blade rows, the complete set of frequencies for blade row j is:

$$\omega^j = \sum_{\substack{i=1 \\ i \neq j}}^{N} \left[m_i B_i\left(\Omega_j - \Omega_i\right) + f_i\right] \tag{1023}$$

where, for blade row or wake i, m_i is the blade row index and can take all values between $-M_i$ and M_i. M_i is the specified number of modes.

When using the **Consider All Blade Rows** option, i represents any blade row, wake, time varying boundary, or flutter condition, except blade row j. The relative rotation rate between any two blade rows is kept non-zero by adding a small fictitious rotation rate to the blade row if necessary; see Fictitious Rotation Rate Computation. When using the **Consider Only Neighbors** option, the only blade rows that are considered are the rows next to j that share a common indirect interface with j, along with any wake, time varying boundary, or flutter conditions that are directly attached to blade row j.

The variables B_i, Ω_i, and f_i have different meanings, depending on what kind of blade row they refer to. These meanings are described in the following table.

Type of Blade Row $\boldsymbol{i}$	Meaning of $\boldsymbol{B_i}$	Meaning of $\boldsymbol{\Omega_i}$	Meaning of $\boldsymbol{f_i}$
Modeled blade row	The number of blades within the given pitch as specified in the *Number of Blades* property of the	Rotation rate of the modeled blade row, which is inferred from the motion specification of the	Always zero. f_i is not used for a modeled blade row.

Type of Blade Row i	Meaning of B_i	Meaning of Ω_i	Meaning of f_i
	Blade Row n > **Blades Per Pitch** node. See Blades Per Pitch Properties.	blade row region. See Setting the Motion Specification for the Region.	
Wake	The number of blades within the given pitch as specified in the *Number of Blades* property of the **Wake** > **Blades Per Pitch** node. See Blades Per Pitch Properties.	Rotation rate that is specified in the *Rotation Rate* property of the **Wake** node under **Regions** > **Boundary** > **Physics Values**. See Understanding Wake Options.	Always zero. f_i is not used for a wake blade row.
Time-varying option	Set equal to $(B_r\sigma)/(2\pi)$, where B_r is the number of blades in the modeled blade row to which this boundary condition applies, and σ is specified in the *Phase Angle* property of the **Time Varying Option Parameters** node. See Time Varying Option.	Set equal to the rotation rate of the modeled blade row to which this boundary condition applies.	The fundamental frequency of variation in the boundary conditions, in radians/s. Specified in the *Frequency* property of the **Time Varying Option Parameters** node. See Time Varying Option.
Blade flutter	Set equal to $(B_r\sigma_{ib})/(2\pi)$, where B_r is the number of blades in the modeled blade row to which this flutter condition applies, and σ_{ib} is the inter-blade phase angle that is specified in the *Flutter Phase Angle* property of the **Flutter Parameters** node. See Flutter Parameters Properties.	Set equal to the rotation rate of the modeled blade row to which this flutter condition applies.	The frequency of unsteadiness ω used in Eqn. (1028). Specified in the *Flutter Frequency* property of the **Flutter Parameters** node. See Flutter Parameters Properties.

The maximum value for ω^j is set with the *Modes Table Filter Option*. When this option is set to **Cut-Off**, the model computes the maximum value $\omega_{\max}$ using the cut-off factor γ:

$$\omega_{\max} = \underset{i}{MAX}\left\{\gamma \ \ \underset{j}{MAX}\left[M_j\left[B_j\left(\Omega_i - \Omega_j\right) + f_j\right]\right]\right\} \tag{1024}$$

When the option is set to **Maximum Limits**, the model sets $\omega_{\max}$ to the value of the *Maximum Frequency* property.

See Also:

Understanding Wake Options

What Is a Blade Row?

Blade Row Nodal Diameter Computation

For a case with N blade rows, the complete set of nodal diameters that correspond to each frequency computed above (in Eqn. (1023)) for blade row j is:

$$N^j = \sum_{\substack{i=1 \\ i \neq j}}^{N} m_i B_i \tag{1025}$$

where, for blade row or wake i:

- m_i is the modal index and can take all values between $-M_i$ and M_i, where M_i is the number of modes.
- B_i is the number of blades.

When using the **Consider All Blade Rows** option, i represents all blade rows except blade row j. When using the **Consider Only Neighbors** option, only the adjacent blade rows that share a common indirect interface with blade row j are considered.

The maximum value for N^j is set with the *Modes Table Filter Option*. When this option is set to **Cut-Off**, the model computes the maximum value $N_{\max}$ using the cut-off factor γ:

$$N_{\max} = \underset{i}{MAX}\left\{\gamma \ \ \underset{j}{MAX}\left[M_j \ B_j\right]\right\} \tag{1026}$$

When the option is set to **Maximum Limits**, the model sets $N_{\max}$ to the value of the *Maximum Nodal Diameter* property.

See Also:

Harmonic Balance Solution View Field Functions

Fictitious Rotation Rate Computation

When you use **Consider All Blade Rows** as your *Frequency Compute Option*, every blade must be assigned a different frequency for the purposes of computation, even if some have identical frequencies in the **User Specified Mode Table**. The *Fictitious Rotation Rate Factor* ε determines the offset from the originally specified rotation rate.

The offset is an integer multiple of f where:

$$f = \varepsilon \Omega_{\max} \tag{1027}$$

$\Omega_{\max}$ is the maximum user-specified rotation rate in the blade row. The default value of ε is 0.001. You can set the value of ε in the *Fictitious Rotation Rate Factor* property of the **Blade Rows** manager node, but it is best to make the value small, so that the departure from the original value stays small.

See Also:

What Is a Blade Row?

Fluttering Motion of a Rotor

Consider the fluttering motion of a rotor. In the rotor frame of reference, an inter-blade phase angle σ, and a frequency ω describe the motion. For example,

$$h_i = h_0 e^{j\omega t} e^{\left(j\sigma\theta_i\right)/G_\theta} \tag{1028}$$

where G_θ is the blade-to-blade gap (in radians). Written in terms of nodal diameters, Eqn. (1028) becomes

$$h_i = h_0 e^{j\omega t} e^{jN\theta_i} \tag{1029}$$

where $N = \sigma/G_\theta$. This motion represents a traveling wave, which is like the motion of a rotor. In this case, we have an N nodal diameter wave moving with a rotational rate of

$$\Omega_{rel} = -\omega/N = -\left(\omega G_\theta\right)/\sigma \tag{1030}$$

This rotor rate is relative to the frame of reference of the fluttering rotor. Therefore, the rotation rate in the absolute reference frame is

$$\Omega_{abs} = \Omega_{rotor} - \omega/N \tag{1031}$$

To compute the frequencies and nodal diameters, another row is added to the mode table. For example, consider a stator with 36 blades and a rotor with 42 blades, rotating at 366 rad/s. Assume that the rotor blade is fluttering with an inter-blade phase angle of 180 ° with a frequency of 2000 rad/s. To construct the mode table:

1. Compute the nodal diameter N for the flutter motion:

   $N = \sigma/G_\theta = (180°)/((360°)/42) = 21$

2. Compute the rate of disturbance Ω_{abs}:

   $\Omega_{abs} = \Omega_{rotor} - \omega/N = 366\ \mathrm{rad/s} - (2000\ \mathrm{rad/s})/21 = 270.76\ \mathrm{rad/s}$

3. Construct the mode table with three rows:

   $\left(B_0, B_1, B_2\right) = (21, 36, 42)$

   $\left(\Omega_0, \Omega_1, \Omega_2\right) = (270.76, 0.0, 366.0)\ \mathrm{rad/s}$

See Also:

Working with Mode Tables

Harmonic Balance Bibliography

[145] *Hall, Kenneth C., Thomas, Jeffrey P. and Clark, W. S. 2002. "Computation of Unsteady Nonlinear Flows in Cascades Using a Harmonic Balance Technique." AIAA Journal 40:5 pp. 879-886.*

[146] *Giles, M. B. 1988 "Calculation of Unsteady Wake/Rotor Interaction", J Propulsion and Power, 4:4 pp. 356–362*

Harmonic Balance Field Functions Reference

The following primitive field functions include Harmonic Balance variants of density, entropy, pressure, temperature, and velocity.

They, along with frequency, are activated by the Harmonic Balance Flow and Energy Model. Modified diffusivity and turbulent viscosity are activated by one of the Harmonic Balance Spalart-Allmaras models.

[Value] at Time-level X	Solutions that are computed at each discrete time level (total number of time levels N = 2M + 1, where M is the number of modes)
[Value] Fourier Mode 0	The zeroth harmonic or the “mean” flow.
[Value] Fourier Mode X Imag	The Xth imaginary component of the Fourier coefficients.
[Value] Fourier Mode X Real	The Xth real component of the Fourier coefficients.
[Value] Harmonic Time Function	The flow quantity at a particular physical time (in seconds) specified by an entry in this field function object’s *Time* property. To calculate the flow using only the mean mode, use the *Mean Flow Reconstruction* property. See Eqn. (1018).
[Value] Harmonic Time Phase Lagged Down Function	The flow quantity at a particular physical time (in seconds) specified by an entry in this field function object’s *Time* property, which is phase-lagged against the direction of rotation. To calculate the flow using only the mean mode, use the *Mean Flow Reconstruction* property. See Eqn. (1018).
[Value] Harmonic Time Phase Lagged Up Function	The flow quantity at a particular physical time (in seconds) specified by an entry in this field function object’s *Time* property, which is phase-lagged in the direction of rotation. To calculate the flow using only the mean mode, use the *Mean Flow Reconstruction* property. See Eqn. (1018).
[Value] Level X Phase Lagged Down	The Xth time level solution which is phase-lagged with respect to the user-specified phase angle, sigma, using -sigma for the phase-lagging.
[Value] Level X Phase Lagged Up	The Xth time level solution which is phase-lagged with respect to the user-specified phase angle, sigma, using +sigma for the phase-lagging.
Displacement at Time-level k	Displacement at the k th time level, $dx(t_k)$ in Eqn. (1022).

Fundamental Frequency	The greatest common divisor of all the unsteady frequencies present in the blade row.

Harmonic Balance Time Field Function Properties

Time	The physical time, in seconds, at which flow is calculated.	
Inverse Distance Weight	Determines whether to use inverse distance weighting to interpolate node values.	
	Activated	Interpolate node values using inverse distance weighting.
	Deactivated	Use nearest neighbor as node values. This is the default.
Mean Flow Reconstruction	When activated, calculation of the variable in question uses only terms for the mean mode and any mode that is artificially introduced because of the Consider All Blade Rows option. See Eqn. (1018).	
	Activated	Calculates values using mean and introduced mode terms.
	Deactivated	Calculates values using all terms. This is the default.

Contents:

Harmonic Balance Solution View Field Functions

See Also:

What Is the Harmonic Balance Flow and Energy Model?

Harmonic Balance Solution View Field Functions

The following primitive field functions are for visualization in harmonic balance solution views.

- **HB: Density**
- **HB: Entropy**
- **HB: Grid Relative Velocity**
- **HB: Mach Number**
- **HB: Mass Flow Rate**
- **HB: Mass Imbalance**
- **HB: Modified Diffusivity**
- **HB: Pressure**
- **HB: Speed of Sound**

- **HB: Temperature**
- **HB: Total Enthalpy**
- **HB: Total Pressure**
- **HB: Total Temperature**
- **HB: Turbulent Viscosity**
- **HB: Turbulent Viscosity Ratio**
- **HB: Velocity**

See Also:

Blade Row Nodal Diameter Computation
What Is the HB Solution View?

Solid Stress

STAR-CCM+ allows you to model the response of a solid continuum to applied loads, including body forces, surface loads, point forces, or thermal loads that result from changes in the solid temperature.

The **Solid Stress** model computes the displacement of the solid structure using the finite element method and then calculates the corresponding stresses and strains. The fundamentals of Solid Mechanics and the finite element implementation in STAR-CCM+ are presented in the Solid Mechanics section of the Theory Guide.

Additional models target specific applications, such as nonlinear geometry, thermal strain, and Fluid-Structure Interaction (FSI). In thermal strain and FSI simulations, you can use the finite element solid stress models in combination with the finite volume fluid models.

Stress analysis simulations rely on a parts-based workflow, including:

- Parts-based mesh operations
- Contact-based interfaces for multi-part assemblies and FSI
- Loads and constraints on part surfaces and points using segments

Overview

Target Applications

- Static or dynamic stress analysis with linear elastic materials and:
  - single parts or multi-part assemblies
  - linear or nonlinear geometry
  - mechanical and/or thermal loads
- One-way or two-way coupled Fluid-Structure Interaction (FSI) analysis

Mesh Requirements

Supported Mesh:

- 3D linear and quadratic finite elements:
  - Hexahedra (Hex8, Hex20)
  - Tetrahedra (Tet4, Tet10)
  - Wedges (Wedge6, Wedge15)
  - Pyramids (Pyramid5, Pyramid13)

Mesh Generation:

- Import of CAE model with compatible mesh
- STAR-CCM+ parts-based mesh operations

Parts/Regions Layout

- Single Part or Multi-Part Assembly
- Parts-based interfaces:
  - Solid/Solid bonded and small sliding frictionless interfaces between solid parts
  - Solid/Fluid Interfaces for FSI applications

Loads and Constraints

With the **Solid Stress** model, STAR-CCM+ introduces a new simulation object called a *segment*, which applies loads and constraints on surfaces and vertices.

- Mechanical loads:
  - Body loads (applied as region inputs)
  - Loads on part surfaces or part points (applied using segments)
  - Centrifugal loads (applied using moving reference frames)
- Thermal Loads (applied as region inputs):

  The temperature at vertices is mapped from a FV Conjugate Heat Transfer analysis in STAR-CCM+, or alternatively, specified through tabular data or a field function.
- Constraints:
  - Prescribed displacement on part surfaces or part points (applied using segments)
  - Symmetry plane boundary condition (sets the displacements normal to the symmetry plane to zero)

Solution

Sparse Direct Solver:

- PARDISO for serial execution [149].
- MUMPS for either parallel or serial execution [150]

Contents:

General Workflow for Stress Analysis
Modeling Fluid-Structure Interaction (FSI)
Solid Stress Reference
Mesh Requirements and Guidelines
Solution Analysis Guidelines
Bibliography

General Workflow for Stress Analysis

The following steps outline the recommended workflow for stress analysis in STAR-CCM+. These steps assume that you are already familiar with general STAR-CCM+ tasks, such as creating parts, assigning parts to regions, or creating reports and scenes.

For practical examples of stress analysis in STAR-CCM+, refer to the tutorials:

- Linear Stress Analysis: Cantilever I Beam
- Plane Stress: Plate with Circular Hole in a Tensile Field
- Thermal Strain: Exhaust Manifold
- Fluid Structure Interaction: Vibrating Pipe

in the Solid Stress section of the STAR-CCM+ Tutorial Guide.

Contents:

Preparing the Geometry
Defining the Regions Layout
Selecting Solid Stress Physics Models
Defining the Solid Materials
Generating a Mesh of 3D Finite Elements
Applying Mechanical Loads
Applying Thermal Loads
Applying Constraints
Preparing Analysis Objects and Running the Simulation
Modeling Nonlinear Geometry

Preparing the Geometry

Create or import the structure geometry in STAR-CCM+ as a single part or a multi-part assembly.

Before you set up the simulation in STAR-CCM+, identify the relevant features of the problem, including the properties of the structure, the time scales, the type of loading, how the structure is supported, and any potential environmental influences. Identify a suitable representation of the problem, including the physical domain, the type of analysis (for example, static or dynamic), any symmetry in the problem, the boundary conditions, and any other assumptions.

Prepare or import the geometry in STAR-CCM+:

1. Launch STAR-CCM+ and start a simulation.
2. Prepare the geometry using 3D-CAD or a third-party software of your choice, defining bodies and surfaces as required:
   a) If the structure is a single material solid, create a single body. If the structure is a multi-material assembly, create a body for each component, or group of components, of the same material.

If the geometry and the physics have cyclic symmetry, you can model a sector of the geometry and use periodic contacts to represent the full part. Instructions are provided in this section.

b) Make sure that the surfaces where you apply loads, constraints, or symmetry conditions, have their own patch and name.

c) If you use 3D-CAD, you can rename the body vertices to make them available as part points, for application of point loads and point constraints. For information, see Setting Face, Edge, and Vertex Names.

Note: In 3D-CAD, you can create additional vertices at any position on the solid body, by imprinting the sketch of an appropriate closed loop onto the solid body.

d) For Fluid-Structure Interaction simulations, also create the required bodies and surfaces to represent the fluid.

e) To limit the computational cost, remove any non-essential CAD features.

When preparing the geometry in 3D-CAD, you can access geometry information that can be useful in stress analysis. This information includes the position of a vertex, the area and surface normal of a surface, and the volume, centre of gravity, and moment of inertia of a body. For more information, see Viewing the Properties of Geometric Objects.

3. Create geometry parts:

- If you prepared the geometry using 3D-CAD, right-click the **Geometry** > **3D-CAD Models** > **[3D-CAD Model]** node and select **New Geometry Part**. In the *Parts Creation Options* dialog, click **OK**.

  STAR-CCM+ creates a geometry part for each body defined in the 3D-CAD model. For each named surface and named vertex of a body, STAR-CCM+ creates corresponding part surfaces and part points. STAR-CCM+ also creates part contacts from coincident surfaces in the 3D-CAD model, such as contact surfaces in a multi-body assembly, or contact surfaces between a solid and a fluid body.

- If you prepared the geometry using a third-party tool, import the geometry in the simulation, using one of the following methods:

  - Import a surface mesh as a part.

    As the FE solid stress framework relies on a parts-based approach, make sure that you set the *Import Mode* to **Create New Part**. During the import process, STAR-CCM+ automatically creates geometry parts and part surfaces based on the surface mesh specifications, and generates part contacts from coincident surfaces. See Importing Surface Data into STAR-CCM+.

  - Import a CAE model with a compatible mesh of 3D solid elements.

    During the import process, STAR-CCM+ identifies the model volume, surface, and vertex grouping specifications (for example, node sets or constraints) and creates corresponding parts, part surfaces, and part points under the **Geometry** > **Parts** node. See Importing a CAE Model: Workflow. For more information on the mesh requirements for stress analysis in STAR-CCM+, see Mesh Requirements and Guidelines.

  Make sure that the imported surface mesh, or imported CAE model, contains volume, surface, and node set definitions as required by your analysis.

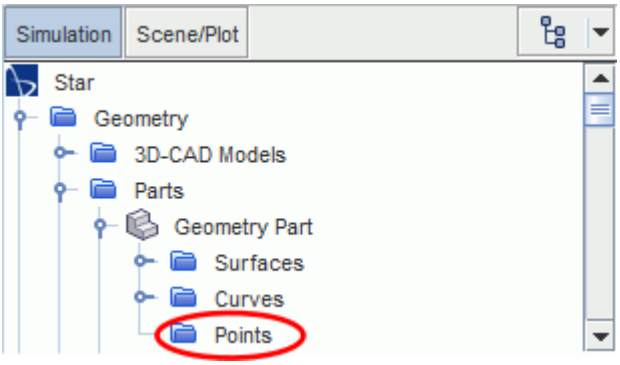
4. For structures with cyclic symmetry, where you only model a sector of the geometry, create a periodic contact to represent the full part. To create a periodic contact:

a) Multi-select the master and slave surfaces of the part that represents the sector. See (Periodic Contacts).

The master and slave surfaces must belong to the same geometry part.

b) Right-click one of the selected part surfaces and select **Create Periodic**.

5. If you wish to apply loads and constraints at points, define part points under the relevant geometry part.



Unlike the **Surfaces** and **Curves** nodes, the **Points** node does not appear under geometry parts by default. The **Points** node appears in the simulation when you either:

- Create the geometry part from a 3D-CAD model containing named vertices—STAR-CCM+ automatically adds the **Points** node and populates it with a part point for each named vertex, or named group of vertices. You can also create additional part points.
- Manually enable part points—STAR-CCM+ adds an empty **Points** node, where you can create part points.
- Import a CAE model containing vertex specifications—STAR-CCM+ automatically adds the **Points** node and populates it with a part point for each named group of vertices defined in the CAE model.

To manually enable part points:

a) Right-click the **Parts** node and select **Enable Part Points**.

STAR-CCM+ adds the **Points** node to all parts in the simulation.

To create part points under the **Points** node, you can either:

- Manually mark vertices as feature vertices and assign them to part points, using the Surface Repair tool. See Repair Tools Reference.
- Automatically compute the feature vertices that lie on selected surfaces, or edges, and assign them to part points

To automatically compute feature vertices and create corresponding part points:

a) Right-click the **Parts** node and select **Compute Part Points...**

b) In the *Mark Part Points Options* dialog, select the input surfaces, or curves, and specify which vertices STAR-CCM+ marks as feature vertices:

Part Point Option	Description
Sharp Vertices Along Part Curves	When activated, STAR-CCM+ marks vertices that connect two feature curves only if the external angle between the feature curves is greater than the specified *Angle Between Edges*.
Curve Free Ends	When activated, STAR-CCM+ marks vertices that are connected with a single feature curve.

Part Point Option	Description
Curve Junctions	When activated, STAR-CCM+ marks vertices that are connected to more than two feature curves.
Curve Type Perimeters	When activated, STAR-CCM+ marks vertices that are connected to two different types of feature curves.
Maintain existing Part Points	When activated, STAR-CCM+ preserves existing part points.

c) Click **Apply** then **Close**.

STAR-CCM+ computes feature vertices on the selected surfaces and curves, based on the specified criteria, and makes them available as part points.

You can also create an empty part point and assign feature vertices to it using the Surface Repair Tool. To create an empty Part Point:

6. Right-click the **Parts** node and select **New Part Point...**

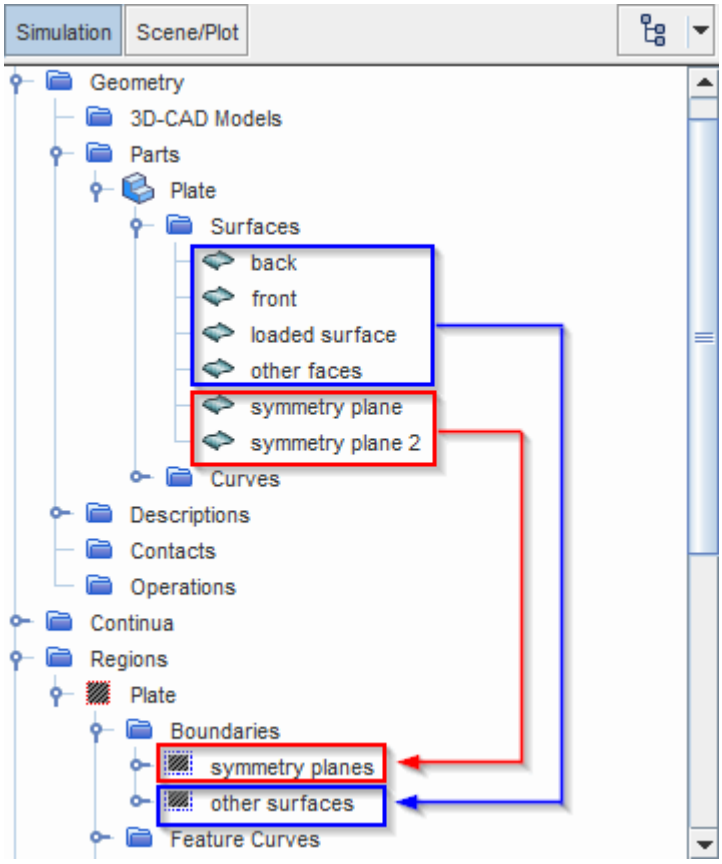
   For information on the Surface Repair tool, see Repair Tools Reference.

You can delete, rename, combine and add part points to a scene, as you do for part surfaces and part curves.

Defining the Regions Layout

Create the required regions, boundaries, and interfaces.

1. Assign the geometry parts representing the solid structure to a region, creating appropriate boundaries and interfaces.
   - If the simulation contains a fluid part for Fluid-Structure Interaction (FSI), assign it to a separate region and create the fluid-structure interface.
   - If the solid part has periodic contacts for cyclic symmetry (see Preparing the Geometry), create an interface from each periodic contact.
   - For simulations with a large number of parts, keep the number of boundaries and interfaces to a minimum. For the solid region, you can assign part surfaces that are subject to different physics conditions to the same boundary, as physics conditions are defined directly on part surfaces and points. In general, the solid region requires two boundaries—one boundary for all part surfaces representing symmetry planes, and one boundary for the remaining part surfaces.



For instructions on creating regions, boundaries, and interfaces, see Regions Layout Workflow.

Note: When you import a CAE model, STAR-CCM+ automatically creates corresponding regions and boundaries. For regions associated with imported CAE models, you can combine boundaries as outlined in the example above.

2. For each solid boundary, set *Type* to either **Wall** or **Symmetry Plane**. See Boundary Types.

   STAR-CCM+ automatically constrains part surfaces that are assigned to symmetry boundaries, preventing displacement in the normal direction. See Applying Constraints.

If you created any interfaces:

3. For each **Interfaces** > **[Interface]** node, set *Type* to one of the following:

Mapped Contact Interface	This type of interface supports both conformal and non-conformal mesh and is suitable for: • solid/solid interfaces • fluid/solid interfaces for FSI applications (see Modeling FSI) • periodic interfaces
Contact Interface	These types of interface require a conformal mesh and are suitable for combining stress analysis with applications that do not support mapped contact interfaces. For example, when using the **Solid Stress** model in combination with electromagnetic models. See Selecting Solid Stress Physics Models.
Internal Interface	

	Not suitable for FSI interfaces. Can be used for periodic interfaces, provided that the mesh is conformal.

If you created any periodic interfaces:

4. Select the **Interfaces** > **[Periodic Interface]** node and set *Topology* to **Periodic**.
5. Select the **[Periodic Interface]** > **Periodic Transformation** node and define the transformation as required. For more information on the available properties, see Periodic Interface Topology.

   STAR-CCM+ always treats periodic interfaces between solids as **Bonded**, that is, the contacting surfaces undergo continuous displacements. See Interface Settings.

Selecting Solid Stress Physics Models

Activate physics models for stress analysis and select the material laws that define the behavior of the solid.

1. Create a physics continuum and assign it to the region that represents the solid material.
2. In the physics continuum, activate the following physics models:

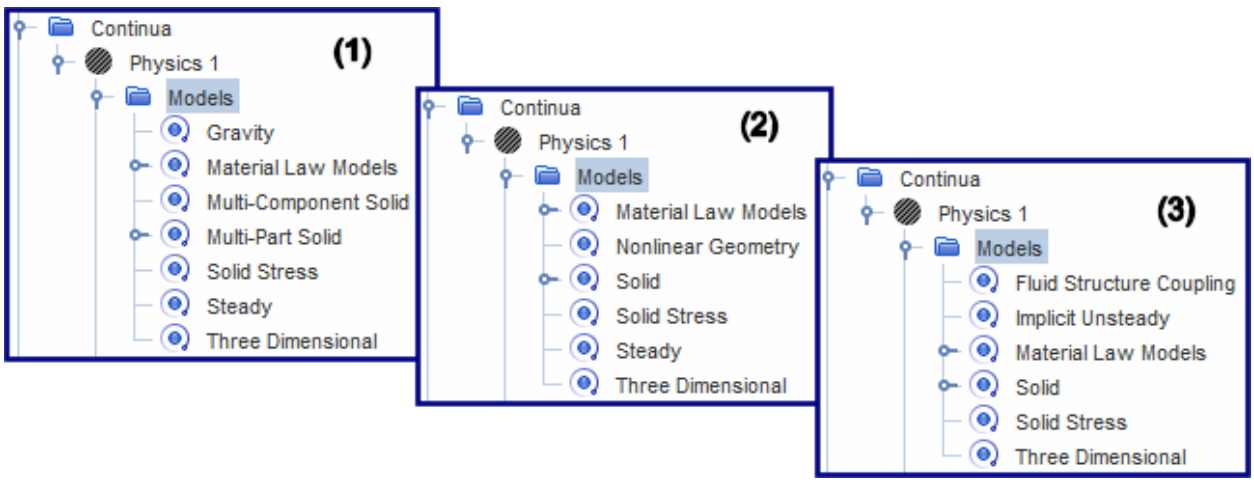
Group Box	Physics Model
Space	**Three Dimensional**
Time	• For static analysis, activate the **Steady** model Static simulations include time-independent analyses and time-dependent analyses for which the time scales to load or to unload the structure are very long compared to the time scales for propagation of flexural or sound waves. See Static and Dynamic Solutions. • For dynamic analysis, where the transients due to the low-frequency flexural waves or high-frequency sound waves are important, activate the **Implicit Unsteady** model. • For quasi-static analysis, where you seek the static solution while having the option to apply the loads progressively over time, activate the **Implicit Unsteady** model in *Static Analysis* mode. See Solid Stress Solver Reference.
Material	• To model a single material solid part, activate the **Solid** model. • To model an assembly of solid parts of different materials, activate the **Multi-Component Solid** > **Multi-Part Solid** model.
Optional Models	**Solid Stress**
	Material Law Models (selected automatically) STAR-CCM+ supports linear elastic materials.

3. Activate additional models from the *Optional Models* group box, based on analysis requirements:
   - To account for the solid displacement induced by fluid forces, activate the **Fluid Structure Coupling** model.
   - To account for thermal expansion, activate the **Specified Temperature Load** model.

- To add mass and stiffness proportional damping, activate the **Rayleigh Damping** model.
- To account for gravity, activate the **Gravity** model.
- By default, the **Solid Stress** model accounts for small deformations in the structure (linear geometry). To model nonlinearities due to large displacements and rotations, activate the **Nonlinear Geometry** model.
- To use electromagnetic forces as body loads for stress analysis, activate the **Electromagnetism** model and any relevant submodels. The **Solid Stress** model is compatible with the **Finite Element Magnetic Vector Potential** model, the **Electrodynamic Potential** model, and their optional models. For more information on setting up electromagnetic analyses, see Modeling Magnetic Fields and Modeling Electric Currents.

For details on the properties and dependencies of the solid stress physics models, see Solid Stress Reference.

Examples of model selections are shown below:



(1)—Static stress analysis of a multi-material solid, including the effect of gravity, with the assumption of small deformations (linear geometry).
(2)—Static stress analysis of a single-material solid, including nonlinearities due to large displacements and rotations (nonlinear geometry).
(3)—Dynamic stress analysis of a single-material solid, subject to fluid forces at the fluid-structure interface (FSI). The physics continuum representing the fluid is not shown in example **(3)**.

At mapped contact interfaces between solid parts, specify the type of mechanical interaction:

4. Select the **[Solid/Solid Interface]** > **Physics Conditions** > **Mechanical Interaction** node and set *Method* to either **Bonded** or **Small Sliding Frictionless**.

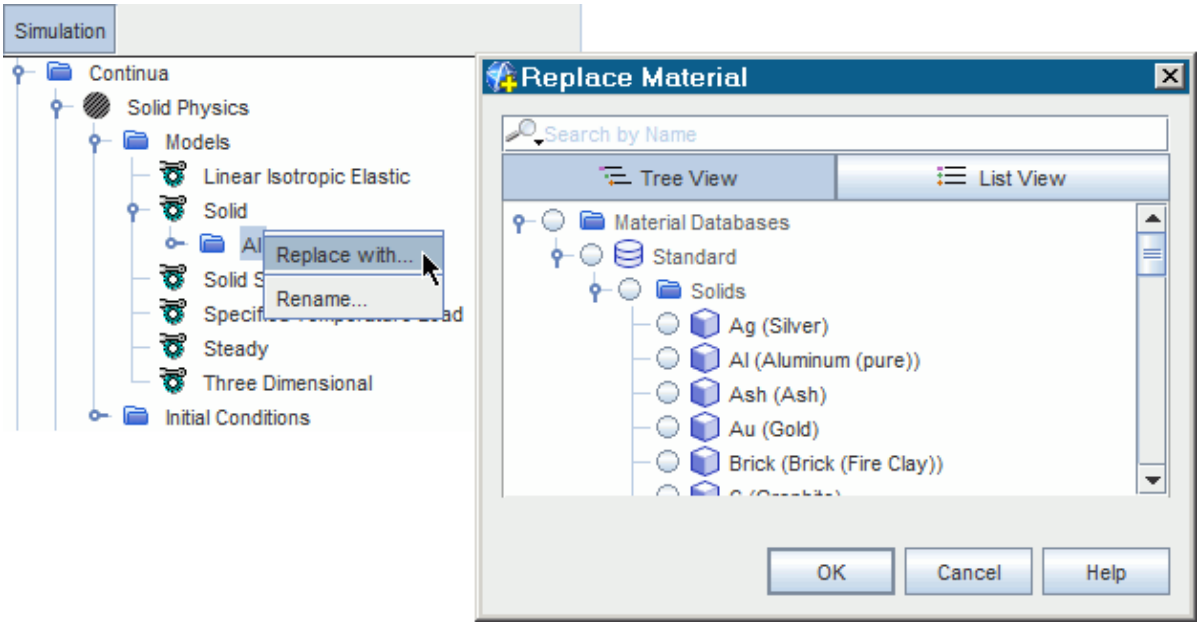
   For more information, see Interface Settings.

Defining the Solid Materials

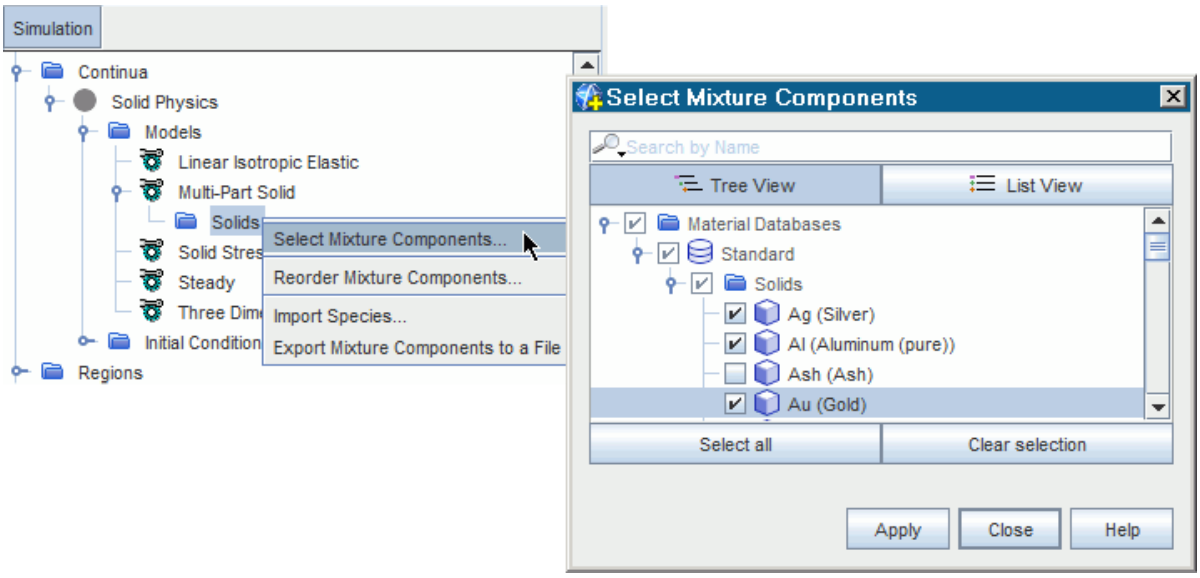
Specify the materials for the solid structure and define their properties.

1. Select the solid materials from the STAR-CCM+ material library:

- If you selected the **Solid** model, choose a single material for the solid part.



- If you selected the **Multi-Part Solid** model, add a material for each part in the assembly.



Assign the materials to the corresponding parts using the **[Region]** > **Physics Values** > **Material Part Groups** node. For more information, see Modeling a Single-Component Material and Modeling Multi-Part Solids.

For each material, you define the relationship between stress and strain by specifying an appropriate material law (see Constitutive Laws). STAR-CCM+ supports linear elastic materials, with either isotropic, orthotropic, or anisotropic properties.

To define a material law:

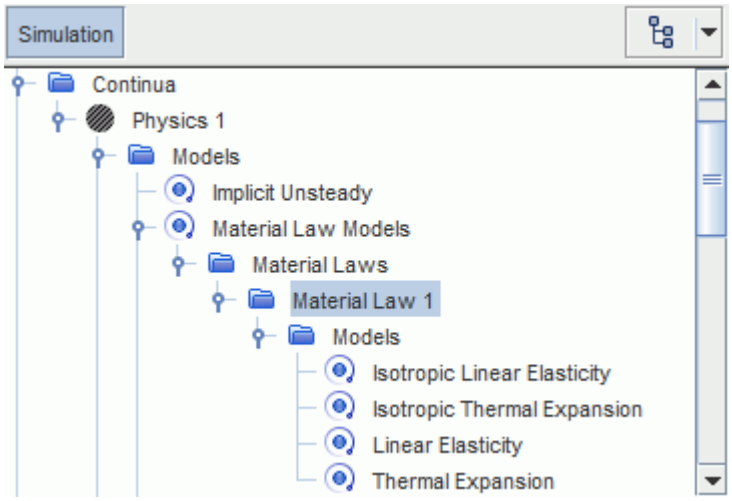
2. Expand the relevant **[Physics Continuum]** > **Models** > **Material Law Models** node.
3. Right-click the **Material Laws** node and select **New**.

   STAR-CCM+ creates a material law node with the default name, **[Material Law 1]**. This node allows you to select the models that approximate the material behavior. You do not require a separate material law for each material. Multiple materials can be described using the same material law.

Choose the models that define the material law:

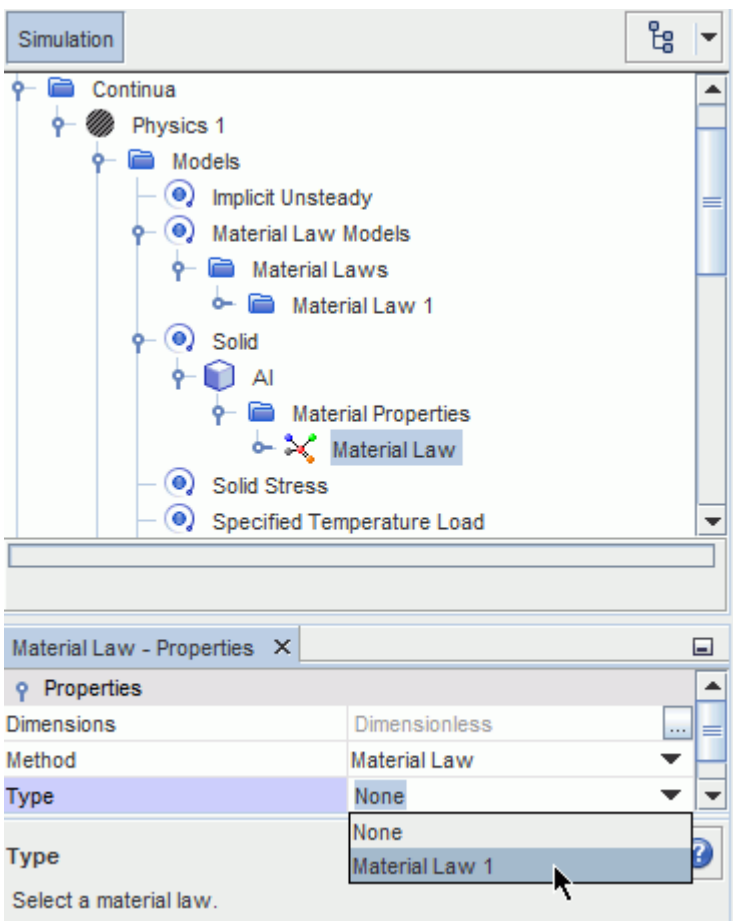
4. Right-click the **Material Laws** > **[Material Law 1]** > **Models** node and choose **Select models...**
5. In the *Material Law 1 Model Selection* dialog, activate the required models:

Group box	Model
Material Stiffness Models	Activate **Linear Elasticity**.
Linear Elastic Material Models	• To model materials that have the same mechanical properties in all directions, activate **Isotropic Linear Elastic**. • To model materials that have independent mechanical properties along three mutually-orthogonal directions, activate **Orthotropic Linear Elastic**. • To model materials that have independent mechanical properties in all directions, activate **Anisotropic Linear Elastic**. For more information, see Linear Elastic Materials.
Optional Models	In thermal strain analyses, also activate the **Thermal Expansion** model. This model requires the **Specified Temperature Load** model.
Thermal Expansion Material Models	When using the **Thermal Expansion** model, activate one of the following: • To model materials that have the same thermal properties in all directions, activate **Isotropic Thermal Expansion**. • To model materials that have independent thermal properties along three mutually-orthogonal directions, activate **Orthotropic Thermal Expansion**. • To model materials that have independent thermal properties in all directions, activate **Anisotropic Thermal Expansion**.

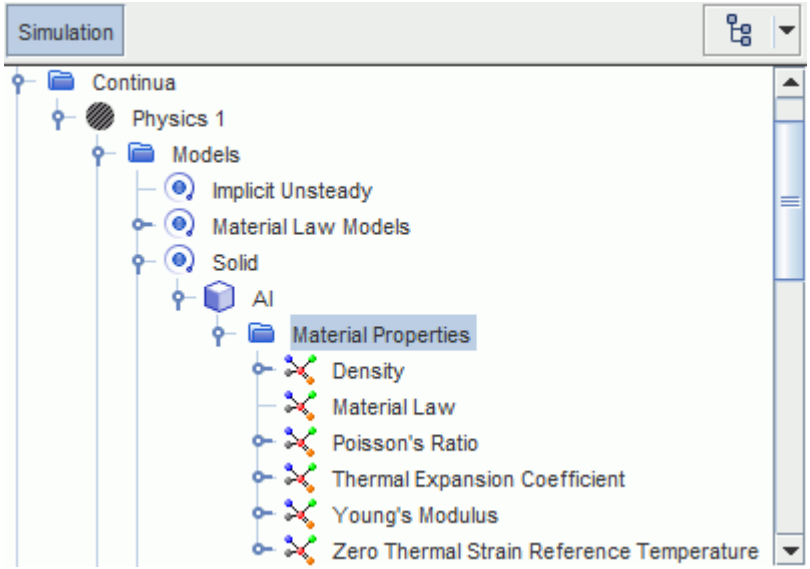


To assign the material law to a solid material:

6. Expand the relevant **[Physics Continuum]** > **Models** > **[Solid]** > **[Material]** node.
7. Select the **Material Properties** > **Material Law** node and set *Type* to **[Material Law 1]**.



STAR-CCM+ adds the relevant material properties under the **Material Properties** node, based on the specified material law.



8. Following the same procedure, assign a material law to all the materials defined in the simulation. Different materials can share the same material law.

When a material is defined by an orthotropic or anisotropic material law, you define its properties with respect to a specified coordinate system. To specify the coordinate system:

- Select the **Material Properties** > **Material Law** node and set *Coordinate System* as required. For information on creating coordinate systems, see Creating a Local Coordinate System.

  You can visualize the material coordinate system in a scene using the basis vector field functions. See Local Material Coordinate System.

This step is not required for isotropic materials, as their properties are the same in all directions.

9. Define the mechanical and thermal material properties as required. For each material:
   a) Expand the **Material Properties** node.
   b) Specify the material **Density**.
   c) For thermal strain analysis, specify the **Zero Thermal Strain Reference Temperature**, which is the temperature for which the thermal strain of the material is assumed to be zero.
   d) Specify the remaining properties based on the type of material:

Isotropic	1. Specify the **Poisson's Ratio** and **Young's Modulus**. See Eqn. (3830). 2. For thermal strain analysis, also specify the **Thermal Expansion Coefficient**. See Eqn. (3831).
Orthotropic	1. Specify the Poisson ratio, shear modulus, and Young's modulus along each axis of the specified coordinate system (see Eqn. (3823)): a. Expand the **Poisson Ratios** > **Poisson Coefficients** node and specify the three coefficients **nu12**, **nu23**, and **nu13**.

	b. Expand the **Shear Moduli** > **Shear Modulus Coefficients** node and define the three coefficients **G12**, **G23**, and **G13**. c. Expand the **Young's Moduli** > **Young's Modulus Coefficients** node and define the three coefficients **E12**, **E23**, and **E13**. 2. For thermal strain analysis, also expand the **Orthotropic Thermal Expansion Coefficients** node and specify the three coefficients **alpha 1**, **alpha 2**, and **alpha 3**. See Eqn. (3827).
Anisotropic	1. Expand the **Anisotropic Stiffness Coefficients** > **Anisotropic Material Stiffness Tangent Coefficients** node and specify the independent components of the material tangent matrix. See Eqn. (3821). 2. In thermal strain analysis, also expand the **Anisotropic Thermal Expansion Coefficients** node and specify the six components of the thermal expansion coefficients vector. See Eqn. (3821).

For more information, see Material Properties.

Generating a Mesh of 3D Finite Elements

This section outlines the steps for generating the volume mesh in STAR-CCM+ using parts-based mesh operations. If you imported a CAE model volume mesh, and you wish to remesh it in STAR-CCM+, you can use the same procedure.

STAR-CCM+ solid stress simulations require a mesh of 3D linear or quadratic elements. To generate the mesh in STAR-CCM+:

1. Right-click the **Geometry** > **Operations** node and create one of the following mesh operations:
   - **Directed Mesh**, to generate hexahedral elements.

     For information on the Directed Mesher, see Directed Meshing. A step-by-step example is also provided in the *Solid Stress > Linear Stress Analysis: Cantilever I Beam* section of the Tutorial Guide.
   - **Automated Mesh**, to generate tetrahedral or wedge elements.

     For this operation, you can either select the **Tetrahedral Mesher** to generate tetrahedral elements, or the **Tetrahedral Mesher** with the **Thin Mesher**. The latter combination generates wedge elements in regular thin geometries, such as a thin plate, or a mix of different elements in more complex geometries. In addition to tetrahedra and wedges, occasional pyramids can be generated as transition elements.

   Set up the mesh controls based on your geometry requirements, and make sure that you use an appropriate mesh resolution in areas where you expect stress concentrations. For general guidelines, refer to the section, Mesh Requirements and Guidelines.
2. If the input parts for the mesh operations contain part points, make sure that the **Solid Stress** model is active in the relevant physics continuum.

   Note: If you do not activate the **Solid Stress** model before executing the mesh operations, STAR-CCM+ will not transfer the part points to the mesh representation.
3. Execute the mesh operation.

By default, the mesh operations generate linear elements. To generate quadratic elements, add the mid-side nodes as interpolates of the corner nodes:

4. Select the **Regions** > **[Solid Region]** > **Physics Conditions** > **Mid-side Vertex Option** node.
5. Set *Mid-Side Vertex* to **Add -- Linear Interpolation**.

   STAR-CCM+ adds the mid-side nodes during initialization of the solution.

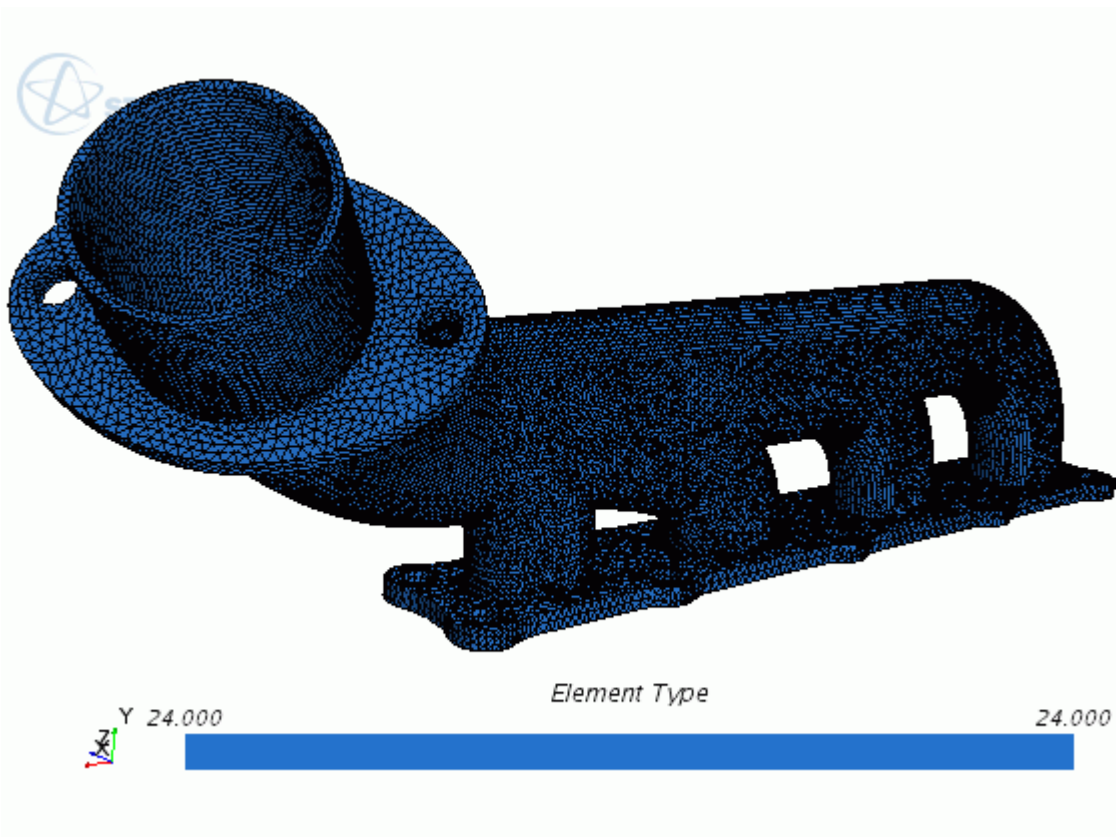
You can visualize the mesh element topology in a scalar scene using the **Element Type** field function, which identifies each element topology with a unique number.

6. Create a scalar scene with the following settings:

Node	Property	Setting
Displayers > **Scalar 1**	*Display Mesh*	**Activated**
└ **Parts**	*Parts*	**Parts** > **[Solid Part/Parts]**
└ **Field Function**	*Function*	**Element Type**

The Element Type field function values for each element topology are listed in the section, Element Type Reference.

The following example shows a tetrahedral mesh with mid-side nodes (Tet10). For Tet10 elements, the Element Type value is 24.



Contents:

Specifying the Orientation of Non-Conformal Interfaces

Specifying the Orientation of Non-Conformal Interfaces

In multi-part assemblies, STAR-CCM+ automatically handles non-conformal solid-solid interfaces based on a master-slave algorithm. In some applications, the presence of multiple interacting non-conformal interfaces may lead to over-constraining of the solid parts. In these cases, STAR-CCM+ allows you to manually set the interface orientation to specify the master and slave boundaries.

In the master-slave algorithm for non-conformal mesh, the slave side of the interface corresponds to the boundary which is more flexible, that is, the boundary with a finer mesh. STAR-CCM+ automatically specifies the slave boundary based on the average dimension of the mesh element faces at each side of the interface. If the automatic specification results in over-constraining of the parts, STAR-CCM+ issues a warning message similar to the following:

```
NOTE: multiple constraints from groups of interacting solid solid
interfaces:{17,18}
NOTE: if the result is incorrect or the matrix singular, it might be
due to over/redundant constraint from interacting solid solid
interfaces.
You may use Manual Orientation to select the slaves manually. In
Manual Orientation, each slave vertex is constrained only once.
```

When STAR-CCM+ issues this warning, specify the orientation of interacting interfaces manually. For all other interfaces, leave the automatic orientation.

For interfaces that interact which other:

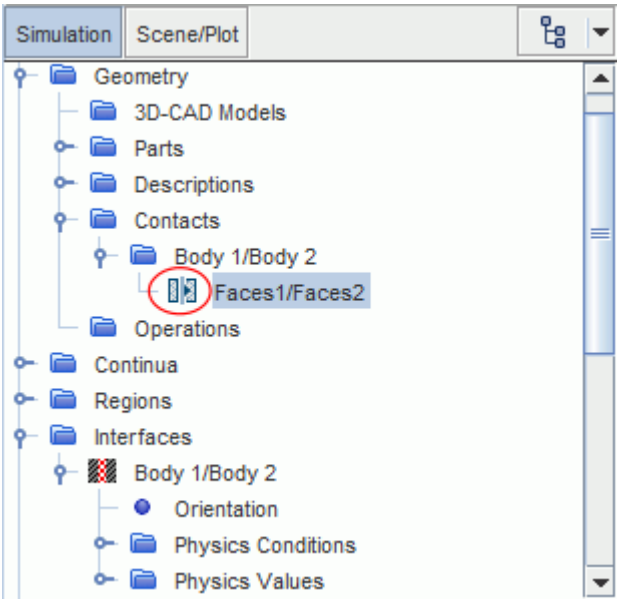
1. Select the relevant **Interfaces** > **[Interface]** > **Orientation** node and set **Mode** to **Manual**.

   The **Orientation** node is only available for interfaces between parts that are assigned to the same region.

When using manual orientation, STAR-CCM+ specifies the interface boundary **Part 1/Part 2 [0]** as the slave boundary. You can change the orientation of the contacts to specify the other side of the interface as boundary [0]:

2. Right-click the **Orientation** node and select **Reverse Contacts...**

   You can check the interface orientation by examining the icon next to the associated contact.



Applying Mechanical Loads

The FE solid mechanics framework offers three types of mechanical loads: point loads, surface loads, and body loads. You define point and surface loads on part points and part surfaces using segments, whereas body loads are specified on the solid region.

Segments are STAR-CCM+ objects that apply loads or constraints on collections of part surfaces or part points. For more information on segments, see Segments Reference.

To apply a load on surfaces or points:

1. Right-click the relevant **Regions** > **[Region]** node, select **Create Segment** and choose either:
   - **Surface Segment**, to apply a surface load on a single or multiple part surfaces.
   - **Point Segment**, to apply a point load on a single or multiple part points.

   Depending on the selection, STAR-CCM+ adds a surface or point segment under the **[Region]** > **Segments** node. After you create the first segment, you can create additional segments by either using the same procedure, or by right-clicking the **Segments** node and selecting **Create Segment**.
2. Specify the part entities where the load is applied:
   - For surface segments, specify the loaded part surfaces using the *Surfaces* property
   - For point segments, specify the loaded part points using the *Points* property

   Alternatively, you can create new segments, or assign part entities to existing segments, by using the **Set Segment** option. To access this option:
   - Expand the **Geometry** > **Parts** > **[Part]** node.
   - Select the relevant **Surfaces** > **[Part Surface]** or **Points** > **[Part Point]** node. If you wish to assign multiple part entities to the same segment, multi-select the corresponding nodes.
   - Right-click one of the selected part entities, select **Set Segment** and choose to either create a new segment or assign the selected part entities to an existing segment. When no suitable segments are available, the **Set Segment** option is replaced with **Set New Segment**.

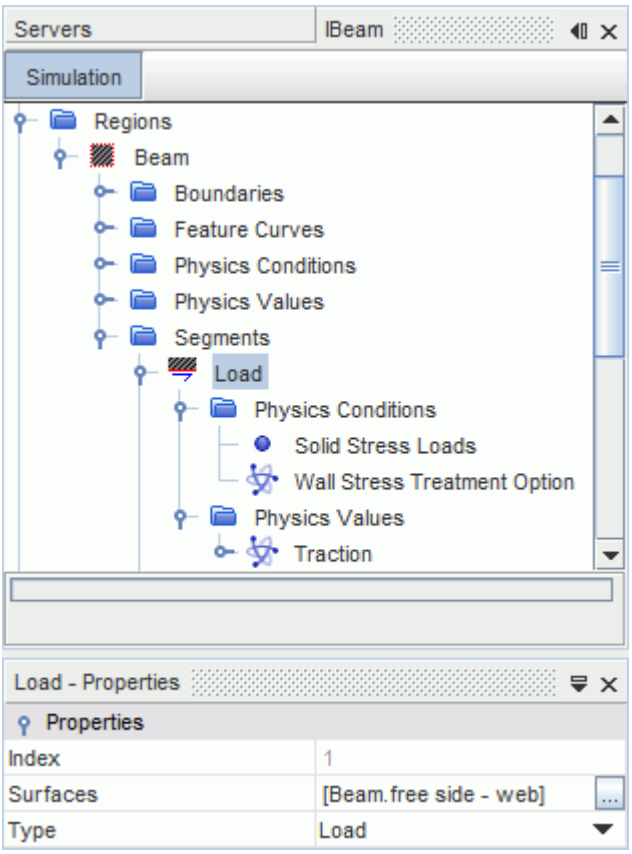
3. Select the **Segments** > **[Segment]** node and set *Type* to **Load**.

To specify the type of load:

4. Select the **[Segment]** > **Physics Conditions** > **Solid Stress Loads** node and set the loading *Method*.

   You specify a surface load as either a **Force**, **Traction**, or **Pressure** profile. You specify a point load as a **Force** profile. For more information, see Segments Reference.

5. Use the **[Segment]** > **Physics Values** > **[Force/Traction/Pressure]** node to set the load profile.



   You can specify additional loads by creating corresponding segments. If you select the same part entity (either part surface or part point) in multiple segments, STAR-CCM+ automatically computes the total load on the part entity.

In addition to the surface and point loads applied using segments, you can apply body loads on the solid region:

6. Select the **[Region]** > **Physics Conditions** > **Body Load Option** node and set *Method* to **Specified**.
7. Use the **[Region]** > **Physics Values** > **Body Load** node to define the load profile.

   You can define the body load in various ways. Common choices are constant body loads or body loads defined by field functions.

   When using the **Solid Stress** model in combination with the **Finite Element Magnetic Vector Potential** model, you can provide user-defined field functions based on electromagnetic fields.

STAR-CCM+ offers other methods to define specific types of body loads, such as gravity and centrifugal loads. For example, you can apply a gravity load by simply activating the **Gravity** model in the solid

continuum. With this method, the resulting body load is $\mathbf{F} = M\mathbf{g} = \rho V\mathbf{g}$. You can apply a centrifugal load by defining a rotating reference frame, an axis of rotation, and a constant angular velocity.

Currently, STAR-CCM+ does not offer a general motion model based on a moving reference frame. Body forces due to a change in translational and angular velocity are not accounted for. There is a slight difference in interpretation of the solid velocity and the fluid velocity in moving frames of reference. The solid velocity is the derivative of the displacement with respect to time, and the displacement is relative to the moving frame. Therefore, the velocity is also relative to the moving frame. The fluid velocity, on the other hand, is the velocity of the fluid relative to the laboratory frame, but the components are expressed in the unit vector basis associated with the moving frame. For more information, see Using Reference Frames.

Applying Thermal Loads

You can apply thermal loads on the solid region. The stress solver computes the structure displacements based on the specified temperature profile.

The thermal loads on the solid region are often the result of a conjugate heat transfer (CHT) analysis between the solid and a fluid, performed in either STAR-CCM+, a third-party software, or a combination of both. In STAR-CCM+, CHT analyses are performed using finite volume (FV) techniques, whereas stress analyses are performed using the finite element (FE) method. Therefore, to run a full thermal-structural analysis in STAR-CCM+, you require two separate regions for the solid: an FV representation of the solid for CHT analysis, and an FE representation of the solid for stress analysis. After you run the CHT analysis on the FV representation, you can map the temperature data from the FV solid mesh to the FE solid mesh. You can then use the mapped temperature data for the stress analysis.

1. In the physics continuum for stress analysis, activate the **Specified Temperature Load** physics model.

   STAR-CCM+ adds a new physics condition, **Specified Temperature Option**, under the relevant solid region.

Specify the temperature profile for the solid region:

2. Select the **Regions** > **[Solid Region]** > **Physics Conditions** > **Specified Temperature Option** node and set *Method* to **Specified**.
3. Select the **Regions** > **[Solid Region]** > **Physics Values** > **Specified Temperature** node and set *Method* as appropriate.

   If the temperature field is available on a different mesh representation of the solid, for example an imported CAE model or a STAR-CCM+ finite volume mesh, set *Method* to **Field Function** and set up a data mapper to interpolate the temperature data from the source mesh to the FE mesh. Instructions for setting up a data mapper and using the mapped temperature loads to specify the temperature profile are given below.

In thermal strain analyses, it is particularly important to define realistic constraints. A fixed condition on a surface segment prevents thermal expansion and leads to high stress concentrations.

4. Test your constraint set, to make sure that you defined realistic constraints.

   Test your constraints in two steps. As a first step, use the minimum necessary boundary conditions to prevent rigid body motion, and allow free expansion of the solid structure. Use the **Specified Temperature** option to increase the temperature uniformly (use a constant value). Run the analysis and examine the stresses, which should be very low or close to zero. As a second step, apply realistic constraints and increase the temperature again. Make sure that the stresses increase as expected.

To map temperature data from a source mesh to the FE mesh for stress analysis:

5. Right-click the **Tools** > **Data Mappers** node and select **New Data Mapper** > **Volume Data Mapper**.
6. Edit the **Volume Data Mapper** node and set the following properties:

Property	Setting
Source Volumes	Select the source mesh from which to take the temperature data. Source volumes can be STAR-CCM+ regions or parts, or volumes defined on an imported CAE model. For example, if the temperature field on the solid was computed in a STAR-CCM+ CHT analysis, select the FV solid part or parts, associated to the solid region that you used in the CHT analysis.
Source Stencil	Select either **Cell** or **Vertex** based on the location where the solution is available on the source mesh. In finite volume source meshes, the solution is typically available at cell centroids. In finite element source meshes, the solution is typically available at vertices. If the temperature distribution was computed in a STAR-CCM+ CHT analysis, select **Cell**.
Scalar Field Functions	**Temperature**
Mapped Names	When you execute the mapping operation, the mapped temperature data become available as a field function. Assign the mapped temperature field function a suitable name, using this property.

Specify the stress analysis FE mesh as the mapping target:

7. Edit the **Volume Data Mapper** > **Target Specifications** > **Volume** node and set the following properties:

Property	Setting
Target Volumes	Select the solid part, or parts, associated with the stress analysis region.
Target Stencil	**Vertex**

8. Execute the mapping operation using one of the following methods:
   - Execute the data mapper manually, after the simulation for the FV solid region is complete.
   - Set up automatic mapping at a specified frequency during the FV simulation run. In this case, set the frequency so mapping occurs once per time-step, or when the solution is complete.

   For more information, see Executing the Mapping Operation.

   STAR-CCM+ interpolates the temperature data to the target FE mesh, and generates the corresponding **[mapped_temperature]** field function.

To specify the thermal loads using the mapped temperature field:

9. Select the **Regions** > **[Solid Region]** > **Physics Values** > **Specified Temperature** node and set *Scalar Function* to the field function generated by the mapper, **[mapped_temperature]**.

Applying Constraints

Applying a constraint on a surface, or point, means prescribing one or more components of the displacement vector at that surface, or point. In stress analysis, you constrain the solid structure to mimic a real life scenario and to prevent the free displacement and rotation of the entire solid. Constraints are an effective way of reducing the number of modeled parts, and hence the simulation cost.

In STAR-CCM+, you constrain the displacement of part surfaces and part points using segments (see Segments Reference). Additionally, for part surfaces that are assigned to symmetry plane boundaries, STAR-CCM+ automatically prevents displacement in the normal direction.

To constraint surfaces or points using a segment:

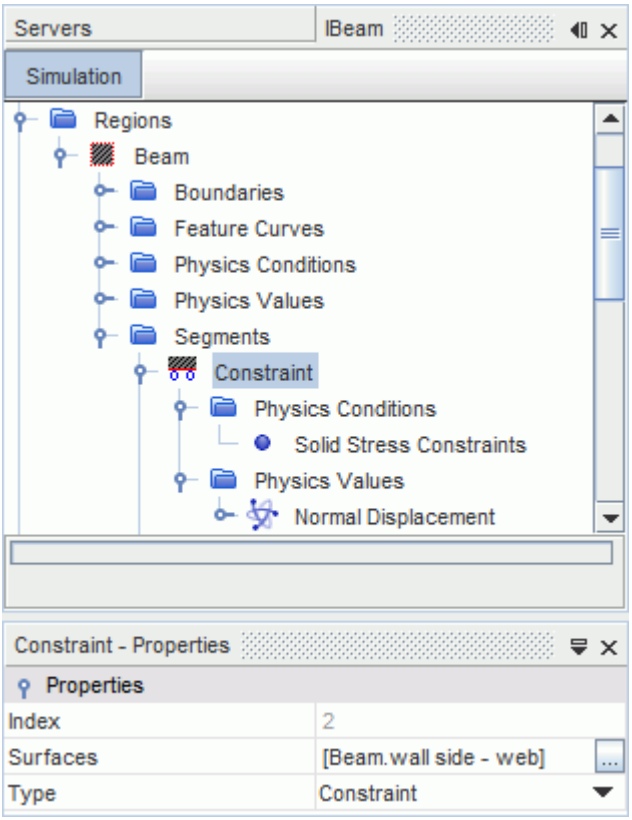
1. Right-click the relevant **Regions** > **[Region]** node, select **Create Segment** and choose either:
   - **Surface Segment**, to apply a constraint on a single or multiple part surfaces.
   - **Point Segment**, to apply a constraint on a single or multiple part points.

   Depending on the selection, STAR-CCM+ adds a surface or point segment under the **[Region]** > **Segments** node. After you create the first segment, you can create additional segments by either using the same procedure, or by right-clicking the **Segments** node and selecting **Create Segment**.
2. Specify the part entities where the constraint is applied:
   - For surface segments, specify the constrained part surfaces using the *Surfaces* property.
   - For point segments, specify the constrained part points using the *Points* property
3. Select the relevant **Segments** > **[Segment]** node and set *Type* to **Constraint**.

To specify the type of constraint:

4. Select the **[Segment]** > **Physics Conditions** > **Solid Stress Constraints** node and set the constraint *Method*.

   For more information, see Segments Reference.
5. Specify the constraint under the **[Segment]** > **Physics Values** > **[Fixed/Normal Displacement/ Displacement]** node.



Note: To prevent conflict, do not apply a nonzero normal displacement on part surfaces that are associated with symmetry boundaries. See Symmetry Plane.

You can create other segments to specify additional constraints. In general, it is good practice to constrain the structure in three orthogonal directions, by either:

- Applying a fixed constraint to at least one surface.
- Applying symmetry plane conditions, or equivalent zero normal displacement constraints, to three mutually orthogonal surfaces.
- Define three non-collinear point constraints.

In thermal strain analyses, it is important to define realistic constraints. A fixed condition on a surface segment does not allow thermal expansion and leads to local stress concentrations. To model a free expansion of a structure, define three non-collinear point constraints, with:

- the first point constrained in the x, y, and z directions
- the second point constrained in the x, and y directions
- the third point constrained in the z direction

In general, when analyzing the computed stress distribution, restrictive constraints can lead to over-estimated stresses near the constrained areas.

Preparing Analysis Objects and Running the Simulation

Prepare analysis object to monitor the solution during the simulation progress, then specify the solver settings and run the simulation.

1. Set up scalar scenes and reports to visualize the solution, and monitor the quantities that are relevant to your analysis.

   To visualize the deformation of the solid structure, you can set up a scalar scene for displacement and use a warped surface to visualize the deformed geometry. You can also set up reports that use probes to monitor key quantities at specific points, or on certain planes. For example, you can set up a report to display the value of a particular component of the stress tensor at a specified point.

   When using field functions in a scene or report, use **Smooth Values** to visualize the nodal values. For more information on the options and field functions that are available to visualize the solution, see Solution Analysis Guidelines and Solid Stress Model Reference.
2. Set the **Solid Stress Solver** properties as required:
   - To run a quasi-static analysis with the **Implicit Unsteady** solver, select the **Solid Stress Solver** > **Static Analysis** node and activate *Static Analysis*.
   - In dynamic simulations using the 2nd order temporal discretization scheme, select the **Solid Stress Solver** > **Integration Method** > **Newmark** node and specify the Newmark parameter γ.
   - When modeling nonlinear geometry, specify the method for updating the stiffness matrix using the **Stiffness Matrix Update** node.

   For more information, see Solid Stress Solver Reference.
3. Define the **Sparse Direct Solver** properties. For large cases, run the **Sparse Direct Solver** in OOC (Out-of-Core) mode, and specify a folder on your system disk that the solver can use for data storage.

   For more information, see Sparse Direct Solver Reference.

To estimate the memory that is required to run the simulation:

4. Right-click the **Solvers** > **Solid Stress Solver** node and select **Estimate Memory**.

   STAR-CCM+ reports the estimate in the *Output* window. The report includes information on the memory required to store and factorize the stiffness matrix, the model degrees of freedom, and the total memory required by the solid stress solvers.
5. Run the simulation.

Modeling Nonlinear Geometry

By default, the **Solid Stress** model accounts for linear geometry phenomena. This section describes the additional steps that are required to model phenomena for which the strain-displacement relationship is nonlinear, such as applications involving large displacements or rotations of the solid structure.

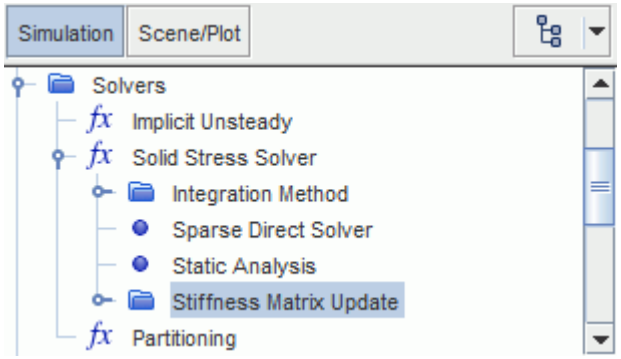
To determine whether your simulation requires a nonlinear analysis:

1. Run a linear analysis first and check the results. If either of the following occurs you require a nonlinear analysis to obtain a more realistic solution:
   - the deformations are large, that is, displacement magnitudes exceed the model dimensions by 10% or part rotations are greater than 4°
   - the structure presents slender parts loaded in tension or compression

To model nonlinear effects:

2. In the physics continuum for stress analysis, activate the **Nonlinear Geometry** model.

   STAR-CCM+ adds the **Stiffness Matrix Update** control under the **Solid Stress Solver** node.



3. In the physics continuum, activate an appropriate *Time* model, based on the type of analysis (static or dynamic) and loading method. As a guideline, ramp loads with time-dependent functions, instead of iteration-dependent functions:
   - In dynamic analyses, activate the **Implicit Unsteady** model.
   - In static analyses with small nonlinearities, you can apply loads in one step. In this case, activate either the **Steady** model, or the **Implicit Unsteady** model in *Static Analysis* mode.
   - Some static analyses with large nonlinearities can require load ramping, so that the loads are gradually applied in several steps. In this case, activate the **Implicit Unsteady** model in *Static Analysis* mode, and define time-dependent functions to apply the loads progressively.

   For more information on the behavior of loads in nonlinear simulations, see Nonlinear Geometry Model Reference.

To solve the nonlinear equations, STAR-CCM+ updates the stiffness matrix using Newton iteration methods. To specify the stiffness matrix update method:

4. Select the **Solvers** > **Solid Stress Solver** > **Stiffness Matrix Update** node and set *Method* to either **Full Newton** or **Modified Newton**.

   You are advised to use the **Modified Newton** method only for moderately nonlinear applications. For more information, see Nonlinear Geometry Model Reference.

You are advised to adjust the default stopping criteria based on the grade of nonlinearity:

5. Edit the **Stopping Criteria** node.
6. If you are using the **Implicit Unsteady** solver, you are advised to set the *Maximum Inner Iterations* to a number between 10 and 30.
7. You are also advised to increase the minimum limit for the convergence criteria, **Displacement Criterion** and **Force Criterion**, to `1E-8` or higher, in case the default values are too tight for nonlinear applications.
8. After you made the changes to the linear analysis, to take into account nonlinear geometry effects, clear the solution and run the simulation.

   Large load increments or large temperature increments can cause large distortions of elements and therefore a singular stiffness matrix. This situation leads to failure of the direct solver with an error message about a negative pivot. In these cases, reduce the simulation time-step, clear the solution and run the simulation again.

   True instabilities of a structure, such as buckling, can also result in a singular stiffness matrix. These problems require a good understanding of buckling loads and shapes.

Modeling Fluid-Structure Interaction (FSI)

STAR-CCM+ allows you to model one-way or two-way coupled FSI applications. In one-way coupled problems, STAR-CCM+ accounts for the effect of the fluid forces on the structure displacements, neglecting the effect of the structure on the fluid flow. In two-way coupled problems, STAR-CCM+ also accounts for the effect of the solid displacement on the fluid flow.

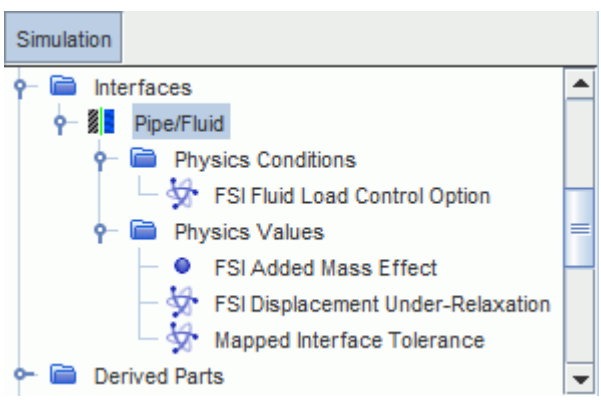
FSI simulations require the **Fluid-Structure Coupling** model, to allow for implicit coupling at the fluid-solid interface, and motion models, to account for the deformation of the fluid and solid meshes.

To model Fluid-Structure Interaction:

1. Define the fluid and solid regions and create the interface between the fluid and solid.
2. Select the relevant **Interfaces** > **[FSI Interface]** node and set *Type* to **Mapped Contact Interface**
3. Define the behavior of the fluid and the solid, as required. See Flow and Energy and General Workflow for Stress Analysis.

   For both the fluid and the solid continua, make sure that you are using a *Time* model appropriate to your analysis. In two-way coupled problems, as the velocity of the solid impacts the flow, run a time-dependent simulation by choosing the **Implicit Unsteady** model. Alternatively, if you are interested in the static deformation of the structure under loads applied by surrounding steady-state flow, choose the **Steady** model to reach the static equilibrium solution.
4. In the solid physics continuum, activate the **Fluid-Structure Coupling** model.

   STAR-CCM+ detects the fluid/solid interface and adds FSI controls.



Specify whether the fluid and the solid are one-way or two-way coupled:

5. Select the **Solvers** > **Fluid Structure Interaction** node and choose one of the following options:
   - For one-way coupled FSI problems, set *Method* to **One Way**.

     This type of simulation is suitable for FSI applications where the effects of the structure displacements on the fluid are negligible. This type of simulation is not suitable for applications where the fluid and solid physics are strongly coupled.
   - For two-way coupled FSI problems, set *Method* to **Two Way**.

This type of analysis is suitable for applications where the fluid and solid physics are weakly or strongly coupled.

Optionally, you can use the **FSI Fluid Load Control Option** to linearly ramp the fluid load, or to limit the values of the applied pressure and shear stress:

6. Select the **[FSI Interface]** > **Physics Conditions** > **FSI Fluid Load Control Option** node and set *Method* to **Linear Ramp & Clipping**.
7. Select the **Physics Values** > **FSI Fluid Load Linear Ramp & Clipping** node. Use the *Traction Clipping Parameters* property to specify a maximum and minimum value for pressure, and a maximum value for the shear stress magnitude, then:
   - If the fluid analysis is transient, use the *Time Ramping Parameters* property to specify the time-interval in which the fluid load value is gradually increased from zero to the actual load value.
   - If the fluid analysis is steady, use the *Start Iteration* and *End Iteration* properties to specify the iteration-interval in which the fluid load value is gradually increased from zero to the actual load value.

   For more information, see Fluid-Structure Coupling Model Reference.

For solution stabilization, STAR-CCM+ automatically estimates the volume of displaced fluid per unit area and uses it as a preconditioner to the implicit algorithm (see Fluid-Structure Coupling Model Reference). If you have an estimate of the volume of displaced fluid per unit area, you can specify its value manually:

8. To manually specify the volume of displaced fluid per unit area:
   a) Select the **[FSI Interface]** > **Physics Conditions** > **FSI Added Mass Control Option** node and set *Method* to **Manual**.
   b) Select the **Physics Values** > **FSI Added Mass Effect** node and set *Displaced Fluid Volume/Surface Area* to the desired value.

      For example, for water flow in a slender pipe, the displaced fluid volume per unit area can be estimated as half of the pipe inner diameter.

   To improve the simulation stability, you can also specify the displacement under-relaxation using the **Physics Values** > **FSI Displacement Under-Relaxation** node. For more information, see Fluid-Structure Coupling Model Reference.

In a two-way coupled FSI simulation, you assign motion models to the fluid and solid regions, to allow STAR-CCM+ to deform the fluid and solid meshes based on the computed displacements. In a one-way coupled FSI simulation, motion models are not necessary. For information on the combinations of motion models and reference frames that you can use in FSI simulations, see Motion Models and Reference Frames.

For two-way coupled problems, assign a **Solid Displacement** motion to the solid region and a **Morphing** motion to the fluid region. The Solid Displacement motion of the solid drives the Morphing motion of the fluid mesh:

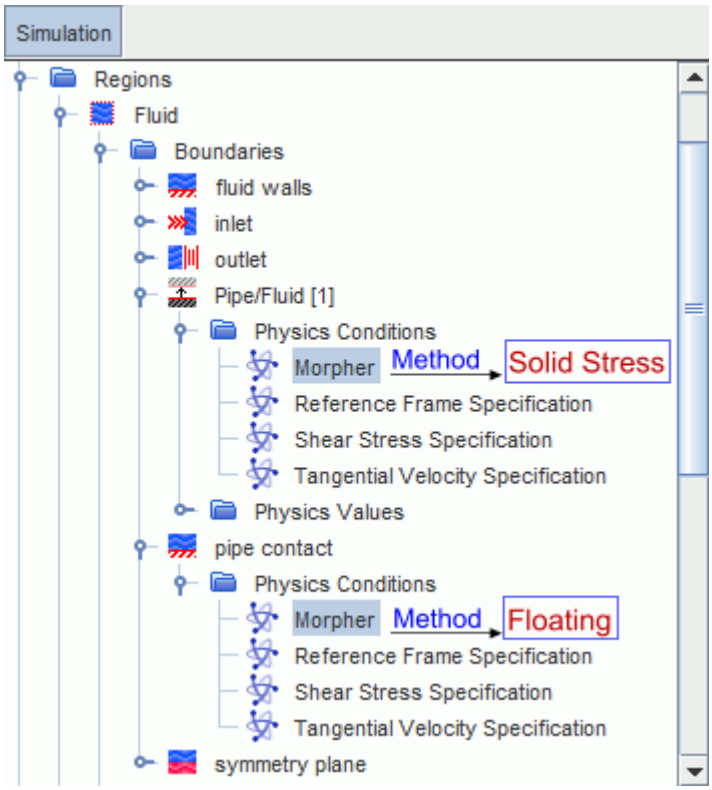
9. Right-click the **Tools** > **Motions** node and select **New** > **Solid Displacement**.
10. Expand the **Regions** > **[Solid Region]** > **Physics Values** node.
11. Select the **Motion Specification** node and set *Motion* to **Solid Displacement**.

    During the simulation, STAR-CCM+ updates the coordinates of the solid mesh vertices at each iteration, or iteration in a time-step, based on the computed displacements. Any scene displaying the solid part, or region, also updates to display the deformed geometry.
12. Right-click the **Tools** > **Motions** node and select **New** > **Morphing**.

13. Select the **[Fluid Region]** > **Physics Values** > **Motion Specification** node and set *Motion* to **Morphing**.

STAR-CCM+ automatically sets the morpher method for the interface under the **[Fluid Region]** > **Boundaries** node, to **Solid Stress**. Set the fluid boundary in contact with the solid to follow the interface motion:

14. Expand the **[Fluid Region]** > **Boundaries** node.
15. Select the **[fluid boundary in contact with the solid]** > **Physics Conditions** > **Morpher** node and set *Method* to **Floating**.



The floating method is specified so that all the faces which do not get mapped become floating.

For transient or quasi-static FSI analyses of rotating structures, it is computationally more efficient to model the solid motion in a rotating frame of reference, with the fluid motion defined in the Laboratory frame of reference. At each iteration, STAR-CCM+ transforms the fluid loads from the Laboratory frame into the solid rotating frame, and calculates the corresponding solid displacements in the rotating frame of reference. The computed displacements are then transformed back into the Laboratory frame. The total displacements are the sum of the computed solid displacements and the displacements resulting from the rigid body rotation motion.

This modeling approach accounts for centrifugal forces.

To define a rotating frame of reference:

16. Right-click the **Tools** > **Motions** node and select **New** > **Rotation**.
17. Set the properties of the **Motions** > **Rotation** node as appropriate to your case. Currently, this modeling approach only supports a constant rotation rate and constant axis of rotation.

Assign the rotation motion to the solid region:

18. Select the **[Solid Region]** > **Physics Values** > **Motion Specification** node and set *Reference Frame* to **Reference Frame for Rotation**.
19. To complete the simulation set up, follow the instructions outlined in the relevant sections of the workflow.

Solid Stress Reference

Contents:

Solid Stress Model Reference
Material Law Models Reference
Rayleigh Damping Model Reference
Fluid-Structure Coupling Model Reference
Specified Temperature Load Model Reference
Nonlinear Geometry Model Reference
Segments (Loads and Constraints) Reference

Solid Stress Model Reference

The **Solid Stress** model allows you to compute the displacement of solid structures under loads, using the finite element approach. This model is the prerequisite for all types of stress analysis.

Theory	See Solid Mechanics.	
Provided By	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **[Solid Physics Continuum]** > **Models** > **Solid Stress**	
Requires	Physics Models: *Space*: **Three Dimensional** *Material*: **Solid** or **Multi-Part Solid**	
Compatible Meshers	**Directed Mesher** **Tetrahedral Mesher** **Thin Mesher**	
Activates	Physics Models	**Material Law Models**. See Material Law Models Reference.
	Material Properties	For each solid material, STAR-CCM+ activates mechanical and thermal properties based on the specified material law. See Material Law Models Reference.
	Reference Values	**Geometry Refinement Specification**
	Initial Conditions	**Displacement** **Velocity** See Initial Conditions.
	Boundary Types	**Wall**, **Symmetry Plane**. See Boundary Types.
	Region Inputs	**Body Load Option** **Mid-side Vertex Option**

		See Region Settings.
	Interface Inputs	**Mechanical Interaction** (for solid/solid interfaces of type **Mapped Contact Interface**). See Interface Settings.
	Solvers	**Solid Stress Solver** **Sparse Direct Solver** See Solid Stress Solver Reference and Sparse Direct Solver Reference.
	Monitors	**Displacement** **Force**
	Field Functions	**Displacement**, **Element Type**, **Strain Tensor**, **Stress Max. Shear**, **Stress Mean**, **Stress Tensor**, **Stress Von Mises**, **Applied Force**, **Body Force**, **Constraint Force**, **Internal Force** See Field Functions.
	Reports	**Sum Force** **Sum Moment** See Reports.

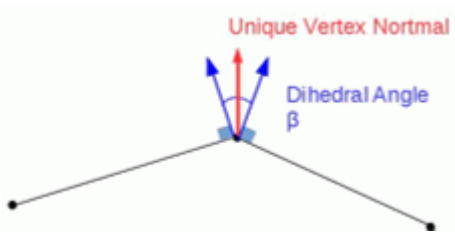
Reference Values

Geometry Refinement Specification

When you constrain a surface, STAR-CCM+ applies the constraint on the mesh vertices that lie on that surface (see Solid Stress Constraints). For accurate boundary conditions, it is important that the geometry of the surface is locally captured.

To represent the geometry of curved surfaces, STAR-CCM+ evaluates unique vertex normals at each mesh vertex by averaging the normals of the adjacent mesh faces [147]. The vertex normals provide a more accurate representation for radial (normal) directions.

On a feature edge, where two different surfaces meet, STAR-CCM+ must retain multiple normals per vertex. STAR-CCM+ recognizes a feature edge between two surfaces based on the specified *Dihedral Angle Tolerance*. If the angle between the adjacent face normals exceeds the specified tolerance, STAR-CCM+ retains multiple normals. If the angle between the adjacent face normals is within the specified tolerance, STAR-CCM+ generates a unique vertex normal.



In general, you are advised to leave the *Dihedral Angle Tolerance* as the default value.

For surfaces meeting at small angles, you can decrease the *Dihedral Angle Tolerance* so that STAR-CCM+ recognizes the feature edge shared by the two surfaces. For coarse meshes, where different faces of the same surface meet at large angles, you can increase the *Dihedral Angle Tolerance* so that STAR-CCM+ recognizes the surface as continuous and generates the unique vertex normal.

You can specify the *Dihedral Angle Tolerance* independently in each region using the **Geometry Refinement Option**. See Region Inputs.

Note: For vertices that lie on symmetry plane boundaries, the calculation of vertex normals is consistent with the full geometry resulting from the symmetry condition.

Initial Conditions

Displacement

Allows you to specify the initial displacement as a vector profile.

Velocity

Allows you to specify the initial velocity as a vector profile.

Boundary Types

Wall

By default, wall boundaries are free from loads and constraints. Wall boundaries present a **Stress/ Displacement** physics condition, which is always set to **Free**. This condition is read only, as you define loads and constraints on the associated part surfaces using segments. See Segments (Loads and Constraints) Reference.

Symmetry Plane

At symmetry plane boundaries, STAR-CCM+ automatically sets the normal component of displacement to zero. As with wall boundaries, you can define loads and constraints on the associated part surfaces using segments (see Segments (Loads and Constraints) Reference.). However, do not define constraints that specify a nonzero normal displacement, as this would result in conflicting constraints.

Region Settings

Applies to solid regions.

Body Load Option

Allows you to apply a body force on the region. Its counterpart in CFD is the Momentum Source Option. The Body Load Option is intended for body loads which are independent of the stress and strain, but may be a function of the displacement field.

Method	Corresponding Physics Value Nodes
None No body forces are applied.	None
Specified Allows you to specify body forces.	**Body Load** Set as a vector profile. **Body Load Derivative** Currently, this option is not active for FE stress analysis.

Geometry Refinement Option

For a given region, allows you to overwrite the *Dihedral Angle Tolerance* specified in the physics continuum (see Geometry Refinement Specification).

Option	Corresponding Physics Value Nodes
Use Continuum Values Sets the *Dihedral Angle Tolerance* to the value specified in the associated physics continuum. See Geometry Refinement Specification	None
Specified Sets the *Dihedral Angle Tolerance* to a specified value.	**Geometry Refinement Specification** Specifies the *Dihedral Angle Tolerance* for the region. This value overwrites the physics continuum **Geometry Refinement Specification**.

Mid-side Vertex Option

Allows you to add or remove mid-side vertices. By default, STAR-CCM+ mesh operations generate linear elements without mid-side vertices.

Although you can change this setting at any time during a simulation, STAR-CCM+ adds or removes the vertices during initialization, when you run the simulation. To verify the mesh element topology, you can visualize the Element Type field function in a scalar scene for the part. For more information, see Mesh Requirements and Guidelines.

The available methods are:

- **Add -- Linear Interpolation**—adds mid-side vertices (nodes) as linear interpolates of the vertices on either end of an edge. This option adds mid-side nodes to each edge of every element of the mesh. When you add mid-side nodes, STAR-CCM+ also interpolates any previous solution, computed on the linear mesh, to the mid-side vertices. The addition of mid-side nodes forces a new evaluation and factorization of the stiffness matrix.
- **Keep Original**—keeps the original mesh element configuration. If you generated the mesh in STAR-CCM+, and you have not added mid-side nodes, the mesh elements remain linear. If you have added mid-side nodes, the mesh elements remain quadratic. If you imported the mesh through an imported CAE model, the element topology remains linear/quadratic, based on the original configuration.
- **Remove All**—removes any mid-side nodes. This option includes mid-side nodes that were added in STAR-CCM+, or mid-side nodes that were imported with a CAE model. The removal of mid-side nodes forces a new evaluation and factorization of the stiffness matrix.

Interface Settings

Applies to solid/solid interfaces of type **Mapped Contact Interface**, with *Topology* other than **Periodic**. STAR-CCM+ automatically treats periodic interfaces, contact interfaces, and internal interfaces as **Bonded**.

Mechanical Interaction

Specifies the type of interaction between the contacting surfaces. The available options are:

- **Bonded**—the contacting surfaces undergo continuous displacements. If the solid regions are different materials, discontinuities in the strain are preserved across the material interface.
- **Small Sliding Frictionless**—allows the contacting surfaces to slide over each other in the direction tangential to the interface, whereas movement in the normal direction is prohibited. The shear stress at the interface is equal to zero, and the normal forces are equal and opposite.

  This option is intended for small sliding at flat interfaces. Sliding of vertices at curved interfaces, or sliding of vertices that exceed the mesh element length, leads to inaccurate results.

  This option targets applications where unrealistic stress concentrations would otherwise occur. For example, thermal-stress analysis involving interfaces between materials with different thermal expansion coefficients.

  This type of interaction is not compatible with the **Solid Displacement** motion.

Field Functions

Displacement

Vector field function that represents the displacement $\mathbf{u}$ in Eqn. (3789).

Element Type

Scalar field function that identifies each element topology with a unique number. To visualize the mesh element type, display this field function in a scalar scene. If all the elements in the mesh are of the same type, the color bar will display a single number. The values associated with each element topology are listed in the section, Element Type Reference.

Strain Tensor

Tensor field function that represents the strain tensor, as defined in Eqn. (3805).

Stress Max. Shear

Scalar field function that represents the maximum shear stress, as defined in Eqn. (3797).

Stress Mean

Scalar field function that represents the mean stress measure, as defined in Eqn. (3798).

Stress Tensor

Tensor field function that represents the Cauchy stress tensor (see Cauchy Stress).

Stress Von Mises

Scalar field function that represents the von Mises stress measure, as defined in Eqn. (3800).

Applied Force

Vector field function that represents the resultant of the applied surface and point loads. At a node M, it is defined as:

$$\mathbf{f}_M^{\mathrm{a}} = \int_S \mathcal{N}_M^{\mathrm{T}} \boldsymbol{\tau} \, \mathrm{d}S + \mathbf{f}_M \tag{1032}$$

where $\mathcal{N}_M$ is the nodal shape function (see Eqn. (3832)), $\boldsymbol{\tau}$ is the prescribed surface traction, and $\mathbf{f}_M$ is the point load prescribed at node M.

Body Force

Vector field function that represents the resultant of the body forces. At a node M, it is defined as:

$$\mathbf{f}_M^{\mathrm{b}} = \int_V \mathcal{N}_M^{\mathrm{T}} \mathbf{b} \, \mathrm{d}V - \mathbf{C}_{MN} \dot{\mathbf{u}}_N - \mathbf{M}_{MN} (\ddot{\mathbf{u}}_N - \mathbf{a}_N - \mathbf{g}_N) \tag{1033}$$

where $\mathbf{b}$ is the prescribed body load, $\mathbf{M}_{MN}$ and $\mathbf{C}_{MN}$ are the mass and damping matrices (see Eqn. (3841) and Eqn. (3851)), $\mathbf{a}_N$ is the prescribed acceleration and $\mathbf{g}_N$ is gravity.

Internal Force

Vector field function that represents the internal elastic forces, as defined in Eqn. (3840).

Constraint Force

Vector field function that represents the reaction forces at constrained boundaries and at interfaces.

The residual forces at a mesh node M, $\mathbf{r}_M = \mathbf{f}_M^{\mathrm{a}} + \mathbf{f}_M^{\mathrm{b}} - \mathbf{f}_M^{\mathrm{int}}$ can be expressed in terms of free and constrained degrees of freedom, $\mathbf{r}_M = \left\{\mathbf{r}_M^{\mathrm{f}}, \mathbf{r}_M^{\mathrm{c}}\right\}$. The residual forces of the free degrees of freedom must approach zero at convergence, $\mathbf{r}_M^{\mathrm{f}} = \mathbf{0}$. The negative residual forces of the constrained degrees of freedom, $-\mathbf{r}_M^c$, are called constraint forces.

For more information on how to use these field functions in scenes and reports, see Reports and Solution Analysis Guidelines.

Reports

The **Sum Force** and **Sum Moment** reports allow you to check for equilibrium of all the forces and moments acting on specified input parts, as required by conservation of linear and angular momentum. Each part and the sum of all input parts must be in equilibrium. The **Sum Force** and **Sum Moment** reports are defined with respect to a Cartesian coordinate system.

Sum Force

Reports the resultant of the applied, body, internal, or constraint forces on the specified input parts. The input parts for this type of report can either be geometry parts or regions.

STAR-CCM+ first calculates the force resultants per input part, by summing all the forces at the part nodes, and then sums the resultants calculated for each input part.

For example, to calculate the **Sum Force** of the **Applied Force**, STAR-CCM+ calculates the resultant of the **Applied Force** for each input part, as:

$$\mathbf{f}^{\mathrm{a}} = \sum_M \mathbf{f}_M^{\mathrm{a}} \tag{1034}$$

and then sums the resultants calculated for each input part.

Sum Moment

Reports the resultant moments of the applied, body, internal, and constraint forces, acting on the selected input parts, with respect to a specified location $\mathbf{x}_0$. STAR-CCM+ first calculates the resultants of the

moments per input part, by summing all the moments at the part nodes, and then sums the moment resultants calculated for each input part.

For example, to calculate the **Sum Moment** of the **Applied Force**, STAR-CCM+ calculates the resultant of the moment of the **Applied Force** for each input part, as:

$$\mathbf{m}^{\mathrm{a}} = \sum_{M} (\mathbf{x}_M - \mathbf{x}_0) \times \mathbf{f}_M^{\mathrm{a}} \tag{1035}$$

and then sums the resultants calculated for each input part.

Contents:

Solid Stress Solver Reference
Sparse Direct Solver Reference

Solid Stress Solver Reference

The **Solid Stress Solver** node allows you to access computation parameters, such as the **Sparse Direct Solver** settings, the update method for the stiffness matrix in nonlinear applications, and the **Static Analysis** option for quasi-static analysis with the Implicit Unsteady model. The **Solid Stress Solver** node also allows you to estimate the simulation memory requirements.

Solid Stress Solver Properties

Verbosity

Allows you to choose how much information STAR-CCM+ reports in the *Output* window while the simulation is running. This property is useful for debugging problems when they occur. The available options are **None** , **Low** , and **High** .

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Temporary Storage Retained

When activated, the solver retains the displacement and force residuals at the end of the iteration and makes it available as field function.

Norm Type

Specifies the method that STAR-CCM+ uses for calculating the norm of the residual force and the displacement increment. The available options are **Max Norm**, **L1 Norm**, and **Euclidean Norm** (see Residual Norms).

When the *Normalization Option* for the **Force** and **Displacement** monitors is set to **Auto**, STAR-CCM+ displays normalized values, that is, the residual force and the displacement increment norms are divided by the norm values computed at the first solver iteration.

Strain Energy

When activated, STAR-CCM+ adds a monitor for the strain energy increment, which can be visualized in the *Residuals* plot.

The strain energy increment is calculated according to Eqn. (3883). When the *Normalization Option* for the **Strain Energy** monitor is set to **Auto**, the strain energy increment is normalized with respect to the strain energy increment computed at the first solver iteration.

Solid Stress Solver Right-Click Actions

Estimate Memory

Estimates the memory required by the **Solid Stress Solver**, based on the **Sparse Direct Solver** settings. When the estimate is complete, the *Output* window reports the following information:

N—model degrees of freedom
NZ(A)—number of non-zero elements in the stiffness matrix
mem(A) (MB)—memory required to store the stiffness matrix
mem(fact) (MB)—memory required to factorize the stiffness matrix
mem (MB)—total memory required by the **Solid Stress Solver**

When you set the **Sparse Direct Solver** mode to **AUTO**, the report displays the memory requirements for both **In-Core** and **Out-of-Core** running modes. If you set the solver mode to either **In-Core** or **Out-of-Core**, the report only displays the memory requirements for the relevant running mode. For more information, see Sparse Direct Solver Reference.

If you use this option before running the simulation, STAR-CCM+ estimates the memory requirements based on the FE mesh. If you use this option again after running the simulation, STAR-CCM+ estimates the memory requirements based on the computed stiffness matrix, which also includes the model constraints.

Solid Stress Solver Controls

Integration Method

The time integration method depends on the temporal discretization scheme for the **Implicit Unsteady** solver. When the **Implicit Unsteady** solver uses 1st-order discretization, STAR-CCM+ sets the **Integration Method** to **Backward-Euler** (see Eqn. (3873)).

When the **Implicit Unsteady** solver uses 2nd-order temporal discretization, STAR-CCM+ sets the **Integration Method** to **Newmark** (see Eqn. (3874)). In this case, you specify the Newmark parameter γ (as defined in Eqn. (3874)) using the **Integration Method** > **Newmark** node. The stiffness matrix for the Newmark method is given by Eqn. (3877).

Sparse Direct Solver

For linear materials and linear geometry, the algebraic equations for the static and dynamic problems, Eqn. (3869) and Eqn. (3872), are linear and can be solved implicitly using a linear direct solver in only one iteration, or time-step. For nonlinear geometry, the equations are nonlinear and the solution requires updates of the stiffness matrix using Newton iteration methods. At each iteration step, the direct solver solves the linearized system of equations.

For linear problems, STAR-CCM+ factorizes and stores the stiffness matrix only once, in the first iteration. Thereafter, if you change the constraints type, the Implicit Unsteady solver time-step, the mesh, or the

settings for the mid-side vertex option, STAR-CCM+ detects the changes in the stiffness matrix and refactorizes it during the next iteration. For nonlinear problems, STAR-CCM+ factorizes the stiffness matrix every time the matrix is updated, based on the specified Newton update method.

For information on the properties available to this solver, see Sparse Direct Solver Reference.

Static Analysis

When using the **Implicit Unsteady** solver for dynamic analysis, you can choose to run the analysis in static mode by activating *Static Analysis*. When you activate this option, the solver neglects the inertial terms and damping in the dynamic equations, and solves the equations for the static problem. In this case, you can create profiles for loads and constraints based on the pseudo-time.

Stiffness Matrix Update

Available when the **Nonlinear Geometry** model is active.

Allows you to choose the method that STAR-CCM+ uses to update the stiffness matrix. The available options are:

- **Full Newton**:
  STAR-CCM+ updates the stiffness matrix at each iteration step.
- **Modified Newton**:
  Allows you to specify the stiffness matrix *Update Frequency*, *M*. In static simulations, STAR-CCM+ updates the stiffness matrix every *M* iterations. In quasi-static or dynamic simulations, STAR-CCM+ updates the stiffness matrix for the first *M* iterations in a time-step. For example, if you specify *M*=1, the stiffness matrix is updated upon the first iteration of each time-step. If you specify *M*=3, the stiffness matrix is updated upon the first, second and third iteration of each time-step.

Note: Every update of the stiffness matrix requires a new factorization step. The **Modified Newton** method allows for reuse of the decomposed stiffness matrix, saving overall run time, but it is less robust than the **Full Newton** method. You are advised to use the **Modified Newton** method only for moderately nonlinear problems.

Sparse Direct Solver Reference

In finite element simulations, STAR-CCM+ solves the system of linear equations using direct solvers.

For parallel execution, STAR-CCM+ employs MUMPS [150]. For serial execution, STAR-CCM+ employs either PARDISO [149] or MUMPS [150]. For more information, see Sparse Direct Solver.

Sparse Direct Solver Properties

Verbosity

Specifies how much information is printed to the *Output* window. The available options are **None**, **Low**, and **High**. To visualize an estimate of your case memory requirements in the *Output* window, choose **Low** or **High**. The memory requirements depend on the model degrees of freedom.

Mode

Specifies whether the solver uses the physical memory (In-Core) or disk memory (Out-Of-Core) to store the matrix. The available options are:

Auto—STAR-CCM+ decides whether to run the solver in **IC** (In-Core) or **OOC** (Out-Of-Core) mode, based on the memory available to the system.
IC—In-Core (IC) mode. The solver only uses the RAM memory. In terms of CPU time, this mode is faster than the **OOC** mode.
OOC—Out-of-Core (OOC) mode. The solver also uses the disk memory. This mode requires that you specify a temporary write directory for data storage. This mode is slower than **IC** but more robust, and is intended for simulations with high memory requirements.

Temp Write Directory

Allows you to specify a directory on your system disk to use for temporary storage of data, when running in **OOC** mode. It is important that you set this property for large model simulations. If you do not set this property, STAR-CCM+ uses the system/tmp directory, which is limited in size.

When running the simulation, STAR-CCM+ creates subdirectories in the specified *Temp Write Directory*, and then removes them automatically when you close the simulation. If the simulation ends unexpectedly, delete the subdirectories manually.

Stop if Not Enough Mem

If activated, it prevents you from running the simulation when the memory requirements for the simulation exceed the memory available to your system. If deactivated, the simulation is run regardless of the memory requirements and availability.

In solid stress simulations, you can use the **Estimate Memory** option for the **Solid Stress Solver** to visualize an estimate of the memory requirements in the *Output* window. See Solid Stress Solver Reference.

Sparse Direct Solver Controls

Direct Solver Option

When running the simulation in serial, this option allows you to specify which direct solver STAR-CCM+ uses for solving the sparse system of linear equations. The available options are PARDISO [149] and MUMPS [150].

When changing the solver option during a simulation run, the change takes effect at the next factorization step (see Sparse Direct Solver).

This option is not available when running the simulation in parallel. For parallel execution, STAR-CCM+ automatically uses MUMPS.

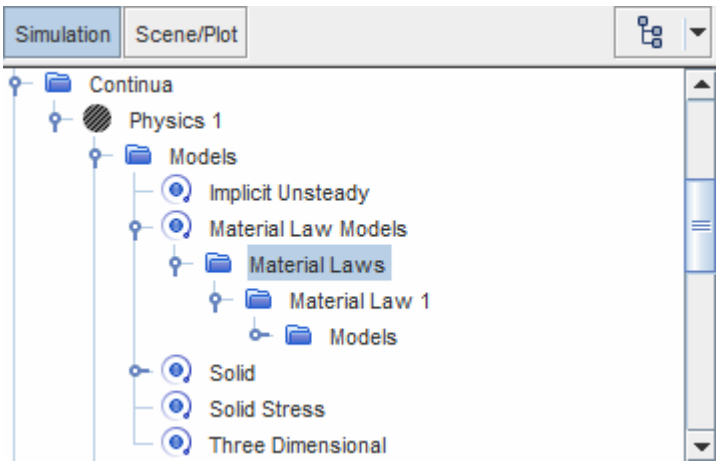
Material Law Models Reference

The **Material Law Models** define the relationship between stress and strain in the solid material. STAR-CCM+ supports linear elastic materials, with either isotropic, orthotropic, or anisotropic properties.

Theory	See Constitutive Laws.
Provided by	**[Physics Continuum]** > **Models** > *Materials Law Model*
Example Node Path	**Continua** > **[Physics Continuum]** > **Models** > **Material Law Models**

Requires	**Solid Stress** model. See Solid Stress Model Reference.	
Activates	Controls (child nodes):	**Material Laws**. See Model Controls.
	Material Properties	See Material Properties.
	Field Functions	**Material Basis Vector 1, 2, 3**. See Field Functions.

Model Controls (Child Nodes)



Material Laws

Allows you to create the material laws that define the stress-strain relationship in solid materials. When you right-click the **Material Laws** node and select **New**, STAR-CCM+ creates a material law with the default name **[Material Law 1]**.

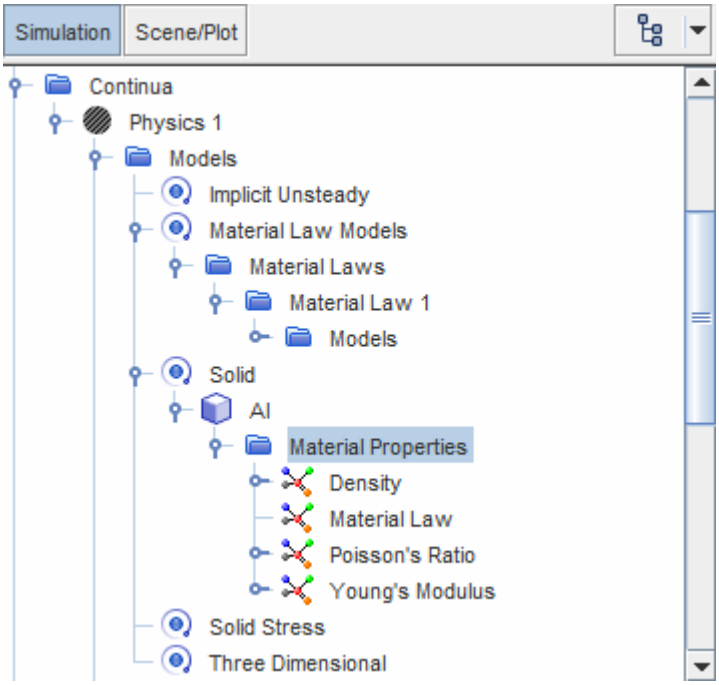
[Material Law 1] > Models

Allows you to select the physics models that define the material law. The available models are:

Group Box	Physics Models
Material Stiffness Models	**Linear Elasticity** Prescribes a linear stress-strain relationship. See Eqn. (3820).
Linear Elastic Material Models	**Isotropic Linear Elastic** Defines an isotropic material law (see Linear Elastic Isotropic Material Law). The mechanical properties of isotropic materials are the same in all directions. See Material Properties.

Group Box	Physics Models
	Orthotropic Linear Elastic Defines an orthotropic material law (see Linear Elastic Orthotropic Material Law). The mechanical properties of orthotropic materials are specified independently along three mutually-orthogonal directions. **Anisotropic Linear Elastic** Defines an anisotropic material law (see Linear Elastic Anisotropic Material Law). The mechanical properties of anisotropic materials depend on direction.
Optional Models	**Thermal Expansion** Accounts for the thermal strain due to temperature loads, by including $\boldsymbol{\varepsilon}_T$ in Eqn. (3820).
Thermal Expansion Material Models	**Isotropic Thermal Expansion** The material is isotropic with respect to thermal expansion, that is, the coefficient of thermal expansion, which relates the thermal strain and the temperature change, is the same in all directions. See Material Properties. **Orthotropic Thermal Expansion** The material is orthotropic with respect to thermal expansion, that is, the coefficient of thermal expansion is different along three mutually-orthogonal directions. **Anisotropic Thermal Expansion** The material is anisotropic with respect to thermal expansion, that is, the coefficient of thermal expansion depends on direction.

Material Properties



Material Law

Type—Allows you to select a material law from the material laws that you created under the **[physics continuum]** > **Models** > **Material Law Models** > **Material Laws** node. See Model Controls.
Coordinate System—Specifies the coordinate system with respect to which the other material properties are defined. This setting has no effect on isotropic materials, as their properties are the same in all directions. You can visualize the material coordinate system in a scene using the material basis vector field functions. See Field Functions.

Density

Defines the mass per unit volume in the undeformed state (ρ_0 in Eqn. (3811)).

Other Material Properties

When you set the **Material Law** property, STAR-CCM+ adds additional material properties based on the material law physics models.

Material Law Models	Material Properties
Isotropic Linear Elastic	**Poisson's Ratio** Defines Poisson's ratio (ν in Eqn. (3829)) for the isotropic material. **Young's Modulus** Defines the Young's modulus (E in Eqn. (3829)) for the isotropic material.

Material Law Models	Material Properties
Orthotropic Linear Elastic	**Poisson Ratios** The nodes **nu12**, **nu23**, and **nu13** define the Poisson coefficients of the orthotropic material (ν_{12}, ν_{23}, ν_{13} in Eqn. (3823) and Eqn. (3824)). **Young's Moduli** The nodes **E1**, **E2**, and **E3** define the Young modulus coefficients of the orthotropic material (E_1, E_2, E_3 in Eqn. (3823) and Eqn. (3824)). **Shear Moduli** The nodes **G1**, **G2**, and **G3** define the shear modulus coefficients of the orthotropic material (G_{12}, G_{23}, and G_{13} in Eqn. (3822) and Eqn. (3823)).
Anisotropic Linear Elastic	**Anisotropic Stiffness Coefficients** Define the independent components c_{ij} of the material tangent $\mathbf{D}$ (see Eqn. (3821)).
All thermal expansion models	**Zero Thermal Strain Reference Temperature** Defines the reference temperature at which the thermal strain is assumed to be zero (T_{ref} in Eqn. (3820)).
Isotropic Thermal Expansion	**Thermal Expansion Coefficient** Defines the thermal expansion coefficient (α in Eqn. (3831)) for the isotropic material.
Orthotropic Thermal Expansion	**Orthotropic Thermal Expansion Coefficients** Defines the components of the thermal expansion vector (α_{11}, α_{22}, α_{33} in Eqn. (3827)) for the orthotropic material.
Anisotropic Thermal Expansion	**Anisotropic Thermal Expansion Coefficients** Defines the components of the thermal expansion vector (α_{ij} in Eqn. (3827)) for the anisotropic material.

Field Functions

Material Basis Vector 1, Material Basis Vector 2, Material Basis Vector 3

Vector field functions that represent the basis vectors of the coordinate system with respect to which the material properties are defined. See Material Coordinate System.

Rayleigh Damping Model Reference

The **Rayleigh Damping** model allows you to model the structural response to dynamic excitation using approximate damping coefficients. You can also use the **Rayleigh Damping** model to include numerical damping that reduces unwanted numerical oscillations.

Where modal data is available, you can calculate the Rayleigh coefficients [148] based on the most significant modes of oscillation, using Eqn. (3861). This damping is an approximate fit to the asymptotic behavior at the high frequency prescribed by the stiffness proportional damping, and the asymptotic behavior at the low frequency prescribed by the mass proportional damping. So if you determine the damping at two different frequencies, you can calculate the damping coefficients. This fit is only an approximation of the real damping in the structure.

Theory	See Rayleigh Damping.
Provided by	**[physics continuum]** > **Models** > *Optional Models*
Example Node Path	**Continua** > **[Solid Continuum]** > **Models** > **Rayleigh Damping**
Requires	Physics models: **Implicit Unsteady** **Solid Stress** or **Finite Volume Solid Stress**
Properties	Key properties are: *Mass Damping Coefficient* and *Stiffness Damping Coefficient*. See Rayleigh Damping Model Properties.

Rayleigh Damping Model Properties

Mass Damping Coefficient

Specifies the low frequency damping coefficient f_M in Eqn. (3861), which is often used to prescribe damping due to an external fluid.

Stiffness Damping Coefficient

Specifies the high frequency damping coefficient τ_K in Eqn. (3861), which is often used to damp both shear and dilation waves.

Note: Setting both damping coefficients to zero is equivalent to running the simulation without the **Rayleigh Damping** model selected. The default values are rarely applicable. To remove high frequency transients that are introduced in dynamic problems, it is useful to apply a condition with no mass proportional damping, and a stiffness proportional damping equal to the time-step. This ensures quick damping of the transient modes with natural frequencies smaller than the time-step.

Fluid-Structure Coupling Model Reference

The **Fluid-Structure Coupling** model allows you to perform one-way and two-way coupled FSI simulations.

Provided by	**[physics continuum]** > **Models** > *Optional Models*

Example Node Path	**Continua** > **[Solid Physics Continuum]** > **Models** > **Fluid-Structure Coupling**	
Requires	Physics Models: *Optional Models*: **Solid Stress**	
Activates	Interface Inputs	The following inputs are only available for interfaces of type **Mapped Contact Interface**: **FSI Fluid Load Control Option** **FSI Added Mass Control Option** **FSI Displacement Under-Relaxation** See Interface Settings.
	Solvers	**Fluid Structure Interaction** See Fluid Structure Interaction Solver.

Interface Settings

FSI Displacement Under-Relaxation

Specifies the factor that STAR-CCM+ uses to under-relax the displacement of the structure.

FSI Added Mass Control Option

Specifies the volume of displaced fluid per unit area.

The solver uses this value as a preconditioner to the implicit algorithm, for solution stabilization. The solver adds a force correction which is proportional to the product of the displaced fluid per unit area and the change in the structure acceleration. This preconditioner is particularly important in strongly coupled FSI applications.

Method	Corresponding Physics Value Nodes
Auto STAR-CCM+ automatically estimates the volume of displaced fluid per unit area.	None
Manual Allows you to manually specify the volume of displaced fluid per unit area.	**FSI Added Mass Effect** Specifies the volume of displaced fluid per unit area, that is, the added mass.

FSI Fluid Load Control Option

Allows you to control how the fluid load is applied on the solid at the fluid-structure interface. You can ramp the fluid load over a specified time-based or iteration-based interval, or specify limit values for the applied load.

Method	Corresponding Physics Value Nodes
None No fluid load control is used.	None

Linear Ramp & Clipping

STAR-CCM+ ramps the fluid load from zero to the actual value, in a specified time or iteration interval. This option also allows you to specify limit values for pressure and shear stress.

FSI Fluid Load Linear Ramp & Clipping

Specifies the ramping interval and traction clipping parameters through the following properties:

- *Traction Clipping Parameters*: To reflect the actual operating conditions, you can limit the applied traction values, specifying the maximum and minimum pressure, as well as the peak shear stress magnitude. Enter the *Traction Clipping Parameters* as `[p_max, p_min, shear_max]`, choosing appropriate units.
- *Time Ramping Parameters* (available for transient fluid analysis): You can specify a start and end time to increase the applied fluid load progressively, from zero to the actual load. Enter the *Time Ramping Parameters* as `[t_start, t_end]`, choosing appropriate units.
- *Start Iteration* and *End Iteration* (available for steady fluid analysis): You can specify a start and end iteration to increase the applied fluid load progressively, from zero to the actual load.

Fluid Structure Interaction Solver

Allows for one-way and two-way implicit coupling of the fluid and solid physics. This solver uses the fixed-point iterative method to under-relax the solid displacement at the fluid-structure interface.

Solver Properties

Method

Allows you to choose between a **One Way** or **Two Way** coupled approach. In two way-coupled analyses, the fluid and solid regions exchange fluid forces and solid displacements across the fluid-structure interface. In one-way coupled analyses, only the fluid forces are transferred to the solid region across the fluid-structure interface. One-way coupled simulations are only suitable for applications where the effects of the structure displacements on the fluid are negligible.

Solver Controls

FSI Displacement Convergence Tolerance

Sets the tolerance to evaluate convergence within a time-step.

Rigid Body Rotation Correction

For applications that involve large displacements, you can activate this option to improve convergence.

The fluid-structure interface is generally non-conformal and the nodes on the fluid mesh can have an offset to the nodes on the solid mesh. When you activate this option, STAR-CCM+ rotates the offset into the deformed configuration to ensure that the fluid nodes adhere to the solid surface, when the displacements are large.

Motion Models and Reference Frames

FSI simulations support the following combinations of motion models and frames of reference:

Motion Models	Reference Frames	Scenario
Fluid: **Stationary** Solid: **Stationary**	Fluid: **Lab Reference Frame** Solid: **Lab Reference Frame**	STAR-CCM+ does not update the fluid mesh based on the computed solid displacements.
Fluid: **Morpher** Solid: **Solid Displacement**	Fluid: **Lab Reference Frame** Solid: either **Lab Reference Frame** or **Reference Frame for Rotation**	In two-way coupling, this combination instructs STAR-CCM+ to morph the fluid mesh based on the computed solid displacements. When modeling rotating solid regions, it is convenient to compute the solid displacements in a rotating frame of reference.
Fluid: **Rigid Body Motion** Solid: ---	Fluid: **Lab Reference Frame** Solid: either **Lab Reference Frame** or **Reference Frame for Rotation**	In one-way coupling, you can assign a motion model to the fluid region. However, since the solid displacements do not affect the flow, STAR-CCM+ ignores any motion of the solid region.

Specified Temperature Load Model Reference

The **Specified Temperature Load** model allows you to apply thermal loads on the solid region. The **Solid Stress Solver** uses the specified temperature profile to compute the structure displacements due to the applied thermal loads.

Theory	See Thermal Strain.	
Provided by	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **[Solid Physics Continuum]** > **Models** > **Specified Temperature Load**	
Requires	Physics Models: **Solid Stress**	
Properties	*Continuum Temperature* See Specified Temperature Load Model Properties.	
Activates	Material Properties	For each solid material, STAR-CCM+ activates thermal material properties based on the specified material law. See Material Law Models Reference.
	Initial Conditions	**Specified Temperature** See Initial Conditions.
	Region Inputs	**Specified Temperature Option** **Specified Temperature**

		See Region Settings.
	Field Functions	**Thermal Strain Tensor** **Reference Temperature at Zero Strain** See Field Functions.

Specified Temperature Load Model Properties

Continuum Temperature

Allows you to specify a temperature load as a scalar profile. STAR-CCM+ applies the specified load to all the regions associated with the continuum, for which the **Specified Temperature Option** has *Method* set to **Continuum**. See Region Settings.

Initial Conditions

Specified Temperature

Allows you to specify the initial temperature as a scalar profile.

Region Settings

Applies to solid regions.

Specified Temperature Option

Allows you to apply a thermal load on the solid region. The **Solid Stress Solver** computes the structure displacements based on the specified temperature profile.

Method	Corresponding Physics Value Nodes
Specified Allows you to apply a temperature load on the solid region. STAR-CCM+ ignores any temperature load that you specify using the *Continuum Temperature* property of the **[physics continuum]** > **Models** > **Specified Temperature Load** node.	**Specified Temperature** Specifies temperature as a scalar profile.
Continuum Allows you to apply a temperature load on the solid region. STAR-CCM+ uses the temperature load that you specify using the *Continuum Temperature* property of the **[physics continuum]** > **Models** > **Specified Temperature Load** node.	None

Field Functions

Thermal Strain Tensor

Represents the thermal strain tensor in Eqn. (3820).

Reference Temperature at Zero Strain

Represents the temperature at which the thermal strain is assumed to be zero. See Reference Temperature.

Note: When you map temperature data from a different mesh representation of the solid, for example, a FV solid region in a STAR-CCM+ CHT analysis, the mapping operation generates a field function with the default name, **MappedVertexTemperature**. See Applying Thermal Loads.

Nonlinear Geometry Model Reference

The **Nonlinear Geometry** model allows you to model nonlinear phenomena, such as large displacements and rotations, and loads applied on slender parts in either tension or compression.

Linear geometry phenomena assume infinitesimal strain (Eqn. (3805)) and a linear relationship between strain and displacement (Eqn. (3837)). For linear elastic materials, the resulting equilibrium equations are linear and can be satisfied in the undeformed configuration. For nonlinear geometry phenomena, the strain-displacement relationship (Eqn. (3834)) is nonlinear. Therefore, the resulting equilibrium equations are also nonlinear and must be satisfied in the deformed configuration. To solve the nonlinear equations, STAR-CCM+ updates the stiffness matrix using Newton iteration methods. At each iteration step, the direct solver solves the linearized system of equations.

Theory	See Strain Definitions and Finite Element Discretization.	
Provided by	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **[Solid Physics Continuum]** > **Models** > **Nonlinear Geometry**	
Requires	Physics Models: **Solid Stress** **Linear Isotropic Elastic** **Note:** For nonlinear geometry, the linear isotropic elastic approximation becomes inaccurate for strains grater than 20%.	
Activates	Solver Controls	**Solid Stress Solver** > **Stiffness Matrix Update** See Solid Stress Solver Controls.

Treatment of Loads in Nonlinear Geometry

Linear geometry applications assume that the loads magnitude and orientation do not change during the simulation. Nonlinear geometry applications require updates of the load orientation, as the surfaces where the loads are applied can significantly change their orientation due to the large deformations. Nonlinear geometry uses follower forces, which are loads that adjust as the structure deforms, so that their orientation relative to the structure geometry remains unchanged.

STAR-CCM+ treats the Force, Traction, and Pressure loads according to the rules described in the table below. The rules are independent of the motion model, that can be either Stationary or Solid Displacement:

- **Force**—the force vector (direction and magnitude) is preserved, the corresponding traction changes with deformation when **Nonlinear Geometry** is active.

- **Traction**—the traction vector (direction and magnitude) is preserved, the force resultant of the segment changes with deformation when **Nonlinear Geometry** is active.
- **Pressure**—the pressure magnitude is preserved. The direction of the pressure is normal to the surface and changes with the deformation when **Nonlinear Geometry** is active.

Load	Linear Geometry	Nonlinear Geometry
Force $\mathbf{F}$	$\mathbf{t}_0 = \mathbf{F}/\mathrm{A}_0$ where $\mathbf{F}$ is preserved for all combinations and A_0 is the undeformed area.	$\mathbf{t} = \mathbf{F}/\mathrm{A}$ where $\mathbf{F}$ is preserved for all combinations and A is the deformed area.
Traction $\mathbf{t}$	$\mathbf{F}_0 = \mathbf{t} \cdot \mathrm{A}_0$ where $\mathbf{t}$ is preserved for all combinations.	$\mathbf{F} = \mathbf{t} \cdot \mathrm{A}$ where $\mathbf{t}$ is non follower.
Pressure p	$\mathbf{F}_0 = \mathrm{p} \cdot \mathbf{n}_0 \cdot \mathrm{A}_0$ where p is preserved for all combinations and $\mathbf{n}_0$ is the normal to the undeformed area.	$\mathbf{F} = \mathrm{p} \cdot \mathbf{n} \cdot \mathrm{A}$ where p is follower and $\mathbf{n}$ is the normal to the deformed area.

Segments (Loads and Constraints) Reference

A segment represents either a load or a constraint on a collection of part surfaces, or part points. All segments appear under the **[Solid Region]** > **Segments** node.

Instructions for creating and setting up segments to define loads and constraints are provided in the sections, Applying Mechanical Loads and Applying Constraints.

Segments Manager Properties

Segments

Displays the number of segments that you created for a region (read-only).

Segments Manager Right-Click Actions

Create Segment

Add a segment under the **Segments** node. You can choose between two classes of segments:

- **Surface Segment**—allows you to apply a load or constraint on part surfaces.
- **Point Segment**—allows you to apply a load or constraint on part points.

For a solid region, you can create as many segments as required by your analysis.

New Group

Allows you to organize the surface segments into groups. This option adds a node, **[New Group]**, under the **Segments** node. You can move segments to groups using drag-and-drop. To move the segments back to their original location and delete a group, right-click the group node and select **Ungroup**.

Group By

Organizes the segments into subfolders, based on specified criteria:

Physics Conditions —groups the segments that share the same physics conditions into a group.
Physics Values—groups the segments that share the same physics values into a group.
Class —groups the segments by class (surface or point).
Surfaces—groups the segments based on the associated part surfaces.
Type —groups the segments by type (load or constraint).

Segment Properties

Index

Specifies the unique number that identifies a segment (read-only).

Surfaces

Available for surface segments, specifies the part surfaces where the load, or constraint, is applied.

Points

Available for point segments, specifies the part points where the load, or constraint, is applied.

Type

Specifies whether the segment represents a load or constraint condition.

Property Value	Activates
Load Allows you to apply a load on the selected *Surfaces*, or *Points*.	**Solid Stress Loads** See Solid Stress Loads.
Constraint Allows you to apply a constraint on the selected *Surfaces*, or *Points*.	**Solid Stress Constraints** See Solid Stress Constraints.

Segment Conditions and Values

You define loads and constraints by setting the physics conditions and values for the segments.

Solid Stress Load

Specifies the type of load that is applied to the part entities selected for the segment. You can specify force, pressure, and traction loads on surfaces, and nodal forces on points. When you define a surface load, STAR-CCM+ obtains the corresponding nodal loads by integration of the surface load, with appropriate nodal shape functions, over the element faces which lie on the part surfaces assigned to the segment. When you define a point load, STAR-CCM+ directly applies the nodal force on the part points assigned to the segment.

Method	Activated Values and Conditions
Force Available for surface and point segments. Specifies the load as a force field.	**Force** Set as a vector with a choice of coordinate system. • **Surface Segment**: STAR-CCM+ converts the specified force vector to a uniform surface traction. You are advised to specify the force using the **Constant** method. Use other methods with care. To convert the specified force to a surface traction, STAR-CCM+ divides the force by the total area of all the surfaces that are assigned to the segment. For **Stationary** motion of the solid region, the total area is the undeformed area, whereas for **Solid Displacement** motion, the total area is the deformed area. The calculated traction is applied uniformly to each surface regardless of the local orientation of the surface. The applied load, that is, the sum of the nodal loads as specified for the segment, is not exactly equal to the sum of the forces that are derived for example from field functions or tabular data. In particular, force is not the nodal load applied to all the nodes on the surface. • **Point Segment**: STAR-CCM+ applies the specified force at each point of the specified part points.
Pressure Available for surface segments. Specifies the load as a pressure field, that is equivalent to specify the normal component of a surface traction with tangential components equal to zero.	**Pressure** Set as a scalar field. Pressure loads are applied in the direction normal to the surface. Positive values correspond to compressive loads, whereas negative values correspond to tensile loads. **Surface Load Treatment Option** Specifies how pressure and traction fields are treated at the solid element faces. Set this option only when the fields are spatially varying across the surfaces of the segment, that is, when you are specifying the pressure/traction profiles with a *Method* other than **Constant**. The available options are: **Continuous**—treats the pressure/traction fields as continuous functions across edges and vertices of neighbor elements. **Discontinuous**—treats the pressure/traction fields as constant over the element face, with field values computed at the face centroid. For constant pressure/traction profiles, the **Continuous** and **Discontinuous** options are equivalent.
Traction Available for surface segments. Specifies the load as a traction field.	**Traction** Set as a vector with a choice of coordinate system. **Surface Load Treatment Option** See Surface Load Treatment Option

Solid Stress Constraints

Specifies the type of constraint that is applied to the part entities selected for the segment. When you define a surface constraint, STAR-CCM+ applies the constraint on all the mesh nodes which lie on the part surfaces that are assigned to the segment. When you define a point constraint, STAR-CCM+ applies the constraint on all the points in the part points that are assigned to the segment.

To prevent conflict, do not prescribe a nonzero normal displacement on part surfaces that are associated with symmetry boundaries. See Symmetry Plane.

These options, applicable to solid elements, do not specify rotations.

Method	Corresponding Physics Value Nodes
Fixed Available for surface and point segments. Sets the components of the displacement field to zero, preventing the corresponding part surfaces, or part points, from moving in any direction.	None
Displacement Available for surface and point segments. Allows you to specify the components of the displacement field. If the values of all components are set to zero, this method is equivalent to the **Fixed** method.	**Displacement** Set as a vector with a choice of coordinate system. You can use the **Composite** method to select the components that you wish to constrain (use the properties of the **Displacement** > **Composite** node). All other methods constrain all the components.
Normal Displacement Available for surface segments. Allows you to specify the displacement field component in the direction normal to the selected part surfaces. Setting the normal displacement to zero prevents the part surfaces from moving in the normal direction. The normal displacement constraint is intended for plane surfaces or cylindrical surfaces, and it is not recommended on arbitrarily curved surfaces. The normal displacement option with single precision geometry may introduce tiny bumps in a plane surface, causing artificial stress concentrations. In general, you are advised to use the **Displacement** option with the **Composite** method, and fix the normal component of the displacement. If the surface is not aligned with the laboratory coordinate system, define an appropriate local coordinate system.	**Normal Displacement** Set as a scalar field. The displacement is applied in the direction normal to the surface. Positive values produce a tensile reaction, whereas negative values produce a compressive reaction.

Segment icons change based on the segment class, type, and settings, as follows:

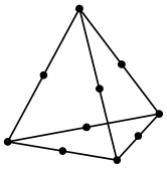
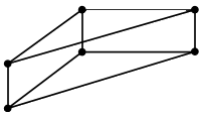
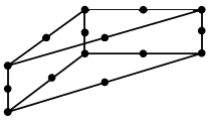
	Point Load (Force)
	Point Constraint (Fixed)
	Point Constraint (Displacement)
	Surface Load (Force)
	Surface Load (Traction)
	Surface Load (Pressure)
	Surface Constraint (Fixed)
	Surface Constraint (Displacement)
	Surface Constraint (Normal Displacement)

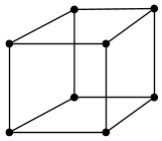
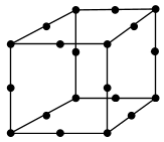
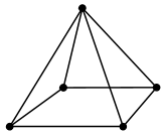
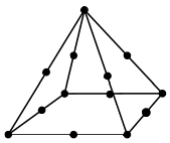
Mesh Requirements and Guidelines

Finite element stress analyses in STAR-CCM+ require a mesh of 3D solid elements generated using parts-based mesh operations, or imported as a CAE model. STAR-CCM+ supports hexahedral, tetrahedral, wedge, and pyramid element types. Each type is available without mid-side nodes (linear element), or with mid-side nodes (quadratic element).

Element Type Reference

In STAR-CCM+, mesh operations generate linear elements by default. You can achieve quadratic elements by using the **Mid-side Vertex Option**, which adds mid-side nodes as interpolates of the corner nodes that are generated by the mesh operations. The **Mid-side Vertex Option** is available as a physics condition for the solid region. For more information, see Region Settings.

Element Topology		Element name	Diagram	Nodes	Faces	Edges	Element Type FF Identifier
Tetrahedron	Linear	**Tet4**		4	4	6	10
	Quadratic	**Tet10**		10	4	6	24
Wedge	Linear	**Wedge6**		6	5	9	13
	Quadratic	**Wedge15**		15	5	9	26
Hexahedron	Linear	**Hex8**		8	6	12	12

Element Topology		Element name	Diagram	Nodes	Faces	Edges	Element Type FF Identifier
							
	Quadratic	**Hex20**		20	6	12	25
Pyramid	Linear	**Pyramid5**		5	5	8	14
	Quadratic	**Pyramid13**		13	5	8	27

In general, quadratic elements are more accurate than linear elements, but increase the simulation degrees of freedom, and therefore, the computation cost. Linear elements can give a reasonably accurate solution in structural analyses where stresses are not varying rapidly across the region. However, linear elements of tetrahedral (Tet4), wedge (Wedge6), and pyramid (Pyramid5) topologies can cause locking, that is, the computed displacements can be orders of magnitude smaller than expected. These elements are particularly "stiff" in problems with thin members under bending loads, where the element aspect ratios can be large.

Linear hexahedral elements (Hex8) are enriched with internal degrees of freedom, which overcome locking, and are more accurate than the other linear element types. Using Hex8 elements, which have edges aligned to the part geometric features, has the following advantages:

1. The stiffness matrix is more compact. Therefore, STAR-CCM+ requires less memory to store and factorize the matrix.
2. The level of accuracy is high even for cases with thin members under bending loads, where the element aspect ratio is large.
3. These elements require fewer integration points, and hence fewer operations, to evaluate the stiffness matrix and the residuals.

Not all geometries can be easily meshed with hexahedra. When a geometry requires elements of other types, or a mix of elements with different topologies, it is important to add mid-side nodes to use quadratic order

elements. In general, pyramids are not accurate and are acceptable only as transition elements, far from areas where large stresses are expected.

Steps for generating element topologies appropriate to finite element stress analysis are provided within the section, General Workflow for Stress Analysis.

Mesh Refinement

Finer meshes are required in areas with rapid changes in stress. Typically, areas that require local mesh refinement to capture stress concentrations are close to:

- Fillets and curved edges
- Re-entrant corners
- Holes
- Cracks
- Welded surfaces which intersect free surfaces

For example, a fillet between two parts which are joined at 90° requires at least 4 elements to span the quarter circle of the radius. A hole should have at least 16 elements to span the circumference. Near the root of a crack, the elements must be highly refined in a region about 2 crack lengths near the surface, with most of the refinement near the root and coarsening as the distance from the crack root increases. When using an elastic model with a crack, the matrix is *singular*, that is, the stresses are infinite at the tip of the crack, regardless of the mesh resolution. Here the principles of Fracture Mechanics must be applied to assess the severity of the loads on the crack. These principles determine whether the loads on the crack tend to cause catastrophic failure or slowly grow the crack under cyclic loads.

Mesh Quality Metrics

The elements should be of reasonable quality so that they have little warpage (nodes on faces not in the same plane), and have isoparametric mappings which are invertible (the counterpart of positive volume cells in FVA). Currently, there are no quality metrics which are specific to FE solid mechanics, but the **Cell Quality** and **Warpage** metrics are reasonable predictors of element quality. Poor element quality can lead to a matrix which is ill-conditioned. The direct solvers are designed to accept some ill-conditioning but not without an overhead cost in time and memory.

Replacing the Mesh

When replacing an existing mesh with a new mesh, STAR-CCM+ automatically interpolates any vertex data (such as Displacement), available on the existing mesh, to the new mesh vertices. For more information on the **Replace Mesh** operation, see Replacing A Mesh Region.

Thin Parts Under Bending Loads

For thin parts under bending, Hex8 elements are preferred. Hexahedral elements with mid-side nodes (Hex20) have better accuracy but at a cost of more memory and operations. Wedge elements with mid-side nodes (Wedge15) are also suitable when the geometry is sufficiently regular, so that a triangular surface mesh can be extruded through the thickness of the part to create wedges. For such geometries, this operation is done automatically when using the Thin Mesher with the Tetrahedral Mesher. However, in transitions from thick sections to thin sections these meshers can generate some pyramid elements, which are not suitable in areas with large stresses.

Thin parts under bending should have at least two aligned elements through the thickness. The number of elements through the thickness primarily affects the level of accuracy of the computed axial stress. Beam theory shows that the axial stress linearly varies from compressive to tensile along the neutral axis. Therefore, thin parts in bending that have the neutral axis in the center have high stress gradients. In these cases, higher order elements are more appropriate. However, the displacements due to bending loads are reasonably approximated with aligned Hex8, Hex20, or Wedge15 elements even with only one element through the thickness. This guideline assumes that the mesh resolution is sufficient in the plane of the thin part. Tet4 and Wedge6 elements are particularly poor in these situations, because they tend to be excessively stiff even when the aspect ratio of the elements is near unity. For Tet4 elements, it is particularly important to add mid-side nodes.

Solution Analysis Guidelines

STAR-CCM+ offers various field functions and tools that you can use to visualize the solution and monitor the relevant quantities. For example, you can visualize the deformed geometry and the stress and strain distribution throughout the solid structure.

For scalar displayers, **Smooth Filled** contours display vertex values (a single value per vertex). **Filled** contours display face values at surfaces (a single value per face) and element values inside an element (one value per element). In Solid Stress applications, the **Automatic** Contour Style is equivalent to **Smooth Filled**.

Displacement

STAR-CCM+ computes the displacement field at element vertices. Displacements are continuous across elements.

You can visualize the displacement field in two ways:

1. In a scalar scene, you can visualize the displacement components i, j, k, or magnitude at any point in the structure.
2. In a vector scene, you can visualize the displacement vectors at the element vertices.

The **Automatic** and **Smooth Filled** contours display the computed vertex values.

You can use a vector warp to visualize the deformed and undeformed shape, together in a scalar scene.

For infinitesimal strain, the displacements are very small, but can be scaled for visualization purposes using a warp surface. In a scalar scene, you can set up a geometry displayer to visualize the undeformed geometry, and use the scalar displayer to visualize the displacement on a vector warp derived part. For the vector warp, set *Vector Field* to **Displacement**. The vector warp auto-scales the displacement field, so that the visualized displacement magnitude is about 15% of the largest diagonal of the model.

Stresses and Strains

STAR-CCM+ calculates stress and strain tensors on demand when you create a scene or report that uses stress or strain field functions.

For each element, STAR-CCM+ extrapolates stresses and strains from Gauss integration points (see Eqn. (3995)) to vertices, faces, and the element centroid. Unlike displacements, stresses and strains are not continuous across elements. For display, the **Automatic** and **Smooth Filled** contours display average vertex values. The **Filled** contour displays face and element centroid values, showing the discontinuities across elements.

In general, for a tensor field function, you can access the following quantities:

Scalars	**Tensor components** in the Laboratory or any local coordinate system. **Note:** The shear components of the strain tensor are not equal to the engineering strain (see Eqn. (3821)).
	Tensor eigenvalues in signed order, $\lambda_0 \geq \lambda_1 \geq \lambda_2$.

	The eigenvalues of the stress tensor are the maximum, intermediate, and minimum principal stresses (see Principal Stresses). The corresponding eigenvectors are also available as field functions.
	Tensor norms: infinity, 1, 2, and Frobenius.
	Tensor invariants: $\mathrm{I}_0 = \mathrm{tr}(\mathrm{A})$ $\mathrm{I}_1 = \left(\mathrm{tr}(\mathrm{A})^2 - \mathrm{tr}\left(\mathrm{A}^2\right)\right)/2$ $\mathrm{I}_2 = \det(\mathrm{A})$
Vectors	**Tensor eigenvectors** 0,1, and 2, corresponding to the eigenvalues λ_0, λ_1, and λ_2, respectively. STAR-CCM+ scales the magnitude of an eigenvector by the value of the corresponding eigenvalue.

The following stress measures are available as separate scalar field functions:

1. Stress Max Shear
2. Stress Mean
3. Stress Von Mises

For more information on the available field functions, refer to the physics model reference sections, under Solid Stress Reference.

Forces and Moments

You can analyze the forces acting on specified input parts, by displaying the corresponding vector field functions in a scene:

Applied Force—resultant of all surface loads and point loads.
Body Force—resultant of all body forces, including body loads, gravity, centrifugal forces, inertia forces, and damping forces.
Internal Force—resultant of all internal elastic forces. When the solution is converged, its value is zero at internal vertices and vertices at free boundaries. The **Internal Force** value is equal and opposite to the **Applied Force** acting on loaded surfaces, and equal and opposite to the **Constraint Force** acting on interfaces and constrained surfaces.
Constraint Force—resultant of all reaction forces.

All of these field functions are available at the mesh vertices. The **Applied Force** is also available on surfaces; the **Body Force** is also available at the element centroids. To visualize vertex data, set the contour style of the relevant scalar or vector displayer to either **Automatic** or **Smooth Filled**. To visualize data at face or element centroids, set the contour style to **Filled**. For more details, see Field Functions.

You can also use the **Sum Force** and **Sum Moment** reports to check for equilibrium of all the forces and moments acting on specified input parts. The **Sum Force** and **Sum Moment** reports are defined in Cartesian coordinate systems only. As these reports use the force field functions, which are defined at the mesh vertices, activate the **Smooth Values** option. The For more details, see Reports.

Local Material Coordinate Systems

In solid stress applications, you define the material properties of anisotropic and orthotropic materials with respect to a coordinate system (see Material Properties). You can define the coordinate system independently for each material. STAR-CCM+ provides material basis vector field functions that allow you to visualize the axes of a material local coordinate system:

Material Basis Vector 1
Material Basis Vector 2
Material Basis Vector 3

Bibliography

[147] *Nelson Max, Weights for Computing Vertex Normals from Facet Normals, Journal of Graphics Tools, vol. 4, Issue 2, 1999*

[148] *Chowdhury and S. P. Dasgupta, Computation of Rayleigh damping coefficients for large systems, Electronic Journal of Geotechnical Engineering, vol. 8, 2003.*

[149] *Intel Math Kernel Library, Reference Manual, Document Number: 630813-060US MKL 11.1.*

[150] *MUltifrontal Massively Parallel Solver (MUMPS 5.0.2) Users' Guide, July 15, 2016.*

Modeling Finite Volume Solid Stress

This section describes how to use the **Finite Volume Solid Stress** model for solid stress analyses.

Unlike the fluid equations, the solid stress is solved in a Lagrangian framework. That is, the mesh follows the material points in the material. This means that the mass in an particular cell will remain unchanged as it deforms because it physically represents the same material. This cell mass is determined as the product of the original density of the material and the original volume. Therefore the density in the deformed state will be the original mass divided by the current volume of the cell. In the “small displacement” FV stress solver, STAR-CCM+ does not actually move the solid mesh because it is assumed that the deformations are so small that these changes in the mass would be negligible. However, with the “large displacement” capability, STAR-CCM+ continually adjusts the solid vertices of the mesh according to the displacements. For more information, see Modeling Large Displacements.

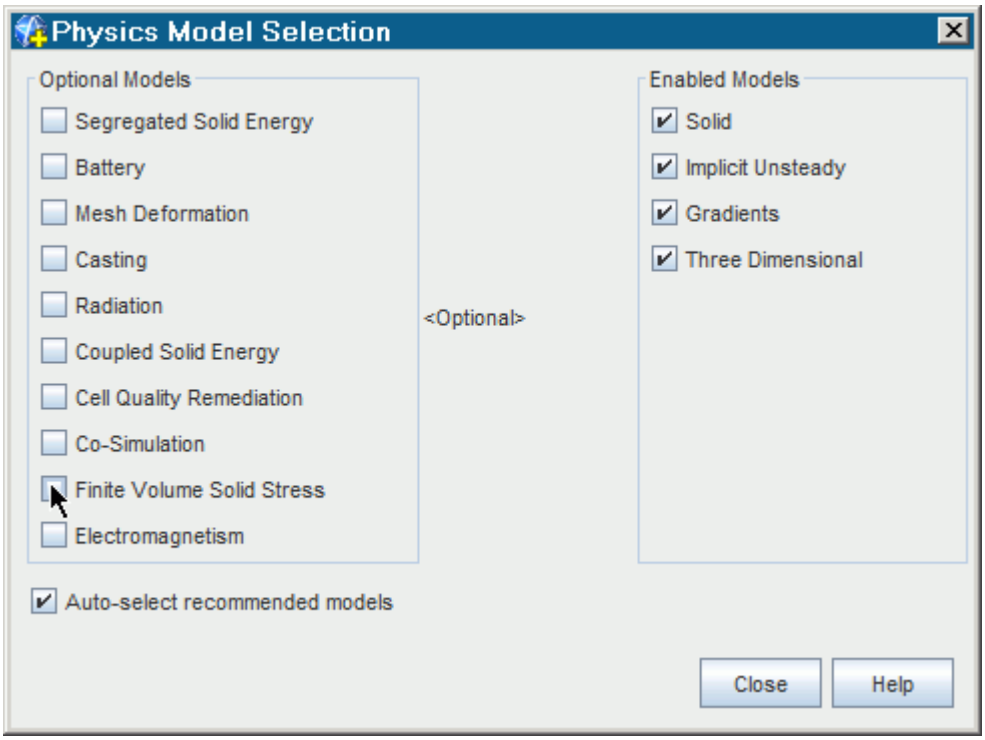
Contents:

Selecting the Finite Volume Solid Stress Model
Boundary and Initial Condition Values
Applying a Body Force
Modeling Dynamic Problems
Modeling Rayleigh Damping
Modeling Large Displacements
Working with Interfaces in Solid Stress
Modeling FSI Using Finite Volume Stress
Using the Summary Report for Bad Cells
Setting Parameters of Other Solvers
Finite Volume Solid Stress Model Reference
What Is the FVA Stress Solver?
Finite Volume Stress Analysis Guidelines
Solid Stress Formulation

Selecting the Finite Volume Solid Stress Model

The Finite Volume Solid Stress model can be used in a simulation either with solid regions only or in conjunction with fluid regions. It can be run in either implicit steady (static) mode, or implicit unsteady (dynamic) mode. The static mode neglects the inertial terms (acceleration), while still allowing for the boundary conditions, including the solid/fluid interface conditions, to be functions or time.

You can select the **Finite Volume Solid Stress** model from the *Optional Models* group box after you select the **Solid** model from the *Material* group box.



Once you select the **Finite Volume Solid Stress** model, the **Linear Isotropic Elastic** material model is selected automatically, provided the automatic selection of recommended models is activated.

Selecting the **Cell Quality Remediation** model is highly recommended for most all solid stress analysis. The solid stress convergence is particularly sensitive to the quality of the grid and the cell quality remediation is ideal for detecting and correcting for poor cells in the model. Generally, the effects of the remediation are only local to the poor cells – as can be expected from Saint-Venant's principle. For more information, see Remedying Cell Quality.

See Also:

FV Stress Solver Reference

Finite Volume Solid Stress Model Reference

Applying Boundary and Initial Conditions

This section provides an introduction into the types of constraints and loads offered by the STAR-CCM+ Finite Volume Solid Stress framework.

Boundary condition methods (values) for the **Finite Volume Solid Stress** model come in two types: constraints and surface loads. The boundary values are elected by the boundary type, and/or by methods selected in corresponding nodes in the **Conditions** node. For a list of all boundary conditions and values available when using the Finite Volume Solid Stress model, see Boundary Conditions and Values.

Constraints

STAR-CCM+ offers three types of constraint boundary conditions that are applied to a boundary surface:

- Fixed
- Prescribed Displacement
- Prescribed Normal Displacement

These conditions may be applied to a boundary surface. The boundary surface may be a collection of cell faces or a little as one cell face.

Constraints are boundary conditions on a part which impose some constraint on its deformation. The deformation is prescribed as some value. For example one can impose a fixed constraint, which requires the displacement vector to be zero on a given boundary. In FEA a constraint is applied to set of nodes on a boundary but in FVA the constraint is applied to a set of boundary cell face centers. The smallest discrete constraint set is a single node in FEA while in FVA the smallest discrete constraint set is a single boundary cell face center.

Constraining a part properly is extremely important in FEA and FVA for identical reasons. If conducting a static analysis, the body must be constrained so that it is not allowed to freely rotate or translate as a whole. The constraints must remove the so-called “rigid body modes” or a linear solution is not possible. The stiffness matrix will be singular and the solvers will not converge. Not constraining a part to remove these rigid body modes is a common “rookie” mistake. If the solution appears to diverge beginning with the first iteration, carefully examine the constraint boundary conditions.

The predicted stress field near the constrained surface should be viewed with some suspicion. For example a fixed constraint is like welding a part to another part that is infinitely rigid and cannot thermally expand. High stresses can develop at such an interface as the parts are loaded mechanically or thermally. Other than welding or braising, most parts are constrained to move by friction. Examples include the contact friction in a bolted parts between washers and bolt heads, the contact between male and female threads, the contact surfaces local to the teeth of a clamp and the contact produced from one part lying on top of another part under the weight. The constraint boundary conditions are only approximations of such loadings. Luckily, Saint-Venant’s principle applies. The stresses computed at a distance large relative to the size of the applied load will be insensitive to the details of the how the load is actually applied. For thermal stresses, the constraints must be carefully chosen so as not to over-constrain the part. Opposing constraints can confine a part producing unrealistically high thermal stresses.

Loads

Loads are surface or body forces on the solid material. The thermal stresses are commonly called loads. The surface loads are prescribed in STAR-CCM+ as boundary and/or interface conditions. There are five types of boundary surface loads:

- Free
- Prescribed Tractions
- Prescribed Pressure
- Prescribed Forces
- Elastic Foundation

Prescribed tractions, pressures and forces can also be applied to interfaces between solid and fluid domains. For more information, see Applying an Interface Load.

Applying a Combination of Constraints and Loads

The **Mixed** boundary condition makes it possible to apply a combination of constraints and loads to a boundary. The constraints and loads are applied as prescribed displacements and prescribed tractions, respectively. These are mixed by applying either a constraint or a load in each of the global directions, and the configuration is set by a Mixed Vector. This is used to differentiate which components will be read from the displacement vector, and which from the traction vector.

Setting Initial Conditions

For dynamic simulations, appropriate initial conditions must be specified for displacement as well as velocity. The default initial values for both displacement and velocity are set to null vectors ([0,0,0]). These values can be changed through the initial condition node. Field functions may also be used to prescribe initial displacement and velocity profiles.

Applying a Body Force

Enable body forces on a region.

Body forces are forces that act throughout the volume of a body, such as weight due to gravity. They can be applied to solid regions in order to model forces acting throughout the region, and are specified in STAR-CCM+ in the units of N/m3.

Body forces can be enabled on a region by selecting the **Regions** > **Solid Region** > **Physics Conditions** > **Body Load Option** node and setting the *Method* to **Specified**.

This adds two nodes to the **Physics Values** for the region:

- **Body Load**

  The Body Load option is used to specify body forces acting on the solid region. It is entered as a vector profile. To specify several body forces acting on a region, you can enter the value of the resultant body force.

- **Body Load Derivative**

  The Body Load Derivative can be used to stabilize the solution.

See Region Conditions and Values.

Modeling Dynamic Problems

Dynamics problems require the addition of the acceleration (inertial) terms into the equations of motion, Eqn. (1059). STAR-CCM+ offers two different implicit time integration methods, which are the 1st order Backward Euler method and the 2nd order Newmark method common to FEA. The Backward Euler method introduces more numerical damping than the Newmark method for a given step size, whereas the Newmark method offers no numerical damping. As a result, for time dependent quasi-static models the Backward Euler is recommended since it will quickly damp any initial transients, while the Newmark method is recommended for dynamic problems due to its accuracy and absence of damping.

In transient problems the effective stiffness matrix is modified by the inertial terms, for example for the Backward Euler method the effective stiffness matrix becomes:

$$\mathbf{K}' = \mathbf{K} + \frac{\mathbf{M}}{\Delta t^2} \tag{1036}$$

In this case, even if the structure is completely free, the effective matrix is not singular, since the mass matrix $\mathbf{M}$ is always symmetric positive definite. In the finite volume method the mass matrix is always a diagonal matrix. When at least one of the six rigid body degrees of freedom (DOF) are unconstrained, the stiffness matrix is singular. The addition of the mass matrix term can lead to better conditioning of the effective matrix provided the step size is sufficiently small. In this case, “sufficiently small” means the time step is equal to or smaller than the period of the fundamental mode of vibration of the structure. In fluid structure interaction (FSI) problems, the step size to provide a non-singular effective stiffness matrix may be smaller than that which would be needed to resolve transients in the fluid when the fundamental natural frequency of the structure is large compared to the fluid frequencies (such as vortex shedding frequencies or buoyancy induced frequencies in free surface flows).

Modeling Rayleigh Damping

The Rayleigh Damping model can be used to mimic the actual structural response to dynamic excitation, by using approximate damping coefficients to model the viscous damping effects in the structure.

The damping coefficients can be estimated based on the dominant modes in the structure, as established in a modal analysis of the structure. The Rayleigh Damping model can also be used to include some numerical damping to damp unwanted numerical oscillations.

The effects of mass and stiffness on the viscous damping are approximated in the Rayleigh Damping model via two damping coefficients, where one damps the low frequency and the other damps higher frequency. The Rayleigh Damping model assumes that the viscous damping matrix for a system is proportional to the mass and stiffness matrices.

The Rayleigh Damping model can be selected from the *Optional Models* group box in the *Physics Model Selection* dialog, after you activate the **Finite Volume Solid Stress** model.

The methodology for Rayleigh damping is provided in the formulation section.

Contents:

Setting Rayleigh Damping Parameters
Rayleigh Damping Bibliography

See Also:

Solid Stress Formulation
Rayleigh Damping Methodology

Setting Rayleigh Damping Parameters

The Rayleigh damping parameters are used to damp low frequency and high frequency oscillations in the solid structure.

Since the finite volume method does not compute the stiffness matrix, an equivalent law is used. For more information, see Rayleigh Damping Methodology.

The Rayleigh damping parameters are properties of the **Continua** > **[Solid Continuum]** > **Models** > **Rayleigh Damping** node:

- Low frequency damping is specified using the *Mass Damping Coefficient*, which is $1/\tau_M$ in Eqn. (1067). When this value is set to zero, that is $\tau_M \rightarrow \infty$, there will be no low frequency damping. This coefficient can be used to damp both shear and dilation waves.
- High frequency damping is specified using the *Stiffness damping coefficient*, which is τ_K in Eqn. (1067). When this value is set to zero there will be no high frequency damping. It can be used to remove high frequency oscillations induced by pressure waves in the fluid.

Setting both of these coefficients to zero will cause no damping to be included in the model, and is equivalent to running the simulation without the Rayleigh Damping model selected.

Selecting appropriate values for the Rayleigh damping parameters depends on the availability of modal data for the structure. Where modal data is available, the Rayleigh parameters can be calculated based on the most significant modes of oscillation, as described by Chowdhury and Dasgupta [151].

Alternatively, you can estimate the Rayleigh damping parameters using the following equation:

$$\zeta = \frac{1}{2\tau_m \omega} + \frac{\tau_k \omega}{2} \tag{1037}$$

where ζ is the damping coefficient, $1/\tau_M$ is the mass damping coefficient, τ_K is the stiffness damping coefficient and ω is the angular frequency.

Rayleigh Damping Bibliography

[151] *Chowdhury, I. and Dasgupta, S. P. 2003. "Computation of Rayleigh Damping Coefficients for Large Systems", The Electronic Journal of Geotechnical Engineering, 8, 8C.*

Modeling Large Displacements

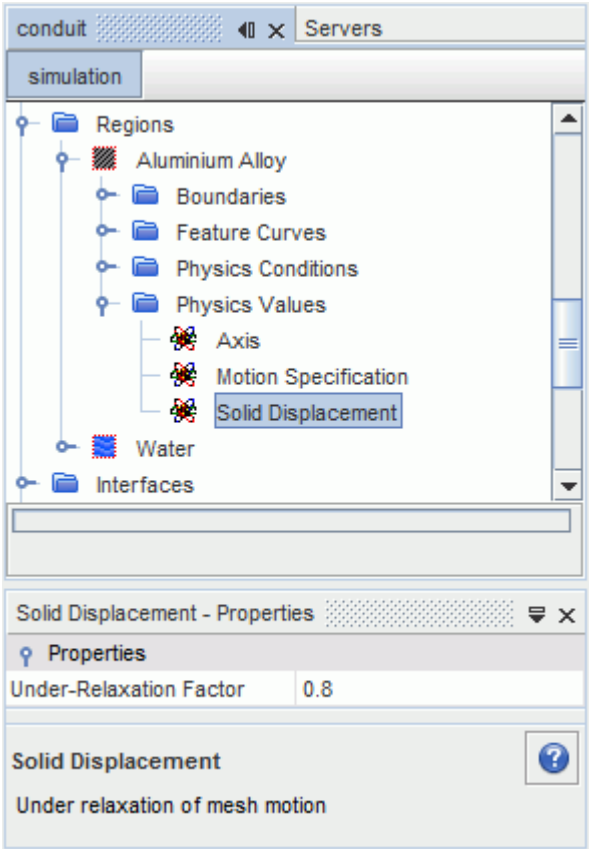
The Large Displacement model, which is enabled via the Solid Displacement motion, allows a solid mesh to be geometrically updated based on the solid stress displacement.

This model can be used to allow for the large displacements associated with rigid body motions, such as in 6 DOF cases, and is particularly useful for FSI simulations, where it is important that the fluid mesh conforms to the large motions of the solid structure. In FEA, this capability is often referred to as “Non-Linear Geometry”.

It is well known that structures can stiffen in tension, for example the strings on a violin become stiffer when they are tightened. This is often known as “Geometric stiffening”. Similarly, compression tends to decrease the stiffness, which can lead to buckling, for example. When using the Large Displacement model, the formulation requires strains that may give rise to geometric stiffening to be small (less than 10%), however, displacements in the solid can be large. In cases where the loading has a significant effect on the apparent stiffness of the structure, the Large Displacement FV solid stress may have difficulty reaching convergence and the numerical solution may show sudden jumps in the computed stresses. In these cases a smaller solid displacement under-relaxation factor should be used. If it is necessary to set the under-relaxation value to below 0.01 for a severe case, the results should be viewed very cautiously. A similar approach should be adopted for structures that are buckling due to a compressive load.

The FV large displacement formulation is completely analogous to the FEA formulation termed “Updated Lagrangian”, in that the geometry (vertex coordinates) is updated after each iteration in a time step for a transient analysis, and after each iteration in a steady-state static analysis.

The large displacement model is enabled by specifying a **Solid Displacement** motion on the solid region. See What Is a Solid Displacement Motion. When the **Solid Displacement** motion is selected for the region, a **Solid Displacement** node is added to the region values.



Properties

Under-Relaxation Factor	At each iteration, governs the extent to which the solid mesh is geometrically updated based on the computed solid stress displacement and gradients. The default value is `0.8`.

See Also:

What Is a Solid Displacement Motion?

Working with Interfaces in Solid Stress

It is possible to create interfaces between a solid and a fluid region, or between two solid regions. The procedures to create these interfaces are identical to that of the Fluid/Fluid interface. It is assumed that two regions share a boundary surface.

One way in which interfaces can be used in solid stress is to model fluid structure interaction (FSI). This involves creating an interface between the fluid and solid domains across which STAR-CCM+ will pass field data, such as pressures, heat fluxes, temperatures and displacements. Coupling between the fluid and solid domains is carried out automatically by STAR-CCM+ when a fluid-solid interface has been set up, and the finite volume solid stress model has been enabled for the solid region.

Depending on the physics models you have selected, one or both of the following coupling controls will be available under the FSI interface node:

- **FSI Coupling Method**: controls whether data will be exchanged across the FSI interface after each time-step, or after each inner iteration.
- **FSI Under-Relaxation**: specifies the extent to which the mesh will be morphed after each time-step for an FSI case using large displacements.

At an interface between the fluid and solid domains it is also possible to apply an additional load that acts on the solid domain.

Contents:

Applying an Interface Load

See Also:

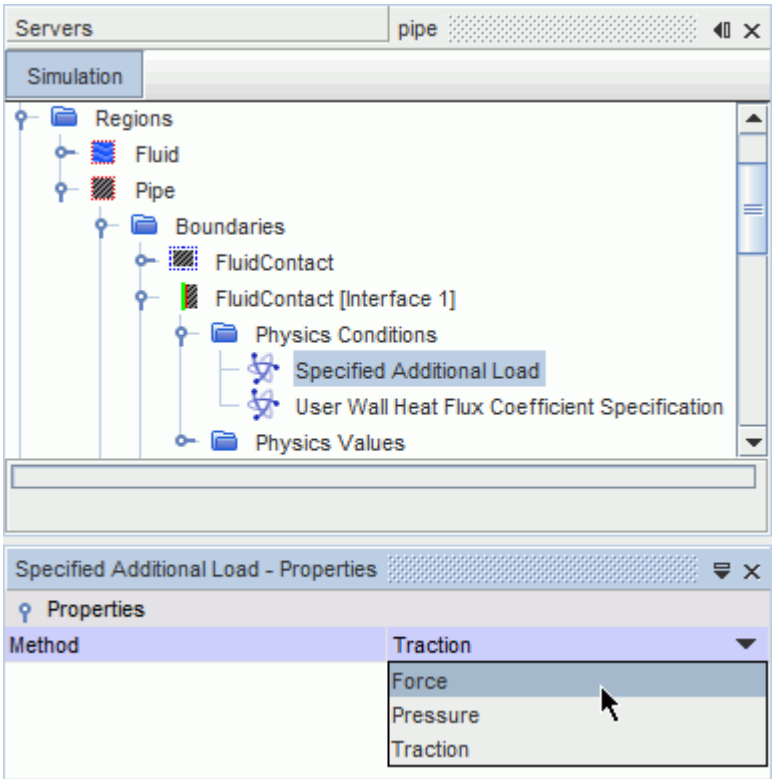
Creating a Boundary Interface

Applying an Interface Load

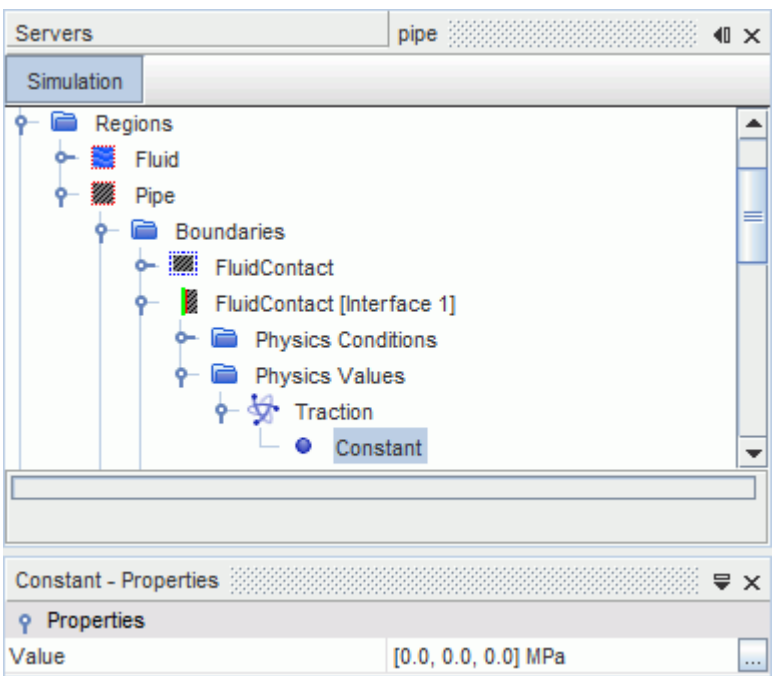
An additional load can be applied at a fluid-solid interface that will act on the solid domain.

It can be used to represent an additional load on the solid structure that may not have been modeled explicitly. For example, for a boat on water where all surfaces of the boat are interfaces with either air or water, an additional load may be applied to a surface at the rear of the boat to model the force from a motor, even though the motor itself is not present in the model. The load can be a traction, pressure or force and is specified for the interface boundary under the relevant **[Region]** node.

You select which type of load to apply at the interface by setting the *Method* for the **[Interface]** > **Physics Conditions** > **Specified Additional Load** node. The available options are **Traction**, **Force**, and **Pressure**.



After you set the type of load, specify the value of the interface load under the **[Interface]** > **Physics Values** > **[Load]** node, where **[Load]** can be either **Traction**, **Force**, or **Pressure**. Values for these nodes are specified in the same way as for prescribed loads on a solid boundary. For more information, see Finite Volume Solid Stress Model Reference. Note that you can specify load on direct or indirect interfaces.



Modeling FSI Using Finite Volume Stress

The **Finite Volume Solid Stress** model (or “FV stress”) makes it possible to simulate fluid-structure interaction (FSI) entirely within in STAR-CCM+.

Depending on the case, large or small displacements of the solid domain can be involved, and guidelines for setting up each case are covered in the following topics:

Contents:

Using Large Displacements for FSI
Using Small Displacements for FSI

Using Large Displacements for FSI

The **Large Displacement** model can be used in a fluid-structure interaction (FSI) case to simulate movement in the solid mesh.

Guidelines describing the settings that are required for preparing such a case using implicit coupling and the large displacement model are described in the following pages:

Contents:

Selecting Physics Models
Setting the Fluid Morpher Boundary Condition
Preparing the Fluid-Structure Interface

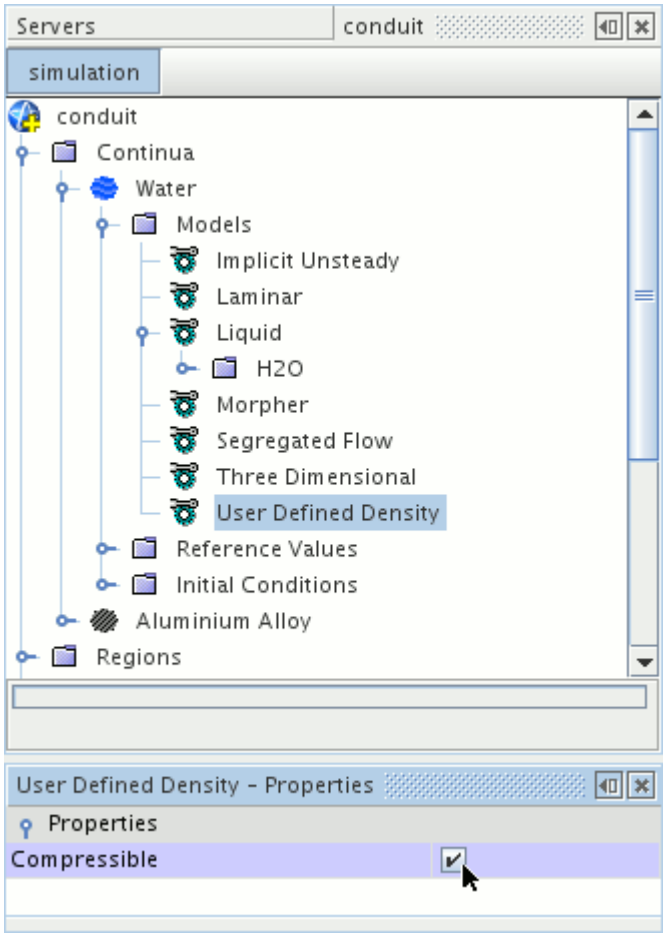
Selecting Physics Models

This section describes the physics models that you select to use large displacements.

1. For the solid continuum, select the **Finite Volume Solid Stress** model and enable Large Displacements by activating the **Solid Displacement** motion.
2. For the fluid continuum, activate the **Morphing** motion so that the fluid mesh can move in response to motion in the solid mesh.

If you are modeling a liquid, it can be useful to employ the **User-Defined Density** model instead of the default **Constant Density** (incompressible) model for the purposes of both numerical stability and accuracy. When you are working with a user-defined density in an FSI case, it can be useful to specify the liquid as compressible. To do so:

3. Activate the **Compressible** option for the **User-Defined Density** model under the **Continua** manager node.



The interaction of a liquid with a solid is often more numerically stable when the liquid is allowed to be slightly compressible, particularly when the flow is internal to the structure. To provide this numerical stability, you can define a user-specified density using a field function. For example, Density `rho = 1000 + $Pressure/($c*$c)`, Density pressure derivative `drhodp = 1/($c*$c)` and Speed of Sound `c = 600`. In this case, it is often beneficial to use a slower speed of sound than the “true” value for the liquid. This setting leads to better numerical stability and faster convergence without a loss of accuracy.

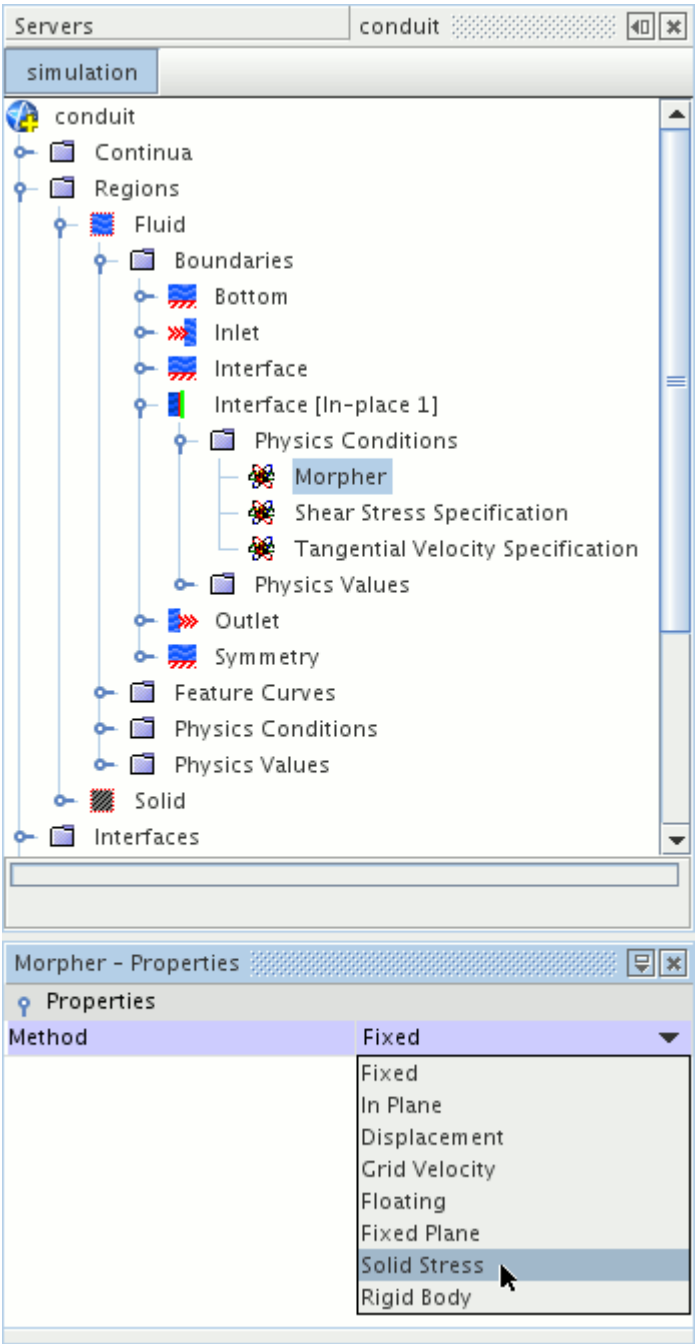
The choice of using a compressible liquid or not depends on the physics and appropriate time scales. When the fluid sound transient times are short relative to other time scales (such as natural frequency of the structure or vortex shedding frequencies), it is possible to reduce the sound speed of the fluid. Reducing this value has little influence on the results, while improving stability and robustness. If the physical relevant time scales are of the same order as the sound transit times (such as an impact or underwater explosion), then it is necessary to account for the exact nature of the fluid compressibility accurately and employ more accurate user-specified density laws for the liquid.

For external flows, it is possible to use an incompressible fluid. If the fluid is a gas, then using a perfect gas law is appropriate. These models can be selected from the *Equation of State* box in the *Physics Model Selection* dialog.

Setting the Fluid Morpher Boundary Condition

Activate the morpher and select a morpher physics condition for the interface boundary.

1. For the fluid region, activate the mesh morphing motion so that the interface boundary in the fluid domain morphs according to the solid stress displacement.
2. Once morphing has been activated for the region, select the morpher physics condition under the boundary interface node and specify **Solid Stress** for the *Method* property.



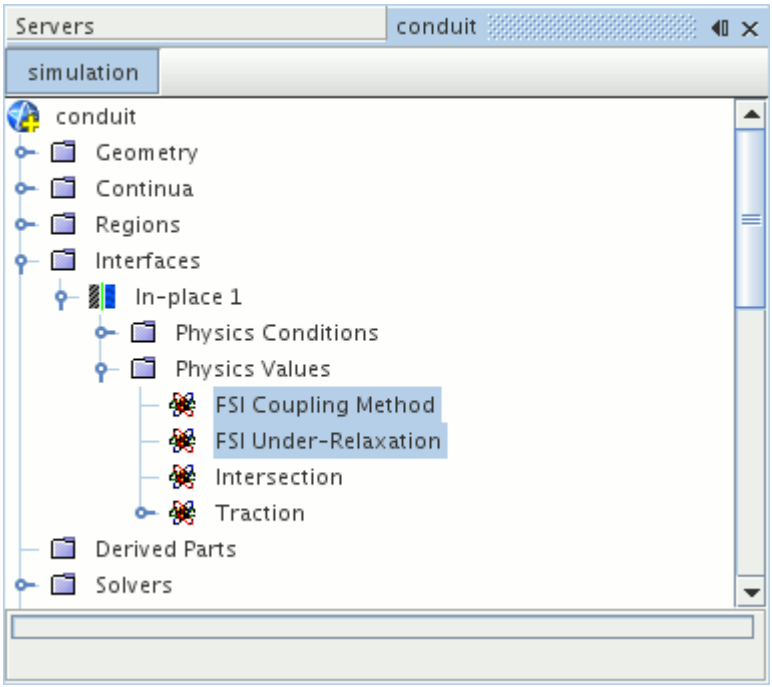
Preparing the Fluid-Structure Interface

At the interface between the solid and fluid domains, there are two nodes available that affect the FSI solution. The **FSI Coupling Method** node allows you to control the frequency of data exchange across the fluid-solid interface. The **FSI Under-Relaxation** node can be used to set the mesh motion under-relaxation factor at the FSI interface.

For an implicit case, set the **FSI Coupling Method** to **Implicit**, which causes data to be exchanged after each inner iteration. The other option is **Explicit**, which causes data to be exchanged after each time-step.

For Explicit, the fluid morphing is only applied once per time-step. For Implicit, there are two options:

- The fluid morpher is executed only on the first iteration of the time-step. Thereafter a correction is made only on the boundary to account for the acceleration of the structure. (Default)
- The fluid morpher is executed for each iteration within a time-step. This function is activated using the *Morph At Inner Iterations* property on the Morpher solver.



Using Small Displacements for FSI

The small displacement theory assumes that the displacements are relatively small compared to the cell size. In this case, the solid mesh is not moved and the adjacent fluid mesh does not need morphed. Instead, the velocity of the solid is imparted as a boundary condition to the fluid.

Generally, the assumption of small displacement FSI is only applicable to vibration problems where the maximum displacements are small. However, the frequency of the oscillations can be large - implying that the velocity and the acceleration of the solid structure are significant. Discussions regarding the compressibility of liquid fluids (see Using Large Displacements for FSI) are still relevant to the small displacement theory.

Using the Summary Report for Bad Cells

You activate the report for solid stress bad-cell statistics by selecting the Cell Remediation model.

This will add a field function named **Solid Stress Bad Cell Indicator**, which is an extension of the Bad Cell Indicator. However, different indices are used to indicate the severity of poor cells. The indices appear as follows:

```
GOOD                      = 0,
BAD                       = 1,
BAD_NO_GOOD_NEIGHBOR      = 2,
GOOD_NO_GOOD_NEIGHBOR     = 3,
BAD_STRESS                = 4,
```

When you initialize the solution or begin a run, a cell quality report will include the following information:

- Total number of cells in each region.
- Total number of bad cells and the percentage.
- Number of bad cells without good neighboring cells, and the percentage.
- Number of good cells without good neighboring cells, and the percentage.

Bad cells which are identified by high stress (an user input value) are not reported as they are based on the stress solution.

Setting Parameters of Other Solvers

The other solvers that become available in STAR-CCM+ involve choices that affect the quality of a simulation with solid stress.

AMG Iteration Tolerance

The default value of the AMG iteration (convergence) tolerance is generally smaller than what is required for the fluid solver. The default is 0.01. Lower values may be necessary for structures with thin members under bending loads.

Implicit Dynamic Time Integration Solvers

In dynamic analysis, there are two methods to integrate the implicit system of equations.

- Euler, First-Order
- Newmark, Second-Order

This selection would be made from the *Temporal Discretization* property of the **Implicit Unsteady** solver.

The Euler Method is not as accurate as the Newmark Method so a smaller time-step is required to obtain the same accuracy. For given time-step size, the Euler method incurs more numerical damping than the Newmark method for modes of vibration which are at a higher frequency than a time-step can resolve. However, some users may find the Euler method more stable and useful for damping out high-frequency, small-wavelength oscillations that are of little consequence to overall transient behavior.

See Also:

What Is the Implicit Unsteady Solver?
AMG Linear Solver Reference

Finite Volume Solid Stress Model Reference

This section details region conditions that are available for the Finite Volume Solid Stress model.

Model Overview

Provided by	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **[Solid Physics Continuum]** > **Models** > **Finite Volume Solid Stress**	
Requires	Physics Models: *Space*: **Three Dimensional** *Material*: **Solid**	
Activates	*Physics Models*	*Constitutive Law*: **Linear Isotropic Elastic**
	Initial Conditions	**Displacement** **Velocity**
	Region Conditions and Values	See Region Conditions and Values.
	Boundary Conditions and Values	See Boundary Conditions and Values .
	Solvers	**FV Stress**
	Field Functions	See Field Functions.
Properties	None	
Actions	None	

Initial Conditions

Displacement	Allows you to specify the initial displacement as a vector profile.
Velocity	Allows you to specify the initial velocity as a vector profile.

Region Conditions and Values

Physics Condition	Associated Physics Values
Body Load Option Allows you to apply a body force on the solid region.	

Physics Condition	Associated Physics Values
***Method*: None** No body forces are applied.	None
***Method*: Specified** Allows you to specify body forces.	**Body Load** Set as a vector profile. **Note:** Gravity and centrifugal loads are body force loads. The gravity or loads due to an accelerating frame of reference are turned on via the gravity option. Centrifugal loads are activated by choosing a rotating reference frame.
	Body Load Derivative Set as a vector profile. This node represents linearization of the body load with respect to x-, y-, [z-] components of the body force, $\mathbf{b}$ in Eqn. (1058). The derivative of the user source term is used to linearize Eqn. (1058). Its value is set to zero by default. Providing a value for the derivative helps stabilize the solution.

Boundary Conditions and Values

Physics Condition	Associated Physics Values
Stress/Displacement Allows you to determine how the stress and displacement across the boundary are specified.	
***Method*: Free** The applied surface traction is zero. Generally much of the surface of a part can be considered free of loads.	None
***Method*: Fixed** The displacement vector is constrained to be zero (all three components).	None
***Method*: Displacement** All three components of the displacement are prescribed a constant vector value or a tabular or field function vector value.	**Displacement** Set as a vector profile. A prescribed vector of zero displacement is identical to a **Fixed** condition.

Physics Condition	Associated Physics Values
***Method*: Normal Displacement** The component of the displacement in the direction normal to the boundary is prescribed. The tangential components are left free and it is assumed that the boundary has zero shear traction.	**Normal Displacement** Set as a scalar profile.
***Method*: Elastic Foundation** Models a compliant foundation, the surface of which is considered frictionless. Where the solid region rests on the foundation the two surfaces will always remain in contact, so no gap can exist between the elastic foundation and the solid region. As a result, the foundation will place a tensile load on the region if the region attempts to pull away from the foundation. The mechanical properties of the elastic foundation are defined by a spring constant K. If A is the area of the boundary, then the spring constant per unit area is K/A which is considered to be uniformly applied across the surface. The normal stress exerted by the foundation is: $$\sigma_n = -\frac{K}{A}\delta_n \quad (1038)$$ where δ_n is the normal displacement as measured outward from the region boundary. A positive normal displacement gives rise to a compressive normal stress.	**Solid Stress Elastic Foundation** Allows you to set K as a scalar profile.
***Method*: Traction** All three components of the surface traction are prescribed.	**Traction** Set as a vector profile, in units of force/area (pressure).
***Method*: Force** The prescribed force is much the same as the prescribed traction. The force vector is divided by the area of the boundary on which it is prescribed. This “average” surface traction is then applied uniformly to the surface regardless of the local orientation of the normal to the surface. Also available for an interface load.	**Force** Set as a vector profile.
***Method*: Pressure** Only the normal component of the surface traction is prescribed. The shear traction components are taken as zero. A positive value of pressure indicates a compressive load while a negative value indicates	**Pressure** Set as a scalar profile.

Physics Condition	Associated Physics Values
tension. Therefore a **Free** condition is the same as the prescribed traction vector load of [0,0,0] or a prescribed pressure load of zero. Also available for an interface load.	
***Method*: Mixed** Applies a combination of constraints and loads to a boundary. It is not possible to apply both a constraint and a load in any one direction, so the conditions are combined by prescribing a displacement constraint along one or two of the principal axes, and a traction load along the remaining axes. To achieve this, both a displacement vector profile and a traction vector profile are entered, and a mixed vector is used to specify which condition will act along each axis. When using this boundary condition it is recommended that all vector profiles are entered with reference to the default laboratory coordinate system. In cases where the vector profile is referenced by a user-specified coordinate system, the vector will be resolved to the laboratory coordinate system before the mixed vector is applied.	**Displacement** Set as a vector profile. **Traction** Set as a vector profile, in units of force/area (pressure). All three components are prescribed. **Mixed Profile** Set as a vector profile. The mixed vector is a dimensionless vector that acts as a reference for differentiating which vector components will be applied as displacements, and which as loads. It is entered as a vector profile, with each value equal to either 1 or 0 (when the value entered is neither 1 nor 0, the value is taken as 1.) A value of 0 corresponds to a displacement constraint, and this is read from the displacement vector. A value of 1 corresponds to a traction load, and this is read from the traction vector. By specifying the mixed vector in this way, a mixed vector of [1, 0, 1], for example, is equivalent to saying “apply a traction load equivalent to the x-component of the traction vector in the x-direction”, “apply a displacement constraint equivalent to the y-component

Physics Condition	Associated Physics Values
	of the displacement vector in the y-direction" and so on.

Field Functions

Solid Strain XX	The measure of strain ε_{xx} in Eqn. (1047) and Eqn. (1048).
Solid Strain XY	The measure of strain ε_{xy} in Eqn. (1047) and Eqn. (1048).
Solid Strain XZ	The measure of strain ε_{xz}in Eqn. (1047) and Eqn. (1048). Activated only in three-dimensional simulations.
Solid Strain YY	The measure of strain ε_{yy} in Eqn. (1047) and Eqn. (1048).
Solid Strain YZ	The measure of strain ε_{yz} in Eqn. (1047) and Eqn. (1048). Activated only in three-dimensional simulations.
Solid Strain ZZ	The measure of strain ε_{zz} in Eqn. (1047) and Eqn. (1048). Activated only in three-dimensional simulations.
Acceleration	The second time derivative of the displacement field: $\ddot{u} = \frac{\partial^2 u}{\partial t^2} = \frac{v^n - v^{n-1}}{\Delta t}$ Activated only when an unsteady time model is used.
Displacement	The displacement field $\mathbf{u}$ in Eqn. (1055).
Solid Stress Equivalent	The equivalent or von Mises stress. It is calculated as: $\sigma_{\mathrm{eq}} = \sqrt{\frac{(\sigma_1-\sigma_2)^2 + (\sigma_2-\sigma_3)^2 + (\sigma_3-\sigma_1)^2}{2}}$
Solid Stress Int. Principal	The intermediate principal σ_2 in the principal stress $\{\sigma_1 \;\; \sigma_2 \;\; \sigma_3\}$. See Stress.
Solid Stress Max. Principal	The maximum principal stress σ_1 in Eqn. (1042).
Solid Stress Max. Shear	The maximum shear stress traction $\tau_{\max}$ in Eqn. (1042).
Solid Stress Mean	The mean stress σ_m in Eqn. (1043).

Solid Stress Min. Principal	The minimum principal stress σ_3 in Eqn. (1042).
Solid Stress Normal Displacement	The normal component of the displacement field $\mathbf{u} \cdot \hat{\mathrm{n}}$.
Solid Stress Pressure	The normal component of the traction on a given surface $(\boldsymbol{\sigma} \cdot \hat{n}) \cdot \hat{n}$.
Solid Velocity	The first time derivative of the displacement field: $\dot{\mathrm{u}} = \frac{\partial \mathrm{u}}{\partial \mathrm{t}} = \frac{\mathrm{u}^n - \mathrm{u}^{n-1}}{\Delta \mathrm{t}}$ Activated only when an unsteady time model is used.
Solid Stress XX	The measure of stress σ_{xx} in Eqn. (1039) and Eqn. (1040).
Solid Stress XY	The measure of stress σ_{xy} in Eqn. (1039) and Eqn. (1040).
Solid Stress XZ	The measure of stress σ_{xz} in Eqn. (1039) and Eqn. (1040). Activated only in three-dimensional simulations.
Solid Stress YY	The measure of stress σ_{yy} in Eqn. (1039) and Eqn. (1040).
Solid Stress YZ	The measure of stress σ_{yz} in Eqn. (1039) and Eqn. (1040). Activated only in three-dimensional simulations.
Solid Stress ZZ	The measure of stress σ_{zz} in Eqn. (1039) and Eqn. (1040). Activated only in three-dimensional simulations.
Thermal Strain	The measure of the thermal strain ε_T in Eqn. (1052).

FV Stress Solver Reference

The FV Stress solver controls the solution of the Finite Volume Solid Stress model in all the continua for which the model is activated.

Least Square Quality Tolerance	Deformation gradients are computed using a “Least Square Method” and a metric is employed to determine if the cell quality is sufficient to compute the local gradient accurately. The Least Square Quality Tolerance defines the metric value below which the cell is pronounced poor and remedial actions are taken on this cell. Smaller values will allow more cells to pass – larger values will mark more cells as poor. The default value is 0.1. Some models will run satisfactorily with tolerances as low as 0.01. The Least Square Quality Tolerance is perhaps the single most important parameter affecting the convergence of the iterative FV solver.	
Successive Over-Relaxation	Controls whether or not to accelerate convergence.	
	Activated	The solver will accelerate convergence.
	Deactivated	The solver will not accelerate convergence.
Temporary Storage Retained	Retains or clears temporary storage at the end of the iteration. Examples include the ap coefficients, the residual, the corrections, the reconstruction gradients and the cell gradients. These quantities will become available as field functions during subsequent iterations.	
	Activated	Retain temporary storage.
	Deactivated	Do not retain temporary storage.
Identify High Stress Cells	Controls whether the *Limiting Equivalent Stress* property is used.	
	Activated	Activate Limiting Equivalent Stress.
	Deactivated	Leave Limiting Equivalent Stress deactivated.
Limiting Equivalent Stress	If this property is activated, then the specified value (default 1000 MPa) is used to flag cells which have equivalent stress above this value. These flagged cells will be treated as bad cells. This marking of high-stress cells will be implemented after 20 iterations to clear the effect of the initial condition. This feature should be used as a last resort. You should run an analysis without activating it and monitor the stress in the solution. If a diverging cell is identified by high equivalent stress values, then activate this feature and use the solution value as an input for this property.	

Solver Frozen	Prevents or allows the solver from updating the solution while iterating. Activating this feature should not be your first choice, as it is primarily a “debugging” tool and not supported for normal simulation work.	
	Activated	The solver will not update the solution.
	Deactivated	The solver will update the solution.

Finite Volume Stress Analysis Guidelines

- FVA (Finite Volume Stress Analysis) has no FEA counterparts for structural elements such as rods, beams, membranes, and shells. These elements may be modeled as three-dimensional components in FVA but some care must be taken to resolve them. For example a shell in bending should have at least 5 cells across the thickness and the cell aspect ratio should be near 1. Shells and beams in bending will also generally converge very slowly. Therefore applications involving sheet metal in bending are not a good application for FVA.
- When generating grids for stress analysis, you should pay attention to stress risers such as reentrant corners, cracks and holes. This is especially the case if you are concerned about the stresses in these regions and need an accurate stress prediction. The grids need to be sufficiently fine to resolve the stresses. The default reference value for defining the surface curvature is 36 points/circle and this should be sufficient for most fillets. But remember that reentrant comers with no fillets may be singular and, no matter how fine the grid, the stresses in the cells adjacent to the singularity will continue to rise as the grid resolution increases. Before constructing the grid, it is a good practice to consider the boundary regions needed to have properly set boundary conditions for the solid stress – not just the solid temperature.
- Traction boundaries (or traction-free boundaries) must be carefully treated in FVA and often require special attention when there are bad cells adjacent to these boundaries. Generally it is a good idea to have a extrusion layer on these boundaries. But the extrusion layer thickness need only be about the nominal size of the cells in that general area. The idea here is to make orthogonal cells without making them too thin. The cell aspect ratio should not be much larger than 4 and should be around 1 if this extrusion is in a thin part in bending. For some geometries the extrusion layers must be necessarily small because there is only a thin layer of material between opposing boundaries. The cells, in this case, should be relatively small in the lateral direction so that cells which span the thickness do not have too large an aspect ratio. The solid/fluid and solid/solid interfaces are treated numerically as traction boundary conditions so rules for extrusions apply in this case as well.
- There must be at least 3 cells that span between opposing solid boundaries. It is a very common error of new users to have plates that are only 1 to 2 cells thick. If thin solid material lies between two other solid materials, then this layer must also be at least 2 cells thick. If you have only 1 cell across the thickness then the solution may not converge!
- The boundary conditions for a static analysis must be sufficient to constrain the part from rigid body rotation and translation. The symptom of ill-constrained models is usually seen as very large displacements or rotations and divergence in the residuals. This is a very common error of new users in both FVA and FEA. The solution will likely not converge.
- It is prudent to check the quality of the cells as indicated by the Solid Stress Bad Cell Indicator. At the initiation of a run, a summary report is provided indicating the overall quality of the solid region mesh for stress analysis. This report gives the number of cells which are considered poor and cell remediation is applied. If the mesh has more than about 1% bad cells then be very suspicious of the grid and the results. Meshes with less than 0.1% bad cells are good quality. Be mindful of the number of poor cells which have no good neighbor cells, because the remediation will be less benign in these areas.
- When thermal stresses are included it is important not to restrain a part in shear. The fixed boundary condition restrains all components of the displacement. This constraint will unrealistically prevent the part from thermally expanding and will generate unrealistic thermal stresses at the fixed boundary. It is as if one is constraining the part to rigidly adhere to an adjacent part which is infinitely stiff and does not thermally expand. It is acceptable to apply this constraint, but be suspicious of the stress results in the

near vicinity of the constraint--especially if the temperature is well above the zero thermal expansion reference temperature. It is better to restrain the part with a fixed normal condition which allows the part to slide freely along this surface. This will prevent translation or rotation but it will generally allow the part to thermally expand realistically. Using symmetry boundaries is also useful for constraining the part. It is also useful to use a displacement boundary defined by a field function of the local temperature. You can assume that the part is freely expanding with the temperature. It is also important not to over-constrain a part. A part prevented from expanding globally can generate unrealistically high compression stress. For example, you could heat up a pipe and fix both ends. This may produce unrealistically high compression stress since in reality the physical parts that would produce that constraint will also expand. Remember that attached parts can also expand, bolts and washers can slip, male and female threads can slip. Often it is best to simply restrain the part to prevent translation and rotation but still allow the part to expand freely.

- Avoid using tetrahedral solid meshes for stress (sparse use is acceptable).
- Solid Stress Divergence is caused by:
  1. Static models which have not been properly constrained from rigid body motion.
  2. Bad quality cells not identified by the automatic cell quality check.
  3. Almost incompressible materials (elastic materials with Poisson ratio of near 0.5).
  4. Crack roots near relatively poor cells.

  If the problems diverge immediately then the most likely cause is most likely [1]. Remember it requires that all rigid body motions be constrained in static models. If reason [2] is causing this problem then normally you will find that the residuals will go down and then start to diverge. This is a very common problem in FVA since the convergence is highly dependent on the grid quality and depends on how well STAR-CCM+ identifies the bad cells that are in the model. For most CFD users the first inclination is to under-relax the solid momentum, but this is not recommended. It will also generally diverge with under-relaxation but it will only diverge slower. The best strategy for removing the divergence problem caused by unidentified bad cells is to activate solver procedures to cap the solid stress effective stress. This is under the expert properties of the FVA Stress Solver. You must define a limiting effective stress that would be considered much larger than the expected peak stress due to the load. The default peak stress is 1 GPA. A good rule of thumb would be to use a value less than 10% of the Young's modulus. Divergence can also occur if the Poisson ratio is too close to 0.5, and you should not use a value greater than 0.45. Finally, if cracks are in the model and the cracks have poor cells, then divergence might be expected. The areas near the cracks need to be meshed properly.
- Slow convergence is caused by:
  1. Bad quality cells not identified by the automatic cell quality check.
  2. SOR and simultaneous solution of flow, thermal and stress in static problems.
  3. Thin members under relatively large bending.

  Bad quality cells may slow or stop convergence. If you see the residual oscillating up and down then generally the accelerated convergence strategy will not work. Selecting the stress cap in the expect mode of the solver may be effective in improving the convergence rate.

  The Successive Over Relaxation SOR under the FVA Stress Solver will be less effective under conditions when solving for the temperature and the displacement simultaneous in steady problems. You may choose to turn off SOR in this case. Often it is best in these thermal stress problems to solve the thermal problem with the solid stress frozen. Once the energy has converged, then turn on the stress solver and freeze the energy solver.

Problems with thin members in bending will be slow to converge--especially when the distance between the supports is greater than about 10 times the thickness of the member. It would be advisable to monitor the displacement near the point of maximum bending displacement. If it is slowly tending to change then you should continue with iterations. This rule applies to “open” thin members where the in-plane stress varies dramatically through the thickness. “Closed” members, like a thin pipe, are an exception to the rule since the bending stresses across the pipe thickness do not change dramatically, and the pipe is of course much stiffer than a pipe with a slot cut down the axis.

- STAR-CCM+ assumes that displacements, rotations and strains are small. The vertices of the solid do not change with time--only the displacements of the cells and face centers do. For large displacements, vertices in the solid mesh are moved based on interpolation of the solid stress displacement and its gradient solution at cell centers. In general, strains should be less than 10% and the rotations should not be more than 10 degrees, to satisfy the small deformation assumption in the FV formulation.
- When you create a solid/solid interface and the solids are different materials, the FV Stress treats this as a material discontinuity and preserves the discontinuity in the strains across this material interface. These materials are treated as if they were welded at the interface (continuous displacements). Contact is not supported. You can imitate contact in static problems by constraining the part, running the solution and determining where the part goes into tension. In the regions the boundary constraints can be removed by setting the area to “free” and then running again.
- When you create a solid/solid interface or a solid/fluid interface then it is prudent to make sure all faces from the original surface are matched. It is possible that some faces are unmatched to the opposing surface and they will be left in the original boundary surface. The default boundary conditions will be adiabatic free surface in the solid and an adiabatic wall in the solid. If these faces are isolated then they will appear much like adiabatic cracks and these cracks can produce unusually high stresses at the solid/solid interface.
- In a transient problem coupled to the fluid, the change in the displacement is introduced as a swept flux into the fluid but the grid does not change. In steady-state problems the displacement of the solid does not change the fluid at all. In a steady-state problem, the pressure and the heat from the fluid affects the solid and the temperature of the solid affects the fluid temperature, but the displacement of the solid has no impact on the fluid.

Contents:

What Constitutes a “Good” Mesh in Finite Volume Stress Analysis?

See Also:

Modeling Large Displacements

What Constitutes a “Good” Mesh in Finite Volume Stress Analysis?

This section provides an overview of how to assess the quality of a mesh in finite volume analyses.

Mesh Resolution

In many respects the requirements on finite volume meshes to resolve displacement and stresses accurately are similar to those for finite element meshes. Regions with stress concentrations need a proper resolution of the mesh. Fillets, the outer diameter of holes, parts that neck down, re-entrant corners, cracks, or welded

surfaces where part of the two surfaces are not fully “wetted” will require some local refinement of the grid. For example, a fillet between two parts which are joined at 90 degrees will require at least 4 cells to span the quarter circle of the radius. A hole should have at least 16 cells to span the circumference. Near the root of a crack, the cells must be highly refined in a region about 2 crack lengths near the surface, with most of the refinement near the root and coarsening as the distance from the crack root increases. One should remember that a elastic model with a crack is “singular”--that is, the stresses are infinite at the tip of the crack. No matter how much one refines the mesh, the numerical stresses in the cell center will continue to increase. Here the principles of “Fracture Mechanics” must be applied to assess the severity of the loads on the crack--that is, whether or not the loads on the crack will have a tendency to cause catastrophic failure or slowly grow the crack under cyclic loads.

There are some differences in the requirements on the finite volume and finite element meshes. The finite volume mesh should have at least two cells that span between surfaces. Unlike a single finite element, at least two cells are needed to form a gradient across the span. At least three cells are recommended.

Cell Quality

The cell quality is the most important factor influencing the convergence and, to a lesser extent, the accuracy of the finite volume method. The finite volume stress analysis is more sensitive to the quality of the mesh than typical CFD. STAR-CCM+ offers many measures of the quality of the grid including skewness, cell-to-cell volume change, and least square gradient. The quality of the grid over all the cells is registered in one field function that is an indicator of bad cells. If less than 0.1% of the cells in the solid stress model are considered poor, then this mesh can be considered of high quality. If the number of cells of poor quality are greater than 1% then this grid is generally poor and more attention to the meshing is required. However, meshes where the number of cells are greater than 1% may be sufficiently good if the normalized residuals converge below 1e-3.

Thin Members under Bending Loads

If a shell is under bending loads then typically more finite volume cells will be required than in a finite elements, in the plane of the shell as well as across the thickness. In most situations with bending stress it is a good rule of thumb to have cells with aspect ratio near one. As the ratio of the distance between supports to the thickness of the shell relative becomes large, greater than 10, keeping the cell aspect ratio high will be extremely important else the members will behave much too stiffly. For this reason, the finite volume stress analysis is not recommended when the ratio of the distance between supports to the thickness becomes greater than 20. This is not to say that, with sufficient cell density between the supports, the bending displacements and stresses cannot be resolved.

This rule of thin members under bending applies to “open” structures. “Closed” structures such as a long pipe or a long conduit can support high bending loads down the axis with the thickness of the pipe or conduit quite small. In this case, the axial stresses do not change appreciably over the thickness of the pipe or conduit wall but rather across the breadth of its diameter. The rule would still apply if large bending occurred in the cross-section of the closed structure.

The user should also be aware that thin members of open structures in bending will converge slowly unless the cell quality is good.

Solid Stress Formulation

This section describes the formulation for solid stress modeling.

Contents:

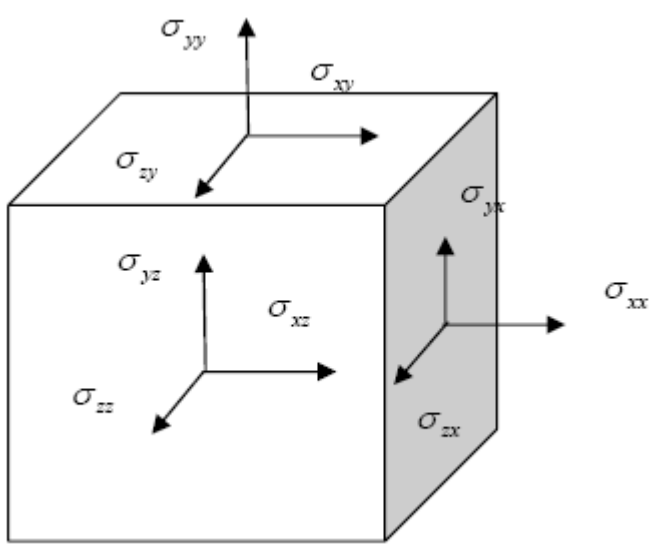
Stress
Deformation
Strain
Thermal Strain
Finite Volume Stress Analysis Methodology
Rayleigh Damping Methodology

Stress

In a free-body diagram of a part, a cross-section of the part is made to expose the internal forces acting on the part. At a point the planes of the cross-section can be cut at many angles – each exposing a different set of internal forces. To describe the stress completely at a point on any plane requires a full tensor description as shown in the free-body of a cube. This tensor is often written in a matrix form or in a vector form:

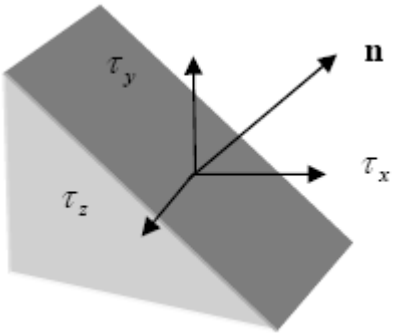
$$\sigma = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{pmatrix} \tag{1039}$$

$$\sigma = \left(\sigma_{xx}\sigma_{yy}\sigma_{zz}\sigma_{xy}\sigma_{yz}\sigma_{zx}\right)^T \tag{1040}$$



Since the sum of moments must go to zero as the cube shrinks to zero, the stress tensor is symmetric. The off diagonal terms are call shear stresses since they act tangentially to the face. The diagonal terms are referred to as normal stresses since they act normal to a face. The traction stress on any plane defined by its unit normal is given as the matrix product of the vector:

$$\tau = \sigma \cdot \mathrm{n}\begin{Bmatrix}\tau_x\\ \tau_y\\ \tau_z\end{Bmatrix} = \begin{pmatrix}\sigma_{xx}\sigma_{xy}\sigma_{xz}\\ \sigma_{yx}\sigma_{yy}\sigma_{yz}\\ \sigma_{zx}\sigma_{zy}\sigma_{zz}\end{pmatrix}\begin{Bmatrix}n_x\\ n_y\\ n_z\end{Bmatrix} \tag{1041}$$



On a certain plane, three orthogonal planes, the shear traction stress is zero and the normal traction stresses on these planes are called the principal stress $\{\sigma_1 \;\; \sigma_2 \;\; \sigma_3\}$. They are ordered by their magnitude $\sigma_1 \geq \sigma_2 \geq \sigma_3$. σ_1 is the maximum principal stress, σ_3 is the minimum principal stress, and σ_2 is the intermediate principal stress. Solid mechanics uses the convention where stresses are positive when in tension and negative in compression so the maximum principal is the most tensile. For this reason it is a often used as measure of tensile failure in a material. It turns out that the principal stresses are the eigenvalues of the stress matrix. In elementary strength of materials, these principal stress are described by the Mohr Circle in two-dimensional stress fields.

On one plane, the shear stress traction is a maximum. The maximum shear stress is the maximum of the differences of the principal stresses:

$$\tau_{\max} = \frac{(\sigma_1 - \sigma_3)}{2} \tag{1042}$$

This is an important measure of ductile failure associated with plastic behavior.

The mean stress is:

$$\sigma_m = \frac{(\sigma_{xx} + \sigma_{yy} + \sigma_{zz})}{3} = \frac{(\sigma_1 + \sigma_2 + \sigma_3)}{3} \tag{1043}$$

If the material is in a hydrostatic state of stress then all principal stresses are equal and the mean stress is interpreted as the negative of the pressure.

If a material is in a state of plane stress, then $\sigma_{zz} = \sigma_{xz} = \sigma_{yz} = 0$. Very thin materials with no loading on the planar surfaces are in a state of plane stress.

Deformation

The deformation of a part is defined by a vector quantity which describes how much a material point on the part is deformed or displaced from its original undeformed configuration.

A perfectly rigid body has points that all move according to a single displacement and rotation vector, but for a deformable body, every point in the body is prescribed a different deformation. If a material point in the original undeformed coordinates is $\mathbf{x}_0$ then the displacement of this point in the deformed configuration is $\mathbf{u}(\mathbf{x}_0)$ and the new coordinates of the point are $\mathbf{x} = \mathbf{x}_0 + \mathbf{u}(\mathbf{x}_0)$. Often in fixed, stiff structures the deformations are generally small and some useful approximations can be made to simplify the mathematical model to describe the deformation. The “small displacement” formulation in STAR-CCM+ relies on the assumption that the displacement is small compared to the cell size. The "large displacement” formulation removes this restriction.

The initial release of the solid mechanics features in STAR-CCM+ will rely on the assumption of small displacement.

The deformation gradient $\nabla\mathbf{u}$ measures how the deformation changes from point to point:

$$\nabla\mathbf{u} = \begin{pmatrix} \frac{\partial u}{\partial x_0} & \frac{\partial u}{\partial y_0} & \frac{\partial u}{\partial z_0} \\ \frac{\partial v}{\partial x_0} & \frac{\partial v}{\partial y_0} & \frac{\partial v}{\partial z_0} \\ \frac{\partial w}{\partial x_0} & \frac{\partial w}{\partial y_0} & \frac{\partial w}{\partial z_0} \end{pmatrix} \tag{1044}$$

where the Cartesian components of material point $\mathbf{x}_0$ are $\{x_0 \;\; y_0 \;\; z_0\}$ and the Cartesian components of the displacement field $\mathbf{u}$ are $\{u \;\; v \;\; w\}$.

See Also:

Modeling Large Displacements

Strain

Strain is the measure of how much a body deforms.

If two material points on an undeformed body are originally a distance of l_0 and in some deformed configuration the two points are distance l, the strain ε is then measured as:

$$\varepsilon_G = \frac{(l - l_0)}{l_0} \tag{1045}$$

or:

$$\varepsilon_A = \frac{(l - l_0)}{l} \tag{1046}$$

The former is what is referred to as a Green strain and the latter is called the Almansi strain. When the strain is small, these measures of strain are almost identical. Like the stress, to describe the strain completely at a point requires a tensor:

$$\varepsilon = \begin{pmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz} \end{pmatrix} \tag{1047}$$

$$\varepsilon = \left(\varepsilon_{xx}\ \varepsilon_{yy}\ \varepsilon_{zz}\ \varepsilon_{xy}\ \varepsilon_{yz}\ \varepsilon_{zx}\right)^T \tag{1048}$$

The strain tensor is also symmetric, the diagonal terms are called normal or extensional strains and the off-diagonal terms are called shear strains. For small strains the relationship between deformation and strain is:

$$\varepsilon = \frac{1}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right) = \left\{\frac{\partial u}{\partial x_0}\ \ \frac{\partial v}{\partial y_0}\ \ \frac{\partial w}{\partial z_0}\ \ \frac{1}{2}\left(\frac{\partial u}{\partial y_0} + \frac{\partial v}{\partial x_0}\right)\ \ \frac{1}{2}\left(\frac{\partial w}{\partial y_0} + \frac{\partial v}{\partial z_0}\right)\ \ \frac{1}{2}\left(\frac{\partial u}{\partial z_0} + \frac{\partial w}{\partial x_0}\right)\right\}^T \tag{1049}$$

If one inscribed a 90-degree angle on the undeformed material, then the shear strains can be geometrically interpreted as half the change in this angle in the deformed state. For this reason, the engineering strain has a slightly different definition:

$$\varepsilon_{eng} = \left(\varepsilon_{xx}\ \varepsilon_{yy}\ \varepsilon_{zz}\ \gamma_{xy} = 2\varepsilon_{xy}\ \gamma_{yz} = 2\varepsilon_{yz}\ \gamma_{zx} = 2\varepsilon_{zx}\right)^T \tag{1050}$$

Adopting the practice in most finite element codes, STAR-CCM+ output strains follow the engineering convention where the shear strains are twice the conventional shear strains. The volume strain is the change of volume relative to the original volume, and is the sum of the normal strains:

$$\frac{\Delta V}{V} = \varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz} \tag{1051}$$

A material is in a state of plane strain when $\varepsilon_{zz} = \varepsilon_{xz} = \varepsilon_{yz} = 0$.

Thermal Strain

Thermal strain is the measure of the elongation of the material due to changes in the temperature.

In an unconstrained state, a homogeneous linear elastic isotropic material with no surface loads is at a uniform temperature, and the material will freely expand and produce no stresses. If the part is constrained, if the part is made of different materials, or if the temperature is changing in the part then thermal stresses will ensue.

In a linear isotropic elastic material the thermal strain is a linear function of temperature and impacts only the volume change in the material. The thermal strain is uniquely described by the coefficient of thermal expansion α and the expansion reference temperature T_{ref}:

$$\varepsilon_T(T) = \alpha\left(T - T_{\mathrm{ref}}\right) \tag{1052}$$

The expansion reference temperature is the temperature at which the thermal strain is zero. In the linear elastic material the mean stress is related to the volumetric strain:

$$\sigma_{\mathrm{m}} = K\left(\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz} - 3\varepsilon_T(T)\right) \tag{1053}$$

This is the linear elastic analog of the equation of state in a fluid relating the pressure, the density and the temperature.

In the full tensor notation the stress is:

$$\sigma = 2G\left(\varepsilon - \varepsilon_T \delta\right) + \lambda\left(\mathrm{tr}\left(\varepsilon\right) - 3\varepsilon_T\right)\delta \tag{1054}$$

Finite Volume Stress Analysis Methodology

The differential form of the solid momentum equation is:

$$\rho \ddot{\mathbf{u}} = \nabla \cdot \sigma + \mathbf{b} \tag{1055}$$

where $\mathbf{u}$ is the displacement field, σ is the stress tensor field and $\mathbf{b}$is the body force.

A general constitutive relation between the stress tensor and the strain tensor is assumed:

$$\sigma = \sigma\left(\varepsilon, T, \kappa\right) \tag{1056}$$

where T is the temperature and κ represents other internal variables. An assumption of small strain allows the strain to be expressed as:

$$\varepsilon = \frac{1}{2}\left(\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \frac{\partial \mathbf{u}^T}{\partial \mathbf{x}}\right) = \frac{1}{2}\left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right) \tag{1057}$$

The finite volume form is determined by integration of parts to obtain:

$$\int_V \rho \ddot{\mathbf{u}} \, \mathrm{d}\, V = \int_A \sigma \, \mathrm{d}\, \mathbf{s} + \int_V \mathbf{b} \, \mathrm{d}\, V \tag{1058}$$

For all cells i in the domain V, the discrete form of the momentum equation becomes:

$$\rho_i \ddot{\mathbf{u}}_i V_i = \sum_{j \in f(i)} \sigma_j \cdot \mathbf{s}_j + \mathbf{b}_i V_i \quad \forall i \tag{1059}$$

where the sum is over all the faces j that form cell i with normal surface vector $\mathbf{s}_j$. If one defines the surface force on face j as:

$$\mathbf{f}_j = \sigma_j \cdot \mathbf{s}_j \tag{1060}$$

then Eqn. (1059) becomes:

$$\rho_i \ddot{\mathbf{u}}_i V_i = \sum_{j \in f(i)} \mathbf{f}_j + \mathbf{b}_i V_i \quad \forall i \tag{1061}$$

For the purposes of illustration we will consider only one material on a single domain. In addition, to simplify the initial analysis we will consider only a static or quasi-static problem and drop the inertial term. In this case the sum of the forces must be zero.

$$\mathbf{r}_i = \sum_{j \in f(i)} \sigma_j \cdot \mathbf{s}_j + \mathbf{b}_i V_i \quad \forall i \tag{1062}$$

In order to compute the stress tensor on the centroids of the faces we need to establish the discrete form of the strain tensor ε and the deformation gradient $\nabla \mathbf{u}$. The deformation gradient is computed at cell centers

based on the difference in displacement between neighboring points. In an identical manner to viscous fluxes in the fluid momentum, the deformation gradients are interpolated to the face between two cell centers as:

$$\nabla \mathbf{u}|_f = \overline{\nabla \mathbf{u}} + \underbrace{\left(\delta \mathbf{u} - \overline{\nabla \mathbf{u}} \cdot \mathbf{d}\right)}_{\text{at least first order}} \otimes \frac{\mathbf{s}}{\mathbf{d} \cdot \mathbf{s}} \tag{1063}$$

Where $\overline{\nabla \mathbf{u}}$ is the mean of the gradient computed at the adjoining face centers $\delta \mathbf{u}$ is the difference in the displacement between two adjoining cell centers, and $\mathbf{d}$ is the vector between the two adjoining cell centers. This deformation gradient is at least first-order accurate. The strain (Eqn. (1057)) on the face center is determined and Eqn. (1056) provides the stress on the face.

An approximation of the stiffness matrix can be established including only the terms which depend on $\delta \mathbf{u}$ in the deformation gradient along with the tangent modulus $\partial \sigma / \partial \varepsilon$ supplied by the material model.

The approximation of the stiffness matrix can be accumulated over all faces:

$$K_{ik} = -\frac{\partial \mathbf{r}_i}{\partial \mathbf{u}_k} = -\sum_{j \in f(i)} \frac{\partial \mathbf{f}_j}{\partial \mathbf{u}_k} \quad \forall i, k \in N(i) \tag{1064}$$

It can then be employed in a Newton's method to correct for the displacement:

$$\mathbf{K}_{ik} \Delta \mathbf{u}_k = \mathbf{r}_i^n, \ \Delta \mathbf{u}_k = \mathbf{u}_k^{n+1} - \mathbf{u}_k^n \tag{1065}$$

This linear system of equations is solved using the AMG iterative solver. For more information, see Algebraic Multigrid.

In transient problems the inertial terms are included so that the residual becomes:

$$\mathbf{r}_i = \sum_{j \in f(i)} \sigma_j \cdot \mathbf{s}_j + \left(\mathbf{b}_i - \rho_i \ddot{\mathbf{u}}_i\right) V_i \tag{1066}$$

The acceleration terms are modeled either with a 1st order (Backward Euler) or a 2nd order (Newmark) method. Both schemes provide an implicit integration of the system of equations, subject to the initial conditions for the displacement and the velocity. Both methods are absolutely stable regardless of step size.

Rayleigh Damping Methodology

Rayleigh damping is often added in dynamic problems to mimic either real physical damping or to include some numerical damping to damp unwanted numerical oscillations.

The damping can be expressed as:

$$\mathbf{C} = \frac{\mathbf{M}}{\tau_M} + \tau_K \mathbf{K} \tag{1067}$$

where $\mathbf{C}$ is the viscous damping matrix, $\mathbf{M}$ is the mass matrix, $\mathbf{K}$ is the stiffness matrix, $1/\tau_M$ is the mass damping coefficient (alpha), and τ_K is the stiffness damping coefficient (beta).

In the FV stress analysis we do not compute the stiffness matrix; an equivalent law for viscous stress is:

$$\boldsymbol{\sigma}_d = 2G\dot{\varepsilon} + \lambda tr\left(\dot{\varepsilon}\right)\boldsymbol{\delta} \tag{1068}$$

Where the rate of change of the strain, $\dot{\varepsilon}$, is simply from the deformation rate gradient:

$$\dot{\varepsilon} = \frac{1}{2}\left(\nabla\dot{\mathbf{u}} + \nabla\dot{\mathbf{u}}^T\right) \tag{1069}$$

The total stress is then the sum of the elastic and viscous damping parts:

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}_{\mathbf{e}} + \boldsymbol{\sigma}_{\mathbf{d}} \tag{1070}$$

By supplying the elastic model with pseudo strain of the form,

$$\widetilde{\boldsymbol{\varepsilon}} = \boldsymbol{\varepsilon} + \tau_K\dot{\boldsymbol{\varepsilon}} \tag{1071}$$

the elastic model will return the total stress due to both the elastic and viscous damping.

Electromagnetism

STAR-CCM+ allows you to model engineering applications involving electromagnetic phenomena. Example applications are electric motors, electric switches, and transformers, which can be modeled based on the classical theory of Electromagnetism.

The governing equations and their implementation in STAR-CCM+ are presented in the Electromagnetism section of the Theory Guide.

Target Applications and Models Overview

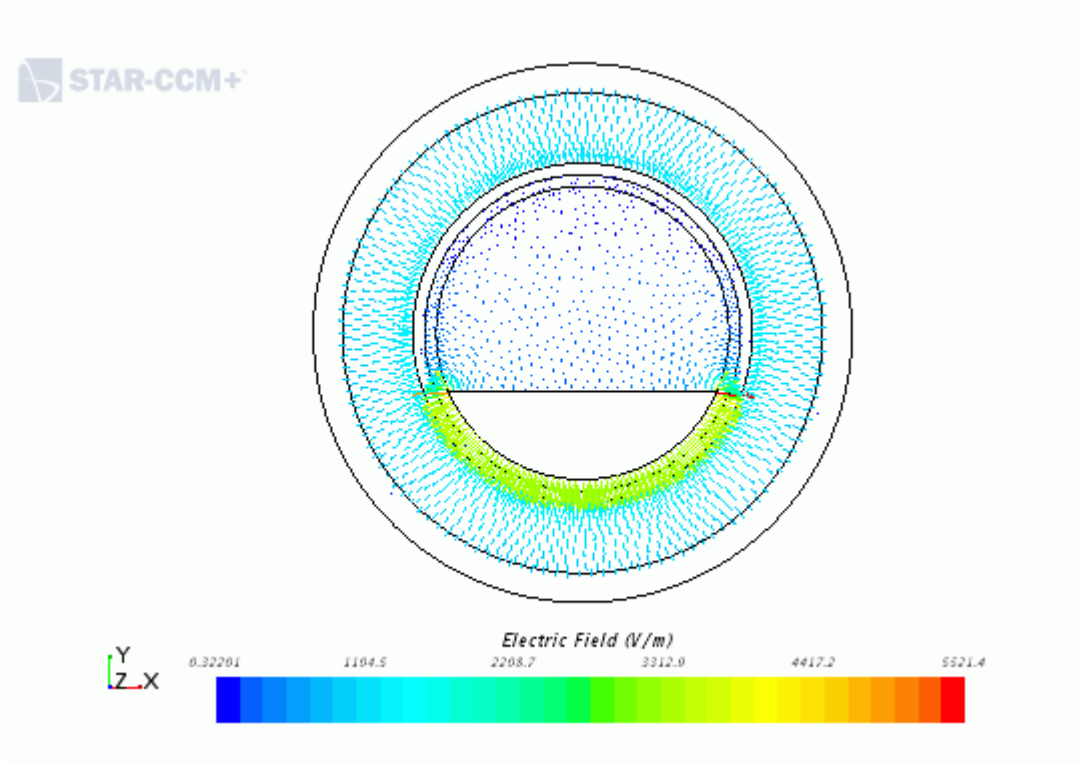
The Electromagnetism module in STAR-CCM+ includes several physics models that target a variety of applications:

Electrostatics

Electrostatic simulations model electric charges in dielectric or poorly-conducting materials. Electrostatic applications assume that the distribution of electric charges does not vary with time, or that the variation occurs over relatively long time scales. Dynamics effects can therefore be neglected.

In STAR-CCM+, you model Electrostatic phenomena using the **Electrostatic Potential** model, which calculates the electric potential due to the electric charge density.

The following example shows the distribution of the electric field in a cylindrical capacitance sensor.



Setup procedure: see Modeling Electrostatics.
Model associations and properties: see Electrostatic Potential Model Reference.

Low-Frequency Electromagnetics

STAR-CCM+ allows you to model low-frequency electromagnetic fields in conducting media. For low-frequency, the conduction currents in the conducting material are orders of magnitude greater than the displacement currents, which are therefore neglected.

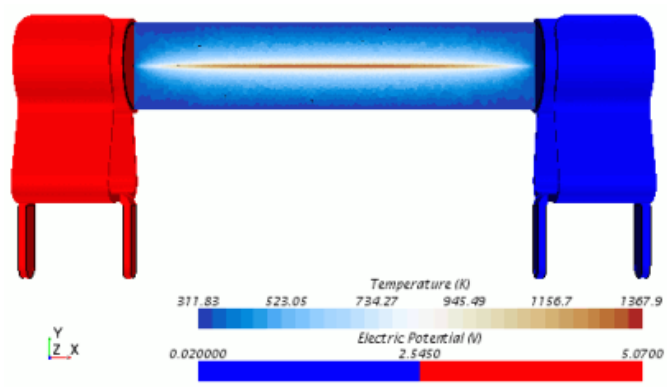
In general, for fields that vary with time, electric and magnetic effects are mutually coupled. Electric currents induce magnetic fields, and magnetic fields induce electric currents. STAR-CCM+ allows you to model electric currents and magnetic fields independently, or together as a single coupled problem.

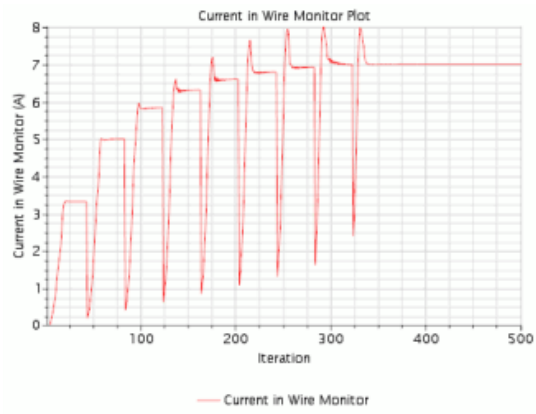
Electric Currents

Electrodynamic simulations model electric currents in conducting materials.

In STAR-CCM+, you model electric currents using the electrodynamic potential models, which calculate the electric potential due to electric currents.

The following example shows the simulation of electric currents flowing through a domestic fuse, increasing its temperature.





Setup procedure: see Modeling Electric Currents.
Model associations and properties: see Electrodynamic Potential Model Reference.

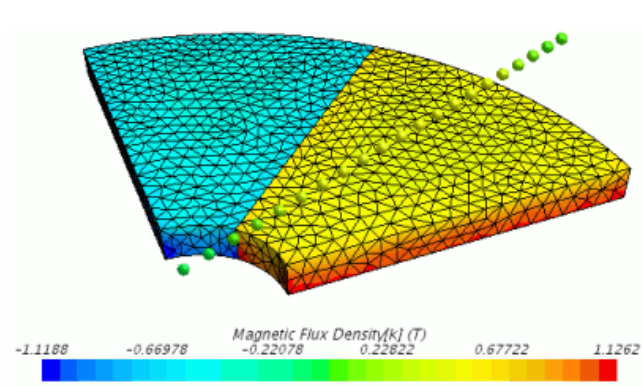
Magnetic Effects

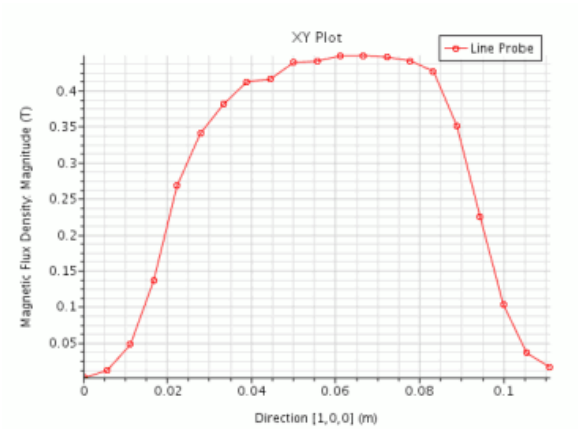
Magnetic field simulations model the magnetic field induced by permanent magnets, excitation coils, and other sources of electric current density.

In STAR-CCM+, you model magnetic effects using the magnetic vector potential models, which calculate the magnetic vector potential due to the electric current density, using either the finite element or the finite volume method. The finite element implementation is ideal for cases with spatial variation in magnetic permeability. The finite volume implementation requires less memory, but it is not suitable for cases with spatial variation in magnetic permeability.

For 2D and axisymmetric transverse magnetic modes (TM), where the magnetic field lies on the plane of a 2D domain, STAR-CCM+ provides the transverse magnetic potential models.

The following example shows the simulation of an axial-flux motor, in which the magnetic flux density induced by permanent magnets produces a torque.





Setup procedure: see Modeling Magnetic Fields.
Model associations and properties: see Magnetic Vector Potential Models Reference.

The following workflow sections assume that you are already familiar with general STAR-CCM+ tasks, such as importing or generating a volume mesh, activating physics models, selecting materials, and creating

reports and scenes. For more information on the STAR-CCM+ general workflow, see STAR-CCM+ Simulation Process.

Note: For electromagnetic simulations, you require a double-precision version of STAR-CCM+.

Contents:

Modeling Electrostatics
Modeling Electric Currents
Modeling Magnetic Fields
Modeling Ohmic Heating
Modeling Magnetohydrodynamics (MHD)
Physics Models Reference
Using the Circuit Model
Accessing Electromagnetism Constants
Troubleshooting

Modeling Electrostatics

The following steps show how to model electric fields induced by static or quasi-static distributions of electric charges.

In electrostatic applications, STAR-CCM+ solves Eqn. (3724) for the electric potential and then calculates the associated electric field $\mathbf{E}$, and electric flux density $\mathbf{D}$.

1. Before you set up the physics continua, prepare the required geometry, regions, interfaces, and mesh, as appropriate to your analysis.

   For information on these general operations, see STAR-CCM+ Simulation Process.
2. Create physics continua and assign them to the relevant regions.
3. In each physics continuum for which you want to model Electrostatics, include the following models:

Group Box	Physics Model
Space	Either **Three Dimensional**, **Two Dimensional**, or **Axisymmetric**.
Time	• To run a static analysis, where the electric charge density does not depend on time, activate the **Steady** model. • To model time-dependent electric charge distributions, activate the **Implicit Unsteady** model. As Eqn. (3724) does not contain any transient term, the analysis is still static. At each time-step, STAR-CCM+ computes the electric potential induced by the electric charge density defined at that time-step.
Material	Choose the relevant material model.
Optional Models	**Electromagnetism**
Electromagnetism	**Electrostatic Potential**

   If relevant, you can use the **Electrostatic Potential** model together with the following models:

   - **Electrochemical Species**—accounts for the distribution of electrochemical species within a fluid (see Modeling Ionic Species Flux).
   - **Plasma**—models non-LTE plasma (see Modeling Cold, Non-Thermal Plasma).
   - **Lagrangian Multiphase**—models the motion of electrically charged material particles (see Charged Particle Models).

By default, the electric charge density ρ within a region is set to zero. To define an electric charge density for the region:

4. Expand the relevant **Regions** > **[Region]** > **Physics Conditions** node.
5. Select the **Electric Charge Density Source** node and activate *Electric Charge Density User Source*.
6. Select the **Physics Values** > **Electric Charge Density User Source** node and specify ρ as a scalar profile.

7. When ρ is a nonlinear function of the electric potential, you can specify the nonlinear relationship by setting the **Electric Charge Density Source Potential Derivative**. This generally improves the convergence of the electric potential solver.

At the domain boundaries, specify either the electric potential or its normal derivative. To define the boundary conditions:

8. Expand the relevant **Regions** > **[Region]** > **Boundaries** > **[Boundary]** > **Physics Conditions** node.
9. Select the **Electric Potential Specification** node and set *Method* to either:
   - **Specific Electric Flux**
   - **Electric Flux**
   - **Electric Flux Density**
   - **Electric Potential**
10. Expand the boundary **Physics Values** node and specify the boundary value under the relevant value node.

To model the interface between a gas and a dielectric (non-conducting) material, you specify the surface charge density at the interface. Example applications include ionic wind and capacitatively coupled plasma (see Modeling Ionic Species Flux and Modeling Cold, Non-Thermal Plasma).

To specify the surface charge density at a contact interface:

11. Expand the relevant **Interfaces** > **[Interface]** > **Physics Conditions** node.
12. Select the **Surface Electric Charge Option** node and set *Method* to **Surface Charge Density**.
13. Select the **Physics Values** > **Surface Charge Density** node and specify the appropriate surface charge density profile.

    You can also set a specific electric flux or total electric flux value on an interface using the **Electric Flux Source Option** node.

Modeling Electric Currents

The following steps outline the recommended workflow for modeling electric currents in conducting media.

In electrodynamic simulations, STAR-CCM+ computes the electric potential ϕ and calculates the associated electric field $\mathbf{E}$ and electric flux density $\mathbf{J}$.

For a practical example of modeling electric currents in STAR-CCM+, refer to the tutorial, Ohmic Heating: Domestic Fuse, which demonstrates how to use the **Electrodynamic Potential** model together with **Ohmic Heating** model to simulate the temperature increase within a domestic fuse due to the flow of electric currents.

The required steps for modeling electric currents are:

1. Activating one of the Electrodynamic Potential models.
2. Defining the electrical conductivity of the conducting materials.
3. Defining boundary conditions, and, if relevant, interface conditions.

To account for unresolved physics, you can prescribe region sources as a transfer current density or an electromotive force density.

Contents:

Selecting the Physics Models and Materials
Defining the Electrical Conductivity
Setting a Transfer Current Density Region Source
Setting the Electromotive Force Density
Defining Boundary Conditions

Selecting the Physics Models and Materials

The **Electrodynamic Potential** model solves for the electric potential to calculate the electric field and the electric current density.

1. Before you set up the physics continua, prepare the required geometry, regions, interfaces, and mesh, as appropriate to your analysis.

   For more information on these general operations, see STAR-CCM+ Simulation Process.
2. Create physics continua and assign them to the relevant regions.
3. In each physics continuum, activate physics models as required. To model electric currents in conducting regions, include the following models:

Group Box	Physics Model
Space	• To model a 3D conducting region, activate **Three Dimensional**. • To model a 2D conducting region, activate either **Two Dimensional** or **Axisymmetric**.

Group Box	Physics Model
	When using the **Axisymmetric** model, make sure that you use an appropriate mesh resolution near the axis. • To model a conducting solid shell region, activate **Shell Three Dimensional**. You typically use this model in electrochemistry applications to model the electrical conductivity in thin shells. The thickness of the conducting shell must be the same at each location on the shell. However, you can modify the shell thickness during the simulation by specifying the thickness using a field function. For more information on shell regions, see Solid Shell Region and Solid Shell Thickness.
Time	Activate either **Implicit Unsteady** or **Steady**.
Material	Choose the relevant material model. You can model electric currents in single-component or multi-component gases, solids, and liquids. Conducting shells are supported only for solid materials. For multi-component materials: • The **Multi-Part Solid** model allows you to define different electrical conductivities for different solid parts within the same physics continuum. • The **Multi-Component Gas** and **Multi-Component Liquid** models allow you to define the effective electrical conductivity of the mixture (you cannot define the electrical conductivity as a function of the conductivities of the components).
Optional Models	Activate **Electromagnetism**.
Electromagnetism	To solve for the electric potential, activate one of the following models: • **Harmonic Balance FV Electrodynamic Potential**—specifically for potentials with sinusoidal time dependence. This model requires the **Harmonic Balance FV Magnetic Vector Potential** model. • **Electrodynamic Potential**—suitable for all other cases.

If relevant, you can activate additional models:

- To include the eddy currents induced by time-varying magnetic fields in transient applications, activate one of the magnetic vector potential models (see Modeling Magnetic Fields).
- To model electrochemical reactions and species, activate the appropriate electrochemistry model (see Modeling Electrochemistry).

  For example, you can simulate corrosion and determine the effectiveness of various cathodic protection scenarios.
- To account for the heat dissipated by electric currents in heat transfer analyses, activate the **Ohmic Heating** model (see Modeling Ohmic Heating).
- To model the interaction between electrically conducting fluids (such as molten metals and plasma) and a magnetic field, activate one of the MHD models (see Modeling Magnetohydrodynamics (MHD)).

4. For each continuum, select appropriate materials from the STAR-CCM+ material library.

   For more information, see STAR-CCM+ Simulation Process.

Defining the Electrical Conductivity

When modeling electric currents in conducting materials, you specify the material electrical conductivity σ, which defines the relationship between the electric current density $\mathbf{J}$ and the electric field $\mathbf{E}$ according to Ohm's law.

For more information, see Electrical Conductivity: Generalized Ohm's Law and Eqn. (3699).

The procedure for defining the electrical conductivity depends on the region type. For fluid and solid regions, you specify the electrical conductivity under the physics continuum associated with the region:

1. Expand the relevant **Continua** > **[Physics Continuum]** > **[Material Model]** > **[Material]** node.
2. Select the **Material Properties** > **Electrical Conductivity** node and choose the *Method* for defining σ.
   - For fluids and isotropic solids, you specify σ as a scalar profile. For anisotropic solids, you specify σ as a tensor profile. For more information on the available methods, see Electrical Conductivity Reference.
   - When using harmonic balance models, you specify σ as a complex profile. Define either its real and imaginary parts, or its phase and magnitude.

In fluid regions, you can artificially raise the electrical conductivity of a gas by specifying a minimum electrical conductivity. For example, in a plasma flow simulation, you can artificially raise the electrical conductivity of the gas in a particular region to model the ignition of an electrical spark. The current flow through this artificial channel heats up the gas so that the gas conductivity soon exceeds the artificial minimum conductivity. To specify the minimum electrical conductivity within a fluid region:

3. Expand the relevant **[Fluid Region]** > **Physics Conditions** node.
4. Select the **Electrical Conductivity Constraint Option** and set *Option* to **Minimum**.
5. Select the **[Fluid Region]** > **Physics Values** > **Minimum Constraint Electrical Conductivity** node, and set the minimum electrical conductivity to a value of your choice.

For porous regions, you define the effective conductivity of the region in terms of the electrical conductivities of the fluid and solid components of the porous region, as follows:

6. Expand the relevant **Regions** > **[Region]** > **Physics Conditions** node.
7. Select the **Effective Porous Electrical Conductivity Option** node and choose the *Method* for defining σ.

   The **Bruggeman (solid)** and **Bruggeman (fluid)** methods define the effective conductivity as a function of σ_{solid} and σ_{fluid}, respectively. The **Volume Fraction Weighted** method defines the effective conductivity as a function of both σ_{solid} and σ_{fluid}. For more information, see Eqn. (3695).
8. To specify σ_{fluid}, in the physics continuum associated with the porous region, select the **[Fluid Continuum]** > **[Fluid Material]** > **[Material]** > **Material Properties** > **Electrical Conductivity** node and define σ_{fluid} as a scalar profile.
9. To specify σ_{solid} select the **Regions** > **[Region]** > **Physics Values** > **Solid Electrical Conductivity** node and define σ_{solid} as a tensor profile.
10. When using one of the Bruggeman methods, also specify the Bruggeman exponent β (see Eqn. (3695)) using the **[Region]** > **Physics Values** > **Bruggeman Approximation Exponent** node.

Setting the Transfer Current Density

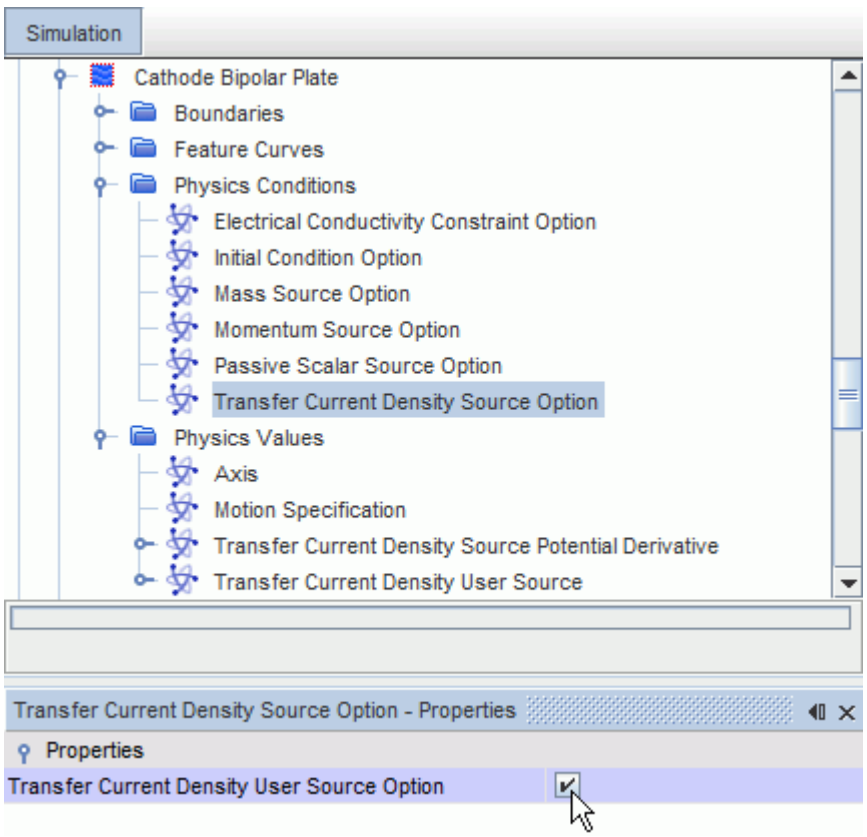
The Transfer Current Density $\mathbf{S}$ is a user-specified contribution to the electric current within a specified region.

The Transfer Current Density region source is primarily used in electrochemistry applications for modeling volume averaged electric current generation in porous regions. The Transfer Current Density (S_ϕ in Eqn. (3727)) is specified as the Transfer Current Density User Source scalar profile within a region. When you set this value, take care not to violate the incompressible nature of the electric current. This condition implies that the applied source terms must be in balance within the solution domain. Alternatively, there must be at least one boundary patch on the solution domain with the Dirichlet boundary condition (prescribed voltage) for electric potential.

When the current source is a nonlinear function of electric potential, the **Transfer Current Density Source Potential Derivative** setting can help to improve the convergence of the electric potential solver.

To define the transfer current density source within a region:

1. Select the **Transfer Current Density Source Option** node within the **Physics Conditions** folder for the region.
2. In the *Properties* window, activate **Transfer Current Density User Source Option**.



   The corresponding nodes, **Transfer Current Density User Source** and **Transfer Current Density Source Potential Derivative**, are added to the **Physics Values** folder for the region.
3. Set the appropriate physics values.

The default options use a constant value. The alternative options are described in the scalar profile properties section.

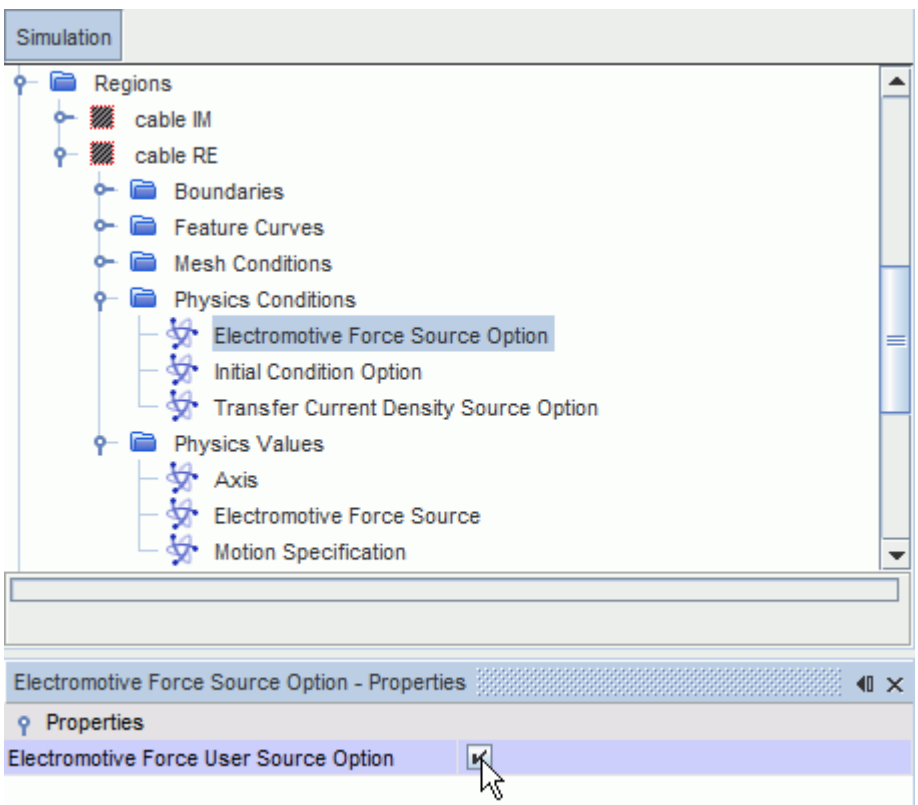
Setting the Electromotive Force Density

The Electromotive Force Density ρ_ϵ is a user-specified contribution to the electric field within a specified region.

Transient electromagnetic effects such as magnetic induction produce a contribution to the electric field, and hence contribute to the electric current density. The electromotive force density ($\boldsymbol{\rho}_\epsilon$ in Eqn. (3727)) is specified as the Electromotive Force Source vector profile within a region.

To define the Electromotive Force Source within a region:

1. Select the **Electromotive Force Source Option** node within the **Physics Conditions** folder for the region.
2. In the *Properties* window, activate **Electromotive Force User Source Option**.



The corresponding node, **Electromotive Force Source**, is added to the **Physics Values** folder for the region.

3. Set the appropriate physics value.

   The Electromotive Force Source is specified as a vector profile. The default option uses a constant value. The alternative options are described in the vector profile properties section.

Defining Boundary Conditions

To allow the current to enter and leave the domain, set the electric potential or specific electric current at the “inlets” and “outlets” to the electrified domain, for example at the terminals. The current density is considered to be solenoidal, which is equivalent to an incompressibility condition for fluids. Therefore, the amount of current going into the domain is the same as the current leaving it.

When defining the boundary conditions, different techniques are available. For example, consider a wire for which the resistance *R* and the current *I* are known. In this case, there are three possible ways to set the boundary conditions:

- Knowing *I* and the inlet area, you can determine *J*. You then apply it with a positive sign at the inlet, and define a constant electric potential at the outlet. (Recommended).
- Based on *I* and *R*, you can deduce the potential difference. You then specify the electric potential at both ends of the wire to reflect the potential difference.
- Knowing *I* and the inlet area, you can determine *J*. You then apply it with a positive sign at the “inlet” and a negative sign at the “outlet”.

Specify appropriate boundary conditions at each boundary:

1. Expand the relevant **Regions** > **[Region]** > **Boundaries** > **[Boundary]** > **Physics Conditions** node.
2. Select the **Electric Potential Specification** node and set *Method* as appropriate:
   - To prevent electric current from crossing the boundary, select **Insulator**.
   - To specify the electric potential, the electric current density, the specific electric current density, or the total electric current at the boundary, select the corresponding method.
   - To define a characteristic relation of electric current to electric potential drop at the boundary, select **Electric Current Potential Characteristic**.
3. Based on the selected method, specify the relevant quantities under the **Boundaries** > **[Boundary]** > **Physics Values** node.

   For information on the available methods and associated values, see:
   - Electrodynamic Potential Model Boundary Inputs
   - Harmonic Balance FV Electrodynamic Potential Model Boundary Inputs

As with boundaries, you can also set a specific electric current or total electric current value on an interface. For more information, see:

- Electrodynamic Potential Model Interface Inputs
- Harmonic Balance FV Electrodynamic Potential Model Interface Inputs

Contents:

Setting the Electric Current Potential Characteristic

Setting the Electric Current Potential Characteristic

For some engineering applications, it is of interest to prescribe a characteristic relation of electric current to electric potential drop at a boundary or an interface. This characteristic relation can also be known as the potential-voltage characteristic or polarization curve.

Example applications where it is convenient to define the I-V characteristic at boundaries are:

- cathodic protection, in which the complexity of the electro-chemical corrosion process is reduced to the polarization curve of a specific setup
- diodes
- photovoltaic elements
- electron/ion emission into plasmas

You can specify an *Electric Current Potential Characteristic* condition at boundaries and contact interfaces when the **Electrodynamic Potential Model** is selected. For example:

1. Select the **[Region]** > **Boundaries** > **[Boundary]** > **Physics Conditions** > **Electric Potential Specification** node and set the *Method* to **Electric Current Potential Characteristic**.
2. Select the **[Region]** > **Boundaries** > **[Boundary]** > **Physics Values** > **Current Voltage Characteristic** node and in the Properties window, set Method to one of the following options:
   - Butler-Volmer (Butler-Volmer Current-Potential Characteristic)
   - Tabular Polarization Curve

Using the Butler-Volmer Method

When using the Butler-Volmer method, you specify the quantities that appear in the Butler-Volmer equation (see Eqn. (3735)) by setting the corresponding properties. See Current Voltage Characteristic Properties.

Using the Tabular Polarization Curve Method

The relationship between the specific electric current, j_n, and the potential drop across the boundary or interface, $\phi_1 - \phi_0$ is specified using tabular data. See Tabular Polarization Curve Properties.

1. Select the **[Region]** > **Boundaries** > **[Boundary]** > **Physics Values** > **Current Voltage Characteristic** > **Tabular Polarization Curve** node and set the appropriate *Tabular Polarization Curve Properties*.
2. Select the **Tabular Polarization Curve** > **Table Sorting** node and specify the *Option*.

   The Table Sorting Option is set to Electric Potential by default, which is appropriate for most applications. However, if you are setting up a plasma simulation with non-monotonic current voltage characteristics, you can select Specific Electric Current. See Table Sorting Properties.

Modeling Magnetic Fields

The following steps outline the recommended workflow for modeling magnetic fields induced by electric currents, excitation coils, and permanent magnets.

In all applications, STAR-CCM+ computes the magnetic vector potential $\mathbf{A}$ induced by an electric current density $\mathbf{J}$. You can prescribe the electric current density $\mathbf{J}$ in the region, or activate models that calculate the relevant contributions to the electric current density in the region (for example, STAR-CCM+ provides models to represent permanent magnets and excitation coils).

After computing $\mathbf{A}$, STAR-CCM+ calculates the associated magnetic field $\mathbf{H}$, the magnetic flux density $\mathbf{B}$, and additional fields such as the stored magnetic energy and the electromagnetic stress tensor.

For a practical example of modeling magnetic fields in STAR-CCM+, refer to the tutorial, Magnetic Vector Potential: Magnetic Flux, which demonstrates how to use the **Finite Element Magnetic Vector Potential** model to compute the magnetic flux density induced by permanent magnets.

Contents:

Selecting the Physics Models and Materials
Setting the Magnetic Permeability
Specifying Boundary Conditions and Electric Current Density Sources
Modeling Permanent Magnets
Modeling Excitation Coils
Modeling Losses
Running the Simulation

Selecting the Physics Models and Materials

As STAR-CCM+ provides several physics models for computing the magnetic vector potential, the following steps include guidelines for choosing the model that is appropriate to your analysis.

STAR-CCM+ offers both a finite element (FE) and a finite volume (FV) approach to solve for the magnetic vector potential. As the FE method imposes specific interface and mesh requirements, choose the discretization strategy before you prepare the required regions, interfaces, and mesh. For more information on the FE mesh and interface requirements, see Finite Element Magnetic Vector Potential Model Reference.

1. Before you set up the physics continua, prepare the required geometry, regions, interfaces, and mesh, as appropriate to your analysis.

   For more information on these general operations, see STAR-CCM+ Simulation Process.
2. Create physics continua and assign them to the relevant regions.
3. In each physics continuum, activate physics models as required. To model magnetic fields, include the following models:

Group Box	Physics Model
Space	• To solve for the magnetic vector potential in a 3D domain, activate the **Three Dimensional** model. • To solve for transverse-electric (TE) or transverse-magnetic (TM) modes in a 2D domain, activate either the **Two-Dimensional** or the **Axisymmetric** model. In the TE configuration, the magnetic field is normal to the 2D domain. The normal component of the magnetic vector potential, and of any external current density, is zero. In the TM configuration, the magnetic field lies in the plane of the 2D domain. The magnetic vector potential and the electric current density have only the vector component normal to the 2D domain. **Note:** The **Axisymmetric** model is only compatible with the FV models. When using the **Axisymmetric** model, make sure that you use an appropriate mesh resolution near the axis.
Material	Choose one of the material models based on requirements. The FE framework supports single-component or multi-component solids, and single-component gases. The FV framework also supports multi-component fluids.
Optional Models	Activate the **Electromagnetism** model.
Electromagnetism	• To solve for transverse-magnetic (TM) modes in a 2D domain, activate one of the following models: ◦ **Transverse Magnetic Potential** (see Transverse Magnetic Potential Reference): Solves for the transverse magnetic potential using the finite volume (FV) approach. (see Transverse Magnetic Potential Reference). ◦ **Harmonic Balance FV Transverse Magnetic Potential** model. Solves for the transverse magnetic potential in the frequency-domain using the finite volume (FV) approach. This solution technique is ideal for modeling fields with harmonic (sinusoidal) time dependence in linear materials. See Harmonic Balance FV Transverse Magnetic Potential Model Reference. • To solve for transverse-electric (TE) modes in a 2D domain, or for the magnetic vector potential in a 3D domain, activate one of the following models: ◦ **Finite Element Magnetic Vector Potential**—ideal for cases with spatial variation in magnetic permeability. This model has specific interface and mesh requirements. See Finite Element Magnetic Vector Potential Model Reference. For one-way coupled electromagnetic-stress analysis, where STAR-CCM+ calculates the displacement of a solid in response to electromagnetic loads, activate this model together with the **Solid Stress** model. See Selecting Solid Stress Physics Models.

Group Box	Physics Model
	◦ **Finite Volume Magnetic Vector Potential**—this solution technique requires less memory, but it is not suitable for cases with spatial variation in magnetic permeability. See Finite Volume Magnetic Vector Potential Model Reference. ◦ **Harmonic Balance FV Magnetic Vector Potential**—specific for fields with harmonic (sinusoidal) time dependence in linear materials. See Harmonic Balance FV Magnetic Vector Potential Model Reference.
Time	• To run a static analysis, where all quantities are time-independent, activate the **Steady** model. • To account for eddy-currents induced by time-varying magnetic fields in conducting media, activate the **Implicit Unsteady** model. • To run a quasi-unsteady analysis, which allows you to define time-dependent quantities, while neglecting eddy-currents, activate the **Implicit Unsteady** model with the **Quasi-Unsteady Losses** model (see Quasi Unsteady Losses Model Reference). This combination of models is appropriate for the physics continua associated with non-conducting materials, in transient simulations.

The source of magnetic field is the electric current density $\mathbf{J}$. You can either prescribe an electric current density, or calculate it in STAR-CCM+ using the following models:

- To account for $\mathbf{J}$ as induced by excitation coils, activate the **Excitation Coil** model. See Modeling Excitation Coils.
- To account for $\mathbf{J}$ as induced by permanent magnets, activate the **Permanent Magnet** model. See Modeling Permanent Magnets.
- To account for $\mathbf{J}$ as induced by differences in electric potential, activate one of the electrodynamic potential models. See Modeling Electric Currents.

4. You can activate additional models from the *Optional Models* group box, based on analysis requirements.

   For example, you can model power losses in electric machines, Ohmic Heating, and the interaction between electrically conducting fluids (such as molten metals and plasma) and the magnetic field. For more information, see Modeling Losses, Modeling Ohmic Heating, and Modeling Magnetohydrodynamics (MHD).
5. For each continuum, select appropriate materials from the STAR-CCM+ material library.

   For more information, see STAR-CCM+ Simulation Process.

Defining the Magnetic Permeability

When modeling magnetic fields, you specify the material magnetic permeability μ, which defines the relationship between the magnetic flux density $\mathbf{B}$ and the magnetic field $\mathbf{H}$.

For more information, see Eqn. (3688).

Note: When the **Harmonic Balance FV Magnetic Vector Potential** model is active, STAR-CCM+ assumes that the magnetic permeability is constant and equal to the vacuum permeability.

To specify μ for a material:

1. Expand the relevant **[physics continuum]** > **Models** > **[Material Model]** > **[Material]** > **Material Properties** node.
2. Select the **Magnetic Permeability** node and set its *Method* based on the type of material that you want to model. The available methods depend on the selected magnetic vector potential model.

Material	Steps
Fluid	1. Set *Method* to either **Constant** or **Field Function**. 2. Specify an appropriate value or field function under the **Magnetic Permeability** node.
Isotropic solid	• For diamagnetic and paramagnetic materials, where the relationship between $\mathbf{B}$ and $\mathbf{H}$ is approximately linear, set *Method* to either **Constant** or **Field Function**, then specify an appropriate value or field function under the **Magnetic Permeability** node. • For ferromagnetic materials, where the magnetization is a nonlinear process, set *Method* to **Table(B,H)**. This method allows you to specify the nonlinear B-H curve using a table of $\mathbf{B}$ and $\mathbf{H}$ values. For further instructions, see Using the Table (B, H) Method.
Anisotropic solid (only available for the **Finite Element Magnetic Vector Potential** model)	1. Set *Method* to **Anisotropic**. This method allows you to define μ as a second-order tensor. You define μ independently in each region associated with the physics continuum. 2. Expand the relevant **[Region]** > **Physics Values** > **Anisotropic Permeability** node. • To model diamagnetic and paramagnetic materials, define μ using any of the common methods for tensor profiles (see Tensor Profiles). • To model ferromagnetic materials: 1. Set *Method* to **Table(B,H) Principal Tensor**. This method allows you to specify up to three different B-H curves (one for each principal direction of a specified coordinate system) using tables of B and H values. 2. Expand the **Anisotropic Permeability** > **Table(B,H) Principal Tensor** node. 3. Specify the coordinate system with respect to which the principal tensor is defined, using the **XX Axis** and **YY Axis** nodes (see Setting Values Using a Principal Tensor). 4. Specify the components of the principal tensor: • To define a component using a nonlinear B-H curve, select the relevant component node (for example, **XX Components**), and set *Method* to **Table (B, H)**. For further instructions, see Using the Table (B, H) Method. • For linear components, set *Method* to **Constant** and specify a constant value in the usual manner.

Contents:

Using the Table (B,H) Method

Using the Table (B,H) Method

The Table (B,H) method allows you to specify a nonlinear B-H curve using a table of $\mathbf{B}$ and $\mathbf{H}$ values.

To specify a table of $\mathbf{B}$ and $\mathbf{H}$ values:

1. Import the B, H table into STAR-CCM+. You can either:
   - import a B, H table as a file table under the **Tools** > **Tables** node
   - import a B, H, μ table into the material database

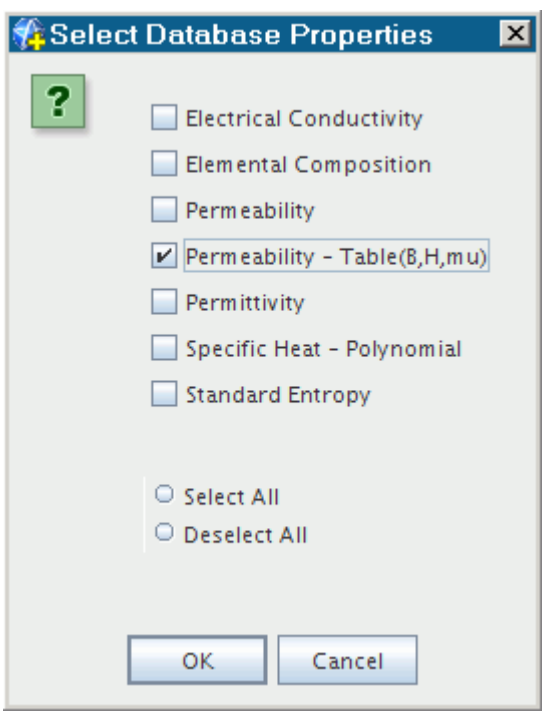
   To import the table into the material database:

   a) Make sure that the column headings are:
      - MagneticField
      - MagneticFluxDensity
      - Permeability

      The heading text is not case-sensitive.

   b) Under **Tools** > **Material Databases**, right-click the database material node and select **New Database Properties...**

   c) In the *Select Database Properties* dialog, activate **Permeability - Table (B, H, mu)**, then click **OK**.



   The new **Permeability** node appears under the material node.

   d) Right-click the **Permeability** > **Table(B,H,mu)** > **Material Database Table** node and select **Import Material Table from a file**.

e) In the *Open* dialog, select the appropriate table file and click **Open**.

The selected table is loaded into the material database.

2. Use the imported table to define μ:

a) Depending on the type of material that you are modeling, select either of the following nodes:

Type of Solid	Table(B,H) Node
Isotropic	**[physics continuum]** > **Models** > **[Solid]** > **[Material]** > **Material Properties** > **Magnetic Permeability** > **Table(B,H)** > **Tabular Data**
Anisotropic	**[Region]** > **Physics Values** > **Anisotropic Permeability** > **Table(B,H) Principal Tensor** > **[Component]** > **Table(B,H)**

b) Specify the *Input table*:

- If you want to use the B-H-mu table from the material database, select **Material Database Table**.
- If you want to use an imported file table, select the relevant table from the list.

c) Specify the columns that represent the values for the magnetic flux density and the magnetic field using the properties, *Table: Magnetic Flux Density* and *Table: Magnetic Field*.

Specifying Boundary Conditions and Electric Current Density Sources

At each boundary, you can either specify the magnetic vector potential $\mathbf{A}$ (Dirichlet boundary condition), or the electric current sheet $\mathbf{J}_S$ (Neumann boundary condition). You can also define electric current density sources on regions. In quasi-static analyses, you can ramp the quantities with appropriate time-dependent functions.

Specify appropriate boundary conditions at each boundary:

1. Expand the relevant **Regions** > **[Region]** > **Boundaries** > **[Boundary]** > **Physics Conditions** node.
2. Select the **Magnetic Vector Potential Specification** node and set *Method* as required.

   For each of the magnetic vector potential models, STAR-CCM+ provides several methods to specify $\mathbf{A}$ and $\mathbf{J}_S$ at a boundary. For information on the methods available to each model, see:

   - FE Magnetic Vector Potential Boundary Inputs
   - FV Magnetic Vector Potential Boundary Inputs
   - FV Harmonic Balance Magnetic Vector Potential Boundary Inputs
   - Transverse Magnetic Potential Boundary Inputs
   - FV Harmonic Balance Transverse Magnetic Potential Boundary Inputs

   When using the harmonic balance models, $\mathbf{A}$ and $\mathbf{J}_S$ are specified as complex profiles.

3. For any region, you can define an electric current density using the *Electric Current Density Source Option*:

   a) Select the relevant **[Region]** > **Physics Conditions** > **Electric Current Density Source Option** node and set *Electric Current Density Source Option* to **Specified**.

   This region input is not available when the associated continuum uses the **Electrodynamic Potential** model or the **Excitation Coil** model, as these models automatically calculate the electric current density in the region.

b) Set the corresponding **[Region]** > **Physics Values** > **Electric Current Density User Source** node:

- In 2D or 3D simulations using either the **Finite Element Magnetic Vector Potential** model or the **Finite Volume Magnetic Vector Potential** model, specify the electric current density as a vector profile. In 2D simulations (TE mode), you only specify the components tangential to the 2D domain. For more information, see FE Magnetic Vector Potential Region Inputs and FV Magnetic Vector Potential Region Inputs.
- In 2D simulations using the **Transverse Magnetic Potential** model (TM mode), specify the electric current density as a scalar profile, which represents the component of the electric current density perpendicular to the 2D region. When the current source is a nonlinear function of magnetic vector potential, specify also the **Electric Current Density Source Potential Derivative**, to improve the convergence of the **Transverse Magnetic Potential** solver. For more information, see TM Region Inputs.
- When using the harmonic balance models, specify the electric current density as a complex profile. See Region Settings and Harmonic Balance FV Transverse Magnetic Potential Model Reference.

The **Finite Element Magnetic Vector Potential** model and the **Transverse Magnetic Potential** model also allow for interface inputs. For more information, see:

- FE Magnetic Vector Potential Interface Inputs
- Transverse Magnetic Potential Interface Inputs

Modeling Permanent Magnets

STAR-CCM+ allows you to model solids and multi-part solid assemblies as permanent magnets.

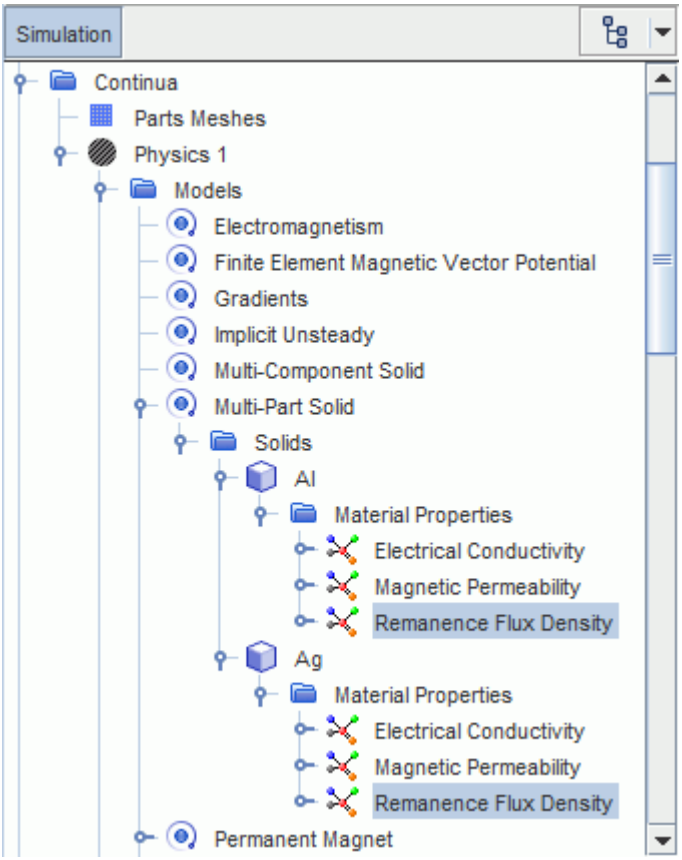
To model permanent magnets:

1. Set up the physics continua as described in Selecting the Physics Models and Materials.
2. Activate the **Permanent Magnet** model in the relevant solid continua.

   For information on the **Permanent Magnet** model, see Permanent Magnet Model Reference.

For each solid continuum using the **Permanent Magnet** model, specify the remanence flux density, that is, the magnetic flux density that exists in a solid material in absence of external magnetic fields:

3. Expand the relevant **[solid continuum]** > **Models** > **Solid** > **Material Properties** node.
4. Select the **Remanence Flux Density** node and specify the remanence flux density as a scalar, with either the **Constant** or **Field Function** method. If you are using the **Multi-Part Solid** model, specify the remanence flux density of each material of the assembly.



For solid materials, you can also store the remanence flux density as a database property as follows:

5. Expand the **Tools** > **Material Databases** > **[Material Database]** node.
6. Right-click the relevant material node and select **New Database Properties...**.
7. Select either **Remanence Flux Density** or **Remanence Flux Density - Polynomial**, click **OK**, then set the new property as appropriate.

After setting the remanence flux density, define the magnetization direction, that is, the direction of the remanence flux density of the permanent magnets. You can specify this direction on the physics continuum, so that it applies to all regions associated with the continuum, or you can specify the magnetization direction independently for each region.

To set the magnetization direction on the physics continuum:

8. Select the **[solid continuum]** > **Models** > **Permanent Magnet** > **Magnetization Direction** node and specify the magnetization direction as a unit vector.

To specify a different magnetization direction on specific regions:

9. Expand the relevant **Regions** > **[solid region]** > **Physics Conditions** node.
10. Select the **Magnetization Direction** node and set *Method* to **Specify Region Values**.
11. Select the **Physics Values** > **Magnetization Direction** node and specify the magnetization direction for the region.

Modeling Excitation Coils

STAR-CCM+ allows you to model solid regions as excitation coils.

The **Excitation Coil** model computes the electric current density $\mathbf{J}_{\mathrm{coil}}$ produced by an excitation coil (see Eqn. (3759)) and calculates the specific magnetic flux linkage according to Eqn. (3763). The magnetic vector potential solvers use the computed $\mathbf{J}_{\mathrm{coil}}$ as electric current density source for the solid region.

In transient simulations, the **Excitation Coil** model also computes the specific electromotive force.

To model an excitation coil:

1. Set up the physics continua as described in Selecting the Physics Models and Materials.
2. Activate the **Excitation Coil** model in the relevant solid continuum.

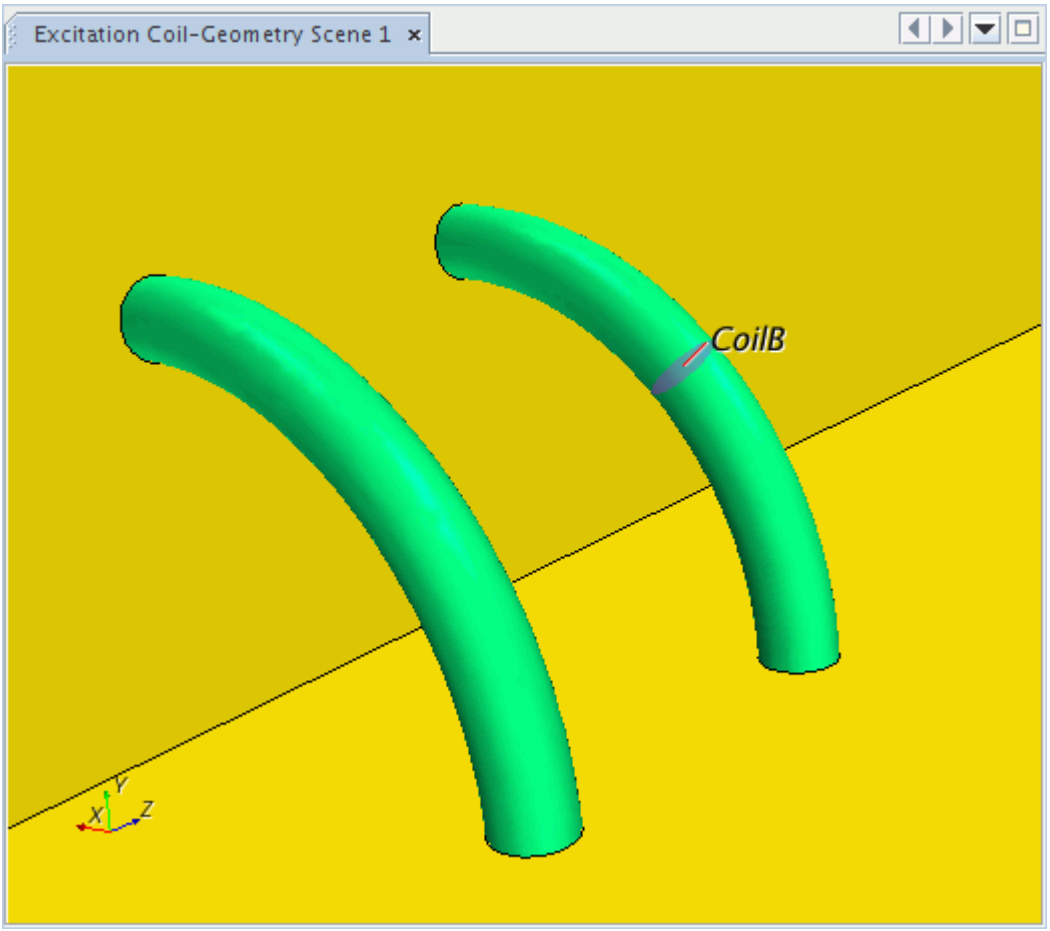
   For more information on the **Excitation Coil** model, see Excitation Coil Model Reference.
3. Make sure that the solid region representing the coil is a closed loop. If the region is not closed, make sure that the open ends of the coil region extend outside the simulation domain.

   If an excitation coil region has ends that lie within the simulation domain, the divergence of the electric current density field is non-zero, and therefore the magnetic field cannot be calculated.

Specify the coil settings as described below. For details on the available settings, see Region Settings.

4. Expand the relevant **Regions** > **[coil region]** > **Physics Values** node.
5. Specify the parameters that are used to calculate the magnitude of the coil electric current density:
   a) Select the **Electric Current Density Magnitude** > **Ampere Turn** node and specify the *Number of Turns* in the coil and the *Number of Sections* through the coil. See Electric Current Density Magnitude.
   b) Select the **Ampere Turn** > **Electric Current** node and specify the electric current that goes through an individual wire.
   c) Set the excitation coil reference cross section or height:
      - In 3D simulations that use either the **Finite Volume Magnetic Vector Potential** or the **Finite Element Magnetic Vector Potential** model, select the **Excitation Coil Reference Cross Section** node and set *Reference Part* to an appropriate input part, such as an internal interface, or a section through the coil region. For more information, see Excitation Coil Reference Cross Section.

        The area a_{ref} of the excitation coil reference cross section is annotated in the following scene. Here, the number of sections n_{a} cutting through the coil region is 1.



- In 2D simulations that use the **Transverse Magnetic Potential** model, select the **Excitation Coil Reference Height** node and set the *Reference Height*. For more information, see Excitation Coil Reference Height.

6. In transient simulations, select the **Electric Current Density Magnitude** > **Ampere Turn** > **Conductor Area** node and specify the area that is covered by conducting material as an absolute or relative value.

   For example, you specify an absolute value according to a certain wire gauge, or a relative value as a percentage of the coil area. STAR-CCM+ uses this value to calculate the direct current resistance of the excitation coil (see Excitation Coil Conductor Resistance).

7. To specify the direction of the electric current density, select the **Local direction Field** node and:
   - In 3D simulations that use either the **Finite Volume Magnetic Vector Potential** or the **Finite Element Magnetic Vector Potential** model:
     1. Set *Method* as appropriate.
     2. When using the **Feature Curve** method, use the *Curve Parts* property to specify the feature curve that represents the local current density field orientation. To correctly represent the local field orientation, select a closed loop or a curve that extends to the ends of the simulation domain.
     3. Set the field Direction to either **Backward** or **Forward**, as appropriate. When defining the local direction field using a feature curve, the field direction is specified relative to the curve orientation.

- In 2D simulations that use the **Transverse Magnetic Potential** model, set *Direction* to either **Forward** or **Backward**, as appropriate.

You can calculate the magnetic flux linkage in the coil (see Eqn. (3760)) by creating magnetic flux linkage reports. The procedure for creating a magnetic flux linkage report depends on the magnetic vector potential model in use.

8. Depending on the magnetic vector potential model in use, create a magnetic flux linkage report in one of the following ways:

Magnetic Vector Potential Model	Steps
Finite Element Magnetic Vector Potential	1. In the physics continuum associated with the coil region, activate the **Excitation Coil Lumped Parameter** model. 2. Right-click the **Reports** node and select **New Report** > **Magnetic Flux Linkage**. 3. Select the **Reports** > **[Magnetic Flux Linkage]** node and set *Parts* to the relevant coil regions. 4. Set the remaining properties as required.
Finite Volume Magnetic Vector Potential **Transverse Magnetic Potential**	1. Right-click the **Reports** node and select **New Report** > **Volume Integral**. 2. Select the **Reports** > **[Volume Integral]** node and set *Parts* to the relevant coil regions. 3. Set *Field Function* to **Specific Magnetic Flux Linkage**. 4. Set the remaining properties as required.

Contents:

Extracting the Excitation Coil Lumped Parameters
Coupling Coil Regions with Coil Circuit Elements

Extracting the Excitation Coil Lumped Parameters

In simulations that use either the **Finite Element Magnetic Vector Potential** model or the **Transverse Magnetic Potential** model, STAR-CCM+ allows you to extract the lumped parameters (that is, resistance and inductance) of the excitation coil region. These parameters can be used to calculate the magnetic inductance of the coil region or to couple an excitation coil region with an excitation coil circuit element.

To extract the lumped parameters of an excitation coil region:

1. Activate the **Excitation Coil Lumped Parameter** model in the physics continuum associated with the coil region.

You can calculate the magnetic inductance by creating magnetic inductance reports. The procedure for creating a magnetic inductance report depends on the magnetic vector potential model in use.

2. Depending on the magnetic vector potential model in use, create a magnetic inductance report in one of the following ways:

Magnetic Vector Potential Model	Steps
Finite Element Magnetic Vector Potential	When using the **Finite Element Magnetic Vector Potential** model, STAR-CCM+ provides the **Magnetic Inductance** report. This type of report calculates either the self-inductance of a coil region, or the mutual inductance between different coil regions: 1. Right-click the **Reports** node and select **New Report** > **Magnetic Inductance**. 2. Select the **Reports** > **[Magnetic Inductance]** node and select the relevant coil region using the *Parts* property. • To calculate the self-inductance, leave the *Parts Mutual* property blank. If you select multiple coil regions under the *Parts* property, STAR-CCM+ calculates the effective self-inductance of the specified group of regions. • If you select a coil region, or multiple coil regions, using the *Parts Mutual* property, STAR-CCM+ calculates the mutual inductance between the group of coil regions selected under *Parts*, and the group of coil regions selected under *Parts Mutual*.
Transverse Magnetic Potential	When using the **Transverse Magnetic Potential** model, STAR-CCM+ allows you to calculate the self-inductance of a coil region, or the mutual inductance between two coil regions: 1. Right-click the **Reports** node and select **New Report** > **Volume Integral**. 2. Select the **Reports** > **[Volume Integral]** node and set *Parts* to the relevant coil region, for example, **Region 1**. • To calculate the self-inductance of **Region 1**, set *Field Function* to **Excitation Coil Flux Linkage** > **Region 1 Specific Magnetic Flux Linkage**. **Note:** This field function is activated by the **Excitation Coil Lumped Parameter** model and is different from the **Specific Magnetic Flux Linkage** field function activated by the **Excitation Coil** model. See Excitation Coil Lumped Parameter Model Reference. • To calculate the mutual inductance between **Region 1** and another region, for example, **Region 2**, set *Field Function* to the specific magnetic flux linkage associated with **Region 2**, that is, **Excitation Coil Flux Linkage** > **Region 2 Specific Magnetic Flux Linkage**. 3. Set the remaining properties as required.

For information on how to couple the excitation coil region with an excitation coil circuit element, see Coupling Coil Regions with Coil Circuit Elements.

Coupling Coil Regions with Coil Circuit Elements

STAR-CCM+ allows you to couple 2D or 3D excitation coil regions with the excitation coil elements of a 1D electric circuit.

To couple an excitation coil region with an excitation coil circuit element:

1. Define the excitation coil region and physics continuum as described in the previous sections.
2. In the physics continuum associated with the coil region, select **Circuit Model** from the *Optional Models* group box.
   - When the physics continuum contains either the **Finite Element Magnetic Vector Potential** model or the **Transverse Magnetic Potential**, also select the **Excitation Coil Lumped Parameter** model, which extracts the lumped parameters (that is, resistance and inductance) of the excitation coil region (see Extracting the Excitation Coil Lumped Parameters. The extracted lumped parameters are used to define the circuit element load (see Circuit Element Properties).
3. Define the electric circuit as appropriate.

   For more information, see Defining Electrical Circuits.

To add an excitation coil element to the circuit:

4. Right-click the **Circuits** > **[Circuit]** > **Circuit Elements** node, select **New Circuit Elements** and then one of the following options, based on the magnetic vector potential model in use:

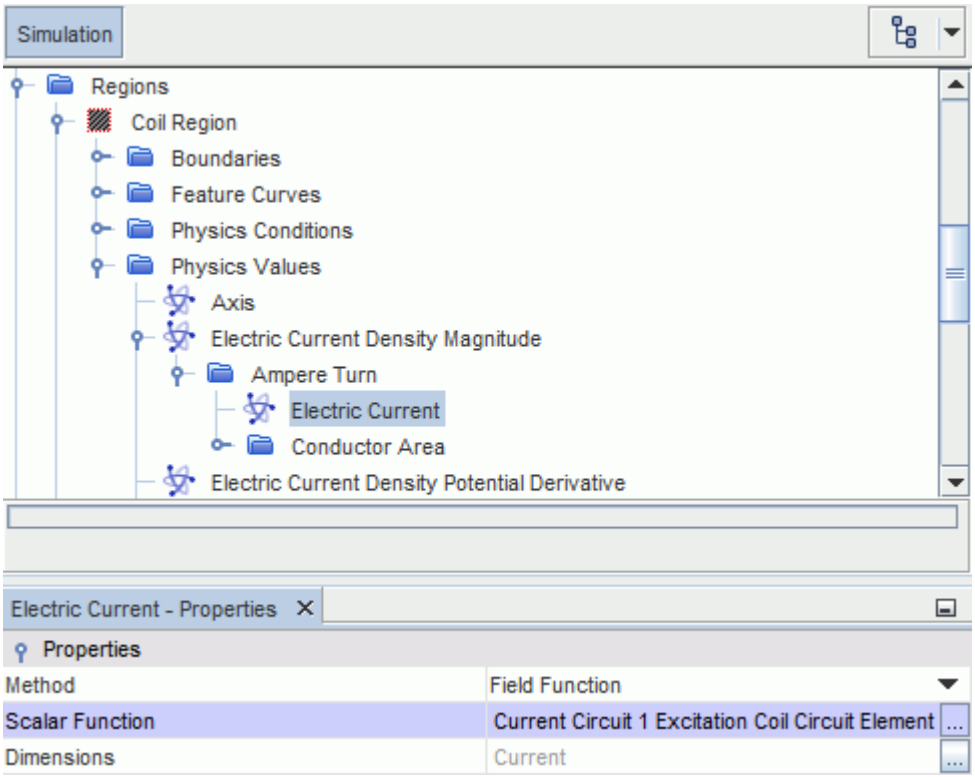
Magnetic Vector Potential Model	Circuit Element
Finite Element Magnetic Vector Potential	**Excitation Coil Circuit Element (Fe)**
Finite Volume Magnetic Vector Potential **Transverse Magnetic Potential**	**Excitation Coil Circuit Element**

   STAR-CCM+ adds the selected circuit element under the **Circuit Elements** node.

To associate the excitation coil circuit element with the corresponding excitation coil region:

5. Select the **Circuit Elements** > **[Excitation Coil Circuit Element]** node and set *Regions* to the relevant coil region.

When you associate an excitation coil region with a circuit element, STAR-CCM+ automatically sets the electric current in the coil region to the electric current computed by the circuit solver.



- If you wish to specify a different current for the coil region, change the settings for the relevant **Electric Current Density Magnitude** > **Ampere Turn** > **Electric Current** node, as desired.

6. Select the **Circuit Elements** > **[Excitation Coil Circuit Element]** node and set *Load Type* to one of the following, based on the magnetic vector potential model in use:

Magnetic Vector Potential Model	Circuit Element Load Type
Finite Element Magnetic Vector Potential **Transverse Magnetic Potential**	**Compound**
Finite Volume Magnetic Vector Potential	**Voltage**

For more information on all the available properties, see Circuit Element Properties.

Modeling Losses

STAR-CCM+ allows you to model hysteresis and eddy-current losses in electric machines.

The Modified Steinmetz model calculates the losses using a modified Steinmetz formula that is based on magnetic flux density calculations (see Eqn. (3766)). The loss terms can be used in thermal analyses.

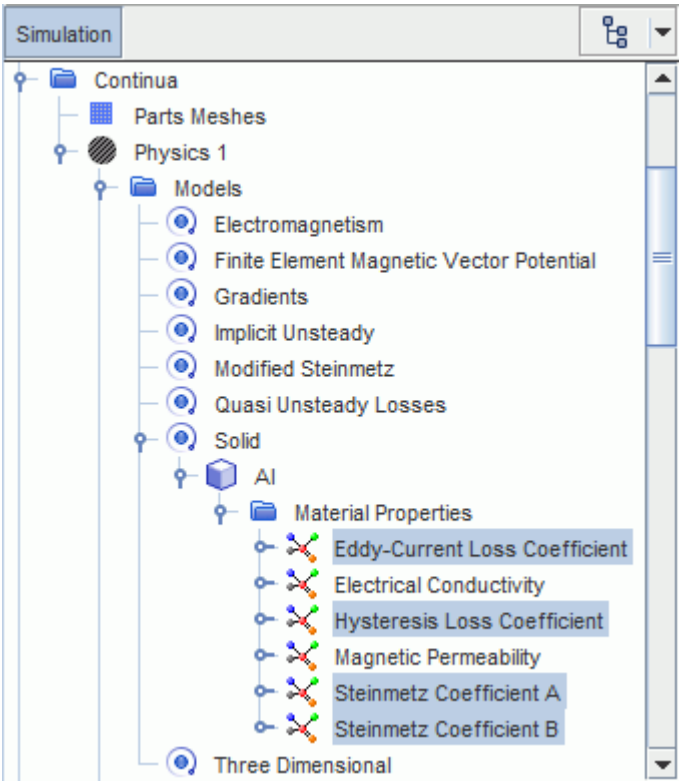
To model hysteresis and eddy-current losses:

1. Set up the physics continua as described in Selecting the Physics Models and Materials.
2. In the relevant solid continua, add the following models:

Group box	Model
Optional Models	**Quasi Unsteady Losses** (neglects the eddy currents in the magnetic vector potential equation)
Optional Models	**Modified Steinmetz**

These models are available for solid physics continua that use the **Implicit Unsteady** model. See Modified Steinmetz Model Reference.

3. Specify the coefficients that appear in the modified Steinmetz formula using the corresponding properties under the solid material model.



You can add a **Modified Steinmetz** monitor, that monitors the peak value and the average time derivative of a specified field function:

4. Right-click the **Monitors** node and select **New Monitor** > **Modified Steinmetz**.
5. Select the monitor and specify the *Field Function* for which you wish to monitor the peak value and average time derivative. The default field function is the magnetic flux density.

   The monitored quantities become available as field functions. See Modified Steinmetz Model Reference.

Running the Simulation

Prepare analysis objects, set up the solvers and run the simulation.

1. Set up scalar scenes, reports, and plots to visualize the solution and monitor the quantities that are relevant to your analysis.

See:

- FE Magnetic Vector Potential Field Functions
- FV Magnetic Vector Potential Field Functions
- Transverse Magnetic Potential Field Functions

By default, the solution is available at cell centroids for all models. Even though the **Finite Element Magnetic Vector Potential** model computes the magnetic vector potential at element edges (see H-Curl Hierarchical Shape Functions), STAR-CCM+ internally interpolates the solution at cell centroids.

2. Specify stopping criteria as required.
3. Expand the **Solvers** node and set up the solver properties, based on analysis requirements:
   - **Finite Volume Magnetic Vector Potential**:
     1. Specify the **Magnetic Vector Potential** and the **AMG Linear Solver** properties. The available solver options are described in the section, Magnetic Vector Potential Solver.
     2. For the **Magnetic Vector Potential** solver, set **Formulation** to either **Non-Conservative** or **Conservative** as required. If you use the **Non-Conservative** formulation, activate the **Magnetic Vector Potential Projection** model on the associated physics continua.
   - **Finite Element Magnetic Vector Potential**:
     1. Specify the **Magnetic Vector Potential** and the **Sparse Direct Solver** properties. The available solver options are described in the sections, FE Magnetic Vector Potential Solver, and Sparse Direct Solver Reference. For large cases, run the **Sparse Direct Solver** in OOC (Out-of-Core) mode, and specify a directory on your system disk that the solver can use for data storage.
4. Run the simulation using a double-precision version of STAR-CCM+.

Modeling Ohmic Heating

In heat transfer analyses involving electrically conducting materials, you can account for the energy dissipation due to the flow of electric currents.

The energy dissipation in conducting materials is proportional to the electric current density and the electric field (see Eqn. (3774)). If this term is significant compared to all other source terms in the energy equation, activate the **Ohmic Heating** model and define the dependency of the material electrical conductivity on temperature.

1. Prepare the electromagnetic analysis following the guidelines outlined in the previous sections, Modeling Electric Currents and Modeling Magnetic Fields. In the physics continua associated with electrically conducting regions, activate the required potential models.
2. Complete the electromagnetic analysis set up, by specifying appropriate boundary, interface, and region settings. For guidelines, see Modeling Electric Currents and Modeling Magnetic Fields.

   If you define an electric resistance at a boundary or interface, STAR-CCM+ also accounts for the energy dissipation due to the specified resistance (see Eqn. (3777)).
3. Set up the simulation for heat transfer analysis.

   This step includes activating an energy model and defining thermal settings on the relevant boundaries and regions. For more information, see Modeling Heat Transfer.
4. Activate the **Ohmic Heating** model in the physics continua that are assigned to the electrically conducting regions.

   For more information on the model dependencies and associated field functions, see Ohmic Heating Model Reference.

In general, the electrical conductivity of a material is a function of temperature and, therefore, it has a high impact on the accuracy of the heat calculation. When the temperature change is relatively small, you can neglect the dependency of the material electrical conductivity on temperature. When the temperature change is significant, define the electrical conductivity as a function of temperature:

5. In all physics continua that use the **Ohmic Heating** model, expand the relevant **[Physics Continuum]** > **Models** > **[Material Model]** > **Material Properties** node.
6. Select the **Electrical Conductivity** node and choose one of the following options:

Option	Steps
To define σ as a polynomial function of temperature	1. Select **Polynomial in T**. 2. Define the polynomial function using the **Electrical Conductivity** > **Polynomial in T** node. See Using Polynomial in T.
To define σ using a table of σ, T values	1. Select **Table (T)**. 2. Define the input table using the **Electrical Conductivity** > **Table (T)** node. See Using Table(T).
To define σ using a user-defined function of temperature	1. Select **Field Function**.

	2. Under the **Tools** > **Field Function** node, create an appropriate temperature-dependent field function (see Creating a User Field Function). 3. Finally, select the **Electrical Conductivity** > **Field Function** node and set *Scalar Function* to the user field function that you created.
To define an anisotropic σ	1. Select **Anisotropic**. 2. In the relevant regions, select the **Anisotropic Electrical Conductivity** node and define the components of the second-order tensor σ as a function of temperature. See Tensor Profile Methods .

7. Prepare the required scenes and plots for post-processing and run the simulation.

The specific Ohmic heat source (Eqn. (3774)) can be significantly large at the beginning of the solution process, causing the energy solution to diverge.

8. If the energy solution is diverging during initial iterations, you can:
   - Examine the critical areas where the ohmic heat source term is large, using the **Specific Ohmic Heat Source** field function. Evaluate whether the mesh quality can be improved in these areas.
   - In the relevant physics continuum **Reference Values**, reduce the maximum allowable temperature.
   - Decouple the thermal and the electromagnetic analyses. That is, freeze the energy solver and get a converged solution for the potentials then activate the energy solver to continue with the thermal analysis.

Modeling Magnetohydrodynamics (MHD)

In electromagnetic applications involving electrically conducting fluids, such as molten metals, electrolytes, and plasmas, STAR-CCM+ allows you to account for the interaction between the conducting fluid and the magnetic field.

A conducting fluid in relative motion to a magnetic field induces an electric current density $\mathbf{J}_L$ (see Eqn. (3778)), which causes the fluid to experience an additional body force, known as the Lorentz force (see Eqn. (3784)). The MHD models in STAR-CCM+ account for these effects in both the fluid flow solution and the electric potential solution (see Magnetohydrodynamics (MHD)).

To define the magnetic field, you can either prescribe the magnetic flux density within the fluid region, or calculate the magnetic flux density using one of the STAR-CCM+ magnetic vector potential models. The first approach is suitable for analyses in which the magnetic flux density induced by the electric current density $\mathbf{J}_L$ can be neglected. For more information, see One-Way Coupled and Two-Way Coupled MHD.

1. Before you set up the physics continua, prepare the required geometry, regions, interfaces, and mesh, as appropriate to your analysis.

   For information on these general operations, see STAR-CCM+ Simulation Process.
2. Create physics continua and assign them to the relevant regions.
3. In each physics continuum, activate physics models as required. To model the interaction between the conducting fluid and the magnetic field, include the following physics models:

Group Box	Physics Model
Space	Either **Three-Dimensional** or **Axisymmetric**
Time	Either **Steady** or **Implicit Unsteady**
Material	Either **Gas**, **Liquid**, **Eulerian Multiphase**, **Multi-Component Gas**, or **Multi-Component Liquid**
Flow	Any
Other Models	**Electromagnetism**
Electromagnetism	• To model the interaction between the conducting fluid and a prescribed magnetic flux density, activate the following models: 1. **Electrodynamic Potential** 2. **One-Way Coupled MHD** • To model the interaction between the conducting fluid and the total magnetic flux density, which is calculated by STAR-CCM+, activate the following models: 1. Either **Finite Volume Magnetic Vector Potential**, **Finite Element Magnetic Vector Potential**, **Transverse Magnetic Potential**, **Harmonic Balance FV Magnetic Vector Potential**, or **Harmonic Balance FV Transverse Magnetic Potential** 2. **Two-Way Coupled MHD** 3. You can also activate the following optional models:

Group Box	Physics Model
	• **Quasi Unsteady Losses**—neglects the contribution to the electric current density coming from eddy currents. • **Electrodynamic Potential**—solves for the electric potential, which also contributes to the total electric current density.

For more information on the MDH models, including the associated field functions and region settings, see One-Way Coupled MHD Model Referenceand Two-Way Coupled MHD Model Reference.

When using the **One-Way Coupled MHD** model, define the magnetic flux density within the relevant regions:

4. Expand the **Regions** > **[Region]** > **Physics Values** node.
5. Select the **Prescribed Magnetic Flux Density** node and specify the magnetic flux density vector profile, as appropriate.
6. Complete the simulation set up as required. This step includes defining the electrical conductivity of the fluid and, when modeling two-way coupled MHD, the magnetic permeability.

   For guidelines on modeling electric currents and magnetic fields, see Modeling Electric Currents and Modeling Magnetic Fields.

Physics Models Reference

Contents:

Electrostatic Potential Model Reference
Electrodynamic Potential Models Reference
Magnetic Vector Potential Reference
Ohmic Heating Model Reference
Magnetohydrodynamic Field Functions Reference
Two-Way Coupled MHD Model Reference

Electrostatic Potential Model Reference

The **Electrostatic Potential** model allows you to calculate the electric field induced by a charge distribution.

STAR-CCM+ calculates the electric field from the electric potential, which is computed from Eqn. (3724).

Model Name	**Electrostatic Potential**	
Theory	See Electrostatic Potential Model.	
Provided By	**[physics continuum]** > **Models** > *Electromagnetism*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Electrostatic Potential**	
Requires	Physics Models: *Space*: one of **Three Dimensional**, **Two Dimensional**, **Axisymmetric** *Time*: one of **Steady**, **Implicit Unsteady** *Material*: one of **Gas**, **Liquid**, **Solid**, **Multi-Component Gas**, **Multi-Component Liquid**, **Multi-Part Solid**, **Eulerian Multiphase** *Optional Models*: **Electromagnetism**	
Activates	Material Properties	**Permittivity**. See Material Properties.
	Initial Conditions	**Electric Potential**. See Initial Conditions.
	Boundary Inputs	**Electric Potential Specification**. See Boundary Settings.
	Region Inputs	**Electric Charge Density Source**. See Region Settings.
	Interface Inputs	**Surface Electric Charge Option**. See Interface Settings.
	Solvers	**Electric Potential**. See Electric Potential Solver Reference.
	Monitors	**Electric Potential**

	Field Functions	**Electric Flux Density**, **Permittivity**, **Displacement Electric Current Density**. See Field Functions.

Material Properties

Permittivity

Specifies the permittivity ε of the material (see Permittivity and Permeability).

Method	Corresponding Method Node
Constant, Field Function Suitable for linear isotropic materials. For multi-component fluids, you specify the effective permittivity of the mixture (and not the permittivity of each component). Available for fluids and solids.	**Permittivity > Constant, Field Function** Specifies ε as a scalar profile.
Anisotropic Suitable for linear anisotropic materials. Available for solids.	**[Region] > Physics Values > Anisotropic Permittivity** Specifies ε for the region as a second-order tensor (see Eqn. (3689) and Eqn. (3690)). For information on how to define second-order tensors, see Tensor Profile Methods .

Initial Conditions

Electric Potential

Allows you to initialize the electric potential ϕ to a specified scalar profile.

Boundary Settings

Applies to all boundary types other than **Overset Mesh** and **Symmetry Plane**.

Electric Potential Specification

At the domain boundaries, allows you to prescribe either the electric potential ϕ or its gradient $\nabla\phi$. See Boundary and Interface Conditions.

Method	Corresponding Physics Value Nodes
Electric Potential	**Electric Potential** Sets the electric potential ϕ at the boundary to a specified scalar profile.
Electric Flux	**Electric Flux** Sets the electric flux through the boundary to a specified scalar profile. See Eqn. (3726).

Specific Electric Flux	**Specific Electric Flux** Sets the electric flux per unit area to a specified scalar profile. See Eqn. (3725).
Electric Flux Density	**Electric Flux Density** Sets the electric flux density $\mathbf{D}$ to a specified vector profile. See Eqn. (3687).

Region Settings

Applies to fluid and solid regions.

Electric Charge Density Source

Specifies the electric charge per unit volume within the region.

Method	Corresponding Physics Value Nodes
Deactivated Sets the region electric charge density to zero, $\rho = 0$.	None
Activated Allows you to specify the region electric charge density.	**Electric Charge Density User Source** Sets the electric charge density ρ to a specified scalar profile. **Electric Charge Density Source Potential Derivative** Specifies the derivative of the electric charge density ρ with respect to the electric potential ϕ. By default, this value is set to zero. In nonliner cases, specifying the dependency of ρ on the potential improves stability.

Interface Settings

Solid/Solid or Fluid/Solid Contact Interface

Surface Electric Charge Option

At an interface, allows you to specify the electric charge per unit area.

Method	Corresponding Physics Value Nodes
None	None. The surface electric charge is zero.
Surface Charge Density	**Surface Charge Density** Sets the surface electric charge to a specified scalar profile.

Fluid/Fluid Fully-Developed Interface

Electric Flux Source Option

Allows you to specify the total or specific electric flux through the interface.

Method	Corresponding Physics Value Nodes
Specific Electric Flux	**Specific Electric Flux** Sets the specific electric flux through the interface to a specified scalar profile. See Eqn. (3725).
Electric Flux	**Electric Flux** Sets the total electric flux through the interface to a specified scalar profile. See Eqn. (3726).

Field Functions

Electric Flux Density

Corresponds to the electric flux density $\mathbf{D}$ in Eqn. (3687).

Permittivity

Corresponds to the material permittivity ε in Eqn. (3687).

Displacement Electric Current Density

In dielectric media, corresponds to the time rate of change of the electric flux density (that is, $\partial \mathbf{D} / \partial t$) due to the motion of dielectric dipoles under the influence of the electric field. Only available for unsteady (transient) simulations.

Contents:

Electric Potential Solver Reference

Electric Potential Solver Reference

The **Electric Potential** solver controls the solution of the electric potential in electrostatic and electrodynamic simulations.

Electric Potential Solver Properties

Under-Relaxation Factor

During the iterative process, the **Electric Potential** solver under-relaxes the changes in the solution, as:

$$\phi^{n+1} = \phi^{n} + \omega\phi'$$

where ω is the under-relaxation factor. At each iteration, the under-relaxation factor governs the extent to which the new solution replaces the old one.

The default value is 1.0. If residuals show solution divergence and do not decrease, reduce the under-relaxation factor.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Temporary Storage Retained

When activated, STAR-CCM+ retains the electric potential residuals computed at each iteration and makes them available as field functions:

- **Electric Potential Correction AP Coefficient**
- **Electric Potential Correction Residual**
- **Electric Potential Gradient**
- **Electric Potential Recon**

Electrodynamic Potential Models Reference

Contents:

Electrodynamic Potential Model Reference
Harmonic Balance FV Electrodynamic Potential Model Reference

Electrodynamic Potential Model Reference

The **Electrodynamic Potential** model allows you to model electric currents in conducting materials.

This model calculates the electric potential according to Eqn. (3727). From the electric potential, STAR-CCM+ calculates the electric field Eqn. (3697) and the electric current density Eqn. (3694). This model can be coupled together with the magnetic vector potential models, which include the magnetic vector potential term in Eqn. (3727).

Model Name	**Electrodynamic Potential**	
Theory	See Potential Formulation of the Governing Equations and Electrodynamic Potential Models.	
Provided By	**[physics continuum]** > **Models** > *Electromagnetism*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Electrodynamic Potential**	
Requires	Physics Models: *Space*: one of **Three Dimensional**, **Shell Three Dimensional**, **Two Dimensional**, **Axisymmetric** *Time*: one of **Steady**, **Implicit Unsteady** *Material*: one of **Gas**, **Liquid**, **Solid**, **Multi-Component Gas**, **Multi-Component Liquid**, **Multi-Part Solid**, **Eulerian Multiphase** *Optional Models*: **Electromagnetism**	
Activates	Material Properties	**Electrical Conductivity**. See Material Properties.

	Initial Conditions	**Electric Potential**. See Initial Conditions.
	Boundary Inputs	**Electric Potential Specification**. See Boundary Settings.
	Region Inputs	**Electrical Conductivity Constraint Option** (for fluid and porous regions) **Effective Porous Electrical Conductivity Option** (for porous regions) **Electromotive Force Source Option** **Transfer Current Density Source Option** See Region Settings.
	Interface Inputs	**Contact Electric Potential Specification** **Electric Current Source Option** **Baffle Electrical Option** See Interface Settings.
	Solvers	**Electric Potential**. See Electric Potential Solver Reference.
	Monitors	**Electric Potential**
	Field Functions	See Field Functions.

Material Properties

Electrical Conductivity

In fluid and solid materials, specifies the electrical conductivity σ (see Electrical Conductivity: Generalized Ohm's Law). In porous materials, specifies the electrical conductivity of the fluid component (σ_{fluid}).

The available methods for defining the electrical conductivity depend on the physics models that you activate in the physics continuum.

For heat transfer analysis, STAR-CCM+ provides specific methods for defining σ as a function of temperature.

Method	Corresponding Physics Value Nodes
Constant, Field Function Available for fluids and solids.	**Electrical Conductivity > Constant, Field Function** Specifies σ as a scalar profile.
Anisotropic Available for solids.	**[Region] > Physics Values > Anisotropic Electrical Conductivity** Specifies σ for the region as a second-order tensor (3x3 matrix in 3D simulations, or 2x2 matrix in 2D simulations). For information on how to define second-order tensors, see Tensor Profile Methods .
Mass-Weighted Mixture Available for multi-component gases. Calculates σ by mass-weighting the	None

electrical conductivity of each gas component. See Using the Mass-Weighted Mixture.	
Polynomial in T Available for fluids and solids, when you activate an energy model in the physics continuum.	**Electrical Conductivity > Polynomial in T** Specifies σ as a polynomial function of temperature, for heat transfer analysis. See Using Polynomial in T.
Table (T) Available for fluids and solids, when you activate an energy model in the physics continuum.	**Electrical Conductivity > Table (T)** Allows you to define σ as a function of temperature by providing a table of σ, T values, from which STAR-CCM+ determines the profile of $\sigma(T)$. See Using Table(T). The temperature range in the table must be consistent with the **Minimum/Maximum Allowable Temperature** settings that are specified under the **Reference Values** node for the physics continuum. This requirement is specific to electrical conductivity.
Table (T,P) Available for compressible gases, when you activate an energy model in the physics continuum.	**Electrical Conductivity > Table (T,P)** Allows you to define σ as a function of temperature and pressure by providing a table of σ, T, p values, from which STAR-CCM+ determines the profile of $\sigma(T, p)$. See Using Table (T,P).
Tabular Available for fluids and solids, when you activate an energy model in the physics continuum.	**Electrical Conductivity > Tabular** Specifies σ as a function of temperature using a table interpolation method. When using this method, specify the independent variables (including temperature) that are used to interpolate the electrical conductivity. See Using Tabular Data.
Table (T,c) Available for liquid materials from the *Electrolyte Material Database*, when you activate either the **Li-Ion Electric Potential** or **Li-Ion Concentration** model.	**Electrical Conductivity > Table (T,c)** Allows you to define the electrical conductivity of the electrolyte as a function of temperature and Li-ion/salt concentration by providing a table of σ, T, c values, from which STAR-CCM+ determines the profile of $\sigma(T, c)$.
Electrochemical Species Available for fluids and multi-component fluids, when you activate the **Electrochemical Species** model. Calculates the electrical conductivity using Eqn. (1110).	None

Initial Conditions

Electric Potential

Allows you to initialize the electric potential ϕ to a specified scalar profile.

Boundary Settings

Applies to all boundary types other than **Overset Mesh** and **Symmetry Plane**.

Electric Potential Specification

At the domain boundaries, allows you to prescribe either the electric potential, the electric current, or the electric current density. See Boundary and Interface Conditions.

Method	Corresponding Physics Value Nodes
Specific Electric Current Allows you to define the electric current density normal to the boundary.	**Specific Electric Current** Sets the specific electric current J_n to a specified scalar profile. See Eqn. (3733). For stabilization, you can optionally specify the derivative of J_n with respect to the electric potential ϕ using the **Specific Electric Current Derivative Option**. See Specific Electric Current Derivative Option.
Electric Potential Allows you to define the electric potential at the boundary according to Eqn. (3729).	**Electric Potential** Defines the potential $\overline{\phi}$ in Eqn. (3729) as a scalar profile. When leaving the **Electrical Resistance** to the default value of zero, the potential at the boundary is simply $\phi = \overline{\phi}$. **Electrical Resistance** Defines an additional source of electrical resistance (R^{ex} in Eqn. (3729)) as a scalar profile. **Electrical Resistance Potential Derivative** For stabilization, allows you to specify the dependence of R^{ex} on ϕ by defining $dR^{\mathrm{ex}}/d\phi$.
Electric Current Allows you to define the total electric current through the boundary.	**Electric Current** Sets the total electric current I_Γ to a specified scalar profile. See Eqn. (3732). When using this option, specify the spatial distribution of electric current using the **Electric Current Distribution** condition. See Electric Current Distribution.
Electric Current Density Allows you to define the electric current density at the boundary.	**Electric Current Density** Sets $\mathbf{J}$ to a specified vector profile. See Eqn. (3694).

Electric Current Potential Characteristic Allows you to define the I-V curve at the boundary.	**Current Voltage Characteristic** Defines the relationship between the electric current flowing through the boundary and the electric potential at the boundary. In electrochemistry applications, you typically define this relationship using the Butler-Volmer method (see Butler-Volmer Current-Potential Characteristic). Alternatively, you can specify the I-V curve by providing tabular data. For more information, see Setting the Electric Current Potential Characteristic and Current-Voltage Characteristic Reference. **Electric Potential** Defines the potential ϕ_1 in Eqn. (3734) as a scalar profile. **Electrical Resistance** Defines the electrical resistance R in Eqn. (3734) as a scalar profile.
Insulator Sets the total electric current through the boundary (Eqn. (3732)) to zero.	None

Electric Current Distribution

When defining the electric current at the boundary, this setting specifies the electric current spatial distribution. The available options are:

Uniform Electric Potential—calculates the specific electric current spatial distribution so that the electric potential is constant along the boundary.
Uniform Specific Electric Current—prescribes a constant specific electric current spatial distribution.

Only available when the method for the **Electric Potential Specification** boundary condition is **Electric Current**.

Specific Electric Current Derivative Option

When defining the specific electric current at the boundary, this option allows you to specify the derivative of the specific electric current with respect to the electric potential, $dJ_n/d\phi$. You typically define $dJ_n/d\phi$ to improve stability, by linearizing the relationship between J_n and ϕ.

Method	Corresponding Physics Value Nodes
None Sets $dJ_n/d\phi$ to zero.	None
Specified Allows you to specify $dJ_n/d\phi$ at the boundary.	**Specific Electric Current Derivative** Defines $dJ_n/d\phi$ as a scalar profile. By default, this value is set to zero (that is, J_n is independent of ϕ).

Only available when the method for the **Electric Potential Specification** boundary condition is **Specific Electric Current**.

Region Settings

Fluid, Solid, and Porous Regions:

The following region settings allow you specify additional sources of electric current, that are not directly modeled in the simulation. When you do not need to account for unresolved physics, you typically leave these source terms to the default value of zero.

Electromotive Force Source Option

Specifies the external source $\boldsymbol{\rho}_\epsilon$ in Eqn. (3727). By default, $\boldsymbol{\rho}_\epsilon = 0$.

Method	Corresponding Physics Value Nodes
Deactivated Sets $\boldsymbol{\rho}_\epsilon = 0$.	None
Activated Allows you to define an electromotive force source term, $\boldsymbol{\rho}_\epsilon$.	**Electromotive Force Source** Sets $\boldsymbol{\rho}_\epsilon$ to a specified vector profile.

Transfer Current Density Source Option

Specifies the external source S_ϕ in Eqn. (3727). By default, $S_\phi = 0$.

Method	Corresponding Physics Value Nodes
Deactivated Sets $S_\phi = 0$.	None
Activated Allows you to define a transfer current density source term, S_ϕ.	**Transfer Current Density Source** Sets S_ϕ to a specified scalar profile. **Transfer Current Density Source Potential Derivative** For stabilization, allows you to specify $dS_\phi/d\phi$ as a scalar profile. By default, this value is set to zero.

Fluid and Porous Regions:

Electrical Conductivity Constraint Option

Available when you activate an energy model, allows you to specify a minimum electrical conductivity for the fluid region.

Method	Corresponding Physics Value Nodes
None Does not set a minimum electrical conductivity.	None
Minimum Allows you to set a minimum value for the fluid electrical conductivity.	**Minimum Constraint Electrical Conductivity** Defines the minimum electrical conductivity as a scalar profile.

Porous Regions:

Effective Porous Electrical Conductivity Option

Specifies the method for calculating the effective electrical conductivity of porous materials.

Method	Corresponding Physics Value Nodes
Volume fraction weighted Calculates the effective electrical conductivity in terms of the electrical conductivity of the fluid and solid components (see Eqn. (3695)). This method is suitable when: • you want to use a specific value for the effective electrical conductivity • the relationship for the effective electrical conductivity does not follow a Bruggeman relationship • you have a mixed conductor, where both the solid and fluid phases conduct charge.	**Solid Electrical Conductivity** Specifies the electrical conductivity of the solid component (σ_{solid} in Eqn. (3695)) as a second-order tensor. For information on how to define second-order tensors, see Tensor Profile Methods. The electrical conductivity of the fluid component, σ_{fluid}, is specified within the material properties of the fluid physics continuum.
Bruggeman (solid) Calculates the effective electrical conductivity in terms of the electrical conductivity of the solid component (see Eqn. (3695)).	**Solid Electrical Conductivity** Specifies σ_{solid} (see Eqn. (3695)) as a second-order tensor. **Bruggeman Approximation Exponent** Specifies the Bruggeman approximation exponent (β in Eqn. (3695)) as a scalar profile. The default value is 1.5.
Bruggeman (fluid) Calculates the effective electrical conductivity in terms of the electrical conductivity of the fluid component (see Eqn. (3695)).	**Bruggeman Approximation Exponent** Specifies β (see Eqn. (3695)) as a scalar profile. The electrical conductivity of the fluid component, σ_{fluid}, is specified within the material properties of the fluid physics continuum.

Interface Settings

Contact Electric Potential Specification

Available for fluid/solid and solid/solid contact interfaces, allows you to prescribe the electric potential at the interface.

Method	Associated Physics Value
None Calculates the electric potential at the two sides of the interface according to Eqn. (3731).	**Electrical Resistance** Defines the resistance R^{ex} in Eqn. (3731) as a scalar profile.
Specific Electric Current Prescribes the specific electric current at the two sides of the interface, J_{n_0}, J_{n_1}.	**Specific Electric Current** Sets the specific electric current $\left\lvert J_{n_0}\right\rvert = \left\lvert J_{n_1}\right\rvert$ to a specified scalar profile. See Eqn. (3733).
Electric Current Prescribes the total electric current through the interface	**Electric Current** Sets the total electric current I_Γ to a specified scalar profile. See Eqn. (3732).
Current Voltage Characteristic Allows you to define the I-V curve at the interface.	**Current Voltage Characteristic** Defines the relationship between the electric current flowing through the interface and the electric potential. In electrochemistry applications, you typically define this relationship using the Butler-Volmer method (see Butler-Volmer Current-Potential Characteristic). Alternatively, you can specify the I-V curve by providing tabular data. For more information, see Setting the Electric Current Potential Characteristic and Current-Voltage Characteristic Reference. **Electrical Resistance** Defines the electrical resistance R in Eqn. (3734) as a scalar profile.

Electric Current Source Option

Available for fluid/fluid interfaces, allows you to prescribe the electric current at the interface.

Method	Associated Physics Value
Specific Electric Current Prescribes the specific electric current at the two sides of the interface, J_{n_0}, J_{n_1}.	**Specific Electric Current** Sets the specific electric current $\left\lvert J_{n_0}\right\rvert = \left\lvert J_{n_1}\right\rvert$ to a specified scalar profile. See Eqn. (3733).

Electric Current Prescribes the total electric current through the interface	**Electric Current** Sets the total electric current I_Γ to a specified scalar profile. See Eqn. (3732).

Baffle Electrical Option

Available for baffle interfaces and porous baffle interfaces, allows you to specify whether electric currents flow through the interface.

Method	Associated Physics Value
Non-Conducting No current flows through the baffle.	None
Conducting Current flows through the baffle.	**Electrical Resistance** Specifies the baffle electrical resistance.

Field Functions

Boundary Specific Electric Current

Represents the specific electric current (Eqn. (3733)), with opposite sign.

Electric Current Density

Represents the electric current density $\mathbf{J}$ in Eqn. (3694).

Electrical Conductivity

Represents the electrical conductivity σ in Eqn. (3694).

This field function is not available for:

- solid regions with **Anisotropic** electrical conductivity
- porous regions

For both of the above cases, the components of the electrical conductivity tensor are available as separate field functions.

Electrical Conductivity Tensor Component

The tensor components of the effective electrical conductivity tensor σ_{eff} for a porous region (see Eqn. (3695)). This field function is also available for solid regions that have a specified electrical conductivity method of Anisotropic.

Contents:

Current-Voltage Characteristic Reference

Current-Voltage Characteristic Reference

This section provides details of the properties of the Current Voltage Characteristic node and sub-nodes.

Current Voltage Characteristic Properties

Method

- **Butler-Volmer**: Uses the Butler-Volmer equation to specify the relation between the specific electric current, J_n, and the overpotential, η. See Eqn. (3735).
- **Tabular Polarization Curve**: Allows you to model arbitrary shaped current-potential curves. It uses tabular data to specify the relationship between the specific electric current, J_n, and the potential drop across the boundary or interface, $\phi_1 - \phi_0$. ϕ_1 and ϕ_0 are defined as in Eqn. (3734). The data in the table must represent mathematical functions of electric potential or specific electric current—depending on the option that is selected for table sorting.

Anodic Transfer Coefficient

The anodic apparent transfer coefficient α_a in Eqn. (3735). This property appears only when the *Method* is set to **Butler-Volmer**.

Cathodic Transfer Coefficient

The cathodic apparent transfer coefficient α_c in Eqn. (3735). This property appears only when the *Method* is set to **Butler-Volmer**.

Specific Exchange Current

This value represents the specific electric exchange current j_0 in Eqn. (3735). This property appears only when the *Method* is set to **Butler-Volmer**.

Zero Electric Current Potential

This value specifies the electric potential drop when there is zero current. The zero electric current potential, ϕ_E in Eqn. (3735). This property appears only when the *Method* is set to **Butler-Volmer**.

Tabular Polarization Curve Properties

Table

Allows you to select the appropriate imported polarization curve file table. This table provides pairs of potential drop ($\phi_1 - \phi_0$) and current values. The table must contain more than two pairs of values.

Note: The polarization curve is an essential part of the simulation and should be specified with care. To enhance convergence of sufficiently complex polarization curves it can help to decrease the under-relaxation parameter of the electric potential solver.

Electric Potential

The column of the table that specifies the electrical potential drop.

Specific Electric Current

The column of the table that specifies the specific current density.

Zero Electric Current Potential

The zero current potential drop ϕ_E in Eqn. (3735) specifies the electric potential drop when there is zero current. The value is extracted from the imported file table and is defined by interpolating the condition $J_n(\phi_E) = 0$

Table Sorting Properties

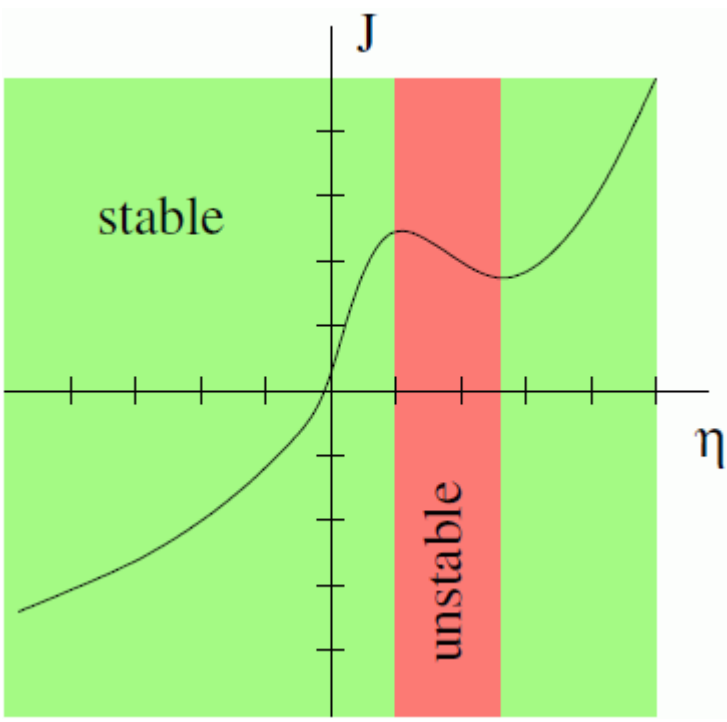
The **Current Voltage Characteristic** > **Tabular Polarization Curve** > **Table Sorting** node provides the following properties:

Option

Allows you to specify how the tabular polarization curve data is sorted upon import to STAR-CCM+.

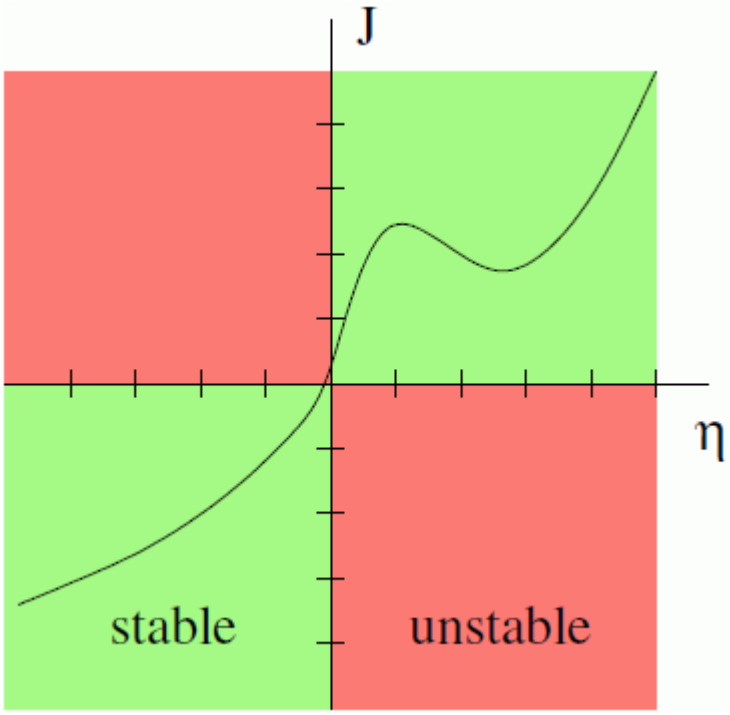
- **Electric Potential**: Sorts the tabular polarization curve data in order of mathematically ascending values of electric potential. The following diagram shows stable and unstable domains of the current-voltage curve when Electric Potential Table Sorting is selected.

  Note: Instability can occur for negative slopes of the current-voltage curve.



- **Specific Electric Current**: Sorts the tabular polarization curve data in order of mathematically ascending values of specific electric current. The following diagram shows stable and unstable domains of the current-voltage curve when the table sorting option is set to Specific Electric Current. The current-voltage curve cannot feature a non-zero electric potential when specific electric current is zero.

  Note: Instability can occur in the second and fourth quadrants of the current-voltage curve.



This feature is useful when the current voltage characteristics represent a non-monotonic mathematical function of specific electric current—such as in some plasma applications.

Note: This option is not available when using the Electrochemical Reactions model.

Harmonic Balance FV Electrodynamic Potential Model Reference

The **Harmonic Balance FV Electrodynamic Potential** model allows you to model electric potentials with harmonic time-dependance using the finite volume approach.

This model calculates the complex electric potential $\widehat{\phi}$ according to Eqn. (3728). From the electric potential, STAR-CCM+ calculates the complex electric field $\widehat{\mathbf{E}}$ and electric current density $\widehat{\mathbf{J}}$ (see Eqn. (3722)). This model requires the **Harmonic Balance FV Magnetic Vector Potential** model, which calculates the complex magnetic vector potential $\widehat{\mathbf{A}}$ in Eqn. (3728).

Model Name	**Harmonic Balance FV Electrodynamic Potential**
Theory	See Harmonic Time Dependence and Electrodynamic Potential Models.
Provided By	**[physics continuum]** > **Models** > *Electromagnetism*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Harmonic Balance FV Electrodynamic Potential**
Requires	Physics Models: *Space*: one of **Three Dimensional**, **Two Dimensional** *Time*: one of **Steady**, **Implicit Unsteady** *Material*: one of **Gas**, **Liquid**, **Solid**

	Other Models: **Electromagnetism** *Other Models*: **Harmonic Balance FV Magnetic Vector Potential**	
Properties	*Secondary Gradients*	
Activates	Material Properties	**Electrical Conductivity**. See Material Properties.
	Reference Values	**Excitation Frequency** See Reference Values.
	Initial Conditions	**Electric Potential**. See Initial Conditions.
	Boundary Inputs	**Electric Potential Specification**. See Boundary Inputs.
	Region Inputs	**Transfer Current Density Source Option** See Region Inputs.
	Interface Inputs	**Contact Electric Potential Specification** See Interface Inputs.
	Solvers	**Harmonic Balance Electric Potential**
	Monitors	**Imaginary Electric Potential**, **Real Electric Potential**
	Field Functions	See Field Functions.

Material Properties

Electrical Conductivity

Defines the electrical conductivity σ (see Eqn. (3717)) as a complex profile. See Complex Profiles.

Reference Values

Excitation Frequency

Allows you to specify the frequency f_0 in Eqn. (3710). STAR-CCM+ requires the frequency to be the same across contact or baffle interfaces. When you associate different continua to contacting regions, make sure that you set the same frequency in both continua.

Initial Conditions

Electric Potential

Allows you to initialize the complex electric potential $\widehat{\phi}$ to a specified complex profile.

Boundary Settings

Applies to all boundary types other than **Overset Mesh** and **Symmetry Plane**.

Electric Potential Specification

At a domain boundary, allows you to prescribe either the electric potential, the electric current, or the electric current density as a complex profile. See Boundary and Interface Conditions.

Method	Corresponding Physics Value Nodes
Specific Electric Current Allows you to define the electric current density normal to the boundary.	**Specific Electric Current** Sets the specific electric current $\hat{J}_n$ to a specified complex profile. $\hat{J}_n$ is the complex equivalent of J_n in Eqn. (3733).
Electric Potential Allows you to define the electric potential at the boundary according to Eqn. (3729).	**Electric Potential** Defines the potential $\overline{\phi}$ in Eqn. (3729) as a complex profile. When leaving the **Electrical Resistance** to the default value of zero, the potential at the boundary is simply $\hat{\phi} = \overline{\phi}$. **Electrical Resistance** Defines an additional source of electrical resistance (R^{ex} in Eqn. (3729)) as a scalar profile. **Electrical Resistance Potential Derivative** For stabilization, allows you to specify the dependence of R^{ex} on ϕ by defining $dR^{\mathrm{ex}}/d\phi$.
Electric Current Allows you to define the total electric current through the boundary.	**Electric Current** Sets the total electric current $\hat{I}_\Gamma$ to a specified complex profile. $\hat{I}_\Gamma$ is the complex equivalent of I_Γ in Eqn. (3732). When using this option, specify the spatial distribution of electric current using the **Electric Current Distribution** condition. See Electric Current Distribution.
Electric Current Density Allows you to define the electric current density at the boundary.	**Electric Current Density** Sets $\hat{\mathbf{J}}$ to a specified complex profile. See Eqn. (3722).
Insulator Sets the total electric current through the boundary (Eqn. (3732)) to zero.	None

Electric Current Distribution

When defining the electric current at the boundary, this setting specifies the electric current spatial distribution. The available options are:

Uniform Electric Potential—calculates the specific electric current spatial distribution so that the electric potential is constant along the boundary.
Uniform Specific Electric Current—prescribes a constant specific electric current spatial distribution.

Only available when the method for the **Electric Potential Specification** boundary condition is **Electric Current**.

Region Settings

Available for fluid and solid regions, to account for unresolved physics.

Transfer Current Density Source Option

Specifies the external source $\widehat{S}_\phi$ in Eqn. (3714). By default, $\widehat{S}_\phi = 0$.

Method	Corresponding Physics Value Nodes
Deactivated Sets $\widehat{S}_\phi = 0$.	None
Activated Allows you to define a transfer current density source term, $\widehat{S}_\phi$.	**Transfer Current Density Source** Sets $\widehat{S}_\phi$ to a specified complex profile. **Transfer Current Density Source Potential Derivative** For stabilization, allows you to specify $d\widehat{S}_\phi/d\phi$ as a complex profile. By default, this value is set to zero.

Interface Settings

Contact Electric Potential Specification

Available for fluid/solid and solid/solid contact interfaces, allows you to prescribe the electric potential at the interface.

Method	Associated Physics Value
None Calculates the electric potential at the two sides of the interface according to Eqn. (3731).	**Electrical Resistance** Defines the resistance R^{ex} in Eqn. (3731) as a scalar profile.
Specific Electric Current Prescribes the specific electric current at the two sides of the interface, $\widehat{J}_{n_0}$, $\widehat{J}_{n_1}$.	**Specific Electric Current** Sets the specific electric current $\left\lvert\widehat{J}_{n_0}\right\rvert = \left\lvert\widehat{J}_{n_1}\right\rvert$ to a specified complex profile. See Eqn. (3733).
Electric Current Prescribes the total electric current through the interface.	**Electric Current** Sets the total electric current $\widehat{I}_\Gamma$ to a specified complex profile. See Eqn. (3732).

Field Functions

Boundary Specific Electric Current (Imag, Real, Phase, Magnitude)

Represent the imaginary part, real part, phase, and magnitude of the complex specific electric current $\hat{J}_n$ (complex equivalent of J_n in Eqn. (3733)), with opposite sign.

Electric Current Density (Imag, Real, Phase, Magnitude)

Represent the imaginary part, real part, phase, and magnitude of the complex electric current density $\hat{\mathbf{J}}$ in Eqn. (3722).

Electrical Conductivity (Imag, Real, Phase, Magnitude)

Represent the imaginary part, real part, phase, and magnitude of the complex electrical conductivity σ in Eqn. (3717).

Magnetic Vector Potential Reference

Contents:

Finite Element Magnetic Vector Potential Model Reference
Finite Volume Magnetic Vector Potential Model Reference
Harmonic Balance FV Magnetic Vector Potential Model Reference
Transverse Magnetic Potential Reference
Harmonic Balance FV Transverse Magnetic Potential Model Reference
Permanent Magnet Model Reference
Excitation Coil Model Reference
Quasi Unsteady Losses Model Reference
Excitation Coil Lumped Parameter Model Reference
Modified Steinmetz Model Reference

Finite Element Magnetic Vector Potential Model Reference

The **Finite Element Magnetic Vector Potential** model allows you to model magnetic fields using the FE (finite element) approach.

The corresponding solver computes the magnetic vector potential $\mathbf{A}$. The magnetic flux density $\mathbf{B}$ and the magnetic field $\mathbf{H}$ are calculated on demand from the magnetic vector potential.

Model Name	**Finite Element Magnetic Vector Potential**
Theory	See Magnetic Vector Potential Models.
Provided By	**[physics continuum]** > **Models** > *Electromagnetism*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Finite Element Magnetic Vector Potential**
Requires	Physics Models:

	Space: one of **Two Dimensional**, **Three Dimensional** *Time*: one of **Steady**, **Implicit Unsteady** *Material*: one of **Gas**, **Liquid**, **Solid**, **Multi-Component Gas**, **Multi-Component Liquid**, **Multi-Part Solid**, **Eulerian Multiphase** *Optional Models*: **Electromagnetism**	
Properties	*Regularization Parameter*, *Integration Order*, *Integration Order Option*. See Finite Element Magnetic Vector Potential Model Properties.	
Activates	Material Properties	**Electrical Conductivity**, **Magnetic Permeability**. See Material Properties.
	Initial Conditions	**Magnetic Vector Potential**. See Initial Conditions.
	Boundary Inputs	**Magnetic Vector Potential Specification**. See Boundary Settings.
	Region Inputs	**Electric Current Density Source Option**. See Region Settings.
	Interface Inputs	**Electric Current Sheet Option**. See Interface Settings.
	Solvers	**Magnetic Vector Potential** (uses the **Sparse Direct Solver**). See FE Magnetic Vector Potential Solver Reference.
	Monitors	**Magnetic Vector Potential**.
	Field Functions	See Field Functions.

Mesh Requirements

The volume mesh (either imported or generated in STAR-CCM+) must meet the following requirements:

- For 2D meshes, the supported element topologies are triangles and quadrilaterals. For 3D meshes, the supported element topologies are tetrahedra, hexahedra, prisms, and pyramids.

  The meshers generally use pyramids as transition elements between different mesh topologies. In general, pyramids are not accurate and should be used only as transition elements.
- At any interface, the mesh must be conformal.

Note: The FE magnetic vector potential solvers only supports geometrically linear elements.

The following table shows how to generate compatible mesh topologies using parts-based mesh operations in STAR-CCM+. If you import a volume mesh, and you wish to remesh it in STAR-CCM+, you can use the same procedure.

Dim	Element Type	Mesh Operation	Meshers
2D	Triangle	**Automated Mesh (2D)**	**Triangular Mesher**
	Quad	**Automated Mesh (2D)**	**Quadrilateral Mesher**
3D	Tetrahedron	**Automated Mesh**	**Tetrahedral Mesher**

Dim	Element Type	Mesh Operation	Meshers
	Prism	**Automated Mesh**	• **Tetrahedral Mesher** with **Thin Mesher**. In regular thin geometries, such as a thin plate, this combination generates prism elements. In more complex geometries, this combination generates a mix of different elements (mainly tetrahedra and prisms). The meshers can generate pyramids as transition elements. • **Tetrahedral Mesher** with **Prism Layer Mesher**. This combination generates a mesh of tetrahedral elements, with prism boundary layers. When using this combination, always set the *Layer Reduction Percentage* to `0`.
		Directed Mesh	**Directed Mesher** (extrude triangular surface mesh). See Directed Meshing.
	Hexahedron	**Directed Mesh**	**Directed Mesher** (extrude quadrilateral surface mesh). See Directed Meshing.

Finite Element Magnetic Vector Potential Model Properties

Regularization Parameter

Allows you to specify the regularization parameter $\tilde{\kappa}$ defined in Eqn. (3742). You are advised to leave this parameter to the default value.

Integration Order Option, *Integration Order*

STAR-CCM+ solves the discretized form of Eqn. (3740) using Gaussian integration rules. These properties allow you to specify the integration order as follows:

Integration Order Option	*Integration Order*
Auto STAR-CCM+ automatically determines an appropriate integration order for the polynomial degree of the shape functions.	Any value that is specified under this property is ignored.
Relative Allows you to increase the integration order that is automatically determined by STAR-CCM+ by an additive value.	Allows you to specify a value which STAR-CCM+ adds to the automatic integration order.
Absolute Gives you full control on the integration order.	Allows you to specify the integration order as an absolute value.

Material Properties

Magnetic Permeability

Specifies the magnetic permeability μ of the material (see Eqn. (3688)).

Method	Associated Value Node
Constant, Field Function Suitable for linear isotropic materials (see Eqn. (3688)). Available for fluids and solids.	**Magnetic Permeability > Constant, Field Function** Specifies μ as a scalar profile.
Table (B,H) Suitable for nonlinear isotropic materials (see Eqn. (3691)). Available for solids.	**Magnetic Permeability > Table (B,H) > Tabular Data** Allows you to specify a nonlinear $\mathbf{B}$-$\mathbf{H}$ curve as a table of B, H values, from which STAR-CCM+ determines the profile of μ. See Using Table(B,H) Method for Permeability. *Table: Magnetic Flux density*—specifies the table column that contains the values of B. *Table: Magnetic Field*—specifies the table column that contains the values of H. *Input Table*—allows you to select the table with the B, H data. You can select an imported file table or a B, H, μ table from the material database.
Anisotropic Suitable for linear (see Eqn. (3688)) and nonlinear (see Eqn. (3691)) anisotropic materials. Available for solids in 3D simulations only. In 2D simulations, $\mathbf{B}$ and $\mathbf{H}$ are normal to the 2D domain and you can always specify μ as a scalar.	**[Region] > Physics Values > Anisotropic Permeability** Specifies μ for the region as a second-order tensor. • For linear materials, you define the magnetic permeability tensor using any of the common methods for tensor profiles (see Tensor Profile Methods). • For nonlinear materials, STAR-CCM+ provides the **Table (B,H) Principal Tensor** method. This method is similar to the **Principal Tensor** method (see Principal Tensor), with the difference that you can specify up to three different nonlinear B-H curves (one for each principal direction) using the **Table (B,H)** method (described above for isotropic solids).

For guidelines on setting the magnetic permeability, see Defining the Magnetic Permeability.

Electrical Conductivity

Specifies the electrical conductivity σ (see Electrical Conductivity: Generalized Ohm's Law) of the material, in transient simulations.

The available methods for defining the electrical conductivity depend on the physics models that you activate in the physics continuum.

For heat transfer analysis, STAR-CCM+ provides specific methods for defining σ as a function of temperature.

Method	Corresponding Physics Value Nodes
Constant, Field Function Available for fluids and solids.	**Electrical Conductivity > Constant, Field Function** Specifies σ as a scalar profile.
Anisotropic Available for solids.	**[Region] > Physics Values > Anisotropic Electrical Conductivity** Specifies σ for the region as a second-order tensor (3x3 matrix in 3D simulations, or 2x2 matrix in 2D simulations). For information on how to define second-order tensors, see Tensor Profile Methods .
Polynomial in T Available for fluids and solids, when you activate an energy model in the physics continuum.	**Electrical Conductivity > Polynomial in T** Specifies σ as a polynomial function of temperature, for heat transfer analysis. See Using Polynomial in T.
Table (T) Available for fluids and solids, when you activate an energy model in the physics continuum.	**Electrical Conductivity > Table (T)** Allows you to define σ as a function of temperature by providing a table of σ, T values, from which STAR-CCM+ determines the profile of $\sigma(T)$. See Using Table(T). The temperature range in the table must be consistent with the **Minimum/Maximum Allowable Temperature** settings that are specified under the **Reference Values** node for the physics continuum. This requirement is specific to electrical conductivity.
Table (T,P) Available for compressible gases, when you activate an energy model in the physics continuum.	**Electrical Conductivity > Table (T,P)** Allows you to define σ as a function of temperature and pressure by providing a table of σ, T, p values, from which STAR-CCM+ determines the profile of $\sigma(T, p)$. See Using Table (T,P).
Null Available for multi-part solids. Suitable for modeling non-conducting parts of a multi-part solid in transient simulations. Sets $\sigma = 0$.	None

Initial Conditions

Magnetic Vector Potential

Allows you to initialize the magnetic vector potential $\mathbf{A}$ to a specified vector profile.

Boundary Settings

Magnetic Vector Potential Specification

STAR-CCM+ provides several methods to specify the magnetic vector potential and the electric current sheet at boundaries (see Boundary and Interface Conditions).

In two-dimensional simulations (Transverse Electric mode), the magnetic field is normal to the 2D domain. Therefore, the magnetic vector potential lies in the 2D domain and can be defined by two components.

Method	Corresponding Physics Value Nodes
Electric Current Sheet Neumann b. c. that sets the electric current sheet $\mathbf{J}_S$ to the tangential component of a specified vector profile, $\bar{\mathbf{J}}_S$: $$\mathbf{J}_S = \bar{\mathbf{J}}_S\big\|_{t_1,\, t_2}$$	**Electric Current Sheet** Allows you to specify a vector profile, $\bar{\mathbf{J}}_S$. STAR-CCM+ applies the components of $\bar{\mathbf{J}}_S$ tangential to the boundary and neglects the component of $\bar{\mathbf{J}}_S$ normal to the boundary.
Magnetic Vector Potential Dirichlet b. c. that sets the tangential component of the magnetic vector potential $\mathbf{A}$ to the tangential component of a specified vector profile, $\bar{\mathbf{A}}$: $$\mathbf{A}\big\|_{t_1,\, t_2} = \bar{\mathbf{A}}\big\|_{t_1,\, t_2}$$	**Magnetic Vector Potential** Allows you to specify a vector profile, $\bar{\mathbf{A}}$. STAR-CCM+ applies the components of $\bar{\mathbf{A}}$ tangential to the boundary, and neglects the component of $\bar{\mathbf{A}}$ normal to the boundary.
Tangential Magnetic Field Neumann b. c. that sets the electric current sheet to the tangential component of a specified magnetic field $\bar{\mathbf{H}}$: $$\mathbf{J}_S = \bar{\mathbf{H}} \times \mathbf{n}$$ where $\mathbf{n}$ is the unit vector normal to the boundary.	**Specific Tangential Magnetic Field** Allows you to specify the magnetic field as a vector profile, $\bar{\mathbf{H}}$. STAR-CCM+ sets $\bar{\mathbf{H}} \times \mathbf{n} = \mathbf{J}_S$. The component of $\bar{\mathbf{H}}$ normal to the boundary is ignored.
Symmetry - Perfect Magnetic Conductor Neumann b. c. that sets the electric current sheet to zero: $$\mathbf{J}_S = 0$$ For example, you specify $\mathbf{J}_S = 0$ at an interface with a highly permeable metal, where the magnetic flux is forced to cross the boundary at an angle of 90°.	None
Anti-Symmetry - Perfect Electric Conductor Dirichlet b. c. that sets the tangential component of the magnetic vector potential $\mathbf{A}$ to zero, while leaving the normal component free:	None

$$\mathbf{A}|_{t_1,\, t_2} = 0$$
$$A_n \;\; \text{free}$$

Most commonly, you specify $\mathbf{A}|_{t_1,\, t_2} = 0$ at a boundary to prevent any magnetic flux from crossing the boundary.

Region Settings

Applies to fluid, porous, and solid regions.

Electric Current Density Source Option

Allows you to specify an external source of electric current density. When you activate the **Electrodynamic Potential** model or the **Excitation Coil** model, which define electric current density sources, this option is not available.

Method	Corresponding Physics Value Nodes
None Sets the user-defined region electric current density to zero ($\mathbf{J}_u = 0$ in Eqn. (3747)).	None
Specified Allows you to define an external electric current density source ($\mathbf{J}_u$ in Eqn. (3747)) for the region.	**Electric Current Density Source** Specifies the region electric current density as a vector profile.

Interface Settings

Applies to internal and direct contact interfaces.

Electric Current Sheet Option

Allows you to specify the electric current sheet at the interface.

Method	Corresponding Physics Value Nodes
None Sets the electric current sheet at the interface to zero.	None
Electric Current Sheet Sets the electric current sheet to the tangential component of a specified vector profile.	**Electric Current Sheet** Allows you to specify a vector profile, $\bar{\mathbf{J}}_S$. STAR-CCM+ applies the components of $\bar{\mathbf{J}}_S$ tangential to the interface and neglects the component of $\bar{\mathbf{J}}_S$ normal to the interface.

Field Functions

Magnetic Vector Potential

Vector field function that represents the magnetic vector potential $\mathbf{A}$ in Eqn. (3706).

Magnetic Flux Density

Vector field function that represents the magnetic flux density, $\mathbf{B}$, which is related to the magnetic vector potential, $\mathbf{A}$, through Eqn. (3698).

Magnetic Field

Vector field function that represents the magnetic field $\mathbf{H}$, which is related to the magnetic flux density, $\mathbf{B}$, through Eqn. (3688) or Eqn. (3691).

Electric Current Density

Vector field function that represents the electric current density $\mathbf{J}$ in Eqn. (3694).

Permeability

Scalar field function that represents the magnetic permeability μ in Eqn. (3688) or Eqn. (3691).

BoundaryElectricCurrentSheet

Vector field function that represents the electric current sheet $\mathbf{J}_S$. See Eqn. (3749).

Electromagnetic Nodal Force

Represents the electromagnetic force at mesh nodes ($\mathbf{F}_M$ in Eqn. (3773)).

To compute the total electromagnetic force on a region, use this field function in a Sum report for the relevant region.

Contents:

FE Magnetic Vector Potential Solver Reference

FE Magnetic Vector Potential Solver Reference

Solves for the magnetic vector potential $\mathbf{A}$ using the FE method.

Magnetic Vector Potential Solver Properties

Under-Relaxation Factor

During the iterative process, the **Magnetic Vector Potential** solver under-relaxes the changes in the solution, as:

$$\mathbf{A}^{n+1} = \mathbf{A}^n + \omega \mathbf{A}'$$

where ω is the under-relaxation factor. At each iteration, the under-relaxation factor governs the extent to which the new solution replaces the old one.

The default value is 1.0. If residuals show solution divergence and do not decrease, reduce the under-relaxation factor.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Temporary Storage Retained

When activated, STAR-CCM+ retains the Magnetic Vector Potential residuals computed at each iteration and makes them available as field functions:

- **Magnetic Vector Potential-X Residual**
- **Magnetic Vector Potential-Y Residual**
- **Magnetic Vector Potential-Z Residual**

Equation Solver

Specifies the solution method. Currently, STAR-CCM+ uses a direct solver.

Magnetic Vector Potential Solver Controls

Sparse Direct Solver

For more information on the direct solver properties and controls, see Sparse Direct Solver Reference.

Matrix Update Strategy

At each solver iteration *i*, STAR-CCM+ solves Eqn. (3996) for the magnetic vector potential. The solver assembles the tangent matrix $\mathbf{T}$ during the first iteration. In subsequent iterations, STAR-CCM+ updates the tangent matrix based on the specified **Matrix Update Strategy**. The available options are:

- **Reuse**—STAR-CCM+ factorizes the tangent matrix before the first solver iteration, and then reuses it in all subsequent iterations.

  This option is available when the simulation only contains linear materials (materials for which the B-H relationship is linear).

  Note: STAR-CCM+ detects linear setups only if all input values are specified using the **Constant** method.

  Nonlinear applications require an update of the tangent matrix with an appropriate Newton scheme.
- **Full Newton**—STAR-CCM+ factorizes the tangent matrix at each solver iteration. This option is always available.
- **Modified Newton**—For many nonlinear applications, it is not necessary to update the tangent matrix at each iteration to reach a converged solution (Full Newton scheme). For these cases, you can use the **Modified Newton** scheme, which is available when the simulation contains nonlinear materials (materials for which the B-H relationship is nonlinear).

  The **Modified Newton Update** > **Start Iteration** node allows you to specify the following properties:
  - *Start Reusing*—Specifies the number of iterations to run before STAR-CCM+ starts reusing the tangent matrix.
  - *Update Frequency*—Specifies the tangent matrix update frequency, *M*. For *M*=0, STAR-CCM+ stops updating the matrix when the *Start Reusing* criterion is met. For *M*>0, STAR-CCM+ continues updating the tangent matrix every *M* iterations, after the *Start Reusing* criterion is met.

Finite Volume Magnetic Vector Potential Model Reference

The **Finite Volume Magnetic Vector Potential** model allows you to model magnetic fields using the FV (finite volume) approach.

The corresponding solver solves for the magnetic vector potential according to Eqn. (3737) or Eqn. (3738). The magnetic flux density $\mathbf{B}$ and the magnetic field $\mathbf{H}$ are calculated on the demand from the magnetic vector potential. You can use the **Finite Volume Magnetic Vector Potential** model in combination with the **Electrodynamic Potential** model, which includes the electric potential term in Eqn. (3737) or Eqn. (3738).

Model Name	**Finite Volume Magnetic Vector Potential**	
Theory	See Magnetic Vector Potential Models.	
Provided By	**[physics continuum]** > **Models** > *Electromagnetism*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Finite Volume Magnetic Vector Potential**	
Requires	*Space*: one of **Two Dimensional**, **Three Dimensional**, **Axisymmetric** *Time*: one of **Steady**, **Implicit Unsteady** *Material*: one of **Gas**, **Liquid**, **Solid**, **Multi-Component Gas**, **Multi-Component Liquid**, **Multi-Part Solid**, **Eulerian Multiphase** *Optional Models*: **Electromagnetism**	
Activates	Material Properties	**Electrical Conductivity**, **Magnetic Permeability**. See Material Properties.
	Initial Conditions	**Magnetic Vector Potential**. See Initial Conditions.
	Boundary Inputs	**Magnetic Vector Potential Specification**. See Boundary Settings.
	Region Inputs	**Electric Current Density Source Option**. See Region Settings.
	Solvers	**Magnetic Vector Potential** (uses the **AMG Linear Solver**. See Magnetic Vector Potential Solver. **Permeability**. See Permeability Solver.
	Monitors	**Magnetic X-Potential**, **Magnetic Y-Potential**, **Magnetic Z-Potential**.
	Report Options	**Magnetic Flux Linkage**. See Reports.
	Field Functions	**Magnetic Vector Potential**, **Magnetic Flux Density**, **Magnetic Field**, **Electric Current Density**, **Permeability**, **Electromechanical Stress Tensor**, **Electromagnetic Force Density**. See Field Functions.

Material Properties

Magnetic Permeability

Specifies the magnetic permeability μ of the material (see Eqn. (3688)).

Method	Associated Value Node
Constant, Field Function Suitable for linear isotropic materials (see Eqn. (3688)). Available for fluids and solids.	**Magnetic Permeability > Constant, Field Function** Specifies μ as a scalar profile.
Table (B,H) Suitable for nonlinear isotropic materials (see Eqn. (3691)). Available for solids.	**Magnetic Permeability > Table (B,H) > Tabular Data** Allows you to specify a nonlinear **B**-**H** curve as a table of B, H values, from which STAR-CCM+ determines the profile of μ. See Using Table(B,H) Method for Permeability. *Table: Magnetic Flux density*—specifies the table column that contains the values of B. *Table: Magnetic Field*—specifies the table column that contains the values of H. *Input Table*—allows you to select the table with the B, H data. You can select an imported file table or a B, H, μ table from the material database.
Tabular Suitable for nonlinear isotropic materials (see Eqn. (3691)). Available for solids.	**Tabular > Table** Allows you to provide a B, H curve using a table, from which STAR-CCM+ determines the profile of μ. See Using Tabular Data. • If the provided B, H curve is monotonous and convex, that is: $\begin{cases} \frac{dB}{dH} > 0 \\ \frac{d^2B}{dH^2} < 0 \end{cases}$ set **H** as the interpolating variable in the method. • If the curve is concave, set the magnetic flux density **B** as the interpolating variable.

For guidelines on setting the magnetic permeability, see Defining the Magnetic Permeability.

Electrical Conductivity

Specifies the electrical conductivity σ (see Electrical Conductivity: Generalized Ohm's Law) of the material, in transient simulations.

The available methods for defining the electrical conductivity depend on the physics models that you activate in the physics continuum.

For heat transfer analysis, STAR-CCM+ provides specific methods for defining σ as a function of temperature.

Method	Corresponding Physics Value Nodes
Constant, Field Function Available for fluids and solids.	**Electrical Conductivity > Constant, Field Function** Specifies σ as a scalar profile.
Polynomial in T Available for fluids and solids, when you activate an energy model in the physics continuum.	**Electrical Conductivity > Polynomial in T** Specifies σ as a polynomial function of temperature, for heat transfer analysis. See Using Polynomial in T.
Table (T) Available for fluids and solids, when you activate an energy model in the physics continuum.	**Electrical Conductivity > Table (T)** Allows you to define σ as a function of temperature by providing a table of σ, T values, from which STAR-CCM+ determines the profile of $\sigma(T)$. See Using Table(T). The temperature range in the table must be consistent with the **Minimum/Maximum Allowable Temperature** settings that are specified under the **Reference Values** node for the physics continuum. This requirement is specific to electrical conductivity.
Table (T,P) Available for compressible gases, when you activate an energy model in the physics continuum.	**Electrical Conductivity > Table (T,P)** Allows you to define σ as a function of temperature and pressure by providing a table of σ, T, p values, from which STAR-CCM+ determines the profile of $\sigma(T, p)$. See Using Table (T,P).

Initial Conditions

Magnetic Vector Potential

Allows you to initialize the magnetic vector potential $\mathbf{A}$ to a specified vector profile.

Boundary Settings

Magnetic Vector Potential Specification

STAR-CCM+ provides several methods to specify the magnetic vector potential and the electric current sheet at boundaries (see Boundary and Interface Conditions).

In two-dimensional simulations (Transverse Electric mode), the magnetic field is normal to the 2D domain. Therefore, the magnetic vector potential lies in the 2D domain and can be defined by two components.

Method	Corresponding Physics Value Nodes
Electric Current Sheet Neumann b. c. that sets the electric current sheet $\mathbf{J}_S$ to the tangential component of a specified vector profile, $\bar{\mathbf{J}}_S$:	**Electric Current Sheet** Allows you to specify a vector profile, $\bar{\mathbf{J}}_S$. STAR-CCM+ applies the components of $\bar{\mathbf{J}}_S$ tangential

$\mathbf{J}_S = \bar{\mathbf{J}}_S\|_{t_1,\, t_2}$	to the boundary and neglects the component of $\bar{\mathbf{J}}_S$ normal to the boundary.
Magnetic Vector Potential Dirichlet b. c. that sets the magnetic vector potential at the boundary to a specified vector profile: $\mathbf{A} = \overline{\mathbf{A}}$ When the external magnetic field is strong, the magnetic vector potential can be written as $\mathbf{A} = \mathbf{A}_{\mathrm{imp}} + \mathbf{A}_{\mathrm{ind}}$, where $\mathbf{A}_{\mathrm{imp}}$ is the magnetic vector potential associated with the external magnetic field and $\mathbf{A}_{\mathrm{ind}}$ is the magnetic vector potential associated with the additional magnetic field induced by electric currents. By explicitly setting the magnetic vector potential at a boundary, you assume that the induced magnetic vector potential $\mathbf{A}_{\mathrm{ind}}$ in the domain decays fast enough so that it can be neglected at the boundary. No electric current flows through the boundary.	**Magnetic Vector Potential** Allows you to explicitly set the magnetic vector potential at the boundary as a vector profile, $\mathbf{A} = \overline{\mathbf{A}}$.
Tangential Magnetic Field Neumann b. c. that sets the electric current sheet to the tangential component of a specified magnetic field $\overline{\mathbf{H}}$: $\mathbf{J}_S = \overline{\mathbf{H}} \times \mathbf{n}$ where $\mathbf{n}$ is the unit vector normal to the boundary.	**Specific Tangential Magnetic Field** Allows you to specify the magnetic field as a vector profile, $\overline{\mathbf{H}}$. STAR-CCM+ sets $\overline{\mathbf{H}} \times \mathbf{n} = \mathbf{J}_S$. The component of $\overline{\mathbf{H}}$ normal to the boundary is ignored.
Symmetry - Perfect Magnetic Conductor Dirichlet b. c. that sets the component of the magnetic vector potential normal to the boundary to zero, while leaving the tangential components free: $A_{t_{1,2}}$ free $A_n = 0$	None
Anti-Symmetry - Perfect Electric Conductor Dirichlet b. c. that sets the tangential component of the magnetic vector potential $\mathbf{A}$ to zero, while leaving the normal component free: $\mathbf{A}\|_{t_1,\, t_2} = 0$ A_n free	None

Most commonly, you specify $\mathbf{A}|_{t_1, t_2} = 0$ at a boundary to prevent any magnetic flux from crossing the boundary.

Region Settings

Applies to fluid, porous, and solid regions.

Electric Current Density Source Option

Allows you to specify an external source of electric current density. When you activate the **Electrodynamic Potential** model or the **Excitation Coil** model, which define electric current density sources, this option is not available.

Method	Corresponding Physics Value Nodes
None Sets the user-defined region electric current density to zero ($\mathbf{J}_u = 0$ in Eqn. (3747)).	None
Specified Allows you to define an external electric current density source ($\mathbf{J}_u$ in Eqn. (3747)) for the region.	**Electric Current Density Source** Specifies the region electric current density as a vector profile.

Magnetic Vector Potential Solver

Solves for the magnetic vector potential, with a choice of conservative or a non-conservative formulation. The available properties are:

Under-Relaxation Factor

During the iterative process, the **Magnetic Vector Potential** solver under-relaxes the changes in the solution, as:

$$\mathbf{A}^{n+1} = \mathbf{A}^n + \omega \mathbf{A}'$$

where ω is the under-relaxation factor. At each iteration, the under-relaxation factor governs the extent to which the new solution replaces the old one.

The default value is 1.0. If residuals show solution divergence and do not decrease, reduce the under-relaxation factor.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

Allows you to set reconstruction gradients to zero. When activated, the solver sets reconstruction gradients to zero at the next iteration. Gradients remain zeroed.

When deactivated, the solver does not reset reconstruction gradients. The gradients are recomputed at the next iteration if they have been zeroed.

Temporary Storage Retained

When activated, the solver retains the following data, computed at each iteration, and makes them available as field functions:

- **Magnetic Vector Potential Correction-X,Y,Z Ap Coefficient**
- **Magnetic Vector Potential Correction-X,Y,Z Residual**
- **Magnetic Vector Potential-X,Y,Z Gradient**
- **Magnetic Vector Potential-X,Y,Z Recon**

Formulation

Specifies which formulation is used for computing the magnetic vector potential. The available options are:

- **Conservative**: solves Eqn. (3737)
- **Non-Conservative**: solves Eqn. (3738), with $\nabla \cdot \mathbf{A} = 0$. This option requires an additional solver, the **Magnetic Vector Potential Projection** solver, that you enable by activating the **Magnetic Vector Potential Projection** model.

Permeability Solver

Controls the solution of the magnetic permeability in all the continua that include either the **Finite Volume Magnetic Vector Potential** model or the **Transverse Magnetic Potential** model.

Reports

Magnetic Flux Linkage

Evaluates the flux linkage through a specified surface. This report allows you to obtain the flux linkage that is induced into a coil. Specify the regions that you want to include in the report. Optionally, activate **Smooth Values** to interpolate reported values to the mesh vertices. By default, this option is deactivated and the report uses the values in the closest cell centroid.

Field Functions

Magnetic Vector Potential

Vector field function that represents the magnetic vector potential $\mathbf{A}$ in Eqn. (3706).

Magnetic Flux Density

Vector field function that represents the magnetic flux density, $\mathbf{B}$, which is related to the magnetic vector potential, $\mathbf{A}$, through Eqn. (3698).

Magnetic Field

Vector field function that represents the magnetic field $\mathbf{H}$, which is related to the magnetic flux density, $\mathbf{B}$, through Eqn. (3688) or Eqn. (3691).

Electric Current Density

Vector field function that represents the electric current density $\mathbf{J}$ in Eqn. (3694).

Permeability

Scalar field function that represents the magnetic permeability μ in Eqn. (3688) or Eqn. (3691).

Electromechanical Stress Tensor

See Eqn. (3769) and Eqn. (3770).

Electromagnetic Force Density

See Eqn. (3772).

For information on the **Magnetic Vector Potential** solver field functions, see Magnetic Vector Potential Solver.

Harmonic Balance FV Magnetic Vector Potential Model Reference

The **Harmonic Balance FV Magnetic Vector Potential** model allows you to model magnetic fields with harmonic time-dependance using the finite volume approach.

The corresponding solver solves for the complex magnetic vector potential according to Eqn. (3713). The magnetic flux $\mathbf{B}$ and the magnetic field $\mathbf{H}$ are calculated from Eqn. (3721). You can use the **Harmonic Balance FV Magnetic Vector Potential** model in combination with the **Excitation Coil** model.

Model Name	**Harmonic Balance FV Magnetic Vector Potential**	
Theory	See Harmonic Time Dependence.	
Provided By	**[physics continuum]** > **Models** > *Electromagnetism*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Harmonic Balance Fv Magnetic Vector Potential**	
Requires	*Space*: one of **Two Dimensional**, **Three Dimensional** *Time*: one of **Steady**, **Implicit Unsteady** *Material*: one of **Gas**, **Liquid**, **Solid** *Optional Models*: **Electromagnetism**	
Activates	Material Properties	**Electrical Conductivity**, **Magnetic Permeability**. See Material Properties.
	Reference Values	**Excitation Frequency**. See Reference Values.
	Initial Conditions	**Magnetic Vector Potential**. See Initial Conditions.

	Boundary Inputs	**Magnetic Vector Potential Specification**. See Boundary Settings.
	Region Inputs	**Electric Current Density Source Option**. See Region Settings.
	Solvers	**Harmonic Balance Magnetic Vector Potential** (uses the **AMG Linear Solver**.
	Monitors	**Imaginary Magnetic X-Potential**, **Imaginary Magnetic Y-Potential**, **Imaginary Magnetic Z-Potential**, **Real Magnetic X-Potential**, **Real Magnetic Y-Potential**, **Real Magnetic Z-Potential**.
	Field Functions	**Magnetic Vector Potential** (Imag, Real, Phase, Magnitude), **Magnetic Flux Density** (Imag, Real, Phase, Magnitude), **Magnetic Field** (Imag, Real, Phase, Magnitude), **Permeability**, **Electrical Conductivity** (Imag, Real, Phase, Magnitude). See Field Functions.

Reference Values

Excitation Frequency

Allows you to specify the frequency f_0 in Eqn. (3710). STAR-CCM+ requires the frequency to be the same across contact or baffle interfaces. When you associate different continua to contacting regions, make sure that you set the same frequency in both continua.

Material Properties

Magnetic Permeability

Read-only. The current formulation assumes that the magnetic permeability is constant and equal to the vacuum permeability, $\mu = \mu_0$.

Electrical Conductivity

σ in Eqn. (3717). Set as a complex profile.

Initial Conditions

Magnetic Vector Potential

Allows you to specify the initial magnetic vector potential $\widehat{\mathbf{A}}$ (see Eqn. (3713)) as a complex profile.

Boundary Settings

Magnetic Vector Potential Specification

STAR-CCM+ provides several methods to specify the magnetic vector potential and the electric current sheet at boundaries (see Boundary and Interface Conditions). For harmonic time dependence, these quantities are specified as complex profiles.

Method	Corresponding Physics Value Nodes
Electric Current Sheet Sets the electric current sheet $\hat{\mathbf{J}}_S$ to the tangential component of a specified complex profile, $\hat{\bar{\mathbf{J}}}_S$: $$\hat{\mathbf{J}}_S = \hat{\bar{\mathbf{J}}}_S\Big\vert_{t_1,\, t_2}$$	**Electric Current Sheet** Allows you to specify a complex profile, $\hat{\bar{\mathbf{J}}}_S$. STAR-CCM+ applies the components of $\hat{\bar{\mathbf{J}}}_S$ tangential to the boundary and neglects the component normal to the boundary.
Magnetic Vector Potential Sets the magnetic vector potential at the boundary to a specified complex profile.	**Magnetic Vector Potential** Allows you to explicitly set the magnetic vector potential at the boundary as a complex profile, $\hat{\mathbf{A}} = \hat{\bar{\mathbf{A}}}$.
Tangential Magnetic Field Sets the electric current sheet to the tangential component of a specified magnetic field $\hat{\bar{\mathbf{H}}}$: $$\hat{\mathbf{J}}_S = \hat{\bar{\mathbf{H}}} \times \mathbf{n}$$ where $\mathbf{n}$ is the unit vector normal to the boundary.	**Specific Tangential Magnetic Field** Allows you to specify the magnetic field as a complex profile, $\hat{\bar{\mathbf{H}}}$. STAR-CCM+ sets $\hat{\bar{\mathbf{H}}} \times \mathbf{n} = \mathbf{J}_S$. The component of $\hat{\bar{\mathbf{H}}}$ normal to the boundary is ignored.
Symmetry - Perfect Magnetic Conductor Sets the component of the magnetic vector potential normal to the boundary to zero, while leaving the tangential components free: $$\hat{A}_{t_{1,2}} \text{ free}$$ $$\hat{A}_n = 0$$	None
Anti-Symmetry - Perfect Electric Conductor Sets the tangential component of the magnetic vector potential $\hat{\mathbf{A}}$ to zero, while leaving the normal component free: $$\hat{\mathbf{A}}\Big\vert_{t_1,\, t_2} = 0$$ $$\hat{A}_n \text{ free}$$ Most commonly, you specify $\hat{\mathbf{A}}\Big\vert_{t_1,\, t_2} = 0$ at a boundary to prevent any magnetic flux from crossing the boundary.	None

Region Settings

Applies to fluid, porous, and solid regions.

Electric Current Density Source Option

Allows you to specify an electric current density for the region. This option is not available when you use the **Harmonic Balance Fv Magnetic Vector Potential** model in combination with the **Excitation Coil** model. In this case, the electric current density is supplied by the **Excitation Coil** model.

Method	Corresponding Physics Value Nodes
None No electric current density source is defined.	None
Specified Specifies a user-defined electric current density source ($\mathbf{J}_u$ in Eqn. (3747)).	**Complex Electric Current Density Source** Allows you to specify the electric current density as a complex profile.

Field Functions

Magnetic Vector Potential (Imag, Real, Phase, Magnitude)

Represent the imaginary part, real part, phase, and magnitude of the complex magnetic vector potential $\widehat{\mathbf{A}}$ (see Eqn. (3711)).

Magnetic Flux Density (Imag, Real, Phase, Magnitude)

Represent the imaginary part, real part, phase, and magnitude of the complex magnetic flux density $\widehat{\mathbf{B}}$ (see Eqn. (3721)).

Magnetic Field (Imag, Real, Phase, Magnitude)

Represent the imaginary part, real part, phase, and magnitude of the complex magnetic field $\widehat{\mathbf{H}}$ (see Eqn. (3721)).

Electric Current Density (Imag, Real, Phase, Magnitude)

Represent the imaginary part, real part, phase, and magnitude of the complex electric current density $\widehat{\mathbf{J}}$ in Eqn. (3722). The electric current density magnitude, $\left|\widehat{\mathbf{J}}\right|$, is also referred to as the peak value.

Permeability

Represents the magnetic permeability $\mu = \mu_0$.

Electrical Conductivity (Imag, Real, Phase, Magnitude)

Represent the imaginary part, real part, phase, and magnitude of the complex electrical conductivity σ in Eqn. (3717).

Transverse Magnetic Potential Reference

The **Transverse Magnetic Potential** model allows you to model transverse-magnetic modes, that is, magnetic fields which lie on a 2D domain.

This model solves for the transverse magnetic potential according to Eqn. (3752). The magnetic flux density $\mathbf{B}$ and the magnetic field $\mathbf{H}$ are calculated on demand from the magnetic potential-Z.

Model Name	**Transverse Magnetic Potential**	
Theory	See Transverse Magnetic Modes.	
Provided By	**[physics continuum]** > **Models** > *Electromagnetism*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Transverse Magnetic Potential**	
Requires	*Space*: one of **Two Dimensional**, **Axisymmetric** *Time*: one of **Steady**, **Implicit Unsteady** *Material*: one of **Gas**, **Liquid**, **Solid**, **Multi-Component Gas**, **Multi-Component Liquid**, **Multi-Part Solid**, **Eulerian Multiphase** *Optional Models*: **Electromagnetism**	
Activates	Material Properties	**Electrical Conductivity**, **Magnetic Permeability**. See Material Properties.
	Initial Conditions	**Magnetic Vector Potential Z**. See Initial Conditions.
	Boundary Inputs	**Magnetic Vector Potential Specification**. See Boundary Settings.
	Region Inputs	**Electric Current Density Source Option**, **Current Conservation Option**. See Region Settings.
	Interface Inputs	**Magnetic Vector Potential Z Periodicity**. See Interface Settings.
	Solvers	**Transverse Magnetic Potential** (uses the **AMG Linear Solver**. See Transverse Magnetic Potential Solver. **Permeability**. See Permeability Solver.
	Monitors	**Transverse Magnetic Potential**.
	Report Options	**Magnetic Flux Linkage**. See Reports.
	Field Functions	**Magnetic Vector Potential**, **Magnetic Flux Density**, **Magnetic Field**, **Electric Current Density**, **Permeability**, **Electromechanical Stress Tensor**, **Electromagnetic Force Density**. See Field Functions.

Material Properties

Magnetic Permeability

Specifies the magnetic permeability μ of the material (see Eqn. (3688)).

Method	Associated Value Node
Constant, Field Function Suitable for linear isotropic materials (see Eqn. (3688)). Available for fluids and solids.	**Magnetic Permeability > Constant, Field Function** Specifies μ as a scalar profile.
Table (B,H) Suitable for nonlinear isotropic materials (see Eqn. (3691)). Available for solids.	**Magnetic Permeability > Table (B,H) > Tabular Data** Allows you to specify a nonlinear **B**-**H** curve as a table of B, H values, from which STAR-CCM+ determines the profile of μ. See Using Table(B,H) Method for Permeability. *Table: Magnetic Flux density*—specifies the table column that contains the values of B. *Table: Magnetic Field*—specifies the table column that contains the values of H. *Input Table*—allows you to select the table with the B, H data. You can select an imported file table or a B, H, μ table from the material database.
Tabular Suitable for nonlinear isotropic materials (see Eqn. (3691)). Available for solids.	**Tabular > Table** Allows you to provide a B, H curve using a table, from which STAR-CCM+ determines the profile of μ. See Using Tabular Data. • If the provided B, H curve is monotonous and convex, that is: $\begin{cases} \frac{dB}{dH} > 0 \\ \frac{d^2B}{dH^2} < 0 \end{cases}$ set **H** as the interpolating variable in the method. • If the curve is concave, set the magnetic flux density **B** as the interpolating variable.

For guidelines on setting the magnetic permeability, see Defining the Magnetic Permeability.

Electrical Conductivity

Specifies the electrical conductivity σ (see Electrical Conductivity: Generalized Ohm's Law) of the material, in transient simulations.

The available methods for defining the electrical conductivity depend on the physics models that you activate in the physics continuum.

For heat transfer analysis, STAR-CCM+ provides specific methods for defining σ as a function of temperature.

Method	Corresponding Physics Value Nodes
Constant, Field Function Available for fluids and solids.	**Electrical Conductivity > Constant, Field Function** Specifies σ as a scalar profile.
Polynomial in T Available for fluids and solids, when you activate an energy model in the physics continuum.	**Electrical Conductivity > Polynomial in T** Specifies σ as a polynomial function of temperature, for heat transfer analysis. See Using Polynomial in T.
Table (T) Available for fluids and solids, when you activate an energy model in the physics continuum.	**Electrical Conductivity > Table (T)** Allows you to define σ as a function of temperature by providing a table of σ, T values, from which STAR-CCM+ determines the profile of $\sigma(T)$. See Using Table(T). The temperature range in the table must be consistent with the **Minimum/Maximum Allowable Temperature** settings that are specified under the **Reference Values** node for the physics continuum. This requirement is specific to electrical conductivity.
Table (T,P) Available for compressible gases, when you activate an energy model in the physics continuum.	**Electrical Conductivity > Table (T,P)** Allows you to define σ as a function of temperature and pressure by providing a table of σ, T, p values, from which STAR-CCM+ determines the profile of $\sigma(T, p)$. See Using Table (T,P).

Initial Conditions

Magnetic Vector Potential Z

Allows you to initialize the transverse magnetic vector potential, that is, the component of the magnetic vector potential normal to the 2D domain, to a specified scalar profile.

Boundary Settings

Magnetic Vector Potential Specification

In transverse magnetic mode, the magnetic vector potential and electric current sheet are normal to the 2D domain. As only the normal component is non-zero, you specify the magnetic vector potential and the current sheet at a boundary using scalar profiles (see Boundary and Interface Conditions).

Method	Corresponding Physics Value Nodes
Electric Current Sheet Z Neumann b. c. that sets the electric current sheet at the boundary to: $$\mathbf{J}_S = \bar{J}_S \mathbf{n}$$	**Electric Current Sheet** Allows you to specify the normal electric current sheet as a scalar profile, $\bar{J}_S$.

where $\mathbf{n}$ is the unit vector normal to the boundary and $\overline{J}_S$ is a specified scalar profile.	
Magnetic Vector Potential Z Dirichlet b. c. that specifies the magnetic vector potential at the boundary as: $$\mathbf{A} = \overline{A}\mathbf{n}$$ where $\mathbf{n}$ is the unit vector normal to the boundary and $\overline{A}$ is a specified scalar profile.	**Magnetic Vector Potential** Allows you to specify the transverse magnetic vector potential as a scalar profile, $\overline{A}$.

Region Settings

Applies to fluid, porous, and solid regions.

Current Conservation Option

When activated, STAR-CCM+ calculates the transverse magnetic potential while enforcing electric current conservation within the region. See Eqn. (3754). Deactivate this option when you explicitly specify the transverse magnetic potential at the outer boundaries of conducting regions (that is, when you use the **Magnetic Vector Potential Z** boundary condition).

Only available for transient simulations.

Electric Current Density Source Option

Allows you to specify the electric current density for the region. This option is not available when you use the model in combination with the **Excitation Coil** model. In this case, the electric current density is supplied by the **Excitation Coil** model.

Method	Corresponding Physics Value Nodes
None No electric current density source is defined.	None
Specified Specifies a user-defined electric current density source ($\mathbf{J}_u$ in Eqn. (3747)). As in Transverse Magnetic mode the current density is normal to the 2D region, you only specify the normal component of the electric current density.	**Electric Current Density Source** Allows you to specify the normal component of the electric current density as a scalar profile. **Electric Current Density Source Potential Derivative** When the current source is a nonlinear function of magnetic vector potential, specifying the derivative of the electric current density with respect to the magnetic vector potential improves the convergence.

Interface Settings

Magnetic Vector Potential Z Periodicity

In 2D domains, allows you to turn periodic and repeating interfaces into anti-periodic interfaces (see Boundary and Interface Conditions).

For example, in electrical machine problems, the cross-sectional field analysis can be reduced to either an odd number or an even number of poles by employing anti-periodic or periodic interface conditions, respectively.

Method	Corresponding Physics Value Nodes
Periodic The magnetic potential has the same sign at each side of the interface (see Eqn. (3756)).	None
Anti Periodic The magnetic potential has the opposite sign at each side of the interface (see Eqn. (3755)).	None

where p is the even or odd number of pole pairs, and r and θ are polar coordinates.

Transverse Magnetic Potential Solver

Solves for the transverse magnetic potential according to Eqn. (3752).

Under-Relaxation Factor

During the iterative process, the **Transverse Magnetic Potential** solver under-relaxes the changes in the solution, as:

$$\mathbf{A}^{n+1} = \mathbf{A}^{n} + \omega \mathbf{A}'$$

where ω is the under-relaxation factor. At each iteration, the under-relaxation factor governs the extent to which the new solution replaces the old one. The default value is 1.0.

If residuals show solution divergence and do not decrease, reduce the under-relaxation factor.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this option for debugging purposes, or to keep one set of equations fixed while you solve another set.

Reconstruction Freeze

Specifies whether reconstruction gradients are updated with each iteration. When deactivated, the solver updates the reconstruction gradients with each iteration. When you activate both this property and *Temporary Storage Retained*, the solver uses the reconstruction gradients from the last iteration in which they were updated.

Reconstruction Zeroed

Allows you to set reconstruction gradients to zero. When activated, the solver sets reconstruction gradients to zero at the next iteration. Gradients remain zeroed.

When deactivated, the solver does not reset reconstruction gradients. The gradients are recomputed at the next iteration if they have been zeroed.

Temporary Storage Retained

When activated, the solver retains the following data, computed at each iteration, and makes them available as field functions:

- **Magnetic Vector Potential Correction-Z Ap Coefficient**
- **Magnetic Vector Potential Correction-Z Residual**
- **Magnetic Vector Potential-Z Gradient**
- **Magnetic Vector Potential-Z Recon**

Permeability Solver

Controls the solution of the magnetic permeability in all the continua that include either the **Finite Volume Magnetic Vector Potential** model or the **Transverse Magnetic Potential** model.

Reports

Magnetic Flux Linkage

Evaluates the flux linkage through a specified surface. This report allows you to obtain the flux linkage that is induced into a coil. Specify the regions that you want to include in the report. Optionally, activate **Smooth Values** to interpolate reported values to the mesh vertices. By default, this option is deactivated and the report uses the values in the closest cell centroid.

Field Functions

Magnetic Vector Potential-Z

Vector field function that represents the transverse magnetic vector potential, that is, the component of $\mathbf{A}$ normal to the 2D domain. See A_z in Eqn. (3752).

Magnetic Flux Density

Vector field function that represents the magnetic flux density, $\mathbf{B}$, which is related to the magnetic vector potential, $\mathbf{A}$, through Eqn. (3698).

Magnetic Field

Vector field function that represents the magnetic field $\mathbf{H}$, which is related to the magnetic flux density, $\mathbf{B}$, through Eqn. (3688) or Eqn. (3691).

Electric Current Density-Z

Vector field function that represents the component of the electric current density $\mathbf{J}$ normal to the 2D domain.

Permeability

Scalar field function that represents the magnetic permeability μ in Eqn. (3688) or Eqn. (3691).

Electromechanical Stress Tensor

See Eqn. (3769) and Eqn. (3770).

Electromagnetic Force Density

See Eqn. (3772).

For information on the **Transverse Magnetic Potential** solver field functions, see Temporary Storage Retained.

Harmonic Balance FV Transverse Magnetic Potential Model Reference

The **Harmonic Balance FV Transverse Magnetic Potential** model allows you to model transverse-magnetic modes (that is, magnetic fields that lie on a 2D domain) with single harmonic time dependence using the finite volume approach.

The model solves for the complex phasor representation of the transverse magnetic potential (see Eqn. (3753)). The magnetic flux $\widehat{\mathbf{B}}$ and the magnetic field $\widehat{\mathbf{H}}$ are calculated from Eqn. (3721). You can use the **Harmonic Balance FV Transverse Magnetic Potential** model in combination with the **Excitation Coil** model (see Excitation Coil Model Reference).

When using the **Harmonic Balance FV Transverse Magnetic Potential** model, you define most quantities (including boundary and region values) using complex profiles. For more information on defining complex profiles, see Complex Quantities.

Model Name	**Harmonic Balance FV Transverse Magnetic Potential**	
Theory	See Harmonic Time Dependence and Transverse Magnetic Modes.	
Provided By	**[physics continuum]** > **Models** > *Electromagnetism*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Harmonic Balance FV Transverse Magnetic Potential**	
Requires	*Space*: one of **Two Dimensional**, **Axisymmetric** *Time*: one of **Steady**, **Implicit Unsteady** *Material*: one of **Gas**, **Liquid**, **Solid** *Electromagnetism*: **Electromagnetism**	
Activates	Material Properties	**Electrical Conductivity**, **Magnetic Permeability**. See Material Properties.
	Reference Values	**Excitation Frequency**. See Reference Values.
	Initial Conditions	**Magnetic Vector Potential Z**. See Initial Conditions.
	Boundary Inputs	**Magnetic Vector Potential Specification**. See Boundary Settings.

	Region Inputs	**Electric Current Density Source Option**, **Current Conservation Option**. See Region Settings.
	Solvers	**Harmonic Balance FV Transverse Magnetic Potential** (uses the **AMG Linear Solver**.
	Monitors	**Imaginary Transverse Magnetic Potential**, **Real Transverse Magnetic Potential**.
	Field Functions	**Magnetic Vector Potential-Z** (Imag, Real, Phase, Magnitude), **Magnetic Flux Density** (Imag, Real, Phase, Magnitude), **Magnetic Field** (Imag, Real, Phase, Magnitude), **Electric Current Density-Z** (Imag, Real, Phase, Magnitude), **Permeability**, **Electrical Conductivity** (Imag, Real, Phase, Magnitude). See Field Functions.

Reference Values

Excitation Frequency

Allows you to specify the frequency f_0 in Eqn. (3710). STAR-CCM+ requires the frequency to be the same across contact or baffle interfaces. When you associate different continua to contacting regions, make sure that you set the same frequency in both continua.

Material Properties

Magnetic Permeability

Allows you to define the magnetic permeability μ in Eqn. (3753) using a constant or a scalar field function.

Electrical Conductivity

Defines σ in Eqn. (3717) as a complex profile.

Initial Conditions

Magnetic Vector Potential Z

Allows you to initialize the transverse magnetic potential $\widehat{A}_z$ (see Eqn. (3753)) to a specified complex profile.

Boundary Settings

Magnetic Vector Potential Specification

In transverse magnetic mode, the magnetic vector potential and electric current sheet are normal to the 2D domain. As only the normal component is non-zero, you specify the magnetic vector potential and the current sheet at a boundary using scalar profiles (see Boundary and Interface Conditions). For harmonic time dependence, these quantities are specified as complex profiles.

Method	Corresponding Physics Value Nodes
Electric Current Sheet Z Sets the electric current sheet at the boundary to: $$\mathbf{J}_S = \widehat{J}_S \mathbf{n}$$ where $\mathbf{n}$ is the boundary normal and $\widehat{J}_S$ is a specified complex profile.	**Electric Current Sheet** Defines the complex profile, $\widehat{J}_S$.
Magnetic Vector Potential Z Sets the transverse magnetic vector potential at the boundary to: $$\mathbf{A} = \widehat{A}\mathbf{n}$$ where $\mathbf{n}$ is the boundary normal and $\widehat{A}$ is a specified complex profile.	**Magnetic Vector Potential** Defines the complex profile, $\widehat{A}$.

Region Settings

Applies to fluid, porous, and solid regions.

Current Conservation Option

When activated, STAR-CCM+ calculates the transverse magnetic potential while enforcing electric current conservation within the region. See Eqn. (3754). Deactivate this option when you explicitly specify the transverse magnetic potential at the outer boundaries of conducting regions (that is, when you use the **Magnetic Vector Potential Z** boundary condition).

Only available for unsteady simulations.

Electric Current Density Source Option

Allows you to specify the electric current density for the region. This option is not available when you use the model in combination with the **Excitation Coil** model. In this case, the electric current density is supplied by the **Excitation Coil** model.

Method	Corresponding Physics Value Nodes
None No electric current density source is defined.	None
Specified Specifies a user-defined electric current density source ($\mathbf{J}_u$ in Eqn. (3747)).	**Complex Electric Current Density Source** Allows you to specify the electric current density as a complex profile.

Field Functions

Magnetic Vector Potential-Z (Imag, Real, Phase, Magnitude)

Represent the imaginary part, real part, phase, and magnitude of the complex magnetic vector potential $\widehat{A}_z$ (see Eqn. (3711)).

Electric Current Density-Z (Imag, Real, Phase, Magnitude)

Represent the imaginary part, real part, phase, and magnitude of the complex electric current density $\widehat{J}_z$. The electric current density magnitude, $\left|\widehat{J}_z\right|$, is also referred to as the peak value.

Magnetic Flux Density (Imag, Real, Phase, Magnitude)

Represent the imaginary part, real part, phase, and magnitude of the complex magnetic flux density $\widehat{\mathbf{B}}$ (see Eqn. (3721)).

Magnetic Field (Imag, Real, Phase, Magnitude)

Represent the imaginary part, real part, phase, and magnitude of the complex magnetic field $\widehat{\mathbf{H}}$ (see Eqn. (3721)).

Permeability

Represents the magnetic permeability μ_0 in Eqn. (3753).

Electrical Conductivity (Imag, Real, Phase, Magnitude)

Represent the imaginary part, real part, phase, and magnitude of the complex electrical conductivity σ in Eqn. (3753).

Permanent Magnet Model Reference

The **Permanent Magnet** model allows you to model the effects of the magnetic fields induced by permanent magnets.

Model Name	**Permanent Magnet**
Theory	See Permittivity and Permeability and Source Terms.
Provided By	**[physics continuum]** > **Models** > *Optional Models*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Permanent Magnet**
Requires	*Space*: one of **Two Dimensional**, **Three Dimensional** *Time*: one of **Steady**, **Implicit Unsteady** *Material*: one of **Solid**, **Multi-Component Solid** > **Multi-Part Solid** *Optional Models*: **Electromagnetism** *Electromagnetism*: one of **Finite Element Magnetic Vector Potential**, **Finite Volume Magnetic Vector Potential**, **Transverse Magnetic Potential**

Activates	Model Controls (child nodes)	**Magnetization Direction**. See Model Controls (Child Nodes).
	Material Properties	**Remanence Flux Density**. See Material Properties.
	Region Inputs	**Magnetization Direction**. See Region Settings.
	Field Functions	**Remanent Magnetic Flux Density**. See Field Functions.

Model Controls (Child Nodes)

Magnetization Direction

Allows you to specify the unit vector that defines the direction of the remanence flux density of the magnet (see Remanence Flux Density). STAR-CCM+ applies the specified direction to all the regions (associated with the continuum) for which the **Magnetization Direction** has *Method* set to **Use Continuum Values**. See Region Settings.

Material Properties

Remanence Flux Density

Specifies the magnetic flux density that remains in a permanent magnet after any external magnetic field is removed ($\mathbf{B}_r$ in Eqn. (3748) and in the B-H Curve). You specify the remanence flux density as a scalar, with either the constant or field function method. The material database allows you to store the remanence flux density value for any material. See Modeling Permanent Magnets.

The direction of magnetization is specified separately. See Magnetization Direction.

Region Settings

Applies to solid regions.

Magnetization Direction

Allows you to specify the unit vector that defines the direction of the remanence flux density.

Method	Corresponding Physics Value Nodes
Use Continuum Values For this region, STAR-CCM+ uses the magnetization direction specified under the **[physics continuum]** > **Models** > **Permanent Magnet** > **Magnetization Direction** node.	None

Specify Region Values

Allows you to specify the magnetization direction for this region. STAR-CCM+ ignores any direction specified under the **[physics continuum]** > **Models** > **Permanent Magnet** > **Magnetization Direction** node.

Magnetization Direction

Specifies the unit vector that defines the direction of the remanence flux density.

Field Functions

Remanent Magnetic Flux Density

Corresponds to $\mathbf{B}_r$ in Eqn. (3748).

Excitation Coil Model Reference

The **Excitation Coil** model allows you to model the electric current density that is produced by an excitation coil in solid materials.

STAR-CCM+ calculates the coil electric current density from Eqn. (3759) and adds it as a source term in the magnetic vector potential equation Eqn. (3706). STAR-CCM+ also calculates the specific magnetic flux linkage according to Eqn. (3763).

In transient simulations, the **Excitation Coil** model also computes the specific electromotive force.

Model Name	**Excitation Coil**	
Theory	See Excitation Coils.	
Provided By	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Excitation Coil**	
Requires	Physics Models: *Space*: one of **Two Dimensional**, **Three Dimensional** *Time*: either **Steady**, or **Implicit Unsteady** with **Quasi Unsteady Losses** *Material*: one of **Solid**, **Multi-Component Solid** > **Multi-Part Solid-** *Optional Models*: **Electromagnetism** *Electromagnetism*: one of **Finite Element Magnetic Vector Potential**, **Finite Volume Magnetic Vector Potential**, **Transverse Magnetic Potential**	
Activates	Region Inputs	**Electric Current Density Magnitude**, **Excitation Coil Reference Cross Section**, **Local Direction Field**. See Region Settings.
	Report Options	**Excitation Coil Conductor Resistance**. See Reports.
	Field Functions	**Specific Electromotive Force**, **Specific Magnetic Flux Linkage**, **Local Direction Field**. See Field Functions.

Region Settings

Applies to solid regions.

Electric Current Density Magnitude

Specifies the parameters that STAR-CCM+ uses to calculate $\mathbf{J}_{\mathrm{coil}}$ (see Eqn. (3759)).

STAR-CCM+ calculates the cross-section of the coil, a_0, with respect to a specified reference area, a_{ref}, which can be the actual cross section, or the area that is obtained by cutting the coil with an arbitrary section (see Eqn. (3757)).

To set the quantities in Eqn. (3759), you use the following child nodes:

- **Ampere Turn**: specifies the number of turns n_{t} of the coil and the number of sections n_{s} that cut though the coil region. When the reference area is equal to the actual cross section of the coil, $n_{\mathrm{s}} = 1$.
  - **Electric Current**: specifies the electric current I that goes though an individual wire as a scalar profile. Together with the common methods for setting scalar profiles, STAR-CCM+ provides an additional method called **Circuit Element**. This method allows you to select a circuit element and use the current through this element as the input for the electric current inside the excitation coil.
  - **Conductor Area**: specifies the area that is covered by conducting material. You can enter an absolute value or a relative value (a percentage of the total coil area). STAR-CCM+ uses the specified value to calculate the direct current resistance of the coil (see Eqn. (3765)).

In three-dimensional analyses that use either the **Finite Volume Magnetic Vector Potential** or the **Finite Element Magnetic Vector Potential** model, the magnitude of the electric current density is calculated with respect to the reference area specified in the **Excitation Coil Reference Cross Section** node. In two-dimensional simulations that use the **Transverse Magnetic Potential** model, the magnitude of the electric current density is calculated with respect to the 2D region area, which is computed automatically.

Excitation Coil Reference Cross Section

In three-dimensional simulations that use either the **Finite Volume Magnetic Vector Potential** or the **Finite Element Magnetic Vector Potential** model, you specify the reference area a_{ref} (see Eqn. (3757)) by selecting an appropriate reference part. The reference part can be an internal interface boundary or one of the following derived parts that cuts through the coil region:

- plane section
- constrained plane section
- cylinder section
- arbitrary section

Excitation Coil Reference Height

In two-dimensional simulations that use the **Transverse Magnetic Potential** model, the cross section a_0 is the area of the 2D domain. You define the volume over which the specific magnetic flux linkage is integrated by specifying a reference height for the 2D region.

Local Direction Field

Sets the direction field $\mathbf{f}$ in Eqn. (3759).

In three-dimensional analyses that use either the **Finite Volume Magnetic Vector Potential** or the **Finite Element Magnetic Vector Potential** model, the available properties are:

- *Method*: the **Feature Curve** option allows you to define the field orientation using a closed loop.

- *Curve Parts*: allows you to select the feature curve that represents the local field orientation. To represent the local field orientation, select a closed loop or a curve that extends to the ends of the simulation domain. Selecting all the feature curves of the coil CAD model does not give a correct representation.
- *Representation*: allows you to choose between the available geometry and mesh representations.
- **Direction**: sets the current flow direction to either **Forward** or **Backward**. This setting is relative to the orientation of the specified feature curve.

In two-dimensional analyses that use the **Transverse Magnetic Potential** model, the **Local Direction Field** lies along the Z-axis. Specify the sign by setting the *Direction* property to the appropriate value, **Forward**, or **Backward**.

Reports

Excitation Coil Conductor Resistance

Evaluates the direct current resistance of the excitation coil region (see Eqn. (3765)).

When you create this report, specify the regions that you want to include in the report.

Field Functions

Specific Electromotive Force

Corresponds to the specific electromotive force ρ_ϵ defined in Eqn. (3764).

Specific Magnetic Flux Linkage

Corresponds to the specific magnetic flux linkage ρ_ψ defined in Eqn. (3763).

Local Direction Field

Corresponds to the local direction field $\mathbf{f}$ defined in Eqn. (3757) and Eqn. (3759).

Quasi Unsteady Losses Model Reference

The **Quasi Unsteady Losses** model allows you to neglect the eddy currents when computing the magnetic vector potential.

Activating the **Quasi Unsteady Losses** model with either the **Finite Volume Magnetic Vector Potential**, the **Finite Element Magnetic Vector Potential**, or the **Transverse Magnetic Potential** model, allows you to calculate the magnetic vector potential in a quasi-unsteady manner. In quasi-unsteady analyses, you can define time-dependent quantities, while neglecting the transient term (that is, the eddy currents) in the magnetic vector potential equation (see Magnetic Vector Potential Models).

In transient analyses, activate this model in all physics continua that are associated with non-conducting regions.

Model Name	**Quasi Unsteady Losses**
Theory	See Magnetic Vector Potential Models.
Provided By	**[physics continuum]** > **Models** > *Optional Models*

Example Node Path	**Continua** > **Physics 1** > **Models** > **Quasi Unsteady Losses**
Requires	Physics Models: *Space*: one of **Three Dimensional**, **Two Dimensional**, **Axisymmetric** *Time*: **Implicit Unsteady** *Material*: one of **Gas**, **Liquid**, **Solid**, **Eulerian Multiphase**, **Multi-Component Gas**, **Multi-Component Liquid**, **Multi-Component Solid** > **Multi-Part Solid** *Electromagnetism*: **Electromagnetism** *Electromagnetism*: one of **Finite Volume Magnetic Vector Potential**, **Finite Element Magnetic Vector Potential**, **Transverse Magnetic Potential**

Excitation Coil Lumped Parameter Model Reference

The **Excitation Coil Lumped Parameter** model allows you to extract the lumped parameters of a coil region, that is, the coil resistance and inductance.

When you use this model in combination with the **Circuit Model**, you can define the load on excitation coil circuit elements using the extracted lumped parameters. See Adding Circuit Elements to a Circuit.

Model Name	**Excitation Coil Lumped Parameter**	
Provided By	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Excitation Coil Lumped Parameter**	
Requires	Physics Models: *Space*: one of **Two Dimensional**, **Three Dimensional** *Time*: **Implicit Unsteady** *Material*: one of **Solid**, **Multi-Component Solid** > **Multi-Part Solid** *Optional Models*: **Electromagnetism** *Electromagnetism*: one of **Finite Element Magnetic Vector Potential**, **Transverse Magnetic Potential** *Optional Models*: **Excitation Coil**	
Activates	Reports	When used together with the **Finite Element Magnetic Vector Potential** model, the **Excitation Coil Lumped Parameter** model activates the following reports: **Magnetic Flux Linkage** **Magnetic Inductance** See Reports.
	Field Functions	**Excitation Coil Load**, **Specific Magnetic Flux Linkage**, **Magnetic Vector Potential**.

Reports (Finite Element Only)

The following reports are available only when you use the **Excitation Coil Lumped Parameter** model together with the **Finite Element Magnetic Vector Potential** model.

Magnetic Flux Linkage

Calculates the magnetic flux linkage in the coil region (see Eqn. (3760)).

Magnetic Inductance

Calculates either the self-inductance of a coil region (or group of regions), or the mutual inductance between two coil regions (or groups of regions). To calculate the self-inductance, you specify the relevant coil regions using the *Parts* property, leaving the *Parts Mutual* property blank. To calculate the mutual inductance, you specify the first coil region (or group of regions) using the *Parts* property and the second coil region (or group of regions) using the *Parts Mutual* property

Field Functions

[Region_Name] Excitation Coil Load

Represents the current load in the coil region.

[Region_Name] Specific Magnetic Flux Linkage

Corresponds to the coil region inductance.

[Region_Name] Magnetic Vector Potential

Represents the magnetic vector potential in the coil region.

Modified Steinmetz Model Reference

The Modified Steinmetz model allows you to model hysteresis and eddy-current losses.

STAR-CCM+ calculates the losses using a modified Steinmetz formula, based on magnetic flux density calculations (see Eqn. (3766)). You can use the loss terms for thermal analysis.

Model Name	**Modified Steinmetz**	
Theory	See Power Losses.	
Provided By	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Modified Steinmetz**	
Requires	*Space*: one of **Two Dimensional**, **Three Dimensional** *Time*: **Implicit Unsteady** *Material*: one of **Solid**, **Multi-Component Solid** > **Multi-Part Solid** *Electromagnetism*: one of **Finite Element Magnetic Vector Potential**, **Finite Volume Magnetic Vector Potential**, **Transverse Magnetic Potential**	
Activates	Material Properties	**Eddy-Current Loss Coefficient**, **Hysteresis Loss Coefficient**, **Steinmetz Coefficient A**, **Steinmetz Coefficient B**. See Material Properties.
	Monitors	**Modified Steinmetz**. See Modified Steinmetz Monitor.
	Field Functions	**Eddy-Current Loss Coefficient**, **Hysteresis Loss Coefficient**, **[SteinmetzMonitorFF Maximum]**, **[SteinmetzMonitorFF Minimum]**, **[SteinmetzMonitorFF Peak]**, **[SteinmetzMonitorFF Mean Square Time**

		Derivative], **[SteinmetzMonitor Eddy-Current Loss]**, **[SteinmetzMonitor Hysteresis Loss]**, **[SteinmetzMonitor Total Loss]**, **Steinmetz Coefficient A**, **Steinmetz Coefficient B**. See Field Functions.

Material Properties

Applies to solid materials.

Eddy-Current Loss Coefficient

The eddy-current loss coefficient C_e in Eqn. (3766).

Hysteresis Loss Coefficient

The hysteresis loss coefficient C_h in Eqn. (3766).

Steinmetz Coefficient A

a in Eqn. (3766).

Steinmetz Coefficient B

b in Eqn. (3766).

Modified Steinmetz Monitor

Allows you to monitor the peak value and the average time derivative of a specified field function. By default, the field function is the magnetic flux density. This monitor type appears under the **New Monitor** right-click action for the **Monitors** node. For this monitor, you specify the following properties:

Parts

Specifies the input parts where you monitor the peak value and the average time derivative of the **Modified Steinmetz** monitor field function.

Enabled

Activates/deactivates the **Modified Steinmetz** monitor.

Field Function

Specifies the field function for which the peak value and the derivative are calculated. The associated maximum, minimum, peak, and time derivative field functions appear as read-only properties. STAR-CCM+ sets the name of the associated field functions based on the specified function. See Field Functions.

Excitation Frequency

Specifies the excitation frequency f, as defined in Eqn. (3766), as a scalar profile.

As for all STAR-CCM+ monitors with time-step trigger, you specify the frequency of update in terms of the time-step, using the **Time-Step Frequency** child node.

Field Functions

Eddy-Current Loss Coefficient

Corresponds to the coefficient C_e in Eqn. (3766).

Hysteresis Loss Coefficient

Corresponds to the coefficient C_h in Eqn. (3766).

Modified Steinmetz Monitor Field Functions

The following field functions are associated with the Steinmetz monitor (see Modified Steinmetz Monitor). The name of these field functions change based on the field function that you specify for the Steinmetz monitor. As by default the field function specified for the monitor is the **Magnetic Flux Density: Magnitude**, the associated field functions are: **Magnetic Flux Density: Magnitude Maximum**, **Magnetic Flux Density: Magnitude Minimum**, **Magnetic Flux Density: Magnitude Peak**, **Magnetic Flux Density: Magnitude Mean Square Time Derivative**.

- **[SteinmetzMonitorFF Maximum]**—corresponds to the maximum value of the field function that you select for the Steinmetz monitor. See Eqn. (3767).
- **[SteinmetzMonitorFF Minimum]**—corresponds to the minimum value of the field function that you select for the Steinmetz monitor. See Eqn. (3767).
- **[SteinmetzMonitorFF Peak]**—corresponds to the peak of the field function that you select for the Steinmetz monitor. See Eqn. (3767).
- **[SteinmetzMonitorFF Mean Square Time Derivative]**—corresponds to the average time derivative of the field function that you select for the Steinmetz monitor. See Eqn. (3768).

The following field functions are also associated with the Steinmetz monitor. The name of these field functions change based on the monitor name. By default, the monitor name is **Modified Steinmetz 1**, so the default names are **Modified Steinmetz 1 Eddy-Current Loss**, **Modified Steinmetz 1 Hysteresis Loss**, and **Modified Steinmetz 1 Total Loss**.

- **[SteinmetzMonitor Eddy-Current Loss]**—corresponds to the losses by eddy-currents
- **[SteinmetzMonitor Hysteresis Loss]**—corresponds to the losses by hysteresis
- **[SteinmetzMonitor Total Loss]**—corresponds to the total core losses W

Steinmetz Coefficient A

Corresponds to the coefficient a in Eqn. (3766).

Steinmetz Coefficient B

Corresponds to the coefficient b in Eqn. (3766).

Ohmic Heating Model Reference

The **Ohmic Heating** model solves for the heat that is generated in a conducting material due to the flow of electric current. This effect is often referred to as the Joule effect.

When using this model, STAR-CCM+ calculates the energy dissipation in an electrically conducting material from Eqn. (3774) and Eqn. (3777) and then adds it as a source term in the energy equation (Eqn. (2076) or Eqn. (2079)).

Model Name	**Ohmic Heating**
Theory	See Ohmic Heating.
Provided By	**[physics continuum]** > **Models** > *Optional Models*

Example Node Path	**Continua** > **Physics 1** > **Models** > **Ohmic Heating**	
Requires	*Space*: one of **Three Dimensional**, **Two Dimensional**, **Axisymmetric** *Time*: one of **Steady**, **Implicit Unsteady** *Material*: one of **Gas**, **Liquid**, **Eulerian Multiphase**, **Solid**, **Multi-Component Solid** > **Multi-Part Solid** *Optional Models*: one of the energy models, for example, **Segregated Solid Energy** for a solid and **Segregated Fluid Temperature** for a fluid *Optional Models*: **Electromagnetism** *Electromagnetism*: one or more of **Electrodynamic Potential**, **Finite Volume Magnetic Vector Potential**, **Finite Element Magnetic Vector Potential**, **Harmonic Balance FV Magnetic Vector Potential**, **Transverse Magnetic Potential**, **Harmonic Balance FV Transverse Magnetic Potential**	
Activates	Field Functions	**Specific Ohmic Heat Source**, **Specific Ohmic Heat Surface Source**. See Field Functions.

Field Functions

Specific Ohmic Heat Source

See Eqn. (3774).

Specific Ohmic Heat Surface Source

See Eqn. (3777).

One-Way Coupled MHD Model Reference

The **One-Way Coupled MHD** model accounts for the interaction between an electrically conducting fluid and the magnetic field. You define the magnetic field by prescribing the magnetic flux density within the fluid region.

Model Names	**One-Way Coupled MHD**	
Theory	See Magnetohydrodynamics (MHD).	
Provided By	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **One-Way Coupled MHD**	
Requires	*Space*: one of **Three Dimensional**, **Axisymmetric** *Time*: one of **Steady**, **Implicit Unsteady** *Material*: one of **Gas**, **Liquid**, **Eulerian Multiphase**, **Multi-Component Gas**, **Multi-Component Liquid** *Optional Models*: **Electromagnetism** *Electromagnetism*: **Electrodynamic Potential**	
Activates	Region Inputs	**Prescribed Magnetic Flux Density**. See Region Settings.

	Field Functions	**Electric Current Density**, **Lorentz Force Density**, **Magnetic Flux Density**. See Field Functions.

Region Settings

Apply to fluid regions.

Prescribed Magnetic Flux Density

Defines the magnetic flux density $\mathbf{B}_p$ as a vector profile (see Eqn. (3780)).

Field Functions

Electric Current Density

See Eqn. (3780).

Lorentz Force Density

See Eqn. (3784).

Magnetic Flux Density

See Eqn. (3780).

Two-Way Coupled MHD Model Reference

The **Two-Way Coupled MHD** model accounts for the interaction between an electrically conducting fluid and the magnetic field. STAR-CCM+ calculates the magnetic field within the fluid region from the magnetic vector potential.

Model Names	**Two-Way Coupled MHD**	
Theory	See Magnetohydrodynamics (MHD).	
Provided By	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Two-Way Coupled MHD**	
Requires	*Space*: one of **Three Dimensional**, **Axisymmetric** *Time*: one of **Steady**, **Implicit Unsteady** *Material*: one of **Gas**, **Liquid**, **Eulerian Multiphase**, **Multi-Component Gas**, **Multi-Component Liquid** *Optional Models*: **Electromagnetism** *Electromagnetism*: one of **Finite Volume Magnetic Vector Potential**, **Finite Element Magnetic Vector Potential**, **Transverse Magnetic Potential**, **Harmonic Balance FV Magnetic Vector Potential**, **Harmonic Balance FV Transverse Magnetic Potential**	
Activates	Field Functions	**Electric Current Density**, **Lorentz Force Density**, **Magnetic Flux Density**. See Field Functions.

Field Functions

Electric Current Density

See Eqn. (3782).

Lorentz Force Density

See Eqn. (3784).

Magnetic Flux Density

See Eqn. (3782).

Using the Circuit Model

The Circuit model lets you model a circuit with specified circuit elements and circuit connections. Selecting this model activates the circuit solver in the physics continuum.

The Circuit model is a required model in battery simulations and is selected automatically when you activate the Battery model. The Circuit model is an optional model that is available for suitable electromagnetic simulations. The Circuit model allows you to create electrical circuits and compute the electrical solution for them.

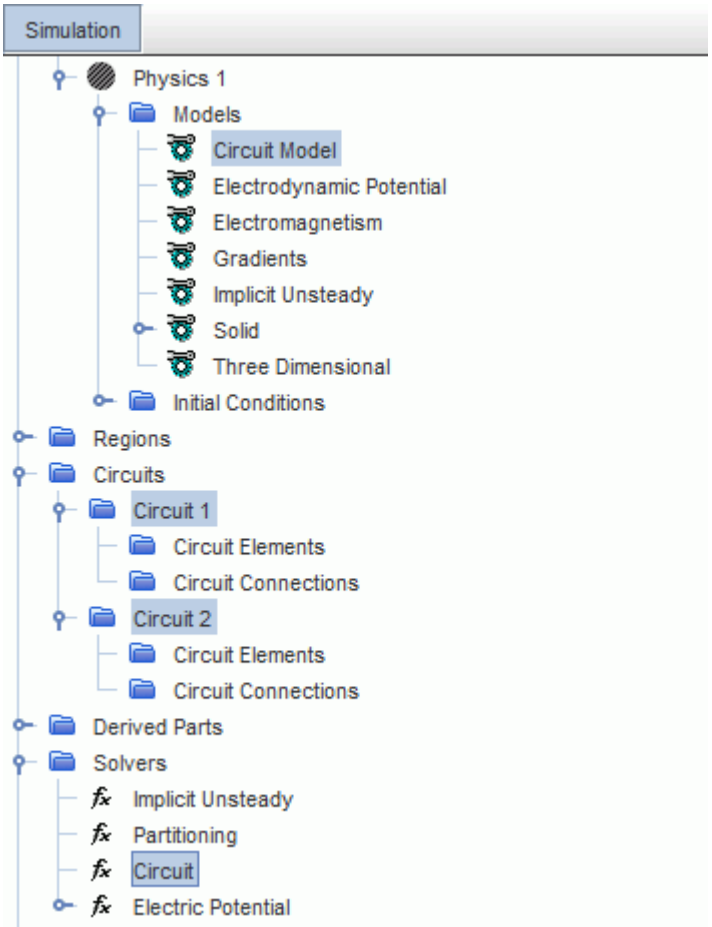
To select the Circuit model:

1. In the *Physics 1 Model Selection* dialog, select the following models in order:

Group Box	Model
Time	**Implicit Unsteady**
Space	The appropriate space model.
Material	The appropriate material model.
Optional Models	**Electromagnetism**
Electromagnetism	The appropriate Electromagnetism model.
Optional Models	**Circuit Model**

2. Select any other models that you require from the *Optional Models* group box.
3. **Close** the *Physics 1 Model Selection* dialog.

The list of selected models can be seen by expanding the **Models** node in the physics continuum.



The Circuit Model has no properties, but it activates the **Circuits** > **[Circuit Name]** and **Solvers** > **Circuit** nodes where you set the circuit solver properties. You can create multiple circuits and set the circuit solver properties independently for each circuit: the elements and connections for each circuit are stored in the corresponding **[Circuit Name]** node under the **Circuits** node.

When you create a circuit element, the corresponding field function **Current <Circuit Element Name> Circuit Element** is made available under the **Tools** > **Field Functions** node. This field function represents the electric current through the circuit element.

Accessing Electromagnetism Constants through the Java API

The Java API defines several compile-time constant expressions, which are common for electrodynamics. These constants can be accessed in a Java macro.

star.electromagnetism.commom.ElectromagnetismConstants		
public static final double	AVOGADRO_CONSTANT	6.022136736E23 l/kmol
public static final double	ELECTRIC_CONSTANT (permittivity)	8.854187817620389E-12 F/m
public static final double	ELEMENTARY_CHARGE	1.602176487E-19 C
public static final double	FARADAY_CONSTANT	9.648525879918127E7 C/kmol
public static final double	MAGNETIC_CONSTANT (permeability)	1.2566370614359173E-6 H/m
public static final double	SPEED_LIGHT	2.99792458E8 m/s

These constants can be invoked inside Java macros. For example, the following command can be used to access the value of the speed of light in a vacuum:

```
double speedOfLight =
star.electromagnetism.common.ElectromagnetismConstants.SPEED_LIGHT;
```

Troubleshooting

This section contains troubleshooting information.

The following sections provide the solutions to common issues:

Meshing recommendation

The use of an extruder mesh at the boundaries can improve the results. Additional guidelines on mesh quality are available in the chapter on checking the mesh.

Electric potential solution converges slowly

The under-relaxation factor can be adjusted to control the convergence rate. The default Under-Relaxation factor is set to 1.0 and should be kept at this value whenever possible. However, if the solution does not converge, a reduction of the under-relaxation factor can stabilize the convergence procedure.

A message appears about maximum temperature

`Max. Temperature limited to ...`—This message does not necessarily mean that the solution is diverging. See the previous note about what to look for.

Electric potential solution is diverging for Electro-Deposition Coating

What to look at in electro-deposition coating:

- Reduce the time-step size.
- A reduction of the under-relaxation factor for the Electro-Deposition Coating Solver can stabilize the convergence procedure even without changing the under-relaxation factor of the Electric Potential Solver. On the other hand, a higher under-relaxation factor can accelerate the convergence rate.

Modeling Electrochemistry

Electrochemistry is the study of chemical reactions which occur due to an imposed electrical charge, or a difference in electrical potential at a boundary between a conductor–such as a metal–and an electrolyte. STAR-CCM+ provides models that allow you to simulate Li-Ion batteries, corrosion, etching, and other electrochemical reactions.

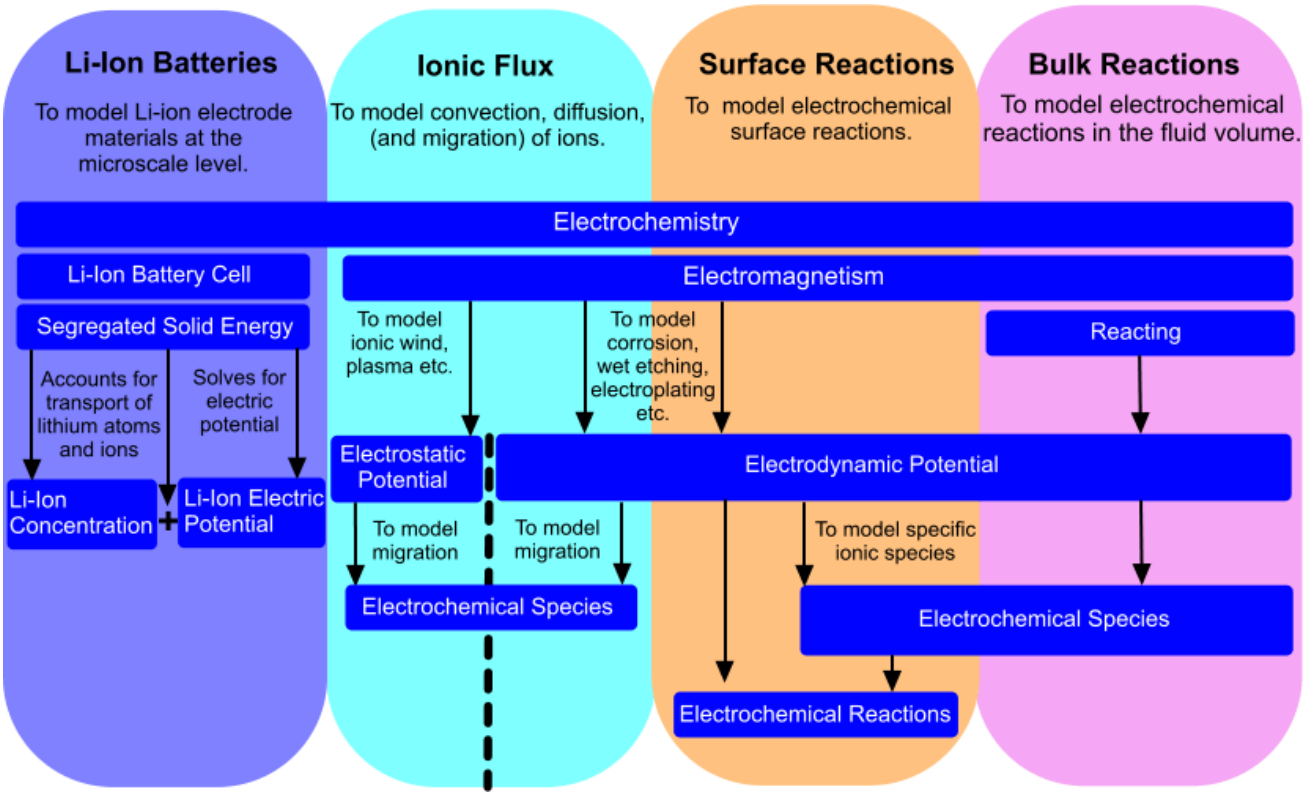
Typically, during electrochemical reactions, electrons are produced or consumed at the boundary between a metal and an electrolyte. Metal and hydrogen atoms can lose electrons to become positive ions, and non-metal atoms can gain electrons to become negative ions. Both negative and positive ions are free to move through an electrolyte solution, however, electrons cannot move through an electrolyte solution—only through a conductor such as a metal. The rate of electrochemical reactions and the transport of ions is directly related to the magnitude of the electrical charge that is imposed, or the difference in electric potential across the boundary.

You can use STAR-CCM+ to model a wide variety of electrochemical applications and simulate behavior such as, movement of ions or specific ionic species, ionic charge density, and accumulation of product species.

STAR-CCM+ supports the following general categories of electrochemical modeling:

- Li-Ion Battery Cells
- Ionic Species Flux
- Electrochemical Surface Reactions
- Bulk Ion Chemical Reactions

The following diagram shows an overview of these general workflows with examples of some common areas of electrochemistry which relate to each workflow, and the corresponding electrochemistry models that are required in STAR-CCM+.



Contents:

Electrochemistry Applications
Li-Ion Battery Cells
Ionic Species Flux
Electrochemical Surface Reactions
Bulk Ion Chemical Reactions
Electro-Deposition Coating
Electrochemistry Formulation
Electrochemistry Bibliography
Electrochemistry Nomenclature

Electrochemistry Applications

The Electrochemistry models that are provided in STAR-CCM+ provide the framework to model many electrochemical applications. This section provides further details about some popular electrochemical applications that can be modeled in STAR-CCM+.

Contents:

Modeling Corrosion
Fuel Cells
Ionic Wind

Corrosion

You can use the Electrodynamic Potential model in STAR-CCM+ to help you to understand and predict the effect of corrosion.

For metal structures that are prone to corrosion, the challenge is to maximize the design life of the structure by preventing corrosion. The ability to understand and predict corrosion behavior using STAR-CCM+ lets you compare corrosion prevention methods and select the optimal solution for each specific scenario—even with complex geometries.

One way to prevent corrosion is using cathodic protection. The concept of cathodic protection is straightforward. However, finding the optimal level of protection is a more complex consideration. Insufficient cathodic protection results in corrosion. Over-protection can result in negative effects such as hydrogen embrittlement, which weakens high-strength steels and increases the likelihood of the structure fracturing.

Using STAR-CCM+, you can optimize the location and size of the sacrificial anodes that prevent corrosion. This process makes sure that the level of cathodic protection that a structure receives is most effective in prolonging the life of the structure. Simulating the cathodic protection of a structure is also cost effective as it prevents wasting expensive anodes unnecessarily.

You can use corrosion modeling in STAR-CCM+ for modeling cathodic protection on:

- ship hulls
- oil rigs and gas pipes
- bridges
- reinforced concrete
- underground structures

To model corrosion, follow the procedure that is set out in Modeling Electrochemical Surface Reactions.

What Is Corrosion?

Corrosion is an electrochemical process which occurs naturally when metals react to form compounds that are more stable than their pure elemental form. During corrosion, a number of electrochemical surface reactions occur at the interface between the metal and the electrolyte. See Electrochemical Surface Reactions.

Metal atoms lose electrons and become ions which are soluble in the electrolyte. When metals lose electrons, the reaction is known as oxidation—an example of an anodic reaction. The following reaction shows the oxidation of iron:

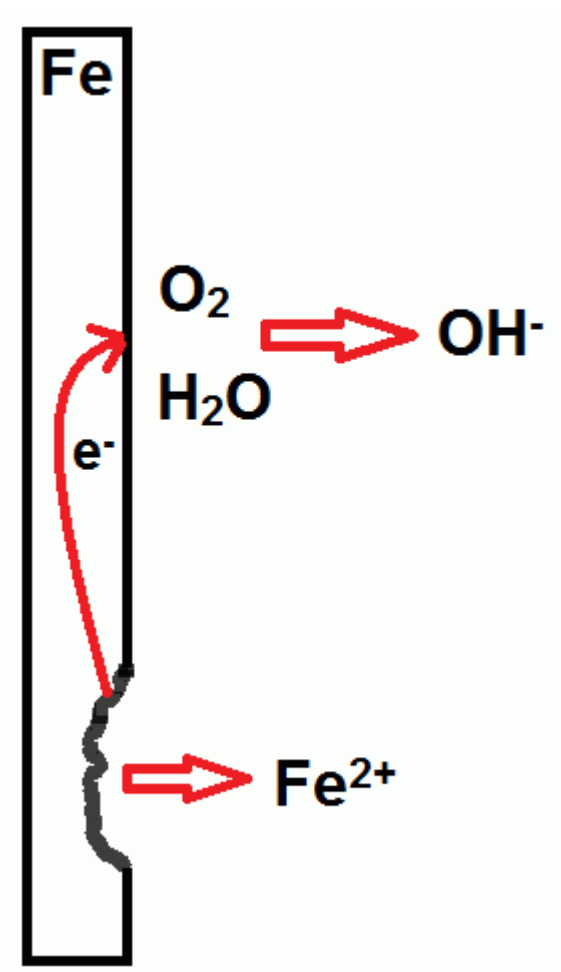
$$Fe \rightleftharpoons Fe^{2+} + 2e^{-}$$

The electrolyte accepts electrons from the metal and forms ions. When substances accept electrons, the reaction is known as reduction—an example of a cathodic reaction. The following reaction shows the reduction of water:

$$2H_2O + 2e^{-} \rightleftharpoons H_2 + 2OH^{-}$$

The ions that are produced during corrosion form compounds that are, in most instances, soluble in the electrolyte. The corrosion rate depends on the potential drop at the interface between the metal and the electrolyte. Overall, corrosion causes metals to break down and so weaken metal structures and reduce their lifespan.

In the diagram below, iron, Fe, is corroding in the presence of water, H_2O, and oxygen, O_2, and losing electrons, e^{-}, to form iron ions, Fe^{2+}. The water and oxygen accept electrons from the iron to form hydroxide ions, OH^{-}:



One method of preventing corrosion is by cathodic protection.

What is Cathodic Protection?

Cathodic protection is a process that is used to prevent a metal from corroding by creating an electrochemical cell in which the protected metal forms the cathode.

There are several ways to create an electrochemical cell for the purpose of cathodic protection, including:

- Actively imposing a direct electrical current on the metal from a power supply
- Using galvanic anodes

Using Galvanic Anodes

When galvanic anodes are used for cathodic protection, metal anodes are attached to the metal structure that requires protection. The anodes are made of a metal with an equilibrium potential that is greater than that of the metal structure. Since the anodes are less noble than the metal, the electrolyte potential is greater near the anodes than near the metal structure. This potential difference results in ion current flow from the anodes to the metal structure. Ions and electrons are produced at the anode, the ions travel through the electrolyte, and electrons travel through the metal circuit. Ions and electrons then recombine at the cathode. The potential drop through the metal circuit is usually much smaller than the potential drop through the electrolyte, since the metal has a much higher conductivity.

The current flow that is supplied to the metal structure prevents the metal structure from undergoing oxidation.

To model cathodic protection, follow the procedure that is set out in Modeling Electrochemical Surface Reactions.

Fuel Cells

Fuel cells produce electrical power from hydrogen and oxygen with by-products of water and heat. Among other uses, fuel cells can be used for transportation and energy storage applications.

Fuel cells convert chemical energy into electrical energy. During electrochemical reactions, there are two electrochemical half reactions that occur—reduction and oxidation. During reduction reactions, reactants gain electrons, and in oxidation reactions, reactants lose electrons. In fuel cells, these half reactions occur at the anodic and cathodic interfaces which are separated by an electrolyte. For example, in solid oxide fuel cells (SOFCs), oxide ions exist in the solid electrolyte membrane and the cathodic reaction produces them on one side of the electrolyte. It is generally assumed that the concentration of oxide ions does not vary in concentration through the electrolyte membrane, however, the oxide ions are driven across the electrolyte membrane due to the electric field. The anodic reaction on the other side of the electrolyte consumes the oxide ions. By separating the electrochemical half reactions (on either side of the electrolyte), fuel cells can therefore harness the transport of species to generate electrical power.

These electrochemical reactions occur at the interface between fluid and solid phases. There are different types of fuel cells which you can categorize by the type of membrane that is present at the interface between the fluid and solid phases.

Solid Oxide Fuel Cells (SOFC)

SOFCs are used to store energy for high temperature applications (900–1000 ºC).

SOFCs are mostly made of porous ceramic materials which require high temperatures before they become ionically conductive. The solid oxide material that is used for the electrolyte must reach a temperature that is high enough to support the kinetic transport of oxide ions through it from the cathode to the anode.

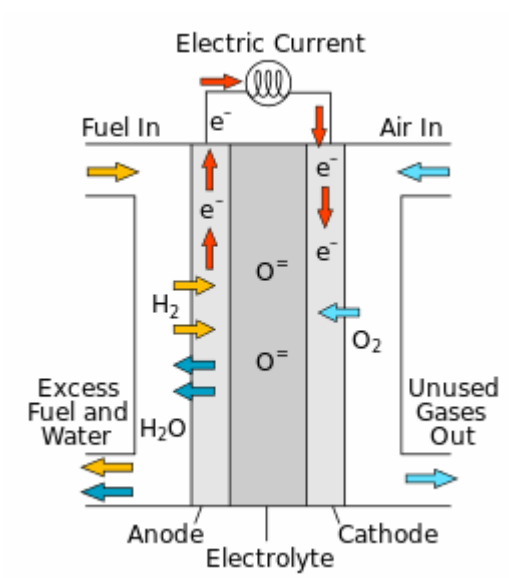
Most of the electrochemical reactions in a SOFC occur at triple phase boundaries (TPB) where the electrode, electrolyte, and gas phase coincide. Hydrogen (H_2) diffuses through a porous anode to the TPB between the anode and electrolyte where it reacts with oxide ions (O^{-2}) from the electrolyte to produce water.

$$\mathrm{H}_2 + \mathrm{O}^{-2} \rightleftharpoons \mathrm{H}_2\mathrm{O} + 2e^- \tag{1072}$$

The electrons (e^-) that are produced in this reaction are conducted electrically from the anode, through an external circuit, to the cathode. This flow of electrons generates electrical power. Oxygen (O_2) diffuses through the porous cathode and reacts with the electrons to form oxide ions (O^{-2}).

$$\mathrm{O}_2 + 4\,e^- \rightleftharpoons 2\mathrm{O}^{-2} \tag{1073}$$

The oxide ions are then transported from the cathode, through the electrolyte, to the anode. The series of reactions continue as long as there is a supply of hydrogen fuel at the anode and oxygen at the cathode.



When designing a fuel cell, one of the most significant challenges is to maximize its efficiency and performance. The aim is to maximize the transport of species and minimize energy losses in the cell, for example, energy loss due to diffusion, or ohmic losses due to ohmic resistance within ceramic components. The transport of species and energy losses are affected by variables such as the thickness, surface area, or material of the porous layers within the cell. The characteristics of the TPB have a significant impact on the electrochemical performance of the cell.

When simulating fuel cells in STAR-CCM+, you can define separate solid continua for the anode, cathode, electrolyte membrane, and tabs—where all components except the tabs are reactive domains. You can then

use separate fluid continua for the air and fuel domains—or if the air and fuel streams mix at any point, use one continuum. Follow the guidelines in the workflow, Modeling Electrochemical Surface Reactions.

There are some important considerations to account for when modeling SOFCs when using the present implementation of STAR-CCM+.

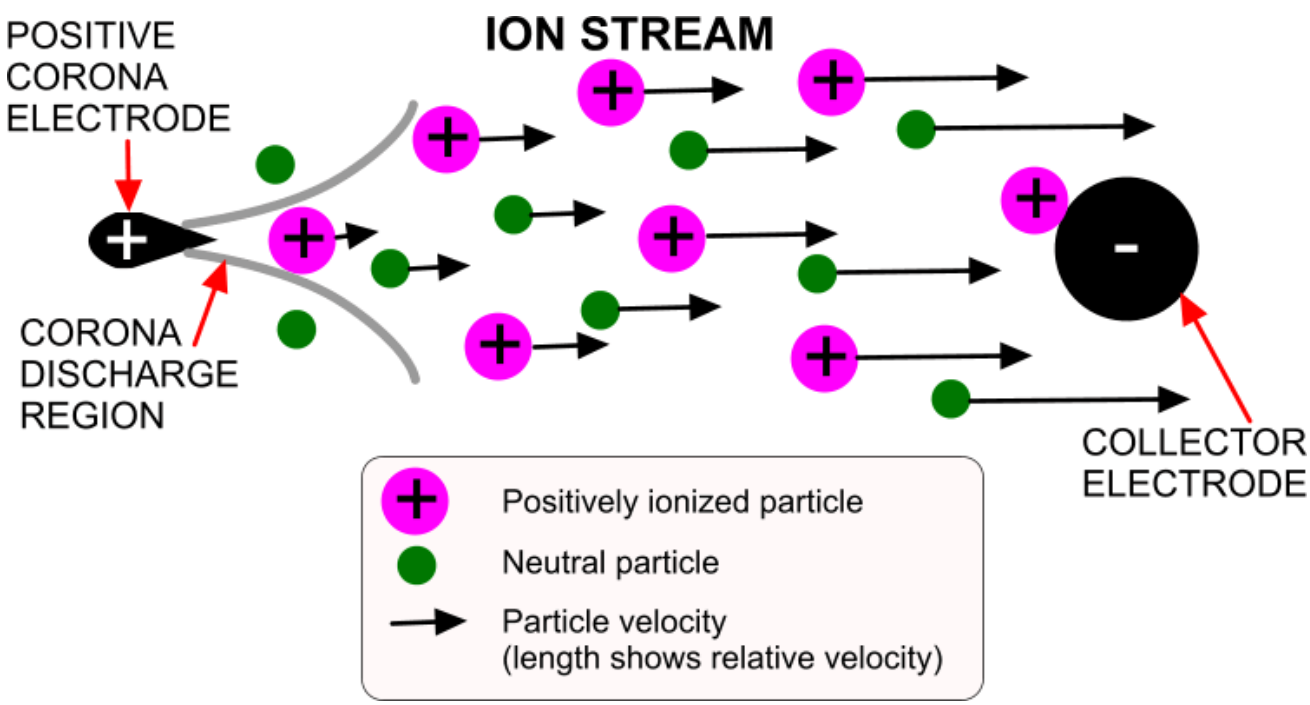
In particular, the setup must obey the first law of thermodynamics—energy cannot be created or destroyed. The difference between the overall energy of the species entering a reactive domain must be equal to the overall energy of the species leaving that domain—after accounting for the sum of the energy that the electrochemical surface reactions create and consume within that domain. You can check the energy conservation in STAR-CCM+ by creating reports for electric power flux and energy loss and then using these reports to create monitors and a single plot.

Since the reaction sites are in close proximity to the fuel and air inlets in the reactive domains, there is significant flow boundary diffusion. With flow boundary diffusion, it is possible that specifying the total flux and species concentrations leads to flux of some species in the opposite direction of the mean flow. Therefore, another important consideration is to decide if the simulation ignores flow boundary diffusion in the overall computations.

Ionic Wind

Ionic wind is a term that is used to describe an electrohydrodynamic flow that occurs due to a corona discharge.

When ions are created at a high-voltage corona electrode, coulombic forces that are exerted on the ions create an ionic wind. The corona electrode can be either positive, as in the example below, or negative. The ions accelerate towards the oppositely charged collector electrode due to Coulomb forces. Ions collide with the electrically neutral gas and increase the overall velocity in the direction of flow. Typically, the collector neutralizes the ions so that the electric current density beyond the collector electrode is close to zero.



The ionization of the fluid surrounding a charged conductor (corona electrode) with high curvature causes an electrical discharge, called the corona discharge. The corona phenomenon occurs when the potential of the electrode is high enough to create a thin plasma sheet around the electrode, but not high enough to form an arc between them.

You can model ionic wind simulations in STAR-CCM+ and modify the set-up to maximize convective flow while minimizing power consumption. For example, you can vary aspects such as voltage or distance between electrodes, and determine the effect on the velocity of the particle flow at the outlet. For many applications, the aim is to minimize the loss of ions beyond the collector electrode. However, you can also adjust the set-up to maximize the electric current density beyond the collector electrode — for example, to produce a focused flow of ions for cleaning applications.

In STAR-CCM+, you can create a surface integral report to test if the electric charge that is transported through the domain is conserved. The Boundary Specific Electric Current field function can be used for the surface integral report along with the boundaries of all regions which make up the simulated ionic wind domain set as parts. In a perfectly converged, steady state simulation the total surface integral over all boundaries is expected to be small compared to the largest integral value of a single boundary.

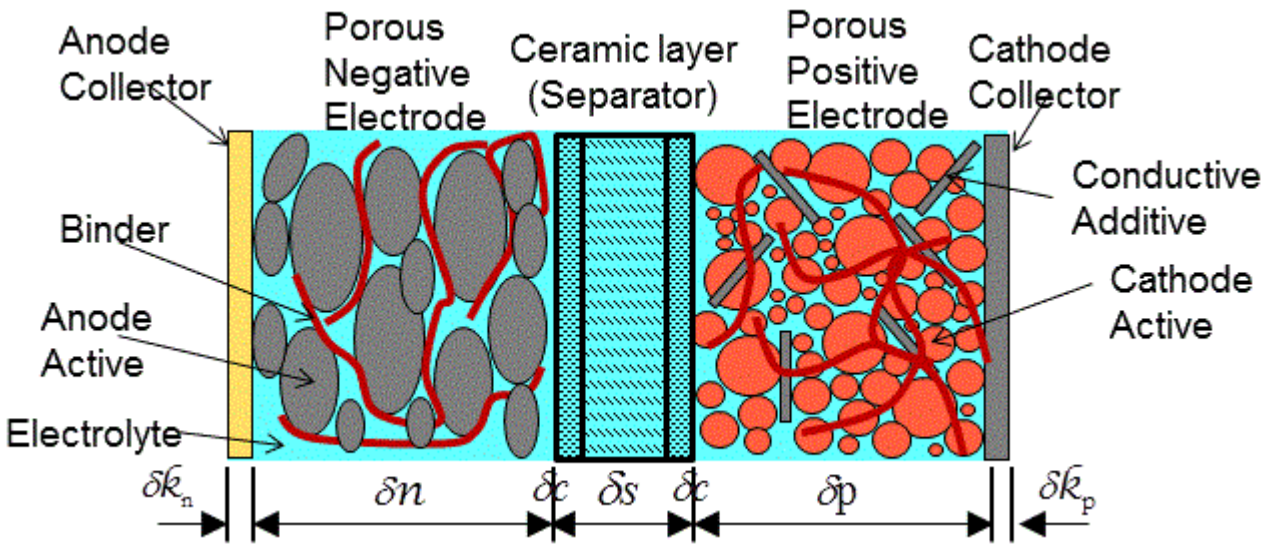
To model ionic wind, follow the steps to model ionic flux using the Electrostatic Potential model. See Modeling Ionic Species Flux.

Li-Ion Battery Cells

You can use the Li-Ion Battery Cell model to simulate the process of charging or discharging rechargeable Li-ion battery cells, or discharging disposable lithium battery cells. This is the start of your concept.

Li-Ion battery cells harness electrical energy that is created due to differences in electrochemical potential between reactants and products in the cells. The electrical current that is created from Li-Ion battery cells can then power devices such as portable electronics, Hybrid Electric Vehicles (HEVs), and Plug-In Hybrid Electric Vehicles (PHEVs).

The following annotated diagram depicts a typical rechargeable Li-ion battery cell. During the discharge process, lithium de-intercalates from the anode active material where it dissociates at the Solid Electrolyte Interface (SEI) into lithium cations (Li+) and an equal number of electrons. The Li+ ions in the electrolyte migrate through the separator and are inserted into the cathode active material. The electrons flow from the solid anode active material through the anode collector and leave the Li-ion battery cell as an electrical current. Charging is the reverse process.



Disposable lithium batteries discharge in the same way, however, instead of using an intercalated lithium compound as the anode active material, a foil of metallic lithium is used – which is not typically recharged.

Battery cells require both an anode and a cathode electrode to complete a circuit and allow current to flow. However, if you are only concerned with the detailed reactions which occur at one of the electrodes – for example, sacrificial anode depletion, you can create a lumped electrode to represent the other electrode.

You can use STAR-CCM+ to model Li-Ion battery cells. See Simulating Charge and Discharge Processes in Li-Ion Battery Cells.

To practice setting up a Li-ion battery cell, try working through the Li-Ion Battery Cell Model: Cell Electrochemistry Analysis tutorial.

Note: The Li-Ion Battery Cell model is available with the double precision version of STAR-CCM+ only.

The ability to model the micro-structure of an electrode and predict its electrochemical reactions, helps to determine the quality and performance of a cell design. For example, it is possible to determine how tortuous a porous electrode can be, and analyze or parameterize particle size and distribution, both key factors for an electrode. You can also determine the most effective setup to optimize high energy density–for mobile phones, high power output–for automotive applications, or a combination of both.

When using the Lithium Ion Battery Cell model, you can simulate lithium-ion batteries which contain electrolytes of a binary lithium-based salt such as:

- LiPF6 (Lithium hexafluorophosphate)
- LiBF4 (Lithium tetrafluoroborate)
- LiClO4 (Lithium perchlorate)
- LiCF3SO3 (Lithium trifluoromethanesulfonate)
- LiFSI (Lithium bis(fluorosulfonyl)imide)
- LiTFSI (Lithium bis(trifluoromethanesulfonyl)imide)

The Li-Ion Battery Cell model serves as a gateway between two Li-Ion physics models:

The Li-Ion Electric Potential Model

The Li-Ion Electric Potential model solves for the electric potential.

Select this model for all the battery model physics continua that are applied to a lithium-ion battery cell.

The Li-Ion Concentration Model

The Li-Ion Concentration model accounts for the transport of lithium in the (solid) electrodes, and transport of lithium ions in the (liquid) electrolyte.

Select this model only for the physics continua that describe the corresponding electrodes and the electrolyte.

At the Solid Electrolyte Interface (SEI) and in the electrolyte, both physics models are coupled in the following manner:

- The interface conditions for both electric fluxes and concentration depend on the same SEI normal component of electric current density. The SEI normal current is computed from the Butler-Volmer relationship, which is in turn a function of both solid/liquid electric potentials and solid/liquid concentrations.
- The electric potential is solved using Eqn. (1081) which depends on the solution for the Lithium/Salt Concentration c.
- Concentration transport depends on electric potential, see Eqn. (1078) and Eqn. (1083).

The model transport equations for the solid phase (Eqn. (1080) and Eqn. (1082)) do not contain terms which are directly coupling the electric potential with the Li-ion concentration.

Contents:

Modeling Charge and Discharge Processes in Li-Ion Battery Cells
Li-Ion Battery Cells Reference

Modeling Charge and Discharge Processes in Li-Ion Battery Cells

You can use the Li-Ion Battery Cell model to simulate the process of charging or discharging rechargeable Li-ion battery cells, or discharging disposable lithium battery cells.

1. Start a **double-precision** version of STAR-CCM+. The double-precision version is required for modeling the transport of Lithium ions.
2. Create a new simulation.

When modeling Li-ion battery cells, it can be difficult to manipulate parts at the microscale level due to the precision of the CAD-based tools (1.0E-8m) in STAR-CCM+. To avoid such issues, import parts or create 3D-CAD parts using the meter length scale initially, then you can scale the mesh at a later stage.

3. Import the geometry parts or create 3D-CAD parts that represent the microscale section of the Li-ion battery cell.
   If you are importing parts, in the import dialog, make sure that you activate **Create Part Contacts from Coincident Entities**. Each Li-ion battery cell must contain at least the following parts:
   - an anode electrode
   - a cathode electrode
   - an electrolyte

   If necessary, you can also include parts for anode and cathode collectors.

   The anode and cathode electrodes can be represented in different ways, depending on the scenario that you are modeling. Choose one of the following options:
   - To obtain the most realistic results, you can import geometry that is created from a three-dimensional reconstruction of an electrode. You can use parts or meshes that are created from imaging technology such as x-ray, or focused ion-beam/scanning electron microscopy (FIB/SEM) imaging.
   - For a simplified representation of the irregular and complex structure of the active material, for each electrode you can create a simple solid structure of randomly distributed solid particles that form one part. The solid active material exists within the electrolyte. You can decrease the porosity by increasing the number of solid particles that are contained in the overall volume, or increase the porosity by decreasing the number of solid particles.
   - When you only want to model one electrode in detail, follow one of the previous options for this electrode, then use the lumped approach for the other electrode. Either electrode can be lumped, depending on what application you are interested in modeling. For example, if you are concerned with modeling discharge of graphite anodes, you can create a lumped cathode. However, if you want to model cathodic protection, you can create a lumped anode.
     1. Approximate the lithium foil by a hexahedral block. The mesh for this block can be coarse.
     2. For the Solid Electrolyte Interface (SEI), use an Electrode Equilibrium Potential that is linear and set the reference foil Electrode Equilibrium Potential $U_{\mathrm{eq}} = 0$ as follows:
        - For *Electrode Equilibrium Potential for y = 0*, set the value to 0.
        - For *Electrode Equilibrium Potential for y = 1*, set the value to 0.
     3. Deactivate **Prevent Electrode Depletion**.
     4. Set the rate constant to a high value, for example `1.0E-3 kmol/m2 s`.

Typically, lumped electrodes are used when the behavior at that electrode is not of interest. Lumped electrodes are a crude approximation of the more detailed electrodes which contain active material dispersed through an electrolyte. The lumped electrode is only required to complete the circuit. Therefore, setting a high rate constant ensures that materials can easily pass across the interface at the lumped electrode and continue the flow of current.

5. To make sure that the resolved, non-foil electrode is capacity limiting, set *Maximum Concentration in Electrode* $c_{s,\max}$, to a high value, for example $c_{s,\max}$= `1E30`.

- If you want to model a disposable battery, create the cathode using the same process as in either of the first two options above. Then, for the anode, create a thin solid part to represent a foil of lithium metal.

Note: To obtain realistic results when you use simplified representations, it is important to make sure that the porosity of the active material in an electrode is represented accurately. You can determine the porosity from the ratio of the solid volume to the total volume. Li-Ion batteries with a high porosity (less solid volume and more electrolyte) have a high-power output since there is less resistivity in ionic conductive pathways which allows high currents to flow. However, Li-Ion batteries with a low porosity (more solid volume and less electrolyte) are better for storing a high energy density as energy is stored in the solid material.

4. If contacts do not exist between coincident parts, create part contacts or imprint parts.
See: Creating Part Contacts Manually or Imprinting Parts On Their Neighbors.
5. Assign parts to regions, make sure that **Create Interfaces from Contacts** is activated, and select the following options:
   - Create a Region for Each Part
   - Create a Boundary for Each Part Surface
   - Create One Feature Curve for All Part Curves
6. Create Physics Continua to represent the regions.
The table below shows an example of how to allocate regions to physics continua:

Physics Continua	Regions
[Anode Electrode Continuum]	[Anode Electrode]
[Anode Collector Continuum]	[Anode Collector]
[Cathode Electrode Continuum]	[Cathode Electrode]
[Cathode Collector Continuum]	[Cathode Collector]
[Electrolyte Continuum]	[Anode Electrolyte]
	[Cathode Electrolyte]
	[Separator]

7. For each Physics Continuum, select the physics models.
The table below shows an example selection of physics models (in order) for each of the different continua:

Note: **Auto-select recommended models** is activated.

Group Box	Model				
	[Anode Electrode Continuum]	[Anode Collector Continuum]	[Cathode Electrode Continuum]	[Cathode Collector Continuum]	[Electrolyte Continuum]
Time	**Implicit Unsteady**				
Material	**Solid**				**Liquid**
Flow					**Segregated Flow**
Viscous Regime					**Laminar**
Equation of State	**Constant Density**				
Space	Either, **Three Dimensional** or **Two Dimensional**				
Optional Models	**Electrochemistry**				**Electrochemistry**
Electrochemistry	(**Li-Ion Battery Cell** is selected automatically)				**Li-Ion Battery Cell**
Optional Models	**Li-Ion Concentration**		**Li-Ion Concentration**		**Li-Ion Concentration** (selected automatically)
	Li-Ion Electric Potential				**Li-Ion Electric Potential** (selected automatically)
	Segregated Solid Energy				
Energy					**Segregated Fluid Temperature**

Note:
- The Li-Ion Battery Cell Model is available only in implicit transient simulations.
- The Li-Ion Concentration model is not used for anode or cathode collectors as this model is designed to model the concentration of lithium in the active material and electrolytes only.

8. Set the required Material Properties for the Solid and Liquid models within each physics continuum.
   Specific Liquid Material Properties:
   - When using either the **Li-Ion Electric Potential** or **Li-Ion Concentration** model, you can specify liquid electrolyte material properties using predefined material property constants or tables which are stored in the materials database. Each table contains values for the material property at varying Li-ion/salt concentrations and temperatures of the electrolyte. Constant values are given for each material property at 30 °C and a molality of 1 mol/kg.

1. Expand the **Continua** > **Continuum** > **Models** > **Liquid** > **[Electrolyte]** > **Material Properties** node.
2. Depending on the case, select one of the following options:
   - When values of Temperature and Lithium/Salt Concentration do not vary much, use the predefined constant data:

     Multi-select the following nodes; **Concentrated Solution Correction to Salt Diffusivity**, **Electrical Conductivity**, **Lithium/Salt Diffusivity**, **Non-Ideal Solution Correction to Salt Migration**, and **Density**. Then in the *Multiple Objects - Properties* window, set *Method* to **Constant**.
   - For detailed setups in which the values of Temperature and Lithium/Salt Concentration vary significantly, use the predefined tabular data:

     Multi-select the following nodes; **Concentrated Solution Correction to Salt Diffusivity**, **Electrical Conductivity**, **Lithium/Salt Diffusivity**, and **Non-Ideal Solution Correction to Salt Migration**. Then in the *Multiple Objects - Properties* window, set *Method* to **Table(T,c)**.

9. Set the boundary conditions. You cannot set independent boundary conditions for the liquid electrolyte phase. This phase has an interface with solid active electrodes. If the Li-Ion Electric Potential and Li-Ion Concentration models are activated in both regions, the SEI interfaces are created automatically. However, one region must be liquid and the other solid; SEI interfaces are not created for solid/solid and liquid/liquid interfaces.

   If you set the **Specific Electric Current** or **Electric Current Density** boundary condition, you are explicitly specifying the electric current that is flowing across that boundary. For example, if you set a value of 1 A/m2, the normal component of the electric current density is set to a constant value of 1 A/m^2 along the entire boundary.

   For the **Electric Current** boundary condition, you are specifying the total electric current that is flowing across the boundary. However, the profile of the specific electric current along the boundary is not necessarily constant. It is the integral value of the current across the boundary that is fixed at the specified value; the specific electric current can change as the solution develops.

   a) Select the **Regions** > **[Region]** > **Boundaries** > **[Boundary]** > **Physics Conditions** > **Electric Potential Specification** node.

   b) In the *Properties* window, set the *Method* property:
      - For a Neumann boundary condition:
        - If you know the total electric current, set *Method* to **Electric Current**.

          The electric current is defined in A. This boundary condition type specifies the total electric current through the boundary. Current entering the domain is a positive electric current. Current leaving the domain is a negative electric current.
        - If you know the electric current density, set *Method* to **Specific Electric Current** or **Electric Current Density**.

          The electric current density is defined in A/m^2. The total current through the boundary is the product of the electric current density with the area of the boundary. The Specific Electric Current is entered as a scalar profile, while the Electric Current Density is entered as a vector profile.
      - For a Dirichlet boundary condition:
        - If you know the electric potential, set *Method* to **Electric Potential**.

The Electric Potential φ is defined in V.

- If you know the electric current density, set *Method* to **Electric Current Density**.

  The electric current density is defined in A/m^2.

c) Select the **Regions** > **[Region]** > **Boundaries** > **[Boundary]** > **Physics Conditions** > **[Specific Electric Current**, **Electric Current**, **Electric Current Density**, or **Electric Potential]** node and set the value.

To define the boundary conditions for the concentration:

a) Select the **Regions** > **[Region]** > **Boundaries** > **[Boundary]** > **Physics Conditions** > **Lithium Wall Concentration Specification** node.

- To specify a Neumann boundary condition, set *Method* to **Flux**.
- To specify a Dirichlet boundary condition, set *Method* to **Concentration**.

b) Select the **Regions** > **[Region]** > **Boundaries** > **[Boundary]** > **Physics Conditions** > **[Lithium Concentration Flux** or **Lithium/Salt Concentration]** node and set the value.

10. Define the properties of the Solid Electrolyte Interface. See Li-Ion Electric Potential Model Interface Settings.
11. Set the Initial Conditions. To determine the appropriate initial (or reinitialization) conditions for Lithium/Salt Concentration in both electrodes:

    a) Create an initial setup with a constant Lithium/Salt Concentration of $c = c_{s,\max}/2$ as initial value in both electrode regions.

    b) Optionally, select the **Reports** > **[Li-Ion Cell Report]** node, and set the *SOC Step Increment* expert property.

    This expert property sets the resolution of the state table.

    c) Run the Li-Ion cell Report.
    In the Output window, the state table gives the conditions for a cell at a given SOC.

    d) Use the value of Average Concentration+ at the required SOC for the concentration at the cathode.

    e) Use the value of Average Concentration- at the required SOC for the concentration at the anode.
12. Set the Solver Parameters and Stopping Criteria.
    See:
    - Electric Potential Solver Reference
    - Lithium/Salt Concentration Solver Properties
13. Prepare Scenes and Plots on which to visualize results.
    For available field functions, see:
    - Li-Ion Concentration Model Field Functions
    - Li-Ion Electric Potential Model
14. Run the Simulation.
15. You can also create Li-ion cell reports after initializing the simulation.

    a) Create a Li-Ion cell report.

    b) Select the **Reports** > **[Li Ion Cell Report]** node and set the Li-Ion Cell Properties.

    c) Optionally, set the *Monitor Value* to the quantity that is monitored when creating a monitor.

Contents:

Troubleshooting Electrochemistry Simulations
What Is the Solid Electrolyte Interface (SEI)
Guidelines for Simulating Lithium Depletion and Saturation

Troubleshooting Li-Ion Battery Cell Simulations

Lithium/Salt Concentration does not change in solid active materials

Make sure that all solid active materials are electrically connected and have electrical connections to the current collectors.

Convergence problems when initializing with an arbitrary potential distribution

Just like velocity and pressure in an incompressible CFD computation, Lithium/Salt Concentration and Li-Ion Electric Potential are closely related. When an arbitrary potential distribution (for example, the default zero field) is chosen as initial condition, the solution must be converged within the first time-step.

In some cases, this convergence is disrupted during the first inner iterations. The value of the SEI surface overpotential η (see Eqn. (1102)) is far off the solution value, leading to a spurious computation of the SEI-specific electric current Eqn. (1100). This spurious computation affects the Li-Ion Concentration solver.

To overcome this problem, deactivate the Li-Ion Concentration Solver during the first few inner iterations, by setting the under-relaxation factor to zero.

Convergence problems when initial Lithium/Salt Concentration set to zero or saturation value (maximum)

A uniform zero Lithium/Salt Concentration in electrolyte corresponds to a pure (that is, non-conducting) solvent. Such a setup is physically possible, but breaks the operability of the Li-ion battery cell, and is thus considered an input error.

When the electrode Lithium/Salt Concentration values are exactly $c_s = 0$, or $c_s = c_{s,\max}$, the derivative of the SEI-specific electric current with respect to Lithium/Salt Concentration becomes infinite (see Eqn. (1101)). The solver becomes unstable.

Such extreme initial conditions are set when investigating the formation process of a Li-ion battery cell. The implementation in STAR-CCM+ cannot handle such cases.

However, simulations in close vicinity of these settings are possible: set initial conditions close to these limits.

Convergence problems when Lithium/Salt Concentration depletes locally in electrolyte, or depletes/saturates in solid electrodes.

When lithium depletes or saturates, the numerical scheme becomes less stable when Eqn. (1100) is evaluated using a 1st-order projection of the Lithium/Salt Concentration towards the SEI. To stabilize the algorithm, use the 0th-order projection.

Convergence problems when setting zero resistance in Butler-Volmer Relationship Parameters

To overcome these issues:

- Start with a non-zero resistance.
- Compute some inner iterations until the residuals for Li-Ion Electric Potential and Lithium/Salt Concentration have decreased substantially.

Li-Ion Cell Report: Cannot find battery cell states where OCV equals voltages at 0% and 100% SOC

This error can occur when a monitor is used for the Li-Ion Cell Report.

Lithium/Salt Concentration is a conserved quantity. However, due to non-converged simulations, Lithium/Salt can be erroneously introduced into the simulation. Due to this accumulation or loss of lithium, the Li-ion battery cell is incapable of assuming states where the OCV corresponds to the user-defined values at 0% and 100% SOC.

This error points at issues where the solution has not properly converged.

What Is the Solid Electrolyte Interface (SEI)

The Solid Electrolyte Interface (SEI) is the interface between the solid electrode material and the liquid electrolyte.

Interface refers to a location. Interphase refers to the processes that take part between the two phases of solid active material, and the liquid electrolyte phase (that is, interphase reaction).

SEIs are created between interfacing liquid-solid regions, where both the Li-Ion Concentration model, and the Li-Ion Electric Potential model are selected on both sides.

Guidelines for Simulating Lithium Depletion and Saturation

High charge or discharge currents can lead to strong gradients of Lithium/Salt Concentration within a lithium-ion battery cell. The enhanced transport, and the concentration differences that are required to sustain the gradients then rapidly deplete or saturate (locally) electrodes or electrolyte.

The Li-Ion Battery Cell model can handle cases of lithium depletion and saturation within solid electrodes, as well as the depletion of salt in electrolyte.

When lithium depletes or saturates, the numerical scheme becomes less stable when Eqn. (1100) is evaluated using a 1st-order projection of the Lithium/Salt Concentration towards the SEI. To stabilize the algorithm, use the 0th-order projection.

To account for the resulting low order of the scheme, use a mesh with sufficiently refined prism layers at the SEI.

Simulation of lithium depletion or saturation requires that an equilibrium state can be found. At that equilibrium state, the driving force for reactions at the SEI vanishes, that is $\eta = 0$ in Eqn. (1102). Make sure that such an equilibrium exists by providing appropriate electrode equilibrium potential profiles. If this requirement is not met, activate **Prevent Electrode Depletion**.

When the electrolyte depletes, activating **Prevent Electrode Depletion** corrects the SEI surface overpotential to become $\eta = 0$ when the electrolyte depletes of salt.

Li-Ion Battery Cells Reference

This section provides detailed information about the models that are available for modeling lithium-ion battery cells.

Contents:

Li-Ion Battery Cell Model Reference
Li-Ion Electric Potential Model Reference
Li-Ion Concentration model Reference

Li-Ion Battery Cell Model Reference

The Li-Ion Battery Cell model can predict the spatial distribution of ions within an arbitrary, multi-phase microstructure region.

Note: The Li-Ion Battery Cell Model requires the double-precision version of modeling the transport of Lithium ions.

The Li-Ion Battery Cell model uses the following battery cell models:

- Li-Ion Electric Potential model

  Allows you to determine the electric potential in solid and electrolyte regions.
- Li-Ion Concentration model

  Allows you to model the salt concentration in an electrolyte, and the concentration of lithium in active parts of an electrode.

Li-Ion Battery Cell Model Reference

Theory	See Li-Ion Concentration model Formulation, Li-Ion Electric Potential model Formulation, and Li-Ion Cell Report	
Provided By	**[physics continuum]** > **Models** > *Electrochemistry*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Li-Ion Battery Cell**	
Requires	Using a double precision version of STAR-CCM+: *Space*: **Three Dimensional** or **Two Dimensional** *Time* : **Implicit Unsteady** *Material*: **Solid** or **Liquid** *Optional Models*: **Electrochemistry**	
Activates	Physics Models	*Energy*
	Report Options	**Li-Ion Cell**. See Li-Ion Battery Cell Reports.

	Field Functions	**Report: Li-Ion Cell 1**. See Li-Ion Battery Cell Field Functions.

Li-Ion Battery Cell Reports

Li-Ion Cell

The Li-Ion cell report gives the following values:

- Open Circuit Voltage OCV
- State of Charge SOC
- Capacity CAP

The following properties are available for the **Reports** > **[Li Ion Cell Report]** node.

Voltage at 100% SOC

The voltage at which the battery is fully charged.

Voltage at 0% SOC

The voltage at which the battery is considered to be fully discharged. (The lower cut-off voltage).

Positive Electrode Parts

The parts that constitute the positive electrode.

Negative Electrode Parts

The parts that constitute the negative electrode.

Field Functions

Report: Li-Ion Cell 1

Li-Ion Electric Potential Model Reference

The Li-Ion Electric Potential model solves the transport equations for electric charge to get values for the electric potential.

All of the physics continua that are applied to a lithium-ion battery cell use the Li-Ion Electric Potential model which assumes a binary lithium-based electrolyte.

The transport equations for electric charge are Eqn. (1080), and Eqn. (1081). The current density, J, is defined using Eqn. (1078).

Li-Ion Electric Potential Model Reference

Theory	See Li-Ion Electric Potential Model Formulation.
Provided By	**[physics continuum]** > **Models** > *Optional Models*

Example Node Path	**Continua** > **Physics 1** > **Models** > **Li-Ion Electric Potential**	
Requires	Using a double precision version of STAR-CCM+: *Space*: **Three Dimensional** or **Two Dimensional** *Time* : **Implicit Unsteady** *Material*: **Solid** or **Liquid** *Optional Models*: **Electrochemistry** *Electrochemistry*: **Li-Ion Battery Cell**	
Properties	Secondary Gradients. See Li-Ion Electric Potential Model Properties.	
Activates	Materials	Electrical Conductivity. See Li-Ion Electric Potential Model Materials and Methods.
	Initial Conditions	Electric Potential. See Li-Ion Electric Potential Model Initial Conditions.
	Boundary Inputs	Electric Potential Specification. See Li-Ion Electric Potential Model Boundary Settings.
	Interface Inputs	See: Li-Ion Electric Potential Model Interface Settings
	Solvers	Electric Potential Electric Potential Solver Reference.
	Field Functions	Electric Potential, Electric Field, Electric Current, Electrical Conductivity, Boundary Specific Electric Current, SEI Surface Overpotential. See Li-Ion Electric Potential Model Field Functions.

Li-Ion Model Properties

Secondary Gradients

Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces.

- **On**: Default value. Solves for interior and boundary types of secondary gradient.
- **Off**: Does not solve for either type of secondary gradient.
- **Interior Only**: Solves for the interior secondary gradients only.
- **Boundaries Only**: Solves for the boundary secondary gradients only.

Materials and Methods

Electrical Conductivity

In transient simulations, specifies the electrical conductivity, σ, of the material (see Electrical Conductivity: Generalized Ohm's Law).

Method	Corresponding Method Node
Constant, Field Function Available for fluids and solids.	**Electrical Conductivity > Constant, Field Function** Specifies σ as a scalar profile.

Initial Conditions

Electric Potential

Allows you to initialize the electric potential ϕ to a specified scalar profile.

Boundary Settings

Electric Potential Specification

Method	Corresponding Physics Value Nodes
Specific Electric Current	
Electric Potential	
Electric Current	
Electric Current Density	
Electric Current Potential Characteristic	
Insulator	

Wall Boundary

Electric Potential Specification

Method	Corresponding Physics Value Nodes
Specific Electric Current Allows you to define the electric current density normal to the boundary.	**Specific Electric Current** Sets the specific electric current J_n to a specified scalar profile. See Eqn. (3733).
Electric Potential Allows you to define the electric potential at the boundary according to Eqn. (3729).	**Electric Potential** Defines the potential $\overline{\phi}$ in Eqn. (3729) as a scalar profile. When leaving the **Electrical Resistance** to the default value of zero, the potential at the boundary is simply $\phi = \overline{\phi}$. **Electrical Resistance** Defines an additional source of electrical resistance (R^{ex} in Eqn. (3729)) as a scalar profile. **Electrical Resistance Potential Derivative** For stabilization, allows you to specify the dependence of R^{ex} on ϕ by defining $dR^{\mathrm{ex}}/d\phi$.

Electric Current Allows you to define the total electric current through the boundary.	**Electric Current** Sets the total electric current I_Γ to a specified scalar profile. See Eqn. (3732).
Electric Current Density Allows you to define the electric current density at the boundary.	**Electric Current Density** Sets $\mathbf{J}$ to a specified vector profile. See Eqn. (3694).
Electric Current Potential Characteristic Allows you to define the I-V curve at the boundary.	**Current Voltage Characteristic** Defines the relationship between the electric current flowing through the boundary and the electric potential at the boundary. In electrochemistry applications, you typically define this relationship using the Butler-Volmer method (see Butler-Volmer Current-Potential Characteristic). Alternatively, you can specify the I-V curve by providing tabular data. For more information, see Setting the Electric Current Potential Characteristic and Current-Voltage Characteristic Reference. **Electric Potential** Defines the potential ϕ_1 in Eqn. (3734) as a scalar profile. **Electrical Resistance** Defines the electrical resistance R in Eqn. (3734) as a scalar profile.
Insulator Sets the total electric current through the boundary (Eqn. (3732)) to zero.	None

Li-Ion Electric Potential Model Interface Settings

Butler-Volmer Relationship Parameters

Once the corresponding physics continua are assigned to regions, you use this node to set the parameters for the Solid Electrolyte Interface (SEI).

Resistance

R_0 in Eqn. (1104).

SEI Activation

E_{SEI} in Eqn. (1104).

Capacitance

C in Eqn. (1100).

Rate Constant

k_0 in Eqn. (1103).

Kinetic Activation Energy

E_a in Eqn. (1103).

Maximum Concentration in Electrode

$c_{s,\max}$ in Eqn. (1101).

Reference Concentration in Electrolyte

$c_{l,\mathrm{ref}}$ in Eqn. (1101).

Anodic Transfer Coefficient

α^a in Eqn. (1100).

Cathodic Transfer Coefficient

α^c in Eqn. (1100).

Electrolyte Reaction Order

α^3 in Eqn. (1101).

Electrode Reaction Order

α^1 in Eqn. (1101).

Vacancies Reaction Order

α^2 in Eqn. (1101).

Under-Relaxation Parameter

At each iteration, governs the extent to which the newly computed solution supplants the old solution. The default value (1.0) means that no under-relaxation is used.

Electrode Equilibrium Potential

Setting	Corresponding Sub-Node
Linear This method inputs a simplified electrode equilibrium potential U_{eq}. You supply the values of U_{eq} for the points $y = 0$ and $y = 1$. The values for the interval $y = [0, 1]$ are linearly interpolated.	**Linear Electrode Equilibrium Potential** *Electrode Equilibrium Potential for y=0* The value of U_{eq} at $y = 0$. *Electrode Equilibrium Potential for y=1* The value of U_{eq} at $y = 1$.

For Li-Ion cells, you can approximate the electrode equilibrium potential by a constant, for example 3.7 V for the cathode. When choosing such a setup, the driving force for discharge does not increase during the charging and the discharging process. If **Prevent Electrode Depletion** is deactivated, this circumstance can lead to numerical problems, where an equilibrium state cannot be attained.	
Tabular Uses a table as input for the electrode equilibrium potential U_{eq}. The electrode equilibrium potential must decrease monotonically with increasing electrode stoichiometry y.	**Tabular Electrode Equilibrium Potential Properties** *Table* The table that contains the values of U_{eq} as a function of y . *Relative Concentration in Electrode (y)* Column entry for the electrode stoichiometry y. *Electrode Equilibrium Potential* Column entry for the electrode equilibrium potential U_{eq} .

Projection Order

Sets the projection order that is used to compute the liquid concentration c_l and solid concentration c_s at the interface.

0th-order

Uses the corresponding cell center values. Use this method when lithium depletes or saturates.

1st-order

Uses a first order projection (default).

Prevent Electrode Depletion

When activated, corrects the user-defined equilibrium voltage curves to approach ∞ asymptotically when $y = 0$, and $-\infty$ when $y = 1$.

Activating this option ensures that a local chemical equilibrium exists under all external load conditions, where $\eta = 0$ in Eqn. (1102).

Prevent Electrolyte Depletion

When activated, corrects the SEI Surface Overpotential to become $\eta = 0$ when the electrolyte depletes of salt. See Eqn. (1102).

Field Functions

Electric Potential

ϕ in Eqn. (1080).

Electric Field

$\mathbf{E}$ in Eqn. (1074).

Electric Current

$\mathbf{J}$ in Eqn. (1074).

Electrical Conductivity

σ in Eqn. (1074), and κ_0 in Eqn. (1078).

Effective Electrical Conductivity

σ in Eqn. (1074), and κ in Eqn. (1078).

Boundary Specific Electric Current

SEI normal component of electric current density J_n

SEI Surface Overpotential

η in Eqn. (1102).

Li-Ion Concentration model Reference

The Li-Ion Concentration model accounts for the transport of lithium in the (solid) electrodes, and transport of lithium ions in the (liquid) electrolyte.

You select this model only for the physics continua that describe the corresponding solid electrodes and the liquid electrolyte. This selection is different to the Li-Ion Electric Potential model, which is activated in all of the battery model physics continua.

Li-Ion Concentration Model Reference

Theory	See Li-Ion Concentration Model Formulation.
Provided By	**[physics continuum]** > **Models** > *Electrochemistry*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Li-Ion Concentration**
Requires	Using a double precision version of STAR-CCM+: *Space*: **Three Dimensional** or **Two Dimensional** *Time* : **Implicit Unsteady** *Material*: **Solid** or **Liquid** *Optional Models*: **Electrochemistry**
Properties	Key properties are: **Secondary Gradients**. See Li-Ion Concentration Model Properties.

Activates	Materials	Lithium/Salt Diffusivity
	Initial Conditions	**Lithium/Salt Concentration**. See Initial Conditions.
	Boundary Inputs	See Li-Ion Concentration Model Boundary Settings.
	Solvers	**Lithium/Salt Concentration**. See Solvers.
	Monitors	**Salt Concentration**
	Field Functions	Lithium/Salt Concentration, Lithium/Salt Diffusivity, Effective Lithium/Salt Diffusivity, Lithium Cation Transference Number, Concentrated Solution Correction to Salt Diffusivity, Non-Ideal Solution Correction to Salt Migration. See Field Functions.

Li-Ion Concentration Model Properties

Secondary Gradients

Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces.

- **On**: Default value. Solves for interior and boundary types of secondary gradient.
- **Off**: Does not solve for either type of secondary gradient.
- **Interior Only**: Solves for the interior secondary gradients only.
- **Boundaries Only**: Solves for the boundary secondary gradients only.

Initial Conditions

Lithium/Salt Concentration

Boundary Settings

Wall

Lithium Wall Concentration Specification

Method	Corresponding Physics Value Nodes
Concentration	**Lithium/Salt Concentration**
Flux	**Lithium Concentration Flux**

Lithium/Salt Concentration Solver Properties

This solver controls the solution of the Li-Ion Concentration model in all of the continua for which the model is activated.

Under-Relaxation Factor

In order to promote convergence, this property is used to under-relax changes of the solution during the iterative process. If residuals show solution divergence and do not decrease, reduce the under-relaxation factor for the relevant solvers.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Field Functions

Lithium/Salt Concentration

c in Eqn. (1082).

Lithium/Salt Diffusivity

D in Eqn. (1082) and D_0 in Eqn. (1084)

Effective Lithium/Salt Diffusivity

D in Eqn. (1082) and Eqn. (1084).

Lithium Cation Transference Number

t_+^0 in Eqn. (1083).

Concentrated Solution Correction to Salt Diffusivity

$\frac{d\left(\ln c_0\right)}{d(\ln c)}$ in Eqn. (1084).

Non-Ideal solution Correction to Salt Migration

$\frac{d\left(\ln f_\pm\right)}{d(\ln c)}$ in Eqn. (1077).

Ionic Species Flux

You can use the Electrochemical Species model within a liquid or gas to model the transport of ionic species, including, diffusion, convection, and migration/drift fluxes.

Ionic Species Flux

There are many chemical processes that rely on the migration of charged particles. In general, charged particles are referred to as ionic species which can represent positively or negatively charged ions. Modeling the flow and concentration of ionic species using STAR-CCM+ provides a convenient and cost effective approach for comparing the efficiency of engineering processes. This ability is even more useful when raw materials for electrochemical applications are rare or expensive. You can simulate the behavior of ionic species realistically and prevent material waste by finding the most efficient setup using STAR-CCM+. The option is provided to solve for electrochemical species concentrations using a coupled or segregated approach.

By modeling all relevant transport mechanisms for ionic species, you can predict the heterogeneity of ion concentrations within an electrolyte. The migration flux of an ionic species is proportional to the electric field in the electrolyte. You can only define the migration of ionic species when the Electrochemical Species model is selected along with one of the following models:

- Electrodynamic Potential model

  Adding the Electrodynamic Potential model allows you to model applications such as:

  - paint delamination
  - pitting corrosion
  - crevice corrosion
  - galvanic corrosion
  - cathodic protection
  - electroplating
  - wet etching
  - flow batteries
  - metal-liquid interface reactions

  You additionally use the Electrochemical Reactions model when you want to model specific surface reactions in these applications.
- Electrostatic Potential model

  Adding the Electrostatic Potential model allows you to model applications such as:

  - ionic wind simulations
  - electrostatic precipitators
  - plasma

If the Electrochemical Species model is used without either the Electrodynamic Potential or Electrostatic Potential model, STAR-CCM+ can only calculate diffusion and convection flux, and not migration flux.

To model electrohydrodynamic forces that arise from local ionic charge densities, use the Coulomb Force model in addition to the Electrochemical Species model and Electrostatic Potential model. See Coulomb Force Formulation.

Contents:

Selecting the Electrochemical Species Model
Ionic Species Flux Reference

Modeling Ionic Species Flux

The Electrochemical Species model uses transport equations to solve the flux of ionic species in electrolytic solutions.

Follow this procedure to model convection, diffusion, and migration of ionic species using the Electrochemical Species model. This workflow does not include modeling reactions between electrochemical species—if you want to model electrochemical reactions involving ionic species, see Modeling Electrochemical Surface Reactions.

Note: A double precision version of STAR-CCM+ is required to model the transport of electrochemical species when coupling the electrochemical species model with an electric potential model.

When modeling any electrochemical application, the Electrochemistry model is required. The Electrochemical Species model is also required for modeling migration of species. Further physics model selections depend on the application that you are modeling.

To model ionic species transport in an electrolyte:

1. For the physics continuum that represents the electrolyte, select the following models, with **Auto-Select recommended models** activated:

Group Box	Model
Space	Select one.
Time	Select one.
Material	Select one of: • **Liquid** • **Multi-Component Liquid** • **Gas** • **Multi-component Gas**
Reaction Regime (when Multi-Component Liquid or Multi-Component Gas is selected)	**Non-Reacting**
Flow	Select either of: • **Segregated Flow** • **Coupled Flow.**
Equation of State	**Constant Density**

Group Box	Model
Viscous Regime	Select one.
Optional Models	**Electrochemistry**
Electrochemistry	**Electrochemical Species**
Optional Models	**Electromagnetism**
Electromagnetism	Select either of: • **Electrodynamic Potential** Select this model when simulating ionic species flux in applications such as corrosion or etching. See What Is the Electrodynamic Potential Model? • **Electrostatic Potential** Select this model when simulating non-corrosion applications, especially if you want to model ionic charge densities or other setups which do not require electroneutrality. See What Is the Electrostatic Potential Model?

Note:
- Use a liquid model for modeling corrosion applications.
- The Electrochemical Species model does not support Multiphase models.
- If you do not select an Electromagnetism model, the migration flux cannot be solved—only diffusion and convection flux are solved.

2. Select any extra *Optional Models* that are required.
   - If the **Electrostatic Potential** model is already selected and you want to model the coulomb force that is exerted due to non-zero ionic charge densities, select the **Coulomb Force** model. See Coulomb Force Solver.
   - If you want to set a constant temperature without solving an additional transport equation, select the **Segregated Fluid Isothermal model**. If no temperature model is selected, the Electrochemical Species model runs with a constant temperature of 293.15 K.
3. Click **Close**.
4. Define the materials. Any fluid materials that you select for the electrolyte continuum act as a solvent for the electrochemical species which are assumed to be part of a dilute solution.
   See either:
   - Managing Single-Component Materials
   - Managing Multi-Component Materials
5. Specify the electrochemical species.
   a) Right-click the **Electrochemical Species** > **Electrochemical Species Components** node and select **Select Mixture Components**.
   b) Select all of the electrochemical species that represent the ionic species that are present in the electrolyte. If you cannot find a complex electrochemical species component, you can either:
      - Close the dialog, add electrochemical species to the database, then reopen the *Select Mixture Components* dialog, and select the species. See Modifying a Copy of the Material Database.

- Select a similar electrochemical species component and modify its material properties to change the charge, and the type and quantity of atoms.

c) Click **Apply**, then **Close**.

d) To modify the composition or properties of an electrochemical species component, expand the **Electrochemical Species Components** > **[Electrochemical Species Component]** node and edit the Material Properties as required. See Electrochemical Species Components Reference.

e) Expand the **Electrochemical Species** > **Electrochemical Species Components** node and for each of the Electrochemical Species Components, specify the Method for each Material Property.
Certain material properties are specific to the Electrochemical Species model and allow you to define the properties of the ionic species directly. You can also define ionic species further under the **Tools** > **Material Databases** node.

Specify the method for solving the concentrations of the electrochemical species:

6. Select the **Electrochemical Species** node and specify the *Electrochemical Species Solver Option* as **Coupled** or **Segregated**.

   The Segregated Electrochemical Species Solver runs faster than the Coupled Electrochemical Species Solver at low current densities. However, the Coupled Electrochemical Species Solver provides the best stability and is more capable of reaching convergence at high current densities than the segregated option. The Coupled Electrochemical Species Solver is intended for use with the Electrochemical Reactions model or the Bulk Ion Chemical Reactions model. If you use the coupled solver without these models, the runtime increases considerably without any benefit.

7. If you use the Electrochemical Species model with the Electrodynamics Potential model:

   a) Expand the **Models** > **[Material]** > **Material Properties** node.

   b) Select the **Electrical Conductivity** node and set *Method* to **Electrochemical Species**.

8. Define the region type and conditions. You can specify a fluid or porous region for the electrolyte continuum. However, when using the Electrochemical Species model, porous baffle interfaces are not allowed.

9. Define the Initial Conditions. You can specify the Molar Concentration or Number Density of each electrochemical species as an initial condition for the electrolyte continuum. See Electrochemical Species Model: Initial Conditions.

10. When modeling ionic species flux in a region, if necessary, you can define specific source terms to solve in the transport equations. This optional feature allows you to adjust the predefined source terms to suit the application that is being modeled. See Electrochemical Species Model: Region Settings.

11. Define the boundary and interface conditions. See Electrochemical Species Model: Boundary Settings.

12. Set up Scenes and Plots to visualize the solution.
    For example, you can visualize the following field functions in a scalar scene or plot them on an XY plot:

    - boundary specific electric current
    - electric current density
    - electric potential
    - charged species mobility of specific electrochemical species
    - migration flux of specific electrochemical species
    - molecular diffusivity of specific electrochemical species
    - molar concentration of specific electrochemical species
    - number density of specific electrochemical species

See: Electrochemical Species Field Functions.

13. Run the simulation.

Ionic Species Flux Reference

This section provides detailed information about the electrochemistry models that are available for modeling ionic species flux.

Contents:

Electrochemical Species Model Reference
Electrochemical Species Components Reference

Electrochemical Species Model Reference

For each electrochemical species, the Electrochemical Species model solves a transport equation for the molar concentration of the species in the fluid solution. These species concentrations are then available for the bulk and surface electrochemical reaction models.

The Electrochemical Species model is an optional electrochemistry model which is available to use with the Electrostatic Potential model or the Electrodynamic Potential model. You can also use the Electrochemical Species model with the Electrodynamic Potential and Electrochemical Reactions models to simulate electrochemical surface reactions which take into account electrochemical species flux due to electrochemical species concentrations.

You can select either a coupled solver or a segregated solver specifically for the Electrochemical Species model. This functionality allows you to, for example, solve a coupled system of equations between the electrochemical species while maintaining a segregated approach to solving electric potential.

When using the Electrochemical Species model with the Electrochemical Reactions model, the Electrochemical Reactions model computes the contributions to the electric current from the species concentrations, the electrochemical species flux, and flux derivatives. The electric current is passed to the Electrodynamic Potential model and the species flux, and flux derivatives, are passed to the Electrochemical Species model.

Electrochemical Species Model Reference

Theory	See Electrochemical Species Model Formulation.
Provided By	**[physics continuum]** > **Models** > *Electrochemistry*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Electrochemical Species Model**
Requires	*Space*: any *Material*: one of **Gas**, **Liquid**, **Multi-Component Gas**, **Multi-Component Liquid** *Optional Models*: **Electrochemistry**
Properties	Key properties are: *Electrochemical Species Solver Option*. See Electrochemical Species Model Properties.

Activates	Physics Models	*Flow*
	Model Controls (child nodes)	Electrochemical Species Components. See Electrochemical Species Components Reference.
	Initial Conditions	Molar Concentration. See Electrochemical Species Initial Conditions.
	Boundary Inputs	See Electrochemical Species Boundary Settings.
	Region Inputs	See Electrochemical Species Region Settings.
	Solvers	Electrochemical Species, Coulomb Force. See Electrochemical Species Solvers and Coulomb Force Solver.
	Field Functions	See Electrochemical Species Field Functions.

Electrochemical Species Model Properties

If you change between using the Electrodynamic Potential model and the Electrostatic Potential model, the Electrochemical Species model property settings for *Charged Species Mobility Dimension* and *Concentration Dimension*, are modified automatically. The Output window displays the modifications, for example:

```
Electrochemical Species Model: Charged Species Mobility Dimension has
been set to Area / Electric Potential Time
```

```
Electrochemical Species Model: Concentration Dimension has been set to
Number Density
```

Electrochemical Species Solver Option

Allows you to specify if the electrochemical species model uses a **Coupled** or **Segregated** solver to solve for the electrochemical species molar concentrations. If you switch between these options, any solver properties that are set are not maintained.

By default, the Electrochemical Species model is set to use the Coupled Electrochemical Species Solver unless you are using a simulation file in STAR-CCM+ v11.04 or later that was saved in an earlier version of STAR-CCM+. In which case, the Segregated solver is used—until you specify otherwise. If you modify the choice of solver, update the solver properties—as properties are not transferred between solvers.

Convection

In transport equations, you can choose from a range of schemes that calculate the convection term at a cell face. This calculation requires STAR-CCM+ to compute the face value of a quantity from the surrounding cell values. The method used for computing this face value has a profound effect on the stability and accuracy of the numerical scheme. For guidance on selecting a convection scheme, see Convective Flux.

- **1st-order**: First-order convection scheme.
- **2nd-order**: Second-order convection scheme.

Secondary Gradients

Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces.

- **On**: Default value. Solves for interior and boundary types of secondary gradient.
- **Off**: Does not solve for either type of secondary gradient.
- **Interior Only**: Solves for the interior secondary gradients only.
- **Boundaries Only**: Solves for the boundary secondary gradients only.

Electroneutrality option

Specifies if the overall ionic charge density is strictly enforced as zero or not. When simulating corrosion applications, maintain the default option (**Scaling**).

- **Not Enforced**: Not Enforced is the default option when using the Electrostatic Potential Model. A uniform zero ionic charge density is not enforced. However, depending on the setup of other parameters, the overall ionic charge density may already equal zero. This option is appropriate for non-corrosion related applications which do not require electroneutrality.
- **Scaling**: Scaling is the default option when using the Electrodynamic Potential Model. This method ensures that the overall ionic charge density is equal to zero by scaling the ionic charge densities for each species proportionally. When the species have equal charge numbers, the total concentration is not changed by scaling. STAR-CCM+ checks the concentration profiles that are set using the concentration dimension property and provides a warning if they are not electrically neutral. This option is appropriate for applications that feature electroneutrality, such as corrosion and etching. See Eqn. (1109).

Transport

Defines how the Electrochemical Species model is coupled to other models and so defines the transport equations that STAR-CCM+ solves.

- **Species <> Electric Potential**: Physical two-way coupling of flux contributions between the Electrochemical Species and Electric Potential models.
- **Electric Potential > Species**: One-way coupling including drift flux / migrative contributions from the Electric Potential model to the Electrochemical Species model.
- **Species > Electric Potential**: One-way transfer of contributions from the Electrochemical Species model to the Electric Potential model.
  - When using the Electrostatic Potential model, the contributions are for the space charge density.
  - When using the Electrodynamic Potential model, the contributions are for the diffusive current.
- **No Electric Potential Coupling**: The Electrochemical Species model and the Electric Potential model act independently.

Charged Species Mobility Dimension

Allows you to set the dimension of the charged species mobility. When the Electrodynamic Potential Model is selected, the default setting for this property is set as **Area Quantity/Energy Time**, and when the Electrostatic Potential model is selected, the default setting for this property is set as **Area/Electric Potential Time**.

- **Area Quantity/Energy Time**: Allows you to specify the mobility of electrochemical species components in dimensions of Area Quantity/Energy Time, or default SI units of $m^2\ kmol/J\ s$

denoted by u_i in Eqn. (1112) and Eqn. (1124).

- **Area/Electric Potential Time**: Allows you to specify the mobility of electrochemical species components in dimensions of Area/Electrical Potential Time or default SI units of m^2/Vs.

  denoted by K_i in Eqn. (1124).

Concentration Dimension

Allows you to specify how the concentration of molecules / particles is defined. When the Electrodynamic Potential Model is selected, the default setting for this property is set as **Molar Concentration**, and when the Electrostatic Potential model is selected, the default setting for this property is set as **Number Density**. If the Electroneutrality property is set to Scaling, STAR-CCM+ checks the concentration profiles that are set and provides a warning if they are not electrically neutral.

- **Molar Concentration**: Specifies the concentration of molecules/particles using the molar concentration in $k\,mol\ /\ m^3$
- **Number Density**: Specifies the molecule/particle concentration in $1\ /\ m^3$ using the actual number of particles—which is the molar concentration multiplied by the Avogadro constant, N_A

Initial Conditions

Molar Concentration

Allows you to specify a composite or constant value for the molar concentration, c_i for species i in Eqn. (1105), of each electrochemical species component that is specified under the Electrochemical Species model.

Boundary Settings

Wall Boundary

Wall Electrochemical Species Option

The wall electrochemical species option for a boundary allows you to define what happens at that boundary regarding electrochemical species.

Method	Corresponding Physics Value Nodes
Impermeable Specifies zero flux at the boundary.	None
Specified Flux You can specify the *Value* as a **Constant** or **Composite** array. The physics condition, **Wall Electrochemical Species Flux Derivative Option** also becomes available to specify.	One of the following nodes appears depending on what you set for *Concentration Dimension* on the Electrochemical Species model. You can specify the *Value* as a **Constant** or **Composite** array. **Molar Concentration Flux** **Number Density Flux**

Specified Value	One of the following nodes appears depending on what you set for *Concentration Dimension* on the Electrochemical Species model. You can specify the *Value* as a **Constant** or **Composite** array. **Molar Concentration** **Number Density**

Wall Electrochemical Species Flux Derivative Option

The Wall Electrochemical Species Flux Derivative Option allows you to specify the derivative of the Molar Concentration Flux of species i (or Number Density Flux) with respect to the molar concentration of species i (or number density) at the wall. The Concentration Flux Derivative represents:

$$\frac{d\mathrm{N}_{n,s,\,i}}{dc_i}$$

and the Number Density Flux Derivative represents:

$$\frac{d\mathrm{N}_{n,s,\,i}}{dn_p}$$

where $\mathrm{N}_{n,\,s,\,i}$ is the molar concentration flux of electrochemical species, i (or number density) at the wall, c_i is the molar concentration of electrochemical species i at the wall, and n_p is the number density at the wall.

This derivative is used for linearization.

This option is available when the **Wall Electrochemical Species Option** boundary condition is set to **Specified Flux**.

Method	Corresponding Physics Value Nodes
None The derivative of the Concentration Flux is not calculated. This is the default option.	None
Specified Allows you to specify the concentration flux derivative that is used for linearization. See Concentration Flux Derivative Formulation.	**Concentration Flux Derivative** Allows you to specify the concentration flux derivative as a **Constant** or **Composite** array. For a stable set-up, make sure that the flux derivative has a negative value.

Region Settings

Electrochemical Species Source Option

Method	Corresponding Physics Value Nodes
deactivated	None
activated Allows you to provide additional physics in Eqn. (1107) to represent a specific molar concentration or number density.	**Electrochemical Species Source Jacobian** Use this node to specify the Jacobian, $d\left(S_{c_i}\right)/d\left(c_i\right)$, of each of the defined species as a scalar array profile. **Electrochemical Species Sources** Use this node to specify the species source, S_{c_i}, of each of the defined species as a scalar array profile. The units that are specified are dependent upon the *Concentration Dimension* setting of the Electrochemical Species model.

Electrochemical Species Solver

A Coupled Electrochemical Species Solver and a Segregated Electrochemical Species Solver are available to solve the electrochemical species concentrations independently from solving the electric potential.

You can select which variant of the Electrochemical Species solver to use within the Electrochemical Species model properties. However, when using the Bulk Ion Chemical Reactions model, only the Coupled option is available.

Having the choice of electrochemical species solvers allows you to solve for the electrochemical species concentrations in the most appropriate way, independent of the method that is used to solve any other factor. For example, electric potential is solved for using a segregated approach, however, you can choose a coupled implementation for solving electrochemical species concentrations.

The Segregated Electrochemical Species Solver runs faster than the Coupled Electrochemical Species Solver. However, for certain problems, especially depletion conditions at high current densities, the coupled solver converges when the segregated solver does not.

The Coupled Electrochemical Species Solver is intended for use with the Electrochemical Reactions model or the Bulk Ion Chemical Reactions model. If you use the coupled solver without these models, the runtime increases considerably—without any benefit.

Since elements cannot be created or destroyed, STAR-CCM+ guarantees conservation of each type of atom when using the coupled solver. However, when source terms and boundary fluxes are provided using field functions, the segregated solver does not guarantee the conservation of elements. Using the segregated approach, the source terms that are computed for each electrochemical species concentration are not computed consistently using the concentrations from the same Newton iteration. Therefore, elements are not conserved at each Newton iteration when field functions are involved with the segregated approach. To guarantee conservation of elements, either use the coupled solver, or use the segregated solver and run Newton's method until it is fully converged for a steady simulation, or for each time-step in an unsteady simulation.

If necessary, you can also stop a simulation and change between the coupled and segregated solver options. Since the Coupled Electrochemical Species Solver provides stability, you can start a simulation with this solver. When the simulation begins to converge, you can change to the Segregated Electrochemical Species Solver—which runs quicker. Switching between solver options removes any solver settings that are defined. Therefore, if you switch solver options, update the solver settings.

Note: For issues involving the Electrochemical Species Solvers, try reducing the under-relaxation factor on the Electric Potential Solver.

The convergence of the coupled solver varies slightly depending on the ordering of the electrochemical species, since the algebraic multigrid solver is constructed based on the matrix components of the first mixture component.

Both implementations, the Coupled Electrochemical Species Solver and the Segregated Electrochemical Species Solver, have the same properties and expert properties.

Under-Relaxation Factor

In order to promote convergence, this property is used to under-relax changes of the solution during the iterative process. If residuals show solution divergence and do not decrease, reduce the under-relaxation factor for the relevant solvers.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Coulomb Force Solver

You can use the Coulomb Force physics model in addition to the Electrochemical Species and Electrostatic Potential physics models to simulate electrohydrodynamic forces that arise from local ionic charge densities. See Modeling Ionic Species Flux. When the Coulomb Force physics model is used, the Coulomb Force Solver becomes available. The Coulomb Force Solver node allows you to specify an Under-Relaxation factor which

you can adjust to help improve stability when simulating ionic wind simulations or plasma actuators. See Coulomb Force Formulation.

Under-Relaxation Factor

In order to promote convergence, this property is used to under-relax changes of the solution during the iterative process. If residuals show solution divergence and do not decrease, reduce the under-relaxation factor for the relevant solvers.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Electrochemical Species Field Functions

Charge Number of [Species]

Charge number, z_i of species i. A separate scalar field function is available for each species.

Diffusion Current Density

The second term in Eqn. (3609).

Ionic Charge Density

The ionic space charge density, ρ_{ion}. Scalar function.

Molar Concentration of [Species]

The molar concentration, c_i, of species i in $k\,mol\ /\ m^3$. A separate scalar field function is available for each species.

Molecular Diffusivity of [Species]

Diffusivity, D_i, of species i. A separate scalar field function is available for each species.

Number Density of [Species]

Displays the number of particles in $1\ /\ m^3$. Calculated by the molar concentration multiplied by the Avogadro constant, N_A. A separate scalar field function is available for each species.

The following field functions are available when using the Electrochemical Species model and Temporary Storage is activated. Using field functions to visualize these quantities can help to diagnose if something is going wrong at a flow boundary, say, where large gradients are not expected. See, Turning On Temporary Storage.

Molar Concentration Residual

The residuals are the imbalance in the discretized equations. Scalar field function.

Molar Concentration Correction

Corrections are added to the solution of the previous iteration to get the updated solution. Scalar function.

Molar Concentration Gradient

Gradients are used to compute values such as diffusion and strain rates. Vector function.

Molar Concentration Recon

Reconstruction gradients are used to interpolate cell values to faces. Vector function.

Molar Concentration Ap

The Ap coefficients are the diagonal coefficients in the matrix of the linear system. Vector function.

Electrochemical Species Components Reference

The Electrochemical Species model provides the Electrochemical Species Components node which you use to define the properties of each electrochemical species.

Electrochemical Species Components Reference

Example Node Path	**Continua** > **Physics 1** > **Models** > **Electrochemical Species** > **Electrochemical Species Components**	
Specific Right-Click Actions	See Electrochemical Species Components Right-Click Actions.	
Activates	Model Controls (child nodes)	[Electrochemical Species]
	Materials	See: [Electrochemical Species] Materials and Methods

Electrochemical Species Components Right-Click Actions

Select Mixture Components

Opens the *Select Mixture Components* dialog, which allows you to select electrochemical species that are available in the material database. See Material Databases.

Reorder Mixture Components

Electrochemical species appear under the **Electrochemical Species Components** node in the order that you select them in the material database. This option opens a dialog which allows you to drag-and-drop electrochemical species to rearrange the order in which they appear in the simulation tree.

Import Species

Displays the *Select file and material database* dialog. The dialog allows you to select a file that contains the required electrochemical species, and specify the Material Database to which the species are added.

Export Mixture Components to a file

Allows you to export and save the electrochemical species components that are specified.

[Electrochemical Species] Properties

Database Material

Displays the read-only description of the material that is chosen for this electrochemical species from the materials database.

ID

Displays a read-only ID number which is unique to this electrochemical species in the surface reaction.

Minimum Allowable Concentration

Threshold value for the molar concentration, c_i, of this species that is used when the molar concentration drops below it. This value overwrites lower values for molar concentration, such as those that are defined as boundary values or initial conditions. Useful to avoid the molar concentrations of electrochemical species reaching values that are close to zero.

[Electrochemical Species] Right-Click Actions

Replace With

Opens the *Replace Mixture Component* dialog which allows you to select a different electrochemical species from the material database.

[Electrochemical Species] Material Properties and Methods

Charge Number

The charge number, z_i of the electrochemical species in Eqn. (1106).

Elemental Composition

The numbers and types of atoms which form the electrochemical species.

Method	Corresponding Method Node
Elemental Composition Method	The **Elemental Composition Method** > **Atoms** node allows you to specify more than one type of atom in an electrochemical species component. To add an atom, right-click the **Atoms** node and select **New Atom**. The *Add atom to elemental composition* dialog allows you to select the type and quantity of atoms in the [Electrochemical Species Component]. An **[Atom]** sub-node is created for each type of atom that is selected and the *Molecular Weight* property displays the combined atomic masses of the atoms within the electrochemical species component.

Mobility

Proportionality factor between the drift velocity that is induced on charged particles and the associated electric field.

Method	Corresponding Method Node
Nernst-Einstein: STAR-CCM+ calculates the mobility of the charged particles differently depending on the dimensions that are used for mobility. See Using the Nernst-Einstein Method for Charged Species. Nernst-Einstein is only available to select when not set for Molecular Diffusivity.	**Nernst-Einstein**

Molecular Diffusivity

Proportionality factor between the diffusion velocity that is induced on electrochemical species and the associated molar concentration gradient.

Method	Corresponding Method Node
Nernst-Einstein: STAR-CCM+ calculates the molecular diffusivity, D_i, of the charged particles using Eqn. (1112). See Using the Nernst-Einstein Method for Charged Species. Nernst-Einstein is only available to select when not set for Mobility.	**Nernst-Einstein**

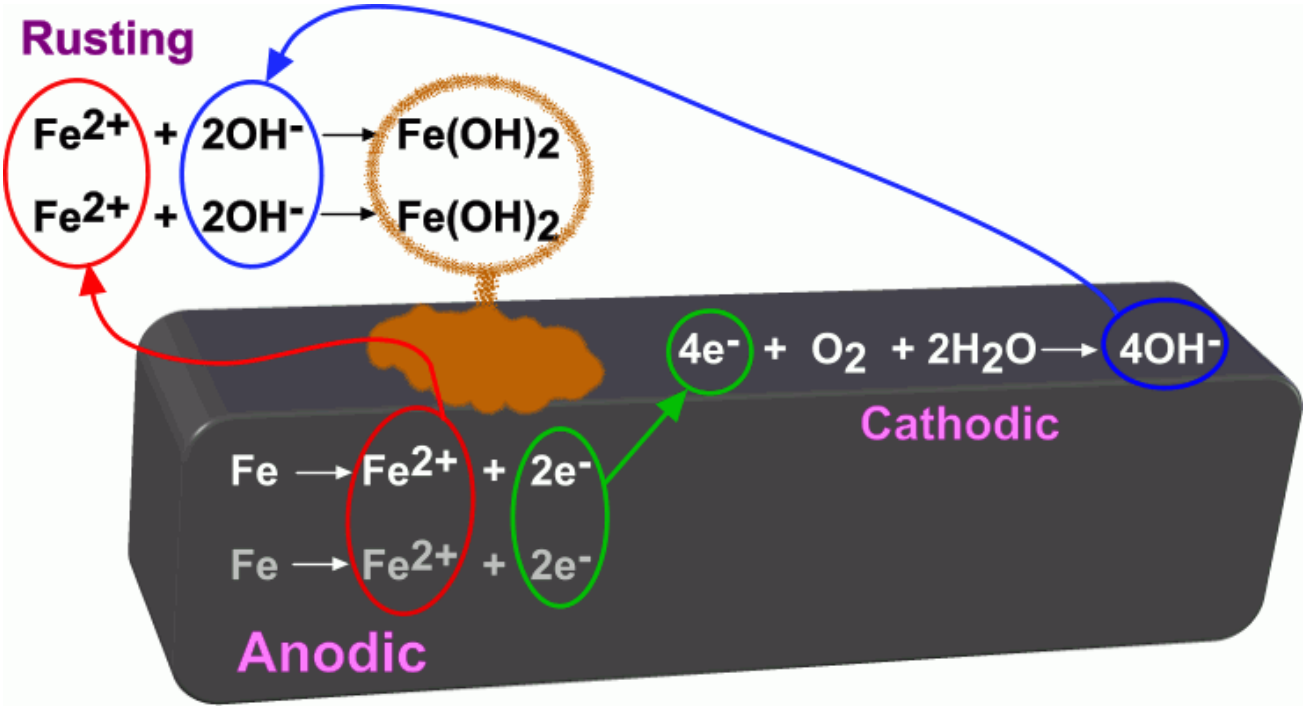
Electrochemical Surface Reactions

When modeling electrochemistry, STAR-CCM+ provides the Electrochemical Reaction model which allows you to model charge transfer reactions at electrode surfaces.

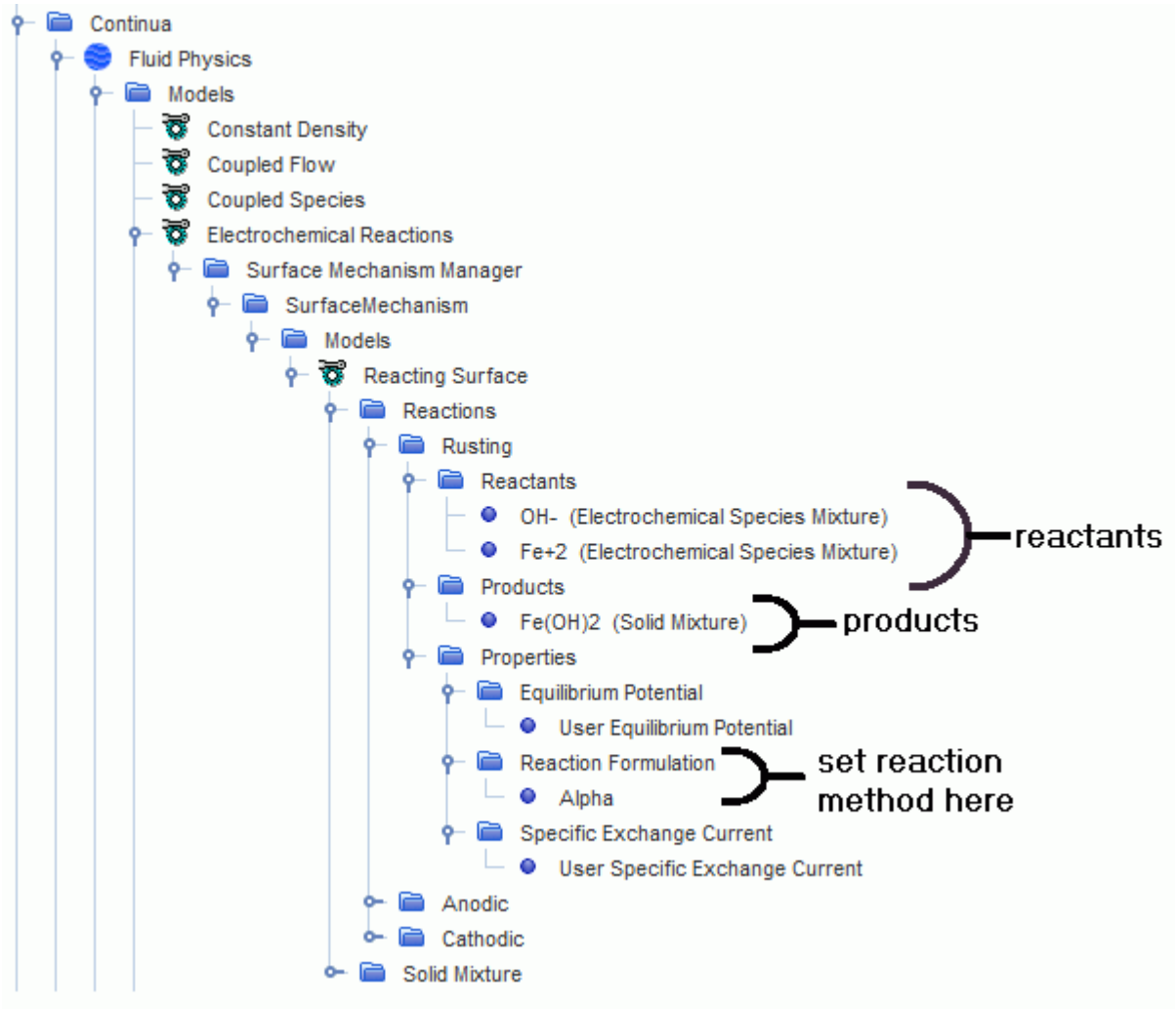
In electrochemistry simulations, electrochemical reactions take place at surfaces, where electrons are consumed or produced. The rate of reaction depends on the concentrations of reactants and products as well as the potential difference across the surface.

Electrochemical reactions that lose electrons are known as oxidation or anodic reactions. Reactions that gain electrons are known as reduction or cathodic reactions. Often these reactions occur together and the electrons that the anodic reactions create, drive the cathodic reactions.

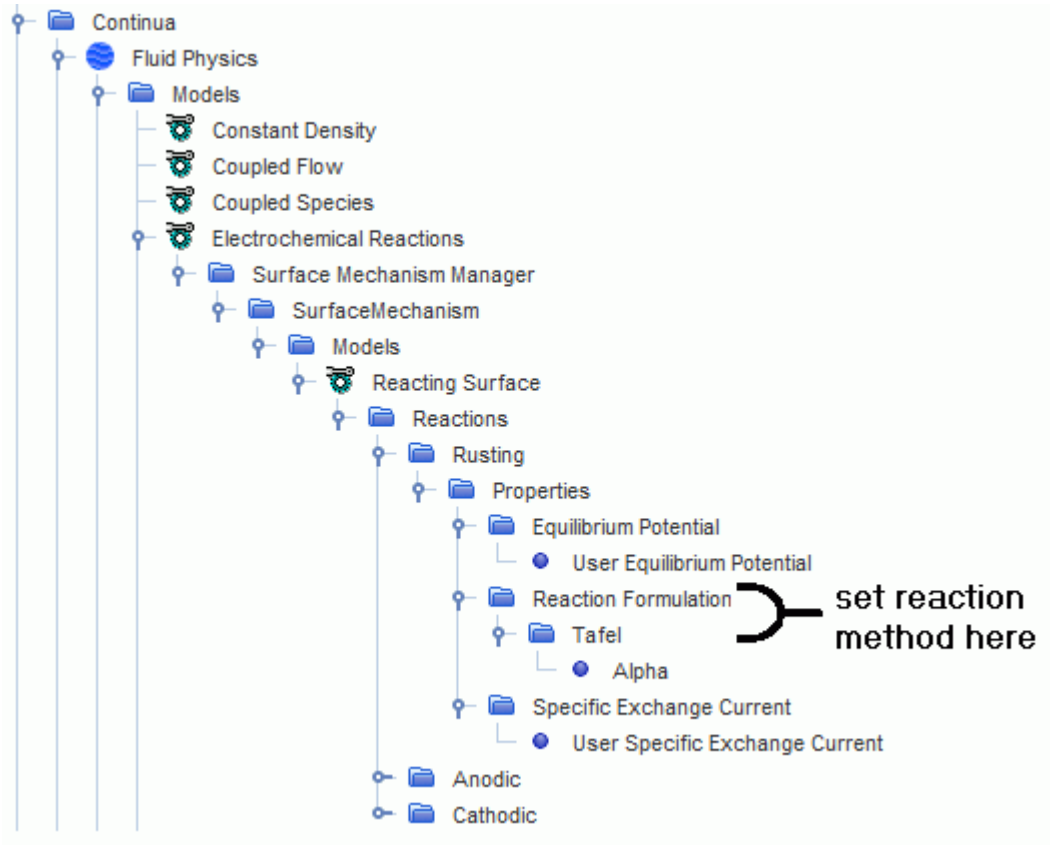
The following example shows the electrochemical reactions that occur within, and surrounding, an iron surface that is in contact with water and oxygen.



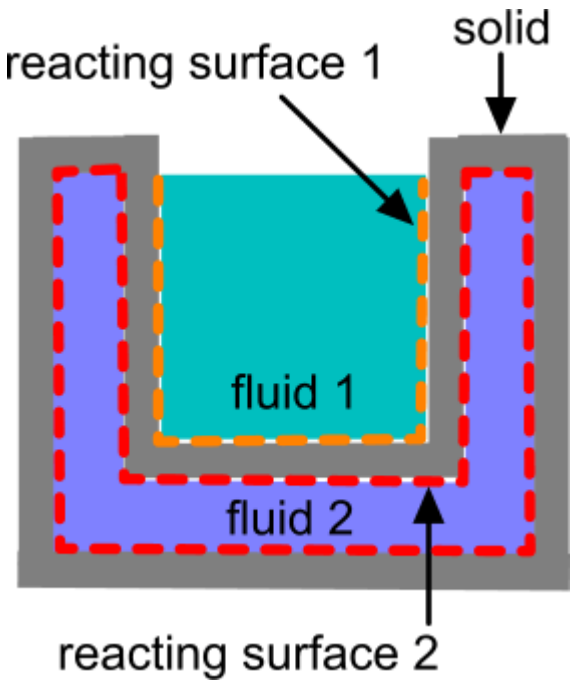
When using the Electrochemistry Reacting model with the Electrochemical Species model, you can specify reaction methods and create mechanisms for surface reactions which include reactant and product species that you specify. Specifying species allows you to include the effects of the ion concentrations on reactions. You can choose a coupled or segregated approach to solve for electrochemical species concentrations.



However, if you are only concerned with electric currents and not concerned with the effects of the ion concentrations on reactions, you can model the same reaction excluding the Electrochemical Species model. Excluding the Electrochemical Species model allows you to create surface mechanisms in which you specify surface reactions without specifying the reactant and product species.



In some cases, the variation of electric potential through the solid is insignificant compared to the variation of electric potential through the fluid. You can choose to account for the electric potential variation only within the fluid by defining the reactions at boundaries. However, you can account for the electric potential variation within both the solid and the fluid at a reacting surface by defining the reactions at interfaces. This feature is useful when the electrical resistance through the solid is significant relative to the electrical resistance through the fluid. At the fluid / solid interface, the surface mechanism is defined within the fluid continuum and not the solid continuum. This approach allows you to model solids which contact multiple fluids:



This feature is compatible with solid shell regions. However, you can only define reactions at interfaces between solid and fluid physics continua, you cannot define the reacting surface interface between two solids. See Modeling Electrochemical Surface Reactions.

Contents:

Modeling Electrochemical Surface Reactions
Electrochemical Surface Reactions Reference

Modeling Electrochemical Surface Reactions

STAR-CCM+ allows you to model electrochemical reactions on boundary surfaces around the physics continuum for the fluid, and also on interface surfaces between a solid continuum and the fluid. In both cases, you can choose to model the production and consumption of electrochemical species from reactions.

Follow this procedure to model electrochemical surface reactions.

Note: A double precision version of STAR-CCM+ is required to model the transport of electrochemical species when coupling the electrochemical species model with an electric potential model.

To simulate electrochemical surface reactions, some applications require that you define several solid and fluid continua. Make sure that you select the necessary models for continua on both sides of the reactive interface.

1. Set up the fluid physics continua.
   a) For physics continua that represents a fluid, select the following models, with **Auto-Select recommended models** activated:

Group Box	Model
Space	Select one
Time	Select one
Material	Either: • **Liquid** • **Multi-Component Liquid** • **Gas** • **Multi-component Gas** • **Eulerian Multiphase**
Reaction Regime (available when Multi-Component Liquid or Multi-Component Gas is selected)	**Non-Reacting**
Flow (single phase flows)	Either: • **Segregated Flow** • **Coupled Flow**
Eulerian Multiphase Model (multiphase flows)	Select one The **Volume of Fluid (VOF)** and **Eulerian Multiphase Mixture** models provide energy models which calculate temperature for the combined mixture of phases. The **Multiphase Segregated Flow** model provides the **Phase Coupled Energy** model which solves for temperature per phase. See Eqn. (1133).
Equation of State (single phase flows)	Select one (**Constant Density** is the most appropriate choice for liquids with very small temperature changes)
Viscous Regime	Select one
Optional Models	**Electrochemistry**
Electrochemistry	**Electrochemical Reactions** **Electromagnetism** (selected automatically) **Electrodynamic Potential** (selected automatically)

b) To include electrochemical species in the surface reaction mechanisms, select the **Electrochemical Species** model from the *Optional Models* group box. The Electrochemical Species model is not available when the Eulerian Multiphase model is selected.

c) Select any extra *Optional Models* that you require for the fluid physics continuum.

- Select an Energy model. If you want to set a constant temperature without solving an additional transport equation, select the **Segregated Fluid Isothermal** model. If no temperature model is

selected and the Electrochemical Species model is selected, the Electrochemical Species model runs with a constant temperature of 293.15K.

- If you want to model heat contributions that are due to electrochemical reactions at interfaces (except sorption), select an energy model and the **Electrochemical Reaction Heating** model for all continua which interface a reactive surface. See Electrochemical Reaction Heating Model Reference. To model heat release due to ion transport in bulk phases and due to resistances at reacting surfaces, you can also select the **Ohmic Heating** model for all ionically conductive domains, see Ohmic Heating Model Reference. Make sure that you select heating and energy models for all physics continua which are next to a reactive interface.

d) Click **Close**.

When simulating fluid domains in which the diffusion at the flow boundaries is high, and not imperative to include in the calculations, you can choose to neglect this diffusion and only account for the diffusion of species within the domain. This feature is useful when reviewing the mass conservation of fuel cells.

2. To neglect diffusion at flow boundaries, within the fluid continuum, multi-select all segregated flow, species, and energy models and deactivate *Flow Boundary Diffusion*.
3. If the simulation includes a solid on which surface reactions take place, add a physics continuum for this solid.

   a) Select the following models (with **Auto-Select recommended models** activated):

Group Box	Model
Space	Select one
Time	Select one
Material	**Solid**
Optional Models	**Electrochemistry**
Enabled Models	**Electrochemical Reactions** (selected automatically)
	Electromagnetism (selected automatically)
	Gradients (selected automatically)
	Electrodynamic Potential (selected automatically)

   b) Select any extra *Optional Models* that you require for the solid physics continuum.

   For example

   - If you are modeling electrochemical reactions in a fluid or porous continuum which include ions from the solid continuum, such as in Solid Oxide Fuel Cells (SOFCs), select the **Solid Ion** model for the solid continuum.
   - An energy model.
   - **Electrochemical Reaction Heating**
   - **Ohmic Heating**

   c) Click **Close**.
4. If you are using the Eulerian Multiphase model, set up the phases and define any phase interactions. See Defining Eulerian Phases and Defining Phase Interactions.

5. If you have a solid continuum present in the simulation, you can optionally choose to solve for the electric potential within the solid material. Otherwise, set a constant electric potential value on the surface that bounds the fluid:
   - To set a constant electric potential on the fluid bounding surface:
     1. Expand the **Regions** > **[Fluid Region]** > **Boundaries** > **[Boundary]** node.
     2. Select the **Physics Conditions** > **Surface Mechanism Option** node and set *Mechanism* to **[Surface Mechanism]**.
     3. Select the **[Boundary]** > **Physics Values** > **Electrode Electric Potential** node and specify the electrode electric potential value.
   - To account for the electric potential variation within the fluid and the solid at a reacting interface, select the **Interfaces** > **[Interface]** > **Physics Conditions** > **Surface Mechanism Option** node and set *Mechanism* to **[Surface Mechanism]**.

   Note: In contrast to using a surface mechanism at boundaries, there is no option to set the Electrode Electric Potential within the interface physics values. The Electric Potential value for the electrode is obtained internally from the solid.

When modeling electrochemical surface reactions without defining electrochemical species, you can choose to account for electrical resistance at boundaries or interfaces.

This feature is also useful when modeling some applications in which electrochemical species are defined, such as catalyst resistance in fuel cells, corrosion products or organic tissues at interfaces, and SEI interfaces in batteries.

6. To account for electrical resistance, choose one of the following options:
   - When a surface mechanism is set on the interface, specify the electrical resistance at the interface. Select the **Interfaces** > **[fluid/solid interface]** > **Physics Values** > **Electrical Resistance** node and in the *Properties* window, specify the electrical resistance.
   - When a surface mechanism is set on the boundary, specify the electrical resistance at the boundary. Select the **[boundary]** > **Physics Values** > **Electrical Resistance** node and in the *Properties* window, specify the electrical resistance.

   When more than one fluid region contacts the solid region, you can specify the electrical resistance at each boundary or interface independently.

When defining electrochemical surface reactions, you can include gas or liquid components and electrochemical species components in the same reaction. Before defining the reactions, you specify the segregated material species and electrochemical species that are present in each continuum.

7. Define the material species.
   See either:
   - Managing Single-Component Materials
   - Managing Multi-Component Materials
8. If the Electrochemical Species model is selected, create the solid species that represent the solid surface in the electrochemical surface reactions:
   a) Expand the **Electrochemical Reactions** > **Surface Mechanism Manager** > **[Surface Mechanism]** > **Models** node.
   b) Right-click the **Reacting Surface** > **Solid Mixture** > **Solid Species** node and select **Select Mixture Components**.

c) Select all of the solid species that represent the solid surface in the electrochemical surface reactions. If you cannot find a solid species component, you can select a similar solid species component and modify its material properties to change the type and quantity of atoms.

d) Click **Apply**, then **Close**.

If you did not select the Electrochemical Species model, the solid material properties are taken from the solid physics continuum.

If you selected the Electrochemical Species model, specify the electrochemical reaction species components and the method for solving the concentrations of the electrochemical species.

9. To modify the Electrochemical Species model, do the following:

   a) Right-click the **Electrochemical Species** > **Electrochemical Species Components** node and select **Select Mixture Components**.

   b) Select all of the electrochemical species that represent the reactants and products in the electrochemical surface reactions. If you cannot find a complex electrochemical species component, you can either:

      - Close the dialog, add electrochemical species to the database, then reopen the *Select Mixture Components* dialog, and select the species. See Modifying a Copy of the Material Database.
      - Select a similar electrochemical species component and modify its material properties to change the charge, and the type and quantity of atoms.

   c) Click **Apply**, then **Close**.

   d) To modify the composition or properties of an electrochemical species component, expand the **Electrochemical Species Components** > **[Electrochemical Species Component]** node and edit the Material Properties as required. See Electrochemical Species Component Reference.

   e) Select the **Electrochemical Species** node and specify the *Electrochemical Species Solver Option* as **Coupled** or **Segregated**.

      The Coupled Electrochemical Species Solver provides the best stability and is more capable of reaching convergence at high current densities than the Segregated Electrochemical Species Solver. However, the Segregated Electrochemical Species Solver runs faster at low current densities.

If you selected the **Solid Ion** model, specify the electrochemical species components in the solid continuum which participate in the electrochemical reactions.

10. To specify the solid ion electrochemical species components, in the solid continuum, right-click the **Solid Ion** > **Electrochemical Species Components** node and select **Select Mixture Components**. You can add and amend species in the same way as described above for the Electrochemical Species model.

If you use the Electrochemical Species model with the Electrodynamics Potential model, you adjust the electrical conductivity.

11. Expand the **Models** > **[Material]** > **Material Properties** node, select the **Electrical Conductivity** node, and set *Method* to **Electrochemical Species**.

Create the electrochemical surface reaction mechanisms. When using the electrochemical reactions model with the Eulerian Multiphase model the model behaves as non-phasic, which means that there is not a reaction mechanism or electric potential per phase, but for the "mixture" of phases.

When specifying the reactants and products, make sure that:

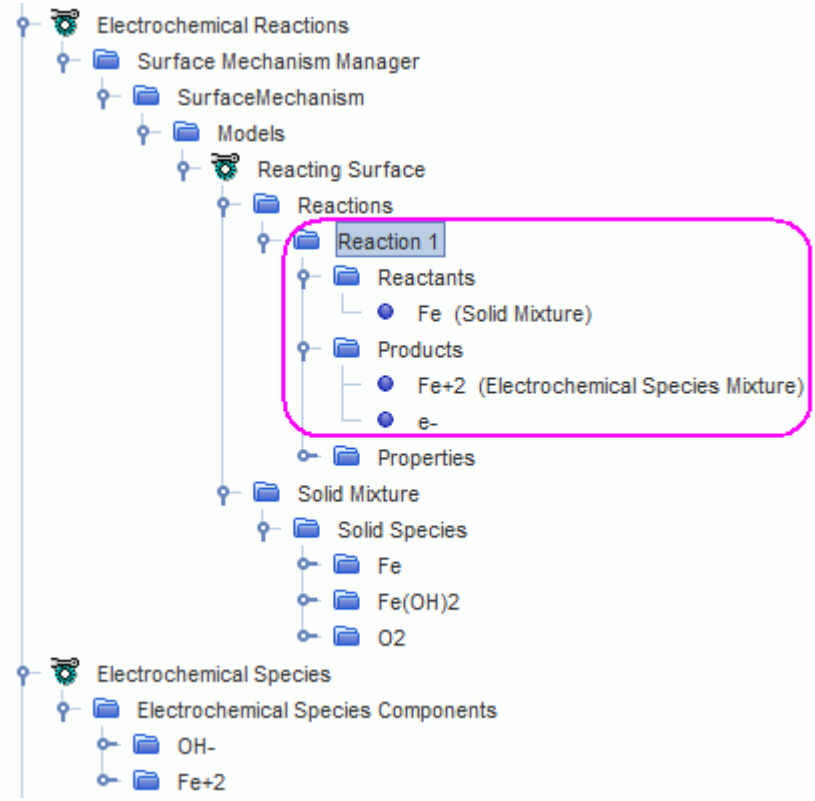
- There are the same number of each type of atom in the reactants and products—in any reaction, matter cannot be created or destroyed.

- The overall charge from the reactants and products must balance so that the total electric charge that is created in each reaction is zero.
- Electrons are specified as reactants or products—in electrochemical reactions, electrons must be produced or consumed to generate an electric current.

12. Expand the **Continua** > **Physics 1** > **Models** > **Electrochemical Reactions** node then right-click the **Surface Mechanism Manager** node and select **New Surface Mechanism**.
13. Right-click the **Surface Mechanism Manager** > **Surface Mechanism** > **Models** > **Reacting Surface** > **Reactions** node and select **New Reaction**. Then specify the properties of each reaction. See Electrochemical Reaction Model Reference.
14. If you are simulating Solid Oxide Fuel Cells, you are required to set some specific properties. Define the equilibrium potentials, thermodynamic polynomial data, exchange current densities, the electrolyte conductivity, the electric power flux, and the mass fluxes. For more information, see the Electrochemistry: Solid Oxide Fuel Cell Tutorial.

If the Electrochemical Species model is selected, or if the Solid Ion model is selected for the corresponding solid continuum, continue to specify reactants and products.

15. Right-click the **Reaction 1** > **Reactants** node and select a reactant. Repeat this step until all reactants are selected.
16. Right-click the **Reaction 1** > **Products** node and select a product. Repeat this step until all products are selected.
    For example:



17. Finalize any remaining settings for the region types and the conditions at the regions, boundaries, and if necessary, the interfaces.

When using the Electrochemical Species model, porous baffle interfaces are not allowed.

18. Define the Initial Conditions.
19. Set up Scenes and Plots to visualize the solution.
    For example, you can visualize the following field functions, among others, in a scalar scene or plot them on an x-y plot:
    - charged species mobility of specific electrochemical species
    - electric current density
    - electric potential
    - electrochemical reaction rate of a specific reaction mechanism
    - migration flux of specific electrochemical species
    - molar concentration of specific electrochemical species
    - number density of specific electrochemical species
    - molecular diffusivity of specific electrochemical species
    - specific electrochemical reaction heat surface source

    Tip: To check energy conservation, add reports for electric power flux and energy loss then create monitors and a single plot from these reports. For more information, see the Electrochemistry: Solid Oxide Fuel Cell Tutorial.

20. Run the simulation.

Contents:

Troubleshooting Electrochemical Surface Reactions

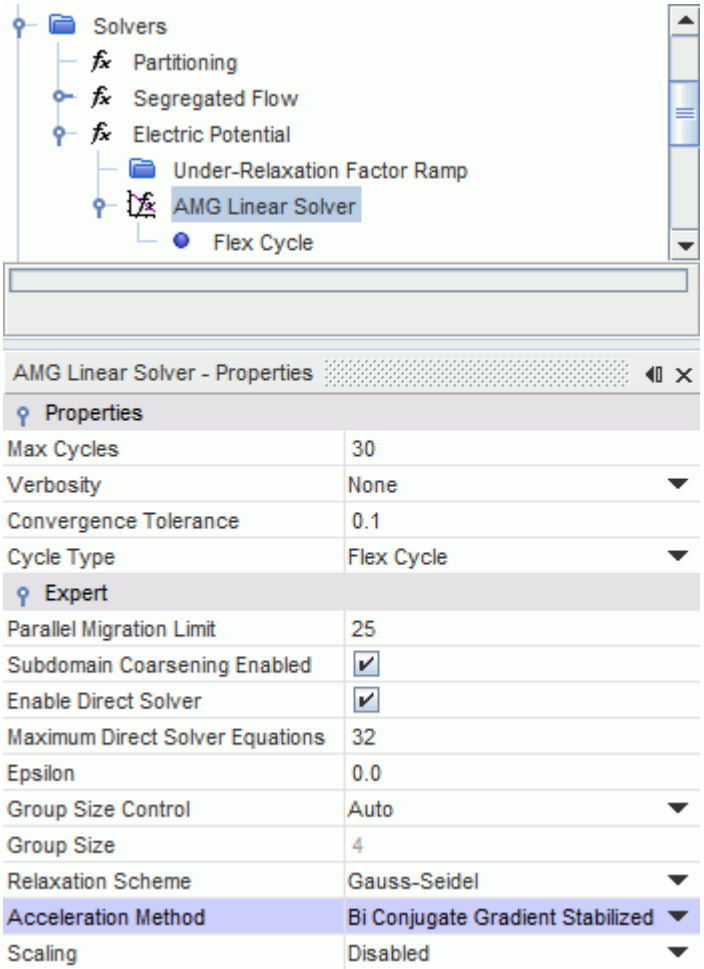
Troubleshooting Electrochemical Surface Reactions

If residuals do not converge while running a surface reaction simulation, you can make adjustments to improve the solution and achieve convergence.

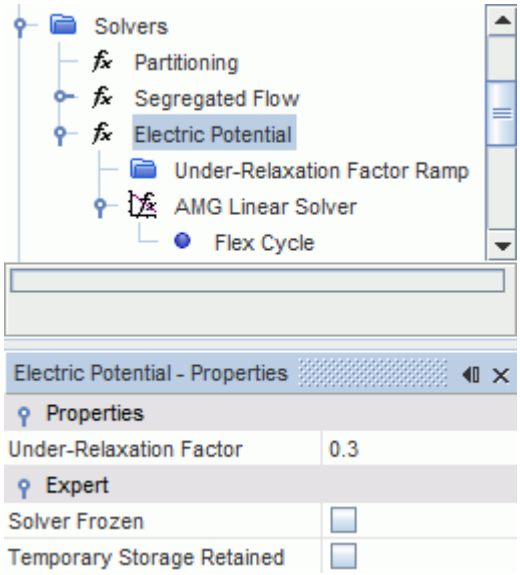
Since the boundary conditions in surface reaction simulations are non-linear, you can adjust the solver settings to account for the non-linear conditions:

To Adjust the Solver Parameters:

1. Select the **Solvers** > **Electric Potential** > **Under-Relaxation Factor Ramp** > **AMG Linear Solver** node.
   a) Activate *Enable direct Solver*.
   b) Set *Acceleration method* to **Bi Conjugate Gradient Stabilized**.



2. Select the **Solvers** > **Electric Potential** node and reduce the *Under-Relaxation Factor.*



Electrochemical Surface Reactions Reference

This section provides detailed information about the electrochemistry models that are available for modeling electrochemical surface reactions.

Contents:

Electrochemistry Model Reference Topic
Electrochemical Reaction Model Reference
Reacting Surface Model Reference
Electrochemical Reaction Heating Model Reference
Solid Ion Model Reference

Electrochemistry Model Reference Topic

Electrochemistry Model Reference

Theory	See Electrochemistry Formulation.	
Provided By	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Electrochemistry**	
Requires	*Space*: one of **Three Dimensional**, **Two Dimensional**, **Axisymetric** *Material*: one of **Gas**, **Liquid**, **Multi-Component Gas**, **Multi-Component Liquid**, **Solid**	
Activates	Physics Models	When *Material* is set to **Solid**, the following physics models are selected by default: • Electrochemical Reactions • Electromagnetism • Gradients • Electrodynamic Potential

Electrochemical Reaction Model Reference

The Electrochemical Reactions model allows you to create surface reaction mechanisms to which you can add reactions and specify reaction methods.

You can use the Electrochemical Reactions model to simulate electrochemical surface reactions that occur during a wide range of electrochemical applications, including:

- Galvanic Corrosion
- Cathodic Protection
- Electroplating
- Wet Etching
- Fuel Cells

- Paint Delamination
- Metal-Liquid Interface Reactions

There are two variants of the Electrochemical Reactions model which you can use depending on what you are modeling.

- Modeling Electrochemical Surface Reactions Without Species requires the following models:
  - Electrodynamic Potential model
  - Electrochemical Reactions model

  When using this option, the Electrochemical Reactions model computes an electric current and electric current derivative from the electrochemical reactions that you define, and passes the information to the Electrodynamic Potential model.

  Use this option when you want to model the secondary current distribution and ignore the effect of species concentrations.
- Modeling Electrochemical Surface Reactions With Species requires the following models:
  - Electrodynamic Potential model
  - Electrochemical Reactions model
  - Electrochemical Species model

  When using this option, the Electrochemical Reactions model computes contributions to the electric current from:
  - species concentrations
  - electrochemical species flux
  - flux derivatives

  The electric current is passed to the Electrodynamic Potential model, then the species flux and flux derivatives are passed to the Electrochemical Species model.

  Use this option when you want to model electrochemical surface reactions which take into account the electrochemical species flux due to electrochemical species concentrations.

Note: The Electrochemical Reactions model only considers concentrations of species components which physically exist in the fluid domains. The concentrations of species that are defined within solid physics models are not considered in rate computations.

Electrochemical Reaction Model Reference

Theory	See Electrochemical Surface Reactions Formulation.
Provided By	**[physics continuum]** > **Models** > *Electrochemistry*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Electrochemical Reactions**
Requires	*Space*: any *Material*: any except **Multi-Component Solid** *Optional Models*: Electrochemistry

Properties	Key properties are: Ionically Conductive Domain. See Electrochemical Reactions Model Properties.	
Activates	Physics Models	*Electromagnetism Option*: **Electromagnetism**, *Gradient Metrics*: **Gradients**, *Electrodynamic Potential*: **Electrodynamic Potential**.
	Model Controls (child nodes)	In Fluid and Porous Continua: Surface Mechanism Manager, [Surface Mechanism]
	Boundary Inputs	See Electrochemical Reaction Model Boundary Settings.
	Field Functions	Electrochemical Surface Overpotential, Electrochemical Reaction Rate, Electrochemical Equilibrium Potential, Boundary Species Electrochemical Reaction Flux. See Field Functions.

Electrochemical Reactions Model Properties

Ionically Conductive Domain

Only available for fluid physics continua. Depending on the application that you are modeling, you can set up the simulation so that either the solid or fluid domain represents an electrolyte—the ionically conductive domain. This property is set in the fluid continuum, even if the solid domain represents the electrolyte.

- **Solid**: Specifies that the corresponding solid continuum which uses the Solid Ion model is the ionically conductive domain. Automatically selected when the Solid Ion model is used for the solid continuum which interfaces this fluid domain.
- **Fluid**: Specifies that this fluid continuum is the ionically conductive domain. Automatically selected when the Electrochemical Species model is selected for the fluid continuum and the Solid Ion model is not selected for the corresponding solid continuum.
- **Does not apply**: Neither the fluid or the solid continuum are specified as ionically conductive, since no electrochemical species are defined in either continuum. Automatically selected when the Electrochemical Species model is not selected for the fluid continuum and the Solid Ion model is not selected for the corresponding solid continuum.

Electrochemical Reaction Model Boundary Settings

Boundaries in Fluid or Porous Regions

Surface Mechanism Option

Mechanism	Corresponding Physics Value Nodes
None	None
[Surface Mechanism]	**Electrode Electric Potential** Set as a scalar profile.

Field Functions

Electrochemical Surface Overpotential

Displays the electrochemical surface overpotential η in Eqn. (1128).

Electrochemical Reaction Rate

Displays the electrochemical reaction rate (specific reaction current density), $j_{n,s}$, in:

- Eqn. (1135) when using the Butler Volmer reaction method
- Eqn. (1138) when using the Tafel reaction method
- Eqn. (1140) when using the Tafel Slope (log 10) reaction method
- Eqn. (1141) when using the Transport Limited Tafel Slope (log 10) reaction method

Electrochemical Equilibrium Potential

Displays the electrochemical equilibrium potential U_{eq} in Eqn. (1128) and Eqn. (1142).

Boundary Species Electrochemical Reaction Flux

Available when using both the Electrochemical Reactions model and the Electrochemical Species model. Shows the reaction flux, N_i, in [kmol/m^{2}s] for all electrochemical species and the solid species that are registered for the specific reaction mechanism.

Surface Mechanism Manager

Right-Click Actions

New Surface Mechanism

Creates a Surface Mechanism manager node which allows you to specify details of that surface mechanism. You can create more than one surface mechanism.

Remove Invalid Surface Mechanisms

Allows you to remove surface mechanisms which are invalid ( ).

[Surface Mechanism]

If the surface mechanism appears with a caution symbol ( ), the surface mechanism is invalid. For example, when the Electrochemical Species model is selected and the surface mechanism is defined, the Reaction [n] Type is set to Electrochemical Species. Deselecting the Electrochemical Species model maintains the reaction type and marks the surface mechanism as invalid. You can then create a surface mechanism which is valid and either; keep the invalid surface mechanism to use again (when reverting to the previous model selections), or remove the invalid surface mechanism. Invalid surface mechanisms cannot be selected for use on boundaries or interfaces. If surface mechanisms that contain electrochemical species become invalid, the electrochemical species are removed from the surface mechanism and must be re-selected if the surface mechanism then becomes valid again.

Properties

Id

Read-only property which displays a unique ID number for that surface mechanism.

Right-Click Actions

Import Species

Displays the *Select file and material database* dialog. The dialog allows you to select a file that contains the required species, and specify the Material Database to which the species are added.

Import Species and Reactions

Displays the *Select file and material database* dialog. The dialog allows you to select a file that contains the required species and reaction mechanisms, and specify the Material Database to which the species are added.

Reacting Surface Model Reference

The Reacting Surface submodel is selected automatically when the Electrochemical Reactions model is selected and contains a surface mechanism. This submodel cannot be deselected.

Reacting Surface Model Reference

Theory	See Electrochemical Surface Reactions Formulation.	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Electrochemical Reactions** > **Surface Mechanism Manager** > **SurfaceMechanism** > **Models** > **Reacting Surface**	
Requires	*Material*: one of **Gas**, **Liquid**, **Eulerian Multiphase**, **Multi-Component Gas**, **Multi-Component Liquid** *Flow*: any *Viscous Regime*: **Turbulent** *Turbulence*: **Reynolds-Averaged Navier-Stokes** *Reynolds-Averaged Turbulence*: **K-Epsilon Turbulence**	
Activates	Model Controls (child nodes)	Reactions, Reaction [n], Reactants / Products, Properties, Solid Mixture

Reactions

Right-Click Actions

New Reaction

Creates a **Reaction [n]** sub-node.

Import Reactions

Displays the *Select file and material database* dialog. The dialog allows you to select a file that contains the reaction mechanism, and specify the Material Database that the reaction components are added to.

Child Nodes

Reaction [n]

Properties

Index

Read-only property which displays a unique number for that reaction.

Type

A reaction *Type* is specified by default, depending wether the Electrochemical Species model is selected when the surface mechanism which contains [Reaction n] is created.

Electrochemistry Species

Available when the Electrochemical Species model is selected.

Electrochemistry Potential

Available when the Electrochemical Species model is not selected.

Sorption

Provides the option to select the type of sorption under the **[Reaction]** > **Properties** > **Reaction Formulation** node.

Child Nodes

Reactants / Products

These nodes are only available to modify when the Electrochemical Species model is used with surface mechanisms which have *Type* set to **Electrochemistry Species**. Allows you to specify species for the reactants and products that participate in this surface reaction. Before adding reactants and products here, the reaction components must be specified within the Electrochemical Species model, Solid Species node, and the material model. See Modeling Electrochemical Surface Reactions.

Right-Click Actions

Add Reactant / Add Product

A **[Reaction Component]** sub-node appears for each reactant or product that is specified.

- **Electrochemical Species Mixture** : Allows you to select electrochemical species that are present as reactants or products in the surface reaction. Only electrochemical species that are defined within the Electrochemical Species model are available.

- **Solid Mixture**: Allows you to select solid species that are present as reactants in the surface reaction. Only solid species that are defined within the Solid Species node are available.
- **[H20]**: Adds the specified substance as a reactant. Each substance that is specified in the Liquid or Multi-component Liquid model is available to select as a reaction component.
- **e-**: Adds electrons as a reactant.

Refresh

Refreshes the list of reaction components that are available to select.

Child Nodes

[Reaction Component]

Properties

Stoich.Coeff

The stoichiometric coefficient specifies the molar ratio of this reaction component in the balanced equation for that electrochemical reaction.

Rate Exponent

The rate exponent, γ_i, for this reaction component, as used to calculate the specific reaction current in Eqn. (1135), Eqn. (1138), or Eqn. (1140). You can set rate exponent values for electrochemical species components that are specified in the Electrochemical Species model, or solvent species from the Liquid model or Multi-Component Liquid model. Molar concentrations of all reaction components which physically exist in the fluid domains are used to compute the electrochemical reaction current.

Properties

Child Nodes

Equilibrium Potential

This node becomes unavailable if the Reaction Formulation method is set to **Tabular**.

Properties

Select Method

Allows you to specify how the equilibrium potential, U_{eq}, is determined.

Select Method	Corresponding Child Node
User Equilibrium Potential	**User Equilibrium Potential** Allows you to specify the equilibrium potential, U_{eq}, as a scalar profile.
Nernst Equilibrium Potential Only available when the Electrochemical Species model is selected—with both Reactant and Product species specified.	**Nernst Standard Potential** Allows you to specify the Nernst Standard Potential, E^0 that is used to calculate the equilibrium potential, U_{eq}, in Eqn. (1142), as a scalar profile. When using the Electrochemical Reaction Heating model and specifying the Nernst standard potential as a function of temperature, make sure to specify the corresponding temperature derivative. See Equilibrium Potential T Derivative.

Equilibrium Potential T Derivative

Available when using the Electrochemical Reaction Heating model.

Properties

Select Method

Allows you to specify the temperature derivative of the equilibrium potential, U_{eq} which is used to compute the reversible heat contribution in Eqn. (1143).

Select Method	Corresponding Child Node
User Equilibrium Potential T Derivative	**User Equilibrium Potential T Derivative** Allows you to specify the equilibrium potential temperature derivative, $\frac{dU_{eq}}{dT}$, as a scalar profile.
Nernst Equilibrium Potential T Derivative Available when the Equilibrium Potential property, *Select Method* , is set to **Nernst Equilibrium Potential**.	**Nernst Standard Potential T Derivative** Allows you to specify the Nernst standard potential temperature derivative, $\frac{dE^0}{dT}$, that is used to calculate $\frac{dU_{eq}}{dT}$ in Eqn. (1143). For example, at the cathode (Eqn. (1073)) and anode (Eqn. (1072)) in solid oxide fuel cells.

Equilibrium Concentration Constant

Appears when both

- the *Type* property is set to Sorption under the **Reactions** > **Reaction [n]** node
- the Method is set to **Springer** or **Wu-Li-Berg** under the **Reaction Formulation** node.

Method	Corresponding Child Node
User Specific Equilibrium Concentration Constant	**User Specific Equilibrium Concentration Constant** Allows you to set a scalar profile for the User Specific Equilibrium Concentration Constant, ξ in Eqn. (3678).

Reaction Formulation

Properties

The property that appears here depends upon the option that is selected for the *Type* property that is set under the **Reactions** > **Reaction [n]** node.

Select Method

Available when the *Type* property is set to Electrochemistry Potential or Electrochemistry Species under the **Reactions** > **Reaction [n]** node.

Allows you to specify how the specific reaction current is calculated.

The sub-nodes that appear depend on whether the electrons are specified as reactants or products.

Select Method	Corresponding Child Node
Butler-Volmer When this method is selected, the boundary conditions at the interface between the metal and the electrolyte are set automatically. However, you can adjust the boundary conditions manually if necessary.	**Butler-Volmer** Allows you to specify scalar values for the apparent anodic charge transfer coefficient, α_a, for the anodic current that is directed from the solid towards the liquid, and apparent cathodic charge transfer coefficient, α_c, for the cathodic current that is directed from the liquid towards the solid. These values are used in Eqn. (1134). Using the Butler-Volmer method without the Electrochemical Species model means that the specific reaction current is calculated without considering the concentrations of the electrochemical species—only the anodic and cathodic current are accounted for. Using the Butler-Volmer method with the Electrochemical Species model means that the specific reaction current is calculated using the molar concentrations of electrochemical species as well as the anodic and cathodic currents. The Butler-Volmer equation Eqn. (1134) combines Tafel equations Eqn. (1138) for both anodic and cathodic reactions which occur when reactions are reversible.
Tabular Selecting this method removes the **Properties** > **Equilibrium Potential** and **Properties** >	**Current Voltage Characteristic** Provides the **Tabular Polarization Curve** sub-node which allows you to select a previously imported table for the Tabular Polarization

Specific Exchange Current nodes.	Curve. You set the specific electric current and equilibrium potential in the Tabular Polarization Curve *Properties*. In contrast to other reaction formulation methods, the tabular polarization curve method lets you import arbitrary forms of polarization curves. You can import the tabular polarization curve by right-clicking the **Tools** > **Tables** node, choosing **New Table** > **File Table**, and selecting the appropriate file. The imported table then becomes available to select within the Tabular Polarization Curve properties. The boundary conditions are then available to specify at the interface between the reacting surface and the electrolyte. When modeling corrosion, STAR-CCM+ uses the polarization curves to calculate the corrosion potential for each metal when the current is zero. Therefore, the tabular data must include positive and negative values for current along with the corresponding monatonically increasing potential values.
Tafel You can use the Tafel method for reactions that exhibit considerable currents in only one direction—either anodic or cathodic. Small apparent transfer coefficients or reactions operating solely in either positive or negative surface overpotential can cause one-directional current. The options that are available when selecting the Tafel method depend on whether the electrons are specified as reactants or products.	**Tafel** Appears when electrons are not registered as reactants or products. Allows you to set the *Reaction Direction*. **Anodic** specifies the direction of the reaction current as anodic—from the solid towards the liquid. **Cathodic** specifies the direction of the reaction current as cathodic—from the liquid towards the solid. A **Tafel** > **Alpha** sub-node also appears which allows you to specify a value for the apparent charge transfer coefficient. **Alpha** Appears when electrons are registered as reactants or products. Allows you to specify a scalar value for the apparent charge transfer coefficient. The direction of the current is determined from settings within the Electrochemical Species model.
Tafel Slope (log 10) Creates a **Properties** > **Specific Exchange Current Y-Intercept (log10)** sibling node.	**Tafel Slope (log 10)** Appears when electrons are not registered as reactants or products. Allows you to set the *Reaction Direction*. • **Anodic** specifies the direction of the reaction current as anodic—from the solid towards the liquid. • **Cathodic** specifies the direction of the reaction current as cathodic—from the liquid towards the solid.

	A **Tafel Slope (log10)** > **Beta (log10)** sub-node also appears which allows you to specify a combined term β for the combined anodic or cathodic apparent transfer coefficients in the Tafel equation. Since the direction of the current is determined from settings within the **Tafel Slope (log10)** node, the β value that you set for the effective transfer coefficient is expected to be positive. **Beta** Appears when electrons are registered as reactants or products. Allows you to specify a term β for the anodic or cathodic apparent transfer coefficients in the Tafel equation.
Transport Limited Tafel Slope (log 10) Creates a **Properties** > **Specific Exchange Current Y-Intercept (log10)** sibling node.	**Transport Limited Tafel Slope (log10)** Provides two sub-nodes: • **Beta (log10)** allows you to specify a combined term β as a scalar profile for the combined anodic or cathodic apparent transfer coefficients in the Tafel equation. You can also approximate effects that are not initially accounted for in the simulation setup. Since the direction of the current is determined from settings within the Electrochemical Species model, this value that you set for the effective transfer coefficient is expected to be positive. • **Limiting Current** allows you to approximate effects that are not initially accounted for in the simulation setup by limiting the specific reaction current. For example, in the case of buried structures [154], you can approximate effects such as oxygen starvation. You can specify the limiting current, j_{lim}, using a scalar profile. The limiting current is assumed to be a positive quantity, regardless of if the reaction is anodic or cathodic. This option allows you to specify a combined term β for the combined anodic or cathodic apparent transfer coefficients in the Tafel equation.

Method

Available when the *Type* property is set to Sorption under the **Reactions** > **Reaction [n]** node.

All sorption methods are intended for reactions which have only one reactant species and one product species (electrons are considered a species).

Method	Corresponding Child Node
Henry Only available when using the Multi-Component Gas model.	**Henry** Uses the Henry method to simulate the sorption process, see Henry. Allows you to set a scalar profile for Henry's constant H in Eqn. (3682).

	Henry's law was developed for the sorption of a single species between a gas and a liquid phase—the reactant must be a gas.
Springer	**Springer** Uses the Springer method [162] to simulate the sorption process, see Springer. Provides the following subnodes which allow you to specify, as a scalar profile, the coefficients that are used to calculate the dimensionless water content at equilibrium: • **Equilibrium Water Content Coefficient, A** in Eqn. (3680). • **Equilibrium Water Content Coefficient, B** in Eqn. (3680). • **Equilibrium Water Content Coefficient, C** in Eqn. (3680). • **Equilibrium Water Content Coefficient, D** in Eqn. (3680).
Wu-Li-Berg	**Wu-Li-Berg** Uses the Wu-Li-Berg method [164] to simulate the sorption process, see Wu Li Berg.

Sorption Rate Constant

Appears when the *Type* property is set to Sorption under the **Reactions** > **Reaction [n]** node.

Method	Corresponding Child Node
User Specific Rate Constant	**User Specific Rate Constant** Allows you to set a scalar profile for the User Specific Rate Constant, $r_{surf,0}$ in Eqn. (3675).

Specific Exchange Current

This node is available when the Reaction Formulation method is set to **Butler-Volmer** or **Tafel**.

Properties

Select Method

Select Method	Corresponding Child Node
User Specific Exchange Current	**User Specific Exchange Current** Allows you to set the specific exchange current, j_0 in Eqn. (3657) or Eqn. (3660), as a scalar profile. The specific exchange current does not include the effects of species concentrations when computing the electric current that

the electrochemical reaction generates. Since the direction of the current is determined from the reaction set-up, this value is expected to be positive.

Specific Exchange Current Y-Intercept (log10)

This node is available when the Reaction Formulation method is set to **Tafel Slope (log 10)** or **Transport Limited Tafel Slope (log 10)**.

Properties

Select Method

Select Method	Corresponding Child Node
User Specific Exchange Current Y-Intercept (log10)	**User Specific Exchange Current Y-Intercept (log10)** Allows you to specify the y-intercept of the common logarithm of the specific exchange current, $\log_{10}(j_0)$ in Eqn. (3663), as a scalar profile.

Solid Mixture

The Solid Mixture node only appears when the Electrochemical Species model is used.

Right-Click Actions

Dependencies

Child Nodes

Solid Species

Right-Click Actions

Select Mixture Components

Opens the *Select Mixture Components* dialog, which allows you to select components that are available in the material database. The chosen components are then available to select as reactants or products for electrochemical reactions. See Material Databases.

Reorder Mixture Components

Materials appear under the Solid Species node in the order that you select them in the material database. This option opens a dialog which allows you to drag-and-drop materials to rearrange the order in which the materials appear in the simulation tree.

Import Species

Displays the *Select file and material database* dialog. The dialog allows you to select a file that contains the required species, and specify the Material Database to which the species are added.

Export Mixture Components to a file

Allows you to export and save the solid species mixture components that are specified.

Child Nodes

[Solid Species Component]

Properties

Database Material

Read only. Describes the material that is chosen for this solid species from the materials database, and *Id* which displays a unique ID number for that surface mechanism.

Right-Click Actions

Replace With

Allows you to select a different electrochemical species from the material database using the *Replace Mixture Component* dialog.

Child Nodes

Material Properties

Elemental Composition

Method	Corresponding Nodes
Elemental Composition Method	**Elemental Composition Method** Provides the **Atoms** sub-node which displays the *Molecular Weight* of the atoms within the **[Solid Species Component]**. You can right-click the the **Atoms** node and select **New Atom**. Opens the *Add atom to elemental composition* dialog in which you can select a type of atom and the quantity of that type of atom that is in the **[Solid Species Component]**. An **[Atom]** sub-node is created for each type of atom that is selected. The**[Atom]** node allows you to specify a *Value* for the number of that type of atom in the formula of the **[Solid Species Component]**.

Electrochemical Reaction Heating Model Reference

The Electrochemical Reaction Heating model accounts for heat contributions due to electrochemical reactions at fluid/solid interfaces (including those made with solid shells).

The heat contributions for many electrochemical processes, such as corrosion, are small. However, for applications with high electrochemical currents at electrochemical interfaces, such as in the case of electroplating, batteries, or in solid oxide fuel cells, the heat sources can reach significant levels.

The Electrochemical Reaction Heating model accounts for heat contributions from electrochemical reactions at interfaces (see Eqn. (1143)). Additionally, you can use the Ohmic Heating model to model heat release due to ion transport in bulk phases and due to resistances at reacting surfaces.

Electrochemical Reaction Heating Model Reference

Theory	See Electrochemical Reaction Heating Formulation.	
Provided By	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Electrochemical Reaction Heating**	
Requires	*Material*: one of **Gas**, **Liquid**, **Multi-Component Gas**, **Multi-Component Liquid**, **Solid** *Space*: **any** *Optional Models*: **Electrochemistry** *Electrochemistry*: **Electrochemical Reactions** *Flow*: any *Optional Models*: one of **Segregated Fluid Enthalpy**, **Segregated Fluid Isothermal**, **Segregated Fluid Temperature**, **Segregated Solid Energy**, **Coupled Energy**	
Activates	Field Functions	Specific Electrochemical Reaction Heat Surface Source. See Field Functions.

Field Functions

Specific Electrochemical Reaction Heat Surface Source

Displays the heat contributions $\dot{q}_A$ that are released by electrochemical reactions at interfaces according to Eqn. (1143).

Solid Ion Model Reference

You use the Solid Ion model for modelling (or transporting) electrochemical species (such as charged ions) in a solid continuum.

The electrochemical species can participate in an electrochemical reaction with a neighbouring fluid domain. For these cases, the Solid Ion model must be used in the solid continuum as a companion model to the Electrochemical Reactions model which is registered in the fluid domain.

Although it is possible to use the Solid Ion model in a shell region as a companion for the electrochemical reactions, the Solid Ion model does not transport / resolve profiles of molar concentration within the shell. This

behavior is as expected for many cases in which thin shells are not able to develop pronounced differences in molar concentration. However, if gradients of molar concentration are expected, use a non-shell region.

Solid Ion Model Reference

Theory	• The Solid Ion model solves a transport equation for molar concentrations in the same way as the Electrochemical Species model (depending on the settings of the model properties). See Coupling Electrochemical Species and Electrodynamic Potential Models. • See Electrochemical Surface Reactions (Heterogeneous).	
Provided By	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Solid Ion**	
Requires	*Space*: any *Material*: **Solid** *Optional Models*: **Electrochemistry** *Electrochemistry*: **Electrochemical Reactions**	
Properties	Key Properties are: *Transport*. See Solid Ion Model Properties.	
Activates	Model Controls (child nodes)	Electrochemical Species Components. See Electrochemical Species Components Reference
	Boundary Inputs	See Boundary Settings.
	Other	Provides the ability to register multi-component liquid or multi-component gas species (as well as electrochemical species) as part of the reaction mechanism for corresponding Reacting Surface models in liquid physics continua.

Solid Ion Model Properties

The following properties are available to specify when using the Solid Ion model—except in shell regions.

Convection

In transport equations, you can choose from a range of schemes that calculate the convection term at a cell face. This calculation requires STAR-CCM+ to compute the face value of a quantity from the surrounding cell values. The method used for computing this face value has a profound effect on the stability and accuracy of the numerical scheme. For guidance on selecting a convection scheme, see Convective Flux.

- **1st-order**: First-order convection scheme.
- **2nd-order**: Second-order convection scheme.

Secondary Gradients

Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces.

- **On**: Default value. Solves for interior and boundary types of secondary gradient.
- **Off**: Does not solve for either type of secondary gradient.
- **Interior Only**: Solves for the interior secondary gradients only.
- **Boundaries Only**: Solves for the boundary secondary gradients only.

Electroneutrality option

Specifies if the overall ionic charge density is strictly enforced as zero or not.

- **Not Enforced**: Not Enforced is the default option when using the Electrostatic Potential model or Solid Ion model. A uniform zero ionic charge density is not enforced. However, depending on the setup of other parameters, the overall ionic charge density may already equal zero. This option is appropriate for non-corrosion related applications which do not require electroneutrality.
- **Scaling**: Scaling is the default option when using the Electrodynamic Potential Model. This method ensures that the overall ionic charge density is equal to zero by scaling the ionic charge densities for each species proportionally. When the species have equal charge numbers, the total concentration is not changed by scaling. STAR-CCM+ checks the concentration profiles that are set using the concentration dimension property and provides a warning if they are not electrically neutral. This option is appropriate for applications that feature electroneutrality, such as corrosion and etching. See Eqn. (1109).

Transport

Defines how the Electrochemical Species model / Solid Ion model is coupled to other models and so defines the transport equations that STAR-CCM+ solves.

- **Species <> Electric Potential**: Physical two-way coupling of flux contributions between the Electrochemical Species model / Solid Ion model and the Electric Potential model.
- **Electric Potential > Species**: One-way coupling including drift flux / migrative contributions from the Electric Potential model to the Electrochemical Species model / Solid Ion model. The Electrochemical Reaction model does not enforce charge conservation.
- **Species > Electric Potential**: One-way transfer of contributions from the Electrochemical Species model / Solid Ion model to the Electric Potential model.
  - When using the Electrostatic Potential model, the contributions are for the space charge density.
  - When using the Electrodynamic Potential model, the contributions are for the diffusive current.
- **No Electric Potential Coupling**: The Electrochemical Species model / Solid Ion model and the Electric Potential model act independently. The Electrochemical Reaction model does not enforce charge conservation.

Note: If a charge is specified for one of the Solid Ion species, then to prevent a "`Charge does not balance`" error, set this property to either **Electric Potential > Species** or **Species > Electric Potential**.

Boundary Settings

The following boundary settings are available to specify when using the Solid Ion model—except in shell regions.

Wall Boundary

Wall Electrochemical Species Option

The wall electrochemical species option for a boundary allows you to define what happens at that boundary regarding electrochemical species.

Method	Corresponding Physics Value Nodes
Impermeable Specifies zero flux at the boundary.	None
Specified Flux You can specify the *Value* as a **Constant** or **Composite** array. The physics condition, **Wall Electrochemical Species Flux Derivative Option** also becomes available to specify.	One of the following nodes appears depending on what you set for *Concentration Dimension* on the Electrochemical Species model. You can specify the *Value* as a **Constant** or **Composite** array. **Molar Concentration Flux** **Number Density Flux**
Specified Value	One of the following nodes appears depending on what you set for *Concentration Dimension* on the Electrochemical Species model. You can specify the *Value* as a **Constant** or **Composite** array. **Molar Concentration** **Number Density**

Wall Electrochemical Species Flux Derivative Option

The Wall Electrochemical Species Flux Derivative Option allows you to specify the derivative of the Molar Concentration Flux of species i (or Number Density Flux) with respect to the molar concentration of species i (or number density) at the wall. The Concentration Flux Derivative represents:

$$\frac{d\mathrm{N}_{n,s,\,i}}{dc_i}$$

and the Number Density Flux Derivative represents:

$$\frac{d\mathrm{N}_{n,s,\,i}}{dn_p}$$

where $\mathrm{N}_{n,\,s,\,i}$ is the molar concentration flux of electrochemical species, i (or number density) at the wall, c_i is the molar concentration of electrochemical species i at the wall, and n_p is the number density at the wall.

This derivative is used for linearization.

This option is available when the **Wall Electrochemical Species Option** boundary condition is set to **Specified Flux**.

Method	Corresponding Physics Value Nodes
None The derivative of the Concentration Flux is not calculated. This is the default option.	None
Specified Allows you to specify the concentration flux derivative that is used for linearization. See Concentration Flux Derivative Formulation.	**Concentration Flux Derivative** Allows you to specify the concentration flux derivative as a **Constant** or **Composite** array. For a stable set-up, make sure that the flux derivative has a negative value.

Bulk Ion Chemical Reactions

STAR-CCM+ provides the Bulk Ion Chemical Reactions model which allows you to model bulk equilibrium reactions that occur within the bulk volume of the electrolyte.

Bulk Ion Chemical Reactions

This model is useful when modeling applications in which the chemical reactions occur in a short timescale, such as electroless deposition and wet etching. Due to the short timescale of such reactions, the electrochemical species are assumed to reach equilibrium instantaneously.

Contents:

Modeling Bulk Ion Chemical Reactions
Bulk Ion Chemical Reactions Reference

Modeling Bulk Ion Chemical Reactions

STAR-CCM+ allows you to model chemical reactions that involve electrochemical species within the bulk fluid volume of the electrolyte. Currently, the ability to model equilibrium reactions is provided.

Follow this procedure to model bulk chemical reactions within a fluid volume.

Note: A double precision version of STAR-CCM+ is required to model the transport of electrochemical species when coupling the electrochemical species model with an electric potential model.

1. Set up the physics continua.
   a) For the physics continuum that represents the electrolyte, select the following models:

      Note: Make sure that **Auto-Select recommended models** is activated.

Group Box	Model
Space	Select one
Time	Select one
Material	Either: • **Liquid** • **Multi-Component Liquid**
Optional Models	**Electrochemistry**
Electrochemistry	**Electrochemical Species**
Reaction Regime (available when Multi-Component Liquid is selected)	**Reacting**

Group Box	Model
Reactive Flow (when Multi-Component Liquid is selected) or *Optional Models* (if Liquid is selected)	**Bulk Ion Chemical Reactions**
Enabled Models	**Electromagnetism** (selected automatically) **Electrodynamic Potential** (selected automatically) **Gradients** (selected automatically)
Flow	Either: • **Segregated Flow** • **Coupled Flow**
Optional Models	**Segregated Species**
Equation of State	Select one (**Constant Density** is the most appropriate choice for liquids with very small temperature changes)
Viscous Regime	Select one

b) To include modeling electrochemical surface reactions, select the **Electrochemical Reactions** model from the *Optional Models* group box. For more information on modeling electrochemical surface reactions, see Modeling Electrochemical Surface Reactions.

c) Select any extra *Optional Models* that you require for the physics continuum.

- If you want to set a constant temperature without solving an additional transport equation, select the **Segregated Fluid Isothermal** model.
- To account for temperature changes due to electrochemical reactions, select the **Electrochemical Reaction Heating** model.
- If no temperature model is selected and the Electrochemical Species model is selected, the Electrochemical Species model runs with a constant temperature of 293.15K.

d) Click **Close**.

2. Set any model Properties that are required. In particular, select the **Models** > **Bulk Ion Chemical Reactions** node and specify the necessary properties. See Bulk Ion Chemical Reactions Model Properties.

   When using the **Bulk Ion Chemical Reactions** model, the only option that is available for the **Electrochemical Species** model property, *Electrochemical Species Solver Option*, is **Coupled**. The Segregated Electrochemical Species Solver option is not available with the Bulk Ion Chemical Reactions model. The Coupled setting ensures the best results for solving the electrochemical species concentrations in the bulk volume.

3. Define the material species.
   See either:
   - Managing Single-Component Materials
   - Managing Multi-Component Materials

If a rate exponent is specified for any mixture components that are in the Bulk Ion Chemical Reactions, the rate exponent is not used to compute the equilibrium reactions. It is the stoichiometric coefficients, v_i in Eqn. (1144), that are used to compute the equilibrium reaction source terms. If a non-zero value is entered for the rate exponent, upon initialization, the rate exponent is zeroed and a message is displayed in STAR-CCM+ saying that the rate exponents are not used.

Specify the chemical reaction species components.

4. Right-click the **Electrochemical Species** > **Electrochemical Species Components** node and select **Select Mixture Components**.
5. Select all of the electrochemical species that represent the reactants and products in the bulk chemical reactions. If you cannot find a complex electrochemical species component, you can either:
   - Close the dialog, add electrochemical species to the database, then reopen the *Select Mixture Components* dialog, and select the species. See Modifying a Copy of the Material Database.
   - Select a similar electrochemical species component and modify its material properties to change the charge, and the type and quantity of atoms.
6. Click **Apply**, then **Close**.
7. To modify the composition or properties of an electrochemical species component, expand the **Electrochemical Species Components** > **[Electrochemical Species Component]** node and edit the Material Properties as required. See [Electrochemical Species—Material Properties and Methods].

Define the reactions that occur in the bulk phase. See Defining Chemical Reactions.

8. Right-click the **[Continuum]** > **Models** > **Reacting** > **Reactions** node and select **New Reaction**.
   a) Define the Reactants and Products, making sure that the charges and proportions of elements balance.
   You can select electrochemical species components and liquid components as reactants and products in each reaction. However, although the concentrations of the liquid components are used to compute the equilibrium constants, it is assumed that these values are constant, and no source term is added to the liquid components.
   Since the equilibrium reactions are solved as a coupled set of equations, each reaction must have linearly independent sets of stoichiometric coefficients. For example, the following reactions are independent from each other:

   $$\mathrm{A} + \mathrm{B} \rightarrow \mathrm{C}$$

   $$\mathrm{B} + \mathrm{C} \rightarrow \mathrm{D}$$

   However, if the following equilibrium reaction is also added:

   $$\mathrm{A} + 2\mathrm{B} \rightarrow \mathrm{D}$$

   this equilibrium reaction is a sum of the two previous reactions and is therefore linearly dependent on the previous two reactions. In STAR-CCM+, if equilibrium reactions have linearly dependent sets of stoichiometric coefficients, an input error occurs upon initialization.
   b) Expand the **Reactions** > **[Reaction]** > **Properties** > **Equilibrium Constant** node.
   c) Select the **Equilibrium Constant** > **Keq Equilibrium Constant** node and specify the equilibrium constant.
   The dimension of the Keq Equilibrium Constant Eqn. (1145) changes depending on the number of reactants and products. Therefore, the *Equilibrium Constant Unit System* property allows you to specify the Keq Equilibrium Constant value using units of either (kmol, m, s, K) or (mol, cm, s, K).

9. Specify the material properties.
   a) Expand the **Models** > **[Material]** > **Material Properties** node.
   b) Select the **Electrical Conductivity** node and set *Method* to **Electrochemical Species**.
   c) Set any other material properties that are necessary.
10. Define the Initial Conditions.
    Specifically, set the Molar Concentration initial conditions. For electrochemical species components which have concentrations of zero—but are involved in equilibrium reactions, set a small molar concentration, such as `1E-20 kmol/m³`.
11. Define electrochemical species concentrations at boundaries and interfaces, as the simulation requires. For wall boundaries, if you are setting concentrations directly, set *Wall Electrochemical Species Option* to **Specified Value**.

    Bulk reactions require equilibrium conditions on boundaries. As a non-equilibrium concentration profile adversely affects the solution, STAR-CCM+ automatically computes equilibrium profiles based on the input concentrations that you specify. You are not required to pre-compute equilibrium concentrations yourself; in each iteration, STAR-CCM+ applies its own computed concentrations to the boundary. These computed concentrations are guaranteed to conserve the charge and elemental composition specified in the original profile.
    - Avoid zero molar concentrations for any species that participates in electrochemical reactions. Instead, set a very small value (for example, 1E-20 kmol / m^3) in lieu of a zero molar concentration.
12. Define the region types.

    You can specify a fluid or porous region for the electrolyte continuum. However, when using the Electrochemical Species model, porous baffle interfaces are not allowed.
13. Specify any necessary Solver settings.
14. Set the Stopping Criteria.
15. Set up Scenes and Plots to visualize the solution.
    For example, you can visualize the following field functions, among others, in a scalar scene or plot them on an x-y plot:
    - bulk ion production rate
    - electrochemical species residence time
    - charged species mobility of specific electrochemical species
    - electric current density
    - electric potential
    - migration flux of specific electrochemical species
    - molar concentration of specific electrochemical species
    - number density of specific electrochemical species
    - molecular diffusivity of specific electrochemical species
16. Run the simulation.

Bulk Ion Chemical Reactions Reference

This section provides detailed information about the electrochemistry models that are available for modeling bulk ion chemical reactions.

Contents:

Bulk Ion Chemical Reactions Model Reference

Bulk Ion Chemical Reactions Model Reference

The Bulk Ion Chemical Reactions model allows you to model chemical reactions that involve electrochemical species which occur within a fluid domain.

The Bulk Ion Chemical Reactions model is an optional electrochemistry model which you use with the Electrochemical Reactions model and the Electrochemical Species model. You can use the Bulk Ion Chemical Reactions model to simulate chemical reactions that occur within the bulk volume of an electrolyte. This model is useful when modeling applications, such as electroless deposition and wet etching, in which the chemical reactions occur in a short timescale. Due to the short timescale of such reactions, the electrochemical species are assumed to reach equilibrium instantaneously.

Bulk Ion Chemical Reactions Model Reference

Theory	See Bulk Ion Chemical Reactions Formulation.	
Provided By	**[physics continuum]** > **Models** > *Reactive Flow*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Bulk Ion Chemical Reactions**	
Requires	*Material*: either, **Liquid**, or **Multi-Component Liquid** *Reaction Regime*: **Reacting** *Space*: any *Optional Models*: **Electrochemistry** *Electrochemistry*: **Electrochemical Species** After selecting **Bulk Ion Chemical Reactions**, also select **Segregated Species** and **Electrochemical Reactions**.	
Properties	Key properties are: [*Enabled Trigger* and *Begin*]. See Bulk Ion Chemical Reactions Properties.	
Activates	Field Functions	Electrochemical Species Residence Time. See Field Functions.

Bulk Ion Chemical Reactions Properties

Enabled Trigger

Controls the sources according to the following options:

- **Never**: Leaves sources deactivated. The *Begin* property is inactive if this option is selected.
- **Iteration**: Activates sources. This option is available in steady-state cases.
- **Timestep**: Activates sources. This option is available in unsteady-state cases.

Begin

Specifies the iteration or time-step after which the reactions are activated. During previous iterations or time-steps, reactions are deactivated. This feature is useful when modeling significant flow as you can deactivate reactions until a flow solution is found.

Field Functions

Electrochemical Species Residence Time

Displays the residence time of electrochemical species within a specified volume.

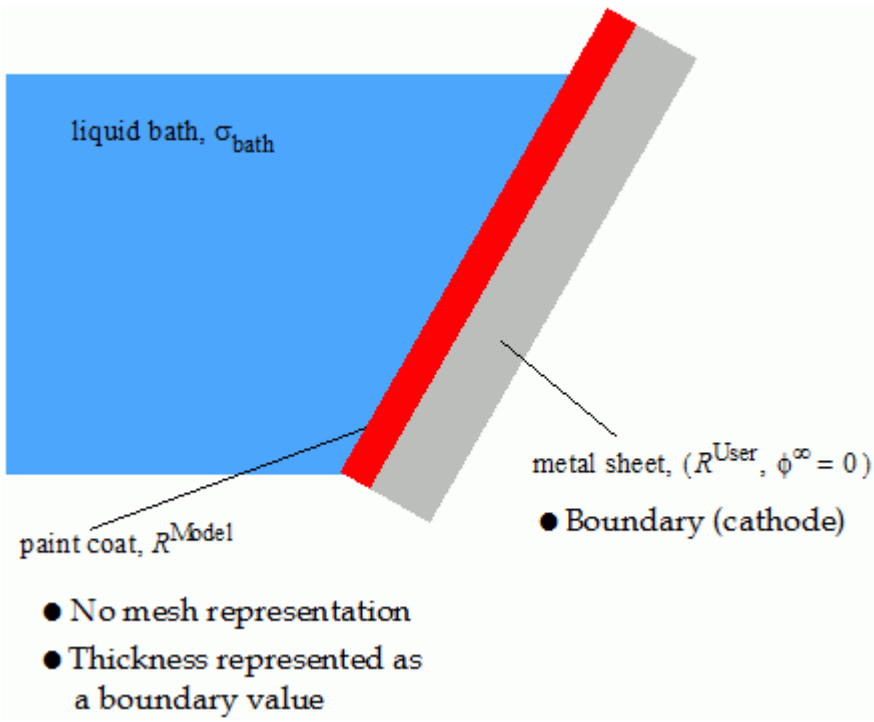
Bulk Ion Production Rate of [electrochemical species]

Displays the total amount of the [electrochemical species] (kmol) that is produced per unit volume by the bulk ion chemical reactions model.

Electro-Deposition Coating

You can use the Electro-Deposition Coating model in STAR-CCM+ to simulate e-coating and determine the paint layer thickness on a specified surface that is covered in a paint liquid.

Calibration is required for a specific paint liquid along with given process parameters. The model can be combined with the VOF Multiphase model to cover gas bubbles resulting from dipping a body into a paint bath. The following figure illustrates the notation that is used in the model formulation.



The formulation makes the following assumptions:

- STAR-CCM+ models the paint layer phase by a function that is available at a coated surface (boundary), rather than represented by a volume mesh.
- The formulation assumes a quasi-steady state for the electric potential. The steady equation is solved together with possible time-dependent boundary conditions.
- The model is calibrated against actual process parameters.
- The electric potential field and the layer growth do not affect the flow field.

Contents:

Electro-Deposition Coating Model Reference

Electro-Deposition Coating Model Reference

Electro-Deposition Coating Model

Theory	See Electro-Deposition Coating Model Formulation.	
Provided By	**[physics continuum]** > **Models** > *Replace with the Group Box name that contains the model choice*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Electro-Deposition Coating**	
Requires	*Material*: **Liquid** *Flow*: **Coupled Flow** or **Segregated Flow** *Space*: **Three Dimensional** or **Two Dimensional** *Optional Models*: **Electromagnetism** *Electromagnetism*: **Electrodynamic Potential** *Equation of State*: any *Time*: **Implicit Unsteady**	
Properties	Key properties are: **Starting Criterion** and **Paint Layer Total Resistance Specification**. See Electro-Deposition Coating Model Properties.	
Activates	Model Controls (child nodes)	• **Electrical Resistance Offset** • **Minimum Accumulated Specific Surface Charge** • **Minimum Specific Current Squared Time Integral** • **Minimum Specific Electric Current** • **Paint Layer** See Electro-Deposition Coating Model: Model Controls.
	Initial Conditions	**Paint Electrical Resistance** and **Paint Layer Thickness**. See Electro-Deposition Coating Model Initial Conditions.
	Boundary Inputs	See Electro-Deposition Coating Model Boundary Settings.
	Solvers	**Electro-Deposition Coating**. See Electro-Deposition Coating Model Solver Properties.
	Field Functions	• **Accumulated Specific Surface Charge** • **Boundary Specific Electric Current** • **Paint Coulomb Efficiency** • **Paint Density** • **Paint Electrical Resistance Area** • **Paint Electrical Resistivity** • **Paint Layer Thickness** • **Paint Layer Thickness Growth Rate** • **Time Integrated Specific Current Density Squared** See Electro-Deposition Coating Model Field Functions.

Electro-Deposition Coating Model Properties

Starting Criterion

Allows you to specify a threshold value after which the paint layer starts to grow.

- **None**
  The paint layer starts growing immediately upon running the simulation.
- **Accumulated Specific Surface Charge**
  The paint layer starts growing after the value that is set for the time integral of the specific electric current is reached. You can specify a threshold value as a scalar profile for the continuum under the **Electro-Deposition Coating** > **Minimum Accumulated Specific Surface Charge** node. If necessary, you can set specific values for each boundary.
- **Time Integral Specific Electric Current Squared**
  The paint layer starts growing after the value that is set for the time integral of the square of the specific electric current is reached. You can specify a threshold value as a scalar profile for the continuum under the **Electro-Deposition Coating** > **Minimum Specific Current Squared Time Integral** node. If necessary, you can set specific values for each boundary.

Paint Layer Total Resistance Specification

Allows you to select the method by which the resistivity of the paint layer is calculated before the ODE (Ordinary Differential Equation) is solved for the paint layer thickness. You can set the resulting profiles under the **Electro-Deposition Coating Model** > **Paint Layer** node.

- **Differential Equation**
  STAR-CCM+ solves a differential equation Eqn. (1147) to calculate the paint layer resistance using the electrical resistivity that is specified for the paint layer.
- **Resistivity**
  Allows you to set the total electrical resistivity of the paint layer from which STAR-CCM+ calculates the paint layer resistance by multiplying the total electrical resistivity by the paint layer thickness Eqn. (1148).
- **Resistance**
  Allows you to specify an expression for the electrical resistance of the paint layer Eqn. (1149).

Electro-Deposition Coating Model Initial Conditions

Paint Electrical Resistance

Allows you to specify the initial electrical resistance of the painted area as a scalar profile.

Paint Layer Thickness

Allows you to specify the initial thickness of the paint layer as a scalar profile.

Electro-Deposition Coating Model Boundary Settings

Wall Boundary

Electro-Deposition Coating Specification

Specifies the electro-deposition coating method that is used.

Method	Corresponding Physics Value Nodes
None Leaves the boundary uncoated (default).	None
Current Density Applies the coating. The coating uses the continuum values or the custom boundary values, depending on the setting of the *Option* property.	None

Option	Corresponding Physics Value Nodes
Use Continuum Values Uses values that are specified for the continuum.	None
Specify Custom Values Uses **Physics Values** that are set for the boundary—which are available to specify when *Method* for Electro-Deposition Coating Specification is set to **Current Density**.	You can set the following values as scalar profiles. **Electrical Resistance Offset** g in Eqn. (1147). Available when the Electro-Deposition Coating property, Paint Layer Total Resistance Specification, is set to Differential Equation. **Initial Paint Electrical Resistance** r_{paint} in Eqn. (1147) **Initial Paint Layer Thickness** h in Eqn. (1147) **Minimum Accumulated Specific Surface Charge** The paint layer only starts to grow when the time integral of the specific electric current reaches this threshold. Available when the Electro-Deposition Coating property, Starting Criterion, is set to Accumulated Specific Surface Charge. **Minimum Specific Current Squared Time Integral** The paint layer only starts to grow when the time integral of the square of the boundary-specific electric current reaches this threshold. Available when the Electro-Deposition Coating property,

Starting Criterion, is set to Time Integral Specific Electric Current Squared.

Minimum Specific Electric Current

$j_{\min}$ in Eqn. (1150)

Electro-Deposition Coating Model Solver Properties

Under-Relaxation Factor

In order to promote convergence, this property is used to under-relax changes of the solution during the iterative process. If residuals show solution divergence and do not decrease, reduce the under-relaxation factor for the relevant solvers.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Electro-Deposition Coating Model Field Functions

Accumulated Specific Surface Charge

Displays on a surface, the time integral of the specific electric current which is used in the starting criterion duriing the current time step, when the Electro-Deposition Coating property, *Starting Criterion*, is set to **Accumulated Specific Surface Charge**.

Boundary Specific Electric Current

See Eqn. (1150).

Paint Coulomb Efficiency

See Eqn. (1150).

Paint Density

See Eqn. (1150).

Paint Electrical Resistance Area

See Eqn. (1147).

Paint Electrical Resistivity

See Eqn. (1147).

Paint Layer Thickness

See Eqn. (1150).

Paint Layer Thickness Growth Rate

See Eqn. (1150).

Time Integrated Specific Current Density Squared

Displays on the surface, the time integral of the square of the specific electric current which is used in the starting criterion during the current time step, when the Electro-Deposition Coating property, *Starting Criterion*, is set to **Minimum Specific Current Squared Time Integral**.

Electro-Deposition Coating Model: Model Controls

Any profiles that are specified for the following child nodes are set for the entire continuum, unless custom values are set for individual boundaries.

- The **Paint Layer** node allows you to set material properties specifically for the paint layer, independent of the material properties that you set for the Liquid. Constant values are set for the Coulomb Efficiency, c_{paint}, and Density, ρ_{paint}—both in Eqn. (1150) and Eqn. (1151).

  A further material property is available to set for the resistivity of the paint layer, r_{paint}, in Eqn. (1147). The particular node that appears depends on the setting that is specified for the Paint Layer Resistance Specification Option:

  - When Differential Equation is selected, you can set a profile for the **Electrical Resistivity**, Eqn. (1147).
  - When Resistivity is selected, you can set a profile for the **Total Electrical Resistivity** , Eqn. (1148).
  - When Resistance is selected, you can set a profile for the **Electrical Resistance**, Eqn. (1149).
- The **Minimum Accumulated Specific Surface Charge** node allows you to specify—for the continuum—a threshold for the time integral of the specific electric current. Above this threshold, the paint layer begins to grow. Available when the Electro-Deposition Coating property, Starting Criterion, is set to Accumulated Specific Surface Charge.
- The **Minimum Specific Current Squared Time Integral** node allows you to specify—for the continuum—a threshold for the time integral of the square of the boundary-specific electric current. Above this threshold, the paint layer begins to grow. Available when the Electro-Deposition Coating property, Starting Criterion, is set to Time Integral Specific Electric Current Squared.
- The **Electrical Resistance Offset** node allows you to specify a scalar profile for g in Eqn. (1147). Available when the Electro-Deposition Coating property, Paint Layer Total Resistance Specification, is set to Differential Equation.
- The **Minimum Specific Electric Current** node allows you to specify a scalar profile for j_{min} in Eqn. (1150).

Electrochemistry Formulation

This section provides formulation for modeling Li-ion batteries, electrochemical surface reactions, and ionic wind.

Contents:

Li-Ion Electric Potential Model Formulation
Li-Ion Concentration Model Formulation
Reporting Li-Ion Cell Parameters
Solid Electrolyte Interface Formulation
Electrochemical Species Model Formulation
Electrochemical Surface Reactions Formulation
Bulk Ion Chemical Reactions Formulation
Electro-Deposition Coating Formulation

Li-Ion Electric Potential Model Formulation

In the solid electrodes and current collectors, the electric current density is defined using Ohm's law .

$$\mathbf{J} = \sigma \mathbf{E} \tag{1074}$$

where $\mathbf{E} = -\nabla\phi$.

Within the electrolyte phase, the electric current density $\mathbf{J}$ is due to diffusion and migration of ions (salt). Using concentrated solution theory, $\mathbf{J}$ can be written as

$$\mathbf{J} = \kappa\mathbf{E} - \frac{\kappa}{F}\left(\frac{s_+}{n\nu_+} + \frac{{t_+}^0}{z_+\nu_+} - \frac{s_0 c}{n c_0}\right)\nabla\mu_e \tag{1075}$$

where the first term of the right-hand side corresponds to migration, and the second term to diffusion. The effective electrical conductivity κ in the liquid phase accounts for an eventual non-unity porosity χ and tortuosity τ within the separator.

$$\kappa = \frac{\chi}{\tau}\kappa_0 \tag{1076}$$

Assuming a binary lithium based electrolyte to parametrize Eqn. (1075):

$$\mathbf{J} = \kappa\mathbf{E} + \frac{2R_u T\kappa}{F}\left(1 - {t_+}^0\right)\left[1 + \frac{\mathrm{d}\left(\ln f_\pm\right)}{\mathrm{d}\left(\ln c\right)}\right]\nabla\left(\ln c\right) \tag{1077}$$

or

$$\mathbf{J} = \kappa\mathbf{E} + \kappa_\mathrm{d}\nabla\left(\ln c\right) \tag{1078}$$

where $\kappa_{\mathrm{d}} = \frac{2R_u T\kappa}{F}\left(1-t_+^{\ 0}\right)\left[1+\frac{\mathrm{d}\left(\ln f_{\pm}\right)}{\mathrm{d}\left(\ln c\right)}\right]$.

STAR-CCM+ uses this electrical current density definition for the electrolyte that is used in the Li-Ion Battery Cell Model.

Due to charge conservation, the electric current density field must be divergence free:

$$\nabla \cdot \mathbf{J} = 0 \tag{1079}$$

Applying Eqn. (1074) and Eqn. (1078) (the electric current density definition) to Eqn. (1079), integration over a control volume, and subsequent application of Gauss's theorem yields the transport equations for electric potential

$$\oint_S \sigma \nabla \phi \cdot \ \mathrm{d}\ \mathbf{s} = 0 \tag{1080}$$

within the (solid) electrode and current collector phases, and

$$\oint_S \kappa \nabla \phi \cdot \ \mathrm{d}\ \mathbf{s} = \oint_S \kappa_{\mathrm{d}} \nabla\left(\ln c\right) \cdot \ \mathrm{d}\ \mathbf{s} \tag{1081}$$

within the (liquid) electrolyte phase.

Li-Ion Concentration Model Formulation

A simple species diffusion transport equation is solved for lithium (Lithium/Salt Concentration) transport within the electrode phase:

$$\frac{\partial}{\partial t}\int_V c\ \mathrm{d}\ V = \oint_S D \nabla c \cdot \ \mathrm{d}\ \mathbf{s} \tag{1082}$$

where D is a constant defined in Eqn. (1084).

An additional transport term for lithium cation migration is considered in the (liquid) electrolyte phase, so that

$$\frac{\partial}{\partial t}\int_V \chi c\ \mathrm{d}\ V = \oint_S D \nabla c \cdot \ \mathrm{d}\ \mathbf{s} - \int_V \frac{\mathbf{J} \cdot \nabla t_+^0}{F}\ \mathrm{d}\ V \tag{1083}$$

The Effective Lithium/Salt Diffusivity Coefficient D accounts for non-unity values of porosity χ and tortuosity τ within the separator, and considers the "Concentrated Solution Correction to Salt Diffusion Coefficient", written as

$$D = D_0 \frac{\chi}{\tau}\left[1-\frac{d\left(\ln c_0\right)}{d\left(\ln c\right)}\right] \tag{1084}$$

Convective transport in electrolyte is not modeled.

Li-Ion Cell Report

The Li-Ion cell report can be run once an initialized setup for a Li-Ion Battery Cell Model is provided.

The Li-Ion cell report gives the following values:

- Open Circuit Voltage OCV
- State of Charge SOC
- Capacity CAP

Using the Li-Ion Cell Report State Table

The Li-Ion Cell Report State Table provides information for:

- the conditions to reinitialize the Lithium/Salt Concentration at different values of state of charge SOC, or
- estimating the initial conditions for the Lithium/Salt Concentration in both electrodes.

The State Table tabulates the open circuit voltage OCV, the stoichiometries, and average concentration values for both electrodes for different values of SOC.

Open Circuit Voltage

The Open Circuit (Cell) Voltage OCV is defined as the cell voltage that is approached asymptotically as the cell rests for an infinite period. Empirically, this quantity is measured as the voltage difference between the positive and negative current collectors after the cell is rested for a few hours.

For a cell in perfect equilibrium (that is, where the stoichiometry is constant), the OCV is computed as

$$OCV = U_{\mathrm{eq},+}(\overline{y}) - U_{\mathrm{eq},\,-}(\overline{x}) \tag{1085}$$

where $U_{\mathrm{eq},+}(\overline{y})$ and $U_{\mathrm{eq},\,-}(\overline{x})$ are the relative galvanic equilibrium potentials of the positive and negative electrode active materials. $\overline{y}$ and $\overline{x}$ denote their respective (average) stoichiometry (or utilization).

$$\overline{y} = \frac{\int_{+} c \,\mathrm{d}\, V}{c_{s,\,\mathrm{max},\,+} V_{+}} = \frac{\overline{c_{+}}}{c_{s,\,\mathrm{max},+}} \tag{1086}$$

$$\overline{x} = \frac{\int_{-} c \,\mathrm{d}\, V}{c_{s,\,\mathrm{max},-} V_{-}} = \frac{\overline{c_{-}}}{c_{s,\,\mathrm{max},-}} \tag{1087}$$

State of Charge

The State of Charge SOC is defined as 1 when the battery is fully charged, and 0 when the battery is fully discharged. You define the voltage values corresponding to $SOC = 0$ and $SOC = 1$ in the properties of a Li-ion cell report. The SOC corresponding to other voltage values is linearly interpolated from these two specified voltage values.

Note: Overcharge, as well as deep-discharge, leads to $SOC \notin [0, 1]$.

Capacity

The capacity of the battery cell depends on the maximum amount of lithium (ions) that can be transferred from one electrode to the other, in the range $OCV = \left[OCV_{SOC=0}, OCV_{SOC=1}\right]$. A lumped model is used within the Li-Ion Cell Report to compute the cell capacity:

$$CAP = n_{-}\Big|_{SOC=0}^{1} F = \left(n_{-}\big|_{SOC=1} - n_{-}\big|_{SOC=0}\right)F \tag{1088}$$

where n_{-} is the number of lithium moles that are stored in the negative electrode.

Solid Electrolyte Interface Formulation

Subsequent sections provide the following information:

- Electric Potential Interfacing Condition
- Concentration Interfacing Condition
- Butler-Volmer Relationship

Electric Potential Interfacing Condition

Conservation of the electric charge that is exchanged across the SEI gives the interface condition for the Boundary Specific Electric Current:

$$j_{n,s} = -j_{n,l} \tag{1089}$$

where $j_{n,s}$ is the Boundary Specific Electric Current on the electrode (solid)

$$j_{n,s} = \mathbf{J}_s \cdot \mathbf{n}_s \tag{1090}$$

and $J_{n,l}$ is the Boundary Specific Electric Current on the electrolyte (liquid)

$$j_{n,l} = \mathbf{J}_s \cdot \mathbf{n}_l \tag{1091}$$

The Boundary Specific Electric Current is the SEI normal component of the Electric Current Density.

The normal vector always points outward of the considered electrode or electrolyte domain:

$$\mathbf{n}_s = -\mathbf{n}_l \tag{1092}$$

Using the definition for the specific electric current (Eqn. (1089)), the potential boundary condition on the electrode side is

$$\left.\frac{\partial \phi}{\partial n}\right|_{\mathrm{SEI}} = -\frac{j_{n,s}}{\sigma} \tag{1093}$$

Using Eqn. (1093), the potential boundary condition on the electrolyte side is

$$\left.\frac{\partial \phi}{\partial n}\right|_{\mathrm{SEI}} = -\frac{j_{n,l}}{\kappa} + \frac{\kappa_d}{\kappa}\frac{\partial\left(\ln c_1\right)}{\partial n} \tag{1094}$$

Concentration Interfacing Condition

By conservation of the flux of Lithium $\mathbf{N}_+$ across the SEI,

$$\mathbf{N}_+ \cdot \mathbf{n}_s = -\mathbf{N}_+ \cdot \mathbf{n}_l \tag{1095}$$

On the electrolyte side, the flux of lithium ions is expressed as

$$\mathbf{N}_+ = \frac{\mathbf{J}}{F} = -D\ \Delta\ c \frac{1}{1-t_+{}^0} \tag{1096}$$

Taking the inner product of Eqn. (1096) with the SEI normal vector $\mathbf{n}_l$, the interfacing condition reads

$$\left.\frac{\partial c}{\partial n}\right|_{\mathrm{SEI}} = -\frac{j_{n,l}}{FD}\left(1-t_+{}^0\right) \tag{1097}$$

On the solid electrode side, the lithium non-ionic flux is expressed as

$$\mathbf{N}_+ = \frac{\mathbf{J}}{F} = -D\ \Delta\ c \tag{1098}$$

Taking the inner product of Eqn. (1098) with $\mathbf{n}_s$, the concentration interfacing condition becomes

$$\left.\frac{\partial c}{\partial n}\right|_{\mathrm{SEI}} = -\frac{j_{n,s}}{FD} \tag{1099}$$

Butler-Volmer Relationship

The Butler-Volmer relationship is used to compute the SEI normal component of the Electric Current Density $j_{n,s} = -j_{n,l}$, written as

$$j_{n,s} = j_0\left(e^{\frac{\alpha_a F\eta}{R_u T}} - e^{\frac{-\alpha_c F\eta}{R_u T}}\right) + C\frac{\partial}{\partial t}\left(\phi_s - \phi_l\right) \tag{1100}$$

$$j_0 = FK\left(\frac{c_s}{c_{s,\max}}\right)^{\alpha_1}\left(1-\frac{c_s}{c_{s,\max}}\right)^{\alpha_2}\left(\frac{c_l}{c_{l,\mathrm{ref}}}\right)^{\alpha_3} \tag{1101}$$

$c_{l,\mathrm{ref}}$ is a molar reference concentration of lithium in the electrolyte.

The SEI Surface Overpotential η is computed as

$$\eta = \phi_s - \phi_l - U_{\mathrm{eq}} - j_{n,s}R_{\mathrm{SEI}} \tag{1102}$$

The rate coefficient K depends on temperature according to:

$$K = K_0 e^{\frac{-E_a}{R_u T}} \tag{1103}$$

Similarly, the electrical resistance of the SEI is dependent on temperature:

$$R_{\mathrm{SEI}} = R_0 e^{\frac{-E_{\mathrm{SEI}}}{R_u T}} \tag{1104}$$

Electrochemical Species Model Formulation

The formulation of transport equations that are solved for electric potential depends on which model you couple with the Electrochemical Species model. However, you can also choose to use a segregated or coupled solver with the electrochemical species model to solve for the molar concentrations of species.

Coupling Electrochemical Species with Electric Potential

If necessary, you can define the coupling option using the Electrochemical Species model expert property, **Transport**. See Electrochemical Species Model Reference.

The options are:

- Coupling Electrochemical Species and Electrodynamic Potential Models
- Coupling Electrochemical Species and Electrostatic Potential Models

The following sections provide further extensions of Electrochemical Species model formulation:

- Coulomb Force Formulation
- Molar Concentration Flux Derivative Formulation

Contents:

Coupling Electrochemical Species and Electrodynamic Potential Models
Coupling Electrochemical Species and Electrostatic Potential Models
Ionic Wind Formulation
Coulomb Force Formulation
Molar Concentration Flux Derivative Formulation

Coupling Electrochemical Species and Electrodynamic Potential Models

The formulation in this section is solved when modeling applications such as corrosion, etching, and electrochemical deposition.

The Electrochemical Species model solves a transport equation for molar concentrations.

Continuous form:

$$\frac{\partial}{\partial t}\left(c_i \chi\right) = -\nabla \cdot \mathbf{N}_i \tag{1105}$$

$$\mathbf{N}_i = -D_i \chi^{\beta} \nabla c_i - z_i F u_i c_i \chi^{\beta} \nabla \phi + c_i \mathbf{v} \tag{1106}$$

Finite Volume or integral form:

$$\frac{\partial}{\partial t}\int_V \chi c_i \mathrm{d}V + \oint_A c_i \mathrm{v} \cdot \mathrm{da} = \oint_A D_i \chi^\beta \nabla c_i \cdot \mathrm{da} + \oint_A z_i F u_i c_i \chi^\beta \nabla \phi \cdot \mathrm{da} + \int_V S_{c_i} \mathrm{d}V \tag{1107}$$

χ^β represents the porosity that is used to calculate the electrochemical conductivity using the Bruggeman (fluid) approach, see Eqn. (3695). The Bruggeman (fluid) approach is used – and not the Bruggeman (solid) approach – because the electrochemical species model that is used for porous regions is present in fluid regions only.

Porosity is already accounted for in the convective term, $c_i\mathbf{v}$, since superficial velocity is solved for in porous regions—which is advection velocity that is multiplied by porosity.

When the Electrochemical Species model is active, the Electrodynamic Potential model solves a transport equation for the ionic space charge density, ρ_{ion}.

Continuous form:

$$\frac{\partial}{\partial t}\left(\rho_{ion}\chi\right) = -\nabla \cdot \mathbf{J} = 0 \tag{1108}$$

$$\mathbf{J} = -\kappa_i \chi^\beta \nabla \phi - F\sum_i z_i D_i \chi^\beta \nabla c_i + \rho_{ion}\mathbf{v} \tag{1109}$$

Assuming electroneutrality, $\rho_{ion} = 0$, the finite volume, or integral form reads:

$$0 = \int_A \kappa_i \chi^\beta \nabla \phi \cdot \mathrm{da} + F\sum_i z_i \int_A D_i \chi^\beta \nabla c_i \cdot \mathrm{da} + \int_V S_{\rho_{ion}} \mathrm{d}V \tag{1110}$$

Substituting expressions for κ_i, the ionic conductivity of species i:

$$\kappa_i = F^2 \sum_i z_i^2 u_i c_i \tag{1111}$$

the Nernst-Einstein relation:

$$u_i = \frac{D_i}{R_u T} \tag{1112}$$

and the relation between the ionic space charge density and molar concentrations,

$$\rho_{ion} = F\sum_i z_i c_i \tag{1113}$$

the following flux expressions are derived:

$$\mathbf{N}_i = -D_i \chi^\beta \nabla c_i - z_i F \frac{D_i}{R_u T} \chi^\beta c_i \nabla \phi + c_i \mathbf{v} \tag{1114}$$

$$\mathbf{J} = -F^2 \sum_i z_i^2 \frac{D_i}{R_u T} \chi^\beta c_i \nabla \phi - F\sum_i z_i D_i \chi^\beta \nabla c_i + F\sum_i z_i c_i \mathbf{v} \tag{1115}$$

Comparison of Eqn. (1114) and Eqn. (1115) produces a relation between the molar species fluxes and the electric current densities:

$$\mathbf{J} = F\sum_i z_i \mathbf{N}_i \tag{1116}$$

Due to the electroneutrality, $\rho_{ion} = 0$, that is assumed in corrosive systems, the convective term in Eqn. (1109) is zero.

Coupling Electrochemical Species and Electrostatic Potential Models

When coupling Electrochemical Species and Electrostatic Potential models, the following equation,

$$\nabla \cdot \mathbf{E} = \rho_{ion}/\varepsilon_0 \tag{1117}$$

replaces Eqn. (1108) to determine the electric potential.

The continuous form is written as:

$$\oint_A \varepsilon_0 \nabla\phi \cdot \mathrm{da} = \int_V \rho_{ion} \mathrm{d}V + \int_V S_{Estat} \mathrm{d}V \tag{1118}$$

Here, ρ_{ion} is non-zero and is computed from Eqn. (1113).

The transport equations for the electrochemical species are expressed in terms of a species number density:

$$\frac{\partial}{\partial t}\left(N_P \chi\right) = -\nabla \cdot \mathbf{N}_P \tag{1119}$$

$$\mathbf{N}_P = -n_P \mathrm{K}_P \chi^\beta \nabla\phi - D_P \chi^\beta \nabla n_P + n_P \mathbf{v} \tag{1120}$$

See Cagnoni and others (2013) [152].

The species number density, n_P, is equal to the molar concentration of the ionic species, c, multiplied by Avogadro's number. Therefore, number density represents the number of ionic species particles per volume.

Often, only a single ionic species with a unity charge number ($z = 1$) is considered.

For a single ionic species, i, the relation:

$$qn_P = \rho_{ion} = qN_A c = Fc \tag{1121}$$

(by definition of Faraday's constant) is used to rewrite Eqn. (1119) and Eqn. (1120) as a transport equation for charge density:

$$\frac{\partial}{\partial t}\left(\rho_{ion} \chi\right) = -\nabla \cdot \mathbf{J} \tag{1122}$$

$$\mathbf{J} = -\rho_{ion} \mathrm{K} \chi^\beta \nabla\phi - D \chi^\beta \nabla \rho_{ion} + \rho_{ion} \mathbf{v} \tag{1123}$$

as in Neimarlija and others (2009) [159].

You can also obtain Eqn. (1122) and Eqn. (1123) by multiplying Eqn. (1105) and Eqn. (1106) with Faraday's constant, F, and charge number, z_i, applying Eqn. (1113), and substituting ionic mobility as:

$$\mathrm{K}_i = F z_i u_i \tag{1124}$$

Although K_i and u_i are commonly termed ion mobility, their dimension and units differ by the dimension of Faraday's constant ($\left[\mathrm{m}^2/\mathrm{V\ s}\right]$ vs $\left[\mathrm{m}^2\mathrm{kmol}/\mathrm{J\ s}\right]$).

Ionic Wind Formulation

Since ionic wind is modeled using the Electrochemical Species model and the Electrostatic Potential model, see the Coupling Electrochemical Species and Electrostatic Potential Models formulation.

The following formulation describes the boundary conditions that are present in ionic wind simulations.

Collecting Electrode

Assuming an infinitely fast neutralization of ions at the collecting electrode boundary, zero space charge density, ρ, is applied together with zero electric potential, ϕ, as a boundary condition. The collecting electrode is assumed to be grounded.

Discharge Electrode

The visual critical electric field, E_{SC}, which is set as a boundary condition, can be computed by:

$$E_{SC} = A\delta + B\sqrt{\frac{\delta}{a}} \tag{1125}$$

where:

- E_{SC} is the critical electric field intensity at the start of the corona
- A and B are empirical constants for air and a negative corona which usually have values of $32.3 \times 10^5\ \mathrm{V/m}$ and $0.846 \times 10^5\ \mathrm{V/m}^{1/2}$ respectively
- δ is the relative density of air which is assumed to be $\delta = 1$
- a represents the local radius of the curvature of the electrode

Note: On symmetry plane boundaries, dependent variables are assumed to have a zero gradient in the direction normal to the boundary.

Dielectric Boundaries

At dielectric boundaries, such as plastic housings of ionic wind generators, where no neutralization of ions is assumed to occur, you can set a zero flux of ions by setting both the specific electric flux and the specified flux of the electrochemical species to zero.

Coulomb Force Formulation

The Coulomb force that is exerted on ionic species in an electric field leads to a volumetric force term which is written as:

$$\mathrm{f}_{Coulomb} = \rho_{ion}\mathrm{E} \tag{1126}$$

See Cagnoni et al., (2013) [152].

The ionic space charge density here is computed from Eqn. (1113) rather than considering a single ionic species as in Eqn. (1121).

Concentration Flux Derivative Formulation

You can define the concentration flux derivative by specifying the wall electrochemical species option *Method* on a boundary.

If you specify a flux derivative:

- molar flux derivative:

  $$\frac{\partial \mathrm{N}_{n,\,s,\,i}}{\partial c_i}$$

- number density flux derivative:

  $$\frac{\partial \mathrm{N}_{n,\,s,\,i}}{\partial n_p}$$

the derivative of the boundary flux with respect to the concentration at the cell centre can then be computed. This term can then be incorporated into the construction of the Jacobian matrix.

where:

$\mathrm{N}_{n,s,\,i}$ = molar flux of species i [kmol/m^2 s]

c_i = molar concentration of species i at the wall boundary [kmol/m^3]

n_p = number density of species i at the wall boundary [1/m^3]

Electrochemical Surface Reactions Formulation

Electrochemical surface reactions either produce electrons (anodic) or consume electrons (cathodic). Any number of these reactions can occur at any given time at the interface between a metal and a fluid.

A general form of electrochemical surface reactions is represented as:

$$j_{n,s} = f(\eta)\,;\ \begin{Bmatrix} j_{n,s} > 0 \\ \eta > 0 \end{Bmatrix} \equiv \text{anodic and} \begin{Bmatrix} j_{n,s} < 0 \\ \eta < 0 \end{Bmatrix} \equiv \text{cathodic} \tag{1127}$$

Reaction formulations provide functional forms which specify the dependence of the specific reaction current density, that is the electrochemical reaction rate, $j_{n,s}$, on the surface overpotential η.

In most electrochemical reaction setups, the electrons exist in the solid and the ions exist in the fluid. For such setups, the surface overpotential, η, is defined as:

$$\eta = \phi_s - \phi_l - U_{eq} - j_{n,s}R \tag{1128}$$

where ϕ_s represents the solid (electric) potential, ϕ_l represents the potential of the liquid solution relative to a reference electrode, $j_{n,s}$ is the boundary-specific electric current, and R is the resistance that is specified at the boundary or interface.

ϕ_l is obtained internally from the electrodynamic potential model, and ϕ_s is specified in the Physics Values at each boundary using a scalar profile.

The electric current is affected when additional electrical resistance is specified on a boundary or interface. The additional voltage drop is calculated according to:

$$\Delta\phi = j_{n,s}R \tag{1129}$$

where ϕ is the electric potential.

The driving force behind these reactions is the difference in Gibbs free energy between the reactants and the products, $\Delta G°$, which equates to the equilibrium potential, U_{eq}. For the electrochemical reaction:

$$\left(s_a A \leftrightarrow s_b B\right) + ne^- \tag{1130}$$

$$U_{corr}(volts) = \frac{\Delta G°}{nF} \tag{1131}$$

Where the Faraday constant, F=96,485,258.799 C/kmol and n is the stoichiometric coefficient of the electrons.

The potential difference, U, at the interface is due to the difference in potential of the metal and the electrolyte.

In STAR-CCM+, the potential difference is defined as:

$$U = \phi_s - \phi_l \tag{1132}$$

Where ϕ_s is the potential of the solid and ϕ_l is the potential of the liquid electrolyte.

The overpotential definition is equivalent to that of the overpotential which is defined for the electrodynamic potential model in Eqn. (3734). Where:

- ϕ_s is equivalent to ϕ_1

  This value represents the potential of the boundary on the solid region side of the interface which corresponds to the second boundary that you select when defining an interface.
- ϕ_l is equivalent to ϕ_0

  This value represents the potential of the boundary on the electrolyte region side of the interface which corresponds to the first boundary that you select when defining an interface.

In Eqn. (3734), ϕ_E represents the equilibrium potential U_{eq}.

If the potential difference at the interface is larger than the equilibrium potential, $U > U_{eq}$, the reaction is anodic. If the potential difference is less than the corrosion equilibrium potential, $U < U_{eq}$, the reaction is cathodic. When the equilibrium potential is zero, the anodic and cathodic reactions balance, and the net flux is zero.

Anodic reactions (that produce electrons) are related to positive overpotential. Cathodic reactions (that consume electrons) are related to negative overpotential. In these reactions, the current density, i, is positive for anodic reactions and negative for cathodic reactions.

The current density can be related to the electrochemical potential of a reacting metal using one of the following reaction formulation property methods:

Methods

Butler-Volmer

Tabular

Tafel

Tafel Slope (log 10)

Transport Limited Tafel Slope (log 10)

Nernst Equilibrium Potential

In each of these reaction formulation property methods, temperature, T, is essential to parameterize the rates of electrochemical reactions. In particular, temperature influences the equilibrium potential, U_{eq}, when it is specified using the Nernst Equilibrium Potential method. When either the gas, liquid, multicomponent gas, or multicomponent liquid model is selected, the temperature is determined by the energy models that are selected. If no energy model is selected, STAR-CCM+ uses a constant temperature of 293.15K. However, when one of the Eulerian Multiphase models is selected, the temperature is calculated as follows:

- When the Eulerian Multiphase Mixture model or Volume of Fluid (VOF) model is selected, the default energy models calculate one temperature for the combined mixture of phases.
- When the Multiphase Segregated Flow model is selected, you can select the Phase Coupled Energy model – this energy model solves for temperature in each phase. The Electrochemical Reactions model then calculates a weighted average temperature, T_{echem}:

$$T_{echem} = \frac{\Sigma_i\left(\alpha_i \rho_i c_{p,\ i} T_i\right)}{\Sigma_i\left(\alpha_i \rho_i c_{p,\ i}\right)} \tag{1133}$$

  where for each phase, i:
  - α_i is the volume fraction.
  - ρ_i is the density.
  - $c_{p,\ i}$ is the heat capacity.

Butler-Volmer

When a reaction is reversible, both anodic and cathodic Tafel equations are combined to form the Butler-Volmer equation:

$$j = j_o\left\{\exp\left(\alpha_a \frac{F}{R_u T}\eta\right) - \exp\left(-\alpha_c \frac{F}{R_u T}\eta\right)\right\} \tag{1134}$$

Where α_a and α_c are respectively the anodic and cathodic charge transfer coefficients.

α_a and α_c consider the effective charge transfer coefficient – they are the intrinsic transfer coefficients that are multiplied by the number of electrons, n, that are involved in the reaction.

The exchange current density and equilibrium potential are often dependent on other variables such as the concentrations of reactants and products, temperature, and surface contaminants.

Using Butler-Volmer kinetics the specific reaction current is expressed as:

$$j_{n,s} = j_0 \prod_i \left(\frac{c_i}{c_{i,ref}} \right)^{\gamma_i} \left(e^{\frac{\alpha_a F \eta}{R_u T}} - e^{\frac{-\alpha_c F \eta}{R_u T}} \right) \tag{1135}$$

where

$$e^{\frac{\alpha_a F \eta}{R_u T}} \tag{1136}$$

accounts for anodic currents that are directed from the solid towards the liquid and

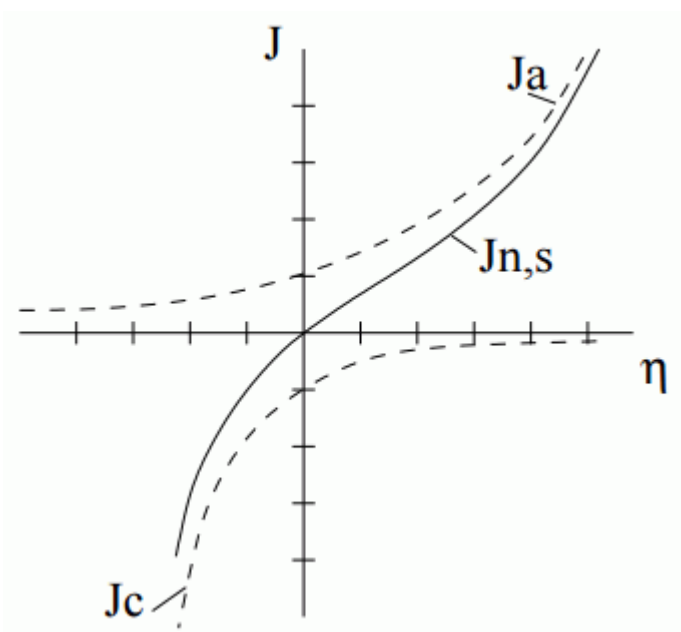
$$e^{\frac{-\alpha_c F \eta}{R_u T}} \tag{1137}$$

accounts for cathodic currents that are directed from the liquid towards the solid.

The anodic and cathodic apparent transfer coefficients, α_a, and α_c, are defined using scalar profiles.

The product term $\prod_i \left(c_i / c_{i,\,ref} \right)^{\gamma_i}$, multiplies the molar concentrations of electrochemical species that are taking part in the reaction, to the power of the rate exponent, γ_i. This term is only considered when the Electrochemical Species model is selected. You can specify γ_i for each reactant and product in the reaction. $c_{i,\,ref}$ represents a reference concentration for species i. In STAR-CCM+, the reference concentration for species i is implicitly assumed to be $c_{i,\,ref}$= 1 kmol / m³.

The following diagram shows the dependence of the specific reaction current $j_{n,s}$ on the surface overpotential, η, for the Butler-Volmer method. Splitting the specific reaction current into anodic and cathodic contributions illustrates the nature of Tafel reactions.



Tabular

The Tabular Polarization Curve method allows you to simulate electrochemical reactions using tabular data that defines the current and electrochemical potential at the interface between the metal and the electrolyte. The Tabular method lets you import polarization curves when modeling species that do not follow the same pattern as generalized by the Butler-Volmer equation.

When modeling corrosion, STAR-CCM+ uses the polarization curves to calculate the corrosion potential for each metal when the current is zero.

When this method is selected, you set the boundary conditions at the interface between the metal and the electrolyte manually. See Tabular Polarization Curve Properties.

Tafel

Butler-Volmer kinetics accounts for both anodic reactions in which free electrons are produced in the solid phase, and cathodic reactions in which free electrons are consumed. In some scenarios, if one of the apparent transfer coefficients is small or operates solely in either positive or negative surface overpotential conditions, reactions can exhibit currents in only anodic or cathodic directions. In such cases, either j_a or j_c is neglected and the negligible reaction term is disregarded. The specific reaction current $j_{n,s}$ then reads:

$$j_{n,s} = \pm j_0 \cdot \prod_i \left(\frac{c_i}{c_{i,\,ref}} \right)^{\gamma_i} \cdot e^{\frac{\pm\alpha \cdot F\eta}{R_u T}} \tag{1138}$$

Since the apparent transfer coefficient α is no longer specified as anodic or cathodic, when using the Electrochemical Species model, the reaction setup determines the direction of the current ($\pm$).

- When electrons are specified as reactants, the current is cathodic ($-\alpha$)
- When electrons are specified as products, the current is anodic ($+\alpha$)

In the absence of the Electrochemical Species model, the direction of the current is specified in the Tafel method properties.

Tafel Slope (log 10)

The Tafel equation is often laid out and parametrized as it is in the Butler-Volmer section, with both exponent variables independent of electric potential. In practice, often these exponent variables are measured as one combined quantity, β, and called the Tafel Slope:

$$\beta = \ln\left(10\right)\frac{R_u T}{\alpha F} = \pm 2.303\frac{R_u T}{\alpha F} \tag{1139}$$

so the specific reaction current is expressed as:

$$j_{n,s} = \pm j_0 \cdot \prod_i \left(\frac{c_i}{c_{i,\,ref}} \right)^{\gamma_i} \cdot 10^{\left(\eta\,\beta^{-1}\right)} \tag{1140}$$

You can specify the combined term β as a scalar profile.

Since β is not specified as anodic or cathodic, the Electrochemical Species model determines the direction of the current ($\pm$). In the absence of the Electrochemical Species model, the direction of the current ($\pm$) is set in the Tafel method properties.

Transport Limited Tafel Slope (log 10)

The Transport Limited Tafel Slope method approximates effects that are not initially accounted for in the simulation setup. When a positive limiting current $j_{\lim}$ is specified, the specific reaction current that is provided by the Tafel method Eqn. (1138) is limited:

$$j_{n,s} = \frac{1}{\frac{1}{j_{\lim}} + \frac{1}{j_{tafel}}} \tag{1141}$$

This method of modeling is empirical and not based on first principles.

Nernst Equilibrium Potential

When the equilibrium potential, U_{eq}, is specified using the Nernst Equilibrium Potential method, U_{eq} is computed as:

$$U_{eq} = E^0 - \frac{R_u T}{nF} \ln\left[\prod_i c_i^{\nu_i}\right] \tag{1142}$$

where ν_i represents the stoichiometric coefficient of electrochemical species i in the reaction. ν_i is positive for anodic reactions / cathodic products, and negative for cathodic reactants / anodic products. E^0 is the Nernst Standard Potential at the reference molar concentration of unity. In STAR-CCM+, the reference concentration for species i is implicitly assumed to be $c_{i,\,ref}$= 1 kmol / m³. To specify alternative reference concentrations, you can define the exchange current density, j_0, using the field function: `j_0=j_0,ref*prod(1/pow(c_i,ref,stoichiometric_coefficient))` where `stoichiometric_coefficient` is ν_i.

Contents:

Electrochemical Reaction Heating Formulation

Electrochemical Reaction Heating Formulation

You can use the Electrochemical Reaction Heating model to account for heat contributions which are due to reversible and irreversible electrochemical processes. This feature is important for applications such as electroplating or solid oxide fuel cells which create high electrochemical currents.

When using the Electrochemical Reaction Heating model, STAR-CCM+ calculates the heat contributions $\dot{q}_A$ that are released from the reacting surface interface according to:

$$\dot{q}_A = \sum_i j_{n,s,i}\, \eta_i + j_{n,s,i} T \frac{dU_{eq}}{dT} \tag{1143}$$

where, for the Butler-Volmer method, the irreversible heat contribution is always positive, $j_{n,\,s,\,i} \cdot \eta_i > 0 \;\forall\; j_{n,\,s,\,i}, \eta_i$.

$j_{n,\,s,\,i}$ represents the boundary specific electric current density on the solid for species i, η_i is the surface overpotential for species i, and $j_{n,s,i} T \frac{dU_{eq}}{dT}$ represents the reversible heat contribution.

Note: If the Tafel method is selected and operates at negative overpotentials that are outside the valid range of approximation, it is possible that negative irreversible heat release rates are predicted as a consequence.

Bulk Ion Chemical Reactions Formulation

For certain types of electrochemistry problems, there are bulk chemical reactions that occur on a short timescale such that they are modeled to reach equilibrium conditions instantaneously.

Instead of specifying forward and reverse rate coefficients, the effect of the reaction is expressed in an equilibrium constant that is defined as:

$$K_{eq} = \frac{k_f}{k_r} \tag{1144}$$

where:

- k_f is the forward reaction rate
- k_r is the reverse reaction rate

Stoichiometric coefficients are used instead of rate exponents since the reactions are assumed to be elementary. K_{eq} is the equilibrium constant:

$$K_{eq} = \prod_i a_i^{v_i} \tag{1145}$$

- v_i is the stoichiometric coefficient of a reactant or product in the reaction (negative for reactants; positive for products)
- $a_{i,\ eq}$ is the activity which is dependant upon the species concentrations of all components, temperature, and other state variables. Since the electrochemical species model assumes that the species are dilute, the following assumption is made:
  - $a_i \sim c_i$ for dilute species
  - $a_i \sim 1$ for pure solvent

  c_i is the molar concentration of a reactant or product, i, in the reaction.

If the order of the reactants and products are switched, to get the same result, use the inverse of K_{eq}.

The equilibrium constant is related to the Gibbs Free Energy change for a given reaction by:

$$\Delta G = -R_u T \ln K_{eq} \tag{1146}$$

The Bulk Ion Chemical Reactions model uses the Coupled Electrochemical Species solver to solve for electrochemical species concentrations.

The residence time is computed as an average of the residence times of each electrochemical species component. You can visualize this value using the field function, **Electrochemical Species Residence Time**.

Electro-Deposition Coating Formulation

This section describes the formulation of the Electro-Deposition Coating model.

The basic transport equation is the equation for the electric potential (Eqn. (3727)) with a zero source term, plus the boundary condition, Eqn. (3729). Here, σ is the electrical conductivity of the liquid bath. At anodes possibly time-varying values for $\overline{\phi}$ are applied. At coated cathodes, you set a user-defined resistance, which is a function of the thickness of the metal sheet and its electrical conductivity. The user-defined resistance contributes to R^{ex} in Eqn. (3729). The potential in the center of the metal sheet is assumed to be zero. The Electro-Deposition Coating model calculates the paint layer resistance, which also contributes to R^{ex} in Eqn. (3729), as follows:

- When using a differential equation to calculate the paint layer resistance, the following starting equation is found to be appropriate:

$$\frac{dR_{\mathrm{paint}}}{dt} = \frac{dh}{dt} r_{\mathrm{paint}} + g \tag{1147}$$

- When specifying the resistivity of the paint layer from which the paint layer resistance is calculated:

$$R_{\mathrm{paint}} = r_{\mathrm{paint}} \times h \tag{1148}$$

- When specifying the paint layer resistance directly:

$$R_{\mathrm{paint}} = R_{\mathrm{paint}}^{\mathrm{user}} \tag{1149}$$

Here h is the paint layer thickness in $[m]$. By default, the offset scalar profile g is zero. The resistivity of the paint layer is r_{paint} and the overall resistance of the paint layer is R_{paint}.

The paint layer growth rate (paint deposition rate) is a function of the boundary-specific electric current j_n at the interface between the paint bath and paint surface. The ODE (Ordinary Differential Equation) for paint layer thickness along a coated body boundary is:

$$\frac{dh}{dt} = \frac{c_{\mathrm{paint}}}{\rho_{\mathrm{paint}}}(j_n - j_{min}) \tag{1150}$$

where j_n is calculated from the boundary conditions. The proportionality factor is the inverse of the electrochemical equivalent E which is:

$$E = \frac{\rho_{\mathrm{paint}}}{c_{\mathrm{paint}}} \tag{1151}$$

ρ_{paint} is the density of the paint solid in $\left[kg/m^3\right]$ and c_{paint} is the solid paint coulomb efficiency in $[kg/C]$. j_{min} is the minimum specific electric current ($\left[A/m^2\right]$) which is required to observe any deposition at the boundary. The formulation assumes that the deposition rate is always positive or zero.

Measurements of polarization effects for different mixtures of paint indicate that there are further mechanisms to control the paint layer growth. Nonlinear functions are integrated and you can use the **Minimum Accumulated Specific Surface Charge** or **Minimum Specific Current Squared Time Integral** scalar profiles to control the contribution to thickness building. Values greater than 0 based on individual paint calibration affect the simulated paint layer thickness distribution.

Electrochemistry Bibliography

[152] *Cagnoni, Davide; Agostini, Francesco; Christen, Thomas; Parolini, Nicola; Stevanović, Ivica; de Falco, Carlo. "Multiphysics simulation of corona discharge induced ionic wind". Journal of Applied Physics, Volume 114, Issue 23, pp. 233301-233301-10 (2013).*

[153] *O. R. J. G. a. J. M. V. Costa-Nunes, "Comparison of the performance of Cu–CeO 2–YSZ and Ni–YSZ composite SOFC anodes with H 2, CO, and syngas," Journal of power sources, vol. 141, no. 2, pp. 241-249, 2005.*

[154] *Douglas P. Riemer, Mark E. Orazem, Application of boundary element models to predict effectiveness of coupons for accessing cathodic protection of buried structures. Corrosion (2000) Vol. 56, No. 8, pp. 794-800.*

[155] *Doyle, C.M. 1995. "Design and Simulation of Lithium Rechargeable Batteries." PhD Thesis, Department of Chemical Engineering, University of California, Berkeley.*

[156] *Georgiadou, Maria and Richard Alkire. "Anisotropic Chemical Pattern Etching of Copper Foil." J. Electrochem Soc (1994): 679-689.*

[157] *K.L. Gering, Prediction of electrolyte viscosity for aqueous and non-aqueoussystems: Results from a molecular model based on ion solvation and a chemicalphysics framework, Electrochimica Acta 51 (2006) 3125.*

[158] *Krupa, Matthew, et al. "Cathodic Protection of Highly Complex and Shielded Components." Corrosion. 2006.*

[159] *Neimarlija N. Demirdžić I. Muzaferija S. "Finite volume method for calculation of electrostatic fields in electrostatic precipitators". Journal of Electrostatics, Volume 67, Issue 1, p. 37-47 (2009).*

[160] *Nels E. Jewell-Larsen, Sergey V. Karpov, Igor A. Krichtafovitch, Vivi Jayanty, Chih-Peng Hsu, Alexander V. Mamishev. "Modeling of corona-induced electrohydrodynamic flow with COMSOL multiphysics". Proc ESA Annual Meeting on Electrostatic 2008, Paper E1.*

[161] *Newman, J. and Thomas-Alyea, K. E. 2004. Electrochemical Systems., 3rd edn. J. Wiley. ISBN: 978-0-471-47756-3.*

[162] *Springer, T.E., Zawodzinski and Gottesfeld, S. 1991. " Polymer Electrolyte Fuel Cell Model", J. Electrochem. Soc., 138(8), pp. 2334-2342.*

[163] *Spotnitz, R. 2006. Porous Electrode Theory, ECS Short Course: Battery Design and Simulation.*

[164] *Wu, H., Li, X. and Berg, P. 2009. " On the modeling of water transport in polymer electrolyte membrane fuel cells" , Electrochimica Acta, 54(27), pp. 6913-6927.*

[165] *Zou, Y., J. Wang, and Y. Y. Zheng. "Electrochemical techniques for determining corrosion rate of rusted steel in seawater." Corrosion Science **53.1** (2011): 208-216.*

Electrochemistry Nomenclature

The following is a guide to the symbols used in this section.

Scalars

α_1	Electrode reaction order [-]
α_2	Vacancies reaction order [-]
α_3	Electrolyte reaction order [-]
α_a	Anodic transfer coefficient [-]
α_c	Cathodic transfer coefficient [-]
$\Delta G°$	Gibbs free energy of reactants and products [J/mol]
η	SEI surface overpotential [V]
η_i	Surface overpotential of species i [V]
κ	Ionic conductivity [S/m] or [A/V m]
μ_e	Chemical potential of electrolyte [J/mol]
ν_+	Number of cations into which a molecule of electrolyte dissociates [-]
$\overline{c_+}$	Uniform/Volume average Lithium/Salt concentration in positive electrode [kmol/m^3]
$\overline{c_-}$	Uniform/Volume average Lithium/Salt concentration in negative electrode [kmol/m^3]
$\overline{x_{init}}$	Volume average initial stoichiometry of negative electrode active materials [-]
$\overline{x}$	Volume average stoichiometry of negative electrode active materials [-]
$\overline{y_{init}}$	Volume average initial stoichiometry of positive electrode active materials [-]
$\overline{y}$	Volume average stoichiometry of positive electrode active materials [-]
ϕ	Electric potential [V]
ϕ_s	Electric potential of the solid [V]
ϕ_l	Electric potential of the liquid electrolyte [V]
ϕ_{corr}	Corrosion potential [V]

ϕ_s^{corr}	Corrosion potential of the solid [V]
ϕ_l^{corr}	Corrosion potential of the liquid electrolyte [V]
σ	Effective electrical conductivity [S/m] or [A/V m]
σ_0	Electrical conductivity [S/m] or [A/V m]
τ	Tortuosity, non-unity only in separator [-]
τ_{eq}	The characteristic timescale for equilibrium.
C	Double-layer capacitance [F/m^2] [C/V m^2]
c	Molar concentration of a single electrolyte [kmol/m^3]
c_0	Molar concentration of solvent [kmol/m^3]
c_i	Molar concentration of species i [kmol/m^3]
c_i^{eq}	Equilibrium molar concentration of a reactant or procuct [kmol/m^3]
c_l	Molar concentration of lithium cations in electrolyte (liquid) [kmol/m^3]
c_s	Molar concentration of lithium in electrode material (solid) [kmol/m^3]
$c_{l,\mathrm{ref}}$	Molar reference concentration of lithium in electrolyte [kmol/m^3]
$c_{s,\max}$	Molar concentration of lithium in saturated electrode material [kmol/m^3]
CAP	Cell capacity [C] [A s]
D	Effective lithium diffusivity [m^2/s]
D_0	Reference lithium diffusivity [m^2/s]
D_i	Diffusivity of species i [m^2/s]
E^0	Nernst Standard Potential at the reference molar concentration of unity [V]
E_a	Kinetic activation energy [J/kmol]
E_{SEI}	SEI activation energy [J/kmol]
F	Faraday's constant [96485339.924 C/kmol]
f	Force [N/m^3]
$f_\pm$	Mean molar activity coefficient of an electrolyte [-]
γ_i	Rate exponent of electrochemical species, i
H	Henry's constant [kmol m^{-3} Pa^{-1}]

j	Specific electric current [A/m^2]
j_0	Exchange current density [A/m^2]
j_n	SEI normal component of electric current density [A/m^2]
$j_{n,l}$	Boundary specific electric current density on the fluid [A/m^2]
$j_{n,s}$	Boundary specific electric current density on the solid [A/m^2]
$j_{n,\ s,\ i}$	Boundary specific electric current density on the solid, for species i [A/m^2]
k_f	Forward reaction rate
k_r	Reverse reaction rate
κ	Ionic conductivity [S/m]
K	Ion mobility [m^2/Vs]
K	Rate coefficient [equiv/s m^2]
K_{eq}	Equilibrium constant
K_0	Reference rate constant [equiv/s m^2]
ξ	Equilibrium concentration constant of the reaction [kg kmol^{-1}]
n	Number of electrons that are involved in electrode reaction [-]
n_+	Li+ cations in positive electrode active material [kmol]
n_-	Li+ cations in negative electrode active material [kmol]
n_{eq}	Equivalent number of electrons per mole of reactant
ν_i	Stoichiometric coefficient of electrochemical species i
N	Number flux [1/m^{2}s]
$\mathrm{N}_{n,\ s,\ i}$	Molar concentration flux of species i from the solid to the liquid [kmol/ m^2 s]
N_A	Avagadro constant 6.022136736 e^{26}[1/kmol]
n_p	Number density [1/m^3]
OCV	Open circuit voltage [V]
$OCV_{SOC=0}$	Open circuit voltage at SOC = 0
$OCV_{SOC=1}$	Open circuit voltage at SOC = 1

q	Elementary charge 1.602176487 e^{-19}[A s]
$\dot{q}_V$	Volumetric heat release [W/m^3]
$\dot{q}_A$	Per-area heat release [W/m^2]
ρ_{ion}	Ionic space charge density [A s/m^3]
r_{surf}	Surface sorption rate
R_u	Universal gas constant [8314.4621 J/kmol K]
R	Electrical resistance [Ω m^2]
R_0	Reference electrical resistance of SEI [Ω m^2]
R_{SEI}	Electrical resistance of SEI [Ω m^2]
s_+	Stoichiometric coefficient of cation in electrode reaction [-]
s_0	Stoichiometric coefficient of solvent in electrode reaction [-]
S_{c_i}	Electrochemical species source term
S_{Estat}	Electric charge density
SOC	State of charge. $SOC \in [0, 1]$ under standard operation conditions. [-]
T	Temperature [K]
t	time [s]
${t_+}^0$	Transference number of cations with respect to the velocity of the solvent species [-]
u_i	Mobility of species i [m^{2}kmol /J s]
U	Potential difference
$U_+(y)$	Relative galvanic equilibrium potential of positive active electrode material [V]
$U_-(y)$	Relative galvanic equilibrium potential of negative active electrode material [V]
U_{eq}	Relative galvanic equilibrium potential of active electrode material [V]
v	Velocity [m^2/s]
V_+	Volume of positive electrode active material [m^3]
V_-	Volume of negative electrode active material [m^3]
χ	Porosity, non-unity only in separator [-]

z_+	Charge number of cation [-]
z_i	Charge number of species i [i]
SEI	Solid electrolyte interface

Tensors or Column Vectors

$\mathbf{E}$	Electric field $-\nabla\phi$ [N/C] or [V/m]
$\mathbf{J}$	Electric current density [A/m^2]
$\mathbf{N}_+$	Lithium (cation) molar flux [kmol/m^{2}s]
$\mathbf{N}_i$	Molar flux [kmol/m^{2}s]
$\mathbf{N}_P$	Number flux [number/m^{2}s]
$\mathbf{v}$	Velocity [m/s]

Cartesian Vectors

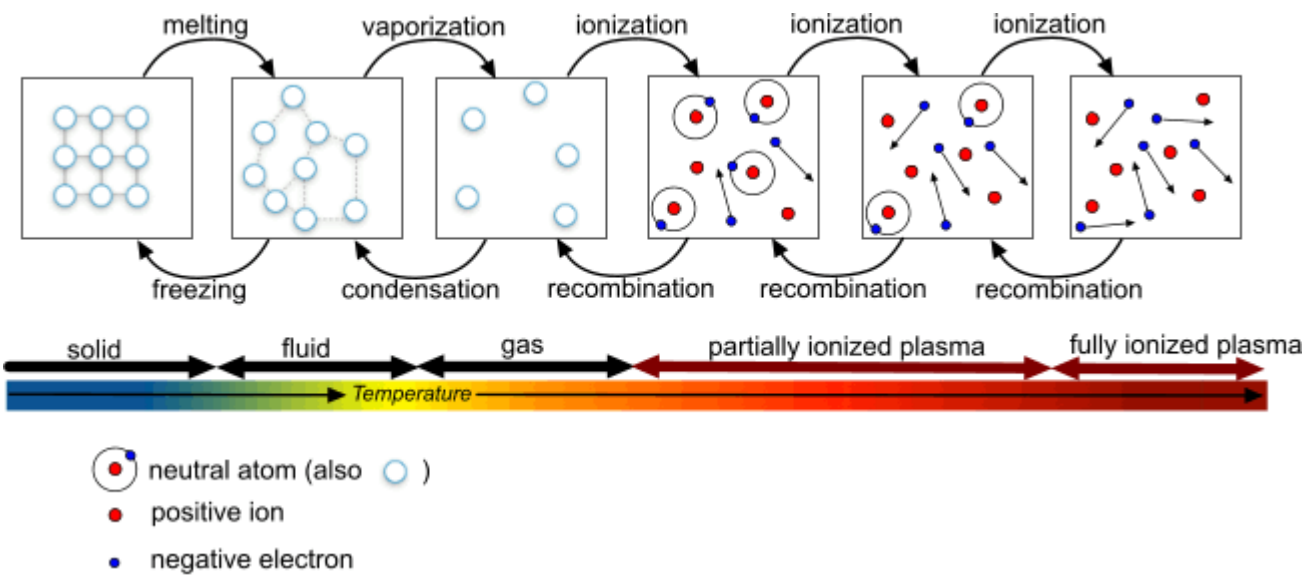
$\mathbf{n}_l$	Normal vector pointing outwards from liquid region [-]
$\mathbf{n}_s$	Normal vector pointing outwards from solid region [-]

Modeling Plasma

Plasma is a state of matter similar to a gas that is composed partially or completely of charged particles such as ions and electrons which are not bound to each other.

What Is Plasma?

Each type of atom has an ionization energy – which is the amount of energy that is required to remove one mole of electrons from one mole of gaseous atoms. When an atom encounters enough energy to overcome its ionization energy, the atom decomposes into an electron and a positively charged ion.



These unbound charged particles are the components of plasma. A gas can display plasma-like properties when only a small proportion of atoms in a sample of the gas are ionized. Electromagnetic fields affect the unbound charged particles – which move collectively as plasma.

There are a large variety of different types of plasma which you can characterize by differences in their properties, such as gas density and electron density. For example, plasma exists in the following places:

Modeling Plasmas

Different types of plasmas are modeled differently depending on their properties and composition. The following modeling assumptions are made:

- When plasmas are in Local Chemical Equilibrium (LCE), the parent gas composition, temperature, and pressure are used to determine the composition of the plasma – which is in quasi-steady state. Since the gas composition, temperature, and pressure are known, you can tabulate the plasma material properties in advance.
- When plasmas are in Non-Local Chemical Equilibrium (non-LCE), the composition of the plasma is fluctuating which means that you must solve the chemical reaction equations.
- When plasmas are in Local Thermal Equilibrium (LTE), the temperature of the electrons is equal to the temperature of the parent gas.
- When plasmas are in Non-Local Thermal Equilibrium (non-LTE), the temperature of the electrons is higher than the parent gas temperature. However, due to the small size of the electrons, the overall temperature of the gas is taken as the parent gas temperature.

Currently in STAR-CCM+, you can model the following types of industrial plasma:

- **Arc plasma**

  This type of plasma consists of highly ionized particles which are generated when a large current of high voltage creates an electric arc between two electrodes. As this electric arc passes through a gas, the gas particles are heated strongly and become ionized. Since the particles are ionized strongly, chemical equilibrium is assumed, and due to the frequent collisions of particles, the electrons remain in thermal equilibrium with the gas temperature. However, it is possible that the particles are in non-thermal equilibrium near the electrodes or walls. In STAR-CCM+, you can use the Ohmic Heating model to model the large temperatures from Joule heating.

  Arc plasmas are used in applications such as plasma torches, circuit breakers, welding torches, and thermal spraying.

  See: Modeling Plasma Arcs.

- **Cold, non-thermal plasma**

  This type of plasma consists of weakly ionized particles which are generated by heating electrons through inductive or capacitive coupling. Since the plasma density is small compared to the gas density, the dilute plasma species have a negligible influence on the gas properties. For this reason, non-LTE plasmas are known as cold plasmas, since the overall temperature of the gas is taken as the parent gas temperature. However, the high energy electrons can still trigger chemical reactions by colliding with background gas particles and causing those particles to ionize. Cold, non-thermal plasmas are used in applications such as plasma-CVD, plasma actuators, coronas, and flow discharges.

See: Modeling Cold, Non-Thermal Plasmas.

Note: If the gas density (Knudsen Number) becomes too low, the continuum assumption – which is required for gas flow modeling using Navier-Stokes equations, becomes invalid.

Contents:

Modeling Cold, Non-Thermal Plasma
Modeling Plasma Arcs
Plasma Nomenclature
Plasma Bibliography

Modeling Cold, Non-Thermal Plasma

Non-Local Thermal Equilibrium (non-LTE) plasma describes a type of plasma in which the electrons in the plasma have a greater thermal energy than the background gas. Despite the electrons having a high thermal energy, the overall temperature of the plasma corresponds with the temperature of the background gas – as the electrons are proportionally much smaller than the gas particles. Therefore, this type of plasma is known as cold, non-thermal plasma. In STAR-CCM+, you can model cold non-thermal plasma using the Coupled Plasma Electron model.

1. For the fluid region in which you are modeling plasma, edit the **[physics continuum]**, make sure that **Auto-select recommended models** is activated, and select the following models:

Group Box	Model
Space	any
Time	**Implicit Unsteady** or **Steady**
Material	**Gas** or **Multi-Component Gas**
Reaction Regime	**Non-Reacting**
Optional Models	**Plasma**
Enabled Models	**Coupled Plasma Electron Model** (selected automatically)
	Electromagnetism (selected automatically)
	Gradients (selected automatically)
	Electrostatic Potential (selected automatically)
Flow	**Coupled Flow**
Enabled Models	**Coupled Species** (selected automatically)
	Coupled Energy (selected automatically)
Equation of State	**Ideal Gas**
Viscous Regime	**Laminar**

2. To model ions within the plasma, also select the **Electrochemistry** model – the **Electrochemical Species** model is selected automatically.
3. Select the **[physics continuum]** > **Models** > **Coupled Plasma Electron** node and set its *Properties*. See Coupled Plasma Electron Model Properties.
4. Expand the **Coupled Plasma Electron** > **e-** > **Material Properties** node and specify values for the Mobility and Molecular Diffusivity of the electrons in the plasma. The Coupled Plasma Electron model solves transport equations using these values that you set.
5. Select the **Gas** or **Multi-Component Gas** node then:

a) Specify the gas components of the background gas in the plasma. See: Managing Single-Component Materials or Managing Mixture Components.

b) For each Gas Component, expand the **[Gas]** > **[Gas Component]** > **Material Properties** node and set its *Properties*. Make sure that you specify the **Elastic Electron Collisional Cross Section**.

c) If the Multi-Component Gas model is selected, you can also set **Multi-Component Gas** > **Material Properties** for the combined gas components.

6. If the Electrochemical Species model is selected for modeling ions:

   a) Select the **Models** > **Electrochemical Species** node and set its *Properties*. See Electrochemical Species Model Properties.

   b) To specify the electrochemical species components, right-click the **Electrochemical Species** > **Electrochemical Species Components** node and choose **Select Mixture Components** or **Import Species**.

   c) When the electrochemical species components are selected, define the Material Properties for each electrochemical species component. See Electrochemical Species Component Properties.

7. Specify conditions and corresponding values for the following:

   - **Continua** > **[Continuum]** > **Initial Conditions**

     Specifically,

     - **Electron Number Density**

       For most cold, non-thermal plasmas, an appropriate value is between $1.0\ \mathrm{E}^{12}$ and $1.0\ \mathrm{E}^{16}$

     - **Electron Temperature**

       For most cold, non-thermal plasmas, an appropriate value is between 0.3 - 2.0 eV

   - **Regions** > **[Region]** > **Physics Conditions**

     Specifically,

     - **Electron Energy Density Source Option**

       Activate this property when you want to define the electron energy density source term, $S_{\in,\ usr}$, which is used to calculate the collective electron energy density source term Eqn. (1158) for the energy density transport equation, Eqn. (1154).

     - **Electron Source Option**

       Activate this property when you want to define the electron source term, $S_{e,\ usr}$, which is used to determine the overall source term contributions for electron number density Eqn. (1156) in the number density transport equation, Eqn. (1152).

   - **[Region]** > **Boundaries** > **[Boundary]** > **Physics Conditions**

     Specifically,

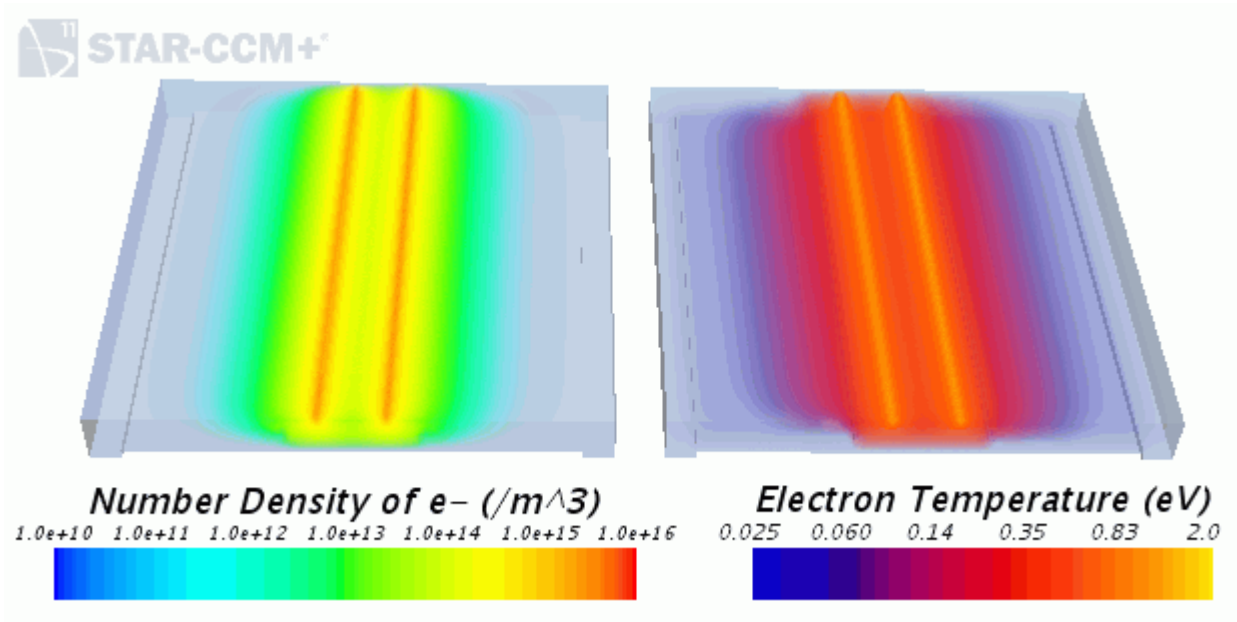
     - **Electron Number Density Specification**
     - **Electron Thermal Specification**

   See: Coupled Plasma Electron Model Reference Topic.

8. Set up any monitors, plots, and scenes that you require.
   For example, you can monitor:

   - Number Density of e-
   - Electron temperature

As a further example, the following scalar scene from STAR-CCM+ shows the distribution of the, number density of electrons and electron temperature, within a process chamber for chemical vapour deposition at a pressure of 133Pa.



9. Select the **Stopping Criteria** > **Maximum Steps** node and set the **Maximum Steps**.
10. Run the Simulation.

Contents:

Plasma Model Reference Topic
Cold, Non-Thermal Plasma Formulation

Plasma Model Reference Topic

You can use the Plasma model with the Coupled Plasma Electron model to model plasmas that are in non-local thermal equilibrium.

Coupled Plasma Electron Model Reference

Theory	See Plasma Formulation.
Provided By	**[physics continuum]** > **Models** > *Plasma*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Coupled Plasma Electron Model**
Requires	*Space*: any *Material*: **Gas** or **Multi-Component Gas** *Optional Models*: **Plasma**
Properties	See Coupled Plasma Electron Model properties.

Activates	Model Controls (child nodes)	**e-**
	Material Properties	**Charge Number**, **Mobility**, **Molecular Diffusivity**, **Molecular Weight**. Also, for the **[gas component]**, the **Elastic Electron Collisional Cross Section** Material Property is available to set.
	Initial Conditions	Electron Number Density, Electron Temperature. See Initial Conditions.
	Boundary Inputs	Electron Number Density Specification, Electron Thermal Specification. See Boundary Settings.
	Region Inputs	Electron Energy Density Source Option, Electron Source Option. See Region Settings.
	Solvers	Coupled Plasma Electron. See Solvers.
	Monitors	Electron Density, Electron Energy
	Field Functions	• Charged Species Mobility of Electron • Elastic Electron Collisional Cross Section • Electron Energy Density • Electron Number Density • Electron Temperature • Molecular Diffusivity of Electron See Field Functions.

Coupled Plasma Electron Model Properties

Transport

Allows you to specify how the electric potential is coupled with the electrons.

- **Electrons < > Electric Potential** This selection is appropriate for modeling most plasma setups.
- **Electric Potential > Electrons**
- **Electrons > Electric Potential**
- **No Electric Potential Coupling**

Secondary Gradients

Neglect or include the boundary secondary gradients for diffusion and/or the interior secondary gradients at mesh faces. See Omitting Secondary Gradients.

- **On**
  Include both secondary gradients.
- **Off**
  Exclude both secondary gradients.
- **Interior Only**
  Include the interior secondary gradients only.

- **Boundaries Only**
  Include the boundary secondary gradients only.

Initial Conditions

Electron Number Density

Sets the electron number density n_e as a scalar profile.

Electron Temperature

Sets the electron temperature T_e as a scalar profile.

Boundary Settings

Wall Boundaries

Electron Number Density Specification

Allows you to choose how the number density, n_e, of the electrons is specified at the boundary.

Method	Corresponding Physics Value Nodes
Impermeable Specifies that the boundary is impermeable to electrons – which is essentially a vanishing flux that is provided at the boundary.	None
Specified Flux Allows you to specify the electron number density as an influx or outflux of electrons. Also provides the physics condition, **Wall Electron Number Density Flux Derivitave Option**.	**Electron Flux** Specifies the electron flux, Γ_e, as a scalar profile.
Specified Value Sets the electron number density to the specified value.	**Electron Number Density** Specifies the electron number density as a scalar profile.

Wall Electron Number Density Flux Derivitave Option

Allows you to specify an electron number density flux derivative.

Method	Corresponding Physics Value Nodes
None	None

Specified	**Electron Flux Derivative** Specifies the electron flux derivative, Γ'_e, as a scalar profile.

Electron Thermal Specification

Allows you to choose how the thermal energy of the electrons is specified.

Method	Corresponding Physics Value Nodes
Specified Electron Temperature Sets the electron temperature T_e.	**Electron Temperature** Specifies the electron temperature as a scalar profile.
Specified Electron Energy Flux Allows you to specify the thermal energy of the electrons as an energy flux. Also provides the physics condition, **Wall Electron Energy Density Flux Derivative Option**.	**Electron Energy Flux** Specifies the electron energy flux, Γ_ϵ, as a scalar profile.

Wall Electron Energy Density Flux Derivitave Option

Allows you to specify an electron energy density flux derivative.

Method	Corresponding Physics Value Nodes
None	None
Specified	**Electron Energy Flux Derivative** Specifies the electron energy flux derivative, Γ'_e, as a scalar profile.

Inflow and Outflow Boundaries

Electron Number Density

Specifies the electron number density, n_e, at inflow and outflow boundaries as a scalar profile.

Electron Temperature

Specifies the electron temperature, T_e, at inflow and outflow boundaries as a scalar profile.

Region Settings

Applies to fluid regions.

Electron Energy Density Source Option

Method	Corresponding Physics Value Nodes

Electron Energy Source When activated, allows you to specify the electron energy density source term and its derivative.	**Electron Energy Source** Specifies the electron energy density source term, $S_{\in, usr}$ as a scalar profile. **Electron Energy Source Derivative** Specifies the derivative of the electron energy density source term.

Electron Source Option

Method	Corresponding Physics Value Nodes
Electron Source When activated, allows you to specify the electron number density source term and its derivative.	**Electron Source** Specifies the electron number density source term, $S_{e, usr}$, as a scalar profile. The electrons are assumed to be cold unless an electron energy source is added. **Electron Source Derivative** Allows you to specify the derivative of the electron number density source term.

Coupled Plasma Electron Solver Properties

Begin

Specifies the iteration or time-step after which the reactions are activated. During previous iterations or time-steps, reactions are deactivated. This feature is useful when modeling significant flow as you can deactivate reactions until a flow solution is found.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Field Functions

Charged Species Mobility of Electron

Mobility of electrons, μ_e

Elastic Electron Collisional Cross Section

Elastic collision momentum transfer cross-section $\sigma_{\zeta e}$ which is used to calculate the collision frequency.

Electron Energy Density

Electron energy density, $\in_e$

Electron Number Density

Electron number density, n_e

Electron Temperature

Electron temperature, T_e

Molecular Diffusivity of Electron

Molecular diffusivity of electrons, D_e

Cold, Non-Thermal Plasma Formulation

This section describes how the Coupled Plasma Electron model behaves when modeling cold, non-thermal plasma.

Transport Equations

The Coupled Plasma Electron model solves two transport equations using a coupled approach – one for the electron number density n_e, and another for the electron energy density, $\in_e$:

- The number density of the electron is evolved using the drift-diffusion-convection-reaction equation:

$$\frac{\partial n_e}{\partial t} + \nabla \cdot \Gamma_e = S_e \tag{1152}$$

  where the electron flux is given by:

$$\Gamma_e = -\mu_e E n_e - D_e \nabla n_e + \upsilon n_e \tag{1153}$$

  where E is the electric field, υ is the gas velocity, D_e is the electron molecular diffusivity, and μ_e is the electron mobility.
- The energy density of the electrons is evolved similarly:

$$\frac{\partial \in_e}{\partial t} + \nabla \cdot \Gamma_{\in} = S_{\in} \tag{1154}$$

where the electron energy flux is given by:

$$\Gamma_{\in} = -\frac{5}{3}\left(-\mu_e E \in_e - D_e \nabla \in_e + \upsilon \in_e \right) \tag{1155}$$

where a Maxwellian electron energy distribution function (EEDF) is assumed and the electron temperature, T_e, is given by:

$$T_e = \frac{2}{3} \frac{\in_e}{k_B n_e}$$

Source Terms

- Source term Contributions for Electron Number Density

  The electron number density source term is given by:

  $$S_e = S_{e,\ usr} \tag{1156}$$

  where $S_{e,\ usr}$ is the electron production / consumption user source term and

  $$S_{e,\ usr} = P_{e,\ usr} \tag{1157}$$

  where $P_{e,\ usr}$ is a user-specified electron source profile in which the electrons are assumed to be cold. Since the electrons are added without energy, the electrons cool down – unless an electron energy source is added.
- The collective electron energy density source term is given by:

  $$S_{\in} = S_{\Omega} + S_{el} + S_{\in,\ usr} \tag{1158}$$

  where:

  - S_{Ω} is the ohmic heating source term and is given by:

    $$S_{\Omega} = -q_e \Gamma_e \cdot E \tag{1159}$$

    This source term accounts for the increase in energy when electrons are running against an electric field and the loss of energy when the electrons are running in the direction of an electric field.
  - S_{el} is the elastic collision source term which accounts for the elastic collisions of the electrons with other main gas species. This source term arises from the collisions of the electrons with the background gas components and is given by:

    $$S_{el} = \sum_{\zeta} -3\frac{M_e}{M_{\zeta}} k_B \left(T_e - T \right) K_{\zeta e} \tag{1160}$$

    where ζ is the sum over the background gas components with a molar mass of M_{ζ}. The electron momentum transfer collisional rate $K_{\zeta e}$ is given by:

    $$K_{\zeta e} = \sigma_{\zeta e} n_{\zeta} n_e \upsilon_{th,\ e} \tag{1161}$$

    where

- $\sigma_{\zeta e}$ is the user-specified effective elastic collision momentum transfer cross-section which is specific to the type of background gas components.
- n_{ζ} is the number density of the main gas.
- $v_{th,\ e}$ is the thermal velocity of the electrons:

$$v_{th,\ e} = \sqrt{\frac{8}{\pi}\frac{k_B T_e}{m_e}} \tag{1162}$$

- $S_{\in,\ usr}$ is the electron energy user source term:

$$S_{\in,\ usr} = P_{e,\ usr} \tag{1163}$$

Modeling Plasma Arcs

When plasma is described as Local Chemical Equilibrium (LCE) plasma, the composition, temperature, and pressure of the background gas and plasma are equal. Since the electrons have the same thermal energy, composition, and pressure as the background gas, an equilibrium is assumed. Assuming an equilibrium where the properties of the gas are known allows you to use these tabulated material properties to define the plasma in STAR-CCM+. Local Thermal Equilibrium (LTE) plasma describes a type of plasma in which the electrons in the plasma have the same thermal energy as the background gas.

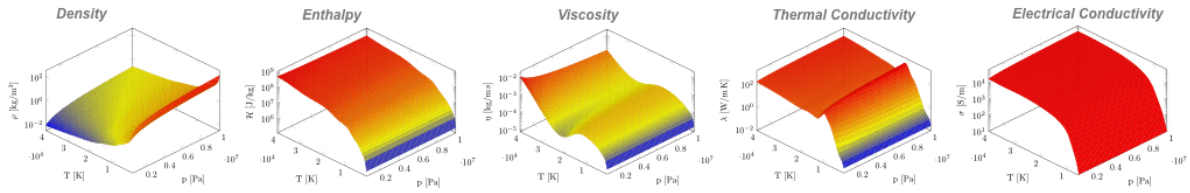
1. For the fluid region in which you are modeling plasma, edit the **[physics continuum]**, make sure that *Auto-select recommended models* is activated, and select the following models:

Group Box	Model
Space	**Axisymmetric**, **Three Dimensional** or **Two Dimensional**
Time	Any implicit
Material	**Gas**, **Multi-Component Gas**, or **Eulerian Multiphase** To model air plasma with temperatures that are below 25,000K, select **Gas**. To model welding applications, select **Eulerian Multiphase**.
Reaction Regime (when Multi-Component Gas is selected)	**Non-reacting**
Flow (when Gas or Multi-Component Gas is selected)	**Coupled Flow** is appropriate for most plasma arc setups **Coupled Species** (selected automatically when Multi-Component Gas is selected)
Eulerian Multiphase Model (when Eulerian Multiphase is selected)	**Volume of Fluid (VOF)** **Multiphase Equation of State** (selected automatically) **Segregated Flow** (selected automatically)
Enabled Models	**Gradients** (selected automatically)
Equation of State (when Gas or Multi-Component Gas is selected)	When you are modeling air plasma with temperatures that are below 25,000K, select **Real Gas** and then the **Equilibrium Air** model. For all other applications, select **User Defined EOS**.
Optional Models	**Electromagnetism**
Electromagnetism	**Electrodynamic Potential**

Group Box	Model
Optional Models	**Magnetic Vector Potential**
	Two-Way Coupled MHD
Viscous Regime	Any, although **Turbulent** is the most appropriate choice when modeling plasma arcs setups.
Enabled Models	**Reynolds-Averaged Navier-Stokes** (selected automatically)
Reynolds-Averaged Turbulence	Any, although **K-Epsilon Turbulence** is the most appropriate choice when modeling plasma arcs setups.
Enabled Models	**Realizable K-Epsilon Two-Layer** (selected automatically)
	Exact Wall Distance (selected automatically)
	Two-Layer All y+ Wal Treatment (selected automatically)
Optional Models	**Radiation**
Radiation	**Participating Media Radiation (DOM)**
Radiation Spectrum (Participating)	Any
Optional Models	**Ohmic Heating**
	Quasi Unsteady Losses

Note: When the **User Defined EOS** model is selected, it is assumed that the plasma / gas is incompressible – and only dependant on temperature. This assumption is valid for flow speeds below Mach 0.7. For flow speeds above Mach 0.7, select the User Defined EOS model node and activate the Compressible property. Then populate the Material Properties with Temperature-Pressure dependant tables. See Using Table (T,P).

2. If you are using the **Gas** or **Multi-Component Gas** model, select the **Gas** or **Multi-Component Gas** node then:
   a) Specify the gas components of the background gas in the plasma. See: Managing Single-Component Materials or Managing Mixture Components.
   b) For each Gas Component, expand the **[Gas]** > **[Gas Component]** > **Material Properties** node and set its *Properties*. Where possible, import the tabulated material property data using the *Table (T, P)* method. For more information on importing tabular data, see Tables and Using Table (T, P).
   The following image shows temperature and pressure dependent thermodynamic and transport material properties of air plasma in LTE and LCE [167].



c) If the Multi-Component Gas model is selected, you can also set **Multi-Component Gas** > **Material Properties** for the combined gas components.

3. If you are using the **Eulerian Multiphase** model, set up the phases and define any phase interactions. See Defining Eulerian Phases and Defining Phase Interactions.

In plasma arc simulations, the Ohmic Heating energy model heats the plasma. However, forcing an electrical current through an isolated gas at room temperature causes numerical instability due to the large scale of ohmic heating.

4. To avoid destabilization, you can either:
   - initialize the plasma arc using a temperature that is large enough to result in a sufficiently large electrical conductivity (1 S/m - 100 S/m), or,
   - set an artificial conductivity by specifying a minimum value for the Electrical Conductivity Constraint Option.
5. Specify conditions and corresponding values for the following:
   - **Continua** > **[Continuum]** > **Initial Conditions**
   - **Regions** > **[Region]** > **Physics Conditions**
   - **[Region]** > **Boundaries** > **[Boundary]** > **Physics Conditions**

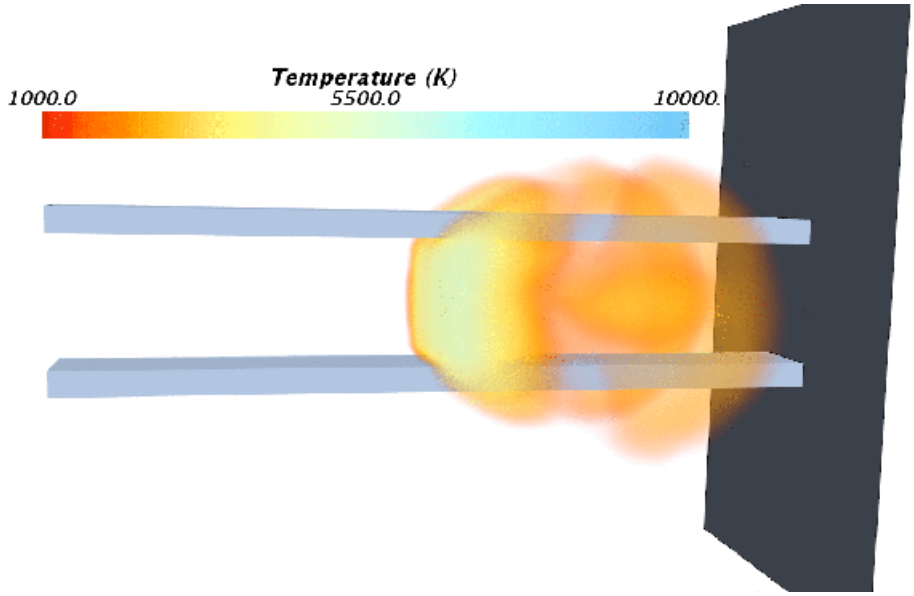
Since plasma simulations are non-linear, lower under-relaxation factors are required for some solvers.

6. Expand the **Solvers** node and check that the Properties for each solver are appropriate for the application that you are modeling. Use the following values as a guideline for modeling most types of plasma arc simulations:

Node	Property	Value
[Coupled Flow]	Courant Number	0.5 – 2.0
Electric Potential	Under-Relaxation Factor	0.3 – 0.9
Magnetic Vector Potential	Under-Relaxation Factor	0.4

If you use the Coupled Flow solver, you can also use the expert driver to further improve the stability of the simulation. See Using the Expert Driver.

7. Set up any monitors, plots, and scenes that you require.
   For example, you can monitor:
   - Temperature
   - Velocity



8. Select the **Stopping Criteria** > **Maximum Steps** node and set the **Maximum Steps**.
9. To improve convergence, you can adjust the time-step size and the number of inner iterations to suit the application that you are modeling.
   a) Select the **Solvers** > **[Implicit Unsteady]** node and set the *Time-Step*. Increase the value for non-compressible flows and decrease the value for supersonic flows.
   b) Select the **Stopping Criteria** > **Maximum Inner Iterations** node and set the *Maximum Inner Iterations*. For unsteady simulations, increase the *Maximum Inner Iterations* to a value that is between 24-96.
10. Run the Simulation.

Plasma Nomenclature

D	Diffusivity [m²/s]
D_e	Electron molecular diffusivity [m²/s]
$\in_0$	Vacuum permittivity [F/m]
$\in_e$	Electron energy density [J/m³]
E	Electric field [V/m]
k_B	Boltzmann constant [m² kg/s²/K]
M_e	Molar mass of electrons [kg/kmol]
M_ζ	Molar mass of background gas components [kg/kmol]
n_e	Electron number density [1/m³]
n_ζ	Number density of the background gas [1/m³]
N_A	Avagadro constant $6.02214129 \times 10^{26}$ [1/kmol]
$\sigma_{\zeta e}$	User specified effective elastic collision momentum transfer cross-section [m²]
$P_{e,\ usr}$	User specified electron source profile [1/m³s]
$P_{\in,\ usr}$	User specified electron energy source term [J/m³/s]
q_e	Electric elementary charge [C]
ρ	Density [kg/m³]
S_e	Electron number density source term [1/m³s]
$S_{e,\ usr}$	Electron production / consumption user source term [1/m³/s]
$S_\in$	Electron energy density source term [J/m³s]
$S_\in$	Electron energy density source derivative [1/m³/s]
S_Ω	Ohmic heating source term [J/m³s]
$S_{\in,\ usr}$	Electron energy source term [J/m³s]
T	Background gas temperature [K]
T_e	Electron temperature [K]
t	Time [s]
Γ_e	Electron flux [1/m²s]

$\Gamma_{\in}$	Electron energy flux [J/m²s]
Γ'_{e}	Electron flux derivative [1/m²s]
$\Gamma'_{\in}$	Electron energy flux derivative [1/m²/s]
μ	Mobility [m²V/s]
μ_{e}	Mobility of electrons [m²/V s]
υ	Gas velocity [m/s]
$\upsilon_{th,\ e}$	Thermal velocity of electrons [m/s]
Z	Charge number

Plasma Bibliography

[166] *GJM Hagelaar and LC Pitchford. Solving the Boltzmann equation to obtain electron transport coefficients and rate coefficients for fluid models. Plasma Science and echnology, 14(4):722, 2005*

[167] *A.D'Angola, G.Colonna, C.Gorse, M.Capitelli. Thermodynamic and transport properties in equilibrium air plasmas in a wide pressure and temperature range. The European Physical Journal D, January 2008, Volume 46, Issue 1, pp 129-150.*

Modeling Batteries

The battery modeling capability in STAR-CCM+ is used to simulate batteries. This process involves designing individual battery cells and battery cycling procedures in the external software package Battery Design Studio® (BDS).

Currently, only spirally wound cylindrical cells and stacked cells that are designed in BDS can be used in STAR-CCM+. Once the battery model has been activated in STAR-CCM+, the output files from BDS can be imported into the simulation and used to set up battery cells, battery modules, and electrical loads. The battery module generation process is then used to first create geometry and connector parts and then regions for the battery objects. A battery mesher is provided to mesh the battery objects. It is also possible to create a volume that automatically encases the battery, which can be used to define a fluid continuum around the battery. A circuit solver is used to connect battery modules with electrical loads.

This section also includes formulation and a reference section for battery model field functions.

Licensing Requirements

For battery modeling, STAR-CCM+ requires a batterysim license in addition to the other license features.

Battery Design Studio®, which is supplied for use with the STAR-CCM+ battery modeling capability, requires a batterydesignstudio license.

The batterysim license cannot be used for Battery Design Studio®.

Contents:

What Is Battery Modeling?
Selecting the Battery Model
What Is the Battery Solver?
Working with Battery Design Studio
Preparing Program (.prg) Files
Working with a Battery Cell
Setting Up a Battery Module
Generating a Battery Module
Generating a Battery Mesh
Setting Up Connector Ohmic Heating
Defining External Battery Casing
Defining an Electrical Load
Creating a Circuit
Exporting Electrochemistry Data
Evaluating Electrode Structure Tortuosity

Battery Modeling Formulation
Battery Field Functions Reference

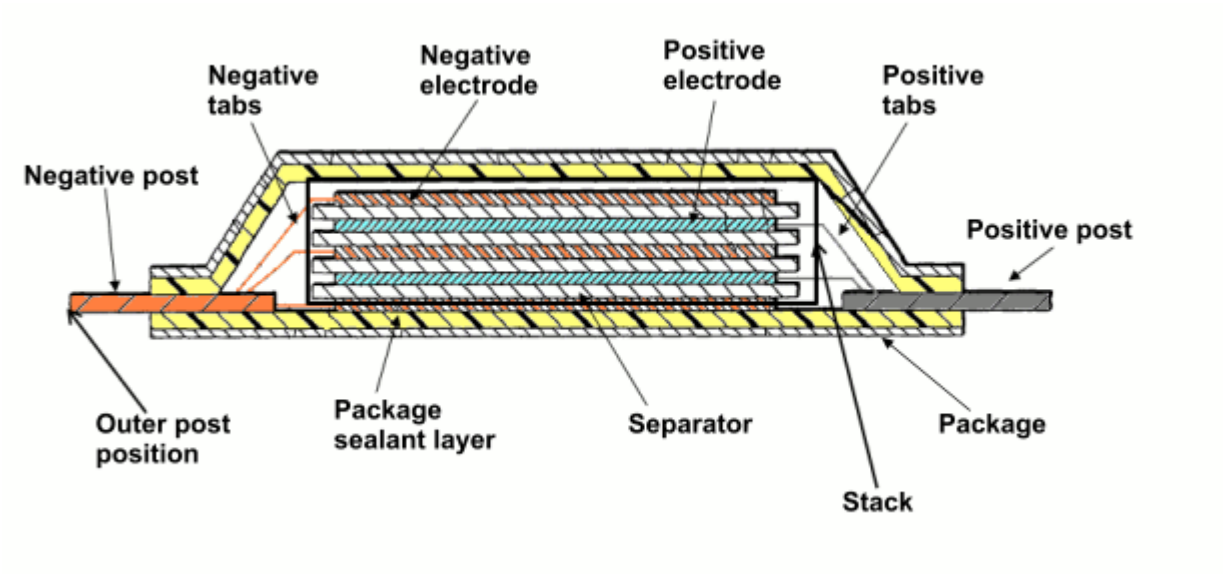
What Is Battery Modeling?

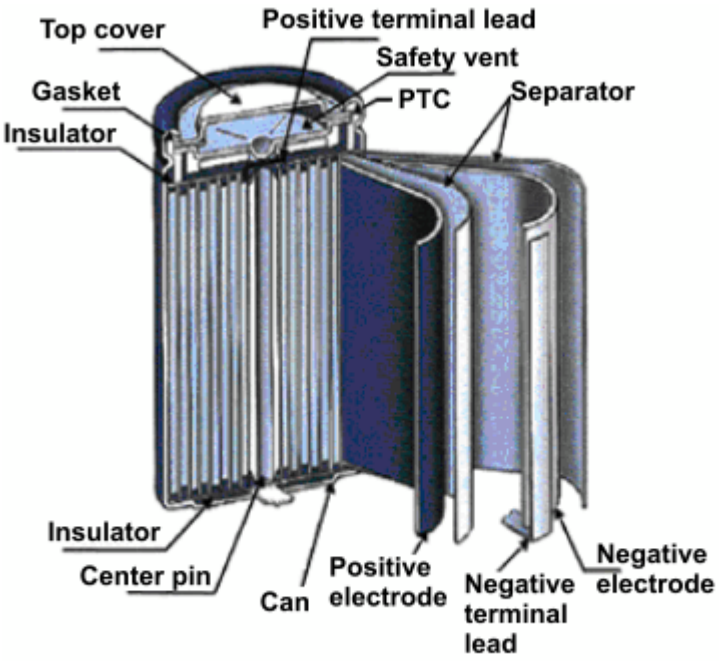
Battery modeling is the process of simulating the electrical and thermal response of a battery over the course of a load cycle.

In STAR-CCM+, this is done using a Battery Simulation Module (BSM), which uses a combination of an electrical solver and a fluid/thermal solver. The electrical solver calculates the electrical behavior of the battery cell while the fluid/thermal solver calculates the heat transfer within the cell and its surroundings.

Battery Structure Terminology

You can model two types of battery cells in STAR-CCM+: stacked and cylindrical spiral cells. Each cell type is associated with its own basic terminology. The following images depict the general terminology that is used to describe each cell structure.





- Electrodes—the cell components that conduct the electrochemical reaction (every electrode has a tab). Electrodes are comprised of coating on a metal foil. The positive electrode is typically a metal oxide or metal phosphate on an aluminum foil while the negative electrode is typically a graphite paste on copper.
- Separator—a layer that is made of non-conductive material that keeps two electrodes separated, thus preventing a short-circuit. A separator allows ionic transport.
- Tabs—flat connectors that protrude from each electrode
- Stack—the electrodes and separators within the battery cell, excluding tabs
- Package/Can—an outer layer that is used to house cell components
- Package sealant layer—the layer that contains the active chemicals
- Posts—the parts that connect electrode tabs to other conducting objects, such as another battery cell (also known as busbars)
- Terminal leads—current collectors from either the positive or negative electrode to the can
- Center pin—a support that is left in the core of a spiral cell to support its shape, typically made of steel or plastic
- Insulators—typically polyolefin disks that are used to prevent the ends of the jellyroll from touching the bottom or top of the package
- Gasket—an insulating sealant material that is used to electronically isolate the top cover from the can
- Top cover—a top cover for the package, typically made of nickel plated steel
- Safety vent—a rupture disk that is used to release pressure
- PTC—Positive Temperature Coefficient device that is put in series between the positive terminal lead and the top cover. The electrical resistance of the PTC greatly increases with temperature. In this way, it functions as a safety device that prevents current flow when the battery is hot, and limits high currents that can heat up the PTC.

Contents:

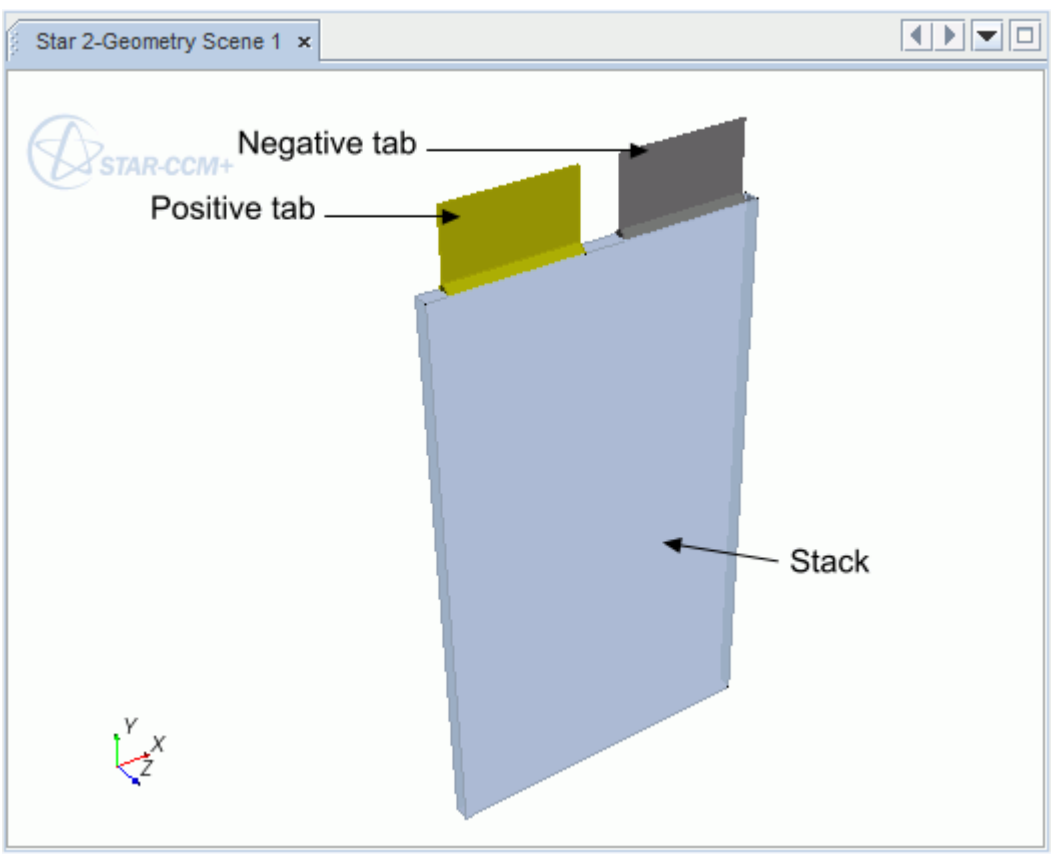
Understanding the Battery Modeling Process
The Battery Modeling User Procedure

Understanding the Battery Modeling Process

Battery modeling in STAR-CCM+ closely follows the physical structure of the battery, defined by either a stacked or spirally wound cylindrical assembly of the battery cell.

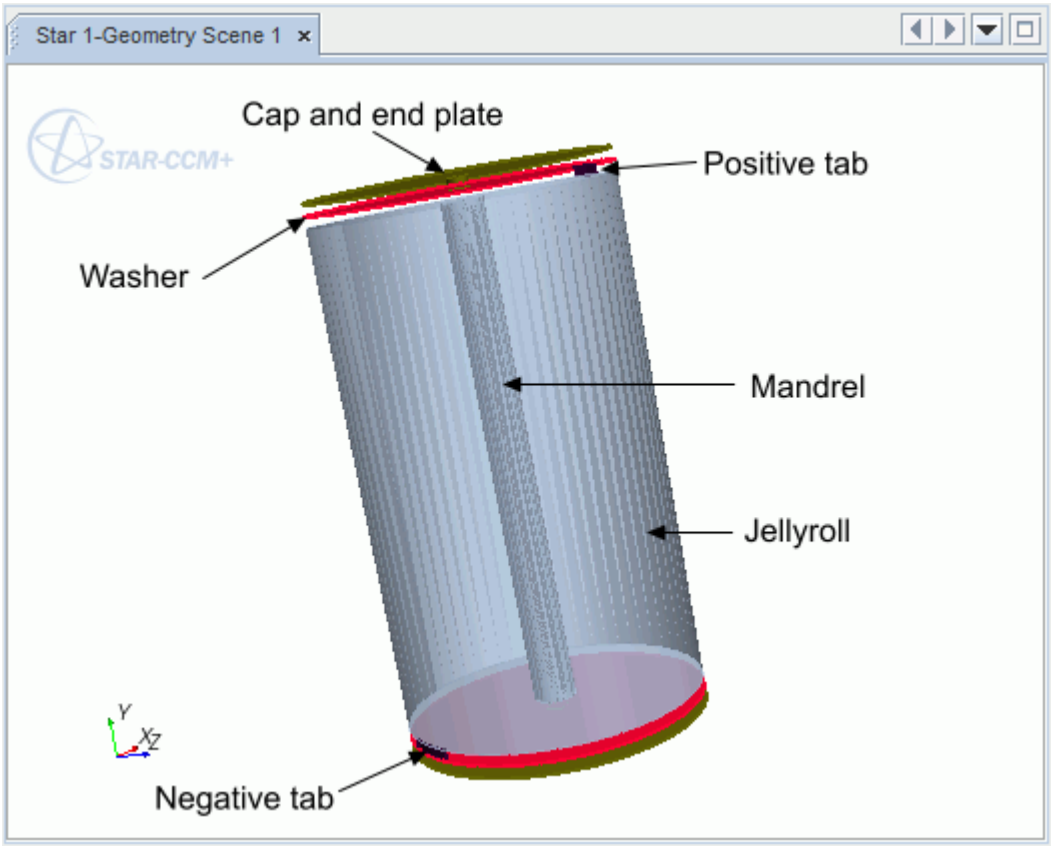
The battery modeling process starts with defining a battery cell design in Battery Design Studio® (BDS), which is a separate software for battery design and simulation. In addition to cell design, BSM also incorporates the battery models and numerical algorithms of BDS into STAR-CCM+. That is, most BDS models have been made a part of STAR-CCM+ and can be used with a more detailed thermal model that is unique to STAR-CCM+.

Once the battery cell is assembled in BDS, it is imported into STAR-CCM+. For a stacked cell, the model only includes the stack and the tabs. In STAR-CCM+, the collection of tabs from all the positive electrodes in the cell is referred to as the positive tab for the battery cell. Similarly, the collection of negative tabs from all the negative electrodes is referred to as the negative tab for the battery cell. The following screenshot shows an example of a single battery cell that has been imported into a simulation.



For a cylindrical spiral cell, the model includes a jellyroll that is made up of a positive and negative electrodes. The jellyroll is wound on a cylindrical mandrel. Also imported are the negative and positive tabs, a washer that

joins all the tabs together, caps (internal posts) that connect the washer to the actual posts, and a can and end plates that enclose the cell. The following screenshot shows an example of a single cell in STAR-CCM+. The can is hidden to show the interior parts.



In STAR-CCM+, sets of battery cells can be connected together to form a battery module.

Contents:

What Is an Ecell?
What Is an Electrical Solver?
What Is a Thermal Solver?
What Is Battery Modeling Solver Coupling?

See Also:

Working with Battery Design Studio
Creating Battery (.tbm or .ebm) Files in BDS

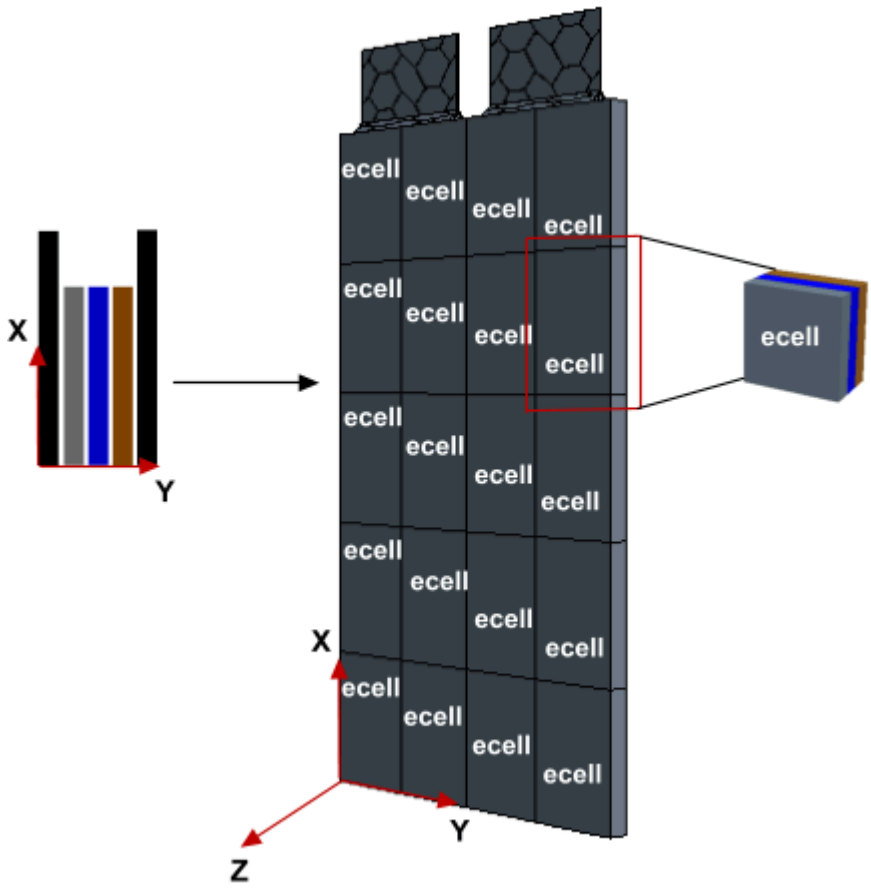
What Is an Ecell?

Ecells define the structured electrical grid on the battery cell and are used in STAR-CCM+ to run the electrical solver portion of the battery modeling process. For each ecell, the electrical solver implements a one-dimensional battery model to compute voltage as a function of time, current, and temperature.

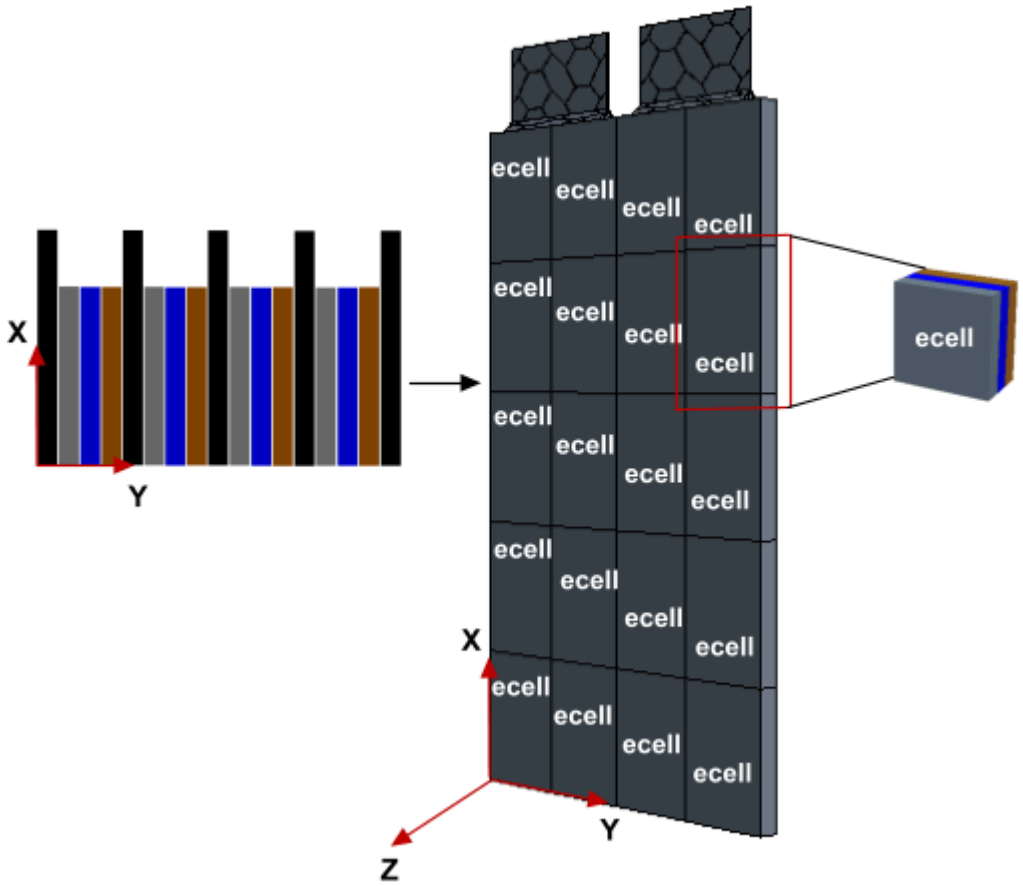
Ecells model actual cells, which are made up of one or more pairs of positive and negative electrodes. Each ecell consists of a positive plate, a unit cell, and a negative plate to represent a positive electrode, separator, and negative electrode of an actual cell. The ecell representation is done internally and cannot be viewed in STAR-CCM+.

Initially, the number of ecells in the x, y, and z directions for a stacked cell, and in the x and y directions for a spiral cell is specified in the imported BDS file. For a spiral cell, the ecells are not of equal size and the specified number in BDS is automatically adjusted based on the layout of the battery cell and tabs. The adjusted number is imported into STAR-CCM+ and cannot be modified. For a stacked cell, the ecells are of equal volume and their number can be changed in STAR-CCM+.

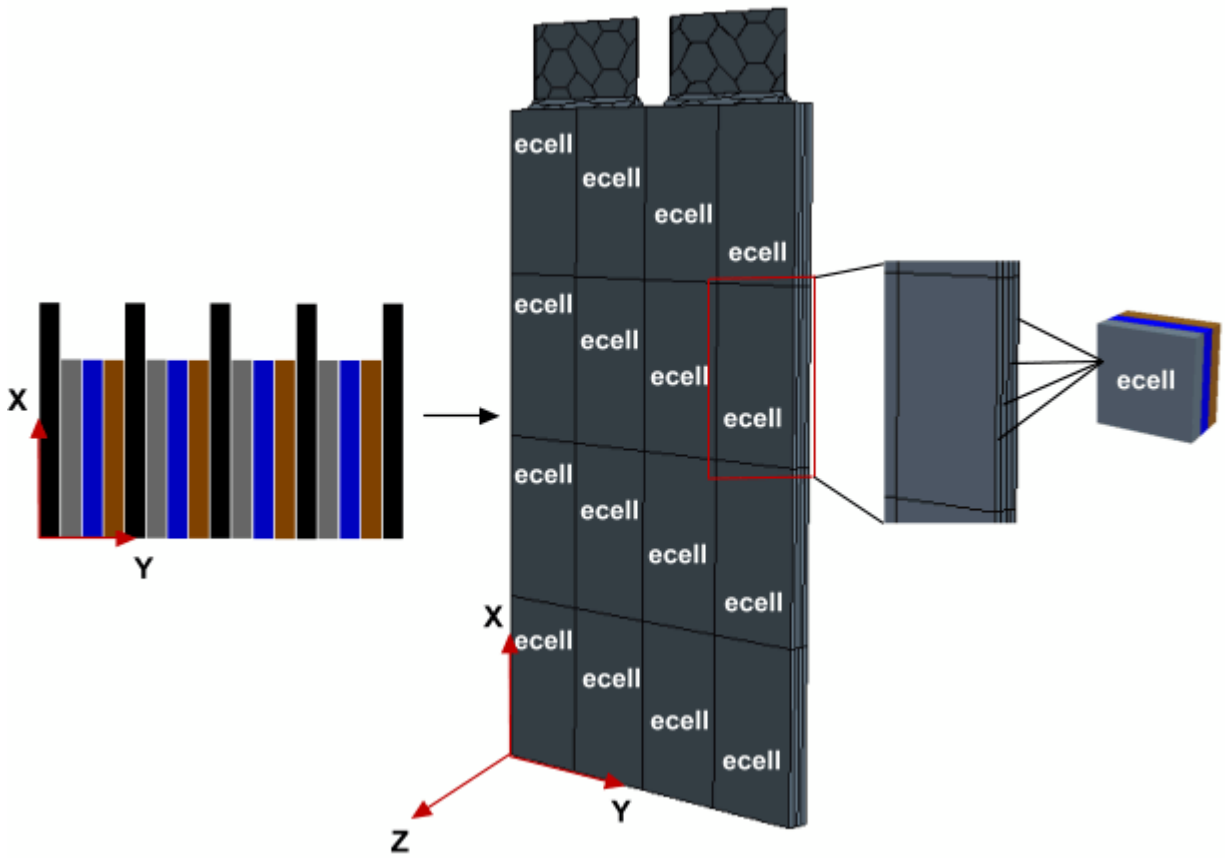
For stacked cells, part of the modeling strategy of STAR-CCM+ is to allow you to decide the extent to which the discretization model gets simplified in order to reduce computational expense. For instance, in the following image an actual cell with one pair of positive and negative electrodes is represented in STAR-CCM+ in terms of ecells on the x-y plane. In this case, there is not any spatial variation of heat generation in the z direction.



For an actual cell with more than one pair of positive and negative electrodes, the STAR-CCM+ representation can again be modeled as one ecell in the z direction, meaning that the spatial variation of heat generation is only on the x-y plane.



Alternatively, this actual cell can be modeled in STAR-CCM+as two, three or four cells in the z-direction. Using this modeling strategy results in spatial variation of heat generation in the x, y and z directions.



The thermal solution within the battery stack is independent of the ecell discretization. Instead, it uses its own user-defined, unstructured grid resolution. The communication between the electrical and thermal solutions is done using internal mapping between the electrical mesh and the thermal mesh.

What Is an Electrical Solver?

The BSM electrical solver calculates the electrical voltage and heat generation as well as, if applicable, chemical solution for each battery cell.

It computes the voltage and heat generation on a grid of ecells incorporating the battery unit cell model type and parameters that are specified in the same imported BDS file that defines the battery cell design. The available unit cell models are NTG, RCR, and DISTNP. Regardless of whether a 1D, 2D, or 3D representation of the model is chosen in BDS, STAR-CCM+ uses the equivalent 3D representation.

See Also:

NTG (Newman, Tiedeman, Gu) Model

RCR Model

DistNP Model

What Is a Thermal Solver?

The BSM thermal solver calculates the temperature field within the battery cell stack and throughout the rest of the thermal model and cooling system.

The thermal solver takes the internal heat generation values calculated by the electrical solver and outputs the local temperature for each thermal grid cell. These values are then mapped to the ecells and used as inputs for the electrical solver.

The thermal solver uses a fixed time step that is controlled by user-defined iteration parameters.

What Is Battery Modeling Solver Coupling?

In STAR-CCM+, the battery modeling process involves running the electrical and thermal solvers sequentially for each thermal time step, starting with the electrical solver.

The thermal time step is determined through a user-defined number of inner iterations. To increment the time variable, the thermal solver needs to complete this number of inner iterations. From this point forward, we refer to the thermal time step as simply time step.

At each time step, the coupled solver process is:

- Map the stack temperature distribution from the previous time step to the electrical mesh. If the current time step is the first, then start from the default initial state.
- Solve the electrical solution using the electrical solution from the previous time step as the initial condition. For the first time step, the solver starts from a default initial state. The first time step may take substantially longer than subsequent time steps as the electrical solver iterates as many times as necessary to guarantee convergence.
- Map the heat generation field from the electrical mesh to the thermal mesh.
- Solve for the thermal solution by using the number of inner iterations that you define.
- Check the solution stopping criteria and either stop or proceed to the next time step.

You control the convergence behavior of the thermal solver by setting the time step parameters. For the electrical solver, because the equations are solved at smaller, adaptive time increments, the BSM handles the solution convergence automatically.

To set an appropriate time step, you should consider the rate of change of the electrical load. The time step must be small enough so that the loading function gets resolved correctly, meaning that if the electrical load changes too abruptly between time steps the electrical solver may fail to converge. Furthermore, if the electrical load is specified using a program file and its time step is much finer than that defined for the thermal solver, the heat generation output from the electrical solver to the thermal solver might not be correctly captured. This occurs if the electrical load changes significantly several times within one time step.

The Battery Modeling User Procedure

This section outlines the general workflow for battery modeling.

The general workflow for battery modeling involves the steps listed below. A battery modeling tutorial is available in the tutorial guide, which introduces and walks you through the main concepts.

1. Design and optimize a battery cell in Battery Design Studio® (BDS).

   This step includes saving battery cell data in a text battery model (`.tbm`) file or an encrypted battery model (`.ebm`) file, and program data in a program file (`.prg`) file, if this is the way you want to specify the load for the electrical solver.
2. Activate the battery model in the physics continuum.
3. Import the battery cell created in BDS into the STAR-CCM+ simulation.
4. In STAR-CCM+, create and set up a battery module:
   - Specify the number of battery cells in the battery module.
   - Arrange the battery cells within the module.
   - Define the electrical cell stack resolution.
   - Prepare terminal connections and tab connections.
   - Set resistance values.
5. Generate the battery module parts and connector parts.

   This step may include specifying an external casing part around the battery module.
6. Generate the battery module regions.
7. Set up the battery mesh.

   Ensure that the external casing is assigned to a different mesh continuum than the battery.
8. Generate the volume mesh.
9. If necessary, set up ohmic heating for the connector parts.
10. Apply an electrical load to the battery by importing the program file (`.prg`) defined in BDS or by using a table.
11. Connect the battery module and the electrical load in a circuit.
12. Run the simulation.
13. Post-process results using the battery field functions.

See Also:

Working with Battery Design Studio
Creating Battery (.tbm or .ebm) Files in BDS
Selecting the Battery Model
Creating a Battery Cell
Setting Up a Battery Module
Generating a Battery Module
Generating a Battery Mesh
Setting Up Connector Ohmic Heating
Defining Electrical Circuits

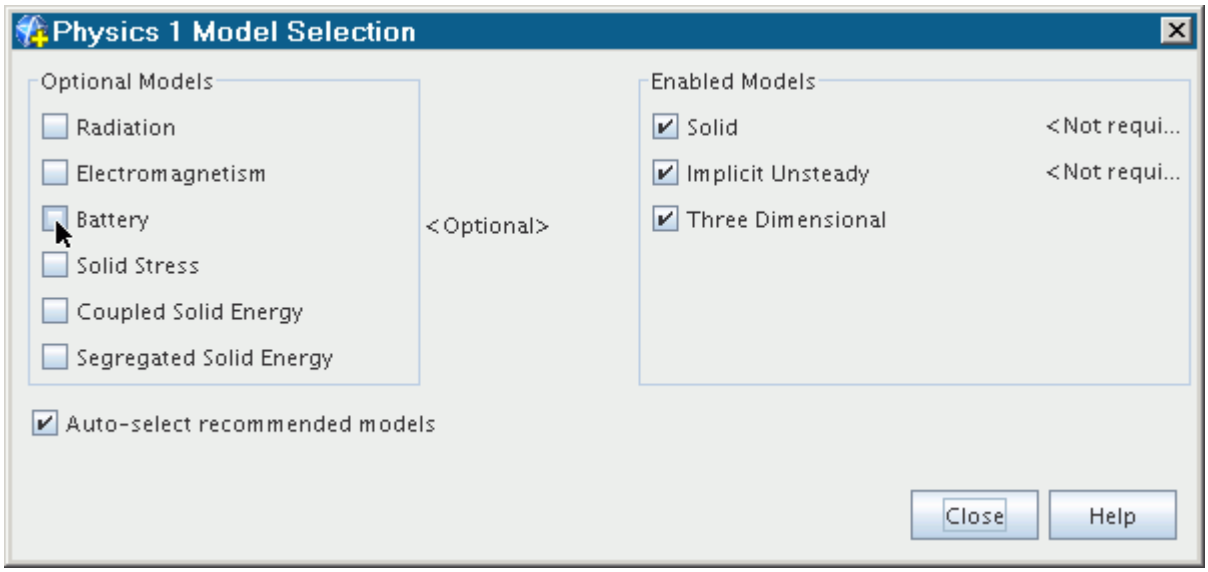
Selecting the Battery Model

The Battery model is only available for solid, three-dimensional continua, and can only be used with the Implicit Unsteady solver.

Selecting the Battery model within a Physics continuum reveals the Batteries manager node in the simulation tree.

To select the Battery model in the *Physics Model Selection* dialog, do the following:

1. Select **Three Dimensional** from the *Space* group box.
2. Select **Implicit Unsteady** from the *Time* group box.
3. Select **Solid** from the *Material* group box.
4. Select **Battery** from the *Optional Models* group box.



When you select the Battery model, the Circuit model is selected automatically.

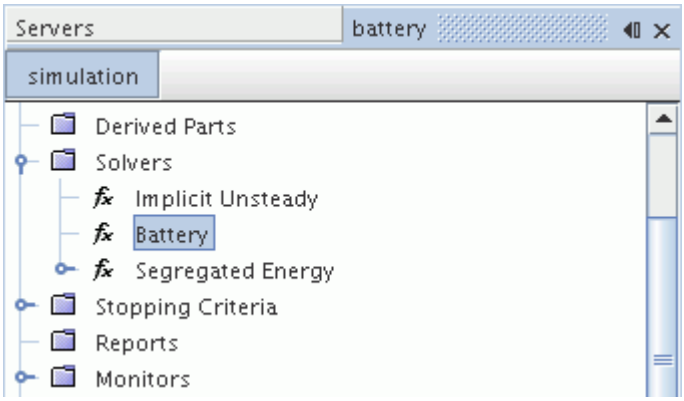
5. Select an *Energy* model and an *Equation of State* model.

   These models are required for Battery simulations.

What Is the Battery Solver?

The Battery solver controls the solution of the Battery model in the continua for which it is activated.

The Battery solver has properties and is subject to common menu actions.



Expert Properties

Solver Frozen	Prevents or allows the solver from updating the solution while iterating. Activating this feature should not be your first choice, as it is primarily a “debugging” tool and not supported for normal simulation work.	
	Activated	The solver does not update the solution.
	Deactivated	The solver updates the solution.

Working with Battery Design Studio

Battery Design Studio® (BDS) is a software package for designing, simulating, and analyzing batteries. It provides tools for costing and sizing batteries, visualizing cell designs, generating test procedures and correlating model results to performance, safety, and life predictions.

BDS serves as an interface for materials suppliers, battery developers, and engineers to collaborate on the battery design. BDS can be used to design stacked cells and spirally wound cylindrical and prismatic cells. In BDS, you can specify the physical construction of the cell, then use a simulation model to estimate how that cell behaves under various use conditions.

In order to run a STAR-CCM+ simulation on a battery, you need to create a battery cell file (.tbm) in BDS. Additionally, you can select to specify the electrical load using a program file (.prg), which is also created in BDS.

This section includes some basic information on defining the battery and program file in BDS. For further information, refer to the BDS documentation. Battery Design Studio® is only available for Windows.

Contents:

Creating Battery (.tbm or .ebm) Files in BDS
Configuring a Stacked Cell
Configuring a Spiral Cell

Creating Battery (.tbm or .ebm) Files in BDS

In Battery Design Studio® (BDS), all the information about the battery is stored in the Text Battery Model file (`.tbm`). This information includes the physical description of the battery as well as the model parameters.

STAR-CCM+ supports stacked cells and spirally wound cylindrical and prismatic cells.

For improved security, you can create an encrypted battery file. An encrypted battery file (an `.ebm` file) can be used in the same way as a `.tbm` file, but the file contents cannot be viewed or modified.

Creating a new battery cell

To create a new battery cell in BDS:

1. In the BDS workspace, right-click the **Batteries** node on the left side of the window.
2. In the context menu, click **Add new Battery** and then select the appropriate battery type.

   The default options for each battery type are:

   - **Lilon Simple** > **LilonSimple**

     Creates a simple cell.
   - **Lilon Spiral** > **LilonSpiral**

     Creates an idealized cell that you can use for modeling a cylindrical spiral cell. It is set up using the Distributed 3D model.
   - **Lilon Stack** > **LilonStack**

Creates a stacked cell.

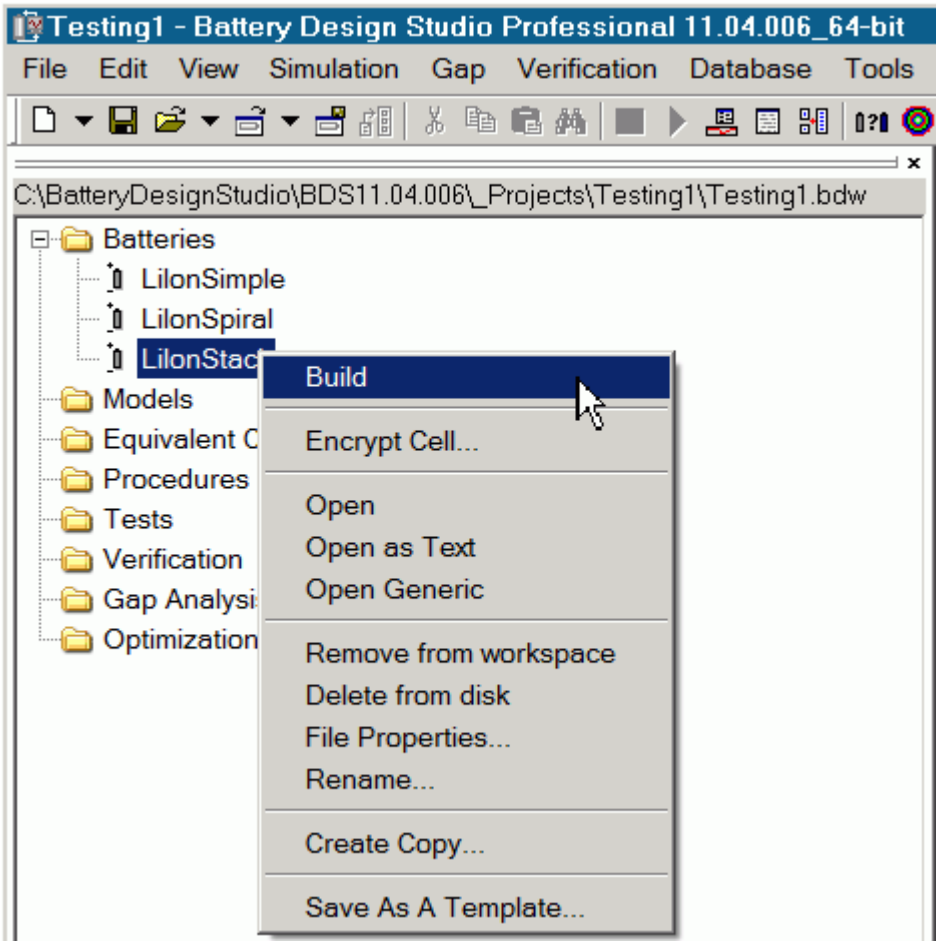
The new battery cell is added under the **Batteries** node and the corresponding battery definition window is displayed in the workspace. You can rename the battery and specify the appropriate configuration settings.

Building the battery file

Once you have configured the battery in BDS, you need to build the battery file. The build process recalculates the values displayed in the Report tab and writes the battery configuration information to a `.tbm` file.

To build the battery file:

3. Under the **Batteries** node, right-click the battery that you want to build, and then select **Build**.



When the build process is complete the message **Build Successful** is displayed. If any errors in the cell setup are detected, an error message is displayed.

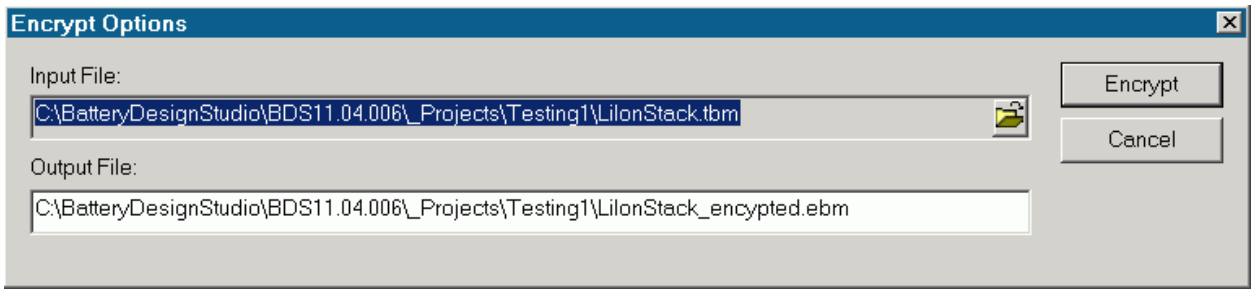
Encrypting the battery file

If you want to distribute your battery file without the contents being exposed, you can encrypt the battery file. You must build the battery file as a `.tbm` file before you can create the encrypted battery file.

To encrypt the battery file:

4. Under the **Batteries** node, right-click the battery that you want to encrypt, and then select **Encrypt Cell**.

   The *Encrypt Options* dialog appears:



5. In the **Output File** box, specify the location and name of the encrypted file that you want to create.
6. Click **Encrypt**.

   The encrypted battery file is added under the **Batteries** node and is saved to the specified location. You cannot modify the encrypted file; when you open the file in BDS you can only view the data sheet. If you want to modify the file contents, you need to make the appropriate changes in the `.tbm` file and then encrypt the file again.

See Also:

NTG (Newman, Tiedeman, Gu) Model

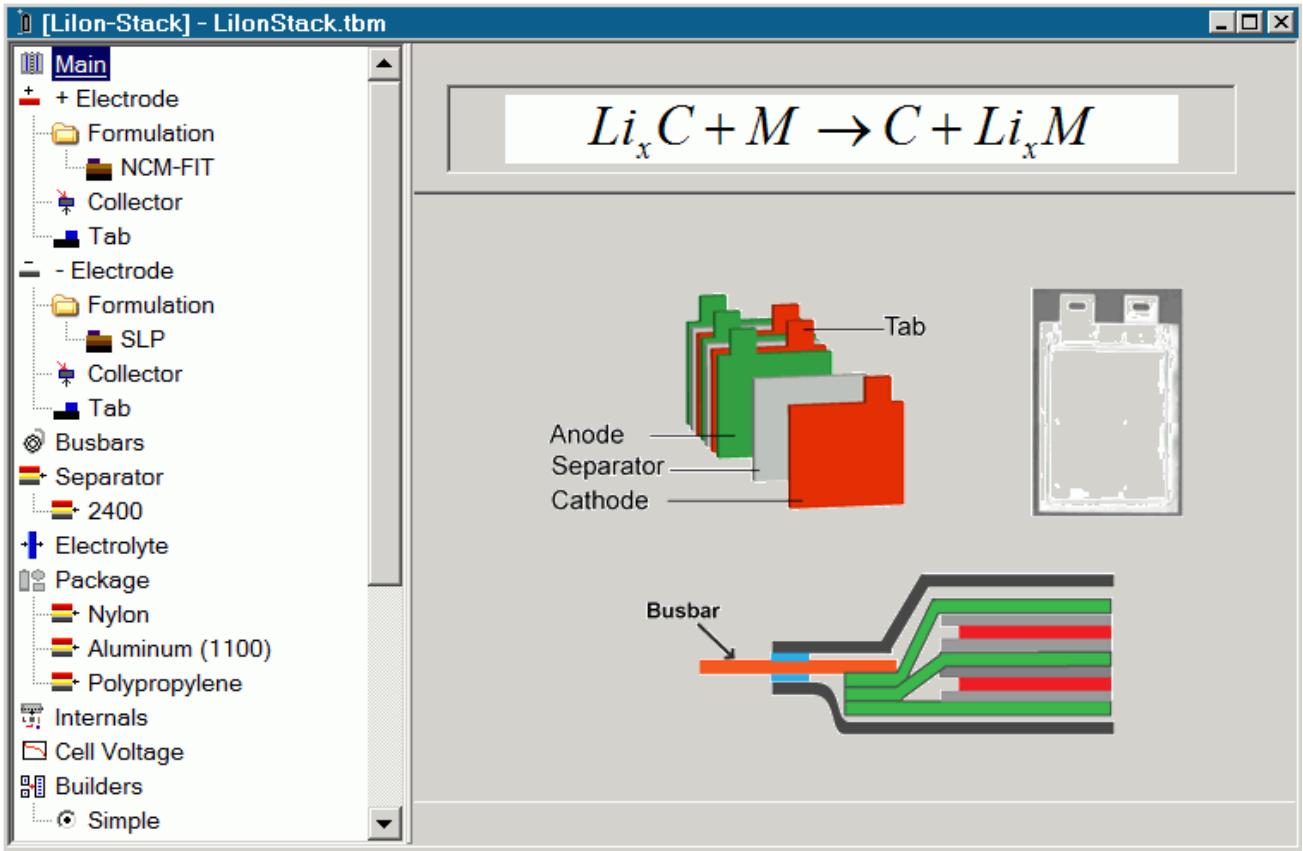
Configuring a Spiral Cell

Configuring a Stacked Cell

Configuring a Stacked Cell

When you create a new stacked cell in Battery Design Studio® (BDS), the appropriate battery definition window is displayed in the workspace.

The following shows the battery definition of a LiIon pouch battery.



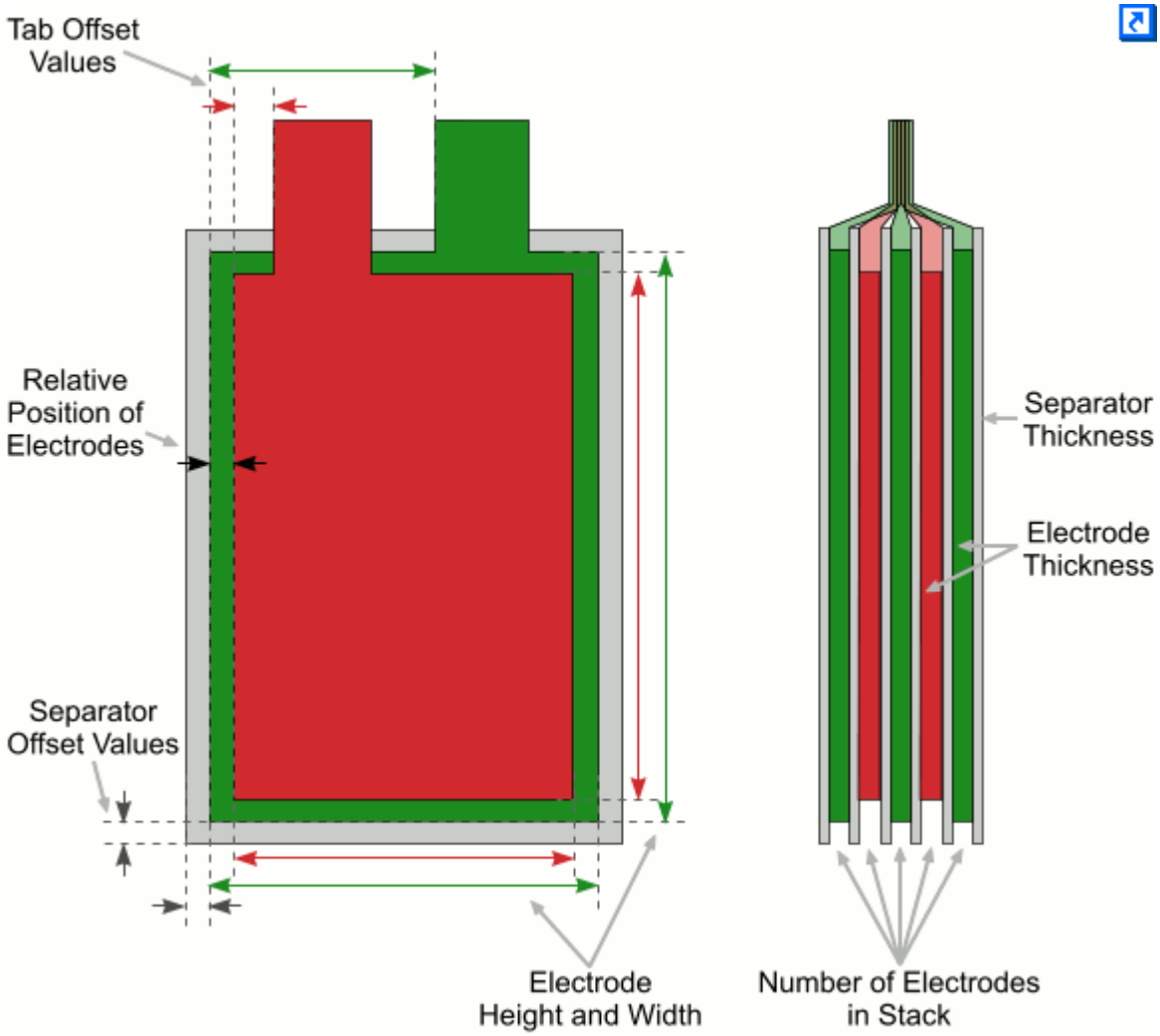
You can use the nodes on the left of the battery window to define characteristics and properties of the battery, including the dimensions of the cell.

Defining the Dimensions of a Stacked Cell

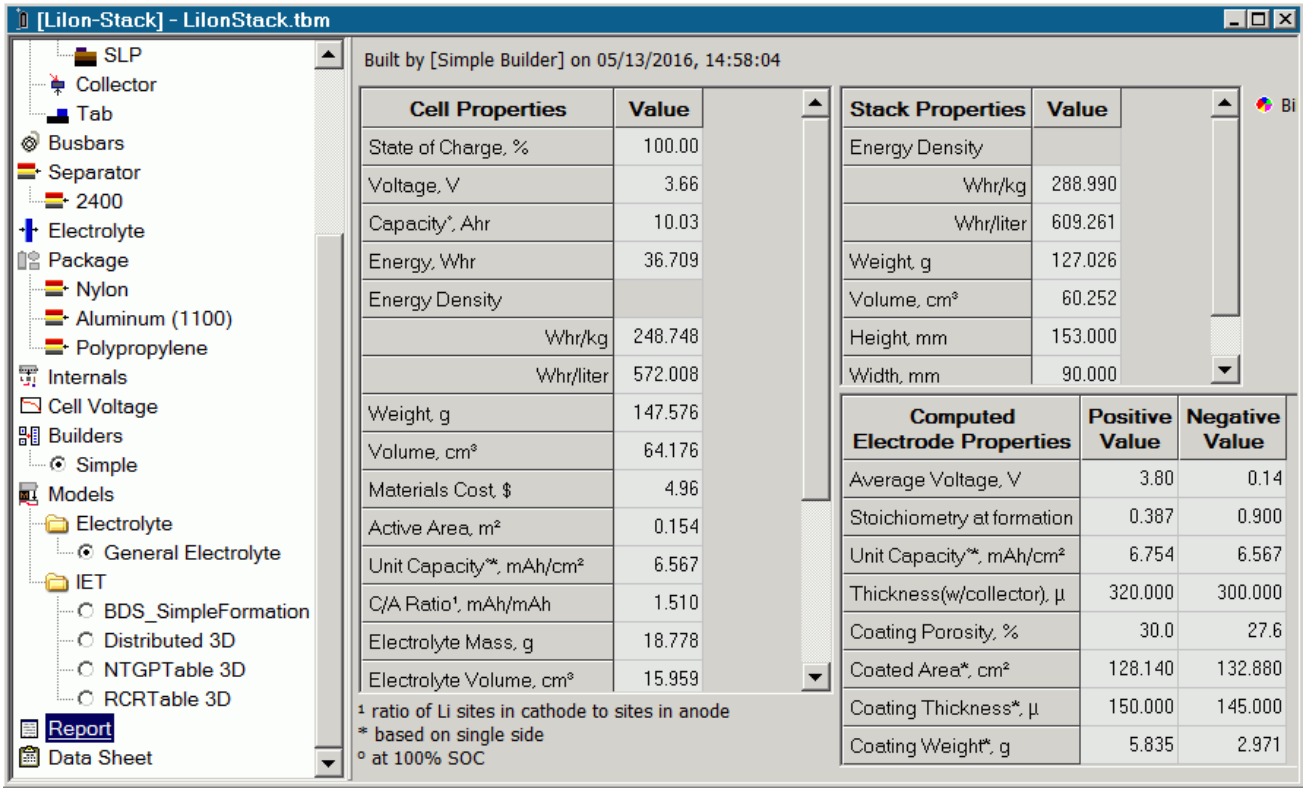
The cell dimensions are specified in BDS within the battery cell definition. For a stacked configuration, the battery cell imported into STAR-CCM+ includes the positive and negative electrodes and tabs, and also the separators.

More information on the parameters that STAR-CCM+ uses is given below. Other cell components, such as packaging and busbars, are not imported into STAR-CCM+.

The cell dimensions depend on the following attributes:



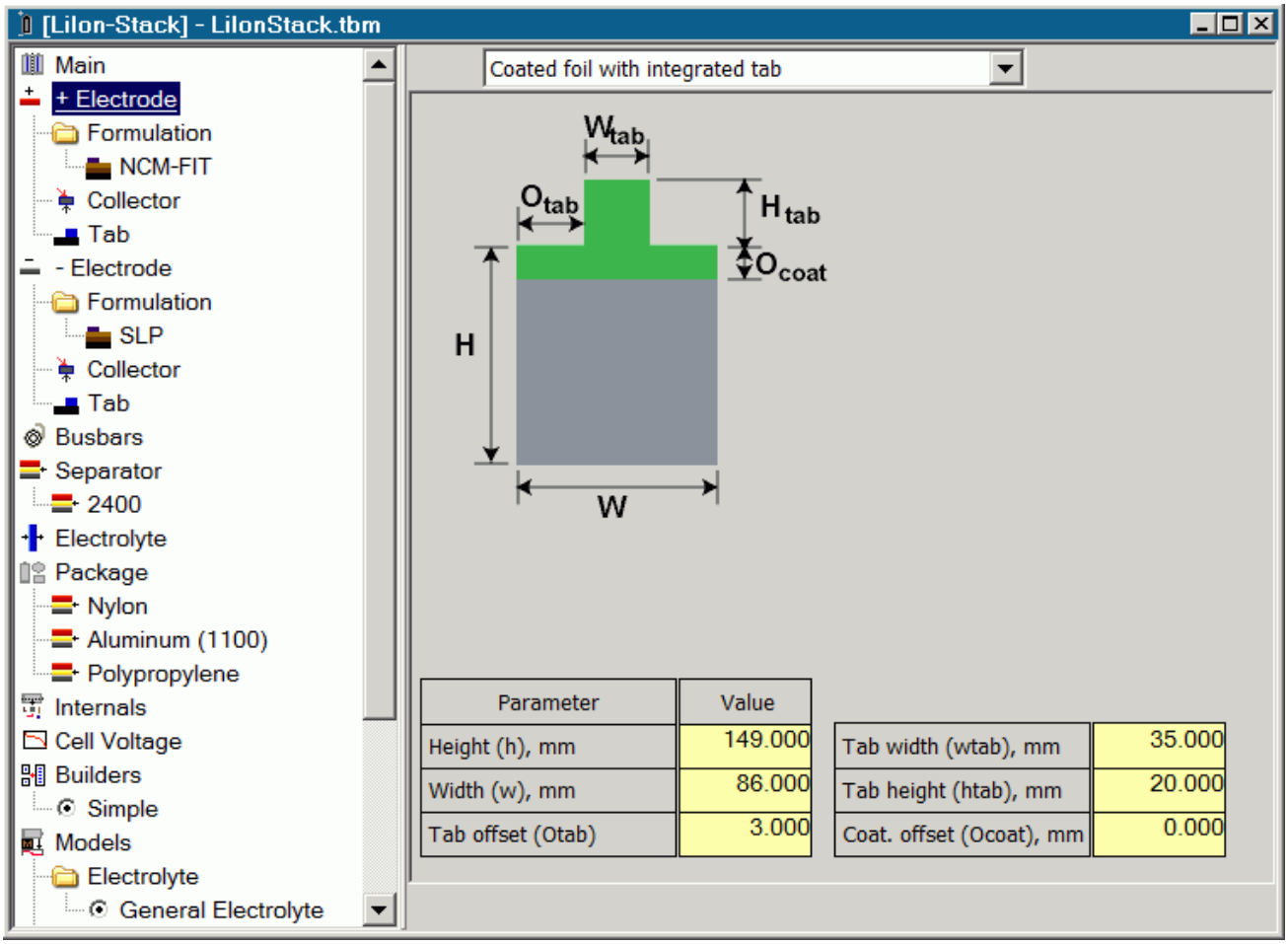
To see a summary of the values that are used to define the cell size, select the **Report** node in the battery window. The cell dimensions are in the Stack Properties section.



Electrode Height and Width

You can define the electrode size in the battery window. In the left panel, select the appropriate node, **+ Electrode** or **- Electrode**. Then, in the right-side of the window, enter the appropriate height and width values for the electrode.

When you set the electrode sizes, ensure that the width of the negative tab is larger than the width of the positive tab.



Tab Height, Width, and Offset Values

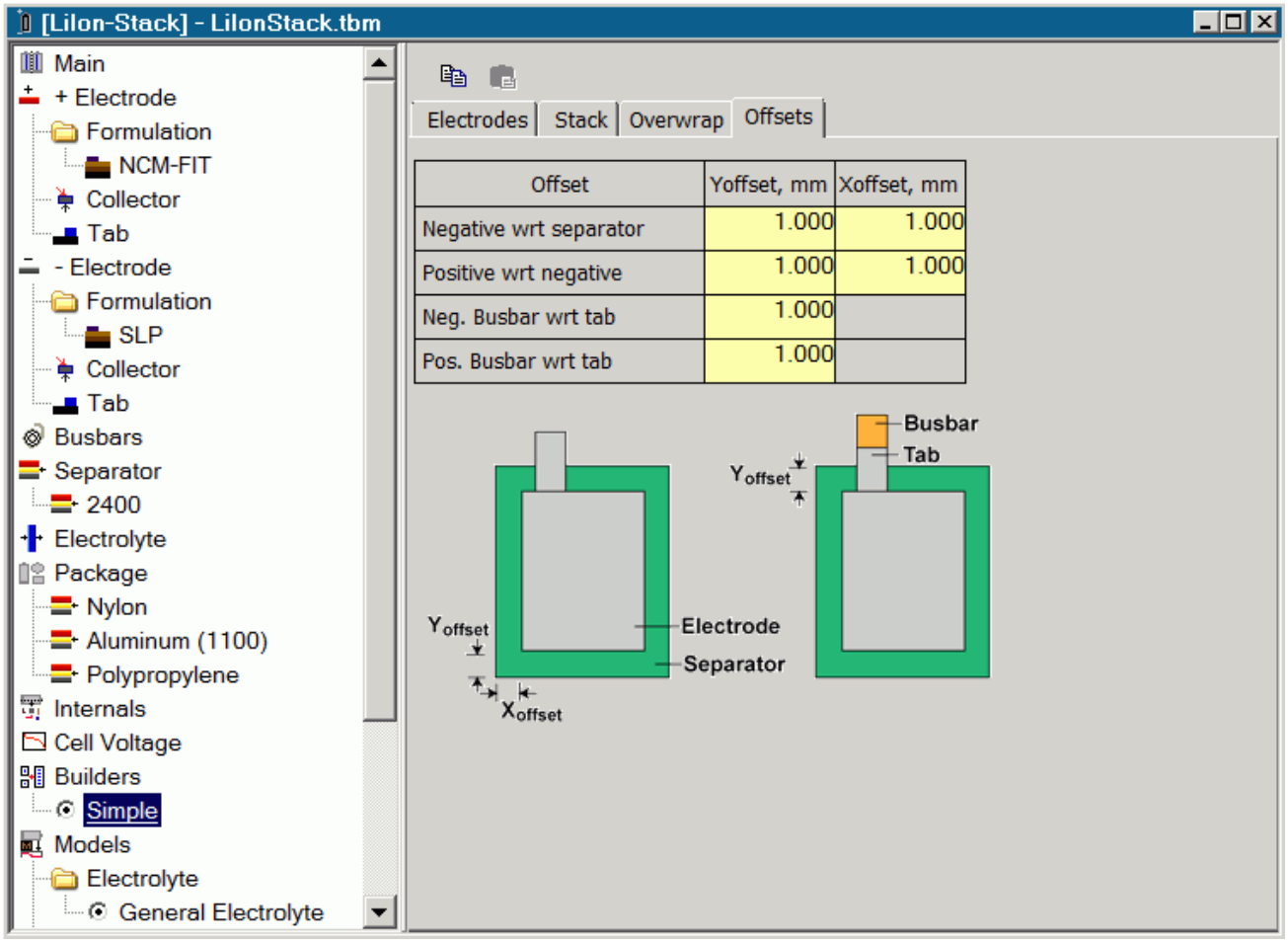
You can also define the tab dimensions in the **+ Electrode** or **- Electrode** node. You can specify the tab height and width, and also define the offset of the tab from the left edge of the electrode.

The tab offset values are specified from the edge of each electrode individually. Since the negative electrode is typically wider than the positive electrode, you need to set the relative position of the electrodes in order to align the tabs correctly.

Separator Offset Values

You can define the size of the separators in the battery window. In the left panel, select the **Builders** > **Simple** node, and then select the *Offsets* tab.

The X and Y offset values of the *Negative wrt separator* property specify how far the separator layer extends from edges of the negative electrode. For example, if the negative electrode was 88 mm wide, specifying an *Xoffset* value of 1 mm would result in a separator layer 90 mm wide.



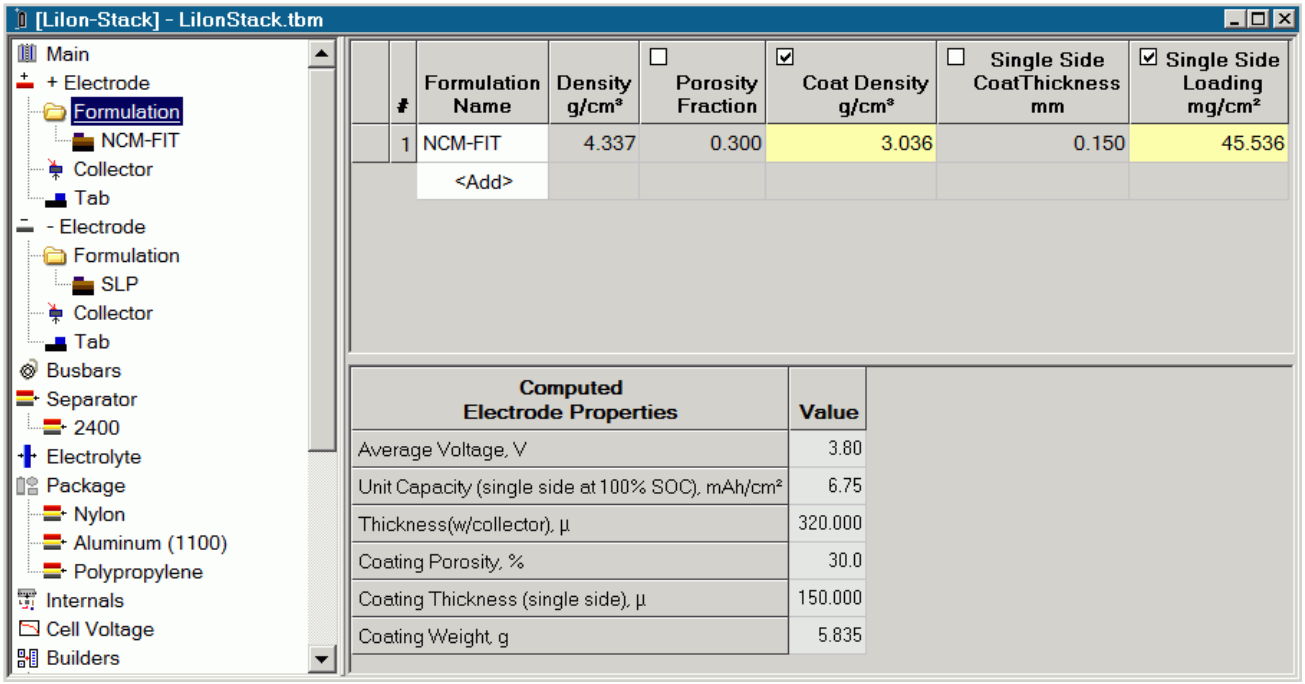
Relative Position of Electrodes

You can also define the relative position of electrodes in the *Offsets* tab of the **Builders** > **Simple** node. The *Positive wrt negative* setting defines the position of the positive electrode relative to the negative electrode. The relative position is defined in the X and Y offset values. These values specify the distances from the left edge of the negative electrode to the left edge of the positive electrode, and from the top edge of the negative electrode to the top edge of the positive electrode.

By default, all these values are set to 1 mm. You need to change these values if you have changed the width or height of the electrodes, or if you want to define wider or narrower separators.

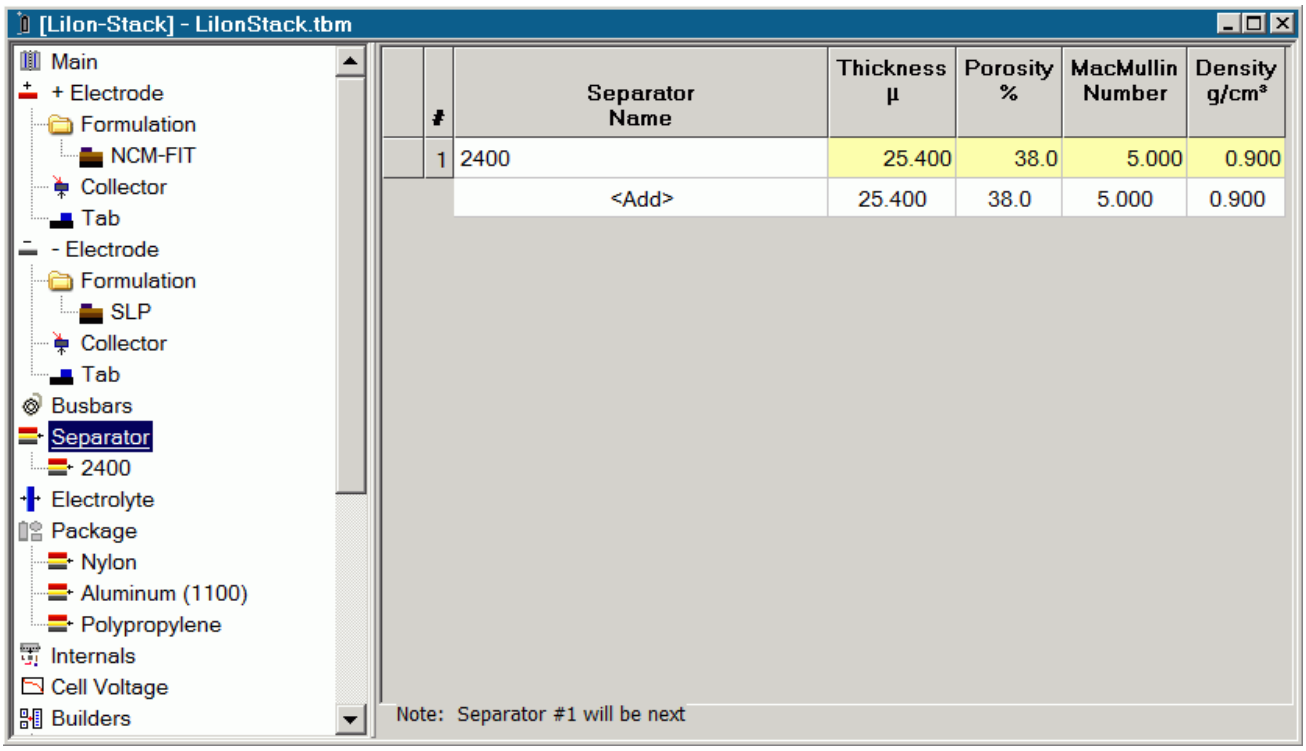
Electrode Thickness

To review the total thickness of each electrode, select the **+ Electrode** > **Formulation** node, or the **- Electrode** > **Formulation** node. The total thickness is displayed under the Computed Electrode Properties, next to *Thickness (w/collector)*. This value includes the thickness of the collector and coating for a single electrode.



Separator Thickness

You can define the thickness of the separators in the battery window. To set the separator thickness, select the **Separator** node. In the table, modify the appropriate values in the *Thickness* column.

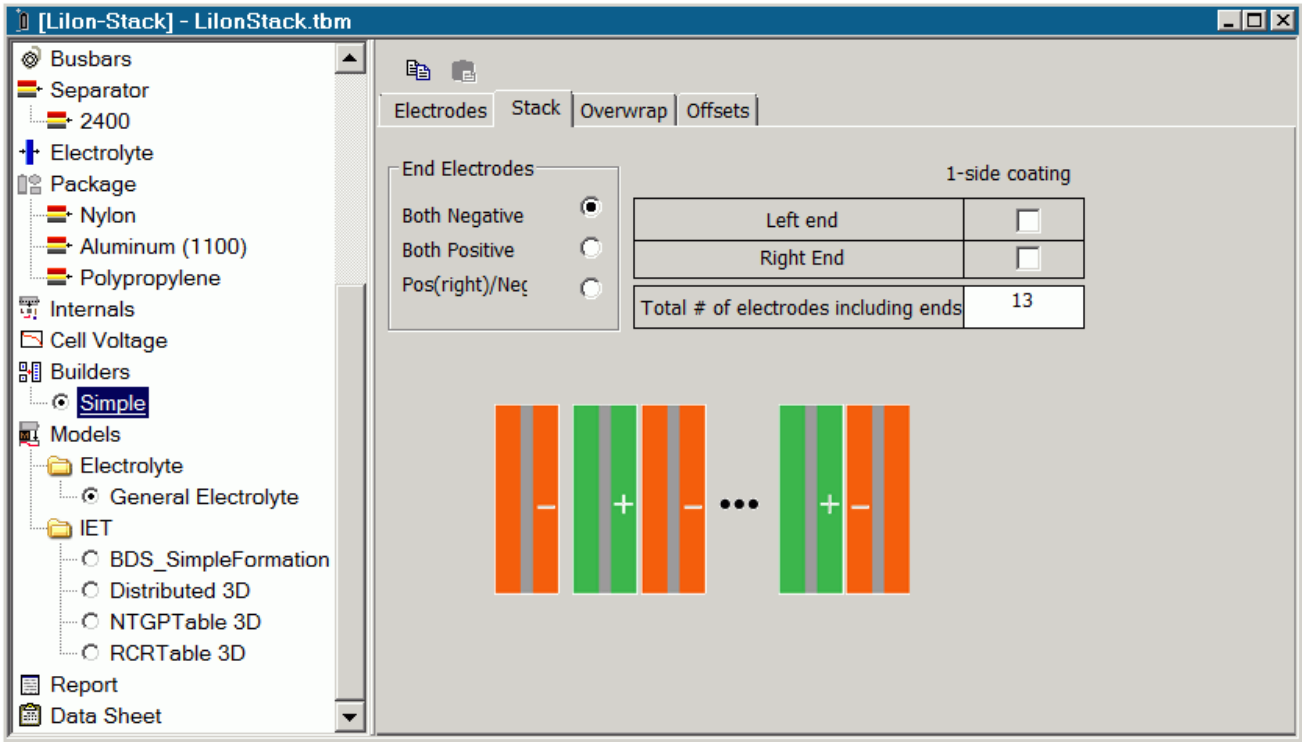


Number of Electrodes in Stack

You can define the number of electrode layers in the stack. In the battery window, select the **Builders** > **Simple** node, then select the *Stack* tab. The **Total # of electrodes including ends** box specifies the total number of electrodes in the cell.

The selected option for *End Electrodes* affects the values that you can specify. If you select **Both Negative** or **Both Positive**, the total number of electrodes must be odd. If you select **Pos(right)/Neg**, the total number of electrodes must be even.

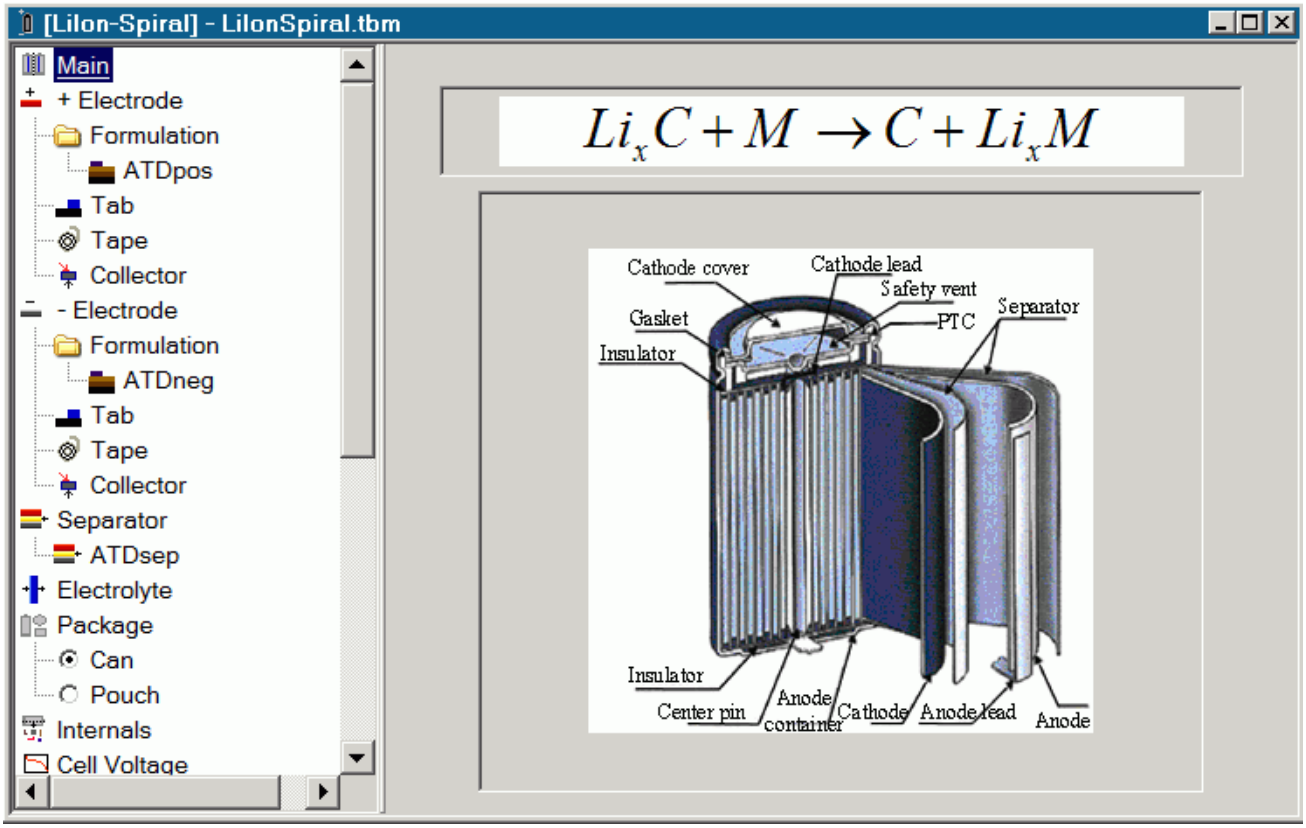
The number of separators in the cell is equal to the (total # electrodes + 1), that is, a separator layer between adjacent electrodes and an additional separator at either end of the stack.



Configuring a Spiral Cell

You can use Battery Design Studio® (BDS) to design spirally wound cylindrical and prismatic cells: the significant difference is the shape of the mandrel around which the cell is wound. In cylindrical cells, the mandrel is a cylinder; in prismatic cells, the mandrel is a geometric prism.

When you create a new spiral cell in BDS, the battery definition window is displayed in the workspace:



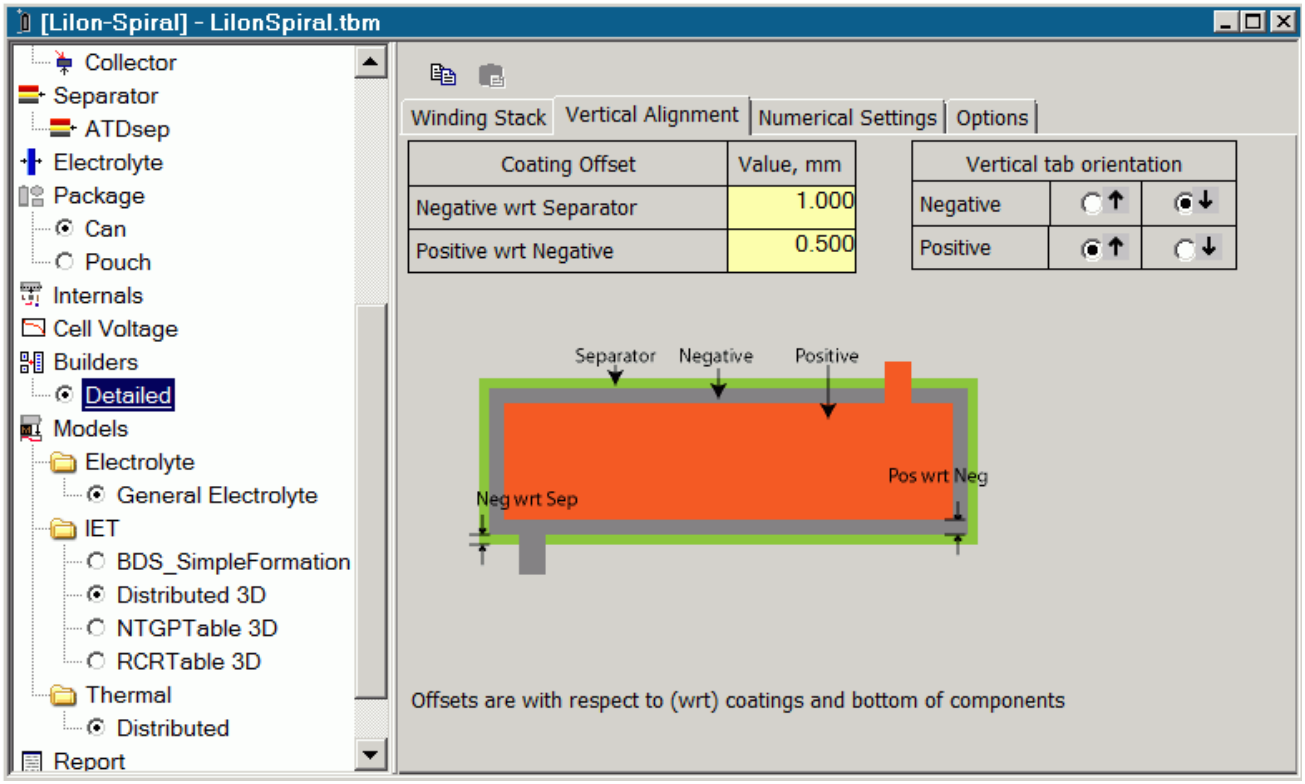
You can use the nodes on the left of the battery window to define characteristics and properties of the battery. You are recommended to start with the default dimensions. However, you can specify the tab configuration and the numerical settings that are used to map data to the thermal mesh in STAR-CCM+.

Tab Vertical Alignment

STAR-CCM+ supports same-face tabs for spiral cells, that is, battery cell configurations where both the positive and negative tabs are on the same side of the cell.

You can check the vertical alignment of the tabs in the **Builders** > **Detailed** node, in the **Vertical Alignment** tab. The example below shows the default cylindrical spiral cell, in which the tabs have opposite orientation.

If you want to modify the orientation of the tabs, make the appropriate changes in the **Vertical tab orientation** box.

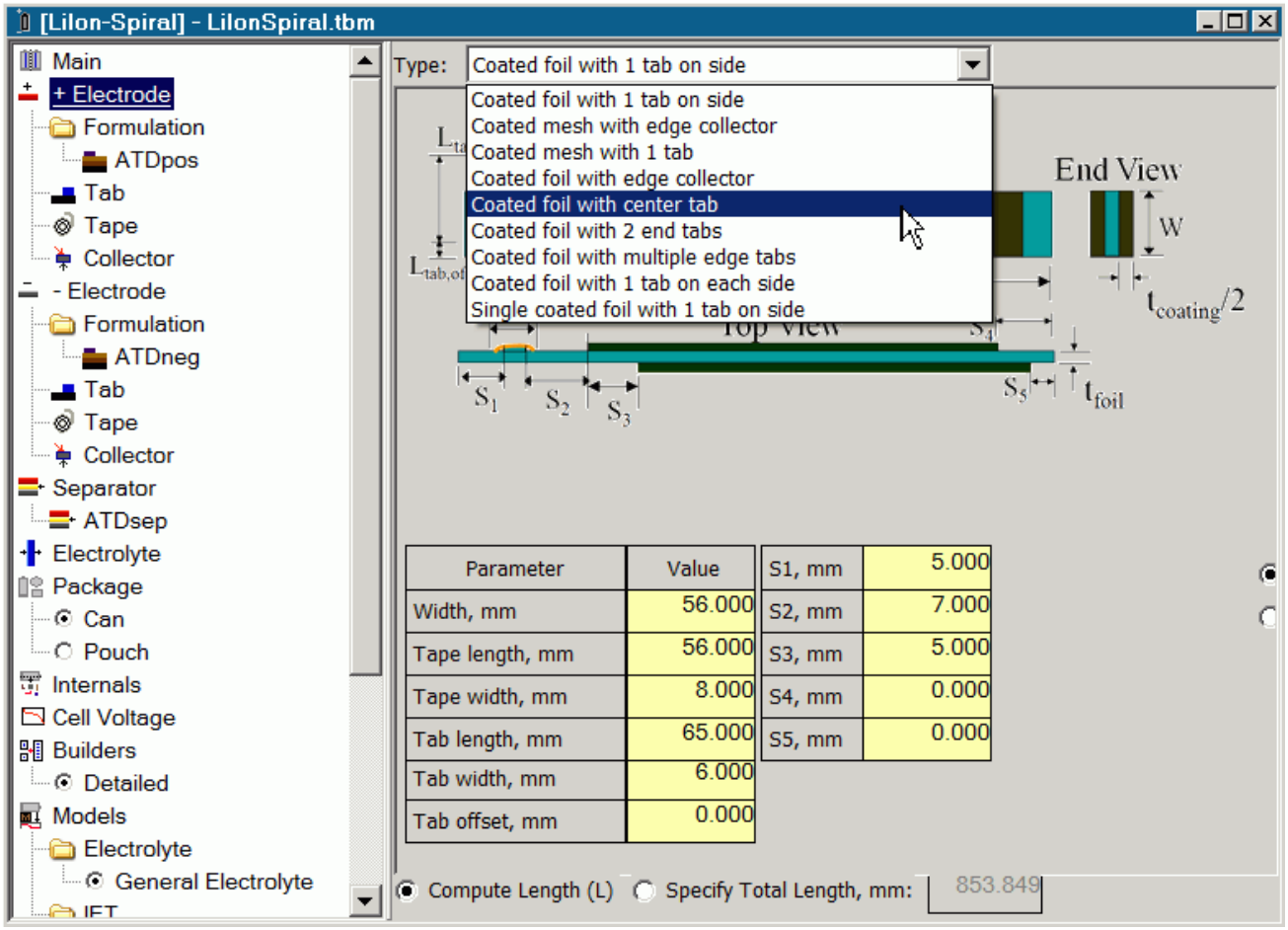


Tab Type

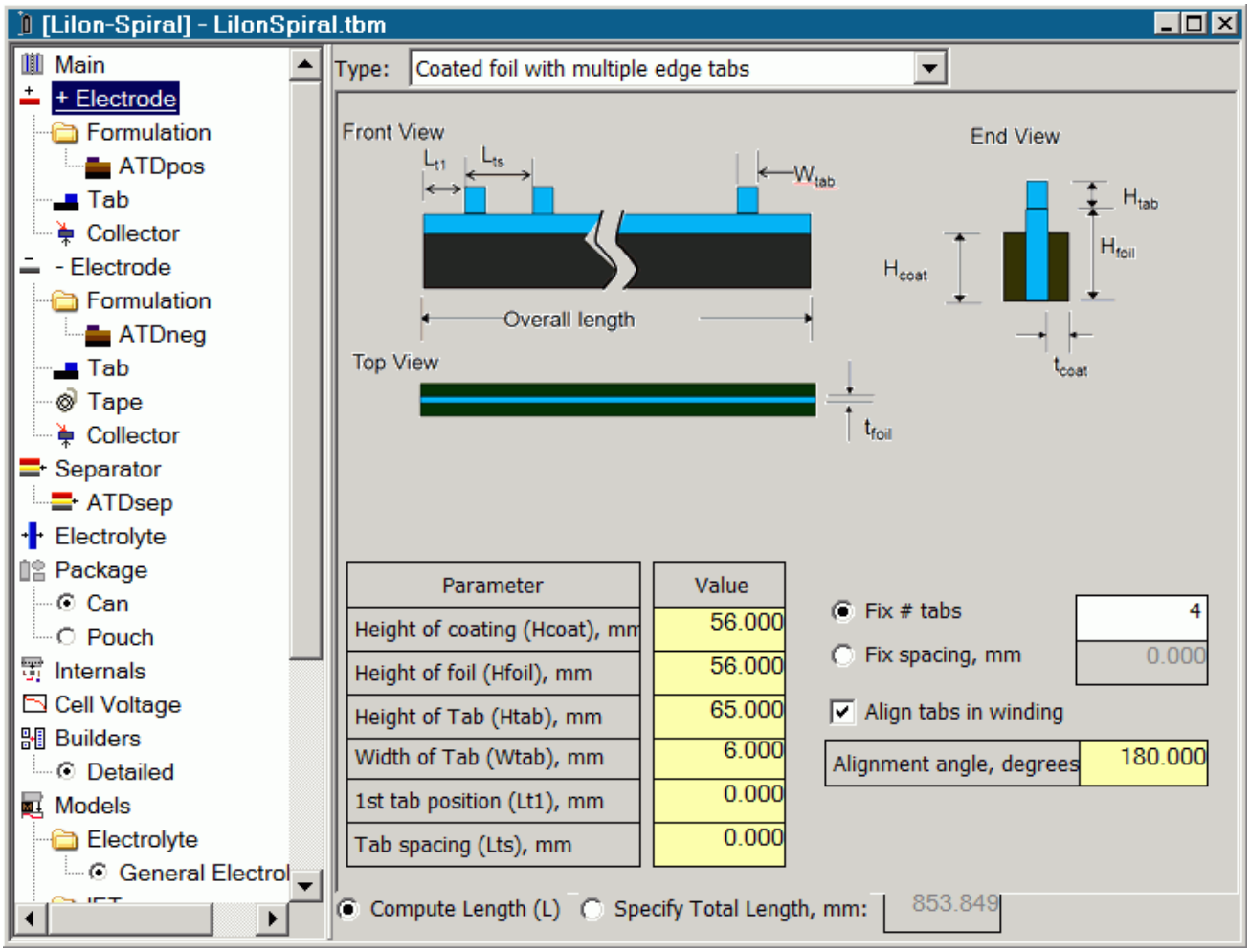
The default spiral cell has the following tab types defined:

- Positive electrode: **Coated foil with center tab**
- Negative electrode: **Coated foil with 1 tab on side**

However, you can specify a different type for either or both electrodes. To modify the tab type assigned to a particular electrode, select the appropriate node, **+Electrode** or **-Electrode**. From the **Type** drop-down menu, select the new tab type.



If you select a tab type that uses multiple tabs, you need to specify the number of tabs to use.



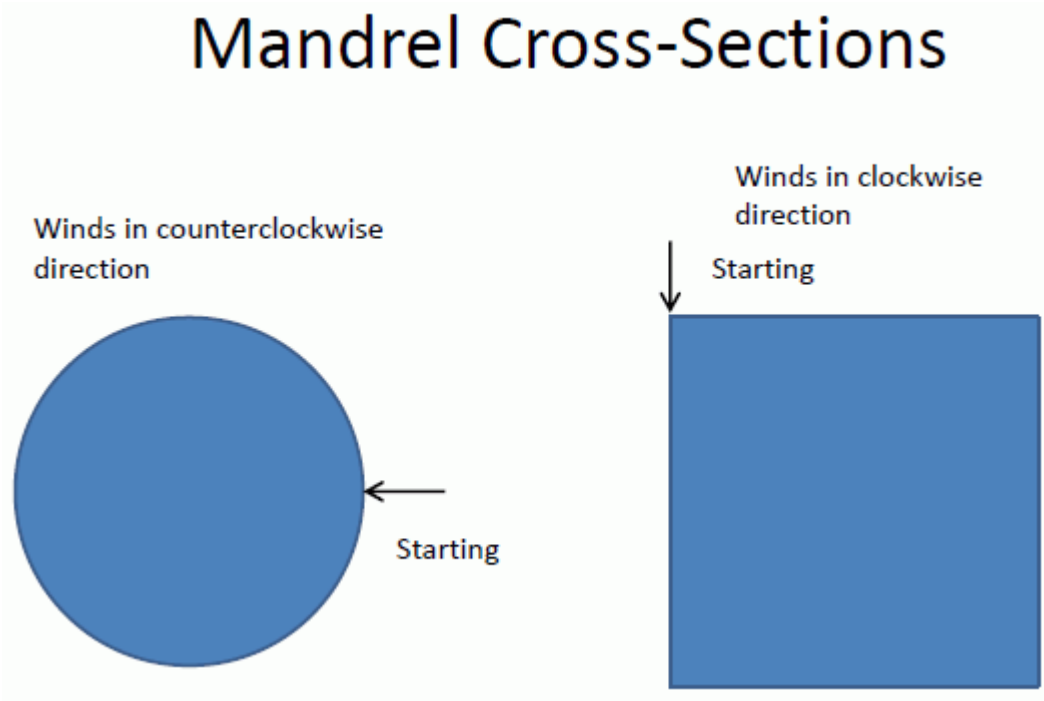
- To specify the number of tabs, in the **Fix # tabs** box, enter the appropriate number.

Tab Angular Alignment

You can specify the angular alignment of the tabs in the winding. For spiral cells, the tabs are used to orient the local coordinate systems (used for mapping) within the cell. If you do not specify the tab alignment, the local coordinate system origin and the Y-axis (axially along the cell) are defined, but there is no way to define the X-axis and Z-axis of the cell.

The alignment angle is with respect to the mandrel and the zero angle is the positive X-axis. The angle is measured in the anti-clockwise direction for cylindrical cells and the clockwise direction for prismatic cells, as viewed from the positive Y-direction.

In previous versions of BDS, the tab angle was measured with respect to the start of the electrode. The tab angle is now measured with respect to the mandrel, regardless of the starting point of the electrode.



- To align the tabs in the winding, activate the **Align tabs in winding** box and then, in the **Alignment angle, degrees** box, specify the angle at which the tabs begin.

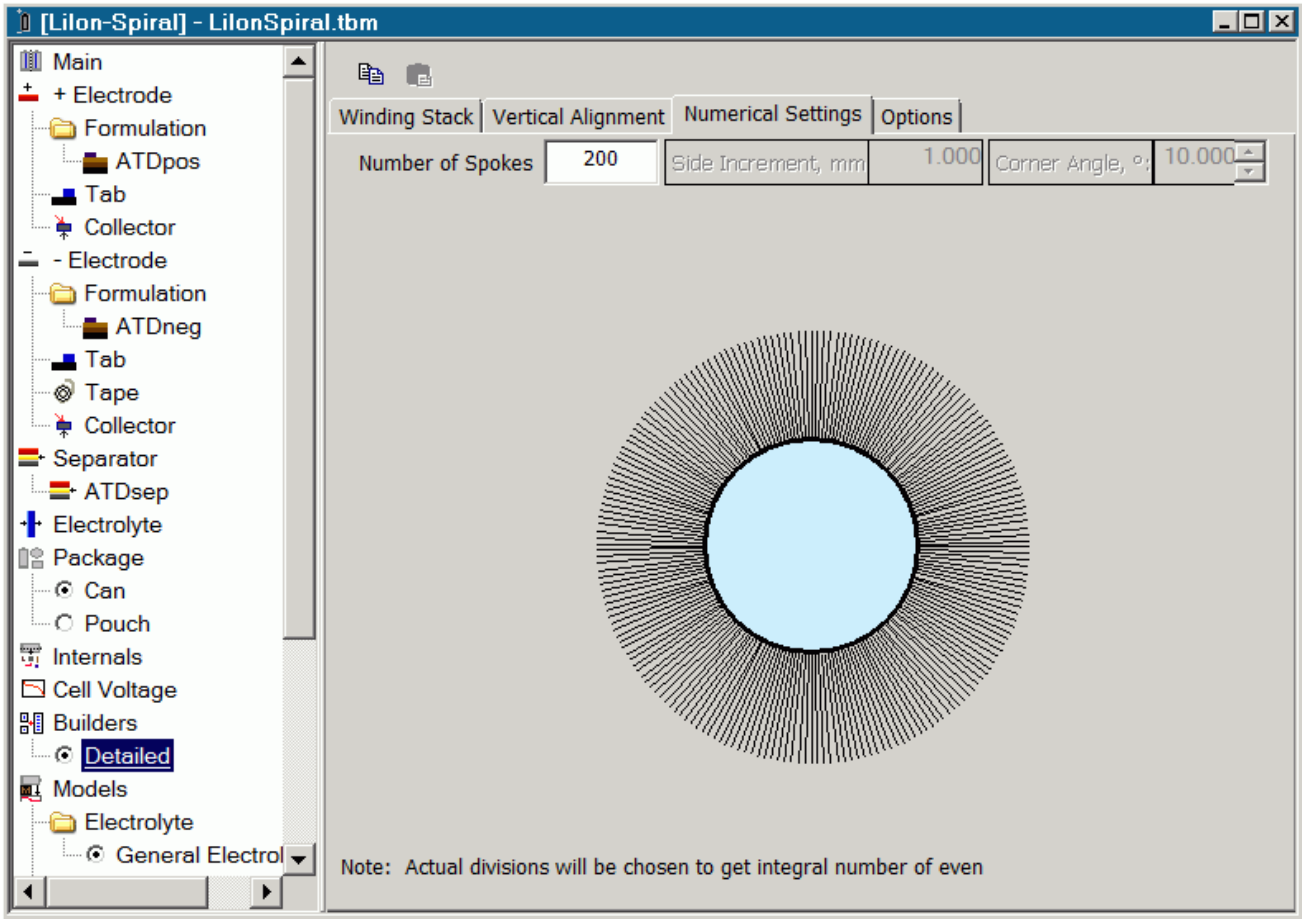
You can use the tab alignment setting to specify a particular angle (A_{sep}) between the tabs. To do this, specify the alignment angles of the tabs in the positive electrode (A_{pos}) and the negative electrode (A_{neg}) such that the difference A_{pos} - A_{neg} = A_{sep}.

Number of Spokes

Spokes are used in the discretization of the spiral cell and affect how the cell is wrapped. The number of spokes determine the location on the cell from where the data is mapped to the thermal mesh. You can reduce the number of spokes to speed up calculations or increase it to improve the accuracy of the model.

For a cylindrical cell, you can specify the number of spokes. The default value in the provided cylindrical cell is 50. The range of valid values is 50 - 1000.

To set the number of spokes for a cylindrical cell, select the **Builders** > **Detailed** node, and then select the **Numerical Settings** tab.

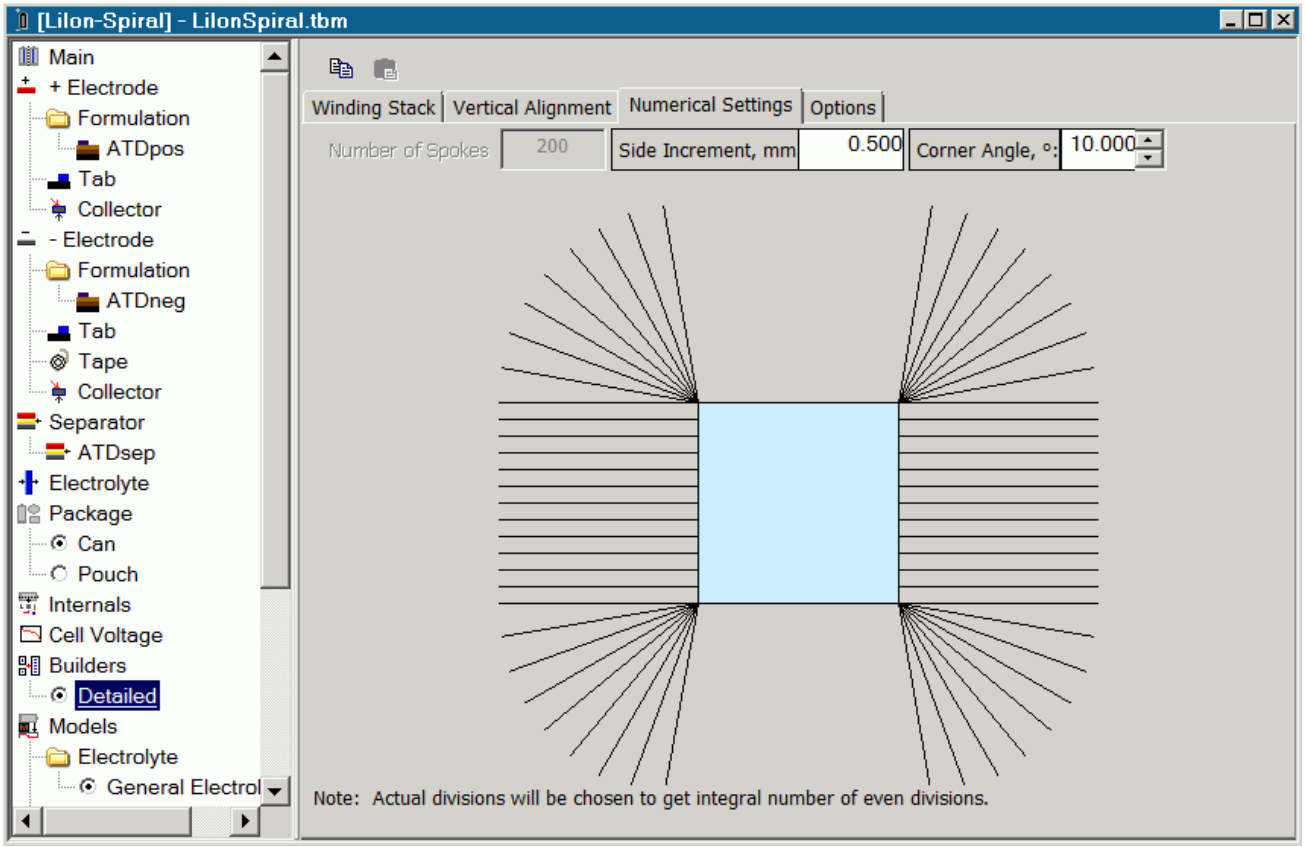


- In the **Number of Spokes** box, specify the appropriate number.

If necessary, you can modify the number of spokes value in STAR-CCM+, but the same range of valid values applies.

For a prismatic cell, you specify the number of spokes in relation to the size and shape of the mandrel.

To set the number of spokes in a prismatic cell, select the **Builders** > **Detailed** node, and then select the **Numerical Settings** tab.



- In the **Side Increment** box, specify the appropriate value.

  The number of spokes on each side of the mandrel is the side length divided by the increment. You can view and modify the mandrel width and thickness in the *Winding Stack* tab.

- In the **Corner Angle** box, specify the appropriate angle.

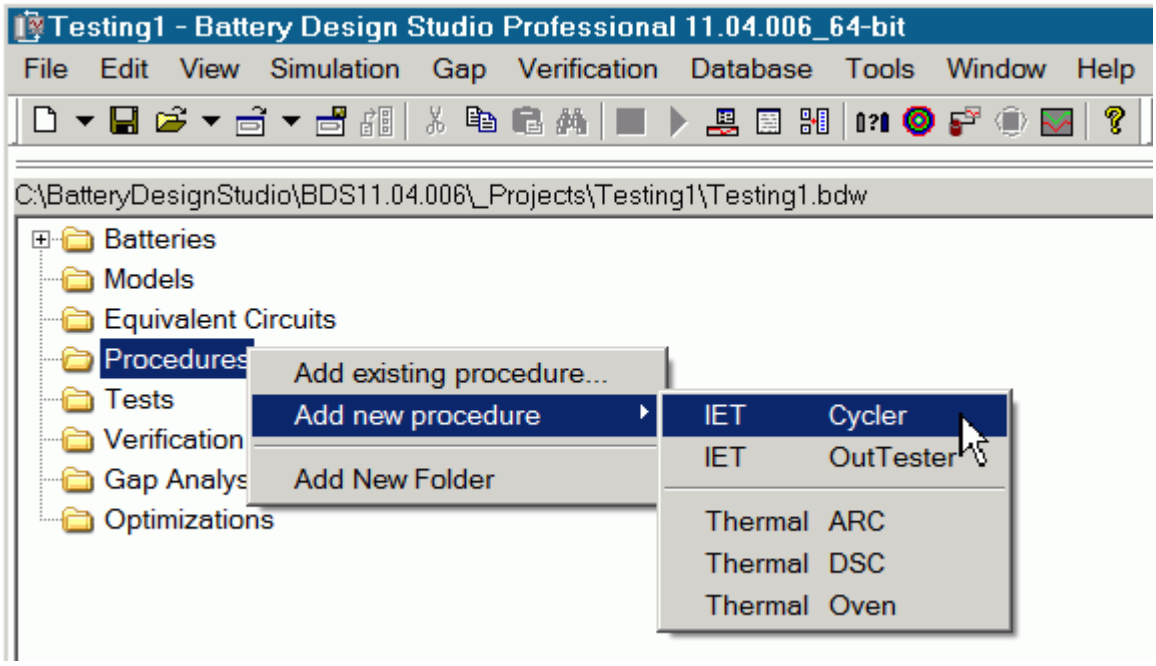
  The number of spokes on each corner is 90 divided by the specified angle. The minimum value is 10 degrees (9 divisions) and the maximum is 22.5 degrees (4 divisions). If you specify an angle that does not divide evenly into 90, BDS automatically uses the nearest integral number of spokes.

Preparing Program (.prg) Files

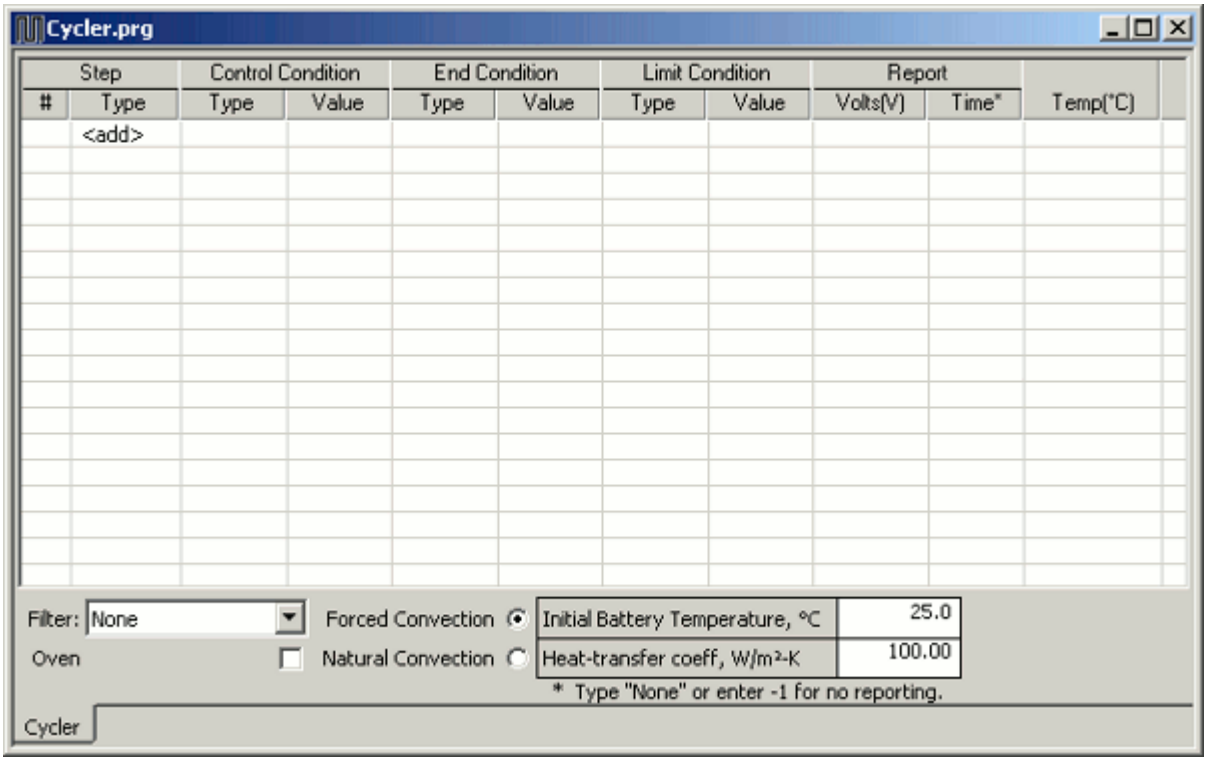
In Battery Design Studio® (BDS), procedures for cycling batteries are stored in program files (`.prg`). In a program file, you can specify periods when the battery is charging, discharging, and resting.

The program file is one way to apply an electrical load to the battery in STAR-CCM+.

To create a procedure in BDS, right-click the **Procedures** node and select **Add new procedure** > **IET Cycler**.



A new cycling procedure is added under the **Procedures** node, and a program definition window is displayed in the workspace.



When the procedure is created, a `.prg` file is automatically created in the workspace directory. You can use the table in the program definition window to set up the cycle.

Selecting a Step Type

In the *Step* column, click the **<add>** button and select from the following types of step:

- **Advance**

  Increments the cycle counter.
- **Rest**

  Sets the applied current to zero.
- **Charge**

  Sets the sign of the applied current to negative (the convention for BDS charging).
- **Discharge**

  Sets the sign of the applied current to positive (the convention for BDS discharging).
- **End**

  Ends the program.

Setting Control Conditions

After selecting a step type, you can specify the control, end, and limit conditions for the step. The *Control Condition* identifies the variable to control during the step, such as current, voltage, power, or resistance. The *End Condition* identifies the value that, when reached, terminates the step.

Setting End Conditions

STAR-CCM+ fully supports **Time** end conditions. **Voltage** end conditions are partially supported.

The **Time** end condition has units of seconds. To specify an end condition of 1 hour, for example, select **Time** in the Type column and enter 3600 in the Value column.

The **Voltage** end condition has units of volts. To specify an end condition of 10 volts, for example, select **Voltage** in the Type column and enter 10 in the Value column.

If the step type is a discharge load, the voltage end value is interpreted as a minimum voltage. The simulation ends with a message indicating that the minimum cell voltage limit has been reached.

If the step type is a charging load, the voltage end value is interpreted as a maximum voltage. The simulation ends with a message indicating that the maximum cell voltage limit has been reached.

When a voltage end condition is specified, the battery simulation continues with the specified load type until the end of the STAR-CCM+ time steps, or until the voltage end condition is met, whichever comes first. Any remaining lines of input in the program file are not used.

Setting Limit Conditions

STAR-CCM+ uses **Voltage** limit conditions to specify a warning message.

The *Limit Condition* is an optional parameter that allows you to specify maximum or minimum values. The most common use of a limit condition in BDS is to limit the voltage of the battery during the cycle. Within STAR-CCM+ a battery can consist of multiple modules, with each module containing more than one battery cell. In STAR-CCM+, the current limit condition option uses the maximum and minimum values to print a warning message when any battery cell exceeds the specified limits. The circuit solver does not switch to a new load type as is the case in single cell battery simulation in BDS.

If the step type is a discharge load, the limit value is interpreted as a minimum voltage. The circuit solver prints a warning message for every time step in which one or more battery cells reach the lower limit voltage.

If the step type is a charging load, the limit value is interpreted as a maximum voltage. The circuit solver prints a warning message for every time step in which one or more battery cells reach the limit voltage.

The simulation does not stop when a limit condition is reached. You can specify a limit condition on each line of input in the program file that is used to define a load.

Setting Up Reports

In the *Report* column of the program file, you can specify the frequency at which data is recorded. You can specify the frequency by time and/or voltage change.

Setting Temperatures

In the *Temp* column, you can specify the temperature at which to carry out the step. If this column is left blank, BDS uses the *Initial Battery Temperature*. This value is defined at the bottom of the program file window.

Working with a Battery Cell

A battery cell is a single unit that is used to store energy. Battery cells are defined in Battery Design Studio® (BDS), and stored in a .tbm file or in an encrypted .ebm file.

Battery cells are used in STAR-CCM+ to create battery modules, which contain a number of battery cells that are connected in series and/or parallel. STAR-CCM+ supports stacked cells and spirally wound cylindrical and prismatic cells.

Contents:

Creating a Battery Cell
Configuring a Battery Cell
Setting the Resolution of the Electrical Mesh
Viewing the Battery Cell Properties
Setting Cell Data Overrides

Creating a Battery Cell

To use a battery cell in STAR-CCM+, you import the `.tbm` or `.ebm` file into the simulation.

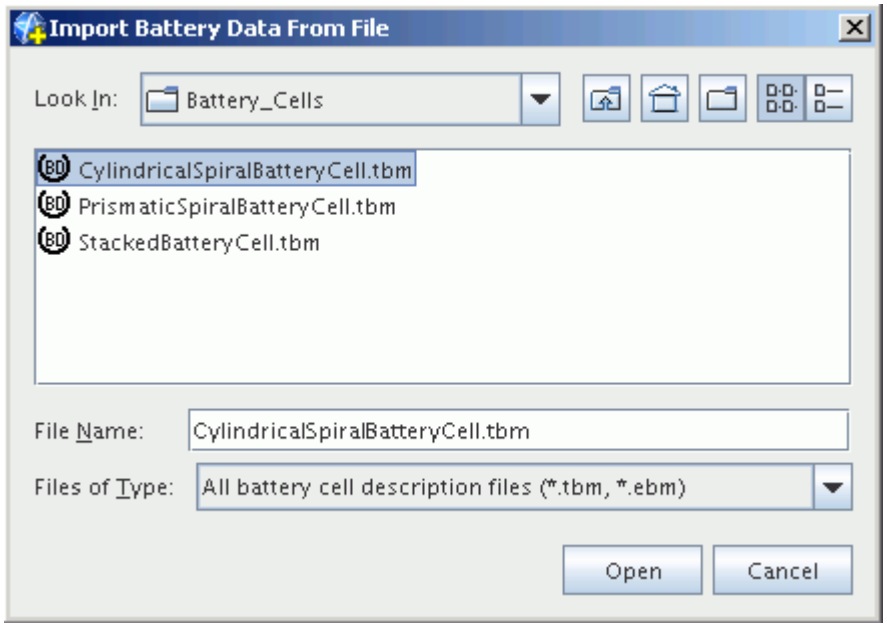
By default, all of the battery parts that are specified in the file are imported. However, if you want to use your own geometry components for some parts, you can choose to leave out the corresponding parts from the imported file.

Creating a Battery Cell

To create a battery cell:

1. Right-click the **Batteries** > **Battery Cells** node and then select **Create from Tbm...**

2. In the *Import Battery Data From File* dialog, select the appropriate `.tbm` or `.ebm` file and then click **Open**.



3. In the *Import Battery Options* dialog, select the battery parts that you want to import.

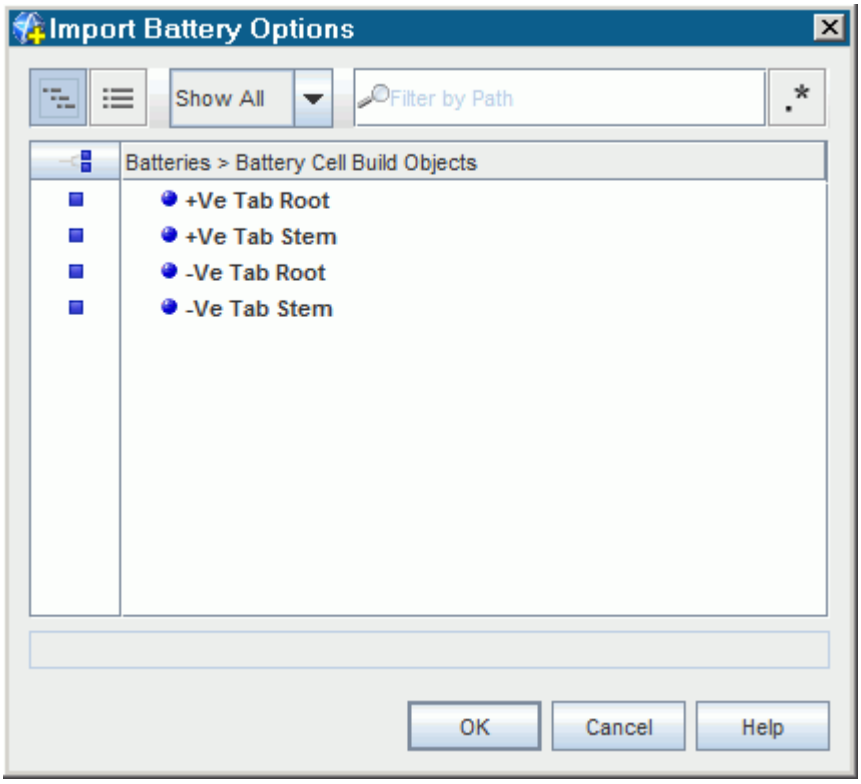
   By default, all parts of the battery cell are selected. However, you can exclude some parts and replace them with your own geometry.

   The active material (the stack for stacked cells and the jellyroll for spiral cells) is always imported.
4. Click **OK**.

STAR-CCM+ automatically recognizes the cell models and updates the appropriate properties of the battery cell, and reads in the mesh settings that are specified in the battery model.

Selecting Optional Parts for a Stacked Cell

For stacked cells, the optional parts are as shown below:

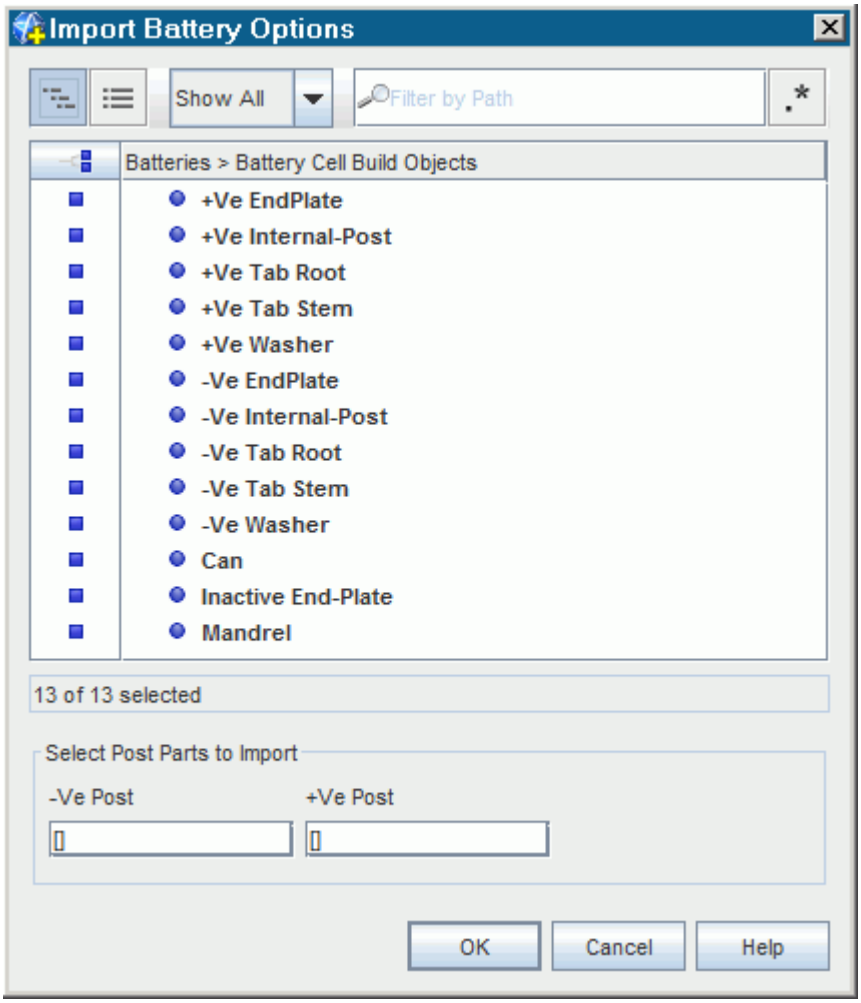


5. If you want to exclude some parts from the imported file, deselect the appropriate parts.

   The active material (the stack) does not appear on the list of optional parts; it is always imported.

Selecting Optional Parts for a Spiral Cell

For spiral cells, the optional parts are as shown below:



6. If you want to exclude some parts from the imported file, deselect the appropriate parts.

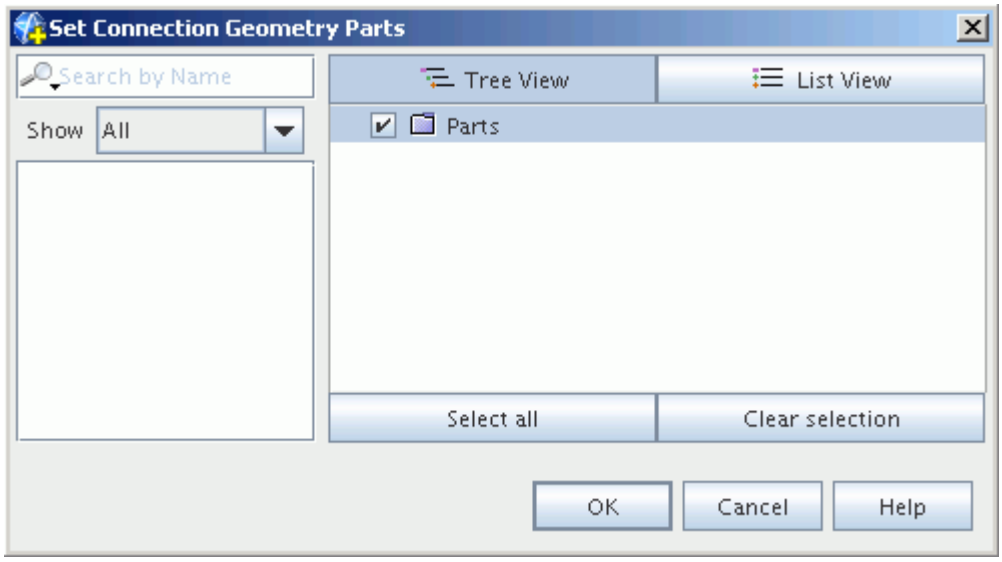
   The active material (the jellyroll) does not appear on the list of optional parts; it is always imported.

Adding External Posts to a Spiral Cell

If you are importing a cell that has same face tabs, you have the option to add external posts to the battery model as part of the import process. The external posts may be complex CAD entities that you have created outside BDS. You need to import the appropriate parts into STAR-CCM+ before you can add them to the battery model.

7. Under *Select Post Parts to Import*, click the appropriate option: **-ve Post** or **+ve Post**.

   The *Set Connection Geometry Parts* dialog appears:



8. In the *Set Connection Geometry Parts* dialog, select the parts that you want to add, and then click **OK**.

   The selected parts are displayed under the **Spiral External Posts** node, under the new **Cylindrical Cell** node.

See Also:

Configuring a Battery Cell

Setting the Resolution of the Electrical Mesh

Configuring a Battery Cell

When the battery cell is imported into the simulation, STAR-CCM+ reads the dimensions of the cell from the `.tbm` or `.ebm` file and generates an idealized battery cell 3D-CAD model. This model is displayed in the *Graphics* window in a *3D-CAD View* scene.

The corresponding **Battery Cell** node, **Cylindrical Cell** node, or **Prismatic Cell** node is added to the tree under the **Batteries** > **Battery Cells** node.

The new **Battery Cell** node, **Cylindrical Cell** node, or **Prismatic Cell** node has properties and a pop-up menu, and is also subject to common menu items.

Battery Cell Pop-Up Menu

The following options are available when you right-click on a battery cell node.

Reload...	Reloads the battery cell data from a `.tbm` or `.ebm` file. This option allows you to update the battery cell if you change the .tbm or .ebm file.

Export...	Exports the battery cell data to a `.tbm` or `.ebm` file. This option allows you to recreate a file from the battery cell information that is stored within the simulation. You cannot choose the export format: it is the same as the format of the original file that you imported.

Stacked Cell Properties

The following properties are available for stacked battery cells.

Electrical Mesh Dimensions	Defines the user-specified electrical mesh resolution.
TBM File	Displays the path to the imported `.tbm` file or `.ebm` file.
Unit Cell Model	Identifies the battery model that is defined in the `.tbm` file for the imported battery cell. For empirical models, you can override the cell capacity or state of charge settings. You cannot override any settings for a battery cell that uses a physical model. For more information, see Setting Cell Data Overrides.

Spiral Cell Properties

The following properties are available for spiral battery cells.

TBM File	Displays the path to the imported `.tbm` file or `.ebm` file.
Unit Cell Model	Identifies the battery model that is defined in the `.tbm` file for the imported battery cell. For empirical models, you can override the cell capacity or state of charge settings. You cannot override any settings for a battery cell that uses a physical model. For more information, see Setting Cell Data Overrides.
Specified Electrical Mesh Dimensions	Defines the user-specified electrical mesh resolution. The default is the mesh that is specified in the `.tbm` file. You can modify the mesh dimensions if necessary. For information on modifying the mesh dimensions, see Setting the Resolution of the Electrical Mesh.
Actual Electrical Mesh Dimensions	Defines the electrical mesh resolution that is computed from the specified layout of a spiral cell and the specified electrical mesh dimensions. STAR-CCM+ uses this resolution. This property is read only.
Number of Spokes	Defines the number of spokes. Spokes are a component of the electrical mesh. You can reduce the number of spokes to speed up calculations or increase it to improve the accuracy of the model.

	For a cylindrical cell, you can specify the number of spokes. The range of valid values is 50 - 1000. You can also change this value in the Specified Electrical Mesh Dimensions property.
Positive Post Parts	If you have added external posts to the battery model, this property specifies the positive post parts. You can modify the selection as necessary. If you have not added any external posts, this property is empty. If you want to add external post parts, click the **...** button and then, in the selection dialog, select the appropriate parts.
Negative Post Parts	If you have added external posts to the battery model, this property specifies the negative post parts. You can modify the selection as necessary. If you have not added any external posts, this property is empty. If you want to add external post parts, click the **...** button and then, in the selection dialog, select the appropriate parts.

See Also:

Creating a Battery Cell

Setting the Resolution of the Electrical Mesh

The electrical mesh defines cells on the battery stack or jellyroll strip. Each cell in the electrical mesh represents an ecell in the electrical calculation. Although you must define the electrical mesh dimensions, it is created internally and cannot be visualized.

The initial resolution for the electrical mesh depends on the battery model that is selected in BDS, and is read into STAR-CCM+ during the import process. The electrical mesh is used during the simulation only for the electrical calculations; a thermal mesh is used for the energy calculations.

You can change the resolution of the electrical mesh to suit the requirements of your simulation.

Under the **Battery Cells** node, select the appropriate battery cell. The electrical mesh dimensions are displayed in the *Properties* window.

For a stacked plate cell, you can specify the number of mesh cells in the X, Y, and Z directions.

For a cylindrical cell, you can specify the number of mesh nodes in the X and Y directions. You can also specify the number of spokes.

Within BDS, the specified electrical mesh dimensions are used to compute the resolution that STAR-CCM+ uses for its electrical solver. The actual dimensions are derived from the specified resolution and the minimum number that is feasible based on the layout of the jellyroll and tabs.

Typically, the electrical mesh is less dense than the thermal mesh or the surrounding fluid mesh. For many applications, it is recommended that you start by using a coarse electrical mesh and increase the mesh density only if necessary. For a stacked cell, a coarse electrical mesh is 10x10x1 or less. The solution time varies linearly with the addition of battery cells to a battery module.

For a stacked cell, you set the value of the electrical mesh resolution in the Z-direction, according to the thickness of the battery cell. For thin cells, there is no advantage in using more than one ecell as there is not normally a significant temperature gradient in the normal direction.

For a spiral cell, the minimum mesh resolution is determined automatically in BDS based on the layout of the cell. If the specified resolution is lower than this minimum, BDS increases it accordingly. For a stacked cell, the minimum mesh resolution is 1x1x1.

The thermal mesh for the battery cell is defined separately in STAR-CCM+ in a mesh continuum. Mesh settings for other parts of the battery, such as tabs and connectors, are also defined in a mesh continuum. These settings are covered in generating battery mesh.

See Also:

Generating a Battery Mesh

What Is a Thermal Solver?

What Is an Ecell?

Viewing the Battery Cell Properties

The **Cell Properties** node under each battery cell node contains the properties that are imported from the `.tbm` file. The properties are organized in a tree structure that is similar to the cell design tree in BDS.

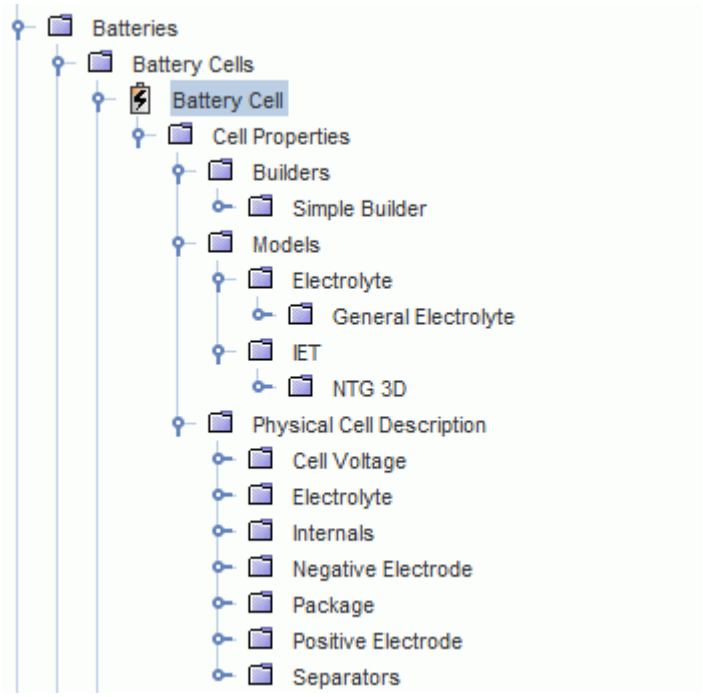
For more information on the BDS cell design tree and the properties that you can specify when you design a battery cell, see the Battery Design Studio User Guide.

Most of these properties are read-only and you cannot change them in STAR-CCM+. The only exceptions are the cell capacity and state of charge settings, for which you can set override values. For more information, see Setting Cell Data Overrides.

If you want an overview of the properties of a particular battery cell, you can view the cell summary report. For more information, see Viewing a Cell Summary Report.

Stacked Cell Properties

The properties tree for a stacked cell is shown below:



The stacked cell properties tree contains the following major nodes:

Builders	**Simple Builder**	Specifies how the components are placed together to make the completed cell. The simple builder specifies the orientation of the plates within the cell, the size and configuration of the stack, and any protective overwrap that is given to the electrode stack. The simple builder also specifies the offsets that the separator introduces between the electrodes and the pouch.
Models	**Electrolyte**	The General Electrolyte model is always specified. This model describes the behavior of the electrolyte within the cell.
	IET	One IET model is specified per cell. This model describes the electronic and thermal behavior of the cell. STAR-CCM+ supports the NTG model, the RCR model, and the DISTNP model. Some other models that are available in BDS are not supported.

		For more information on the IET models that are available in Battery Design Studio, see the Battery Design Studio User Guide.
Physical Cell Description	**Cell Voltage**	Specifies the current state of the cell, how the lithium is distributed between the two electrodes, and the allowable voltage range of the cell.
	Electrolyte	Specifies the formulation of the chemical mix that makes up the electrolyte. An electrolyte is a combination of multiple salts and solvents, and each ingredient is specified separately. For more information, see Electrolyte Properties, below.
	Internals	Specifies the fillers that are inserted in cell packaging alongside the active components. These items are used to provide shape and stability to the components within the packaging and are effectively inert.
	Negative Electrode	Specifies the physical and thermal properties of the electrode, the formulation (the chemical substance that makes up the electrode), the tabs, and the collector that connects the electrode to the tab. The busbars are also specified. The two busbars span the cell, joining tabs together. For more information, see Electrode Properties, below.
	Positive Electrode	
	Package	Specifies the packaging of the cell. The packaging affects the thermal performance of the cell.
	Separators	Specifies the separator that is used to separate the various chemically reactive components within the cell to limit reaction rates.

Electrolyte Properties

STAR-CCM+ supports the CAEBAT electrolyte property sets that are available in Battery Design Studio. These property sets are based on a model that was developed by K. Gering and were generated as part of the Computer-Aided Engineering of Batteries (CAEBAT) program.

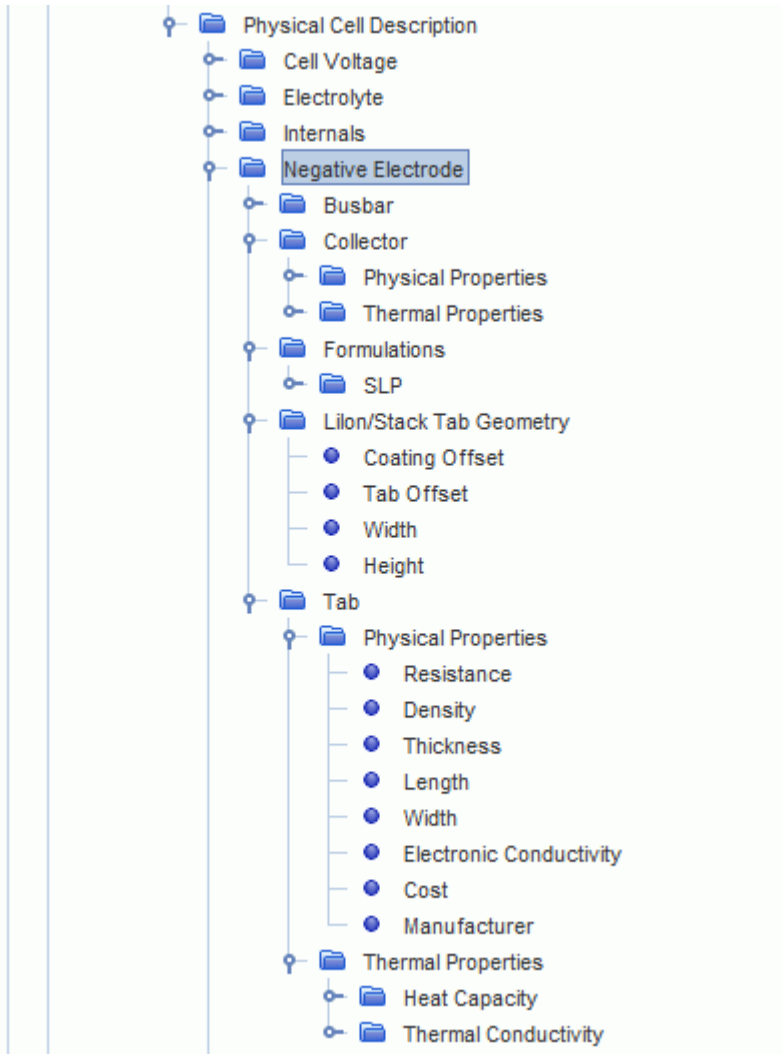
For more information, see Abstract #593, 224th Meeting of the Electrochemical Society:

https://ecs.confex.com/ecs/224/webprogram/Abstract/Paper20427/B4-0593.pdf

For more information on the electrolyte property sets that are available in Battery Design Studio, see the section *Using Predefined Electrolyte Property Sets* in the *Battery Design Studio User Guide*.

Electrode Properties

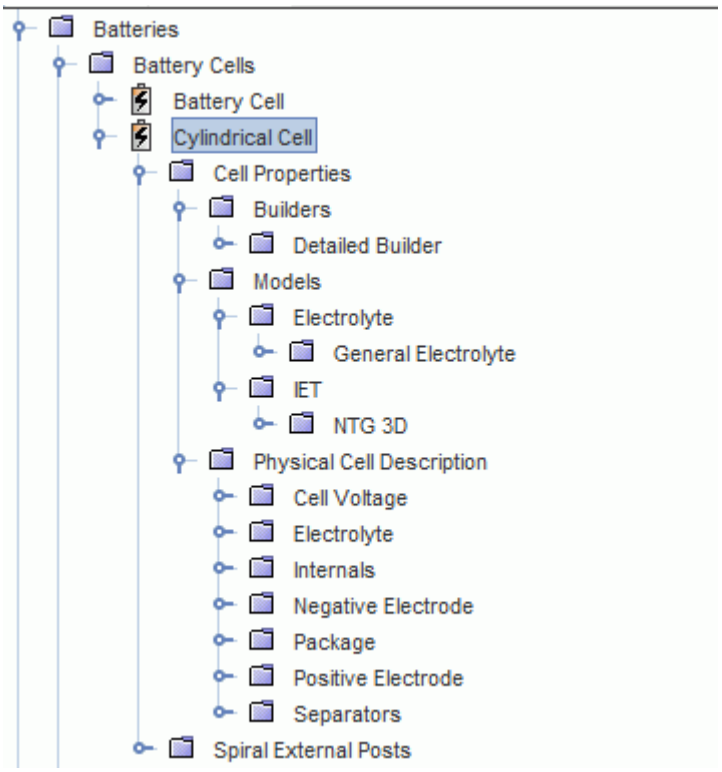
The example below shows the expanded Negative Electrode node:



For more information on the electrode properties, see the *Battery Design Studio User Guide*.

Cylindrical Cell Properties

The properties tree for a cylindrical cell is shown below:



The cylindrical cell properties are similar to the stacked cell properties. The cylindrical cell properties tree contains the following major nodes:

Builders	**Detailed Builder**	Specifies how the components are placed together to make the completed cell. The detailed builder specifies the orientation of the cell tabs, the winding and vertical alignment of the cell.
Models	**Electrolyte**	See the electrolyte description in the stacked cell properties tree.
	IET	See the IET description in the stacked cell properties tree.
Physical Cell Description	**Cell Voltage**	See the cell voltage description in the stacked cell properties tree.
	Electrolyte	See the electrolyte description in the stacked cell properties tree.

	Internals	See the internals description in the stacked cell properties tree.
	Negative Electrode	Specifies the physical and thermal properties of the electrode, the formulation (the chemical substance that makes up the electrode), the tabs, and the collector that connects the electrode to the tab. The tape that connects the tab to the electrode is also specified. For more information, see Electrode Properties.
	Positive Electrode	
	Package	See the package description in the stacked cell properties tree.
	Separators	See the separators description in the stacked cell properties tree.

Contents:

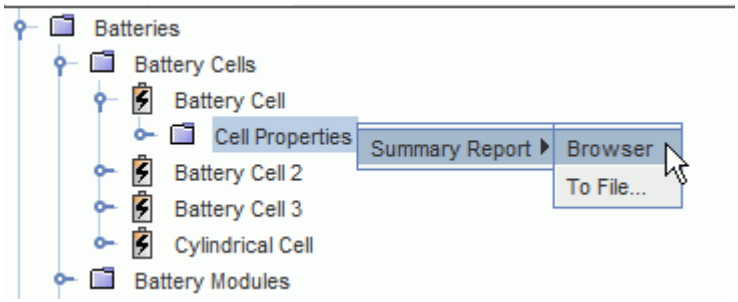
Viewing a Cell Summary Report

Viewing a Cell Summary Report

A cell summary report provides an overview of the battery cell properties. This information is taken directly from the <REPORT> section of the `.tbm` file.

To view a cell summary report for a particular battery cell:

1. Right-click **Batteries** > **Battery Cells** > **[Battery Cell]** > **Cell Properties**.



2. In the pop-up menu, click **Summary Report** and then click one of the following:
   - **Browser**: Displays the summary report in your default browser.
   - **To File**: Saves the summary report as an HTML file in a specified location.

Cell Summary Report

An example of a cell summary report is shown below:

Cell Report: BatteryCellProperties

1 **Cell Properties**	State of Charge, %	100
	Voltage, V	3.65879
	Capacity, Ahr	10.0331
	Energy, Whr	36.7092
	Energy Density, Whr/kg	248.748
	Energy Density, Whr/liter	572.008
	Weight, g	147.576
	Volume, cm^3	64.176
	Material Cost, $	4.95685
	Active Area, m^2	0.153768
	Unit Capacity, mAh/cm2	6.56695
	C/A Ratio, mAh/mAh	1.50999
	Electrolyte Mass, g	18.7778
	Electrolyte Volume, cm^3	15.9594
	Separator Area, m^2	0.16524
	Heat Capacity@25C, J/g-K	0.871747
	Cell Dimensions Length, mm	190
	Cell Dimensions Breadth, mm	100
	Cell Dimensions Thickness, mm	4.6156
2 **Active Material Properties**	Energy Density, Whr/kg	288.99
	Energy Density, Whr/liter	609.261
	Weight, g	127.026
	Volume, cm^3	60.252
	Height, mm	153
	Width, mm	90
	Thickness, mm	4.3756
3 **Electrode Properties**		
+-1 **Negative Electrode**	Average Voltage, V	0.141296
	Stoichiometry at formation	0.9
	Unit Capacity, mAh/cm^2	6.56695
	Thickness (w/collector), um	300

Setting Cell Data Overrides

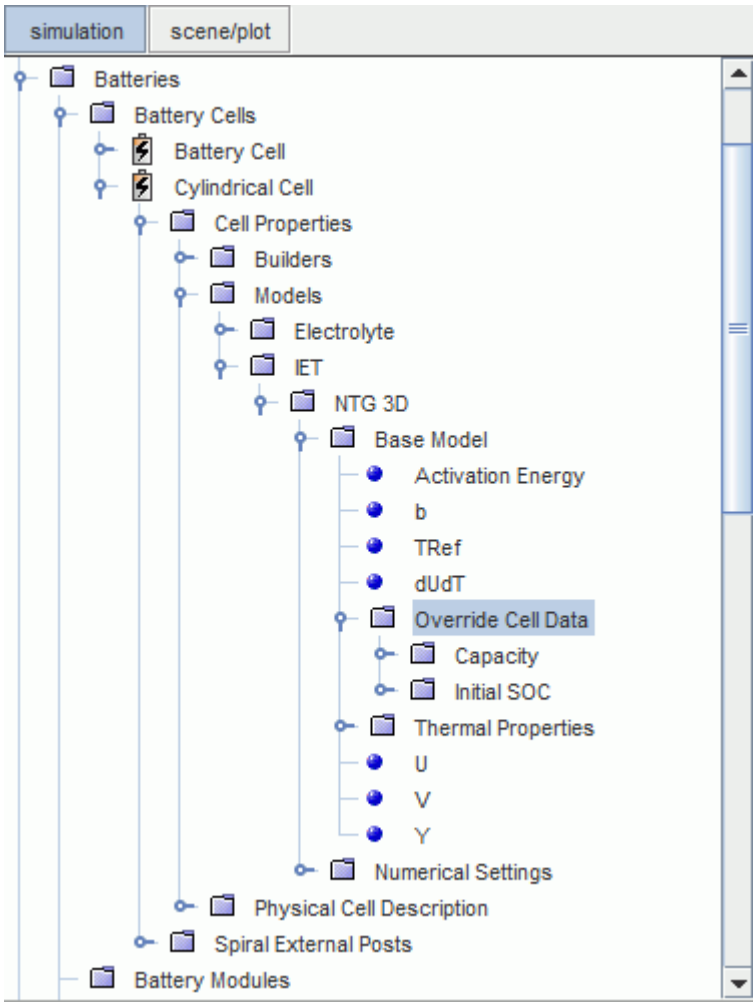
You can override the cell capacity or state of charge settings for a battery cell that uses an empirical model (such as the NTG model or the RCR model). However, you cannot override any settings for a battery cell that uses a physical model (such as the DISTNP model).

If you change the cell capacity or state of charge settings during a simulation, the new values do not take effect immediately. To apply the new values to the simulation, you must clear the solution.

When you load a simulation into STAR-CCM+, the restored simulation uses the cell capacity or state of charge settings that were in effect when the simulation was saved. However, if you clear the solution, any override settings are lost: the simulation reads the cell capacity and state of charge settings from the original `.tbm` file. If you want to use the same values again, you must re-enter the appropriate cell capacity or state of charge override settings.

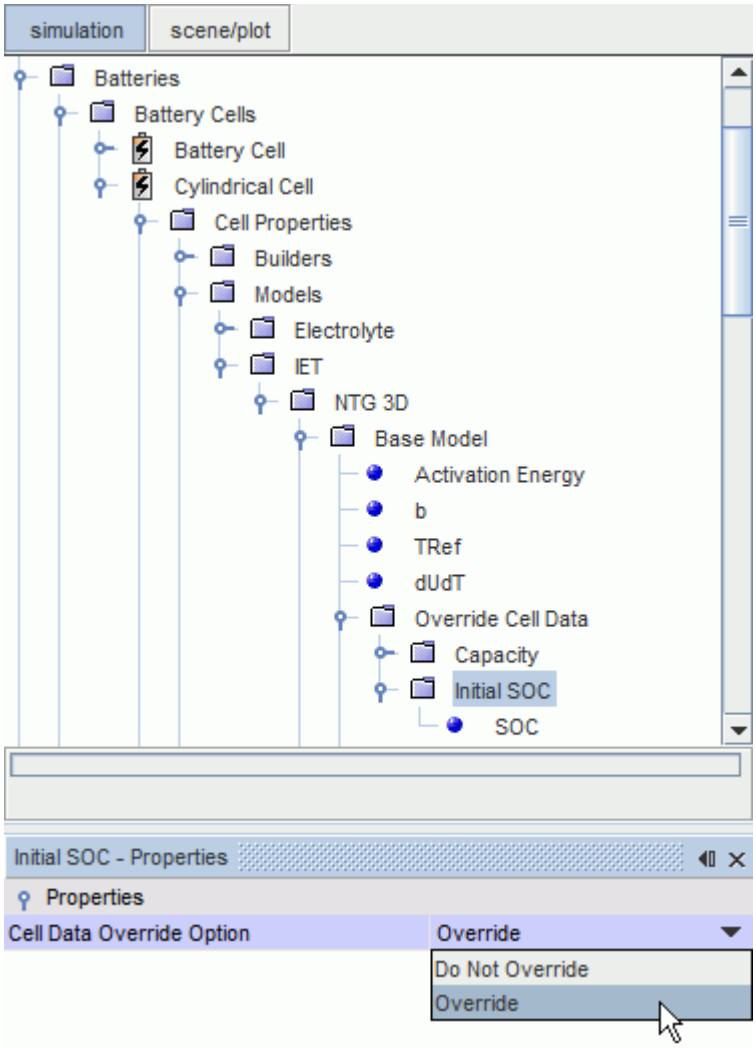
To override the cell capacity or state of charge settings:

1. Under the appropriate battery cell node, expand the **Cell Properties** > **Models** > **IET** > **Model Name** > **Override Cell Data** node.



2. Select the **Capacity** node or the **Initial SOC** node.
3. Set the *Cell Data Override Option* property to **Override**.

   The corresponding node is added. The example below shows setting the override for the **Initial SOC** node.



4. Select the **Capacity** or **SOC** node and set the corresponding override property:
   - **Value** (for *Capacity*): Set the cell capacity in Ah or coulombs.
   - **Percentage** (for *SOC*): Set the cell state of charge as a percentage of its capacity.

Note: When you run a simulation, if the battery cell state of charge reaches 0% or 100%, the state of charge of the cell becomes fixed at that value. The simulation continues to run, but uses the constant value that the model computes at the fixed state of charge. Obviously, a real battery cell does not behave in this way.

See Also:

DistNP Model

NTG (Newman, Tiedeman, Gu) Model

RCR Model

Setting Up a Battery Module

The battery module is the core component in battery module simulations. It contains sets of battery cells, and is connected directly to the electrical load.

The process for setting up a battery module involves the following steps:

- Creating a New Battery Module and Defining the Number of Cells in the Module
- Setting Up Battery Module Cells
- Setting Up Terminal Connections
- Setting Up Tab Connections
- Specifying the Battery Module Resistance Values

Once these steps have been carried out, you can generate the parts, connector parts, and regions of the battery module.

Contents:

What Is a Battery Module?
Defining the Number of Cells in the Module
Defining the Module Configuration
Setting Up Battery Module Cells
Creating an Unrolled View of a Battery Cell
Post-Processing Results with Electrode Masks
Setting Up Terminal Connections
Setting Up Tab Connections
Specifying the Battery Module Resistance Values

See Also:

Generating a Battery Module

What Is a Battery Module?

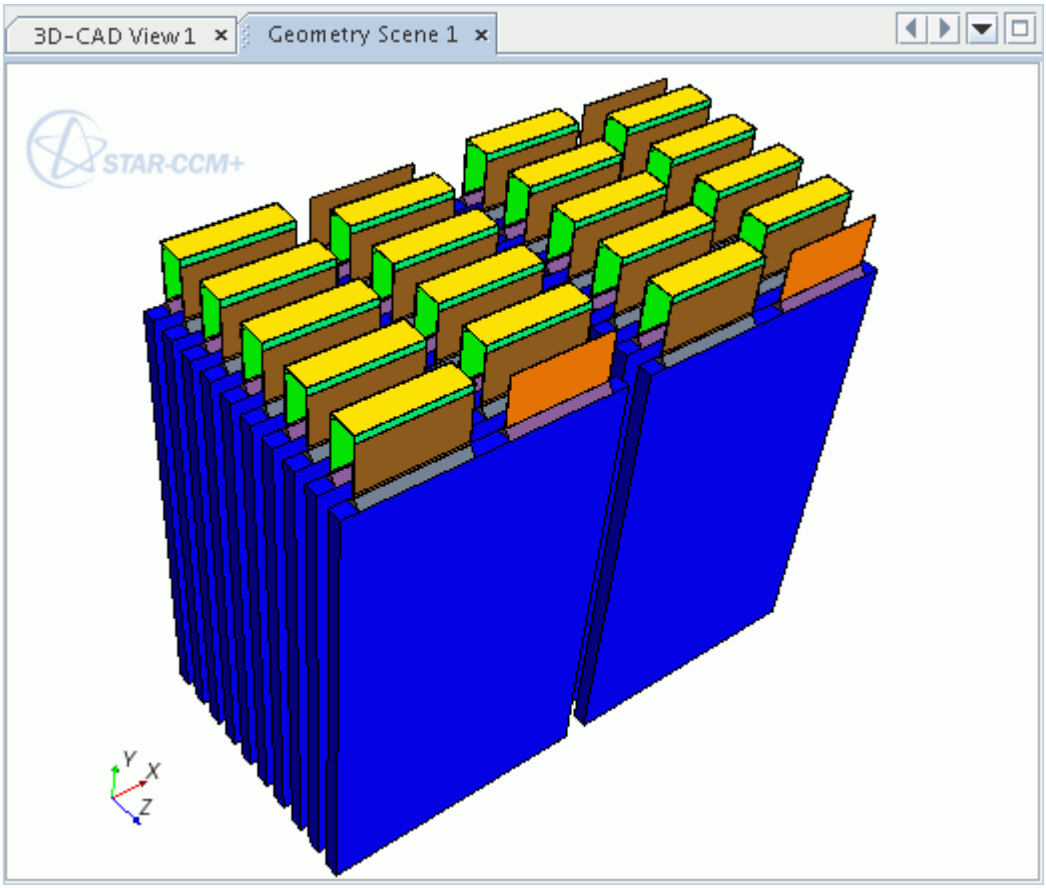
A battery module consists of one or more battery cells. The cells within a module are electrically connected in a series/parallel configuration. A simulation can contain several battery modules.

A set of cells that are connected in series is referred to as a string and the strings of cells are connected in parallel. The number of cells that are connected in each string is N Series, and the number of strings that are connected in parallel within each module is N Parallel. The total number of cells in a module is the product of N Series and N Parallel. The circuit solver for each module automatically makes electrical connections according to the user-designated values for N Series and N Parallel.

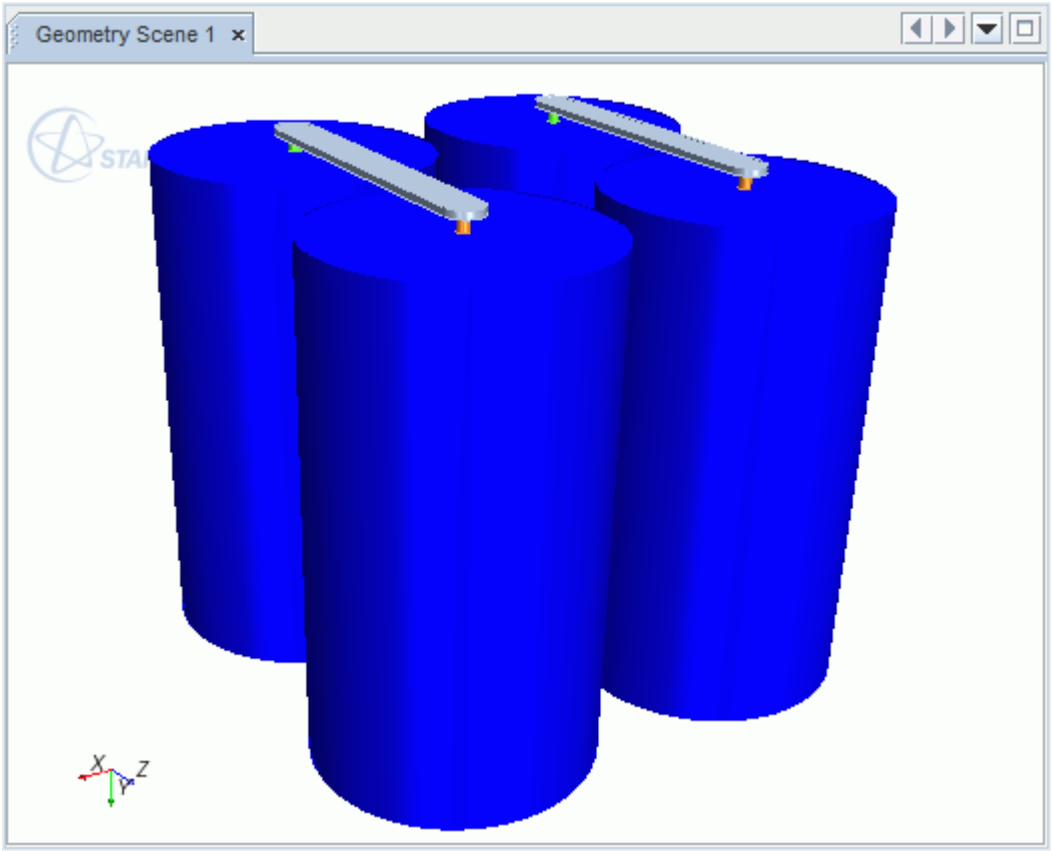
The electrical connections between cells are not explicitly displayed in STAR-CCM+. However it is possible to observe the connections after an electrical time step solution using the Tab Voltage field function. Each battery

cell in a module has a positive and negative tab. The Tab Voltage function displays the circuit voltage that is computed at each tab that is displayed in a module.

You can add physical connectors between the cells of the battery modules. The main purpose of adding physical tab and terminal connections is to model the thermal solution more accurately. The physical connectors that you add must be consistent with the electrical connections that the circuit solver makes automatically. Within a string of cells, tab connectors join a positive or negative tab of one cell to the complementary tab of an adjoining cell. The open tabs at either end of the string of cells are referred to as terminals and these connections can be physically modeled using terminal connections. Both tab and terminal connections can be set up as illustrated below.

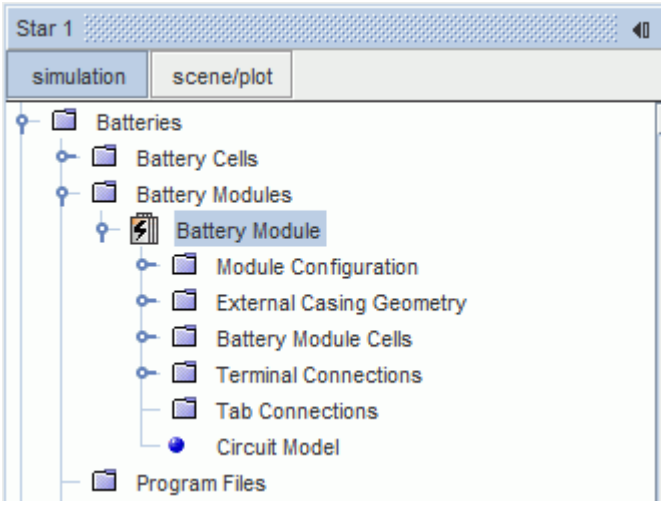


For spiral cells, a battery module looks as follows, showing two parallel strings of cells that are connected in series.



In the above examples, alternate cells have been rotated in order to align the positive tab of one cell with the negative tab of its neighbor.

Battery modules are stored under the **Battery Modules** manager node. Each **Battery Module** node has properties and a pop-up menu.



Battery Module Pop-Up Menu

The following options are available when you right-click on a battery module node.

Create Circuit Element	Creates a circuit element from the battery module. A circuit element is required to add the battery module to a circuit.
Generate Battery Parts	Generates the battery parts.
Generate Connector Parts	Generates the battery connector parts.
Create External Casing Part	Creates external battery casing part.
Generate Regions...	Generates the battery regions.
Setup Connector Ohmic Heating	Sets up connector ohmic heating.

Battery Module Properties

Mesh Continuum	Assigns a mesh continuum to the battery module.
Index	Provides an index value for the battery module.
Module Local Coordinate System	Sets the local coordinate system that is associated with the battery module.
N Series	Sets the number of battery cells that are connected in series in each string.
N Parallel	Sets the number of strings in the module. Each string is connected in parallel.

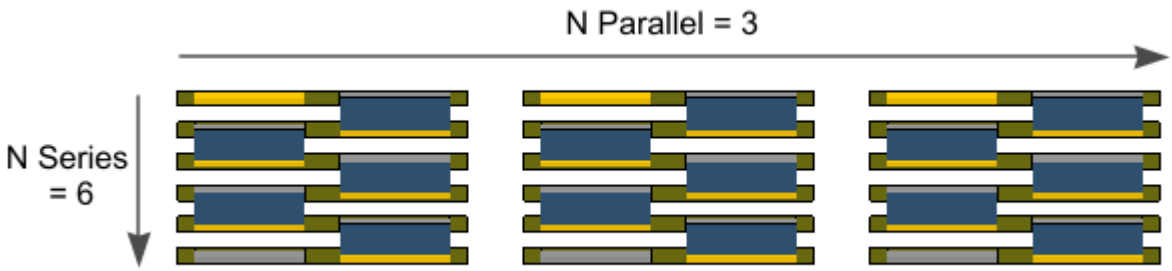
See Also:

Adding Circuit Elements to a Circuit
Defining External Battery Casing
Generating a Battery Module
Setting Up Connector Ohmic Heating
Setting Up Tab Connections
Setting Up Terminal Connections
What Is a Circuit Element?
Working with a Battery Cell

Defining the Number of Cells in the Module

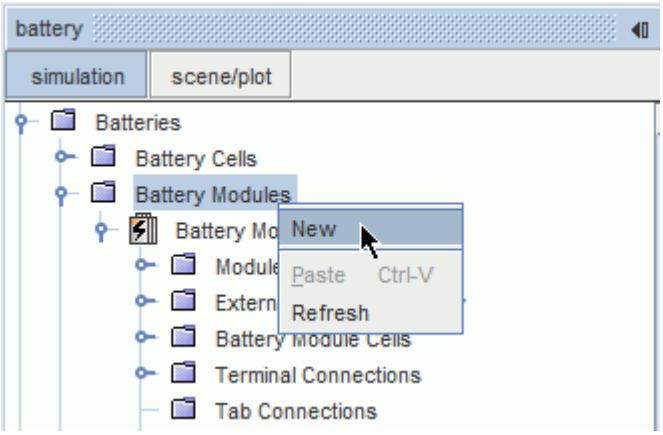
The number of battery cells in the battery module is defined by specifying the values N Series and N Parallel. N Series is the number of cells that are connected in each series string, and N Parallel is the number of series strings that are connected in parallel. Strings are connected in parallel, and each string contains the same number of battery cells.

You can set the number of cells in the battery module in the properties of the **Battery Module** node: *N Series* and *N Parallel*. These properties are shown in the following diagram using stacked cells.



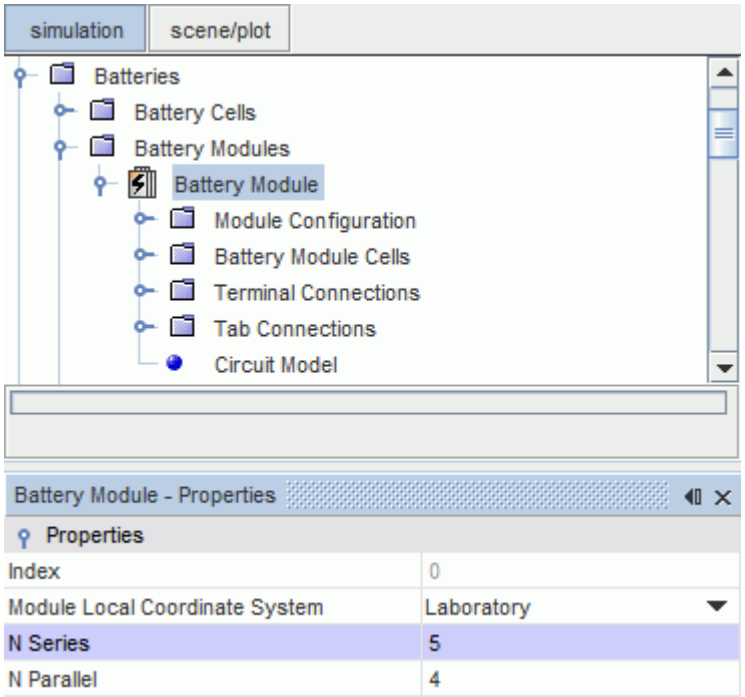
To create a battery module and set the number of cells in the module:

1. Right-click the **Battery Modules** node and select **New**.



2. Select the new **Battery Modules** > **Battery Module** node.
3. In the *Properties* window, enter the values of **N Series** and **N Parallel**.

   The number of cells that are shown in the battery scene is not updated until you generate the battery parts.



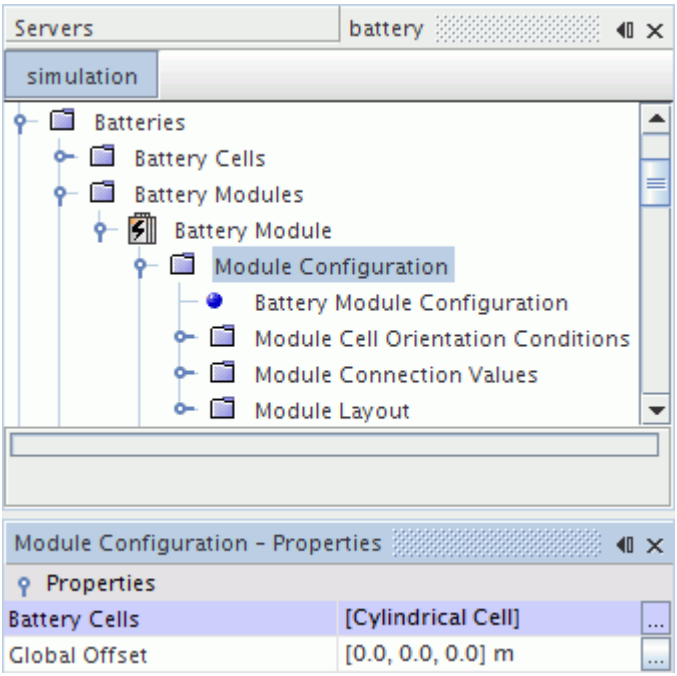
You can also set N Series and N Parallel using the **Battery Module** > **Module Configuration** > **Module Layout** > **Module Grid Size** menu or by using the **Setup Battery Module Cells** option.

Defining the Module Configuration

For most cases, the **Module Configuration** node is the recommended module setup option. You specify the *Battery Cell* property before you can configure any module setup parameters. The other configuration options are not visible until you have selected a battery cell.

To define the module configuration:

1. Select the **Module Configuration** node and, in the *Battery Cells* property, specify the battery cell that the battery module contains.



STAR-CCM+ adds a **Battery Module Configuration** sub-node, along with **Module Cell Orientation Conditions**, **Module Connection Values**, and **Module Layout** subfolders to the **Module Configuration** node.

The **Module Configuration** node allows you to change the position of each individual battery module by specifying its global offset vector. The offset is defined in terms of the local coordinate system that is associated with the battery module. The offset is used to set the battery module layout using uniform spacing between cells.

2. In the *Global Offset* property, specify the appropriate global offset vector.

If you define the position and orientation of the cells using the module configuration tool, STAR-CCM+ uses these settings instead of the settings that are set in the **Battery Module Cells** node.

Contents:

Setting the Battery Module Configuration
Specifying the Module Cell Orientation
Defining the Module Layout

Setting the Battery Module Cell Configuration

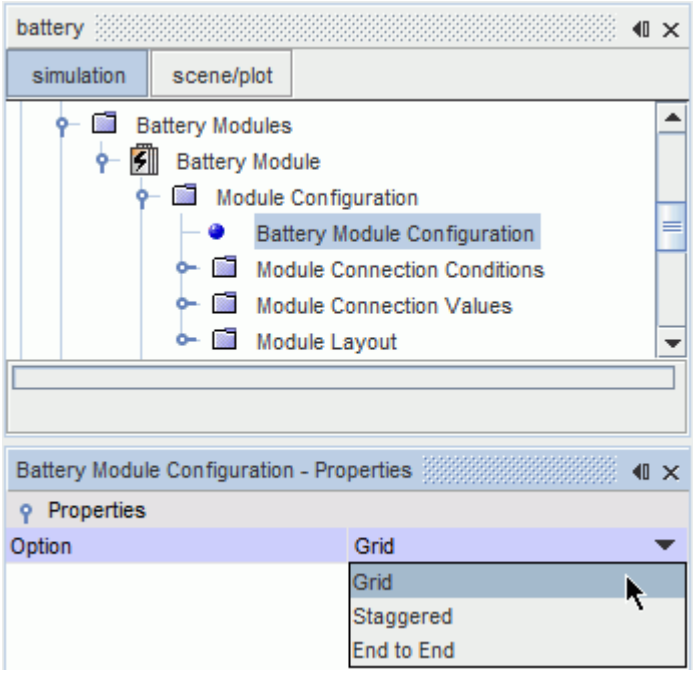
The **Battery Module Configuration** node defines how the battery cells are configured in the module.

The following options are available:

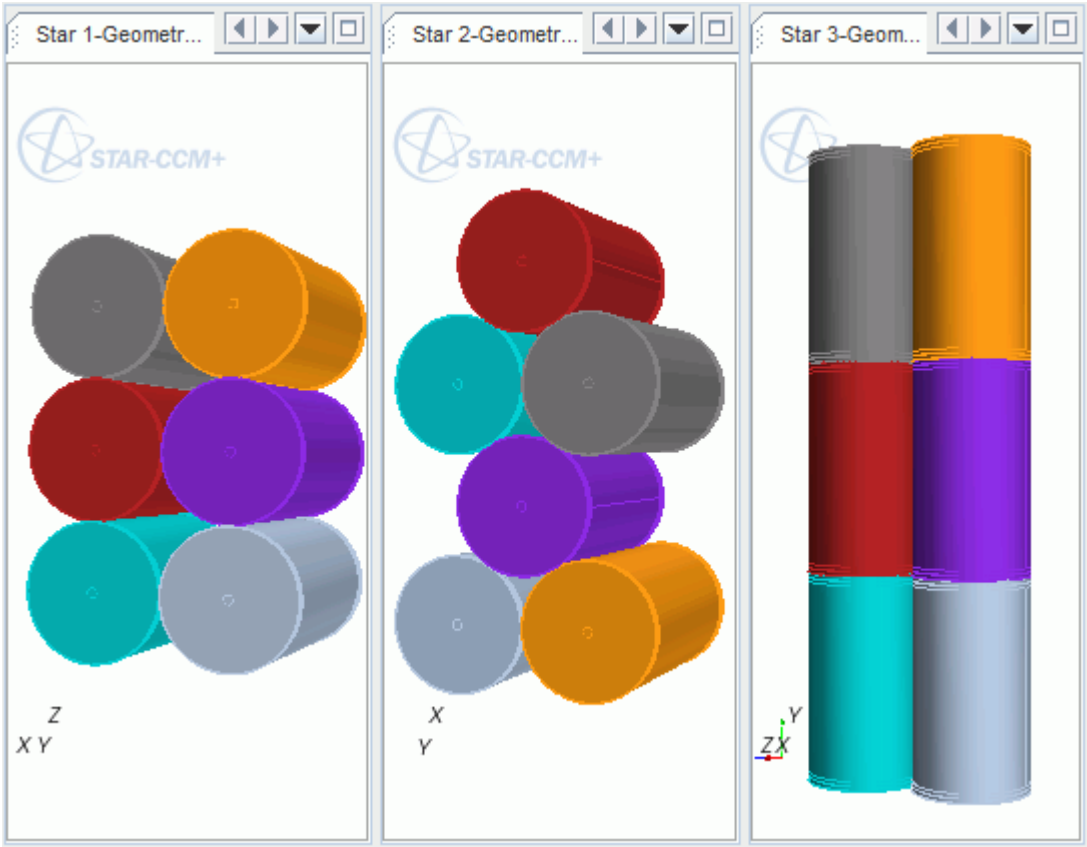
- Grid—cells are placed on a regular grid that defines the number of cells in series (rows) and parallel (columns).

- Staggered—spiral cells are placed in staggered rows. The number of cells in each series string determines the number of cells. The number of parallel strings defines the number of staggered rows in the configuration.
- End to End—spiral cells are stacked end to end based on the number of cells in series. The number of parallel strings sets the number of stacks.

The staggered and end to end options are only available for spiral cells.



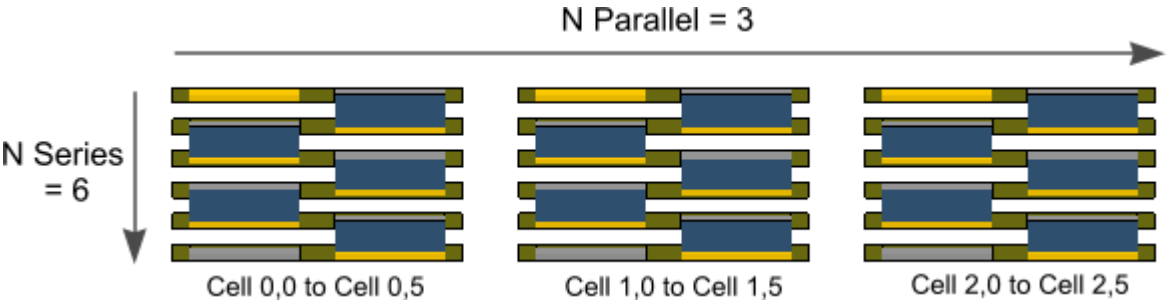
The following screenshot shows the different configurations using three cells in series and two cells in parallel. From left to right, it shows grid, staggered, and end to end configurations.



Specifying the Module Cell Orientation

In a battery module, the cells are connected in an *N Series* X *N Parallel* grid.

For a module with *N Series* = 6 and *N Parallel* = 3, there are 18 cells. **Cell 0,0** is connected in series to **Cell 0,1**, **Cell 0,2**, **Cell 0,3**, **Cell 0,4**, and **Cell 0,5**. Similarly, **Cell 1,0** to **Cell 1,5** (and **Cell 2,0** to **Cell 2,5**) are connected in series.

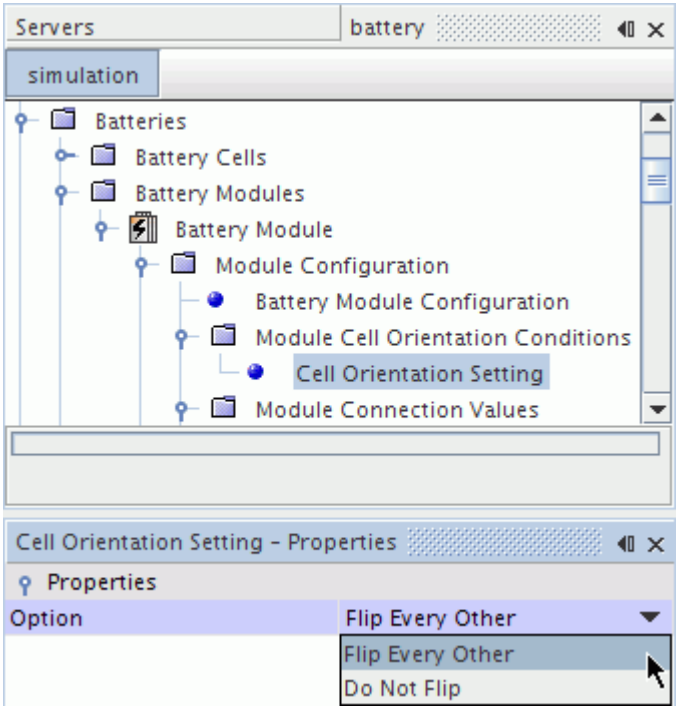


The tab locations in an individual battery cell are determined from the `.tbm` file. You determine the cell configuration within a battery module as described in Setting the Battery Module Configuration. Typically, the cells are oriented as shown in the previous figure. The negative tabs of cells **0,1** - **0,4** are connected to the positive tabs of cells **0,2** - **0,5** respectively. For the parallel connections, the positive terminals of **Cell 0,0**, **Cell**

1,0, and **Cell 2,0** are connected to the negative terminal of the load. Similarly, the negative terminals of **Cell 0,5**, **Cell 1,5**, and **Cell 2,5** are connected to the positive terminal of the load.

STAR-CCM+ offers an option to flip the cells within each string so that the negative tabs of each cell align with the positive tabs of the adjacent cells. Flipping the cells does not change the electrical connection: flipping the cells make tab connectors that follow the electrical circuit connections easier to model.

You set this property in the **Module Cell Orientation Conditions** > **Cell Orientation Setting** node. To flip the series cells, set *Option* to **Flip Every Other**.

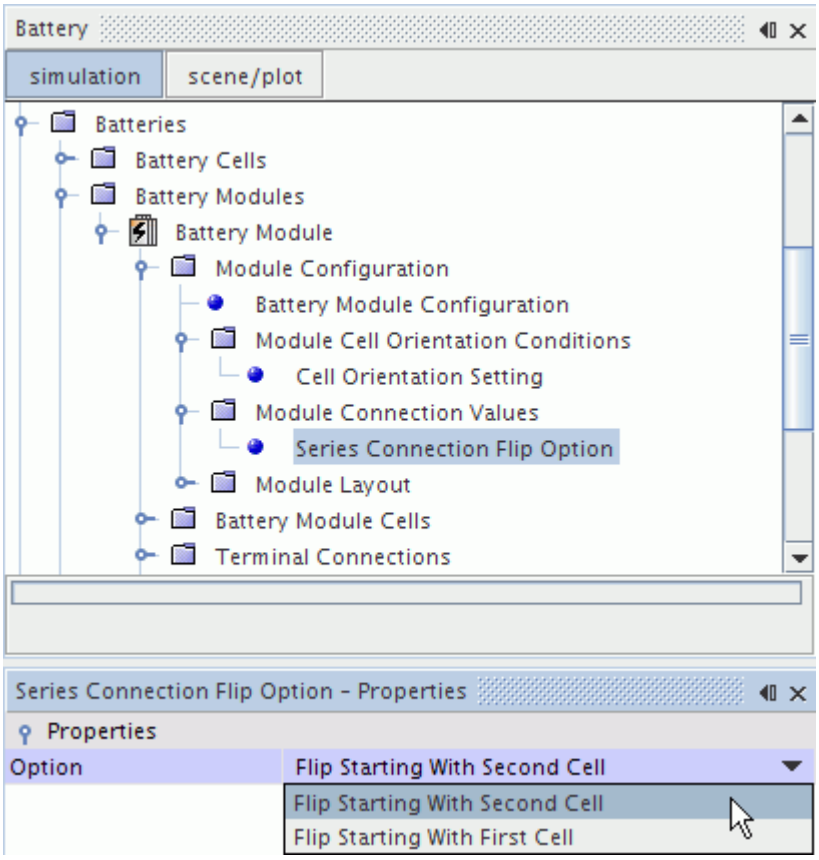


You can also leave the tabs unaligned (for example, in setting up a string of cylindrical cells that uses the staggered cell arrangement).

- Set *Option* to **Do Not Flip**.

Note: Changing the cell orientation does not change the electrical connections. Take care to add tabs consistent with the underlying electrical connections.

To change which side of the displayed cell corresponds to the open positive tab, you can flip either the odd or the even cells. You set this property in the **Module Connection Values** > **Series Connection Flip Option** node.

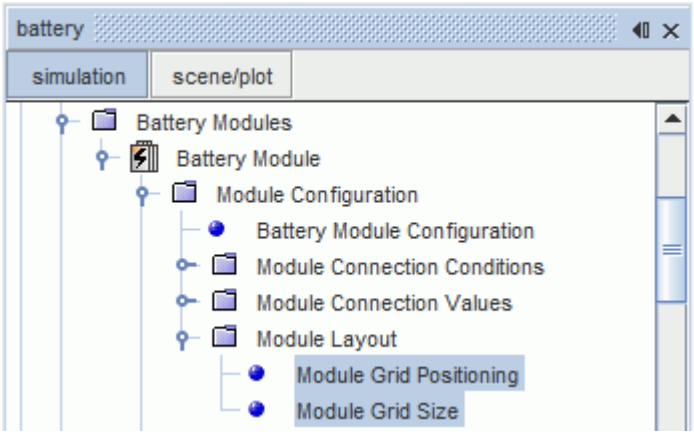


The first cell in a row of a module (for example, **Cell 0,0**) is an odd cell. The second cell in a row of a module (for example, **Cell 0,1**) is an even cell.

- To flip the odd cells, set *Option* to **Flip Starting With First Cell**.
- To flip the even cells, set *Option* to **Flip Starting With Second Cell**.

Defining the Module Layout

In the sub-nodes of the **Module Layout** manager node, you can set the position of the cells in the configuration and the size of the battery module.



Setting the Module Positioning

The **Module Positioning** node lets you specify the properties that are appropriate to the module configuration. The settings for each of the possible battery module configurations are summarized in the following tables:

Module Grid Positioning Properties

Offset Type	Specifies if the parallel and series offsets are absolute values or are based on the size of the battery cell.
Parallel Offset	Specifies the parallel offset in the units that are appropriate to the specified Offset Type.
Parallel Direction	Specifies the vector direction that is applied to the parallel offset.
Series Offset	Specifies the series offset in the units that are appropriate to the specified Offset Type.
Series Direction	Specifies the vector direction that is applied to the series offset.
Local Coordinate System	Sets the local coordinate system that is associated with the battery module.

The default offset and spacing between cells is set to 0, which produces a valid configuration of cells in contact with each other. For a staggered configuration, if you use the default positioning, the cells are offset based on the exterior diameter of the can.

If you want to perform a global rotation on the battery module, create a coordinate system in the appropriate orientation and then apply it to the battery module. The coordinate system handles the rotation and the global offset.

Module Staggered Positioning Properties

Staggered Row Offset	Specifies the geometric offset between staggered rows.
Staggered Row Direction	Specifies the vector direction that is applied to each row of staggered cells. The default is [1.0, 0.0, 0.0].
Gap Between Cells	Specifies the distance between each cell in a staggered row.
Local Coordinate System	Sets the local coordinate system that is associated with the battery module.

Module End to End Positioning Properties

Stack Offset	Specifies the geometric offset between stacks.
Local Coordinate System	Sets the local coordinate system that is associated with the battery module.

Setting the Module Size

The **Module Size** node lets you specify the module size parameters. The default size of the battery module is the same as the number of cells in series and parallel that you set (the N Series and N Parallel properties) using the Battery Module node.

The settings for each of the possible battery module configurations are:

Module Grid Size	• N Series • N Parallel
Module Staggered Size	• Number of Cells in Staggered Row (the default value is the N Series setting) • Number of Staggered Rows (the default value is the N Parallel setting)
Module End to End Size	• Cells in Stack (the default value is the N Series setting) • Number of Cell Stacks (the default value is the N Parallel setting)

Setting Up Battery Module Cells

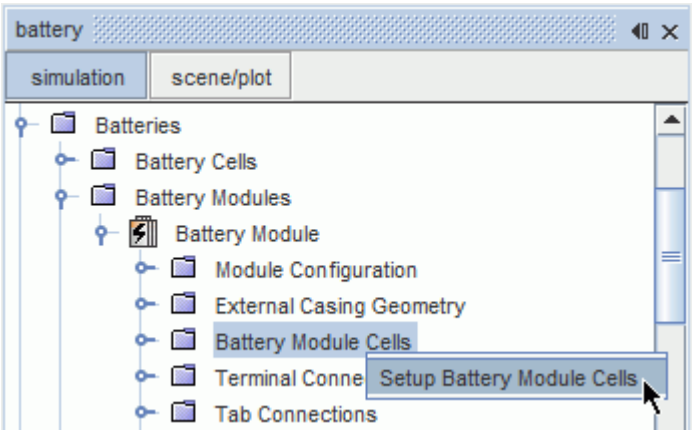
As an alternative to using the Module Configuration feature, you can define the position and orientation of each cell within the module using the battery module cell setup tool. This option lets you set non-uniform spacing in your cell formation.

This option also lets you align the series cells along different axes and define the rotation of cells that are flipped along any axis or angle. However it is slower to use as you set up each parallel string of cells manually.

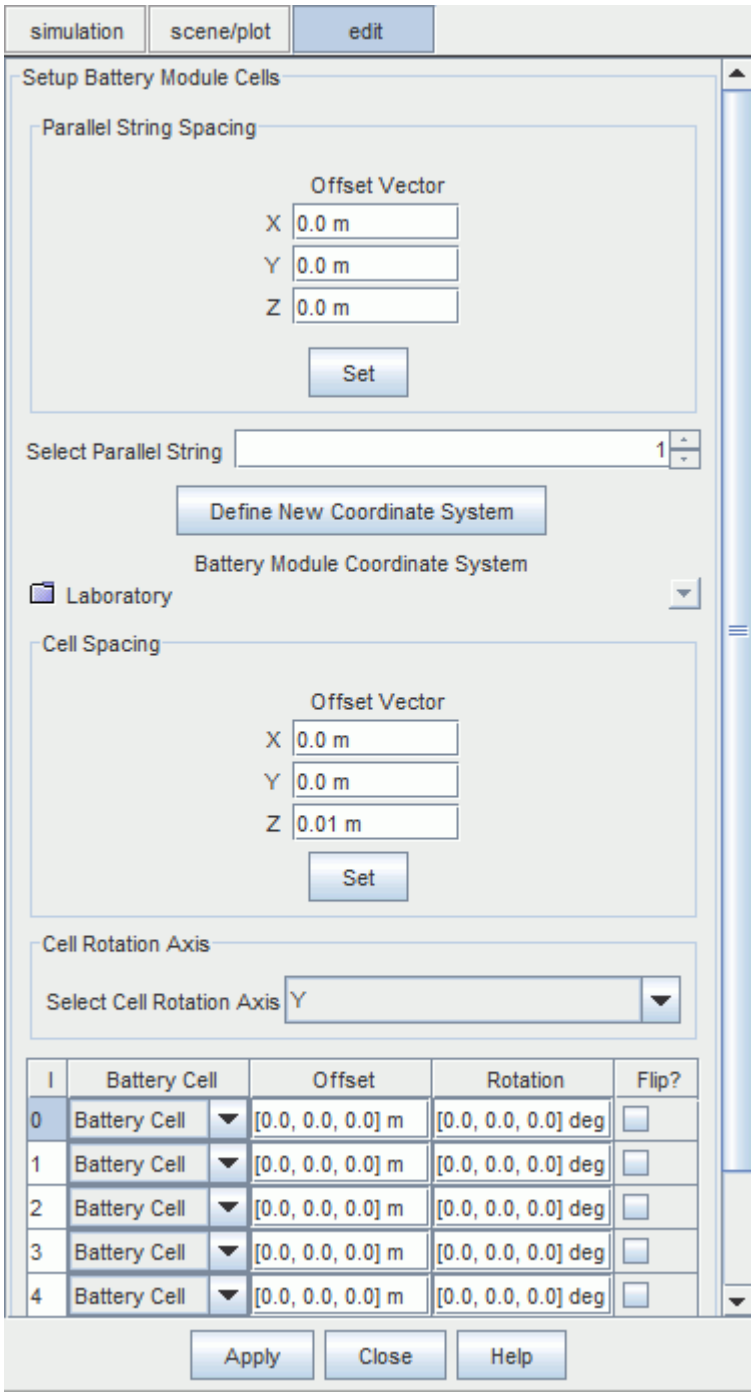
If you configure your module using the **Module Configuration** node, these settings supersede the ones that are defined using the **Battery Module Cells** node.

To set up battery module cells:

1. Right-click the **Battery Module** > **Battery Module Cells** node and select **Setup Battery Module Cells**.



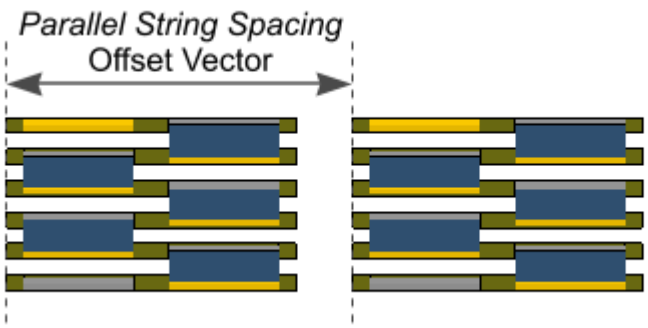
The *Setup Battery Module Cells* panel appears.



For a battery module containing multiple cells, you can specify the spacing between the parallel strings using the *Parallel String Spacing* box.

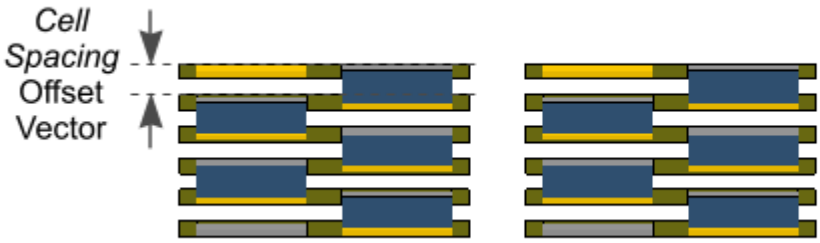
2. Enter a value for the Offset Vector.
3. Click **Set**.

   Each successive parallel string is offset from the previous string by the specified amount.

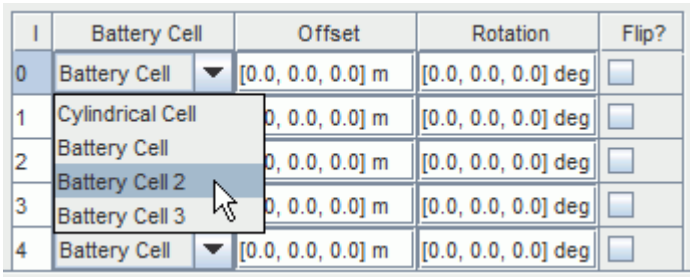


After setting the spacing of each parallel string, you can define the spacing of the cells within each string. You work on each string independently. To select the string that you wish to work on, use the Select Parallel String box. Choose the local Cartesian coordinate system for your battery module from the **Battery Module Coordinate System** drop-down menu. You can create one interactively by either clicking the **Define New Coordinate System** button or using the coordinate systems tool.

In the *Cell Spacing* box, enter an Offset Vector for the cells in the selected string and click **Set**. The position of the cells in the string is updated. For modules containing multiple parallel strings, select the next string using the **Select Parallel String** box and enter the desired cell spacing as described above.



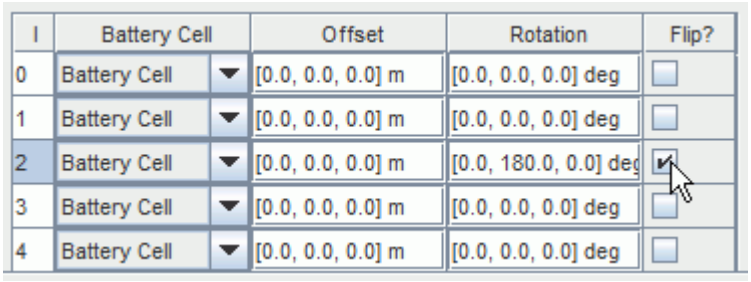
The table at the bottom of the dialog shows a summary of the position of each cell in the selected string. You can use this table to change the position of individual cells by typing a position vector directly into the Offset column. If a battery cell has been imported into the simulation, STAR-CCM+ automatically assigns this cell to each position in the battery module. If more than one cell has been imported into the simulation, you can choose to assign a different battery cell to a particular position in the module. You assign a battery cell to a position by using the drop-down menu in the Battery Cell column.



In order to align the negative and positive electrodes of adjacent cells, flip some of the battery cells in the module:

1. Select the axis about which the cells are flipped from the **Select Cell Rotation Axis** menu in the *Cell Rotation Axis* box.
2. Activate the **Flip?** checkbox for any cells that you wish to flip.

   The rotation value in the Rotation column is updated automatically.



You can now also define a new Cartesian coordinate system for your battery cells:

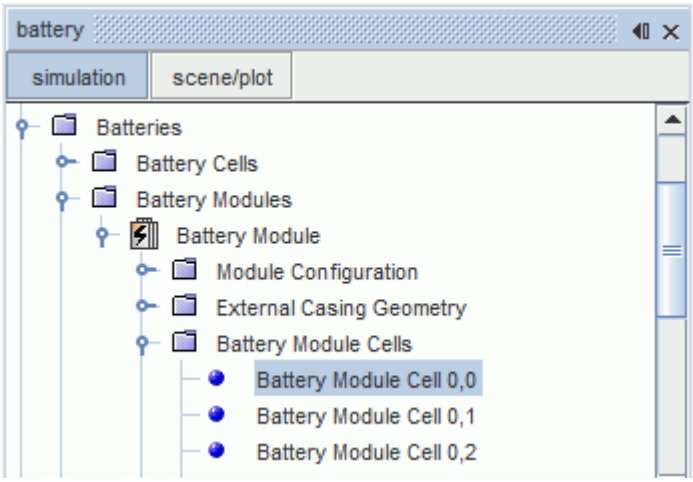
1. Click the **Define New Coordinate System** button.

   In the *Edit* panel, change the axis definition of the coordinate system.
2. When you have completed the battery module setup, click **Apply**, and then click **Close**.

You can return to this panel to change the position of the cells, if necessary.

Setting Properties of Battery Module Cells

The cells that are defined in the battery module are stored under the **Battery Module** > **Battery Module Cells** node. You can define the position, type, and orientation of each cell in the module by specifying the properties of the battery module cells node. These settings can also be applied using the *Setup Battery Module Cells* panel, and are updated at the end of the battery module cell setup.



Battery Module Cell Properties

Battery Cell	Selects the imported battery cell.	
Orientation	Specifies whether the cell is flipped (rotated through 180 degrees).	
	Activated	Cell is flipped.
	Deactivated	Cell is not flipped.
Rotation	Specifies the rotated position of the cell.	
Translation	Specifies a position vector for the cell.	
I Series	Indicates the i component for the cell index, in a module of size [i, j].	
I Parallel	Indicates the j component for the cell index, in a module of size [i, j].	

Creating an Unrolled View of a Battery Cell

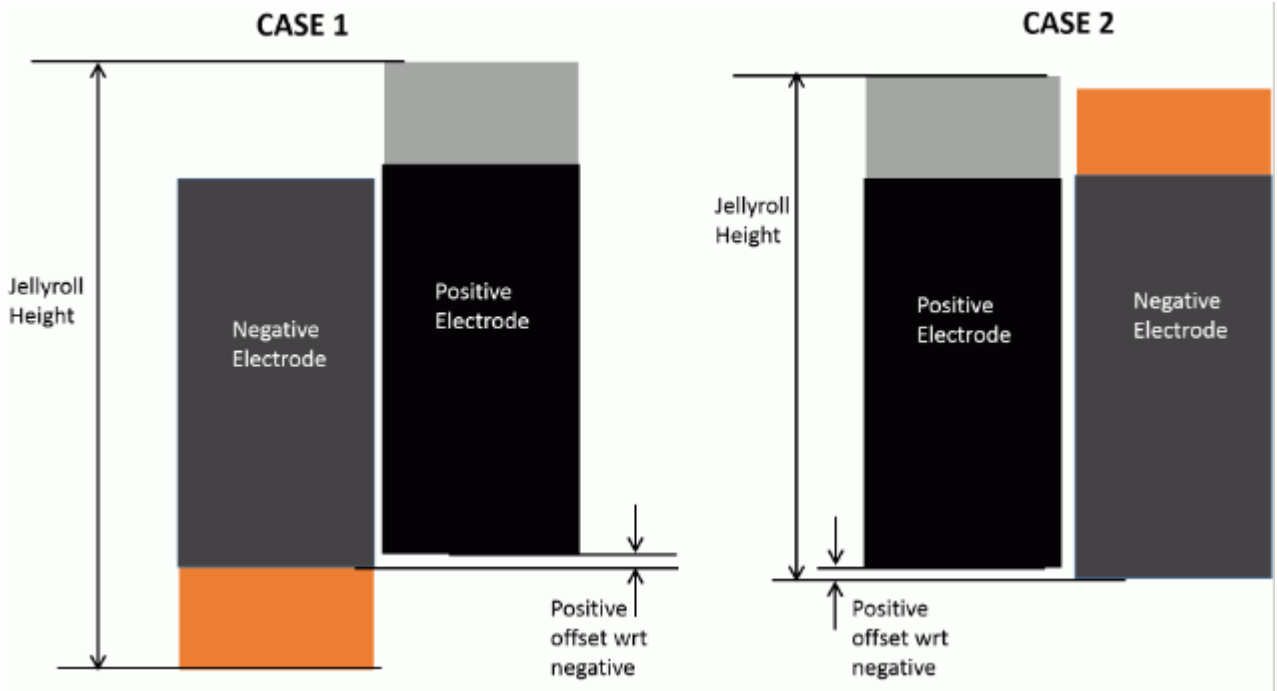
Cylindrical and prismatic wound cells are created by winding the electrodes and separator around a central mandrel. The resulting cell is known as a jelly roll. You can create an unrolled view of the jelly roll to provide a picture of the internal connectivity of the electrical mesh and the battery cell structure. The unrolled view is available only for wound cells—it is not available for coin cells or stacked cells.

When you load a .tbm file and generate the corresponding cell, STAR-CCM+ creates 3D parts for the jelly roll. When you transfer geometric parts to regions and perform the simulation, the solution is stored in the volume mesh for the jelly roll. However, in many cases it is important to understand what is happening along the length of the electrodes that are contained within the jelly roll.

The unrolled view of the jelly roll is a derived part with the same dimensions that the positive or negative electrode would have if it was flat, that is, if it was unrolled from the jelly roll. The inner and outer e-cell centroids in the jelly roll mesh are averaged to develop a spiral representation. This representation is unrolled along the spiral path that is defined by the winding of the electrodes from the cell definition, and displayed as a 2D view.

Separate unrolled views are available for the negative electrode, positive electrode, and both electrodes together. Separate unrolled views are also available for the inner electrochemical cell, outer electrochemical cell, and both together.

STAR-CCM+ calculates the cell height as illustrated below:

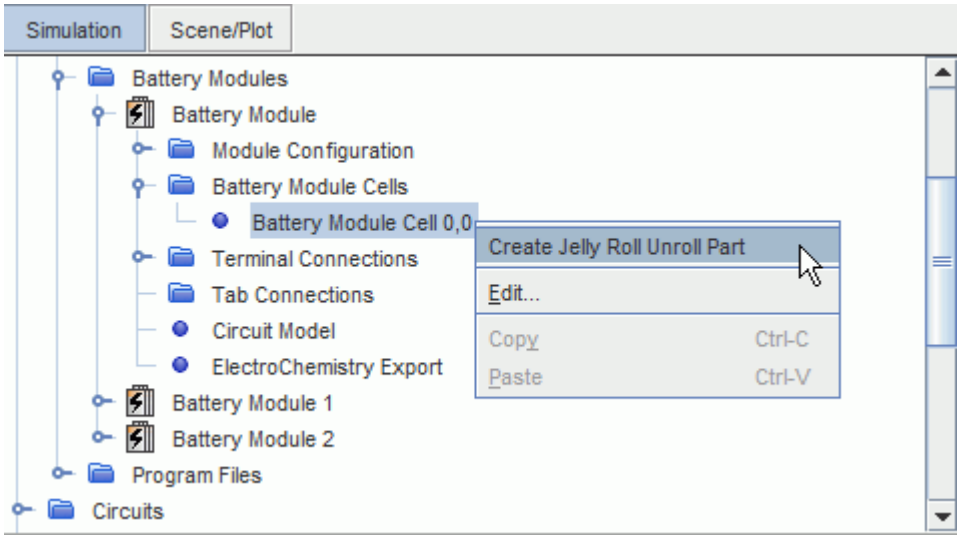


The Vertical tab orientation settings in the battery cell can have a significant effect on the calculated jellyroll height. For example, the overhang that is caused by the difference in height of the coated and uncoated parts of the electrodes can increase the cell height significantly.

Note: In some batteries, the negative and positive electrodes can be different sizes. This size difference results in regions outside of the jelly roll that contain only one electrode and therefore contain no real e-cells. When the jelly roll is unrolled, the composite view has a non-rectangular outline.

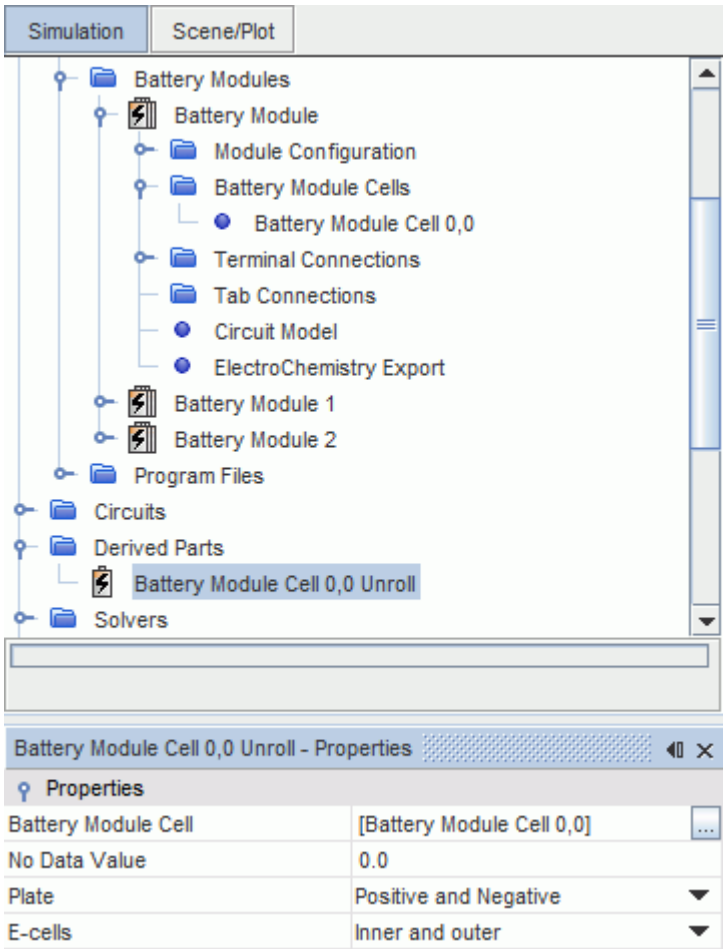
To create an unrolled view of a prismatic or cylindrical battery cell:

1. Expand the **Battery Modules** > **Battery Module** > **Battery Module Cells** node.
2. Right-click the appropriate battery module cell and click **Create Jelly Roll Unroll Part**.



The new part appears under the **Derived Parts** node.

The new part is named after the battery cell and is appended with Unroll.



Jelly Roll Unroll Part Properties

Battery Module Cell	Selects the battery module cell.	
No Data Value	Specifies the value that is used when no e-cell centroid data is available.	
Plate	Specifies the plates that are unrolled and displayed.	
	Positive and Negative	Both the positive and negative plates are displayed.
	Positive	Only the positive plates are displayed.
	Negative	Only the negative plates are displayed.
E-cells	Specifies the e-cells that are unrolled and displayed. For any of the e-cell centroids that do not have values, the No Data Value is used.	
	Inner and Outer	Data is displayed if both the inner and the outer e-cell centroids have values. The average value is used.

	Inner	The values of the inner e-cell centroids are displayed.
	Outer	The values of the outer e-cell centroids are displayed.

Contents:

Displaying an Unrolled View of a Battery Cell

Displaying an Unrolled View of a Battery Cell

You can use an unrolled view of a prismatic or cylindrical battery cell to visualize the distribution of state of charge, temperature, exchange current density, and other values, within the cell. You add the unrolled derived part to a scene and plot the corresponding field functions along the length of the positive and negative electrodes.

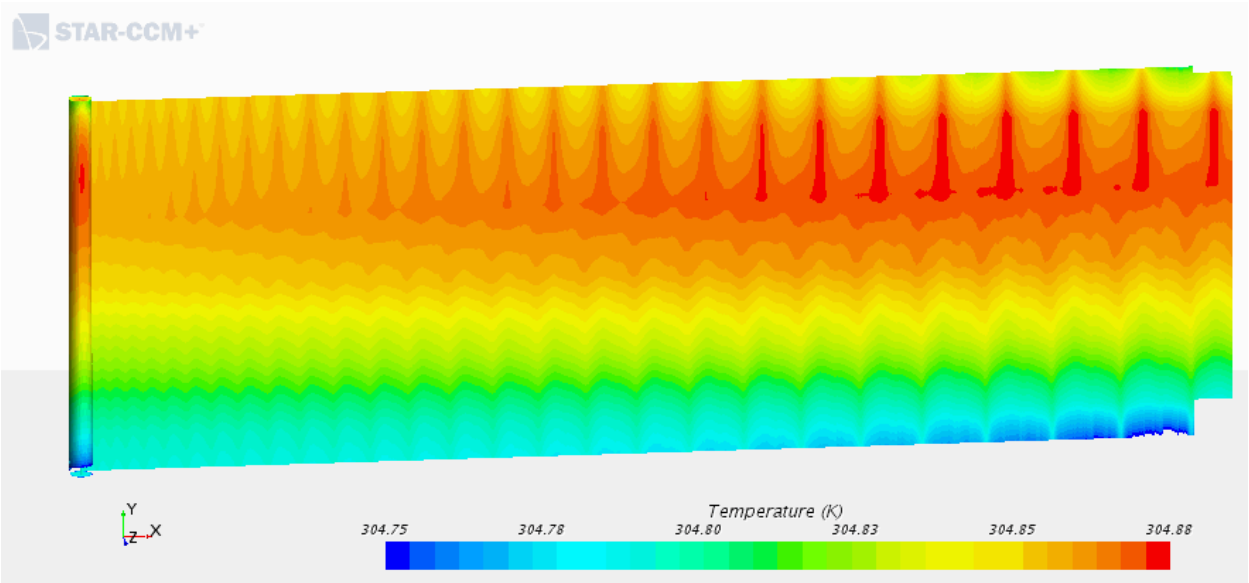
Note the following:

- Field functions that are computed within STAR-CCM+ (for example, temperature) are mapped to the average spiral representation.
- Field functions that are computed from the battery model are averaged from the original e-cell centroid values.

To display an unrolled view of a prismatic or cylindrical battery cell:

1. Create a scalar scene or a vector scene.
2. Under the **Displayers** > **Scalar 1** or **Displayers** > **Vector 1** node, add the unrolled jelly roll as a derived part.
3. Select the field function that you want to display.

   The example below shows the temperature profile of a jelly roll.



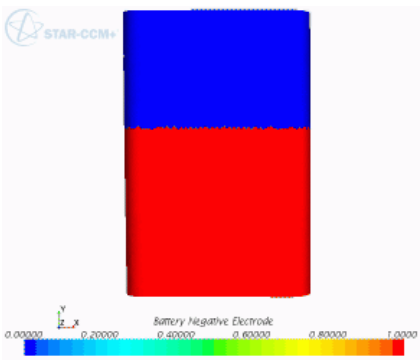
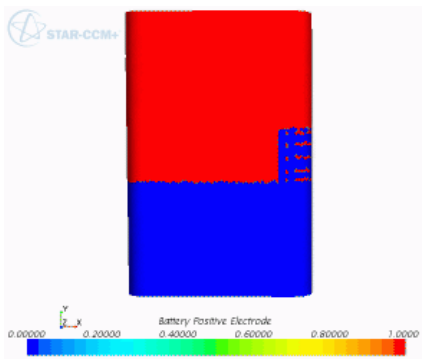
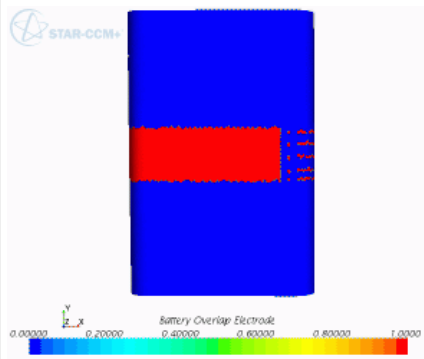
4. If necessary, transform the unrolled view using the standard displayer properties.

   To make the displayed image easier to view, the example above has the Y axis scaled by a factor of 10.

Post-Processing Results with Electrode Masks

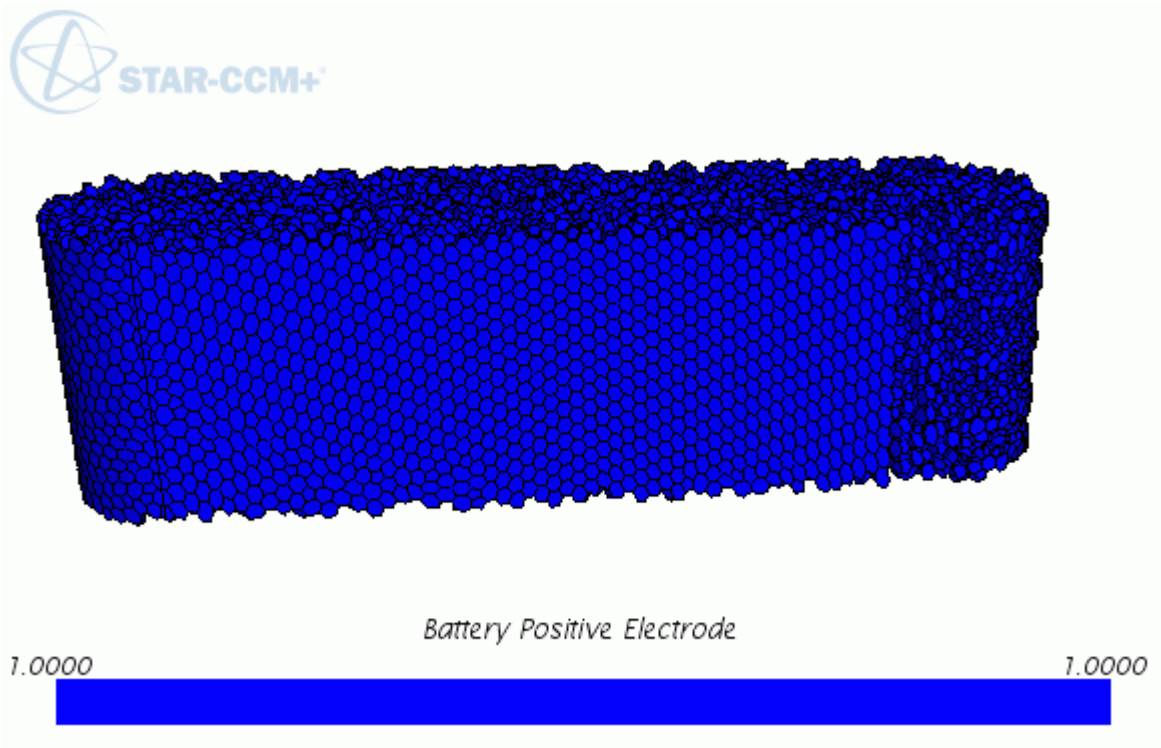
In order to allow a more detailed post-processing analysis of the battery cells, field functions which indicate the location of the electrodes within the battery volume are available.

These field functions are cell masks that can be used to select areas of the jellyroll that contain either the positive electrode, the negative electrode or both electrodes (the overlap). The field functions provide a value of 1 or 0 to indicate the presence or absence of the item of interest in each volume cell within the jellyroll. Examples of these masks are shown below:

Negative Electrode	Positive Electrode	Overlap Electrode
		

These functions are suitable for use in a derived part, such as a threshold part, to select a subset of the mesh cells within the selected region. This derived part can then be used in post-processing to give results that can be more readily compared to BDS values.

An example of a derived part created using the Overlap Electrode mask is shown below:



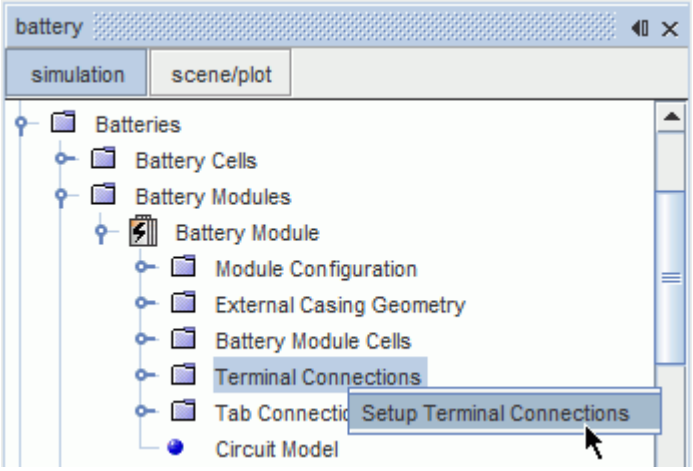
Setting Up Terminal Connections

In a battery module, parallel strings can be connected using terminal connections, which lets you use imported geometry to define physical connections between the strings. These connections are used to compute the thermal solution only.

Whether or not you set up physical connections between the parallel strings, an electrical connection always exists between the parallel strings in the circuit solver.

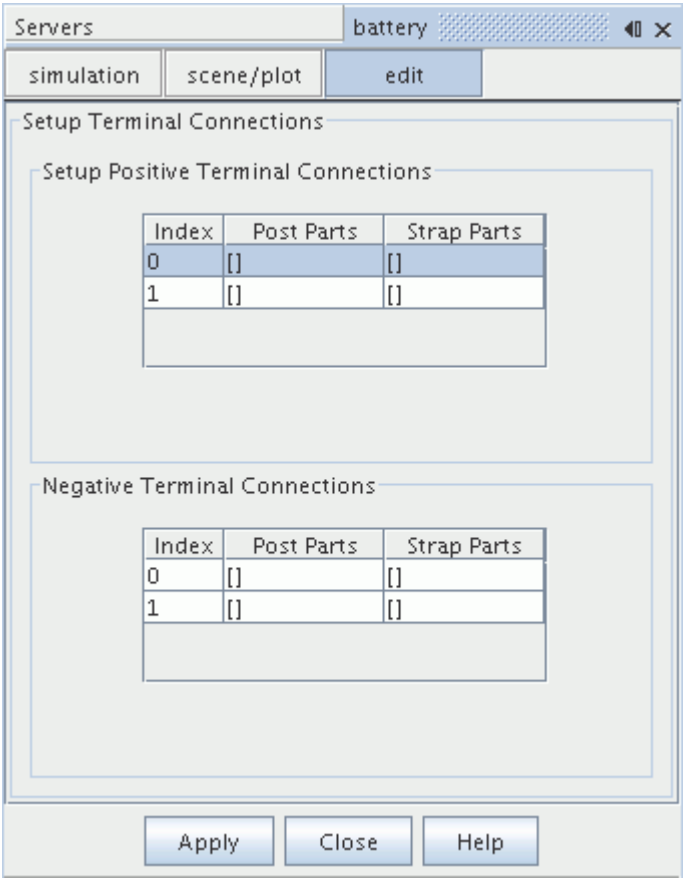
To set up terminal connections:

1. Import the terminal connection geometry into the simulation.
2. Right-click the **Battery Module** > **Terminal Connections** node and select **Setup Terminal Connections**.



3. In the *Setup Terminal Connections* panel, use the *Positive Terminal Connections* and the *Negative Terminal Connections* tables to select the parts for each connection.

   The *Index* value indicates which parallel string the terminal connection corresponds to, that is, 0 is the first parallel string, 1 is the second string, and so on.



4. When the terminal connection setup is complete, click **Apply**, and then click **Close**.

It is also possible to set up terminal connections using the terminal connections nodes.

Setting Properties of Terminal Connections

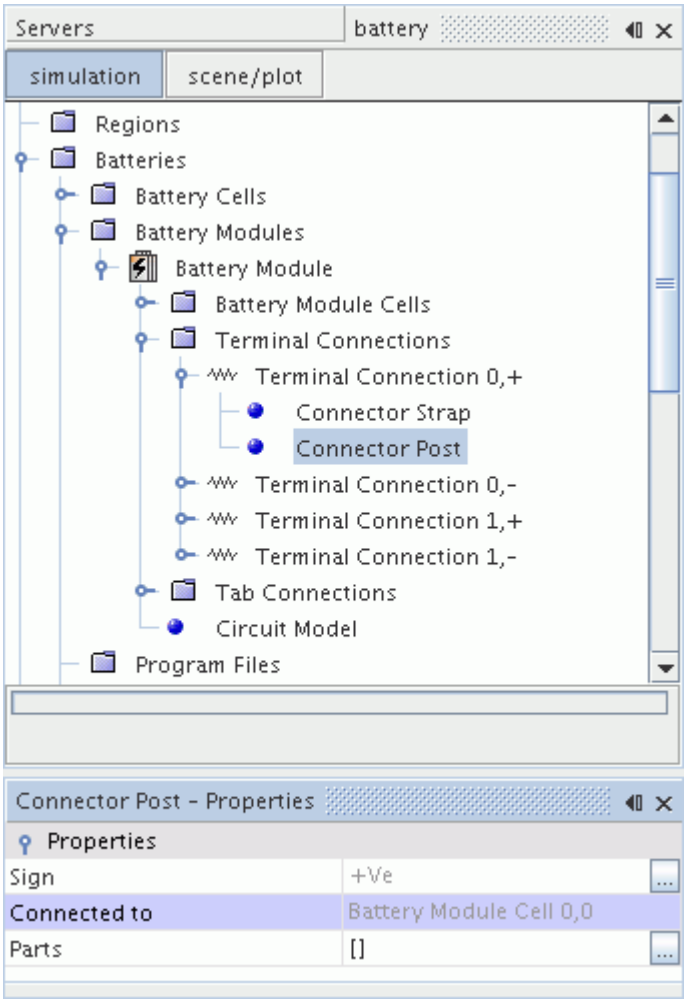
Terminal connections are stored under the **Battery Module** > **Terminal Connections** node. In the properties for the terminal node, you can review which terminals the connection is applied to, and also assign parts to the terminal connection.

Terminal Connection Properties

Post Parts	Selects the parts that define the post.	
Strap Parts	Selects the parts that define the strap.	
Copy Connection Parts	Specifies whether the post and strap parts are copied to other terminal connections.	
	Activated	Parts are copied.
	Deactivated	Parts are not copied.

Reviewing Connected Terminals

You can review the terminals that a particular connection connects using properties of the **Connector Strap** and **Connector Post** nodes. These properties tell you which terminal of a particular battery cell the connection is applied to, for example, the positive terminal of battery module cell 0,0. To review the terminals that are assigned to the connection, select **Connector Strap** or **Connector Post** under the **Battery Module** > **Terminal Connections** > **Terminal Connection** node. The terminal that the connection connects is displayed next to the *Connected to* property.



Setting Up Tab Connections

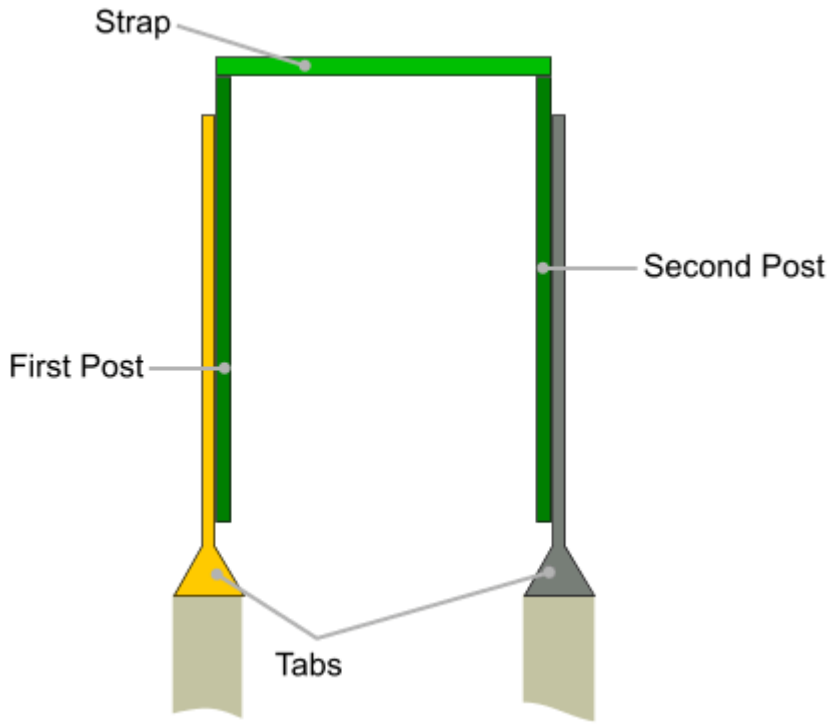
In a battery module, tab connections are used to connect battery cells within a string. Setting up tab connections allows you to use imported geometry to define physical connections between the cells. These connections are used to compute the thermal solution only.

Whether or not you set up physical tab connections, the circuit solver always makes electrical connections between the cells in a series string.

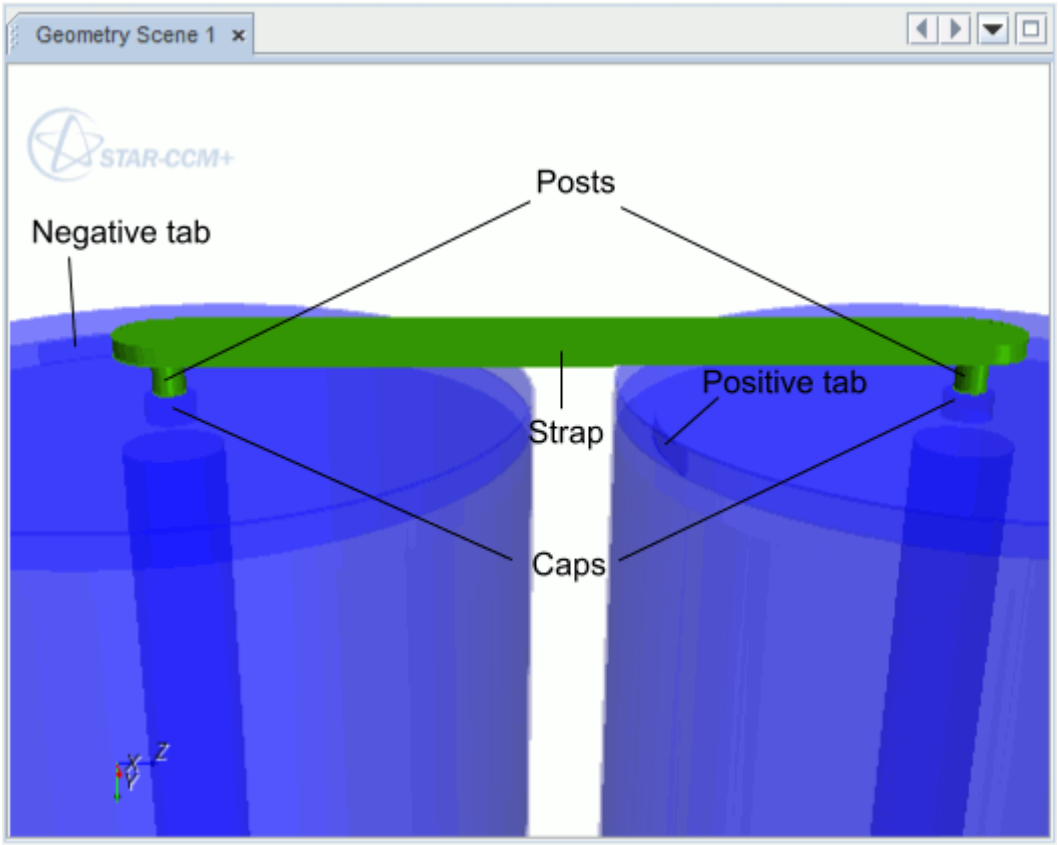
Typically, the tab connector consists of three parts:

- First Post
- Second Post
- Strap

For stacked cells, the post parts must touch the tabs, as shown in the following diagram.



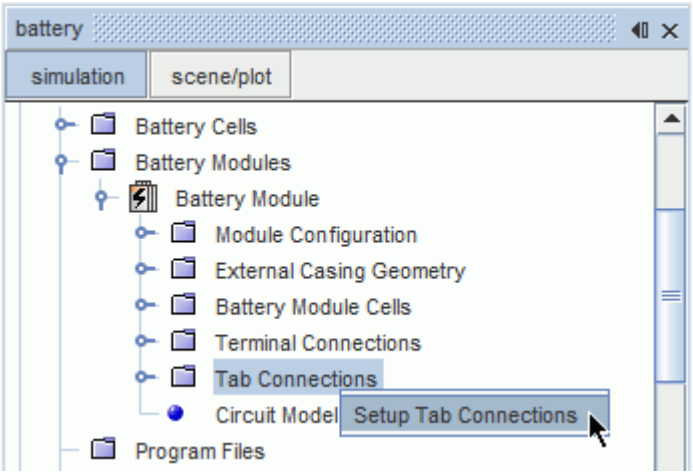
For cylindrical spiral cells, the posts must be in contact with the end plates at the location of the caps (internal posts). This configuration is demonstrated in the following screenshot.



You can prepare the tab connector geometry in an external CAD package, or use the 3D-CAD modeler available in STAR-CCM+. If you use the 3D-CAD modeler, you set up tab connections after generating battery parts.

To set up tab connections:

1. Import the connector geometry into STAR-CCM+.
2. Right-click the **Battery Module** > **Tab Connections** node and select **Setup Tab Connections**.



In the *Setup Tab Connections* panel, you can choose to use the same connector parts for each tab connection, or use different parts for each connection.

The positive tab on the first battery cell is always the open tab; the tab on the Nth cell has the negative tab open. This setup is consistent with the electrical connections that are set in the circuit solver.

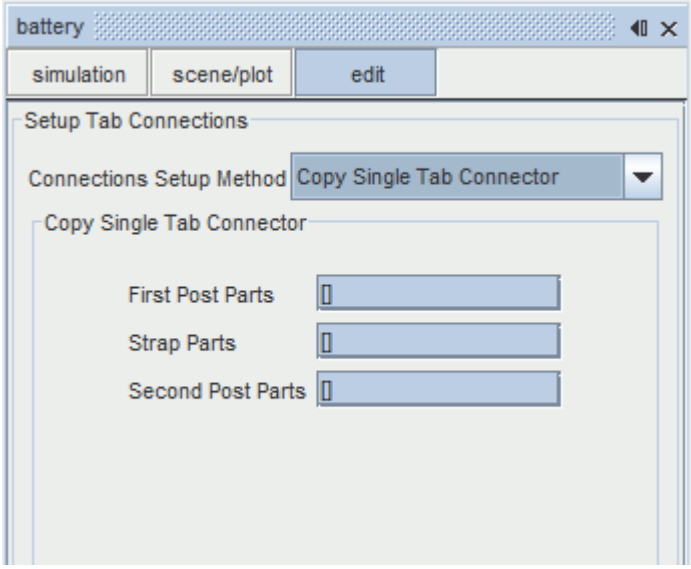
You can check that the tab connectors are placed consistent with the electrical circuit by using the Tab Voltage field function.

Verify that the maximum voltage is at the open positive tab and the minimum voltage within each string is at the open negative tab. Additionally, the voltage that is displayed at tabs which are connected together is nearly identical (that is, the same color), assuming the resistance that is modeled for tab connections is small.

Using a Single Tab Connector

For a battery module whose cells are distributed in a regular pattern, you can import a single tab connector and use it multiple times to connect all the battery cells. In this method, STAR-CCM+ automatically copies and repositions the connector according to the arrangement of cells in the battery module.

To use this method, make sure that the **Connections Setup Method** is set to **Copy Single Tab Connector**.

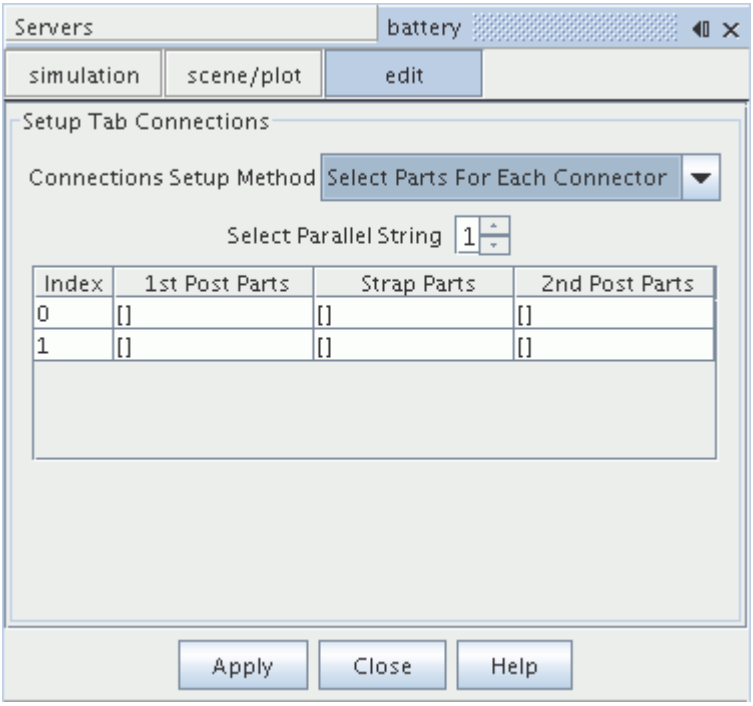


In the **First Post Parts**, **Strap Parts**, and **Second Post Parts** boxes, select the corresponding tab connector parts. When you have selected the parts, click **Apply** and **Close**.

The tab geometry is multiplied and positioned between each tab pair when the tab connector parts are generated.

Selecting Parts for Each Connector

You can use different geometry parts to create each tab connection. This method is useful if the battery cells are not laid out in a regular pattern. To use this method, set the Connections Setup Method to **Select Parts For Each Connector**.



Connector parts are specified for each parallel string separately. To select the first set of connector parts, click the [] button for each option in the table, and select the corresponding post and strap parts. Once you have selected tab connectors parts for each tab connection in the table, increase the value in the Select Parallel String box and define the next set of tab connections. Continue until all tab connections have been defined.

When the settings are complete, click **Apply** and **Close**.

Setting Properties of Tab Connections

Tab connections are stored under the **Battery Module** > **Tab Connections** node. In the properties for each tab connection, you can select the parts that are assigned to the connection. Expanding each tab connection node also allows you to review which components each post and strap is connected to.

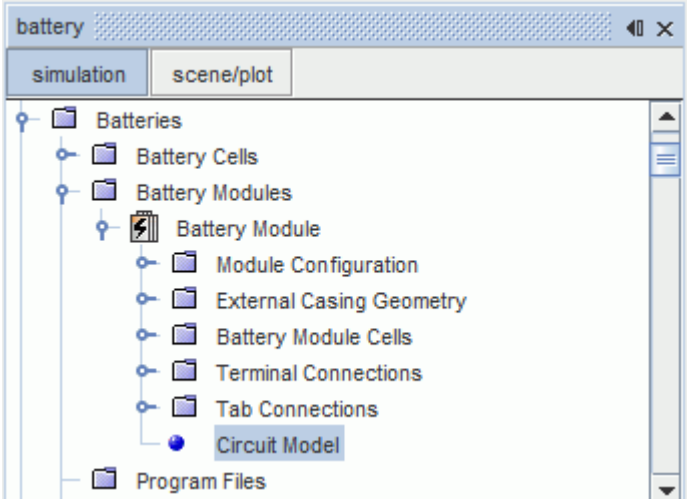
Tab Connection Properties

First Post Parts To Copy	Selects the parts that define the first post.	
Second Post Parts To Copy	Selects the parts that define the second post.	
Strap Parts To Copy	Selects the parts that define the strap.	
Copy Connection Parts	Specifies whether the post and strap parts are copied to other tab connections.	
	Activated	Parts are copied.
	Deactivated	Parts are not copied.

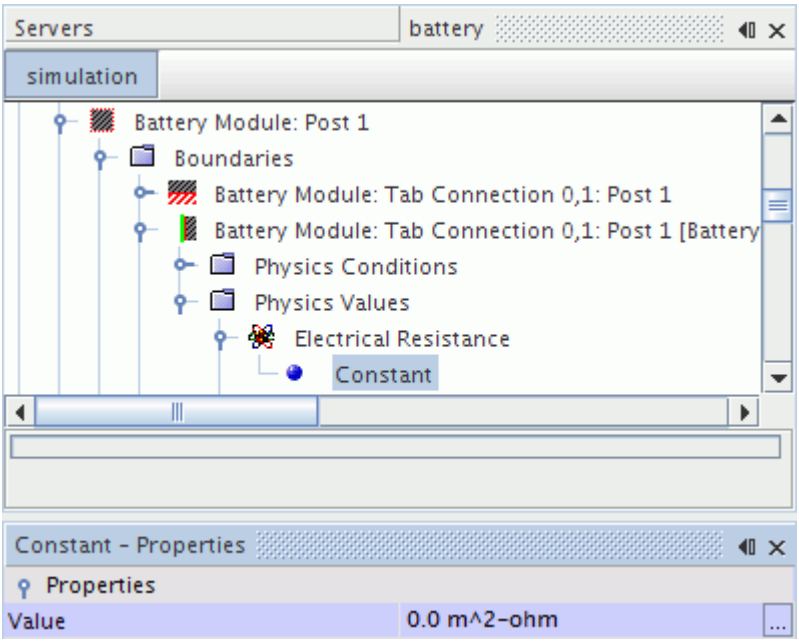
Specifying the Battery Module Resistance Values

The circuit solver allows you to set resistance values for the various connectors that are used within the battery module. By default, the resistance of each connector type is set to 0 Ω.

You can set the resistances on the **Circuit Model** node as shown below.



In a battery, tab stems and tab connectors are welded together. These welds are not directly modeled as part of the geometry. They do not make perfect electrical contact, resulting in some electrical resistance between these parts. You can model this electrical contact resistance using the **Physics Values** > **Electrical Resistance** node of a stem/connector interface. The electrical resistance value is entered as a scalar profile.



The Electrical Resistance option of a contact interface is made available only when the Electrodynamic Potential model is activated.

Circuit Model Properties

Straps Resistance	Sets the resistance of battery module straps.
Positive Terminal Resistance	Sets the resistance of positive terminals.
Negative Terminal Resistance	Sets the resistance of negative terminals.

Positive Posts Resistance	Sets the resistance of positive posts.
Negative Posts Resistance	Sets the resistance of negative posts.

Generating a Battery Module

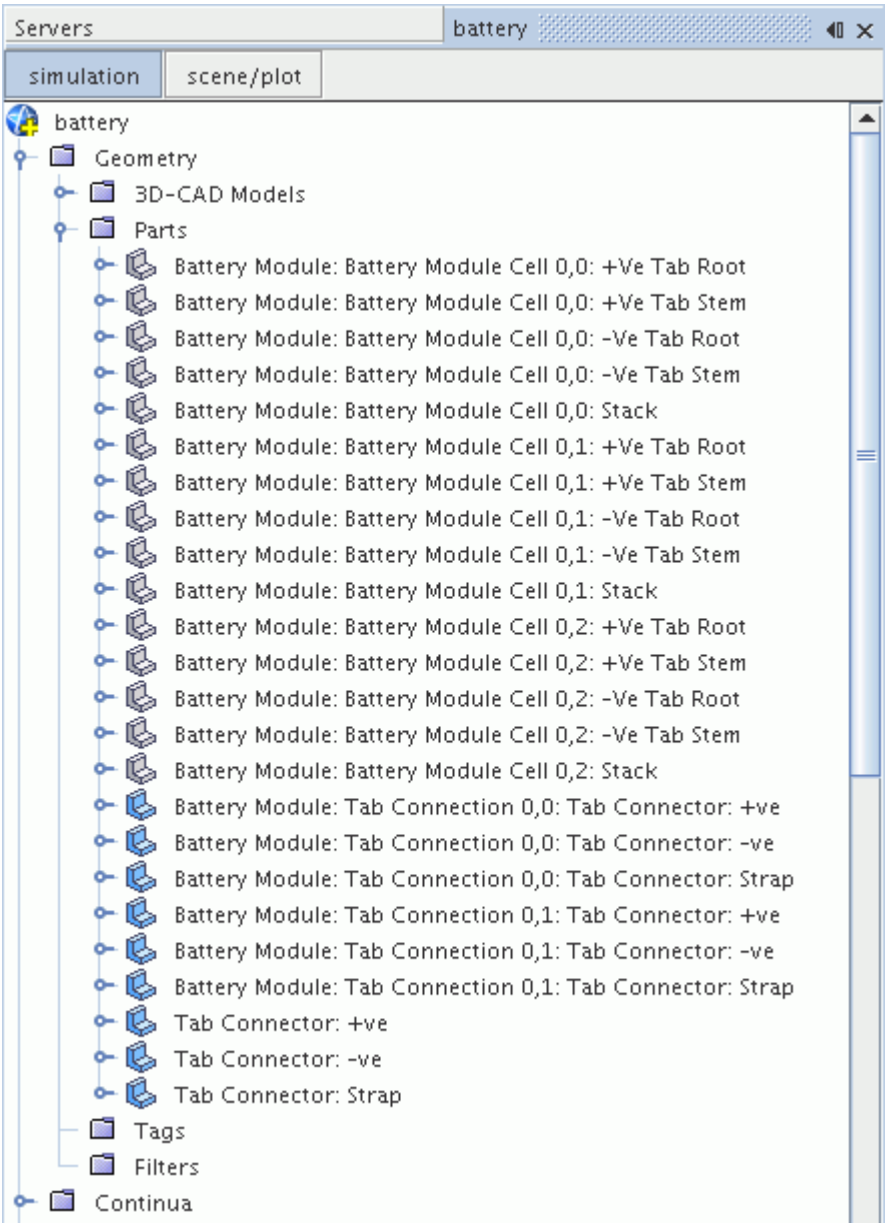
Once the battery module has been fully defined, the next step is to generate the battery module. This process requires creating geometry parts and connector parts for the battery that is based on the layout of the battery cells and module definition, and generating regions from the necessary parts/contacts.

Generating Battery Parts

To generate the battery module parts:

1. Right-click the **Battery Modules** > **Battery Module** node and select **Generate Battery Parts**.

   The **Geometry** > **Parts** node is automatically populated with the battery parts that are designed in BDS. This action also activates the **Generate Connector Parts** and **Generate Regions...** options.



You can use individual parts to visualize results in a scalar scene when you run the simulation. This feature is useful if you wish to view results on a cell in the middle of the module.

2. As with battery module parts, right-click the **Battery Modules** > **Battery Module** node and select **Generate Connector Parts** to generate connector parts, if tab connectors are used.

Before you generate regions, you imprint the parts that are in contact, as well as tab connectors and tabs. The imprinting is required to create all the necessary interfaces. Use the surface repair tool to find and imprint parts in contact, using an appropriate tolerance for your geometry. Furthermore, you can either edit or create new CAD data and merge/imprint your preferred module geometry to the idealized battery cell created by the battery simulation module.

Generating Battery Regions

After completing all the desired modifications to the battery module CAD at the Parts level, battery module regions and interfaces can be set up.

Before you generate the battery module regions, it is recommended that you use the surface repair tool to eliminate any possible surface errors. Before you continue, check that all the regions and interfaces have been properly created.

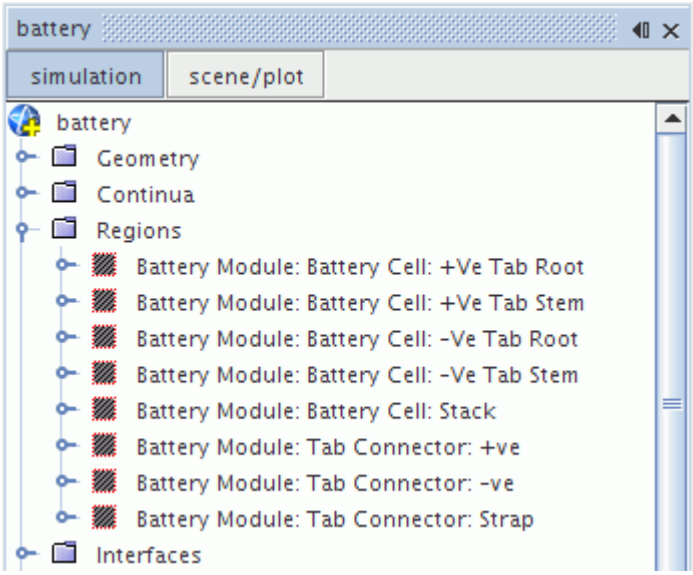
To generate the battery module regions:

- Right-click the **Battery Modules** > **Battery Module** node and select **Generate Regions...**

The **Generate Regions...** action allows you to select the non-battery parts that you want to use to generate regions. You make the selection using the *Additional (non-Battery) Parts to Generate Regions* dialog.

If you did not create or import any non-battery parts, this dialog is empty.

One region is created for each type of battery component, for example all the battery stack parts are assigned to a single battery stack region.



When the battery module regions are generated, STAR-CCM+ recognizes each part of the battery. Where appropriate, interfaces between battery parts are created automatically.

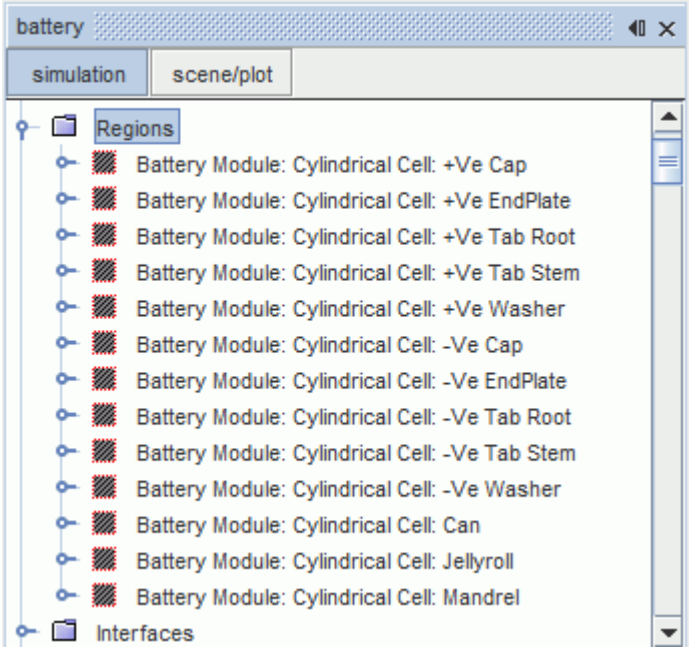
Contents:

Working with Battery Regions

Working with Battery Regions

Battery regions are regions that reference battery objects. They are created when the **Generate Regions...** action is selected in the battery module generation process.

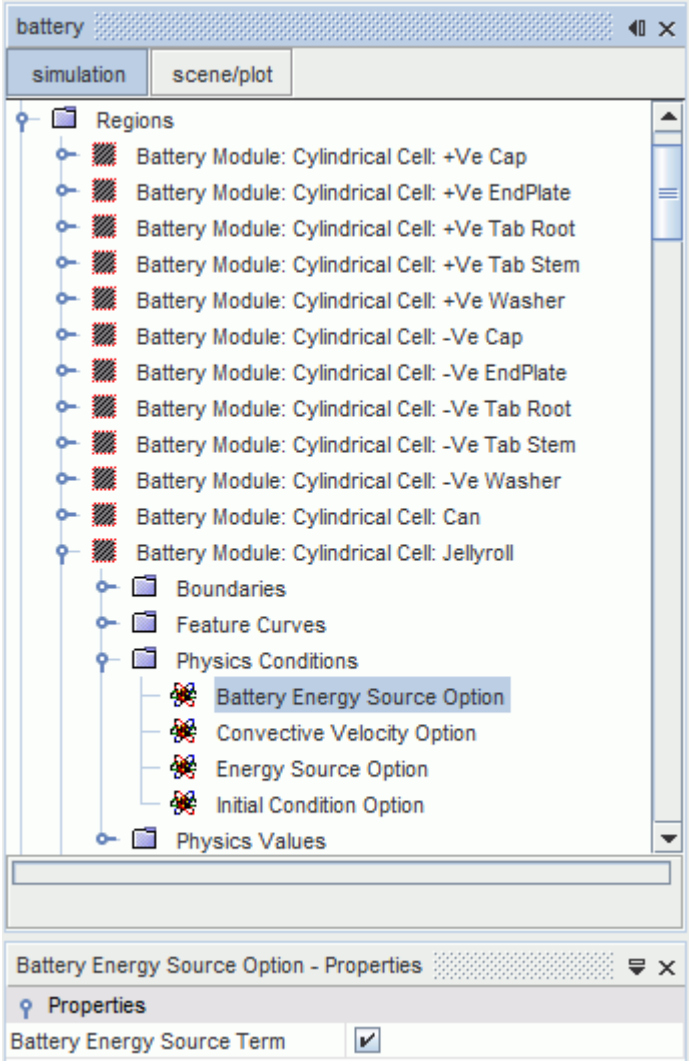
By default, each battery region is defined as a battery energy source. These settings are defined automatically when the regions are generated, and do not need to be changed. Regions that are created for the tab connectors are not defined as battery energy sources.



Defining a Region as a Battery Energy Source

The thermal output of the battery during the load cycle is activated in a battery region using the battery energy source option. This setting is set up automatically on the region when the regions are generated and you do not need to change it. For a region specified as a battery energy source, the energy source type can be specified in the Physics Values for the region.

The **Battery Energy Source Option** node is available in the **Physics Conditions** for the region. It has properties and is subject to common menu items.



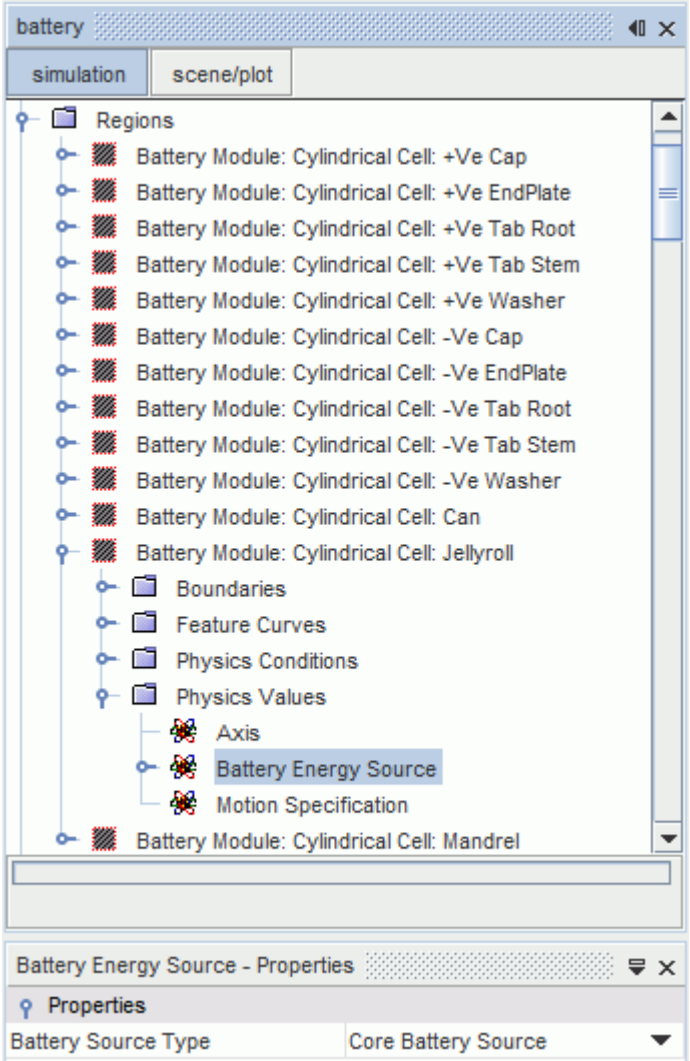
Battery Energy Source Option Properties

Battery Energy Source Term	Specifies whether the region is activated as a battery energy source.	
	Activated	Region is a battery energy source.
	Deactivated	Region is not a battery energy source.

Selecting a Battery Energy Source Type

The battery energy source type can be specified for a region that is activated as a battery energy source. STAR-CCM+ sets this option automatically when the battery module regions are generated and you do not need to change it.

The **Battery Energy Source** node is available in the **Physics Values** for the region. In the properties for the **Battery Energy Source** node, you can specify whether the region is a Core or Tab energy source. The Core option is used for the battery stack, while the Tab option is used for the tab roots and stems.



Battery Energy Source Properties

Battery Source Type	Specifies whether the region is a core or tab battery energy source.

Generating a Battery Mesh

The thin parts of a battery require specific meshing models and associated settings. The STAR-CCM+ battery simulation module provides an option for automatically creating a mesh continuum that contains the models required to mesh a battery.

The mesh continuum contains:

- Thin Mesher and Surface Remesher models.

  The thin mesher recognizes thin parts of geometries and produces prismatic cells within the thin parts. For non-thin parts of the geometry, the thin mesher automatically changes to produce polyhedral or tetrahedral cells. The mesh remains conformal.

  You can control the relative or absolute thickness of the thin parts by activating the Customize Thickness Threshold property.
- Reference values suitable for battery meshing.

  You can keep the default reference values in the battery mesh continuum or change them if required.
- A Volumetric Control.

  The volumetric control is created for the battery tabs and tab connectors with a reference value of 7.5% of the surface base size.

The typical steps for generating a battery mesh are:

1. Set up battery meshing.
2. Create mesh continua for non-battery parts.
3. Generate the volume mesh.

Contents:

Setting Up the Battery Mesh Continuum

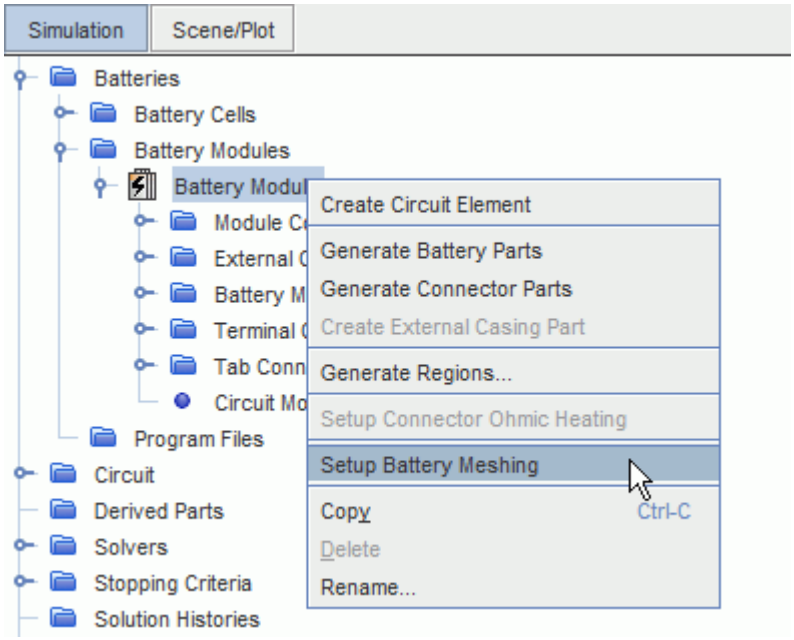
Setting Up the Battery Mesh Continuum

The battery mesh continuum is assigned to the battery module region by default. You can change the pre-defined battery settings as necessary.

Note: You can use the **Mesh** > **Scale Mesh** option to scale the battery mesh. However, the scaling operation can affect the results, so it is recommended that you use this option only when essential.

To set up a battery mesh:

1. Right-click the **Batteries** > **Battery Modules** > **Battery Module** node and select **Setup Battery Meshing**.



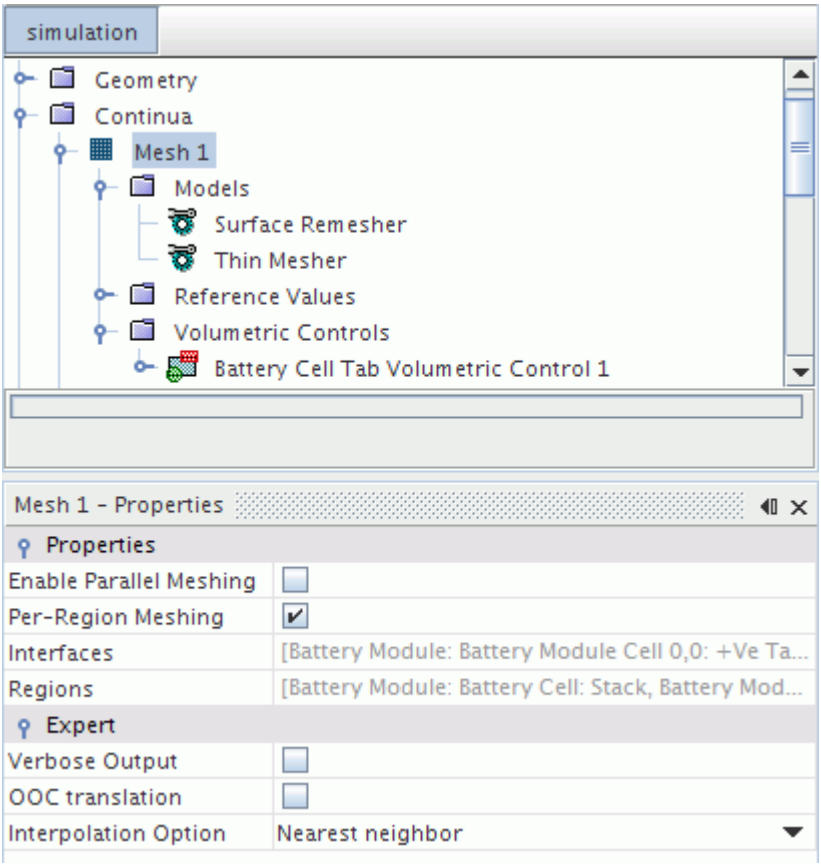
A new **Mesh 1** node appears under the **Continua** manager node. When a battery mesh continuum is created, it is assigned to a battery module region by default.

The battery mesh continuum contains the **Surface Remesher** and **Thin Mesher** models as well as a volumetric control for the tab parts.

To view the meshing models and volumetric control:

2. Expand the **Continua** > **Mesh 1** > **Models** node.
3. Expand the **Mesh 1** > **Volumetric Controls** node.

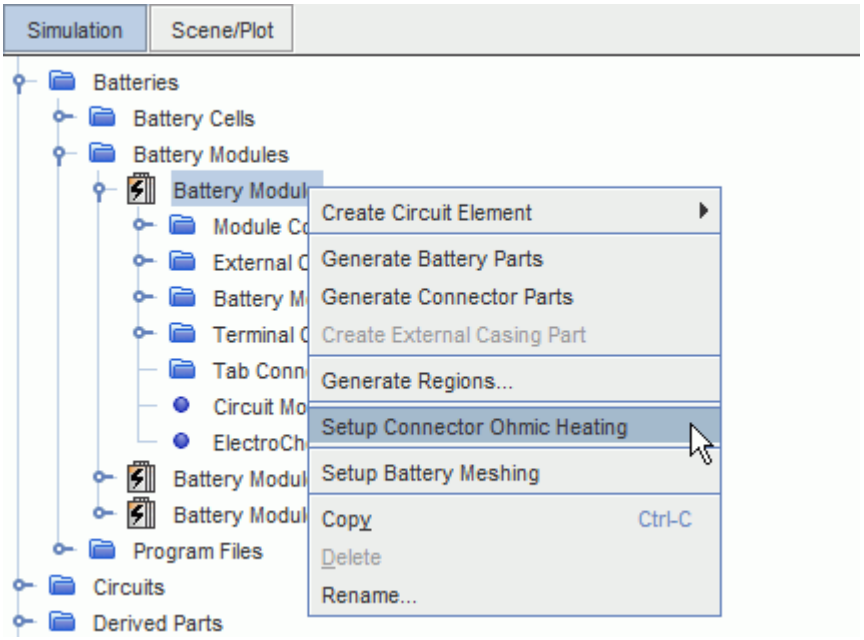
   The nodes are listed in the tree as shown below:


simulation
Geometry
Continua
Mesh 1
Models
Surface Remesher
Thin Mesher
Reference Values
Volumetric Controls
Battery Cell Tab Volumetric Control 1
Mesh 1 - Properties
Properties
Enable Parallel Meshing
Per-Region Meshing
Interfaces
[Battery Module: Battery Module Cell 0,0: +Ve Ta...
Regions
[Battery Module: Battery Cell: Stack, Battery Mod...
Expert
Verbose Output
OOC translation
Interpolation Option
Nearest neighbor

Setting Up Connector Ohmic Heating

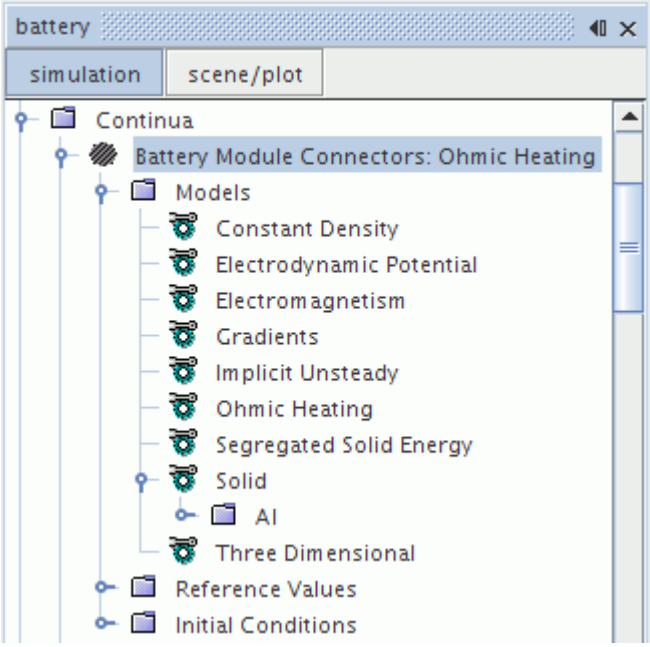
The Ohmic Heating model in tab connector regions can be set automatically for the battery module. The model accounts for the ohmic heating effect in the tab connectors when a current passes through them.

The ohmic heating for the connectors is defined using the **Battery Module** > **Setup Connector Ohmic Heating** option. This option becomes available once a battery mesh has been generated.

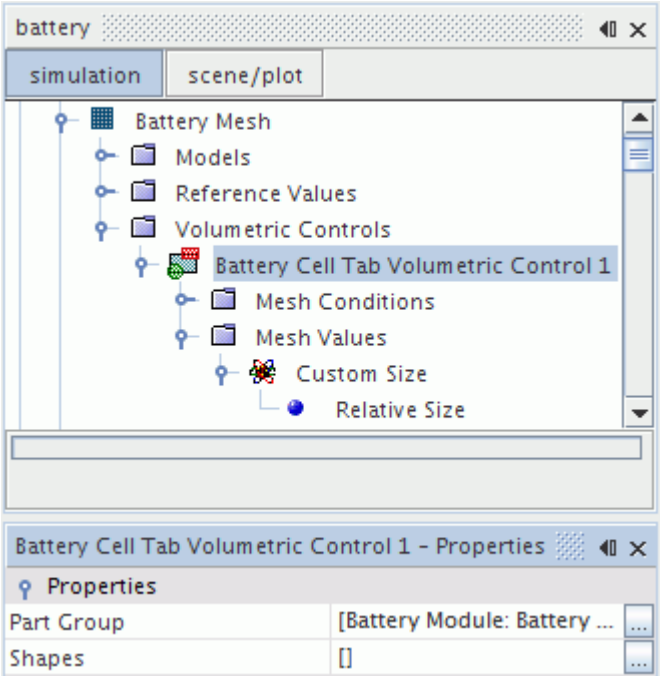


When ohmic heating in connectors is selected:

1. STAR-CCM+ creates a physics continuum, **Battery Module Connectors: Ohmic Heating**, for the Ohmic Heating model and assigns it to the tab connector regions.

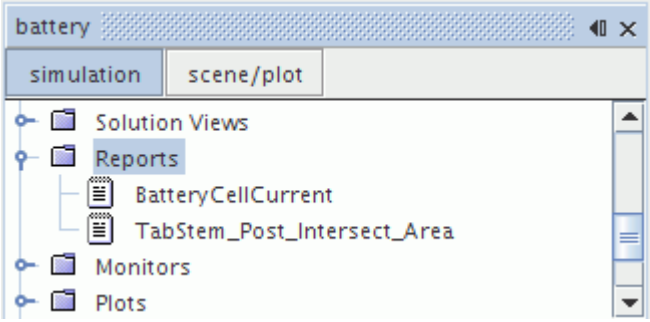


2. STAR-CCM+ creates a Volumetric Control for the tabs and connectors with a default value of 4% of the surface base size. We recommend that you keep the size of the mesh on the tabs and connectors at roughly 4% of the tab stem height. The refinement of the mesh in the connectors helps with the convergence of the energy solver when the Ohmic Heating model is turned on.

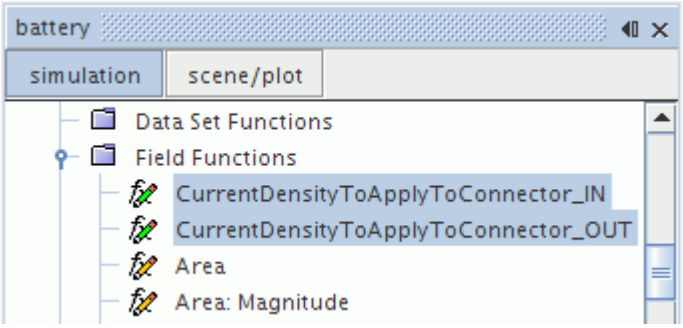


3. STAR-CCM+ creates two reports for the current and interface surface area: **BatteryCellCurrent** and **TabStem_Post_Intersect_Area**. The battery cell current report is a volume average report that collects the current going through the cell. The battery module stack or the jellyroll region is automatically

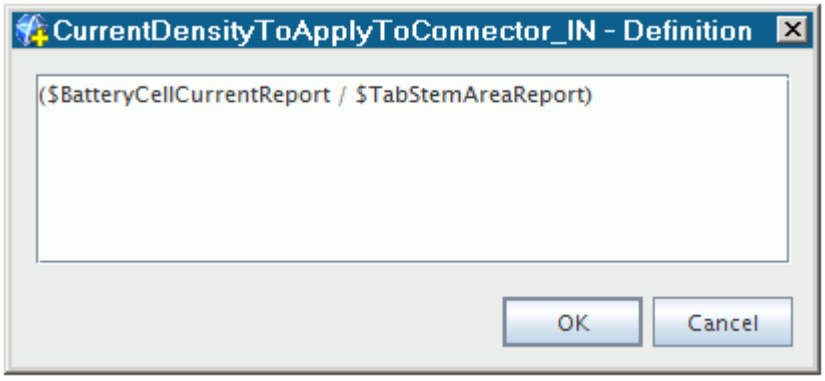
assigned to it. The tab stem/post intersect area report returns the area of the intersection between the stem and the post for the stacked cell, and end plate and post for the cylindrical cell.



4. STAR-CCM+ creates current density field functions from the reports: **CurrentDensityToApplyToConnector_IN** and **CurrentDensityToApplyToConnector_OUT**. The current flows into the post for the negative tab and flows out of the post for the positive tab. The field functions are automatically applied to the appropriate interfaces.



The two field functions compute the current density at the interfaces, with one for current flowing in and the other for current flowing out. These values are of equal magnitude but opposite sign. The definition for the positive current density is shown below. The convention is to have the positive current density represent the current flowing in.



Contents:

Improving Convergence of the Segregated Energy Solver

See Also:

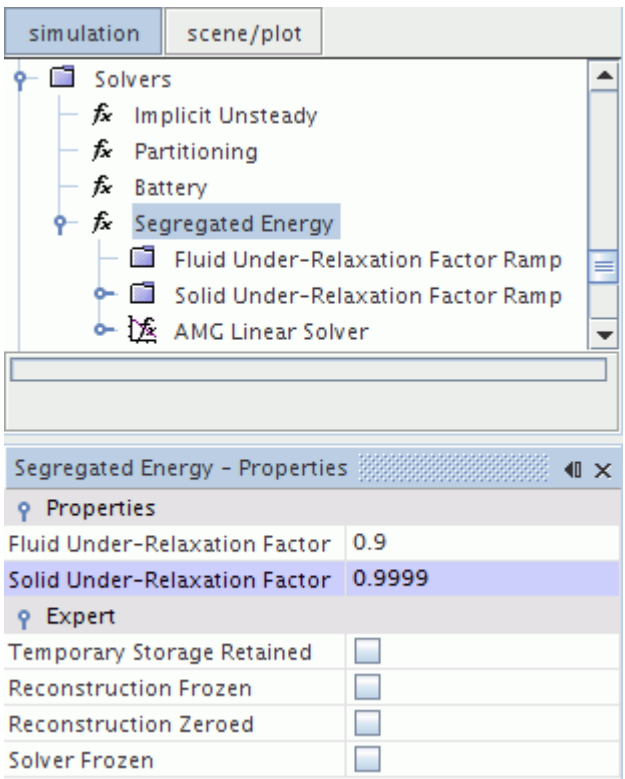
Generating a Battery Mesh

Improving Convergence of the Segregated Energy Solver

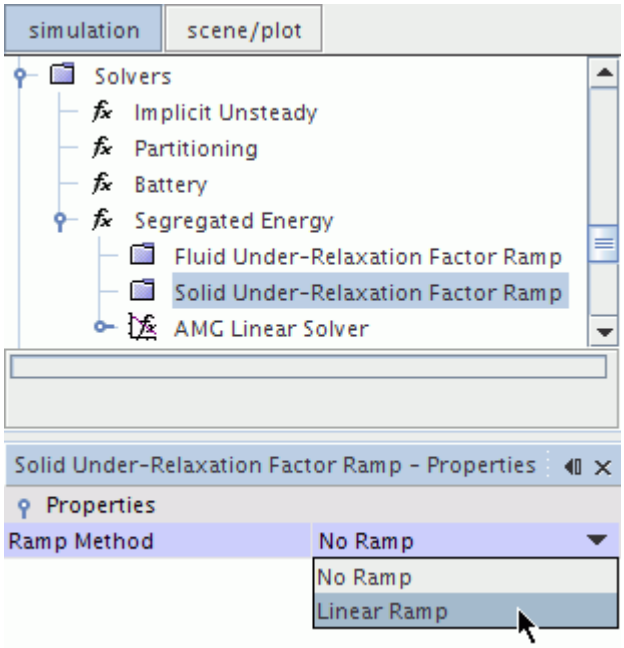
As a result of non-converged initial electric potential, spurious heat is introduced into the Segregated Energy solver in the first iterations of an analysis. This behavior can lead to high temperatures in intermediate non-converged states.

To improve convergence of the Segregated Energy solver:

1. Set a *Solid Under-Relaxation Factor* of `0.9999` for the **Segregated Energy** solver.

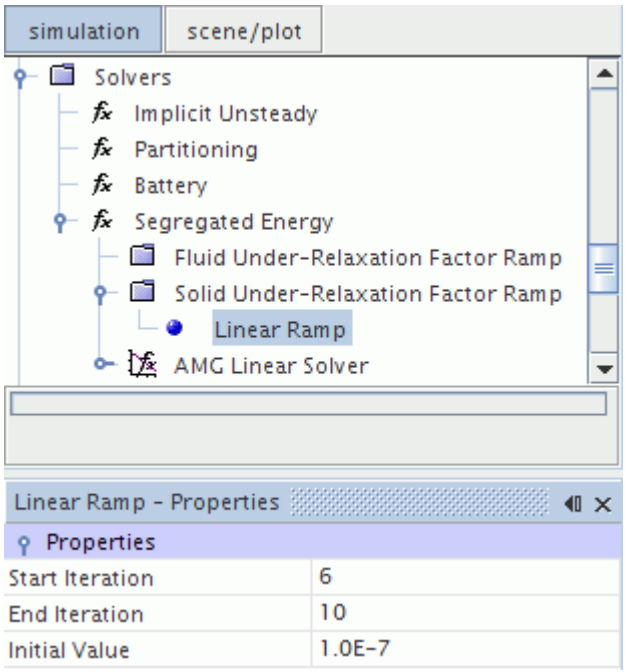


2. Set a solid under-relaxation factor **Linear Ramp**.



3. Set the *Linear Ramp - Properties* as follows:

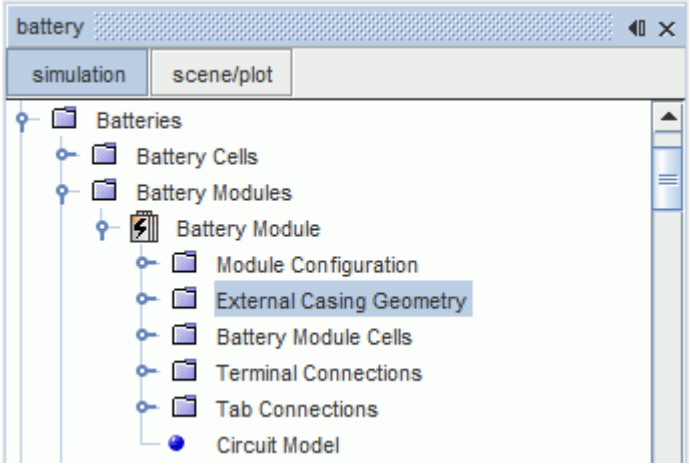
Start iteration	6
End iteration	10
Initial value	1.0E-7



Defining External Battery Casing

The external battery casing feature lets you automatically create a volume to fill the void between the battery parts and the surface of the part that you use for the casing. The casing part can be an external bounding box, or it can be a part that you specify, depending on the selected casing option.

External battery casing is specified using the **Battery Module** > **External Casing Geometry** node. To reveal this node in the simulation tree, you generate battery parts.



The external casing requires a separate mesh continuum from the mesh continuum that is assigned to the battery module. The Battery Mesher can only be used for battery objects. If you are using tab connectors with thin geometry, add extra volumetric controls for the connectors inside the external casing mesh continuum.

Contents:

Selecting a Casing Option
Defining a Bounding Box
Selecting a Casing Part
Creating External Battery Casing

See Also:

Generating a Battery Module
Setting Up the Battery Mesh Continuum

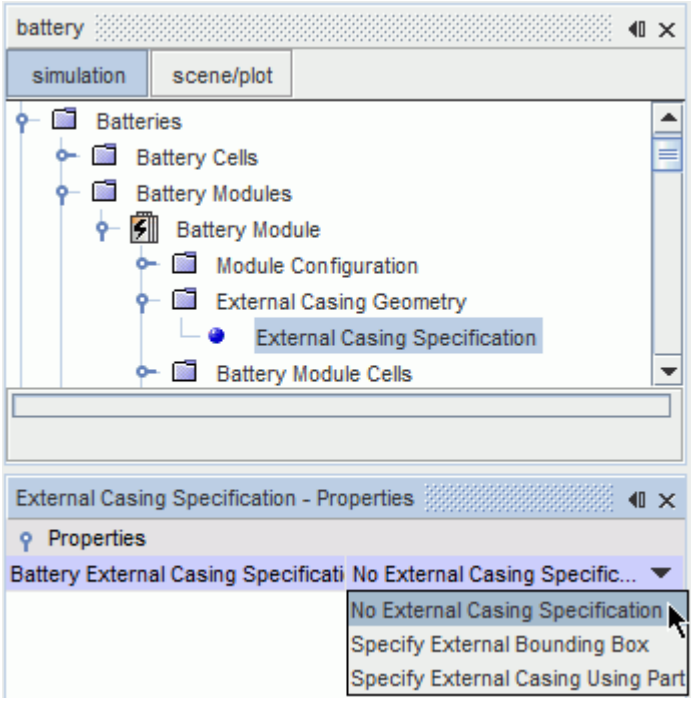
Selecting a Casing Option

This section outlines the methods for specifying the external casing.

The external casing option is selected using the **Battery Module** > **External Casing Geometry** > **External Casing Specification** node.

There are two methods for specifying the casing:

- Create a basic box around the module
- Use a geometry part that defines the casing



Selecting one of the options for the *Battery External Casing Specification* property allows you to define the size and shape of the external casing. Once you have fully defined the casing, you can create the external casing part.

See Also:

Creating External Battery Casing

Defining a Bounding Box

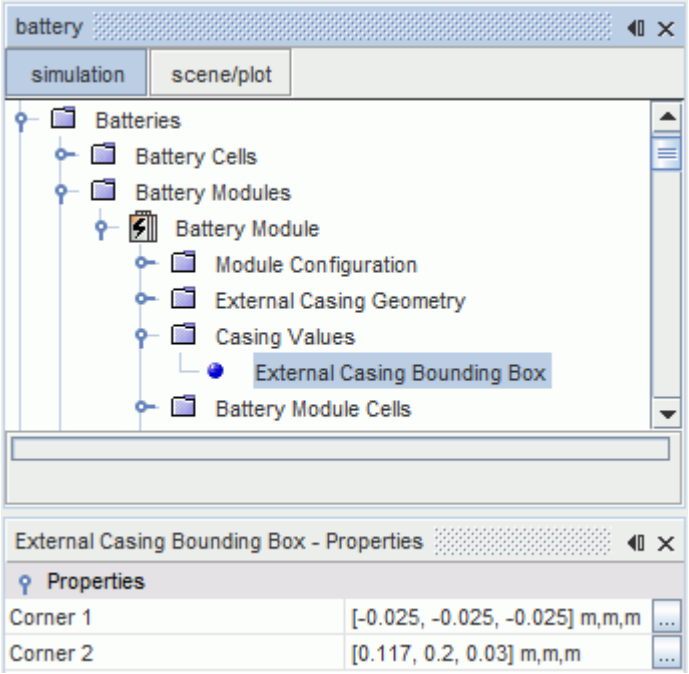
Selecting a Casing Part

Defining a Bounding Box

This section describes how to create a bounding box around the battery module.

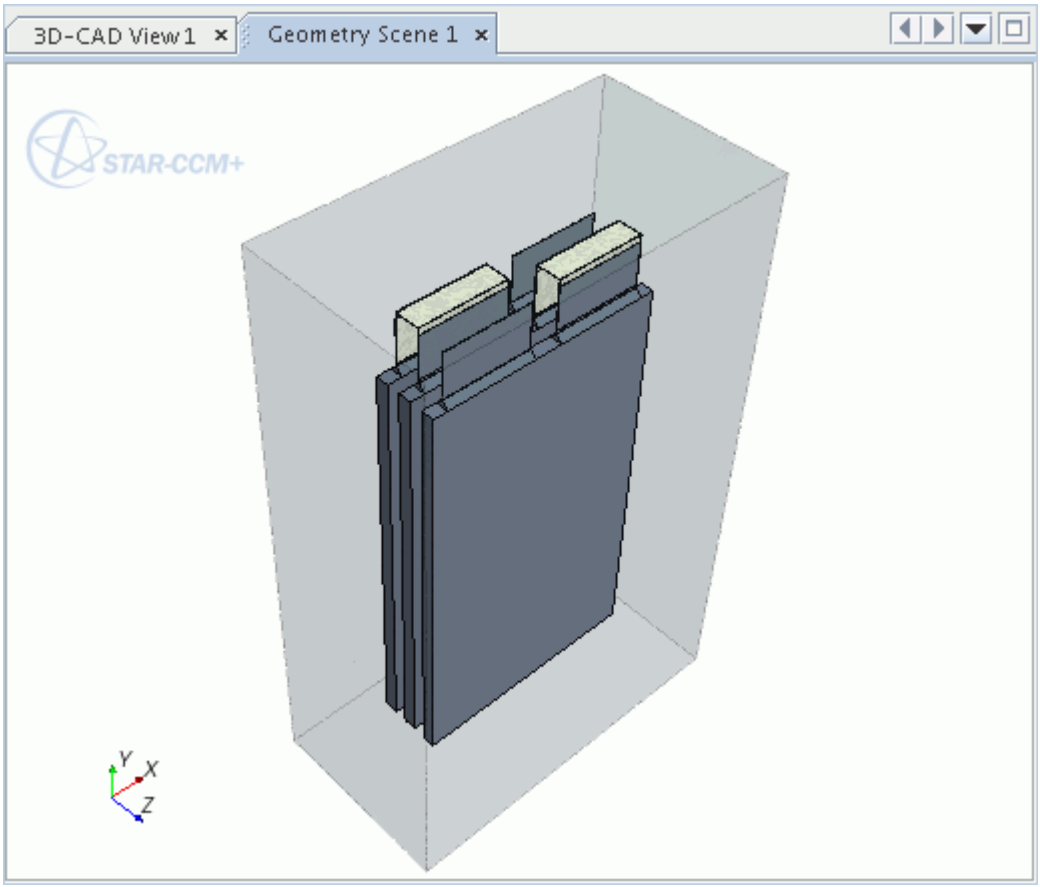
To define a bounding box:

1. Select the **Specify External Bounding Box** casing option.
2. Select the **Battery Module** > **Casing Values** > **External Casing Bounding Box** node and set the position of the first and second corners of the box properties.

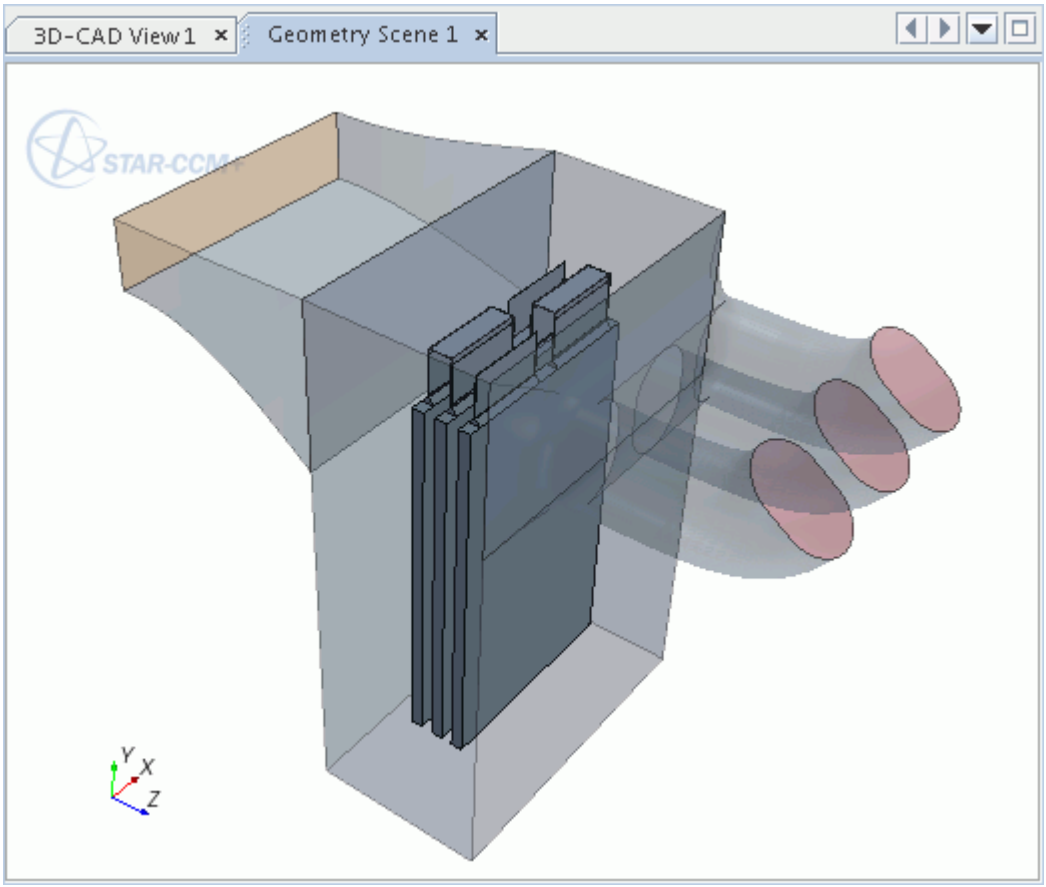


After defining the bounding box, the next step is to create the external casing part.

An example of a bounding box around a battery consisting of three series cells in a single parallel string is shown below.



The bounding box that is created around the battery module has a single external surface. If you wish to define one surface of the box as an inlet and another as an outlet, for example, you can split the surface using the standard techniques. If necessary, you can define an inlet and an outlet by importing separate geometry parts. These parts can be interfaced with the bounding box before meshing, allowing you to create geometry such as that shown below.



Selecting a Casing Part

You can create external battery casing that is a specific shape by using a geometry part.

To select a casing part:

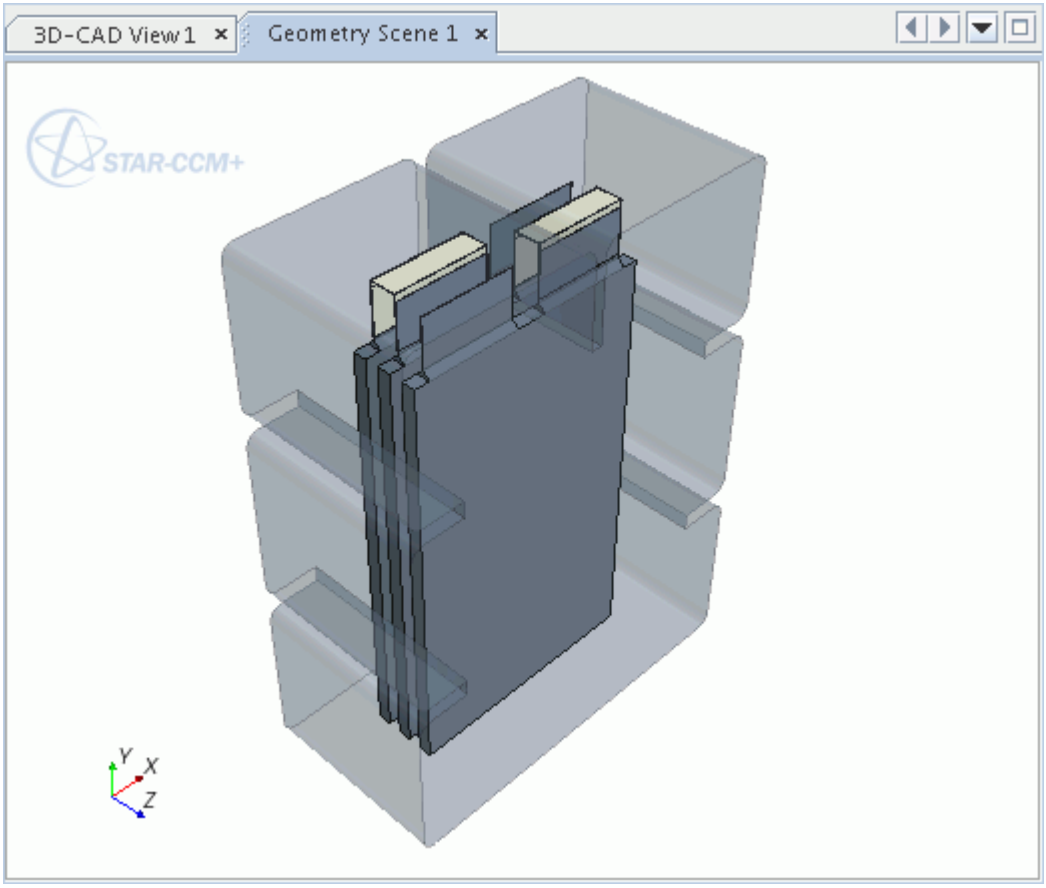
1. Create the geometry of the external casing; one method for creating the geometry is to use the embedded CAD modeler in STAR-CCM+, 3D-CAD.

   The casing part is a single volume. You do not need to subtract the battery module from the casing volume; STAR-CCM+ subtracts the module automatically when you create the casing part.
2. If the external casing part is created in a different simulation, import the casing geometry into the battery simulation.

   Make sure that the *Import Mode* is set to **Create new Part**.
3. Select the **Specify External Casing Using Part** casing option.
4. Select the **Battery Module** > **Casing Values** > **External Casing Part** node and set the **External Casing Part** property to the casing part.

After selecting the part that you want to use as the battery module casing, the next step is to create the external casing part.

An example of a custom external casing around a battery consisting of three series cells in a single parallel string is shown below.

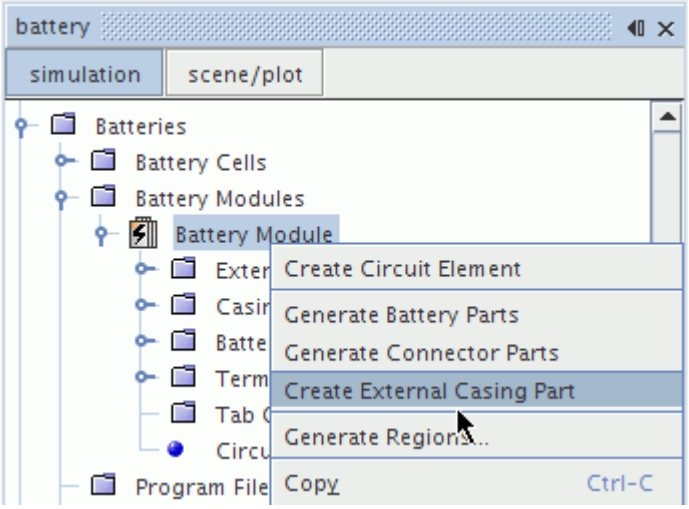


Creating External Battery Casing

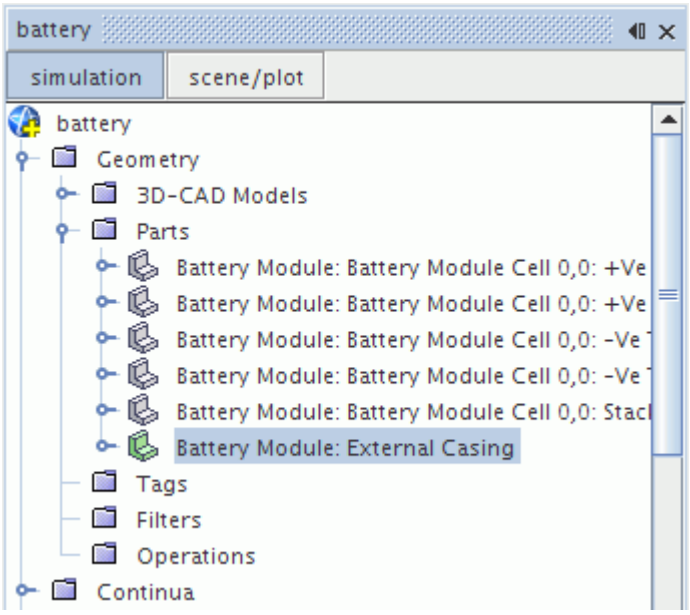
This section describes how to create an external battery casing.

To create an external casing:

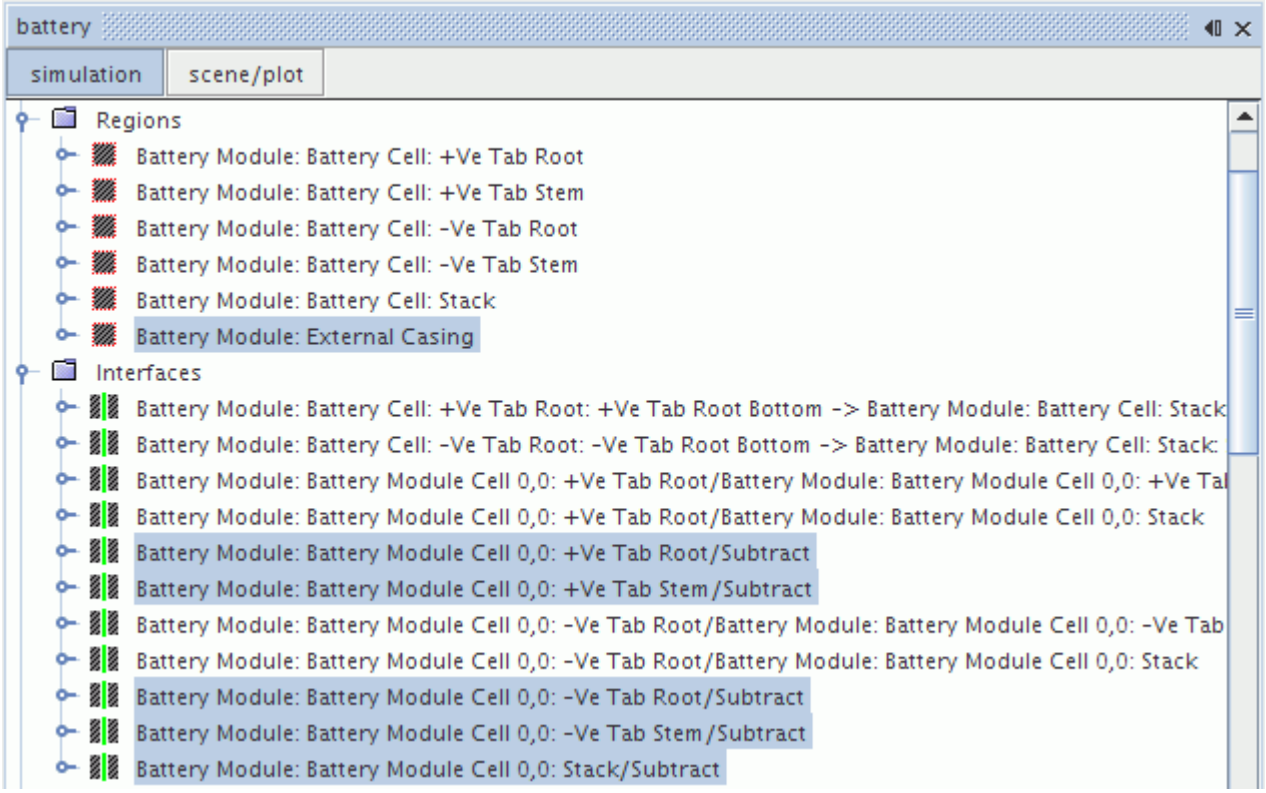
- Right-click the **Battery Module** node and select **Create External Casing Part**.



During the process of creating external casing, STAR-CCM+ creates a bounding box, or takes the selected casing part, and subtracts the battery module parts from it. The resulting part is named **Battery Module: External Casing**.



STAR-CCM+ also generates a region for the external casing during the generate regions step. An example of the regions and interfaces that are created for a single cell battery module with an external casing is shown in the following screenshot.



The final step in setting up the external casing is to define the mesh settings for the external casing region in a separate mesh continuum. It is only possible to use the battery mesh continuum for battery regions, such as the regions for the stack, tab stems, and tab roots. When setting up the mesh continuum for the external casing region, a good technique is to use volumetric controls to refine the mesh around the tab parts, including root, stems, posts, and straps. This technique promotes a good quality mesh around the thin tab and connector parts.

Defining an Electrical Load

This section describes how to define an electrical load.

You can define the electrical load on the battery in the following ways:

- program file (.prg)

  Specifies a load for a particular duration. For example, 10 amps for 10 seconds, then 8 amps for the next 20 seconds, and then 6 amps for the next 30 seconds.
- table

  Each table row specifies a load at a particular time. The interval between specified time values can be flexible: you do not need to specify values at regular time intervals. However, the load values are linearly interpolated between the specified time values to provide data at the time step of the thermal solver.

  Because of this interpolation, using a table to define a load can provide more flexibility than using a program file.
- scalar function

  Specifies a load as a scalar function. The scalar function can be a constant value or an expression which is evaluated over time. The expression can use field functions that you define.

Once you have created the program file in BDS or a table, you can import the load into STAR-CCM+ and connect it to the battery.

Contents:

Using a Program File
Using a Table
Using a Scalar Function

See Also:

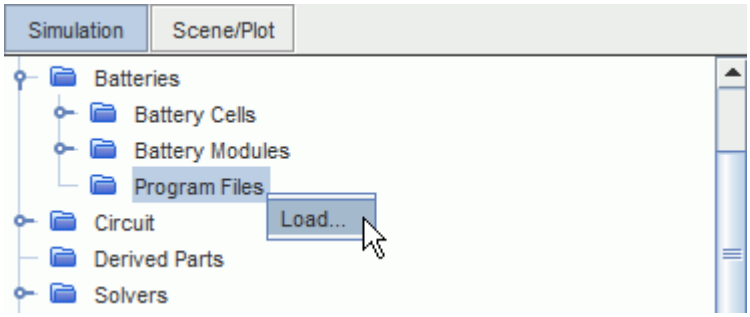
Defining Electrical Circuits
What Is a Thermal Solver?

Using a Program File

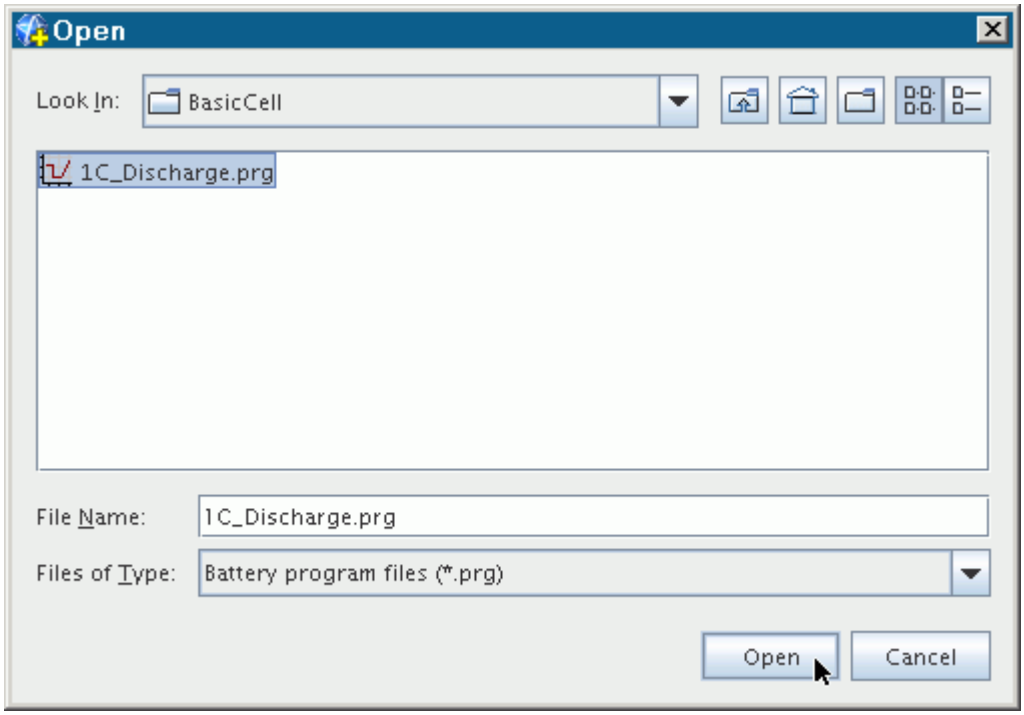
This section describes how to use a program file to define an electrical load.

If you are using a program file:

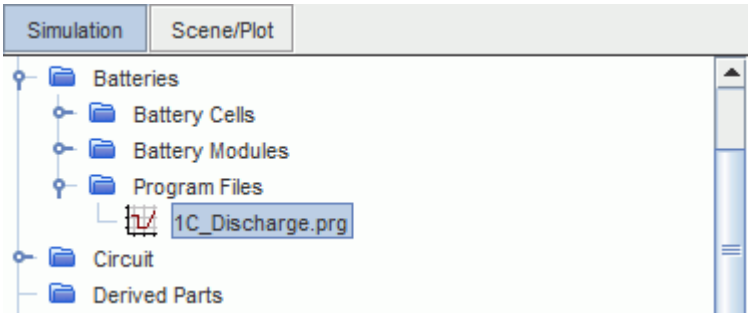
1. Right-click the **Batteries** > **Program Files** node and select **Load...**



2. In the *Open* dialog, select the program file and click **Open**.



A program file is added under the **Program Files** manager node. The **Program Files** node has properties and a pop-up menu. You can edit the program file using the *Program Data* property.



Currently not all the functionality for end and limit conditions is available in STAR-CCM+.

- STAR-CCM+ fully supports **Time** end conditions. **Voltage** end conditions are partially supported.
- STAR-CCM+ uses **Voltage** limit conditions to specify a warning message.

Program File Pop-Up Menu

The following options are available when you right-click on a program file node.

Create Circuit Element	Creates a circuit element, which is used to connect the electrical load to the battery module.
Reload...	Reloads the program file data into the simulation.
Export...	Exports program file data to a .prg file.

Program File Properties

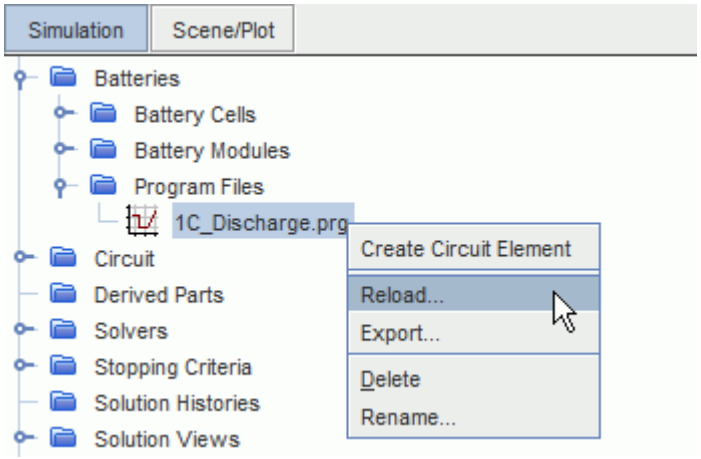
File name	Displays the path to the .prg file.
Program Data	Displays the data that is contained in the .prg file.

Reloading Program Data

If you have changed the program data outside of the simulation, for example, by modifying the .prg file in BDS, you must reload the data into STAR-CCM+. To reload the data, the name and path of the .prg file must be the same.

To reload the program data:

- Right-click the **Program Files** > **<filename>.prg** node and select **Reload...**



STAR-CCM+ rereads the contents of the .prg file, and loads the data into the simulation. The progress of the reload operation is displayed in the *Output* window.

Exporting a Program File

You can export data from a simulation to a .prg file. Exporting the file allows you to recreate the .prg file from the program data that is stored within the simulation.

To export program file data to a .prg file:

- Right-click the program file node and select **Export...**

The *Save* dialog allows you to save the data in a .prg file.

See Also:

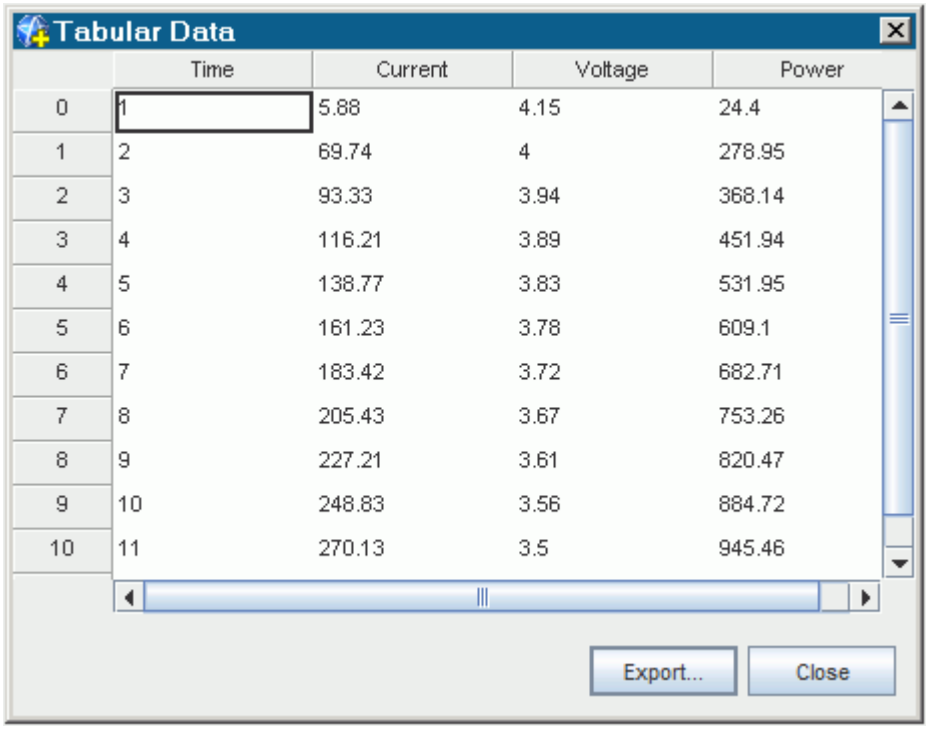
Preparing Program (.prg) Files

Using a Table

As an alternative to using a program file, you can use a table to define the electrical load. The table must have a time column, which is interpreted as the independent variable, and a load column, specifying current, power, or voltage.

You can define the units in the column heading using parentheses, for example Time (min) or Power (kW). STAR-CCM+ converts these units to the default SI units: seconds for time, amps for current, watts for power, and volts for voltage. If no units are specified, the values are imported in their default units. The load values are linearly interpolated to provide data at the time steps that the thermal solver requires.

Below is an example of a table for the electrical load. Although it shows three different load columns, only one such column is needed.



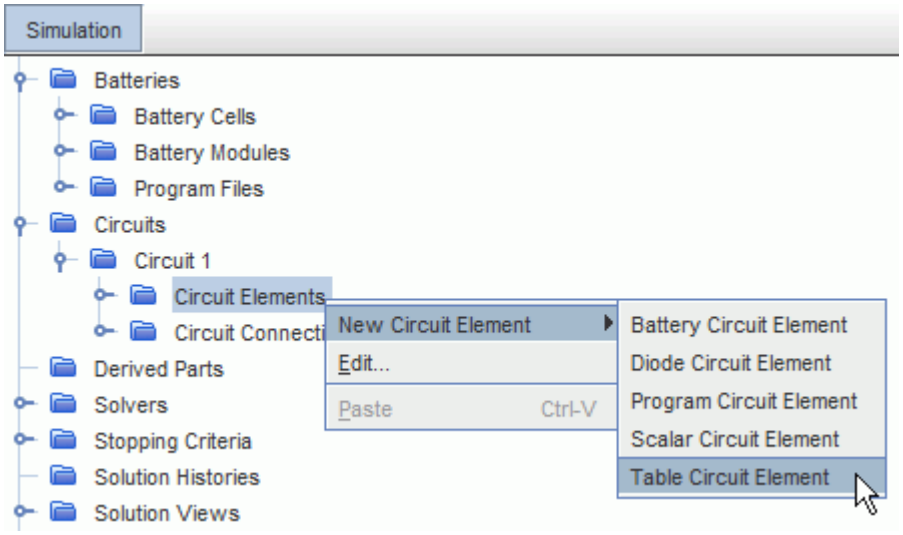
Tabular Data

	Time	Current	Voltage	Power
0	1	5.88	4.15	24.4
1	2	69.74	4	278.95
2	3	93.33	3.94	368.14
3	4	116.21	3.89	451.94
4	5	138.77	3.83	531.95
5	6	161.23	3.78	609.1
6	7	183.42	3.72	682.71
7	8	205.43	3.67	753.26
8	9	227.21	3.61	820.47
9	10	248.83	3.56	884.72
10	11	270.13	3.5	945.46

Export... Close

To specify the electric load using a table:

1. Load a table using the **Tools** > **Tables** node.
2. Right-click the **Circuits** > **[Circuit Name]** > **Circuit Elements** node and select **New Circuit Element** > **Table Circuit Element**.



STAR-CCM+ adds a **Table Circuit Element** node to the **Circuit Elements** node.

3. In the *Properties* window of the **Table Circuit Element** node, select a load table and use the column headings of the table to set the load for the circuit solver.

Table Circuit Element Properties

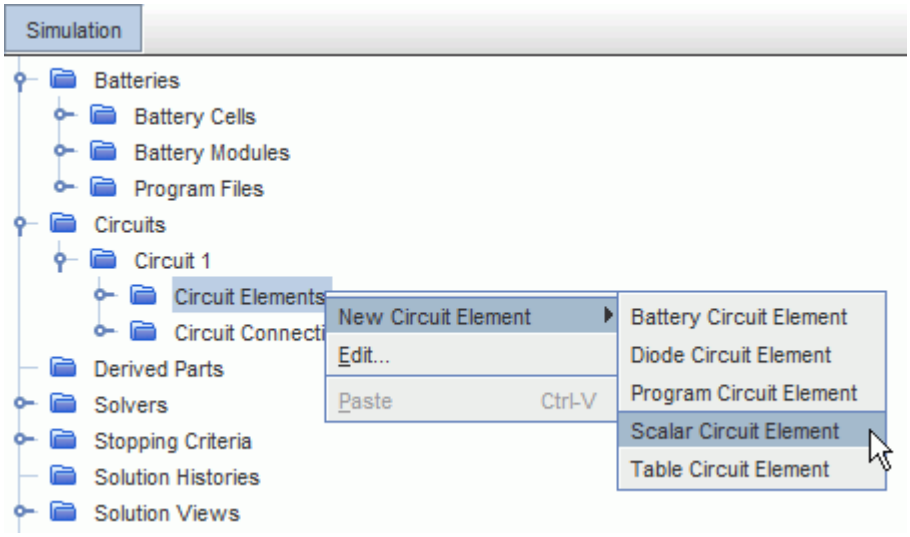
Table	Sets the table that is used for the load.	
Time Data	Specifies the time column of the table.	
Load data	Specifies the load column of the table.	
Load Type	Specifies the load type that is read in.	
	Current	The load is current load.
	Power	The load is power load.
	Voltage	The load is voltage load.
	Resistance	The load is a resistor load.
	Inductance	The load is an inductor load.
	Capacitance	The load is a capacitor load.

Using a Scalar Function

As an alternative to using a program file, you can use a scalar function define the electrical load. The scalar function can be a constant value or an expression which is evaluated over time. The expression can use field functions that you define.

To specify the electrical load using a scalar function:

1. If necessary, set up the appropriate field functions using the **Tools** > **Field Functions** node.
2. Right-click the **Circuits** > **Circuit 1** > **Circuit Elements** node and select **New Circuit Element** > **Scalar Circuit Element**.



   STAR-CCM+ adds a **Scalar Circuit Element** node to the **Circuit Elements** manager node.
3. In the *Properties* window of the **Scalar Circuit Element** node, specify the scalar function and the load type.

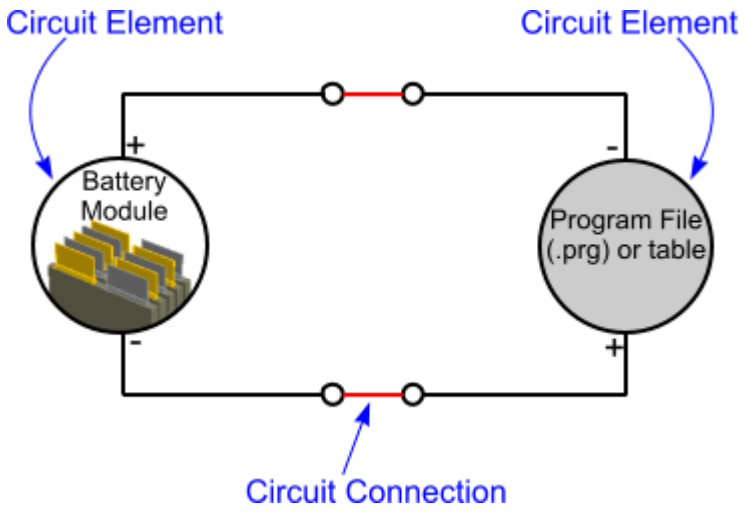
   Scalar Circuit Element Properties

Load	Specifies the scalar function that is used for the load.	
Load Type	Specifies the load type of the scalar function.	
	Current	The load is current load.
	Power	The load is power load.
	Voltage	The load is voltage load.
	Resistance	The load is a resistor load.
	Inductance	The load is an inductor load.
	Capacitance	The load is a capacitor load.

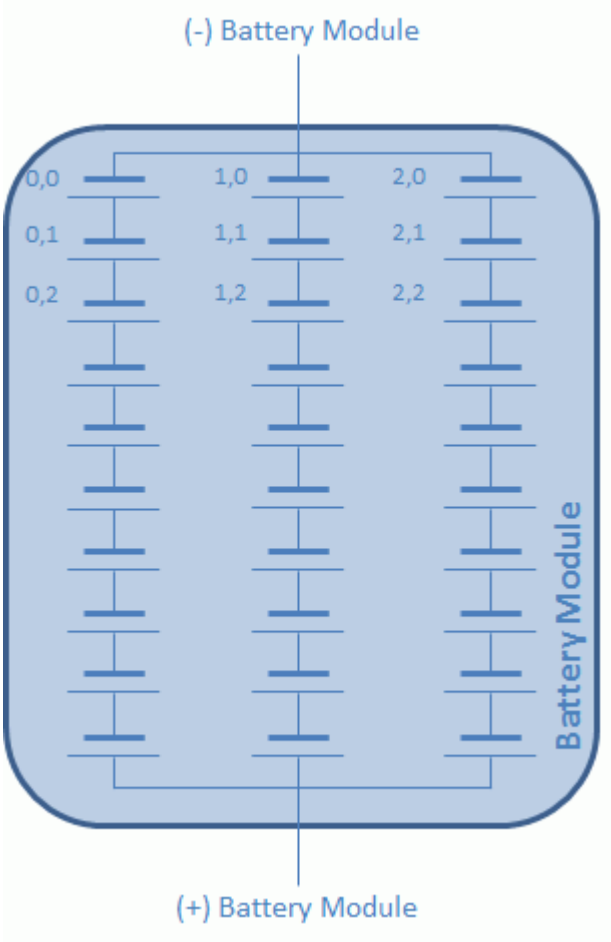
Defining Electrical Circuits

To simulate the response of a battery to an applied electrical load, you connect the battery and the load in an electrical circuit. When the simulation is run, the Circuit Solver computes the electrical solution for the circuit.

An electrical circuit is formed from elements and connections, as shown in the diagram below.



The internal structure of the battery module is illustrated below.



You can create multiple circuits and set the properties independently for each circuit. Using multiple circuits lets you simulate coupled circuits, such as in a transformer-type environment where the change in magnetic energy in one circuit is the voltage source for another circuit. You also use multiple circuits when you have multiple batteries in the same thermal environment, where each battery has its own load.

The elements and connections for each circuit are stored in the corresponding **[Circuit Name]** node under the **Circuits** node.

To define an electrical circuit:

1. If necessary, create a circuit.

   The **Circuits** > **Circuit 1** node is created automatically when you activate the Battery model or the Circuit Model in the physics continuum. To add additional circuits, right-click the **Circuits** node and select **New**.
2. Set the circuit properties. These settings are used by the Circuit Solver.
3. Add the circuit elements that you require.

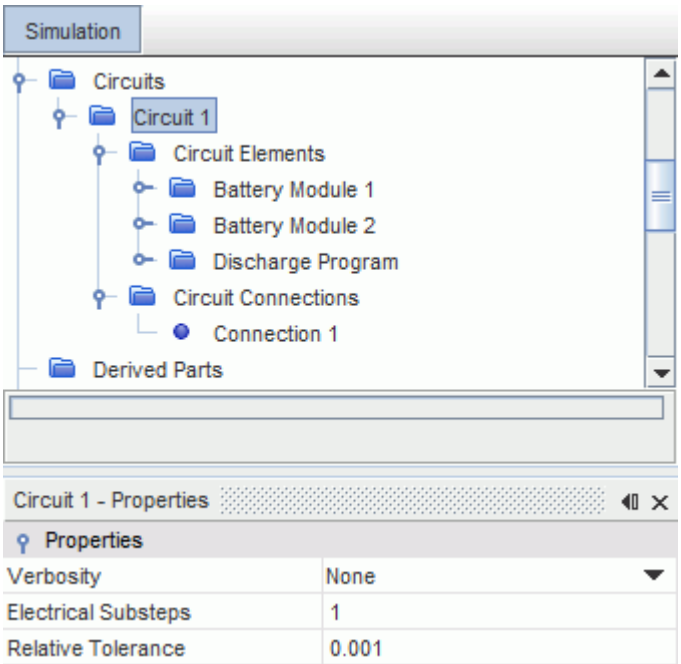
   Circuit elements can be derived from battery modules and program files, or created directly from the **Circuit Elements** node.
4. Create the appropriate connections between the circuit elements.

   The following conditions apply:

- A battery module can be used once only in a simulation: you cannot use the same battery module in two or more circuits.
- A program file can be used in multiple circuits.
- A circuit can have only one grounded connection.

Circuit Properties

The **Circuits** > **[Circuit Name]** node stores information relating to circuit elements and connections. You can create multiple circuits and set the properties independently for each circuit: the elements and connections for each circuit are stored in the corresponding **[Circuit Name]** node under the **Circuits** node.



Verbosity	Sets the level of output information that is displayed while the solver is running. By default this option is set to **None**. Setting the level to **Low** displays circuit voltage, current, and power after each time-step as well as CPU time and memory use for the circuit solver.
Electrical Substeps	Sets the number of Electrical Solver substeps per Thermal Solver time-step. The default setting is 1 electrical substep per thermal solver time-step. This option can be useful when there is detailed transient load input for the electrical load but the thermal solution does not require small time-steps. However, it can be expensive for DIST model runs to take multiple time-steps for each thermal time-step.
Relative Tolerance	Specifies a convergence parameter that the circuit solver uses internally. You are advised not to change this parameter.

Number of allowed groundings	Specifies the total number of grounded terminals.

Contents:

What Is the Circuit Solver?
What Is a Circuit Element?
Adding Circuit Elements
Connecting Circuit Elements
Reporting on Circuit Elements

What Is the Circuit Solver?

The circuit solver operates with the electrical solver to compute the electrical solution for the circuit.

The circuit solver is used in battery and electromagnetic simulations.

In battery simulations, the role of the circuit solver is to calculate the global battery voltage, and current in the case of a power load, based on the voltage of each battery cell.

The main computational cost of the circuit solver is in repeated calls to the electrical solver to obtain the battery cell voltage as the circuit solver iterates to a converged solution at each time-step. The number of ecells within a battery cell directly impacts the time that is required to reach each battery cell solution. Furthermore, the iterative solution of the electrical circuit for each battery cell can be fast when the current load is changing slowly. However, if the specified load is a power load for which both the voltage and current must be determined, or a rapidly varying current load, the circuit solver takes more iterative steps to reach a converged solution. For a power load, the specified load must not exceed the capacity of the battery circuit. In this case, no electrical solution exists and the electrical solver can fail to converge.

At each electrical time-step, the circuit solver does the following to determine the electrical solution:

- updates the load
- iteratively finds the final current and voltage for each battery cell that satisfies the specified circuit load
- computes heat generation for circuit connectors

To calculate voltage and current values, the circuit solver calls the battery cell model for each battery cell in the circuit. You cannot specify the battery cell model time-step. The initial time-step that is used in the battery cell model is the same as the time-step that is used in the circuit solver. However, when necessary, STAR-CCM+ automatically adjusts the battery cell model time-step to stabilize the circuit solver and obtain a solution.

To determine a solution, the battery cell model calls the unit cell battery model. You can specify the maximum time-step that the unit cell model can use.

The circuit load can be defined using either a separate program file that is created in BDS, or using STAR-CCM+ table or scalar load options. The program file contains one or more lines of input that specify a load type (voltage, current, resistance, or power), the numerical value of the load, and the time duration for that load. The table load option allows you to specify a load type (voltage, current, resistance, inductance, capacitance, or power), and then set the values using a table from which interpolated values can be easily obtained. The scalar load option allows you to specify a load type and then set constant value loads, or use user-defined field functions to set time-dependent loads.

The local temperatures that are input values to each ecell are mapped from the thermal solution. To start the simulation, the local temperatures are taken from the default initial state. The electrical solver uses BDS solvers that have been integrated into STAR-CCM+ to incorporate the user-determined unit cell model. The electrical solver receives a current value from the circuit solver and computes the voltage and the internal heat generation of each ecell, and computes the output voltage for each battery cell for the circuit solver.

After each circuit solver time-step, the heat generation from each ecell is mapped to the thermal grid and used as input for the thermal solver.

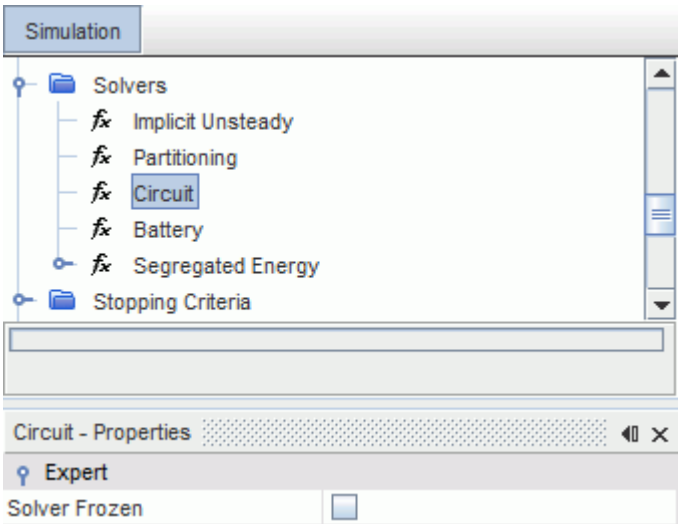
The circuit solver uses an adaptive time stepping scheme such that there is an electrical solution for each thermal solution time-step. The circuit solver and the thermal solver normally use the same time-step size, but the time-step sizes can be different.

If the electrical load is specified using a program file that is created in BDS, the electrical solver can take multiple substeps for each thermal time-step and is able to evaluate the load at intermediate thermal time values. If the electrical load is specified using a table, the load values are linearly interpolated so that only one load value is evaluated for each thermal time-step. It is possible to take substeps within each thermal time-step using a table load but the load value changes at the thermal time-step only. Similarly, using a scalar load option, the electrical solver evaluates the load at each thermal time-step.

Besides current and voltage, the ecells within each battery cell maintain a state of charge parameter. The normal operating range for this parameter is from 0 through 1: a state of 0-percent charge (depletion of the cell) and a state of 100-percent charge (fully charged cell). When one or more ecells reach either of these limit values, the simulation can continue but the results can become unphysical.

Circuit Solver Properties

The **Solvers** > **Circuit** node lets you activate or deactivate the circuit solver.



Solver Frozen	When **On**, prevents the solver from updating the solution while iterating or responding to other solvers. This feature is primarily a “debugging” tool and is not supported for normal simulation work.

Contents:

Specifying the Maximum Time Step for the Unit Cell Model
Working with the Battery Cell State of Charge

Specifying the Maximum Time Step for the Unit Cell Model

When necessary, STAR-CCM+ automatically adjusts the battery cell model time step to stabilize the circuit solver and obtain a solution.

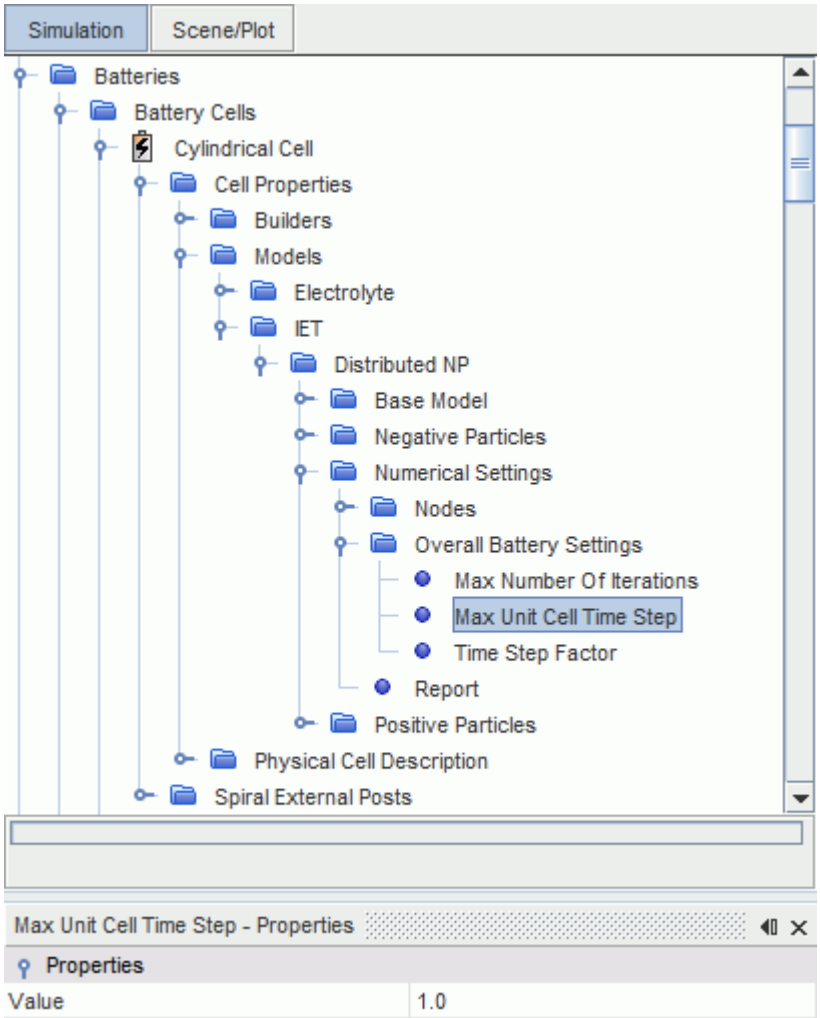
For example, if a time step of 100 seconds is specified in the Implicit Unsteady solver, the battery cell model normally delivers results in 100 seconds. However, if STAR-CCM+ reduces the battery cell model time step (for example, to 25 seconds), the battery model calls the unit cell model to deliver the results at 25-second increments.

You can specify what the unit cell model does in this scenario: attempt the full 25-second time step, or use a specified maximum time step. For example, if you specify a maximum time step of 1 second (the default value), the unit cell model makes 25 substeps, each of 1 second, to reach the battery cell model time step (25 seconds). This substepping usually enhances stabilization but requires a longer time to compute. However, if you specify a maximum time step greater than 25 seconds, the unit cell model makes a single time step of 25 seconds.

The larger maximum time step setting allows the battery cell model to achieve the Implicit Unsteady time step quickly. You can use this setting in cases where CFD calculations are not needed, such as conduction heat transfer problems or a single cell analysis with no cooling system. In these cases, change is slow and large time steps can be taken without affecting the stability of the simulation.

To specify the maximum unit cell time step:

1. Under the **Cell Properties** node for the cell, expand the **Models** > **IET** > **Distributed NP** > **Numerical Settings** > **Overall Battery Settings** node.
2. Select the **Max Unit Cell Time Step** node and, in the *Properties* window, set the appropriate maximum time step value in seconds.



Working with the Battery Cell State of Charge

If a battery cell state of charge reaches 0% or 100%, the simulation results can become unphysical, but the simulation can continue to run.

If the battery cell uses a physical model (such as the DISTNP model), the simulation stops automatically when the state of charge reaches 0% or 100%.

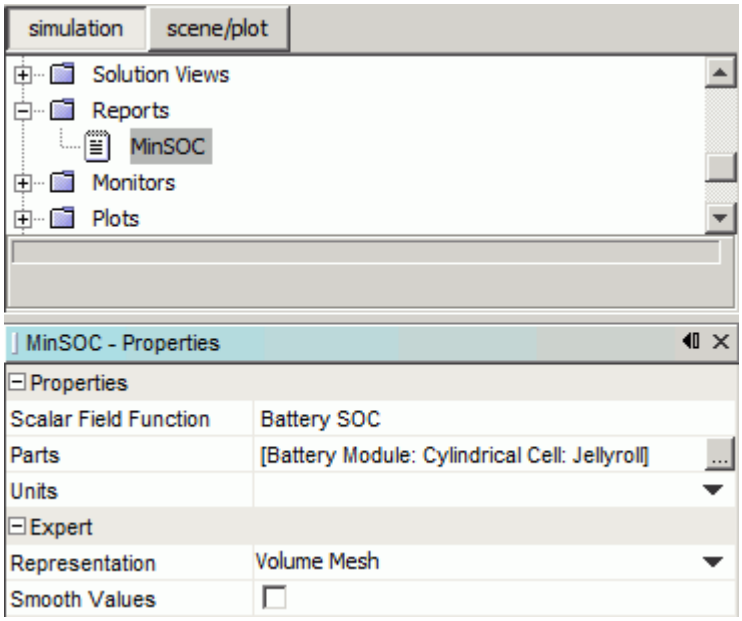
However, if the battery cell uses an empirical model (such as the NTG model or the RCR model), the state of charge of the cell becomes fixed at 0% or 100%. The simulation continues to run, but uses the constant value that the model computes at the fixed state of charge. Obviously, a real battery cell does not do this.

If you want your simulation to stop when the state of charge of any battery cell reaches 0% or 100%, you can set the appropriate stopping criteria.

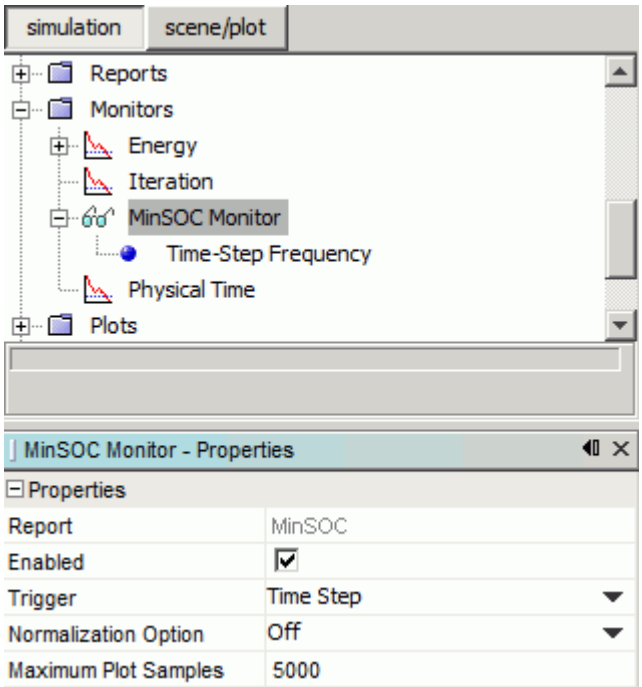
To set the stopping criteria:

1. Create a new minimum report and, in the *Properties* window, do the following:
    - Set *Scalar Field Function* to **Battery SOC**.
    - Set *Parts* to the jellyroll region.

This setting is important because the state of charge is defined only on the jellyroll region. If you choose any other regions, the monitor cannot find the minimum state of charge value.



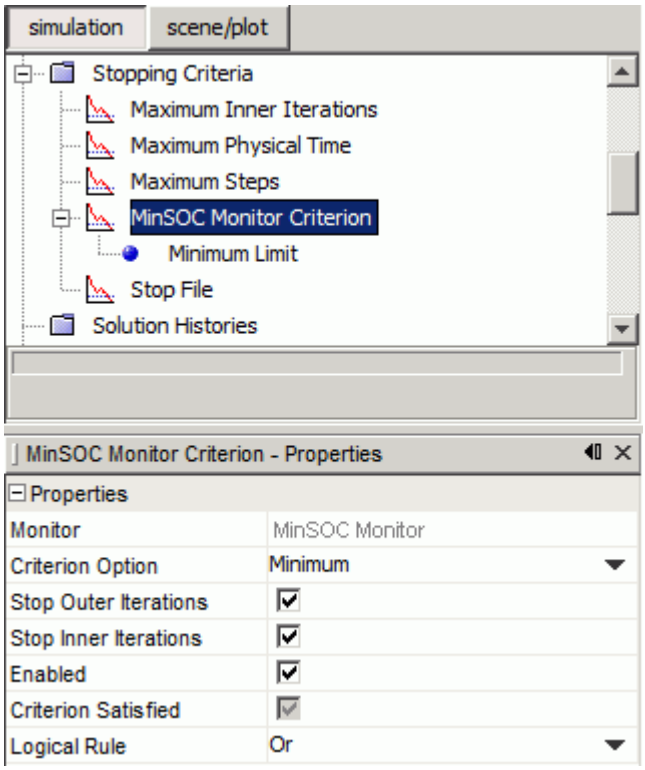
2. Create a monitor from the minimum state of charge report.



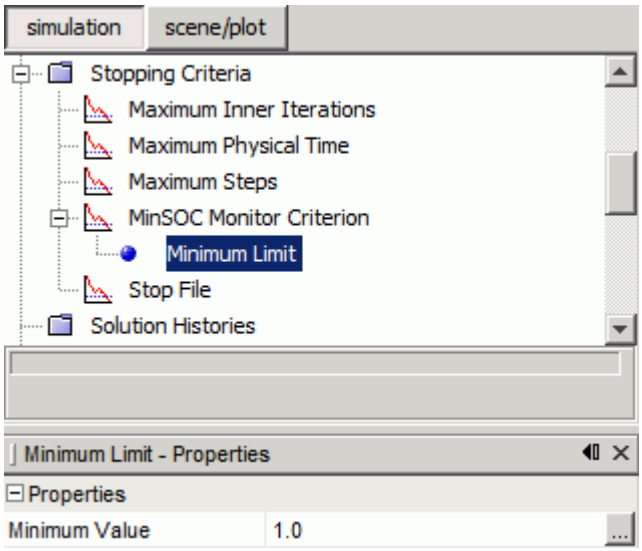
If you want to visually monitor the state of charge values, create a plot from the minimum state of charge report as well.

3. Create a stopping criteria from the minimum state of charge monitor and, in the *Properties* window, do the following:

- Set *Criterion Option* to **Minimum**.
- To stop the simulation when the minimum state of charge is reached, activate *Stop Outer Iterations*.



4. Select the **Minimum Limit** node and, in the *Properties* window, set the *Minimum Value* as a percentage of the cell capacity.



See Also:

DistNP Model

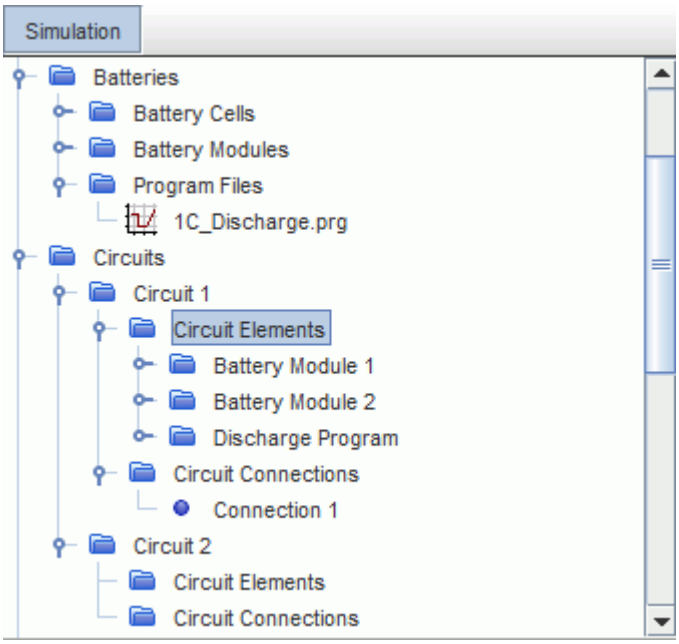
NTG (Newman, Tiedeman, Gu) Model
RCR Model

What Is a Circuit Element?

A circuit element is a component of a circuit. Multicontact circuit elements have a specified number of contacts. All other circuit elements have a positive and a negative terminal. When two or more circuit elements are defined, you can create a circuit by specifying connections between the contacts, or terminals.

You can derive a circuit element from a battery module or a program file, or create a circuit element directly from the **Circuit Elements** node.

The circuit elements for each circuit are stored under the **Circuits** > **[Circuit Name]** > **Circuit Elements** node.



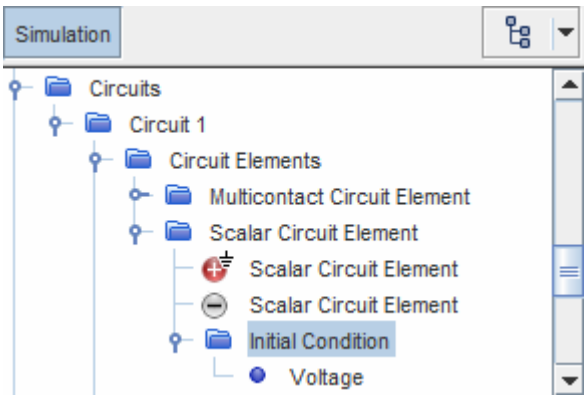
Circuit Element Properties

Battery Assembly	Available for Battery circuit elements. These elements can be created directly or derived from battery modules. Allows you to specify the battery module that the circuit element is associated with. Each battery module can be selected in one Battery circuit element only.
Forward Voltage	Available for Diode circuit elements. These elements can only be created directly. Allows you to specify the forward voltage of the diode. If the voltage across the diode exceeds the specified value, the diode behaves like a linear resistor plus a series voltage source. The value of

	the **On Resistance** property specifies the resistance of the linear resistor. If the voltage across the diode is less than the specified value, the diode behaves like a linear resistor with low conductance specified by the value of the **Off Conductance** property.
On Resistance	Available for Diode circuit elements. Allows you to specify the resistance of the diode.
Off Conductance	Available for Diode circuit elements. Allows you to specify the conductance of the diode.
Number of Contacts	Available for multicontact circuit elements. Allows you to specify the number of contacts *n* representing the connected coils.
Program Data	Available for Program circuit elements. These elements can be created directly or derived from program files. Allows you to specify the program file that the circuit element is associated with.
Load	Available for Scalar circuit elements. These elements can only be created directly. Allows you to specify the load with a scalar function. The scalar function can be a constant value or an expression which is evaluated over time. The expression can use field functions that you define.
Load Type	Available for Scalar, Excitation Coil, and Table circuit elements. These elements can only be created directly. Specifies the load type that is read in.
	Current — The load is current load.
	Power — The load is power load.
	Voltage — The load is voltage load.
	Resistance — The load is a resistor load.
	Inductance — The load is an inductor load.
	Capacitance — The load is a capacitor load.
	Voltage Source — In Excitation Coil circuit elements, defines the load using the electromotive force calculated by the Excitation Coil model in the specified region. See Excitation Coil Model Reference.
	Compound — In Excitation Coil circuit elements, defines the load as: $V = RI + L\frac{dI}{dt}$

		where R and L are the excitation coil region lumped parameters (that is, resistance and inductance) calculated by the **Excitation Coil Lumped Parameter** model. See Excitation Coil Lumped Parameter Model Reference.
Table	Available for Table circuit elements. These elements can only be created directly. Allows you to specify the load with a table. The complete list of table circuit element properties is listed under using tables.	
Regions	Available for Excitation Coil circuit elements. These elements can only be created directly. Specifies the regions that comprise the excitation coil.	

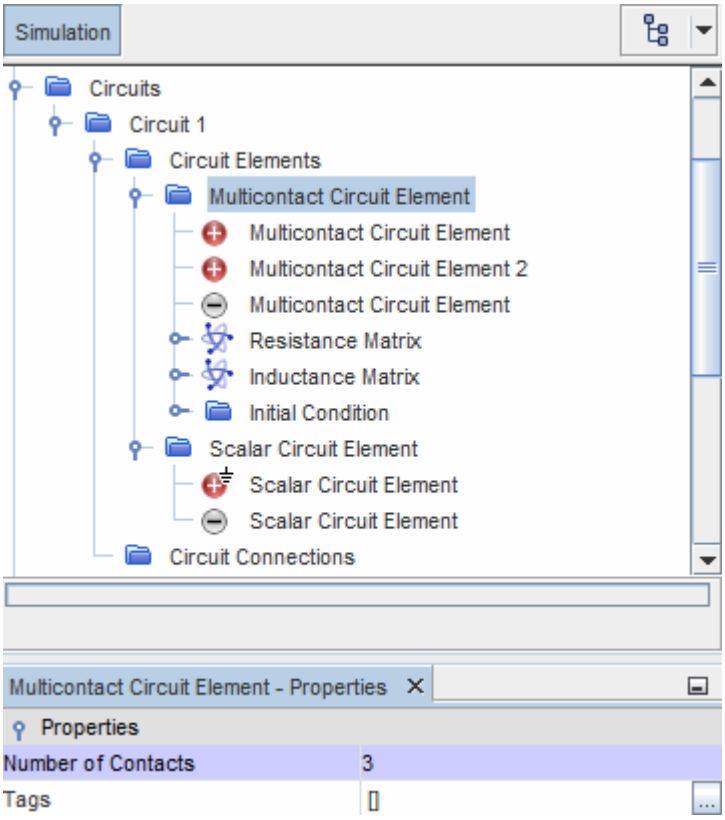
STAR-CCM+ allows you to specify initial conditions for multi-contact circuit elements and scalar circuit elements with either an inductance or a capacitance load.



What Is a Circuit Element Terminal?

All circuit elements other than multicontact circuit elements have a positive and a negative terminal. By default, STAR-CCM+ assumes that the positive terminal of the program file or table circuit element is grounded.

Multicontact circuit elements have a specified number of contacts *n*, set using the *Number of Contacts* property. The negative contact represents the voltage reference. The positive *n*-1 contacts represent a connected group of coils.



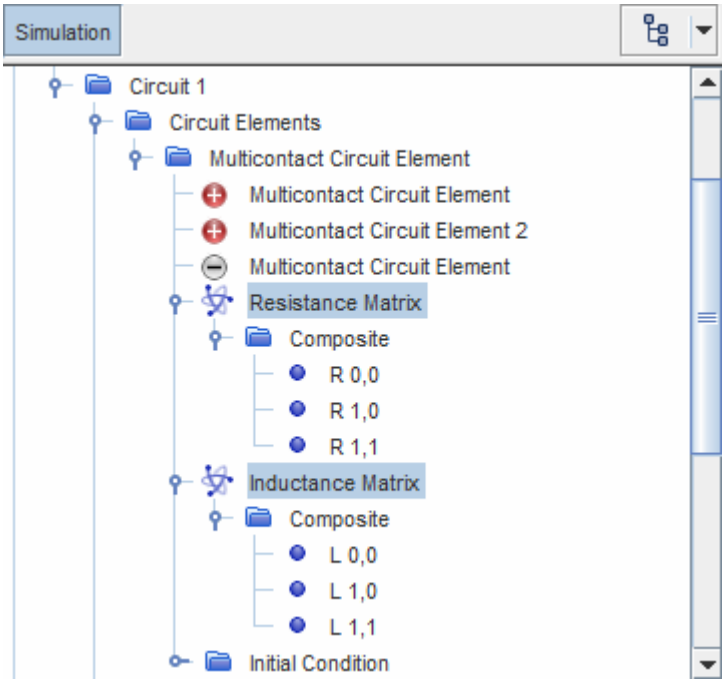
The terminal and contact nodes are stored under the circuit element node, and are used to create connections between different circuit elements.

Circuit Element Terminal Properties

Connection	Displays the name of the connection to which the terminal is assigned.	
Grounded	Specifies whether the terminal is grounded, or “earthed”.	
	Activated	Terminal is grounded.
	Deactivated	Terminal is not grounded.

Multi-Contact Circuit Element Controls

In multicontact circuit elements, you specify the resistance and inductance of the connected contacts using the **Resistance Matrix** and **Inductance Matrix** nodes. If *n* is the number of contacts of a multicontact circuit element, you specify the resistance and inductance as symmetric $(n-1) \times (n-1)$ matrices. The diagonal components represent the resistance, or inductance, from each positive terminal to the negative terminal. The off-diagonal components represent the mutual resistance, or mutual inductance.



See Also:

Adding Circuit Elements to a Circuit

Connecting Circuit Elements

Defining an Electrical Load

Setting Up a Battery Module

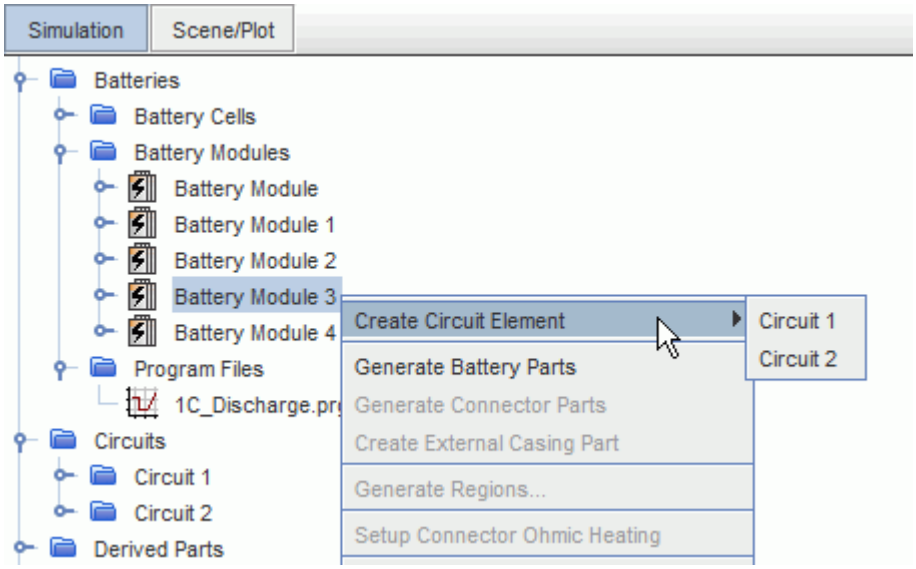
Adding Circuit Elements to a Circuit

Circuit elements can be derived from battery modules and program files, or created directly from the **Circuit Elements** node.

To derive a circuit element from a battery module or program file:

- Right-click the **Battery Modules** > **Battery Module** node or the **Program File** > **<program_file>.prg** node and select **Create Circuit Element**.

  If two or more circuits exist, select the circuit in which to create the circuit element.



To create a circuit element directly:

- Right-click the **Circuits** > **[Circuit Name]** > **Circuit Elements** node, select **New Circuit Element** and then select the appropriate option:
  - **Battery Circuit Element** (available only when the Battery model is activated in the physics continuum)
  - **Multicontact Circuit Element**
  - **Diode Circuit Element**
  - **Program Circuit Element** (available only when the Battery model is activated in the physics continuum)
  - **Scalar Circuit Element**
  - **Table Circuit Element**
  - **Excitation Coil Circuit Element** (available only when the Excitation Coil model is active in the physics continuum)

The new circuit element is added under the **Circuits** > **[Circuit Name]** > **Circuit Elements** node.

When you create a circuit element, the corresponding field function **Current <Circuit Element Name> Circuit Element** is made available under the **Tools** > **Field Functions** node. This field function represents the electric current through the circuit element.

If you create a circuit element directly from the battery module or program file, the corresponding **Battery Assembly** or **Program Data** property is set automatically.

If you create a circuit element directly from the **Circuit Elements** node, you must specify the necessary properties in the *Properties* window.

Connecting Circuit Elements

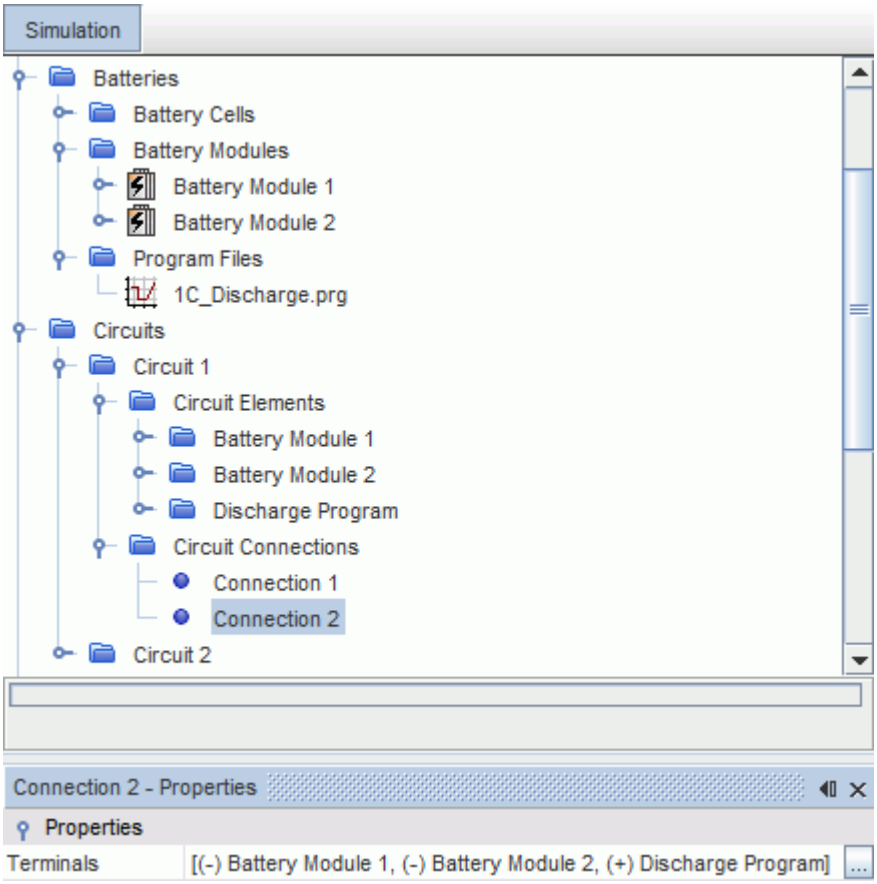
Circuit elements are connected by creating circuit connections within the circuit solver. The process involves connecting the terminals of different circuit elements.

It is possible to connect circuit elements in parallel and in series. In a series connection, the positive terminal of one circuit element is connected with the negative terminal of another. In a parallel connection, the circuit connection contains three or more terminals, depending on the layout of the circuit. An example of creating parallel connections is shown below.

There are two ways to create connections between circuit elements:

- Using the terminal nodes (Recommended)
- Using the Circuit Connections Manager

Each circuit connection is stored under the **Circuits** > **[Circuit Name]** > **Circuit Connections** node. Each **Connection** node has properties and is subject to common menu items.

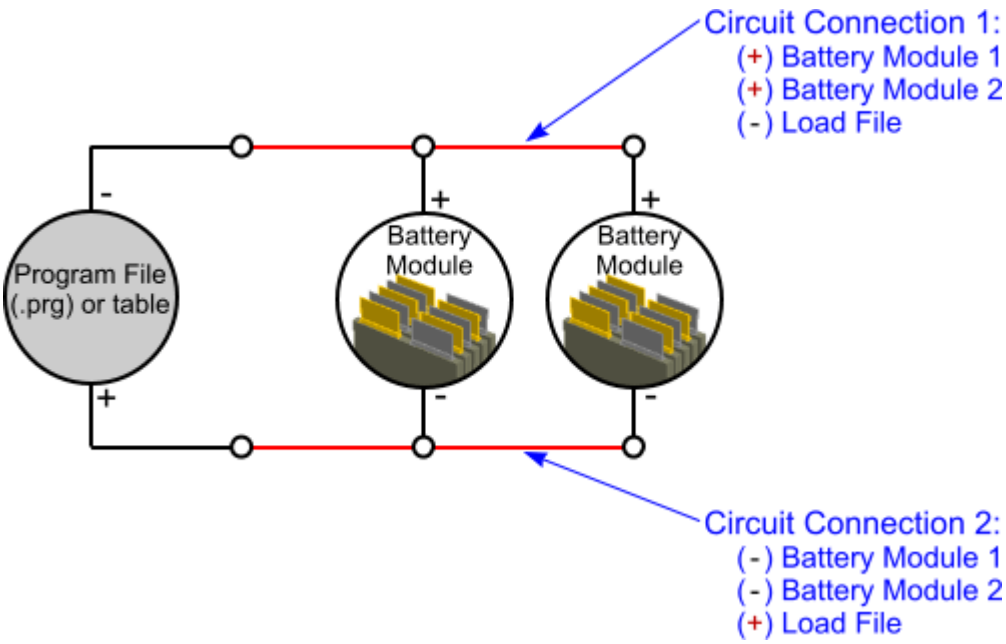


Circuit Connection Properties

Terminals	Displays the names of the terminals that are connected in this connection.

Creating Parallel Connections

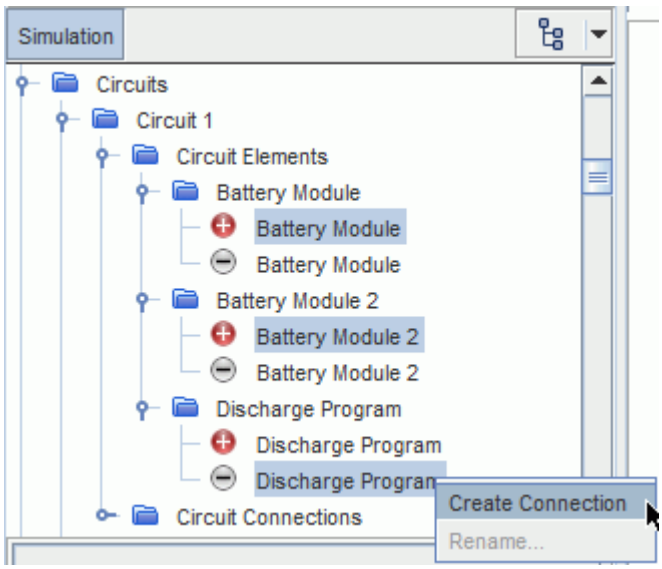
To create a circuit that consists of several circuit elements that are connected in parallel, add all the terminals for a parallel connection to a single circuit connection.



In the above example, two connections are required. The first contains the positive terminal from each battery module and the negative terminal from the load. The second contains the negative terminal from each battery module, and the positive terminal from the load.

To create the first connection from the example above:

1. Multi-select the positive terminal from each battery module and the negative terminal from the load.
2. Right-click any selected node and then click **Create Connection**.



The new connection is added under the **Circuit Connections** node.

3. Repeat the previous two steps to create the second connection with the appropriate terminals, that is, the negative terminal from each battery module, and the positive terminal from the load.

Contents:

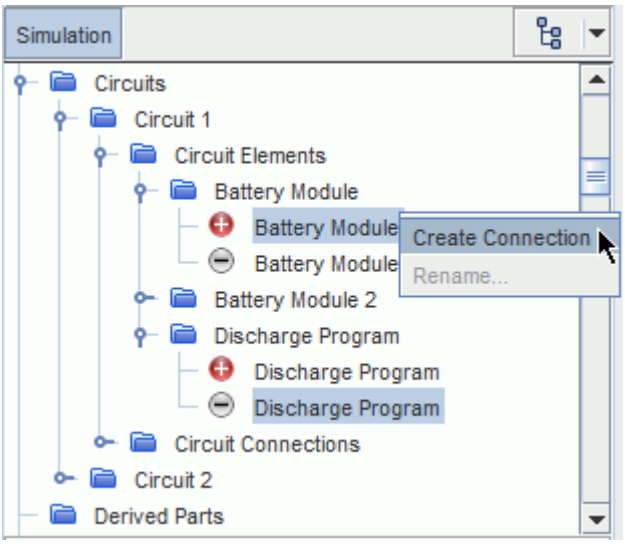
Using the Terminal Nodes to Create Connections
Using the Circuit Connections Manager to Create Connections
Creating Mutual Inductance Connections

Using the Terminal Nodes to Create Connections

This section describes how to use the terminal nodes to create connections.

Before you begin, ensure that you have created circuit elements that you wish to connect in the circuit.

1. Expand the circuit element nodes under the **Circuits** > **[Circuit Name]** > **Circuit Elements** node.
2. Select the terminal nodes that you wish to connect by using `<Ctrl><Click>`, for example, **(+) Battery Module 1** and **(-) Discharge Program** in the following screenshot.
3. Right-click one of the selected terminals and select **Create Connection**.



The connection is added under the **Circuits** > **[Circuit Name]** > **Circuit Connections** node. You can change the terminals that are connected by this connection by editing the properties of the **Connection** node.

Using the Circuit Connections Manager to Create Connections

In this method, an empty circuit connection is created first and the circuit element terminals to include in the connection are defined later.

To create a connection:

1. Right-click the **Circuits** > **[Circuit Name]** > **Circuit Connections** node and select **New**.

   The connection is added under the **Circuits** > **[Circuit Name]** > **Circuit Connections** node. To activate the connection, terminals must be added to it.
2. Select the **Circuit Connections** > **Connection** node.

3. In the *Properties* window, click ... to the right of the *Terminals* property.
4. In the *Terminals* dialog, select the terminals to be connected.

   The *Terminals* dialog only shows terminals that have not been used in any other connections.
5. Click **OK** to confirm the selection.

Creating Mutual Inductance Connections

Mutual inductance connections account for the mutual interaction between the inductances of the connected circuit elements.

You can create mutual inductance connections between:

- Scalar circuit elements, when the *Load Type* of each circuit element is set to **Inductance**.
- Excitation coil circuit elements, when the *Load Type* of each circuit element is set to **Compound**.

To create a mutual inductance connection between circuit elements:

1. Multi-select the circuit elements to connect (either scalar circuit elements or excitation coil circuit elements).
2. Right-click one of the selected elements and select **Create Mutual Inductance Connection**.

   STAR-CCM+ adds the connection under the **Circuit** > **Mutual Inductance Connections** node.
3. Depending on the type of connected circuit elements, define the mutual inductance connection in one of the following ways:

Circuit Element Type	Procedure
Excitation Coil	• In the physics continua associated with the coil regions, activate the **Excitation Coil Lumped Parameter** model. STAR-CCM+ automatically calculates the mutual inductance from the inductances of each coil region. See Excitation Coil Lumped Parameter Model Reference.
Scalar	• Select the relevant **Circuit** > **Mutual Inductance Connections** > **[Mutual Inductance Connection]** node and set the *Mutual Inductance* value.

Reporting on Circuit Elements

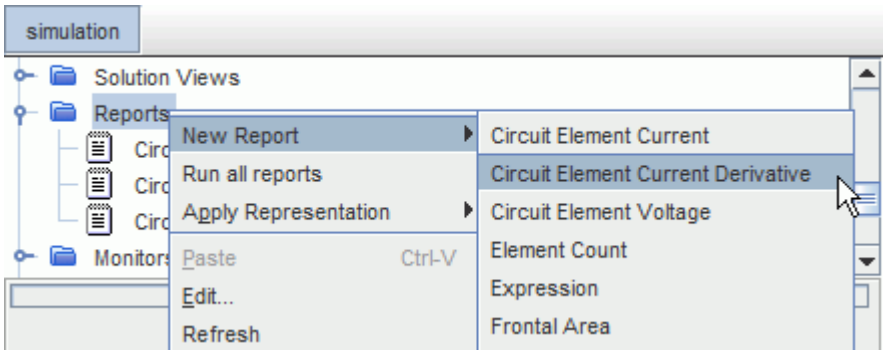
You can report on specific circuit elements to monitor the voltage, current, or current derivative across the element.

The following reports are available:

- Circuit Element Current
- Circuit Element Current Derivative
- Circuit Element Voltage

To set up a circuit element report:

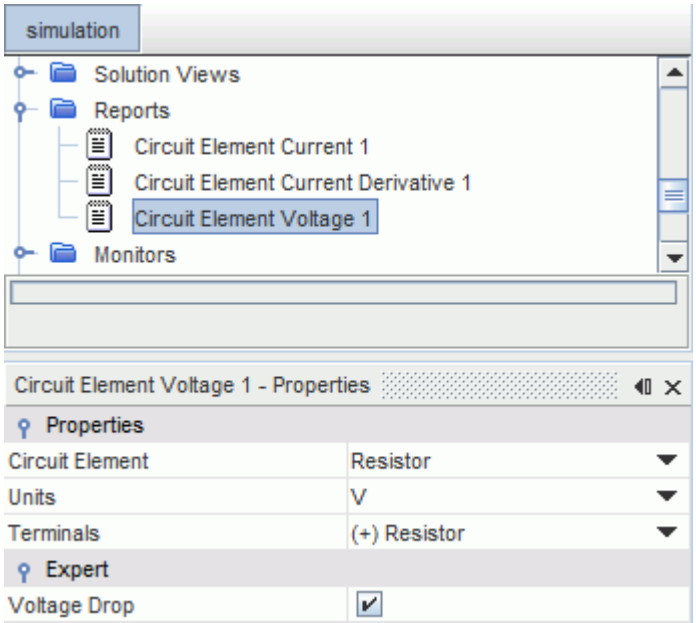
1. Right-click the **Reports** node, click **New Report**, and then click the appropriate report type.



   The corresponding node is added under the **Reports** node.

2. Select the new report node and set the appropriate properties:
   - Circuit Element Current Report Properties
   - Circuit Element Current Derivative Report Properties
   - Circuit Element Voltage Report Properties

   The example below is for the **Circuit Element Voltage** report.



Circuit Element Current Properties

Circuit Element	Specifies the circuit element to report on. The drop-down list contains all of the circuit elements that are defined under the **Circuits** > **[Circuit Name]** > **Circuit Elements** node.

Units	Specifies the units to use for the current values. The default is A (Amps).
Terminals	Specifies the terminal for which the current is measured. By default, the report uses the negative terminal of the circuit element. For multicontact circuit elements, you can change the default setting to any contact of the circuit.

Circuit Element Current Derivative Properties

Circuit Element	Specifies the circuit element to report on. The drop-down list contains all of the circuit elements that are defined under the **Circuits** > **[Circuit Name]** > **Circuit Elements** node.
Units	Specifies the units to use for the current derivative values. The default is A/s (Amps per second).
Terminals	Specifies the terminal for which the current derivative is measured. By default, the report uses the negative terminal of the circuit element. For multicontact circuit elements, you can change the default setting to any contact of the circuit.

Circuit Element Voltage Properties

Circuit Element	Specifies the circuit element to report on. The drop-down list contains all of the circuit elements that are defined under the **Circuits** > **[Circuit Name]** > **Circuit Elements** node.	
Units	Specifies the units to use for the voltage values. The default is V (Volts).	
Terminals	Specifies the terminal for which the voltage is measured.	
Voltage Drop	Specifies whether the voltage is measured with respect to the other terminal or with respect to ground. This option is activated by default.	
	Activated	The voltage at the selected terminal is measured with respect to the other terminal.
	Deactivated	The voltage at the selected terminal is measured with respect to ground.

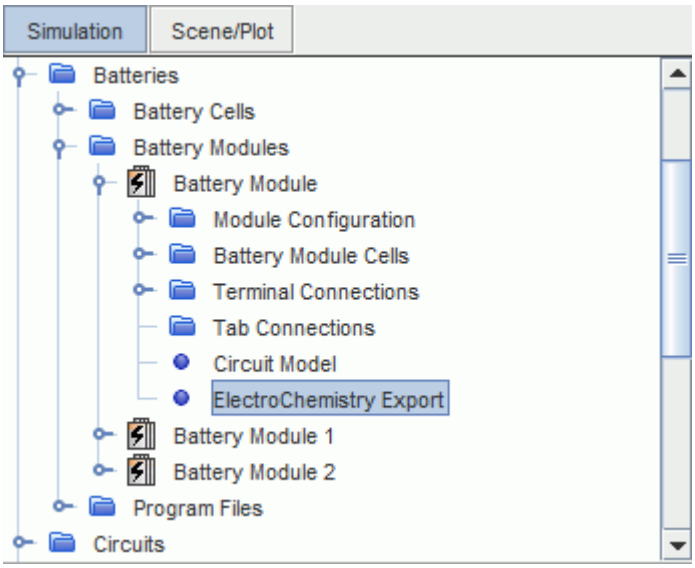
Exporting Electrochemistry Data

You can export detailed electrochemistry data for a battery module at each specified time-step. Electrochemistry data is exported to profiles (.pr2 files), which are text files that can be read into the BDS Profile Viewer.

One `.pr2` file is generated for each ecell in each battery cell. For example, for a single cell battery module where the battery cell contains 3x3x1 ecells, 9 `.pr2` files are generated.

This feature is only available for battery modules containing battery cells that use a Distributed battery model. The Distributed model is selected in Battery Design Studio® (BDS) when you set up the `.tbm` file, for example Distributed 1D, or Distributed 3D. You can check which battery model a battery cell uses by checking the *Unit Cell Model* property of the **Battery Cell** node. It is set to Distributed if a Distributed battery model was selected in BDS.

Electrochemistry data export is set up using the properties of the **ElectroChemistry Export** node of the battery module of interest. This node is added to the tree after setting up and generating the battery module, and once the battery mesh continuum has been assigned.



Electrochemistry Export Properties

Export Directory	Sets the directory for the exported data files.
Export Frequency	Sets the time interval at which to export data. The default value of 1 results in data being exported at each time-step. Setting the frequency to 4, for example, would export data at every fourth time-step.
Export Start Time	Sets a specific time at which data export begins (s).
Exported Battery Module Cells	Selects cells in the battery module for which to export data.

See Also:

Configuring a Battery Cell

Creating Battery (.tbm or .ebm) Files in BDS

Generating a Battery Mesh

Generating a Battery Module

What Is a Battery Module?

What Is an Ecell?

Evaluating Electrode Structure Tortuosity

This section outlines a technique that you can use to evaluate an electrode structure tortuosity with STAR-CCM+. This technique can be helpful when you are creating your own geometry from idealized particles.

You can use STAR-CCM+ to calculate the tortuosity of porous media from the data that is obtained with laboratory X-ray instruments. The tortuosity is evaluated from a mass transport perspective.

The analog between Fick's law for mass transport driven by a concentration gradient and Fourier's law for heat transfer across a temperature gradient is well documented (for example, [169]). The mathematical and physical similarities of the heat and mass transport calculations allow you to effectively utilize a heat-transfer simulation in STAR-CCM+ to predict the tortuosity of a pore network. This technique provides the same tortuosity value but has boundary conditions that are easier to implement.

STAR-CCM+ can use tomography data sets that are formatted as stereolithography (*.stl) files, which describe the interface between the solid and pore phases. STAR-CCM+ can then generate adaptive polyhedral volume meshes from the .stl files.

Using this microstructurally faithful modeling framework, you can run simulations by applying the boundary conditions that are suitable for the heat transfer analog, using nominal values for the thermal conductivity and temperature difference.

To minimize the effect of small scale heterogeneities, run the simulations on the largest volume that is fully bounded by the sample data. To measure the tortuosity in all three directions, you should run three separate simulations. For each simulation, set the two parallel faces that are normal to the direction being investigated to fixed temperatures, while specifying all of the other faces as adiabatic.

The heat transfer measured can then be put into a modified form of the heat transfer equation, and the tortuosity determined as follows:

$$\tau = \phi \frac{Q}{Q_p} \tag{1164}$$

where:

- Q = heat transferred through a solid block (W)
- Q_p = heat transferred through a porous media (W)
- τ = Tortuosity
- ϕ = Porosity

Battery Modeling Formulation

This section provides the battery modeling formulation.

A bibliography is also included.

Contents:

NTG (Newman, Tiedeman, Gu) Model
RCR Model
Physics-based Models
Battery Modeling Bibliography

NTG (Newman, Tiedeman, Gu) Model

The NTG model uses a polynomial representation to capture the voltage performance of a battery cell.

This model was proposed in a number of papers ([172], [174], [176]).

The NTG model equations are as follows.

$$V_{\text{cell}} = V\left(\text{SOC}_i, T\right) - \frac{i}{Y\left(\text{SOC}_i, T\right)} \tag{1165}$$

Discharge:

$$\text{SOC}_i = 1 - (1 - \text{SOC})\frac{C_{\text{Ah-}m^{-2},0}}{C_{\text{Ah-}m^{-2},i}} \tag{1166}$$

Charge:

$$\text{SOC}_i = \text{SOC}\frac{C_{\text{Ah-}m^{-2},0}}{C_{\text{Ah-}m^{-2},i}} \tag{1167}$$

$$\text{SOC} = \frac{\int i \, \text{d}\, t}{C_{\text{Ah-}m^{-2},0}} \quad i = \frac{I}{A} \tag{1168}$$

$$Q = I \cdot \left(U_{\text{oc}} - V_{\text{cell}} - T\frac{dU}{dT}\right) \tag{1169}$$

where:

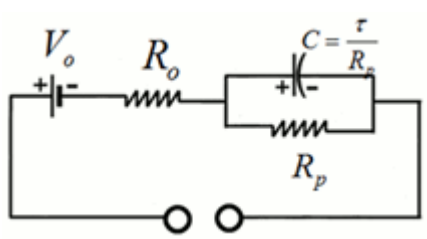
A	Electrode area, m^2
$C_{\text{Ah-}m^{-2}0}$	Nominal cell capacity, $\text{Ah-}m^{-2}$
$C_{\text{Ah-}m^{-2}i}$	Cell capacity at current density *i*, $\text{Ah-}m^{-2}$

E_a	Activation energy, $J\text{-mol}^{-1}K^{-1}$
i	Current density, $A\text{-}m^{-2}$
I	Current, A
SOC	State of Charge (fraction)
Q	Electrochemical heat generation, W. This value is Q_{echem} in the .out file.
T	Temperature, K
U_{oc}	Equilibrium voltage, V
V	Voltage term, V
V_{cell}	Working cell voltage, V
Y	Admittance term, $\mathrm{Ohm}^{-1}\text{-}m^{-2}$

RCR Model

The RCR model uses a simple equivalent circuit consisting of a resistance R_o in series with a resistor R_p and capacitor C in parallel, along with a state-of-charge dependent voltage source V_o to represent the behavior of the battery cell.

Due to the presence of the capacitor within the model, it is more suited to battery cells which contain high discharge or charge peaks, allowing for the temporal effects within the battery cell. This model can be created using the regression process within Battery Design Studio, depending on pulse power curves presented from testing. The five input parameters to the model can be represented by fifth-order polynomials. The coefficients should then be normalized by the active area of the battery cell [179].



$$V_o = \sum a_i \cdot SOC^i \tag{1170}$$

$$R_o = e^{-\frac{E_o}{RT}} \sum R_{o,i} \cdot SOC^i \tag{1171}$$

$$E_o = \sum E_{o,i} \cdot SOC^i$$

$$R_p = e^{-\frac{E_p}{RT}} \sum R_{p,i} \cdot SOC^i \tag{1172}$$

$$E_p = \sum E_{p,i} \cdot SOC^i$$

$$\tau = e^{-\frac{E_\tau}{RT}} \sum \tau_i \cdot SOC^i \tag{1173}$$

$$E_\tau = \sum E_{\tau,i} \cdot SOC^i$$

$$V_L = V_o - I \cdot R_o - \left(\frac{I}{C}\right)\Delta t + \left(V_L - V_o + I \cdot R_o\right)_{t-\Delta t} \exp\left[\frac{-\Delta t}{\tau}\right] \tag{1174}$$

$$\frac{dV_o}{dT} = \sum \frac{dV_o}{dT_i} \cdot SOC^i \tag{1175}$$

$$Q = I \cdot \left(V_o - V_L + T\frac{dV_o}{dT}\right) \tag{1176}$$

Physics-based Models

This section introduces the physics-based battery models.

The physics-based battery models in STAR-CCM+ are derived from the DualFoil model [170], [171]. This model is a detailed electrochemistry model that is used to predict battery cell performance. The physical formulation is common to both of the physical models that are implemented in STAR-CCM+: the DistNP model and the DistSegregated model.

The model is essentially one-dimensional in nature in that it resolves the liquid-phase as well as solid-phase electric charge transport (electric potential), and the liquid-phase salt transport from the negative current collector to the opposing positive current collector. An additional, one-dimensional transport equation is solved for the solid-phase lithium concentration, which models the lithiation of a respective active material particle. The equation is also solved for the flux of lithium cations across the Solid-Electrolyte Interface (SEI). The temperature is computed using an energy balance around the whole cell [168].

The physics-based model formulation is presented in the following sections.

Contents:

Potential Model
Concentration Model
Solid Electrolyte Interface Current Density
DistNP Model
DistSegregated Model

Potential Model

This section describes the Potential model.

Solid Active Material

Integral equation

$$\int_A -\sigma \nabla \phi_s \cdot \mathrm{d}\,\mathbf{A} = -\int_V a J_n \,\mathrm{d}\,V - \int_V aC \frac{\partial(\phi_s - \phi_l)}{\partial t} \,\mathrm{d}\,V \tag{1177}$$

where the terms are defined as follows:

σ	electrical conductivity [S/m]
ϕ	electric potential [V]
a	specific interfacial area [m^2/m^3]
C	double-layer capacitance [Farad/m^2]
J_n	SEI normal component of electric current density [A/m^2]

Effects introduced by surface chemistry are modeled by the terms on the right-hand side of Eqn. (1177), where the second term corresponds to the double-layer capacitance effect.

(Electric) Current definition

In the solid electrodes and current collectors, the electric current density is defined using Ohm's law as:

$$\mathbf{J}_s = -\sigma \nabla \phi_s \tag{1178}$$

Electrolyte

Integral equation

$$\int_A -\kappa \nabla \phi_l \cdot \mathrm{d}\,\mathbf{A} = \int_A -\kappa_d \nabla(\ln c_l) \cdot \mathrm{d}\,\mathbf{A} + \int_V a J_n \,\mathrm{d}\,V + \int_V aC \frac{\partial(\phi_s - \phi_l)}{\partial t} \,\mathrm{d}\,V \tag{1179}$$

The effective electrical (ionic) conductivity κ in the liquid phase accounts for an eventual non-unity porosity ε and tortuosity τ within the separator:

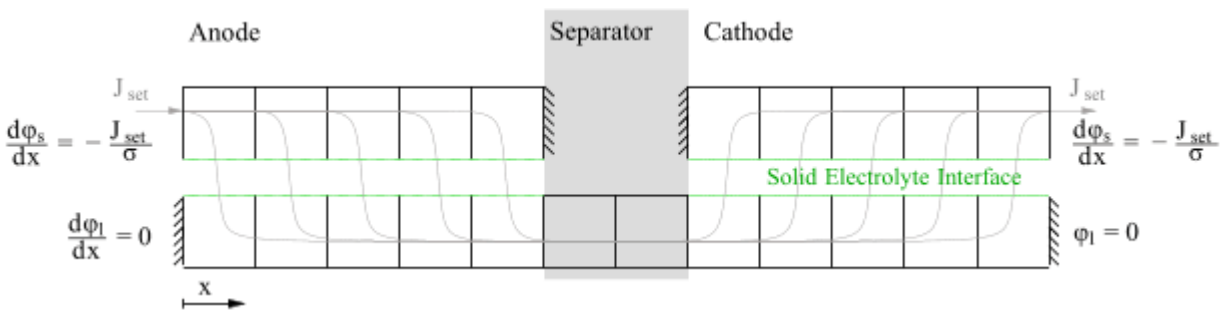
$$\kappa = \kappa_0 \frac{\varepsilon}{\tau} \tag{1180}$$

$$\kappa_d = \frac{2RT\kappa}{F}\left(1 - t_+{}^0\right)\left[1 + \frac{\mathrm{d}\left(\ln f_\pm\right)}{\mathrm{d}\,(\ln c)}\right] \tag{1181}$$

(Ionic) Current definition

$$\mathbf{J}_l = -\kappa \nabla \phi_l + \kappa_d \nabla(\ln c_l) \tag{1182}$$

A schematic illustration describing the electric potential formulation, including the prescribed boundary conditions, is shown below:



Concentration Model

This section describes the Concentration model.

Solid Active Material

Integral equation

A simple species diffusion transport equation is solved for lithium (lithium/salt concentration) transport within the electrode phase:

$$\frac{\partial}{\partial t}\int_V c_s \, \mathrm{d}\, V = \int_A D_s \nabla c_s \cdot \, \mathrm{d}\, \mathbf{A} \tag{1183}$$

where c_s is the molar concentration [kmol/m^3] and D_s is the effective lithium diffusivity [m^2/s] (if the **Concentration Dependent Diffusion Coefficient** setting is selected in Battery Design Studio):

$$D_s = -D_{s,0}\frac{Fy}{RT}\frac{\mathrm{d}\, U_{eq}}{\mathrm{d}\, y} \tag{1184}$$

The terms are defined in the Electrochemistry Nomenclature.

Electrolyte

Integral equation

An additional transport term for lithium cation migration is considered in the (liquid) electrolyte phase, so that:

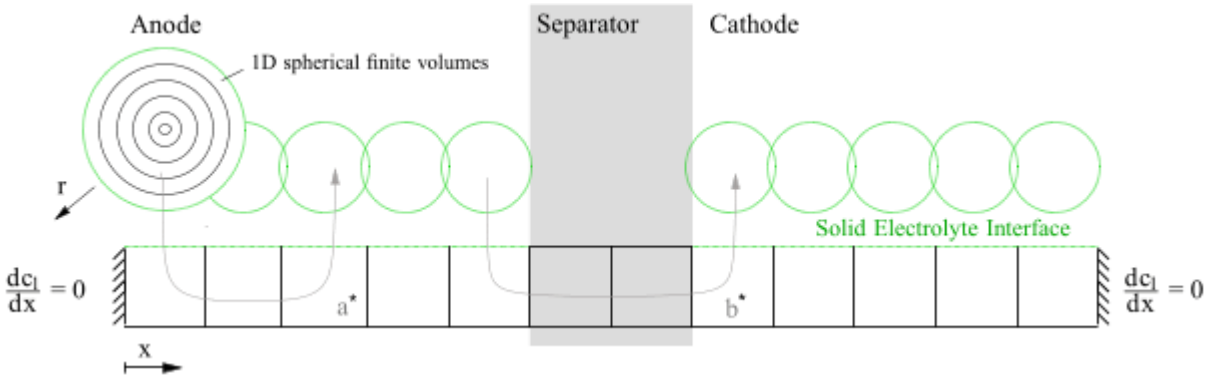
$$\frac{\partial}{\partial t}\left(\int_V \varepsilon c_l \, \mathrm{d}\, V\right) = \int_A D_l \nabla c_l \cdot \, \mathrm{d}\, \mathbf{A} - \int_V \frac{\mathbf{J}_l \cdot \, \nabla t_+^0}{F} \, \mathrm{d}\, V + \int_V \frac{a\mathbf{J}_n}{F}\left(1 - t_+^0\right) \, \mathrm{d}\, V \tag{1185}$$

where ${t_+}^0$ is the transference number of cations and where the effective lithium/salt diffusivity coefficient:

$$D_l = D_{l,0}\frac{\varepsilon}{\tau}\left[1 - \frac{\mathrm{d}\,\left(\ln c_0\right)}{\mathrm{d}\,\left(\ln c\right)}\right] \tag{1186}$$

accounts for non-unity values of porosity ε within the separator.

The model that is assumed for lithium (salt) transport is illustrated below:



In this figure, a* illustrates an internal relaxation current and b* illustrates salt transport during discharge.

See Also:

Electrochemistry Nomenclature

Solid Electrolyte Interface Current Density

This section describes the solid electrolyte interface current density formulation.

The Butler-Volmer relationship is used to compute the solid electrolyte interface (SEI) electric current density, which is written as:

$$J_n = J_0\left(e^{\frac{\alpha_a F\eta}{RT}} - e^{\frac{-\alpha_c F\eta}{RT}}\right) \tag{1187}$$

where:

$$J_0 = Fk\left(\frac{c_s}{c_{s,\max}}\right)^{\alpha_1}\left(1 - \frac{c_s}{c_{s,\max}}\right)^{\alpha_2}\left(\frac{c_l}{c_{l,\mathrm{ref}}}\right)^{\alpha_3} \tag{1188}$$

The SEI Surface Overpotential η is computed as:

$$\eta = \phi_s - \phi_l - U_{\mathrm{eq}} - J_{n,s}R_{\mathrm{SEI}} \tag{1189}$$

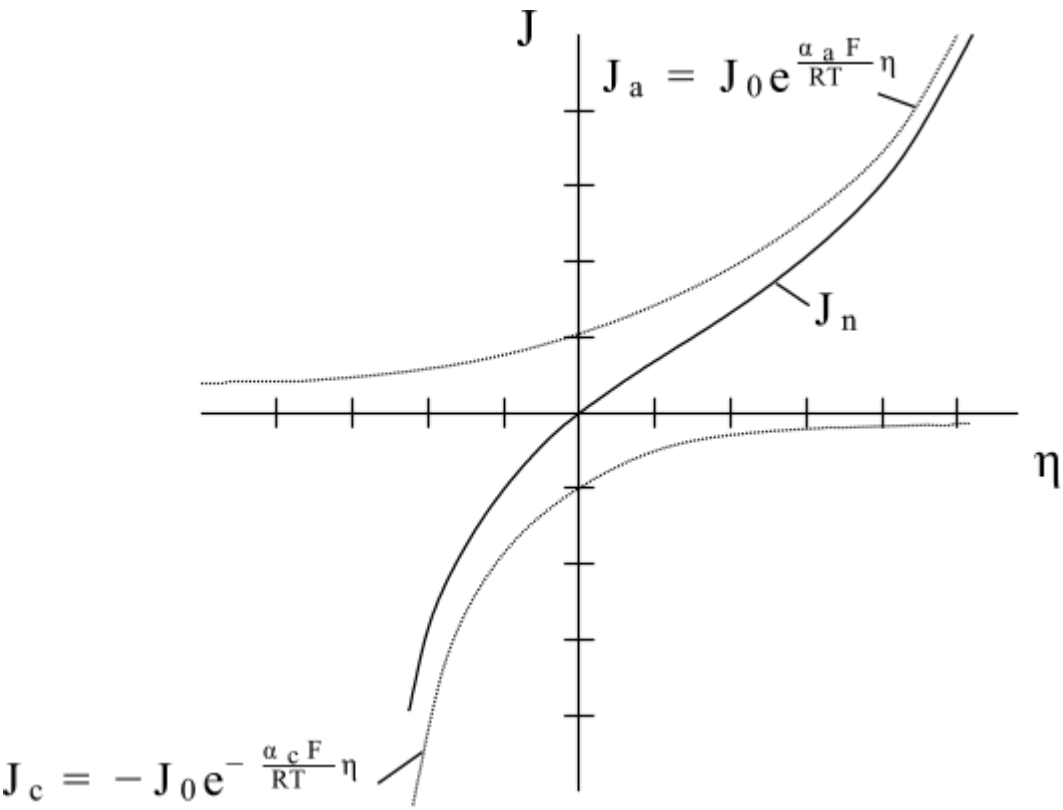
The rate coefficient k depends on temperature according to:

$$k = k_0 e^{\frac{-E_a}{RT}} \tag{1190}$$

Similarly, the electrical resistance of the SEI is dependent on temperature:

$$R_{\mathrm{SEI}} = R_0 e^{\frac{-E_{\mathrm{SEI}}}{RT}} \tag{1191}$$

The Butler-Volmer relationship is illustrated below:



J_a represents the anodic current contribution, while J_c represents the cathodic current contribution to J_n.

DistNP Model

The DistNP model is an extension of the DualFoil model. The STAR-CCM+ implementation allows for several active materials within the same model and also accounts for complex expressions to model diffusion coefficients.

This model uses a pentadiagonal version of Newman's band algorithm [178]. The DistNP model solves numerically for the solid-phase diffusion equation, instead of using Duhamel's superposition integral [170]. This allows the usage of concentration-dependent diffusion coefficients. Active material particles are modeled either as spherical, cylindrical, or planar.

In the present implementation, output quantities within the battery cell, such as lithium concentration, are not available from the STAR-CCM+ user interface. You can apply such a model but only the post-processing quantities that are generic to all three models are available.

To check the 1-D model data, you can export it to .pr2 files, which are text files that can be read into the BDS Profile Viewer.

The DistNP model requires many input parameters to describe the battery cell but these parameters are used in a model based on first principles and hence produce a more robust prediction. The model created should be checked in Battery Design Studio before proceeding in STAR-CCM+.

DistSegregated Model

The DistSegregated model is an extension of the DualFoil model. The STAR-CCM+ implementation employs an iterative, segregated solution algorithm, but allows only a single active material per electrode. Active material particles are modeled as spheres.

The solver for electric potential is numerically coupled in active electrodes and the electrolyte. The DistSegregated model has a fully segregated solution procedure for calculating the concentration in active electrodes and the electrolyte. The model uses “delta form”.

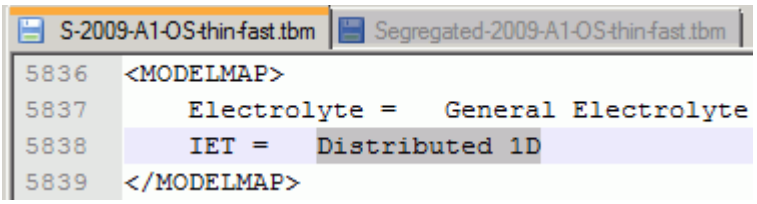
Activating the DistSegregated Model

The DistSegregated model cannot be created in Battery Design Studio.

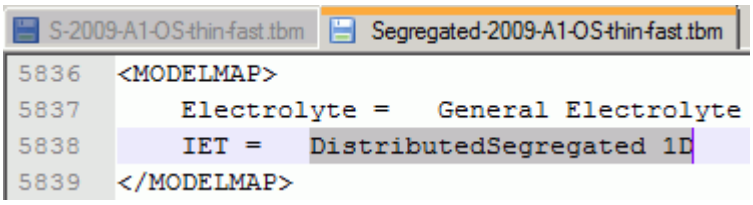
Note: The DistSegregated model can only be activated for Distributed models that have only one positive particle and one negative particle per electrode.

To activate the DistSegregated model in STAR-CCM+:

1. Open a suitable DistNP model .tbm file for editing.
2. Make the following changes to the file:
   - In the `<MODELMAP>` section:



replace the **IET=** keyword `Distributed` with the keyword `DistributedSegregated`:



Leave the remaining text (1D in this example, but can be 3D or NP) unchanged.

   - Make the same change in the corresponding <SIMMOD> section.

There can be several of these sections in the .tbm file, and they usually appear before the `<MODELMAP>` section. You edit only the variant (`1D`, `3D`, or `NP`) that is listed in the `<MODELMAP>` section (in this example, `1D`):

```
S-2009-A1-OS-thin-fast.tbm   Segregated-2009-A1-OS-thin-fast.tbm
5082  <SIMMOD>
5083  Distributed 1D
5084  LiIon\Pouch
5085  IET
5086      Negative Particle 1  Activation Ene
```

Replace the keyword `Distributed` with the keyword `DistributedSegregated`, but leave the remaining text (1D in this example) unchanged:

```
S-2009-A1-OS-thin-fast.tbm   Segregated-2009-A1-OS-thin-fast.tbm
5082  <SIMMOD>
5083  DistributedSegregated 1D
5084  LiIon\Pouch
5085  IET
5086      Negative Particle 1  Activation Ene
```

3. Save the modified .tbm file under a suitable name.

Battery Modeling Bibliography

[168] *Bernardi, D., Pawlikowski, E. and Newman, J. 1985. "A general energy balance for battery systems." Journal of The Electrochemical Society., 132 (1), pp. 5–12.*

[169] *Bird, R. B., Stewart, E., and Lightfoot, E. N., Transport Phenomenon, ISBN 0-471-07392-X*

[170] *Doyle, M., Fuller, T. F. and Newman, J. 1993. "Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell." Journal of The Electrochemical Society., 140 (6), pp. 1526–1533.*

[171] *Fuller, T.F., Doyle, M. and Newman, J. 1994. J. Electrochem. Soc., 141(1).*

[172] *Gu, H. 1983. J. Electrochem. Soc. 130 (7) pp. 1459-1464.*

[173] *Jang, Y.-I., Neudecker, B. J. and Dudney, N. J. 2001. "Lithium Diffusion in LixCoO2 (0.45 < x < 0.7) Intercalation Cathodes." Electrochemical and Solid-State Letters., 4 (6), pp. A74–A77.*

[174] *Kim, U.S., Shin, Ch. B., Kim, C.S. 2009. J. Power Src. 189 pp. 841-846.*

[175] *Newman, J. and Thomas-Alyea, K. E. 2004. Electrochemical Systems., 3rd edn. Wiley. ISBN: 978-0-471-47756-3.*

[176] *Newman, J. and Tiedeman, W. 1993. J. Electrochem. Soc. 140 (7) pp. 1961-1968.*

[177] *Ong, I. J. and Newman, J. 1999. "Double-layer capacitance in a dual lithium ion insertion cell." Journal of The Electrochemical Society., 146 (12), pp. 4360–4365.*

[178] *J. Van Zee, G. Kleine, R. White, J. Newman, "Extension of Newman's Numerical Technique to Pentadiagonal Systems of Equations", in Electrochemical Cell Design, Edited by R. White, Plenum Press (1984).*

[179] *Verbrugge, M. and Tate, E. 2004. J. Power Src. 126 pp. 236-249.*

Battery Field Functions Reference

This section describes the field functions that are made available to the simulation when a battery module is prepared.

Battery Functions

The Battery field functions are available under the **Battery** node in the *Scalar Field Function* selection window.

The names in this table are all prefixed with the label **Battery** in the STAR-CCM+ user interface.

These objects all have the same base properties.

Cell Current	The current, I, passing between the posts of the battery cell (A).
Cell Voltage	The voltage drop, V, between the posts of the battery cell (V).
Electrical Solver Total Heat	The total generated heat from the electrical solver before it is mapped onto the thermal mesh (W).
Electrode Polarization Heat	Volume-based heat generated due to polarization of the two electrodes. This is initiated by the electrochemistry reaction (W/m^3).
Negative Current Density	A vector field denoting the flux of current through the negative electrode for each ecell (A/m^2).
Negative Electrochemical Current Density	A vector field denoting the flux of current through the negative electrode for each ecell in the axial direction of the cell (A/m^2). This field function is available for cylindrical and prismatic cells only.
Negative Plate Voltage	The voltage across the negative electrode for each ecell (V).
Ohmic Heat	Volume-based heat generated by Joule Effect for each ecell (W/m^3).
Positive Current Density	A vector field denoting the flux of current through the positive electrode for each ecell (A/m^2).
Positive Electrochemical Current Density	A vector field denoting the flux of current through the positive electrode for each ecell in the axial direction of the cell (A/m^2). This field function is available for cylindrical and prismatic cells only.

Positive Plate Voltage	The voltage across the positive electrode for each ecell (V).
Single True Cell Current	The current through a single true cell for each ecell (A).
SOC	The battery cell state of charge represents the current battery cell capacity as a percentage of its total capacity (%).
Tab Voltage	The voltage in each tab (V).
Thermal Mass	A material property which is an analogy for the heat capacity, Thermal Mass = Cp x Mass (J/K).
True Cells	Number of true cells in the battery pack. A true cell = 1 positive electrode + 1 negative electrode + 1 separator.
Volts	Equal to (Positive Plate Voltage - Negative Plate Voltage) for each ecell (V).
Volumetric Heat	Volume-based sum of all the generated heat: Electrode Polarization Heat (includes entropic heat) + Ohmic Heat (W/m^3).

Electrode Mask Fields

In order to allow a more detailed post-processing analysis of the battery cells, field functions which indicate the location of the electrodes within the battery volume are available.

Masks > **Battery Negative Electrode**	Provides a scalar value of 1 in the mesh cells where the negative electrode is present. Provides 0 where the negative electrode is not present.
Masks > **Battery Overlap Electrode**	Provides a scalar value of 1 in the mesh cells where both the negative and positive electrodes are present. Provides 0 in all other locations.
Masks > **Battery Positive Electrode**	Provides a scalar value of 1 in the mesh cells where the positive electrode is present. Provides 0 where the positive electrode is not present.

Modeling Region Sources

This section provides some guidelines on how to use region sources for some common problems.

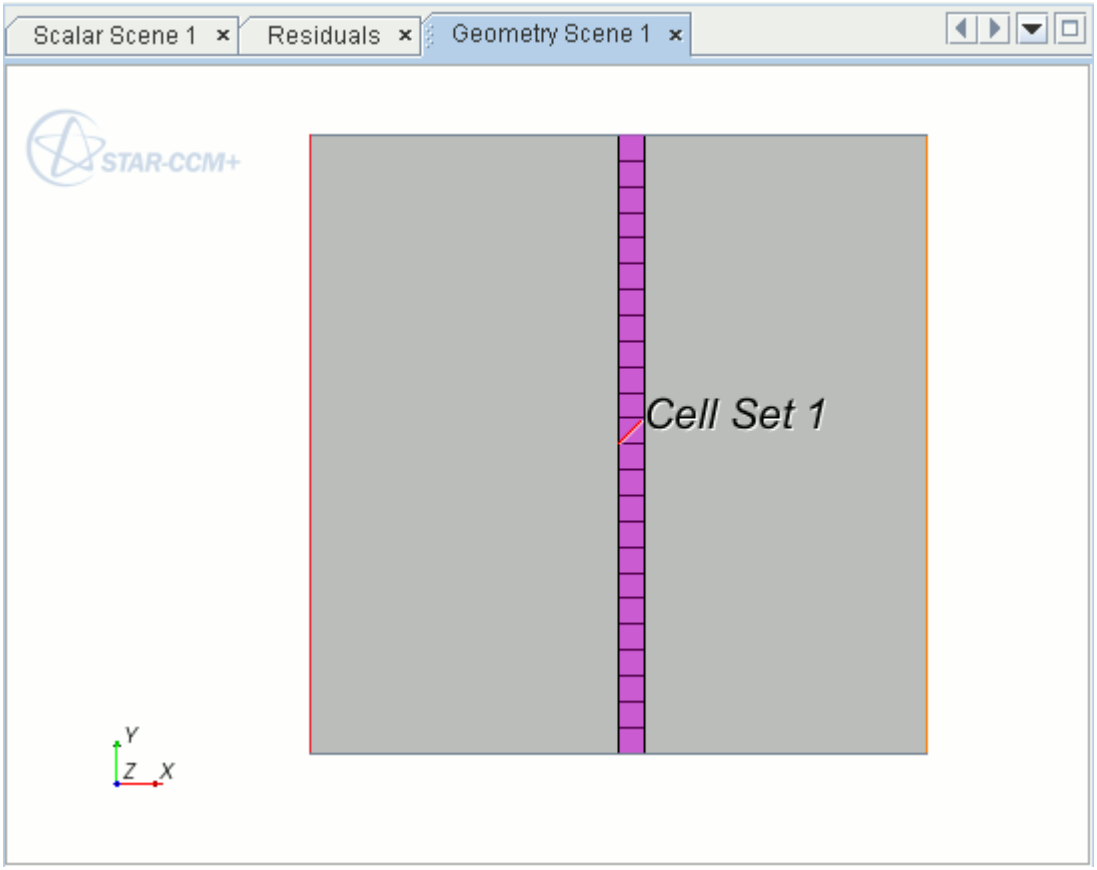
Contents:

Marking Cells for Region Sources
Setting Fluid Sources or Sinks
Prescribing Constant Temperature in Fluid Cells

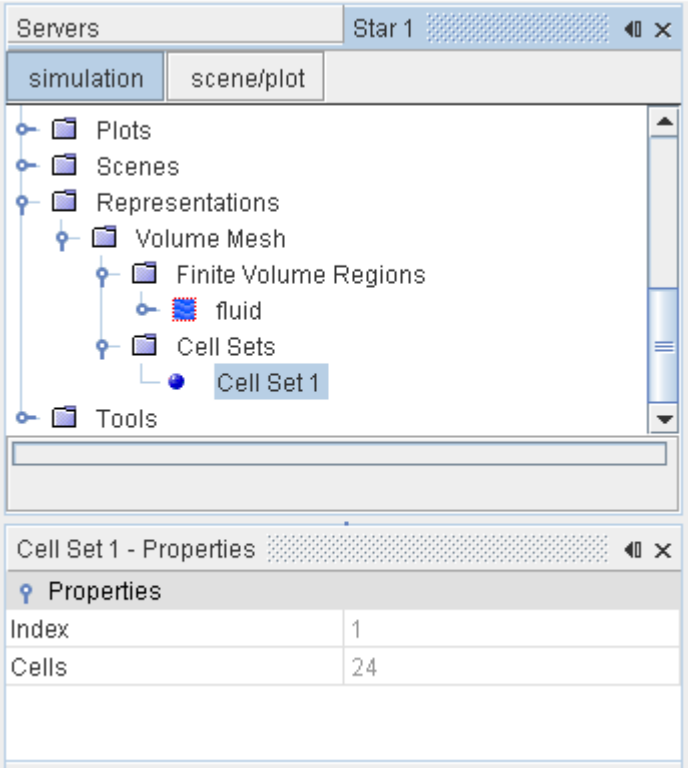
Marking Cells for Region Sources

Typically it is not necessary to apply a region source to an entire region, but to a small zone of cells within the region. It is possible to use cell sets, and their associated field functions, to mark such a region of cells.

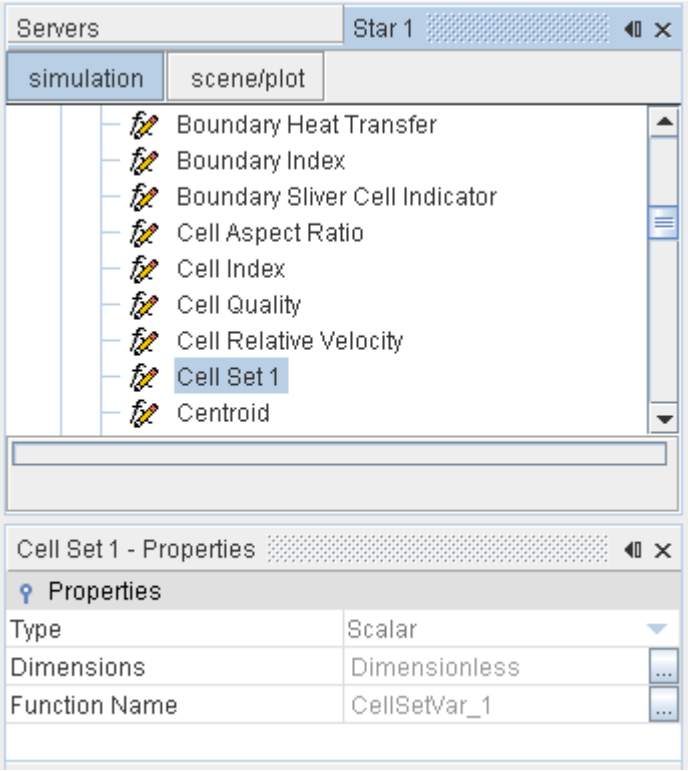
In the example that is shown below, it is desired to set an energy source on a vertical column of cells. The cell set defining this can be created using a geometric range where the limits are set just outside the column of interest.



The cell set was created on the **Volume Mesh** representation.



This cell set, which has been named **Cell Set 1** by default, has a corresponding field function whose *Function Name* is a contracted form of the cell set name. In this instance, the *Function Name* is `CellSetVar_1`.



To use the cell set in another field function expression, simply call `$CellSetVar_1`. This call returns the value of 1 for all cells that are contained within the cell set, and 0 otherwise.

Setting Fluid Sources or Sinks

It is sometimes desirable to inject or extract fluid mass directly in simulations without having to define a geometrical representation of the real aperture that is used to do so. For example, the length scale of the actual geometry of the aperture may be too small to resolve adequately. Therefore, injecting the fluid from a much smaller region is an appropriate method.

In STAR-CCM+, various region sources are provided to make this kind of approach possible. Cell sets can be used to define the cells to which the region source applies.

The general rule to follow in applying this method is that every transport equation active in the region should have a corresponding source or sink term of the form

$$S_\phi = \dot{m}'''\phi$$

where $\dot{m}'''$ is the mass flow rate of the injected/extracted fluid per unit volume, and ϕ is the corresponding transported quantity. For the continuity equation, $\phi = 1$.

For each transport equation, the appropriate region condition, and the number of corresponding value nodes that may be added, are given as follows:

- Continuity: Mass Source Option (providing up to four additional value nodes)
- Momentum: Momentum Source Option (providing up to four additional value nodes)
- Energy: Energy Source Option (providing up to four additional value nodes)
- Turbulence: Turbulence Source Option (providing up to two additional value nodes)
- Species: Species Source Option (providing two additional value nodes)

Caution: If you define a mass source or sink, you must also define a corresponding momentum, energy, turbulence, or any other transport variable source or sink. Otherwise, the physics of your simulation is not sensible. For example, if you define a mass sink without a corresponding energy sink, the temperature of the system increases: the same amount of energy stays in the system, with less mass to hold it.

Prescribing Constant Temperature in Fluid Cells

In modeling some applications, such as air conditioning units, it may be desirable to impose the temperature distribution within a range of fluid cells. This section describes a procedure that uses the discretized form of the energy equation to fix a target temperature that you specify.

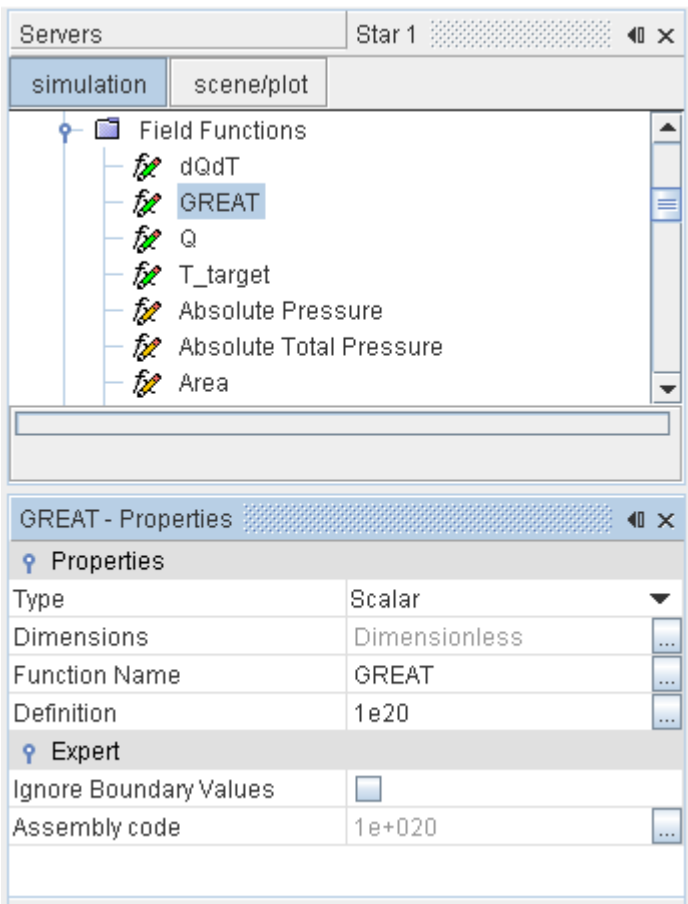
The steps are as follows:

1. Create a cell set containing the cells for which the temperature is to be fixed.

   This cell set has a corresponding field function with a default name. Here, the field function is `CellSetVar_1`.

2. Create a field function, `GREAT`, that contains a number sufficiently large to mask the contribution from the usual energy fluxes in the designated cells, but not too high to cause a floating point exception.

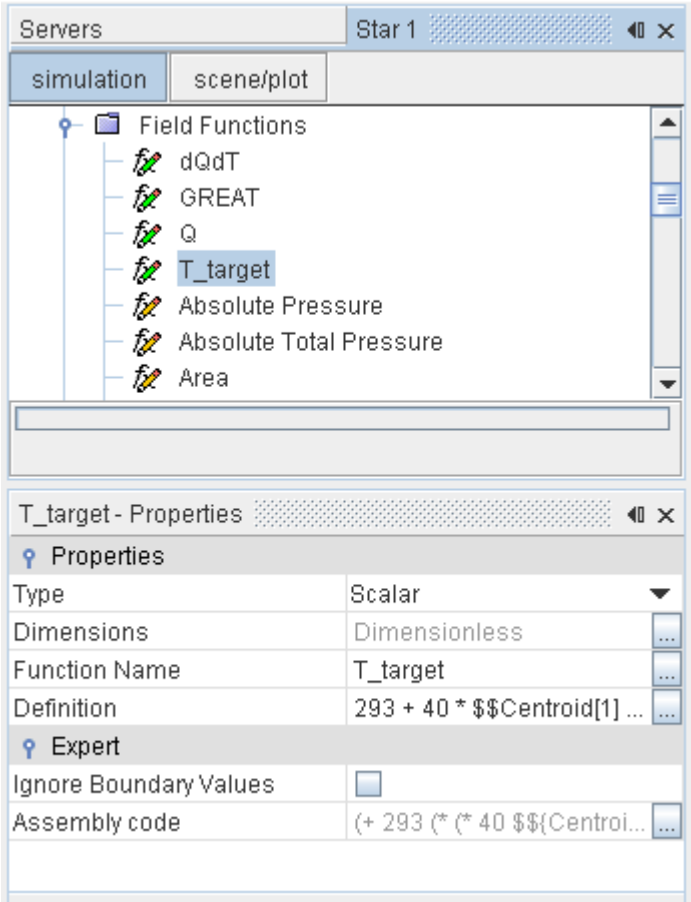
   For example, you could use 1000 times the maximum local value of $\left(mC_p\right)/V$ or kL/V, where m is the mass flow rate into the cell, C_p is the specific heat, V is the cell volume, k is the thermal conductivity, and L is some characteristic cell length. In this example, `GREAT` is set to `1e20`.



3. Create a field function for the user-defined temperature field.

   Here, for example, `T_target` is set to:

```
293 + 40 *$$Centroid[1] * (1 - $$Centroid[1])
```



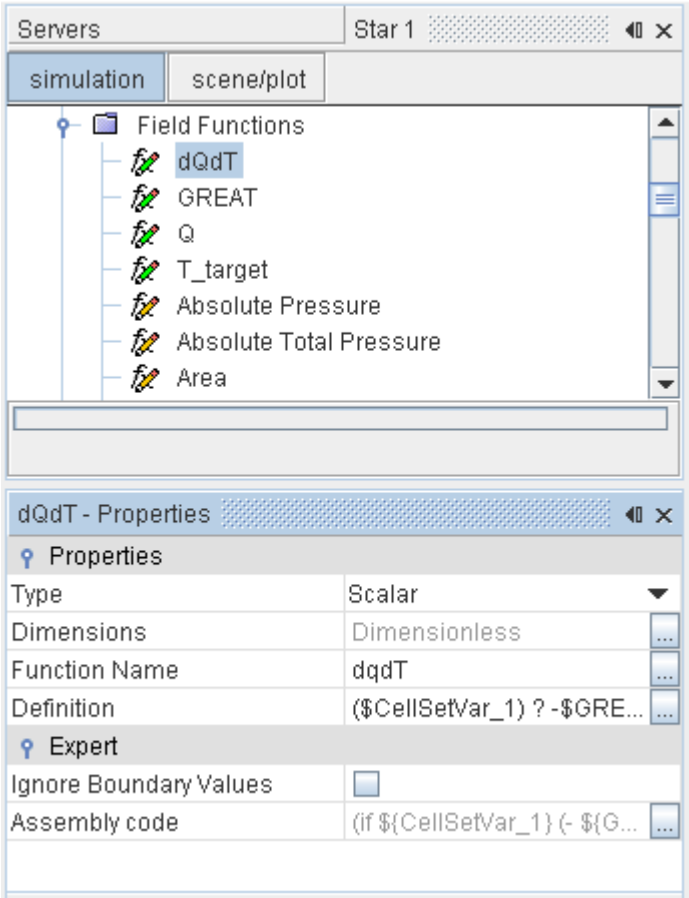
4. Using these three field functions, it is possible to define a further field function for the Volumetric Heat Source, Q, that applies to cells in the cell set:

   ```
   ($CellSetVar_1) ? $GREAT*($T_target - $Temperature) : 0
   ```

5. The differential of this field function with respect to temperature is also required as a field function.

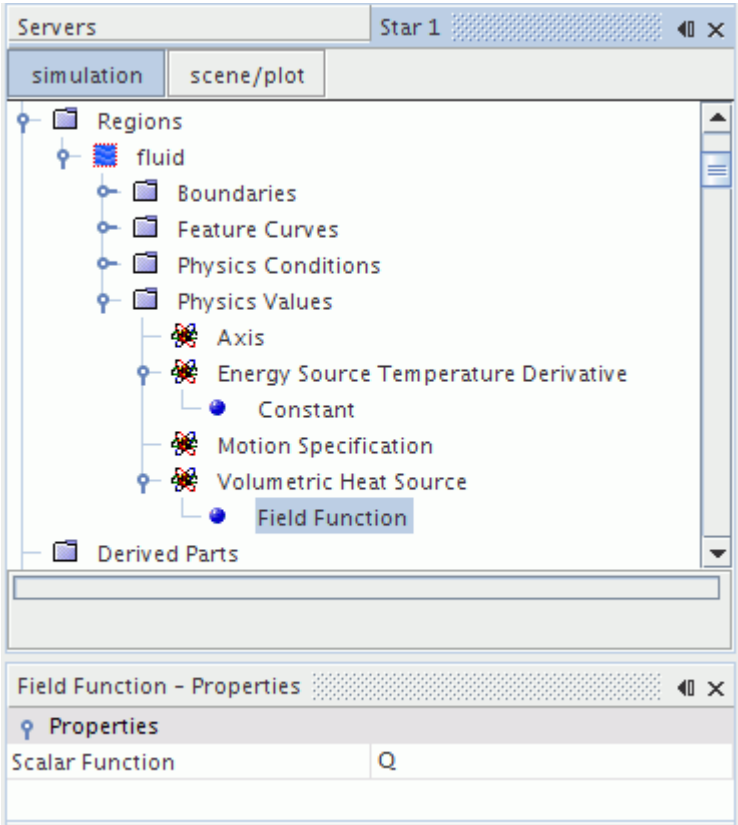
   Here, it is defined as **dQdT**, with the formula:

   ```
   ($CellSetVar_1) ? -$GREAT : 0
   ```

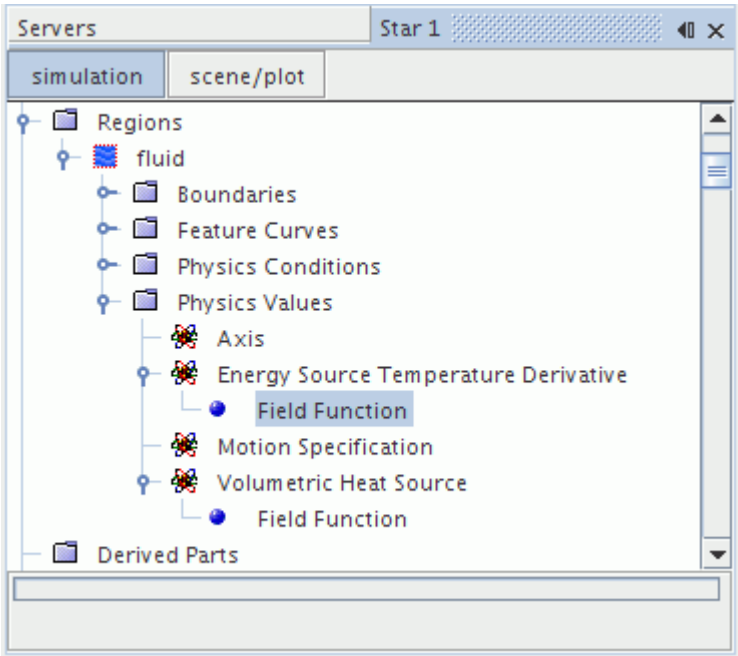


Having created the necessary field functions, the final step is to apply them to the relevant region values:

6. Set the Volumetric Heat Source of the region to the field function **Q**:
7.



8. Set the **Energy Source Temperature Derivative** to the field function **dQdT**:



Remedying Cell Quality

The Cell Quality Remediation model helps you get solutions on a poor-quality mesh. This model identifies poor-quality cells, using a set of predefined criteria, such as Skewness Angle exceeding a certain threshold. Once these cells and their neighbors have been marked, the computed gradients in these cells are modified in such a way as to improve the robustness of the solution.

In general, the effect of Cell Quality Remediation is confined to the immediate vicinity of poor-quality and/or degenerate cells, so that the influence on overall solution accuracy is minimal. However, a side effect is that the mesh quality metrics used to mark the poor-quality cells are recomputed each time the mesh changes. Therefore, in certain situations such as moving meshes, using the model can have a significant computational cost.

Contents:

Troubleshooting
Cell Quality Remediation Field Functions Reference

Troubleshooting

One or more cells have an invalid least squares tensor

You can encounter an error message similar to the following:

```
Invalid least squares tensor in cell with Local Index 193495 in region
inlet on partition 19
```

This error comes from the least squares (reconstruction) gradient calculation. When the least squares tensor is inverted using the Jacobi method, this error indicates that the tensor is invalid, and the algorithm cannot proceed. Since the least squares tensor is derived from geometrical considerations, the error signifies an invalid cell. Diagnose and repair or regenerate the mesh.

You can use the information in the error message to identify the problem:

- Create a user field function with the following definition:

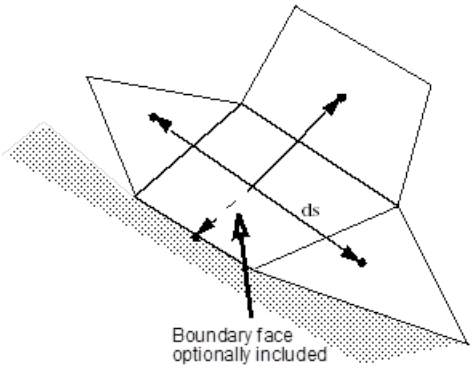
  ```
  ($LocalCellIndex == 193495) && (PartitionID == 19)
  ```

- Create a threshold derived part where this quantity equals 1.

These steps help you visualize the cell. The cell can be flat, and/or have almost no neighboring cells. In addition, proximity to a poorly intersected interface can contribute to the bad cell.

Cell Quality Remediation Field Functions Reference

The following field functions are made available to the simulation when the Cell Quality Remediation model is used.

Bad Cell Indicator	This field function is activated with the Cell Quality Remediation Model, and it is defined as an integer value between 0 and 3: • 0: The good cell has only good neighbors. • 1: The bad cell has good neighbors. • 2: The good cell has bad neighbors. • 3: The bad cell has no good neighbors.
Boundary Sliver Cell Indicator	This field function is used to indicate cells that can be considered “boundary slivers” and hence could be removed from the solution domain without affecting the topology of the mesh. To be considered a boundary sliver, two conditions must be satisfied: 1. All vertices of the cell must lie on a boundary. 2. The Cell Quality must be below the specified tolerance (1e-6 by default), indicating that the cell is too flat. The field function has an integer value of either 0 (not a boundary sliver cell) or 1 (a boundary sliver cell).
Least Squares Quality	This field function is an indicator of the quality of a cell, using the physical location of a cell centroid relative to the cell centroid locations of its face-neighbors. 

	The field function value is based on the ratio of the maximum and minimum eigenvalue ratios of the symmetrical matrix: $$\overline{\overline{M}} = \sum_{\text{cell faces } i}^{n} \frac{d\mathbf{s_i} \otimes d\mathbf{s_i}}{\left\lvert d\mathbf{s_i} \right\rvert^2}$$ where $\mathbf{ds}_i$ represents the vector from the cell to the neighbor centroid. Two methods are optionally provided to evaluate the eigenvalue ratio: approximate and exact. The exact method uses a Jacobi transformation. The approximate method uses the Frobenius norm of the matrix to estimate the eigenvalue ratio. It is slightly more economical to evaluate than the exact method, and is guaranteed to be less than or equal to the exact value. A perfect cell has a Least Squares Quality computed by the exact algorithm equal to unity. The same perfect cell would have an approximate value of 1/N, where N is the dimension of the mesh (2 or 3). As the Least Squares Quality becomes small (< 0.1), the approximate and exact values yield almost identical results. An option also exists to include the boundary faces of the cell in the computation for boundary cells.
Zero Face Area Indicator	This field function has the value of unity if the area of any of the faces approaches zero; otherwise it has a value of zero. It is used to diagnose potential mesh problems.

Controlling Domain Decomposition

To make use of parallel processing, STAR-CCM+ breaks the computational domain into separate sections (called sub-domains) and assigns each to a separate process. This approach is called domain decomposition. Once the domain is decomposed, the processes work in parallel on their individual sub-domains, periodically communicating with each other.

The goals of domain decomposition are to balance the work as evenly as possible among the processes and to minimize the amount of communication that is needed between them. You can control domain decomposition through the partitioning solver.

Contents:

The Partitioning Solver

The Partitioning Solver

The partitioning solver is responsible for controlling domain decomposition (or partitioning) for parallel simulations. Before any physics solvers are invoked on a given global iteration or time-step, the partitioning solver is first invoked to ensure that the domain decomposition is up-to-date.

The solver generally repartitions the domain when one of the following conditions is met:

- The cell connectivity has changed materially since the last partitioning.
- The computational loads have become imbalanced.
- The partitioning method is changed.

Contents:

Partitioning Solver Operation
Cell Connectivity Changes
Load Balancing
Partitioning Method
Partitioning Solver Parameters

Partitioning Solver Operation

Changes in inputs to the partitioning solver – including but not limited to the specified partitioning method, cell connectivity, and load balancing settings and weights – are only applied when either of certain conditions are met.

These conditions are:

- The partitioning solver is next run in the normal course of stepping or iterating,

  or
- The partitioning is manually updated with the Repartition Domain action.

The **Repartition Domain** action is included as a convenience; it is not needed to keep the partitioning up-to-date with the inputs. If the partitioning is out-of-date, STAR-CCM+ automatically repartitions the domain on the next global iteration or time-step. The **Repartition Domain** action lets you immediately inspect the impact of repartitioning before advancing the solution.

With respect to save-restore, the partitioning solver generally behaves like other STAR-CCM+ solvers – if the input settings are changed just before saving and exiting the simulation, they will not be applied to the saved simulation until the next time the solver runs after the simulation is restored. Restoring into a different process count only changes the partition count; it does not change the applied partitioning method, which remains the same as it was after the last time the partitioning solver ran before the simulation was saved.

Cell Connectivity Changes

The solver partitions the computational domain according to the connectivity graph between the cells. The solver partitions the graph to balance the work, and to minimize communication between the sub-domains.

The solver usually balances work by balancing the number of computational cells and faces, but it can also consider work-per-cell from specific physics solvers. Communication between sub-domains is minimized by minimizing the number of cut edges in the connectivity graph.

Because performance depends on partitioning and partitioning depends on the cell connectivity, the solver recomputes the partitioning when a tangible change in cell connectivity takes place. For instance, if you add an interface between regions, the interface introduces new cell connectivity between the regions, which requires more information transfer. If the partitioning does not consider this connectivity, the resulting performance can be lower than it need be.

Currently, the domain is repartitioned when a cell connectivity change results from:

- Adding, removing, or modifying either:
  - An intra-region interface (an interface within a region) when per-region partitioning is activated,

    or:
  - An intra-continuum interface (an interface within a continuum) when per-continuum partitioning is activated.
- Splitting a region.
- Combining multiple regions.
- Modifying the mesh within a region such that it modifies the connectivity.

Load Balancing

The second role of the partitioning solver is to repartition the domain when the per-process loads become imbalanced. The goal is to keep loading as evenly balanced as possible.

The partitioning solver:

1. Periodically checks if specific physics computations have become imbalanced.
2. If computations have become imbalanced, the solver repartitions the domain according to the load information that those physics solvers provide.

Load balancing is available for:

- Lagrangian transport

  See Lagrangian Load Balancing .
- Overset mesh

  See Overset Mesh Load Balancing

Because load balancing itself has a cost, it is not always desirable on every global iteration. Usually, the computational solution changes gradually from iteration to iteration, so a partitioning based on the loads at a given global iteration is often suitable for a number of future iterations.

You can control the load balancing frequency using the *Rebalance frequency* property on the **Load balancing options** node within the partitioning solver. Load balancing and the associated **Load balancing options** node only become available when a physics solver that supports load balancing is active.

Partitioning Method

Every simulation has an overall computational domain that can be defined as the union of all the regions in the simulation upon which physics is being simulated.

One approach to partitioning the overall computational domain is a monolithic one. The simulation creates a graph of the entire domain and decomposes it in a single operation, producing a single set of self-contiguous subdomains. This approach, however, can lead to poor performance and scalability in some situations. For example, if one continuum has more complex physics than another, this approach can result in severe load imbalances.

For this reason, partitioning over domains of finer granularity is usually desirable. In STAR-CCM+, you can partition over the following extents:

- Per-region partitioning: Each region is partitioned individually. For each region, a connectivity graph for all the cells in the region is created, and that graph is partitioned to balance the work and minimize partition boundaries for that region. The result is that the cells in each region are divided evenly between processes, and the sub-domains within each region are contiguous.

  Note: For a multi-region continuum, the sub-domains are not contiguous with respect to the containing continuum when per-region partitioning is employed.

- Per-continuum partitioning: Each continuum is partitioned individually. For each continuum, a connectivity graph is created for all the cells in all the regions in that continuum, and that graph is partitioned to balance the work and minimize partition boundaries for that continuum. The result is that the cells in each continuum are balanced, and the sub-domains within each continuum are contiguous.

In STAR-CCM+, per-region partitioning has been the historical approach. While per-region partitioning often balances loads successfully, it sometimes has the following disadvantages:

- The overall decomposition is more discontiguous from a process perspective, ultimately resulting in more inter-process communication in terms of both of data volume and message count.
- Connectivity across inter-region interfaces is not considered in the partitioning process, often resulting in both boundary face imbalances and increased inter-process communication.

These issues may adversely affect both performance and scalability.

Per-continuum partitioning, in contrast, mitigates or eliminates these issues. Because partitioning takes place over a larger extent, the per-process sub-domains are more contiguous. This reduces communication and other parallel overhead relative to per-region partitioning. As a result, per-continuum partitioning can significantly improve both performance and scalability relative to per-region partitioning in many situations.

In general, the benefits of per-continuum partitioning increase with both increasing number of regions per continuum and increasing process count.

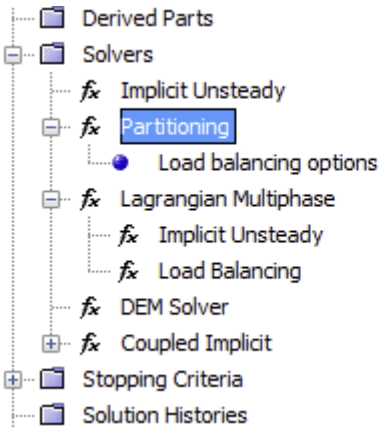
Despite the general benefits of per-continuum partitioning and most applications usually performing better with per-continuum partitioning, some very specific STAR-CCM+ application types can still sometimes benefit from

per-region partitioning. For instance, some harmonic balance applications still perform better with per-region partitioning at this time.

A change in partitioning method goes into effect the next time the partitioning solver runs. To apply the specified method immediately, use the **Repartition Domain** item from the partitioning solver popup menu.

Partitioning Solver Parameters

The partitioning solver can be found at the usual solver location in the simulation object tree.



Expert Properties

Partitioning method	Controls how the solver partitions the domain. A change in partitioning method becomes effective the next time the solver runs. See Partitioning_Method.	
	Per-Region	Partitions each region individually.
	Per-Continuum	Partitions across all regions in each continuum. This is the default for new simulations.
Solver Frozen	Controls whether the Partitioning solver is active or inactive (frozen).	
	Activated	Partitioning solver is inactive or frozen.
	Deactivated	Partitioning solver is active (the default).

Partitioning Solver Pop-Up Menu Items

Repartition Domain	Immediately repartitions the domain according to the current mesh connectivity. This action is seldom needed because, if the partitioning is out-of-date, the domain is automatically repartitioned on the next global

	iteration or time-step. The action is included as an option to update the partitioning manually outside the context of advancing the solution.

Contents:

Load Balancing Options Node Properties

Load Balancing Options Node Properties

The **Load balancing options** node, which is located under the partitioning solver in the tree, becomes available only when a physics solver supporting it is active.

Rebalance frequency	Controls the frequency of load balancing in terms of global iterations or time steps. The default frequency is: • Once per iteration for steady state problems. • Once per time step for unsteady problems.

Guidelines for Applying STAR-CCM+

This part of the documentation provides guidelines on applying STAR-CCM+ models to your applications.

Contents:

Guidelines for Fluid Flow and Energy
Guidelines for Aeroacoustics Calculations
Guidelines for Aerodynamics Calculations
Guidelines for Simulating Air Flow in Electronic Systems
Modeling Fluid-Structure Interaction
Modeling Atmospheric Boundary Layers
Preparing CAD Models for Import
Migrating from Other Codes

Guidelines for Fluid Flow and Energy

You can apply these guidelines to a wide range of simulations that involve fluid flow and energy.

Mesh

The construction of the volume mesh has a direct influence on how well STAR-CCM+ simulates fluid flow and energy. It influences the rate of convergence and the accuracy of the final solution. In general, construct a volume mesh according to the following guidelines:

- Design a mesh that gives adequate resolution in regions where spatial gradients are high. Such regions occur when:
  - The mean flow changes rapidly. Examples are: irrigation canals, vortex shedding, forced mixing, and atmospheric flows.
  - There are strong shear layers. Examples are: atmospheric flows, fluid jets, flows past solids, and flows with strong vorticity.
- Whenever possible, align the mesh with the flow to improve accuracy and increase the rate of convergence.
- In order to make sure that simulation results are independent of the mesh, perform a grid sensitivity study with two or more meshes.

  Note: Turbulence has a dominant role in the transport of momentum and other scalars for most complex turbulent flows. For this reason, simulations that involve turbulent flow tend to be more dependent on the mesh resolution than simulations that involve laminar flow.

- For external flows, perform domain convergence studies to make sure that what is happening at the flow boundaries does not affect the solution. For example, if the simulation has an outflow boundary close to a bluff body, the flow cannot recover from the effect of the wake before it meets the boundary.

Near-Wall Prism Layers

- In order to minimize numerical diffusion in the near-wall region, create layers of prism cells on walls and the fluid side of solid-fluid interfaces.
- The thickness of prism layers next to the wall has a direct impact on fluid y+ values. Check the y+ values in the solution, and make sure that they come within the range of validity recommended for the turbulence model and wall treatment that you have selected.
- For high accuracy modeling of boundary layers, create 5–10 cell layers within the boundary layer when using wall functions for the turbulence model. Create around 40 cells in the boundary layer when using a low-Reynolds number turbulence model.

Cell Aspect Ratio

- Avoid creating a mesh that has excessive stretching in the direction normal to a wall.

- Keep cell aspect ratios as small as possible (for example, below 1000) to avoid convergence problems in the solver. Some problems require lower cell aspect ratios, such as simulations that involve solid stress calculations.

Mixing Applications

These applications typically have two regions: a stationary region and a region with a rotating reference frame. A polyhedral mesh can have cell faces that are not aligned with the flow, which can introduce numerical diffusion. In some simulations, this numerical diffusion can produce a significant discontinuity in the volume fraction at the interface between the stationary and rotating regions. To eliminate this discontinuity, use trimmed meshes for these applications. Trimmed meshing produces mesh that is better-aligned with the interface and minimizes numerical diffusion in the simulation.

Thermophysical Properties

The following guidelines apply only when you use the energy equation in the simulation:

- Include the effects of temperature on thermophysical properties such as viscosity, specific heat, and thermal conductivity, when you have the information.
- When using temperature-dependent thermophysical properties, make sure that relevant dimensionless numbers maintain the values that you expect them to have. Examples of dimensionless numbers are the Reynolds number and the Prandtl number. As the Prandtl number appears explicitly in temperature wall functions, wall heat transfer predictions are sensitive to the value of thermophysical properties.
- When you use constant thermophysical properties, that is, independent of temperature, make sure that you specify all physical properties at the same temperature.
- When the Prandtl number is small, (for example, less than 0.1), use a mesh suitable for low values of y+ at the wall.
- When you specify surface properties for radiation heat transfer, note that in some circumstances the properties of real surfaces change significantly during service. For example, it is not unusual for the emissivity of an aluminized shield to increase several times through oxidation, dust, or the presence of moisture. This change of emissivity is not automatically captured by STAR-CCM+: if you wish to simulate such effects you must alter the surface properties.

Boundary Conditions

Use the following guidelines for defining the boundary conditions.

- Choose the shape and location of boundaries so that flow passes through a boundary in one direction only, that is, entering or leaving the domain. Avoid any recirculation across a flow boundary.
- As far as possible, choose a static pressure boundary for the outlet boundary.
- When possible, extend the computational domain to have flow regions upstream and downstream of the main region of interest. Extending the computational domain in this way minimizes the effect that boundary conditions have on the solution within the main region of interest. You can specify a coarser mesh in extended regions, so that the computational requirements are not increased unduly.
- In some simulations, the solution does not converge with the boundary conditions you require. When this situation occurs, start with a different set of boundary conditions and change them as the simulation

proceeds. For example, when solving high-Mach number flows over bluff bodies, gradually increase the free-stream Mach number instead of starting with the final value.

- Choose boundary types that you know are compatible with the type of flow you are simulating. For example, a velocity inlet boundary is not a well-posed type for a highly compressible flow.

Conjugate Heat Transfer (CHT)

For applications that involve simultaneous heat transfer in both a solid and a fluid, there are two options for obtaining thermal solutions in the solid and the fluid. These options are:

- STAR-CCM+ can obtain solutions in both the solid and the fluid.
- STAR-CCM+ can obtain the flow and thermal solution in the fluid only, and pass thermal information to another application that solves for the thermal solution in the solid. An example of this arrangement is where STAR-CCM+ exchanges thermal data with Abaqus in an explicitly coupled simulation.

Modeling Conjugate Heat Transfer

For applications that involve simultaneous heat transfer in both a solid and a fluid, such as modeling the fluid and solid in a heat exchanger, the following apply:

- Fluid and solid equations are implicitly coupled and are solved simultaneously.
- The following surface conditions (temperature and energy) are enforced at the fluid-solid interface:

$$T_{solid}|_{surface} = T_{fluid}|_{surface}$$

$$q_{solid}|_{surface} = q_{fluid}|_{surface}$$

$$-K\frac{\partial T_{solic}}{\partial n}\bigg|_{surface} = h_f\left(T_{surface} - T_{ref}\right) + q_{rad} + q_c + \cdots$$

where:

- n is the surface outward normal
- K_s is the solid conductivity
- h_f is the fluid heat transfer coefficient
- T_{ref} is the characteristic fluid temperature (for example, the domain mean or near wall cell temperature)
- q_{rad} and q_c correspond to net radiative heat flux and added heat flux due to phase change, respectively.

Using Co-Simulation

You can use a single STAR-CCM+ simulation to obtain thermal solutions in both the solid and the fluid, where the fluid and the solid are implicitly coupled at the interface. Alternatively, you can use co-simulation to obtain each solution in a separate simulation, where the fluid and the solid are explicitly coupled at the interface and exchange data back and forth as the simulation progresses.

For more information, see STAR-CCM+ to STAR-CCM+ Co-Simulation.

You can use co-simulation when the fluid and solid time scales are significantly different. For example, if the flow is strongly turbulent and well-mixed, and solid conduction is slow, you can assume that the flow

adjusts to the changes in the wall temperature quickly. This adjustment is essentially instantaneous with respect to the solid time scale, so the fluid-only simulation can be run steady and the solid-only simulation can be run unsteady. This arrangement allows you to update the fluid solution only when the solid wall temperature has changed significantly.

Co-simulation can provide significant run-time savings, with recent results showing significant speed-ups.

If you use co-simulation, be aware of the following issues:

- In co-simulation, convergence problems can arise because the fluid and solid are explicitly coupled. Ensure that:
  - The correct information is passed between the simulations.

    For example, the heat transfer coefficient and its corresponding reference temperature are passed from the fluid-only simulation to the solid-only simulation. The wall temperature is passed from the solid-only simulation to the fluid-only simulation.

    Although there are other choices of boundary conditions to pass between the simulations (for example, you could pass the wall heat flux from the fluid-only simulation), for numerical stability reasons their use is not recommended.

    The recommended heat transfer coefficient and corresponding reference temperature are:
    - Fluid flow with no radiative transport or phase change:

      Use the Specified Y+ Heat Transfer Coefficient and Specified Y+ Heat Transfer Reference Temperature. Start with a y+ of 100, but if needed for better convergence behavior, you can try a different y+. Increasing y+ decreases the Specified Y+ Heat Transfer Coefficient while decreasing y+ increases the Specified Y+ Heat Transfer Coefficient.
    - Fluid flow with radiative transport and/or phase change:

      Use the Local Heat Transfer Coefficient and Local Heat Transfer Reference Temperature. If necessary, you can linearly transform this pair to another pair that has better convergence behavior.
  - The correct co-simulation parameters are used. For example, the number of iterations, time-step, and update frequency.
- Loss of accuracy. The amount of accuracy loss strongly depends on the co-simulation parameters that are used and the amount of speed-up desired.

Co-Simulation Time Scales

In a co-simulation, the fluid and solid systems are solved independently: a separate system of equations is solved for each of the fluid and solid regions. Some of the most common time scales for the fluid region are as follows:

- Diffusion time scale:

$$t_{f1} = \frac{{L_f}^2}{\nu}$$

- Inertial time scale:

$$t_{f2} = \frac{L_f}{U_f}$$

- Internal/external forcing time scale:

$$t_{f3} = \frac{1}{\omega_f}$$

The time scale for the solid region is:

- Diffusion time scale:

$$t_s = \frac{{L_s}^2}{\alpha}$$

The time scale ratios are:

$$\left[\frac{t_s}{t_{f1}} \frac{t_s}{t_{f2}} \frac{t_s}{t_{f3}}\right] = \left[\left(\frac{{L_s}^2 v}{{L_f}^2 \alpha_s}\right)\left(\frac{{L_s}^2 U_f}{L_f \alpha_s}\right)\left(t_s \omega_f\right)\right]$$

where:

- L_f and L_s are characteristic length scales (such as domain or grid size) for the fluid and solid, respectively
- v is the fluid kinematic viscosity
- U_f is a characteristic velocity scale (for example, domain mean or cell velocity)
- ω_f is a characteristic forcing frequency
- $\alpha_s = \frac{K_s}{\rho C_p}$ is the solid diffusivity.

Run Time Speed-Up

The run time of an implicitly coupled (unsteady) CHT simulation is:

$$t_{total}^{IC} = \frac{t_{final}}{\Delta t}\left(N_{it} \tau_{it}\right)$$

where:

- t_{final} is the simulation ending time
- Δt is the time-step
- N_{it} is the number of inner iterations per time-step
- τ_{it} is the CPU time per inner iteration.

The run time of an explicitly coupled CHT co-simulation is:

- Fluid (steady): short time scale
- Solid (unsteady): long time scale

$$t_{total}^{EC} = \left(\frac{t_{final}^s}{t_{ex}^s}\right)\left[\left(N_{it}^f \tau_{it}^f\right) + \left(\frac{t_{ex}^s}{\Delta t^s}\right)\left(N_{it}^s \tau_{it}^s\right)\right]$$

where:

- t_{final}^s is the solid simulation ending time
- t_{ex}^s is the exchange period between the solid and fluid simulations
- N_{it}^f and N_{it}^s are the fluid and solid number of inner iterations, respectively

- τ_{it}^{f} and τ_{it}^{s} are the fluid and solid run times per inner iteration, respectively
- Δt^{s} is the solid simulation time-step.

In general, the speed-up in run time of an explicitly coupled CHT co-simulation when compared to an implicitly coupled CHT simulation is:

$$\frac{t_{total}^{IC}}{t_{total}^{EC}} = \frac{N_{it}\tau_{it}t_{ex}^{s}}{\frac{\Delta t}{\Delta t^{s}}\left[\Delta t^{s}N_{it}^{f}\tau_{it}^{f} + t_{ex}^{s}N_{it}^{s}\tau_{it}^{s}\right]}$$

Setting the Co-Simulation Parameters

When you set up a CHT co-simulation, you specify various co-simulation parameters, including:

- Mapping Mode
- Iteration Count
- Update Frequency
- Time Step

For the best results, use a conformal mesh at the solid to fluid interface. A conformal mesh is one whose vertices and faces match exactly on either side of an interface.

Setting the Initial Parameter Values

After setting the initial settings for these parameters, run the co-simulation. If necessary, you can perturb near these initial settings and look for combinations that give better results (that is, a set of parameters that give good accuracy and short run times).

Fluid-Only Simulation (Steady)

- Iterations

  Set the number of iterations, which in general can be a strong function of time, to whatever it takes for convergence (that is, for the residuals to stop changing).

Solid-Only Simulation (Unsteady)

- Update Frequency

  Set the initial update frequency to:

  $$f_{s} = \frac{1}{\beta t_{s}} = \frac{\alpha_{s}}{\beta L^{2}} = \frac{K_{s}}{\rho_{s}C_{p,s}L^{s}\beta}$$

  where:

  - L is a characteristic length scale for the solid (for example, overall length, half length, grid size)
  - β is a scaling factor less than one (for example, 0.2 is a good first choice).
- Time-step

  Set the solid-only simulation time-step Δt^{s} to be some fraction of the update period. For example, $\beta t_{s}/10$ can be appropriate if the solution is smooth and not too steep.

Alternatively, you can slave the update period βt_s to a particular amount of change in wall temperature (for example, the maximum change in one cycle). This allows the update period to adjust itself and the time-step as the simulation progresses.

- Iterations

  Set the number of inner iterations to whatever it takes for convergence.

Cyclone Separators

This section gives some guidelines for modeling cyclone separators. When you are modeling cyclone separators, consider the following:

Meshing

In order to minimize unwanted numerical diffusion effects, the best type of mesh is a hexahedral type with the mesh elements aligned with the circumference of the cyclone.

You can use the Directed Meshing feature to construct this mesh within STAR-CCM+.

Turbulence Model

An anisotropic turbulence model is required to model accurately the free to forced vortex transition that occurs in cyclonic flows. Standard K-Epsilon models and other models that are based on assumptions of isotropic turbulence are not suitable. These models tend to over-predict the turbulent viscosity and exaggerate the forced vortex.

Boundary Layer Resolution

Wall resolution is not critical as the turbulence is generated in the main flow.

Freezing the Solvers

If used correctly, freezing the solvers can be of great use in promoting initial simulation stability, computational efficiency, and total run time. There are two primary situations in which freezing solvers can help:

- Early in the simulation, convergence issues can arise because of the lack of a good physical initial condition for transient simulations or initial iteration for steady simulations. At this point, you can activate solver verbosity so that you see exactly what each solver is doing and where problems occur. Freezing the solvers that are associated with more complex physics can allow you to obtain approximate converged solutions. Thereafter you can unfreeze the solvers one by one, continuing the solution each time, to see if any convergence issues remain.
- Later in the simulation, because of differing physical time scales or for purely numerical reasons, some solvers reach their asymptotic states much faster than others and are no longer significantly changing the solution.

  For instance, in a typical conjugate heat transfer problem, the flow time scale can be small and the solid conduction time scale can be large. Consider, for example, the flow of hot water in a thick-walled pipe. The temperature of the inner pipe wall quickly reaches the temperature of the hot water; the flow solution stabilizes to a steady flow condition. If the only remaining solution concerns the thermal distribution within the pipe wall, then the flow solvers can be frozen while the energy solver runs.

For quantities that have reached their asymptotic states in steady simulations, you can freeze the solvers for the remainder of the simulation. Solvers can be frozen for the remainder of steady simulations. For unsteady simulations where transient effects occur, do not keep solvers frozen for a time longer than the time-scale of the transient effects. (That is, freeze the solvers for a short time and then unfreeze them to see if their asymptotic states have changed. Repeat the procedure as needed). This workflow is similar in nature to co-simulation but is not as flexible.

Guidelines for Aeroacoustics Calculations

This section provides guidelines for conducting aeroacoustics analyses in the following applications.

You must use the double precision version of STAR-CCM+ for all aeroacoustic and hydroacoustic analyses.

To understand this requirement, consider the following example scenario:

Assume that you have a 60 dB sound in a Mach 0.3 turbulent flow. The atmospheric pressure p_{atm} is 101325 Pa, and the mean flow velocity U is 100 m/s.

The pressure fluctuations p' are of the order of 0.02 Pa, and the velocity fluctuations u' are of the order of 5×10^{-5} m/s.

Therefore:

- p'/p_{atm} is of the order of 2×10^{-7}
- u'/U is of the order of 5×10^{-7}

Consider a typical velocity $U = 102.2693842736423$ m/s.

- The first two digits (100) represent the contribution from the mean convective flow.
- The next five digits (2.2693) represent the contribution from turbulence.
- The remaining digits (0.0000842736423) represent the contribution that is responsible for the sound.

Clearly, an accuracy of greater than eight decimal places is required to calculate the acoustic component. To achieve this accuracy, a double precision version of STAR-CCM+ must be used.

Contents:

Automobile Aerodynamics
Sunroof Buffeting
Radial Compressors
Axial Fans
Choosing DES, LES, or URANS for Aeroacoustics
Mesh Issues
Freestream Boundary Condition Issues
Acoustics in Liquid Media
Post-Processing the Acoustic Pressures and Pressure Perturbations
Exporting Data to LMS Virtual.Lab Acoustics
Exporting Data to wave6

Automobile Aerodynamics

At higher speeds, the aeroacoustic noise dominates noise from the engine, exhaust, and tires.

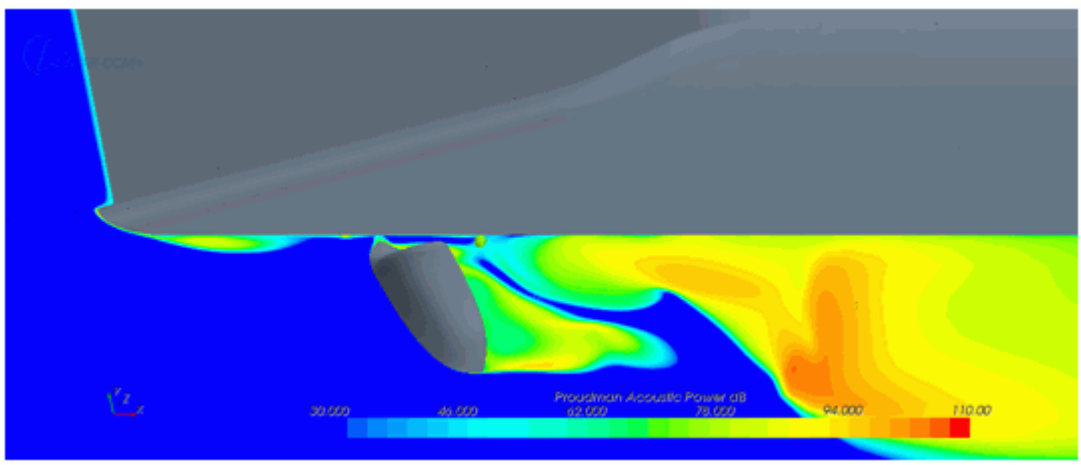
The aeroacoustic noise comes from air flows over pillars, wing-mirrors, and the underbody, and by flow through ducts (including fan noise). This section presents the following recommended practices to model automobile aerodynamics:

- Recommended Process
- Recommended Mesh Settings
- Recommended Time-Step
- Recommended Solver Settings

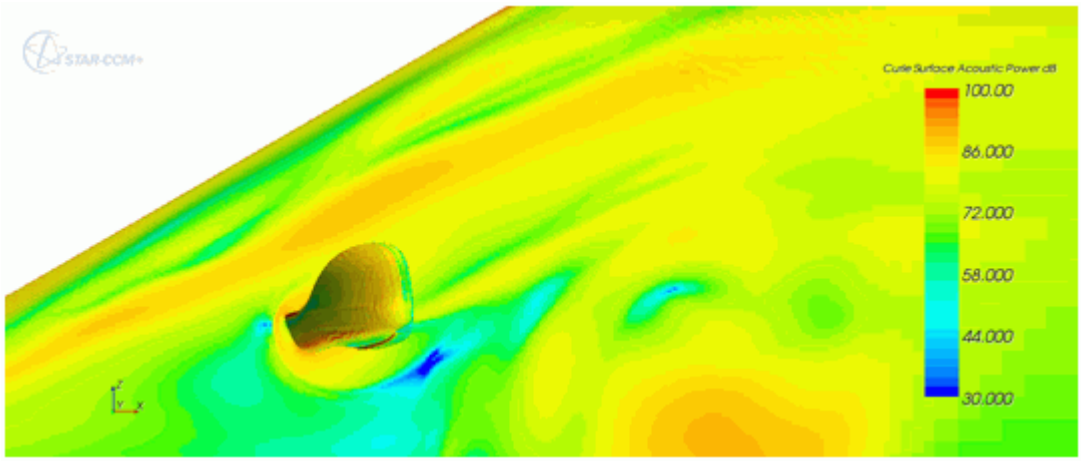
Recommended Process

1. Apply guidelines for mesh-resolution for a RANS-based steady state aerodynamics case. It is vital to get the mean flow correct. Compromise the wake refinement only if rear-vehicle aeroacoustics are not important.
2. Plot the mesh cut-off frequency to judge if the refinement in the acoustic source regions resolves frequencies in the region of 3–5 kHz. In practice, a mesh cell size of 2 mm or less is required. Refine appropriately and map the solution to the refined mesh.

   You can identify the acoustic source regions from the regions where the acoustic correlations are large. Use Proudman for volume sources:



   and Curle for surface sources:



3. Ensure that the mesh in the near field is adequate for local sound propagation. For example, use 3.5 mm or less in the region of the A-pillar, side-view mirror, and side glass to propagate a 5 kHz signal.
4. Choose a time-step size suitable for the most demanding (that is, smallest) of the following constraints: frequency resolution, explicit CFL, or implicit CFL (as appropriate) and Courant number.
5. Restart a transient case from the initial steady state simulation that you used to judge the previous requirements. Switch the turbulence model from RANS to DES or LES. Use second-order time discretization, and a spatial discretization that is based on central differencing.
6. Select and fine-tune the numerical parameters to ensure convergence within each time-step. You can conclude that the solution has converged within a time-step when all (not a subset) of the following criteria are satisfied:
   a. Monotonically reducing or minimized residuals at the end of the time-step.
   b. An unchanging monitor. The monitor can be: pressure at a point, lift, or drag.

Recommended Mesh Settings

Select cell size, Δ, based on criteria:

Criterion	Comments / Description	Formula	Typical Values
Mean Flow	Use guidelines for external steady aerodynamics. Refine in the near-wall region to capture separation from the body.		$\Delta_{\min}$= 2 mm y+ = 1 for low-Re wall treatment
Turbulence integral scales, $L_{integral}$	A measure of the “typical” size of dissipative eddies. Approximated from steady state, $\Delta \leq 0.5\, L_{integral}$	$L_{integral} = \frac{C_\mu^{0.75} \cdot k^{1.5}}{\epsilon}$	$L_{integral}$ > 10 mm $\Delta \leq 2$ mm

Criterion	Comments / Description	Formula	Typical Values
Frequencies that are associated with turbulent structures	Mesh cut-off frequency, that is based on local turbulent fluctuations compared to mesh size recommended in guidelines. Approximated from steady state.	$F_{MCF} = \frac{\sqrt{\frac{2}{3}k}}{2\Delta}$	Δ= 4 mm captures frequencies in the range 1000–1500 Hz.
Propagation of pressure waves	Based on second order space and time discretization, 20 cells per acoustic wavelength, λ (less for higher-order spatial schemes). a, is the speed of sound.	$\lambda = \frac{a}{F}$ $\Delta \leq \frac{\lambda}{20}$	$\Delta \leq 3.5$ mm. Based on a requirement to propagate a pressure wave of F = 5000 Hz in air.

Recommended Time-Step

Select the time-step size Δt, based on criteria:

Criterion	Comments / Description	Formula	Typical Values
A signal of frequency F	The Nyquist sampling theorem states that a signal with frequency, F, can be resolved by sampling at a rate greater than $2F$. For second-order spatial schemes, tests show that around 15 points are required to resolve the wave amplitude.	$\Delta t_F \leq \frac{1}{15F}$, recommended	Δt_F = 2.22E-5 s. Based on requirement to capture F = 3000 Hz.
Convection and acoustics	Based on the convection speed, U, and speed of sound (acoustic pressure waves propagate at the local speed of sound, a). The Courant (CFL) number is based on $(U + a)$	Explicit: $\Delta t_{exp} = \frac{\Delta}{(U + a)}$ Implicit: $\Delta t_{imp} = CFL \cdot \Delta t_{exp}$	Δt_{exp} = 5.0E-6 s. Based on Δ = 2.0 mm, U = 50 m/s and a = 350 m/s. Δt_{imp} = 5.0E-5 s. Based on CFL = 10.

Criterion	Comments / Description	Formula	Typical Values
Convection only	Based on convection speed, U. Convective Courant Number is based on U only.	$\Delta t_{Courant} = \frac{\Delta}{U}$	$\Delta t_{Courant}$ = 4.0E-5 s. Based on Δ = 2.0 mm and U = 50 m/s.
Diffusive time-scales	Most important in the boundary layer where the local (eddy) viscosity, V_{eff}, is largest. Unimportant to acoustic phenomena	$\Delta t_{diff} = \frac{\Delta_{normal}^2}{V_{eff}}$	Δt_{diff} = 1.0E-6 s. Based on Δ_{normal}= 5.0E-6 m, V_{eff} = 2.0E-5 m^2/s.

Recommended Solver Settings

You can select solver settings for a slower (“regular”) mesh and time-step setup, a faster (“aggressive”) setup, or the default settings.

	Velocity URF	Pressure URF	# Iterations Per Time step
Default	0.8	0.2	5
Regular Δt = 1.0E-4 to 1.0E-5 s. Δ_{min} = 1–4 mm.	0.7	0.7	4–10
Aggressive Δt = 2.0E-5 s. Δ_{min} = 2 mm.	1.0	0.9	3

Sunroof Buffeting

At low speeds, flow resonance can occur when the shear layer coming off the front of a vehicle passes over its open sunroof and generates periodic pressure fluctuations within the cabin.

This phenomenon is known as sunroof buffeting, and can be modeled:

- Recommended Process
- Recommended Mesh Settings
- Recommended Time-Step
- Recommended Solver Settings

Recommended Process

The recommended process is:

1. Assess the resonance of the modeled cabin volume by performing a FRET (**F**requency **RE**sponse **T**est):
   a. Apply a pressure disturbance to the sunroof aperture with no flow using an initial condition and field function.
   b. Run the compressible solver with the same settings as a flow case to set up a compressible standing wave in the cavity.
   c. Perform an FFT of the internal pressure-time signal, to identify the cavity resonance frequencies.
2. The resonance of the real vehicle cavity (from experiment) can be different from the FRET for various reasons, namely: panel impedance, leakages, or surface attenuation. Use an artificial compressibility factor, ϕ, based on the ratio of experimental CFD frequencies to modify the response of the modeled cavity as follows:

$$a' = \phi a_0 \tag{1192}$$

$$T'_{in} = \phi^2 T_{in} \tag{1193}$$

$$M' = \frac{M_0}{\phi} \tag{1194}$$

$$p'_{ref} = \phi^2 p_{ref} \tag{1195}$$

w$a_0$here:

- is the speed of sound in the fluid.
- T_{in} is the inlet temperature.
- M_0 is the mass of the air.
- p_{ref} is the reference pressure.

Recommended Mesh Settings

Select cell size, $\Delta_{shear-layer}$, based on criteria:

Criterion	Comments / Description	Formula	Typical Values
Shear layer	Full length of aperture and height of shear-layer oscillations.		$\Delta_{shear-layer}$ = 4 mm

Recommended Time-Step

Select the time-step size Δt, based on criteria:

Criterion	Comments / Description	Formula	Typical Values
Buffeting frequency F_B	Shear-layer oscillates over the opening. Buffeting frequency F_B, is directly proportional to the free-stream speed, U_∞, and is in the range 20-30 Hz.	$F_B = \frac{U_\infty}{3L}\left(N - \frac{1}{4}\right)$ where: N = oscillation mode L = length of sunroof aperture	Δt_F= 1.0E-4 s.

Recommended Solver Settings

The recommended solver settings are:

	Velocity URF	Pressure URF	# Iterations Per Time-step
Default	0.8	0.2	5
Regular Δt = 1.0E-4 s. Δ_{min} = 1–4 mm.	0.7	0.5	5–10
Aggressive Δt = 1.0E-4 s. Δ_{min} = 1–4 mm.	0.9	0.7	5

Radial Compressors

Compressors are usually open to the air side of engines. The generated noise is audible outside the engine. Noise from compressors can be associated with blade passing, blade stall, and other dynamic flow mechanisms.

This section presents recommended practices to model acoustic sources in radial compressors:

- Recommended Process
- Recommended Time-Step
- Recommended Solver Settings

Recommended Process

Qualify Using Steady State

Before running an unsteady analysis, it is vital to first qualify the model by ensuring that the fan performance curve is predicted well from steady state simulations. Use the Realizable K-Epsilon or the K-Omega SST turbulence models for consistency with the DES model in the unsteady analysis.

To avoid pressure reflections, set a free-stream boundary at either the inlet or the outlet, depending on which side is most important. Having a free-stream boundary on both inlet and outlet is unstable in a steady analysis. If it is necessary to control the mass flow, then set the mass flow on the downstream boundary, since acoustics on the inlet side are often more important.

Use Compressible Flow

It is essential to use compressible flow modeling because of the operating conditions of these machines. The peak Mach number is generally in the supersonic range, and a 'choke' regime is present due to a passage shock. Choke occurs when a shock wave spans the full space between the blade pressure and suction sides.

Recommendations for Obtaining Performance Curves in Steady State

1. Initialize the flow field using GSI (Grid-Sequencing Initialization).
2. For high speed radial compressors, validations show that the mixing plane interface is superior to the frozen rotor.
3. The segregated solver is preferred to the coupled solver due to the follow-on step to transient where it is definitely faster. Prediction accuracy between the two is marginal.
4. Marginal differences exist between high and low Reynolds number resolution on the blade. A high Reynolds number wall treatment is preferred as the mesh can be coarser and the grid sequencing initialization is more effective. Practice shows that low Reynolds number meshes produce more accurate predictions. You can refine the mesh and interpolate between high and low Reynolds number predictions.
5. The preferred boundary condition combination is to use stagnation inlet and static pressure outlet. Start with a low pressure ratio between the inlet and outlet. To get points at higher pressure ratios, increase the back pressure. As the pressure ratio reaches the maximum for the device, activate the target mass flow setting in the pressure boundary properties. This setting allows you to fine-control the mass flow while STAR-CCM+ still applies an outlet pressure through successive corrections. (Alternative boundary combinations are also possible - you can choose either an inlet mass flow boundary or an outlet mass flow boundary in steady state. An outlet mass flow boundary is preferred when moving to an unsteady analysis.)
6. The SST K-Omega Turbulence model seems to be marginally more accurate than Spalart Allmaras turbulence model as conditions approach choked flow. Otherwise both perform with a similar level of accuracy.

A performance curve of 10 points can be generated in approximately a day when running a 4–6 million-cell mesh on 48 processors.

Note: Some over-prediction of mass flow at the choked condition is observed with the recommended settings. The over-prediction could be due to the inflow profile, or geometric simplifications like neglecting the under-wheel leakage.

Eliminating Reflective Boundary Effects

1. Extract reports of the mass-flow averaged Mach number, pressure, and temperature from the steady-state run at inlet and outlet boundaries.
2. Change the steady state case inlet boundary type to *Freestream* and set the Mach number, pressure, and temperature from the steady state reports.

3. Run the simulation to convergence with these new boundary settings.
4. Ensure that the pressure ratio and mass flow remain unchanged. Alternatively, wait until the mass flow balances across the inlet and outlet.

Recommendations for the Unsteady Simulation

1. Start the unsteady simulation from a converged steady solution, as described in the previous section.
2. Convert to transient.
3. Change the turbulence model to Detached Eddy Simulation (DES) using the SST k-omega model.
4. Apply the recommended Segregated Flow settings.
5. Use the Segregated Flow Model setting Hybrid-Bounded Central Differencing (BCD) or full BCD, if cell quality allows with blend=0.3. Use second-order time.
6. Freeze the Wall Distance Solver.
7. Run the case.
8. Run for between 100 and 200 full rotations to get a good statistical sample for digital signal processing.

Using these settings approximately 15 full rotations a day are possible when running a 4–6 million-cell mesh on 48 processors.

Choosing Transient or Harmonic Balance

Radial compressors are usually close to a downstream scroll volute, diffuser vanes, or other geometries which are not rotating. The correct way to simulate this system is to account for the periodic interaction of the rotating and stationary parts, namely downstream wake, and upstream pressure gusts of turning vanes, blades, and supports.

There are two ways to do model radial compressors in STAR-CCM+: using transient Rigid Body Motion (sliding mesh) or the Harmonic Balance method. Both are more time-consuming, but more accurate, than steady state approximations that use frozen-rotor or mixing-plane stator-rotor interfaces. The harmonic balance method runs faster than transient. However, the harmonic balance method only resolves tones that are based on the modes solved (blade passing frequency, harmonics, and Strouhal shedding). It does not resolve broadband noise which is associated with resolved turbulence.

Recommended Time-Step

Select the time-step size Δt, based on criteria:

Criterion	Comments / Description	Formula	Typical Values
Rotation speed ω	Aim for 1° rotation per time-step, or less if the number of blades exceeds 24. Then select Δt to give at least 15 time-steps per blade passage.	Minimum of $\Delta t = \frac{1}{6\omega}$ (1° rotation per time-step) and $\Delta t = \frac{4}{N_b \omega}$	Δt = 8.3E-7 s. Based on ω = 200,000 RPM.

Criterion	Comments / Description	Formula	Typical Values
		(15 time-steps per blade passage) where: N_b = number of blades.	

Recommended Solver Settings

The recommended solver settings are:

	Velocity URF	Pressure URF	# Iterations Per Time-step
Default	0.8	0.2	5
Regular	0.8	0.7	3–4
Aggressive	1.0	0.9	3

Axial Fans

Axial fans, commonly used for cooling and ventilation, are placed in locations where they are highly audible.

This section presents recommended practices to model acoustic sources in axial fans:

- Recommended Practices
- Recommended Time-Step
- Recommended Solver Settings

Recommended Practices

Qualify Using Steady State

Before running a transient analysis, it is vital to first qualify the model by ensuring that the fan performance curve is predicted well using steady state simulations. Use the Realizable K-Epsilon or the K-Omega SST turbulence models for consistency with the DES model in the unsteady analysis.

To avoid pressure reflections, set the inlet or outlet boundaries to be freestream depending on which is the important side. If it is necessary to control the mass flow, then set the mass flow at a boundary farther away from the fan. Aim to place boundaries at least ten fan diameters away from the fan.

Use Compressible Flow

Rotors passing relative to stators create flow and pressure pulsations which are strictly a compressible phenomenon. The pulsations can result in accumulated mass (monopoles) within the system, which an incompressible calculation does not permit. Even at low Mach-numbers, compressibility is preferred.

Recommendations for Obtaining Steady Performance Curves

1. No special initialization procedures are required. To accelerate convergence, set the initial axial velocity to give the expected mass flow through the fan.
2. No preferences exist between mixing plane and frozen rotor.
3. The segregated solver is preferred to coupled due to the existence of a follow-on step to transient where it is definitely faster. Prediction accuracy between the two is marginal. Use the same solver for both steady state and transient calculations to minimize flow-wide disturbances.
4. Low Reynolds number resolution on the blade is preferred for fan performance prediction.
5. Start by running with the known boundary conditions (pressure-pressure or mass flow-pressure) obtained from the experimental conditions.
6. Sample the Mach number at the boundary and reset the steady boundary conditions to freestream with the sampled pressure, Mach number, and temperature balance.
7. Run with freestream boundaries, and the system reregulates itself to find the correct mass-flow / pressure balance.

Using the recommended settings, a performance curve of 10 points can be generated in approximately a day when running a 4–6 million-cell mesh on 48 processors.

Recommendations for the Unsteady Simulation

1. Start the acoustics simulation from the steady solution with non-reflective boundaries as described.
2. Change the turbulence model to Detached Eddy Simulation (DES) using the SST k-omega model.
3. Use the recommended Segregated Flow settings.
4. Use the Segregated Flow Model setting Hybrid-Bounded Central Differencing (BCD) or full BCD, if cell quality allows with blend = 0.3. This higher blending factor can be necessary due to non-orthogonality across the sliding mesh. Use second-order time.
5. Freeze the wall-distance calculation. The wall distances are needed mainly near to the surfaces to calculate wall damping characteristics. Since the near-wall distances are unlikely to change, it is inefficient to calculate this distance at every time-step.
6. Convert to transient and run.
7. Run for between 100–200 full rotations to get a good statistical sample for digital signal processing.

Using the settings approximately 15 full rotations a day are possible when running a 4–6 million-cell mesh on 48 processors.

Choosing Transient or Harmonic Balance

Axial fans, that are used for under-hood cooling or building HVAC, are placed close to casing, mounting struts or other geometries which are not rotating. The correct way to simulate this system is to account for the periodic interaction of the rotating and stationary parts, namely downstream wake, and upstream pressure gusts of turning vanes, blades, and supports.

There are two ways to model axial fans in STAR-CCM+: using transient Rigid Body Motion (sliding mesh) or the Harmonic Balance method. Both are more time-consuming, but more accurate, than steady state approximations that use frozen-rotor or mixing-plane stator-rotor interfaces. The Harmonic Balance method runs faster than transient. However, the Harmonic Balance method resolves tones that are based on the modes solved (blade passing frequency, harmonics, and Strouhal shedding). It does not resolve broadband noise which is associated with resolved turbulence.

Recommended Time-Step

Select the time-step size Δt, based on criteria:

Criterion	Comments / Description	Formula	Typical Values
Rotation speed ω	Aim for 1° rotation per time-step, or less if the number of blades exceeds 24. Then select Δt to give at least 15 time-steps per blade passage.	Minimum of $\Delta t = \frac{1}{6\omega}$ (1° rotation per time-step) and $\Delta t = \frac{4}{N_b \omega}$ (15 time-steps per blade passage) where: N_b = number of blades.	Δt = 5.0E-5 s. Based on ω = 3,333 RPM.

Recommended Solver Settings

The recommended solver settings are:

	Velocity URF	Pressure URF	# Iterations Per Time-step
Default	0.8	0.2	5
Regular	0.7	0.5	5–10
Aggressive	1.0	0.8	5

Choosing DES, LES, or URANS for Aeroacoustics

Where boundary layer structures do not directly affect the aeroacoustics, DES and LES using Bounded Central Differencing perform similarly in terms of pressure spectra.

LES is quicker as it solves fewer equations. When the hybrid BCD discretization scheme is used in DES, some key regions, such as an A-pillar or separated shear layers, suffer from numerical damping of the pressure fluctuations. The guideline (at this time) is to resolve the boundary-layer down to y+ equal to 1.

In general, Unsteady Reynolds Averaged Navier Stokes (URANS) is not recommended, unless the flow features are guaranteed to be tonal (such as obtained from blade passing or coherent shedding). Never use URANS when there is broadband content arising from turbulence. DES uses RANS only in the boundary layer. RANS-type meshes are sufficient in the bulk flow (away from boundary layers). You can investigate the mesh suitability by interrogating the spatial distribution of the turbulence integral scales arising from a steady RANS calculation. Use a mesh size less than 0.5x(local integral turbulence length scale).

Mesh Issues

Highly Skewed Cells at Sliding or Rotating Interfaces or Small Gaps

In circumstances where the cell orthogonality or connectivity is compromised, the solver can experience instability with Bounded Central Differencing (for LES) or Hybrid-Bounded Central Differencing (for DES). Under these circumstances, second-order upwind spatial discretization is recommended; it is more robust than Bounded Central Differencing. However, expect the flow structures to suffer smearing from the upwind contribution of the second-order scheme. The main effect is the reduction of the spectral magnitudes at higher frequencies, and, possibly, the redistribution of energy to lower frequencies.

Refinement Interfaces on a Trimmed Mesh

There is some evidence that (acoustic) pressure reflections or spurious (acoustic) convective disturbances can occur at mesh coarsening interfaces for trimmed meshes. As a result, the local flow fluctuating structures swamp the acoustic pressure effects. The likely cause is local orthogonality errors that reduce the spatial accuracy toward first-order accuracy. Using 20 cells per acoustic wavelength is recommended. But, where the local discretization due to refinement is reduced to first order, use 40 cells per acoustic wavelength.

Another solution to the issue is to use an equivalent polyhedral mesh with gradual coarsening/refinement. The polyhedral mesh eliminates reflections, but a speed penalty factor of around 1.5 exists for an equivalent mesh size—directly proportional to the increased number of faces per cell.

Freestream Boundary Condition Issues

The freestream boundary, usable with an ideal-gas formulation only, is non-reflective to one-dimensional disturbances incident (that is, normal) to the boundary.

It is based on the one-dimensional method of characteristics, calculating the boundary face fluxes for all the (inviscid) transported properties from the classical “compression”, “expansion” and “convective” waves. These waves have eigenvalues $U + a$, $U - a$ and U. However, the following limitations exist when using a freestream boundary:

- This boundary treatment is found to be ill-posed when flow structures cross the boundary. When the flow structure crosses a boundary, spurious compressible pressure waves can emanate from the interaction and propagate upstream to influence the source field.
- Where the boundary is not normal to the incident compressible waves, errors can occur in the non-normal directions.

Acoustics in Liquid Media

In STAR-CCM+, hydroacoustic analysis is an extension of aeroacoustic analysis.

The ability to analyze sounds in liquids is possible under the following conditions:

- Use a double precision version of STAR-CCM+ to improve the numerics in calculating the speed of sound in liquids.

- Select a compressible EOS (Equation of State) for the fluid medium.
- Select an appropriate boundary condition. There are no non-reflective boundary treatments for liquid compressible flows, so use treatments to damp out the pressure waves from the source region before they reach the inflow and outflow boundaries.

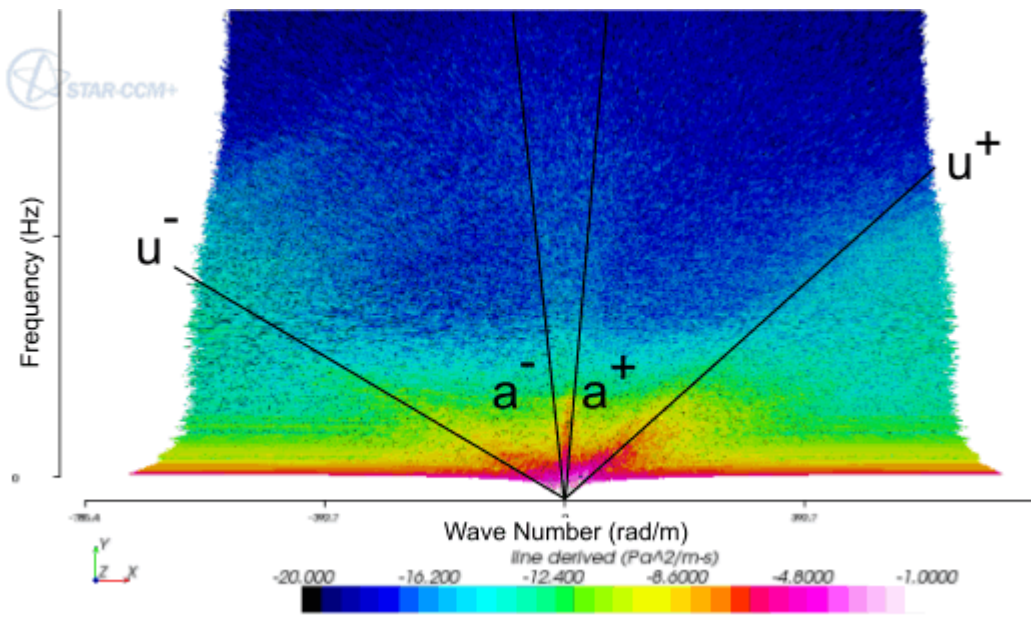
For more information, see Modeling Hydroacoustics.

Post-Processing the Acoustic Pressures and Pressure Perturbations

Although there are several ways to visualize the acoustical pressures and pressure perturbations, two useful methods are:

Using Wavenumber Decomposition

Using a line spatial Fourier transform, you can visualize the separate acoustical and convective energy content. The screenshot below shows the upstream (u^+) and downstream (u^-) convecting waves and the upstream (a^+) and downstream (a^-) acoustically propagating waves for a generic side-view mirror case:



For more information, see What Are Line Spatial Fourier Transforms?.

Using More Field Functions

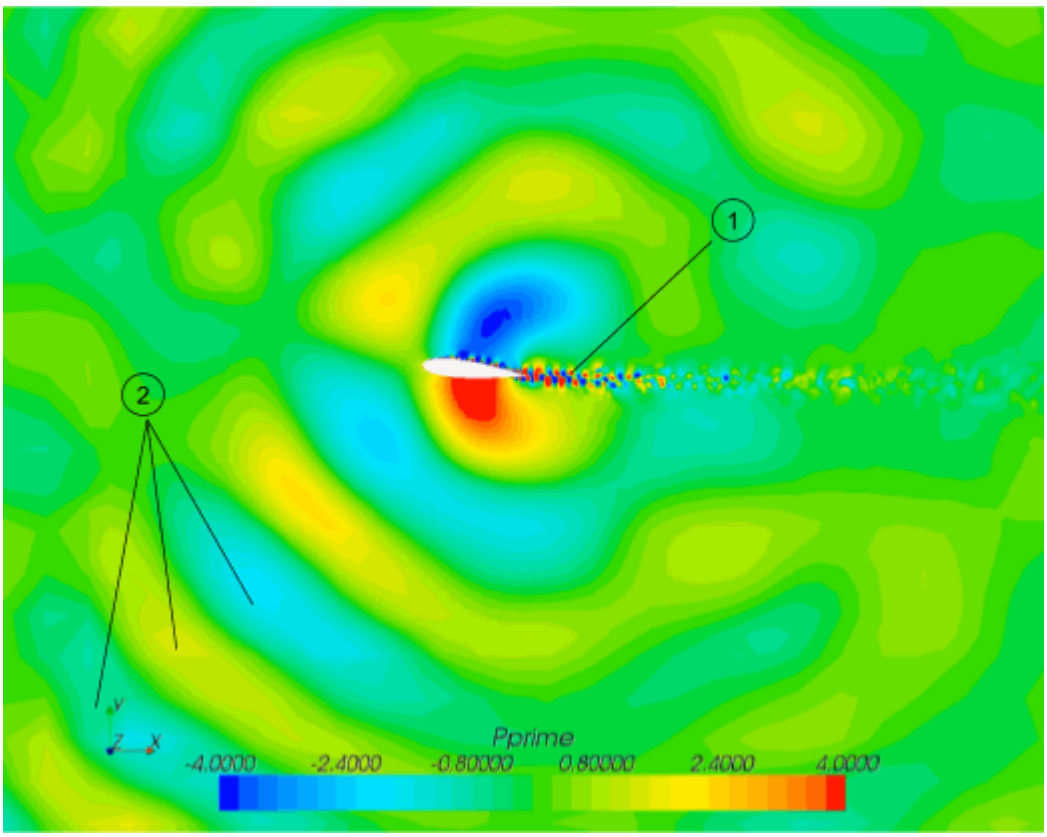
A useful method of visualizing the fluctuating flow field is to subtract the average flow from the instantaneous flow, that is:

$$p' = p - \overline{p} \tag{1196}$$

In aeroacoustics, use the technique below to display the fluctuating pressure p':

1. Run the transient calculation so that it is statistically steady. That is, the initial state has been removed from the domain and the fluctuations in the flow are unchanging.
2. Create the field mean value for pressure. To create this value, right-click **Monitors** and choose **New Monitor** > **Field Mean**. For the *Field Function*, select `Pressure`.
3. In order to obtain a statistical mean for each point in the field, run the simulation for several further cycles.
4. Create a field function, **Pprime**, defined as `${Pressure} - ${PressureMeanMonitor}`
5. Create a plot of **Pprime** to show the hydrodynamic and acoustical pressures.

An example plot is shown below:



Index	Description
1	Turbulent fluctuation in the near field
2	Acoustical fluctuation in the far field

Exporting Data to LMS Virtual.Lab Acoustics

STAR-CCM+ allows the export of mesh and solution data for import into LMS Virtual.Lab Acoustics. LMS Virtual.Lab Acoustics is an integrated CAE suite designed by Siemens PLM Software to perform 3D vibro-acoustics simulations for sound engineering.

Transferring data from STAR-CCM+ to LMS Virtual.Lab Acoustics requires the following steps:

1. Export of mesh and simulation results from STAR-CCM+ in `*.ccm` file format

   Note: For parallel runs, make sure that you export the mesh and simulation results using the same mesh partitioning.

2. Conversion of the exported `*.ccm` files to `*.cgns` format using the standalone program `ccmtocgns.exe`, which is part of the LMS Virtual.Lab Acoustics installation
3. Import of the `*.cgns` file into LMS Virtual.Lab Acoustics

This section contains guidelines on:

- Exporting Meshes from STAR-CCM+
- Exporting Fluctuating Solution Data from STAR-CCM+
- Exporting Steady-State Solution Data from STAR-CCM+
- Converting Exported Data
- Importing Converted Data into LMS Virtual.Lab Acoustics

Export of Meshes from STAR-CCM+

The transfer of solution data from STAR-CCM+ to LMS Virtual.Lab Acoustics requires the export of the corresponding mesh information to a `.*ccm` file.

To export mesh data from STAR-CCM+:

1. Select **File** > **Export...**.
2. In the *Export File* dialog, set the following properties:

Node	Value
File Name	Specify the name of the file to use when exporting the mesh.
Files of type	**CD-adapco CCM files (*.ccm)**
Export Data	Depending on where the solution data of interest are stored, select the relevant boundaries or regions to export the mesh from.
Export Type	**Mesh Only**

3. Click **Save**. `*.ccm`

   STAR-CCM+ exports a file that contains the mesh of the specified regions.

Export of Fluctuating Solution Data from STAR-CCM+

Exporting fluctuating solution data enables you to take pressure and velocity data from an unsteady STAR-CCM+ simulation and use them to define dipoles, pressure loads, or quadrupoles in an LMS Virtual.Lab Acoustics model.

To export transient simulation results from a STAR-CCM+ simulation for import into LMS Virtual.Lab Acoustics:

1. Before running the simulation, select **File** > **Auto Export...**.

2. In the *Auto Export* dialog, set the following properties:

Node	Value
File Format	**CD-adapco CCM files (*.ccm)**
Base Name	Specify the prefix of the file name to use when exporting the data.
Export Directory	Specify the directory where you want to save the exported data files. The default path is the same as the path of your simulation file.
Append to File	When activated, adds the exported data to a collective `*.ccm` file. This file would be the one you specify in the **Base Name**, **Export Directory**, and **File Format** properties. When deactivated, exports one `*.ccm` file per specified time-step. LMS Virtual.Lab Acoustics can import both a collective data file or separate data files. Usually, separate files are easier to handle for data transfer and management.
Export Solution Data Only	**Activated**
Update	
└ **Enabled**	**Activated**
└ **Trigger**	**Time Step**
└ **Time-Step Frequency**	
└ **Frequency**	Specify the number of time-steps between exports.

3. Depending on the analysis that you want to perform in LMS Virtual.Lab Acoustics, do one of the following:
   - If you want to define dipoles or pressure loads on structures in an LMS Virtual.Lab Acoustics model, set the following properties:

Node	Value
Scalar Functions	**Pressure**
Boundaries	Select the boundaries to export the pressure data from.

   - If you want to define quadrupoles in an LMS Virtual.Lab Acoustics model, set the following properties:

Node	Value
Vector Functions	**Velocity**
Regions	Select the regions to export the velocity data from.

4. Click **Close**.
5. Run the simulation.

STAR-CCM+ automatically exports the solution data to `*.ccm` files at the specified intervals.

Export of Steady-State Solution Data from STAR-CCM+

Exporting steady-state solution data allows you to take absolute pressure and temperature data from a flow and thermal analysis in STAR-CCM+ and use them to calculate the acoustic properties in an LMS Virtual.Lab Acoustics model. This approach is useful for applications that involve hot gases, where a non-uniform thermal/pressure distribution affects the acoustic wave propagation.

To export steady-state solution data from STAR-CCM+ for import into LMS Virtual.Lab Acoustics

1. After running the simulation, select **File** > **Export...**.
2. In the *Export File* dialog, set the following properties:

Node	Value
Save in	Specify the directory where you want to save the data. The default path is the same as the path of your simulation file.
File Name	Specify the name to use when exporting the data.
Files of Type	**CD-adapco CCM files (*.ccm)**
Export Data	Select the required field functions, such as *Absolute Pressure* and *Temperature*.
Export Type	**Solution Data Only**

3. Click **Save**.

STAR-CCM+ exports the solution data to a `*.ccm` file as specified.

Conversion of Exported Data

To convert the exported `*.ccm` files to `*.cgns` format:

1. Navigate to the `ccmtocgns.exe` executable that is contained in the LMS Virtual.Lab Acoustics installation directory.

   Example: `C:\Program Files\LMS\Virtual.Lab.RXX-SLX\VirtualLab\Sysnoise\5.6\bin\Windows_x86_64\ccmtocgns\` or `C:\Program Files\LMS\LMS Virtual.Lab XXX\Sysnoise\5.6\bin\Windows_x86_64\ccmtocgns\`.

   The exact folder depends on the software release and your installation settings.
2. Ensure that the following files are contained in the same folder (the last two may be located one level higher in the directory structure):
   - `ccmtocgns.exe`
   - `libifcoremd.dll`
   - `libmmd.dll`
3. Open a Command Prompt:
   a. Click **Start**.
   b. In the search field at the bottom of the **Start** menu, type `cmd`.
   c. From the search results, select `cmd.exe`.
4. In the Command Prompt, navigate to the directory where the `*.ccm` data are stored.

   Example: `cd C:\Software\LMS\ccmtocgns\TestData`

5. Launch the converter executable `ccmtocgns.exe` in the cmd window:
   a. Using the Windows Explorer, open the folder that contains the converter executable.
   b. Drag and drop `ccmtocgns.exe` into the cmd window.
6. The converter requests you to specify the following information:

Required Input	Entry
Enter Geometry File Name:	Enter the name of the file that contains the exported mesh, such as `mesh.ccm`.
Enter <First> Result File Name:	Enter the name of the solution data file that STAR-CCM+ exported first, such as `sdg_transient@00001.ccm`.
Enter Data Type:	Enter the data type to convert: • `1`: Unsteady Pressure on Boundary Cell for Generating Dipoles or Pressure Loads • `2`: Unsteady Velocity in 3D Elements for Generating Quadrupoles • `3`: Steady Flow Data (Temperature, Density, Mean Velocity, Mean Absolute Pressure) • `4`: Unsteady Pressure on Boundary Cell for Generating Pressure Loads from APE
Number of Result Files:	Enter the total number of the exported solution data files.

The converter automatically creates the following `*.cgns` files:

- `mesh.cgns`: Mesh file that contains the coordinates of the CFD centroids
- `envelope.cgns`: Mesh file that contains the CFD envelope mesh (for visualization only)
- `00000X.cgns`: Data files that contain the CFD data to be imported in LMS Virtual.Lab Acoustics. Each file corresponds to a different time-step.

Import of Converted Data into LMS Virtual.Lab Acoustics

When importing the data into LMS Virtual.Lab Acoustics, select the first time-step file that you want to import. The mesh, and data from each successive time step, is then imported automatically.

Exporting Data to wave6

STAR-CCM+ allows the export of geometry, mesh, and solution data for import into wave6, a frequency domain tool for noise and vibration simulations. wave6 models wave propagation in structures, fluids, foams, and fibers across the audible frequency range.

This section contains guidelines on the export of:

- Surface Geometries
- Volume Meshes

- Fluctuating Surface Pressures
- Fluctuating Fan Sources
- Fluctuating Surface Pressure Gradients
- Fluctuating Surface Stresses
- Time-Averaged Volumetric Data

Export of Surface Geometries

Exporting surface geometry data enables you to benefit from the surface handling capabilities of STAR-CCM+, such as CAD import or surface preparation, before you create a vibro-acoustics model in wave6.

To transfer surface geometry data from STAR-CCM+ to wave6 use one of the following file formats:

- **Nastran Shells (*.nas)**
- **Stereolithography File (*.stl)**

The .nas file format is useful for exporting geometries in which the surfaces are defined as separate groups. Be aware that .nas is an ASCII file format and thus file size can be large for large models. The .stl file format is a binary format and therefore more compact. The .stl format is the recommended format for exporting surfaces when you do not require grouping of individual surfaces.

To export from STAR-CCM+, follow the instructions for Exporting a Part.

To import into wave6:

1. In the object tree, right-click the **Operations** > **Import** > **FE importers** node and select **Create FE Model Importer**.
2. Select the **New Fe importer** node and set the following properties:

Property	Setting
File name	Click on the folder icon to browse to the .nas or .stl input file.
File format	**.bdf (Nastran)** (equivalent to .nas) or **.stl (STereoLithography)**
Import mesh as	**Geometry**

3. Right-click the **New Fe importer** node and select **Import**.

Export of Volume Meshes

Exporting volume meshes enables you to benefit from the volume meshing capabilities of STAR-CCM+ before you define finite element solids and cavities in wave6.

wave6 supports arbitrary linear and quadratic finite elements. In STAR-CCM+ you can generate appropriate element topologies by using the following mesh operations:

- **Directed Mesh**
- **Automated Mesh** that uses the **Tetrahedral Mesher** and, optionally, the **Thin Mesher**.

The **Directed Mesh** operation generates linear hexahedral elements (Hex8) and the **Tetrahedral Mesher** generates linear tetrahedral elements (Tet4). An **Automated Mesh Operation** that combines the **Tetrahedral Mesher** and the **Thin Mesher** generates linear wedge elements (wedge6) in thin geometry parts, but can also create linear hexahedral (Hex8) or linear pyramids (Pyramid5) as transition elements.

To transfer volume meshes from STAR-CCM+ to wave6 use the following file format:

- **CD-adapco CCM files (*.ccm)**

To export from STAR-CCM+, follow the instructions for Exporting Mesh Data.

To import into wave6:

1. Create a new Fe Model Importer.
2. Select the **New Fe importer** node and set the following properties:

Property	Setting
File name	Click on the folder icon to browse to the .ccm input file.
File format	**.ccm (StarCCM+)**
Import mesh as	**FE model**

3. Right-click the **New Fe importer** node and select **Import**.

Export of Fluctuating Surface Pressures

Exporting fluctuating surface pressures enables you to take time domain surface pressures from an unsteady STAR-CCM+ simulation and use them to define surface loads on structures in a wave6 model. This approach is particularly useful to model wind noise effects, such as side mirrors generate in the automotive industry.

To transfer fluctuating surface pressures from STAR-CCM+ to wave6 use the following file format:

- **Surface fft files (*.trn)**

To export from STAR-CCM+:

1. Before running the simulation, set up an **Auto Export** following the instructions for Automatically Exporting Mesh and Solution Data at Regular Intervals
2. In the *Auto Export* dialog set the following nodes specifically for surface pressure export:

Node	Value
File Format	**Surface fft files (*.trn)**
Scalar Functions	Select the relevant pressure fields. For example: for side mirror wind noise problems, export **Pressure** and **Acoustic Pressure** (available through the **Acoustic Wave Model**).
Boundaries	Select the boundaries of interest.

3. Run the simulation.

To import into wave6, do the following for each field function:

1. In the object tree, right-click the **Operations** > **Import** > **CFD importers** node and select **Create Fluctuating Surface Pressure Importer**.
2. Select the **New fluctuating surface pressure importer** node:
   a. Set the following **General** properties:

Property	Setting
File name	Click on the folder icon to browse to the .trn input file.
File format	**.trn (STAR-CCM+ 3D)**
Field name	Enter the name of the field function exported from STAR-CCM+.
Boundary name(s)	Optionally, enter the names of specific STAR-CCM+ boundaries.
Fluctuating surface pressure load	Select the relevant **Fluctuating Surface Pressure** object defined within your wave6 model (located under **Loads** > **Surface Load** > **Fluctuating Surface Pressure**).

b. Set the **Mapping Options** and **Signal Processing Options** properties as required.

3. Right-click the **New fluctuating surface pressure importer** node and select **Import**.

Export of Fluctuating Fan Sources

Exporting fluctuating fan sources enables you to take time domain surface pressures from an unsteady STAR-CCM+ simulation and use them to define noise sources in an acoustic fluid in a wave6 model. This approach is particularly useful to model sound radiation, propagation, and scattering of rotating fans and propellers to account for installation effects.

To transfer fluctuating fan sources from STAR-CCM+ to wave6 use the following file format:

- **Surface fft files (*.trn)**

To export from STAR-CCM+:

1. Before running the simulation, set up an **Auto Export** following the instructions for Automatically Exporting Mesh and Solution Data at Regular Intervals.
2. In the *Auto Export* dialog set the following nodes specifically for fan source export:

Node	Value
File Format	**Surface fft files (*.trn)**
Scalar Functions	**Pressure**
Boundaries	Select the boundaries of interest.

3. Run the simulation.

To import into wave6:

1. In the object tree, right-click the **Operations** > **Import** > **CFD importers** node and select **Create Fluctuating Fan Source Importer**.
2. Select the **New fluctuating fan source importer**:
   a. Set the following **General** properties:

Property	Setting
File name	Click on the folder icon to browse to the .trn input file.
File format	**.trn (STAR-CCM+ 3D)**

Property	Setting
Field name	`Pressure`
Boundary name(s)	Optionally, enter the names of specific STAR-CCM+ boundaries.
Fluctuating surface pressure load	Select the relevant **Fluctuating Fan Source** object defined within your wave6 model (located under **Loads** > **Volume Load** > **Fluctuating Fan Source**).

b. Set the **Mapping Options** and **Signal Processing Options** properties as required.

3. Right-click the **New fluctuating fan source importer** node and select **Import**.

Export of Fluctuating Surface Pressure Gradients

Exporting fluctuating surface pressure gradients enables you to take time domain surface pressure gradients on rigid surfaces in an unsteady STAR-CCM+ model and use them to define aero-acoustic loads on the boundaries of acoustic spaces in a wave6 model. An example application is modeling aero-acoustic rush noise in a muffler at higher frequencies.

To transfer fluctuating surface pressure gradients from STAR-CCM+ to wave6 use the following file format:

- **Surface fft files (*.trn)**

To export from STAR-CCM+:

1. Before running the simulation, select the **Solvers** > **Segregated Flow** node and activate *Temporary Storage Retained* to provide the field function **Apparent Pressure Gradient**.
2. Create the following field function to calculate the surface normal pressure gradient:

```
dot($$ApparentPressureGradient,$$Area.unit())
```

3. Set up an **Auto Export** following the instructions for Automatically Exporting Mesh and Solution Data at Regular Intervals
4. In the *Auto Export* dialog set the following nodes specifically for the export of surface pressure gradients:

Node	Value
File Format	**Surface fft files (*.trn)**
Scalar Functions	Select the field function created in step 2.
Boundaries	Select the boundaries of interest.

5. Run the simulation.

To import into wave6:

1. In the object tree, right-click the **Operations** > **Import** > **CFD importers** node and select **Create Fluctuating Surface Pressure Gradient Importer**.
2. Select the **New fluctuating surface pressure gradient importer** node:

a. Set the following **General** properties:

Property	Setting
File name	Click on the folder icon to browse to the .trn input file.

Property	Setting
File format	**.trn (STAR-CCM+ 3D)**
Field name	Enter the name of the field function created to calculate the surface normal pressure gradient.
Boundary name(s)	Optionally, enter the names of specific STAR-CCM+ boundaries.
Fluctuating surface pressure gradient load	Select the relevant **Fluctuating Surface Pressure Gradient** object defined within your wave6 model (located under **Loads** > **Surface Load** > **Fluctuating Surface Pressure Gradient**).

b. Set the **Mapping Options** and **Signal Processing Options** properties as required.

3. Right-click the **New fluctuating surface pressure gradient importer** node and select **Import**.

Export of Fluctuating Surface Stresses

Exporting fluctuating surface stresses enables you to predict a time domain 2D Maxwell surface stress along a curve in SPEED/STAR-CCM+ and then bring it into wave6 to describe a frequency domain surface load in a 3D VA model. This approach is particularly useful for modeling noise and vibration from electric machines when you have excitation from rotating magnetic fields.

To transfer fluctuating surface stresses from STAR-CCM+ to wave6 use the following file format:

- **Surface fft files (*.trn)**

To export from STAR-CCM+:

1. Before running the simulation, set up an **Auto Export** following the instructions for Automatically Exporting Mesh and Solution Data at Regular Intervals
2. In the *Auto Export* dialog set the following nodes specifically for the export of surface stresses:

Node	Value
File Format	**Surface fft files (*.trn)**
Scalar Functions	**Electromagnetic Force Density**
Boundaries	Select the boundaries of interest.

3. Run the simulation.

To import into wave6:

1. In the object tree, right-click the **Operations** > **Import** > **CFD importers** node and select **Create Fluctuating Surface Stress Importer**.
2. Select the **New fluctuating surface stress importer** node:

a. Set the following **General** properties:

Property	Setting
File name	Click on the folder icon to browse to the .trn input file.
File format	**.trn (STAR-CCM+ 2D)**

Property	Setting
Field name	`Electromagnetic Force Density`
Boundary name(s)	Optionally, enter the names of specific STAR-CCM+ boundaries.
Number of layers	Set the number of layers for the signal processing extrusion of the 2D load to a 3D surface load.
Extrusion length	Set the extrusion length for the signal processing extrusion.
Fluctuating surface stress load	Select the relevant **Fluctuating Surface Pressure Gradient** object defined within your wave6 model (located under **Loads** > **Surface Load** > **Fluctuating Surface Pressure Gradient**).

b. Set the **Mapping Options** and **Signal Processing Options** properties as required.

3. Right-click the **New fluctuating surface stress importer** node and select **Import**.

Export of Time-Averaged Volumetric Data

Exporting time-averaged volumetric data enables you to take pressure and temperature data from a thermal analysis in STAR-CCM+ and use them to calculate the acoustic properties in a wave6 finite element acoustic model. This approach is particularly useful for applications that involve hot gases, where a non-uniform thermal/pressure distribution affects the acoustic wave propagation, such as in mufflers or combustion chambers.

To transfer time-averaged volumetric data from STAR-CCM+ to wave6 use the following file format:

- **CD-adapco CCM file (*.ccm)**

To export from STAR-CCM+:

1. Before running the simulation, create a **Field Mean Monitor** for each of the **Pressure** (absolute or relative) and **Temperature** field functions, see Sampling Data Values During and After the Solution.
2. Set the **Field Mean Monitor** properties accordingly to average the fields for the relevant regions over time, see Field Function Monitor Properties.

   Field functions related to the **Field Mean Monitors** are created automatically. Check the *Title* property of the monitors for the names of the field functions.
3. Run the simulation.
4. Export the field functions related to the **Field Mean Monitors** to a **CD-adapco CCM file (*.ccm)** following the instructions for Exporting Mesh and Solution Data.

To import into wave6:

1. In the object tree, right-click the **Operations** > **Import** > **CFD importers** node and select **Create Acoustic Property Map Importer**.
2. Select the **New acoustic property map importer** node and set the following properties:

Property	Setting
File name	Click the folder icon to browse to the .ccm input file.
File format	**.ccm (StarCCM+)**

Property	Setting
Temperature field name	Enter the name of the field function that is related to the temperature averaging **Field Mean Monitor**.
Pressure field name	Enter the name of the field function that is related to the pressure averaging **Field Mean Monitor**.
Pressure option	**Absolute** or **Relative**, depending on the exported pressure data.

3. Right-click the **New acoustic property map importer** node and select **Import**.

   wave6 calculates the relevant acoustic properties, such as effective speed of sound or density, based on the imported data.

Guidelines for Aerodynamics Calculations

The following sections provide guidelines when using a steady state RANS approach for:

Contents:

Incompressible External Aerodynamics: Steady State RANS Approach
Transonic/Supersonic External Aerodynamics: Steady State RANS Approach
Vehicle Simulations

Incompressible External Aerodynamics: Steady State RANS Approach

The following sections describe guidelines for estimation of external aerodynamic/hydrodynamic loads for incompressible flow using a steady-state RANS approach.

This approach is valid when the maximum Mach number in the domain is less than 0.2–0.3, and when unsteady wake effects are not important.

Note: If unsteady wake effects are important, use a DES, LES, or URANS approach.

This page contains the following sections:

- Recommended Domain Settings
- Recommended Physics Setup
- Recommended Mesh Settings

Recommended Domain Settings

- Use a hexahedral or "bullet-shaped" far-field boundary, with domain extents around 8–10 body lengths or wing spans, whichever is larger, from the body. The mesh size on this boundary can be large: typically of the order of the body length.
- Use a velocity inlet boundary condition type for the upstream and lateral boundaries
- Use a pressure outlet boundary condition for the downstream boundary.
- Using a half-body with a symmetry plane is appropriate for cases with no sideslip. Take care to match the symmetry plane mesh growth with the mesh growth of the volume mesh.
- Optional: Angles of attack/sideslip are typically defined by changing the far-field boundary condition flow direction, not the orientation of the body.

Recommended Physics Setup

To set up an incompressible external aerodynamics model using the steady state RANS approach, choose the following physics models in order:

Group Box	Model
Space	**Three Dimensional**
Time	**Steady**
Material	Select a material model that is appropriate for your simulation.
Flow	**Segregated Flow**
Equation of State	**Constant Density**
Viscous Regime	**Turbulent**
Turbulence	**Reynolds-Averaged Navier-Stokes** (Selected automatically)
	Realizable K-Epsilon or **Spalart-Allmaras**

To select a specific turbulence model, choose from the following:

- The **Realizable K-Epsilon** model is preferred, especially if there is a blunt boundary at the end. Either the **low y+** or **all y+** boundary treatments can be used with this model, depending on the mesh.

  This model works best when complex geometric protrusions exist, bluff body separation occurs, or both.
- The **Spalart-Allmaras** turbulence model can be used in cases of streamlined geometries without large base separation regions.

  This model works best for attached boundary layers or mildly separated flows (that is, flow past a wing at or below stall).

Recommended Mesh Settings

Both polyhedral and trimmed meshes have proven successful on this class of problem.

In either case, typical rules for good meshing practices apply and are important. The mesh should give proper resolution of geometry and flow field features, along with good cell quality.

For instance, refine the mesh:

- Around leading and trailing edges of lifting surfaces, control surfaces, and other protrusions.
- At tips of lifting surfaces and control surfaces in order to capture vortices that are shed in these locations.
- At body/lifting-surface junctions.
- To resolve any bluff-body wakes.
- On airfoil leading edges. Use aligned surface meshes in these regions when possible.
- If feasible, use at least two cells to resolve thin blunt leading edges, or other thin step-like protrusions.

Note: External aerodynamics simulations can be sensitive to the quality of the surface geometry representation. Care must be taken when using the surface wrapper such that junctions/intersections result in smooth, well-defined edges. Whenever possible, the preferred approach is to start with clean CAD geometry and bypass surface wrapping.

Polyhedral Meshes

- Polyhedra have the advantage of smooth growth away from the body.
- Due to the smooth growth behavior, polyhedral are typically preferred when aerodynamic coefficients are of key importance.
- Due to the pseudo-random orientation of the faces, polyhedra also generally give better results than trimmed meshes when large sweeps in flow direction (such as angle of attack or sideslip) are used.
- Typical polyhedral meshes contain 2–20 million cells, depending on the complexity of the geometry and flow field, and the near-wall treatment.
- Use Volume Growth Rate, with the expansion rate set between 1.05 and 1.15.

Caution: Be careful when using volumetric controls with polyhedral meshes: abrupt changes in cell size can occur if there is a significant difference in the sizes that are specified on the surface and in the volumetric control.

Trimmed Meshes

- If you are modeling a single flow direction, or only small variation in flow, a trimmed mesh with the grid lines aligned with the flow often provides the most efficient approach to getting good answers.
- Typical trimmed meshes contain 4–40 million cells, depending on the complexity of the geometry and flow field, and the near-wall treatment.
- In order to capture the near field properly, set the volume growth rate to **Slow** or **Very Slow**.

Prism Layers

- Proper resolution of the boundary layer is critical. Choose the thickness of the prism layer such that the entire boundary layer is contained within it.
- The use of wall functions rather than integrating to the wall can be appropriate depending on several factors, including:
  - Desired accuracy
  - Relative importance of skin friction drag
  - Importance of transition
  - Existence/importance of separation/reattachment on smooth boundaries
  - For integrating to the wall, 20–30 prism layers are typically used, with the near-wall y+ being of the order 1.
  - For wall functions, 5–8 prism layers are typically used, with the near-wall y+ being of the order 50–150.
  - When using the Spalart-Allmaras Turbulence model, integrate to the wall (do not use wall functions).

Transonic/Supersonic External Aerodynamics: Steady State RANS Approach

The following sections describe guidelines for estimating external aerodynamic loads for compressible flow, with freestream Mach numbers ranging from 0.5 to 2.5.

If unsteady phenomena such as vortex shedding are not important, and angles of attack are not extreme, steady state RANS solutions are acceptable for this type of analysis.

Recommended Domain Settings

- Use a spherical far-field boundary, with domain extents around 8–10 body lengths or wing spans, whichever is larger, from the body. The mesh size on this boundary can be large: typically of the order of the body length.
  - When replicating a wind-tunnel test, modeling the wind-tunnel walls is appropriate if blockage effects are important. In this case, slip walls are often defined at the tunnel walls, and no prism layers are constructed.
- Use a freestream boundary condition type at the far-field.
- Using a half-body with a symmetry plane is appropriate for cases with no sideslip. Take care to match the symmetry plane mesh growth with the volume mesh growth.
- Optional: Angles of attack/sideslip are typically defined by changing the far-field boundary condition flow direction, not the orientation of the body.

Recommended Physics Setup

To set up a transonic/supersonic external aerodynamics model using the steady state RANS approach, choose the following physics models in order:

Group Box	Model
Space	**Three Dimensional**
Time	**Steady**
Material	**Gas**
Flow	**Coupled Flow**
Equation of State	**Ideal Gas**
Optional Physics Model	**Coupled Energy** (Selected automatically)
Viscous Regime	**Turbulent**
Turbulence	**Reynolds-Averaged Navier-Stokes** (Selected automatically)
	SST K-Omega or **Spalart-Allmaras**

To select the specific turbulence model, choose from the following:

- The **SST K-Omega** model is preferred. This model is robust, is well-behaved near walls, and does well in recirculation regions.

  Either the low y+ or all y+ boundary treatments can be used with this model, depending on the mesh. This model works best when complex geometric protrusions exist, bluff body separation occurs, and at supersonic speeds.
- The **Spalart-Allmaras** turbulence model can be used in cases of streamlined geometries without large base separation regions. This model works best for attached boundary layers or mildly separated flows (that is, flow past a wing at or below stall), and at subsonic speeds.

Recommended Mesh Settings

Both polyhedral and trimmed meshes have proven successful on this class of problem. In either case, typical rules for good meshing practices apply and are important. The mesh should give proper resolution of geometry and flow field features, along with good cell quality.

For instance, refine the mesh:

- Around leading and trailing edges of lifting surfaces, control surfaces, and other protrusions.
- At tips of lifting surfaces and control surfaces in order to capture vortices that are shed in these locations.
- At body/lifting-surface junctions.
- Where shocks exist.
- To resolve any bluff-body wakes.
- On airfoil leading edges. Use aligned surface meshes in these regions when possible.
- If feasible, use at least two cells to resolve thin blunt leading edges, or other thin step-like protrusions.

Note: External aerodynamics simulations can be sensitive to the quality of the surface geometry representation. Care must be taken when using the surface wrapper such that junctions/intersections result in smooth, well-defined edges. Whenever possible, the preferred approach is to start with clean CAD geometry and bypass surface wrapping.

Polyhedral Meshes

- Polyhedra have the advantage of smooth growth away from the body.
- Due to the smooth growth behavior, polyhedral are typically preferred when aerodynamic coefficients are of key importance.
- Due to the pseudo-random orientation of the faces, polyhedra also generally give better results than trimmed meshes when large sweeps in flow direction (such as angle of attack or sideslip) are used.
- Typical polyhedral meshes contain 2–20 million cells, depending on the complexity of the geometry and flow field, and the near-wall treatment.
- Use Volume Growth Rate, with the expansion rate set between 1.05 and 1.15.

Note: Be careful when using volumetric controls with polyhedral meshes: abrupt changes in cell size can occur if there is a significant difference in the sizes that are specified on the surface and in the volumetric control.

Trimmed Meshes

- If you are modeling a single flow direction, or only small variation in flow, a trimmed mesh with the grid lines aligned with the flow often provides the most efficient approach to getting good answers.
- Typical trimmed meshes contain 4–40 million cells, depending on the complexity of the geometry and flow field, and the near-wall treatment.
- In order to capture the near field properly, set the volume growth rate to **Slow** or **Very Slow**.

Prism Layers

- Proper resolution of the boundary layer is critical. Choose the thickness of the prism layer such that the entire boundary layer is contained within it.
- The use of wall functions rather than integrating to the wall can be appropriate for the SST K-Omega turbulence model depending on several factors, including
    - Desired accuracy

- Relative importance of skin friction drag
- Importance of transition
- Existence/importance of separation/reattachment on smooth boundaries
- For integrating to the wall, 20–30 prism layers are typically used, with the near-wall y+ being of the order 1.
- For wall functions, 5–8 prism layers are typically used, with the near-wall y+ being of the order 50–150.

When using the Spalart-Allmaras Turbulence model, integrate to the wall (do not use wall functions).

Solution Procedure

To improve convergence, especially as the Mach number increases, order the cells in the general direction of the flow using directional reordering.

Initialization

- Grid sequencing initialization with default values typically works well for this type of problem, and is recommended up to around Mach 2.
- The flow field can be initialized to uniform conditions matching the far-field boundary—recommended for Mach numbers greater than 2. However, an arbitrary linear boundary layer profile in the initialization (using a field function) may be required.

Solution

- The Expert Driver with default values works well.
- Appropriate CFL values are dependent on mesh quality and Mach number.

  For a high-quality mesh, a general rule-of-thumb is $CFL \approx 20/Mach$ for this regime. For example $CFL \approx 40$ for Mach 0.5, and $CFL \approx 10$ for Mach 2.

  If the Expert Driver is used, CFL ramping is necessary only if poor-quality cells are present or if the flow field is especially complex.
- Residuals may drop by only 2–3 orders of magnitude for this type of problem, especially if grid sequencing initialization is used (because grid sequencing initialization provides a better initial estimate of the flow field).
- Monitor the forces and moments on the object to determine convergence.

Vehicle Simulations

This section presents the following recommended practices to model external automobile aerodynamics:

- Choosing Between an Unsteady or Steady State Model
- Recommended Turbulence Model
- Recommended Mesh Settings
- Recommended Number of Inner Iterations
- Recommended Time-Step

- Recommended Monitors and Reports

Choosing Between an Unsteady or Steady State Model

Steady State

Steady state simulations are excellent tools for optimization studies to reduce drag. The advantages of running a simulation with steady state include:

- Quick turnaround times.
- Accuracy in absolute values and trends.

Unsteady

Unsteady simulations must be used when looking at cases such as vehicle handling, overtaking, or fleet modeling. In these cases, understanding the unsteady flow characteristics can improve the aerodynamics. Optionally, unsteady simulations can also be used in cases where transient eddy effects are important, such as aero-acoustics or wheel well modeling.

Unsteady simulations take longer to run. However, the overall turnaround time of an unsteady simulation can be reduced as follows:

- The time step can be increased using transient SIMPLE. However, if the time step is raised too high, the solution becomes non-physical.
- Run transient simulations with higher under-relaxation factors than steady simulations. The higher the relaxation faction, the fewer the inner sweeps that are required.
- Initialize transient simulations using a steady flow field. Use the K-Omega SST model to initialize the flow field and reduce the jump between steady and DES simulations.
- You can use either coupled or segregated solvers to initialize the flow field. The coupled solver has the advantage of reduced run time for larger case studies.

Recommended Turbulence Model

- Either SST (Menter) K-Omega or Spalart-Allmaras Detached Eddy Simulation model is recommended for vehicle aerodynamics. Both seem to give similar solutions.
- For vehicle aerodynamics, DES offers improved information on large eddy structures in the wake. For wake flow, DES is more accurate than URANS. So, for all studies of vehicle passing or wake flow, DES is recommended.
- Aero-acoustics studies have shown some cases where Spalart-Allmaras offers slightly better trends. Typically, for passenger cars, the SST K-Omega turbulence model is used.
- Large time periods are more critical than local flow structure for thermal transient simulations. Therefore, URANS is appropriate for this class of problem.

Recommended Mesh Settings

- The grid density must be refined enough to capture local physics. However, over-refinement slows down performance. Use volumetric sources to refine the mesh only in the required areas.
- The grid size must be fine enough to capture the turbulent length scale.

- Resolve the wake of vehicles and tires when modeling passenger cars. Smaller length scales are less critical. The wake is generally refined for half of the car length to the rear.

  A more realistic estimation is to calculate the frontal area. Based upon the frontal area of the vehicle, extend the same distance in the wake.
- Examining the results of a steady state analysis can help determine grid resolution for DES simulation using the following field functions:

Name	Function Name	Definition	Purpose
Turbulent Length Scale	**TurbulentLengthScale**	`pow(0.09,-1)*sqrt($TurbulentKineticEnergy)/$SpecificDissipationRate`	Determines Turbulence Length Scale for SST (Menter) K-Omega model
Grid Size	**GridSize**	`pow($Volume, 1.0/3.0)`	Estimate Edge Length of cell (based upon cubes)
Length Ratio	**LengthRatio**	`$TurbulentLengthScale/(2*$GridSize)`	To resolve turbulence properly, length ratio > 1 would be needed

1. Run a preliminary steady simulation.
2. Use the `TurbulentLengthScale` and `GridSize` field functions to calculate the Length Ratio, as defined in the table.

   The turbulent length scale is based upon the turbulent dissipation. The grid size is estimated based upon the volume of each cell
3. Use a mesh that yields a length ratio greater than one in the preliminary steady simulation for the transient analysis.

Recommended Number of Inner Iterations

- Lowering inner iterations to speed up the solution is a better strategy than increasing the time step. Large time steps adversely affect results.
- In most cases, 8 inner iterations are enough to converge a time step. Lowering inner iterations between 3–5 helps in reducing turnaround time.
- The overall drag value is not sensitive to the number of inner iterations. The grid density and refinement locations are more critical in getting accurate drag values.

Recommended Time-Step

- Time-step size depends on the physics that is being resolved.
  - For aero-acoustics, the time step must be small enough to resolve the desired frequency.
  - For aerodynamics, larger time steps are used to capture the large eddies in the flow field.

- For general drag values, resolve the large wake structures behind the tires/vehicle. A good estimate for a time step is approximately 100 time steps for a particle to reach the span of the vehicle:

$$\Delta t = \frac{\text{Length of Vehicle}}{100 \times \text{Vehicle Speed}}$$

- The courant number must be below 1 for most of the flow field. Small local peaks from flow accelerating around a sharp corner are fine if they spike over 1.

Recommended Monitors and Reports

Use the following monitors to check for convergence in inner iterations:

- Force on vehicle: this monitor reaches a steady value at convergence.
- Pressure probes in tire and vehicle wake

  80% convergence to final value is fine for drag prediction
- Velocity magnitude

  The peak velocity must be below 150 m/s. If the peak velocity is higher, raise the time scale coefficient for upwind blending (*CdesTimeLim*) expert property of the SST (Menter) K-Omega model to improve stability.

Guidelines for Simulating Air Flow in Electronic Systems

This section describes guidelines for simulating heating and air flow in electronics systems.

These guidelines are an iterative process, using the following steps, repeated until a satisfactory result is obtained:

1. Preparing and refining the geometry:

   Start with a model in 3D-CAD or a similar CAD modeler, consisting of composite parts and regions. Simplify this and define the interfaces.
2. Preparing and refining the mesh:

   Use pre-defined parts-based mesh (PBM) operations.
3. Setting up the physics:

   Use ideal gas and Boussinesq formulations for the air, and formulations for isotropic and orthotropic solids commonly used in electronics.
4. Setting up the boundary conditions:

   Include heat sources, pressure settings for inlets and outlets, the effects of any fans, ambient temperature, and, where applicable, radiation effects including solar loading and transparency.
5. Setting up the solution:

   Use segregated flow and energy methods with under-relaxation, include gravity, and include radiation effects when necessary.
6. Monitoring the results:

   Include temperature reports, geometry and mesh scenes, and plane sections showing temperatures and velocity vectors.

Geometry Preparation

The two main geometrical divisions are solid parts and air volumes.

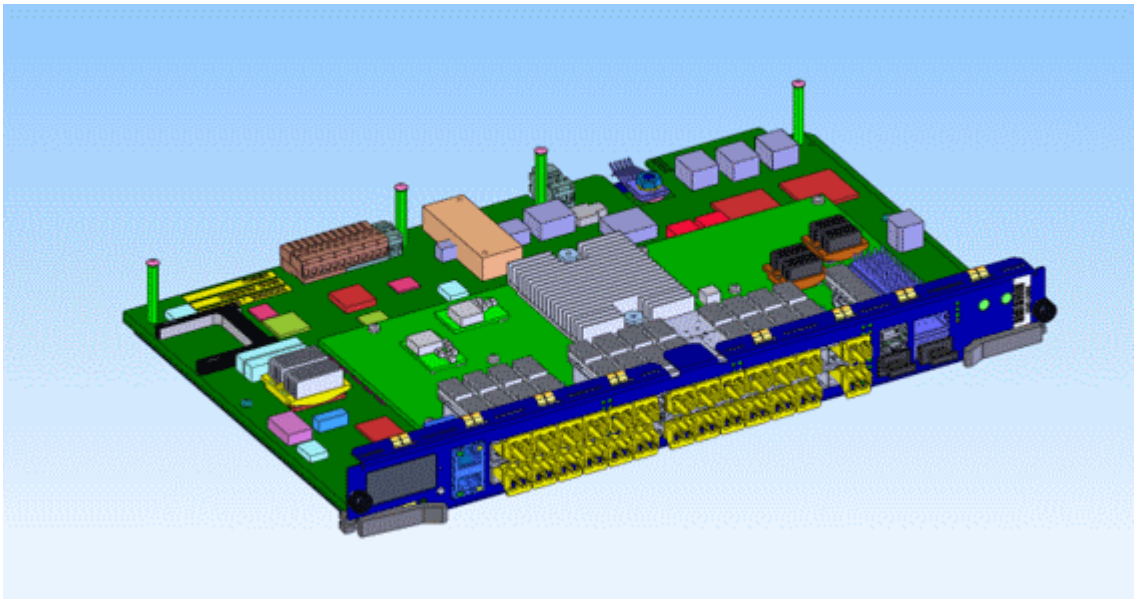
Solids

Simplify the heat flow path in the simulation by simplifying the solid geometry:

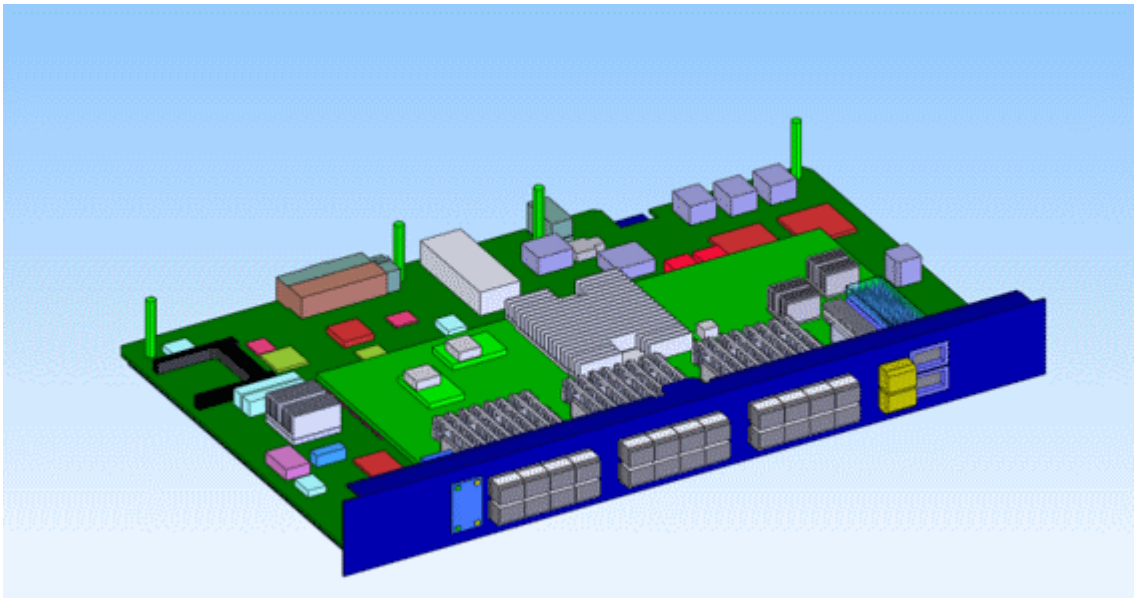
- Simplify or eliminate components away from the primary heat flow path.
- Eliminate any parts such as fasteners that are not critical to the heat flow path.
- Fill any holes created by eliminating irrelevant parts.
- Simplify components such as connectors, threaded stand-offs, and transformers.
- For sheet metal parts, get rid of the rounds resulting from the bends, and fill in or eliminate small features created by bend reliefs and the gaps resulting from rip corners.
- Eliminate interferences.
- Fill any irrelevant gaps. Many gaps in the CAD model come from standard manufacturing and assembly tolerances, but get closed on assembly.

Often, these simplifications can reduce the part count by 10% to 20%.

Example model before simplification:



Example model after simplification:



Air

The geometry of the air is determined by the cooling approach – forced or natural – and whether the internal air, external air, or both are involved.

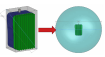
- Fill the internal air space with one or more parts so that the internal air flow can be computed.
- For the external air in a system cooled by natural convection, represent any physical flow barriers in the air domain, such as walls, ceiling, floor, or table. Create a shape to let the air move around the domain in

a sphere or hemisphere, where the outside faces represent the far-field conditions. Make the diameter of the sphere or hemisphere three to five times the length of the longest diagonal of the model.

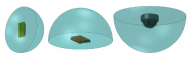
- For a system cooled by forced convection, where the air blown through the system provides the cooling, the air typically enters through a well-defined inlet or inlets in the enclosure and exhausts through well-defined outlets. Identify or create a short inlet and extended outlet.

3D-CAD has several features to aid in the creation of the air, including the **Extract Internal** and **Extract External Volume** features, plus the set of Boolean features including **Subtract**, **Unite**, and **Intersect**. Use the **Imprint** feature so that all the contacts and interfaces are automatically created as you move down the simulation tree.

Natural convection example: Consider an external radio cooled by natural convection, where the radio fills an envelope approximately 200 mm on the diagonal, and is suspended in air far from any walls or other flow obstructions. To simulate this case, make a sphere around the radio that is at least 600 mm in diameter.



If the device is not suspended in the air, but instead is mounted on a wall, table, or ceiling, make a hemisphere like one of the following.



Forced convection example: Consider a device such as the one shown below, cooled by a fan that draws air in through two inlets and exhausts it out one.

There are two situations to consider:

- Not considering the air outside the module. Then, a simple short inlet and extended outlet suffice. Notice that the outlet is long enough for the flow to have time to be nearly uniform.

- Considering the outside air. Then the air domain looks very similar to a natural convection case.

  Notice from this example that the air pattern coming out of the outlet with the spherical external air quite closely resembles the extended outlet. The primary difference between the two involves the heat loss (or lack of it) through the exterior walls of the module.

Mesh Preparation

Mesh each solid that is part of the thermal path.

- When feasible, create a conformal polyhedral mesh across the air and solids, using the same parts-based mesh operation or the same mesh continuum. A conformal mesh is good for both forced and

natural convection situations. It is particularly good for natural convection situations because of some particulars related to the radiation calculations.

- Alternatively, for forced convection models, you can also mesh only the air as polyhedral (and conformal if multiple air parts are used) and mesh the solids using trimmed cells. As a result the overall mesh is non-conformal. This approach can be easier and faster to mesh, but is not recommended for cases using radiation.

Both of the above options can be generated using a parts-based or regions-based meshing approach, according to your preference, but, if you want to use the embedded thin mesher to create a conformal thin mesh, you need to use a regions-based approach.

Typical mesh sizes are 500,000 to 5,000,000 cells.

The example below shows a model with more than 500 parts with an efficient, fully conformal mesh.

Physics Set-Up

Assign properties to the air and the solids.

Air

Air density must vary with temperature, so use the Ideal Gas model.

Density variation with pressure is negligible, so the Boussinesq model is sufficient for modeling buoyancy effects with natural convection.

Solids

Assign thermal properties to solids:

- Many solids, including metals, are isotropic. For these, the default solid material assignment in STAR-CCM+ is sufficient.
- Other solid parts, particularly printed circuit boards, are best described with orthotropic properties. Define the solid continuum as orthotropic, then specify the particular properties in the individual regions. The specific values depend on the number, thickness, and density of trace layers.

  For a typical circuit board, values are $k_{planar} \approx 10$ W/m-K and $k_{through\ plane} \approx 0.5$ W/m-K
- You can often assign properties to chips by describing the chip as a two-resistor thermal resistance network, using contact resistances. If you do not have any resistor network data, model the resistor as a ceramic such as aluminum oxide.

Boundary Conditions

Assign boundary conditions to the model for both heating and flow.

For both forced and natural convection:

- Assign temperatures on all air inlets and outlets.
- When applicable, assign a boundary condition that represents the heat exchange with the surrounding air.

- If there is no air surrounding the enclosure in the model (a common occurrence in forced convection cases), add convection on the boundary to model heat loss to the ambient:
  - For external natural convection, h ≈ 5 – 10 W/m^2-K
  - For external forced convection, h > 20 W/m^2-K
- Assign a heat power to every heat-dissipating component. Assign the thermal condition to the component region, calculating the power value as follows:
  1. Start with the actual power drawn by the component. This can be different from the power budgeted in the electrical design.
  2. Consider what the component does with the power:
     - If the component is a power transistor or IGBT, it delivers most of that electrical power elsewhere.
     - If the component is a radio chip or LED, it can be emitting the power as RF energy or visible light.

     The remaining power is typically dissipated as heat.

For forced convection, the three commonest cases are:

- Flow pushed into the system by fan or air supply at inlet. Specify:
  - At inlets:
    - Positive flow speed, positive pressure, positive mass flow rate, or fan pressure jump
    - Ambient temperature
  - At outlets:
    - Pressure outlet boundary condition (0 Pa)
    - Ambient temperature (for any reverse flow)
- Flow pulled through system by fan or air supply at outlet. Specify:
  - At inlets:
    - Stagnation inlet boundary condition (0 Pa)
    - Ambient temperature
  - At outlets:
    - Negative flow speed, negative pressure, negative mass flow rate, or fan pressure jump
    - Ambient temperature (for any reverse flow)
- Fan inside the system.

  Use the fan interface option.
  - Split out the boundary faces used to define the interface so that only the circular or annular faces are used as the boundaries for the specific interface.
  - Set flow direction from Boundary-0 to Boundary-1, swapping boundaries on the interface as needed.
  - Define the fan curve interface as a polynomial, OR

    Input the fan curve using **Tools** > **Tables** and then select the curve.

  Specify:
  - At inlets:
    - Stagnation inlet boundary condition (0 Pa)

        - Ambient temperature
    - At outlets:
        - Pressure outlet boundary condition (0 Pa)
        - Ambient temperature (for any reverse flow)

For natural convection:

- Set the following convection conditions on the exterior air boundary:
    - Stagnation inlet boundary condition (0 Pa)
    - Total temperature = ambient temperature
- Set the following radiation conditions on the exterior air boundary:
    - Boundary transparency = 1.0
    - Specify radiation temperature in the air continuum:
        - Indoors: Radiation temperature = wall temperature.
        - Outdoors: Turn on solar during the day. At night, radiation temperature = sky radiation temperature.

Solution Set-Up

Use the Segregated Energy and the Segregated Flow solvers. Set under-relaxation factors to:

- **Fluid Under-Relaxation Factor** = 0.99
- **Solid Under-Relaxation Factor** = 0.9999

Set **Stopping Criteria** to 300 to 500 iterations. This typically produces energy residuals under 1E-5 and moment residuals under 1E-8.

Models of 500,000 to 5,000,000 cells show good scalability up to 8 cores.

Results Monitoring

In general, monitor the same results you want to get from physical testing. Common results to monitor include:

- Residuals
- A 3-D image of the model showing temperature and velocity of flow
- Thermal resistance

Modeling Fluid-Structure Interaction

This section describes the various techniques and considerations for fluid-structure interaction (FSI) cases, and how to deal with FSI cases in STAR-CCM+.

FSI can be modeled entirely within STAR-CCM+, without resorting to a third-party code. You can model FSI as a fluid interacting with a rigid body using dynamic fluid-body interaction (DFBI), or as a fluid interacting with a deformable body using the finite element solid stress capabilities (see Modeling Fluid-Structure Interaction (FSI)).

You can also model FSI problems by coupling STAR-CCM+ to a third-party code, whereby STAR-CCM+ solves for the fluid domain and the third-party code solves for the structure. STAR-CCM+ can be coupled to NASTRAN, ANSYS, Abaqus and Radtherm using the File-Based Coupling technique, or to Abaqus using Co-Simulation.

Contents:

What Is Fluid-Structure Interaction?

See Also:

Modeling FSI Using Finite Volume Stress

What Is Fluid-Structure Interaction?

Fluid-structure interaction (FSI), broadly speaking, is the thermo-mechanical interaction of a fluid and a solid structure.

If one asked one hundred engineers what FSI means to him/her, one would likely get one hundred different answers. This section attempts to make some classifications of the interactions and the capabilities that are required to address FSI problems.

Classifications of FSI Coupling

This section describes the various types of fluid-structure interactions.

FSI can be grouped into two distinct categories:

- “One-way” interactions
- “Two-way” interactions

FSI problems can also be grouped by the kind of coupling used:

- Weak coupling
- Strong coupling
- Other coupling algorithms

Each type of interaction is discussed below.

One-Way Interactions

In “one-way” interactions, the fluid can impart some action on the structure but the response of the structure to the fluid loading does little to affect the fluid motion. For example, a fluid can heat or cool a stiff, supported structure, which produces thermal stress loads and deformations in the solid material. However, these thermal deformations do not necessarily lead to any significant change in the flow patterns of the fluid. Such a problem involves a two-way exchange of heat, which is referred to as Conjugate Heat Transfer. Nonetheless, the mechanical exchange is only one way, so it is sufficient to compute the temperature in the fluid and solid domains in separate simulations. Then you can compute the stresses in another simulation using the results of the temperature that is computed in the previous simulation.

The one-way interaction can also be the other way. For example, the motion of a piston and valves in an internal combustion engine certainly affects the fluid flow patterns in the combustion chamber. However, the motion of the fluid has little effect on the deformation of the piston and valves (other than, or course, the thermal effects).

Two-Way Interactions

In “two-way” interactions, the fluid motion and pressure affect the displacement and deformation in the structure. The response of the structure has a significant effect on the fluid flow. A prime example is the Dynamic Fluid Body Interaction capability of 6DoF to model the coupled interaction between a rigid body and the fluid.

Traditionally, FSI implies two-way coupling of a fluid and a deformable structure, such that the deformation and rigid body DOF are coupled with the fluid. Even under this definition, there are classifications of the degrees of coupling, ranging from “weak” or “loose” coupling to “strong” coupling.

In two-way interactions, it is clear that the solid structure responds to the pressure or viscous shear traction in the fluid. On the other hand, the fluid responds to the structure in various ways. The shape of the structure obviously affects the flow field, but it is the velocity of the structure (or change in displacement) that has the greatest effect. In fact, it can be shown that the acceleration of the structure causes proportional pressure responses much like how the acceleration of the fluid leads to a hydrostatic pressure gradient. For example, striking a hammer on a constrained pipe. In the fluid, the response would be a pressure wave that could be heard but not seen, and the small deformations in the pipe would be visually imperceptible. In this case, the response is determined by how fast the structure moved, or more accurately, how the structure accelerated. It is not determined by how much the structure moved.

Aero (Hydro) Elastic Equilibrium

In most cases, FSI implies “dynamic” simulations, or what would be referred to in CFD as “transient” simulations. However, in some situations, “static” solutions can be important. For example, the static deformation of a wing structure due to a “steady-state” airflow around the deformed wing. In such cases, a transient analysis in the fluid and structure can still be used to reach the static steady-state solution. However, the time-step can be thought of as a pseudo time-step, since the simulation does not need to be “time accurate” in its search for the steady solution. The fluid and solid time integration can be reduced to first order. Also, by definition, “static” implies that the velocity of the structure is zero. Therefore in the static solution you can ignore the grid flux terms in the fluid, since the grid flux terms are zero. On the structural side, you can increase the material damping or Rayleigh damping for the same reason. That is, when the velocity of the structure goes to zero, the damping forces are zero. More precisely, setting the damping to “critical” damping is the quickest way to drive the structure to the steady-state. Thus, ignoring the grid flux and adding structural damping promotes stability.

Weak Coupling

In some two-way FSI problems, you can regard the coupling to be “weak”, and so employ a “loose” coupling algorithm to find the solution.

Consider a structure that bends in steady-state flow. This bending can significantly alter the flow patterns around the structure, which, in turn, affect the bending. When searching for a steady-state condition, one is interested in the shape the structure takes under the steady condition. As the structure and fluid reach the steady-state, the structure material point velocities approach zero. In this situation, the coupling is “weak” since in the steady state the structure imparts no motion on the fluid. In fact, it is almost “one-way”, since the primary influence of the structure on the fluid is from its velocity, which in steady state is zero. In this situation, a “loose” coupling algorithm between the solid and fluid is sufficient to determine the end steady condition.

For example, the fluid can initially assume the structure is rigid and constrained, and a flow field can be computed independently. The pressure that is generated can be passed to a separate structural solver and the deformations in the structure can be computed from this initial guess of the fluid loading. Once the deformations in the structure are known, they can be provided to the CFD analysis to compute the steady flow with the new deformed geometry. This iteration can be repeated until a steady-condition is met. The important thing to notice is that this analysis has no consideration for the time scales. Since the point of interest is not how it reaches the steady-state solution, but simply what steady-state solution the fluid/structure reaches. The fluid solver only requires to know where the fluid/structure interface lies, not how fast it is moving or accelerating.

In a general sense, weak coupling implies that the response times of the structure to a disturbance in the fluid are slow, compared to the fluid. The opposite is also true, the response times in the fluid to a disturbance in the solid are slow, compared to the solid.

A distinction must be made between “loose coupling” and “loose coupling algorithms”. Loose coupling implies the physical strength of the coupling, whereas loose coupling algorithms imply the degree to which the fluid and structure codes are coupled. The loose coupling algorithm can mean simply passing data back and forth between different codes using files on the disk. It can also mean the fluid and structure solvers are not necessarily in the processor memory at the same time. Also, most certainly it means a superficial coupling strategy that does not go deep into the solution algorithms of either code. In particular, loose coupling algorithms are often explicit, in that the response of the solid depends on the conditions of the fluid at the previous time step. The opposite can also be said. This coupling strategy is often referred to as “staggered” coupling.

Strong Coupling

In “strong” coupling, the physical coupling is two way and the coupling between the codes is pronounced.

Often “strong” coupling is associated with “dynamic” problems where the hydrodynamic loads and the structural velocities change dramatically. In strong coupling algorithms, the fluid and structure solvers can be resident in processor memory at the same time. Data is passed at regular intervals (often referred to as coupling time or coupling step) from the memory that the structure solver uses to/from the memory that the fluid solver uses. Depending on the actual strength of the coupling it can be necessary to exchange data between the solvers more than once per time step for the simulation to remain stable. This type of coupling implies a deeper communication between the structure and fluid codes and is known as: “implicit”, “iterative staggered”, “iterative successive substitution”, or “multiple iterative coupling”.

Implicit coupling is necessary when a relatively light or compliant structure interacts with a relatively heavy, almost incompressible fluid. For example, a steel structure interacting with air is not coupled strongly in the

physical sense. However, a steel structure in water can be coupled strongly as long as the structure is relatively light compared to the fluid it displaces or moves. This phenomenon is true even although the structure is relatively stiff compared to the bulk modulus of the water. A ship only floats because it is lighter than the water it displaces, and thus the interaction is coupled strongly. Indeed the Dynamic Fluid Body Interaction (DFBI) capabilities in STAR-CCM+ couple the rigid body dynamics of a structure with the fluid flow in an implicit manner.

Strong coupling can occur in situations that are not immediately evident. For example, flow of water through a stiff, vibrating pipe. In this case, the mass of the pipe can be smaller than the mass of the water that is contained within the pipe. Unless the coupling time step is small, explicit coupling is not stable in this situation. Such cases can be dealt with in STAR-CCM+ by taking advantage of the implicit coupling between the CFD and FV Stress solvers. This coupling algorithm is strong and deep, and the exchange of data occurs at each iteration within a time step. This coupling also considers the load balancing and the partitioning, and minimizes the cross-talk among different processors between the fluid and the solid.

Other Categories of Coupling Algorithms

As mentioned in the previous section, the DFBI capability in STAR-CCM+ is implicitly coupled. You can take advantage of this stable implicit coupling even when you are interested in the deformations and resulting stresses in the ship. If actual deformations about the rigid body modes are relatively small, then the fluid motion is not sensitive to the deformations, only the rigid body motions. Thus, for example, the CFD and rigid body modes can be solved together in this implicit fashion. Then the transient loads can then be applied to the structural solver, which accounts for deformations and stresses. This process, in effect, is a one-way coupling, since the deformations computed in the structure code are not passed back to the CFD solver. STAR-CCM+ can be controlled using Java macros to map and export fluid dynamic loads for use in a structural solver such as Abaqus.

Dynamic Mesh Evolution

The motion of a structure in a fluid requires the CFD solver to account for changes in the shape and position of the solid structure. In STAR-CCM+, various strategies are used for this problem. The strategies are discussed in the following sections:

Mesh Rotation and Translation in FSI

The simplest type of mesh motion strategy is to displace the fluid mesh as if it were a rigid body. In this case, all the cells maintain their shape and the description of the mesh motion is from a displacement vector and Euler angles.

In DFBI, the position of the grid is determined by solving the equations of the assumed rigid structure, and the fluid transport equations automatically account for motion of the grid. This type of mesh motion is suitable when the rigid structure does not approach other solid structures. For example, this technique could be used for a ship that is in relatively deep water with no other structures or ships nearby. If other structures were nearby, the mesh is sometimes required to morph to account for the relative motion between the structure and other structures.

STAR-CCM+ also allows the fluid mesh to rotate with a rigid structure, and the rotating mesh to be embedded within another stationary mesh in a different region. The two mesh regions can then slide with respect to each other at an interface. The solver automatically calculates this interface projection and imprinting at each time step.

Morphing in FSI

Morphing, in this context, is the deformation of the fluid grid. Morphing occurs by moving the fluid vertices in such a manner as to conform to the solid structure and maintain a reasonable quality fluid grid.

This process is often referred to as a “topologically constant” operation since all the cells maintain the same neighbors but the shapes of the cells can change over time. The fluid transport equations are solved to account for this arbitrary motion of the mesh. This technique is sometimes referred to as the Arbitrary Lagrangian Eulerian (ALE) technique, but often ALE is associated with FEA, rather than FVA. STAR-CCM+ uses a “space conservation law” to conservatively and accurately express the transport motion. The key word in ALE is arbitrary, since the motion of the vertices is arbitrary except that it must conform to the moving boundaries and it must provide a reasonable quality mesh.

The morpher in STAR-CCM+ currently uses a multi-quadric morphing model that is based on radial basis functions to define the motion of interior vertices. This model originates from the motion of the vertices on the structural surface. When the structure moves in relation to other solid structures, the fluid mesh is distorted to conform to the structures. However, when two structures come into contact or move nearby, a morphing strategy can lead to poor cells.

It is not possible to morph between topologies that are dramatically different. In this case, you can:

- use the overset mesh capability and morph the overset mesh around the deforming structure, or
- use the advanced automatic meshing capabilities in STAR-CCM+ to remesh the model completely, using the most recent boundaries as a starting point.

Overset Mesh for FSI

When two structures approach each other, the fluid mesh between them must be squeezed in such a way as to retain the boundary shape that is formed by the structures. When two structures move apart, the fluid mesh must expand. In both cases, there is a threshold beyond which the morpher cannot avoid creating poor quality cells. For such situations, the overset mesh technology in STAR-CCM+ provides an effective solution.

Overset meshes that wrap around deformable structures can still be morphed using the structural deformation for the control points. If structures drift apart, however, most of the distance between the structures is treated using the background mesh.

Overset interpolation requires that the overset mesh always overlaps the background mesh by at least 4–5 cells rows. For this reason, it is not possible to close a gap completely when using the overset technique. However, you can still satisfy the overall interpolation requirement for a small gap size by adding more prism cell layers on the walls of both approaching structures.

Gap Closure

To mimic gap closure in STAR-CCM+, make the final gap sufficiently small that it gives a high resistance to fluid moving through it. If you intend to stop the flow (as in a stop valve), then drive the flow using pressure boundary conditions at both the outlet and inlet. In these circumstances, the maximum flow rate is proportional to the pressure difference across the gap, and inversely proportional to the viscous resistance of the flow within the gap. Ultimately the gap does leak fluid, but potentially at a small enough rate for the simulation to be an acceptable engineering approximation for the needs of FSI analysis.

Mapping Between Structure and Fluid Grids

Another challenge that must be overcome in FSI cases is the difference in the resolution between the fluid and structure grids. Often the grids are different due to the difference in physical processes in the fluid and the structure.

If the structural mesh and the fluid mesh are both constructed in STAR-CCM+, the two meshes can be made conformal at the fluid-structure interface. That is, the vertex positions on the fluid surface match the vertex positions on the surface of the structure. In this case, the mapping of fluid loads and structural displacements is more or less trivial.

When the opposing meshes are not conformal, STAR-CCM+ offers various strategies to map the data efficiently and accurately to and from meshes that are imported from other CAE products. This mapping is accurate, continuous, bounded, and as conservative as possible. When the opposing surface is an FEA mesh, STAR-CCM+ uses the innate shape functions of the finite element topology to map between the structure and fluid meshes.

Reference Configuration

Mapping requires that fluid and solid meshes must be similar at some reference configuration. The meshes must be sufficiently similar that the neighbors of faces on one side are matched to faces on the opposite side. The reference configuration is usually the geometry at the initial time. The fluid and solid mesh are assumed to move together so that their neighbors remain the same. For this reason, the weights of the mapping are defined entirely by the geometry of the mesh at the reference configuration—the mapping does not depend on the current deformed geometry.

In the data mappers, you can choose whether the reference configuration is set by the original mesh or by the current mesh, which could be deformed as a result of morphing. When the reference configuration is set to use the current mesh, STAR-CCM+ performs a neighbor search each time the mesh is moved or morphed. As this neighbor search is an expensive operation, you are recommended to set the reference configuration to the original mesh. STAR-CCM+ generally uses the original mesh configuration when it automatically sets the data mapping schemes appropriate for the physics in co-simulation. It also uses this configuration for mapped fluid/solid interfaces within a single simulation.

The assumption that neighbors at the reference configuration will always be neighbors is violated if the fluid mesh slides with respect to the solid mesh. In this case, the reference configuration must always be the current deformed mesh and neighbors must be re-established at each new time step.

The reference configuration need not always be at the initial time or at a re-meshing event. For example a structure can be pre-loaded to deform the shape. The fluid mesh can be initially constructed to conform around the deformed shaped of the structure (for example, the deformed tire loaded by the weight of the car and contact with the pavement). The reference configuration is now the deformed shape at this point in time.

HPC and Grid Partitioning on Multiple Processors

Modeling FSI also presents challenges regarding the partition of the fluid and structural grids.

In STAR-CCM+, when one employs the FV Stress solver for the structure, the grid is automatically partitioned for load balancing and to minimize processor-to-processor communications. In this case, the partition is not required to respect the boundaries between the structure and the fluid. In other words, any particular partition can have part of the fluid grid and part of the structural grid. It is this unique feature that makes STAR-CCM+ efficient for FSI applications on multiple processors.

Modeling Atmospheric Boundary Layers

This section describes a recommended technique for ABL (Atmospheric Boundary Layer) modeling in STAR-CCM+.

The ABL is an important part of simulating wind and is used in modeling weather, wind-borne pollution, and designing wind-farms.

Contents:

What is the ABL?
Setting ABL Parameters in STAR-CCM+
Recommended Values for ABL Modeling
Bibliography

What is the ABL?

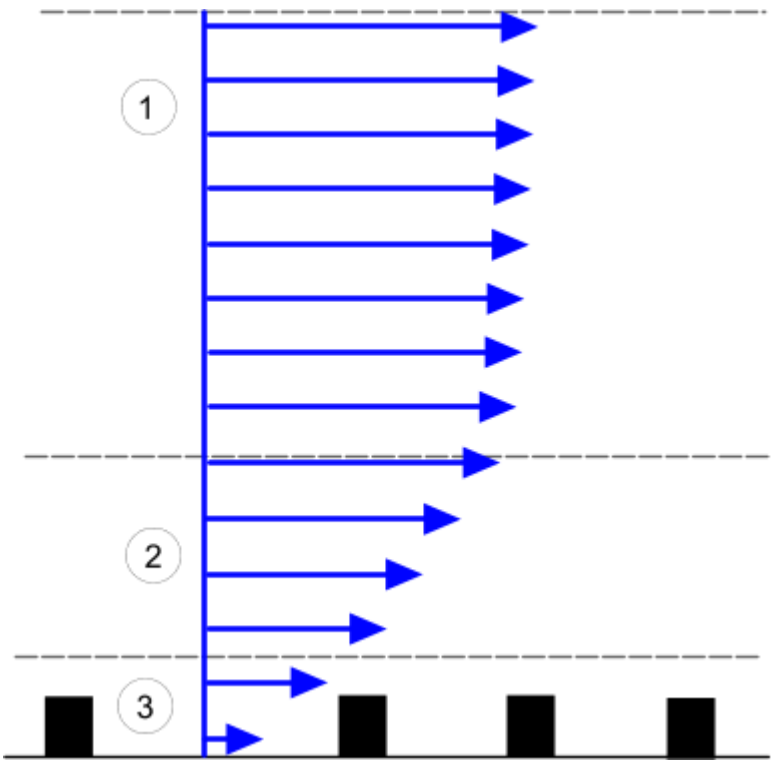
The ABL (Atmospheric Boundary Layer) is the bottom layer of the atmosphere which is in contact with the surface of the Earth.

Within this layer, the presence of the surface of the Earth directly affects flow: frictional effects and temperature create turbulence and vertical mixing. Above this layer, rotational effects dominate the flow rather than turbulence. The depth of the ABL can vary in both time and space.

Garratt [180] decomposes the boundary layer in the atmosphere of the Earth into two sublayers:

- An outer Ekman sublayer where there is little dependence on the nature of the surface.
- An inner sublayer where surface effects such as friction and heating are important.

There is an interfacial sublayer within this inner layer that defines the height and structure (together termed “roughness”) of obstacles to the flow on the ground.



Index	Description
1	Ekman sublayer
2 & 3	Inner sublayer
3	Interfacial sublayer

Formulation for ABL in STAR-CCM+

In the literature, the preferred method of modeling the roughness is to use the Richard and Hoxey Law [182] for the velocity distribution, u_{RH}^{+}:

$$u_{RH}^{+} = \frac{1}{\kappa}\ln\left(\frac{y_0 + y}{y_0}\right) \tag{1197}$$

where:

- κ is the von Karman constant.
- y_0 is the physical roughness length
- y is normal distance from the wall to the wall-cell centroid.

For ABL simulations, the flows are predominantly turbulent, and the velocity distribution maps to Eqn. (416) in STAR-CCM+. Following Martinez [181], let Eqn. (1197) equal Eqn. (416) and cancel the common terms to give:

$$\frac{y_0 + y}{y_0} = E' y^{+} \tag{1198}$$

Substitute E' using Eqn. (417) and Eqn. (432) for rough walls ($R^+ > 90$, $B = 0$):

$$\frac{y_0 + y}{y_0} = \frac{Ey^+}{CR^+} \tag{1199}$$

Substitute y^+ using Eqn. (410) and R^+ using Eqn. (431). Cancel common terms, giving:

$$\frac{y_0 + y}{y_0} = \frac{Ey}{Cr} \tag{1200}$$

Finally, assume that $y \gg y_0$, so that:

$$\frac{y}{y_0} = \frac{Ey}{Cr} \tag{1201}$$

Canceling common terms provides the relation to modify the STAR-CCM+ wall roughness equations to model an ABL:

$$r = \frac{Ey_0}{C} \tag{1202}$$

where:

- E and C are wall function coefficients.

To guarantee physical results, ($u^+ > 0$) make sure that $y \gg y_0$ is satisfied and deactivate the *Roughness Limiter* wall roughness parameter.

See Also:

Wall Treatment

Setting ABL Parameters in STAR-CCM+

To model an ABL, use a high Reynolds number turbulence model and a high y^+model.

For more information on the settings for the K-Epsilon and K-Omega models, see Recommended Values for ABL Modeling.

Activate wall roughness and specify the **Roughness Height** using values from the **Standard Wall Function** and the **Wall Roughness Parameters** nodes using Eqn. (1202). For example, in a case with y_0 equal to 0.01 m, E equal to 8.75, and C equal to 0.25, and $y \gg y_0$, the calculated roughness height r is 0.35 m. To set the values that are mentioned:

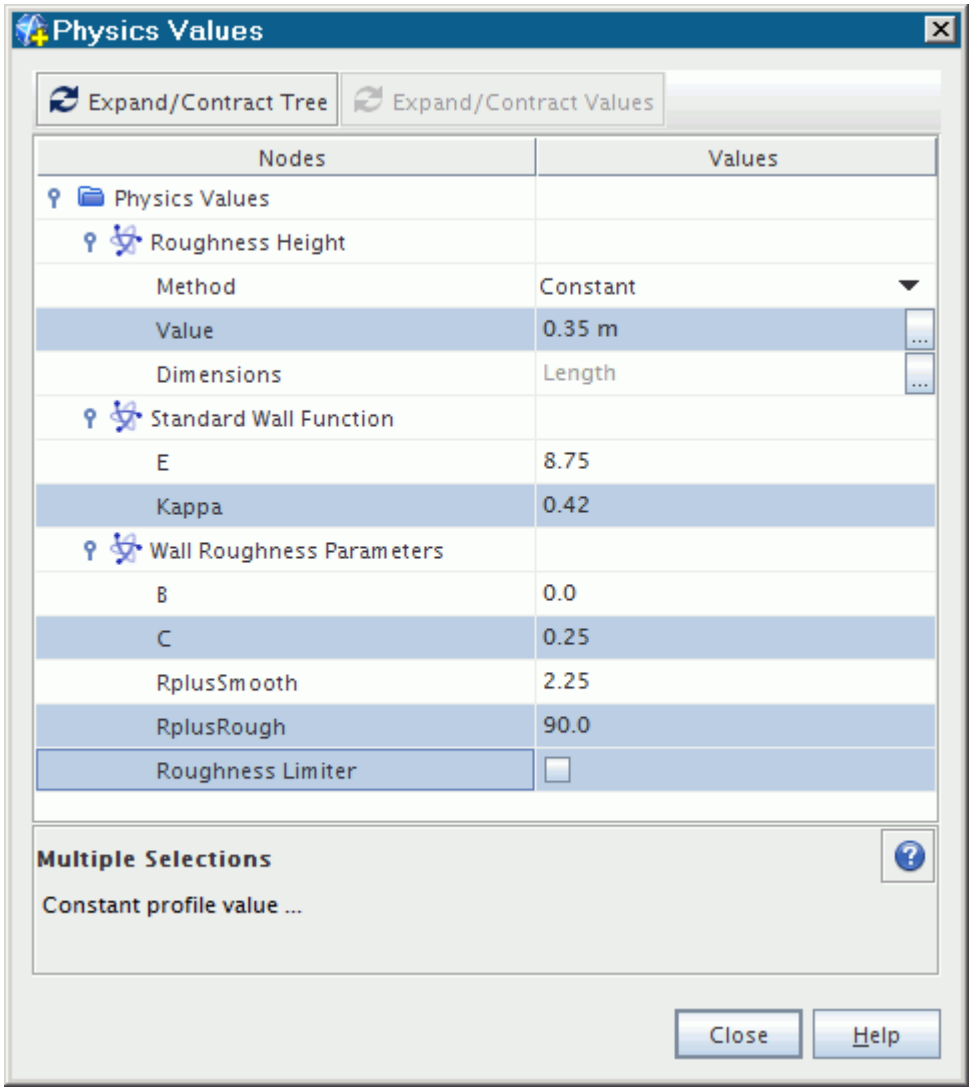
1. For the wall boundary, select the **Physics Conditions** > **Wall Surface Specification** node and set *Method* to **Rough**.
2. Right-click the **Physics Values** node and select **Edit...**

In the *Edit* panel, enter the values for the example which is specified above:

3. Expand the **Standard Wall Function** node and set *E* to `8.75`.

4. Expand the **Wall Roughness Parameters** node and set *C* to `0.25`.
5. Make sure that *RplusRough* (R^+) is set to `90` or greater so Eqn. (1199) is valid.
6. Make sure that *Roughness Limiter* is deactivated because the assumption $y \gg y_0$ from Eqn. (1201) is valid.
7. Expand the **Roughness Height** node and set the resulting calculated *Value* to `0.35 m`.

   The panel appears as follows:



Recommended Values for ABL Modeling

Use the settings described in this section when creating an ABL model.

Turbulent Viscosity Ratio

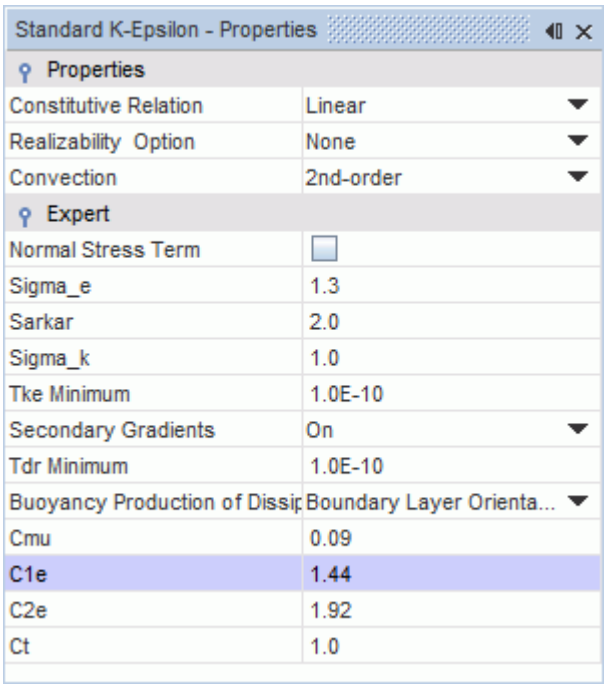
- Set the *Turbulent Viscosity Ratio* to `1E10`.

K-Epsilon

- Set *Cmu* (c_μ) according to the inlet velocity profile. Typical values are between `0.03` and `0.11`.
- Set the *Ce1* ($c_{\epsilon 1}$) coefficient using the following equation:

$$c_{\epsilon 1} = c_{\epsilon 2} - \frac{\kappa^2}{\sqrt{c_\mu}\sigma_\epsilon} \tag{1203}$$

The coefficients are defined in the turbulence modeling section, Standard K-Epsilon Model.



K-Omega

- Set *BetaStar* (β^*) according to the inlet velocity profile, where $\beta^* \approx c_\mu$.
- Set *Beta* (β) using the following equation:

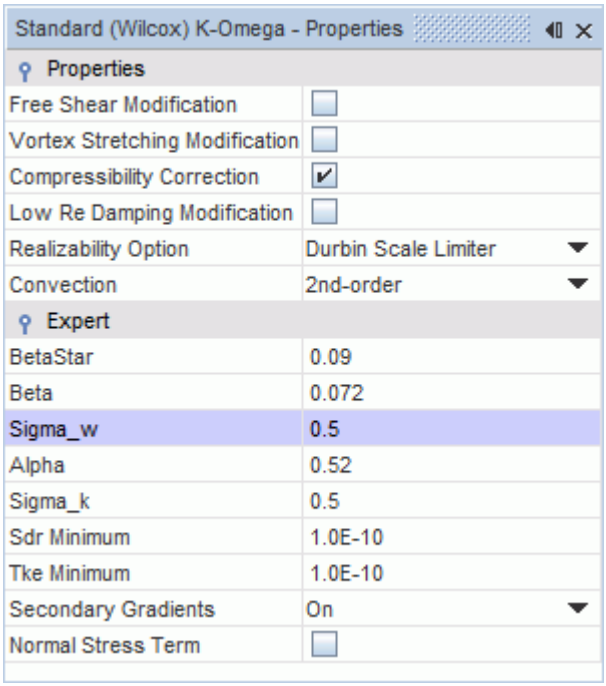
$$\beta = \frac{\beta^*}{1.25} \tag{1204}$$

- Set *Alpha* (α) using the following equation:

$$\alpha = \frac{1}{1.25} - \frac{\sigma_\omega \kappa^2}{\sqrt{\beta^*}} \tag{1205}$$

- Set *Sigma_w* (σ_ω) to `0.5` (its default value).

The coefficients are defined in the turbulence modeling section, Standard K-Omega Model.



Mesh Considerations

From sensitivity studies [182], use the following values when setting up the mesh:

- Make sure the shortest distance from the wall to a cell centroid, y, so that $y > 0.2r$ and $y \geq 10y_0$.
- Make sure that $y^+ \geq 30$, due to the Richards Hoxley model [182] starting in the logarithmic region of the boundary layer.

Bibliography

[180] *Garratt, J.R. 1992. "The atmospheric boundary layer", Cambridge University Press, pp. 1–15.*

[181] *Martinez, B. 2011. "Wind resource in complex terrain with OpenFOAM", MSc Thesis, Riso DTU, Denmark.*

[182] *Richards, P. J., and Hoxey, R. P. 1993. "Appropriate boundary conditions for computational wind engineering models using the k-epsilon model", Journal of Wind Engineering and Industrial Aerodynamics, **46**, pp. 145–153.*

Preparing CAD Models for Import

When you import a geometry file into STAR-CCM+, separate part surfaces are created for each named face. You can name faces in most CAD packages.

Instructions on how to name faces are provided for the following CAD packages:

Contents:

Preparing CAD Models in CATIA
Preparing CAD Models in SolidWorks
Preparing CAD Models in NX
Preparing CAD Models in Creo and Creo Elements/Pro

Preparing CAD Models in CATIA

To name a face in CATIA you must extract the face you want to name and then sew it back onto the geometry. The name of the extracted feature is the name of the face.

To name a face in CATIA:

1. Launch CATIA and load the geometry.
2. Select **Start** > **Shape** > **Generative Shape Design**.
3. Click (**Extract**) and select a face to extract.
4. Click **OK**.
5. Select **Start** > **Mechanical Design** > **Part Design**.
6. Click (**Sew Surface**) and select the extracted surface.
7. Deactivate the *Simplify Geometry* option.
8. Click **OK**.
9. Hide the **Extract** feature.
10. Rename the **Sew Surface** feature.

    This feature is imported into STAR-CCM+ as a separate surface with the name you specified.
11. Save the geometry.

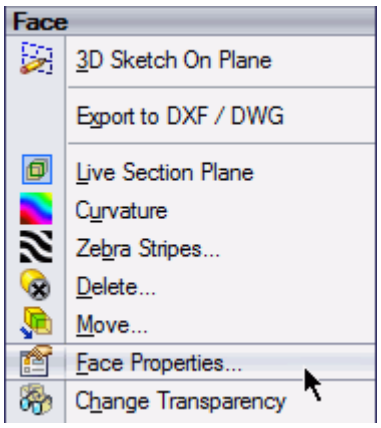
Preparing CAD Models in SolidWorks

In SolidWorks you can name a face by editing its face properties.

To edit the face properties:

1. Launch SolidWorks and load the geometry.
2. Right-click the face you want to name and click to expand the menu.

3. Select **Face** > **Face Properties...**



4. In the *Entity Property* dialog, enter the name of the face.
5. Click **OK**.
6. Save the geometry.

Preparing CAD Models in NX

In NX, you can name faces by editing their properties.

To edit the face properties:

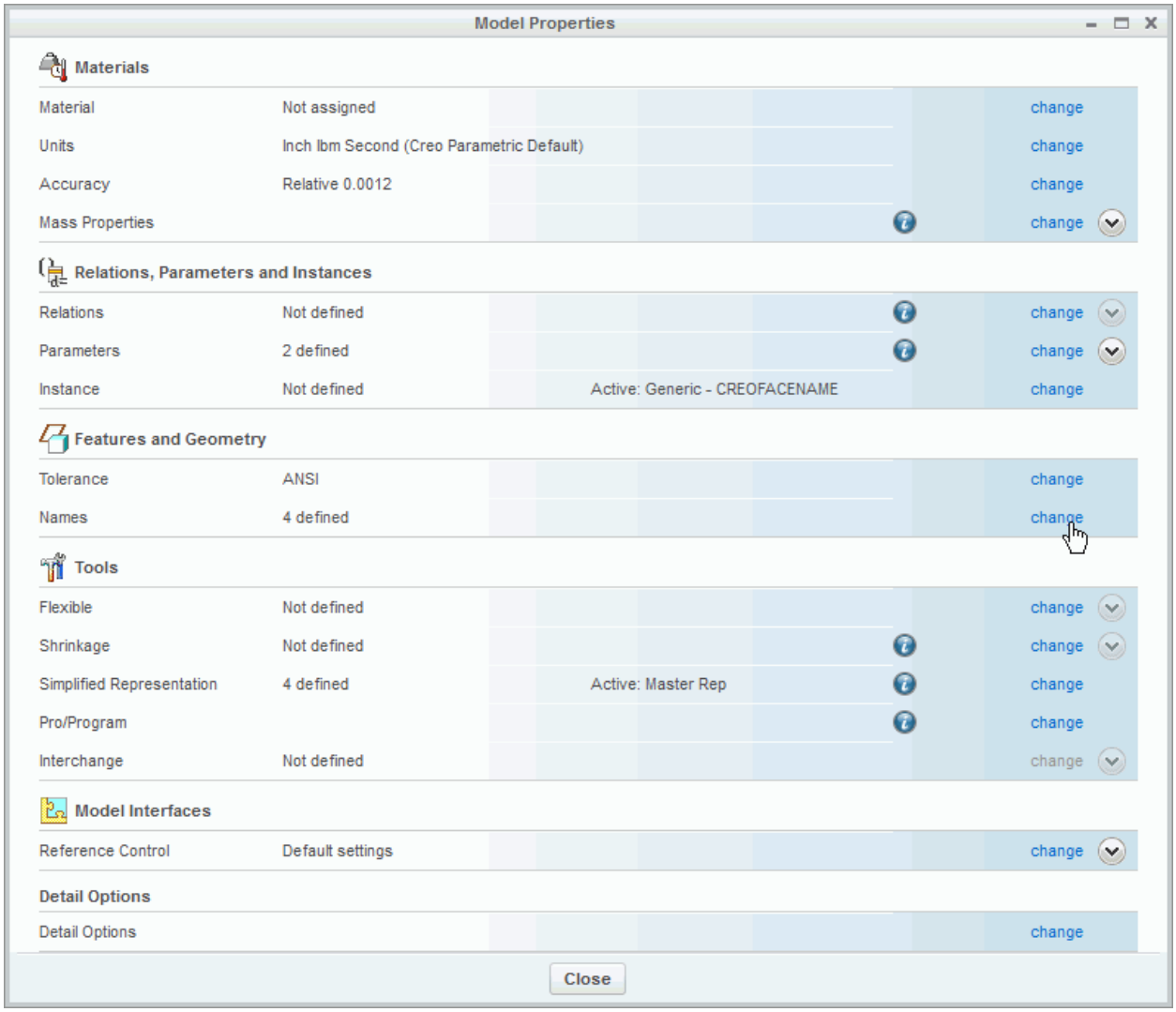
1. Launch NX and load the geometry.
2. Set the *Type Filter* to **Face**.
3. Right-click the face you want to name and select **Properties**.
4. In the *Face Properties* dialog, enter the name of the face.
5. Click **OK**.
6. Save the geometry.

Preparing CAD Models in Creo and Creo Elements/Pro

In Creo and Creo Elements/Pro, you can name faces through the Model Properties.

To name the faces:

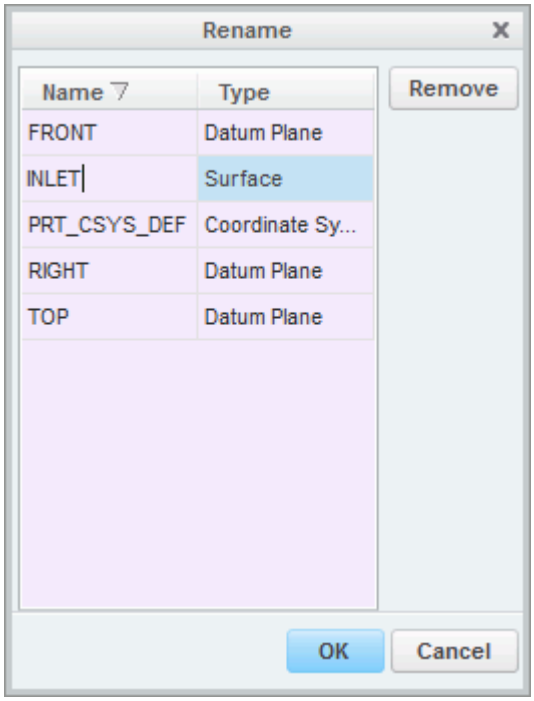
1. Launch Creo and load the geometry.
2. Open the *Model Properties* dialog:
   - In Creo, select **File** > **Prepare** > **Model Properties**.
   - In Creo Elements/Pro, select **File** > **Properties**.
3. In the *Model Properties* dialog, locate the *Features and Geometry* group box and click **change** next to the *Names* property.



4. With the *Rename* dialog open, select the face that you want to rename on the geometry model.

   An entry is added to the *Rename* dialog. If the face corresponds to an existing entry, the name is highlighted.

5. Enter the new name for the face and press `<Enter>`.



6. Click **OK**.
7. Click **Close**.
8. Save the geometry.

Migrating from Other Codes

This section contains guidelines for migrating from the following codes:

- Fluent
- STAR-CD

Contents:

Migrating from Fluent
Migrating from STAR-CD

Migrating from Fluent

This section lists details and limitations that guide you in migrating your Fluent files to STAR-CCM+.

To import Fluent `.cas` files into STAR-CCM+, remember:

- Only mesh data is imported. No boundary conditions, model selections, solver parameters or solution fields are imported. Therefore, any solution parameters or boundary conditions originally that were contained in Fluent must be redefined in STAR-CCM+.
- The Fluent importer in STAR-CCM+ reads the following sections of the file:

```
FLUENT_HEADER_SECTION = 1,
FLUENT_DIMENSION_SECTION = 2,
FLUENT_MACHINE_SECTION = 4,
FLUENT_NODE_SECTION = 10,
FLUENT_CELL_SECTION = 12,
FLUENT_FACE_SECTION = 13,
FLUENT_PERIODIC_SECTION = 18,
FLUENT_ZONE_SECTION = 39,
FLUENT_NAME_SECTION = 45,
```

- The following region, boundary, and interface types are recognized:

```
FLUENT_FLUID = 1,
FLUENT_INTERIOR = 2,
FLUENT_WALL = 3,
FLUENT_PRESSURE_INLET = 4,
FLUENT_PRESSURE_OUTLET = 5,
FLUENT_SYMMETRY = 7,
FLUENT_PERIODIC_SHADOW = 8,
FLUENT_PRESSURE_FAR_FIELD = 9,
FLUENT_VELOCITY_INLET = 10,
FLUENT_PERIODIC = 12,
FLUENT_POROUS_JUMP = 14,
```

```
FLUENT_SOLID = 17,
FLUENT_MASS_FLOW_INLET = 20,
FLUENT_INTERFACE = 24,
FLUENT_PARENT = 31,
FLUENT_HANGING_NODE_PARENT = 32,
FLUENT_OUTFLOW = 36,
FLUENT_AXIS = 37
```

Boundaries with those types are assigned to equivalent boundary types in STAR-CCM+. If boundary types are not specified in Fluent, STAR-CCM+ tries to infer them from the boundary names. Any name that cannot be recognized as a particular type is interpreted as a wall.

- Changing the internal boundaries to the porous jump type in Fluent helps retain them as interfaces in STAR-CCM+. Unlike Fluent, STAR-CCM+ only allows one set of interior faces per region. However, the STAR-CCM+ importer converts porous jump boundaries to interfaces.

Migrating from STAR-CD

This section lists details and limitations that guide you in migrating your STAR-CD files to STAR-CCM+.

To convert a STAR-CD simulation to run under STAR-CCM+, remember:

- STAR-CCM+ contains a *Model Selection* dialog where all physical and numerical models are chosen. The dialog contains an **Auto-select recommended models** checkbox which is ticked by default. Accept this setting unless you require special selections.
- The high-Reynolds number formulation of the “standard” K-Epsilon turbulence model is commonly used in STAR-CD simulations. On the other hand, the **Auto-select** option is likely to lead to a Realizable Two-Layer K-Epsilon model. In most cases, the Realizable Two-Layer K-Epsilon model has acceptable, or even better results.
- The default STAR-CD setting for initial k and ϵ values (as displayed on the STAR-CD GUI) is 0. When migrating your simulation, you are advised to use the STAR-CCM+ (non-zero) defaults. Better still, use values that are in keeping with representative velocity, turbulence intensity and turbulent viscosity values in your problem.
- There are important differences in the definition and required boundary conditions for inlet boundaries. Note in particular that, for mass flow inlets, the STAR-CD boundary conditions include velocity, density, static temperature and a “Flow Switch” setting. The corresponding STAR-CCM+ conditions are mass flow rate, flow angle, pressure and total temperature.
- For problems that involve porous media, STAR-CD uses settings for α and β. The corresponding settings in STAR-CCM+ are porous inertial resistance $\mathbf{P}_{\mathrm{i}}$ and porous viscous resistance $\mathbf{P}_{\mathrm{v}}$, respectively.
- Follow STAR-CCM+ guidelines on choosing suitable boundary conditions for your type of problem. For example, use pressure boundaries rather than outlets for compressible flows. As a result, change the definition of some of your existing STAR-CD boundary regions to more appropriate types.
- There is no direct equivalent for the STAR-CD Maximum Residual Error Tolerance for use as an overall solution convergence criterion in steady-state problems. Similar criteria exist in STAR-CCM+ and are available via the Monitors node, with each transport equation being solved. To form a composite stopping criterion:
  - Set up monitors for one or more of these equations.

  - Select the Create Stopping Criterion from Monitor option
  - Combine all monitor stopping criteria with the Maximum Steps criterion.
- The default setting for convective flux discretization is a likely cause for significant differences in convergence rate between the two codes. STAR-CD uses first-order upwind differencing. STAR-CCM+ uses a second-order differencing scheme which, in general, offers a better trade-off between accuracy and convergence rate.
- To produce the equivalent of a STAR-CD surface (or hidden-line) velocity vector plot near wall boundaries, use the **Cell Relative Velocity** field function.
- All of the STAR-CCM+ post-processing plot types are defined in a suitable **Scene**. At the end of the simulation run certain plots, such as the **Cell Relative Velocity**, require that you click the **Initialize** button before the plot can be produced.
- The STAR-CCM+, option **Thermodynamic Polynomial Data** for specific heat is equivalent to the Polynomial option in STAR-CD when specifying physical properties for chemical reaction components. The polynomial curve fits for Cp (plus enthalpy and entropy) are read in from the Chemkin database. With these curve fits, values for *Heat of Formation* and *Standard State Temperature* are implicit in the curve fit for enthalpy. STAR-CCM+ offers other options for specific heat: **Polynomial in T** (Cp is a piece-wise polynomial in temperature) or **Constant** (Cp is constant). If you use either option alone, specify the *Heat of Formation* and *Standard State Temperature* to determine the enthalpy in a simulation involving combustion.
- The default setting in STAR-CD for combustion problems using the Eddy Break-up model is to include reaction product mass fractions in the calculation of the chemical reaction rate (option Use Products). STAR-CCM+'s default setting for the rate calculation is 'Do not use products'. This means that the chemical reaction simulation starts as soon as fuel and oxidizer come into contact. If you tick the 'Use products' option, set up an ignition model explicitly under the **Ignitors** node .

Solving Transport Equation

The mechanics of continua (both solid and fluid) are described by transport equations derived from the assumption that mass, momentum, and energy are conserved in the continuum.

For more information on the transport equations, see Numerical Flow Solution.

Contents:

Gradient Computation
Algebraic Multigrid
Setting URF (Under-Relaxation Factor) Ramping
Transport Equation Solvers Reference

Gradients Model Reference

Gradients are used in several places within the transport equation solution methodology. The Gradients model lets you specify the gradient method and limiter method.

Gradients Model Reference

Theory	See Gradients.
Provided By	**[physics continuum]** > **Models** > *Gradient Metrics*
Example Node Path	**Continua** > **Physics 1** > **Models** > **Gradients**
Requires	The **Gradients** model is automatically selected when the **Space** model is defined.
Properties	Key properties are: *Gradient Method* and *Limiter Method*. See Gradients Model Properties.

Gradients Model Properties

Gradient Method

Specifies which gradient method to use.

- **Hybrid LSQ-Gauss**
  Selects the Hybrid Gauss-Least Squares Method (default option).
- **Green-Gauss**
  Selects the Weighted Least Squares method.

Limiter Method

Specifies which limiter method to use.

- **MinMod**
  Selects the MinMod method that is given by Eqn. (1621).
- **Venkatakrishnan**
  Selects the Venkatakrishnan method that is given by Eqn. (1620).
- **Modified Venkatakrishnan**
  Selects the Modified Venkatakrishnan method that is given by Eqn. (1622).

Custom Accuracy Level Selector

If necessary, you can introduce more damping in the numerical scheme and so improve convergence.

This property specifies the accuracy of the numerical scheme at any level between 1.0 (1st-order) and 2.0 (2nd-order). You can use any real value between 1.0 and 2.0. The default value is 2.0, meaning no accuracy changes.

When computing steady-state solutions of complex flow problems using high-order discretization, you can sometimes encounter convergence stall or even lack of convergence. These issues are due to flow instabilities, sometimes local ones, that are generally caused by numerical artifacts.

One solution for these situations is to tune the numerical scheme dissipation to a user-specified level, between the first order and second order of accuracy.

The accuracy level selector applies only when the second-order scheme is selected. It is implemented as a global scaling applied to all reconstruction gradients, applying a single blending factor to all equations, thus affecting the convection term for all transport equations.

Note: Gradients used for diffusion are not affected by this scaling.

Verbose

When activated, provides more output while the simulation is running. This property can be useful for debugging problems when they occur.

Acceptable Field Variation (Factor)

Sets the allowed variation of the field due to TVB Gradient Limiting, as a fraction of the largest difference between local maximum and local minimum anywhere in the simulation. Default 0.05.

Maximum Safe (Positive) Skewness Angle (deg.)

Below the positive only skewness angle, no related specific corrections are applied to least-squares computed gradients. Default 75.0.

Normalized Flat Cells Curvature Factor

Acceptable ratio between the tangent of cell-skewness angle and the aspect ratio of the cell. Default 1.0.

Least-Squares Quality Criterion

Makes gradient corrections for cells with poor least-squares tensor quality. Default on and is read-only.

Cell Skewness Criterion

Makes gradient corrections for cells with large skewness angle. Default on.

Chevron-Cell Criterion

Makes gradient corrections for all chevron cells. Default on.

Least-Squares Tensor Minimum Eigenvalues Ratio

Minimum admissible value of ratio between the minimum and maximum eigenvalue of the least-squares tensor. Default 0.1.

Use TVB Gradient Limiting

TVB (Total Variation Bounded) limiting prevents excessive gradient limiting, which can cause a slow or stalled convergence. See Total Variation Bounded Gradient.

Minimum Unsafe (Positive) Skewness Angle (deg.)

Above the positive skewness angle and for all negative skewness angle cells, gradients are corrected to ensure robustness. Default 88.0.

Flat Cells Curvature Criterion

Makes gradient corrections for flat cells with sensible curvature. Default on.

AMG Linear Solver Reference

The algebraic multigrid methods solve the discrete linear system iteratively. The **AMG Linear Solver** node contains the objects that you would use for Multigrid Cycling Strategies.

AMG Linear Solver Reference

Theory	See Algebraic Multigrid.
Example Node Path	**Solvers** > **Multiphase Segregated Flow** > **Phase Coupled Velocity** > **AMG Linear Solver**
Properties	Key properties are: *Cycle Type* and *Convergence Tolerance*. See AMG Linear Solver Properties.

AMG Linear Solver Properties

Max Cycles

Specifies the maximum number of multigrid cycles to run in case the residuals of the linear system remain above *Epsilon* and the *Convergence Tolerance* not be met.

Verbosity

Specifies the amount of feedback (directed to the console window) by the AMG Linear Solver.

- **None**
  The AMG Linear Solver runs silently. There is no feedback.
- **Low**
  Residuals of the linear system are printed at the completion of each multigrid cycle.
- **High**
  Same as Low, plus the number of multigrid levels and number of equations at each level are printed once all multigrid cycles are complete.
- **Diagnostics**
  Provides the highest degree of information regarding performance of the AMG system. This option can affect performance, and therefore is not recommended for normal iteration.

Convergence Tolerance

Factor by which residuals of the linear system (that are above the value of *Epsilon*) must be reduced before multigrid cycling stops (provided *Max. cycles* have not been reached). For example, a convergence tolerance of 0.01 requires that cycling continue until residuals of the linear system have dropped by two orders of magnitude. A tolerance of 0.001 requires a drop of four orders of magnitude, and so on.

Cycle Type

Specifies the multigrid cycling strategy. See Multigrid Cycling.

- **F Cycle**

After restriction one W-cycle is applied on the next coarsest level, followed by one V-cycle (also on the next coarsest level).

- **V Cycle**
  After restriction one cycle (also a V-cycle) is applied on the next coarsest level.
- **W Cycle**
  After restriction two cycles (also W-Cycles) are applied on the next coarsest level.

  The three Fixed Cycle nodes have the following properties:

Pre-Sweeps

Number of smoothing sweeps at the start of each cycle.

Post-Sweeps

Number of smoothing sweeps at the end of each cycle.

Max Levels

Maximum number of coarse levels in the sequence.

- **Flex Cycle**
  Rather than using all multigrid levels in a regular pattern, the residuals are monitored after every sweep on a given grid level. If the ratio of the residuals exceeds a given threshold, the solution continues on a coarser level. If the residual on a given level is reduced more than a specified tolerance, the solution moves to a finer level. A limit is further imposed on the number of sweeps that are allowed at any level.

  This type of cycle is a more economical cycling strategy for linear systems that are not stiff.

  The Flex Cycle node has the following properties:

Restriction Tolerance

Ratio of two successive residuals of the linear system below which progression to the next coarsest level is suspended.

Prolongation Tolerance

Specifies the amount by which the residuals of the linear system must be reduced before the solution is stopped on any given level.

Sweeps

Number of smoothing sweeps at the start of each cycle.

Enable Direct Solver

Controls whether to use the direct solver for AMG coarse levels.

Maximum Direct Solver Equations

Sets the maximum number of equations for the direct solver.

Epsilon

Cutoff value for residuals of the linear system below which convergence is considered complete and multigrid cycling stops, regardless of the *Convergence Tolerance* setting. For example, an Epsilon value of 1.e-08 implies that cycling stops when residuals of the linear system drop below 1.e-08.

Group Size Control

Specifies how the group size must be set.

- **Auto**
  The group size is set automatically to 2 in two-dimensional cases and 4 in three-dimensional cases.
- **Specified**
  Allows a user-specified group size. Adds a **Group Size** sub-node to this node.

Group Size

The number of equations on a given multigrid level that are agglomerated to form one equation on the next coarsest level. This value is read-only and reports the current group size being used, as determined by the Auto setting or from the **Group Size** sub-node when Specified.

Relaxation Scheme

Specifies the method for iteratively correcting (relaxing) the linear equation that is set during multigrid cycling. See AMG Solver.

- **Jacobi**
  A slow, simple method.
- **Gauss-Seidel**
  Provides better convergence. The default option.
- **ILU**
  A more efficient but more expensive method.

Acceleration Method

Specifies the acceleration type for the AMG solver.

- **None**
  No acceleration method used.
- **Conjugate Gradient**
  Uses the Conjugate-Gradient Method with the AMG solver as the preconditioner. This method is available with the Constant Density and Segregated Flow models only. It is accessed through the Expert Properties of the **AMG Linear Solver** sub-node of the **Segregated Flow** > **Pressure** solver node.
- **Bi-Conjugate Gradient Stabilized**
  Uses the Bi-Conjugate Gradient Stabilized Method to solve nonsymmetric (general) linear systems.

Scaling

Controls the manner in which corrections are prolongated from the next coarsest level. Scaling is only appropriate when the linear system matrix A (Eqn. (1686)) is symmetric positive-definite.

- **Auto**
  Scaling is automatically applied to the coarse level corrections to achieve optimal convergence.
- **Disabled**
  Coarse level corrections are prolongated without any scaling applied (the default).

Contents:

Troubleshooting the AMG Linear Solver

See Also:

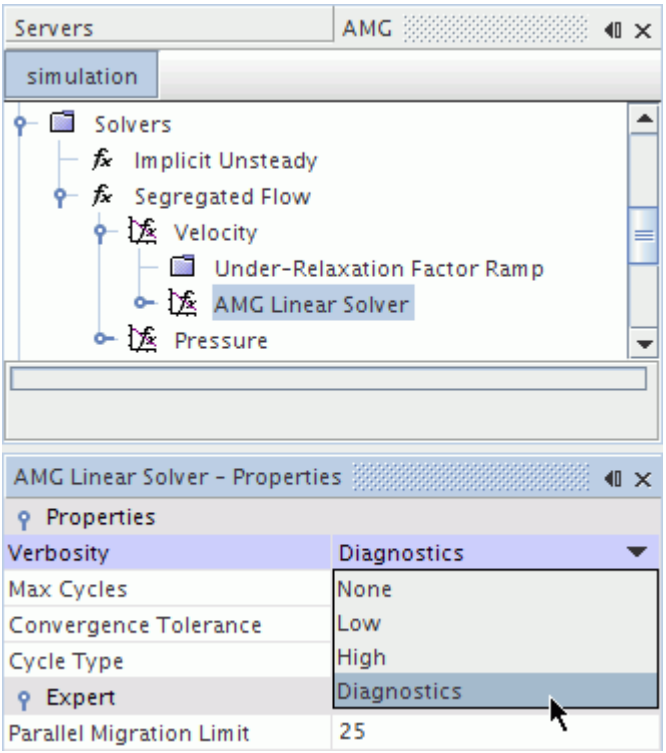
Setting Algebraic Multigrid Parameters

Setting Algebraic Multigrid Parameters

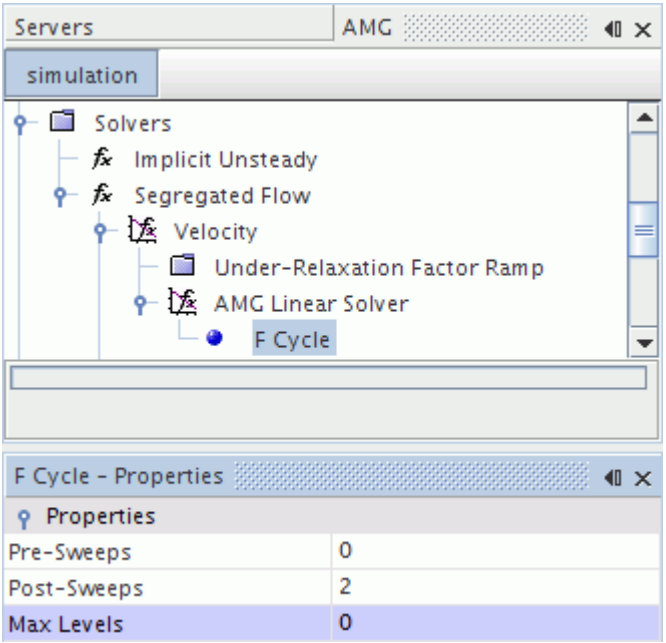
Troubleshooting the AMG Linear Solver

If you suspect that the **AMG Linear Solver** causes convergence problems, you can try the following troubleshooting procedure:

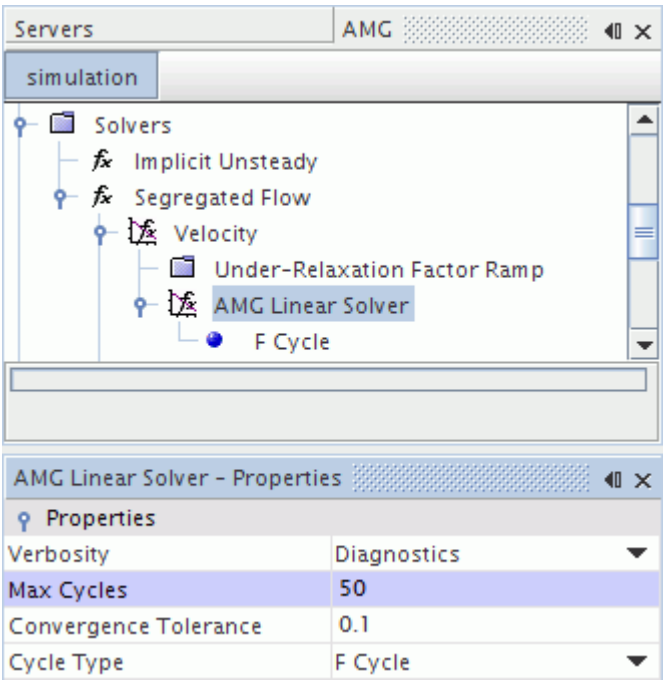
1. In the *AMG Linear Solver Properties* window, set the *Verbosity* to **Diagnostics**.



2. Continue to iterate, and in the *Output* window, identify the solver for which AMG is crashing.
3. For the solver in which the **AMG Linear Solver** is crashing, deactivate the multigrid procedure.
   a) Select the **Cycle** node.
   b) In the *Properties* window, set *Max Levels* to `0`.



4. Select the **AMG Linear Solver** node and, in the *Properties* window, change *Max Cycles* to a larger value (around 50 - 100).



5. Run the simulation again.

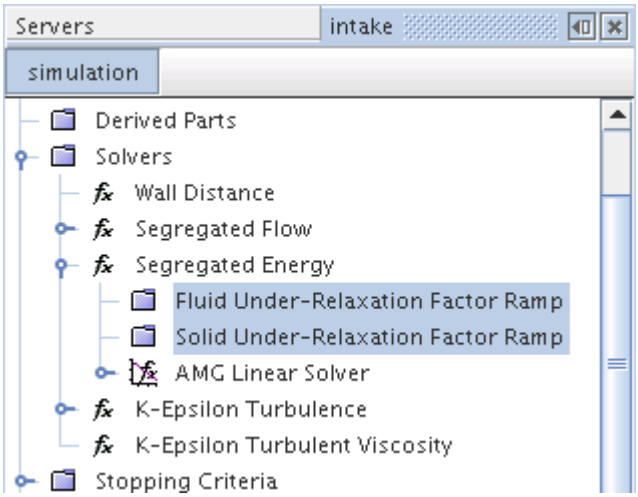
If divergence still occurs, it indicates an error in the coefficient of the linear system coming from the discretization. The problem is not located in the **AMG Linear Solver** – it could come from a meshing issue, for example.

If Step 2 resolves the problem, it could indicate a technical issue inside the **AMG Linear Solver**. Contact your Siemens PLM Software support engineer.

Under-Relaxation Factor Ramp Reference

The **Under-Relaxation Factor Ramp** node is a subnode of various solvers. You can activate or deactivate linear URF ramping in its properties.

Variations exist for this object, depending on the type of model you activate. For example, the Segregated Energy solver has separate ramps for the Fluid and Solid URFs.



Under-Relaxation Factor Ramp Properties

Ramp Method

- **No Ramp**
  Leaves off ramp calculation.
- **Linear Ramp**
  Activates linear ramp value calculation. The **Linear Ramp** node becomes available under the **Under-Relaxation Factor Ramp** node.

  The **Linear Ramp** node has the following properties:

Start Iteration

Specifies the iteration on which to begin ramping, typically 1.

End Iteration

Specifies the iteration on which to stop ramping, typically within the first 10–100 iterations.

Initial Value

Uses a text entry to specify the value before ramping starts.

Transport Equation Solvers Reference

The solvers that control the transport equation models have common properties.

Transport Equation Solver Expert Properties

Unless you are thoroughly familiar with the discretization techniques that are used in STAR-CCM+, do not change any properties within the *Expert* category. The values in that category have been optimized for accuracy and performance.

The following expert properties are available with all solvers that control transport equation models.

Reconstruction Frozen

When activated, STAR-CCM+ does not update reconstruction gradients with each iteration, but rather uses gradients from the last iteration in which they were updated. Activate **Temporary Storage Retained** in conjunction with this property. This property is deactivated by default.

Reconstruction Zeroed

When activated, the solver sets reconstruction gradients to zero at the next iteration. This action means that face values used for upwinding (Eqn. (1600)) and for computing cell gradients (Eqn. (1612) and Eqn. (1623)) become first-order estimates. This property is deactivated by default. If you deactivate this property after having it active, the solver recomputes the gradients on the next iteration.

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Post-Processing

STAR-CCM+ allows you to post-process the solution after each iteration or time-step, (while the simulation is running), as well as when the simulation completes. For this reason, you generally create post-processing objects before running the simulation. Post-processing objects are also known as analysis objects.

All simulation data is stored in field functions. Field functions give you access to mesh data, geometry data, data computed by solvers, and to any data that is derived from solver data. You can also define your own field functions and access their data using the same post-processing objects as for simulation data.

STAR-CCM+ provides three categories of post-processing object:

- Reports: Four different types of report exist, including one that accepts a user-defined expression in the same format as a user-defined field function. In order to obtain a history of the reported value, reports can be monitored while a simulation is running.
- Scenes: If these scenes are created before you run the simulation, they are updated continuously as the solution progresses.
- Plots: You can create monitor and residual plots, XY plots, and histogram plots.

For each of these strategies, you generally choose the parts from which the solution data is extracted. These parts can be regions, boundary surfaces, geometry parts, or derived parts, depending on the type of analysis. Derived parts are additional lines and surfaces that you create in addition to the surfaces already present on geometry parts and regions.

Monitors are used to sample and save solution data while the simulation is running. Some monitor types are made available in response to the solvers you have active. Additional monitors can be created from a report.

Several features of STAR-CCM+ require that you define a trigger for when an update event takes place. See Triggers and Update Events.

For data that have been obtained over time in a transient analysis, such as pressure data at a point, data set functions are provided. These data set functions permit spectral analysis through Fast Fourier Transforms and related tools.

Contents:

Analysis Workflow
Accessing Solution Data through Derived Parts
Visualizing the Solution
Recording and Viewing Solutions
Sampling Data Values During and After Running
Reporting Results

Monitoring the Solution
Plotting Results
Exploring Solutions with Data Focus
Triggers and Update Events
Using Data Set Functions

Analysis Workflow

The key steps in successfully analyzing simulation results are to: (1) be clear on what quantities you want to extract; (2) locate or define surfaces on which to obtain the data; and, (3) present the data in visual scenes or numeric output.

1. Specify the quantities that you require, and identify the field functions that support them:
   a) Decompose the engineering design goal into specific quantities that you must extract from the simulation.
   b) Within a simulation that already has the physics models selected, under **Tools** > **Field Functions**, review the list of field functions available for those models. Some solvers expose additional field functions when you activate the property, *Temporary Storage Retained*.
   c) Review the built-in reports in case one already matches a quantity that you intend to compute. To do this, right-click **Reports** and hover on **New Report**.
   d) If you cannot directly access a field function that matches the required data, create a user-defined field function whose definition computes the quantity. STAR-CCM+ provides an expression editor that helps you create these functions.
2. If existing simulation surfaces are not sufficient for you to use in plotting, visualizing, or reporting data, create additional derived parts.

   For details about these parts, see Accessing Solution Data through Derived Parts.

   - To cut through surfaces or volumes with a geometric shape, use implicit sections. See Defining Implicit Sections.
   - To represent subsets of scalar or vector values, use parts that change with those values, such as a threshold. Specific examples are:
     - isosurface
     - threshold
     - scalar warp surface
     - vector warp surface
   - To depict particle paths, use streamlines. See Defining a Streamline.
   - To take samples of data at specified coordinates or intervals, use a probe. See Defining Probes.
3. In unsteady simulations, prepare for analysis before you run the solver, as the simulation file does not store complete field data at every time step.
   a) To store selected field data at regular intervals, create a solution history file. See Creating a Simulation History File.
   b) To export hardcopies of visualization scenes, configure the **Update** properties of each scene you wish to export.
4. Create the analysis objects:

a) To create reports and associated monitors, see Sampling Data Values During and After Running. This step includes monitors that you use to assess convergence while the solution develops.

b) To create scalar or vector scenes, see Visualizing the Solution.

c) To create XY plots (or graphs) of data, see Plotting Results.

d) To specify certain colors and shades for displaying data, use colormaps. See Using Colormaps.

5. Decide how you want STAR-CCM+ to update results while the simulation is running. You can control the frequency of updates for scenes, plots, reports, and monitors. To customize these updates, use triggers.

   See Triggers and Update Events.

6. Run the simulation.

7. After the simulation completes, you can export hardcopies of scenes or plots that you created before. You can also perform additional post-processing, as follows:

   a) To render detailed scalar data in a volume, particularly for complex 3D flows, use volume data resampling. See Defining a Resampled Volume Part.

   b) To render detailed vector fields on a surface, activate Line Integral Convolution on a vector displayer.

   c) To apply signal processing on time-history data from points, lines, or surfaces, see Using Data Set Functions.

   d) To view the flow in motion, you can animate data, including simulation history (`.simh`) data, in scenes. You can also record the motion onto separate video files. See Animating Scenes and Animating Solution Views.

See Also:

Creating a User Field Function

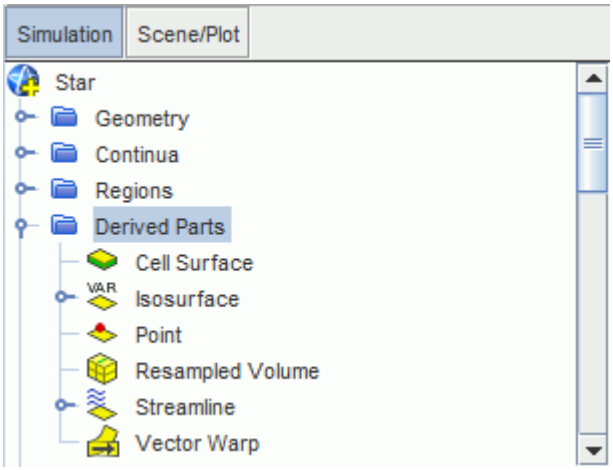
Accessing Solution Data through Derived Parts

Although STAR-CCM+ can display, analyze, and report simulation data on regions, boundaries, and parts, there are many cases when you want to access data in other entities of the solution space. Derived parts provide a mechanism for defining additional entities on which you can access solution data.

A part that is based on another part is known as a derived part. A derived part is created using one or more other parts as parent parts, or inputs, and it is dependent on these parent parts. If any of the parent parts change, all parts depending on those parent parts are automatically recalculated and redisplayed to reflect the change.

For example, consider the following: An isosurface of pressure is created in a three-dimensional region and a plane-section is created from the isosurface resulting in a curve in space. The parent of a plane is the isosurface, and the parent of an isosurface is the three-dimensional region. If the region is modified somehow, for instance by either the mesh or the solution changing, first the isosurface is recalculated, and the plane follows. In this way, consistency is maintained for all parts.

The **Derived Parts** manager node contains all of the derived part nodes in the simulation.



You can add and edit any of various derived parts and use them to examine solution data. Creating and editing can be done visually using the powerful in-place dialog. A derived part gets its data when you select input (parent) parts for it. Certain derived parts can be defined using local coordinate systems.

You can control the naming of derived parts. If a derived part is no longer needed, it can be deleted. You can also copy a derived part to use it as a starting point for a new part, or to create a deep copy for use in a different simulation. It is possible to manipulate several of these derived parts visually using tools. Some derived parts use data from an external file. You can also export data that the derived parts extract.

Derived Part Types

Cell Surface	Represents all the cells adjacent to or intersected by a specified input part.
Isosurface	Surface over which a scalar function has a constant value.
Probe	Takes samples of data in a region based on coordinates or intervals that you specify.
Section	Based on planes or geometrical shapes, which can be cut through one or more volumes or surfaces.
Streamline	Represents particle paths.
Constrained Streamline	Represents particle paths that are constrained to lie in a surface.
Vortex Core	Detects vortex cores in a flow field.
Separation/Attachment Line	Detects where flow separates from or attaches to a surface.
Threshold	Isolates a group of cells based on their relationship to specified scalar values.
(m', theta) warp	Unwraps results to a flat plane.
Resampled Volume	Volume rendering is used to display volumetric objects that are defined in a 3D domain.
Warp	Moves the point coordinates on the surfaces of its input parts along a selected vector multiplied by a scale factor (vector warp), or along the point normal vectors by an amount equal to a specified scalar times a scale factor (scalar warp).

Contents:

Defining Derived Parts: General Workflow
Using the Visual Tools
Reading an External File
Exporting to an External File
Calculating the Surface Area of a Part
Defining Implicit Sections
Defining an Isosurface
Defining a Threshold
Defining a Scalar Warp Surface
Defining a Vector Warp Surface
Defining a Streamline
Defining an (m',theta) Warp
Defining a Separation/Attachment Line
Defining a Vortex Core Part
Defining Probes
Defining a Cell Surface
Defining a Resampled Volume

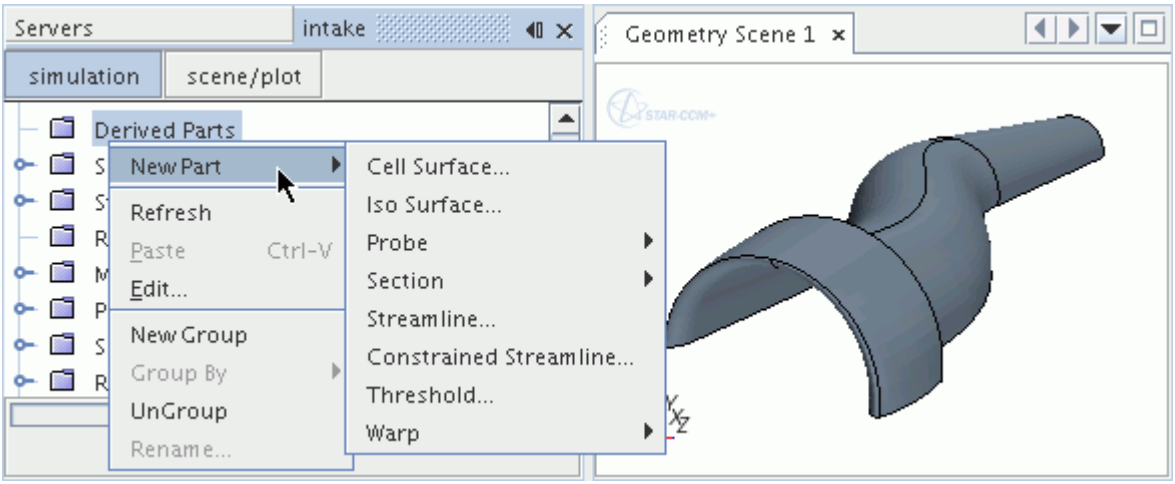
Derived Parts Reference
Derived Parts Bibliography

Defining Derived Parts: General Workflow

You can define a derived part interactively in a scene, through the derived part in-place dialog, or by setting the derived part node properties in the *Properties* window. Defining a derived part through the in-place dialog gives you access to graphical tools that you can use in an active scene.

The following steps outline the workflow for adding a derived part to a simulation and setting its general properties. For instructions on setting properties that are specific to the derived part type, refer to the relevant derived part sections.

1. If you wish to define a derived part interactively in a scene, activate a scene in the *Graphics* window.
2. Right-click the **Derived Parts** node, select **New Part** and choose the type of derived part to create.

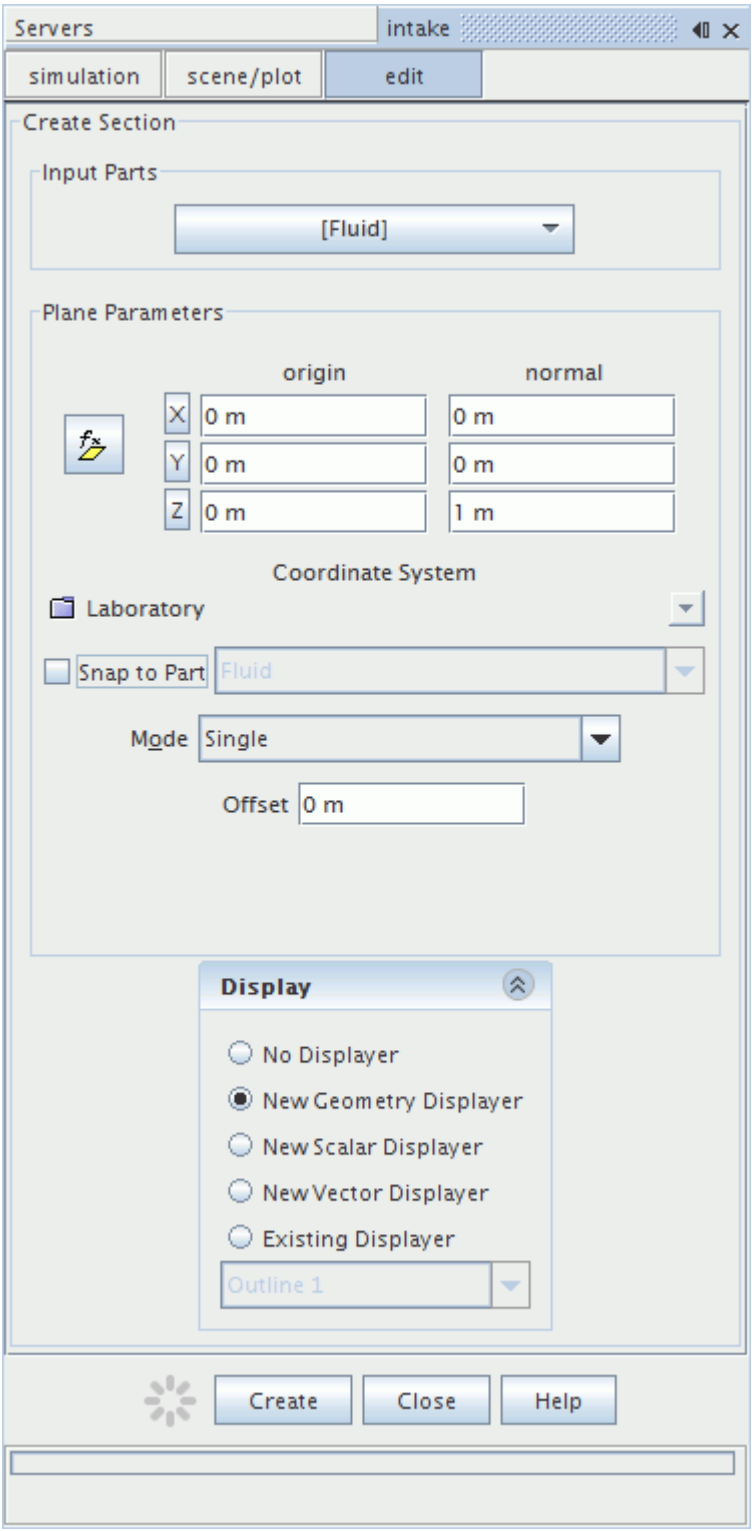


The create in-place dialog appears as soon as you execute this action, providing that:

- A scene is active in the *Graphics* window.
- Another derived part is not being edited in the current scene.

If you do not activate a scene before executing this action, STAR-CCM+ immediately add the new derived part to the simulation tree. You can then define the derived part by setting its properties in the *Properties* window.

The following screenshot is an example of the in-place dialog; depending on the derived part type, the in-place dialog shows different properties and inputs. The **Input Parts** and **Display** options are common to all types of derived parts.



This dialog is connected to the particular scene that is active when it opens. It disappears when the scene becomes inactive (for example, when another display obscures it), and reappears when the scene becomes active again.

When created, derived parts have a default name that is based on their type. When an additional derived part of the same type is created in a simulation, a number is added to its default name, such as `plane`

`section N.N` represents one plus the number of derived parts of that type existing in the simulation. Derived parts can be renamed.

Specify the input parts for the derived part. An input part can be envisioned as a parent or source part from which the derived part obtains its data.

3. To select the input boundaries or regions that you want to include in a derived part, you can use one of the following methods:
   - In the in-place dialog, click **Select...** in the *Input Parts* box, then use the *Select Objects* dialog to select the input parts.
   - In the derived part node *Properties* window, specify the input parts by setting the *Parts* property.

When defining a derived part interactively, you can use the *Display* group box in the in-place dialog to add the part to the currently active scene:

4. In the *Display* group box, select one of the displayer options.

   You can assign a derived part to a new or existing displayer of type outline, geometry, scalar, or vector.

   The (**Up-Arrow**) button, at the top of the group box, lets you minimize the group box temporarily.

   When the group box is minimized, its only visible parts are the title bar and a (**Down-Arrow**) button. Click anywhere within the title bar to restore the group box. The *Display* group box does not appear if you are editing a part that is already displayed in the active scene. If you are editing a part that is not in the active scene, the list appears as a separate dialog ahead of the in-place dialog.

You can link some types of derived part to a local coordinate system, and thus simplify your direction calculation by changing the frame of reference.

Linking to a coordinate system can be useful if, for example, you are modeling a certain part that is oriented differently from standard (laboratory) coordinates. Additionally, you can change the coordinates for more than one derived part by editing a single object, rather than modifying each derived part individually.

The following types of derived parts use local coordinate systems:

- Plane Section
- Constrained Plane Section
- Cylinder Section
- Sphere Section
- Point Probe
- Line Probe
- Presentation Grid

5. To use a local coordinate system with a derived part:
   - In the in-place dialog, select a local coordinate system in the *Coordinate System* group box.
   - In the derived part node *Properties* window, set the *Coordinate System* property.

   For more information on the available general properties, see Derived Parts Reference.
6. In either the in-place dialog or the *Properties* window, complete the derived part definition by specifying the settings that are specific to the type of derived part.

   For more information on the available properties, see Specific Derived Parts Reference.

7. If you are using the in-place dialog, click **Create** to put your specifications into effect, and to add the part to the simulation tree, then click `Close`. Alternatively, click **Close** to cancel the creation process and discard all your settings.

   If you click **Create** more than once, you get an additional identical derived part each time. Also, if the *Display* group box is set for a new displayer, STAR-CCM+ adds a new displayer each time you click **Create**.

You can edit existing derived parts as needed, using the in-place dialog or the derived part *Properties* window.

8. To edit an existing derived part, you can use one of the following methods:
   - Edit the derived part interactively in a scene:
     1. Activate a scene in the *Graphics* window.
     2. Right-click the relevant **[Derived Part]** node and select **Edit Part in Current Scene**.

        The **Edit Part in Current Scene** option is only available when:
        - A scene display is active.
        - Another derived part is not being edited in the current scene.
        - The derived part that you are selecting is not already being edited in a scene.
     3. Edit the relevant properties using the in-place dialog.
     4. Click **Apply** to put your changes into effect, then click `Close`. Alternatively, click `Close` to cancel the editing process and discard all your changes.

        The part that you are editing is modified each time that you click **Apply**. Also, if the *Display* group box is set for a new displayer, clicking the **Apply** button produces the additional displayer.
   - Edit the derived part properties:
     1. Select the relevant **[Derived Part]** node.
     2. Edit the relevant properties in the *Properties* window.
9. Once you have finished, rotate and zoom in the display to get a clear view of the plane section. If you did not assign the part to the current scene, you can drag and drop its node into the visualization display.

Using the Visual Tools

Included in STAR-CCM+ are tools that let you manipulate derived parts visually.

If a part becomes highlighted while you are using a tool, you can hold down the `<Ctrl>` key and click anywhere in the display to deactivate the highlighting.

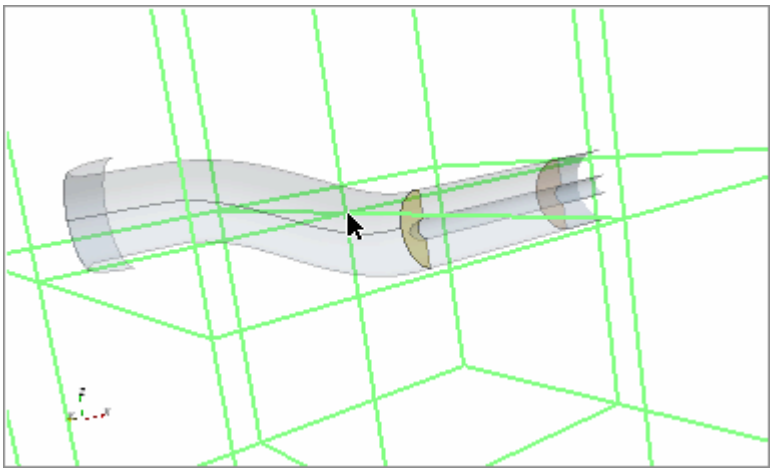
Contents:

Using the Point Tool
Using the Line Tool
Using the Implicit Plane Tool
Using the Bounded Implicit Plane Tool
Using the Explicit Plane Tool
Using the Sphere Tool
Using the Resampled Volume Tool

Using the Point Tool

The point tool allows you to position a point in three-dimensional space using a three-dimensional cursor.

The cursor has an outline bounding box, axes-aligned cross-hairs, and axes shadows.



This widget operates in the following ways:

- To move the point, click any widget line and “slide” the point into position.
- To scale the widget, click and drag with the right mouse button up or down. The upward direction makes the widget bigger, while the opposite direction makes it smaller.
- To translate the widget, drag it while holding down the middle mouse button. All of the translation interactions can be constrained to one of the x, y, or z axes by using the `<Shift>` key.

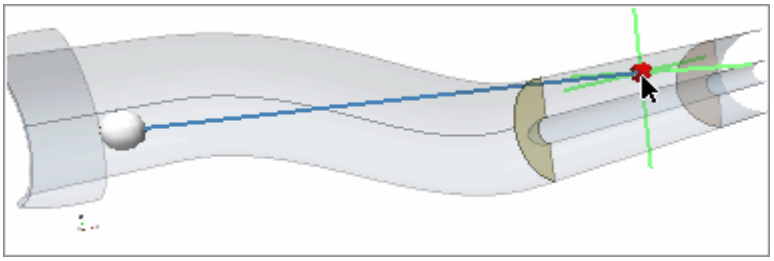
The constrained translation/sliding action (when the `<Shift>` key is depressed) along the axes is based on a combination of a “hot” spot around the cursor focus plus the initial mouse motion after selection. That is, if you select an axis outside of the hot spot, then the motion is constrained along that axis. If you select the point widget near the focus (within the hot spot), the initial motion defines a vector which is compared to the x-y-z axes. The motion is constrained to the axis that is most parallel to the initial motion vector.

- The widget can be picked even when it is “behind” other actors. It is not a bug: this feature is intended.

Using the Line Tool

The three-dimensional line tool defines a line that can be interactively placed in a scene.

The line has two handles (at its endpoints), and the line can be picked to translate it in the scene.

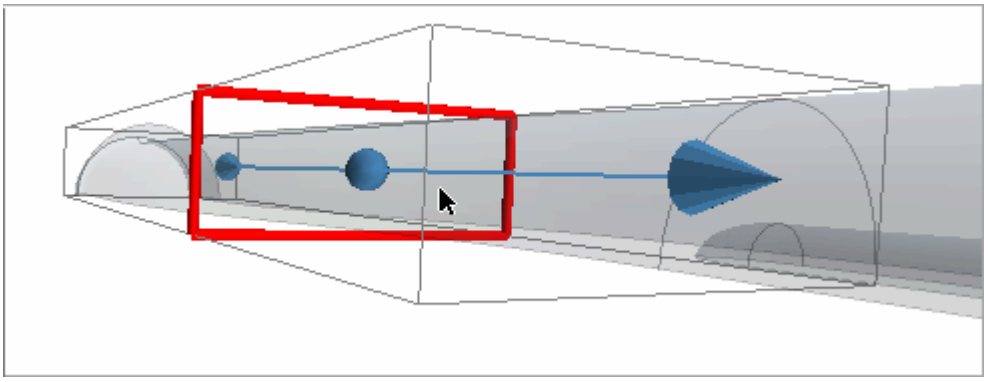


- To orient and stretch the line, click one of the two end-point handles and drag it. (The other end point remains fixed.)
- To scale the line from its center, click and drag with the right mouse button up or down. The upward direction makes the line bigger, while the opposite direction makes it smaller.
- To translate the line, drag it while holding down the middle mouse button. All of the translation interactions can be constrained to one of the x, y, or z axes by using the `<Shift>` key.
- The handles and line can be picked even when it is “behind” other actors. It is not a bug: this feature is intended.

Using the Implicit Plane Tool

This three-dimensional tool defines an infinite plane that can be interactively placed in a scene.

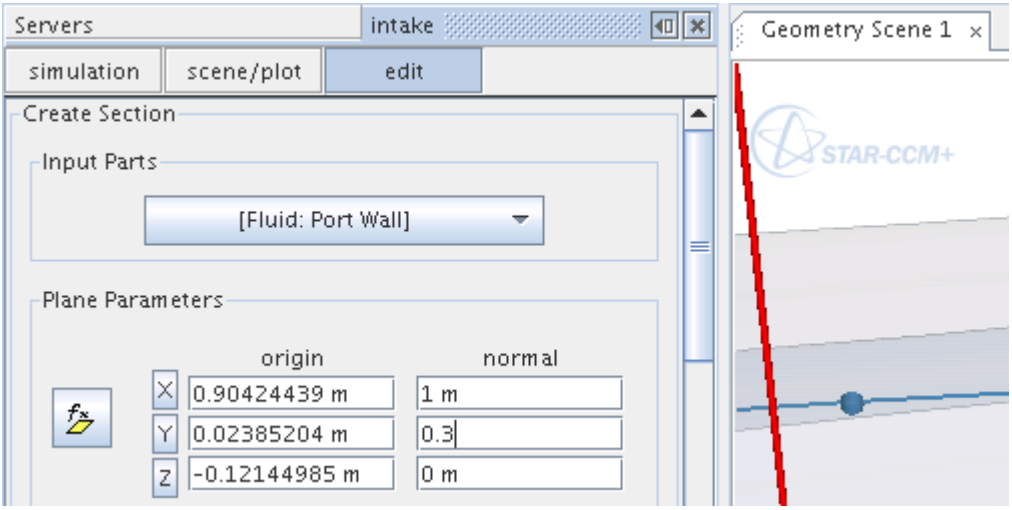
The widget appears as a plane with a normal vector; the plane is contained by a bounding box. The intersection edges of the plane and the bounding box are shown tubed.



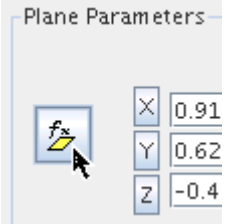
The normal vector handle can be selected and moved to rotate the plane; the plane itself can be selected and translated in various directions.

- The scale of the bounding box is set automatically to the combined size of the input parts that you select for the plane.
- The widget can be picked even when it is “behind” other actors. It is not a bug: this feature is intended
- If you select the normal vector, you can rotate the plane arbitrarily. You can also translate the plane along the normal by clicking and dragging.
- You can arbitrarily move the plane origin by clicking and dragging the plane with the middle mouse button.
- You can constrain the translation of the plane by ticking the **Snap To Part** option in the *Plane Parameters* group box, and selecting the part in the drop-down menu. This option makes it easy to align a plane with a certain part. (This action is essentially snapping to the bounding box of the part.)
- To refine the position of the plane, you can adjust the coordinates with text entries.

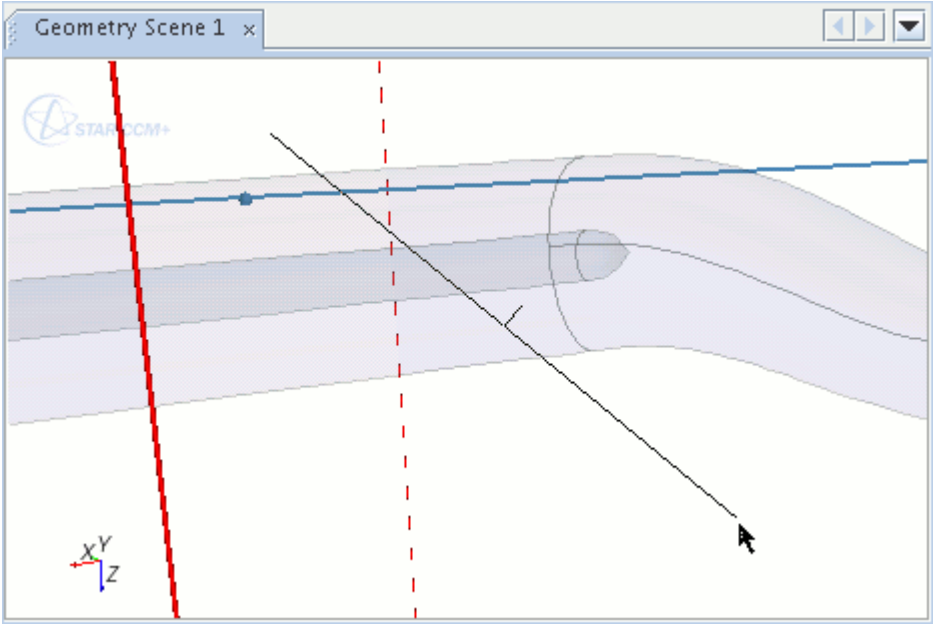
  After each text entry, press `<Enter>` to apply your adjustment.



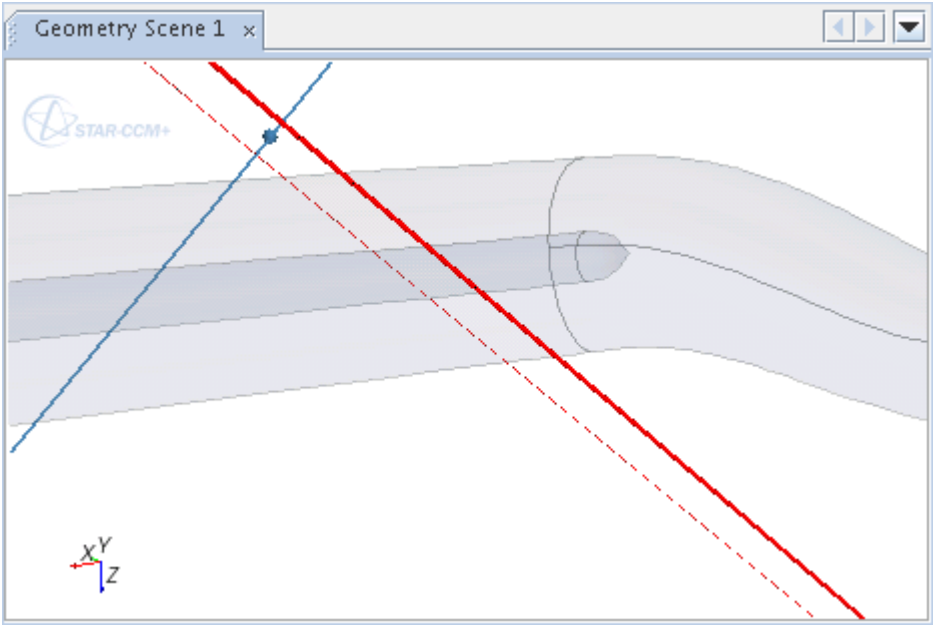
Alternatively, you can define the plane with two mouse clicks in the display. First, position the object in the scene so that your view is in the direction of the plane you want to define. Click the button to the left of the coordinate text boxes.



Click once in the display for the first point, and then click the second point to draw the line defining the plane.



This action repositions the plane tool.

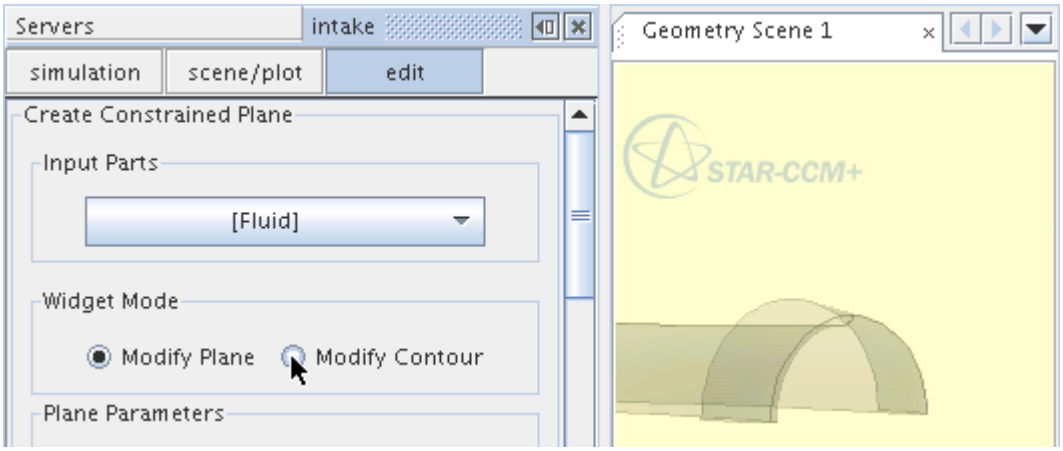


Using the Bounded Implicit Plane Tool

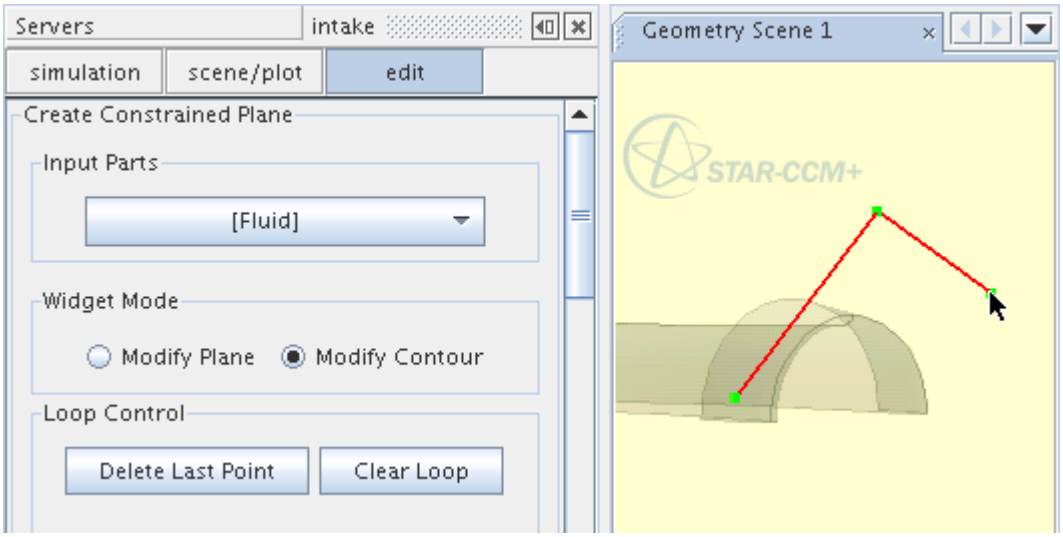
This tool lets you create a loop that would limit the extent of the implicit plane.

This device gives you greater control of the scope of data that you sample from the plane section. Before using this tool, finish using the implicit plane tool.

Select the **Modify Contour** button to activate the loop feature.



To begin creating the loop, click once in the plane tool to define the first point. Each subsequent click in the plane draws a line connecting your last point with the new one, drawing the bounding loop.

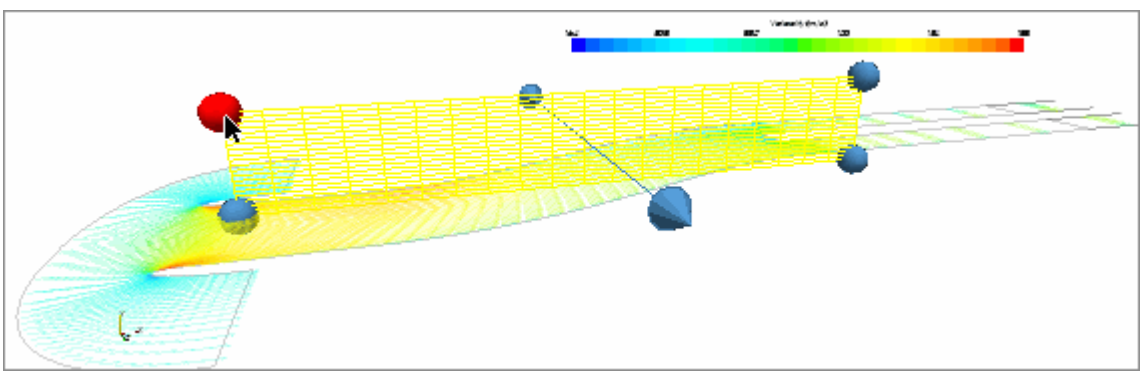


To close the loop, click the first point. You can reverse your steps by using the **Delete Last Point** button, or you can start over by clicking the **Clear Loop** button.

Using the Explicit Plane Tool

This three-dimensional tool defines a finite (bounded) plane that can be interactively placed in a scene.

The plane has four handles (at its corner vertices), a normal vector, and the plane itself. The handles are used to resize the plane; the normal vector to rotate it, and the plane can be picked and translated.



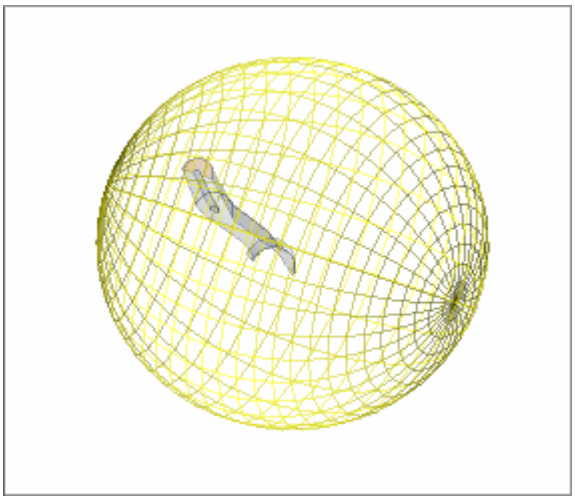
- If you click the plane and drag it while holding down the `<Ctrl>` key, it rotates around the normal.
- To resize the plane, click and drag one of the four handles on the corners.
- If you select the normal vector, you can rotate the plane arbitrarily.
- You can translate the plane along its normal by clicking and dragging it with the middle mouse button.
- To scale the plane from its center, click and drag with the right mouse button up or down. The upward direction makes the plane bigger, while the opposite direction makes it smaller.
- You can constrain the translation of the plane by ticking the **Snap To Part** option in the *Plane Parameters* group box, and selecting the part in the drop-down menu. This option makes it easy to align the plane with a certain part. (This action is essentially snapping to the bounding box of the part.)

- To refine the position of the plane, you can adjust the coordinates with text entries. After each text entry, press `<Enter>` and click `Update Tool` to apply your adjustment.
- The plane can be picked even when it is “behind” other actors. It is not a bug: this feature is intended.

Using the Sphere Tool

This three-dimensional tool defines a sphere that can be interactively placed in a scene.

You can click and drag the sphere to change its position in the scene.

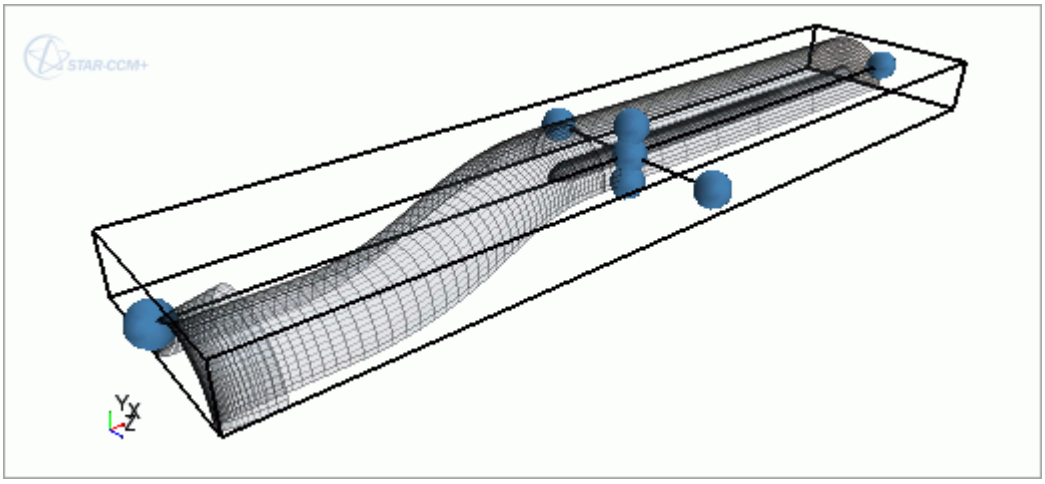


- To resize the sphere, you can change the radius with a text entry. After the entry, press `<Enter>` to apply your adjustment.
- To refine the position of the sphere, you can adjust the coordinates of the origin with text entries. After each text entry, press `<Enter>` to apply your adjustment.
- You can constrain the translation of the sphere by ticking the **Snap to Part** option in the *Sphere Parameters* group box, and selecting the part in the drop-down menu. This option makes it easy to align the sphere with a certain part. (The sphere is snapped to the bounding box of the part.)

Using the Resampled Volume Tool

This three-dimensional tool defines a resampled volume that can be interactively placed in a scene.

To control the size and position of the resampled volume, use the handles and the bounding box.



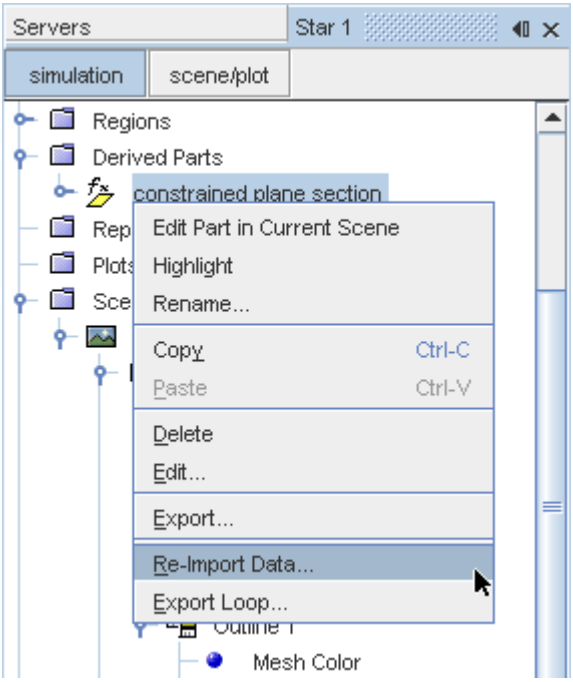
- To resize the resampled volume, click and drag the handle at any of the sides of the bounding box.
- To rotate the resampled volume, click and drag the outline of the bounding box.
- To translate the resampled volume, click and drag the handle at the center of the bounding box.
- To constrain the translation of the resampled volume, activate the **Snap to Part** option and select a part. This option makes it easy to align the resampled volume with a certain part. (The resampled volume is snapped to the bounding box of the part.)

Reading an External File

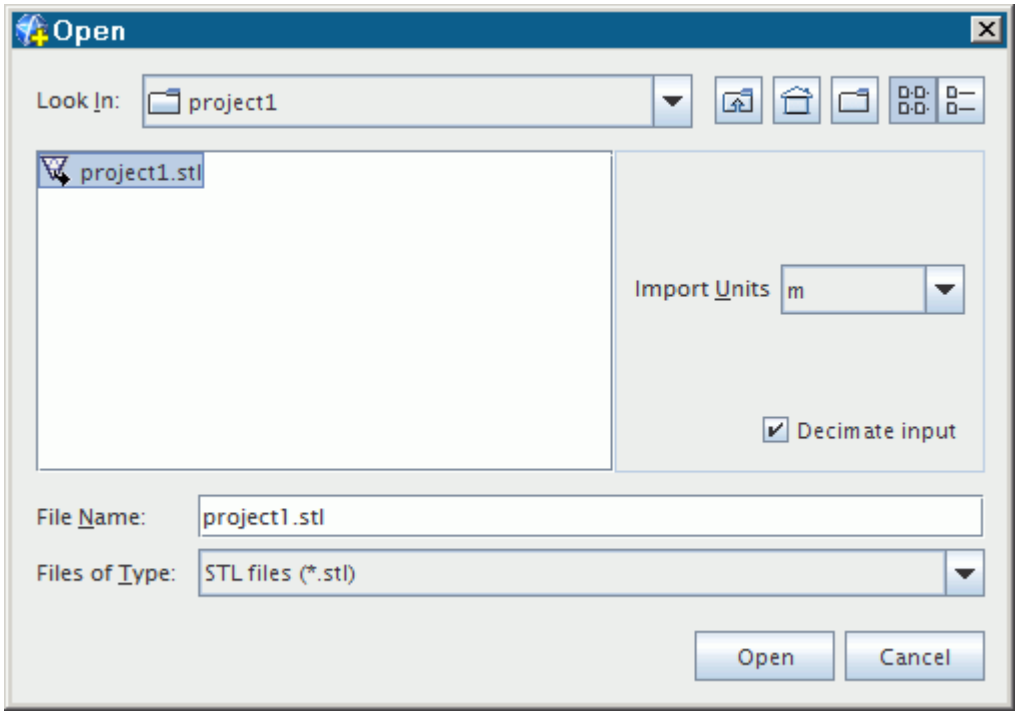
An arbitrary section, arbitrary probe, or constrained plane section can read data from an external file (STL or VTK).

Note: Before importing the data for a constrained plane section, create an arbitrary constrained planar section.

1. To access this file, right-click the node of the derived part and select **Re-Import Data...**



2. Select an available file from the *Open* dialog, then click **Open**. This action adds the data from the specified file.



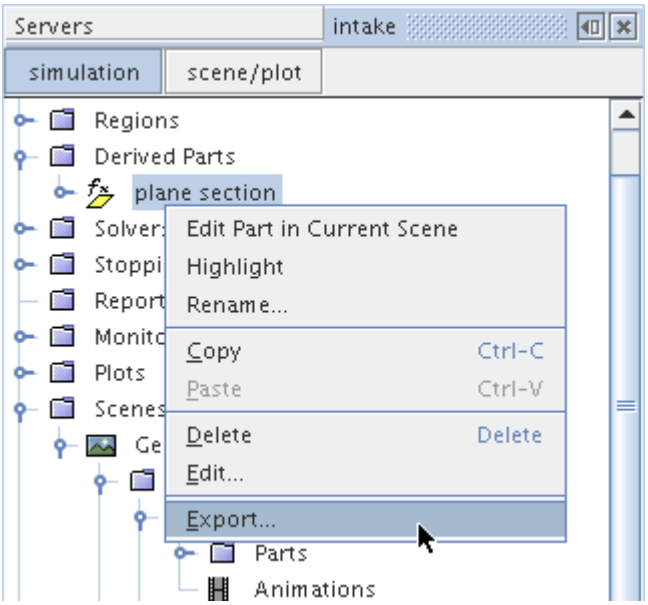
The **Decimate input** checkbox, which is activated by default, facilitates creation of an arbitrary section.

Exporting to an External File

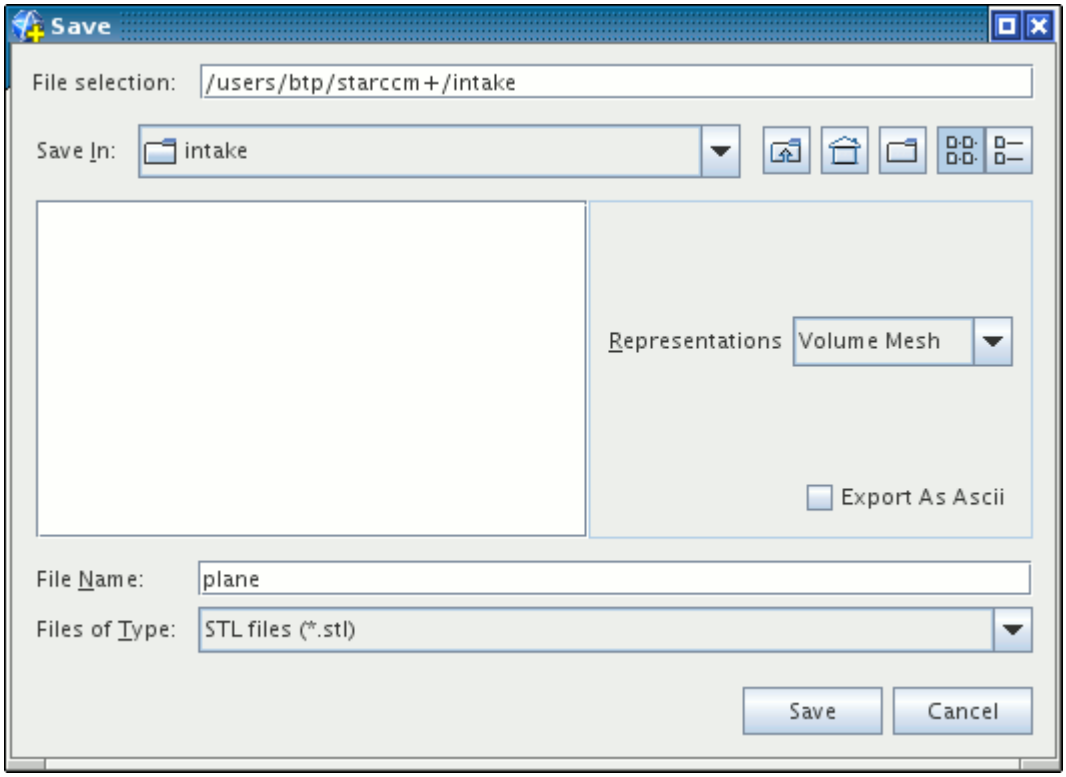
You can export the data of derived parts to an external file in STL or VTK format.

Both file formats can be binary or ASCII, and STL files are a universal method of transferring surface data using the least possible number of triangles.

1. To export the data of a derived part, right-click the node of the part in the object tree and select **Export...**

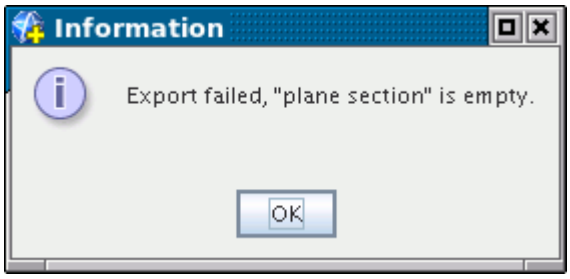


2. In the *Save* dialog, enter a name for the `.stl` or `.vtk` file and ensure that the correct representation is selected in the **Representations** menu. By default, the file is saved in binary format. To save the file in ASCII format, tick the **Export as Ascii** checkbox.



For a constrained plane section, arbitrary section or arbitrary probe, the file that this action generates is different from the imported file, as only the extracted triangle data is exported. When streamlines are exported as an STL or VTK file, they are defined as thin ribbons.

The derived part must have one or more input parts, otherwise STAR-CCM+ considers it empty. An attempt to export it fails with a message like this one:

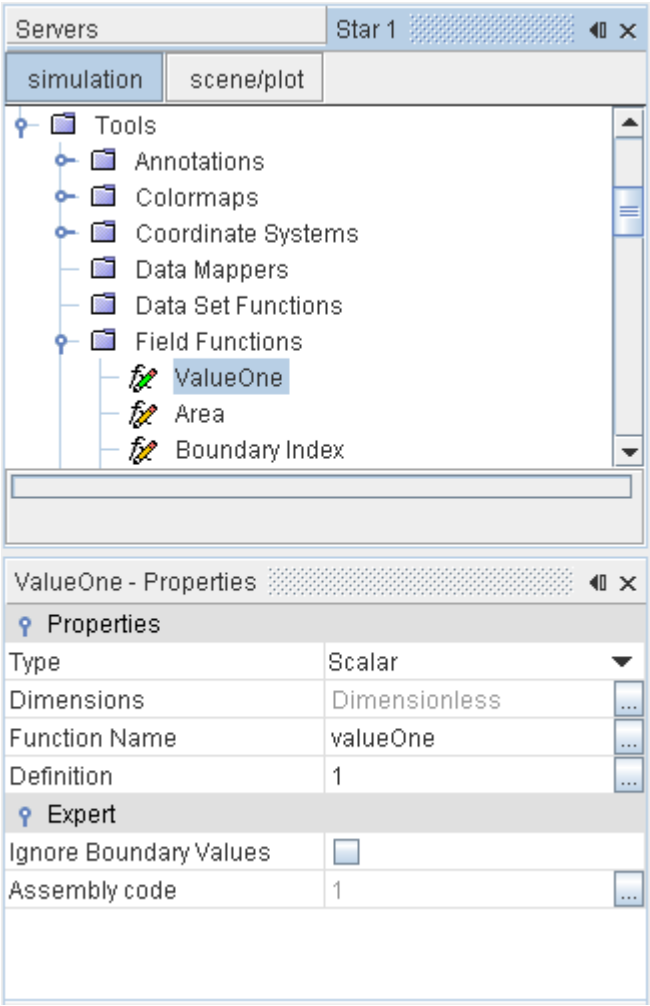


Calculating the Surface Area of a Part

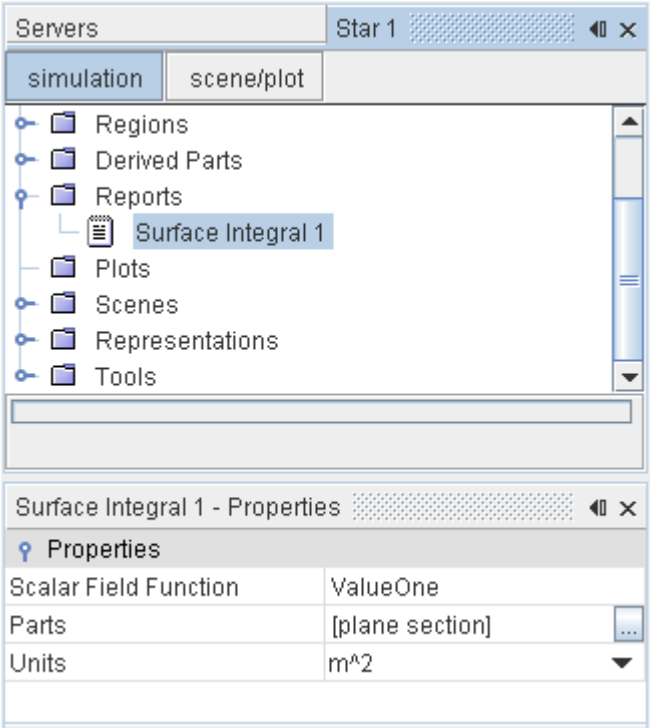
When a volume representation is active, the surface area of a part such as a plane section, a cylinder surface, a sphere, or even an isosurface can be calculated using a surface integral report with a value of 1.

To calculate the surface area:

1. Create a user-defined scalar field function with the constant value of 1. Such a function is shown below.



2. Create a surface integral report using the right-click menu of the top-level Report node. Set the *Parts* property of the new report node to the part whose surface area you require, and set the *Scalar Field Function* property to the user-defined field function created in the previous step. An example of such a report is shown below.



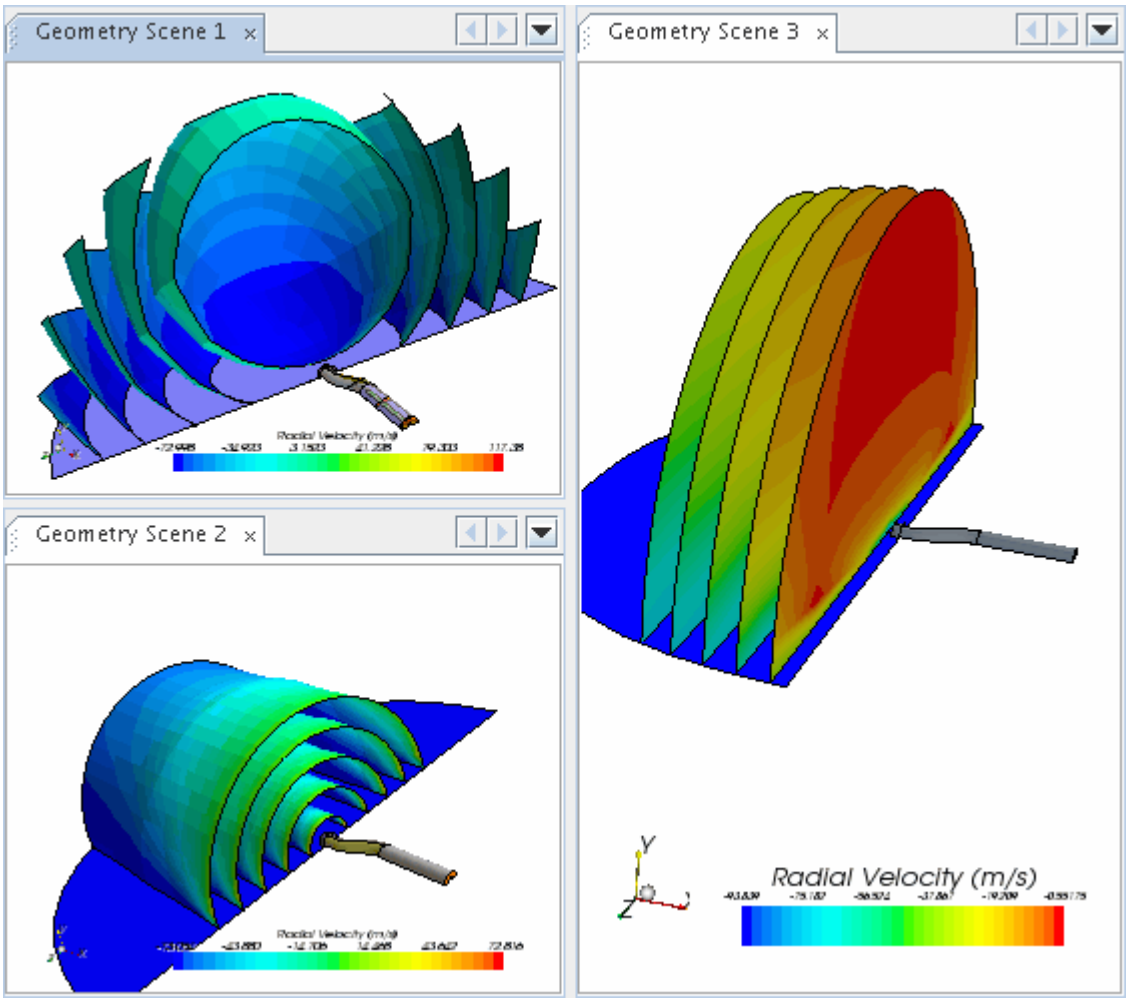
3. Right-click the newly created report node and choose **Run Report**. The surface area of the part is displayed in the *Output* window.

Defining Implicit Sections

Implicit sections are derived parts that are based on planes or geometrical shapes, which can cut through one or more volumes or surfaces.

These parts are implicit because they are functions of coordinates, rather than combinations of mesh cells.

Within an implicit section part, you can create multiple instances of the section by defining offset copies of the original. This feature gives you the benefit of sampling data from multiple sections while you still work with just one part.



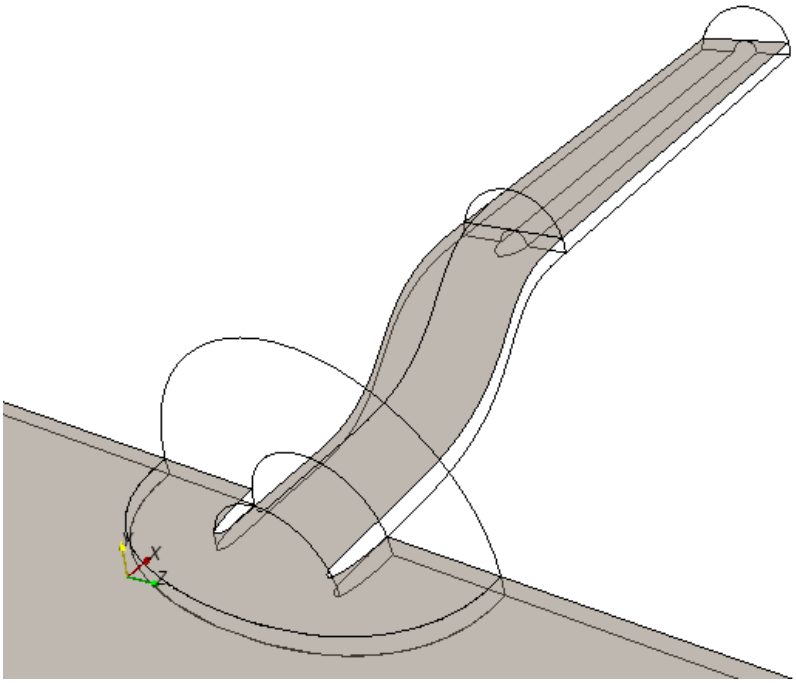
Contents:

Defining a Plane Section
Defining a Constrained Plane Section
Defining a Cylinder Section
Defining a Sphere Section
Defining an Arbitrary Section

Defining a Plane Section

The plane section is an implicit part that uses a plane as the implicit function to cut. STAR-CCM+ allows the creation of a planar section from parts comprising regions, boundaries, and geometry parts, however input parts are enforced to be of the same dimensionality.

The resulting output is dependent on the topology and discretization of the input parts. The exact form of output depends on the dimensionality of the input parts. Input data consisting of three-dimensional cells result in output surfaces, input data consisting of two-dimensional cells result in output lines, and input data consisting of one-dimensional data result in points.

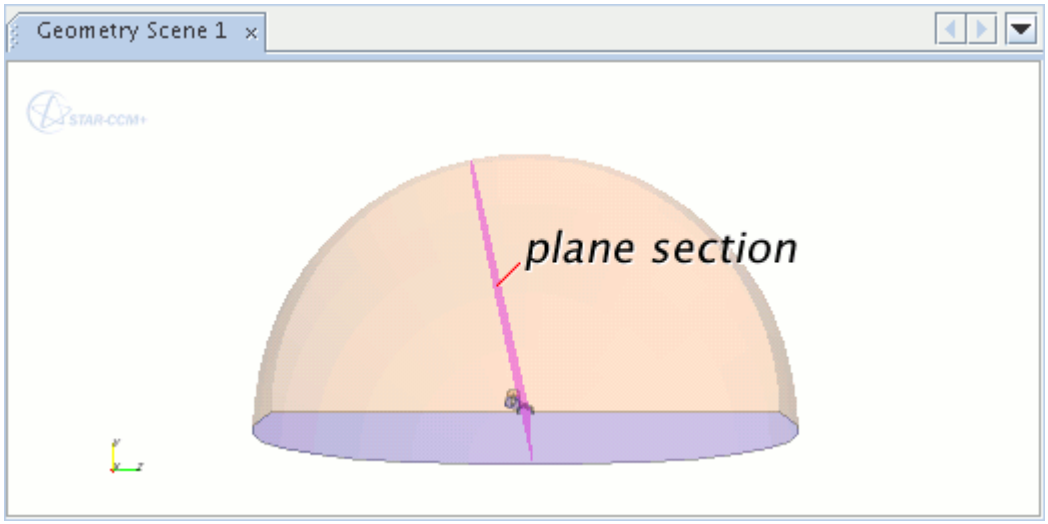


1. Create a plane section using one of the following methods:
   - Follow the general procedure outlined in the section, Defining Derived Parts General Workflow. Choose the menu option, **New Part** > **Section** > **Plane...**.
   - Use the toolbar button, , to create a plane section in an open scene quickly. To use this method:
     1. Position the image in the scene so that your view is exactly perpendicular to the plane.

        The plane is defined with two points that you specify.
     2. Click (**Create Plane Section**), then click at two points in the display to define the plane.
     3. In the *Select Displayer* dialog that appears, select a displayer option.

        You can choose one of the existing displayers, create a new one, or create the derived part without assigning it to any displayer.

        The plane section appears in the scene. By default, it has the appropriate input parts from among the ones that are visible in the scene. You can edit this derived part to change the input parts or other settings.



You can define the section properties in the derived part in-place dialog or *Properties* window, as described in Defining Derived Parts General Workflow. For plane sections, the visual plane tool is also available. You can use the plane tool, which is activated automatically with the in-place dialog, to position the plane visually in the scene. See Using the Implicit Plane Tool.

To completely define the plane section:

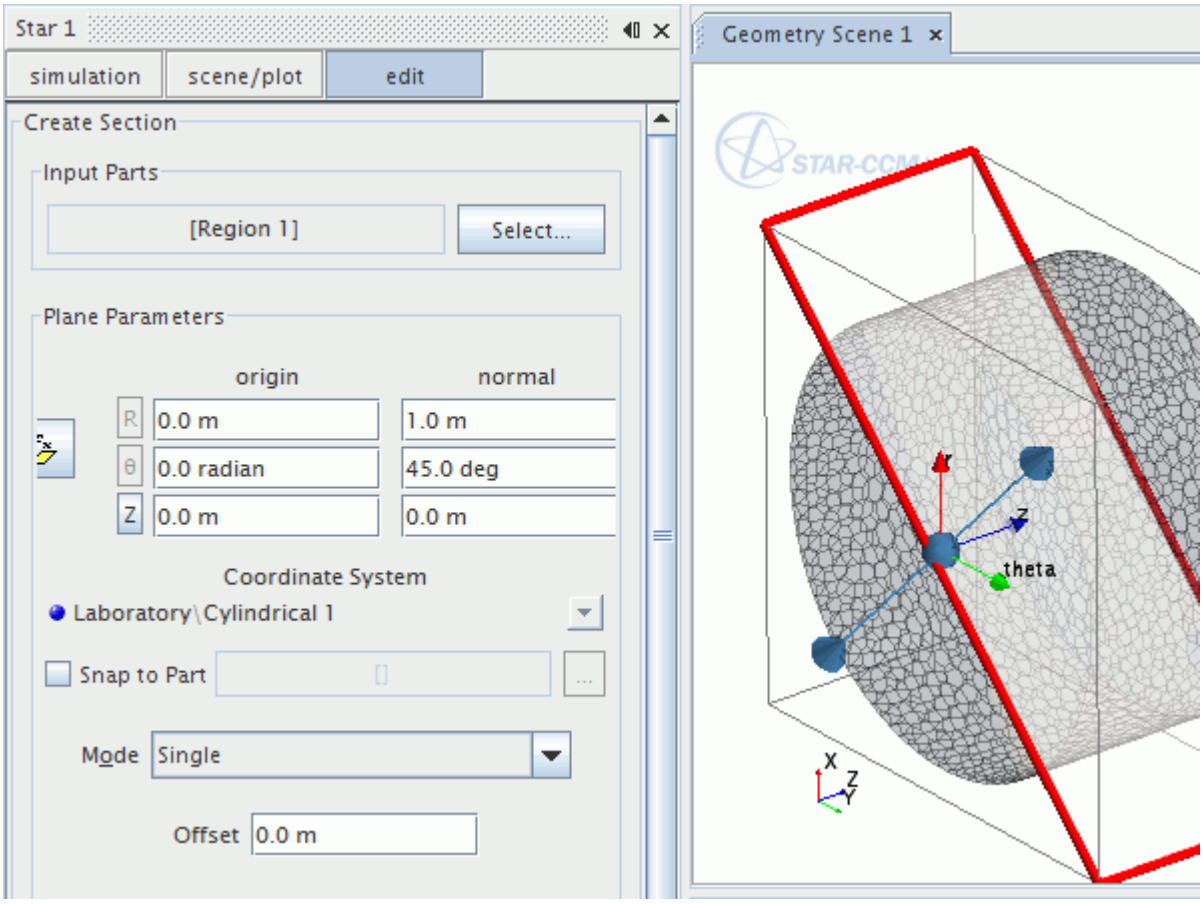
2. Specify the general properties (**Input Parts**, **Display option**, and **Coordinate System**) as described in the section, Defining Derived Parts General Workflow.
3. Complete the plane section definition by specifying the following properties:
   - Origin — set the x, y, and z coordinates for the origin. You can include units in the entry.
   - Normal — set the x, y, and z coordinates for a point that is used to define the surface normal. The normal is defined with the vector between the origin of the selected coordinate system and this point. You can include units in the entry.
   - Mode — specify single or multiple sections.

   For more information on the available properties, see Implicit Section Properties.

Example of Creating a Plane Section with a Cylindrical Coordinate System

If you create a plane section with a cylindrical coordinate system, specify r, θ, and z coordinates. The resulting normal vector consists of the r-direction (the cylinder radius), the z-direction (the cylinder length), and θ, the angle of rotation about the z-axis. The plane section is perpendicular to the normal.

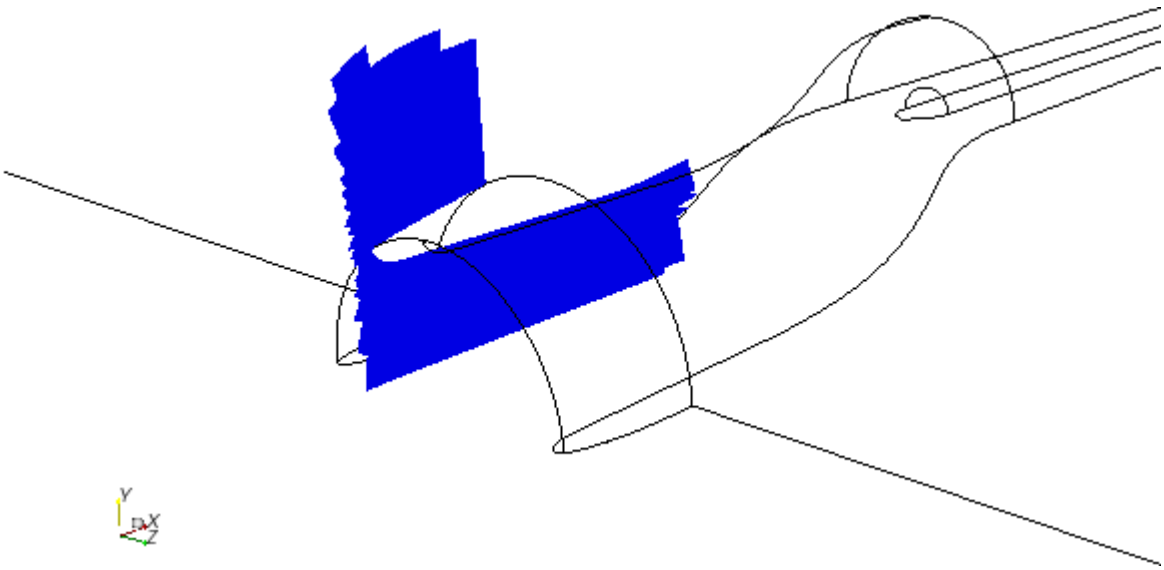
In the example illustrated below, the normal is 1 m long, with values of 45 degrees for θ and 0 for z, so the normal lies in the r-θ plane with the plane section being perpendicular to it.



Defining a Constrained Plane Section

A constrained plane section is similar to a plane section, except that it is bounded in a loop (a set of coordinates that you specify). Also, a constrained plane section can read shape data from an external file.

A volume mesh must be present for the plane to appear. The edges of the constrained plane snap to the nearest whole cell, as shown below.



An arbitrary constrained planar section must be created first, before an external file can be read. The external file must be in .stl or .vtk file format, and contain a surface that is used as the constrained planar section. The loop of the constrained plane consists of the perimeter edges of the imported surface. If the surface you are importing was created in a commercial CAD package, adjust its position and orientation as needed using the in-place dialog. This task is done by editing the constrained plane. Ensure that the units of the exported `.stl` file and the STAR-CCM+ file match.

To create a constrained plane section:

1. Follow the general procedure outlined in the section, Defining Derived Parts General Workflow. Choose the menu option, **New Part** > **Section** > **Constrained Plane...**.
2. Define the constrained planar section using one of the following methods:
   - Read in an external file as described in the section, Reading an External File.

     The external file must be in .stl or .vtk file format, and contain a surface that is used as the constrained planar section. The loop of the constrained plane consists of the perimeter edges of the imported surface. If the surface you are importing was created in a commercial CAD package, adjust its position and orientation as needed using the in-place dialog. This task is done by editing the constrained plane in an active scene. Ensure that the units of the exported `.stl` file and the STAR-CCM+ file match.
   - Set the constrained plane section properties and define the loop:
     1. Specify the input parts, display option, and coordinate system as described in the section, Defining Derived Parts General Workflow.
     2. Specify the plane parameters (that is, origin, normal, and mode) as described in the section, Defining a Plane Section.
     3. Define the limits of the constrained plane section using the Bounded Implicit Plane Tool. See Using the Bounded Implicit Plane Tool.

        If you didn't create the constrained plane section using the in-place dialog, you can bring up the dialog by editing the constrained plane in an active scene. See Defining Derived Parts General Workflow.

   For more information on the available properties, see Implicit Section Properties.

The loops that you create in STAR-CCM+ can be exported to external files. To export a loop:

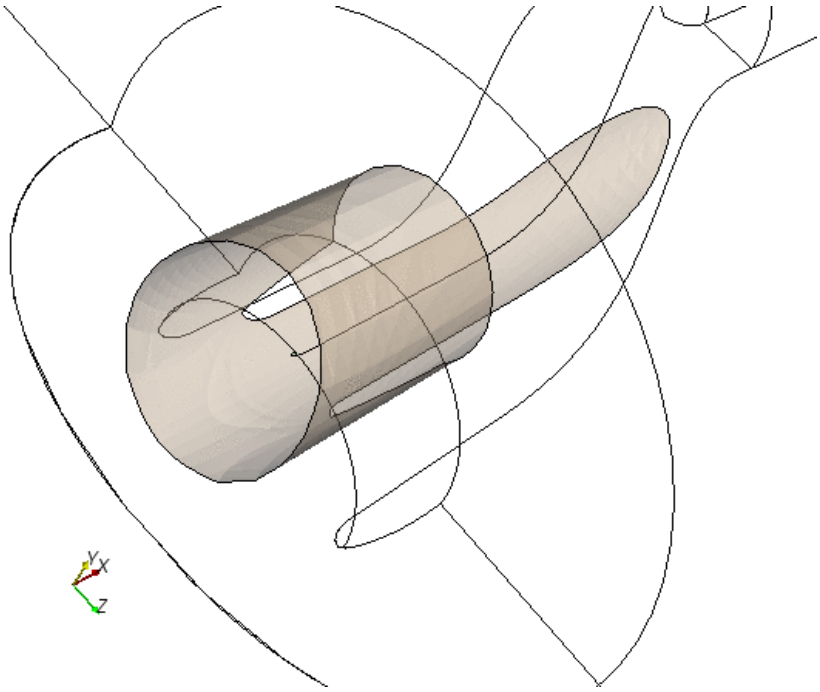
3. Right-click the **Derived Parts** > **[constrained plane section]** node and select **Export Loop...**

   The loop is saved in `.stl` format.

Defining a Cylinder Section

A cylinder section uses a cylinder as the implicit function to cut. STAR-CCM+ allows the creation of a cylinder section from parts comprising regions, boundaries, and other parts, provided all input parts are of the same dimensionality.

The exact form of output depends on the dimensionality of the input parts. Input data consisting of three-dimensional cells result in output surfaces, input data consisting of two-dimensional cells result in output lines, and input data consisting of one-dimensional data result in points.



To create a cylinder section:

1. Follow the general procedure outlined in the section, Defining Derived Parts General Workflow. Choose the menu option, **New Part** > **Section** > **Cylinder...**.
2. Specify the input parts, display option, and coordinate system as described in the section, Defining Derived Parts General Workflow.
3. Complete the cylinder section definition by specifying the following properties:
   - Origin — set x, y, and z coordinates for the origin of a rotation axis through the center of the cylinder. You can include units in the entry.
   - Orientation — set x, y, and z coordinates for a point that is used to define the central axis of the cylinder. The axis is defined by the vector between the origin of the selected coordinate system and this point. You can include units in the entry.

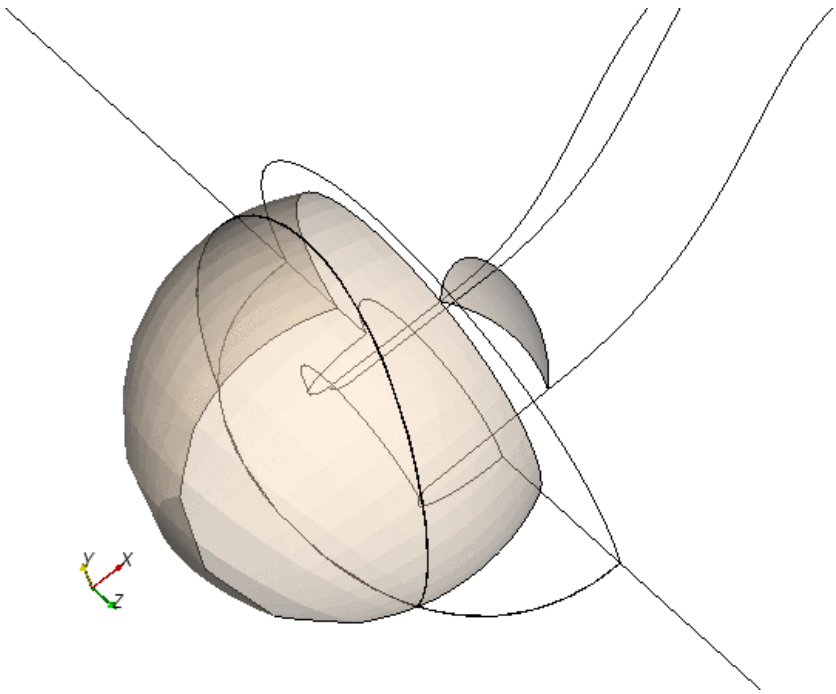
- Radius — specify the distance in a perpendicular direction from the center axis, defining the outer wall of the cylinder. You can include units in the entry.
- Mode — specify single or multiple sections.

For more information on the available properties, see Implicit Section Properties.

Defining a Sphere Section

A sphere section uses a sphere as the implicit function to cut. STAR-CCM+ allows the creation of a sphere section from parts comprising regions, boundaries, and other parts.

The exact form of output depends on the dimensionality of the input parts. Input data consisting of three-dimensional cells result in output surfaces, input data consisting of two-dimensional cells result in output lines, and input data consisting of one-dimensional data result in points. Input parts are enforced to be of the same dimensionality.



To create a sphere section:

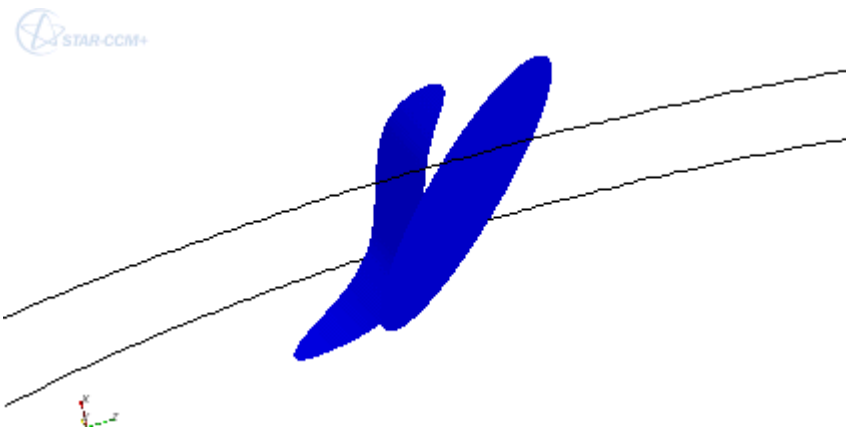
1. Follow the general procedure outlined in the section, Defining Derived Parts General Workflow. Choose the menu option, **New Part** > **Section** > **Sphere...**.
2. Specify the input parts, display option, and coordinate system as described in the section, Defining Derived Parts General Workflow.
3. Complete the sphere section definition by specifying the following properties:
   - Sphere origin — set x, y, and z coordinates for the sphere origin. You can include units in the entry.
   - Sphere radius—set the distance from the sphere origin, defining the outer wall of the sphere. You can include units in the entry.
   - Mode — specifies single or multiple sections.

For more information on the available properties, see Implicit Section Properties.

Defining an Arbitrary Section

The arbitrary section is an implicit part that samples one or more input parts or regions. The geometry of an arbitrary section can be created from either a CAD part surface (Root description only) or directly from a STL file. Arbitrary sections should be used when a nonplanar sampling surface is desired.

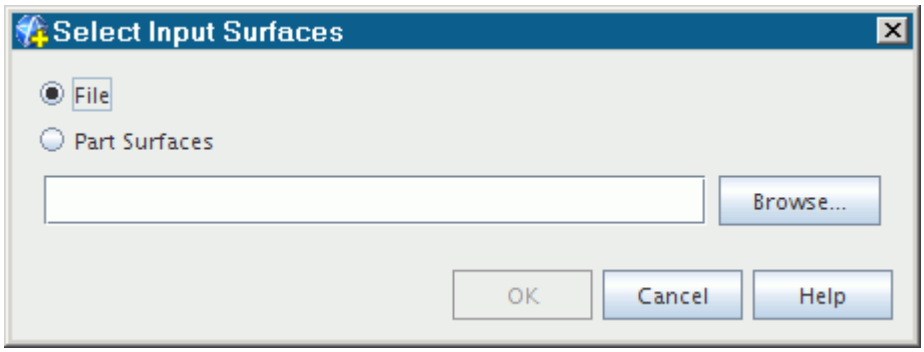
This representation of the input geometry is mesh-resolution-dependent. A lower polygon count in an arbitrary section results in a faster intersection computation between the section and the sampled region.



To create an arbitrary section:

1. Follow the general procedure outlined in the section, Defining Derived Parts General Workflow. Choose the menu option, **New Part** > **Section** > **Arbitrary...**

   If the conditions for activating the in-place dialog are not met, the *Select Input Surfaces* dialog appears.



To define the limits of the arbitrary section, get shape data from part surfaces or an external file:

2. Specify the part surfaces or import an STL file using one of the following methods:
   - To import an STL file, activate **File** in either the *Select Input Surfaces* dialog, or in the *Surface Input Options* group box in the in-place dialog, then click **Browse...** and use the *Open* dialog to select the `.stl` file to import.

This action adds the data from the specified file. During the import, STAR-CCM+ retriangulates the surface to reduce the number of triangles. The resulting coarser resolution can reduce processing time significantly. However, this retriangulation can also cause problems for surfaces that have curves or other complex characteristics. To disable this feature and retain the resolution of the input file, clear the **Decimate input** checkbox in the dialog (which is activated by default). Since the surface decimation algorithm is based on the local curvature, planar sections of the surface will not be retriangulated.

- To use part surfaces, activate **Part Surfaces** in either the *Select Input Surfaces* dialog, or in the *Surface Input Options* group box in the in-place dialog, then click **Select...** and use the *Select Objects* dialog to specify the part surfaces.

  This action adds the data from the specified part surfaces. For parts that have multiple descriptions, STAR-CCM+ chooses surfaces that belong in the Root description only.

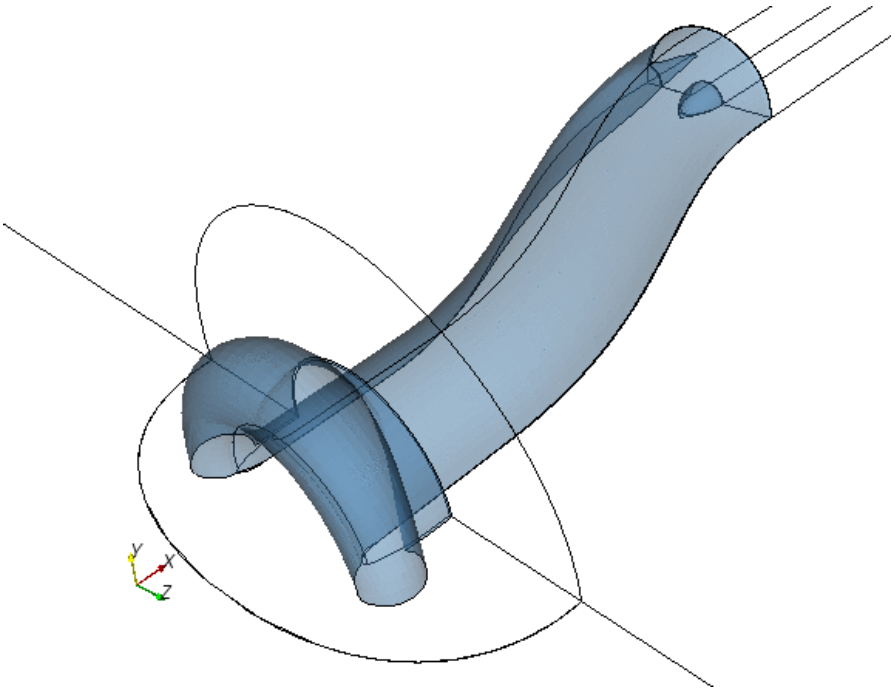
3. Specify the input parts and display option as described in the section, Defining Derived Parts General Workflow.

   For more information on the available properties, see Implicit Section Properties.

Defining an Isosurface

An isosurface is a surface representing a constant-value scalar function. STAR-CCM+ allows the creation of an isosurface from parts comprising regions, boundaries, and other parts, however input parts are enforced to be of the same dimensionality.

The exact form of the output of the isosurface depends on the dimensionality of the input parts. Input data consisting of three-dimensional cells result in output isosurfaces, input data consisting of two-dimensional cells result in iso lines, and input data consisting of one-dimensional data result in points.



To create an isosurface:

1. Follow the general procedure outlined in the section, Defining Derived Parts General Workflow. Choose the menu option, **New Part** > **Isosurface...**.
2. Specify the input parts and display option as described in the section, Defining Derived Parts General Workflow.
3. Complete the isosurface definition by specifying the following properties:
   - Scalar field — select the scalar field function as the basis for the isosurface. The list of available scalar field functions depends on the physics models that you selected.

     You can specify the scalar field in the *Scalar Field* property, or the *Scalar* group box in the in-place dialog.
   - Mode — specify a single or multiple isosurfaces.

     You can specify the mode in the *Mode* property, or the *Extraction Mode* box in the in-place dialog.
   - Isovalues — specify a single scalar value for a single isosurface, a range of scalar values for multiple (range) isosurfaces, or a list of scalar values for multiple (arbitrary) isosurfaces.

You can specify the scalar values in the in-place dialog, or in the properties of the isosurface child nodes. Click **Query** to select a scalar range automatically.

- Number of isosurfaces — For multiple isosurfaces, specify the number of isosurfaces that you want to define.

For more information on the available properties, see Isosurface Properties.

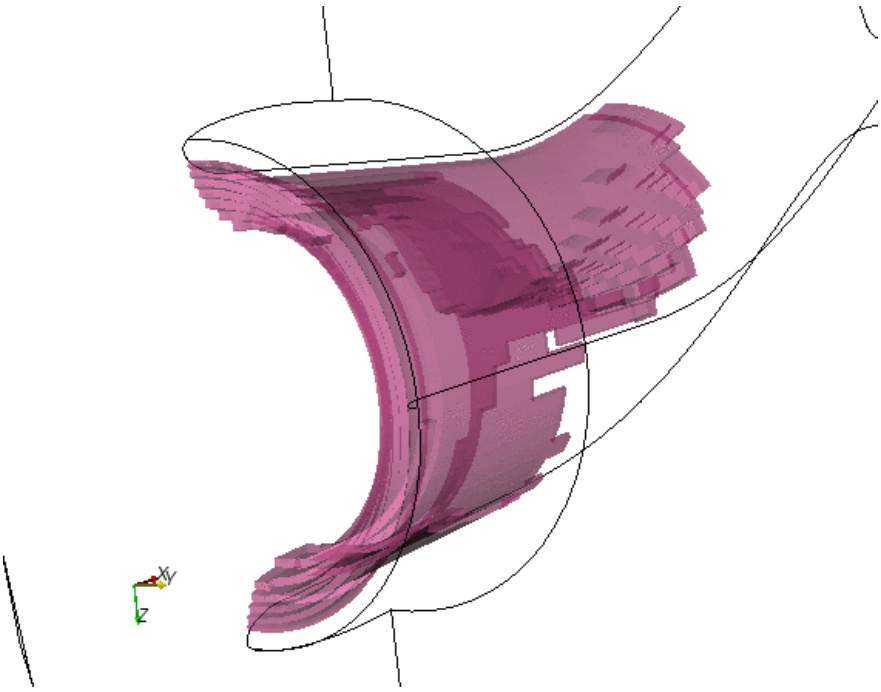
Once you have finished the creation process, rotate and zoom in the display to get a clear view of the isosurface. If you did not assign the part to the current scene, you can drag and drop its node into the visualization display. You can show a smooth surface by activating the *Smooth Shade* property of the geometry displayer.

Defining a Threshold

The threshold part isolates a group of cells based on their relationship to specified scalar values. STAR-CCM+ allows the creation of a threshold part from one or more parent parts, provided the set of parent parts are of the same dimensionality.

When created from a region, a threshold represents the boundaries of the cells containing data corresponding to the threshold criterion. When created from a boundary, it shows that boundary clipped to the specified threshold criterion.

A threshold part can also use a cell set as a scalar value. One application for this feature is a coolant drain problem, in which several pockets of coolant are left behind in the water jacket. A cell set threshold can help you determine the amount of fluid that is left in each pocket.



To create a threshold:

1. Follow the general procedure outlined in the section, Defining Derived Parts General Workflow. Choose the menu option, **New Part** > **Threshold...**.
2. Specify the input parts and display option as described in the section, Defining Derived Parts General Workflow.
3. Complete the threshold definition by specifying the following properties:
   - Scalar field — select the scalar field function as the basis for the threshold. The list of available scalar field functions depends on the physics models that you selected.

     You can specify the scalar field using the *Scalar Field* property, or the *Scalar* group box in the in-place dialog.
   - Range — specify a range of scalar values to use in combination with the extraction mode.

You can specify the range using the *Range* property, or the *Scalar Range* box in the in-place dialog. You can enter a number, or click **Query** to instruct STAR-CCM+ to probe the selected scalar and fill in the values automatically.

- Mode — specify whether to choose cells where the specified field has values within or outside the specified range, above the maximum value in the range, or below the minimum value in the range.

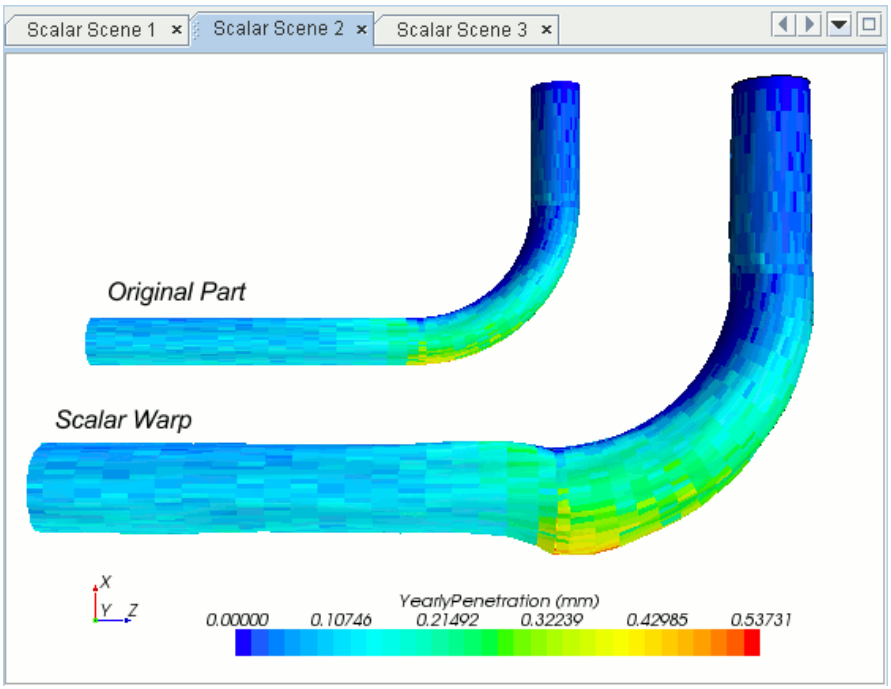
  You can specify the mode in the *Mode* property, or the *Extraction Mode* box in the in-place dialog.

For more information on the available properties, see Threshold Properties.

Defining a Scalar Warp Surface

The scalar warp surface moves the point coordinates on the surface of its input parts along the point normal vectors by an amount equal to the scalar value times the scale factor.

This feature is useful for showing flow profiles, mechanical deformation, or other desired user-defined functions. An example of the latter is shown below, which corresponds to visualizing an approximation of material loss due to particle erosion in a bend.



To create a scalar warp:

1. Follow the general procedure outlined in the section, Defining Derived Parts General Workflow. Choose the menu option, **New Part** > **Warp** > **Scalar Warp...**.
2. Specify the input parts and display option as described in the section, Defining Derived Parts General Workflow.
3. Complete the scalar warp definition by specifying the following properties:
   - Scalar field — select the scalar field function as the basis for the warp surface. The list of available scalar field functions depends on the physics models that you selected.

     You can specify the scalar field using the *Scalar Field* property, or the *Scalar* group box in the in-place dialog.
   - Scale Factor — specify the factor by which the scalar field is multiplied.

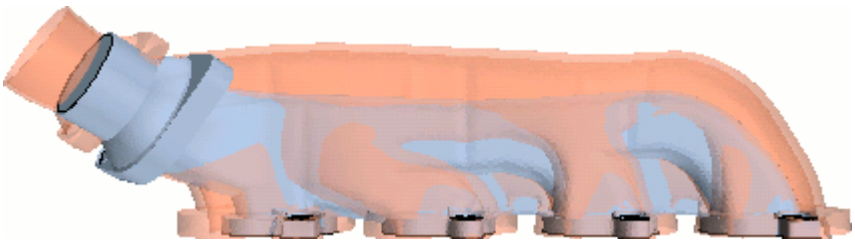
     You can specify the *Scale Factor* in the in-place dialog or the *Properties* window.

   For more information on the available properties, see Scalar Warp Properties.

Defining a Vector Warp Surface

A vector warp surface is a part that modifies input surface point coordinates by moving points along the selected vector multiplied by the scale factor.

This feature is useful for showing flow profiles or mechanical deformation.



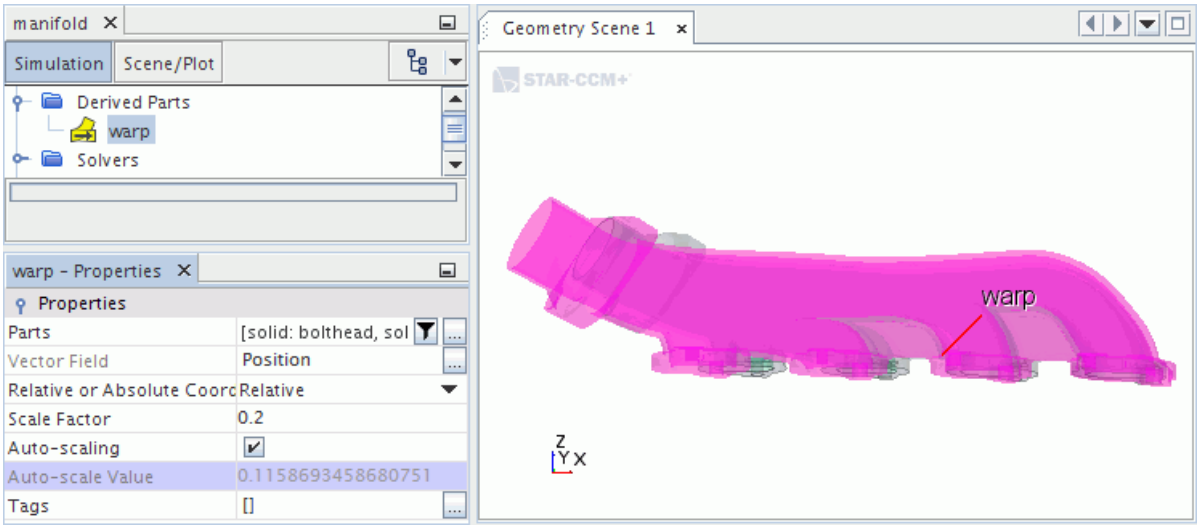
To create a vector warp surface:

1. Follow the general procedure that is outlined in the section, Defining Derived Parts General Workflow. Choose the menu option, **New Part** > **Warp** > **Vector Warp...**.
2. Specify the input parts and display option as described in the section, Defining Derived Parts General Workflow.
3. Complete the vector warp definition by specifying the following properties:
   - *Vector Field* — select the vector field function as the basis for the warp surface. The list of available vector field functions depends on the physics models that you selected.

     You can specify the vector field using the *Vector Field* property, or the *Vector* group box in the in-place dialog.
   - *Scale Factor* or *Auto-scaling* — control the factor by which the field is multiplied. These settings are available in the in-place dialog or the *Properties* window.
     - To specify the scale factor manually, enter the fraction in the *Scale Factor* property.
     - To have STAR-CCM+ calculate the scale factor automatically, activate the *Auto-scaling* property.

       When you activate *Auto-scaling*, the numerical value of the scale factor is displayed in the *Auto-scale Value* property.

     When you select the node of the warp surface with a scene open, the effect of the scale factor is displayed in a highlighted image of the derived part. The distortion that you see depends on whether you are using the auto-scale value or the manually specified scale factor.



For more information on the available properties, see Vector Warp Properties.

Defining a Streamline

A streamline part represents particle paths, computed from a vector field. STAR-CCM+ offers two types of streamline parts: streamlines and constrained streamlines. Constrained streamlines are particle paths that are constrained to lie in a surface, and can be used to simulate effects such as oil streaks.

Streamline parts may be computed from any vector field, although it generally makes the most physical sense to construct streamline parts from a velocity field. You can create a standard streamline from input parts consisting of regions, or other parts which define a volume such as a threshold part. Constrained streamlines require surface input parts such as boundaries, or sections. The constrained streamline results from integration of the velocity field near the surface of interest. This velocity field is modified by subtracting the dot product of the local velocity with the surface normal, and the result is projected onto the surface.

Streamlines include a seed object that defines the particle seed points. There are three types of seeds:

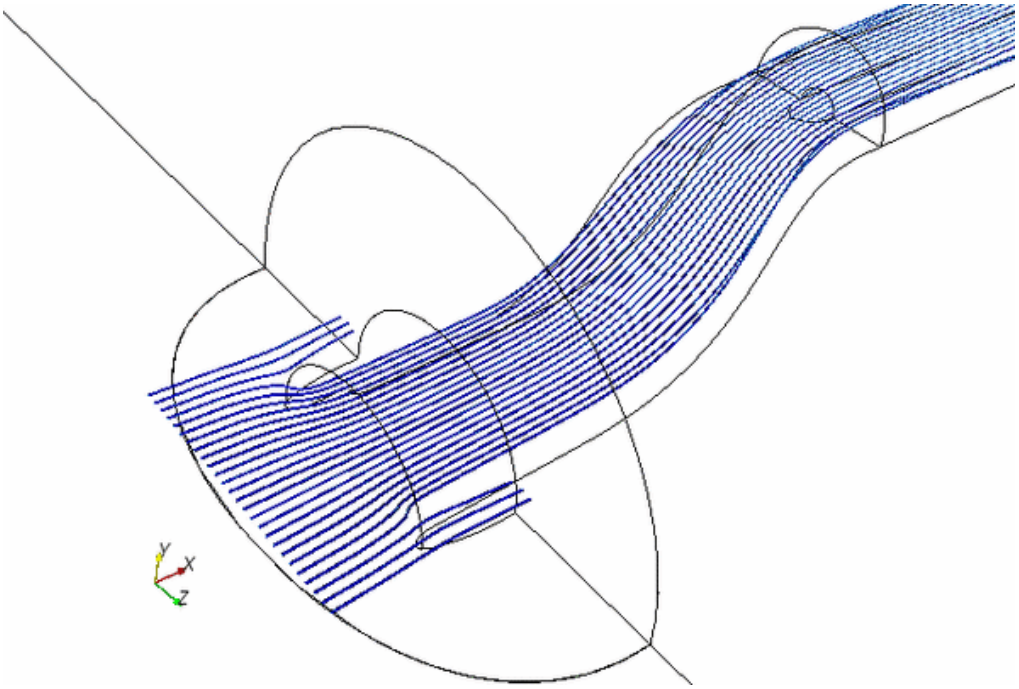
- A part seed, which takes its points from an existing part.
- A line seed, which takes its points from a line
- A point seed, which allows the definition of either a single point or a cloud of points with a given radius

Streamlines are calculated using Runge-Kutta integration, accessible through RK nodes in the object tree, and can integrate both forward and backward. The integration is controlled by the following properties:

- *Maximum Propagation*—specifies the maximum arc length of a streamline defined in STAR-CCM+ internal SI units. The streamline computation is started at an initial position p0. The next position, p1, is a small increment along the direction vector located at p0. Subsequent positions are similarly evaluated. While stepping tangential to the vector field, the distances between the positions are accumulated. This accumulated distance is the arc length of the streamline, and is the quantity that STAR-CCM+ compares to the Maximum Propagation value.
- *Max Steps*—specifies the maximum number of steps to execute in the integration.

A streamline terminates when either of these criteria are met. A streamline may also terminate because it exited the domain, or entered a stagnation region. Integration quality is controlled by the *Initial Integration Step* property which is the step-size that is used by the integration. If chosen to be much smaller than the average cell-length, the resulting streamline likely contains many more points than desired. Conversely if the step size is too large, cells may be skipped during the integration, resulting in a streamline of degraded quality.

In the following screenshot, particles are released from a line seed, and a line tool is used to assist with positioning the line endpoints.



To create a streamline:

1. Follow the general procedure outlined in the section, Defining Derived Parts General Workflow:
   - To create a standard streamline, choose the menu option, **New Part** > **Streamline...**
   - To create a constrained streamline, choose the menu option, **New Part** > **Constrained Streamline...**.
2. Specify the input parts and display option as described in the section, Defining Derived Parts General Workflow. For constrained streamlines, input parts must be surfaces.

   You can select a part along which to fix the location of the derived part (this action is essentially snapping to the bounding box of the part). To make the selector available in the in-place dialog, activate the **Snap To Part** option and click the ellipsis button at the right of the field.
3. Complete the streamline definition by specifying the following properties:
   - Vector field — set the *Vector Field* in the *Properties* window or in the in-place dialog. The list of available vector field functions depends on the physics models that you selected.
   - Seed Mode — select the type of seed and set its properties. The available options are Part Seed, Point Seed, and Line Seed. For more information, see Streamline and Constrained Streamline Properties.

     You can specify the *Seed Type* in the in-place dialog or the *Properties* window. For the line seed and point seed, the units can be changed by typing them in after the values. Coordinates are relative to the laboratory (Cartesian) coordinate system, which cannot be changed.

     When using a part seed, you can use the in-place dialog to specify the *Part U-Resolution* and *Part V-Resolution*. These properties allow you to specify the maximum number of points extracted from the part in the u- and v- directions. The u- and v- directions are part of the computation of an average plane, based on the specified parts, for the seed source to use.

     The u-direction is the vector in this plane that is most closely aligned with the global, Cartesian x-axis. If the average plane is an x-y plane (perpendicular to the z-axis), u corresponds exactly to

global x. If the average plane is skewed from x-y, the u-axis points toward the global x-axis. The y-axis is then the cross-product of the computed x-axis and the normal. For the x-y plane, this y-axis is exactly in the y-direction.

Once the average plane for the seed source is computed, the plane size is determined from the projection of the seed source onto the average plane. Suitable u- and v- directions are computed, then the plane is divided into the specified number of points along the u- and v- directions. The points are then projected back onto the boundary and used as the seed points for the part.

Note: This feature is only designed to work with planar or reasonably planar boundaries. Therefore, if you are working with a curved surface, STAR-CCM+ attempts to project it to the average plane, but accuracy is limited. Accuracy decreases with the amount of curvature in the surface that you project.

- Integration Solver — select the Runge-Kutta integration solver and set the integration properties in the *Properties* window. Refer to the section, Streamline and Constrained Streamline Properties, for a list of available options and properties.

It is also possible to create a streamline on an arc or a circle through the following steps:

- Create a cylinder section. See Defining a Cylinder Section.
- Create a plane section to intersect with the cylinder section (by selecting the cylinder section as the input part). Depending on the angle at which the two sections intersect, the resulting plane section will be a circle, an ellipse or an arc.
- Create streamlines, setting the seed type as `Part` and selecting the plane section as the seed part.

Defining an (m',theta) Warp

In STAR-CCM+, the (m', theta) warp is used to present a flattened surface of revolution.

For turbomachinery, such surfaces are blade-to-blade. They usually originate from a cylindrical plane at constant radius, or an isosurface of the block-mapped coordinate that varies from hub to case.

This derived part requires three pieces of information in its properties:

- The *Origin* of the axis of rotation
- The vector specifying the *Rotation Axis*
- A vector specifying the *Tangential Axis*—this axis is used to compute the angular coordinate.

The m' coordinate is the arc length along the surface projected into the (z,r) plane and normalized by the local radius ([183]):

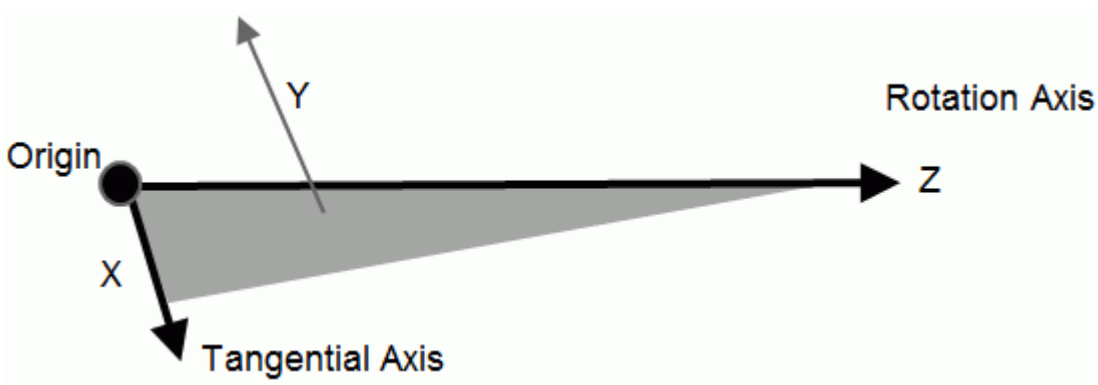
$$m' = \int \left(\frac{\sqrt{dr^2 + dz^2}}{r} \right) \tag{1206}$$

- z is the axial coordinate along the axis of rotation.
- r is the radial coordinate perpendicular to the axis of rotation.

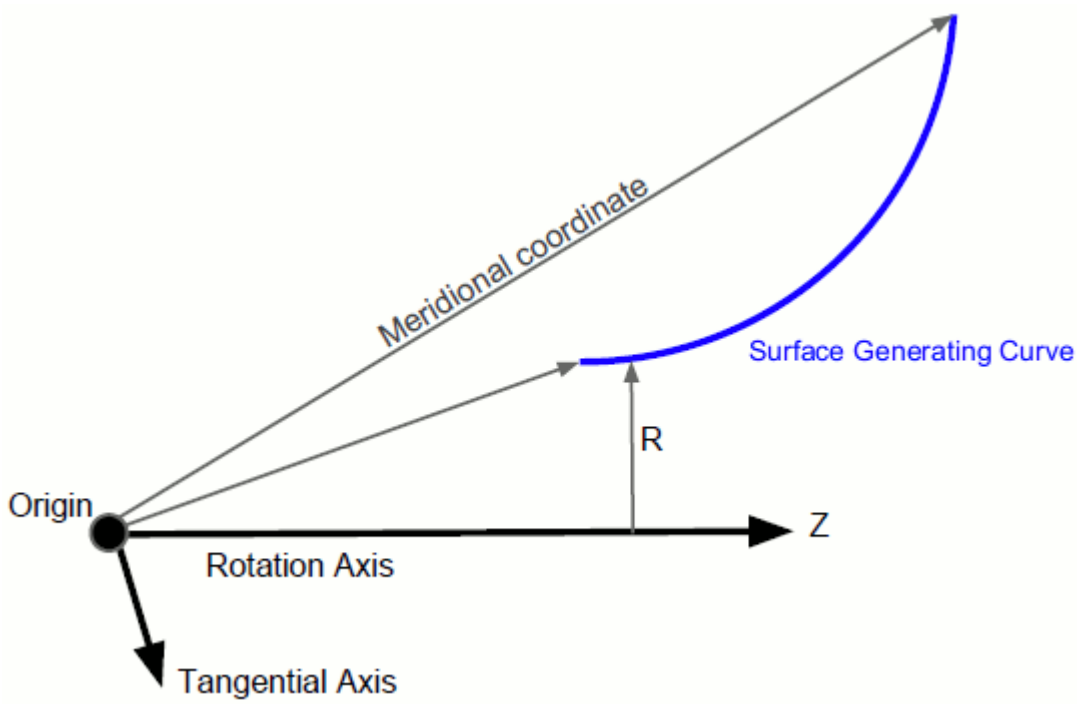
Theta is calculated as the angle from the tangential axis:

$$\theta = 2\pi + \arctan\left(\frac{-y}{-x}\right) \tag{1207}$$

where y is the distance in any direction from the plane that is formed by the tangential and rotation axes, x is the tangential axis, and z is the rotation axis.



For example, if you take the points of the surface and project them into the (z,r) plane, you can expect them to fall along the defining line of the surface (the generating curve). This result is depicted in the following diagram.

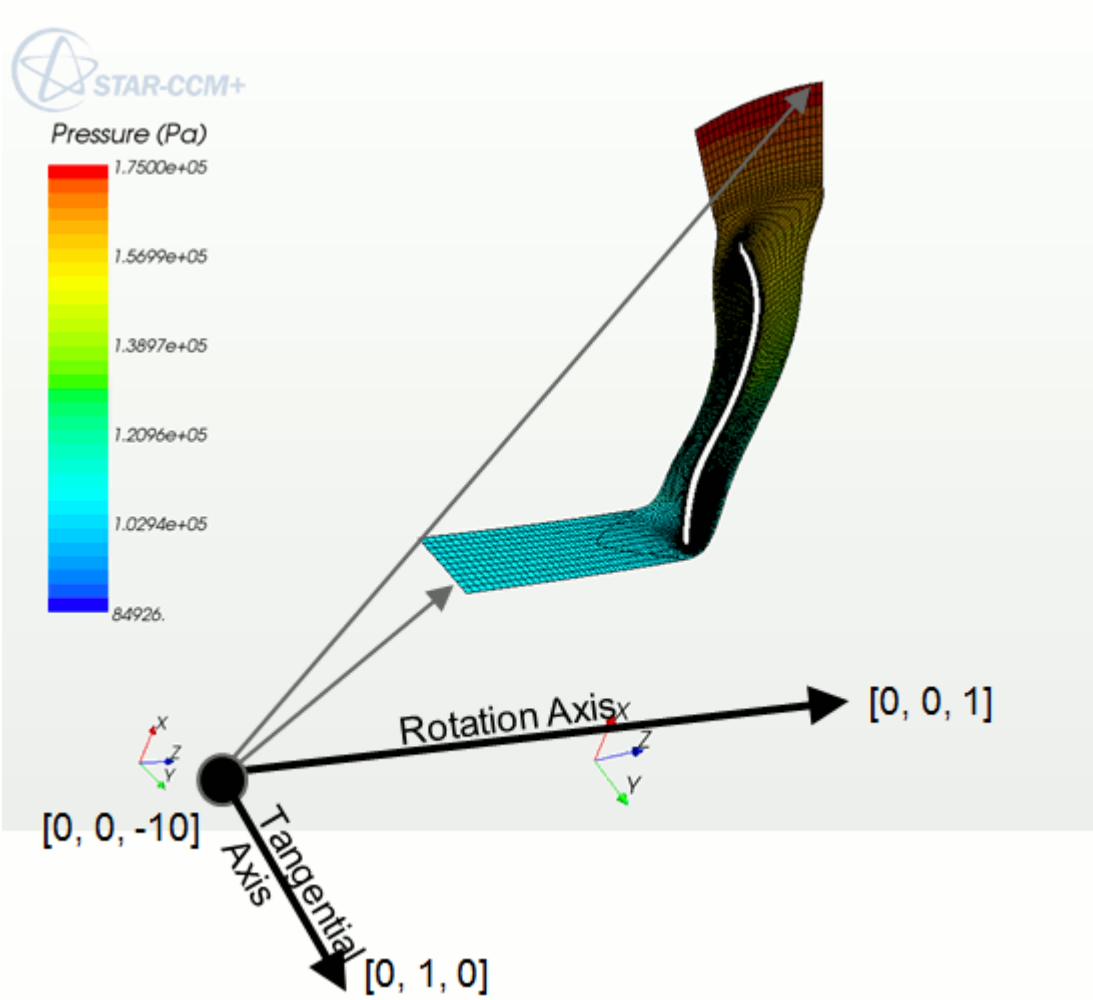


The meridional coordinate is defined as:

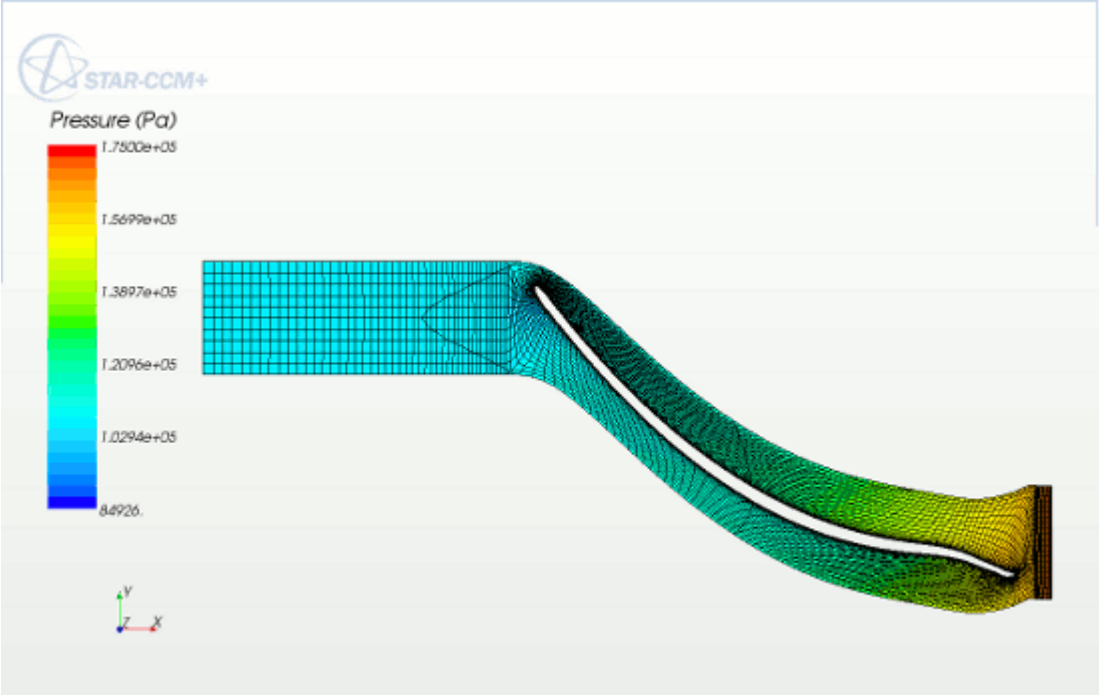
$$m = \sqrt{\left(r^2 + z^2\right)} \tag{1208}$$

The *Tangential Axis* property does not have to be orthogonal to the *Rotation Axis* since only the component orthogonal to the rotation axis is used.

The following example uses the (m', theta) derived part:



- The rotation axis is [0, 0, 1] (aligned with the z-axis).
- The origin is upstream of the surface at [0, 0, -10].
- The tangential axis, [0, 1, 0], is orthogonal to the rotation axis.
- The following screenshot shows the view down the z-axis. The derived part maps the m’ coordinate to x and the theta coordinate to y.



The derived part is computed by determining the generating curve from the part. So if there are multiple parts in a displayer, they must form a continuous surface for the derived part to be valid.

To create an (m',theta) Warp:

1. Right-click the **Derived Parts** node and select **New Part** > **(m', theta) warp**.

   An (m', theta) warp part appears in the simulation tree immediately.
2. In the *Properties* window, specify the input parts, the *Rotation Axis* and its *Origin*, and the *Tangential Axis*.

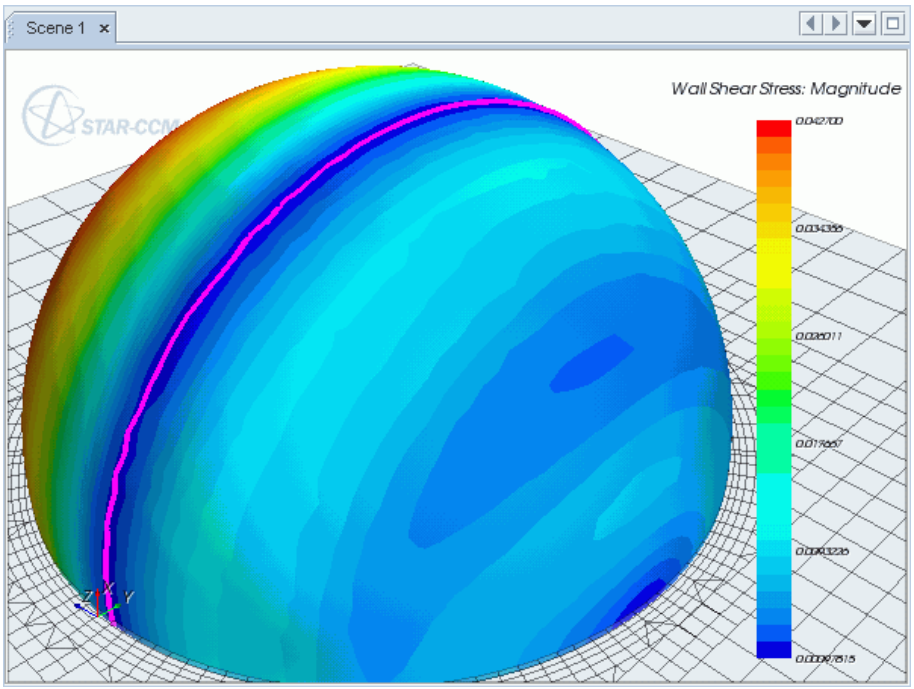
   For more information on the available properties, see (m',theta) Warp Properties.

Defining a Separation/Attachment Line

Separation/attachment lines are lines along which flow separates from or attaches to a surface.

This phenomenon is induced by viscous effects that take place in direct proximity to the point of separation or attachment. When a fluid flows over a no-slip wall boundary, the velocity magnitude goes to zero as one approaches the surface along a normal direction. Therefore, the extraction of separation and attachment lines focuses on the non-zero, tangential shear-stress vector field defined over the surface that exhibits the same flow patterns as nearby streamlines.

The following image shows how STAR-CCM+ uses the wall shear stress vector field to identify separation lines (the line is shown in magenta for clarity). However, several other parameters are used to identify a separation line, and you can also use cell relative velocity as the vector field. The cell relative velocity is the fluid flow velocity at the cell centroid relative to the cell centroid due to motion.



To create a separation/attachment line:

1. Follow the general procedure outlined in the section, Defining Derived Parts General Workflow. Choose the menu option, **New Part** > **Separation/Attachment Line...**.
2. Under *Input Parts*, specify the input boundaries that are used to generate the separation/attachment line. Typically, this part is a no-slip wall, over which fluid is flowing. Also, specify a display option.

   For more information on specifying input parts and display options, see Defining Derived Parts General Workflow
3. Complete the separation/attachment line definition by specifying the following properties:

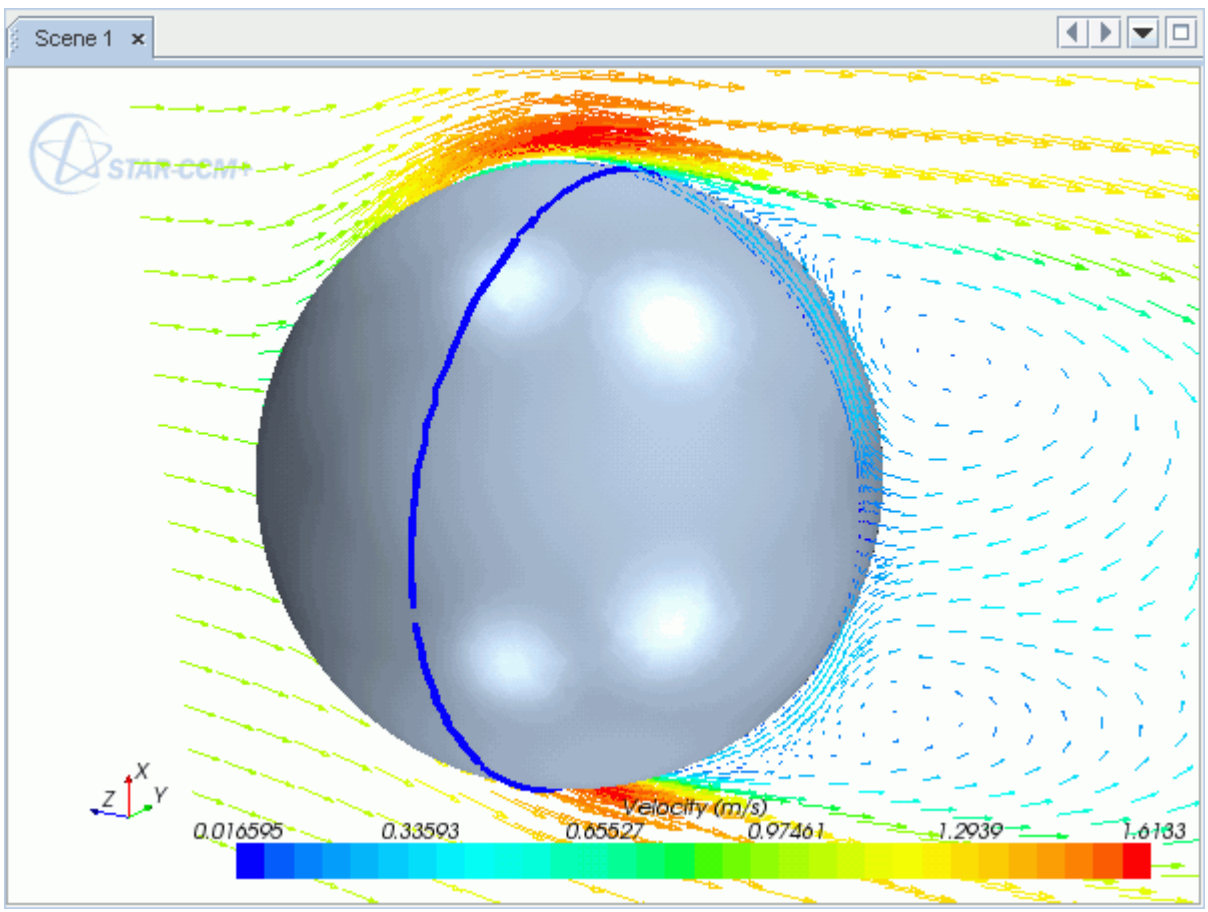
- *Vector Field* — select the vector field used by the algorithm to calculate the separation/attachment line. This must be either wall shear stress or cell relative velocity. If you select another vector field, a separation/attachment line is still created, but the result will be invalid.
- *Display Attachment Lines* and *Display Separation Lines* — specify whether attachment or separation lines are included in the derived part. Attachment occurs when flow meets a wall boundary and flows along the surface. Separation occurs when flow leaves the surface.
- *Line Strength* — this property does not represent a physical quantity, but is used by the code to filter out weak results (that is, lines that do not represent a strong flow separation).
- *Minimum Line Length* — specify a minimum length for the derived part lines. The lines that are shorter than the minimum length are excluded.
- *Line Quality* — specify the angle between the separation/attachment line and the flow vector, which filters lines from the derived part that are at a large angle to the flow.

You can set all properties in either the in-place dialog or the *Properties* window.

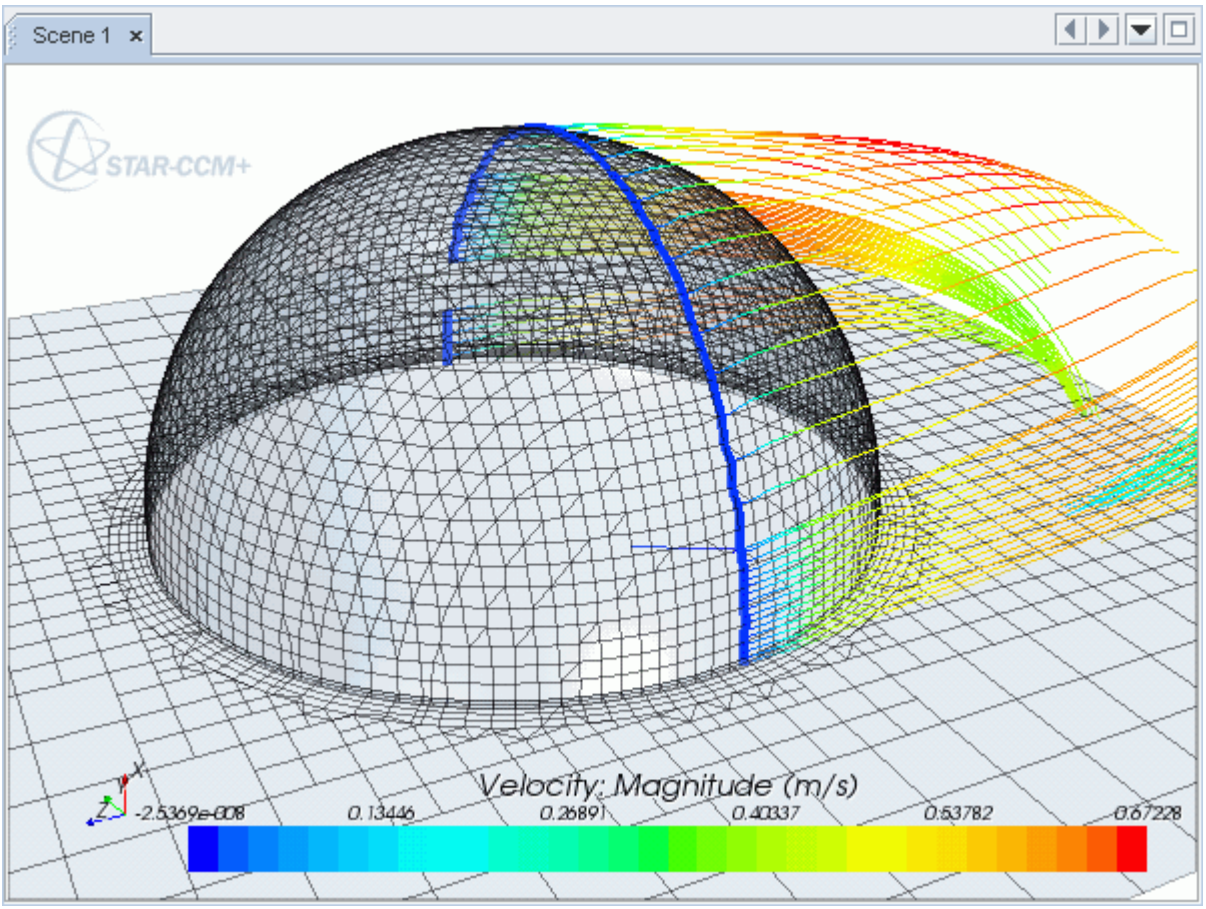
For more information on the available properties, see Separation/Attachment Line Properties.

Once you have extracted a separation/attachment line, you can display it in a graphics scene with a scalar color, if desired. You can also use it as the input part for a streamline displayer, so that the streamlines begin along the separation line.

The following image shows a separation line that is displayed using a streamline displayer. The thickness of the line has been increased to make it more prominent on the geometry surface. The velocity vectors illustrate where flow separation occurs, which matches the separation line location.



You can also use separation/attachment lines as a seed part for a streamline derived part. This can be used to analyze the flow path after the point of flow separation, which is illustrated in the example below.



Defining a Vortex Core Part

In STAR-CCM+, the vortex core part automatically extracts vortex cores in a flow field.

Vortices are features in the flow data which are of great importance in fluid dynamics, both in theoretical and practical research. In some cases, vortices (turbulence) have to be impelled, for example to stimulate mixing of fluids, or to reduce drag. In other cases, vortices have to be prevented, for example around aircraft, where they can reduce lift.

The vortex core is the axis line around which the flow field rotates. The core provides a precise position of the vortex and also allows you to distinguish between different vortical structures close to each other.

Vortex core visualization belongs to the category of feature-based flow visualization techniques. The methods and algorithms in this category present the flow data in a higher level of abstraction. The flow data is described by a set of interesting and important objects or structures in the data called features. In fluid dynamics vortices, shock waves, separation lines, and attachment lines are examples of such features. Since only a small percentage of the data is of interest, and the features can be described compactly, it is possible to achieve considerable data reduction. This enables an interactive visualization of large amounts of fluid simulation data.

The vortex core extraction algorithm needs the velocity field and a three-dimensional region to generate poly lines representing vortex cores. It has three filtering parameters to filter out unwanted vortex cores.

There are many different definitions of vortices and likewise many different vortex detection algorithms. These definitions are based on certain field values being high or low, like high vorticity magnitude, high helicity magnitude, or low-pressure value. However, there is no accepted threshold value for what is considered high or low.

Definitions of features are usually implicit because they are procedural. However, precise mathematical definitions are attainable with the parallel vectors operator proposed by Peikert and Roth [185], [186]. This operator is an elementary operation on two vector fields, which in turn make it possible to define and calculate global line-type features of vector or scalar fields. The parallel vectors operator can be extended to unsteady fields. Advantages include:

- Comparing feature definitions
- Reusing algorithms and implementations

The parallel vectors operator causes the feature definition to work separately from the actual implementation. According to Roth and Peikert, along with various line-type features, the operator can implement algorithms to extract the following:

- Vortex core lines
- Lines of separation or attachment

STAR-CCM+ uses the parallel vectors operator to implement the Eigenvector method for vortex core extraction proposed by Sujudi and Haimes [188], [184]. According to these authors, a rotational motion is present when the Jacobian (velocity gradient matrix) has a conjugate complex pair of Eigenvalues. The description of how the parallel vectors operator can be applied to the Eigenvector method is in the papers by Peikert and Roth [187], [185]as well as in the dissertation by Roth [186].

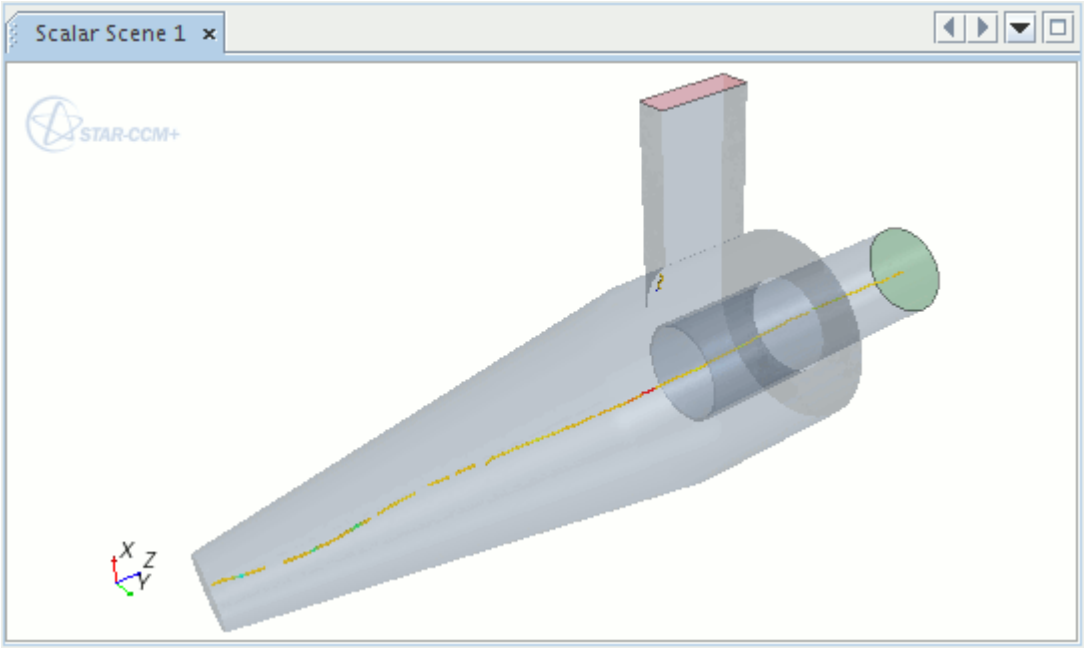
To create a vortex core:

1. Follow the general procedure outlined in the section, Defining Derived Parts General Workflow. Choose the menu option, **New Part** > **Vortex Core...**
2. Specify the input parts and display option as described in the section, Defining Derived Parts General Workflow.
3. Complete the vortex core definition by specifying the following properties:
   - Vector field
   - Vortex strength
   - Vortex quality
   - Minimum core length

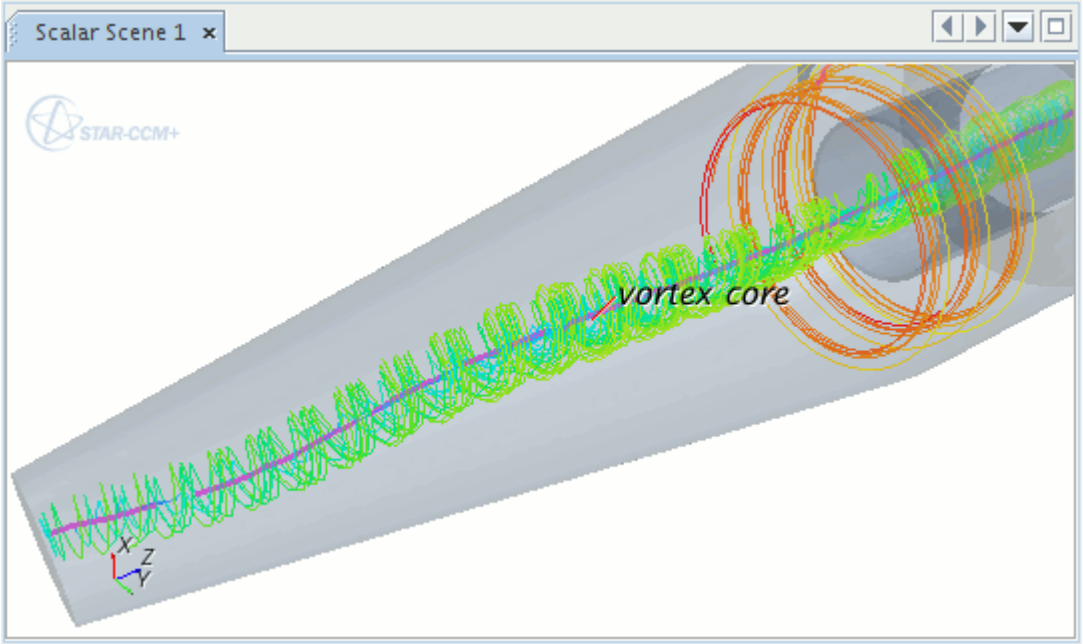
   You can set all properties in either the in-place dialog or the *Properties* window.

   For more information on the available properties, refer to the section Vortex Core Properties.

   You can display vortex cores on a geometry scene or as parts in streamline displayers. Within streamline displayers, you can display them as lines, ribbons, or tubes and color code them by a scalar as shown below.



   The vortex core can be used to seed streamlines in order to visualize the vortical structure as shown in the figure below.


Scalar Scene 1
STAR-CCM+
vortex core
X
Z
Y

Defining Probes

This section explains the derived parts that take samples of data in a region based on coordinates or intervals that you specify.

These parts are:

- Point
- Line-probe
- Presentation grid
- Arbitrary probe

How data for a particular point is extracted depends on your selection of input parts.

Input Parts Type	Reported Data Value
Region	Cell value
Region where Probe Point is outside of Region	Meaningless value (Probe Point is discarded)
Boundary	Surface cell value of the face that is closest in Euclidian distance to the Probe Point
Region and Boundary	Cell value
Region and Boundary where Point is outside of Region	Surface cell value of the face that is closest in Euclidian distance to the Probe Point.

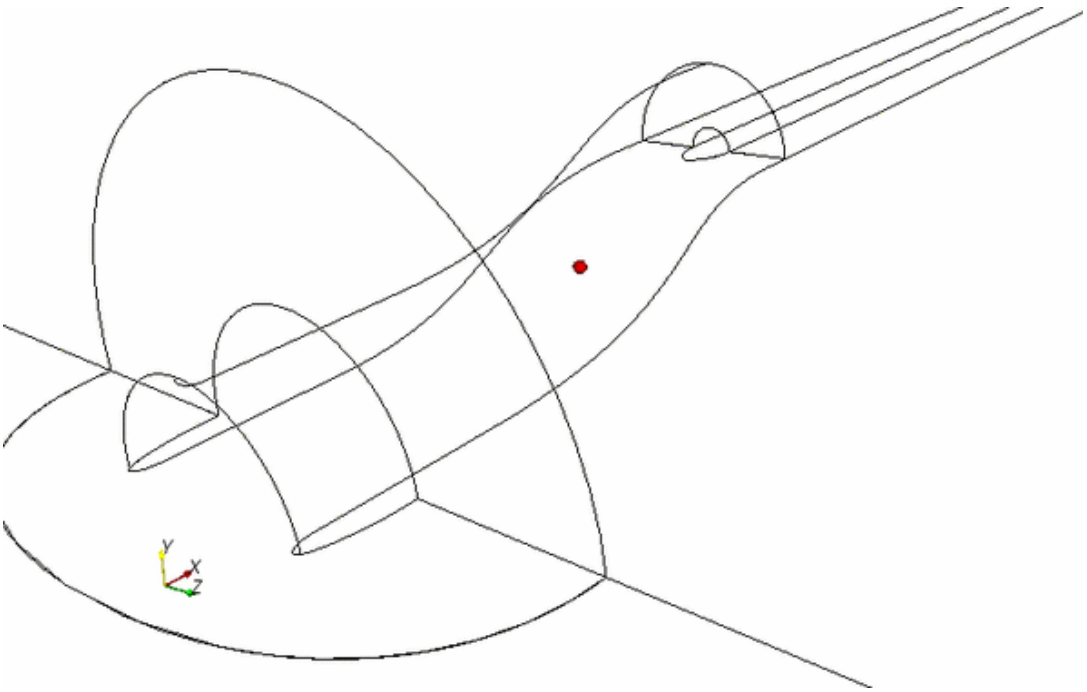
Contents:

Defining a Point Probe
Defining a Line Probe
Defining a Presentation Grid
Defining an Arbitrary Probe

Defining a Point Probe

The point part samples its input part at a set of coordinates in a region.

STAR-CCM+ allows the creation of a point part from parent parts consisting of regions and boundaries only. The point location is specified by entering the coordinates of the point. A point tool is available to assist with positioning the point graphically. The point is represented in the scene by a sphere of constant radius in screen coordinates.



To create a point probe:

1. Follow the general procedure outlined in the section, Defining Derived Parts General Workflow. Choose the menu option, **New Part** > **Probe** > **Point...**
2. Specify the input parts, display option, and coordinate system as described in the section, Defining Derived Parts General Workflow.

   From the in-place dialog, you can use the point tool, which allows you to position the point visually. The **Snap To Part** option lets you select a part along which you can position the point (this action is essentially snapping to the bounding box of the part).
3. In the *Properties* window or in-place dialog, specify the point x-, y-, and z-coordinates with respect to the specified coordinate system, using the *Point* property/group box.

   In the in-place dialog, you can activate *Lock* to fix the position of any coordinate or any combination of them.
4. To monitor data on a moving mesh, activate *Follow Motion* in the *Properties* window.

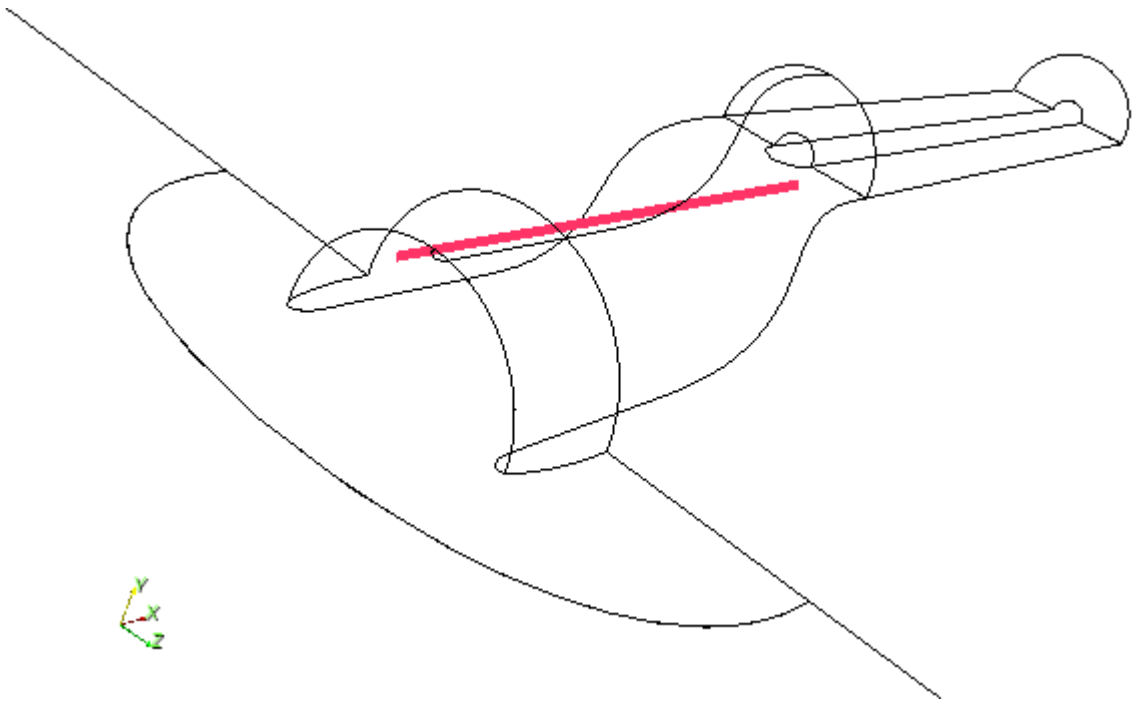
   For more information on the available properties, see Point Probe Properties.

   You can use reports to extract data from point probes. To extract data at Point Probes previously specified, create a Report of the type Maximum or Minimum (see Reporting Results). When the *Smooth Values* expert option in the *Properties* window of the report is selected, vertex values for the cell are interpolated to the Point Probe.

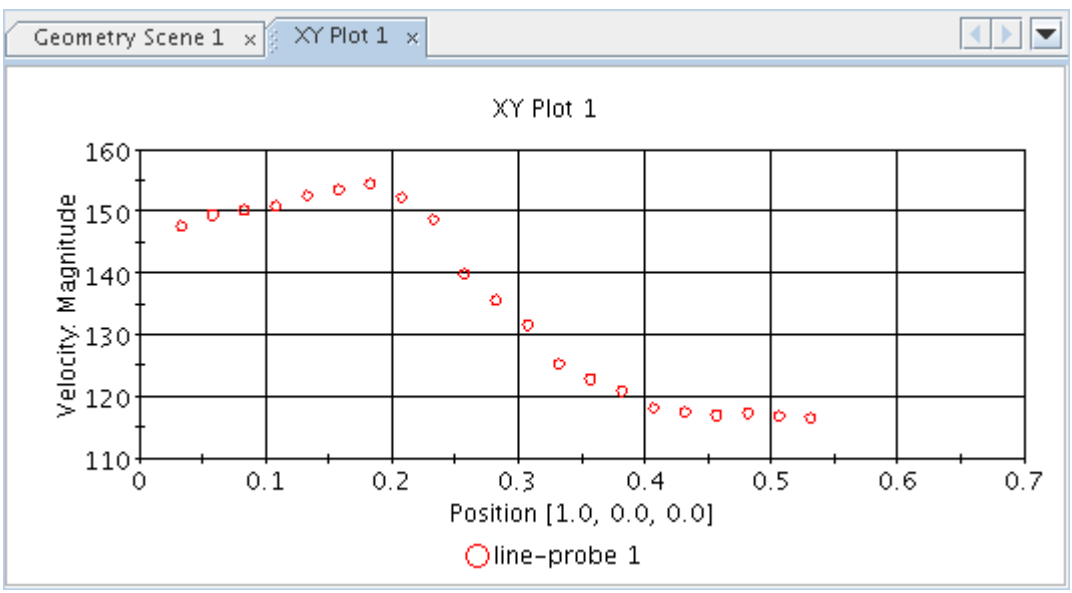
Defining a Line Probe

The line-probe part samples its data from evenly spaced points along a line in a region.

STAR-CCM+ allows the creation of a line-probe part from parent parts consisting of regions and boundaries only. The line location is specified by entering the coordinates of the line. A line tool is available to assist with positioning the line endpoints graphically. A line-probe can be incorporated into an XY plot.



The line-probe can be useful for extracting data along a line in a region, and can be used to display scalars or vectors. The most common use of data extracted with a line-probe is selecting it for an XY plot.



To create a line probe:

1. Follow the general procedure outlined in the section, Defining Derived Parts General Workflow. Choose the menu option, **New Part** > **Probe** > **Line...**
2. Specify the input parts, display option, and coordinate system as described in the section, Defining Derived Parts General Workflow.

From the in-place dialog, you can use the line tool, which allows you to position the line visually. The **Snap To Part** option lets you select a part along which you can position the line (this action is essentially snapping to the bounding box of the part). See Using the Line Tool.

3. In the *Properties* window or in-place dialog, set *Point 1* and *Point 2* to specify the x-, y-, and z-coordinates of the points at the two ends of the line, with respect to the specified coordinate system.
4. In the *Properties* window or in-place dialog, set the *Resolution*, that is, the number of evenly spaced segments along the line.

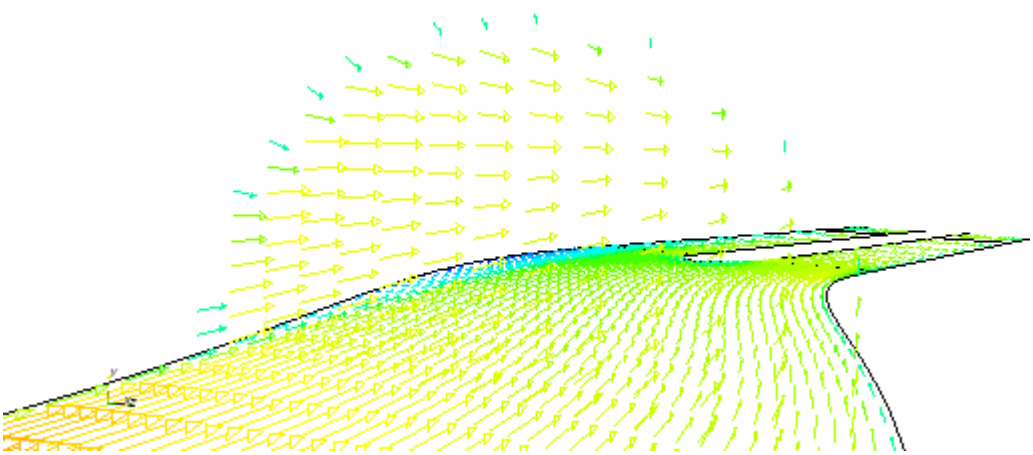
   The overall number of points is one greater than this; fewer points may be generated if part of the probe lies outside the model geometry.

   For more information on the available properties, see Line Probe Properties.

Defining a Presentation Grid

The presentation grid part samples data from regularly spaced intervals on a finite plane in a region.

This part is similar to the plane section, the difference being that the output points of a presentation grid are sampled on a regular grid, whereas points from the plane section are based on the topology and discretization of the underlying mesh. STAR-CCM+ allows the creation of a presentation grid part from parent parts consisting of regions and boundaries only. The finite plane can be defined by specifying a point for the origin, and two other points which, together with the origin define two axes for the plane. An explicit plane tool is available to assist with positioning the plane graphically. The presentation grid can be useful for extracting data at structured intervals in a region.



To create a presentation grid:

1. Follow the general procedure outlined in the section, Defining Derived Parts General Workflow. Choose the menu option, **New Part** > **Probe** > **Presentation Grid...**
2. Specify the input parts, display option, and coordinate system as described in the section, Defining Derived Parts General Workflow.

   If a presentation grid spans two parts, selecting only one part as input part will ignore all data points in the second part.
3. Complete the presentation grid definition by setting the following properties:

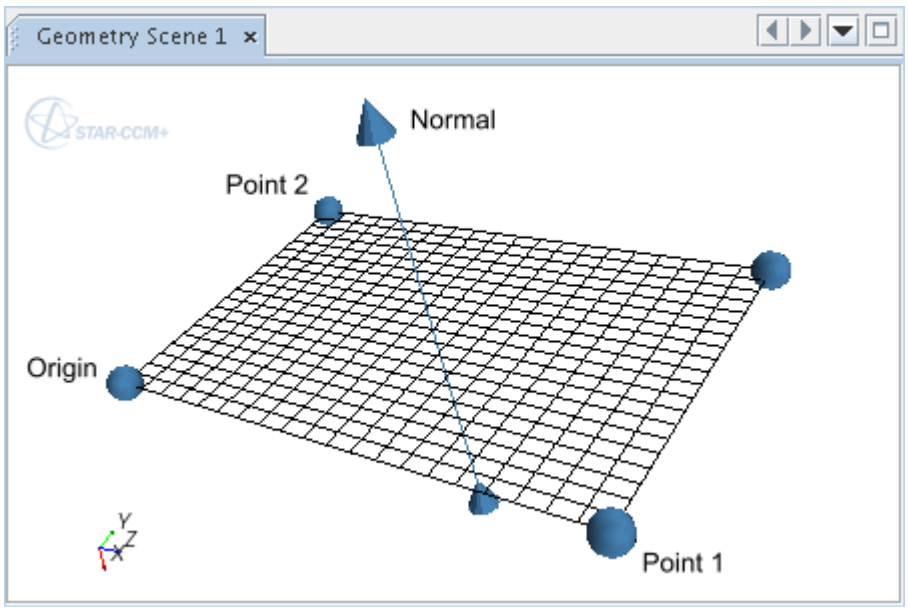
- In the in-place dialog, position the plane by either setting the *Center* and *Normal* properties, or by using the plane tool in the scene display. See Using the Explicit Plane Tool.

  You can activate the **Snap To Part** option to position the plane along a part (this action is essentially snapping to the bounding box of the part).

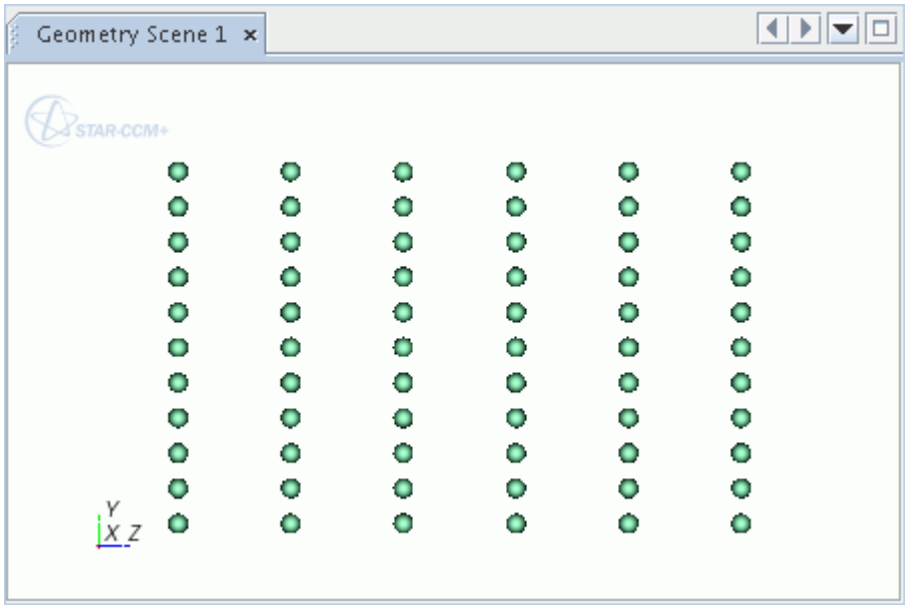
  STAR-CCM+ automatically calculates the coordinates of the origin, point 1, and point 2 based on the plane position.
- Alternatively, specify the x, y, and z coordinates of the *Origin* in the *Properties* window. Then, specify the normal or the coordinates of *Point 1* and *Point 2*.

The coordinates are relative to the specified coordinate system. If using a cylindrical or spherical coordinate system, the units for position vectors update accordingly.

Note: Using a cylindrical or spherical coordinate system does not create a cylindrical or spherical presentation grid.



4. Specify the *X Resolution* and the *Y Resolution* in the *Properties* window or the in-place dialog.
   For example, the following grid has an *X Resolution* of `5` and a *Y Resolution* of `10`.

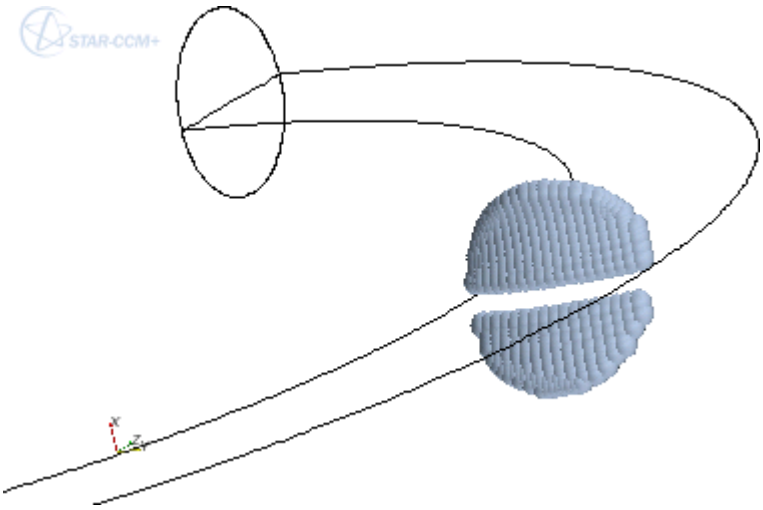


For more information on the available properties, see Presentation Grid Properties.

Defining an Arbitrary Probe

The arbitrary probe is an implicit part that samples its input part or region from a set of specific points.

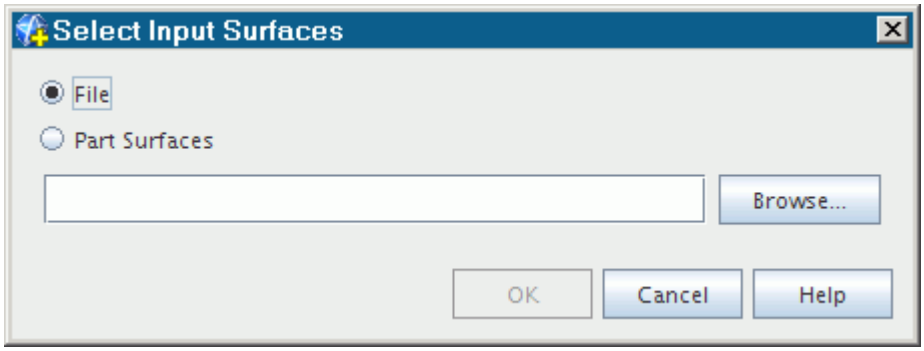
An arbitrary probe works in a manner similar to the other probe derived parts, except that it extracts its sample points from either a CAD part surface or from an STL or VTK file.



To create an arbitrary probe:

1. Follow the general procedure outlined in the section, Defining Derived Parts General Workflow. Choose the menu option, **New Part** > **Probe** > **Arbitrary...**

   If the conditions for activating the in-place dialog are not met, the *Select Input Surfaces* dialog appears.



To define the limits of the arbitrary section, get shape data from part surfaces or an external file:

2. Specify the part surfaces or import an STL file using one of the following methods:
   - To import an STL file, activate **File** in either the *Select Input Surfaces* dialog, or in the *Surface Input Options* group box in the in-place dialog, then click **Browse...** and use the *Open* dialog to select the `.stl` file to import.
   - To use part surfaces, activate **Part Surfaces** in either the *Select Input Surfaces* dialog, or in the *Surface Input Options* group box in the in-place dialog, then click **Select...** and use the *Select Objects* dialog to specify the part surfaces.

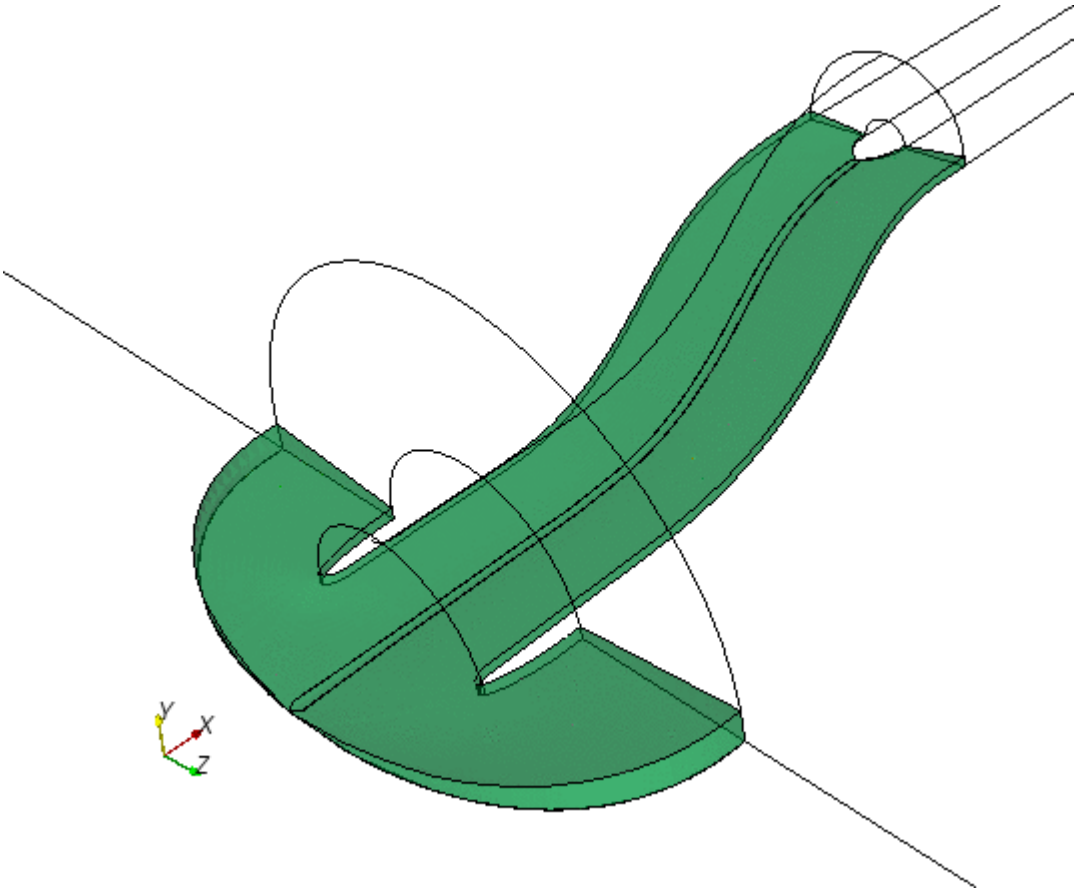
     This action adds the data from the specified part surfaces.
3. Specify the input parts and display option as described in the section, Defining Derived Parts General Workflow.

   For more information on the available properties, see Arbitrary Probe Properties and Actions.

Defining a Cell Surface

The cell surface part represents all the cells adjacent to or intersected by a given part.

A cell surface part can be useful for extracting boundary or interior cells. For example, a cell surface created from a boundary shows the cells adjacent to the boundary, whereas a cell surface created from a plane section shows all cells intersected by the plane. A cell surface part can be created using any part of dimensionality two or lower.



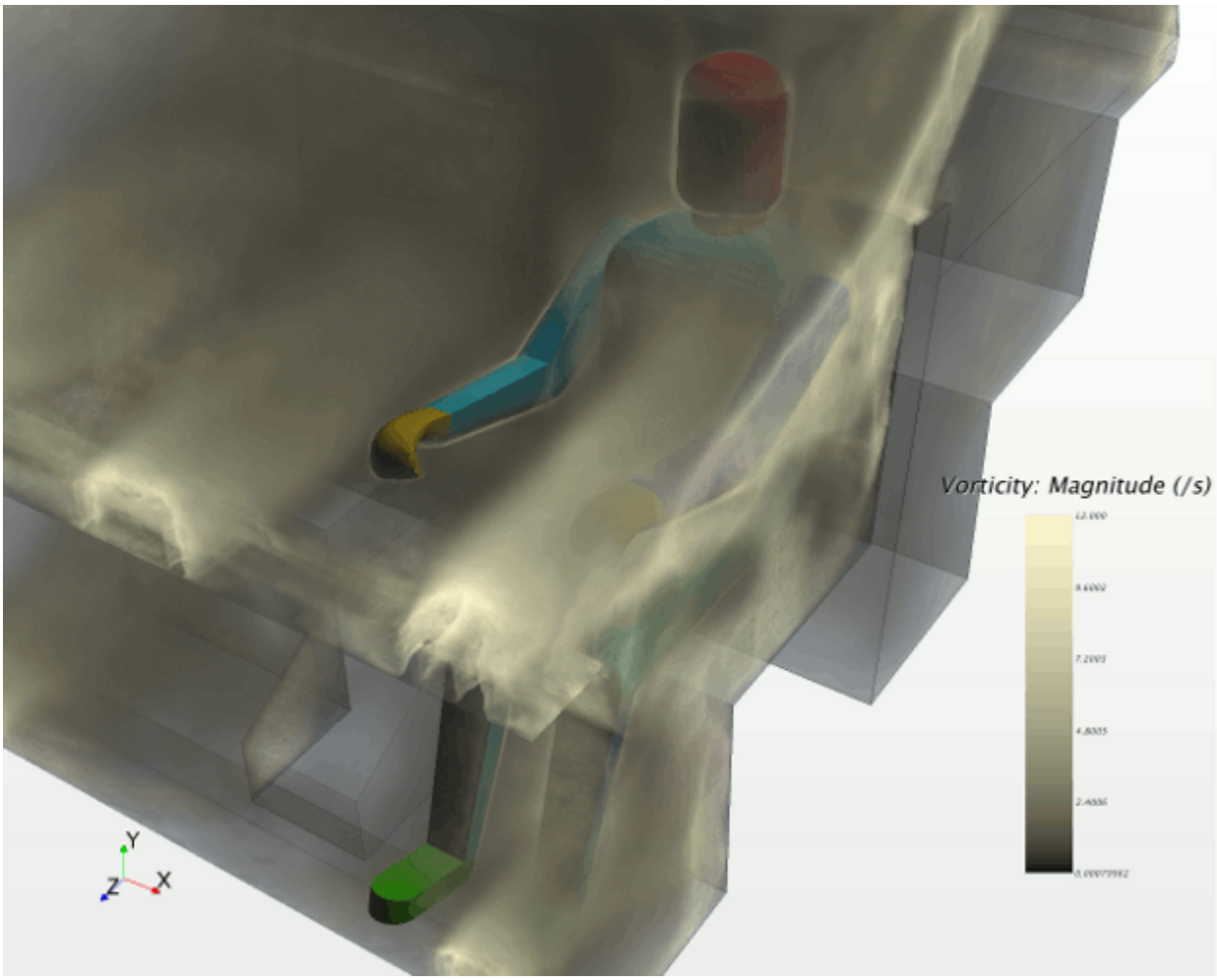
To create a cell surface:

1. Follow the general procedure outlined in the section, Defining Derived Parts General Workflow. Choose the menu option, **New Part** > **Cell Surface...**
2. Specify the input parts and display option as described in the section, Defining Derived Parts General Workflow.

Defining a Resampled Volume

Volume rendering is used to display (semitransparent) volumetric objects that are defined in a 3D domain. If your graphics card does not support volume rendering, activate advanced rendering for the relevant scene.

The resampling process generates an approximation of the result values that reside in the underlying 3D finite volume regions. For each position in the 3D domain, color and opacity values are specified.



To create a resampled volume:

1. Follow the general procedure outlined in the section, Defining Derived Parts General Workflow. Choose the menu option, **New Part** > **Resampled Volume...**.
2. Specify the input parts and display option as described in the section, Defining Derived Parts General Workflow. The resampled volume part can be rendered only in the scalar displayer.
3. To display the resampled volume data, set the *Representation* property of the scalar displayer to `Volume Mesh`.

4. Specify the dimensions and position of the bounding box by setting the following properties, in either the in-place dialog or the *Properties* window:
   - *Origin*
   - *Size*
   - *Rotation Axis*
   - *Rotation Angle*

   When using the in-place dialog, you can activate **Snap to part** to position the bounding box along a part. This feature is available when the resampled volume tool is active. See Using the Resampled Volume Tool.
5. To control the number of voxels in the resampled volume, set the following properties:
   - *Cell Size to Voxel Ratio*
   - *Minimum Voxel Size*

   For more information, see Resampled Volume Properties.

   When using the in-place dialog, you can click **Preview** to get a graphical preview of the resampled volume and a preliminary estimate of the number of voxels.

When you add a resampled volume part to a scalar displayer, that displayer acquires a new sub-node, **Volume Rendering Settings**. This node allows you to choose a rendering mode, activate the lighting mode, and specify accuracy:

6. Set the properties of the **Volume Rendering Settings** node to control how the data is displayed.

   For more information on the available properties, see Resampled Volume Properties.
7. Complete the resampled volume definition by specifying level surfaces, as described in Working with the Level Surface Mode.

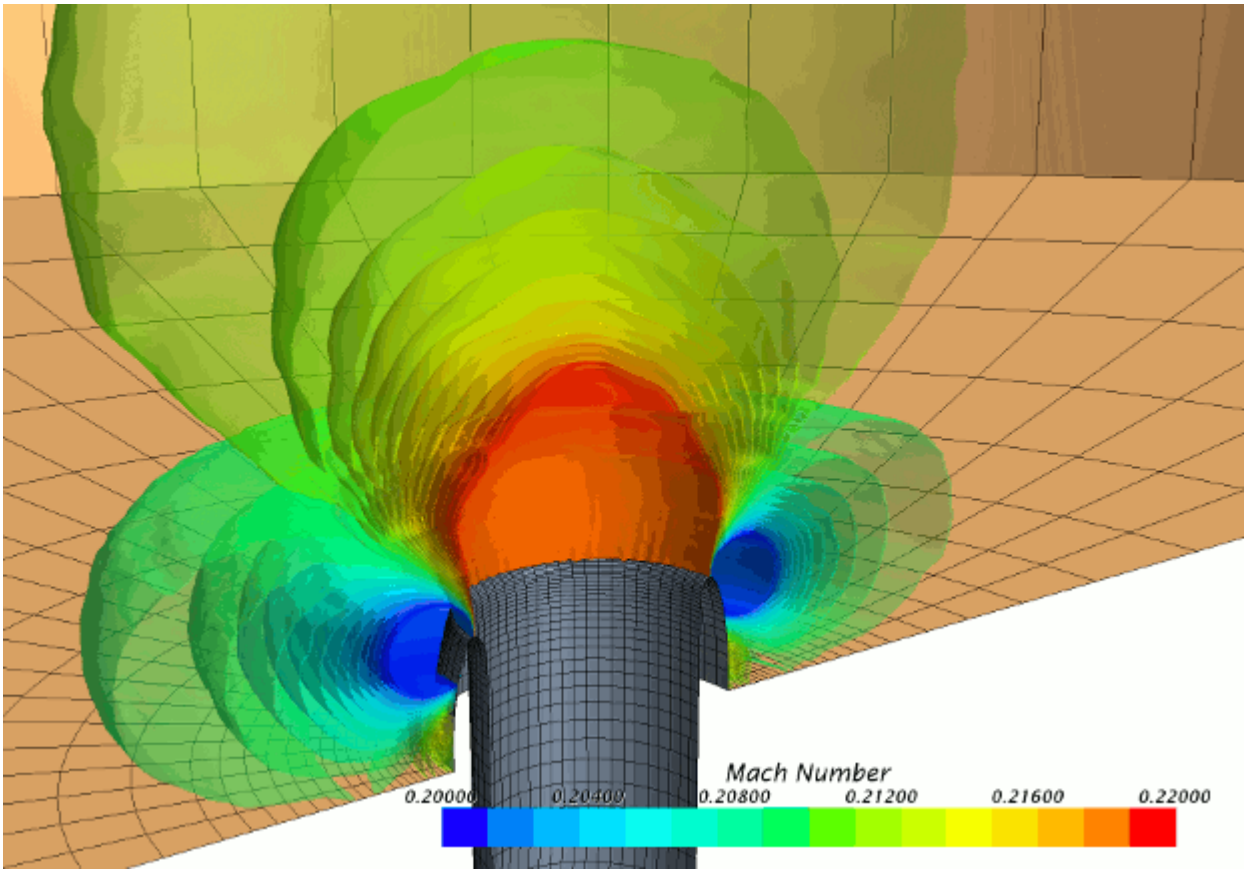
   Once you have finished creating a resampled volume part, you can rotate and zoom in the display to get a clear view of the resampled volume. If you did not assign the part to the current scene, you can drag and drop its node into the visualization display. The resampled volume part can be rendered only in the scalar displayer.

Contents:

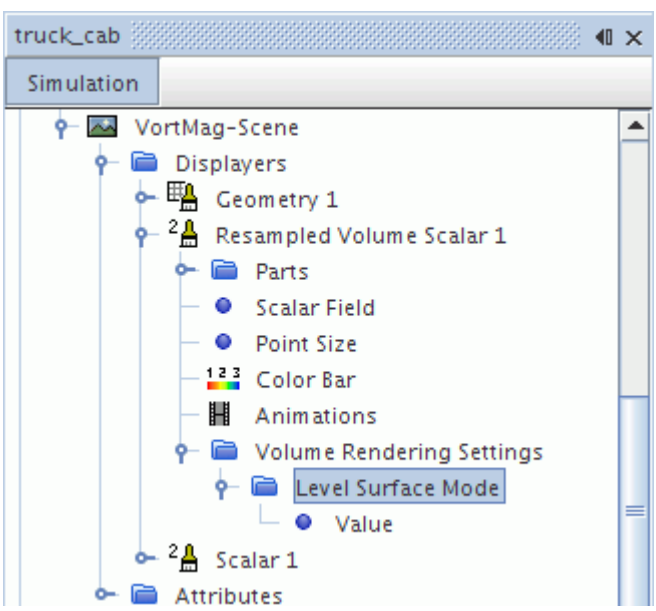
Working with the Level Surface Mode
Working with Volume Rendering Colormaps

Working with the Level Surface Mode

Volume rendering in STAR-CCM+ is based on a ray-casting approach that is implemented in graphics hardware. The level-surface volume rendering mode tries to capture user-defined level values of interest during ray-casting through the resampled volume derived part.

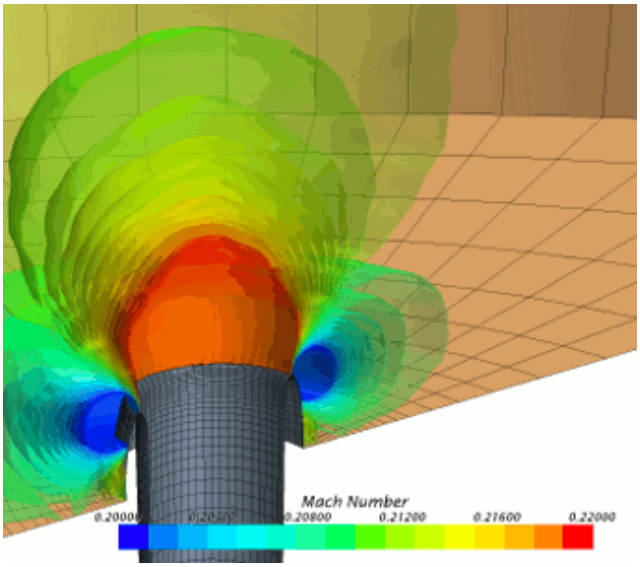
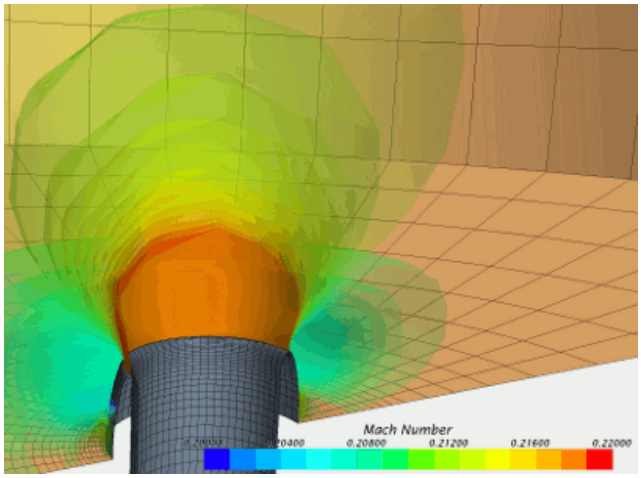


The **Level Surface Mode** node allows you to choose between single and multiple surfaces for volume rendering in its properties.



Level Surfaces Compared with Isosurfaces

Level surfaces are in principle similar to isosurfaces. However, level surfaces provide a better visual accumulation of partly transparent surfaces than isosurfaces with the same opacity settings. The data inputs, data extraction approaches, and storage locations of these types of objects are different. The effects of the inputs and extraction depend on the mesh, the native data location of the scalar field, and the accuracy settings for the resampled volume part.

	
Level Surfaces	Isosurfaces

Feature	Data Input	Extraction Approach	Hardware Employed	Storage	Location
Level Surfaces	Resampled Volume derived part	Ray-Casting	Graphics Hardware / GPU	Image buffer in graphics memory	Client
Isosurfaces	Discrete meshed region or part	Polygonalization of intersected mesh edges	CPU	Polygonal surface mesh in main memory	Client and Server

See also Examples of Effects of Volume Rendering Settings on Level Surfaces.

For an interactive exploration of simulation results, level surfaces have the following advantages over isosurfaces:

- Typically they update very fast, at the speed of volume rendering.
- They provide better results that are based on the opacity that you specify in the displayer and the colormaps.

In comparison to isosurfaces, the level surfaces have the following limitations:

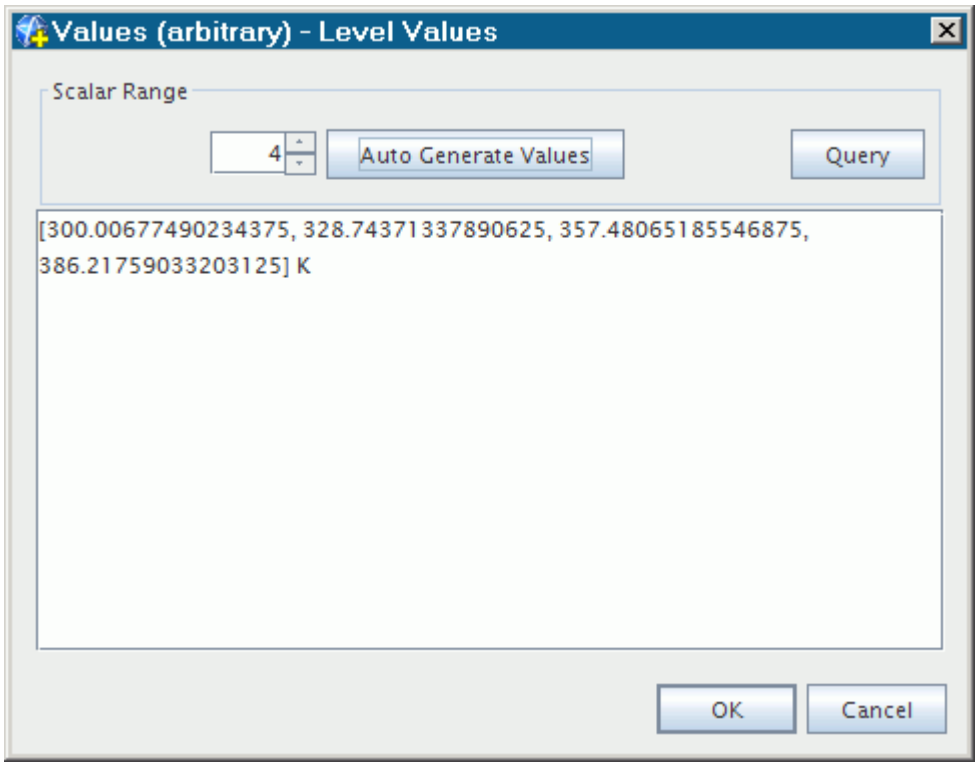
- They cannot show a secondary scalar field on their surface.
- They cannot serve as an input to other derived parts or reports.

- They support a maximum of 32 level values.
- Their accuracy depends on the resolution of the underlying resampled volume. An example is bumpy disruptions of an otherwise flat surface. If such artifacts occur, use isosurfaces.
- They do not work with software-based OpenGL implementations such as Mesa OpenGL, or Microsoft OpenGL (remote desktop).

Specifying Level Surfaces

1. To work with a single level surface, select the **Level Surface Mode** > **Value** node and enter a value for *Level Value*.
2. To work with multiple level surfaces that are equally spaced in a range:
   - Select the **Level Surface Mode** node and set *Level Surface Mode* to **Multiple (range)**.
   - Select the **Level Surface Mode** > **Values (range)** node and set the following properties:
     - *Number of Levels*
     - *Minimum Level Value*
     - *Maximum Level Value*

   For more information on the available properties, see Level Surface Mode Properties .
3. To work with multiple level surfaces that you arbitrarily set:
   - Select the **Level Surface Mode** node and set *Level Surface Mode* to **Multiple (arbitrary)**.
   - Select the **Level Surface Mode** > **Values (arbitrary)** node and set the properties using one of the following techniques:
     - Enter settings for *Level Values*.
     - Click ... (**Property Customizer**) of the *Level Values* property.



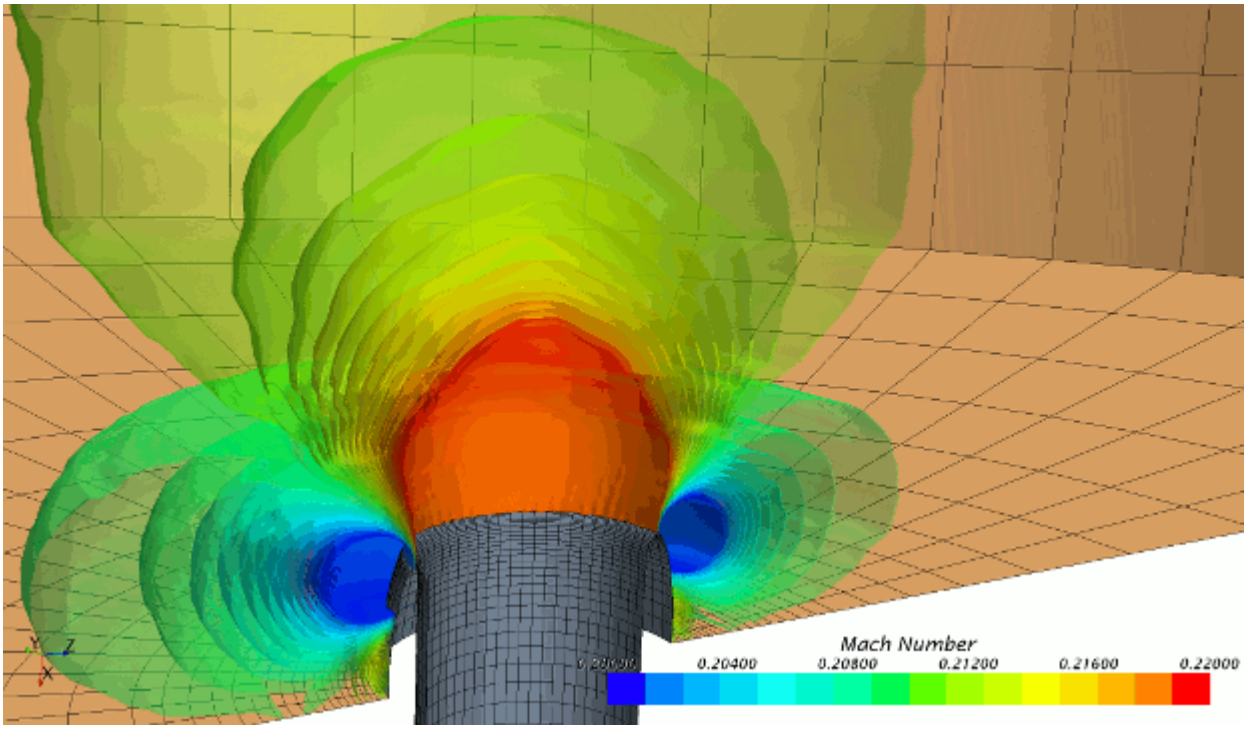
In the dialog that appears, enter values as follows:

- Set the number of level surfaces in the spin box.
- If you know the exact values to display, enter them manually in the text box and click **OK**.
- If you want STAR-CCM+ to generate values, click **Query**, then click **Auto Generate Values**. Click **OK**.

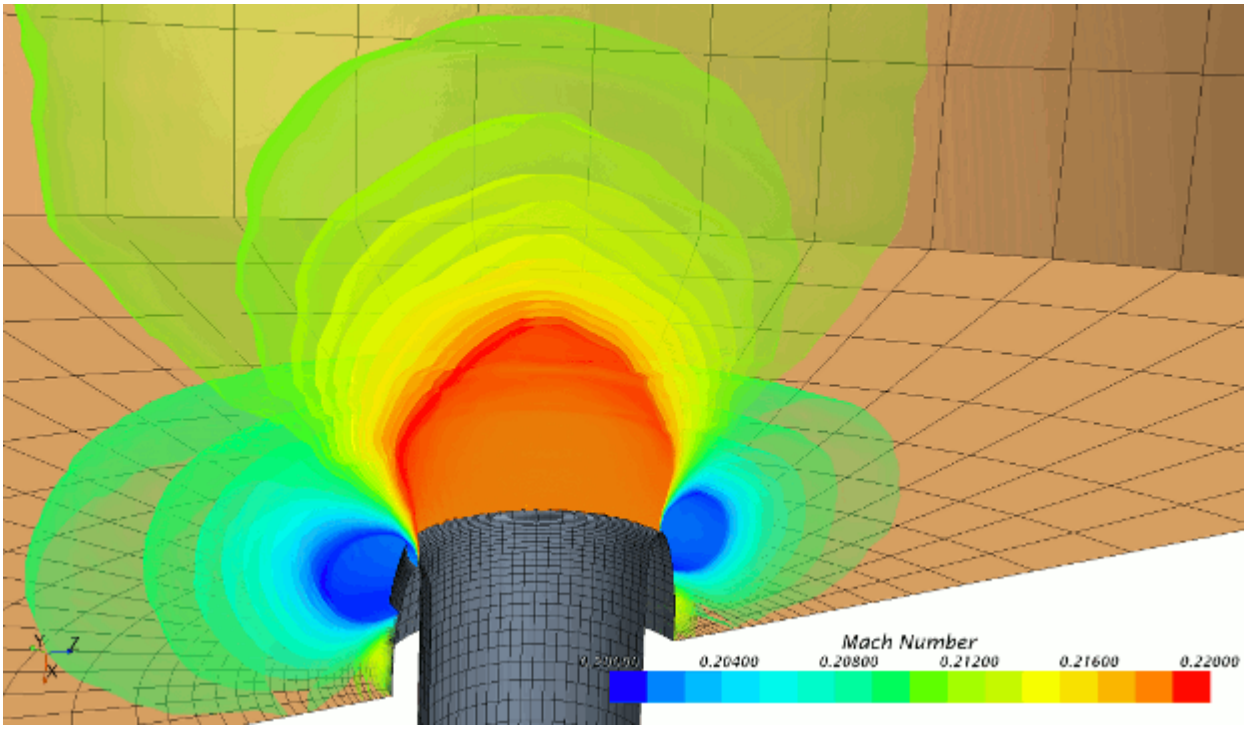
  The **Query** action retrieves the scalar range from the server. The **Auto Generate Values** action produces an even set of values in the text box according to the number of level surfaces that you specify.

Examples of Effects of Volume Rendering Settings on Level Surfaces

Lighting mode:

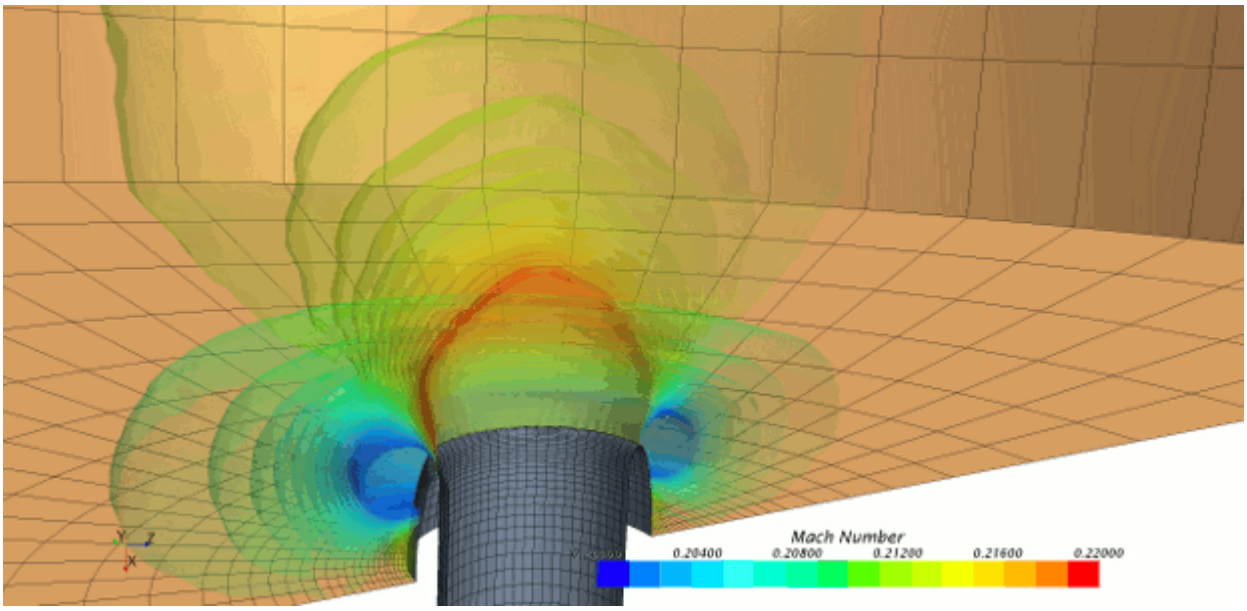


Local lighting on

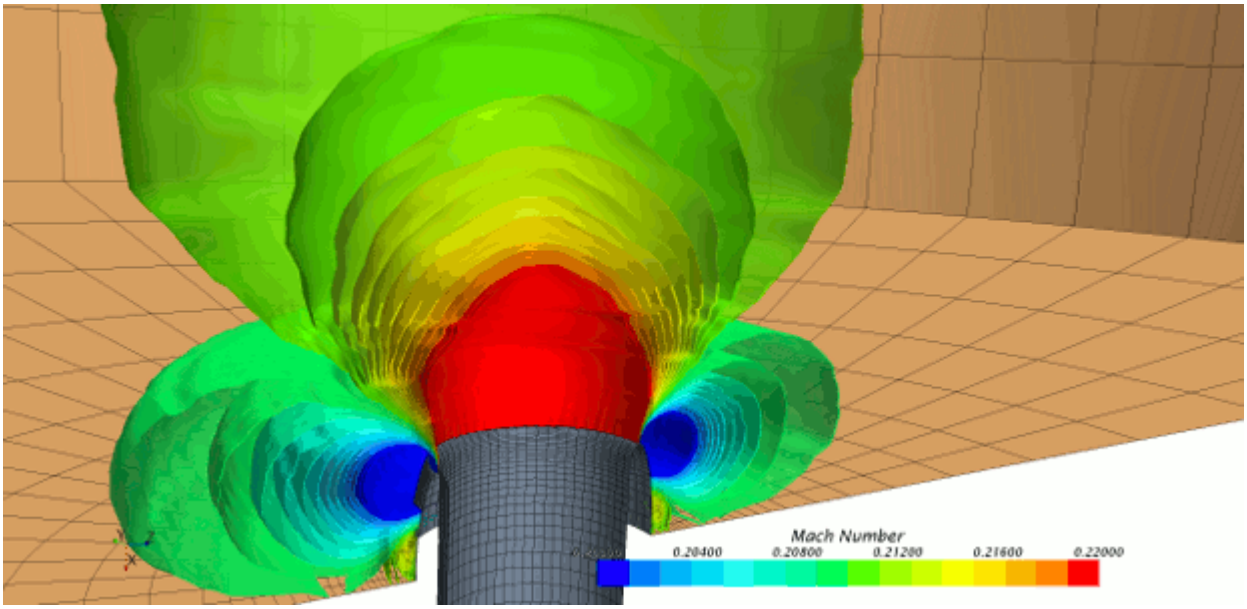


Local lighting off

Opacity:

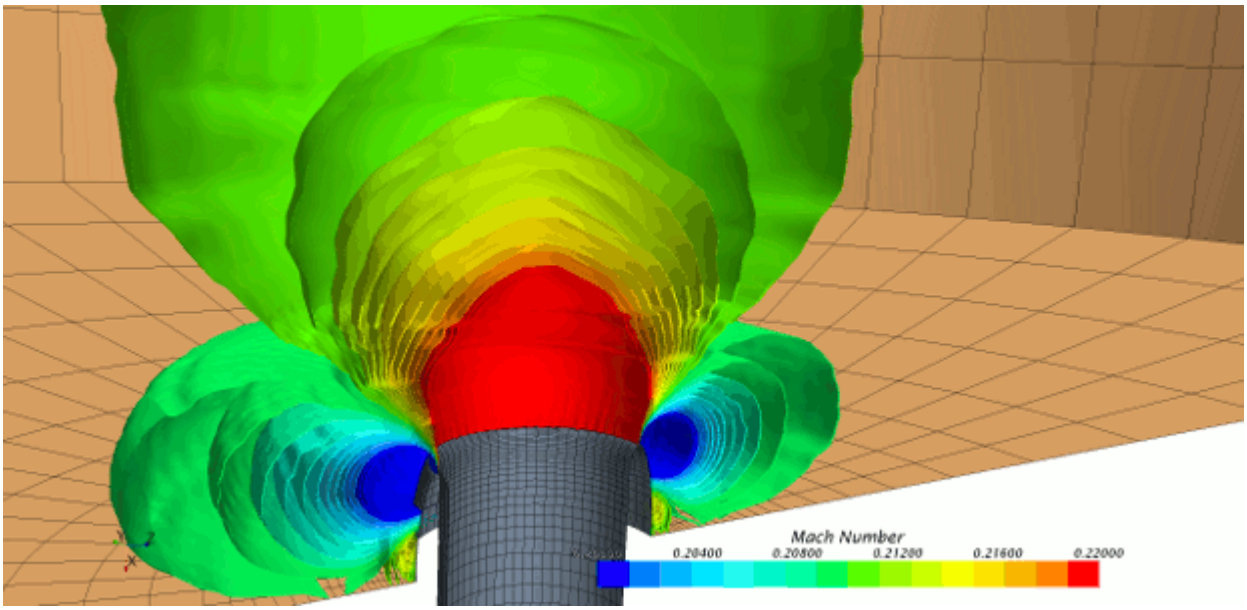


Displayer Opacity = 0.05

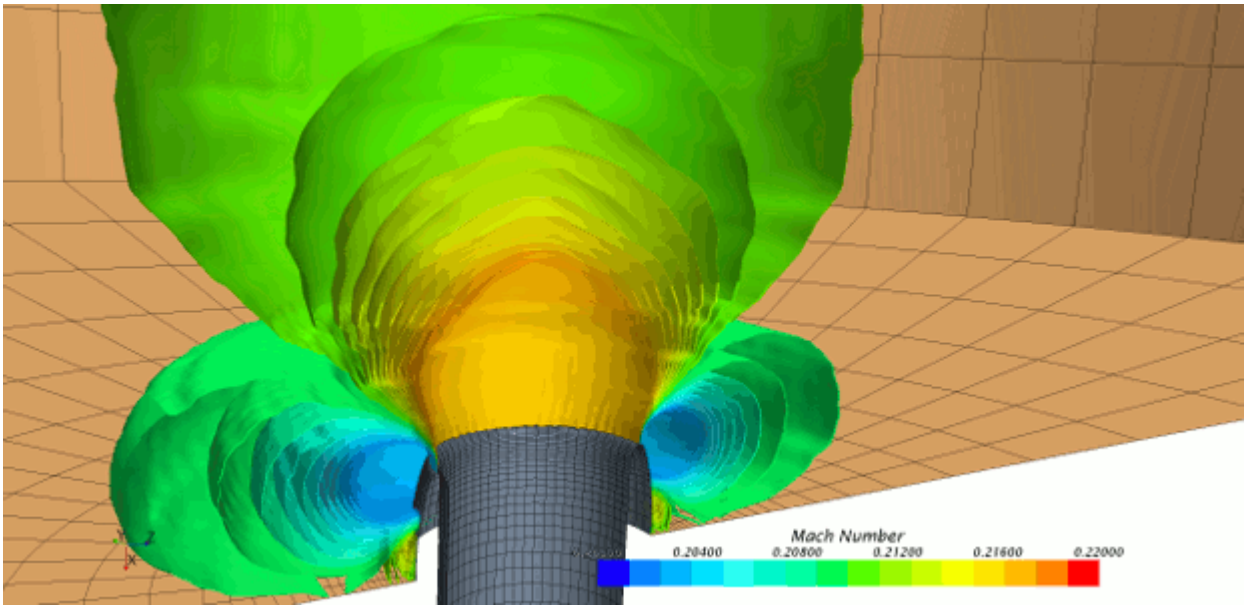


Displayer Opacity = 0.75

Colormaps:



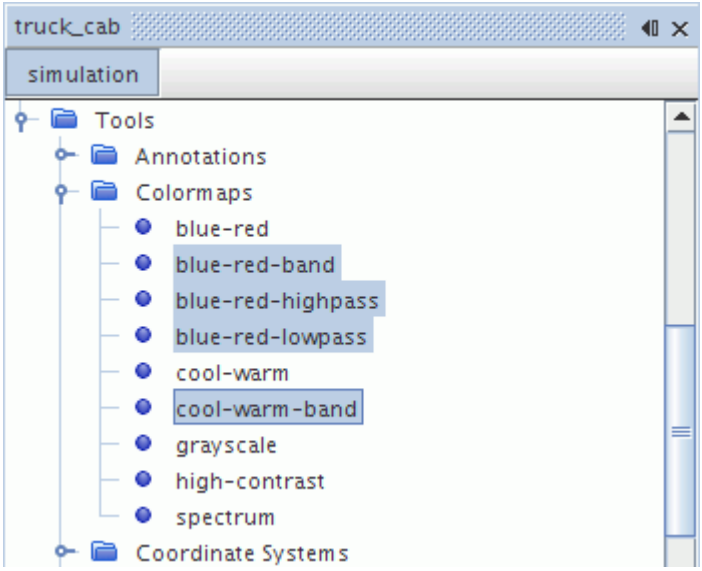
Blue-red (default)



Blue-red-band

Working with Volume Rendering Colormaps

Some colormaps are designed for use with volume rendering. These colormaps provide a mapping of the colored scalar to opacity.



If existing color maps are chosen for volume rendering, then they apply full opacity to all scalar values, visually resulting in a compact volume block. The volume rendering colormaps should not be used for coloring in other scalar displayers that show surfaces or streamlines.

The colormap types are as follows:

- Blue-red-band -- rainbow color map with the opacity decreasing from center towards minimum and maximum (band pass filter)
- Blue-red-highpass -- rainbow color map with the opacity increasing from minimum to maximum (high pass filter)
- Blue-red-lowpass -- rainbow color map with the opacity increasing from maximum to minimum (low pass filter)
- Cool-warm-band -- cool/warm color map with the opacity increasing from center towards minimum and maximum (band elimination filter)

Derived Parts Reference

Derived Parts Manager Right-Click Actions

New Part > [Type]

Creates a new derived part of the selected type, using a graphical editor when a visualization display is open. For a description of all the available types of derived part, see Accessing Solution Data through Derived Parts.

New Group

Facilitates the grouping of objects by creating a group node within the manager node.

Group By

Allows you to group the objects according to any of the properties of the objects in the manager node, which are listed in a submenu of this menu item.

Ungroup

Reverses the grouping process.

Derived Part Common Properties

The following properties are common to most types of derived parts.

Coordinate System

Applies to plane, constrained plane, cylinder, and sphere sections.

Specifies a list of coordinate systems in the simulation for use by the derived part.

Parts

Selects the input parts as the source for deriving the derived part.

Derived Part Common Right-Click Actions

The following right-click actions are common to most types of derived parts.

Edit Part in Current Scene

Allows you to set the properties of this part interactively. This option is only available if a scene is open. See Defining Derived Parts.

Export...

Exports the extracted data of the part to an external STL or VTK file. See Exporting to an External File.

Contents:

Specific Derived Parts Reference

Specific Derived Parts Reference

For information on the properties and actions that are common to all types of derived parts, see Derived Parts Reference. For information on the properties and actions that depend on the derived part type, see:

- Implicit Section Properties and Implicit Section Right-Click Actions
- Isosurface Properties
- Threshold Properties
- Scalar Warp Properties
- Vector Warp Properties
- Streamline and Constrained Streamline Properties
- (m', theta) Warp Properties
- Separation/Attachment Line Properties
- Vortex Core Properties
- Point Probe Properties
- Line Probe Properties
- Presentation Grid Properties
- Arbitrary Probe Properties and Actions
- Resampled Volume Properties

Implicit Section Properties

Section Mode

Defines whether the part includes a single or multiple sections.

Option	Activated Child Nodes
Single Section Specifies a single section (default), meaning that there is only one instance of the section within the derived part.	**Single section** Provides an *Offset* value for the implicit section, which allows you to offset the section from its original location. For planes, this value is the offset distance along the normal from the origin, and for cylinders or spheres, it the offset value for the radius.
Multiple Sections (range) Specifies a range of sections that are evenly distributed between the start and end offset section positions.	**Multiple Sections (range)** Provides the total *Number of Sections* to create, and the *Offset Range*, which is the offset position of the first and last sections. For planes, the offset position is the distance along the normal from the origin, and for cylinders or spheres, it is the offset value for the radius.
Multiple Sections (delta) Specifies a pattern of sections. This pattern uses the total number of sections and the	**Multiple Sections (delta)** Provides the total *Number of Sections* to create, the *Starting Offset*, that is the distance of the first section along the axis normal to the section, or the first offset radius for spheres or cylinders, and

incremental distance between each section.	the *Delta Value*, that specifies the incremental offset between successive sections.
Multiple Sections (arbitrary) Allows you to specify a series of sections that are based on individual offset values.	**Multiple Sections (arbitrary)** Specifies a series of sections using offset values that you enter in a comma-separated list. For example, setting *Offsets* to `[0.0, 0.01, 0.05, 0.09, 0.1]` creates 5 offset sections at the specified offset distances. For planes, the offset value is the distance along the normal from the origin, and for cylinders or spheres, it is the offset value for the radius.

Displayed Index

Controls the display of sections when a multiple mode is used. Enter either a positive integer to display the corresponding section, or a value of -1 to display all sections.

Origin

Plane Section/**Constrained Plane Section**: a Cartesian position vector that defines the origin of the plane. You can include units.

Cylinder Section: a Cartesian position vector that defines the origin of the cylinder axis. You can include units.

Sphere Section: a Cartesian position vector that defines the location of the center of the sphere. You can include units.

Normal

Plane Section/**Constrained Plane Section**: a Cartesian position vector that defines the second point through which the plane normal passes. The first point is the origin of the selected *Coordinate System*. You can include units.

Orientation

Cylinder Section: a Cartesian position vector that defines the second point through which the cylinder axis passes. The first point is the origin of the selected *Coordinate System*. You can include units.

Radius

Cylinder Section/**Sphere Section**: the radius of the cylinder or sphere section. You can include units.

Surface Input Option

Arbitrary Section: Specifies the source of the input geometry. The available options are **File**, that allows you to import arbitrary STL data, and **Part Surfaces**, that allows you to select one or more of the existing CAD part surfaces in the simulation.

Extract Approximate Faces

Arbitrary Section: when deactivated, the arbitrary section retains its original surface geometry and slices through the sample region. When activated, STAR-CCM+ creates an approximation of the surface that maps to the closest mesh cell faces. This option is faster because it does not have to compute the intersection between the section's surface geometry and the part or region.

Flip Orientation

Arbitrary Section: when activated, STAR-CCM+ reverses the directions of the normal vectors of the sample surface faces. The results become evident in reporting for relevant quantities such as mass flux. When deactivated, the face normals are right-hand oriented with respect to the counter-clockwise winding order of the original triangles in the section geometry.

Implicit Section Right-Click Actions

Re-Import Data...

Constrained Plane: Imports data from either an STL or VTK file. This option can be used to replace data that are imported during the creation of this derived part.

Export Loop...

Constrained Plane or **Arbitrary**: Exports the polygon bounding the constrained plane section to an external an STL file.

Isosurface Properties

Scalar Field

Specifies the scalar field function upon which the isosurface is based. See Field Functions.

Mode

Specifies whether a single isosurface or multiple isosurfaces are created.

Option	Activated Child Nodes
Single Allows you to create one isosurface corresponding to a single scalar value.	**Value** Within the properties of this node, you can set the *Isovalue*, that specifies the scalar value for the isosurface.
Multiple (range) Allows you to create multiple isosurfaces corresponding to a range of equally spaced scalar values.	**Values (range)** Within the properties of this node, you can specify the *Number of isosurfaces*, that is the total number of equally spaced isosurfaces in the range, and the *Isovalue Range*, which is the lowest and highest values in the range of isosurfaces.
Multiple (arbitrary) Allows you to create multiple isosurfaces corresponding to a set of scalar values that you specify with numeric entries.	**Values (arbitrary)** Within the properties of this node, you can specify the *Isovalues*, that is, a list of scalar values for the multiple isosurfaces. When you set this property by clicking ... (**Custom Editor**), the *Values (arbitrary) - Isovalues* dialog appears. In the dialog, you can click **Query** to select a scalar range automatically. You can also click **Auto Generate Values** to select the isovalues automatically.

Threshold Properties

Scalar Field

Specifies the field function that is used to determine the threshold range. See Field Functions.

Range

Specifies the minimum and maximum of the threshold range.

Mode

Specifies how to set the threshold with respect to the specified scalar range:

- **Between**: the threshold encompasses cells with all values between the minimum and the maximum of the *Range*.
- **Above Max**: the threshold encompasses cells with values greater than the maximum of the *Range*.
- **Below Min**: the threshold encompasses cells with values less than the minimum of the *Range*.
- **Outside**: the threshold encompasses cells with values less than the minimum and greater than the maximum of the *Range*.

Scalar Warp Properties

Scalar Field

Specifies the scalar field function upon which the warp surface is based. See Field Functions.

Scale Factor

Specifies the scale factor for the warp surface.

Vector Warp Properties

Vector Field

Specifies the vector field function upon which the warp surface is based. See Field Functions.

Relative or Absolute Coordinates

Lets you control the "base" vector, the original position or the updated position:

- **Relative**: refers to the updated position.
- **Absolute**: refers to the original position.

Scale Factor

Specifies the scale factor for the warp surface.

In finite element (FE) solid stress applications, use this feature for very small deformations that need to be exaggerated for visualization. For example, if the maximum magnitude of the displacement vector is less than 5% of the maximum model dimension, scale the warp to 20%.

Auto-scaling

When activated, automatically calculates the scale factor based on the sizes of the vector warp and its input parts.

This option is especially useful with visualization of very small and very large warps. When a vector warp is too small to see, STAR-CCM+ will calculate an increased scale factor to make the deformation visible. Also, for very large warps, a smaller scale factor is calculated.

Auto-scale Value

When *Auto-scaling* is activated, displays the automatically calculated value of the scale factor. Otherwise the displayed value is `0.0`.

Streamline and Constrained Streamline Properties

Vector Field

Selects the type of vector field function to show, such as velocity or flow direction. See Field Functions.

Seed Type

Specifies how the seed point is clustered and defined:

Option	Activated Child Nodes
Point Uses a single point or a cloud of points as the seed. Use this option only when you have a particular set of points, which you could obtain by cutting a surface with a plane or drawing an isosurface.	**Point Seed** Allows you to define the point seed by setting the following properties: • *Center*: allows you to specify the position of the point cloud. • *Radius*: specifies the radius of the point cloud. It should remain at the default value of zero if you use only one point. The units can be changed by typing them in after the values. • *Resolution*: specifies the number of points in the point cloud.
Line Uses a line of points as the seed. As with the point seed, you should have a specific set of points to use this option. Moreover, the line seed would be effective only on a perfectly flat surface.	**Line Seed** Allows you to define the line seed by setting the following properties: • *Line Point 1*: allows you to specify the position of one end point of the line. • *Line Point 2*: allows you to specify the position of the other end point of the line. • *Resolution*: specifies the number of evenly spaced points along the line.
Part Uses an existing surface, line, or point part as the seed. Regions cannot be selected.	**Source Seed** Allows you to define the seed part by setting the following properties:

The part (source) seed takes its points from one or more existing parts, therefore it is certain to create a streamline or constrained streamline that lies directly on the part surfaces. This type of seed is the recommended option.

When using a part seed, you can use the in-place dialog to specify the *Part U-Resolution* and *Part V-Resolution*. These properties allow you to specify the maximum number of points extracted from the part in the u- and v- directions. See Defining a Streamline.

Note For a part seed, the chosen part must exist within the same representation as the region containing the solution through which STAR-CCM+ integrates the streamline. As such, you cannot use geometry part surfaces for the seed part, as the geometry representation has no solution data.

- *Seed Parts*: allows you to select the parts to be used as the seed. Regions are not included among the options.
- *On Ratio*: the ratio of available vertices to the maximum numbers of seed points specified for the u- and v-directions. This value is automatically calculated by STAR-CCM+. It is recalculated each time you change the u or v points or the *Seed Parts* property.
- *Randomize*: when activated, a random pattern of points is used for the seed points. When deactivated, the vertices of the parts are used for the seed points. It is useful to randomize if the *Part* uses a highly unstructured grid or mixture of fine and coarse grids.
- *N Grid Points*: specifies the maximum numbers of seed points to use in the u- and v- directions. If the number of vertices on the part is less than either value, the number of vertices is used.

Integration Solver

Allows you to select the Runge-Kutta integration solver. The available options are **2nd-Order RK**, **4th-Order RK**, and **Adaptive-Step RK**, which activate corresponding child nodes: **2nd Order Integrator**, **4th Order Integrator**, and **Adaptive Integrator**. You can define the integration process by setting the properties of the integrator child nodes:

- *Integration Direction*: specifies the direction of integration from the seed location. The available options are:

  Forward: the integration proceeds in the forward direction (for instance with the flow if velocity is the vector field).
  Backward: the integration proceeds in the backward direction (for instance against the flow if velocity is the vector field).
  Both: the integration proceeds both forward and backward from the seed location.
- *Initial Integration Step*: specifies the initial integration step size in units of cell length. Reducing the value creates a higher resolution for the streamlines.
- *Maximum Error*: specifies the initial integration step size in units of cell length.
- *Maximum Integration Step*: specifies the maximum integration step size in units of cell length.
- *Maximum Propagation*: specifies the maximum length of the streamlines in units of cell length.
- *Max Steps*: specifies the maximum number of integration steps.

Rotation Scale

The scale factor for twist.

(m', theta) Warp Properties

Rotation Origin

Sets the rotation origin for an (m', theta) warp.

Rotation Axis

Sets the rotation axis for an (m', theta) warp.

Tangential Axis

Sets the tangential axis for an (m’, theta) warp.

Separation/Attachment Line Properties

Display Attachment Lines

Toggles whether lines displaying flow attachment are displayed or not.

Display Separation Lines

Toggles whether lines displaying flow separation are displayed or not.

Vector Field

Specifies the vector field that is used by the algorithm to extract the separation/attachment line. This must be set to wall shear stress or cell relative velocity. If you use a different three-dimensional vector field, it does not produce valid results.

Line Quality

Filters lines where the angle between the extracted separation/attachment line and the flow vector are below the user-defined value. The quality is entered on a scale of 0.0 to 90.0 degrees.

Line Strength

Sets the strength of the separation/attachment. This setting is used to filter out weak results. Line strength is not a physical quantity, but a filtering property. The strength is entered on a scale of 0–1, where 0 includes all separation/attachments, and 1 includes only the strongest results.

Minimum Line Length

Sets the minimum separation/attachment line length. Separation/attachment lines which are shorter than this value are eliminated.

Vortex Core Properties

Vortex Quality

Computes the angle between the extracted core line and the velocity vector at this position. Small angles mean high quality; this may not apply to all simulation data.

Vortex Strength

Sets the strength of the rotational motion. This setting is used to filter out weak vortices.

Minimum Core Length

Sets the minimum length of a line to be used by a vortex core. Below this setting, the core lines are eliminated.

Vector Field

Specifies the vector field that is used by the algorithm to extract the vortex core. This must be set to velocity. If you use a different three-dimensional vector field, this would not lead to a valid result.

Point Probe Properties

Point

Specifies a Cartesian position vector that defines the location of the point.

Follow Motion

Controls whether the point probe moves together with a moving mesh or not. When activated, the point probe attaches itself to the moving mesh and keeps track of the cell it is in and its relative position within that cell. Every time the mesh moves, the point checks its position to ensure that it is inside the correct cell and in the correct position within that cell. When deactivated, the point probe remains stationary (assuming that its assigned coordinate system is not moving). This option works with all types of motion.

Line Probe Properties

Point 1

Specifies a Cartesian position vector defining one end of the line.

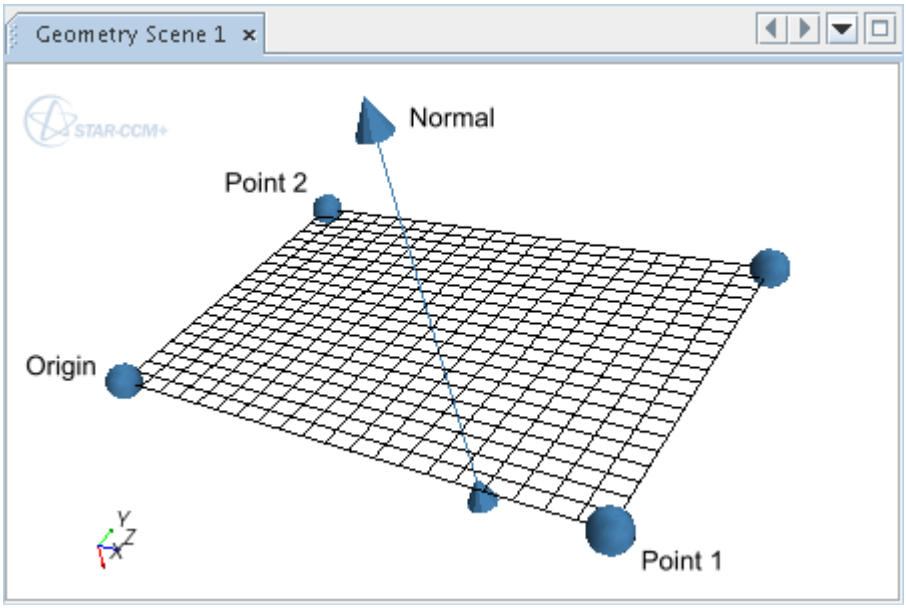
Point 2

Specifies a Cartesian position vector defining the other end of the line.

Resolution

Specifies the number of evenly spaced segments along the line. The actual number of points in the probe is one greater than the number of segments; fewer points may be generated if part of the probe lies outside the model geometry.

Presentation Grid Properties



Normal

Specifies the vector passing through the center of the presentation grid at a right-angle. Entering a new value for *Normal* automatically adjusts the values for *Point 1*, *Point 2* and *Origin*, in order to define a grid that matches the current properties.. Use this property as an alternative to *Point 1* and *Point 2* to define the plane. You can include units in the entry, or select the units in the dialog that is activated by clicking ... (**Custom Editor**).

Origin

Specifies the lower-left corner on the presentation grid relative to the selected coordinate system. Entering a new value for *Origin* automatically adjusts the *Normal*, in order to define a grid that matches the current properties. You can include units in the entry, or select the units in the dialog that is activated by clicking ... (**Custom Editor**).

Point 1

Specifies the lower-right corner on the presentation grid, relative to the selected coordinate system. Entering a value for *Point 1* automatically adjusts the *Normal*, in order to define a grid that matches the current properties. Use this property along with *Point 2* as an alternative to *Normal* to define the plane. You can include units in the entry, or select the units in the dialog that is activated by clicking ... (**Custom Editor**).

Point 2

Specifies the upper-left corner on the presentation grid, relative to the selected coordinate system. Entering a new value for *Point 2* automatically adjusts the *Normal*, in order to define a grid that matches the current properties. Use this property along with *Point 1* as an alternative to *Normal* to define the plane. You can include units in the entry, or select the units in the dialog that is activated by clicking ... (**Custom Editor**).

X Resolution

Specifies the number of intervals between data points along the x-axis (that is, Number of Data Points = X Resolution + 1). The x-axis is the axis between the *Origin* and *Point 1*.

Y Resolution

Specifies the number of intervals between data points along the y-axis. The y-axis is the axis between the *Origin* and *Point 2*.

Note: The *Normal*, *Point 1*, *Point 2* and *Origin* properties override one another according to the most recently entered value.

Arbitrary Probe Properties and Actions

Surface Input Option

Specifies the source of the input geometry. The available options are:

- **File**: allows you to read in points from a STL or VTK file.
- **Part Surfaces**: allows you to select one or more CAD part surfaces from which to extract the sample points.

Re-Import Data... **(right-click action)**

Allows you to load new sample points from an STL or VTK file.

Resampled Volume Properties

Origin

A Cartesian position vector that defines the origin of the resampled volume. You can include units in the entry, or select the units in the dialog that is activated by clicking ... (**Custom Editor**).

Rotation Axis

Sets the rotation axis for the resampled volume.

Size

Sets the size of the resampled volume in three dimensions. You can include units in the entry, or select the units in the dialog that is activated by clicking ... (**Custom Editor**).

Rotation Angle

Specifies the angle of rotation along the specified rotation axis in radians.

Cell Size to Voxel Ratio

Determines how fine the resulting structured grid is. Increasing this value increases the number of voxels that represent the resampled volume. As this value represents the ratio between the cell size and voxel size, specify a value between 0 and 1.

Minimum Voxel Size

Sets the minimum size of a structured cell. Increasing this value decreases the number of voxels that represent the resampled volume. The *Minimum Voxel Size* has precedence over the *Cell Size to Voxel Ratio*.

Volume Rendering Settings Properties

Rendering Mode

Allows you to select a mode of rendering. The following options are available:

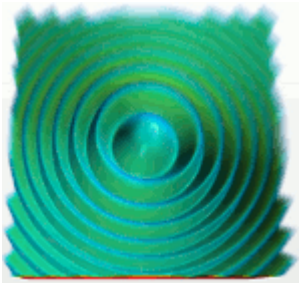
Direct - Every scalar sample value is mapped to the displayer opacity plus color, and is then attenuated along the view direction.
Maximum Opacity Projection - A transparent mode with high contrast to highlight volume areas that contain scalars with high opacity. An example is when velocity magnitude is mapped to a high-pass color map. The transparent mode reveals additional points of interest, regardless of the opacity attenuation which is depending on the viewpoint. However, the projective nature of this mode removes clues for depth perception.
Level Surface - Allows you to choose between single or multiple surfaces. Activates a Level Surface Mode sub-node.

Lighting Mode

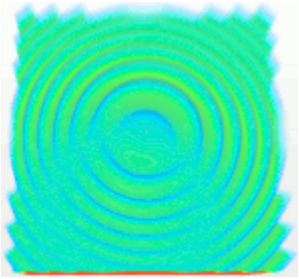
Allows you to apply displayer lighting to the volume rendering. The primary application for the Local Lighting mode is when the *Rendering Mode* property is set to `Level Surface`. In the Level Surface mode, standard lighting will be applied to the level surface exactly as if it were a proper surface. Additionally, local lighting is used in the Direct rendering mode for a more “surface-like” image without actually reverting to simple surfaces.

None - Does not use displayer lighting.
Local Lighting - Applies the settings of the displayer surface lighting. The advantage of this technique is that details in the scalar field can be discerned more easily in many cases. The following Marschner-Lobb examples of volume rendering illustrate the benefit of local lighting.

- Local lighting:



- No local lighting:



Quality

Determines how accurately the resampled volume is processed during ray-casting.

Volume Rendering Level Surface Mode Properties

Level Surface Mode

The mode for level surface volume rendering, specifying whether a single surface or multiple surfaces are created.

Single - Uses one surface (default).
Multiple (range) - Selects a range technique for creating multiple surfaces.
Multiple (arbitrary) - Selects an arbitrary technique for creating multiple surfaces.

Derived Parts Bibliography

[183] *Drela, M. and Youngren, H. 2008. "A User's Guide to MISES 2.63", MIT Aerospace Computational Design Laboratory.*

[184] *Haimes, R. and Kenwright, D. N. 1999. "On the velocity gradient tensor and fluid feature extraction", AIAA 14th Computational Fluid Dynamics Conference, Paper 99-3288.*

[185] *Peikert, R. and Roth, M. 1999. "The 'parallel vectors' operator -- a vector field visualization primitive", Proceedings of the Conference on Visualization '99, San Francisco, California, USA, pp. 263–270.*

[186] *Roth, M. 2000. "Automatic Extraction of Vortex Core Lines and Other Line-Type Features for Scientific Visualization", PhD thesis, Swiss Federal Institute of Technology, Zurich, Switzerland.*

[187] *Roth, M. and Peikert, R. 1998. "A higher-order method for finding vortex core lines", Proceedings of the Conference on Visualization '98, IEEE Computer Society Press, pp. 143–150.*

[188] *Sujudi, D. and Haimes, R. 1995. "Identification of swirling flow in 3-D vector fields", AIAA 12th Computational Fluid Dynamics Conference, Paper 95-1715.*

Visualizing the Solution

The visualization tools in STAR-CCM+ are integrated with the analysis, allowing for the interactive extraction and viewing of solution data from either a running or converged simulation.

It is possible to watch a flow field evolve as the simulation iterates, change parameters and immediately see the effects of those changes. This level of feedback allows for the dynamic supervision and control of the simulation, as well as allows you to gain insight on a particular aspect of the simulation. STAR-CCM+ provides identical visualization capabilities in both serial and parallel modes, with no additional user intervention. Since visualization in STAR-CCM+ operates in a client-server environment, the bulk of data processing takes place on the server process. Only the lightweight graphics data is sent to the client, allowing for a massively parallel simulation to be visualized from a client workstation.

In STAR-CCM+, a scene encapsulates the complete representation of the components that are required to generate an image or animation including: lights, cameras, parts, properties, transformations, geometry, textures, and other pertinent information.

To assist in managing the collection of objects in a scene, STAR-CCM+ provides another high-level object that is called a displayer. A displayer brings together parts and other attributes (such as scalars, vectors, and surface properties) to produce graphics primitives which are displayed in the scene. In order to construct visualizations it is important to understand the concepts of both scenes and displayers and how they interact. A scene can be comprised of one or more displayers, allowing for a flexible way to view parts with various different attributes. For example, a scene could have two displayers, one showing a part in outline mode, and another showing the same part with a scalar field.

Contents:

Working with Scenes
Changing the Attributes of a Scene
Displaying Geometry and Mesh
Displaying Scalar Quantities
Displaying Vector Quantities
Displaying Streamlines
Displaying Dynamic Fluid Body Interaction (DFBI)
Animating Scenes
Using Displayers
Advanced Rendering
Saving and Printing Scene Data
Controlling Display Formats
Annotations
Color Maps

Using Transforms
Working with Views
Obtaining Benchmark Information for a Scene
Linking Scene Views

Working with Scenes

This section defines what is meant by a scene, and describes how to use scenes to visualize the results of a simulation.

The special scenes available in STAR-CCM+ for visualizing models that are created using 3D-CAD are described in Visualizing A 3D CAD Model.

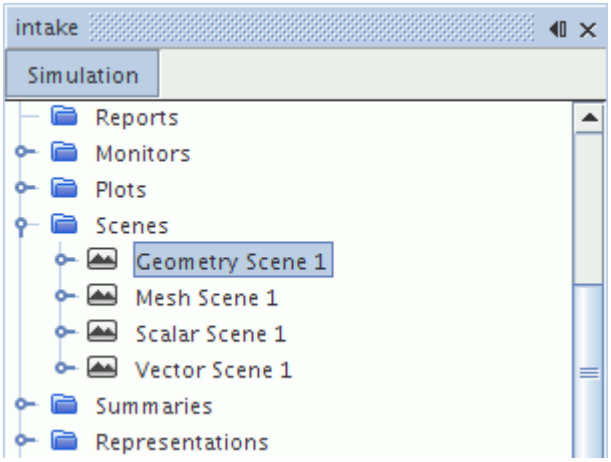
Contents:

What Is a Scene?
Creating a Scene
Interacting with the Scene
Managing Parts Through a Scene
Adding Extra Displayers
Removing Displayers
Copying Scenes
Managing Scenes

What Is a Scene?

A scene allows you to visualize the model.

The scene node, which has its own properties and pop-up menu, defines annotations and coordinate systems to be displayed, and serves as a manager node for additional objects that control various aspects of that scene.



A scene also manages the view. Using the mouse and keyboard, you can manipulate the scene view, select parts and other objects and interact with them.

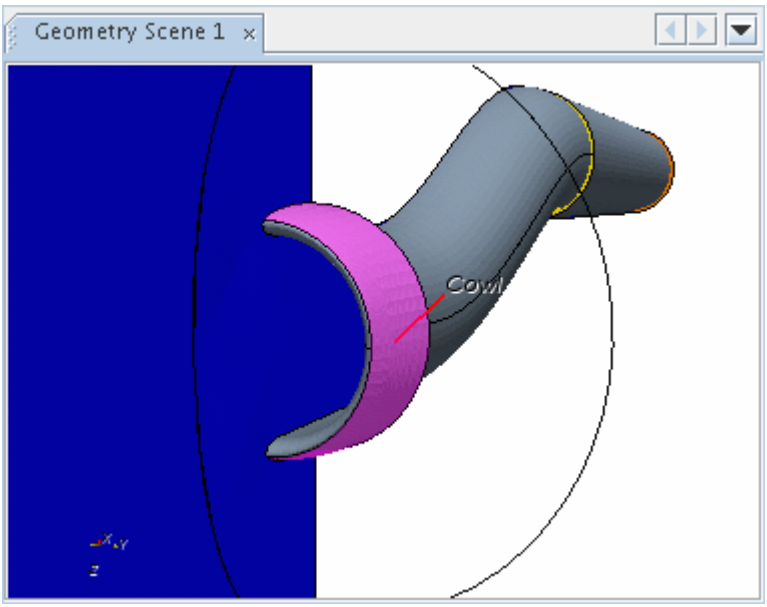
If the scene is open, all scenes are continuously updated as the simulation changes for the duration of the session.

Scenes are given default names when they are created. However they can be renamed at any time using any of the standard methods, as well as from within the visualization display. If multiple simulations are open at once, then the tab includes the name of the corresponding simulation. That part of the tab name cannot be changed unless you save a copy of the simulation with a new name.

Scenes can also be copied and deleted.

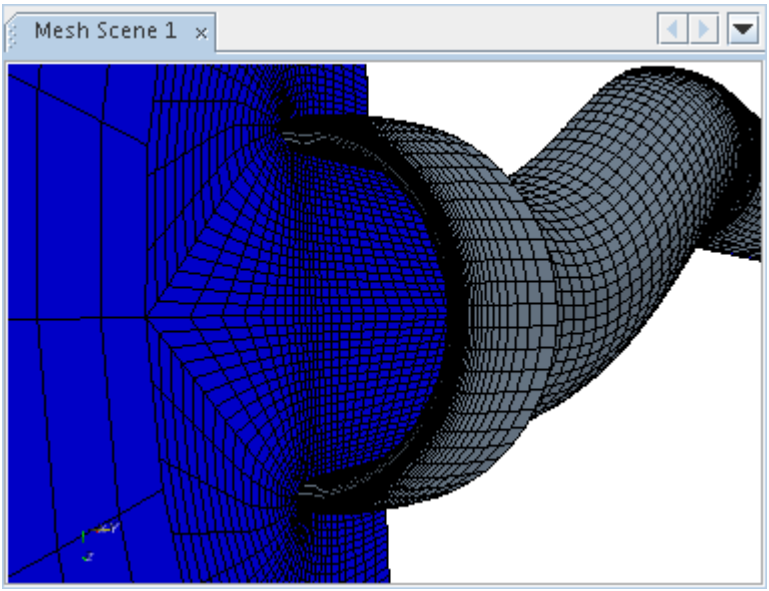
STAR-CCM+ offers five default scene templates:

- Geometry—for viewing surface geometry



  By default a geometry scene includes:

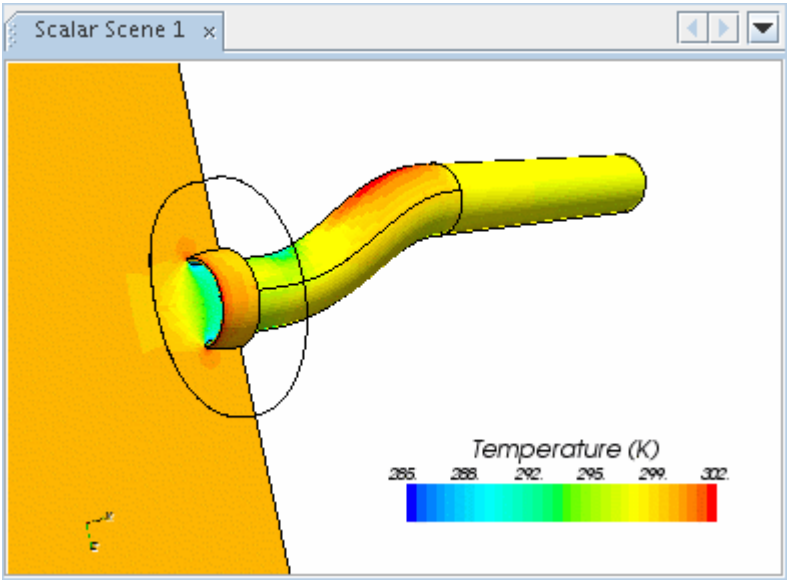
  - A geometry displayer, which contains all boundaries in the simulation and is preset to display the outlines of the parts
  - A second geometry displayer, which contains all boundary parts in the simulation and is preset to display the shaded surface
- Mesh—for viewing mesh grid lines and surfaces



By default a mesh scene includes:

- A geometry displayer, which contains all boundaries in the simulation and is preset to display the outlines of the parts
- A second geometry displayer, which contains all boundary parts in the simulation and is preset to display the mesh cells on the shaded surface

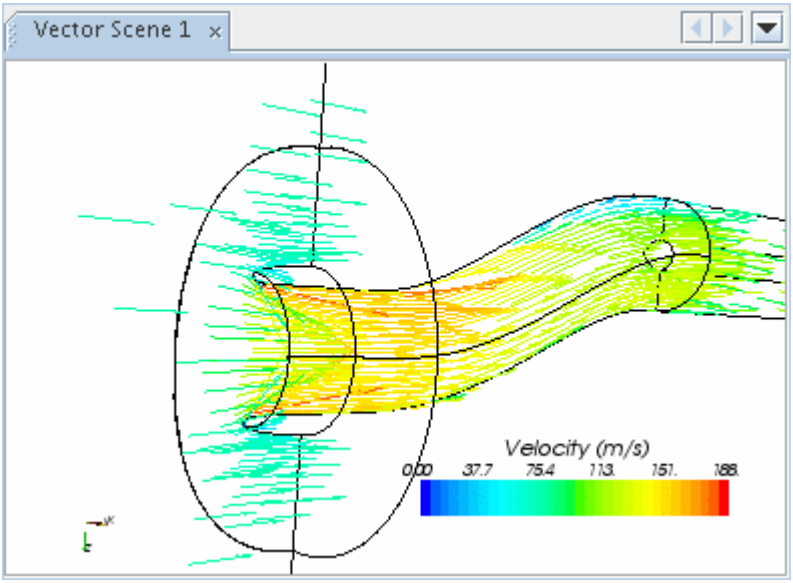
- Scalar—for viewing scalar fields



By default a scalar scene includes:

- A geometry displayer, which contains all boundaries in the simulation and is preset to display the outlines of the parts
- A scalar displayer, which contains an undefined scalar field

- Vector—for viewing vector fields



By default a vector scene includes:

  - A geometry displayer, which contains all boundaries in the simulation and is preset to display the outlines of the parts
  - A vector displayer, which contains a vector field (set to velocity by default)

- Empty—a blank canvas on which you can start from scratch. Each new displayer contains no parts, and is set to the default settings. Alternatively, you can add parts directly using the drag-and-drop technique.

Contents:

Scene Properties
Scene Pop-Up Menu

See Also:

Working with Displays

Scene Properties

Use the scene properties to control characteristics such as transparency and mesh display.

Transparency Override	Overrides displayer settings for opacity in the scene.	
	Make Scene Transparent	Makes all parts transparent so that you can see all of them from any angle, and overrides the opacity settings of all the displayers in the scene.

	Make Scene Opaque	Makes all parts in the scene completely opaque, and overrides the opacity settings of all the displayers in the scene.
	Use Displayer Property	Reverts to the individual displayer settings for opacity.
Mesh Override	Overrides displayer settings for displaying the mesh in the scene.	
	Show All Meshes	Adds mesh lines, and overrides the mesh display settings of all the displayers in the scene.
	Hide All Meshes	Removes mesh lines, and overrides the mesh display settings of all the displayers in the scene.
	Use Displayer Property	Reverts to the individual displayer settings for mesh display.
Coordinate Systems	A list of the coordinate systems added to the scene. Editing the property accesses the object selection dialog to add or remove coordinate systems.	

Scene Expert Properties

Aspect Ratio	In the drop-down list, **Fit Screen** is the default. Selection of any of the following aspect ratios can leave blank space in the graphics display: • **16:9** • **16:10** • **4:3** • **1:1** Use this property to preview the effect on a particular aspect ratio before proceeding to export high-definition hardcopies or movies. See Saving Scene Image Files.	
Orientation	In the drop-down list, choose between **Landscape** (default) and **Portrait**.	
Enable Advanced Rendering	When activated, applies advanced rendering to this scene. For details, see Advanced Rendering.	
Transparency Mode	Provides a choice between visual accuracy and speed of rendering of transparent images.	
	Alpha Blending	Renders the image faster.
	Depth Peeling	Provides a more accurate transparency.
Width	Width of the scene display (read-only).	

Height	Height of the scene display (read-only).

Scene Pop-Up Menu

The pop-up menu of the scene includes refreshing the display and exporting the image to a graphics file.

Open	Executes the open scene action to make the associated visualization display visible. See Interacting with the Scene.
Refresh	Executes the refresh display action.
Restore Hidden Part	Executes the restore hidden part action to redisplay a part that is hidden by the hide part action. You choose which part to restore from a submenu of this menu item. See Restoring a Hidden Part in a Scene.
Restore All Hidden Parts	Executes the restore hidden parts action to redisplay all parts that is hidden by previous hide part actions. See Restoring All Hidden Parts in a Scene.
Hardcopy...	Executes the save scene to file action to create an image file of the current scene. See Saving Scene Image Files.
Report	Executes the test graphics action to display graphics hardware-related information in the *Output* window.

Creating a Scene

Each default scene template contains a specific combination of displayers with certain properties set, and with a default set of parts, all of which you can change.

To create a scene:

1. Right-click the **Scenes** node.
2. In the context menu, select **New Scene** and one of the options in the submenu.

   The new scene appears as a node in the object tree and as a window in the graphics display.

Interacting with the Scene

A scene also provides simple but powerful ways to manipulate the image.

The main features are discussed in the Using STAR-CCM+ section; however, they include the ability to change the view dynamically by panning, rotating, and zooming the model.

In addition to simple orientation manipulations you can also change the properties of the scene, as well as manipulate the components of the model within the scene using the right-click functionality. For example, you can manipulate the regions and boundaries of the model from within the scene without having to revert to the tree. This provides a powerful visual aid to model manipulations.

The visualization toolbar at the top of the workspace also provides easy access to common functionality, including a back and forward button between recent views, and easy access to a transparency mode.

Any type of scene can be opened by double-clicking its node in the simulation tree, or by right-clicking that node and selecting **Open**. Either operation shows the selected scene in a new display in the *Graphics* window, or brings that display to the foreground if the scene is already open. A scene is opened with the view set to the last setting used, or with all parts fitted in the display looking down the Z axis with the Y axis up if it has never been opened before.

When more than one simulation is open, the tab at the top of the display includes the name of the corresponding simulation. This lets you know at a glance which simulation a display represents.

See Also:

Working with Displays

Managing Parts Through a Scene

You can use the pop-up menu to work with parts in a scene.

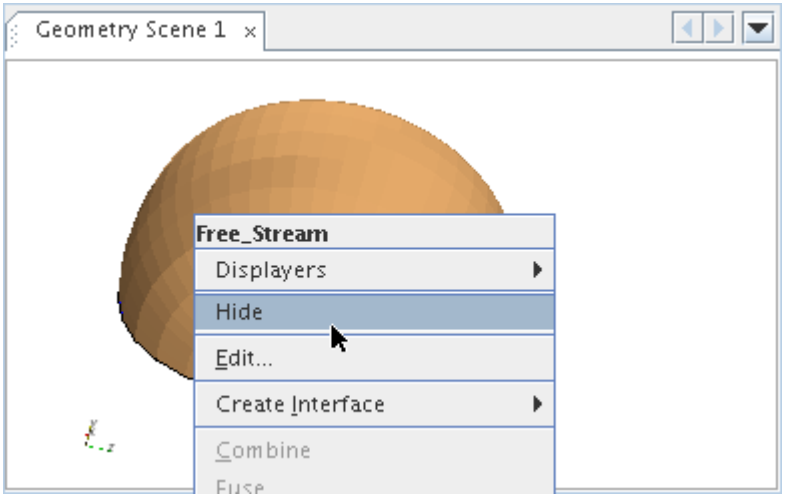
Most of the work you do with parts in a scene, including adding and removing them, is through individual displayers. However, the following actions are possible with pop-up menu items of scenes:

Contents:

Hiding a Part in a Scene
Hiding All Parts in a Scene
Restoring a Hidden Part in a Scene
Restoring All Hidden Parts in a Scene

Hiding a Part in a Scene

Hiding a part in a scene can be achieved by right-clicking on the part in the visualization display and selecting **Hide** from the pop-up menu.



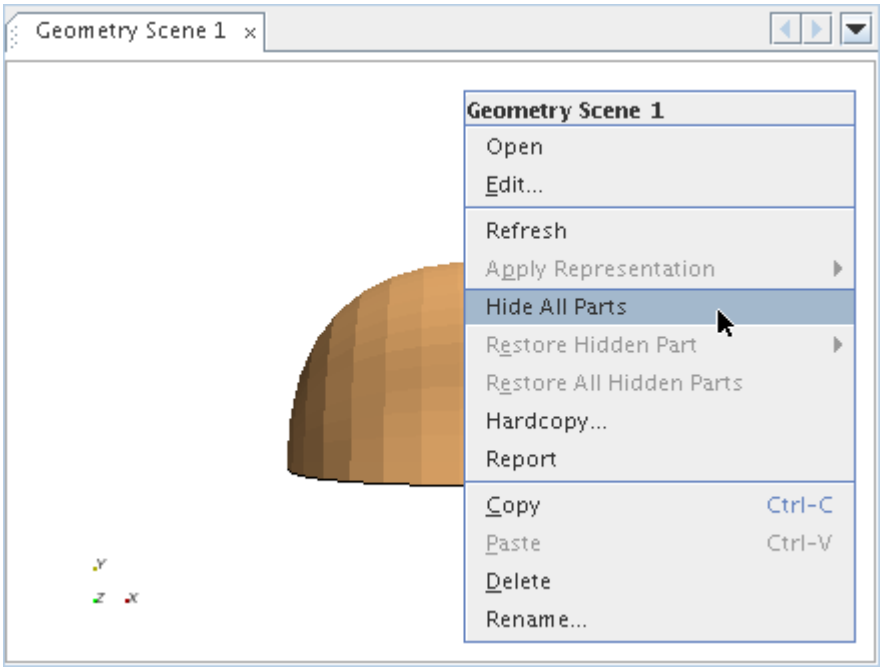
This pop-up menu item is also available if you right-click a displayer part node in the object tree.

See Also:

Using Scenes to Interact with the Object Tree

Hiding All Parts in a Scene

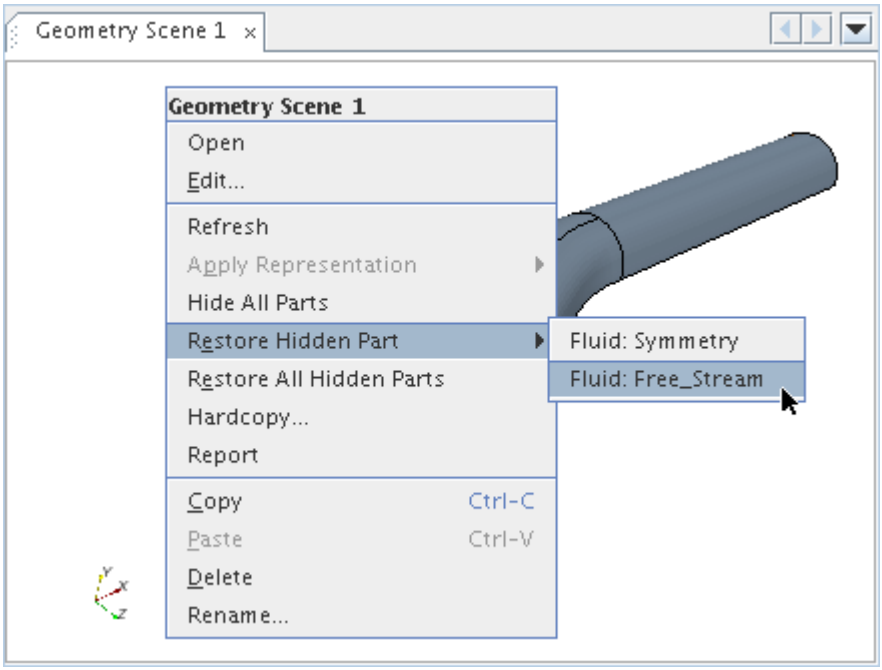
You can hide all the parts of a scene at once by right-clicking in the visualization display background and selecting **Hide All Parts** from the pop-up menu.



This pop-up menu item is also available if you right-click the scene node or the displayer Parts manager node in the object tree.

Restoring a Hidden Part in a Scene

To restore a hidden part, right-click in the visualization display background, select **Restore Hidden Part** from the pop-up menu, and then select the part from the submenu that appears.

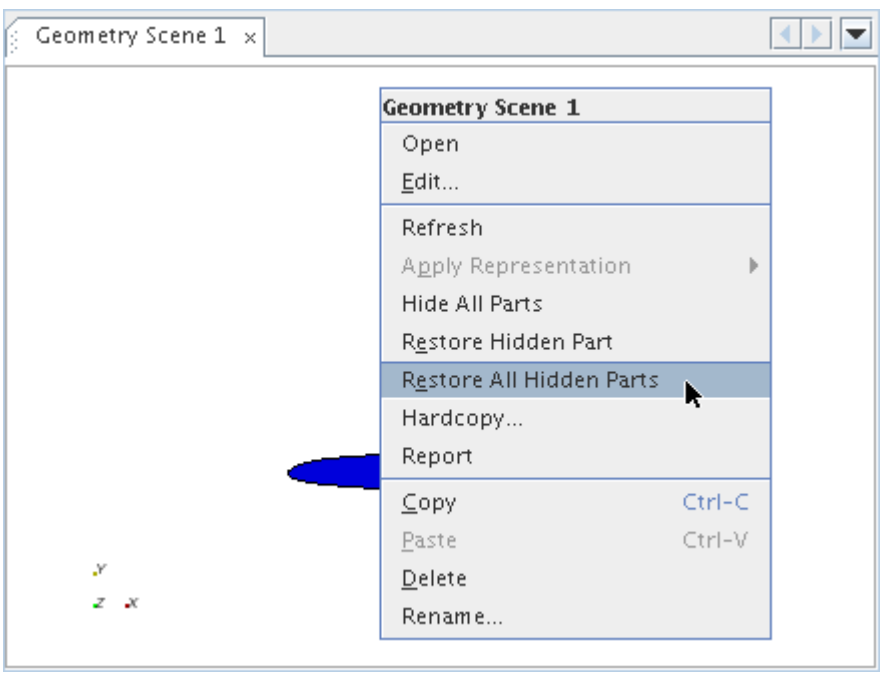


You can also find this pop-up menu item by right-clicking the scene node or the displayer Parts manager node in the object tree.

This menu item is disabled (grayed out) if no parts in the scene are hidden.

Restoring All Hidden Parts in a Scene

To restore all hidden parts at once, right-click in the visualization display background and select **Restore All Hidden Parts** from the pop-up menu.



You can also find this pop-up menu item by right-clicking the scene node or the displayer Parts manager node in the object tree.

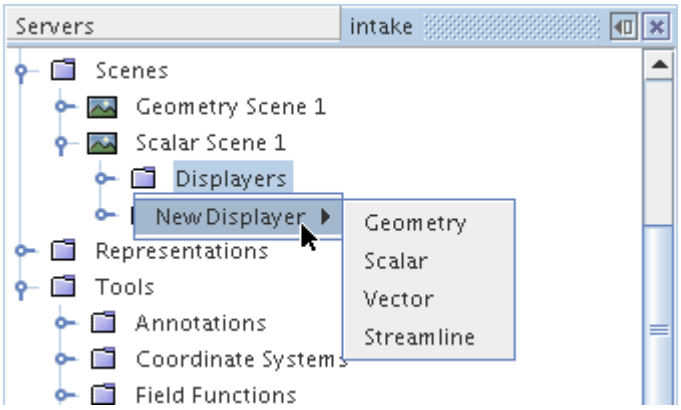
This menu item is disabled (grayed out) if no parts in the scene are hidden.

Adding Extra Displayers

In addition to the default displayers, it is also possible to add displayers to scenes.

Except for empty scenes, by default scenes are created with two displayers. The first displayer adds the mesh outline of all the boundaries. The second displayer depends upon the type of scene. For example, a scalar scene has a scalar displayer with no default parts, on which is displayed an undefined scalar.

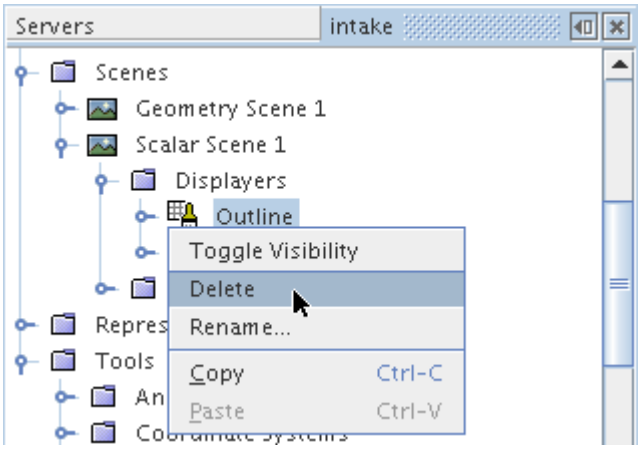
To add a displayer, expand the **Scenes** node and the node of the chosen scene. Right-click on the **Displayers** node, select **New Displayer** from the pop-up menu and choose the type of displayer from the submenu.



Any new displayers added to the scene has default properties. The new displayer is named `Displayer N` where `Displayer` maps to the type of displayer added, and `N` is a numeric suffix to ensure that all displayers are uniquely named. All new displayers added with this action have an empty *Parts* manager, a new scalar displayer and streamline displayer have an undefined scalar field node, and a vector displayer has a vector field node set to velocity by default.

Removing Displayers

To remove a displayer from a scene, simply right-click on the displayer node you wish to remove and select the **Delete** action.



As a result, the displayer object and all associated graphics entities are removed from the scene.

Copying Scenes

When a scene is copied, all displayers and properties are copied, resulting in an exact duplicate of the original.

In addition to duplicating content, this copy action captures the open/closed state of the original: copy an open scene and the duplicate opens, copy a closed scene and the duplicate stays closed.

This process differs from copying properties of a scene.

Contents:

Copying Properties of a Scene

Copying Properties of a Scene

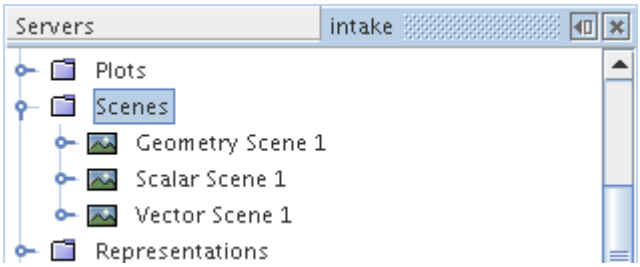
You can copy the properties of a scene, though certain limitations exist.

When properties are copied from one scene to another, a new set of displayers appears in the destination scene alongside the originals. Those new displayers match the ones in the source scene. Applying the displayers' properties requires copying those properties separately. See Copying Properties of a Displayer.

Managing Scenes

STAR-CCM+ provides for easy management of scenes.

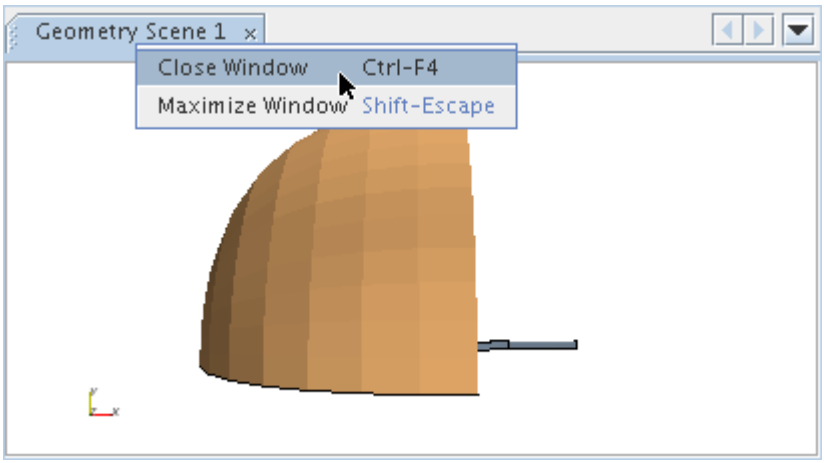
The **Scenes** node, which has a pop-up menu, is the manager object for working with scenes in the simulation tree.



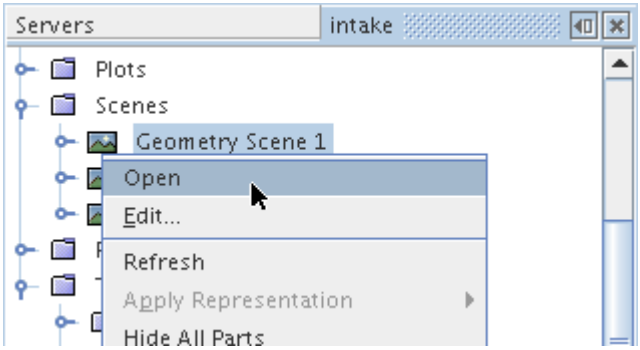
It exists even when empty to allow creation of the first scene. The scenes manager can hold any number of scenes up to the memory limits of the computer.

When a scene is created or opened, it creates a visualization display. The scene is docked in the workspace in the same way as any other display, using a tabbed layout. When there are multiple scenes present, you can select which scene is active by selecting the appropriate tab. If multiple simulations are open at once, then the tab includes the name of the corresponding simulation.

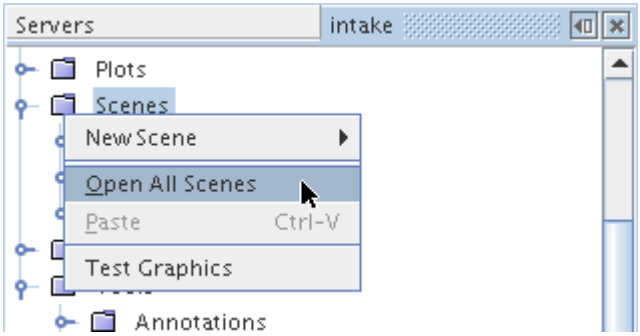
A scene can be closed by clicking the **X** button on the visualization display tab, or by right-clicking the tab and selecting **Close Window** from the pop-up menu. When a scene is closed, it does not continuously update during the iteration.



To open a scene expand the **Scenes** node, right-click on the scene and select **Open** from the pop-up menu.



To open all scenes at once, right-click on the **Scenes** node and select **Open All Scenes**.



A scene can be maximized to occupy the entire workspace by double-clicking its tab, and minimized by double-clicking the tab when in the maximized state. These operations are also available as a right-click action on the tab.

Using drag-and-drop, scenes can be moved anywhere in the workspace. To move a scene, click the scene's header or tab and drag it to the desired location. A red box shows where the scene is placed when the scene is dropped. You can do any of the following:

- Split any open scene. Drag the scene tab to any of the edges of the open scene. When the red box shows a split in the open scene, drop the scene.
- Move a scene from one open window to another. Drag the scene tab over the open tabs of the destination window. When the red box shows a tab in the new window, drop the scene.
- Arrange the order of the scenes within a window. Drag the scene tab to any location in the window's tabs and drop it.

Contents:

Pop-Up Menu of the Scenes Manager

Pop-Up Menu of the Scenes Manager

These pop-up menu items are available when you right-click the **Scenes** manager node.

New Scene	Executes the create scene action to create a scene of the selected type. See Creating a Scene.	
	Geometry	Creates a geometry scene.
	Mesh	Creates a mesh scene.
	Scalar	Creates a scalar scene.
	Vector	Creates a vector scene.
	Empty	Creates a new empty scene.
Open All Scenes	Executes the open all scenes action to display all scenes in the scenes manager.	
Test Graphics	Executes the test graphics action to report on the state of your graphics hardware. See A Scene Will Not Open.	

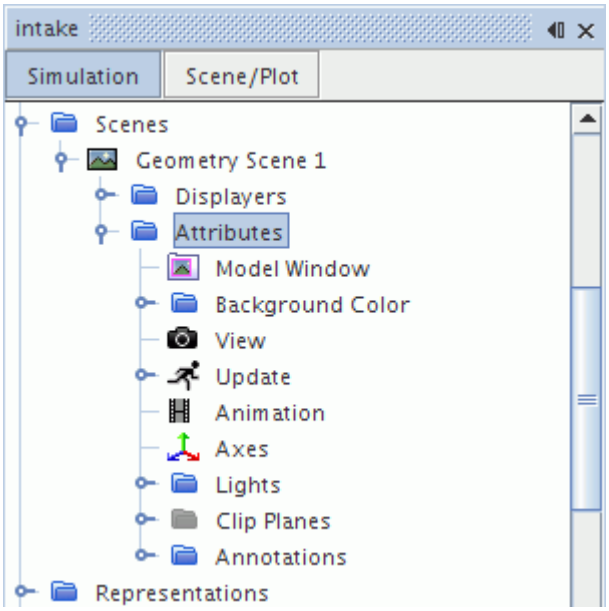
New Group	Facilitates the grouping of objects by creating a group node within the manager node. See Grouping Objects.
Ungroup	Reverses the grouping process.

See also Common Pop-Up Menu Items.

Changing the Attributes of a Scene

The **Attributes** node includes objects that let you control various characteristics of the scene, such as background colors, lighting, and animations.

This object exists within every scene node.



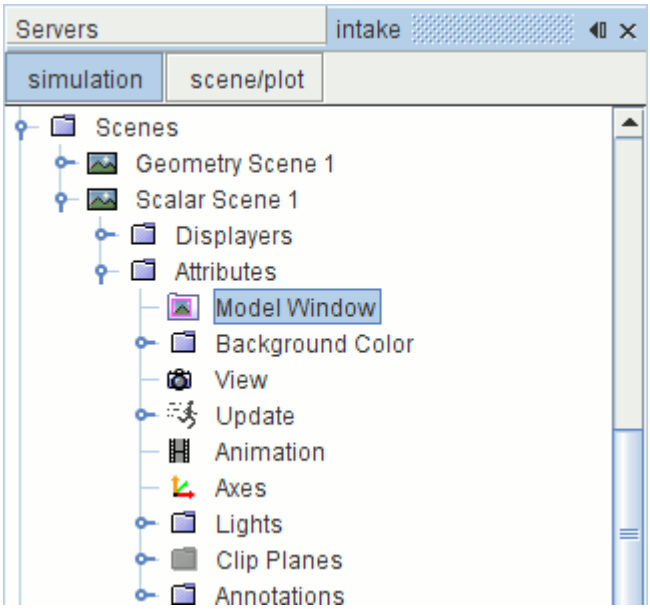
Contents:

Working with the Model Window
Setting the Scene Background
Environment Maps
Working with the Current View of the Scene
Updating Scenes
Creating Images for an Animation
Controlling the Frame Rate of the Scene Animation
Working with the Axes
Lighting
Working with Lighting Objects
Using Clip Planes
Working with Clip Plane Objects
Customizing an Annotation in a Scene

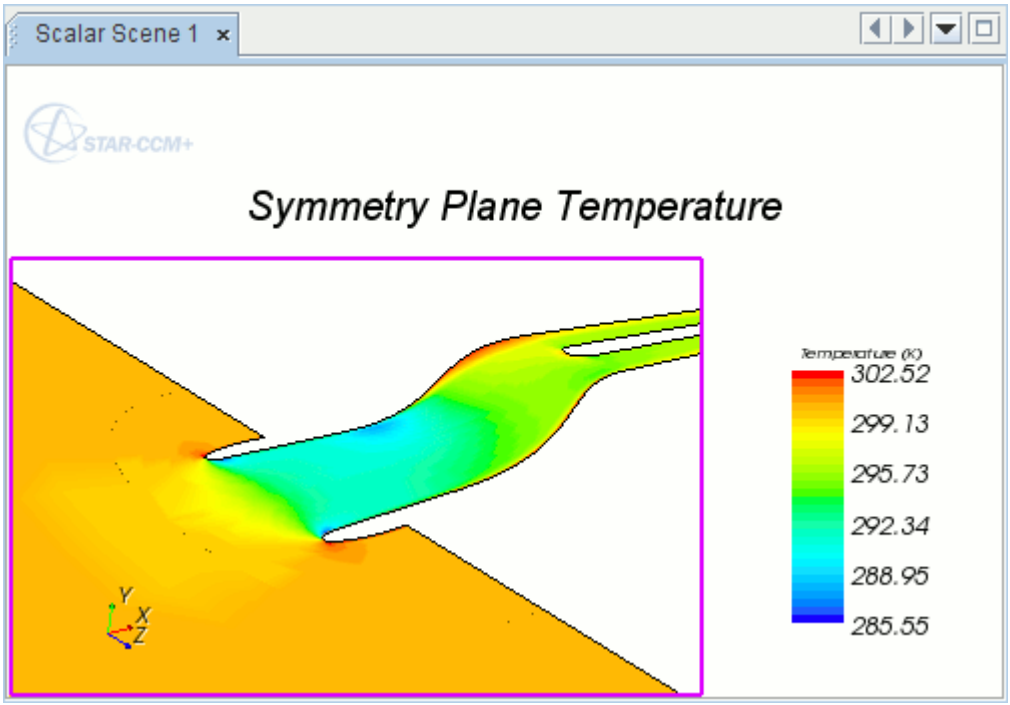
Working with the Model Window

The model window feature lets you crop part of the scene to show the model, while you use the rest of the space for elements such as annotations or scalar bars.

The **Model Window** node, which has properties and is subject to common menu items, represents this feature.



Use the properties to set the width, height, and position. In the example below, the width and height have been set to 0.7 of the display dimensions, allowing space for a simple text annotation, and a scalar color bar that is set to a vertical orientation. (The border of the model window appears only when the node of this feature is selected.)



Contents:

Model Window Properties

Model Window Properties

Use these properties to control the width, height, and position of the model window.

Width	The width of the model window as a fraction of the visualization display width, where 1.0 equals the full width.
Height	The height of the model window as a fraction of the visualization display height, where 1.0 equals the full height.
Position	The position of the lower left corner of the model window in the visualization display.

Setting the Scene Background

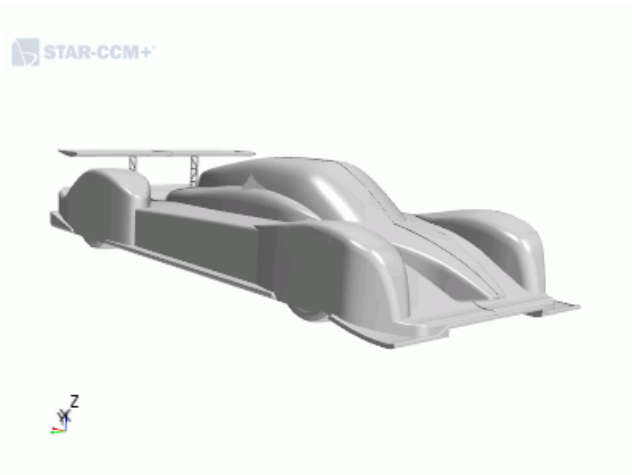
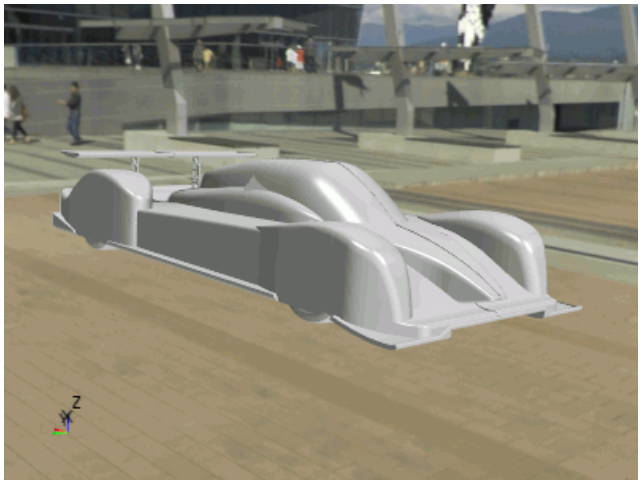
You can control the mode of the background of the scene: solid, gradient, checkerboard, or environment map.

The default mode of the scene background is **Gradient**. To change it:

1. Select the **Scenes** > **[scene]** > **Attributes** > **Background Color** node.
2. In the *Properties* window, select one of the following:
   - **Solid**
   - **Checkerboard**

- **Environment Map**

  Note: While the environment map background changes with the view of the scene, the other background types are independent of the view.

	
Gradient	Solid
	
Checkerboard	Environment Map

Alternatively, deactivate the gradient in every scene by default. If you do this, the **Checkerboard** option is not available either.

1. Select **Tools** > **Options**.
2. In the *Options* dialog, select the **STAR-CCM+** > **Visualization** node.
3. Activate the *Disable Gradient Background* expert property.
4. Click **Close**.

Background Color

This node lets you select the background mode of the scene.

Option	Activated Child Nodes
Solid	**Solid Background** Provides the *Color* setting, which controls color.
Gradient	**Gradient Background** Provides the following settings: *Color 1* The first color that is used in the gradient background color of the visualization display. The color can be input directly in RGB format (for example, white would be entered as `255,255,255`). A comma must separate the numeric values. The property customizer executes the define color action. See Defining Colors. *Color 2* The second color that is used in the gradient background color of the visualization display. *Gradient Mode* Controls the direction from Color 1 to Color 2 with the following options: • **Top-Bottom**: Color 1 at the top to Color 2 at the bottom. • **Bottom-Top**: Color 1 at the bottom to Color 2 at the top. • **Left-Right**: Color 1 on the left to Color 2 on the right. • **Right-Left**: Color 1 on the right to Color 2 on the left. • **Radial**: Color 1 at the center to Color 2 at the outer edges.
Checkerboard	**Checkerboard Background** Provides the following settings: *Color 1* The first color that is used in the gradient background color of the visualization display. The color can be input directly in RGB format (for example, white would be entered as `255,255,255`). A comma must separate the numeric values. The property customizer executes the define color action. See Defining Colors. *Color 2* The second color that is used in the gradient background color of the visualization display. *Squares per Color* Controls the checkerboard resolution. Each color has an equal number of squares. The available range is 1 to 100.

Environment Map	**Environment Map** See Environment Maps.

Environment Maps

With environment maps, you can wrap a photorealistic background around the contents of a scene.

An environment map is an image that surrounds your model and rotates along with the view. While this quality of background is best suited for advanced rendering, it works in any rendering mode.

Twelve environment maps are supplied with the current version of STAR-CCM+:

- **Desert Road**
- **Dirt Road**
- **Fall Road**
- **Heroes Square**
- **Lake At Sunset**
- **Meadow At Night**
- **Mountain Road**
- **Pebbles**
- **Snow Road**
- **Vancouver Convention Centre**
- **Warehouse**
- **Warehouse Interior**

These are six-sided and spherical licensed textures provided by Humus (http://www.humus.name/index.php?page=About) and HDRMAPS (http://hdrmaps.com/). While other environment maps may be downloadable online, it is your obligation to check the licensing terms on using them.

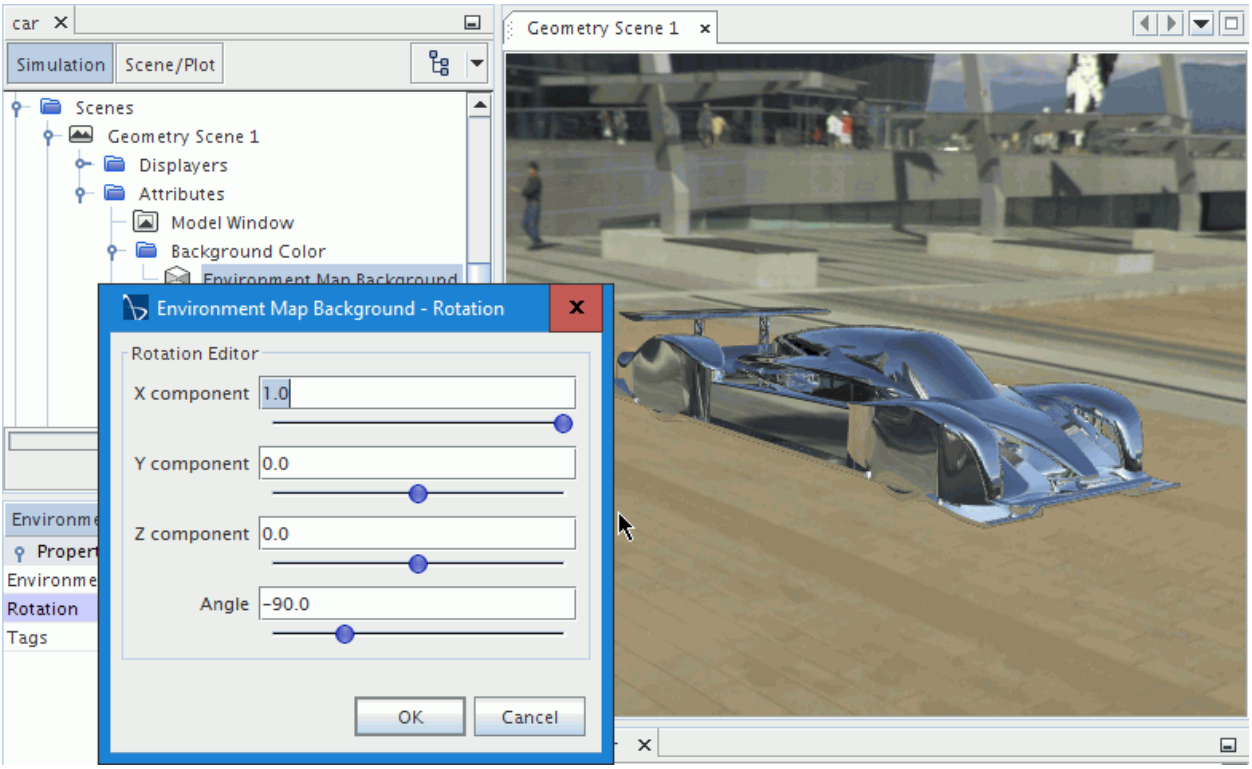
Using Environment Maps

To activate the environment map:

1. Select the **Scenes** > **[scene]** > **Attributes** > **Background Color** node.
2. In the *Properties* window, select **Environment Map**.
3. Select the **Environment Map** node and choose an available environment map from the drop-down list.

To align the environment map with the scene:

1. Select the **Background Color** > **Environment Map** node.
2. Click ... (**Custom Editor**) in the *Rotation* property.
3. Use the sliders in the editor dialog to update the scene interactively.



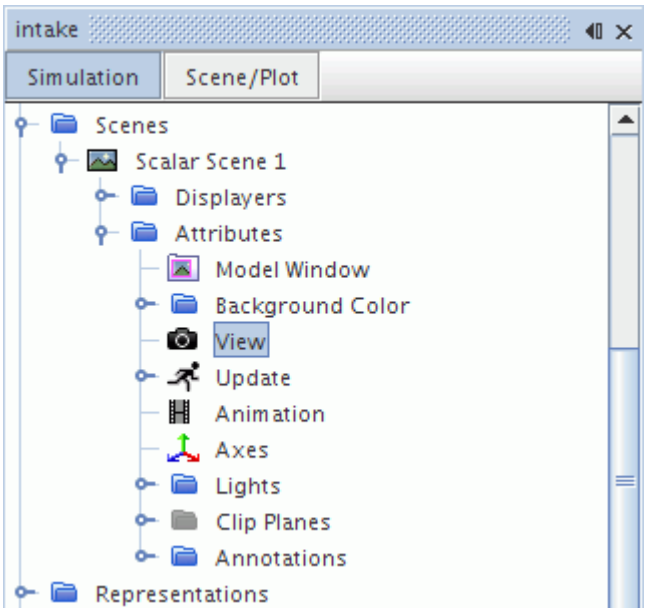
To add your own environment map:

1. Right-click the **Tools** > **Environment Maps** node and select one of the following options from the **New Environment Map** submenu:
   - **Cube**: an image on each of six faces
   - **Spherical**: a single 360-degree panoramic image
2. Import the images by specifying their relative paths through the corresponding *Image* property. (The cube has six separate paths in properties such as *Left Image* and *Right Image*.) The formats that can be imported are PNG, JPEG, TIFF, PNM, and BMP. Imported images are embedded in the simulation file by default. To link to an external image file (for example, for the purpose of limiting the simulation file size), *before you import,* deactivate the *Save to Simulation* property of the environment map node.
3. If you modify an image file after importing it into an environment map, you can update it within the simulation file. To update the image, right-click the node of its environment map in the **Tools** > **Environment Maps** node and select **Reload Image(s)**.
4. You can export the content of an environment map from within the simulation. This option can be useful when you want to use the images by themselves. From a cube environment map, the export action creates six images with numbers appended to the file name; from a spherical environment map, one image is created. To export images, right-click the node of their environment map in the **Tools** > **Environment Maps** node and select **Export Image(s)**. In the *Save* dialog, enter a file name and select the location for the image files, which are saved in PNG format.

Working with the Current View of the Scene

Information regarding the layout of the display contents of a scene, such as the projection and view angle, is stored in STAR-CCM+ as a view.

Specific views can be created and saved, and are stored in the simulation tree under the **Tools** > **Views** manager node. Each scene also has its own **View** node, under the **Attributes** node for the scene, which provides details of the current view in the selected scene. The property values of this node change dynamically if the view is altered by using the mouse, and can be controlled interactively by editing the values. This node has its own properties and pop-up menu.



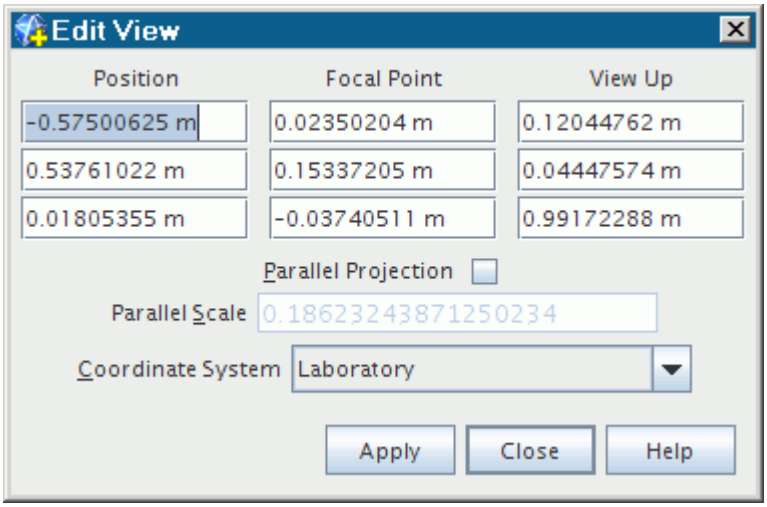
Contents:

Controlling the Current View
Setting the Parallel Scale in a Scene
View Properties
View Pop-Up Menu

Controlling the Current View

The current view of the scene can be controlled by right-clicking on the **Attributes** > **View** node of a scene and selecting **Edit...**

Use the *Edit View* dialog to define each of the view properties. This technique can also be used to edit the properties of an existing view under the **Tools** > **Views** manager node.



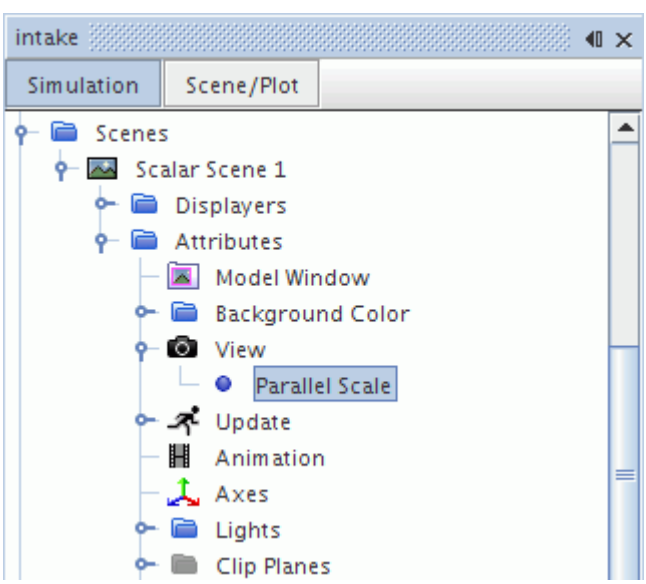
It is also possible to change the value of each property directly in the *Properties* window. When this method is used, STAR-CCM+ attempts to reconcile the new value with the existing view properties in order to update the scene. As a result, STAR-CCM+ may alter some of the properties values, and it is therefore recommended that the view properties are controlled using the *Edit View* dialog, as described above.

The *Parallel Scale* property in the dialog is used in parallel projection mode to scale the image. The parameter works as an inverse scale, in that larger numbers produce smaller images. Any user-defined value must be greater than zero. When you activate parallel projection mode, a parallel scale subnode is added to the **View** node.

Setting the Parallel Scale in a Scene

The **Parallel Scale** subnode of the **View** node is used to scale the image in parallel projection mode.

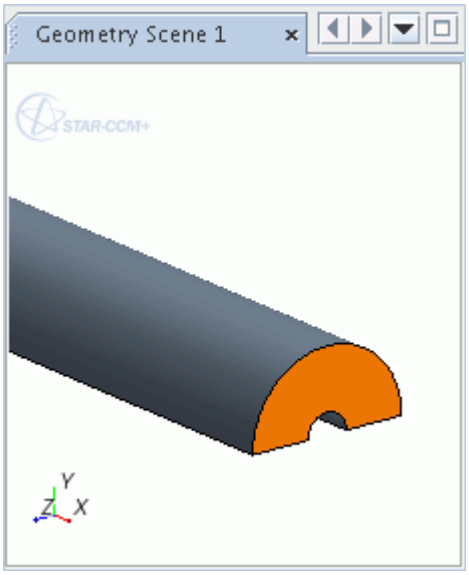
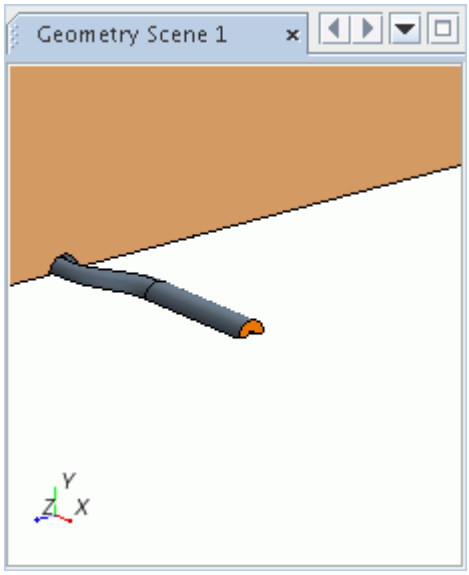
This subnode is made available when you set the *Projection mode* property of **View** node to `Parallel`. You can specify the scale in this object's properties.



Parallel Scale Properties

Parallel Scale	The user-defined value of the scale.

The parameter works as an inverse scale, in that larger numbers produce smaller images. Any user-defined value must be greater than zero.

	
Setting of 0.2	Setting of 1.2

View Properties

Use the view properties to control the position, coordinate system, and projection mode.

Focal Point	Cartesian vector specifying the position of the focal point. This property also changes when you zoom, pan, or rotate the visualization display.	
Coordinate System	A list of coordinate systems in the simulation for use by this view.	
Position	Cartesian vector specifying the position of the camera. This property also changes when you zoom, pan, or rotate the visualization display.	
Projection Mode	Defines the projection mode of the objects in the visualization display.	
	Perspective	Displays the objects in perspective mode with a vanishing point.
	Parallel	Displays the object with edges parallel to each other.

View Up	Cartesian vector specifying the direction where the camera is pointing. This property also changes when you zoom, pan, or rotate the visualization display.

View Pop-Up Menu

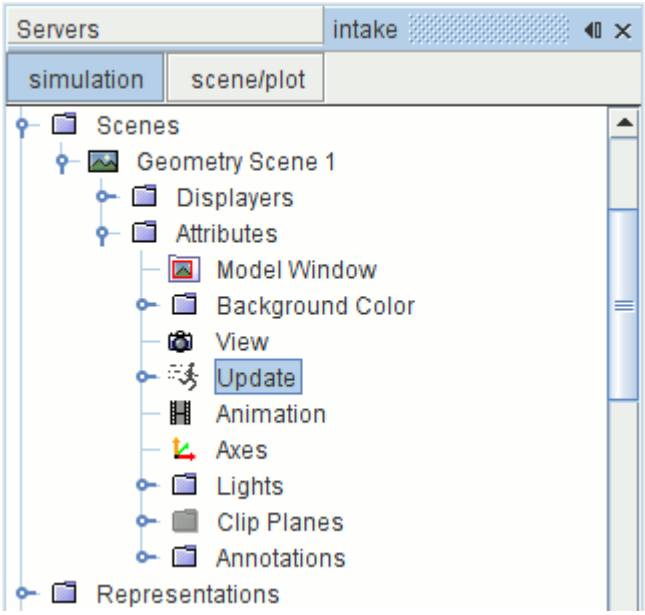
The view pop-up menu lets you edit views and store new ones.

Edit...	Opens the *Edit View* dialog, which can be used to control the current view of the scene. See Controlling the Current View.
Store View	Executes the store view action to save the view of the current scene. See Saving a View.

Updating Scenes

You can control how open scenes update by changing the trigger type that is located in the **Update** node.

This node, which has its own properties and is subject to common menu items, customizes how objects in the display of the *Graphics* window are updated as the simulation iterates. See Trigger Types.

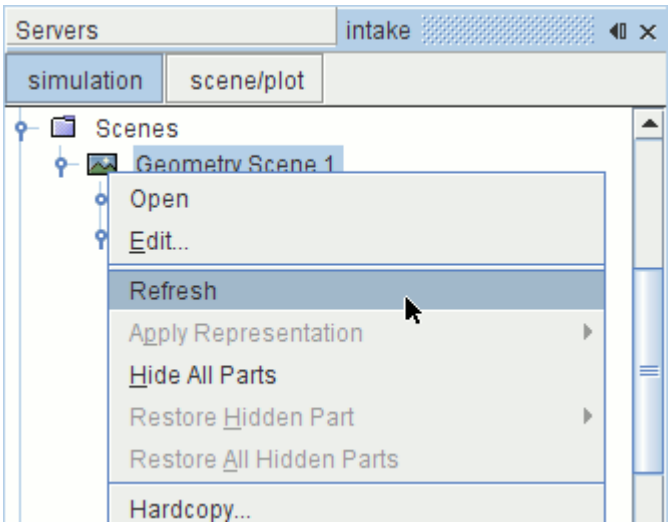


The **Update** node has sub-nodes with which to control the frequency of the update (see Trigger Properties) and the specifications of images (see Update and Hardcopy Properties).

For more information on selecting the trigger type and defining the update frequency, refer to Setting the Trigger for an Update.

Updating a Scene Manually

The scene can also be refreshed manually. Expand the **Scenes** node and right-click on the chosen scene. Choose **Refresh** from the pop-up menu.



Generating Images from a Scene

You can also generate an image of the scene after each iteration or time-step to compile an animation of the scene. This is different from the single image that is generated when you save a hardcopy of the scene.

If all clients are disconnected from a running server, no graphics scenes are saved to files.

Note:

- Multiple displayers within a scene that is displaying different scalar quantities update simultaneously whenever an **Update Frequency** setting triggers an update. For example, the scalar quantities that are displayed in a scalar displayer are updated to represent the new solution. Closed scenes are not updated until they are opened.
- Multiple scenes displaying the same scalar quantity update simultaneously whenever an **Update Frequency** setting triggers an update. For example, if you have a simulation with a 1 s time-step, and have one scene set to update every 5 time-steps (5 s), and a second scene set to update with a delta time of 2 s, you ultimately get updates to both scenes at: 2 s, 4 s, 5 s, 6 s, 8 s, 10 s, and so on.
- Multiple scenes using the same parts with the same or different scalar quantities also update simultaneously whenever an **Update Frequency** setting triggers an update. For example, if you have two scenes displaying the same part, and their **Update Frequency** settings are 10 and 15, both of them will update after every multiple of 10 or 15 iterations, from the start to the completion of the simulation run. This reflects the philosophy that all client scenes reflect an up-to-date snapshot of the server data.
- Hardcopies are only written to disk at the intervals that are dictated by the **Update Frequency** setting for a particular scene.

Contents:

Update and Hardcopy Properties

Update and Hardcopy Properties

The update and hardcopy properties allow you to control how the display is updated, as well as whether and how to generate image files of the display during a simulation run.

Update Properties

File Format	The graphics format that is used to save images of the visualization display. You have the same options as provided in the save scene to file action, as well as the scene file (`.sce`) format which is used by STAR-View+.
Enabled	Uses a checkbox with which you can activate or deactivate this feature.
Save to File	Determines whether multiple images of the scene display are saved as the solution progresses. This property can be used to create an animation showing how the object changes while the simulation runs. STAR-CCM+ brings the selected scene to the front when it produces a screenshot of that frame.
Output Directory	Path to where the files specified in the *Base Filename* property are saved.
Base Filename	The prefix of the file name to use when saving scenes to file. The full file name is `prefixNNNNN.ext` where `prefix` is this property, `NNNNN` is a number to indicate the sequence of the files, and `ext` is an extension based on the graphics format. If you leave the text box blank, STAR-CCM+ restores the name that was there previously once that text box no longer has the focus.
Trigger	Determines how the display is updated to reflect changes in the simulation. Change the trigger type to control the method that updates the scene. See Trigger Types.

Hardcopy Properties

The properties of the **Hardcopy** node let you control the specifications of the images generated by the scene updates. However, if all clients are disconnected from a running server, no graphics scenes are saved to files.

Output Magnification	Produces a larger image than the image in the original display, based on the factor you enter.
Output Width	Sets the width of the image of the scene display when the save scene to file action is executed. To make this specification effective, change it while recording a macro for saving scene images in batch.
Output Height	Sets the height of the image of the scene display when the save scene to file action is executed. To make this specification effective, change it while recording a macro for saving scene images in batch.
Use Anti-Aliasing	When activated, this option reduces distortion in the rendered image, creating a smoother appearance on screen.

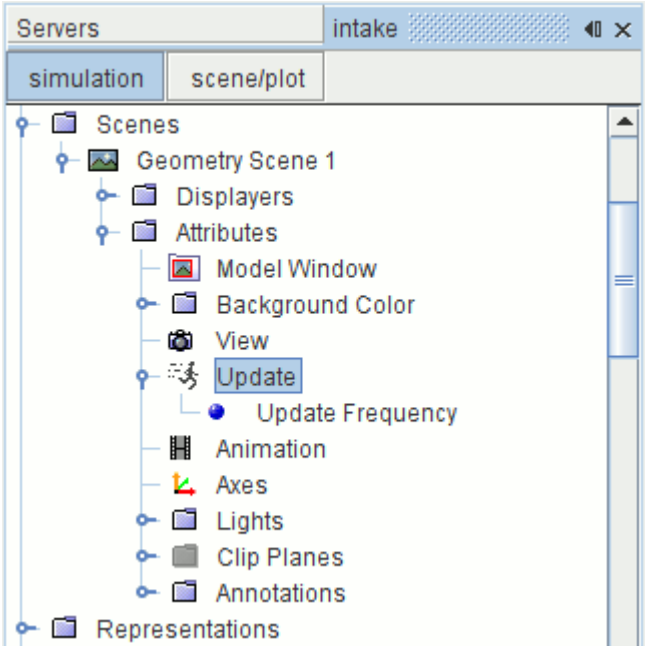
	However, anti-aliasing also increases rendering time. It is deactivated by default.	
Transparent Background	When activated, this option generates an image with a transparent background. This option is available for `.png` and `.tif` files only.	
Use Current Resolution	Controls how the images are sized. **Note:** To avoid unintended variation in the resolution of the images during a simulation run, deactivate this option after you finish preparing the scene.	
	Activated	Uses the latest scene display resolution, which changes as you resize the scene display. This setting is the default for scenes that you create in the current version of STAR-CCM+.
	Deactivated	Uses the pixel resolution values that you set in the **Output Width** and **Output Height** properties. This setting is the default for scenes in simulation files that were saved in STAR-CCM+ v11.02 or earlier.

Creating Images for an Animation

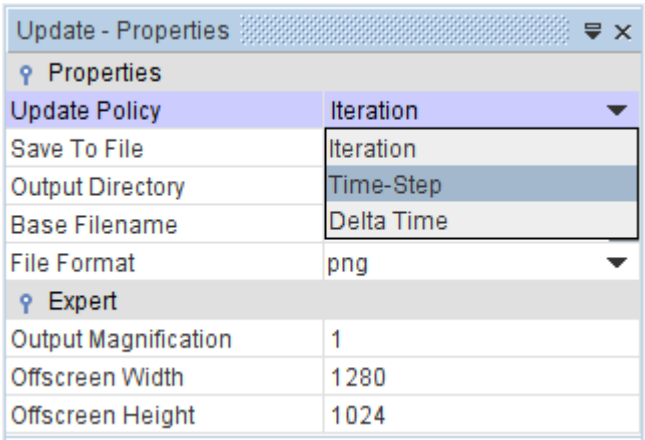
You can create images for an animation of the simulation, since STAR-CCM+ lets you output the image of the scene to a unique graphic file after each update of a scene during a simulation run.

Note: If all clients are disconnected from a running server, no graphics scenes are saved to files.

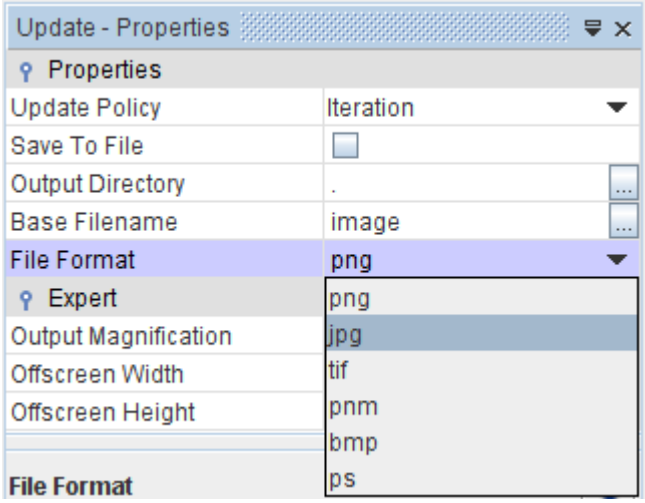
In the tree, expand the **Scenes** node, the node of the chosen scene and the **Attributes** node. Select the **Update** node.



Decide how frequently you want the scene to be updated by changing the properties of the **Update Frequency** child node. The default setting of 1 for the **Iteration** and **Time-Step** options means that the scene will update after each iteration or time-step. The default setting of 1 s for the **Delta Time** option means that the scene will update after 1 s in actual time. If you are working with a transient analysis, change the *Update Policy* from **Iteration** to **Time-Step** or **Delta Time**.

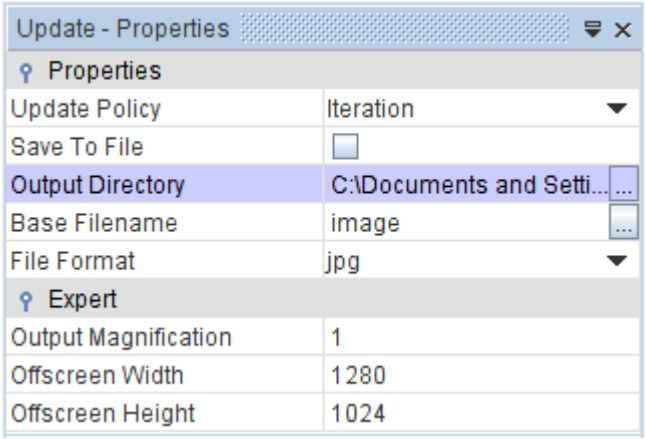


The format of the file can be selected from the options in the drop-down menu under the *File Format* property.

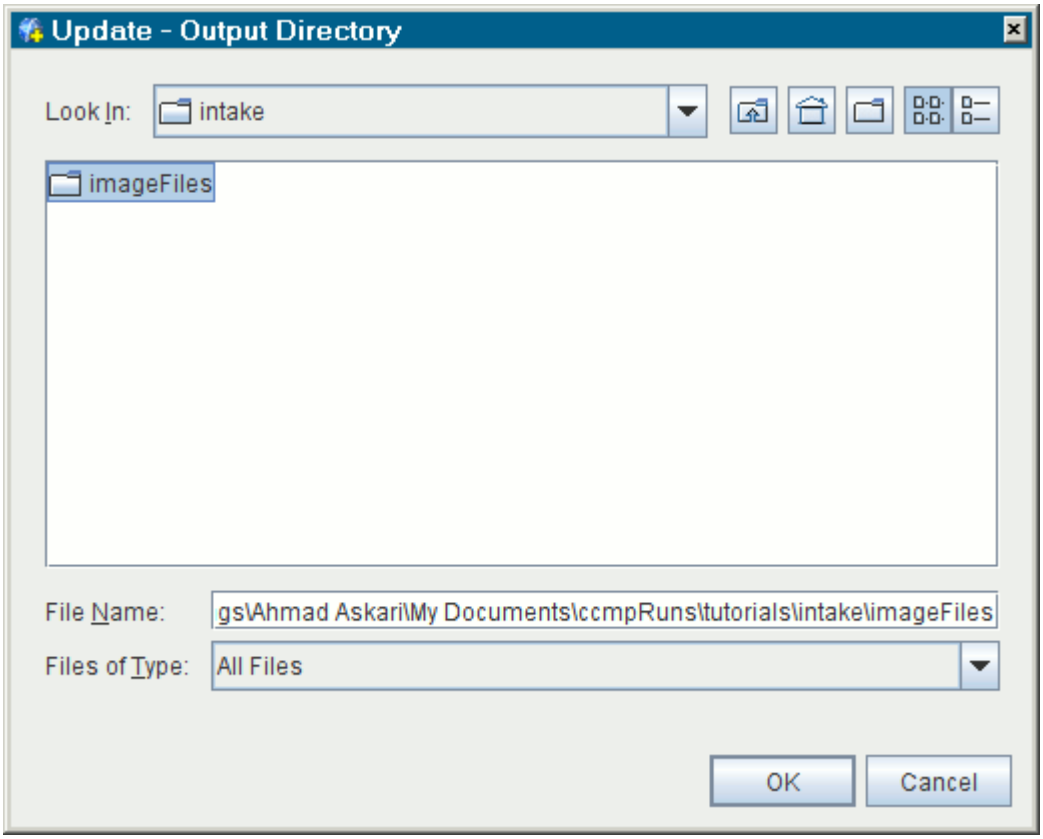


You have the same format options as those provided when you save a scene to a file. See Saving Scene Image Files.

In the *Output Directory* property, you can control where the files are created one of two ways. If you know the full path, simply enter it into the text box of the property.

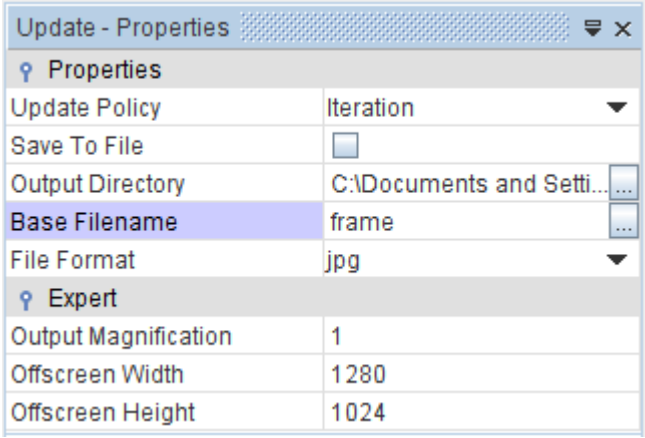


Otherwise, click the property customizer (...) to activate the browser dialog.

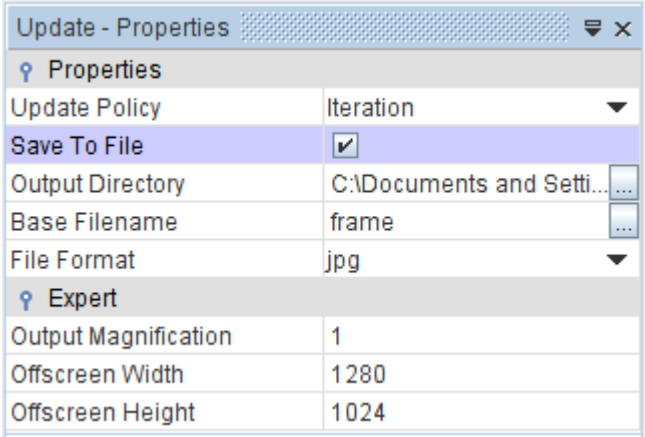


Choose the location carefully; if, for example, you generate one image per iteration and you run 1000 iterations, then 1000 image files are created.

Change the *Base Filename* to something other than `image` if you prefer.



Tick the checkbox of the *Save to File* property.

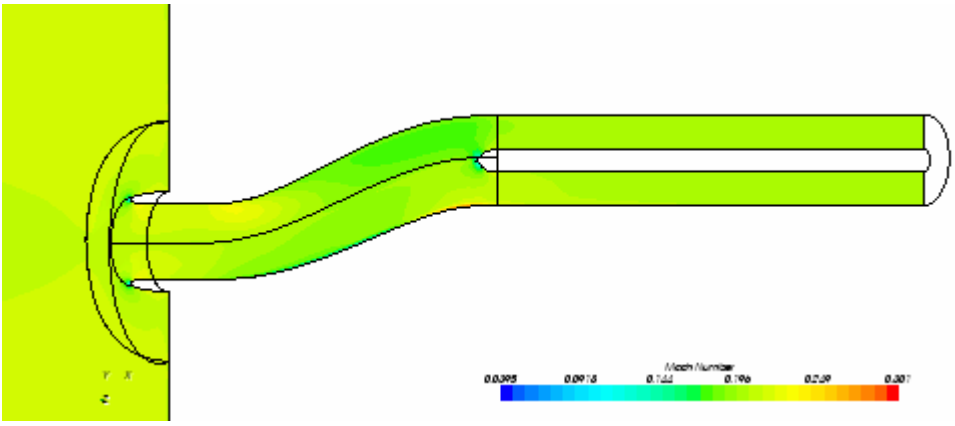


The update feature is ready for you to begin the run. Before you do, however, make sure that your visualization display is not obscured by anything else on your screen. If anything is covering the display at the time an image file is saved, it appears in the image. For example, be sure to deactivate your screen saver; otherwise all you have at the end of the run is an animation of your screen saver.

Also, have only one scene active at a time. Depending on your system and your version of STAR-CCM+, the scenes may conflict with each other and result in an incomplete set of images.

Run the simulation for the desired number of steps. As it runs STAR-CCM+ will save an image file after each update. The name of each image file (the base file name) is concatenated with the number of the update to ensure that each file is unique.

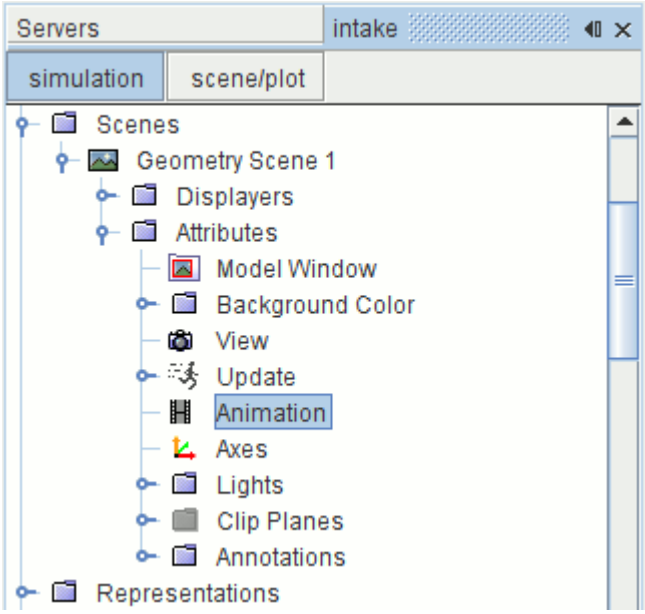
These files can then be animated with your media program of choice.



Controlling the Frame Rate of the Scene Animation

You can adjust the number of frames per second of all scene animations with the **Animation** node.

This object is in the **Attributes** subnode of a scene node.



Contents:

Animation Properties

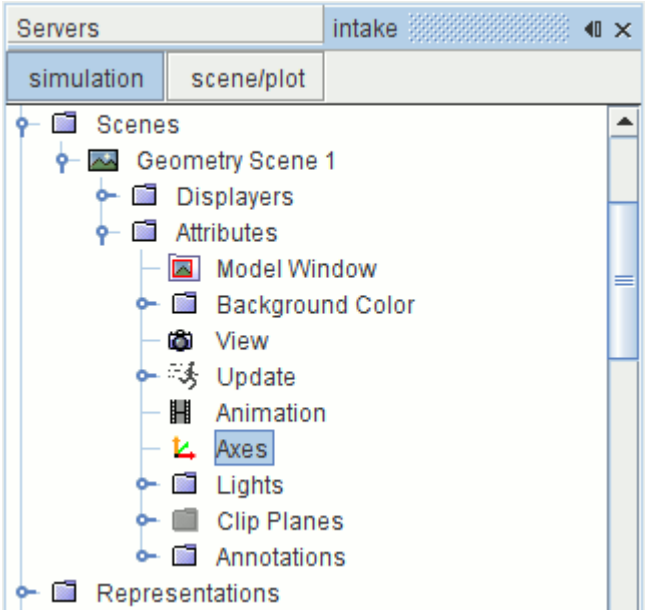
Animation Properties

In the Animation properties, set the frame rate of the animation.

Target frame rate (fps)	Maximum number of animation frames that are displayed in one second (less for complex scenes, or when less computing power is available).

Working with the Axes

It is possible to control the orientation axes in the visualization display. The **Axes** node, which has its own properties, lets you do this.



To hide the axes, expand the **Scenes** node, the node of the chosen scene and the **Attributes** node. Select the **Axes** node. In the *Properties* window clear the checkbox of the *Visible* property.

To show the axes again, tick the checkbox of that property.

Contents:

Axes Properties

Axes Properties

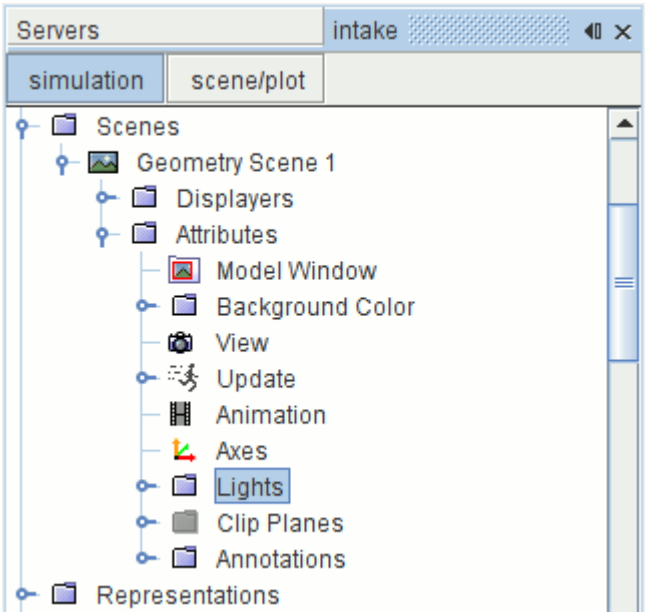
The properties of the **Axes** node let you specify various characteristics of the axes in the scene display.

Text Color	Controls the color of the X, Y, and Z labels of the axes. The property customizer executes the define color action. See Defining Colors.	
Visible	Sets visibility of the axis indicator. You may want to hide the axis indicator when it overlaps with an object in the visualization display, or when you want to produce a printout or a screenshot.	
	Activated	Axis indicator is visible.
	Deactivated	Axis indicator is hidden.
Viewport	Gives the coordinates, in the form `[x0, y0, x1, y1]`, of the lower left and upper right corners of a rectangle containing the axes. By changing the coordinates, you can control the position and size of the axes. Units are fractions of the viewport height and width. The axes do not fill the rectangle.	

Lighting

Each scene includes a sophisticated set of lights that function like those in a real display setting. The **Lights** manager node contains all of the light nodes in the scene.

These lights can help you study an object's design, or generate images for presentation documents. Unlike the lighting of displayer surfaces, which determines how parts within a particular displayer absorb the light, this lighting affects the scene as a whole.



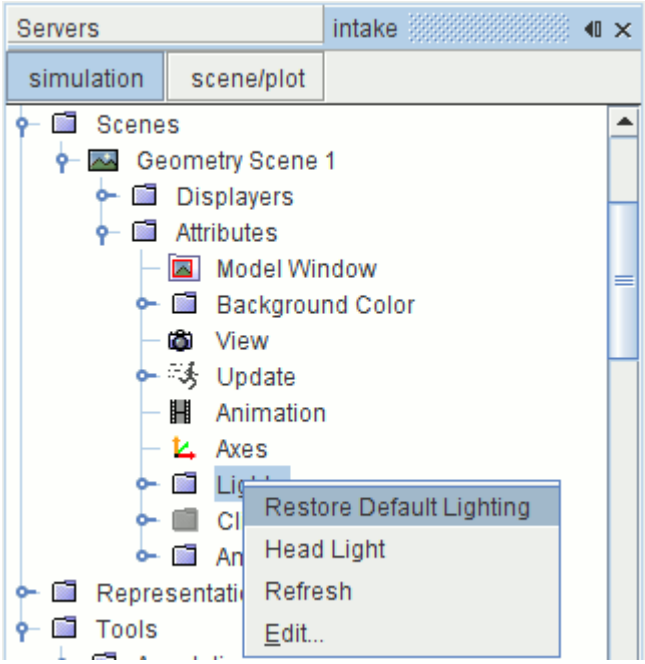
In particular, the properties of the manager object offer a choice of lighting models. The pop-up menu of the manager allows you to restore default settings, use the head light feature, and refresh the lighting.

Contents:

Restoring Default Lighting Settings
Using the Head Light Feature
Using the Refresh Action
Lights Manager Properties
Pop-Up Menu of the Lights Manager

Restoring Default Lighting Settings

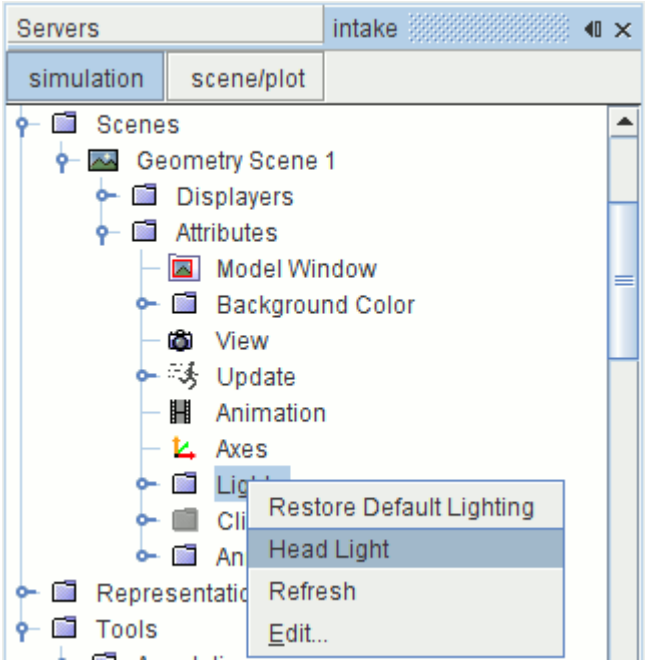
To reverse all of your changes to the light settings, simply right-click on the **Lights** manager node and select **Restore Default Lighting**.



If no adjustments have been made to properties of any lights during this session, then this action has no effect.

Using the Head Light Feature

To focus Light 1 on the object in the visualization display, right-click on the **Lights** manager node and select **Head Light**.



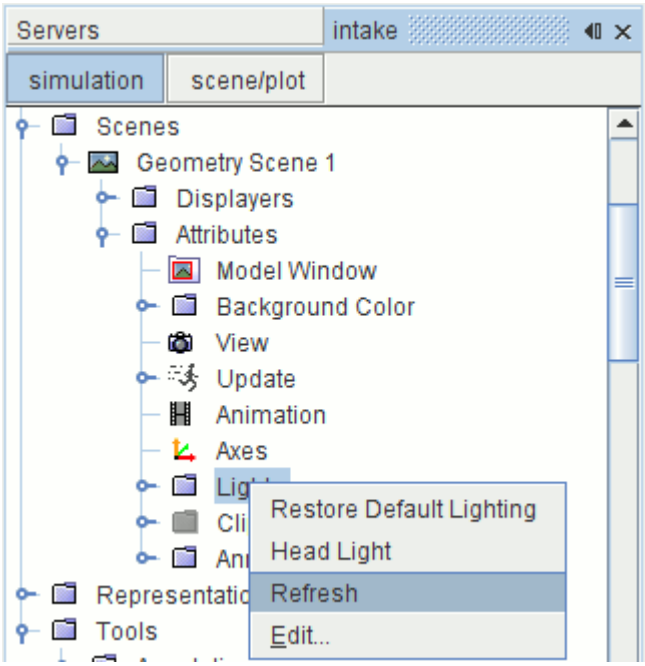
This action does the following:

- Turns off the other three lights
- Sets the intensity of Light 1 at the maximum
- Reverts the color of Light 1 to white
- Sets the azimuth and elevation of Light 1 to zero

Using the Refresh Action

After you finish changing the light settings, you may need to refresh the lighting to see the effects of your changes.

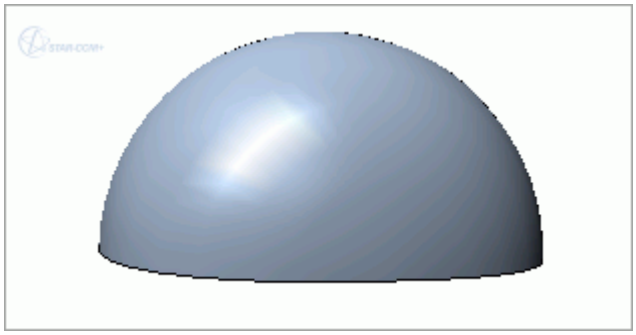
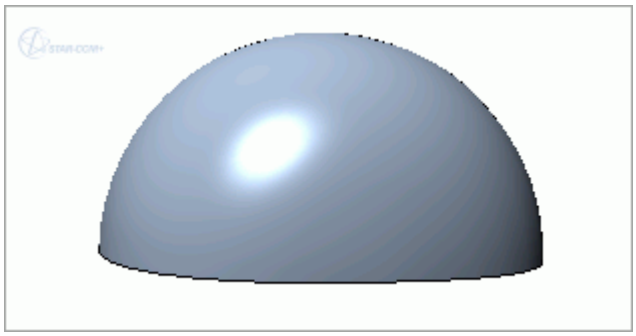
To refresh the lights, right-click the **Lights** manager node and select **Refresh**.



Lights Manager Properties

To specify the lighting model, use the properties of the **Lights** manager node.

Lighting Model	Chooses a mode for lighting and shading.	
	Per Vertex Lighting	This option calculates lighting by interpolating among vertices. It is relatively economical with graphics processing requirements.

		
	Per Pixel Lighting	This option calculates lighting based on pixels. While it requires more graphics processing power than the **Per Vertex Lighting** option, it also provides a more precise image. Also, this option makes it possible to light streamlines. 

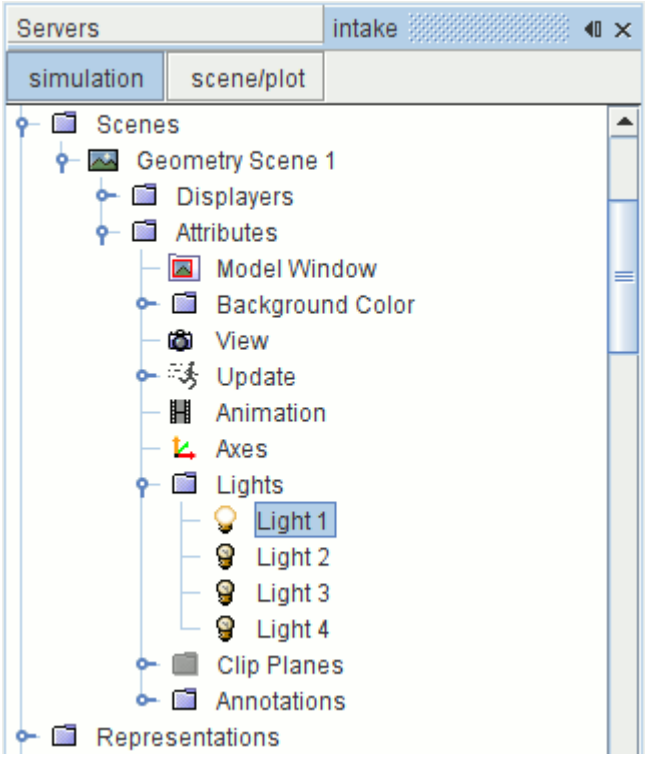
Pop-Up Menu of the Lights Manager

These pop-up menu items are available when you right-click the **Lights** manager node.

Restore Default Lighting	Executes the restore default lighting action. See Restoring Default Lighting Settings.
Head Light	Executes the head light action. See Using the Head Light Feature.
Refresh	Executes the refresh action. See Using the Refresh Action.
Edit	Executes the edit action. See Using the Table Editor.

Working with Lighting Objects

A **Light** node, which has properties and is subject to common menu items, contributes to lighting in the scene.



It is not possible to create, delete, or rename these objects. There are four of them in a scene, all identical except that Light 1 cannot be disabled with the *Enabled* property.

Restoring default settings and using the head light feature are part of working with the **Lights** manager node.

Contents:

How Do Lighting Objects Work?
Deactivating and Reactivating Lighting Objects
Controlling the Intensity of Lighting Objects
Controlling the Color of Lighting Objects
Light Properties

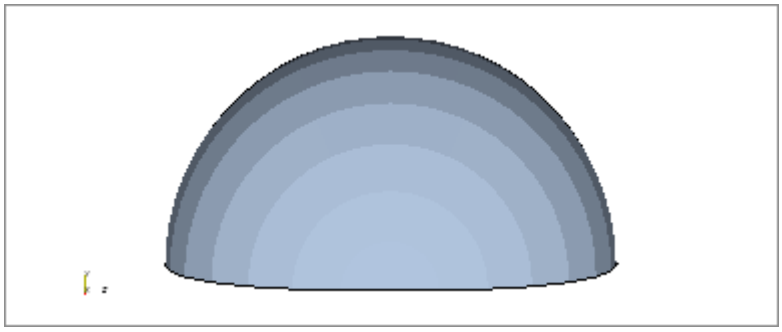
How Do Lighting Objects Work?

This section explains the principles by which lights in STAR-CCM+ scene displays work.

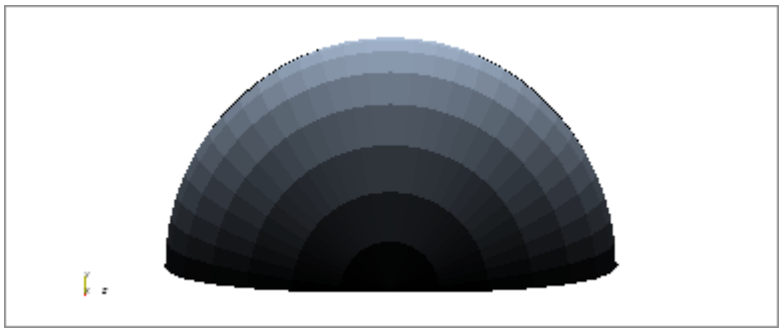
All lights are directional lights (infinitely far away with no falloff). Lights move with the camera, which is controlled by the view.

For simplicity, the position of lights in STAR-CCM+ can only be specified using angles. The *Elevation* (latitude) and *Azimuth* (longitude) properties of each light are expressed in degrees with respect to the camera. (Lights always shine on the camera's focal point.) For example:

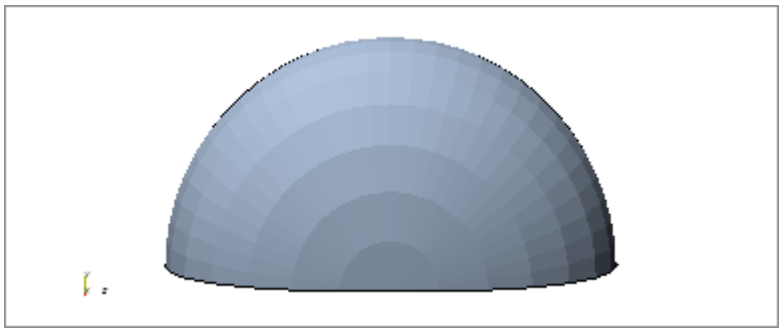
- A light at (*Elevation* = `0`, *Azimuth* = `0`) is at the camera (a headlight).



- A light at (*Elevation* = `90`, *Azimuth* = `0`) is above the focal point, shining down.



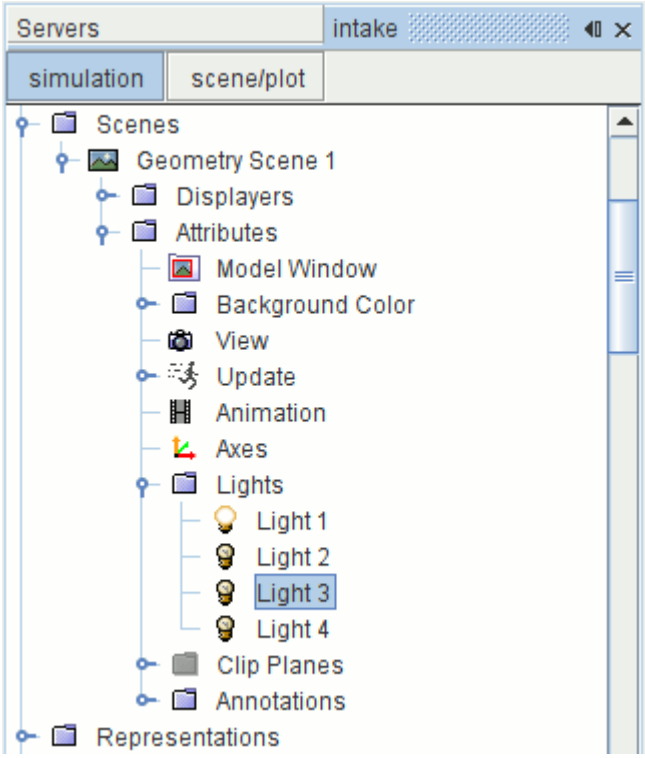
- Negative azimuth values move the lights counter-clockwise as seen from your point of view, positive values clockwise. So, a light at (*Elevation* = `45`, *Azimuth* = `-20`) is above and in front of the object, shining slightly from the left side.



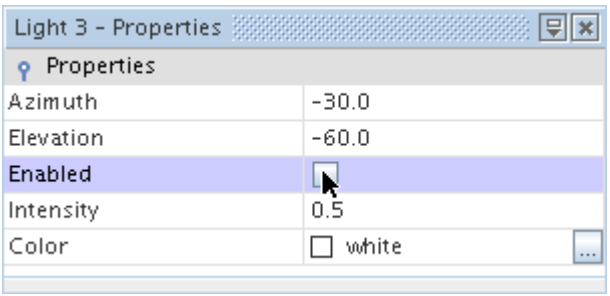
Deactivating and Reactivating Lighting Objects

Individual lights can be deactivated or reactivated, except for **Light 1**.

To turn off a particular light, select its node in the object tree.



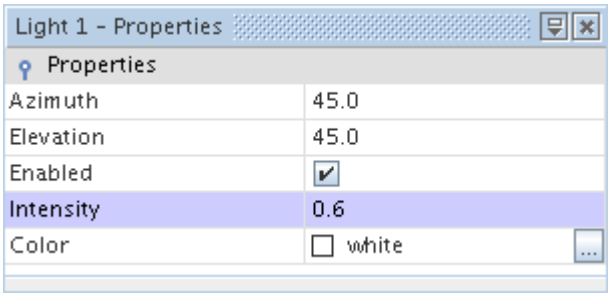
Clear the checkbox of its *Enabled* property.



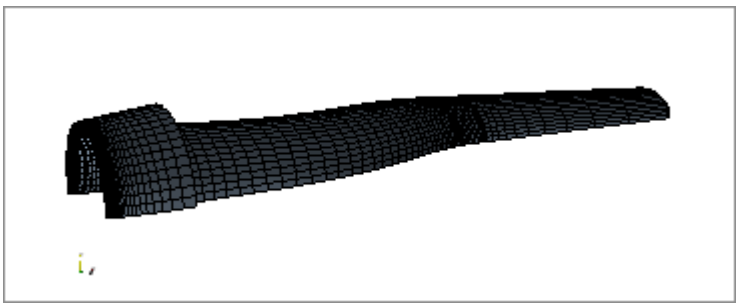
(Although this cannot be done for **Light 1**, you can turn off that light by setting its *Intensity* property to `0`.) To turn the light back on, tick the checkbox.

Controlling the Intensity of Lighting Objects

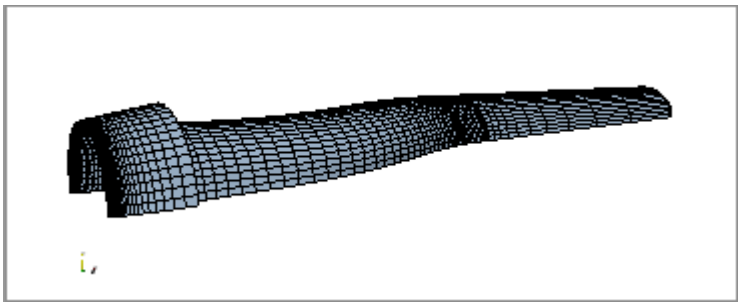
The *Intensity* property controls the brightness of the light.



Zero is dim.



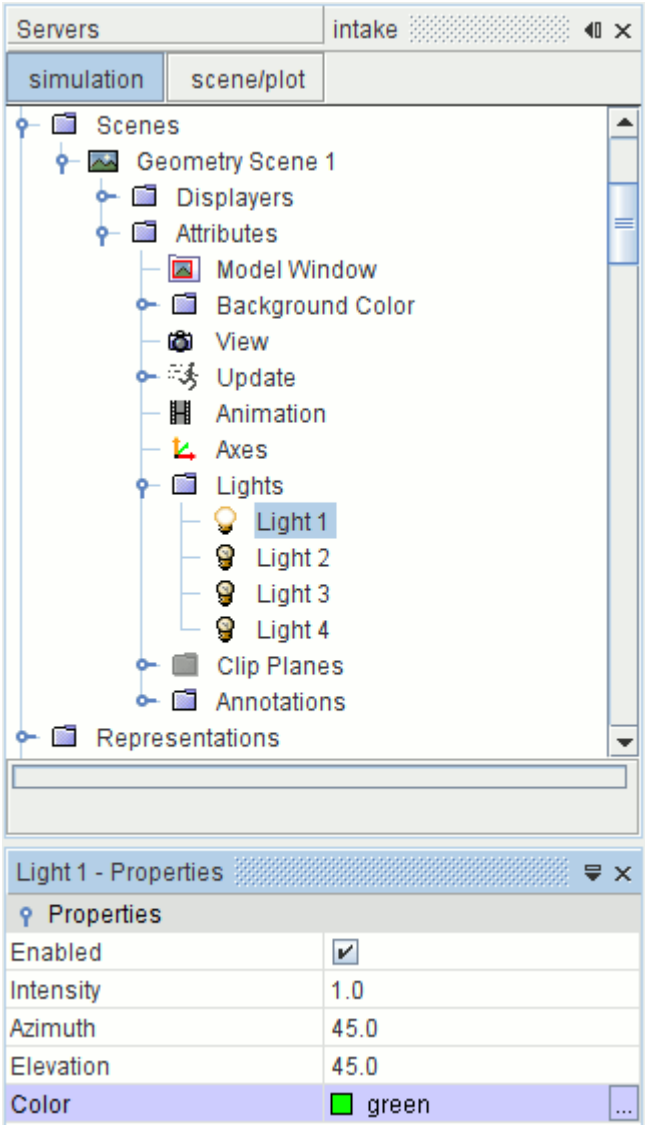
One is maximum brightness.



Controlling the Color of Lighting Objects

The *Color* property of lights lets you add another visual effect to the display.

You can use colors for highlighting parts of the model, or simply for a preferred aesthetic effect.



Light Properties

The properties of a typical light object lets you activate or deactivate it, as well as control its lighting characteristics.

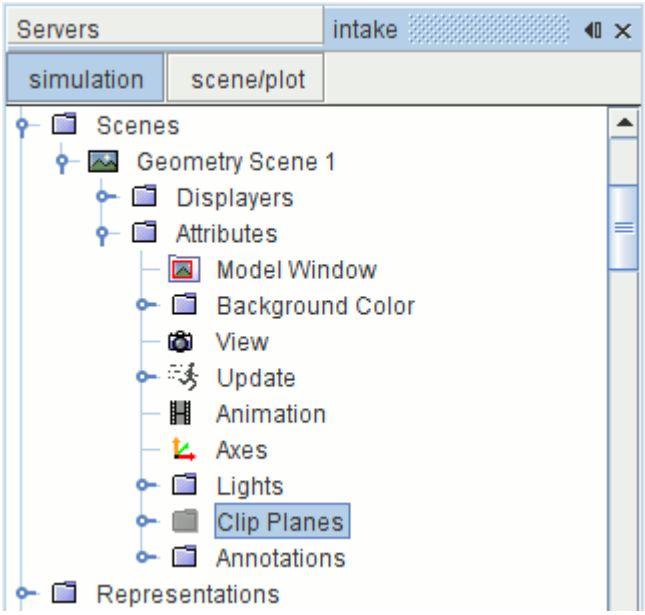
Azimuth	Azimuthal location of light in degrees.	
Elevation	Elevation of light in degrees.	
Enabled	Uses a checkbox to turn on or off the light.	
	Activated	Turns on the light (default).
	Deactivated	Turns off the light.
Intensity	Controls the strength of the light with text entries as fractions of 1. An entry of 0 effectively turns the light off, while an entry of 1 sets the light to its maximum intensity. Any entry greater than 1 yields an error message.	

Color	Controls the color of the light. The property customizer executes the define color action. See Defining Colors.

Using Clip Planes

In every scene, you can render portions of the model invisible with clip planes.

The **Clip Planes** manager node, which has its own pop-up menu, contains all of the clip plane object nodes in the scene. By default it is disabled, as shown by the icon's gray color.



Clipping parts in a scene differs from removing them in that they are still in that scene. It also differs from hiding a part in that you are cropping the view by a set of coordinates that you control, rather than excluding an entire part from view.

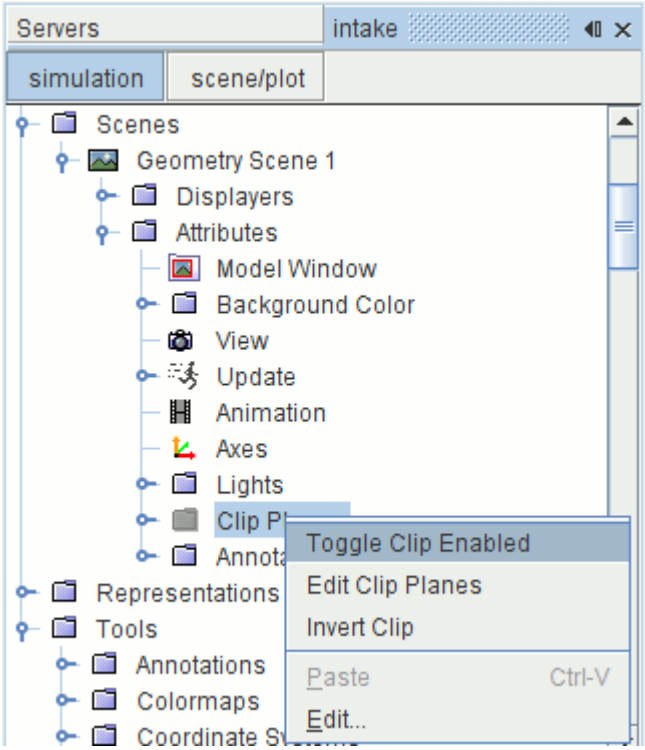
This feature allows you to display information, such as scalars or vectors, within a space of your choice. In addition, it allows you to display a smaller portion of the model.

Contents:

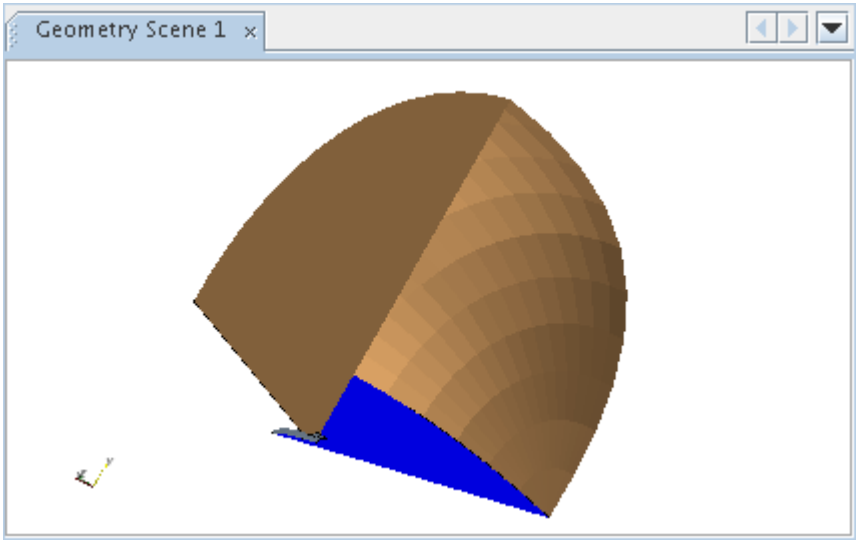
Enabling Clip Planes
Modifying Clip Planes
Inverting the Coordinates of Clip Planes
Pop-Up Menu of the Clip Planes Manager

Enabling Clip Planes

To enable the clip planes in a scene, right-click the **Clip Planes** manager node and select **Toggle Clip Enabled**.



The scene shows the objects clipped along the planes, based on their default coordinates.

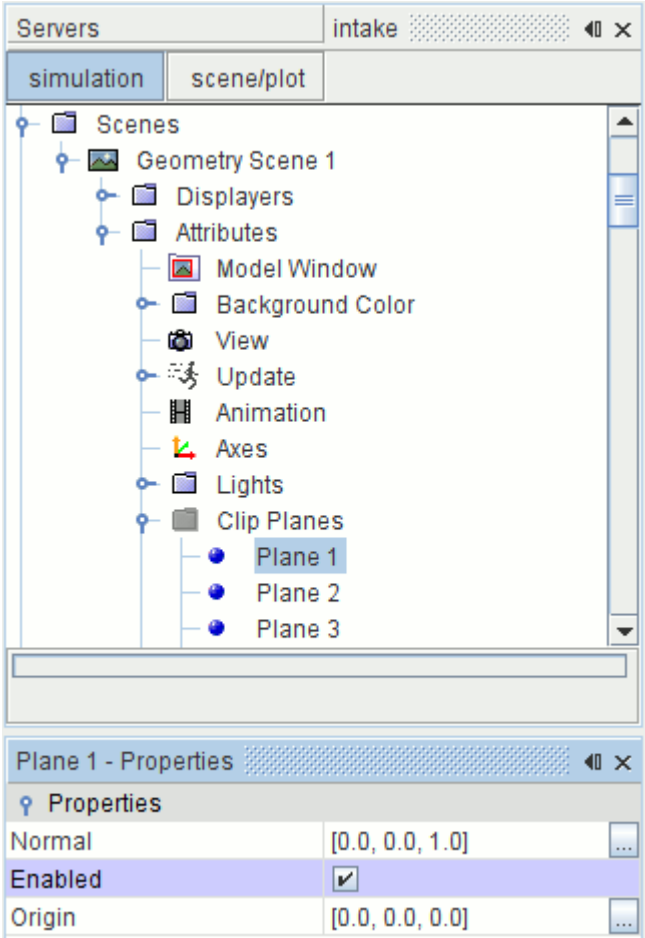


To reverse this step, you can disable the clip planes with the same pop-up menu item. You can also disable and enable individual clip planes. See Disabling and Enabling Clip Plane Objects.

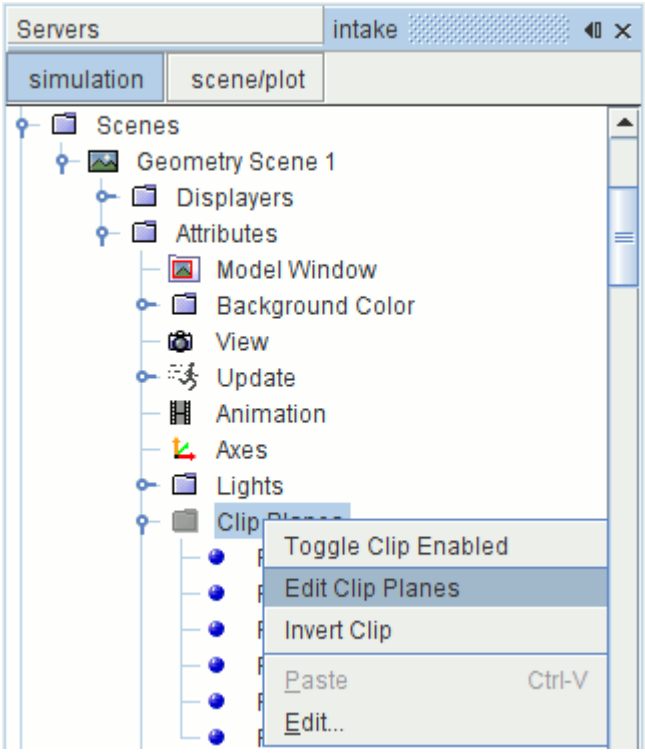
Modifying Clip Planes

Multiple techniques exist for modifying clip planes.

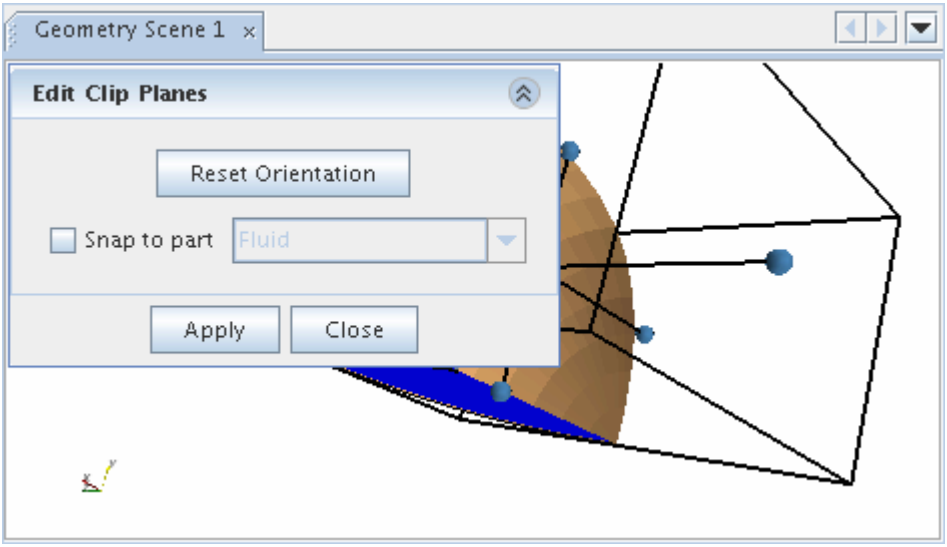
To edit a clip plane, select its node in the object tree and edit the properties.



Alternatively, to modify the planes visually, right-click the **Clip Planes** manager node and select **Edit Clip Planes**.

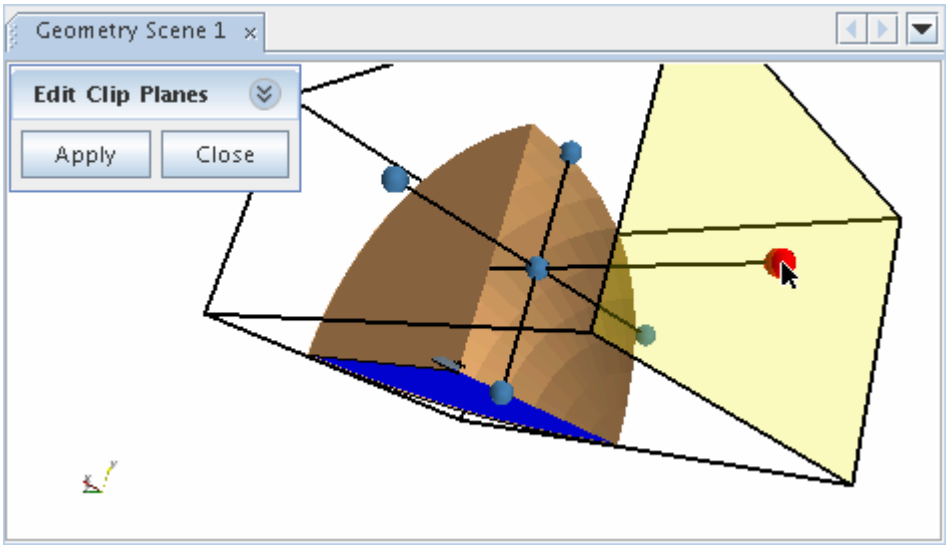


An interactive in-place dialog appears in the scene. (If the scene display is not in view, the **Edit Clip Planes** action causes the display to be opened or become active.)



Use the (**Up-Arrow**) button to minimize the in-place dialog to see the entire scene.

You can adjust the tool by clicking and dragging any of the end point handles of the lines, or any of the sides of the bounding box.

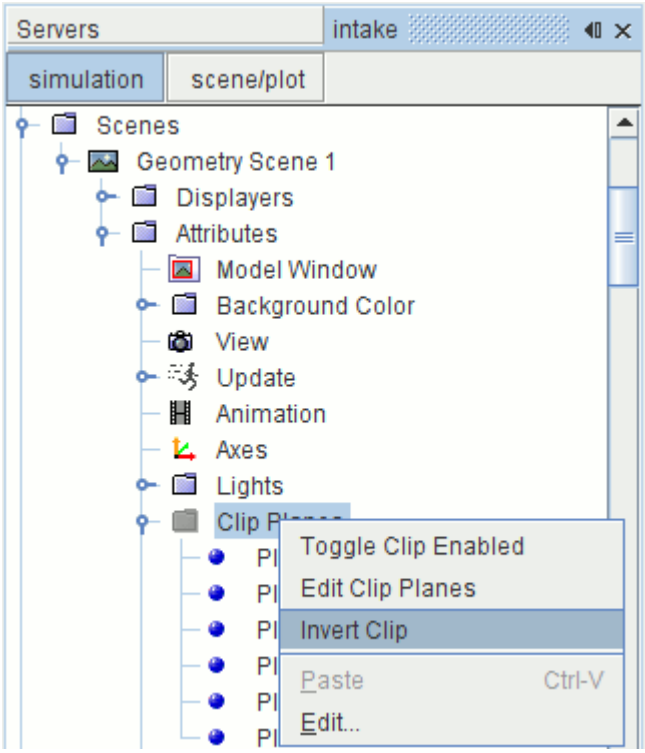


When done with the tool, restore the dialog with the (**Down-Arrow**) button. Click **Apply** to implement your changes. If you have no further changes to make with the tool, click **Close** to remove the dialog.

Inverting the Coordinates of Clip Planes

This action changes all normal values from positive to negative, and vice versa.

To invert the normal values of all clip planes in a scene at once, right-click the **Clip Planes** manager node and select **Invert Clip**.



Pop-Up Menu of the Clip Planes Manager

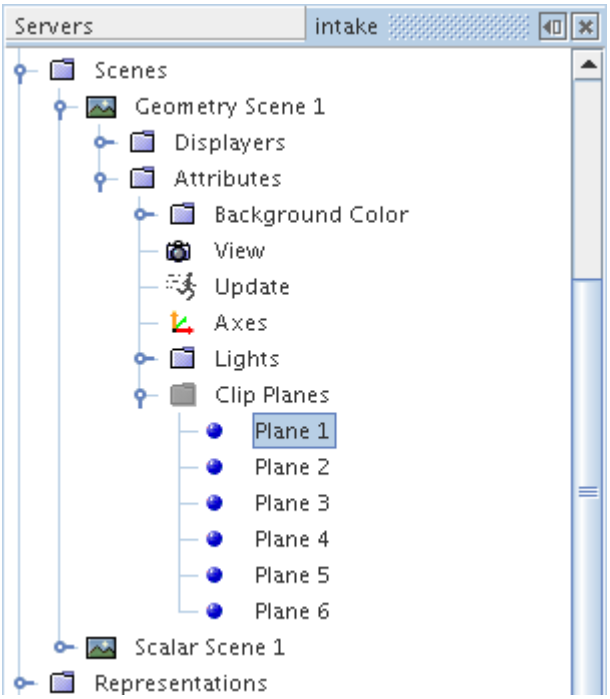
You can carry out a variety of operations with the pop-up menu of the **Clip Planes** manager node.

Toggle Clip Enabled	Enables and disables the clip planes.
Edit Clip Planes	Opens an in-place dialog to modify the clip planes visually.
Invert Clip	Switches the normals of the clip planes to their negative equivalents.
Paste	Pastes the copied clip plane to the clip planes object tree.
Edit	Opens the edit dialog to modify the clip planes properties.

Working with Clip Plane Objects

Clip plane objects represent individual sets of adjustable coordinates for clipping an object along a particular plane in a scene.

There are six of these clip plane nodes in any scene, all with identical properties.



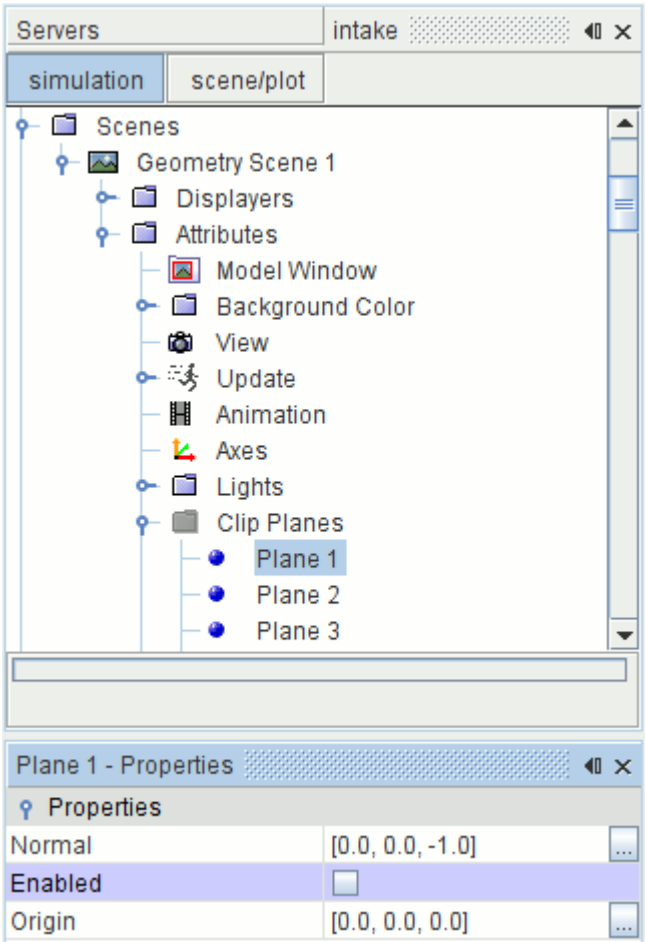
It is not possible to create or delete these objects, though they can be renamed with the inline editor, copied using the clipboard, as well as individually disabled or enabled.

Contents:

Disabling and Enabling Clip Plane Objects
Clip Plane Object Properties

Disabling and Enabling Clip Plane Objects

You can deactivate an individual clip plane by selecting its node and clearing the *Enabled* property checkbox.



Tick the same checkbox again to reactivate it.

Clip Plane Object Properties

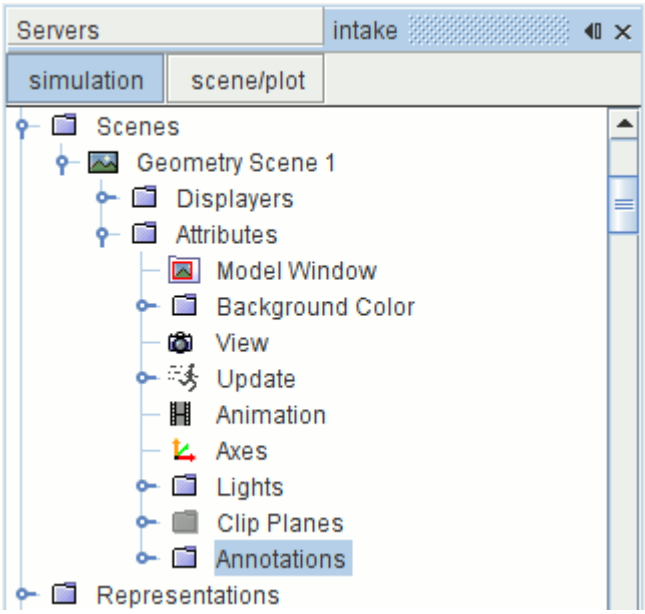
Use the properties of a clip plane object to activate it, deactivate it, and specify its coordinates.

Enabled	Uses a checkbox to turn on or off the clip plane.	
	Ticked	Turns on the clip plane (default).
	Cleared	Turns off the clip plane.
Normal	Sets x, y, and z coordinates for the second point of the plane.	
Origin	Sets x, y, and z coordinates for the origin of the plane.	

Customizing an Annotation in a Scene

Scenes can have their own annotation objects that you can customize separately from the original annotations.

Customized annotations are also called applied annotations, and they are stored in the **Attributes** > **Annotations** node within the scene node. The **Annotations** node has properties and is subject to common menu items.



Contents:

What Are Applied Annotations?
Annotations Manager Properties

What Are Applied Annotations?

An applied annotation node represents the addition of an annotation to a scene. Its name matches that of the original annotation.

To create an applied annotation, simply follow the procedures for adding an annotation to a scene. Once an applied annotation is in a scene, it can only be changed by modifications to its own properties. If you edit the original annotation subsequently, it does not affect the applied annotation. The exception is the text content of a simple text annotation, which can only be edited from the original.

If you have an annotation that you copied to multiple scenes, it is possible to apply a change globally with the following steps:

- Modify the original annotation.
- Delete the applied annotations that you wish to update.
- Add the modified annotation to all the scenes, creating replacement applied annotations.

To rename an applied annotation, rename the original annotation. To hide or show the applied annotation in the scene, clear or tick the checkbox of its *Visible* property.

Applied Annotation Properties

These properties are shared by all applied annotations other than the **Background** type, except where otherwise noted.

Width	The width of the annotation as a fraction of the visualization display width, where 1.0 equals the full width (available for **3D Image** and **Plot Image** types only).	
Height	The height of the annotation as a fraction of the visualization display height, where 1.0 equals the full height (not available for **Scene Grid** type).	
Position	The position of the lower left corner of the text or graphic in the visualization display based on the coordinate system specified in the *Coordinate System* expert property (not available for **Scene Grid** type).	
Rotation	Rotates the image in a counterclockwise direction (**3D Image** type only).	
Normal	Sets x, y, and z coordinates for the second point of the image plane (**3D Image** type only).	
Visible	Shows or hides the annotation in the scene (available for all types, including **Background**).	
	Activated	The annotation is visible.
	Deactivated	The annotation is hidden.
Scale Components	Controls whether to scale the elements of a plot image (such as font and line thickness) based on the overall size of the plot image annotation as it appears in the scene (**Plot Image** type only).	
	Activated	Scales the plot image in the scene.
	Deactivated	Respects the original pixel size specified for the chart elements.
Width Stretch Factor	Scales the width of the annotation, with the default of 1.0 equaling 100% of the original width (not available for **3D Image**, **Plot Image** and **Scene Grid** types).	

Applied Annotation Expert Properties

These expert properties are not available for the **3D Image** and **Background** types.

Location	Controls the visibility of the annotation relative to the objects in the display.	
	Background	Shows the objects in front of the annotation.
	Foreground	Shows the annotation in front of the objects.

Representation	Controls which mesh representation is used in the scene. Select the desired representation from the drop-down list. The **Solution Time**, **Iteration**, and **Time Step** annotations can display data from a recorded solution. The annotations update automatically as you retrieve results from the solution history file. See What Are Representations?.	
Coordinate System	Selects the type of coordinates to use for position (not available for **Scene Grid** type).	
	Normalized Viewport	Shows x-y values as fractions of the length and height of the *Graphics* window display, respectively. The origin is the lower left-hand corner of that display.
	World	Shows x-y-z values as global coordinates. As you rotate the object, the annotation keeps its position relative to that object.
Interactive	Controls whether the annotation can be moved in the scene display with the mouse (not available for **Scene Grid** type).	
	Activated	Allows you to move the annotation in the display.
	Deactivated	Locks the position of the annotation in the display.

Applied Annotation Pop-Up Menu

Edit Annotation...	Opens the table editor for the parent annotation. See Using the Table Editor.
Remove from Scene	Removes the applied annotation from the active scene.

Annotations Manager Properties

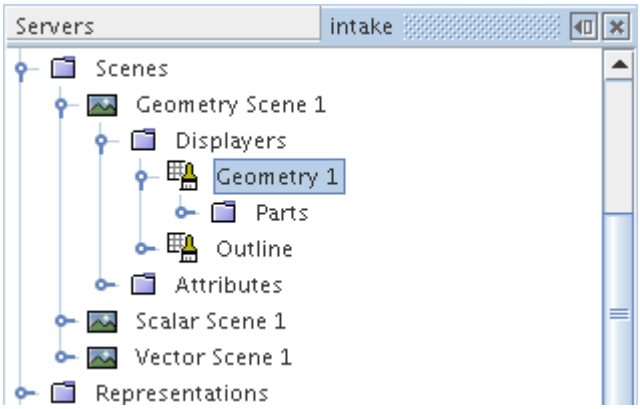
To add or remove annotations in a particular scene, access the properties of the **Annotations** manager node.

Annotations	A list of the annotations added to the scene. Editing the property accesses the in-place object selector to add or remove annotations. See Activating the Object Selection Dialog.

Displaying Geometry and Mesh

Geometry displayers are used to display the outline and surfaces of the parts.

The **Geometry** displayer node controls the appearance of the surface and mesh aspects of the parts in the scene. This node has its own properties and shares a pop-up menu with other types of displayers.



Use the displayer to specify whether the model appears with outlines, a complete grid or shaded surfaces or a mixture of these aspects.

Contents:

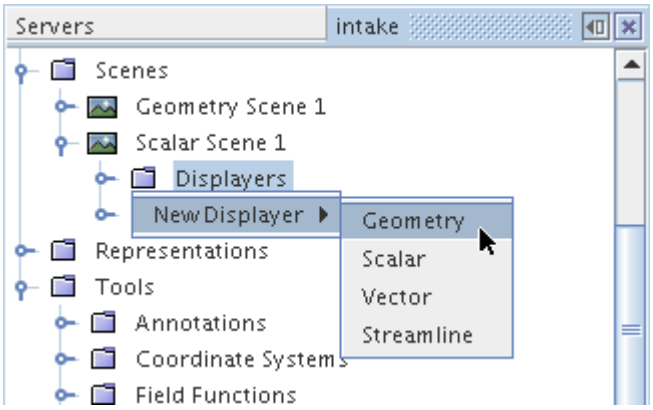
Creating Geometry Displayers
Using Surface and Outline Displays
Displaying Mesh Outlines and Surfaces
Displaying Feature Lines
Geometry Displayer Properties

Creating Geometry Displayers

Two geometry displayers are created automatically when a geometry scene is created. Alternatively, you may wish to add a geometry displayer to any scene.

The first displayer is set to display the outline of all the boundary parts whereas the second is set to show the surfaces of all the boundary parts. A similar setup exists when you create a mesh scene, with the surface geometry displayer also set to display the mesh grid. In addition, a geometry displayer is created automatically when you create either a scalar scene or a vector scene. In these cases, the geometry displayer is set to display the outline of the boundary parts.

To add a geometry displayer manually, open the **Scenes** node and the node of the chosen scene. Right-click on the **Displayers** node and select **New Displayer** > **Geometry**.



By default the geometry displayer will contain no parts after creation and you must add parts to the displayer. This feature allows you to choose only those parts of the model that you wish to view.

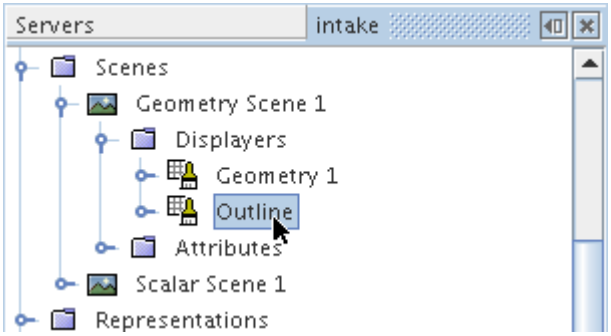
The default name of the displayer is based on its type. However, renaming the displayer is possible.

Using Surface and Outline Displays

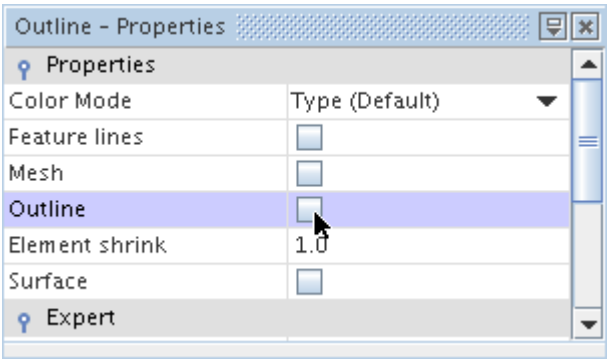
Geometry displayers control the visibility of surfaces and outlines in the scene display, and allow you to show or hide them.

Often, outlines are visible through a geometry displayer named **Outline**, which accompanies another displayer in most types of scenes (any default geometry, scalar or vector scene).

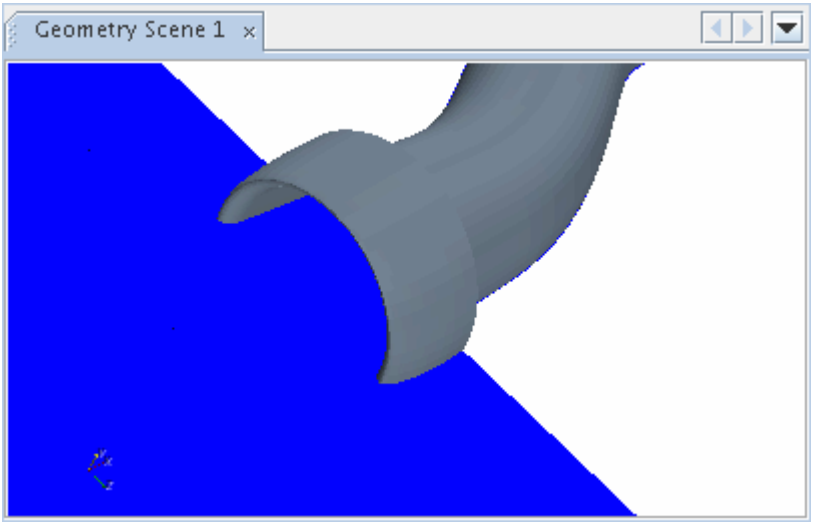
To make the outlines invisible, expand the **Scenes** node and expand the scene that includes a geometry displayer. Expand the **Displayers** node and select the geometry displayer node, commonly named **Outline**. (In an empty scene in which you create a geometry displayer, the default name of the displayer might be different but this procedure works the same way.)



In the *Properties* window, deactivate the checkbox of the *Outline* property.

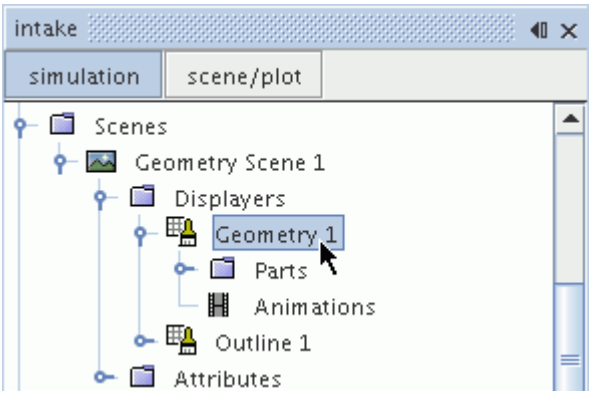


In the scene display, the outlines are gone. In this example with a geometry scene, only the surfaces are displayed by the other geometry displayer. (The boundary type of the part dictates the color of the surface unless this has been changed.)

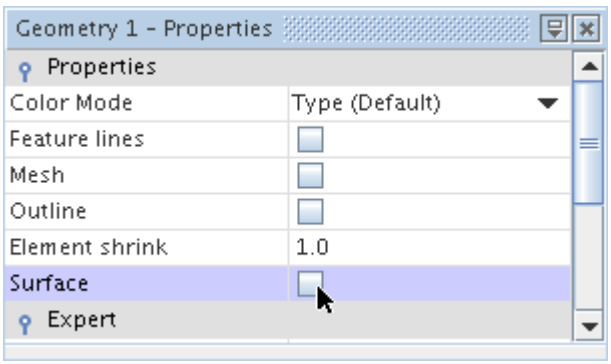


To restore the outlines, simply activate the same checkbox.

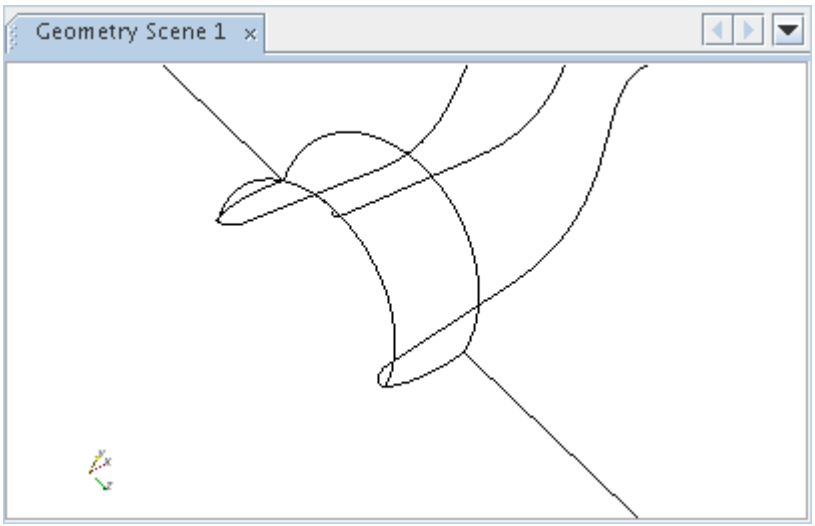
To make the surfaces invisible, expand the **Scenes** node and expand the scene that includes the geometry displayer. Expand the **Displayers** node and select the geometry displayer node.



In the *Properties* window, deactivate the checkbox of the *Surface* property.



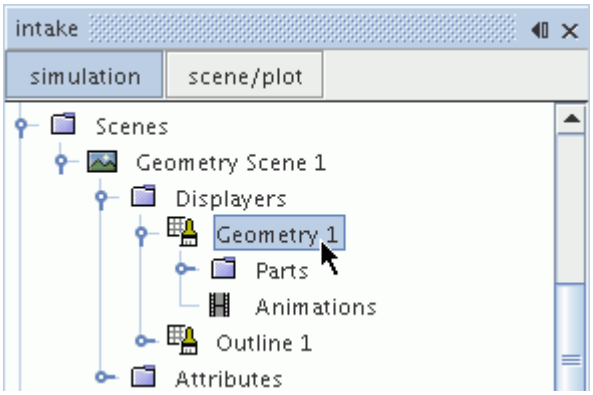
In the scene display, the surfaces are gone. To restore them, simply activate the same checkbox.



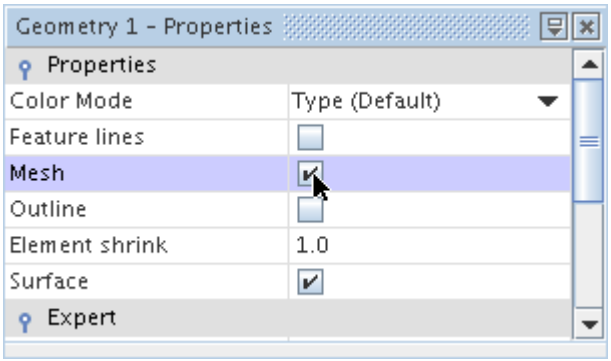
Displaying Mesh Outlines and Surfaces

You may display the mesh on the surface of any part in a geometry displayer.

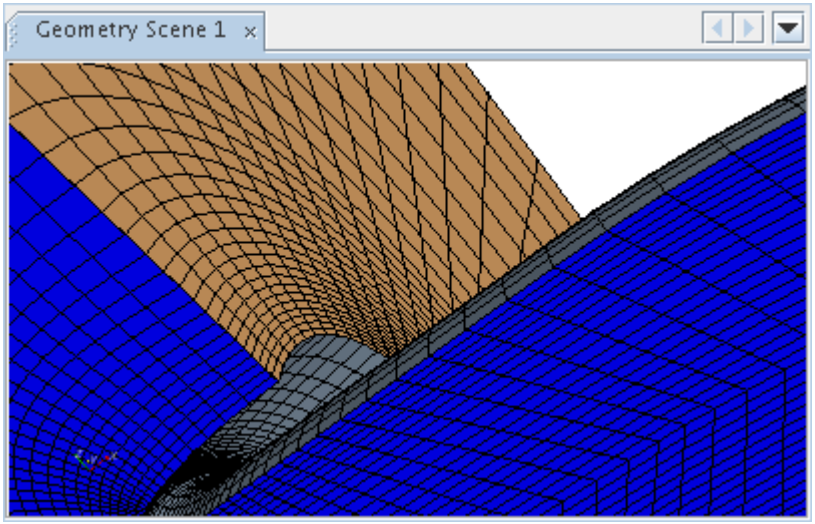
To display a mesh in a geometry displayer, expand the **Scenes** node and expand any scene that includes a geometry displayer (any default geometry, scalar or vector scene, or an empty scene in which you create a geometry displayer). Expand the **Displayers** node and select the geometry displayer node.



In the *Properties* window, activate the checkbox of the *Mesh* property.



The mesh is visible in the scene display.

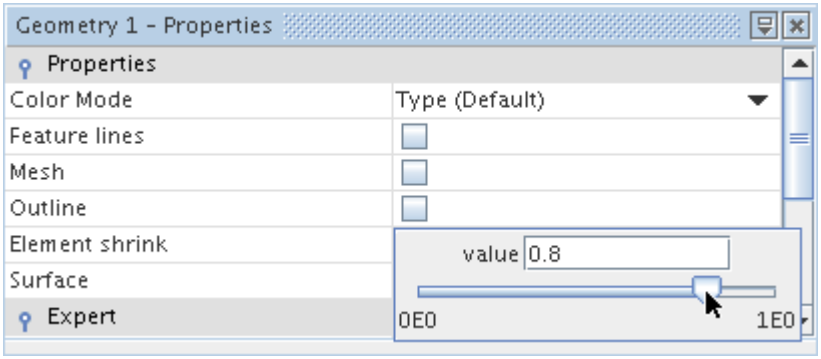


The *Display Mesh* property of the scalar displayer works the same way.

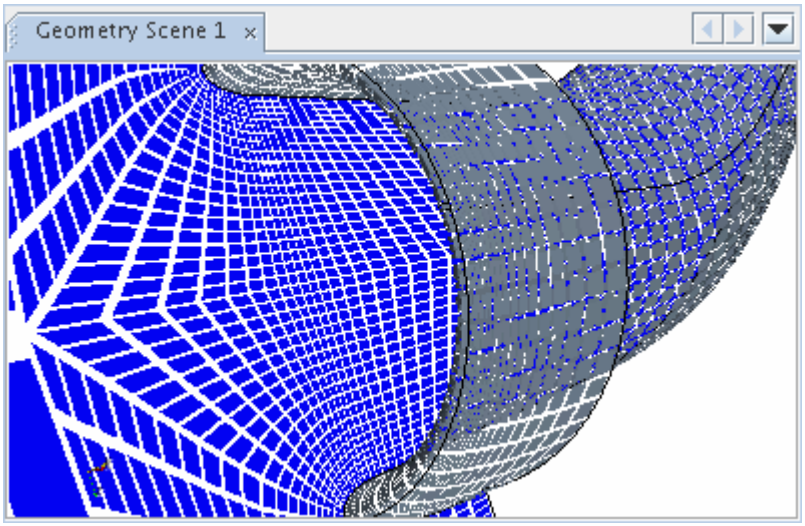
In addition to displaying the mesh outlines, the surfaces of each cell may be illustrated by shrinking each surface element.

Begin by making sure that the surface is visible.

To shrink the elements, click the right column of the *Element shrink* property in the *Properties* window. In the slider box that appears, select a value between `0` and `1`, where `0` results in no surface and `1` results in whole surface elements. You can also enter your selection directly in the **value** text box above the slider. Click anywhere in the workspace around the slider box to close it.



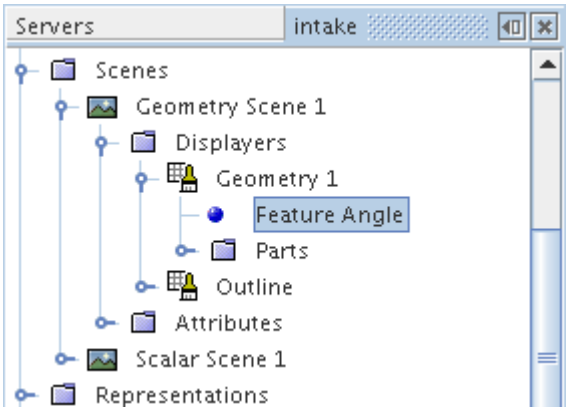
The separate cell elements are visible in the scene display.



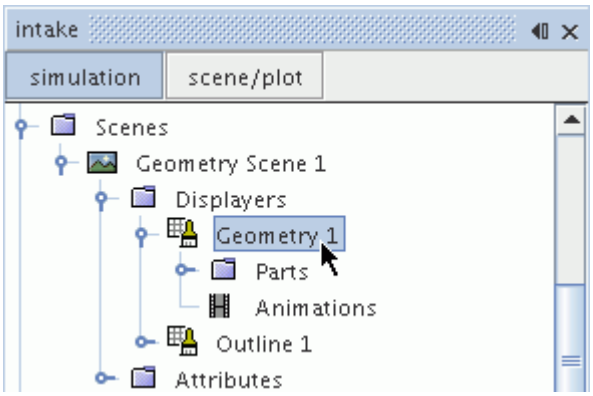
Displaying Feature Lines

You can capture and display feature lines. This enables you to display mesh edges that are extracted by feature angle (the angle between normal vectors of adjacent polygons).

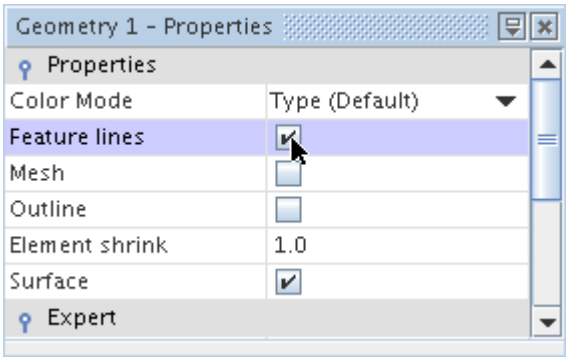
The **Feature Angle** node controls the feature lines. This node has properties and is subject to common menu items. The node appears only when the *Feature lines* property of the displayer is activated.



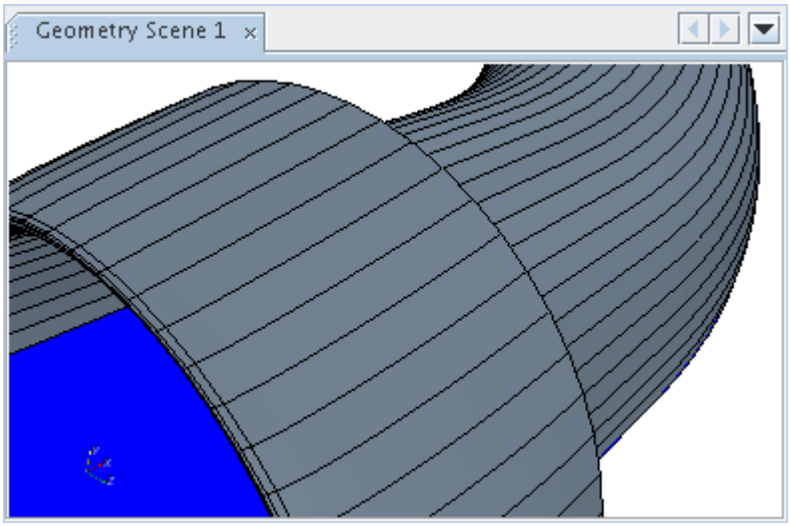
To make this object available, expand the **Scenes** node and expand any scene that includes a geometry displayer (any default geometry, scalar or vector scene, or an empty scene in which you create a geometry displayer). Expand the **Displayers** node and select the geometry displayer node.



Activate the checkbox of the *Feature lines* property in the *Properties* window.



Once you activate the node, you can use its *Feature Angle* property to change the angle between surface elements that defines a feature. The smaller the angle, the more edges are extracted.



Feature Angle Properties

Feature Angle	Sets the minimum angle between polygonal facets when displaying feature lines.

Geometry Displayer Properties

Use the properties of the geometry displayer to control the appearance of the surface, outline, and mesh lines.

Outline	Controls the appearance of the outlines.	
	Activated	Shows the outlines of the parts.
	Deactivated	Hides the outlines of the parts. When you choose this option and activate the *Surface* property, you generate a more realistic image of the model.
Surface	Controls the appearance of parts of the model.	
	Activated	Activates a solid coloring of the parts. When you choose this option, you generate a more realistic image of the model.
	Deactivated	Leaves no shading; the surfaces are completely transparent except for outlines, if outlines are activated.
Mesh	Controls the appearance of the mesh grid.	
	Activated	Shows the complete grid and adds the **Mesh Color** subnode to the

		displayer node to control the color of the mesh lines. This option lets you examine the density and cell structure of various parts of the model, and make judgments about accuracy.
	Deactivated	Hides the grid except for the outlines. When you choose this option and activate the *Surface* property, you generate a more realistic image of the model.
Feature Lines	Controls the appearance of the feature lines.	
	Activated	Adds the **Feature Angle** subnode to the displayer node to show feature lines.
	Deactivated	Hides the feature lines and removes the **Feature Angle** subnode. When you choose this option and activate the *Surface* property, you generate a more realistic image of the model.
Element Shrink	Contracts the vertices of each polygon toward the polygon centroid when set below 1.0. This lets you see the surface cells of the model more clearly. If you enter `0`, the shading disappears; only the edges are visible if they are activated. The *Surface* property must be activated in order for the effects of this function to be visible. This property can be set with a slider.	
Color Mode	Controls the arrangement of colors.	
	Type (Default)	Colors the part surfaces according to their boundary type.
	Constant	Adds the **Color** subnode to the displayer node to display surfaces using a custom color.
	Distinguish Regions	Colors the surfaces by region, so that different regions can be identified.
	Region Type	Colors the surfaces by region type, that is, fluid, porous or solid.
	Distinguish Inputs	Colors regions by boundaries, or parts by part surfaces - depending if regions or parts are displayed in the scene.

Opacity	Adjusts the transparency of the part. The settings range from 0.0 (invisible) to 1.0 (opaque), and can be controlled with a slider. This property allows you to show an overlay of different parts of the model.

Geometry Displayer Expert Properties

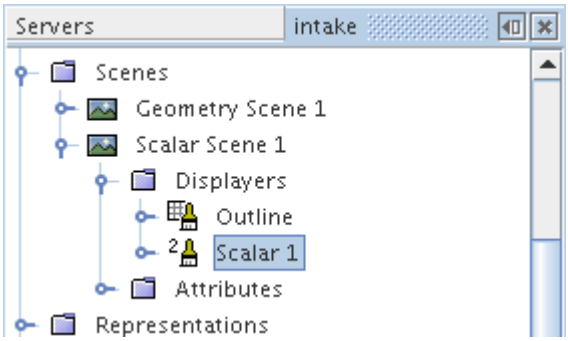
Representation	Selects a representation. Additional selections would appear as representations are added to the simulation.	
Point Size	Sets the point size in pixels. This property can be set with a slider.	
Line Width	Adjusts the width in pixels of lines that are used in the display, depending on the type of displayer. For example, in a scalar displayer, this setting would be effective only if the *Contour Style* property were set to `Line`. This property can be set with a slider.	
Specular Highlight Mode	Specular highlighting is the reflection of a light source on a shiny surface (such as polished metal).	
	Material	Uses the color of the material (default). For example, on a black surface, no specular highlighting is visible.
	Light	Uses the color of the light. To specify this color, see Controlling the Color of Lighting Objects.
Lighting	Controls the lighting of the part. Accessing the property customizer opens the *Lighting* dialog.	
Data Focus	Selects a data focus object to apply to this displayer. The choice of data focus objects depends on the total number of these objects in the simulation. When a data focus object is selected, a data focus sub-node is added to this node. See Exploring Solutions with Data Focus.	
Smooth Shade	Controls the transition between segments of the image.	
	Activated	Makes the transitions smooth for a more realistic effect.
	Deactivated	Leaves this function off.
Ignore Clip	Controls whether the parts in the displayer are affected by any enabled clip plane.	
	Activated	Shows the parts unaffected by the clip plane.
	Deactivated	Shows the parts that are trimmed by the clip plane.
Face Culling	Controls whether and how to cull polygons.	
	None	Does not cull any polygons.

	Front	Culls (omits from the display) all polygons that are oriented toward the front of the view.
	Back	Culls (omits from the display) all polygons that are oriented toward the back of the view.
Transform	Selects a transform to apply to all parts in the displayer. **None** is the default selection. Additional selections would appear when you create transforms. This feature allows for the creation of symmetric and periodic repeats, as well as arbitrary transforms to be used.	

Displaying Scalar Quantities

Scalar displayers are used to display scalar field functions or field function monitors on parts using contours.

The **Scalar** displayer node, which has its own properties and shares a pop-up menu with other types of displayers, controls the display of data along the surface of a part.



In the icon, the numeral next to the paint brush signifies the depiction of quantities.

Contents:

Creating Scalar Displayers
Changing the Scalar Quantity
Changing the Contour Style
Understanding Smoothed Values
Labeling Line Contour Values
Changing the Scalar Range
Changing the Scalar Units
Working with Point Scale Modes
Copying Properties of a Scalar Field
Defining a Custom Scalar
Scalar Displayer Properties

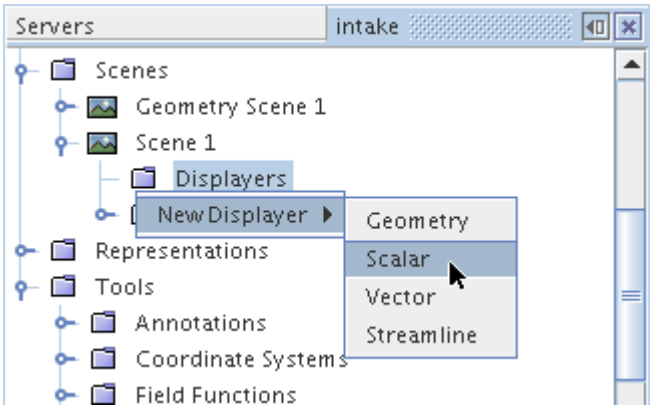
See Also:

Using the Scalar and Vector Bars

Creating Scalar Displayers

A scalar displayer is created automatically when a scalar scene is created. Alternatively, you may wish to add a scalar displayer to any scene.

To add a scalar displayer manually, open the **Scenes** node and the node of the chosen scene, right-click the **Displayers** node and select **New Displayer** > **Scalar**.



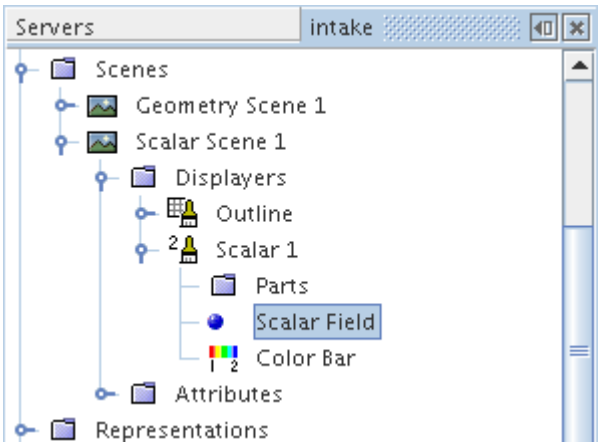
By default the scalar displayer will contain no parts after creation and you must add parts to the displayer. This feature allows you to choose only those parts of the model on which you wish to display scalar quantities.

The default name of the displayer is based on its type. However, renaming the displayer is possible.

Changing the Scalar Quantity

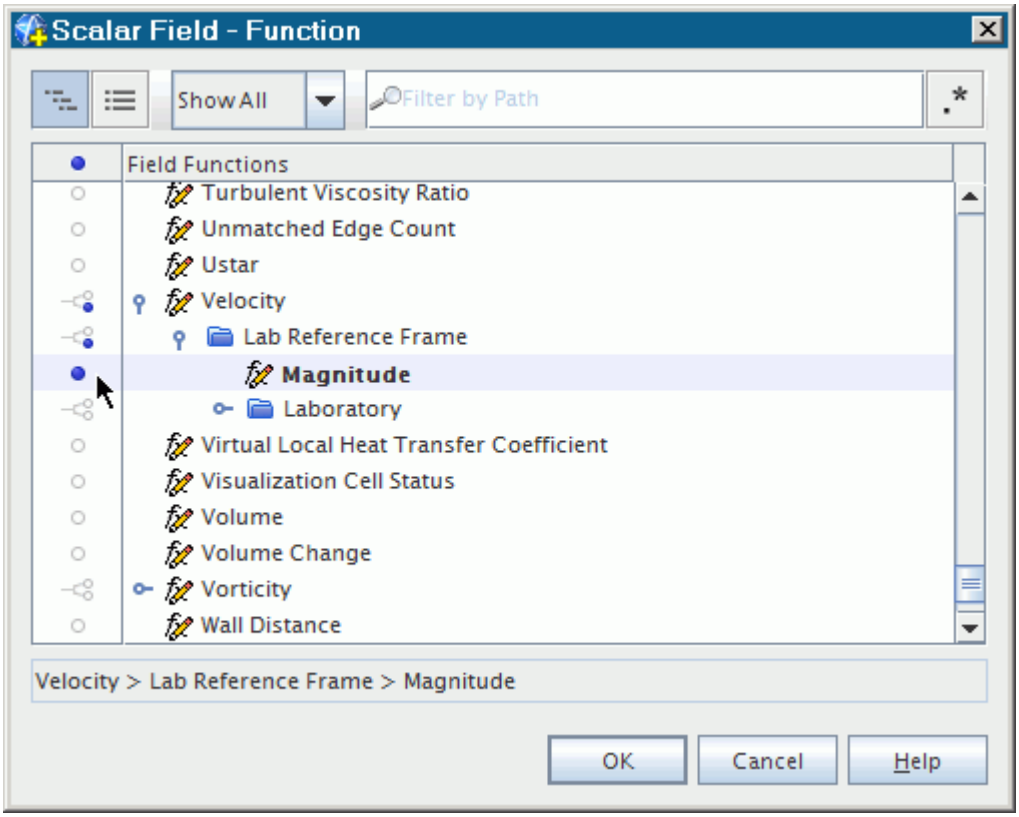
The scalar displayer and streamline displayer have no default scalar, so you must select one from a list of scalar field functions to display data in a scene.

The list of field functions is available through the **Scalar Field** node, which has properties and is subject to common menu items.

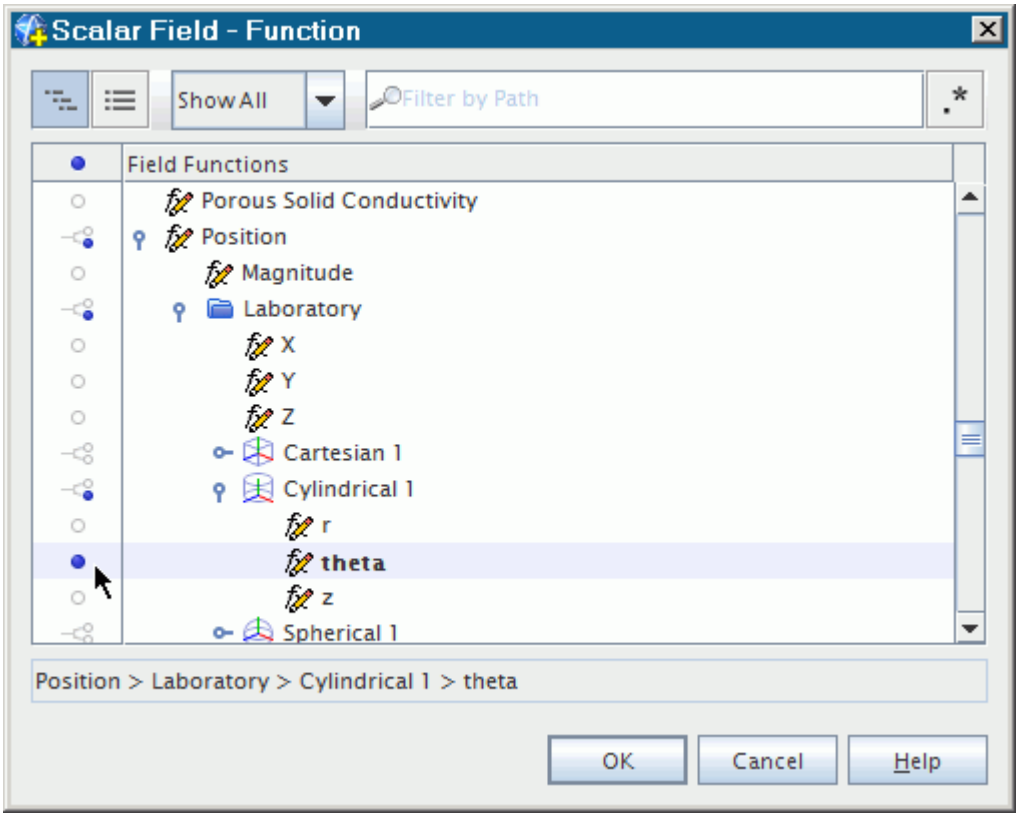


To access the **Scalar Field** node, expand the following nodes: the **Scenes** node, the node of a scene with a scalar displayer, the **Displayers** node within that scene, and the scalar displayer node.

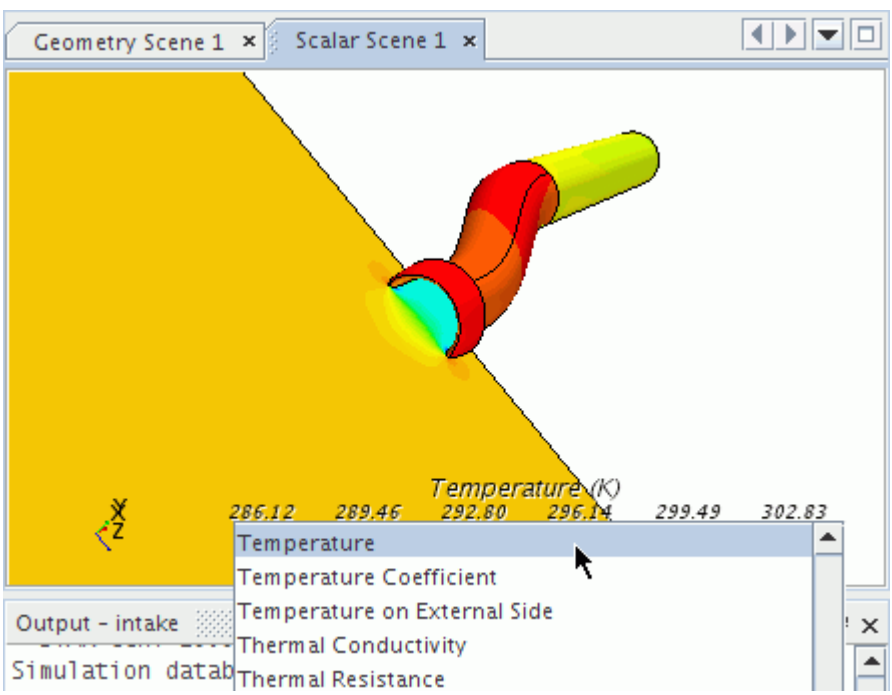
Once you select the node, the scalar field function can be chosen from the list in the object selection dialog. To activate the dialog, edit the *Function* property.



This list also allows you to select the individual components of certain functions, such as positions and centroids, in local coordinate systems.



You may also choose the scalar quantity using a drop-down list by right-clicking the scalar bar in the visualization display. This may be more convenient when you manipulate the scene.



Scalar Field Properties

Function	Selects the type of scalar field function or field function monitor to show, such as temperature or velocity magnitude. See What Is a Field Function Monitor? and Field Functions.	
Auto Min	Indicates the minimum value among all parts in the displayer (read-only).	
Auto Max	Indicates the maximum value among all parts in the displayer (read-only).	
Auto Range	Determines whether to set the range of values to display automatically. See Changing the Scalar Range.	
	Off	Does not use auto ranging, rather use the values that are set in *Min* and *Max*.
	Min and Max Values	Automatically determines the range from the field function data.
	Min Value	Automatically determines the minimum value from the field function data. The property switches automatically from `Min and Max Values` to this setting if you make an entry for *Max Value*.
	Max Value	Automatically determines the maximum value from the field function data. The property switches automatically from `Min and Max Values` to this setting if you make an entry for *Min Value*.
Auto Range Mode	Chooses the domain for auto-range minimum and maximum value computation that is used for color coding.	
	Displayer Parts	Parts added to the particular displayer.
	Displayer Regions	Regions from which the displayer extracts data.
	All Regions	All regions in the simulation.
Clip	Determines how to display values outside of the range that is specified by *Min* and *Max*. See Changing the Scalar Range.	
	Off	Values above *Max* are displayed in the maximum color, and values below *Min* are displayed in the minimum color.

	Below Min, Above Max	Does not display values outside the range.
	Below Min	Does not display values below *Min*.
	Above Max	Does not display values above *Max*.
Units	Selects the type of units to display. See Changing the Scalar Units.	
Min	Sets the minimum value in the displayed range.	
Max	Sets the maximum value in the displayed range.	

Changing the Contour Style

The *Contour Style* property has several options for showing scalar contours.

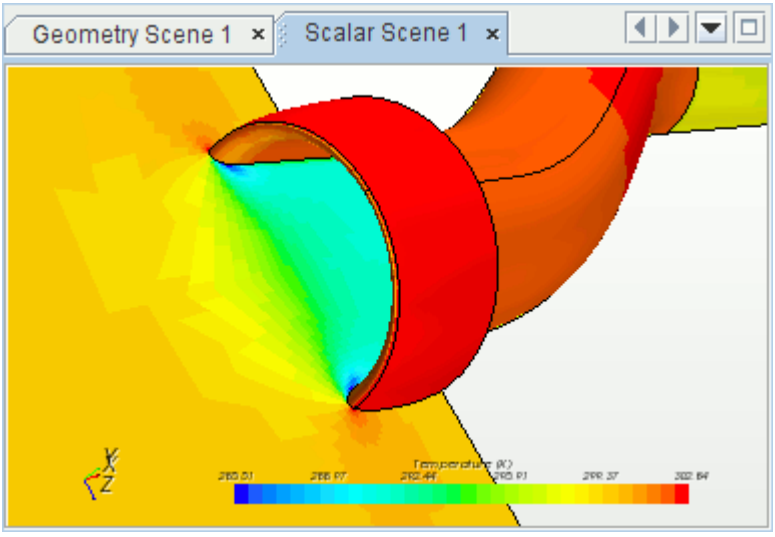
The default setting is `Automatic`. This setting is an automatic selection of the most appropriate contour style, depending on the specified scalar function and parts.

To change the contour style of the displayer, open the **Scenes** node, the node of the particular scene which includes a scalar displayer, and the **Displayers** node within that scene. Select the node of the scalar displayer.

Select the new style from the drop-down list of the *Contour Style* property in the *Properties* window.

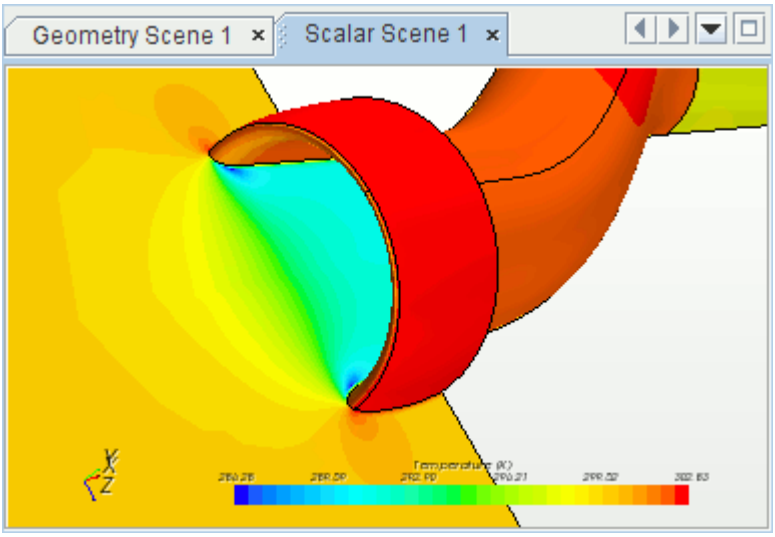
The property can be set as follows:

- `Filled`



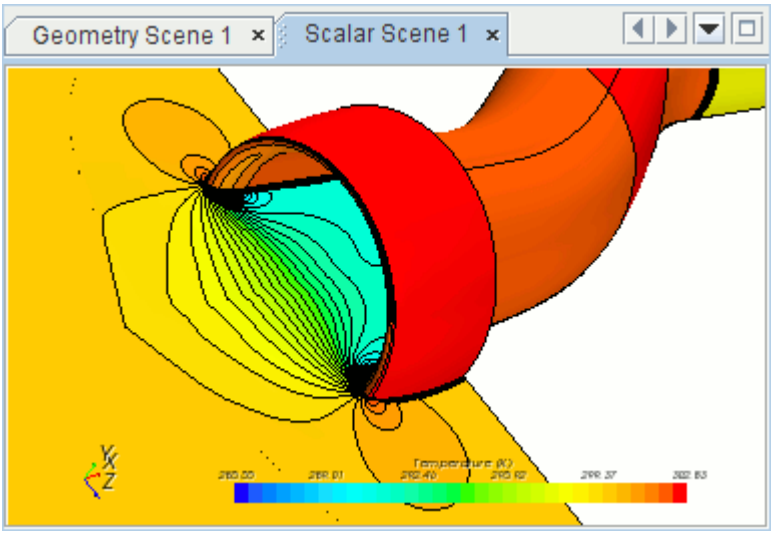
The filled contour style fills each cell face of the surfaces that are displayed with the scalar value stored at the center of its corresponding cell.

- `Smooth Filled`

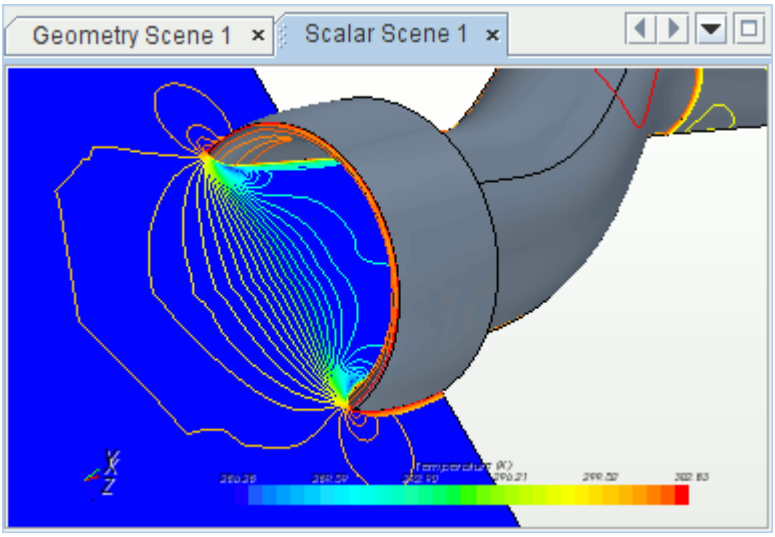


The smooth filled contour is generated using vertex values rather than cell centroid-based values and appears smoother than the default filled contour style. However, for most scalar fields values are interpolated from the cell centers to the vertices (or nodes) of each cell. Thus, data extraction with the smooth-filled option is more expensive in terms of computing power.

- `Smooth Filled + Lines`

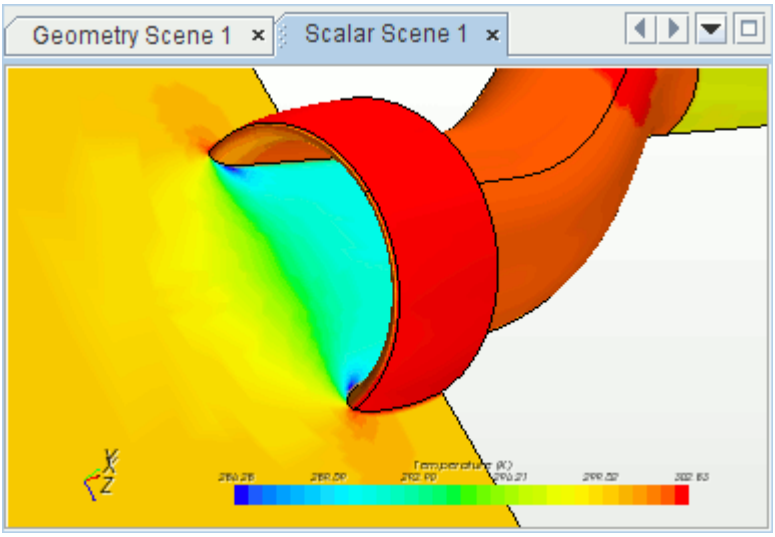


- `Line` (The illustration below shows a scene with a geometry displayer that is added to provide a background for the lines.)



The line contour style displays isolines of the vertex-based scalar quantity that is colored with the value of the scalar.

- `Smooth Blended`



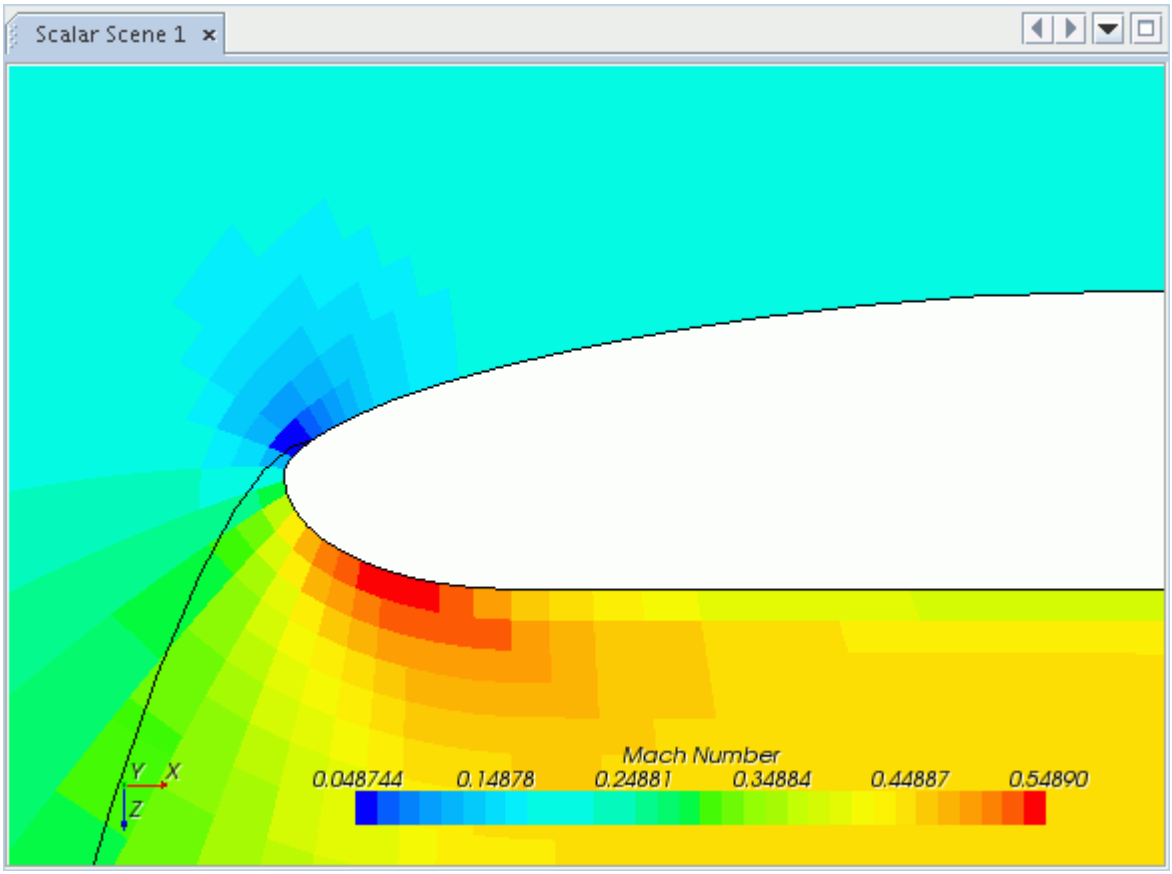
The smooth blended option is based on vertex values, similar to the smooth filled option. However, it is less computationally expensive than the smooth filled option due to the technique used to blend colors across the cell surfaces.

Understanding Smoothed Values

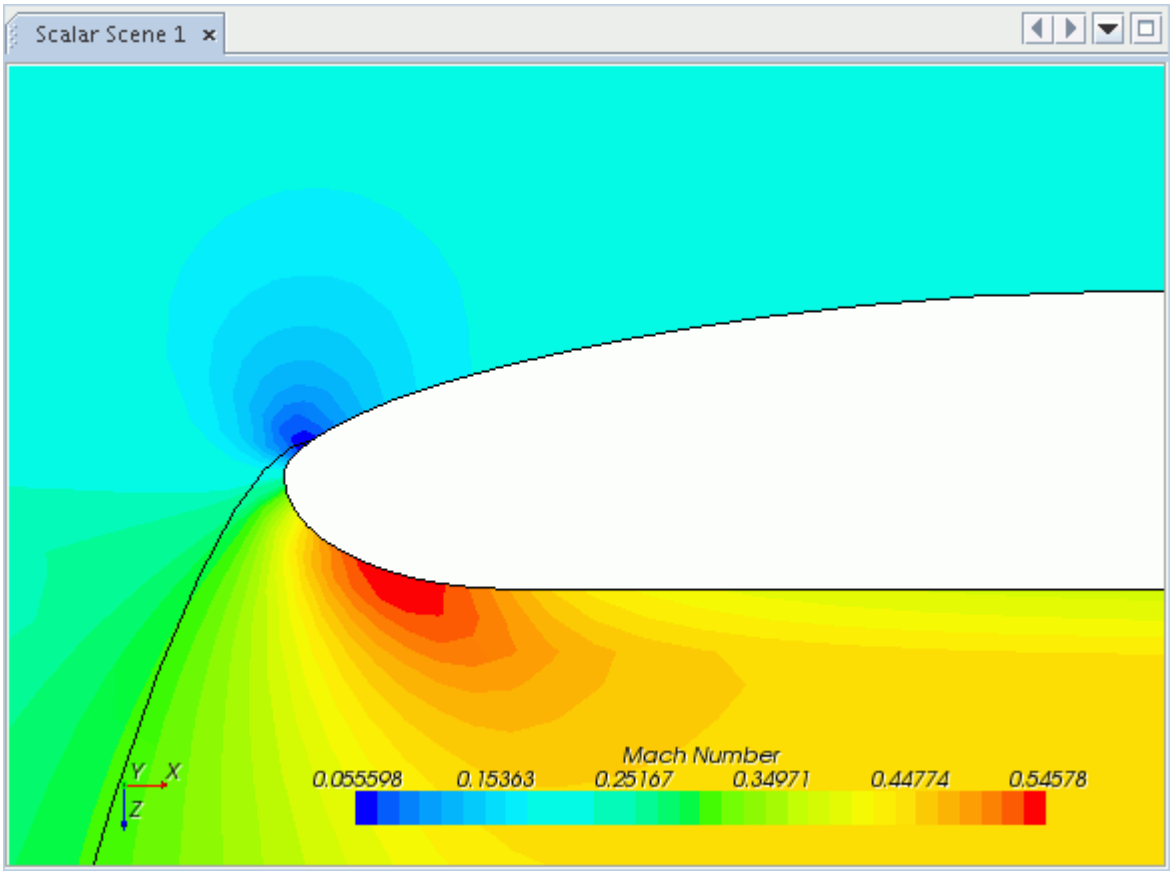
To avoid the default stepped appearance of a scalar scene, STAR-CCM+ also provides options that apply smoothing to scalar scenes.

When STAR-CCM+ runs a simulation, it stores many computed quantities at the location of the cell centroid. When you create scalar scenes, the default is to display results based on these cell center values only. While

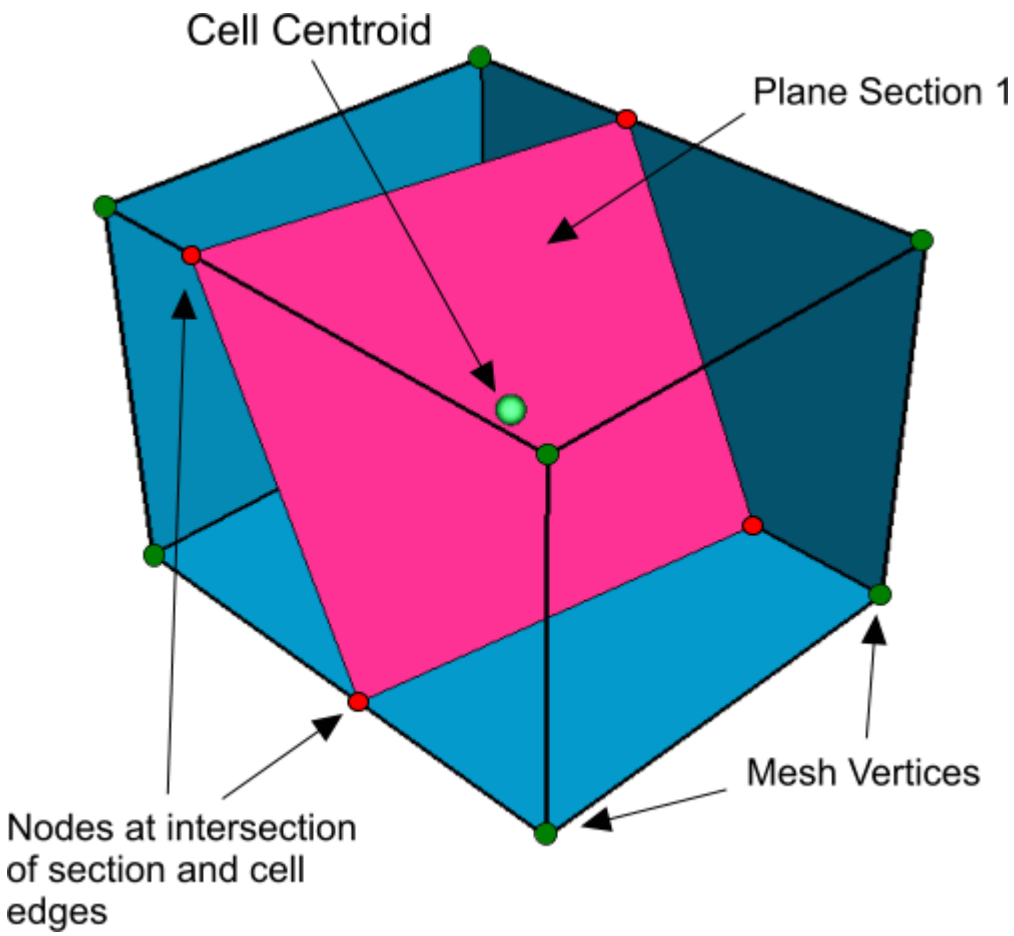
this option is essential for accurate reporting of computed quantities, it results in scalar scenes that have a stepped appearance. An example of this is shown below.



When the smooth filled option is chosen, for example, the following image results.



To obtain the nodal values that are used in smoothed scalar scenes, STAR-CCM+ first maps the cell center values (where the raw simulation data is stored), to the vertices of the mesh. It then applies a weighted interpolation between these vertex values and the points at which a chosen section (for example, a plane section derived part) cuts through the edges of the mesh between the mesh vertices. This is illustrated in the following diagram.



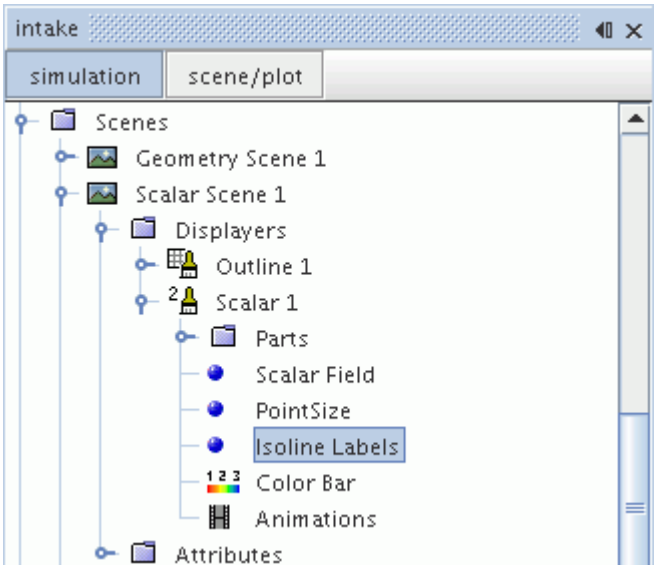
When you choose one of the smooth or blended options for the scalar displayer, the results you see is based on this mapping and interpolation.

Note: If you apply the smooth filled option to a section to visualize field variables that are only defined on boundaries, the face data is interpolated onto the boundary vertices to create vertex data. The section then takes up these values from the boundary vertices.

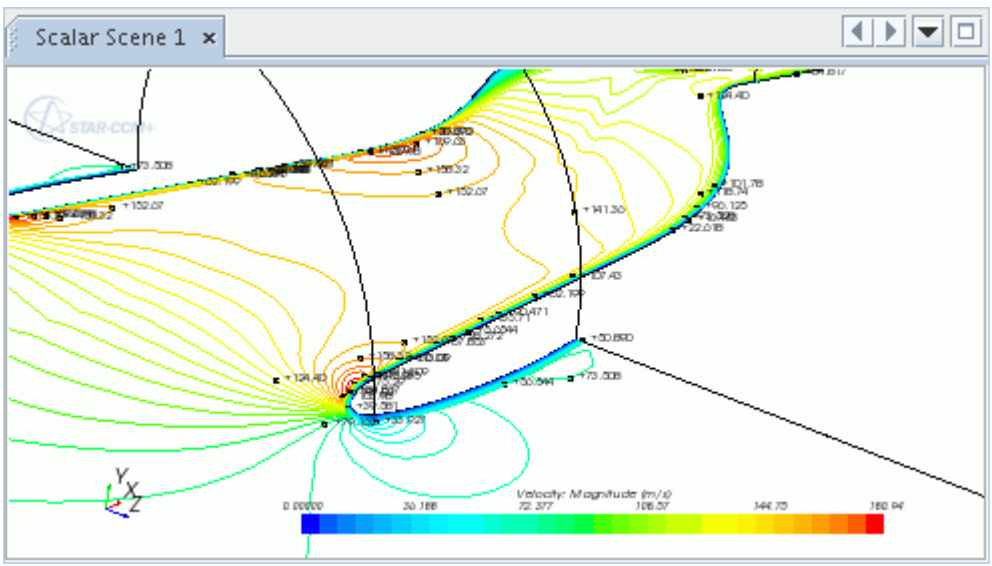
Labeling Line Contour Values

You can plot the actual values of the scalar contour lines on the lines themselves.

When you set the Contour Style property of the scalar displayer to either `Smooth Filled + Lines` or `Line`, the **Isoline Labels** node is made available in the scalar displayer node. This node has properties and is subject to common menu items.



To make the labels appear in the scene, activate the checkbox in the *Visible* property of the node.



Isoline Labels Properties

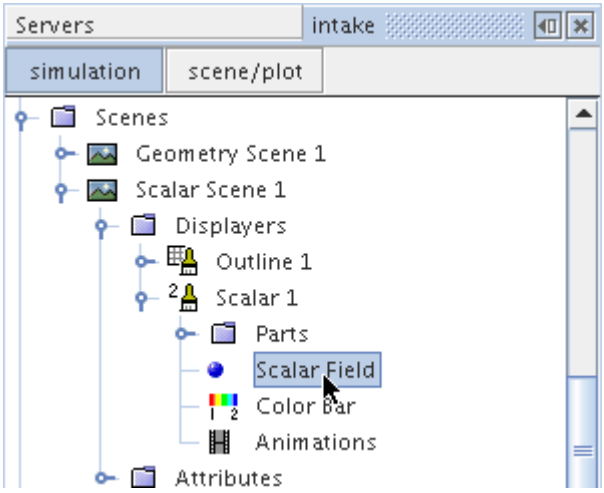
Point Size	Sets the point size in pixels.	
Visible	Shows or hides the labels and points in the scene.	
	Activated	The labels and points are visible.
	Deactivated	The labels and points are hidden.
Label Font	Sets the font of the labels.	
Label Style	Uses a drop-down list to display the labels as normal, bold, italic or bold and italic.	

Label Color	Sets the color to display the labels. The property customizer executes the define color action. See Defining Colors.	
Label Precision	Specifies the total number of digits in the label. Raising this number adds decimal places.	
Shadow	Activates a shadow effect for the labels.	
	Activated	Shadow effect is on.
	Deactivated	Shadow effect is off.
Point Color	Sets the color to display the points. The property customizer executes the define color action. See Defining Colors.	
Label Size	Sets label size as a percentage of window size.	

Changing the Scalar Range

By default the range of the scalar quantity is automatically based upon the range of the field function on the parts in the displayer. However, the range can also be set manually.

Expand the following nodes: the **Scenes** node, the node of a scene with a scalar displayer, the **Displayers** node within that scene, and the scalar displayer node. Select the **Scalar Field** node.

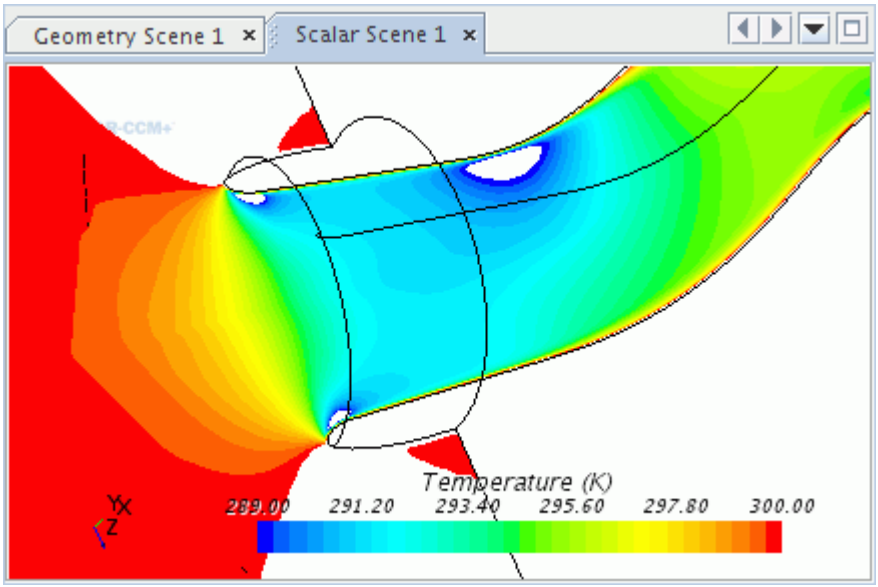


In the *Properties* window, enter the minimum value under the *Min* property.

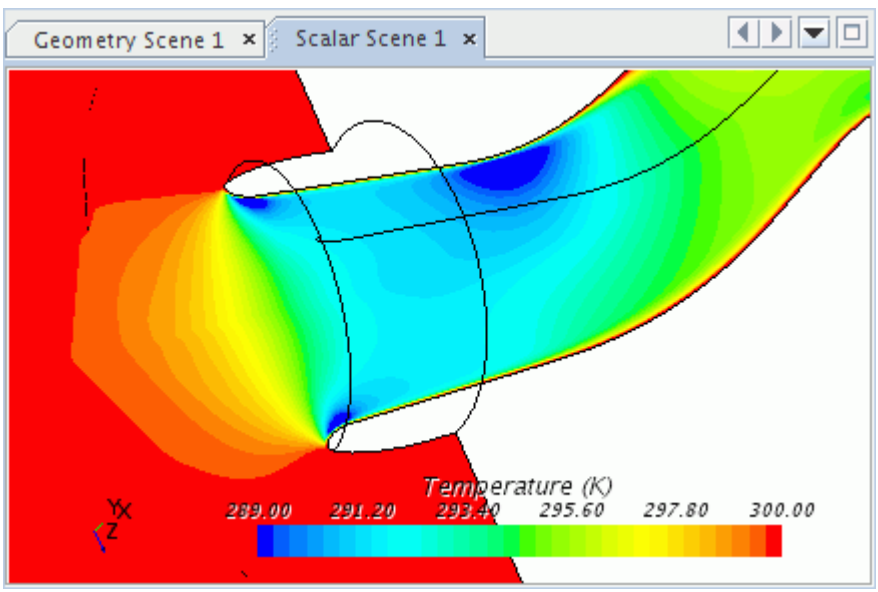
This causes the *Auto Range* property to switch to `Max Value` automatically, and the result appears in the scalar scene display. If you set the *Max* property first, the *Auto Range* property setting switches to `Min Value`.

Now enter a value for the *Max* property, and the *Auto Range* property sets itself to `Off`, since you have manually set both values.

If the manually entered range is smaller than the bounds of the scalar quantity, the contours are clipped by default.



If you select `Off` in the drop-down list of the *Clip* property, scalar quantities out of range appear in the color contours at the extremes of the color gradient.



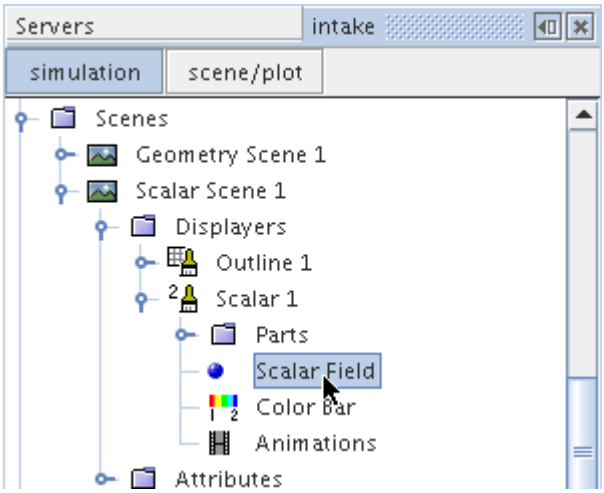
Alternatively, you can set the *Clip* property to `Below Min` to exclude only the scalar quantities below the range, or `Above Max` to do this with the quantities above the range.

You can revert to the automatic range setting anytime by selecting `Min and Max Values` in the drop-down list of the *Auto Range* property.

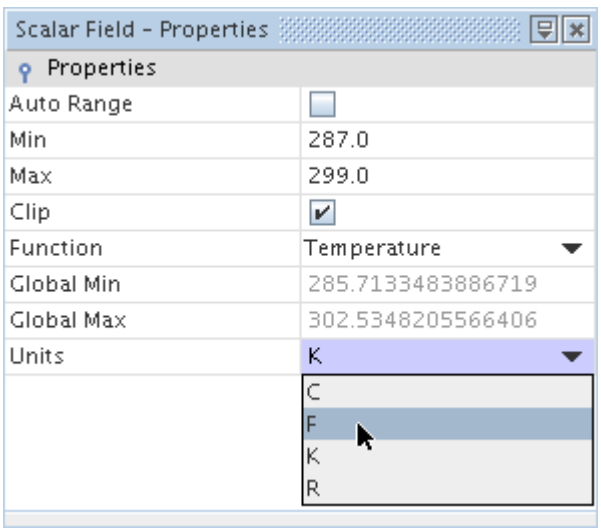
Changing the Scalar Units

If alternative units are available, the units displayed with the scalar legend in the scene can be changed.

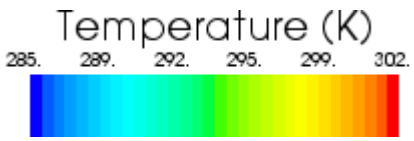
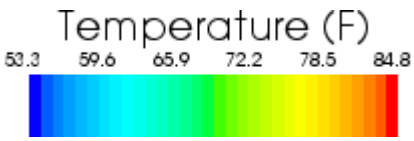
Expand the following nodes: the **Scenes** node, the node of a scene with a scalar displayer, the **Displayers** node within that scene, and the scalar displayer node. Select the **Scalar Field** node.



Select a unit from the drop-down list of the *Units* property in the *Properties* window.



The units along the scalar bar change accordingly.

	
Previous units	New units

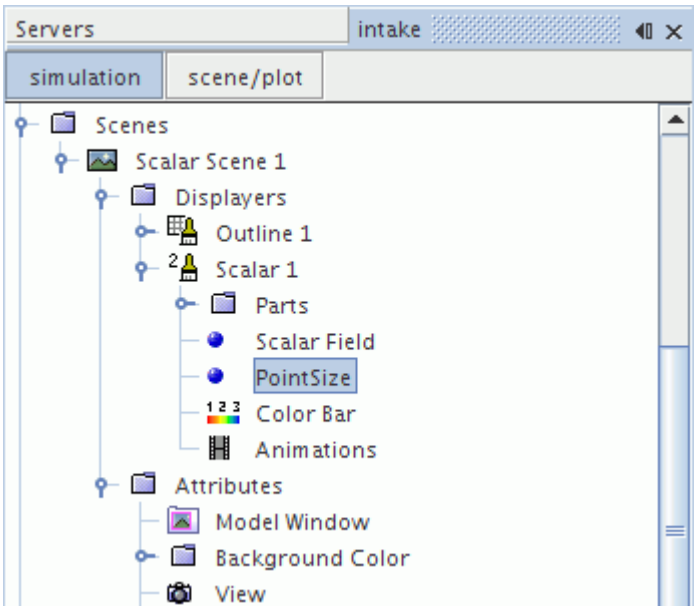
Working with Point Scale Modes

The screen display size of parts that are represented by points (such as Lagrangian parcels or points on a presentation grid) can either be set to a constant pixel size or be scaled according to a scalar function.

The display size options are available in the *Point Scale Mode* property of the scalar displayer. The option you select depends largely on what you are displaying.

What Is the Point Size?

This parameter, represented by the **PointSize** node, is the default setting. It sets the screen display size of a point in pixels in its properties, independent of the physical size of the point. This option is likely to be useful when you want to display the points on a presentation grid. Lagrangian particles or parcels are not sized according to their physical size if this option is chosen.

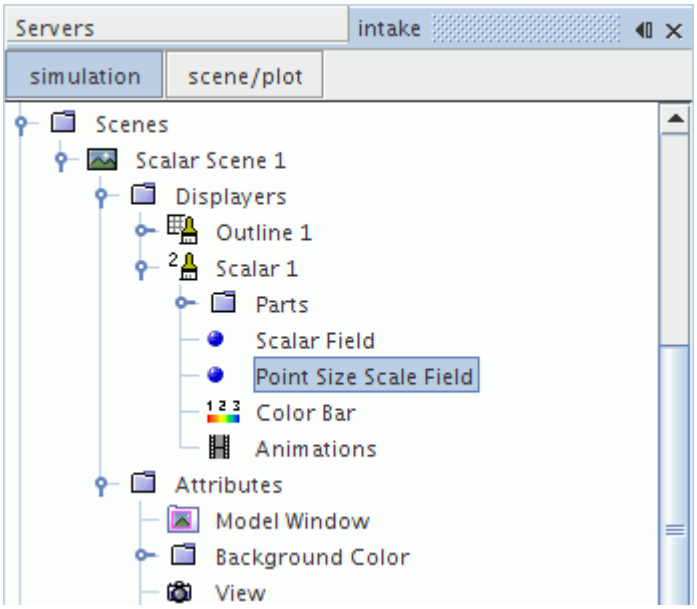


PointSize Properties

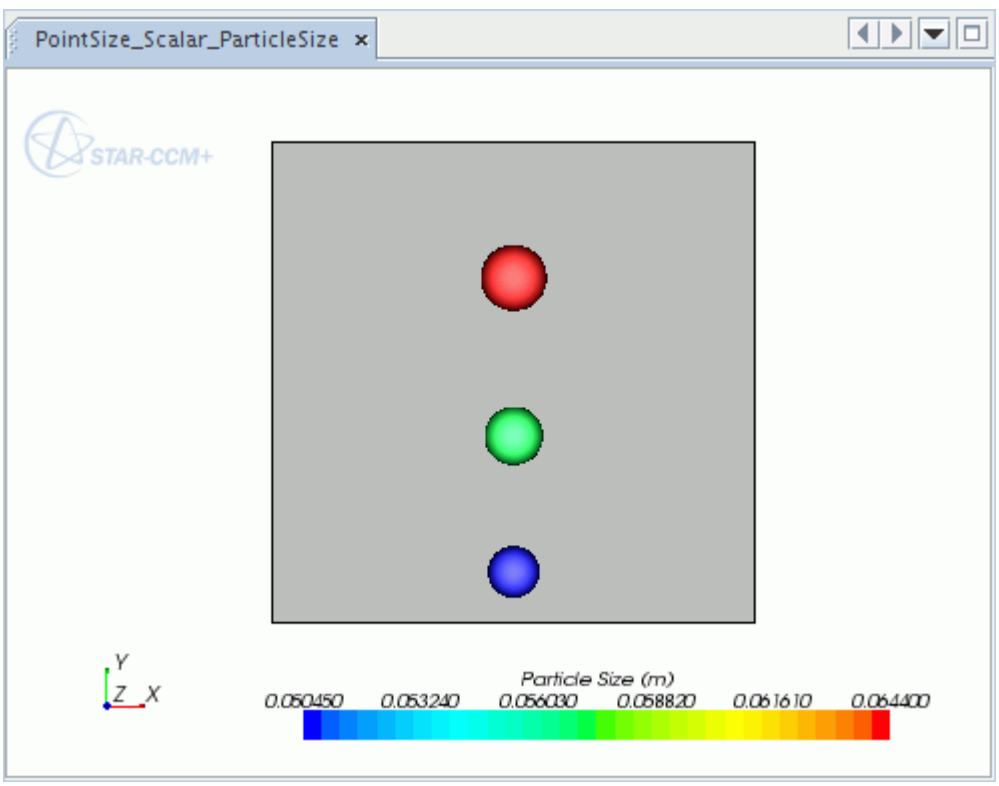
Size	Sets the point size in pixels.

What Is the Point Size Scale Field?

The Point Size Scale Field lets you scale a point's screen dimension based on scalar quantity, such as particle size or fluid temperature. It is represented by the **Point Size Scale Field** node, which is activated when you set the *Point Scale Mode* property of the scalar displayer to `Scalar`.



In the properties, select a scalar field function on which to base the screen point size. The Point Size Scale Field can be useful in cases involving particles in Discrete Element modeling or parcels in Lagrangian Multiphase modeling. It is recommended that you use the scalar Particle Size when you wish to plot particles according to their actual size. The size of the particle on the screen is calculated relative to the overall dimensions of the mesh. An exaggerated example of this is shown below.



If you want to scale by something else (such as velocity), map the velocity range to a physical size range, where the physical size range sets the minimum and maximum size of particles relative to the mesh dimensions. Activate the *Map values* property to activate this mapping. The formula that is used for the mapping is:

$$\text{Point Size} = (\text{Scalar Value} - \text{Min})\left(\frac{\text{Map size maximum} - \text{Map size minimum}}{\text{Max} - \text{Min}}\right) + \text{Map size minimum}$$

which means that is mapping values in interval [Min, Max] to interval [Map size minimum, Map size maximum].

Point Size Scale Field Properties

Function	Selects the type of scalar field function or field function monitor to show, such as temperature or velocity magnitude. See What Is a Field Function Monitor? and Field Functions.	
Auto Min	Indicates the minimum value among all parts in the displayer (read-only).	
Auto Max	Indicates the maximum value among all parts in the displayer (read-only).	
Auto Range	Determines whether to set automatically the range of values to display.	
	Off	Does not use auto ranging, rather uses the values that are set in *Min* and *Max*.
	Min and Max Values	Automatically determines the range from the field function data.
	Min Value	Automatically determines the minimum value from the field function data. The property switches automatically from `Min and Max Values` to this setting if you make an entry for *Max Value*.
	Max Value	Automatically determines the maximum value from the field function data. The property switches automatically from `Min and Max Values` to this setting if you make an entry for *Min Value*.
Auto Range Mode	Chooses the domain for auto-range minimum and maximum value computation that is used for color coding.	
	Displayer Parts	Parts added to the particular displayer.

	Displayer Regions	Regions from which the displayer extracts data.
	All Regions	All regions in the simulation.
Clip	Determines how to display values outside of the range that is specified by *Min* and *Max*. See Changing the Scalar Range.	
	Off	Values above *Max* are displayed in the maximum color, and values below *Min* are displayed in the minimum color.
	Below Min, Above Max	Does not display values outside the range.
	Below Min	Does not display values below *Min*.
	Above Max	Does not display values above *Max*.
Units	Selects the type of units to display. See Changing the Scalar Units.	
Min	Sets the minimum scalar value to include in the displayed range.	
Max	Sets the maximum scalar value to include in the displayed range.	
Map Values	Activates or deactivates the setting of a displayed range with the *Min* and *Max* properties.	
	Activated	Mapping of values is active.
	Deactivated	Mapping of values is not active.
Map Size Minimum	The physical size of the particle that represents the Min scalar value. The size is relative to the overall dimension of the mesh.	
Map Size Maximum	The physical size of the particle that represents the Max scalar value. The size is relative to the overall dimension of the mesh. Some graphics hardware may impose a limit on how large this size can be.	

Copying Properties of a Scalar Field

It is possible to copy certain properties of one scalar field object onto another.

When properties are copied from one scalar field to another, the following properties are changed:

- *Auto Range*
- *Clip*
- *Function*

However, since the other properties are affected by which parts are displayed in the scene, you must copy the displayer properties to make the destination scalar field match the source one completely.

Defining a Custom Scalar

In addition to the standard system field functions, you can also display custom user-defined field functions.

Once you create the desired field function, simply select it from the list of available field functions as before.

Scalar Displayer Properties

Use the properties of the scalar displayer to control the contour style, point scale, and smoothness of the contours.

Contour Style	Controls how the values, represented by colors, are separated from each other. See Changing the Contour Style.	
	Automatic	Automatically selects the most appropriate contour style, depending on the specified scalar function and parts (default).
	Filled	Assigns values based on cell values. The values that are shown are the values stored at the cell centroids on the server. Scenes have a stepped appearance.
	Smooth Filled	Assigns values by node. See Understanding Smoothed Values.
	Smooth Filled + Lines	Adds lines to separate the colored values in the `Smooth Filled` style. Smoothing is applied to get the node values. See Understanding Smoothed Values.
	Line	Draws lines between the different values.
	Smooth Blended	Displays a smooth transition from one value to another. Smoothing is applied to get the node values.
Display Mesh	Controls the appearance of the mesh grid. See Displaying Mesh Outlines and Surfaces.	
	Activated	Shows the complete grid and adds the **Mesh Color** subnode to the displayer node to control the color of the mesh lines. This lets you examine the density and cell structure of various parts of the model, and thereby make judgments about accuracy. See

		Changing the Color of the Mesh Lines Using the Displayer .
	Deactivated	Hides the grid except for the outlines. When you choose this option and activate the *Surface* property, you generate a more realistic image of the model.
Opacity	Adjusts the transparency of the part. The settings range from 0.0 (invisible) to 1.0 (opaque), and can be controlled with a slider. This property allows you to show an overlay of different parts of the model. See Making Displayer Surfaces Transparent.	

Scalar Displayer Expert Properties

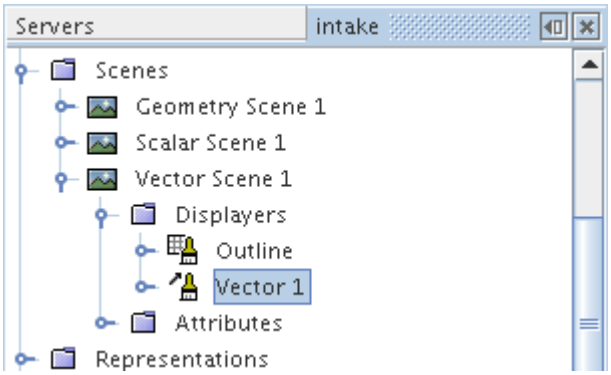
Representation	Selects a representation. Additional selections would appear as representations are added to the simulation.	
Point Scale Mode	Selects a mode for scaling points. See Working with Point Scale Modes.	
	Pixels	Activates the **PointSize** node to let you set the display size of points on the screen in pixels.
	Scalar	Activates the **Point Size Scale Field** node to let you scale points based on a scalar function.
	Particle Model	Scales points based on data in the particle phase. Activate when you are visualizing Lagrangian or DEM particles.
Line Width	Adjusts the width in pixels of lines that are used in the display, depending on the type of displayer. For example, in a scalar displayer, this setting would be effective only if the *Contour Style* property were set to `Line`. This property can be set with a slider.	
Specular Highlight Mode	Specular highlighting is the reflection of a light source on a shiny surface (such as polished metal).	
	Material	Uses the color of the material (default). For example, on a black surface, no specular highlighting is visible.
	Light	Uses the color of the light. To specify this color, see Controlling the Color of Lighting Objects.

Lighting	Controls the lighting of the part. Accessing the property customizer opens the *Lighting* dialog. See Lighting Displayer Surfaces.	
Data Focus	Selects a data focus object to apply to this displayer. The choice of data focus objects depends on the total number of these objects in the simulation. When a data focus object is selected, a data focus sub-node is added to this node. See Exploring Solutions with Data Focus.	
Smooth Shade	Controls the transition between segments of the image.	
	Activated	Makes the transitions smooth for a more realistic effect.
	Deactivated	Leaves this function off.
Ignore Clip	Controls whether the parts in the displayer are affected by any enabled clip plane. See Using Clip Planes.	
	Activated	Shows the parts unaffected by the clip plane.
	Deactivated	Shows the parts that are trimmed by the clip plane.
Face Culling	Controls whether and how to cull polygons. See Culling Polygons.	
	None	Does not cull any polygons.
	Front	Culls (omits from the display) all polygons that are oriented toward the front of the view.
	Back	Culls (omits from the display) all polygons that are oriented toward the back of the view.
Transform	Selects a transform to apply to all parts in the displayer. **None** is the default selection. Additional selections would appear when you create transforms. See Creating a New Transform. This feature allows for the creation of symmetric and periodic repeats, as well as arbitrary transforms to be used.	

Displaying Vector Quantities

Vector displayers are used to display glyphs that represent vector field functions.

The **Vector** displayer node, which has its own properties and shares a pop-up menu with other types of displayers, controls the display of vectors along the surface of a part.



In the icon, the arrow next to the paint brush signifies the depiction of vectors. The vector displayer also includes an undefined vector field node.

You can use this displayer to display the direction of quantities on boundaries without running the simulation.

As an alternative to glyphs, this displayer can visualize vector fields with a texture-based technique called line integral convolution.

Contents:

Creating Vector Displayers
Changing the Vector Quantity
Displaying the Direction of Quantities on Boundaries
Working with Line Integral Convolution
Customizing Glyphs
Changing the Size of Glyphs
Controlling the Spacing of Glyphs
Changing the Style of the Glyphs
Controlling the Scale of the Glyph Heads
Controlling the Projection of the Glyphs
Working with the Vector Field Object
Copying Properties of a Vector Field
Defining a Custom Vector
Using the Color Bar to Modify the Vector Display
Coloring Glyphs Based on Scalar Values
Vector Displayer Properties

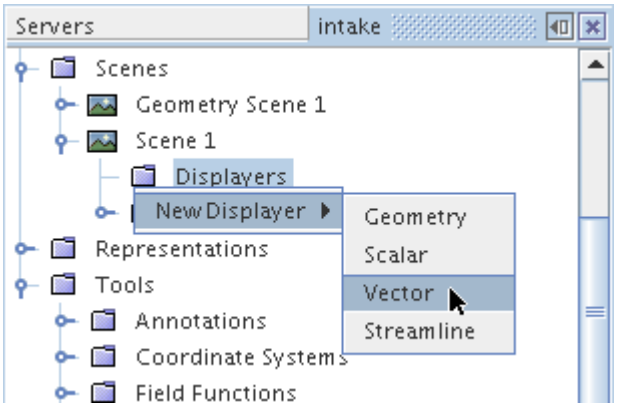
See Also:

Using the Scalar and Vector Bars

Creating Vector Displayers

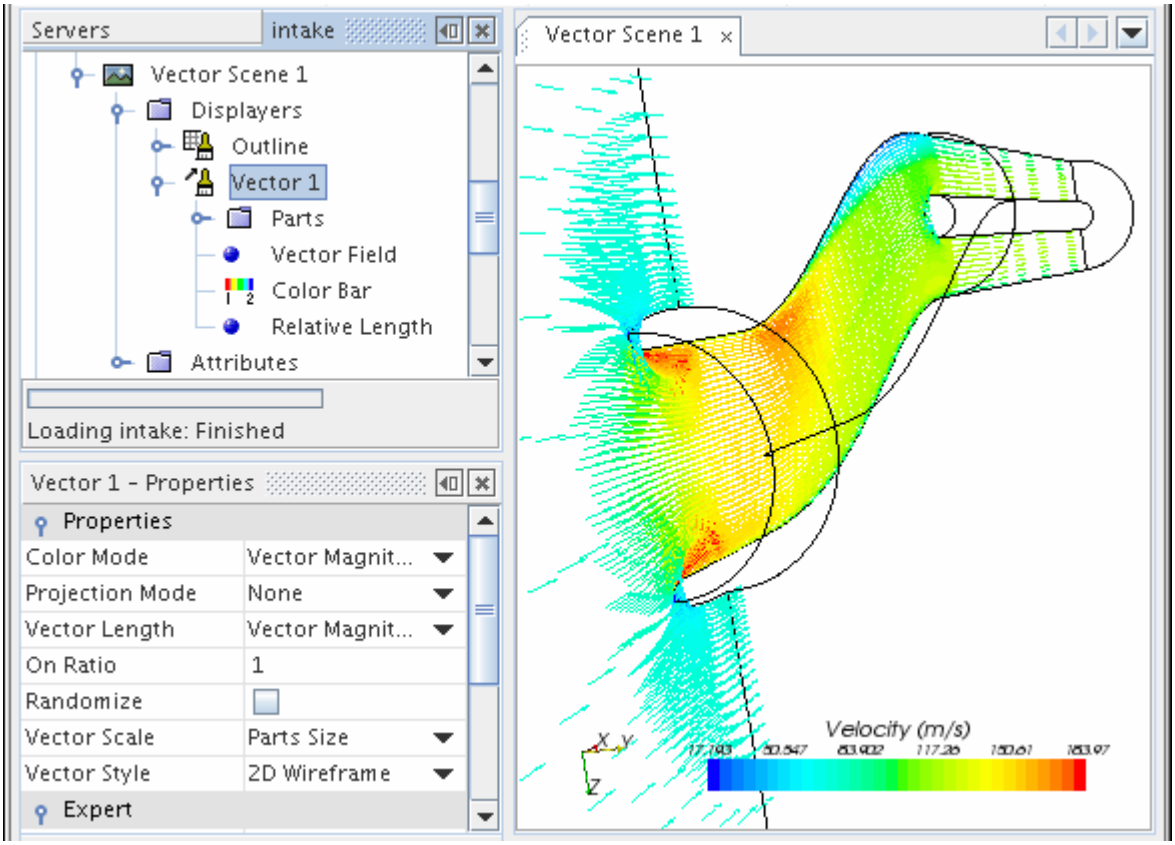
Vector displayers are created automatically when a vector scene is created. Alternatively you may wish to add a vector displayer to any scene.

To add a vector displayer manually, open the **Scenes** node and the node of the chosen scene, right-click on the **Displayers** node and select **New Displayer** > **Vector**.



By default the vector displayer will contain no parts after creation and you must add parts to the displayer. This feature allows you to choose only those parts of the model on which you wish to view vectors.

Also by default a vector displayer uses velocity as the vector quantity. Glyphs are plotted on the parts' surfaces to represent the direction and the magnitude of the vector quantity at each boundary face centroid. The glyphs are also colored according to their magnitude.

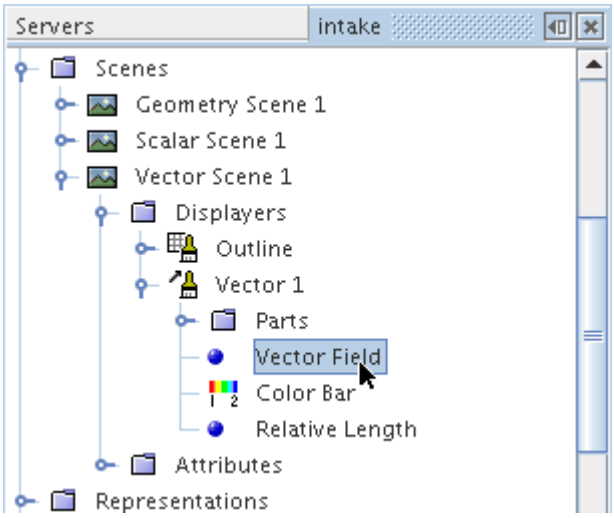


The default name of the displayer is based on its type. However, renaming the displayer is possible.

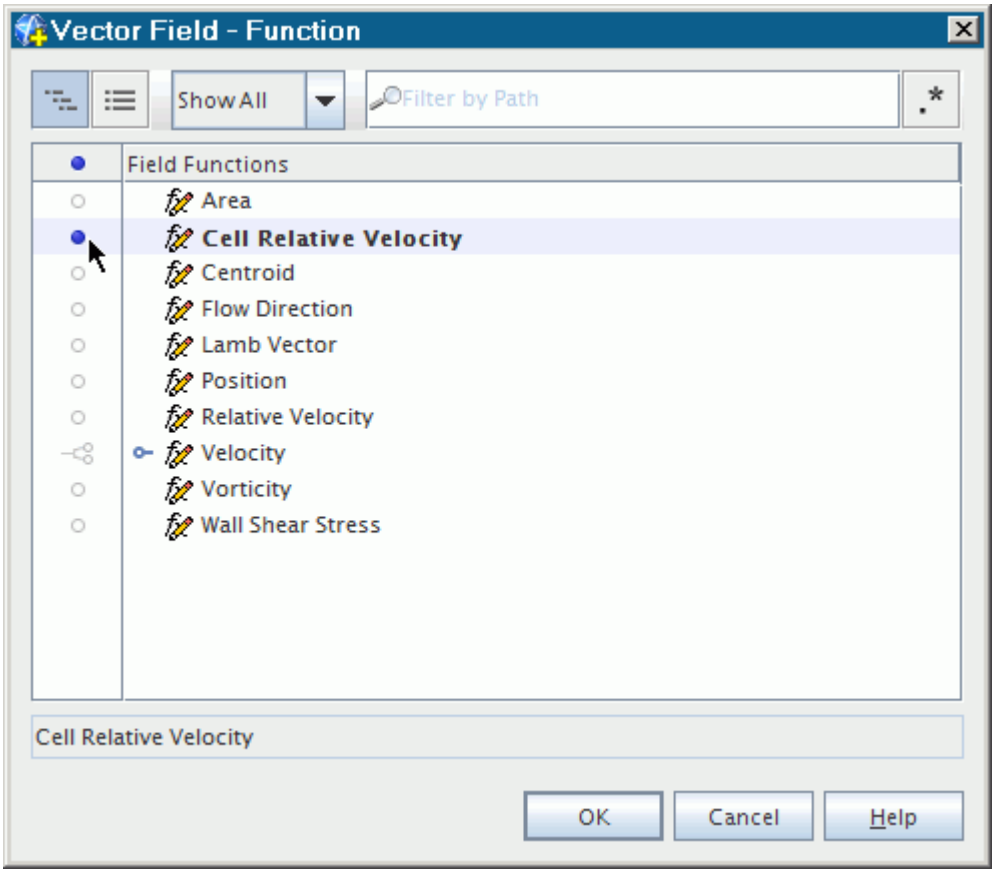
Changing the Vector Quantity

By default the vector displayer uses the Velocity field function; however any vector field function can be used in a vector displayer.

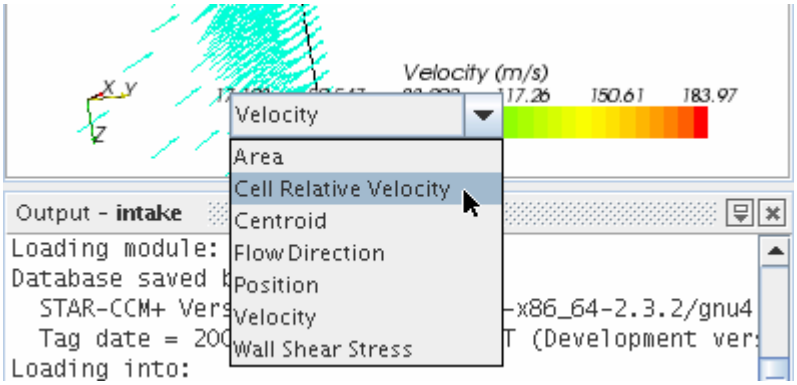
Expand the **Scenes** node, and select any scene which includes a vector displayer (any default vector scene). Expand the **Displayers** and vector displayer nodes, and select the **Vector Field** node.



Choose the vector field function from the list in the object selection dialog. To activate the dialog, edit the *Function* property.



Alternatively you can change the vector field function from the visualization display. Right-click on the vector bar and select a vector field function from the same drop-down menu.

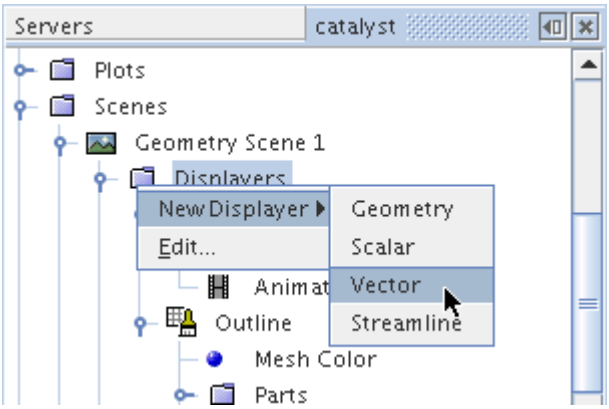


Displaying the Direction of Quantities on Boundaries

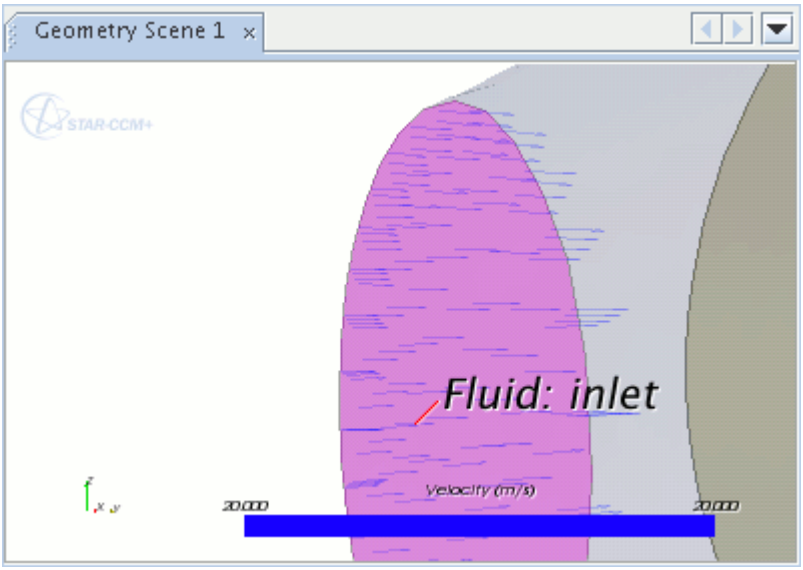
There is a simple way to show the direction of a quantity on a boundary even before you get a solution.

For example, the following steps show how to display the direction of velocity on an inlet in a geometry scene:

- Initialize the solution by clicking the (**Initialize Solution**) button in the toolbar or by using the **Solution** > **Initialize Solution** menu item.
- Create a vector displayer in the scene.



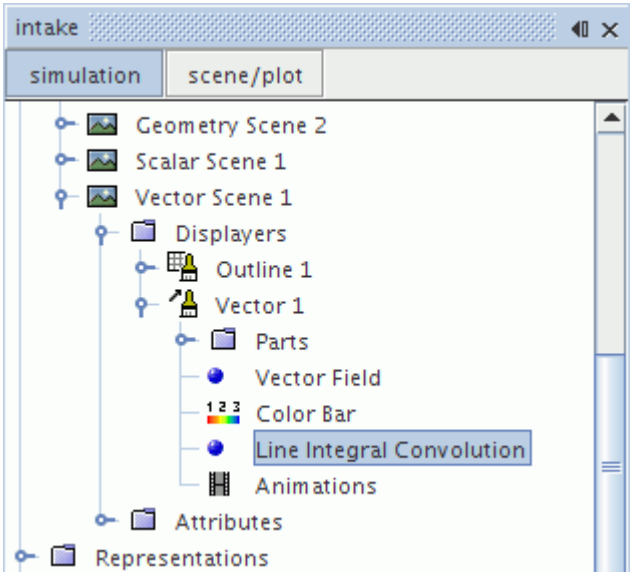
- Add the inlet through the drag-and-drop technique or using the Parts Selector from the properties of the new vector displayer. The velocity vectors appear in the scene. (A convenient way to view them completely is using the (**Make Scene Transparent**) toolbar button, as shown in the following screenshot.)



Working with Line Integral Convolution

Line integral convolution (LIC) is a texture-based technique for visualizing vector fields.

The **Line Integral Convolution** node represents the activation of this option in the STAR-CCM+ vector displayer. Select this node to access its properties.

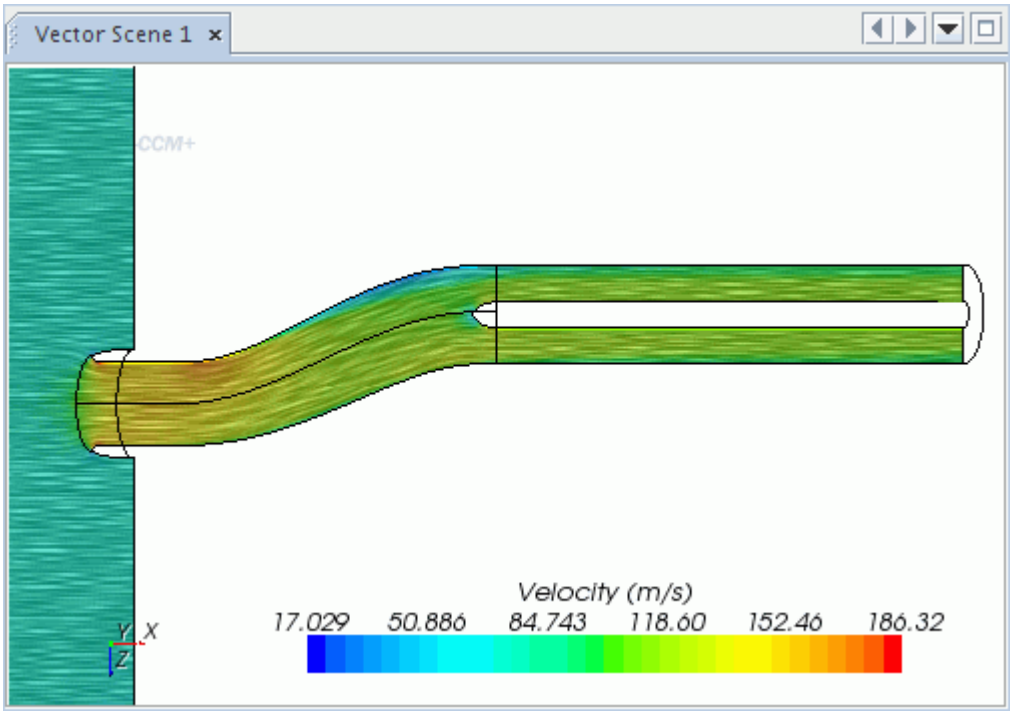


Note: On Linux, line integral convolution is not supported on Intel graphics chipsets.

To activate LIC:

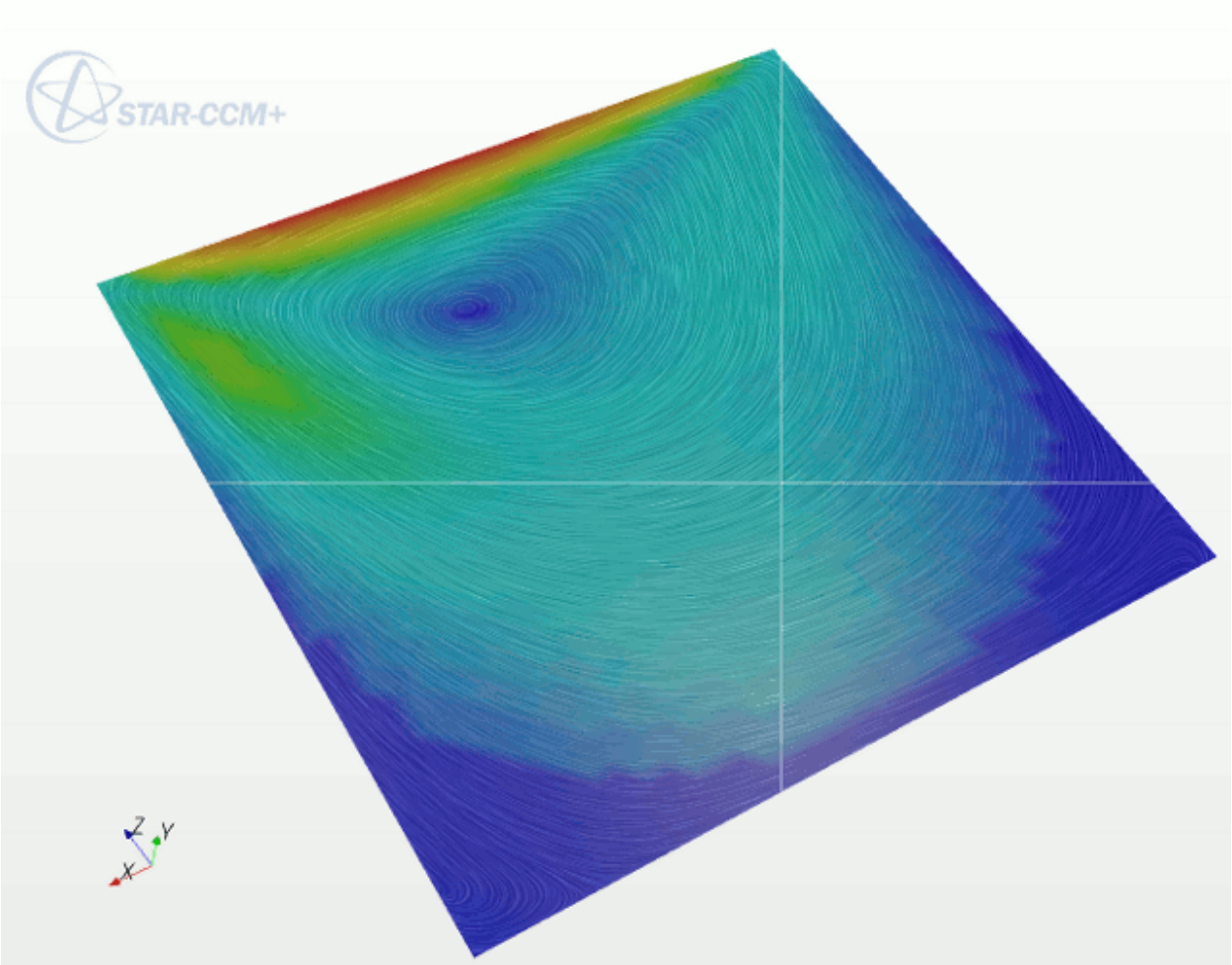
- Select `Line Integral Convolution` for the Display Mode property of a vector displayer node.

Cabral and Leedom first introduced LIC for imaging two-dimensional vector fields. Generally this technique involves convoluting a white noise image along streamlines computed from the vector field. In the resulting image, these streamlines cover the entire domain of the vector field. This technique has the advantage of being able to visualize large and detailed vector fields in a reasonable display area. Compared with simpler integration-type approaches, which entail following the flow vector at each point to produce a line, LIC produces a whole image at every step.



Limitations

- For the most accurate display of the direction of vectors, set the projection mode of the scene to parallel. See Changing the Projection Mode.
- This feature is designed to use the Graphics Processing Unit (GPU) of your graphics hardware. It can work slowly in conjunction with software (mesa) rendering.
- For hardcopy with a magnification factor greater than 1, seams appear. These seams come from an attempt by STAR-CCM+ to perform tiled rendering, with results differing at the boundaries of adjacent tiles.



Line Integral Convolution Properties

Enhanced	Controls whether to use enhanced LIC computation.	
	Activated	Activates enhanced LIC computation. This option produces better-looking results, but is computationally more expensive.
	Deactivated	Leaves this feature deactivated.
Number Steps	Number of integration steps to perform during the LIC image generation. This property can be set with a slider. See Using a Slider.	
Blending Factor	The contribution of the LIC image when blending with the surface color, using linear interpolation between the LIC image color and the surface color. Ranges are from 0 to 1. 0 for surface color only, and 1 for LIC only (grayscale image). This property can be set with a slider.	
Step Size	The length of the integration step (in pixels). This property can be set with a slider.	

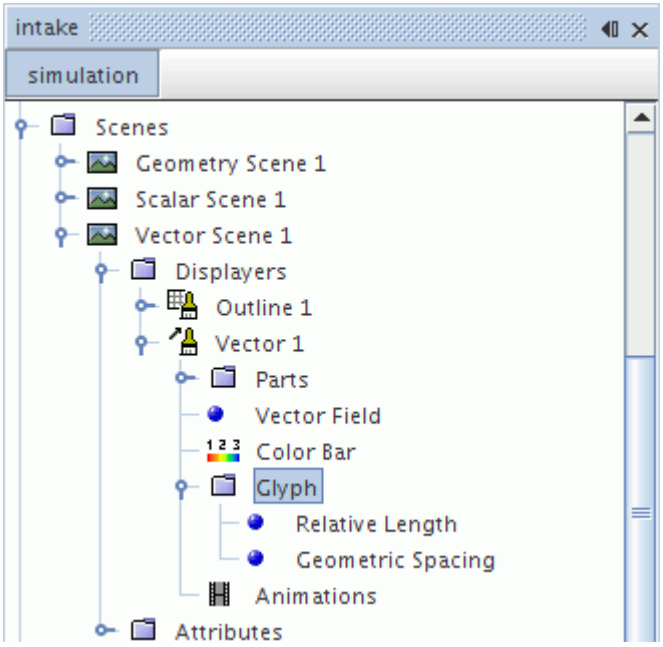
Line Integral Convolution Bibliography

[189] *Cabral, B., Leedom, L.C., 1993. "Imaging Vector Fields Using Line Integral Convolution", Proceedings of ACM SigGraph 93, August 2-6, Anaheim, California, pp. 263-270.*

Customizing Glyphs

The **Glyph** manager node allows you to customize the size and spacing of the glyphs.

Open the node to access its sub-nodes. Select the node to access its properties.



Properties of the Glyph Manager

Vector Spacing	Defines how the vectors are distributed. See Controlling the Spacing of Glyphs.	
	Geometric	Vectors are placed on cell faces, and the density of the vectors can be controlled by changing the *On Ratio* and *Randomize* properties.
	Uniform Screen	Vectors are placed at each grid point on a uniform grid that lies in the plane of the screen. The density of the vectors can be controlled by changing the grid dimensions with the *N Grid Points* property.
Vector Scale	Defines how to scale the vectors. See Changing the Size of Glyphs.	

	Absolute	Scales the vectors based on a specified length in the *Glyph Length* property of the **Absolute Length** node.
	Parts Size	Scales the vectors relative to the size of the parts in the *Glyph Length* property of the **Relative Length** node.
	Screen Size	Scales the vectors relative to the display size in the *Glyph Length* property of the **Relative Length** node.
	Direct	Scales the vectors based on a proportion specified in the *Glyph Length* property of the **Direct Length** node.
Vector Length	Quantity that determines vector length.	
	Vector Magnitude	Uses relative vector magnitude, that is, the scales of all vectors are fractions of the largest one.
	Constant Length	Lets you scale vectors on the order of the model rather than on the data.
Vector Style	Defines how the vector are displayed. See Changing the Style of the Glyphs.	
	2D Wireframe	A simple line.
	2D Filled	A line with finite diameter.
	3D Head, 2D Tail	Conical head.
	3D Head, 3D Tail	Conical head and tail.
	Cone	Cone only.
	Sphere	Sphere only.

Expert Properties of the Glyph Manager

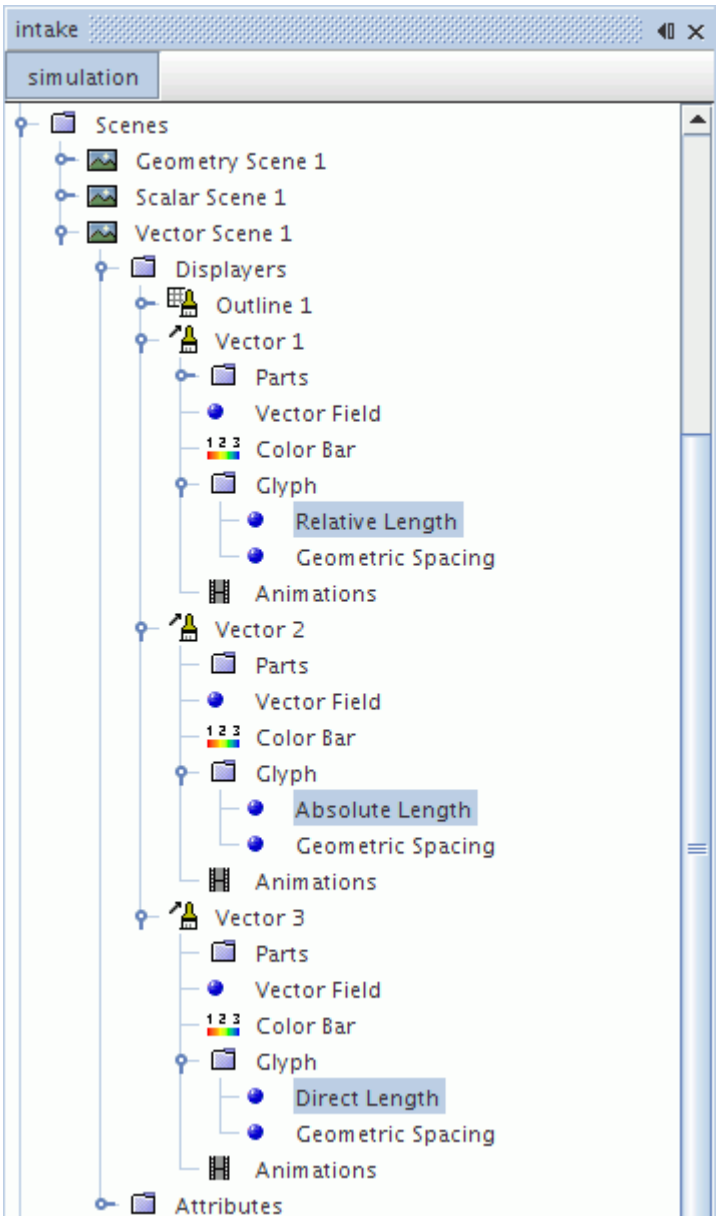
Glyph Tip Scale	Sets the scale of the glyph heads. See Controlling the Scale of the Glyph Heads.	
Constant Tip Size	Controls whether to keep the size of the glyph heads constant.	
	Activated	Causes the head of the arrow in the vector displayer to be scaled equally for all arrows independent of the underlying velocity function.

	Deactivated	Leaves this function off.

Changing the Size of Glyphs

The glyph manager provides several features with which you can control the size of the glyphs.

These features include the **Relative Length**, **Absolute Length** and **Direct Length** objects, whose nodes have properties.



The **Relative Length** node appears when the *Vector Scale* property of the glyph manager is set to `Parts Size` or `Screen Size`, the **Absolute Length** node appears when this property is set to `Absolute`, and the **Direct Length** node appears with a setting of `Direct`.

In addition to the *Vector Scale* property of the glyph manager, its *Vector Length* property provides more flexibility. The relationships among these properties and the relative or absolute length of the vector glyphs are outlined in the table below.

		Vector Length	
		`Vector Magnitude`	`Constant Length`
Vector Scale	`Absolute`	• Glyph size remains the same independent of the parts or the display. • *Glyph Length* is a value; applies to size of largest glyph.	• Glyph size remains the same independent of the parts or the display. • *Glyph Length* is a value; applies to size of all glyphs.
	`Parts Size`	• Glyph size changes with parts size. • *Glyph Length* is a percentage; applies to size of largest glyph.	• Glyph size changes with parts size. • *Glyph Length* is a percentage; applies to size of all glyphs.
	`Screen Size`	• Glyph size remains the same while parts size changes. • *Glyph Length* is a percentage; applies to size of largest glyph.	• Glyph size remains the same while parts size changes. • *Glyph Length* is a percentage; applies to size of all glyphs.
	`Direct`	• Glyph size represents the vector quantity, based on the scale factor entered in the *Glyph Length* property. • *Glyph Length* is the scale factor of the unit chosen.	• Glyph size remains the same. • *Glyph Length* is a value; applies to size of all glyphs.

By default the glyphs are scaled using the magnitude of the vector. To change the scaling mode, expand the **Scenes** node and expand a scene which contains a vector displayer (all default vector scenes). Expand the **Displayers** node and the vector displayer node, select the glyph manager node and choose settings for the *Vector Length* and *Vector Scale* properties.

Although the **Relative Length**, **Absolute Length** and **Direct Length** nodes seem to have the same *Glyph Length* property, entries are retained separately for each combination of settings that are listed in the table above. For example, if your entry for *Glyph Length* with an `Absolute` vector scale causes the glyph to

appear visibly longer than it did with a `Parts Size` (relative) vector scale, and then you switch back to `Parts Size`, you see the glyphs revert to their earlier size.

The Direct Length feature allows you to define a vector scale as a function of velocity. In this manner you can create animations where the vector length is consistent between frames, and also compare similar simulations with different maximum velocities.

Relative Length Properties

Glyph Length (%)	Specifies the length of the largest glyph or all of them as a percentage of parts size or screen size. (See preceding table.)

Properties of Absolute Length and Direct Length

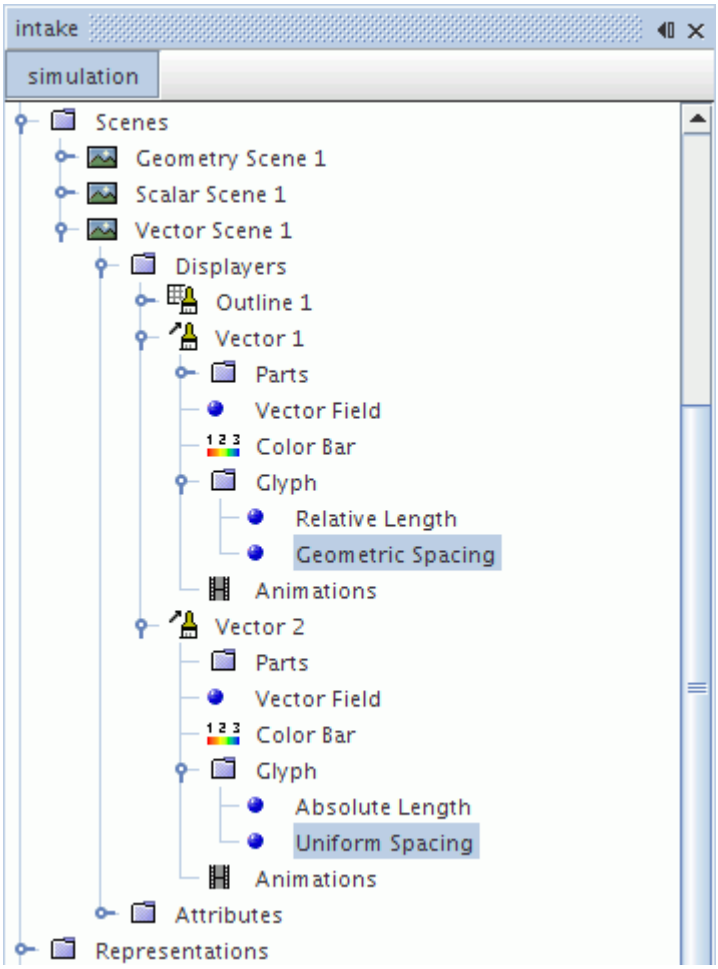
Glyph Length	Specifies the length of the largest glyph or all of them as a specific value or a proportion of a unit. (See preceding table.)
Units	Selects the type of units to use.

Controlling the Spacing of Glyphs

The spacing and density of glyphs can be specified in two ways: geometric spacing and uniform spacing.

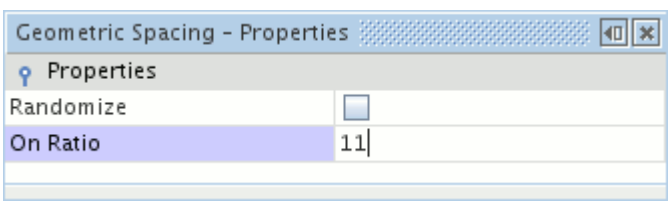
Geometric spacing places a vector glyph at the vertex or at the centroid of each cell face in the mesh. This type of spacing is well suited to cases where there is little variation in mesh cell size. Uniform spacing places a vector glyph at each intersection in a user-specified uniform grid that lies in the plane of the screen, and is particularly useful when the mesh has a large variation in cell size.

To access these options select a scene containing vectors, expand the **Displayers** node and the vector displayer node, and select the **Glyph** node. In the *Properties* window, select the *Vector Spacing* property and use the drop-down list to select either **Geometric** for geometric spacing, or **Uniform Screen** for uniform spacing. Depending on the selection either a **Geometric Spacing** node or a **Uniform Spacing** node is added to the tree, and these nodes have properties and are subject to common menu actions.

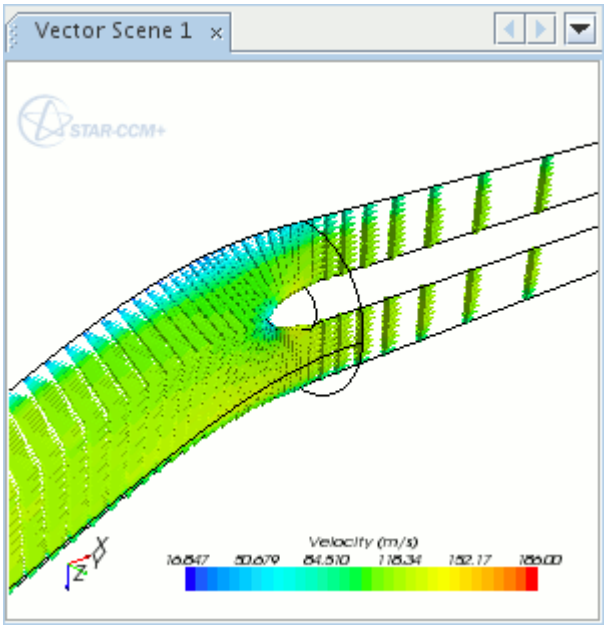
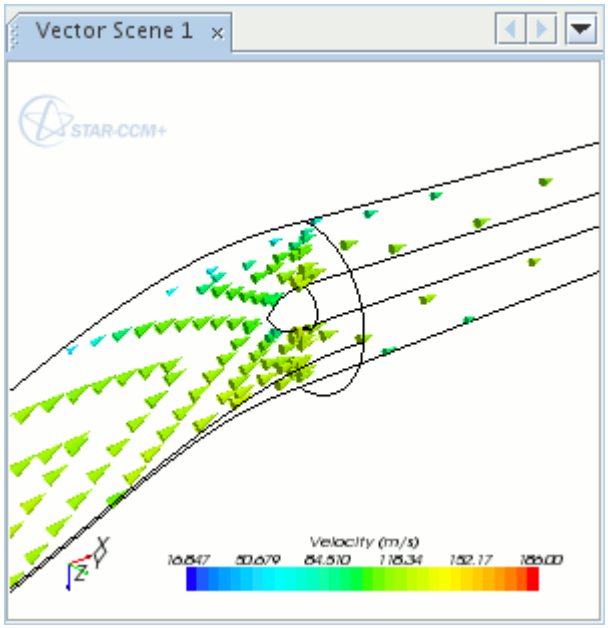


Geometric Spacing

When the **Geometric Spacing** node is selected, the density of glyphs can be controlled by increasing the *On Ratio* property (using integers only).



The number of glyphs that are shown are decreased as the integer value under the *On Ratio* property is increased. (In the following screenshots the vector style has been changed to **Cone** to show the effects more clearly.)

	
On Ratio = 1 (default number)	On Ratio = 11

The choice of glyphs to display can be randomized by checking the box for the *Randomize* property.

By default, the location of glyphs is the same as their native data location, but you can reposition the glyphs with the *Glyph Location* property. The native data location can be the cell face centroid or the vertex, depending on the type of field function and the part that is assigned to the displayer. For example, Position is a natively vertex-based field function, while Vorticity is cell-centroid based.

Vector glyphs are cell-centroid based by default under the following conditions:

- They appear in simulation files that were saved in versions of STAR-CCM+ prior to 9.06.
- Their native data location cannot be determined, due to either of the following causes:
  - The input parts of a derived part exhibit conflicting native data locations.
  - The internal field function definition is ambiguous or undefined.

Note: Certain parts only allow vertex-based vector glyphs: feature lines, streamlines, and isosurfaces.

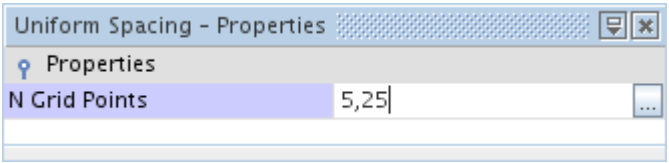
Geometric Spacing Properties

Randomize	Determines whether the pattern of vector glyphs is defined by the mesh grid.	
	Activated	Ignores the structure of the model, including differences in density of the grid. This can result in a better distribution of vectors if the mesh is highly unstructured or has great variation in cell surface area.

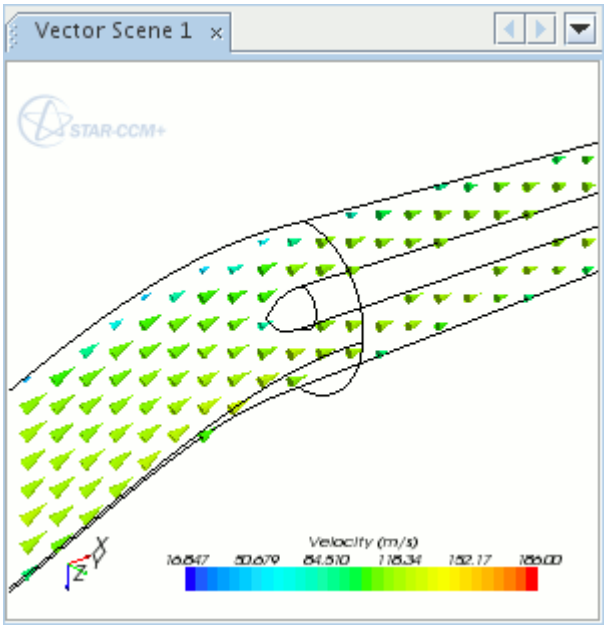
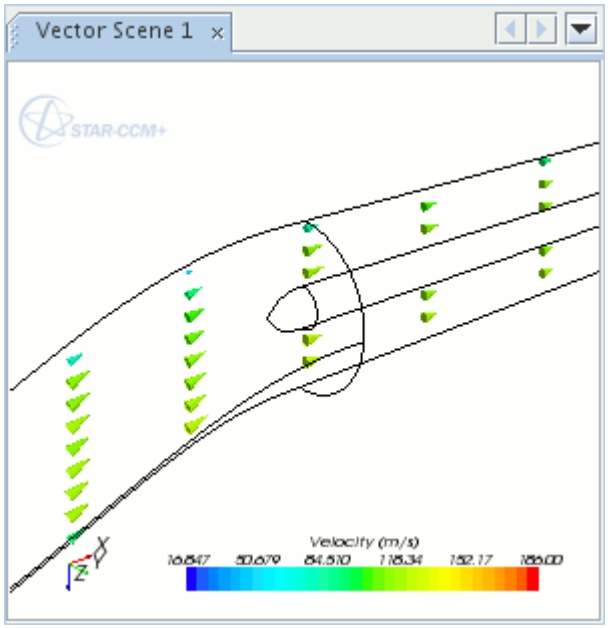
	Deactivated	Takes the structure of the model into account.
On Ratio	The number of cells that display a single vector quantity. You can use this function to control the density of vectors, and thereby study broader or narrower patterns. At the default setting of 1, every cell displays a vector. Higher settings cause fewer vectors to be displayed. The value must be an integer.	
Glyph Location	Sets the location of the vector glyphs.	
	Cell	Sets the glyph location at the cell face centroid.
	Vertex	Sets the glyph location at the cell vertex.
	Native	Automatically uses the native data location, which depends on the assigned field function and the displayer part.

Uniform Spacing

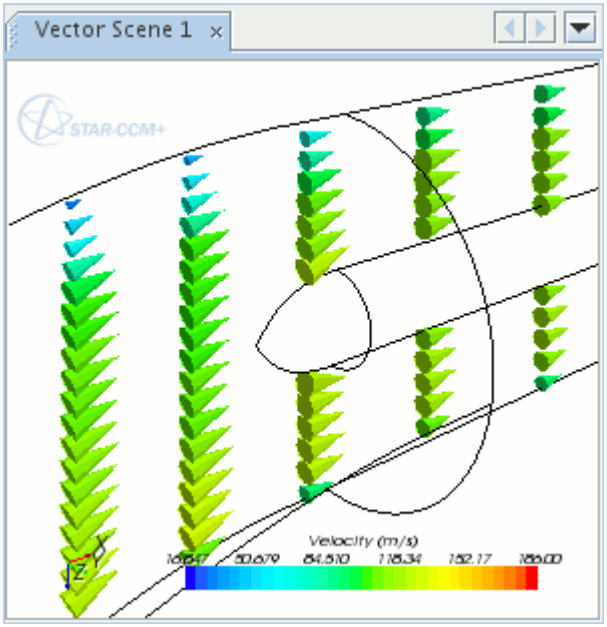
When the **Uniform Spacing** node is selected, the density of glyphs can be controlled by setting the dimensions of the grid. The grid is bound by the extents of the scene display, so the values entered for the *N Grid Points* property control how many grid points are placed horizontally and vertically in the scene, or `u` and `v`, respectively.



A vector glyph is placed at each grid point on the parts in the display.

	
N Grid Points = [20, 20] (default)	N Grid Points = [5, 25]

Zooming in does not affect the grid size; in the following screenshot the *N Grid Points* property is still set to `[5, 25]` and the zoom level has been increased.



The uniform grid is linked to the size of the scene display, so shrinking the window increases the glyph density, and enlarging the window decreases the glyph density.

Uniform Spacing Properties

N Grid Points	Specifies the u and v values of the grid, where u is the number of grid points in the horizontal direction and v is the number of grid points in the vertical direction.

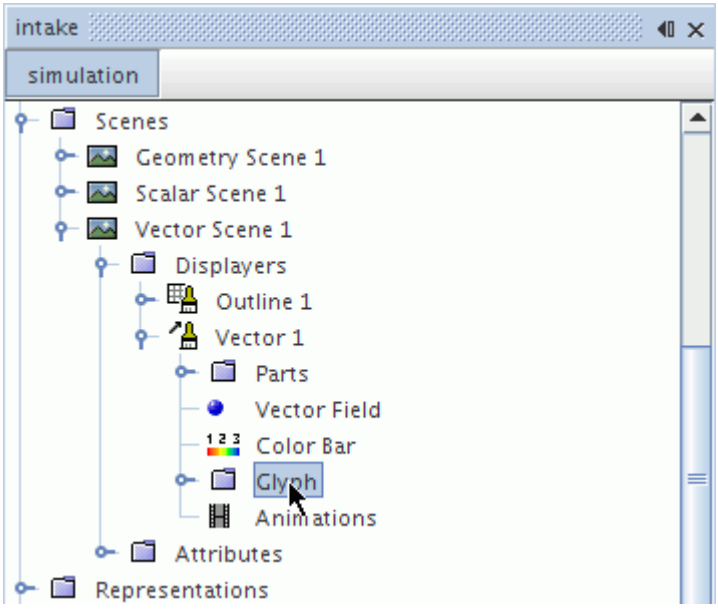
Changing the Style of the Glyphs

By default the glyph style is a two-dimensional wire-frame arrow. Five alternative styles are available.

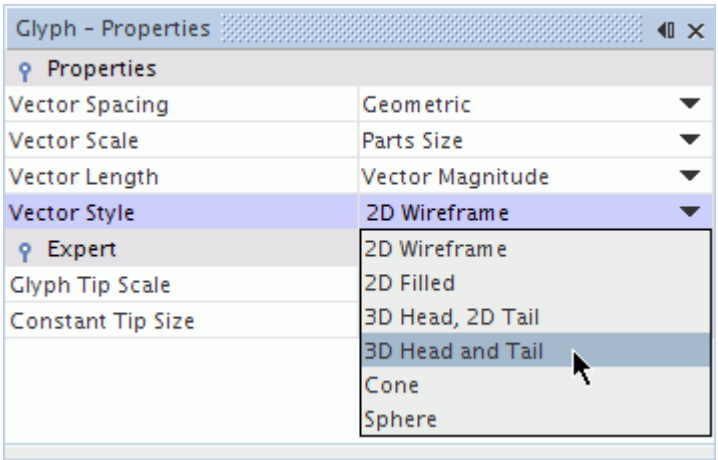
The styles are:

- Two-dimensional with a filled arrow head
- Three-dimensional arrow head with two-dimensional tail
- Three-dimensional arrow head with three-dimensional tail
- Cone
- Sphere

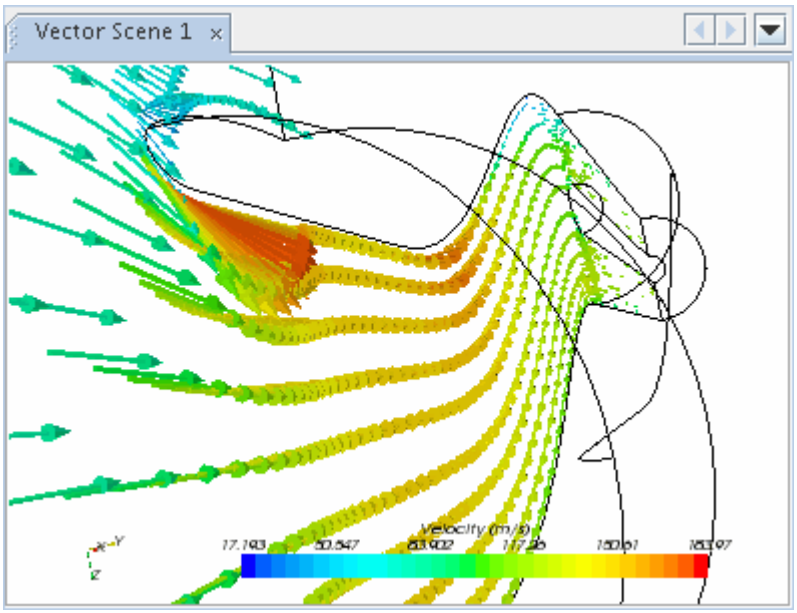
Expand the **Scenes** node and expand a scene node that includes a vector displayer. Expand the **Displayers** node and the vector displayer node, and select the **Glyph** node.



Select the glyph style from the drop-down menu under the *Vector Style* property in the *Properties* window.



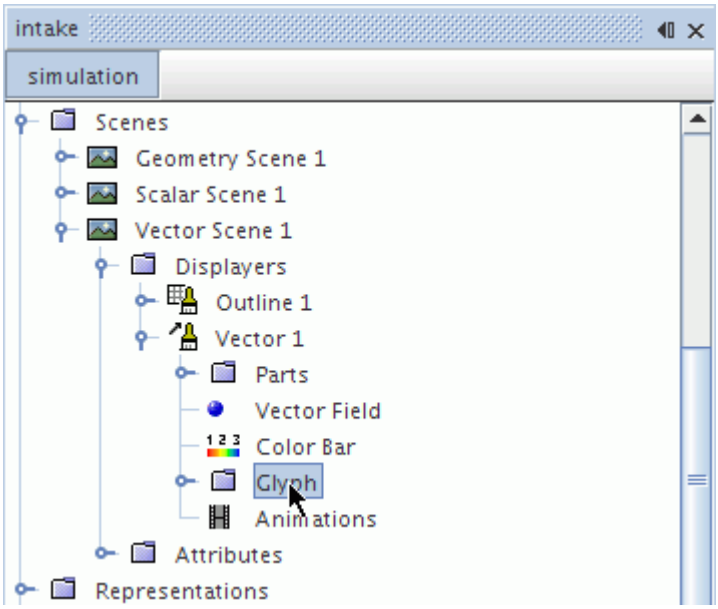
This changes the appearance of the glyphs in any open vector scene display.



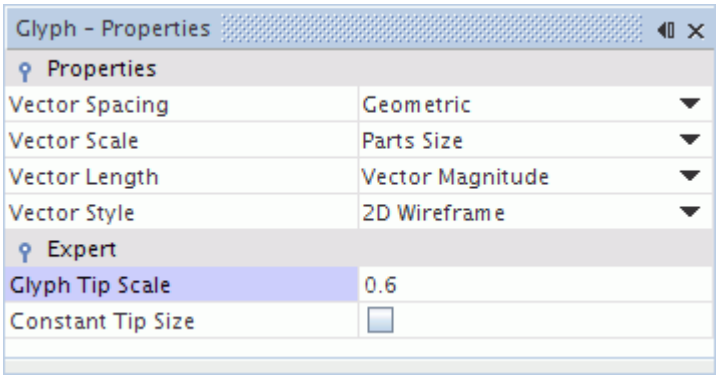
Controlling the Scale of the Glyph Heads

It is possible to control the scale of the head of the vector glyph independent of the tail.

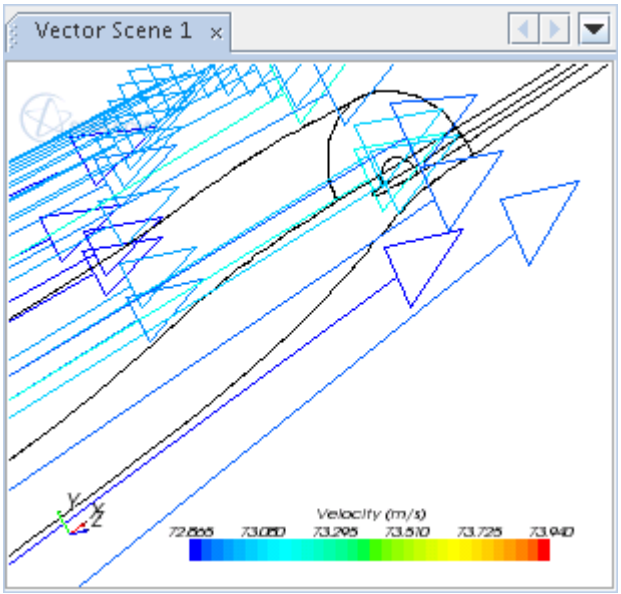
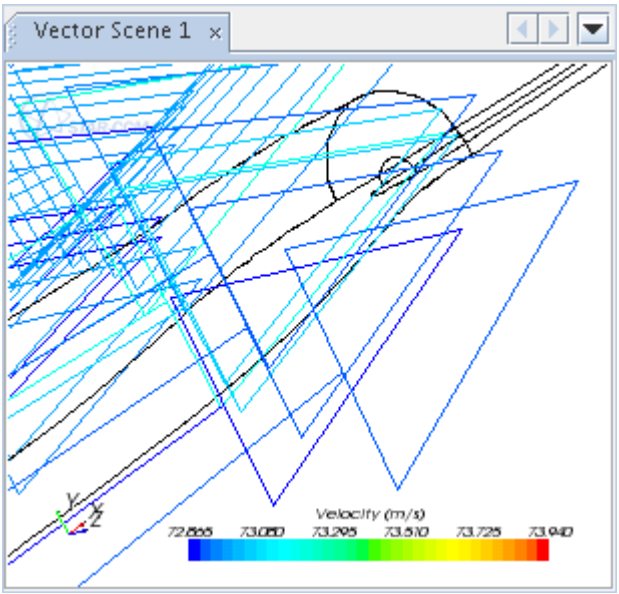
Expand the **Scenes** node and expand a scene node that includes a vector displayer. Expand the **Displayers** node and the vector displayer node, and select the **Glyph** node.



In the *Glyph Tip Scale* expert property, enter a value between `0.0` and `1.0`.



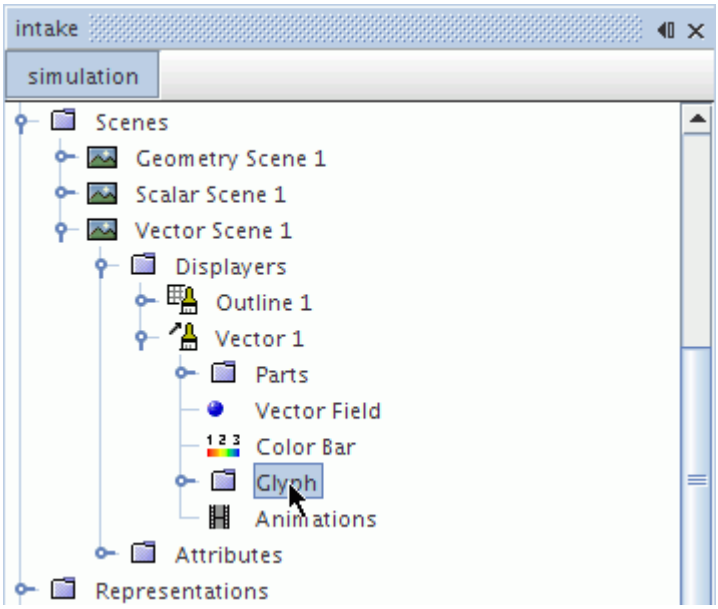
This changes the scale of the glyph heads with no change to the appearance of the tail.

	
0.2 (default number)	0.6

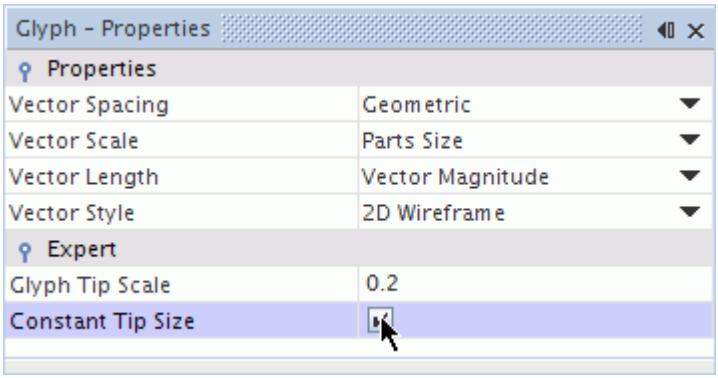
You can also keep the size of the glyph heads constant.

Keeping the Glyph Head Size Constant

The head of the arrow in the vector displayer can be scaled equally for all arrows independent of the underlying velocity function. Expand the **Scenes** node and expand a scene node that includes a vector displayer. Expand the **Displayers** node and the vector displayer node, and select the **Glyph** node.



In the *Properties* window, activate the checkbox of the *Constant Tip Size* expert property.



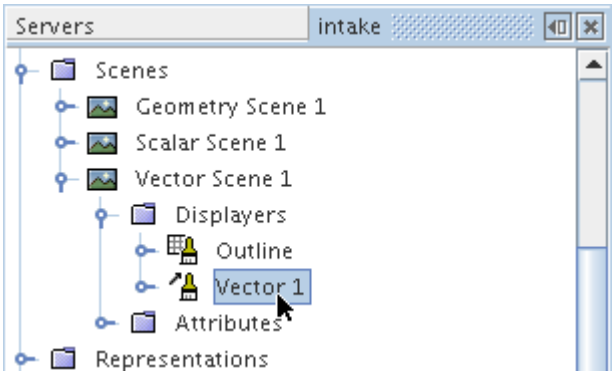
This feature works as follows:

- The head scaling is controlled by the *Glyph Tip Scale* expert property only.
- The head size is never bigger than the actual arrow size; therefore the arrows still disappear where there is no flow.
- If the *Vector Style* property is set to `3D Head and Tail`, the tail radius can be scaled by changing the *Line Width* expert property. The *Line Width* expert property can be set with a slider.
- If the *Vector Style* property is set to `Cone` or `Sphere`, then the glyphs are unaffected by the *Glyph Tip Scale* or *Constant Tip Size* options, since the velocity function determines the scaling.
- If the *Vector Length* property is set to `Constant Length`, then the *Constant Tip Size* option has no effect on the display.

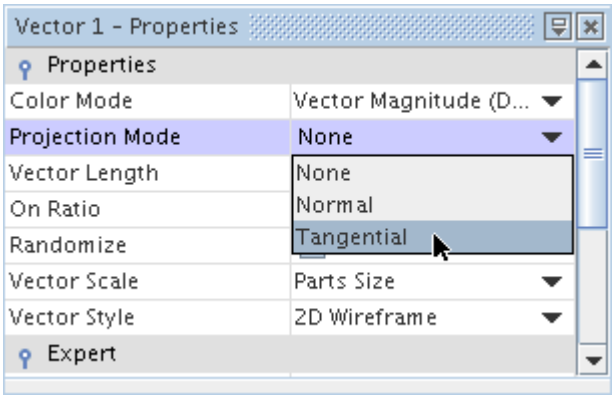
Controlling the Projection of the Glyphs

You can project the vector glyphs using only the normal component or the tangential component.

Expand the **Scenes** node and expand a scene node that includes a vector displayer. Expand the **Displayers** node and select the vector displayer node.



Select an option in the drop-down list of the *Projection Mode* property in the *Properties* window.



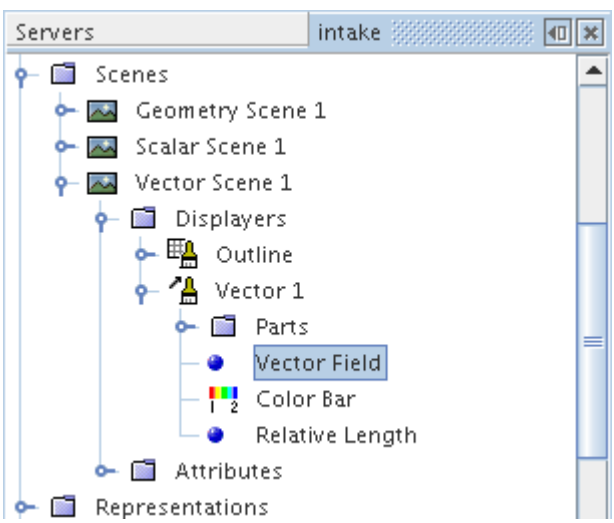
The options work as follows:

- `None`, the default, leaves the vectors with both normal and tangential components.
- `Normal` removes the tangential component of the vectors, projecting them with the normal component.
- `Tangential` removes the normal component of the vectors, projecting them with the tangential component.

Working with the Vector Field Object

The **Vector Field** node controls data in a scene with a vector displayer; by default it is set to velocity.

Use the properties of this node to control what is displayed and the range of values.



Vector Field Properties

Function	Selects the type of vector field function to show, such as velocity or flow direction. See Field Functions.
Auto Min	Indicates the minimum value among all parts in the displayer (read-only).

Auto Max	Indicates the maximum value among all parts in the displayer (read-only).	
Auto Range	Determines whether to set the range of values to display automatically. See Changing the Scalar Range.	
	Off	Does not use auto ranging, rather uses the values that are set in *Min* and *Max*.
	Min and Max Values	Automatically determines the range from the field function data.
	Min Value	Automatically determines the minimum value from the field function data. The property switches automatically from `Min and Max Values` to this setting if you make an entry for *Max Value*.
	Max Value	Automatically determines the minimum value from the field function data. The property switches automatically from `Min and Max Values` to this setting if you make an entry for *Min Value*.
Auto Range Mode	Chooses the domain for auto-range minimum and maximum value computation that are used for color coding.	
	Displayer Parts	Parts added to the particular displayer.
	Displayer Regions	Regions from which the displayer extracts data.
	All Regions	All regions in the simulation.
Clip	Determines how to display values outside of the range that is specified by *Min* and *Max*. See Changing the Scalar Range.	
	Off	Values above *Max* are displayed in the maximum color, and values below *Min* are displayed in the minimum color.
	Below Min, Above Max	Does not display values outside the range.
	Below Min	Does not display values below *Min*.
	Above Max	Does not display values above *Max*.
Units	Selects the type of units to display. See Changing the Scalar Units.	

Min Magnitude	Sets the minimum value in the displayed range.
Max Magnitude	Sets the maximum value in the displayed range.

Copying Properties of a Vector Field

It is possible to copy certain properties of one vector field object onto another.

When properties are copied from one vector field to another, the following properties are changed:

- *Auto Scale*
- *Function*
- *Units*

However, since the other properties are affected by which parts are displayed in the scene, you must copy the displayer properties to make the destination vector field match the source one completely.

Defining a Custom Vector

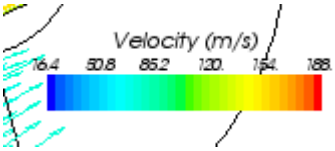
In addition to the standard system field-functions, you can also display custom user-defined field functions.

Once you create the desired field function, simply select it from the list of available field functions as before.

Using the Color Bar to Modify the Vector Display

The vector bar of the scene display shows the scale of the magnitude of the chosen vector field function.

The appearance of the vector bar can be manipulated using the color bar in the object tree.

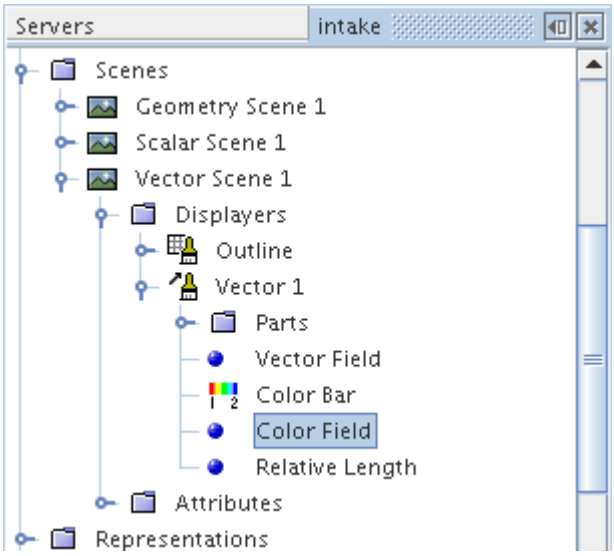


The range that is shown on the vector bar corresponds to the full range of the vector magnitude on all parts in the displayer. The units may be changed in the same way as the units of a scalar displayer.

Coloring Glyphs Based on Scalar Values

You can display scalar values on the vector glyphs.

The **Color Field** node, which has properties and is subject to common menu items, appears under the vector displayer node when the displayer's *Color Mode* property is set to `Scalar Field`.



It colors the glyphs based on the values of the scalar field function that you select.

Color Field Properties

Function	Selects the type of scalar field function to show, such as temperature or velocity magnitude. See Field Functions.	
Auto Min	Indicates the minimum value among all parts in the displayer (read-only).	
Auto Max	Indicates the maximum value among all parts in the displayer (read-only).	
Auto Range	Determines whether to set the range of values to display automatically. See Changing the Scalar Range.	
	Off	Does not use auto ranging, rather uses the values that are set in *Min* and *Max*.
	Min and Max Values	Automatically determines the range from the field function data.
	Min Value	Automatically determines the minimum value from the field function data. The property switches automatically from `Min and Max Values` to this setting if you make an entry for *Max Value*.
	Max Value	Automatically determines the minimum value from the field function data. The property switches automatically from `Min`

		`and Max Values` to this setting if you make an entry for *Min Value*.
Auto Range Mode	Chooses the domain for auto-range minimum and maximum value computation that is used for color coding.	
	Displayer Parts	Parts added to the particular displayer.
	Displayer Regions	Regions from which the displayer extracts data.
	All Regions	All regions in the simulation.
Clip	Determines how to display values outside of the range that is specified by *Min* and *Max*. See Changing the Scalar Range.	
	Off	Values above *Max* are displayed in the maximum color, and values below *Min* are displayed in the minimum color.
	Below Min, Above Max	Does not display values outside the range.
	Below Min	Does not display values below *Min*.
	Above Max	Does not display values above *Max*.
Units	Selects the type of units to display. See Changing the Scalar Units.	
Min	Sets the minimum value in the displayed range.	
Max	Sets the maximum value in the displayed range.	

Vector Displayer Properties

Use the properties of the vector displayer to control characteristics such as display mode, colors, and projection.

Display Mode	Controls how the vector field is displayed.	
	Glyph	Uses glyphs to display vectors. Activates the Glyph manager node. See Customizing Glyphs.
	Line Integral Convolution	Uses a texture-based technique to display vectors. Activates the Line Integral Convolution node. See Working with Line Integral Convolution.
Color Mode	Controls the arrangement of colors.	

	Vector Magnitude (Default)	Colors the glyphs based on values of a selected vector field.
	Scalar Field	Adds the **Color Field** subnode to the displayer node to color glyphs based on values of a selected scalar field. See Coloring Glyphs Based on Scalar Values.
	Constant	Adds the **Color** subnode to the displayer node to display using a custom color. See Changing the Color of the Surface Using the Displayer.
	Region	Colors the glyphs by region.
Projection Mode	Controls the projection of vectors relative to the surface. See Controlling the Projection of the Glyphs.	
	None (Default)	Leaves the vectors with both normal and tangential components.
	Normal	Removes the tangential component of the vectors, projecting them with the normal component.
	Tangential	Removes the normal component of the vectors, projecting them with the tangential component.
Opacity	Adjusts the transparency of the part. The settings range from 0.0 (invisible) to 1.0 (opaque), and can be controlled with a slider. This property allows you to show an overlay of different parts of the model. See Making Displayer Surfaces Transparent.	

Vector Displayer Expert Properties

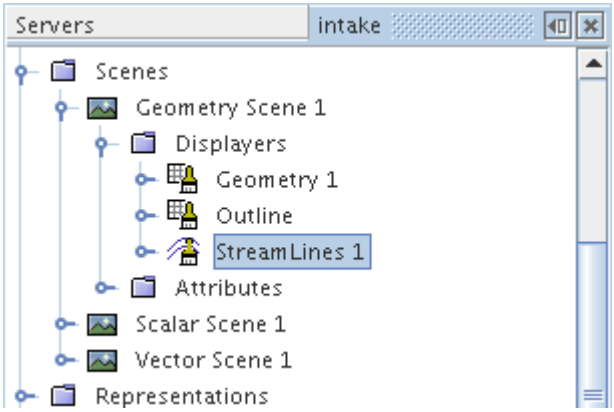
Representation	Selects a representation. Additional selections would appear as representations are added to the simulation.	
Line Width	Adjusts the width in pixels of lines that are used in the display, depending on the type of displayer. For example, in a scalar displayer, this setting would be effective only if the *Contour Style* property were set to `Line`. This property can be set with a slider.	
Specular Highlight Mode	Specular highlighting is the reflection of a light source on a shiny surface (such as polished metal).	
	Material	Uses the color of the material (default). For example, on a black

		surface, no specular highlighting is visible.
	Light	Uses the color of the light. To specify this color, see Controlling the Color of Lighting Objects.
Lighting	Controls the lighting of the part. Accessing the property customizer opens the *Lighting* dialog. See Lighting Displayer Surfaces.	
Data Focus	Selects a data focus object to apply to this displayer. The choice of data focus objects depends on the total number of these objects in the simulation. When a data focus object is selected, a data focus sub-node is added to this node. See Exploring Solutions with Data Focus.	
Smooth Shade	Controls the transition between segments of the image.	
	Activated	Makes the transitions smooth for a more realistic effect.
	Deactivated	Leaves this function off.
Ignore Clip	Controls whether the parts in the displayer are affected by any enabled clip plane. See Using Clip Planes.	
	Activated	Shows the parts unaffected by the clip plane.
	Deactivated	Shows the parts that are trimmed by the clip plane.
Face Culling	Controls whether and how to cull polygons. See Culling Polygons.	
	None	Does not cull any polygons.
	Front	Culls (omits from the display) all polygons that are oriented toward the front of the view.
	Back	Culls (omits from the display) all polygons that are oriented toward the back of the view.
Transform	Selects a transform to apply to all parts in the displayer. **None** is the default selection. Additional selections would appear when you create transforms. See Creating a New Transform. This feature allows for the creation of symmetric and periodic repeats, as well as arbitrary transforms to be used.	

Displaying Streamlines

Streamline displayers are used to display derived streamline parts in a scene (provided you have created such parts). They cannot be used to create streamlines.

The **Streamlines** displayer node, which has its own properties and shares a pop-up menu with other types of displayers, controls the appearance of those streamlines.



In the icon, the lines next to the paint brush signify the depiction of streamlines.

The scene that displays streamlines includes a streamline bar which functions like a scalar bar. This bar shows the scale of the magnitude of the field function that is used to generate the streamlines (by default velocity magnitude). Its appearance can be manipulated using the color bar in the object tree.

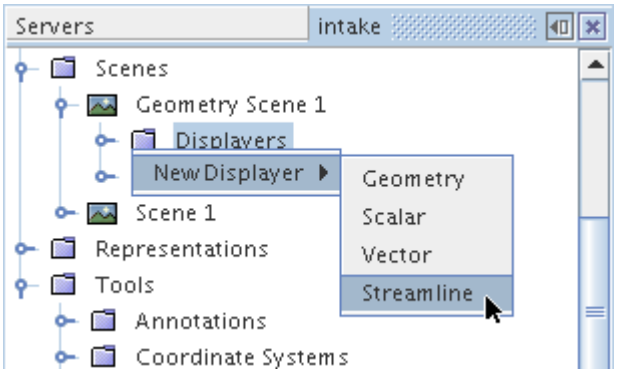
Contents:

Creating Streamline Displayers
Changing the Streamline Style
Changing the Size of the Streamlines
Displaying a Scalar on Streamlines
Adding Arrows to Streamlines
Streamline Displayer Properties

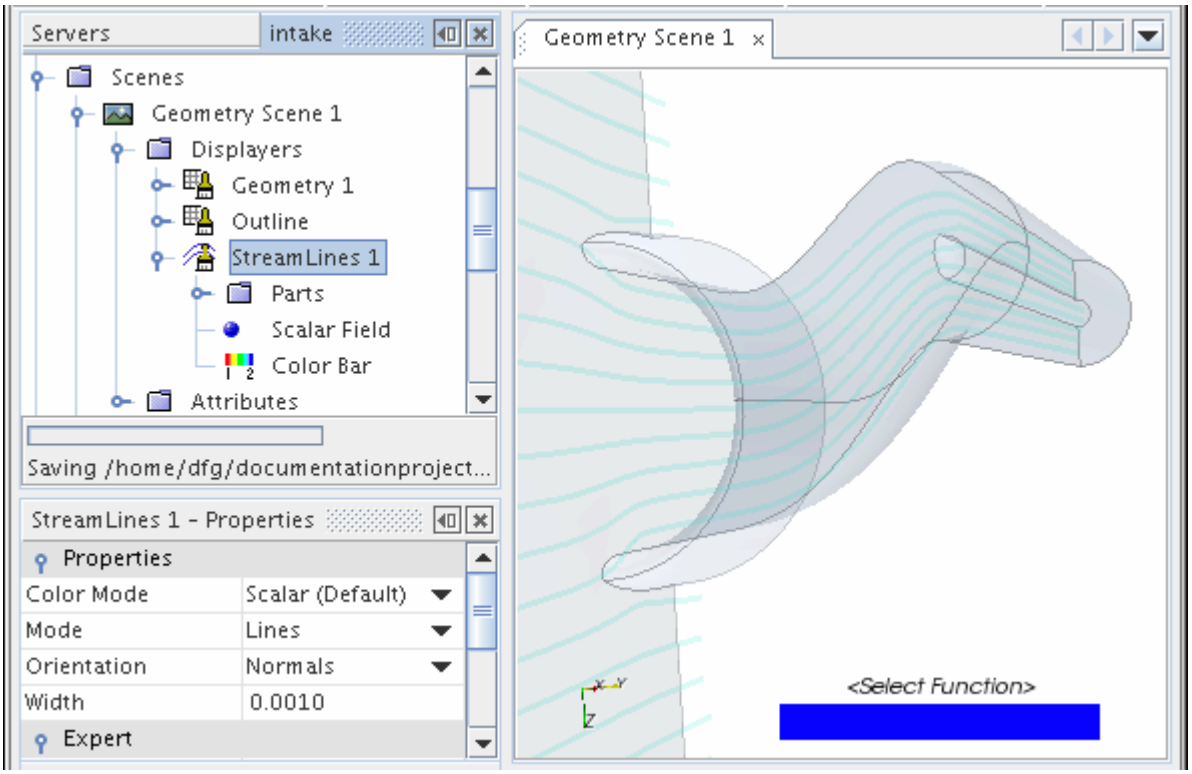
Creating Streamline Displayers

Streamline displayers are not created automatically through the creation of any scene and have to be added.

To add a streamline displayer, open the **Scenes** node and the node of the chosen scene, right-click on the **Displayers** node and select **New Displayer** > **Streamline**.



By default streamline displayers contain no parts and only derived streamline parts may be added. Thus before you can display streamlines you must create at least one derived streamline part. Any number of derived streamline parts can be added to a streamline displayer.

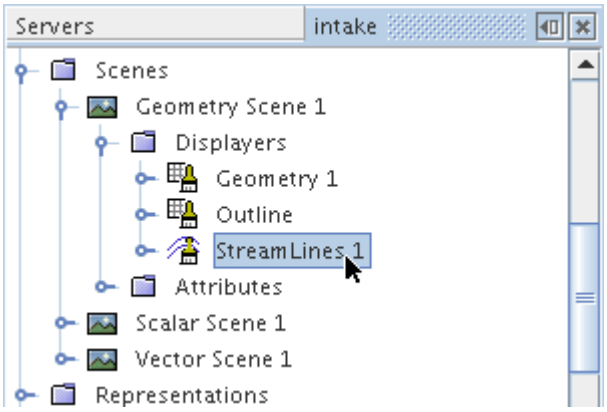


The default name of the displayer is based on its type. However, renaming the displayer is possible.

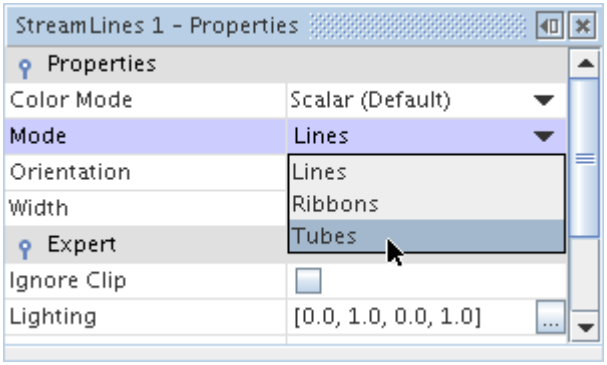
Changing the Streamline Style

The streamlines can be displayed using various styles: lines, ribbons, or tubes.

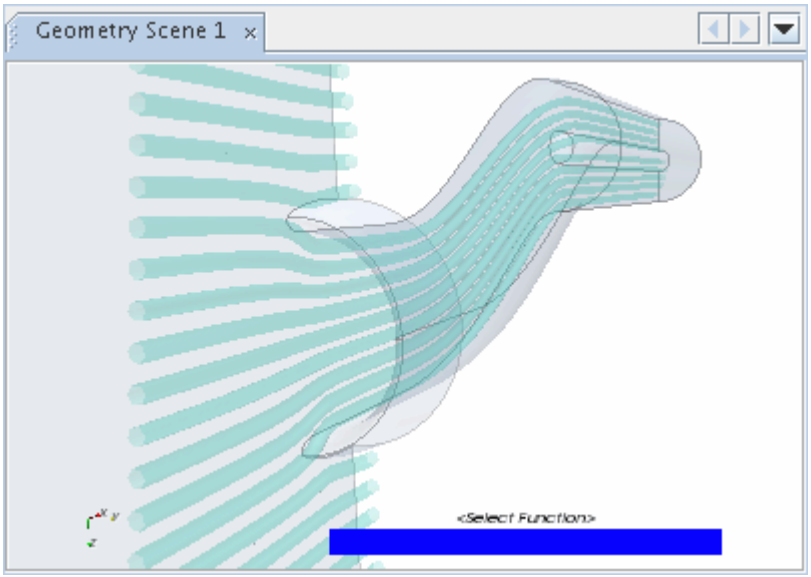
By default, the streamlines are displayed as lines. To change the style of the streamlines, expand the **Scenes** node and expand a scene which contains a streamline displayer. Expand the **Displayers** node and select the streamline displayer node.



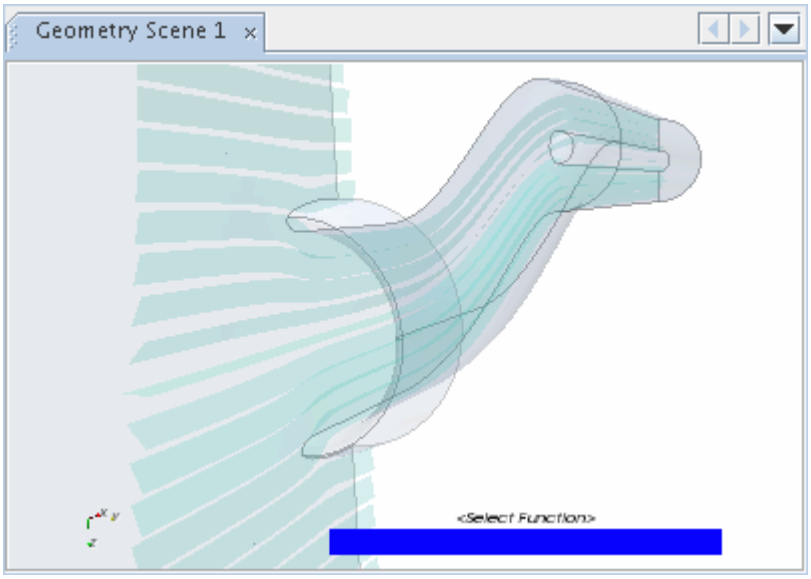
Set the style in the drop-down menu under the *Mode* property in the *Properties* window.



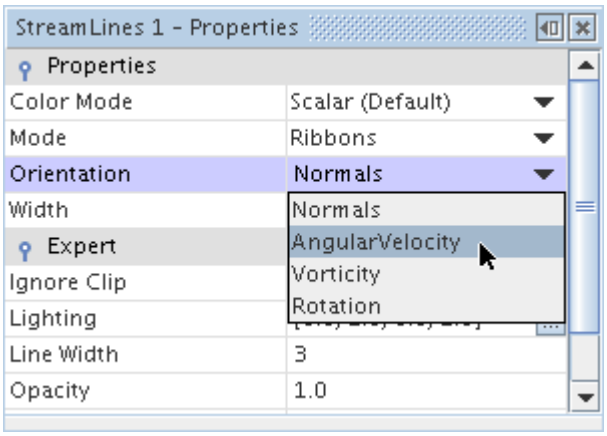
The `Tubes` option displays the streamlines as cylinders with the diameter of the cylinder being governed by the *Width* property.



The `Ribbons` mode displays the streamlines as flat surfaces with a width governed by the *Width* property.



In addition, the ribbons can be twisted based on the setting of the *Orientation* property.

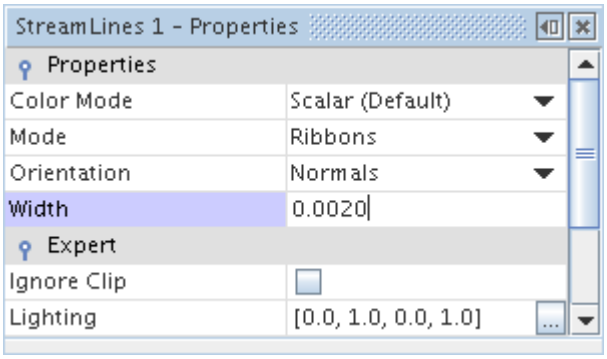


Changing the Size of the Streamlines

If either the ribbon or tube mode is used to display the streamlines, the size of the ribbons or tubes can be changed.

Expand the **Scenes** node and expand a scene which contains a streamline displayer. Expand the **Displayers** node and select the streamline displayer node.

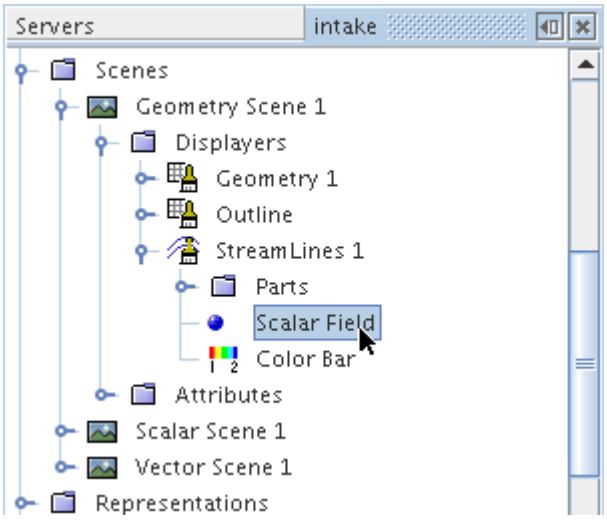
Adjust the width of the ribbons or tubes using the *Width* property in the *Properties* window.



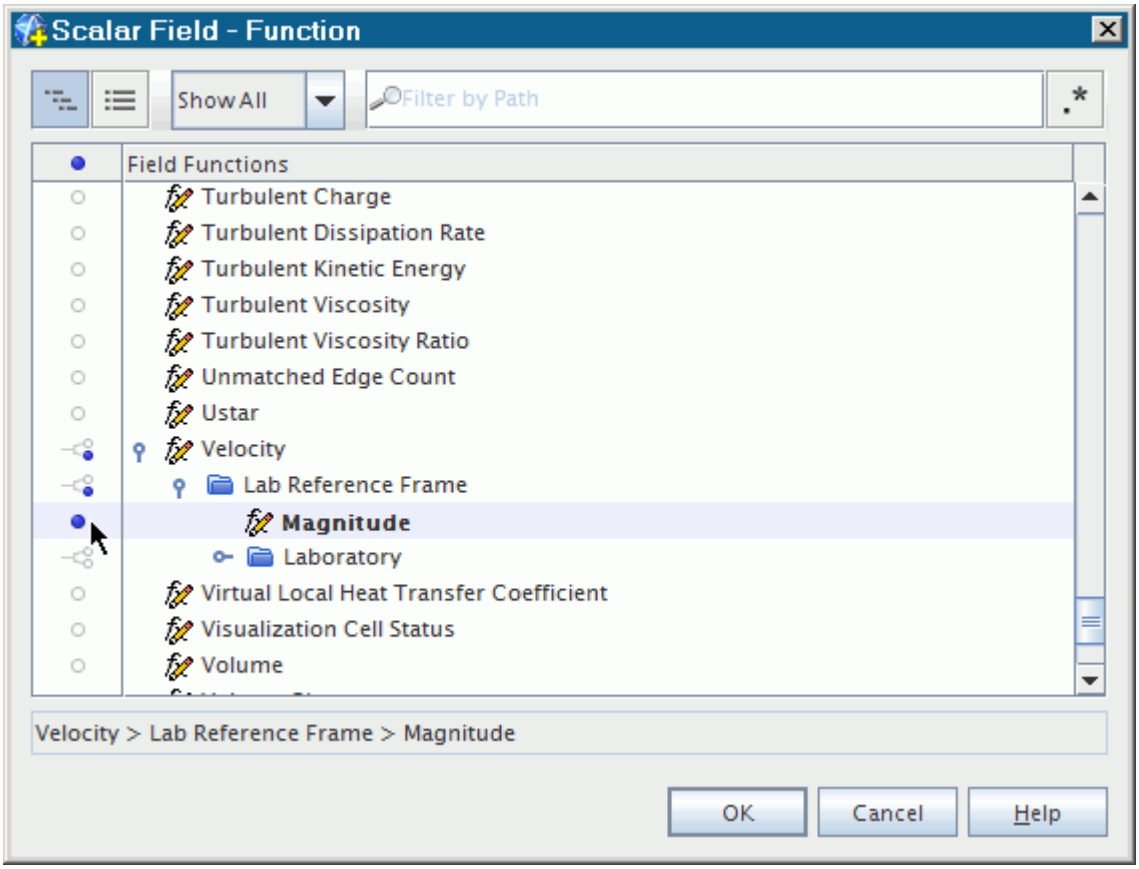
Displaying a Scalar on Streamlines

Streamlines can be colored using any scalar field function.

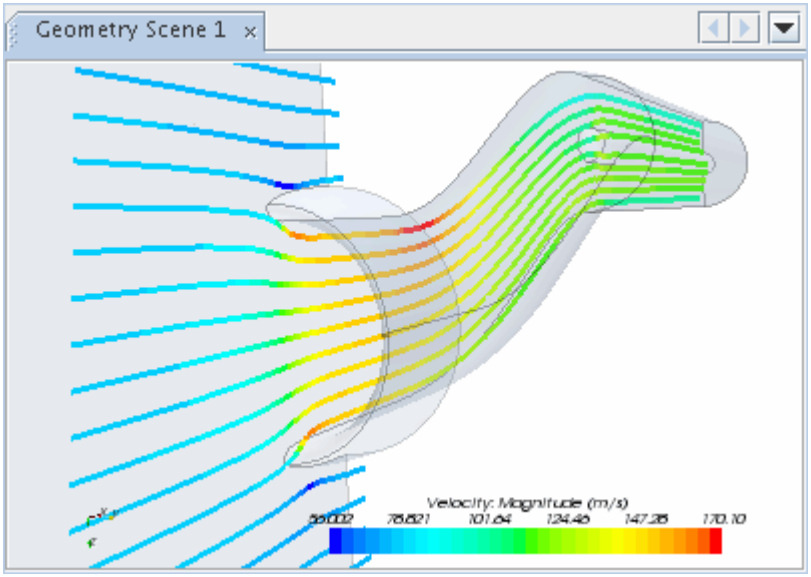
Expand the **Scenes** node and expand a scene which contains a streamline displayer. Expand the **Displayers** node and the streamline displayer node, and select the **Scalar Field** node.



By default the function is not set. To set the scalar field function, choose from the list in the object selector dialog. To activate the dialog, edit the *Function* property.



The scalar values appear on the streamlines in the display.

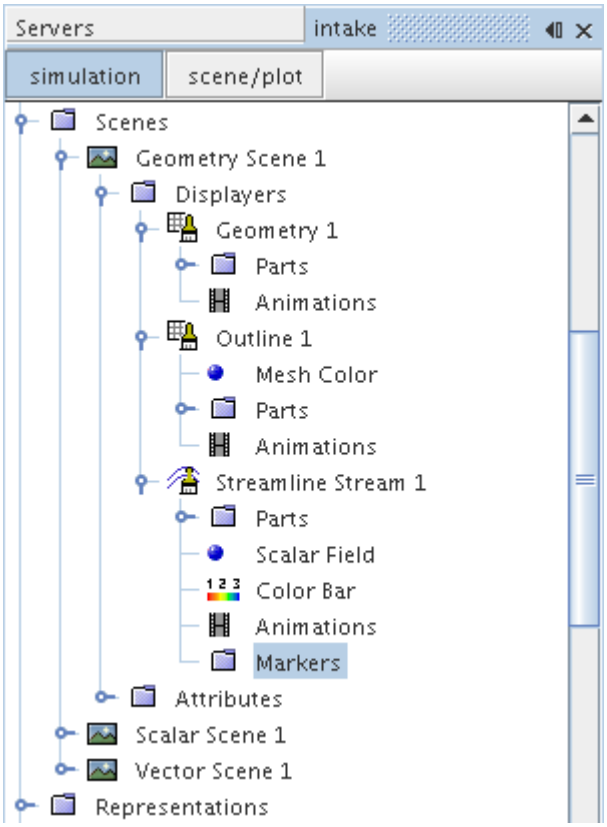


The units and range of the scalar field function can be changed in the same way as in the Scalar Displayer.

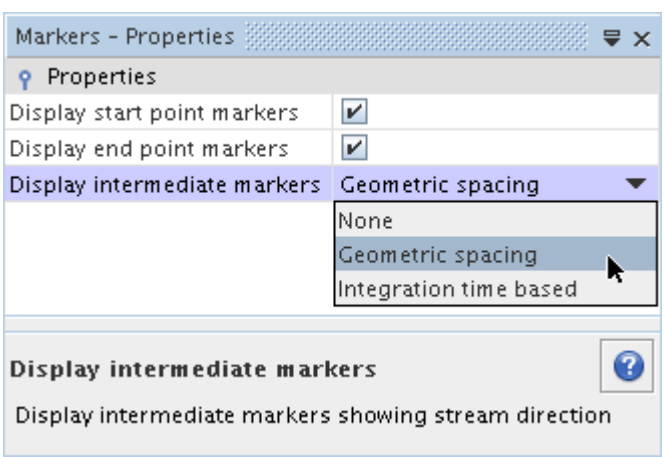
Adding Arrows to Streamlines

Arrows can be added to streamlines in order to give them a direction.

Expand the **Scenes** node and expand a scene which contains a streamline displayer. Expand the **Displayers** node and the streamline displayer node, and select the **Markers** node.

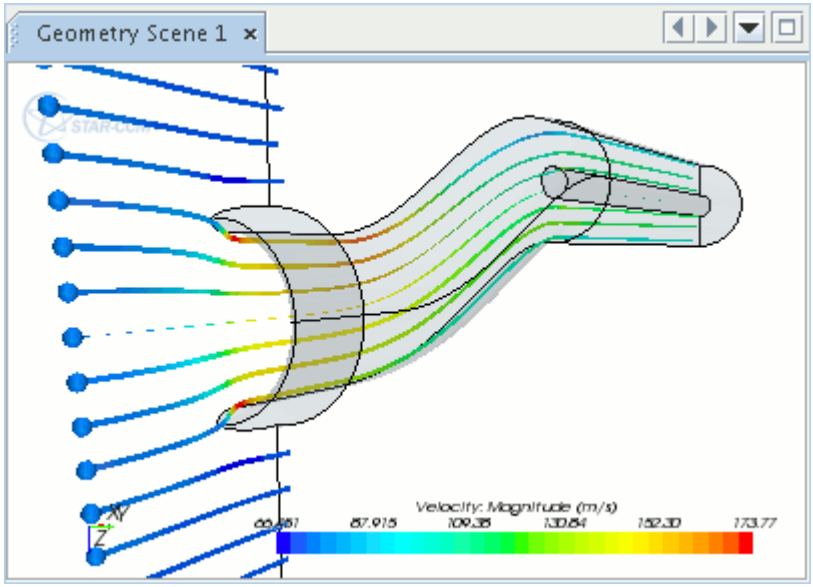


By default, there are no arrows that are displayed on streamlines. To display arrows on streamlines, select the desired function in the *Properties* window.

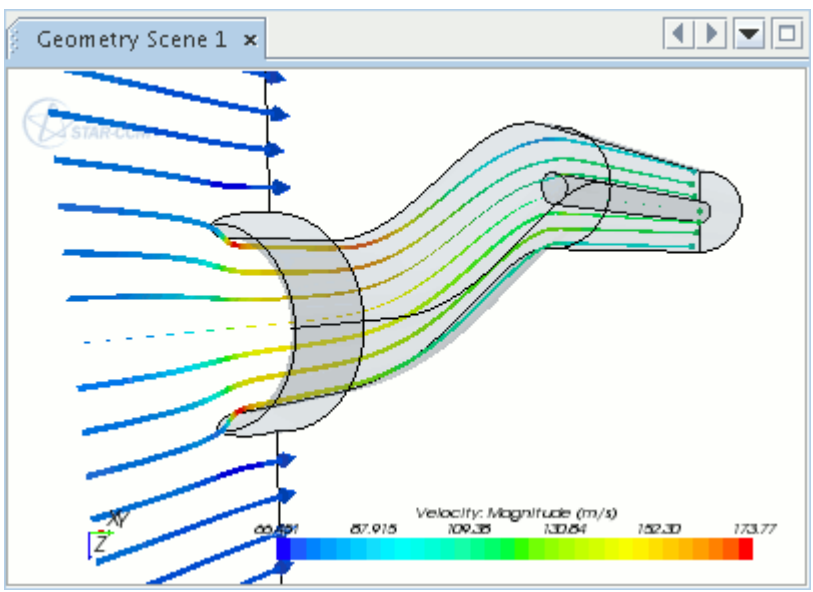


The following properties can be set in the *Markers - Properties* window:

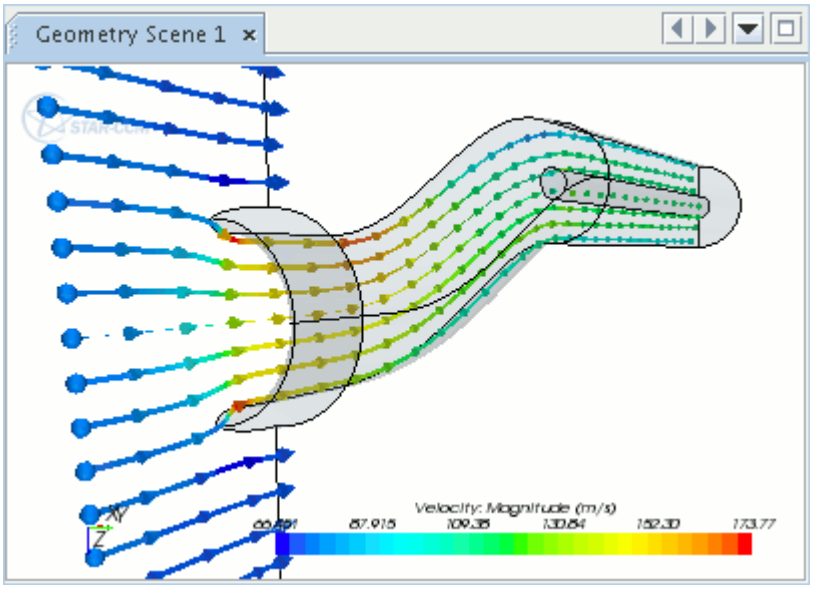
- *Display start point markers*: activating this checkbox displays spheres at the start point of the streamlines.



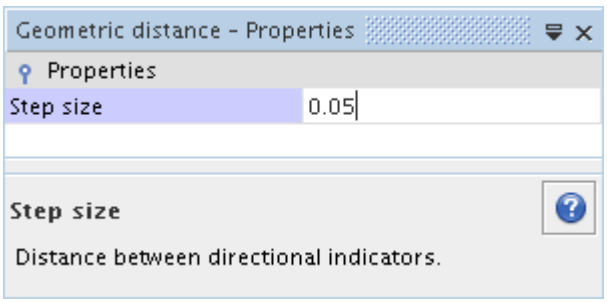
- *Display end point markers*: activating this checkbox displays conical arrow heads at the end points, pointing in the direction of the flow.



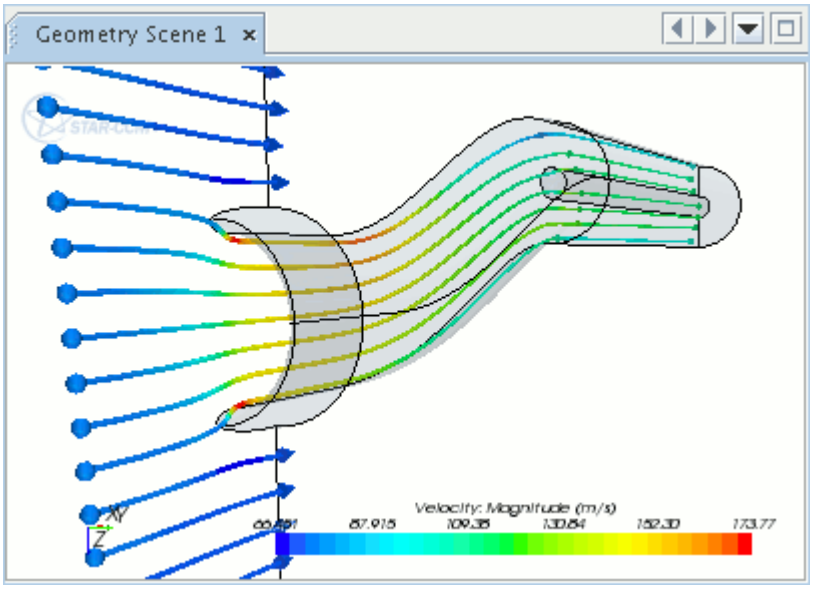
- *Display intermediate markers*:
  - **None**: no arrow appears between the start points and the end points on the streamlines.
  - **Geometric Spacing**: selecting this adds arrows, equally spaced out, between the start points and end points on the streamlines.



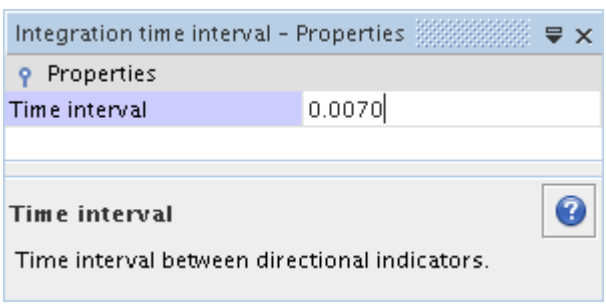
Selecting this option creates a node in the **Markers** folder where the distance between the arrows can be controlled using the *Step size* in the *Geometric Distance - Properties* window.



- **Integration time based**: this adds arrows between the start points and end points on the streamlines that are displayed with relation to the time step.



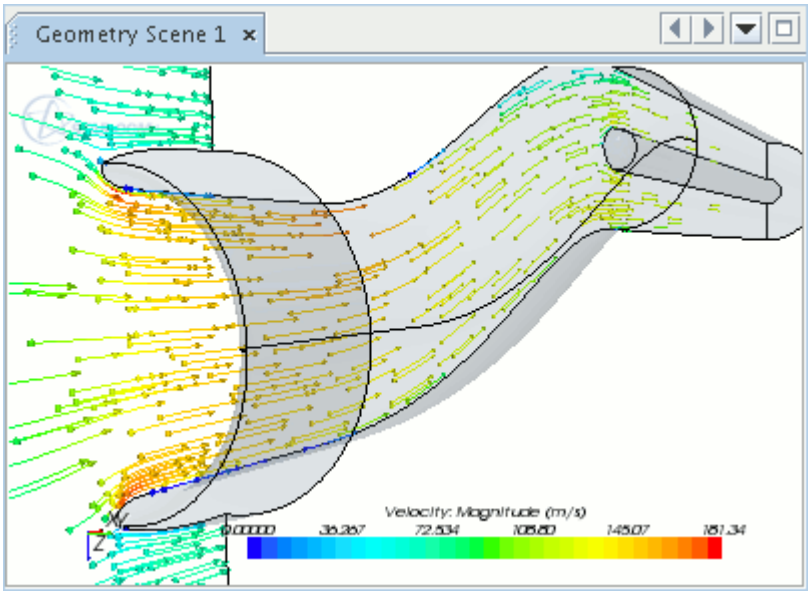
Selecting this option creates a node in the **Marker** folder where the gap between the arrows can be controlled by adjusting the *Time interval* in the *Integration time interval - Properties* window. This option is useful for visualizing the speed of the flow.



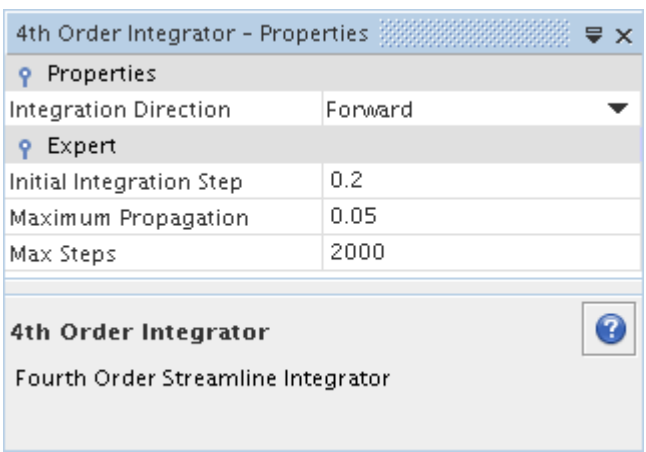
The color of the arrows is controlled by the scalar field chosen for the streamlines or by the color selected in the *Properties* window when the Color Mode is set to **Constant**. The size of the spheres and cones is scaled according to the size of the streamlines and can be controlled using the *Width* property in the *Properties* window.

Creating Streamlets

Reducing the streamline length and increasing the number of seed points can produce streamlets (short streamlines) as shown below.



One way of achieving this effect is to create a constrained streamline on a plane or part, reduce the integration time of the streamlines (*Initial Integration Step* and *Maximum Propagation Step*), and increase the number of Seed Points. The integrator properties that are used for the above example are shown below.



To achieve the streamlets effect shown above, it is best to only have start_point and end_point markers displayed, although it is possible to have streamlets with no markers displayed. The streamlets effect can be achieved with any type of streamlines.

Streamline Displayer Properties

Use the properties of the streamline displayer to control characteristics such as streamline style, width, and colors.

Width	Controls the size of the ribbons or tubes. The units are of the mesh that is imported into the simulation. See Changing the Size of the Streamlines.

Orientation	Defines how ribbon streamlines are twisted. Only valid if the *Mode* property is set to `Ribbons`.	
	Normals	Normal twisting.
	Angular Velocity	Use the angular velocity of the vector to twist then ribbon.
	Vorticity	Use the local vorticity to twist the ribbon.
	Rotation	Use rotation to twist the ribbon.
Mode	Defines the style of streamline. See Changing the Streamline Style.	
	Lines	Shows streamlines in a linear format. This is the default setting.
	Ribbons	Shows streamlines as flat bands. This format depicts the local vorticity of particles moving through flow. Unlike lines, ribbons provide surfaces that can be viewed clearly with the Surface Lighting feature. See Lighting Displayer Surfaces.
	Tubes	Shows streamlines as tubes. Tubes are more suitable for presentation purposes, since, unlike ribbons, they do not indicate particle orientation.
Color Mode	Controls the arrangement of colors.	
	Scalar (Default)	Colors the streamlines according to the selected scalar field function.
	Constant	Adds the **Color** subnode to the displayer node to display surfaces using a custom color.
	Region	Colors the surfaces by region.
Opacity	Adjusts the transparency of the part. The settings range from 0.0 (invisible) to 1.0 (opaque), and can be controlled with a slider. This property allows you to show an overlay of different parts of the model. See Making Displayer Surfaces Transparent.	

Streamline Displayer Expert Properties

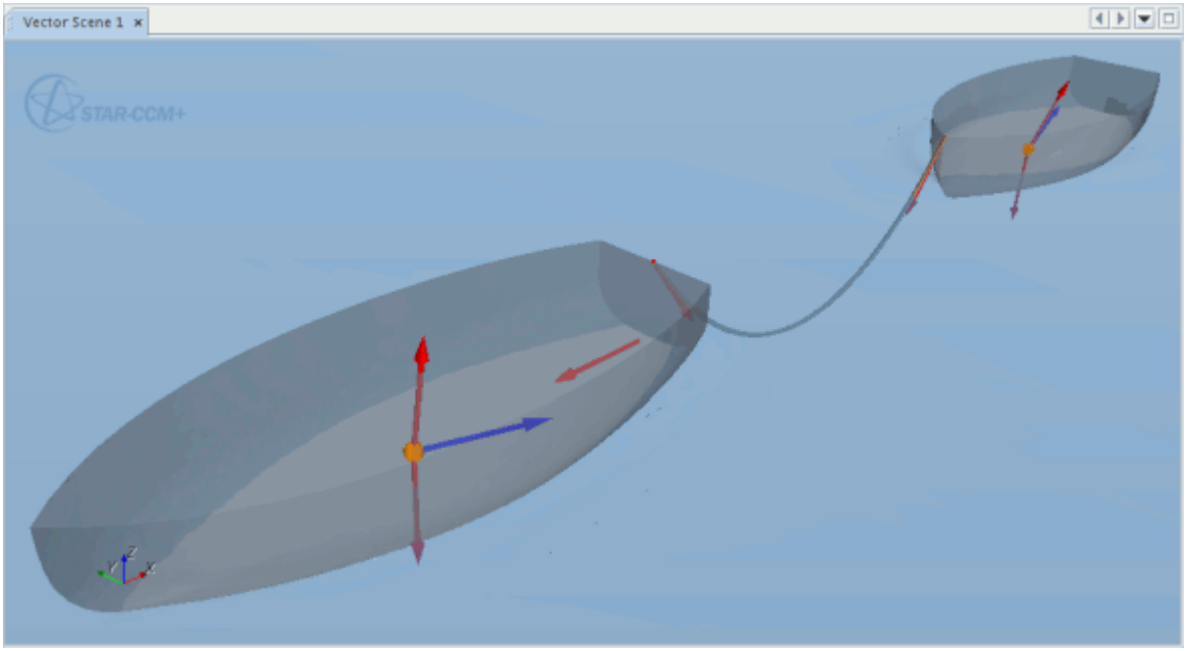
Representation	Selects a representation. Additional selections would appear as representations are added to the simulation.

Line Width	Adjusts the width in pixels of lines that are used in the display, depending on the type of displayer. For example, in a scalar displayer, this setting would be effective only if the *Contour Style* property were set to `Line`. This property can be set with a slider.	
Specular Highlight Mode	Specular highlighting is the reflection of a light source on a shiny surface (such as polished metal).	
	Material	Uses the color of the material (default). For example, on a black surface, no specular highlighting is visible.
	Light	Uses the color of the light. To specify this color, see Controlling the Color of Lighting Objects.
Lighting	Controls the lighting of the part. Accessing the property customizer opens the *Lighting* dialog. See Lighting Displayer Surfaces.	
Data Focus	Selects a data focus object to apply to this displayer. The choice of data focus objects depends on the total number of these objects in the simulation. When a data focus object is selected, a data focus sub-node is added to this node. See Exploring Solutions with Data Focus.	
Smooth Shade	Controls the transition between segments of the image.	
	Activated	Makes the transitions smooth for a more realistic effect.
	Deactivated	Leaves this function off.
Ignore Clip	Controls whether the parts in the displayer are affected by any enabled clip plane. See Using Clip Planes.	
	Activated	Shows the parts unaffected by the clip plane.
	Deactivated	Shows the parts that are trimmed by the clip plane.
Face Culling	Controls whether and how to cull polygons. See Culling Polygons.	
	None	Does not cull any polygons.
	Front	Culls (omits from the display) all polygons that are oriented toward the front of the view.
	Back	Culls (omits from the display) all polygons that are oriented toward the back of the view.
Transform	Selects a transform to apply to all parts in the displayer. **None** is the default selection. Additional selections would appear when you create transforms. See Creating a New Transform.	

	This feature allows for the creation of symmetric and periodic repeats, as well as arbitrary transforms to be used.

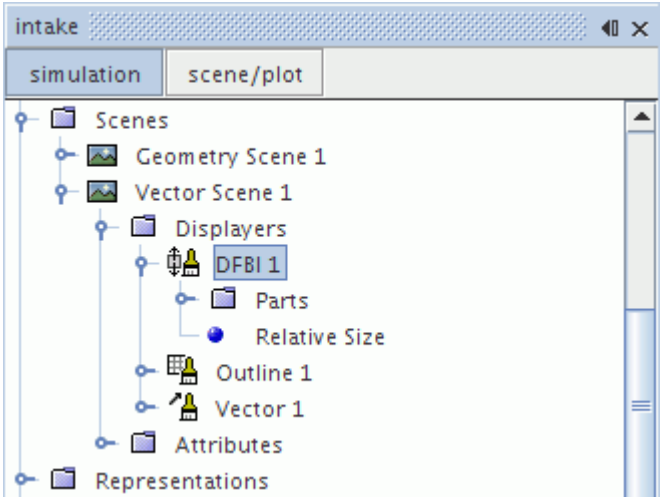
Displaying Dynamic Fluid Body Interaction (DFBI)

It is possible to display DFBI entities such as bodies (including the center of mass), external forces and moments, and body couplings (springs and catenaries) in a scene.



To make the DFBI displayer available, activate DFBI in the simulation.

A DFBI displayer can be created using the pop-up menu of the displayer manager. The **DFBI** displayer node, which has properties and shares a pop-up menu with other displayer types, controls the display of this content. Specifically, it contains subobjects that let you change the size of glyphs.



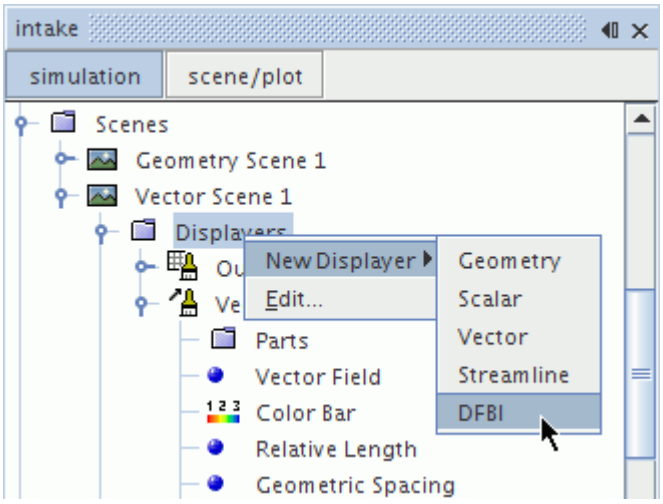
Contents:

Creating DFBI Displayers
Changing the Size of Glyphs
DFBI Displayer Properties

Creating DFBI Displayers

DFBI displayers are not created automatically through the creation of any scene and have to be added.

To add a DFBI displayer, open the **Scenes** node and the node of the chosen scene, right-click on the **Displayers** node and select **New Displayer** > **DFBI**.



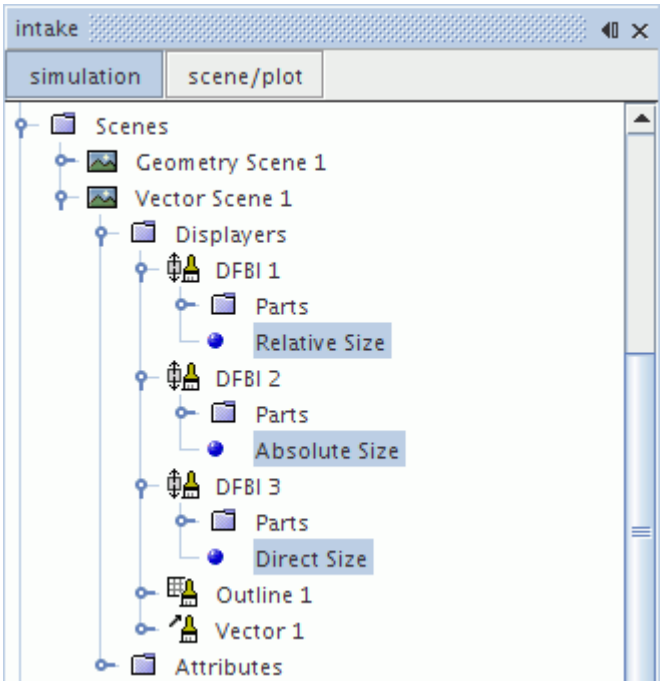
By default DFBI displayers contain no parts and only 6-DOF bodies, external forces and moments, or body couplings may be added. Thus before you can display DFBI you must create at least one of these related objects. Any number of 6-DOF bodies, external forces and moments, or body couplings can be added to a DFBI displayer.

The default name of the displayer is based on its type. However, renaming the displayer is possible.

Changing the Size of Glyphs

The DFBI displayer provides several features with which you can control the size of the glyphs.

These features include the **Relative Size**, **Absolute Size** and **Direct Size** objects, whose nodes have properties and are subject to common menu items.



The **Relative Size** node appears when the *Glyph Scale* property of the DFBI displayer is set to `Model Size` or `Screen Size`, the **Absolute Size** node appears when this property is set to `Absolute`, and the **Direct Size** node appears with a setting of `Direct`.

Vectors are scaled according to the *Glyph Scale* property of the DFBI displayer and its *Vector Length* property. The relationships among these properties and the relative or absolute length of the vector glyphs are outlined in the table below.

		Vector Length	
		`Vector Magnitude`	`Constant Length`
Glyph Scale	`Absolute`	• Glyph size remains the same independent of the model or the display. • *Vector Glyph Length* is a value; applies to size of largest vector glyph.	• Glyph size remains the same independent of the model or the display. • *Vector Glyph Length* is a value; applies to size of all vector glyphs.
	`Model Size`	• Glyph size changes with model size. • *Vector Glyph Length* is a percentage; applies to size of largest vector glyph.	• Glyph size changes with model size. • *Vector Glyph Length* is a percentage; applies to size of all vector glyphs.

		Vector Length	
	`Screen Size`	• Glyph size changes with screen size. • *Vector Glyph Length* is a percentage; applies to size of largest vector glyph.	• Glyph size changes with screen size. • *Vector Glyph Length* is a percentage; applies to size of all vector glyphs.
	`Direct`	• Glyph size represents the vector quantity, based on the scale factor entered in the *Vector Glyph Length* property. • *Vector Glyph Length* is the scale factor of the unit chosen.	• Glyph size remains the same independent of the model or the display. • *Vector Glyph Length* is a value; applies to size of all vector glyphs.

By default the vector glyphs are scaled to constant length. To change the scaling mode, expand the **Scenes** node and expand a scene which contains a DFBI displayer. Expand the **Displayers** node, select the DFBI displayer node and choose settings for the *Vector Length* and *Glyph Scale* properties.

Although the **Relative Size**, **Absolute Size** and **Direct Size** nodes seem to have the same properties, entries are retained separately for each combination of settings that are listed in the table above. For example, if your entry for *Vector Glyph Length* with an `Absolute` glyph scale causes the glyph to appear visibly longer than it did with a `Model Size` (relative) vector scale, and then you switch back to `Model Size`, you see the glyphs revert to their earlier size.

The Direct Length feature allows you to define a vector scale as a function of the absolute value of the vector. In this manner you can create animations where the vector length is consistent between frames, and also compare similar simulations with different maximum vector lengths.

Relative Size Properties

Vector Glyph Length (%)	The user-defined length of a vector glyph, in percent of the model size or the screen size, depending on the setting of the *Glyph Scale* property.
Coupling Glyph Diameter (%)	The user-defined diameter of a body coupling glyph, in percent of the model size or the screen size, depending on the setting of the *Glyph Scale* property.
Center Of Mass Glyph Diameter (%)	The user-defined diameter of a center of mass glyph, in percent of the model size or the screen size, respectively, depending on the setting of the *Glyph Scale* property.

Properties of Absolute Size and Direct Size

Units	The unit in which the size of a glyph is specified.	
Vector Glyph Length	The user-defined length of a vector glyph, given in the unit specified in the *Units* property.	
Coupling Glyph Diameter	The user-defined diameter of a body coupling glyph, given in the unit specified in the *Units* property.	
Center Of Mass Glyph Diameter	The user-defined diameter of a center of mass glyph, given in the unit specified in the *Units* property.	

DFBI Displayer Properties

Use the properties of the DFBI displayer to control characteristics such as DFBI entities, glyph scale, and colors.

Color Mode	Controls the arrangement of colors.	
	Type (Default)	Colors the glyphs according to their type (body hull, body center of mass, force vector, moment vector, linear spring coupling, catenary coupling). The default colors are as follows: • Body hull: slate gray • Body center of mass: dark orange • Force vector: red • Moment vector: blue • Linear spring coupling: steel blue • Catenary coupling: beige
	Distinguish Forces and Moments	Colors forces and moments according to their specific type (External Fluid Force and Moment, External Gravity Force, Force CM, Moment CM, Generalized Force, Propulsion Force, Damping Force, Damping Moment, Linear Spring Force, Catenary Force). All other parts are colored as in the setting `Type (Default)`.
	Constant	Adds the **Color** subnode to the displayer node to display parts

		using a custom color. See Changing the Color of the Surface Using the Displayer.
Glyph Scale	Defines how to scale the glyphs. See Changing the Size of Glyphs.	
	Absolute	Scales the glyphs based on the specified sizes in the **Absolute Size** node.
	Model Size	Scales the glyphs relative to the maximum body length based on the specified sizes in the **Relative Size** node. The maximum body length is the length of the largest body in the simulation. The length of a body is its largest extent measured in the three directions given by the axes of the body coordinate system.
	Screen Size	Scales the glyphs relative to the display height based on the specified sizes in the **Relative Size** node.
	Direct	Scales the glyphs based on the proportions specified in the **Direct Size** node.
Bodies	Controls the appearance of bodies.	
	Activated	Bodies are displayed in the scene.
	Deactivated	Bodies are not displayed.
Forces	Controls the appearance of forces.	
	Activated	Forces are displayed in the scene.
	Deactivated	Forces are not displayed.
Moments	Controls the appearance of moments.	
	Activated	Moments are displayed in the scene.
	Deactivated	Moments are not displayed.
Couplings	Controls the appearance of body couplings.	
	Activated	Body couplings are displayed in the scene.
	Deactivated	Body couplings are not displayed.
Vector Length	Mode that determines vector length.	

	Vector Magnitude	Uses relative vector magnitude, that is, the scales of all vectors are fractions of the largest one.
	Constant Length	Lets you scale vectors on the order of the model rather than on the data.
Opacity	Adjusts the transparency of the parts. The settings range from 0.0 (invisible) to 1.0 (completely opaque), and can be controlled with a slider. This property allows you to show an overlay of different parts of the model. See Making Displayer Surfaces Transparent.	

DFBI Displayer Expert Properties

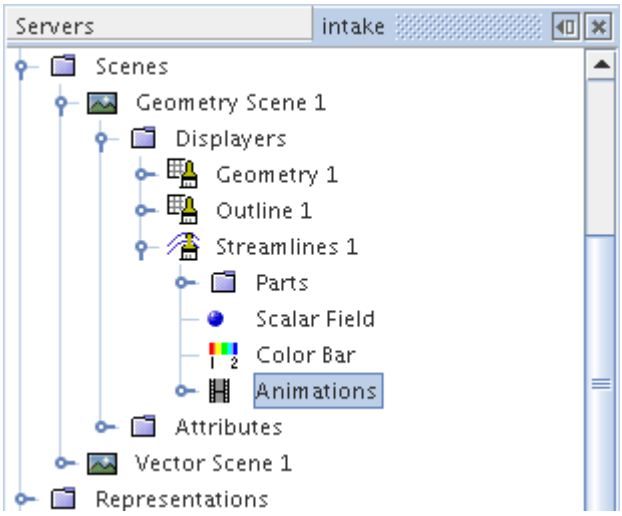
Vector Tip Scale	Sets the scale of the vector heads. See Controlling the Scale of the Glyph Heads.	
Specular Highlight Mode	Specular highlighting is the reflection of a light source on a shiny surface (such as polished metal).	
	Material	Uses the color of the material (default). For example, on a black surface, no specular highlighting is visible.
	Light	Uses the color of the light. To specify this color, see Controlling the Color of Lighting Objects.
Lighting	Controls the lighting of the parts. Accessing the property customizer opens the *Lighting* dialog. See Lighting Displayer Surfaces.	
Ignore Clip	Controls whether the parts in the displayer are affected by any enabled clip plane. See Using Clip Planes.	
	Activated	Shows the parts unaffected by the clip plane.
	Deactivated	Shows the parts that are trimmed by the clip plane.
Face Culling	Controls whether and how to cull polygons. See Culling Polygons.	
	None	Does not cull any polygons.
	Front	Culls (omits from the display) all polygons that are oriented toward the front of the view.
	Back	Culls (omits from the display) all polygons that are oriented toward the back of the view.

Transform	Selects a transform to apply to all parts in the displayer. **None** is the default selection. Additional selections would appear when you create transforms. See Creating a New Transform. This feature allows for the creation of symmetric and periodic repeats, as well as arbitrary transforms to be used.

Animating Scenes

You can animate information in a scene by activating the animation object within the displayer of the scene.

This object is represented by the **Animations** node, which has properties and is subject to common menu items.

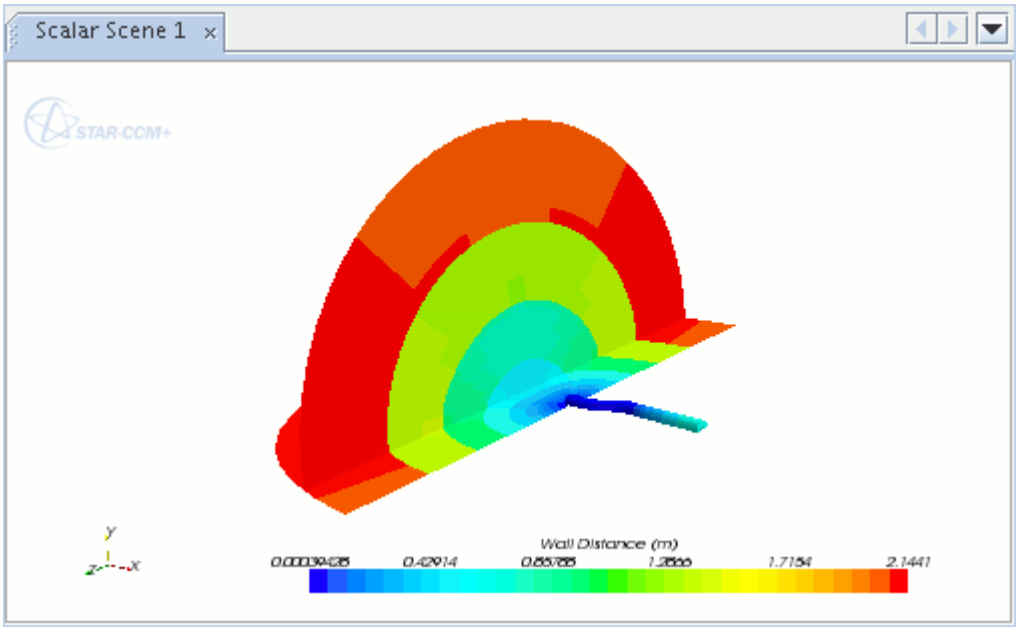


Contents:

Which Kinds of Animation Are Available?
Controlling the Animation
Recording the Animation onto Separate Files
Using Vector Glyphs
Using Streamline Tracers
Sweeping Parts Animation
Animations Properties

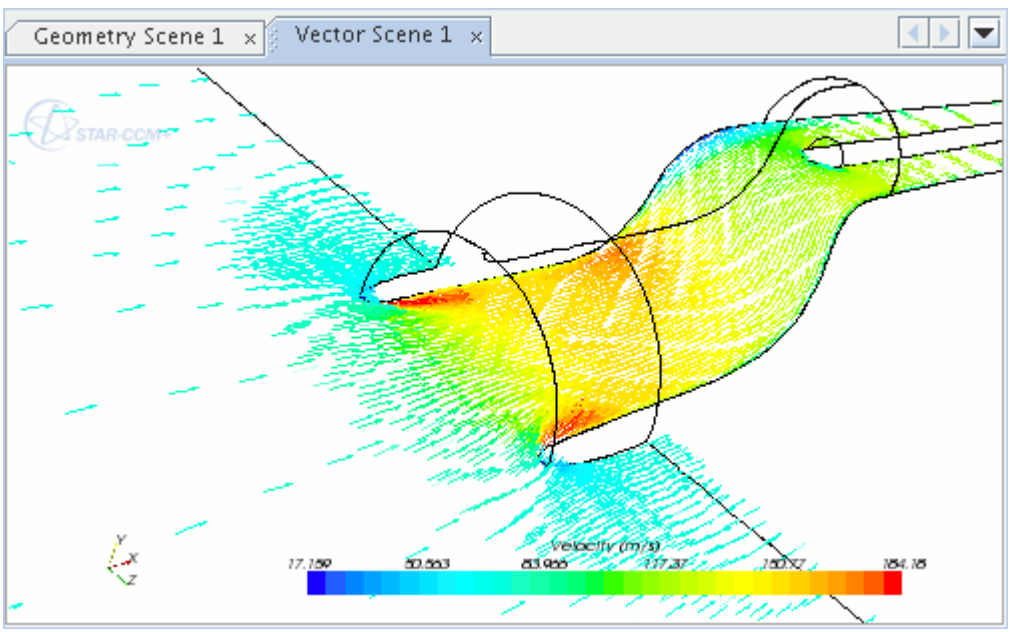
Which Kinds of Animation Are Available?

Scalars can be animated with the sweeping of a plane section or an iso surface. However, their other characteristics are controlled through the scalar displayer.



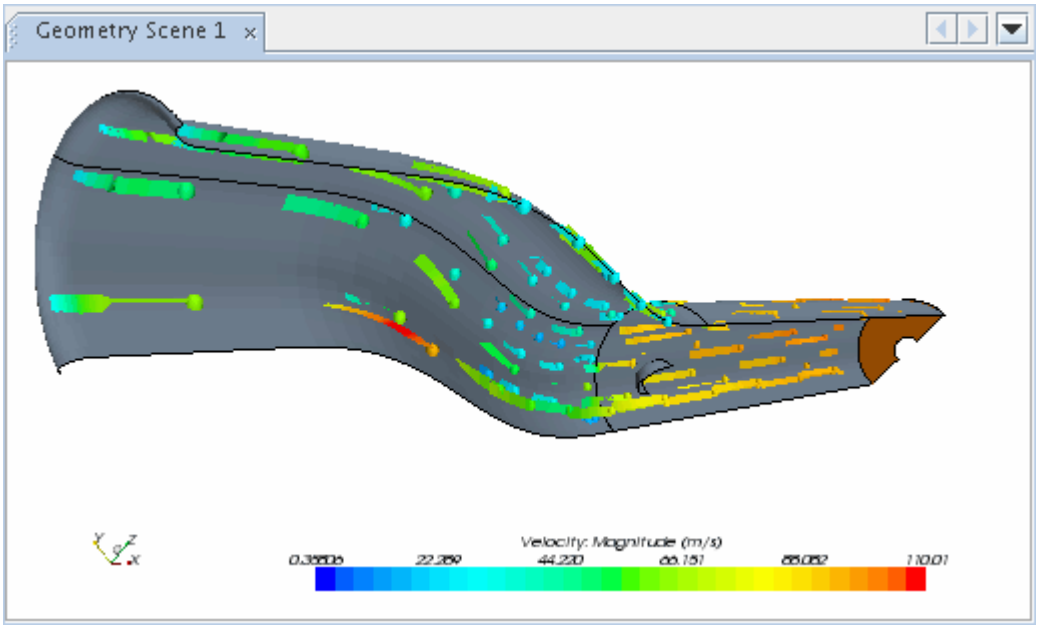
The scalars that are shown here represent wall distance.

Vector glyphs are animated with the **Animations** node. They can also move while a sweeping derived part is moving. However, their other characteristics are controlled through the vector displayer.



For example, the preceding screenshot shows pulsing vectors with a glyph length setting of 0.5% and a field function of **Velocity**.

Streamline animation allows you to depict flow in motion with a series of objects that are called tracers, which move along the streamline paths. Each tracer consists of a particle, or head, and a tail.



Many of the characteristics of the tracers can be adjusted through the properties of the **Streamline Settings** node. However, the shape of the tail is set by the streamline style, and the color indicates the streamlines' scalar selection. For example, the preceding screenshot shows tracers with a streamline setting of **Ribbons** and a scalar setting of **Velocity: Magnitude**.

Controlling the Animation

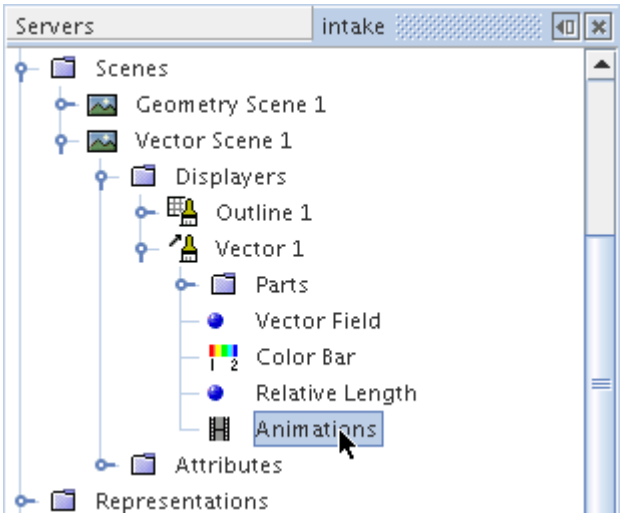
The various types of animation of a scene share several common controls.

These controls are:

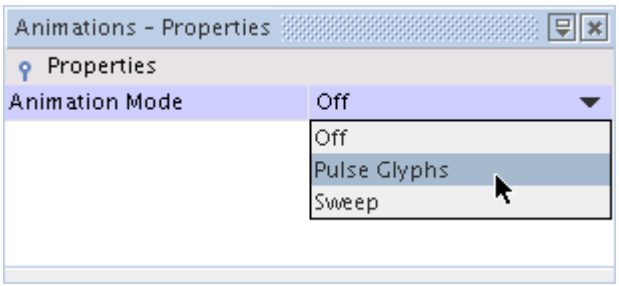
- Setting up the Animation
- Controlling the Movement of the Animation
- Controlling the Speed of the Animation

Setting up the Animation

To set up the animation, select the **Animations** node of the displayer.

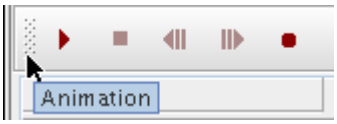


In the *Properties* window, change the *Animation Mode* property from `Off` to `Pulse Glyphs` for vectors (as shown in the following screenshot), to `Tracers` for streamlines, or to `Sweep` for a derived part.



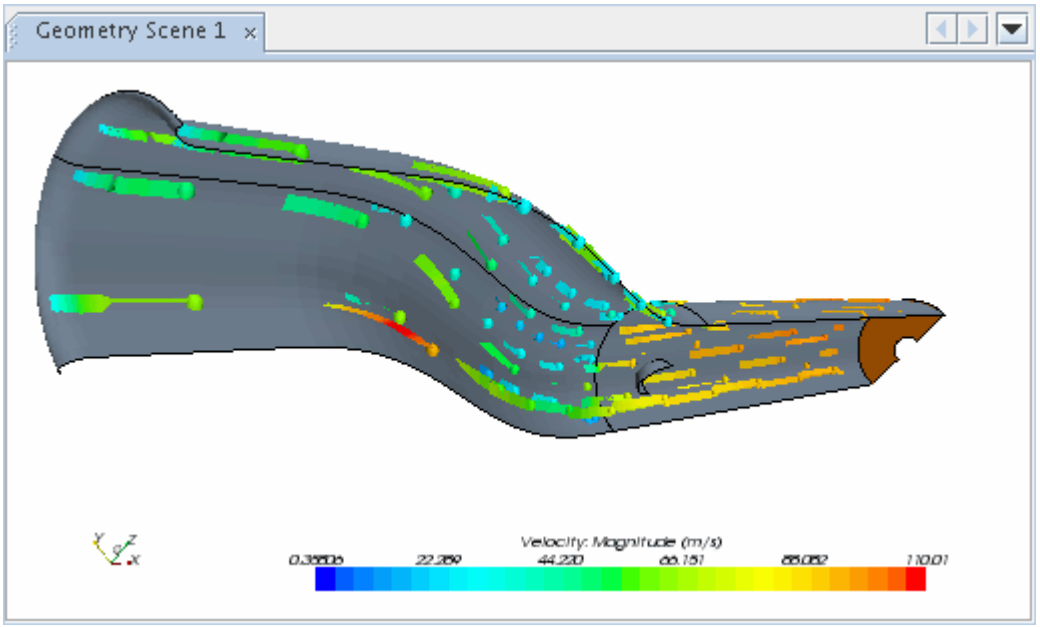
This step adds a subnode to the **Animations** node.

The **Animation** toolbar is available by default. If not, display it.



Controlling the Movement of the Animation

After you set up the animation, click the (**Play/Pause Animation**) toolbar button to start it. This causes the animation to move at its default settings.



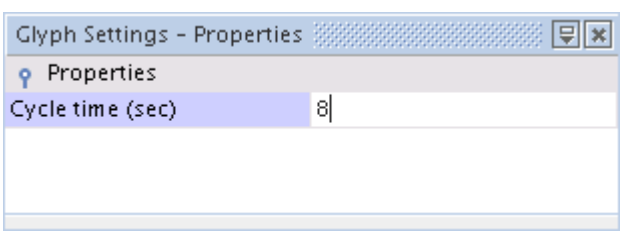
To pause the motion, click the (**Play/Pause Animation**) button again. This button toggles between motion and freeze-frame.

When the animation is in freeze-frame, you can move the animation ahead or back one frame at a time. Click the (**Single Step Forward**) to move ahead and the (**Single Step Backward**) button to move back.

To stop the motion, click the (**Stop Animation**) button. In streamline animations, this restores the standard appearance of the streamlines.

Controlling the Speed of the Animation

To make the tracer, vector glyph, or sweeping derived part move faster or slower, select its node and enter a number of seconds in the *Cycle time* property.



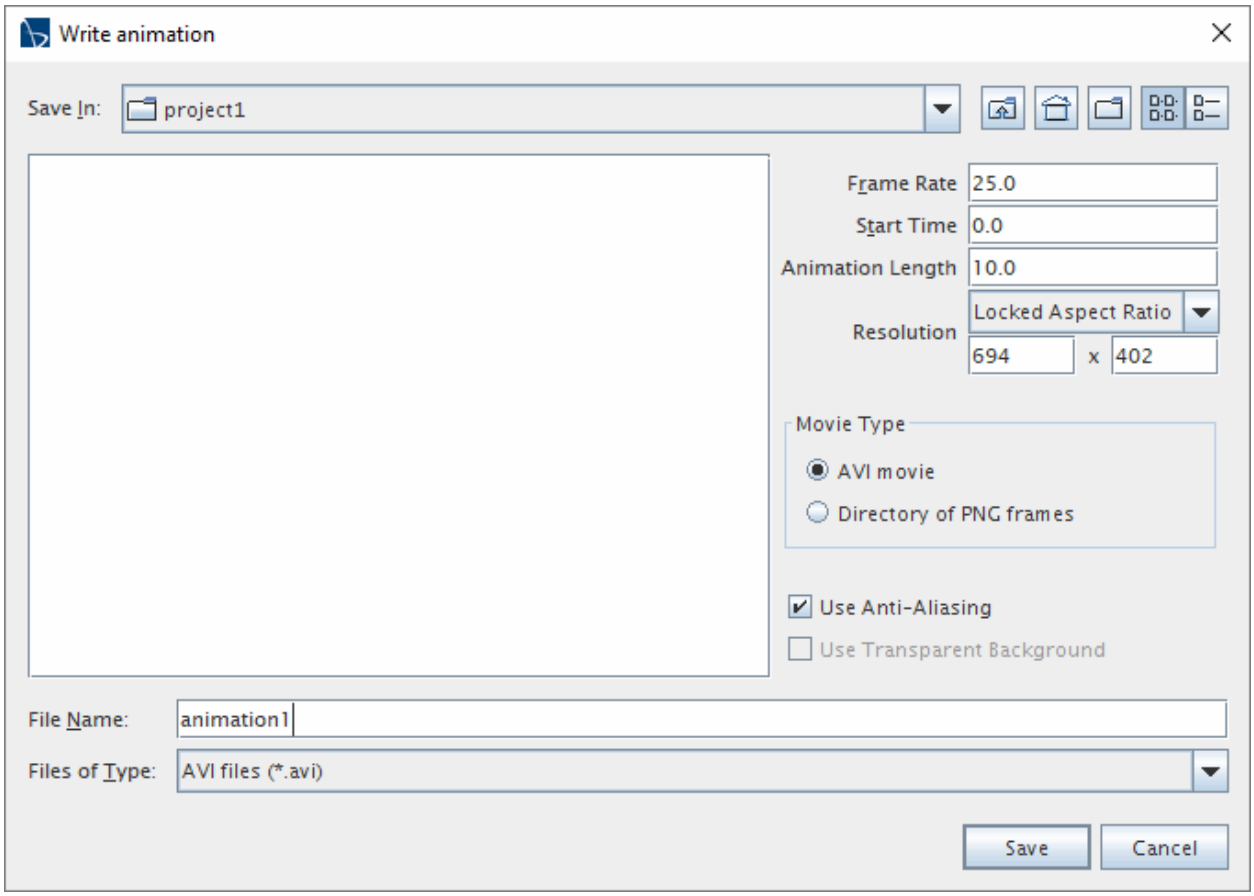
Additionally, the number of frames per second can be controlled in the scene attributes.

Recording the Animation onto Separate Files

It is possible to record a segment of the animation onto separate files: either a commonly used type of video file, `.avi`, or a set of frames in `.png` graphics format.

For best results, follow these steps:

1. Make sure that the particles appear and move exactly as you want them. Adjust all the properties and view the particles in motion.
2. Press the (**Write Movie**) toolbar button in the **Animation** toolbar.
3. In the *Write animation* dialog, prepare the specifications for the recording:



1. **Frame Rate** -- Decide on the number of frames per second (fps). More frames show smoother movement but take more time and computing power for recording and playback. Fewer frames allow the file to run more efficiently, but may show breaks in the movement. The default rate for animation in STAR-CCM+ is 25 fps. To try an alternate rate, change it in the properties of the **Animation** object in the scene attributes, and then preview the animation again as in Step 1.
2. **Start Time** -- Start the animation at a time other than t = 0.
3. **Animation Length** -- Decide on the time length of the animation.
4. **Resolution** -- To specify the movie size (x and y values in pixels) for viewing, use one of the two available options in the drop-down list:
   - **Locked Aspect Ratio** -- enter a value for the **X Resolution** or **Y Resolution**, and STAR-CCM+ computes the other one to preserve the aspect ratio.
   - **Custom Resolution** -- enter any value up to `4096` for either resolution and the aspect ratio is ignored. This choice can produce unexpected viewports and cutoffs of images.
5. *Movie Type* -- Select the method for recording the animation:
   - **AVI movie** -- This is the default setting.

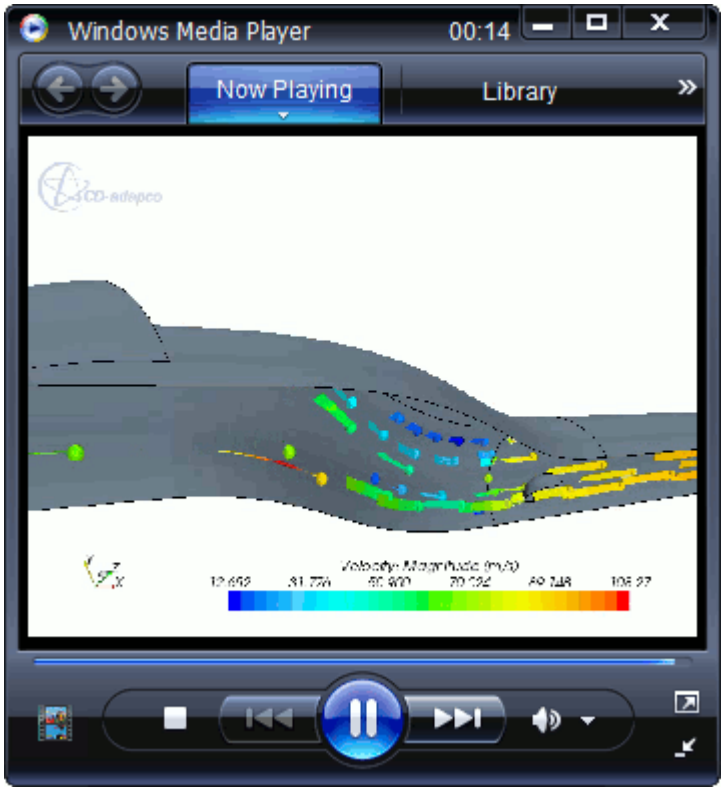
- **Directory of PNG Frames** -- Click this radio button if you prefer to record each individual frame as a graphic file. Type the directory path name into the **File Name** text box of the dialog. The frames are saved in that directory as the name `frameXXXXX.png` where `XXXXX` is the five-digit frame number (padded with zeros if necessary).

  Note: If you choose to create `.png` files, consider the previous specifications carefully, since you may create many graphics files. For example, the default settings would record 25 frames per second for 15 seconds, generating 375 `.png` files.

6. **Use Anti-Aliasing** -- When activated, this option reduces distortion in the rendered image, creating a smoother appearance on screen. While anti-aliasing increases rendering time, it also reduces the file size of exported `.avi` movies.

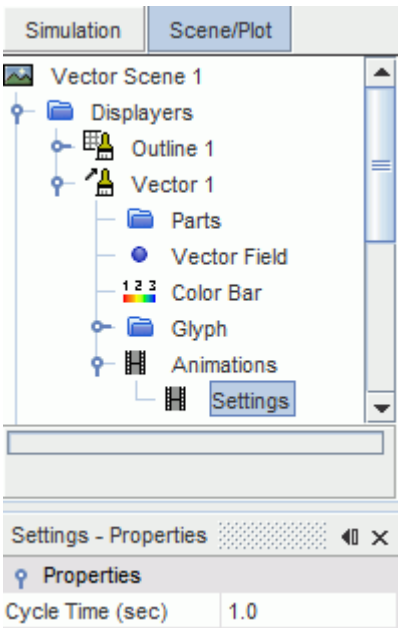
4. To generate an animation with a transparent background, activate the **Use Transparent Background** option. This option is available for exporting `.png` files only.
5. Navigate to the file location of your choice. For an `.avi` movie, enter a file name as you would for saving any other file.
6. Click **Save**. STAR-CCM+ generates the file(s).

   The `.png` files can then be animated with your media program of choice. The `.avi` file can be played in a typical Windows Media Player, or its equivalent in Linux.



Using Vector Glyphs

You can animate vectors by activating the **Animations** > **Settings** node.

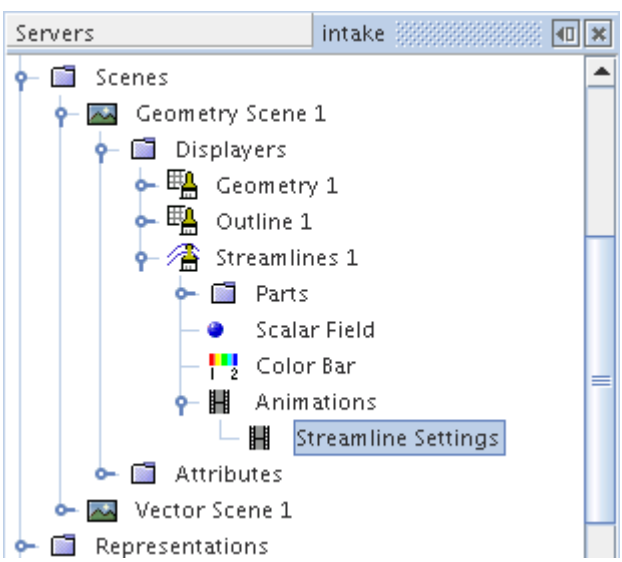


Animations Settings Properties

Cycle Time (sec)	Sets the total time for a glyph's travel from the inlet to the outlet.

Using Streamline Tracers

You can animate streamlines by activating the **Streamline Settings** node.

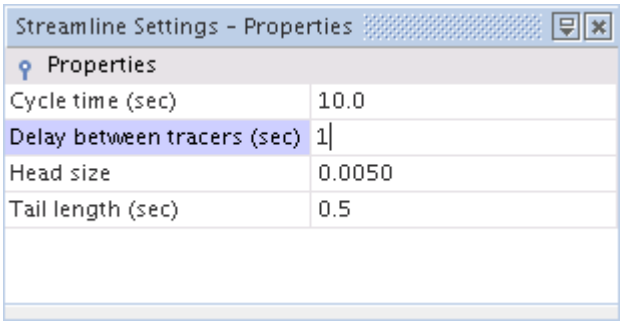


Streamline Settings Properties

Cycle time (sec)	Sets the total time for a tracer's travel from the inlet to the outlet.
Delay between tracers (sec)	Sets the time elapsed between repetitions of the appearance of a particular tracer.
Head size	Specifies the radius of the particle in the units that are used in the simulation.
Tail length (sec)	Defines the length of the tail in terms of how far the particle travels in the specified length of time.

Changing the Tracer Frequency

You can make the particles appear in a sparse or dense pattern.

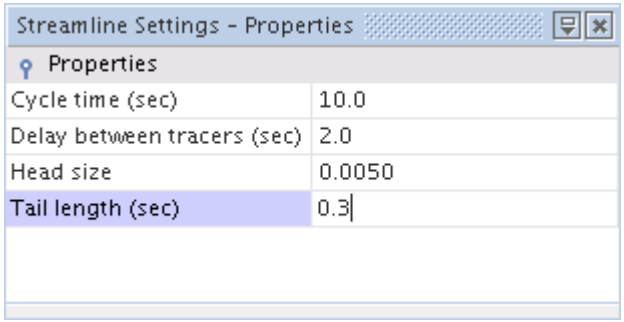


The *Delay between tracers* property specifies how soon the next set of tracers is injected into the flow.

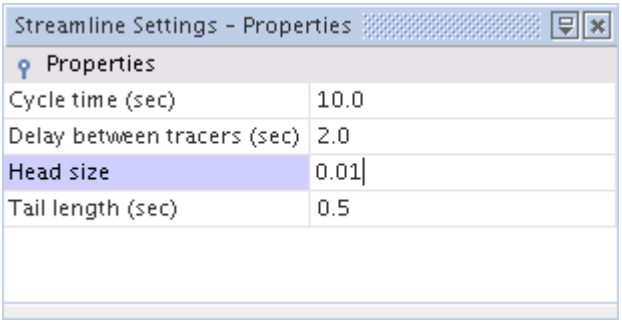
Changing the Tracer Dimensions

You can change the dimensions of the tracer.

- The *Tail length* property controls the length of the tail. Select the **Streamline Settings** node and enter a number of seconds.



  These seconds are the time taken for the particle head to travel a certain distance: the longer the time, the longer the distance, and hence the longer the tail length.
- The *Head size* property specifies the radius of the particle in the units that are used in the simulation.



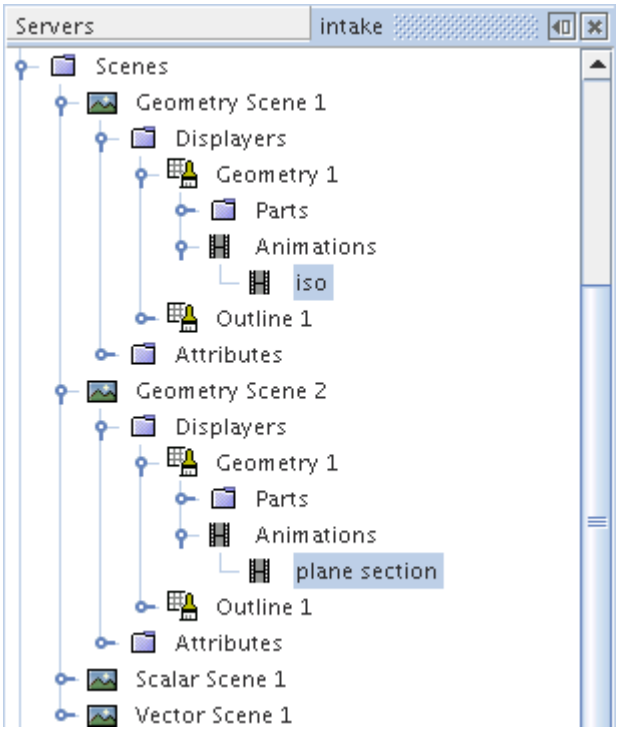
Sweeping Parts Animation

STAR-CCM+ can sweep isosurfaces or plane sections from a start point to an end point so that you can visualize the plotted quantity throughout the domain.

In the *Animation Mode* property of the **Animations** sub-node of a displayer, you have a choice of options for sweeping:

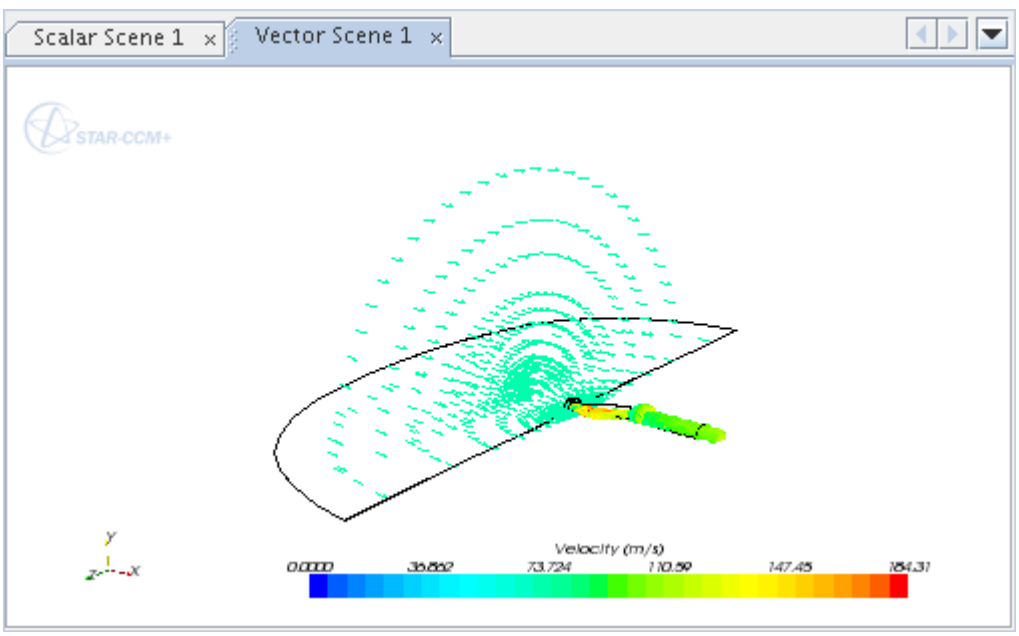
- When you choose **Continuous Sweep**, the sweep occurs steadily through the entire region. Therefore, when using this option with a simulation history (`.simh`) file, you must export to that file the region along with the scalar or vector function. (You would export a region to a `.simh` file *only* if you want continuous sweep.) This export generates a larger `.simh` file and costs more computing power.
- If you are animating a plane section with multiple sections, you can use the **Discrete Sweep** option, which only loops through each of the multiple sections of the plane section. When used for a representation from a `.simh` file, you only need export the multi-plane section to the `.simh` file along with the relevant field functions. This option requires less computing power than the continuous sweep.

The settings nodes for these parts are named according to the type of part, but they have identical properties.



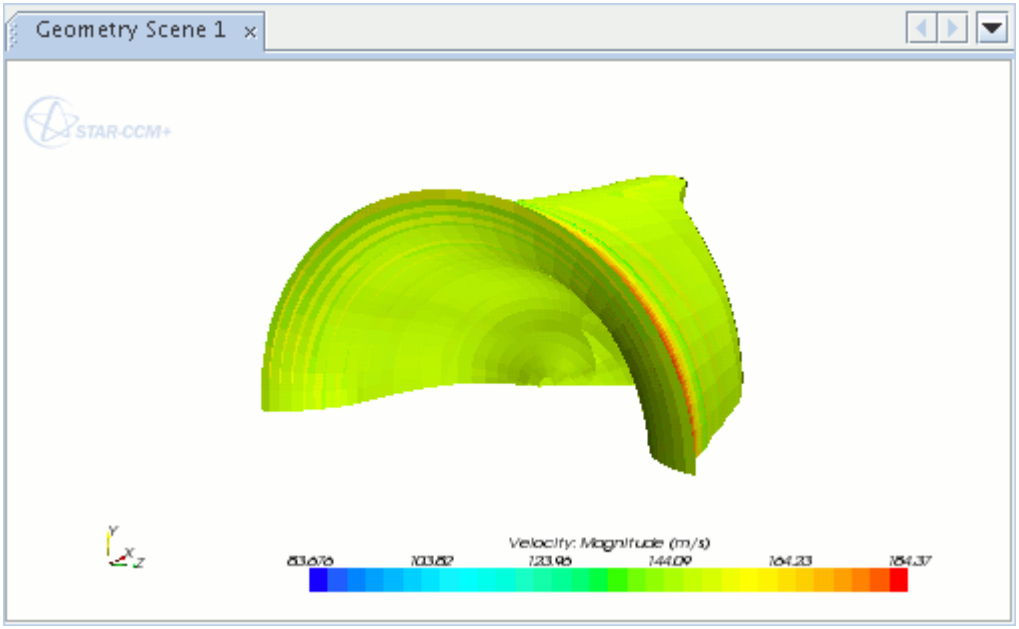
Each controls the animation for all the parts of that type in the displayer. If the displayer contains an isosurface and a plane section, the nodes that represent them can both exist at the same time in an **Animations** node.

The following screenshot shows an example of a plane section with vectors.



Additionally, you can animate the vector glyphs themselves (in a separate displayer) while they are being swept on the derived part.

This example shows an isosurface being used to sweep scalars.



You can choose between two types of animation cycles in the properties:

- Repeat -- this default setting causes the sweeping motion to go from the beginning to the end of the range, and then start again from the beginning.
- Bounce -- this setting specifies a sweep from beginning to end, then back to the beginning.

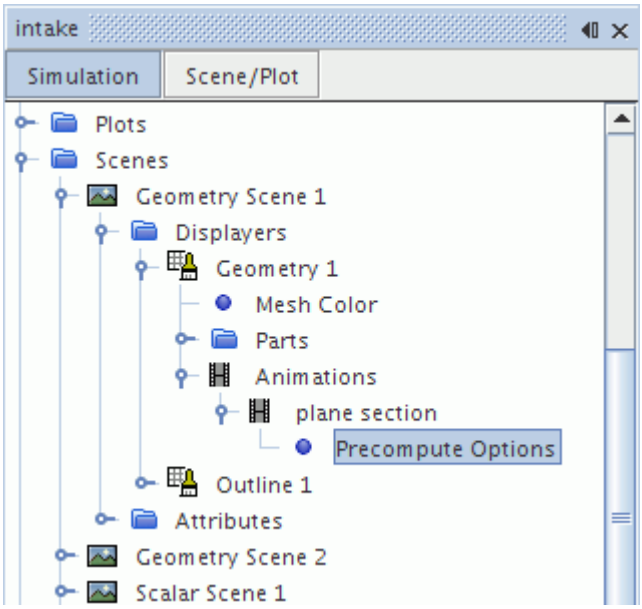
If you have multiple plane sections in a displayer, they must all have the same cycle time, but an isosurface in that displayer can have a different cycle time. To have different cycle times among plane sections, put them in different displayers.

To obtain consistent color mapping over the whole animation time, use a fixed scalar range.

Note: The animated derived parts require default units of STAR-CCM+ for the start and end values, not the preferred units of your simulation. For example, with a plane section, the units are always in meters. To animate with an isosurface, work with the default STAR-CCM+ units for the particular scalar that you have in that isosurface.

Using Precomputed Sweep Animation

You can animate sweeping derived parts with precomputed steps. By default, STAR-CCM+ must access the server during each animation step to calculate values and geometry for the derived part. When you activate precomputation, STAR-CCM+ prepares these calculations before you start the animation, allowing you to animate the derived part without constant server access. Precomputed steps improve the performance of the animation with a higher average frame rate, but the precomputation requires more time, and the animation requires more memory. The requirements for time and memory increase proportionally to the specified number of steps.



To prepare a sweep animation with precomputed steps:

1. Select the **Animations** node in the displayer node.
2. Make sure that *Animation Mode* is set to **Sweep**.
3. Expand the **Animations** node and select the sub-node that corresponds to the derived part (plane section or isosurface).
4. Activate the *Precompute Steps* property.

   In the **Precompute Options** sub-node that appears, specify the number of precomputed steps in the *Number of Steps* property.

   Note: While increasing *Number of Steps* generates a smoother motion in the animation, it also increases memory consumption.

Sweep Settings Properties

Cycle time (sec)	Defines how many seconds it should take to sweep over the whole configured range.	
Auto Range	Determines whether to set the range of values to display automatically.	
	Activated	Automatically determines the range of values. This range begins with the starting offset for sections and the lowest iso value for isosurfaces. It ends with the ending offset for sections and the highest iso value for isosurfaces.

	Deactivated	Does not use auto ranging, rather uses the values that are set in *Min* and *Max*.
Precompute Steps	Controls whether to use precomputed steps for the sweep animation.	
	Activated	A user-defined number of steps is precomputed over the configured sweep range. A suitable precomputed step is chosen and rendered while the sweep animation is running in the scene display, resulting in an increased average frame rate.
	Deactivated	All sweeping parts are continuously updated while the sweep animation is running. The playback performance within a scene display is limited by the computation performance of sweeping derived parts.
Start Value	The starting offset for sections and the lowest iso value for isosurfaces. See Single Section.	
End Value	The ending offset for sections and the highest iso value for isosurfaces. See Single Section.	
Repeat Mode	Controls how the derived part moves once a sweep is finished.	
	Repeat	Sweeps from the beginning to the end of the range, and then starts over from the beginning (default). One cycle is from beginning to end.
	Bounce	Sweeps from beginning to end, then back to the beginning. One cycle is from beginning to end to beginning, so a bouncing derived part moves twice as fast as a repeating one.

Animations Properties

These properties vary depending on the type of displayer.

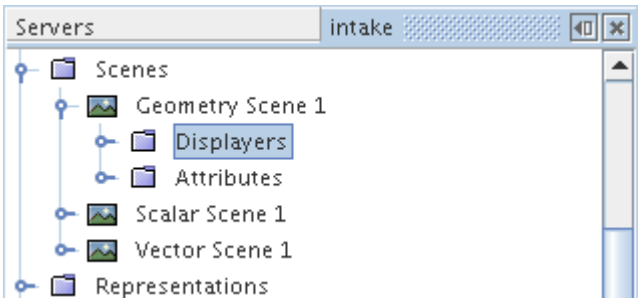
Animation Mode	Controls the animation feature for vectors and streamlines.	
	Pulse	Makes glyph animation available. Adds a Settings subnode to the

		Animations node (vector displayer only). See Using Vector Glyphs.
	Tracers	Makes tracer animation available. Adds a Streamline Settings subnode to the **Animations** node (streamline displayer only). See Using Streamline Tracers.
	Continuous Sweep	Makes sweep animation available. Adds an **iso** and/or **plane section** sub-node to the **Animations** node, depending on whether the displayer contains a derived part of the respective type. See Sweeping Parts Animation. When using this approach with a simulation history (`.simh`) file, you must export to that file the region along with the scalar or vector function. This export can generate a larger `.simh` file and cost more computing power.
	Discrete Sweep	This option activates a **plane section** sub-node to the **Animations** node. It only loops through each of the multiple sections of the plane section. With this option, you can work with a smaller `.simh` file: you *only* need to export the multi-plane section along with the scalar or vector functions. This option requires less computing power than the continuous sweep. **Note:** This option is only available for use with a plane section with multiple sections.
	Off	Leaves animation feature off.

Using Displayers

This section defines what a displayer is and how to use displayers to build scenes in STAR-CCM+.

The **Displayers** node, which has a pop-up menu, is the manager object for all displayers in a scene.



It is included automatically in every scene. The displayer manager can hold any number of displayers up to the memory limits of the computer.

Contents:

What Is a Displayer?
Displaying Cell Values in Scalar Scenes
Managing Parts in Displayers
Working with Parts-Based Interfaces in Displayers
Changing the Color of the Surface
Using the Color Bar
Using a Combination of Displayers
Lighting Displayer Surfaces
Hiding and Showing All Parts in a Displayer
Making Displayer Surfaces Transparent
Smooth Shading Displayer Surfaces
Culling Polygons
Copying Displayers
Pop-Up Menu of the Displayers Manager

What Is a Displayer?

A displayer is the basic building block used to build a scene. It provides a flexible way of defining and controlling the graphics entities appearing in a scene.

There are four types of displayers available in STAR-CCM+, and they share a common pop-up menu:

- A part displayer to control the appearance of geometry
- A scalar displayer to control the appearance of scalar contours

- A vector displayer to control the appearance of vector glyphs
- A streamline displayer to control the appearance of streamlines

A displayer includes one or more parts, an attribute such as a scalar or vector field, and other properties such as surface lighting, color, opacity, normals for smooth-shading, and transforms. A displayer takes these parts and attributes and produces the appropriate graphics entities to represent the parts in the scene.

A displayer provides a convenient way to control the appearance of many parts with the same attributes. A displayer also provides a way to view the same part in many different ways in the same scene. For example, consider viewing the scalar field Mach number on a wing and fuselage. The displayer manages the wing and fuselage parts, and produces an image of the Mach number contours on those parts. If you want now to see pressure on those parts, it is a simple matter of changing the scalar field of the displayer to pressure. Next, consider viewing the wing surface with velocity vectors. In this case, you can create two displayers, one a geometry displayer showing the wing surface, the other a vector displayer showing the wing velocity vectors.

Displayers can be transformed and repeated, allowing for any user-defined or solver-defined transform to be applied to the parts within the displayer. This allows for the parts in the scene to be repeated or moved from their default location within a given scene.

A displayer can be renamed or deleted using the standard techniques. When a displayer is deleted, all part representations controlled by that displayer are removed from the scene as well.

Contents:

Displayer Pop-Up Menu

Displayer Pop-Up Menu

Use the pop-up menu of the displayer to toggle its visibility.

Toggle Visibility	Renders all the parts in the displayer visible or invisible with a single command.

See also Common Pop-Up Menu Items.

Displaying Cell Values in Scalar Scenes

To view the cell values for a region on one side of the interface only, you must create a threshold derived part and use this in the scalar displayer.

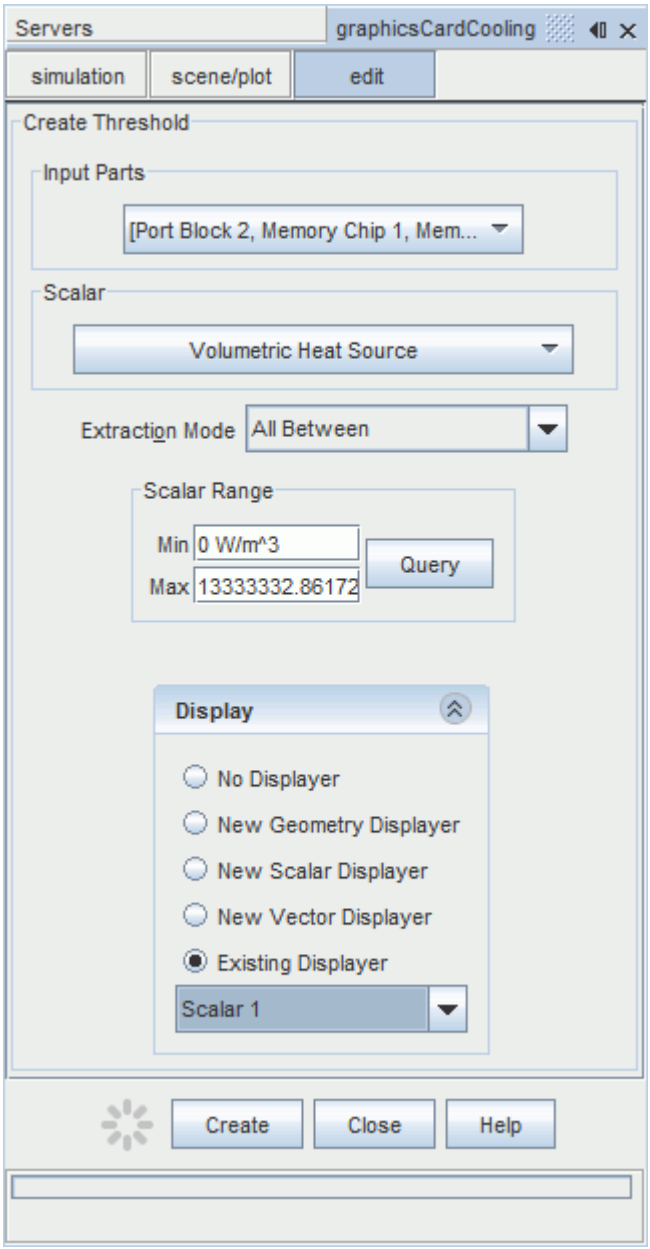
When a case contains multiple regions, such as in a Conjugate Heat Transfer (CHT) analysis, a scalar scene will show averaged values for boundaries that belong to interfaces between regions. STAR-CCM+ calculates the averaged values based on solution data from both sides of the interface.

The recommended steps to follow are:

- Create a threshold derived part. The Create Threshold in-place dialog appears.
- For the *Input Parts* property, select the parts that you want to view the cell values on.
- For the *Scalar* property, select the field function that you want to view on the cells.
- Ensure that the *Extraction Mode* is set to **All Between**.

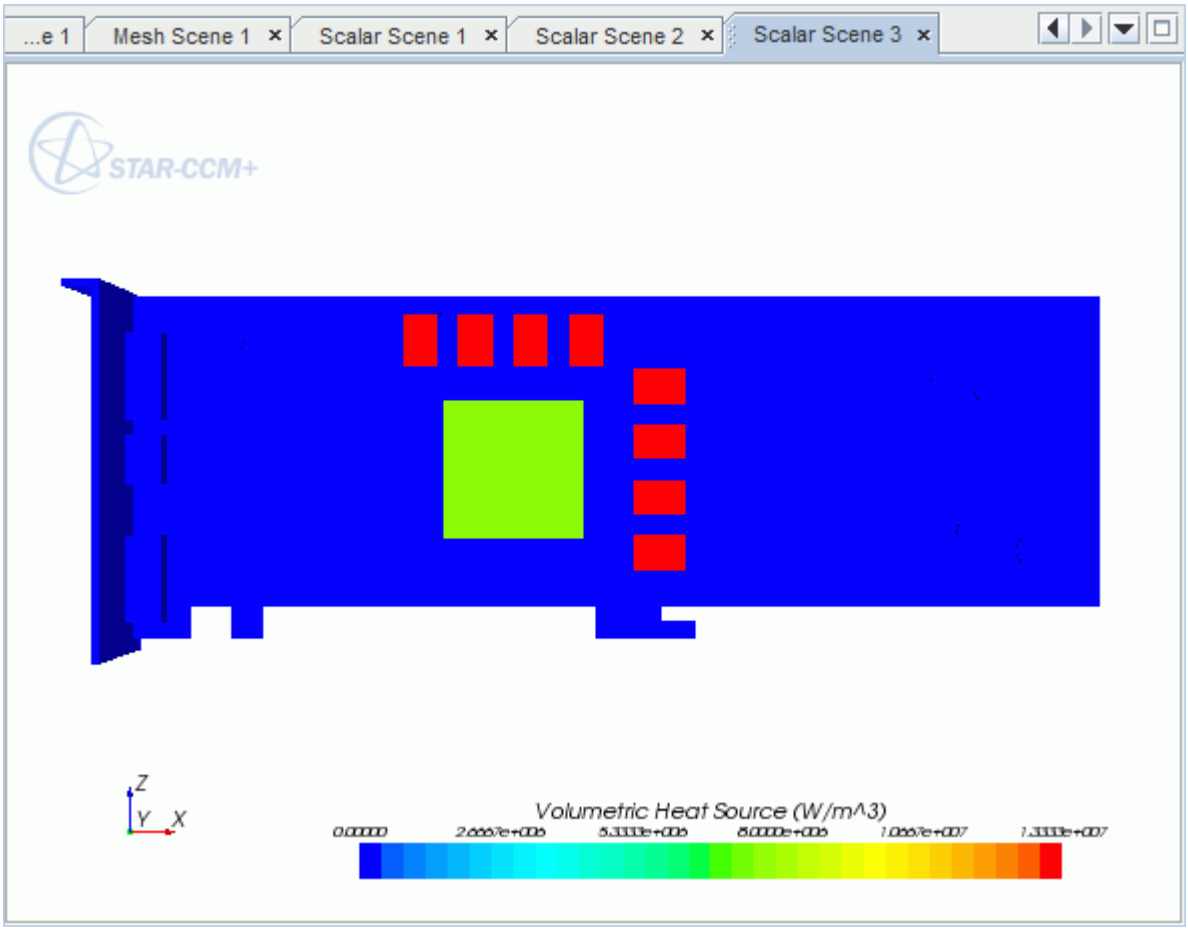
- Click the **Query** button to populate the *Min* and *Max* values.

A completed dialog is shown below.



- Create the threshold.
- Select the threshold part for the scalar displayer in the active scene.
- In the scalar scene, apply the field function that you want to view.

An example scene that displays the cell values is shown below.

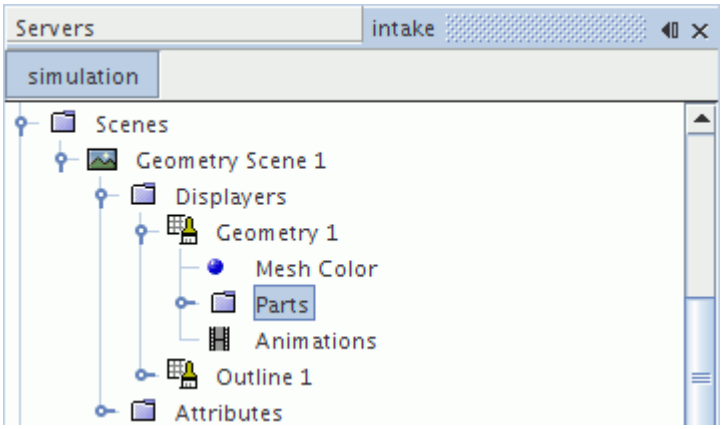


In this case, the eight memory chips on the graphics card have the same volumetric heat source, while the GPU chip has a different value. The remaining parts have no volumetric heat source.

Managing Parts in Displayers

The **Parts** manager object contains all the parts in a scene displayer.

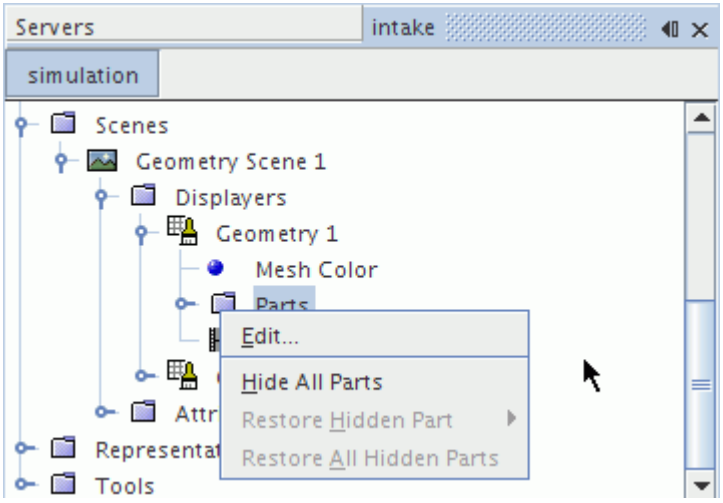
The parts are represented as a distinct group of part nodes in the object tree. The **Parts** node has its own properties and pop-up menu.



This section explains other techniques for adding parts to a scene and removing them.

Hiding and Showing Parts with the Displayer Parts Manager

The pop-up menu of the **Parts** manager node includes actions for hiding and showing parts in a scene.



Contents:

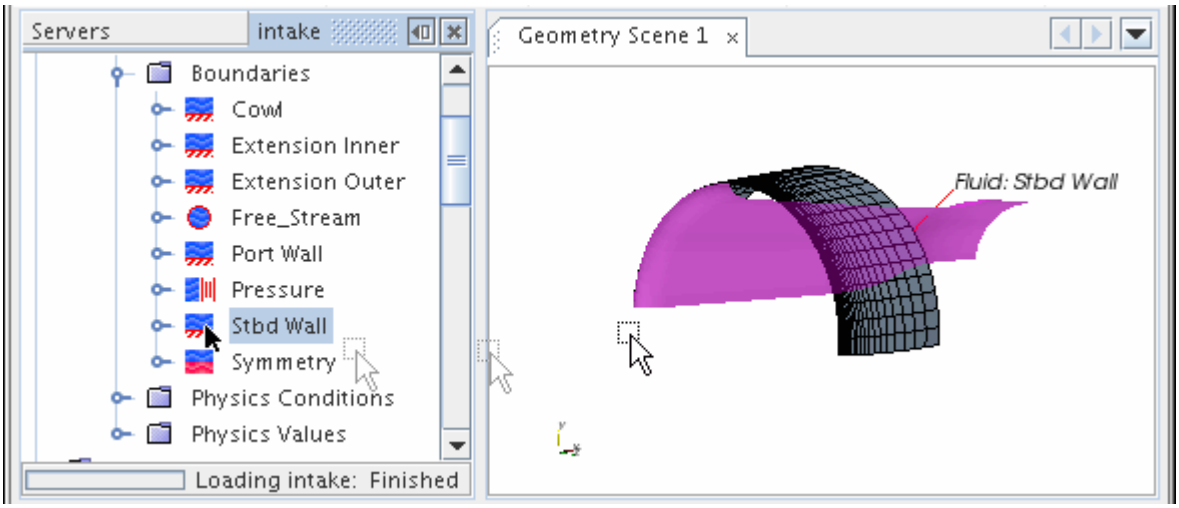
Adding Parts to Displayers
Removing Parts from Displayers
Working with Displayer Part Nodes
Properties of the Parts Manager
Pop-Up Menu of the Parts Manager

Adding Parts to Displayers

This section explains two techniques for adding parts: drag-and-drop and the object selector.

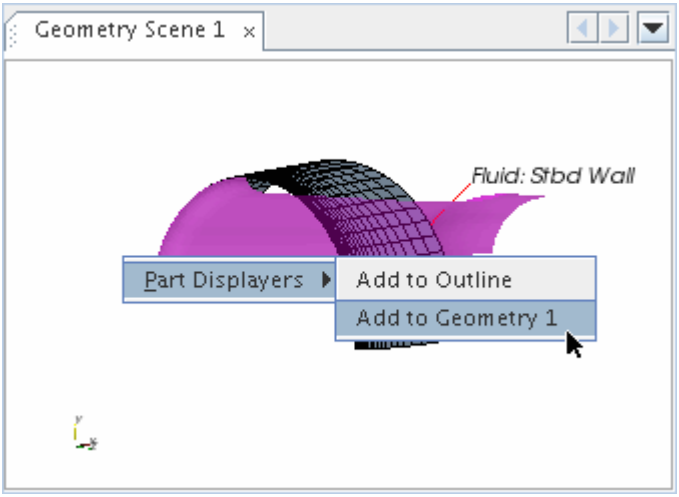
Adding Parts Using Drag-and-Drop

You can drag-and-drop a region, boundary, or derived part into a scene, even if that scene is empty (no displayers). To do this, select a part from the **Regions** and **Boundaries** nodes and/or the **Derived Parts** node. Drag the selected part into the visualization display.

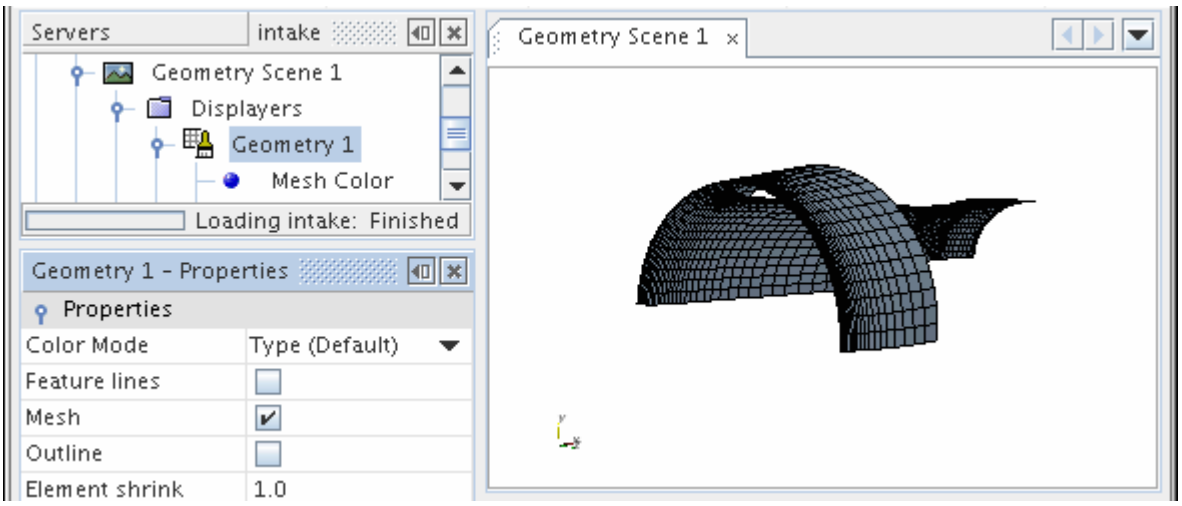


Once the part has been dragged to the visualization display, STAR-CCM+ responds depending on the number of displayers in the scene:

- If there are no displayers, a geometry displayer showing surfaces is created automatically by default, and the part is added to it.
- If there is only one displayer, the part is added automatically to that displayer.
- If there are more than one displayer, a pop-up menu lets you choose which displayer should receive the part.



After the part has been dropped into the visualization display, it has the properties that are prescribed by the displayer. In the example below, the displayer has the mesh display enabled so the mesh is displayed on the new part.

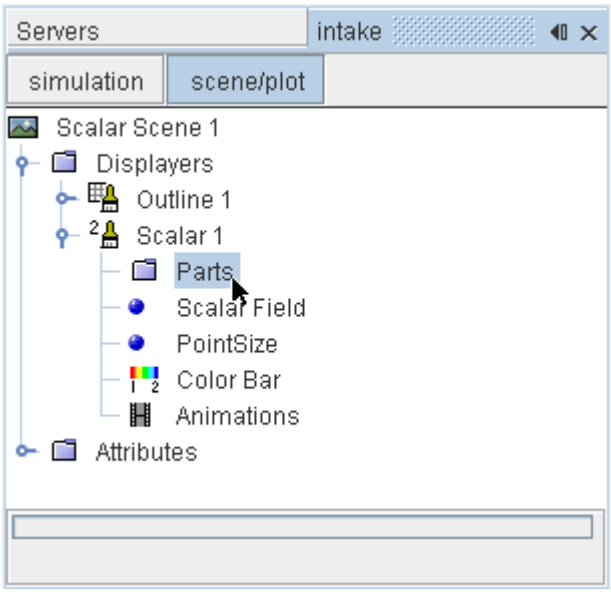


If a region part is added to a displayer, all of the boundaries of the region are also added to the displayer. Multiple parts can be added at the same time by multi-selecting nodes.

Alternatively parts can be dragged into the displayer node.

Adding Parts Using the Object Selector

Parts may also be added to the displayer using the object selector. Expand the **Scenes** node, the node of the chosen scene, the **Displayers** node and the displayer node to which you wish to add the part. Select the **Parts** node.



Open the object selector dialog by clicking the right column of the *Parts* property in the *Properties* window.

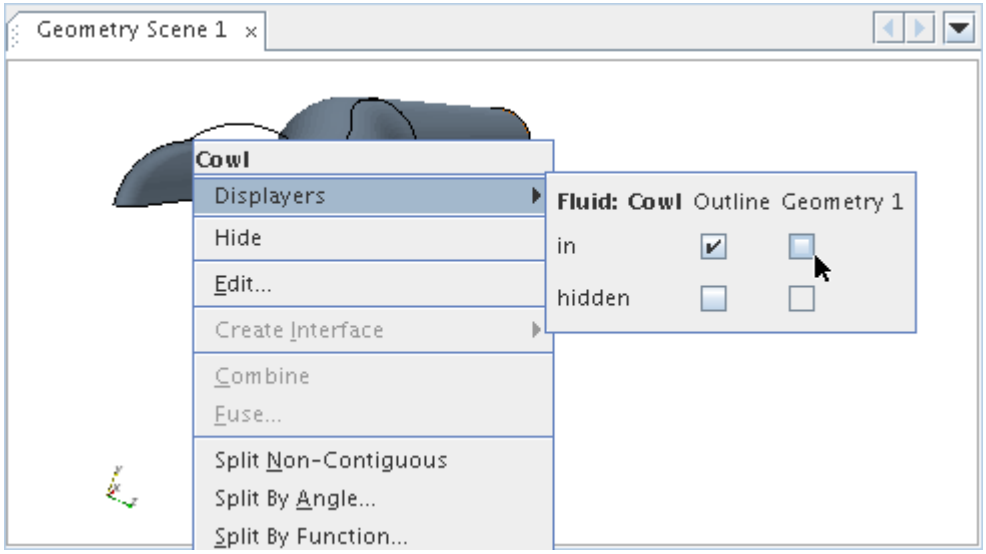
Click **Close** when you have finished selecting parts.

Removing Parts from Displayers

Parts can be removed from a displayer with one of multiple techniques.

The available techniques are:

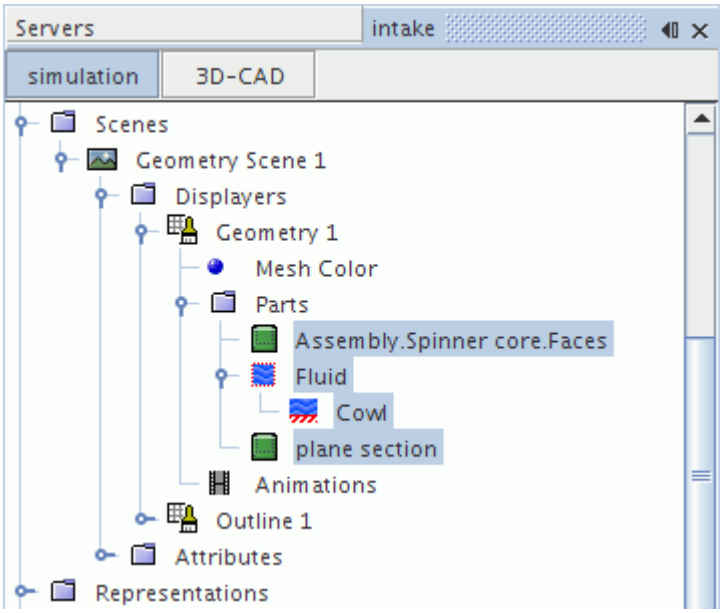
- Using the same part selector that is used for adding parts to the displayer
- Right-clicking a displayer part node
- Directly from the visualization display - right-click the part in the display, select **Displayers** from the pop-up menu to activate a set of checkboxes, and deactivate the **in** checkbox that corresponds to the displayer from which you want to remove the part.



Working with Displayer Part Nodes

The Parts manager node of a displayer contains subnodes resembling those of regions, boundaries, derived parts and/or geometry parts.

These subnodes, which have their own properties and pop-up menu, are a visual representation of the parts in the displayer.

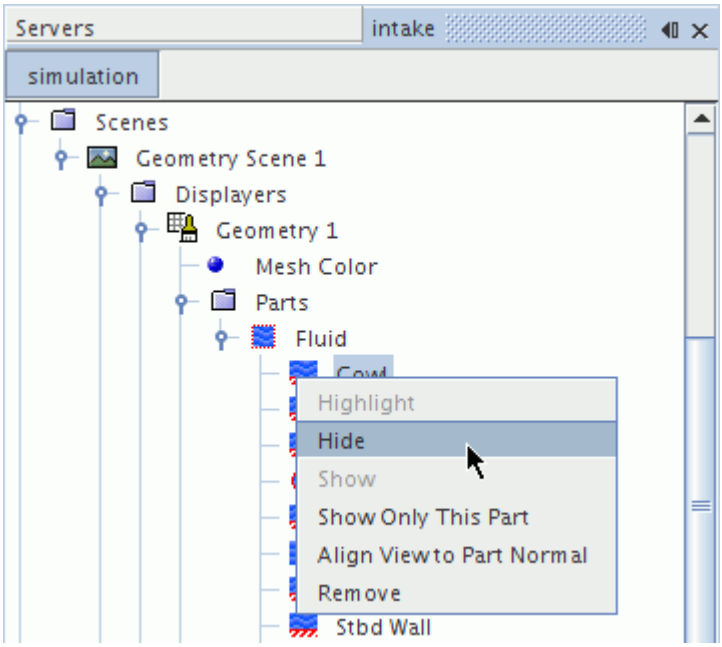


When you select parts in the **Parts** sub-node of a **Displayer** node, those parts get highlighted in the scene that is displayed in the *Graphics* window. When the displayer is associated with a transform (see Using Transforms), the selected part gets highlighted using this transform. Selecting the same part in multiple displayer nodes results in multiple highlights if the transforms of those displayers are different.

Hiding and Showing Parts

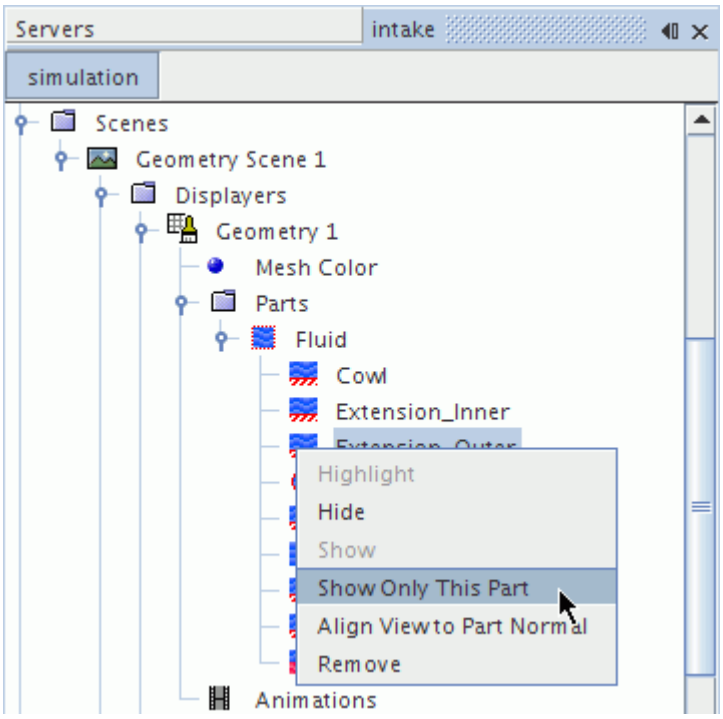
You can use these nodes to control the visibility in the scene of the parts they represent.

To hide a part from view, either deactivate the checkbox in its *Visible* property, or right-click its node and select **Hide** from the pop-up menu.



To reverse this step, activate the checkbox in the node's *Visible* property or select **Show** from the pop-up menu.

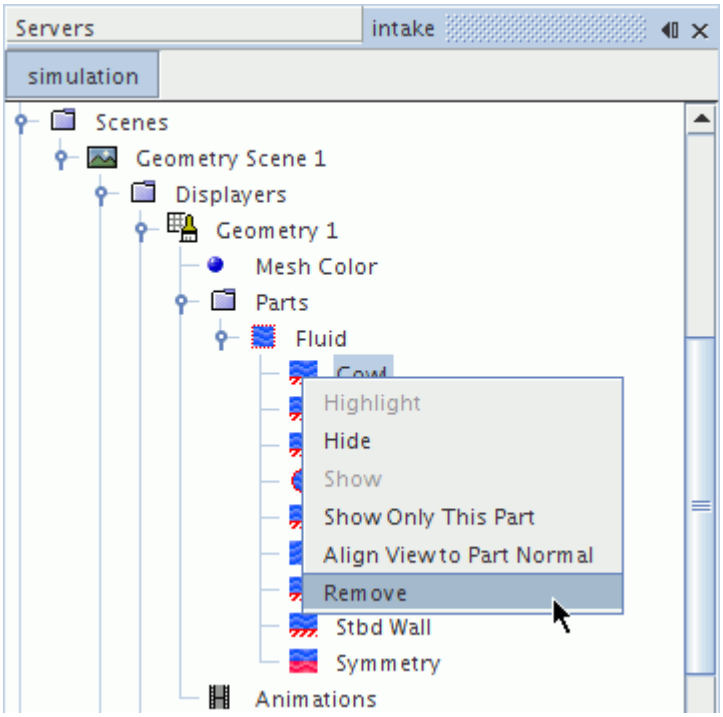
To make only one or more parts visible, right-click the corresponding node(s) and select **Show Only This Part** from the pop-up menu. This has the same effect as selecting all the other parts and hiding them.



When you hide or show an entire displayer, STAR-CCM+ remembers the settings you applied with these actions to the individual parts.

Removing Parts

To remove a part from the displayer completely, right-click its node and select **Remove** from the pop-up menu.



Displayer Part Properties

Visible	Shows or hides the part in the scene.	
	Activated	The part is visible.
	Deactivated	The part is hidden.

Displayer Part Pop-Up Menu

Highlight	Highlights the part in the display by changing its color and labeling it.
Hide	Executes the hide part action.
Show	Executes the restore hidden part action to redisplay a part that was hidden by the hide part action.
Show Only This Part	Isolates one or more selected parts by executing the hide part action for all the other parts in the displayer.
Align View to Part Normal	Aligns the view to the surface normal of a part. See Aligning to a Normal.
Remove	Executes the remove part from displayer action to remove the selected part from the relevant displayer's **Parts** node.

Properties of the Parts Manager

Use the properties of the Parts manager to view and control which parts appear in the displayer.

Parts	List of parts in this displayer. Editing the property accesses in-place object selector to add or remove parts. See Activating the Object Selection Dialog.

Pop-Up Menu of the Parts Manager

Use the pop-up menu of the Parts manager to hide and restore parts in the scene.

Hide All Parts	Executes the hide all parts action to make all parts that are controlled by that displayer invisible in the scene. See Hiding All Parts in a Scene.
Restore Hidden Part	Executes the restore hidden part action to redisplay a part that is hidden by the hide part action. You choose which part to restore from a submenu of this menu item. See Working with Displayer Part Nodes.
Restore All Hidden Parts	Executes the restore hidden parts action to redisplay all parts that are hidden by previous hide part actions. See Working with Displayer Part Nodes.

See also: Common Pop-Up Menu Items.

Working with Parts-Based Interfaces in Displayers

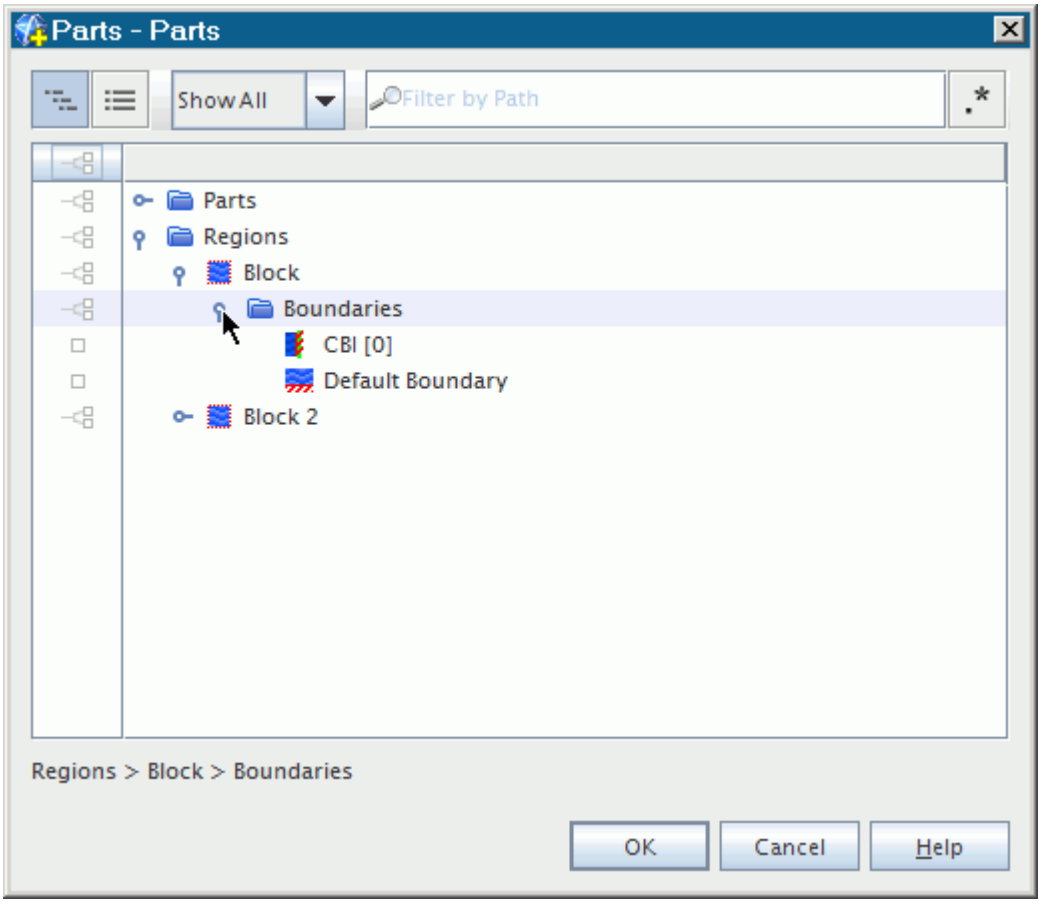
You can display *parts-based interfaces* -- the areas where the surfaces of parts come into contact with each other, or the remainders of those surfaces. Using parts-based interfaces improves performance for simulations that have many parts.

Note: To manage parts-based interfaces effectively, consider naming surfaces as you prepare the geometry of the simulation. Named surfaces facilitate query-based selection.

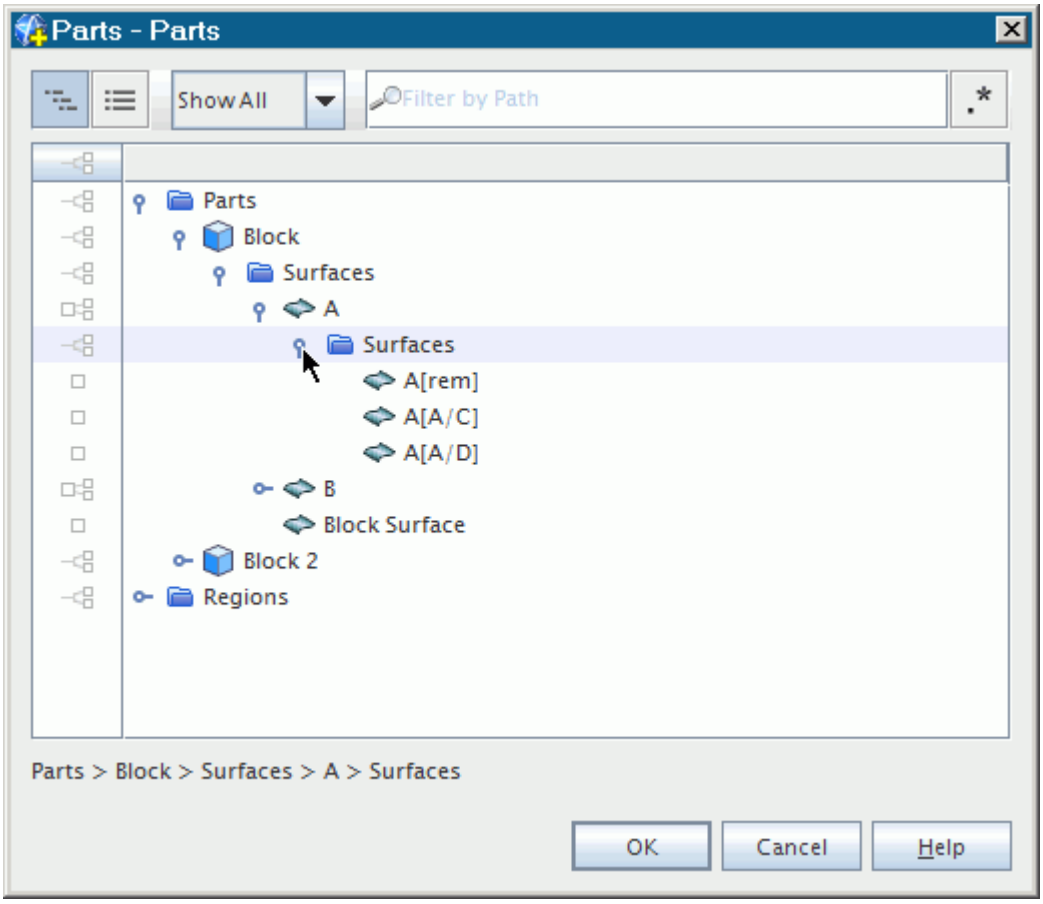
You can visualize parts-based interfaces in one of two ways:

- Displaying the whole interface boundaries (the complete intersected portion of the original parent boundaries)
- Displaying individually the intersected and remaining portions of part surfaces in the contacts of the interface

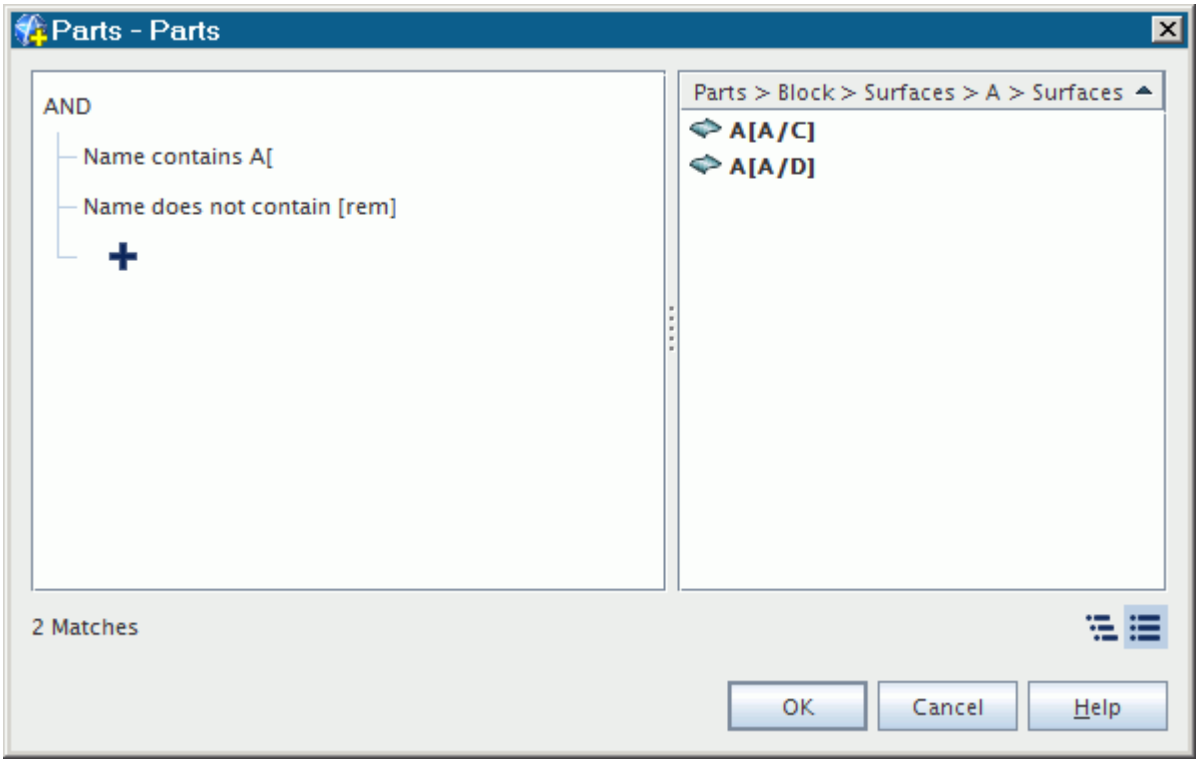
To view a whole interface boundary (using the object selector), select the respective interface boundary for the associated region in the **Regions** node. This functionality works the same as for boundary-based interfaces.



To view the intersected and remaining portions of part surfaces that are in contact, expand the nodes of the intersected parent part surfaces and select the desired subsurfaces. A subsurface exists for: (a) each contact in which the parent part surface participates, and (b) the remaining portion of the parent part surface that is not intersected.



To select parts-based interfaces with a query, specify the appropriate information from the names of the interfaces or the part surfaces. See Using Query-Based Selection.



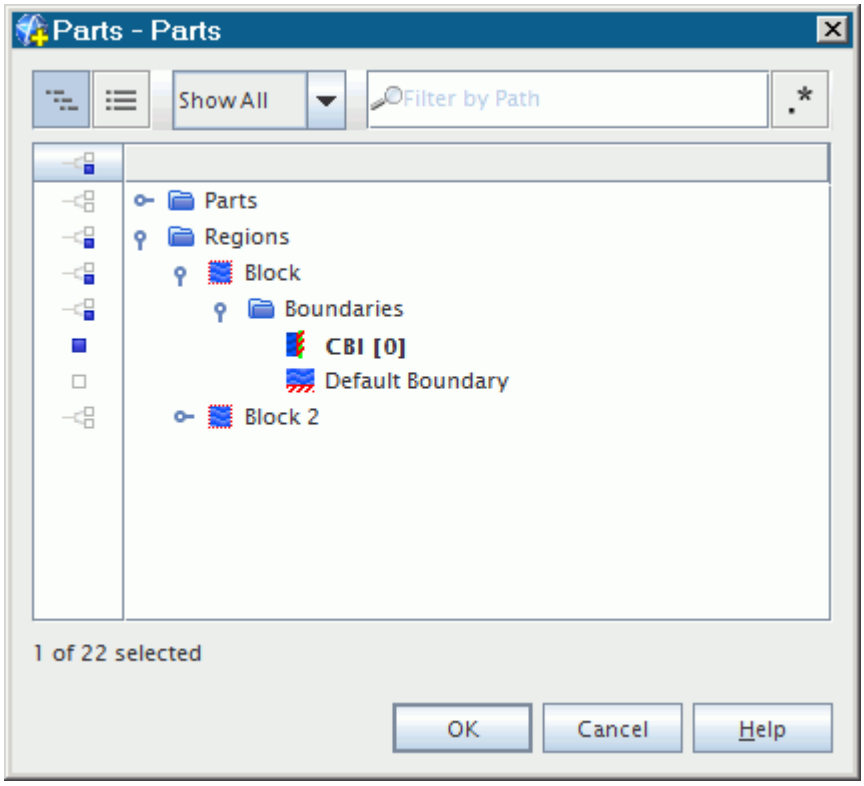
Example of Displaying Parts-Based Interfaces

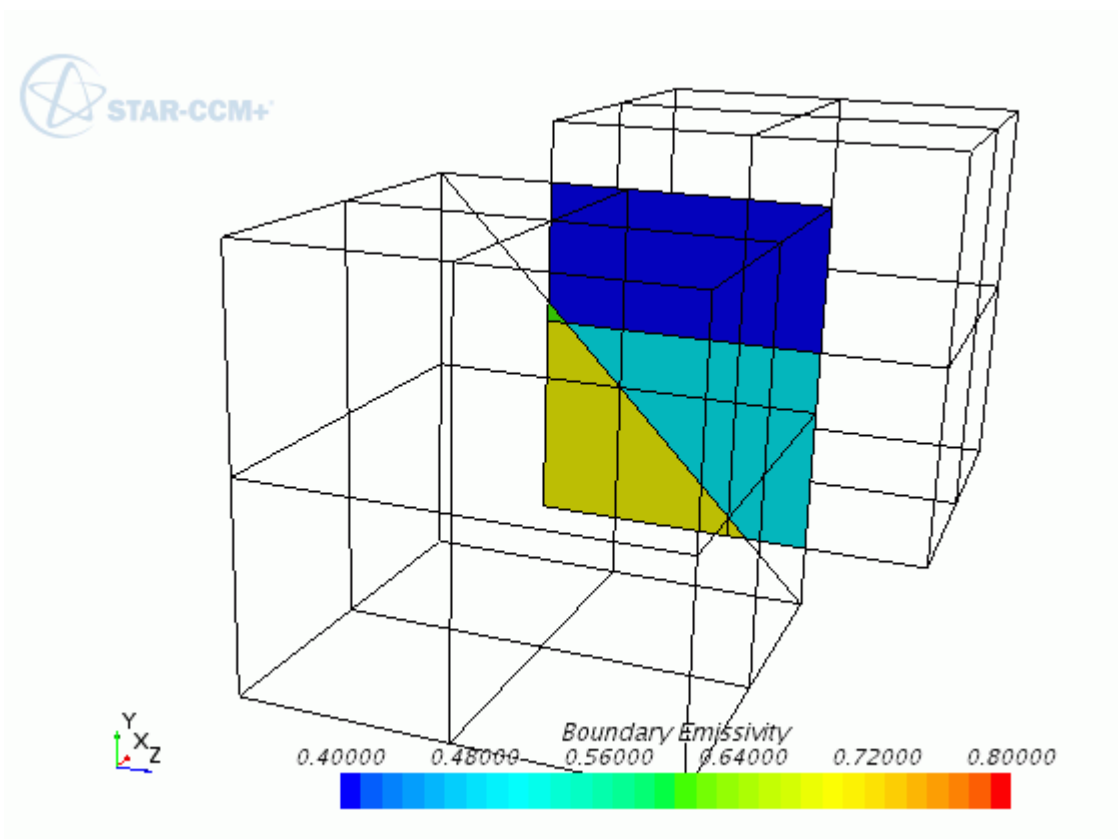
This example uses the intersection of two blocks to illustrate how you can visualize contact between surfaces. One block has surfaces A and B, the other has surfaces C and D. Each part surface participates in two contacts (one with each of the two part surfaces on the other side), yielding four contacts in the interface: A/C, A/D, B/C, B/D. For easier viewing, each part surface was assigned a different value of emissivity:

Part Surfaces	Emissivity Value
Contact A/C	0.4
Contact A/D	0.5
Contact B/C	0.6
Contact B/D	0.7
Remainders of Part Surfaces A and B	0.8

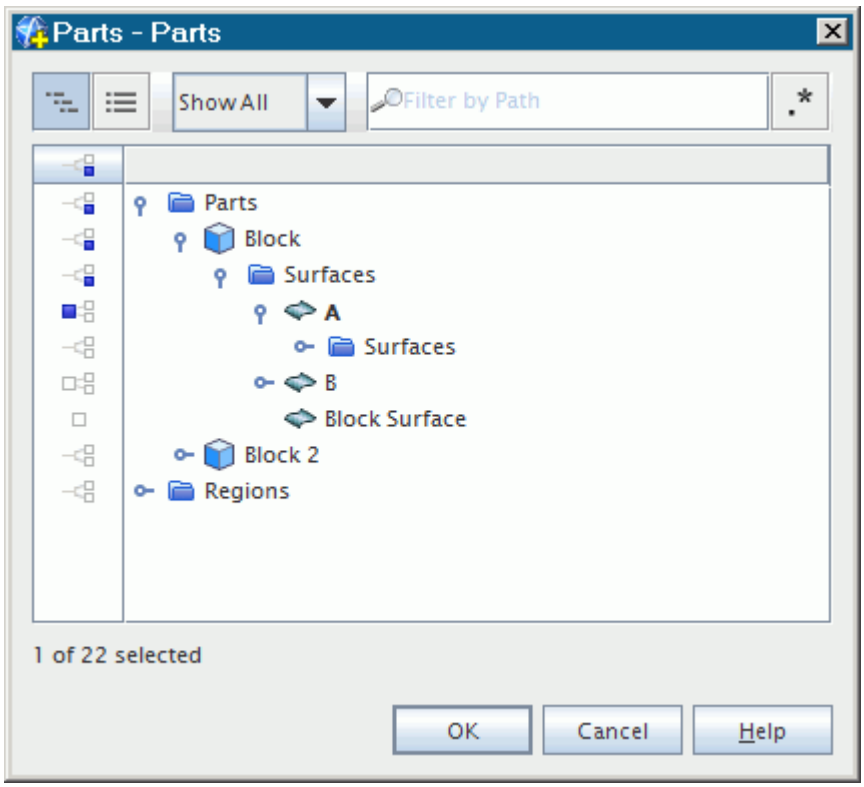
Four scalar scenes follow, each with its part selection and a display of the Boundary Emissivity field function:

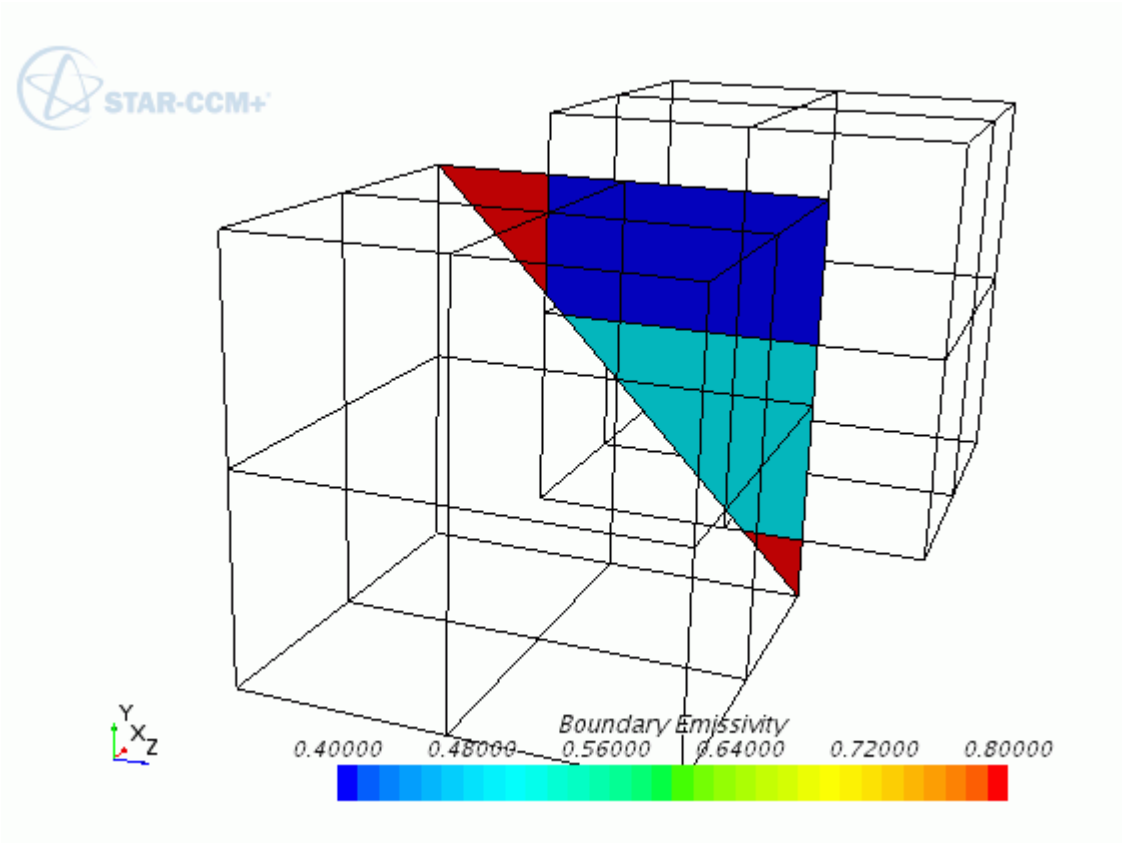
1. The interface boundary **CBI [0]**, on the side of the interface that is part of the region **Block** -- this scene shows the overall area of contact between the blocks.



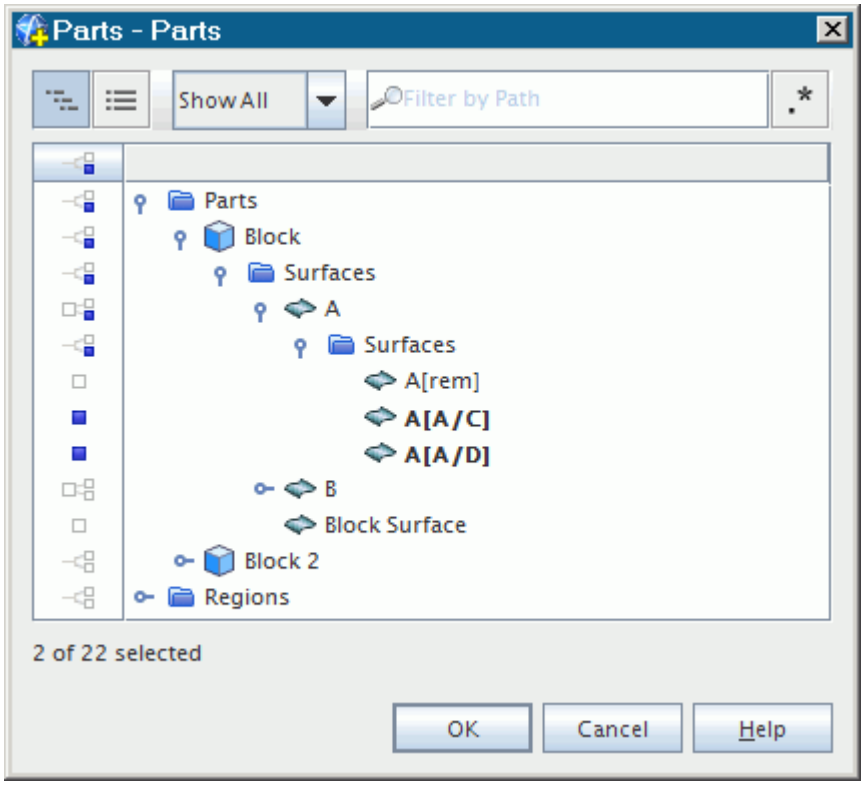


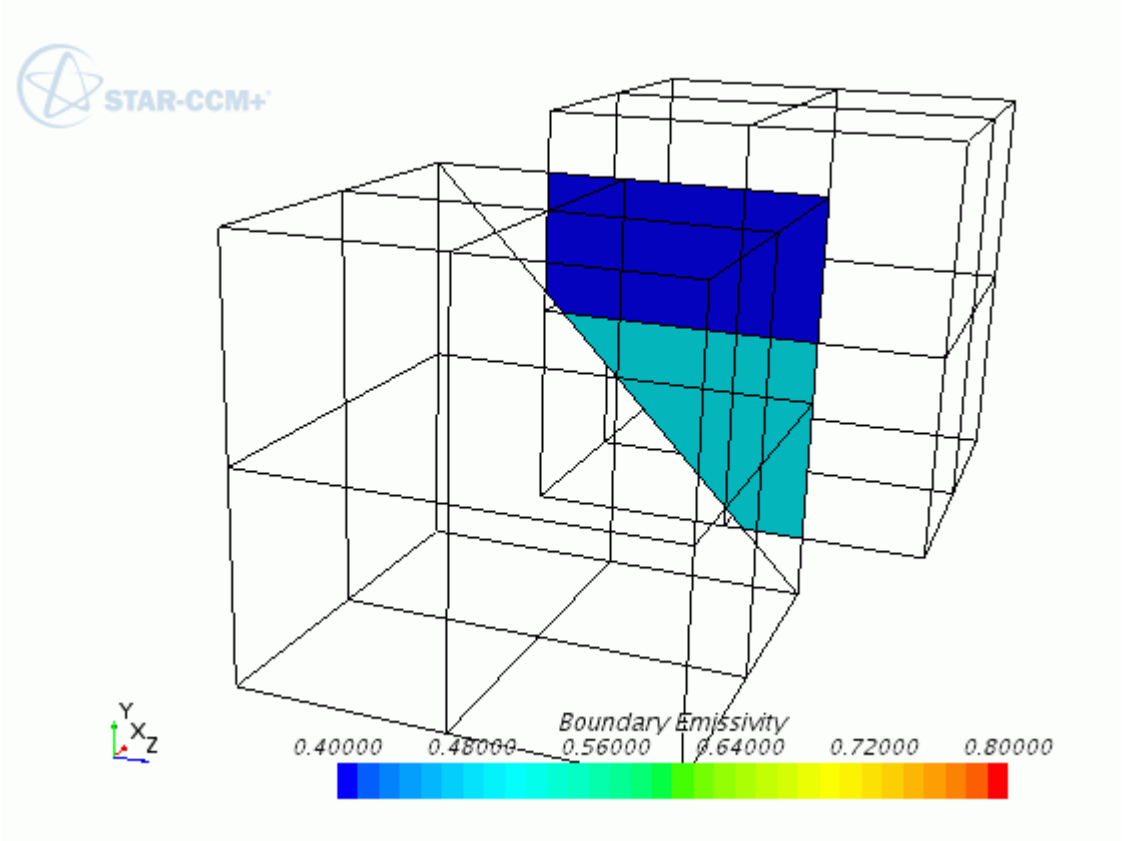
2. The entire Part Surface **A**:


Parts - Parts
Show All
Filter by Path
Parts
Block
Surfaces
A
Surfaces
B
Block Surface
Block 2
Regions
1 of 22 selected
OK
Cancel
Help

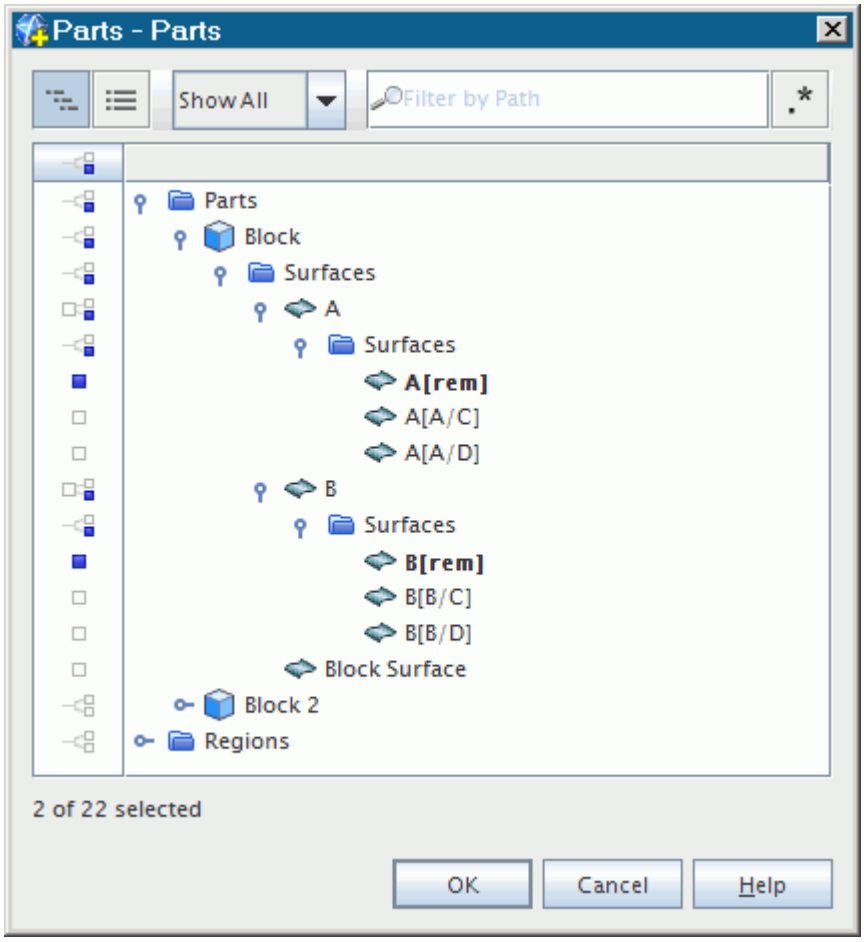

STAR-CCM+
Boundary Emissivity
0.40000
0.48000
0.56000
0.64000
0.72000
0.80000
Y
X
Z

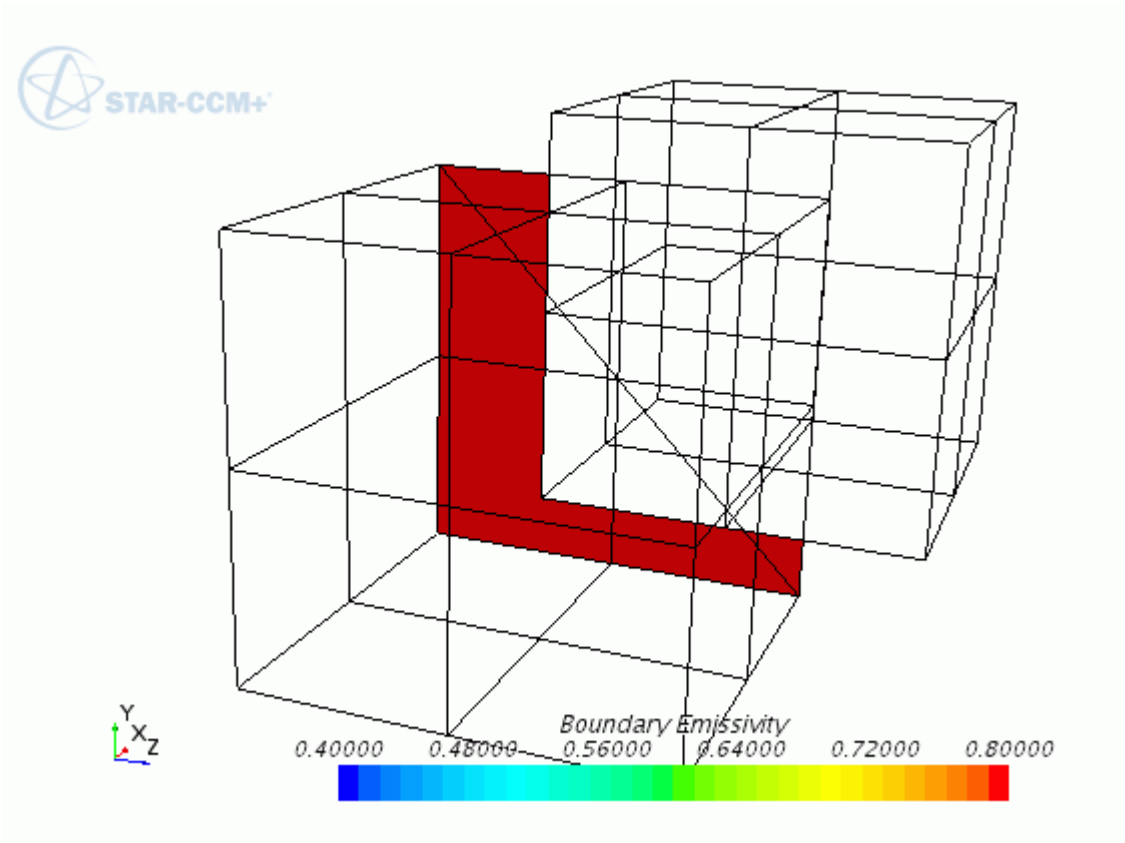
3. The contact surfaces **A[A/D]** and **A[A/C]**, where:
   - **A[A/D]** is the area of contact between Part Surface **A** and Part Surface **D**.
   - **A[A/C]** is the area of contact between Part Surface **A** and Part Surface **C**.





4. The remainders of Part Surfaces **A** and **B**, that is, the portions of the part surfaces that do not come into contact with other part surfaces:


Parts - Parts
Show All
Filter by Path
Parts
Block
Surfaces
A
Surfaces
A[rem]
A[A/C]
A[A/D]
B
Surfaces
B[rem]
B[B/C]
B[B/D]
Block Surface
Block 2
Regions
2 of 22 selected
OK
Cancel
Help



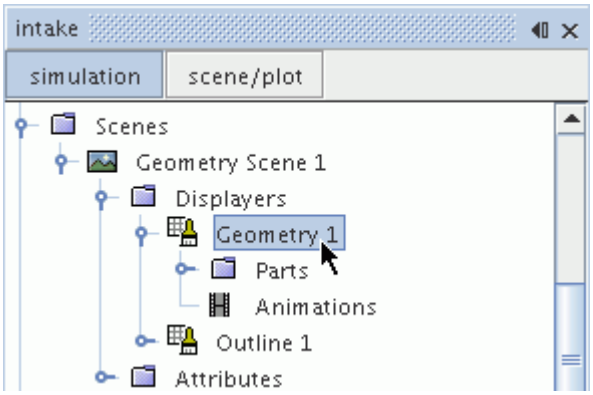
Changing the Color of the Surface

By default the part surfaces are colored according to their boundary type. However it is possible to change the color of the parts included within a displayer.

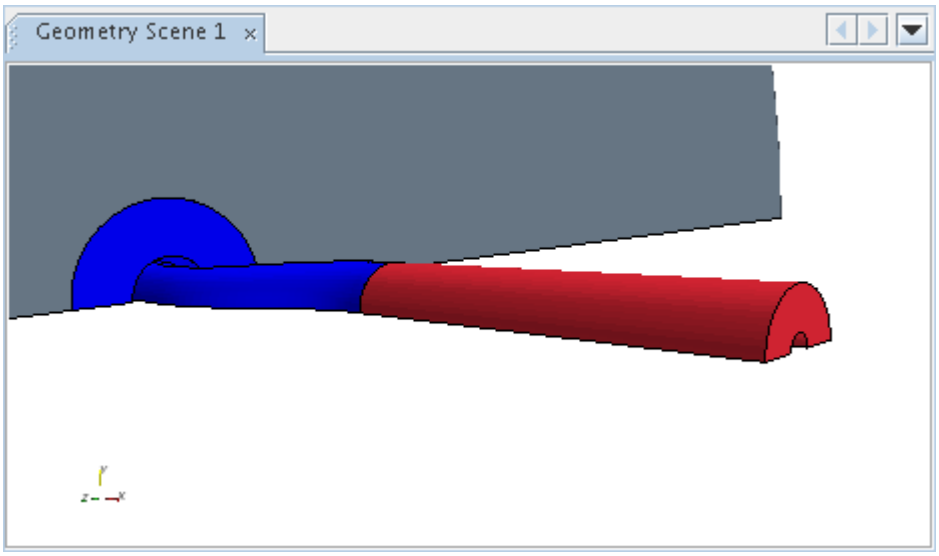
Color arrangements can be as follows:

- A different color for each region
- The same color for all parts in the displayer
- A different color for each part

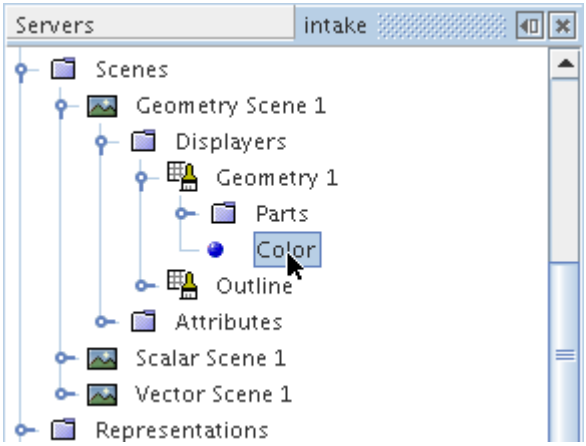
Expand the **Scenes** node and expand any scene that includes a geometry displayer (any default geometry, scalar or vector scene, or an empty scene in which you create a geometry displayer). Expand the **Displayers** node and select the geometry displayer node.



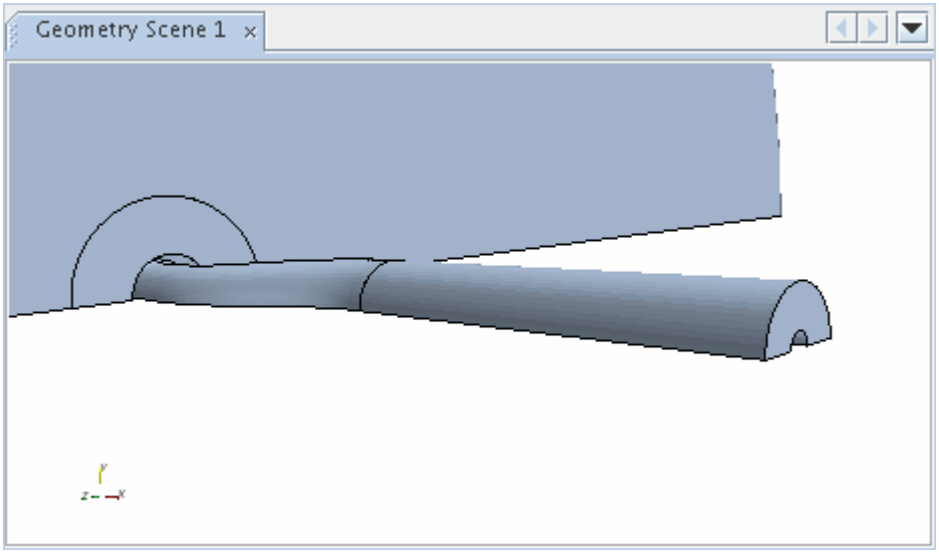
For coloring by region, set the *Color Mode* property to `Distinguish Regions`.



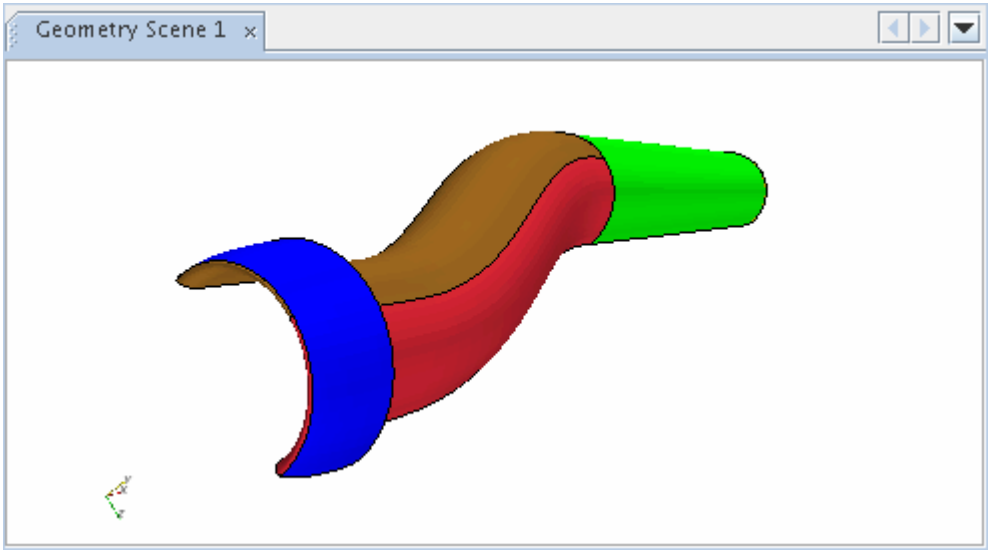
To apply the displayer color, set it to `Constant`. This makes the **Color** subnode available.



To change the displayer color, either enter the RGB value under the *Color* property of this node or click ... (**Custom Editor**) to open the color selector dialog.



If you wish to color different parts with different colors, set the *Color Mode* property to `Distinguish Inputs`.



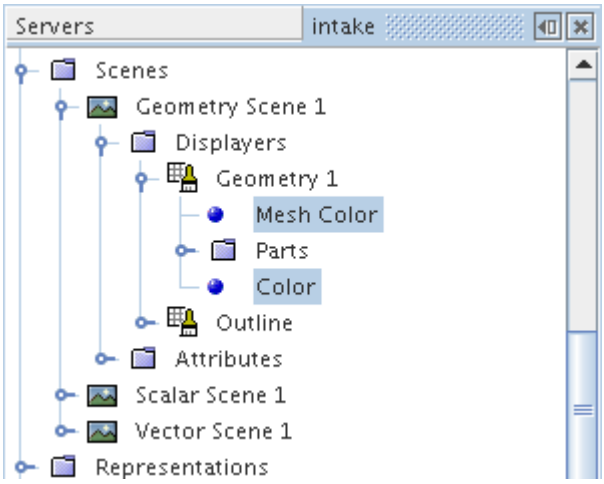
Contents:

Changing the Color of Parts in a Displayer
Changing the Color of the Surface Using the Displayer
Changing the Color of the Mesh Lines Using the Displayer

Changing the Color of Parts in a Displayer

You can change the colors of surfaces or mesh lines of parts within an individual displayer.

The **Color** and **Mesh Color** nodes are subnodes of the displayer node when activated, and let you control the surface color and mesh color, respectively.



Changing the Color of the Surface Using the Displayer

The **Color** node is activated when the displayer's *Color Mode* property is set to `Constant`.

This node controls the color of surfaces on parts that are included in the displayer.

Color Properties

Color	Sets the color to display a surface. The property customizer executes the define color action. See Defining Colors.

Changing the Color of the Mesh Lines Using the Displayer

The **Mesh Color** node is activated when the checkbox of the displayer's *Mesh* property is activated.

This node controls the color of mesh lines on parts included in the displayer.

Mesh Color Properties

Mesh Color	Sets the color of the mesh lines. The property customizer executes the define color action. See Defining Colors.

Using the Color Bar

Scene displays include multi-colored bars to map values to colors for scalars, vectors or streamlines.

You can control the appearance of the bar in the scene with the displayer's color bar object.

Contents:

What Is the Color Bar?
Changing the Position and Orientation of the Bar
Resizing the Bar
Switching the Scale Mode
Changing the Text Properties of the Color Bar
Changing the Background Color of the Bar
Changing the Colormap of the Bar
Copying Properties of a Color Bar

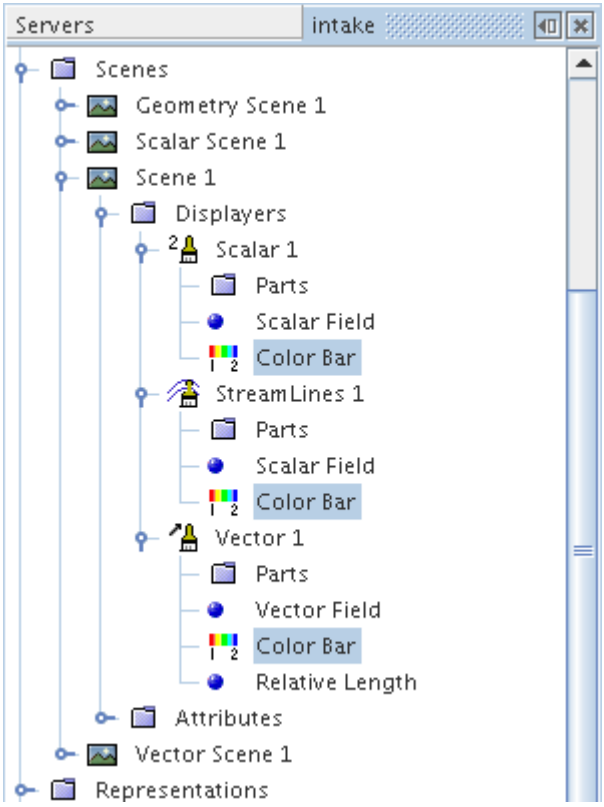
See Also:

Using the Scalar and Vector Bars

What Is the Color Bar?

The color bar controls the appearance of a scalar bar or vector bar in a scene display.

The **Color Bar** node has properties and is subject to common menu items.



Color Bar Properties

Color Map	Selects colors to indicate values from lowest to highest. This property lets you customize the use of colors in the scalar scene. For a list of the pre-defined choices, see Pre-Defined Colormaps.	
Levels	Sets the number of levels into which the range of data should be divided. Higher numbers produce finer blends between values. The default setting is 32.	
Width	Controls the width of the color bar. The number is a fraction of the display width; 1.0 equals the full width of the display.	
Font	Sets the font of the color bar text.	
Background	Activates or deactivates the color bar background.	
	Activated	Background color is active.
	Deactivated	Background color is not active.
Background Color	Controls the color of the background. The property customizer executes the define color action. See Defining Colors.	
Shadow	Activates a shadow effect for the text of the color bar.	
	Activated	Shadow effect is on.
	Deactivated	Shadow effect is off.
Reverse	Reverses the sequence of the selected colormap. See Using Colormaps.	
	Activated	Color sequence is reversed (for example, red to blue rather than blue to red).
	Deactivated	Uses the default color sequence.
Text Position	Sets the position of the text relative to the color bar.	
	Preceding	The text appears before (above) the color bar.
	Succeeding	The text appears after (below) the color bar.
Height	Sets the height of the color bar and its label.	
Title Height	Specifies a height for the title (field function label) of the color bar relative to the display height.	
Label Height	Specifies a height for the numeric labels of the color bar relative to the display height.	
Orientation	Defines orientation of the color bar. See Changing the Position and Orientation of the Bar.	
	Horizontal	Color bar is horizontal.

	Vertical	Color bar is vertical.
Position	Specifies the position in the display of the visualization display as X and Y coordinates from 0 to 1.	
Visible	Shows or hides the color bar.	
	Activated	Color bar is visible.
	Deactivated	Color bar is hidden.
Scale Mode	Activates a logarithmic scale mode for the color bar. See Switching the Scale Mode.	
	Linear	Uses linear scaling.
	Log10	Uses logarithmic scaling.
Text Color	Controls the color of the text. The property customizer executes the define color action. See Defining Colors.	

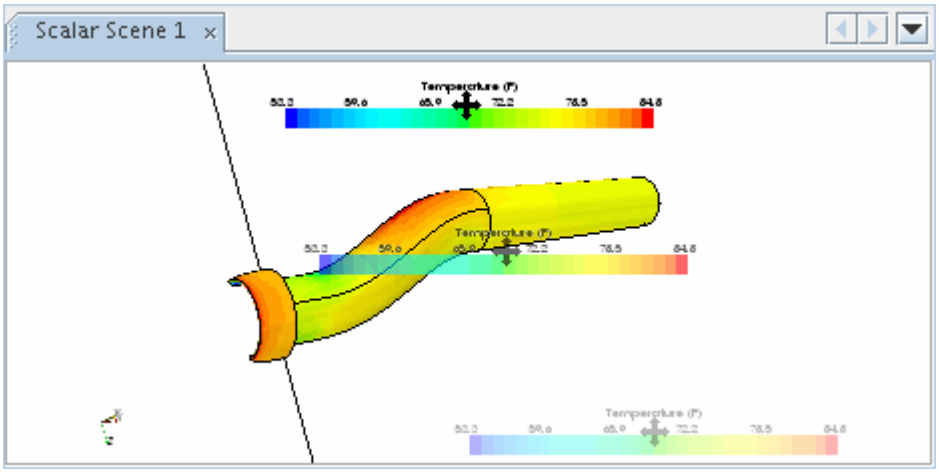
Expert Properties

Opacity	Adjusts the transparency of the color bar. The settings range from 0.0 (invisible) to 1.0 (opaque). This property allows you to show the bar as an overlay in front of the model.	
Label Format	Defines the numeric formatting of the numbers that are displayed along the color bar. The property customizer activates a dialog with interactive controls. See Controlling the Labels. Alternatively, text elements that can be entered here are identical to those described in the section What Is the Number Format?	
Ramp Type	Sets the shape of the table ramp to either linear, S-curve, or square-root.	
	Linear	Specifies a linear ramp, .
	S-curve	Uses the S-curve, which tails off gradually at each end: $y = \frac{\sin((x - 1/2)\pi) + 1}{2}$
	Sqrt	Uses the square root, $y = \sqrt{x}$.
Number of Labels	Controls the number of value labels along the color bar.	

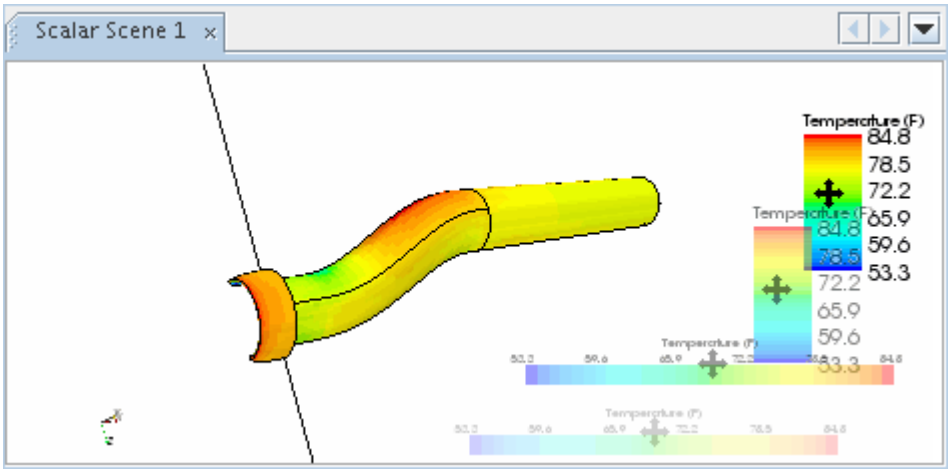
Changing the Position and Orientation of the Bar

The scalar bar or vector bar can be repositioned in the scene.

Click in the interior of the bar and drag it to the desired location.

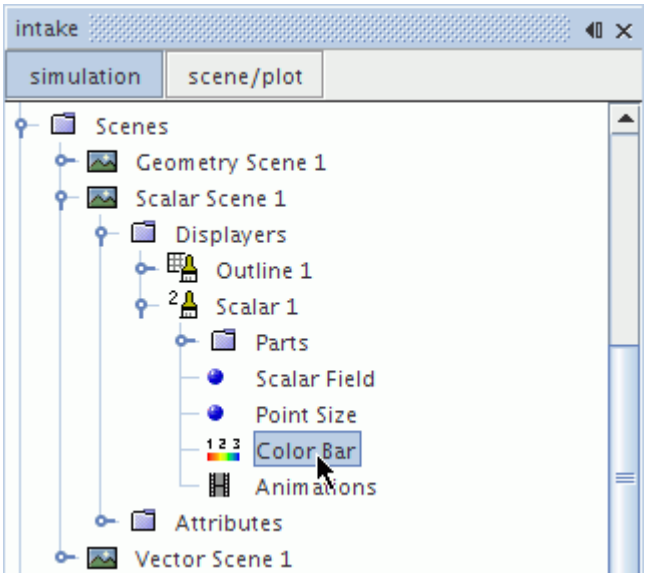


As the bar is moved close to the edges of the visualization display, the orientation of the bar automatically changes between horizontal and vertical.

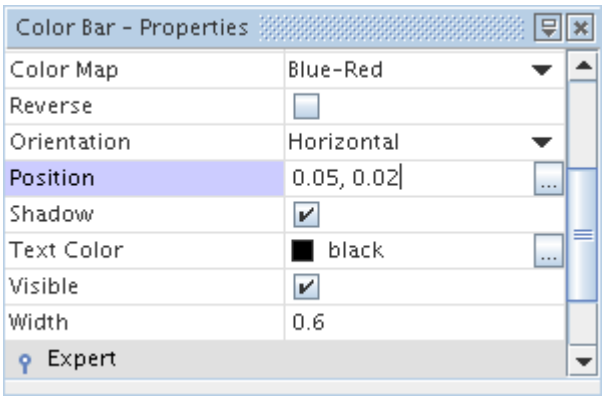


The bar may also be repositioned by entering the normalized coordinates of the location within the visualization display.

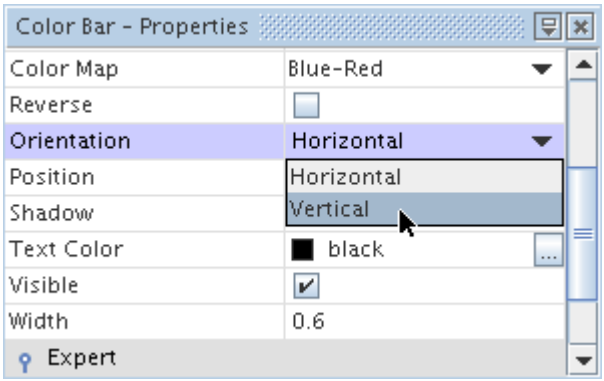
Expand the following nodes: the **Scenes** node, the node of a scene with a scalar or vector displayer, the **Displayers** node within that scene, and the displayer node. Select the **Color Bar** node.



Either directly enter the normalized coordinates under the *Position* property in the *Properties* window or click the property customizer (...) to enter the coordinates through a text input dialog.



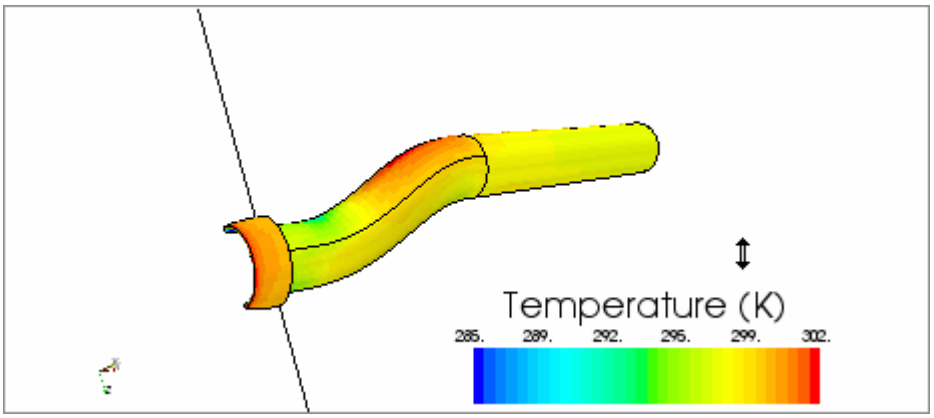
The orientation (horizontal or vertical) of the bar does not change automatically but you may change the orientation by selecting from the drop-down menu under the *Orientation* property.



The bar may be removed completely from the display by deactivating the checkbox of the *Visible* property, or by dragging the bar to locations outside the scene. If you drag the bar out of view by mistake, simply activate the *Visible* property's checkbox to restore it.

Resizing the Bar

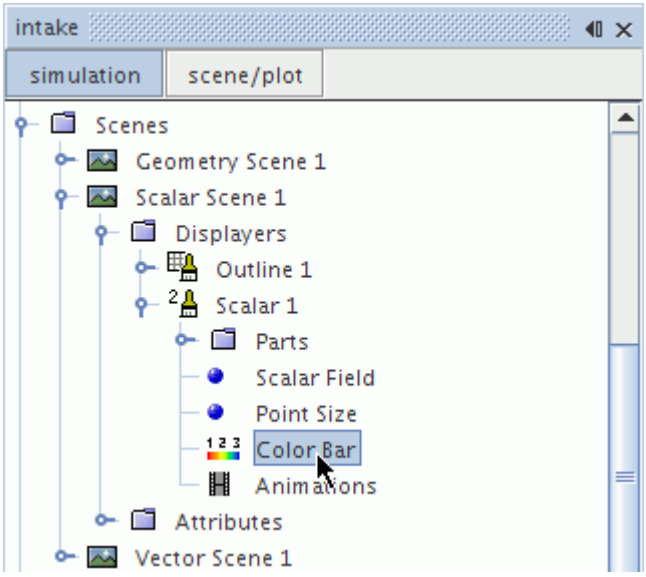
The scalar bar or vector bar can be resized by clicking the perimeter of the bar and dragging to enlarge or shrink it.



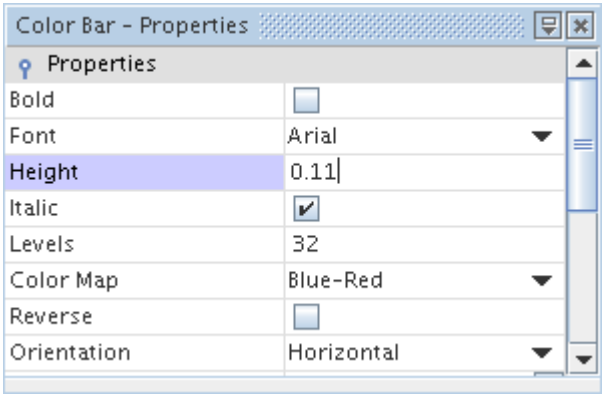
As the bar is enlarged, the text automatically changes size to maintain a good balance between the text size and bar size.

The size of the bar may also be changed by entering the width and height.

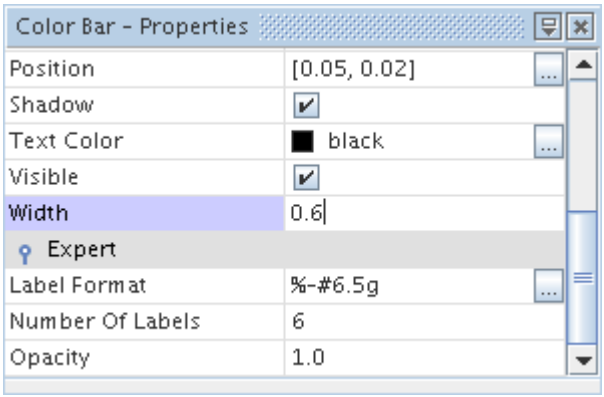
Expand the following nodes: the **Scenes** node, the node of a scene with a scalar or vector displayer, the **Displayers** node within that scene, and the displayer node. Select the **Color Bar** node.



In the *Properties* window, enter the height.

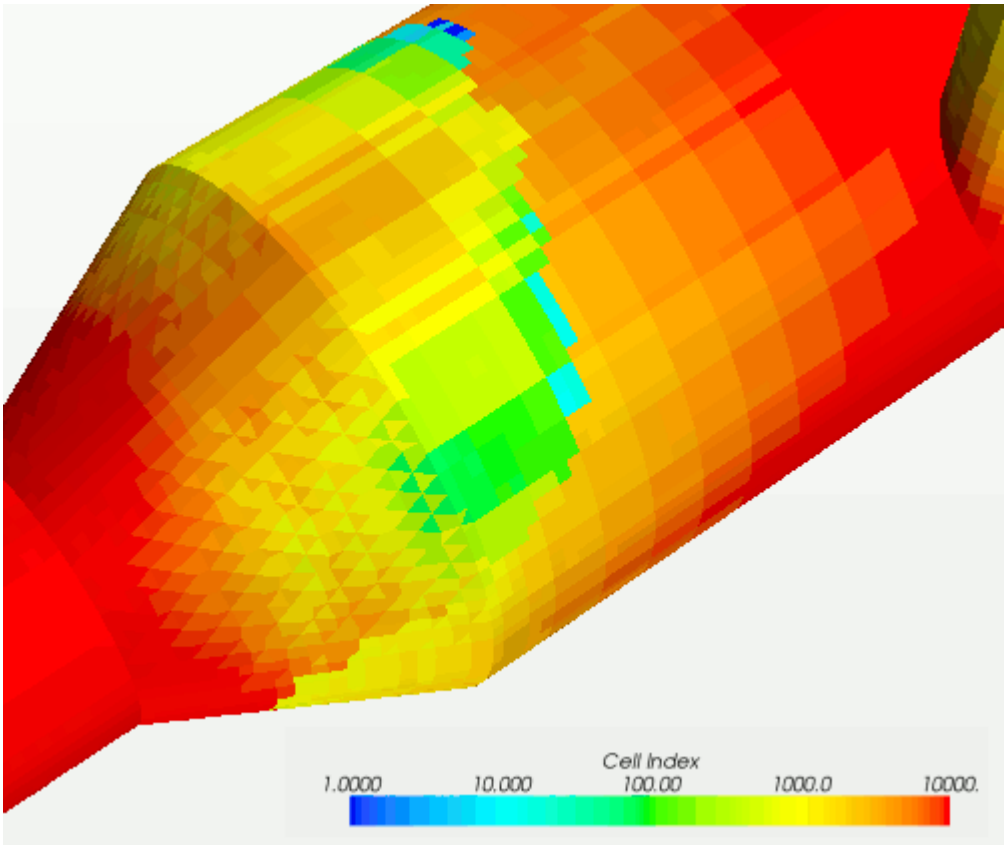


Then enter the width.

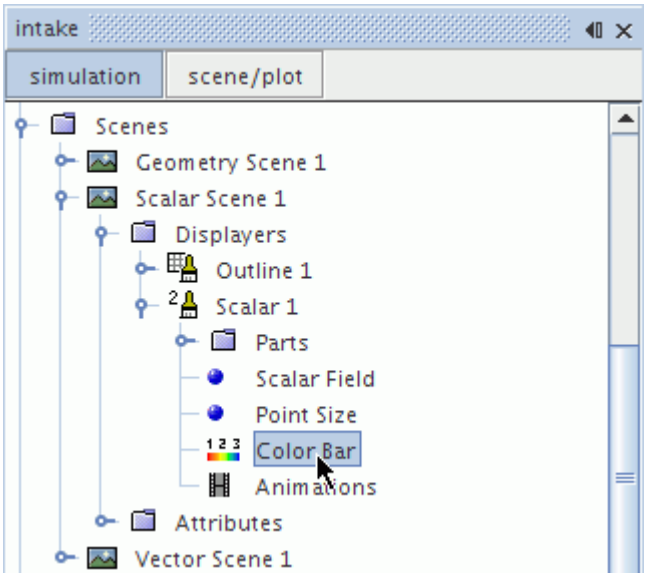


Switching the Scale Mode

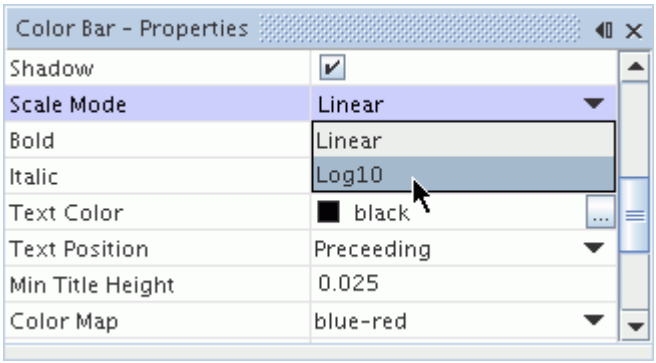
It is possible to switch the scale of the scalar bar or vector bar from linear to logarithmic, or vice versa. The logarithmic scale is helpful if you have a wide range of values to display.



To use this feature, expand the following nodes: the **Scenes** node, the node of a scene with a scalar or vector displayer, the **Displayers** node within that scene, and the displayer node. Select the **Color Bar** node.



The scale can be changed using the drop-down menu under the *Scale Mode* property in the *Properties* window.

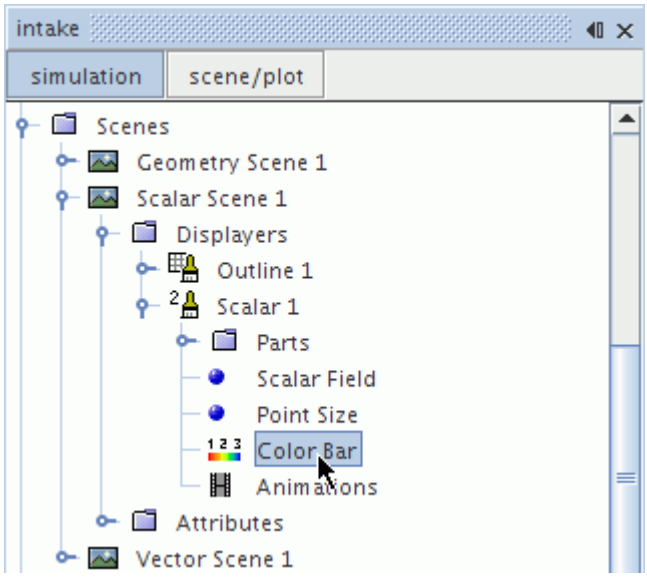


This feature works only if the function field ranges are positive or negative. If the ranges cross zero, then the display and the bar are disabled, showing the geometry model colored by default colors.

Changing the Text Properties of the Color Bar

The color bar text consists of the title (such as Velocity Magnitude) and the labels (numerical values). Controls are available for the entire text, the title, and the labels.

- To gain access to the color bar controls, select the **Color Bar** node.



Controlling the Text

Controls for the text (title and labels combined) are available in the properties:

- To change the font, or to make the text bold and/or italic, click ... (**Custom Editor**) of the *Font* property.
- To change the color of the text, select a color from the list in the *Text Color* property. Alternatively, click ... (**Custom Editor**) to activate the color selector.
- To control whether the text is above or below the color bar, use the *Text Position* property:

  - The `Preceding` option (default) puts the text above the bar.
  - The `Succeeding` option puts the text below the bar.
- To adjust the size of the entire text, resize the bar in the display.

Controlling the Title

The title of the color bar comes from the name of the selected scalar or vector.

- To specify a height for the title, enter a value for the *Title Height* in the properties. The value is a fraction of the height of the display window, where 1.0 is the full height.

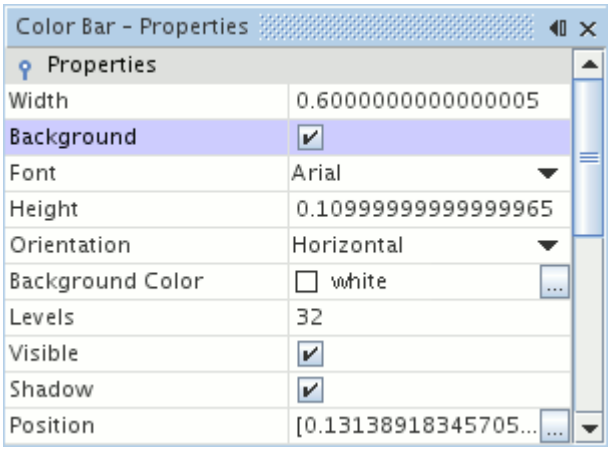
Controlling the Labels

- To change the number of labels in the color bar, enter a different value for the *Number of Labels* expert property.
- To specify a height for the labels, enter a value for the *Label Height* property. The value is a fraction of the height of the display window, where 1.0 is the full height.
- In the *Label Format* expert property, enter text elements that are identical to those described in the section What Is the Number Format?
- To make other changes to the labels, see Controlling Display Formats.

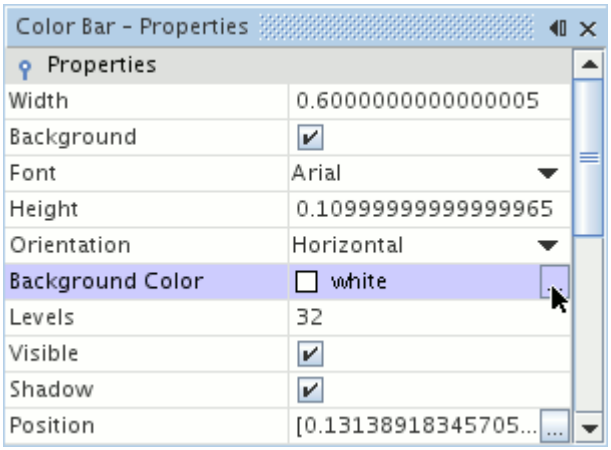
Changing the Background Color of the Bar

To change the background color of the scalar bar or vector bar, select the **Color Bar** node.

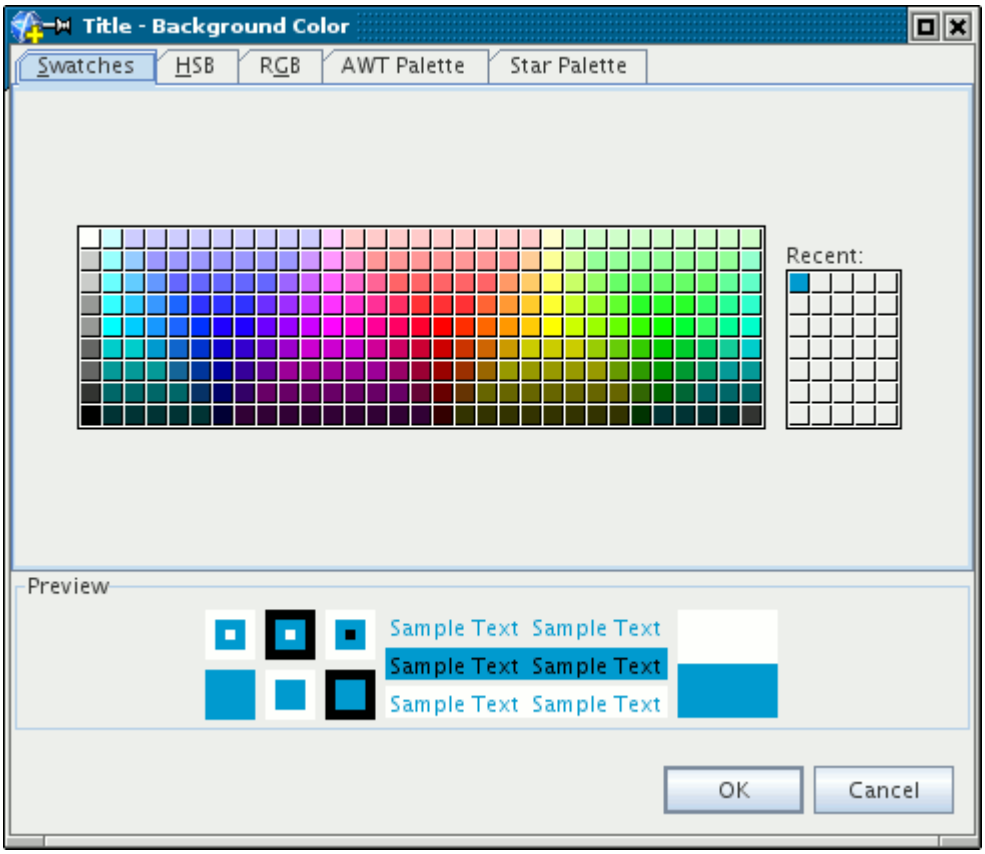
Within the *Properties* window, activate the *Background* property checkbox.



Then, click the ellipsis button to the right of the *Background Color* property.



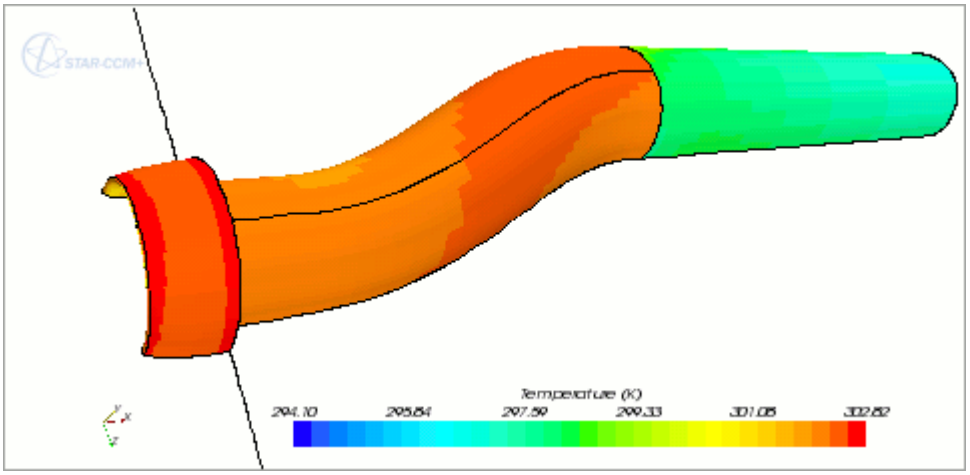
This opens the *Background Color* dialog, where colors can be selected.



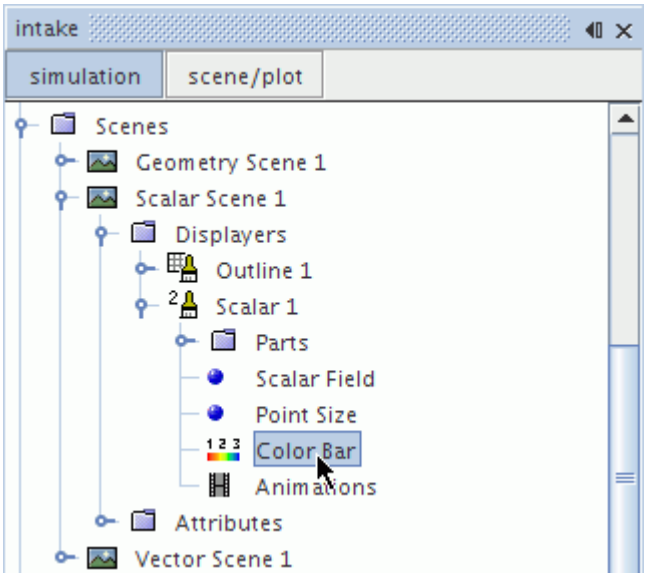
Changing the Colormap of the Bar

Various colormaps (combinations of colors) are available for you to use in the display.

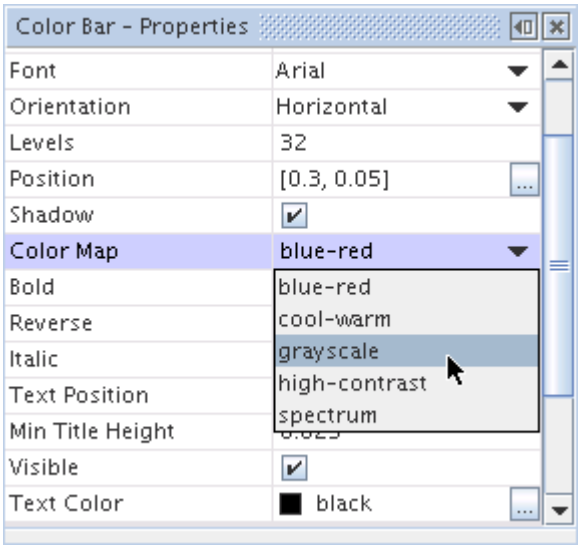
By default the colormap of the scalar bar or vector bar uses the blue-red gradient.



To switch from the default, expand the following nodes: the **Scenes** node, the node of a scene with a scalar or vector displayer, the **Displayers** node within that scene, and the displayer node. Select the **Color Bar** node.

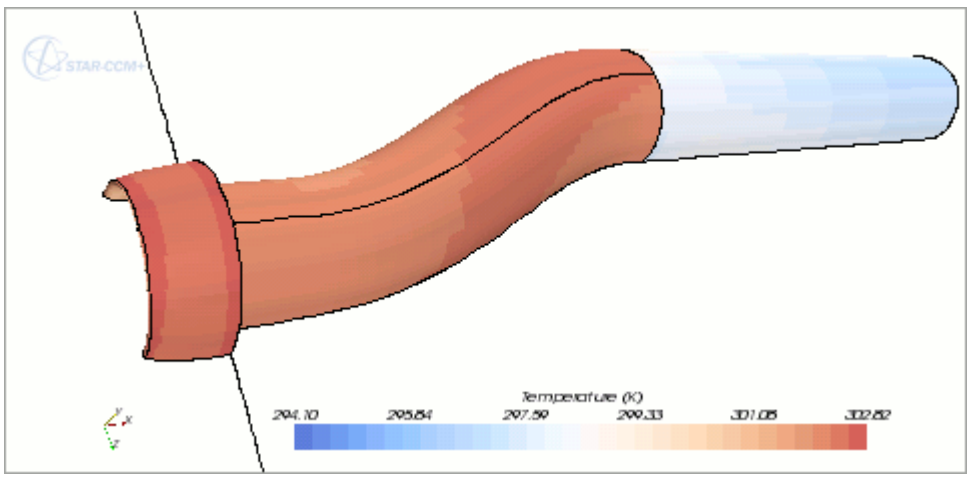


Select a colormap from the drop-down menu under the *Color Map* property in the *Properties* window.

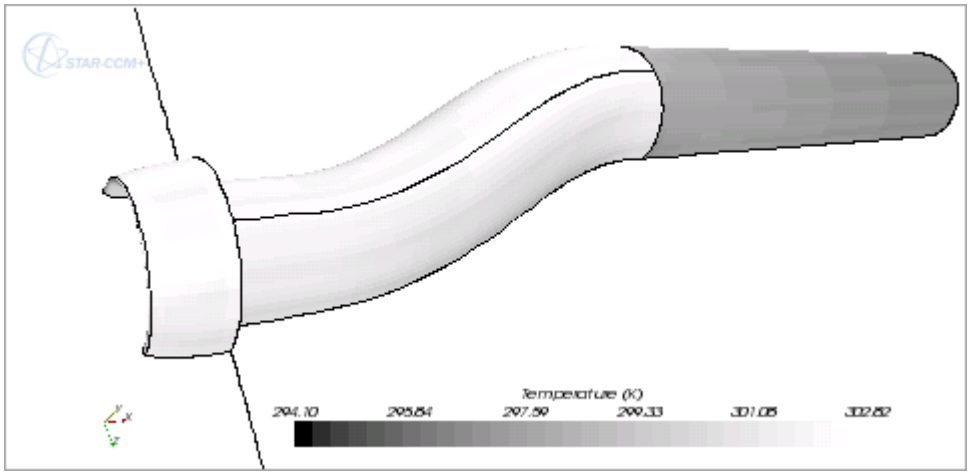


The options besides the default include:

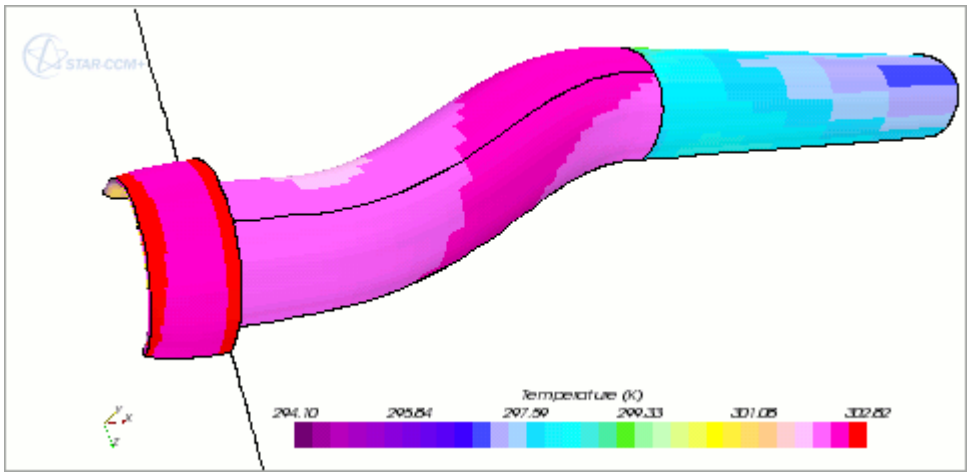
- `Cool-warm`



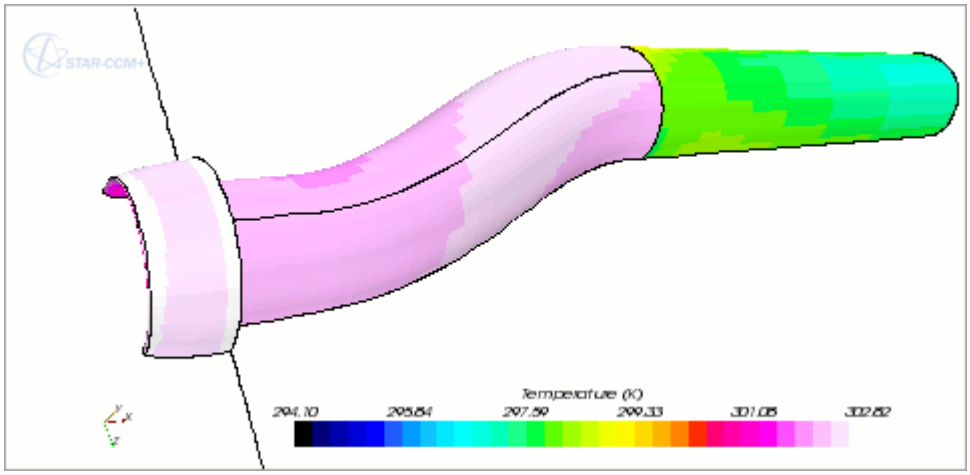
- `Grayscale` (useful for illustrating a printed report)



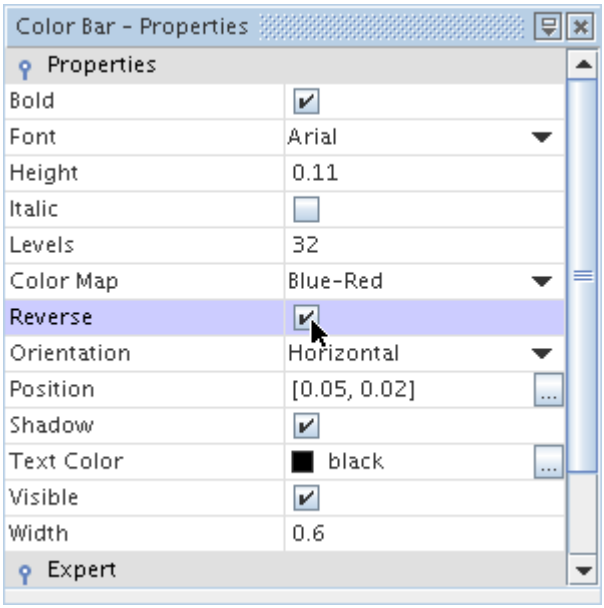
- `High-contrast` (shades that contrast sharply for easy viewing)



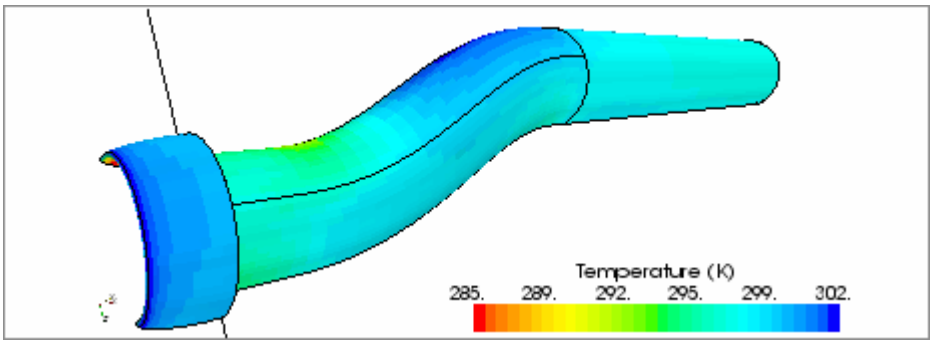
- `Spectrum`



In addition, the *Reverse* property lets you display the color in the opposite sequence when you activate the checkbox.

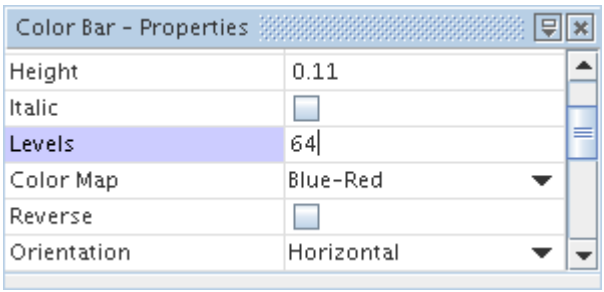


For example, the default blue-red setting appears as red-blue when the *Reverse* property is activated.

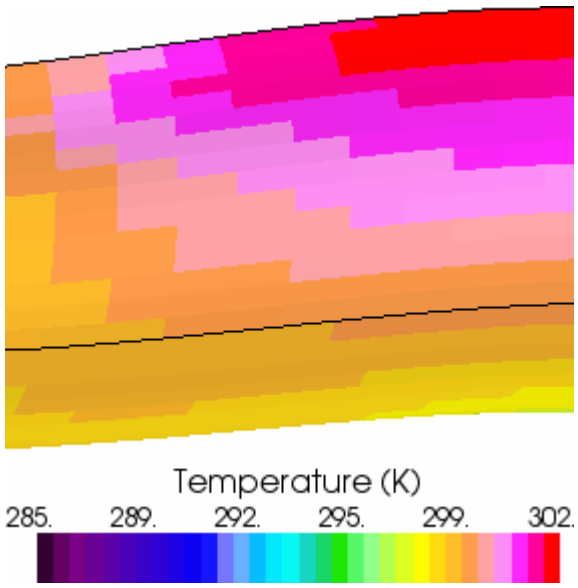
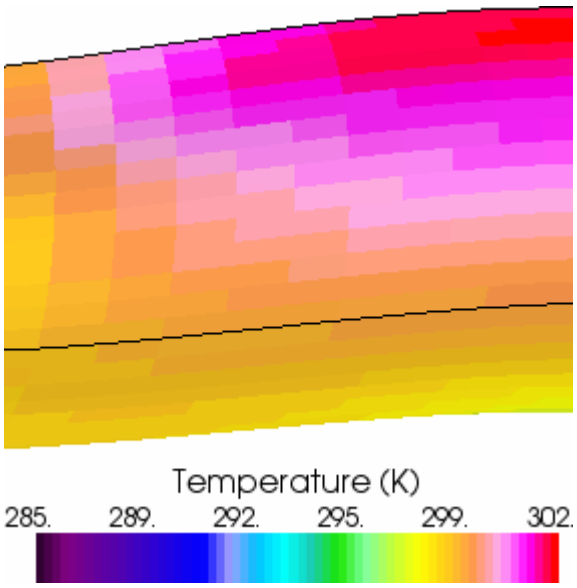


The number of color bands can be changed using the *Levels* property.

In the *Properties* window, enter the new number of color levels.



The change is visible in the display for both the bar and the visible parts.

	
32 levels (default number)	64 levels

Copying Properties of a Color Bar

When properties are copied from one color bar node to another, all of the properties are applied so that the destination color bar matches the source one.

Because color bars in scalar and vector scenes have the same set of properties, you can copy color bar properties between the two types of scenes.

Using a Combination of Displayers

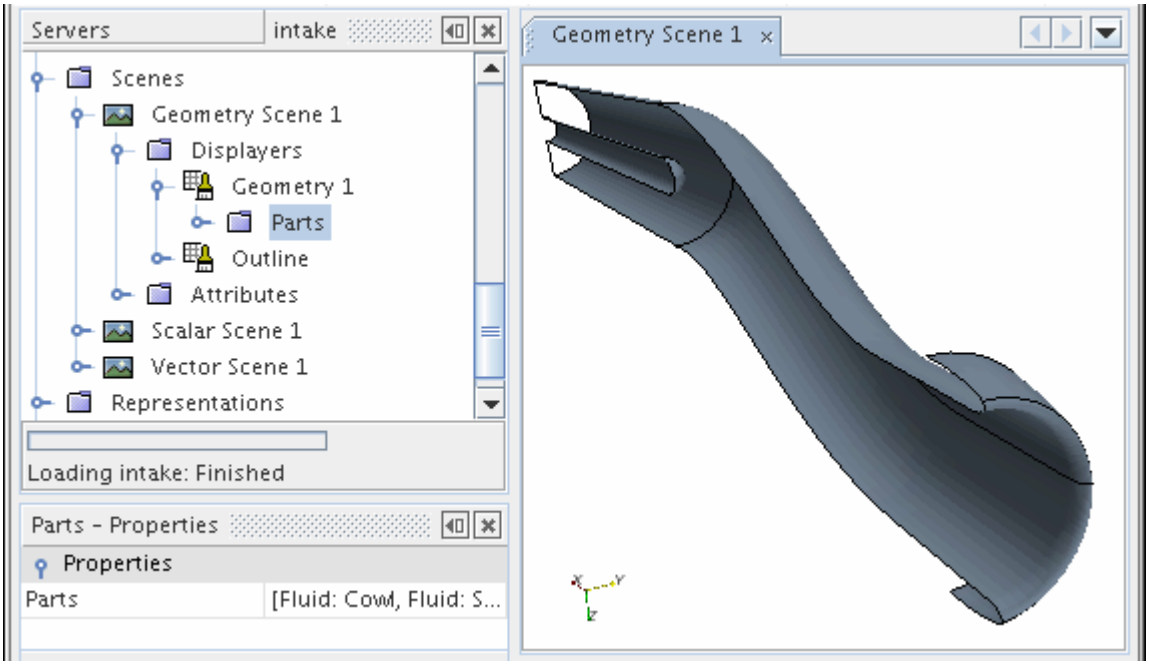
Combinations of displayers can be used to construct visualizations with a combination of surfaces, scalars, vectors and streamlines.

By default: geometry, scalar, and vector scenes use more than one displayer to show both the outline of the boundary parts and either surfaces, scalars and vectors.

This section describes a simple example in which a combination of displayers are used to construct a scene.

Initial Geometry Scene

Initially a geometry scene is opened which includes two geometry displayers. The first displayer displays the outline of all the boundary parts. In the example, some parts have been removed from both of the displayers. The scene is viewed through the symmetry plane to display the inside of the cowl and manifold.

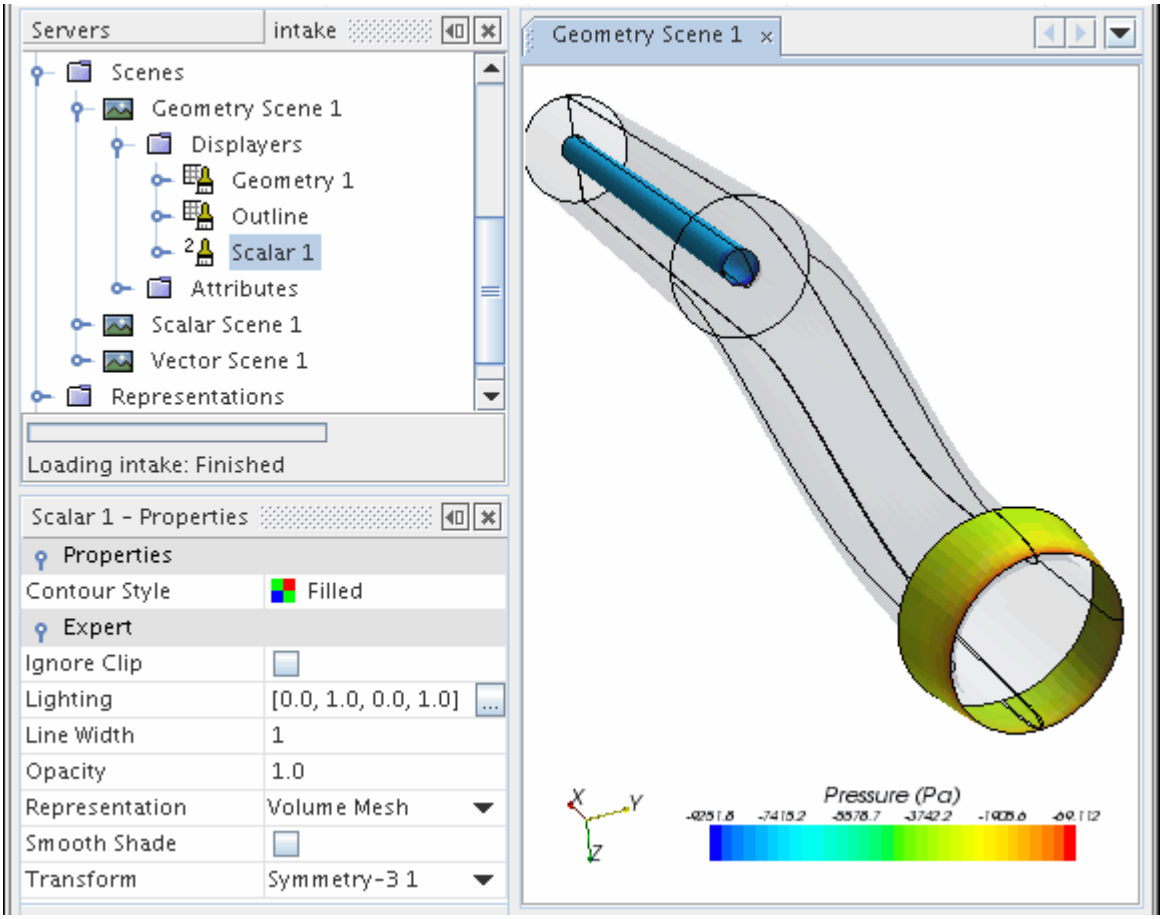


Displayer Transformation

Both displayers are then transformed using a transform automatically generated from the symmetry plane part. The opacity of the surface geometry displayer is decreased to provide transparency. This is useful to maintain surface definition but allow inspection of parts beyond the surface.

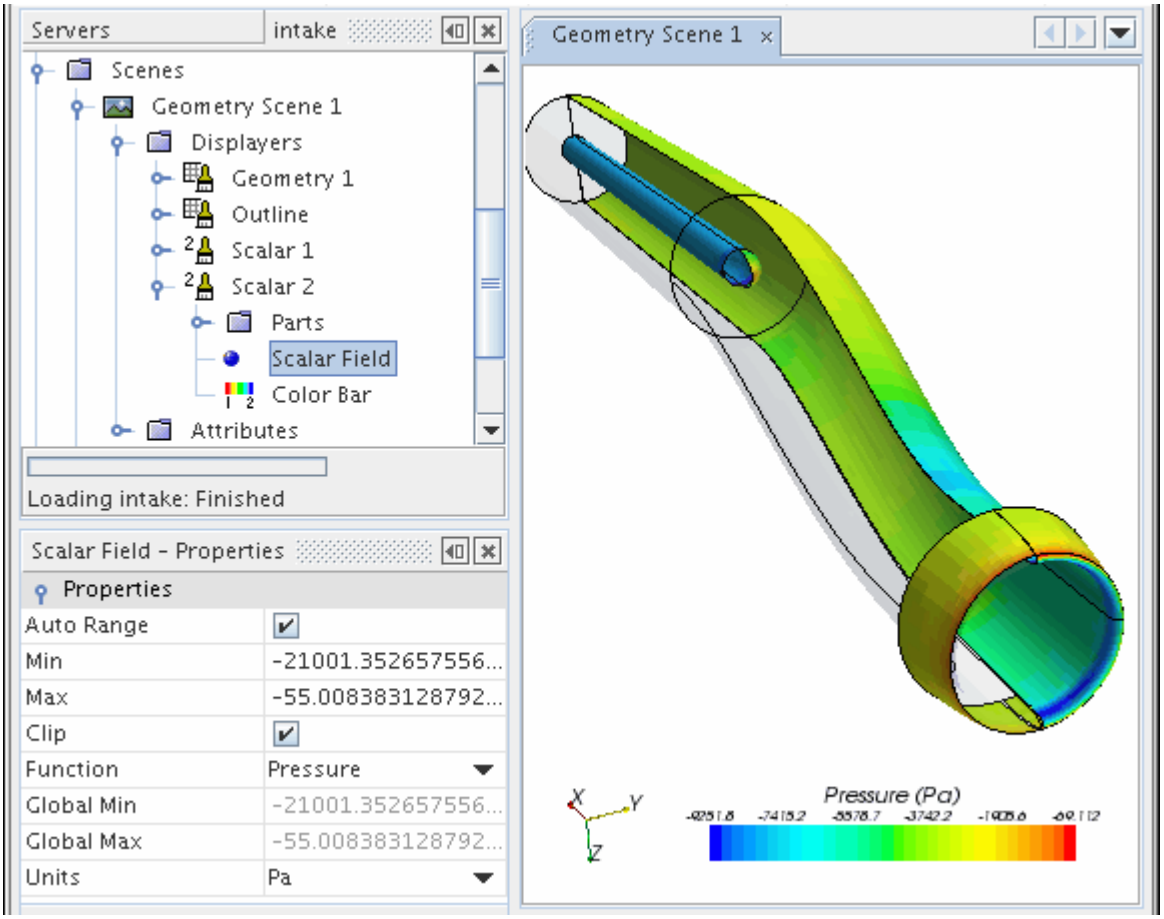
Addition of a Scalar Displayer

A scalar displayer is then added to the scene and transformed using the same transform. Only the cowl and splitter parts are added to the displayer and the Pressure field function is displayed on the surfaces.



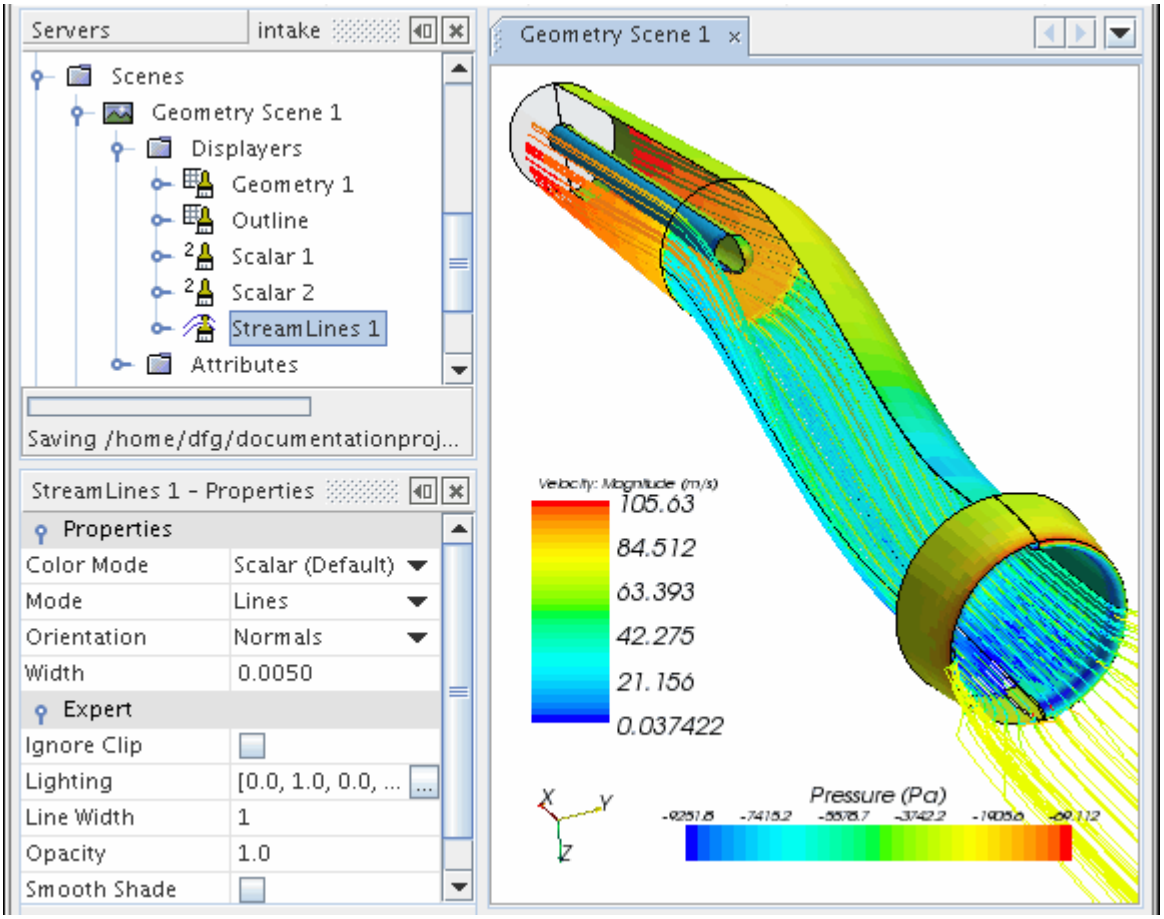
Addition of a Second Scalar Displayer

Another scalar displayer is then created with only one side of the manifold added. Again the Pressure field function is displayed on the surface. The only difference with the previously added scalar displayer is that it is not transformed, maintaining the ability to view inside the manifold. Since both scalar displayers use the same field function the second scalar bar is rendered invisible.



Addition of a Streamline Displayer

Finally a streamline displayer is added to the scene with the same transform applied and the Velocity Magnitude field function used to color the streamlines. The streamline bar is moved to the left-hand side of the visualization display which automatically changes its orientation.



The result is a complex visualization constructed using a few simple steps.

Lighting Displayer Surfaces

The surface lighting properties of all parts in a displayer can be adjusted using a combination of parameters.

The parameters are:

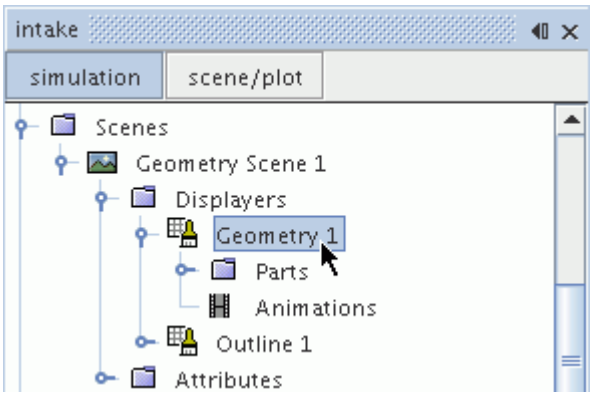
- Ambient lighting - accounts for light reflected or scattered from other surfaces in the scene
- Diffuse lighting - accounts for the angle of incidence of light onto a surface
- Specular lighting - accounts for the direct reflection of the light source off a shiny object
- Specular power - adjusts the intensity of specular lighting

The combination of these four parameters represents how the surface reflects and absorbs the scene lighting. By default, the ambient parameter is set to 0.0, and the diffuse parameter is set to 1.0, resulting in a relatively good setting to distinguish easily the depth and proximal relationships between three-dimensional objects in the scene. The default setting may not always be suitable for viewing scalar contours, as surfaces which are perpendicular to the view normal may become dark. In this case, it is recommended to set the ambient parameter to 1.0 and the diffuse parameter to 0.0 to see the scalar colors clearly.

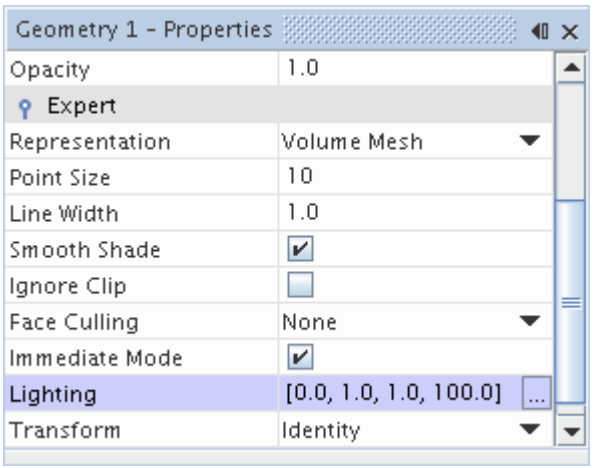
Your settings of the specular and specular-power parameters depend on how you want your simulated surface to appear. The specular parameter controls how the light is reflected from the surface. Specular light comes

from a particular direction, and it tends to bounce off the surface in a preferred direction. The level of specularity is essentially how “shiny” a surface is. Shiny metal or plastic has a high specular component, and chalk or carpet has almost none. The specular-power parameter controls the tightness of the light source: higher values of specular power produce a tighter light source, while lower values broaden the light source.

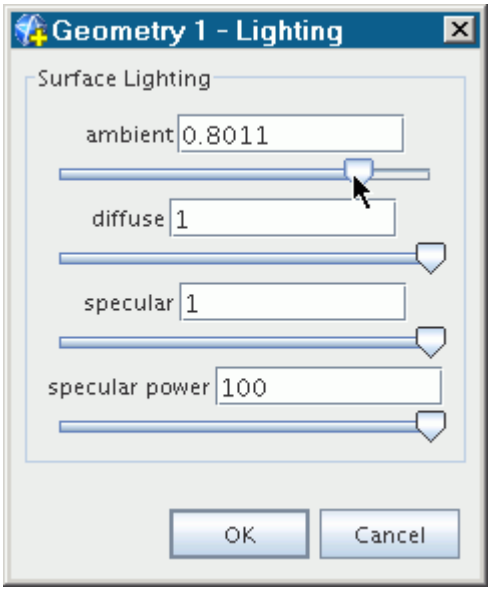
To change any of the parameters expand the **Scenes** node, any scene node and the **Displayers** node. Select any displayer which includes surfaces (geometry or scalar surfaces).



The value of each parameter may be entered directly under the *Lighting* expert property in the *Properties* window.



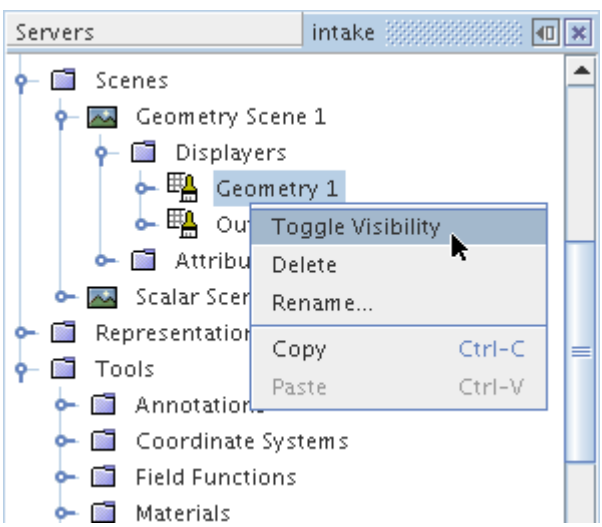
Alternatively, you can click ... (**Custom Editor**) to open the *Lighting* dialog through which the parameters can be set using sliders.



Hiding and Showing All Parts in a Displayer

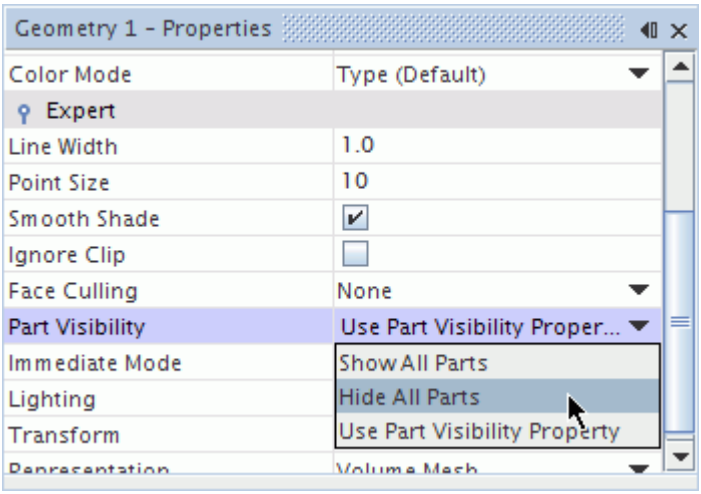
You can render all the parts in the scene displayer visible or invisible with a single command.

To do this, right-click the displayer node and select **Toggle Visibility**.



When you use this feature, STAR-CCM+ remembers the visibility settings of the individual parts in the scene. Hence, when you render the displayer visible again, the individually hidden parts remain hidden.

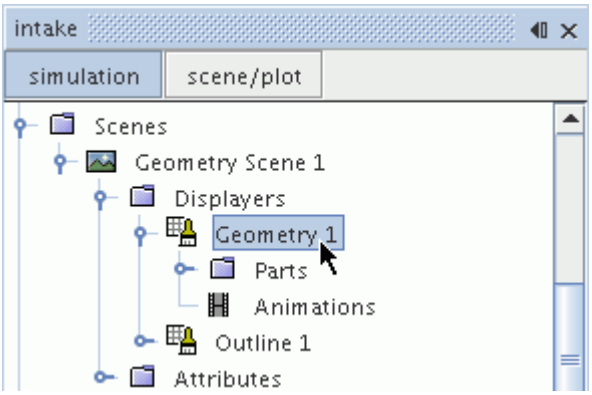
You can also control the visibility of parts with the *Parts Visibility* expert property of the displayer. It lets you show all parts, hide all parts, or revert to the visibility settings of the individual parts.



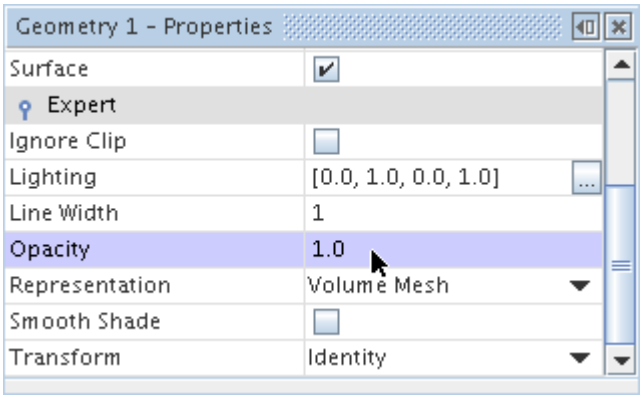
Making Displayer Surfaces Transparent

Displayer surfaces can be made transparent. This is useful when surface definition is desirable but you wish to see what is beyond the surface.

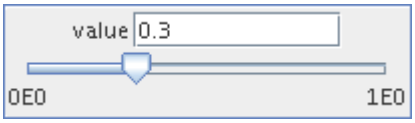
To make displayer surfaces transparent expand the **Scenes** node, any scene node and the **Displayers** node. Expand any displayer which displays surfaces (geometry or scalar displayers).



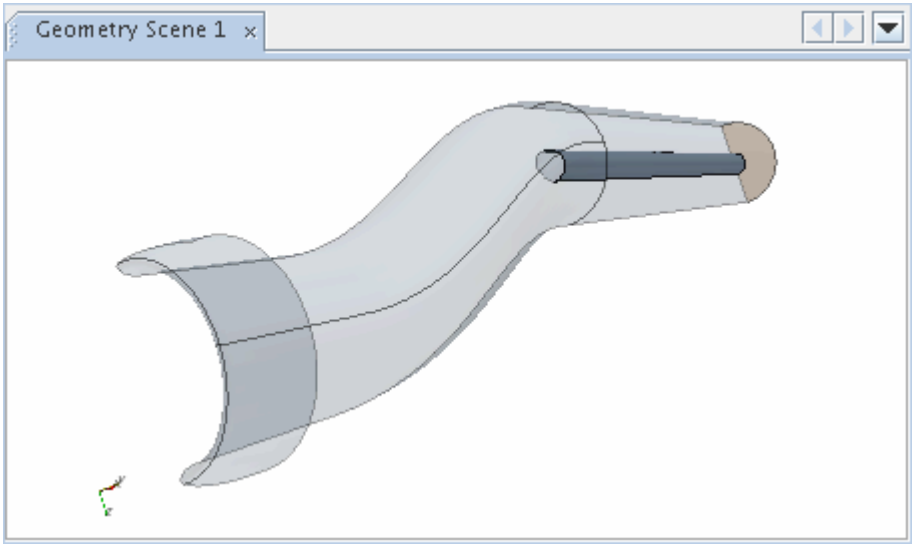
Expand the **Expert** category in the *Properties* window and click the right half of the *Opacity* property.



Use the slider or the text box to set a value between 0 and 1.0, where 0 means the parts are fully transparent and 1.0 means the parts are fully opaque.



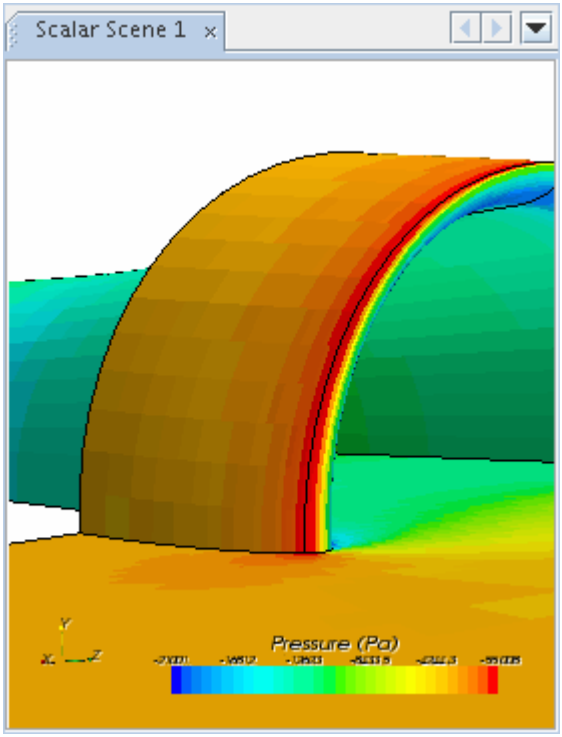
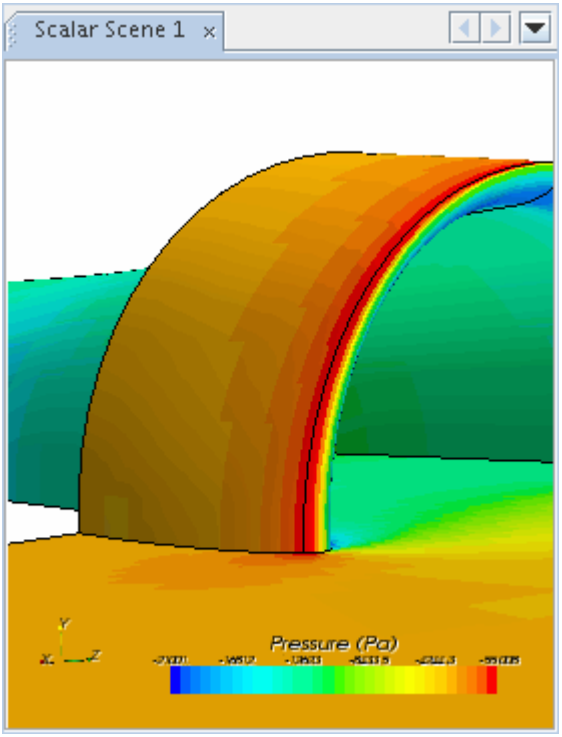
In the example below two geometry displayers are used to display different parts with different degrees of transparency.



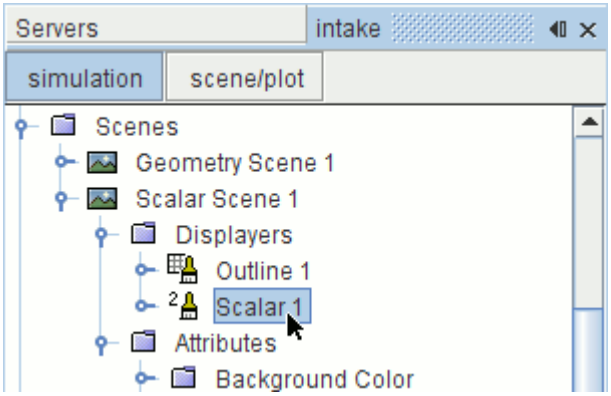
Smooth Shading Displayer Surfaces

The normals of the displayer surfaces determine how light is reflected from the surface and thus the appearance of the surface.

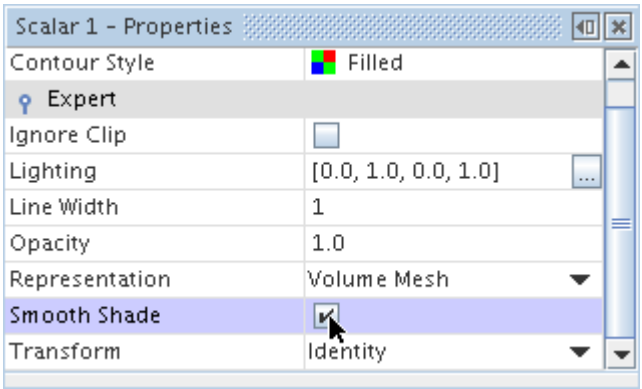
By default the normals are calculated at the center of the boundary face polygons which result in a faceted surface appearance. When the normals are calculated at boundary face vertices, the resulting appearance of the surface is smooth.

	
Faceted surface	Smooth surface

To obtain smooth part surfaces expand the **Scenes** node, any scene node and the **Displayers** node. Expand any displayer which displays surfaces (geometry or scalar displayers).



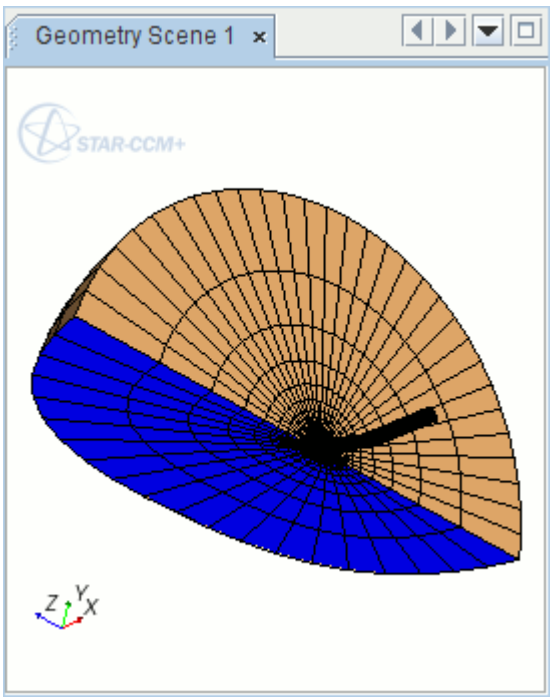
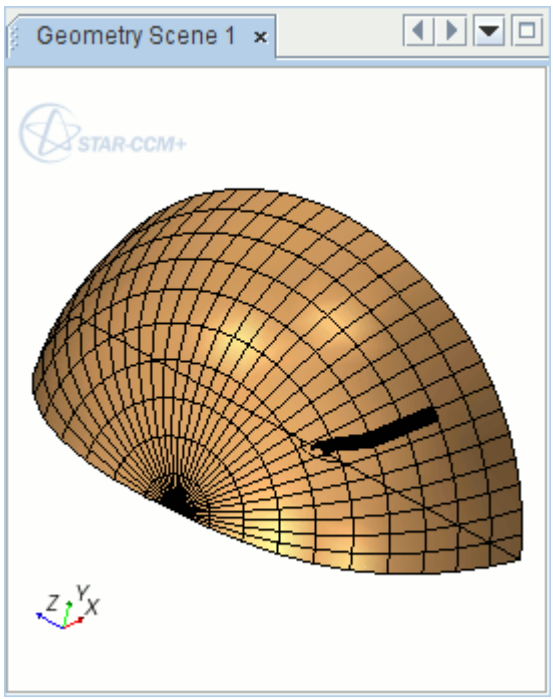
Expand the **Expert** category in the *Properties* window and activate the checkbox of the *Smooth Shade* property.



Culling Polygons

You can control the display of polygons depending on their orientation.

Each face in a polygonal surface has two sides: outward-facing, and inward-facing, determined by the vertex ordering of the polygon. A displayer's *Face Culling* expert property tells the display to not draw (to cull) faces which have the orientation selected in the property's drop-down list. This results in faster display because only half the faces are actually drawn.

	
No culling	Front faces culled

The best uses for this are mainly with baffles, where turning on this feature speeds up display in some cases. Culling can also be used to check the surface for defects; that is, turn on backface culling and you can see holes, which are possible signs of an improperly oriented surface.

Copying Displayers

A displayer can be copied. When a displayer is copied, all properties are copied, resulting in an exact duplicate of the original.

You can also copy the properties of one displayer onto another, provided they are of the same type.

Contents:

Copying Properties of a Displayer

Copying Properties of a Displayer

When properties are copied from one displayer to another, all of the properties are applied so that the destination displayer matches the source displayer.

Note: It is not possible to copy properties between different types of displayers; for example, the properties of a geometry displayer cannot be copied onto a vector displayer.

Pop-Up Menu of the Displayers Manager

These pop-up menu items are available when you right-click the **Displayers** manager node.

New Displayer	Creates a displayer of the type you select in the submenu.	
	Geometry	Executes the add displayer action to add a geometry displayer to the scene. With more than one geometry displayer in the same scene, you can show various parts in different modes. For example, a rotor blade can be shown as a solid (shaded) part with no grid, while the rotor hub can be shown with grid cells (all edges) only.
	Scalar	Executes the add displayer action to add a new scalar displayer to the scene. See Creating Geometry Displayers. Creating a scalar displayer allows you to select a different part on which to show scalar data, either the same type of data or a different type.
	Vector	Executes the add displayer action to add a vector displayer to the

		scene. See Creating Scalar Displayers. Creating a vector displayer allows you to select a different part on which to show vectors, either the same type of vector or a different type.
	Streamline	Executes the add displayer action to add a new streamline displayer to the scene. See Creating Streamline Displayers. No parts other than streamline derived parts can be selected by this displayer. If your simulation has no streamline derived parts, you can still create this displayer. However, you have nothing to add to its Parts manager.

Advanced Rendering

Advanced rendering provides realistic visual presentation at an economic computational cost through *ray tracing*. Ray tracing simulates the travel of light as it passes through the domain and comes into contact with objects in that domain.

Advanced rendering effects include:

- Shadows: If a surface is between the light and the rendered surface, the rendered surface is not illuminated by the light.
- Reflection: Surfaces can be given reflective effects through material properties, resulting in mirror effects.
- Refraction: Surfaces can be given refractive effects through material properties. Refraction changes the light directions based on the index of refraction values of air and neighboring objects (for example looking through a clear glass of water).
- Accurate transparency

Using this technique, you can depict your project in a photorealistic display.

Note: GPU Utilization options do not apply to advanced rendering. (For details about GPU Utilization, see Controlling Graphics Performance.)

Limitations in the Current Release

- Advanced rendering effects are not supported with the following:
  - 3D-CAD views
  - Mesh scenes
  - Electronics cooling
- Advanced rendering is not performed on the following objects, although these objects are still rendered in the default rendering mode and blended as well as possible with the advanced rendering result:
  - Lines
  - Points

- Spheres or other elements rendered as point sprites; for example, such rendering occurs when the *Point Scale Mode* property of a scalar displayer is set to **Pixels**.

  A point sprite is a quadrilateral 2D space onto which STAR-CCM+ projects a view of the 3D geometry without rendering the entire 3D object.
- Text
- Image annotations other than 3D
- Line Integral Convolution

- Advanced rendering is not supported by STAR-View+.
- When spheres or other elements are scaled with a transform, advanced rendering does not use implicit representation for better quality and performance. For example, when a simple transform with a *Scale* setting of `2,1,1` is applied to a sphere, that sphere becomes an ellipsoid. The ellipsoid is triangulated and then advanced rendering is applied to the best degree possible.

Contents:

Advanced Rendering Workflow
Advanced Rendering Reference

Advanced Rendering Workflow

You can set up your display with photorealistic graphics.

Start advanced rendering:

1. Set the number of threads available for advanced rendering using one of the following techniques:
   - The user interface:
     1. Select **Tools** > **Options**.
     2. In the left window of the dialog, select the **Services** > **Visualization** node.
     3. Enter `9999` for the *Number of Client Render Threads* property. This entry causes STAR-CCM+ to determine automatically the maximum number of processes that you can actually set.
     4. Click **Close**.
   - The command line, for advanced users (the following example is a typical set of line commands):

     ```
     % starccm+ -batch -renderthreads <number of threads>
     ```

     The value specified with the `-renderthreads` command line option takes precedent over the corresponding setting that is available through **Tools** > **Options**.

     The default value for `-renderthreads` is 1 if you do not specify a value. When you use the `-batchsystem` option, `-renderthreads` is restricted to 1. If you attempt to specify a `-renderthreads` value greater than 1 when using `-batchsystem`, STAR-CCM+ generates a warning message.

Note: If you specify a high value for render threads, you can incur resource contentions that interfere with the proper functioning of your computer.

Note: Increasing render threads is expected to scale performance linearly. However, linear scaling can fail with high thread counts on hyper-threaded or multi-CPU systems, although performance can still increase. Complexity of the scene can also impact performance and prevent scaling.

2. Within the **Scenes** node, select the scene object on which you want to apply advanced rendering and activate *Enable Advanced Rendering*.

   The initial rendering resembles the normal STAR-CCM+ visualization mode (OpenGL).

Set up the rendering:

3. Expand the scene node. Decide what settings to apply to all the displayers in the scene using the **Attributes** > **Advanced Rendering Properties** node. The key setting is the *Quality* property; a higher value requires longer rendering time, and STAR-CCM+ shows an estimate of the time in the scene.. For details, see Advanced Rendering Properties.

   In particular, the *Shadow Override* property lets you control whether you apply the same set of shadow characteristics to the entire scene, or customize the shadows for each displayer.

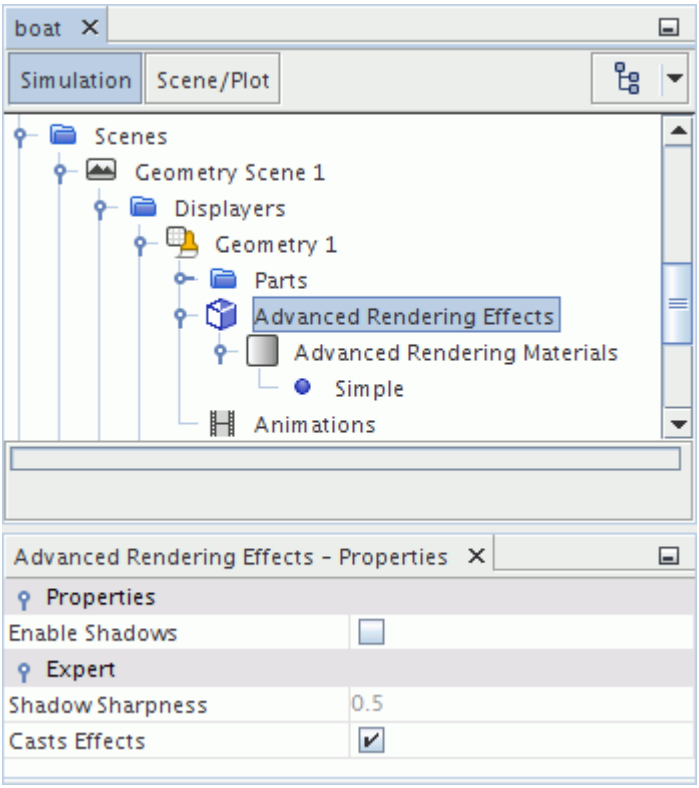
4. If you want to apply a preset material to the surfaces defined within a displayer:
   a) Select the displayer node and set *Color Mode* to **Preset Material**.
   b) Select the **Preset Material** sub-node of the displayer node and choose from the drop-down list of the *Material* property.

Note: If you choose the **Preset Material** setting, you do not have the use of the **Advanced Rendering Effects** > **Advanced Rendering Materials** node.

5. If you want to define an additional preset material, right click the **Tools** > **Rendering Materials** node and select a type of material from the **New Rendering Material** submenu. Enter the characteristics of your new material in the properties of the node. For details, see Rendering Materials.

Note: If you want to create a material that is similar to one of the supplied ones, make a copy of that node, and then edit its properties.

6. To apply rendering effects on objects within the displayer, work with the **Advanced Rendering Effects** node (for details, see Advanced Rendering Reference):



a) Use the properties of this node for the following:
   - Allowing shadows for the parts in the displayer
   - Shadow sharpness
   - Casting effects onto other surfaces; including shadows, reflections, and refractions
   - Removing transparency, self-shadowing, and self-reflection.

b) Select the **Advanced Rendering Effects** > **Advanced Rendering Materials** node. These material options let you apply more custom settings in the displayer. These settings are not available if you set the *Color Mode* of the displayer to **Preset Material**.

c) Select a type of rendering material from the drop-down list in the *Effect* property.

   Depending on your choice, a sub-node is available for you to specify more settings. For example, the **Metallic** sub-node includes specular color and roughness.

   Note: Photorealistic metal surfaces do not use RGB colors.

If the scene takes time to render, text appears with the estimated time to completion. During this time, you can still interact with the scene.

When you generate a hardcopy of the scene with advanced rendering, the *Save to File* dialog also includes an estimated time. The time estimate is based on the duration of the previous render pass, when available.

7. Decide whether to use a photorealistic environment map in your scene. See Environment Maps.

Advanced Rendering Reference

Within the **Scenes** node and the **Tools** node, you can access objects that let you activate and control advanced rendering.

Specific nodes referenced here are:

Scenes > [scene] > Attributes > Advanced Rendering Properties

See Advanced Rendering Properties.

Scenes > [scene] > Displayers > [displayer] > Advanced Rendering Effects

See Advanced Rendering Effects.

Scenes > [scene] > Displayers > [displayer] > Advanced Rendering Effects > Advanced Rendering Materials

See Advanced Rendering Materials.

Scenes > [scene] > Displayers > [displayer] > Preset Material

See Preset Material.

Tools > Rendering Materials

See Rendering Materials.

Advanced Rendering Properties

This node provides parameters for advanced rendering capabilities, which you apply to the entire scene rather than to an individual displayer.

Quality

Controls quality of the final image in a range from 0 (minimum) to 1.0 (maximum), with higher quality requiring longer rendering time. Some effects, such as soft shadows or certain materials, require higher quality settings.

Shadow Override

Sets shadow sharpness for the entire scene, overriding displayer settings for shadow sharpness:

- **Use Displayer Property**: this property is deactivated, leaving the displayer settings in effect.
- **No Shadows**: deactivates all shadows in the scene.

- **Soft Shadows**: gives all shadows in the scene a soft edge.

- **Hard Shadows**: gives all shadows in the scene a hard edge.

Blend Epsilon

Controls blending, in a range from 0 to 1.0, among the multiple rendering modes that are used with advanced rendering. This parameter helps lines and points to blend correctly in the advanced rendering.

Max Inner Render Iterations

Sets a maximum number of recursive iterations for rendering the image. The available range is 0 to 50.

For example, this value controls how many recursive reflections take place in the scene, and how many surfaces a transparent ray can penetrate.

[displayer] > Advanced Rendering Effects

This node becomes available in a displayer when you activate the *Enable Advanced Rendering* property of the scene.

Enable Shadows

When activated, computes shadowing on part surfaces in the displayer.

Shadow Sharpness

On a scale from 0 to 1, ranges from a very soft shadow, to various degrees of softness of shadows, to completely sharp. For information on setting shadow sharpness for the entire scene, see Shadow Override.

Casts Effects

When activated, the geometry in the displayer casts shadows, reflections, and refractions onto surfaces. It also prevents self-shadowing, self-reflection, and transparency.

Advanced Rendering Materials

This node lets you add special rendering effects. The **Advanced Rendering Materials** node allows you to use the color that the displayer uses, for example a constant surface color or a scalar value color, and apply additional effects such as the reflection and refraction of the fresnel. This approach differs from using the **Preset Material** object, which is activated by an entirely separate color mode, **Preset Material**.

You can access the following rendering options in the *Effect* property of this node:

Option	Activated Child Nodes
Default By setting the *Reflection* and *Refraction* values greater than `0`, you can imitate water. For a more realistic rendering of water, use a fresnel-type rendering material (see Rendering Materials).	**Simple** This option imitates OpenGL rendering, particularly when the *Absorption* property is set to the default value of `1.0`. It provides the following settings: *Index of Refraction* The index of refraction (IOR) that is used with the *Refraction* setting, on a range from 1.0 to ∞. The following settings control the percent contributions, along a range from 0 to 1.0, of surface characteristics: • *Absorption*: Surface color • *Reflection*: Reflected light • *Refraction*: Refracted light **Note:** Beginning in v12.06, the absorption, reflection, and refraction values can add up to any sum you want. However, for advanced rendering images prepared in versions prior to 12.06, continue to keep these settings summing to 1.0.
Metallic	**Metallic** Provides the following settings: *Specular Color* This color approximates a mirrorlike finish. It is defined by RGB colors. *Roughness* Controls the roughness of the surface with a numeric parameter ranging from 1.0×10^{-4} to 1.0.
Matte	**Matte** Provides the *Sigma* setting, which controls the roughness of the surface, on a range from 0 to 90 degrees. Lower values give the appearance of a smoother surface, while higher values result in a rougher surface.

Water/Glass	**Water/Glass** For the effect of a fresnel (water or glass-like) material, provides the *Index of Refraction* setting, on a range from 1.0 to ∞.

Preset Material

This object gives access to predefined materials that you can save and quickly apply to a displayer. The materials are represented by objects in the **Tools** > **Rendering Materials** node. The **Preset Material** node makes it easier to simulate metals without an RGB color, such as gold or copper, and to record a physically accurate index of refractions.

The *Material* property of this node lets you select rendering materials from a drop-down list.

The **Preset Material** node becomes available in a displayer when the *Color Mode* property of that displayer is set to **Preset Material**. When activated, it hides the **Advanced Rendering Effects** > **Advanced Rendering Materials** node. This option is not available for the scalar displayer, which does not have a *Color Mode* property.

Rendering Materials

This manager node within the **Tools** node contains the supplied material effects with which you can set up your advanced rendering. You can also create your own materials.

Note: These materials also work in the default OpenGL mode.

The **New Rendering Material** submenu lets you add a material from among the following types:

- **Simple**: material simulating simple rendering shading, with options of adding reflection or refraction effects.
- **Fresnel**: material simulating fresnel properties, such as water or glass. This option produces refraction and reflection based on the direction of the rays.
- **Metal**: material that realistically simulates metal appearance, such as gold or copper.
- **Simple Metal**: material simulating metal with user-defined RGB color.
- **Matte**: material with a diffusely lit appearance (no specular component), such as rubber.

Properties Lookup

The following table shows which properties are used by which material type.

Note: Beginning in v12.06, the absorption, reflection, and refraction values can add up to any sum you want. However, for advanced rendering images prepared in versions prior to 12.06, continue to keep these settings summing to 1.0.

	Simple	Fresnel	Metal	Simple Metal	Matte
Absorption Percent contribution of surface color (range 0 to 1.0).	✓				

	Simple	Fresnel	Metal	Simple Metal	Matte
Absorption Coefficient Coefficients for red, blue, and green wavelengths.			✓		
Diffuse Color Enter RGB values for the base body color.	✓			✓	✓
Index of Refraction The index of refraction (IOR) that is used with the *Refraction* setting, on a range from 1.0 to ∞.	✓	✓	✓		
Reflected Color This color tints the reflected ray.		✓			
Reflection Percent contribution of reflected light (range 0 to 1.0).	✓				
Refraction Percent contribution of refracted light (range 0 to 1.0).	✓				
Roughness Controls the roughness of the surface with a numeric parameter ranging from 1.0 x 10^{-4} to 1.0.			✓	✓	
Sigma Controls the roughness of the surface, on a range from 0 to 90 degrees.					✓

	Simple	Fresnel	Metal	Simple Metal	Matte
Specular Color This color approximates a mirrorlike finish. However, it is defined by RGB colors, as opposed to photorealistic materials which do not use RGB.	✔			✔	
Transmitted Color This color tints the refracted ray.		✔			

Additional Rendering Materials - Values

The rendering materials that are listed in this table are not included in STAR-CCM+ by default. To add these materials to your simulation:

1. Right-click the **Tools** > **Rendering Materials** node and select **New Rendering Material** > **Metal**.
2. Rename the new **Metal** node to indicate your choice of material.
3. Enter the corresponding values for the material in the *Index of Refraction* and *Absorption Coefficient* properties.

Additional material information is available at the RefractiveIndex.INFO database: https://refractiveindex.info/.

Metals/Compounds	Index of Refraction	Absorption Coefficient
Aluminium (Al)	(1.65394092, 0.878498793, 0.520122647)	(9.20430565, 6.25620461, 4.82674885)
Aluminium arsenide (AlAs)	(3.59736872, 3.22613931, 2.21235132)	(0.000665507279, -0.000498788548, 0.00740946969)
Aluminium antimonide (AlSb)	(4.70830059, 4.05196238, 4.57418489)	(-0.0291775465, 0.0934454054, 1.29783976)
Amorphous silicon (a-Si:H)	(5.59115076, 4.77824545, 4.89232111)	(0.227737248, 0.617842078, 1.605533)
Beryllium (Be)	(4.17617416, 3.1783011, 2.77819276)	(3.82729554, 3.00373626, 2.86292768)
Caesium Iodide (CsI)	(2.14034843, 1.69870293, 1.65889668)	(0, 0, 0)
Copper Oxide (CuO)	(3.24062014, 2.44260335, 2.20555592)	(0.519990921, 0.568822145, 0.720639527)
Diamond (d-C)	(2.90008092, 2.29322004, 2.22275257)	(0, 0, 0)

Metals/Compounds	Index of Refraction	Absorption Coefficient
Mercury (Hg)	(2.39383841, 1.43696785, 0.907622635)	(6.31419611, 4.36266136, 3.41453838)
Iridium (Ir)	(3.07986569, 2.07777429, 1.61445522)	(5.58028269, 4.05854368, 3.2603364)
Potassium (K)	(0.0639137104, 0.0463115685, 0.0381032713)	(2.09975433, 1.34607041, 0.911276996)
Lithium (Li)	(0.265251547, 0.195191294, 0.220449701)	(3.53304243, 2.30618262, 1.66505289)
Magnesium Oxide (MgO)	(2.08521605, 1.64720988, 1.59149516)	(4.44628362e-012, -3.69021567e-012, 2.56077087e-011)
Molybdenum (Mo)	(4.47417498, 3.51799154, 2.77018428)	(4.10238791, 3.41361785, 3.1439271)
Sodium (Na)	(0.0601384044, 0.0560220294, 0.061859671)	(3.17254162, 2.10799718, 1.57574701)
Sodium chloride (NaCl)	(1.85314035, 1.46401298, 1.41509354)	(0, 0, 0)
Niobium (Nb)	(3.41287279, 2.78427768, 2.39050436)	(3.43407393, 2.73183012, 2.57445478)
Rhodium (Rh)	(2.58031034, 1.85623479, 1.55113983)	(6.76788616, 4.69296551, 3.96766329)
Silicon carbide (SiC)	(3.16561365, 2.52060437, 2.47410822)	(7.26809503e-007, -6.84328938e-007, 9.99246458e-006)
Silicon monoxide (SiO)	(2.3598175, 1.89618707, 1.89423704)	(0.0118285827, 0.023843335, 0.0780569166)
Tantalum (Ta)	(2.05820084, 2.38801837, 2.6225028)	(2.40293002, 1.73766804, 1.9429121)
Tellurium (Te)	(7.37519979, 4.47256279, 2.63149166)	(3.24919248, 3.51991487, 3.28520942)
Titanium oxide (TiO2)	(3.44928217, 2.79576182, 2.89899373)	(0.000102369493, -8.94818368e-005, 0.000633714546)
Vanadium (V)	(4.26843071, 3.50570178, 2.75527763)	(3.48376703, 2.88321757, 3.1051085)
Tungsten (W)	(4.36142588, 3.29329371, 2.99190831)	(3.49324656, 2.59933424, 2.2683773

Saving and Printing Scene Data

In STAR-CCM+ you can produce output of the contents of a scene display in a variety of ways.

The options are:

Contents:

Exporting an Interactive 3D Scene to STAR-View+ or VRML
Saving Scene Image Files
Saving Scene Images in Batch
Printing a Scene

Exporting an Interactive 3D Scene to STAR-View+ or VRML

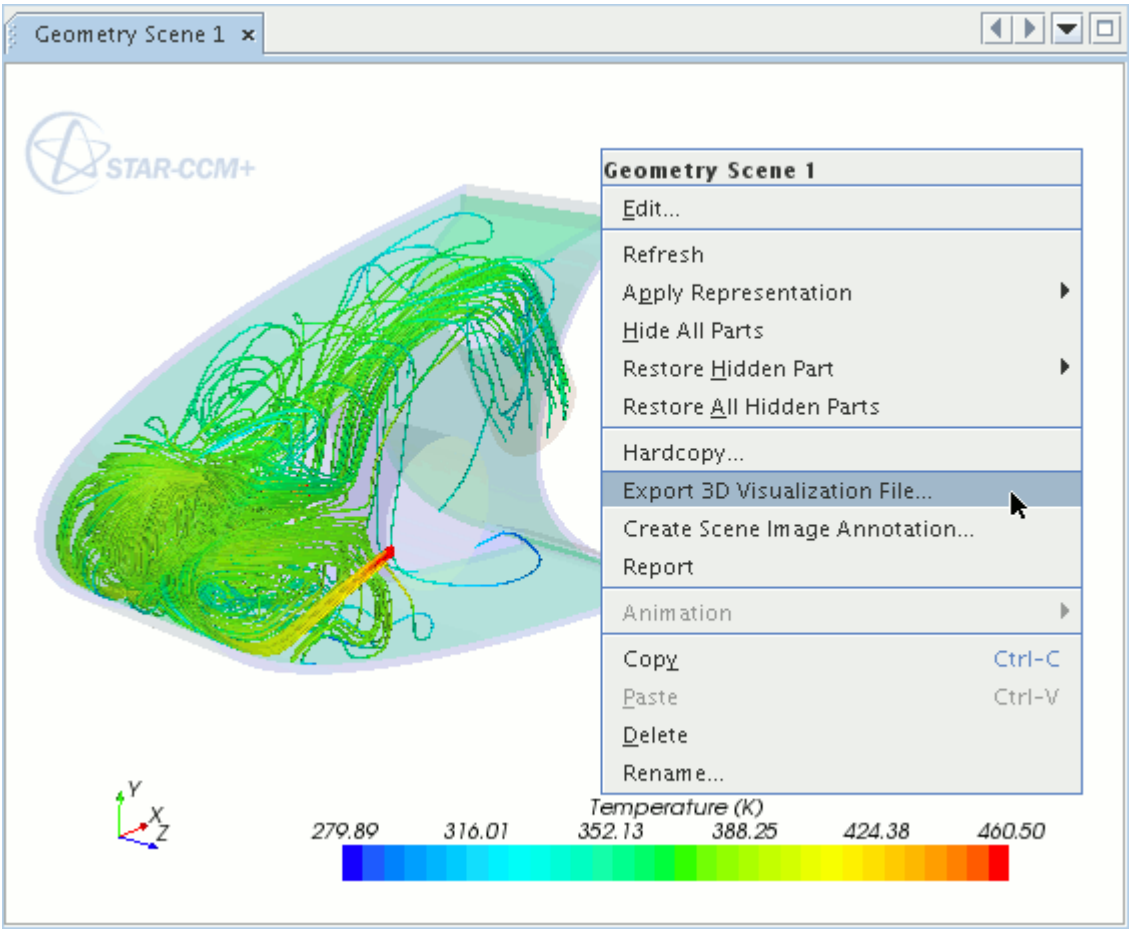
You can export the contents of a scene to a `.sce` or VRML file from STAR-CCM+.

You can open the `.sce` file in STAR-View+ and work with the same three-dimensional image that you had in STAR-CCM+. The VRML file can be viewed using a suitable third-party viewer such as Cortona3D Viewer. In both cases you can interact with the image to rotate, pan, and zoom.

Note: If you are exporting a scene that includes sweep animation, you must first activate the *Precompute Steps* option on the animation control node for the relevant parts. See Sweeping Parts Animation.

To save the scene contents to these formats:

1. Right-click the scene node or the scene display, and select **Export 3D Visualization File...** from the pop-up menu.

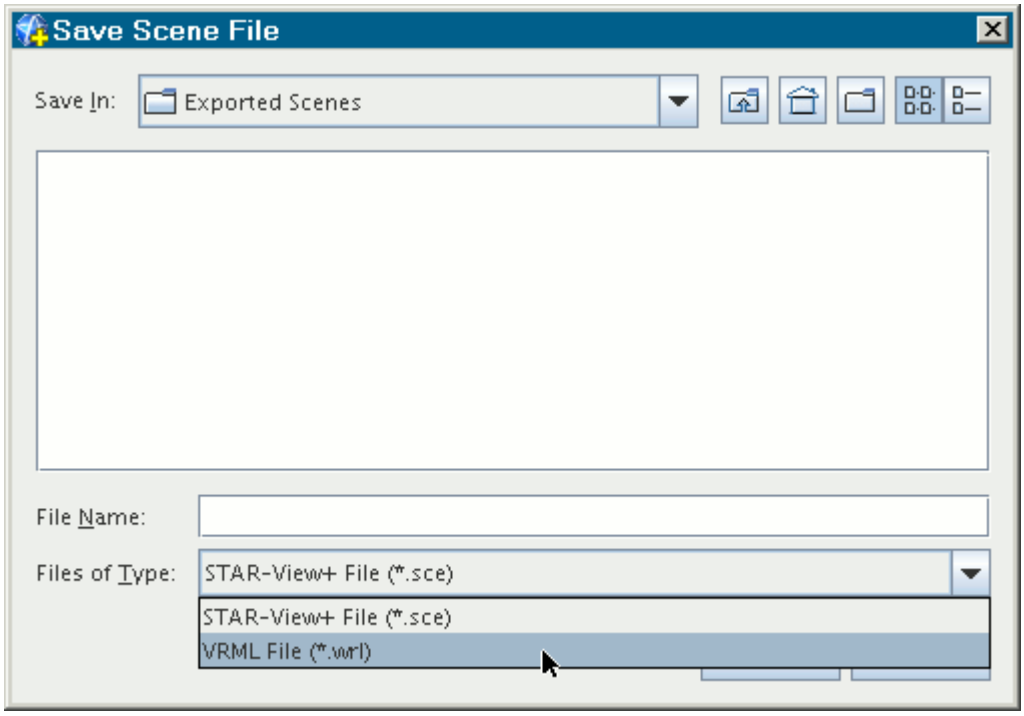


Note: This option is deactivated if more than one scene node is selected.

In the *Save Scene File* dialog, the **Files of type:** filter is used to select the format of the saved file. This filter is set to the **STAR-View+ (*.sce)** format by default.

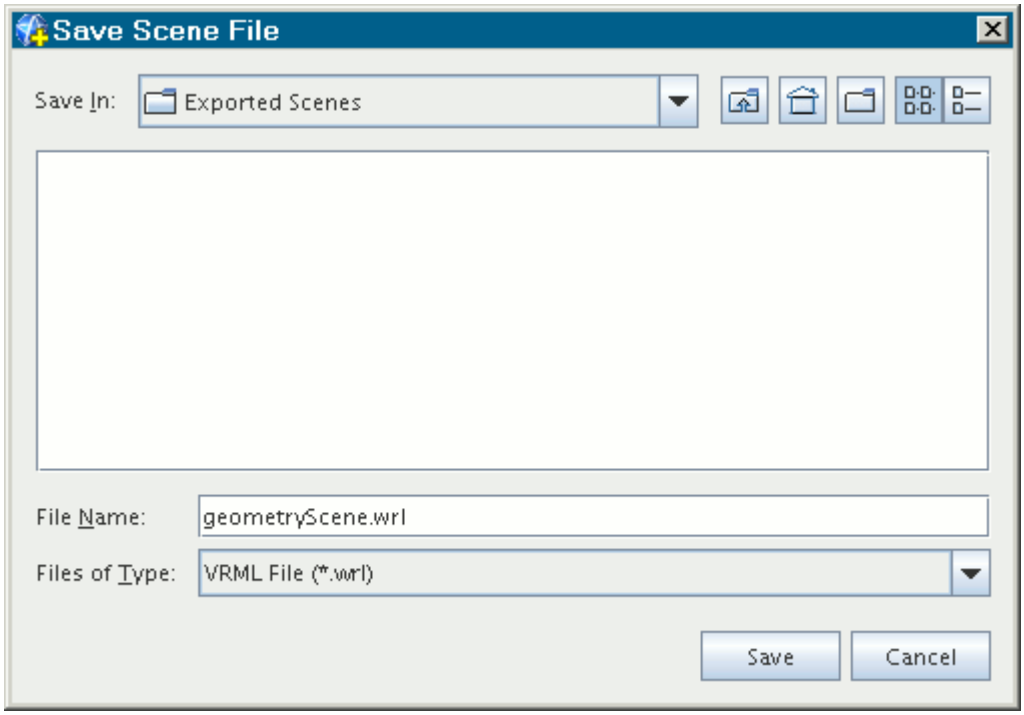
To export the scene as a VRML file:

2. Select the **VRML (*.wrl)** format from the drop-down list.



Note: Volume-rendered data cannot be exported to VRML format. If you export a scene that contains a resampled volume part, the scene export process skips that part.

3. Navigate to the location where you want to save the file, enter the file name in the File Name: text box, and click **Save**.



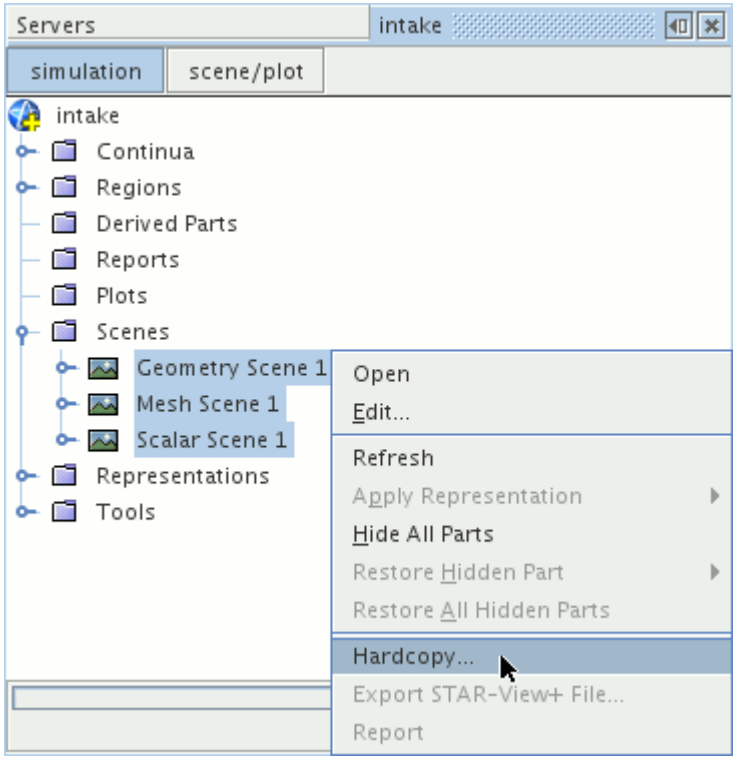
Saving Scene Image Files

The **Hardcopy...** pop-up menu item lets you save one or more scenes to image files.

To save a scene to an image file:

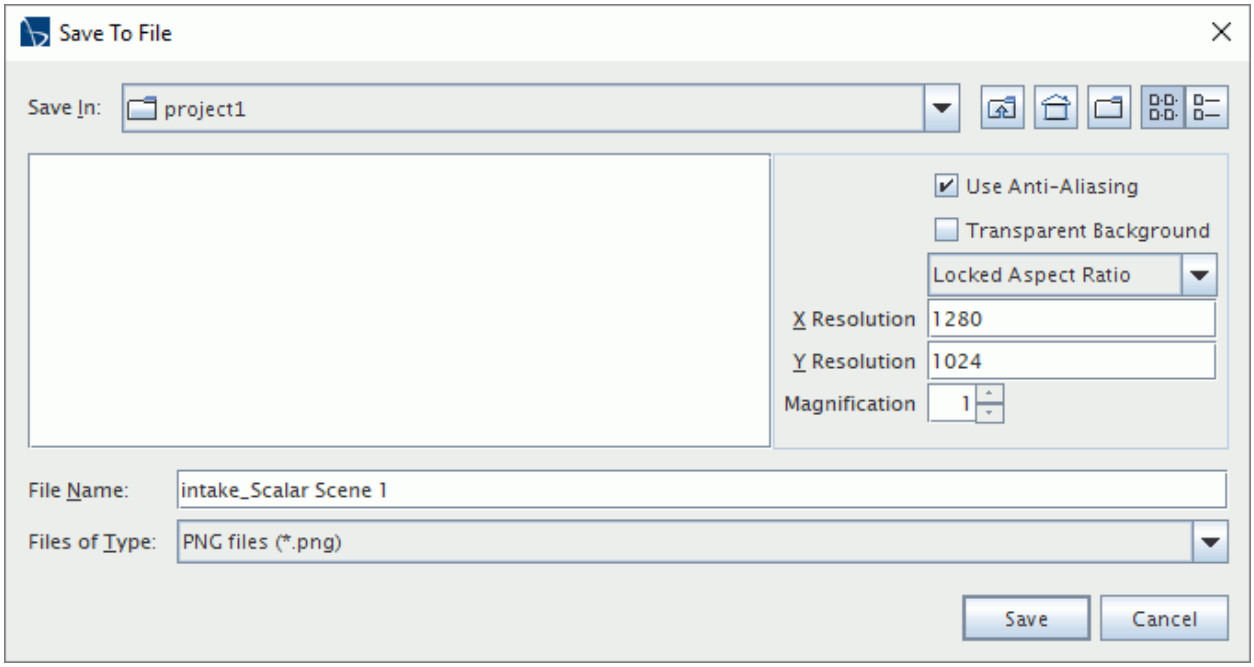
1. Right-click the scene node and select **Hardcopy...**

   To save multiple scenes use `<Shift><Click>` or `<Ctrl><Click>` to select the scene nodes, then right-click one of the selected nodes and select **Hardcopy...**.



   This pop-up menu also appears if you right-click the background of a scene in the *Graphics* window, from which a single image file of the current scene can be saved.

2. Use the *Save To File* dialog to save the selected scene(s) to file:



a) Enter a name in the **File Name** text box.

b) Select any of the following formats for the image file from the **Files of Type** drop-down list:

- Portable network graphics (`.png`)
- Bitmap (`.bmp`)
- Netpbm (`.pnm`)
- Joint Photographic Experts Group (`.jpg`)
- PostScript (`.ps`)

c) The **Use Anti-Aliasing** option, when activated, reduces distortion in the rendered image at the expense of higher processing time. This option is recommended for hardcopy output.

d) To generate an image with a transparent background, activate the **Transparent Background** option. This option is available for `.png` files only.

e) To set specific values for the scene image resolution, in pixels, use one of the two available options in the drop-down list:

- **Locked Aspect Ratio** -- enter a value for the **X Resolution** or **Y Resolution**, and STAR-CCM+ computes the other one to preserve the aspect ratio.
- **Custom Resolution** -- enter any value up to `4096` for either resolution and the aspect ratio is ignored. This choice can produce unexpected viewports and cutoffs of images.

f) To enlarge the picture by an integer value, use the **Magnification:** text box.

(STAR-CCM+ ignores fractional and negative entries by reading them as 1.)

A magnification value of 1 means the size and resolution of the resulting file will be the same as that of the screen (typically 72 dpi), so this is essentially a screenshot of what you see in the visualization display. A larger magnification creates a larger file at the same resolution, giving the effect of a higher resolution image if the file is imported into a document and scaled down. For example, suppose you want an image approximately 300 dpi, suitable for printing in an office document. A magnification of 4 would produce an image four times as large as the original, still at 72 dpi. However, by specifying a

25% scale when importing the image into a presentation program or image editor, you can obtain an effective resolution of 288 dpi.

3. Click **Save**.

   When you save multiple scenes, the settings specified in the *Save To File* dialog are applied to all scenes. The image files are saved under the name provided in the **File Name** text box and numbered incrementally according to their position in the simulation tree. In the example shown in the above screenshots, the image files would be named `intake-scene_0.png`, `intake-scene_1.png`, and `intake-scene_2.png`, which correspond to **Geometry Scene 1**, **Mesh Scene 1** and **Scalar Scene 1** respectively.

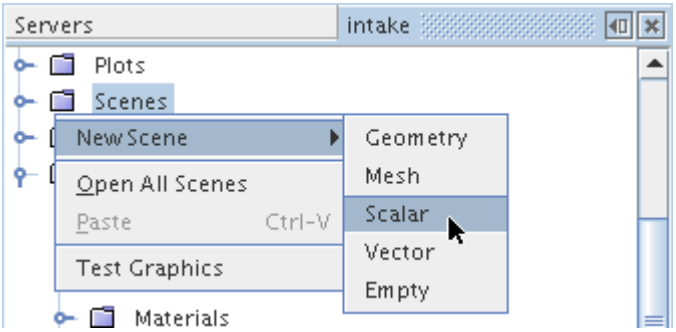
   The saved image(s) can then be incorporated into a printed report, a slide presentation, a Web page or another commonly used medium.

Saving Scene Images in Batch

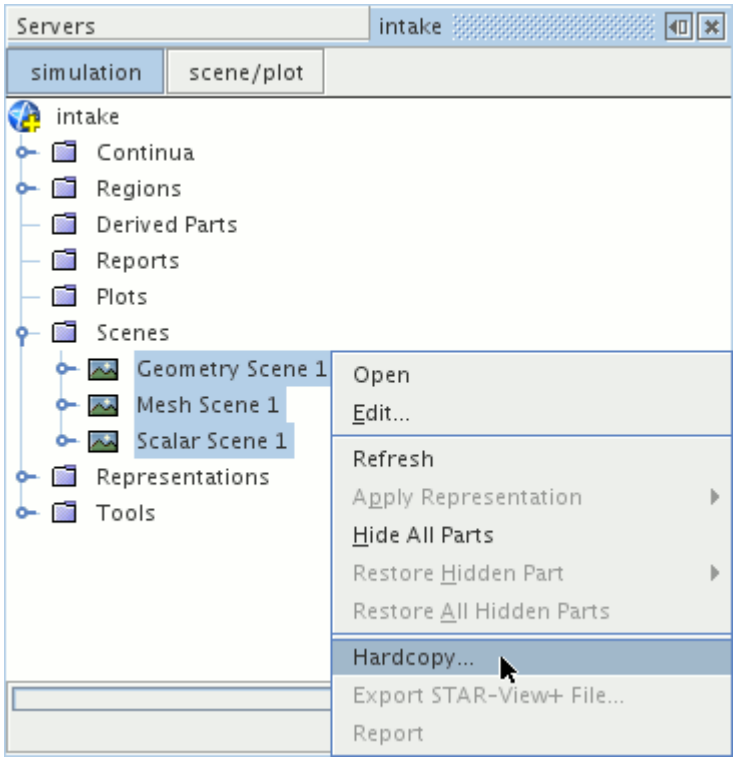
You can generate scene images offscreen, that is, without having the STAR-CCM+ workspace active on your monitor.

This option may be necessary if you are running the client remotely, and/or if you need to generate a large number of these images. STAR-CCM+ lets you do this easily by running a macro in batch.

1. Start recording a macro. See Recording the Macro.
2. Create a new scene. For example, to create a new scalar scene, right-click on the **Scenes** node and select **New Scene** > **Scalar** from the pop-up menu.



3. Right-click on the scene node and select **Hardcopy...**as shown below.



4. Save (stop recording) the macro.
5. Run the macro in batch mode:
   - Batch mode is run from the command line using the `-batch` option:

     ```
     % starccm+ -batch macroname.java
     ```

   - If the simulation file is not open:

     ```
     % starccm+ -batch macroname.java simulationname.sim
     ```

   If all clients are disconnected from a running server, no graphics scenes will be saved to files.

Printing a Scene

You can print scene images on paper.

To send a scene image to a printer connected to your system, select the scene node, then use the **File** > **Print** menu item to invoke the print action.

Controlling Display Formats

You can make a wide range of precise changes to the appearance of text and numbers in scenes, using either a dialog or the number format.

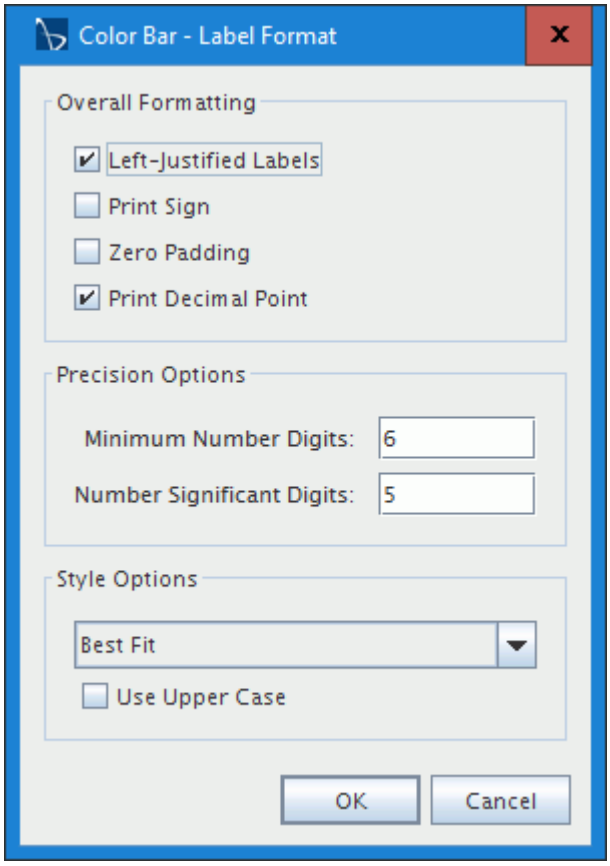
Controlling Formats with the Dialog

A dialog lets you make changes to text and numeric labels that are displayed in scenes. To work with the dialog:

- Click ... (**Custom Editor**) in any of the following:
  - The *Label Format* property of the **Color Bar** node in scene displayers
  - The *Default Solution Time Format* property of the **Iteration** and **Solution Time** annotations
  - The *Number Format* property of the report annotation

  Note: When you add the **Iteration** or **Solution Time** annotation to a scene, the applied annotation node in the scene has a *Solution Time Format* property. This property lets you customize the annotation format for the particular scene.

  This action activates the format dialog. The following example works with the color bar, so its title is *Color Bar - Label Format*.



- To adjust the appearance of the labels, use the options in the *Overall Formatting* group box:
  - To justify the labels toward the left end of the color bar, activate the **Left-Justified Labels** checkbox.
  - To indicate whether the values are positive or negative, activate the **Print Sign** checkbox.
  - To include zeros at the left of the labels (for a uniform number of digits), activate the **Zero Padding** checkbox.
  - To display decimal points for all numbers, including integers, activate the **Print Decimal Point** checkbox.
- To specify the minimum number of digits overall, and the number of digits after the decimal point, use the text boxes in the *Precision Options* group box. Press `<Enter>` after making an entry in each of these text boxes.
- In the *Style Options* group box, select a style from the drop-down list:
  - Best Fit - this default option presents the data in a visually optimal manner.
  - Decimal - this option displays the full decimal values based on the entry in the **Number Decimal Digits** text box.
  - Scientific - this option displays the values in scientific notation. To use an uppercase “e”, activate the **Use Upper Case** checkbox.

What Is the Number Format?

The *Number Format* property of the report annotation, which is similar to the *Default Solution Time Format* property of the **Iteration** and **Solution Time** annotations, is used to convert the value to a suitable display format.

Note: When you add the **Iteration** or **Solution Time** annotation to a scene, the applied annotation node in the scene has a *Solution Time Format* property. This property lets you customize the annotation format for the particular scene.

The property is entered as text and should follow the pattern:

```
%[flags][width][.precision]specifier
```

where the elements in brackets are optional, and the specifier is a character that indicates the style of format required. The table below provides the accepted values for each element.

For report annotations, the measurement unit appearing on the display is defined by the report itself.

flags	An optional element that controls some overall format behavior.	
	-	An optional character that will cause the converted value to be left-justified within the width of the output.
	+	An optional character that will place + or − in front of the output value.

	0	An optional zero that will pad unused digits in front of the output with 0's.
	#	An optional character that will cause the decimal point to be printed even when no nonzero digits follow.
width	The minimum number of characters to be printed in the output value. If the output value has fewer characters than the field width, it will be blank-padded on the left (or right, if the − flag has been given) to make up the width. If the `0` flag has been given, 0's will be used for padding instead of blanks.	
.precision	An optional period followed by a number that specifies the number of digits to appear after the decimal point. For example, `.3` would mean that three digits will appear after the decimal point in the output value.	
specifier	A character that indicates the style of output required.	
	f	The value will be output in the form [-]m.ddd where the mantissa m will have as many digits as it needs, and the number of d's after the decimal point is given by the precision. Six d's will be used if the precision has not been set.
	e	The value will be output in scientific notation, that is [-]m.ddde +xxx or [-]m.ddde-xxx. The mantissa, m, consists of one digit, and the number of d's is set by the precision (six if not given).
	E	Same as e, but with an uppercase E to identify the exponent.
	g	Use either f or e depending on which style will give the greatest precision in minimum space.
	G	Same as g, but with an uppercase E used to identify the exponent (if present).

Some examples of the effect of these formats on a report value are as follows:

Report value = `467699.65625 Pa`

Number Format	Output	Comment

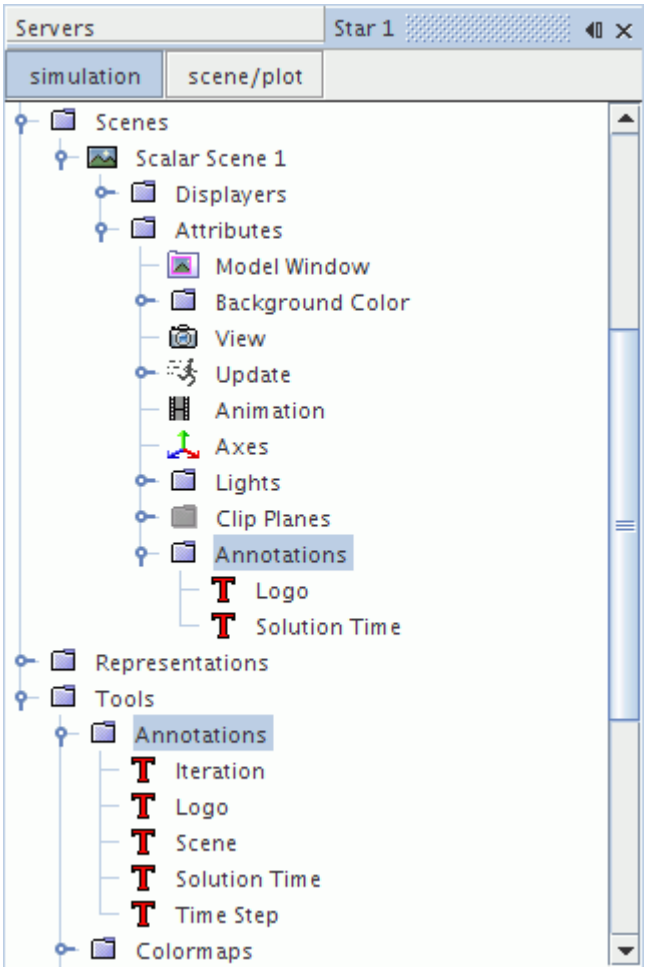
`%f`	`467699.656250 (Pa)`	Default number of d's after decimal point (six).
`%6f`	`467699.656250 (Pa)`	Width of 6 has no effect as more than six digits are present
`%10.2f`	`467699.66 (Pa)`	One blank inserted at start to pad number to full width of 10 characters.
`%+012.2e`	`+004.68e+005 (Pa)`	+ symbol inserted for positive number; 0's used to pad output to full width of 12 characters.
`%8.3E`	`4.677E+005 (Pa)`	Three digits after decimal point; no padding required.

Annotations

An annotation is an object that you can use to add information to the scene display in the *Graphics* window.

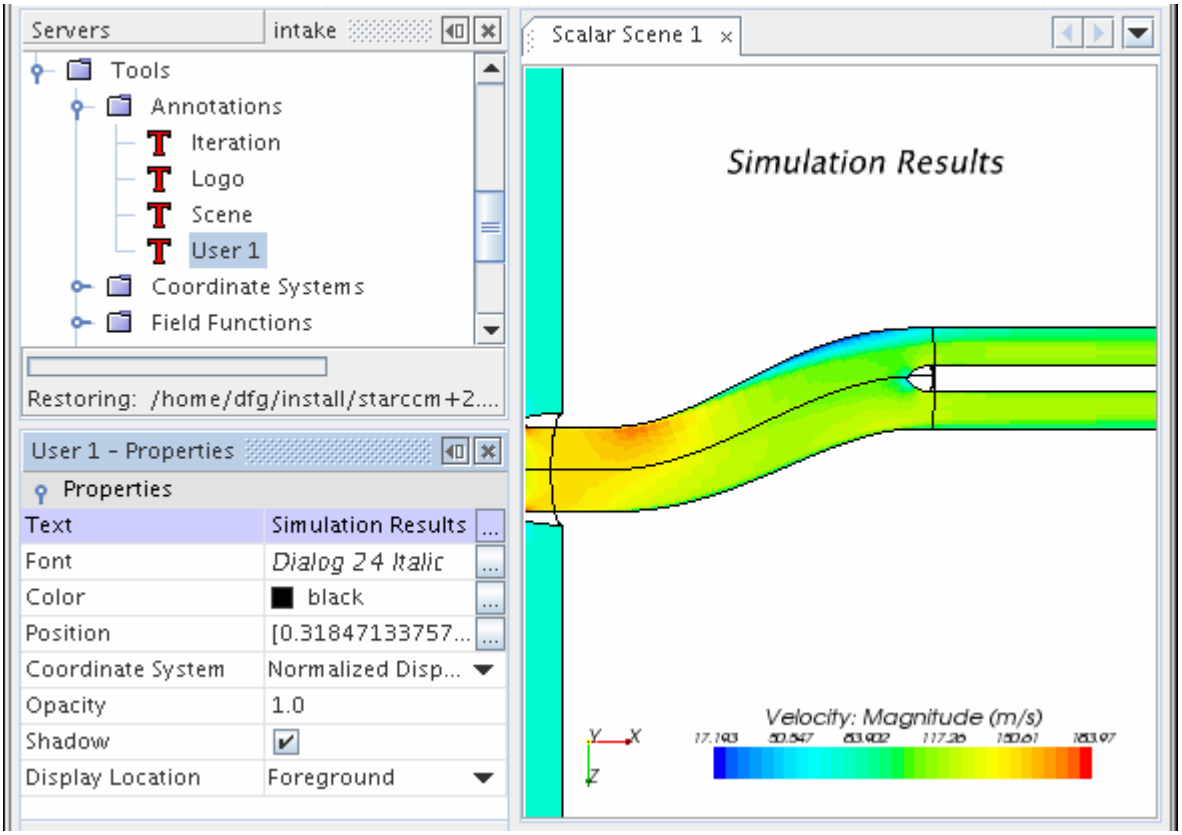
Five annotations named **Iteration**, **Logo**, **Scene**, **Solution Time**, and **Time Step** are created by default with the simulation, and you can create new annotations with the pop-up menu of the main annotations manager. The properties of an annotation become active when you select its node.

The **Annotations** manager node contains template annotations. When you apply one of these annotations to a scene, a copy of the original node is created, and stored within the equivalent manager node belonging to the scene. Scenes can have their own annotation objects that you can customize separately from the original annotations. These types of annotations are called applied annotations, and they are stored in the **Attributes** > **Annotations** node within the scene node. For more information, see [What Are Applied Annotations?](#).

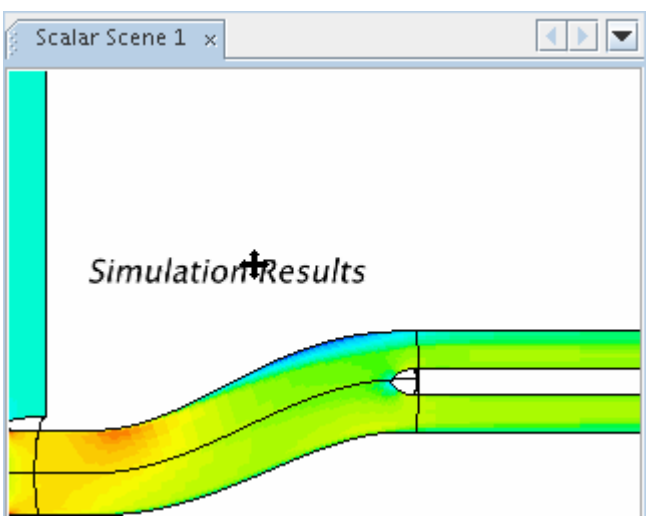


Note: The properties of the applied annotation (located in the **Scene** > **Attributes** > **Annotations** node) are different from the properties of the template annotation (in the **Tools** > **Annotations** node). The properties of the applied annotation override the properties of the original annotation.

You can add annotations to any number of scenes, where they may be placed anywhere in the *Graphics* window. You can change the properties of the text such as the size, font, and color.



You can use the mouse to relocate an annotation in a scene, provided that the annotation has the *Interactive* expert property and that property is activated.



You may add user-defined annotations or default annotations where the text is pre-defined. Pre-defined annotations are provided to let you use common and useful annotations more easily.

The process by which a scene accesses an annotation is very similar to the way a scene's part displayer accesses parts. Annotations are stored in a manager separate from the scenes just as parts are stored separately as boundaries, regions or derived parts. Also, annotations can be added and removed through the use of an object selector. See The Object Selection Dialog.

All user-defined annotations can be renamed. Pre-defined annotations cannot be renamed.

Contents:

Annotations Property Reference
Annotations Workflow
Simple Text Annotation
Image Annotations
Iteration, Time-Step, and Solution Time Annotations
Scene Annotation
Report Annotation
Plot and Scene Image Annotations
Scene Grid Annotation
Working with the Derived Data Annotation

Annotations Property Reference

Annotation Names and Abbreviations

2D Image	**2DI**
3D Image	**3DI**
Background	**BG**
Derived Data Annotation	**DDA**
Iteration	**I**
Logo	**L**
Plot Image	**PI**
Report	**Rp**
Scene	**S**
Scene Grid	**SG**
Scene Image	**SI**
Simple Text	**STx**
Solution Time	**STi**
Time-Step	**TS**

Properties Lookup

The following table shows which properties are used by which annotation. Use the abbreviations given above.

	2 DI	3 DI	B G	D D A	I	L	PI	R p	S	S G	SI	S T x	S Ti	T S
Background When activated, displays the text with a colored background box.				✓	✓			✓	✓			✓	✓	✓
Background Color The color of the background text box (visible when the **Background** checkbox is activated), which can be selected from a drop-down list. The property customizer executes the define color action. See Defining Colors.				✓	✓			✓	✓			✓	✓	✓
Color The color of the text, which can be selected from a drop-down list. The property customizer executes the define color action. See Defining Colors.				✓	✓			✓	✓			✓	✓	✓
Date When activated, includes the date and time (timestamp) of the last save in the scene information.									✓					
Default Height The default character height as a fraction of the visualization display height, where 1.0 equals the full height.	✓	✓		✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
Default Position The default position of the text in the visualization display based on the coordinate system specified in the *Coordinate System* property.	✓	✓		✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
Default Scaling When activated, scales the elements of a plot image (such as font and line thickness) based on the overall size of the plot image annotation as it appears in the scene.							✓							

	2 DI	3 DI	B G	D D A	I	L	PI	R p	S	S G	SI	S T x	S Ti	T S
Default Solution Time Format Defines the numeric formatting of the solution time value, using either a text entry or a dialog. For details, see Controlling Display Formats.					✓								✓	
Default Width The default width of the annotation as a fraction of the visualization display width, where 1.0 equals the full width.		✓					✓				✓			
Default Width Stretch Factor The default amount by which the annotation is stretched in the horizontal direction.	✓			✓	✓	✓		✓	✓			✓	✓	✓
Derived Data The derived data object, which is selected from a drop-down list.				✓										
Font The font and style of the text. These settings can be entered directly, or you can click the property customizer to activate a dialog. To change the point size, use the *Default Height* property.				✓	✓			✓	✓			✓	✓	✓
Grid Color The color of the grid, which can be selected from a drop-down list. The property customizer executes the define color action. See Defining Colors.										✓				
Line Thickness Controls the grid line width in pixels.										✓				
Number Format Defines the numeric formatting of the value associated with the report, using either a text entry or a dialog. For details, see Controlling Display Formats.								✓						

	2 DI	3 DI	B G	D D A	I	L	PI	R p	S	S G	SI	S T x	S Ti	T S
Opacity Adjusts the transparency of the annotation. The settings range from 0.0 (completely invisible) to 1.0 (completely opaque). This option allows you to use the annotation as an overlay.	✔	✔	✔	✔	✔	✔		✔	✔	✔	✔	✔	✔	✔
Path The relative path to the imported graphic. The formats that can be imported are PNG, JPEG, TIFF, PNM, and BMP.	✔	✔	✔			✔								
Plot Selects a plot from a drop-down list of the existing plots in the simulation.							✔							
Precision Controls the grid coordinate precision. The default value is 3; this will display three digits after the decimal place.										✔				
Product Name When activated, includes the name of the program in the scene information.									✔					
Product Version When activated, includes the version of the program in the scene information.									✔					
Report The name of the associated report (read-only).								✔						
Save to Simulation When activated, retains imported images in the simulation file.	✔	✔	✔			✔								
Scene Selects a scene from a drop-down list of the existing scenes in the simulation.											✔			

	2 DI	3 DI	B G	D D A	I	L	PI	R p	S	S G	SI	S T x	S Ti	T S
Shadow When activated, displays the text with a shadow effect.				✓	✓			✓	✓			✓	✓	✓
Simulation When activated, includes the simulation title in the scene information.									✓					
Spacing Mode Selects the method for setting the grid spacing. • **Display**: Sets the grid spacing in display units. Activates the **Display Spacing** subnode. See What Is the Display Spacing? • **Model**: Sets the grid spacing in model units. Activates the **Model Spacing** subnode. See What Is the Model Spacing?										✓				
Solution Time When activated, reports the solution time in the annotation. This is useful for unsteady simulations.					✓									
Tags Selects a tag from a drop-down list of the existing tags in the simulation.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Text The text of the annotation. The property customizer opens a dialog with a larger space for more text. If you enter the function name of any report prepended by '`$`', STAR-CCM+ will replace this by the numerical value provided by the report. For example, `$InletPressureReport` will return the numerical value given by the report **Inlet Pressure**. For report annotations, displays the title and value of the associated report (read-only). For solution state annotations, displays one of the following values (read-only):				✓				✓				✓		

	2 DI	3 DI	B G	D D A	I	L	PI	R p	S	S G	SI	S T x	S Ti	T S
• number of iterations • number of time-steps • elapsed physical time • current completed time-step For derived data annotations, displays the following (read-only): • For time histories, the text displays the name of the derived data object, the display time, and the name of the field function that is recorded in the history. • For spectra, the text displays the name of the derived data object and object properties and their values, indicating how the data was processed.														
Text Color The color of the text, which can be selected from a drop-down list. The property customizer executes the define color action. See Defining Colors.										✔				
Time Step When activated, reports the time-step in the annotation. This is useful for unsteady simulations.					✔									
Units Selects the type of units to use.										✔				
User When activated, includes the STAR-CCM+ user name in the scene information.									✔					

Annotation Pop-Up Menu

This pop-up menu becomes available if you right-click an annotation in the scene display or in the object tree.

Remove	Removes the annotation from the active scene.

Annotations Workflow

A variety of options exist for working with the annotations in STAR-CCM+.

To create an annotation:

1. Expand the **Tools** node and right-click the **Annotations** node.
2. Select an annotation from the **New** submenu.

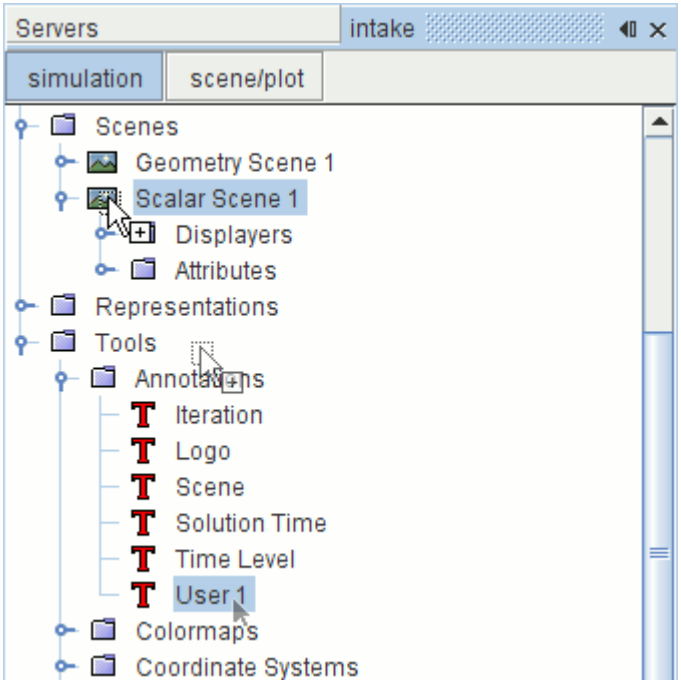
   A new annotation appears under the **Annotations** node.

   Once the annotation has been created, it can be renamed at any time, modified to specify the required text, or added to any scene.

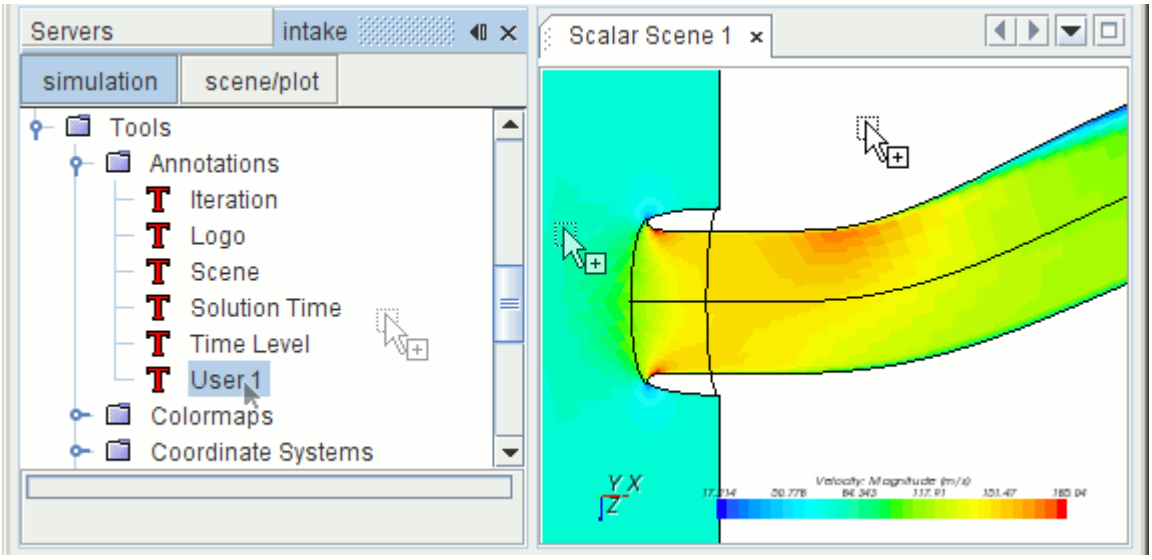
   Note: For plot image and scene image annotations, you can create the annotation and apply it to a scene in one action. Drag and drop the node of the plot or the scene into the visualization display, or onto the node, of the destination scene.

To add an annotation to a scene:

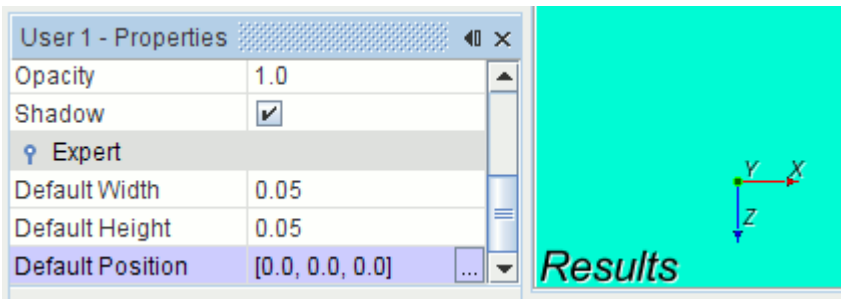
3. Use any of the following techniques:
   - Using drag-and-drop to drag the annotation onto the scene node.



   - Using drag-and-drop to drag the annotation into the visualization display.

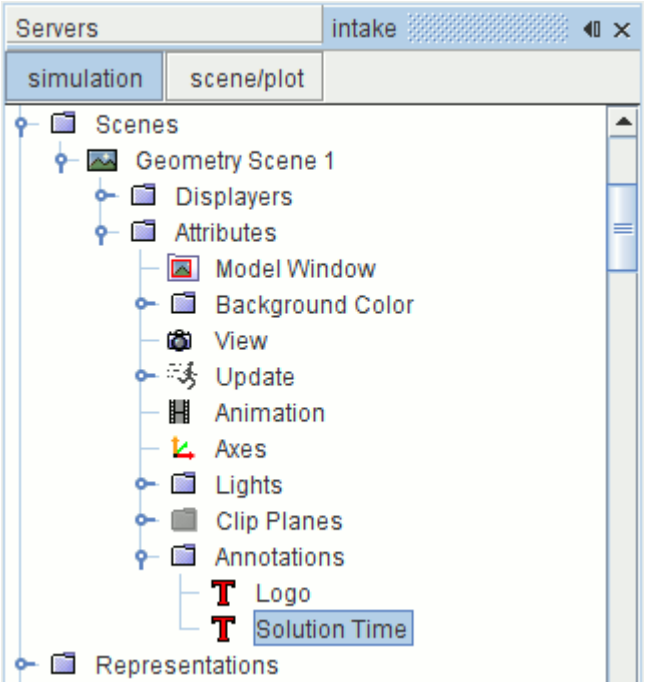


Wherever you drop the annotation in the scene, it will appear at the coordinates specified in the *Default Position* expert property.



- Using the object selection dialog.

After you add the annotation in the scene, it is represented in the object tree by an *applied* annotation -- the object in the scene representing the annotation that you added. (Generally once an applied annotation is in a scene, it can only be changed by modifications to its own properties.) For more information, see What Are Applied Annotations?.



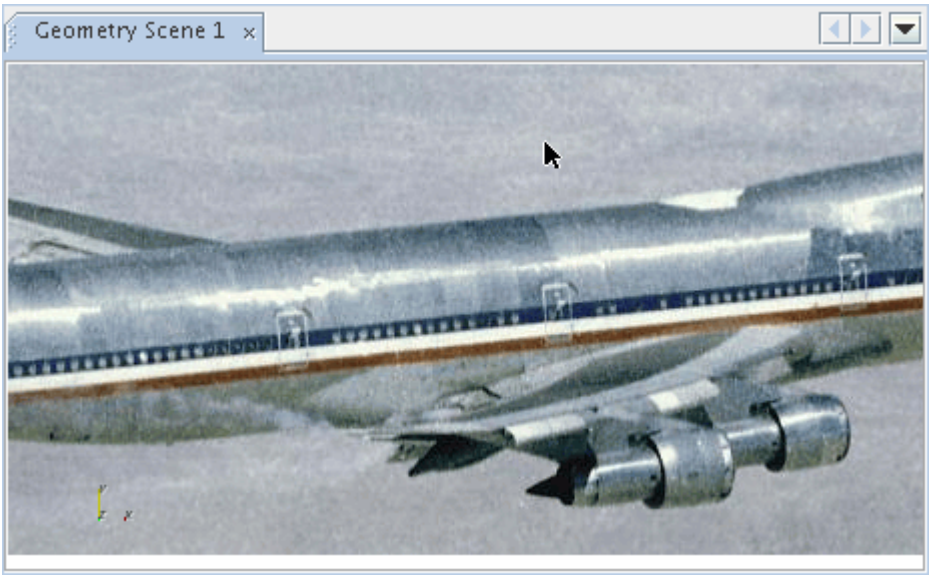
After you add this object to the scene, editing the properties of the original annotation does not affect the applied annotation. You must remove the applied annotation from the scene(s) and then put a revised annotation there. Alternatively, you can edit that scene's applied annotation directly. (The exception is the text content of a simple text annotation, which can only be edited from the original.)

To manipulate a graphic from an annotation:

4. Import a graphic into the annotation by entering the path to the graphic file in the *Path* property.

   Alternatively, click the ... (**Custom Editor**) of that property to open a separate dialog.

5. You can enter coordinates in the *Default Position* property. Alternatively, you can click and drag the image provided the *Interactive* expert property is activated.

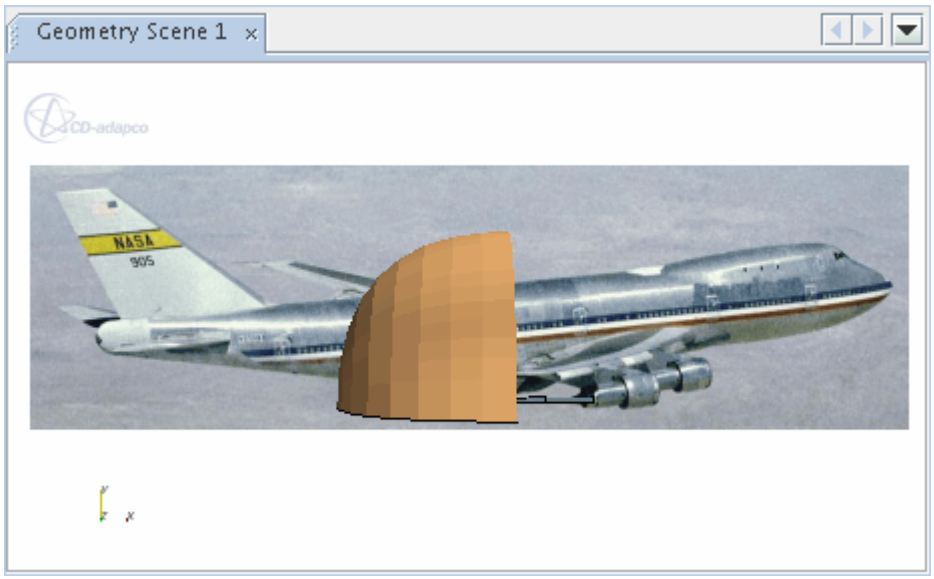


6. To reduce the image, navigate to the properties of the applied annotation of the scene. Enter a decimal fraction in the *Height* property to adjust the size as a percentage of the scene. For example, setting the *Height* to `0.45` scales down the image to 45% of its original size.

   Note: While this property allows you to enlarge the image beyond its original size, this step is not recommended since it may give the image a grainy appearance.

7. To control whether the image appears behind or in front of the model, use the drop-down list in the *Location* property.

   The results appear in the scene display if you have it open. In this illustration, `Background` is selected so that the picture serves as a backdrop for the geometry model.



To manipulate the logo in the display:

8. To import your own logo, enter the path to the graphic file in the *Path* property. Alternatively, click the ... (**Custom Editor**) of that property to open a separate dialog.
9. To restore the STAR-CCM+ logo, clear the path to the graphic in the *Path* property.
10. To shift the annotation's position, you can enter coordinates in the *Position* property.

    Alternatively, you can click and drag the logo provided the checkbox of the *Interactive* expert property is ticked.

11. To adjust the transparency of the annotation, change the value of the *Opacity* property.

    The settings range from 0.0 (completely invisible) to 1.0 (completely opaque).

    To set the default transparency of the annotation in every scene:

    a) Select **Tools** > **Options**.
    b) In the *Options* dialog, select the **STAR-CCM+** > **Visualization** node.
    c) Change the value of the *Logo Opacity* property.
    d) Click **Close**.

12. To control whether the logo appears behind (`Background`) or in front of (`Foreground`) the simulation object(s), use the drop-down list in the *Display Location* property.
13. To change the logo size, enter decimal fractions in the *Height* property to represent percentages of the visualization display height (1.0 equals the full height).
14. Clear the checkbox of the *Interactive* expert property if you want to ensure that mouse movements will not accidentally move the logo.

To create a report annotation:

15. Right-click the relevant report node and choose **Create Annotation from Report**.

    This action adds an annotation node with the same name to the **Tools** > **Annotations** node.

To customize annotations:

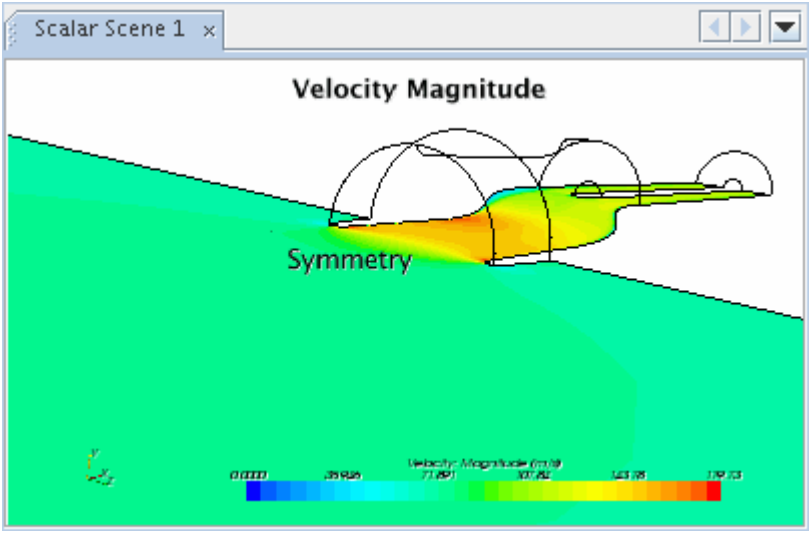
16. To have solution state data (iteration, solution time, or time-step) come from a recorded solution view, update the *Representation* property.
17. To change the position of a particular annotation, use one of the following techniques:
    - Click and drag it within the scene display, provided that the annotation has the *Interactive* expert property and that property is activated.
    - Enter coordinates in the properties of the applied annotation.
      - Enter the position vector directly in the *Position* property.
      - For a 3D image annotation, this type of entry is in two properties: *Position* and *Normal*.
      - Alternatively, click the ... (**Custom Editor**) of that property to open a separate text input dialog.

      For most applied annotations, the position can be entered using a choice of visualization display-specific coordinate systems:

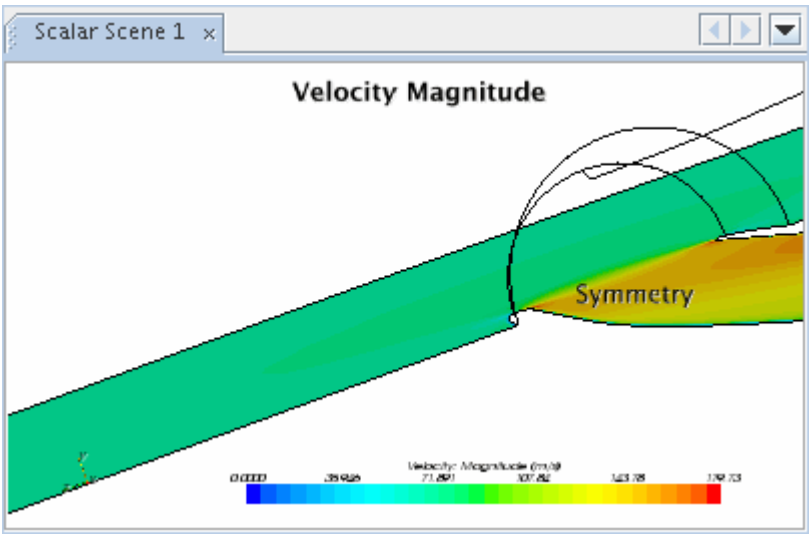
      - Normalized Viewport: x,y (0, 1) normalized values in viewport; use this to attach an annotation to a fixed location in the visualization display.
      - World: x,y,z global coordinate values; use this to attach an annotation to a point in the model space. This will result in text that moves with the attachment point when the view is changed.

      These options can be selected from the drop-down list of the *Coordinate System* property of the appropriate applied annotation.

    In the example below, an applied annotation has been used with the **Normalized Viewport** coordinate system to add a title into the *Scalar Scene 1* display. A label has also been added using the **World** coordinate system.

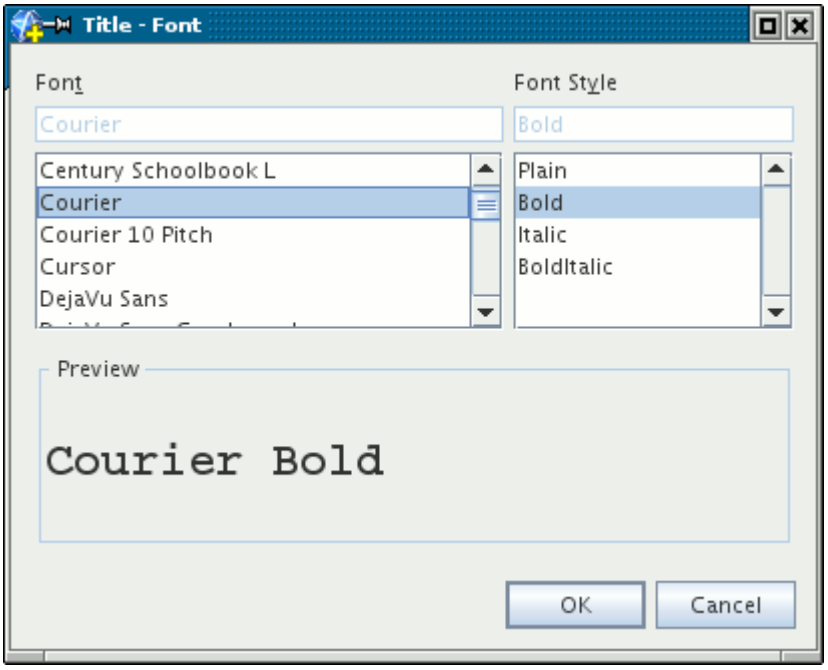


As the scene is manipulated the title remains in the same location, but the label remains near the same part of the symmetry plane as it moves.



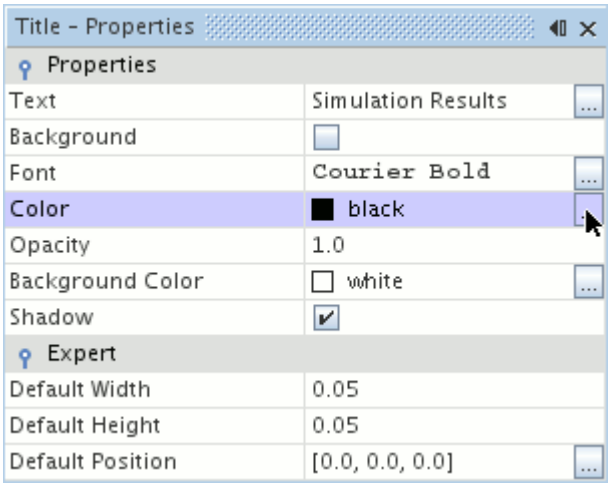
18. To change the font of an annotation:
    a) Select an annotation from the feature tree.
    b) Within the *Properties* window, click the ellipsis button to the right of the *Font* property.

    This action opens the *Font* dialog, where a **Font** and **Font Style** can be selected.

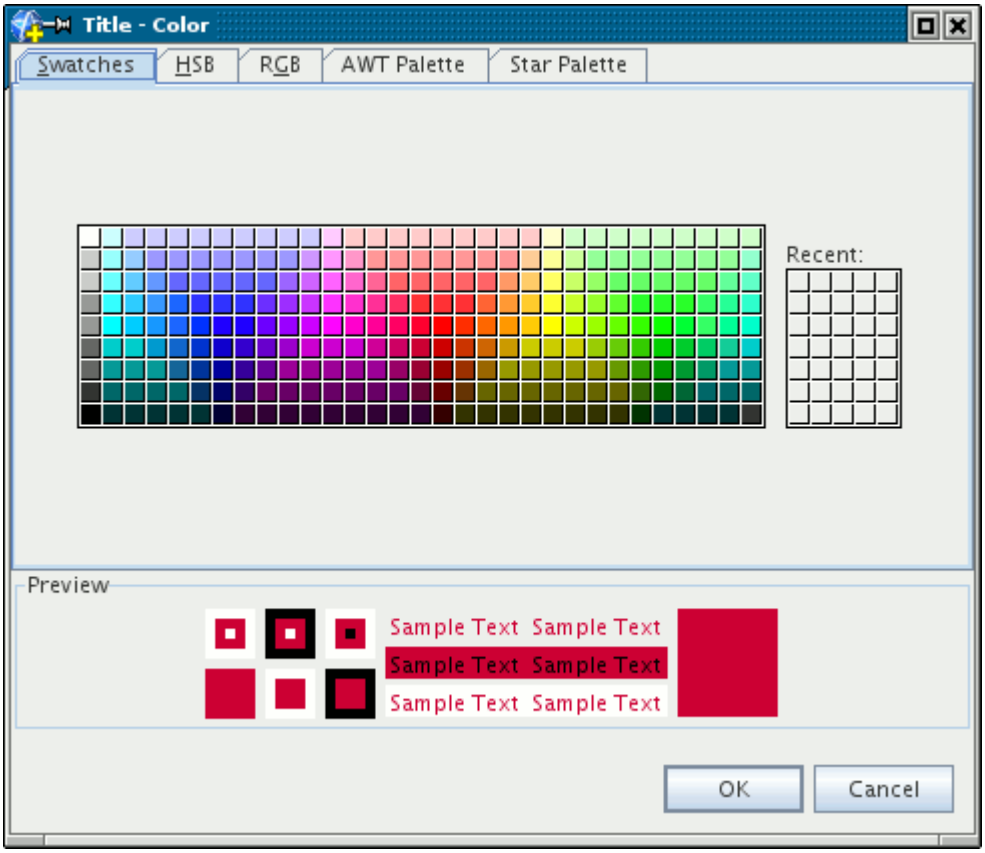


Clicking **OK** will apply the font to the currently selected annotation.

19. To change the font color:
    a) Click the ellipsis button to the right of the *Font Color* property.

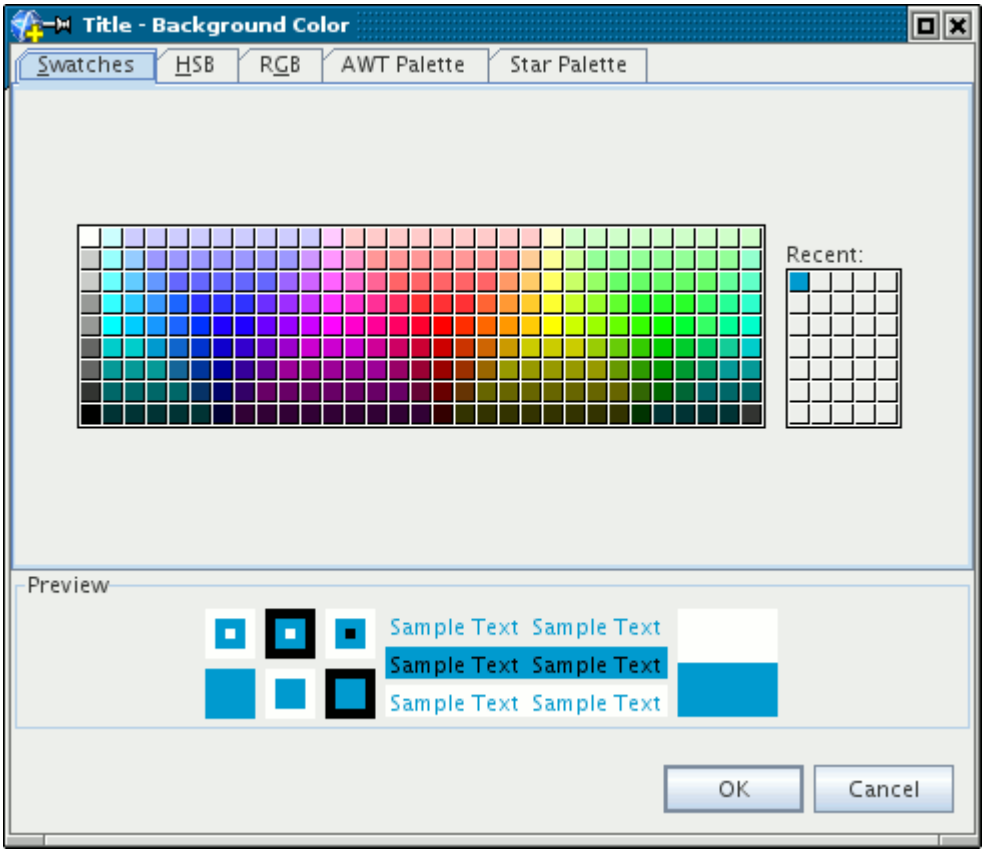


This opens the *Color* dialog, where colors can be selected.



20. To change the background color of an annotation:
    a) Select an annotation.
    b) Within the *Properties* window, activate the *Background* property.
    c) Then, click the ellipsis button to the right of the *Background Color* property.

    This opens the *Background Color* dialog, where colors can be selected. See Defining Colors.

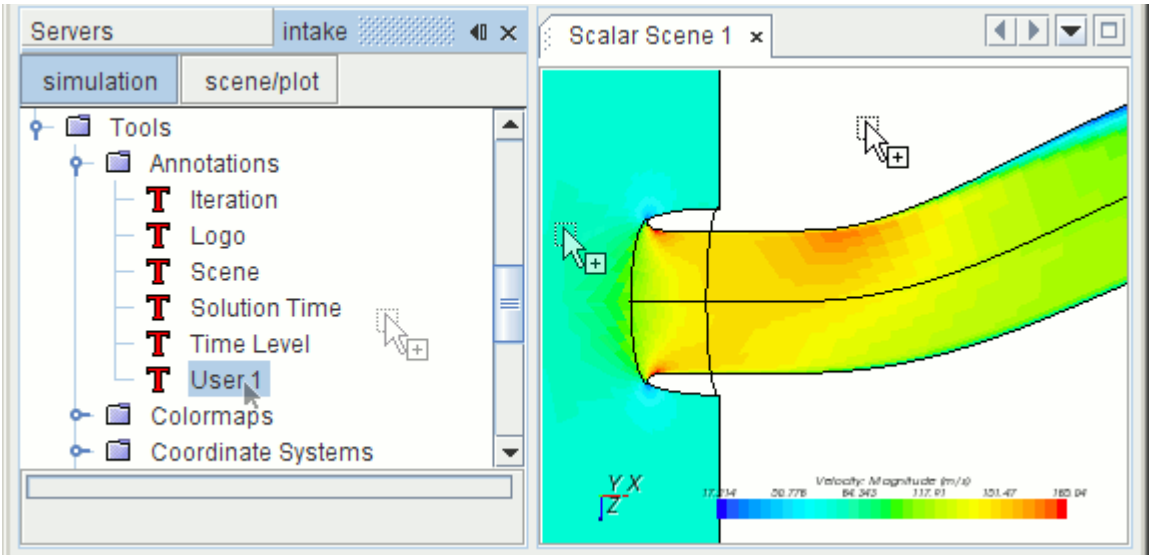


Other available options:

21. To copy properties of annotations:

    When properties are copied from one annotation to another, all of the properties are applied so that the destination annotation matches the source annotation. This applies only to user-defined annotations of the same type. For example, the properties of a 2D image annotation can be copied to another 2D image annotation, but not to a 3D image one.

    The drag-and-drop technique for annotations is to drag the annotation node into the visualization display of the destination scene.



22. To remove an annotation from a particular scene without eliminating it, you can use either of the following techniques:
    - Access the *Annotations* property of the scene node.
    - Bring into focus the scene from which you want to remove the annotation, then right-click the annotation and select **Remove**. The pop-up menu with this item is available when you right-click either the annotation in the scene display or its node in the object tree.

Simple Text Annotation

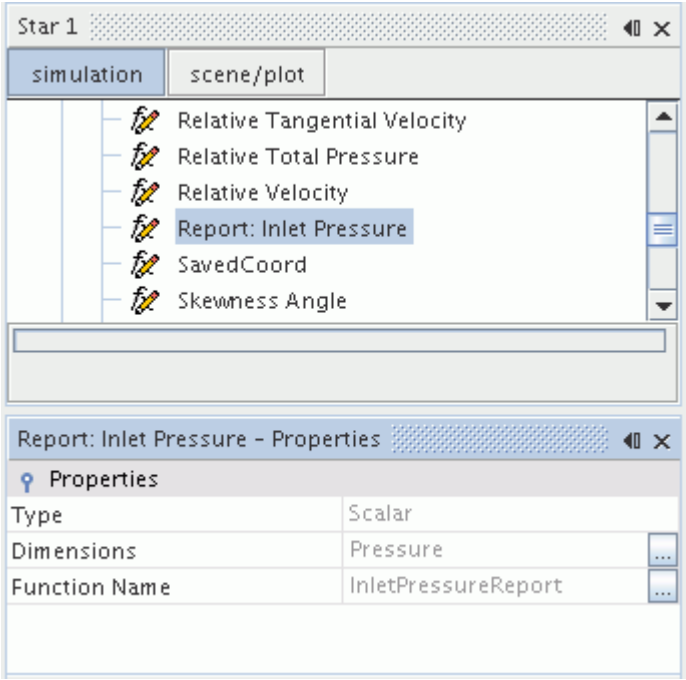
A simple text annotation is used to place a customized text label in a scene display.

Displaying a Report Value in a Simple Text Annotation

You can include the value of any report within the text of a simple text annotation. To do this:

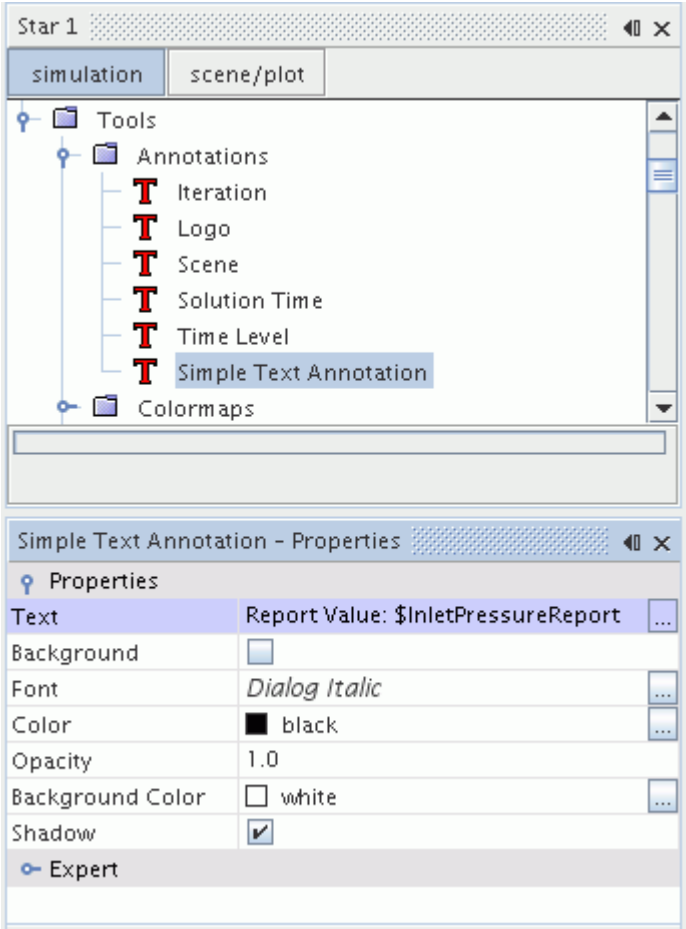
1. Find the field function that corresponds to the report within the **Tools** > **Field Functions** manager and record the name given in the *Function Name* property.

   The example below shows the field function corresponding to a report called **Inlet Pressure**. The *Function Name* is `InletPressureReport`.



2. Use this function name in the text of the simple text annotation by adding '`$`' to the front of the name.

   This would be `$InletPressureReport` for the example shown above. Note that this text can be surrounded by any other text if desired.



The function name will be replaced by the reported value, and will be updated whenever the report updates.

Image Annotations

You can use annotations to import 2D or 3D images into a display. The images are saved in the simulation, or can be linked through relative paths.

When you import an image, it is embedded in the simulation file by default. To link to an external image file (for example, for the purpose of limiting the simulation file size), *before you import,* deactivate the *Save to Simulation* property of the annotation node.

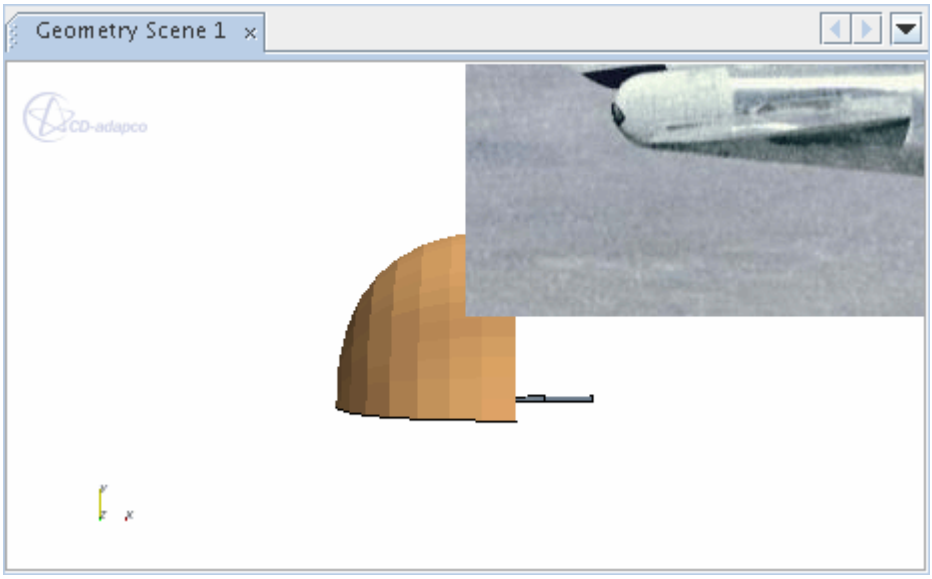
If you modify an image file after importing it into an annotation, you can update it within the simulation file. To update the image, right-click the node of its annotation in the **Tools** > **Annotations** node and select **Reload Image(s)**.

To export an image from within the simulation, right-click the node of its annotation in the **Tools** > **Annotations** node and select **Export Image(s)**. In the *Save* dialog, enter a file name and select the location for the image file, which is saved in PNG format. This option can be useful when you want to use the image by itself, but you only have the simulation file that contains it.

2D Image Annotation

A 2D image annotation is used to import a graphic into the background of the scene display. The formats that can be imported are PNG, JPEG, TIFF, PNM, and BMP.

Once you have added this annotation to any scene, the picture will appear in the background. Its lower left corner will be set at the center of the display by default.



Logo Annotation

A *logo annotation* is similar to a 2D image annotation in that you can import an alternate graphic using the *Path* property.

Background Annotation

A *background annotation* functions in a manner similar to that of a 2D image annotation, except with fewer settings. It is useful when you simply want to import a ready-made image into the background of a scene display.

3D Image Annotation

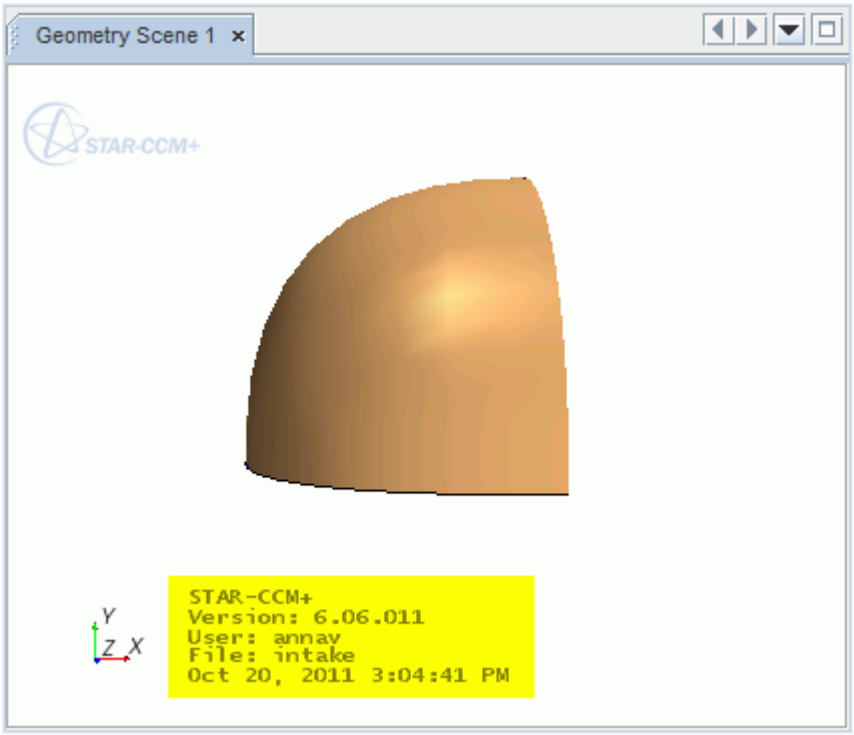
The 3D image annotation lets you import an image and position it within three-dimensional space in the scene, rather than in the background of the scene display. This feature facilitates more precise placement of an image near a part of the model.

Iteration, Time-Step, and Solution Time Annotations

Some annotations allow you to display the solution state in the display. The read-only text of these annotations is updated automatically as more steps are completed.

Scene Annotation

The scene annotation, one of five pre-defined annotations created with each new simulation, displays information about the simulation in the scene display.

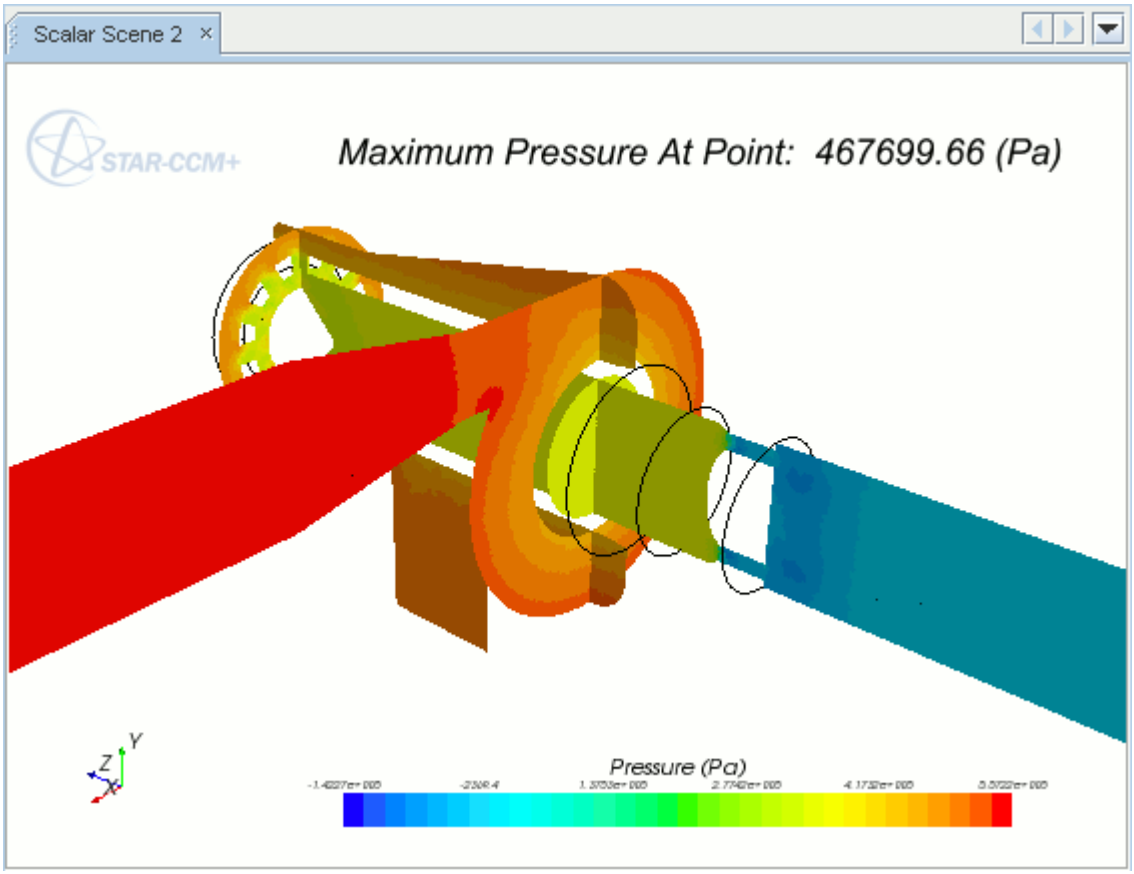


Report Annotation

A report annotation displays the result of a report as formatted text in the scene display. You create a report annotation from the right-click menu of a report node, and then apply the resulting annotation to the scene.

The text is refreshed whenever the report itself updates. This annotation has its own properties and shares a pop-up menu with other types of annotations. In the screenshot below, **Inlet Mass Flow** and **Outlet Mass Flow** are report annotations.

The *Number Format* property of the report annotation changes the formatting of the report value in the scene display. Full details are provided in the section What Is the Number Format? As an example, the scene below shows the value of a pressure report annotation having the *Number Format* set to `%10.2f`. This format means that the number will be presented in decimal notation, with a minimum of ten characters overall and two characters after the decimal point. As only nine characters are required to display the number in this format, a space is inserted before the leading digit.



Plot and Scene Image Annotations

Plot and scene image annotations are used to import a plot or a scene as an image in another scene display.

The annotation is updated in real-time as the solver is running or when changes are made to the plot or scene that is used for the annotation. Furthermore, the annotation is refreshed whenever any of its properties are modified.

However, when the solver is running, the update frequency setting of the source object (the plot or scene shown as the annotation) is ignored. When the update frequency criterion is met for the destination scene (the scene containing the annotation) both the source object and destination scene get updated.

See Also:

Working with Scenes

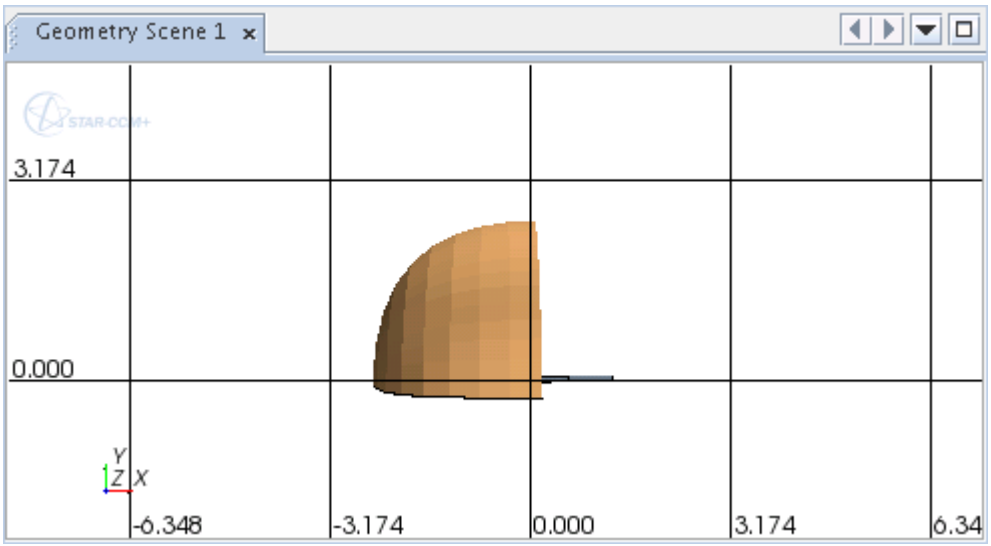
Scene Grid Annotation

A scene grid annotation lets you add a grid to a scene, allowing easier interpretation of visual data.

The annotation is represented by a node with a default name of **Scene Grid n**.

This object has properties for customizing this grid, including subnodes that let you set the grid spacing with display units or model units. It also shares a pop-up menu with other types of annotations.

The grid is perpendicular to your view of the scene display.



What Is the Display Spacing?

The **Display Spacing** subnode of the scene grid annotation represents the use of scene display units to specify the spacing of the grid, using entries in this object's properties.

Display Spacing Properties

Horizontal Spacing	Sets spacing in the horizontal direction.
Vertical Spacing	Sets spacing in the vertical direction.

What Is the Model Spacing?

The **Model Spacing** subnode of the scene grid annotation represents the use of model units to specify the spacing of the grid, using entries in this object's properties.

Model Spacing Properties

Horizontal Spacing	Sets spacing in the horizontal direction.
Vertical Spacing	Sets spacing in the vertical direction.

Working with the Derived Data Annotation

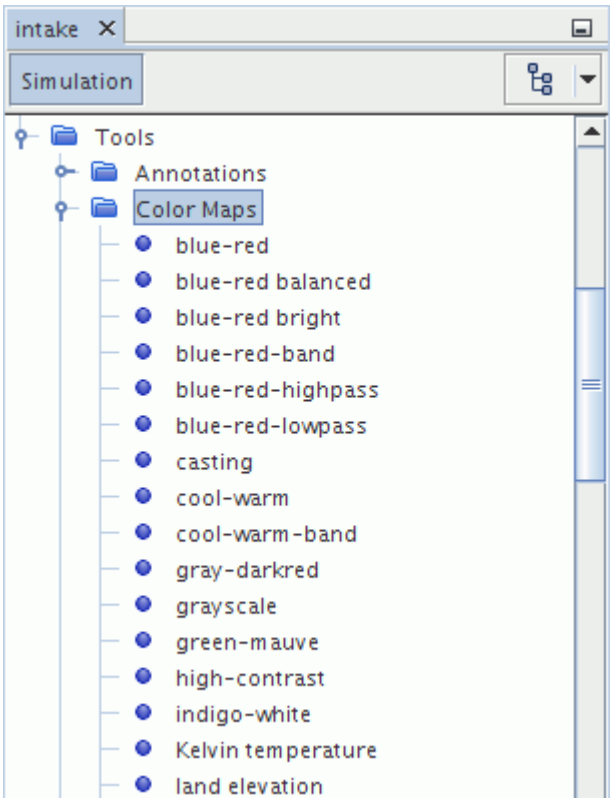
A derived data annotation is used to display information about derived data objects that are associated with data set functions.

This annotation has a node with the default name **Derived Data Annotation n**, with **n** being the total number of annotations of this name in the simulation.

Color Maps

The **Color Maps** node is the main manager object for all color maps that have been created in the simulation.

By default this node contains a range of pre-defined color maps when the simulation is created.



Many of the controls for colors in a scene are in the color bar of a displayer. However, these objects let you create custom color configurations. While you cannot edit pre-defined color maps, you can define your own.

Of the 31 pre-defined color maps that are available, most of them belong in one of three categories: Perceptual, Qualitative, and Divergent. For details about these categories, see Pre-Defined Color Maps.

In addition, a number of the color maps are color-blind accessible. These color maps include:

- green-mauve
- orchid-green
- tropical
- water

Managing Color Maps and Setting Defaults

Color maps are stored in the `config\visSettings.props` file in the user settings folder of STAR-CCM+, independent of any simulation. (For more information on the settings folder, see Resetting STAR-CCM+ Default Options.) This storage has the following effects on the availability and management of user-defined color maps:

- When you create a color map in a simulation, it is added to the `visSettings.props` file, and is available for selection in any simulation that is opened subsequently in that copy of the STAR-CCM+ client. However, your new color map will not be saved in the simulation file separately unless that color map is used by another object in the simulation, such as a displayer.
- Alternatively, you can save a color map separately in a simulation by importing it from an external `.colormap` file (see Exporting and Importing Color Maps).
- When you delete a user-defined color map from a simulation, it also gets deleted from the settings folder. However, if that color map is specified as the default, it is retained in the settings folder.
- A color map in the settings folder is not available in a simulation that already has a color map with matching properties.
- If you modify the properties of a user-defined color map, that change does not appear in any simulations that are already open. However, the change does appear in simulations that are opened subsequent to that change.

You can specify which pre-defined color map is used by default. To select the default color map, use one of the following techniques:

- Right-click the node of the color map and select **Set as Default Color Map**.
- Use the default options of the STAR-CCM+ client:
  1. Select **Tools** > **Options**.
  2. In the *Options* dialog, select the **STAR-CCM+** > **Visualization** node.
  3. Make a selection from the drop-down list of the *Default Color Map* property.

Note: Displayers that were created in earlier simulations retain their color map selections. However, any displayer that is added to an earlier simulation complies with the default color map setting.

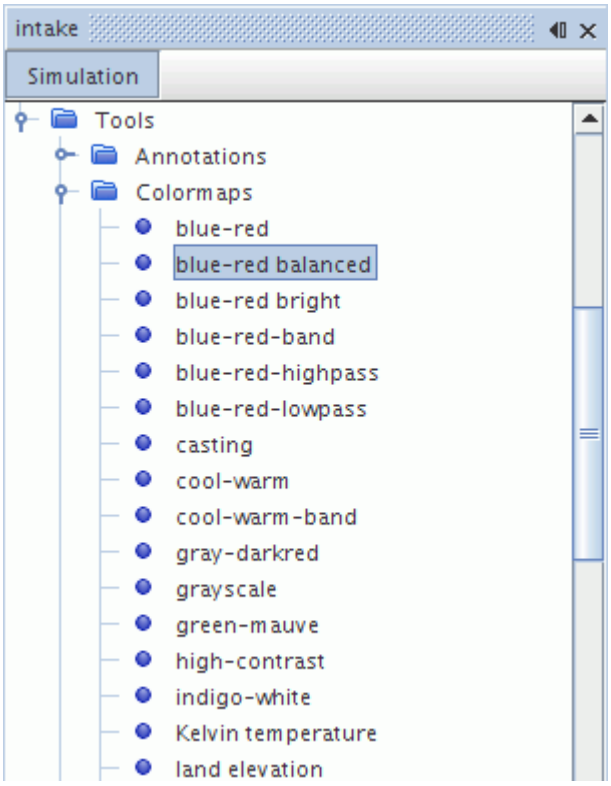
Contents:

What Is a Color Map?
Pre-Defined Color Maps
Creating Color Maps
Defining Color Maps
Color Map Dialog Controls
Exporting and Importing Color Maps
Deleting and Removing Color Maps
Pop-Up Menu of the Color Maps Manager

What Is a Color Map?

A color map is a table of color values that specifies certain colors and shades for data in the scene display.

Each color map object in the simulation resides in the main color maps manager, and is accessed by scenes in which you select that color map.



User-defined color maps can be edited in the color map editor.

All user-defined color maps may be renamed. Pre-defined color maps cannot be renamed. When you rename a color map that is being used in a scene, the new name appears automatically in the drop-down list of the displayer color bar's properties.

Pre-Defined Color Maps

STAR-CCM+ includes a set of pre-defined color maps for you to use in controlling the display of colors in scenes.

Color Map Categories

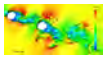
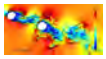
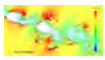
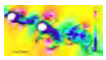
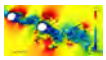
The pre-defined color maps can be aligned with at least one of the categories below:

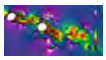
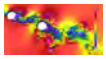
- Perceptual
  - Color maps in this category have well ordered colors and are nearly or fully colorblind accessible.
- Qualitative
  - Use these color maps to create high impact visualizations. Some of these maps are based largely on saturated colors and are intended to deliver strong visual aesthetics.
- Divergent
  - Use these color maps to highlight differences across a scalar range. Some maps are effective at highlighting small differences while other maps are designed to highlight large differences in range.
- Specialty

- These color maps are recognizable and designed for specific uses. The land elevation (or land sea elevation) maps are based on geographical surface elevation contours, for example.

Color Maps

In the table below, each of the pre-defined color maps is applied to the the same visualization scene.

blue-red (qualitative)	
blue-red balanced (qualitative)	
blue-red bright (qualitative)	
blue-red-band (qualitative)	
blue-red-highpass (qualitative)	
blue-red-lowpass (qualitative)	
casting (specialty map)	
cool-warm (qualitative)	
cool-warm-band (qualitative)	
gray-darkred (qualitative)	
grayscale (perceptual)	
green-mauve (perceptual)	
high-contrast (qualitative)	
indigo-white (qualitative)	
Kelvin temperature (specialty)	
land elevation (specialty)	
land sea elevation (specialty)	
orchid-green (perceptual)	
purple-red (qualitative)	

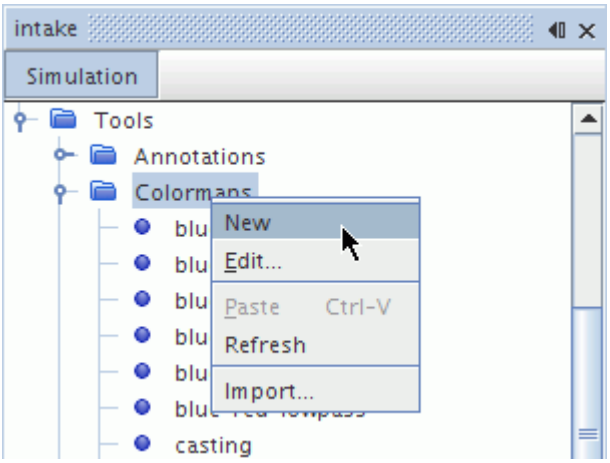
purple-red (large difference) (divergent)	
purple-red basic (large difference) (divergent)	
radiation spectrum (divergent)	
red-blue (small difference) (divergent)	
red-green (small difference) (divergent)	
red-purple desaturated ends (qualitative)	
spectrum (qualitative)	
thermal (specialty)	
tropical (perceptual)	
water (perceptual)	
white-pink (qualitative)	
white-red (qualitative)	

Creating Color Maps

This section explains how to create color maps.

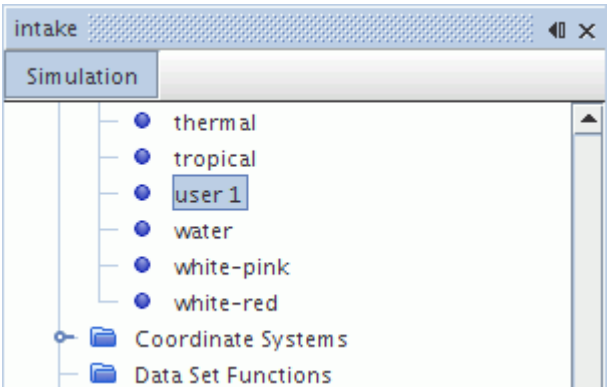
To create a color map:

1. Expand the **Tools** node and right-click the **Color Maps** node. Select **New** from the pop-up menu.



A new color map appears under the **Color Maps** node with a default name. If you have a visualization scene open in the *Graphics* window, the *Create Color Map* dialog opens in the *Edit* panel.

2. Proceed to define the color map.



Once the color map has been created, it can be renamed at any time, modified to specify the required colors, or selected in any scene.

Activating the Color Map Editor

3. To activate the color map editor, use one of the following techniques:
   - Select the color map node and click the right half of the *Color Map* property.
   - Right-click the color map node and select **Edit with Color Map Editor**.

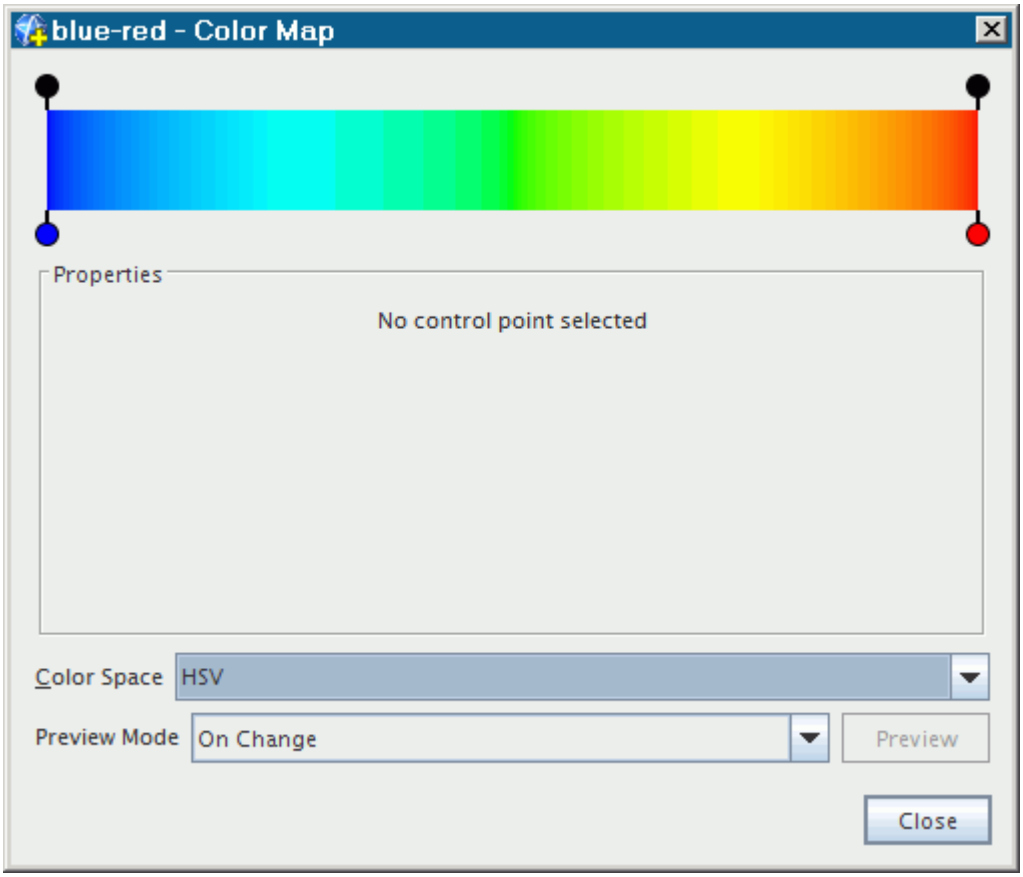
   When you use the **Edit with Color Map Editor** menu item with a scene open, the dialog appears in the *Edit* tab of the simulation window.

Defining Color Maps

When you create a color map, by default it has the properties of a blue-red color map which is 100% opaque. The color map editor lets you change those properties.

With the color map editor, you can do the following:

- Modify the colors at the end of the color bar
- Add specific colors anywhere along the spectrum of the color bar
- Control the opacity of the colors along any part of the color bar



As you hover the mouse pointer over the top and bottom edges of the color bar, STAR-CCM+ displays prospective control point positions. Control points above the color bar define opacity; control points below the color bar define color. Click existing control points to edit their properties.

Adding a Color to the Spectrum

1. Click the position beneath the color bar where you want to place the color point.

   The *Properties* of the color point appear in the dialog.
2. Click the **Custom Editor** button to reveal the *Change Color* dialog.
3. In the *Change Color* dialog, select the color that you require and click **OK**.

Adding Variation in Opacity to the Spectrum

4. Click the position above the color bar where you want to place the control point for opacity.

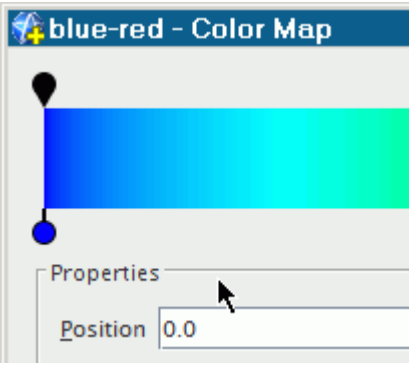
The *Properties* of the opacity control point appear in the dialog.

5. Adjust the slider or enter a specific value. Its values range from 0.0 (100% transparent) to 1.0 (100% opaque).

Modifying Existing Control Points

6. To access the properties of a control point, do one of the following:
   - Click the control point.

     The control point takes the shape of a teardrop, indicating that it has the focus. The properties of that control point appear in the *Properties* group box.



   - Double-click the control point.

   A separate dialog appears for adjusting color or opacity.
7. To change the position of a control point, click and drag it.

   The *Position* property of the control point changes as you move it.

Color Map Dialog Controls

The *Edit Color Map* dialog exposes several properties for the color spectrum and control points along it.

Preview Mode	Controls how the dialog previews your changes.	
	On Change	Shows the effect as soon as you complete a change.
	Manual	Shows the effect after you complete a change and click **Preview.**
	Continuous	Shows the effect as you are making a change.
Color Space	Sets the color space that is used for interpolation.	
	RGB (Red, Green, and Blue)	The color model that is commonly used in video displays.
	HSV (Hue, Saturation, and Value)	Works like mixtures of paint.
	Lab	The CIE-L*a*b system.
	Diverging	Involves a special mode where colors pass through white when interpolating between two saturated colors.

	HSV-Wrapped	Uses HSV, but takes the shortest path in Hue (going back through 0 if that is the shortest way around the hue circle).

Exporting and Importing Color Maps

Once you set up a custom color map in a simulation, you can share it by exporting it to file. This technique lets you make your color map accessible beyond the STAR-CCM+ settings folder where it is stored by default.

To copy color maps from one simulation to another when both simulations are active, use copy and paste actions on the object tree..

To work with color maps as separate files:

1. In the simulation file that contains your custom color map, open the **Tools** > **Color Maps** node.
2. Right-click the node of your custom color map and select **Export**.

   Note: You can export more than one color map to a single file. To do so, multiselect the nodes and then right-click.

3. In the *Save* dialog:
   a) Navigate to the correct directory where you want to store the color map.
   b) In the **File Name** text box, enter an appropriate name for the file, for example something that associates the file with the related project.
4. Click **Save**.

   The color map is available as a separate file with the `.colormap` extension.

To import a color map into a simulation file:

5. Right-click the **Tools** > **Color Maps** node and select **Import**.
6. In the *Open* dialog, navigate to the desired `.colormap` file and click **Open**.

   The node of the color map appears in the **Tools** > **Color Maps** node.

   Note: If the `.colormap` file contains more than one color map, the nodes for all of them appear as a result of a single import action.

Deleting and Removing Color Maps

All user-defined color maps may be deleted, provided they are not used by any scene displayer. Pre-defined color maps cannot be deleted.

When you delete a user-defined color map in a simulation, it also gets deleted from the settings folder of STAR-CCM+. For more details on how color maps are stored and deleted, see Managing Color Maps and Setting Defaults.

If you want to remove a color map from a particular scene without eliminating it, access the *Color Maps* property of the scene displayer color bar node and select a different color map.

Pop-Up Menu of the Color Maps Manager

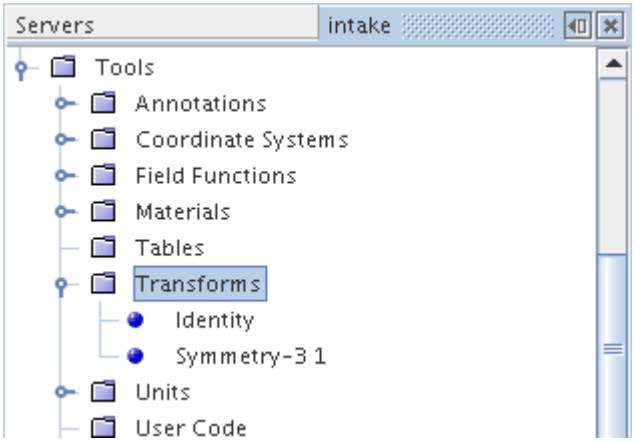
Use this pop-up menu to create color maps.

New	Adds a new color map to the **Color Maps** manager node.

Using Transforms

The **Transforms** node, which has a pop-up menu, is a manager object for all transforms that have been created in the simulation.

By default it contains one transform named **Identity**, which cannot be deleted or renamed, when the simulation is created. You can create additional transforms directly through the pop-up menu. The transforms manager can hold any number of transforms up to the memory limits of the computer.



This section defines what is meant by a transform and describes how to create, use, and manage transforms in STAR-CCM+.

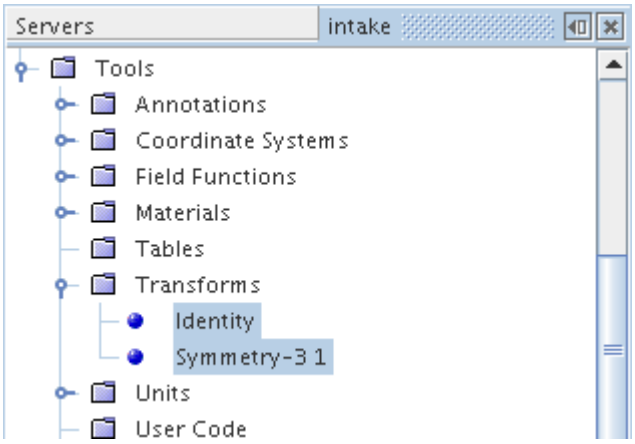
Contents:

What Is a Transform?
How Transforms Work with Coordinate Systems
Using Automatically Created Transforms
Creating a New Transform
Deleting Transforms
Using a Symmetry Transform
Displaying Different Scalar Quantities on Either Side of a Symmetry Transform
Pop-Up Menu of the Transforms Manager

What Is a Transform?

Displayers use transforms to modify the default position, orientation, or size of the part in the displayer.

They are represented by nodes in the object tree which have properties and are subject to common menu items.



Transforms can repeat or mirror parts, or reposition, resize, or reorient them. The transform feature is for viewing only.

In STAR-CCM+, a transform is used with a displayer to allow for the repositioning, reflection, or repeat of parts in a scene. Transforms also work with coordinate systems (the Laboratory coordinate system by default).

Caution: When you switch the coordinate system of a transform, you change the spatial position of that transform. To compensate for this change, adjust spatial properties (such as *Rotation Axis* and *Rotation Origin*) of the transform. For details, see How Transforms Work with Coordinate Systems.

STAR-CCM+ provides the following types of transforms:

- Periodic transform—is automatically generated from a periodic interface once you initialize the solution, and is deleted if the interface is deleted. This transform gets most of its values automatically from the interface, such as the rotation angle, axis of rotation and translation vector, so the only property you can change is the number of repeats (copies). The graphics periodic transform lets you edit everything, but you must set all of its properties from scratch.

  Applying a periodic transform to a displayer shows the specified number of copies of the parts in the displayer, at periodic intervals based on information from the interface. Hence, the portion of the device that gets displayed depends on the following:

  - The parts included in the displayer
  - The number of copies
  - The intervals between the copies
- Symmetry transform—is an automatically generated transform from a symmetry boundary. This transform automatically extracts the plane of symmetry from the boundary. This transform is tied to the boundary and hence cannot be edited, and is deleted if the boundary type is changed, or the boundary is deleted. Applying a symmetry transform to a displayer shows the mirror image of the parts in the displayer about the symmetry plane.
- Graphics periodic transform—this transform behaves the same as a periodic transform, but it is your responsibility to define the rotation angle, rotation axis, or translation vector.
- Graphics symmetric transform—this transform behaves the same as a symmetric transform, but it is your responsibility to define the plane of symmetry. The plane is of the form ax + by + cz + d = 0, where a, b, and c are the x, y, and z components of the plane normal and d = -ax0 - by0 -cz0 where x0, y0, and z0 are the origin of the plane.

For an axisymmetric problem, the axis is always the x-axis. Therefore you should enter `0,1,0,0` in the *Plane* property of this transform.

- Meridional projection transform—this transform is a simple projection of the points of the part into the (z,r) plane.
- Simple transform—this transform does not copy the original part, but translates, rotates, or scales the part accordingly. Applying a simple transform to a displayer transforms the parts in the displayer accordingly.
- Identity transform—this transform is automatically generated, and is the default transform for all displayers. There are no properties associated with this transform, and when applied to a displayer, shows all parts in the displayer at their natural position, size, and orientation.

A transform can be applied to a displayer to view the entire model for a solution in which only a symmetric or periodic repeating portion was solved. For example, when modeling a compressor stage, one often only solves for a single blade row. It becomes useful to view adjacent blade rows, or even the entire repeating solution. Using a periodic transform as the displayer's transform provides a way to view the complete geometry. Similar arguments apply for a problem with mirror symmetry.

A simple transform can be used in a displayer to view parts which would normally occlude each other by separating them.

All transforms can be renamed, except for the identity transform. All transforms can be copied provided they are of the same type.

Contents:

Transform Properties

Transform Properties

These properties reflect the types of boundaries you created.

The transform named **Identity** does not have any of these properties.

For the properties that include units in the values, you can change the units. For details on transforms and units, see How Transforms Work with Coordinate Systems.

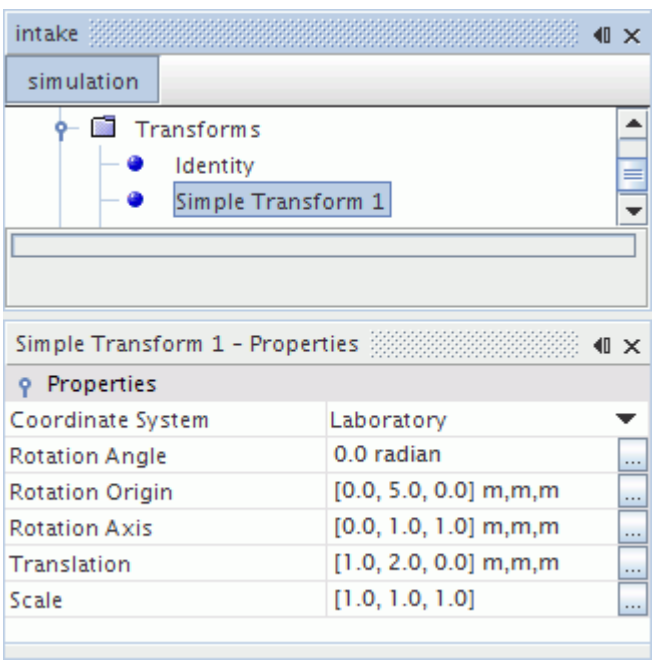
Coordinate System	A list of coordinate systems in the simulation for use by the transform.
Boundary	The read-only name of the boundary whose type is reflected in the particular transform.
Plane	Indicates the coordinates of the plane along which the symmetry transform is created.
Number of Repeats	Controls the number of times the image is repeated in the display of the *Graphics* window. This is for a periodic repeat only.
Interface	Indicates which interface is used automatically for a periodic repeat (read-only).
Rotation Angle	Sets the rotation angle for a simple transform; is set automatically and is read-only for a periodic repeat.

Rotation Axis	Sets the rotation axis for a simple transform and meridional projection transform; is set automatically and is read-only for a periodic repeat.	
Rotation Origin	Sets the rotation origin for a simple transform and meridional projection transform; is set automatically and is read-only for a periodic repeat.	
Scale	Scales the part (simple transform only). By saving this setting in a transform, you can access it readily by switching to this transform in a displayer.	
Translation	Shifts the Cartesian coordinates of the part in the display of the *Graphics* window; is set automatically and is read-only for a periodic repeat.	
Symmetry Mode	Selects the use of a symmetry transform.	
	Repeat	Displays mirror image of the parts in the displayer about the symmetry plane.
	Flip	Facilitates displaying different scalar quantities on either side of a symmetric transform.

How Transforms Work with Coordinate Systems

Transforms have units associated with each position property and axis property depending on the selected coordinate system.

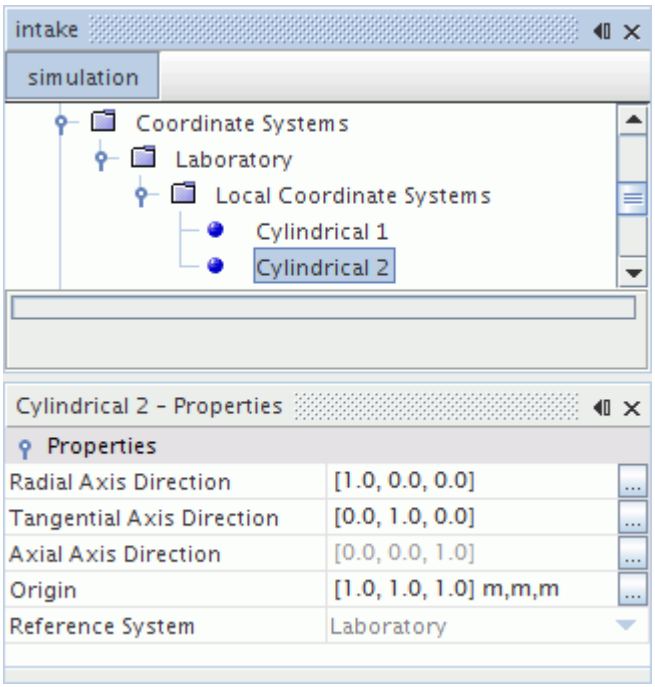
The following example shows a simple transform with its properties.



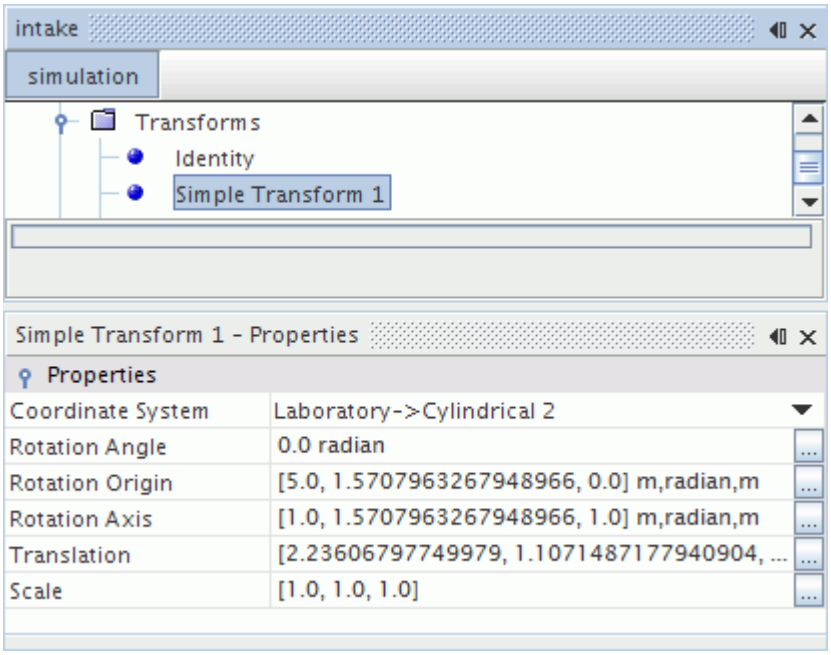
This transform has three properties whose specification and units depend on the chosen coordinate system: *Rotation Origin*, *Rotation Axis*, and *Translation*. Each of these properties is represented as a coordinate in the specified coordinate system, but the interpretation of the components depends on the type of property:

- The *Rotation Origin* property specifies a position in the chosen coordinate system which defines the rotation origin relative to the referencing coordinate system origin. In the above example, the position is `[0.0, 5.0, 0.0] m, m, m`, indicating the three components each have units of meters.
- The *Rotation Axis* property is a vector, but is also represented as a position in the chosen coordinate system. It is the vector formed from the origin of the chosen coordinate system to the specified position in that coordinate system.
- The *Translation* property is a vector, and shifts the Cartesian coordinates of the part in the display of the *Graphics* window relative to the origin of the chosen coordinate system.

When the coordinate system is changed, each property that depends on the coordinate system has its components recalculated with the units pertaining to that system. Consider the following example of a cylindrical coordinate system.



The Cylindrical 2 coordinate system has the Laboratory coordinate system as its reference system. The *Origin* property reads `1.0,1.0,1.0` because that is the position of the Cylindrical 2 coordinate system relative to the origin of the Laboratory coordinate system. Changing the *Coordinate System* property of Simple Transform 1 to `Cylindrical 2` results in a change of its properties as shown in the following screenshot.



The components of coordinate system-dependent properties are recomputed to reflect the appropriate units for the cylindrical coordinate system.

In this example, the position of Simple Transform 1 has shifted relative to the Laboratory coordinate system. This shift is because the transform now uses the Cylindrical 2 coordinate system for its origin, and the Cylindrical 2 coordinate system is positioned at 1, 1, 1 relative to the Laboratory coordinate system.

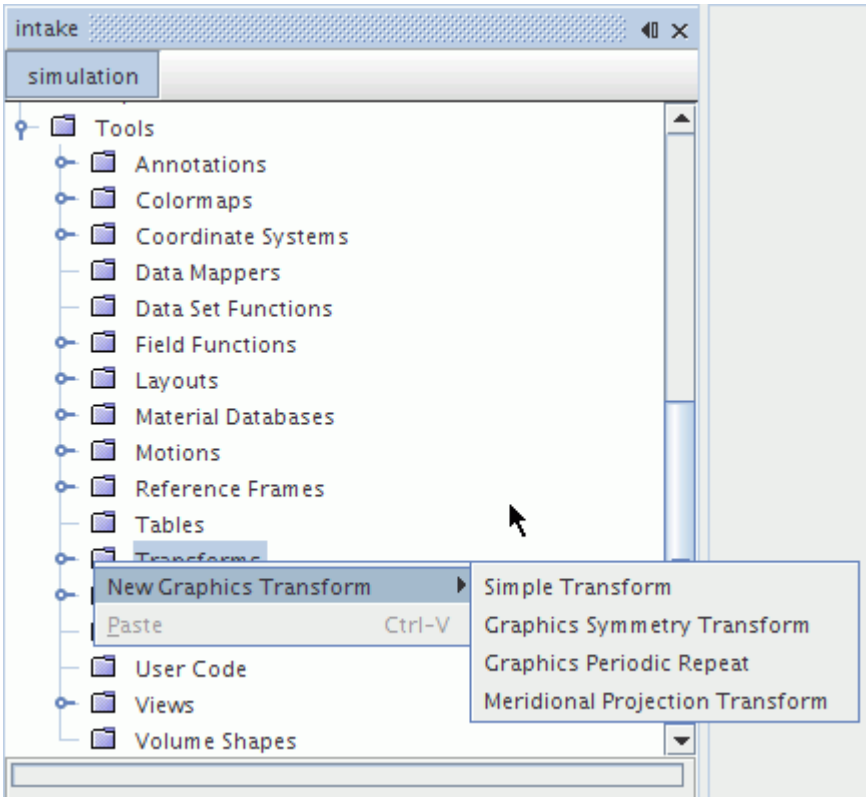
Using Automatically Created Transforms

In STAR-CCM+, there are three types of automatically created transforms:

- A symmetric transform—if a boundary is of type symmetry, then a corresponding symmetry transform node is created and stored under the **Tools** > **Transforms** node, its name being that of the boundary.
- A periodic transform—if there is a periodic interface in the simulation, then a corresponding periodic transform node is created and stored under the **Tools** > **Transforms** node, its name being that of the interface.
- The identity transform—this is the default transform, and lives under the **Tools** > **Transforms** node.

Creating a New Transform

To create a transform, simply right-click the **Tools** > **Transforms** node and create the desired type of transform.



A new transform node of the selected type appears in the tree under the **Tools** > **Transforms** node. The default name of the transform is based on its type. However, renaming the transform is possible.

You can edit the properties of the new transform to suit your needs. The new transform can now be used in any displayer, and appears in the list of available transforms in the *Transform* property of all displayers.

Transforms are shared, so changing the properties of a transform appropriately changes every displayer using that transform. For example, consider a scene with a part displayer and a scalar displayer both using a graphics symmetry transform. If you change the symmetry plane definition in the graphics symmetry transform, then both the part displayer and scalar displayer update and change the scene accordingly.

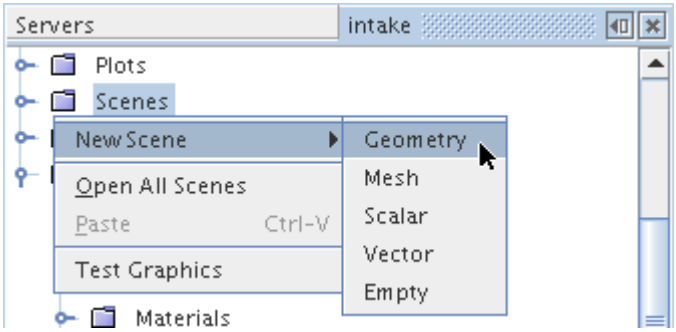
Deleting Transforms

All user-defined transforms can be deleted by right-clicking on the transform node and selecting **Delete**.

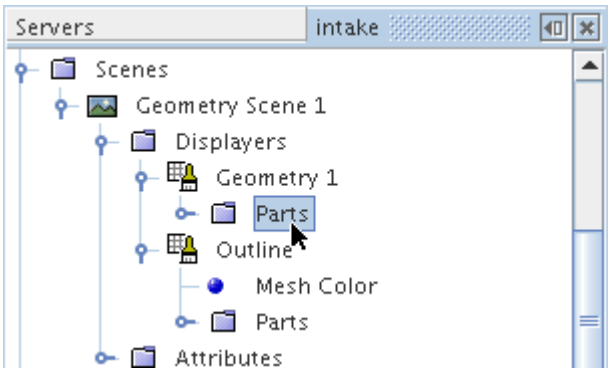
Automatically created transforms are managed by the simulation, and are only deleted when deemed necessary by the simulation. For example, a periodic transform is tied to an interface and is deleted if the interface is deleted or the interface type is changed from periodic. Similarly, a symmetry transform is tied to a symmetry boundary and is deleted if the boundary is deleted or the boundary type changes. An identity transform cannot be deleted.

Using a Symmetry Transform

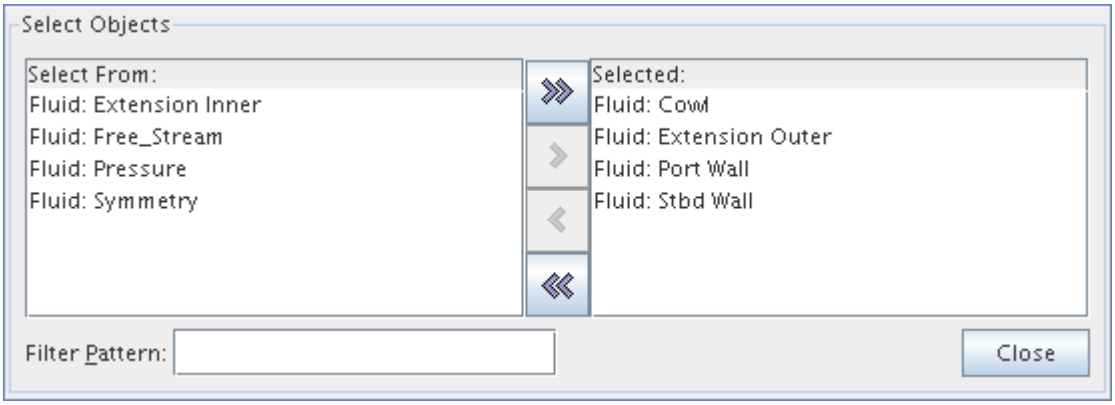
If you want to avoid changing the view that you might have set up in an existing geometry scene, right-click the **Scenes** node and select **New Scene** > **Geometry** from the pop-up menu.



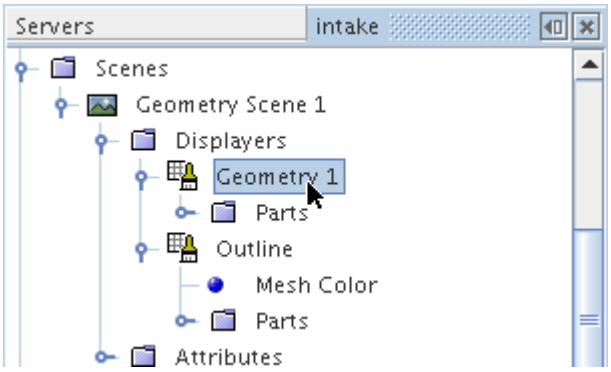
If you do create a geometry scene, remove some default parts to isolate the ones in which you are interested. Expand the node of the new scene, the **Displayers** node, and the **Geometry 1** node. Select the **Parts** node.



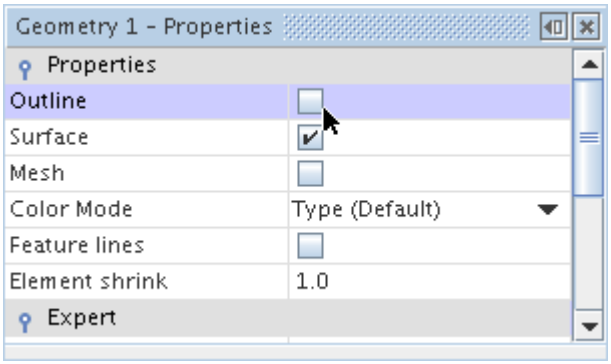
Use the *Parts* property to remove all parts except for the ones you want to view.



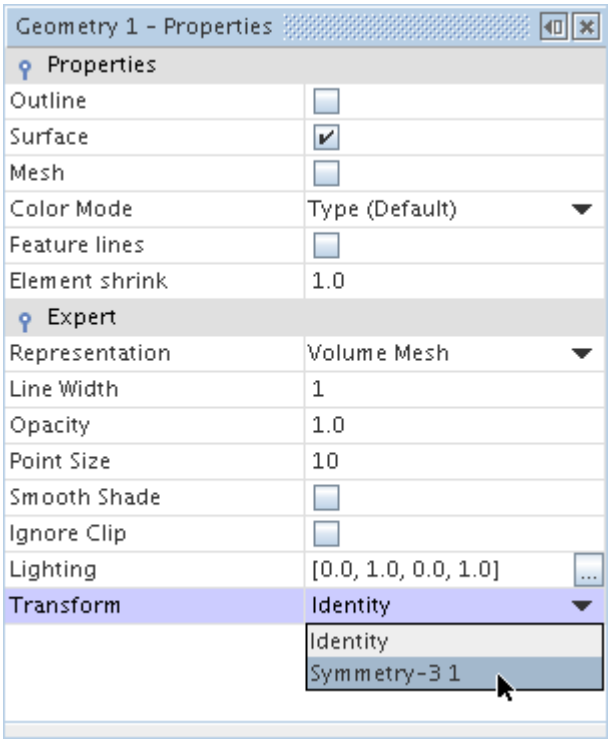
To see the design without the outline, select the **Geometry 1** node within the **Displayers** node.



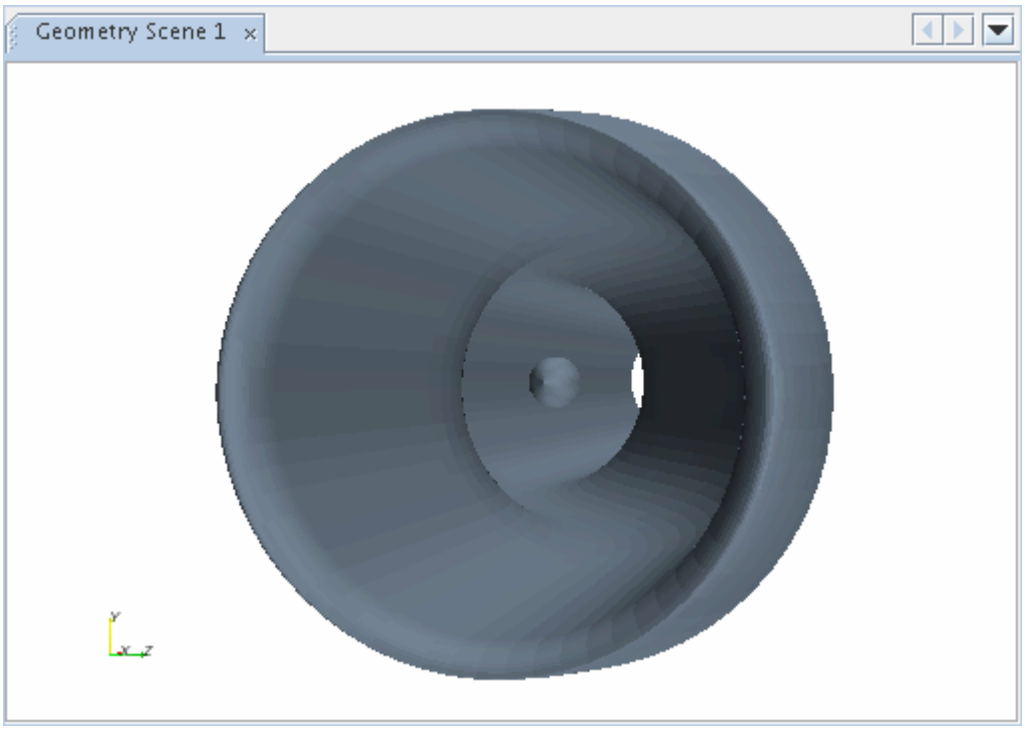
Deactivate the *Outline* property.



The transform lets you show the complete image in the *Graphics* window. With the **Geometry 1** node selected, set the **Transform** expert property to one of the `Symmetry` items in the drop-down list.



The model appears in full view in the *Geometry Scene* display. (The illustration below shows the model after rotating and zooming.)



Contents:

Unwrapping, Unrolling, and Projecting with Transforms
Using a Meridional Projection Transform

Unwrapping, Unrolling, and Projecting with Transforms

Some visualization requires unwrapping results to a flat plane.

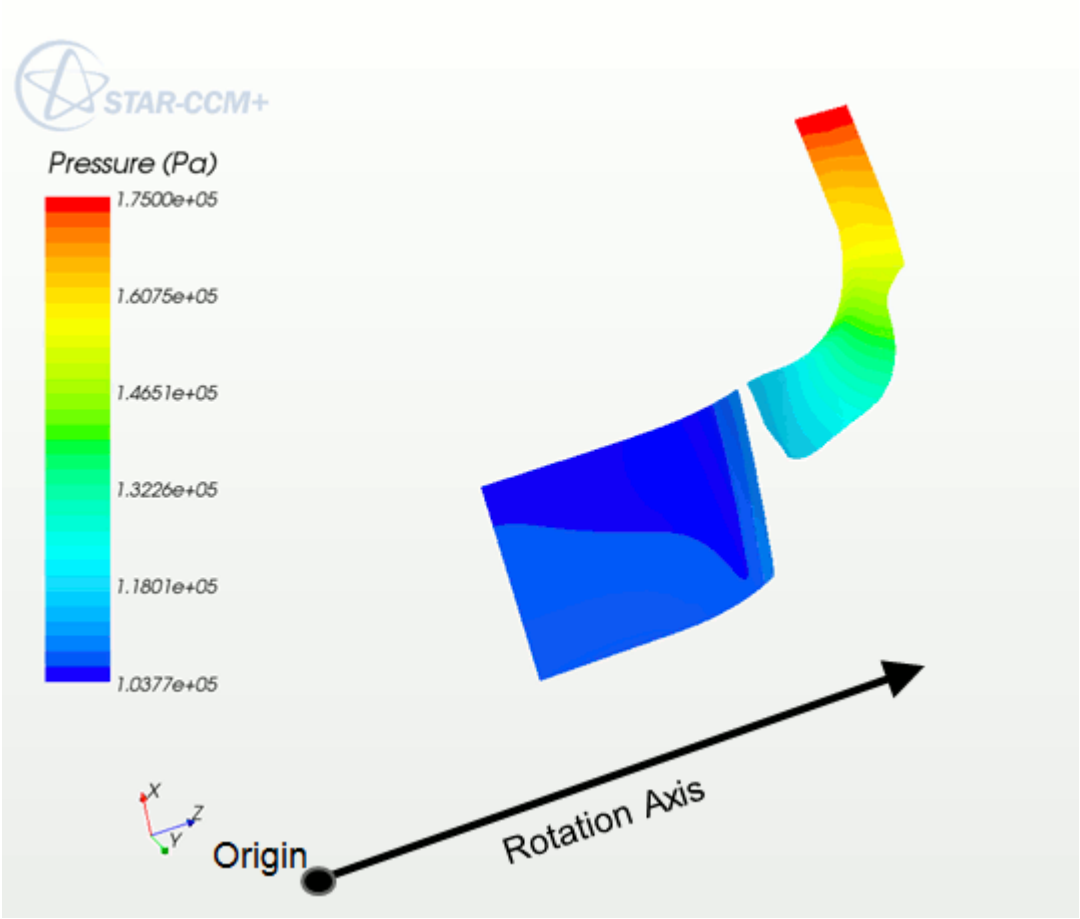
Consider, for example, a turbomachinery case that has a blade-to-blade view for a radial compressor that was created with the block-mapped coordinate system. The flow enters the domain in the axial direction and exits in the radial direction. As a result the blade-to-blade view sweeps around from the axial to the radial direction. Viewing results on such a curved surface would be difficult. Additionally for a meridional view, it would be useful to project the results on the radial-axial plane. For these reasons STAR-CCM+ provides the Meridional Projection transform.

In addition, the (m', theta) warp is available as a derived part.

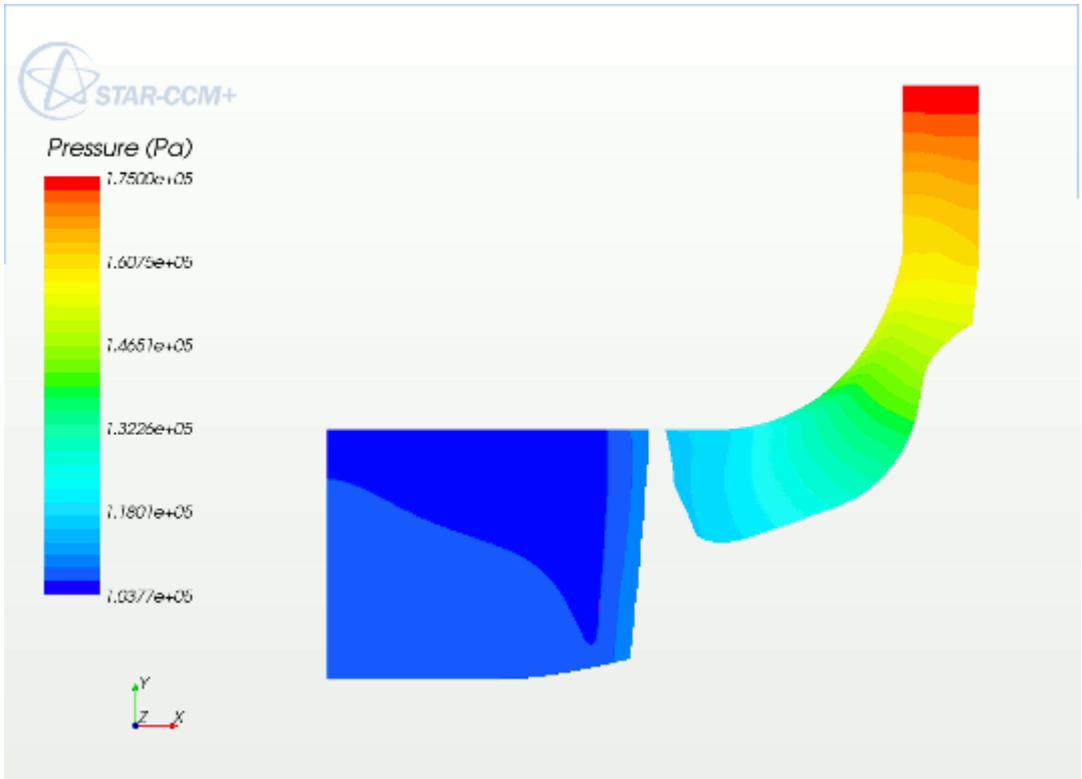
Using a Meridional Projection Transform

This transform is a simple projection of the points of the part into the (z,r) plane. It requires only two vectors in its properties: *Rotation Origin* and *Rotation Axis*.

The following example uses the meridional projection transform:



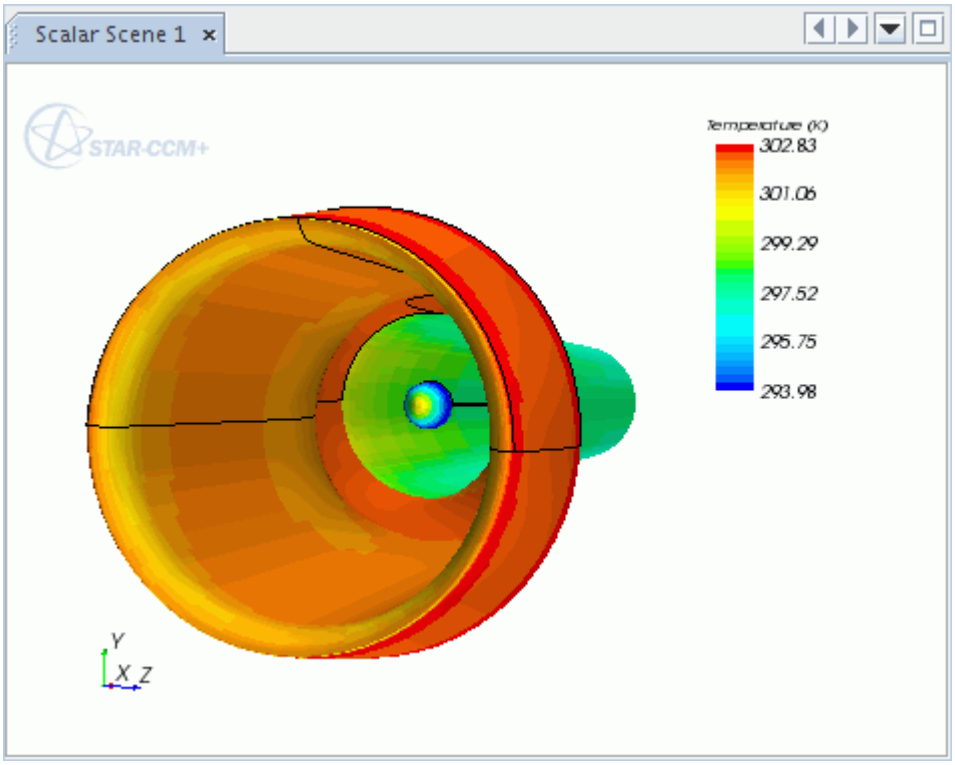
- The rotation axis is [0, 0, 1] (aligned with the z-axis).
- The origin is upstream of the surface at [0, 0, 0].
- The following screenshot shows the view down the z-axis. The transform maps the part to the (x, y) axis.



Displaying Different Scalar Quantities on Either Side of a Symmetry Transform

The symmetry transform's **Repeat** option allows you to create a reflection of a scalar displayer's parts, but it shows the same parts and scalars in both their actual and transformed locations.

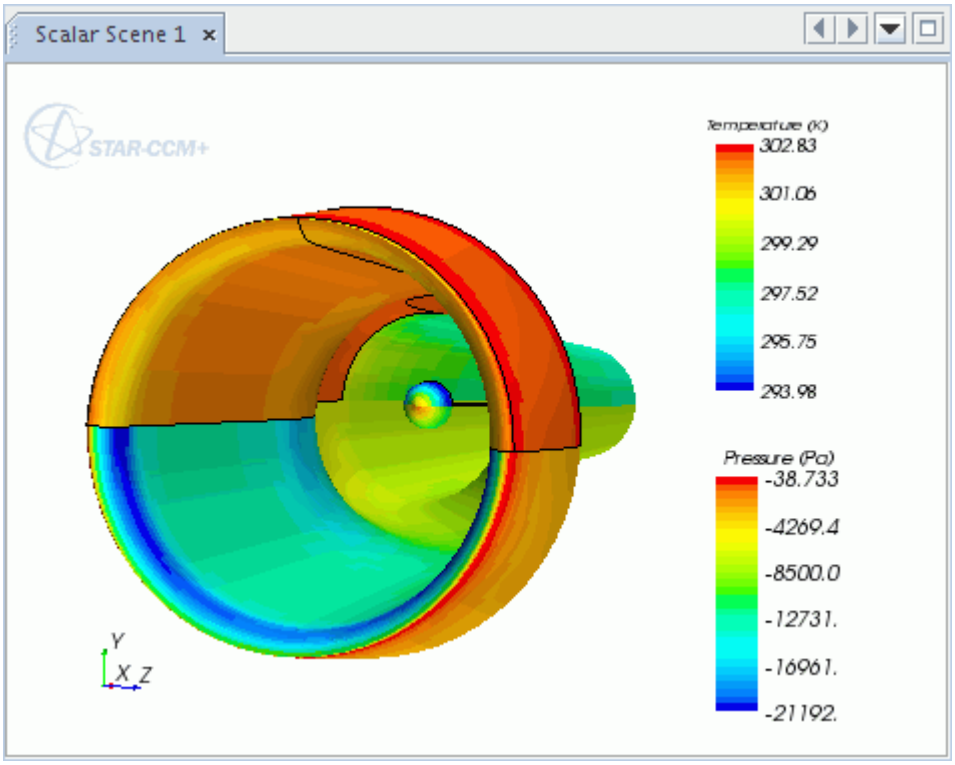
The following screenshot shows an example of a displayer using this option with temperature as the scalar.



By using this transform's **Flip** option, you can display a different scalar quantity on the mirrored set of parts. For example, to show temperature on the original and pressure on the mirror, do as follows:

- Create a graphics symmetry transform, make sure that the *Plane* property conforms to the symmetry plane that you need, and then set the *Symmetry Mode* property to `Flip`.
- Set up one scalar displayer showing temperature with no transform (that is, the default `Identity` selection).
- Set up a second scalar displayer as a copy of the first, except it shows pressure and uses the symmetry transform.

You have a mirrored displayer with different scalar quantities.



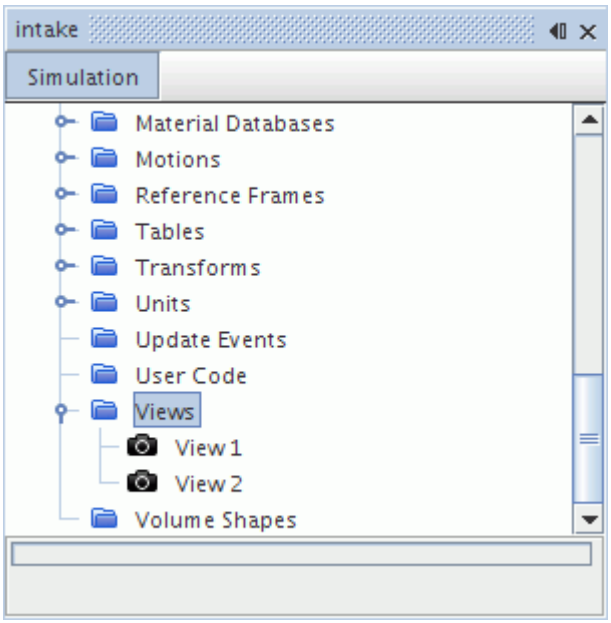
Pop-Up Menu of the Transforms Manager

These pop-up menu items are available when you right-click the **Transforms** manager node.

New Graphics Transform	Creates a transform of the type selected in the submenu:	
	Simple Transform	Executes the create simple transform action to add a simple transform to the **Transforms** node.
	Graphics Symmetry Transform	Executes the create graphics symmetry transform action to add a graphics symmetry transform (mirror image) to the **Transforms** node.
	Graphics Periodic Repeat	Executes the create graphics periodic repeat action to add a graphics periodic repeat transform (multiple repeats of the model that comprise a larger object) to the **Transforms** node.
	Meridional Projection Transform	Executes the create meridional projection transform action to add a meridional projection transform to the **Transforms** node.

Working with Views

The **Views** node, which has a pop-up menu, is the manager object for all views that have been saved in the simulation.



It exists even when empty to allow creation of the first view. The views manager can hold any number of views up to the computer's memory limits.

This section defines what a view is and describes how to use it.

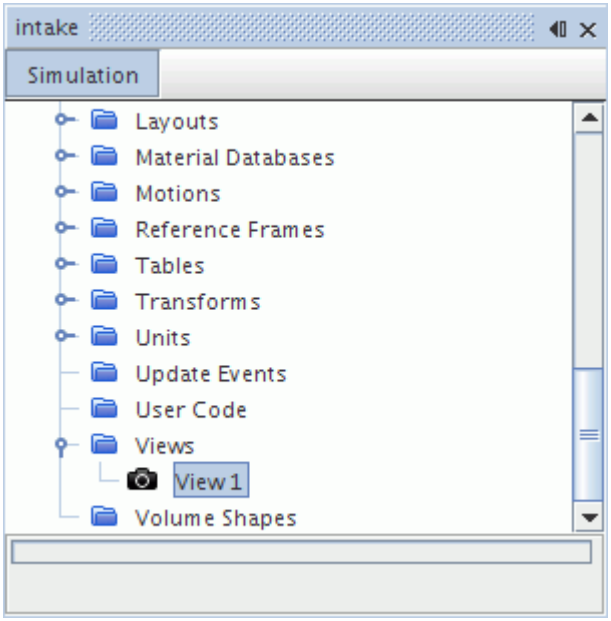
Contents:

What Is a View?
Setting a Predefined View
Setting the Coordinate System of the View
Copying Properties of Views
Saving a View
Creating a View
Restoring a View
Modifying Views
Changing the Projection Mode
Following Mesh Motion in a View
Pop-Up Menu of the Views Manager

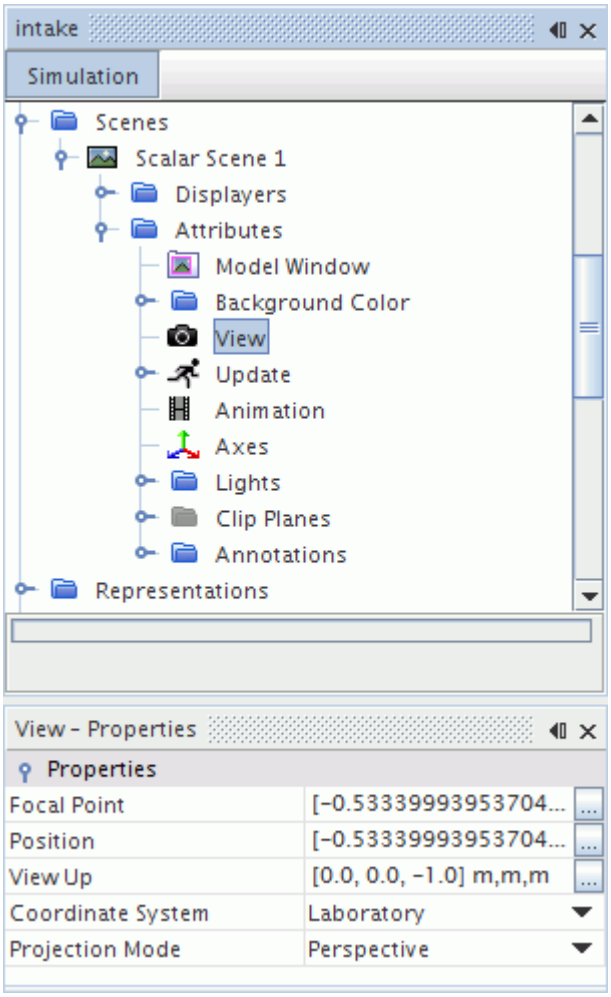
What Is a View?

In STAR-CCM+, every scene has a camera to convert the three-dimensional objects in a scene into the two-dimensional image visible on the screen. The information defining the orientation, position and projection of the three-dimensional objects is stored in a **View** object in STAR-CCM+.

When a specific view is created or saved a **View** node is added to the simulation tree, which has its own properties and is subject to common menu items.



Information on the current view of a scene is displayed by selecting the **Attributes** > **View** node for the scene. The current view can be controlled interactively by the changing the property values of this node, by using the mouse, or by defining specific views using the methods described in this chapter.



There are no restrictions on renaming views. Saved views can be deleted.

View Properties

The properties of a view consist of the position, focal point (target), orientation (view-up) and projection mode of the image in a scene display. The focal point and position vector are displayed under their respective properties in the *Properties* window. Another vector is displayed under the *View Up* property which dictates the vertical vector in the scene window. You can set vector coordinates with a single copy-and-paste action (see Pasting Coordinate Sets Directly into Dialog Fields).

The *Projection Mode* property controls how the camera maps world coordinates into view coordinates. The two options are:

- Perspective - implies a mapping of world coordinates into view coordinates that roughly approximates a camera lens. The center of projection is a finite distance from the view plane. As a result, the scene will have a vanishing point, and closer objects appear larger than more distant objects.
- Parallel - implies a mapping of world coordinates into view coordinates that preserves all parallel lines. In this mode, an object appears the same size regardless of how far away it is from the viewer. This is equivalent to having a center of projection that is infinitely far away in perspective mode.

The view properties are summarized below.

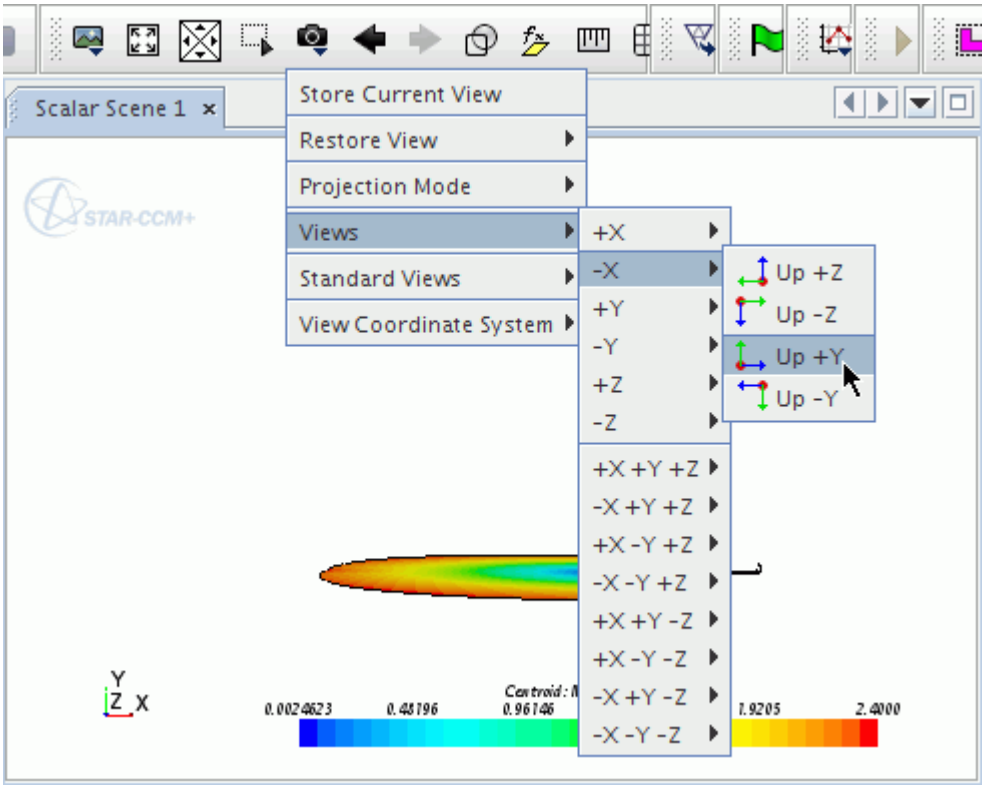
Focal Point	Numerical coordinates of the focal point. A comma or space must separate the numeric values. To see the effect, select the **Restore View** submenu of the (**Save-Restore-Select Views**) menu in the toolbar. The <H> key sets the focal point of the camera to the position where you hover with the mouse.	
Position	Numerical coordinates of the position. A comma or space must separate the numeric values. To see the effect, select the **Restore View** submenu of the (**Save-Restore-Select Views**) menu in the toolbar.	
View Up	The coordinates for viewing. A comma or space must separate the numeric values.	
Coordinate System	A list of coordinate systems in the simulation for use by this view.	
Projection Mode	Provides options for projecting the object in the display:	
	Perspective	Displays the object with a vanishing point.
	Parallel	Displays the object with edges parallel to each other.

Setting a Predefined View

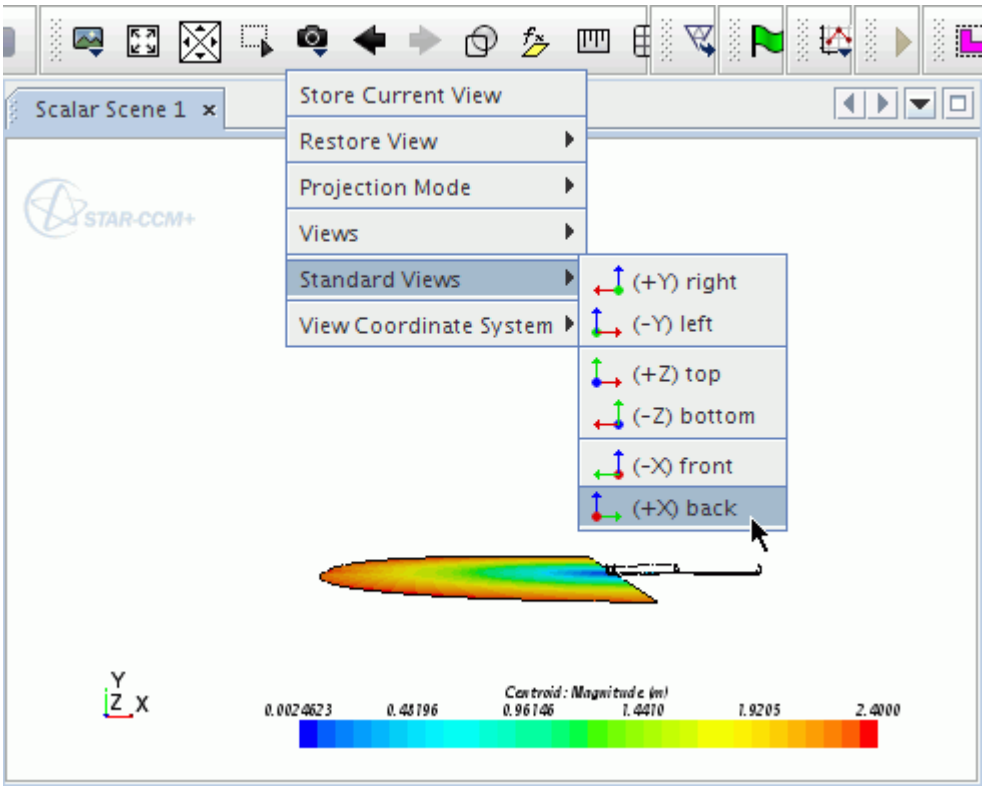
STAR-CCM+ provides a number of predefined views which allow scenes to be displayed relative to common orientations of the xyz axes.

These are useful when you wish to orientate the scene with an axis or when the orientation of the scene has become difficult to determine and you wish to reset the scene relative to an axis.

From the *Scene* toolbar, click on the camera icon and select **Views** from the drop down menu. This reveals another drop-down menu from which you must choose the axis to look down. For instance, if the +X axis is selected the resulting view will have you looking along the x-axis with positive x pointing out of the screen. If you choose the -X axis the resulting view will have you looking along the x-axis with negative x pointing out of the screen. Once an axis and direction have been chosen you must then select a second axis and direction from the drop-down menu which will define the view up.



Another set of predefined views consists of standard views, accessed in the submenu by that name. These options correspond to aspects of the object in the scene.



An additional option to view the scene along the vector `[1,1,1]` is also provided in the view's properties.

See Also:

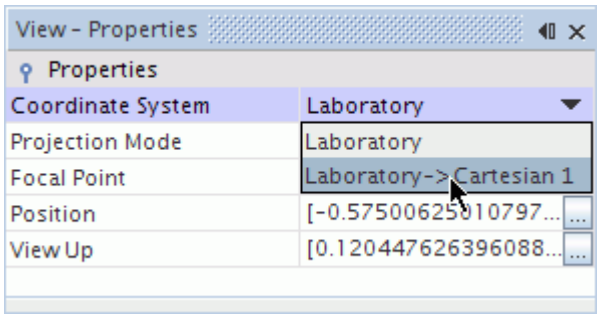
Using the Distance Measuring Tool

Setting the Coordinate System of the View

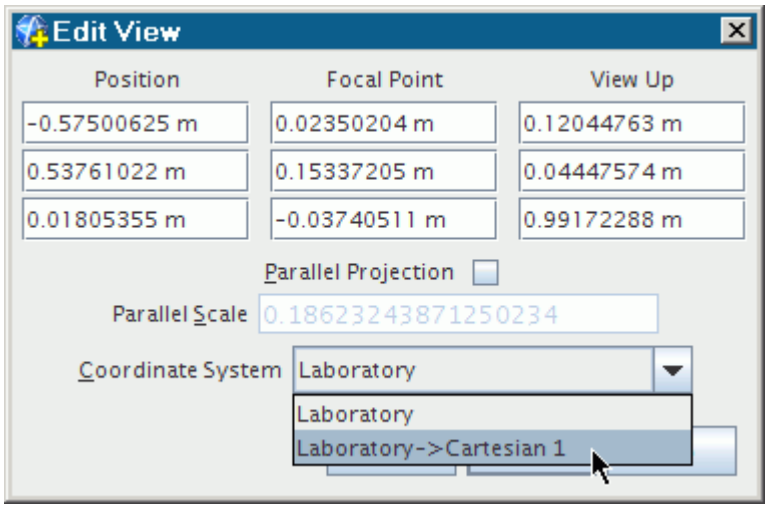
You can select one of the coordinate systems in the simulation to be used by a view. One benefit of doing this is to follow mesh motion.

Selecting a coordinate system can be done using the same techniques as other modifications:

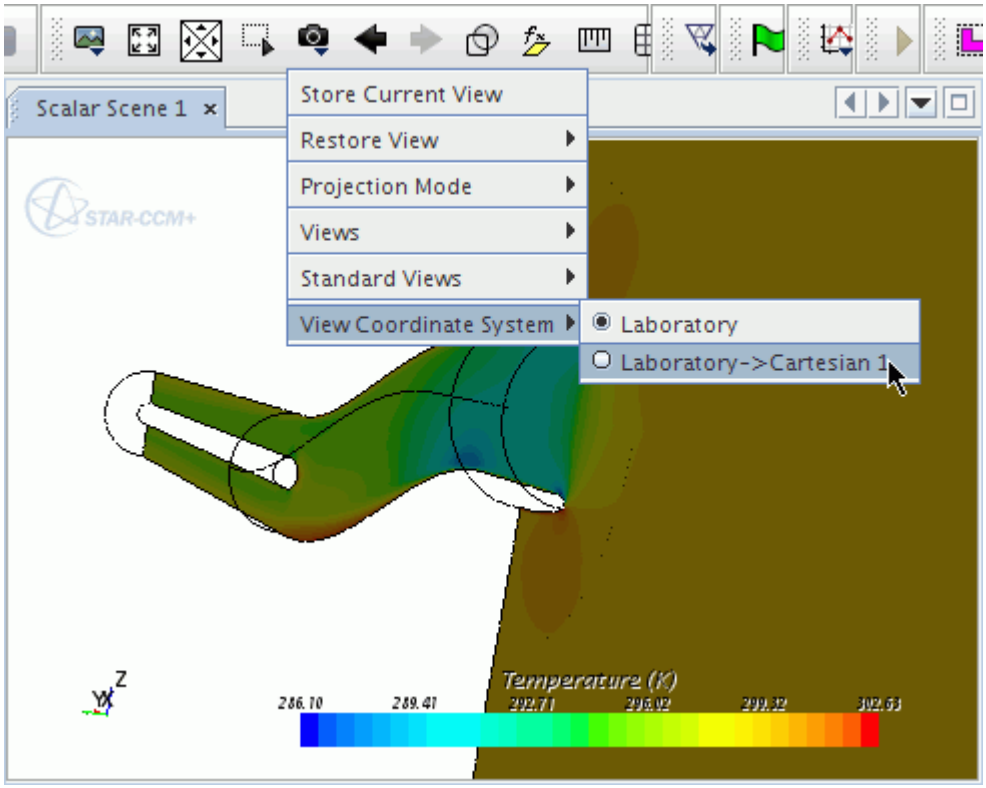
- In the *Properties* window



- In the *Edit View* dialog



- In the (**Save-Restore-Select Views**) menu



This menu setting changes, and can be changed by, the *Coordinate System* property setting for the scene's current view node. It does not affect the properties of existing saved view objects in the **Tools** > **Views** node.

If you save a view in a scene, the new view object is saved with the coordinate system currently used in that scene. At the same time, if you restore a view (that is, access a saved view object), you will replace the scene's current coordinate system selection with that of the saved view object.

Copying Properties of Views

When properties are copied from one view to another, all of the properties are applied so that the destination view matches the source view.

This has the same effect as restoring the view.

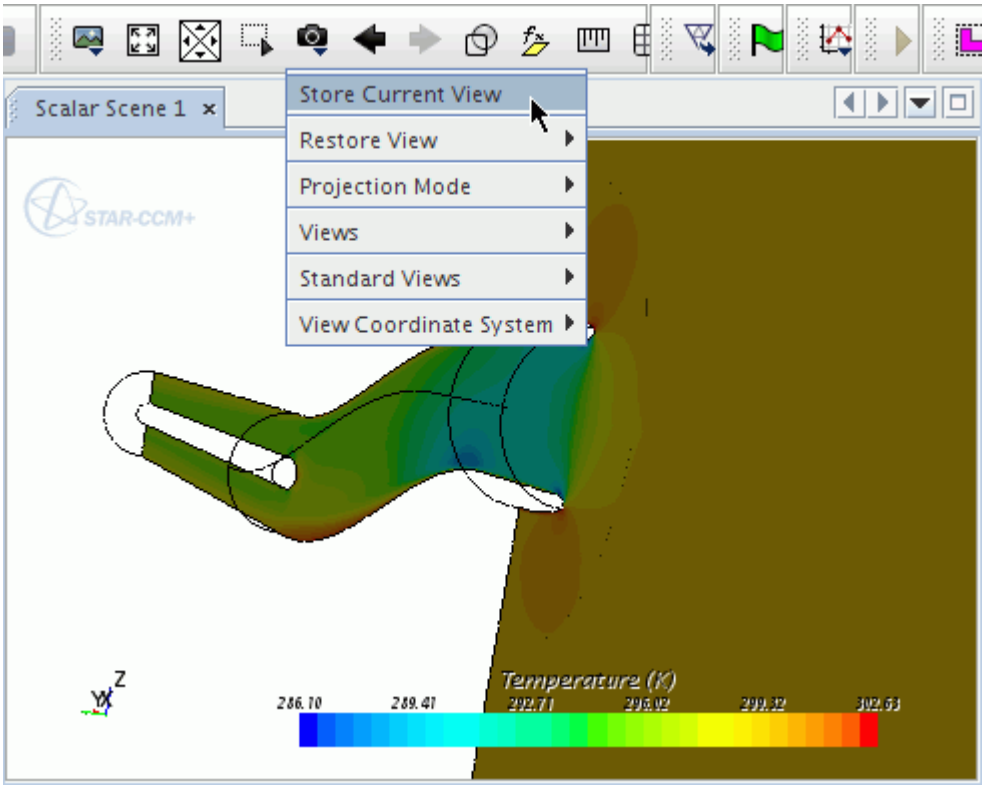
The drag-and-drop technique in particular can be used to copy properties of views in two ways. The typical way is to drag the source node onto the destination node, which requires navigating the tree. But an additional technique for views is to drag the view node into the visualization display of the destination scene.

Saving a View

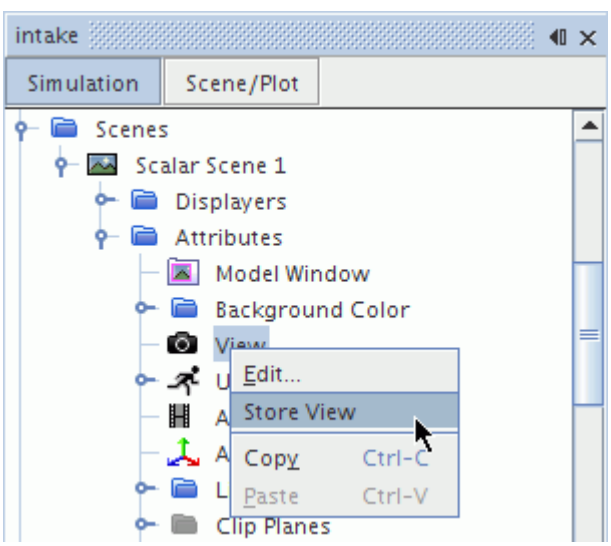
One of the most useful features of views is the ability to save them. This allows you to manipulate a scene and save an interesting view and then continue manipulating the scene.

The view may then be recovered without additional manipulation of the scene. However, only the viewing position of the object(s) displayed in the scene is saved; positions of two-dimensional annotations, scalar bars or other display features are not saved.

Once you have manipulated the scene so that interesting details are displayed, click on the **View** toolbar button and select **Store Current View**.



Alternatively, right-click the **Attributes** > **View** node in the object tree and select **Store View**.



A new view node will appear in the tree under **Tools** > **Views**. Expand the **Tools** node and then expand the **Views** node. The default name of the new view node will be **View** along with a number equal to 1 plus the number of view nodes already existing. However, it is possible to rename the view node.

Creating a View

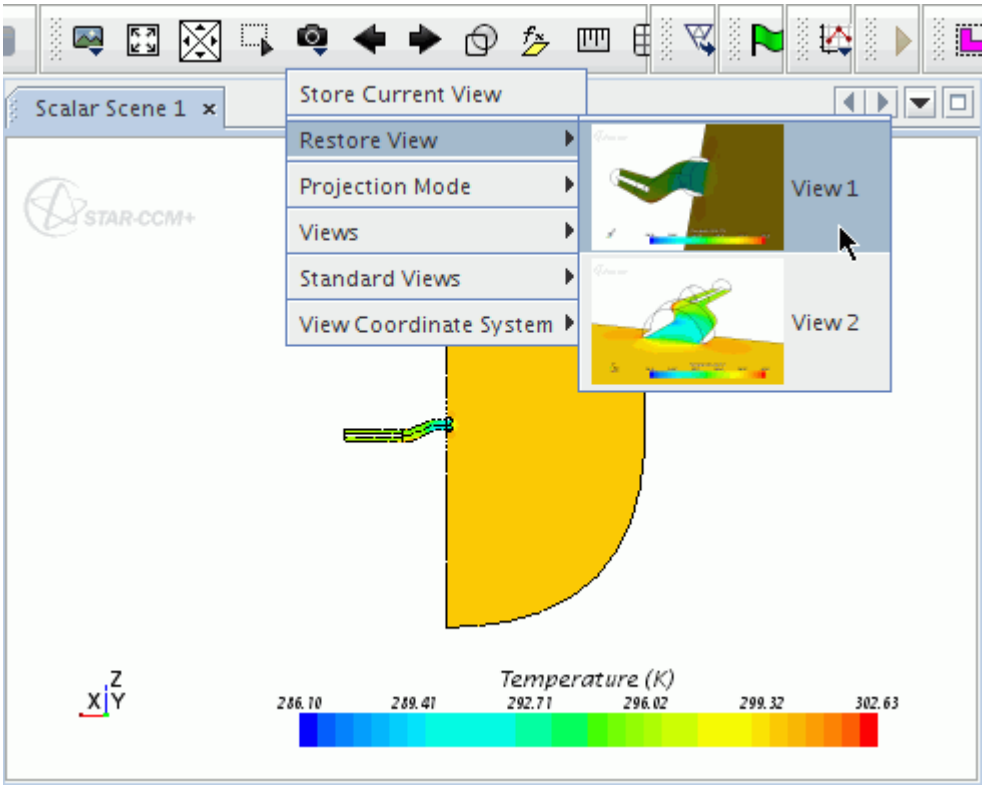
To create a new view object, right-click the **Views** manager node and select **New**.

This has the same effect as saving a view.

Restoring a View

Each of the saved views can be restored.

Click on the **View** icon in the Visualization toolbar and select the **Restore View** submenu of the (**Save-Restore-Select Views**) menu. A drop-down menu with image previews of each of the views will be shown. Once you have chosen a view the scene will orientate itself to that view. The view is panned, rather than snapped, into place so that you keep your visual context relative to the view you had. However, only the viewing position of the object(s) displayed in the scene is restored; positions of two-dimensional annotations, scalar bars or other display features are not saved in views.



Views can also be restored by copying them into the current view.

A view can be restored into any scene even though it was not saved from that scene.

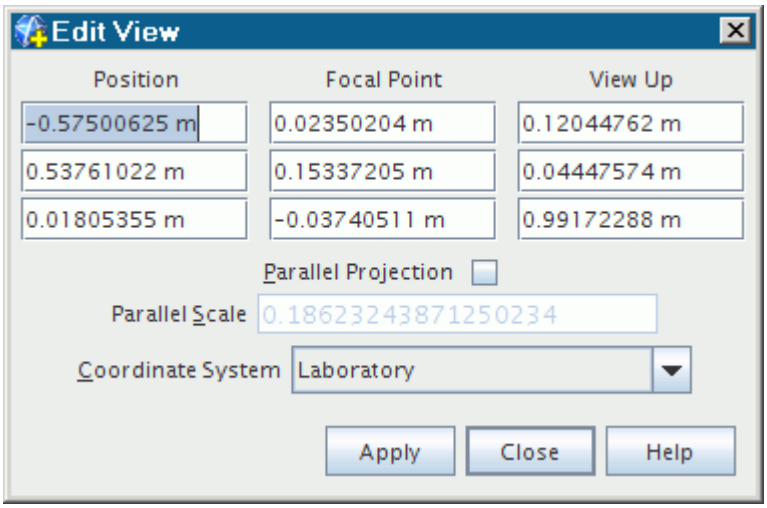
See Also:

Using the Distance Measuring Tool

Modifying Views

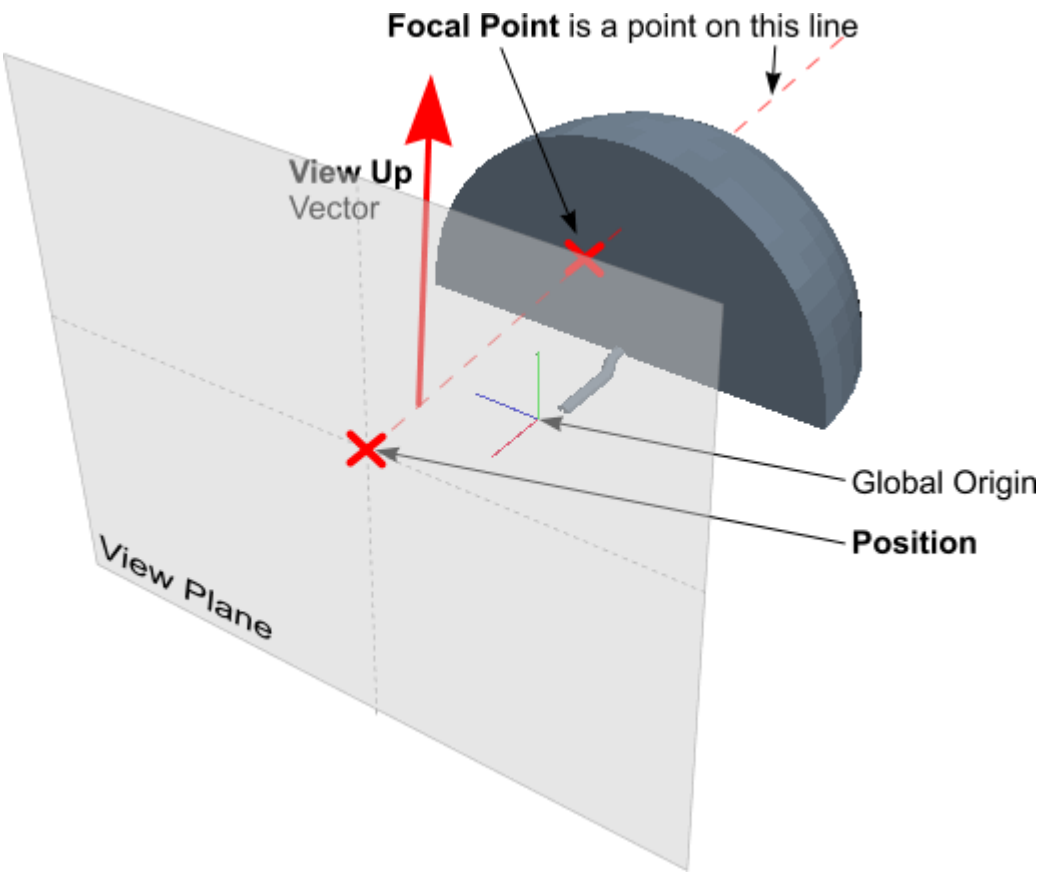
To modify a view simply select its node, right-click and select **Edit...**

This action makes the *Edit View* dialog available.



The *Parallel Scale* property in the dialog is used to scale the image in parallel projection mode (when the **Parallel Projection** checkbox is activated). The parameter works as an inverse scale, in that larger numbers will produce smaller images. Any user-defined value should be greater than zero. When you activate parallel projection mode, a parallel scale subnode is added to the **View** node.

The diagram below shows how these settings relate to the scene. Change the settings in the fields as desired and click **Apply**, then click **Close**.



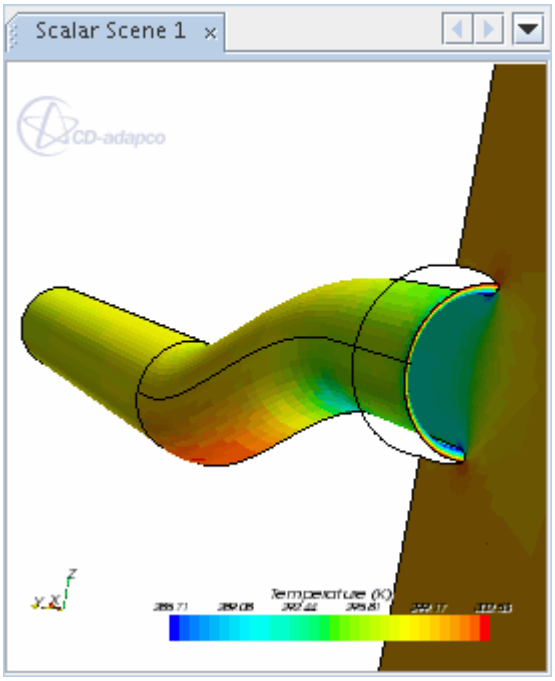
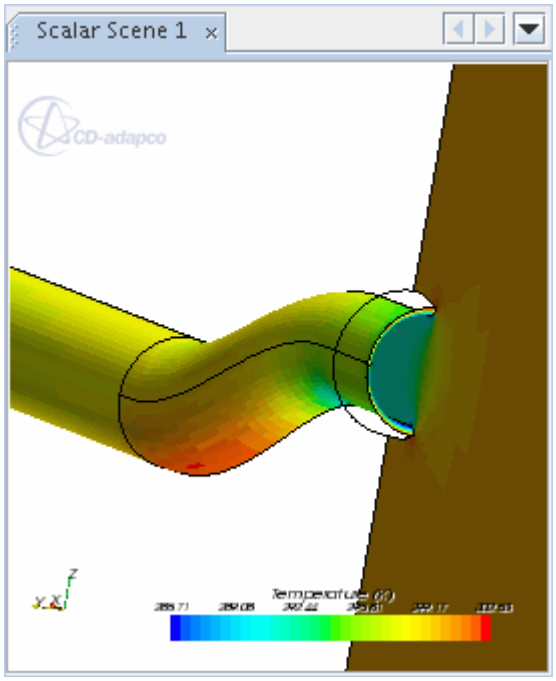
Note: STAR-CCM+ may change some of the settings entered here in order to resolve the view in the scene.

You can also modify a view by selecting the **View** node and editing its properties. This technique can only be used to edit individual properties, and is not recommended for changing the *Position*, *Focal Point* or *View Up* properties; in this case the method described above should be used instead.

Changing the Projection Mode

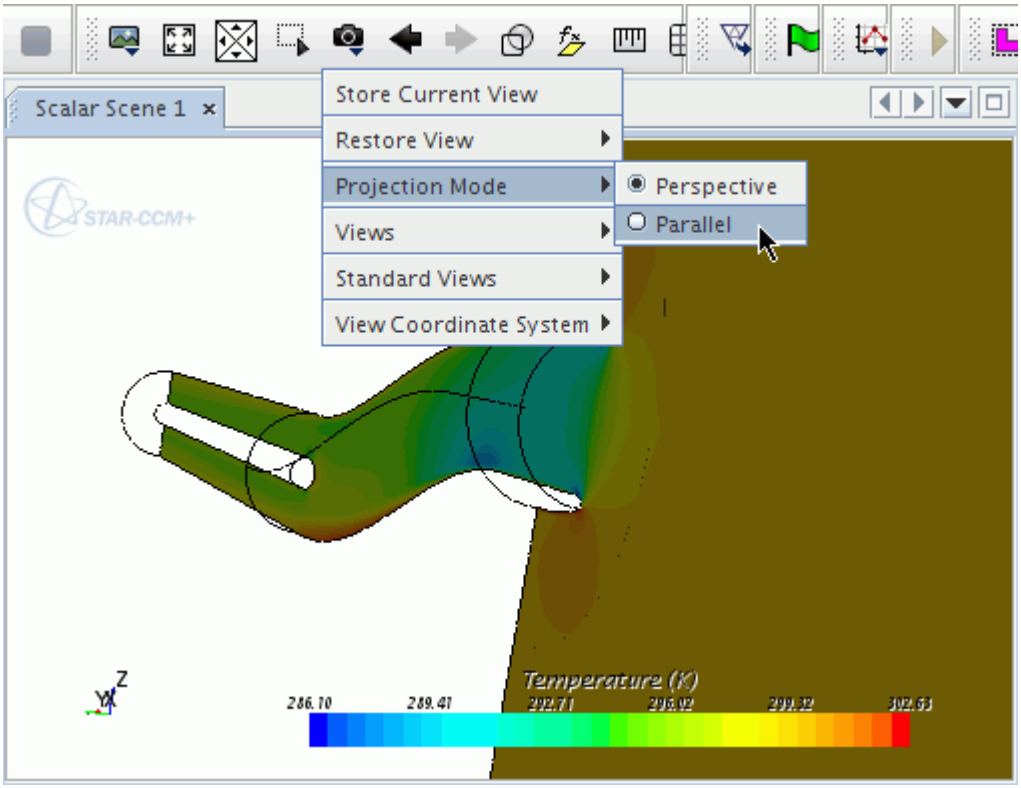
Each scene can be viewed in two projections modes; parallel projection, and perspective projection.

The perspective projection will provide a realistic view of the scene whereas the parallel mode will look less realistic and will be similar in style to technical diagrams.

	
Perspective projection mode	Parallel projection mode

By default the scene is viewed using the perspective projection mode. To change to parallel projection mode in the current view expand the **Scenes** node, expand the scene node and click on **View**. Under the *Projection Mode* in the *Properties* window choose **Parallel** under the pull-down menu. The view of the scene will automatically update.

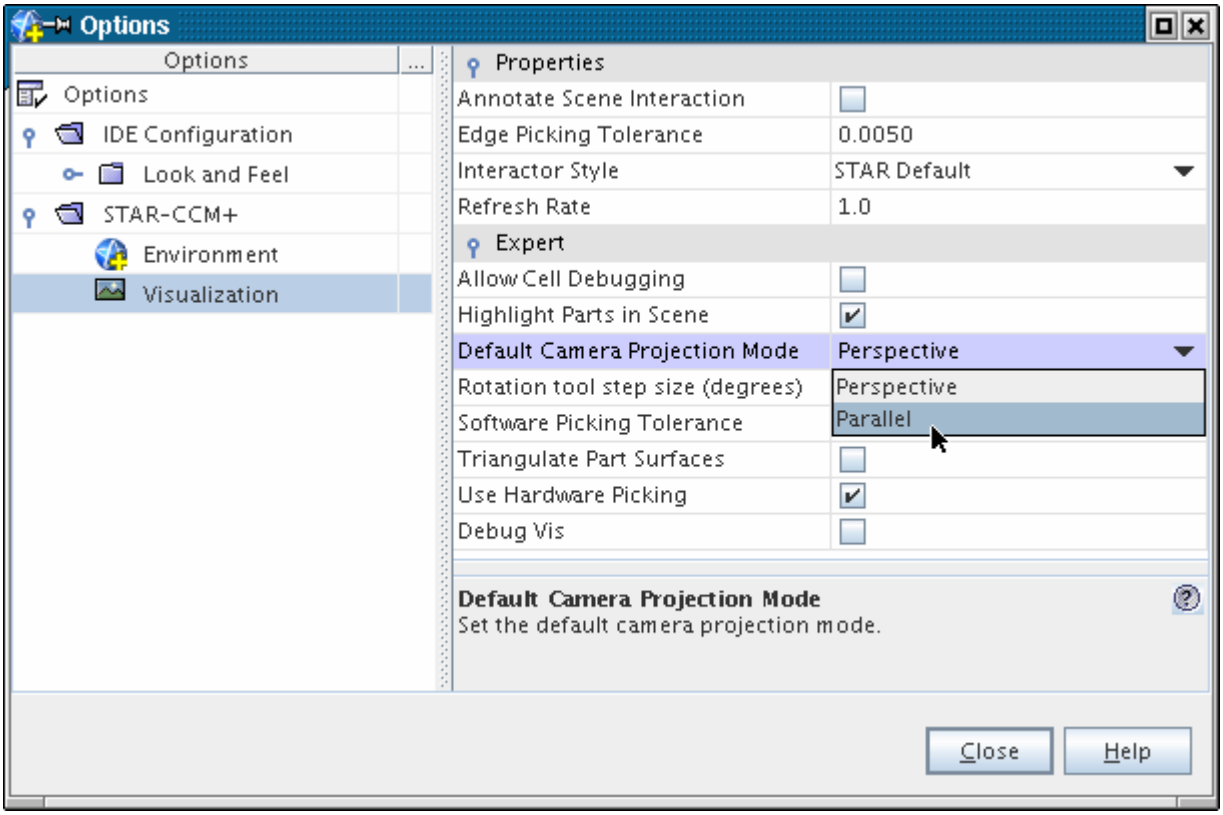
Another approach involves the **Projection Mode** submenu of the (**Save-Restore-Select Views**) menu. This setting changes, and can be changed by, the *Projection Mode* property setting for the scene's current view node.



These actions also add a parallel scale subnode to the **View** node.

The projection mode of saved views may also be changed. Expand the **Tools** node, expand the **Views** node and click on the chosen view. You may then change the mode using the same pull-down menu. However, if that view is used by a scene and you want the scene to reflect the mode change, then you must refresh the scene's current view by restoring the view.

You can also change the default projection mode. Select **Tools** > **Options**. This activates the *Options* dialog.



Open the STAR-CCM+node in the **Options** node. Select the **Visualization** node and use the drop-down list of the *Default Camera Projection Mode* property to change the default projection of the object in the display.

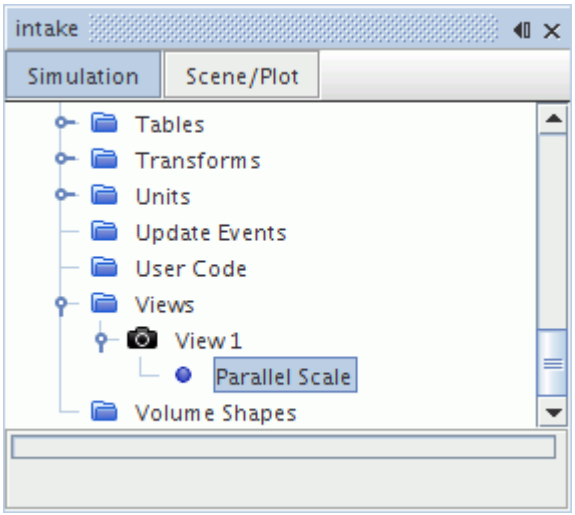
Contents:

Setting the Parallel Scale

Setting the Parallel Scale

The **Parallel Scale** subnode of the **View** node is used to scale the image in parallel projection mode.

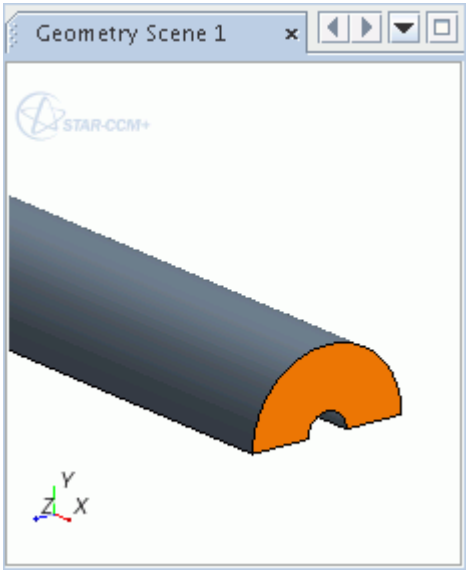
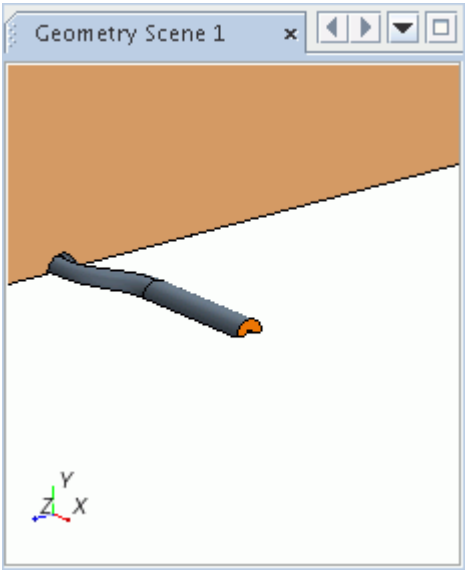
It is made available when you set the *Projection mode* property of **View** node to `Parallel`. You can specify the scale in this object's properties.



Parallel Scale Properties

Parallel Scale	The user-defined value of the scale.

The parameter works as an inverse scale, in that larger numbers will produce smaller images. Any user-defined value should be greater than zero.

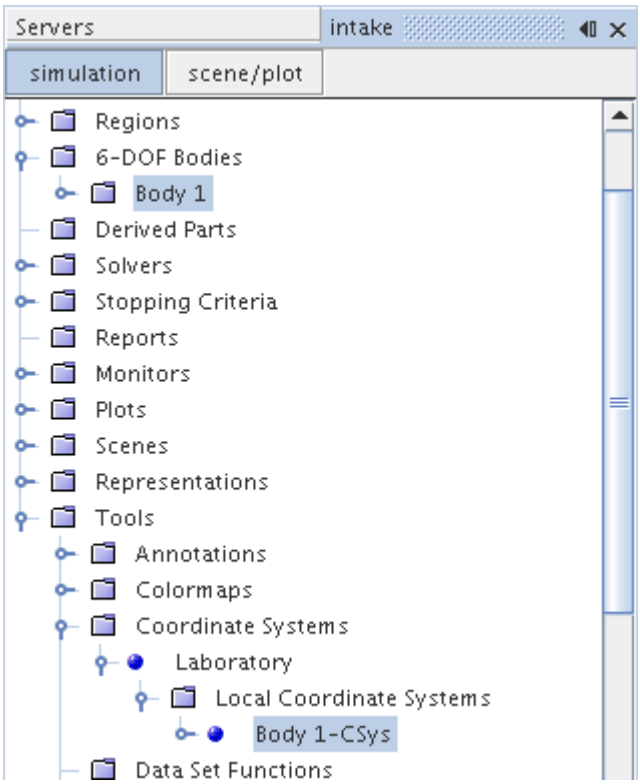
	
Setting of 0.2	Setting of 1.2

Following Mesh Motion in a View

If you are using DFBI motion, you can make a view follow the motion of the mesh.

This is done by linking a view to a 6-DOF body through its coordinate system. During iteration, the 6-DOF body's movements update that coordinate system, which in turn keeps the view centered on the 6-DOF body.

To make this work, first create a 6-DOF body. This object will have a body-fitted coordinate system created automatically along with it.



This type of coordinate system is updated by the movements of the 6-DOF body with each iteration.

In the *Coordinate System* property of the view object, select that coordinate system. This selection will associate the view with the 6-DOF body.

Finally, add the view to a scene by dragging the view node into the visualization display of the destination scene. That will cause the display to be updated with the 6-DOF motions.

Pop-Up Menu of the Views Manager

These are the pop-up menu items available when you right-click the **Views** manager node.

New View	Executes the create view action to add a view to the **Views** node.

See also: common menu items.

Obtaining Benchmark Information for a Scene

You can get a report about the rendering of a particular scene. For you to access this report, the scene must be open.

To obtain display and performance information for the scene:

1. Right-click the node of the scene, or in the scene display, and select **Benchmark**.
2. In the *Benchmark Properties* dialog, decide how many scene render passes you want to use as the sample for statistics of rendering time. Set this number in the *Number of Render Passes* drop-down list.
3. Click **OK**.

   The report appears in a separate tab in the *Output* window with the following details about how the scene renders:

```
***Geometry Scene 1 Graphics Benchmark***
Image Resolution: 694 x 232
Camera:
  Focal Point: [-0.102521, 0.083506, 0.526289]
  Position: [-0.325932, 0.110643, 0.78934]
  View Up: [0.0406461, 0.996836, -0.0683144]
  Projection Mode: Perspective
Internal render times computed over 5 render passes:
  Average: 0.0108 seconds (variance = 1.36e-06)
  Note: reported timing does not reflect scene initialization,
streamline generation, or any other computations which may increase
the total time to create a scene's first rendering.
Geometry Rendered with OpenGL:
  Lines: 17177
  Triangles: 15570
Advanced Rendering Disabled
***End Geometry Scene 1 Graphics Benchmark***
```

   These details include the following:

   - `Image Resolution`: the actual scene display resolution, which changes as you resize the scene display
   - `Camera`: the properties of the **Attributes** > **View** sub-node of a scene node. For details, see Working with the Current View of the Scene.
   - Average render time, calculated based on the number of render passes that you select
   - Numbers of objects that compose the scene geometry
4. When advanced rendering is activated, details about it are available in the report:

```
***Geometry Scene 1 Graphics Benchmark***
Image Resolution: 694 x 402
Camera:
  Focal Point: [0.858327, 0.285087, 0.814146]
```

```
  Position: [-5.19194, 5.84705, 2.17382]
  View Up: [0.160156, -0.0665372, 0.984847]
  Projection Mode: Perspective
Internal render times computed over 5 render passes:
  Average: 0.0626 seconds (variance = 8.424e-05)
  Note: reported timing does not reflect scene initialization,
streamline generation, or any other computations which may increase
the total time to create a scene's first rendering.
Advanced Rendering Enabled:
  Threads Used: 8
  Average CPU Rendering Times:
    Read pixels from OpenGL: 0 seconds (variance = 0)
    Write pixels to OpenGL: 0.00140004 seconds (variance =
2.40042e-07)
    Render and misc: 0.0368 seconds (variance = 4.41599e-05)
  Geometry Rendered:
    Triangles: 491024
  Computations Per Render Pass:
    Total Rays: 444485
    CPU Volume Render Samples: 0
  Data Structure:
    Build Time: 0.039 seconds
    Memory Usage: 74.6729 MBs
***End Geometry Scene 1 Graphics Benchmark***
```

The advanced rendering information includes the following:

- `Threads Used`: the setting in the *Number of Client Render Threads* default option, or in the command line option `-renderthreads`. For more information, see Advanced Rendering Workflow.
- `Average CPU Rendering Times`: These numbers are specific to the advanced rendering pipeline. The times are also reflected in the average rendering time displayed earlier.
- `Geometry Rendered`: This information is similar to that in the `Geometry Rendered with OpenGL` section, but is part of the advanced rendering pipeline.
- `Computations Per Render Pass`:
  - `Total Rays`: total rays traced for a single render pass
  - `CPU Volume Render Samples`: If this number is greater than zero, then CPU volume rendering is active.
- `Data Structure`:
  - `Build Time`: the time to build the advanced rendering data structure. This is a one-time overhead cost that occurs when certain properties in the scene, including the geometry, are changed.
  - `Memory Usage`: the memory usage for the advanced rendering data structure

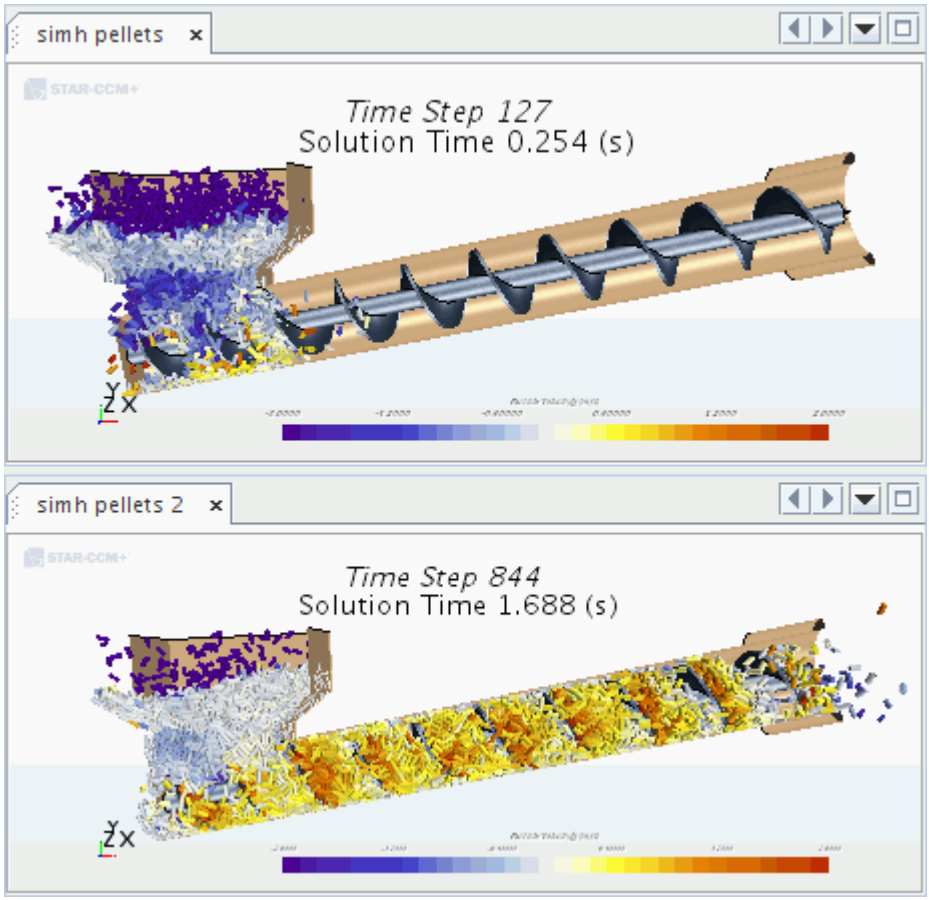
    Note: This value may not include memory for actual geometric representations such as the triangle mesh.

Linking Scene Views

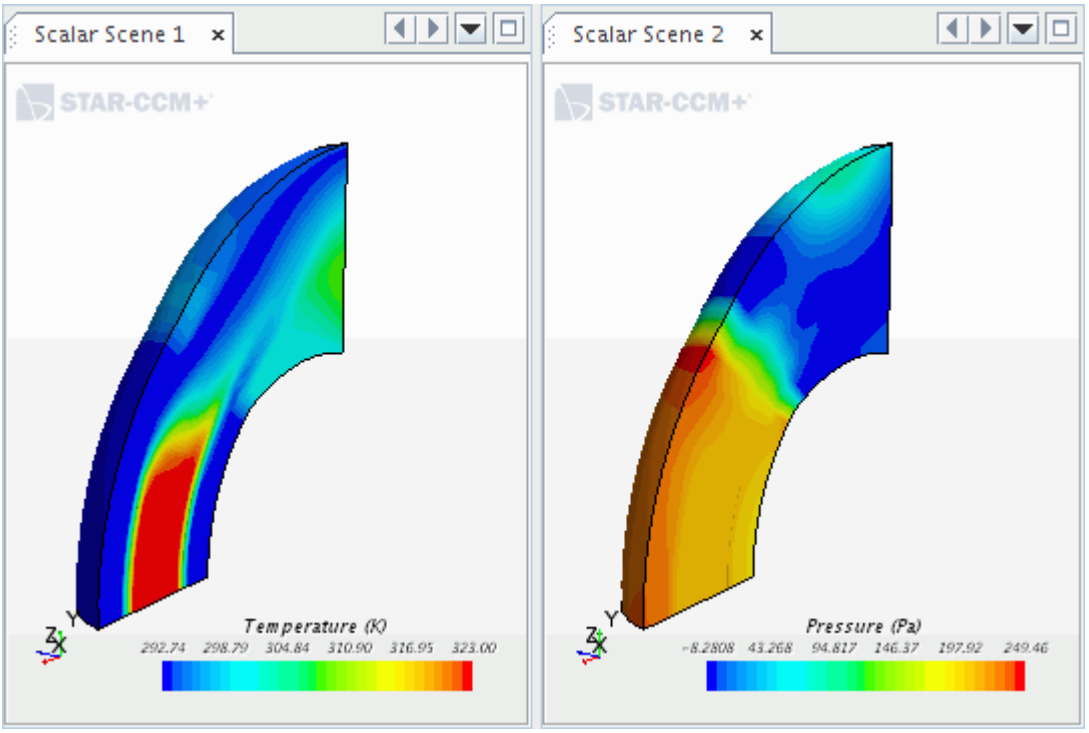
To improve understanding of results, you can link multiple scenes so that they move in tandem as you interact with one of them. This visual parallelism allows you to compare and contrast information concurrently without trying to remember views.

The uses of this feature include:

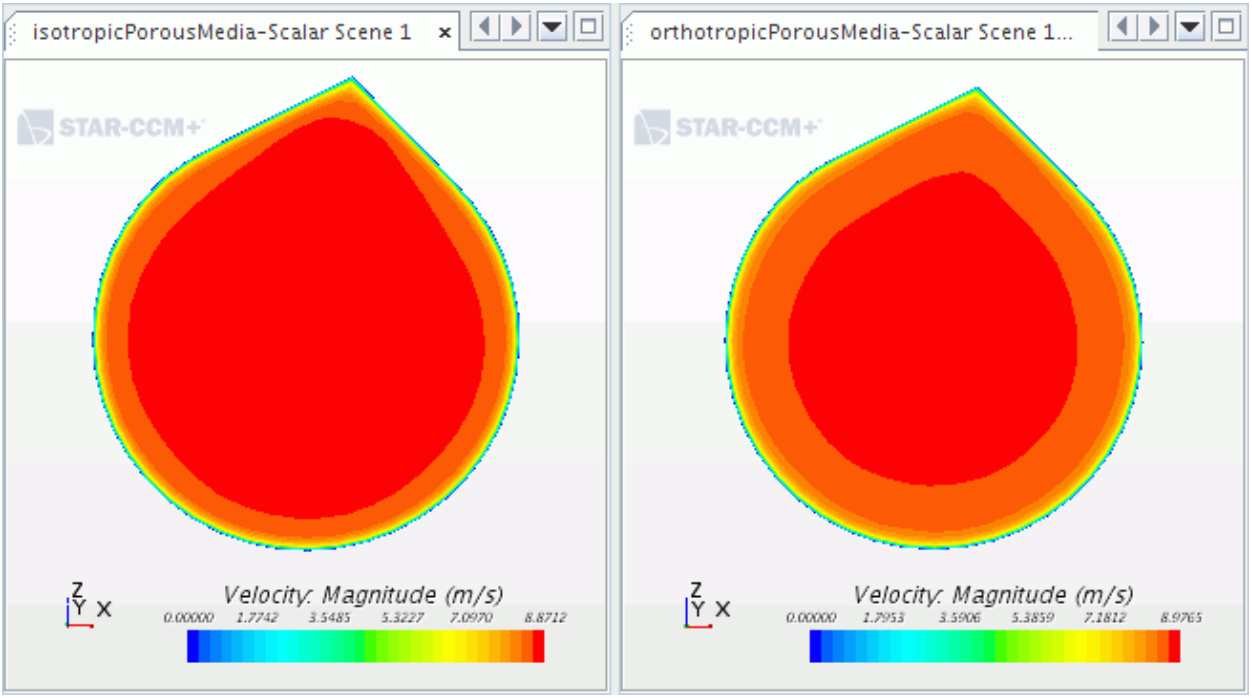
- Different solution times using the same dataset (using the simulation history and solution view):



- Different scalars using the same dataset:



- Different datasets with the same scalar (these images are created from two separate simulation files):



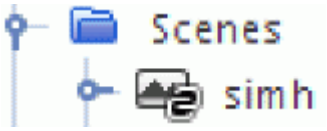
When you are comparing the same scalar, specify the same range in the scalar displayers of each linked scene. That way, the colors represent the scalar values consistently.

Using Scene Links

To set up a link among scenes in a simulation:

1. For every scene you wish to link, right-click the scene node (or inside the open scene display) and activate the **Linked View** option in the context menu.

   A chain-link icon appears at the bottom right corner of the scene node icon.



2. To compare different states of a solution history, prepare a duplicate set of objects as follows:
   a. Make a copy of the scene that shows the solution view.
   b. Create another solution view by right-clicking the solution history node and selecting **Create Recorded Solution View**.
   c. Select the representations of the new solution view for use by the displayers, as well as any annotations, in the new scene.
   d. Set the *Time Step* property of each solution view as desired.
3. To compare views between simulations, open both simulations and follow the steps.

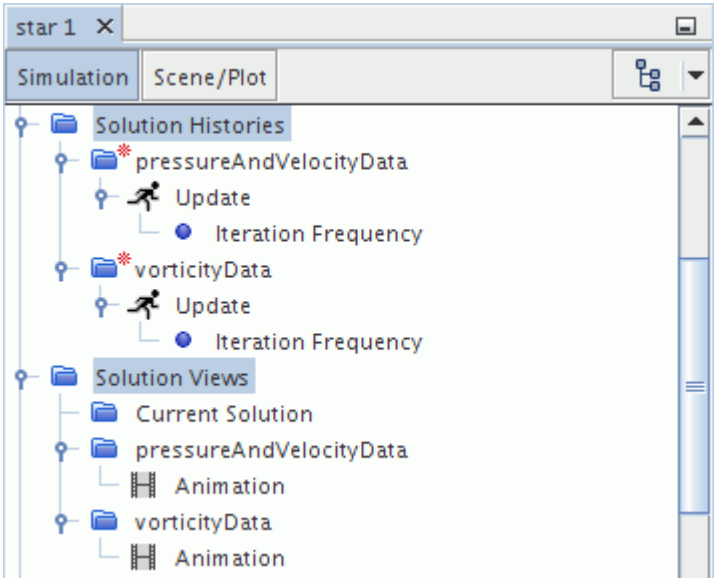
   Note: If you have multiple simulations with scene-linking activated, and you do not want to change the views simultaneously, open each simulation in a separate session of the STAR-CCM+ client.

4. To stop simultaneous viewing, deactivate the **Linked View** option.

Recording and Viewing Solutions

STAR-CCM+ can save selected scalar and vector functions to a simulation history file while a simulation is running.

You can control the choice of functions, file location, and frequency of writing, in the simulation history nodes that are located within the **Solution Histories** manager node.



To access the data that is stored within a simulation history file, create and use a solution view. Solution views are located within the **Solution Histories** manager node. You can create and view snapshots of the current solution in steady state simulations, which is useful for comparing the results between different design iterations. You can animate the recorded solution data and record this animation to file. You can also use recorded solution data in reports and plots. Information from the simulation history file can be used to annotate scenes to display the iteration number, the solution time, and the time step of the displayed state.

Contents:

Understanding the Simulation History File
Recording Solution Data
Retrieving Results from the Simulation History Files
Using Solution Snapshots
Animating Solution Views

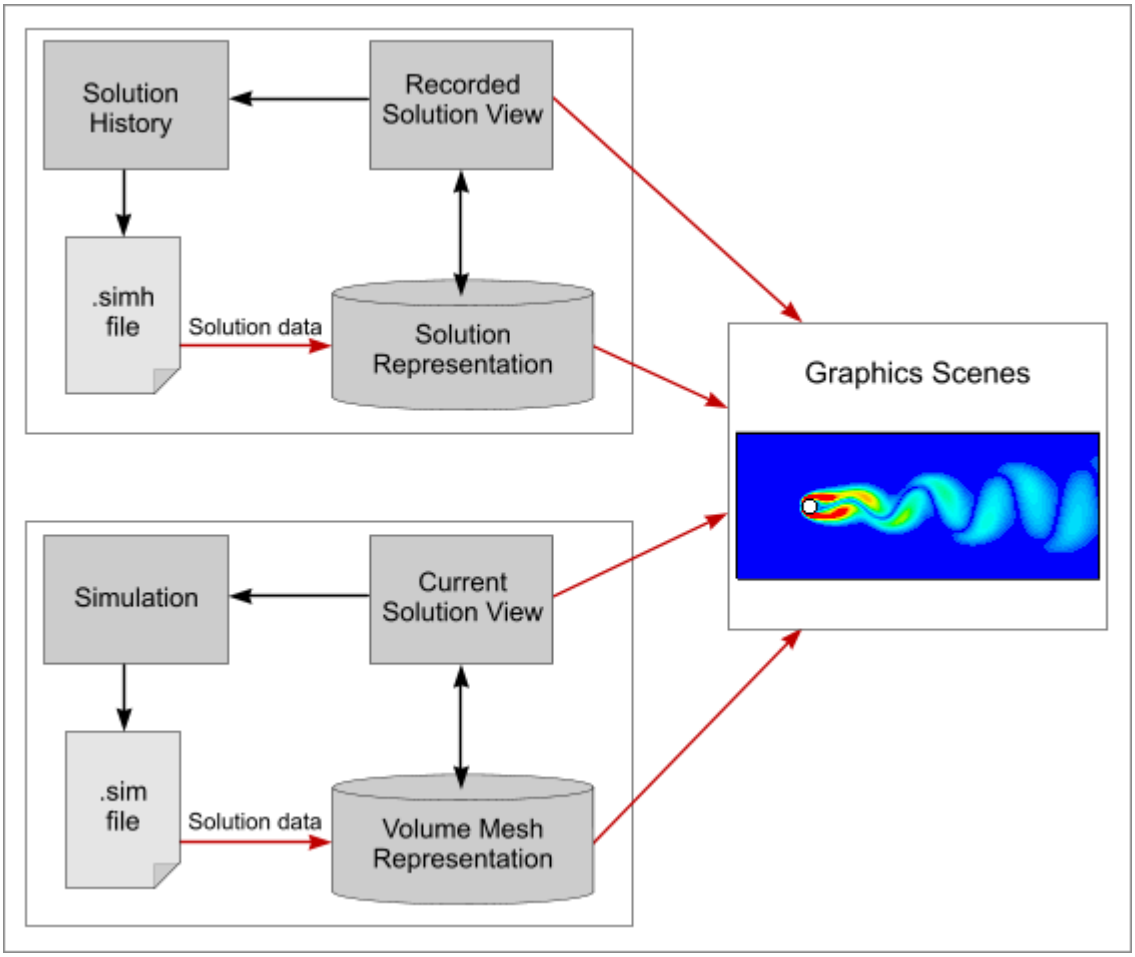
Understanding the Simulation History File

The solution data that you select is saved to an external file, called the simulation history (`.simh`) file, at the specified time intervals.

The simulation history file grows in size as more solution data is saved to it. Simulation history files are designed to work with the simulation (`.sim`) file that created them; simulation history files do not work with different or empty simulation files.

In addition to the solution data, each simulation history file can contain a copy of the volume mesh, the standalone boundary surfaces, or both. When recording solution history, the mesh is only written when it changes. If only the vertex coordinates change for the same mesh connectivity, such as during morphing or rigid body motion, only the modified coordinate array is written in the subsequent states. This behavior is true for recorded snapshots and automatic recording to a simulation history file.

A diagram demonstrating how the data is handled in STAR-CCM+ is shown below.



Additionally, no solution settings, such as solver parameters, are contained in this file.

Moving or deleting the simulation history file breaks the link between it and the simulation file. Any scenes that display data from the simulation history file are not rendered. If the link between the simulation history file and

the simulation file is broken, you can re-establish it by right-clicking on the **Solution Histories** node and selecting **Load**. You can then navigate to the simulation history file and load it into the simulation.

Recording Solution Data

Use the **Solution Histories** node to create and configure simulation history files. Solution data that you select is stored in these simulation history files.

The simulation history files that you create are located under the **Solution Histories** manager node. You can select what data is stored in these files and how often the data is saved to file.

Each simulation history file node contains two sub-nodes: the **Update** node, that controls how the update frequency is defined, and the **Update Frequency** node, where you can specify how often the updates take place.

Contents:

Creating a Simulation History File
Configuring the Simulation History File
Selecting the Update Method and Setting the Update Frequency
Clearing the Solution

See Also:

Understanding the Simulation History File

Creating a Simulation History File

You can store specified simulation data in simulation history files.

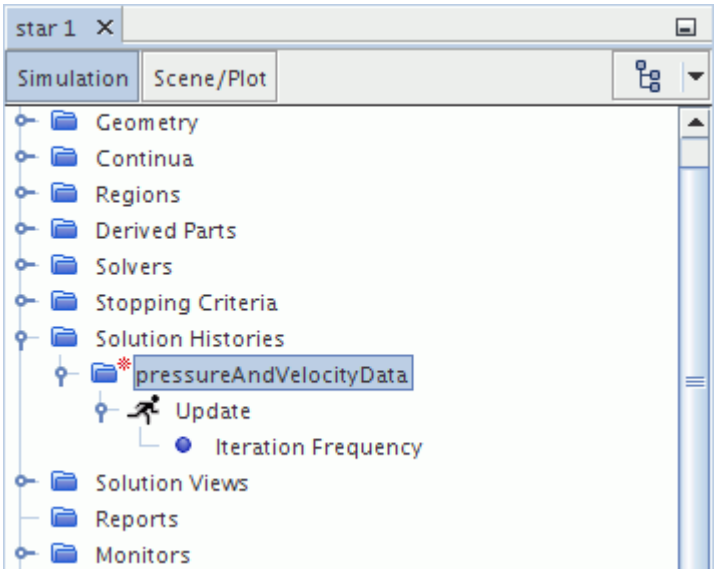
To create a simulation history file:

1. Right-click on the **Solution Histories** node and select **New...**

   The *Save* dialog appears and asks you where you would like to save the simulation history file.
2. Choose a location that has adequate space and give the file an appropriate name.

   Note that you must have write access to the location you choose, otherwise the solution data cannot be saved to the file.
3. Click **Save** to create the simulation history file and close the dialog.

A new sub-node with the name of the simulation history file appears in the **Solution Histories** node. In the following example, the simulation history file was given the name `pressureAndVelocityData`.



The red asterisk next to this node indicates that STAR-CCM+ is set to write data to the simulation history file when you run the simulation.

The following actions are available in the context menu of the simulation history node:

Edit...	Brings up an Edit dialog from where you can modify all the parameters of the simulation history file in one operation.
Set New Tag...	Enters a tag to associate with solution history objects. See Tagging Simulation Objects.
Create Snapshot	Creates a snapshot of the current solution and adds it to the simulation history file. This option is useful for comparing different design iterations.
Create Recorded Solution View	Creates a recorded solution view sub-node in the **Solution Views** node. From here, you are able to control the time of the solution that is stored in the simulation history file.
Rescan	Rescans the simulation history file to determine whether there were any states added to it since you last used the file.
Clear	Clears the solution history data from the simulation history file. The simulation history file is deleted and recreated the next time data is written to it, either through a manual capture (snapshot) or automatically (auto-record).
Delete	Removes the simulation history file from the STAR-CCM+ object tree; this action does not delete the simulation history file from the disk.
Rename	Brings up a window from where you can rename the node in the STAR-CCM+ object tree. This action does not affect the simulation history file on disk; the original name of the file is retained.

Configuring the Simulation History File

You can configure the simulation history file by adjusting file properties through the *Properties* window. You can select what scalar and vector data you want to save and choose the parts from which to save that data:

one or more regions, boundary surfaces, Lagrangian phases, DEM Phase Interactions, certain derived parts, or a combination of these.

Available derived parts for the simulation history are:

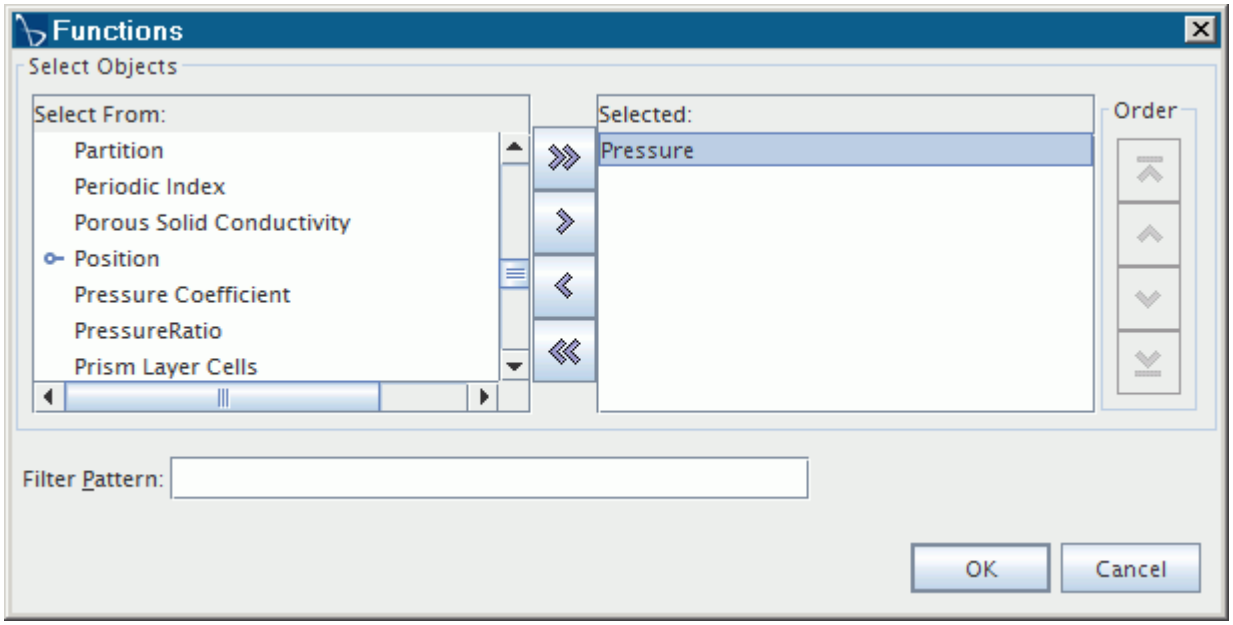
- Isosurfaces
- Probes:
  - Point
  - Line
  - Presentation grid
  - Arbitrary
- Sections:
  - Plane
  - Constrained plane
  - Cylinder
  - Sphere
  - Arbitrary
- Streamlines and constrained streamlines
- Warps

Note: A region is selected by default for the simulation history file. To export objects other than the region, deselect the region and select the other objects.

To record the desired scalar or vector data:

1. Click the ... (**property customizer**) button located next to the *Functions* property.

   A selection manager dialog appears.
2. Use the **>** (**Add selected**) button to select the desired scalars or vectors. The selections are added to the right-hand side column.



3. Click **OK** to close the dialog.
4. To include all auxiliary field functions in the simulation history file, activate the **Export for reporting** property before you record.

Using the same technique, you can select the regions, boundaries, and derived parts from which this data comes.

Further information on simulation history file properties is provided in the following table:

Simulation History Properties

Auto-record

When activated, the selected solution data is written to the simulation history file at the specified time intervals. When this option is activated, an asterisk appears besides the simulation history file node in the object tree. This property is activated by default.

Export for reporting

When activated, the export includes any auxiliary field functions that are required for all standard scalar and vector reports. This property is deactivated by default. See Simulation History Phase Data.

When you are exporting data from a case that has overset meshes, and you include standalone boundaries, activate this property. When activated for an overset case, STAR-CCM+ automatically includes the field, **Visualization Cell Status** among the selections for the *Functions* property. Thereafter, when visualizing this data from the simulation history file, STAR-CCM+ uses the V**isualization Cell Status** to correctly display the standalone boundary. That is, faces that are culled by the overset method are not shown in the display.

Export inputs at non-native locations

Activate this property when you include standalone boundaries and surfaces in the simulation history file, and when you are likely to require smoothed data scenes in subsequent uses of the simulation history

data. For Finite Volume meshes, this property causes STAR-CCM+ to export data at vertices of standalone boundaries and surfaces in addition to data at face centers. The data is interpolated to vertices before export. For Finite Element meshes, this property causes STAR-CCM+ to export data at face centers in addition to data at vertices (after interpolation).

Note: Activating this option can significantly increase the size of the simulation history file.

Functions

Allows you to select the desired scalar or vector field functions you wish to save to the simulation history file. See "Simulation History Phase Data" for Lagrangian Data and for DEM Data.

Regions

Allows you to select the regions that you wish to save to the simulation history file.

Inputs

Allows you to select the boundaries, derived parts, Lagrangian phases, and phase interactions that you wish to save to the simulation history file.

Note: The boundaries, derived parts, and phases that you select with this option are exported as standalone objects. Their underlying 3D source data does not get exported. This omission reduces the size of the simulation history file to which they are exported.

Path

Displays the relative path to the simulation history file where the solution data is written. Clicking ... (**Property Customizer**) displays the absolute path to the simulation history file. You cannot edit this path.

State

Displays the number of saved states that are present in the simulation history file.

Tags

Allows you to associate the displayed tags with simulation objects. See Tagging Simulation Objects.

Selecting the Update Method and Setting the Update Frequency

In the **Update** sub-node of the simulation history file node, select the trigger (the method that defines how often the solution data is saved to the simulation history file). See Trigger Types.

To control how often the solution data is saved to the simulation history file, use the **<Trigger> Frequency** node. See Trigger Properties.

For more information on selecting the trigger type and defining the update frequency, refer to Setting the Trigger for an Update.

Clearing the Solution

Clearing the solution from a simulation file also clears the data that is stored in the simulation history file. Only data that was stored by the *auto-record* property is cleared. If a simulation history file contains saved snapshots, no solution data is cleared.

Caution: If you do not want to lose the solution data that is stored in the simulation history file, deactivate the *Recorded Solution History* checkbox in the *Clear Solution* dialog window. If this option is activated, the simulation history file will be removed from your disk—this action is irreversible.

The solution history settings are retained in the simulation file. Next time you run the solver, the simulation history file will be created with the same settings as before it was deleted.

Retrieving Results from the Simulation History Files

Use the **Solution Views** node to create solution views. Solution views allow you to retrieve and view recorded solution data that is stored on a simulation history file, and control the display of recorded solution data.

You can create multiple solution views that reference the same simulation history file. This capability is useful for performing side-by-side comparisons of different case set-ups or design iterations. In each solution view, you can choose to display the transient data that is stored in each state, or the snapshots that you created.

Caution: Simulation history files created in STAR-CCM+ v7.04 or earlier are missing data at interfaces for cases run in parallel. In order to restore the missing data, clear the simulation history files and re-run the simulation in a later version of STAR-CCM+. No action is required for cases run in serial.

Contents:

Creating a Solution View
Applying Recorded Solution Views to Scenes
Visualizing Recorded Solution Data
Reporting and Monitoring Recorded Solution Data
Using Recorded Solution Data with Reports
Using Recorded Solution Data with Plots

See Also:

Using Solution Snapshots

Creating a Solution View

Solution views allow you to retrieve and view recorded solution data that is stored on a simulation history file.

To create a solution view:

1. Right-click on the simulation history file sub-node (in the **Solution Histories** node), and select **Create Recorded Solution View**.

   A new node that is named after the simulation history file is added to the **Solution Views** manager node.

You can apply the recorded solution to scenes, reports, and plots. You can control the solution that is displayed through its Properties window. The **Current Solution** node is automatically generated and contains the data for the current solution (the last iteration) in the simulation file.

When you create a solution view, a new representation is also added to the **Representations** node.

This representation contains a copy of the volume mesh or standalone surface that is stored in the simulation history file. For a surface, the representation indicates the region to which the surface belongs. A solution view is associated with its representation. The properties of the solution view determine which data is taken from the simulation history file and stored in the representation. Changing the properties in the solution view node updates the representation with the new data from the simulation history file.

See Also:

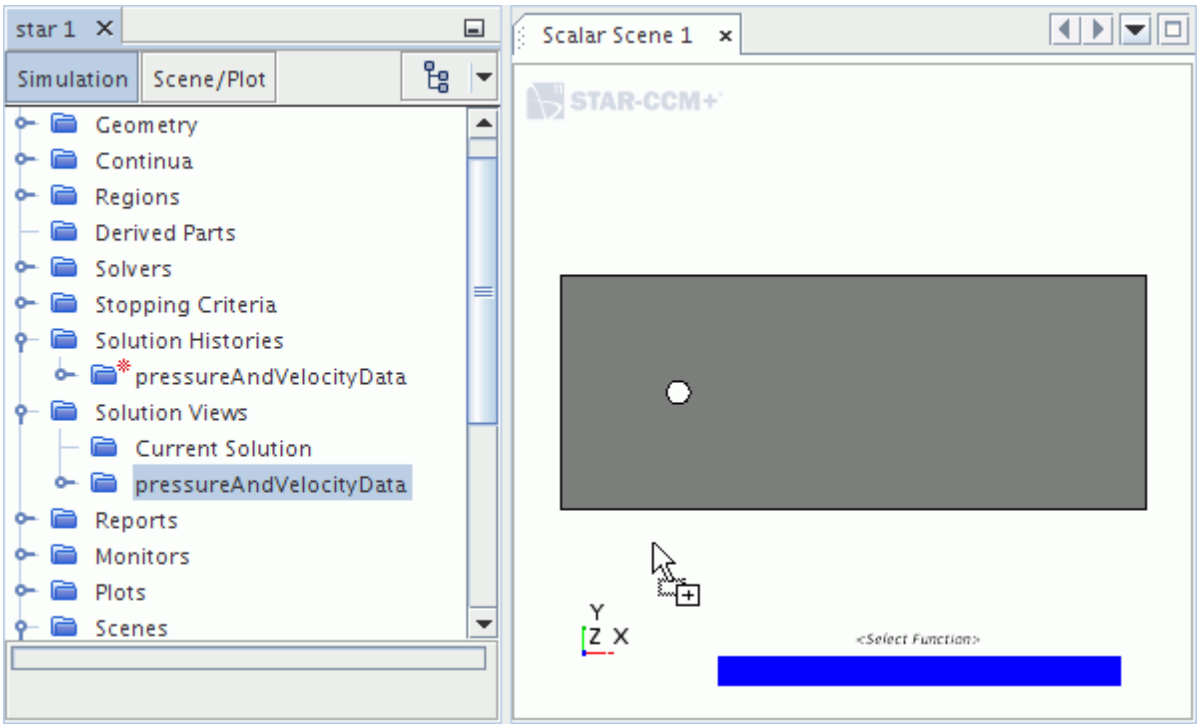
Applying Recorded Solution Views to Scenes
Using Recorded Solution Data with Plots
Using Recorded Solution Data with Reports
What Are Representations?

Applying Recorded Solution Views to Scenes

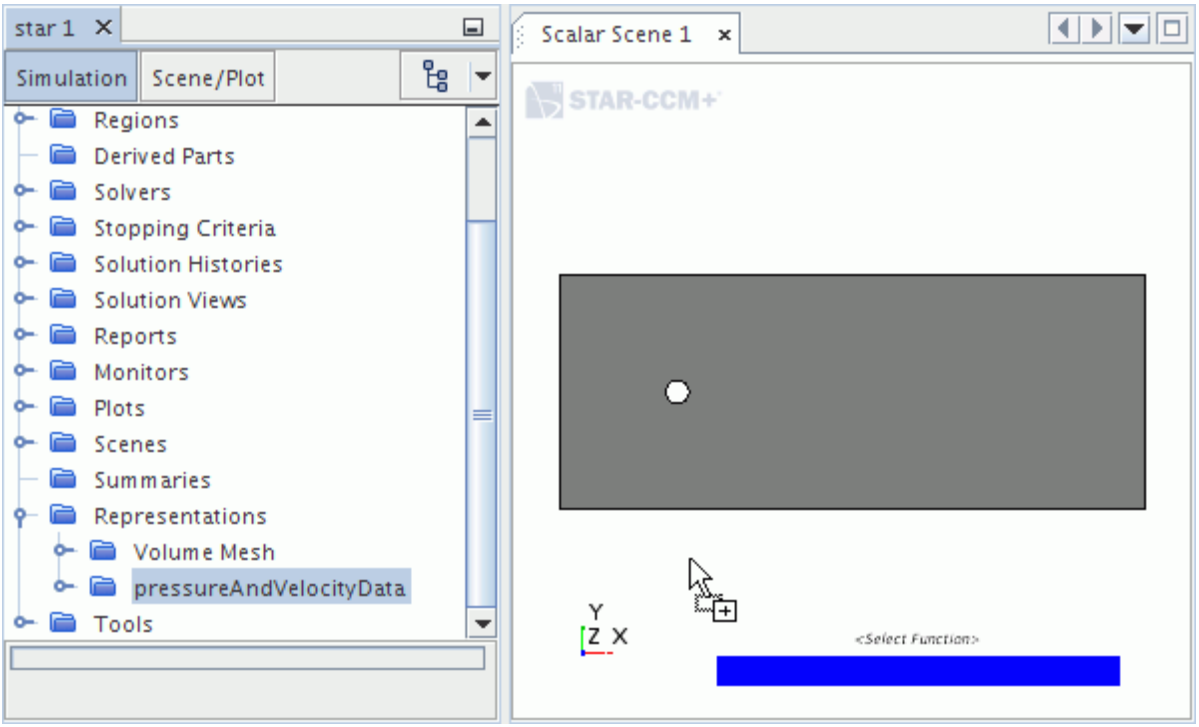
Recorded solution views contain stored data from your simulation. You can view this stored data by applying it to scenes.

To apply a recorded solution view to a scene:

1. Do one of the following:
   - Drag the recorded solution view from the object tree into the scene.



   - Drag the solution view representation from the **Representations** node into the scene.



- Select the **Displayers** > **Scalar 1** node in the scene and set the *Representation* property in the *Properties* window to the representation of the recorded solution view.

Note: Selecting the representation in the *Properties* window only applies the recorded solution to the selected displayer. Using one of the drag and drop methods applies the recorded solution to all displayers and annotations in the scene.

Visualizing Recorded Solution Data

You can change the time of the displayed solution in the properties of the solution view node.

When you first apply a recorded solution view to a scene, the scene displays the data for the first time-step that is stored in the simulation history file.

To change the solution time, select the solution view node in the **Solution Views** manager node, and adjust one of the time properties such as *Solution Time*.

The scene updates to display the relevant solution data.

To display the data that is contained in a snapshot, select the desired snapshot from drop-down list next to the *State Name* property. You can also select specific states from this property.

More information about each property is provided below:

Solution History	Selects which simulation history file to read the saved data from.
State Name	Displays the name of the currently loaded state. Click to the right of this property to bring up a drop-down menu from which to select states.

State Index	Displays the index of the currently loaded state. Click to the right of this property to reveal a slider where the state can be changed by moving the slider.
Iteration	Displays the iteration number of the currently loaded state. Click to the right of this property to reveal a slider for changing the iteration number.
Time Step	Displays the currently loaded time step. Click to the right of this property to reveal a slider for changing the time step.
Solution Time	Displays the currently loaded time. Click to the right of this property to reveal a slider for changing the solution time.

See Also:

Applying Recorded Solution Views to Scenes

Reporting and Monitoring Recorded Solution Data

You can incorporate the data from a solution history into a report and monitor, letting you animate the monitor plot and view how it changes over time.

Note: This feature applies to report monitors, not field function monitors.

To monitor the solution history:

1. Load a solution history (`.simh`) file into the simulation.
2. Select the report that you wish to use with the solution history.
3. In the *Field Function* property of the report, select a field function that is contained in the simulation history.
4. In the *Representation* property of the report, select the name of the solution history representation.
5. Right-click the report and select **Create Monitor and Plot from Report**.
6. Expand the **Solution Views** node. Right-click the node of the solution view node and select **Animation** > **Play**.
7. To view the monitor plot at a particular point in time, select **Animation** > **Stop**.

To monitor multiple field functions from a solution history:

1. Using the technique described above, prepare multiple reports, each with the appropriate field function and solution view representation selected.
2. Multi-select the reports and select **Create Monitor and Plot from Report**.
3. In the dialog that appears, select **Single Plot**.

The monitor plot displays data for the multiple solution history field functions that you specified.

Using Recorded Solution Data with Reports

You can use recorded solution data with reports. This ability is useful for looking at engineering quantities at a particular instance in time.

To use the recorded solution data in a report:

1. In the *Representation* expert property of the report, select the appropriate recorded solution view representation from the drop-down list.

   Note: You can only use data that is stored in the simulation history file with reports. To include all auxiliary field functions in the simulation history file, activate the **Export for reporting** option before you record. A list of common reports and their required field functions is provided at the end of this section.

2. Run the report to display the data.

The recorded solution data is controlled using the solution view Properties window—run the report after changing the solution time to display the corresponding data in the *Output* window.

Required Field Functions for Common Report Types

A list of common reports and required field functions is provided below.

Report Type	Required Field Functions
Single Stream Heat Exchanger/ Heat Exchanger	Face Flux Pressure Temperature
Thrust	Pressure Velocity Face Flux Density
Mass Average	Density Cell Relative Velocity
Mass Flow Average	Density Cell Relative Velocity
Mass Flow	Face Flux Density Cell Relative Velocity
Momentum/ Momentum Coefficient	Pressure Wall Shear Stress
Force	Pressure Wall Shear Stress

Note: For any other type of report, make sure that you record the appropriate field functions to the simulation history file.

See Also:

What Is a Report?

Visualizing Recorded Solution Data

Using Recorded Solution Data with Plots

You can use recorded solution data with XY Plots. This feature allows you to plot two-dimensional profiles of solution data.

To use recorded solution data in a plot:

- In the *Representation* property of the XY plot, select the appropriate recorded solution view representation from the drop-down list.

Note that only data that is stored in the simulation history file can be used with plots. The recorded solution data is controlled using the solution view Properties window. Changing the solution time, for example, causes the plot to update automatically and display the data for that moment in time.

See Also:

What Is an XY Plot?

Using Solution Snapshots

A snapshot is essentially a copy of the selected solution data (scalars/vectors for that moment in time) in a simulation file. Snapshots can be used to compare multiple steady state solutions in a single simulation file, such as different design iterations.

To create a snapshot:

1. Right-click on the simulation history file in the object tree, and select **Create Snapshot**.
2. In the *Create Snapshot* dialog, give the snapshot a suitable name.
3. Click **OK** to close the dialog and add the solution data to the simulation history file.

A suggested procedure for comparing steady state simulations is as follows:

1. Create a simulation history file.
2. In the *Properties* window, disable the Auto-Record property so that no data is saved at each iteration.
3. Continue by selecting the scalars/vectors you want to save.
4. Set up a case and run it to convergence.
5. Create a snapshot of the solution, as described above, giving it an appropriate name.
6. Clear the solution.
7. Set up the case with the new settings you want to test.
8. Run the solver to convergence.
9. Create a snapshot of the new results.
10. Repeat steps 6 - 9 as many times as you want.
11. Create a new Recorded Solution View. This action provides access to all the saved snapshots.
12. Apply the recorded solution view to a scene and switch between the snapshots.

To perform a side-by-side comparison:

1. Create multiple recorded solution views, where each solution view displays the data from different snapshots.
2. Create multiple scenes.
3. Assign a different recorded solution view to each scene.
4. Arrange the scenes side by side.

See Also:

Recording Solution Data
Configuring the Simulation History File
Applying Recorded Solution Views to Scenes

Animating Solution Views

You can animate the transient and steady state solution data that is stored in a simulation history file.

For transient solutions, you can view the solution data against time. For steady state solutions, you can view a slideshow of the different snapshots. To control the animation of a recorded solution, access the **Animation** sub-node of that recorded solution view node.

Select a mode from the drop-down list of the *Animation Mode* property:

- **State Index** -- solution states as shown in the solution history
- **Iterations** -- solution steps for a steady simulation
- **Time Steps** -- solution steps for an unsteady simulation
- **Solution Time** -- elapsed time of the calculations

Your selection adds a sub-node with the name of the selected mode to the **Animation** sub-node of the solution view. For example if you select **State Index**, the new sub-node is named **State Index Animation**. Define the recorded solution view animation parameters in the properties of the sub-node.

Recorded solution view animations are controlled in the same way as standard scene animations. Similarly, you can record a copy of the animation to disk as you would with a normal scene.

For more details on how the parameters in the *Write animation* dialog affect the playback of the recorded movie, see Defining Playback Settings for Recorded Movies.

More details on what each property controls are provided below. Note that the properties that are shown below are for the **State Index Animation** node. The name of the *Start* and *End* properties vary depending on what *Animation Mode* you selected.

Cycle Time	Specifies the time that it takes to loop through all the state indexes.	
Auto-Range	Controls whether the values of the *Start Index* and *End Index* properties are automatically populated when playing the animation.	
	Activated	The values of *Start Index* and *End Index* are automatically filled in when playing the animation.
	Deactivated	The values of *Start Index* and *End Index* are not automatically filled in. These values must be entered manually.
Start Index	The state index for which the animation starts.	
End Index	The state index for which the animation ends.	

Animating Vectors and Sweeping Section Planes in Solution History Views

You can view animated vectors or sweeping section planes in the scene at any of the saved states.

The sweeping rate for plane sections is specified with a *Cycle Time*, like in the solution view animation. For both types of animation, the animation attribute of the scene specifies a target frame rate in frames per

second (fps). It is called the target frame rate because a scene can be too complicated to draw at the rate you specified.

Together, the *Cycle Time* and the *Target frame rate (fps)* determine how many frames of the animation are used to animate the effect and the total number of frames in the animation. For example, if the *Cycle Time* is `10` seconds and the *Target frame rate (fps)* is `25` fps, the total number of frames in one cycle is 250 frames (10 * 25).

Plane sweeps and vector animations interpolate the positions of their objects at each frame to build a smooth animation. Solution state changes do not interpolate between states that are stored in the simulation history file. Scenes can therefore change abruptly when switching from one state to the next.

Animating Streamline Tracers in Solution History Views

You can view animated streamlines in the scene at any of the saved states. If you have a streamline (or constrained streamline) derived part in a simulation, use the following procedure to view the streamlines in a solution history playback.

1. If you have a streamline (or constrained streamline) derived part in a simulation, use the following procedure to view the streamlines in a solution history playback.
   a. In the *Inputs* property, select the streamline derived part.
   b. To minimize the size of the `.simh` file, deselect the regions in the *Regions* property.
   c. Select the desired scalar or vector field functions in the *Functions* property.
2. Run the simulation for at least one iteration or time-step. This part of the process is required so that a solution state is saved in the solution history file.
3. Create the solution view. See Retrieving Results from the Simulation History Files.
4. Select the streamline displayer node of the desired scene, and set *Representation* to the solution view.
5. Select the **Animations** sub-node of the streamline displayer node, and set *Animation Mode* to **Tracers**.

   The **Animation** toolbar is associated with the streamline displayer.
6. View the solution history with the streamlines one of two ways:
   - Play the animation using the controls in the **Animation** toolbar.
   - Choose to view individual states of the solution history by selecting the solution view node and adjusting any of the following properties:
     - *State Name*, using the drop-down list
     - *State Index*, using the slider or a numeric entry
     - *Iteration* (for steady cases), using the slider or a numeric entry
     - *Time Step* (for transient cases), using the slider or a numeric entry

Controlling the Animation Playback

If the animation settings are such that the number of frames in the animation is less than the number of states between the *Start Index* and *End Index*, some states are skipped. On the other hand, if the animation settings are such that the number of frames in the animation is greater than the number of states between the *Start Index* and *End Index*, states are repeated on multiple frames of the animation.

When an applied solution view has the animation enabled, you can also use the **Animation** toolbar to animate through the solution states.

As with vector and plane sweep animations, the **Animation** toolbar can be used to pause an animation that is playing. When paused, the frame stepping buttons are enabled so that you can step through the animation forward or backward one frame at a time. The following behavior can be observed:

- In the case where the number of frames in the animation is less than the number of states between the *Start Index* and **End Index**, pressing the forward button skips over one or more states.
- In the case where the number of frames in the animation is greater than the number of states between the *Start Index* and *End Index*, you need to press the forward button several times to advance to the frame so that the next state appears. If there are vector animations or plane section sweep animations that are enabled in the scene, their animations change from one frame to the next. If there are no vector or plane section sweep animations, the frame stepping buttons appear to be doing nothing, especially when the same state is repeated over several frames.

Defining Playback Settings for Recorded Movies

The playback settings for recorded movies are controlled in the *Write animation* dialog. The settings that affect the recorded movie are:

- **Frame Rate** - this property controls the playback of the recorded movie. Higher frame rates take longer to write, and result in a larger file size for the movie.
- **Start Time** - this property sets the time at which the recorded solution data starts writing to the simulation history file. For example if the fluid interaction you are interested in starts after 5 s of recorded solution data, then it is recommended that you set the movie to start recording at 5 s.
- **Animation Length** - this property defines the length of the recorded movie. This parameter must match the recorded *Cycle Time* if you want the movie to play once, otherwise the animation loops, and plays the same content multiple times. For example if the *Cycle Time* is 20 s, and the *Animation Length* is set to 60 seconds, the animation plays the same content 3 times (60/20).

If you want the recorded movie to play back at the normal speed (not slowed down/speed up), the *Animation Length* must match the *Cycle Time*. Some examples are provided below:

Cycle Time (s)	Number of States	Animation Length (s)	Frame Rate (fps)	Playback Speed
10	250	10	25	x1
10	250	20	25	x1 (looped twice)
10	500	20	25	x0.5
10	500	10	25	x1
10	500	5	50	x2

See Also:

Animating Scenes

Controlling the Animation

Recording the Animation onto Separate Files

Sampling Data Values During and After Running

This workflow incorporates the roles of both reports and monitors.

Decide on the data values that you want to extract, and create reports according to those values:

- Specific types -- with some types, such as force, the scalar is inherently defined for use with certain physics models.
- Statistical -- you can apply quantitative definitions, such as minimum and maximum, to the scalar of your choice.
- System -- these reports allow you to assess performance-related information.
- An expression report is available that depends on a combination of reports other than system reports.

Prepare derived parts and any field functions that are necessary to compute the data.

To set up a report:

1. To create a report, right-click the **Reports** node and select **New Report**, followed by the type of report that you wish to create.

   STAR-CCM+ adds the corresponding child object. In addition, STAR-CCM+ adds a field function by which you can access the report value in other field functions or expressions. The report field function name begins with **Report:** . For details on working with a report-based field function, see Using Reports as Field Functions.
2. If necessary, rename the report to reflect its purpose within the simulation. The corresponding field function name updates to reflect the change. For example, `Mass Flow at Inlet`.
3. If your simulation uses parts-based interfaces, you can report on those interfaces. See Working with Parts-Based Interfaces in Reports.
4. To change the frame of reference of a report, switch the *Coordinate System* property from the default **Laboratory** setting to a coordinate system, if applicable. This option is available for only certain types of reports. For more details, see Using Coordinate Systems with Force and Moment Reports.
5. Set any other properties that the report requires:
   - For system reports, such as **Solver Iteration CPU Time**, you can select the units in which STAR-CCM+ reports the quantity.
   - For statistical reports, such as Maximum, Sum, or Surface Integral, you typically select the reporting surface, the scalar field, and the units. For details, see Statistical Reports.
   - Expression reports are designed to operate on the results of one or more other reports in the simulation. For details, see Expression Report.
6. To add flexibility in force and moment calculations, divide the object into bands. See Using Bands with Force and Moment Reports.
7. For a Minimum or Maximum report, you can add other field functions at the location of the primary field function of the report. See Reporting Additional Field Functions at Maximum or Minimum.

Reports can be monitored while the solution is iterating by creating a monitor. This capability is useful when you are interested in the history of the quantity as the solution evolves, or if you want to use this report as one of your stopping criteria. Report monitors can also be plotted. See What Is a Report-Based Monitor?.

To set up report-based monitors, and (optionally) plots:

8. To create a monitor from a report, right-click on the report node and choose **Create Monitor from Report**.

   The default name of the monitor will be based on the report type. However, renaming the monitor is possible.
9. To create a monitor plot simultaneously with a report-based monitor, right-click on the report node and choose **Create Monitor and Plot from Report**.
10. For a report-based monitor, to extract scalar values for a specific part, see Extracting Report Results on a Per-Part Basis.

Special types of monitors exist for sampling field function data.

To set up field function monitors:

11. To create a field function monitor, right-click the **Monitors** node and select the type you wish to create from the **New Monitor** pop-up menu.
12. Rename the field function monitor if you wish.

    If you are using anything other than alphanumeric characters to name a field function monitor, you need to use curly brackets (`{}`). For example, if a field function monitor happens to have the name `Mass Flow * 2`, its corresponding field function can be interpreted by STAR-CCM+ to multiply the mass flow by 2 in the calculations. Therefore the field function monitor would need to be renamed `{MassFlow} * 2`. These brackets are also necessary for names that include characters from non-Western alphabets.
13. Set properties of the field function monitor, including additional sampling features that are described in the following sections:
    - Using a Sliding Sample Window
    - Specifying When Data Collection Stops

Set up monitors generally:

14. Set any properties that the monitor requires:
    - Change the trigger event. See Changing the Trigger Event and Sampling Frequency.
    - If you are using a monitor plot, set the upper limit on the number of points that appear in that plot. See Setting Maximum Plot Samples.
    - To adjust normalization, see Normalizing the Monitor Data.

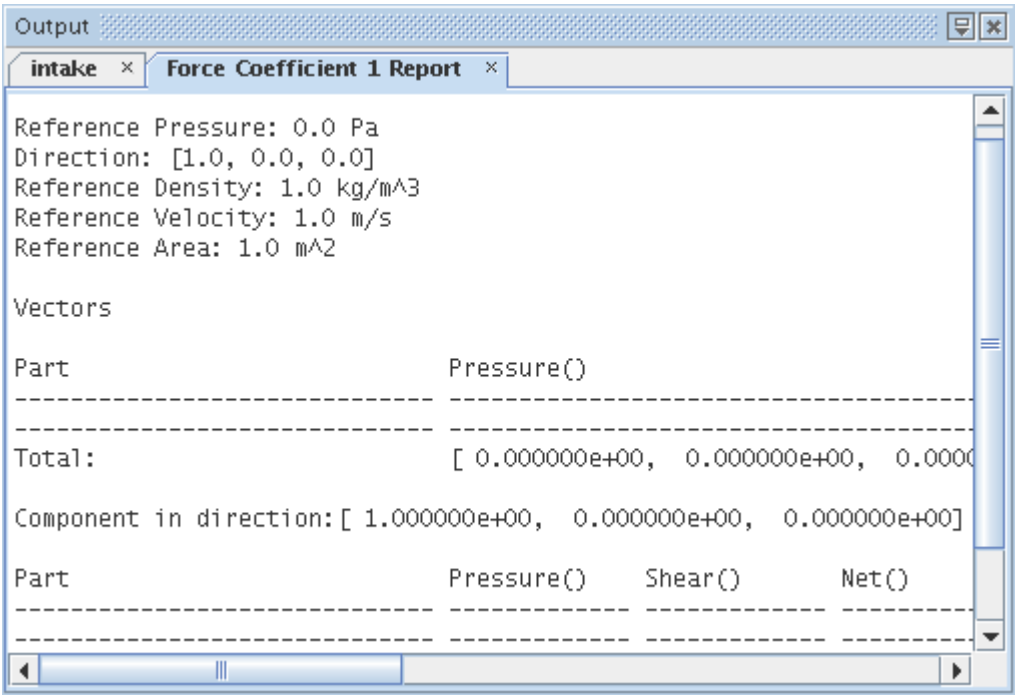
Work with the reports:

15. Run the simulation.
16. To generate the report results in the *Output* window, right-click the report node and selecting **Run Report**.

    STAR-CCM+ reports generate error messages when they lack sufficient data or when something else is wrong. See Understanding Report Errors.

    The results appear in the *Output* window sorted alphabetically, first by region, then by boundary. This order is the same as the order used in the parts selector for regions and boundaries.

With specific report types, the result of the report is printed to a new tab in the *Output* window, which you can work with like any *Output* tab.



If you have already run the report once before, then subsequent runs send the results to the same tab that was created with the first execution of the action.

17. To run all the reports in your simulation file with a single command, right-click the **Reports** node and select **Run all reports**.

    The results of all the reports appear in the *Output* window, in a separate tab for each report.

18. To display the sampled value on a scene, create a report annotation:

    a) Right-click the relevant report node and choose **Create Annotation from Report**.

       This action adds an annotation node with the same name to the **Tools** > **Annotations** node.

    b) Add the annotation to the scene. See Annotations Workflow.

19. To copy properties of reports:

    It is possible to copy properties among similar reports of all categories except for system reports, which do not have properties.

    Among reports of the same type, this operation causes all of the properties of the source report to be applied except for *Force Option*, which must be set separately for the destination report.

    It is not possible to copy properties between different types of reports.

20. To delete a report:

    Reports may be deleted using the standard methods, unless a monitor is using the report. When you attempt to delete a report in such a case, STAR-CCM+ opens a warning dialog about the dependency. For details, see Object Dependencies.

Work with the monitors:

21. Copy a field function monitor within a simulation or between simulations.

    When a field function monitor is copied within a simulation, its properties are copied. If you copy this object between simulation files, some or all of the properties may get copied, depending on the model selections of the destination simulation.

22. To delete a report-based monitor or field function monitor:

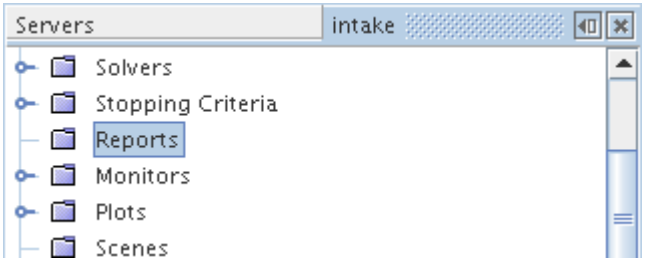
    Report-based monitors can only be deleted if they are not used by a monitor plot.

    Field function monitors can only be deleted if they are not displayed in an XY plot or used to define a stopping criterion.

Reporting Results

Reports provide summaries of solution data.

The **Reports** manager node, which has its own pop-up menu, contains all of the report nodes in the simulation tree.



A reference section on report types is also provided.

Contents:

What Is a Report?
System Reports
Statistical Reports
Specific Reports
Expression Report
Understanding Report Errors
Using Reports as Field Functions
Reporting Additional Field Functions at Maximum or Minimum
Using Coordinate Systems with Force and Moment Reports
Using Bands with Force and Moment Reports
Working with Parts-Based Interfaces in Reports
Reports Bibliography

What Is a Report?

A report presents a computed summary of the current simulation or CPU data. A node represents each report in the simulation tree, each with its own pop-up menu. Report properties, however, vary by report type.

The report summaries are useful for post-processing, and enable engineering quantities such as drag, lift, torque, or mass flow to be computed. They are also useful for computing diagnostic quantities such as minimum cell volume or average wall y^+.

There are three basic categories of report types:

- Specific types -- with some types, such as force, the scalar is inherently defined for use with certain physics models. For other types, such as the heat exchanger, there is a limited set of choices of scalars that pertain to the related physics model.
- Statistical -- you can apply quantitative definitions, such as minimum and maximum, to the scalar of your choice.
- System -- these reports allow you to assess performance related information. For example, the Solver Iteration CPU Time report indicates the total time taken for the last iteration by all the CPUs that you are using for the simulation.

Additionally, an expression report is available that depends on a combination of reports other than system reports.

The first two categories of report require a selection of parts over which the report is calculated, and they are available on:

- Regions only
- Surfaces only
- Boundary surfaces only
- Derived parts only
- Lagrangian parcels only
- A combination of the above

The reports that are made available to a simulation depend on the models that are activated. However, six reports (solver CPU time, solver elapsed time, maximum, minimum, memory, and sum) are always available.

Contents:

Report Pop-Up Menu
Reports Manager Pop-Up Menu

Report Pop-Up Menu

These pop-up menu items are available when you right-click a report node.

Run Report	Runs the report once, generating the results in the *Output* window.

Highlight	Highlights the input parts of the report in the active scene. This option is only available if: • The report has a *Parts* property. • The **Highlight Parts in Scene** option is activated (the default setting). See Highlighting Objects.
Create Monitor and Plot from Report	Creates a monitor for this report and then also creates a plot from that monitor. The new plot takes the name of this report suffixed with " Plot". The plot is not automatically displayed on creation, but the new report node is selected in the tree.
Create Monitor from Report	Creates a monitor for this report. The monitor takes the name of the report suffixed with " Monitor" and its node is automatically selected in the tree.
Create Annotation from Report	Creates an annotation for this report. The annotation takes the name of the report and its node is automatically selected in the tree.
Dependencies	Generates a *Dependencies* tab in the *Explorer* window to show the objects that require the presence of the selected object. See Checking an Object for Dependencies.
Error Log	Generates detailed information about the errors of a report. See Understanding Report Errors.

Reports Manager Pop-Up Menu

Several actions are available when you right-click the **Reports** manager node.

New Report	Creates a report of the type you select in the submenu. Some reports are only available when you choose certain models in the physics continuum.
Run all reports	Runs all reports at once, generating the results in separate tabs in the *Output* window.
New Group	Facilitates the grouping of objects by creating a group node within the manager node.
Group By	Allows you to group the objects according to any of the properties of the objects in the manager node, listed in a submenu of this menu item.
Ungroup	Reverses the grouping process.

System Reports

System reports provide information related to the running of the simulation.

No scalar is selected, and no input parts need be defined. When one of these reports is added to the report manager node, it is removed from the pop-up menu of that node, as you only require one instance of each system report.

System Report Properties

Units	Units that are used for specifying the quantity.

Contents:

Memory Report
PISO Correctors Report
Solver Iteration CPU Time
Solver Iteration Elapsed Time
Solver CPU Time Per Time-Step
Solver Elapsed Time Per Time-Step
Total Solver CPU Time
Total Solver Elapsed Time

Memory Report

The Memory Report returns the memory usage of STAR-CCM+.

Under the Memory Report there are options to return different memory metrics:

- Virtual
- Resident
- Virtual High Water Mark
- Resident High Water Mark

The high-water-mark (HWM) values essentially return the peak memory usage during the life of the simulation. The HWM values should never decrease during the simulation. In STAR-CCM+ the memory peaks of different parallel simulation processes should be reached at similar times and the accumulated HWM values should be a good estimate of the global high-water-mark.

The other metrics return the peak during each iteration.

The peak-per-iteration values incur a small performance penalty due to the necessary sampling of the memory metric from the system (in a manner similar to well-known system tools).

The HWM metrics do not require constant sampling and do not impose any additional performance penalty.

On Linux there are some important well known limitations of the memory metrics available. The memory metrics extracted for each process include resources that are shared between processes. When accumulating

per-process metrics, shared resources are accounted multiple times and thus the accumulated values overestimate the true memory usage. This is particularly acute when measuring Virtual memory use but even the Resident values are overestimated.

Memory Report Properties

Units	Units that are used for specifying the quantity.	
Metric Type	Allows you to select a type of memory metric.	
	Virtual	The maximum virtual memory that is accumulated across all processes per iteration.
	Resident	The maximum physical memory (RAM) that is accumulated across all processes per iteration.
	Virtual High Water Mark	The accumulated virtual high water mark across all processes.
	Resident High Water Mark	The accumulated resident high water mark across all processes.

Memory Report Expert Properties

Sampling Frequency	The frequency at which the system is probed for memory consumption. Applies to all memory reports. This setting is useful for extracting "maximum per iteration" values (Virtual, Resident). The frequency has been set to minimize the cost of the sampling while still returning accurate values. Increasing the sampling frequency will increase the granularity of the sampling and thus make it more accurate, but it will also increase the cost of the sampling. Conversely reducing the sampling frequency will reduce the accuracy and decrease the cost of the sampling. You are advised to modify this value only if you have expert understanding of the implications.

PISO Correctors Report

Reports the number of PISO corrector steps taken witih each time-step.

Available only with the PISO Unsteady model. When this model is selected, the **PISO Correctors Monitor** is added under Monitors and the **Maximum PISO Correctors** and **PISO Residual Reduction** criteria are added under Stopping Criteria.

The **Maximum PISO Correctors** criterion allows you to set *Maximum PISO Correctors* allowed in an iteration. The default is 20.

The **PISO Residual Reduction** criterion allows you to set *Minimum PISO Correctors* required per iteration (default of 2) and *PISO Residual Reduction* (default of 0.25).

Solver Iteration CPU Time

The solver CPU time is the accumulated CPU time for all processes for the last iteration executed.

Solver Iteration Elapsed Time

The solver elapsed time shows how long it took to execute the last iteration.

Solver CPU Time Per Time-Step

The solver CPU time is the accumulated CPU time for all processes, per time-step (for unsteady simulations), for the last time-step executed.

Solver Elapsed Time Per Time-Step

The solver elapsed time shows how long it took to execute the last time-step (for unsteady simulations).

Total Solver CPU Time

The total solver CPU time is the accumulated CPU time for all processes. Clearing the solution resets the total CPU time to zero.

Total Solver Elapsed Time

The total solver elapsed time is the cumulative time taken to run the entire solution. Clearing the solution resets the total solver elapsed time to zero.

Statistical Reports

Statistical reports apply statistical functions to scalar data sourced from chosen parts in the simulation.

In the part selector for a statistical report, STAR-CCM+ exposes only those parts that are relevant to the report. For example, a surface average report only accepts surfaces or lines, whereas the volume averaged report only accepts regions.

Note: When the chosen scalar quantity is not defined on a face, and yet the parts for the report include face parts, STAR-CCM+ provides the report with scalar values on the cells adjacent to the face parts. For example, you could have a Sum report that requires Volume for the Scalar Field Function, and the outer boundaries only of the domain for the input Parts. Instead of returning a zero value for the volume of the boundary faces, STAR-CCM+ returns the sum of the volumes of the cells adjacent to the boundary faces.

Statistical Report Properties

Parts	A list of parts that is used as the source for this report. Opens the in-place part selector for editing. The property customizer (...) button activates a modeless version of the selector dialog (it lets you interact with other portions of the workspace).
Scalar	Scalar field function value ϕ upon which the report is based. The particular formulation for the report can be found in the links above.
Units	Units that are used for specifying the quantity.

Statistical Report Expert Properties

Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.
Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Contents:

DarsCFD Reports
Element Count
Frontal Area
Harmonic Mass Average
Harmonic Mass Flow Average
Line Integral
Mass Averaged
Mass Flow Averaged
Maximum
Minimum

Overset Surface Integral
Overset Volume Integral
Particle Average
Sum
Surface Area Averaged
Surface Average
Surface Integral
Surface Standard Deviation
Surface Uniformity
Volume Average
Volume Integral
Volume Standard Deviation
Volume Uniformity

DarsCFD Reports

The DarsCFD Reports have their own properties and use a common report pop-up menu.

There are three types of DarsCFD Report:

- DarsCFD Quantity Maximum
- DarsCFD Quantity Minimum
- DarsCFD Quantity Sum

These reports take a particular value and perform a global operation (maximum, minimum or sum) over all of the processors. For the serial run, all three types of reports report the same value. Several of the DarsCFD Variable options only produce reports with values when the ISAT model is available. If the ISAT model is not available, these reports have a zero value.

DarsCFD Reports Properties

DarsCFD Variable	Choice of *DarsCFD Variables* available to plot.	
	None	This is the default option.
	Cpu Time	The time that the CPU takes with complex chemistry calculations.
	Elapsed Time	The time that has elapsed during complex chemistry calculations.
	Adds	The number of points that have been added to the table. (ISAT only).
	Grows	The number of points in the table that have increased their region of accuracy. (ISAT only).

	Retrieves	The number of points that are retrieved from the ISAT table; these retrieves avoid expensive direct ODE solver calculations (ISAT only).
	Direct Evaluations	The number of points that are calculated by calling the ODE solver. (ISAT only).
	Table Size	The ISAT table size.
	Queries	The total number of queries given to ISAT.
	Direct ODE Queries	The number of queries that result in direct ODE solver calculations. Low values in this report reflect a good setup of ISAT parameters.
Variable Accumulation Type	Format of displayed value for those *DarsCFD Variables* that report a value of time.	
	Cumulative	The time taken for the complex chemistry solver calculations in all iterations. This is the default option.
	Per Iteration	The time taken for the complex chemistry solver calculation for each iteration
Units	Units that are used for specifying the quantity.	

When you select the properties you require, the name of the report node in the simulation tree updates to reflect your chosen properties.

Element Count

The element count report has its own properties and uses a common report pop-up menu.

It counts the number of elements (cells, faces, parcels, tracks) in the data source.

Element Count Report Properties

Parts	A list of parts that is used as the source for this report. Opens the in-place part selector for editing. The ... (**property customizer**) button activates a modeless version of the selector dialog (it lets you interact with other portions of the workspace).
Units	Units that are used for specifying the quantity.

Frontal Area

The frontal area report has its own properties and uses a common report pop-up menu.

It calculates the combined projected area of all parts in the report onto a user-specified plane. A rasterization technique is used to first generate an image in memory, then sum pixels resulting from the rasterization, and finally scale the result by the characteristic pixel area. You can link this report to a local coordinate system and thus simplify your direction calculation by changing the frame of reference. This can be useful if, for example, you are modeling a certain part that is oriented differently from standard (laboratory) coordinates.

Note: A region does not have a discretization for all representations, so depending on which representation you choose, the region may or may not contribute to the computed area. For consistent results across all representations, it is recommended that you select parts other than a region for this report.

To use a local coordinate system in a report, first add it to the simulation. Then select the node of the report and choose that coordinate system from the drop-down list of the *Coordinate System* property.

Frontal Area Report Properties

Coordinate System	Specifies the coordinate system in which the *Normal* and *View Up* vectors are defined.
Normal	Specifies the normal vector for the plane from which the projection is calculated.
View Up	Used to define the plane that the geometry is projected onto. The cross product of the **Normal** and the **View Up** vectors defines the third leg of an orthonormal basis that is used to define the transformation matrix to set up the geometry. The projection plane is defined by the **View Up** vector and the **Normal** x **View Up** vector. You are advised to set the **View Up** vector normal to the **Normal**vector.
Parts	A list of parts that is used as the source for this report. Opens the in-place part selector for editing. The ... (**property customizer**) button activates a modeless version of the selector dialog (it lets you interact with other portions of the workspace).
Units	Units that are used for specifying the quantity.

Harmonic Mass Average

The mass average scalar quantity is computed over a volume as:

$$Mass\ Average = \frac{\hat{q}_{\phi 0}}{\hat{q}_0} \tag{1209}$$

where:

$$\hat{q}_{\phi 0} = \left[\mathbf{C} \cdot \mathbf{q}_{\phi}^{*}\right]_{0th\ harmonic} \tag{1210}$$

$$\widehat{q}_0 = [\mathbf{C} \cdot \mathbf{q}^*]_{0th\ harmonic} \tag{1211}$$

$$\mathbf{q}^*_\phi = \sum_C \rho^*_C \phi^*_C V_C \tag{1212}$$

$$\mathbf{q}^* = \sum_C \rho^*_C V_C \tag{1213}$$

ρ^*_C is the time-level cell density; ϕ^*_C is the time-level cell scalar quantity, where the scalar could be one of the quantities listed under the *Scalar Option* property; V_c is the cell volume; $\mathbf{C}$ is the complex Fourier operator that converts time-level solutions into Fourier coefficients; and the subscript “0th harmonic” denotes the time-mean or the 0th harmonic of the Fourier coefficients.

The harmonic mass average report has its own properties and uses a common report pop-up menu.

Harmonic Mass Average Report Properties

Scalar Option	Selects one of the following values: • `Pressure` • `U_Velocity` • `V_Velocity` • `W_Velocity` • `Density` • `Temperature` • `Total Enthalpy` • `Total Pressure` • `Total Temperature` • `Entropy`
Parts	A list of parts that is used as the source for this report. Opens the in-place part selector for editing.
Units	Units that are used for specifying the quantity.

Harmonic Mass Average Report Expert Properties

Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.
Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Harmonic Mass Flow Average

The mass flow average scalar quantity through a surface is computed as:

$$Mass\ Flow\ Average = \frac{\hat{q}_{\phi 0}}{\hat{q}_0} \tag{1214}$$

where:

$$\hat{q}_{\phi 0} = \left[\mathbf{C} \cdot \mathbf{q}_{\phi}^{*}\right]_{0th\ harmonic} \tag{1215}$$

$$\hat{q}_0 = \left[\mathbf{C} \cdot \mathbf{q}^{*}\right]_{0th\ harmonic} \tag{1216}$$

$$\mathbf{q}_{\phi}^{*} = \sum_f \rho_C^* \phi_C^* \left| \mathbf{v}_f^* \cdot \mathbf{a}_f \right| \tag{1217}$$

$$\mathbf{q}^{*} = \sum_f \rho_f^* \left| \mathbf{v}_f^* \cdot \mathbf{a}_f \right| \tag{1218}$$

ρ_f^* is the time-level face density; ϕ_f^* is the time-level face scalar quantity, where the scalar could be one of the quantities listed under the *Scalar Option* property; $\mathbf{v}_f^*$ is the time-level face velocity vector; $\mathbf{a}_f$ is the face area vector; $\mathbf{C}$ is the complex Fourier operator that converts time-level solutions into Fourier coefficients; and the subscript “0th harmonic” denotes the time-mean or the 0th harmonic of the Fourier coefficients.

The harmonic mass flow average report has its own properties and uses a common report pop-up menu.

Harmonic Mass Flow Average Report Properties

Scalar Option	Selects one of the following values: • `Pressure` • `U_Velocity` • `V_Velocity` • `W_Velocity` • `Density` • `Temperature` • `Total Enthalpy` • `Total Pressure` • `Total Temperature` • `Entropy`
Parts	A list of parts that is used as the source for this report. Opens the in-place part selector for editing.
Units	Units that are used for specifying the quantity.

Harmonic Mass Flow Average Report Expert Properties

Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.

Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Line Integral

The line integral report is represented in the simulation tree by a node. Right-click this node to access the pop-up menu of the line integral report.

As a statistical report, the line integral largely has the same properties as the others of this type. However its *Field Function* property selects a scalar or vector field function value upon which the report is based.

A line integral refers to an integral of a function, whether it is a scalar or a vector field, along a curve. One example of using a line integral is calculating the amount of work done on a particle by a force field as it moves along an arbitrary curve C in that vector field.

To compute a line integral, a parameterization of the curve C is used, that is, C is expressed by a vector function $\mathbf{r}(t)$, with $a \leq t \leq b$. Here, $\mathbf{r}(a)$ and $\mathbf{r}(b)$ are the initial and end point of the curve, respectively.

The line integral of a scalar field f along a parametric curve C is:

$$\int_C f \; ds = \int_a^b f(\mathbf{r}(t))|\mathbf{r}'(t)| \, dt \approx \sum_i f\left(\mathbf{r}\left(t_i\right)\right)\Delta s_i \tag{1219}$$

where Δs_i is the distance between points on the curve.

The line integral of a vector field $\mathbf{F}(\mathbf{r})$is:

$$\int_C \mathbf{F}(\mathbf{r}) \cdot d\mathbf{r} = \int_a^b \mathbf{F}(\mathbf{r}(t)) \cdot \mathbf{r}'(t) \, dt \;\approx\; \sum_i \mathbf{F}\left(\mathbf{r}\left(t_i\right)\right) \cdot \Delta \mathbf{s}_i \tag{1220}$$

where $\Delta \mathbf{s}_i$ is the distance vector between points on the curve.

Reversing the orientation of curve C does not change the value of the line integral of a scalar field but reverses the sign of the line integral of a vector field.

The following limitations exist when you work with vector quantities:

- The start and end points for the integration cannot be user-specified.
- The display does not indicate the direction of integration of the line.

Mass Averaged

The mass averaged scalar quantity is computed over a volume as:

$$Mass\ Average \;\equiv\; \frac{\int_V \rho\, \phi \, dV}{\int_V \rho \; dV} \;=\; \frac{\sum_c \rho_c \phi_c V_c}{\sum_c \rho_c V_c} \tag{1221}$$

where ϕ_c is the cell value of the selected scalar, ρ_c is the cell density and V_c is the cell volume.

The mass average over parcels in a Lagrangian phase is computed as:

$$Mass\ Average \equiv \frac{\int_V \rho\ \phi\ dV}{\int_V \rho\ dV} = \frac{\sum_\pi \phi_p m_p n_\pi}{\sum_\pi m_p n_\pi} \tag{1222}$$

where m_p is the mass of a single particle represented by a parcel and $m_p n_\pi$ is the mass of the parcel.

See also the properties of this statistical report.

Mass Flow Averaged

The mass flow averaged scalar quantity through a surface is computed as:

$$Mass\ Flow\ Average \equiv \frac{\oint \rho\ \phi\ |\mathbf{v} \cdot\ d\mathbf{a}|}{\oint \rho\ |\mathbf{v} \cdot\ d\mathbf{a}|} = \frac{\sum_f \rho_f \phi_f \left|\mathbf{v}_f \cdot \mathbf{a}_f\right|}{\sum_f \rho_f \left|\mathbf{v}_f \cdot \mathbf{a}_f\right|} \tag{1223}$$

where ϕ_f is the face value of the selected scalar, ρ_f is the face density, $\mathbf{v}_f$ is the face velocity vector, and $\mathbf{a}_f$ is the face area vector.

See also the properties of this statistical report.

Maximum

The maximum value report returns the maximum cell value in the selected regions, the maximum face value in the selected boundaries, or the maximum value of ϕ for each parcel in a Lagrangian phase.

Maximum Report Properties

Collocated Field Functions	Adds other field functions at the location of the primary field function of this report. See Reporting Additional Field Functions at Maximum or Minimum.

See also the properties of this statistical report.

Minimum

The minimum value report returns the minimum cell value in the selected regions, the minimum face value in the selected boundaries, or the minimum value of ϕ for each parcel in a Lagrangian phase.

Minimum Report Properties

Collocated Field Functions	Adds other field functions at the location of the primary field function of this report. See Reporting Additional Field Functions at Maximum or Minimum.

See also the properties of this statistical report.

Overset Surface Integral

This report is particularly useful for overset mesh simulations. It computes the surface integral of a scalar quantity by taking into account the area of the overlapping cells of overlapping regions only once.

The surface integral of a scalar quantity is computed as:

$$\text{Scalar Surface Integral} \equiv \int \phi \, da \; = \sum_f \phi_f A_f \tag{1224}$$

where ϕ_f is the face value of the selected scalar and A_f is the corrected face area magnitude. The corrected cell face area is obtained by evaluating the **Overset Grid Area** field function. See Overset Mesh Field Function Reference

For axisymmetric cases, the mesh is assumed to be swept through an angle of 1 radian. For applications that use a mass flow inlet, the mass flow is therefore given in kg/rad s. Any volumetric or area quantities reported for the Axisymmetric model are assumed to be for a sector of 1 radian.

For two-dimensional models, the surface is assumed to have a unit depth (in SI units). Therefore any volumetric or area quantities that are reported for a two-dimensional model are assumed to be “per meter”.

The surface integral of a vector quantity is computed as:

$$\text{Vector Surface Integral} = \int_S (\mathbf{F} \cdot \hat{\mathbf{n}}) \, da \tag{1225}$$

where $\hat{\mathbf{n}}$ is a unit normal vector and $\mathbf{F}$ is the vector field. Similarly to the scalar surface integral calculation, the quantity in the integrand is computed for each face and the result is the sum over all faces. This calculation is supported for surfaces in both 2D and 3D models. Vector surface integrals are not supported for surface intersections with shell regions (e.g., thin films). Vector surface integrals cannot be computed for degenerate geometries (geometries for which the surface normal is not well-defined), such as plane-boundary intersections or plane-plane intersections within a region.

Overset Volume Integral

This report is particularly useful for overset mesh simulations. It computes the volume integral of a scalar quantity by taking into account the volume of the overlapping cells of overlapping regions only once.

The volume integral of a scalar quantity is computed as:

$$\text{Overset Volume Integral} \equiv \int_V \phi \; dV \; = \sum_c \phi_c \, V_c \tag{1226}$$

where ϕ_c is the cell value of the selected scalar and V_c is the corrected cell volume. The corrected cell volume is obtained by evaluating the **Overset Grid Volume** field function. See Overset Mesh Field Function Reference.

Particle Average

This report type is available only when the Lagrangian Multiphase model is activated.

The particle average of ϕ for reports that use Lagrangian phases is represented by:

$$\frac{\sum_{\pi} \phi_p n_\pi}{\sum_{\pi} n_\pi} \tag{1227}$$

where the sum is over all the parcels in the data source and n_π is the number of particles represented by a parcel, the “particle count”.

As a statistical report, the particle average has the same properties as the others of this type. However its *Parts* property only selects from among the Lagrangian phases in the simulation.

Sum

The sum report is computed over surfaces as:

$$Area \quad Sum \; = \sum_{f} \phi_f \tag{1228}$$

The sum report is computed over volumes as:

$$Volume \quad Sum \; = \sum_{c} \phi_c \tag{1229}$$

The sum report is computed over parcels in a Lagrangian phase as:

$$Parcel \quad Sum \; = \sum_{\pi} \phi_p \tag{1230}$$

See also the properties of this statistical report.

Surface Area Averaged

This report type is available only when the Lagrangian Multiphase model is activated.

The surface area average over parcels in a Lagrangian phase is computed as:

$$Area \quad Average \; \equiv \frac{1}{a} \int \phi \, da = \frac{\sum_{\pi} \phi_p A_s n_\pi}{\sum_{\pi} A_s n_\pi} \tag{1231}$$

where A_s is the surface area of a single particle that is represented by a parcel.

See also the properties of this statistical report.

Surface Average

The surface average of a scalar quantity ϕ is computed on a surface as:

$$\text{Surface Average} \equiv \frac{1}{a}\int \phi da = \frac{\sum_f \phi_f A_f}{\sum_f A_f} \tag{1232}$$

where ϕ_f is the face value of the selected scalar and A_f is the face area.

For axisymmetric cases, the mesh is assumed to be swept through an angle of 1 radian. For applications that use a mass flow inlet, the mass flow is therefore given in kg/rad s. Any volumetric or area quantities reported for the Axisymmetric model are assumed to be for a sector of 1 radian.

For two-dimensional cases, the mesh is assumed to have a unit depth (in SI units). Therefore any volumetric or area quantities that are reported for the two-dimensional model are assumed to be “per meter”.

See also the properties of this statistical report.

Surface Integral

The surface integral of a scalar quantity is computed as:

$$\text{Scalar Surface Integral} \equiv \int \phi\, da = \sum_f \phi_f A_f \tag{1233}$$

where ϕ_f is the face value of the selected scalar and A_f is the face area magnitude.

For axisymmetric cases, the mesh is assumed to be swept through an angle of 1 radian. For applications that use a mass flow inlet, the mass flow is therefore given in kg/rad s. Any volumetric or area quantities reported for the Axisymmetric model are assumed to be for a sector of 1 radian.

For two-dimensional models, the surface is assumed to have a unit depth (in SI units). Therefore any volumetric or area quantities that are reported for a two-dimensional model are assumed to be “per meter”.

The surface integral of a vector quantity is computed as:

$$\text{Vector Surface Integral} = \int_S (\mathbf{F} \cdot \hat{\mathbf{n}})\, da \tag{1234}$$

where $\hat{\mathbf{n}}$ is a unit normal vector and $\mathbf{F}$ is the vector field. Similarly to the scalar surface integral calculation, the quantity in the integrand is computed for each face and the result is the sum over all faces. This calculation is supported for surfaces in both 2D and 3D models. Vector surface integrals are not supported for surface intersections with shell regions (e.g., thin films). Vector surface integrals cannot be computed for degenerate geometries (geometries for which the surface normal is not well-defined), such as plane-boundary intersections or plane-plane intersections within a region.

See also the properties of this statistical report.

Surface Standard Deviation

The standard deviation of a scalar quantity is computed across a surface as:

$$Standard\ \ deviation\ \ of\ \ \phi\ =\ \sqrt{\frac{\sum_f \left(\phi_f - \overline{\phi}\right)^2 A_f}{\sum_f A_f}} \tag{1235}$$

where $\overline{\phi}$ is the surface average of ϕ, ϕ_f is the face value of the selected scalar and A_f is the face area.

Multiple Parts

When you select multiple parts for this report:

- The average value $\overline{\phi}$ is the collective mean for all the parts—not for each part individually.
- The *Output* window reports the standard deviation of ϕ for the surfaces of each of the component parts individually, as well as for all parts combined. In all cases, however, the reported values use the collective average of ϕ.

See also the properties of this statistical report.

Surface Uniformity

The uniformity of a scalar quantity is computed on a surface as:

$$Uniformity\ \ index\ \ of\ \ \phi\ =\ 1 - \frac{\sum_f \left|\phi_f - \overline{\phi}\right| A_f}{2\left|\overline{\phi}\right| \sum_f A_f} \tag{1236}$$

where $\overline{\phi}$ is the surface average of ϕ, ϕ_f is the face value of the selected scalar and A_f is the area of a face.

The uniformity index describes the distribution of a certain quantity on a surface. If the quantity is distributed equally, the resulting number is 1. This report is useful in applications where a uniform flow rate is desired across a whole area. Heat exchangers, catalysts, and filters are examples of such applications.

Multiple Parts

When you select multiple parts for this report:

- The average value $\overline{\phi}$ is the collective mean for all the parts—not for each part individually.
- The *Output* window reports the uniformity of ϕ for the surfaces of each of the component parts individually, as well as for all parts combined. In all cases, however, the reported values use the collective average of ϕ.

See also the properties of this statistical report.

Volume Average

The volume average of a scalar quantity is computed as:

$$Volume\ \ Average \equiv \frac{1}{V}\int \phi\ dV\ = \frac{\sum_c \phi_c V_c}{\sum_c V_c} \tag{1237}$$

where ϕ_c is the cell value of the selected scalar and V_c is the cell volume.

The volume average over parcels in a Lagrangian phase is computed as:

$$Volume\ \ Average \equiv \frac{1}{V}\int \phi\ dV\ = \frac{\sum_\pi \phi_p V_p n_\pi}{\sum_\pi V_p n_\pi} \tag{1238}$$

where V_p is the volume of a single particle represented by a parcel.

See also the properties of this statistical report.

Volume Integral

The volume integral of a scalar quantity is computed as:

$$Volume\ \ Integral \equiv \int \phi\ dV\ = \sum_c \phi_c\ V_c \tag{1239}$$

where ϕ_c is the cell value of the selected scalar and V_c is the cell volume.

See also the properties of this statistical report.

Volume Standard Deviation

The standard deviation of a scalar quantity is computed in a volume as:

$$Standard\ \ deviation\ \ of\ \ \phi\ = \sqrt{\frac{\sum_c (\phi_c - \overline{\phi})^2 V_c}{\sum_c V_c}} \tag{1240}$$

where $\overline{\phi}$ is the volume average of ϕ, ϕ_c is the value of the selected scalar in a cell and V_c is the cell volume.

Multiple Parts

When you select multiple parts for this report:

- The average value $\overline{\phi}$ is the collective mean for all the parts—not for each part individually.
- The *Output* window reports the standard deviation of ϕ for the volumes of each of the component parts individually, as well as for all parts combined. In all cases, however, the reported values use the collective average of ϕ.

See also the properties of this statistical report.

Volume Uniformity

The uniformity of a scalar quantity is computed in a volume as:

$$Uniformity\ index\ of\ \phi = 1 - \frac{\sum_c \left|\phi_c - \overline{\phi}\right| V_c}{2\left|\overline{\phi}\right| \sum_c V_c} \tag{1241}$$

where $\overline{\phi}$ is the volume average of ϕ, $\phi_c$is the value of the selected scalar in a cell and V_c is the cell volume.

This uniformity index describes the distribution of a certain quantity in a volume. If the quantity is distributed equally, the resulting number is 1. This report is useful in applications where a uniform flow rate is desired throughout an entire volume. Heat exchangers, catalysts, and filters are examples of such applications.

See also the properties of this statistical report.

Specific Reports

Some reports in STAR-CCM+ are only available in conjunction with specific physical models. For example, STAR-CCM+ can only compute the acceleration of a rigid body if the simulation contains one or more rigid bodies. The reports manager pop-up menu only exposes reports that STAR-CCM+ can compute for the current simulation.

6-DOF Report Shared Properties

Coordinate System	The coordinate system in which the direction $\mathbf{d}$ is specified.
Direction	A vector defining the direction $\mathbf{d}$ in which the acceleration is reported.
Body	The 6-DOF body for which the acceleration is reported.
Units	Units that are used for specifying the quantity.

Contents:

Force
Force (Harmonic)
Force Coefficient
Generalized Vibratory Force
Harmonic Mass Flow
Heat Exchanger
Heat Transfer
Isentropic Efficiency Report
Li-Ion Cell
Mass Flow
Phase Mass Flow
Moment
Moment (Harmonic)
Moment Coefficient
Overset Interpolation Quality
Penetration—Parcel and Vapor
Particle Mass Flow
Pressure Correction
Pressure Drop
Reaction Force
Reaction Moment
Rigid Body Acceleration
Rigid Body Angle
Rigid Body 1-DOF Angle

Rigid Body Angular Acceleration
Rigid Body Angular Momentum
Rigid Body Angular Velocity
Rigid Body Component Distance
6-DOF Constraint Deviation
Rigid Body Force
Rigid Body Moment
Rigid Body Rotational Energy
Rigid Body Spring Elongation
Rigid Body Catenary Length
Rigid Body Total Distance
Rigid Body Translation
Rigid Body Velocity
Sauter Mean Diameter
Temperature Correction
Thrust
Tumble/Swirl Ratio
Uniformity Deviation Report
Velocity Correction
Work Per Cycle
Coal Volatile Matter Yield
Char Burn Out

Force

The force report is represented in the simulation tree by a node. It has its own properties but is subject to the report node pop-up menu. It also has an expert property for obtaining values using bands.

The force on a surface is computed as:

$$f = \sum_{f} \left(\mathbf{f}_f^{pressure} + \mathbf{f}_f^{shear} \right) \cdot \mathbf{n}_f \tag{1242}$$

where:

- $\mathbf{f}_f^{pressure}$ and $\mathbf{f}_f^{shear}$ are the pressure and shear force vectors on the surface face.
- f, and $\mathbf{n}_f$ is a user-specified direction vector that indicates the direction in which to compute the force.

The pressure force vector on surface face f is computed as:

$$\mathbf{f}_f^{pressure} = \left(p_f - p_{ref} \right) \mathbf{a}_f \tag{1243}$$

where p_f is the face static pressure, $\mathbf{a}_f$ is the face area vector, and p_{ref} is the reference pressure. As defined, $\mathbf{f}_f^{pressure}$ is the force that the fluid exerts on the surface. The shear force on surface face f is computed as:

$$\mathbf{f}_f^{shear} = -\mathbf{T}_f \cdot \mathbf{a}_f \tag{1244}$$

where $\mathbf{T}_f$ is the stress tensor at face f. This shear force is exerted on the surface by the fluid.

Note: The report reference pressure p_{ref} is defined relative to the simulation reference pressure. This definition means that a p_{ref} of 0 Pa does not indicate absolute zero pressure, but is rather the value of the simulation reference pressure. Further, when the surface on which the force is computed is a closed surface, the choice of p_{ref} affects only the overall accuracy of the result: therefore set it to a value that is close to the average pressure on the surface. For an unclosed surface, p_{ref} is effectively the pressure that is applied on the back-side of the surface, and therefore directly affects the result.

Note: The force report calculates the force on the boundaries (or boundary threshold parts). The boundary of a Porous Baffle Interface consists of an open portion and a closed portion: the force report calculates the total force on both portions. Whereas pressure acts on both the open and closed portions, wall shear acts only on the closed portion. Therefore, take care when interpreting the results of force reports on these interfaces.

Force Report Properties

In addition to constants, you can enter field function syntax directly.

Units	Units that are used for specifying the quantity.	
Coordinate System	A list of coordinate systems in the simulation for use by this report.	
Direction	A Cartesian vector that indicates the direction in which the force is computed.	
Force Option	The forces (shear and pressure) that are considered in the computation of force.	
	Shear	Only the force due to shear, $\mathbf{f}_f^{shear}$ in Eqn. (1242) is considered.
	Pressure	Only the force due to pressure, $\mathbf{f}_f^{pressure}$ in Eqn. (1242) is considered.
	Pressure + Shear	Both pressure and shear forces, $\mathbf{f}_f^{pressure}$ and $\mathbf{f}_f^{shear}$ in Eqn. (1242) are considered.
Reference Pressure	The reference pressure for the computation of Eqn. (1243).	
Parts	A list of parts that are used as the source for this report. Opens the in-place part selector for editing. The ... (**property customizer**) button activates a modeless version of the selector dialog (it lets you interact with other portions of the workspace).	

Force Report Expert Properties

Number of Bands	A text entry for the number of bands into which to divide the model.
Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.
Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Force (Harmonic)

The force (harmonic) report is similar to the Force report. It has an expert property for obtaining values using bands.

This report computes the real or imaginary component of the Fourier coefficient of the surface pressure force for a user-specified mode n as:

$$\hat{f}_n = \frac{1}{N}\sum_{j=0}^{N-1} \mathbf{f}_j e^{-i\omega_n t_j} \tag{1245}$$

where:

- N is the number of time levels
- ω_n is the unsteady frequency at mode n
- t_j is the instantaneous time at time level j
- $\mathbf{f}_j$ is the surface force at time level j which is computed as:

$$\mathbf{f}_j = \sum_f \left(\mathbf{f}_{jf}^{pressure} + \mathbf{f}_{jf}^{shear}\right) \cdot \mathbf{n}_f \tag{1246}$$

where:

- $\mathbf{f}_{jf}^{pressure}$ and $\mathbf{f}_{jf}^{shear}$ are the pressure and shear force vectors at time level j and surface face f
- $\mathbf{n}_f$ is a user-specified direction vector that indicates the direction in which to compute the force

The pressure force vector $\mathbf{f}_{jf}^{pressure}$ at time level j and surface face f is computed as:

$$\mathbf{f}_{jf}^{pressure} = \left(p_{jf} - p_{ref}\right)\mathbf{a}_{jf} \tag{1247}$$

where:

- p_{jf} is the static pressure at time level j and surface face f
- p_{ref} is the reference pressure
- $\mathbf{a}_{jf}$ is the face area vector

The note on reference pressure that is provided for the Force report applies to this report.

The shear force on surface face f is computed as:

$$\mathbf{f}_{jf}^{shear} = -\mathbf{T}_{jf} \cdot \mathbf{a}_{jf} \tag{1248}$$

where $\mathbf{T}_{jf}$ is the stress tensor at time level j on surface face f. This shear force is exerted on the surface by the fluid.

Force (Harmonic) Report Properties

Units	Units that are used for specifying the quantity.	
Coordinate System	A list of coordinate systems in the simulation for use by this report.	
Direction	A Cartesian vector that indicates the direction in which the force is computed.	
Force Option	The forces (shear and pressure) that are considered in the computation of force.	
	Shear	Only the force due to shear, $\mathbf{f}_{jf}^{shear}$ in Eqn. (1246) is considered.
	Pressure	Only the force due to pressure, $\mathbf{f}_{jf}^{pressure}$ in Eqn. (1246) is considered.
	Pressure + Shear	Both pressure and shear forces, $\mathbf{f}_{jf}^{pressure}$ and $\mathbf{f}_{jf}^{shear}$ in Eqn. (1246) are considered.
Reference Pressure	The reference pressure for the computation of Eqn. (1247).	
Complex Option	Which component of the force is considered in the computation.	
	Real	Selects the real component.
	Imaginary	Selects the imaginary component.
	Magnitude	Includes the complete force in the computation rather than one of the components.
Mode	The user-specified mode n of the surface pressure force for which the Fourier coefficient is computed. See Eqn. (1245).	
Parts	A list of parts that is used as the source for this report. Opens the in-place part selector for editing.	

Force (Harmonic) Report Expert Properties

Number of Bands	A text entry for the number of bands into which to divide the model.
Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.

Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Force Coefficient

The force coefficient report calculates a dimensionless force in a user-specified direction, and can be used for lift and drag calculations.

It is represented in the simulation tree by a node, which has its own properties but is subject to the report node's pop-up menu. It also has an expert property for obtaining values using bands.

The force coefficient of a surface is defined as follows:

$$C_D = \frac{\sum_f \left(\mathbf{f}_f^{pressure} + \mathbf{f}_f^{shear}\right) \cdot \ \mathbf{n}_D}{\frac{1}{2}\left(\rho_{ref}\ v_{ref}^2\ a_{ref}\right)} \tag{1249}$$

where $\mathbf{f}_f^{pressure}$ and $\mathbf{f}_f^{shear}$ are the pressure and shear force vectors on the surface face f, and ρ_{ref}, v_{ref} and a_{ref} are the reference density, velocity, and area, respectively. $\mathbf{n}_D$ is a user-specified direction vector. To compute drag, for example, specify the direction vector as the direction of travel, and for lift, specify the direction as normal to the direction of travel.

The pressure force vector on surface face is computed as:

$$\mathbf{f}_f^{pressure} = \left(p_f - p_{ref}\right)\ \mathbf{a}_f \tag{1250}$$

where p_f is the face static pressure $\mathbf{a}_f$ is the face area vector, and p_{ref} is the reference pressure. As defined, this is the force that the fluid exerts on the surface. The note on reference pressure provided for the Force report applies here.

The shear force on surface face f is computed as:

$$\mathbf{f}_f^{sheaf} = -\mathbf{T}_f \cdot \mathbf{a}_f \tag{1251}$$

where $\mathbf{T}_f$ is the stress tensor at face f. This shear force is exerted on the surface by the fluid.

Force Coefficient Report Properties

In addition to constants, you can enter field function syntax directly.

Coordinate System	A list of coordinate systems in the simulation for use by this report.	
Direction	A Cartesian vector that indicates the direction in which the force is computed.	
Force Option	The type of force to be computed.	
	Shear	Only the force due to shear, $\mathbf{f}_f^{shear}$ in Eqn. (1249) is considered in the computation.

	Pressure	Only the force due to pressure, $\mathbf{f}_f^{pressure}$ in Eqn. (1249) is considered in the computation.
	Pressure & Shear	Both pressure and shear forces, $\mathbf{f}_f^{pressure}$ and $\mathbf{f}_f^{shear}$, are considered in the computation of Eqn. (1249).
Parts	A list of parts that is used as the source for this report. Opens the in-place part selector for editing. The ... (**property customizer**) button activates a modeless version of the selector dialog (it lets you interact with other portions of the workspace).	
Reference Area	The reference area for the computation of Eqn. (1249).	
Reference Density	The reference density for the computation of Eqn. (1249).	
Reference Pressure	The reference pressure for the computation of Eqn. (1250).	
Reference Velocity	The reference velocity for the computation of Eqn. (1249).	
Units	Units that are used for specifying the quantity.	

Force Coefficient Report Expert Properties

Number of Bands	A text entry for the number of bands into which to divide the model.
Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.
Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Generalized Vibratory Force

This report type is available only when Harmonic Balance Flutter motion is activated.

It has its own properties but is subject to the report node's pop-up menu.

This report computes the modal force of the user-specified flutter displacement mode on the structure. It is computed as:

$$F = \sum_f \mathbf{x}_f^1 \left(p_f^1 \mathbf{a}_f^0 + p_f^0 \mathbf{a}_f^1 \right) \tag{1252}$$

where $\mathbf{x}_f^1$ is the first Fourier mode of the displacement of face f, $\mathbf{a}_f^0$ and $\mathbf{a}_f^1$ are, respectively, the zeroth and first Fourier modes of the face areas, and p_f^0 and p_f^1 are, respectively, the zeroth and first Fourier modes of the pressure at the face. The generalized vibratory force F is a complex number and the report outputs the real and imaginary components of this force.

You can use the *Monitor Option* property of this report to monitor the real and imaginary parts of this quantity individually, as well as the magnitude of the complex number F.

Generalized Vibratory Force Report Properties

Monitor Option	Which component of the force is considered in the computation.	
	Real	Selects the real component.
	Imaginary	Selects the imaginary component.
	Magnitude	Includes the complete force in the computation rather than one of the components.
Parts	A list of parts that is used as the source for this report. Opens the in-place part selector for editing.	
Units	Units that are used for specifying the quantity.	

Generalized Vibratory Force Report Expert Properties

Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.
Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Harmonic Mass Flow

The total mass flow through a boundary is computed as:

$$\dot{m} = \int \widehat{\rho}_0 \widehat{\mathbf{v}}_0 \, d\mathbf{a} = \sum_f \widehat{\rho}_{0f} \widehat{\mathbf{v}}_{0f} \cdot d\mathbf{a}_f \tag{1253}$$

where $\widehat{\rho}_{0f}$ is the time-mean face density, $\widehat{\mathbf{v}}_{0f}$ is the time-mean face velocity vector, and $\mathbf{a}_f$ is the face area vector. The sign convention is such that a positive value of $\dot{m}$ implies that the mass flow is out of the domain.

The harmonic mass flow report has its own properties and is subject to the report node's pop-up menu.

Harmonic Mass Flow Report Properties

Units	Units that are used for specifying the quantity.
Parts	A list of parts that is used as the source for this report. Uses the in-place part selector for editing.

Harmonic Mass Flow Report Expert Properties

Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.
Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Heat Exchanger

There are two types of heat exchanger reports: single and dual stream:

- The single stream heat exchanger report uses calculations with one region. By this method, one stream is assumed to have a uniform temperature and the other stream is modeled by specifying the heat exchanger enthalpy source.
- The dual stream heat exchanger report uses calculations with an interface between two regions. This technique involves the heat exchange between a cold fluid stream and a hot fluid stream. These streams are modeled as two different physics continua (each having different material properties).

  For this report, the monitor values Mass Flux, Inlet Temperature, Outlet Temperature, and Pressure Drop may be selected for both the regions that comprise the dual stream heat exchanger.

  This report also gives efficiency for the two streams as follows:

$$eff_{cold} = \frac{\left(T_{cold\ -\ out} - T_{cold\ -\ in}\right)}{\left(T_{hot\ -\ in} - T_{cold\ -\ in}\right)} \tag{1254}$$

$$eff_{hot} = \frac{\left(T_{hot\ -\ in} - T_{hot\ -\ out}\right)}{\left(T_{hot\ -\ in} - T_{cold\ -\ in}\right)} \tag{1255}$$

Each of these reports is represented in the simulation tree by a node, which has its own properties but is subject to the report node's pop-up menu.

Heat Exchanger Report Properties

Monitor Units	Units in which the Monitor Value is monitored and plotted, but the preferred units are used in the report output for the heat transfer rate, pressures and temperatures.	
Monitor Value	Selects the quantity to be monitored.	
	Mass Flux	Selects mass flux $\dot{m}$.
	Heat Transfer	Selects heat transfer Q.
	Inlet Temperature	Selects inlet temperature T_{in}.
	Outlet Temperature	Selects outlet temperature T_{out}.
	Pressure Drop	Selects pressure drop Δp.
Heat Exchanger	Selects a region (single stream) or interface (dual stream) from a drop-down list of applicable objects in the simulation.	
Cold Downstream Boundaries	Downstream boundary interfaces of the cold stream (dual stream only)	
Hot Downstream Boundaries	Downstream boundary interfaces of the hot stream (dual stream only)	
Parts	A list of parts that is used as the source for this report. Opens the in-place part selector for editing. The ... (**property customizer**) button activates a modeless version of the selector dialog (it lets you interact with other portions of the workspace).	

Heat Exchanger Report Expert Properties

Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.
Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Heat Transfer

The heat transfer report is represented in the simulation tree by a node, which has its own properties but is subject to the report node's pop-up menu.

The heat transfer report gives the total heat transfer in Watts at a boundary. It is defined as:

$$q = \int \dot{\mathbf{q}}'' \cdot d\mathbf{a} = \sum_f \dot{\mathbf{q}}_f'' \cdot \mathbf{a}_f \tag{1256}$$

where $\dot{\mathbf{q}}_f''$ is the face heat flux vector and $\mathbf{a}_f$ is the face area vector. The sign convention is such that a positive value of q implies that heat transfer is from the fluid to the boundary.

It is possible to calculate the heat passing through a portion of a boundary by creating a boundary threshold derived part and running a heat transfer report on that part.

For harmonic balance cases, the reported heat transfer is a time-mean.

Heat Transfer Report Properties

Units	Units that are used for specifying the quantity.
Parts	A list of parts that is used as the source for this report. Uses the in-place part selector for editing.

Isentropic Efficiency Report

The isentropic efficiency report is available when an Equation of State model, such as Ideal Gas or Real Gas, is selected.

This report calculates the isentropic efficiency of a process between an inlet boundary (or boundaries) and outlet boundary (or boundaries). For expansion, this efficiency η is defined as the ratio of the actual change in enthalpy to the theoretical change in enthalpy, which is based on an isentropic process, measured between inlet and outlet:

$$\eta = \frac{h_i - h_o}{h_i - h_{oS}} \tag{1257}$$

where:

- h_i is the inlet enthalpy.
- h_o is the outlet enthalpy for the actual case.

- h_{oS} is the outlet enthalpy for a fully isentropic case.

For compression, the efficiency is the reciprocal, $\frac{h_i - h_{oS}}{h_i - h_o}$.

Isentropic Efficiency Report Properties

Units	None.	
Inlet	Specifies boundary surfaces for the inlet.	
Outlet	Specifies boundary surfaces for the outlet	
Process	Describes the overall process that is undergone by the fluid. Use the **Expansion** option for turbines and nozzles. Use the **Compression** option for compressors.	
Efficiency Option	The **Total-to-total** option calculates efficiency using total enthalpy at both inlet and outlet. The **Total-to-static** calculates efficiency using total enthalpy at the inlet and static enthalpy at the outlet.	

Isentropic Efficiency Report Expert Properties

Representation	Chooses the mesh representation from which the report takes its solution data.	
Smooth Values	When this property is **On**, the report calculates the value using values that are interpolated between nodes. When **Off**, the report calculates the value using cell values.	

See Also:

General Equation of State Models

Li-Ion Cell

The Li-ion cell report is represented in the simulation tree by a node which has properties and is subject to the report node's pop-up menu.

For more information about setting up Li-ion cell reports, refer to Reporting Li-Ion Cell Parameters.

Li-Ion Cell Report Properties

Monitor Value	Sets the value that the report monitors.	
	State of Charge	Monitors the state of charge SOC.
	Open Circuit Potential	Monitors the open circuit voltage OCV.
Voltage at 100% SOC	The voltage at which the battery is fully charged.	
Voltage at 0% SOC	The lower cut-off voltage.	
Positive Electrode Parts	A list of parts that is used as the source for positive electrodes for this report. Uses the in-place part selector for editing.	
Negative Electrode Parts	A list of parts that is used as the source for negative electrodes for this report. Uses the in-place part selector for editing.	
Units	The units of the monitored value. This property is updated automatically when the monitor value is changed.	

Li-Ion Cell Report Expert Properties

SOC Step Increment	The state of charge step increment in the state table.	
Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.	
Smooth Values	Whether to calculate the report using node (interpolated) values of the chosen quantity or cell values.	
	Activated	Use node (interpolated) values.
	Deactivated	Use cell values.

Mass Flow

The mass flow report is represented in the simulation tree by a node which has properties and is subject to the report node's pop-up menu.

The total mass flow through a boundary is computed as:

$$\dot{m} = \int \rho \mathbf{v} \cdot d\mathbf{a} = \sum_f \rho_f \mathbf{v}_f \cdot \mathbf{a}_f \tag{1258}$$

where ρ_f is the face density, $\mathbf{v}_f$ is the face velocity vector and $\mathbf{a}_f$ is the face area vector. The sign convention is such that a positive value of $\dot{m}$ implies that mass flow is out of the domain.

Mass Flow Report Properties

Units	Units that are used for specifying the quantity.
Parts	A list of parts that is used as the source for this report. Uses the in-place part selector for editing.

Mass Flow Report Expert Properties

Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.
Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Phase Mass Flow

The phase mass flow report is represented in the simulation tree by a node which has properties and is subject to the report node's pop-up menu.

The mass flow rate of a phase i through a boundary B is computed as:

$$\dot{m} = \int_B \alpha_i \rho_i \mathbf{v} \cdot d\mathbf{a} \approx \sum_f \alpha_{i,f} \rho_{i,f} \mathbf{v}_f \cdot \mathbf{a}_f \tag{1259}$$

where $\alpha_{i,f}$ is the volume fraction of phase i at boundary face f, $\rho_{i,f}$ is the density, $\mathbf{v}_f$ is the velocity vector, and $\mathbf{a}_f$ is the area vector. The sign convention is for a positive value of $\dot{m}$ to indicate mass flow out of the domain.

Phase Mass Flow Report Properties

Phase	The phase for which the mass report is run.
Units	Units that are used for specifying the quantity.
Parts	A list of parts that is used as the source for this report. Uses the in-place part selector for editing.

Phase Mass Flow Report Expert Properties

Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.

Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Moment

The moment report is represented in the simulation tree by a node, which has its own properties but is subject to the report node's pop-up menu. It also has an expert property for obtaining values using bands.

The moment of force on a surface about an axis is defined:

$$M = \sum_f \left[\mathbf{r}_f \times \left(\mathbf{f}_f^{pressure} + \mathbf{f}_f^{sheaf} \right) \right] \cdot \mathbf{a} \tag{1260}$$

where $\mathbf{f}_f^{pressure}$ and $\mathbf{f}_f^{shear}$ are the pressure and shear force vectors and $\mathbf{a}$ is a vector defining the axis through point $\mathbf{x}_0$ about which the moment is taken and $\mathbf{r}_f$ is the position of face f relative to $\mathbf{x}_0$.

The pressure force vector on surface face is computed as:

$$\mathbf{f}_f^{pressure} = \left(p_f - p_{ref} \right) \mathbf{a}_f \tag{1261}$$

where p_f is the face static pressure, $\mathbf{a}_f$ is the face area vector, and p_{ref} is the reference pressure. As defined, the fluid exerts this force on the surface. The note on reference pressure provided for the Force report applies here.

The shear force on surface face is computed as:

$$\mathbf{f}_f^{shear} = -\mathbf{T}_f \cdot \mathbf{a}_f \tag{1262}$$

where $\mathbf{T}_f$ is the stress tensor at face f. This shear force is exerted on the surface by the fluid.

Moment Report Properties

In addition to constants, you can enter field function syntax directly.

Units	Units that are used for specifying the quantity.	
Coordinate System	A list of coordinate systems in the simulation for use by this report.	
Axis	A Cartesian vector specifying the axis $\mathbf{a}$ of Eqn. (1260) about which the moment is computed.	
Force Option	Which of the forces (shear and pressure) are considered in the computation of the moment.	
	Shear	Only the force due to shear, $\mathbf{f}_f^{shear}$ in Eqn. (1260), is considered.
	Pressure	Only the force due to pressure, $\mathbf{f}_f^{pressure}$ in Eqn. (1260), is considered.

	Pressure + Shear	Both pressure and shear forces, $\mathbf{f}_f^{pressure}$ and $\mathbf{f}_f^{shear}$ in Eqn. (1260) are considered.
Reference Pressure	The reference pressure for the computation of Eqn. (1261).	
Axis Origin	A Cartesian position vector specifying the point $\mathbf{x}_0$ on the axis $\mathbf{a}$ about which the moment is taken.	
Parts	A list of parts that is used as the source for this report. Opens the in-place part selector for editing. The ... (**property customizer**) button activates a modeless version of the selector dialog (it lets you interact with other portions of the workspace).	

Moment Report Expert Properties

Number of Bands	A text entry for the number of bands into which to divide the model.
Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.
Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Moment (Harmonic)

For the chosen surface and mode, the moment (harmonic) report computes the moment of force about an axis.

The moment of force on a surface about an axis is defined as:

$$\widehat{M}_m = \frac{1}{N}\sum_{j=0}^{N-1} M_j e^{-i\omega_m t_j} \tag{1263}$$

where:

- m is the user-specified mode.
- M_j is the pressure force moment at time level j, where:

$$M_j = \sum_f \left[\mathbf{x}_{jf} - \mathbf{x}_0\right] \times \left[\mathbf{f}_{jf}^p + \mathbf{f}_{jf}^s\right]$$

  where:

  - $\mathbf{x}_{jf}$ is the centroid at time level j and cell face f.
  - $\mathbf{x}_0$ is a user-specified origin vector.
  - $\mathbf{f}_{jf}^p$ is the pressure force at time level j and cell face f.
  - $\mathbf{f}_{jf}^s$ is the shear force at time level j and cell face f.

- N is the number of time levels.
- ω_m is the unsteady frequency at mode m.

Moment Report Properties

In addition to constants, you can enter field function syntax directly.

Units	Units that are used for specifying the quantity.	
Coordinate System	A list of coordinate systems in the simulation for use by this report.	
Axis	A Cartesian vector specifying the axis $\mathbf{a}$ of Eqn. (1260) about which the moment is computed.	
Force Option	Which of the forces (shear and pressure) are considered in the computation of the moment.	
	Shear	Only the force due to shear, $\mathbf{f}_f^{shear}$ in Eqn. (1260), is considered.
	Pressure	Only the force due to pressure, $\mathbf{f}_f^{pressure}$ in Eqn. (1260), is considered.
	Pressure + Shear	Both pressure and shear forces, $\mathbf{f}_f^{pressure}$ and $\mathbf{f}_f^{shear}$ in Eqn. (1260) are considered.
Reference Pressure	The reference pressure for the computation of Eqn. (1261).	
Axis Origin	A Cartesian position vector specifying the point $\mathbf{x}_0$ on the axis $\mathbf{a}$ about which the moment is taken.	
Mode	The user-specified mode m of the moment of surface pressure force for which the Fourier coefficient is computed. See the equation above for $\widehat{M}_m$.	
Parts	A list of parts that is used as the source for this report. Opens the in-place part selector for editing. The ... (**property customizer**) button activates a modeless version of the selector dialog (it lets you interact with other portions of the workspace).	

Moment Report Expert Properties

Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.
Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Moment Coefficient

The moment coefficient report is represented in the simulation tree by a node, which has its own properties but is subject to the report node's pop-up menu. It also has an expert property for obtaining values using bands.

The moment coefficient of force on a surface about an axis is defined as:

$$C_M = \frac{\left(M = \sum_f \left[\mathbf{r}_f \times \left(\mathbf{f}_f^{\text{pressure}} + \mathbf{f}_f^{shear}\right)\right] \cdot \mathbf{a}\right)}{\left(\frac{1}{2}\rho_{\text{ref}} v_{\text{ref}}^2\right) a_{\text{ref}} r_{\text{ref}}} \tag{1264}$$

where $\mathbf{f}_f^{\text{pressure}}$ and $\mathbf{f}_f^{\text{shear}}$ are the pressure and shear force vectors and $\mathbf{a}$ is a vector defining the axis through point $\mathbf{x}_0$ about which the moment is taken and $\mathbf{r}_f$ is the position of face f relative to $\mathbf{x}_0$. ρ_{ref}, v_{ref}, a_{ref} and r_{ref} are the reference density, velocity, area, and radius, respectively.

The pressure force vector on surface face f is computed as:

$$f_f^{\text{pressure}} = \left(p_f - p_{\text{ref}}\right)\mathbf{a}_f \tag{1265}$$

where p_f is the face static pressure, $\mathbf{a}_f$ is the face area vector, and p_{ref} is the reference pressure. As defined, the fluid exerts this force on the surface. The note on reference pressure provided for the Force report applies here.

The shear force on surface face is computed as:

$$f_f^{\text{shear}} = -\mathbf{T}_f \cdot \mathbf{a}_f \tag{1266}$$

where $\mathbf{T}_f$ is the stress tensor at face f. This shear force is exerted on the surface by the fluid.

Moment Coefficient Report Properties

In addition to constants, you can enter field function syntax directly.

Axis	A Cartesian vector specifying the axis $\mathbf{a}$ of Eqn. (1264) about which the moment is computed.	
Axis Origin	A Cartesian position vector specifying the point $\mathbf{x}_0$ on the axis $\mathbf{a}$ about which the moment is taken.	
Coordinate System	A list of coordinate systems in the simulation for use by this report.	
Force Option	Which of the forces (shear and pressure) are considered in the computation of the moment.	
	Shear	Only the force due to shear, $\mathbf{f}_f^{\text{shear}}$ in Eqn. (1264), is considered.
	Pressure	Only the force due to pressure, $\mathbf{f}_f^{\text{pressure}}$ in Eqn. (1264), is considered.

	Pressure & Shear	Both pressure and shear forces, $\mathbf{f}_f^{\text{pressure}}$ and $\mathbf{f}_f^{\text{shear}}$ in Eqn. (1264) are considered.
Parts	A list of parts that are used as the source for this report. Opens the in-place part selector for editing. The ... (**property customizer**) button activates a modeless version of the selector dialog (it lets you interact with other portions of the workspace).	
Reference Area	The reference area for the computation of Eqn. (1264).	
Reference Density	The reference density for the computation of Eqn. (1264).	
Reference Pressure	The reference pressure for the computation of Eqn. (1265).	
Reference Radius	The reference radius for the computation of Eqn. (1264).	
Reference Velocity	The reference velocity for the computation of Eqn. (1264).	
Units	Units that are used for specifying the quantity.	

Moment Coefficient Report Expert Properties

Number of Bands	A text entry for the number of bands into which to divide the model.
Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.
Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Overset Interpolation Quality

This report is available only when the simulation contains an overset interface between regions. You use this report to monitor the quality of the interpolation between the overset mesh and the background mesh over time. With this information, you can detect during which time-steps low quality interpolation occurs.

The overset interpolation quality report returns the `Overset Interpolation Quality Measure for Overset Mesh`. This value represents the largest deviation from the optimum interpolation that occurs during one time-step. A value of 0 indicates no deviation from the best possible interpolation.

Overset Interpolation Quality Report Properties

Overset Interface

Selects the overset interface for which you wish to monitor overset interpolation quality.

Absolute Value

When activated, the overset interpolation quality measure is not normalized. When de-activated, the overset interpolation quality measure is normalized by the largest extent of the computational domain.

Penetration—Parcel and Vapor

The Parcel and Vapor Penetration reports return the distance interval that contains a given mass percentage of injected parcels or of vapor arising from injected droplets, measuring from the injection point. In both cases, if the given mass percentage is 90%, the report gives the point with 90% of the mass lying between it and the injection point.

The reports apply to the Solid Cone, Hollow Cone, and Nozzle injectors.

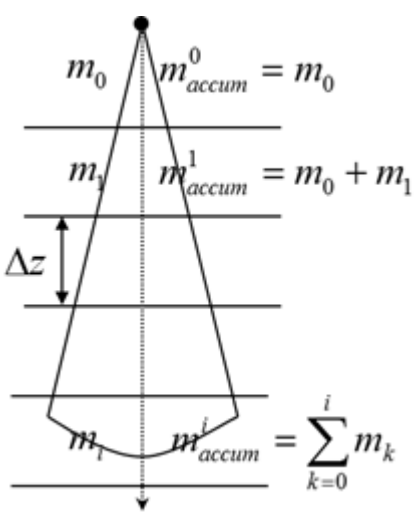
- To use the Parcel Penetration report, activate the Material Particle physics model and an unsteady time model. The report only includes mass from the injector it is associated with
- To use the Vapor Penetration report, activate an evaporation physics model. The report does not differentiate the components of a multicomponent vapor. It measures the total mass of all components.

The reports measure the distance along the injector axis. In the following diagrams, the injection points are at the top. The reports lay out grids with intervals Δz along the injection axes. The accumulated mass, progressing along anaxis, is:

$$m^i_{accum} = \sum_{k=0}^{i} m_k \tag{1267}$$

where:

- m_i is the amount of mass in the ith interval.
- m^i_{accum} is the cumulative mass from the injection point to the ith interval.



The penetration distance z_X where fraction X of the mass is accumulated is:

$$z_X = i \cdot \Delta z + \Delta z_X \tag{1268}$$

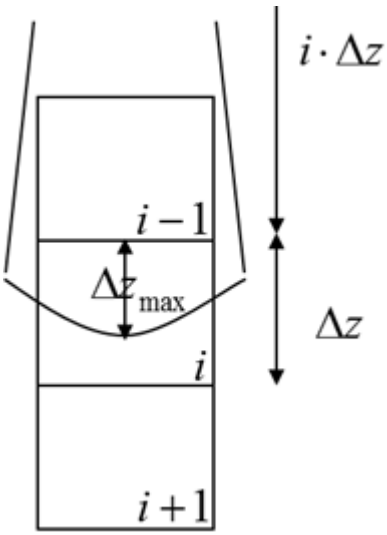
For parcel penetration: $\Delta z_X = \min\left(\Delta z, \Delta z_{\max}\right) \cdot \frac{X \cdot m_{total} - m^{i-1}_{accum}}{m_i}$

For vapor penetration: $\Delta z_X = \Delta z \cdot \frac{X \cdot m_{total} - m_{accum}^{i-1}}{m_i}$

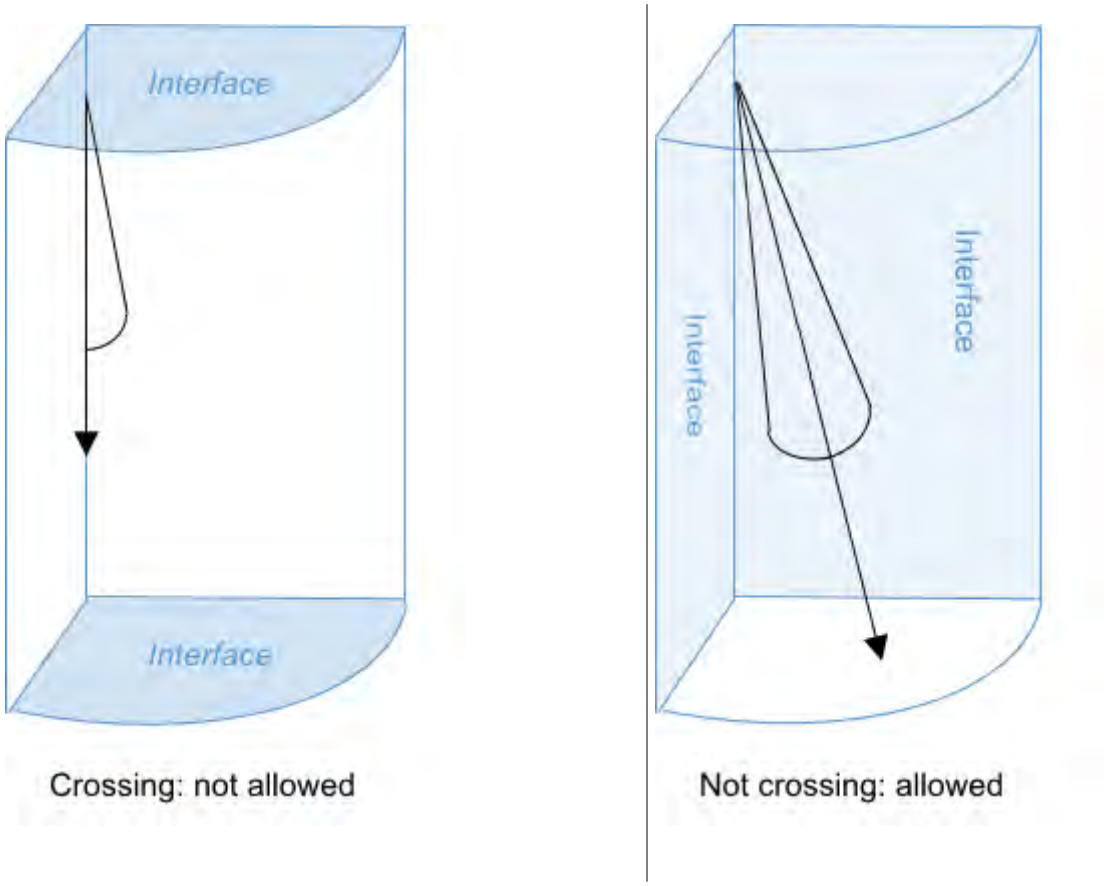
where:

- $\Delta z_{\max}$ is the maximum distance parcels penetrate into the ith interval.
- m_{total} is the total accumulated parcel or vapor mass.

The calculation assumes mass is distributed evenly along the axis in the ith interval.



The report can be used with periodic interfaces only if the injector axis does not cross a periodic interface.



Penetration Report Properties

Units	Units that are used for specifying the quantity.
Cut parameter	The mass percentage used for defining the penetration distance. The default value is `0.95`.
Grid interval	The spacing for the report grid, controlling output resolution. A finer spacing gives a more exact distance, at a higher computational cost. The default value is `5.0E-4` m.

Penetration Report Expert Properties

Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.
Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Particle Mass Flow

The particle mass flow report describes the rate at which mass in the form of particles flows across a specified boundary. It does not report mass flow due to continuous phases.

The mass flow rate of particles i through a boundary B is computed as:

$$\dot{m} = \sum_{f} \dot{m}_f \tag{1269}$$

where $\dot{m}_f$ is the mass flow rate of Lagrangian parcels crossing each face f. The sign convention is for a positive value of $\dot{m}$ to indicate mass flow out of the domain.

Note: Particles rebounding or sticking at a boundary are not considered to be crossing the boundary, and hence do not contribute to the mass flow rate.

For each face,

$$\dot{m}_f = \sum_{i} \dot{m}_{p,i} \tag{1270}$$

where $\dot{m}_{p,i}$ is the particle mass flow rate of the *ith* parcel, for parcels crossing face f.

Particle Mass Flow Report Properties

Phase	The phase for which the mass report is run.
Units	Units that are used for specifying the quantity.
Parts	A list of parts that is used as the source for this report. Uses the in-place part selector for editing.

Particle Mass Flow Report Expert Properties

Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.
Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Pressure Correction

The report calculates the scaled correction to the pressure calculation at the end of each iteration.

$$\overline{C}_p = \frac{1}{4\sigma_p}\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\delta p_i)^2} \tag{1271}$$

- $\overline{C}_p$ is the scaled correction for the pressure calculation.
- N is the number of cells.
- δp_i is the pressure correction in the ith cell.
- σ_p is the standard deviation of the pressure:

$$\sigma_p = \sqrt{\frac{\sum_{i=1}^{N}(p_i - \overline{p})^2}{N}}$$

and

$$\overline{p} = \frac{1}{N}\sum_{i=1}^{N} p_i$$

Pressure Drop

The report calculates the mass-flow averaged absolute total pressure difference between two groups of boundary surfaces or interfaces representing the high and low pressure boundaries of the system.

$$dp = \left[\frac{\sum_f \left|\dot{m}_f\right| P_{t,abs_f}}{\sum_f \left|\dot{m}_f\right|}\right]_H - \left[\frac{\sum_f \left|\dot{m}_f\right| P_{t,abs_f}}{\sum_f \left|\dot{m}_f\right|}\right]_L \tag{1272}$$

Pressure Drop Report Properties

Units	Units that are used for specifying the quantity.
High Pressure	Specify the boundary surfaces or interfaces with the high pressure value for which to calculate the pressure drop. Click ... (**custom editor**) to open the object selection dialog.
Low Pressure	Specify the boundary surfaces or interfaces with the low pressure value for which to calculate the pressure drop.

Pressure Drop Report Expert Properties

Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.
Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Reaction Force

The reaction force report is available only when the Solid Stress model is activated. It calculates the reaction force and applied force on a surface in a user-specified direction.

As well as the reaction and applied forces, the report also displays the body forces acting on the body. The reaction force report is represented in the simulation tree by a node, which has its own properties but is subject to the report node's pop-up menu.

The reaction force on a surface is the summation of the forces due to prescribed displacements, and fixed and elastic foundation boundary conditions. It is computed as:

$$\mathbf{f}_{reaction} = \sum_f \left(\mathbf{f}_f^{disp.} + \mathbf{f}_f^{n.\ disp.} + \mathbf{f}_f^{c.\ found.} + \mathbf{f}_f^{fixed} + \mathbf{f}_f^{mixed-disp.}\right) \cdot \mathbf{n}_f \tag{1273}$$

where $\mathbf{f}_f^{disp.}$, $\mathbf{f}_f^{n.\ disp.}$, $\mathbf{f}_f^{c.\ found.}$, $\mathbf{f}_f^{fixed}$ and $\mathbf{f}_f^{mixed-disp.}$ are the force vectors due to displacement, normal displacement, an elastic foundation, a fixed condition, and the displacement part of a mixed boundary condition on the surface face f, respectively. The variable $\mathbf{n}_f$ is a user-specified direction vector that indicates the direction in which the force should be computed.

The displacement force vector f on surface face is computed as:

$$\mathbf{f}_f^{disp.} = \sigma_f \cdot \mathbf{a}_f \tag{1274}$$

where σ_f is the stress tensor and $\mathbf{a}_f$ is the face area vector. The force vectors due to an elastic foundation, a fixed boundary condition, and the displacement part of the mixed boundary condition, $\mathbf{f}_f^{c.\ found.}$, $\mathbf{f}_f^{fixed}$ and $\mathbf{f}_f^{mixed-disp.}$ are also computed using this equation.

The normal displacement force vector on surface face f is computed as:

$$\mathbf{f}_f^{n.\ disp.} = \left(\left(\sigma_f \cdot \mathbf{a}_f\right) \cdot \mathbf{n}_f\right)\ \mathbf{n}_f \tag{1275}$$

where $\mathbf{n}_f$ is the unit vector normal to the cell face.

The applied force on a surface is the summation of pressures, tractions, and forces, and it is computed as:

$$\mathbf{f}_{applied} = \sum_f \left(\mathbf{f}_f^{pressure} + \mathbf{f}_f^{traction} + \mathbf{f}_f^{force} + \mathbf{f}_f^{mixed-traction}\right) \cdot \mathbf{n}_f \tag{1276}$$

where $\mathbf{f}_f^{pressure}$, $\mathbf{f}_f^{traction}$, $\mathbf{f}_f^{force}$ and $\mathbf{f}_f^{mixed-traction}$ are the force vectors due to pressure, traction, force, and the traction part of the mixed boundary condition on the surface face f. The variable $\mathbf{n}_f$ is a user-specified direction vector that indicates the direction in which the force is computed.

The pressure force vector on surface face f is computed as:

$$\mathbf{f}_f^{pressure} = p_f \mathbf{a}_f \tag{1277}$$

where p_f is the applied face pressure, or the fluid gauge pressure if the surface is a fluid-solid interface, and $\mathbf{a}_f$ is the face area vector.

The traction force vector on surface face f is computed as:

$$\mathbf{f}_f^{traction} = \mathbf{t}_f \|\mathbf{a}_f\| \tag{1278}$$

where $\mathbf{t}_f$ is the traction vector. The force vector due to the traction part of the mixed boundary condition, $\mathbf{f}_f^{mixed-traction}$, is also computed using this equation.

Reaction Force Report Properties

Coordinate System	A list of coordinate systems in the simulation for use by this report.
Direction	A Cartesian vector that indicates the direction in which the force is computed.

Parts	A list of parts that is used as the source for this report. Opens the in-place part selector for editing. The ... (**property customizer**) button activates a modeless version of the selector dialog (it lets you interact with other portions of the workspace).
Units	Units that are used for specifying the quantity.

Reaction Force Report Expert Properties

Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.
Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Reaction Moment

The reaction moment report is available only when the Solid Stress model is activated.

It is represented in the simulation tree by a node, which has its own properties but is subject to the report node's pop-up menu.

The reaction moment due to the reaction forces on a surface about an axis is defined:

$$M_{reaction} = \sum_f \left[\mathbf{r}_f \times \left(\mathbf{f}_f^{disp.} + \mathbf{f}_f^{n.\ disp} + \mathbf{f}_f^{c.\ found.} + \mathbf{f}_f^{fixed} + \mathbf{f}_f^{mixed-disp.} \right) \cdot \mathbf{n}_f \right] \cdot \mathbf{a} \tag{1279}$$

where $\mathbf{f}_f^{disp.}$, $\mathbf{f}_f^{n.\ disp.}$, $\mathbf{f}_f^{c.\ found.}$, $\mathbf{f}_f^{fixed}$ and $\mathbf{f}_f^{mixed-disp.}$ are the force vectors due to displacement, normal displacement, an elastic foundation, a fixed condition, and the displacement part of a mixed boundary condition on the surface face f, respectively. The variable $\mathbf{a}$ is a vector defining the axis through point $\mathbf{x}_0$ about which the moment is taken and $\mathbf{r}_f$ is the position of face f relative to $\mathbf{x}_0$.

The displacement force vector on surface face is computed as:

$$\mathbf{f}_f^{disp.} = \sigma_f \cdot \mathbf{a}_f \tag{1280}$$

where σ_f is the stress tensor and $\mathbf{a}_f$ is the face area vector. The force vectors due to an elastic foundation, a fixed boundary condition, and the displacement part of the mixed boundary condition, $\mathbf{f}_f^{c.\ found.}$, $\mathbf{f}_f^{fixed}$ and $\mathbf{f}_f^{mixed-disp.}$ are also computed using this equation.

The normal displacement force vector on surface face is computed as:

$$\mathbf{f}_f^{n.\ disp.} = \left(\left(\sigma_f \cdot \mathbf{a}_f \right) \cdot \mathbf{n}_f \right) \mathbf{n}_f \tag{1281}$$

where $\mathbf{n}_f$ is the unit vector normal to the cell face.

The reaction moment due to the applied forces on a surface about an axis is defined:

$$M_{applied} = \sum_f \left[\mathbf{r}_f \times \left(\mathbf{f}_f^{pressure} + \mathbf{f}_f^{traction} + \mathbf{f}_f^{force} + \mathbf{f}_f^{mixed-traction}\right)\right] \cdot \mathbf{a} \tag{1282}$$

where $\mathbf{f}_f^{pressure}$, $\mathbf{f}_f^{traction}$, $\mathbf{f}_f^{force}$ and $\mathbf{f}_f^{mixed-traction}$ are the force vectors due to pressure, traction, force, and the traction part of the mixed boundary condition on the surface face f. The variable $\mathbf{n}_f$ is a user-specified direction vector that indicates the direction in which the force is computed.

The pressure force vector on surface face f is computed as:

$$\mathbf{f}_f^{pressure} = p_f \mathbf{a}_f \tag{1283}$$

where p_f is the applied face pressure, or the fluid gauge pressure if the surface is a fluid-solid interface, and $\mathbf{a}_f$ is the face area vector.

The traction force vector on surface face f is computed as:

$$\mathbf{f}_f^{traction} = \mathbf{t}_f \|\mathbf{a}_f\| \tag{1284}$$

where $\mathbf{t}_f$ is the traction vector. The force vector due to the traction part of the mixed boundary condition, $\mathbf{f}_f^{mixed-traction}$, is also computed using this equation.

Reaction Moment Report Properties

Axis	A Cartesian vector specifying the axis $\mathbf{a}$ of Eqn. (1279) and Eqn. (1282) about which the moment is computed.
Axis Origin	A Cartesian position vector specifying the point $\mathbf{x}_0$ on the axis $\mathbf{a}$ about which the moment is taken.
Coordinate System	A list of coordinate systems in the simulation for use by this report.
Parts	A list of parts that is used as the source for this report. Opens the in-place part selector for editing. The ... (**property customizer**) button activates a modeless version of the selector dialog (it lets you interact with other portions of the workspace).
Units	Units that are used for specifying the quantity.

Reaction Moment Report Expert Properties

Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.
Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Rigid Body Acceleration

This report type is available only when 6-DOF Motion is activated.

See also the properties of this 6-DOF report.

This report calculates the linear acceleration of a body in a given direction. It is determined as follows:

$$a_{\mathbf{d}} = \frac{d\mathbf{v}}{dt}\mathbf{d} \tag{1285}$$

Here, $a_{\mathbf{d}}$ is the acceleration of the body in the direction $\mathbf{d}$, $\mathbf{v}$ is the velocity vector of the body, and t is the time.

Rigid Body Angle

This report type is available only when 6-DOF Motion is activated.

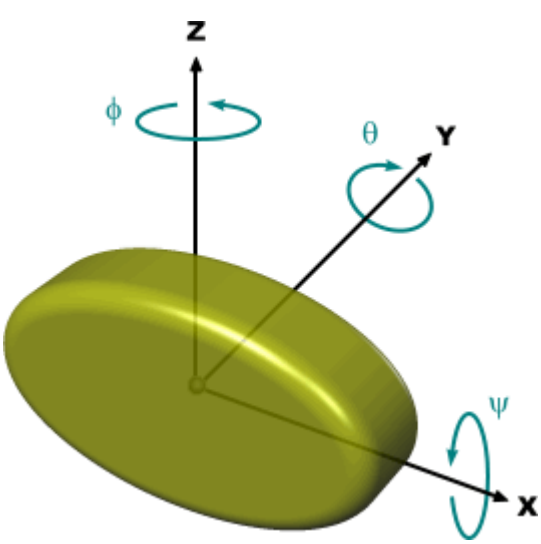
It has its own properties but is subject to the report node's pop-up menu.

This report calculates the angle of rotation of a body relative to the initial body orientation. The angle of rotation is either constrained in the interval $[-\pi, \pi]$, or unconstrained to reflect the full rotation of the body:

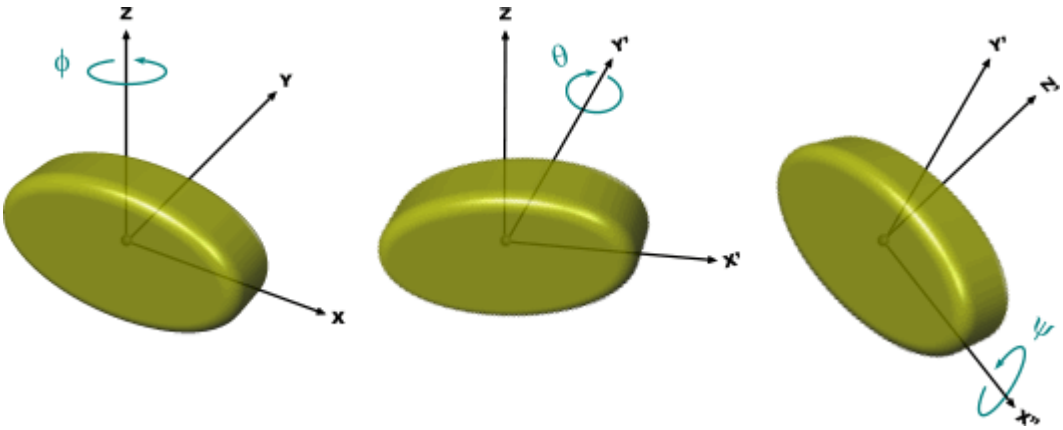
- Using unconstrained angles is advantageous in cases with large angular motions of the body (for example, a body undergoing multiple revolutions around an axis). When choosing the unconstrained angle option, the reported rotation angle reflects the full rotation of the body. For example, if the body has made two revolutions, the reported angle is 4π. The default option is to report the unconstrained angle.
- The constrained angle is clamped to the interval $[-\pi, \pi]$, or $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ depending on the selected *Angle Report Option* and *Euler Angle Convention*. The constrained angle is provided when backward compatibility is required.

Euler Angle Convention

The *Euler Angle Convention* is interpreted in the following manner. Consider **Rotation X-Y-Z Axis** as an example. For this convention, three rotations describe the orientation of the body. The first rotation is around the x axis of the lab coordinate system (giving the first Euler angle). The second rotation is around the y axis of the lab system (giving the second Euler angle). The third is a rotation around the z axis of the lab system (giving the third Euler angle). The Euler angles are given with respect to fixed rotation axes.



It is also possible to interpret the above Euler angles with respect to moving rotation axes that result from previous rotations. For this purpose, read the axes from back to front. For example, **Rotation X-Y-Z Axis** means that the first rotation is about the z axis of the lab system. The second is around the y axis of the new coordinate system created by rotating the lab system around the z axis. The third rotation is about the x axis of another new coordinate system created by rotating the previous system around its y axis.



The order of rotation is read in the reverse order for the following reason: Let R_X, R_Y and R_Z denote the rotation matrices around the X, Y and Z axes of the laboratory system respectively. For the **Rotation X-Y-Z Axis** option, the overall matrix is $R_Z R_Y R_X$. The same matrix is obtained for moving axes with the order of rotations reversed. If the rotation is first around Z, then Y' (obtained after rotation about Z) and X'' (obtained after a double rotation) - where the primes indicate moved coordinate system axes - then

$$R_{X''} R_{Y'} R_Z = R_Z R_Y R_X \tag{1286}$$

Gimbal Locks

A gimbal lock occurs when the second Euler angle reaches $\frac{\pi}{2}$ or $-\frac{\pi}{2}$. In a gimbal lock, only the sum of the first and the third Euler angles is determined. The reported Euler angles describe the orientation of the body correctly even in a gimbal lock. However, plots of Euler angles over time can show discontinuities. Therefore, try to avoid gimbal locks by choosing an Euler Angle Convention in which the Euler angle reaching a magnitude of $\frac{\pi}{2}$ is the first or the third angle.

Rigid Body Angle Report Properties

Angle Report Option	The axis of the coordinate system relative to which rotation is measured. For fixed axes, the axes of the lab coordinate system are used. For moving axes, the corresponding rotated axes of the lab system are used.	
	Rotation Z Axis	Select rotation about the z-axis.
	Rotation Y Axis	Select rotation about the y-axis.
	Rotation X Axis	Select rotation about the x-axis.

Euler Angle Convention	Select the order of the rotations describing the orientation of the body. For fixed axes, the first rotation is around the first axis of the lab system in the option (giving the first Euler Angle). The second rotation is around the second mentioned axis of the lab system (giving the second Euler Angle). The third rotation is around the last axis of the lab system (giving the third Euler angle). For moving axes, read the order of rotation from back to front.	
	Rotation Y-Z-X Axis	The order of rotation is first Y, then Z and X for fixed axes.
	Rotation X-Y-Z Axis	The order of rotation is first X, then Y and Z for fixed axes.
	Rotation Y-X-Z Axis	The order of rotation is first Y, then X and Z for fixed axes.
	Rotation Z-X-Y Axis	The order of rotation is first Z, then X and Y for fixed axes.
	Rotation X-Z-Y Axis	The order of rotation is first X, then Z and Y for fixed axes.
	Rotation Z-Y-X Axis	The order of rotation is first Z, then Y and X for fixed axes.
Constrained Angle Range	Whether to report the unconstrained angle or the constrained angle in the interval $[-\pi, \pi]$.	
	Activated	Report the constrained angle.
	Deactivated	Report the unconstrained angle. (default)
Body	The 6-DOF body for which the angle of rotation is reported.	
Units	Units that are used for specifying the quantity.	

Rigid Body 1-DOF Angle

This report type is available only when 6-DOF Motion is activated.

It has its own properties but is subject to the report node's pop-up menu.

This report calculates the angle of rotation of a body undergoing One-DOF Rotating Motion relative to the initial body orientation. It reports zero for all other types of motions.

Rigid Body 1-DOF Angle Report Properties

Body	The 6-DOF body for which the angle of rotation is reported.
Units	Units that are used for specifying the quantity.

Rigid Body Angular Acceleration

This report type is available only when 6-DOF Motion is activated.

See also the properties of this 6-DOF report.

This report calculates the angular acceleration of a body with respect to a given direction. It is determined as follows:

$$\alpha_{\mathbf{d}} = \frac{d\vec{\omega}}{dt}\mathbf{d} \tag{1287}$$

Here, $\alpha_{\mathbf{d}}$ is the angular acceleration of the body with respect to the direction $\mathbf{d}$, $\vec{\omega}$ is the angular velocity vector of the body, and t is the time.

Rigid Body Angular Momentum

This report type is available only when 6-DOF Motion is activated.

See also the properties of this 6-DOF report.

This report calculates the angular momentum of a body with respect to a given direction. It is determined as follows:

$$L_{\mathbf{d}} = \mathbf{d} \cdot I\vec{\omega} \tag{1288}$$

Here, $L_{\mathbf{d}}$ is the angular momentum of the body with respect to the direction $\mathbf{d}$, I is the moment of inertia tensor of the body, and $\vec{\omega}$ is the angular velocity vector of the body.

Rigid Body Angular Velocity

This report type is available only when 6-DOF Motion is activated.

See also the properties of this 6-DOF report.

The result of this report is the angular velocity of a body with respect to a given direction. It is calculated as follows:

$$\omega_{\mathbf{d}} = \vec{\omega} \cdot \mathbf{d} \tag{1289}$$

Here, $\omega_{\mathbf{d}}$ is the angular velocity of the body with respect to the direction , and is the angular velocity vector of the body.

Rigid Body Component Distance

This report type is available only when 6-DOF Motion is activated.

It has its own properties but is subject to the report node's pop-up menu.

The report calculates the distance between two points along a user-specified direction, that is,

$$r_p = (\mathbf{x}_2 - \mathbf{x}_1) \cdot \mathbf{s}_d \tag{1290}$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are the two user-specified positions and $\mathbf{s}_d$ is the direction vector.

Rigid Body Component Distance Report Properties

Coordinate System	A list of coordinate systems in the simulation for use in defining the *Direction*.
Direction	Direction vector $\mathbf{s}_d$ on which the distance is projected.
Position 1	The coordinates of the first point, $\mathbf{x}_1$, with respect to *Coordinate System 1*.
Position 2	The coordinates of the second point, $\mathbf{x}_2$, with respect to *Coordinate System 2*.
Coordinate System 1	A list of coordinate systems in the simulation for use in defining *Position 1*.
Coordinate System 2	A list of coordinate systems in the simulation for use in defining *Position 2*.
Units	Units of length in which the distance is reported.

6-DOF Constraint Deviation

This report is available only for 6-DOF motion. You typically use this report when performing a multi-body DFBI simulation. The report calculates the maximum normalized deviation from the constraint equations. The maximum normalized deviation is the one maximum value of all mechanical joints that are defined in the simulation.

For more information, see Constraint Stabilization.

Depending on the type of mechanical joint, the deviations are translational or angular. A spherical joint can have translational deviations only. A revolute joint can have translational and angular deviations. Translational deviations are normalized by the extent of the smallest of the two bodies that are attached to the joint. The angular deviation for a revolute joint is the sine of the angle between the two rotation axes that characterize the revolute joint (one axis for each of the attached bodies).

For example, a reported value of 10^{-5} means that all translational deviations (normalized per joint by the smallest of the attached bodies) and all angular deviations (sine of deviation angles) are below or equal to 10^{-5}.

See Also:

Setting up Multi-Body Motion

Rigid Body Force

This report type is available only when 6-DOF Motion is activated.

It has its own properties but is subject to the report node's pop-up menu.

The resultant force $\mathbf{f}$ acting on the body is computed as:

$$\mathbf{f} = \left(\mathbf{f}_g + \mathbf{f}_{ff} + \mathbf{f}_{ext}\right) f_r \tag{1291}$$

where f_r is the ramping factor and $\mathbf{f}_g$ is the gravity force acting on the body, computed as:

$$\mathbf{f}_g = m\mathbf{g} \tag{1292}$$

where $\mathbf{g}$ is the gravitational vector and m is the mass of the body.

The fluid force $\mathbf{f}_{ff}$ acting on the body is calculated as specified in Eqn. (1242). You specify the external force $\mathbf{f}_{ext}$.

Rigid Body Force Report Properties

Coordinate System	A list of coordinate systems in the simulation for use by this report.
Direction	A Cartesian vector that indicates the direction in which the force is computed.
Forces	Specifies which of the body forces to include in the report. The reported value is the sum of the selected forces. Clicking the custom editor button opens the *Forces* selection dialog.
Body	A list of 6-DOF bodies in the simulation for use by this report.
Units	Units that are used for specifying the quantity.

Rigid Body Force Report Expert Properties

Use Raw Forces	Provides option to report resultant force without smoothing or ramping.	
	Activated	The instantaneous force acting on the body (without smoothing or ramping) is reported.
	Deactivated	The instantaneous force as applied in the solver (after any smoothing and ramping) is reported.

Rigid Body Moment

This report type is available only when 6-DOF Motion is activated.

It has its own properties but is subject to the report node's pop-up menu.

The resultant force $\mathbf{n}$ acting on the body (Eqn. (2182)) is computed as:

$$\mathbf{n} = \left(\mathbf{n}_{ff} + \mathbf{n}_{ext}\right) f_r \tag{1293}$$

where f_r is the ramping factor.

The fluid moment $\mathbf{n}_{ff}$ acting on the body is calculated as specified in Eqn. (1260) with respect to the current center of mass of the body. You specify the external moment $\mathbf{n}_{ext}$.

Rigid Body Moment Report Properties

Coordinate System	A list of coordinate systems in the simulation for use by this report.
Direction	A Cartesian vector that indicates the direction in which the force is computed.
Moments	Specifies which of the body moments to include in the report. The reported value equals the sum of all the moments. Clicking the custom editor button opens the *Moments* selection dialog.
Body	A list of 6-DOF bodies in the simulation for use by this report.
Units	Units that are used for specifying the quantity.

Rigid Body Moment Report Expert Properties

Use Raw Moments	Provides option to report moments without smoothing or ramping.	
	Activated	The instantaneous moment acting on the body (without smoothing or ramping) is reported.
	Deactivated	The instantaneous moment as applied in the solver (after any smoothing and ramping) is reported.

Rigid Body Rotational Energy

This report type is available only when 6-DOF Motion is activated.

It has its own properties but is subject to the report node's pop-up menu.

This report calculates the rotational energy of a body. It is determined as follows:

$$K_{rot} = \frac{1}{2} I \vec{\omega} \cdot \vec{\omega} \tag{1294}$$

Here, K_{rot} is the rotational energy of the body, I is the moment of inertia tensor of the body, and $\vec{\omega}$ is the angular velocity vector of the body.

Rigid Body Rotational Energy Report Properties

Body	The 6-DOF body for which the rotational energy is reported.
Units	Units of length in which the distance is reported.

Rigid Body Spring Elongation

This report type is available only when 6-DOF Motion is activated.

It has its own properties but is subject to the report node's pop-up menu.

The report calculates the elongation of a linear spring that is held between a 6-DOF body and the environment, or between two 6-DOF bodies, as defined in a linear spring coupling. If the *Total Length* property is activated, the reported length is:

$$r_p = |\mathbf{x}_2 - \mathbf{x}_1| \tag{1295}$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are the end-points where the spring is attached. If the *Total Length* property is not activated, the reported length is:

$$r_p = |\mathbf{x}_2 - \mathbf{x}_1| - l_{eq} \tag{1296}$$

where l_{eq} is the relaxation length of the spring.

Rigid Body Spring Elongation Report Properties

Linear Spring Coupling	A list of linear spring couplings for use by this report.	
Total Length	Sets whether to include the relaxation length of the spring in the reported distance or not.	
	Activated	The total length of the spring including its relaxation length is reported.
	Deactivated	The relaxation length of the spring is subtracted from the total length.
Units	Units of length in which the distance is reported.	

Rigid Body Catenary Length

This report type is available only when 6-DOF Motion is activated. It has its own properties but is subject to the report node's pop-up menu.

The report calculates the length of a catenary that is held between a 6-DOF body and the environment, or between two 6-DOF bodies, as defined in a catenary coupling. If the *Total Length* property is activated, the reported length is:

$$r_p = L = \int_{\mathbf{x}_1}^{\mathbf{x}_2} \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx \tag{1297}$$

where L is the total length of the catenary, y(x) is the catenary curve defined in Eqn. (2244), and $\mathbf{x}_1$ and $\mathbf{x}_2$ are the end-points where the catenary is attached. If the *Total Length* property is not activated, the reported length is:

$$r_p = L - l_{eq} \tag{1298}$$

where l_{eq} is the relaxation length of the catenary.

Rigid Body Catenary Report Properties

Catenary Coupling	A list of catenary couplings for use by this report.	
Total Length	Sets whether to include the relaxation length of the catenary in the reported distance or not.	
	Activated	The total length of the catenary including its relaxation length is reported.
	Deactivated	The relaxation length of the catenary is subtracted from the total length.
Units	Units of length in which the catenary length is reported.	

Rigid Body Total Distance

This report type is available only when 6-DOF Motion is activated.

It has its own properties but is subject to the report node's pop-up menu.

The report calculates the absolute value of the distance r_p between two given points, that is,

$$r_p = |\mathbf{x}_2 - \mathbf{x}_1| \tag{1299}$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are two user-specified positions.

Rigid Body Total Distance Report Properties

Position 1	The coordinates of the first point, $\mathbf{x}_1$, with respect to *Coordinate System 1*.
Position 2	The coordinates of the second point, $\mathbf{x}_2$, with respect to *Coordinate System 2*.

Coordinate System 1	A list of coordinate systems in the simulation for use in defining *Position 1*.
Coordinate System 2	A list of coordinate systems in the simulation for use in defining *Position 2*.
Units	Units of length in which the distance is reported.

Rigid Body Translation

This report type is available only when 6-DOF Motion is activated.

See also the properties of this 6-DOF report.

This report calculates the translational displacement of a body with respect to its initial position in a given direction. It is determined as follows:

$$\delta_{\mathbf{d}} = \vec{\delta} \cdot \mathbf{d} \tag{1300}$$

Here, $\delta_{\mathbf{d}}$ is the translational displacement of the body in the direction $\mathbf{d}$, and $\vec{\delta}$ is the translational displacement vector of the body.

Rigid Body Velocity

This report type is available only when 6-DOF Motion is activated.

See also the properties of this 6-DOF report.

This report calculates the velocity of a body in a given direction. It is determined as follows:

$$v_{\mathbf{d}} = \mathbf{v} \cdot \mathbf{d} \tag{1301}$$

Here, $v_{\mathbf{d}}$ is the velocity of the body in the direction $\mathbf{d}$, and $\mathbf{v}$ is the velocity vector of the body.

Sauter Mean Diameter

This report type is available only when the Lagrangian Multiphase model is activated.

It has its own properties but is subject to the report node's pop-up menu.

The Sauter mean diameter (SMD) applies to parcels, with the Material Particles model.

$$SMD = \frac{\sum_{\pi} D_p^3 n_{\pi}}{\sum_{\pi} D_p^2 n_{\pi}} \tag{1302}$$

in which D_p is the particle diameter and n_{π} is the number of particles that is represented by a parcel: the "particle count".

Sauter Mean Diameter Report Properties

Parts	A list of parts that is used as the source for this report. Opens the in-place part selector for editing. The ... (**property customizer**) button activates a modeless version of the selector dialog (it lets you interact with other portions of the workspace).
Units	Units that are used for specifying the quantity.

Sauter Mean Diameter Report Expert Properties

Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.
Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Temperature Correction

The report calculates the scaled correction to the temperature calculation at the end of each iteration. This report becomes available when models for energy, temperature, or enthalpy are used.

$$\overline{C}_T = \frac{1}{4\sigma_T}\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\delta T_i)^2} \tag{1303}$$

- $\overline{C}_T$ is the scaled correction for the temperature calculation.
- N is the number of cells.
- δT_i is the temperature correction in the ith cell.
- σ_T is the standard deviation of the temperature:

$$\sigma_T = \sqrt{\frac{\sum_{i=1}^{N}(T_i - \overline{T})^2}{N}}$$

and

$$\overline{T} = \frac{1}{N}\sum_{i=1}^{N} T_i$$

Thrust

The thrust report integrates the momentum flux and the pressure force over a specified boundary or derived part.

The thrust report is represented in the simulation tree by a node, which has its own properties but is subject to the report node's pop-up menu. This report is active for flow models.

The thrust is computed as:

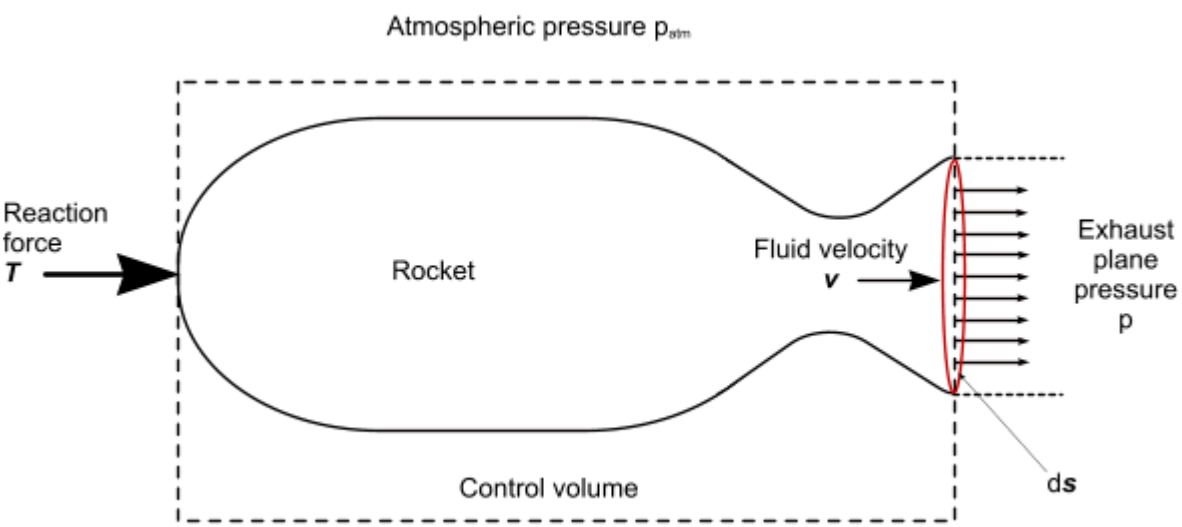
$$T = \sum_f \rho_f \mathbf{v}_f \mathbf{v}_f \cdot \mathbf{a}_f + \sum_f \left(p_f - p_{atm} \right) \cdot \mathbf{a}_f \tag{1304}$$

where ρ_f is the fluid density, $\mathbf{v}$ is the velocity vector, p_f is the exit pressure, p_{atm} is the atmospheric pressure, and $\mathbf{a}_f$ is the surface area vector of the face.

The governing equation that is being solved is the reaction force at the boundary:

$$T = \int \rho \mathbf{v}\mathbf{v} \cdot d\mathbf{s} + \int \left(p - p_{atm} \right) d\mathbf{s} \tag{1305}$$

This equation is illustrated in the following diagram:



Thrust Report Properties

Direction	Specifies the Cartesian vector that is used to compute the scalar product with the reaction thrust vector. This lets you calculate the component of the thrust that is acting in a particular direction.
Atmospheric Pressure	The atmospheric pressure for the computation of Eqn. (1304).
Coordinate System	A list of coordinate systems that this report can use.
Parts	A list of parts that are used as the source for this report. Opens the in-place part selector for editing. The ... (**property customizer**) button activates a modeless version of the selector dialog (it lets you interact with other portions of the workspace).
Units	Units that are used for specifying the quantity.

Thrust Report Expert Properties

Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.
Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Tumble/Swirl Ratio

This report is typically used for ICE applications and calculates swirl or tumble (dimensionless).

In the engine industries, typically three index numbers are used to quantify in-cylinder flow motion:

- Swirl:

  Rotational motion with its axis aligned with the cylinder axis.
- Main-tumble:

  Rotational motion with its axis normal to the cylinder axis and the symmetry plane of the ports (assuming a symmetric 4-valve engine).
- Cross-tumble:

  Rotational motion with its axis normal to the cylinder axis and parallel to the symmetry plane of the ports.

Using the STAR-CCM+ swirl report, all three index numbers can be derived by adjusting the report properties accordingly:

Property	Value
Coordinate System	Set a local coordinate system with its Z-axis aligned with the cylinder axis and its X-axis parallel to the symmetry plane of the ports.
Axis	Set one of the following: • `[0,0,1]` for swirl. • `[0,1,0]` for main-tumble (also referred to as Y-tumble). • `[1,0,0]` for cross-tumble (also referred to as X-tumble).

On the other hand, the STAR-CCM+ tumble report can be used to calculate a combination of cross-tumble and main-tumble, as the formulation explains:

Tumble is defined as:

$$T = \frac{1}{2}\left(T_u^2 + T_v^2 + T_w^2\right) \tag{1306}$$

with

$$\begin{aligned} T_u &= Axis_y \cdot S_z - Axis_z \cdot S_y \\ T_v &= Axis_z \cdot S_x - Axis_x \cdot S_z \\ T_w &= Axis_x \cdot S_y - Axis_y \cdot S_x \end{aligned} \tag{1307}$$

which, written in vector-form, corresponds to:

$$\mathbf{T} = \mathbf{Axis} \times \mathbf{S} \tag{1308}$$

Swirl is defined as:

$$Sw = Axis_x \cdot S_x + Axis_y \cdot S_y + Axis_z \cdot S_z \tag{1309}$$

where:

$$S_x = \frac{2\pi}{\omega_c}\frac{M_x}{I_x} = \frac{2\pi}{\omega_c}\frac{\sum_{\text{Cells}} \rho_i V_i\left[\left(Y_i - Y_m\right)w_i - \left(Z_i - Z_m\right)v_i\right]}{\sum_{\text{Cells}} \rho_i V_i\left[\left(Y_i - Y_m\right)^2 + \left(Z_i - Z_m\right)^2\right]} \tag{1310}$$

$$S_y = \frac{2\pi}{\omega_c}\frac{M_y}{I_y} = \frac{2\pi}{\omega_c}\frac{\sum_{\text{Cells}} \rho_i V_i\left[\left(Z_i - Z_m\right)u_i - \left(X_i - X_m\right)w_i\right]}{\sum_{\text{Cells}} \rho_i V_i\left[\left(X_i - X_m\right)^2 + \left(Z_i - Z_m\right)^2\right]} \tag{1311}$$

$$S_z = \frac{2\pi}{\omega_c}\frac{M_z}{I_z} = \frac{2\pi}{\omega_c}\frac{\sum_{\text{Cells}} \rho_i V_i\left[\left(X_i - X_m\right)v_i - \left(Y_i - Y_m\right)u_i\right]}{\sum_{\text{Cells}} \rho_i V_i\left[\left(Y_i - Y_m\right)^2 + \left(X_i - X_m\right)^2\right]} \tag{1312}$$

where:

ω_c	Angular speed of crankshaft (rad/s)
M_x, M_y, M_z	Angular momentum about the x-, y-, and z-axes (kg m^2/s)
I_x, I_y, I_z	Mass moment of inertia about the x-, y-, and z-axes (kg m^2)
X_i, Y_i, Z_i	Centroid coordinate of cell (m)
X_m, Y_m, Z_m	Center of mass/volume/origin of the cylinder (m)
u_i, v_i, w_i	Velocity at cell (m/s)

Tumble/Swirl Report Properties

Units	Units that are used for specifying the quantity.	
Evaluation Mode	**Tumble**	Selects the tumble report. Tumble is defined as the fluid rotation about the plane that contains the *Tumble Origin* and that is orthogonal to the cylinder *Axis*. For example, if the cylinder *Axis* is `[0, 0, 1]`, that is, the Z direction, $T_w = 0$ and $T = \frac{1}{2}\sum_{\text{Cells}}\left(T_u^2 + T_v^2\right)$ as given by Eqn. (1306) and Eqn. (1307).
	Swirl	Selects the swirl report. Swirl is defined as the fluid rotation around the cylinder *Axis*.

		For example, if the cylinder *Axis* is `[0, 0, 1]`, that is, the Z direction, swirl corresponds to $Sw = S_z$ as given by Eqn. (1309).
Origin Mode	**Automatic**	Uses the center of mass to calculate the origin.
	Manual	Uses the point that you specify for *User Origin*.
User Origin	When *Origin Mode* is set to **Manual**, specifies the point within the cylinder. This point is usually the center of the cylinder and corresponds to X_m, Y_m, Z_m in Eqn. (1310) to Eqn. (1312). Defines the plane upon which the principal tumble axis lies. Tumble is calculated about this plane.	
Rotational Rate	Rotational speed of the IC engine. See ω_c in Eqn. (1310) to Eqn. (1312).	
Axis	Defines the orientation, that is, the axis of the cylinder with respect to *Coordinate System*.	
Coordinate System	A list of coordinate systems that this report can use.	
Parts	Selects the regions and geometry parts for which STAR-CCM+ computes the report. If multiple objects are selected, STAR-CCM+ calculates the report for each object individually, and then sums the results.	

Tumble/Swirl Report Expert Properties

Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.

Uniformity Deviation Report

This report calculates the deviation of the local normal velocity from the mass-flow averaged value on a specified set of boundary surfaces or on interfaces.

The uniformity deviation of the velocity normal to the surface (at each face $u_f = \mathbf{u}_f \cdot \mathbf{n}_f$) is:

$$\gamma = \sigma / \overline{u} \tag{1313}$$

where the variance of normal speed is:

$$\sigma^2 = \frac{\sum_f \left(u_f - \overline{u}\right)^2 \|\mathbf{a}_f\|}{\sum_f \|\mathbf{a}_f\|} \tag{1314}$$

and the average normal speed is:

$$\overline{u} = \frac{\sum_f u_f \cdot \mathbf{a}_f}{\sum_f \|\mathbf{a}_f\|} \tag{1315}$$

Uniformity Deviation Report Properties

Parts	A list of parts that is used as the source for this report. Opens the in-place part selector for editing.
Units	Units that are used for specifying the quantity.

Uniformity Deviation Report Expert Properties

Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.
Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Velocity Correction

The report calculates the scaled correction to the calculation for the X, Y or Z velocity at the end of each iteration.

There are three separate reports available: the X-Velocity Correction report, the Y-Velocity Correction report, and, for three-dimensional cases, the Z-Velocity Correction report.

$$\overline{C}_v = \frac{1}{4\sigma_v}\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\delta v_i)^2} \tag{1316}$$

- $\overline{C}_v$ is the scaled correction for the velocity calculation.
- N is the number of cells.
- δv_i is the velocity correction in the ith cell. v can be the X, Y, or Z velocity.
- σ_v is the standard deviation of the velocity:

$$\sigma_v = \sqrt{\frac{\sum_{i=1}^{N}(v_i - \overline{v})^2}{N}}$$

and

$$\overline{v} = \frac{1}{N}\sum_{i=1}^{N} v_i$$

Work Per Cycle

This report type is available only when Harmonic Balance Flutter motion is activated.

It has its own properties but is subject to the report node's pop-up menu.

This report computes the work done by the fluid on the structure (or the boundary surfaces selected). The work per cycle is computed as:

$$W = T\sum_f \hat{q}_f^0 \tag{1317}$$

where the summation is over all the faces of the boundary surface, T is the time period, and $\hat{q}_f^0$ is the mean of the integrand $\hat{q}_f^*$ (which represents the power at each time level) over all the time levels. The power at each time level is computed as:

$$\hat{q}_f^* = \hat{p}_f^*\left(D\mathbf{x}_f^*\right)\ \mathbf{a}_f^* \tag{1318}$$

where p_f^* is the time level pressure at the face, D is the pseudo-spectral time derivative operator, $\mathbf{x}_f^*$ is the time level displacement of the face, $\mathbf{v}_f^* = D\mathbf{x}_f^*$ is the time level velocity of the face, and $\mathbf{a}_f^*$ is the time level face area.

A negative value of the converged work-per-cycle shows that the blade is dynamically stable and does not flutter.

Work Per Cycle Report Properties

Parts	A list of parts that is used as the source for this report. Opens the in-place part selector for editing.
Units	Units that are used for specifying the quantity.

Work Per Cycle Report Expert Properties

Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.
Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Coal Volatile Matter Yield

This report type is available only when any of the Raw Coal Devolatilization models is activated.

It has its own properties but is subject to the report node's pop-up menu.

The Coal Volatile Matter (VM) Yield report applies to parcels with the Coal Combustion Model and is calculated as follows:

$$\text{VM Yield} = \frac{\text{Volatile Matter Produced}}{\text{Initial Mass of Raw Coal}} \tag{1319}$$

Coal Volatile Matter Report Properties

Parts	A list of parts that is used as the source for this report. Opens the in-place part selector for editing.
Units	Units that are used for specifying the quantity.

Coal Volatile Matter Report Expert Properties

Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.
Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Char Burn Out

This report type is available only when any of the Raw Char Oxidation models is activated.

It has its own properties but is subject to the report node's pop-up menu.

The Char Burn Out report applies to parcels with the Coal Combustion Model and is calculated as follows:

$$\text{Char Burn Out} = \frac{\text{Char Oxidized}}{(\text{Char Produced} + \text{Initial Mass of Char})} \tag{1320}$$

Char Burn Out Report Properties

Parts	A list of parts that is used as the source for this report. Opens the in-place part selector for editing.
Units	Units that are used for specifying the quantity.

Char Burn Out Report Expert Properties

Representation	Selects a representation. Additional selections appear as representations that are added to the simulation.
Smooth Values	When this property is activated, the report is calculated using node (interpolated) values of the chosen quantity rather than cell values.

Expression Report

The expression report depends on a combination of reports, except for system reports. It works with the field function generated for each report you want to use.

This report is represented in the simulation tree by a node, which has its own properties but is subject to the report node's pop-up menu.

You can post-process scalar values for individual parts, in addition to the overall report results. See Accessing Report Results on a Per-Part Basis.

For example, suppose that you want to plot the pressure drop between two boundaries, such as the inlet and outlet of a pipe. Assuming you created a report of surface average pressure for each boundary, you can directly create a report of those reports by entering the following function names in the *Definition* property:

```
$AveragePressure1Report - $AveragePressure2Report
```

The field functions that can be used to define expressions cannot be a function of space; they have to evaluate to a single number as the expressions are constants in space. Variables such as `$Time`, `$TimeStep` and `$Iteration`, as well as the field functions associated with reports, are valid. Variables such as `$Temperature` or `$Pressure` have different values at every different point in space and cannot be used in expressions.

Some expression reports return values that wrap-around as the solution proceeds. For example, if you have a report that returns the rotation angle of a fan, then you know that the angle must always fall between zero and 2π. If you have an update event, such as a delta or waypoint event, that depends on the difference between two successive values of this report, the event must account for the periodic nature of the reported values. In the following diagram, the difference between values **a** and **b** is $\mathbf{d}_1 + \mathbf{d}_2$ rather than $\mathbf{d}_0$.

Use the *Periodicity* property to identify reports whose values wrap-around. When calculating the difference between two values, STAR-CCM+ chooses the minimum of $\mathbf{d}_0$ and $\mathbf{d}_1 + \mathbf{d}_2 = \mathrm{M} - \mathbf{b} + \mathbf{a} - \mathrm{m}$.

Expression Report Properties

Periodicity	When set to **Periodic**, this property tells related update events that values from this report wrap around when they reach a certain value. This behavior must be respected when calculating the difference between successive reported values. Provide the minimum and maximum wrap-around values in the **Periodic Range** sub-node.
Dimensions	The physical dimensions of this report, which you can customize.
Definition	The formulation of this report using the function name of the STAR-CCM+ field functions corresponding to the reports selected. Entering an invalid definition causes an `Unable to compile` error message.
Units	Units that are used for specifying the quantity.

Periodic Range Properties

Periodic Maximum	The highest value expected for this expression report.
Periodic Minimum	The lowest value expected for this expression report.

Understanding Report Errors

STAR-CCM+ reports generate error messages when they lack sufficient data or when something else is wrong.

Reports notify you with any of four types of error messages, described in the following table.

Error Text	Meaning
`no data`	No data is available for the given part or report. For example, this error occurs for a report on a plane section that cuts no geometry.
`invalid part`	The given part is not valid for use in this context. For example, this error occurs for a surface integral report that is applied to a three-dimensional region.
`unable to compute`	STAR-CCM+ cannot compute a required result. For example, this error occurs if a required field function is not available on an input part of a report.
`incomplete computation`	STAR-CCM+ computes part of the requested result, but errors prevent computing all of it. For example, this error occurs when you run a report on more than one part; one part produces one of the errors that are listed above, but another part is processed successfully.

Concise versions of the messages appear in the *Output* window. The following example shows the output from a Surface Standard Deviation report that was run with an inappropriate input part.

```
        Surface Standard Deviation of Temperature on Volume Mesh

Part                                  Value (K)            Errors
------------------------------        -------------
-------------------------------
Threshold                              3.343456e+02        no data,
invalid part
                                      -------------
-------------------------------
                        Total:         0.000000e+00        no data,
incomplete computation
```

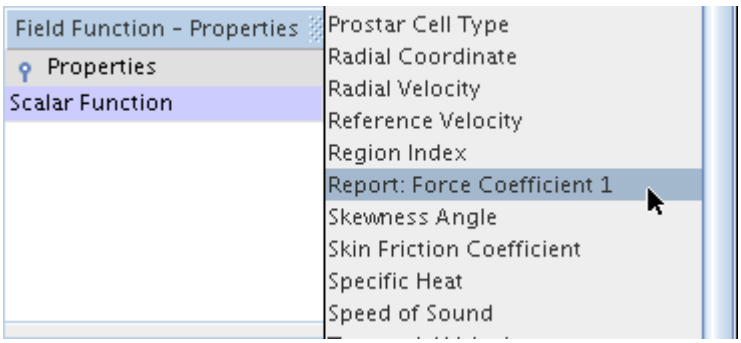
For detailed information, right-click the report and select **Error Log**. Details appear in the *Output* window. The following example is the error log for the Surface Standard Deviation report shown previously:

```
Surface Standard Deviation 1: no data
  Threshold: no data, invalid part
    OutputPiece of Threshold from Region 1: no data
```

Using Reports as Field Functions

When you create a report, a field function corresponding to that report is added automatically.

For example, when you add a report named `Force Coefficient 1`, a drop-down list with field function selections, wherever it is used, includes an item named `Report: Force Coefficient 1`.



Reporting Additional Field Functions at Maximum or Minimum

When you use collocated field functions, you can get the value of other field functions at the location of the minimum or maximum of one.

This feature has a variety of uses. For example, it allows you to get the location of the maximum pressure in the computational domain by adding the components of the Position field function as collocated field functions, or to get the concentrations of chemical species at the location of maximum temperature.

If you use this optional feature, the report displays values for all the specified field functions at the minimum or maximum of each part, and over all parts.

To add collocated field functions in the UI:

1. Create a Maximum or Minimum report.
2. Specify the primary field function in the *Field Function* property.
3. Select the additional field functions in the *Collocated Field Functions* property.

   Note: For collocated field functions to be available on most parts, a Position field function must be among the field functions available for post-processing. Exceptions are phase interactions, which require the Contact Centroid, and tracks, which require the Track Coordinate.

4. Specify one or more parts for the report.
5. To view the values, run the report.

   The primary and collocated values for individual parts and over all parts appear in the *Output* window.
6. If collocated field function data is unavailable, STAR-CCM+ generates an error message in the report output. To investigate the error, right-click the node of the report and select **Error Log**.

Alternatively, add collocated field functions through the macro API:

7. Use the following methods for a Minimum report:
   - `MinReport.getCollocatedValue(FieldFunction)`—gives the value of the specified collocated field function at the global minimum of all input parts of the report.
   - `MinReport.getCollocatedValue(FieldFunction, ClientServerObject)`—gives the collocated field function value at the local minimum on the specified part (the `ClientServerObject` argument).

   These methods return a double object, and return `null` if the specified field function is not a collocated field function of the report, or (for the second method) if the specified `ClientServerObject` is not an input part of the report.

   Analogous methods exist for the Maximum report.

If you create a monitor from a Maximum or Minimum report, the monitor has options in its *Value Type* property for specifying collocated field functions and parts for monitoring:

- **Collocated Value**—activates the **Collocated Field Function** sub-node to let you select any of the collocated field functions of the source report.
- **Part Collocated Value**—activates the **Collocated Field Function** and **Report Part** sub-nodes to let you select collocated field functions on particular parts from the source report.

Using Coordinate Systems with Force and Moment Reports

You can link a report to a Cartesian coordinate system, and thus simplify your direction calculation by changing the frame of reference.

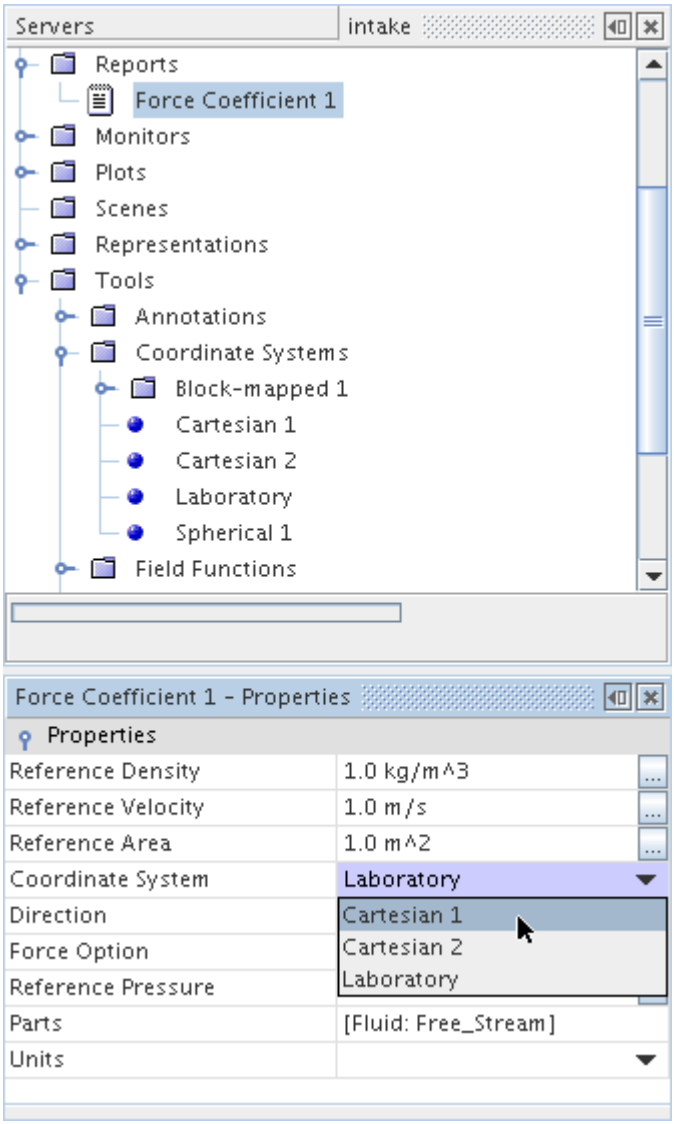
This can be useful if, for example, you are modeling a certain part that is oriented differently from standard (laboratory) coordinates. Additionally, you can change the coordinates for more than one report by editing a single object, rather than modifying each report individually.

The following types of reports use Cartesian coordinate systems:

- Force
- Force coefficient
- Moment
- Moment coefficient
- Reaction force
- Reaction moment

Other than the default laboratory coordinate system, which is suitable for simpler problems, this feature uses the Cartesian coordinate system only. Even if you have various coordinate systems in your simulation, the report's *Coordinate System* property uses a filter that recognizes only the Cartesian coordinate systems in the simulation (in addition to the laboratory coordinate system).

To use a Cartesian coordinate system in a report, first add it to the simulation. Then select the node of the report and choose that coordinate system from the drop-down list of the *Coordinate System* property.


Servers
intake
Reports
Force Coefficient 1
Monitors
Plots
Scenes
Representations
Tools
Annotations
Coordinate Systems
Block-mapped 1
Cartesian 1
Cartesian 2
Laboratory
Spherical 1
Field Functions
Force Coefficient 1 - Properties
Properties
Reference Density
1.0 kg/m^3
Reference Velocity
1.0 m/s
Reference Area
1.0 m^2
Coordinate System
Laboratory
Direction
Cartesian 1
Force Option
Cartesian 2
Reference Pressure
Laboratory
Parts
[Fluid: Free_Stream]
Units

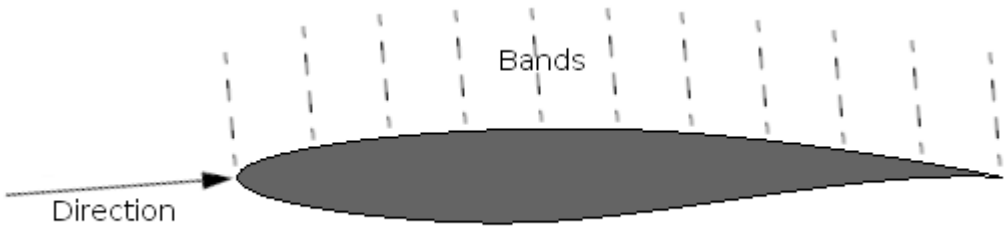
Using Bands with Force and Moment Reports

With advanced Java techniques, you can add flexibility in calculating force and moment by dividing the object into bands.

The following types of reports use bands:

- Force
- Force coefficient
- Force (harmonic)
- Moment
- Moment coefficient

STAR-CCM+ starts creating bands by locating the beginning and end of the object along the direction that you specify. (Specify the direction with the *Direction* property of the force, force (harmonic) and force coefficient reports, or the *Axis* property of the moment and moment coefficient reports.)



The distance between these two locations is divided based on your number of bands. Then values are calculated for each band.

Set this up as follows:

- Complete your entries in the report's properties and its *Number of Bands* expert property.
- Access the Java API. Use the following methods from the Method Summary of the `ForceAndMomentBaseReport` class:
  - `getBandPositions()`
  - `getBandPressureValues()`
  - `getBandShearValues()`
  - `getBandValues()`

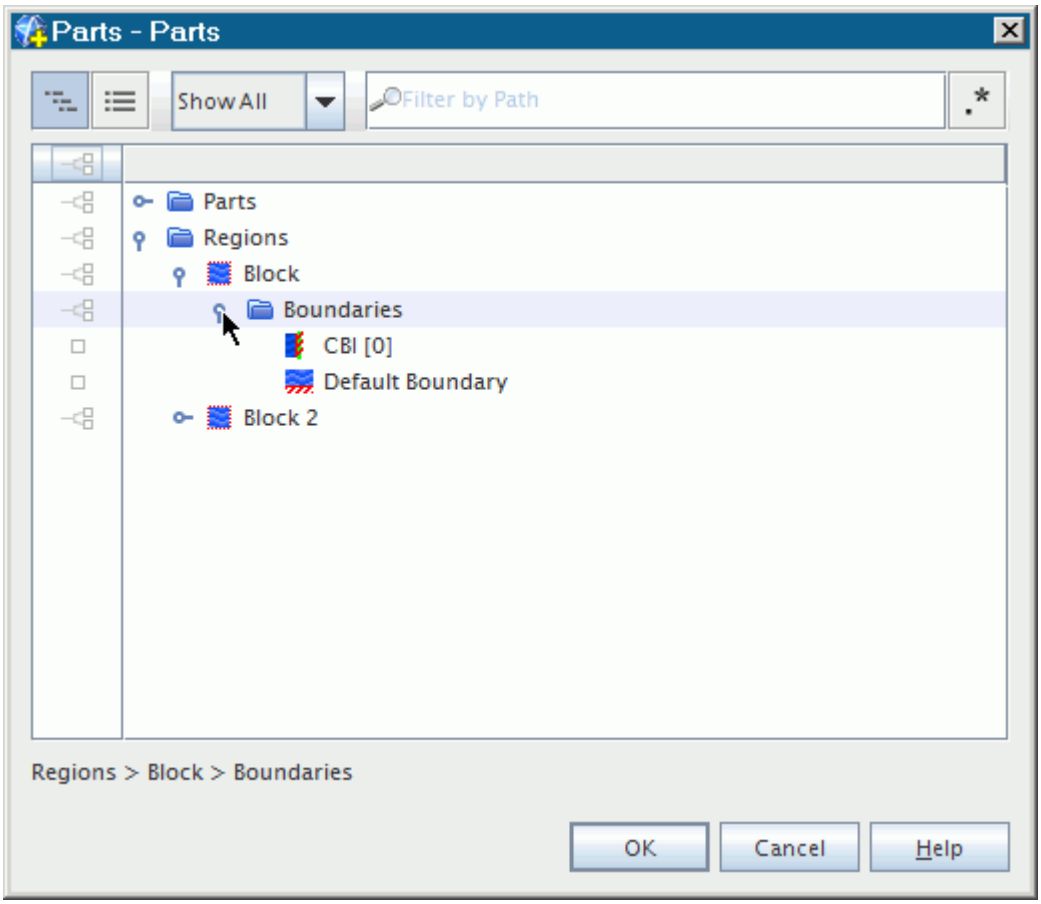
Working with Parts-Based Interfaces in Reports

You can report results for *parts-based interfaces* -- the areas where the surfaces of parts come into contact with each other, or the remainders of those surfaces. Using parts-based interfaces improves performance for simulations that have many parts.

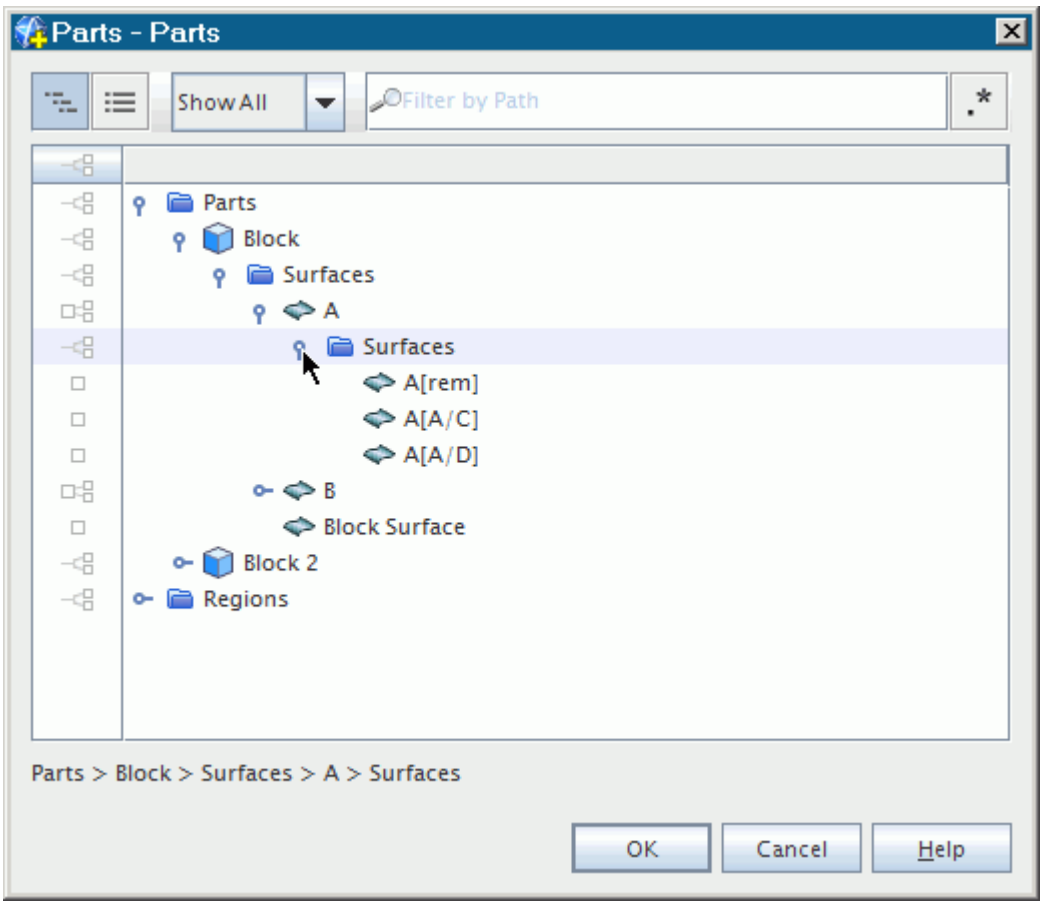
You can report on parts-based interfaces in one of two ways:

- Accessing the whole interface boundaries (the complete intersected portion of the original parent boundaries)
- Accessing individually the intersected and remaining portions of part surfaces in the contacts of the interface

To access a whole interface boundary (using the object selector), select the respective interface boundary for the associated region in the **Regions** node. This functionality works the same as for boundary-based interfaces.



To access the intersected and remaining portions of part surfaces that are in contact, expand the nodes of the intersected parent part surfaces and select the desired subsurfaces. A subsurface exists for: (a) each contact in which the parent part surface participates, and (b) the remaining portion of the parent part surface that is not intersected.

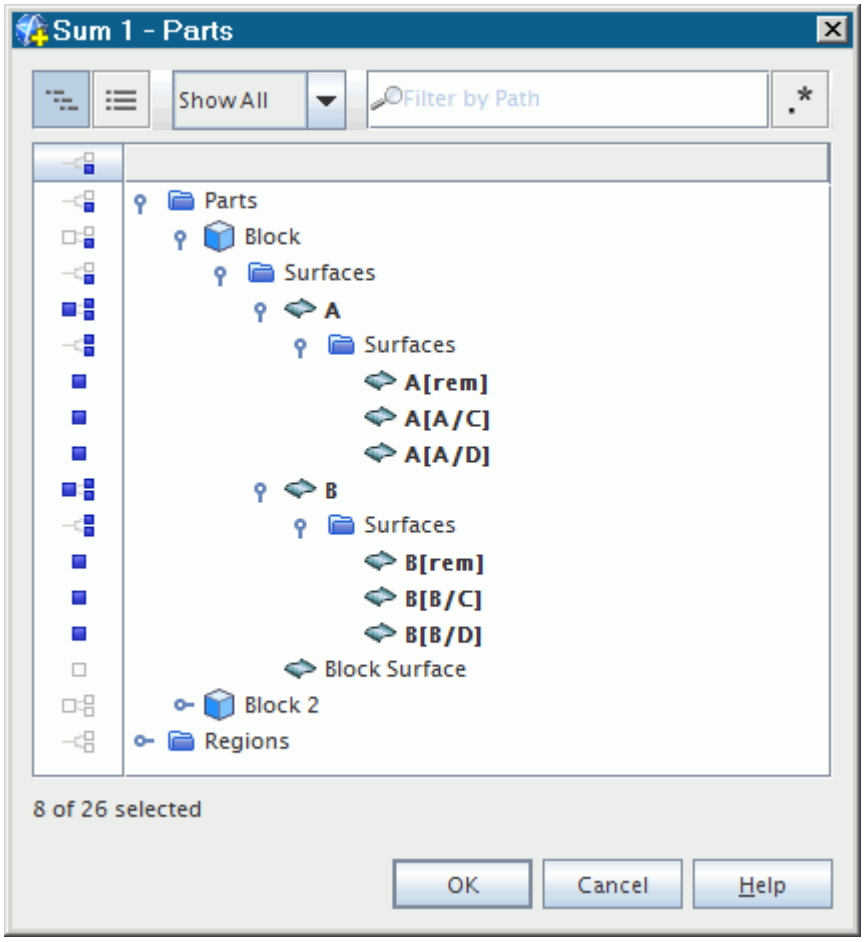


Example of Reporting on Parts-Based Interfaces

This example uses the intersection of two blocks to illustrate how you can report on contacts between surfaces. One block has surfaces A and B, the other has surfaces C and D. Each part surface participates in two contacts (one with each of the two part surfaces on the other side), yielding four contacts in the interface: A/C, A/D, B/C, B/D.

Three separate reports are defined, each with the Area Magnitude field function, and with the following respective selected parts:

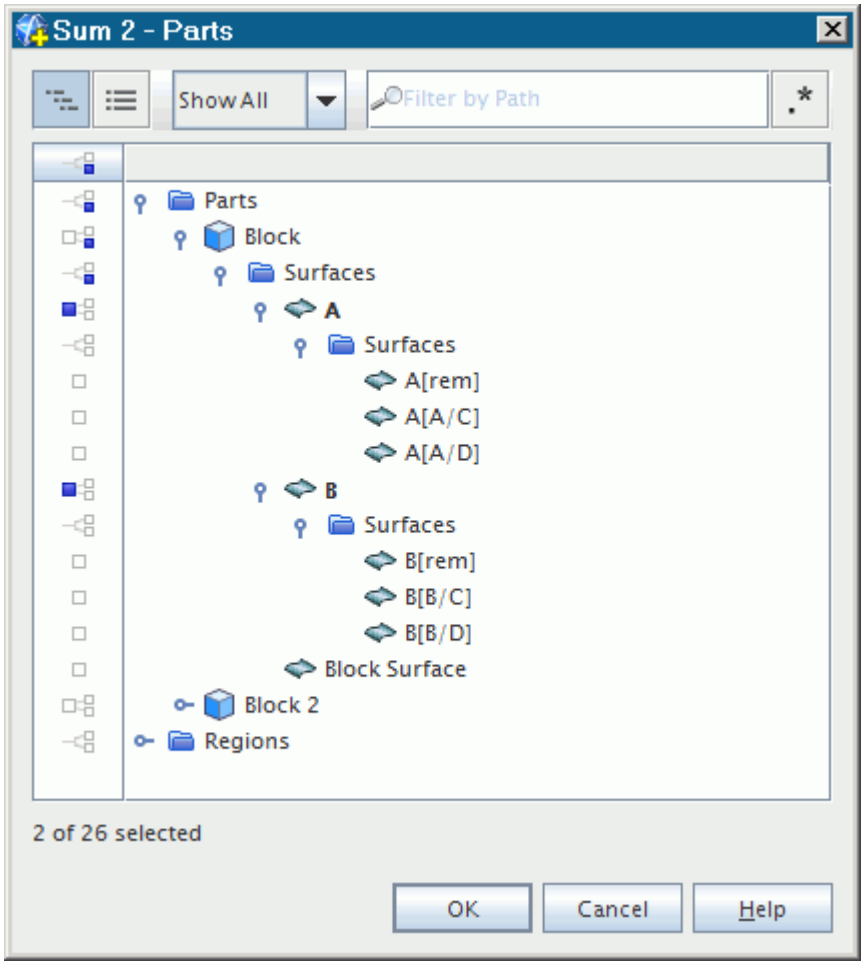
1. Selection of aggregate Part Surfaces **A** and **B** and all of their part subsurfaces (areas of contact and remainders):



The report **Sum 1** computes the results on the aggregate part surfaces as the accumulation of the results from their subsurfaces.

```
        Sum of Area: Magnitude on Volume Mesh

Part                                Value (m^2)
-----------------------------       -------------
Block.A                              5.000000e-01
Block.A[A/C]                         2.437500e-01
Block.A[A/D]                         2.000000e-01
Block.A[rem]                         5.625000e-02
Block.B                              5.000001e-01
Block.B[B/C]                         1.250000e-03
Block.B[B/D]                         1.500000e-01
Block.B[rem]                         3.487500e-01
                                    -------------
                        Total:       2.000000e+00
```

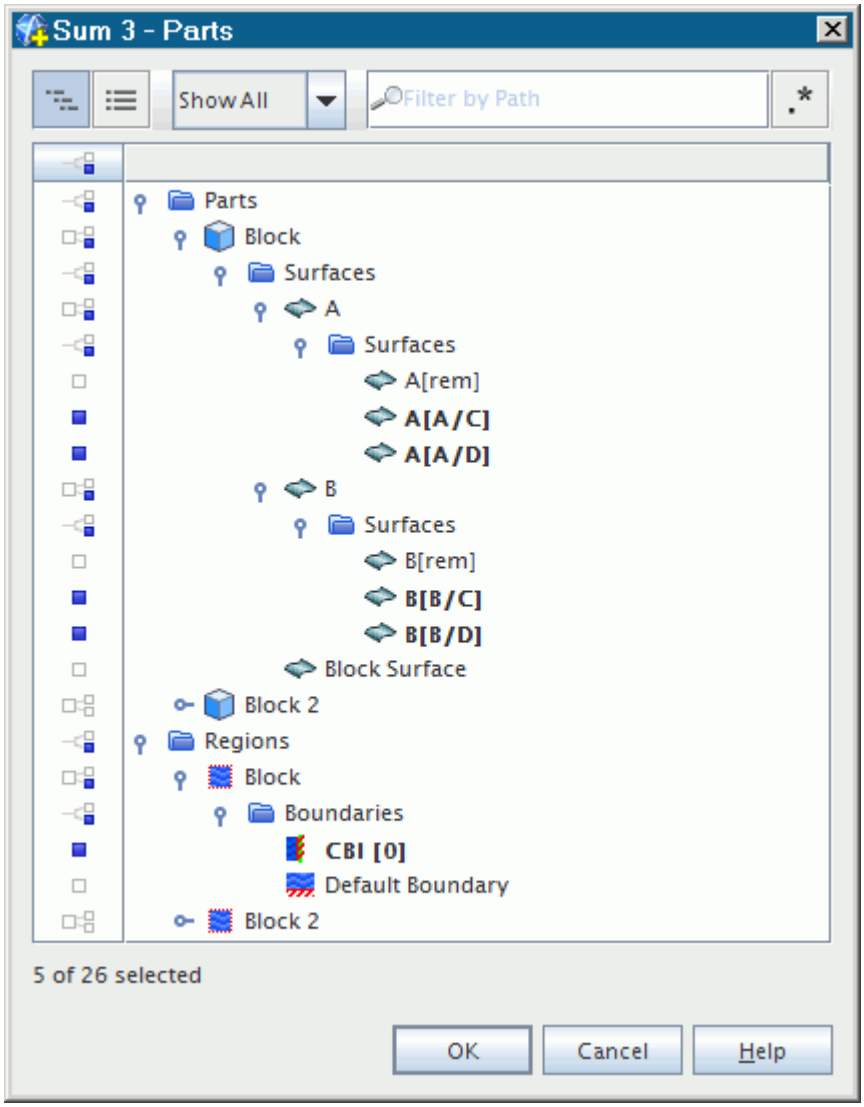
2. Selection of only aggregate Part Surfaces **A** and **B**:



The report **Sum 2** computes the results on the aggregate part surfaces even if none of their subsurfaces are explicitly selected.

```
        Sum of Area: Magnitude on Volume Mesh

Part                                Value (m^2)
-----------------------------       -------------
Block.A                              5.000000e-01
Block.B                              5.000001e-01
                                    -------------
                        Total:       1.000000e+00
```

3. Selection of the following:
   a. The interface boundary **CBI [0]**, on the side of the interface that is part of the region **Block**
   b. All four contact surfaces.



The report **Sum 3** computes results for the interface boundary **CBI [0]** that are consistent with the results from the four corresponding contact surfaces:

0.595 = 0.2 + 0.24375 + 0.15 + 0.00125.

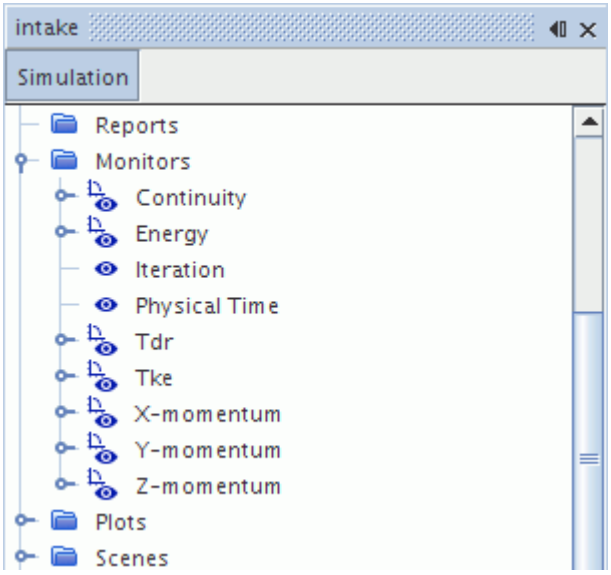
Reports Bibliography

[190] *Weltens, H., Bressler, H., Terres, F., Neumaier, H. et al., "Optimisation of Catalytic Converter Gas Flow Distribution by CFD Prediction," SAE Technical Paper 930780, 1993.*

Monitoring the Solution

Monitors sample, save, and plot solution data.

The **Monitors** manager node, which has its own properties and pop-up menu, contains all of the monitor nodes in the simulation tree.



Residual monitors are a specific kind of monitor created automatically.

This section explains the following about STAR-CCM+ monitors:

Contents:

What Is a Monitor?
Customizing Monitors
Using Report-Based Monitors
Using Field Function Monitors
Properties of the Monitors Manager
Pop-Up Menu of the Monitors Manager

What Is a Monitor?

Monitors enable summary information from the simulation to be sampled and saved during the solution.

This information can then be displayed in an XY plot while the solution is progressing, or post-processed after iteration or unsteady time-stepping is complete. They are useful for watching the behavior of numerical or engineering quantities as the solution evolves, and can be used to define stopping criteria that can stop the solution iteration.

Specific monitors, represented in the simulation tree by nodes that have properties and pop-up menus, may be added one of two ways:

- Automatically, based on the residuals computed by solvers in the simulation (which, in turn, are driven by the model selection that you made in the continua)
- Manually from existing reports. For these monitors, you can extract scalar values for a particular part that is included in a scalar report, in addition to the overall report results. See Extracting Report Results on a Per-Part Basis.

The residual monitors that are added by solvers are listed in the solvers section of the relevant modeling chapter.

Contents:

Monitor Properties
Monitor Pop-Up Menu

Monitor Properties

Properties are described for residual monitors and report monitors.

Because of the similarity, the information for both types is documented here, and differences between the two monitor types are noted where relevant.

Report	This property appears only if the monitor was created from a report. It selects the source report.	
Active	This property appears only if this is a residual monitor (read-only).	
	Activated	The monitor is active in the current simulation.
	Deactivated	The monitor is inactive. This would happen if you changed models in the middle of the simulation and you had not deleted any monitors specific to the previous models.
Enabled	Controls whether to update the particular monitor.	
	Activated	Allows updates of the monitor.

	Deactivated	Does not allow updates of the monitor.
Value Type	Controls whether to access overall report results or a particular part for a report monitor. See Extracting Report Results on a Per-Part Basis. For a monitor that is generated from a Minimum or Maximum report, this property includes additional options for collocation of field functions. See Reporting Additional Field Functions at Maximum or Minimum.	
	Total Value	Selects the overall report results.
	Part Value	Allows access to values for a particular part. Activates the **Report Part** sub-node which lets you select that part in the *Parts* property.
Trigger	Defines what action will trigger collection of a sample. For report monitors, the default setting depends on whether you are using the Steady model (iterations) or Implicit Unsteady model (time-steps).	
	Iteration	A sample will be taken every `N` iterations, where `N` is defined by the *Frequency* (default for residual monitors).
	Time-Step	A sample will be taken every `N` time-steps, where `N` is defined by the **Frequency**.
	Inactive	No sample is taken.
Maximum Plot Samples	Limits the number of sample points used in the monitor plot.	
Frequency	Determines how often the sample will be taken, based on the setting in *Trigger*. Samples are taken when the number of samples is evenly divisible by the frequency.	
Normalization Option	Defines how the monitor data will be normalized.	
	Auto	Lets STAR-CCM+ do the normalization as derived in the Automatic Normalization node that is added as a sub-node when you choose this option.
	Manual	Lets you define a specific normalization factor in the Manual Normalization node that is added as a sub-node when you select this option.

Monitor Pop-Up Menu

This pop-up menu is the same for residual monitors and report monitors.

Create Plot from Monitor	Creates a monitor plot by adding a node to the **Plots** node named `Monitor Plot N` where `N` is the total number of monitor plot nodes.
Create Stopping Criterion from Monitor	Creates a monitor-based stopping criterion by adding a node to the **Stopping Criteria** node named `Aaa Criterion` where `Aaa` is the name of this monitor node.
Export...	Exports the data to a `.csv` file. The exported data retains the preferred units.

Customizing Monitors

The way in which monitors sample data and send data to an XY plot can be customized to suit the particular quantity being monitored.

This section describes how to change the following:

Contents:

Changing the Trigger Event and Sampling Frequency
Setting Maximum Plot Samples
Normalizing the Monitor Data

Changing the Trigger Event and Sampling Frequency

Trigger events control how often the monitor samples data.

See Trigger Types.

The frequency of trigger events in which a monitor samples data can be modified from the sub-node properties. See Trigger Properties.

For more information on selecting the trigger type and defining the update frequency, refer to Setting the Trigger for an Update.

If you wish to average results per iteration, it is recommended that you first read the section on choosing between steady and unsteady models.

Note: The *Trigger* property of a monitor does not control the frequency at which the data is plotted. It specifies the frequency at which the monitor collects the data. To control the frequency at which the data is queried and plotted, use the Update feature of the plot node itself. If you want to plot data against physical time, make sure that the same **Physical Time** monitor exists throughout the solution. Otherwise the plot data appears incomplete. If you are running a steady solution but expect to need a plot with physical time, start by running the unsteady solver then switch to the steady one; this ensures that Physical Time exists in the simulation.

Setting Maximum Plot Samples

The *Maximum Plot Samples* property sets an upper limit on the number of points that will be displayed in the monitor plot.

When this number of points has been sampled, STAR-CCM+ will begin to reuse plot storage for points obtained earlier in the simulation. This will result in fewer data points being allocated to earlier parts of the simulation. This allocation applies to the plot only; all of the monitor's solution data points are retained in the simulation file.

In a long simulation STAR-CCM+ might generate many thousands of points, for example one per iteration. After a certain number of points, however, additional ones do not provide any visual improvement to the graph

and may result in performance degradation. Limiting the number of points may improve performance in drawing the graph (specifically graphics performance, client-server data exchange and memory consumption).

The monitor tends to retain more data toward the end of the run when the number of samples has exceeded the maximum specified in this property.

Normalizing the Monitor Data

The data in a monitor can be normalized one of two ways: manually or automatically.

Normalizing Manually

To normalize the data using a specified value, click the monitor node and select **Manual** from the pull-down menu under the *Normalization Option* property in the *Properties* window.

The **Manual Normalization** node, with its own properties, will be added as a sub-node of the monitor node.

The normalization factor can be specified by expanding the monitor node, clicking on the **Manual Normalization** child node, and then modifying the value in the *Normalization Factor* property in the *Properties* window.

Manual Normalization Properties

Normalization Factor	The normalization factor for the monitored quantity.

Normalizing Automatically

To normalize the data automatically, click the monitor node and select **Auto** from the pull-down menu under the *Normalization Option* property in the *Properties* window.

The **Automatic Normalization** node, with its own properties, will be added as a sub-node of the monitor node.

You must enter the number of samples from the start from which the monitor will search for a maximum normalization factor. Expand the monitor node, click the **Automatic Normalization**child node and modify the *Normalization Number* property in the *Properties* window. The normalization factor found is displayed under the read-only *Normalization Factor* property.

Automatic Normalization Properties

Normalization Factor	The normalization factor for the monitored quantity. Computed as the maximum monitor value over the first N iterations or time-step events of the monitor, where N is the *Normalization Number* (read-only).
Normalization Number	The number of iterations or time-step events, from the start of the solution, used to compute the *Normalization Factor*.

Copying Properties of a Normalization Node

When properties are copied from one normalization node to another, all of the properties are applied so that the destination normalization node matches the source one. However, it is not possible to copy properties between different types of normalization nodes.

Using Report-Based Monitors

This section explains:

Contents:

What Is a Report-Based Monitor?
Extracting Report Results on a Per-Part Basis

See Also:

Using Maximum and Minimum Reports

What Is a Report-Based Monitor?

Report-based monitors are useful for monitoring quantities of engineering interest during a simulation. They provide a dynamic record of these quantities during solution iteration.

They may be viewed, either as the solution iterates, or after the solution has converged, using a monitor plot.

Any report can be used as the basis of a monitor. Report-based monitors can be important for judging convergence, and may be used as stopping criteria.

You can incorporate the data from a solution history into a report-based monitor. See Reporting and Monitoring Recorded Solution Data.

Similar to other objects, report-based monitors can be renamed within STAR-CCM+. Report-based monitors can only be deleted if they are not displayed in an XY plot or used to define a stopping criterion.

When a report-based monitor is copied within a simulation, its properties are copied. You can only copy this object between simulation files if the destination simulation has a report of the same name.

Extracting Report Results on a Per-Part Basis

For a report-based monitor, you can extract scalar values for a specific part that is included in a scalar report, in addition to the overall report results.

You can also use an expression report to access parts in a user-defined field function. See Accessing Report Results on a Per-Part Basis.

The following report types do not yet support per-part value access:

- Boundary Parameter Sensitivity
- Element Count
- Frontal Area
- Generalized Vibratory Force
- Heat Exchanger (Single Stream and Dual Stream)
- Reaction Force (Finite Element and Finite Volume)

- Reaction Moment (Finite Volume only)

To specify a part of the report results using the object tree:

1. Expand the **Monitors** node, and select the node of the report monitor.
2. Set *Value Type* to **Part Value**.

   The **Report Part** sub-node appears.
3. Select the **Report Part** sub-node and specify the report part in the *Parts* property.

   Note: You cannot select more than one part at a time for this property.

Using Field Function Monitors

This section explains:

Contents:

What Is a Field Function Monitor?
Using a Sliding Sample Window
Specifying When Data Collection Stops

What Is a Field Function Monitor?

There are six varieties of field function monitor. During a simulation run, they collect data samples for a selected part at a frequency that you specify.

You can control the sample count by specifying the start time and stop time, and by using a sliding sample window.

- Field co-variance -- computes the co-variance of two field functions.
- Field max -- obtains the highest value of a field function for the selected part.
- Field mean -- obtains the average value of a field function for the selected part.
- Field min -- obtains the lowest value of a field function for the selected part.
- Field sum -- provides the total value of a field function for the selected part.
- Field variance -- computes the variance of a field function.

These monitor nodes generally share common properties and a pop-up menu.

Once you set their *Field Function* and *Parts* properties, these monitors can be displayed as scalar quantities just like other field functions.

Caution: Data mappers can cause inconsistent reporting if you use them with field function monitors or with update events other than iteration, time-step, or delta time.

Field Function Monitor Properties

Publish Mean	For the Field Variance and Field Co-Variance monitors, determines whether STAR-CCM+ will generate a separate field function of the mean of the function being sampled. Two separate properties exist for the Field Co-Variance monitor's field function selections: *Publish Mean 1* and *Publish Mean 2*.	
	Activated	Enables this feature.
	Deactivated	Leaves this feature disabled.
Field Function	One or more field functions which the monitor is to sample in order to perform its calculations. Two separate properties exist for the Field Co-	

	Variance monitor's field function selections: *FieldFunction 1* and *FieldFunction 2*.	
Trigger	Provides choices for triggering the action:	
	Iteration	Calculates based on the specified number of iterations in the *Frequency* property. If you wish to average results per iteration, it is recommended that you first read the section on choosing between steady and unsteady models.
	Time-Step	Calculates based on the specified number of time-steps in the *Frequency* property (default).
	Inactive	No sample is taken.
Parts	The model parts over which the field function monitor collects samples. Opens the in-place part selector for editing.	
Sample Count	The number of data samples collected for each cell within the monitor (read-only).	
Enable sliding sample window	Calculates a set of the most recent samples, with the number of samples in that set specified by the user. Activates the Sliding Sample Window Size node.	
	Activated	Enables this feature.
	Deactivated	Leaves this feature disabled.
Title	The name of the field function which contains the values computed by the field function monitor (read-only).	
Collection Start	The solution step at which to begin collecting data. Samples will be collected at time-steps or iterations after this start time, based on the *Frequency* property.	
Enable collection stop	Specifies the final sample time or iteration after which to stop collecting data. Activates the Stop Specification node.	
	Activated	Enables this feature.
	Deactivated	Leaves this feature disabled.
Frequency	Determines how often the sample will be taken, based on the setting in *Trigger*. Samples are taken when the number of samples is evenly divisible by the frequency.	

Field Function Monitor Pop-Up Menu

Reset Monitor Data	Clears all samples and accumulated data from the field function monitor.

Contents:

What Is Field Co-Variance?
What Is Field Max?
What Is Field Mean?
What Is Field Min?
What Is Field Sum?
What Is Field Variance?

What Is Field Co-Variance?

The Field Co-Variance is the statistical co-variance of two field function values as the solution progresses, computed for each cell or boundary face in the selected parts.

Two separate properties exist for the Field Co-Variance monitor's field function selections: *FieldFunction 1* and *FieldFunction 2*.

The co-variance is based on data samples collected at a user-specified frequency for each cell of the parts being monitored. The sample co-variance, $\mathrm{Cov}(x, y)$, is computed for each cell as:

$$\mathrm{Cov}\left(x, y\right) = \frac{1}{N-1}\left[\sum_{i=1}^{N} x_i y_i - \frac{1}{N}\left(\sum_{i=1}^{N} x_i\right)\left(\sum_{i=1}^{N} y_i\right)\right]$$

where:

- $x_i$is the Field Function 1 value at collection step i.
- y_i is the Field Function 2 value at collection step i.
- N is the number of time-steps or iterations for which data has been collected.

What Is Field Max?

The Field Max is the maximum value of the specified field function which occurred within each cell or boundary face in the selected parts.

The field function is sampled at a user-specified frequency for each cell of the parts being monitored and the maximum value retained as:

$$\mathrm{Cell}_{\max} = \max\left(\mathrm{Cell}_{\max}, \mathrm{Cell}_{current}\right)$$

where $\mathrm{Cell}_{current}$ indicates the field function value at the current iteration or time-step.

What Is Field Mean?

The Field Mean is the average of a field function value as the solution progresses, computed for each cell or boundary face in the selected parts.

This arithmetic mean is based on data samples collected at a user-specified frequency for each cell of the parts being monitored. The arithmetic mean, $\overline{x}$, is computed for each cell as:

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

where x_i is the field function value at collection step i, and N is the number of time-steps or iterations for which data has been collected.

What Is Field Min?

The Field Min is the minimum value of the specified field function which occurred within each cell or boundary face in the selected parts.

The field function is sampled at a user-specified frequency for each cell of the parts being monitored and the minimum value retained as:

$$\mathrm{Cell}_{\min} = \min\left(\mathrm{Cell}_{\min}, \mathrm{Cell}_{current}\right)$$

where $\mathrm{Cell}_{current}$ indicates the field function value at the current iteration or time-step.

What Is Field Sum?

The Field Sum is the total of a field function value as the solution progresses, computed for each cell or boundary face in the selected parts.

This sum is composed of data samples collected at a user-specified frequency for each cell of the parts being monitored. The total, x_N , is computed for each cell as:

$$x_N = \sum_{i=1}^{N} x_i$$

where x_i is the field function value at collection step i, and N is the number of time-steps or iterations for which data has been collected.

What Is Field Variance?

The Field Variance is the sample variance of a field function value as the solution progresses, computed for each cell or boundary face in the selected parts.

This statistical variance is based on data samples collected at a user-specified frequency for each cell of the parts being monitored. The sample variance, $s^2(x)$, is computed for each cell as:

$$s^2\left(x\right) = \frac{1}{N-1}\left[\sum_{i=1}^{N} x_i^2 - \frac{1}{N}\left(\sum_{i=1}^{N} x_i\right)^2\right]$$

where x_i is the field function value at collection step i , and N is the number of time-steps or iterations for which data has been collected.

Using a Sliding Sample Window

The sliding sample window is the N most recent samples of data, (based on iteration or time-step depending on setup), during the simulation.

To evaluate this set of samples for a field function monitor, you need to activate the *Enable Sliding Sample Window* property of that monitor first.

This adds the **Sliding Window** node to the object tree under the parent field function monitor node. To specify the number of samples (the “window”), select this node and set the *Sliding Sample Window Size* property to the desired value in the properties window.

Sliding Window Properties

Use this property to specify the window size.

Sliding Sample Window Size	Specifies the number of most recent samples to keep. If the monitor Trigger property is set to Iteration, this property sets the number of most recent iterations. If the trigger property is set to Time-step, this property sets the number of most recent time-steps.

Specifying When Data Collection Stops

You can set a time or iteration at which the monitor stops collecting data. This can let you get results for specific intervals within the simulation run.

For instance, if you are simulating an impeller and you want to obtain the mean pressures for seven revolutions lasting 0.08 seconds each, you would create seven field mean monitors. The monitors would have settings that increase by 0.08, for example:

- **Field Mean 1** -- collection start = 0 and collection stop = 0.08
- **Field Mean 2** -- collection start = 0.08 and collection stop = 0.16
- **Field Mean 7** -- collection start = 0.48 and collection stop = 0.56

To apply this setting to a particular field function monitor, first you need to select the trigger type (time-step or iteration) in that monitor’s *Trigger* property.

Next, make an entry in the *Collection Start* property, and activate the *Enable collection stop* property.

This adds the **Stop Specification** node to the object tree under the parent field function monitor node. To specify when data collection stops, select this node and set the *Collection stop* property to the desired value in the properties window.

The values you enter in the *Collection stop* property and the parent node's *Collection Start* property will be for a time-step or iteration, depending on what you selected in the parent node's *Trigger* property.

Stop Specification Properties

Use this property to specify when the data collection stops during the simulation run.

Collection stop	Specifies the number of time-steps or iterations to collect data.

Properties of the Monitors Manager

These properties allow you to control how the monitor data appears in the output.

Monitors to Print	Selects the monitors to print in the output using a standard selection dialog. By default all residual and report monitors in the simulation are selected.
Output Direction	Selects a horizontal or vertical direction for displaying the output.
Heading Print Frequency	Exposes the number of iterations to print between the header lines in the console output. The default value is 10. Only entries greater than zero are valid. You can enter a very large number to get the header printed only once, at the beginning.

Pop-Up Menu of the Monitors Manager

These are the pop-up menu items available when you right-click the **Monitors** manager node.

New Monitor	Creates a new field function monitor of the type you select in the submenu.	
	Field CoVariance	Creates a field co-variance monitor.
	Field Max	Creates a field max monitor.
	Field Mean	Creates a field mean monitor.
	Field Min	Creates a field min monitor.
	Field Sum	Creates a field sum monitor.
	Field Variance	Creates a field variance monitor.
New Group	Facilitates the grouping of objects by creating a new group node within the manager node.	
Group By	Allows you to group the objects according to any of the properties of the objects in the manager node, listed in a submenu of this menu item.	
Ungroup	Reverses the grouping process.	

Plotting Results

The **Plots** node, which has a pop-up menu, is the manager object for all monitor plots and XY plots that have been created in the simulation.

It exists even when empty to allow creation of the first plot, but will typically contain the **Residuals** plot, which is created automatically once you start iterating.

The manager object can contain any number of plots up to the computer's memory limits and the plots are saved in the database when the simulation is saved.

The plotting features in STAR-CCM+ allow you to create three kinds of two-dimensional plots:

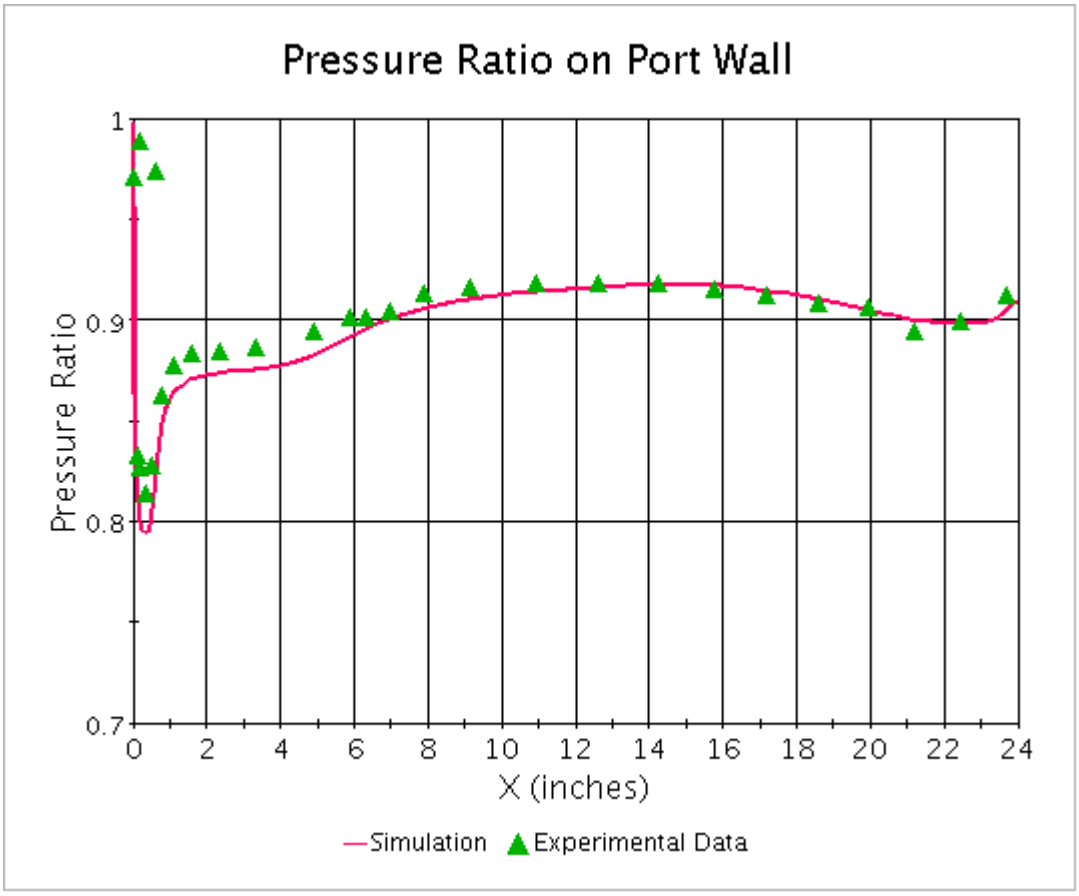
- Monitor plots that can display data from monitors, data set functions and/or tables. STAR-CCM+ uses two varieties of monitor plots:
  - Monitor plots are user-created plots.
  - The residual plot is generated and modified automatically by the system.
- XY plots that display solution data from the simulation and/or tables
- Histogram plots that display solution data from the simulation and/or tables

As with many other simulation objects, plots can be deleted, renamed and copied, and can be extensively customized. If the plot is open in the *Graphics* window while you are customizing properties, the relevant object on the plot display will change as you enter the new value. This is a useful way to become familiar with the various properties. All plots, along with any customization, are saved in the simulation file. You can also customize how an XY plot or histogram plot is updated as the simulation iterates.

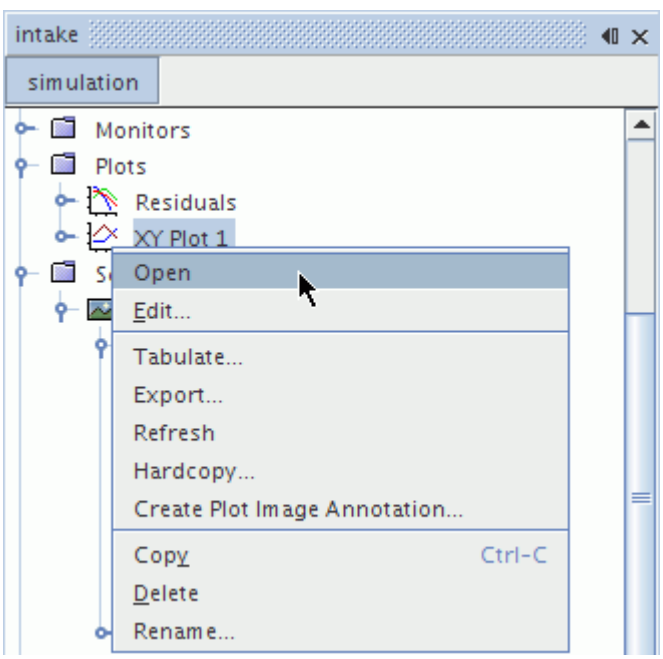
The plot display is highly interactive, having its own toolbar, zoom, pan and pop-up operations as well as being linked to the simulation tree through drag-and-drop operations. These operations include plotting data from a table and plotting data from a data set, which can also be done using the *Properties* window.

The data being plotted can be viewed in a spreadsheet-like dialog or it can be exported to file for manipulation in other programs. The plot can also be printed to file or to a printer.

A sample XY plot produced by STAR-CCM+ is shown below. In this plot, simulation data is shown on the same plot with data from an external file.

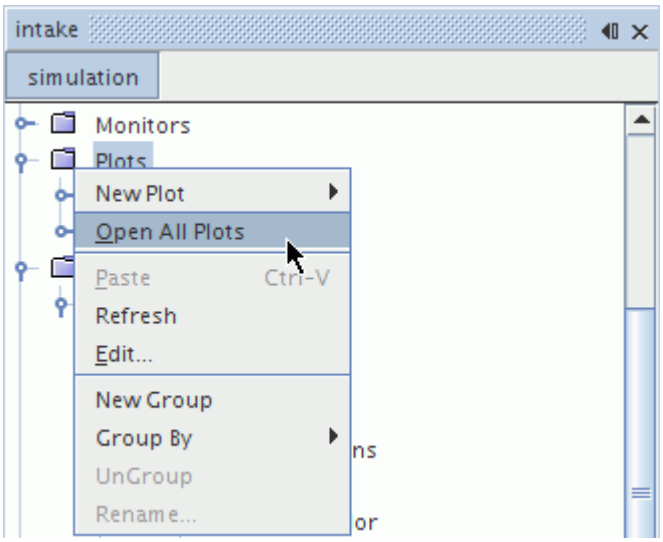


Any type of plot can be opened by double-clicking its node in the simulation tree, or by right-clicking that node and selecting **Open**.



Either operation opens the selected plot to display it in a new document in the *Graphics* window, or brings the document to the foreground if the plot is already open.

You can open all the plots of a simulation at once by right-clicking the **Plots** manager node and selecting **Open All Plots**.



Since there is a lot of similarity between monitor plots and XY plots, parts of this reference section cover the nodes that are specific to monitor plots and XY plots before covering all nodes that are common to both.

Contents:

Plotting Workflow
Working with Monitor Plots
Working with XY Plots
Working with Histogram Plots
Plotting with Multiple Axes
Updating Plots
Deleting Plots
Renaming Plots
Copying Plots
Customizing Plots
Adding and Removing Data in a Plot
What Is a Tabular Data Set?
What Is a Derived Data Set?
Tabulating Data
Exporting Data
Printing and Saving Plot Images
Plot File Formats Reference
Pop-Up Menu of the Plots Manager

See Also:

Working with Displays

Plotting Workflow

Decide on the quantities to plot, and based on that choice, select the plot type. Set up the plot with the appropriate field functions and parts. Decide how to customize the appearance of the plot, and whether to export images for presentation purposes.

1. Decide on the quantities to plot, and select a plot type accordingly:
   - To plot cell- or vertex-based values of a part, create an XY plot. See Creating an XY Plot.
   - To use data distribution, create a histogram plot. See Creating a Histogram Plot.
   - To work with data that is summarized in a report, create a monitor plot. See Monitor Plots.

XY and Histogram Plots

2. Select the parts for which to plot the quantities.
3. Select the scalar values to plot:
   - For an XY plot, see Plotting a Solution Variable.
   - For a histogram plot, see Setting up a Histogram.
4. To work with high concentrations of data in an XY plot, you can use the heatmap style to view the data more clearly. See Using Heatmaps.
5. If applicable, add tabular or reference data to the plot. See Adding and Removing Data in a Plot.

Monitor Plots

6. Create the monitor plot using one of the following techniques:
   - After preparing the relevant report, right-click the report node and select **Create Monitor and Plot from Report**.
   - Right-click the monitor node and select **Create Plot from Monitor**.
   - Right-click on the **Plots** node and select **New Plot** > **Monitor**.
7. If you choose to create an empty monitor plot from the **Plots** node, set it up with the same techniques as for adding tabular or reference data. See Adding and Removing Data in a Plot.
8. Decide whether to normalize the values. See Normalizing the Monitor Data.
9. For better post-processing performance, decide whether to limit sampling. See Setting Maximum Plot Samples.

Finishing the Plot

10. Customize the plot as you see fit. See Customizing Plots.
11. Decide how frequently to update the plot. See Triggers and Update Events.
12. Decide whether to export plot images during the simulation run. See Update and Hardcopy Properties.

You are ready to run the simulation.

Working with Monitor Plots

This section explains the use of plots that display data from monitors.

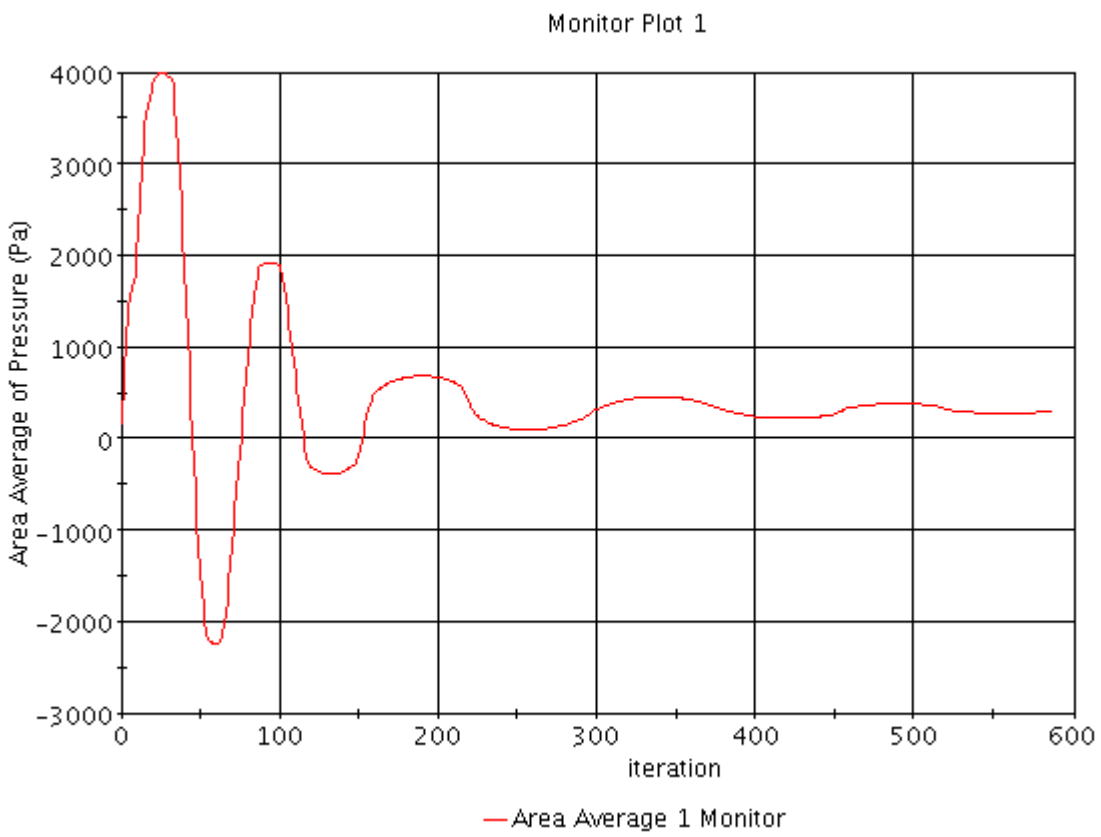
Contents:

What Is a Monitor Plot?
Creating a Monitor Plot
Working with Monitor Data Sets
Normalizing the Plot
Plotting One Monitor against Another
Creating a Custom Residual Plot

What Is a Monitor Plot?

Monitor plots display the values collected by report-based monitors as the solution is advanced.

The monitor plot node has properties and a pop-up menu. An example of a monitor plot is shown below.



You can incorporate the data from a solution history into a monitor plot. See Reporting and Monitoring Recorded Solution Data.

Monitor plots can be renamed and deleted when they are no longer needed. Data from tables and data set functions may be plotted with the monitored data.

Contents:

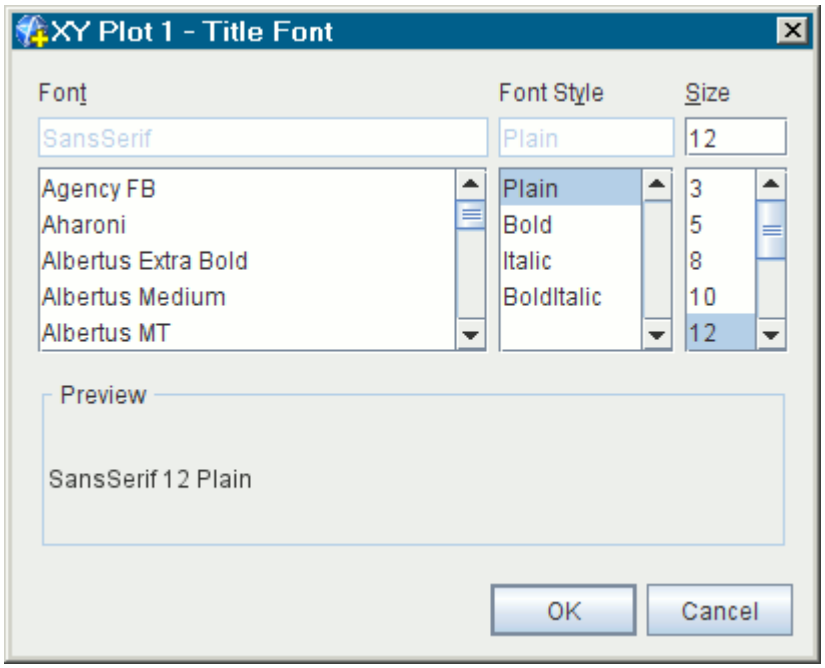
Monitor Plot Properties
Monitor Plot Pop-Up Menu

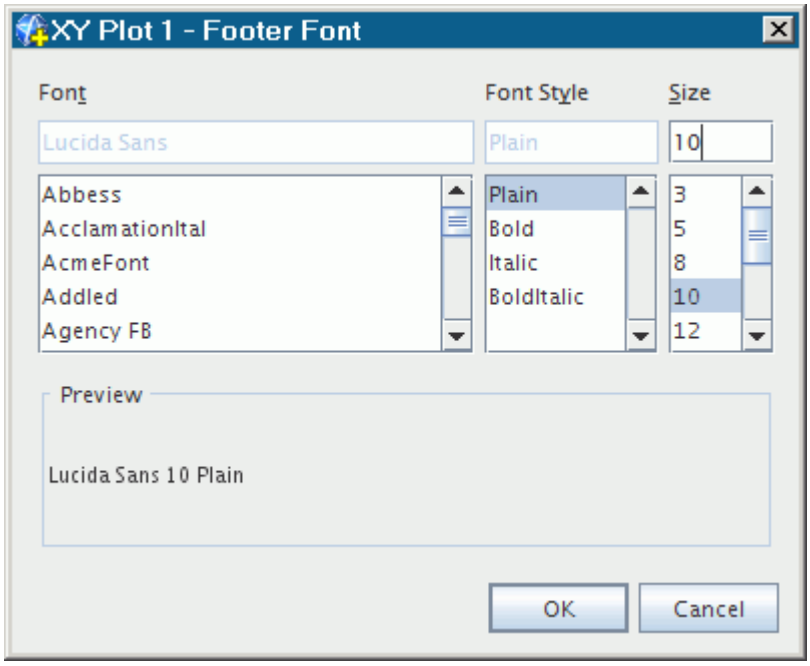
Monitor Plot Properties

Title	The overall title of the plot. This label appears centered at the top of the plot. Its position and format attributes cannot currently be modified. To remove the title set this value to blank.
X-Axis	Selects the x-axis for the monitor plot. The choice of axes depends on the total number of axes in the plot.
X-Axis Monitor	Selects a monitor for the x-axis. This allows plotting one monitor against another.
X-Axis Monitor Units	Specifies the units that are applied to the x-values of the monitor data series.

Monitor Plot Expert Properties

Aspect Ratio	In the drop-down list, **Fit Screen** is the default. Selection of any of the following aspect ratios can leave blank space in the graphics display: • **16:9** • **16:10** • **4:3** • **1:1** See Saving a Plot Image File.	
Orientation	In the drop-down list, choose between **Landscape** (default) and **Portrait**.	
Antialiasing	Controls appearance of labels to visual quality.	
	Activated	Uses antialiasing (default), which blends the edge of the characters with the background, creating a smoother appearance on screen.
	Deactivated	Leaves antialiasing off. This setting causes edges of characters to remain sharp, allowing for better printing.

Hover Interactions	Controls whether the display responds to mouse hovering.	
	Activated	Provides interactive responses based on the information under the mouse pointer.
	Deactivated	Leaves the display at its default, with no response to hovering.
Show Highlights	When activated, highlights selected objects in the plot display. See Highlighting Objects.	
Title Font	The font family, font style and font size of the title. Use the customizer to activate a dialog to set this property (changing the text will have no effect).  The property value is displayed using the selected font family, style and size.	
Footer	A text box in which you can type a footer for the plot.	
Footer Font	The font family, font style and font size of the footer. Use the customizer to activate a dialog to set this property (changing the text will have no effect).	

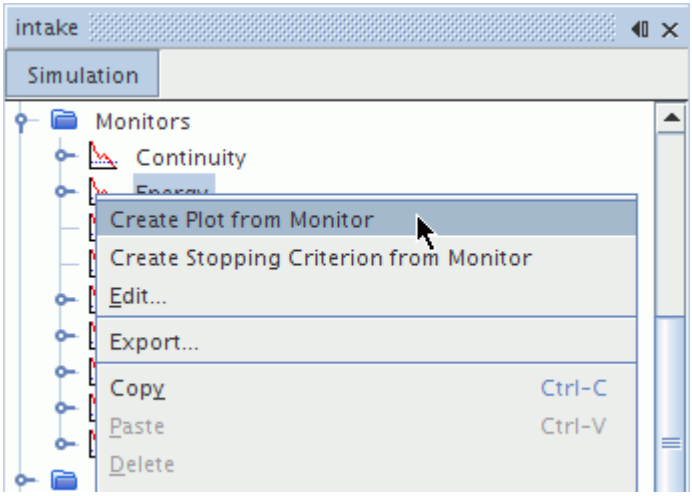
	 The property value is displayed using the selected font family, style and size.
Data Series Order	Lets you adjust the order of appearance of data series in a plot display. See Changing the Data Display Order.

Monitor Plot Pop-Up Menu

Open	Executes the open plot action to display this plot in the *Graphics* window or make it active if it is already open.
Reorder Data Series	Activates a dialog in which you adjust the order of appearance of data series in a plot display. See Changing the Data Display Order.
Tabulate	Executes the tabulate action to display the data in a spreadsheet-like dialog. If there is no data in the series, or if there are no series added to the plot, then this dialog will be empty.
Export...	Executes the export plot data action allowing you to save the series data in a .csv file.
Hardcopy...	Saves the plot to a file in one of several formats.
Create Plot Image Annotation	Creates a plot annotation under the Annotation Manager that uses the selected plot.
Apply Plot Options	Sets preferred plot attributes.

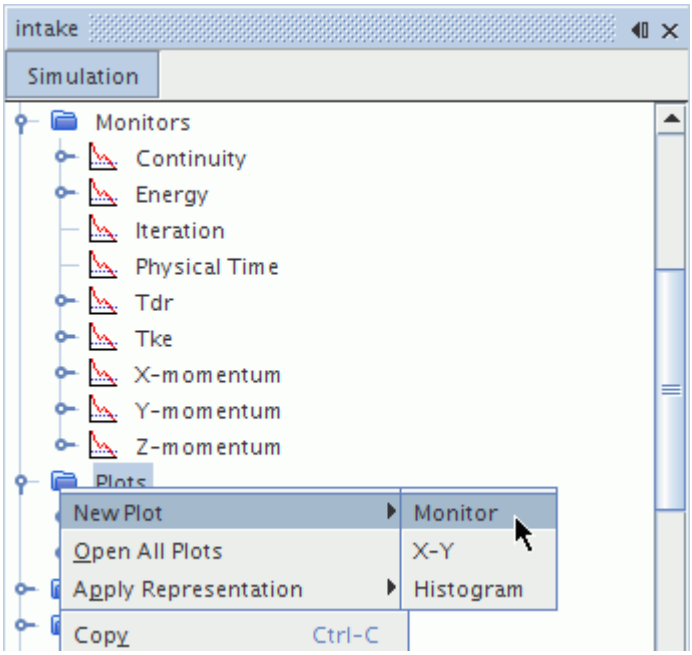
Creating a Monitor Plot

To create a plot displaying the collected values of a monitor, right-click on the monitor node and select **Create Plot from Monitor**.



If more than one monitor is selected, then a plot is created based on the monitor that was selected first.

Alternatively, you can right-click on the **Plots** node and select **New Plot** > **Monitor** and then add the desired monitors to the plot.



See Also:

Using Maximum and Minimum Reports

Working with Monitor Data Sets

A monitor data set, represented in the simulation tree by a node with properties and a pop-up menu, defines the source of data and the style of plotting the data.

Monitor Data Set Properties

Legend Name	Changes the legend text in the plot display.
Lock Legend Name	When activated, prevents any automatic changes to the legend name. The legend name is linked to the node name which in turn reflects the name of the monitor that the data series represents. Therefore, name changes to the monitor are automatically reflected in the name of the data series node and the legend name. If this property is activated, the current legend name is locked, effectively unlinking the property from the node name. Setting the legend name automatically activates this lock.
Monitor	The name of the monitor that is being used for this series. The monitor linked to this data set cannot be deleted while a link is in place (read-only).
Y-Units	Specifies the units that are applied to the y-values of the data series.
Y-Axis	Selects the axis on which this data series is plotted. The choice of axes depends on the total number of axes in the plot.

Expert Properties

Sort Plot Data	Determines whether the data will be sorted based on x-values--this ensures that the abscissa (horizontal axis) values run in an increasing order.	
	Activated	Sort the data
	Deactivated	Do not sort the data
X Offset	A scalar offset that will cause all values of the x-axis data to be shifted by this amount. The units of the offset are implicitly equivalent to those associated with the x-axis data.	
Y Offset	A scalar offset that will cause all values of the y-axis data to be shifted by this amount. The units of the offset are implicitly equivalent to those associated with the y-axis data.	
X Scale Factor	A scaling factor that will cause all values of the x-axis data to be scaled by this amount.	
Y Scale Factor	A scaling factor that will cause all values of the y-axis data to be scaled by this amount.	

Monitor Data Set Pop-Up Menu

Tabulate	Executes the tabulate data action to view the data in a two-column spreadsheet-style view of the data in this data set. The abscissa (horizontal axis) data will be in the first column and the ordinate (vertical axis) data will be in the second column. If there is no data in the data set, the dialog will be empty.
Export...	Executes the export plot data action allowing you to save the series data in a .csv file.
Apply Plot Options	Sets preferred plot attributes.
Delete	Removes the data from the plot. It does not delete the associated monitor.

Normalizing the Plot

To normalize the values in a monitor plot, use the Normalization Option of the monitor being plotted.

Plotting One Monitor against Another

A monitor plot lets you plot any monitor against any other.

In a default monitor plot, the y-axis uses any one of various monitors while the x-axis represents solution progress. (The x-axis uses the **Iteration** monitor in a steady simulation and the **Physical Time** monitor in an unsteady one.)

However, monitor plots have the added flexibility of plotting monitors against one another with the *x-Axis Monitor* property, which contains a list of all monitors in the simulation. This feature helps you determine correlations between monitors, especially over time. As the simulation run progresses, some monitors define non-monotonic functions: periodic ones such as crank angle, and oscillating ones such as wave height.

Each monitor has an independent data sampling policy based on its *Trigger* and *Frequency* properties. It is not guaranteed that the x- and y-axis monitors sample data at the same rate. Therefore, while the y-axis monitor collects a sample at a given iteration, the plot does not update until the x-axis monitor has also collected a sample.

See the example in the next section.

Note: The *Trigger* property of a monitor does not control the frequency at which the data is plotted. It specifies the frequency at which the monitor collects the data. To control the frequency at which the data is queried and plotted, use the Update feature of the plot node itself. If you want to plot data against physical time, make sure that the same **Physical Time** monitor exists throughout the solution. Otherwise the plot data appears incomplete. If you are running a steady solution but expect to need a plot with physical time, start by running the unsteady solver then switch to the steady one; this ensures that Physical Time exists in the simulation.

Contents:

Example: The Pitch of a Boat

Example: The Pitch of a Boat

This example uses an unsteady case from a tutorial simulating the motion of a boat in head waves.

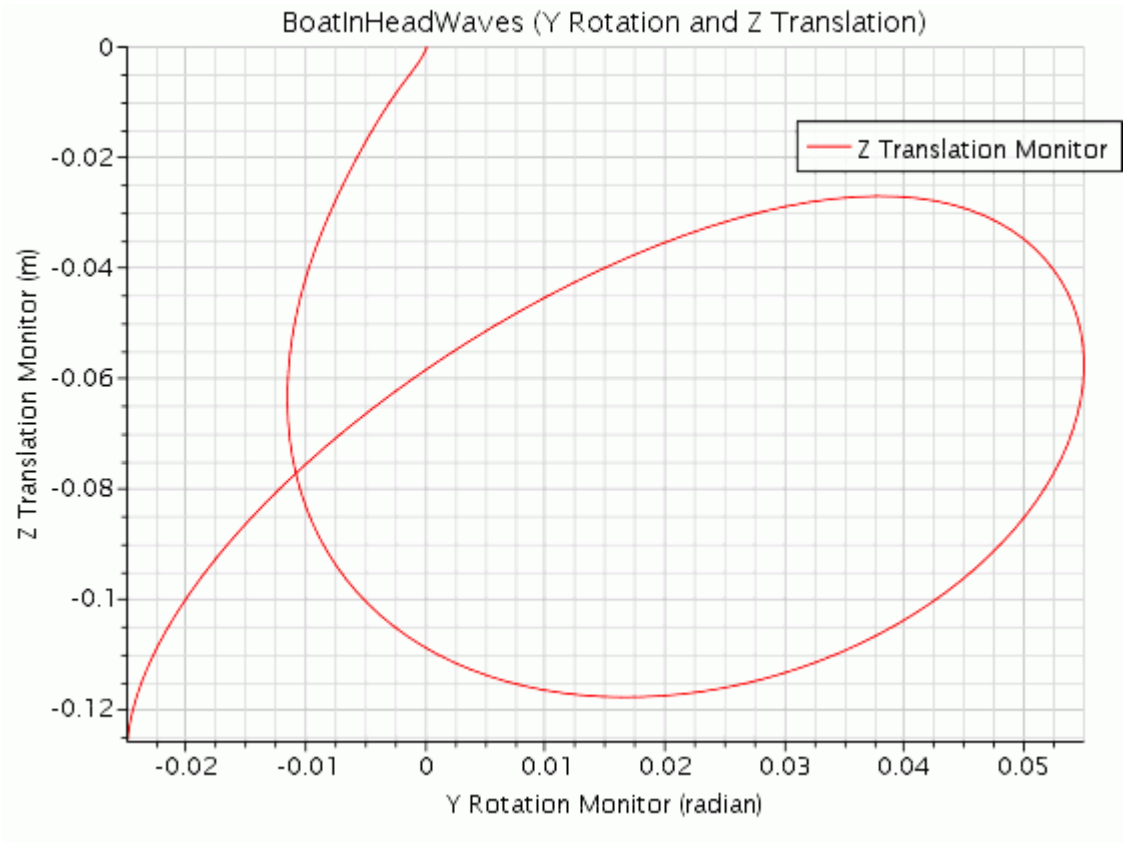
The purpose is to show a correlation analysis between different monitored variables, in this example y-rotation against z-translation:

- Right-click the monitor node (**Z Translation Monitor** in this example) and select **Create Plot from Monitor**.

  Note: It is possible to set up a monitor plot using two other techniques: creating an empty monitor plot node and adding the monitors manually, or creating a monitor plot simultaneously with a report-based monitor.

- By default this plot has the following characteristics:
  - The title of the plot has the name of the monitor.
  - The monitor that was used to create this plot is selected for the y-axis.
  - The x-axis is `Physical Time` because this case is unsteady. (In a steady case, the x-axis would be `Iteration`.)
- To reflect the content of the plot, change the *Title* property of the plot node. In this example, the new name is `BoatInHeadWaves (Y Rotation and Z Translation)`.
- To add a monitor to the x-axis of the monitor plot, select the node of that plot and choose the monitor in the drop-down list of the *x-Axis Monitor* property. (In this example it is the `Y Rotation Monitor`.)

The resulting plot is shown in the following screenshot.



Creating a Custom Residual Plot

Since residual monitors are monitor objects, they can be added to any monitor plot in order to create custom residual plots.

Working with XY Plots

This section explains XY plots, which can display scalar values or table data.

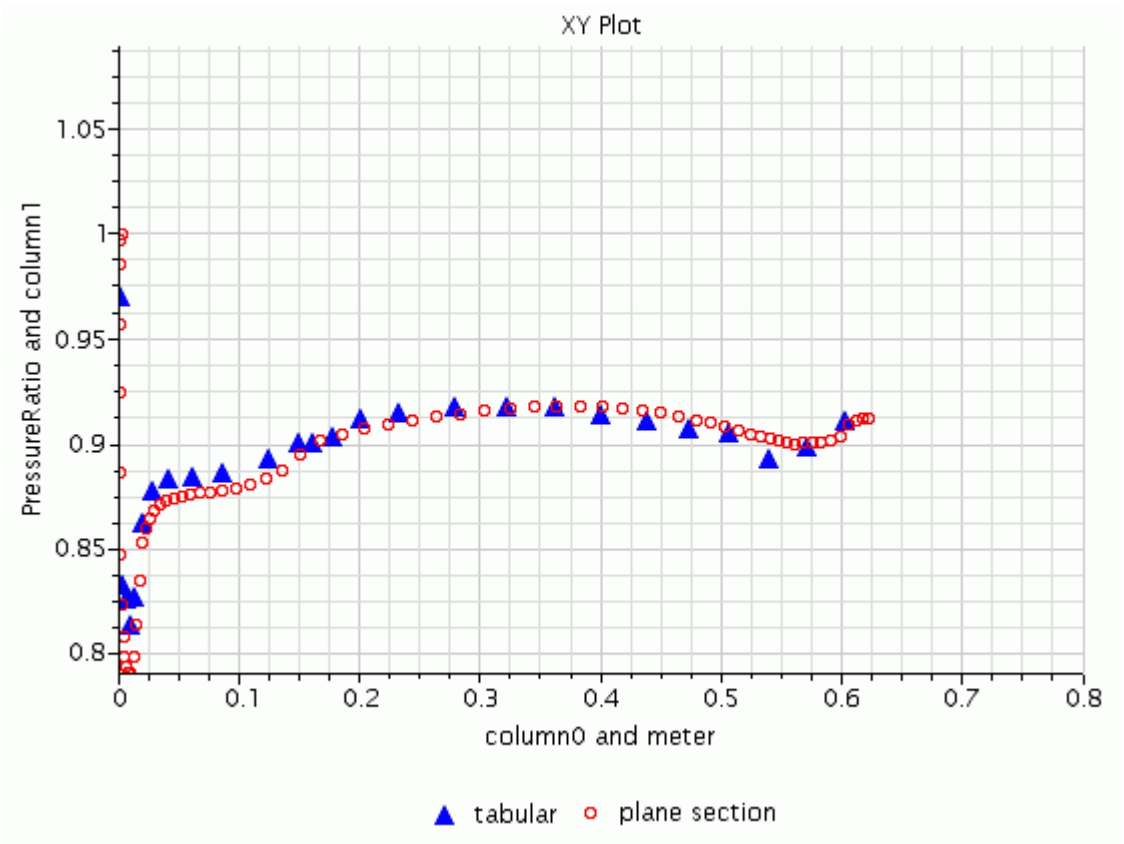
Contents:

What Is an XY Plot?
Creating an XY Plot
Plotting a Solution Variable
Using Heatmaps

What Is an XY Plot?

XY plots are two-dimensional graphs that can be used to display graphs of scalar values from the set of available solution variables or from table data.

The XY plot node has properties and a pop-up menu. An example of an XY plot is shown below.



Contents:

XY Plot Properties
XY Plot Pop-Up Menu

See Also:

Working with Displays

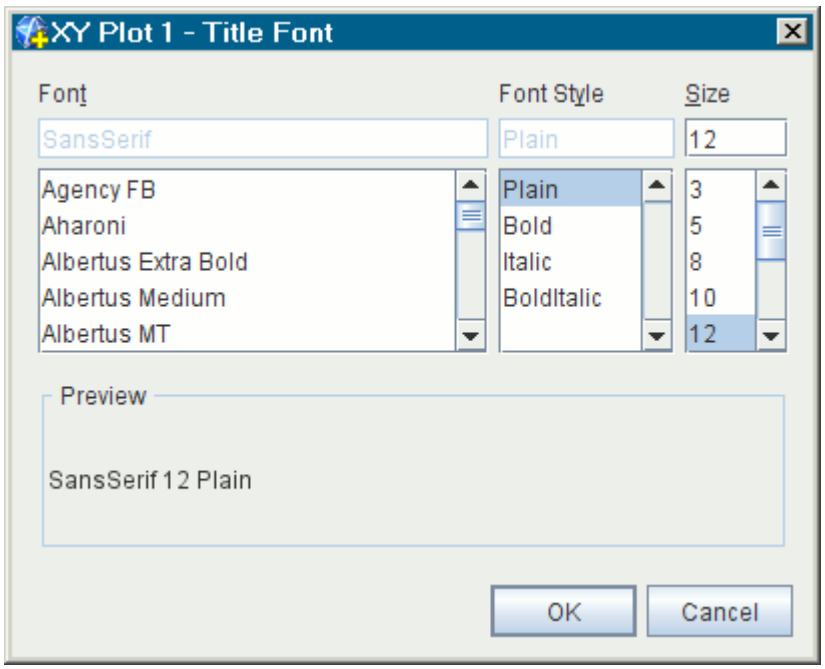
XY Plot Properties

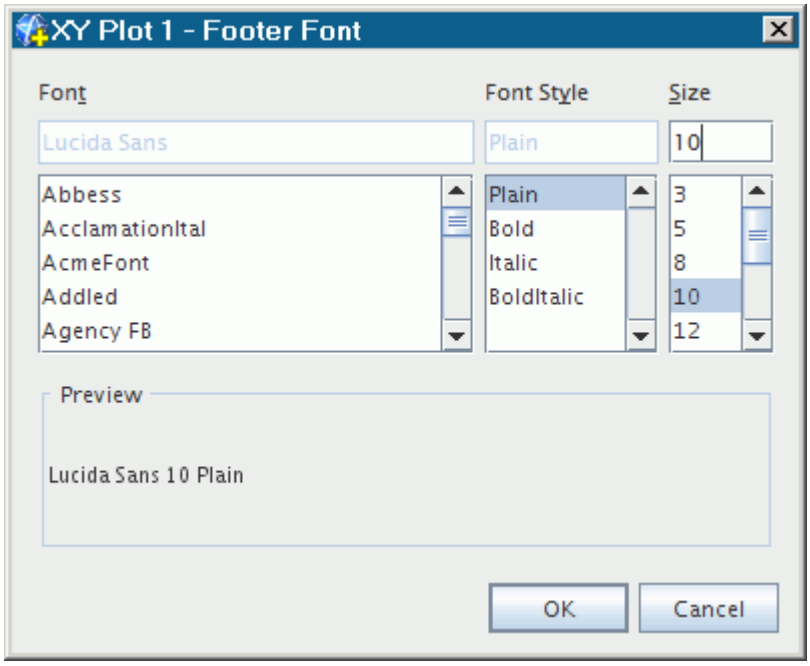
You can control some of the characteristics of the XY plot with the properties of the node.

Title	The overall title of the plot. This label appears centered at the top of the plot. Its position and format attributes cannot currently be modified. To remove the title set this value to blank.
Parts	A list of the model and derived parts from which the data are extracted. Editing the property opens the in-place object selector.
Representation	Selects a representation. Additional selections will appear as representations are added to the simulation.

XY Plot Expert Properties

Aspect Ratio	In the drop-down list, **Fit Screen** is the default. Selection of any of the following aspect ratios can leave blank space in the graphics display: • **16:9** • **16:10** • **4:3** • **1:1** See Saving a Plot Image File.	
Orientation	In the drop-down list, choose between **Landscape** (default) and **Portrait**.	
Antialiasing	Controls appearance of labels to visual quality.	
	Activated	Uses antialiasing, which blends the edge of the characters with the background, creating a smoother appearance on screen.
	Deactivated	Leaves antialiasing off. This setting causes edges of characters to remain sharp, allowing for better printing.
Hover Interactions	Controls whether the display responds to mouse hovering.	

	Activated	Causes the display to respond to hovering.
	Deactivated	Leaves the display at its default, with no response to hovering.
Show Highlights	When activated, highlights selected objects in the plot display. See Highlighting Objects.	
Title Font	The font family, font style and font size. Use the customizer to activate a dialog to set this property (changing the text will have no effect).  The property value is displayed using the selected font family, style and size.	
Footer	A text box in which you can type a footer for the plot.	
Footer Font	The font family, font style and font size of the footer. Use the customizer to activate a dialog to set this property (changing the text will have no effect).	

	 The property value is displayed using the selected font family, style and size.
Data Series Order	Lets you adjust the order of appearance of data series in a plot display. See Changing the Data Display Order.

XY Plot Pop-Up Menu

Open	Executes the open plot action to display this plot in the *Graphics* window or make it active if it is already open.
Reorder Data Series	Activates a dialog in which you adjust the order of appearance of data series in a plot display. See Changing the Data Display Order.
Tabulate...	Executes the tabulate action to display the data in a spreadsheet-like dialog. If there is no data in the series, or if there are no series added to the plot, then this dialog will be empty.
Export...	Executes the export plot data action allowing you to save the series data in a .csv file.
Refresh	Forces the plot display to refresh, overriding the update policy that is defined on the plot.
Hardcopy...	Saves the plot to a file in one of several formats.
Create Plot Image Annotation	Creates a plot annotation under the Annotation Manager that uses the selected plot.
Apply Plot Options	Sets preferred plot attributes.

Creating an XY Plot

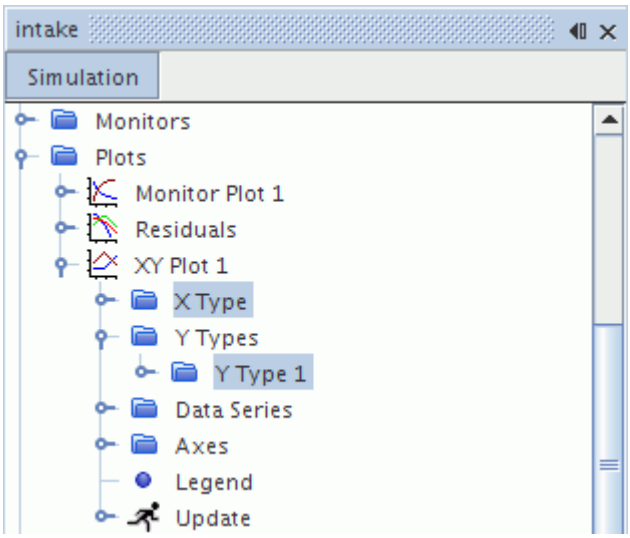
To create an XY plot, right-click on the **Plots** node and select **New Plot** > **XY Plot**.

The XY plot will initially be empty, so you will need to add solution variables and/or table data to be displayed.

Plotting a Solution Variable

The **X Type** and **Y Type** nodes let you display a solution variable in an XY plot.

These nodes function in a similar fashion in that they share a property. As the labels indicate, the **X Type** node sets the type of the abscissa (horizontal axis), while the **Y Type** node sets the type of the ordinate (vertical axis). Both these nodes contain a sub-node **Vector Quantity** or **Scalar Function** that maintains properties about the units and plotted variables.



Other differences between the **X Type** and **Y Type** nodes are as follows:

- The **Y Type** node contains the *Smooth* property to define where data is evaluated for its datasets. If this property is activated, data is evaluated at vertices; otherwise the data is evaluated at cells.
- The **Y Type** node contains dataset sub-nodes that correspond directly to the parts that are defined in the properties of the parent plot node. The dataset sub-nodes are similar to the ones in the **Data Series** sub-node of the monitor plot, and maintain the plotting style information.

In addition, the **Y Type** node is one of possibly multiple Y-type nodes within the **Y Types** manager node. The label of the **Y Type** node includes a number, and one node is created by default with the XY plot, having the label **Y Type 1**.

There are three steps to plotting a solution variable in an XY plot:

- Select a part or parts on which to plot the data.
- Select an axis and a scalar function to display along that axis.
- Specify a direction vector or another scalar function for the other axis.

It is also possible to have a plot with multiple Y types.

Once the solution variable or variables to plot have been specified, the appearance of the plot can be customized. If you wish to compare the solution variable values to reference data, you may plot table data in addition to the solution variable. If an XY plot is open while the simulation is iterating, the plot display updates in accordance to the policy defined by its Update sub-node. See Trigger Types.

X Type Properties

Type	Specifies what data will be used for this axis. Typically, one axis will be *Scalar* and the other *Direction* or *Scalar*. Although *Direction* is permitted on both axes, it makes no real sense to do this.	
	Direction	The axis data corresponds to point/cell position vectors dotted with the normalized direction vector specified in the **Vector Quantity** sub-node.
	Scalar	The axis data corresponds to a scalar field function at the point/cell defined in the **Scalar Function** sub-node.
X-Axis	Selects the x-axis for the plot. The choice of axes depends on the total number of axes in the plot.	

Y Type Properties

Type	Specifies what data will be used for this axis. Typically, one axis will be *Scalar* and the other *Direction* or *Scalar*. Although *Direction* is permitted on both axes, it makes no real sense to do this.	
	Direction	The axis data corresponds to point/cell position vectors dotted with the normalized direction vector specified in the **Vector Quantity** sub-node.
	Scalar	The axis data corresponds to a scalar field function at the point/cell defined in the **Scalar Function** sub-node.
Y-Axis	Selects the axis on which the data for this Y-type is plotted. The choice of axes depends on the total number of axes in the plot.	
Smooth Values	Defines where the data is evaluated on the associated part.	
	Activated	Interpolates the variables to the vertices of the parts.
	Deactivated	Evaluates the variables at the cells of the parts.

Expert Properties

Data Focus	Selects a data focus object to apply to this Y-type. The choice of data focus objects depends on the total number of these objects in the simulation. When a data focus object is selected, a data focus sub-node is added to this node. See Exploring Solutions with Data Focus.

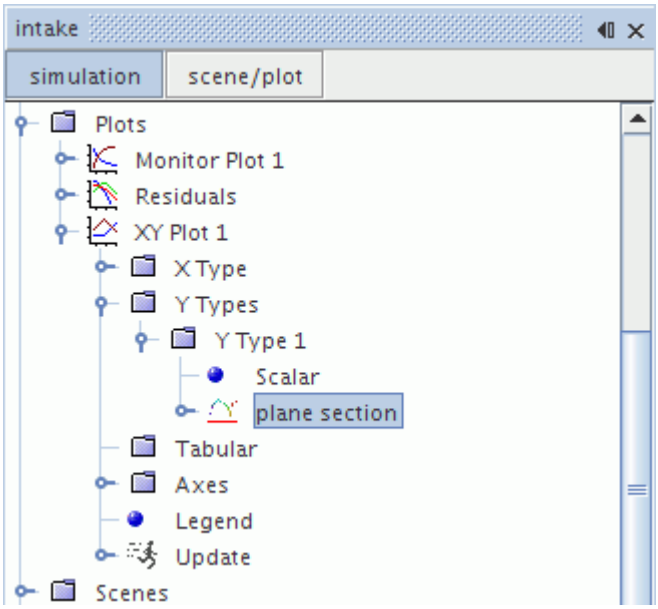
Contents:

Selecting Parts
Selecting a Scalar Function on an Axis
Specifying a Direction Vector on an Axis
Adding Multiple Y Types to a Plot
Changing the Variable Plotting Style
Plotting the Velocity Profile Across a Pipe Flow
Using Line Probes over Planar Boundaries

Selecting Parts

XY plots can display data from model parts, such as boundaries, or derived parts, such as line and point probes.

A part data set (or internal data set) node, which has its own properties and pop-up menu, defines the source of data and the style of plotting the data.



For example, if you have reference data at a set of locations, you might create point probes at the corresponding coordinates in your simulation and add the point probe parts to your XY plot.

To add an existing part to an XY plot, select the node of that XY plot and click the *Parts* property. This will bring up the object selector dialog, which allows you to add parts to the plot.

A part can also be copied into the current plot in two ways using drag-and-drop. The first is to drag the part node into the plot window. The second is to drag the part node to the XY plot node, which requires navigating the tree.

Part Data Set Properties

Legend Name	Changes the legend text in the plot display.
Lock Legend Name	When activated, prevents any automatic changes to the legend name. The legend name is linked to the node name which in turn reflects the name of the part that the data series represents. Therefore, name changes to the part are automatically reflected in the name of the data series node and the legend name. If this property is activated, the current legend name is locked, effectively unlinking the property from the node name. Setting the legend name automatically activates this lock.
Part	The associated part (read-only).
Y Type	The associated Y type object (read-only).
Styles	Chooses between the symbol/line style (default) and the heatmap style. The **Heatmap** option activates the **Heatmap Style** node. See Heatmap Objects Reference.

Expert Properties

Sort Plot Data	Determines whether the data will be sorted--this ensures that the abscissa (horizontal axis) values run in an increasing order. **Note:** In parallel, cell data is not sorted across partitions. This does not affect the rendered result for a scatter plot. However, in the case of a line chart, if the source data is not previously sorted, this affects how the data is plotted.	
	Activated	Sort the data
	Deactivated	Do not sort the data
X Offset	A scalar offset that will cause all values of the x-axis data to be shifted by this amount. The units of the offset are implicitly equivalent to those associated with the x-axis data.	
Y Offset	A scalar offset that will cause all values of the y-axis data to be shifted by this amount. The units of the offset are implicitly equivalent to those associated with the y-axis data.	
X Scale Factor	A scaling factor that will cause all values of the x-axis data to be scaled by this amount.	
Y Scale Factor	A scaling factor that will cause all values of the y-axis data to be scaled by this amount.	

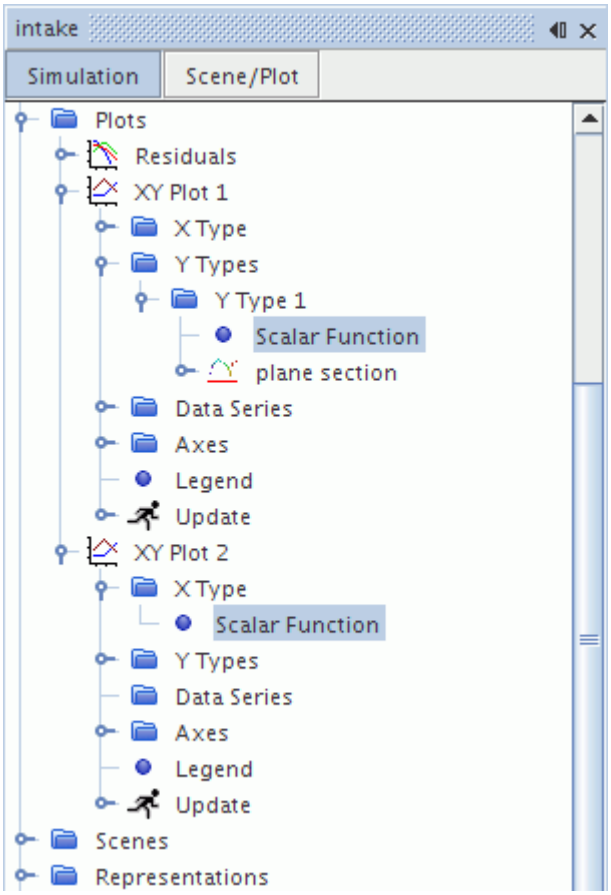
Part Data Set Pop-Up Menu

Tabulate	Executes the tabulate data action to view the data in a two-column spreadsheet-style view of the data in this series. The abscissa (horizontal axis) data will be in the first column and the ordinate (vertical axis) data will be in the second column. If there is no data in the series, the dialog will be blank.
Export...	Executes the export plot data action allowing you to save the series data in a .csv file.
Apply Plot Options	Sets preferred plot attributes.
Delete	Removes the data from the plot. It does not delete the associated part.

Selecting a Scalar Function on an Axis

Each XY plot node contains objects within it to specify what should be plotted along the X and Y axes of the graph, respectively.

To select a solution variable to be plotted along one of the axes, open the corresponding node (either **X Type** or **Y Types** > **Y Type**) and select **Scalar** for the *Type* property. This will activate the **Scalar Function** node, which has its own properties.



This node defines the data for the axis of this node's parent to come from a scalar field function. It is only visible if the *Type* property of the parent node is set to **Scalar**.

To specify the solution variable, select the **Scalar Function** sub-node and use the *Field Function* property. Select units with the *Units* property.

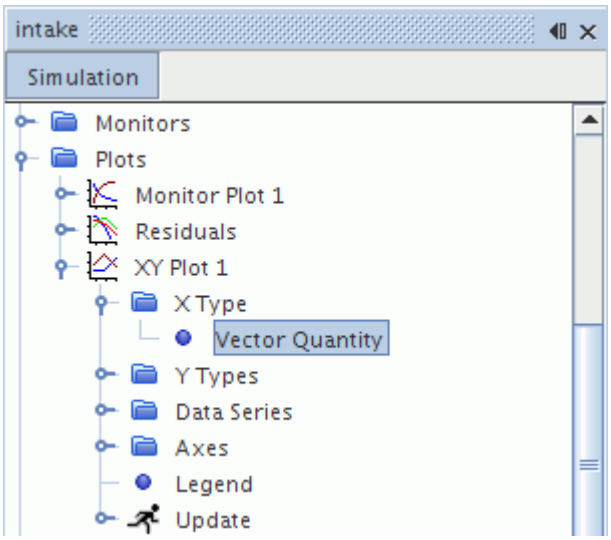
Scalar Function Node Properties

Field Function	The specific scalar field function to be plotted as the data value for this axis.
Units	Defines the units that are applied to the selected function.

Specifying a Direction Vector on an Axis

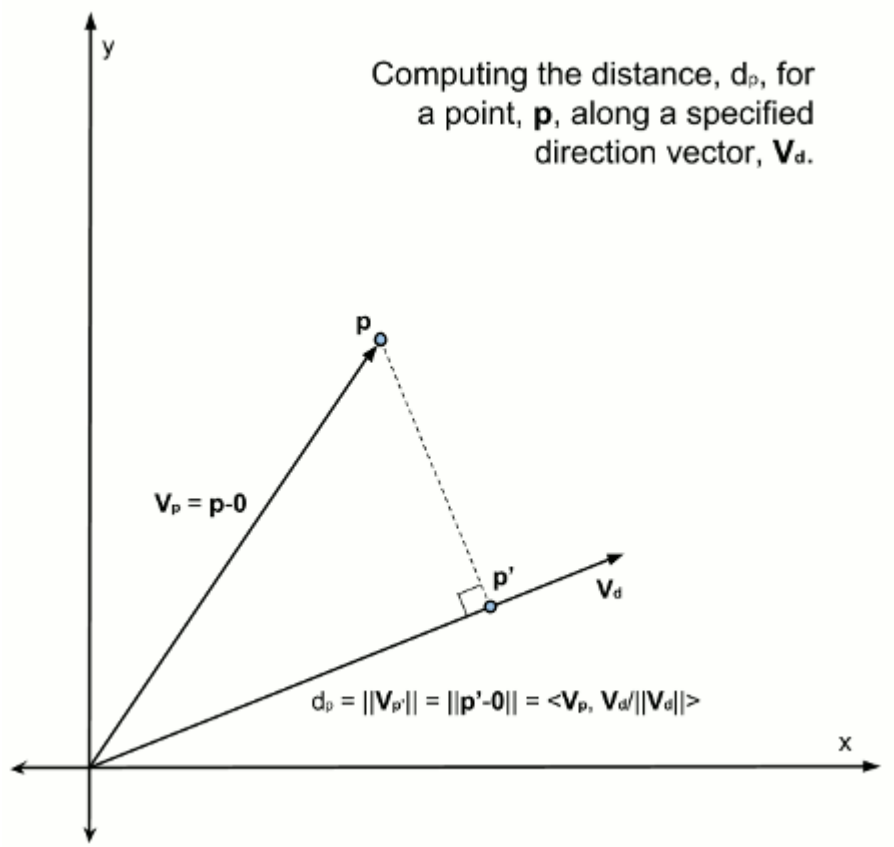
Each XY plot node contains **X Type** and **Y Type** sub-nodes to specify what should be plotted along the X and Y axes of the graph, respectively. To plot position information along one of the axes, select the corresponding node (either **X Type** or **Y Type**) and select **Direction** for the *Type* property.

This activates the **Vector Quantity** node, which has its own properties.



This node defines the data for the axis of this node's parent to come from a direction vector. It is only visible if the *Type* property of the parent node is set to **Direction**.

Since simulation parts may exist in three-dimensional space while each axis of an XY plot represents only a single dimension, the three-dimensional coordinates of the part being plotted must be reduced to one. This objective is accomplished by specifying a direction vector to which the position of sample locations is projected.



To specify the direction in STAR-CCM+, select the **Vector Quantity** node under the **X Type** or **Y Type** nodes. For example, in order to plot X coordinates of the selected parts, you would specify a direction vector of [1,0,0].

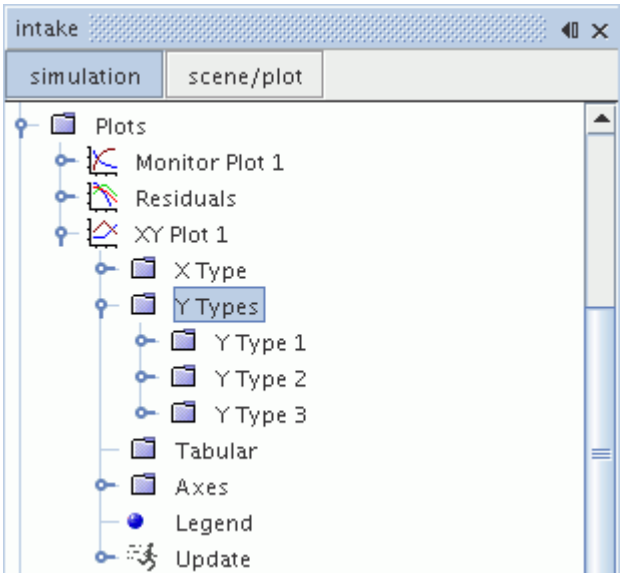
Vector Quantity Node Properties

Value	A Cartesian vector that specifies the orientation (in the space of the simulation) of the coordinate axis.

Adding Multiple Y Types to a Plot

It is possible to plot multiple scalars in a single plot by using a separate Y type object for each scalar.

The **Y Types** manager node, which has its own pop-up menu, contains all of the report nodes in the simulation tree.



By default a single **Y Type** node exists within the **Y Types** manager node. To add an extra **Y Type** node, right-click the **Y Types** manager node in the XY plot node and select **New**.

Select the scalar for each Y type, then initialize the solution. The Y types will be plotted in the plot display.

By default the y-axis title concatenates the different field functions that are plotted, corresponding to the different Y types.

Y Types Pop-Up Menu

New	Creates a new **Y Type** node.

Changing the Variable Plotting Style

You can make XY plots appear smoother by using interpolation.

To do this, open the **Plots** node, the node of the particular XY plot, and the **Y Types** node. Select the specific **Y Type** node.

Activate the *Smooth Values* property in the *Properties* window.

This produces a smoother contour of the plot.

The smooth plot is generated using vertex values rather than cell centroid-based values and appears smoother than the default style. However, for most scalar fields vertex values are interpolated from cell values, and thus data extraction with the smooth-filled option is more expensive in terms of computing power.

Plotting the Velocity Profile Across a Pipe Flow

A common plot in pipe flow simulations is the velocity profile. Plotting this profile in STAR-CCM+ requires that you first create a line probe derived part, and then refer to that part in a separate XY plot.

The following steps describe an example workflow of how to visualize a scalar velocity profile through a section of a pipe flow:

1. Create a 2D plane section across the width of the pipe flow geometry, parallel to the inlet and outlet part surfaces.
   Although this step is not crucial, it makes sure that the line probe lies along a surface that is parallel with the outlet. See Defining a Plane Section.
2. Create a 1D line probe across the diameter of the plane section. To make sure that the line probe is created precisely through the diameter of the plane section, in the *Create Line Probe* in-place dialog:
   a) Activate *Snap to Part*.
   b) To the right of the *Snap to Part* option, click ...(**Custom Editor**).
   c) In the *Select Object* dialog, select the **Derived Parts** > **[Plane Section]** part and click **OK**.
3. Click **Create**, then **Close**.

Each point in the line probe represents the location from which data is gathered. The number of points in the line probe represents the number of plots which appear on an XY plot.

4. To increase the number of points across the line probe, select the **Derived Parts** > **Line Probe** node and set *Resolution* to a higher value.
5. Create an XY plot. Right-click the **Plots** node and select **New Plot** > **XY Plot**.
6. Expand the **XY Plot 1** node and set the properties that you require. For example:

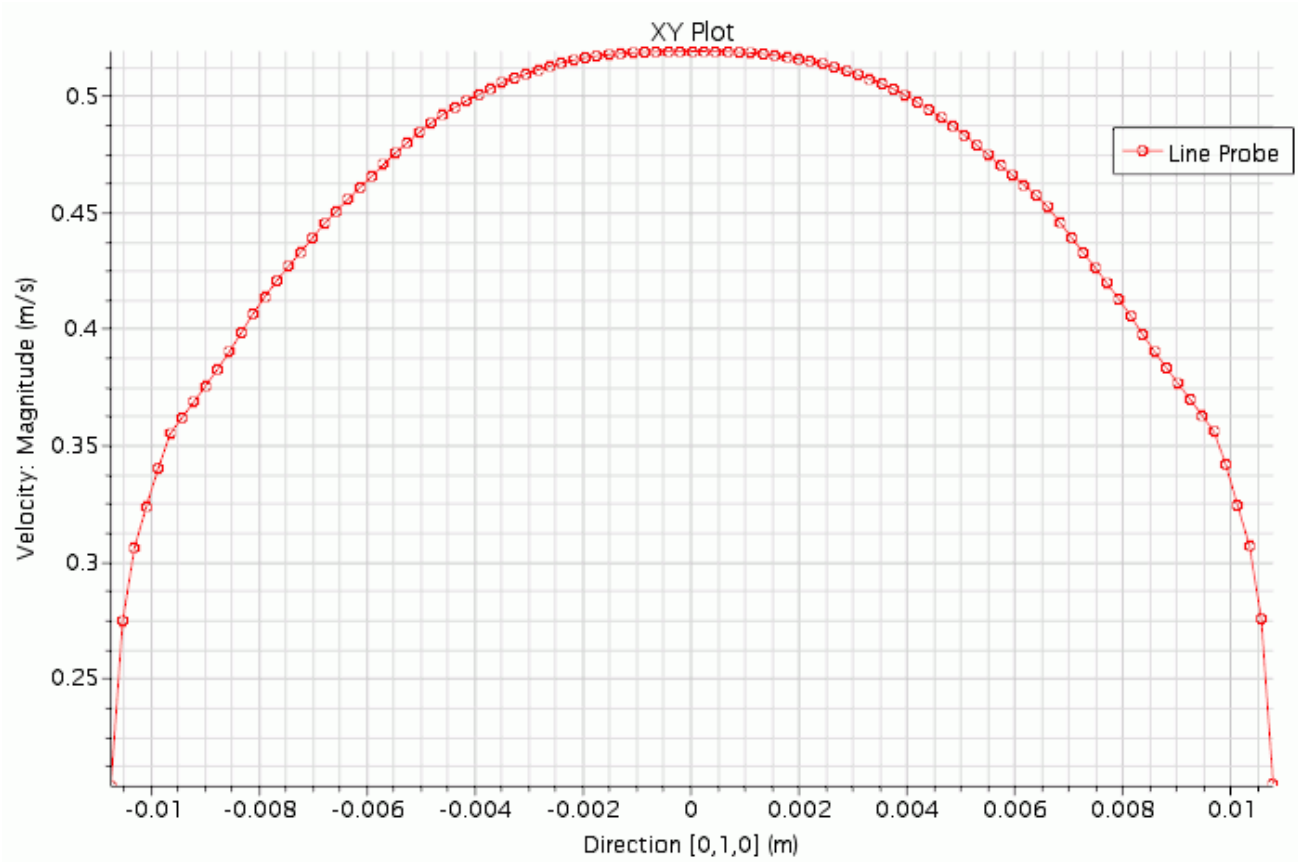
Node	Property	Setting
XY Plot 1	*Parts*	**Derived Parts** > **Line Probe**
X-Type		
⌊ **Vector Quantity**	*Value*	Set the X, Y, or Z coordinate to represent the direction of the line probe. Typically you set one of these coordinate directions to 1.0.
Y Types		
⌊ **Y Type 1**	*Smooth Values*	When this option is activated, the X-Y plot appears as a smooth curve as the line probe linearly interpolates from values that are located at the vertices of the region or boundary. These values are computed by averaging cell centre values from neighboring cells (when solving using a finite-volume method). If this option is deactivated, nearest cell centre values are used – which the solver produces

Node	Property	Setting
		directly – and the X-Y plot appears stepped, not smooth.
⌊ **Line Probe**		
⌊ **Line Style**	*Style*	Select one
⌊ **Scalar Function**	*Field Function*	**Velocity** > **Magnitude**

7. Open the XY plot in the *Graphics* window.

Note: Modifying derived parts or X-Y plots does not modify the mesh that the solution uses. Therefore, you do not need to rerun mesh operations or solvers.

The following image shows the scalar velocity magnitude on an XY plot which uses data that is taken from the line probe with *Smooth Values* activated.



Using Line Probes over Planar Boundaries

You can plot results accurately along a line over a planar boundary.

With the right settings, a line-probe can help you plot results, such as heat transfer, in a line on a planar boundary. See Defining a Line Probe.

1. With a scene open, create a line probe.
2. Select a planar boundary as the input part.
3. Activate the **Snap to Part** option.

   This option attaches the line to the selected surface.
4. Create an XY plot and select the line-probe in the **Parts** property. See Selecting Parts.

Using Heatmaps

Use the heatmap style to view a high concentration of data more clearly.

The heatmap, a style for XY plots, is designed to manage heavy plot data when the content would otherwise overwhelm the Symbol/Line style. Essentially, it uses the plot axes to define the rows and columns of a spreadsheet. Each cell defines a "box" whose upper and lower limits are based on a small range of both the vertical and horizontal axes.

The boxes are filled, matching your input data to the bounding range for each cell. Some boxes are empty. You can get a sense of how full each box is by using a scalar function to weight the results. If you color each cell in the spreadsheet by the weighting, you can see how the data is distributed within the XY plot.

Contents:

Setting and Viewing Heatmaps
Heatmap Objects Reference

Setting and Viewing Heatmaps

Accessing heatmaps begins with the *Styles* property of an internal data set.

1. In the data set sub-node of the **Y Type** node of the plot, make sure that the *Styles* property is set to **Heatmap**. This setting activates the **Heatmap Style** sub-node.
2. To increase the number of data points in either the x or y direction, enter settings for the *Bins* property of the **Heatmap Style** node.

   When you increase the number of bins, the bin values decrease since fewer cells are accumulated in each bin.

   To mitigate this effect, activate the *Normalize by Bin Size* option in the Expert Properties. That option divides the bin value by the actual size of the bin, ensuring that increasing or decreasing the number of bins does not lead to drastic changes in the bin values.
3. To activate a color bar that shows density of data, activate the *Use Color Lookup Table* property.

For information about more heatmap options, see Heatmap Objects Reference.

Heatmap Objects Reference

The **Heatmap Style** node and its sub-nodes let you control the appearance of concentrations of data in a plot display.

Specific nodes referenced here are:

Heatmap Style

See Heatmap Style.

Color

Uses the *Color* property to color unselected data. See Defining Colors.

Color Bar

See Color Bar.

Range

This node appears when the *Range Mode* property of the **Heatmap Style** node is set to **Manual**. To set the minimum and maximum bin values along each axis, use the *X-Axis Range* and *Y-Axis Range* properties.

Heatmap Style

A manager node that consists of properties and sub-nodes for controlling the heatmap in the plot. The sub-nodes control color and data range.

Heatmap Style Properties

Weighting	Determines how a cell falling into a bin influences the value of the bin.	
	Default	The value of a bin is equal to the number of cells that fall into it.
	User Specified	For each cell falling into a bin, a value from a field function is read and added to the value of the bin. To specify this field function, use the **Weighting** sub-node. (Currently values stored in the field function must be positive.) A typical use case is selecting Volume as the field function. This field function ensures that small cells only contribute a small amount to the final bin value. Without volume weighting, the heatmap shows simply the distribution of functions in terms of number of cells. With volume weighting, the heatmap shows the distribution of the function values in terms

		of occupied volume. When cell sizes in the mesh differ substantially, volume weighting makes a significant difference.
Bins	The number of bins along the x- and y-axes.	
Range Mode	Determines the bounds along the x- and y-axes. In **Auto** mode, x- and y-axis ranges are set to the min/max values of all data sets shown in the plot. In **Manual** mode you can specify both ranges using the **Range** sub-node.	

Heatmap Style Expert Properties

Clip Empty Bins	When this option is activated, bins into which no cell has fallen are not rendered.
Use Color Lookup Table	When this option is activated, a color lookup table is used to assign a color to the values that are stored in each bin. Settings for this lookup table can be changed using the **Color Bar** sub-node.
Normalize by Bin Size	When this option is activated, the bin values are divided by the size of the bin in data space (x-axis extent multiplied by y-axis extent).

Color Bar

This node replaces the **Color** node when the *Use Color Lookup Table* expert property of the **Heatmap Style** node is activated.

Color Bar Properties

Color Map	Selects colors to indicate values from lowest to highest. This property lets you customize the use of colors in the heatmap plot. For a list of the pre-defined choices, see Pre-Defined Colormaps.	
Levels	Sets the number of levels into which to divide the range of data. Higher numbers produce finer blends between values. The default setting is 32.	
Reverse	Reverses the sequence of the selected colormap. See Using Colormaps.	
	Activated	Color sequence is reversed (for example, red to blue rather than blue to red).
	Deactivated	Uses the default color sequence.
Visible	When activated, shows the color bar.	
Scale Mode	Activates a logarithmic scale mode for the color bar. See Switching the Scale Mode.	
	Linear	Uses linear scaling.

	Log10	Uses logarithmic scaling.

Color Bar Expert Properties

Ramp Type	Sets the shape of the table ramp to either linear, S-curve, or square-root. If you use the Linear ramp, colors vary linearly to bin values. The other ramp modes distort this color mapping according to the corresponding functions.	
	Linear	Specifies a linear ramp, .
	S-curve	Uses the S-curve, which tails off gradually at each end: $y = \frac{\sin((x - 1/2)\pi) + 1}{2}$
	Sqrt	Uses the square root, $y = \sqrt{x}$.

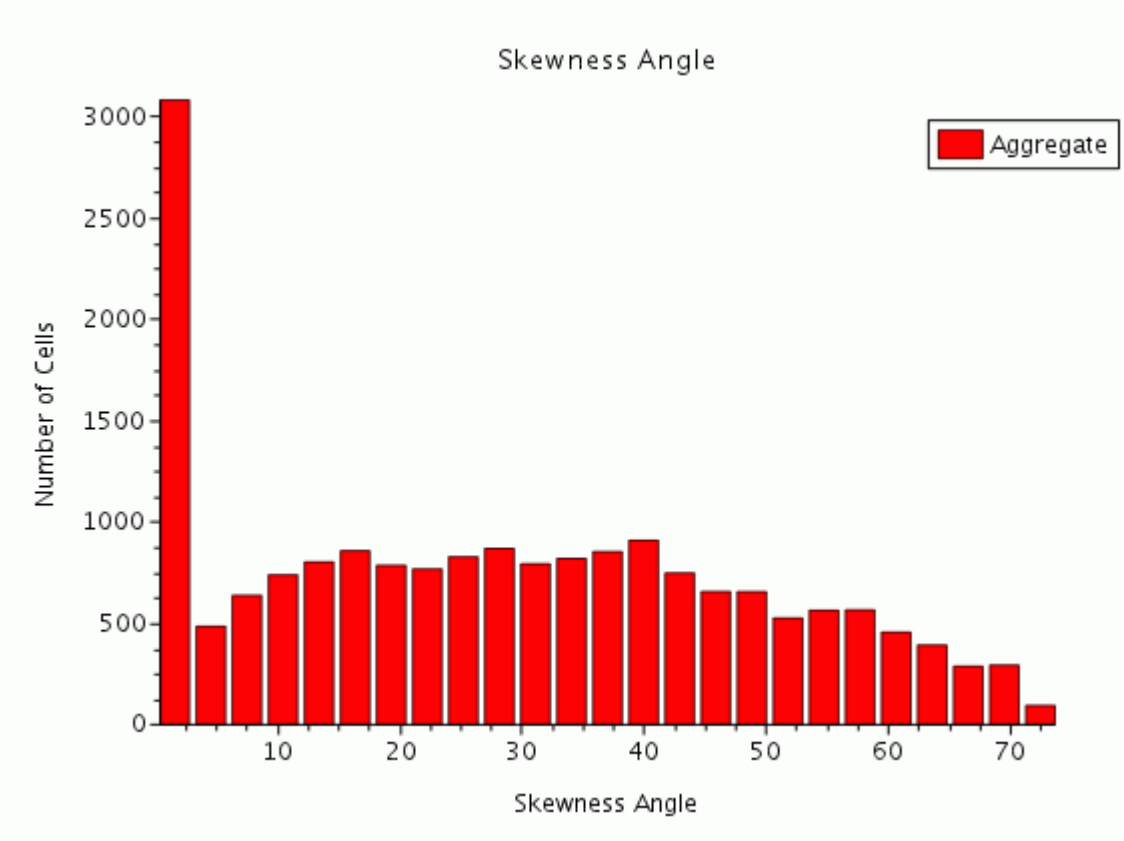
Working with Histogram Plots

Histogram plots are two-dimensional bar charts that can be used to display data distribution.

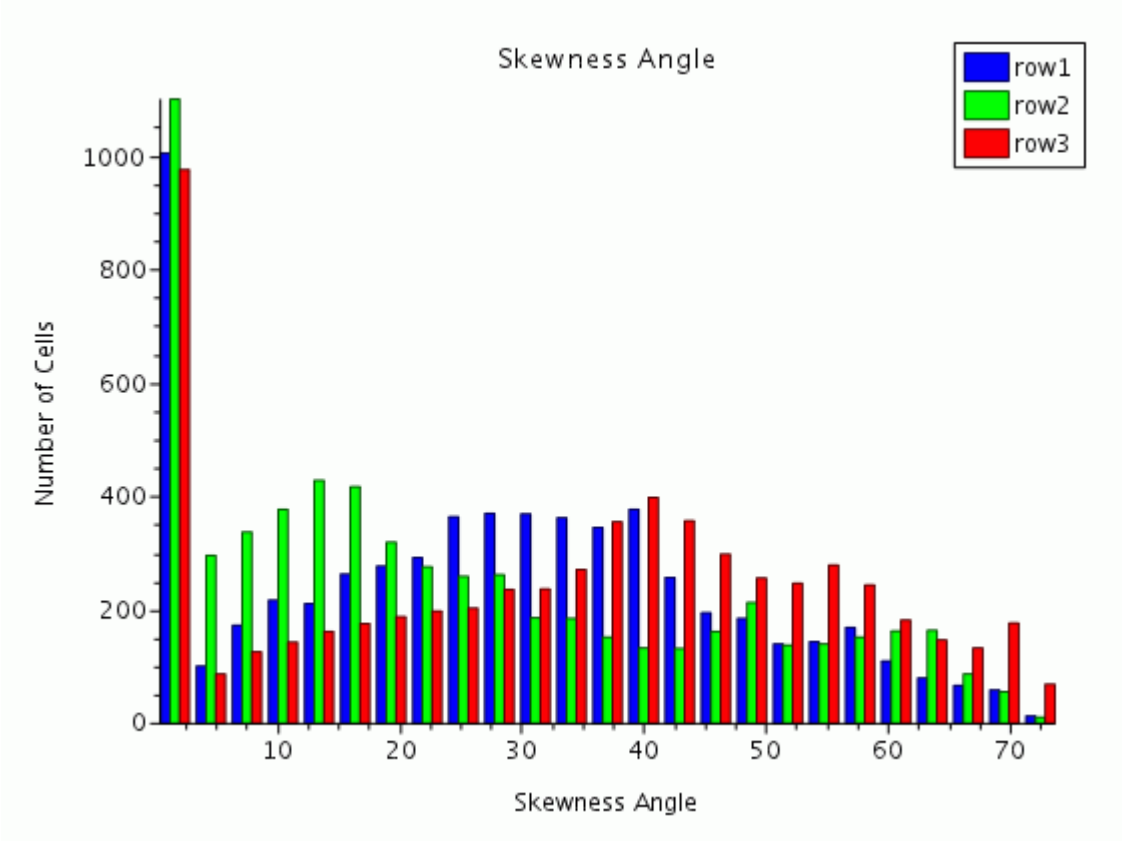
The data population is divided into discrete categories, called “bins”. The frequency of occurrences in each bin is given on the vertical scale. The **Histogram Plot** node has properties and a pop-up menu.

The **Display Mode** property provides three different options for presenting data in a histogram plot:

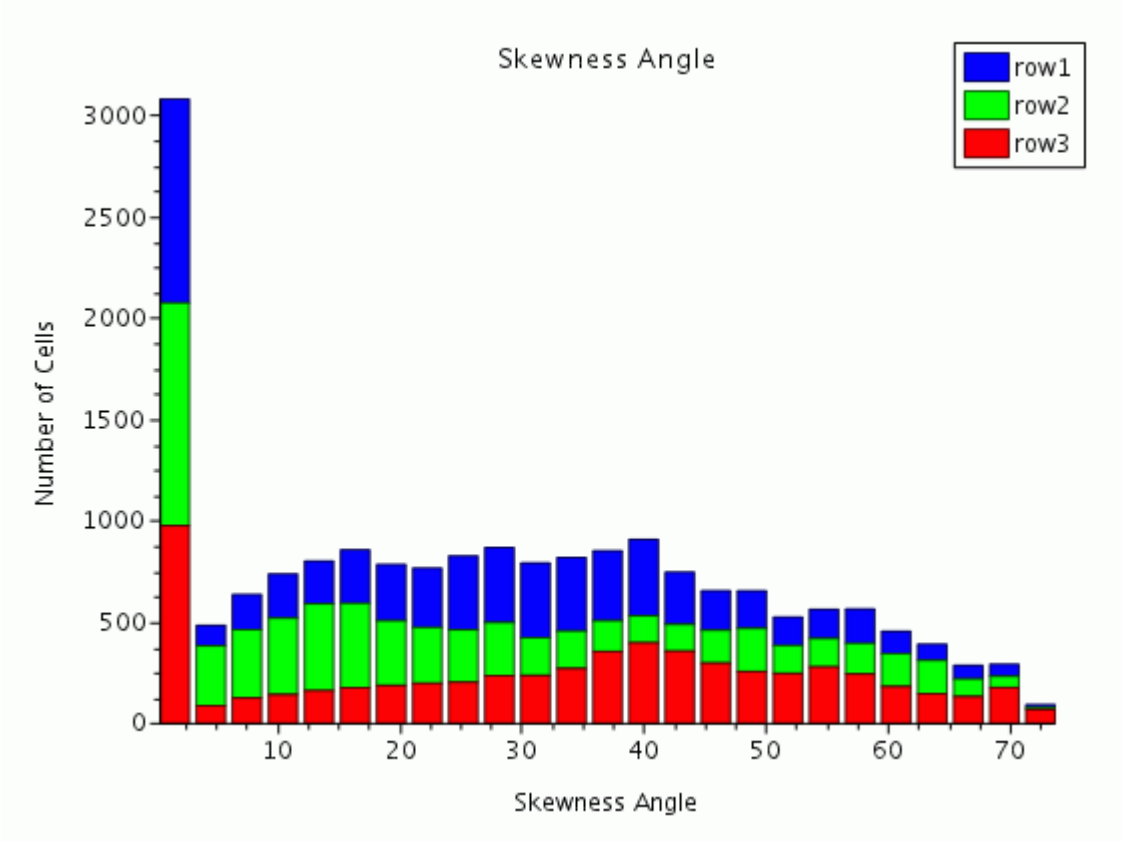
- **Aggregate**: All parts in the plot are treated as a combined single entity.



- **Side-by-Side**: Parts are separated as individual bars that are easier to compare, but are not as accurately aligned with the axis values underneath them.



- **Stacked**: This option divides the bars into quantities for each part.



Creating a Histogram Plot

To create a new histogram plot, right-click on the **Plots** node and select **New Plot** > **Histogram**.

The histogram plot will initially be empty, so you will need to add data to be displayed by accessing the plot node's *Parts* property.

Setting up a Histogram

To specify the details of how your data is to be displayed, use the contents of the **Histogram Plot** > **Axis Type** node:

1. Set the number of intervals along the x-axis in the *Bins* property of the **Axis Type** node.
2. Specify the field function in the *Field Function* property of the **Axis Type** > **Bin** node.
3. To set the range:
   a. Set the *Range Mode* property of the **Axis Type** node to **Manual**.
   b. Select the **Axis Type** > **Range**node and set the minimum and maximum values in the *Range* property.
4. To customize the weighting with a scalar function:
   a. Set the *Weighting* property of the **Axis Type** node to **Scalar Function**.
   b. Select the **Axis Type** > **Weighting** node and make a selection in the *Field Function* property.

Histogram Plot Properties

Title	The overall title of the plot. This label appears centered at the top of the plot. Its position and format attributes cannot currently be modified. To remove the title set this value to blank.	
Parts	A list of the model and derived parts from which the data are extracted. Editing the property opens the object selector dialog.	
Representation	Selects a representation. As representations are added to the simulation, additional selections will appear.	
Display Mode	Controls how the histogram plot works with parts.	
	Aggregate	Treats parts as a single entity. This option is the default in restored simulations that have the *Aggregate Parts* property activated.
	Side-by-Side	Works with each part as a separate entity. This option is the default in restored simulations that have the *Aggregate Parts* property deactivated.
	Stacked	The various data series in the plot are stacked one on top of the other instead of side-by-side. The stacking order matches the order that appears in the legend. This option is the default setting in new simulations.
Space Segments	When activated, causes a space to appear between bar segments. Activated by default.	
Aspect Ratio	In the drop-down list, **Fit Screen** is the default. Selection of any of the following aspect ratios can leave blank space in the graphics display: • **16:9** • **16:10** • **4:3** • **1:1** See Saving a Plot Image File.	
Orientation	In the drop-down list, choose between **Landscape** (default) and **Portrait**.	
Antialiasing	Controls appearance of labels to visual quality.	
	Activated	Uses antialiasing, which blends the edge of the characters with the

		background, creating a smoother appearance on screen.
	Deactivated	Leaves antialiasing off. This setting causes edges of characters to remain sharp, allowing for better printing.
Hover Interactions	Controls whether the display responds to mouse hovering.	
	Activated	Causes the display to respond to hovering.
	Deactivated	Leaves the display at its default, with no response to hovering.
Show Highlights	When activated, highlights selected objects in the plot display. See Highlighting Objects.	
Title Font	The font family, font style and font size. Use the customizer to activate a dialog to set this property (changing the text will have no effect). The property value is displayed using the selected font family, style and size.	
Footer	A text box in which you can type a footer for the plot.	
Footer Font	The font family, font style and font size of the footer. Use the customizer to activate a dialog to set this property (changing the text will have no effect). The property value is displayed using the selected font family, style and size.	
Data Series Order	Lets you adjust the order of appearance of data series in a plot display. See Changing the Data Display Order.	

Histogram Plot Pop-Up Menu

Open	Executes the open plot action to display this plot in the *Graphics* window or make it active if it is already open.
Reorder Data Series	Activates a dialog in which you adjust the order of appearance of data series in a plot display. See Changing the Data Display Order.
Tabulate	Executes the tabulate action to display the data in a spreadsheet-like dialog. If there is no data in the series, or if there are no series added to the plot, then this dialog will be empty.
Export...	Executes the export plot data action allowing you to save the series data in a .csv file.
Refresh	Forces the plot display to refresh, overriding the update policy that is defined on the plot.
Hardcopy...	Saves the plot to a file in one of several formats.

Create Plot Image Annotation	Creates a plot annotation that uses the selected plot.
Apply Plot Options	Sets preferred plot attributes.

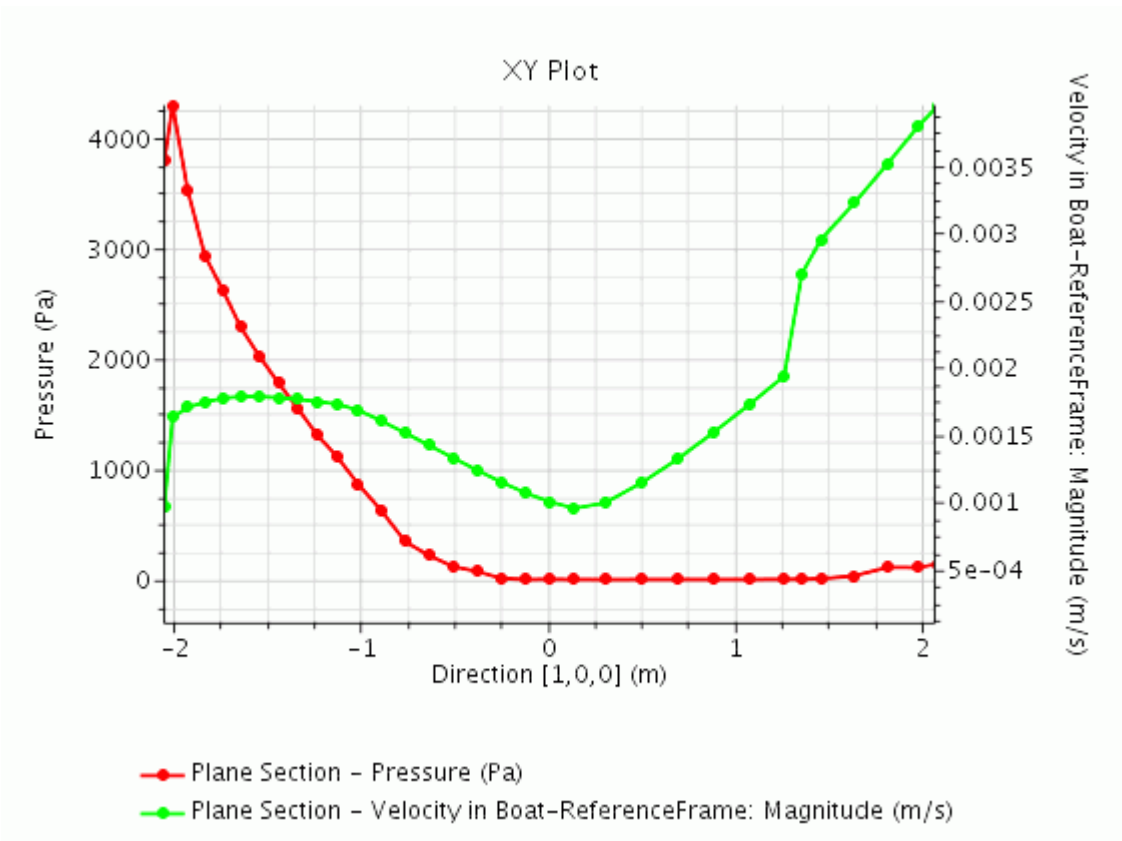
Plotting with Multiple Axes

STAR-CCM+ lets you add an axis to a plot to allow the comparison of multiple variables in a single chart display.

With multiple axes, you can compare correlations within STAR-CCM+. For example, you can view horsepower against RPM and torque against RPM on the same plot. Additionally, when you have quantities with widely different scales of values, such as meters and Pascals, it helps to plot them on separate axes to make the graphs readable.

To view two correlations in an XY plot:

1. Create the XY plot and select the parts from which to plot data.
2. Select the quantities for the axes that are present by default, at bottom and at left.
3. Right-click the **Axes** node and select **New Axis** > **Right**.
4. To create a second **Y Type** node, right-click the **Y Types** node and select **New**.
5. Select the new **Y Type** node and set *Y-Axis* to **Right Axis**.
6. Select the quantity for the **Y Type** node that you have mapped to the right axis.
7. To change the orientation of the axes, see Changing the Axis Orientation.
8. Customize the data points, ranges, and grid lines as you see fit.



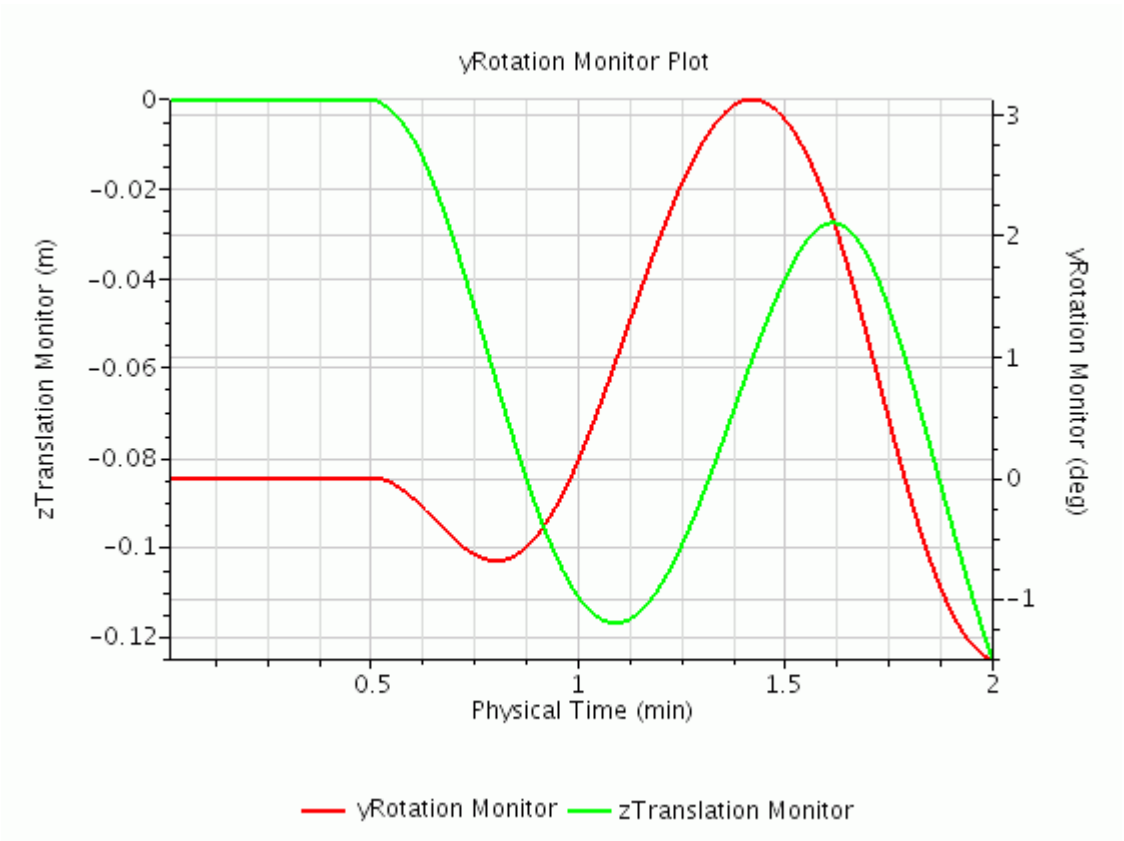
To use multiple axes with tabular data sets:

9. Add a table in the simulation. See Tables.
10. Add the table as a data set to the plot. See Adding and Removing Data in a Plot.
11. To display the tabular data on the axes of your choice, use the *X-Axis* and *Y-Axis* properties of the data set node.

To view multiple monitors, with widely differing ranges, in a single monitor plot:

12. Create the monitor plot -- either an empty one, or one from a report or a monitor.
13. Once the first quantity is plotted, right-click the **Axes** node and select **New Axis** > **Right**.
14. Create the second monitor.
15. Add the second monitor to the plot. See Adding and Removing Data in a Plot.
16. Select the newly added monitor data set node and set *Y-Axis* to **Right Axis**.

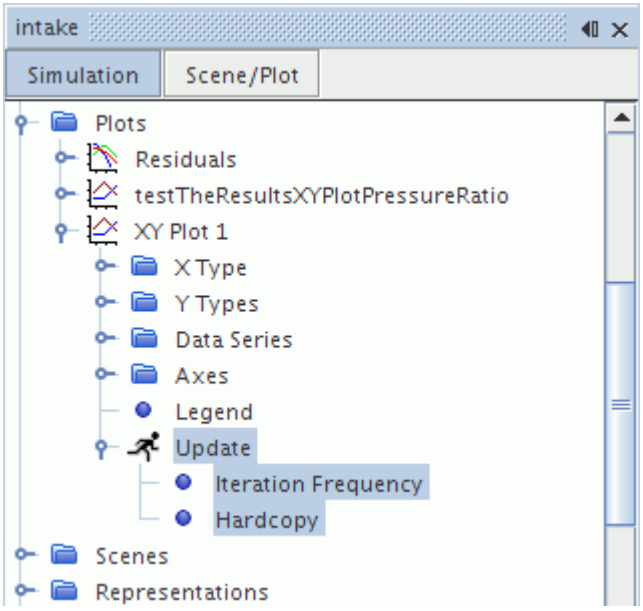
    You now have a single plot of the values from two separate monitors.



Updating Plots

You can control how open XY, monitor, and histogram plots update by changing the trigger type that is located in the **Update** node.

This node, which has its own properties and is subject to common menu items, customizes how plots in the display are updated as the simulation iterates.



The **Update** node has two sub-nodes:

- A trigger node to control the frequency of updates. The name of this node reflects the choice of trigger type. See Trigger Types and Trigger Properties.
- A **Hardcopy** node to control the specifications of the images that are generated by the plot updates. See Update and Hardcopy Properties.

For more information on selecting the trigger type and defining the update frequency, refer to Setting the Trigger for an Update.

Note that:

- The update mechanism is only active whilst the simulation is iterating.
- Multiple Y types within a XY plot displaying different scalar quantities will update simultaneously whenever an **Update Frequency** setting triggers an update. For example, the scalar quantities displayed in a Y types are updated to represent the new solution. Closed XY plots are not updated until they are opened.
- Hardcopies will only be written to disk at the intervals dictated by the **Update Frequency** setting for a particular plot.

The plot may also be refreshed manually. Expand the **Plots** node and right-click on the chosen plot. Choose **Refresh** from the pop-up menu.

You can also generate an image of the plot after each iteration or time-step to compile an animation of the plot. This is different from the single image generated when you save a hardcopy of the plot.

If all clients are disconnected from a running server, no plots will be saved to files.

Contents:

Update and Hardcopy Properties

Update and Hardcopy Properties

Update Properties

File Format	The graphics format that is used to save images of the visualization display. You have the same options as provided in the save plot to file action.
Enabled	Uses a checkbox with which you can activate or deactivate this feature.
Save to File	Determines whether or not multiple images of the plot display are saved as the solution progresses. This can be used to create an animation showing how the data changes while the simulation runs. STAR-CCM+ will bring the selected plot to the front when it produces a screenshot of that frame.
Output Directory	Path to where the files specified in the *Base Filename* property will be saved.
Base Filename	The prefix of the file name to use when saving plots to file. The full file name will be `prefixNNNNN.ext` where `prefix` is this property, `NNNNN` is a number to indicate the sequence of the files and `ext` is an extension based on the graphics format. If you leave the text box blank, STAR-CCM+ will restore the name that was there previously once that text box no longer has the focus.
Trigger	Determines how the display should be updated to reflect changes in the simulation. Change the trigger type to control the method that updates the plot. See Trigger Types.

Hardcopy Properties

The properties of the **Hardcopy** node let you control the specifications of the images generated by the plot updates. However, if all clients are disconnected from a running server, no graphics plots will be saved to files.

Output Magnification	Produces a larger image than the image in the original display, based on the factor you enter.
Offscreen Width	Sets the width of the image of the plot display when the save plot to file action is executed.
Offscreen Height	Sets the height of the image of the plot display when the save plot to file action is executed.

Use Current Resolution	Controls how the images are sized. **Note:** To avoid unintended variation in the resolution of the images during a simulation run, deactivate this option after you finish preparing the display.	
	Activated	Uses the latest plot display resolution, which changes as you resize the plot display. This setting is the default for plots that you create in the current version of STAR-CCM+.
	Deactivated	Uses the pixel resolution values that you set in the **Output Width** and **Output Height** properties. This setting is the default for plots in simulation files that were saved in STAR-CCM+ v11.02 or earlier.

Deleting Plots

Monitor, XY, and histogram plots can be deleted when no longer needed. The residual plot, which is created automatically, cannot be deleted.

Deleting a plot does not delete the underlying objects that are associated with the datasets of the plot. For instance, a monitor plot deletion will not impact the underlying monitor, which will continue to collect data.

Renaming Plots

Plots may be renamed as desired. Renaming the plot node will not change the displayed title of the plot.

Copying Plots

When a plot is copied, its properties and sub-nodes are copied. You can also copy the plot's information separately between simulation files, depending on whether the destination simulation has the same dependent objects.

For example, if you want to copy a plot that uses a file table, you need to follow these steps:

- Make sure that the destination simulation contains a table with the same presentation name.
- Copy and paste the plot.

Customizing Plots

The appearance of all plots can be modified in a number of ways.

Contents:

Changing the Title of the Plot
Changing the Data Display Order
Changing the Data Display Style
Customizing the Legend
Customizing Axes
Setting Preferred Attributes for Plots

See Also:

Plot Tab Reference

Changing the Title of the Plot

The title of a plot may be modified by changing the *Title* property.

The title text font may also be changed in the expert properties of the plot node. Use the property customizer button (...) to activate a dialog to set this property (changing the text will have no effect).

Changing the Data Display Order

If a plot display contains multiple data series, you can control the order in which each data series is displayed relative to the rest. This control is important for plots where data series overlap.

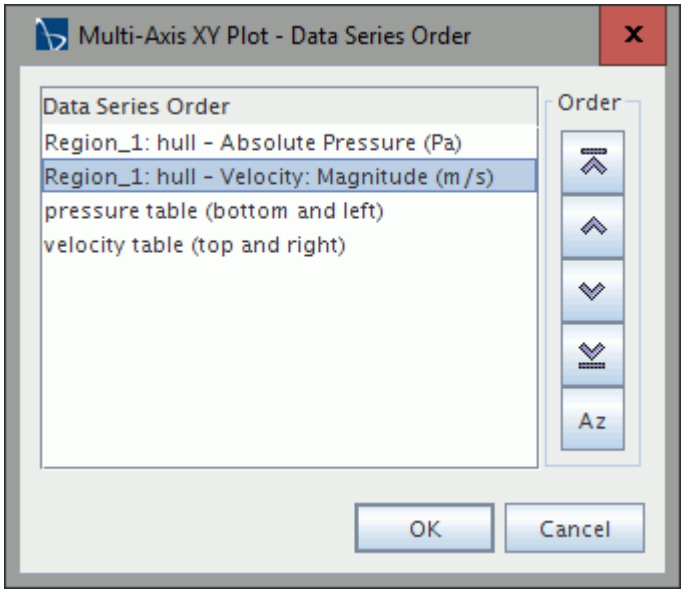
You can see the order of data series in three places in the workspace: the plot display, the legend, and the node tree (where applicable).

To change the order of data series in a plot, access the *Data Series Order* dialog:

1. Activate the dialog using one of the following techniques:
   - Right-click any of the following objects and select **Reorder Data Series**.
     - The node or display of any plot
     - The **Data Series** sub-node of a plot, or the corresponding data series in the display
     - The **Axis Type** sub-node of a histogram plot
     - The **[XY plot]** > **Y Types** > **[YType]** node
   - Click the right half of the *Data Series Order* expert property of the plot node, or click ... (**Custom Editor**).
2. Set the order in the dialog.

The names of the data series in the list match their names in the legend.

The data series at the top of the list is plotted on screen first, and the rest are successively plotted in front.



Select one or more items in the *Data Series Order* list, and use any of the buttons (top to bottom) as follows:

- **Move to top**: move series to the top of the list and render first
- **Move up**: move series up one position
- **Move down**: move series down one position
- **Move to bottom**: move series to the bottom of the list and render last
- **Alphabetize entire list**: reorders the items alphabetically.

3. Click **OK**.

Alternatively, move a data series by accessing its node.

4. Right-click the data series node.
5. Select one of the four options that appear in the sub-menu of the **Reorder** menu item:
   - **Move to top**
   - **Move up**
   - **Move down**
   - **Move to bottom**

Changing the Data Display Style

The style and color of data can be modified with the line style, symbol style, and fill style nodes.

Each style is associated with the source of the data: monitors, parts, or tables. Expand the **Plots node** and expand the chosen plot.

To reveal the **Line Style** and **Symbol Style** nodes for a monitor plot, expand the **Data Series** node and expand the chosen monitor.

To reveal the **Line Style** and **Symbol Style** nodes for an XY plot, do either of the following:

- Expand the **Y Types** node, the chosen Y Type node, and expand the chosen part.
- Expand the **Data Series** node, and expand the chosen data set node.

To reveal the **Line Style** and **Fill Style** nodes for a histogram plot, do either of the following:

- Expand the **Histogram** manager node and the histogram data set node.
- Expand the **Data Series** node, and expand the chosen data set node.

Contents:

Modifying the Line Style
Modifying the Symbol Style
Customizing the Appearance of the Histogram

Modifying the Line Style

The **Line Style** node, through its properties, lets you control the appearance of the line of a particular set of data.

The appearance of the line can be customized in several ways:

- The style of the line can be changed (none, solid, dashed).
- The color of the line can be changed.
- The width of the line can be changed.

To modify the color or style of the line click on the **Line Style** node. To change the color of the line either enter the RGB value of the color under the *Color* property in the *Properties* window or open the color selector by clicking on the ellipses button. To change the style of the line choose from the list of styles in the drop-down menu under the **Style** property in the *Properties* window. To change the width of the line modify the *Width* property.

Line Style Properties

Style	Specifies the line style.	
	None	No line displayed.
	(**Solid**)	Displays a solid line.
		Displays a dashed line.

	(**Dash**)	
	(**Dot**)	Displays a dotted line.
	(**Dash-Dot-Dash**)	Displays a line that alternates between dashes and dots.
	(**Dash-Dash-Dot**)	Displays a line that alternates between two dashes and one dot.
Width	Controls the line width in pixels.	
Color	Determines the color used to draw the line. The color can be input directly in RGB format (e.g., red would be entered as `255,0,0`). A comma must separate the numeric values. The property customizer executes the define color action.	

Modifying the Symbol Style

The **Symbol Style** node, through its properties, lets you control the appearance of the symbol of the points in a particular set of data.

The symbols can be customized in several ways:

- The shape of the symbol can be changed.
- The color of the symbol can be changed.
- The size of the symbol can be changed.
- The width of the stroke of the symbol can be changed.
- The spacing between symbols can be changed.

To modify the color or the shape of the symbol click on the **Symbol Style** node. To change the color of the symbol either enter the RGB value of the color under the *Color* property in the *Properties* window or open the color selector by clicking on the ellipses button. To change the shape of the symbol select from the drop-down menu under the *Shape* property in the *Properties* window. The size of the symbol can be changed by modifying the *Size* property.

Symbol Style Properties

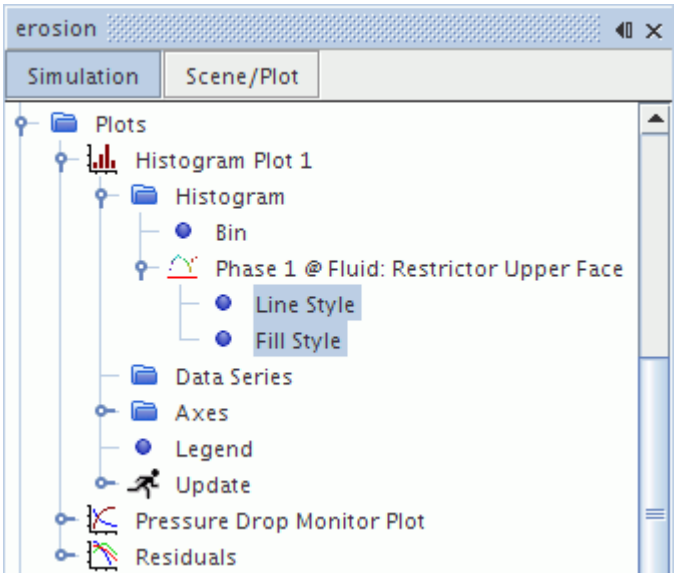
Shape	The shape of the symbol.	
	None	Makes the symbol invisible.

	Horizontal Line	Displays the symbol as a short horizontal line.
	Vertical Line	Displays the symbol as a short vertical line.
	Cross	Displays the symbol as a cross.
	Plus	Displays the symbol as a plus sign.
	Star	Displays the symbol as an asterisk type symbol (as opposed to a star shape).
	Filled Square	Displays the symbol as a filled square.
	Empty Square	Displays the symbol as a square with the outline only.
	Filled Circle	Displays the symbol as a filled circle.
	Empty Circle	Displays the symbol as a circle with the outline only.
	Filled Triangle	Displays the symbol as a filled triangle.
	Filled Diamond	Displays the symbol as a filled diamond.
Size	Sets the size of the symbol in pixels.	
Color	The color of the symbol. The color can be input directly in RGB format (e.g., red would be entered as `255,0,0`). A comma must separate the numeric values. The property customizer executes the define color action.	
Width	Sets the width of the stroke of the symbol in pixels.	
Spacing	Sets the interval between symbols.	

Customizing the Appearance of the Histogram

The histogram's appearance, as with other types of plots, is controlled in part by the Axes and Legend objects.

However, other characteristics are affected by the **Fill Style** and **Line Style** nodes, which are under the data node.



Fill Style Properties

Color	Determines the color of the bars. The color can be input directly in RGB format (e.g., red would be entered as `255,0,0`). A comma must separate the numeric values. The property customizer executes the define color action.
Secondary Color	Controls the color of the background. The property customizer executes the define color action.
Pattern	Selects a pattern with which to fill the bars.

For detailed information about the **Line Style** node, see Modifying the Line Style.

Customizing the Legend

The **Legend** node, with its properties, lets you change various aspects of the legend, including the position, orientation and visibility.

By default a legend is generated with the data point symbols and the names of the sources of the data whether that be parts, monitors or external data. The legend can be customized in several ways:

- It can be moved to different positions around the plot.
- The format, font and orientation of the text can be changed. (The text itself can be changed in the display using other objects.)
- It can be removed.

Expand the **Plots** node and expand the chosen plot node. Click on the **Legend** node. The legend location can be customized by selecting an option in the drop-down menu under the *Position* property in the *Properties* window. The legend may be located outside of the plotted area at eight points of the compass, or overlaid on the plot at a **Custom** location. To remove the legend deactivate the checkbox of the *Visible* property.

The orientation of the legend can either be horizontal or vertical and can be changed by selecting from the drop-down menu under the *Layout* property. It is also possible to control the number of columns or rows which affects the layout of the legend.

The format of the legend text can also be modified in the *Expert* properties window. The spacing around legend entries can be customized by modifying the *Item Spacing* property.

The legend text font may also be changed in the *Font* expert property. Use the property customizer button

(...) to activate a dialog to set this property (changing the text will have no effect).

Renaming the Plot Legend for a Data Set

The legend text that is associated with a dataset can be modified using the properties of that dataset. This may be useful if the default text happens to be fairly long.

Select the node of the dataset and enter the new text in the *Legend Name* property. Note that the *Lock Legend Name* property will automatically become activated.

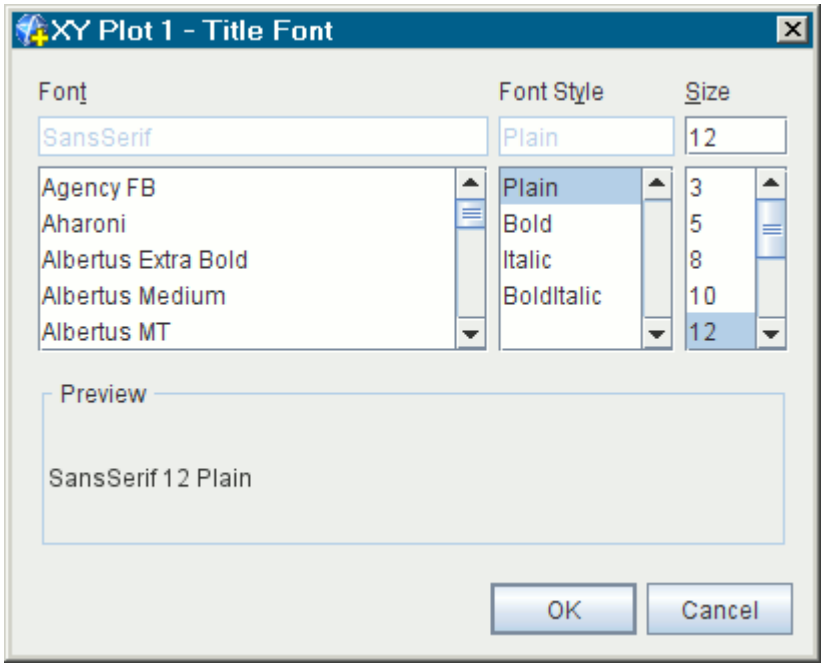
Your changes will appear in the plot display.

Legend Properties

Visible	Controls visibility of the legend.	
	Activated	Shows the legend.
	Deactivated	Hides the legend.
Position	Controls the position of the legend relative to the chart area:	
	North	Top center
	North East	Top right
	East	Middle right
	South East	Lower right
	South	Lower center
	South West	Lower left
	West	Middle left
	North West	Top left

	Custom	Puts the legend into a floating box that you can reposition anywhere in the display by clicking and dragging.
Layout	Controls the position of the legend objects relative to each other. This feature requires more than one data set to be represented in the plot display.	
	Horizontal	Displays the objects in a row.
	Vertical	Displays the objects in a column.

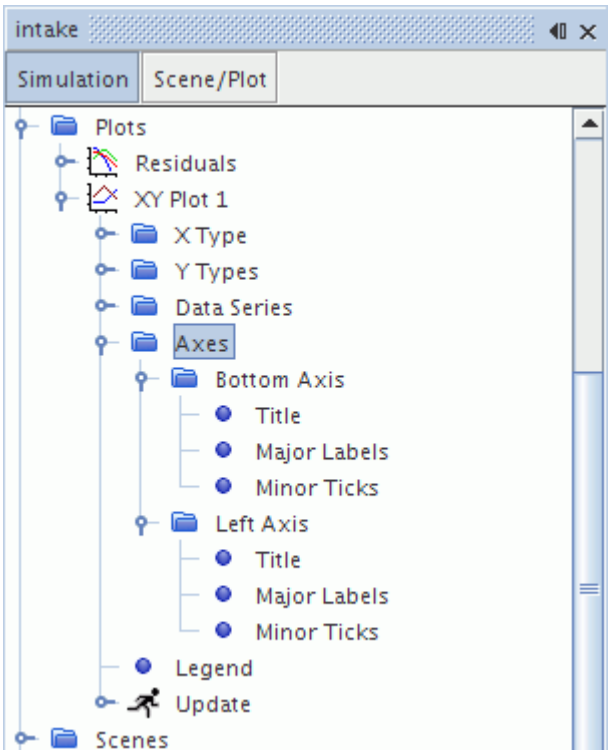
Expert Properties

Font	The font family, font style and font size. Use the property customizer button ( ) to activate a dialog to set this property (changing the text will have no effect).  The property value is displayed using the selected font family, style and size.
Item Spacing	Controls the spacing around items in the legend.
Number Of Rows	Specifies the number of rows for the legend.
Number Of Columns	Specifies the number of columns for the legend.

x-Position	Relative x-position on the screen, defined over the interval [0,1].
y-Position	Relative y-position on the screen, defined over the interval [0,1].

Customizing Axes

The **Axes** node contains several sub-nodes that let you customize the axes of the plot.



Customization is available in a variety of features.

Axes Properties

Axes	The number of axes in this plot (read only).
Preferred X-Axis	Selects the axis to be assigned to new data sets as the x-axis -- bottom axis by default.
Preferred Y-Axis	Selects the axis to be assigned to new data sets as the y-axis -- left axis by default.

Contents:

Changing the Axis Orientation
Setting the Scale
Customizing the Title on the Axis
Customizing the Major Labels on the Axis

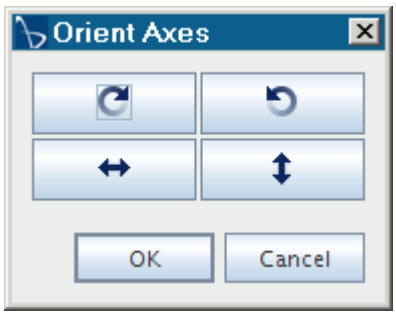
Customizing the Minor Ticks on the Axis
Hiding the Axes

Changing the Axis Orientation

The orientation of the axes determines the position and direction of increasing values for each axis.

The default is the axis origin to be based in the bottom left-hand corner of the plot with the x-axis along the base of the plot. To change the orientation of the axes:

1. Right-click the **Axes** node and select **Orient Axes**.
2. In the *Orient Axes* dialog, use any of the buttons to change the orientation of the axes:



- -- rotates the plot clockwise 90 degrees
- -- rotates the plot counterclockwise 90 degrees
- -- flips the plot horizontally
- -- flips the plot vertically

3. Click **OK**.

Setting the Scale

Each axis can be customized to have either a linear or logarithmic scale, or can be removed from the plot. It is also possible to focus on a region of the plot by customizing the minimum and maximum axis values.

Expand the **Axes** node and click on any of the individual axis nodes.

To change from the linear to logarithmic scale activate the checkbox of the *Logarithmic* property. To remove the axes from the plot deactivate the checkbox of the *Visible* property.

X Axis and Y Axis Properties

Note: Zero and negative values are not defined for logarithms, and the plot skips data at such points.

Visible	Determines whether the axis is visible. This property affects the sub-nodes--title, ticks and grid. The sub-nodes can be individually made visible or invisible.	
	Activated	Makes the title, ticks and grid visible.
	Deactivated	Hides the title, ticks and grid.
Logarithmic	Determines whether the axis is logarithmic or linear.	
	Activated	Makes the axis logarithmic.
	Deactivated	Makes the axis linear.

Expert Properties

Minimum	The minimum value shown on the axis.
Lock Minimum	When activated, locks the axis' minimum to the defined value.
Maximum	The maximum value shown on the axis.
Lock Maximum	When activated, locks the axis' maximum to the defined value.
Reverse	When activated, causes the scale and the corresponding plot points to appear in the reverse order. Use this feature as an alternative to flipping the entire plot, which changes the appearance of data on other axes as well.
Position	Changes the position of the axis on the chart.

Customizing the Title on the Axis

It is also possible to change the title of the axes by:

- Changing the text of the title
- Changing the font of the title
- Changing the orientation of the text
- Removing the axes title from the plot

To change the title of an axis:

- Open the **Axes** node and an axis node.
- Select the **Title** node.
- Modify the *Title* property.

The title can be taken out of the plot altogether by clearing the checkbox of the *Visible* property. The title may be rotated through 90 degree intervals by selecting the rotation angle from the drop-down menu under the *Rotation* property.

The axis title text font may also be changed in the *Font* expert property. Use the property customizer button (...) to activate a dialog to set this property (changing the text will have no effect).

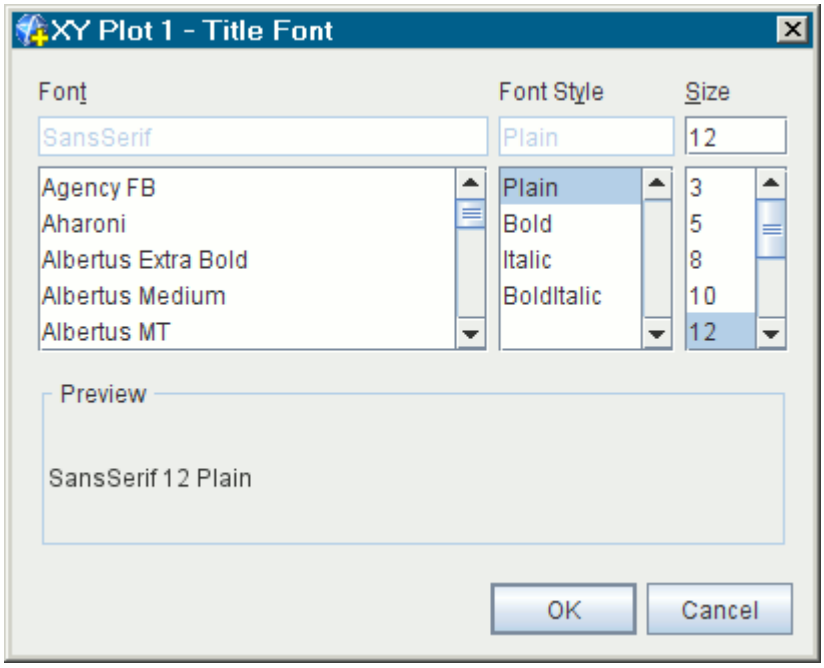
You can make this font change for multiple plots at the same time using the table editor dialog (in multi-edit mode).

Title Properties

Visible	Controls visibility of the title.	
	Activated	Shows the title.
	Deactivated	Hides the title.
Title	The text of the axis title. An empty string will hide the title. No additional space is gained.	
Lock Title Name	Controls whether to lock the title to the specified value.	
	Activated	Locks the specified title.
	Deactivated	An automated naming convention builds an axis title based on the datasets that are plotted.
Rotation	Selects an angle of rotation for the title.	
	0 degrees	Leaves the title at its normal position.
	90 degrees	Rotates the title 90 degrees clockwise.
	180 degrees	Rotates the title 180 degrees (upside down).
	270 degrees	Rotates the title 270 degrees clockwise (= 90 degrees counterclockwise).

Expert Properties

Font	The font family, font style and font size. Use the property customizer button (...) to activate a dialog to set this property (changing the text will have no effect).

	 The property value is displayed using the selected font family, style and size.

Customizing the Major Labels on the Axis

The labels, ticks, and grid associated with each axis can be customized.

Options for customization include:

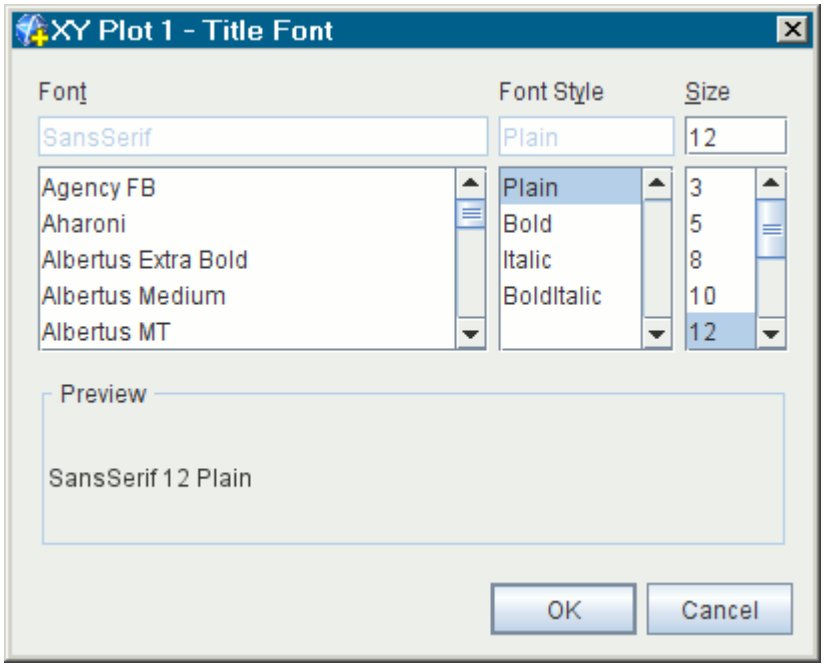
- Changing the spacing of the ticks to determine how many total ticks are displayed
- Changing the spacing of the labels to avoid over-crowding the axes
- Independent customization of grid lines associated with labeled (major) and non-labeled (minor) tick marks

To access the **Major Labels** node, open any axis node.

Major Labels Properties

Visible	Determines whether the labels are visible.	
	Activated	Makes the labels visible.
	Deactivated	Hides the labels.
Grid Visible	Determines whether the background grid lines of the labels are visible.	
	Activated	Makes the background grid visible.
	Deactivated	Hides the background grid.

Expert Properties

Font	The font family, font style and font size. Use the property customizer button (...) to activate a dialog to set this property (changing the text will have no effect).  The property value is displayed using the selected font family, style and size.	
Precision	Controls the maximum number of digits that appear after the decimal place in the axis labels.	
Auto Precision	Determines whether or not to automatically determine the axis label precision.	
	Activated	Automatically sets the precision.
	Deactivated	Uses the value set in the *Precision* property.
Pad Decimals	Determines whether or not to pad labels with zeroes to fill the precision of the labels.	
	Activated	Applies the label precision to annotations.
	Deactivated	Leaves annotations separate.
Spacing	Sets the spacing between axis labels in the units of the axis. For example, with a *Minimum* of `0.0` and a *Maximum* of `8.0`, a Label Spacing of `4.0` will place labels at 0.0, 4.0 and 8.0. Setting this value to `0.0` hides the tick labels.	

Auto Spacing	Determines whether or not to set the spacing automatically between axis labels.	
	Activated	Automatically sets the spacing.
	Deactivated	Uses the value set in *Label Spacing*.
Grid Style	Specifies the grid line style.	
	None	No line displayed.
	(**Solid**)	Displays a solid line.
	(**Dash**)	Displays a dashed line.
	(**Dot**)	Displays a dotted line.
	(**Dash-Dot-Dash**)	Displays a line that alternates between dashes and dots.
	(**Dash-Dash-Dot**)	Displays a line that alternates between two dashes and one dot.
Grid Color	Determines the color used to draw the grid line. The color can be input directly in RGB format (e.g., red would be entered as `255,0,0`). A comma must separate the numeric values. The property customizer executes the define color action.	
Grid Width	Controls the grid line width in pixels.	

Customizing the Minor Ticks on the Axis

The style and color of the minor ticks can be modified, or turned off altogether, using the properties of the **Minor Ticks** node.

The lines associated with each axis are modified independently of each other.

To access the **Minor Ticks** node, open any axis node.

In the *Properties* window, you can change the color, style and width of the line. If you do not wish to see the minor ticks then deactivate the checkbox of the *Visible* property. If you do not wish to see the grid lines then deactivate the checkbox of the *Grid Visible* property.

Contents:

Minor Ticks Properties

Minor Ticks Properties

Visible	Determines whether the minor ticks are visible.	
	Activated	Makes the minor ticks visible.
	Deactivated	Hides the minor ticks.
Grid Visible	Determines whether the background grid lines of the minor ticks are visible.	
	Activated	Makes the background grid visible.
	Deactivated	Hides the background grid.

Expert Properties

Count	Sets the number of minor ticks between the major labels.	
Grid Style	Specifies the grid line style.	
	None	No line displayed.
	(**Solid**)	Displays a solid line.
	(**Dash**)	Displays a dashed line.
	(**Dot**)	Displays a dotted line.
	(**Dash-Dot-Dash**)	Displays a line that alternates between dashes and dots.
		Displays a line that alternates between two dashes and one dot.

	(**Dash-Dash-Dot**)	
Grid Color	Determines the color used to draw the grid line. The color can be input directly in RGB format (e.g., red would be entered as `255,0,0`). A comma must separate the numeric values. The property customizer executes the define color action.	
Grid Width	Controls the grid line width in pixels.	

Hiding the Axes

To hide an axis, select its corresponding node and deactivate the checkbox of the *Visible* property in the *Properties* window.

Setting Preferred Attributes for Plots

You can create a preferred set of attributes for the plots in a simulation and apply them easily.

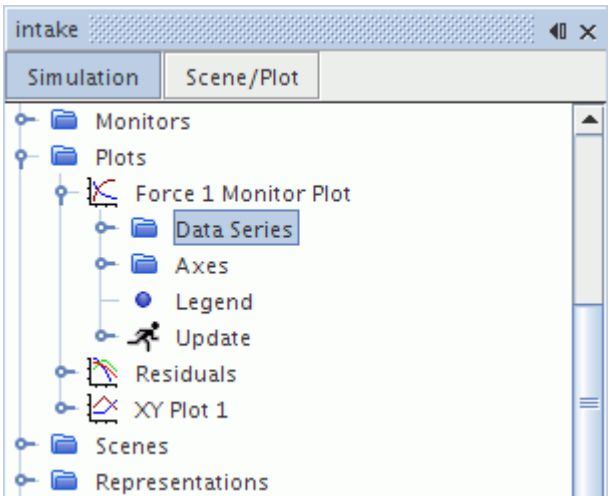
- To create a preferred set of plot attributes:
  - From the menu bar, select **Tools** > **Options**.
  - In the *Options* dialog, select the **STAR-CCM+** > **Plotting** node.
  - In the *Properties* window, specify attributes such as title fonts and plot line thickness.
- To apply your preferred attributes to a plot, right-click either on the plot node or in the plot display and select **Apply Plot Options**.
- To apply the attributes to multiple plots at once, multi-select the plot nodes, right-click, and select **Apply Plot Options**.
- To have a plot show more recent changes in plotting properties in the *Options* dialog, use the **Apply Plot Options** action again.

Note: STAR-CCM+ retains the attributes that you specify in the *Options* dialog, even in subsequent simulations. However, plots -- whether they are created before or after you set the attributes -- do not show those attributes until you apply them with the right-click action.

Adding and Removing Data in a Plot

You can make various types of simulation data available for plotting: monitors, tabular data, and derived data.

The **Data Series** manager node contains the objects representing the data sets that are used in the plot.



1. To add a data set, right-click on the **Data Series** node and select **Add Data**.
2. In the object selection dialog that appears, make your selections and click **OK**.

   Note: The objects that appear in the object selector dialog are of all available data objects in the simulation, not necessarily the ones that are used by the plot.

   XY and histogram plots can receive tabular data sets. Monitor plots, however, can receive monitors, tabular data sets, and derived data sets.
3. To customize the appearance of the data in the plot, See Changing the Data Display Style.
4. To adjust the order of appearance of data sets in a plot display, right-click on the **Data Series** node and select **Reorder Data Series**. See Changing the Data Display Order.

You can add multiple data sets at once with the drag-and-drop technique.

5. Multi-select the nodes in the object tree and drag the group onto the Data Series node.

You can also remove any data sets from a plot.

6. To remove a data set, right-click on its node and select **Delete**.

What Is a Tabular Data Set?

A tabular data set defines the source of data to be plotted and the style of plotting the data.

A tabular data set represents the contents of a table made available to a plot. This can be useful for plotting the data alone or against other data for comparison. However, you must have read a table into STAR-CCM+ beforehand. Once this is done, you can plot more than one set of table data in a plot, provided all the tables have identical column names. Simply follow the same steps for each table.

Contents:

Tabular Data Set Properties
Tabular Data Set Pop-Up Menu
Swapping Data Columns between Axes

Tabular Data Set Properties

Legend Name	Changes the legend text in the plot display.
Lock Legend Name	When activated, prevents any automatic changes to the legend name.
Table	Selects a table from among the ones represented by nodes in the **Tables** node of the **Tools** node.
Columns	Identifies the columns in the table file.
X-Column	Selects a column in the table to be plotted on the x-axis.
X-Units	Units of the plotted x-values.
X-Axis	Selects the x-axis for the table. The choice of axes depends on the total number of axes in the plot.
Y-Column	Selects a column in the table to be plotted on the y-axis.
Y-Units	Units of the plotted y-values.
Y-Axis	Selects the y-axis for the table. The choice of axes depends on the total number of axes in the plot.

Expert Properties

Sort Plot Data	Determines whether the data will be sorted--this ensures that the abscissa (horizontal axis) values run in an increasing order. This does not affect the rendered result for a scatter plot. However, in the case of a line chart, if the source data is not previously sorted, this affects how the data is plotted.	
	Activated	Sorts the data.
	Deactivated	Does not sort the data.

X Offset	A scalar offset that will cause all values of the x-axis data to be shifted by this amount.
Y Offset	A scalar offset that will cause all values of the y-axis data to be shifted by this amount.
X Scale Factor	A scaling factor that will cause all values of the x-axis data to be scaled by this amount.
Y Scale Factor	A scaling factor that will cause all values of the y-axis data to be scaled by this amount.

Tabular Data Set Pop-Up Menu

Reorder	Activates a submenu of four options for adjusting the order of appearance of this data set in a plot display. See Changing the Data Display Order.
Swap Columns	Switches the axis to which either column of data is assigned.
Tabulate	Executes the tabulate data action to view the data in a two-column spreadsheet-style view of the data in this series. The abscissa (horizontal axis) data will be in the first column and the ordinate (vertical axis) data will be in the second column. If there is no data in the series, the dialog will be blank.
Export...	Executes the export plot data action allowing you to save the series data in a .csv file.
Delete	Executes the remove data set action to remove this data set from the plot. It does not delete the associated file.

Swapping Data Columns between Axes

You can control which column of data is represented on a particular axis by using the object tree.

Open the **Plots** node, the individual plot (monitor or XY) node and the **Tabular** node. Right-click the node of the tabular object and select **Swap Columns**.

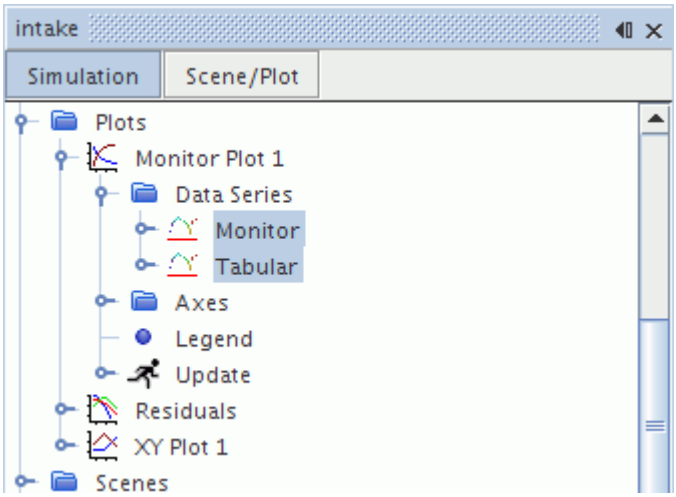
Assuming you have the plot display open, the effect will appear immediately.

Your next step involves changing the titles of the axes.

What Is a Derived Data Set?

A derived data set represents a monitor plot's access to the contents of a derived monitor data set or derived tabular data set within a data set function, such as a Fast Fourier Transform.

Its node in the object tree has its own properties and pop-up menu.



Contents:

Derived Data Set Properties
Derived Data Set Pop-Up Menu

Derived Data Set Properties

Legend Name	Changes the legend text in the plot display.
Lock Legend Name	When activated, prevents any automatic changes to the legend name.
X-Units	Units of the plotted x-values.
Y-Units	Units of the plotted y-values.
Input Data	The monitor or tabular data set function that is the source of this object's data. This read-only value reflects the selection in the **Data Series** manager node. The data set function linked to this data set cannot be deleted while a link is in place.
Percentage RMS Variation	Percentage variation in the sum of RMS from the original signal (read-only).

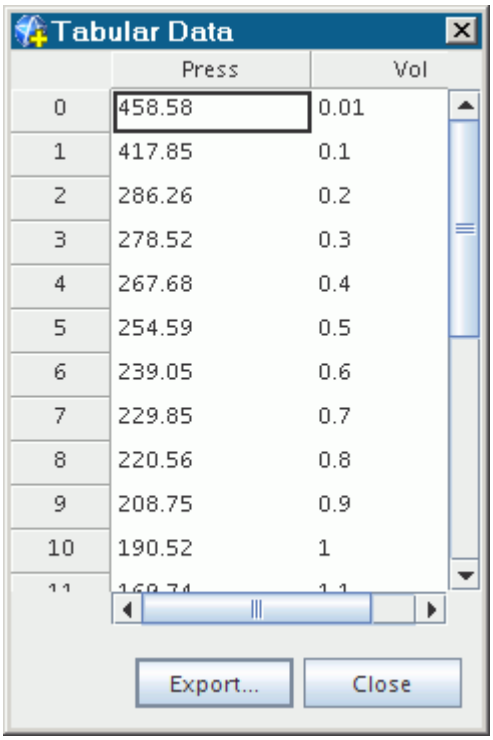
Derived Data Set Pop-Up Menu

Reorder	Activates a submenu of four options for adjusting the order of appearance of this data set in a plot display. See Changing the Data Display Order.
Tabulate	Executes the tabulate data action to view the data in a two-column spreadsheet-style view of the data in this data set. The abscissa (horizontal axis) data will be in the first column and the ordinate (vertical axis) data will be in the second column. If there is no data in the data set, the dialog will be empty.
Export...	Executes the export plot data action allowing you to save the series data in a `.csv` file.
Apply Plot Options	Sets preferred plot attributes.
Delete	Executes the remove data series action to delete the data from the plot. It does not delete the associated data set function.

Tabulating Data

To display the data from a plot in tabular form, right-click on the plot node and select **Tabulate...**

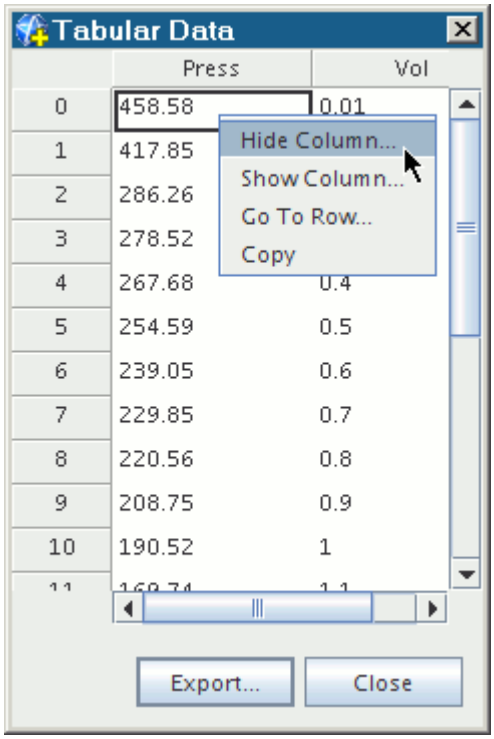
The data from the plot will be presented in the *Tabular Data* dialog, displaying the X and Y data values for one or more data series in a spreadsheet-like view.



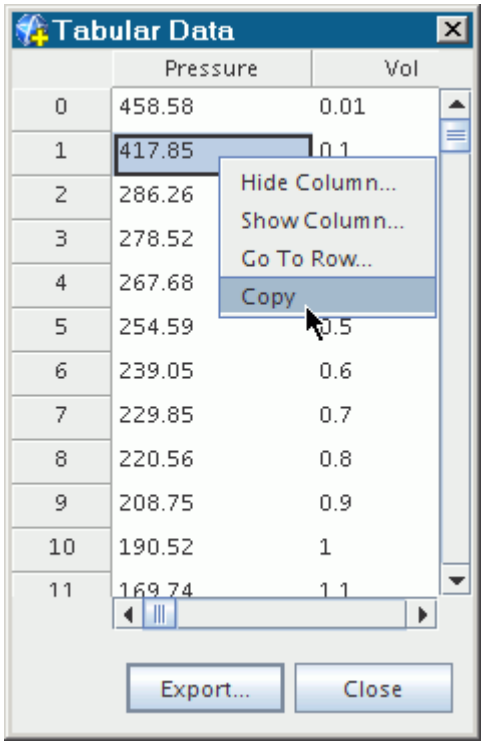
If this action is triggered from a plot, all data series in the plot will be displayed in pairs of X and Y value columns (as shown above). If triggered from a dataset, only the data contained by that object is shown.

Features of this dialog include:

- Horizontal and vertical scroll bars
- Column and row resizing by dragging when the mouse cursor changes to an east-west icon or a north-south icon over the column and row dividers
- Hiding columns, showing columns and jumping to a particular row number by right-clicking anywhere on the spreadsheet and selecting from the pop-up menu



- Copying a single value or set of selected values from the tabulated data to the clipboard -- one use for this is to copy a value to another object in the simulation tree. Specifically, select the particular value in the table. Right-click and select **Copy**.



Paste the value into the relevant constant in the simulation tree.

Exporting Data

To export data from one of the curves displayed in a plot, right-click on the dataset node corresponding to the data that you wish to export and select **Export...**

You can export all of a plot's data at once with the same type of operation by right-clicking on the node of that plot.

This action uses an operating system-specific standard *Save* dialog, via the save as action, to store the data points of the selected monitor or data set to file. The exported file is written as comma separated values (`.csv`)--a format supported by many spreadsheets.

The headers of the `.csv` file contain useful information for each column of data:

- For plot data, the header includes the scalar and the part name:

```
"PressureRatio-plane section 1"
```

- For monitor data, the header includes the monitor name and scalar:

```
"Heat Transfer 1 Monitor: Heat Transfer of Boundary Heat Transfer
(W)"
```

Printing and Saving Plot Images

Plot images can be saved as image files or printed.

To send a plot image to a printer connected to your system, select the plot node, then use the **File** > **Print** menu item to invoke the print action.

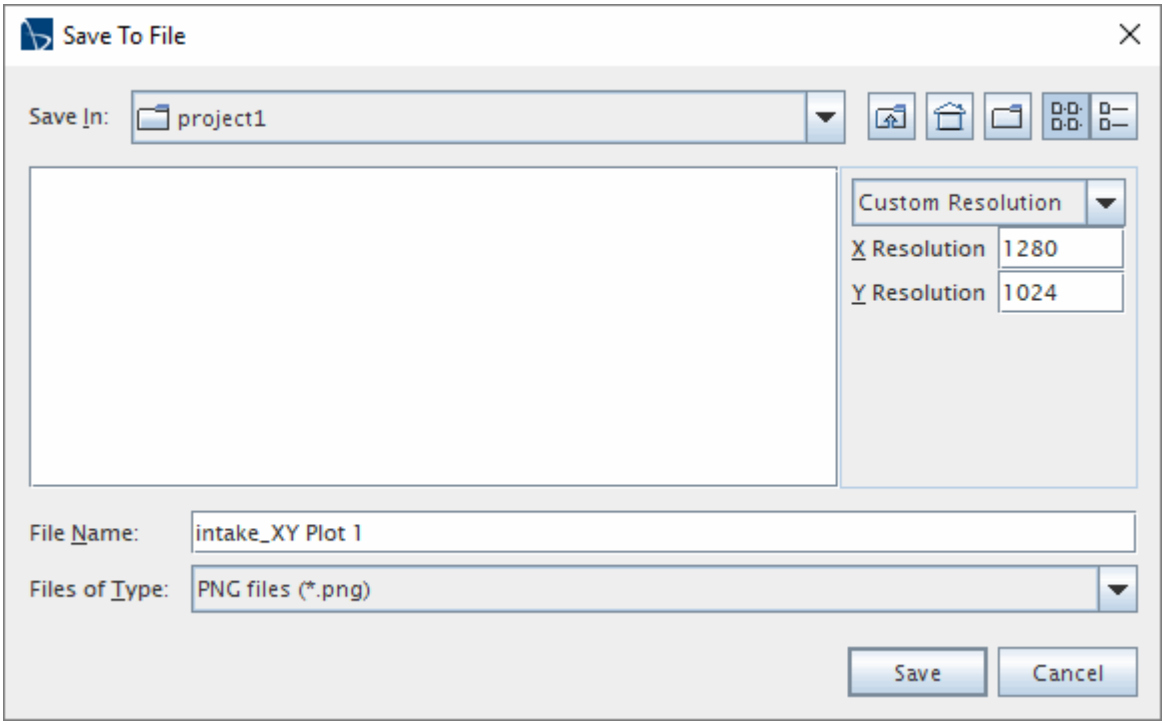
Contents:

Saving a Plot Image File

Saving a Plot Image File

To save a plot image to a file, right-click on the plot node and select **Hardcopy**.

A customized, operating system-specific *Save To File* dialog appears, allowing you to write the selected plot to file. The saved image can then be incorporated into a printed report, a slide presentation, a Web page or another commonly used medium.



To set specific values for the plot image resolution, in pixels, use one of the two available options in the drop-down list:

- **Locked Aspect Ratio** -- enter a value for the **X Resolution** or **Y Resolution**, and STAR-CCM+ computes the other one to preserve the aspect ratio.

- **Custom Resolution** -- enter any value up to `4096` for either resolution and the aspect ratio is ignored. This choice can produce unexpected viewports and cutoffs of images.

The image file can be written in any of the following formats:

- Portable network graphics (`.png`)
- Joint Photographic Experts Group (`.jpg`)
- PostScript (`.ps`)
- Portable document format (`.pdf`)
- Extended PostScript (`.eps`)

This action is disabled if more than one plot is selected.

Plot images can also be saved in batch mode. The procedure is similar to that for saving scene images in batch.

Plot File Formats Reference

This section describes external file formats for plotting.

Contents:

plt Format
xy Format

plt Format

This text-based file format can hold multiple data series against the same abscissa (horizontal axis) data in an array format.

There are four types of lines:

- A definition line containing four space separated fields:
  - The keyword ARRAY
  - The plot title (in single quotes)
  - A count of how many series are stored
  - The number of points per series
- The abscissa data as space separated fields holding the axis title followed by the number of data points specified in the first line (field 4)
- Multiple ordinate data lines (as specified in field 3 of the first line), each as space separated fields holding the series title followed by the number of data points specified in the first line (field 4)
- Optional comment lines beginning with the '#' character

An example of a file is shown below.

```
# The data is in array format
# It has one set of x-axis data, multiple sets of y-axis data
# It can have names for the data view and for the x and y series
# Each x point can have a point label
ARRAY 'Reference Plot' 1 3
'' -0.035750001668930054 -0.046149998903274536 -0.04582050070166588
'Lower Temperature' 371.477294921875 327.20318603515625 327.66650390625
```

xy Format

This format is a simple two column file with the abscissa (horizontal axis) data in the first column and the ordinate (vertical axis) data in the second column, with the columns separated by one or more spaces.

There should only be one data pair per line. An example of this file is shown below.

```
-4.035 0.8604
-3.386 0.8584
-2.487 0.84544
-2.017 0.81856
-1.695 0.78999
-0.937 0.67503
-0.761 0.64874
-0.455 0.5848
-0.288 0.56893
-0.11 0.56395
0 0.55086
```

Pop-Up Menu of the Plots Manager

These are the pop-up menu items available when you right-click the **Plots** manager node.

New Plot	Offers a choice of plot type to create.	
	Monitor	Creates a new monitor plot with a default name of `Monitor Plot N`, and adds a new plot tab in the *Graphics* window for that plot. The plot is created with the following defaults: • No data series • An x-axis label of `X Axis` • A y-axis label of `Y Axis` • A title of `Monitor Plot N`
	X-Y	Creates a new XY plot with a default name of `XY Plot N`, and adds a new plot tab to the *Graphics* window for that plot. The plot is created with the following defaults: • No data series • An x-axis label of `Position [1,0,0,] (m)` • A y-axis label of `<Select Function>` • A title of `XY Plot`
	Histogram	Creates a new histogram plot with a default name of `Histogram Plot N`, and adds a new plot tab to the *Graphics* window for that plot. The plot is created with the following defaults: • No data series • An x-axis label of `<Select Function>` • A y-axis label of `Frequency` • A title of `Histogram Plot`
Open All Plots	Displays all plots in the *Graphics* window.	

New Group	Facilitates the grouping of objects by creating a new group node within the manager node.
Group By	Allows you to group the objects according to any of the properties of the objects in the manager node, listed in a submenu of this menu item.
Ungroup	Reverses the grouping process.

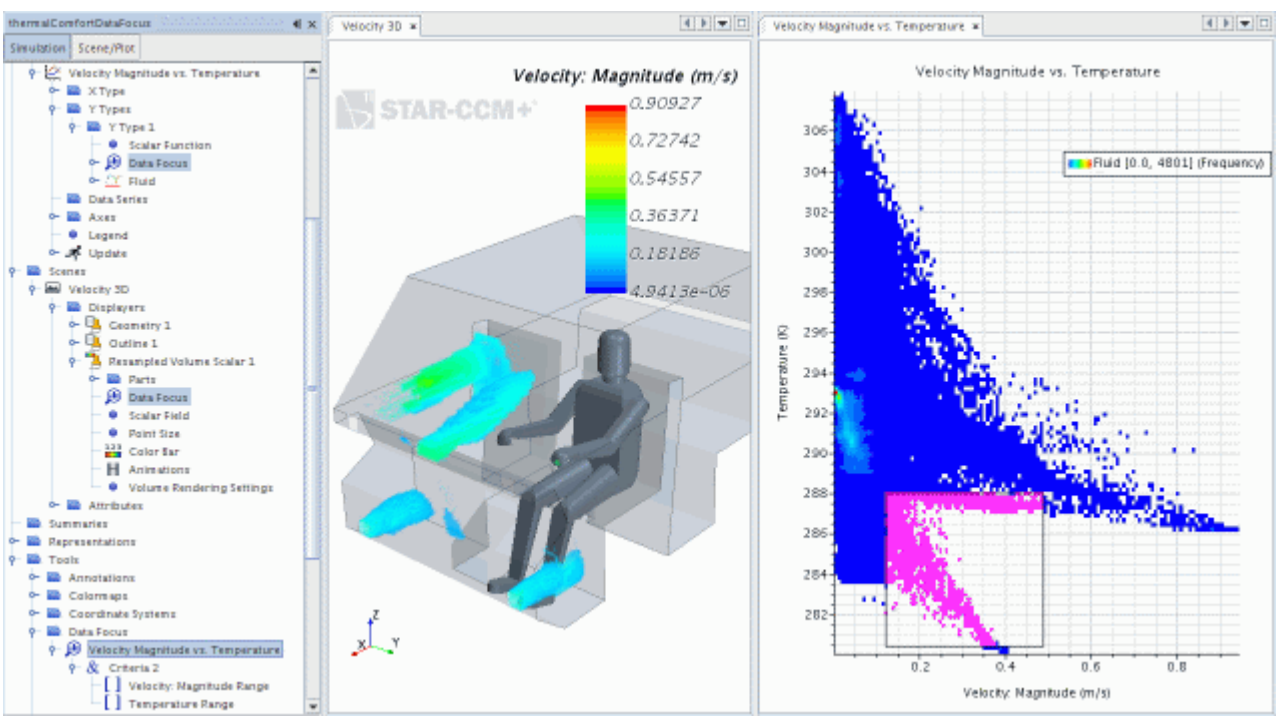
Exploring Solutions with Data Focus

This interactive tool lets you extract knowledge of merit -- information that facilitates a correct engineering decision -- from a large volume of data.

Data focus is a method to explore and interrogate data so that you can answer questions such as the following:

- How do I locate unburned fuel in areas with high local temperatures, where the fuel would be expected to burn?
- In a fairly complex combustion flow, what are the effects of local mixing (through turbulence or through the geometry of the device itself) on products and reactants?
- Contrary to expectations, are there areas with both a high residence time and a high flow speed? If so, where?
- Can bubbles promote better mixing by enhancing the local turbulence? If so, where?

The data focus technique in STAR-CCM+ is *linking and brushing*. After setting up plots and scenes, select one or more portions of a plot to define criteria of interest. To visualize the data that match your criteria, link the plots to scenes.



Contents:

Data Focus Workflow
Data Focus Objects Reference

Data Focus Workflow

Select portions of plots to isolate data of interest in one or more scenes.

Decide on the quantities that you want to examine, and then plot and visualize those quantities.

1. Determine the quantities.
2. Choose the parts. If none of the existing parts in the simulation are appropriate, create a derived part.
3. Set up plots for your correlations. The most useful type of plot to use with the data focus feature is a scatter plot.

   To view all the data in the plot at once, select the region as the part.

   In addition, you can use the heatmap style to view the concentration of data more clearly. See Using Heatmaps.

   - If you plan to use multiple correlations in the simulation, consider renaming each plot according to its quantities, for example `Velocity Magnitude vs. Temperature`.
4. Set up scenes to visualize your quantities.
   - If you create a vector scene, you can include line integral convolution. See Working with Line Integral Convolution.
   - To visualize data in 3D, consider using a resampled volume derived part. If your graphics card does not support volume rendering, activate advanced rendering for the relevant scene. See Defining a Resampled Volume.

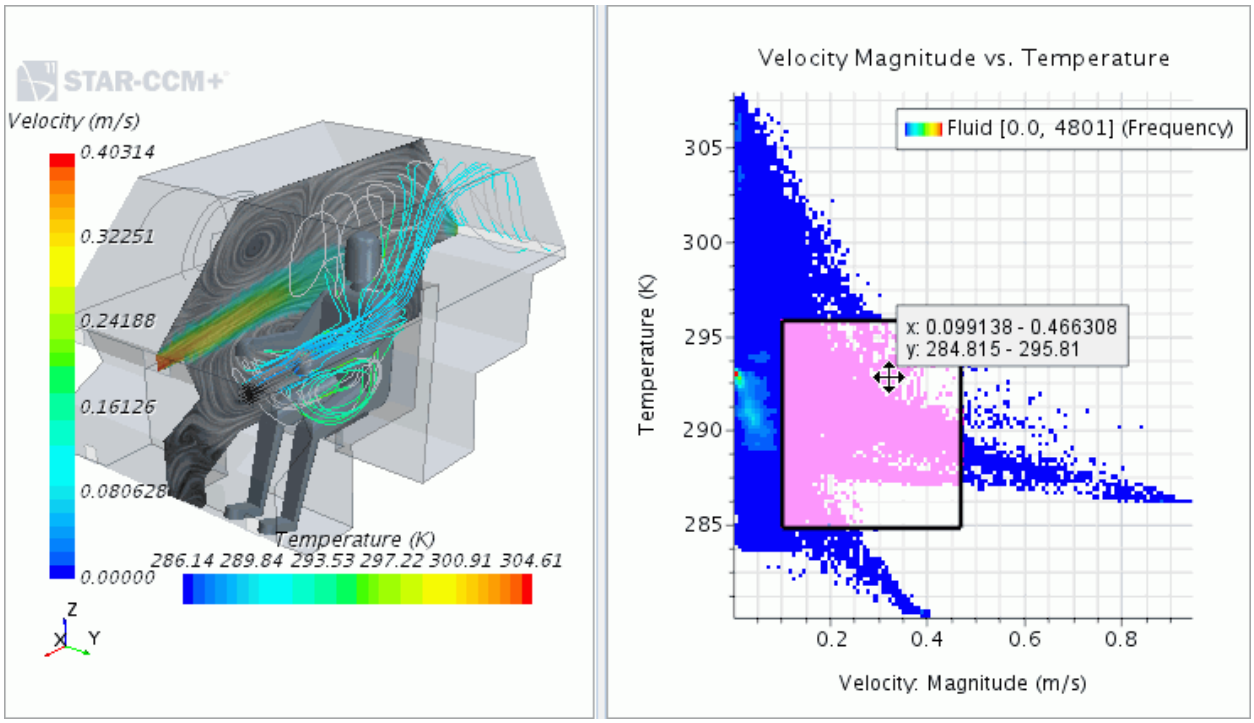
Prepare the data focus and the displays.

5. Right-click the **Tools** > **Data Focus** node and select **New**.
   - Although common practice involves interacting with plots and scenes, you can use the object tree nodes of the data focus to set up all its specifications, similar to a search query:
     1. After creating the data focus object, decide whether to set the *Operator* property of the **Criteria** node, a compound object that contains multiple ranges, to **AND** or **OR**.
     2. Right-click the node to add a range or another compound object.
     3. For each compound object that you add, specify **AND** or **OR**.
     4. Add as many ranges as you wish.
     5. For each range, specify the field function in the *Function* property, as well as values for *Minimum* and *Maximum*.
   - If you plan to use multiple correlations in the simulation, consider renaming the new data focus object according to its quantities, for example `Velocity Magnitude vs. Temperature`.
6. To apply a data focus to plots and scenes, use one of the following techniques:
   - Drag and drop the data focus node into the scene or plot. In the scene, select the part displayer to which you are applying the data focus.
   - Work with the object tree:
     1. Expand the node of each plot that you want to link, along with its **Y Types** sub-node.
     2. Select the appropriate Y type node and select the data focus object in the drop-down list of the *Data Focus* property.

3. Expand the scene node and its **Displayers** sub-node.
4. Select the appropriate displayer node and select the data focus object in the drop-down list of the *Data Focus* property.

- Work with the data focus object:
  1. Expand the **Tools** > **Data Focus** node, right-click the data focus object node, and select **Apply To**.
  2. In the *Apply Data Focus* object selector, expand the nodes of plots and scenes that you want to link.
  3. Select the appropriate Y type nodes for the plots, and the appropriate displayer nodes for the scenes.

7. To work with a linked plot and scene in the *Graphics* window, arrange them so that they are side by side.
8. Select the plot. In the **Plot** toolbar, click (**Create Data Focus Selection**), select **Add Selection to Data Focus**, and draw a box across the plot.
9. Move the box, the "brush", to examine portions of the data.



10. In addition to moving the brush, use other techniques for changing the data focus:
    - Add brushes to the plot.
    - Modify the extent of the brushes using any of the following techniques:
      - Drag the borders of the brush.
      - Right-click the brush, and do any of the following:
        - Add or remove constraints by selecting a range in the submenu.
        - Select **Edit Constraints**, select a range in the submenu, and enter values in the *Edit Range* dialog.

- Edit the properties of the value range sub-nodes within the data focus object node. See Data Focus Objects Reference.

Note: The data focus selection may appear outside the defined brush when you apply it to a heatmap. This additional area results from the bounds of the brush partially selecting heatmap bins in the plot. Increase the resolution of the heatmap (values of the *Bin* property) to reduce the area of the selection outside the data focus brush.

Data Focus Objects Reference

As an alternative to interacting with the plot, you can specify filter criteria in the properties of the sub-nodes of the **Tools** > **Data Focus** node.

Specific nodes referenced here are:

Data Focus (n)

A filter that filters data according to various restrictions. The restrictions, which consist of criteria and ranges, are accessible in the form of sub-nodes of this node.

Criteria (n)

Criteria to define data that the data focus presents. To choose AND (all objects in the set) or OR (any combination of objects in the set), select the corresponding item in the drop-down list of the *Operator* property of this node.

[Value] Range

See [Value] Range.

[Value] Range

This node specifies the scalar function, and its range of values, for the criteria. The node takes the name of the selected function. For example, with the selection of **Absolute Pressure**, the range node is named **Absolute Pressure Range**.

[Value] Range Properties

Function	Selects the type of scalar field function or field function monitor.
Minimum	The value above which data is focused.
Maximum	The value below which data is focused.

Triggers and Update Events

Several features of STAR-CCM+ require that you define a trigger for when an action takes place. STAR-CCM+ provides three built-in trigger types, and one custom trigger type.

The custom trigger type is defined by an update event.

Features for which you set triggers are as follows:

- Solution Histories—control how often the solution writes to the simulation history file.
- Scenes—control how often the scene updates with the calculated solution.
- Plots—control how often the plot updates with the calculated solution.
- Tables—control how often the table updates with the calculated solution
- Auto Exports—control how often the solution data exports to external files.
- Auto Saves—control how often the simulation file saves.
- Monitors—control how often the monitor extracts new data from the calculated solution.

Contents:

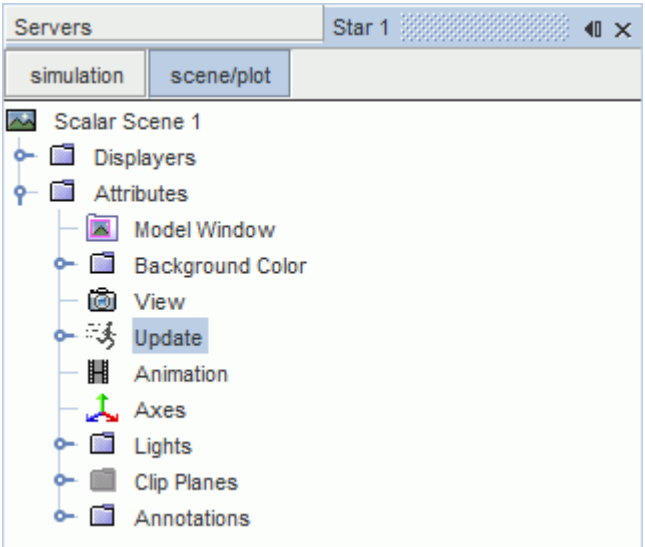
Setting the Trigger for an Update
Trigger Types
Trigger Properties
User-Defined Update Events

Setting the Trigger for an Update

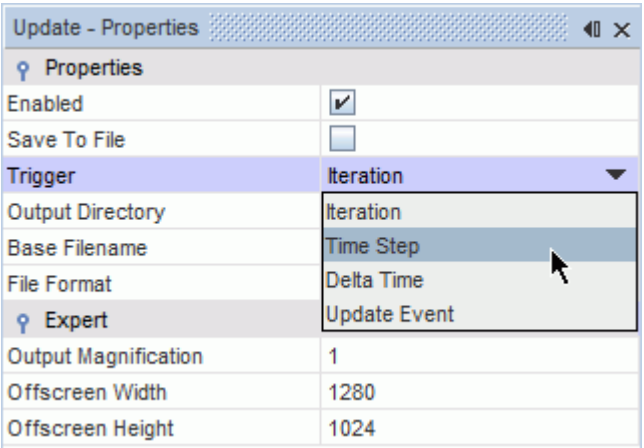
In general, features that require you to define a trigger expose the **Update** node. Use the properties of this node, and any child nodes, to define when an update action occurs.

To set the trigger type:

1. Navigate to the **Update** node (this example shows the **Update** node in a scene).



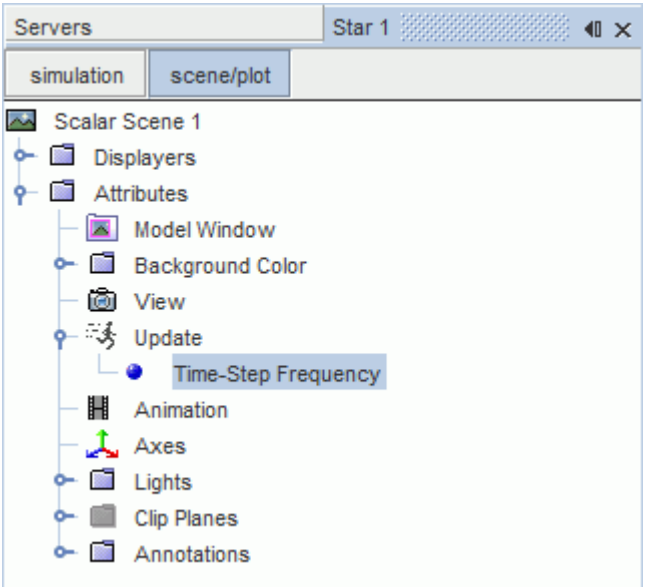
2. In the *Properties* window, select the trigger type that you want from the *Trigger* property. See Trigger Types.



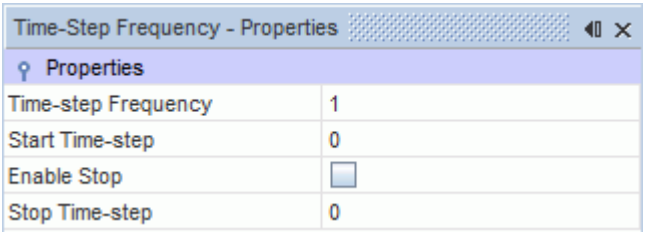
Note: Changing a monitor trigger during (or after) a simulation run does not result in the clearing of the previously collected data.

The child node changes according to the trigger type.

3. Select the child node.



4. Set the trigger properties. See Trigger Properties.



Note: When you attempt to multi-edit objects with properties that differ extensively, the **Edit...** context menu item is unavailable (grayed out). This restriction includes the editing of monitors, plots, and scenes. To multi-edit monitors, plots, or scenes, multi-select only the ones of the same type.

Trigger Types

There are four methods of triggering an update event:

Trigger Type	Description
Iteration	Triggers an update every n iterations. Specify the update frequency in the *Number of Iterations* property field.
Time Step	Triggers an update every n time-steps, where the *Number of Time Steps* property specifies the update frequency. This trigger type is only appropriate for transient simulations.
Delta Time	Triggers an update when the simulation time advances by the amount you specify in the *Delta Time* property field. This trigger type is only appropriate for transient simulations.
Update Event	Triggers an update based on a user-defined update event. User-defined update events are based on monitor values and provide you with more flexibility in controlling the updates. See User-Defined Update Events.

Note: For each trigger type, apart from **Update Event**, you can control the start, end, and update interval in the properties panel of the child node. See Trigger Properties. For **Update Event**, you can specify these settings in the relevant update event, located under the **Tools** > **Update Event** node. See User-Defined Update Events.

Trigger Properties

Triggers control when an update event occurs.

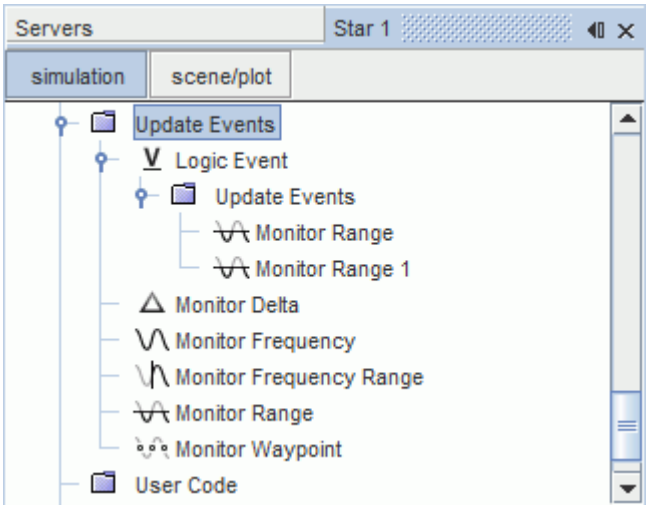
The properties for the various trigger types are listed below:

[Trigger] Frequency	Specifies the frequency of the update event. (Built-in trigger types).
Start [Trigger]	Specifies the point after which the update event starts updating. (Built-in trigger types).
Enable Stop	Specifies whether the *Stop [Trigger]* property is active or not. (Built-in trigger types).
Stop [Trigger]	Specifies from what point the update event stops updating. (Built-in trigger types).
Update Event	Specifies which user-defined update event to use for the selected object. This property only appears for the **Update Event** trigger.

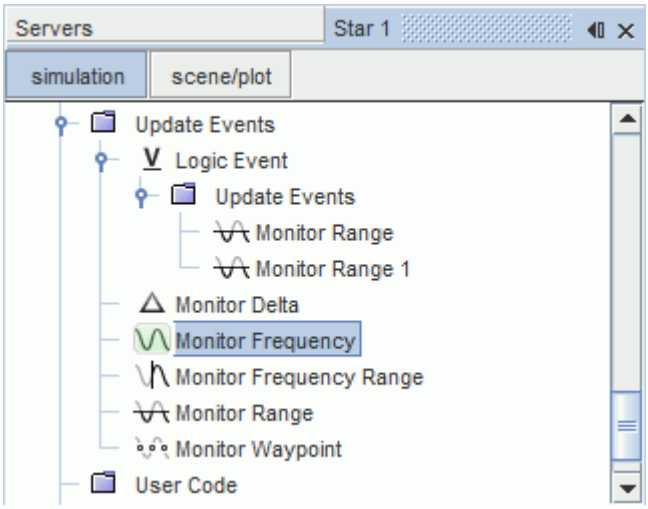
User-Defined Update Events

User-defined update events are custom triggers that provide you with more flexibility compared to the built-in trigger types. With a user-defined update event, you can trigger an update based on monitor values.

You create and manage user-defined update events under the **Tools** > **Update Events** node.



When the conditions for a user-defined update event are satisfied, the event triggers an update and the icon next to the corresponding update event flashes green.



Caution: Data mappers can cause inconsistent reporting if you use them with field function monitors or with update events other than iteration, time-step, or delta time.

Contents:

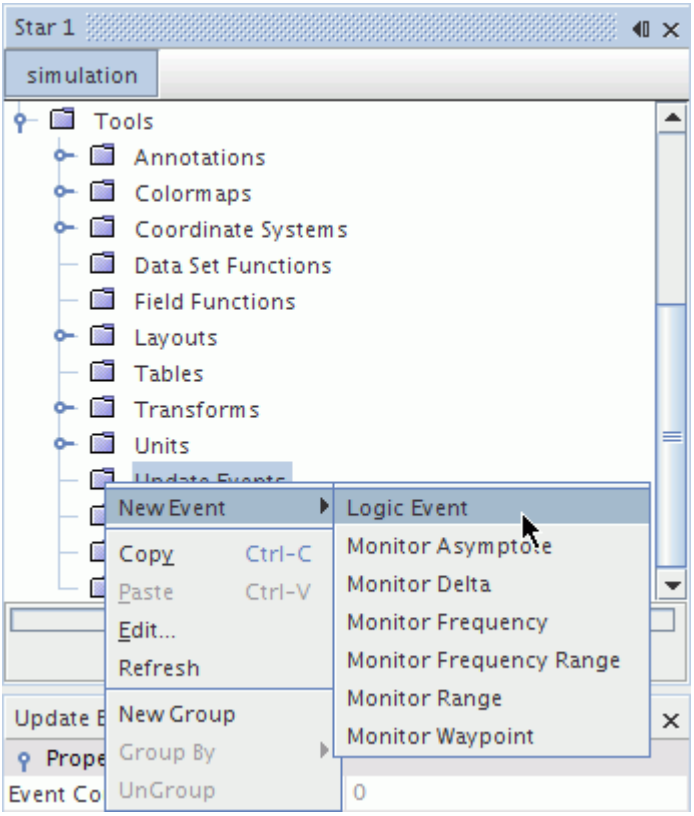
Creating User-Defined Update Events

Creating User-Defined Update Events

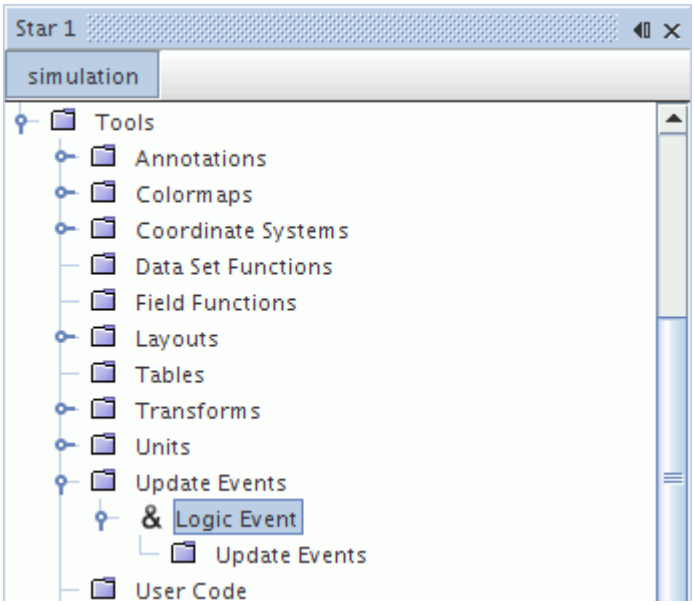
You can create user-defined update events using various update event parameters.

To create a user-defined update event:

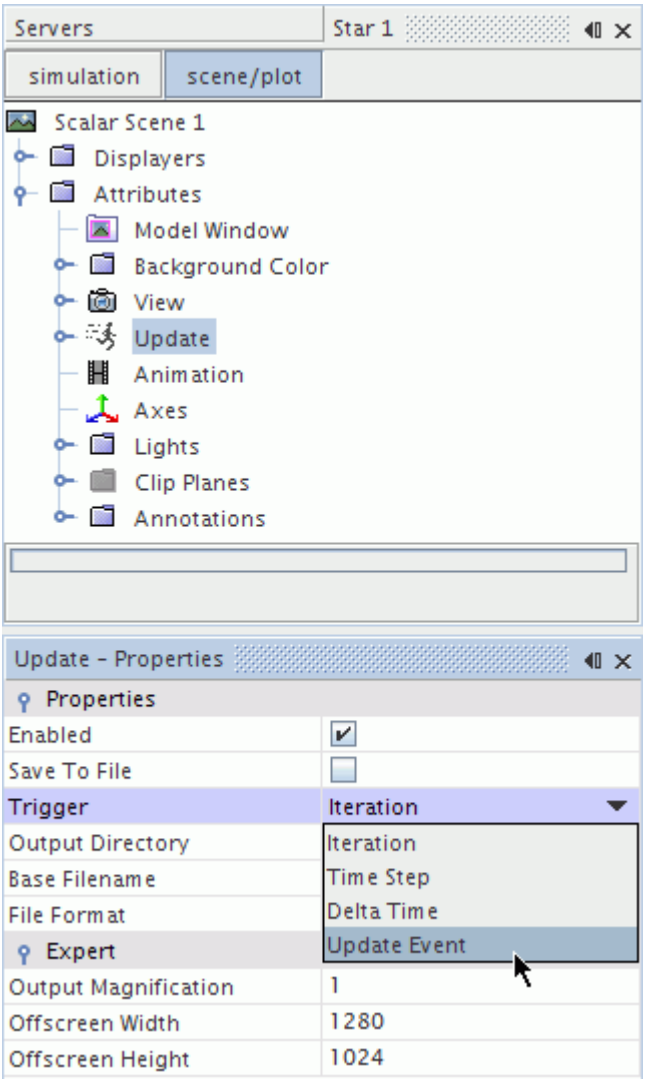
1. Right-click **Tools** > **Update Events** and select **New Event** > **[Update_Event_Type]**.



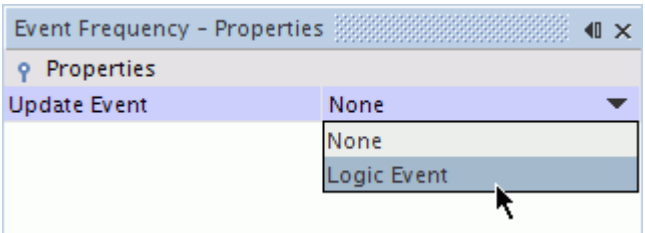
A new event node appears.



2. Set up the update event parameters. Refer to the individual update event documentation for more information.
   - The following update events are available for all types of simulations:
     - Logic Event
     - Monitor Asymptote
     - Monitor Delta
     - Monitor Frequency
     - Monitor Range
     - Monitor Frequency Range
     - Monitor Waypoint
   - If you perform a DFBI simulation that sets the **Equilibrium** motion option, you can define the following update event:
     - DFBI Equilibrium Forces Converged
3. In the feature of your choice, navigate to the **Update** node, and set *Trigger* to **Update Event**. In this example, the **Update** node of a scalar scene is selected.



4. Select **Update** > **Event Frequency** and set *Update Event* to the update event you created, in this case **Logic Event**.



Contents:

Logic Event
Monitor Asymptote
Monitor Delta
Monitor Frequency

Monitor Range
Monitor Frequency Range
Monitor Waypoint
DFBI Equilibrium Forces Converged Event

Logic Event

The logic event allows you to apply logic arguments to multiple update events. Three logical expressions are available: AND, OR, and XOR.

Logic Event Properties

Enabled	Specifies whether the update event is active or not.	
Satisfied	Indicates whether the update event has been satisfied. This property is read only.	
Iteration	Displays the iteration at which the event was last evaluated. This property is read only.	
Logic Operator	Specifies which logic argument to use for the event.	
	AND	All of the events that you specify must be satisfied for the logic to be true.
	OR	Any one event must be satisfied for the logic to be true. If all events are satisfied, the logic is also true.
	XOR	Any odd number of events must be satisfied for the logic to be true.

Contents:

Creating a Logic Event

Creating a Logic Event

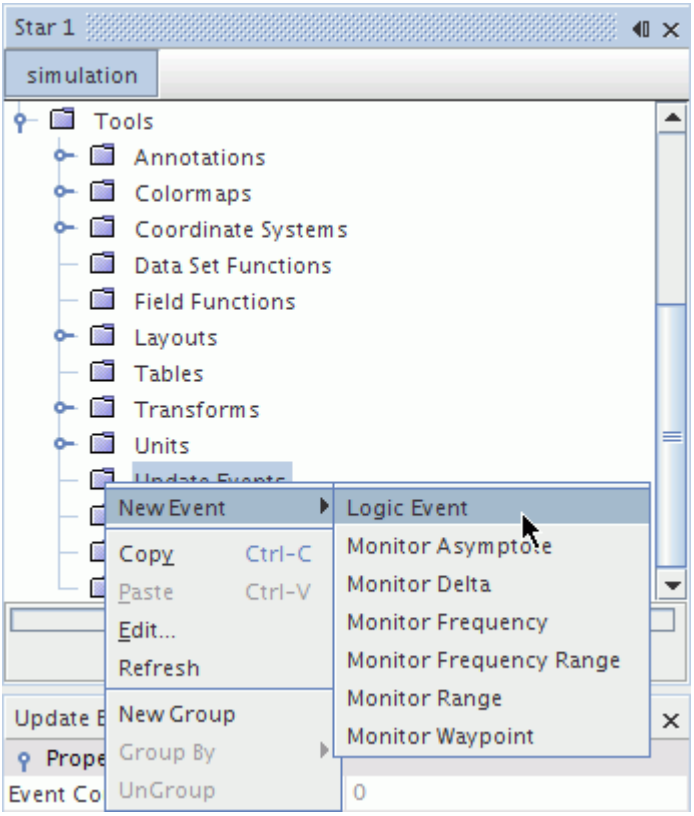
Logic events require multiple update events to function.

In the following example, a simple OR logic event is set up to trigger when one or all of the following update events is satisfied:

- the simulation time is 0.5 seconds
- at every ten iterations
- the user-defined report value (pressure) is greater than or equal to 10 Pascal.

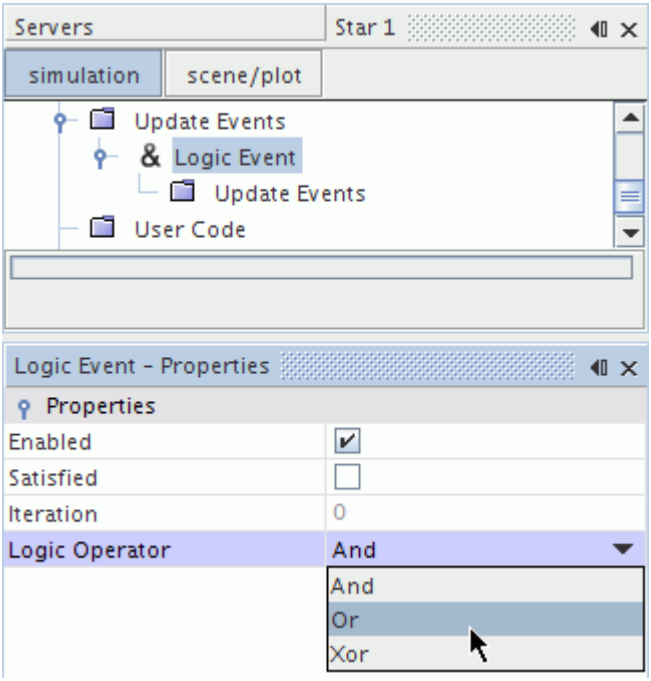
To set up the logic event:

1. Right-click the **Tools** > **Update Events** node and select **Logic Event**.

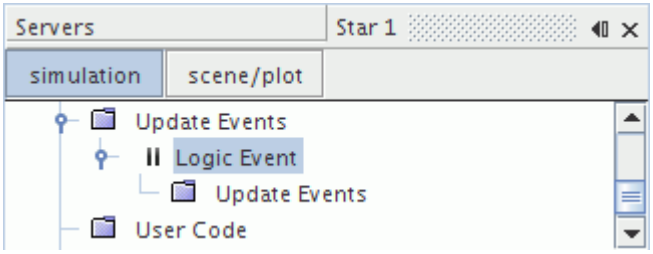


A logic event is added to the **Update Events** node.

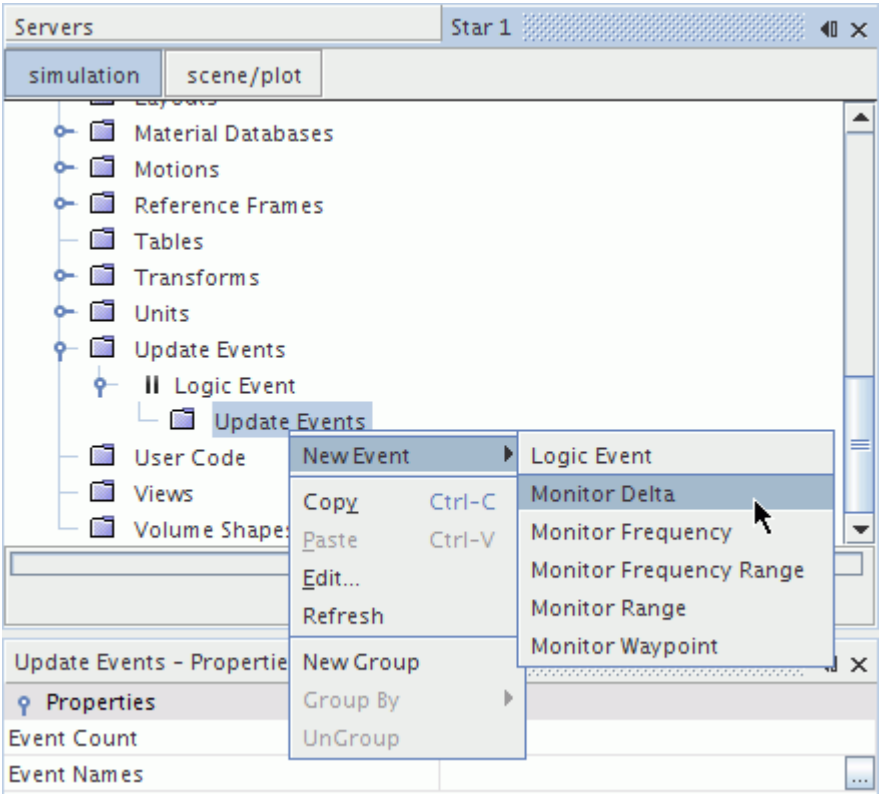
2. Select the logic event node and set *Logic Operator* to **Or**.



The icon next to the logic event changes to reflect the new logic.

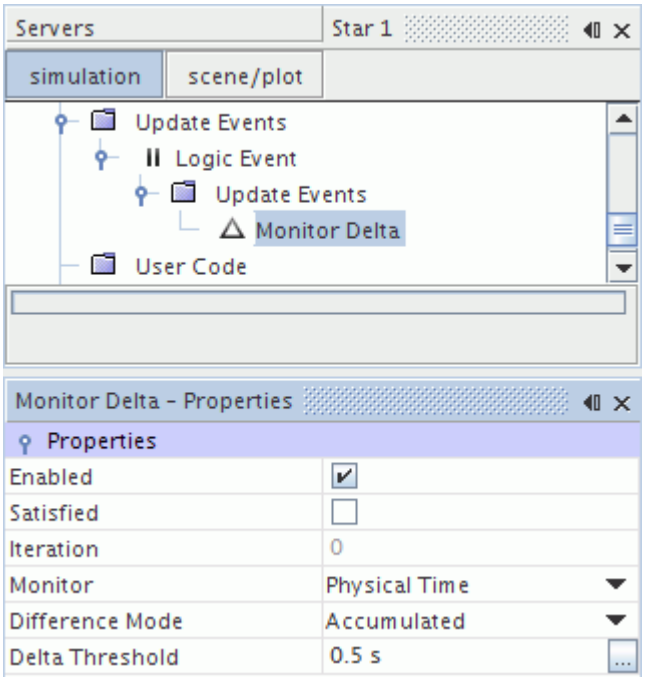


3. Right-click **Logic Event** > **Update Events** and select **New Event** > **Monitor Delta**.

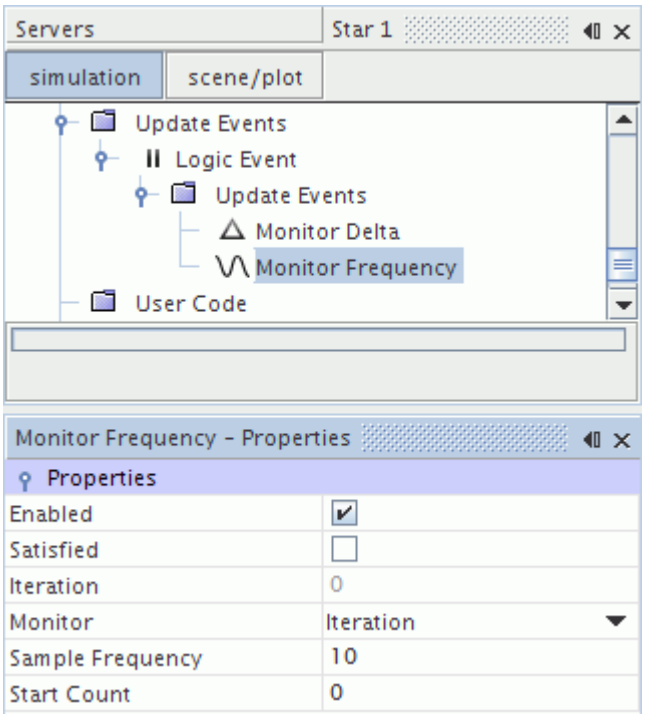


A monitor delta event is added to the logic event node.

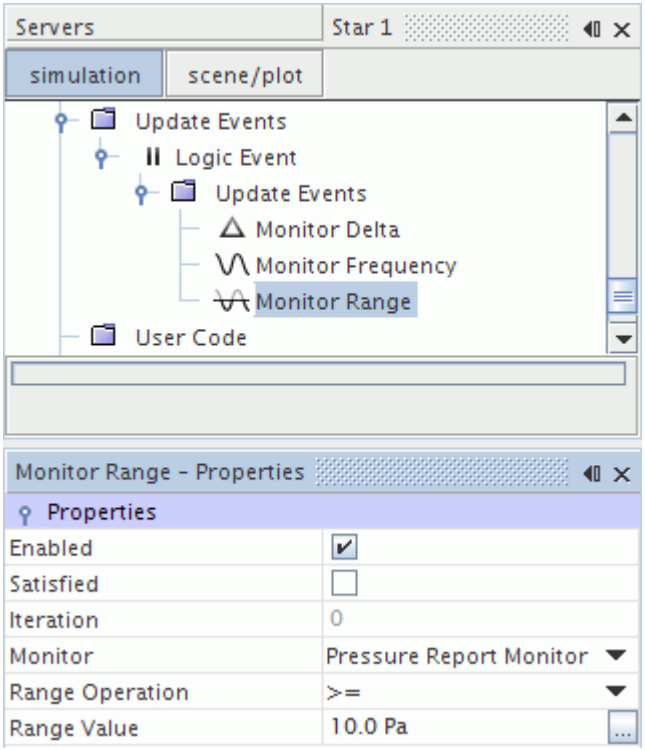
4. Select the **Monitor Delta** node and set:
   - *Monitor* to **Physical Time**
   - *Delta Threshold* to `0.5 s`



5. Using the same technique, create a **Monitor Frequency** event and set:
   - *Monitor* to **Iteration**
   - *Sample Frequency* to `10`

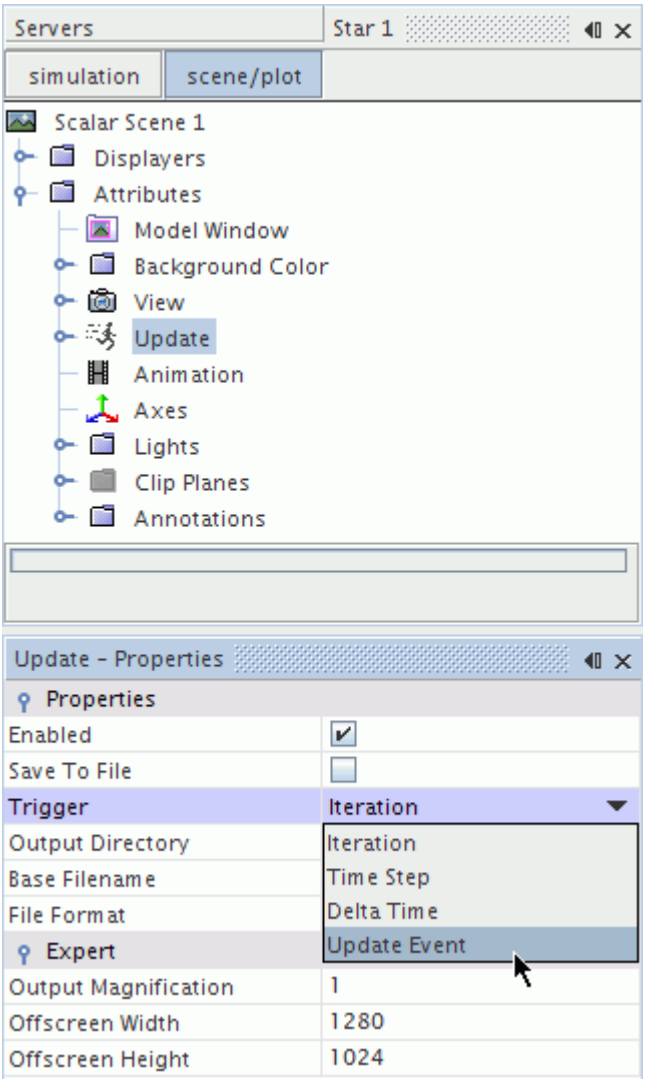


6. Using the same technique, create a **Monitor Range** and set:
   - *Monitor* to [**Pressure Report**]
   - *Range Value* to `10 Pa`

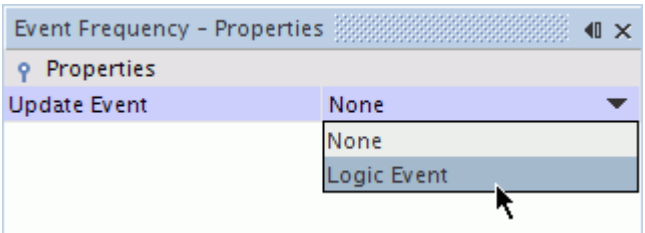


The logic event is ready to be applied.

7. In the feature of your choice, navigate to the Update node, and set *Trigger* to **Update Event**. In this example, the **Update** node of a scalar scene is selected.



8. Select **Update** > **Event Frequency** and set *Update Event* to the update event you created, in this case **Logic Event**.



Monitor Asymptote

When the calculations reach a specified threshold, the monitor asymptote event triggers an update. This threshold is passed when the maximum difference among a set of samples drops below a user-specified value.

The set is a user-specified number of the most recent samples.

This feature is useful when the exact trigger value is unknown, but you do know the maximum change in the monitored value that you consider sufficient for an update.

Monitor Asymptote Properties

Enabled	Specifies whether the update event is active or not.	
Satisfied	Indicates whether the update event has been satisfied. This property is read only.	
Iteration	Displays the iteration at which the event was last evaluated. This property is read only.	
Monitor	Specifies the monitor for this update event.	
Delta Threshold	The threshold value that triggers the update event.	
Sample Count	The number of most recent samples that apply to the asymptotic calculation.	
Inner Samples	Controls whether only the iterations from the current time-step are observed. Leave this option deactivated if your monitored data transcends time-steps.	
	Activated	Disregards the values that are collected during the previous time-steps.
	Deactivated	Considers the values that are collected during the previous time-steps.

Monitor Delta

The monitor delta event observes the change in values that are collected by a monitor.

For example, observing the physical time monitor results in the event being updated every `n` seconds.

Monitor Delta Properties

Enabled	Specifies whether the update event is active or not.
Satisfied	Indicates whether the update event has been satisfied. This property is read only.
Iteration	Displays the iteration at which the event was last evaluated. This property is read only.
Monitor	Specifies which monitor to use for this update event.
Difference Mode	Specifies how the difference (or delta) is calculated.

	Accumulated	The difference is calculated as the accumulation of differences between each consecutive pair of collected monitor values. When the delta is exceeded, the difference between the accumulated delta and the threshold is carried over to the next accumulation.
	Non- Accumulated	The difference is calculated between the current monitor value and the value at the last satisfied iteration. This is an absolute difference, ignoring the intermediate values. When the delta is exceeded, the last satisfied value is updated, resulting in a new comparison point from which the next delta is evaluated.
Delta Threshold	The threshold value that triggers the update event.	

Monitor Frequency

The monitor frequency event observes the number of samples collected by a monitor.

For example, observing the physical time monitor results in the event being updated when the monitor collects a number of samples that is divisible by `f`, where `f` is a frequency count.

Monitor Frequency Properties

Enabled	Specifies whether the update event is active or not.
Satisfied	Indicates whether the update event has been satisfied. This property is read only.
Iteration	Displays the iteration at which the event was last evaluated. This property is read only.
Monitor	Specifies which monitor to use for this update event.
Sample Frequency	Specifies the number of samples that trigger the event.
Start Count	Specifies the number of samples that are required before the event triggers.

Monitor Range

The monitor range event updates when the monitor value is within the specified range.

There are two types of ranges:

- Greater or equal to the specified monitor value.
- Less than or equal to the specified monitor value.

Monitor Range Properties

Enabled	Specifies whether the update event is active or not.	
Satisfied	Indicates whether the update event has been satisfied. This property is read only.	
Iteration	Displays the iteration at which the event was last evaluated. This property is read only.	
Monitor	Specifies which monitor to use for this update event.	
Range Operation	Specifies the range for the update event.	
	>=	The event is satisfied when the monitor value is greater or equal to the `Range Value`.
	<=	The event is satisfied when the monitor value is less than or equal to the `Range Value`.
Range Value	Specifies the value that is used in the *Range Operation*.	

Monitor Frequency Range

The monitor frequency range event triggers an update every time that the number of samples collected by a monitor are within a specified range.

For example, observing the physical time monitor results in the event triggering an update every time-step until the number of time-steps exceeds the limit set by the frequency range.

Monitor Frequency Range Properties

Enabled	Specifies whether the update event is active or not.	
Satisfied	Indicates whether the update event has been satisfied. This property is read only.	
Iteration	Displays the iteration at which the event was last evaluated. This property is read only.	
Monitor	Specifies which monitor to use for this update event.	
Range Operation	Specifies the comparison operator made in determining the range of satisfiable values.	
	>=	The event triggers an update when the monitor collects `n` or more samples, where `n` is the `Range Value`.

	<=	The event triggers an update when the monitor collects `n` samples or less, where `n` is the `Range Value`.
Range Value	Specifies the value that is used in the *Range Operation*.	
Start Count	Specifies the number of samples that are required before the event triggers.	

Monitor Waypoint

The monitor waypoint event assumes that the monitor is sampling a continuous function over time, which could be periodic.

The waypoint event triggers an update when the values from two consecutive samples cross the `Waypoint Value`. That is, for two consecutive samples, `S1` and `S2`, the monitor waypoint event trigger when:

- `S1` is less than the `Waypoint Value` and `S2` is greater than the `Waypoint Value`.

or

- `S1` is greater than the `Waypoint Value` and `S2` is less than the `Waypoint Value`.

Monitor Waypoint Properties

Enabled	Specifies whether the update event is active or not.
Satisfied	Indicates whether the update event has been satisfied. This property is read only.
Iteration	Displays the iteration at which the event was last evaluated. This property is read only.
Monitor	Specifies which monitor to use for this update event.
Waypoint Value	Specifies at which monitor value the event updates.

DFBI Equilibrium Forces Converged Event

The **DFBI Equilibrium Forces Converged** event is available only when a 6-DOF body is associated with **Equilibrium** motion. The update event is created under **Tools** > **Events**. This event enables you to monitor the converged force or moment values that act on a rigid body. The event triggers an update when the forces and moments are converged after a motion step. Typically, a new motion step is performed afterwards.

DFBI Equilibrium Forces Converged Properties

Enabled	Specifies whether the update event is active or not.
Satisfied	Indicates whether the update event has been satisfied. This property is read only.

Iteration	Displays the iteration at which the event was last evaluated. This property is read only.
Body	Selects the rigid body that has the **Equilibrium** motion option set.

See Also:

Equilibrium

Using Data Set Functions

Data set functions apply signal processing to sets of wave data.

There are two kinds of data set functions:

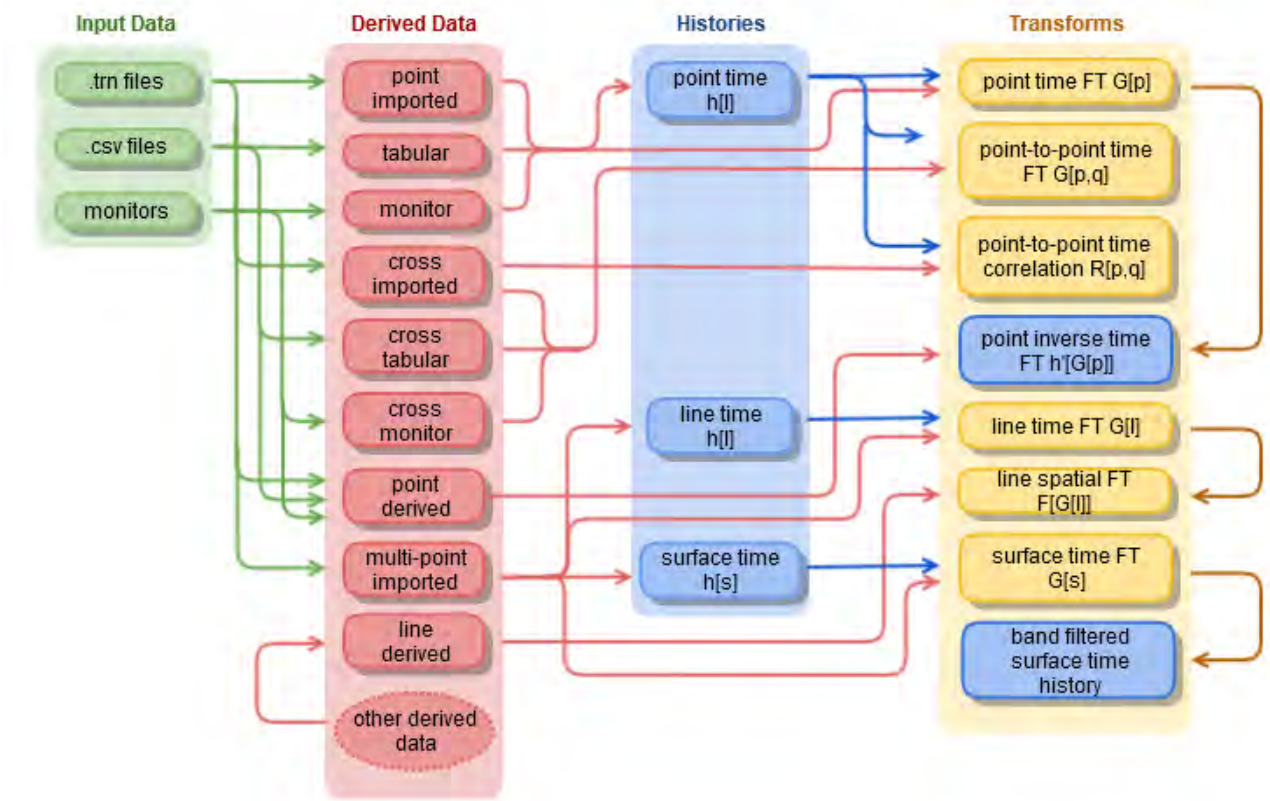
- Histories, which collect data at specified locations that are organized into points, lines, or surfaces.
- Transforms, which are either Fourier transforms or inverse Fourier transforms. Fourier transforms derive frequency data from signal data. Inverse Fourier transforms derive signal data from frequency data.

Data set functions depend on derived data sets, which you specify for each function. Derived data, in turn, depends on input data from the simulation.

In general, the flow of information is:

1. Input data from the simulation is organized into derived data sets. Different forms of data sets use different types of input data.
2. Derived data sets supply information to data set functions - histories and transforms. Different functions take information from different data sets.
3. Histories supply information to transforms. Different transforms use information from different types of histories.
4. Some transforms supply information to others, the inverse transforms, some of which are reconstructed histories.

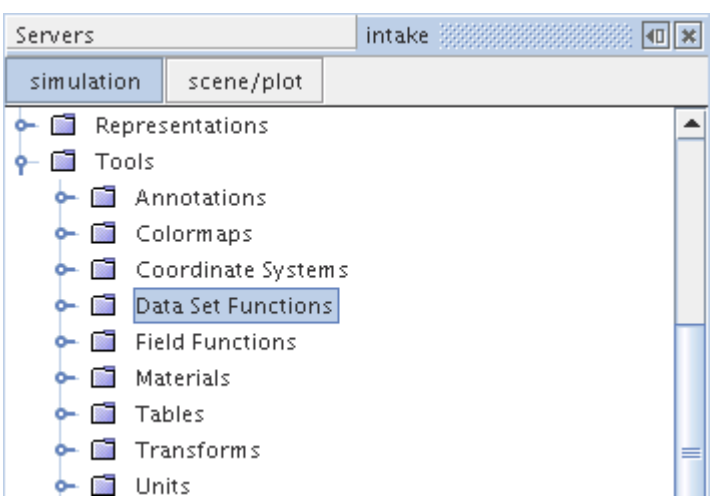
The details of information flow are shown in the following chart:



For signal processing in particular, any signal can be represented either in the time or the frequency domain.

These representations are equivalent, that is, $S[T] \equiv S[F]$. Fourier transforms are used to transform a signal from the time domain to the frequency domain, and inverse Fourier transforms are used to do the reverse. Transforming a signal to the frequency domain makes it possible to identify the dominant modes in the signal.

Data set functions are used to apply signal processing to sets of data, which can come from STAR-CCM+ simulations or experiments. The **Data Set Functions** node, which has a pop-up menu, is the manager object for working with data set functions in the simulation tree. The data set functions manager can hold any number of data set functions up to the memory limits of the computer.



Signal processing functions can be used in two contexts in STAR-CCM+:

- They can be applied to data stored in monitors, such as would be obtained from a report of a particular value at a point. The results of this analysis of data at a point can be visualized in monitor plots.
- They can be applied to data stored on selected surfaces at specified time intervals in the `.trn` file. The results of this analysis of data on a surface can be visualized in scenes.

One example of using data set functions for frequency analysis would be a study in which the goal was to reduce the noise that a room fan generates. The initial part of the study would involve a validation exercise, in which you would take experimental measurements of pressure against time at a location set some distance from the fan. These measurements would be used in a frequency analysis to identify the dominant modes for this signal and the frequency bands containing these modes. Large modes within the range of frequency audible to human hearing would suggest a noisy environment. Following the experimental analysis, a transient simulation would be run for the same configuration, with a monitor being set at the same location as the pressure probe used in the experiment. Frequency analysis would be applied to the monitored data to obtain the dominant modes predicted by the simulation. Comparing the predicted modes to the measured modes gives an understanding of the uncertainty that is associated with the simulation. If uncertainty is within acceptable levels, further simulations could be performed for varying configurations of the fan. These simulations would investigate ways of reducing the dominant modes, or at least moving them out of the range of human hearing.

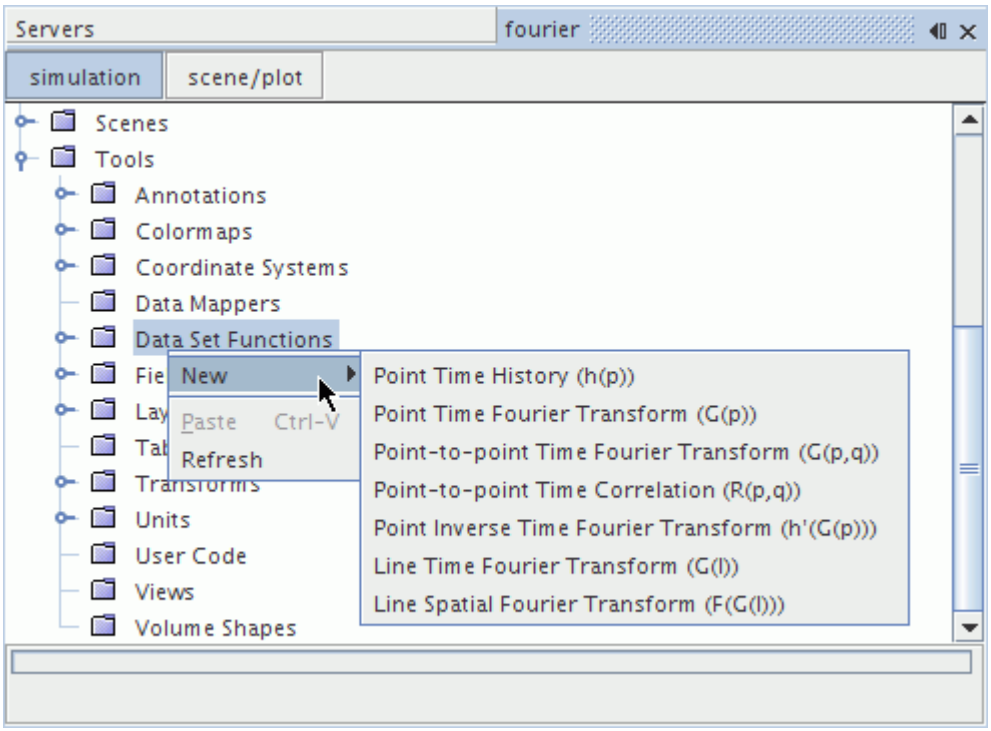
This section explains the following:

Contents:

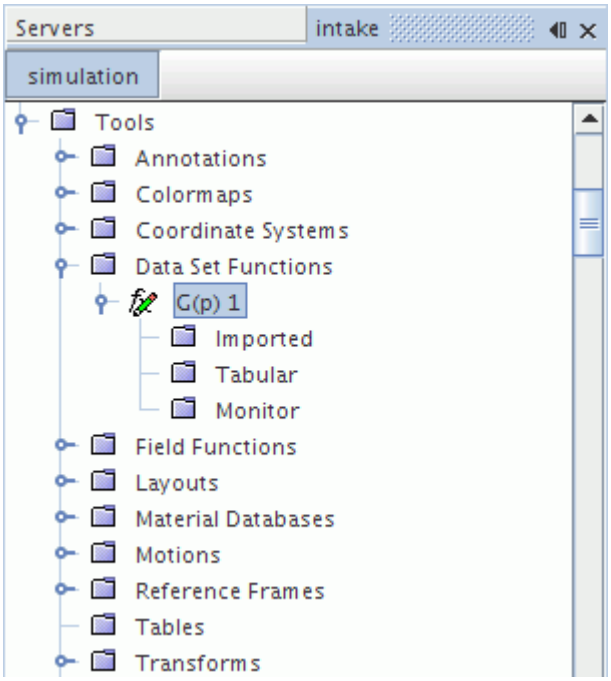
Creating Data Set Functions
What Are Point Time Histories?
What Are Point Time Fourier Transforms?
Using Point Time Fourier Transforms
What Are Point-to-Point Time Fourier Transforms?
What Are Point-to-Point Time Correlations?
What Are Point Inverse Time Fourier Transforms?
What Are Line Time Histories?
What Are Line Time Fourier Transforms?
Using Line Time Fourier Transforms
What Are Line Spatial Fourier Transforms?
Using Line Spatial Fourier Transforms
What Are Surface Time Histories?
What Are Surface Time Fourier Transforms?
FIR Filtering to Compute Band Filtered Surface Time Histories
What Is Derived Data?
Post-Processing Data Set Functions
Pop-Up Menu of the Data Set Functions Manager
Data Set Functions Formulation

Creating Data Set Functions

To add a data set function, right-click the **Data Set Functions** node and select a type from the **New** submenu of the pop-up menu.



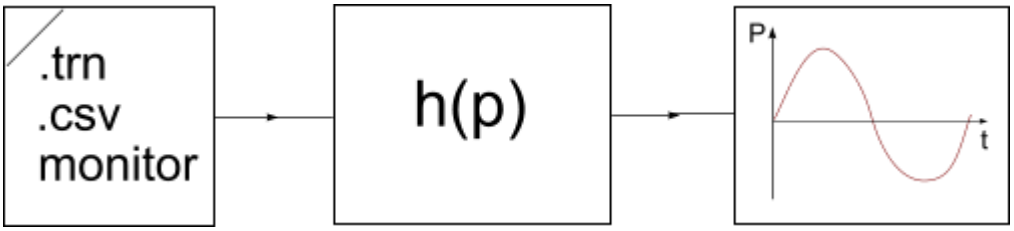
The node of the new object appears in the simulation tree.


Servers
intake
simulation
Tools
Annotations
Colormaps
Coordinate Systems
Data Set Functions
C(p) 1
Imported
Tabular
Monitor
Field Functions
Layouts
Material Databases
Motions
Reference Frames
Tables
Transforms

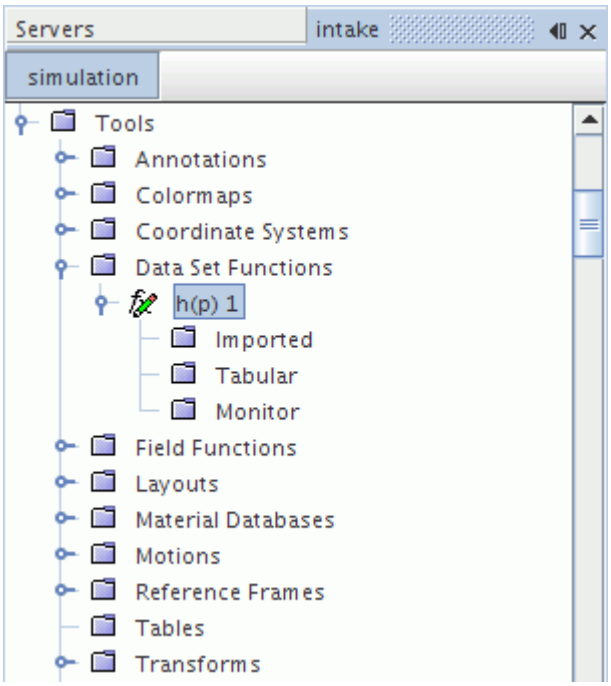
What Are Point Time Histories?

Point Time Histories are time histories at a point in the input signal which are then processed using a Fourier transform.

Point Time Histories can also be used to apply filters to signals.



The **h(p)** node (with a numerical suffix) represents this type of data set function in the object tree. This node has properties and is subject to common menu items.



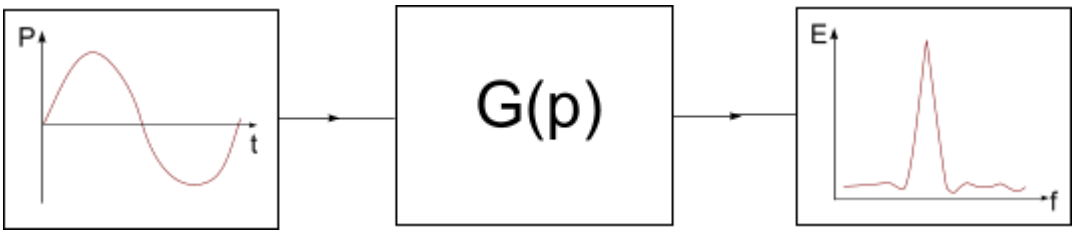
Point Time History Properties

Window Function	Specifies the window function that you require for this data set function. These options are all standard options for this type of analysis. • **None** - This default setting leaves the data set function without a window function until you select one of the other options in this list. • **Hamming**

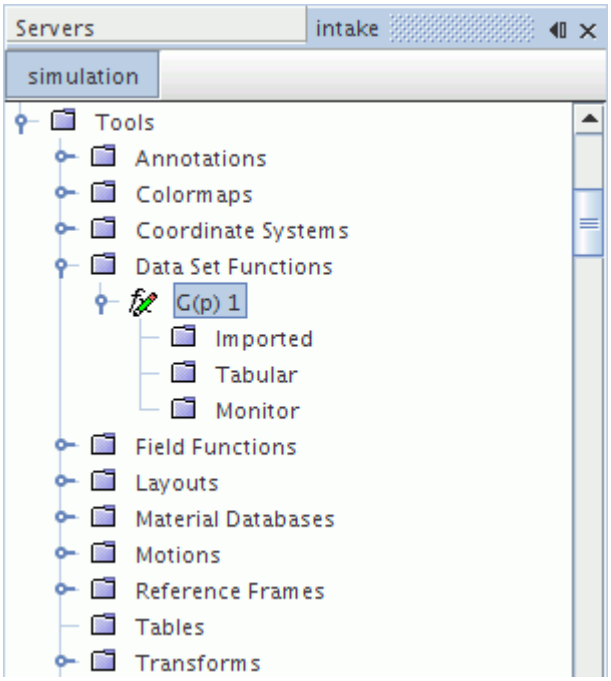
	• **Hann** (Hanning) • **Bartlett** • **Blackman**	
Update Interval	Sets the number of time-steps after which to update the time history.	
Filter Type	Specifies the type of filter that is applied.	
	None	Does not use a filter.
	Band pass filter	Keeps frequencies between low and high cut-off levels. Adds a FIR Filter sub-node in which you can set these levels.
	Band stop filter	Discards frequencies between low and high cut-off levels. Adds a FIR Filter sub-node in which you can set these levels.
Start Time	The start time for the time sampling of this object.	
Cut-off time	The end time for the time sampling for this object. Clicking the ... property customizer button in the right-hand column activates a property customizer that lets you specify both the time value and the appropriate time units (**hr**, **min** or **s**). If you have multiple data set functions, you can vary the settings of this property to sample multiple time intervals from the simulation run.	

What Are Point Time Fourier Transforms?

Point Time Fourier Transforms are Fourier transforms for one time point in the input signal.



Choosing this type of data set function creates a **G(p)** node with a numerical suffix in the object tree. This node has properties and is subject to common menu items.



The time Fourier Transform (FT) (see Eqn. (1322)) lets you transform time signals that are stored in STAR-CCM+ to the frequency domain so that you can identify dominant modes in the signals.

Point Time Fourier Transform Properties

Update Interval	Sets the number of time-steps after which to update the FT.	
Filter Type	Specifies which type of filter is applied. Digital filters allow you to filter out parts of a signal which correspond to certain frequencies.	
	None	Does not use a filter.

	Band pass filter	Keeps frequencies between low and high cut-off levels. Adds an Ideal Filter sub-node in which you can set these levels.
	Band stop filter	Discards frequencies between low and high cut-off levels. Adds an Ideal Filter sub-node in which you can set these levels.
Frequency Function	Specifies the frequency function that is generated. These functions are all standard functions for this type of analysis.	
	Frequency	Frequency in Hz.
	Octave Bands	Discrete frequency bands corresponding to different octaves within hearing range. Each successive band represents twice the range of the previous one.
	3rd Octave Bands	Discrete frequency bands corresponding to one-third of each octave within hearing range.
	Fourier Mode	The index in the Fourier calculations.
	Strouhal Number	The non-dimensional version of the `Frequency` option. Selecting this option adds a Strouhal Number sub-node to the FT node.
	Custom Bands	Lets you define a set of custom frequency bands by adding a Custom Bands sub-node to the FT node.
Start Time	The start time of the time sampling for this object.	
Cut-off time	The end time of the time sampling for this object. Clicking the ... property customizer button in the right-hand column activates a property customizer that lets you specify both the time value and the appropriate time units (**hr**, **min**, or **s**). If you have multiple FTs, you can vary the settings of this property to sample multiple time intervals from the simulation run.	
Amplitude Function	Specifies the amplitude function that is generated. These functions are all standard functions for this type of analysis. **Note:** The function domain is limited to positive frequencies.	

	Power Spectral Density	Frequency-domain description of the signal power.
	Sound Pressure Level	Sound pressure in decibels. Adds the Reference Sound Pressure node.
	Root Mean Square	The square root of the `Power Spectral Density`.
	A-Weighted Sound Pressure Level	Weights the `Sound Pressure Level` with the A-scale function, for loudness less than 55 phons (at 1000 Hz, 1 phon = 1 dB). This function is typically for problems that involve the frequency range for human hearing. Adds the Reference Sound Pressure node.
	B-Weighted Sound Pressure Level	Weights the `Sound Pressure Level` with the B-scale function, for loudness of 55-85 phons. Adds the Reference Sound Pressure node.
	C-Weighted Sound Pressure Level	Weights the `Sound Pressure Level` with the C-scale function, for loudness greater than 85 phons. Adds the Reference Sound Pressure node.
	Re{FFT Coefficients}	The real parts of the FFT coefficients for positive frequencies.
	Im{FFT Coefficients}	The imaginary parts of the FFT coefficients for positive frequencies.
	Phase{FFT Coefficients}	The phases of the FFT coefficients ($[Im/\mathrm{Re}]$) for positive frequencies.
	Abs{FFT Coefficients}	The absolute values of the FFT coefficients ($\sqrt{\lvert\mathrm{Re}\rvert^2 + \lvert Im\rvert^2}$) for positive frequencies.
Analysis Blocks	Specifies the number of analysis blocks N_b (see Eqn. (1356)) for this data set function to use.	
Overlap Factor	Sets the overlap factor α (see Eqn. (1356)) between blocks, if any, in a range from `0` to `0.9`. This feature allows for smooth transitions between blocks. This property can be set with a slider.	

Window Function	Specifies the window function that you require for this FT. Fourier transforms assume a periodic signal; to enforce this condition apply a window function. Window functions smear the dominant modes in the frequency domain. If the signal is broken into several analysis blocks, the window function is applied to each block. There are a number of commonly used window functions. All taper a signal at the beginning and/or end. • **None** - This default setting leaves the data set function without a window function until you select one of the other options in this list. • **Hamming** • **Hann** (Hanning) • **Bartlett** • **Blackman**

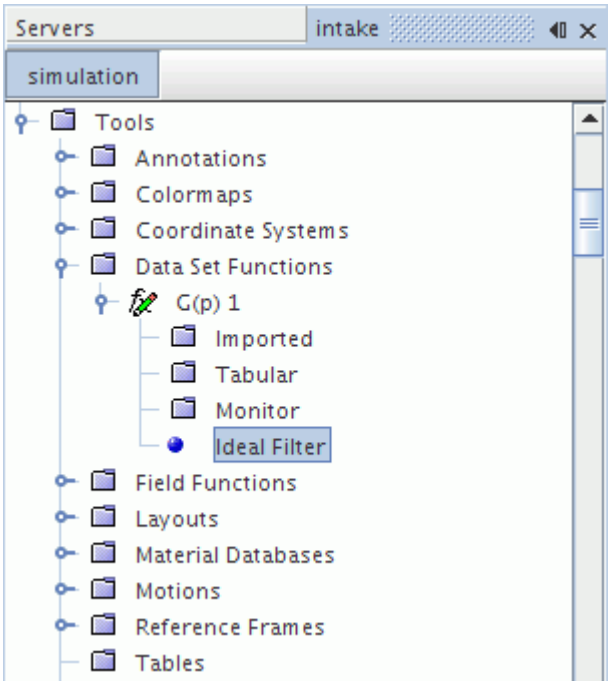
Contents:

What Is the Ideal Filter?
What Are Bands?
What Is the Strouhal Number?
What is Reference Sound Pressure?

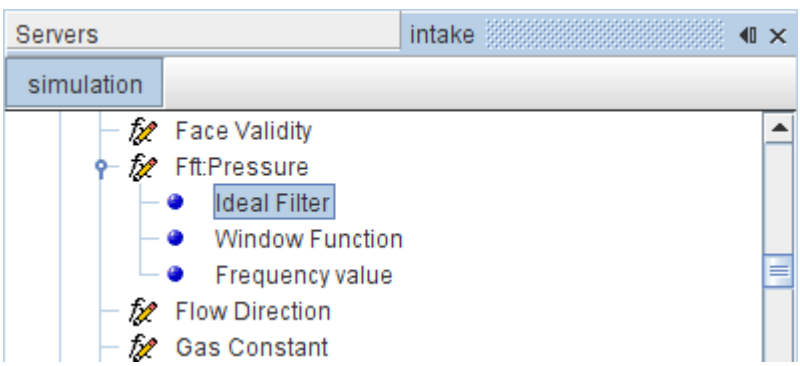
What Is the Ideal Filter?

The **Ideal Filter** node represents this feature.

This node has properties and is subject to common menu items. It is made available when the *Filter Type* property of the Fourier transform is set to either `Band pass filter` or `Band stop filter`.



The same type of sub-node can be activated in the Fourier Transform field function.

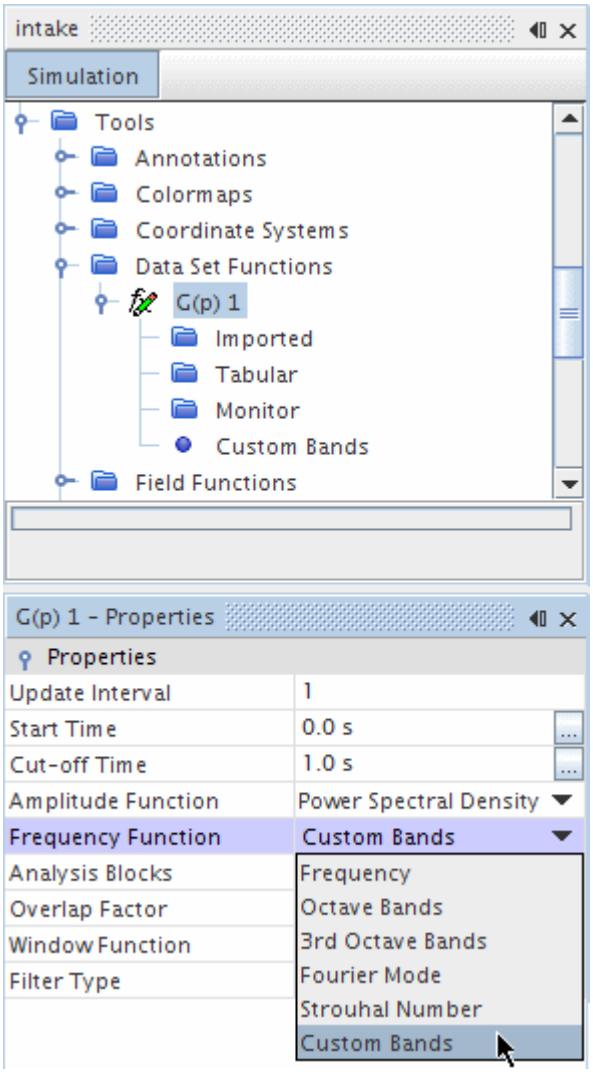


Ideal Filter Properties

Low Cut-off Frequency	Retains frequencies above f_1 (see Eqn. (1370)) when the band pass filter is used, and discards them when the band stop filter is used.
High Cut-off Frequency	Retains frequencies below f_2 (see Eqn. (1371)) when the band pass filter is used, and discards them when the band stop filter is used. Clicking the ... property customizer button in the right-hand column activates a property customizer that lets you specify both the desired frequency and the appropriate time units (**/s** or **Hz**)

What Are Bands?

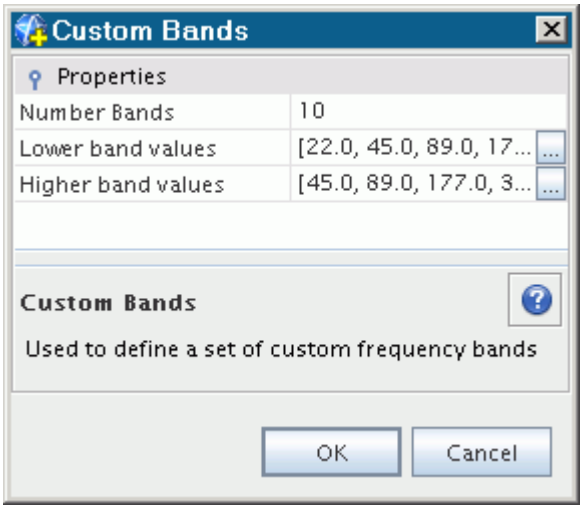
The *Frequency Function* property can be set to **Octave Bands**, **3rd Octave Bands**, or **Custom Bands**.



Each setting divides the frequency range differently:

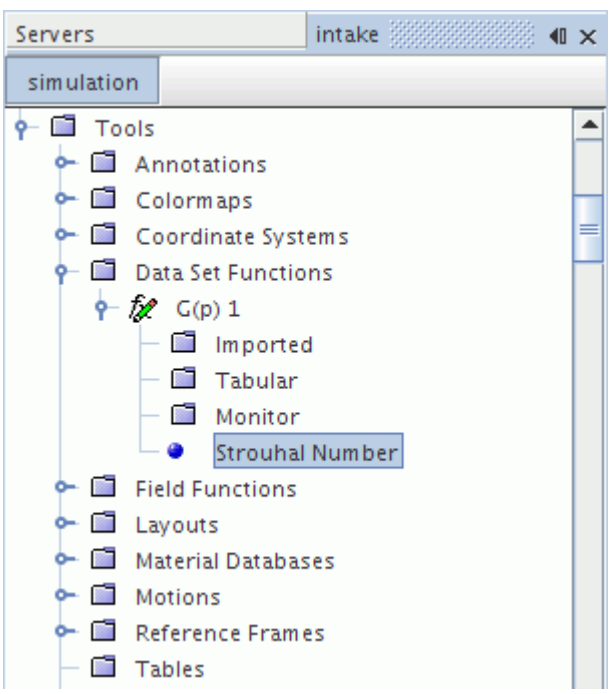
- **Octave Bands** divides the range into bands one octave wide.
- **3rd Octave Bands** divides the range into bands one third of an octave wide.
- **Custom Bands** divides the range into the bands you have selected.

When **Custom Bands** is selected, an object for it appears in the node of the Fourier Transform. By clicking the ... property customizer button of the *Custom Bands* property of this object, you can activate a dialog that makes the settings more accessible.



What Is the Strouhal Number?

The **Strouhal Number** object, which has properties, appears in the node of the Fourier Transform when the *Frequency Function* property is set to `Strouhal Number`.



Strouhal Number Properties

These properties let you adjust the Strouhal number settings, expressed as $l_{ref} \bullet p/u_{ref}$.

Reference Length	The reference length l_{ref}, which can be specified with a text entry. The default is 1.0.
Reference Velocity	The reference velocity u_{ref}, which can be specified with a text entry. The default is 1.0. Clicking the ... property customizer button in the right-hand column activates a property customizer that lets you specify both the desired value and the appropriate units.

What is Reference Sound Pressure?

Reference sound pressure is a constant, used to calculate the Sound Pressure Level.

It specifies the reference ambient pressure p_{ref} that is used when calculating A-, B- or C-Weighted Sound Pressure Level.

The default value of this constant depends on the medium:

- 2.0E-5 Pa for air
- 1.0E-6 Pa for water

Using Point Time Fourier Transforms

You typically run the Point Time Fourier Transform on time-history data that the simulation generates.

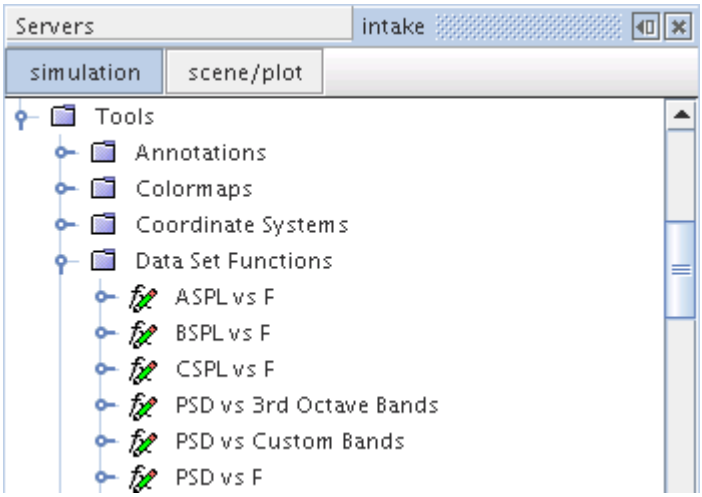
Therefore, before you start running the simulation:

1. Identify the variables for which time-history data is required (for example, pressure reported at a point probe).
2. Activate the transient time model.
3. Set up reports, monitors, and monitor plots that use the variables of interest.

The monitors control the trigger and frequency of the calculations. You can create plots that are based on those monitors alone, or based on the FTs that you add. The latter option is possible through the derived data set objects of a monitor plot.

4. Create Fourier Transforms and set their properties:
   a) Right-click the **Tools** > **Data Set Function**s manager node and choose the required transform from the pop-up menu.
   b) Select from among sets of standard options for amplitude, frequency, and window functions.
   c) Break the signals into analysis blocks, working with each block separately. An overlap factor can be applied to analysis blocks for smooth transitions.

You can create any number of FTs with different combinations of settings, including different sampling time intervals. To track your combinations of settings, simply rename each FT node according to its settings.

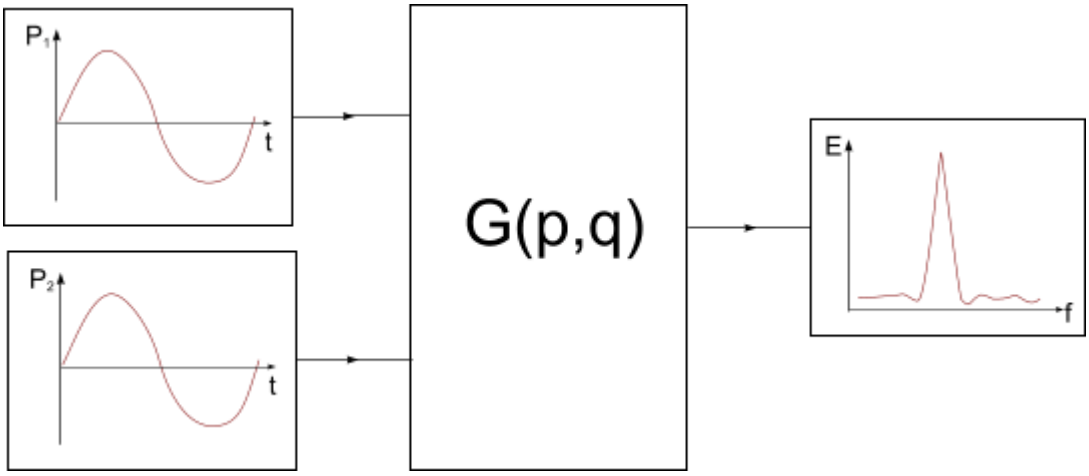


Run the simulation and examine the results.

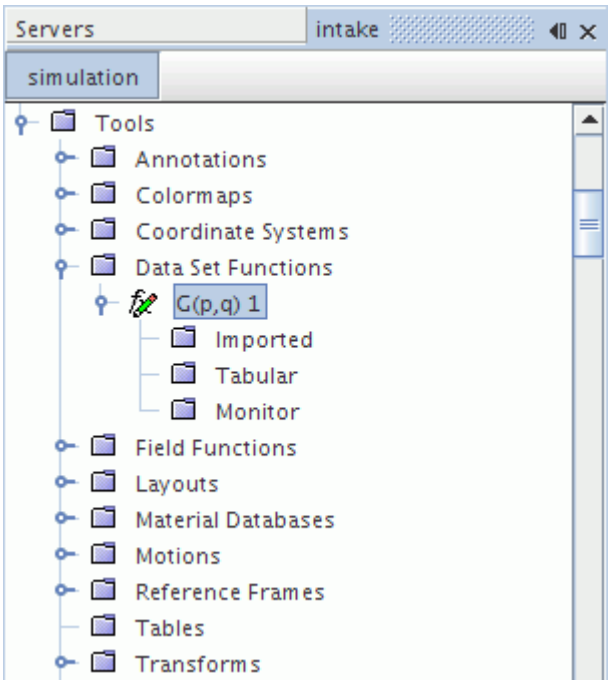
If applicable, validate the simulation results with your own experimental data, which the FT can access through its derived tabular data set.

What Are Point-to-Point Time Fourier Transforms?

You use Point-to-Point Time Fourier transforms to determine the correlation between two input signals. This data set function applies a Fourier transform to each of the input signals, then performs a convolution on both resulting transforms.



Choosing this type of data set function creates the **G(p,q)** node (with a numerical suffix) in the object tree. This function has double the number of properties of a 1-point transform because a set of properties is defined for each point. The corresponding node has properties and is subject to common menu items



Point-to-Point Time Fourier Transform Properties

Update Interval	Sets the number of time-steps after which to update the Fourier transform.	
Filter Type	Specifies the type of filter that is applied. Digital filters allow you to filter out parts of a signal which correspond to certain frequencies.	
	None	Does not use a filter.
	Band pass filter	Keeps frequencies between low and high cut-off levels. Adds an Ideal Filter sub-node in which you can set these levels.
	Band stop filter	Discards frequencies between low and high cut-off levels. Adds an Ideal Filter sub-node in which you can set these levels.
Frequency Function	Specifies the frequency function that is generated. All these functions are standard functions for this type of analysis.	
	Frequency	Frequency in Hz.
	Octave Bands	Discrete frequency bands corresponding to different octaves within hearing range. Each successive band represents twice the range of the previous one.
	3rd Octave Bands	Discrete frequency bands corresponding to one-third of each octave within hearing range.
	Fourier Mode	The index in the Fourier calculations.
	Strouhal Number	The non-dimensional version of the `Frequency` option. Selecting this option adds a Strouhal Number sub-node to the FT node.
	Custom Bands	Lets you define a set of custom frequency bands by adding a Custom Bands sub-node to the FT node.
Start Time	The start time of the time sampling for this object.	
Cut-off time	The end time of the time sampling for this object. Clicking the ... property customizer button in the right-hand column activates a property customizer that lets you specify both the time value and the appropriate time units (**hr**, **min**, or **s**). If you have multiple FTs,	

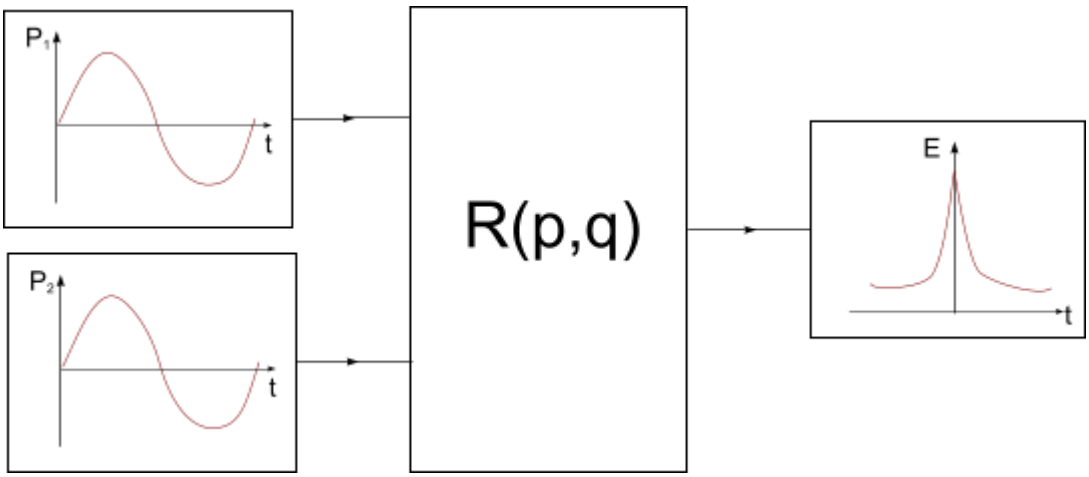
	you can vary the settings of this property to sample multiple time intervals from the simulation run.	
Amplitude Function	Specifies the amplitude function that is generated. All these functions are standard functions for this type of analysis. **Note:** The function domain is limited to positive frequencies.	
	Cross Spectral Density	The power spectral density of the cross-spectrum signal, $G(x_1, x_2; \omega)$ (see Eqn. (1333)). This option is comparable to the auto-spectral density of 1-point Fourier transforms, $G(x; \omega)$ (see Eqn. (1332)).
	Cross Root Mean Square	The root mean square of the `Cross Spectral Density`. This option is comparable to the RMS of 1-point Fourier transforms. See Frequency-Space Parseval's Relation.
	Coherence	γ in Eqn. (1339).A measure of the correspondence between the magnitude of two signals at different frequencies. Its range is 0 - 1.
	Phase	θ in Eqn. (1338). A measure of the offset (phase difference) between two signals. Its range is -p to +p.
Analysis Blocks	Specifies the number of analysis blocks N_b (see Eqn. (1356)) for this data set function to use.	
Overlap Factor	Sets the overlap factor α (see Eqn. (1356)) between blocks, if any, in a range from `0` to `0.9`. This property allows for smooth transitions between blocks.	
Window Function	Specifies the window function that you require for this FT. Fourier transforms assume a periodic signal; to enforce this condition apply a window function. Window functions smear the dominant modes in the frequency domain. If the signal is broken into several analysis blocks, the window function is applied to each block. There are a number of commonly used window functions. All taper a signal at the beginning and/or end. • **None** - This default setting leaves the data set function without a window function until you select one of the other options in this list.	

	• **Hamming** • **Hann** (Hanning) • **Bartlett** • **Blackman**

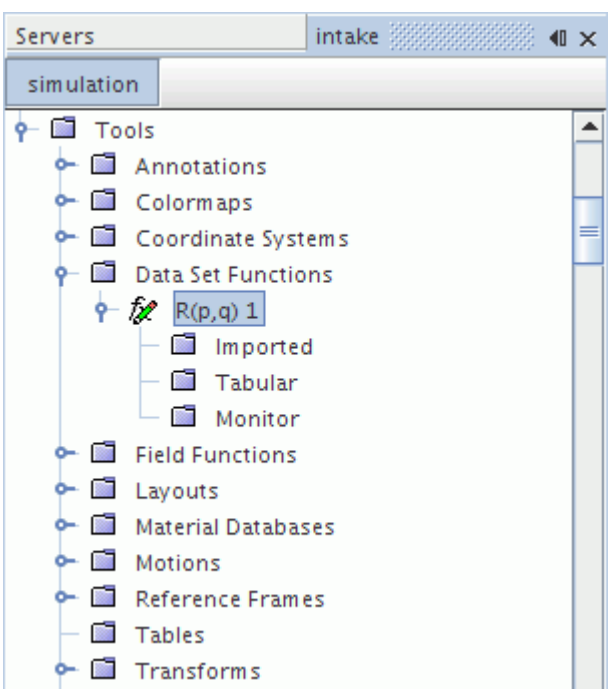
What Are Point-to-Point Time Correlations?

You use Point-To-Point Time correlations to determine the phase differences between the dominant modes of two input signals.

Point-to-Point Time Correlations are time cross-correlations at any two points x_1 and x_2 in space (see Time Cross-Correlations for the formulation)..



Choosing this type of data set function creates the **R(p,q)** node (with a numerical suffix) in the object tree.



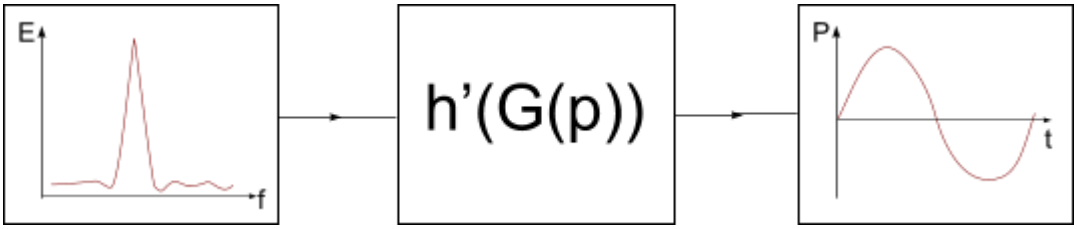
Point-to-Point Time Correlation Properties

Amplitude Function	Specifies the function that is used to represent the cross-correlation. **Note:** The function domain is limited to positive frequencies. • **Correlation Value** • **Correlation Magnitude** • **Correlation Coefficient** - The magnitude is scaled to be in the range -1 to +1.	
Update Interval	Sets the number of time-steps after which to update the time history.	
Filter Type	Specifies the type of filter that is applied.	
	Band pass filter	Keeps frequencies between low and high cut-off levels. Adds a FIR Filter sub-node in which you can set these levels.
	Band stop filter	Discards frequencies between low and high cut-off levels. Adds a FIR Filter sub-node in which you can set these levels.
Start Time	The start time for the time sampling of this object.	
Cut-off time	The end time for the time sampling for this object. Clicking the ... property customizer button in the right-hand column activates a property customizer that lets you specify both the time value and the appropriate time units (**hr**, **min**, or **s**). If you have multiple data set functions, you can vary the settings of this property to sample multiple time intervals from the simulation run.	

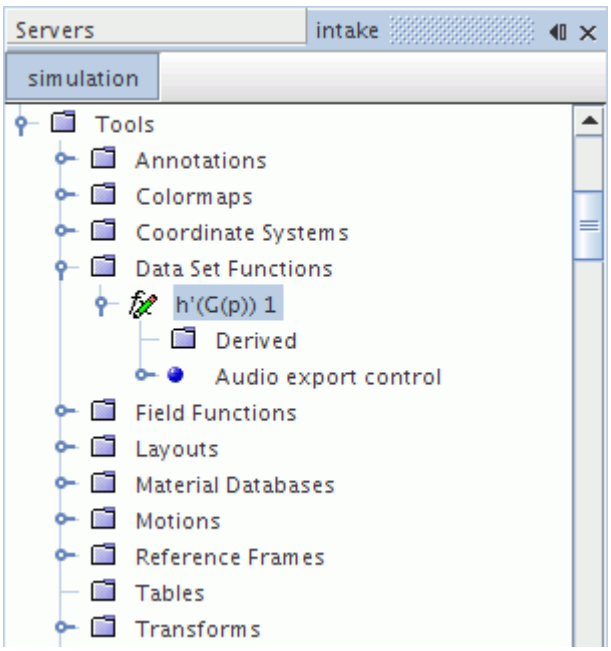
What Are Point Inverse Time Fourier Transforms?

This inverse transform allows data to be transformed from the frequency to the time domain.

This transform is given by Eqn. (1324)



The **h'(G(p))** node (with a numerical suffix) represents this type of data set function in the object tree. This node has properties and is subject to common menu items.



Point Inverse Time Fourier Transform Properties

Amplitude Function	Selects the method by which an Inverse Fourier Transform can be visualized as its magnitude or using its real or imaginary parts. The transform can be attached to both auto- and cross-correlation spectra. **Note:** The function domain is limited to positive frequencies.	
	Real	The inverse of auto-correlation spectra has only a real part

	Imaginary	The inverse of cross-correlation spectra has both real and imaginary parts
	Magnitude	The magnitude of the spectra.
Added Mean Type	Selects the method that is used to generate the mean value added to the inverse Fourier transform.	
	Calculated	Generated by STAR-CCM+.
	User specified	Entered manually. A User Specified Added Mean node is added to the object tree.
Signal Repeats	Selects the number of times the signal is repeated (without introducing discontinuities). It is set to zero by default.	

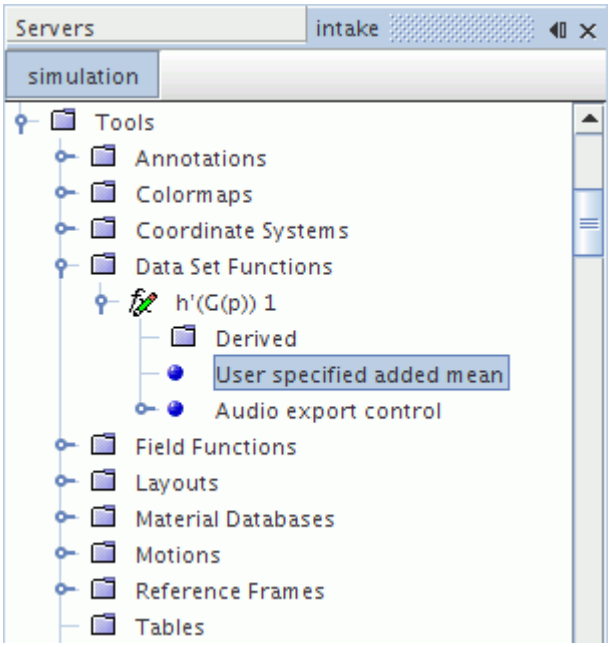
Contents:

What Is the User Specified Added Mean?
Exporting Audio

What Is the User Specified Added Mean?

The **User Specified Added Mean** node represents this feature.

This node has properties and is subject to common menu items. It is made available when the *Added Mean Type* property of the point inverse time Fourier transform is set to **User specified**.



User Specified Added Mean Properties

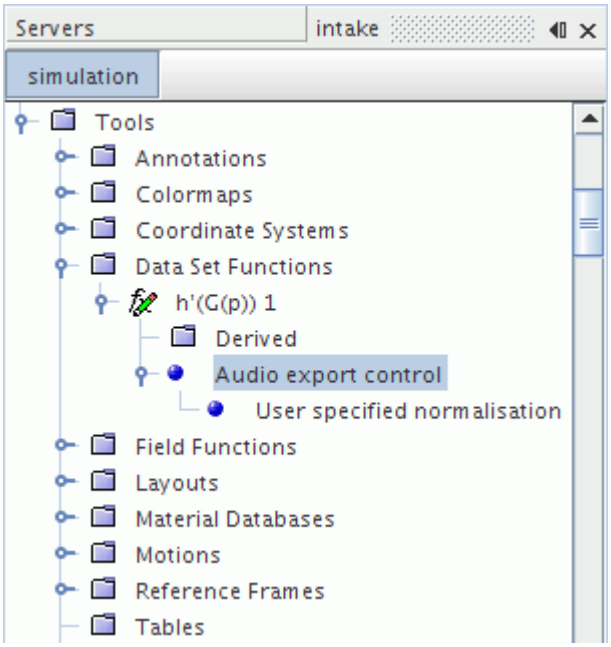
Added Mean	Specifies the mean value that is added to the inverse Fourier transform. Clicking the ... property customizer button in the right-hand column activates a property customizer that lets you specify the added mean.

Exporting Audio

You can export the time signal that the point inverse time Fourier transform generates as an audio file using the **Export Audio...** action on the derived data node pop-up menu.

STAR-CCM+ currently supports the Windows Media Player format (`.wav`) for exported audio files.

The **Audio export control** node allows you to scale all audio exports from a point inverse time Fourier transform to the same maximum amplitude. This feature gives similar volume levels across all the signals.



Audio Export Control Properties

Normalization	Specifies whether normalization of exported audio data is required.	
	No normalization	Normalization is not required.
	User specified normalization	Activates normalization of all audio data to the same value. This option creates an additional User specified normalization node in the simulation tree.

Contents:

User Specified Normalization

User Specified Normalization

This node specifies a *Normalization comparison value* that is used to scale all exported audio signals to the same amplitude range.

Set this value to the peak value of the derived data set that has the highest-amplitude signal. The highest amplitude signal can be detected by subtracting the mean value from all of the derived data sets belonging to the point inverse time Fourier transform and comparing the resulting traces on a plot.

To subtract the mean value from the derived data sets:

1. Select option **User Specified** for the Added Mean Type property on the Inverse Fourier Transform node.
2. Enter a value of `0.0` for the User specified added mean.

When you have estimated the peak amplitude across all the derived data sets, enter this value in the *Normalization comparison value* property.

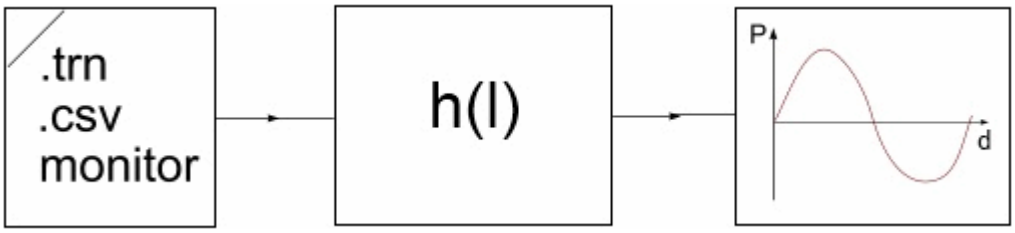
User Specified Normalization Properties

Normalization comparison value	All exported signals are normalized against this value.

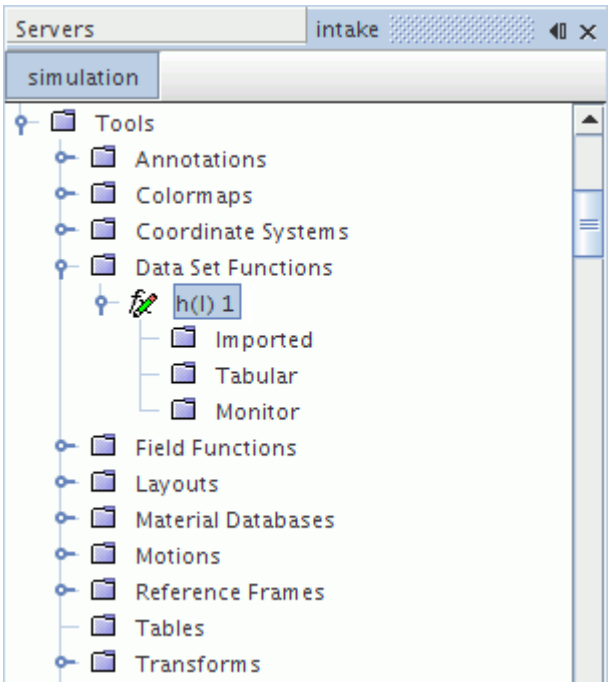
What Are Line Time Histories?

A Line Time History is a collection of Point Time Histories that are drawn from the points that make up a line probe.

A line time history can then be processed using a Line Time Fourier Transform.



The **h(l)** node (with a numerical suffix) represents this type of data set function in the object tree. This node has properties and is subject to common menu items.



Line Time History Properties

Start Signal	The start time for the mapped signal.
End Signal	The end time for the time sampling for this object. Clicking the ... property customizer button in the right-hand column activates a property customizer that lets you specify both the time value

	and the appropriate time units (**hr**, **min** or **s**). If you have multiple data set functions, you can vary the settings of this property to sample multiple time intervals from the simulation run.

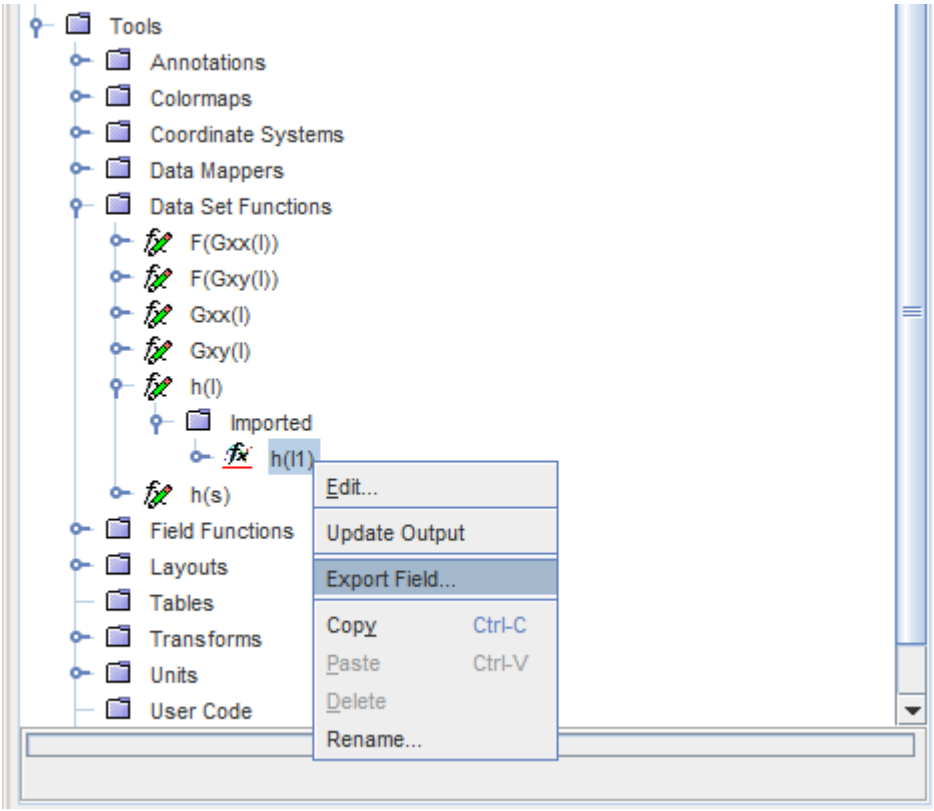
Contents:

Exporting Fields

Exporting Fields

You can export derived transient field data as a Matlab `.mat` file.

To specify the name and location of the `.mat` file, right-click the function node, select **Export Field...**, and complete the dialog.



h(l1) Node Properties

Part	The line part containing the positions from which *Field Function 1* is extracted.
Data Surfaces 1	The imported surface on which *Field Function 1* is extracted.
Field Function 1	The signal is extracted from this scalar field function.
Update Active	Whether the update is active for the imported history.

	Activated	The update is active.
	Deactivated	The update is inactive. This state is the default.

Exportable Field Data

VERSION	Version string for the file.
X, Y, Z	Position components of the point samples in the file.
X_UNITS, Y_UNITS, Z_UNITS	Units of position components for the point samples in the file.
S	Samples for series that is associated with the point.
S_UNIT	Units of samples for series that is associated with each point.
M_TITLES	Titles of each field that is stored in the file.
M_UNITS	Units for each field that is stored in the file.
Field<i>_<j>	The jth sample of ith field for each point sample.

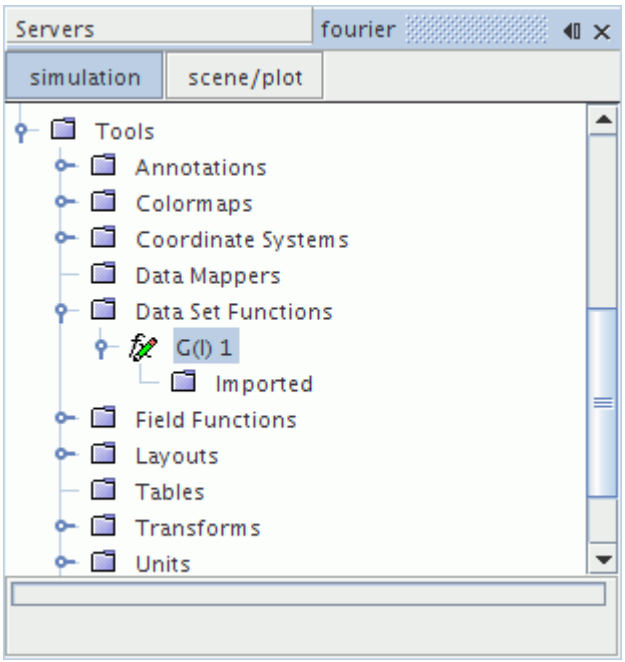
What Are Line Time Fourier Transforms?

Line Time Fourier Transforms let you calculate Fourier transforms at various point along a line probe.

They allow the creation of either auto-spectra (see Eqn. (1332)) or cross-spectra (Eqn. (1333)) along the line probe.

$$h(\underline{x};t) \rightarrow G(\underline{x};\omega)$$

Choosing this type of data set function creates a **G(I)** node (with a numerical suffix) in the object tree. This node has properties and is subject to common menu items.



Line Time Fourier Transform Properties

Start Signal	The start time for the sampling of this object.
End Signal	The end time for the sampling for this object. Clicking the ... property customizer button activates a property customizer that lets you specify both the time value and the appropriate time units (**hr**, **min**, or **s**). If you have multiple data set functions, you can vary the settings of this property to sample multiple time intervals from the simulation run.
Amplitude Function	Specifies the amplitude function that is generated. These functions are all standard for this type of analysis. **Note:** The function domain is limited to positive frequencies.

	Power Spectral Density	Frequency-domain description of the signal power.
	Cross Spectral Density	The power spectral density of the cross-spectrum signal. This option is comparable to the auto-spectral density of 1-point Fourier transforms.
Analysis Blocks	Specifies the number of analysis blocks N_b (see Eqn. (1356)) for this data set function to use.	
Overlap Factor	Sets the overlap factor α (see Eqn. (1356)) between blocks, if any, in a range from `0` to `0.9`. This property allows for smooth transitions between blocks.	
Window Function	Specifies the window function that you require for this FT. Fourier transforms assume a periodic signal; to enforce this condition, apply a window function. Window functions smear the dominant modes in the frequency domain. If the signal is broken into several analysis blocks, the window function is applied to each block. There are a number of commonly used window functions. All taper a signal at the beginning and/or end.	
	None	This default setting leaves the FT without a window function until you select one of the other options in this list.
	Hamming	Uses the Hamming function.
	Hann	Uses the Hann (Hanning) function.
	Bartlett	Uses the Bartlett function.
	Blackman	Uses the Blackman function.

Contents:

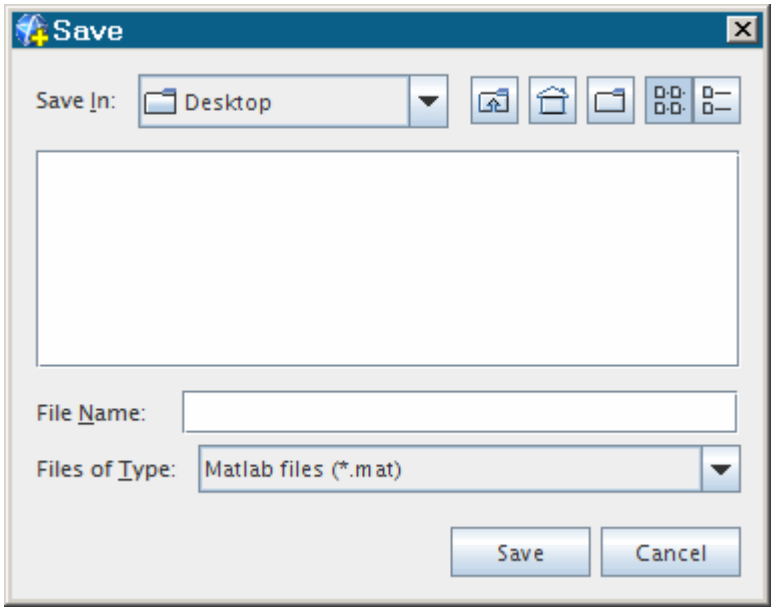
Exporting Fields

Exporting Fields

You can export derived transient field data as a Matlab `.mat` file.

To specify the name and location of the `.mat` file:

1. Right-click the function node and select **Export Field...**
2. In the *Save* dialog, navigate to a location, enter a file name, and click **Save**.



Multi-Point Imported Properties

Input time history	The line time history **G(In)** to use as input, selected from pull-down menu.	
X Units	Displays the calculated units for the x-axis data.	
Y Units	Displays the calculated units for the y-axis data.	
Update Active	Whether the update is active for the imported line.	
	Activated	The update is active.
	Deactivated	The update is inactive. This state is the default.

Multi-Point Imported Expert Properties

X Surface Axis Units	The units in which the X axis is specified.	
Y Surface Axis Units	The units in which the Y axis is specified.	
Z Surface Axis Units	The units in which the Z axis is specified.	

Exportable Field Data

Re{FFT Coefficients}	The real parts of the FFT coefficients.
Im{FFT Coefficients}	The imaginary parts of the FFT coefficients.

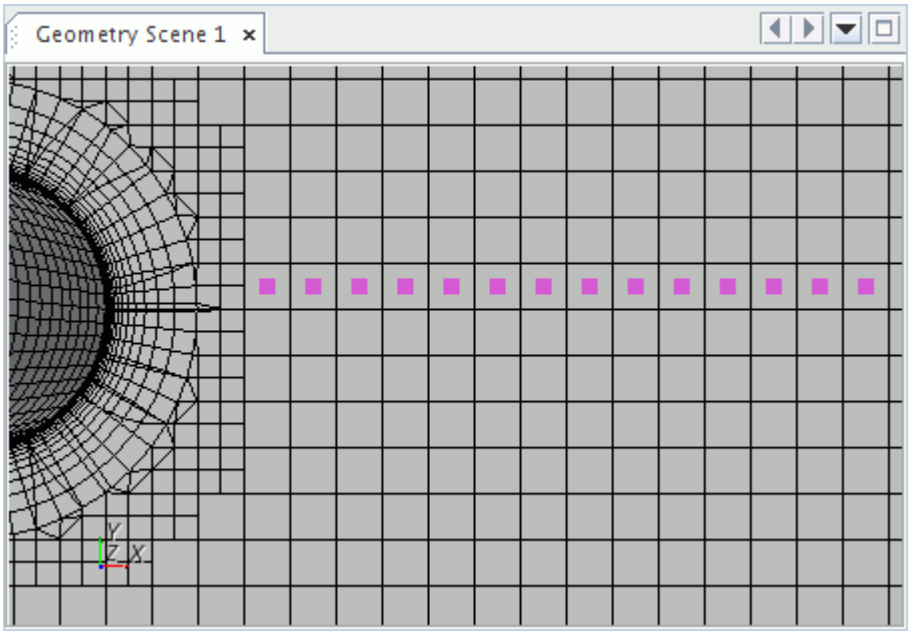
Using Line Time Fourier Transforms

You can set up a line time Fourier transform for a transient property and visualize the transform.

To set up a line time Fourier transform:

1. Create a line probe in the region of interest.

To minimize inaccuracies due to visualization interpolation, adjust the probe properties such that data is extracted at cell centers.



2. Create a derived line data set using the line probe.
3. Set the properties of the derived line data set and calculate the transform.
4. Set the properties for the transform plots.
5. Create a monitor plot and plot the derived line data set.
6. Visualize the spectra in a scene.

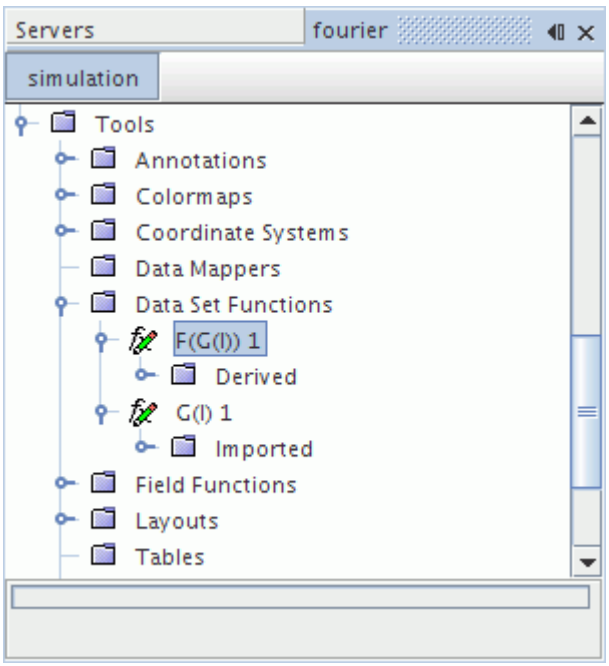
What Are Line Spatial Fourier Transforms?

Line Spatial Fourier Transforms allows the creation of a matrix of energy densities dependent on wavenumber and frequency along a line probe.

These data set functions map signals from the spatial and frequency domains to wavenumber and frequency domains.

$$G(\underline{x}; t) \rightarrow F(\underline{k}; \omega)$$

Choosing this type of data set function creates an **F(G(l))** node in the object tree, which has properties and is subject to common menu items.



Line Spatial Fourier Transform Properties

Start Signal	The start length for the length sampling of this object.
End Signal	The end length for the length sampling for this object. Clicking the ... property customizer button activates a property customizer that lets you specify both the time value and the appropriate length units. If you have multiple data set functions, you can vary the settings of this property to sample multiple length intervals from the simulation run.
Padding Factor	Specifies the degree by which the signal is zero-padded. The factor is a proportion of the original signal strength.
Window Function	Specifies the window function that you require for this FT.

	Fourier transforms assume a periodic signal; to enforce this condition, apply a window function. Window functions smear the dominant modes in the frequency domain. If the signal is broken into several analysis blocks, the window function is applied to each block. There are a number of commonly used window functions. All taper a signal at the beginning and/or end.	
	None	This default setting leaves the FT without a window function until you select one of the other options in this list.
	Hamming	Uses the Hamming function.
	Hann	Uses the Hann (Hanning) function.
	Bartlett	Uses the Bartlett function.
	Blackman	Uses the Blackman function.

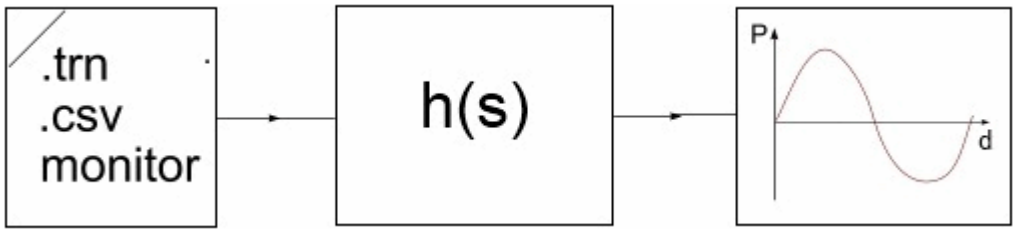
Using Line Spatial Fourier Transforms

To set up a line spatial Fourier transform for a transient property:

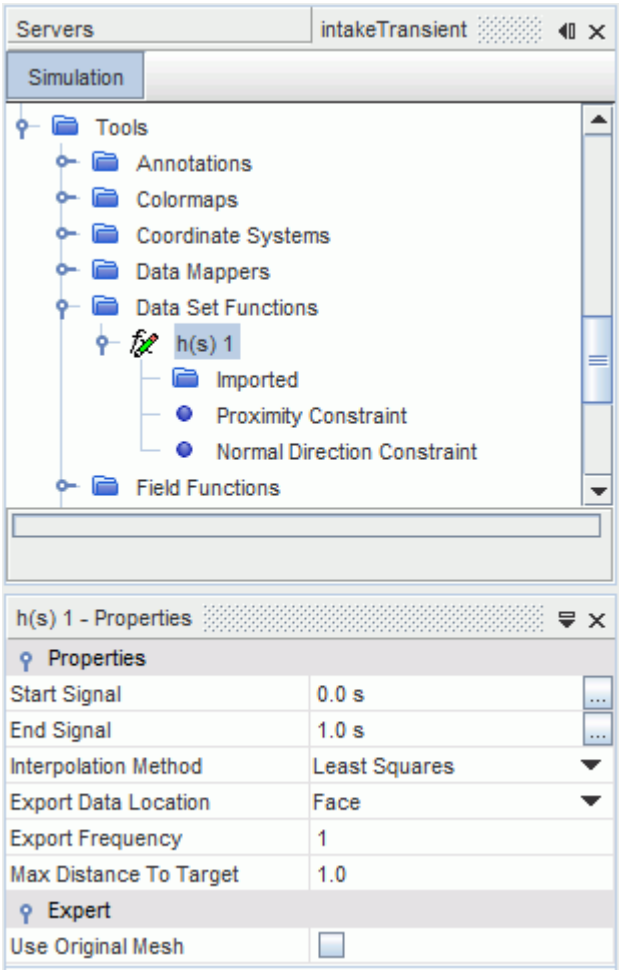
1. Create and set up a line time Fourier transform.
2. Create a line derived data set using a derived line data set from a line time Fourier transform.
3. Set the properties of the line derived data set to calculate the transform.
4. Set the properties for the transform plots.
5. Create a monitor plot and plot the derived line data set.
6. Visualize the spectra in a scene.

What Are Surface Time Histories?

A Surface Time History is a collection of Point Time Histories that are drawn from the points that make up a presentation grid.



The **h(s)** node (with a numerical suffix) represents this type of data set function in the object tree. This type of data set uses CAE data mappers.



This node has properties and is subject to common menu items.

Surface Time History Properties

Start Signal	The start time for the mapped signal.
End Signal	The end time for the time sampling for this object. Clicking the property customizer button in the right-hand column activates a property customizer that lets you specify both the time value and the appropriate time units (**hr**, **min** or **s**). If you have multiple data set functions, you can vary the settings of this property to sample multiple time intervals from the simulation run.
Interpolation Method	Interpolation method when mapping data from source to target surface.
Export Data Location	Data location on target surface for mapped data.
Export Frequency	Frequency of time steps for which data is mapped.
Max. Distance To Target	Maximum distance to target for which a source point is projected to the target surface. Otherwise the source point is left unmoved.

Surface Time History Expert Properties

Use Original Mesh	Allows to use the original mesh for mapping.	
	Activated	The original mesh is used.
	Deactivated	The current mesh is used.

The Surface Time History node has three child nodes. The Imported node allows to create derived transient surface data sets, and has no properties. Mapping constraints can be set through the **Proximity Constraint** node properties and **Normal Direction Constraint** node properties.

Contents:

Exporting Fields

Exporting Fields

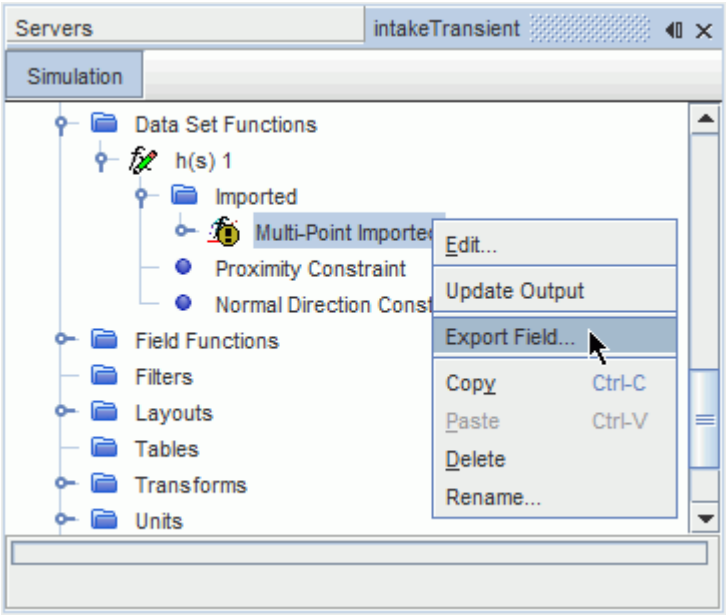
You can export derived transient field data as a Matlab `.mat` file.

The settings for automatic mapping can be entered through the surface time history node properties, whereas the mapping and export process is carried out through the derived transient surface data set node. For more information about the derived transient surface data set node and its properties, see Creating Derived Transient Surface Data Sets.

To export the derived transient field data as a Matlab `.mat` file:

1. Select the surface time history node, **h(s)**, and set its properties to specify the automatic mapping settings.
2. Expand the **h(s)** > **Imported** node.
3. Select the derived transient surface data set node,**[Multi-Point Imported]**, and activate *Update Active* to enable automatic mapping.

4. Set the remaining properties to specify the field function, data surfaces and target surfaces for mapping. You can select more than one data surface, as STAR-CCM+ allows transient mapping from multiple data surfaces.
5. Set the **Proximity Constraint** node properties and **Normal Direction Constraint** node properties.
6. Right-click the **[Multi-Point Imported]** node and select **Export Field...**



7. In the dialog, specify the name and location of the `.mat` file and click **Save**.

Transient surface data map and export operations can also be carried out through the transient imported model node.

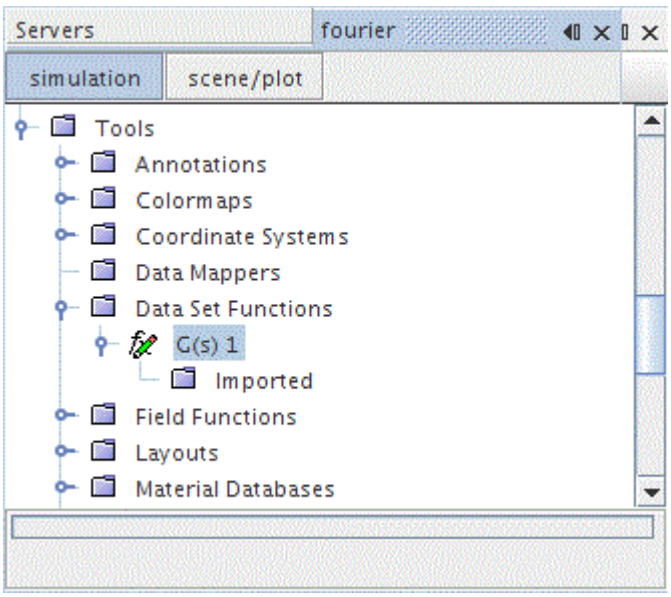
What Are Surface Time Fourier Transforms?

Surface Time Fourier Transforms let you calculate Fourier transforms at various point within a presentation grid.

They allow the creation of auto-spectra (Eqn. (1333)) over the presentation grid.

$$h(\underline{x};t) \rightarrow G(\underline{x};\omega)$$

Choosing this type of data set function creates a **G(s)** node (with a numerical suffix) in the object tree.



Surface Time Fourier Transform Properties

Start Signal	The start time for the sampling of this object.
End Signal	The end time for the sampling for this object. Clicking the ... property customizer button activates a property customizer that lets you specify both the time value and the appropriate time units (**hr**, **min**, or **s**). If you have multiple data set functions, you can vary the settings of this property to sample multiple time intervals from the simulation run.
Analysis Blocks	Specifies the number of analysis blocks N_b(see Eqn. (1356)) for this data set function to use.
Overlap Factor	Sets the overlap factor α (see Eqn. (1356)) between blocks, if any, in a range from `0` to `0.9`. This property allows for smooth transitions between blocks.

Window Function	Specifies the window function that you require for this FT. Fourier transforms assume a periodic signal; to enforce this condition, apply a window function. Window functions smear the dominant modes in the frequency domain. If the signal is broken into several analysis blocks, the window function is applied to each block. There are a number of commonly used window functions. All taper a signal at the beginning and/or end.	
	None	This default setting leaves the FT without a window function until you select one of the other options in this list.
	Hamming	Uses the Hamming function.
	Hann	Uses the Hann (Hanning) function.
	Bartlett	Uses the Bartlett function.
	Blackman	Uses the Blackman function.

Contents:

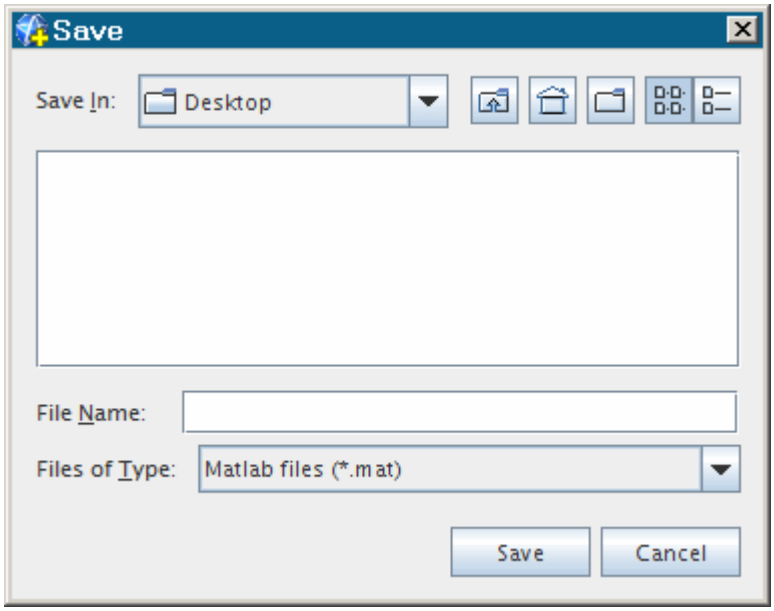
Exporting Fields

Exporting Fields

You can export derived transient field data as a Matlab `.mat` file.

To specify the name and location of the `.mat` file:

1. Right-click the function node and select **Export Field...**
2. In the *Save* dialog, navigate to a location, enter a file name, and click **Save**.



Multi-Point Imported Properties

Input time history	The surface time history **G(sn)** to use as input, selected from pull-down menu.	
X Units	Displays the calculated units for the x-axis data.	
Y Units	Displays the calculated units for the y-axis data.	
Update Active	Whether the update is active for the imported surface.	
	Activated	The update is active.
	Deactivated	The update is inactive. This state is the default.

Multi-Point Imported Expert Properties

X Surface Axis Units	The units in which the X axis is specified.
Y Surface Axis Units	The units in which the Y axis is specified.
Z Surface Axis Units	The units in which the Z axis is specified.

Exportable Field Data

Re{FFT Coefficients}	The real parts of the FFT coefficients.

FIR Filtering to Compute Band Filtered Surface Time Histories

You can use FIR filters to create frequency-band filtered surface time histories, which is similar to carrying out inverse surface FFT. Using FIR filtered surface time histories enables you to visualize flow quantitites in a specified frequency range, which can be useful for identifying and analyzing periodic flow structures or time-harmonic phenomena such as acoustic waves.

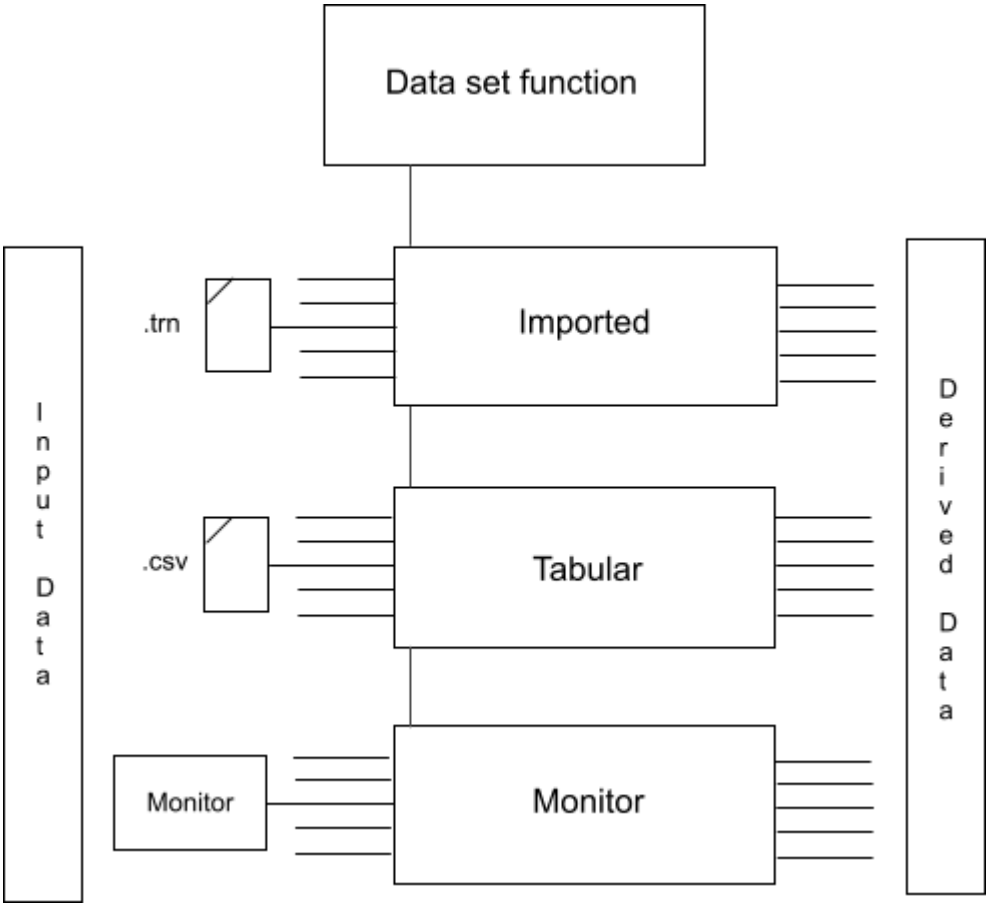
FIR filtering can only be carried out on scalar quantities saved in .trn files. FIR filter surface time histories can be calculated using the following procedure:

1. Import the `.trn` file.
2. Create a scalar scene and populate it with surfaces from the `.trn` file. (For an example of similar work with surfaces in a scene, see Plotting Pressure at Different Time-Steps in the aeroacoustics tutorial "Signal Post-Processing: FFT and Wavenumber.”)
3. Use the Time History field function for the scalar in question (for example *Time History:Pressure*) to display and filter the scalar field in the time domain. See Working with Surface Spectra Field Functions.
4. Open the **Tools** > **Field Functions** node and select the *Time History* field function for your scalar. Set the **Filter Type** property to **Band pass Filter** or **Band stop Filter**.
5. To determine the necessary number of coefficients, use a point FFT for guidance:
   a. Create a Point Time Fourier Transform data set function for any point, specify Filter Type to Band pass Filter, and set the cut-off in the Ideal Filter node to the same as specified in the Time History field function.
   b. Create a Point Inverse Time Fourier Transform data set function to transfer the Fourier transformed signal back to the time domain.
   c. Create a Point Time History data set function for the point. Specify the Filter Type to Band pass Filter, and set the cut-off in the Ideal Filter node to the same as specified in the Point Time Fourier Transform data set function.
   d. Plot the inverse FFT signal and the FIR filtered signal in the same figure. Increase the number of Filter coefficients for the FIR signal until it is sufficiently close to the inverse FFT signal. Use this number of filter coefficients in the Time History field function. As the number of coefficients increases, the two results converge. To change the number of coefficients in the FIR filter, select the **Time History** > **FIR Filter** node and set the *Filter Coefficients* property. The default number is 10.
6. Once you have determined the number of coefficients for an FIR filter that give results equivalent to the ideal filter plus inverse FFT, use that number of coefficients for an FIR filter that is applied to the entire surface.

What Is Derived Data?

Derived data objects allow a data set function to access data from various other objects in the simulation, as well as from external files.

These objects are added under the **Imported**, **Tabular**, and **Monitor** child nodes of a data set function.



The child nodes of the **Imported**, **Tabular**, and **Monitor** nodes represents the derived data set. The input data is specified in the properties of these nodes. Once the properties are defined, the derived data (that is, the transform, convolution or filter function) are computed. The sub-nodes of a data set function can hold as many derived data sets as the memory of the computer permits.

This section explains the following:

Contents:

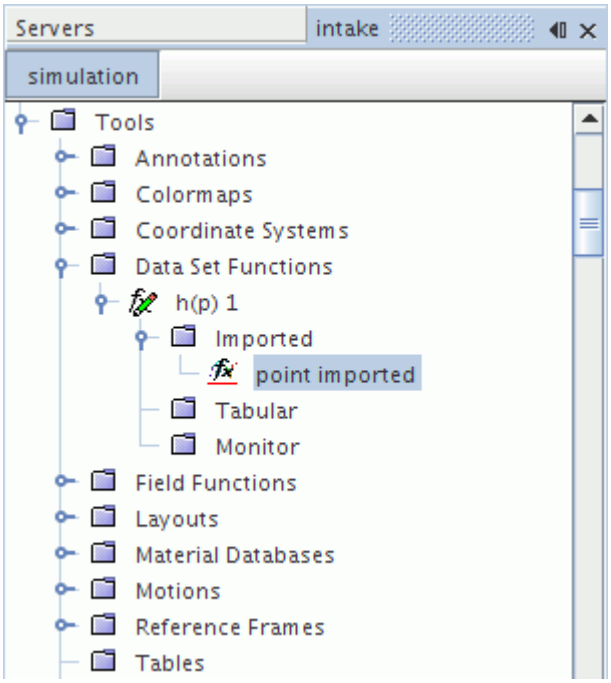
What Are Derived Point Imported Model Data Sets?
What Are Derived Cross Imported Model Data Sets?
Creating Derived Point or Cross Imported Model Data Sets
What Are Derived Imported Surface Data Sets?
Creating Derived Imported Surface Data Sets

What Are Derived Transient Surface Data Sets?
Creating Derived Transient Surface Data Sets
What Are Derived Tabular Data Sets?
What Are Derived Cross Tabular Data Sets?
Creating Derived Tabular or Cross Tabular Data Sets
What Are Derived Monitor Data Sets?
What Are Derived Cross Monitor Data Sets?
Creating Derived Monitor or Cross Monitor Data Sets
What Are Point Derived Data Sets?
Creating Point Derived Data Sets
Calculating the Line Time Fourier Transform
Calculating the Line Spatial Fourier Transform

What Are Derived Point Imported Model Data Sets?

Derived point imported model data sets hold a time history, FIR Filter function or Fourier transform.

They are used to visualize data at a point on a surface that is imported from a `.trn` file (see also Importing Surface Spectral Data) or the Fourier transform. The **point imported** node represents a derived point imported model data set in the simulation tree.



This object cannot be deleted while a link is in place from derived data sets in a plot. This object shares pop-up menu items with other derived data sets.

The function definition requires:

- Importing the surface from an existing `.trn` file.

- Extracting a field function from that surface.
- Selecting a point part as a probe.

The above quantities can be specified using the properties of this object. If the *Update Active* property is activated, the point time history or point time Fourier transform is calculated after the last property assignment. Alternatively, you can calculate the transform by right-clicking the **point imported** node and selecting `Update Output`, irrespective of the state of the *Update Active* property.

Derived point imported model data sets are not updated automatically when you change a part position. A modified part can have many associated derived point imported model data sets. An automatic update would take some time and cause the STAR-CCM+ user interface to lock until the process is complete.

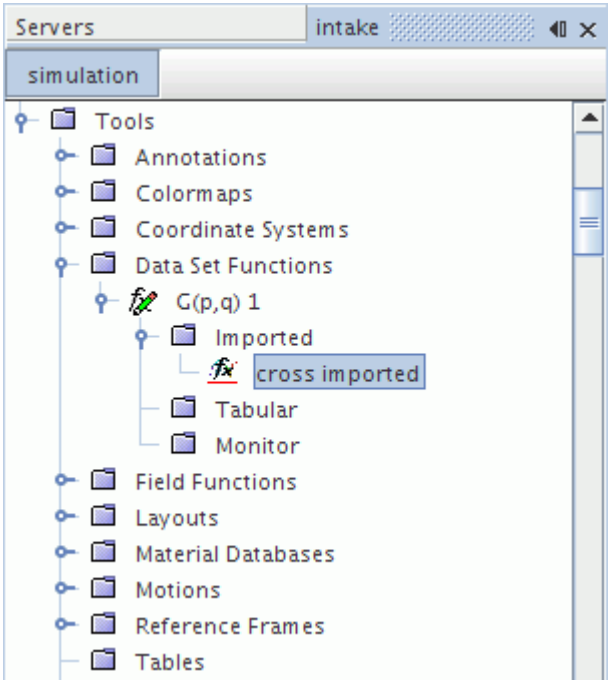
When any part is modified, all of its derived point imported model data sets are shown as requiring an update. The *Update Active* property for the affected derived point imported model data sets is automatically deactivated.

- To update a derived point imported model data set, do one of the following:
  - In the *Properties* window, manually activate the *Update Active* property.
  - Right-click the derived point imported model data set and then click **Update Output**.

What Are Derived Cross Imported Model Data Sets?

Derived cross imported model data sets hold correlations from two input signals.

The **cross imported** node represents a derived cross imported model data set in the simulation tree.



This object cannot be deleted while a link is in place from derived data sets in a plot. This object shares pop-up menu items with other derived data sets.

The function definition requires:

- Importing the surface from an existing `.trn` file.
- Extracting a field function from that surface.
- Selecting a point part as a probe.

The above quantities can be specified using the properties of this object. If the *Update Active* property is activated, the point-to-point time Fourier transform or the point-to-point time correlation is calculated after the last property assignment. Alternatively, you can calculate the transform by right-clicking the **cross imported** node and selecting `Update Output`, irrespective of the state of the *Update Active* property.

Derived cross imported model data sets are not updated automatically when you change a part position. A modified part can have many associated derived cross imported model data sets. An automatic update would take some time and cause the STAR-CCM+ user interface to lock until the process is complete.

When any part is modified, all of its derived cross imported model data sets are shown as requiring an update. The *Update Active* property for the affected derived cross imported model data sets is automatically deactivated.

- To update a derived cross imported model data set, do one of the following:
  - In the *Properties* window, manually activate the *Update Active* property.
  - Right-click the derived cross imported model data set and then click **Update Output**.

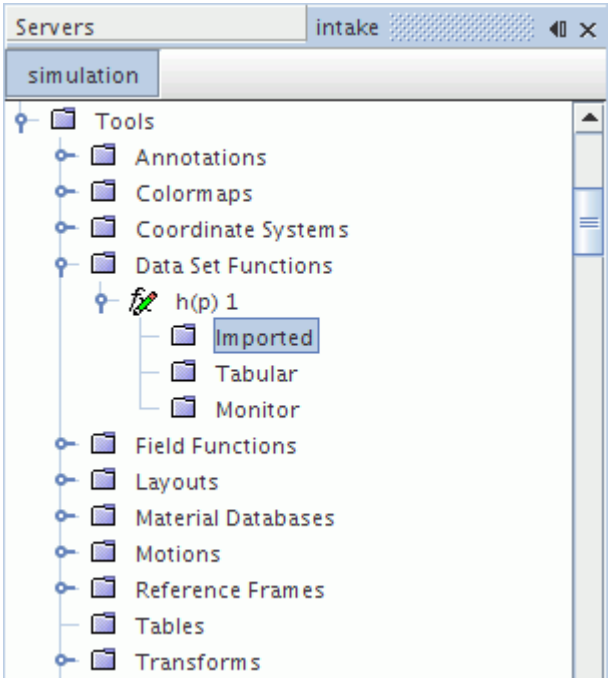
Cross Imported Properties

Field Function 2	Defines the field function that is extracted. Only field functions added by a `.trn` file can be selected.	
Imported surface 2	Defines the imported surface from which data is extracted. Only surfaces from a `.trn` file can be selected.	
Point part 2	Defines the point x'on the surface where data is extracted. This point is the point on the surface closest to a selected point derived part.	
Point part 1	Defines the point x on the surface where data is extracted. This point is the point on the surface closest to a selected point derived part.	
Field Function 1	Defines the field function that is extracted. Only field functions added by a `.trn` file can be selected.	
Imported surface 1	Defines the imported surface from which data is extracted. Only surfaces from a `.trn` file can be selected.	
X Units	Displays the calculated units for the x-axis data.	
Y Units	Displays the calculated units for the y-axis data.	
Update Active	Whether the update is active for the imported point. When a part is modified, this property is automatically deactivated for all of the affected derived cross imported model data sets.	
	Activated	The update is active.
	Deactivated	The update is inactive.

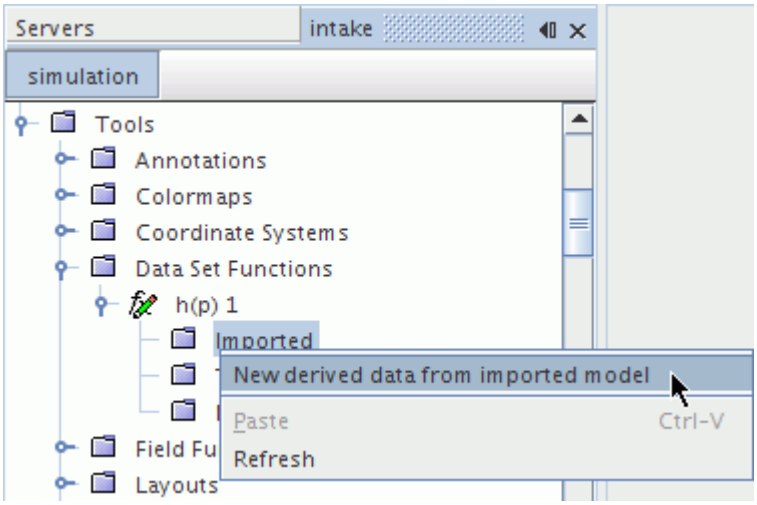
Creating Derived Point or Cross Imported Model Data Sets

Derived point or cross imported data sets are represented as child nodes of the **Imported** manager.

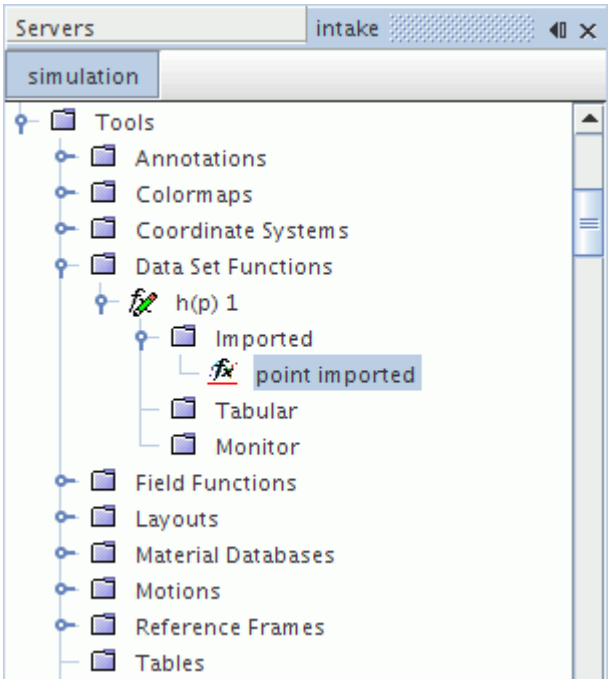
This manager has a pop-up menu.



To create a derived point or cross imported data set, right-click the **Imported** node and select **New derived data from imported model**.



A point imported or cross imported node appears as a child to this node.



Imported Properties

None.

Imported Pop-Up Menu

New derived data from imported model	Creates a point imported or cross imported child node in which data sets can be imported.
Paste	Places the contents of the clipboard into the selected node.
Refresh	Refreshes the sub-nodes of the selected node in the simulation tree.

See also common menu items.

Point Imported Properties

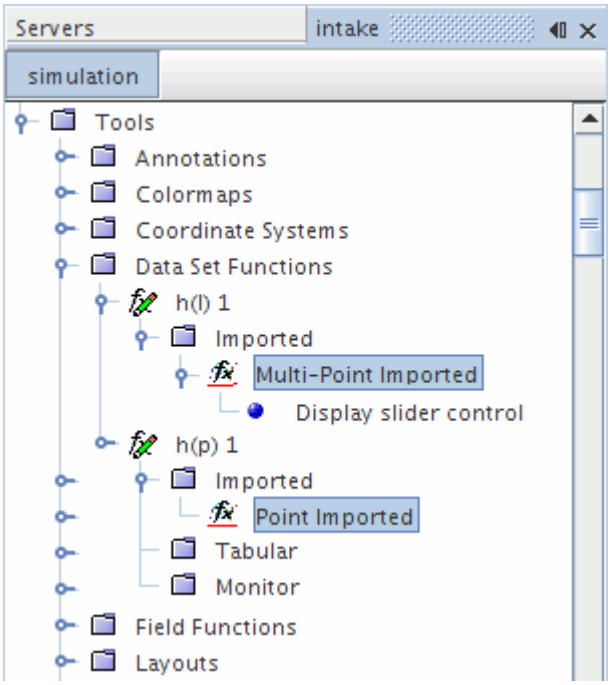
Data Directory	Specifies a file storing the derived data for later use. Data is saved in Mat7.3 format and is HDF5-compatible.
Point part 1	Defines the point x on the surface where data is extracted. This point is the point on the surface closest to a selected point derived part.
Field Function 1	Defines the field function that is extracted. Only field functions added by a `.trn` file can be selected.
Imported surface 1	Defines the imported surface from which data is extracted. Only surfaces from a `.trn` file can be selected.
X Units	Displays the calculated units for the x-axis data.

Y Units	Displays the calculated units for the y-axis data.	
Update Active	Whether the update is active for the imported point. When a part is modified, this property is automatically deactivated for all of the affected derived point imported model data sets.	
	Activated	The update is active.
	Deactivated	The update is inactive.

What Are Derived Imported Surface Data Sets?

Derived imported surface data sets hold surface time histories.

The **Point imported** node represents a derived imported surface data set in the simulation tree. The **Multi-Point imported** node represents a data set derived from imported surface data and a derived part.



This object cannot be deleted while a link is in place from derived data sets in a plot. This object shares pop-up menu items with other derived data sets.

The function definition requires:

- Importing the surface from an existing `.trn` file.
- Extracting a field function from that surface.
- Selecting a derived part or a point part as a probe.

The above quantities can be specified using the properties of this object. If the *Update Active* property is activated, the point-to-point time Fourier transform or the point-to-point time correlation is calculated after the last property assignment. Alternatively, you can calculate the transform by right-clicking the **Point imported** or

Multi-Point imported node and selecting `Update Output`, irrespective of the state of the *Update Active* property.

Derived imported surface data sets are not updated automatically when you change a part position. A modified part can have many associated derived imported surface data sets. An automatic update would take some time and cause the STAR-CCM+ user interface to lock until the process is complete.

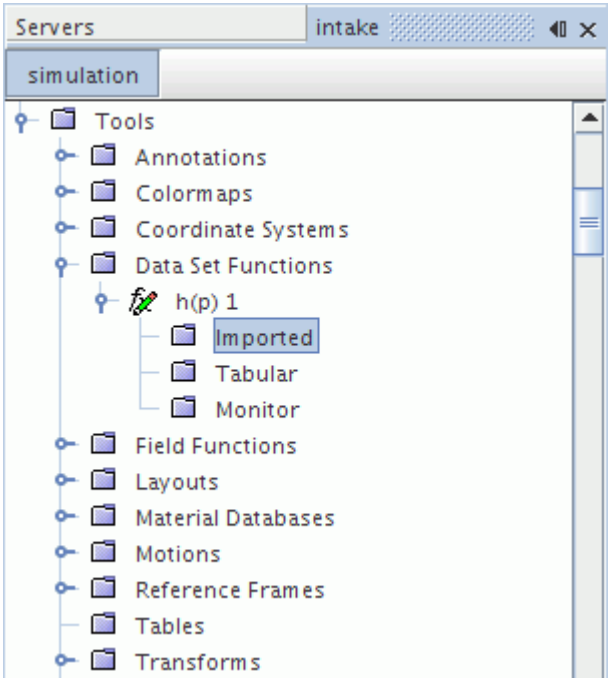
When any part is modified, all of its derived imported surface data sets are shown as requiring an update. The *Update Active* property for the affected derived imported surface data sets is automatically deactivated.

- To update a derived imported surface data set, do one of the following:
  - In the *Properties* window, manually activate the *Update Active* property.
  - Right-click the imported surface data set and then click **Update Output**.

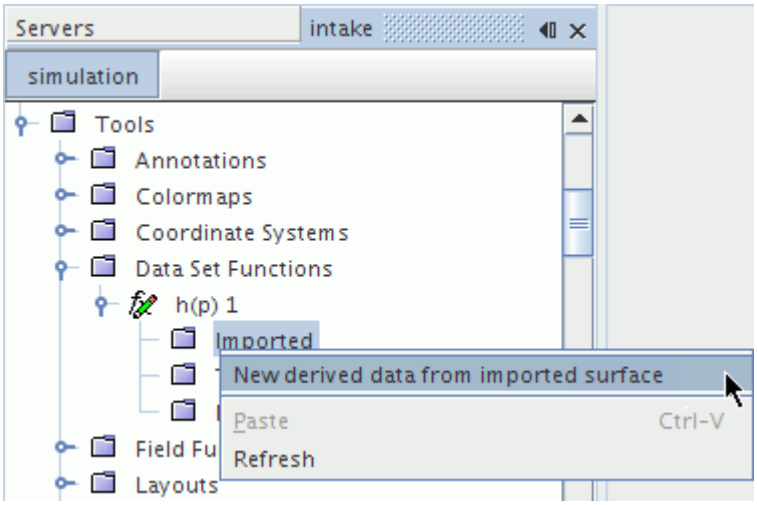
Creating Derived Imported Surface Data Sets

Derived imported surface data sets are represented as child nodes of the **Imported** manager.

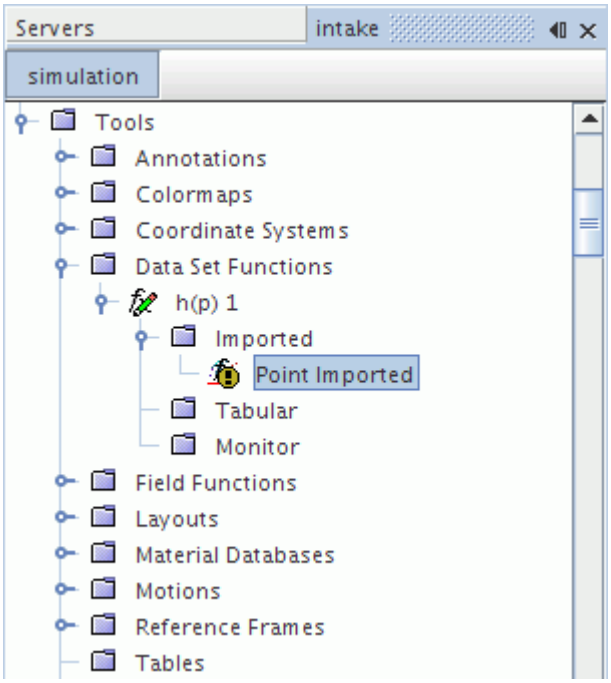
This manager has a pop-up menu.



To create a derived imported surface data set, right-click the **Imported** node and select **New derived data from imported surface** or **New derived data from imported surface & derived part**



A **Point Imported** or **Multi-Point Imported** node appears as a child to this node.



Imported Properties

None.

Imported Pop-Up Menu

New derived data from imported surface	Creates a Point Imported or Multi-Point Imported child node in which data sets can be imported.

See also common menu items.

Multi-Point Imported Properties

Part	Specifies the derived part on the surface where data is extracted.	
Field Function 1	Defines the field function that is extracted. Only field functions added by a `.trn` file can be selected.	
Data surfaces 1	Defines the imported surfaces from which data is extracted. Only surfaces from a `.trn` file can be selected.	
Update Active	Whether the update is active for the imported point. When a part is modified, this property is automatically deactivated for all of the affected derived imported surface data sets.	
	Activated	The update is active.
	Deactivated	The update is inactive.

Display slider control Properties

Slider Value	Determines the current display time. The slider can vary from the maximum to the minimum time of the current signal.

Point Imported Properties

Point Part 1	Defines the point x on the surface where data is extracted. This point is the point on the surface closest to a selected point derived part.	
RMS Magnitude	Specifies the size of the root mean square value of the signal.	
Field Function 1	Defines the field function that is extracted. Only field functions added by a `.trn` file can be selected.	
Data Surfaces 1	Defines the imported surfaces from which data is extracted. Only surfaces from a `.trn` file can be selected.	
X Units	Displays the calculated units for the x-axis data.	
Y Units	Displays the calculated units for the y-axis data.	
Update Active	Whether the update is active for the imported point. When a part is modified, this property is automatically deactivated for all of the affected derived imported surface data sets.	
	Activated	The update is active.
	Deactivated	The update is inactive.

What Are Derived Transient Surface Data Sets?

Derived transient surface data sets hold fields that can be mapped onto unstructured surfaces.

Using them, you can:

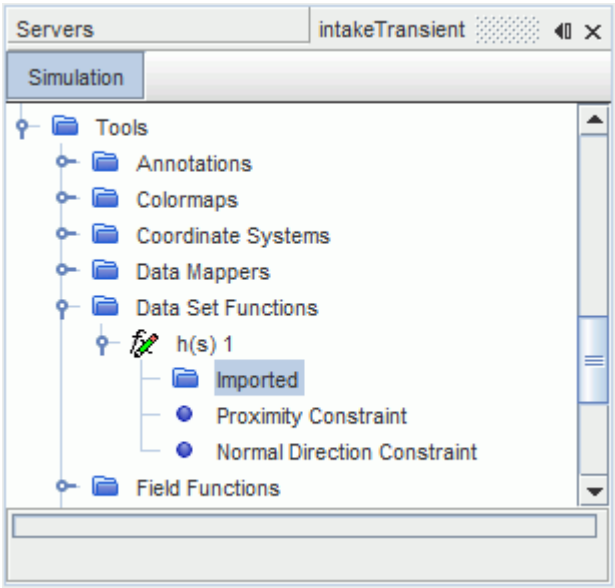
- Map results on `.trn` file meshes at all time steps onto the `.trn` file mesh at a single time step.
- Map results on a mesh from a `.trn` file onto an external mesh.

The `.mat` file format can store unstructured surface meshes as well as structured meshes.

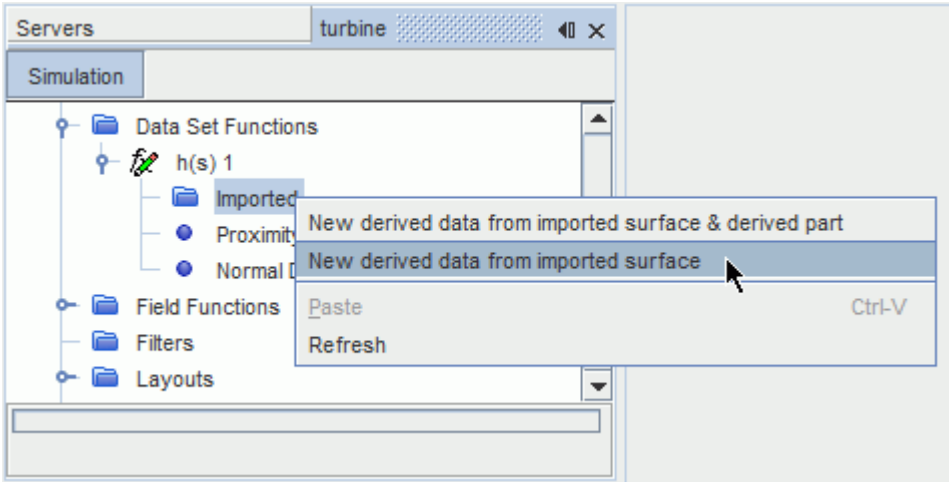
Creating Derived Transient Surface Data Sets

Derived transient surface data sets are represented as child nodes of the **Imported** manager.

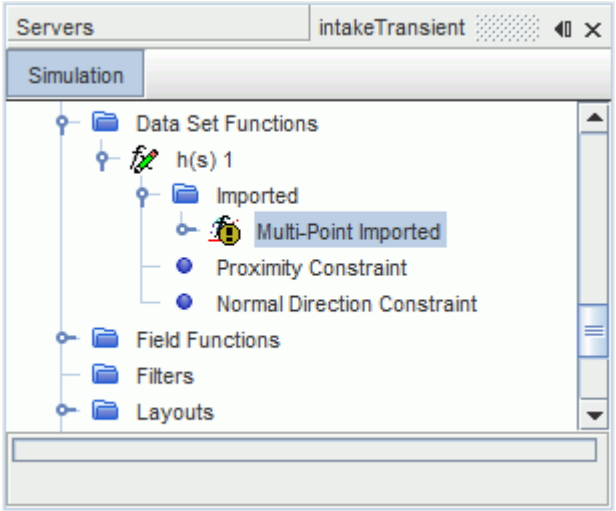
This manager has a pop-up menu.



To create a derived transient surface data set, right-click the **Imported** node and select **New derived data from imported surface** or **New derived data from imported surface & derived part**.



A **Multi-Point Imported** node appears as a child to this node.



Imported Properties

None.

Imported Pop-Up Menu

New derived data from imported surface	Creates a Multi-Point Imported child node in which data sets can be imported.

Multi-Point Imported Properties

Target Surfaces	Specifies the surfaces to which the field function is mapped.	
Field Function 1	Defines the field function to be mapped. Only field functions added by a `.trn` file can be selected.	
Data surfaces 1	Defines the list of surfaces from which data is mapped. Only surfaces from a `.trn` file can be selected.	
Update Active	Specifies whether the update is active for the imported point. When a part is modified, this property is automatically deactivated for all of the affected derived transient surface data sets.	
	Activated	The update is active.
	Deactivated	The update is inactive.

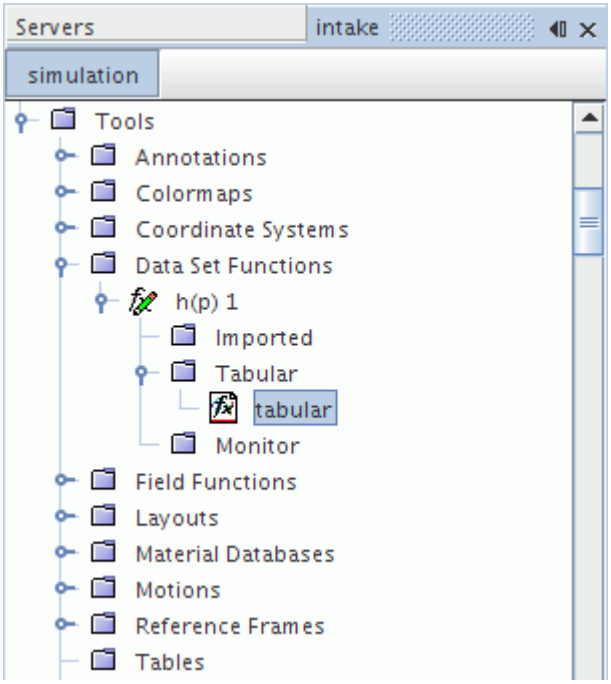
Display Slider Control Properties

Slider Value	Determines the current display time. The slider can vary from the maximum to the minimum time of the current signal.

What Are Derived Tabular Data Sets?

Derived tabular data sets allow a Point Time Fourier Transform or Point Time History to access tables that you read into this simulation.

The tables can contain experimental data that you would use to validate simulation results.



This object cannot be deleted while a link is in place from derived monitor data sets in a plot. This object shares pop-up menu items with other derived data sets.

To access external data, carry out the following steps:

1. Read the data into a table in STAR-CCM+.
2. Select that particular table in the properties of this object.
3. In the same properties, assign the data columns of the table to the x- and y-axes.

The above quantities can be specified using the properties of this object. If the *Update Active* property is activated, the point time history or point time Fourier transform is calculated after the last property assignment. Alternatively, you can calculate the transform by right-clicking the **tabular** node and selecting `Update Output`, irrespective of the state of the *Update Active* property.

Tabular Properties

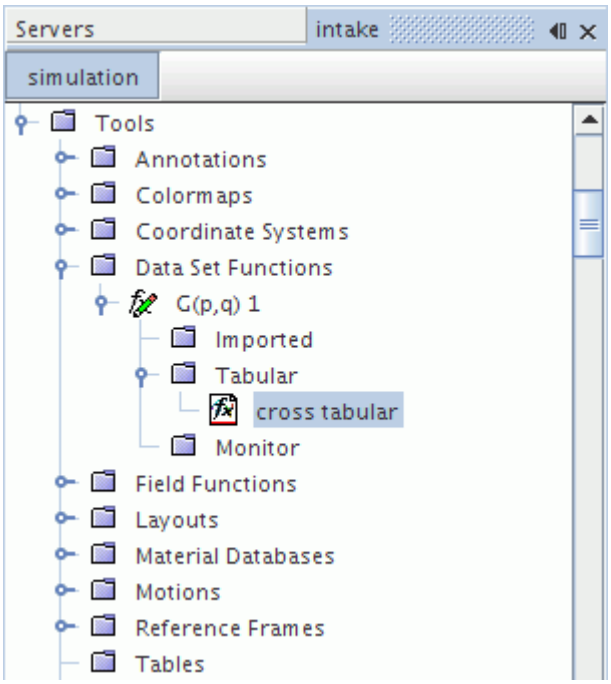
Input Data	Specifies the table to use. The available tables are represented as nodes in the **Tables** node of the **Tools** node.
X Column 1	Selects a column in the table to plot on the x-axis.
Y Column 1	Selects a column in the table to plot on the y-axis.
RMS Magnitude	Specifies the size of the root mean square value of the signal.

X Units	Displays the calculated units for the x-axis data.	
Y Units	Displays the calculated units for the y-axis data.	
Update Active	Whether the update is active for the imported table.	
	Activated	The update is active.
	Deactivated	The update is inactive. This state is the default.

What Are Derived Cross Tabular Data Sets?

Derived cross tabular data sets allow correlations to access tables that you read into this simulation.

The tables can contain experimental data that you would use to validate simulation results.



This object cannot be deleted while a link is in place from derived monitor data sets in a plot. This object shares pop-up menu items with other derived data sets.

To access external data, carry out the following steps:

1. Read the data into a table in STAR-CCM+.
2. Select that particular table in the properties of this object.
3. In the same properties, assign the data columns of the table to the x- and y-axes.

The above quantities can be specified using the properties of this object. If the *Update Active* property is activated, the point-to-point time Fourier transform or the point-to-point time correlation is calculated after the last property assignment. Alternatively, you can calculate the transform by right-clicking the **cross tabular** node and selecting `Update Output`, irrespective of the state of the *Update Active* property.

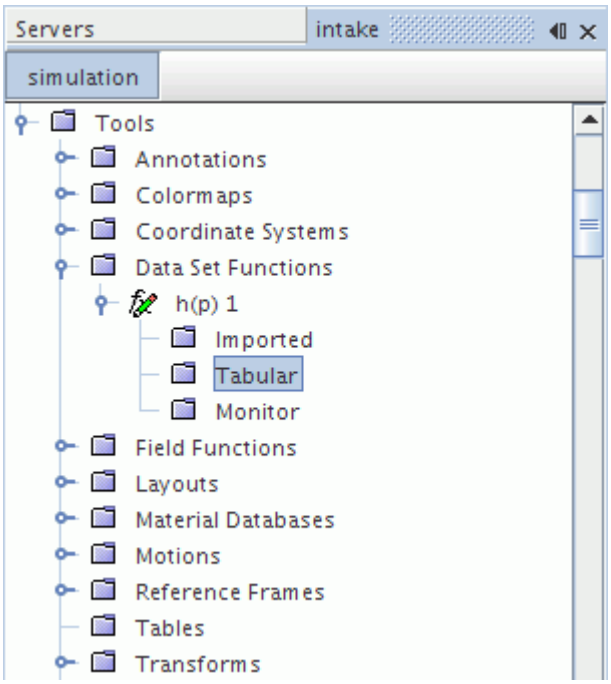
Cross Tabular Properties

Input Data 2	Selects a table from among the ones that are represented by nodes in the **Tables** node of the **Tools** node.	
X Column 2	Selects a column in the table to plot on the x-axis.	
Y Column 2	Selects a column in the table to plot on the y-axis.	
Input Data 1	Selects a table from among the ones that are represented by nodes in the **Tables** node of the **Tools** node.	
X Column 1	Selects a column in the table to plot on the x-axis.	
Y Column 1	Selects a column in the table to plot on the y-axis.	
X Units	Displays the calculated units for the x-axis data.	
Y Units	Displays the calculated units for the y-axis data.	
Update Active	Whether the update is active for the imported table.	
	Activated	The update is active.
	Deactivated	The update is inactive. This state is the default.

Creating Derived Tabular or Cross Tabular Data Sets

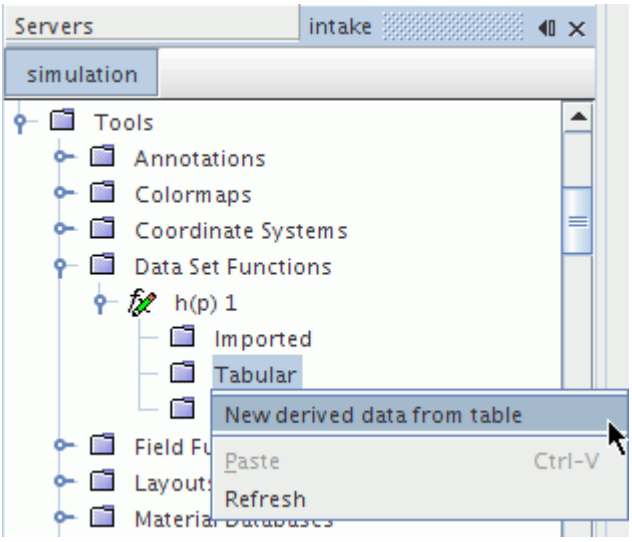
Derived tabular and cross tabular data sets are represented as child nodes of the **Tabular** manager.

This manager has a pop-up menu.

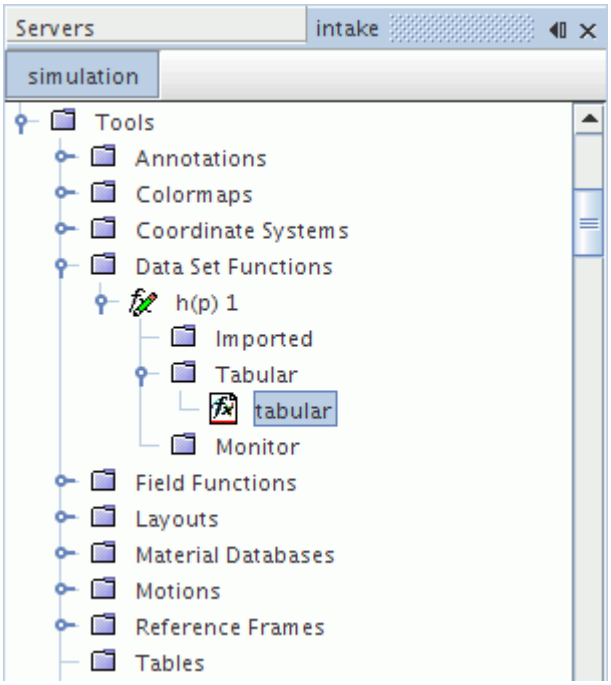


To create a derived tabular or cross tabular data set:

- Right-click the **Tabular** node and select **New derived data from table**.



A tabular or cross tabular node appears as a child to this node.



Tabular Properties

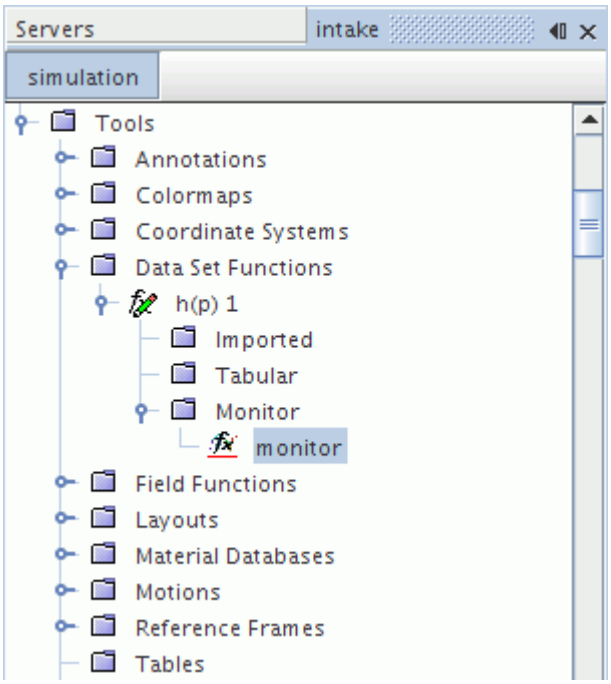
None.

Tabular Pop-Up Menu

New derived data from table	Creates a tabular or cross tabular child node in which tables can be accessed.
Paste	Places the contents of the clipboard into the selected node.
Refresh	Refreshes the sub-nodes of the selected node in the simulation tree.

What Are Derived Monitor Data Sets?

Derived monitor data sets allow a Fourier Transform or time history function to access the monitors that you create for this simulation.



This object cannot be deleted while a link is in place from derived data sets in a plot. This object shares pop-up menu items with other derived data sets.

To access a particular monitor, simply select it in the properties of this object. If the *Update Active* property is activated, the point time history or point time Fourier transform is calculated after the assigning a particular monitor. Alternatively, you can calculate the transform by right-clicking the **monitor** node and selecting `Update Output`, irrespective of the state of the *Update Active* property.

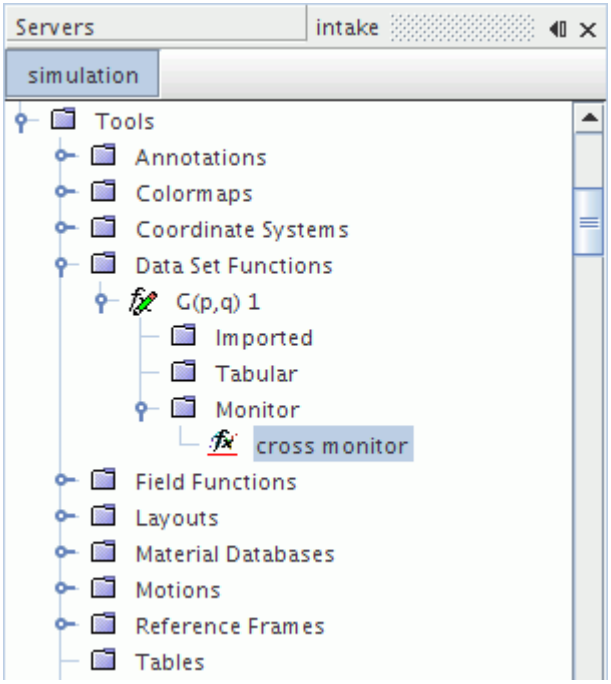
Monitor Properties

Data Directory	Specifies a file storing the derived data for later use. Data is saved in Mat7.3 format and is HDF5-compatible.
Input Data 1	The monitor that is the source of the data of this object.

X Units	Displays the calculated units for the x-axis data.	
Y Units	Displays the calculated units for the y-axis data.	
Update Active	Whether the update is active for the imported monitor.	
	Activated	The update is active.
	Deactivated	The update is inactive. This state is the default.

What Are Derived Cross Monitor Data Sets?

Derived cross monitor data sets allow a correlation to access the monitors that you create for this simulation.



This object cannot be deleted while a link is in place from derived data sets in a plot. This object shares pop-up menu items with other derived data sets.

To access a particular monitor, simply select it in the properties of this object. If the *Update Active* property is activated, the point-to-point time Fourier transform or the point-to-point time correlation is calculated after the monitor assignment. Alternatively, you can calculate the transform by right-clicking the **cross monitor** node and selecting `Update Output`, irrespective of the state of the *Update Active* property.

Cross Monitor Properties

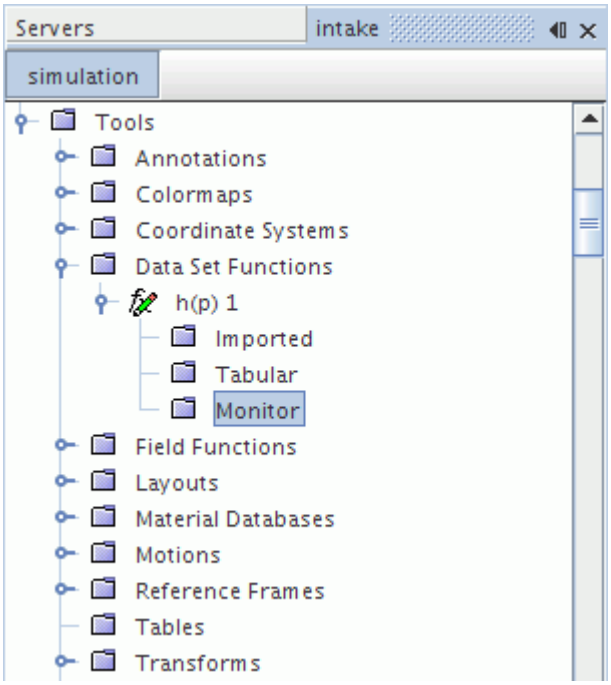
Input Data 2	The monitor that is the source of the data of this object.
Input Data 1	The monitor that is the source of the data of this object.
X Units	Displays the calculated units for the x-axis data.

Y Units	Displays the calculated units for the y-axis data.	
Update Active	Whether the update is active for the imported monitor.	
	Activated	The update is active.
	Deactivated	The update is inactive. This state is the default.

Creating Derived Monitor or Cross Monitor Data Sets

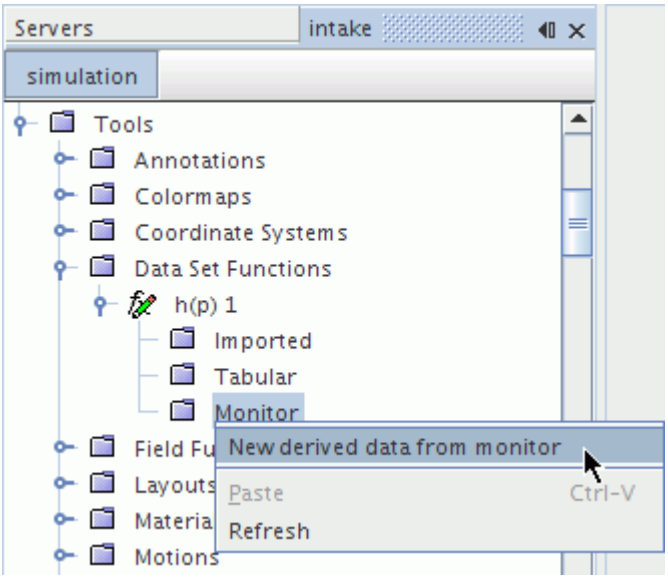
Derived monitor and cross monitor data sets are represented as child nodes of the **Monitor** manager.

This manager has a pop-up menu.

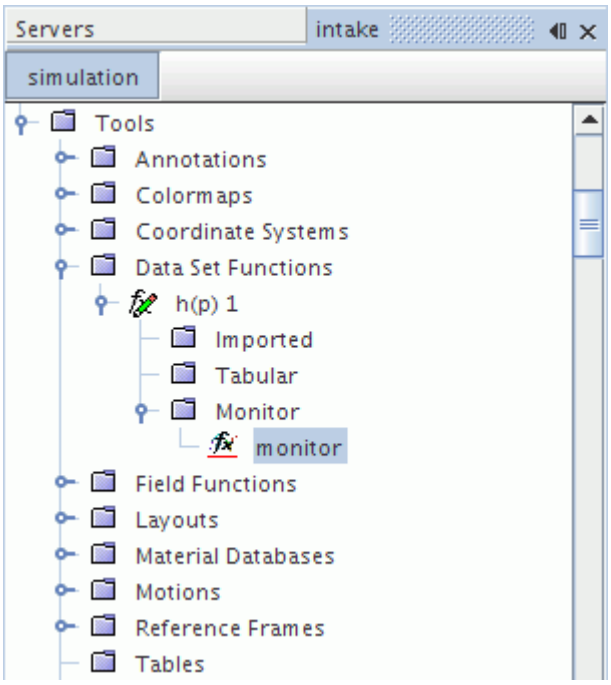


To create a derived monitor or cross tabular monitor set:

- Right-click the **Monitor** node and select **New derived data from monitor**.



A monitor or cross monitor node appears as a child to this node.



Monitor Properties

None.

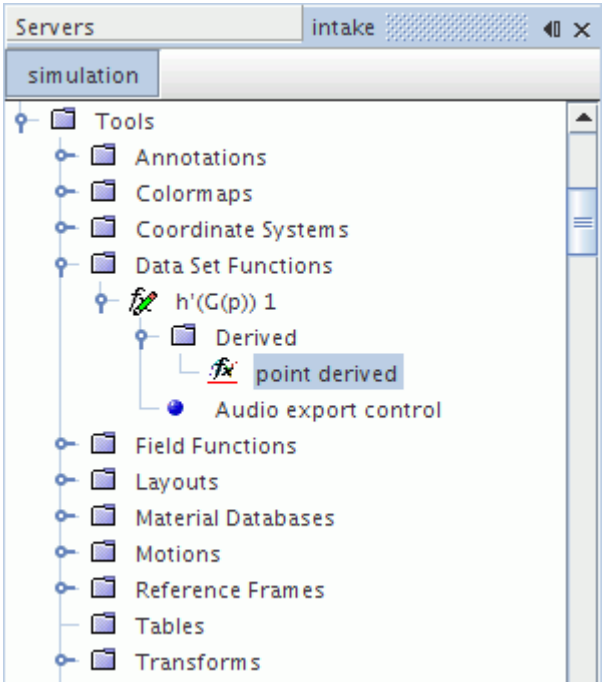
Monitor Pop-Up Menu

New derived data from monitor	Creates a monitor or cross monitor child node in which monitors can be accessed.

Paste	Places the contents of the clipboard into the selected node.
Refresh	Refreshes the sub-nodes of the selected node in the simulation tree.

What Are Point Derived Data Sets?

Derived data sets allow a point inverse time Fourier transform data set function to access any of the existing derived data objects.



This object cannot be deleted while a link is in place from derived data sets in a plot. This object shares pop-up menu items with other derived data sets.

To access a particular derived data object, simply select it in the properties of this object. If the *Update Active* property is activated, the point inverse time Fourier transform is calculated after the derived data assignment. Alternatively, you can calculate the inverse transform by right-clicking the **point derived** node and selecting `Update Output`, irrespective of the state of the *Update Active* property.

Point Derived Properties

Input Data	Accesses a particular type of derived data object from a drop-down list of the derived data objects that currently exist in the simulation.	
	None	Does not use a derived data object.
	Derived imported model	Accesses a derived imported model data set.

	Derived monitor	Accesses a derived monitor data set.
	Derived tabular	Accesses a derived tabular data set.
X Units	Displays the calculated units for the x-axis data.	
Y Units	Displays the calculated units for the y-axis data.	
Update Active	Whether the update is active for the imported data.	
	Activated	The update is active.
	Deactivated	The update is inactive.

Contents:

Derived Data Pop-Up Menu

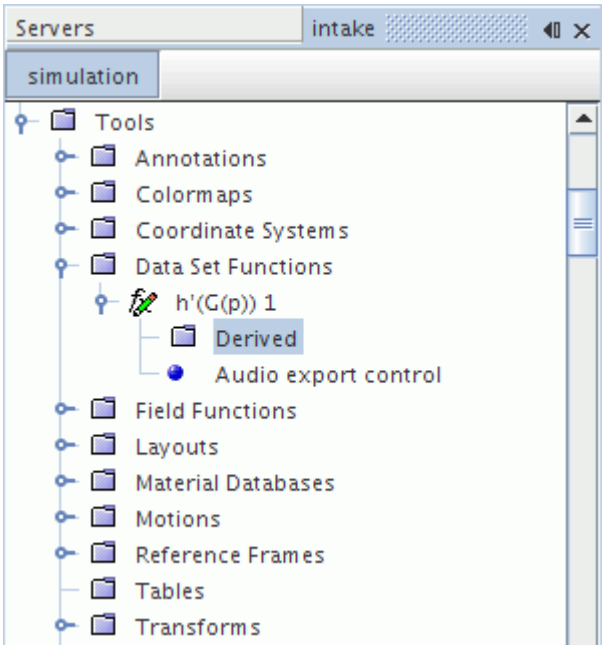
Derived Data Pop-Up Menu

These objects cannot be deleted while a link to a plot data set is in place.

Update Output	Updates the derived data set.
Export...	Exports the data to a `.csv` file.
Export Audio...	Exports an audio (.wav) file. Applies only to derived data nodes for Inverse Fourier Transforms.
Tabulate	Executes the tabulate data action. This action presents the data in a two-column spreadsheet-style view of the data in this data set. The abscissa (horizontal axis) data is in the first column and the ordinate (vertical axis) data is in the second column. When there is no data in the data set, the dialog is empty.

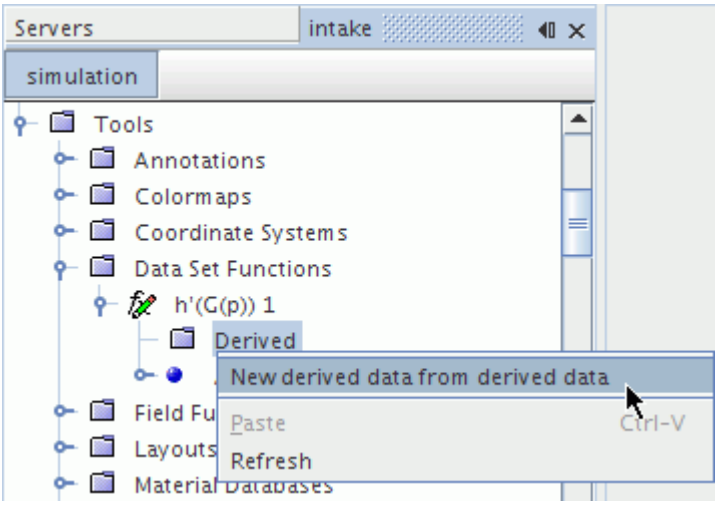
Creating Point Derived Data Sets

A point derived data set is represented as a child node of the **Derived** manager, which has a pop-up menu.

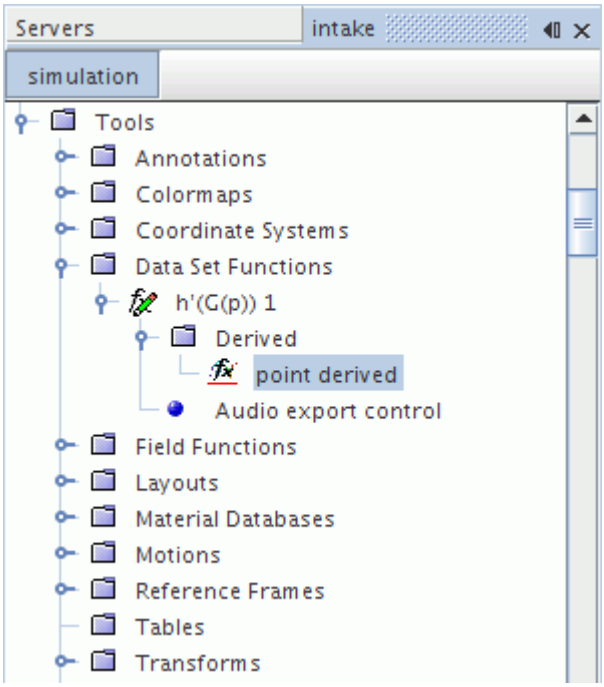


To create a point derived data set:

- Right-click the **Derived** manager node and select **New derived data from derived data**.



A point derived node appears as a child to this node.



Derived Properties

None.

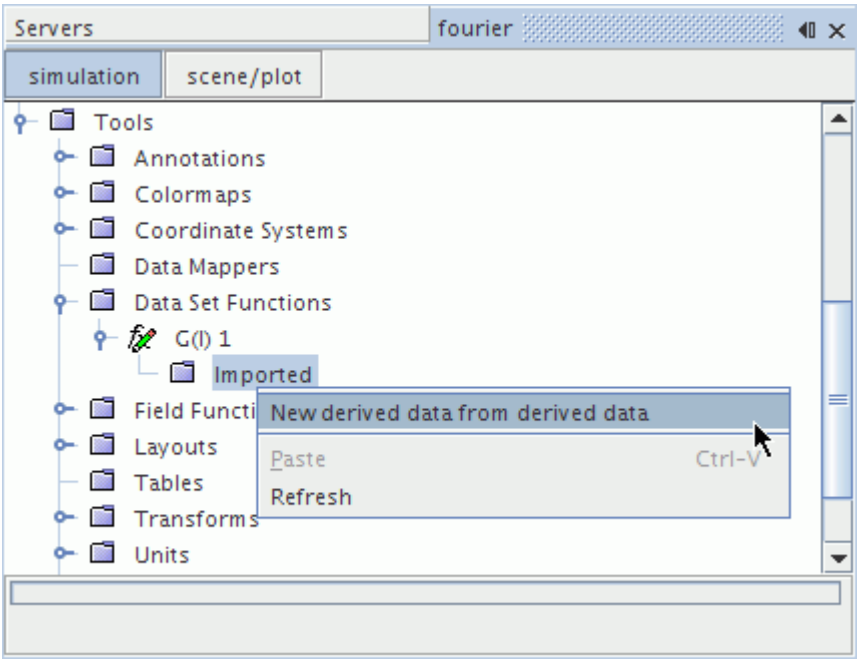
Derived Data Pop-Up Menu

New derived data from derived data	Creates a point derived child node in which monitors can be accessed.
Paste	Places the contents of the clipboard into the selected node.
Refresh	Refreshes the sub-nodes of the selected node in the simulation tree.

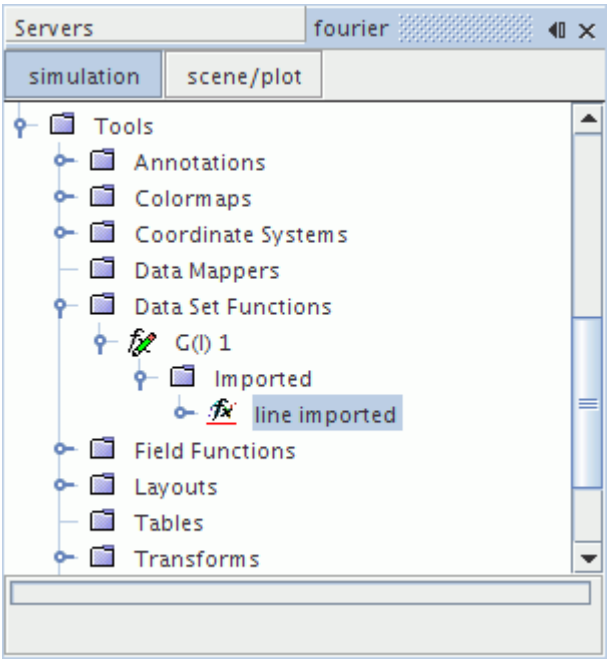
Calculating the Line Time Fourier Transform

Before calculating a line time Fourier transform from a line probe, create a derived data set under the **Imported** child node of the transform:

1. Right-click **Tools** > **Data Set Functions** > **G(I) 1** > **Imported** and select **New derived data from derived data**.



A **line imported** node appears in the simulation tree.



2. Right-click the **line imported** node, select **Export Field...**, and complete the dialog to specify the name and location of the `.mat` file.

Line Imported Properties

Input time history	The line time history **h(In)** to use as input, selected from pull-down menu.

X Units	The units of the X axis.	
Y Units	The units of the Y axis. Equals the square of the units of *Field function 1* divided by the units of frequency for power spectral density plots.	
Update Active	Whether the update is active for the imported line.	
	Activated	The update is active.
	Deactivated	The update is inactive.

If the **Update Active** property is activated, the line time Fourier transform is calculated after the last property assignment. Alternatively, you can calculate the transform by right-clicking the **line imported** node and selecting **Update Output**, irrespective of the state of the **Update Active** property.

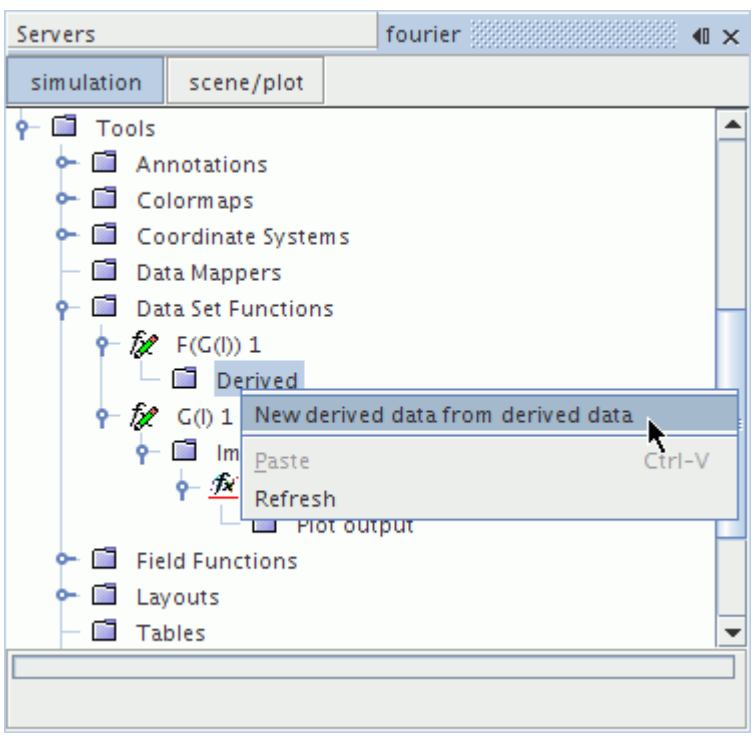
Line Imported Expert Properties

X Surface Axis Units	The units in which the X axis is specified.
Y Surface Axis Units	The units in which the Y axis is specified.
Z Surface Axis Units	The units in which the Z axis is specified.

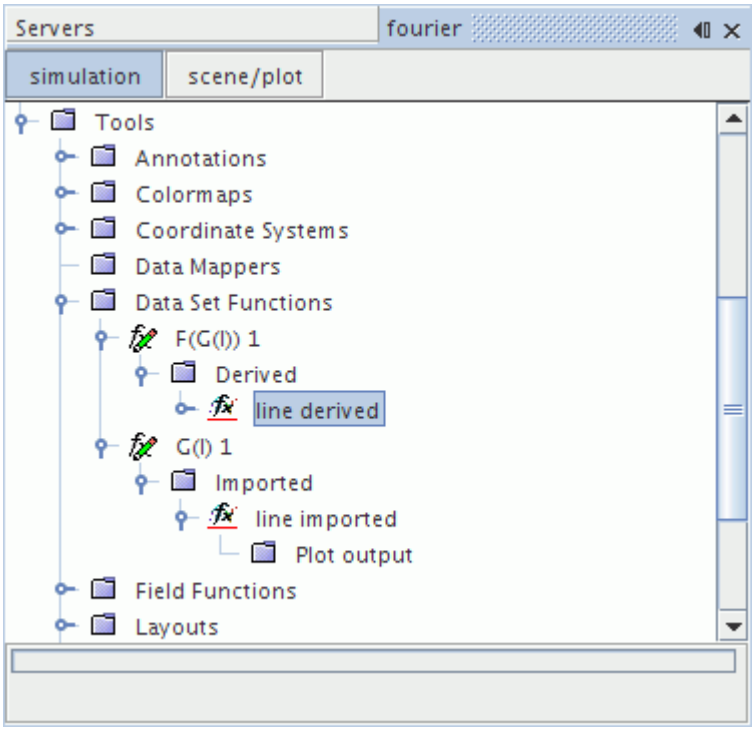
Calculating the Line Spatial Fourier Transform

Before calculating the line spatial Fourier transform from an existing line time Fourier transform, import the data under the **Derived** child node of the line spatial Fourier transform:

1. Right-click **Tools** > **Data Set Functions** > **F(G(I)) 1** > **Derived** and select **New derived data from derived data**.

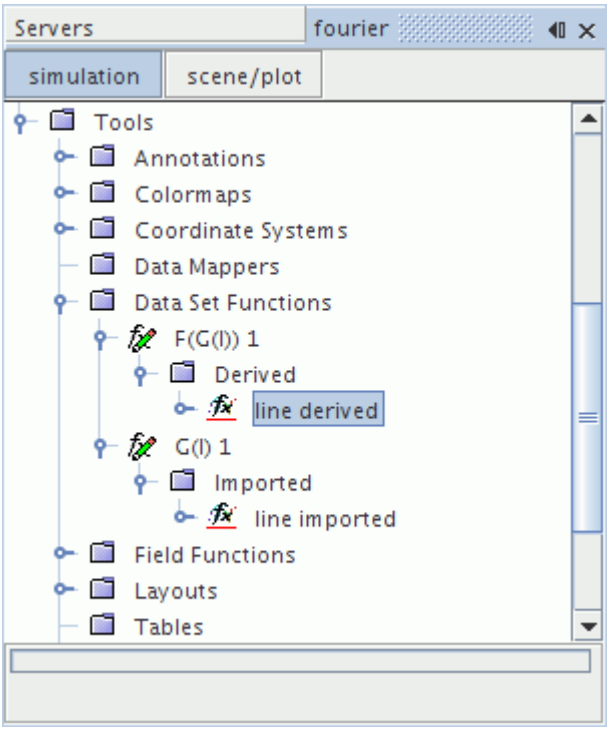


A **line derived** node appears in the simulation tree.



To calculate the line spatial Fourier transform, select the line derived data that was created for the line time Fourier transform:

2. Select the **F(G(I)) 1** > **Derived** > **line derived** node.



3. In the *Properties* window, set *Input derived data* to the derived data you created for the line time Fourier transform.

If the *Update Active* property is activated, the line spatial Fourier transform is calculated after the property assignment. Alternatively, you can calculate the transform by right-clicking the **line derived** node and selecting `Update Output`, irrespective of the state of the *Update Active* property.

Line Derived Properties

Input derived data	The derived data that was created for the line time Fourier transform.	
X Units	The units of the X axis.	
Y Units	The units of the Y axis.	
Update Active	Whether the update is active for the derived line.	
	Activated	The update is active.
	Deactivated	The update is inactive.

Line Derived Expert Properties

X Surface Axis Units	The units in which the X axis is specified.
Y Surface Axis Units	The units in which the Y axis is specified.
Z Surface Axis Units	The units in which the Z axis is specified.

Post-Processing Data Set Functions

STAR-CCM+ allows you to visualize Fourier transforms and surface spectra.

The follow sections explain the following:

Contents:

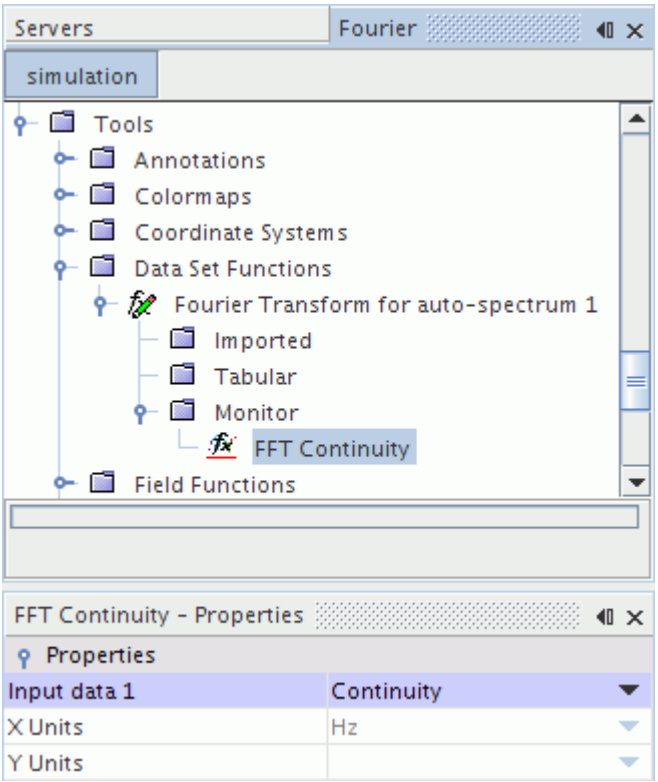
Plotting Fourier Transforms
Setting the Transform Plot Properties of Line Data Sets
Post-Processing Surface Spectra

Plotting Fourier Transforms

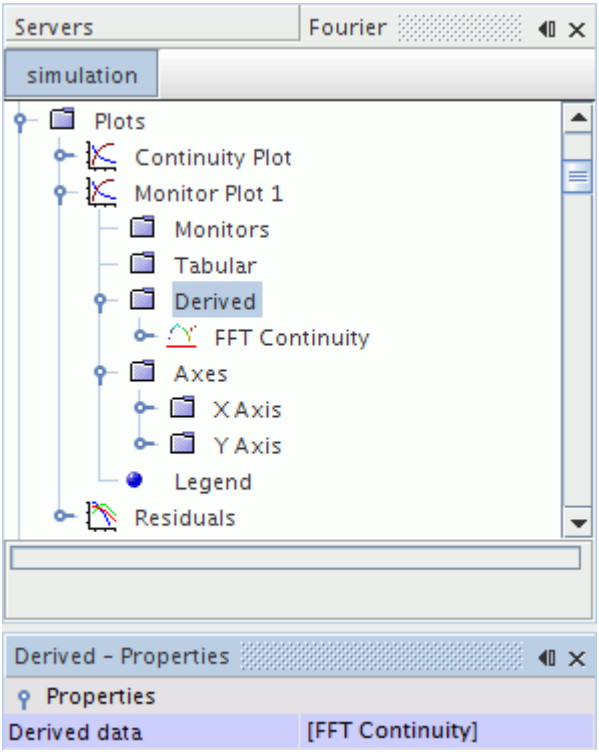
You can plot a Fourier transform from a monitor containing periodic data.

To plot a Fourier Transform:

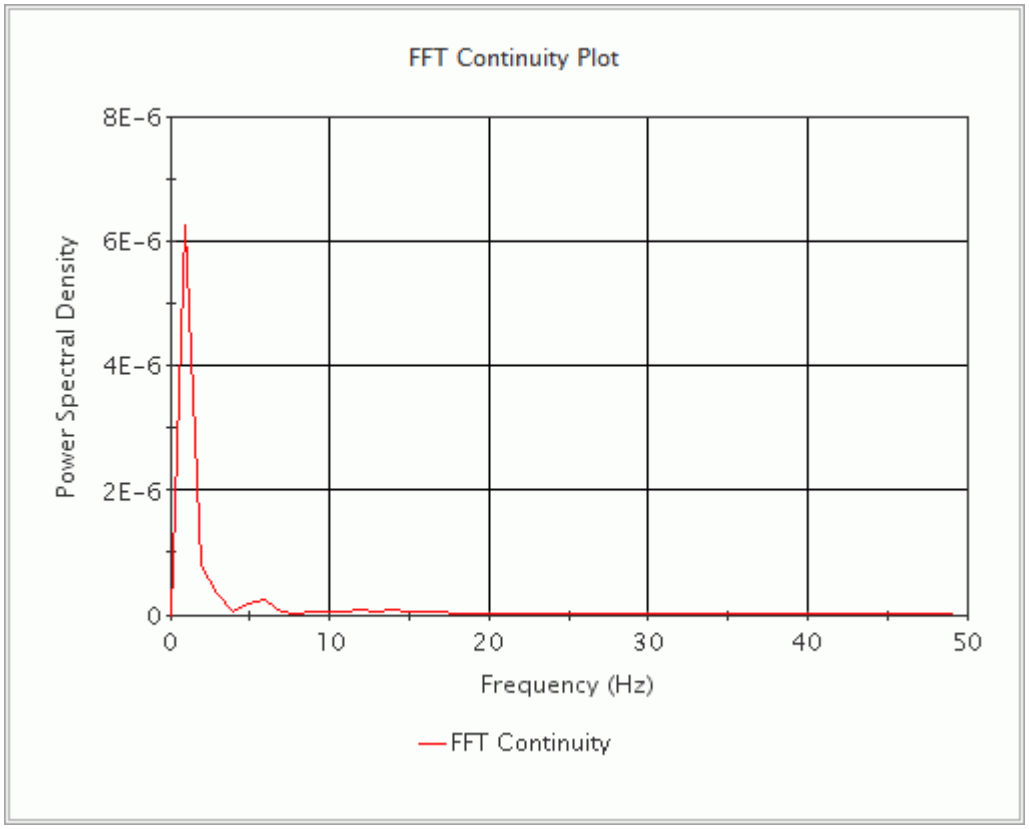
1. Ensure that the *Trigger* property of the monitor is set to `Time Step`.
2. Create a Fourier Transform and set its properties.
3. Right-click the **Monitor** node under this transform and choose **New derived data from monitor**.
4. Give the derived monitor a meaningful name and in the derived monitor *Properties* window, assign the *Input data 1* property to the monitor containing the periodic data.



5. Create a monitor plot after the simulation has run.
6. Click the **Derived** node under this monitor plot, and in the *Properties* window, select the derived data monitor to assign it to the *Derived data* property.

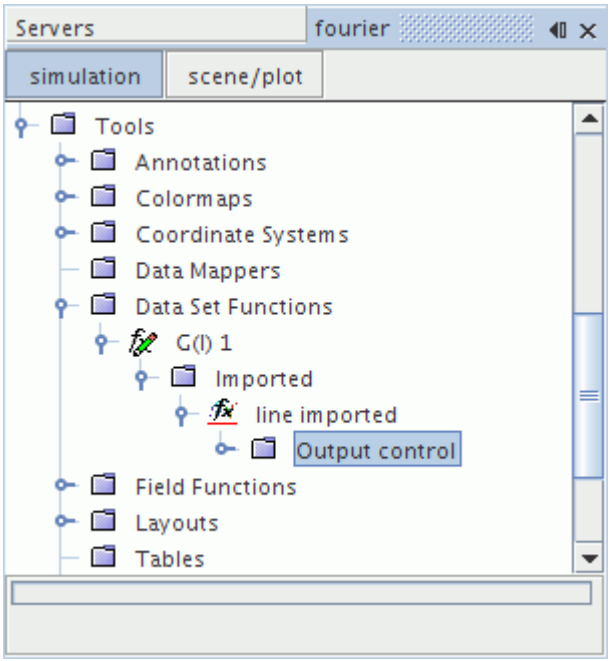


7. Double-click the monitor plot node to display the Fourier Transform plot.



Setting the Transform Plot Properties of Line Data Sets

You can set the plot properties of a Line Time Fourier Transform or a Line Spatial Fourier Transform using the **Output control** node.



Output control Properties

3D profile type	Selects the type of three-dimensional display.	
	F-X spectra surface	The auto- or cross-spectra plotting frequency vs. spatial dimension.
	3D spectra surface	The auto-spectra on 3D geometry for a given frequency.
Output field	Selects which kind of FFT information to visualize.	
	Re{FFT Coefficients}	The real parts of the FFT coefficients.
	Im{FFT Coefficients}	The imaginary parts of the FFT coefficients.
	Power Spectral Density	Frequency-domain description of the signal power.
	Phase{FFT Coefficients}	The phases of the FFT coefficients ($atan[Im/Re]$).
	Abs{FFT Coefficients}	The absolute values of the FFT coefficients ($\sqrt{\lvert \mathrm{Re} \rvert^2 + \lvert Im \rvert^2}$).
	Sound Pressure Level	Sound pressure in decibels. Adds the Reference Sound Pressure node.
	A-Weighted Sound Pressure Level	Weights the `Sound Pressure Level` with the A-scale function,

		for loudness less than 55 phons (at 1000 Hz, 1 phon = 1 dB). This function is typically for problems that involve the frequency range for human hearing. Adds the Reference Sound Pressure node.
	B-Weighted Sound Pressure Level	Weights the `Sound Pressure Level` with the B-scale function, for loudness of 55-85 phons. Adds the Reference Sound Pressure node.
	C-Weighted Sound Pressure Level	Weights the `Sound Pressure Level` with the C-scale function, for loudness greater than 85 phons. Adds the Reference Sound Pressure node.
	Root Mean Square	The square root of the `Power Spectral Density`.
Log data	Displays the log of current data in the current scene or plot.	
	Activated	Log displayed.
	Deactivated	Log not displayed. The default.
2D profile range	Defines the two-dimensional profile.	
	All	Uses all signal frequencies.
	Value	Uses a single value frequency. Adds the Single Value child node to this node.
	Band	Uses a band of frequencies. Adds the Single Band child node to this node.
2D profile axis	Defines the two-dimensional profile.	
	Spatial axis	The abscissa is in length units.
	Frequency axis	The abscissa is in frequency units.

Display slider control Properties

Slider Value	Determines the current display time. The slider can vary from the maximum to the minimum time of the current signal.

Contents:

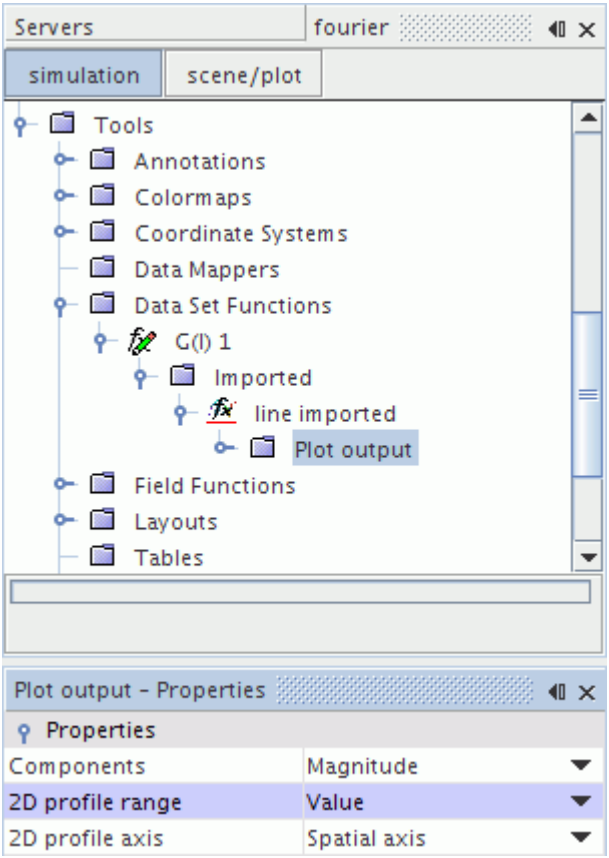
Setting a Two-Dimensional Plot of the Transform Using a Single Frequency
Setting a Two-Dimensional Plot of the Transform Using a Frequency Band

Setting a Two-Dimensional Plot of the Transform Using a Single Frequency

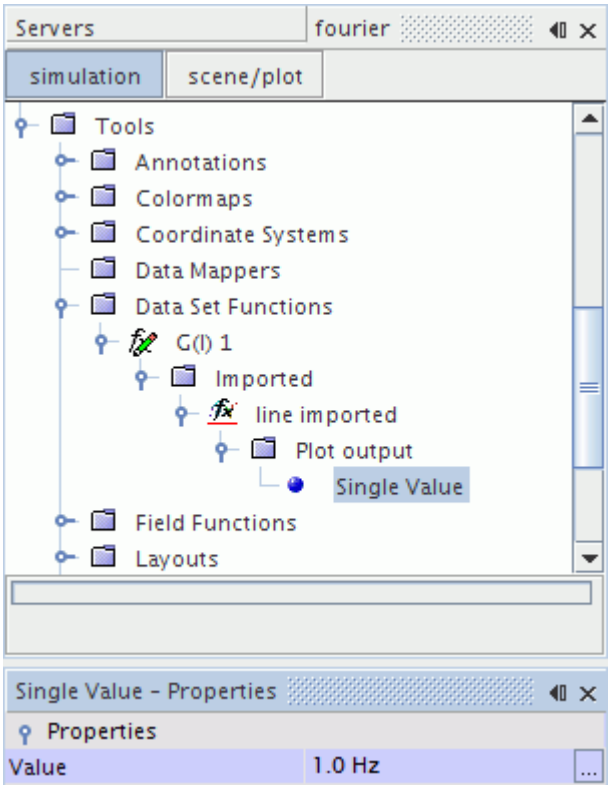
You can use a single frequency to plot a Line Time Fourier Transform or a Line Spatial Fourier Transform in two dimensions.

To plot a Line Time Fourier Transform or a Line Spatial Fourier Transform:

1. Select the **Plot output node** and set its *2D profile range* to **Value**.



The **Single Value** node appears as a child node of **Plot output**.



2. Select the **Plot output** > **Single Value** node and set its *Value* to the required frequency.

Single Value Properties

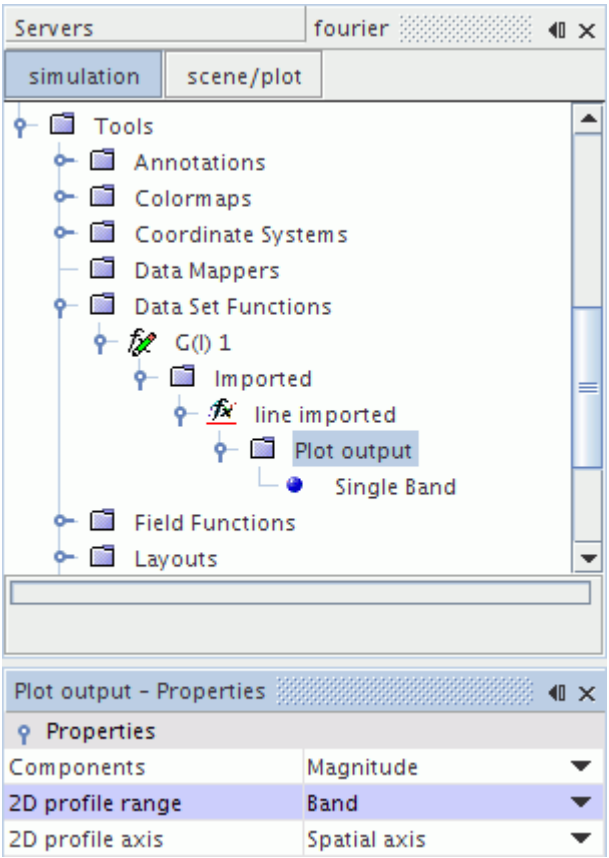
Value	The value of the frequency for which the Fourier transform is plotted in two dimensions. Clicking the ... property customizer button activates a property customizer that lets you specify both the frequency value and the appropriate time units (**/s** or **Hz**).

Setting a Two-Dimensional Plot of the Transform Using a Frequency Band

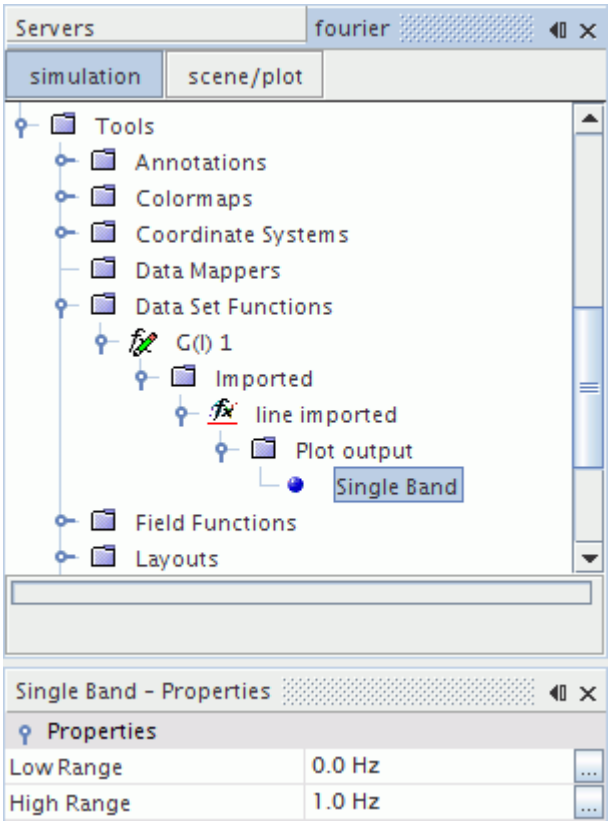
You can use a frequency band to plot a Line Time Fourier Transform or a Line Spatial Fourier Transform in two-dimensions.

To plot a Line Time Fourier Transform or a Line Spatial Fourier Transform:

1. Select the **Plot output node** and set its *2D profile range* to **Band**.



The **Single Band** node appears as a child node of **Plot output**.



2. Select the **Plot output** > **Single Band** node and set its *Value* to the required frequency.

Single Band Properties

Low Range	The value of the lowest frequency for which the Fourier transform is plotted in two dimensions.
High Range	The value of the highest frequency for which the Fourier transform is plotted in two dimensions. Clicking the ... property customizer button activates a property customizer that lets you specify both the frequency value and the appropriate time units (**/s** or **Hz**).

Post-Processing Surface Spectra

The process of setting up the post-processing of surface spectra involves several steps. The .trn file is used to display data set functions.

The procedure to post-process and visualize surface spectra is as follows:

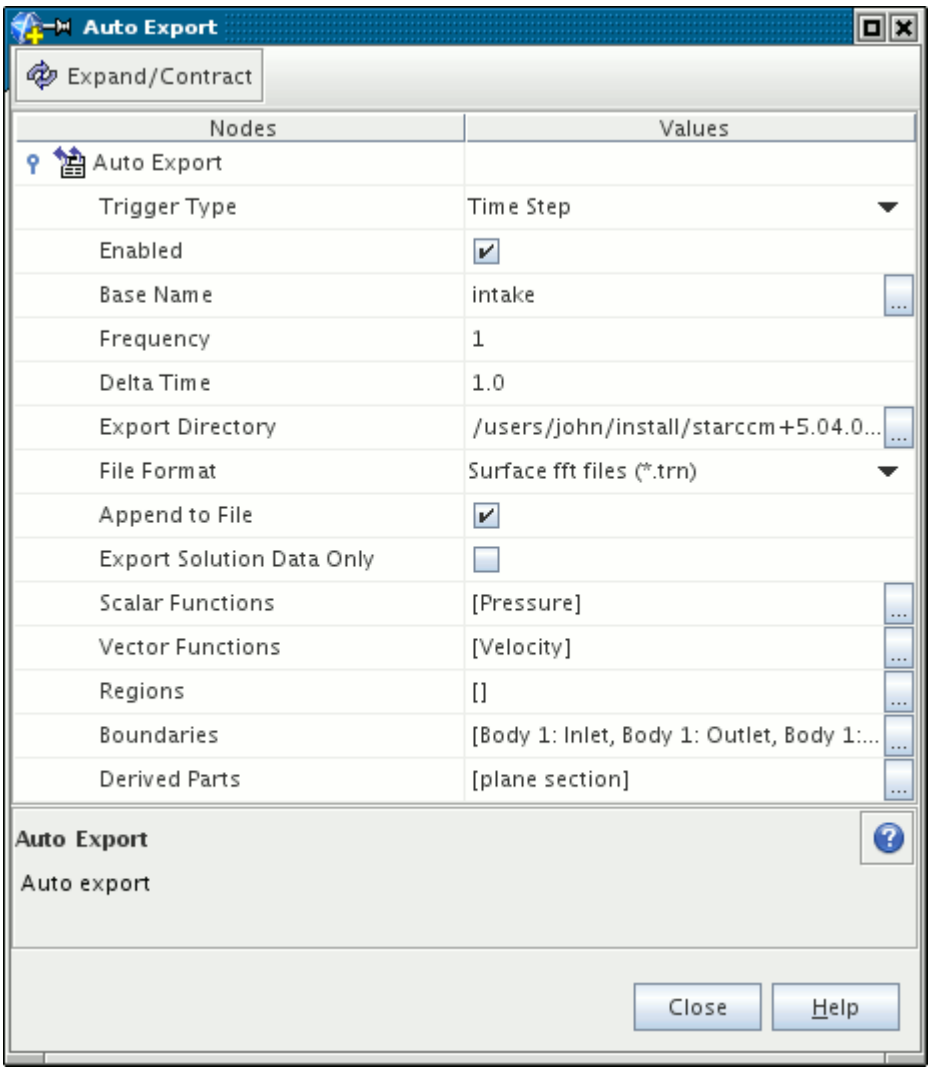
Contents:

Exporting Surface Spectral Data
Importing Surface Spectral Data

Working with Surface Spectra Field Functions
Selecting Units for FT Field Functions
Using the Display Time
Extracting the F-X Spectra Using a Line Probe

Exporting Surface Spectral Data

To export surface spectral data, from the menu bar select **File** > **Auto Export...** opening the *Auto Export* dialog.



The available property settings are detailed below:

- *Trigger Type* -- The `.trn` file can be exported for any transient case setting, with the following options available:
  - **Iterations** -- Data is exported every n iterations, where n is specified in the *Frequency* property.
  - **Time Steps** -- Data is exported every n time steps, where n is specified in the *Frequency* property.

  - **Delta Time** -- Data is exported as defined by the *Delta Time* property.
- *Enabled* -- Must be activated in this instance, to allow the auto export of data.
- *Base Name* -- The file name of the `.trn` file.
- *Frequency* -- Specifies the number of *Iterations* or *Time Steps* between the exports. Neglected if *Delta Time* is set as the *Trigger Type*.
- *Delta Time* -- Specifies the duration of solution time between exports. Neglected if *Iterations* or *Time Steps* is set as the *Trigger Type*.
- *Export Directory* -- The directory to which the `.trn` file is exported.
- *File Format* -- Must be set to **Surface fft files (*.trn)**.
- *Append to File* -- Must be activated to add data to the file defined by the *Base Name* and *Export Directory* properties.
- *Export Solution Data Only* -- If activated the mesh is exported once only. Therefore, effects such as a moving mesh are not captured.
- *Scalar Functions* -- Define the scalar functions to export.
- *Vector Functions* -- Define the vector functions to export.
- *Regions* -- Neglected when exporting to a `.trn` file.
- *Boundaries* -- Define the boundaries to export.
- *Derived Parts* -- Define the derived parts to export.

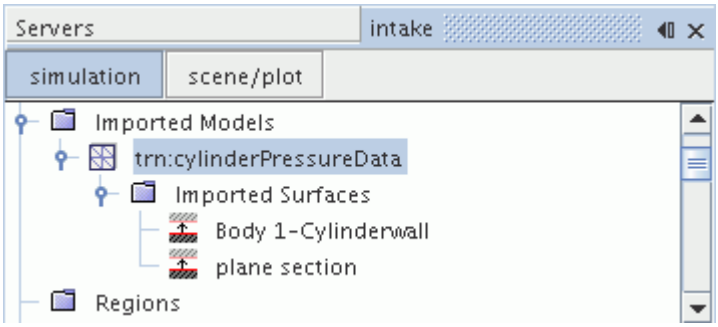
When the case is run, a `.trn` file is generated, which can then be imported.

Importing Surface Spectral Data

To import surface spectral data, from the menu bar select **File** > **Import** > **Import CAE Model...** and navigate to the generated `.trn` file.

You can import surface spectral data for a pre-existing simulation or a new one. When you try this feature for the first time, you can use a new simulation so it is evident what has been imported.

After the import process, a new **Imported Models** manager node appears in the object tree. This node contains imported surfaces corresponding to the previously exported surfaces.

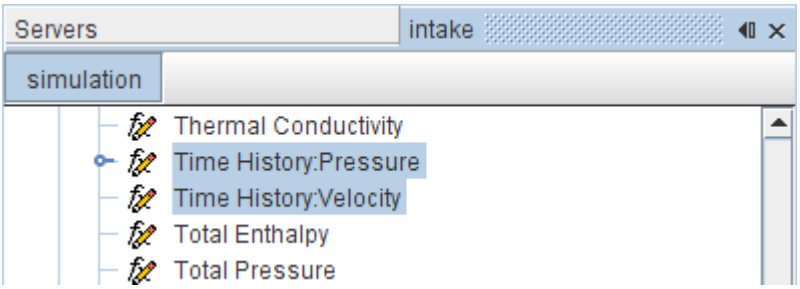


Data set functions can only be performed on surfaces whose discretization does not change with time. For example, if a data set function is calculated on an exported isosurface, values of zero are returned.

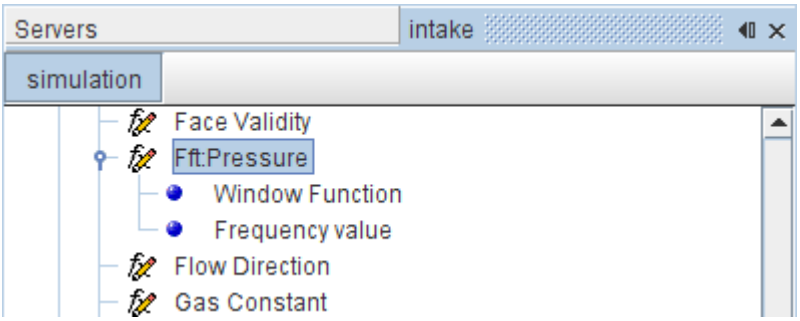
Working with Surface Spectra Field Functions

Importing a `.trn` file registers the following field functions:

- Time History:



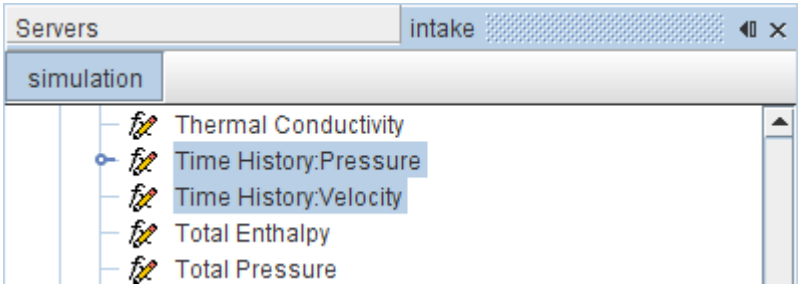
- Fourier Transform:



The time history and field functions can be displayed in a scene, by setting the appropriate field function as a scalar displayer.

Using the Time History Field Functions

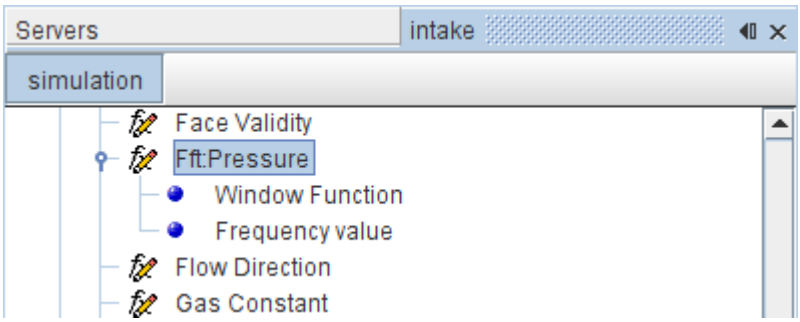
The `.trn` import registers a number of time history field functions relating to the exported field data.



These field functions have mostly read-only properties, except that the **Time History:Pressure** field function lets you select a filter type, which activates a FIR Filter sub-node.

Using the Fourier Transform Field Function

The import registers an FFT field function, relating to the exported scalar data. This field function has a number of properties that allow you to control the Fourier transform, in addition to the base properties of a typical field function.



Fourier Transform Field Function Properties

Fft Input Start Time	The start time (in seconds) for the time sampling for this object. If you have multiple FTs, you can vary the settings of this property to sample multiple time intervals from the simulation run.	
Fft Input Cut-off time	The end time (in seconds) for the time sampling for this object. If you have multiple FTs, you can vary the settings of this property to sample multiple time intervals from the simulation run.	
Amplitude Function	Specifies the amplitude function that is generated. These functions are all standard for this type of analysis. **Note:** The function domain is limited to positive frequencies.	
	Power Spectral Density	Frequency-domain description of the signal power.
	Sound Pressure Level	Sound pressure in decibels. Adds the Reference Sound Pressure node.
	Root Mean Square	The square root of the `Power Spectral Density`.
	A-Weighted Sound Pressure Level	Weights the `Sound Pressure Level` with the A-scale function, for loudness less than 55 phons (at 1000 Hz, 1 phon = 1 dB). This option is typically for problems that involve the frequency range for human hearing. Adds the Reference Sound Pressure node.

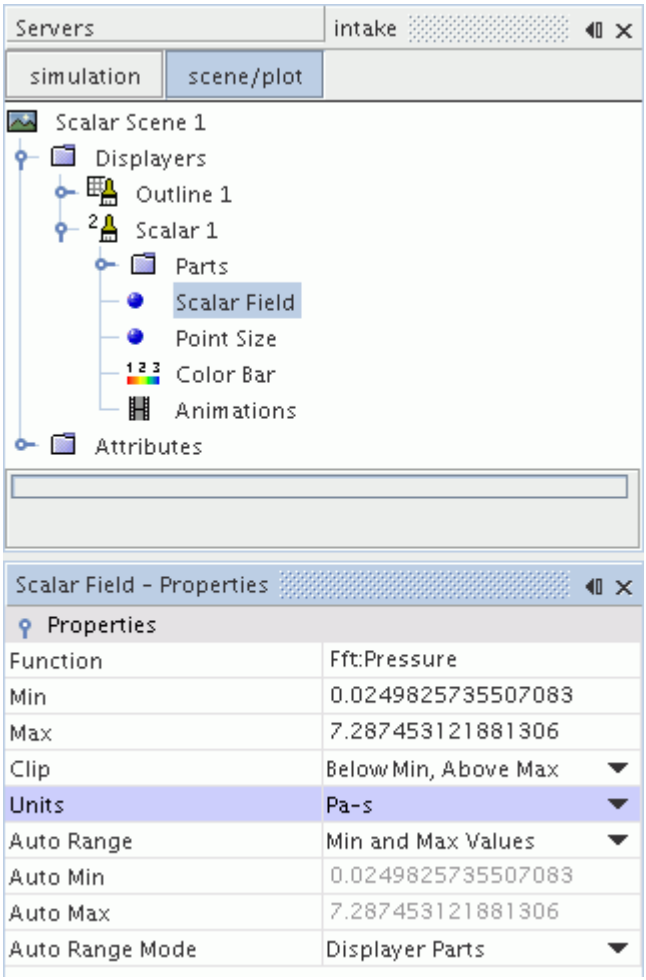
	B-Weighted Sound Pressure Level	Weights the `Sound Pressure Level` with the B-scale function, for loudness of 55-85 phons. Adds the Reference Sound Pressure node.
	C-Weighted Sound Pressure Level	Weights the `Sound Pressure Level` with the C-scale function, for loudness greater than 85 phons. Adds the Reference Sound Pressure node.
Frequency Range Type	Specifies the frequency function that is used. The sub-node that is contained within the FT field function is used to define the input for each frequency range type.	
	Frequency Value	Frequency in Hz. You specify a single value.
	Frequency Range	Frequency in Hz over a range of values. You specify the lower and upper limits.
	Octave Bands	Discrete frequency bands corresponding to different octaves within hearing range. Each successive band represents twice the range of the previous one. The current frequency band defines the hearing range.
	3rd Octave Bands	Discrete frequency bands corresponding to one-third of each octave within hearing range. The current frequency band defines the hearing range.
Analysis Blocks	Specifies the number of analysis blocks used.	
Overlap Factor	Sets the overlap factor between blocks, if any, in a range from `0` to `0.9`. This overlap allows for smooth transitions between blocks.	
Field update	When **On**, updates the FT field function when the display time or when a field function property is changed. The **Off** option can reduce the amount of time that is taken to update the results.	
Window Function	Specifies the window function that you require for this FT. These options are all standard for this type of analysis. • **None** - This default setting leaves the FT without a window function until you select one of the other options in this list. • **Hamming** • **Hann** (Hanning)	

	• **Bartlett** • **Blackman**	
Nyquist Frequency	A read-only value in Hz.	
Filter Type	Specifies the type of filter that is applied.	
	None	Does not use a filter.
	Band pass filter	Keeps frequencies between low and high cut-off levels. Adds an Ideal Filter sub-node in which you can set these levels.
	Band stop filter	Discards frequencies between low and high cut-off levels. Adds an Ideal Filter sub-node in which you can set these levels.
Dimensions	A read-only value of the dimensions that STAR-CCM+ calculates for the FT field function.	

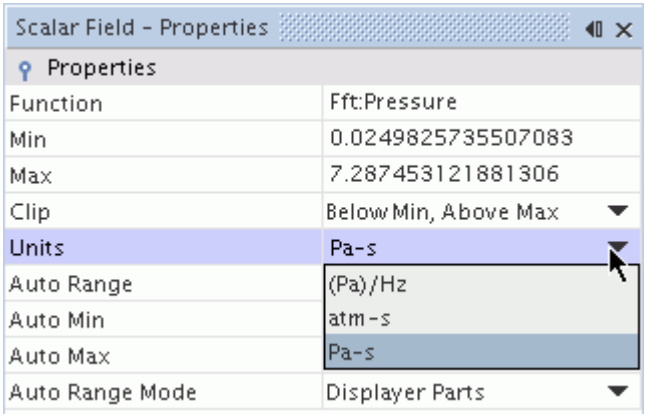
Selecting Units for FT Field Functions

STAR-CCM+ calculates the dimensions for the FT field functions and provides a list of suggested units that are based on these dimensions.

The units for the FT field function are located in *Properties* window of the **Displayers** > **Scalar 1** > **Scalar Field** node of the scene.



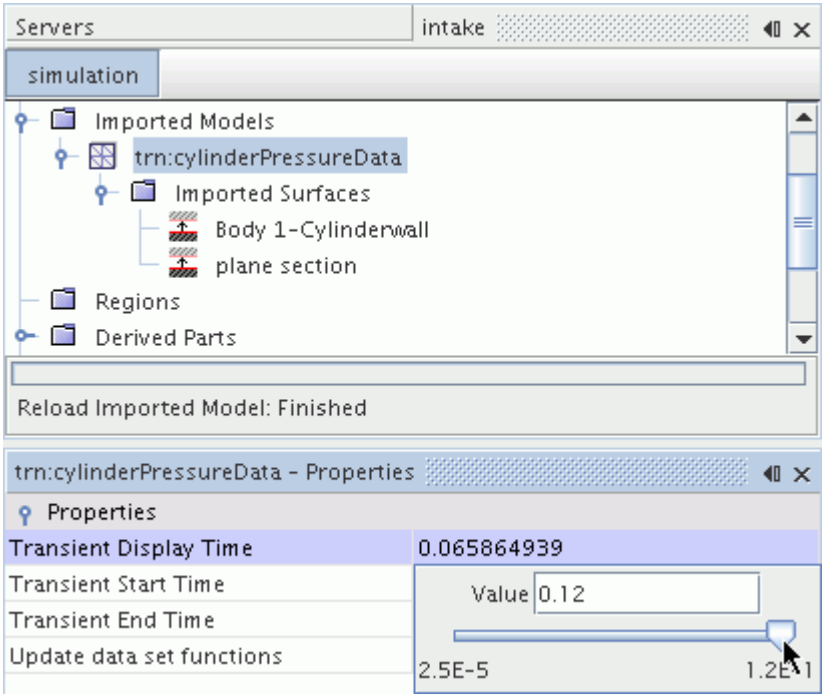
In the example that is shown above, STAR-CCM+ calculated the units **using the** dimensions **for the Fft: Pressure** field function (**Pressure** and **Time)**. The suggested units are shown below.



Select the most relevant units from this list. In this particular example, **(Pa)/Hz** is the most suitable choice.

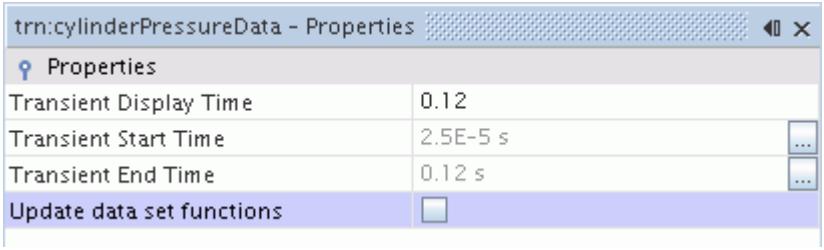
Using the Display Time

Display time is controlled from a slider on the transient model, which updates the scene automatically.



This slider does not affect FT field functions which are controlled entirely from their own properties.

When the *Update data set functions* property of the **Imported Models** > **trn:fileName** node is activated, the data set functions that contain data from a `.trn` file recalculate whenever the *Transient Display Time* is changed. As this calculation can take some time, you can deactivate the update option until all the times are set to the values you require.



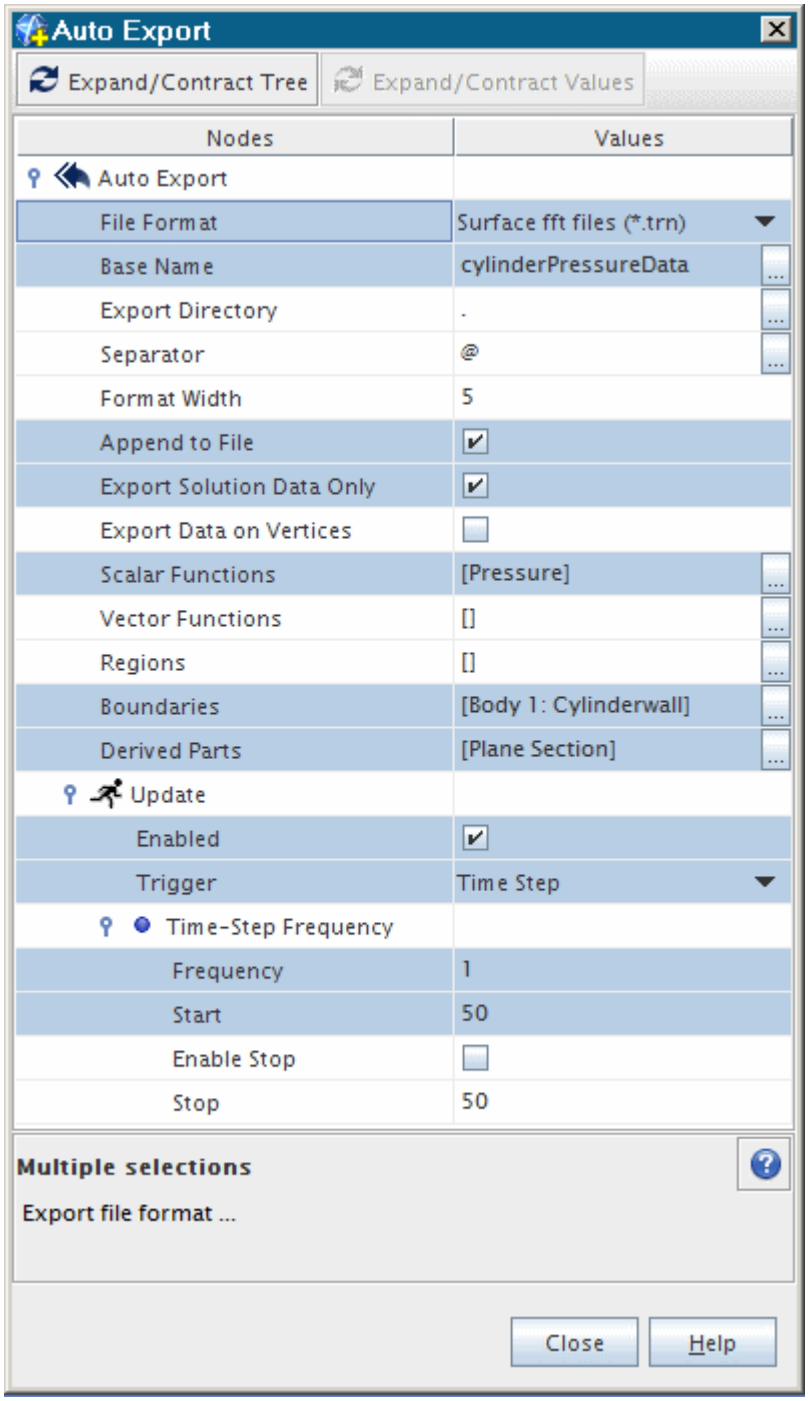
Extracting the F-X Spectra Using a Line Probe

Spectra contours give useful clues on the dominant frequencies in a problem. To visualize spectra contours, write data out on a plane section, then, in post-processing, position a line probe on the plane in different locations.

(The `.trn` file does not accept line probes.)

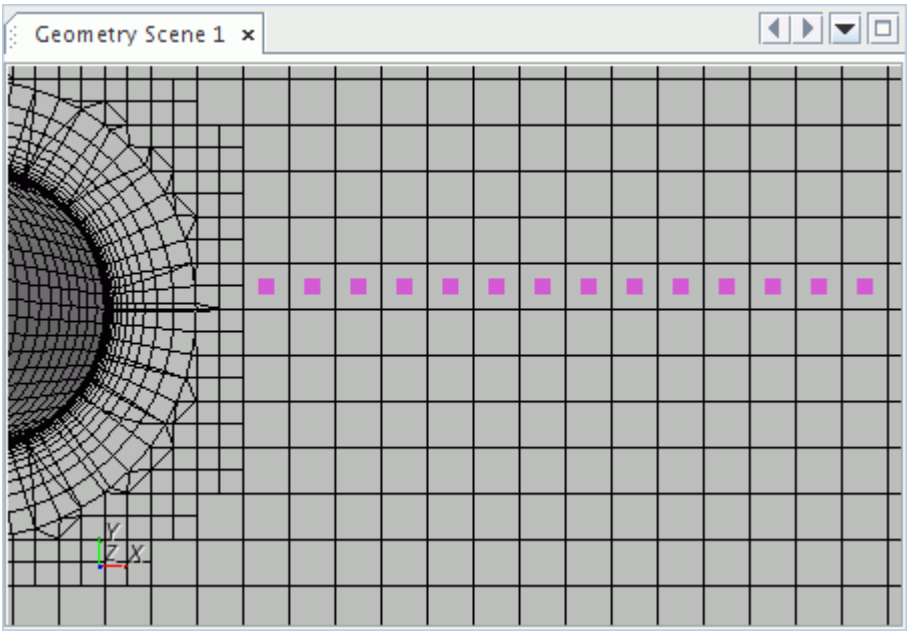
Use the following steps:

1. Create a plane section in derived parts, then set the Auto-Export to export the file in **Surface fft files (.trn)** format and use the previously created plane section as the input part. See below for an example of Auto Export settings.



2. Run the simulation, then save the `.trn.sim` file. This action also saves the `.trn` file.
3. Import the file as a CAE model, using the original `.sim` file or a new one.

4. Create a line probe on the plane section from the CAE model. Keep the line points equally spaced and close to cell centers to reduce interpolation error, as shown below.



5. Map the `.trn` file into the Line Time History object:
   a) Right-click the **Tools** > **Data Set Functions** node and select **New** > **Line Time History (h[l])**.

      The **h(l)** node appears under **Data Set Functions**. This node stores the history on the line. Rename if desired.
   b) Right-click the **h(l)** > **Imported** node and select **New derived data from imported surface and derived part**. The node **Multi-Point Imported** opens below.
   c) Select **Multi-Point Imported** and set the following properties:

Property	Setting
Field Function 1	**Time History:Pressure**
Data Surfaces 1	the plane section
Part	the line probe
Update Active	Activated

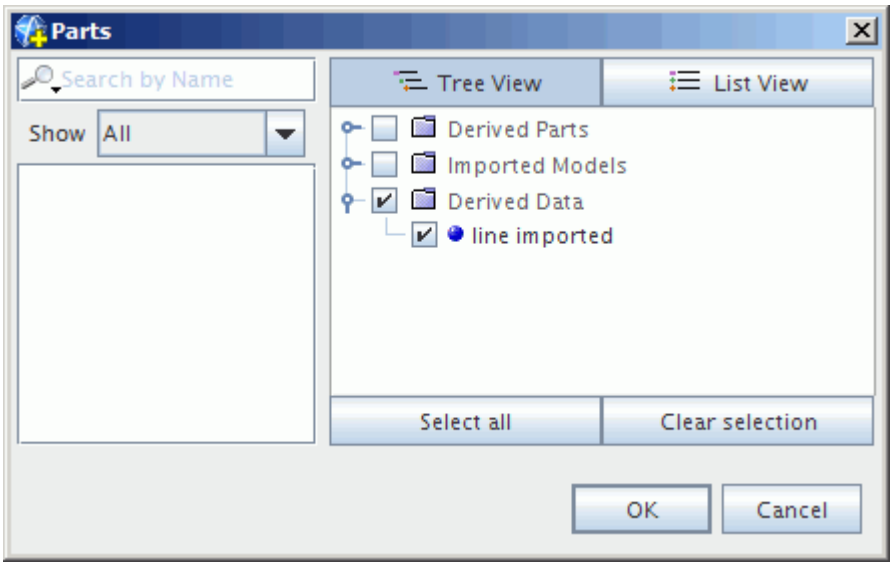
6. Right-click the **Tools** > **Data Set Functions** node and select **New** > **Line Time Fourier Transform (G[h[l]])**.

   Time-steps must be equally spaced for this step to work properly.

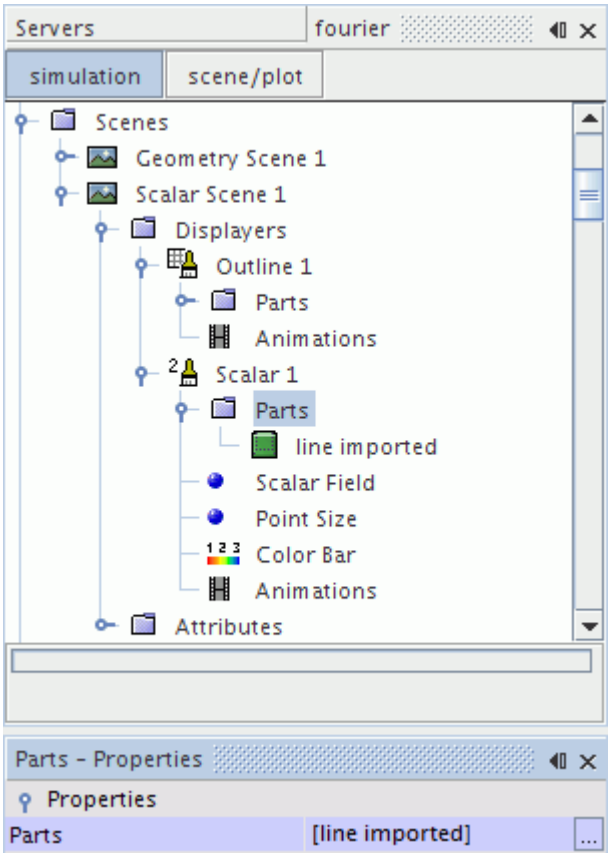
   The **G[h[l]]** node appears under **Data Set Functions**. This node stores the spectrum from history **h(l)** on the line. Rename if desired.
7. Right-click the **G[h[l]]** > **Imported** node and select **New derived data from derived data**.
8. Select **Multi-Point Imported** and set the following properties:

Property	Setting
Input Time History	**G[h[l]]** under its current name
Update Active	Activated

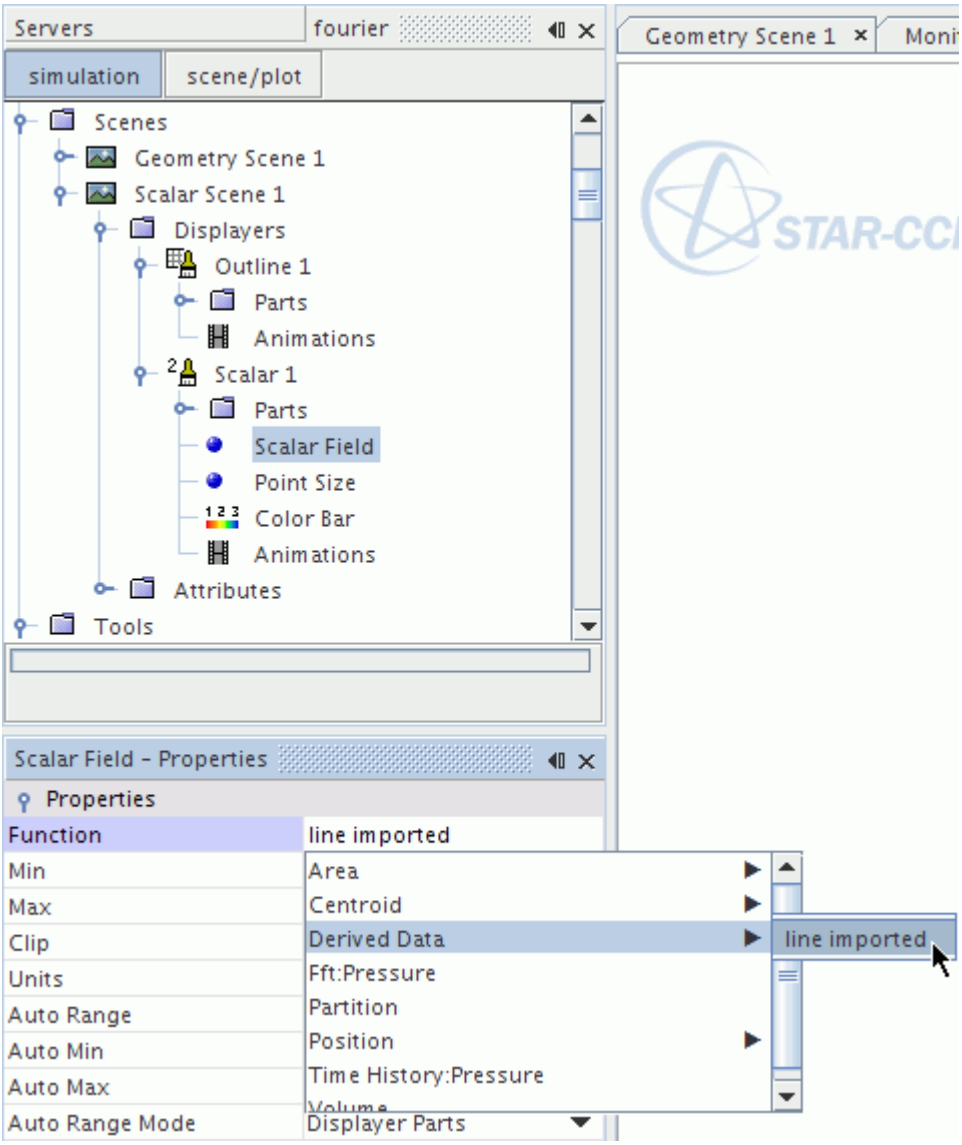
9. Select **Output control** under **G[h[l]]** and set *3D profile type* to the **G[h[l]]** node under its current name. This stores an F-X spectra surface, plotting frequency vs. distance.
10. In post-processing, create a scalar scene.
11. Select the derived data using the *Parts* property of the scalar displayer.
    a) Select the **Scalar 1** > **Parts** node.
    b) In the *Properties* window, click the ... property customizer button for the *Parts* property.
    The *Parts* dialog appears.
    c) Select the derived data set.



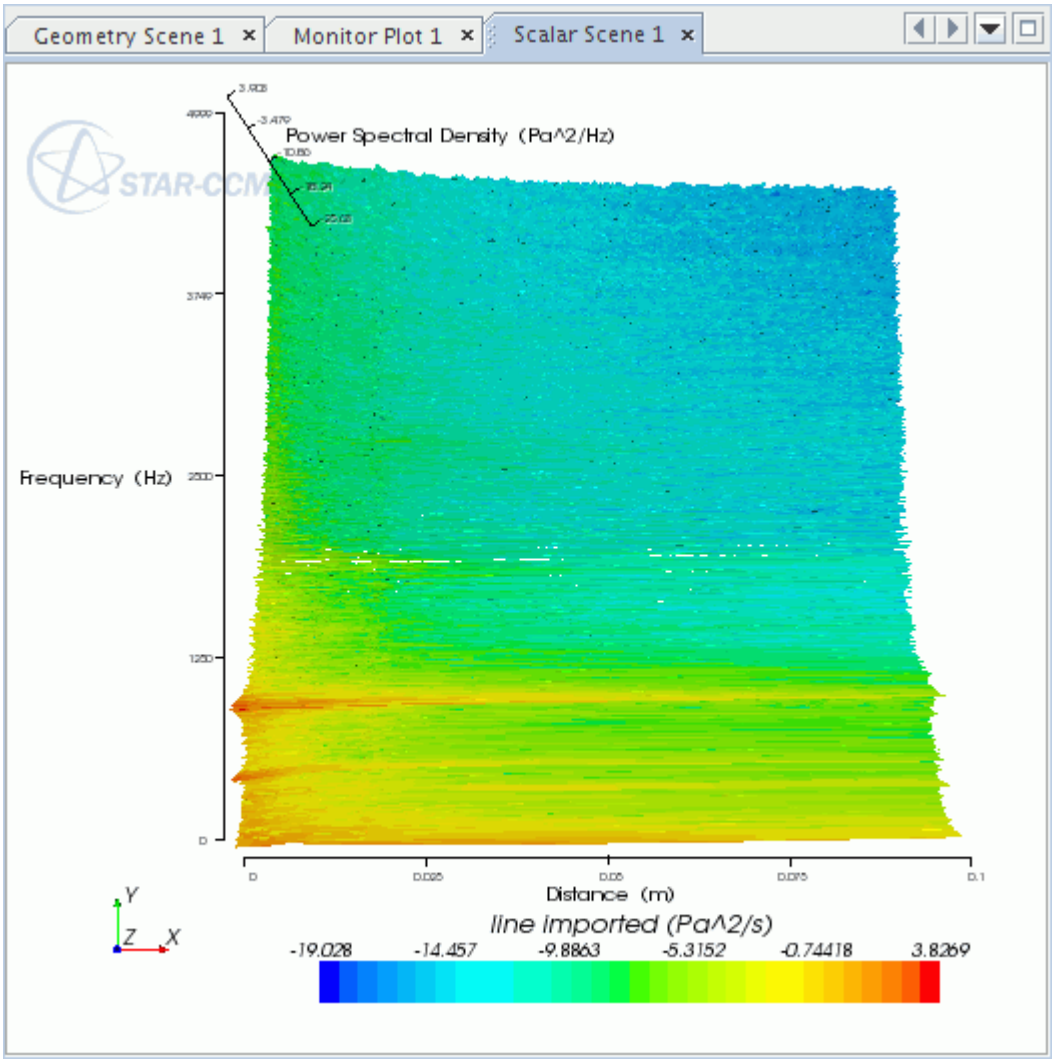
    d) Click **OK**.



12. Select the **Scalar Field** node and set its *Function* to the derived data.



The spectra contour is displayed in the scalar scene window. In the following example, the dominant frequency mode is around 900 Hz.



Repeat these steps with line probes in different positions, or after changing the position of the existing probe, to get a more comprehensive picture of the frequency distribution.

See also the Signal Post-Processing tutorial.

Pop-Up Menu of the Data Set Functions Manager

New	Creates a data set function of the type you select in the submenu:	
	Point Time History (h(p))	Creates a point time history.
	Point Time Fourier Transform (G(p))	Creates a point time Fourier transform.
	Point-to-point Time Fourier Transform (G(p,q))	Creates a point-to-point time Fourier transform.
	Point-to-point Time Correlation (R(p,q))	Creates a point-to-point time correlation.
	Point Inverse Time Fourier Transform (h'(G(p)))	Creates a point inverse time Fourier transform.
	Line Time Fourier Transform (G(l))	Creates a line time Fourier transform.
	Line Spatial Fourier Transform (F(G(l)))	Creates a line spatial Fourier transform.
	Surface Time History (h(s))	Creates a surface time history.
	Surface Time Fourier Transform (G(h(s)))	Creates a surface time Fourier transform.

Data Set Functions Formulation

This section includes the following formulation:

Contents:

Basic Relations
Positive and Negative Frequency Ranges
Time-Space Fourier Transforms
Frequency-Space Auto-Spectrum and Cross-Spectrum
Time Cross-Correlations
Frequency-Space Parseval's Relation
Frequency-Wavenumber Auto-Spectrum
Spatial Cross-Correlations
Frequency-Wavenumber Parseval's Relation
Wave Group Velocity
Spectra Averaging
Spectra Filtering
Power Spectra Density
Sound Pressure Level

Basic Relations

The time Fourier transform is defined as

$$FT_t[h(x;t)] = \int_{-\infty}^{\infty} h(x:t) e^{-i\omega t} dt = g(x;\omega) \quad \forall \omega \in [-\infty, \infty] \tag{1321}$$

where $h(x;t)$ is a real function which depends on the position x and time $t \cdot \omega$ is angular frequency.

The integration limits are over the whole time domain $t \in [\infty, \ -\infty]$. However, the integration is over a given period $t \in [-T/2, \ T/2]$ in practice. The function $h(x;t)$ is periodic in time (or is altered to be periodic).

More commonly, a "finite" Fourier transform for a specific time block T is used:

$$FT_t[h(x;t)] = \int_{-T/2}^{T/2} h(x:t) e^{-i\omega t} dt = g(x;\omega,T) \quad \forall \omega \in \left[-\frac{m\pi}{T}, \frac{m\pi}{T}\right] \tag{1322}$$

where m is the number of sampling points.

If the finite transform is conducted on a kth record of block T, then the transform is:

$$FT_t[h(x;t)] = \int_{-\left(t_k + (T/2)\right)}^{\left(t_k + (T/2)\right)} h(x:t) e^{-i\omega t} dt = g(x;\omega,T) \quad \forall \omega \in \left[-\frac{m\pi}{T}, \frac{m\pi}{T}\right] \tag{1323}$$

The inverse time Fourier transform is

$$FT_t^{-1}[g(x;\omega)] = \frac{1}{2\pi}\int_{-\infty}^{\infty} g(x;\omega)e^{-i\omega t}\,d\omega = h(x;t) \quad \forall t \in [-\infty,\infty] \tag{1324}$$

Similarly, although the integration domain is infinite, it is understood that this integration is over a finite frequency space.

The space Fourier transform is defined as

$$FT_s[g(x;\omega)] = \int_x g(x;\omega)e^{-ik\cdot x}\left|\,dx\right| = f(k;\omega) \quad \forall k \in K; \quad \omega \in [-\infty,\infty] \tag{1325}$$

The function is periodic is space (or is altered to be periodic in space).

Using a compact notation

$$g(x;\omega)e^{-ik\cdot x}|dx| \equiv g(x(\xi);\omega)e^{-ik\cdot s(\xi)}\left|\frac{\partial x}{\partial \xi_1}\times\frac{\partial x}{\partial \xi_2}\right|\partial\xi_1\partial\xi_2 \tag{1326}$$

The space Fourier transform is actually in the parametric space instead of the three-dimensional space of the coordinates of a point on a surface. This parametric representation maps to a two-dimensional space which is both planar and regularly sampled.

The inverse space Fourier transform is

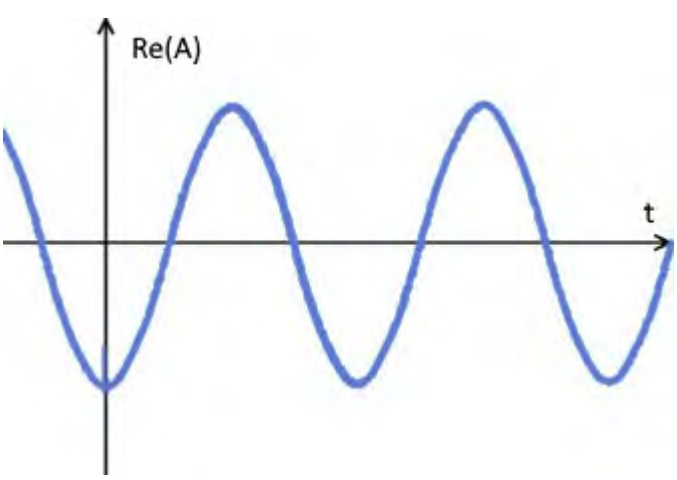
$$FT_s^{-1}[f(x;\omega)] = \frac{1}{(2\pi)^n}\int_K f(x;\omega)e^{-ik\cdot x}|dk| = g(x;\omega) \quad \forall x \in S;\ \omega \in [-\infty,\infty] \tag{1327}$$

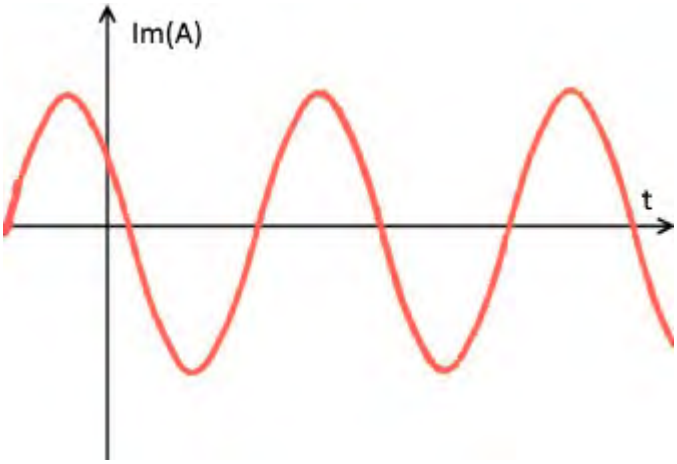
Here, n is the dimensionality of the wavenumber space K. For transforms on a surface $k = \{k_1, k_2\} \in K$, $n = 2$. For transforms on a curve, $n = 1$.

Positive and Negative Frequency Ranges

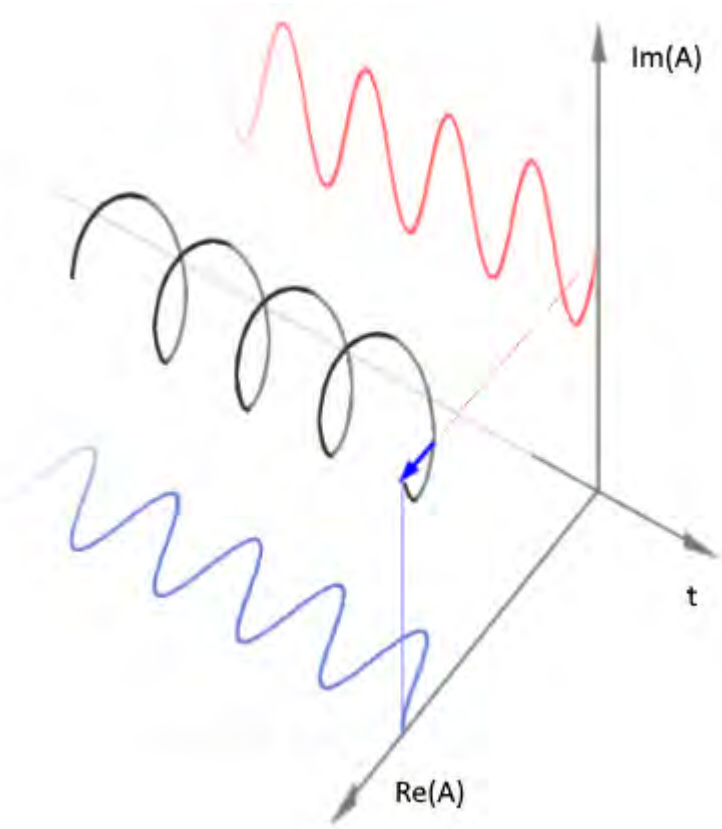
Enter a short description of your concept here (optional).

The real and imaginary parts of a frequency composing an FFT input each describe a sinusoidal curve of amplitude vs. time.





Combined, the two parts describe a helix of amplitude in the complex plane vs time. The original sinusoids are now projections along the real or imaginary axis of the helix.



The helix can be either clockwise or counter-clockwise. Projections of clockwise helices are considered positive frequencies. Projections of counter-clockwise helices are considered negative frequencies.

In a full spectrum, the combined negative frequencies always compose an exact mirror image of the combined positive frequencies, so the negative half of a spectrum adds no information. Therefore, amplitude functions use only the positive frequencies as input.

Negative frequencies contain half the energy in the spectrum, so for energy calculations, both sets of frequencies are considered.

Time-Space Fourier Transforms

The time-space Fourier transform is defined as

$$FT_{ts}[h(x;t)] = \int_{s}\left(\int_{-\infty}^{\infty} h(x,t)e^{-i\left(\omega t + \dot{k}\cdot s\right)}\,dt\right)|dx = f(k;\omega) \quad \forall k \in K; \quad \omega \in [-\infty, \infty] \tag{1328}$$

The inverse time-space Fourier transform is

$$FT_{ts}^{-1}[f(k;\omega)] = \frac{1}{(2\pi)^3}\int_{K}\left(\int_{-\infty}^{\infty} f(k;\omega)e^{-i\left(\omega t + \dot{k}\cdot s\right)}\,d\omega\right)|dx| = h(x;t) \quad \forall x \in S;\ t \in [-\infty, \infty] \tag{1329}$$

The following relationships hold between the transforms:

$$FT_{ts}[h(x;t)] = f\left(k;\omega\right) = FT_{s}[g(x;\omega)] = FT_{s}\left[FT_{t}[h(x;t)]\right] \tag{1330}$$

$$FT_{ts}^{-1}[f(k;\omega)] = h\left(x;t\right) = FT_{t}^{-1}[g(x;\omega)] = FT_{t}^{-1}\left[FT_{s}^{-1}[f(k;\omega)]\right] \tag{1331}$$

Frequency-Space Auto-Spectrum and Cross-Spectrum

The frequency-space (fs) auto-spectrum is defined as

$$G(x;\omega) = g^{*}(x;\omega)g(x;\omega) \tag{1332}$$

where g^{*} is the complex conjugate of g.

The frequency-space (fs) cross-spectrum is

$$G\left(x_1, x_2;\omega\right) = g^{*}\left(x_1;\omega\right)g\left(x_2;\omega\right) \quad \forall x_1 \in S, \forall x_2 \in S \tag{1333}$$

Defining the “finite Fourier transform” for the kth sample time block of length T:

$$g(x;f,T) = \int_{0}^{1} h(x,t)e^{-i\omega t}\,dt \tag{1334}$$

The one-sided cross-spectral density between random processes at any two points in space $x_{1,}\ x_2$ is defined as

$$G\left(x_1, x_2;f\right) = \lim_{T\to\infty}\frac{2}{T}E\left\langle g_k^{*}\left(x_1;f,T\right)g_k\left(x_2;f,T\right)\right\rangle \tag{1335}$$

The corresponding one-sided auto-spectral density of the random signal at a single point is:

$$G(x,x;f) = \lim_{T \to \infty} \frac{2}{T} E\left\langle g_k^*(x;f,T) g_k(x;f,T) \right\rangle = \lim_{T \to \infty} \frac{2}{T} E\left\langle \left| g_k(x;f,T) \right|^2 \right\rangle \tag{1336}$$

where $E\langle\,\rangle$ represents the expectation operator averaging over all indices k. The expectation is important to obtain the coherence between the two random processes, which are defined below.

Since the cross spectral density is in general complex, it is convenient to convert to magnitude and phase. Magnitude is defined as:

$$\left| G(x_1,x_2;f) \right| = \sqrt{\left[Re\ \ G(x_1,x_2;f)\right]^2 + \left[Im\, G(x_1,x_2;f)\right]^2} \tag{1337}$$

and phase is defined as:

$$\theta(x_1,\ x_2;f) = atan\left(\frac{Im\, G(x_1,x_2;f)]}{Re\, G(x_1,x_2;f)]}\right) \tag{1338}$$

The coherence is the measure of how much of the random signal at position x' exhibits that time-averaged phase correlation with the random signal at position x. Coherence is defined as

$$\gamma^2(x_1,x_2;f) = \frac{\left| G(x_1,x_2;f) \right|^2}{G(x_1,x_1;f) G(x_2,x_2;f)} \tag{1339}$$

Time Cross-Correlations

The time cross correlation is

$$R(x_1,x_2;t) = \int_{-\infty}^{\infty} h(x_1;\tau) h(x_2;t+\tau)\, d\tau \quad \forall x_1 \in S,\ \forall x_2 \in S \tag{1340}$$

This convolution goes over the finite period. Periodic boundary conditions must be applied. The Fourier transform makes the same assumption of periodicity. When a Fourier transform is calculated, all notions of causality are lost. You can no longer refer to driver and response signals, but only signals which are correlated. That is, it is impossible to say which signal drives which.

The relationship between the time cross-correlation and the time cross-spectrum is

$$R(x_1,x_2;t) = FT_t^{-1}\left[G(x_1,x_2;\omega)\right] \tag{1341}$$

Equivalently,

$$G(x_1,x_2;\omega) = FT_t\left[R(x_1,x_2;t)\right] \tag{1342}$$

Frequency-Space Parseval's Relation

Parseval's relation states that the total square of pressure is

$$p^2(x) = \int |h(x;t)|^2\, dt = \frac{1}{2\pi} \int G(x;\omega)\, d\omega \tag{1343}$$

where the square root of p^2 gives the root mean square pressure p_{rms}.

Frequency-Wavenumber Auto-Spectrum

The frequency-wavenumber (fk) auto-spectrum is defined as

$$F(k;\omega) = f^*(k;\omega)f(k;\omega) \quad \forall k \in K; \quad \omega \in [-\infty, \infty] \tag{1344}$$

Spatial Cross-Correlations

The spatial cross-correlation is defined as

$$C(x_1, x_2; \omega) = \int_s g(x_1;\omega)g(x_2;\omega)|dx| \tag{1345}$$

Assuming that the cross-correlation is weakly homogeneous and relabeling,

$$C(x_1, x_2; \omega) \approx G_{xy}(r;\omega) \tag{1346}$$

where $r = x_1 - x_2$

The convolution (or cross-relation) in physical space is equal to the frequency-wavenumber (fk) auto-spectrum in wavenumber space. Therefore,

$$F_{xy}(k;\omega) = FT_s\left[G_{xy}(r;\omega)\right] \tag{1347}$$

Equivalently,

$$G_{xy}(r;\omega) = FT_s^{-1}\left[F_{xy}(k;\omega)\right] \tag{1348}$$

Frequency-Wavenumber Parseval's Relation

Parseval's relation states that the total square of pressure is

$$p^2 = \iint |h(x;t)|^2|dx|dt = \frac{1}{(2\pi)^{n+1}}\iint F_{xx}(k;\omega)|dk|d\omega \tag{1349}$$

where the square root of p^2 gives the root mean square pressure p_{rms}.

Wave Group Velocity

The group velocity v_g of a wave is the velocity of propagation of the envelope (or modulation) of the wave in space.

The relationship between the phase velocity , wavelength and frequency of a wave is

$$v_\theta = f\lambda \tag{1350}$$

k is defined as

$$k = \frac{2\pi}{\lambda} \tag{1351}$$

In STAR-CCM+, the wavenumber k is plotted in the range $[-\pi, \pi]$. Therefore,

$$v_\theta = \frac{\pi f}{k} \tag{1352}$$

The magnitude of the group velocity is equal to π times the slope of a frequency-wavenumber (fk) plot:

$$v_g = \pi \frac{\partial f}{\partial k} \tag{1353}$$

Spectra Averaging

Defining the time series,

$$\{h(\mathbf{x})\} = \{h(\mathbf{x}; t),\ t \in \{T\}\} \tag{1354}$$

$$\{T\} = \{t_0, \ \dots\ , t_{N-1}\} \tag{1355}$$

One method of temporal spectral averaging is considered here: blocking.

Blocking is the process of smoothing a spectra. The initial signal is broken into a number of subsignals of the same length. At the same time, sampling granularity is held constant.

$$\{h\mathbf{x}\} = \{\{h_j^b(\mathbf{x})\},\ j = 1,\ N_b\} \tag{1356}$$

$$\{h_j^b(\mathbf{x})\} = \{h(\mathbf{x}\,; t),\ t \in \{T_j\}\} \tag{1357}$$

$$\{T_j\} = t_{\frac{jN}{N_b}},\ t_{\frac{jN}{N_b}+1},\ \dots\ ,\ t_{\frac{(j+1)N}{N_b}-1} \tag{1358}$$

Additional blocks can be overlapped using an overlap factor, which can have values in the range $0 < \alpha < 0.9$. In this case, the number of blocks is

$$N_{b,\alpha} = N_b + (N_b - 1)\left[\frac{\alpha}{1-\alpha}\right] \tag{1359}$$

and they are defined as

$$\{h_{j,\alpha}^b(\mathbf{x})\} = \{h(\mathbf{x}; t),\ t \in \{T_j\}\} \tag{1360}$$

$$\{T_j\} = t_{\frac{j+(1-\alpha)N}{N_b}},\ t_{\frac{j+(1-\alpha)N}{N_b}+1},\ \dots\ ,\ t_{\frac{(j+1)N}{N_b}-1} \tag{1361}$$

Then, treating each subsignal as an independent signal, define the blocking periods

$$\{T_b\} = \left\{ t_0,\ t_1,\ \ldots\ ,\ t_{\frac{N}{N_b}-1} \right\} \tag{1362}$$

over which the mean time history due to blocking can be calculated.

$$\left\{ \overline{h}^{\,b}(\mathbf{x}) \right\} = \frac{1}{N_{b,\alpha}} \sum_{j=N_b} \left\{ h^b_{j,\alpha}(\mathbf{x}) \right\} \tag{1363}$$

Defining a temporal mean spectrum due to the mean of the Fourier transforms of a set of subsignals,

$$\overline{g}_t(\mathbf{x};\omega) = \frac{1}{N} \sum_{j=1,\ N} g_j(\mathbf{x},\omega) \tag{1364}$$

From the Fourier transform addition theorem, you can see that:

$$FT\left[\overline{h}(\mathbf{x};t)\right] = FT\left[\frac{1}{M} \sum_{j=1,\ M} h_j(\mathbf{x};t)\right] = \frac{1}{M} \sum_{j=1,\ M} FT\left[h_j(\mathbf{x};t)\right] \tag{1365}$$

$$\overline{g}_t(\mathbf{x};\omega) = FT\left[\overline{h}(\mathbf{x};t)\right] \tag{1366}$$

Blocking reduces the length of time series. As a result, low frequencies are less well characterized.

It is also possible to average spectra spatially. The spatial mean spectra of the points $\left\{\boldsymbol{P}_i,\ i = 1, N_p\right\}$ can be expressed as

$$\overline{g}_s(\omega) = \frac{1}{N_p} \sum_{i=1,\ N_p} g\left(\boldsymbol{P}_i;\ \omega\right) \tag{1367}$$

The temporal-spatial mean is defined as

$$\overline{g}_{ts}(\omega) = \frac{1}{N_p} \sum_{i=1,\ N_p} \overline{g}_t\left(\boldsymbol{P}_i;\ \omega\right) \tag{1368}$$

Spectra Filtering

It is possible to filter time histories using an ideal filter in the frequency domain.

The band pass filter is defined as follows

$$T^{bp}_{\omega_1,\omega_2}(\omega) = \begin{Bmatrix} 1,\ \omega \in \left[\omega_1,\ \omega_2\right] \\ 0,\ \omega \notin \left[\omega_1,\ \omega_2\right] \end{Bmatrix} \tag{1369}$$

with the band stop filter defined as the inverse of Eqn. (1369).

$$\omega_1 = 2\pi f_1 \tag{1370}$$

where f_1 is the low cut-off frequency.

$$\omega_2 = 2\pi f_2 \tag{1371}$$

where f_2 is the high cut-off frequency.

The filtering operation is then a product in the frequency domain. A frequency filtered time history $h'(x;t)$ can then be defined as follows

$$h'(x;t) = FT_t^{-1}\left[T^{bp}_{\omega_1,\omega_2}(\omega)g(x;\omega)\right] \tag{1372}$$

Power Spectra Density

The Power Spectra Density describes the total signal power over the frequency domain.

The Power Spectra Density (PSD) over a frequency band is the sum of the discrete PSD quantities, which are divided by the frequency ranges in the band.

$$PSD = \sum_i \frac{PSD_i}{\omega_i^{high} - \omega_i^{low}} \tag{1373}$$

where:

- PSD_i is the power spectra density of the ith frequency band.
- ω_i^{high} is the highest frequency of the ith band.
- ω_i^{low} is the lowest frequency of the ith band.

See What Are Bands?

Sound Pressure Level

The sound pressure is the local change in pressure from the reference ambient pressure that a sound wave causes.

The Sound Pressure Level (SPL) over frequency (in Hz) is a logarithmic measure of the effective sound pressure relative to a reference value, and is measured in decibels (dB):

$$SPL = 10\ \log_{10}\left(\frac{p_{rms}^2}{p_{ref}^2}\right) \tag{1374}$$

where:

- p_{rms} is the root mean square sound pressure.
- p_{ref} is the reference sound pressure.

The Sound Pressure Level over a series of frequency bands is:

$$SPL = 10\ \log_{10}\left(\sum_i 10^{\left(SPL_i/10\right)}\right) = 10\ \log_{10}\left(\frac{\sum_i \left(p_{rms}^2\right)_i}{p_{ref}^2}\right) \tag{1375}$$

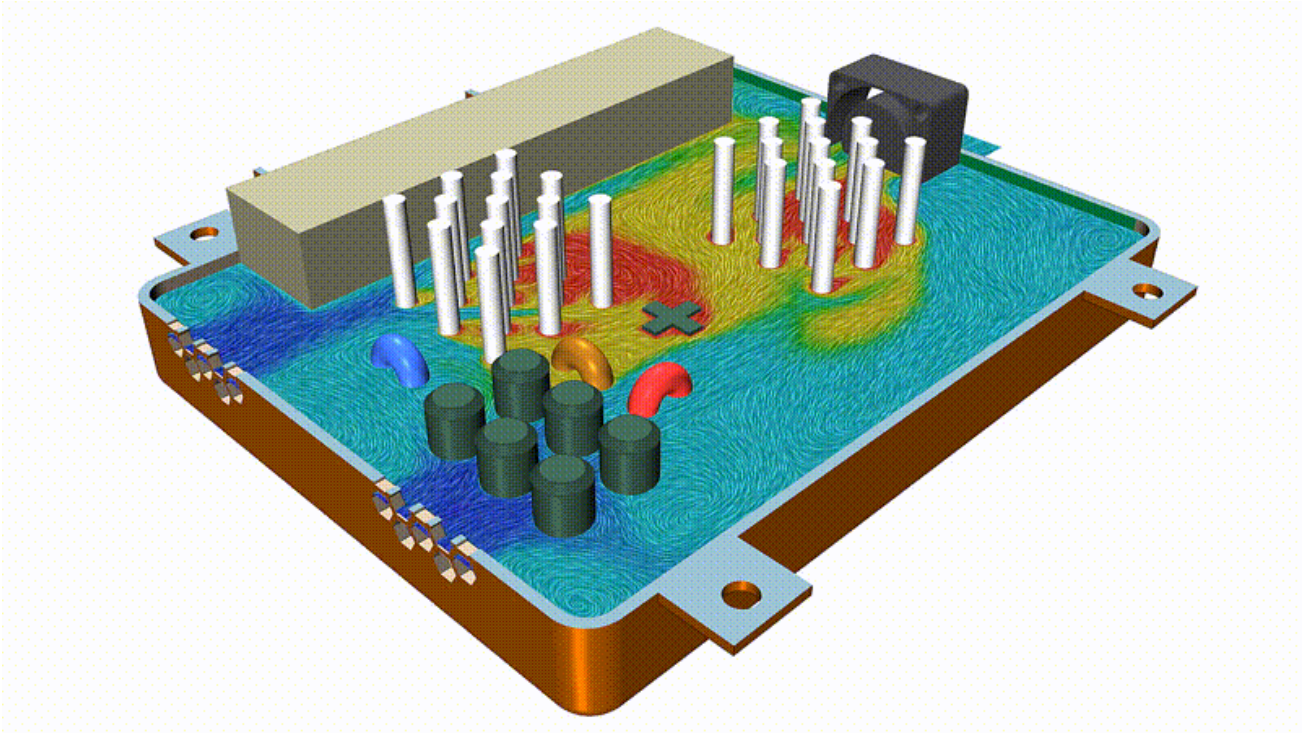
where:

- SPL_i is the sound pressure level for the ith frequency band.
- $\left(p_{rms}\right)_i$ is the individual p_{rms} for the ith frequency band.

See What Are Bands?

Design Manager

Design Manager provides an automated approach within STAR-CCM+ to run design exploration studies. Design exploration covers both performance assessment and optimization.



The development and improvement of most engineering products comes from a systematic analysis of the parameters that affect the product performance. The performance of a product can be associated with its primary operating characteristics and involves measurable quantities—such as weight, size, or drag. The process of identifying key parameters and adjusting their values in order to improve the performance of the product usually requires a large number of design iterations.

Each iteration of such an exploration makes the product somehow better or worse. However, each design gives valuable information about how to navigate through the infinite number of potential designs towards some final goal. Examples of questions that motivate such an exploration of the design space are:

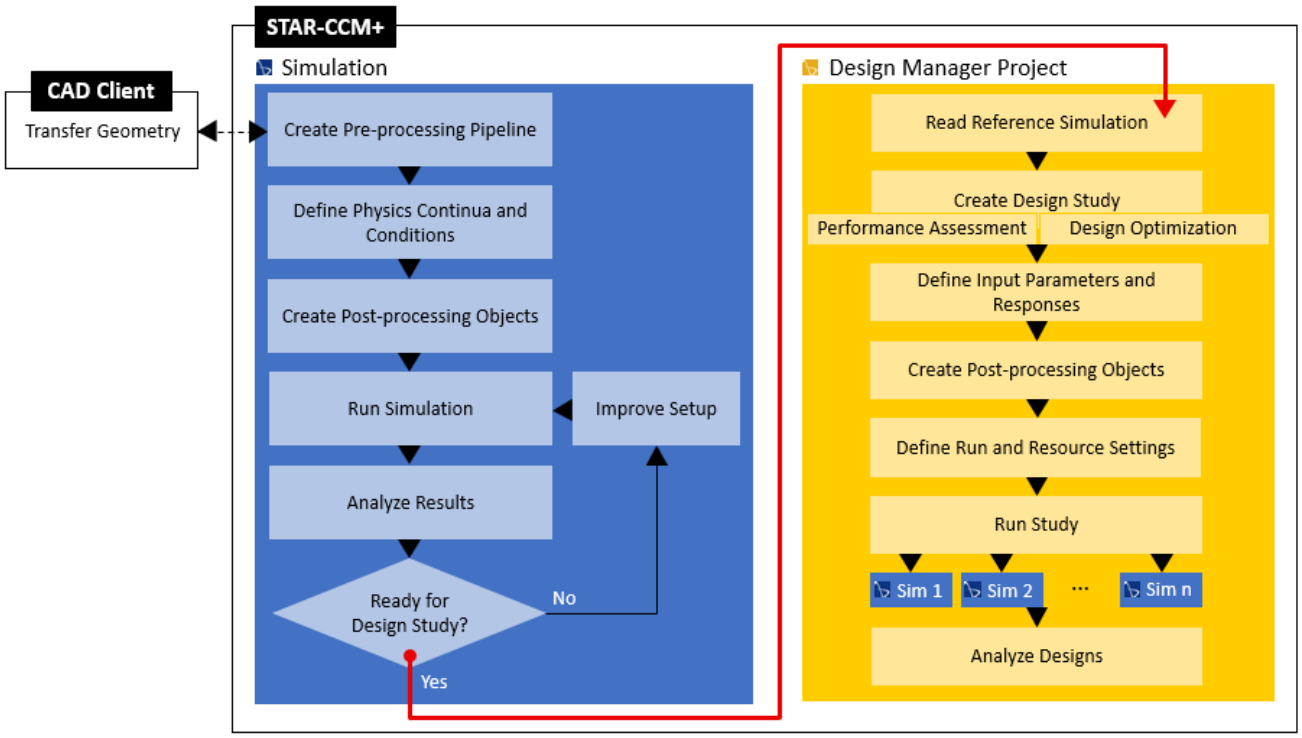
- How can I reduce the drag of a car by changing the position of the side mirrors?
- What effect does the bend radius have on the pressure drop across a tube?
- How can I reduce the mass of the heat sinks in an electronic device and also improve the cooling rate?

There are two main categories in the design exploration process: performance assessment and design optimization.

A performance assessment study runs a product through a pre-defined set of conditions in order to understand the influence of key parameters on the performance of the product. In a design optimization study, an optimization algorithm determines the input conditions automatically in order to improve the product with respect to a specific goal. As these two categories provide different kind of information, a constructive and efficient design exploration process is usually a combination of both.

In STAR-CCM+, Design Manager allows you to automate the design exploration process for your product. For performance assessment as well as design optimization studies, Design Manager controls the execution of the respective simulations in STAR-CCM+.

The following diagram gives you an overview of the design exploration process in STAR-CCM+:



The starting point of each design exploration in STAR-CCM+ is a simulation file that is configured to provide the necessary inputs and outputs. In this **Reference Simulation**, you define the following:

- Design and global parameters that Design Manager is allowed to modify.
- Reports that extract simulation results and define engineering objectives and constraints.
- Scenes and plots for design analysis.

In Design Manager, you read these simulation objects and make them available for the design study. Currently, the following types of design study are available:

- For performance assessment: Manual Analysis, Parameter Sweep
- For design optimization: Single-Objective Optimization, Multi-Objective Optimization (Pareto), DOE (Design Of Experiments), Stochastic Analysis

Independent of the study type, Design Manager supports multiple design studies in one Design Manager project. Design Manager also supports the seeding of design studies with designs of previous studies. This way, Design Manager allows you to fully automate a sequential run of design studies, where the results of one study determine the design space of the following study.

Design Manager provides various settings to control the run of a design study and the compute resources to use. A design study can run its designs sequentially or concurrently, in serial or in parallel, on your local machine or on a high performance cluster. For cluster runs, Design Manager offers a pre-allocation mode that reserves the required resources for the study in the queue. Design Manager then controls the submission of the designs within the reserved resources.

When running a particular design, Design Manager launches STAR-CCM+ in batch mode and modifies the relevant simulation settings. During the run, Design Manager checks the status of the running design and, when a design completes, records the outputs.

After the run, various post-processing capabilities allow you to analyze the results of a design study:

- Output tables that list the designs and their details in tabular form. Smart query capabilities allow you to filter the designs.
- Snapshots for visual side-by-side comparisons of different designs.
- Plots that visualize results spanning multiple designs, such as design history plots.

Contents:

Study Inputs
Study Outputs
Study Types
Study Runs
Study Analysis
Design Manager Licensing
Design Manager Workflow
Design Manager Reference

Study Inputs

In Design Manager, study inputs are parameters that you define in the reference simulation and that Design Manager can modify during design studies.

The types of STAR-CCM+ simulation objects that can be selected as Design Manager input parameters are:

- 3D-CAD Design Parameters
- CAD Client Design Parameters
- Global Parameters

3D-CAD Design Parameters

Design parameters are used in 3D-CAD to specify characteristic dimensions of a model that can be modified outside 3D-CAD. For more information, see Design Parameters.

The design parameters that you expose in your reference simulation are available as inputs for Design Manager. Changing a design parameter pushes the modified geometry through the mesh pipeline so that the geometry is remeshed. Make sure that the specified range for the variables allows the geometry to remesh successfully. If a change to a design parameter causes the geometry to become invalid, the Design Manager analysis fails to run.

You can create variables from the following 3D-CAD design parameters:

- Scalar quantities—values such as dimension distances, radii, extrusion lengths, taper angles.
- Translation components—Design Manager attempts to translate a coordinate system with the same name as the current design parameter. This feature is useful if you want to rotate a component after the translation. It is important to create the translation variables first, before any rotation variables.

CAD Client Design Parameters

Design parameters that originate from CAD Clients are also supported as input parameters. With a CAD client, you can transfer your CAD geometry and design data—such as design parameters—directly from within your CAD package environment to STAR-CCM+. For the list of supported CAD Packages, see CAD Packages Support.

Global Parameters

A global parameter is a scalar or vector physical quantity (for example, 120.0 K or [0.0, 1.0, 0.0] m) that you define once in your STAR-CCM+ simulation and can be used in multiple objects of the simulation. Changing a global parameter modifies the value of the quantity that refers to the parameter. For more information, see Global Parameters.

Using global parameters, you can instruct Design Manager to control various scalar and vector parameters, for example:

- Mesh values—such as the base size
- Region physics values—such as volumetric heat sources
- Boundary physics values—such as inlet velocities, turbulence scalars, or thermal scalars
- Motion parameters—such as rotation rates or translation velocities

Study Outputs

For each design simulation in a design study, Design Manager allows you to record and write specific outputs.

The outputs of a design study refer to simulation objects that are defined in the reference simulation. In Design Manager, you can define the following output objects:

- Responses
- Scenes and Plots

Additionally, after each design simulation completes, Design Manager gives each associated design a Performance Rating.

Responses

Responses provide summaries of solution data—such as pressure drop or drag coefficient—and are based on Reports that you define in the reference simulation. For each design in a design study, Design Manager records the values of the specified reports at the end of the design simulation. Each design study in Design Manager requires at least one response.

Responses serve three main purposes in a design study:

Information Only

For each design simulation, responses record simulation results. The recorded response values allow you to compare designs with each other.

Objectives

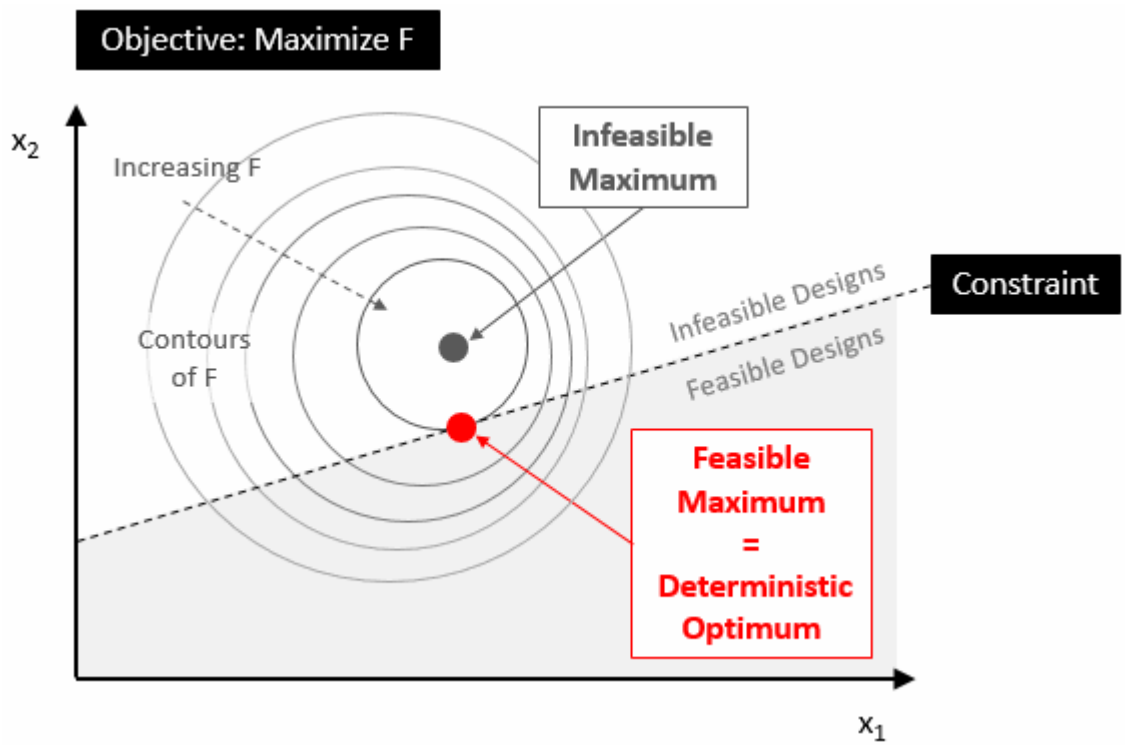
Objectives represent the engineering goals of the analysis. For example, a report that records the pressure drop can be used by Design Manager to explore different designs aimed to maximize or minimize its value.

In Optimization studies, objectives drive the optimization algorithm towards better designs. In Sweep studies, objectives enable you to rate the performance of the designs.

Constraints

Constraints limit the range of acceptable designs by setting limits on the design space. For example, you can set an upper limit for a report that records the mass of your engineering product. Designs that satisfy all constraints are called feasible. If any constraint is not satisfied, the design is called infeasible.

The following diagram displays how constraints limit the design space and affect the optimum design:



Scenes and Plots

Scenes and plots are based on Scenes and Plots from the reference simulation. At the end of each design simulation, Design Manager writes the respective scenes and plots to files.

The following file formats are supported:

- `*.sce` (STAR-View+ Scene file)
- `*.png` (Portable Network Graphics file)

PNG files are two-dimensional representations of the scenes and plots, whereas STAR-View+ Scene files allow you to interact with the exported scenes and plots as you would with the respective scenes or plots in STAR-CCM+.

Performance Rating

In Design Manager, the performance of a design is based on the value returned for the objectives together with the degree to which a design satisfies its constraints. A high-performance design is one that satisfies all constraints and has a good rating on its objectives.

All designs that satisfy the constraints ignore the margin by which they meet those constraints. Once the constraints are satisfied, only the objectives contribute numerically to the performance evaluation. The performance value of each design is calculated using the following equation:

$$\sum_{i=1}^{N_{obj}} \frac{LinWt_i \cdot Sign_i \cdot Obj_i}{Norm_i} - \sum_{j=1}^{N_{con}} \frac{QuadWt_j \cdot ConViol_j^2}{Norm_j^2} \tag{1376}$$

where:

- N_{obj} is the number of objectives in the design study.
- $LinWt_i$ is the linear weight that you specify for the *i*-th objective.
- $Sign_i$ is the sign for the *i*-th objective. The value is -1 for objectives being minimized and +1 for objectives being maximized.
- Obj_i is the response value for the *i*-th objective for that design.
- $Norm_i$ is the normalization value that you specify for the *i*-th objective.
- N_{con} is the number of constraints in the design study.
- $QuadWt_j$ is the quadratic weight for the *j*-th constraint. This value is 10000.0.
- $ConViol_j$ is the amount by which the *j*-th constraint is violated. $ConViol_j$ is 0.0 if the constraint is met.
- $Norm_j$ is the normalization value for the *j*-th constraint, which is the value of the constraint itself. If the constraint value is 0, then the normalization value is 1.

Study Types

Design Manager offers several types of design study by which you can explore the design space of your product. These studies are divided into two categories: performance assessment and optimization.

While performance assessment studies do not require specific licensing, design optimization studies are only available with STAR-Innovate licensing.

Performance Assessment

Manual

In a Manual study, you define a set of designs using tabulated data, where each design is a certain combination of input parameter values. You provide the set of designs before starting the analysis.

A Manual study allows you to automate the process of running a collection of specific designs.

Sweep

In Sweep mode, before starting the analysis, you provide values for the input parameters. For each parameter, you can either provide a constant value or a list of discrete values. You can also define a lower and an upper bound for a parameter in conjunction with a specified increment or resolution.
Design Manager then creates a full factorial design sweep that goes through all combinations of parameter values.

A Sweep allows you to replicate test results, create aerodynamic load databases, create compressor maps, or to run any application where you know the configuration for each job in advance.

Design Optimization (STAR-Innovate)

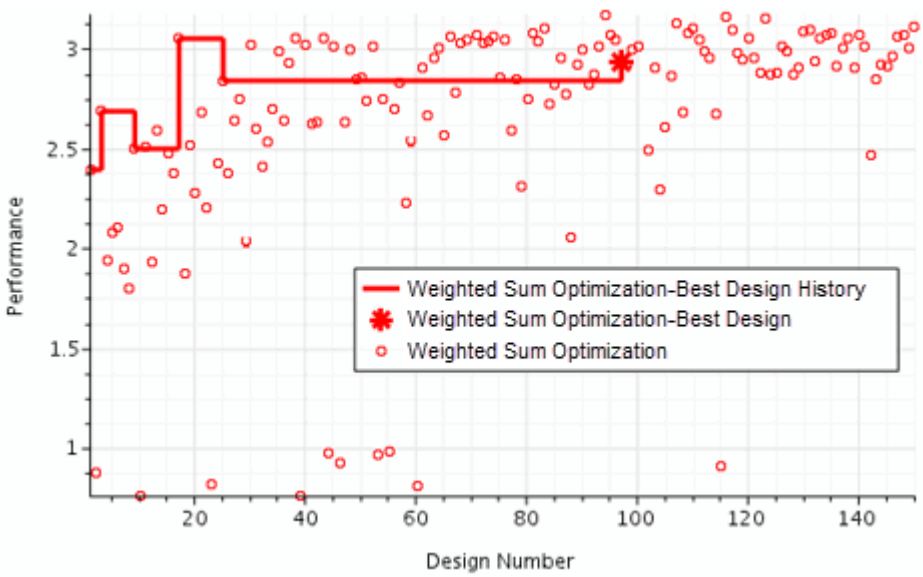
Optimization

In an Optimization study, you define one or more objectives that the analysis mut meet as best it can. Along with the objectives, you define input parameters in the same manner as for a Sweep study. During the analysis, a search algorithm chooses the parameter values for the designs so as to best meet the analysis objectives.

There are two types of Optimization study:

- *Weighted sum of all objectives*

  This type of Optimization study allows you to optimize based on a single objective or based on multiple objectives. If you define multiple objectives, a linear weighting is used that combines all objectives into a single performance function. Running a weighted sum of all objectives analysis with multiple objectives returns a single best design:



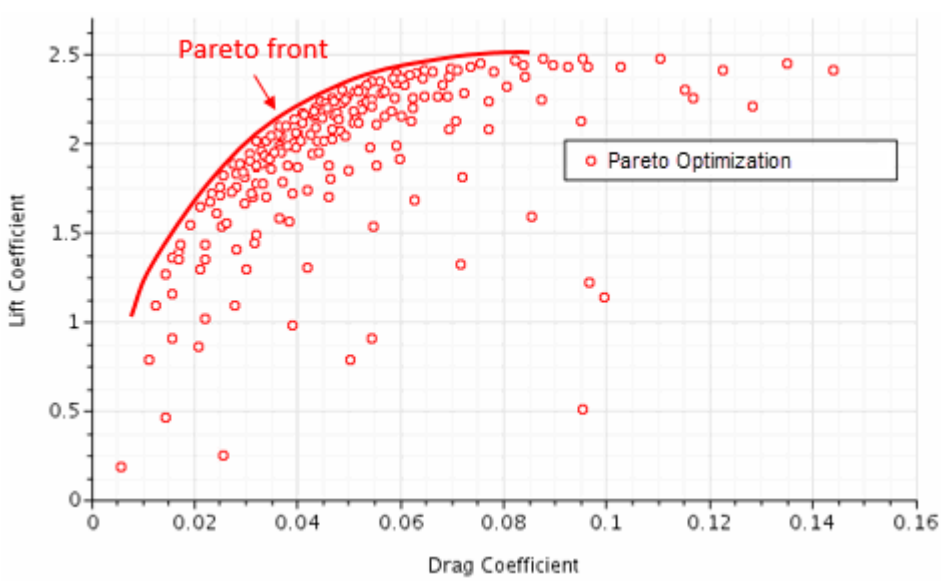
For this type of Optimization study, typical goals include minimizing pressure drop, maximizing outlet uniformity, minimizing component temperatures, or minimizing stress.

Weighted sum of all objectives optimization uses the SHERPA search algorithm.

- *Multiple objective tradeoff study (Pareto front)*

  In this type of Optimization study, you optimize based on competing objectives.

  Pareto optimization is suited to cases where there are two objectives that are competitive in nature. In such cases, there is no single optimum design. Instead, the optimization returns a curve along which all designs are optimum in one objective for a given value in the other objective—known as the non-dominated design condition. This curve, known as Pareto front, expresses the optimum trade-off relationship between two competing objectives:



Finding the Pareto front for cases where there are more than two competing objectives requires a large number of design evaluations. In such cases, you are advised to investigate if all objectives are truly competing or if possibly some of them are analogous and can be combined. Alternatively, you can try to pose the problem in terms of two objectives and constraints.

You can optionally choose to define a weighting for the objectives before you run the analysis.

A multiple objective tradeoff study (Pareto front) uses the MO(Multi Objective)-SHERPA search algorithm.

For information on the search algorithm, see The SHERPA Algorithm.

DOE

A DOE (Design Of Experiments) study is used for local exploration around a particular optimized design in order to identify the parameters that have the greatest impact on the performance. You can assess the sensitivities of the input parameters and their interaction with each other.

For a DOE, you supply the input parameter values at two or three levels:

- *2 Level Full Factorial*

  A two-level full factorial DOE study evaluates all factors (input parameters) and their combinations at two levels: low and high. The number of designs in a 2-level full factorial study is 2^n, where *n* is the number of DOE factors.

- *3 Level Full Factorial*

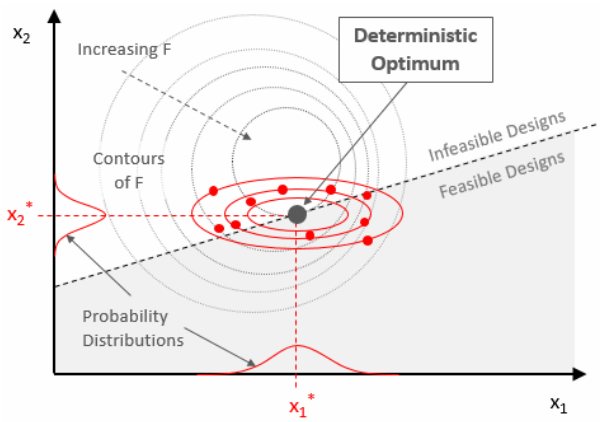
  In a three-level full factorial DOE study, all factors and their combinations are evaluated at three levels: low, mid, and high. The number of designs in a three-level full factorial study is 3^n, where *n* is the number of DOE factors.

Stochastic

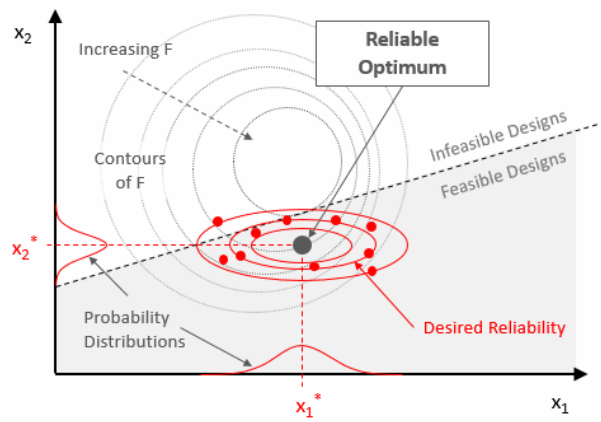
A Stochastic study helps you assess the robustness of a design, that is, how minor variations in the input parameter values affect the outputs.

In a stochastic analysis, you apply a probability distribution to the input parameters. This probability distribution describes the likelihood that an input parameter will take a particular value.

The motivation for a Stochastic study stems from the fact that Optimization studies seek to find designs that are optimal without accounting for small perturbations, such as installation or manufacturing tolerances. When constraints limit the design space, these deterministic optima often lie on constraint boundaries (see Study Outputs). However, small changes in input parameter values can affect the values of the outputs in a non-trivial way, resulting in a large number of infeasible designs. Accounting for variations leads to more reliable designs, which effectively pushes the optimal designs away from the constraint boundaries:



Variations in input parameter values give large number of infeasible designs around deterministic optimum.



Pushing optimum away from constraint boundary reduces number of infeasible designs.

Note: Usually, the process of design optimization combines Optimization, DOE, and Stochastic studies:

1. An **Optimization** study evaluates designs over a broad range of input parameters and parameter values. This study gives you the best design.
2. A **DOE** study evaluates the input parameters in a small range of parameter values about the best design. This approach gives you the parameters that have the greatest impact on the performance of the design.
3. A **Stochastic** study uses the sensitive parameters identified in the DOE study to determine the robustness of the best design.

Contents:

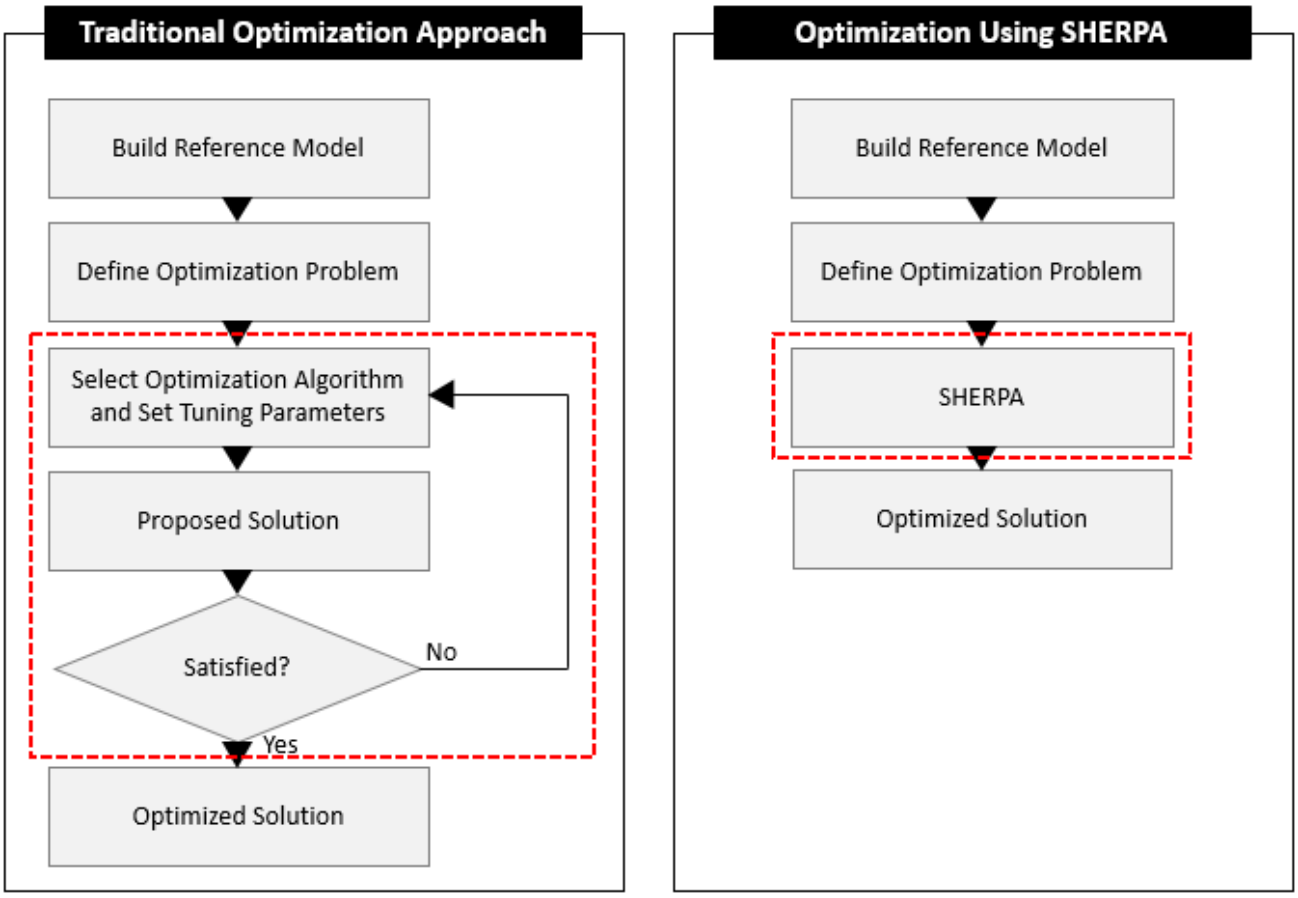
The SHERPA Algorithm

The SHERPA Algorithm

SHERPA is a unique search algorithm—it performs a **S**imultaneous **H**ybrid **E**xploration that is **R**obust, **P**rogressive, and **A**daptive.

During a single search, SHERPA uses multiple search methods simultaneously rather than sequentially. This approach uses the best attributes of each search method. If a particular search method is deemed to be ineffective, SHERPA reduces its participation.

A comparison of a traditional optimization approach and of the SHERPA approach is displayed below.



While running the SHERPA algorithm, a combination of global and local search methods are used. At any given time, the number of different methods that are used can range between two and ten. Unlike traditional optimization algorithms that require you to tune parameters manually, the tuning parameters in each method that SHERPA uses are modified automatically during the search. As SHERPA learns more about the design space, it determines when and to what extent to use each search method.

The general advantages of using SHERPA are:

- You need not spend time understanding the design space before choosing a suitable algorithm. SHERPA learns about the design space.

- You do not require any expertise in optimization algorithms and applications. SHERPA makes all the decisions about which method to use and how to tune them.
- You can define realistic problems that are based on real engineering or business costs and benefits, without being constrained by the limitations of a particular search method. With SHERPA, your analysis can contain much broader goals and can include a larger number of variables compared to traditional approaches.

Study Runs

Design Manager allows you to run design studies on your local workstation or on a high performance cluster. You can run the design simulations sequentially or concurrently, in serial or in parallel.

For each design simulation in a design study, Design Manager performs the following steps:

1. Launches STAR-CCM+ in batch
2. Loads the reference simulation
3. Modifies the relevant input parameters
4. In case of geometry changes, remeshes the geometry
5. Runs the design simulation until the stopping criteria are met
6. Records the outputs

Cluster Submission

As Design Manager is integrated within STAR-CCM+, no additional installation or configuration is required to run Design Manager on a cluster. Design Manager only requires that a Batch Management System is present on the cluster.

Design Manager offers two modes in which to run Design Manager on a cluster:

- General job submission mode
- Pre-allocation mode

Note: Currently, only Linux clusters are supported.

General Job Submission Mode

In general job submission mode, Design Manager submits each design simulation as a single job to the queue. For this mode, the Design Manager server must run on a node that has the ability to submit jobs to the queue—such as the cluster head node or a visualization node. This cluster node and the compute nodes must share the same file system with the same mount path. After starting the design study from the Design Manager client, Design Manager submits the design simulations to the queue, tracks their progress, and extracts their output on completion.

The licenses that are required for a design simulation are checked out as soon as the design simulation starts.

Benefit:

- Simple setup that allows you to submit design simulations to the queue directly from the User Interface.

Drawback:

- The Design Manager server process, although lightweight, must be running on the head node or a visualization node until the entire design study completes.

Pre-Allocation Mode

The pre-allocation mode allows you to submit the entire Design Manager project as a single job to the cluster queue. Within the allocated resources, Design Manager acts as its own queueing system and submits the design simulations one by one. The resources that you allocate for the Design Manager project must be sized according to the resources desired for the design studies in the project. Design Manager keeps the allocated resources until the design studies of the project complete.

The licenses that are required for the compute simulations of a design study are checked out in advance, before any compute simulations begin. Design Manager keeps the licenses until the last design simulation completes.

Benefits:

- The compute resources are allocated for the entire project. This approach allows the design studies in the project to complete without having individual design simulations held up in the queue.
- The required licenses are allocated in advance. No other user can pull pre-allocated licenses away from running design studies. Without licenses, design simulations can not be launched.
- Most efficient usage for the following study types: Manual, Sweep, DOE, and Stochastic

Drawback:

- For Optimization studies, Design Manager must wait for an entire SHERPA cycle to finish before starting the next cycle. For this reason, using pre-allocation for Optimization studies can counteract an efficient usage of resources.

Note: Currently, you cannot use STAR-CCM+ Power Session licenses (`ccmpower` feature license) with pre-allocation mode.

Study Analysis

Design Manager provides post-processing capabilities by which you can analyze the outputs of a design study.

In Design Manager, you can create the following types of post-processing objects:

- Output Tables
- Snapshots
- Project Plots

With these post-processing objects you can analyze the designs of a **Design Set**. A design set is a collection of designs that meet some specific criteria. Design Manager provides you with the following predefined design sets:

- All—all designs in the design study
- Error—designs that encountered an error when they ran
- Successful—all designs apart from the Error designs

Additionally, you can create new design sets that are based on specified query rules. For example, you can filter for designs with a certain response below a specified value.

Output Tables

Output tables display the designs within a design set and their details in tabular form. For each design, an output table lists the following items:

- Design#—design number in the design exploration history
- Name—unique label for the design
- State—progress of the design simulation:

  No icon: This design has not started.

  This design is queued to run.

  This design has been assigned to run.

  This design has started to run.

  This design is running. This state includes updating the CAD data and parameter values, meshing, and solving.

  This design has executed.

  This design is processing outputs.

  This design has completed successfully.

  This design has failed.

In most cases, the states and change so quickly that you do not notice them in the GUI.

- [Response]—recorded response from completed design
- Performance—performance value of the design
- [Parameter]—input parameter applied to the design

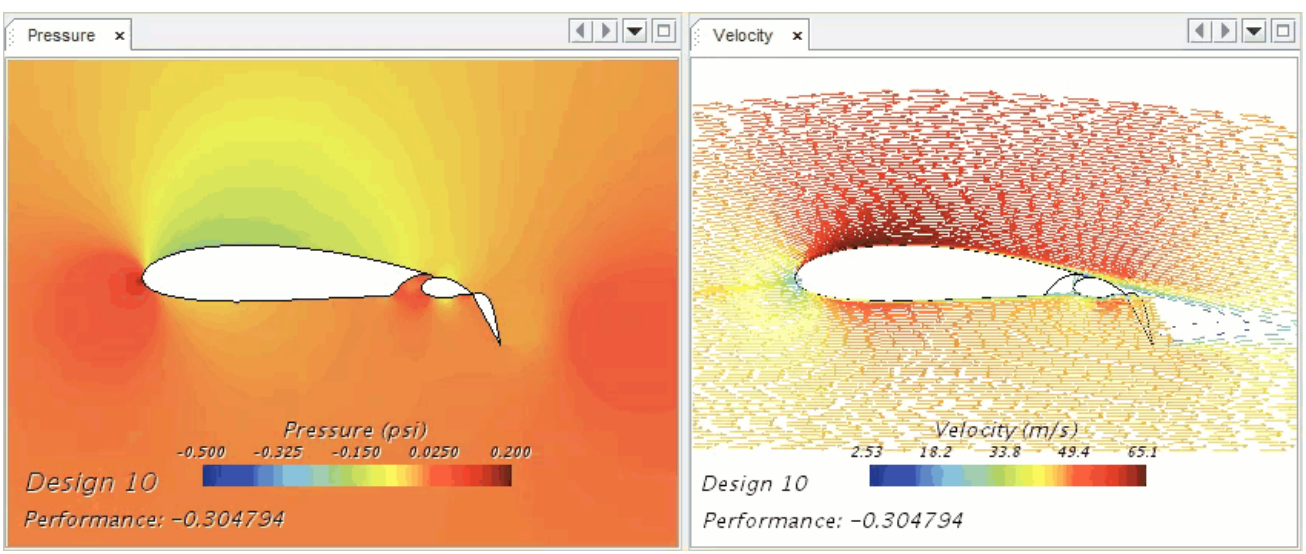
Using output tables, you can track which design simulations are queued to run, which have started to run and which are currently running. You can also see which design simulations are complete and, of these completed designs, which simulations completed successfully and which received an error.

Design#	Name	State	Cd	Cl	Avg Cl over Cd	Cl over Cd	Mass Cons	Performance	LeadingFlapRotateAngle	TrailingFlapRotateAngle	TranslateLeac
1	Design 1	✓	0.07907653462630737	2.394952791716276	26.39045254657961	30.286516765487058	0.12981705367565155 kg/s	2.4746196536362675E-8	0.0 deg	0.0 deg	0.0 n
2	Design 2	✓	0.0525951862715568	0.8734923405068495	14.546952198473805	16.607838139347546	-0.10981904529035091 kg/s	-0.30039532247468087	-10.0 deg	-0.8000000000000007 deg	0.00879999999
3	Design 3	✓	0.07924532868875012	2.6885804040440595	28.120529781658973	33.927304593611176	0.12373624742031097 kg/s	0.1204681322506993	-4.4 deg	-4.4 deg	0.01600000000
4	Design 4	✓	0.09285772053464059	1.9436429474877346	18.286450932943122	20.931409217208365	-0.0563635379076004 kg/s	-0.362718589739328	-2.8 deg	4.4 deg	-0.00879999999
5	Design 5	✓	0.07341238711026744	2.07998058015124	24.285761044218795	28.33282858691206	0.2787340581417084 kg/s	-0.05988629896658104	0.8000000000000007 deg	10.0 deg	0.01280000000
6	Design 6	✓	0.1055702163072622	2.105333498932663	16.967871428002805	19.942494887053073	0.001094236969947815 kg/s	-0.45596746620823236	-0.8000000000000007 deg	0.8000000000000007 deg	0.00560000000
7	Design 7								6.399999999999999 deg	6.399999999999999 deg	-0.00560000000
8	Design 8								4.4 deg	-2.8 deg	-0.02
9	Design 9								-8.0 deg	8.0 deg	-0.016
10	Design 10								-6.4 deg	-8.0 deg	-0.00160000000
11	Design 11								2.8000000000000007 deg	-6.4 deg	-0.012
12	Design 12								8.0 deg	2.8000000000000007 deg	0.02

Snapshots

Snapshots allow you to view the scenes and plots that are exported from the design simulations of a design set.

Each snapshot refers to a specific scene or plot where you can visualize the results of different designs. By default, all snapshots display the same design, which allows you to perform side-by-side comparisons of different scenes or plots.



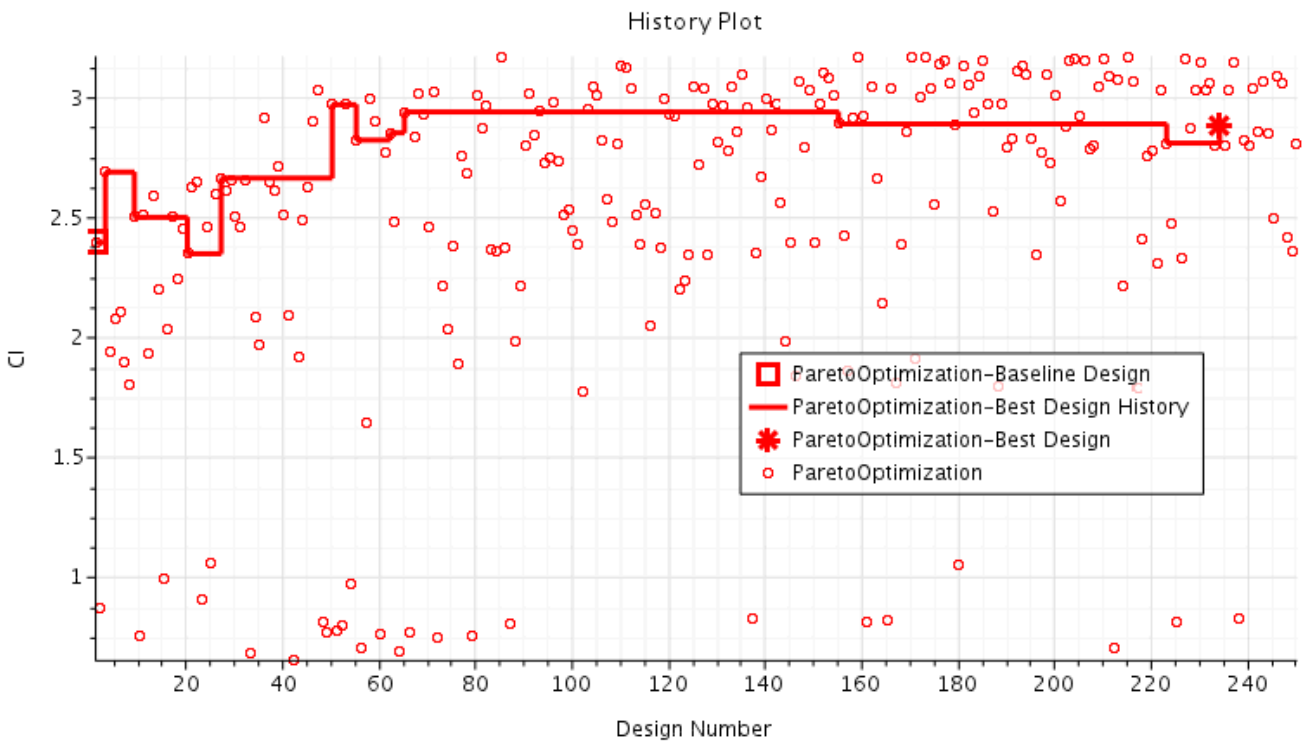
You can create multiple snapshots that display the same scene or plot. When you lock one of the snapshots to a specific design, you can compare results between different designs. You can link snapshots so that they move in tandem as you interact with one of them.

Project Plots

Project plots allow you to visualize results that span all the designs of a design set in two-dimensional graphs. For example, you can create the following types of project plot:

- History plots—show a response or a parameter against the design number for a range of designs.
- 2D relation plots—show a response against a parameter for a range of designs.
- Pareto plots—display the tradeoff between two competing responses.

Typically, each design is represented by a single point. You can apply markers to highlight the baseline design and the best design of a study. The best design history visualizes the path of a design exploration study to the best design. If you defined constraints for your study, you can show the designs that violate each of the constraints.



Note: Output tables, snapshots and project plots are linked to each other. When you select a certain design within one of these post-processing objects, the other objects are synchronized automatically to reflect the selected design. This feature enables you to flick through the designs of a design set.

Design Manager Licensing

The licenses that are required for running a design study in Design Manager depend on the type of study and the specified run settings. Also, different steps within the Design Manager workflow have particular license requirements.

The following licenses are required during the various steps of a design study in Design Manager:

Read Reference Simulation		**Setup Design Studies**		**Run Locally or on Cluster**		**Analyze Designs**
1 STAR-CCM+ product license (`ccmpsuite`) or 1 Power Session license (`ccmppower`) Only used when reading the reference simulation or when updating the link to the reference simulation	➡	No license	➡	Design Manager supports two licensing schemes: • Default • Unlicensed Design Manager Server See Licensing Schemes. For both licensing scheme, the following licenses are supported: • Power Tokens (`DOEtoken`) • Power Session licenses (`ccmppower`) • Power on Demand licenses • STAR-CCM+ product licenses (`ccmpsuite`) You can control the check out order of licenses using command line options. For examples, see Common Licensing Scenarios.	➡	No license
				1 STAR-Innovate license (`innovatesuite`) for the following study types: • Optimization • DOE • Stochastic		

Licensing Schemes

Design Manager supports two different licensing schemes—Default and Unlicensed Design Manager Server. You control the licensing scheme using the following command line options:

Licensing Scheme	Command Line Option
Default	None
Unlicensed Design Manager Server	`-dmnoshare`

Default

By default, when you run a design study, Design Manager—that is, the Design Manager server—acquires one session license and shares it with the STAR-CCM+ design simulations. This session license can be a serial STAR-CCM+ product license (`ccmpsuite`), a Power Session license (`ccmppower`), or a Power-On-Demand license. The Design Manager server keeps the session license until the last design simulation completes.

The STAR-CCM+ design simulations obtain their own licenses:

- 1 core license for each concurrent design simulation
- Parallel licenses for the processors on which to run each design simulation

By default, these licenses are Power Tokens (`DOEtoken` feature licenses), but you can also use other licenses following the rules in the License Options Table.

When you run Design Manager in Pre-allocation mode, the Design Manager server automatically pulls all required licenses in advance and shares them with the STAR-CCM+ design simulations.

Unlicensed Design Manager Server

In this licensing scheme, Design Manager does not pull any licenses. The STAR-CCM+ design simulations start up as standard STAR-CCM+ simulations following the standard stand-alone STAR-CCM+ licensing rules. See STAR-CCM+ Licensing.

This scheme allows you to perform design studies using only a single `ccmppower` license without requiring an additional session license. Use this licensing scheme when you have only `ccmppower` licenses or Power-On-Demand licenses available.

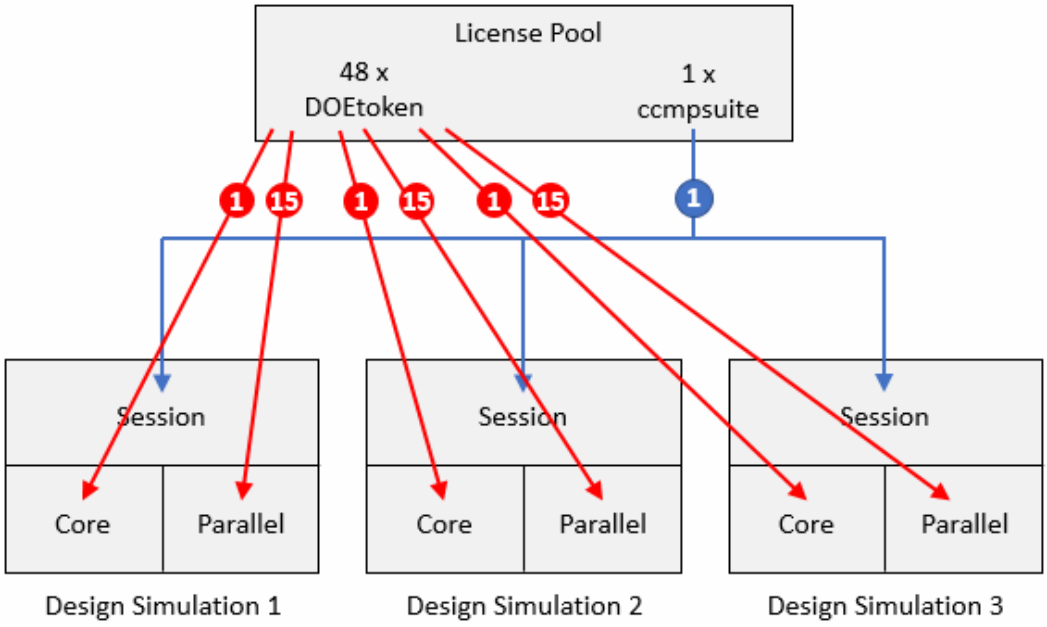
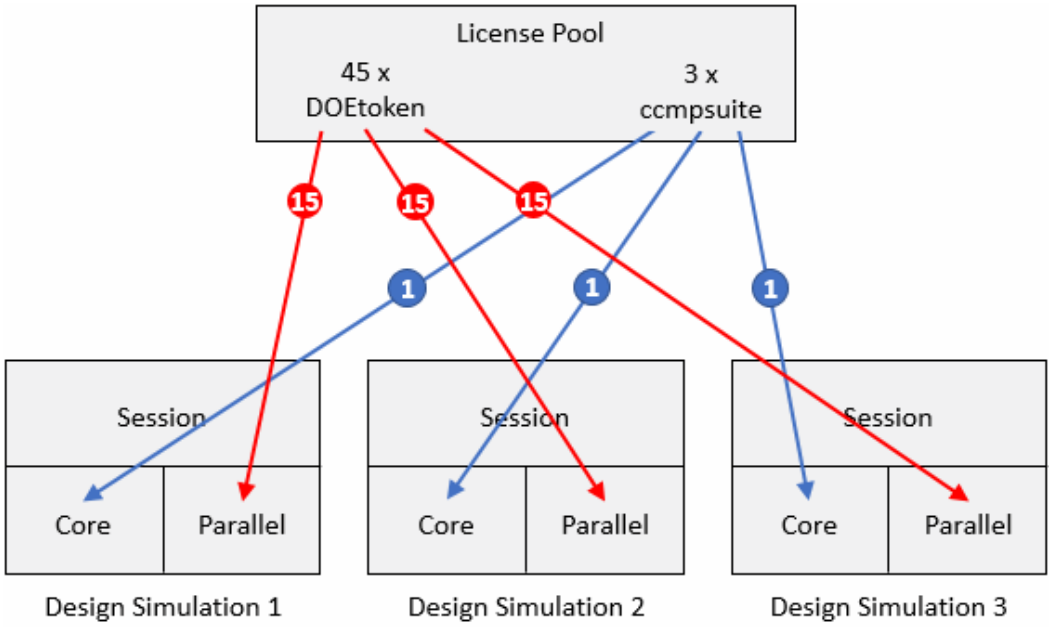
Note: The Unlicensed Design Manager Server scheme is not supported when running Design Manager in Pre-Allocation mode.

Example 1: Suppose you have the following licenses available:

- 3 `ccmpsuite`
- 50 `DOEtoken`

You want to run 3 design simulations concurrently, each on 16 processors. Design Manager pulls the following licenses:

Licensing Scheme	License Usage
Default	

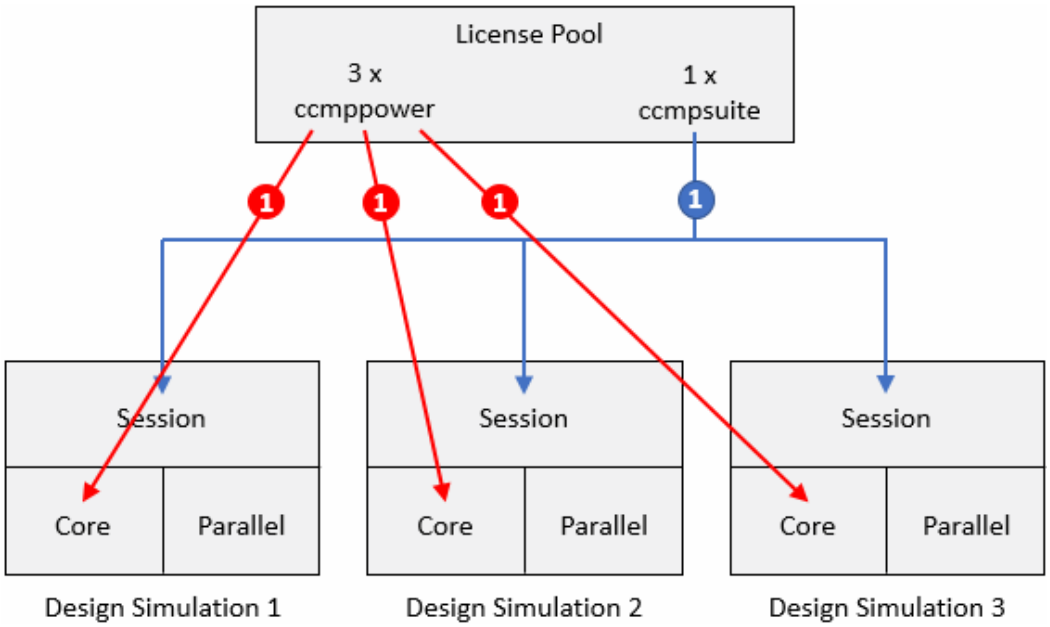
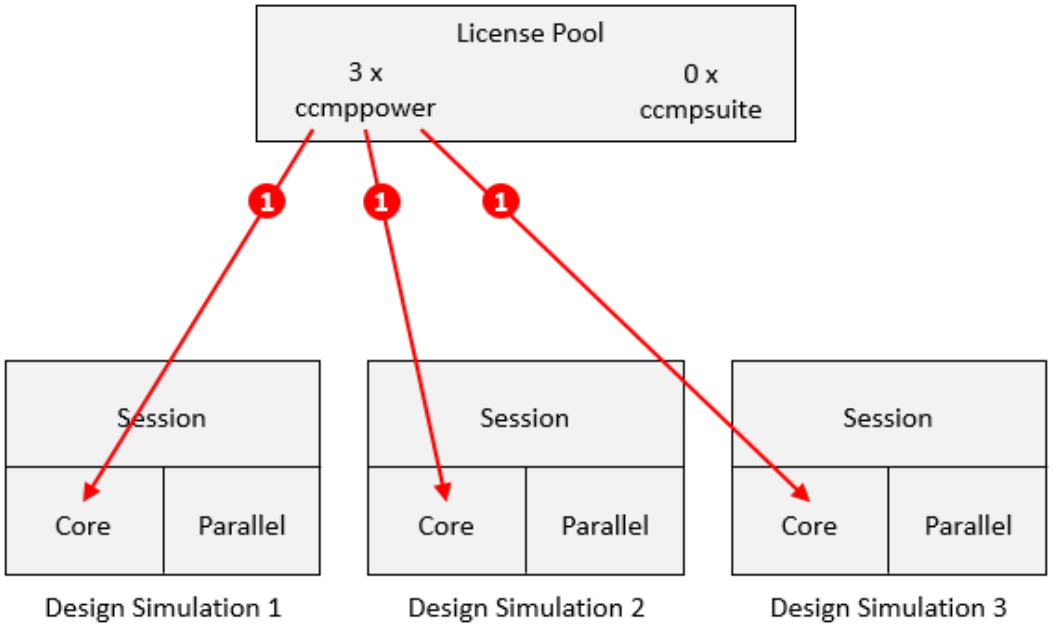
Licensing Scheme	License Usage
	
Unlicensed Design Manager Server	

In this case, the default licensing scheme is more appropriate: The Unlicensed Design Manager Server scheme saves 3 `DOEtoken` but requires two more `ccmpsuite`, which are more valuable than `DOEtoken`.

Example 2: Suppose you have the following licenses available:

- 3 `ccmpsuite`
- 3 `ccmppower`

Again, for 3 concurrent design simulations, each on 16 processors, the following licenses are pulled:

Licensing Scheme	License Usage
Default	
Unlicensed Design Manager Server	

In this case, the Unlicensed Design Manager Server scheme is more appropriate, as it requires no `ccmpsuite`.

Common Licensing Scenarios

The following table lists common licensing scenarios and their associated command line options:

License Usage	Design Manager License Options	Design Simulation License Options
Default behaviour First take Power Tokens (`DOEtoken`), then fallback to STAR-CCM+ product licenses (`ccmpsuite`).		
Power option* First take Power Tokens, then fallback to Power Sessions (`ccmppower`).		`-power`
Power Tokens only		`-nosuite`
Power Sessions only*	`-dmnoshare`	`-power`
Power on Demand*	`-dmnoshare`	`-power -podkey <value> -licpath 1999@flex.cd-adapco.com`

*This license usage is not supported in Pre-Allocation mode.

For more information, see Command Line Options Reference.

You must pass the Design Simulation License Options directly to the design simulations. Design Manager allows you to pass these license options from the GUI (using the STAR-CCM+ Command Line Options property) or from the command line (using the -passtodesign <options> option).

Example:

The following `starccm+` command runs the design studies in the Design Manager project `[PROJECT].dmprj` locally from a batch client using Power Sessions:

```
[STAR-CCM+_INSTALL_DIR]/star/bin/starccm+ -batch run [PROJECT_ROOT_DIR]/
[PROJECT].dmprj -dmnoshare -passtodesign "-power"
```

Design Manager Workflow

The following steps outline the recommended workflow to set up, run, and post-process a design study in Design Manager.

Every design study in Design Manager is based on a STAR-CCM+ reference simulation.

To perform a design study in Design Manager:

1. Set up the reference simulation in STAR-CCM+, focusing on what you want to achieve in Design Manager:
   a. If you want Design Manager to modify the geometry of your product, build and parameterize your geometry using one of the following approaches:
      - Set up the geometry in 3D-CAD using design parameters. See Using 3D-CAD and Working with Design Parameters.
      - Set up the geometry in your CAD tool using design parameters and import the geometry directly from the respective CAD client. This approach transfers the design parameters in addition to the geometry model. See **CAD Clients**.
   b. Set up your simulation with suitable mesh and physics settings.

      Construct the mesh in a manner that accounts for expected geometry variations. For example, use refinement zones that capture the varying geometry of the product. Field functions allow you to dimension refinement zones and cell sizes as functions of design parameters.
   c. If you want Design Manager to modify physical quantities such as flow inlet velocity, define and apply global parameters to the physical quantities of interest. See Working with Global Parameters.
   d. Create reports for all objectives and constraints.

      Additionally, create reports for quantities that you want to monitor for each design. Examples are pressure drop, lift force, drag force, or the Reynolds number.
   e. Create the relevant scenes and plots that you want Design Manager to use for post-processing.

      Make sure that the scenes and plot are appropriate for design comparison. For example, for scenes, use fixed minimum and maximum values for the displayed solution fields so that the color bars are identical for all designs. However, in case you expect large variations in the displayed solution fields, use variable maximum and minimum values.
   f. Make sure that the stopping criteria are appropriate for running various designs.

      To reach convergence, some designs might require less iterations than the reference design, others might need a few more. For this reason, avoid a fixed number of iterations as only stopping criterium. Instead, combine the fixed number of iterations criterium with monitor-based criteria where possible. See Setting Up Stopping Criteria.
   g. If your simulation requires specific input files, set absolute paths to these files. Examples:

    - If your simulation connects to a remote STAR-NX CAD client, set the absolute path to the CAD file location.
    - If your simulation uses table data files, set the absolute File Path.
2. Create a Design Manager project.
3. Set up a design study.
4. Run the design study.
5. Monitor the running design study.
6. Analyze the results of the design study.

For practical examples of performing design studies in Design Manager, refer to the following tutorials:

- Design Manager: Optimizing a Static Mixer
- Design Manager: Design Sweep of a Static Mixer

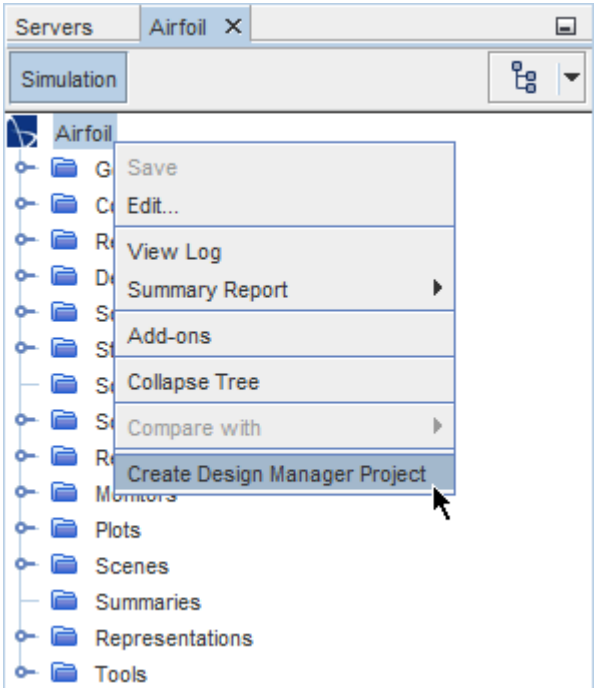
Contents:

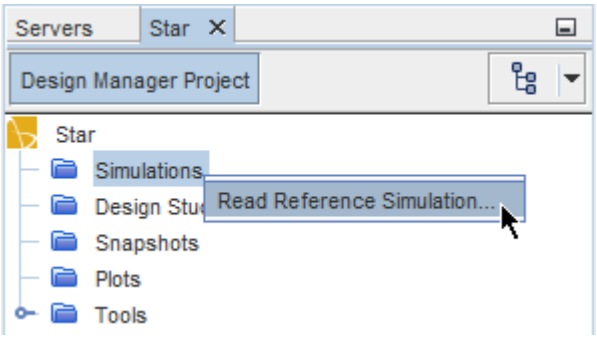
Creating a Design Manager Project
Setting Up a Design Study
Running a Design Study
Monitoring a Design Study
Analyzing the Results of a Design Study

Creating a Design Manager Project

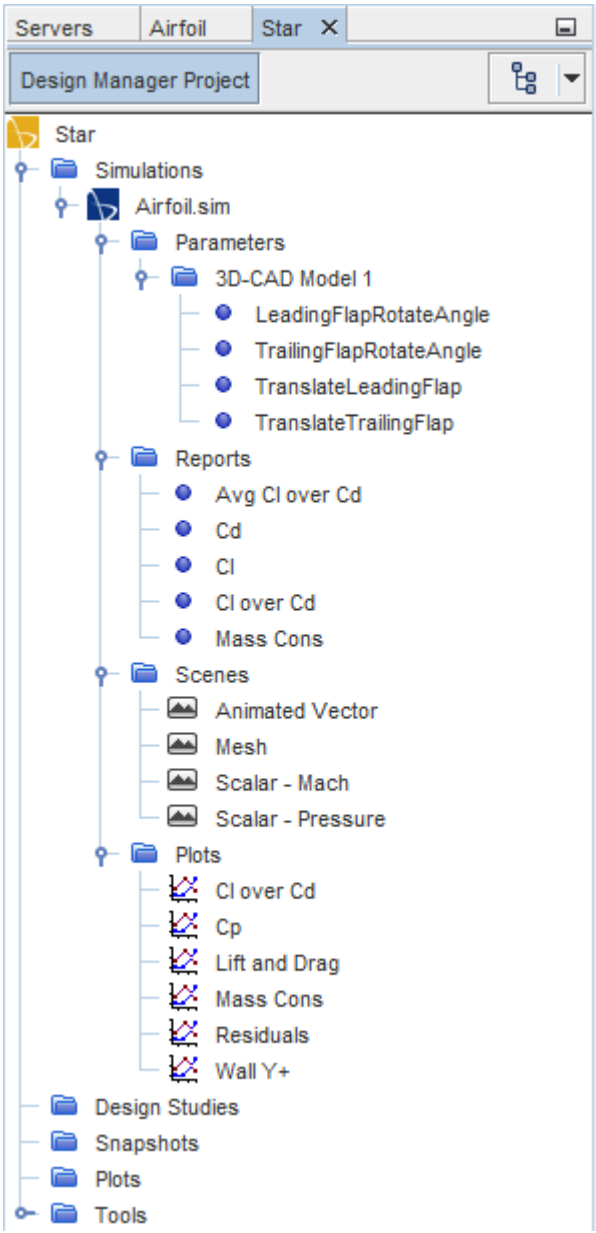
The process for creating a Design Manager project is integrated within the same user interface as you use for a STAR-CCM+ simulation.

There are two ways to create a Design Manager project:

Method	Procedure
Creating a Design Manager project from the reference simulation	1. In the STAR-CCM+ simulation tree, right-click the root node and select **Create Design Manager Project**.  The Design Manager object tree appears. Parameters, reports, scenes, and plots are transferred from the simulation to the Design Manager project.
Creating an empty Design Manager project and reading the reference simulation	1. Launch STAR-CCM+. 2. Click (**Create a File**). 3. In the *Create a File* dialog, set the following properties: • **Type**: **Design Manager Project** • **Command**: For the Unlicensed Design Manager Server scheme, add `-dmnoshare`, see Design Manager Options.

Method	Procedure
	Example: `starccm+ -server -cpubind -rsh ssh -dmproject -dmnoshare` For default licensing, no additional argument is necessary. 4. Click **OK**. The Design Manager object tree loads. 5. Right-click the **Simulations** node and select **Read Reference Simulation...**.  6. In the *Select Reference Simulation* dialog, browse to the corresponding directory, select the reference `.sim` file, and click **Open**. 7. Click **OK**. While STAR-CCM+ reads the reference simulation, it launches a server process. The relevant contents of the simulation file are read in to the Design Manager project.

Example: The screen-shot below displays the Design Manager object tree after transferring parameters, reports, scenes, and plots from a reference simulation.



Contents:

Saving, Closing, and Loading a Design Manager Project
Updating a Design Manager Project

Saving, Closing, and Loading a Design Manager Project

The Design Manager project settings are stored in a separate file, called Design Manager project file (`*.dmprj`).

Objective	Procedure
Saving a Design Manager project	1. In Design Manager, select **File** > **Save**.

Objective	Procedure
	2. In the *Save* dialog, enter the name and location where you want to save the project. 3. Click **Save**.
Saving a copy of a Design Manager project	1. In Design Manager, select **File** > **Save As**. 2. In the *Save* dialog, enter the name and location. 3. Click **Save**.
Closing a Design Manager project	1. In Design Manager, click the **X** button next to the Design Manager project in the *Explorer* pane.
Loading a Design Manager project	1. Launch STAR-CCM+. 2. Select **File** > **Load**. 3. In the *Load a File* dialog, set **Type** to **Design Manager Project**. 4. Navigate to and select the `*.dmprj` file of your Design Manager project. 5. Click **OK**.

Updating a Design Manager Project

As you set up a design study, you are likely to find that you require other parameters, reports, scenes, or plots from the reference simulation. You can not create these object in the Design Manager project. Instead, you must go to the reference simulation directly.

To apply changes to the reference simulation:

1. Load the reference simulation:
   a) Right-click the **Simulations** > **[reference simulation]** node and select **Load...**.
   b) In the *Load a File* dialog, set the startup options (see Interactive Startup Reference) and click **OK**.
      The simulation launches in its own explorer panel.

   Alternatively, you can open the reference simulation in a separate STAR-CCM+ workspace.
2. Modify the simulation.
3. Save and close the reference simulation.

For the following modifications, you must update the reference simulation in Design Manager:

- Creating or removing parameters, reports, scenes, or plots.
- Changing parameter values.
- Renaming parameters, reports, scenes, or plots.

For all other modifications, such as changing the scalar field of a scene, the following step is not required:

4. Right-click the **Simulations** > **[reference simulation]** node and select **Update**.
   The Design Manager project reflects the latest changes.

For more information, see Simulations Reference.

Setting Up a Design Study

Setting up a design study requires that you choose the type of study, set its properties, and choose appropriate input parameters and responses. Additionally, you choose scenes and plots that you want Design Manager to export from each design simulation.

Design Manager supports multiple design studies in one Design Manager project.

To set up a design study:

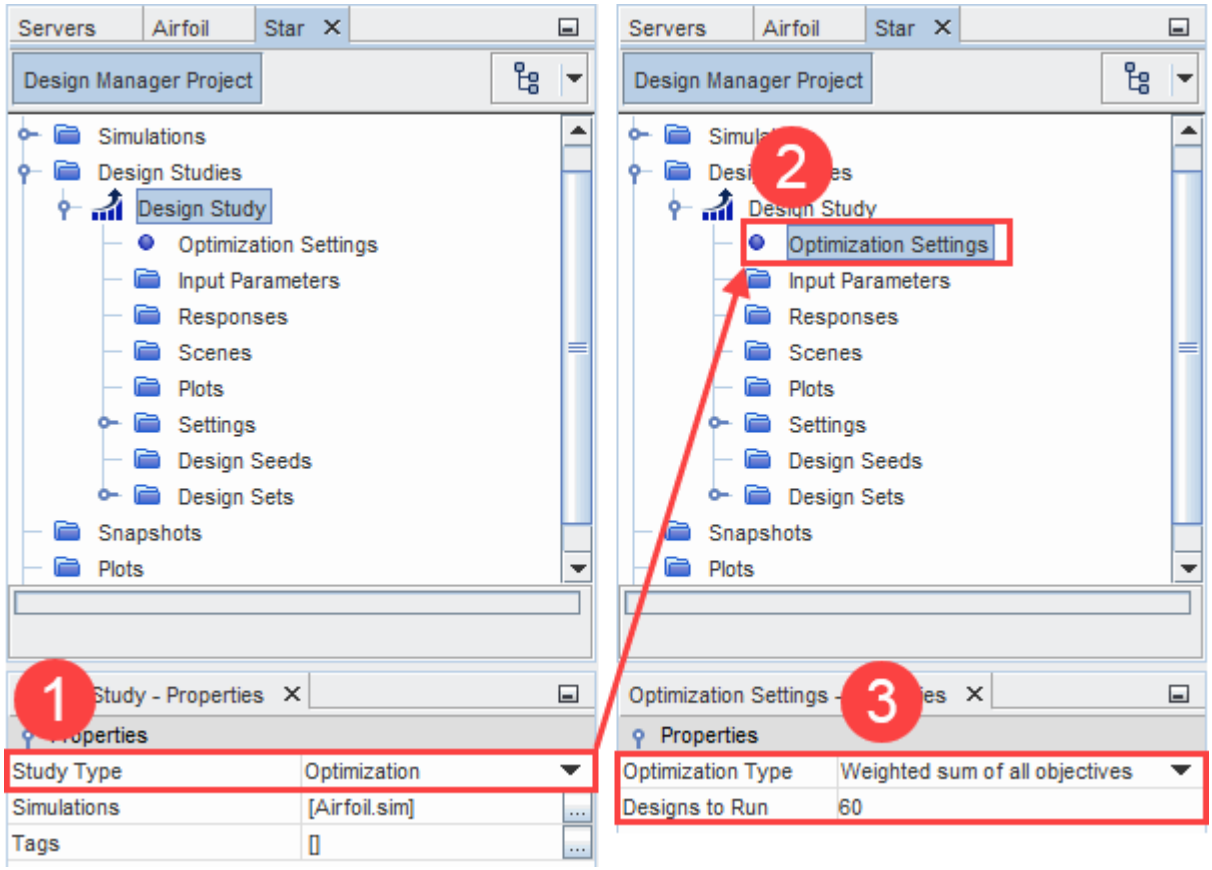
1. Right-click the **Design Studies** node and select **New**.

   STAR-CCM+ creates a **[Design Study]** node, which you can rename as desired. If you run the design study on Windows, make sure that the design study name has less than 49 characters.
2. Configure the design study. For any of the child nodes, you can copy and paste nodes from other studies in the project.
   a) Select the **Design Studies** > **[design study]** node and set *Study Type* to the type of design study that you want Design Manager to run, such as **Optimization**.

      For information on the various study types, see Study Types.
   b) Set *Simulation* to the STAR-CCM+ simulation which you want the design study to reference.

      For each design in a design study, Design Manager runs a copy of the reference simulation using the settings that you define in the design study.
   c) For study types **DOE**, **Optimization**, or **Stochastic**, select the **[design study]** > **[study type] Settings** node and set additional properties.

      Example: For an **Optimization** study, select the **Optimization Settings** node and set the *Optimization Type* and the *Designs to Run*.



For detailed information on the settings that Design Manager requires for each study type, see Study Type Settings Reference.

3. Define the input parameters that you want Design Manager to modify in the design study.

   Design Manager input parameters are the parameters from the STAR-CCM+ reference simulation that Design Manager can vary.

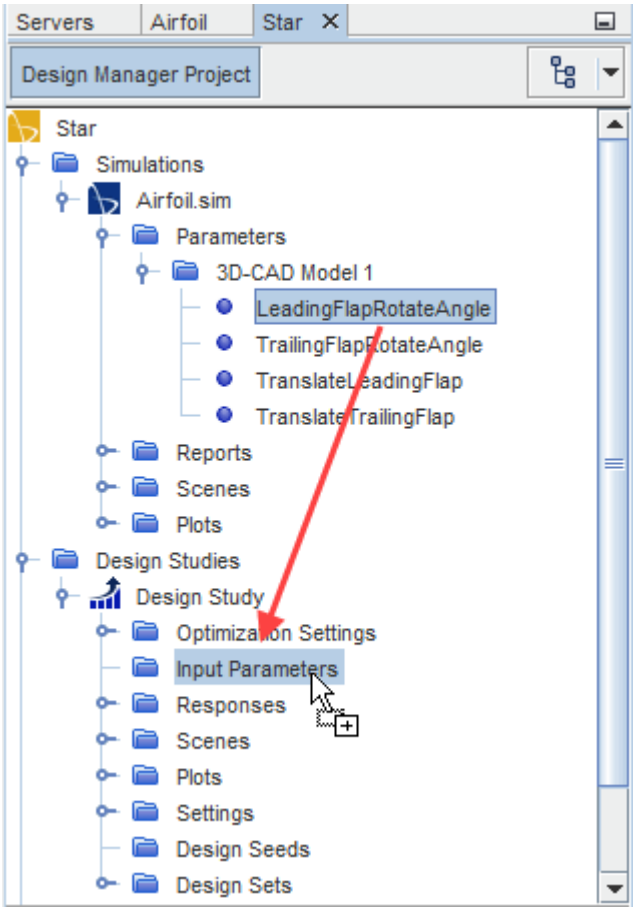
   Each study requires at least one input parameter. A **DOE** study requires at least two input parameters to run.

   For information on the supported input parameters, see Study Inputs.

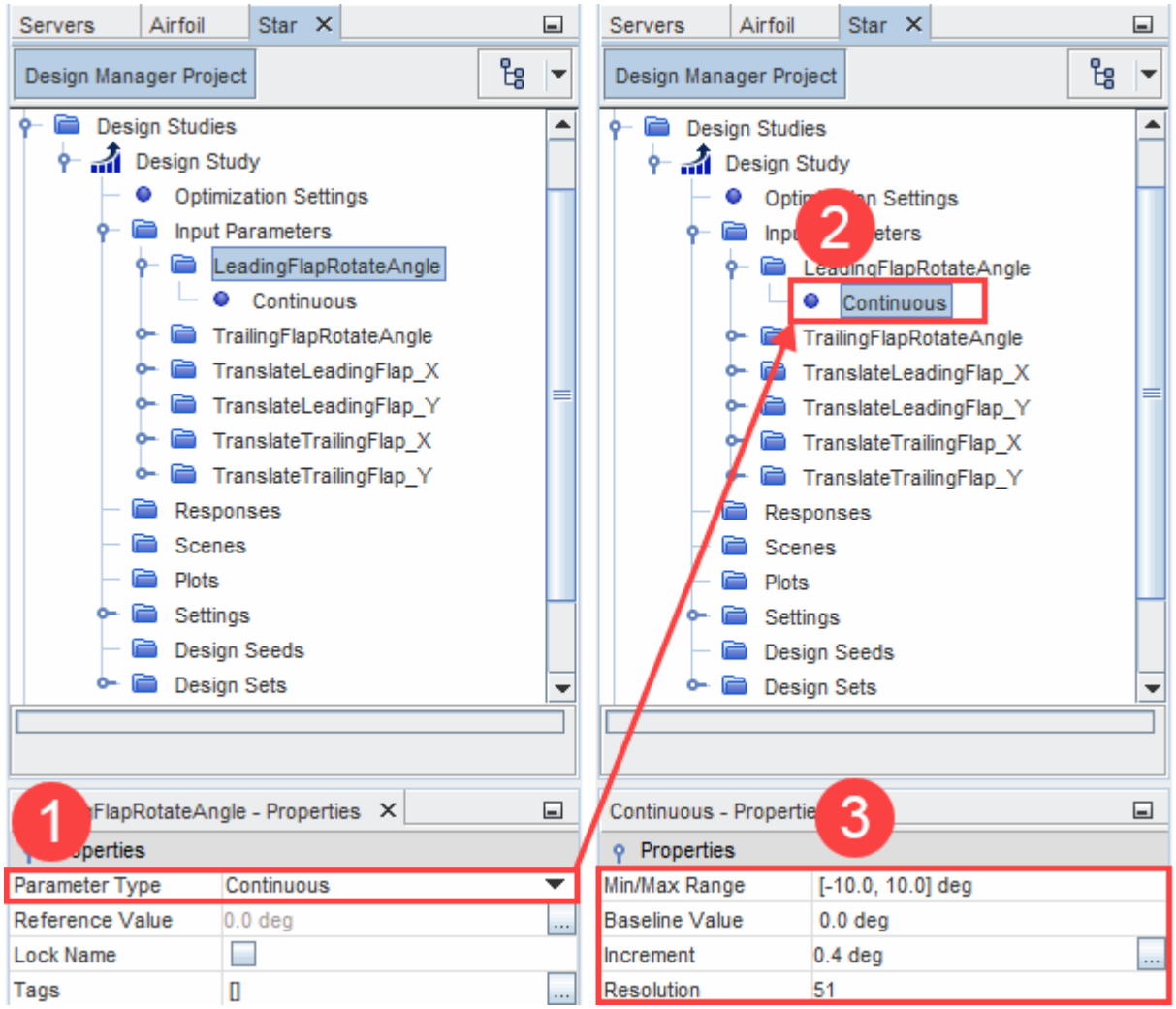
   To set input parameters:

   a) Select the **[design study]** > **Input Parameters** node and set *Parameters* to the specific parameters that you want to control.

      Alternatively, you can drag-and-drop variables from the **Parameters** node of the reference simulation to the **Input Parameters** node of the design study.



b) For each parameter, select the respective **[design study]** > **Input Parameters** > **[input parameter]** node and set its properties. The list of properties varies according to the study type.
Example: For an input parameter in an **Optimization** study, you must set the *Parameter Type*. Setting the *Parameter Type* to **Continuous**, allows Design Manager to choose the input parameter values within a specified range. To control the range limits and the values within the range, select the **Continuous** child node and set the *Min/Max Range*, the *Baseline Value*, and the *Increment* or *Resolution*, respectively.



For more information, see Input Parameters Reference.

Note: For a **Manual** study, you provide a table that contains parameter values for the designs that you want Design Manager to run. See Creating Manual Designs.

4. For an **Optimization** study, you can supply the optimization algorithm with guesses of good design. By this means you can potentially reduce the path from the initial to the best design. You can either enter designs manually or use designs from a previous design study.

   For more information, see Seeding an Optimization Study with Predefined Designs.

5. Set the responses whose values you want Design Manager to record at the end of each design simulation or which influence the study as objectives or constraints.

   Responses represent the outputs of your design study and are based on reports from the reference simulation. Each study requires at least one response.

   To create responses, select the **[design study]** > **Responses** node and, for *Response*, choose one or more reports from the reference simulation.

   Alternatively, you can drag-and-drop reports from the **Reports** node of the reference simulation to the **Responses** node of the design study.

   For more information, see Responses Reference.

6. Set the objectives of the design study.

   For an **Optimization** study, objectives define the goals of the optimization algorithm. A **Weighted sum of all objectives** optimization study requires at least one objective and a **Multiple objective tradeoff study (Pareto front)** requires at least two objectives.

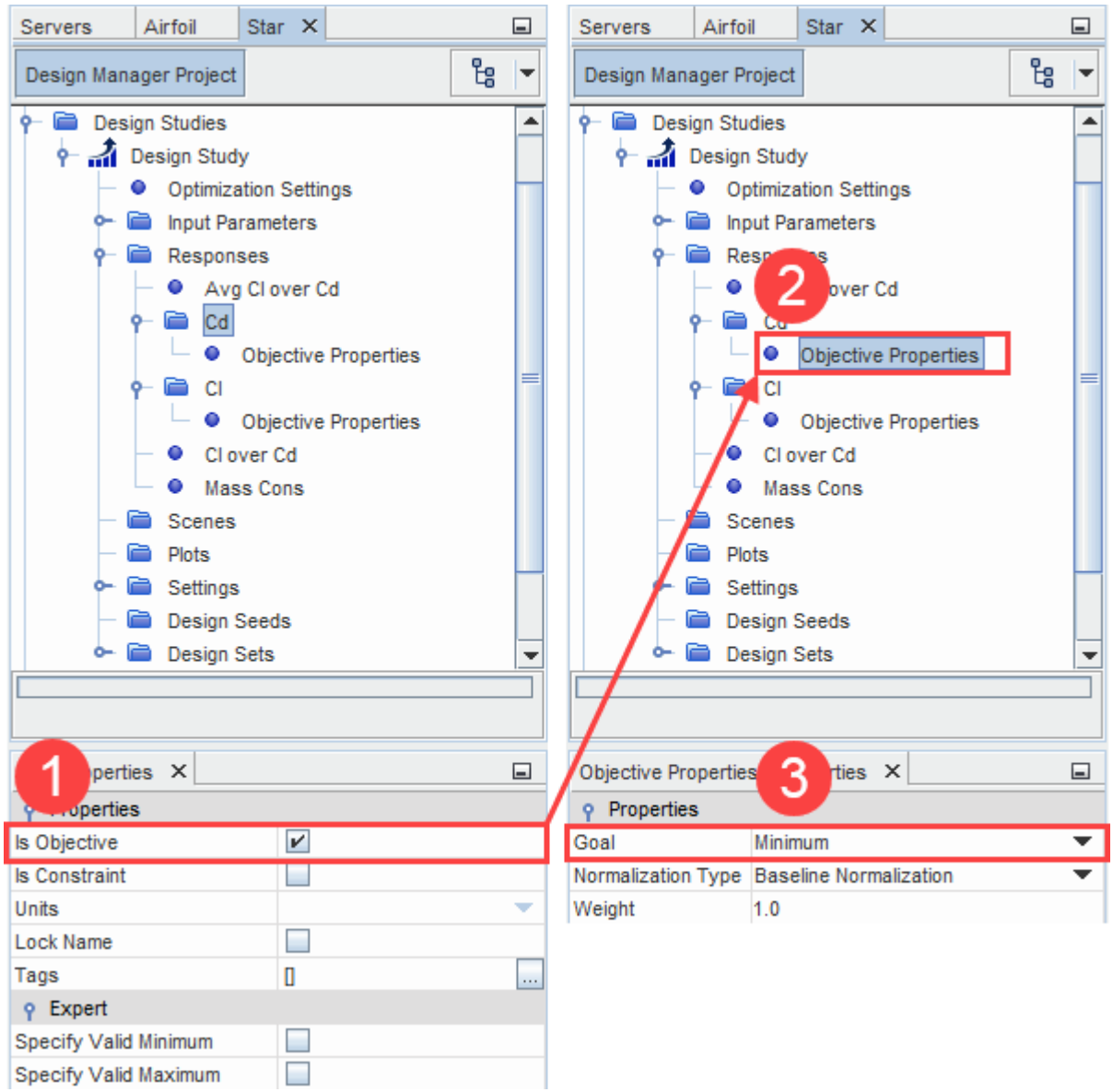
   For a **Sweep** study, objectives are optional and only affect the performance value of the designs.

   To set objectives:

   a) Create responses for the reports whose values drive the optimization.

   b) For each response, select the respective **Responses** > **[response]** node and activate *Is Objective*.

   c) Select the **Responses** > **[response]** > **Objective Properties** node and set *Goal* to **Maximum** or **Minimum**.



For more information, see Objective Properties.

7. Set the constraints of the design study (optional).

   Constraints allow you to define the design space in which acceptable designs are found.

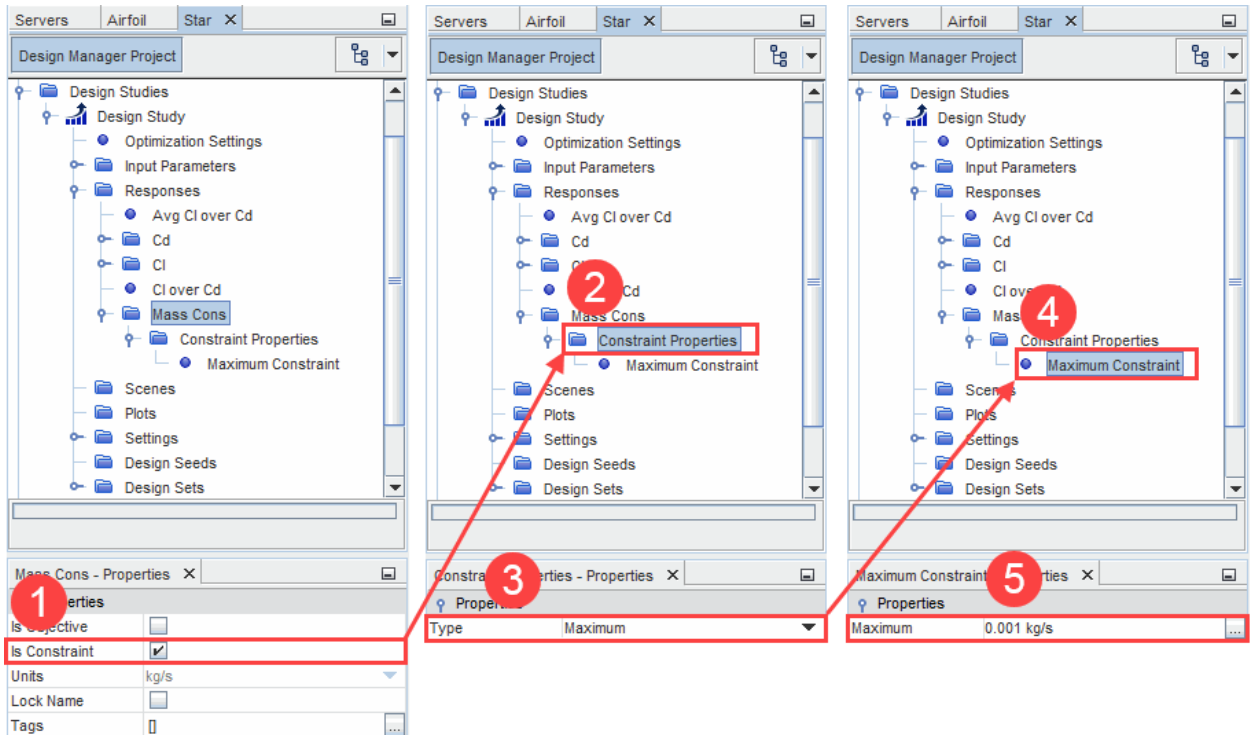
   To set constraints:

   a) Create responses for the reports whose values you want to constrain.

   b) For each response, select the respective **Responses** > **[response]** node and activate *Is Constraint*.

   c) Select the **Responses** > **[response]** > **Constraint Properties** node and set *Type* to the type of constraint that you want to impose, such as **Maximum**.

   d) Select the **Responses** > **[response]** > **Constraint Properties** > **[constraint]** node and set the constraint value.



   For more information, see Constraint Properties.

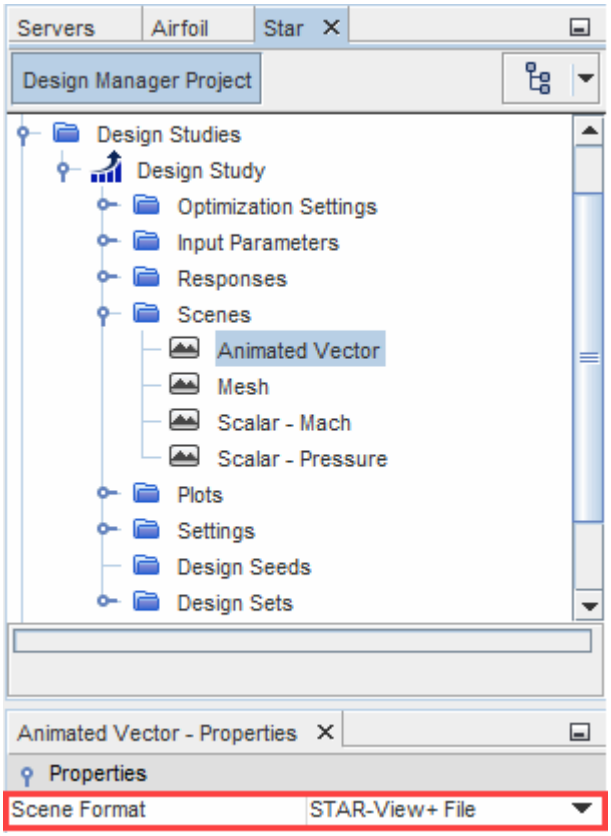
8. Set the scenes and plots that you want Design Manager to export from each design simulation.

   To set scenes:

   a) Select the **Scenes** node and set *Scenes* to the selection of scenes that Design Manager should export.

      Alternatively, you can drag-and-drop scenes from the **Scenes** node of the reference simulation to the **Scenes** node of the design study.

   b) For each scene, select the respective **Scenes** > **[scene]** node and set the *Scene Format* to the file format to which you want to export the scene, such as **STAR-View+ File**.



c) For plots, repeat the same procedure using the **Plots** node.

For more information, see Scenes and Plots Reference.

9. If you want to add extra processing steps in the workflow for each design simulation, you can add Java macros that Design Manager executes for each design. You script the required steps in a Java file and add the file to the design study. Design Manager can perform these steps before meshing, after meshing, or after running. For more information, see Customizing the Workflow Using Java Macros.
10. Select **File** > **Save As...** and save the project.

Contents:

Creating Manual Designs
Seeding an Optimization Study with Predefined Designs
Customizing the Workflow Using Java Macros

Creating Manual Designs

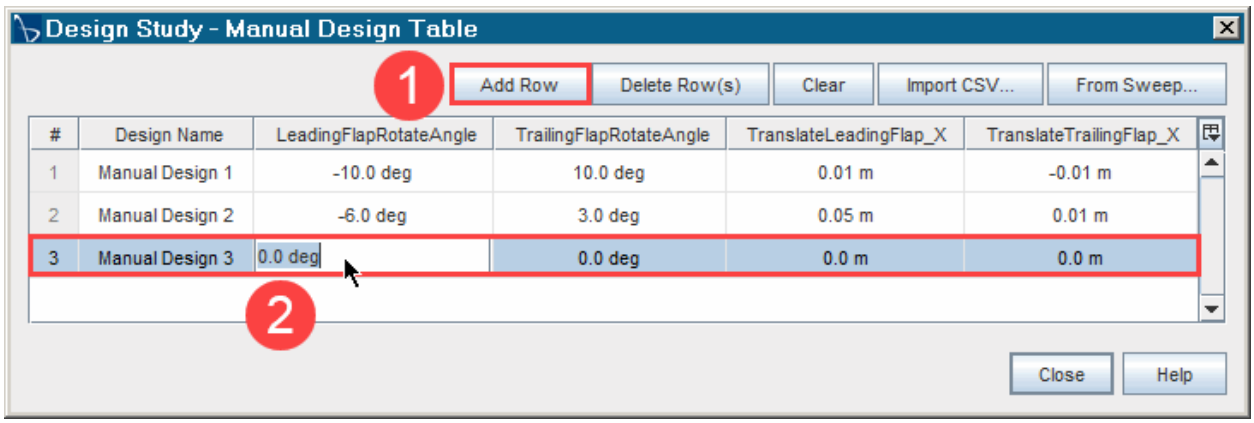
When performing a Manual study or when seeding an Optimization study, you can define the respective designs using a table editor. You can enter the tabular data manually, import a `.csv` file, or create the tabular data using a design sweep table.

Before creating manual designs, define the input parameters of your design study.

To instruct Design Manager to use tabular data:

1. Launch the table editor:
   - For a Manual study, edit the **[design study]** > **Design Table** node.
   - For an Optimization study that you want to seed with a manual design seed (see Seeding an Optimization Study with Predefined Designs), edit the **[design study]** > **Design Seeds** > **[manual design seed]** node.
2. In the *Manual Design Table* dialog, click **Add Row** and enter the appropriate values for each parameter.

   You can add further designs using the same technique.



To import designs from a `.csv` file:

3. In the *Manual Design Table* dialog, click **Import CSV...**.

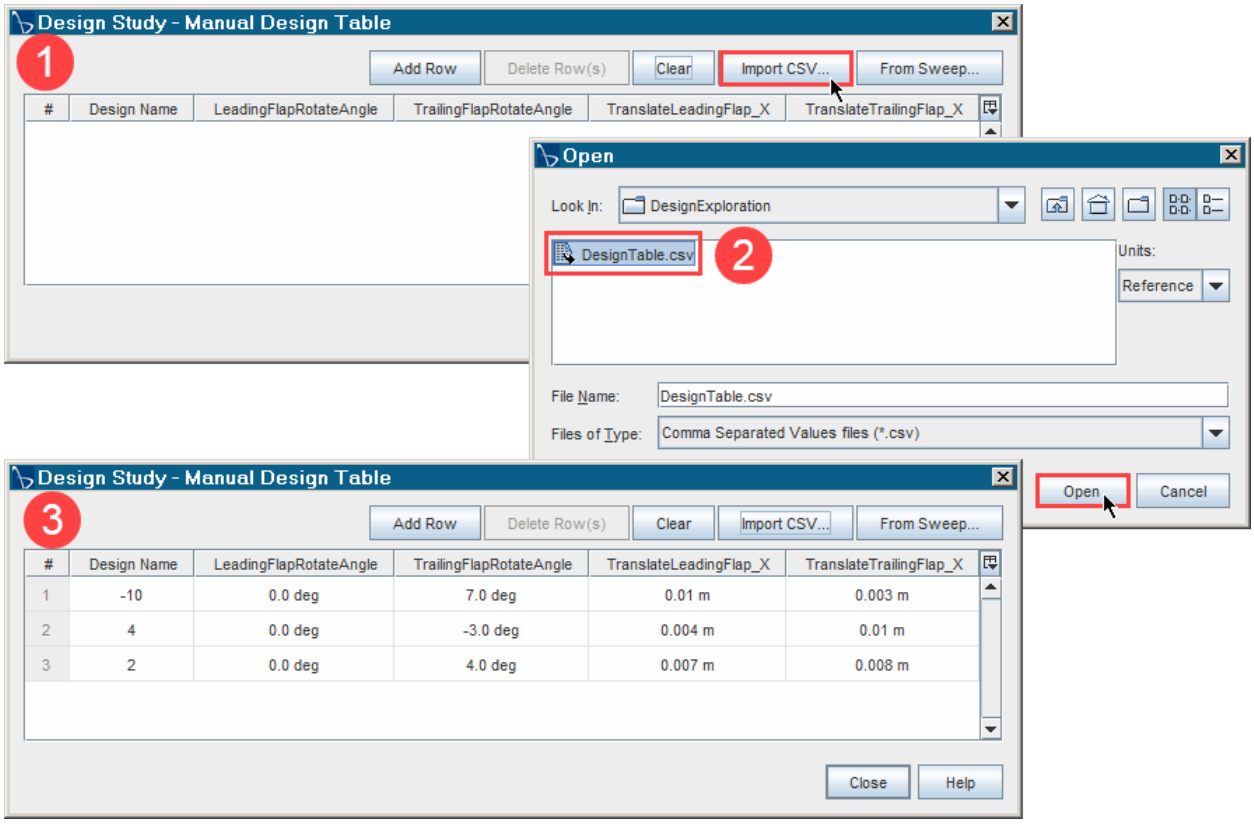
   An example format for the `.csv` file data is provided below:

```
LeadingFlapRotateAngle, TrailingFlapRotateAngle,
TranslateLeadingFlap_X, TranslateTrailingFlap_X
-10, 7, 0.01, 0.003
4, -3, 0.004, 0.01
2, 4, 0.007, 0.008
```

Note: The names of the columns must match the names of the input parameters that you want to define using this file.

4. In the *Open* dialog, navigate to the desired `.csv` file and click **Open**.

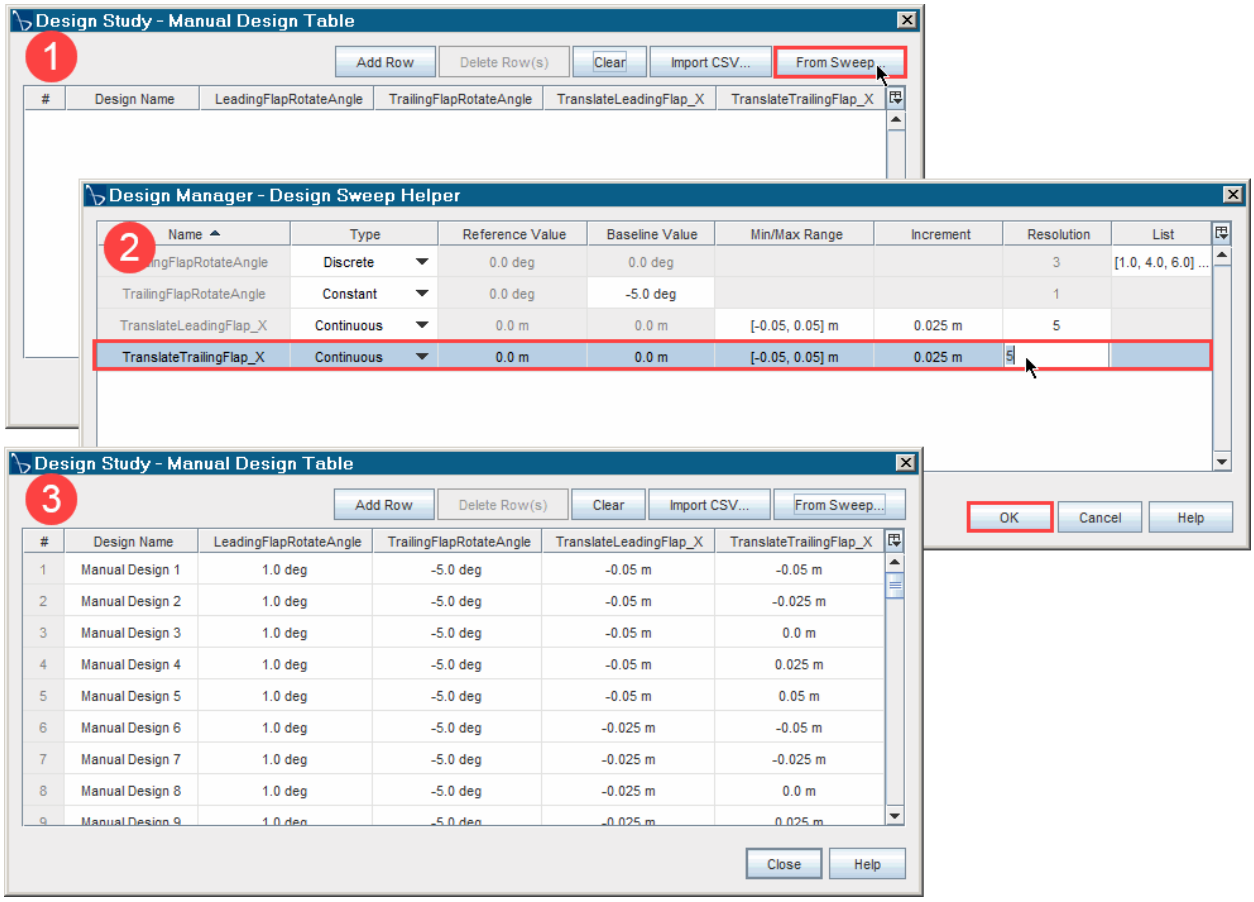
   New design rows appear, where the design parameters display the data from the file.



To create tabular data using a design sweep table:

5. In the *Manual Design Table* dialog, click **From Sweep...**.
6. In the *Design Sweep Helper* dialog, set the *Type* for each input parameter, such as **Continuous**.
7. Set the properties that are required for the respective parameter type, such as *Min/Max Range*, *Increment*, and *Resolution*. For more information, see Input Parameters Reference—Parameter Type.
8. Click **OK**.

   The design table displays the designs that result from a sweep through the specified parameter values.



To remove designs:

9. Select the relevant row and click **Delete Row(s)**.

   Using `<Ctrl>`, you can select and delete multiple rows.

   To delete all designs, click **Clear**.

To close the table editor:

10. Click **Close**.

Seeding an Optimization Study with Predefined Designs

When seeding an Optimization study, you can enter designs manually or refer to designs of a previous design study.

To seed an Optimization study with specific designs that you enter manually:

1. In the Optimization study, right-click the **Design Seeds** node and select **New Design Seed** > **Manual Design Seed**.
2. Edit the design seed node and, in the *Manual Design Table* dialog, set the designs with which you want to seed the design study. See Creating Manual Designs.

To seed an Optimization study with designs of a previous design study:

3. Within the previous design study, create a design set that contains the relevant designs. See Creating Design Sets. If required, create multiple design sets.
4. If you want to use designs from a previous study that belongs to another Design Manager project, first export the desired designs from that project and import them into the current project. To export and import designs:
   a) Open the relevant Design Manager project containing the previous design study.
   b) In the previous design study, right-click the relevant design set node and select **Export to CSV...**.
   c) In the *Export to CSV* dialog, save the design set as `.csv`.
   d) If no further exports are required, close the previous Design Manager project.
   e) In the current Optimization study, right-click the **Design Seeds** node and select **New Design Seed** > **Manual Design Seed**.
   f) Import the `.csv` file. See Creating Manual Designs.
   g) Repeat steps a. to d. to refer to multiple design sets.
5. If you want to use designs from an earlier design study within the current Design Manager project, you can refer directly to design sets. To choose design sets within the current project:
   a) Right-click the **Design Seeds** node and select **New Design Seed** > **Seed From Design Set**.
   b) Select the design seed node and set *Design Sets* to the chosen design sets from the previous study.

Design Manager evaluates the specified designs and supplies the results to the optimization algorithm.

Customizing the Workflow Using Java Macros

You can customize the Design Manager workflow using Java macros.

There are two instances in the Design Manager workflow where you can use `*.java` macros to perform custom tasks:

- When setting up and starting design studies in the Design Manager project
- When running the STAR-CCM+ design simulations

Customizing the Design Manager Project

Similar to STAR-CCM+ (see Scripting the Application), you can script the process of setting up and starting design studies in Design Manager. Scripting the Design Manager application is particularly useful when you want to automate repetitive tasks or when you want to perform certain actions within the project when running in batch (locally or on a cluster).

Example: Running a Design Manager project in batch using the default command option (`-batch run`) runs all design studies within the project sequentially from first to last. However, if you want to run only one or more specific design studies, you can record a macro in Design Manager that runs only the relevant studies. You then specify this Java macro on the command line when running the project. For more information, see Running from a batch client.

Customizing the STAR-CCM+ Design Simulations

When running a particular design of a design study, Design Manager launches STAR-CCM+, loads the reference simulation, modifies the relevant parameters, and runs the simulation until certain stopping

criteria are met. You can expand this default workflow by adding custom macros to the respective design study.

There are three areas where you can expand the workflow:

- Before meshing
- After meshing
- After running

Example: One example of how you can customize the workflow is to change the run settings and then re-run the solver. You can record a macro in STAR-CCM+, where you modify the run settings, for example the Mach number, extend the number of iterations in the stopping criteria, and then re-run the solver. Using this technique you can perform two runs with different mach numbers for a single design. You then insert this macro after running the solver. When Design Manager reaches this stage, it performs the custom steps and then proceeds with the remaining default workflow.

To customize the STAR-CCM+ design simulations:

1. In STAR-CCM+, record the steps that you want to include in the design simulation workflow.

   For more information, see Recording a Macro.
2. In Design Manager, set up your design study.
3. Right-click the **Settings** > **Macro Files** node and select **New**.
4. In the *Open* dialog, navigate to and select the appropriate `*.java` file.
5. Set *Macro Insertion Point* to the option that is relevant to your situation:
   - **Before Meshing**—if you want to perform the actions before executing the meshing pipeline.
   - **After Meshing**—if you want to perform the actions after executing the meshing pipeline.
   - **After Running**—if you want to perform the actions after the STAR-CCM+ solver completes, and before any results are returned to Design Manager.
6. Click **Open**.

   The macro is added to the **Macro Files** node.

   You can add multiple macros to a design study. For the list of available macro file properties, see Macro Files Reference.

When you run the Design Manager study, the custom steps are run at the appropriate stage of the workflow.

Running a Design Study

Design Manager provides various settings to control the run of a design study and the compute resources to use.

During the run of a design study, Design Manager creates a project artifact directory within your project root directory, which contains all the outputs of the design study.

Note: You can access the artifact directory from the *Design Details* panel. To access the *Design Details* panel, right-click a design in the *Output Table* and select **Show Design Details**. In the *Design Details* panel, click the link in the *Design Artifact* entry to browse the design artifact folder.

The following table displays the directory and file structure within your project root directory:

Directories	Description	Naming Convention
Design_Exploration	Directory that contains the Design Manager project file (`*.dmprj`) and associated input files such as the reference simulation file (`*.sim`) and custom java macro files (`*.java`). **Note:** You must run the design study from this directory. Running design studies from a symbolically linked directory is not supported.	Project root directory
∟ **Design_Manager_Project**	Directory that is created at run time and contains the study artifact directories. This directory is named after the Design Manager project file.	Project artifact directory
∟ **Design_Study_n**	Directory for each design study. This directory is named after the study name in the Design Manager project and contains the design artifact directories.	Study artifact directory
∟ **Design_m**	Directory for each design in the study, which contains the specified outputs, such as: • Simulation file (`*.sim`) • Log file (`*.log`) • Scene and plot files (`*.sce` or `*.png`)	Design artifact directory
∟ **.mdxruntime**	Temporary folder that is created at run time and deleted after the run. This directory contains a replication of the project artifact directory.	Project execution directory

Note: You can customize the path to the project artifact directory within your Design Manager project. See Design Manager Reference.

Contents:

Running Locally
Running on a Cluster Using General Job Submission
Running on a Cluster Using Pre-Allocation
Re-Running a Design Study with Modified Settings
Stopping, Extending, and Resuming a Design Study
Resuming a Design Study after Abort

Running Locally

For small simulation sizes, a local machine with a high number of cores and memory can give sufficient compute resource for a design study.

To run design studies on your local machine:

1. For each design study in your Design Manager project, set the run and resource settings:
   a) Select the **Design Studies** > **[design study]** > **Settings** > **Run Settings** node, where [design study] represents the name of the design study, and set the following properties:
      - *Save Simulation Files*: Specify the designs whose STAR-CCM+ simulation files you want to save.
      - *Save Log files*: Choose whether to save log files to the design directories or not.
      - *Auto Save*: Control whether to save the Design Manager project after each design simulation completes. You are strongly advised to activate this option. If the Design Manager project crashes for any reason, you can resume the project from the completed designs. However, be aware that auto-saving during the run can slow down the process.
      - *Simultaneous Jobs*: Set the number of design simulations to run concurrently.
      - *Compute Processes*: Set the number of cores to run with each design simulation.

      When setting the numbers for *Simultaneous Jobs* and *Compute Processes*, check that you have enough cores available (simultaneous jobs * compute processes).

      For more information, see Run Settings Reference.
   b) Select the **Compute Resource** node and set the following properties:
      - *Type*: **Direct**
      - *STAR-CCM+ Command Line Options*: Set command line options to pass to the design simulations, such as License Options or other STAR-CCM+ options. See Command-Line Reference.

        Example: `-power` (Power Session licensing).

      For more information, see Compute Resource Reference.
2. You can run design studies on your workstation directly from the Graphical User Interface (GUI) or in batch mode from the command prompt:

Method	Procedure
Running directly from the GUI	• To run a specific design study, right-click the design study node and select **Run Study**. • To run all design studies sequentially from first to last, right-click the **Design Studies** manager node and select **Run All Studies**. The design simulations are submitted in the background. The output from the Design Manager server is displayed in the *Output* window. The STAR-CCM+ servers for each running design simulation are visible in the STAR-CCM+ server list.
Running from a batch client For more information, see Running the Client in Batch.	• If you want to run specific design studies: 1. Save the following Java code to a macro file (`*.java`) and name it `runSpecificStudies`:

```
// STAR-CCM+ macro: runSpecificStudies.java

package macro;

import java.util.*;
import star.common.*;
import star.mdx.*;

public class runSpecificStudies extends
MdxMacro {

  public void execute() {

    MdxProject mdxProject_0 =
getActiveMdxProject();

    String[] designStudies = {"Design Study
1", "Design Study 2", "Design Study 3"};

    for (int i = 0; i < designStudies.length; i
++){

        MdxDesignStudy mdxDesignStudy_0 =
mdxProject_0.getDesignStudyManager().getDesignS
tudy(designStudies[i]);

        mdxDesignStudy_0.runDesignStudy();

    }
```

```
    }
}
```

In the Java code above, the `designStudies` array defines the design studies to run. Replace the default design study names (`Design Study 1`, `Design Study 2`, and `Design Study 3`) with the names of the design studies that you want to run.

2. Move the macro file to your project root directory, which contains the relevant Design Manager project file (`*.dmprj`).
3. Open a command prompt and enter the following command line: `[STAR-CCM+_INSTALL_DIR]\star\bin\starccm+ -batch [PROJECT_ROOT_DIR]\runSpecificStudies.java [PROJECT_ROOT_DIR]\[PROJECT].dmprj [-dmnoshare] [-passtodesigns <options>]`

   This command line only applies on Windows. For Linux, you must use forward slashes.

- If you want to run all design studies sequentially from first to last:
  1. Open a command prompt and enter the following command line: `[STAR-CCM+_INSTALL_DIR]\star\bin\starccm+ -batch run [PROJECT_ROOT_DIR]/[PROJECT].dmprj [-dmnoshare] [-passtodesign <options>]`

where:

- `[STAR-CCM+_INSTALL_DIR]` represents the absolute path to the STAR-CCM+ installation directory.
- `[PROJECT_ROOT_DIR]` represents the absolute path to the project root directory.
- `[PROJECT]` represents the name of the Design Manager project file.
- `-dmnoshare` activates the Unlicensed Design Manager Server license scheme. Omitting this option uses the Default license scheme. For more information, see Design Manager Licensing.
- `-passtodesign <options>` allows you to pass additional simulation command line options, such as license options, to the design simulations. Passing simulation command line options directly from the command line is useful for machine-specific or user-specific options, which you do not want to save within your Design Manager project file (see step 1b). However, command line options that require double quotes are not supported—you must set these options in the *STAR-CCM+ Command Line Options* property. For more information, see Design Manager Options.

Note: When running from a batch client, you must set the absolute path to the project root directory.

Running on a Cluster Using General Job Submission

For medium to large simulations, you are likely to require a cluster on which to run design studies.

General job submission means that Design Manager submits each design simulation individually to the batch queue on the cluster. Before using this approach, consider whether pre-allocation mode would give you better performance.

To run a design study using general job submission:

1. For each design study in your Design Manager project, define the run and resource settings:
   a) Select the **Design Studies** > **[design study]** > **Settings** > **Run Settings** node and set the following properties:
      - *Save Simulation Files*: Specify the designs whose STAR-CCM+ simulation files you want to save.
      - *Save Log files*: Choose whether to save log files to the design directories or not.
      - *Auto Save*: Control whether to save the Design Manager project after each design simulation completes. You are strongly advised to activate this option. If the Design Manager project crashes for any reason, you can resume the project from the completed designs. However, be aware that auto-saving during the run can slow down the process.
      - *Simultaneous Jobs*: Set the number of design simulations to run concurrently.
      - *Compute Processes*: `1`. This value disables the *Compute Processes* property. Instead, you specify the cores to run with each design simulation in the *Job Submit Command*. See step 1c.

      When setting the numbers for *Simultaneous Jobs* and *Compute Processes*, check that you have enough cores available (simultaneous jobs * compute processes).

      For more information, see Run Settings Reference.
   b) Select the **Compute Resource** node and set the following properties:
      - *Type*: **Linux Cluster**
      - *STAR-CCM+ Command Line Options*: Set command line options to pass to the design simulations, such as License Options or other STAR-CCM+ options, see Command-Line Reference.

        Example: `-power -mpidriver platform:"-aff:automatic:bandwidth"` (Power Session licensing and Platform MPI).

      For more information, see Compute Resource Reference.
   c) Select the **Compute Resource** > **Linux Cluster** node and specify the following cluster settings:
      - *Job Submit Command*: Set the command to submit each design simulation to the cluster queue. The default value (`qsub -l nodes=1:ppn=8`) is set up for a cluster running OpenPBS where each design simulation is submitted on one node running eight cores per node.
      - *Script File*: Set the absolute path and the full name of your job submission script on the cluster. Make sure that the job submission script meets the General Job Submission Script Requirements and has execute permission.

      For more information, see Compute Resource Reference—Linux Cluster Properties.
2. Select **File** > **Save** and save the project file.

3. Transfer the Design Manager project file (`*.dmprj`) and associated input files such as the reference simulation file (`*.sim`) and custom Java macro files (`*.java`) to your cluster project root directory.
4. Run the design studies.

   As the general job submission mode leverages the client-server capability of STAR-CCM+, you can run design studies directly from the Graphical User Interface (GUI) on your workstation. Alternatively, you can run design studies on your cluster from a batch client:

Method	Procedure
Running directly from a local GUI client For more information, see Working With Remote Servers. **Note** Running the Design Manager server remotely from your local workstation requires a password-free SSH tunnel between your local workstation and the cluster.	1. On your local workstation, launch STAR-CCM+ and click (**Load a File**). 2. In the *Load a File* dialog, set **Type** to **Design Manager Project**. 3. Activate the **Remote Server** option and set the remote server properties: • **Remote Host**: Specify the name of a cluster node that has the ability to submit jobs to the queue, such as the head node. If you require the domain name to connect from your local machine, include it here. • **Remote User**: Specify the user name that is required to log into the specified remote host. • **Remote Shell**: Choose the type of remote shell command. 4. In the **File** text box, specify the absolute path and the full name of the Design Manager project file in your cluster project root directory. 5. In the **Command** text box, insert the absolute path and the name of the STAR-CCM+ executable on the cluster. To activate the Unlicensed Design Manager Server license scheme, add `-dmnoshare` (optional). For more information, see Design Manager Licensing. 6. Click **OK**. The Design Manager object tree loads. 7. Run the design studies interactively: • To run a specific design study, right-click the design study node and select **Run Study**. • To run all design studies sequentially from first to last, right-click the **Design Studies** manager node and select **Run All Studies**. 8. If your local workstation restarts or shuts down during the night, you can disconnect from and reconnect to the remote Design Manager server from your local client, see Making Client-Server Connections. The Design Manager server remains active on the head node until you close the project.
Running from a batch client	1. On a cluster node that has the ability to submit jobs to the queue, such as the head node, submit the Design Manager project in batch. **Example**:

For more information, see Running the Client in Batch.

```
[STAR-CCM+_INSTALL_DIR]/star/bin/starccm+ -batch
run [PROJECT_ROOT_DIR]/[PROJECT].dmprj [-
dmnoshare] [-passtodesign <options>]
```

where:

- `[STAR-CCM+_INSTALL_DIR]` is the absolute path to the STAR-CCM+ installation directory on your cluster.
- `[PROJECT_ROOT_DIR]` is the absolute path to your cluster run directory.
- `[PROJECT]` is the name of your Design Manager project file.
- `-dmnoshare` activates the Unlicensed Design Manager Server license scheme. Omitting this option uses the Default license scheme. For more information, see Design Manager Licensing.
- `-passtodesign <options>` allows you to pass additional simulation command line options, such as license options, to the design simulations. Passing simulation command line options directly from the command line is useful for machine-specific or user-specific options, which you do not want to save within your Design Manager project file (see step 1b). However, command line options that require double quotes are not supported—you must set these options in the *STAR-CCM+ Command Line Options* property. For more information, see Design Manager Options.

Note: When running from a batch client, you must set the absolute path to the project root directory.

2. To follow the progress of your design studies, you can connect to and disconnect from the running Design Manager server, see Making Client-Server Connections.

The Design Manager server shuts down automatically as soon as all design simulations are complete.

The Design Manager server submits the specified number of simultaneous design simulations to the queue. As soon as one design simulation completes, Design Manager submits the next design simulation. This job submission procedure continues until all design simulations are complete.

The output from the Design Manager server is displayed in the *Output* window. The STAR-CCM+ servers for each design simulation are visible in the STAR-CCM+ server list.

Note: The Design Manager server does not have any capabilities to interact with the job scheduler on your cluster. Design Manager can not detect what type of scheduler is used on your cluster. The tracking of each design—the detection of failure or completion—solely relies on files that are written to disk by each design simulation. Therefore, when a Design Manager project submits a design simulation to the cluster and the submission of the simulation fails or anything unexpected happens in the scheduler the Design Manager server can not detect this failure. However, as soon as a resource is assigned and STAR-CCM+ is executed if the simulation fails or crashes due to any other unexpected issues, then Design Manager detects this failure. Design Manager marks the design simulation as failed and moves to the next design simulation to complete the design study.

Contents:

General Job Submission Script Requirements

General Job Submission Script Requirements

When running a design study, the Design Manager server passes specific instructions to each design simulation. These instructions are communicated via files that are written to the temporary directory `.mdxruntime` that is located in your project root directory. When a design simulation is submitted to the cluster scheduler, the Design Manager server executes the user job submission script in the temporary directory `.mdxruntime/$project/$study/$design`. The `.mdxruntime/$project/$study/$design` directory contains all the files that Design Manager requires to execute the design simulation.

The Design Manager server does not interact with the job scheduler on your cluster. For this reason, in the job submission script, you must read the following files from `.mdxruntime/$project/$study/$design` and redirect them to the Design Manager server:

- `dm_design_execute_options.txt`

  This file contains additional command line options, such as license details, and other options that you define in the design study settings.
- `dm_design_output_info.msg`

  This file contains the name of the log file that Design Manager expects to be available when the design simulation executes.

Within your job submission script, you read these two files using the `cat` command.

Example

The following example shows a general job submission script for tcsh (Tenex-C-Shell):

```
#!/bin/tcsh -f

STARCCMHOME=[STAR-CCM+_INSTALL_DIR]   #[STAR-CCM+_INSTALL_DIR]: set
absolute path to the STAR-CCM+ installation directory on the cluster

# Design Manager definitions
```

```
# ----------------------------------
# dm_design_execute_options.txt contains starccm+ command line
arguments
# dm_design_output_info.msg contains log output file name to dump.

STARTCASE=`cat dm_design_execute_options.txt`
CASELOG=`cat dm_design_output_info.msg`

# Call STAR-CCM+
# ----------------------------------
$STARCCMHOME/star/bin/starccm+ \
  -rsh ssh \
  -batch -batchsystem pbs \
  $STARTCASE \
  >> $CASELOG 2>&1
```

This example script code checks for the files `dm_design_execute_options.txt` and `dm_design_output_info.msg` and assigns the content to two independent variables. The `STARTCASE` variable contains the additional command line options and the `CASELOG` variable contains the name of the log file. These two variables are then used when calling STAR-CCM+ at the end of the script.

Note: For other Unix shells, such as bash, you must modify the syntax accordingly.

Running on a Cluster Using Pre-Allocation

In pre-allocation mode, Design Manager allocates all necessary resources and licenses before the compute simulations begin. You submit the whole Design Manager project to the cluster queue in one step.

To submit the entire Design Manager project as a single job to the cluster queue:

1. For each design study in your Design Manager project, set the run and resource settings:
   a) Select the **Design Studies** > **[design study]** > **Settings** > **Run Settings** node, where [design study] represents the name of the design study, and set the following properties:
      - *Save Simulation Files*: Specify the designs whose STAR-CCM+ simulation files you want to save.
      - *Save Log files*: Choose whether to save log files to the design directories or not.
      - *Auto Save*: Control whether to save the Design Manager project after each design simulation completes. You are strongly advised to activate this option. If the Design Manager project crashes for any reason, you can resume the project from the completed designs. However, be aware that auto-saving during the run can slow down the process.
      - *Simultaneous Jobs*: Set the number of design simulations to run concurrently.
      - *Compute Processes*: Set the number of cores to run with each design simulation.

      When setting the numbers for *Simultaneous Jobs* and *Compute Processes*, check that you have enough cores available (simultaneous jobs * compute processes).

      For more information, see Run Settings Reference.

b) Select the **Compute Resource** node and set the following properties:

- *Type*: **Direct**
- *STAR-CCM+ Command Line Options*: Set command line options to pass to the design simulations, such as License Options or other STAR-CCM+ options. See Command-Line Reference.

  Example: `-mpidriver platform:"-aff:automatic:bandwidth"` (Platform MPI).

For more information, see Compute Resource Reference.

2. Select **File** > **Save** and save the project file.
3. Transfer the Design Manager project file (`*.dmprj`) and associated input files such as the reference simulation file (`*.sim`) and custom java macro files (`*.java`) to your cluster project root directory.
4. Prepare your cluster job submission script to meet the Pre-Allocation Submission Script Requirements. Make sure that the script file has execute permission.
5. From your cluster project root directory, submit the Design Manager project using the appropriate queue submission command for the cluster.
   Example: `qsub -l nodes=6:ppn=16 <submissionScript>.sh`, where `-l nodes=6:ppn=16` specifies the required resources for the simultaneous design simulations.

The job scheduling system on your cluster allocates the resources, such as 6 * 16 cores = 96 cores in the example above. Within the allocated resources, the Design Manager server starts on the first assigned compute node and acquires the resources to submit the design simulations one by one. Each design simulation requests the number of cores that you specify in the **Run Settings**.

The Design Manager server ensures that you do not overload the allocated resources. Suppose you specify 4 simultaneous design simulations and 32 compute processes for each. The Design Manager server runs the first 3 design simulations using 96 cores (3*32) and starts the 4th simulation as soon as one of the first three completes.

When submitting the Design Manager project, make sure that you do not allocated resources inefficiently. Suppose you specify 2 simultaneous design simulations and 32 compute processes to run with each design simulation, then 32 of the 96 allocated cores remain unused until the design study completes.

6. To follow the progress of your design studies, you can connect to and disconnect from the running Design Manager server, see Making Client-Server Connections.

Contents:

Pre-Allocation Submission Script Requirements
Running in an Interactive Session
Running on a Group of Machines

Pre-Allocation Submission Script Requirements

In pre-allocation mode, you submit the entire Design Manager project as a single job to the cluster queue.

Within the allocated resources, a *Resource Manager* provides the actual resource management functionality. Resource Manager keeps track of the design simulations that are launched as part of the job and the compute resources assigned to them. Resource Manager organizes the resources for the Design Manager project,

provides an interface for launching the design simulations, and makes sure that together with the design simulations the overall job does not exceed the allocated resources.

To submit a Design Manager project in pre-allocation mode, you must explicitly launch Resource Manager in the submission script using the following command:

```
starlaunch jobmanager --command <batch_command> --slots 0 --
batchsystem <system>
```

where:

- `starlaunch jobmanager` launches Resource Manager.
- `--command <batch_command>` specifies the `starccm+` command that runs the Design Manager project in batch.
- `--slots 0` prevents the Resource Manager process from allocating cores.
- `--batchsystem <system>` specifies the batch management system on your cluster.

For more information, see Job Manager Options.

Batch management systems are responsible for the allocation of jobs based on the requested resources. Currently the supported systems are PBS (`<system>=pbs`), LSF (`<system>=lsf`), Sun Grid Engine (`<system>=sge`), and Load Leveler (`<system>=ll`). For more information, see Using Batch Management Systems.

Example:

The examples scripts in this topic are written for tcsh (Tenex-C-Shell). For other Unix shells, such as bash, you must modify the script accordingly.

The following example shows a pre-allocation submission script for running a Design Manager project on a cluster using the batch management system PBS:

```
#!/bin/tcsh -f

PRDIR=[PROJECT_ROOT_DIR]            #[PROJECT_ROOT_DIR]: set absolute
path to your project root directory on the cluster
STARTCASE=[PROJECT_NAME].dmprj      #[PROJECT_NAME]: set name of your
Design Manager project file
CASELOG=output.log

STARCCMHOME=[STAR-CCM+_INSTALL_DIR] #[STAR-CCM+_INSTALL_DIR]: set
absolute path to the STAR-CCM+ installation directory on the cluster

# Call Job Manager
# ----------------------------------
$STARCCMHOME/star/bin/starlaunch jobmanager \
  --command "$STARCCMHOME/star/bin/starccm+ -rsh ssh -batch run
$PRDIR/$STARTCASE [-passtodesign <options>]" \
  --slots 0 \
```

```
  --batchsystem pbs \
  >> $CASELOG 2>&1
```

If your cluster uses the Sun Grid Engine batch management system and requires that you set the full path to `ssh`, you must specify this path for both `starccm+` and `starlaunch`:

```
#!/bin/tcsh -f

PRDIR=[PROJECT_ROOT_DIR]            #[PROJECT_ROOT_DIR]: set absolute
path to your project root directory on the cluster
STARTCASE=[PROJECT_NAME].dmprj      #[PROJECT_NAME]: set name of your
Design Manager project file
CASELOG=output.log

STARCCMHOME=[STAR-CCM+_INSTALL_DIR] #[STAR-CCM+_INSTALL_DIR]: set
absolute path to the STAR-CCM+ installation directory on the cluster

# Call Job Manager
# ----------------------------------
$STARCCMHOME/star/bin/starlaunch --rsh /usr/bin/ssh jobmanager \
  --command "$STARCCMHOME/star/bin/starccm+ -rsh /usr/bin/ssh -batch
run $PRDIR/$STARTCASE [-passtodesign <options>]" \ #
  --slots 0 \
  --batchsystem sge \
  >> $CASELOG 2>&1
```

`-passtodesign <options>` allows you to pass additional STAR-CCM+ simulation command line options, such as license options, to the design simulations. Passing command line options directly from the command line is useful for machine-specific or user-specific options that you do not want to save within your Design Manager project file (see Running on a Cluster Using Pre-Allocation, step 2b). However, command line options that require double quotes are not supported—you must set these options in the *STAR-CCM+ Execution Command* property. For more information, see Design Manager Options.

Note: In the pre-allocation submission script, you must set the absolute path to the project root directory.

Running in an Interactive Session

If your cluster management system allows interactive login sessions, you can run design studies in an interactive session.

An interactive session allows you to reserve a set of cluster nodes for interactive use. Interactive sessions are useful for debugging or testing Design Manager projects.

To run design studies in an interactive session:

1. On the cluster head node, request the resources for an interactive session.

Example: `qsub -l nodes=5:ppn=12 -I` (for a cluster using the cluster management system PBS).

For multiple nodes, ensure that the nodes share the same filesystem.

2. If the requested resources are available, the cluster management system allocates the specified resources and returns an interactive console on the first of the allocated nodes.
3. In the interactive console, launch Design Manager.

   In an interactive session, the Design Manager server must be associated with Resource Manager. For this reason, you must explicitly launch Resource Manager in the command line.

   Example:

```
[STAR-CCM+_INSTALL_DIR]/bin/starlaunch jobmanager
--command "[STAR-CCM+_INSTALL_DIR]/star/bin/starccm+ -rsh ssh -
server [PROJECT_ROOT_DIR]/[PROJECT].dmprj"
--slots 0 --batchsystem pbs
```

4. On your local machine, launch STAR-CCM+ and connect to the running Design Manager server, see Making Client-Server Connections.

   The Design Manager object tree appears.
5. Start the design studies interactively from the GUI.

   Design Manager runs the associated design simulations on the compute resources that are allocated to the interactive session.
6. After running the design studies save the project as required and exit the application.

Running on a Group of Machines

If you do not use a dedicated cluster with its own batch management system, you can submit a Design Manager project on a group of networked machines using the command line.

Note: Before you submit a Design Manager project on a group of machines, ensure that the machines share the same file system.

To submit a Design Manager project, you must explicitly launch Resource Manager using the following command:

```
starlaunch jobmanager --command <batch_command> --slots 0 --resources
<host[:np][,host[:np]]>
```

where:

`starlaunch jobmanager` launches Resource Manager.
`--command <batch_command>` specifies the `starccm+` command that runs the Design Manager project in batch.
`--slots 0` prevents the Resource Manager process from allocating cores.
`--resources <host[:np][,host[:np]]>` lists the hosts on which to run the Design Manager project.

If `:np` is specified, then `np` processes are put on that machine. There must be no spaces in the list of hosts. The processes are allocated to the machines in the resource list starting with the first entry adding `np`

processes before moving on to the next machine in the list. If the requested processes exceed the number of machines that are listed, the remaining processes continue to be allocated starting with the first entry in the resource list.

When multiple machines are being used it is useful to define the machines in an MPI file and use the following command:

```
starlaunch jobmanager --command <batch_command> --slots 0 --
resourcefile <file>
```

The format of the file is simply a list of machines on which you wish to place processes. The name of the machine can be concatenated with a colon and an integer to represent the number of processes you wish to place on the machine. An example of such a file could be:

```
node1
node2:3
node3:2
```

This file dictates that processes are allocated to machines node1, node2, and node3. 3 processes are allocated to node2, and 2 processes are allocated to node3.

Example:

The following example shows the command for running a Design Manager project on a group of machines using a machine file:

```
[STAR-CCM+_INSTALL_DIR]/star/bin/starlaunch jobmanager --command
"[STAR-CCM+_INSTALL_DIR]/star/bin/starccm+
-batch run [PROJECT_ROOT_DIR]/[PROJECT].dmprj [-passtodesign
<options>]" --slots 0 --resourcefile [MACHINE_FILE_DIR]/[MACHINES]
```

where:

`[STAR-CCM+_INSTALL_DIR]` is the absolute path to the STAR-CCM+ installation directory.
`[PROJECT_ROOT_DIR]` is the absolute path to your project root directory.
`[PROJECT]` is the name of your Design Manager project file.
`[MACHINE_FILE_DIR]` is the absolute path to your machine file.
`[MACHINES]` is your machine file.
`-passtodesign <options>` allows you to pass additional simulation command line options to the design simulations. Passing command line options directly from the command line is useful for machine-specific or user-specific options, which you do not want to save within your Design Manager project file (see Running on a Cluster Using Pre-Allocation, step 2b). However, command line options that require double quotes are not supported—you must set these options in the *STAR-CCM+ Execution Command* property. For more information, see Design Manager Options.

Re-Running a Design Study with Modified Settings

After running a design study you can re-run the design study with modified settings.

You can modify every property of a design study, such as the study type, the input parameters, the responses, the scenes and plots for post-processing, or the run and resource settings.

To re-run a design study with modified settings:

1. In the Design Manager project, right-click the respective design study node and select **Clear Study**.

   The result data are removed from the Design Manager project.
2. Clearing a design study does not remove the Design Manager output files from the disk.
   - If you want to remove the output files from the disk, navigate to the study artifact directory and delete it manually.
   - If you keep the previous output files, Design Manager uses subscripts for the new design artifact directories in your study artifact directory.
3. Modify the properties of the design study as required.
4. Select **File** > **Save** and save the project file.
5. Run the design study.

Stopping, Extending, and Resuming a Design Study

You can stop, extend, and resume a design study.

To stop a design study:

1. In the Design Manager project, right-click the respective design study node and select **Stop Study**.

   Stopping a design study prevents Design Manager from starting new design simulations. Design simulations that are already running are not killed but run until completion.

If you stopped a design study, you can resume it where it picks up from the last completed design:

2. Right-click the design study node and select **Run Study**.

   Design Manager submits the outstanding design simulations.

If a design study completes but further design simulations are needed to achieve the desired results, you can extend the design study:

3. Adapt the settings of the design study as required.
   Example: To extend a study, you can decrease the increment of the input parameter values or increase the number of designs to run.
4. Select **File** > **Save** and save the project file.
5. Run the design study.

   Design Manager automatically detects which designs are already evaluated and submits design simulations for the additional designs.

Resuming a Design Study after Abort

Design Manager can abort for reasons such as hardware conflicts or lack of system resources. When it does abort, you can resume the aborted design study from the last completed design.

You can also manually abort a running design study. In contrast to stopping a design study, where the running design simulations run until completion, aborting a design study kills the running design simulations. This is

particularly useful if you want to cancel a design study without having to wait until the running design simulations complete.

To manually abort a running design study from the UI:

1. Right-click the design study node and select **Abort Study**

To manually abort a running design study in batch mode:

2. Depending on your operating system, do one of the following:
   - On Windows, place a file with the file name `DM_ABORT` in the run directory; an empty text file is sufficient for this purpose.
   - On Linux, use the following line command:

     ```
     % touch DM_ABORT
     ```

   You can customize the abort file name in your Design Manager project, see Design Manager Reference.

Design Manager kills the running design simulations. The results of the completed designs are available in the study artifact directory.

To resume after abort:

3. Run the design study.

   Design Manager automatically detects the completed designs and runs the remaining design simulations until the design study completes.

Note: Design Manager can only resume from the last completed design, if the Design Manager project is automatically saved when each design simulation completes. For this reason, you are advised to always activate the *Auto Save* option (see Run Settings Reference).

Monitoring a Design Study

In the GUI, you can check the progress of a running design study in the progress and status bar beneath the explorer window. You can also view post-processing objects while a design study runs, allowing you to view progress interactively.

To track the status of all the design simulations of a design study in tabular form:

1. Select the **Design Sets** > **All** node and select **Open Output Table**.

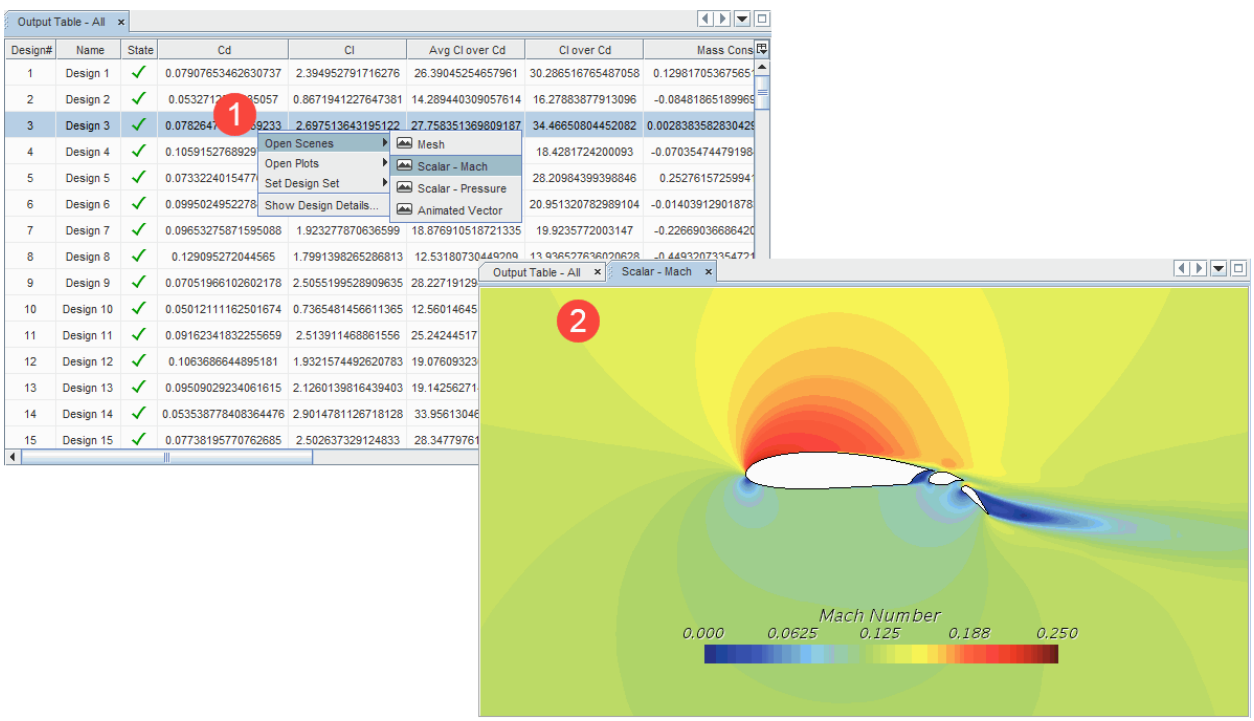
   The output table opens in the *Graphics* window and displays all designs of the design study together with their respective input parameter and response values. The **State** of a design indicates the current stage of the design simulation. The **Performance** value of a design is a rating value that allows you to judge the design. For more detail, see Output Table Reference.

As soon as a design simulation completes successfully, that is when **State** equals ✓, you can view the exported scenes and plots of that particular design:

2. In the output table, right-click the design of interest and select **Open Scenes** > **[scene]**, where [scene] is the scene you want to visualize. For plots, you select **Open Plots** > **[plot]**.

   The scene or plot opens in the *Graphics* window.

   Once you open a scene or a plot, a snapshot is created automatically. See Snapshots Reference.



If you prepared project plots before the run, you can monitor the progress of the design study using these plots:

3. In the object tree, right-click the respective **Plots** > **[project plot]** node and select **Open**.

The project plot opens in the *Graphics* window. Once a design simulation completes, a new data point appears in the plot.

Design Manager automatically updates the marker of the best design and the path to the best design during the run. For more information, see Plots Reference.

For more detailed information, see Analyzing the Results of a Design Study.

Analyzing the Results of a Design Study

In Design Manager, post-processing objects reference design sets. Design sets contain one or more designs, according to how you define them. You can view results in tables, scenes, plots, or project plots.

The following predefined design sets are available:

- All—contains all designs evaluated during the study.
- Successful—contains all designs apart from those that encountered an error.
- Error—contains the designs that encountered an error during the run.

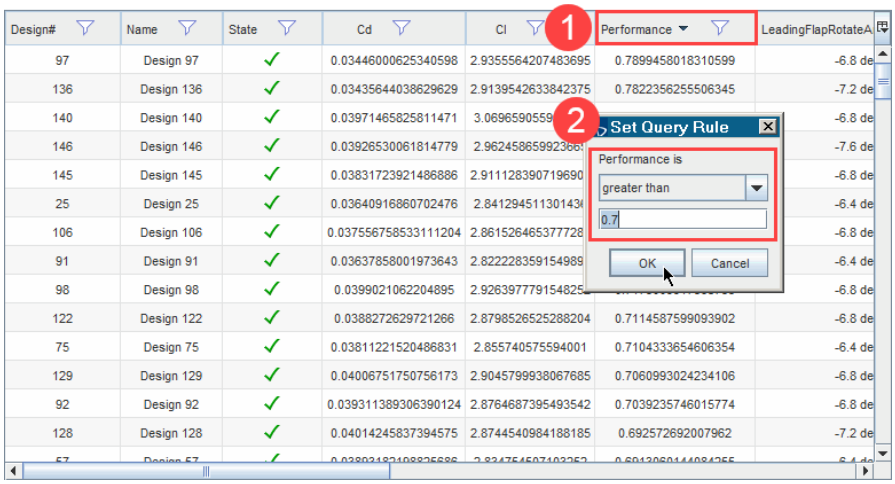
For more information, see Design Sets Reference.

To create additional design sets for a design study:

1. Right-click the **Design Studies** > **[design study]** > **Design Sets** node, and select **New Design Set**.

   In the *Graphics* window, the associated output table opens and displays all designs of the design study.
2. To define a subset of designs, you can set the designs manually or use the filter capabilities of the output table:

Method	Procedure
Setting the designs manually	1. In the project tree, select the associated design set node. 2. Set *Designs* to the designs of interest using the object selector.
Using the filter capabilities of the output table	1. In the output table header, next to the design property of interest, click . 2. In the *Set Query Rule* dialog, set the property to the desired range and click **OK**. For more information, see Output Table Filter Query Rules. The design set and its output table is updated to match the query rule.



Design Manager also allows you to create designs sets interactively from project plots or from output tables of other design sets. For more information, see Exploring the Analysis in Design Manager.

Contents:

Tabulating Designs
Visualizing Scenes and Plots
Creating Project Plots
Exploring the Analysis in Design Manager
Exporting the Results of a Design Study to HEEDS|post
Launching HEEDS|post

Tabulating Designs

In Design Manager, output tables allow you to tabulate the designs of a design set and their properties in the *Graphics* window. You can also export the designs of a design set to a `*.csv` table file.

To open the output table of a design set in the *Graphics* window:

1. Right-click the **Design Sets** > **[design set]** node and select **Open Output Table**.

   The output table appears in the *Graphics* window and displays the properties of each design—such as its parameter and response values. For more information, see Output Table Reference.

   Example:

Design#	Name	State	Cd	Cl	Avg Cl over Cd	Cl over Cd	Mass Cons	Performance	LeadingFlapRotateAngle	TrailingFlapRotateAngle	Transl
1	Design 1	✓	0.0790765346263073 7	2.394952791716276	26.39045254657961	30.286516765487058	0.12981705367565155 kg/s	2.4746196536362675E-8	0.0 deg	0.0 deg	
2	Design 2	✓	0.0525951862715568	0.8734923405068495	14.546952198473805	16.607838139347546	-0.10981904529035091 kg/s	-0.30039532247468087	-10.0 deg	-0.8000000000000007 deg	0.00879
3	Design 3	✓	0.0792453286875012	2.6885804040440595	28.120529781658973	33.927304593611176	0.12373624742031097 kg/s	0.1204681322506993	-4.4 deg	-4.4 deg	0.01600
4	Design 4	✓	0.0928577205346405 9	1.9436429474877346	18.286450932943122	20.931409217208365	-0.0563635379076004 kg/s	-0.362718589739328	-2.8 deg	4.4 deg	-0.00879
5	Design 5	✓	0.0734123871102674 4	2.07998058015124	24.285761044218795	28.33282858691206	0.2787340581417084 kg/s	-0.05988629896658104	0.8000000000000007 deg	10.0 deg	0.01280
6	Design 6	✓	0.1055702163072622	2.105333498932663	16.967871428002805	19.942494887053073	0.00109423696994781 5 kg/s	-0.45596746620823236	-0.8000000000000007 deg	0.8000000000000007 deg	0.00560
7	Design 7	✓	0.0920662876802787 2	1.8961323048929064	19.471108457177777	20.59529446302495	-0.1986854076385498 kg/s	-0.37254796885305996	6.399999999999999 deg	6.399999999999999 deg	-0.00560
8	Design 8	✓	0.0997586052273389 4	1.8016264227009375	12.676409687523925	18.059859784479023	-0.471043631434440 6 kg/s	-0.5092852708699838	4.4 deg	-2.8 deg	
9	Design 9	✓	0.0722216637384601	2.5000527477319965	26.941091025980718	34.616382651963846	0.30146050453186035 kg/s	0.1305704962547699	-8.0 deg	8.0 deg	
10	Design 10	✓	0.0490720388716377 74	0.7562536700375898	13.154375078986511	15.4110912737046	0.02471381798386573 8 kg/s	-0.3047940774017099	-6.4 deg	-8.0 deg	-0.00160
11	Design 11	✓	0.0865325775122332	2.5114733912602545	25.528802529266127	29.023443695586266	-0.02644813060760498 kg/s	-0.04563635018404888	2.8000000000000007 deg	-6.4 deg	
12	Design 12	✓	0.0927408437111152 4	1.9357131025374916	19.30245899896549	20.872282643526255	-0.15597060322761536 kg/s	-0.364551633467878	8.0 deg	2.8000000000000007 deg	
13	Design 13	✓	0.0806399038892015 9	2.5914775077969368	27.556304177432306	32.13641612665611	0.09593065083026886 kg/s	0.06228756105069943 6	-2.8 deg	-3.5999999999999996 deg	
14	Design 14	✓	0.0740661923046966	2.1981736283378366	25.11105436293256	29.678501890510287	0.37539660930633545 kg/s	-0.01880341509498417 7	-1.1999999999999993 deg	8.399999999999999 deg	0.01359
15	Design 15	✓	0.0449269654896322 1	0.8350783197741191	15.654468836188299	18.587463245583102	-0.05207767337560654 kg/s	-0.21946275758500505	-8.4 deg	-6.0 deg	0.00240

Design Manager allows you to reorder the designs in an output table. For example, to find the best design in a design set, you can order the designs according to their performance value.

To reorder the designs in an output table:

2. In the output table, click the header of the design property by which you want to reorder the table, such as **Performance**.

   The designs are reordered to display the property in ascending order. Clicking the header again reverses the order.

If an output table contains many columns, it can be useful to hide columns to improve the readability of the table.

To hide a column in an output table:

3. In the upper right corner of the output table, click .

   A drop-down menu opens that lists the headers of the columns that are displayed in the output table.
4. Deactivate the column that you want to hide.

For more information on interacting with designs in output tables, see Exploring the Analysis in Design Manager.

To export the design information of a dataset to a table file:

5. Right-click the **Design Sets** > **[design set]** node and select **Export to CSV...**.
6. In the *Export to CSV* dialog, save the design set as `*.csv`.

Visualizing Scenes and Plots

When each design simulation completes, Design Manager exports chosen scenes and plots and makes them available for you to view.

In Design Manager, a snapshot allows you to display a specific scene or a specific plot for the designs within a design set. Locking a snapshot to a specific design allows you to compare the design with other designs in the design set.

To create a snapshot:

1. Right-click the **Snapshots** node and select **New** > **[snapshot type]**, where [snapshot type] can be **Scene Snapshot** or **Plot Snapshot**.
2. Select the new snapshot node and set the following properties:
   - *Design Study*: The design study from which to extract the snapshot data.
   - *Design Set*: The design set within the specified design study from which to extract data.
   - *Scene*/*Plot*: The scene or plot of interest.

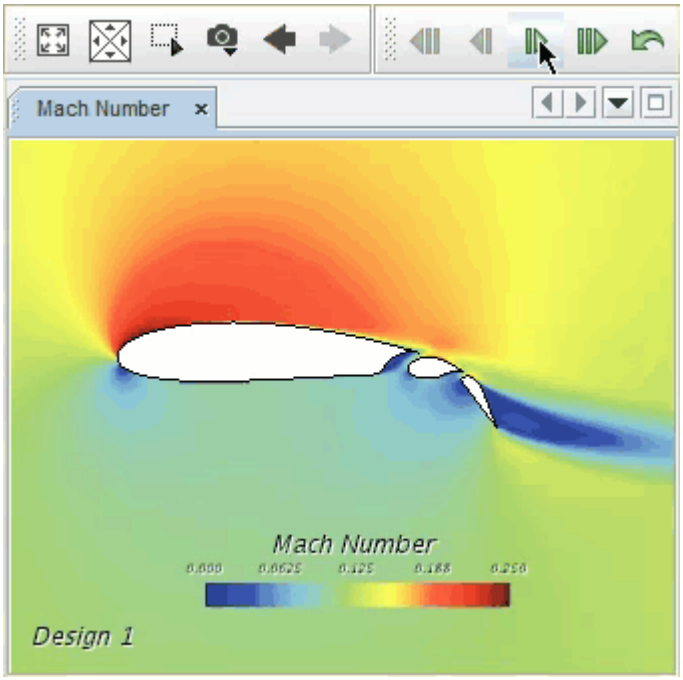
   Design displays the name of the design that is visualized in the specified scene or plot.

To add additional information to the snapshot display in the *Graphics* window, such as the name or the performance value of the visualized design:

3. Select the **Snapshots** > **[snapshot]** > **Annotations** node and set *Annotations* to **Design Name** or **Performance Value**, respectively.

   Design Manager allows you to create and add additional simple text annotations. For more information, see Tools Reference—Annotations.

You can step through the designs of the specified design set using the *Design Manager Snapshot* toolbar, see Toolbar Reference:

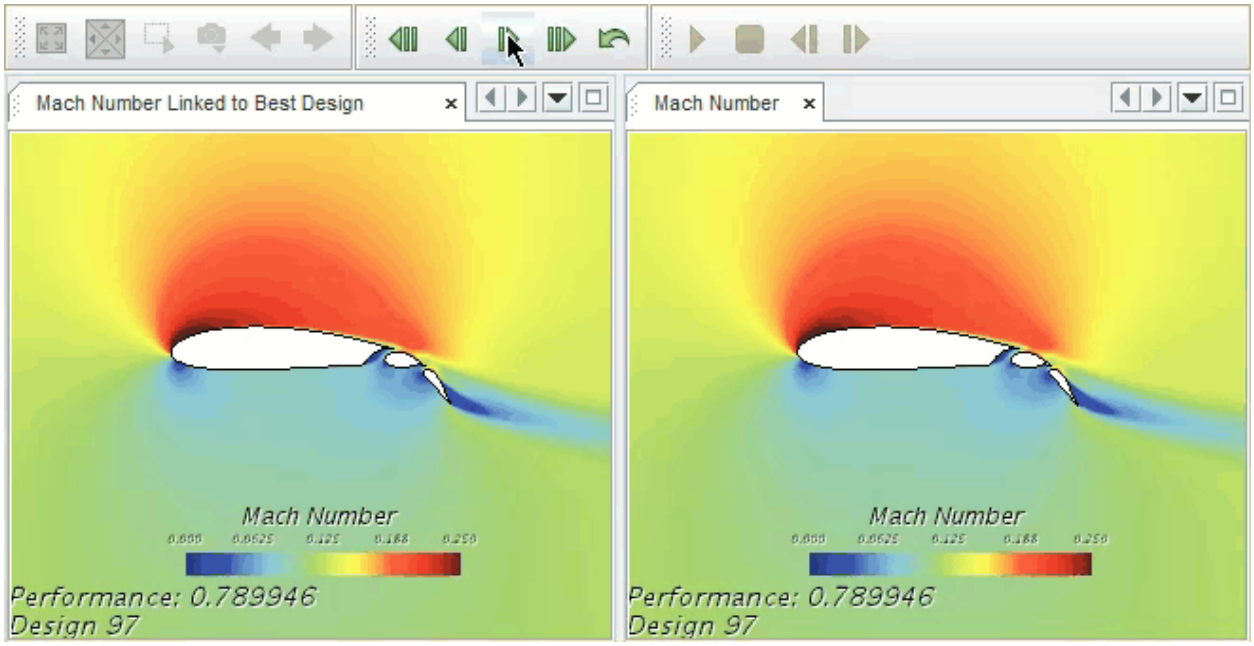


Alternatively, you can select a design of interest in an output table or in a project plot. The snapshot is then updated automatically to reflect the selected design. For more information, see Exploring the Analysis in Design Manager.

To compare snapshots for different designs:

4. Create multiple snapshots, where each snapshot displays the same scene or plot.
5. In the *Graphics* window, arrange the snapshot panels side by side.
6. For each snapshot, select the snapshot node and set the following properties:
   - *Lock Design*: Activate (activates the *Design* property).
   - *Design*: Set the design you want to compare.

   When you unlock one of the snapshots, you can step through the designs of the specified design set and compare the locked designs with the remaining ones.



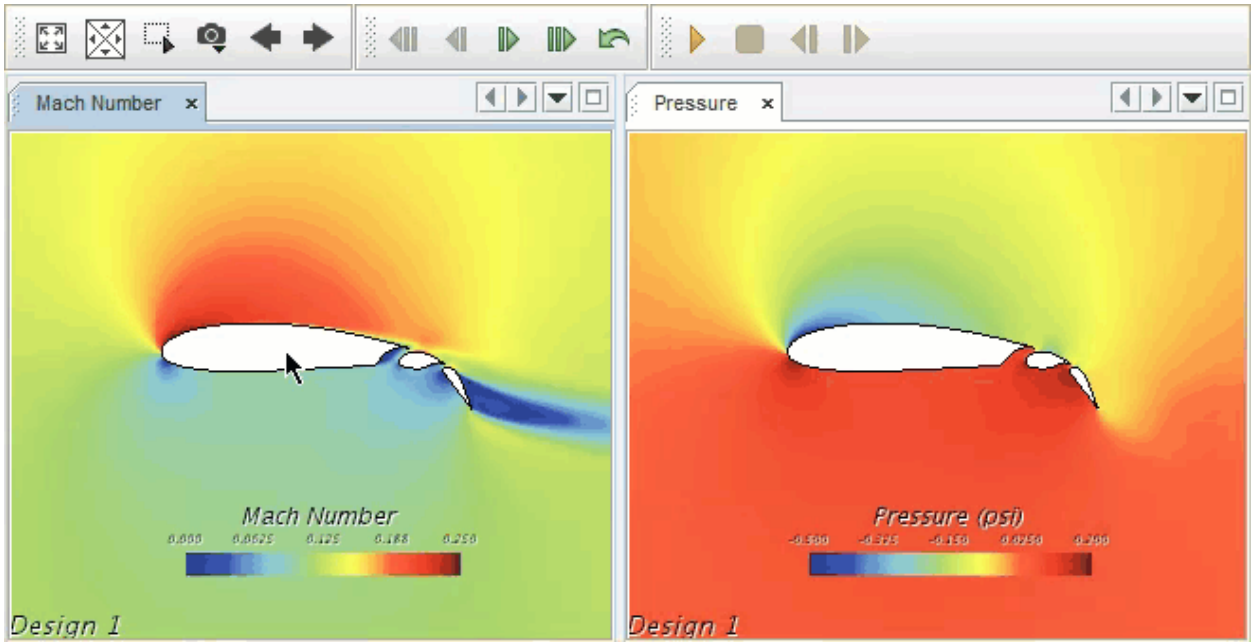
You can interact with snapshots that refer to scenes and plots that are exported as **STAR-View File** (see Setting Up a Design Study) as you would with the respective scene or plot in STAR-CCM+. For example, you can rotate the view, zoom in and out, or play animations. To improve understanding of results, you can link multiple snapshots so that they move in tandem as you interact with one of them.

To set up a link among snapshots:

7. Using `<Ctrl>`, multi-select the snapshots you wish to link.
8. Right-click and select **Linked View**.

   A chain-link icon appears at the bottom right corner of the snapshot node icons.
9. Interact with one of the snapshots.

   The linked snapshots are synchronized automatically.



To export a snapshot to an image file:

10. Right-click the snapshot node and select **Hardcopy**. Using `<Ctrl>`, you can select and export multiples snapshots.

    A customized, operating system-specific *Save To File* dialog appears, which allows you to write the selected snapshot to file. See Saving Scene Image Files. This dialog also appears if you right-click the background of a snapshot in the *Graphics* window, from which a single image file of the current snapshot can be saved.

Creating Project Plots

Project plots are two-dimensional graphs whose data spans all designs of a design set. These plots can show history, 2D relations, or Pareto analysis.

To create a project plot:

1. Right-click the **Plots** node and select **New Plot**.

To add data to the plot:

2. Right-click the **Plots** > **[project plot]** > **Data Series** node and select **Add Data**.
3. In the *Add Data Providers to Plot* dialog, select the design study from which to extract data and click **OK**.

   A new data series, which is named after the selected design study, is added to the **Data Series** node. If you select multiple design studies, multiple data series are created automatically.
4. Select the **[project plot]** > **Data Series** > **[design study]** node and set *Input Design Sets* to the design sets whose data you want to display in the plot.

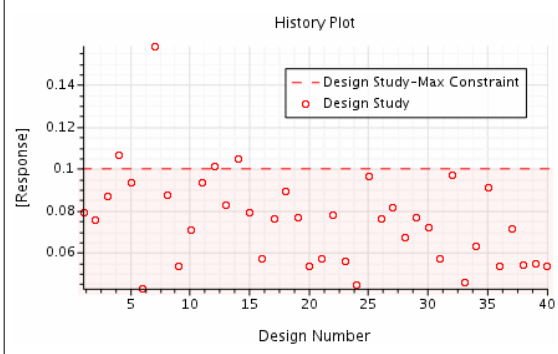
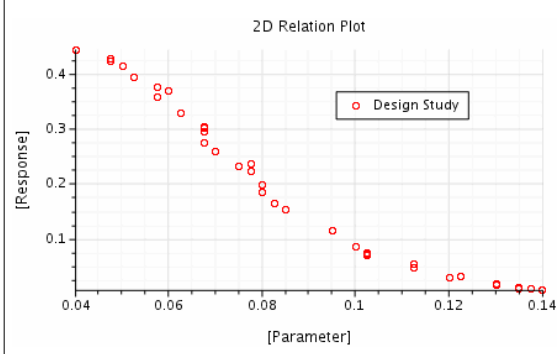
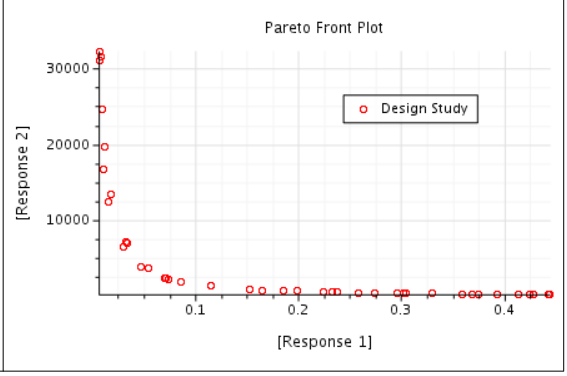
To set the type of data that you want to display on the plot axes:

5. Select the **[project plot]** > **Data Series** > **[design study]** > **Bottom Axis Data** node and set *Type* to one of the following:

- **Parameter**: Allows you to display an input parameter of the design study, which you specify in the **Parameter** child node.
- **Response**: Allows you to show a response of the design study, which you specify in the **Response** child node. When you activate the *Show Constraints* property of the specified response, the plot highlights the field of feasible designs, where the field is determined by the constraints that you set for the response.
- **History** : Displays the design number.

6. Repeat the last step for the **Left Axis Data** node.

Examples: The following table lists the required axis data types for plots that are commonly used in the analysis of design studies:

Plot	**Bottom Axis Data** Type	**Left Axis Data** Type	Example
History Plot A history plot shows a parameter or a response against the design number.	**History**	**Parameter / Response**	
2D Relation Plot A 2D relation plot shows a response against a parameter.	**Parameter**	**Response**	
Pareto Plot A pareto plot shows one response against another response. This plot is particularly useful for multi-objective optimization, where you want to display the tradeoff between two competing objectives.	**Response**	**Response**	

Plot	**Bottom Axis Data** Type	**Left Axis Data** Type	Example

Alternatively, you can create a **History Plot** directly from the response or parameter of interest:

7. Right-click the respective response or parameter node and select **Create History Plot**.

   The plot opens in the *Graphics* window.

Design Manager allows you to mark specific designs in a study plot.

To mark the data point of the best design, where the best design is the design with the highest performance value:

8. Select the **[project plot]** > **Data Series** > **[design study]** > **Best Design** node and activate *Visible*.

   The respective data point is marked in the plot and is listed in the legend.

In a history plot, you can display the path that the design study follows in reaching the best design:

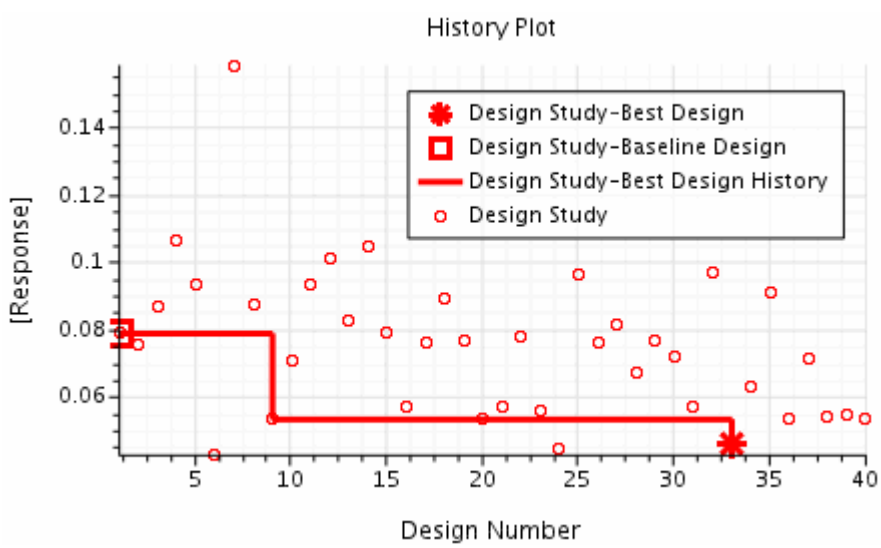
9. Select the **[project plot]** > **Data Series** > **[design study]** > **Best Design History** node and activate *Visible*.

   A line displays the path from the first to the best design.

To mark the data point of the baseline design, where the baseline design is the design of the reference simulation:

10. Select the **[project plot]** > **Data Series** > **[design study]** > **Baseline Design** node and activate *Visible*.

The following screenshot displays a history plot, where the best design, the best design history, and the baseline design are visible:



To determine the value of a particular data point:

11. Hover the mouse over the data point of interest.

    The coordinates are displayed next to the pointer.

You can export the data series that are displayed in a study plot:

12. To export data series, do one of the following:
    - To export all data series in a plot, right-click the plot node and select **Export**.
    - To export specific data series, multi-select the respective data series nodes, right-click, and select **Export**.
13. In the *Export to CSV* dialog, save the design set as `*.csv`.

For more information, see Plots Reference.

Contents:

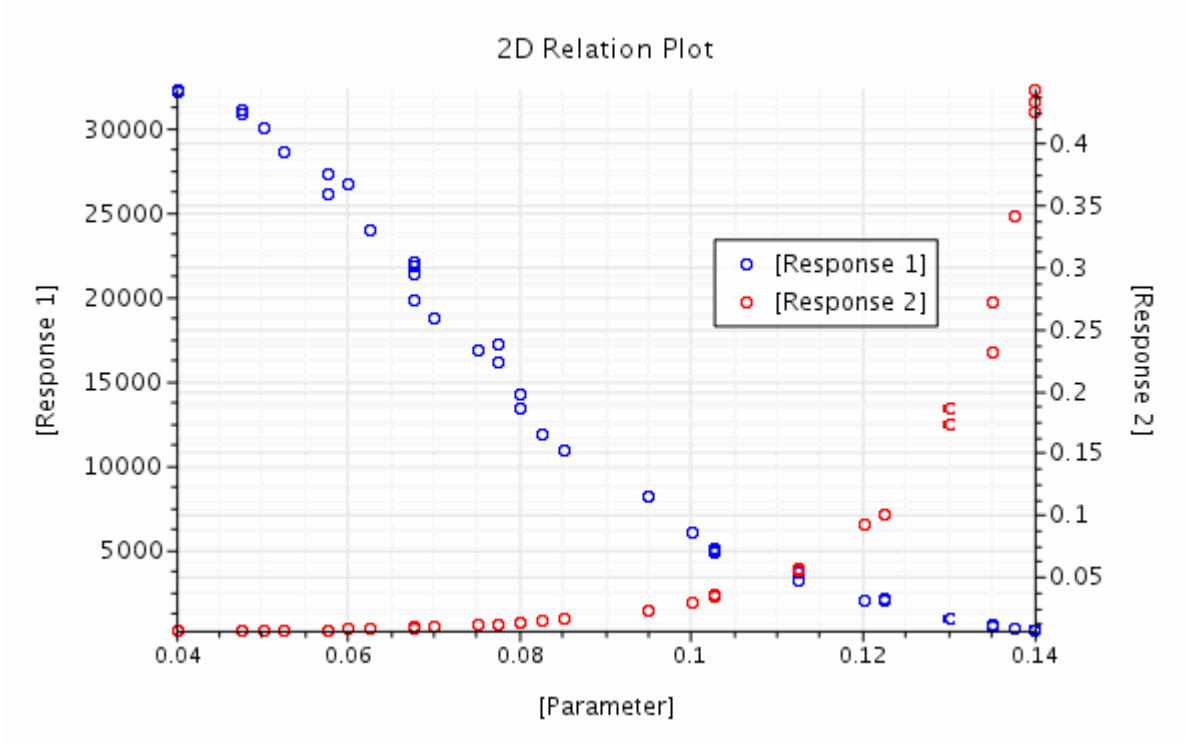
Plotting Multiple Data Series

Plotting Multiple Data Series

Design Manager allows you to add multiple data series to a single project plot. These data series can refer to the same design study or to different design studies within your Design Manager project.

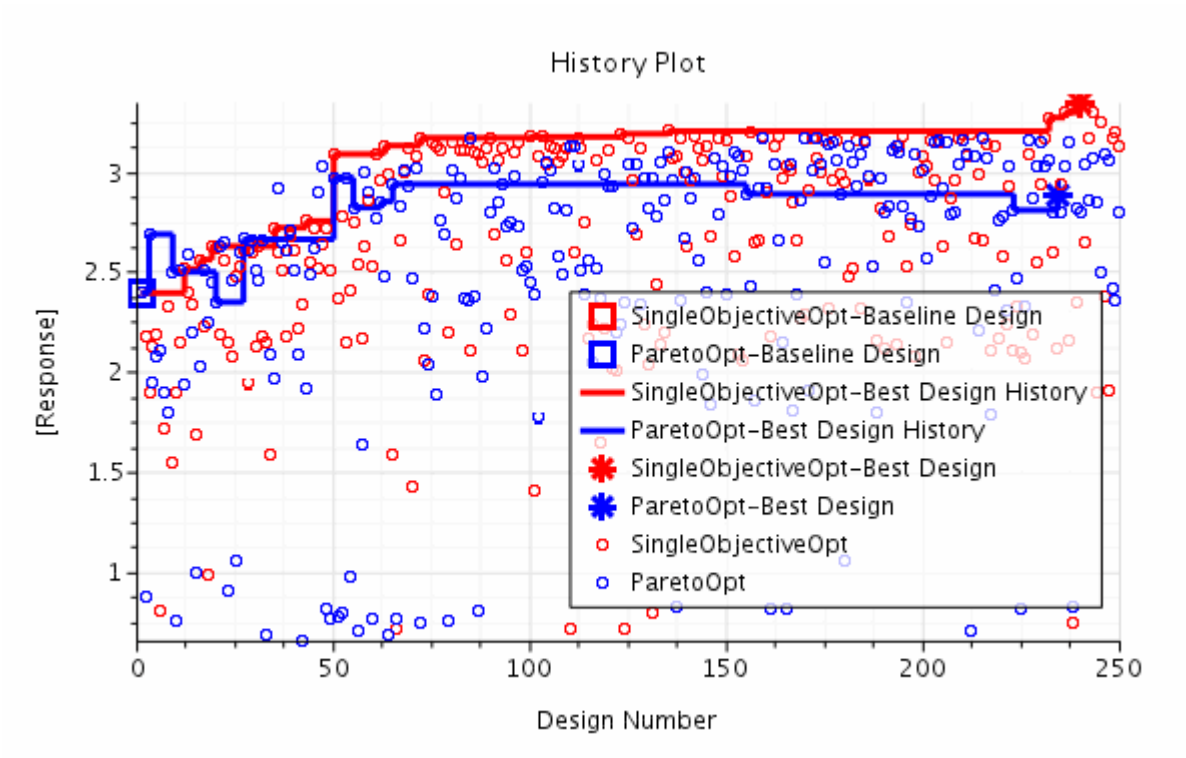
Referring to the same design study is useful when you want to display multiple variables of a design study—responses, parameters, or design numbers—in a single chart display. For example, you can view two different responses versus a parameter in the same plot. Viewing the variables on different axes improves the readability of such plots, especially when the variables have widely different scales of values, such as meters and Pascals.

Example:



Referring to various design studies is useful mainly for optimization studies. Suppose you run a weighted sum of multiple objectives optimization study. By looking at the responses it appears that there is possibly a competitive relationship between them. To explore this possibility, you run a Pareto optimization study. Showing the data from both of these studies on the same plot can indicate the part of the Pareto front on which the weighted sum study was focused.

Example:



To plot multiple variables from the same or from different design studies:

1. Create a project plot.
2. For each pair of variables that you want to display for a certain design set:
   a) Add a data series and select the design study and design set from which you want to extract the variables.
   b) Set the axes data that you want to display, such as parameters and responses, for the default axes—at the bottom and at the left.

   To use multiple axes:
3. Right-click the **[project plot]** > **Axes** node and select **New Axis** > **[position]**, where **[position]** can be **Right** or **Top**.
4. For the variable that you want to display on the new axis, select the respective **[data series]** > **[axis data]** node, where **[axis data]** can be **Left Axis Data** or **Bottom Axis Data**, and set *Axis* to **Right Axis** or **Top Axis**, respectively.

For more information, see Plots Reference.

Exploring the Analysis in Design Manager

Follow these steps to assess the design connectivity between output tables, snapshots, and project plots.

In Design Manager, output tables, snapshots, and project plots are connected to each other. When you select a specific design in one of these post-processing objects the others are synchronized automatically to reflect the selected design.

To visualize the design connectivity:

1. Arrange multiple panels in the *Graphics* window. A panel is an individual section of the *Graphics* window that contains an output table, a project plot, or a snapshot (scene or plot). You can save a window layout to make the current *Graphics* display available for restore. For more information, see Preserving the Window Layout.
2. Using the *Design Manager Snapshot* toolbar, step through the designs one by one. See Toolbars Reference.
3. In an output table or in a project plot, click a specific design item:
   - In an output table, select the respective row.
   - In a project plot, select the respective data point.

   The other panels are updated automatically to reflect the selected design.

Output tables and project plots provide further features for interacting with a selected design item. You can open scene and plot snapshots, define design sets, or display design details.

To open a scene or plot snapshot from a selected design item (row in a table or point in a project plot):

4. Right-click the design item and select **Open Scenes/Open Plots** > **[scene/plot]** > **[snapshot]**, where [snapshot] can be a new snapshot or a previously created snapshots for the respective scene or plot.

To create a new design set from specific designs or to add specific designs to a previously created design set:

5. Using `<Ctrl>`, multi-select the designs of interest.
6. Right-click and select **Set Design Set** > **[design set]**, where [design set] can be a new design set or a previously created design set.

To display design details:

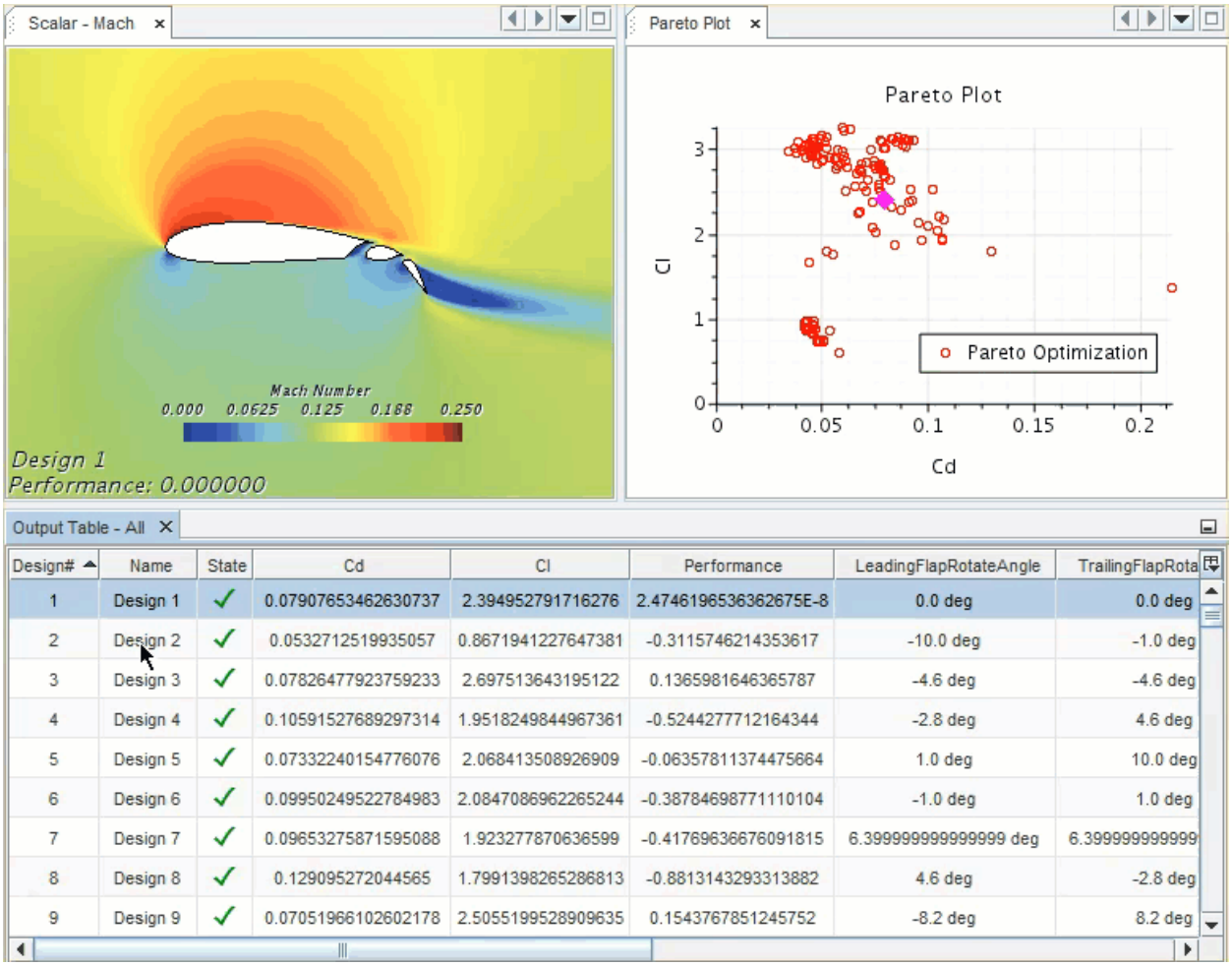
7. Right-click and select **Show Design Details...**.

   The *Design details* dialog displays the following information:

   - Design
   - Design status
   - Performance value
   - Design artifact directory
   - Messages

   For more information, see Output Table Right-Click Actions and Plots Right-Click Actions.

   Example: The following video displays the connectivity between output tables, project plots, and snapshots for an airfoil pareto optimization study.



Design#	Name	State	Cd	Cl	Performance	LeadingFlapRotateAngle	TrailingFlapRota
1	Design 1	✓	0.07907653462630737	2.394952791716276	2.4746196536362675E-8	0.0 deg	0.0 deg
2	Design 2	✓	0.0532712519935057	0.8671941227647381	-0.3115746214353617	-10.0 deg	-1.0 deg
3	Design 3	✓	0.07826477923759233	2.697513643195122	0.1365981646365787	-4.6 deg	-4.6 deg
4	Design 4	✓	0.10591527689297314	1.9518249844967361	-0.5244277712164344	-2.8 deg	4.6 deg
5	Design 5	✓	0.07332240154776076	2.068413508926909	-0.06357811374475664	1.0 deg	10.0 deg
6	Design 6	✓	0.09950249522784983	2.0847086962265244	-0.38784698771110104	-1.0 deg	1.0 deg
7	Design 7	✓	0.09653275871595088	1.923277870636599	-0.41769636676091815	6.399999999999999 deg	6.399999999999
8	Design 8	✓	0.129095272044565	1.7991398265286813	-0.8813143293313882	4.6 deg	-2.8 deg
9	Design 9	✓	0.07051966102602178	2.5055199528909635	0.1543767851245752	-8.2 deg	8.2 deg

Exporting the Results of a Design Study to HEEDS|post

Once a design study is complete, you can export the study results to HEEDS|post. HEEDS|post is a plotting and graphing package that offers enhanced post-processing capabilities for design exploration studies.

Note: Whenever possible, perform the post-processing for your design studies in Design Manager. Only use HEEDS|post if the post-processing capabilities of Design Manager do not suit your requirements. HEEDS|post requires a STAR-Innovate license.

To export the results of a design study as a HEEDS|post project:

1. In Design Manager, right-click the design study node and select **Export to HEEDS|post...**.
2. In the *Export to HEEDS|post* dialog:
   a) To launch HEEDS|post automatically after exporting the data, activate the **Launch HEEDS|post** option.

      If the Design Manager server is running on a remote server—that is, if you are running STAR-CCM+ locally and connecting to a remote Design Manager server, the **Launch HEEDS|post** option is grayed out. You can launch HEEDS|post at a later time and load the exported HEEDS|post project manually. See Launching HEEDS|post Manually.

b) Enter the output *File name*.

c) Click **Save**.

Design Manager writes the HEEDS|post project file `[File name]0.rpt`. Depending on the study type of your design study, Design Manager writes additional output files, such as:

- `[File name]0.res`
- `[File name]0.gph`
- `[File name]0.hst`
- `[File name]0.plot`

Additionally, Design Manager creates a `[File name]0_POST` folder, which contains copies of the available scene, plot, and log files. The structure of this folder duplicates the structure of your project artifact directory.

Note: HEEDS|post cannot view `*.sce` files. For this reason, only scenes and plots that are written as `*.png` are copied to the `[File name]0_POST` folder.

Launching HEEDS|post

You can launch HEEDS|post from Design Manager and analyse the exported results of a design study in HEEDS|post.

To launch HEEDS|post manually and open an exported HEEDS|post project:

1. In Design Manager, select **Tools** > **Launch HEEDS|post**.
2. In HEEDS|post, select **File** > **Open**.
3. In the *HEEDS|post - Open Project* dialog, set the file type to **HEEDS POST Project (*.rpt)**.
4. Navigate to and select the exported `[File name]0.rpt` file, and click **OK**.

Design Manager Reference

The following properties and right-click actions are available for a Design Manager project.

Design Manager Project Properties

The following properties are available for the Design Manager project node at the root of the object tree.

Collaboration

When **On**, allows other users to connect to the server running the Design Manager project.

Name

Displays the name of the Design Manager project (read only).

Project Artifact Directory

Specifies the path to the project artifact directory, which contains all the outputs of the design studies—such as simulation files, log files, scene files, and plot files. For more information, see directory file structure.

By default, Design Manager uses the relative path to the directory that contains the `*.dmprj` file.

Study Execution Abort File Name

Allows you to specify the name of a file that, once placed in the project root directory, causes the design studies in your Design Manager project to abort. This action is useful for stopping a batch run. The default file name is `DM_ABORT`.

Design Manager Project Right-Click Actions

The following specific actions are available when you right-click the Design Manager project node. For information on the common right-click actions, see Common Pop-Up Menu Items.

View Log

Allows you to view the log file of the Design Manager project.

Whenever you save the Design Manager project, an entry is added to a project log. The project log is a record of each save operation through the life of the project, which specifically records the following for every save operation:

- Date
- Time
- Version number
- Architecture that was used.

The information is stored in the project file itself, and not in a separate file. The log could be useful if you are checking compatibility issues between architectures, or between greatly disparate version numbers, for example.

You can also intersperse a log with comments of your own, see Adding Comments.

Collapse Tree

Shrinks the object tree back to its default size.

Contents:

Simulations Reference
Design Studies Reference
Snapshots Reference
Plots Reference
Tools Reference
Toolbars Reference
Command Line Options Reference

Simulations Reference

The following properties and right-click actions are available for reference simulations in Design Manager.

Simulations Properties

Object	Properties
Simulations	*Children* Displays the number of reference simulations in the Design Manager project.
⌊ **[simulation]**	*Simulation File* Specifies the path to the STAR-CCM+ simulation file (`*.sim`). By default, Design Manager uses the relative path to the project root directory.
⌊ **Parameters**	*Children* Displays the number of parameters, reports, scenes, and plots in the reference simulation, respectively.
⌊ **Reports**	
⌊ **Scenes**	
⌊ **Plots**	

Simulations Right-Click Actions

Object	Right-Click Actions
Simulations	**Read Reference Simulation...** Reads the parameters, reports, scenes, and plots from the reference simulation into the Design Manager project.
⌊ **[simulation]**	**Load...** Opens the reference simulation in STAR-CCM+. **Update** Updates the parameters, reports, scenes, and plots to reflect the latest changes to the reference simulation. Use this action if your Design Manager project is out of date and you encounter a problem regarding a parameter, report, scene, or plot update.

Design Studies Reference

The following properties and right-click actions are available for design studies in Design Manager.

Design Study Properties

Object	Properties
Design Studies	*Design Studies* Displays the number of design studies in the Design Manager project (read-only).

└ [design study]	*Study Type*
	Controls the type of design study that you want Design Manager to run. For information on the various study types, see Study Types.
	• **Manual**
	Activates the **Design Table** node that allows you to create the designs to run in tabular form. See Creating Manual Designs.
	• **Sweep**
	Activates the **Design Table** node. Edit this node to display the list of designs that are created based on the **Input Parameters** settings. The number of designs to run is determined by the number of variables and how many values are used for each variable.
	• **DOE**
	Activates the **DOE Settings** node that allows you to set the *DOE Type* to run:
	  • **2 Level Full Factorial**: Carries out a two-level full factorial DOE study.
	  • **3 Level Full Factorial**: Carries out a three-level full factorial DOE study.
	• **Optimization**
	Activates the **Optimization Settings** and **Design Seeds** nodes.
	The **Optimization Settings** node controls the following properties:
	  • *Optimization Type*: Specifies the type of optimization to run.
	  Weighted sum of all objectives specifies a design study where all objectives are combined into a single objective, which is optimized by the algorithm. The objectives are combined using the performance function given by Eqn. (1376).
	  Multiple objective tradeoff study (Pareto front) specifies a design study where each objective is independently considered by the algorithm.
	  This option activates the **Archive Size** child node. The archive size is the number of non-dominated designs that you want the algorithm to find during the search in order to resolve the Pareto front.
	  During the run of a Pareto optimization study, Design Manager runs several cycles, where each cycle runs a number of designs that is equal to the archive

Object	Properties
	size. After each cycle, Design Manager ranks the designs in order to find all the designs that satisfy the non-dominated design condition. This procedure continues until the specified number of designs to run is met. The value of the archive size must be evenly divisible by four and greater than or equal to 12. This value should also be greater than or equal to the number of input parameters in the design study. As an example, a Pareto optimization study with 8 variables would have an archive size of 12, while one with 13 variables would have an archive size of 16. Single-objective and multi-objective studies use the SHERPA algorithm. • *Designs to Run*: Specifies the total number of designs to run in the design study. For single-objective Optimization studies, run at least 30 to 35 designs. As a guideline, set this value to 10 times the number of input parameters in your design study. For multi-objective Optimization studies, run at least 40 designs, which is the default value. As a guideline, for multi-objective Optimization studies, set this value to 8 to 12 times the archive size. If the archive size is larger than 20, the number of designs to run is multiplied by 2 to give 40 designs. The SHERPA algorithm works best when it can learn as many times as possible. The **Design Seeds** node controls seed values for the optimization study. You can seed an optimization study with manually entered designs or with design sets of a prior design study. See Design Seeds Reference. • **Stochastic** Activates the **Stochastic Settings** node that allows you to set the total number of *Designs to Run* in the design study. *Simulations* Specifies the STAR-CCM+ reference simulation that provides the inputs for the design study. *Tags* Specifies the tags for the design study. For more information, see Tagging Simulation Objects.

Design Studies Right-Click Actions

Object	Right-Click Actions
Design Studies	**New** Creates a new design study. **Run All Studies** Runs all design studies sequentially from first to last.

Object	Right-Click Actions
	Stop All Studies Stops all design studies. **Abort All Studies** Aborts all design studies. **Clear All Studies** Clears the designs and study results from all design studies.
└ [design study]	**Run Study** Runs the selected design study. **Stop Study** Stops the selected design study. Stopping a design study does not kill the STAR-CCM+ jobs that are already running. **Abort Study** Aborts the selected design study. Aborting a design study kills the STAR-CCM+ jobs that are already running. **Clear Study** Removes all designs and study results from the selected design study. **Create Plot > [Plot Type]** Creates a new project plot. The following plot types are available: • **History Plot** • **Parameter vs Response Plot** • **Response vs Response Plot** For more information, see Creating Project Plots. **Open All Plots** Opens all project plots in the *Graphics* window. **Export to HEEDS Post...** Writes the results of the selected design study to the following post-processing files that can be loaded into HEEDS Post: • `star0.gph` • `star0.hst` • `star0.res` • `star0.rpt` See Exporting the Results of a Design Study to HEEDS Post..

Object	Right-Click Actions
	Set New Tag... Creates a new tag and adds it to the selected design study. For more information, see Tagging Simulation Objects. **Set Tags > [Tag]** Allows you to set a new tag or a previously created tag to the selected design study.

Contents:

Input Parameters Reference
Responses Reference
Scenes and Plots Reference
Settings Reference
Design Seeds Reference
Design Sets Reference

Input Parameters Reference

The following properties and right-click actions are available for input parameters.

Input Parameters Properties

Object	Properties
Input Parameters	*Parameters* Specifies the parameters from the reference simulation to use as input parameters for the design study.
∟ **[input parameter]**	The properties that are available for an input parameter depend on study type, see table below.

Properties	Manual	Sweep	DOE	Optimization	Stochastic
Baseline Value Specifies the value of the input parameter in the baseline design. The baseline design is the first design that Design Manager runs in the design study.					✔

Properties	Manual	Sweep	DOE	Optimization	Stochastic
Distribution Specifies the type of probability distribution to use for the input parameter. The probability distribution is centered around the *Baseline Value* of the input parameter. • **Normal**: Specifies a normal distribution. The standard deviation is specified through the *St. Deviation* property in the corresponding child node. • **Triangular**: Specifies a Triangular distribution. The lower and upper limits of the distribution are specified through the *Min Variation* and *Max Variation* properties in the corresponding child node, respectively. • **Uniform**: Specifies a uniform distribution. The maximum variation in either direction is specified through the *Max Variation* property in the corresponding child node.					✔
Factor Mode Controls whether the lower and upper values for the input parameter are absolute or relative to a mid value. • **Absolute** Activates the **Absolute** child node, where you set absolute values for the lower value (*Low*), mid value (*Mid*), and upper value (*High*). The *Mid* property is only available for 3 Level Full Factorial DOE studies. • **Relative** Activates the **Relative** child node, where you set a constant mid value (*Mid*) and relative low and high values (*Low(-%)* and *High(+%)*). The low and high values are relative to the specified mid value.			✔		
Lock Name When **On**, preserves the name of the selected input parameter. When **Off**, the name of the input parameter is linked to the name of the associated parameter from the reference simulation. When you change the name of the parameter in the reference simulation and update the reference simulation in Design Manager, the input parameter name is updated automatically.	✔	✔	✔	✔	✔
Mode Specifies whether the distribution properties (*Min Variation*, *Max Variation,* and *St. Deviation*) are absolute, or relative to the baseline value.					✔

Properties	Manual	Sweep	DOE	Optimization	Stochastic
• **Relative** All property values are specified as a percentage of the *Baseline Value*. • **Absolute** All property values are specified as absolute values.					

Parameter Type
Controls how Design Manager sets the input parameter values.
• **Continuous**
Allows Design Manager to set the input parameter to values within a specified range. This option activates the **Continuous** child node, which controls the following properties:
  • *Min/Max Range*: Specifies the minimum value and the maximum value that Design Manager can set for the input parameter.
  • *Baseline Value*: Specifies the value of the input parameter in the baseline design. The baseline design is the first design that Design Manager runs in the design study.
  • *Increment*: Sets the increment size. When set, *Resolution* is automatically updated.
  • *Resolution*: Sets the number of different values to look at for the input parameter. When set, *Increment* is automatically updated.
• **Discrete**
Sets the input parameter to discrete values. This option activates the **Discrete** child node, where you specify the *List* of values, separated with commas or spaces, and the *Baseline Value* of the parameter.
• **Constant**
Sets the input parameter to a constant value, without exploring other values. This option activates the **Constant** child node, where you set the *Value* of the constant.
		✔		✔	

Reference Value Displays the value of the associated parameter in the reference simulation.	✔	✔		✔	
Tags Specifies the tags for the input parameter. For more information, see Tagging Simulation Objects.	✔	✔	✔	✔	✔

Input Parameters Right-Click Actions

Object	Right-Click Actions
Input Parameters	**Edit...** Allows you to create and edit the properties of the input parameters in tabular form.
∟ **[input parameter]**	**Create History Plot** Creates a project plot that displays the selected input parameter against the design number. **Relink Parameter...** Allows you to set the selected input parameter to another parameter from the reference simulation. This option is particularly useful if changes in the reference simulation break the link to the associated parameter. **Set New Tag...** Creates a new tag and adds it to the selected input parameter. For more information, see Tagging Simulation Objects. **Set Tags > [Tag]** Allows you to set a new tag or a previously created tag to the selected input parameter.

Responses Reference

The following properties and right-click actions are available for responses.

Responses Properties

Object	Properties
Responses	*Response* Sets the responses for the design study. See Study Outputs—Responses.
∟ **[response]**	*Is Objective* **(only for Optimization and Sweep studies)** When **On**, instructs Design Manager to use the response as an objective for the design study. Activating this property activates the **Objective Properties** child node. In STAR-CCM+ 12.04, this property was supported for all study types. From version 12.06 onwards, if you play a macro recorded in version 12.04 that activates the **Is Objective** property for a now unsupported study type, the read-only property is activated but no longer used for the performance evaluation.

Object	Properties
	Is Constraint When **On**, instructs Design Manager to use the response as a constraint for the design study. Activating this property activates the **Constraint Properties** child node. *Units* Displays the unit of the selected response. *Lock Name* When **On**, preserves the name of the selected response. When **Off**, the name of the response is linked to the name of the associated report from the reference simulation. When you change the name of the report in the reference simulation and update the reference simulation in Design Manager, the response name is updated automatically. *Tags* Specifies the tags for the response. For more information, see Tagging Simulation Objects. *Specify Valid Minimum* When **On**, allows you to set a valid minimum for the response value. A design with a response value below the value that you specify in the corresponding **Valid Minimum** child node is classed as Error design. *Specify Valid Maximum* When **On**, allows you to set a valid maximum for the response value. A design with a response value above the value that you specify in the corresponding **Valid Maximum** child node is classed as Error design.
└ Objective Properties	*Goal* Sets the analysis goal for the selected response. You can choose to **Minimize** or **Maximize** the value of the response. The specified goal sets the $Sign_i$ in the performance evaluation of each design (see Eqn. (1376)). The value is -1 for objectives being minimized and +1 for objectives being maximized. *Normalization Type* Sets the type of normalization to use for the selected response. The specified value determines the normalization value $Norm_i$ in the performance evaluation of each design (see Eqn. (1376)).

Object	Properties
	• **Baseline Normalization**: Sets the normalization value automatically to the objective's value in the baseline design. The baseline design is the first design that Design Manager runs in the design study. For the various study types, this design is determined as follows: ◦ Manual: The first design that you specify in the list of designs to run. ◦ Sweep/DOE: The design that combines the minimum values for all of your input parameters. ◦ Optimization/Stochastic: The design that combines the values that you set as *Baseline Value* for each input parameter. • **Normalization Value**: Allows you to specify the normalization in the **Normalization Value** child node. Only use this property if there are multiple objectives. Set this property so that all normalized objectives are of a similar order of magnitude. *Weight* Allows you to take into account the difference in importance between objectives. For example, the mass of a cooling device can be of less importance than its cooling rate. The specified value determines the linear weight of the objective $LinWt_i$ in the performance evaluation of each design (see Eqn. (1376)).
∟ **Constraint Properties**	*Type* Sets the type of the constraint. You can set a **Minimum** or a **Maximum** constraint for the response. You can also combine both limits (**Minimum/ Maximum**). Any designs with a response value outside of the range that you specify in the corresponding child node is classed as infeasible. You specify the constraint values in the corresponding child nodes.

Responses Right-Click Actions

Object	Right-Click Actions
Responses	None.
∟ **[response]**	**Create History Plot** Creates a project plot that displays the selected response against the design number. **Relink Response...** Allows you to set the selected response to another report from the reference simulation. This option is particularly useful if changes in the reference simulation break the link to the associated report.

Object	Right-Click Actions
	Set New Tag... Creates a new tag and adds it to the selected response. For more information, see Tagging Simulation Objects. **Set Tags > [Tag]** Allows you to set a new tag or a previously created tag to the selected response.

Scenes and Plots Reference

The following properties and right-click actions are available for scenes and plots in Design Manager.

Scenes and Plots Properties

Object	Properties
Scenes/Plots	*Scenes/Plots* Specifies the scenes or plots from the reference simulation that you want to save for each design in the design study. You can analyze the scenes and plots in Design Manager using Snapshots.
∟ **[scene/plot]**	*Scene Format* Controls the output file format. • **Star-View+ File**: Writes the scene or plot to a scene file (`.sce`), which allows you to work with the same three-dimensional image that you had in STAR-CCM+. You can open a `*.sce` file in STAR-View+. • **Hardcopy**: Writes the scene or plot to a portable network graphics file (`*.png`).

Scenes and Plots Right-Click Actions

Object	Right-Click Actions
Scenes/Plots	None.
∟ **[scene/plot]**	**Relink Scene/ Relink Plot** Allows you to set the selected scene or plot to another scene or plot from the reference simulation, respectively. This option is particularly useful if changes in the reference simulation break the link to the associated object.

Settings Reference

The following properties and right-click actions are available for the settings of a design study in Design Manager.

Contents:

Run Settings Reference
Compute Resource Reference
Macro Files Reference

Run Settings Reference

The following properties are available for the run settings of a design study in Design Manager.

Run Settings Properties

Save Simulation Files

Specifies the designs whose STAR-CCM+ simulation files you want to save.

- **None**: Instructs Design Manager to not save the `*.sim` files of the design simulations.
- **All Best Designs** (for Optimization studies only): Instructs Design Manager to save the `*.sim` files for those designs that are identified as best designs during the run of the design study. For example, the first design in a design study is always identified as the first best design, however, as subsequent design simulations run, Design Manager identifies improved designs and the first design is no longer classed as the best design. Activating this option retains the directory for the first design because it was identified as the best design at one point.

  The best design in a set of designs is the design with the highest performance value given by Eqn. (1376).
- **Last Best Design** (for Optimization studies only): Instructs Design Manager to save the `*.sim` file for the design that is identified as the best design in the design study.
- **All Designs**: Instructs Design Manager to save the `*.sim` files of all design simulations.

Save Log files

Controls whether to save log files to the design directories.

Auto Save

Controls whether to auto save the Design Manager project after each design simulation completes. When **On**, Design Manager saves the simulation file, scenes, and plots of each design according to the specified save options as soon as the design completes. When **Off**, Design Manager writes the output files of the designs at the end of the design study.

Simultaneous Jobs

Specifies how many design simulations are run at the same time. When running an Optimization study, this value must not be greater than the value specified in *Designs to Run* divided by 10. If computational and license resources permit, set this value to 1/5 to 1/7 of the value set in *Designs to Run*.

Compute Processes

Specifies how many cores to run with each STAR-CCM+ design simulation.

Compute Resource Reference

The following properties are available for the compute resources of a design study in Design Manager.

Compute Resource Properties

Type

Controls the type of compute resource that you want Design Manager to use for the design study.

- **Direct**: Runs the design study using direct execution, such as a local machine or workstation.
- **Linux Cluster**: Runs the design study using a generic scheduler.

  This option activates the **Linux Cluster** child node, where you specify the following properties:

Job Submit Command

Command that is used to submit each design simulation on the cluster. The default value (`qsub -l nodes=1:ppn=8`) is set up for a cluster running OpenPBS where each design simulation is submitted on one node running eight cores per node.

To avoid underloading or overloading of compute resources, set Compute Processes to `1`. This value disables the *Compute Processes* property.

Name Identifier

Specifies the flag for the name identifier of a design simulation being submitted to the queue. The default value (`-N`) is set up for a cluster running OpenPBS.

Prefix

Specifies the value for the name identifier that is passed on to the job scheduler. The default value is `mdx_study_design`.

Script File

Allows you to use a submission script that contains the STAR-CCM+ execution command and common command line options. You specify the absolute path and the full name of the script file, which must be available on the cluster drive. Make sure that the script file has execute permission.

STAR-CCM+ Command Line Options

Specifies simulation command line options to pass to the design simulations—such as License Options or other STAR-CCM+ options, see Command-Line Reference.

This property is useful for general simulation command line options that you want to save within your Design Manager project.

Example:`-pio power` (parallel I/O and power licensing).

Alternatively, you can pass simulation options from the command line, see Design Manager Options, which is useful for machine-specific or user-specific options.

As Design Manager controls the resource management for the design simulations, you must not pass resource management options to the design simulations. Therefore, the following resource specification options are not supported:

- `-batchsystem`
- `-bs`
- `-hostregexpr`
- `-machinefile`
- `-np`
- `-on`

Macro Files Reference

The following properties and right-click actions are available for macros in Design Manager.

Macro File Properties

Object	Properties
Macro Files	None.
∟ **[macro]**	*Macro Insertion Point* Controls the stage in the workflow of a design simulation where Design Manager performs the custom steps that are defined in the macro. • **Before Meshing**: Runs the macro immediately before executing the meshing pipeline. • **After Meshing**: Runs the macro immediately after executing the meshing pipeline. • **After Running**: Runs the macro immediately after the STAR-CCM+ solver completes, and before any results are returned to Design Manager. *Macro File* Specifies the path to the macro file (`*.java`). By default, Design Manager uses the relative path to the project root directory. *Tags* Specifies the tags for the macros. For more information, see Tagging Simulation Objects.

Macro Right-Click Actions

Object	Right-Click Actions
Macro Files	**New** Adds a new macro to the workflow of a design simulation.

Object	Right-Click Actions
∟ **[macro]**	**Set New Tag...** Creates a new tag and adds it to the selected macro. For more information, see Tagging Simulation Objects. **Set Tags > [Tag]** Allows you to set a new tag or a previously created tag to the selected macro.

Design Seeds Reference

The following properties and right-click actions are available for design seeds in Design Manager.

Design Seeds Properties

Object	Properties
Design Seeds	None.
∟ **[design seed]**	*Design Sets* **(only for design seeds from design sets)** Specifies design sets from a previous design study. *Number of Seed Designs* Displays the number of designs in the design seed. *Tags* Specifies the tags for the design seed. For more information, see Tagging Simulation Objects.

Design Seeds Right-Click Actions

Object	Properties
Design Seeds	**New Design Seed > [Seed Type]** Creates a new design seed. The following types of design seeds are available: • **Manual Design Seed**: Allows you to enter the designs manually. • **Seed from Design Set**: Allows you to use design sets from a previous design study within the same Design Manager project. For more information, see Seeding an Optimization Study with Predefined Designs.
∟ **[design seed]**	**Edit...** Displays the number of designs in the design seed.

Object	Properties
	Set New Tag... Creates a new tag and adds it to the selected design seed. For more information, see Tagging Simulation Objects. **Set Tags > [Tag]** Allows you to set a new tag or a previously created tag to the selected design seed.

Design Sets Reference

The following properties and right-click actions are available for design sets in Design Manager.

Design Sets Properties

Object	Properties
Design Sets	None.
∟ **[design set]**	*Number of Designs* **(for predefined design sets)** Specifies the number of designs in the selected design set (read-only). *Designs* **(for user-defined design sets)** Specifies the designs in the design set. A user-defined design set contains all designs of the study, by default. You can filter for specific designs or select the designs of interest manually. See Analyzing the Results of a Design Study. *Tags* Specifies the tags for the design set. For more information, see Tagging Simulation Objects.

Design Sets Right-Click Actions

Object	Right-Click Actions
Design Sets	**New** Creates a new design set, which contains all designs of the design study by default.
∟ **[design set]**	**Open Output Table** Opens the Output Table for the selected design set in the *Graphics* window. For information on the displayed values see Output Tables Reference. **Clear All (user-defined design sets)** Removes all designs from the selected design set.

Object	Right-Click Actions
	Export to CSV Exports the input parameter and response values for each designs in the design set. This action uses an operating system-specific standard *Save* dialog to store the designs to file. The exported file is written as comma separated values (`.csv`)—a format supported by many spreadsheets. **Set New Tag...** Creates a new tag and adds it to the selected design set. For more information, see Tagging Simulation Objects. **Set Tags > [Tag]** Allows you to set a new tag or a previously created tag to the selected design set.

Contents:

Output Tables Reference

Output Tables Reference

The following properties and right-click actions are available for the output table of a design set.

Output Table Properties

Design#

Displays the identification numbers of the designs in the design study.

Name

Displays the names of the designs.

State

Displays the run states of the designs. The following states are available:

- No icon: This design has not started.
- This design is queued to run.
- This design has been assigned to run.
- This design has started to run.
- This design is running. This state includes updating the CAD data and parameter values, meshing, and solving.
- This design has executed.
- This design is processing outputs.
- This design has completed successfully.

- This design has failed.

In most cases, the states and change so quickly that you do not notice them in the GUI.

Performance

Displays the performance values of the designs, which allow you to rate the designs. The formulation of the performance value is given by Eqn. (1376).

Parameter/Response

Displays the values of the respective input parameter or response.

Output Table Filter Query Rules

The following filter operators are available for the design properties in an output tables of a user-defined design set:

Properties	Available Operators
Name	**contains** Matches all designs with names that include the specified text string. **does not contain** Matches all designs with names that do not include the specified text string. **matches** Matches all designs with the specified name. **does not match** Matches all designs with names other than the specified name. **starts with** Matches all designs with names that begin with the specified text string. **does not start with** Matches all designs with names that do not begin with the specified text string. **ends with** Matches all designs with names that end with the specified text string. **Note:** A name can include a trailing number which is considered part of the name, such as Design 1 . Therefore, for example, an entry of `ign` does not find the Design 1. **does not end with** Matches all designs with names that do not end with the specified text string.

Properties	Available Operators
State	**All** Matches all designs in the design study. **Successful** Matches all designs that have completed successfully. **Error** Matches all designs that have failed. **Not Started** Matches all designs that have not started.
Design# *Performance Parameter Response*	**all values** Matches all designs in the design study. **less than** Matches all designs with respective quantity being smaller than a specified value. **greater than** Matches all designs with respective quantity being larger than a specified value. **between** Matches all designs with respective quantity being in a specified range.

Output Table Right-Click Actions

The following actions are available when you right-click the row of a specific design in the output table:

Open Scenes > [Scene] > [Snapshot]

Allows you to create a new scene snapshot or open a previously created scene snapshots for the selected design. See Snapshots.

Open Plots > [Plot] > [Snapshot]

Allows you to create a new plot snapshot or open a previously created plot snapshot for the selected design. See Snapshots.

Set Design Set > [Design Set]

Allows you to assign the selected design to a new design set or a previously created design set. Using `<Ctrl>`, you can multi-select and assign designs to a design set.

Show Design Details

Opens the *Design details* dialog that displays the following information:

- **Design**: The number of the design.

- **Design status**: The simulation state of the design.
- **Performance value**: The performance value of the design given by Eqn. (1376).
- **Design Artifact Directory**: The path to the directory where the simulation results of the design are stored.
- **Messages**: Error messages that show the reasons for a possible failure of the design simulation.

Snapshots Reference

The following properties and right-click actions are available for snapshots in Design Manager.

Snapshots Properties

Object	Properties
Snapshots	*Children* Specifies the number of snapshots in the Design Manager project (read-only).
└ **[snapshot]**	*Design Study* Specifies the design study from which the snapshot extracts the data. *Design Set* Specifies a design set in the specified design study, from which the snapshot extracts the data. *Scene/Plot* Specifies a scene or plot in the specified design study. *Tags* Specifies the tags for the snapshot. For more information, see Tagging Simulation Objects. *Lock Design* Locks the snapshot display to the specified design. *Design* Specifies the design to display in the snapshot.
└ **Animation**	*Target Frame Rate (fps)* Maximum number of animation frames that are displayed in one second (less for complex scenes, or when less computing power is available).
└ **Annotations**	*Annotations* A list of the annotations added to the snapshot. Editing the property accesses the in-place object selector to add or remove annotations.
└ **[annotation]**	*Width Stretch Factor* Scales the width of the annotation, with the default of 1.0 equaling 100% of the original width

Object	Properties
	Height The height of the annotation as a fraction of the visualization display height, where 1.0 equals the full height *Position* The position of the lower left corner of the text in the *Graphics* window based on the specified *Coordinate System*. *Visible* When **On**, shows the annotation in the display. When **Off**, the annotation is hidden. *Coordinate System* Selects the type of coordinates to use for *Position*. • **Normalized Viewport**: Shows x-y values as fractions of the length and height of the Graphics window display, respectively. The origin is the lower left-hand corner of that display. • **World**: Shows x-y-z values as global coordinates. As you rotate the object, the annotation keeps its position relative to that object. *Interactive* When **On**, allows you to move the annotation in the display using the mouse. When **Off**, locks the position of the annotation in the display. *Location* Controls the visibility of the annotation relative to the objects in the display. • **Background**: Shows the objects in front of the annotation. • **Foreground**: Shows the annotation in front of the objects.

Snapshots Right-Click Actions

Object	Right-Click Actions
Snapshots	**New > [Snapshot Type]** Creates a new snapshot of the specified type, where **[Snapshot Type]** can be **Plot** or **Scene**.
∟ **[snapshot]**	**Open** Opens the snapshot in the *Graphics* window. **Hardcopy** Saves the snapshot to an image file. A customized, operating system-specific *Save To File* dialog appears, which allows you to write the selected snapshot to file.

Object	Right-Click Actions
	Linked View Allows you to link several snapshots together so that they move in tandem as you interact with one of them. **Set New Tag...** Creates a new tag and adds it to the selected snapshot. For more information, see Tagging Simulation Objects. **Set Tags > [Tag]** Allows you to set a new tag or a previously created tag to the selected snapshot.
⌊ **Animation**	None.
⌊ **Annotations**	
⌊ **[annotation]**	

Plots Reference

The following properties and right-click actions are available for project plots in Design Manager.

Plots Properties

Object	Properties
Plots	**Plots** Specifies the number of project plots in the Design Manager project (read-only). **Output Verbosity** Provides additional output in the *Output* window during the run.
└ [plot]	**Title** The title of the project plot, which appears centered at the top of the plot. Its position and format attributes cannot currently be modified. To remove the title, set this value to blank. **Tags** Specifies the tags for the project plot. For more information, see Tagging Simulation Objects. **Antialiasing** Controls appearance of labels to visual quality. When **On**, blends the edge of the characters with the background, creating a smoother appearance on screen. When **Off**, causes edges of characters to remain sharp, allowing for better printing. **Hover Interactions** When **On**, causes the display to respond to mouse hovering. When **Off**, leaves the display at its default, with no response to hovering. **Show Highlights** When **On**, highlights objects that you select in the plot tree in the plot display. **Title Font** Specifies the font family, font style and font size. Use the customizer to activate a dialog to set this property (changing the text will have no effect). **Footer** A text box in which you can type a footer for the plot. **Footer Font** The font family, font style and font size of the footer. Use the customizer to activate a dialog to set this property (changing the text will have no effect).

Object	Properties
	Data Series Order Specifies the order of appearance of data series in the plot display. See Changing the Data Display Order.
∟ **Data Series**	None.
∟ **[design study]**	See Data Series Reference.
∟ **Axes**	See Customizing Axes.
∟ **[axis]**	
∟ **Legend**	See Customizing the Legend.

Plots Right-Click Actions

Object	Right-Click Actions
Plots	**New Plot** Creates a new project plot. **Open All Plots** Displays all project plots in the *Graphics* window. **New Group** Facilitates the grouping of project plots by creating a new group node within the manager node. **Group By** Allows you to group project plots according to any of the properties of the plots in the manager node, listed in a submenu of this menu item. **Ungroup** Reverses the grouping process.
∟ **[plot]**	**Open** Displays the selected project plot in the *Grapics* window. **Reorder Data Series** Activates a dialog in which you adjust the order of appearance of data series in the project plot.

Object	Right-Click Actions
	If the project plot display contains multiple data series, you can control the order in which each data series is displayed relative to the rest. This control is important for plots where data series overlap. You can see the order of data series in three places in the workspace: the plot display, the legend, and the node tree. See Changing the Data Display Order. **Tabulate** Displays the data series in the project plot in pairs of X and Y value columns in a spreadsheet-like view. **Export** Exports the data that are displayed in the project plot. This action uses an operating system-specific standard *Save* dialog to store the data points of the plot to file. The exported file is written as comma separated values (`.csv`)—a format supported by many spreadsheets. **Hardcopy** Saves the project plot to an image file. A customized, operating system-specific *Save To File* dialog appears, allowing you to write the selected plot to file. **Apply Plot Options** Sets preferred plot attributes. **Set New Tag...** Creates a new tag and adds it to the selected project plot. For more information, see Tagging Simulation Objects. **Set Tags > [Tag]** Allows you to set a new tag or a previously created tag to the selected project plot.
└ Data Series	**Add Data** Allows you to add data of a design study to the project plot. **Reorder Data Series** As for **Plot n**.
└ [design study]	See Data Series Reference.
└ Axes	See Customizing Axes
└ [axis]	
└ Legend	See Customizing the Legend

Contents:

Data Series Reference
Data Points Reference

Data Series Reference

Design Manager data series specify the designs of a design study that are displayed in a project plot.

Data Series Properties

Object	Properties
[design study]	*Input Design Study* Specifies the design study whose designs you want to analyze. *Input Design Sets* Specifies the design set of the selected design study that you want to display in the project plot. *Legend Name* Changes the legend text in the project plot display. *Lock Legend Name* When **On**, prevents any automatic changes to the legend name. The legend name is linked to the node name which in turn reflects the name of the part that the data series represents. Therefore, name changes to the part are automatically reflected in the name of the data series node and the legend name. If this property is activated, the current legend name is locked, effectively unlinking the property from the node name. Setting the legend name automatically activates this lock. *Tags* Specifies the tags for the design study data set. For more information, see Tagging Simulation Objects. *Sort Plot Data* When **On**, the data are sorted—this ensures that the abscissa (horizontal axis) values run in an increasing order. **Note:** In parallel, cell data is not sorted across partitions. This does not affect the rendered result for a scatter plot. However, in the case of a line chart, if the source data is not previously sorted, this affects how the data is plotted. *X Offset* A scalar offset that will cause all values of the x-axis data to be shifted by this amount. The units of the offset are implicitly equivalent to those associated with the x-axis data.

Object	Properties
	Y Offset A scalar offset that will cause all values of the y-axis data to be shifted by this amount. The units of the offset are implicitly equivalent to those associated with the y-axis data. *X Scale Factor* A scaling factor that will cause all values of the x-axis data to be scaled by this amount. *Y Scale Factor* A scaling factor that will cause all values of the y-axis data to be scaled by this amount.
∟ **Bottom Axis Data** ∟ **Left Axis Data**	*Type* Controls the type of data to be displayed on the axis: • **Parameter**: Displays a specific input parameter of the design study, that you specify in the corresponding **Parameter** child node. See Input Parameters Reference. • **Response**: Displays a specific response of the design study, that you specify in the corresponding **Response** child node. When you activate the *Show Constraints* property, the plot highlights the field of feasible designs, where the field is determined by the constraints that you set for the specified response. See Responses Reference. • **History**: Displays the design numbers. *Axis* Controls the axis on which the data is plotted. The choice of axes depends on the total number of axes in the project plot. *Tags* Specifies the tags for the axis data. For more information, see Tagging Simulation Objects.
∟ **Best Design History**	*Visible* When **On**, the project plot displays the path of the design study to the best design. This path passes the designs that are identified as best designs at one point during the run of the design study. The best design in a set of designs is the design with the highest performance value (see Eqn. (1376)). You specify the style of the path in the **Line Style** child node. See Line Style Properties. *Legend Name* Specifies the legend text that is associated with the best design history.

Object	Properties
∟ **Best Design**	*Visible* When **On**, the project plot marks the best design in the design study, where the best design is the design with the highest performance value (see Eqn. (1376)). You specify the style of the mark in the **Symbol Style** child node. See Symbol Style Properties. *Legend Name* Specifies the legend text that is associated with the best design.
∟ **Baseline Design**	*Visible* When **On**, the project plot marks the baseline design. The baseline design is the first design that Design Manager runs in the design study. For the various study types, this design is determined as follows: • Manual: The first design that you specify in the list of designs to run. • Sweep/DOE: The design that combines the minimum values for all of your input parameters. • Optimization/Stochastic: The design that combines the *Baseline* values that you specify for each input parameter. You specify the style of the mark in the **Symbol Style** child node. See Symbol Style Properties. *Legend Name* Specifies the legend text that is associated with the baseline design.
∟ **Line Style**	Controls the appearance of the line for the design study data set. See Line Style Properties.
∟ **Symbol Style**	Controls the appearance of the point symbols for the design study data set. See Symbol Style Properties.

Data Series Right-Click Actions

Object	Right-Click Actions
[design study]	*Reorder* Activates a submenu of four options for adjusting the order of appearance of the data series in the project plot display. See Changing the Data Display Order. **Tabulate** Displays the data of the **Input Design Sets** in a spreadsheet-like view.

Object	Right-Click Actions
	Export Exports the data of the **Input Design Sets**. This action uses an operating system-specific standard *Save* dialog to store the data points of the project plot to file. The exported file is written as comma separated values (`.csv`)—a format supported by many spreadsheets. **Apply Plot Options** Sets preferred plot attributes. **Set New Tag...** Creates a new tag and adds it to the selected data series. For more information, see Tagging Simulation Objects. **Set Tags > [Tag]** Allows you to set a new tag or a previously created tag to the selected data series.
└ Bottom Axis Data	**Set New Tag...** Creates a new tag and adds it to the selected data series. For more information, see Tagging Simulation Objects. **Set Tags > [Tag]** Allows you to set a new tag or a previously created tag to the selected data series.
└ Left Axis Data	
└ Best Design History	None.
└ Best Design	
└ Baseline Design	
└ Line Style	**Apply Plot Options** Sets preferred plot attributes.
└ Symbol Style	

Data Points Reference

The following actions are available when you right-click a specific data point in a plot that displays in the *Graphics* window:

Open Scenes > [scene] > [snapshot]

Allows you to create a new scene snapshot or open a previously created scene snapshots for the selected design. See Snapshots.

Open Plots > [plot] > [snapshot]

Allows you to create a new plot snapshot or open a previously created plot snapshot for the selected design. See Snapshots.

Set Design Set > [design set]

Allows you to assign the selected design to a new design set or a previously created design set. Using <Ctrl>, you can multi-select and assign designs to a design set.

Show Design Details

Opens the *Design details* dialog that displays the following information:

- **Design**: The number of the design.
- **Design status**: The simulation state of the design.
- **Performance value**: The performance value of the design, see Eqn. (1376).
- **Design Artifact Directory**: The path to the directory where the simulation results of the design are stored.
- **Messages**: Error messages that show the reasons for a possible failure of the design simulation.

Tools Reference

The **Tools** manager node contains various objects that facilitate your analysis.

The following objects are available:

- Annotations
- Custom Trees
- Filters
- Layouts
- Tags
- Units
- Views

Annotations

An annotation is an object that you can use to add information to a snapshot display in the *Graphics* window. The following annotations are available by default:

- **Design Name**: Displays the name of the respective design that is analysed within a design study.
- **Performance Value**: Displays the performance value of a design given by Eqn. (1376). The perforamans value is a rating value that allows you to judge the design.

Additional simple text annotations allow you to place customized text labels in snapshot displays.

For more information, see Annotations.

Custom Trees

If you have a large simulation with many objects in the tree, custom trees allow you to quickly access the objects that you use most.

For more information, see Using Custom Simulation Trees.

Filters

A filter uses queries to narrow groups of STAR-CCM+ objects. A query consists of one or more conditional statements, also called "predicates", by which STAR-CCM+ searches among the objects in the simulation.

For more information, see Object Filters.

Layouts

A window layout is an arrangement of windows in the graphics display, which consists of one or more modes. They can be used to save, restore, and edit a specified visual display.

For more information, see Preserving the Window Layout.

Tags

Tags can help you find and select objects easier. Once a tag has been created, it can be applied to one or more objects to flag them as being different from other objects.

For more information, see Tagging Simulation Objects.

Units

STAR-CCM+ has flexible and customizable support for engineering units. The standard SI and USCS systems are built in, and you can add new units as desired.

For more information, see Engineering Units.

Views

In STAR-CCM+, every scene has a camera to convert the three-dimensional objects in a scene into the two-dimensional image visible on the screen. The information defining the orientation, position and projection of the three-dimensional objects is stored in a view object.

For more information, see Working with Views.

Toolbars Reference

The following toolbars are specific to Design Manager projects.

: Design Manager Animation toolbar

: Design Manager Snapshot toolbar

: Design Manager Vis toolbar

For information on the toolbars that are common in STAR-CCM+, see STAR-CCM+ Toolbar Reference.

Design Manager Animation

This toolbar resembles the controls of a common video player to animate scene snapshots, where the scenes are exported as Scene Files (`*.sce`).

(Play/Pause)

Toggles between playing the scene animation and pausing it.

(Stop)

Stops the animation in the scene.

(Single Step Forward)

Moves the paused animation forward one frame.

(Single Step Backward)

Moves the paused animation backward one frame.

Design Manager Snapshot

This toolbar lets you step through the designs of a design set in snapshots, output tables, and study plots.

(First Design)

Jumps to the first design in the design set.

(Previous Design)

Steps backward to the previous design in the design set.

(Next Design)

Steps forward to the next design in the design set.

(Last Design)

Jumps to the last design of the design set.

(Reset Current Design)

Resets the design, that is, no design is displayed.

Design Manager Vis

This toolbar lets you work interactively with scene snapshots, where the scenes are exported as Scene Files (`*.sce`).

(Reset View)

Restores the view to a zoom factor such that the model takes up most of the display. The view angle is unchanged. Use this function if the model has somehow zoomed or panned to such a position that you can no longer see it on the display.

(Rotate)

Rotates the view around a focal point that your most recent mouse click in the display sets.

(Rubberband Zoom)

Allows you to zoom to a specific area in the display by clicking two points that define the opposing corners of the zoomed area. Press `<Esc>` to cancel.

(View)

Opens a drop-down menu that allows you to set, save or restore a view.

- **Store Current View**
  Records properties about the projection mode, zoom factor, pan position and view angle and adds a node to the **Views** node of the simulation tree. It is named View N, where N is the first available number that is not used by existing views with the same prefix. So if you have a view, View 1, then this operation creates a view, View 2.
- **Restore View**
  Contains a submenu of saved views, including a thumbnail of the view. Select one of the submenu items to restore that view.
- **Projection Mode**
  Contains a submenu that toggles between perspective and parallel mode for the current view.
- **Views**
  Contains a submenu of standard view angles using the axes as reference vectors.
- **Standard Views**
  Contains a submenu of standard views using the aspects of the object in the scene.
- **View Coordinate System**
  Contains a submenu of coordinate systems in the simulation that a view can use.

(Previous View)

Reverses the latest change to the view.

(Next View)

Repeats a change that was subjected to the `Previous View` command.

Command Line Options Reference

The following command line options are available for Design Manager.

Note: Some commands accept arguments with embedded quotes. When using C-shell, add `set backslash_quote` to the `~/.cshrc` file. This variable ensures that embedded quotes are correctly escaped.

Design Manager Options

The `starccm+ [project]` command launches the Design Manager server and loads the specified Design Manager project file (`*.dmprj`). The following Design Manager server options are available:

`-dmnoshare`

Specifies the Unlicensed Design Manager Server Licensing Scheme.

Note: This option is not supported when running Design Manager in Pre-Allocation mode.

`-passtodesign <options>`

Specifies simulation options to pass to the design simulations, such as License Options or other STAR-CCM+ options, see Command-Line Reference.

Passing simulation options from the command line is useful for machine-specific or user-specific options—such as a Power-On-Demand license key.

Example: `starccm+ [project] -dmnoshare -passtodesign "-power -podkey <value> -licpath 1999@flex.cd-adapco.com"`

Alternatively, you can pass command line options from the GUI, see STAR-CCM+ Execution Command. Passing command line options from the GUI is useful for general simulation command line options that you want to save within your Design Manager project.

As Design Manager controls the resource management for the design simulations, you must not pass resource management options to the design simulations. Therefore, the following resource specification options are not supported:

- `-batchsystem`
- `-bs`
- `-hostregexpr`
- `-machinefile`
- `-np`
- `-on`

Note: The `-passtodesign <options>` option does not support arguments that require double quotes, such as `-mpidriver platform:"-aff:automatic:bandwidth"`. You must pass these options from the GUI.

-doepower

Specifies to pull a Power Session license (`ccmppower`) instead of a `ccmpsuite` license as session license when using the Default Licensing Scheme. Using this option together with `-podkey <value>` uses a Power-on-Demand license.

When running Design Manager in Pre-Allocation mode and using the `-doepower` option, you must pass this option also to the Design Manager server.

Example: `starlaunch jobmanager --command "starccm+ [project] -doepower -passtodesign \"-doepower\""`

-notokens

Specifies to not use `doetoken` licenses for the design simulations.

When running Design Manager in Pre-Allocation mode and passing the `-notokens` option to the design simulations, you must pass this option also to the Design Manager server.

Example: `starlaunch jobmanager --command "starccm+ [project] -notokens -passtodesign \"-notokens\""`

Resource Manager Options

When running Design Manager in Pre-Allocation mode, Resource Manager provides the actual resource management functionality. The following command line options are available for `starlaunch jobmanager`:

--command <batch_command>

Specifies the command that runs your Design Manager project in batch.

--slots 0

Specifies to allocate no processor for the job manager process.

--batchsystem <system>

Specifies the batch management system on your cluster.

--resources <host[:np][,host[:np]]>

Specifies a group of network machines on which Design Manager is allowed to run the design simulations.

--resourcefile <file>

Specifies an MPI file that lists the machines on which Design Manager is allowed to run the design simulations.

Example: `starlaunch jobmanager --command "[STARCCM+_INSTALL_DIR]/star/bin/starccm+ -rsh ssh -batch run [project]" --slots 0 --batchsystem pbs`

License Options

The following options control the license usage when running a design study in Design Manager:

-doepower

Uses a Power Session license (`ccmppower`) instead of a `ccmpsuite` license as session license when using the Default Licensing Scheme. Using this option together with `-podkey <value>` uses a Power-on-Demand license.

-licpath <path:...>

Specifies a license path that overrides the default license path. When this option is used, STAR-CCM+ uses the path to define a new `CDLMD_LICENSE_FILE` environment variable for the spawned server session.

Example: `-licpath 1999@flex.cd-adapco.com` (Power-on-Demand license server)

-nosuite

Prevents `ccmpsuite` licenses from being used to enable cores for parallel computation. Only parallel licenses, such as Power Tokens, are used to enable cores for parallel computation.

-notokens

Specifies to not use `doetoken` licenses for the design simulations.

-podkey <value>

Specifies a Power-On-Demand license key.

-power

Specifies to check out Power Session licenses for each design simulation. Using this option together with `-podkey <value>` enables Power-on-Demand licensing.

Note: This option is not supported when running Design Manager in Pre-Allocation mode.

-tokensonly

Specifies to only use `doetoken` licenses for the design simulations. With this option, you can also use `doetoken` licenses for regular parallel runs in STAR-CCM+.

For the Default Licensing Scheme, the following table displays the supported combinations of license option and the associated license usage:

Command Line Option	Session License	Core License	Parallel Licenses
None.	`ccmpsuite`	`DOEtoken` / `ccmpsuite`	`DOEtoken` / `ccmpsuite`
`-tokensonly`	`ccmpsuite`	`DOEtoken`	`DOEtoken`
`-notokens`	`ccmpsuite`	`ccmpsuite`	`ccmpsuite`
`-notokens -power`	`ccmpsuite`	`ccmppower`	N/A
`-nosuite`	`ccmpsuite`	`DOEtoken` / `ccmpsuite`	`DOEtoken`

Command Line Option	Session License	Core License	Parallel Licenses
-power	ccmpsuite	DOEtoken / ccmppower	DOEtoken / ccmpsuite
-power -nosuite	ccmpsuite	DOEtoken / ccmppower	DOEtoken
-doepower	ccmppower	DOEtoken / ccmpsuite	DOEtoken / ccmpsuite
-doepower -tokensonly	ccmppower	DOEtoken	DOEtoken
-doepower -notokens	ccmppower	ccmpsuite	ccmpsuite
-doepower -notokens -power	ccmppower	ccmppower	N/A
-doepower -nosuite	ccmppower	DOEtoken / ccmpsuite	DOEtoken
-doepower -power	ccmppower	DOEtoken / ccmppower	DOEtoken / ccmpsuite
-doepower -power -nosuite	ccmppower	DOEtoken / ccmppower	DOEtoken

where DOEtoken / ccmpsuite or DOEtoken / ccmppower means:

1. Design Manager tries to satisfy the license requirement with DOEtoken.
2. If insufficient licenses, Design Manager tries to satisfy the remaining requirement with ccmpsuite or ccmppower, respectively.

For the Unlicensed Design Manager Server Licensing Scheme, the command line options and the corresponding licensing rules follow the stand-alone STAR-CCM+ Licensing.

Interacting with CAE Software

STAR-CCM+ can integrate a wide range of other computer-aided engineering (CAE) tools into a simulation.

The basic import and export features let you use external geometry, surface and volume mesh data, as well as export such data from a STAR-CCM+ simulation. Therefore you can incorporate a host of external file formats and CAE models into your simulation.

A more advanced integration of CAE with STAR-CCM+ is coupling. This capability allows you to use STAR-CCM+ to obtain part of the solution, and a second code to obtain another part of the solution. By coupling the two codes together, solution data is passed between them at specified intervals during the simulation, and you can obtain a solution for the whole model. STAR-CCM+ offers two methods of coupling: file-based coupling, whereby data is exchanged between codes via files stored in the computer disk, and co-simulation, in which data is exchanged automatically while the STAR-CCM+ simulation and the partner simulation are running.

File-based coupling is very versatile and can be used for most types of coupled simulation. It relies on the user to manually execute the coupling steps via the STAR-CCM+ interface, although the process can be automated via scripts. Co-simulation allows for data exchange between STAR-CCM+ and the following codes:

- STAR-CCM+ (co-simulation between two STAR-CCM+ simulations)
- LMS Imagine.Lab Amesim
- Abaqus
- GT-SUITE
- WAVE
- RELAP
- Reacting Channel (1D Plug Flow Reactor in DARS)
- Isight

For co-simulation with one-dimensional codes (that is, with GT-SUITE, LMS Amesim, WAVE, and RELAP), coupling works best if the interfaces are located where the flow is largely one-dimensional, as this will reduce errors due to averaging. The worst place for a coupling interface is where there is partial reverse flow, because the mean flow is not representative of either the forward or reverse flows.

The methods available for each coupling technique depend on which third party software you are coupling with STAR-CCM+.

Note: Currently, co-simulation does not support host specification using the **IPv6** communication protocol.

On Windows version 7 or older, the length of the PATH environment variable can affect the co-simulation functionality. You are advised to reduce the length of the PATH by storing the path to the STAR-CCM+ installation in a separate environment variable:

```
CCM=[path to STAR-CCM+ INSTALL_DIR]
PATH=%CCM%\lib;%CCM%\vtk;...
```

Contents:

Importing into STAR-CCM+
Exporting Mesh and Solution Data
File-Based Coupling
STAR-CCM+ to STAR-CCM+ Co-Simulation
LMS Amesim Co-Simulation
Abaqus Co-Simulation
GT-SUITE Co-Simulation
WAVE Co-Simulation
RELAP Coupling
Reacting Channel Coupling
Co-Simulation API
Data Mappers
The STAR-CCM+ Component for Isight

Importing into STAR-CCM+

STAR-CCM+ allows you to import models from a number of other CAE packages so that you can use data associated with the model in your STAR-CCM+ simulation.

You can use an imported CAE model in a standalone STAR-CCM+ analysis, or for coupling STAR-CCM+ with other codes using file-based coupling. You can also import data from STAR-CD for post-processing in STAR-CCM+.

Contents:

Importing Mesh and Solution Data from STAR-CD
Importing CAE Models

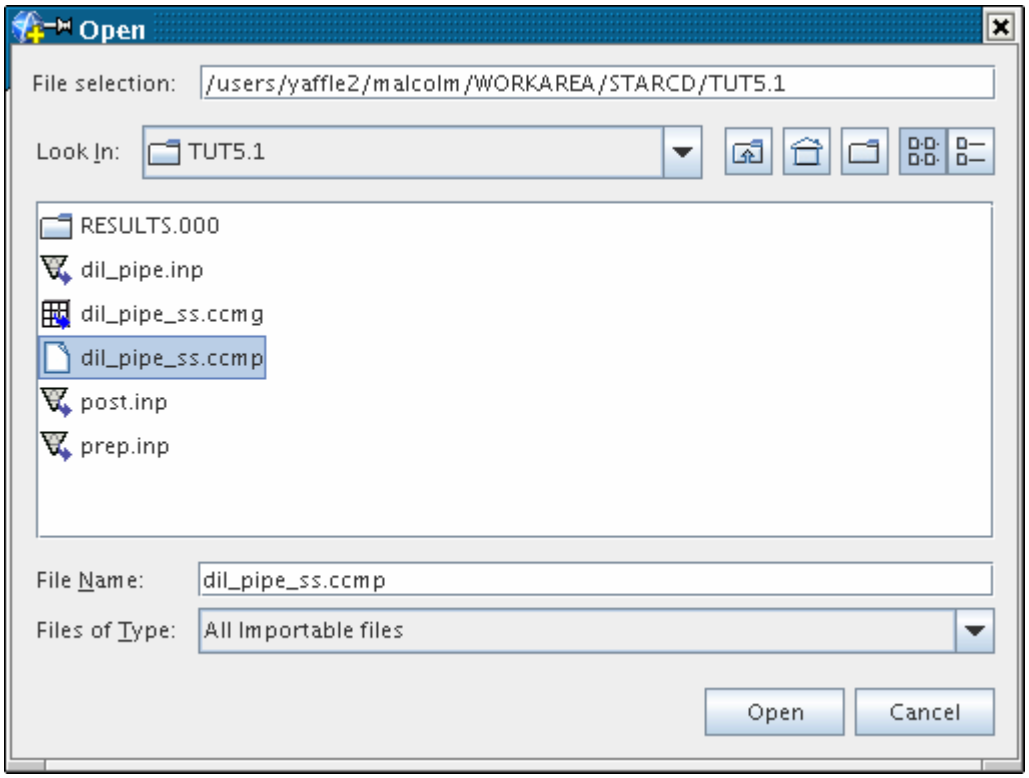
Importing Mesh and Solution Data from STAR-CD

STAR-CCM+ allows you to import mesh and solution data from STAR-CD, in order to post-process the STAR-CD solution data using the STAR-CCM+ visualization tools. This import option does not support Lagrangian solution data.

Since the STAR-CD v4.04 release, the solution data provided by STAR (the solver) has been stored in two files: the `.ccmp` file for standard (restart) solution data, and the `.ccmt` file for transient solution data. Both of these files contain a link to the corresponding `.ccmg` file for the mesh data; if this `.ccmg` file is missing, it is not possible for STAR-CCM+ to read either the `.ccmp` or the `.ccmt` file. This import option does not support solution data from moving mesh calculations.

The procedure for importing both mesh and solution data from STAR-CD is described as follows (assuming that the necessary STAR-CD files have been copied into the working directory):

1. Launch STAR-CCM+ and start a new simulation in the manner appropriate to your environment.
2. Select **File** > **Import** > **Import Volume Mesh...**
3. In the *Open* dialog, select the desired `.ccmp` or `.ccmt` file. Make sure that the corresponding `.ccmg` file is also available.



4. Click **Open**.

If solution data is successfully read from the file, information on the process is displayed in the *Output* window. This information includes details of the solution data found in the file, and details of the scalar fields in STAR-CCM+ to which the solution has been mapped. An example of this output is shown below.

```
CCM file version 20617
```

```
Cell type[1] label: "Default_Fluid", material: "fluid" (1) porosity: 0
spin: 1 group: 0
Boundary type[0] id: 0, name: "Default_Boundary_Region", label: "",
type: "wall"
Boundary type[1] id: 1, name: "Inner_inlet", label: "", type: "inlet"
Boundary type[2] id: 2, name: "Outer_inlet", label: "", type: "inlet"
Boundary type[3] id: 3, name: "", label: "", type: "pressure"
Boundary type[4] id: 4, name: "", label: "", type: "symmetry plane"
Reading solution from state Restart_1
Read 'Velocity Component U' (SU) m/s  at 2000 cells
Read 'Velocity Component U' (SU) m/s  at 960 faces
Read 'Velocity Component V' (SV) m/s  at 2000 cells
Read 'Velocity Component V' (SV) m/s  at 960 faces
Read 'Velocity Component W' (SW) m/s  at 2000 cells
Read 'Velocity Component W' (SW) m/s  at 960 faces
Read 'Mass Flux' (MassFlux) kg/s  at 5500 faces
Read 'Mass Flux' (MassFlux) kg/s  at 960 faces
```

... and so on ...

```
Processing mesh:
Mapping imported data DENS to Density
Mapping imported data ED to Turbulent Dissipation Rate
Mapping imported data H to Enthalpy
Mapping imported data P to Pressure
Mapping imported data SU to Velocity-X
Mapping imported data SV to Velocity-Y
Mapping imported data SW to Velocity-Z
Mapping imported data T to Temperature
Mapping imported data TE to Turbulent Kinetic Energy
Mapping imported data VIS to Turbulent Viscosity
```

It is observed that solution data is read for both cells and faces during the import process. This corresponds to cell and wall data in STAR-CD.

Importing CAE Models

Imported models are CAE models that have been prepared in other CAE products, for example an FEA code, and imported into STAR-CCM+.

The model can be formed from surfaces, shells, beams and/or volume elements. When you import a CAE model, STAR-CCM+ recognizes geometrical elements that are defined in the input file, and adds a region and a geometry part for each volume that is defined in the CAE model. STAR-CCM+ also adds corresponding boundaries and part surfaces for any surface that is defined in the CAE model. If the CAE model contains vertex specifications, including single feature vertices or groups of vertices, STAR-CCM+ creates associated part points, which can be used for stress analysis with the **Solid Stress** model (see Solid Stress: Preparing the Geometry).

Note: When you import a CAE model, do not delete or modify the association between the regions and parts that represent the imported CAE model. For imported CAE models, part-region associations cannot be restored once they are deleted.

A CAE model can be imported as a Region, or as a Region And Imported Model. The first option allows you to import a CAE model into STAR-CCM+ as a region, which can then be used in the same way as any other region in the simulation. The second option allows you to couple STAR-CCM+ to other codes using the file-based coupling method: the CAE model is still imported as a region, but an extra node called **Imported Models** is added to the simulation tree. From this node, you can map solution data from the STAR-CCM+ model onto the CAE model, and export it in a format that can be read into the third-party software. Similarly, you can import solution data that is associated with the CAE model into STAR-CCM+, and use it to define a boundary condition in the simulation.

You can also use an imported CAE model for FE stress analysis. As STAR-CCM+ FE simulations are entirely parts-based, both import options generate a geometry part under the **Geometry** > **Parts** node. When using a CAE model for FE stress analysis, make sure that the model contains the surface definitions required by your analysis.

An imported CAE model can be visualized in a scene, in the same way as regions, boundaries, and parts. For more information, see Visualizing an Imported CAE Model.

STAR-CCM+ supports the import of CAE models containing shells, surfaces and/or volumes from the following codes:

- Abaqus
- ANSYS
- Nastran
- RadTherm (shells only)
- es-ice or pro-STAR

It is also possible to import a CAE model in the STAR-CCM+ boundary format (`.sbd`). These files are used in the same way as CAE models imported from the codes that are listed above, but you cannot import them as regions. STAR-CCM+ also supports the import of Ensight input files, which can be used to visualize and compare multiple solutions in one simulation. For more information, see Working with the STAR-CCM+ Boundary Data (.sbd) Format and Working with Ensight.

For Nastran models, you can import data that was defined in local coordinate systems. STAR-CCM+ transforms the data into a global coordinate system during the import process. All other formats are imported in a global coordinate system, as importing from local coordinate systems is not supported.

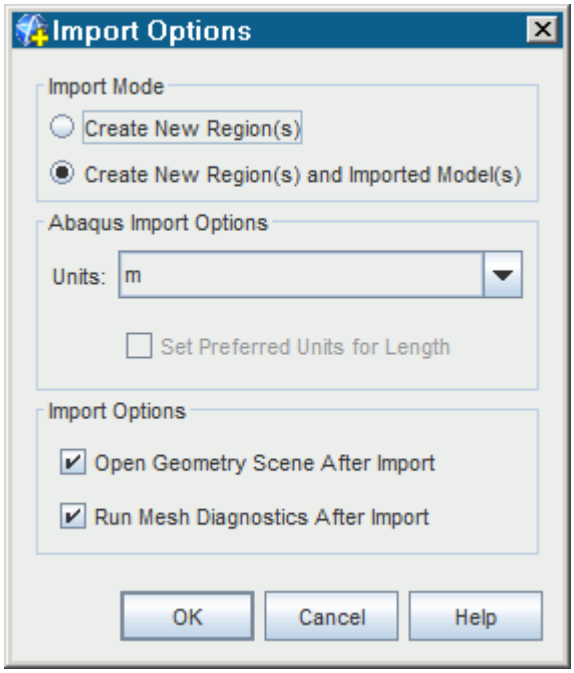
Contents:

Importing a CAE Model: Workflow
What Is the Imported Models node?
Visualizing an Imported CAE Model

Importing a CAE Model: Workflow

To import a CAE model:

1. Select **File** > **Import** > **Import CAE Model...**
2. In the *Open* dialog, select the file to import and click **Open**.
3. In the *Import Options* dialog, set the *Import Mode* to one of the following options:
   - **Create New Region(s)** - This import option allows you to import a CAE model as a STAR-CCM+ region. The imported region can then be used in the same way as any other region in STAR-CCM+.
   - **Create New Region(s) and Imported Model(s)** - This import option allows you to access the Imported Models framework for file-based coupling operations. See Using File-Based Coupling.



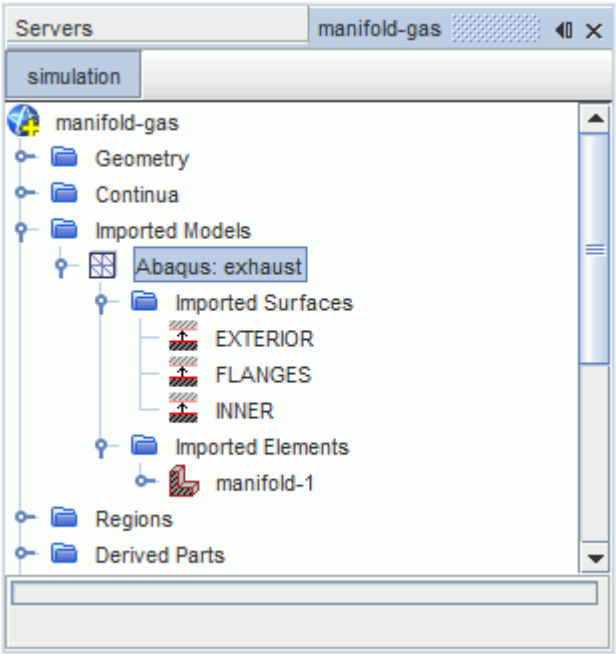
   You can use the *Units* menu to specify the units of the CAE model.

4. Click **OK.**

   The progress is shown in the *Output* window. STAR-CCM+ creates a new region from each body that is defined in the CAE model. If the CAE model contains any other surface definitions, STAR-CCM+ creates

corresponding boundaries. Surfaces that are not included in any surface definitions form a default boundary.

If you set *Import Mode* to **Create New Region(s) and Imported Model(s)**, STAR-CCM+ also adds the imported model under the **Imported Models** node, for file-based coupling with the corresponding CAE code.



The imported CAE model node typically contains two subnodes, **Imported surfaces** and **Imported elements**. These nodes show the imported surfaces and elements that are defined in the CAE model file.

Note: When you import a CAE model containing a volume, STAR-CCM+ automatically extracts the free surface of the imported body and creates a corresponding boundary. When you import a model containing shells, STAR-CCM+ automatically locates the positive and negative side of each shell.

Note: The import procedure is different when importing a CAE model from pro-STAR or es-ice.

Contents:

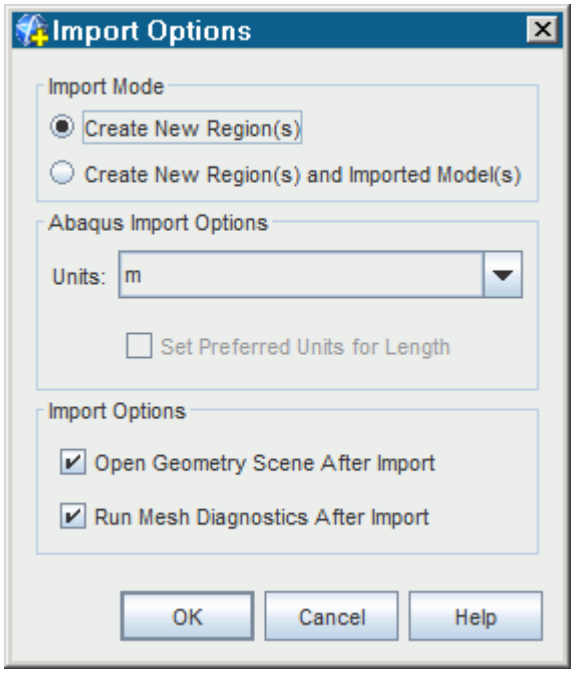
Importing a CAE Model as a Region
Importing a CAE Model as a Region and Imported Model

Importing a CAE Model as a Region

Importing a CAE model as a region causes STAR-CCM+ to create a new region from each volume that is defined in the CAE model. If the CAE model contains any surface definitions, these create new boundaries. Surfaces that are not included in any surfaces definitions form a default boundary.

To import a model as a region:

1. Follow the steps outlined in the previous section, Importing a CAE Model.
2. In the *Import Options* dialog, set *Import Mode* to **Create New Region(s)**.



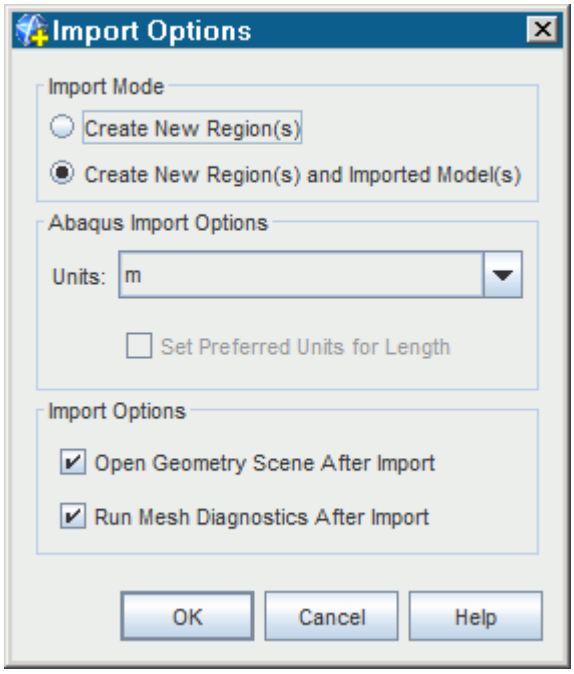
When a CAE model is imported as a region, it can be used in the same way as any other region in the simulation.

Importing a CAE Model as a Region and Imported Model

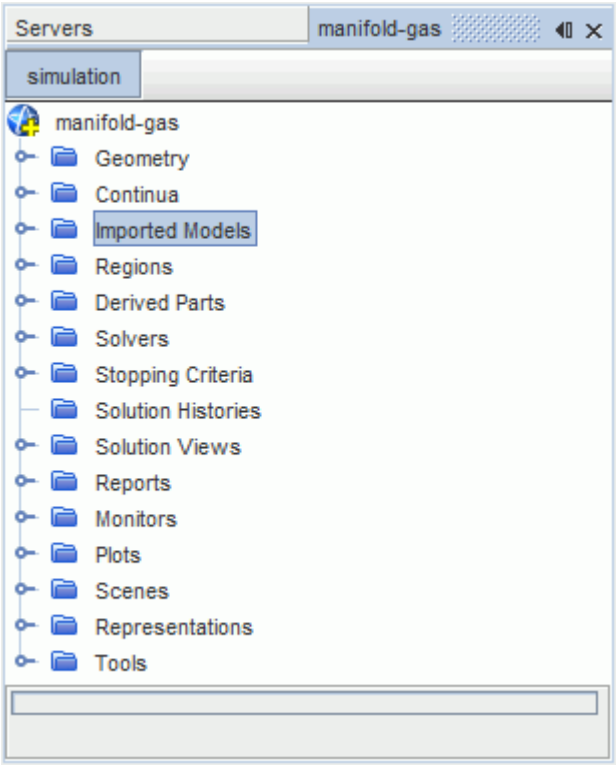
When importing a CAE model as a region and imported model, STAR-CCM+ creates a new region from each volume that is defined in the CAE model, and adds the **Imported Models** node, containing the relevant imported CAE model node, to the simulation tree. This import option allows you to access the Imported Models framework for file-based coupling operations.

To import a model as a region and imported model:

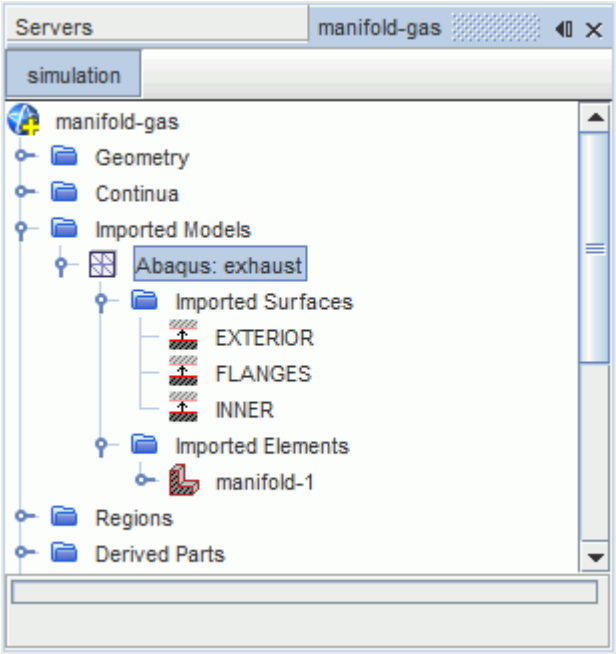
1. Follow the steps outlined in the section Importing a CAE Model.
2. In the *Import Options* dialog, set *Import Mode* to **Create New Region(s) and Imported Model(s)** then click **OK**.



A new top-level node, **Imported Models**, appears in the simulation tree.



3. Open the **Imported Models** node to visualize the imported CAE model node.
   The following image shows the imported CAE model node that is created when importing an Abaqus input file called `Manifold.inp`.



The imported CAE model node typically contains two subnodes, **Imported surfaces** and **Imported elements**. These nodes show the imported surfaces and elements that are defined in the CAE model file.

Note: When you import a CAE model containing a volume, STAR-CCM+ automatically extracts the free surface of the imported volume. When you import a model containing shells, STAR-CCM+ automatically locates the positive and negative side of each shell.

What Is the Imported Models node?

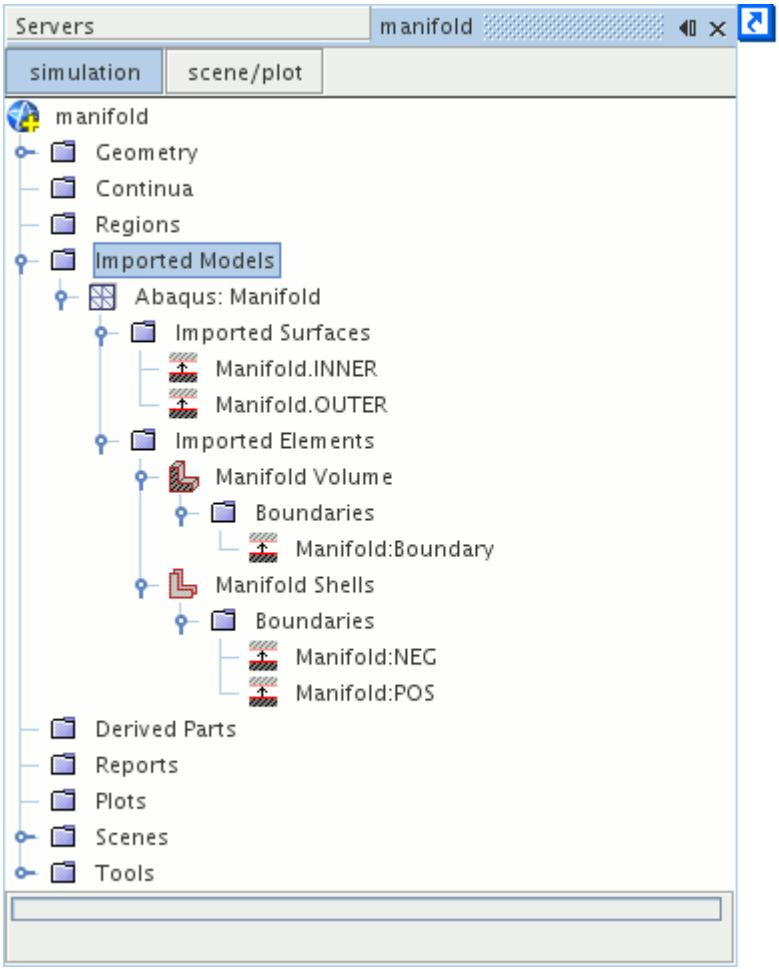
When the *Import Mode* is set to **Create New Region(s) and Imported Model(s)**, STAR-CCM+ creates a new region for every imported volume, shell, or beam, and boundaries for the imported surfaces. STAR-CCM+ also adds a new node called **Imported Models** to the simulation tree.

Every CAE model that is imported in the simulation is represented by a node within the **Imported Models** node. These imported CAE model nodes, located within the **Imported Models** node, provide an interface for file-based coupling.

The node layout for an imported CAE model is as follows:

- [Imported CAE model] node
  - Imported Surfaces node
    - Imported surfaces
  - Imported Elements node
    - Imported elements (includes shells, beams, and volumes)

The **Imported Models** node has no properties but is subject to common menu items.



Contents:

What Is the Imported CAE Model Node?
What Is the Imported Surfaces Node?
What Is the Imported Elements Node?
Imported Shell Elements
Imported Beam Elements
Imported Volume Elements

What Is the Imported CAE Model Node?

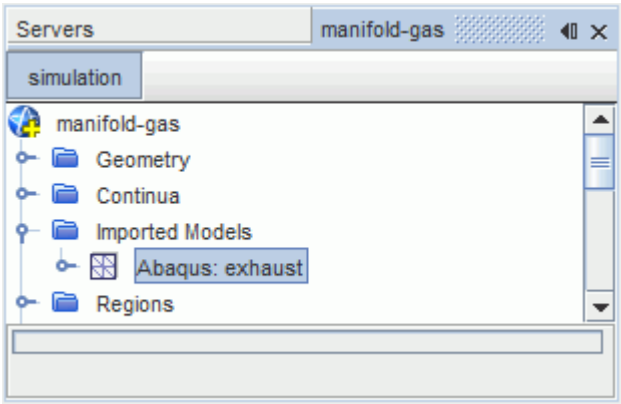
The imported CAE model node, located within the **Imported Models** node, provides an interface for file-based coupling.

From an imported CAE model node, it is possible to map solution data from the STAR-CCM+ model onto the CAE model, and export it in a format that can be read into the third-party software. You can also import solution data for a CAE model, computed by the third-party code, map the data onto the STAR-CCM+ model, and use the data to define boundary conditions, or as an input to the morpher.

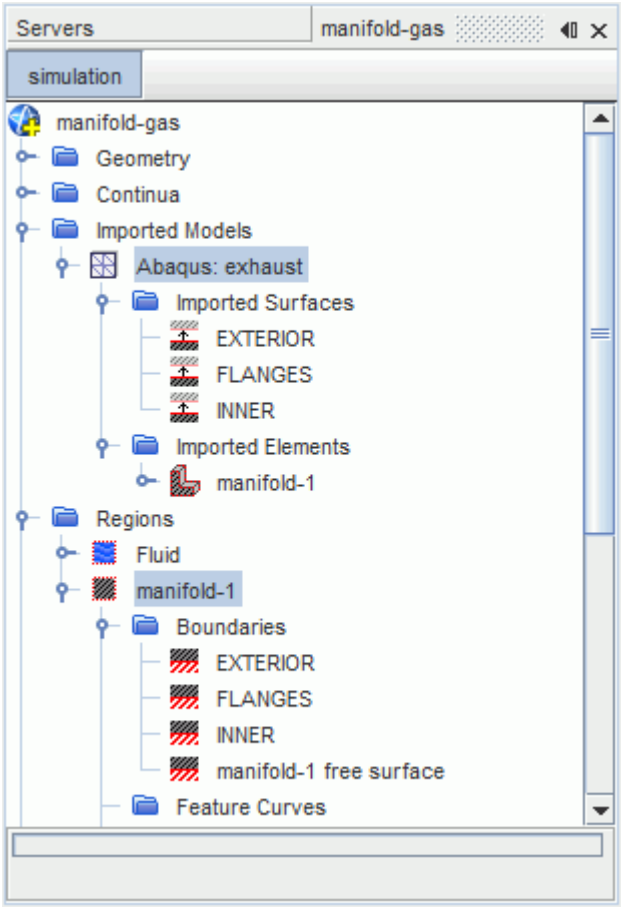
STAR-CCM+ automatically names the imported CAE model node according to the following convention:

```
<File Type>: <File Name>
```

For example, when the Abaqus input file `exhaust.inp` is imported, the corresponding imported CAE model node is named `Abaqus: exhaust`.



When a CAE model is imported, STAR-CCM+ recognizes geometrical elements that are defined in the input file, and creates corresponding volume, shell, or beam regions. The imported surfaces, volumes, shells, and beams are also listed under the relevant imported CAE model node.



The imported CAE model features are fully stored in STAR-CCM+ under the **Regions** and **Representations** nodes, whereas the imported CAE model node simply provides an interface for file-based coupling operations. After the import process, the imported CAE model nodes are connected to the corresponding regions. Deleting a region associated with an imported CAE model deletes the imported model itself. The regions created for the imported CAE models are subject to all the operations that apply to regions in STAR-CCM+, for example, transform operations such as rotation and translation, mesh reordering, and so on. When these operations are performed, the imported CAE model updates accordingly.

The imported CAE model node is used for file-based coupling, which is carried out using the pop-up menu. The node has no properties.

Imported CAE Model Pop-Up Menu

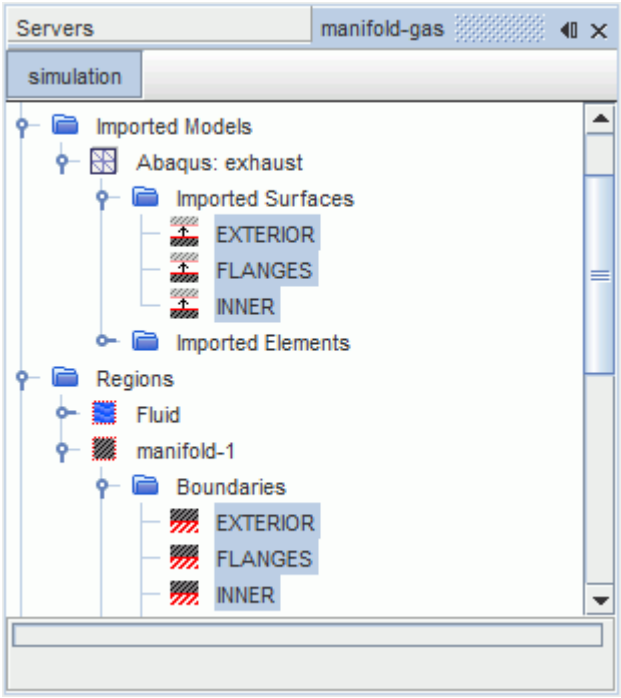
Import CAE Solution Data...	Imports data from an external file that is associated with the imported CAE model, such as an Abaqus `.odb` file.	
Map Data	**Surface Data...**	Maps surface data between CAE model surfaces or vertices and STAR-CCM+ boundary faces or vertices.

	Volume Data...	Maps volume data between CAE model vertices and STAR-CCM+ region cells or vertices.
	Beam Data...	Maps beam data between CAE model vertices (beam nodes) and STAR-CCM+ boundaries.
Export Mapped Data...	Exports mapped data to a separate file for use in an analysis with the other software.	
Deform Using Imported Surface Displacements	Deforms the CAE model mesh based on imported surface vertex displacements.	

What Is the Imported Surfaces Node?

When a CAE model is imported, STAR-CCM+ automatically recognizes the surfaces that have been defined in the CAE model, and defines corresponding boundaries.

Each imported surface is also represented by a node within the **Imported Surfaces** node.



Imported Surfaces Node Properties

Imported Surfaces	Specifies the number of surfaces imported into STAR-CCM+ (Read-only).

[Imported Surface] Properties

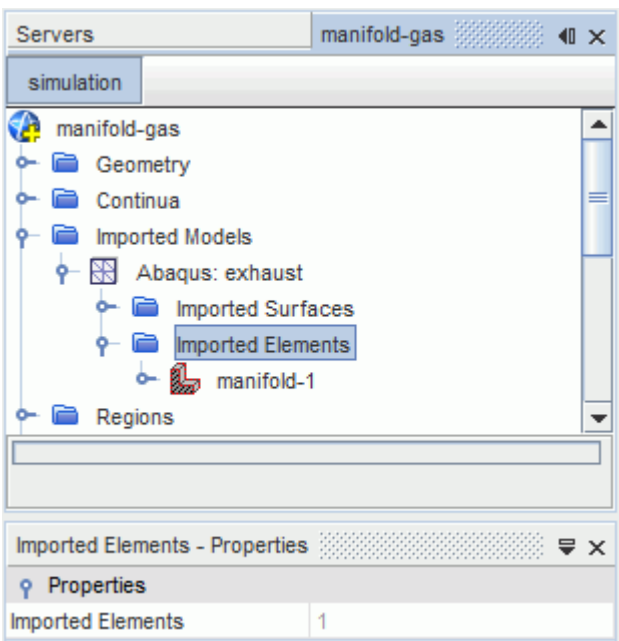
Index	Identifies the surface with a unique number (Read-only).
Boundaries	Specifies the region boundaries that are associated with the imported surface (Read-only).

The **[Imported Surface]** nodes can be selected in the relevant dialogs when mapping/exporting solution data from STAR-CCM+ to the third-party CAE package, or when importing/mapping CAE model solution data to the STAR-CCM+ model. These processes are used for file-based coupling and are carried out using the pop-up menu on the imported CAE model node.

What Is the Imported Elements Node?

When a CAE model is imported, STAR-CCM+ automatically recognizes the shell, beam, and volume elements that have been defined in the CAE model, and creates corresponding regions and boundaries.

Each imported shell, beam and volume element is also represented by a node within the **Imported Elements** node.



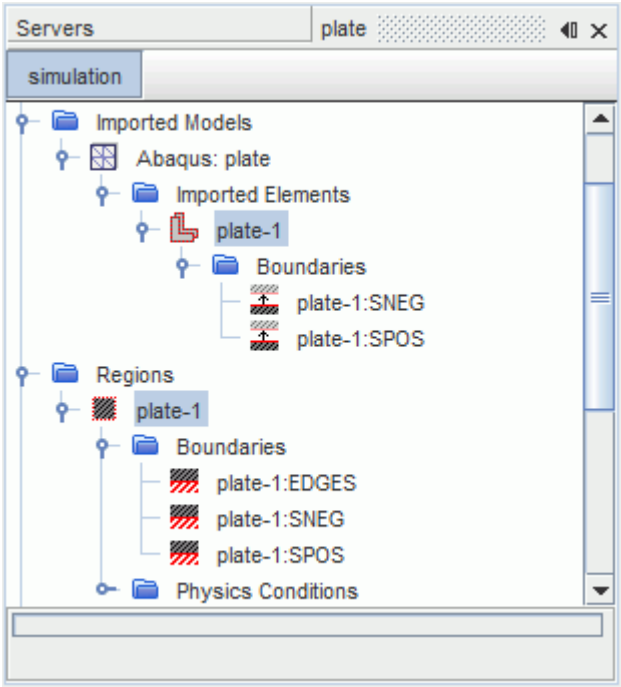
Imported Elements Node Properties

Imported Elements	Specifies the number of elements imported into STAR-CCM+ (Read-only).

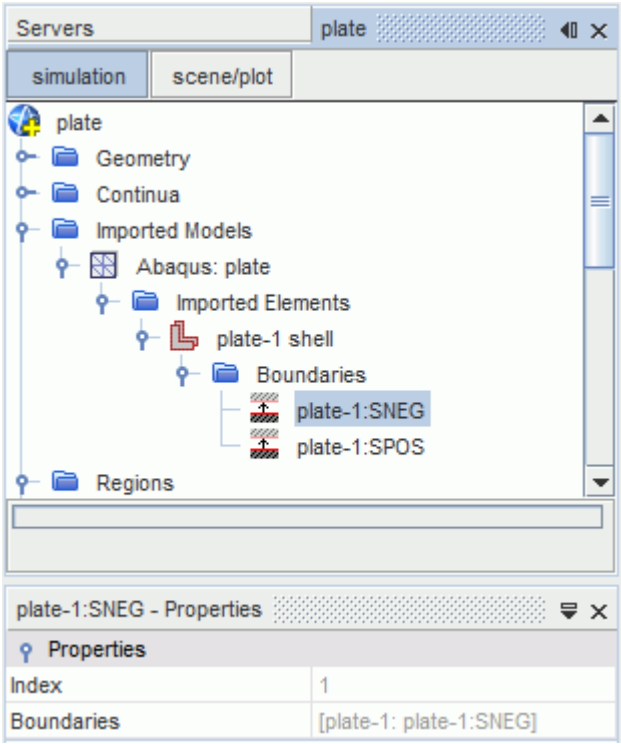
Imported Shell Elements

Shell elements have a positive and a negative side. When a CAE model is imported into STAR-CCM+, a shell element defines a region, and the surfaces of the shell element are stored as boundaries within that region.

Each shell element is represented by a node within the **Imported Elements** node.



The shell element nodes contain a **Boundaries** sub-node, in which the two surfaces of the shell are listed.



The surface that corresponds to the positive facing side of the shell are suffixed with **POS**, **SPOS**, or **Front**, depending on the input format. Similarly, the surface that corresponds to the negative facing side of the shell are suffixed with **NEG**, **SNEG** or **Back**. The positive direction is found by applying the right-hand rule to the vertex numbering on a face. If the fingers of your right hand are wrapped around the face in the direction in which vertex numbers increase, then your thumb points in the positive direction.

The shell surfaces can be used for file-based coupling in the same way as imported surfaces. When mapping onto shell surfaces, it is essential to ensure that you map to the correct side of the shell. If in doubt, activate the **Check Source/Target Face Normals** in the *Nearest Neighbor Search Constraints* group box, that is located on the *Map Imported Model Data* dialog. You can also visualize the direction of normals on the different surfaces. The normals of the source and target faces face each other.

[Imported Shell] Properties

Index	Identifies the shell with a unique number (Read-only).
Regions	Specifies the region that is associated with the imported shell (Read-only).

[Imported Shell Surface] Properties

Index	Identifies the shell surface with a unique number (Read-only).
Boundaries	Specifies the region boundaries that are associated with the imported shell surface (Read-only).

Imported Beam Elements

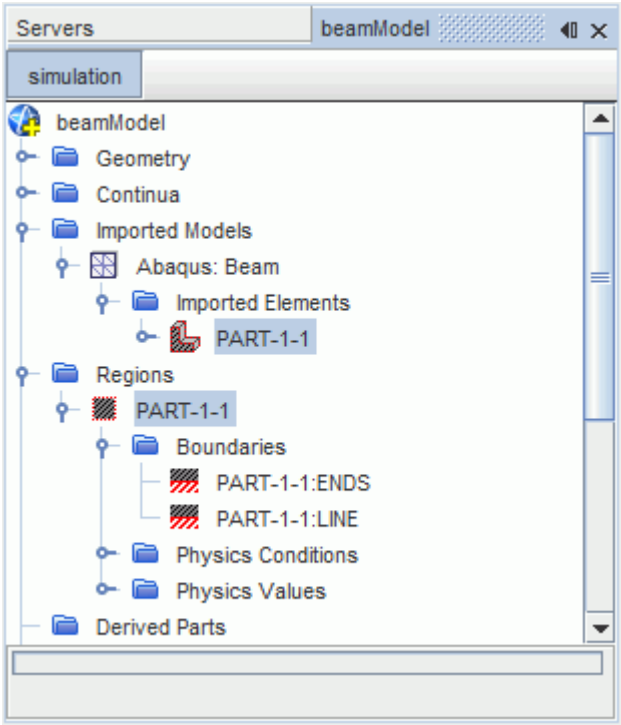
STAR-CCM+ supports the import of linear beam elements from Abaqus (.inp), Nastran (.bdf and .dat), and ANSYS (.cdb).

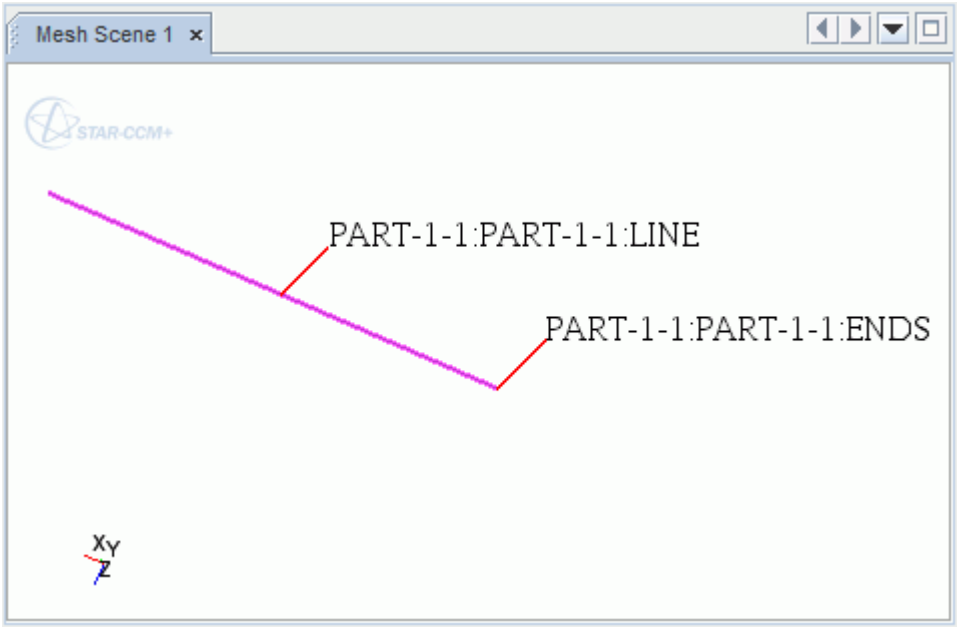
Beam elements are one-dimensional objects in three-dimensional space, and are often used to model 3D structures in which one dimension is greater than the other two dimensions. This approximation is based on the fact that the relevant physical quantities are mainly functions of the position along the dominant dimension, that is, the length of the structure. The cross section of a beam element defines the beam profile and its dimension is negligible compared to the beam length.

Beam elements have a two or three node representation, depending on the order of interpolation used for its finite element formulation. A beam element deforms under the application of either a bending moment, axial force, torque or a combination of these loads.

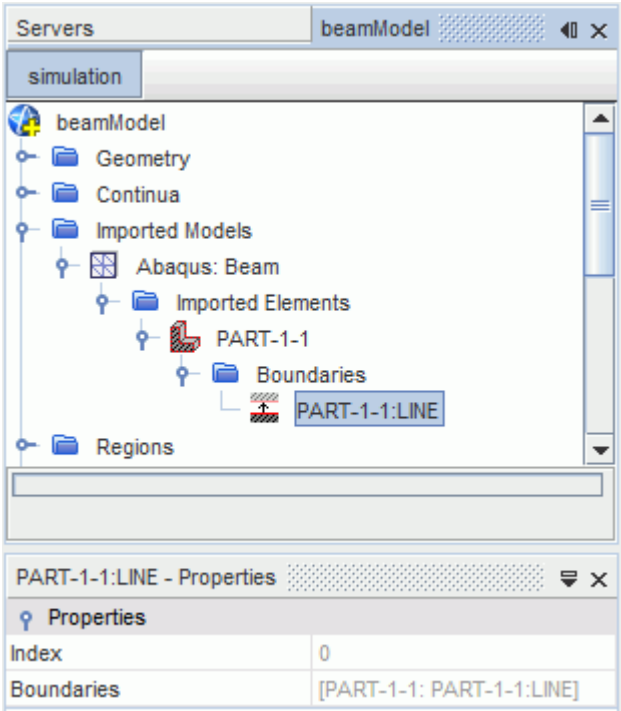
When a CAE model is imported into STAR-CCM+, the body and ends of a beam element are stored as two individual boundaries within the relevant region node. The name of the boundary that corresponds to the beam body is suffixed with **LINE**; the name of the boundary that corresponds to the beam ends is suffixed with **ENDS**.

Each beam element is also represented by a node within the **Imported Elements** node.





The beam element nodes contain a **Boundaries** sub-node, which contains a single boundary. This boundary represents both the ends and body boundaries that are located under the associated region node.



It is possible to map fluid traction data (pressure and shear forces) from the wetted surface in STAR-CCM+ onto the imported beam element as forces, and export the mapped field functions to an external file that can be read into the third party software. It is also possible to map nodal displacement and rotation data from the imported beam to the wetted surface in STAR-CCM+. Displacement and rotation data can then be used by the mesh morpher to deform the model. These file-based coupling operations are carried out using the pop-up menu for the imported CAE model node.

[Imported Beam] Properties

Index	Identifies the beam with a unique number (Read-only).
Regions	Specifies the region that is associated with the imported beam (Read-only).

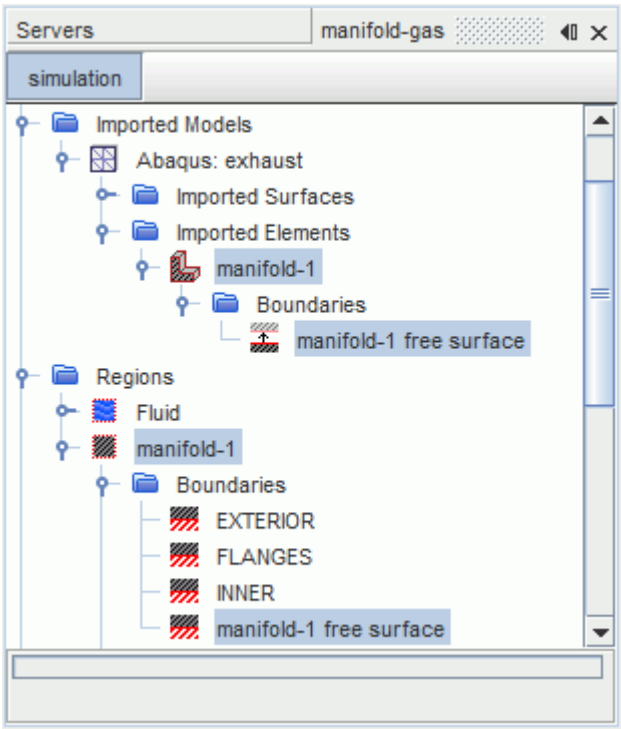
[Imported Beam Boundary] Properties

Index	Identifies the beam boundary with a unique number (Read-only).
Boundaries	Specifies the region boundaries that are associated with the imported beam (Read-only).

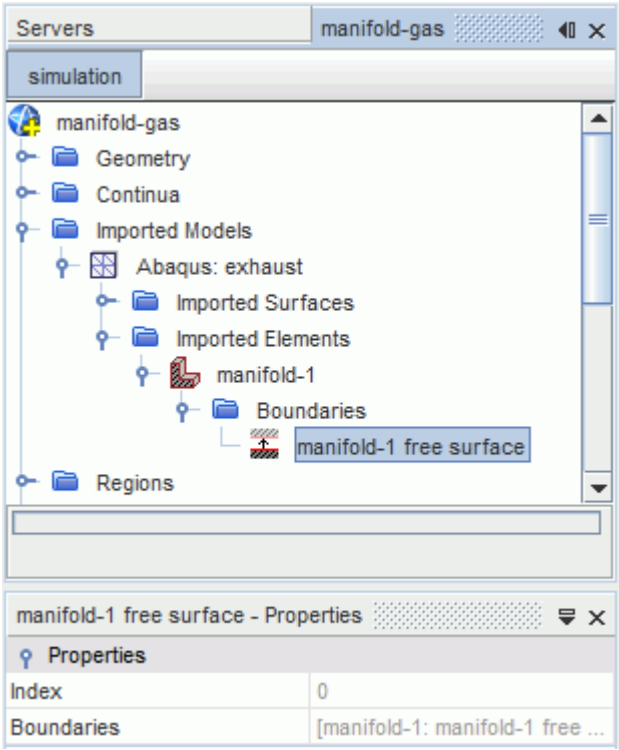
Imported Volume Elements

When a CAE model is imported into STAR-CCM+, volume elements are stored as regions under the **Regions** node. The extracted surface of the volume element is stored as a boundary under the relevant region **Boundaries** node.

Each volume element is represented by a node within the **Imported Elements** node. The name of the node depends on the names that are specified in the input file, for example **manifold-1** in the image below.



Volume element nodes also contain a **Boundaries** node, under which the extracted surface of the volume element appears.



It is possible to map region temperature data from STAR-CCM+ onto an imported volume via the imported CAE model node, and export this temperature to an external file. This file can be read into the third-party software and used to define a temperature field throughout the volume. This operation is used for file-based coupling and carried out via the pop-up menu for the imported CAE model node. Upon import, STAR-CCM+ automatically extracts the free surface of a volume element. The free surface can be used in the same way as an imported surface to import, export, and map solution data, and also for visualization.

[Imported Volume] Properties

Index	Identifies the volume element with a unique number (Read-only).
Regions	Specifies the region that is associated with the imported volume (Read-only).

[Imported Volume Surface] Properties

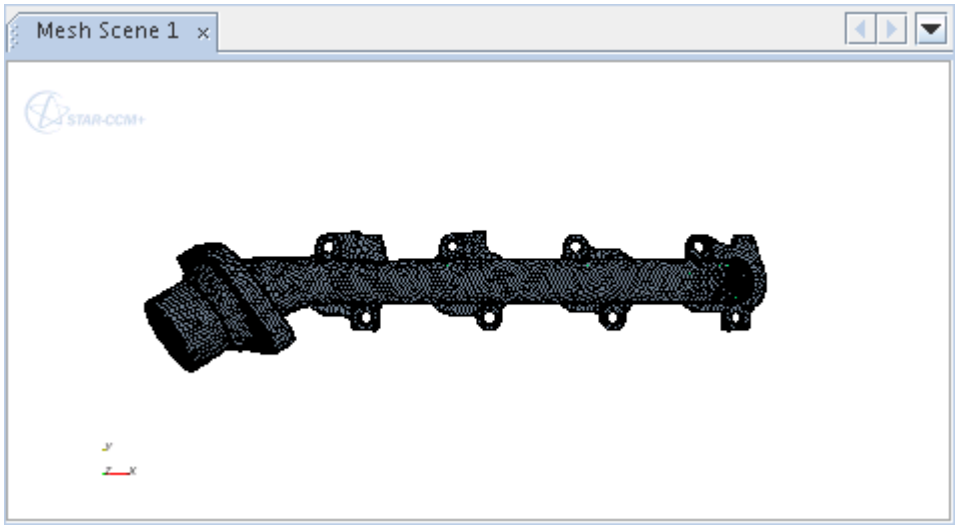
Index	Identifies the volume surface with a unique number (Read-only).
Boundaries	Specifies the region boundaries that are associated with the extracted surface (Read-only).

Visualizing an Imported CAE Model

To display the surface mesh geometry of the imported CAE model:

1. Right-click the **Scenes** node and select **New Scene** > **Mesh**.

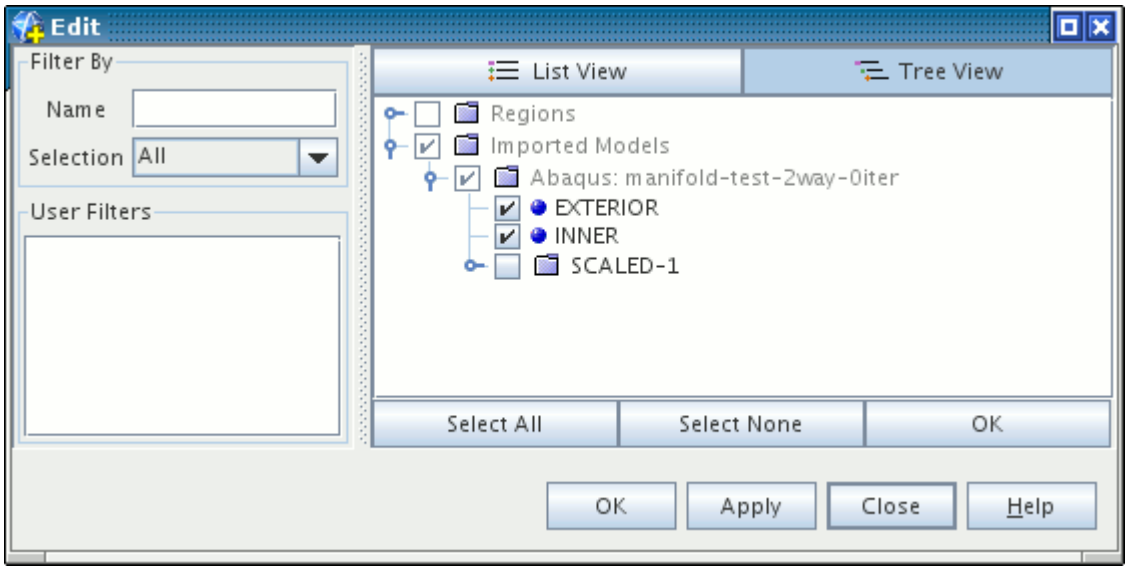
   A new scene appears with the mesh geometry appearing.



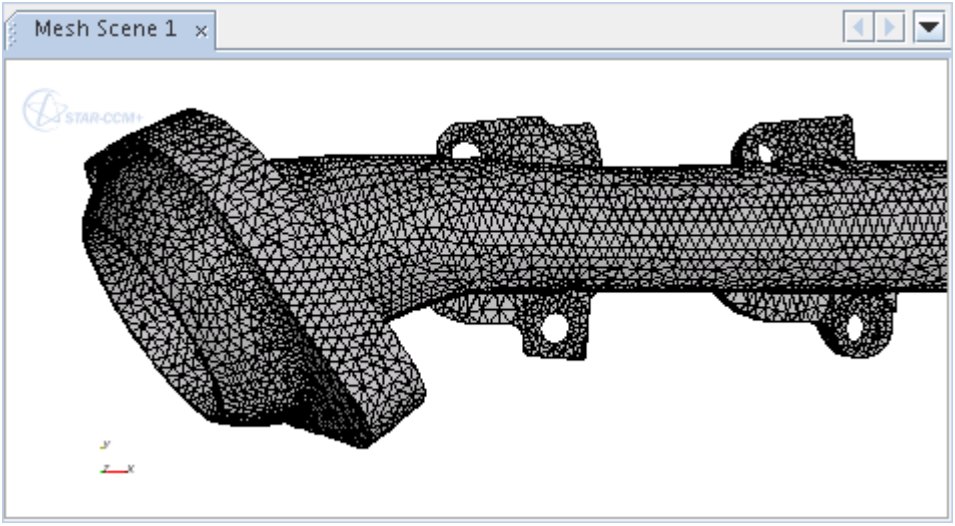
2. Select the **Mesh Scene 1** > **Displayers** > **Mesh 1** > **Parts** node.

Depending on the type of data you have imported, there may be parts in addition to the surfaces that you want to display.

3. In the *Properties* window, click in the right half of the *Parts* property.
4. In the selection dialog:
   a) Remove all the parts that may currently be selected by clicking **Select None**.
   b) Expand the **Imported Models** node and select the CAE model elements that you wish to display (for example, **EXTERIOR** and **INNER** in the screenshot below), then click **OK**.



The imported surface appears in the mesh scene.



Contents:

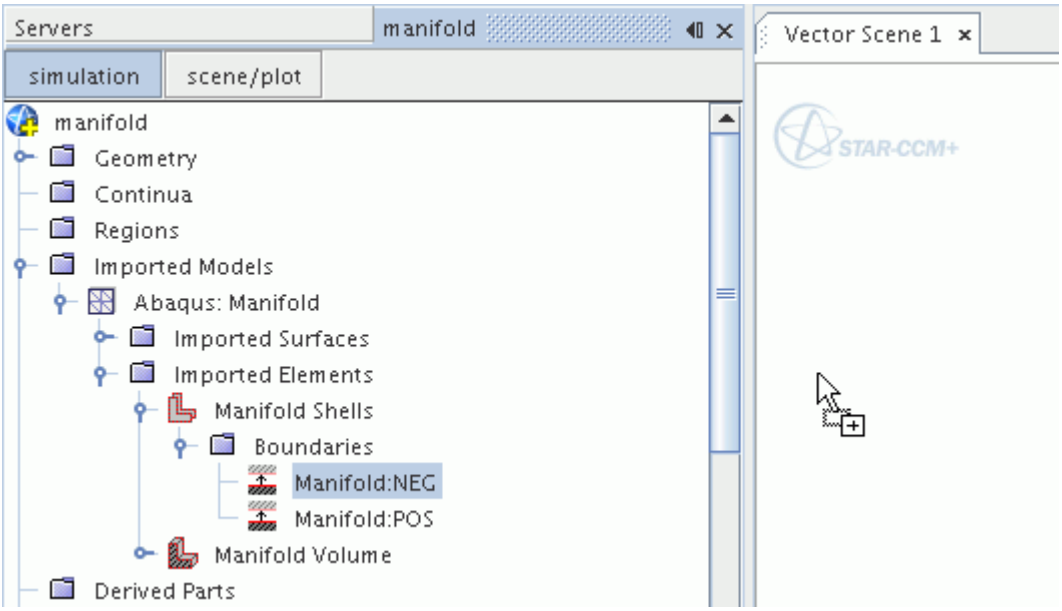
Visualizing Shell Normals

Visualizing Shell Normals

When shells are imported in a CAE model, STAR-CCM+ automatically locates the positive and negative sides, or front and back, of each shell. It is often useful to be able to check which side of the shell is which, and this can be achieved by visualizing the face normals of each shell side.

Start by adding one of the shell sides to the vector displayer of a vector scene:

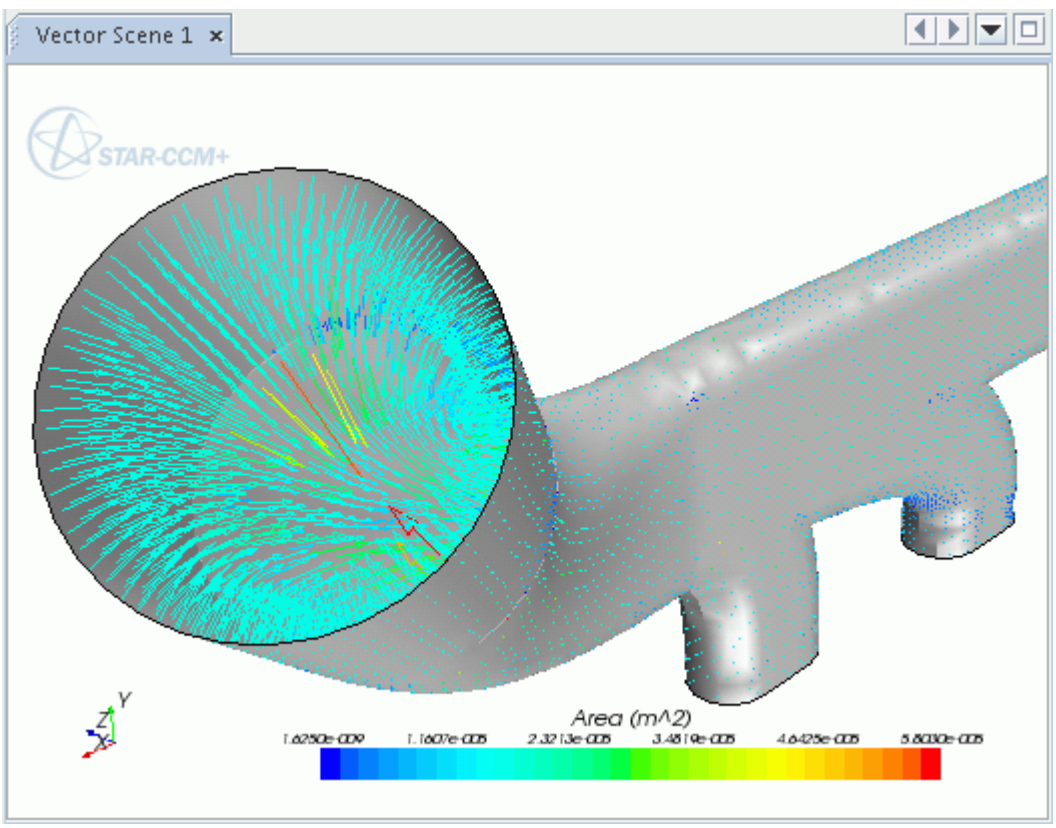
1. Create a vector scene.
2. Click-and-drag the **Imported Elements** > **shell** > **Boundaries** > **shell side** node onto the scene.
3. Release the mouse button and select **Add to Vector 1** in the pop-up menu.
   In the example below, the negative side of the manifold shell, **Manifold: NEG**, is added to the scene.



4. In the scene, set the *Vector Field* to **Area**.

   Vector glyphs are displayed on the shell surface. The direction of the vectors shows which side of the shell the surface is on.

   For example, in the following screenshot, the negative surface is on the inside of the manifold shell. Had you displayed face normals on the positive surface of the shell, the arrows would be facing outward, indicating that the positive surface is on the outside of the manifold shell.



In the above screenshot, the gray surface was activated by adding the same negative shell surface, **Manifold:NEG**, to a geometry displayer and enabling surfaces on the displayer.

Exporting Mesh and Solution Data

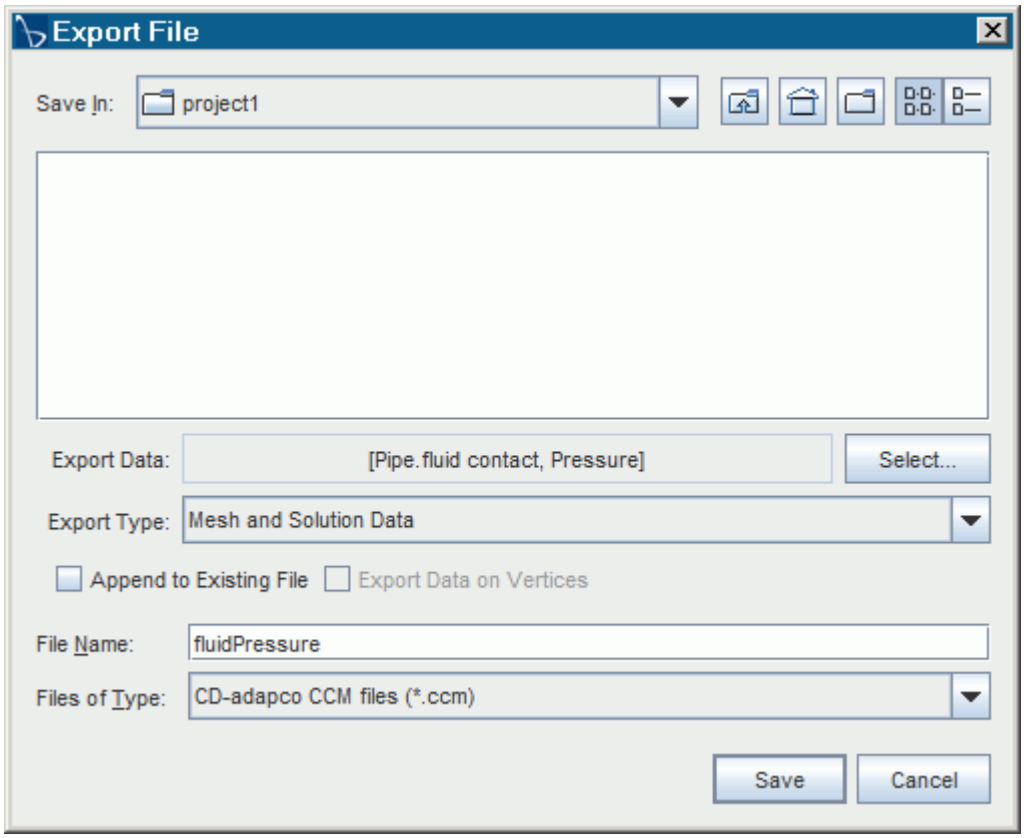
This section outlines the steps for exporting STAR-CCM+ mesh and solution data to formats that can be used in other CAE packages.

Separate sections provide instructions and format information for exporting the following:

- 3D-CAD geometry
- Mesh surfaces

1. Select **File** > **Export ...**

   This menu item opens a modified form of the operating system-specific standard *Save To File* dialog. In this dialog, you can specify the file type, the name of the new file, and which types of data to export to the file.



2. Choose the scope of the data to select from the *Export Type* drop-down list.

   The following options are available:

   - **Mesh Only**

- **Mesh and Solution Data**
- **Solution Data Only**

3. Use the *Files of Type* drop-down list to choose the output format.

   The currently supported export formats include:

Type of File	Type of Data Exported	Further Information
CCM (*.ccm)	Mesh data and solution data	Exports volume mesh and solution data. If you select the CCM format, you can activate Append to existing file to add the exported information to a .ccm file that you select. For example ESI/VA One, FFT/Actran, and LMS/Virtual.Lab (used in noise propagation cases), can read this format. When exporting a user-defined field function, make sure that the function property, *Function Name*, only contains characters of the following types: • A-Z, a-z, 0-9 • special characters other than the forward slash "/" (that is, ASCII values from 32 to 126, except for 47) **Note:** In CCM files, mesh indices are written as 32-bit integers. This approach sets a limit to the size of meshes that you can export using the CCM format. Estimated upper limits are ~700 million cells for trimmed meshes and ~350 million cells for polyhedral meshes.
CGNS, CFD General Notation System (HDF5) (*.cgns)	Mesh data and solution data	Exports volume mesh and solution data. Open format, CFD General Notation System. Appending is allowed. When exporting polyhedral volume meshes to CGNS format, regions and boundaries can import into some third-party codes as a single region and single boundary. This limitation only applies to meshes containing polyhedral cells.
CGNS, CFD General Notation System (ADF) (*.cga)		
Siemens Virtual.Lab Custom CGNS (HDF5) (*.cgns)	Mesh data and solution data	Exports volume mesh and solution data in a format that is suitable for LMS Virtual.Lab. This option is intended for exporting transient pressure data to Siemens Virtual.Lab, for aeroacoustic analysis.

Type of File	Type of Data Exported	Further Information
		As LMS Virtual.Lab does not support polyhedral meshes, do not export regions with polyhedral meshes to this file format. However, you can export boundaries with polygonal surface meshes. Before exporting to CGNS, STAR-CCM+ merges the boundaries and triangulates any polygonal faces. For automatic export of the transient data, set *Trigger* to either **Time Step** or **Delta Time**, as required. When you export solution data, but not mesh data, make sure that the mesh does not change during the analysis. If the simulation requires mesh repartitioning, choose to export both mesh and solution data. STAR-CCM+ automatically appends the exported data to the file that is saved during the first export. See Automatically exporting Mesh and Solution data at Regular Intervals.
Ensight Gold (*.case)	Mesh data and solution data	Exports surface mesh and solution data—from cells and vertices—although, you can also export the solution on the surface. However, when a region is selected, only surface mesh is exported. Appending to the file is allowed.
FIELDVIEW (*.fvuns)	Mesh data and solution data	Exports surface mesh and solution data. When exporting to this format from a parallel session, STAR-CCM+ includes grid partition information in the exported file. A parallel session of FIELDVIEW can use this grid partition information to distribute the data among the processes that are available to it.
Tecplot 360 (Version 2008 onwards) (*.plt)	Mesh data and solution data	Exports surface mesh and solution data—from cells and vertices—although, you can also export the solution on the surface. However, when a region is selected, only surface mesh is exported. Appending to the file is not allowed.
Boundary Shell (*.cel)	Multiple files	Creates files with `.cel` and `.vrt` formats. A third format, `.daten`, appears when you specify data by selecting field functions, boundaries, and/or part surfaces.

Type of File	Type of Data Exported	Further Information
		Note: When exporting polyhedral meshes to `Boundary Shell (*.cel)` format, the surface mesh is triangulated before exporting.
STAR-CCM+ **Boundary Data** (*.sbd)	Mesh data and solution data	Exports surface mesh and solution data. Post-Processing data. Appending to the file is not allowed.
RadTherm Patran Neutral (*.trn)	Mesh data and solution data	Exports surface mesh and solution data. RadTherm/Patran recognizes this format. Exports heat transfer coefficient and reference temperature data.
Surface FFT (*.trn)	Mesh data and solution data	Exports surface mesh and time history of the solution.
Nastran load history (*.bdf)	No mesh data—only STAR-CCM+ load data	Intended for writing time history of fluid load data for file-based FSI coupling with Nastran. For use with Auto-Export.

4. To specify the exported data, click **Select...** next to *Export Data*.
5. Use the *Select Objects* dialog to select objects for export. The objects that are available for selection depend on the specified *Export Type* and output format (that is, *Files of Type*).

   More than one variable can be selected using `<Ctrl><Click>` to select discontinuous items and `<Shift><Click>` to select a continuous range of items. To deselect an item in a list, use `<Ctrl><Click>`.

   - You can export solution data by selecting the corresponding field functions.
   - You can export mesh data by selecting the corresponding objects (that is, regions, boundaries, parts, or parts surfaces, based on the output format). The `Boundary Data (*.sbd)` and `RadTherm Patran Neutral (*.trn)` output formats only allow you to export boundaries. The `Ensight Gold files (*.case)` or `Surface fft (.trn)` allow you to export mesh data available on derived parts. The `Boundary Shell (*.cel)` output format also allows you to export mesh data available on parts and part surfaces.
6. Enter the output *File Name* and click **Save**.

   It is also possible to export automatically at regular intervals.

Contents:

Automatically Exporting Mesh and Solution Data at Regular Intervals
Automatically Exporting Mesh and Solution Data at Regular Intervals

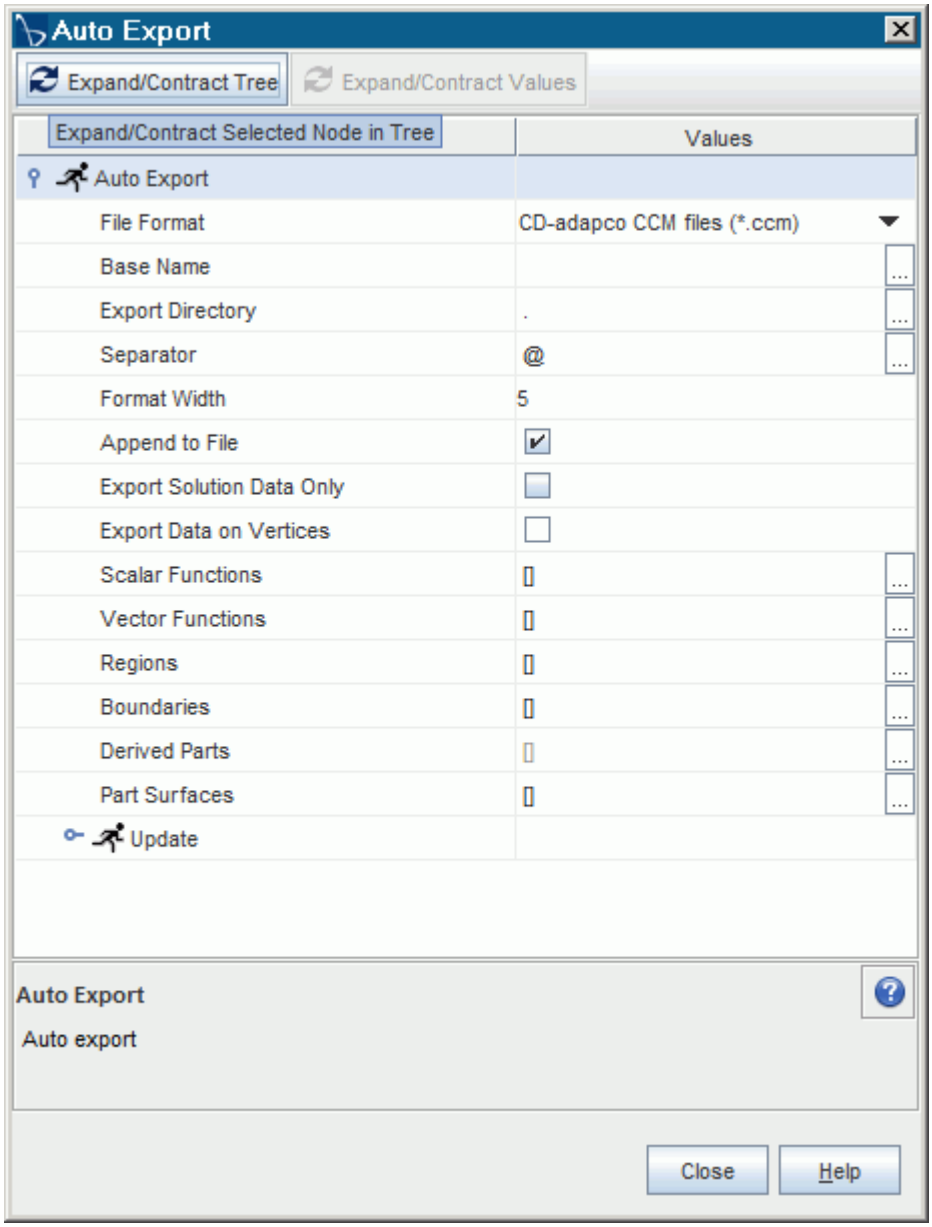
Automatically Exporting Mesh and Solution Data at Regular Intervals

When running a long simulation (either interactively or in batch mode) it is useful to enable the auto export feature. This feature causes STAR-CCM+ to export the solution data automatically at regular intervals during the run, when the **Export...** command is disabled. This section includes a description of the properties of the *Auto Export* dialog.

To set up the auto-export of solution data:

1. Select **File** > **Auto Export...** from the menu.

   The *Auto Export* dialog appears.



2. Expand the **Auto Export** > **Update** node and activate *Enabled* .

To specify the file format that you want to export to:

3. Select the file format from the drop-down list in the Values column of the *File Format* property.
4. Specify the *Base Name*.

   The base name is the prefix of the name of the file that you want to export your data to.
5. Specify the *Export Directory* if necessary.

   This directory is the location of the file that you want to export your data to. The default path is the same as the path of your simulation file. Clicking ... (**Custom Editor**) activates a dialog similar to the *Save* dialog. You can use this dialog to navigate to a new location for the file.
6. Enter the type of separator that should be used in the *Separator* property.

   By default, when STAR-CCM+ saves copies of the simulation file, it places a `@` character to separate the base file name and the number of the trigger event. If the use of this character in this manner causes any problems with your system, you can enter a different one as the separator. If you attempt to leave this field blank, STAR-CCM+ puts the `@` character there by default.
7. If you want to add the exported data to an existing file, activate *Append To File*.
8. If you prefer not to export the mesh data, activate *Export Solution Data Only*.
9. To export vertex data, activate *Export Data on Vertices*.

   This option is only available when your export includes solution data, and when you are exporting to an Ensight Gold file.

To select the scalar values that you want to export:

10. Click ... (**Custom Editor**) next to *Scalar Functions*.
11. In the *Auto Export - Scalar Functions* dialog, select the desired functions and click **OK**.
12. Using the same method as defining the scalar functions, you can select vector functions, regions, boundaries, derived parts, and parts surfaces.
13. Select the trigger type from the drop down menu of the *Trigger* property.

    This property specifies how the auto export operation is triggered. Unless you are running a transient analysis, leave the setting at `Iteration`. For more information, see Trigger Types.
14. To control the number of intervals that pass between auto exports, expand the **Trigger** > **Frequency** node and enter the value that you want. See Trigger Properties.
15. Click **Close**.

STAR-CCM+ automatically exports during the next run at the intervals you specified. To deactivate this feature, simply reopen the *Auto Export* dialog and deactivate *Enabled*.

Contents:

Auto Export Properties
Exporting Region and Boundary Data
Auto Export Properties
Exporting Region and Boundary Data

Auto Export Properties

You can use the following properties to activate and customize this feature:

File Format

Selects the format of the file from a drop-down list.

Base Name

The prefix of the file name to use when exporting your data.

Export Directory

The directory that contains your file with exported data.

Separator

Specifies the character that appears between the base file name and the number of the trigger event (`@` by default).

Format Width

Width of the iteration, time, or step number in the formatted file name.

Append to File

When activated, adds the exported information to an existing file. This file would be the one you specify in the *Base Name*, *Export Directory* and *File Format* properties.

This option works for `.ccm`, Ensight and `.trn` file formats only.

Export Solution Data Only

When activated, exports solution data without any mesh data.

Export Data on Vertices

When activated, exports vertex data. This option is only available when your export includes solution data, and when you are exporting to an Ensight Gold file.

Scalar Functions

Determines which scalar values to export.

Vector Functions

Determines which vectors to export.

Regions

Selects regions to export. See Exporting Region and Boundary Data.

Boundaries

Selects boundaries to export. See Exporting Region and Boundary Data.

Derived Parts

Selects derived parts to export. See Accessing Solution Data through Derived Parts.

Part Surfaces

Selects part surfaces to export.

Update

When expanded, this property makes available the controls listed below.

Enabled

When activated, enables the auto export feature.

Trigger

Provides choices for triggering the automatic export. See Trigger Types.

Iteration Frequency

Controls the number of trigger events between exports. See Trigger Properties.

The settings are saved in the simulation database.

Exporting Region and Boundary Data

The following rules apply when exporting region and/or boundary data:

- If no regions and no boundaries are selected, data for all the regions (including the boundaries) in the simulation is exported.
- Selecting a region automatically selects the boundaries that are associated with that region. Data for the selected region (including the associated boundaries) is exported. This action has the same effect as manually selecting a region and the boundaries in that region.
- Selecting a boundary does not automatically select the region that is associated with that boundary. Data for the selected boundary is exported.
- Selecting boundaries without also selecting regions results in a `.ccm` file that contains only boundary data. This option is not compatible with pro-STAR.

File-Based Coupling

STAR-CCM+ can be coupled with external packages using the file-based coupling method. This allows you to use the STAR-CCM+ solvers to obtain part of the solution, and another CAE code to obtain another part of the solution.

This technique can be used in cases such as fluid-structure interaction (FSI), where STAR-CCM+ would solve for the fluid, a third party code would solve for the structure, and data would be exchanged between the two codes at the solid-fluid interface.

Contents:

Understanding File-Based Coupling
Mapping STAR-CCM+ Solution Data to a CAE Model
Exporting Mapped Data
Mapping and Exporting Transient Surface Data
Importing CAE Solution Data
Mapping CAE Solution Data to a STAR-CCM+ Model
Exchanging Heat Transfer Coefficients
Working with Specific CAE Products
Working with Abaqus
Working with ANSYS
Coupling with NASTRAN
Working with TAITherm
Working with es-ice or pro-STAR
Working with Ensight
Working with the STAR-CCM+ Boundary Data (.sbd) Format
Working with the Visualization Toolkit (VTK) Format
Coupling with RCAS

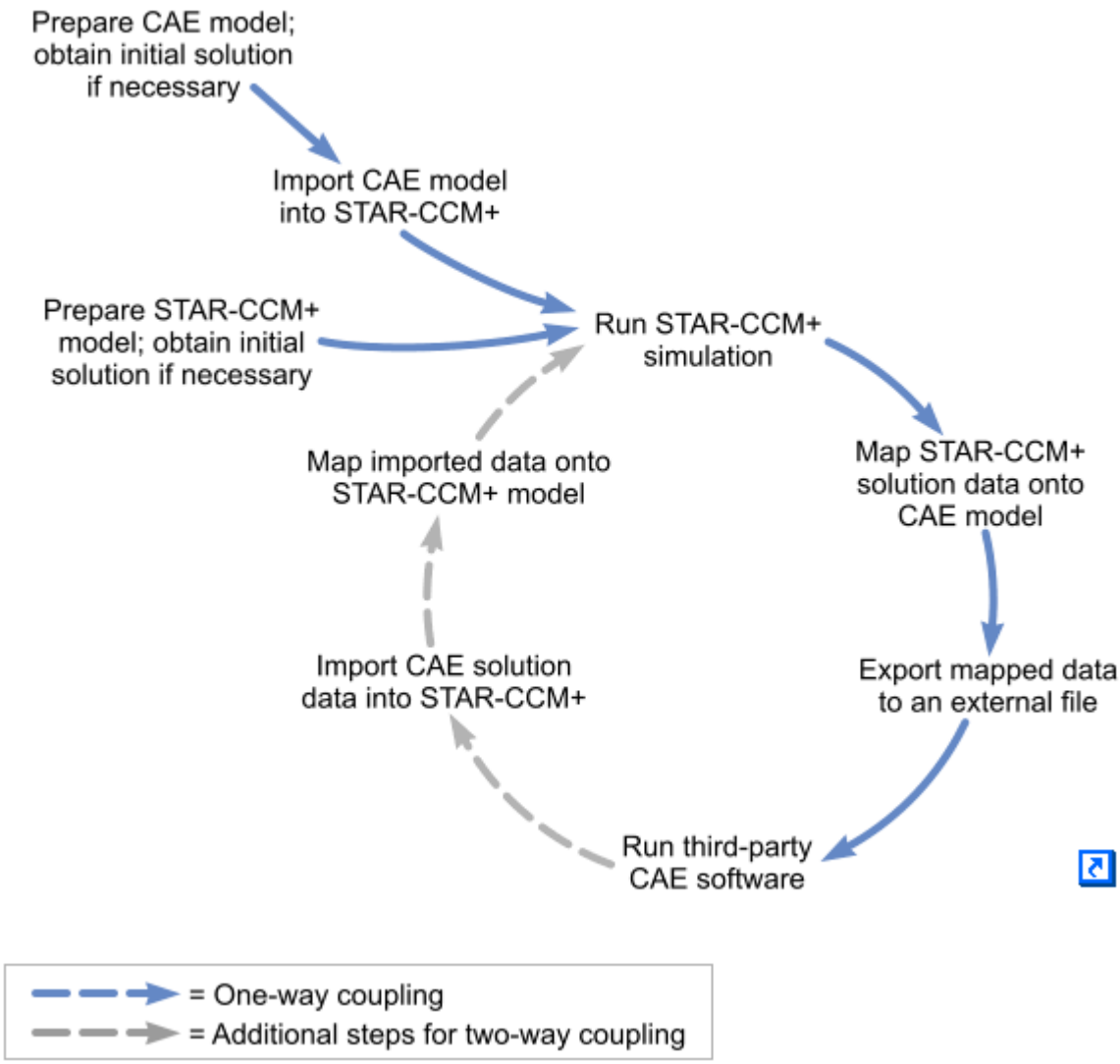
Understanding File-Based Coupling

File-based coupling is a method used to couple STAR-CCM+ with another CAE code via external files stored in the computer memory.

The method allows you to obtain a solution using separate CAE codes to solve for each part of the model, such as the fluid and solid domains in an FSI problem. The powerful set of tools available in STAR-CCM+ allow you to import a CAE model into STAR-CCM+ and carry out a one-way or two-way coupled simulation.

For one-way coupling, solution data from STAR-CCM+ can be mapped onto the CAE model and exported in a format that can be read into the third-party application. The exported data can then be used to specify boundary or operating conditions in the CAE model. An example of this is using STAR-CCM+ to compute the conjugate heat transfer solution in the fluid and solid domains of a model, then exporting the solid region temperature solution to an external file. This file can then be read into a third party code, and used to calculate the thermal stresses in the solid. An example of the typical workflow for one-way coupling is shown in the diagram below (solid blue arrows).

For two-way coupling, you can continue by importing solution data from the third-party code into STAR-CCM+. This data can then be mapped onto the STAR-CCM+ model and used to define a boundary condition in the simulation, or to update the mesh. An example of two-way coupling is using STAR-CCM+ to compute the pressure and wall shear stress at a fluid-solid interface, using this data to define a traction load on the FSI surface in the CAE model, and calculating the resulting displacements using the third-party code. The nodal displacements in the solid model can be imported into STAR-CCM+ and used as an input to the mesh morpher. The STAR-CCM+ simulation can then be run using the updated geometry, and the whole process can be repeated until the solution has converged. The typical workflow for two-way coupling is an extension of that for one-way coupling. The additional steps for two-way coupling are shown in the diagram below (dashed gray arrows).



An important part of file-based coupling is the mapping process, and this is discuss in greater detail in the next section.

Contents:

Understanding Data Mapping

Understanding Data Mapping

Mapping is the process of interpolating a set of data between two separate models. For example, pressure data from a fluid domain can be mapped onto the corresponding surface of a solid domain and used as a traction load.

Mapping data is an important part of the file-based coupling method, as it allows you to use data computed in STAR-CCM+ to specify a boundary or field condition in a third-party CAE code. Similarly, you can use data computed in a third-party code to set boundary conditions in the STAR-CCM+ model. Typically data is mapped at an interface between two models, and this may involve interpolating data between two surfaces, a

surface and a set of vertices, or between a region and a volume. Mapping presents several challenges to the user, such as how to overcome differences in mesh density or topology at the interface between the two models, or how to preserve the accuracy of the solution. These challenges, along with the type of mapping algorithms used in STAR-CCM+ and the typical workflow, are discussed below. For details on how to carry out mapping operations, refer to the following chapters:

- Mapping STAR-CCM+ solution data to a CAE model
- Mapping CAE solution data to a STAR-CCM+ model

All of the formats in STAR-CCM+ file-based coupling use the same core interpolation algorithms to map data from one surface to another. In the current version of STAR-CCM+ the accuracy of the mapping techniques has been improved from zeroth-order to at least first-order. Mapping in STAR-CCM+ is carried out by performing operations via the object tree as described below.

Mapping Operations

A key operation in coupling with imported models is to map data between STAR-CCM+ boundaries and/or regions and imported surfaces and/or volumes. It is also possible to map data between STAR-CCM+ boundaries and imported beam curves. The pop-up menus for imported surfaces, beams and volumes present several map actions. Each of these map actions creates a Data Mapper object that contains the mapping functionality and is listed under the **Tools** > **Data Mappers** node for you to modify or reuse. Depending on whether the map action is associated with surfaces, beams, or volumes, a Surface Data Mapper, Beam Data Mapper or Volume Data Mapper is created. For more information, see Using Data Mappers.

The map action dialog (*Map Imported Model Data*) allows you to choose a STAR-CCM+ boundary/region, an imported surface/volume, and a field function to map between the source and target boundaries/regions. In addition, you can specify certain parameters to control the underlying data mapper. These options include proximity tolerance and normalcy tolerance (only for the surface data mapper). For more information, see Setting Data Mapper Properties. Your selections from the map action dialog are automatically populated in the created data mapper object, thereby allowing you to achieve the same mapping either by creating mappers as described here or by manually creating them. Moreover, in case you perform subsequent mappings between the same surface or volume pairings, the interpolation stencil will be reused, thereby optimizing the computational time.

Interpolation Techniques

The interpolation techniques are based on either the least squares method or shape function method. The choice is based on whether the source data is face-centric or node-centric. For instance, consider a conjugate heat transfer (CHT) application in which the thermal analysis of the solid is being done using ABAQUS and the flow/thermal analysis of the fluid is being performed using STAR-CCM+. The coupling of these analyses takes place in two steps:

1. Mapping the heat transfer coefficient and reference temperature from the STAR-CCM+ fluid boundary to the ABAQUS solid surface.
2. Mapping the surface temperature from the ABAQUS solid surface to the STAR-CCM+ fluid boundary.

In Step 1, the STAR-CCM+ boundary is the source surface and the source data is face-centric. The target points are the ABAQUS face centers. In this case, we use least squares interpolation to map boundary face-centered data to ABAQUS face centers. In Step 2, the solid surface temperatures are imported from ABAQUS. In the above-mentioned CHT case, the ABAQUS surface is the source surface and the source data

resides on the nodes. The shape functions for the ABAQUS elements are used to interpolate functions to map data from the ABAQUS nodes to STAR-CCM+ boundary face centers. The shape functions are implemented for linear/quadratic tri/quad elements only.

Meeting Challenges

One of the challenges in mapping data is to identify the matching faces or surfaces on the source and target sides. In cases of complex applications like the under-hood of an automobile, heat transfer between solid and fluid takes place across many common surfaces attached to many different parts. These surfaces and parts vary in size, shape and thickness. Depending on the analytic requirement, the parts are grouped together differently. For instance, the fluid region can group many of the exposed surfaces together in doing the CFD analysis. However, the solid region needs to treat each of the parts separately to account for different material properties and to analyze the thermal stresses in each of them. Moreover, the discrete representation of the common surfaces can be drastically different in the solid and fluid regions. The mesh topology and mesh density on the solid and fluid surfaces can also be very different. For instance, STAR-CCM+'s mesh engine generally creates polygonal faces and cells, while most of the finite element packages operate on tri/quad elements of various orders. In addition, the CFD analysis generally requires higher mesh densities (than solids) to capture boundary layers. Depending on the quality of mesh generated, the geometric detail of common surfaces can be compromised. These challenges can be addressed by using the additional mapping options. For more information, see Nearest Neighbor Search Constraints.

See Also:

Mapping Fields Using Data Mappers

Examples of Using Data Mappers

File-Based Coupling

Mapping STAR-CCM+ Solution Data to a CAE Model

Solution data from a STAR-CCM+ model can be mapped onto a CAE model for use with third-party CAE software.

This could involve using STAR-CCM+ to compute the flow solution in the fluid domain, then mapping and exporting traction data to the third party software to compute strains in the solid. Another example would be to use STAR-CCM+ to compute a full CHT (conjugate heat transfer) solution across the fluid and solid domains in the model, then map and export the solid region temperature data to a third-party code, and hence compute the thermal stress solution.

To map STAR-CCM+ solution data to a CAE model, first you need to import the CAE model into STAR-CCM+. It is then possible to map solution data from STAR-CCM+ onto faces, vertices, or volumes in the CAE model. Following this, you can export the mapped data, and use it in a third-party code.

Contents:

Mapping Boundary Data to Imported Surfaces
Mapping Boundary Data to Imported Beams
Mapping Region Data to Imported Volumes
Nearest Neighbor Search Constraints
Selecting a Reference Temperature Evaluation Option

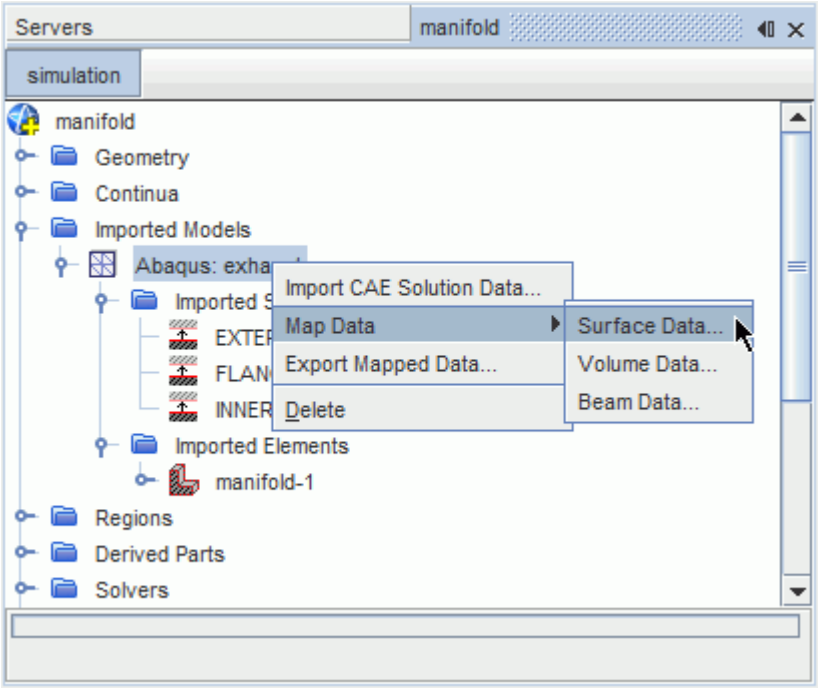
See Also:

Importing CAE Models
Importing a CAE Model: Workflow
Exporting Mapped Data

Mapping Boundary Data to Imported Surfaces

To map boundary data to the faces or vertices of an imported surface:

1. Expand the **Imported Models** node.
2. Right-click the **[Imported CAE Model]** node of interest and select **Map data** > **Surface Data...**



This menu option is also used to map imported CAE data to the STAR-CCM+ model.

In the *Map Imported Model Data* dialog:

3. Set *Map Direction* to **STAR-CCM+ to Imported Model**.
4. In the *Mapping Inputs* box, select the STAR-CCM+ boundaries you wish to map from, and the imported model surfaces you wish to map to.
5. Use the *Data to map* property to select which field data will be mapped.

   The drop-down list displays field functions that are available for mapping in the simulation.

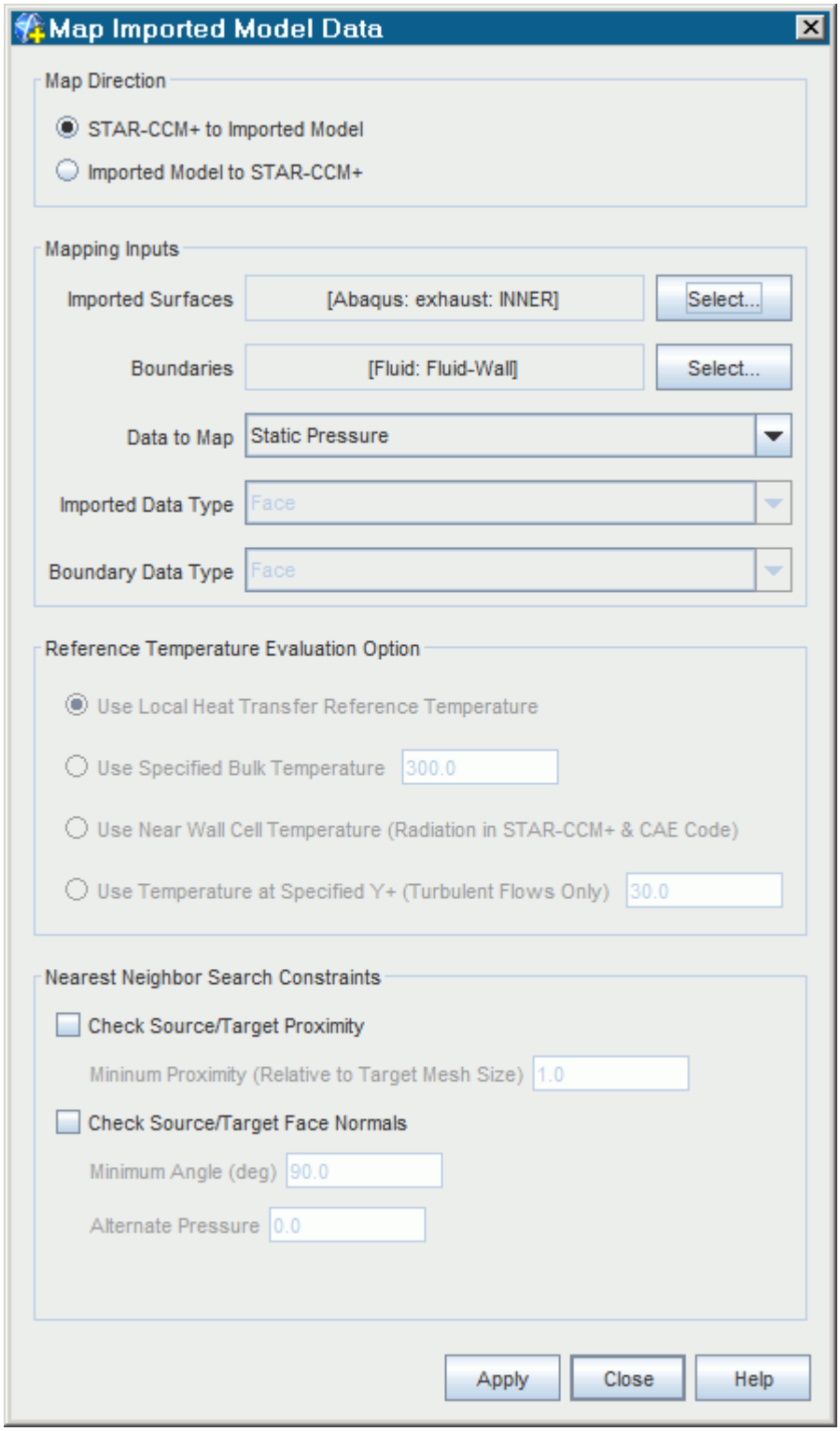
   When you select the field data, the **Imported Data Type** and **Boundary Data Type** are set automatically. If you are mapping local heat transfer coefficients, you can also select a reference temperature evaluation option. For more information, see Selecting a Reference Temperature Evaluation Option.

In some cases the mapping algorithm may fail to find appropriate neighbors for each face on the source boundary. If this happens:

6. Use the additional constraints provided in the *Nearest Neighbor Search Constraints* box at the bottom of the dialog. Activating these constraints also enables you to check the success of the mapping operation using the mapped flag field function. See Using the Mapped Flag Field Function.
7. When the settings in the dialog are complete click **Apply** then **Close**.

   The time taken to carry out the mapping operation will be displayed in the *Output* window, followed by `Done.` A field function will be exposed corresponding to the mapped data, for example the `Mapped Pressure` field function will be exposed if you map `Pressure` data. You can use this field function to export the mapped data.

   In the following example, the settings specify that static pressure data will be mapped from the**Fluid Wall** boundary of the STAR-CCM+ model to the **INNER** surface of the imported Abaqus model.



See Also:

Importing a CAE Model: Workflow

Mapping CAE Solution Data to a STAR-CCM+ Model

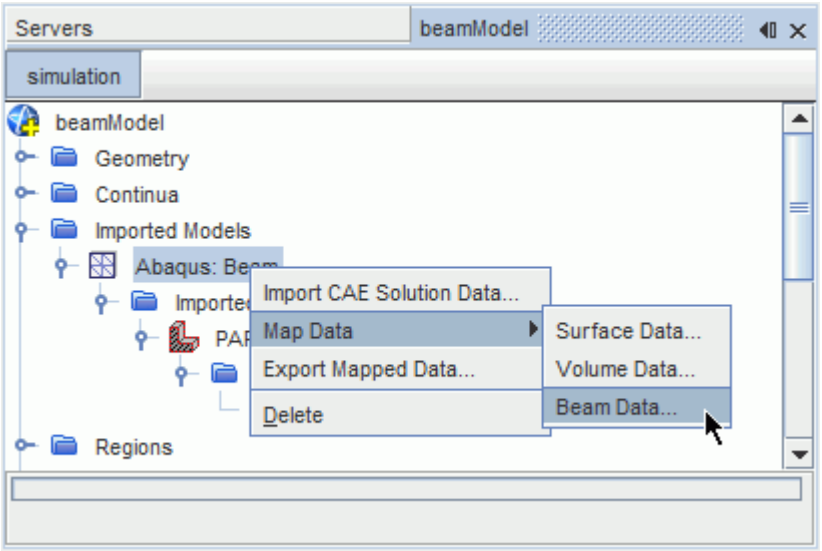
Exporting Mapped Data

Mapping Boundary Data to Imported Beams

It is possible to map pressure and shear stress data, that are defined on the faces of the wetted surface in STAR-CCM+, to the nodal forces and moments of an imported beam model.

To map boundary data to the nodes of an imported beam:

1. Expand the **Imported Models** node.
2. Right-click the **[Imported CAE Model]** node of interest and select **Map data** > **Beam Data...**



This menu option is also used to map imported CAE data to the STAR-CCM+ model.

This activates the *Map Imported Model Data* dialog.

Use the *Map Imported Model Data* dialog as follows:

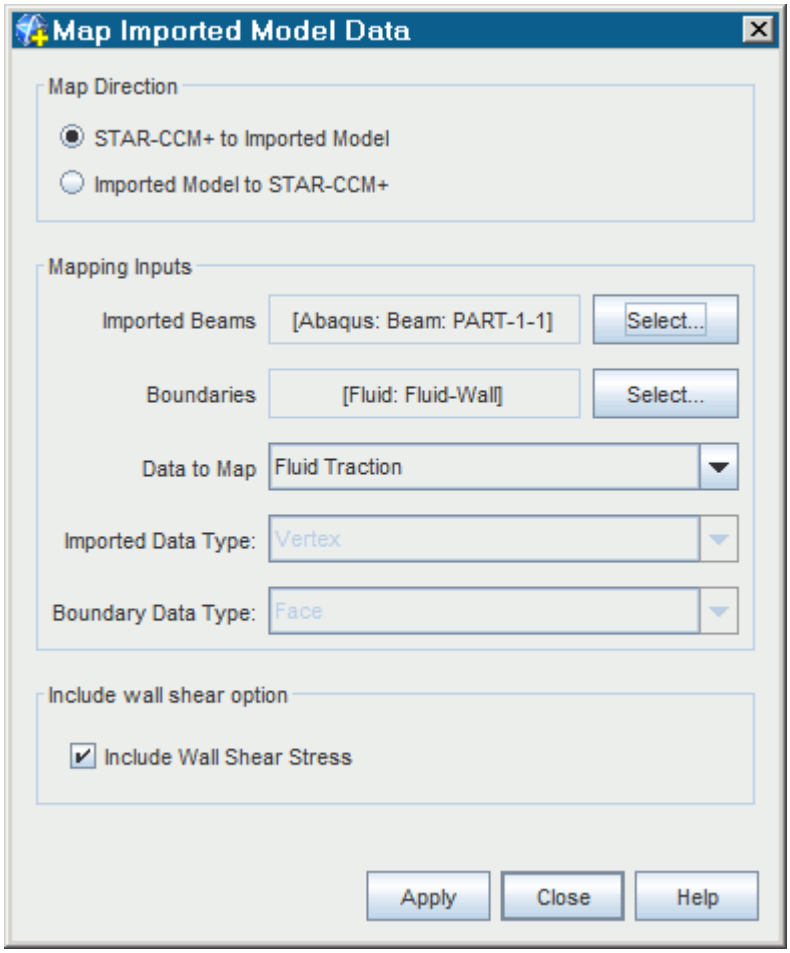
3. Set *Map Direction* to **STAR-CCM+ to Imported Model**.
4. In the *Mapping Inputs* box, select the STAR-CCM+ boundaries you wish to map from, and the imported beam you wish to map to.
5. Use the *Data to map* drop-down menu to select which field data will be mapped.

   This menu displays field functions that are available for mapping in the simulation.

   When you select the field data, the **Imported Data Type** and **Boundary Data Type** are set automatically.
6. Activate *Include Wall Shear Stress* if you wish to map wall shear stress data to the imported beam.
7. When the settings in the dialog are complete click **Apply**, then **Close**.

   The time taken to carry out the mapping operation will be displayed in the *Output* window, followed by `Done.` A field function will be exposed corresponding to the mapped data, for example the `Mapped Traction` field function will be exposed if you map `Traction` data. You can use this field function to export the mapped data.

   In the following example, the settings specify that fluid traction data will be mapped from the**Fluid-Wall** boundary of the STAR-CCM+ model to the imported beam element, **PART-1-1**.



See Also:

Importing a CAE Model: Workflow

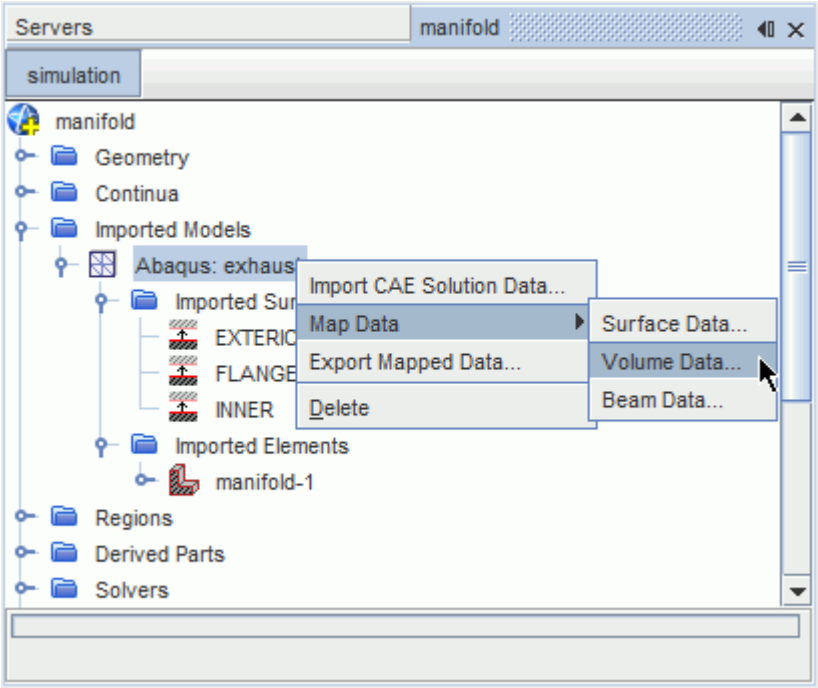
Mapping CAE Solution Data to a STAR-CCM+ Model

Exporting Mapped Data

Mapping Region Data to Imported Volumes

To map region cell data to the volume vertices of an imported model:

1. Expand the **Imported Models** node.
2. Right-click the imported CAE model node and select **Map data** > **Volume Data...**



This activates the *Map Imported Model Data* dialog.

Use the dialog as follows:

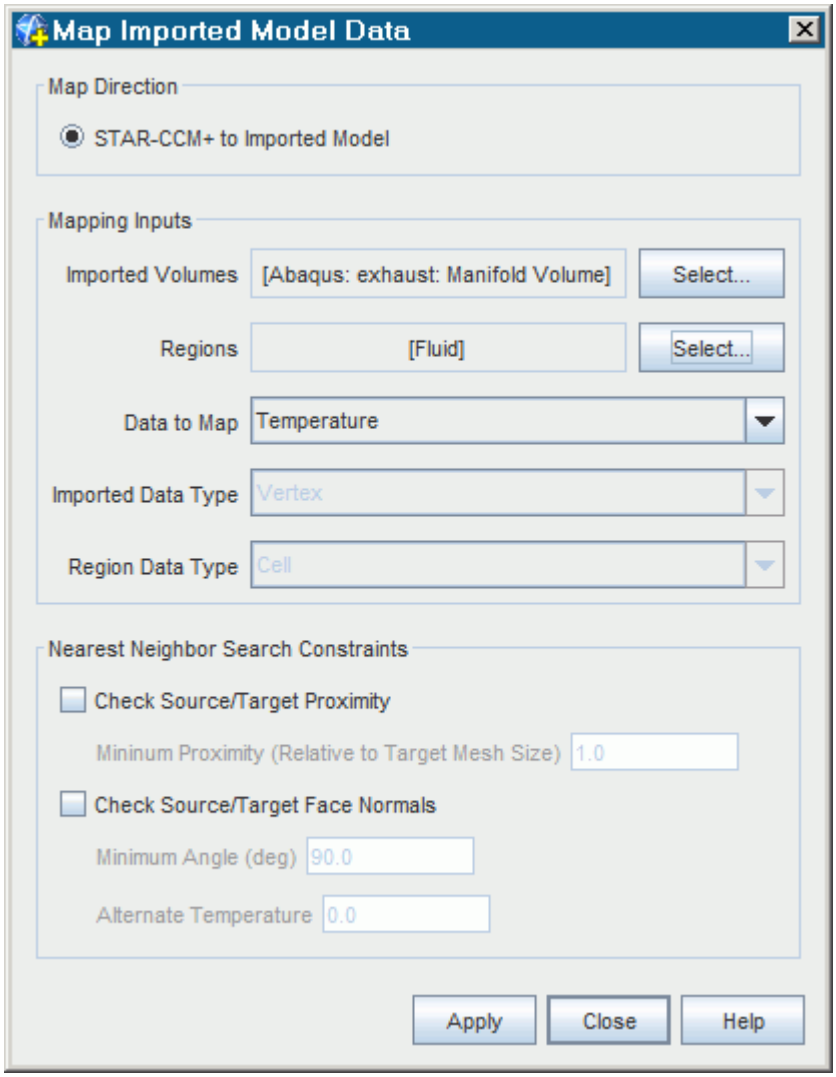
3. Set *Map Direction* to **STAR-CCM+ to Imported Model**.
4. In the *Mapping Inputs* box, select the STAR-CCM+ regions you wish to map from, and the imported model volumes you wish to map to.
5. Use the *Data to map* drop-down menu to select which field data will be mapped.

   Currently, it is only possible to map temperature data from a region to an imported volume.

   When you select the field data, the **Imported Data Type** and **Boundary Data Type** are set automatically.
6. Use the additional constraints in the *Nearest Neighbor Search Constraints* box at the bottom of the dialog to ensure the success of the mapping operation.
7. When the settings in the dialog are complete click **Apply**, then **Close**.

   The time taken to carry out the mapping operation will be displayed in the *Output* window, followed by `Done.` A field function will be exposed corresponding to the mapped data, called `Mapped Vertex Temperature`. You can use this field function to export the mapped data.

   The settings in the following example indicate that temperature field data will be mapped from the **Fluid** region to the **Manifold Volume**.



See Also:

Importing a CAE Model: Workflow

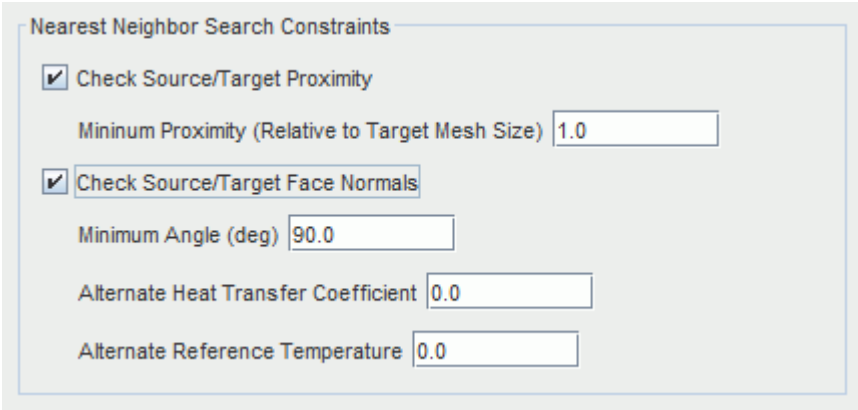
Nearest Neighbor Search Constraints

Exporting Mapped Data

Nearest Neighbor Search Constraints

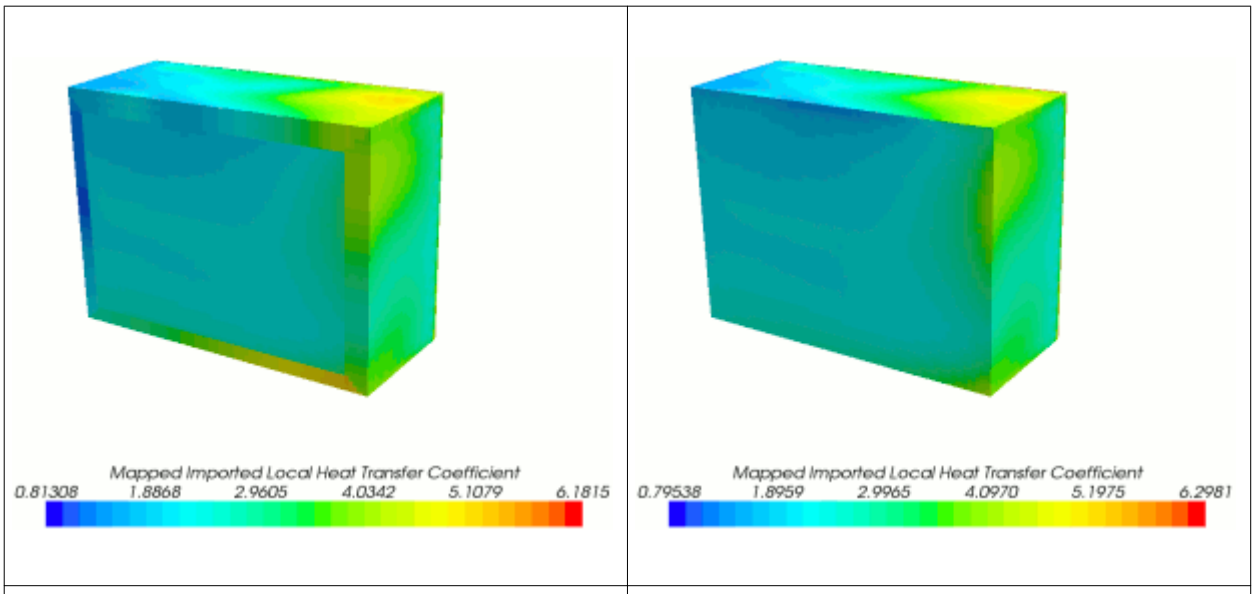
In some cases you will need to use additional checks to ensure that the matching entities of the source and target objects are correctly identified. These checks, or constraints, are provided in the *Nearest Neighbor Search Constraints* group box of the *Map Imported Model Data* dialog.

Using these constraints also allows you to use the mapped flag field function to assess the result of the mapping. For more information, see Using the Mapped Flag Field Function.



- **Check Source/Target Proximity** -- The source face from which data is being mapped should lie within a certain distance of the target face. This ensures that data is not mapped from surfaces which are too far away. The tolerance is set by specifying a fraction of the maximum of the mesh sizes on the source/target surfaces. The default value is 1.0.
- **Check Source/Target Face Normals** -- This check enforces the rule that the source face normal and target face normal be facing each other. The **Minimum angle** option specifies the minimum angle permissible between opposing face normals.

  This constraint is essential when a single surface wraps around edges and corners of the geometry. In the example below, the constraint ensures that data on each side of the source box is mapped to the correct side of the target box.



Without face normals check	With face normals check

This check should also be activated when you are mapping to shell elements--particularly when the imported shell does not have a consistent distribution of positive and negative faces. If the geometries of the source and target surfaces match closely, you can choose to map to both sides of the shell element and rely on this constraint to choose the correct side (positive or negative) on which data is mapped.

If some faces are not intended to be mapped, and will fail either of the criteria above, you can choose to assign reasonable alternate values for the unmapped faces. This can be done by specifying a value in the alternate value text boxes. The default alternate values are 0.0.

Contents:

Using the Mapped Flag Field Function

Using the Mapped Flag Field Function

The faces that satisfy the specified nearest neighbor search constraints are successfully mapped using an appropriate interpolation technique. The faces that fail these tests are not mapped at all. To allow you to troubleshoot and choose appropriate values for the tolerances, a field function named `Mapped Flag` is created upon each mapping operation.

This field has the following values:

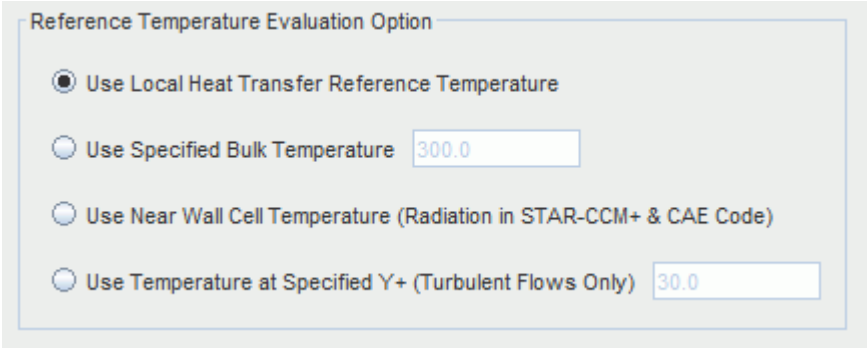
- 0 -- mapping was successful
- 1 -- proximity check failed
- 2 -- normalcy check failed

You can visualize the unmapped faces by plotting contours of `Mapped Flag`.

Selecting a Reference Temperature Evaluation Option

When you map heat transfer coefficients from STAR-CCM+ to an imported CAE model, you can select the method by which STAR-CCM+ evaluates the heat transfer coefficient and reference temperatures.

When *Data to map* is set to **Local Heat Transfer Coefficient**, the *Reference Temperature Evaluation Option* box appears within the *Map Imported Model Data* dialog.



You can choose the method by which STAR-CCM+ evaluates the reference temperature by selecting one of the four options available. The particular form of heat transfer coefficient that is mapped will depend on the reference temperature chosen. For the first three options, the mapped heat transfer data will be stored in the field function `Mapped Local Heat Transfer Coefficient`.

- **Local heat transfer reference temperature**

This option allows you to map the local heat transfer reference temperature, which results in the creation of the `Mapped Local Heat Transfer Reference Temperature` field function. With this option, STAR-CCM+ will base its mapping on the Local Heat Transfer Coefficient.

- **Specified bulk temperature**

  This option allows you to specify a single temperature to use as the reference temperature. Selecting this option results in the creation of the `Mapped Bulk Temperature` field function. STAR-CCM+ will base its mapping on the Heat Transfer Coefficient.

- **Near wall cell temperature**

  This option allows you to map the cell-based reference temperature values. It results in the creation of the `Mapped Cell Based Temperature` field function. STAR-CCM+ will base its mapping on the Local Heat Transfer Coefficient.

- **Specified y+ temperature**

  This option is for turbulent flows and allows you to map reference temperatures at a specified y+ value. This option is preferred over the local heat transfer reference temperature. Selecting this option results in the creation of the `Mapped Heat Transfer Reference Temperature User Y+`. Note that the specified y+ heat transfer coefficient and reference temperature do not take account of boiling heat transfer and radiation. For cases involving boiling and radiation, you should map and export the boundary heat flux, or alternatively the local heat transfer coefficient and reference temperature.

  With this option, STAR-CCM+ will base its mapping on the Specified y+ Heat Transfer Coefficient, and will store the mapped heat transfer data in the field function `Mapped Specified y+ Heat Transfer Coefficient`.

Further discussion is provided in the section, Exchanging Heat Transfer Coefficients.

See Also:

What Methods Are Available for Exchanging Heat Transfer Coefficients?

Exporting Mapped Data

STAR-CCM+ solution data that has been mapped onto an imported CAE model can be exported from STAR-CCM+ so that it can be used in an external CAE package. Data is exported from STAR-CCM+ in text files that can be read by the respective CAE package.

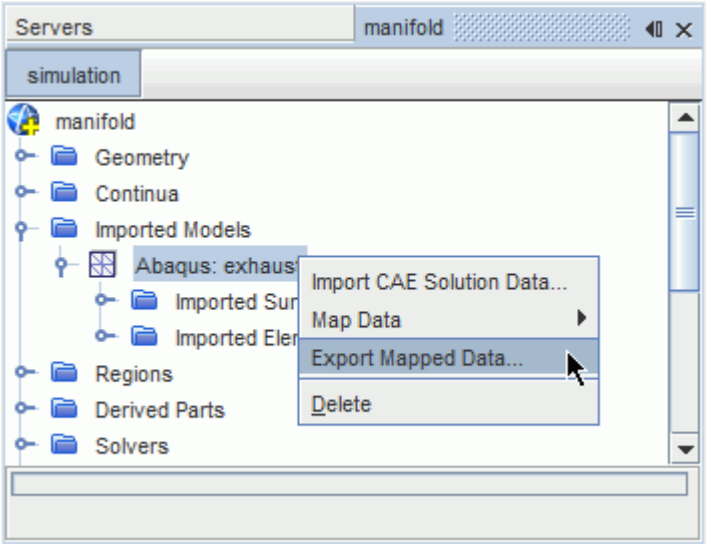
The following table shows the export options supported by STAR-CCM+:

	CAE Package			
Field Function	**NASTRAN**	**Abaqus**	**ANSYS**	**RadTherm**
Pressure	✓	✓	✓	-
Traction	✓	✓	✓	-
Heat Transfer Coefficient and Reference Temperature	✓	✓	✓	✓
Heat Flux	✓	✓	✓	✓
Temperature	✓	✓	✓	-

Before exporting data, ensure you have mapped the surface, beam or volume data from the STAR-CCM+ model to the imported model.

To export mapped data:

1. Right-click the **Imported Models** > **[Imported CAE model]** node of interest and select **Export Mapped Data...**

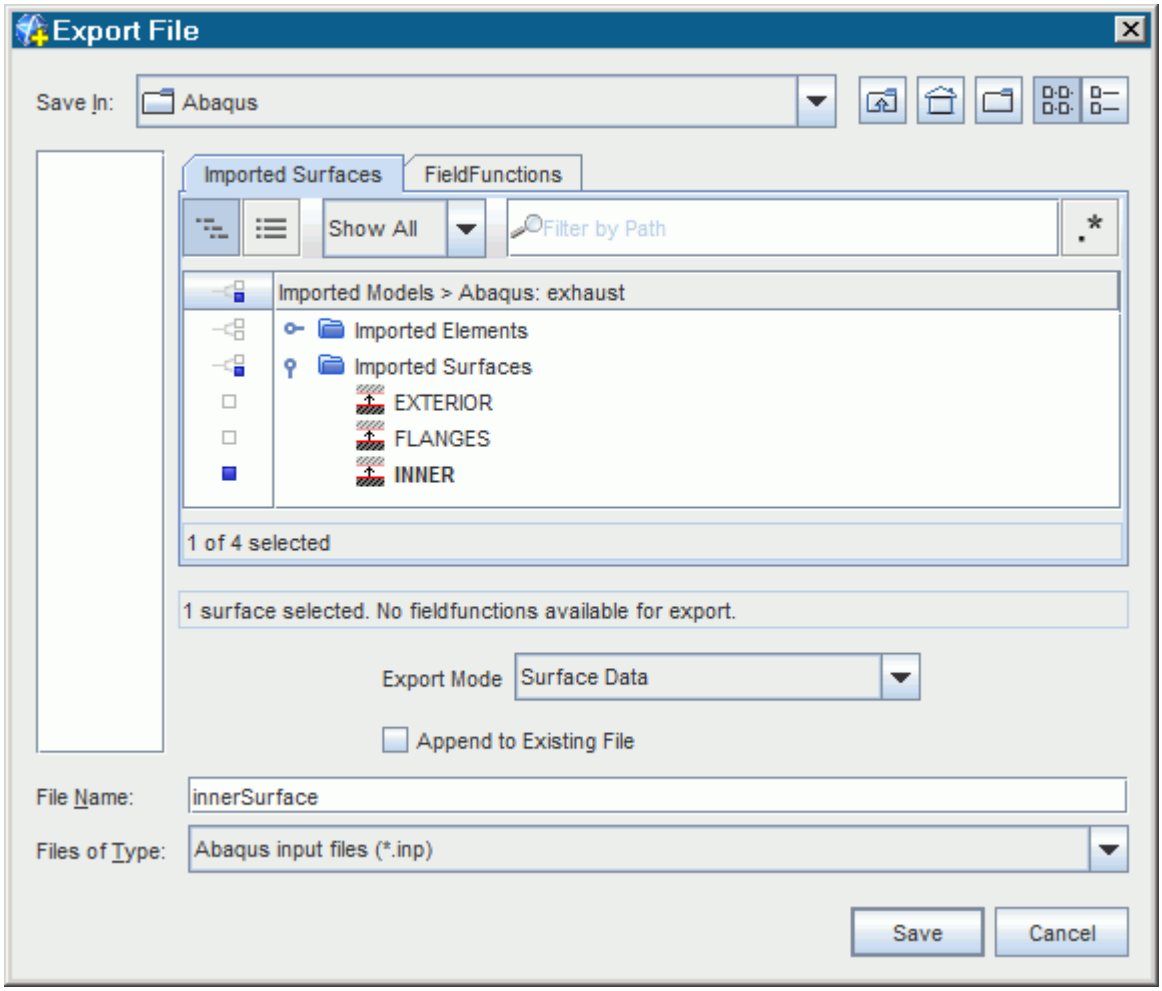


In the *Export File* dialog:

1. Navigate to the location you want to save the file and enter a file name, or select an existing file. If you activate **Append to existing file**, the exported data will be added to the end of the selected file.
2. Use the *Export Mode* menu at the bottom-right of the dialog to select either the **Surface Data** or **Volume Data** export option.

   This selection will affect the options shown in the dialog.
3. In the *Imported Surfaces* or *Imported Volumes* tab (depending on the selected export mode), select the surfaces or volumes for which you wish to export data.

   In the following screenshot, the settings specify that surface data will be exported for the imported surface **INNER**.

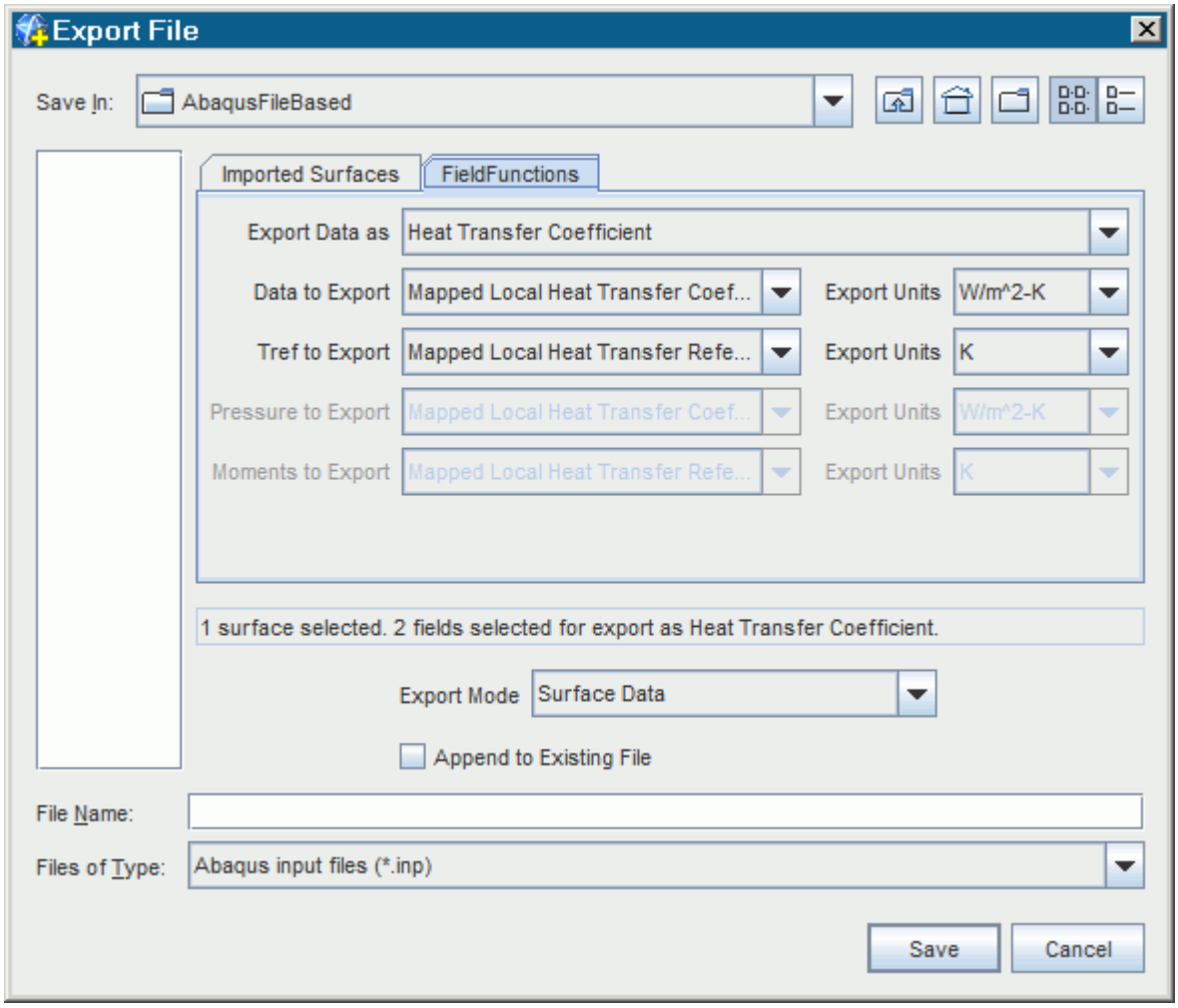


4. Select the *FieldFunctions* tab.
5. From the *Export data as* menu, select the field data you wish to export.
6. From the *Data to export* menu, select the corresponding field function.

   Note that to export heat transfer coefficient data, select the mapped heat transfer coefficient field function in the *Data to export* menu and the mapped reference temperature field function in the *Tref to export* menu. To export traction data, select the mapped wall shear stress field function in the *Data to export* menu and the mapped pressure field function in the *Pressure to export* menu.

   If you are exporting Temperature data for surfaces, you should select `Temperature BC`. If you are exporting Temperature data for shell surfaces, you can choose either temperature option.
7. Ensure the correct units are selected in the *Export units* menu.
8. A summary of the selected surfaces/volumes and field functions will be displayed in the box underneath the tabbed panel.

   In the screenshot below, the settings specify that the `Mapped Local Heat Transfer Coefficient` and `Mapped Local Heat Transfer Reference Temperature` field functions will be exported as heat transfer coefficient data.



9. To export the data, click **Save**.

If no mapped field functions are available for the selected surface or volume, a message will be displayed in the summary box in the *Export File* dialog. Further information on exporting data to NASTRAN, Abaqus, ANSYS, and RadTherm are provided in a separate section.

Contents:

Exporting Temperature Data on Shells

See Also:

Importing a CAE Model: Workflow

Mapping STAR-CCM+ Solution Data to a CAE Model

Coupling with NASTRAN

Working with Abaqus

Working with ANSYS

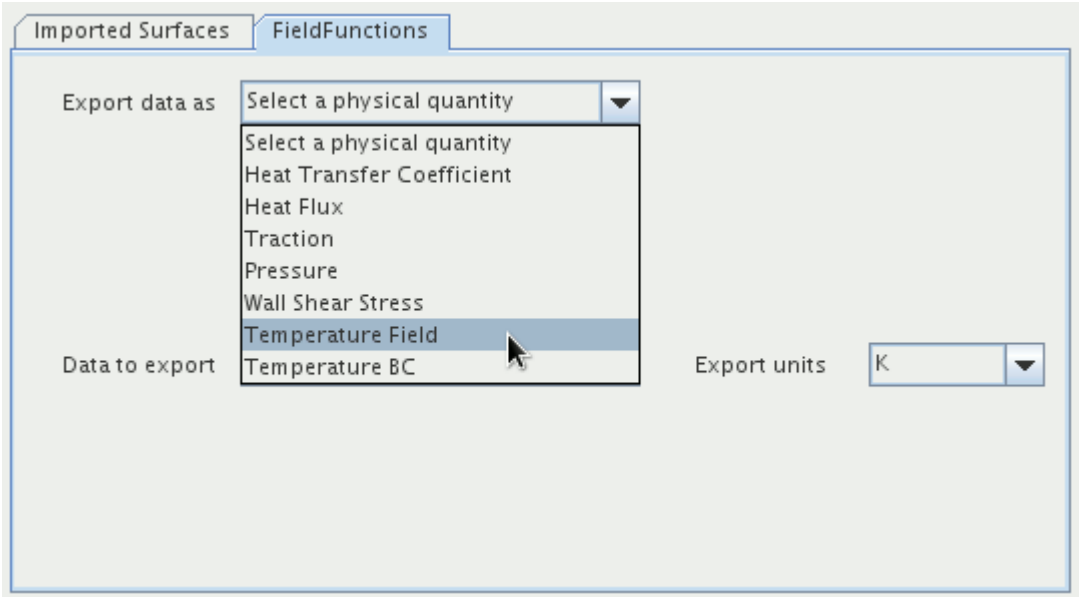
Working with TAITherm

Exporting Temperature Data on Shells

When exporting temperature data on shell surfaces you can choose to export the data as a temperature field, or as a boundary condition.

When the data is exported as a temperature field, it can be used to specify a constant temperature across the thickness of the shell in the external CAE package. This is used for very thin shells, where any temperature gradient across the shell thickness would be negligible.

Temperature data exported as a boundary condition specifies the temperature on one side of the shell. The data can then be used in an external CAE package to compute the temperature gradients across the shell thickness.



To export the temperature data:

1. Map the temperature data to the relevant shell surface (for field data export, you can export the data to either shell surface).
2. Follow the instructions given in the previous section to export the data that has been mapped to the shell surface.
3. From the *Export data as* menu of the *FieldFunctions* tab, select one of the following options:
   - **Temperature Field** -- to export temperature data as a field
   - **Temperature BC** -- to export temperature data as a boundary condition
4. From the *Data to export* menu, select the **Mapped Vertex Temperature** field function.

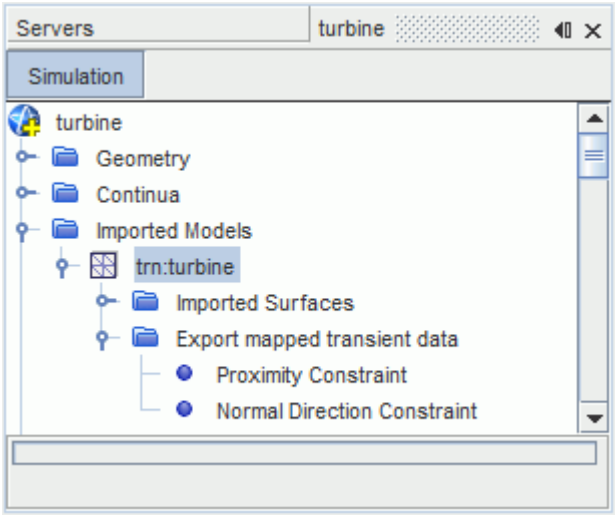
See Also:

Mapping Boundary Data to Imported Surfaces

Mapping and Exporting Transient Surface Data

STAR-CCM+ allows you to map transient surface data onto the surfaces of an imported transient model, and to export the mapped data to an external `.trn` or `.mat` (Matlab) file.

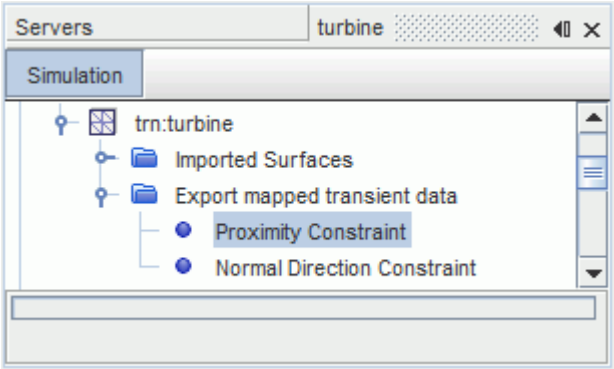
Map and export operations are carried out through the imported transient model node **trn:[fileName]** (**trn:turbine** in the image below) and its subnode, **Export mapped transient data**.



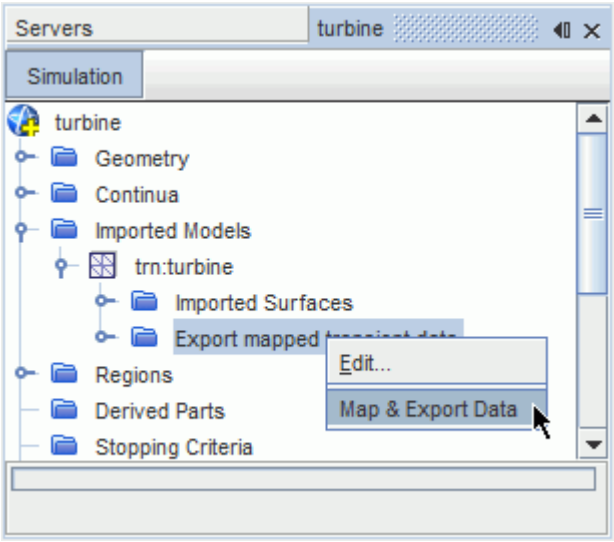
To map and export transient surface data to target surfaces of an imported transient model:

1. Import the transient model into STAR-CCM+. See Importing a CAE Model.
2. Expand the **Imported Models** node.
3. Select the relevant imported transient model node.
4. Activate *Export Mapped Transient Data*.

   A new node, *Export mapped transient data*, is added within the imported transient model node.
5. Select the *Export mapped transient data* node and set its properties to specify the map and export settings. See Export Mapped Transient Data Properties.
6. Specify mapping constraints by setting the **Proximity Constraint** node properties and/or the **Normal Direction Constraint** node properties. All the available properties are listed in the sections Proximity Constraint Check Reference and Normal Direction Constraint Check Reference.



7. Right-click the *Export mapped transient data* node and select **Map & Export Data**.



An alternative map and export procedure that uses data set functions is described under Using Data Set Functions.

Contents:

[Imported transient model] Properties
Export Mapped Transient Data Properties

[Imported transient model] Properties

Transient Display Time	Specifies the current display time. The slider can vary from the maximum to the minimum time of the current signal.
Transient End Time	Displays the maximum time in the file.
Export Mapped Transient Data	Allows to activate the **Export mapped transient data** node, for map and export operations.

	Activated	Enables map and export of transient surface data through the **Export mapped transient data** node.
	Deactivated	Map and export of transient surface data is not enabled.
Update Data Set Functions	Allows automatic update of the data set functions.	
	Activated	The data set functions are automatically updated after any change to the imported transient model node.
	Deactivated	The data set functions are not updated automatically.

Export Mapped Transient Data Properties

Export Directory	Specifies the directory to which the file containing the transient surface data is exported.
Mapped File Format	Specifies the format of the file containing the transient surface data. Two options are available, `.trn` and `.mat`.
Source Surfaces	Allows to select the transient model surfaces from which the transient surface data is mapped.
Target surfaces	Allows to select the transient model surfaces onto which the transient surface data is mapped.
Scalar Functions	Allows to select the scalar fields to be mapped from the source surfaces to the target surfaces.
Vector Functions	Allows to select the vector fields to be mapped from the source surfaces to the target surfaces.
Interpolation Method	Specifies the interpolation technique that is used for mapping.
Export Data Location	Specifies whether the transient surface data is mapped to the target mesh vertices or surfaces.
Export Frequency	Specifies the number of time-steps between exports.
Start Export At Time Step	Specifies at what time-step the transient surface data export starts.
End Export At Time Step	Specifies at what time-step the transient surface data export ends.

Importing CAE Solution Data

In two-way coupling, a solution is obtained by sequentially running STAR-CCM+ and a third-party code to obtain a complete solution. This may involve using STAR-CCM+ to compute the traction load on a flexible structure caused by fluid flow, and a third-party code to compute the deflection of the structure and associated displacements and stresses. In order to carry out a two-way coupled simulation, not only do you need to map and export solution data from STAR-CCM+, but you will also need to import solution data from the third-party software, and map it onto the STAR-CCM+ model.

Importing CAE solution data from es-ice or pro-STAR is different to other CAE products in terms of limitations and available data. For more information, see Working with es-ice or pro-STAR.

The vertex data associated with an imported CAE model can be imported into STAR-CCM+ via the imported CAE model node. The following table shows the import options currently supported by STAR-CCM+, not including es-ice or pro-STAR:

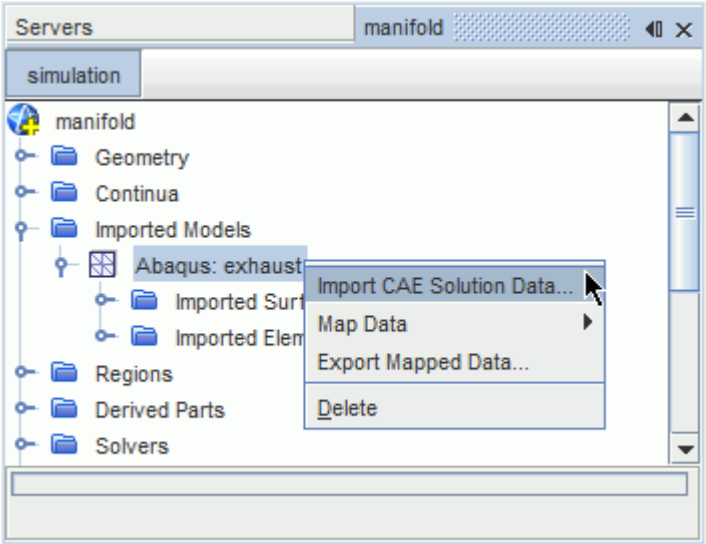
	CAE Package			
Field Function	**NASTRAN**	**Abaqus**	**ANSYS**	**RadTherm**
Temperature	✓	✓	✓	✓
Displacement	✓	✓	✓	-
Eigenmodes	✓	✓	✓	-
Qvol	✓	-	-	-
Temperature (volume import)	✓	✓	✓	-

For Abaqus data import, in the *Open* dialog file selection pathway, it is also necessary to enter the command used to launch Abaqus, and the **Load Step Name** for the data. Additional settings are available when you are importing eigenmodes. When you import Nastran data using the Qvol card, in the *Import Data Options* dialog, set the *Data Type* to **Cell** as the values are present at the cell centers.

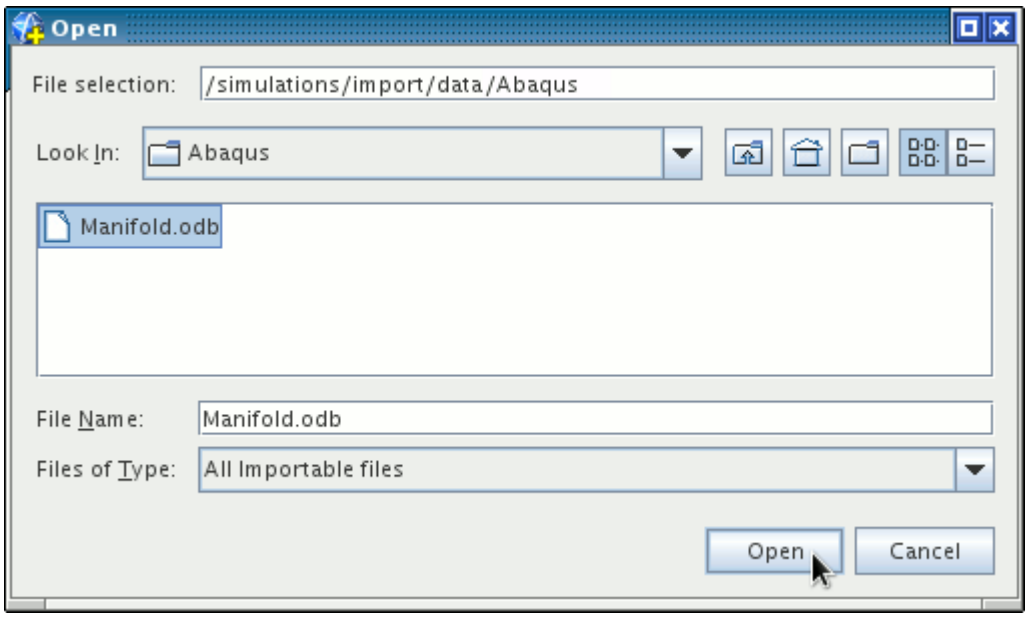
When vertex data is imported into the simulation it is stored in the simulation ready to be mapped onto a boundary. The data cannot be used in the simulation until it has been mapped onto the STAR-CCM+ model, but it can be visualized as described below. If displacement data has been imported into the simulation you can deform the CAE model based on the displacements.

To import CAE solution data:

1. Expand the **Imported Models** node.
2. Right-click the **[Imported CAE Model]** node and select **Import CAE Solution Data...**

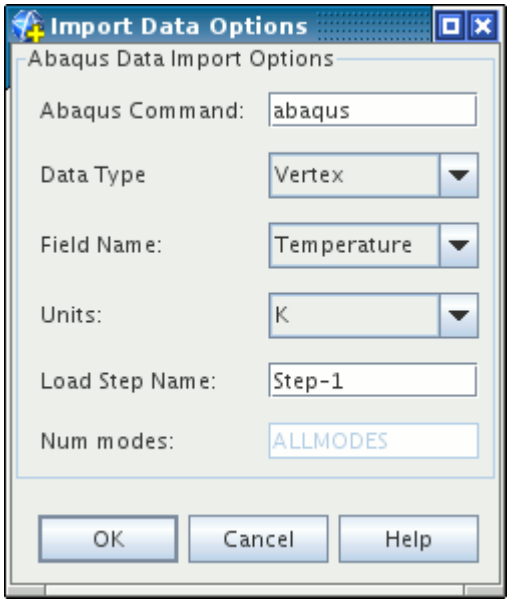


3. In the *Open* dialog, select the output file containing the solution data. In the *Files of Type* drop-down list you can select the appropriate format for the CAE model.



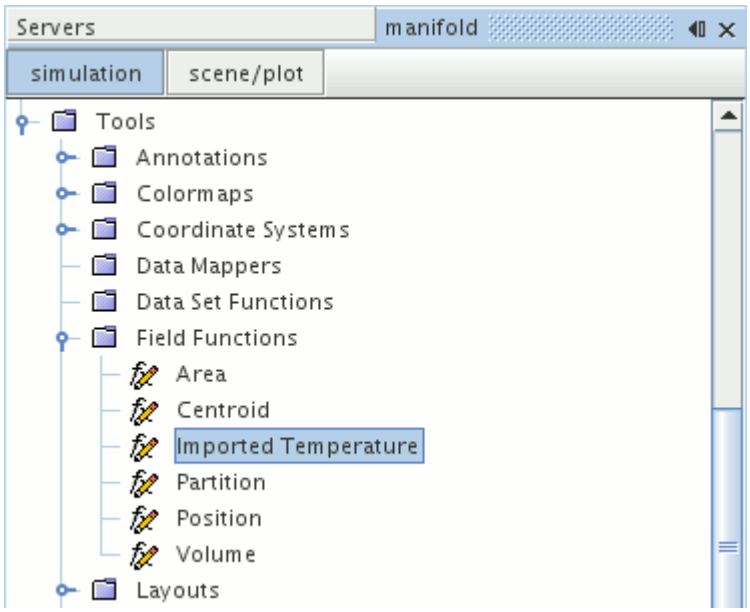
This activates the *Import Data Options* dialog, which is used to specify the properties of the imported data.

4. In the *Import Data Options* dialog, use the *Data Type* menu to select the type of data being imported, for example **Vertex**. From the *Field Name* menu, select the field data, then select the corresponding units from the *Units* menu.

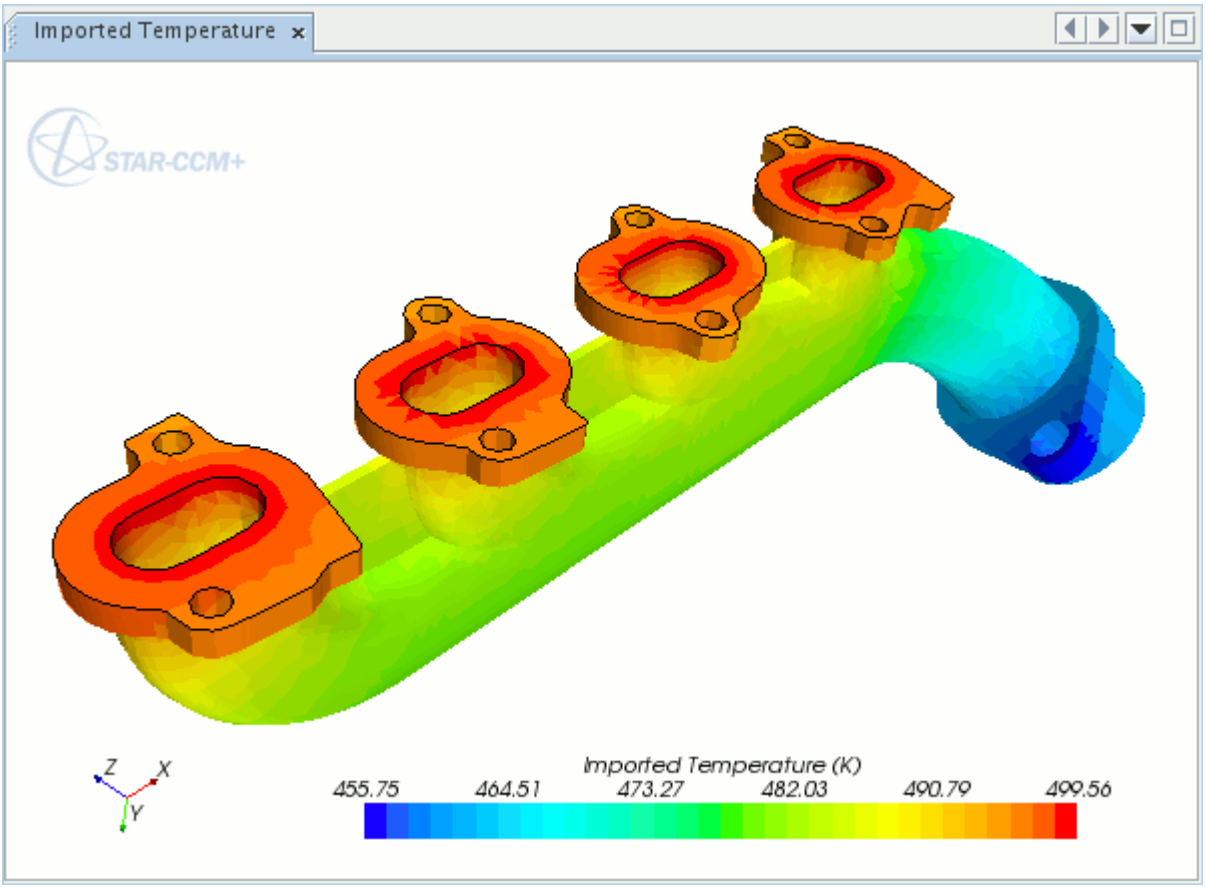


5. Click **OK**.

   The import process results in the data being represented as one or more field functions in STAR-CCM+.



This field function can be displayed just like any other scalar value.



Contents:

Importing Eigenmode Data
Deforming the CAE Model Using Imported Displacements

See Also:

Mapping CAE Solution Data to a STAR-CCM+ Model

Importing Eigenmode Data

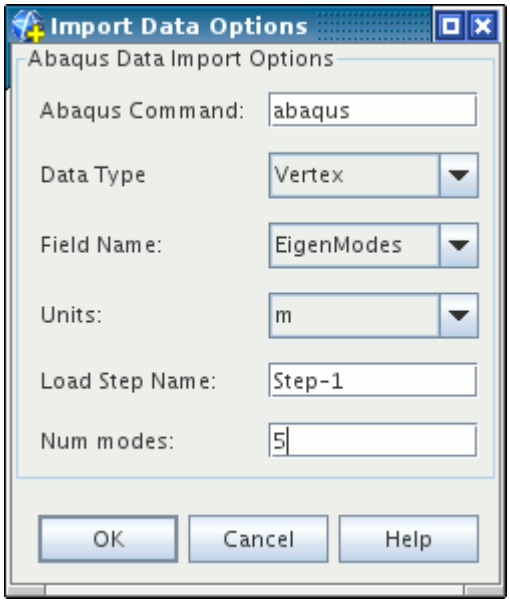
When importing eigenmode data for Nastran and Abaqus models, you can select the number of modes to import.

To do this:

1. Open the *Import Data Options* dialog and set *Field Name* to **EigenModes**.

   This activates the *Num modes* text box. You can enter the following options to specify the number of modes:

   - `ALLMODES` -- imports data for all modes in the data file.
   - `X` -- to import the first X modes, for example `3`, `5`, `10`.



Deforming the CAE Model Using Imported Displacements

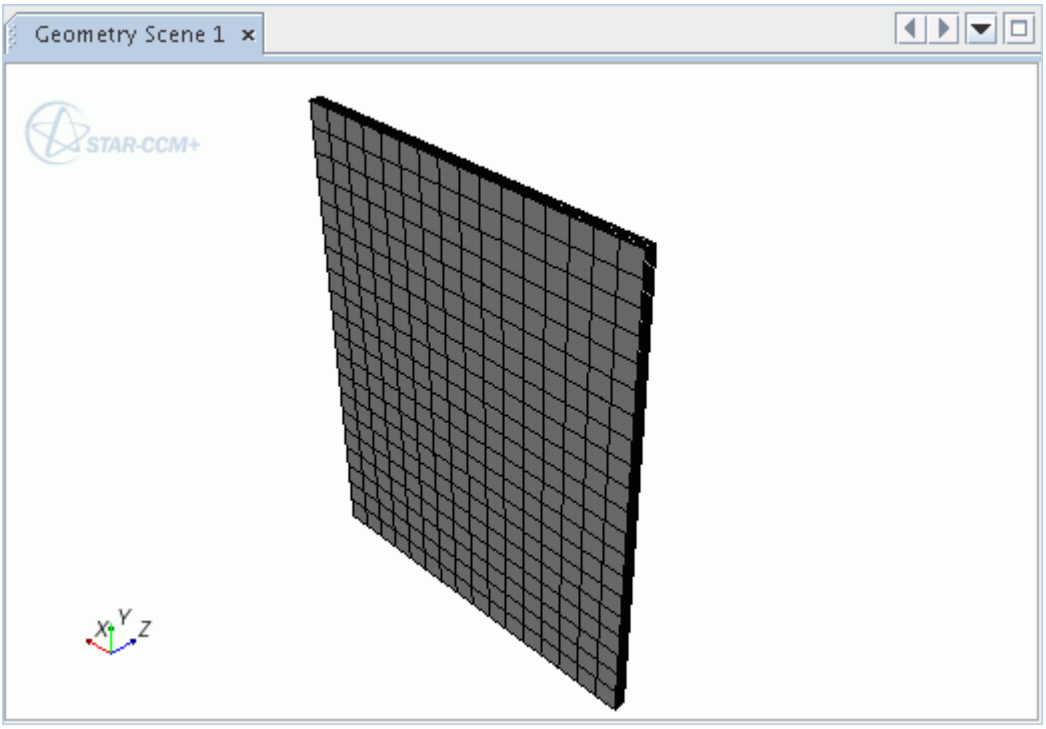
The CAE model can be deformed based on imported displacement data. This process deforms the CAE model mesh by applying the displacement data at each surface vertex. This allows you to view the deformed shape of the CAE model within STAR-CCM+. It is also possible to display scalar quantities on the deformed model within a scalar scene.

To deform a CAE model:

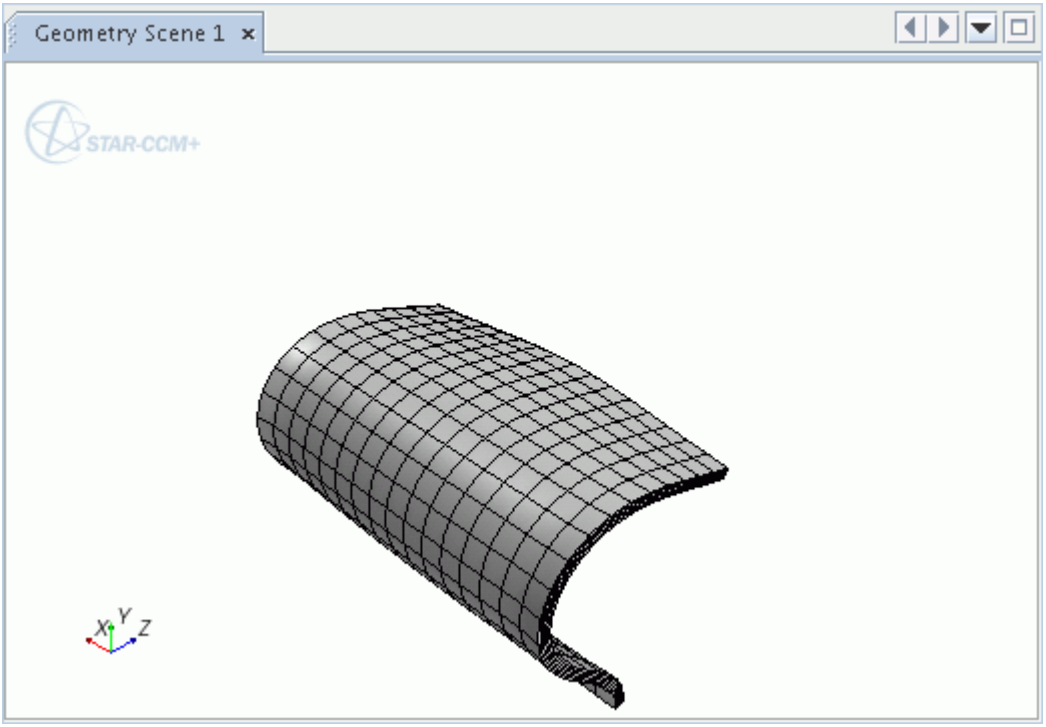
1. Import displacement data for the model.

   This reveals the deform option in the CAE model pop-up menu.
2. Right-click the **[Imported CAE Model]** node and select **Deform Using Imported Surface Displacements**.

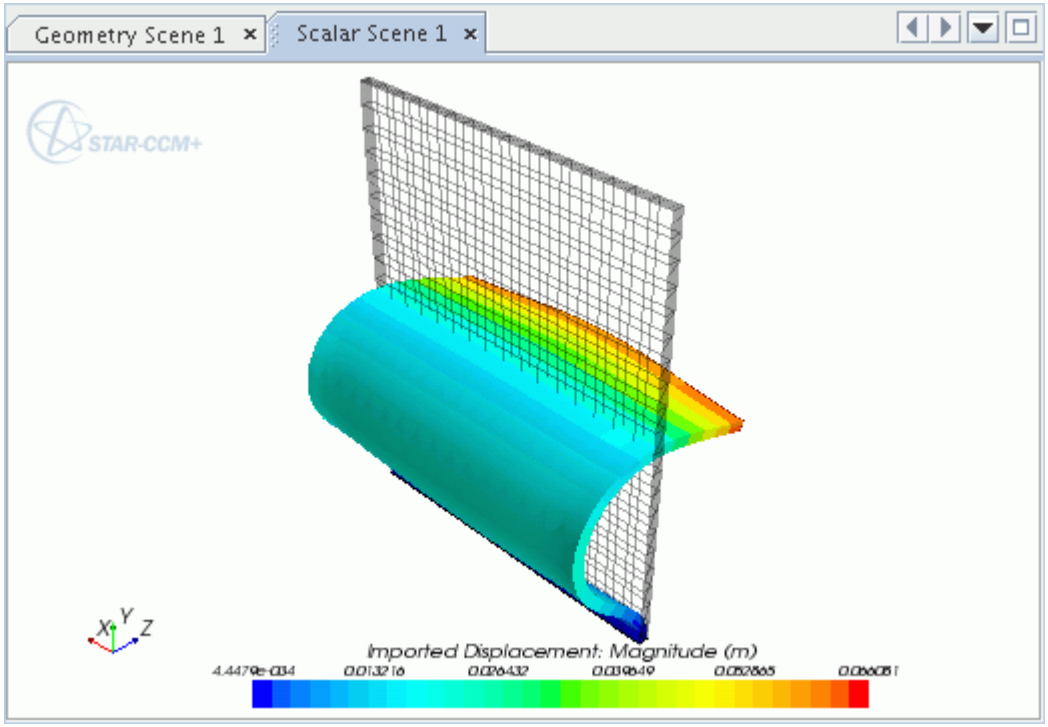
   The following example shows an imported CAE model of a plate fixed at the base. In the external CAE package, a traction load was placed on the front surface of the plate.



In the following screenshot the CAE model has been deformed based on the imported surface displacements. Note that it is often necessary to change the parts displayed in the scene in order to view the deformed mesh. This is because the displacement data is typically associated with a particular surface of the model, such as an imported surface, or volume surface. Therefore, you may need to add one of the other CAE model surfaces to the scene.



The deformed model can also be used to display scalar data. In the example below, the `Imported Displacement` field function is displayed on the surface. A geometry displayer has also been used to show the original position of the plate.



See Also:

Imported Volume Elements

What Is the Imported Surfaces Node?

Adding Parts to Displayers

Mapping CAE Solution Data to a STAR-CCM+ Model

In two-way coupling, data from the third-party CAE solution is used to specify boundary conditions in STAR-CCM+.

To do this, the data must first be imported into STAR-CCM+, then it can be mapped onto the corresponding boundary. For example, nodal displacements computed in an external FEA package may be mapped onto an FSI boundary in STAR-CCM+, and used to define mesh motion via the morpher. Currently, it is possible to map surface data and beam data from a CAE model to a STAR-CCM+ model.

Contents:

Mapping Imported Surface Data to STAR-CCM+ Boundaries
Mapping Imported Beam Data to STAR-CCM+ Boundaries

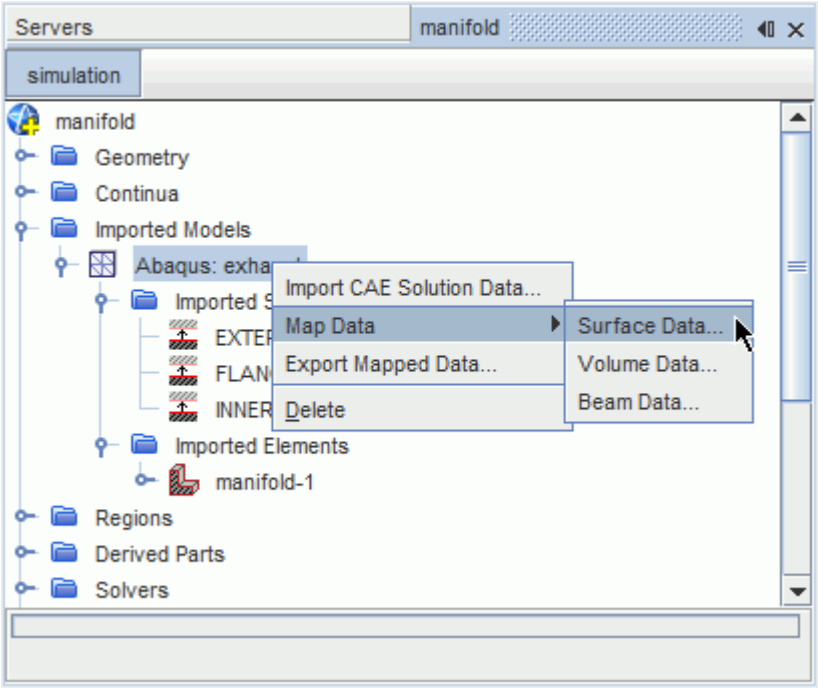
See Also:

Importing a CAE Model: Workflow

Mapping Imported Surface Data to STAR-CCM+ Boundaries

To map surface data from the CAE model to STAR-CCM+:

1. Right-click the **[Imported CAE Model]** node and select **Map Data** > **Surface Data...**



This activates the *Map Imported Model Data* dialog.

In the *Map Imported Model Data* dialog:

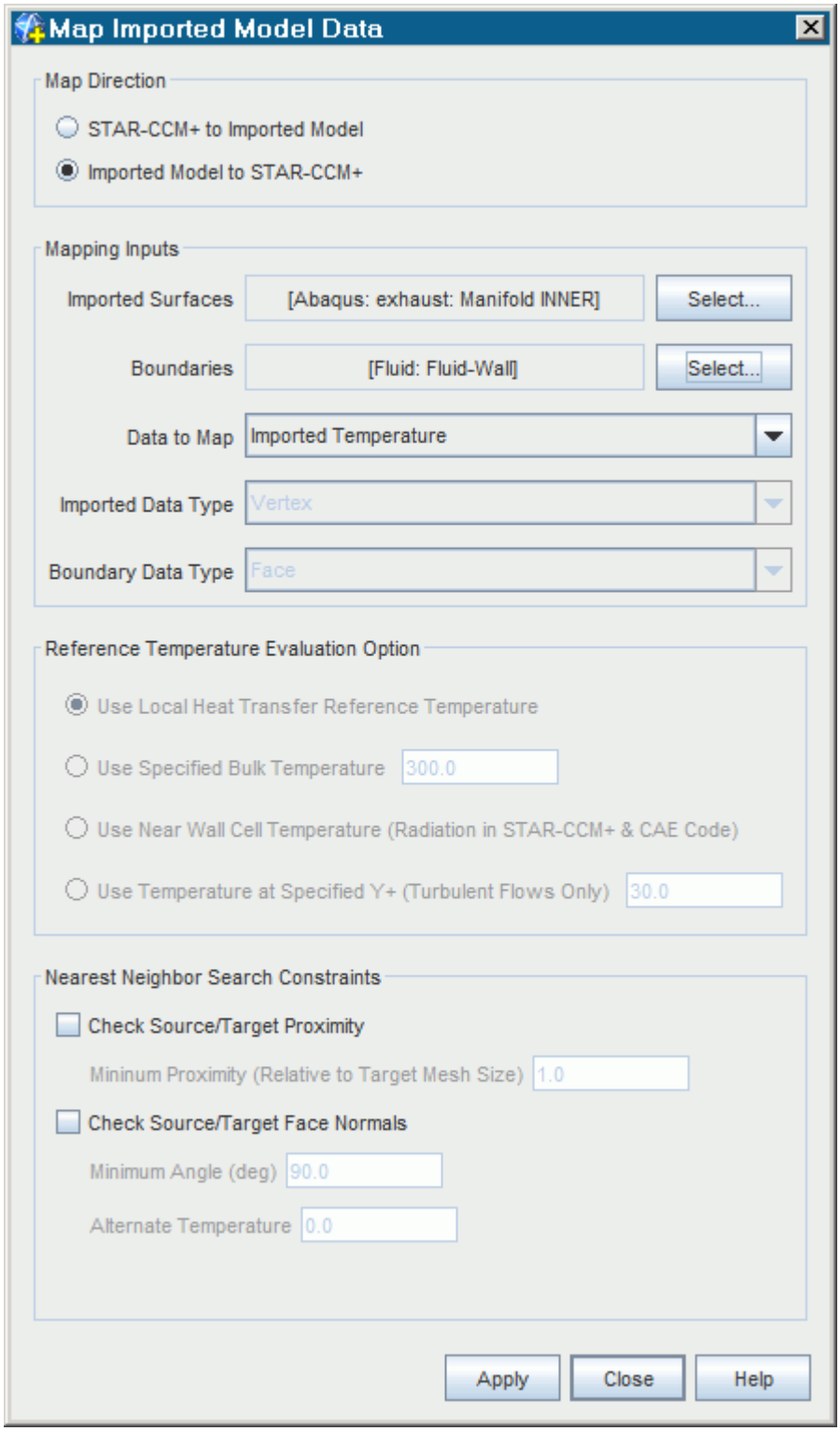
2. Set *Map Direction* to **Imported Model to STAR-CCM+**.
3. In the *Mapping Inputs* box, select the imported model surfaces you wish to map from, and the STAR-CCM+ boundaries you wish to map to.
4. The *Data to map* menu allows you to select which field data will be mapped.

   This menu displays field functions that are available for mapping in the simulation. When you select the field data, the *Imported Data Type* and *Boundary Data Type* are set automatically. If you are mapping local heat transfer coefficients, you can also select a reference temperature evaluation option. For more information, see Selecting a Reference Temperature Evaluation Option.

In some cases the mapping algorithm may fail to find appropriate neighbors for each face on the source boundary. If this happens:

5. Use the additional constraints provided in the *Nearest Neighbor Search Constraints* box at the bottom of the dialog. Activating these constraints also enables you to check the success of the mapping operation using the mapped flag field function. See Using the Mapped Flag Field Function.

   In the following example, the settings specify that imported temperature data will be mapped from the **Manifold INNER** surface of the imported Abaqus model to the**Fluid Wall** boundary of the STAR-CCM+ model.

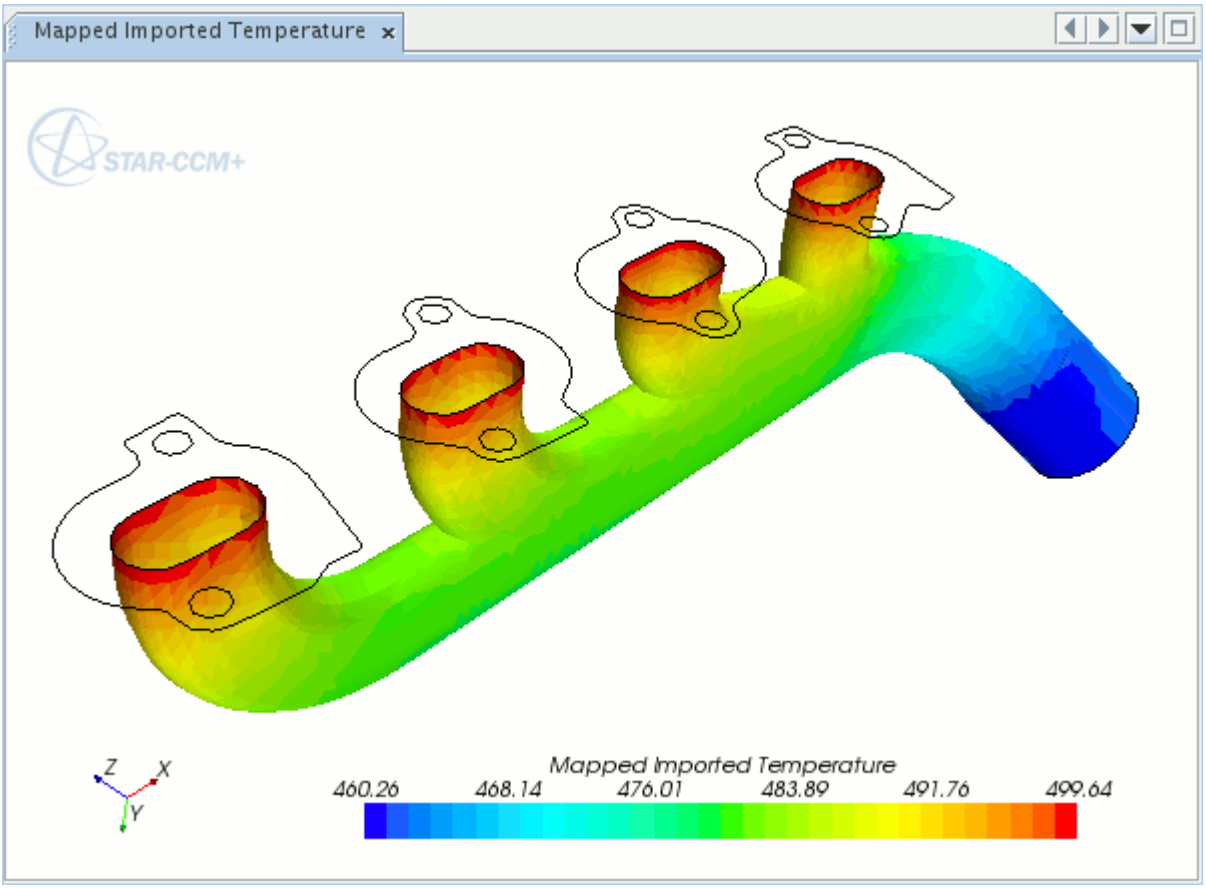


6. When the settings in the dialog are complete click **Apply**, then **Close**.

   The time taken to map the data will be shown in the *Output* window, and the message `Done.` will be displayed when the mapping operation is finished.

To display the mapped data results:

7. Create a scalar scene.
8. Select the scalar function and the mapped surface part to display.



When the mapping operation is carried out for the first time, a new field function will be created, according to the type of data that has been mapped from the CAE model. For example, if temperature data was imported in the previous step, an `Imported Temperature` field function will be created automatically by STAR-CCM+. When this data is mapped onto a boundary, the field function `Mapped Imported Temperature` is created. This field function can then be used when specifying a boundary condition, for example by setting the Thermal Specification type to Field Function, then selecting `Mapped Imported Temperature` from the list.

See Also:

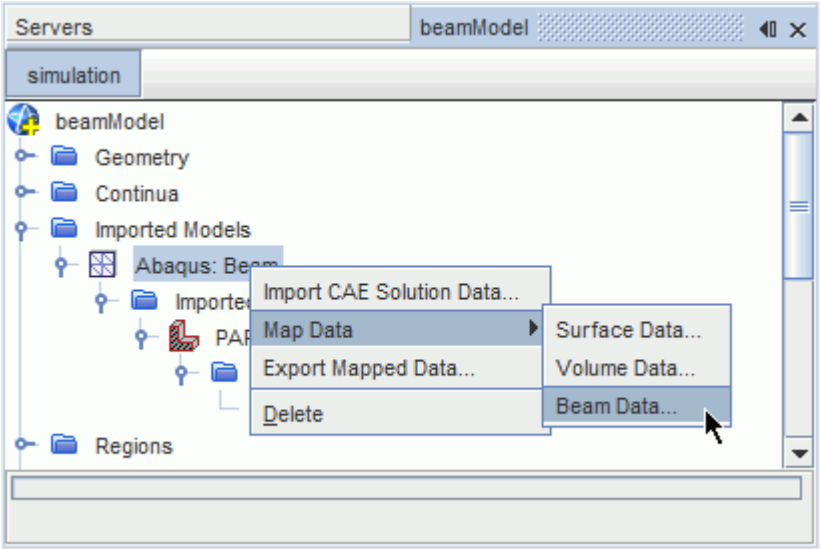
Nearest Neighbor Search Constraints

Mapping Imported Beam Data to STAR-CCM+ Boundaries

It is possible to map nodal displacement and rotation data, defined for the imported beam, onto the wetted surface in STAR-CCM+. Imported displacement and rotation data can be used as input for the morpher to deform the model.

To map beam data from the CAE model to STAR-CCM+:

1. Right-click the **[Imported CAE Model]** node and select **Map Data** > **Beam Data...**



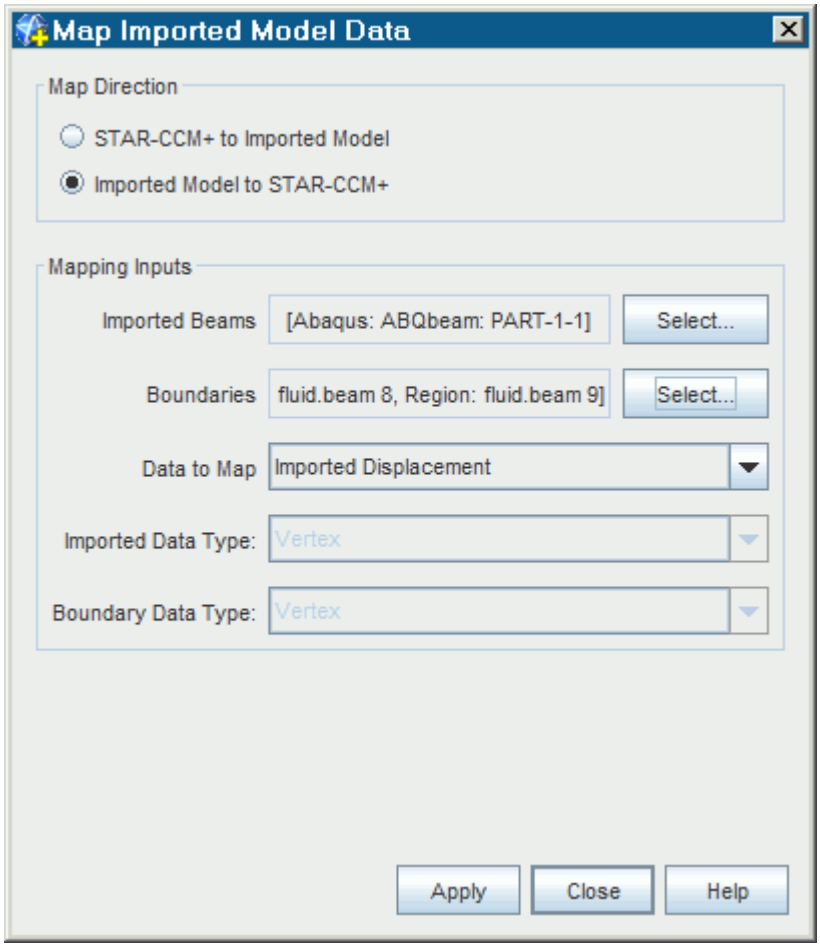
This activates the *Map Imported Model Data* dialog.

In the *Map Imported Model Data* dialog:

2. Set *Map Direction* to **Imported Model to STAR-CCM+**.
3. In the *Mapping Inputs* box, select the imported model beams you wish to map from, and the STAR-CCM+ boundaries you wish to map to.
4. The *Data to map* menu allows you to select which field data will be mapped.

   This menu displays field functions that are available for mapping in the simulation. When you select the field data, the *Imported Data Type* and *Boundary Data Type* are set automatically. Beams only support **Vertex** data.
5. When the settings in the dialog are complete, click **Apply** then **Close**.

   In the following image, the settings specify that imported displacement data will be mapped from the imported Abaqus model to multiple boundaries of the **fluid** STAR-CCM+ model.



See Also:

Deforming the CAE Model Using Imported Displacements

Exchanging Heat Transfer Coefficients

In convective heat transfer studies, the local heat flux at the wall is given by Newton's law of cooling, that is,

$$\dot{q}''_w = h\left(\mathrm{T}_w - \mathrm{T}_r\right) \tag{1377}$$

where h is the heat transfer coefficient, T_w is the wall temperature, and T_r is the fluid reference temperature. Examples of fluid reference temperatures are the near-wall temperature, or the mean fluid temperature. An important point is that the heat transfer coefficient and reference temperature are paired; there are many pairs that give the same wall heat flux.

In conjugate heat transfer calculations, where all calculations are part of the same simulation, there is no concern over the choice of heat transfer coefficient and reference temperature. In conjugate simulations the heat flux and wall temperature are calculated by the implicitly coupled system of equations. In co-simulation calculations, where the fluid and solid parts of the geometry are treated separately, you must choose the information to exchange at the interface between the simulations.

This section of the documentation is intended to help you choose the right approach for exchanging heat transfer coefficients.

Contents:

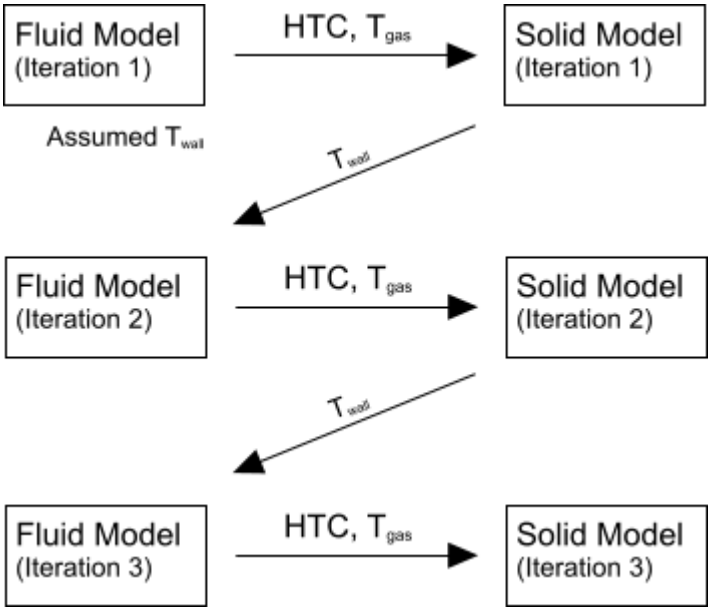
What Is a General Methodology For an Explicitly Coupled Heat Transfer Analysis?
What Methods Are Available for Exchanging Heat Transfer Coefficients?
Guidelines for Exchanging Thermal Data

What Is a General Methodology For an Explicitly Coupled Heat Transfer Analysis?

An explicitly coupled thermal analysis is one in which the non-isothermal flow problem is solved in one STAR-CCM+ simulation, and the solid thermal problem is solved in a second simulation either using STAR-CCM+ or another CAE code. This is typically done for analyses in which solid components exchange heat with a fluid whose properties vary in a cyclic manner, or cases in which the fluid and solid have disparate time scales.

An example for which this approach would be appropriate is an analysis of the heat transfer between the exhaust gases of an internal combustion engine (ICE) and its exhaust manifold. In such an analysis, the flow field leaving the engine will fluctuate in response to the exhaust valve dynamics. This means that the temperature of the fluid entering the exhaust manifold will experience rapid changes to the order of several hundred degrees. The solid body of the manifold will not respond so quickly to these temperature changes due to its greater thermal inertia relative to the fluid, so this results in a conjugate heat transfer problem where the time step that is appropriate for the fluid analysis is smaller than the time step that is appropriate for the solid thermal analysis -- hence the need to have two distinct simulations explicitly coupled by an exchange of thermal information on their common boundary. The solid thermal analysis will, in fact, tend towards a steady state solid conduction problem, as small fluctuations in solid temperatures during an engine cycle are not usually of interest.

A suggested methodology for conducting this analysis is illustrated in the figure below. Two separate simulations are set up, one for the fluid analysis and another for the solid thermal analysis.



In the first step, an assumed wall temperature is applied to the wall boundaries of the transient flow problem. This flow problem is solved for a total time equal to one engine cycle. It is important that this initial assumed wall temperature be reasonable in this step, as otherwise the solution may not converge. Following the simulation, a time-averaged, spatially varying, heat transfer coefficient and its corresponding reference temperature (both of which are calculated using field mean monitors and field functions) are calculated on the wall boundaries and then passed to the solid thermal simulation which is being run in steady state. To ensure that the time averaged heat transfer at the interface is conserved, the reference temperature (which is not the time averaged heat transfer reference temperature) that must be used is defined as:

$$\hat{\mathrm{T}}_r = \mathrm{T}_w - \frac{\overline{\dot{q}''}}{\overline{h}} \tag{1378}$$

where $\overline{\dot{q}''}$ is the time averaged interface heat flux and $\overline{h}$ is the time averaged heat transfer coefficient. The solid thermal simulation is then run to convergence.

In the second step, the wall temperatures predicted by the solid thermal simulation are passed back to the transient flow simulation and applied to the wall boundaries. The flow simulation is solved for a second time to obtain a second set of heat transfer coefficients and reference temperatures. This new set of data is passed to the solid thermal simulation for a second time.

This iterative process continues until successive steps show little or no meaningful change in the values of the predicted heat transfer coefficients or temperatures. Some situations may require many co-simulation cycles until the overall converged state is achieved.

A key question in this approach is: what heat transfer coefficient and reference temperature should be exchanged between the simulations? STAR-CCM+ provides several choices. For more information, see What Methods Are Available for Exchanging Heat Transfer Coefficients?

What Methods Are Available for Exchanging Heat Transfer Coefficients?

For turbulent flows, the central concept for modeling convective heat transfer at the wall is given by the standard wall function, which can be written as:

$$\dot{q}_w'' = \frac{\rho_f(y_c)C_{p,\,f}(y_c)u^*(y^+(y_c))}{\mathrm{T}^+(\mathrm{y}^+(\mathrm{y_c}))}\left(\mathrm{T}_w - \mathrm{T}_r\right) \tag{1379}$$

where y_c is the distance to the near wall cell center, y^+ is the non-dimensional wall distance, u^* is the reference velocity, and T^+ is the non-dimensional temperature -- both of which are determined by the chosen wall treatment. ρ_f and $C_{p,\,f}$ are the density and specific heat of the fluid. Note that ρ_f, $C_{p,\,f}$, u^*, and T^+ are functions of the distance from the wall.

Comparing Eqn. (1379) with Eqn. (1377), the heat transfer coefficient can be written as:

$$h\left(y_c\right) = \frac{\rho_f(y_c)C_{p,\,f}(y_c)u^*(y^+(y_c))}{\mathrm{T}^+(\mathrm{y}^+(\mathrm{y_c}))} \tag{1380}$$

This expression is the foundation for various definitions of heat transfer coefficient in STAR-CCM+. These coefficients, and their corresponding fluid reference temperatures, are listed in the table below.

Field Function	Reference Temperature for Data Exchange
Local Heat Transfer Coefficient	Local Heat Transfer Reference Temperature
Specified Y+ Heat Transfer Coefficient	Specified Y+ Heat Transfer Reference Temperature
Heat Transfer Coefficient	User-defined reference temperature
Virtual Local Heat Transfer Coefficient	N/A

Local Heat Transfer Coefficient

This is the heat transfer coefficient evaluated at the centroid of the cell next to the boundary, that is $h(y = y_c)$. It is given by:

$$h_{\mathrm{local}} = \frac{\rho_f(y_c)C_{p,\,f}(y_c)u^*}{\mathrm{T}^+(\mathrm{y}^+(\mathrm{y_c}))} \tag{1381}$$

The corresponding fluid reference temperature is given by:

$$\mathrm{T}_{\mathrm{local}} = \mathrm{T}_w + \frac{\dot{q}_w''}{h} \tag{1382}$$

Specified Y+ Heat Transfer Coefficient

This is the heat transfer coefficient evaluated at a user specified value of y^+, such that $y^+(y) = \mathrm{Y}^+_{\mathrm{user}}$. It follows that:

$$h_{y^+} = \frac{\rho_f\left(y_c\right)C_{p,\,f}\left(y_c\right)u^*}{\mathrm{T}^+\left(\mathrm{Y}^+_{\mathrm{user}}\right)} \tag{1383}$$

The corresponding fluid reference temperature, the Specified Y+ Heat Transfer Reference Temperature is given by:

$$\mathrm{T}_{f,\,y^+} = \mathrm{T}_w + \frac{\dot{q}''_w}{h_{y^+}} \tag{1384}$$

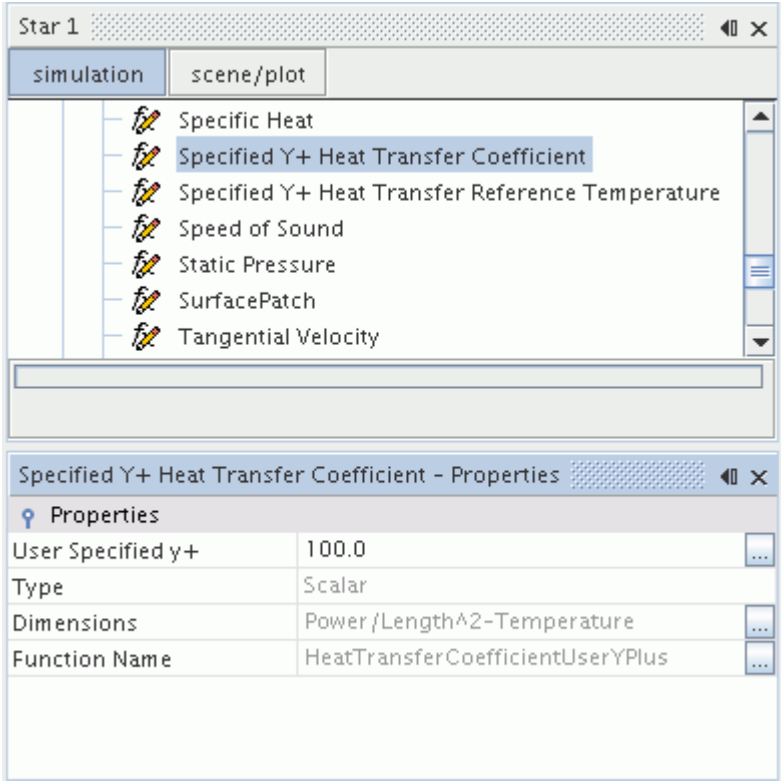
As this definition does not guarantee that $\mathrm{T}_{f,\,y^+}$ lies within the temperature range of the local region, the following conditions are enforced to give realistic values:

$$\mathrm{T}_{f,\,y^+} = \begin{cases} \mathrm{T}_w + \dfrac{\dot{q}''_w}{h_{y^+}} & \mathrm{T}_{\min} \le \mathrm{T}_w + \dfrac{\dot{q}''_w}{h_{y^+}} \le \mathrm{T}_{\max} \\ \mathrm{T}_{\min} & \mathrm{T}_w + \dfrac{\dot{q}''_w}{h_{y^+}} < \mathrm{T}_{\min} \\ \mathrm{T}_{\max} & \mathrm{T}_w + \dfrac{\dot{q}''_w}{h_{y^+}} > \mathrm{T}_{\max} \end{cases} \tag{1385}$$

In the latter two cases, h_{y^+} will be re-evaluated such that

$$h_{y^+} = \frac{\dot{q}''_w}{\left(\mathrm{T}_{f,\,y^+} - \mathrm{T}_w\right)} \tag{1386}$$

For these field functions, the y^+ value is set on the **Specified Y+ Heat Transfer Coefficient** field function node within **Tools** > **Field Functions**.

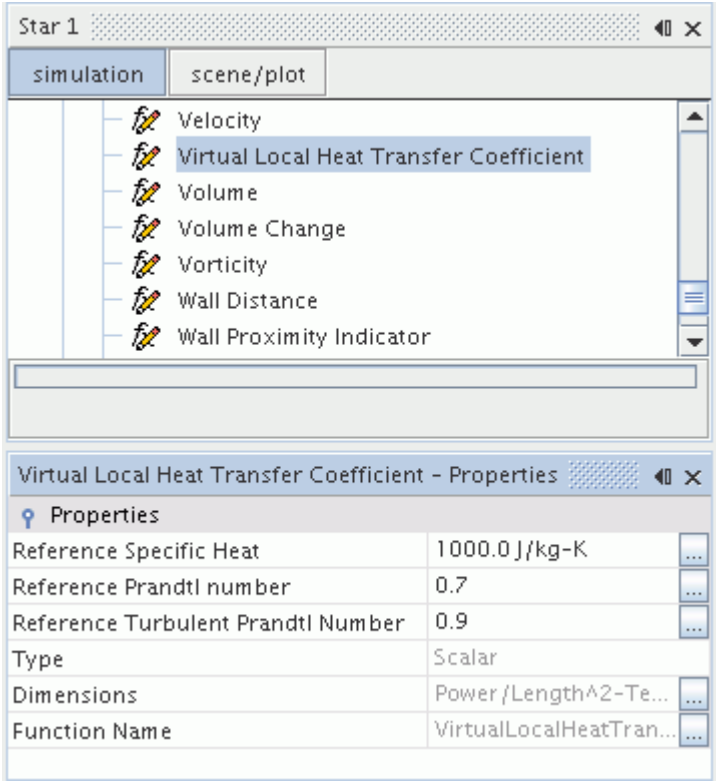


Virtual Local Heat Transfer Coefficient

This is an approximation to the true Local Heat Transfer Coefficient that does not require an energy model to be activated. It is defined as the heat transfer coefficient evaluated at $y = y_c$, where y_c is the distance of the near wall cell center from the wall.

$$h_{\text{virtual}} = \frac{\rho_f(y_c)C_{p,\,f}(y_c)u^*}{\mathrm{T}^+(y^+(y_c))} \tag{1387}$$

Since thermal material properties are not available, the specific heat, $C_{p,\,f}$ is specified as a reference value, as well as the molecular and turbulent Prandtl numbers. These are set on the **Virtual Local Heat Transfer Coefficient** field function node, as shown below. The fluid density, ρ_f is obtained locally from the simulation.



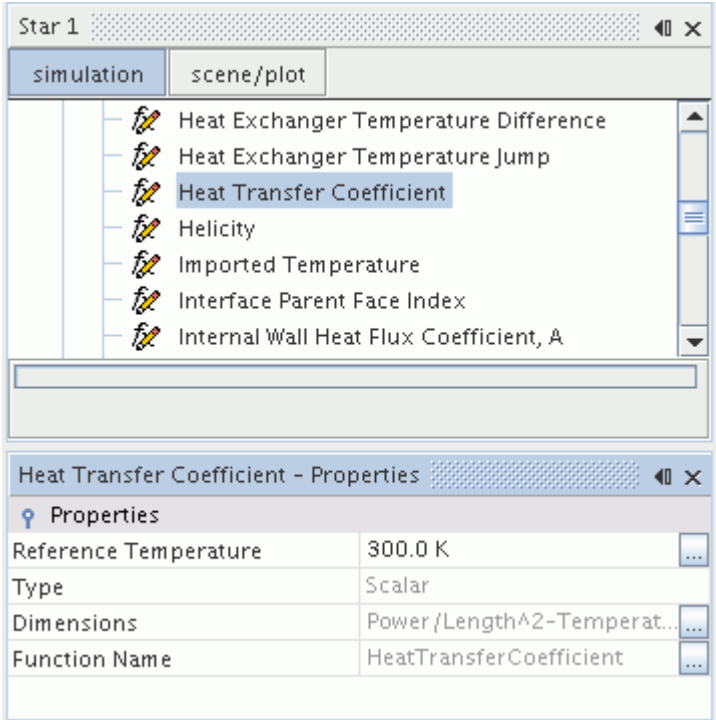
Heat Transfer Coefficient

This is yet another choice for defining heat transfer coefficients. It is evaluated based on a specified bulk temperature $\mathrm{T}_{\mathrm{bulk}}$.

$$h_{\mathrm{bulk}} = \frac{\dot{q}''_w}{(\mathrm{T}_{\mathrm{bulk}} - \mathrm{T}_w)} \tag{1388}$$

This definition does not guarantee that the evaluated heat transfer coefficient is positive. When exchanging data with other CAE codes, if the evaluated h_{bulk} is not positive, then the local heat transfer coefficient, h_{local}, and its corresponding reference temperature, $\mathrm{T}_{\mathrm{local}}$, are exchanged instead.

The bulk temperature is set as a property of the **Heat Transfer Coefficient** field function node.



Guidelines for Exchanging Thermal Data

The following guidelines should govern the exchange of thermal data in a file-based coupling of STAR-CCM+ with another CAE code:

1. When explicitly coupling a fluid and a solid simulation, the suggested practice is for:
   - the fluid model to provide spatially varying heat transfer coefficients and their corresponding reference temperatures to the solid model, and,
   - the solid model to return the local wall temperatures to the fluid model.

   In order to avoid issues with numerical stability, you are advised to avoid alternative techniques of exchanging thermal boundary data between STAR-CCM+ and other applications.

   Two specific scenarios are:

   a. For the case of fluid flow with, for example, no radiative transport or phase-change, use the Specified y+ Heat Transfer Coefficient together with the Specified y+ Heat Transfer Reference temperature. A relatively large value of y^+ such as 100 should be specified, in which case, the local temperature difference $(\mathrm{T}_w - \mathrm{T}_r)$ in Eqn. (1377) will be relatively large due to the relatively large wall distance. This in turn should allow the solid temperatures room to change significantly, and so allow the models to converge to a steady state in fewer exchanges between them.

   b. For the case of fluid flow with radiative transport and/or phase-change, use the Local Heat Transfer Coefficient together with the Local Heat Transfer Reference Temperature.

2. Exchanging local heat fluxes in one direction (whether fluid to solid, or solid to fluid), and receiving local wall temperatures in return, is not a recommended practice for reasons of numerical stability.

3. More than a single round of data exchange between the fluid and solid models will be required. A minimum of two rounds is recommended. This is because the initial guess of wall temperature for the fluid model may have been significantly different from the final solution.
4. During the first round, the initial temperature distribution in the solid model should be the same as the initial wall temperature used in the fluid model.
5. As either the fluid or the solid model must be solved first in the initial round, it is suggested that the fluid model be solved first. An initial guess for the value of wall temperature is required.
6. When comparing the heat transfer coefficient predictions with benchmarks (either measurements or model predictions), it is important to ensure that the heat transfer reference temperature is the same as in the benchmark. Otherwise the comparisons of heat transfer coefficient can be significantly different.
7. In a thermo-fluids CFD analysis, the energy equation should be solved (which is usually relatively cheap, and should always be done when circumstances allow, especially when temperature-dependent properties are involved as in buoyant flows).
8. When the energy equation is not solved, the local heat transfer coefficient is called the Virtual Local Heat Transfer Coefficient.
9. When using a low-y+ mesh for turbulent flows, you should reduce the thickness of the prism cells adjacent to walls until the fluxes at the walls become grid independent.
10. For laminar flows, you should reduce the thickness of the prism cells adjacent to walls until the fluxes at the walls become grid independent.
11. As the near wall region is coupled to the flow in the interior of the domain, it is important that the interior solution is also grid independent within practical constraints.

Working with Specific CAE Products

The following sections discuss the specific aspects of file-based coupling with each of the following software packages:

- Abaqus
- ANSYS
- NASTRAN
- RadTherm
- es-ice or pro-STAR

Also covered are the following topics:

- Use of Ensight input files for comparing multiple solutions in the same simulation. See Working with Ensight.
- The special format available in STAR-CCM+, .sbd (STAR-CCM+ Boundary Data). See Working with the STAR-CCM+ Boundary Data (.sbd) Format.
- The Visualization Toolkit (VTK) format. See Working with the Visualization Toolkit (VTK) Format.

The supported versions of these software packages are listed in the Release Notes.

Working with Abaqus

This section describes how to couple STAR-CCM+ with Abaqus using file-based coupling.

Contents:

Preparing Data in Abaqus
Importing Abaqus Solution Data
Exporting Data to Abaqus

Preparing Data in Abaqus

The Abaqus geometry data is imported from a standard ASCII-format Abaqus keyword input file (`.inp`). This data is for surfaces and volume elements that are defined in Abaqus. Surfaces are defined as a set of faces of 3D continuum elements and/or faces of shell elements. Volumes are defined as sets of 3D continuum elements. STAR-CCM+ supports surface-to-surface and volume-to-volume mapping, and the meshes do not need to be conformal.

Before proceeding in STAR-CCM+, first prepare the input file in Abaqus as follows:

1. Define the Abaqus surfaces for data to be mapped, using the `*Surface` keyword command. Generate this surface using the **Tools** > **Surfaces** > **Create** dialog in Abaqus/CAE.
2. For volume import and mapping, define nodes and elements in the `.inp` file (using the `*NODE` and `*ELEM` commands).

   STAR-CCM+ automatically imports them into the simulation and create imported volume objects.

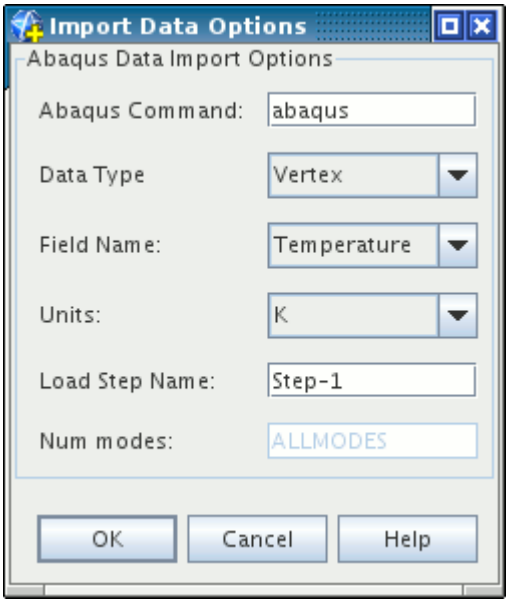
   Data can then be mapped to the surface and/or volume element from selected parts of the STAR-CCM+ mesh.

See Also:

Importing a CAE Model: Workflow

Importing Abaqus Solution Data

To import data from Abaqus, you must have access to an Abaqus installation. In the *Import Data Options* dialog for an Abaqus model, additional options are required in order to set up the import operation. The dialog shown in the example below is for importing Temperature data.



To set the additional options, do the following:

1. In the *Abaqus Command* text box enter the command you use to launch Abaqus, for example `abq6141`.

   Typically this is based on your installation path for Abaqus. On Windows the command is `abaqus.bat`.
2. In the *Load Step Name* text box, enter the name of the step in the Abaqus analysis job for which you would like to import data.

   The default name for the first load step in an Abaqus input file is `Step-1`.

When you click **OK** the data is imported into STAR-CCM+. During this process STAR-CCM+ launches Abaqus in the background. If you do not have access to an Abaqus installation, an error message is displayed.

Exporting Data to Abaqus

When the export action is executed, an Abaqus `.inp` file is produced that contains mapped data from the STAR-CCM+ solution. This file is a standard ASCII-format keyword-driven Abaqus file.

STAR-CCM+ automatically writes the corresponding keyword for the type of field data that is exported from the CFD model at the top of the data set:

- Pressure loads are defined via the `*DLOAD` keyword.
- The heat flux is defined via the `*DFLUX` keyword.
- Film coefficient and temperatures are defined via the `*FILM` keyword.
- Temperatures on surfaces are defined via the `*BOUNDARY` keyword for use in a heat transfer analysis.
- Temperatures on a volume are defined via the `*TEMPERATURE` keyword, which can only be used in an uncoupled temperature-displacement analysis. It cannot be used for heat transfer or coupled temperature-displacement analyses.

You can use the exported field data from STAR-CCM+ in an analysis job by adding an extra line in the job file (`.inp`) that instructs Abaqus to read in the load data. This line uses the `*INCLUDE` keyword, and is added in

the history section of the input file in the appropriate load step. For example, the following line would be added between the `*STEP` and `*End Step` keywords for the step in which the load data is applied to the Abaqus model:

```
*INCLUDE, input=mappedHTC.inp
```

Where `mappedHTC.inp` is the name of a data file containing mapped heat transfer coefficients from STAR-CCM+.

See Also:

Exporting Mapped Data

Working with ANSYS

This section describes how to couple STAR-CCM+ with ANSYS using file-based coupling.

Contents:

Preparing Data in ANSYS
Importing ANSYS Solution Data
Mapping the STAR-CCM+ solution to the ANSYS mesh
Exporting to ANSYS

Preparing Data in ANSYS

An imported model from ANSYS contains the full FEA mesh as well as FSI interfaces for mapping data to and from STAR-CCM+ and ANSYS. The wetted FSI interfaces are defined through Components of Nodes and Elements in ANSYS, which are named lists of nodes or elements. Use either surface or volume FSI interfaces.

Defining Surface and Volume Interfaces using Nodes and Elements

Node components can only be used to specify the FSI surface interfaces. The nodes must define an external (free) surface on the FEA model (use the EXT item of the ANSYS command NSEL), and only element faces for which all the nodes are selected are included in the surface definition. Node components are the only method for defining FSI surfaces on ANSYS SOLID elements.

It is recommended that element components are used to define FSI surface interfaces on SHELL elements. Element components are also used to define FSI volume interfaces for collections of SOLID elements. Both wall shear stress and pressures can be used with this component type.

For a Node or Element component to be imported for mapping results, include the string “FSI” or “fsi” somewhere in its name. When the ANSYS model is imported into the simulation, STAR-CCM+ recognizes these components and display them in the object tree; other components are automatically filtered out.

ANSYS Element Types

The following are the list of all supported ANSYS elements:

Type	Element Name	Description
3D Shells	SHELL43	4-Node Plastic Large Strain Shell
	SHELL63	4-Node Elastic Shell
	SHELL93	8-Node Structural Shell
	SHELL131	4-Node Layered Thermal Shell
	SHELL132	8-Node Layered Thermal Shell
	SHELL143	4-Node Plastic Small Strain Shell

Type	Element Name	Description
	SHELL150	8-Node Structural Shell p-Element
	SHELL163	Explicit 4-Node Thin Structural Shell
	SHELL181	4-Node Finite Strain Shell
	SHELL281	8-Node Finite Strain Shell
3D Continuum Solid	SOLID5	8-Node Coupled-Field Solid
	SOLID45	8-Node Structural Solid
	SOLID46	8-Node Layered Structural Solid
	SOLID62	8-Node Magneto-Structural Solid
	SOLID64	8-Node Anisotropic Structural Solid
	SOLID65	8-Node Reinforced Concrete Solid
	SOLID69	8-Node Coupled Thermal-Electric Solid
	SOLID70	8-Node Thermal Solid
	SOLID72	4-Node Linear Tet, obsolete
	SOLID73	8-Node Linear Brick, obsolete
	SOLID87	10-Node Tetrahedral Thermal Solid
	SOLID90	20-Node Thermal Solid
	SOLID92	10-Node Tetrahedral Structural Solid
	SOLID95	20-Node Structural Solid
	SOLID98	Tetrahedral Coupled-Field Solid
	SOLID147	Brick Structural Solid p-Element
	SOLID148	Tetrahedral Structural Solid p-Element
	SOLID164	Explicit 3D Structural Solid
	SOLID168	Explicit 3D 10-Node Tet Structural Solid
	SOLID185	8 Structural Solid or Layered Solid
	SOLID186	20 Structural Solid or Layered Solid
	SOLID187	10-Node Tetrahedral Structural Solid
	SOLSH190	8-Node Layered Solid Shell

Type	Element Name	Description
	SOLID191	20-Node Layered Structural Solid
	SOLID226	20-Node Coupled-Field Solid
	SOLID227	10-Node Coupled-Field Solid
	VISCO89	20-Node Viscoelastic Solid
	VISCO107	8-Node Viscoplastic Solid
Surface Elements 3D	SURF152_8_N	See notes below.
	SURF154_8_N	See notes below.

In ANSYS, tractions are applied to special “Surface” element types: SURF154. This is the only element type that can have a traction other than pressures in a three-dimensional analysis.

Thermal loads are applied directly to solid element faces but can also be applied to special “Thermal” element types: SURF152.

There are two methods of defining a fluid-structure interaction (FSI) surface of these SURFACE elements:

1. As an ANSYS component of Elements, with the letters “FSI” in the name.
2. As a surface node set like all of the other surface definitions for solid elements, with the letters “FSI” in the name.

The first method is the most straightforward. With method (2), using a node set, there are two surfaces created: one from the faces of the solid elements, and one from faces of the surface elements. In this case, “-SOLID” are appended to the name of the surface comprised of solid element faces, and “-SURF” are appended to the name of the surface that is comprised of surface elements.

Exporting using the .cdb file

ANSYS models are imported into STAR-CCM+ in the `.cdb` format. When the file is generated via the CDWRITE command in ANSYS, only some of the options write enough information so that STAR-CCM+ can reliably read the file.

Use the following options:

CDWRITE, **DB**, <filename>.cdb

CDWRITE, **COMB**, <filename>.cdb

CDWRITE, **ALL**, <filename>.cdb

Do not use the following options:

CDWRITE, **GEOM**, <filename>.cdb

CDWRITE, **CM**, <filename>.cdb

CDWRITE, **LOAD**, <filename>.cdb

CDWRITE, **SOLID**, <filename>.cdb

Further Considerations for Model Preparation

The Imported Model FSI Surface for a single ANSYS component is divided into multiple surfaces by element style: SOLID, SHELL, and/or SURF. If the ANSYS component name is “fsi_surface”, then the FSI surface names in the STAR-CCM+ imported model are:

Element Style	Example Surface Name
SOLID	`fsi_surface-SOLID_FACE`
SURF	`fsi_surface-SURF-EL`
SHELL	`fsi_surface-SHELL:POS` `fsi_surface-SHELL:NEG`

This is useful for solid models with Node components, where SURF or SHELL elements are overlaid on the SOLID free surface. This allows you to apply traction loads to the SURF element set and heat flux loads to the SOLID element set in the same simulation.

For each SHELL element, note that there are two sides: a positive and a negative side. These are determined by the right-hand rule applied to the nodal sequence of the element definition. If you know that the side of a SHELL element is either positive or negative, you can include “POS” or “NEG” in the component name, and the Imported Model only includes one surface for the SHELL elements, for example either “fsi_surface-SHELL:POS” or “fsi_surface-SHELL:NEG”.

See Also:

Importing a CAE Model: Workflow

Importing ANSYS Solution Data

Displacement, eigenmode and eigenvector, and temperature results can be imported from ANSYS listing files for an ANSYS model into a STAR-CCM+ simulation. Write the results to a `.lis` file from ANSYS in order for STAR-CCM+ to import them.

Before listing results in ANSYS, load the results using the following commands in ANSYS:

`/POST1`	Enter the ANSYS Post Processor.
`SET,LS,SS`	Set load step and substep to load.
`INRES,NSOL`	Load the nodal solution for the selected load step, substep combination.

To create the listing file, the following commands are needed:

`/OUTPUT,fname,ext`	Open the file `fname.ext` for output.
`PRNSOL,NAME,OPT`	Print the output to `fname.ext`.
`PRNSOL,TEMP`	Print temperature to the file.

PRNSOL,U,COMP	Print the three displacement or eigenvector components to the file, depending on whether analysis is structural or modal.
/OUTPUT	An additional /OUTPUT command with no parameters to close the file and send further output back to the screen.

The above commands are general instructions for obtaining output data from ANSYS. Before importing the data into the STAR-CCM+ simulation, export the data to a `.lis` file (the default format for the /OUTPUT command in ANSYS).

Note: While STAR-CCM+ reads displacements and temperatures for a particular load step from a `.lis` file, it reads all eigenmode results in the file.

Example Temperature Listing creation:

/POST1	
SET,1	! set load step 1, (substep 1 is default)
INRES,NSOL	
/OUTPUT,ansys_temperature,lis	! open new file ansys_temperature.lis
PRNSOL,TEMP	
/OUTPUT	! turn off output to file

Example Displacement Listing creation:

/POST1	
SET,1	! set load step 1, (substep 1 is default)
INRES,NSOL	
/OUTPUT,ansys_disp,lis	! open new file ansys_disp.lis
PRNSOL,U,COMP	! print all displacement components
/OUTPUT	! turn off output to file

Example Eigenmode Listing creation for the first six modes of a modal analysis:

/POST1	
SET,1	! set load step 1, (substep 1 is default)
INRES,NSOL	
/OUTPUT,ansys_modes,lis	! open new file ansys_modes.lis
SET,FIRST	
PRNSOL,U,COMP	
SET,NEXT	

PRNSOL,U,COMP	
SET,NEXT	
PRNSOL,U,COMP	
SET,NEXT	
PRNSOL,U,COMP	
SET,NEXT	
PRNSOL,U,COMP	
SET,NEXT	
PRNSOL,U,COMP	
/OUTPUT	! turn off output to file

Mapping the STAR-CCM+ solution to the ANSYS mesh

In the STAR-CCM+ CAE mapping tools, map each fluid boundary to the associated ANSYS exterior surface detected using the **source/target proximity** checks. Depending on the conformality of the grids at the edges of the wetted area and the proximity tolerance, there may be some stretching of the wetted area onto non-wetted faces by the amount of the proximity tolerance.

See Also:

Mapping STAR-CCM+ Solution Data to a CAE Model

Exporting to ANSYS

Data is exported from STAR-CCM+ to ANSYS via ANSYS macro scripts. Load the data onto the ANSYS model with the appropriate load step activated.

The ANSYS command to input a macro script is “/INPUT,<filename><extension>”, or select **File** > **Read Input From**.

In the ANSYS macro scripts, data exported from STAR-CCM+ set values in the ANSYS model, as follows:

- For Surface interfaces:
  - Vertex temperatures are used to set ANSYS boundary conditions (ANSYS command: D).
  - Face values, including pressures, local heat transfer coefficient and reference temperature, and heat flux, are used to specify ANSYS surface loads (ANSYS command: SFE).
- For Volume interfaces:
  - Vertex temperatures are used as ANSYS loads or field variables (ANSYS command: BF).

See Also:

Exporting Mapped Data

Coupling with NASTRAN

You can couple STAR-CCM+ with Siemens NX Nastran using file-based coupling, for fluid-structure interaction (FSI) applications. In an FSI file-based coupled analysis, STAR-CCM+ solves for the fluid loads and Siemens NX Nastran solves for the response of the solid structure. You can export the field data computed in each code to external files that are then used in the partner code to set appropriate boundary conditions.

An example application is structural design exploration, where you test different structural designs in Siemens NX Nastran using the fluid loads computed in a single STAR-CCM+ run. In this example, the coupling is only one-way, that is, the structure response is not imported back into STAR-CCM+. For more information on file-based coupling, see Understanding File-Based Coupling.

The following sections assume that you already prepared the STAR-CCM+ fluid simulation and the Siemens NX Nastran structure model, as appropriate to your analysis. STAR-CCM+ supports the import of shells, volumes (solids), and surface elements from Siemens NX Nastran model text files with the `.nas`, `.bdf` or `.dat` file extensions.

Exporting STAR-CCM+ Data to NASTRAN

STAR-CCM+ computes the fluid loads at the fluid-structure interface. You export the computed loads to external files that Siemens NX Nastran can read.

Export the fluid loads in one of the following ways:

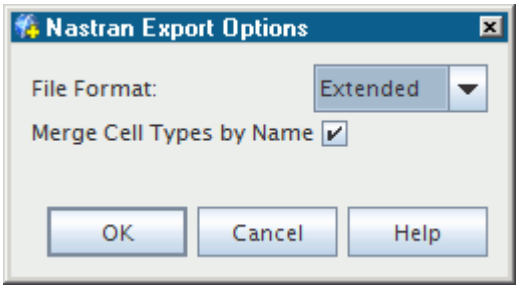
- In static analyses, first run the STAR-CCM+ simulation until converged. Then, export the computed fluid loads to a `.nas` Siemens NX Nastran file and use the file as a static load for structural analysis. See Exporting Data to NASTRAN.
- In transient analyses, you can export the fluid loads at regular intervals while the STAR-CCM+ simulation is running. The fluid loads are saved to a `.bdf` file. You then use a java macro, which is provided in the STAR-CCM+ installation, to convert the `.bdf` files to a transient load file that can be used in Siemens NX Nastran for structural analysis. See Exporting Data to NASTRAN (Transient Run).

Importing NASTRAN Solution Data

Once a Siemens NX Nastran model has been imported, its associated solution data can be imported into STAR-CCM+ from the OUTPUT2 (`.op2`) binary file format.

Exporting a Model to NASTRAN

When you export a model to Siemens NX Nastran you can choose if patches with the same name are merged or not. Activating *Merge Cell Types by Name* exports the boundary/part surfaces after merging the patches with the same name. This typically leads to one Property ID (PID) for the whole boundary/part surface instead of one PID per patch.



Contents:

Exporting Data to NASTRAN
Exporting Data to NASTRAN (Transient Run)

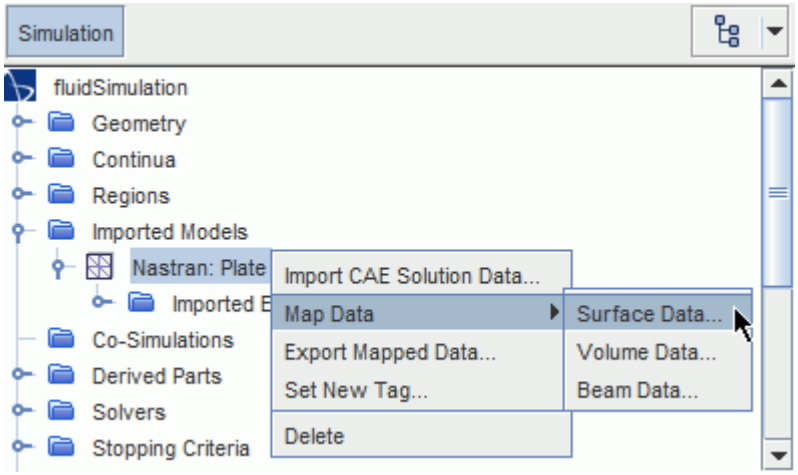
Exporting Data to NASTRAN

You can export the fluid loads obtained at the end of a STAR-CCM+ run to an external file, which can be read into Siemens NX Nastran to calculate the response of the solid structure.

1. Import the Siemens NX Nastran model into STAR-CCM+ (for instructions, see Importing a CAE Model).
2. Run the STAR-CCM+ simulation.

Map the computed fluid loads onto the imported Siemens NX Nastran model:

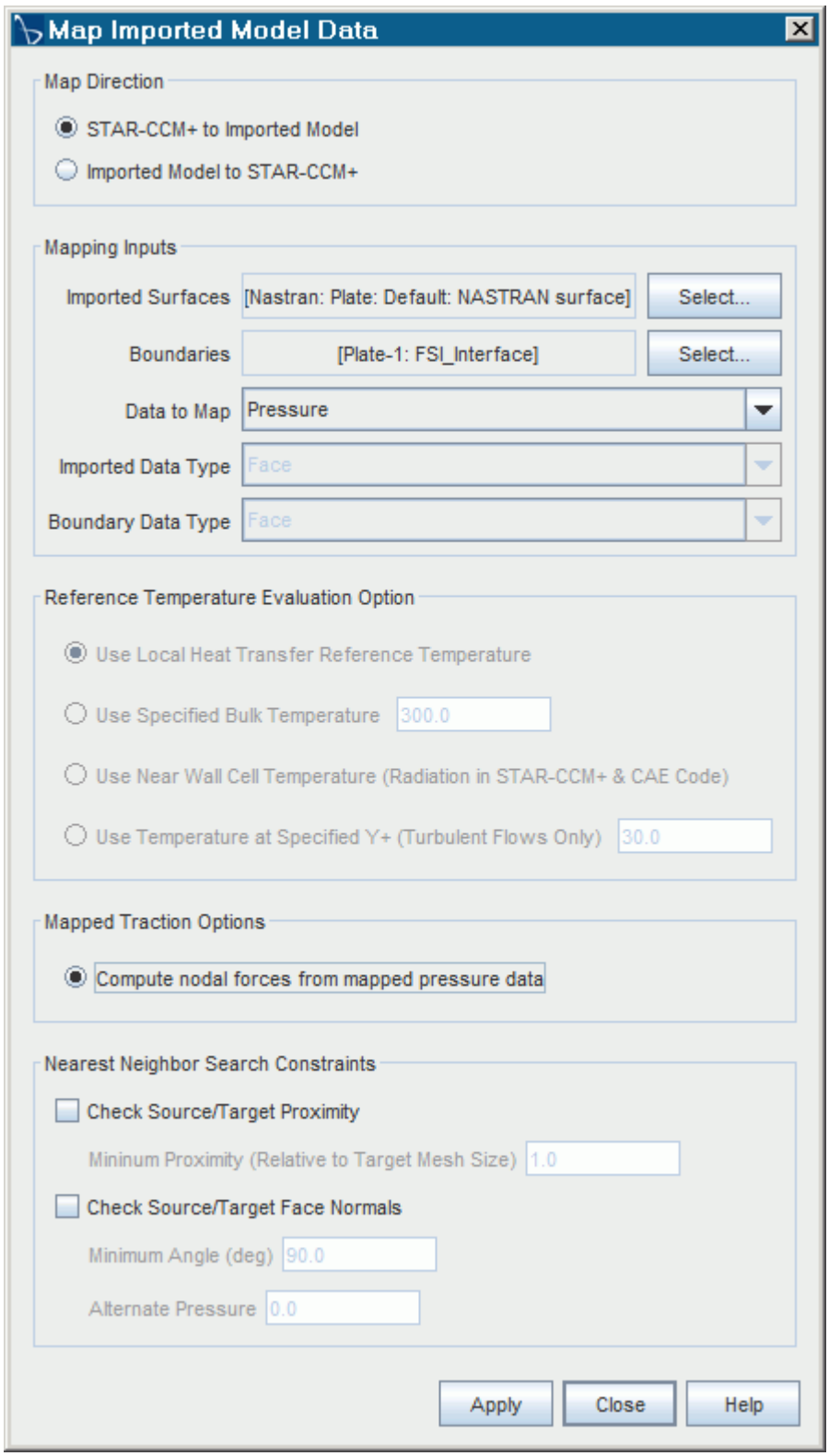
3. Right-click the imported Siemens NX Nastran model and select **Map Data** > **Surface Data...**



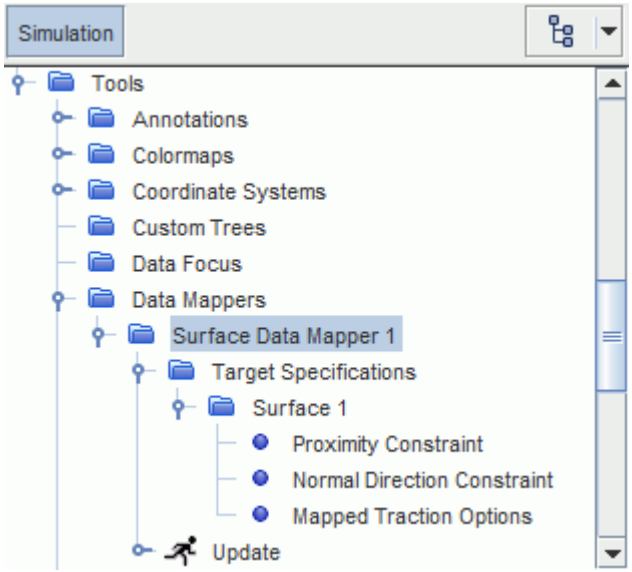
4. Set up the *Map Imported Model Data* dialog as explained in the section, Mapping Boundary Data to Imported Surfaces. In the dialog, set *Data to Map* to either temperature, pressure, or traction (that is, pressure and wall shear stress).

   When mapping pressure or traction data, you can instruct STAR-CCM+ to compute the corresponding nodal forces and map them onto the Siemens NX Nastran model vertices:

   - To map nodal forces, activate *Compute nodal forces from mapped pressure data* in the *Map Imported Model Data* dialog.

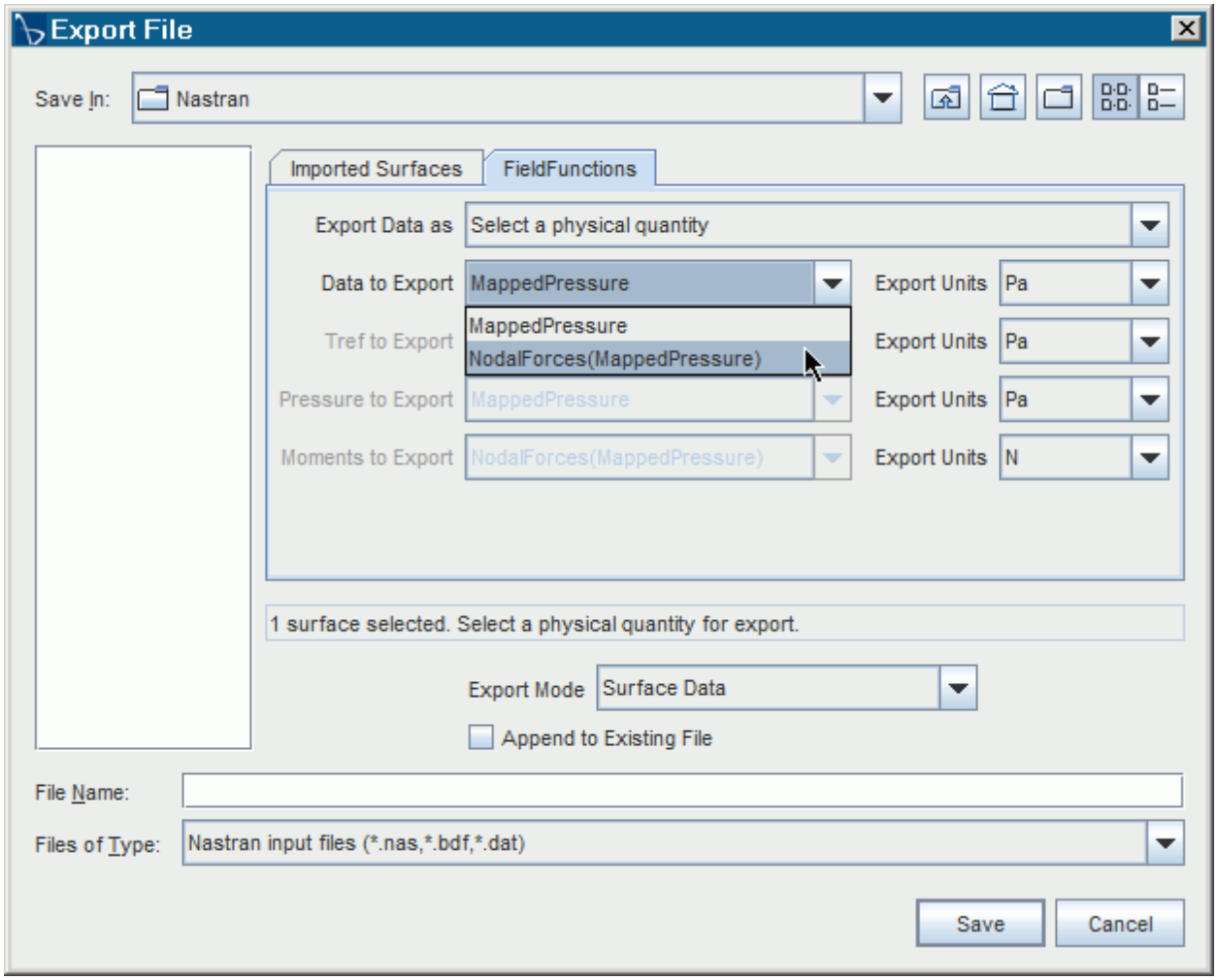


STAR-CCM+ automatically adds a surface mapper under the **Tools** > **Data Mappers** node that represents the mapping operation that you set up using the dialog. When you click **Apply** in the dialog, STAR-CCM+ automatically executes the mapping operation.



5. Export the mapped data to an external file following the instructions in the section, Exporting Mapped Data. Choose `.nas` as file extension.

   When computing nodal forces, you can choose to export either the pressure (or traction) surface data, or the nodal forces:



The exported `.nas` file is a text-based file that contains the Bulk Data keywords for the selected export quantity.

6. To use the exported data in a Siemens NX Nastran simulation, you can append it to the Siemens NX Nastran simulation file by using the `INCLUDE` keyword, for example `INCLUDE 'starccm_pressure.nas'`.

Exporting Data to NASTRAN (Transient Run)

You can export the fluid loads at regular intervals while the STAR-CCM+ simulation is running. You can then convert the exported `*.bdf` history file to a format that is specific to Siemens NX Nastran transient loads, for use in dynamic Siemens NX Nastran analyses.

1. Import the Siemens NX Nastran model into STAR-CCM+ (for instructions, see Importing a CAE Model).
2. Set up automatic mapping at a desired time interval using a surface data mapper. Instructions for creating a data mapper and setting up automatic mapping are given in the sections, Mapping Data from Surface, Volume, or Beam Sources and Executing the Mapping Operation.
3. In STAR-CCM+, select **File** > **Auto Export...**
4. In the *Auto Export* dialog:

a) Set *File Format* to **Nastran load history file (*.bdf)**.

b) Activate *Append to File*.

c) Make sure that *Export Solution Data Only* is **Deactivated**.

d) Specify the mapped data to export (for example temperature, pressure, or pressure and wall shear stress) using the *Scalar Functions* and/or *Vector Functions* properties.

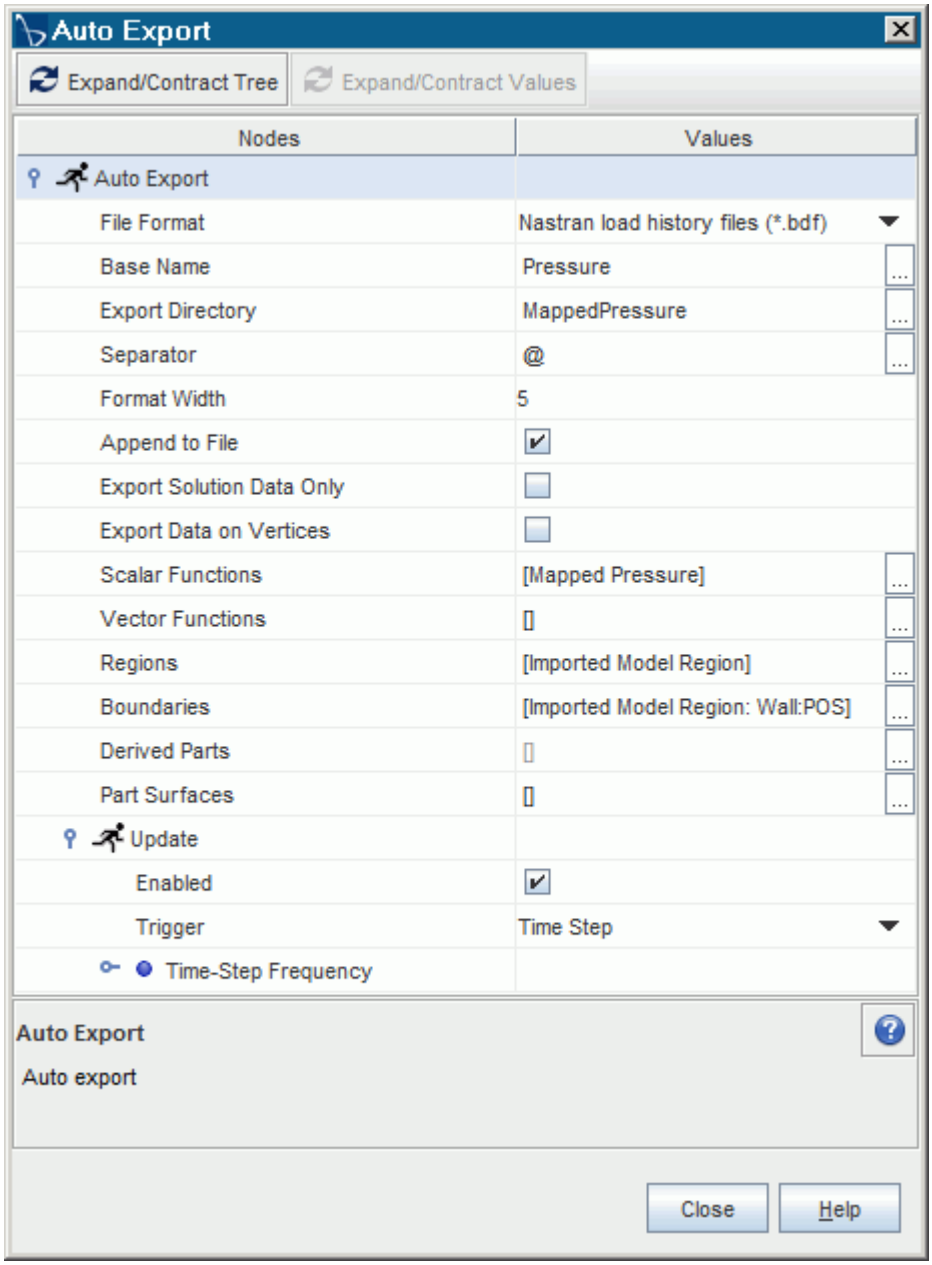
e) Set *Regions* to the region associated with the imported Nastran model.

f) Set *Boundaries* to the imported model boundaries where you mapped the fluid loads.

g) Expand the **Update** node and set *Trigger* to **Time Step**.

h) Expand the **Update** > **[Frequency]** node and specify the export frequency using the *Frequency* property.

i) Set any other properties as required. For more information, see Automatically Exporting Mesh and Solution Data at Regular Intervals.



5. Run the STAR-CCM+ simulation.

To convert the exported `*.bdf` file:

6. Navigate to `[INSTALL_DIR]/star/cae/bin` and save a copy of the Java macro, `NastranTLoadConverter.java`, in the same directory of the `*.bdf` file.
7. Open the macro with a text editor of your choice and specify the name of your input `*.bdf` file.

```
final String io_home = System.getProperty("user.dir");   // working
dir., in/out file location
    final String baseName = "exportHistory3";              //
AutoExport history file base name
    final String extName  =  ".bdf";
```

8. Run the Java macro.

Working with TAITherm

This section describes how to couple STAR-CCM+ with TAITherm using file-based coupling.

Contents:

Preparing a Model in TAITherm
Importing the TAITherm Model
Importing Temperature Data from TAITherm
Mapping and Exporting to TAITherm
Analyzing with TAITherm
Known Limitations

Preparing a Model in TAITherm

To be imported into STAR-CCM+, export a TAITherm file to the Patran Neutral file format.

A common technique is the TAITherm command-line utility, `tdf2patran` (`tdf2patran.exe` on Windows) that is bundled with TAITherm. In this utility you would execute the following at the command prompt:

```
tdf2patran -temp -v -f back <RadThermFileName>.tdf
```

The `-temp` and `-v` options above are required to ensure that nodal temperatures are exported, rather than the temperature at the surface cell centroids. The `-f back` (or `front`) option can be omitted to export both the front/back nodal temperatures.

A quick inspection of the generated file is recommended so as to ensure that at a minimum, the generated file contains the following Patran “packets”, which are required for successful import into STAR-CCM+:

- Packets 1 & 2 for the nodal coordinates and element connectivity/type respectively
- Packet 4 for the element/nodal properties that are associated with the parts
- Packet 10 for the exported nodal temperatures at both the front/back faces of each shell
- Packet 21 for the parts definition, which consists of listings of the element type, ID for each exported part or component, and the part name

By default the TAITherm Patran utility exports thermal quantities at both the front/back surfaces of a shell or component. A negative temperature value of `-990` is assigned to the nodes of a face that does not have a user-specified temperature during model export. However, you can select either the front or back shell surface when you map the surface data.

See Also:

Mapping CAE Solution Data to a STAR-CCM+ Model

Importing the TAITherm Model

After you launch STAR-CCM+ and start a simulation, you can import the TAITherm model (`.ntl` or `.neu`) in one of two ways:

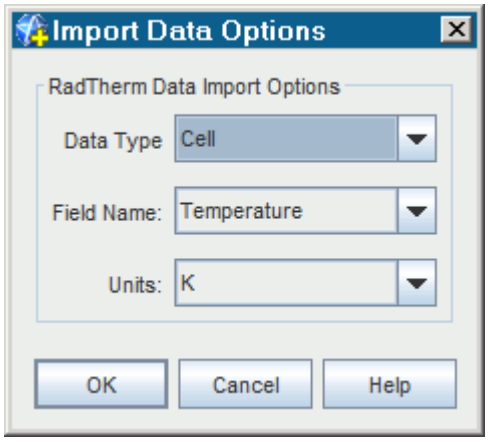
- If the CFD model is being developed from scratch, then select **File** > **Import Surface Mesh...** to import the TAITherm model as a surface mesh or geometry part. This import option is useful if you wish to generate a volume mesh from the TAITherm model, for example to compute a fluid solution for a volume that is adjacent to the TAITherm shells.
- If the STAR-CCM+ simulation model is already loaded, then select **File** > **Import CAE Model...** to load the TAITherm model file. This option allows you to import and map solution data for the TAITherm model. Note that the **Create new Region** option in the *Import Options* dialog cannot be used for TAITherm model import, as it does not apply to shells.

To set up a fluid-structure interaction case, it may be necessary to use the first import method to create the fluid volume, then use the second method to import the structural shells. File-based coupling techniques can then be used to couple the STAR-CCM+ and TAITherm models.

Importing Temperature Data from TAITherm

This section describes how to import temperature data from TAITherm. In TAITherm models, the solution data is available at face centroids.

1. Right-click the **Imported Models** > **[TAITherm Model]** node and select **Import CAE Solution Data...**
2. In the *Open* dialog, set *Files of Type* to **RadTherm Patran Neutral files (*.ntl, *.neu)**, and then select the TAITherm output file containing the solution data.
3. In the *Import Data Options* dialog, set *Data Type* to **Cell**.



   Avoid setting *Data Type* to **Vertex**. If you choose **Vertex**, before importing the solution, STAR-CCM+ interpolates the data from the face centroids to the vertices of the TAITherm model. When you then map the imported data onto the STAR-CCM+ model, the mapper has to reinterpolate the vertex data to the face centroids.
4. Click **OK**.

   The imported temperature becomes available as a field function in STAR-CCM+.

Mapping and Exporting to TAITherm

Solution data from STAR-CCM+ can be mapped and exported to the imported TAITherm model.

When you map surface data from STAR-CCM+ to the imported model, you can select which shell surface to export the data for. To do this:

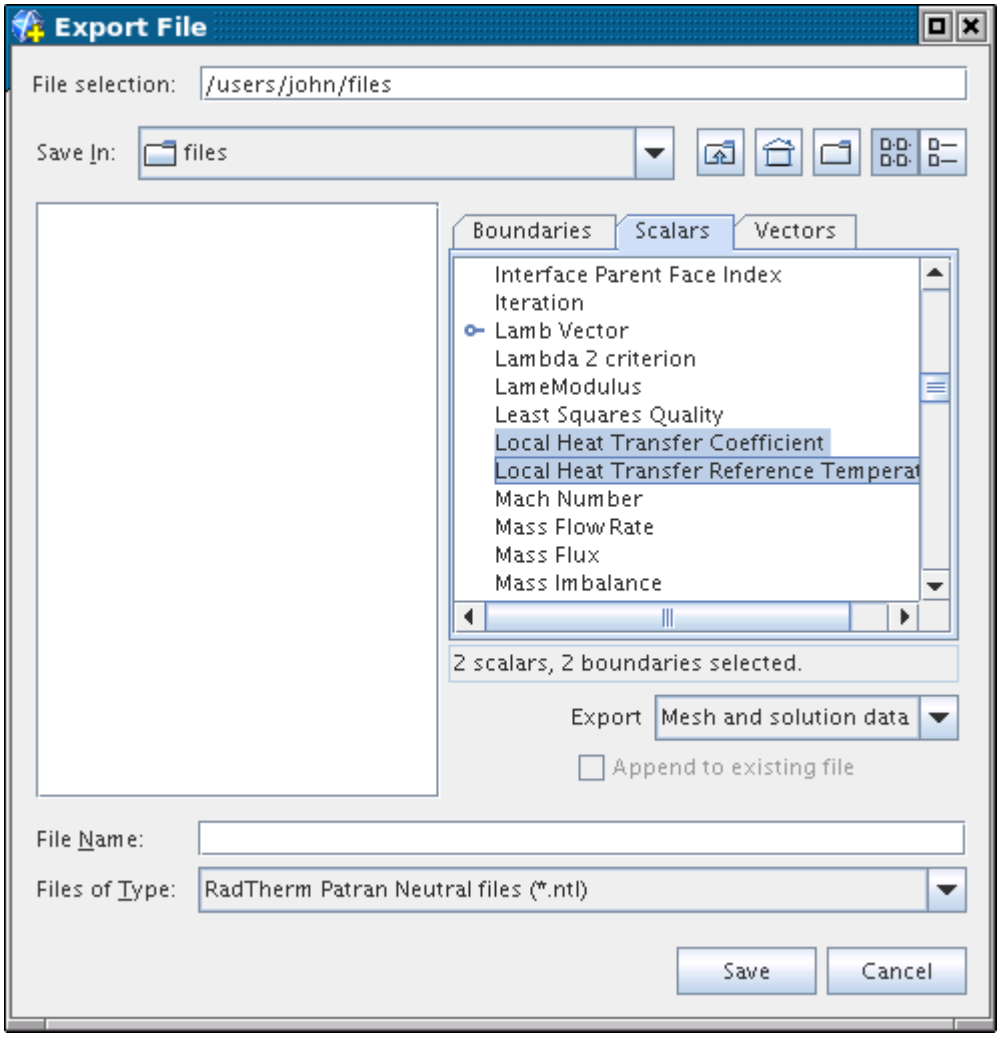
1. Right-click the **Imported Models** > **[TAITherm Model]** node and select **Map data** > **Surface Data...**
2. In the *Map Imported Model Data* dialog, set *Imported Surfaces* to one of the shell surfaces, referred to as **Front** and **Back**.

Currently it is only possible to map heat transfer coefficient data to the TAITherm model using the mapping dialog. However, if you wish to map a different field function, the full list of field functions can be made available by manually creating data mappers via the following steps:

- Setting data mapper properties
- Mapping the data
- Specifying CAE-specific properties

You can export solution data to Radtherm using similar techniques as for other CAE models. You can also export mesh and solution data from STAR-CCM+ to TAITherm:

3. Select **File** > **Export ...** to open the *Export File* dialog.



4. In this dialog, set up the export as follows:
5. Select the `RadTherm Patran Neutral Files (*.ntl)` option from the *Files of Type* drop-down list.
6. Select the interface boundaries of interest from the list in the *Boundaries* tab.
7. Switch to the *Scalars* tab and select `Local Heat Transfer Coefficient` and `Local Heat Transfer Coefficient Reference Temperature` from the list. Only these two quantities are currently supported for this direct type of export to TAITherm.
8. Enter a name for the output file and click **Save**.

   The generated file is ready for import into TAITherm.

See Also:

Mapping Boundary Data to Imported Surfaces

Mapping Fields Using Data Mappers

Exporting Mesh and Solution Data

Exporting Mapped Data

Analyzing with TAITherm

When you have finished the mapping work in STAR-CCM+, export the data to the Patran Neutral file format (`.ntl`). See Exporting Mapped Data.

Then do as follows:

1. Import the file into TAITherm.
2. Set up the boundary conditions and thermal parameters in TAITherm and run the analysis.
3. Prepare and export the results from the TAITherm file to the Patran Neutral file format. See Preparing a Model in TAITherm.
4. Import the data into STAR-CCM+.
5. Export the STAR-CCM+ fluid analysis results into the Patran Neutral file format.
6. Back in the TAITherm file, import the Patran file created by STAR-CCM+.
7. If there are no error messages after importing the geometry, use TAITherm's Import CFD Results feature to map the STAR-CCM+ results to the geometry of the TAITherm file.
8. Use TAITherm's *Model Parameter Display* feature to view the fluid reference temperature values, and make sure they are consistent with those generated by STAR-CCM+.

Known Limitations

The following are the known limitations of the current implementation:

- The exchange of data is limited to steady-state, or data at a single time-step. The import/export of thermal data at multiple time steps, or time-levels is not supported.
- The current implementation does not support the conversion of units of measurement. It is the user's responsibility to ensure consistency of units in both TAITherm and STAR-CCM+.
- It is currently not possible to import into STAR-CCM+, "dummy" Patran shells, which are defined as Packet-21 entries that have no element type, and ID attributes. Delete dummy entries from the Patran Neutral file after export from TAITherm, since they are not required for surface definition in STAR-CCM+.

Working with es-ice or pro-STAR

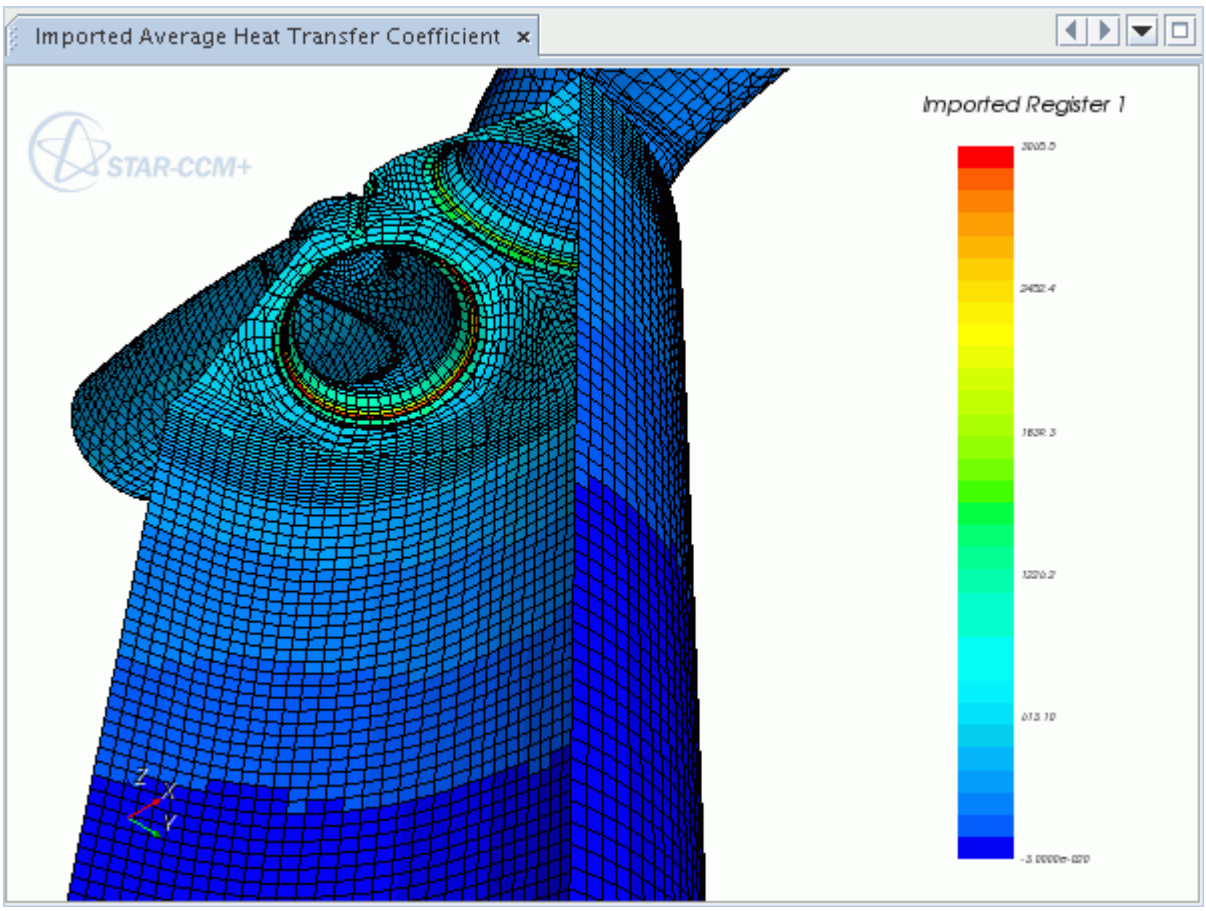
The `.dbs` file format is commonly used to import a surface mesh. It can also be used to map solution data from pro-STAR or es-ice to STAR-CCM+ by importing the file as a CAE model, rather than a surface mesh.

When used in this manner, store the solution data in a `.usr` file that was created by the same session of pro-STAR or es-ice that created the `.dbs` file.

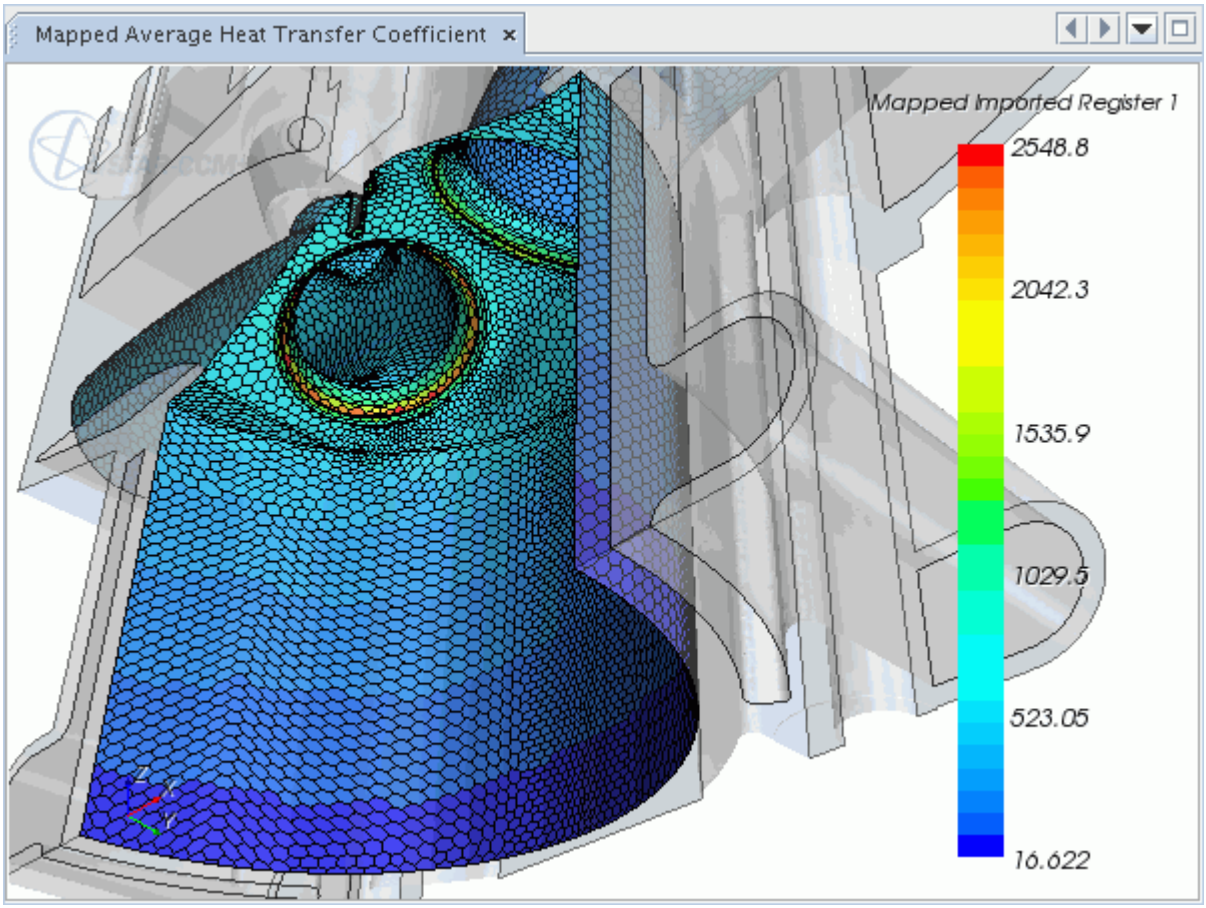
For pro-STAR, up to six sets of wall solution data can be stored in the six Registers available in the `.usr` file. STAR-CCM+ can import solution data from these six Registers to the CAE model, and then map the solution data on to boundaries.

For es-ice, two `.usr` files are used to export cycle-averaged wall heat transfer data.

An example of the es-ice cycle-averaged heat transfer coefficient displayed as a scalar on the imported CAE model is shown below.



The following image shows the same imported heat transfer coefficient mapped onto the boundaries of a STAR-CCM+ volume mesh.



Contents:

Preparing Wall Heat Transfer Data in es-ice
Preparing Wall Data in pro-STAR
Importing the es-ice or pro-STAR CAE Model
Importing the es-ice or pro-STAR Solution Data
Mapping data from es-ice or pro-STAR to STAR-CCM+

Preparing Wall Heat Transfer Data in es-ice

Calculating spatially varying, cycle-averaged wall heat transfer data in es-ice produces CAE data, which can be imported into STAR-CCM+. This es-ice post-processing facility generates a `.dbs` file, containing the CAE model and calculates eight data-sets.

As the `.usr` file can only contain six data-sets at most, es-ice writes two `.usr` files, with the data described below:

- Data file 1:

Register 1: Average Heat Transfer Coefficient (Watts/m2-K)

$$\overline{h} = \frac{1}{\theta_2 - \theta_1}\int_{\theta_1}^{\theta_2} h(\theta)\mathrm{d}\theta$$
Register 2: Average Near-Wall Gas Temperature (K)
$$\overline{\mathrm{T}}_g = \frac{1}{(\theta_2 - \theta_1)\overline{h}}\int_{\theta_1}^{\theta_2} h(\theta)\mathrm{T}_g(\theta)\mathrm{d}\theta$$
Register 3: Average Heat Flux (Watts/m2)
$$\overline{q''} = \frac{1}{(\theta_2 - \theta_1)}\int_{\theta_1}^{\theta_2} q''(\theta)\mathrm{d}\theta$$
Register 4: Average Wall Boundary Temperature (K)
$$\overline{\mathrm{T}}_w = \frac{1}{(\theta_2 - \theta_1)\overline{h}}\int_{\theta_1}^{\theta_2} h(\theta)\mathrm{T}_w(\theta)\mathrm{d}\theta$$
Register 5: Average y+
$$\overline{y^+} = \frac{1}{(\theta_2 - \theta_1)}\int_{\theta_1}^{\theta_2} y^+(\theta)\mathrm{d}\theta$$
Register 6: Average Distance from Boundary to y+=100 (m)
$$\overline{D} = \frac{1}{(\theta_2 - \theta_1)}\int_{\theta_1}^{\theta_2} D(\theta)\mathrm{d}\theta$$

- Data file 2:

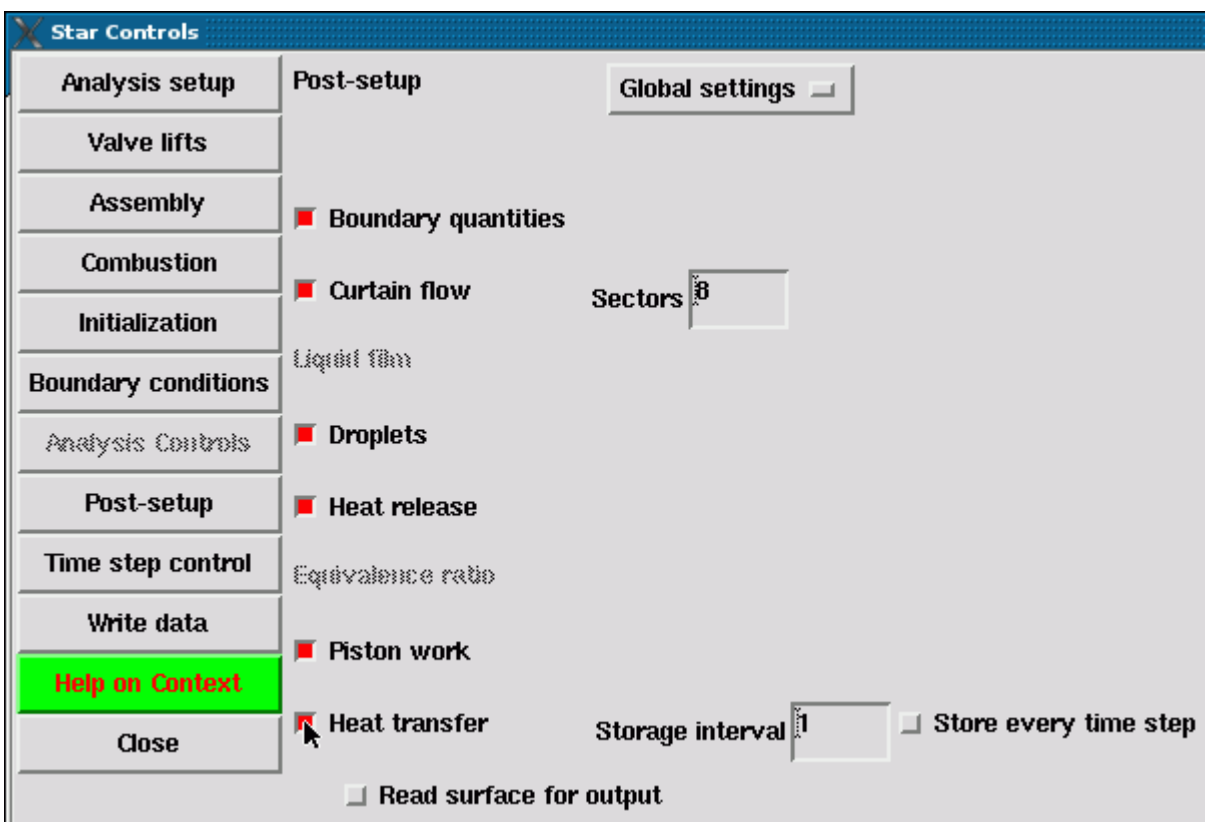
Register 1: Average Heat Transfer Coefficient at y+=100 (Watts/m2-K)
$$\overline{h_y} = \frac{1}{\theta_2 - \theta_1}\int_{\theta_1}^{\theta_2} h_y(\theta)\mathrm{d}\theta$$
Register 2: Average Near-Wall Gas Temperature at y+=100 (K)
$$\overline{\mathrm{T}}_y = \frac{1}{(\theta_2 - \theta_1)\overline{h}}\int_{\theta_1}^{\theta_2} h_y(\theta)\mathrm{T}_y(\theta)\mathrm{d}\theta$$

The nomenclature for the above equations is provided below:

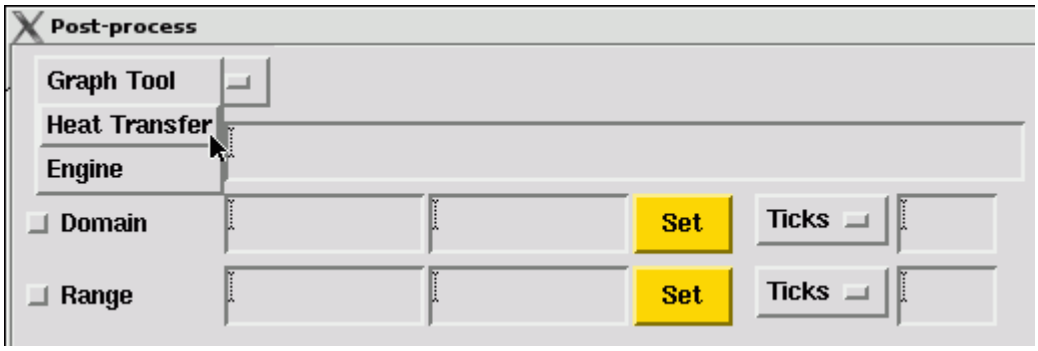
Notation	Description
θ	Crank Angle (degrees)
θ_1	Start Crank Angle (degrees)
θ_2	End Crank Angle (degrees)
h	Heat Transfer (W/m2-K)
h_y	Heat Transfer at y+=100 (W/m2-K)
T_g	Temperature of near-wall gas (K)
T_w	Temperature of wall boundary (K)
T_y	Temperature at y+=100 (K)
q''	Heat Flux (W/m2)
y^+	Non-dimensional wall co-ordinate (see Eqn. (410))
D	Distance from boundary to y+=100 (m)

The solution data is defined with respect to cell IDs, which must match exactly to the cell IDs in the `.dbs` file. To ensure the cell IDs correlate, follow the procedure below to generate this solution data.

1. Run an es-ice analysis with the **Heat Transfer** option selected in the *Star Controls* > *Post-processing* > *Global Settings* panel. This generates heat transfer data (`es-ice_htx.pos`) while STAR is obtaining a solution.

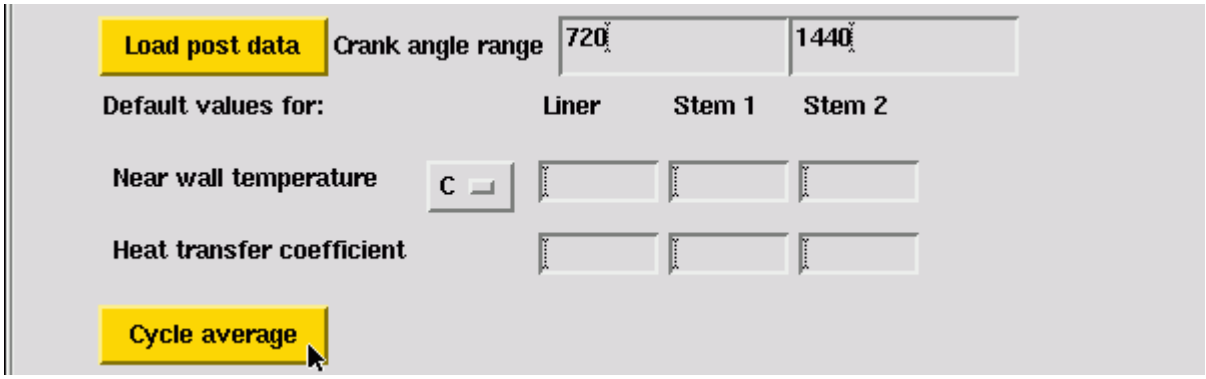


2. When the solver is complete, open the *Post-process* panel and select **Heat Transfer** from the drop-down menu at the top of the panel.

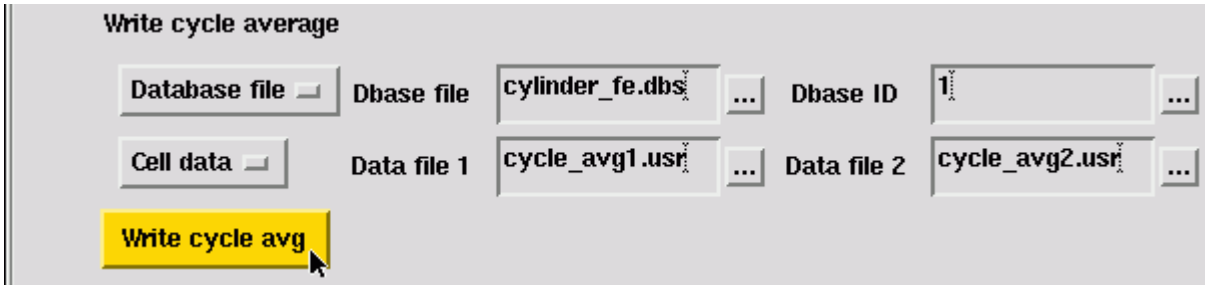


3. Click the ellipsis to select `es-ice_htx.pos` from the file browser and then click **Add after selected**. STAR has generated this file when running the analysis.

4. Click **Load post data** and enter an appropriate *Crank angle range* to cover the last full engine cycle. Then click **Cycle average** to calculate the cycle-averaged wall heat transfer data. The start crank angle (θ_1 in the equations above) is entered in the first text field, and the end crank angle (θ_2) in the second text field.



5. Specify the file names of the `.dbs` and two `.usr` files, set the database ID to 1 and click **Write cycle avg**. The CAE model (`.dbs`) and the solution data (`.usr`) is located in the current working directory.



Currently, STAR-CCM+ can only import coded (ASCII) format `.usr` files, whereas es-ice generates binary format files. Therefore, an extra step is required to convert the binary format file to a coded format file in pro-STAR.

1. Import the `.dbs` file into pro-STAR using the following commands:

```
dbase, open, <filename>.dbs
dbase, get, 1
```

   This ensures the cell IDs in the solution data match the CAE model.

2. Read the binary format `.usr` file into pro-STAR using the following commands:

```
getu, <filename>.usr, cell, all, binary
```

3. Save the solution data to a coded format `.usr` file with the following commands:

```
savu, <filename>.usr, all, coded, all
```

4. Repeat steps 2 and 3 for the second `.usr` file.

See Also:

Importing the es-ice or pro-STAR CAE Model

What Is the Imported CAE Model Node?

Preparing Wall Data in pro-STAR

pro-STAR can export the surface of a mesh into a `.dbs` file. This surface can be imported into STAR-CCM+ as either a surface mesh or a CAE model. pro-STAR can also export solution data (steady-state or transient) to a `.usr` file, which can be imported into STAR-CCM+ and mapped onto boundaries.

Only surface meshes from pro-STAR can be imported into STAR-CCM+ as a CAE model. Therefore, you can only export wall (surface) post-processing data from pro-STAR, as opposed to cell (volume) data.

Note: STAR-CCM+ can only import solution data from a coded (ASCII) format `.usr` file. Remember these limitations when preparing data in pro-STAR.

The solution data is defined with respect to cell IDs, which must match exactly to the cell IDs in the `.dbs` file. Otherwise, the solution data are imported incorrectly.

To ensure the cell IDs correlate, follow the procedure below to generate the CAE model and solution data:

1. Load the pro-STAR model and post-processing data. If you are running a transient analysis, you must also store the data for a given time-step.
2. Read the required solution data from the stored post-processing data. The command shown below reads the wall temperature data:

```
getw, t
```

3. Plot and export the surface mesh to a `.dbs` file using the following commands:

```
wplot
dbase, open, <filename>.dbs
dbase, put, <id>, star
```

4. Use the post-processing Registers to store up to six data sets. By default, scalar data is always read into Register 4 and the x, y, and z components of vector data are read into Registers 1, 2, and 3 respectively. The commands shown below swap the wall temperature data currently stored in Register 4 to Register 6, and reads the wall pressure data into the now empty Register 4:

```
operate, swap, 4, 6
getw, p
```

5. Save the solution data to a coded format `.usr` file using the following commands:

```
savu, <filename>.usr, all, coded, all
```

See Also:

Importing the es-ice or pro-STAR CAE Model

Importing the es-ice or pro-STAR Solution Data

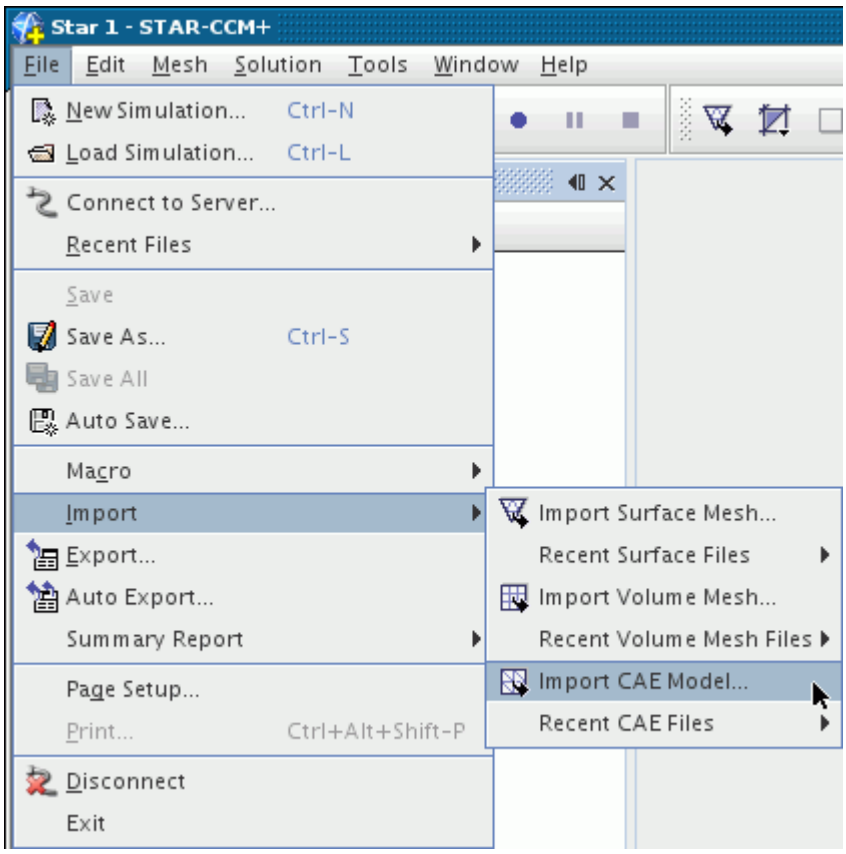
What Is the Imported CAE Model Node?

Importing the es-ice or pro-STAR CAE Model

The procedure for importing the CAE model from es-ice or pro-STAR is different from the procedure for importing models from other CAE products.

To import a CAE model from es-ice or pro-STAR:

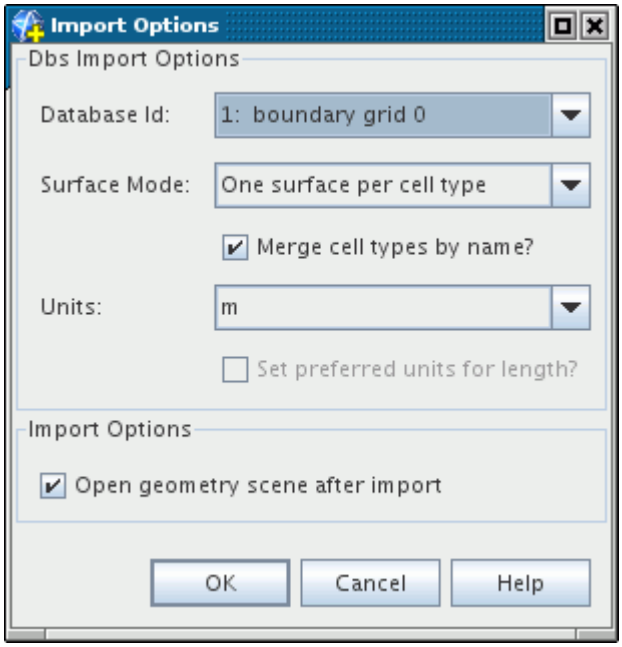
1. Select **File** > **Import** > **Import CAE model...**



   The *Open* dialog appears.

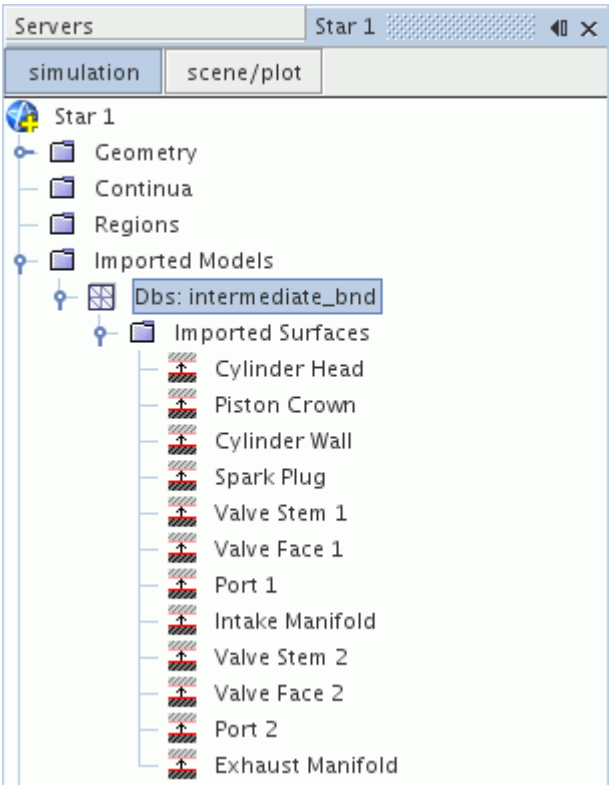
2. Select the `.dbs` file of a surface mesh (shells) from es-ice or pro-STAR.

   Selecting a file opens the *Import Options* dialog.



Note: There is no option to import a CAE model from es-ice or pro-STAR as a region. Also note that the *Units* must correspond to the length unit used to generate the model in es-ice or pro-STAR. For es-ice, is typically millimeters.

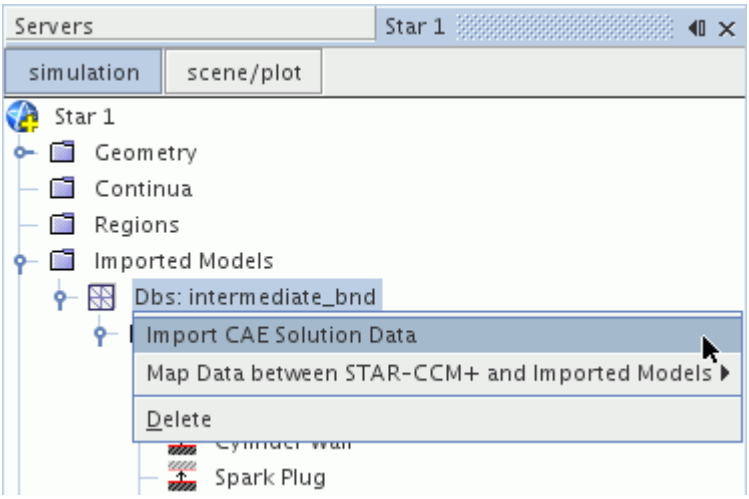
3. When you click **OK**, the CAE model appears in the simulation tree as a sub-node of the **Imported Models** node.



Importing the es-ice or pro-STAR Solution Data

To import solution data from es-ice or pro-STAR:

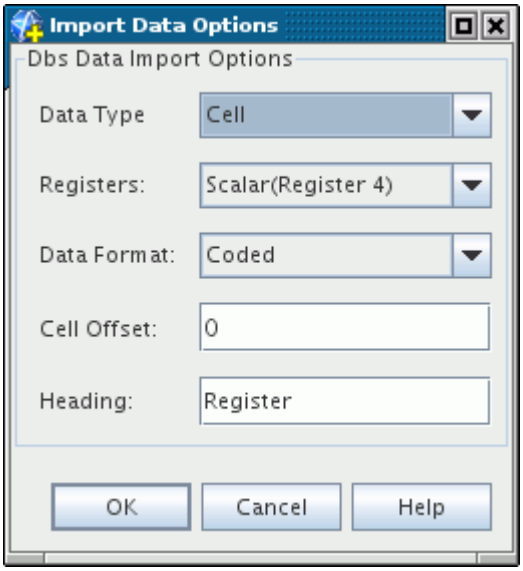
1. Right-click the CAE model node and select **Import CAE Solution Data**.



   The *Open* dialog appears.

2. Select the `.usr` file in a coded (ASCII) format.

   Selecting a file opens the *Import Data Options* dialog.



   The options available in the panel are described below:

   - *Data Type* – Currently, only cell data is available.
   - *Registers* – Selects the Registers that contain the required post-processing data. If you choose to import a Register that does not contain any data, the import fails. The following options are available:
     - **Scalar (Register 4)** – Selects the scalar data that is stored in Register 4.
     - **Vector (Registers 1-3)** – Selects the vector data that is stored in Registers 1-3.

  - **Both** – Selects both the vector and scalar data that is stored in Registers 1-3 and 4 respectively.
  - **All** – Selects all Registers. This is recommended when importing data from es-ice.
  - **Register n** – Selects Register n.
- **Data Format** – Currently, only coded format files are permitted.
- **Cell Offset** – to the cell ID numbers in the solution data. Typically this option is not required, especially you have followed the guidelines for preparing data in es-ice or pro-STAR.
- **Heading** – Label used to identify the Register in STAR-CCM+. For example, using the default heading (Register) stores the imported data in a field function called Imported Register n, where n is the Register number.

See Also:

Preparing Wall Data in pro-STAR

Preparing Wall Heat Transfer Data in es-ice

Mapping data from es-ice or pro-STAR to STAR-CCM+

Data can only be mapped from es-ice or pro-STAR to STAR-CCM+, not the opposite. This means that only one-way coupling is possible.

To map imported solution data from es-ice or pro-STAR to STAR-CCM+:

1. Right-click the CAE model node and select **Map Data** > **Surface Data...**

   This opens the *Map Imported Model Data* dialog.
2. In the *Map Direction* group box, select **Imported Model to** STAR-CCM+.
3. In the *Imported Surface* tree, tick the surfaces of the imported model, from which data is to be taken.
4. In the *Boundaries* tree, tick the equivalent boundaries to which the imported data is to be mapped.
5. In some cases, you might need to select additional options in the Nearest Neighbor Search Constraints group box. Use these options if your initial attempt at mapping fails.
6. Select the Register to be mapped from the **Data to map** drop-down menu.
7. Click **Apply** to map the imported solution data to the STAR-CCM+ mesh.

   This generates a field function with the same name as the imported data, prefixed by `Mapped` (e.g Mapped Imported Register 4). This field function can then be used in the same way as any other scalar field function.

See Also:

Nearest Neighbor Search Constraints

Working with Ensight

You can use the export capabilities of STAR-CCM+ to compare multiple solutions in one simulation. This approach involves saving the results of each run in a separate Ensight file and is only possible for boundaries and parts, not volumes.

Note: If results are to be imported directly into Ensight, boundary and region names must not exceed 80 characters.

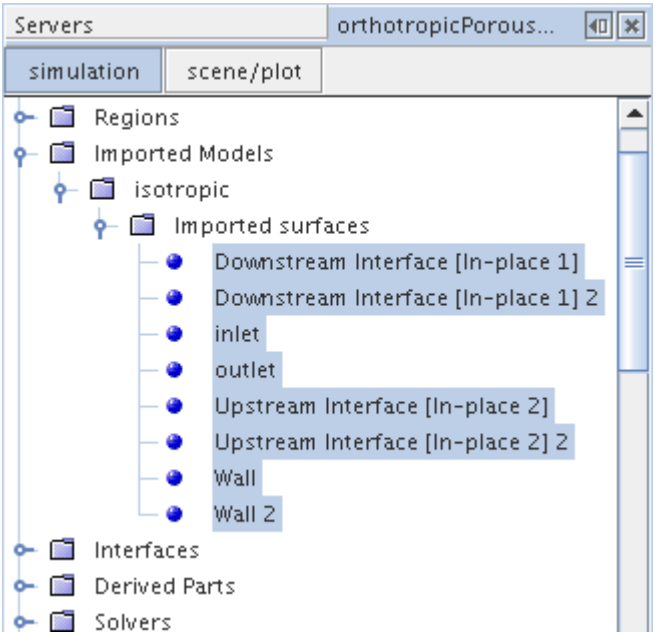
To compare results:

1. Complete the first run.
2. Export the results to an Ensight `.case` file (see Exporting Mesh and Solution Data).

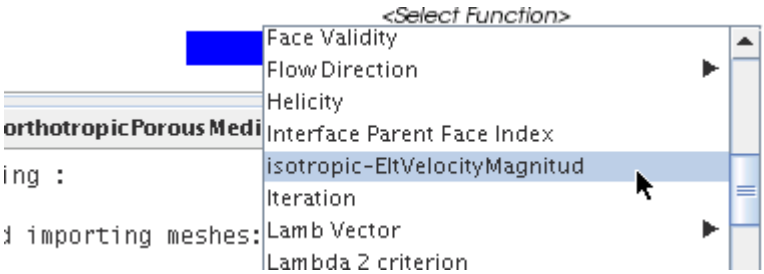
   When your export includes solution data, you have the following options:

   - To export vertex data, activate the **Export Results on Vertices** checkbox in the *Export File* dialog. This option is only available when your export includes solution data.
   - To export cell data, leave the **Export Results on Vertices** checkbox deactivated.
3. Clear the solution and make your modifications to the simulation (see Clearing The Solution).
4. Run the simulation again.
5. Import the data from the `.case` file (see Importing a CAE Model).

   The mesh surfaces appear in the object tree.



The scalars that you exported appear in the simulation as field functions, with the name of your Ensight file preceding the scalar name. You can select them just like any other scalar values, as shown in the following screenshot.



You can compare the results side-by-side, selecting the simulation's scalars in one scene and the imported ones in the other. For more information, see Viewing Two Scenes Side-by-Side.

Note that the minus operator ("−") can appear in the field functions generated by an import of an Ensight file. A temperature function imported from a file called `test.case` would have the name `test-Temperature`. In order for such a function to be referenced by another field function, the following syntax would have to be used:

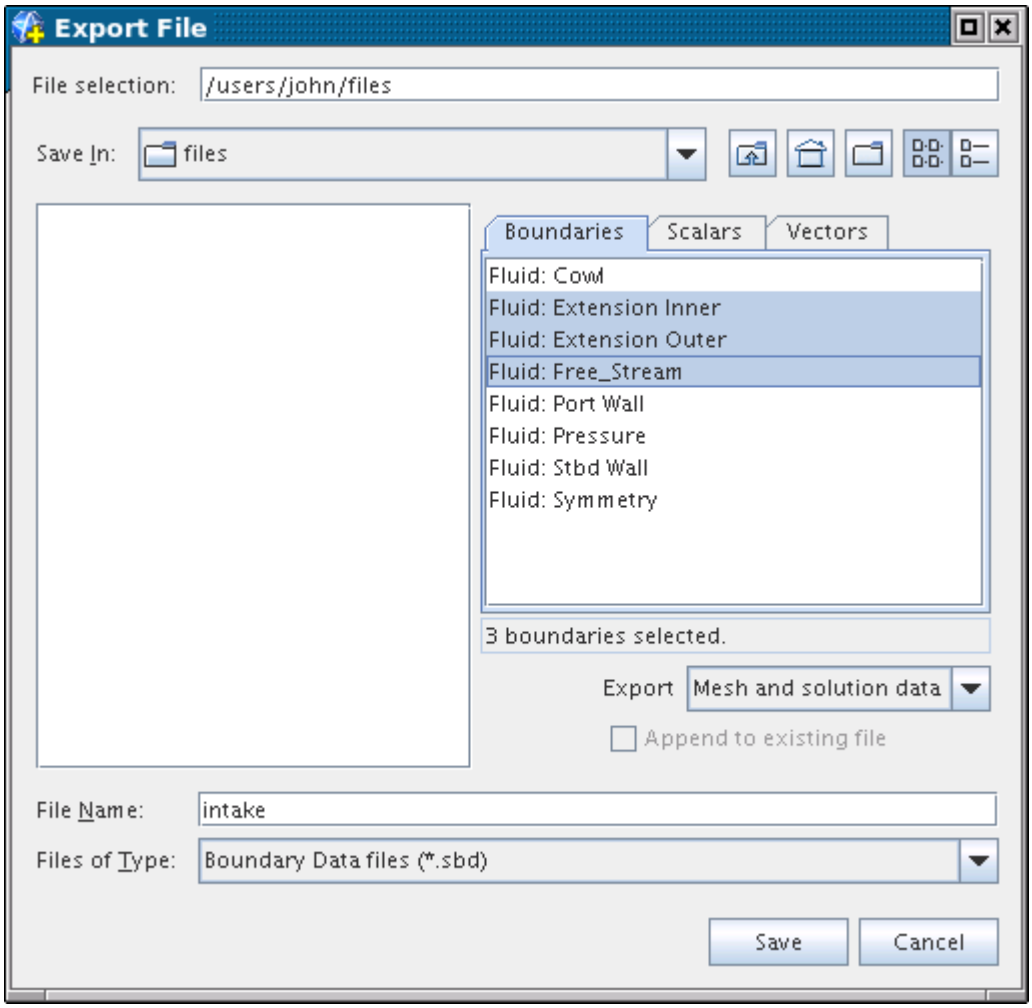
```
${test-Temperature}
```

Working with the STAR-CCM+ Boundary Data (.sbd) Format

The STAR-CCM+ Boundary Data (`.sbd`) file format is an internal STAR-CCM+ format that allows you to exchange boundaries across different simulations. For boundaries that you choose, this binary file contains mesh information as well as selected scalar or vector solution data.

The advantages of this format for exchanging boundaries include:

- It enables coupling of simulations with different physics.
- Each of the analyses can be performed separately and the coupling can take place at desired intervals.
- It is especially suitable when each of the analyses requires significant computer resources.
- You can isolate boundaries alone for coupling, rather than importing a whole region. The emphasis on boundaries is shown in the *Export File* dialog, which replaces the *Regions* tab with one for boundaries when you select the `.sbd` format in the *Files of Type* drop-down list.



An example of using this file format is a coupled conjugate heat transfer (CHT) analysis to optimize the design of an automobile. This optimization involves exchanging the heat transfer coefficient and reference temperature with vehicle thermal analysis and retrieving surface temperatures. Given the distinct nature of

physics to be captured in the under-hood fluid analysis and in the vehicle thermal analysis, the models created are drastically different. Differences include the mesh densities and topologies, the grouping of parts and boundary definitions, and the preservation of geometric features such as sharp corners and edges. However, the `.sbd` file makes this possible with the following procedure:

1. Choosing the boundaries representing the under-hood surface from the fluid analysis and selecting the local heat transfer coefficient and reference temperature for export.
2. Exporting the chosen boundaries and the selected fields to an external `.sbd` file. See Exporting Mesh and Solution Data.
3. Importing the `.sbd` file into the solid analysis, and mapping the imported surface data to the corresponding under-hood surface in the vehicle thermal model. For more information, see Importing a CAE Model and Mapping CAE Solution Data to a STAR-CCM+ Model.
4. Performing thermal analysis.
5. Choosing the boundaries comprising the solid under-hood surface and selecting the surface temperature for export. See Exporting Mapped Data.
6. Exporting the chosen boundaries and the selected fields to an external `.sbd` file.
7. Importing the `.sbd` file into the fluid analysis and mapping data onto the surface.

Contents:

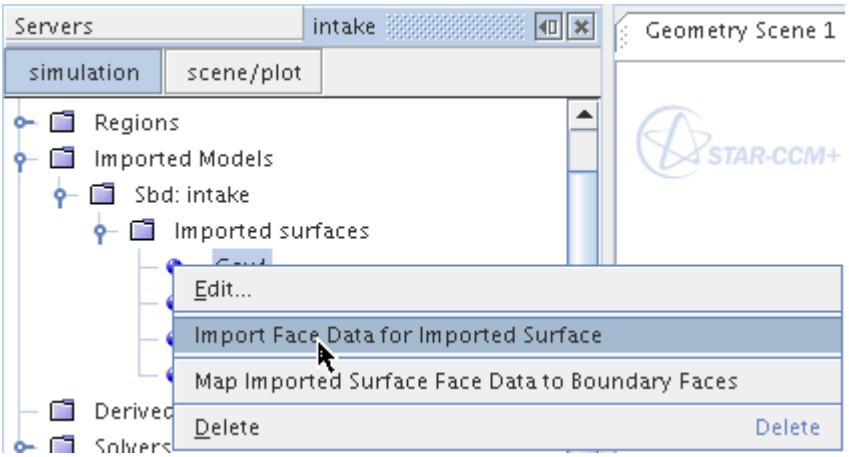
Importing and Mapping the .sbd Format
Working with Field Functions
Examples of Post-Processing

Importing and Mapping the .sbd Format

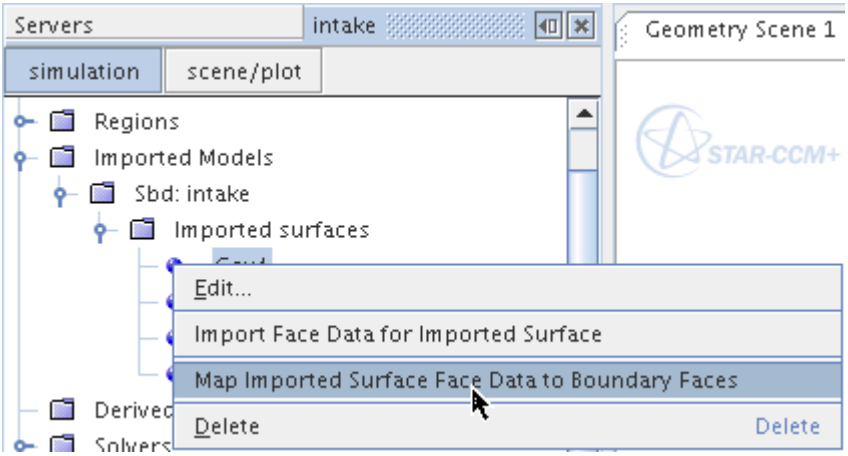
The sequence of operations is similar to those for coupling with external CAE packages such as Abaqus or TAITherm. The only difference is that the imported data is node-centric in case of the finite element packages and face-centric in case of the `.sbd` format.

The pop-up menus of the imported surface objects indicate this, as they refer to faces rather than vertices:

1. **Import Face Data for Imported Surface**



2. **Map Imported Surface Face Data to Boundary Faces**



See Also:

What Is the Imported Surfaces Node?

Importing a CAE Model: Workflow

Importing CAE Solution Data

Mapping Imported Surface Data to STAR-CCM+ Boundaries

Working with Field Functions

Field functions work with the `.sbd` format as follows:

- Imported Field Functions -- When an `.sbd` file is imported, field functions are automatically created for all the imported data fields. These field functions have storage only on the imported surfaces.
- Mapped Field Functions -- When data is mapped between the imported surfaces and STAR-CCM+ boundaries, a new set of field functions are automatically generated. These field functions have storage only on the target surfaces. Apart from this mapped field, a field function named `Mapped Flag` is created for all mappings. This field has a value of 0, 1 or 2. This field function also has storage only on the target surface. On any face, a value of 0 indicates that mapping was successful, a value of 1 indicates that the mapping failed the proximity check and a value of 2 indicates failure of the normalcy check. The proximity and normalcy checks are additional mapping options.

See Also:

Importing CAE Solution Data

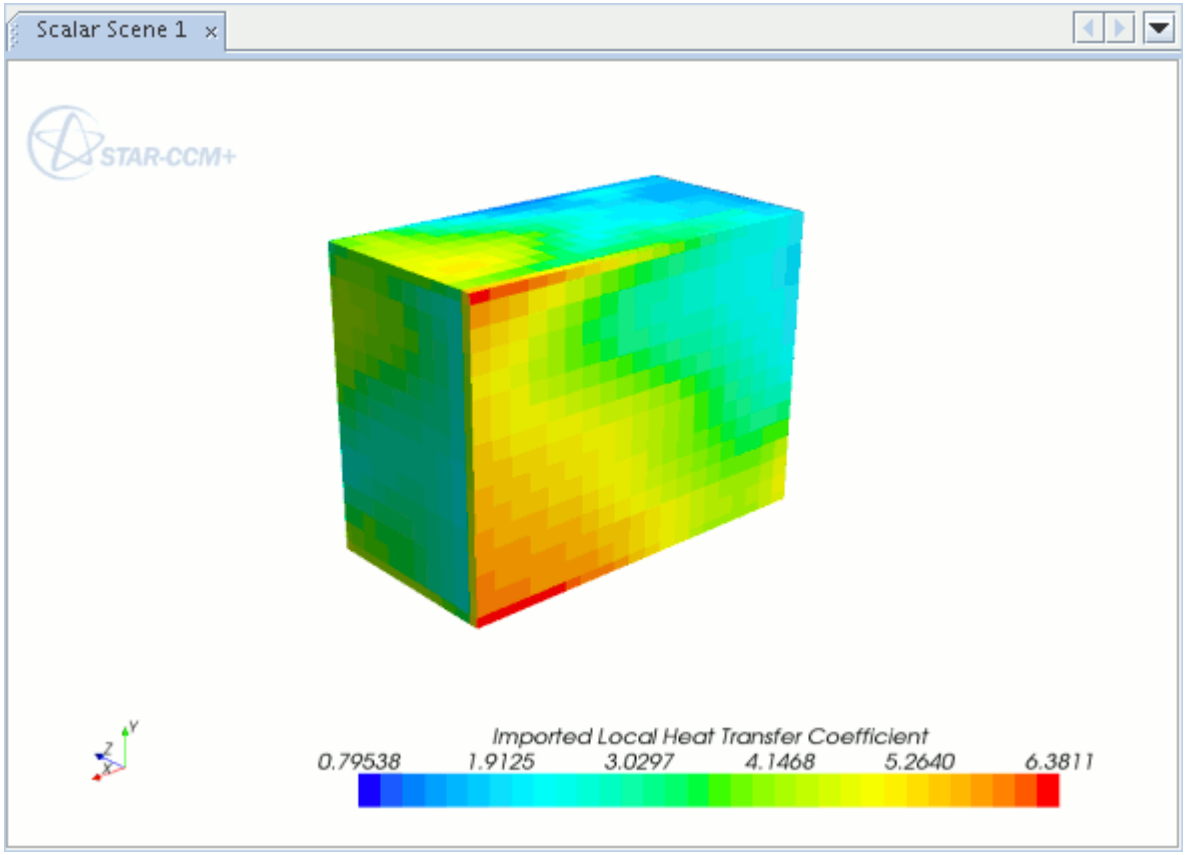
Mapping Imported Surface Data to STAR-CCM+ Boundaries

Nearest Neighbor Search Constraints

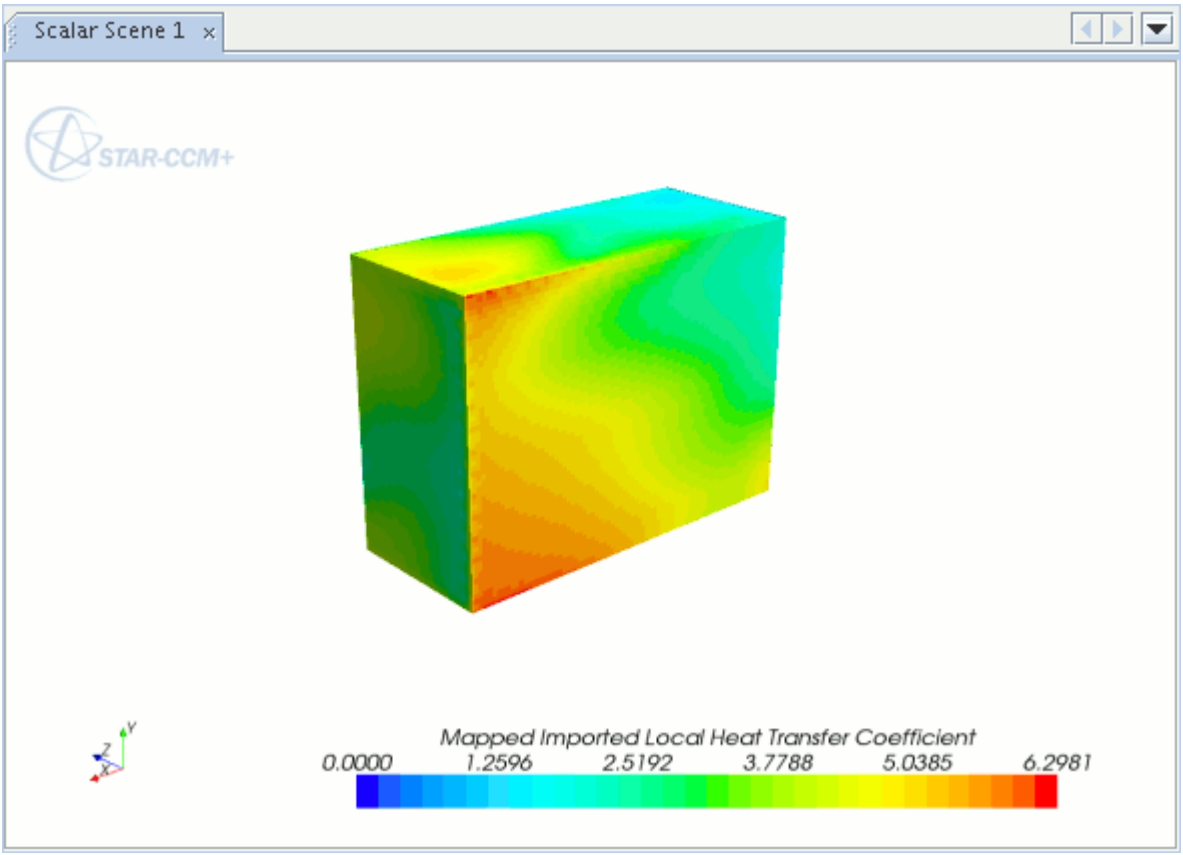
Examples of Post-Processing

The following screenshots show how the imported quantities are displayed in STAR-CCM+.

- Imported local heat transfer coefficient



- Mapped imported local heat transfer coefficient


Scalar Scene 1
STAR-CCM+
Mapped Imported Local Heat Transfer Coefficient
0.0000
1.2596
2.5192
3.7788
5.0385
6.2981

Working with the Visualization Toolkit (VTK) Format

The VTK format is part of a system of visualization and computer graphics programming known as the Visualization Toolkit. The `.vtk` format itself is designed to present different types of data consistently. Two styles of this file format exist: legacy (serial) and XML-based. STAR-CCM+ works with the first style, either binary or ASCII, to import surface data. VTK files are also used by the arbitrary probe derived part.

STAR-CCM+ only supports POLYDATA datasets. For more information, see Visualization Toolkit.

Creating a VTK File

It is possible to create a VTK file by exporting data from a derived part. See Exporting to an External File.

For creating a file manually to import into STAR-CCM+, the following example shows how to input 7 data points.

```
# vtk DataFile Version 2.0
Example Data for STAR-CCM+ Arbitrary Probe
ASCII
DATASET POLYDATA
POINTS 7 float
0.0 0.0 0.0
1.0 0.0 0.0
2.0 0.0 0.0
0.0 1.0 0.0
1.0 1.0 0.0
2.0 1.0 0.0
0.0 0.0 1.0
```

The sensor and/or probe locations must lie inside of the geometry in order for a successful import to take place.

See Also:

Importing a CAE Model: Workflow

Coupling with RCAS

RCAS (Rotorcraft Comprehensive Analysis System) is a Computational Structural Dynamics (CSD) tool for modeling rotor blades. STAR-CCM+ allows you to import RCAS models for fluid-structure interaction analyses using file-based coupling. The RCAS model mesh and motion data are stored within the same file, with the standard CSD `*.in` format.

Contents:

Importing the RCAS Beam Model
Importing Displacement and Rotation Data
Mapping Field Data
Exporting to RCAS

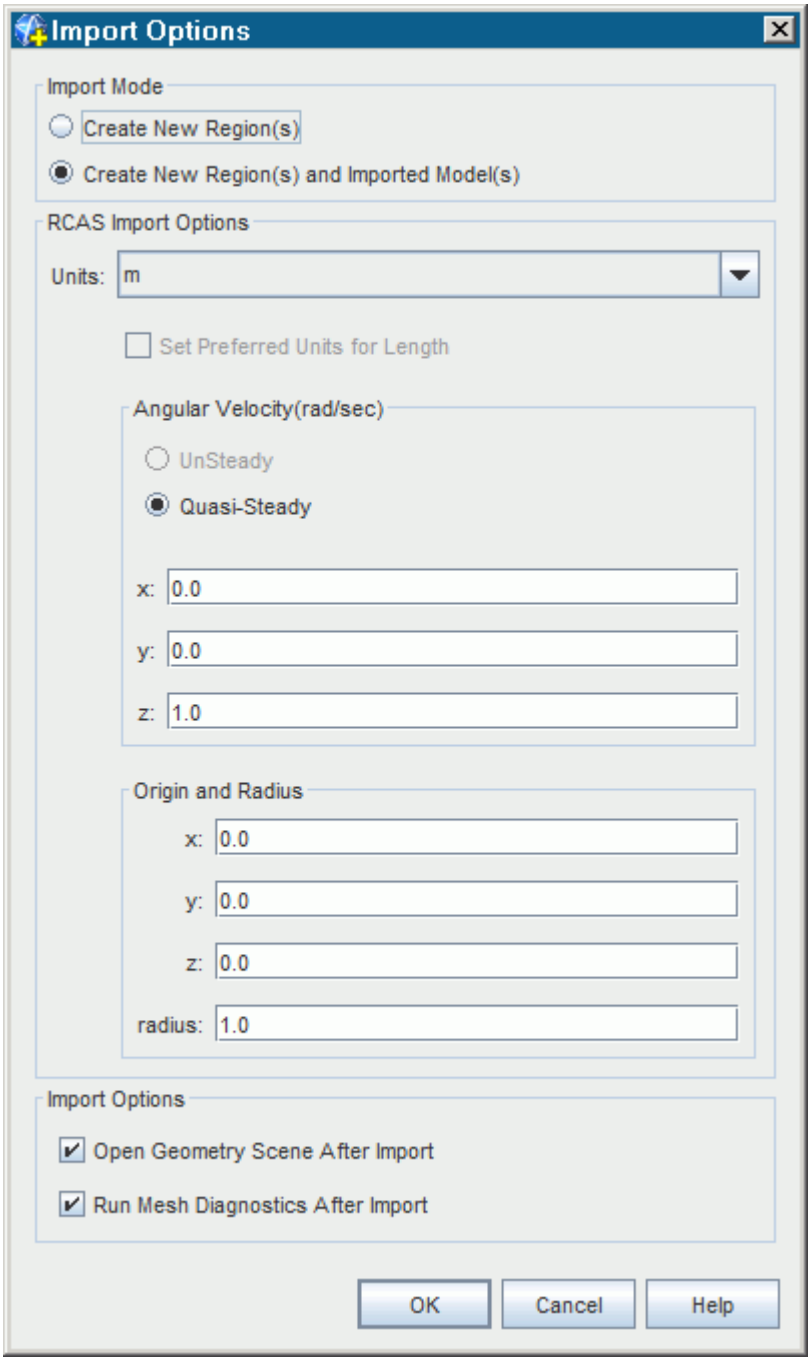
Importing the RCAS Beam Model

Rotor blades are modeled in RCAS as 6-DOF beams and can be imported into STAR-CCM+ as beam regions.

For more information on beam elements in STAR-CCM+, see Imported Beam Elements.

To import the rotor blade mesh in a STAR-CCM+ simulation:

1. Select **File** > **Import** > **Import CAE Model...**
2. Select the RCAS `*.in` file that contains the model mesh and motion data.
3. In the *Import Options* dialog:
   a) Set *Import Mode* to **Create new Region(s) and Imported Model(s)**.
   b) Specify the *RCAS Import Options*, that is, the rotor *Angular Velocity*, *Origin*, and *Radius*.
   These data items are stored within the imported beam model node, for later use.



c) Click **OK**.

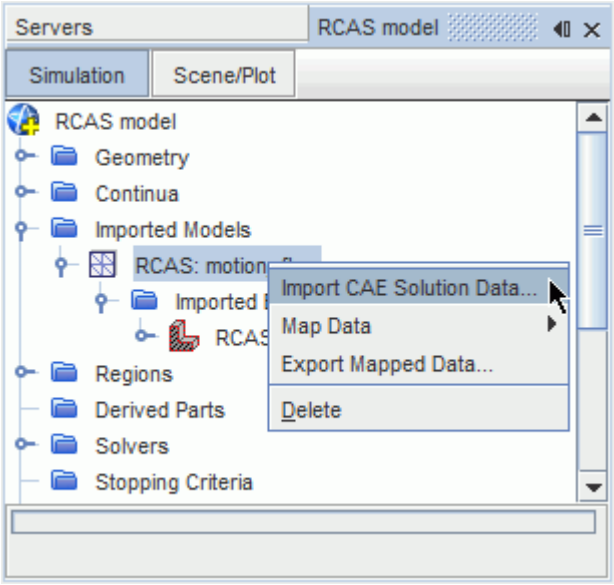
Importing Displacement and Rotation Data

In STAR-CCM+, you can import the RCAS displacement and rotation data from the RCAS **Imported Model** node. The imported RCAS displacement and rotation data become available in STAR-CCM+ as the **Imported Displacement** and **Imported Rotation** field functions.

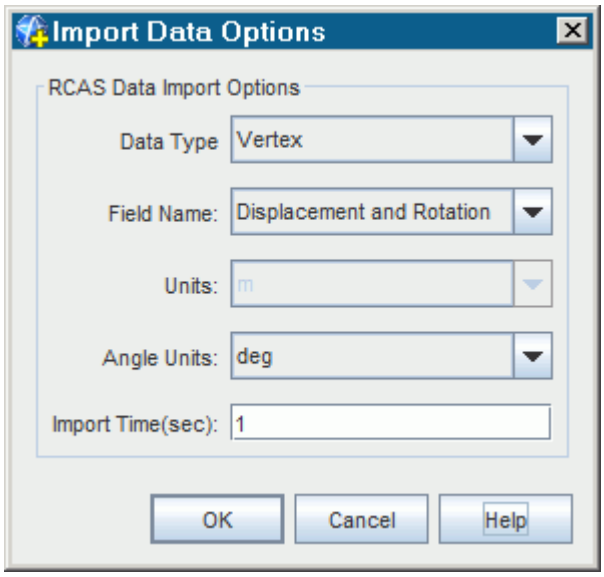
The RCAS mesh and motion data are stored in the same `*.in` file.

To import the RCAS motion data:

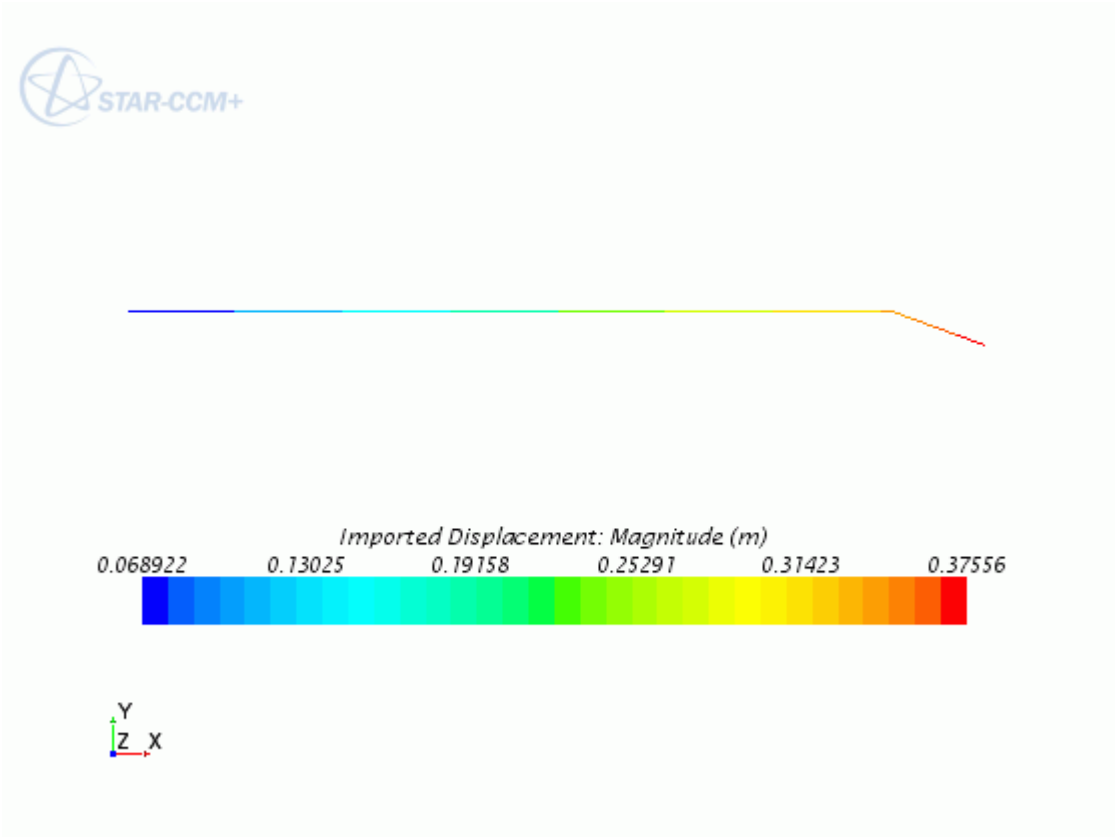
1. Right-click the imported RCAS model node and select **Import CAE Solution Data...**



2. In the *Open* dialog, select the RCAS model `*.in` file that contains the mesh and motion data.
3. In the *Import Data Options* dialog, specify the *Angle Units* and *Import Time*, then click **OK**.



You can view the imported displacement and rotation data in a scene by selecting the corresponding field functions.



Mapping Field Data

You can map pressure and shear stress data, defined on the faces of the wetted surface in STAR-CCM+, onto the nodal forces and moments for each node on the RCAS beam model. Similarly, you can map the nodal displacement and rotation data, defined on the RCAS beam model, onto the wetted surface in STAR-CCM+.

Instructions for performing mapping operations from/to beam models are given in the sections, Mapping Boundary Data to Imported Beams and Mapping Imported Beam Data to STAR-CCM+ Boundaries.

Exporting to RCAS

Data that has been mapped onto an imported RCAS model can be exported from STAR-CCM+ to external `*.out` files.

To export data from STAR-CCM+ to the RCAS model:

1. Import the RCAS beam model into STAR-CCM+. For more information, see Importing the RCAS Beam Model.
2. Map the data that you want to export to RCAS onto the imported RCAS model, as described in the previous section.
3. Export the mapped data to RCAS following the instructions outlined in the section, Exporting Mapped Data. Choose `.in` as file extension.

As the export of CFD loads is transient, the export file is appended for discrete values of the azimuth angle (rotor plane) for each revolution.

STAR-CCM+ to STAR-CCM+ Co-Simulation

STAR-CCM+ allows you to couple two running simulations together in a co-simulation analysis. The STAR-CCM+ to STAR-CCM+ co-simulation capability targets analyses involving conjugate heat transfer (CHT), fluid-structure interaction (FSI), and exchange of any scalar or vector fields across the coupled boundaries or regions.

The co-simulation is between 3D geometries that have one or more shared interfaces but different mesh densities and time scales. Thermal and mechanical fields can be exchanged across the fluid-solid interface; auxiliary fields can be exchanged across fluid-solid, fluid-fluid, or solid-solid interfaces. One instance of STAR-CCM+ solves the solid domain and the other solves the fluid domain. This technique is similar to the technique described in the section, Explicit Coupling within STAR-CCM+, but without the requirement to manually create data mappers in each simulation or to create any Java macros that automate the exchange between two instances of STAR-CCM+.

Conjugate heat transfer simulations are commonly used to model both solid and fluid domains in a single instance of STAR-CCM+. In these simulations, however, it is not possible to account for the different time scales present in the solid and fluid domains. For scenarios where the time scales of the thermal conduction in the solid are much longer than the time scales in the fluid, STAR-CCM+ to STAR-CCM+ co-simulation offers a particular advantage. For example, the exhaust gases escape the tail pipe almost immediately after the engine is started, but the engine itself may take minutes to reach its maximum temperature.

For simulations with widely varying time scales, STAR-CCM+ to STAR-CCM+ co-simulation allows you to treat the fluid and solid domains separately. The fluid is treated in a steady state simulation and the solid in a transient simulation. The solid and fluid simulations must exchange data at a frequency that captures the slow change in the solid temperature.

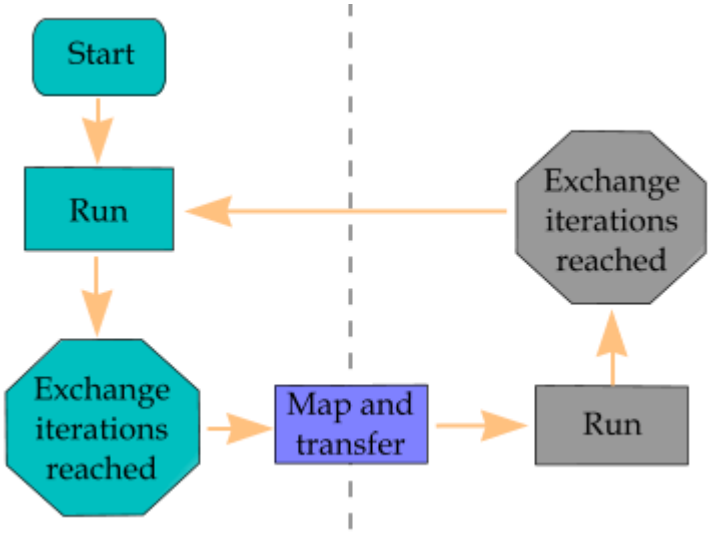
A solid thermal simulation typically requires fewer iterations per time-step to reach convergence than is required by the fluid simulation. By allocating a separate simulation to the solid thermal analysis, the compute time that is required to solve the overall solid-fluid analysis can be considerably smaller when the co-simulation technique is used. And although the computer CPU time needed to reach the final time in a STAR-CCM+ to STAR-CCM+ CHT co-simulation over a single STAR-CCM+ CHT simulation can be considerably smaller, with a properly configured co-simulation the same level of accuracy is achieved as in a fully-implicit conjugate simulation.

This section describes the types of co-simulation and how to use STAR-CCM+ to STAR-CCM+ co-simulation.

One-Way Co-Simulation

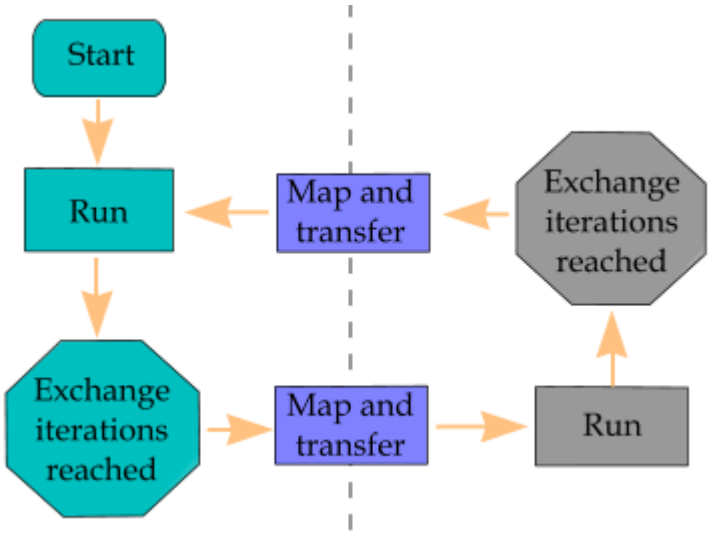
The diagram below demonstrates a one-way STAR-CCM+ to STAR-CCM+ co-simulation process. In the diagram, the leading partner provides data to the lagging partner but does not receive any data from it. You can invert the relationship so the lagging partner provides information to the leading partner but not receive information from it. After the leading simulation has run for a specified number of iterations, a mapping operation occurs and then the data in the shared zone is transferred to the waiting lagging simulation. The

lagging simulation uses the data as part of its solution and runs. However, the leading simulation does not use the data in the shared zone before recommencing. The process of transferring data to the lagging partner simulation, running, and waiting continues until both solutions have satisfied their stopping criteria. The setup process is similar to two-way coupling, except deactivate thermal or mechanical import.



Two-Way Co-Simulation

The diagram below demonstrates the two-way STAR-CCM+ to STAR-CCM+ co-simulation process. After the leading simulation has run for a specified number of iterations, a mapping operation occurs and then the data in the shared zone is transferred to the waiting lagging simulation. The lagging simulation uses the data as part of its solution and runs. The process of transferring data to the partner simulation and waiting continues until both solutions have satisfied their stopping criteria.



Contents:

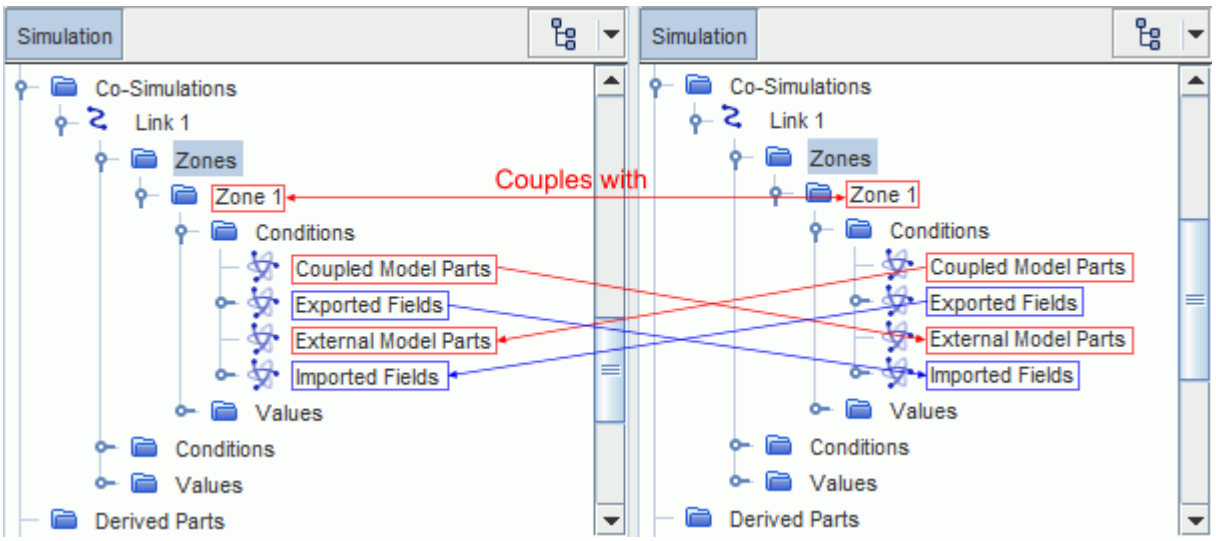
Co-Simulation Interface and Exchanged Fields
STAR-CCM+ Co-Simulation Workflow
STAR-CCM+ Co-Simulation Models Reference
Partitioned Coupling Model Reference
Uncoupled Mesh Entities Treatment Model Reference

Co-Simulation Interface and Exchanged Fields

STAR-CCM+ to STAR-CCM+ co-simulation allows for data exchange across the coupled boundaries, or regions, of two running STAR-CCM+ simulations.

Co-Simulation Interface

In a STAR-CCM+ to STAR-CCM+ co-simulation, the partner simulations exchange fields across a shared interface that you specify by defining co-simulation zones in each partner simulation. A co-simulation zone specifies a set of model parts (boundaries or regions) that couple with the model parts that are specified within the corresponding co-simulation zone in the partner simulation. A co-simulation zone also specifies which fields are exchanged between the coupled model parts.



You can define multiple co-simulation zones that specify different conditions on different sets of coupled model parts. Each zone is identified with an index and connects to the zone with the same index that is defined in the partner simulation. The number of co-simulation zones defined in a simulation must match the number of zones in the partner simulation.

Exchanged Fields

STAR-CCM+ allows you to specify the fields that are exchanged between the simulations using either predefined fields, or auxiliary fields.

- When using predefined fields, the simulations can exchange the following fields at the coupled boundaries:
  - Predefined **Thermal Fields**: one simulation exports **Temperature** and the other exports **Convective Flux** (heat transfer coefficient + reference temperature).
  - Predefined **Mechanical Fields**: one simulation exports **Pressure** or **Traction** (pressure + wall shear stress), and the other exports **Displacement**.

You can set up a co-simulation analysis using either a managed or an unmanaged approach. In the managed approach, STAR-CCM+ automatically manages the boundary conditions at the coupled boundaries, based on the fields that are imported from the partner simulation. In the unmanaged approach, you set the boundary conditions manually. You can only exchange predefined fields between boundaries (surface-to-surface coupling).

- When using auxiliary fields, the simulations can exchange any field functions defined on the coupled model parts (either boundaries or regions). You set boundary or region conditions manually, as STAR-CCM+ does not automatically apply the imported fields on the coupled parts. You can exchange auxiliary fields between coupled boundaries (surface-to-surface coupling) or coupled regions (volume-to-volume coupling).

You can use predefined thermal and mechanical fields, and auxiliary fields, together in the same simulation.

STAR-CCM+ Co-Simulation Workflow

This section outlines the recommended workflow for coupling two STAR-CCM+ simulations in a co-simulation analysis.

Contents:

Preparing the Simulations for Co-Simulation
Specifying the Coupled Model Parts
Specifying the Exchanged Fields
Partitioned Coupling for FSI Applications
Specifying the Data Exchange Frequency
Specifying Boundary Conditions on Uncoupled Faces
Running the Co-Simulation

Preparing the Simulations for Co-Simulation

Modify standalone STAR-CCM+ simulations for interactive co-simulation.

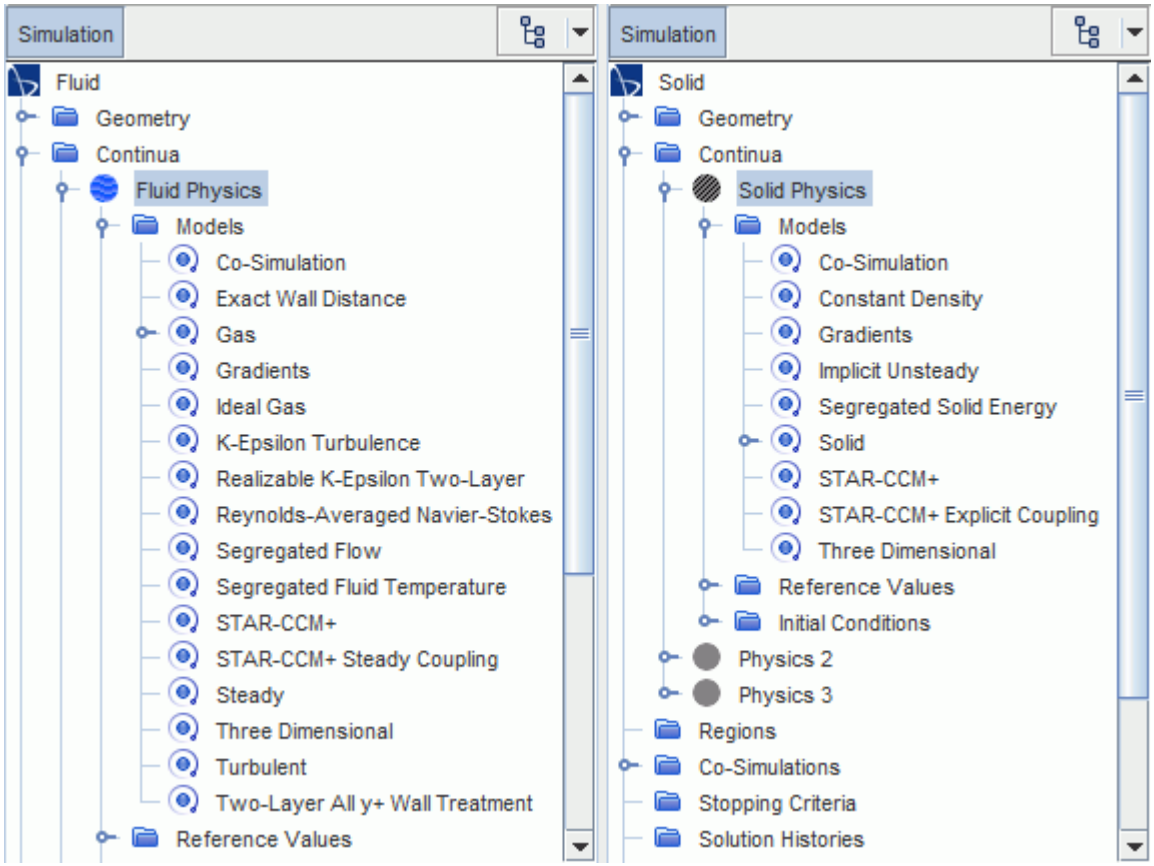
1. Develop standalone STAR-CCM+ simulations, without any co-simulation settings. In each simulation, activate physics models as required by your analysis:
   - For conjugate heat transfer analysis, activate an energy model in both simulations.
   - For FSI applications that model the displacement of a solid in response to fluid loads:
     - In the simulation representing the solid, activate the **Finite Volume Solid Stress** model. Currently, the FE **Solid Stress** model is not compatible.
     - In the simulation representing the fluid, assign a morphing motion to the fluid region, so that the fluid mesh updates in response to the solid displacements.
2. In each simulation, use provisional boundary conditions that are close to the expected values from the partner simulation.
3. Run the standalone simulations to convergence before attempting co-simulation.

   If either standalone model is improperly defined or has numerical problems, you are unlikely to obtain a solution for a co-simulation.
4. In each simulation, activate additional physics models to enable co-simulation:

Group	Physics Model
Optional Models	**Co-Simulation**
Co-Simulation Models	**STAR-CCM+**
STAR-CCM+ Coupling Models	• In steady simulations, STAR-CCM+ automatically activates the **STAR-CCM+ Steady Coupling** model, which allows the steady simulation to exchange data at a specified frequency that you specify in terms of the of solver iterations.

Group	Physics Model
	• In transient (unsteady) simulations, activate either one of the following models: ◦ **STAR-CCM+ Explicit Coupling**— allows the simulation to exchange data once per time-step, at a specified coupling time interval. You can either instruct STAR-CCM+ to automatically adjust the simulation time-step to meet the coupling time, or you can preserve the specified time-step and allow STAR-CCM+ to adjust the coupling time. See Time-Step Adjustment. ◦ **STAR-CCM+ Implicit Coupling**—allows the simulation to exchange data more than once per time-step, at a frequency that you specify in terms of the solver inner iterations. In general, for CHT and FSI problems, the **STAR-CCM+ Implicit Coupling** model allows you to reach a stable solution with larger time-steps than the time-steps required by the **STAR-CCM+ Explicit Coupling** model. Using the **STAR-CCM+ Implicit Coupling** model is also recommended for FSI problems that require multiple data exchanges per time-step to reach a stable solution. **Note:** You can couple two steady simulations, two transient simulations, or a steady simulation and a transient simulation. In co-simulation between two transient simulations, activate the same STAR-CCM+ coupling model in both simulations. Coupling between a simulation that uses the explicit scheme and a simulation that uses the implicit scheme is not supported.
Optional Models	For FSI applications, activate the **Partitioned Coupling** model. This model allows you to control the convergence of the coupled problem.

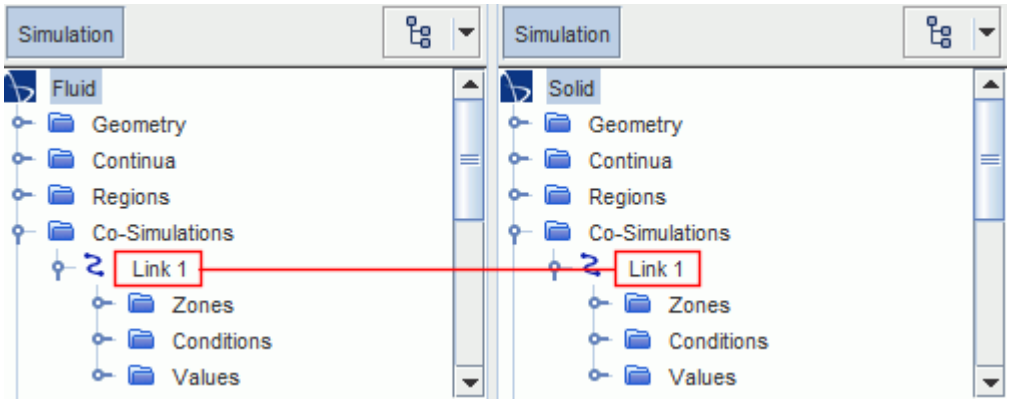
The following image shows an example of model selection for two simulations that are coupled in a STAR-CCM+ to STAR-CCM+ thermal co-simulation.



When you activate a STAR-CCM+ co-simulation model, STAR-CCM+ adds additional physics continua under the **Continua** node that represent the partner simulation. Since the co-simulation framework allows for both surface-to-surface and volume-to-volume coupling, STAR-CCM+ adds two physics continua, one for each type of coupling. STAR-CCM+ automatically uses the relevant continuum.

STAR-CCM+ also adds a co-simulation link to the simulation tree, **Co-Simulations** > **[Link 1]**, which allows you to define co-simulation zones and to specify general conditions and values.

- You can rename co-simulation links as desired. However, the name of a link must match the name of the corresponding link in the partner simulation.



Specify a leading and a lagging simulation. When you run the co-simulation, the leading simulation runs first, whereas the lagging simulation waits for data from the leading simulation:

5. In one of the simulations, select the **[Link 1]** > **Conditions** > **Concurrency Option** node and set *Method* to **Lead**.

   For more information, see [Link 1] Conditions and Values.
6. Using the same procedure, set the other simulation to **Lag**.
7. In each simulation, set the **[Link 1]** > **Conditions** > **Mapping Option** to specify how the data mapping occurs. For more information, see [Link 1] Conditions and Values.

To run the co-simulation, you can click **Run** in either simulation. Clicking **Run** in either simulation does not determine which simulation runs first. The leading simulation, as specified within the **Concurrency Option**, always runs first. The simulation where you click **Run** handles the connection to the partner simulation and, in the case of inconsistent settings, enforces the correct settings in the partner simulation.

Optionally, you can set the simulation where you click **Run** to launch the partner simulation automatically on co-simulation run.

8. In the simulation where you click **Run**, select the **Conditions** > **Launch Partner Option** node and specify the method for launching the partner simulation:

Option	Step
Launch the partner simulation manually	Set **Launch Partner Option** to **Do not Launch**
Launch the partner simulation automatically when the co-simulation starts	Set **Launch Partner Option** to **Launch Default Application**

   STAR-CCM+ automatically specifies consistent settings in the partner simulation.

   For more information, see STAR-CCM+ Link Reference.
9. Specify the method for connecting to the partner simulation:
   - If the simulations are located in the same directory and you launch the simulations separately, you do not need to specify host and port details. In each simulation, select the **Conditions** > **Connect Option** node and set *Option* to **Connection Settings via File**. STAR-CCM+ saves the host and port details to connection files, which are automatically used for connection when you click **Run** in either simulation. For more information, see Connection Settings via File.
   - If you launch the simulations manually, but they are not located in the same directory, specify the host and port of the partner simulation in the simulation where you click **Run**:
     1. Select the **Conditions** > **Connect Option** node and set *Option* to **Host:Port**.
     2. Select the **Values** > **STAR-CCM+ Partner Host:Port**. In the *Properties* window, click ... (**Custom Editor**) and select the partner simulation from the list of all currently active STAR-CCM+ simulations.
   - If you instruct one simulation to launch the partner simulation automatically, the simulation connects to the partner using the launching details. The **Values** > **Assigned Partner Host:Port** node displays the host and port where the partner simulation is running.

Specifying the Coupled Model Parts

In each simulation, create co-simulation zones to couple boundaries or regions across simulations.

By default, the **[Link 1]** > **Zones** node contains one co-simulation zone. If you wish to apply different field import/export settings on different sets of coupled model parts, you require multiple co-simulation zones.

To create additional zones:

1. Right-click the **Zones** node and select **New**.

   Each zone connects to the zone with the same index that is defined in the partner simulation.

   Note: The number of co-simulation zones defined in a simulation must match the number of zones in the partner simulation.

For each zone, specify the zone type and the coupled model parts (boundaries or regions). Make sure that the settings that you specify for a co-simulation zone are consistent with the settings that you specify for the corresponding zone in the partner simulation:

2. Select the **Zones** > **[Zone]** node and specify the *Zone Type*:
   - To exchange data between coupled boundaries (surface-to-surface coupling), set *Zone Type* to **Surface Zone** in each simulation.
   - To exchange data between coupled regions (volume-to-volume coupling), set *Zone Type* to **Volume Zone** in each simulation. This option requires you to specify the imported and exported data using auxiliary fields. See Exchanged Fields.

Assign one or more model parts to the co-simulation zone. The coupled model parts selected for a zone couple to the parts selected for the zone of same index in the partner simulation (see Co-Simulation Interface):

3. To couple one or more boundaries to corresponding boundaries in the partner simulation:
   a) Expand the **Regions** > **[Region]** > **Boundaries** > **[Boundary]** nodes of interest.
   b) Select the **Physics Conditions** > **External Code Coupling Specification** node and use the *External Code Coupling Specification* property to assign the boundary to a co-simulation zone of type **Surface Zone**.
   c) Repeat these steps in the partner simulation.
4. When you specify the imported and exported data using auxiliary fields, you can also couple one or more regions with corresponding regions in the partner simulation:
   a) Expand the **Regions** > **[Region]** nodes of interest.
   b) Select the **Physics Conditions** > **External Code Coupling Specification** node and use the *External Code Coupling Specification* property to assign the region to a co-simulation zone of type **Volume Zone**.
   c) Repeat these steps in the partner simulation.

   When you assign model parts to a zone, the **[Zone]** > **Conditions** > **Coupled Model Parts** node shows the specified coupled model parts. When you start the co-simulation, the **External Model Parts** condition is updated automatically based on the coupled model parts that you specify for the corresponding zone in the partner simulation.

Specifying the Exchanged Fields

For each co-simulation zone, specify the fields that are exported and imported during the co-simulation. STAR-CCM+ allows you to specify the exchanged fields using predefined mechanical and thermal fields, for standard Fluid-Structure Interaction (FSI) and Conjugate Heat Transfer analyses (CHT), or auxiliary fields, for generic data exchange.

When you assign boundaries to a co-simulation zone, STAR-CCM+ automatically adds predefined exported and imported field specifications to the zone, based on the simulation set up:

- If the simulations contain mechanical settings (that is, the fluid region is using the **Morphing** motion, or the solid continuum contains the **Finite Volume Solid Stress** model), STAR-CCM+ adds predefined imported and exported fields for FSI (fluid loads and solid displacement).
- If the simulations contain thermal settings (that is, the fluid and solid physics continua contain *Energy* models, such as the **Segregated Fluid Temperature** and the **Segregated Solid Energy** models), STAR-CCM+ adds predefined imported and exported fields for thermal co-simulation (convective flux and temperature).

You can modify the predefined set up or add auxiliary fields. Predefined fields, and auxiliary fields, can coexist in the same co-simulation zone. For more information, see Exchanged Fields Reference.

To exchange predefined mechanical fields, that is, fluid loads and solid displacement:

1. In the fluid simulation, define a morphing motion for the fluid region:
   a) Right-click the **Tools** > **Motions** node and select **New** > **Morphing**.
   b) Expand the **Regions** > **[Region]** > **Physics Values** node.
   c) Select the **Motion Specification** node and set *Motion* to **Morphing**.
2. In the solid simulation, activate the **Finite Volume Solid Stress** model.

   The finite element **Solid Stress** model is not compatible with co-simulation.

Specify the imported and exported fields in the fluid simulation:

3. In the fluid simulation, expand the relevant **[Zone]** > **Conditions** > **Exported Fields** node.

When the physics continuum contains a *Flow* model, STAR-CCM+ automatically adds **Pressure** to the exported fields. You can change the predefined exported field to **Traction** (pressure + wall shear stress), or deactivate mechanical export for the fluid simulation to run a one-way coupled simulation:

- To deactivate mechanical export, right-click the **Pressure** node and select **Delete**.
- To export **Traction** (pressure and wall shear stress):
  1. Delete the **Pressure** node.
  2. Right-click the **Exported Fields** node and select **Add Mechanical Field** > **Traction**.

4. Expand the **Conditions** > **Imported Fields** node.

   By default, the fluid simulation imports **Displacement**. This is the only predefined mechanical field available for import in the fluid simulation. You can delete the **Displacement** node to deactivate mechanical import for the fluid simulation (one-way coupling).

Specify the exported and imported fields in the solid simulation.

5. In the solid simulation, identify the **[Zone]** that is coupled with the fluid zone that you set in the previous steps. Expand the **[Zone]** > **Conditions** node.

6. Specify the imported field using the same procedure for specifying the exported fields in the fluid simulation.

   By default, the solid simulation imports **Pressure** and exports **Displacement**. **Traction** is also available for import. Make sure that the fields that are imported or exported by the solid simulation match the fields that are exported or imported by the partner simulation.

   Note: In mechanical co-simulation analyses, you can also access the settings for the FSI field treatment by activating the **Partitioned Coupling** model in each of the partner simulations. For more information, see Partitioned Coupling for FSI Applications.

To exchange predefined thermal fields, that is, convective fluxes and temperature:

7. In both simulations, activate an *Energy* model on the relevant physics continua.
8. In the fluid simulation, expand the relevant **[Zone]** > **Conditions** > **Exported Fields** node.

   By default, the fluid simulation exports **Convective Flux**. This is the only predefined thermal field available for export in the fluid simulation. You can delete the **Convective Flux** node to deactivate thermal export for the fluid simulation (one-way coupling).

When exporting *Convective Flux*, choose consistent heat transfer coefficient and reference temperature, for example:

Heat Transfer Coefficient	Reference Temperature
Local Heat Transfer Coefficient	**Local Heat Transfer Reference Temperature**
Specified Y+ Heat Transfer Coefficient	**Specified Y+ Heat Transfer Reference Temperature**

9. Select the **Exported Fields** > **Convective Flux** node and set *Heat Transfer Coefficient* and *Reference Temperature* to the relevant field functions:
   - If the simulation involves radiation, use the **Local Heat Transfer Reference Temperature**.
   - For most FSI applications, use the **Specified Y+ Heat Transfer Reference Temperature**.

When the fluid continuum contains a *Flow* physics model, STAR-CCM+ automatically adds **Pressure** to the fields exported by the fluid simulation. In thermal co-simulation, exporting pressure is not required. To remove pressure from the exported fields:

- Right-click the **Pressure** node and select **Delete**.

10. Expand the **Conditions** > **Imported Fields** node.

    By default, the fluid simulation imports **Temperature**. This is the only predefined thermal field available for import in the fluid simulation. You can delete the **Temperature** node to deactivate thermal import for the fluid simulation (one-way coupling).

Specify the exported and imported fields in the solid simulation.

11. In the solid simulation, identify the **[Zone]** that is coupled with the fluid zone that you set in the previous steps.
12. Expand the **[Zone]** > **Conditions** > **Exported Fields** node.

    By default, the solid simulation exports **Temperature**. This is the only predefined thermal field available for export in the solid simulation. You can delete the **Temperature** node to deactivate thermal export for the solid simulation (one-way coupling).

The convective fluxes exported by the fluid simulation can be imported by the solid simulation as convective fluxes or environmental fluxes. Both the **Convective Flux** and **Environmental Flux** import methods allow the solid simulation to import the same thermal fields, but under different conditions. See Thermal Specification.

13. Expand the **Conditions** > **Imported Fields** node.

    By default, the solid simulation imports **Convective Flux**.
14. You can change the predefined imported field to **Environmental Flux**, or deactivate thermal import for the solid simulation (one-way coupling).
    - To deactivate thermal import, right-click the **Convective Flux** node and select **Delete**.
    - To import **Environmental Flux**:
      1. Delete the **Convective Flux** node.
      2. Right-click the **Imported Fields** node and select **Add Thermal Field** > **Environmental Flux**.

In each simulation, STAR-CCM+ automatically applies the predefined imported fields to the coupled boundaries, setting the relevant boundary conditions.

- If you wish to manipulate the imported fields before applying them to the boundaries, deactivate the **Managed** option on the relevant imported fields and set all boundary conditions manually. For more information, see Exchanged Fields Reference.

To export or import auxiliary fields:

15. Right-click the **Exported Fields** or the **Imported Fields** node, as required.
16. Select **Add Auxiliary Field** and choose between **Scalar** or **Vector**, as appropriate.
17. Select the **[Auxiliary Field]** node and specify the field function to export/import.

    Set all boundary conditions manually.

For more information on all the properties available to imported and exported fields, see Exchanged Fields Reference.

To exchange data between partner simulations, STAR-CCM+ maps the specified fields between simulations across the coupled boundaries or regions. You control which fields are mapped by the current simulation, and which fields are mapped by the partner simulation, using the **Mapping Option** condition. See STAR-CCM+ Link Reference.

For each field mapped by the current simulation, STAR-CCM+ adds a mapper control node under the relevant **[Zone]** > **Values** node. These mapper nodes reference data mappers that are defined within the **Tools** > **Data Mappers** node.

You can either create data mappers manually or rely on STAR-CCM+ to create them automatically. If you do not create a data mapper for each of the mappers under the **[Zone]** > **Values** node, then one is created using default values when the co-simulation starts.

If the current simulation requires non-default control of the data mapping:

18. Create a **Data Mapper** within the **Tools** > **Data Mappers** node and assign it to the corresponding mapper node under the **[Zone]** > **Values** node.
19. For the data mapper, specify proximity constraints, normal direction constraints, interpolation method and any other settings. STAR-CCM+ automatically sets part and field properties in both the source and target.

    Note: You can inspect properties of a mapper you have created at any time, but you can inspect the properties of default mappers only after the co-simulation run has started.

Partitioned Coupling for FSI Applications

The **Partitioned Coupling** model allows you to control the convergence of the coupled problem in FSI applications. This model exposes additional settings for controlling the application of the fluid loads and convergence of the solid displacement.

1. Set up the fluid and solid simulations for co-simulation, as required.

   In a two-way mechanical co-simulation analysis, the solid simulation solves for the structural displacement in response to the fluid loads imported from the fluid simulation. The fluid simulation, in turn, uses the structural displacements from the solid simulation as inputs to the mesh morpher. For more information, see STAR-CCM+ Co-Simulation Workflow.
2. In the fluid simulation, set up the morpher for the fluid region. For instructions, see Activating the Morpher.

   Since the **Partitioned Coupling** model controls the mesh morpher in the fluid simulation, make sure that the *Morph at Inner Iterations* property for the **Mesh Morpher Solver** is deactivated.
3. In both simulations, activate the **Partitioned Coupling** model in the relevant physics continuum.

   The **Partitioned Coupling** model becomes available for selection after you select a STAR-CCM+ co-simulation model. See Preparing the Simulations for Co-Simulation. When you activate the **Partitioned Coupling** model, STAR-CCM+ automatically adds the **Fixed-Point Iteration** model.

   - When using the **Partitioned Coupling** model with the **STAR-CCM+ Explicit Coupling** model, the coupling algorithm only works at the beginning of each time-step.

     Under-relaxation of solid displacement and fluid load control is applied only at the beginning of the time-step. No equilibrium is ensured at the fluid-structure interface during the time-step evaluation.
   - When using the **Partitioned Coupling** model with the **STAR-CCM+ Implicit Coupling** model, the coupling algorithm works at all inner iterations within each time-step. This ensures equilibrium at the fluid-structure interface in each time-step.
4. In each simulation, set the FSI coupling controls:
   - In the fluid simulation, you can specify the under-relaxation factor for the imported solid displacement. See FSI Under-Relaxation.
   - In the solid simulation, you can linearly ramp the imported fluid loads over a specified time interval, and limit their values. See FSI Fluid Load Control Option.

In each simulation, STAR-CCM+ adds a mapper control for each field that is exchanged during the co-simulation. For information on how to set up the data mapper properties, see Specifying the Exchanged Fields and STAR-CCM+ Link Reference.

5. When using the **Partitioned Coupling** model, activate the *Use Original Mesh* mapper option in all cases that do not involve sliding between the fluid and solid boundaries.

   The *Use Original Mesh* mapper option determines whether mapping is performed based on the original mesh or the deformed mesh. When this option is activated, the data mapper performs mapping based on the initial configuration of the fluid-structure boundaries. When this option is deactivated, the data mapper information is recomputed for every mapping operation based on the current coordinates of the updated meshes, increasing the computational effort.

Specifying the Data Exchange Frequency

In steady simulations, you specify the frequency at which data is exchanged in terms of the solver iterations. In transient simulations that use the implicit coupling scheme, you specify the exchange frequency in terms of the solver inner iterations. In transient simulations that use the explicit coupling scheme, you specify a coupling time interval.

1. In each simulation, specify the frequency at which data is exchanged. The method for specifying the exchange frequency depends on the STAR-CCM+ co-simulation model in use:

STAR-CCM+ Co-Simulation Model	Workflow
STAR-CCM+ Steady Coupling	Specify the number of iterations between exchanges: 1. Select the **[Link 1]** > **Conditions** > **Field Exchange Controls** node and set *Iterations Per Exchange* to the desired value.
STAR-CCM+ Implicit Coupling	Specify the number of inner iterations between exchanges: 1. Select the **[Link 1]** > **Conditions** > **Field Exchange Controls** node and set *Iterations Per Exchange* to the desired value.
STAR-CCM+ Explicit Coupling	Specify the coupling time interval and the strategy that STAR-CCM+ uses to adjust the solver time-step: 1. Select the **[Link 1]** > **Conditions** > **Coupling Negotiation Option** node and choose the method that STAR-CCM+ uses to determine the coupling time. You can specify the same coupling time interval in both simulations, or set the simulations to have independent coupling time intervals. You can specify a value or you can use the time-step of one of the simulations as coupling time interval. For more information on the available settings, see Coupling Negotiation Option. 2. Select the **[Link 1]** > **Values** > **Coupling Time Interval** node and specify the desired coupling time. Depending on the selected **Coupling Negotiation Option**, this option can be read-only. As the explicit scheme allows the simulations to exchange data not more than once per time-step, make sure that the specified coupling time interval is equal to or greater than the simulation time-step. 3. Select the **[Link 1]** > **Values** > **Time-Step Adjustment Option** node and specify how STAR-CCM+ treats the simulation time-step when the remaining time that is required to meet the next coupling time is not an exact multiple of the simulation time-step. For details, see Time-Step Adjustment Option.

Specifying Boundary Conditions on Uncoupled Faces

At a coupled boundary, you define boundary conditions using the data imported from the partner simulation. However, if the coupled boundaries are only partially in contact, or have very different meshes, not all the

faces on a coupled boundary mesh are successfully matched with faces on the partner boundary mesh. For these uncoupled faces, which do not participate in the co-simulation, you can define independent boundary conditions.

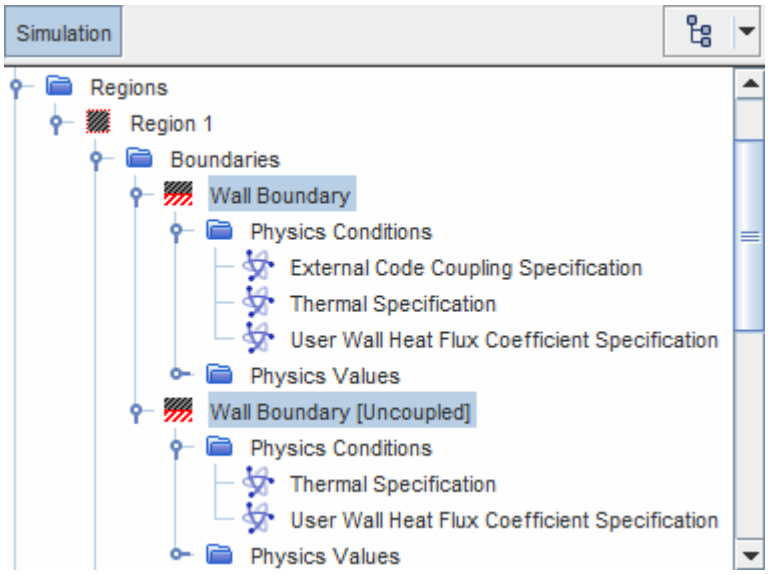
To define physics conditions on uncoupled faces:

1. In each simulation, activate the **Uncoupled Mesh Entities Treatment** model in the relevant physics continuum.

   For information on the model requirements and settings, see Uncoupled Mesh Entities Treatment Model Reference.

2. Define co-simulation settings as outlined in the previous sections.

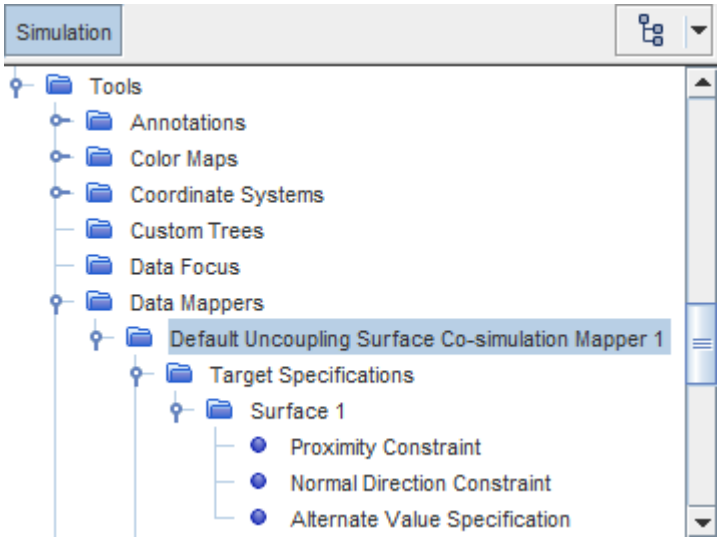
   When you assign a boundary to a co-simulation zone, STAR-CCM+ splits the original boundary into two different boundaries:



   On initialization, STAR-CCM+ uses a data mapper to identify the boundary mesh faces that do not map successfully to the partner boundary mesh. STAR-CCM+ assigns these faces to the **[Uncoupled]** boundary, which does not exchange data with the partner simulation.

By default, STAR-CCM+ creates a data mapper that splits the boundary faces based on default constraints. To change the default constraints:

3. Expand the **Tools** > **Data Mappers** > **[Default Uncoupling Surface Co-simulation Mapper]** node.
4. Select the **Target Specifications** > **[Surface]** > **Proximity Constraint** node or **Normal Direction Constraint** node and specify the desired settings.



Note: STAR-CCM+ only uses the uncoupling mapper to identify unmatched faces. As this mapper is not used for mapping solution data, STAR-CCM+ ignores any setting other than the **Proximity Constraint** and the **Normal Direction Constraint**.

If you want to select a different uncoupling mapper:

5. Expand the relevant **Co-Simulations** > **Link** > **Zones** > **[Zone]** node.
6. Select the **Values** > **Uncoupling Mapper** node and set `Uncoupling Mapper` to the desired mapper.

   For information on creating data mappers, see Creating a Data Mapper.
7. Specify the relevant boundary conditions for the **[Uncoupled]** boundary.

Running the Co-Simulation

Before running a co-simulation interactively, specify the co-simulation settings in each simulation as described in the previous sections.

To run a co-simulation:

1. Load the simulations.
2. Specify stopping criteria in each simulation, so that each simulation provides the correct number of exchanges.

By default, the simulations disconnect at the end of the co-simulation run. Optionally, you can instruct STAR-CCM+ to preserve the connection:

- Select the **Co-Simulations** > **[Link 1]** > **Conditions** > **Disconnect Option** node and set *Option* to **Stay Connected**.

3. Click (**Run**) in either simulation.

   The simulation where you click **Run** creates the connection and overrides any inconsistent settings in the partner simulation. The simulation that is set to **Lead** runs first. The coupled model parts are exchanged as an imported region in each simulation. Mapping of solution data occurs as specified in the **Mapping Option**.

STAR-CCM+ reports details for each data exchange process in the *Output* window. You can monitor the co-simulation progress by selecting the **[Link 1]** > **Values** > **Co-Simulation Progression** node.

The simulations continue to run and send data until their individual stopping criteria are satisfied.

To manually stop a co-simulation:

4. Press **(Stop Iterating)** in any of the participating simulations.

   If you stop in the simulation that is running, the co-simulation stops immediately. If you stop the simulation that is waiting, the co-simulation stops in the following iteration update, that is, after the first iteration is run.

Note: Do not press **(Stop Iterating)** in both simulations as this option is not supported.

If the simulation is still connected to the partner, you can manually disconnect it after the co-simulation run:

5. Right-click the **Co-Simulations** > **[Link 1]** node and select **Disconnect**.

Contents:

Running the Co-Simulation in Batch Mode
Restarting the Co-Simulation

Running the Co-Simulation in Batch Mode

When you run a co-simulation in batch, you manually submit one of the simulations to the batch queue. This simulation automatically launches the partner simulation in server mode and connects to it.

1. Before running the co-simulation, specify co-simulation settings in each simulation as described in the previous sections.

During the co-simulation, the client only connects to the server that you submit to the batch queue. For this reason, the simulation that you submit to the batch queue is typically the simulation that you want to monitor during the co-simulation run.

2. In the simulation that you submit to the batch queue, specify the settings for automatically launching the partner simulation:
   a) Select the **Co-Simulations** > **[Link 1]** > **Conditions** > **Launch Partner Option** node and set *Option* to **Launch Default Application**.
   b) Select the **[Link 1]** > **Values** > **Launch STAR-CCM+** node and set *STAR-CCM+ Launch Command* to:
      - On Linux:

```
% <Path_to_STAR-CCM+_executable>/starccm+ -load -server
<partnerSimulationName>.sim
```

      - On Windows:

```
> <Path_to_STAR-CCM+_executable>\starccm+ -load -server
<partnerSimulationName>.sim
```

3. Submit the simulation to the batch queue using the following command:

```
% starccm+ -batch -np N,M <SimulationName>.sim
```

   where `N` and `M` are the number of processes for the simulation and its partner, respectively.

   This command launches the simulation that you submitted to the batch queue on N processes. This simulation automatically launches the specified partner simulation on M processes and connects to it.

   Depending on the run environment, specify all the command line options that are required for running a regular STAR-CCM+ session in parallel, such as `-mpi` or `-ssh`.

In case of errors:

4. Make sure that the standalone simulations can be run in parallel on the same environment.

Restarting the Co-Simulation

You can run a co-simulation in multiple stages. STAR-CCM+ co-simulation supports most of the common restart scenarios, providing that the partner simulations were in a consistent state when the co-simulation stopped.

When running a co-simulation in multiple stages, the **Disconnect Option** allows you to specify whether the simulations stay connected at the end of a run. At the end of a run, you can modify the settings of both or one of the simulations and then restart the co-simulation.

Note: Do not change any properties while the co-simulation is running. Any discrepancy in the settings can break the synchronization between the coupled simulations.

To restart a co-simulation:

1. If one or both simulations are in an inconsistent state, either clear the solution in both simulations, or step each simulation until both reach a consistent state.
2. In both or one of the simulations, change the settings as required.

   Changes to the regions layout, such as combining regions in both or one of the simulations, are not supported in restart scenarios.

To restart the co-simulation:

3. Press (**Run**) in one of the simulations.

   If a synchronization problem occurs, clear the solution in both simulations and then press **Run**.

STAR-CCM+ Co-Simulation Models Reference

The STAR-CCM+ co-simulation models allow you to couple two STAR-CCM+ simulations in a co-simulation analysis.

Model Names	**STAR-CCM+ Steady Coupling**	Available for steady simulations.
	STAR-CCM+ Explicit Coupling	Available for transient simulations. Allows the simulations to exchange data once per time-step.
	STAR-CCM+ Implicit Coupling	Available for transient simulations. Allows the simulations to exchange data more than once per time-step.
Provided By	**[physics continuum]** > **Models** > *STAR-CCM+ Coupling Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **STAR-CCM+ Implicit Coupling**	
Requires	Physics Models: *Space*: **Three Dimensional** *Time*: any *Material*: any *Optional Models*: **Co-Simulation** *Co-Simulation Models*: **STAR-CCM+**	
Activates	Boundary Inputs	**External Code Coupling Specification**. See Boundary Settings.
	Region Inputs	**External Code Coupling Specification**. See Region Settings.
	Solvers	**STAR-CCM+ Co-Simulation**
	Field Functions	**Co-Simulation Exchange**. See Field Functions.
	Other Nodes	**Co-Simulations** > **Link 1** . See STAR-CCM+ Co-Simulation Link Reference.

Boundary Settings

Available for boundaries of type: Outlet, Mass Flow Inlet, Pressure Outlet, Velocity Inlet, Stagnation Inlet, and Wall.

External Code Coupling Specification

Allows you to assign the boundary to a co-simulation zone for coupling. During the co-simulation, the STAR-CCM+ boundary exchanges data with the corresponding coupled surface in the partner simulation, according to the co-simulation zone specifications.

Region Settings

Available for all type of regions. Only available for volume-to-volume coupling.

External Code Coupling Specification

Allows you to assign the region to a co-simulation zone of type **Volume Zone** for coupling. During the co-simulation, the STAR-CCM+ region exchanges data with the corresponding coupled volume in the partner simulation, according to the co-simulation zone specifications.

Note: Volume-to-volume coupling requires you to specify any imported or exported fields using auxiliary fields.

Field Functions

Co-Simulation Exchange

Represents the number of exchanges between the coupled simulations.

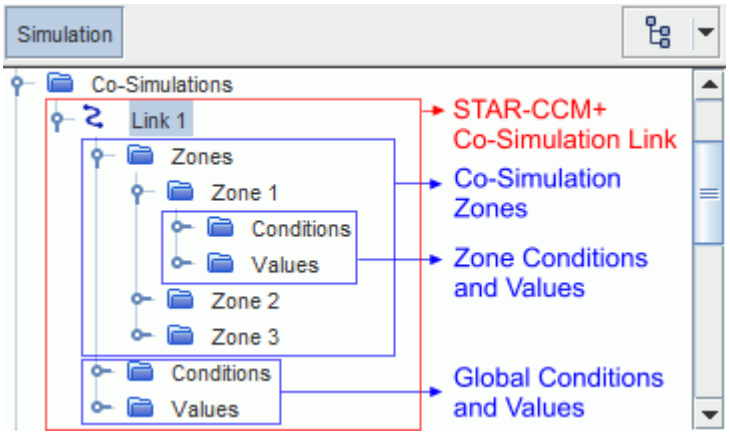
Contents:

STAR-CCM+ Link Reference
Co-Simulation Zones Reference
Exchanged Fields Reference

STAR-CCM+ Link Reference

In STAR-CCM+, you specify the settings for co-simulation in a co-simulation link under the **Co-Simulations** node.

A co-simulation link node, **Co-Simulations** > **[Link 1]**, appears in the simulation tree when you activate a STAR-CCM+ co-simulation model. This node allows you to specify co-simulation global settings, such as the connection method and concurrency mode, as well as to define and set up co-simulation zones, which define sets of boundaries, or regions, where fields are exchanged with the partner simulation.



For details on the co-simulation zone settings, see Co-Simulation Zones Reference.

[Link 1] Properties

Co-Simulation Type

Displays which partner code is coupled to STAR-CCM+ in the co-simulation analysis represented by this link. Read-only.

[Link 1] Conditions and Values

You specify the general settings of the co-simulation under the **[Link 1]** > **Conditions** and **[Link 1]** > **Values** nodes.

Launch Partner Option

Allows you to specify which method STAR-CCM+ uses to launch the partner simulation. You only need to set this option in the simulation where you click **Run**, as STAR-CCM+ automatically specifies consistent settings in the partner simulation.

Method	Corresponding Value Nodes
Do Not Launch Requires you to manually launch the partner simulation. Choose this option when the partner simulation (either a STAR-CCM+ server or a STAR-CCM+ API server) is already launched.	None
Launch Default Application Automatically launches a specified partner simulation on co-simulation run.	**Launch STAR-CCM+** Allows you to specify the options for launching the partner simulation in a new STAR-CCM+ session. You can type the launch command directly, or click ... (**Custom Editor**) to select the simulation file and run mode in the *Load Simulation* dialog.
Launch Application via Command Line Automatically launches a specified partner program on co-simulation run. Selecting this option when coupling partner code to STAR-CCM+ through the Co-Simulation API allows you to use a single shared license for both the STAR-CCM+ and the STAR-CCM+ API server. For more information, see Running a Co-Simulation with Partner Code.	**Launch Command Line** Allows you to type the command to launch the partner program.

Connect Option

Allows you to specify which method the simulation uses to connect to the partner simulation. The available options depend on the selected **Launch Partner Option** method.

Method	Corresponding Value Nodes
Host:Port Allows you to specify the host and port where the partner simulation is running. You only need to set this option in the simulation where you click **Run**. When you click **Run**, the simulation connects to the partner simulation (either a STAR-CCM+ simulation or a STAR-CCM+ API server) that is running on the host and port that you specified. This option is available when you launch the simulations separately, that is, when **Launch Partner Option** is set to **Do Not Launch**.	**STAR-CCM+ Partner Host:Port** Specifies the host and port where the partner simulation is running. Click ... (**Custom Editor**) to select the partner simulation from a list of running STAR-CCM+ servers.
Connection Settings via File Allows you to connect simulations that are located in the same directory without specifying the connection details manually. When you select this option, STAR-CCM+ automatically writes the host and port details of the simulation to a file with extension `.cosim`. The connection file is saved to the simulation directory and has the unique name, `<Link>.<UUID>.cosim`, where `<Link>` is the name of the co-simulation link and `<UUID>` is a unique identifier. If a simulation is new and has not been saved to a directory, STAR-CCM+ saves the connection file to a temporary directory and then moves it to the relevant directory when you save the simulation. As this connection mode requires co-located connection files for both simulations, make sure that the simulations are in the same folder and that you activate this option in both simulations. When you click **Run** in either simulation, the simulation automatically connects to the partner simulation based on the host and port details that are contained in the partner connection file. This option is available when you launch the simulations separately, that is, when **Launch Partner Option** is set to **Do Not Launch**. STAR-CCM+ automatically removes a connection file when you close the corresponding simulation. You can manually delete all connection files or regenerate a connection file using the co-simulation link right-click options. See [Link 1] Right-Click Actions.	**Partner File (read only)** Displays the name of the connection file containing the host and port details of the partner simulation. As STAR-CCM+ identifies the partner connection file by link name, make sure that the co-simulation links associations are uniquely defined. That is, make sure that the directory does not contain more than two simulations with the same co-simulation link name.
Use Launch Info Available when the current simulation automatically launches the partner simulation, that is, when **Launch Partner Option** is set to either **Launch Default Application** or **Launch Application via Command Line**. The current simulation automatically connects to the partner simulation based on the partner launching details.	**Assigned Partner Host:Port** Displays the host and port where the partner simulation is running.

You only need to set this option in the simulation where you click **Run**.

Concurrency Option

Allows you to specify which simulation runs first when the co-simulation starts. The available options are **Lead** and **Lag**. The leading simulation runs first and, when the exchange criteria are met, it sends initial data to the partner simulation. The lagging simulation waits for initial data from the leading simulation before running. When the exchange criteria are met, it sends data back to the leading simulation. The exchange process continues until the stopping criteria are satisfied.

Coupling Negotiation Option

In unsteady simulations, specifies the method to determine the time interval between data exchanges. The following methods are available:

Method	Corresponding Value Nodes
Matched Coupling Time The simulations exchange data at a specified time interval. The time interval between exchanges is the same in both simulations. Select this option in both simulations.	**Coupling Time Interval** Specifies the time interval at which the simulations exchange data. Make sure that you enter a value that is consistent with the simulations time-step (that is, the time-step that you set for the Implicit or Explicit Unsteady solvers).
Independent Coupling Time The simulations exchange data at a specified time interval. You set the time interval between exchanges independently in each simulation. This method is particularly useful when the simulations have different time scales, as the simulations can transfer data at convenient, independent frequencies. Select this option in both simulations.	**Coupling Time Interval** Specifies the time interval at which the current simulation exchanges data. Make sure that you enter a value that is consistent with the simulation time-step.
Time-Step The simulations exchange data at a time interval equal to the current simulation time-step. The time interval between exchanges is the same in both simulations. When using this option, set the partner simulation to use the **Partner Time-Step** option.	**Coupling Time Interval** Displays the time interval at which data is exchanged. Read only.
Partner Time-Step The simulations exchange data at a time interval equal to the partner simulation time-step. The time interval between exchanges is the same in both simulations.	**Coupling Time Interval** Displays the time interval at which data is exchanged. Read only.

When using this option, set the partner simulation to use the **Time-Step** option.	
Maximum Time-Step The simulations exchange data at a time interval equal to the largest of the time-steps (either the current simulation time-step or the partner simulation time-step). The time interval between exchanges is the same in both simulations.	**Coupling Time Interval** Displays the time interval at which data is exchanged. Read only.

Note: To prevent conflict, the simulation where you click **Run** dictates the method that is used in the partner simulation.

Time-Step Adjustment Option (Only for Explicit Coupling)

When using the explicit coupling scheme, the time interval between data exchanges cannot be less than the solution time-step. The time interval between exchanges is determined by the **Coupling Negotiation Option** (see Coupling Negotiation Option). STAR-CCM+

- If the coupling time interval is greater than the simulation time-step, the **Time-Step Adjustment Option** allows you to choose the strategy for adjusting the time-step to reach the coupling time. The available options are:

  Adjust All Time-Steps—At each time-step, if the time to the next exchange is not an exact multiple of the simulation time-step, STAR-CCM+ automatically reduces the size of the subsequent time-steps to meet the coupling time exactly.
  Adjust Last Time-Step—During the last time-step, if the time to the next exchange is not an exact multiple of the simulation time-step, STAR-CCM+ automatically reduces the time-step to meet the coupling time exactly. The previous time-steps are not adjusted and coupling occurs when the coupling time is exactly met or exceeded.
  Do Not Adjust—STAR-CCM+ does not adjust the simulation time-step. Coupling occurs when the coupling time is exactly met or exceeded.
- If the coupling time interval is equal to the simulation time-step or is an exact multiple of the simulation time-step, STAR-CCM+ ignores the settings for this option, as the time-step always meets the coupling time exactly.
- If the coupling time interval is less than the simulation time-step, STAR-CCM+ automatically reduces the simulation time-step to match the coupling time.

This option is only available when using the **STAR-CCM+ Explicit Coupling** model.

Mapping Option

Determines which simulation controls mapping of the exchanged fields. Corresponding mapper controls are added to the relevant **[STAR-CCM+ Co-Simulation]** > **Zones** > **[Zone]** > **Values** nodes. The available options are:

Map Before Export: After the solver runs, the simulation maps the solution data (that you specified for export) to the partner simulation mesh. To have consistent settings, select this option in both simulations, so that each simulation is responsible for mapping the fields that are sent to the partner. As

both simulations are mapping data, each simulation has a copy of both meshes. This option is typically optimal when the solver is run in parallel.

Map After Import: Before the solver runs, the simulation maps the solution data (that you specified for import) from the partner simulation mesh. To have consistent settings, select this option in both simulations, so that each simulation is responsible for mapping the fields that are imported from the partner. As both simulations are mapping data, each simulation has a copy of both meshes.

Map Both Directions: Before the solver runs, the simulation maps the solution data (that you specified for import) from the partner simulation mesh. After the solver runs, the simulation maps the solution data (that you specified for export) to the partner simulation mesh. To have consistent settings, set the partner simulation to use **Partner maps both directions**. Only the simulation that is doing the mapping has a copy of both meshes. You can use this option when one of the simulations has many more CPUs than the other, that is, allow the simulation with the most processors to do the mapping.

Partner Maps Both Directions: The partner simulation maps both imported and exported data. To have consistent settings, set the partner simulation to use **Map Both Directions**.

Note: To prevent conflict, the simulation where you click **Run** enforces the correct mapping option in the partner simulation.

Disconnect Option

Allows you to specify whether the simulation disconnects or stays connected to the partner simulation when the co-simulation is complete.

Field Exchange Controls

Allows you to set the number of iterations between data exchanges:

- When using the **STAR-CCM+ Steady Coupling** model, specify the number of iterations to run before transferring data.
- When using the **STAR-CCM+ Implicit Coupling** model, which allows for data exchange at a frequency smaller than a simulation time-step, specify the number of inner iterations (within a simulation time-step) between exchanges.

Co-Simulation Progression

Displays the current stage in the co-simulation.

Connection to Partner Simulation

Displays the state of connection between the simulations.

External Continua

Displays the physics continua that represent the partner simulation in the current simulation.

[Link 1] Right-Click Actions

Disconnect

Closes the connection between the simulations.

Delete Connection Files

Deletes all connection files from the directory where the current simulation is located.

Create Connection File

Generates a connection file containing the host port details of the current simulation. This option does not generate a connection file for the partner simulation.

Co-Simulation Zones Reference

Co-simulation zones allow you to couple boundaries or regions with corresponding boundaries or regions in the partner simulation. Co-simulation zones also define the fields that are exchanged between the coupled model parts during the co-simulation.

Zones Manager Properties

Co-Simulation Zones

Displays the number of co-simulation zones defined for this link. Read-only.

Zones Manager Right-Click Actions

New

Adds a co-simulation zone under the `Zones` node.

Zone Properties

Index

Specifies a unique index which identifies a zone. A zone is connected with the zone of same index in the partner simulation. Read-only.

Zone Type

Specifies what type of STAR-CCM+ objects are assigned to the co-simulation zone for coupling. The available options are:

Surface Zone: allows you to couple STAR-CCM+ boundaries to their surface counterparts in the partner simulation (surface-to-surface coupling).
Volume Zone: allows you to couple STAR-CCM+ regions to their volume counterparts in the partner simulation (volume-to-volume coupling).

Zone Conditions and Values

The conditions and values of a co-simulation zone allow you to specify the coupled boundaries, or regions, and the fields that are exchanged during the co-simulation.

Coupled Model Parts

Displays the parts on the STAR-CCM+ model that exchange data with the partner simulation, through this co-simulation zone. The coupled model parts can be boundaries (surface-to-surface coupling) or regions (volume-to-volume coupling). Volume-to-volume coupling is only available for exchange of data through auxiliary fields.

This property is read-only. To assign a STAR-CCM+ boundary or region to a co-simulation zone, use the **External Code Coupling Specification** condition for the relevant boundary or region.

External Model Parts

When you run the co-simulation, this property displays the partner simulation model parts that couple with the **Coupled Model Parts** specified for this co-simulation zone. Read-only.

Exported Fields, Imported Fields

Allow you to specify the fields that are exchanged during the co-simulation. For more information, see Exchanged Fields Reference.

Export Mappers

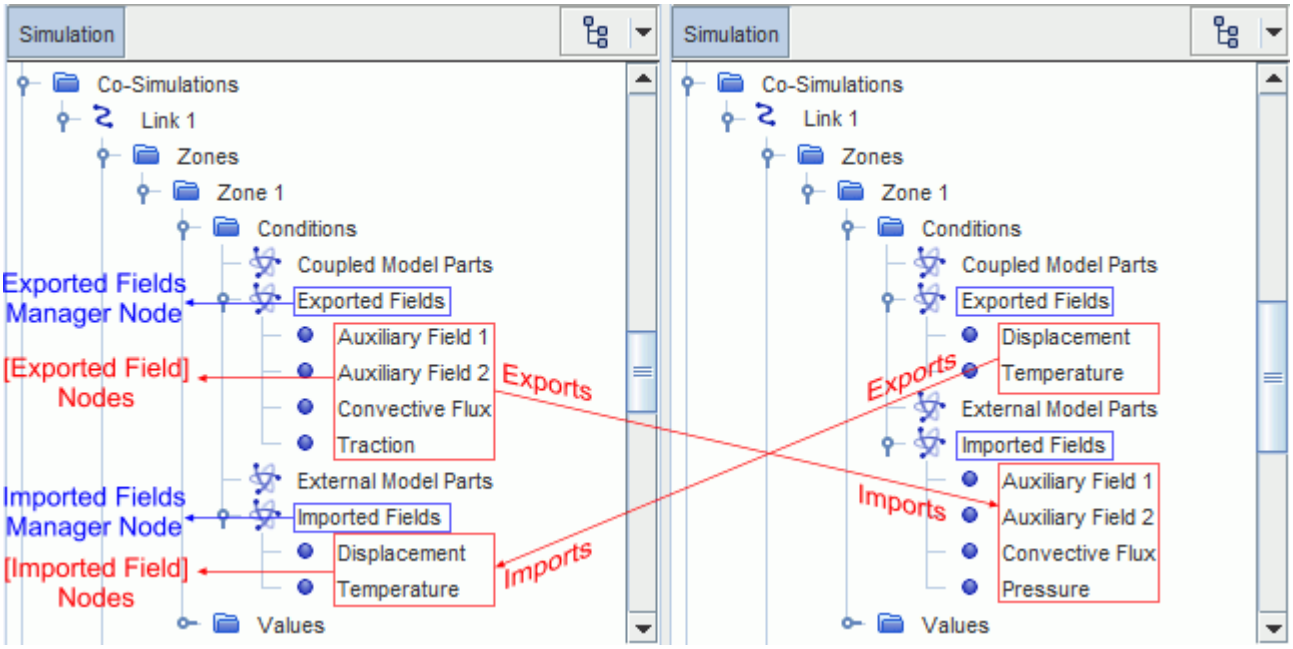
When you set **Mapping Mode** to either **Map Before Export** or **Map Both Directions**, STAR-CCM+ adds a mapper control for each of the field functions selected for export. Each mapper control allows you to select the data mapper that map the field to the partner simulation. You can choose between the default co-simulation mapper, that STAR-CCM+ creates automatically, or any custom mapper available under the **Tools** > **Data Mappers** node.

Import Mappers

When you set **Mapping Mode** to either **Map After Import** or **Map Both Directions**, STAR-CCM+ adds a mapper control for each of the field functions selected for import. Each mapper control allows you to select the data mapper that map the field to the partner simulation. You can choose between the default co-simulation mapper, that STAR-CCM+ creates automatically, or any custom mapper available under the **Tools** > **Data Mappers** node.

Exchanged Fields Reference

In each co-simulation zone, you specify exported and imported fields using predefined thermal and mechanical fields, or auxiliary fields. In each simulation, the exported fields must match the imported fields of the partner simulation.



Exported/Imported Fields Manager Right-Click Actions

Add Thermal Field

Allows you to select a predefined thermal field for export or import. This option becomes available when you activate an *Energy* model on the relevant physics continuum. Before you can add a thermal field, you must delete any existing thermal fields under the manager node.

- **Convective Flux**: available as exported field in fluid simulations, and as imported field in solid simulations.
- **Environmental Flux**: only available as imported field in solid simulations.
- **Temperature**: available as imported field in fluid simulations, and as exported field in solid simulations.

Add Mechanical Field

Allows you to select a predefined mechanical field for export or import.

In the fluid simulation, this option is available for the **Exported Fields** node when you activate a *Flow* physics model, and for the **Imported Fields** node when you assign a **Morphing** motion to the fluid region. In the solid simulation, this option is available for both the **Exported Fields** and the **Imported Fields** nodes when you activate the **Finite Volume Solid Stress** model.

Before you can add a mechanical field, you must delete any existing mechanical fields under the manager node.

The available predefined mechanical fields are:

- **Pressure**: available as exported field in fluid simulations, and as imported field in solid simulations.
- **Traction**: available as exported field in fluid simulations, and as imported field in solid simulations.
- **Displacement**: available as imported field in fluid simulations, and as exported field in solid simulations.

Add Auxiliary Field

Allows you to add a scalar or vector auxiliary field, for export or import of any field function defined on the coupled model parts (either boundaries or regions).

[Exported Field] Properties

The fields under the **Exported Fields** node are subject to the following properties:

Index

Identifies the exported field with a unique number (read-only).

Target Field Location

Allows you to select the location, on the partner simulation mesh, where STAR-CCM+ maps the exported fields. When you couple boundaries across simulations (surface-to-surface coupling), you can choose to interpolate the exported fields onto the vertices or onto the face centroids of the partner boundaries mesh. When you couple regions across simulations (volume-to-volume coupling), you can choose to map the exported auxiliary fields onto the vertices or onto the cell centroids of the partner region mesh.

[Exported Field Function]

Allows you to select the field functions that are exported to the partner simulation. STAR-CCM+ adds a field function to represent the exported field named `Co-Simulation: Exported <field_name>`. For example, `Co-Simulation: Exported Pressure`.

[Imported Field] Properties

Index

Identifies the imported field with a unique number (read-only).

Target Field Location

Displays the location on the mesh, where STAR-CCM+ maps the imported fields (read-only).

Field Treatment

- *Managed*:
  Only available for predefined thermal and mechanical fields. This option is not available for auxiliary fields. STAR-CCM+ automatically applies the imported field to the coupled boundaries, setting the relevant boundary conditions and values as follows:

Imported Field	Wall Boundary Conditions and Values
Displacement	**Morpher Specification** Set to **Displacement**. **Morpher Displacement Specification** Set to **Total**. **Morpher Total Linear Displacement** Set to **Displacement applied by Co-Simulation**.
Pressure	**Stress/Displacement** Set to **Pressure**. **Pressure** Set to **Pressure applied by Co-Simulation**.
Traction	**Stress/Displacement** Set to **Traction**. **Traction** Set to **Traction applied by Co-Simulation**.
Temperature	**Thermal Specification** Set to **Temperature**. **Static Temperature** Set to **Temperature applied by Co-Simulation**.

Convective Flux	**Thermal Specification** Set to **Convection**. **Ambient Temperature** Set to **Reference Temperature applied by Co-Simulation**. **Heat Transfer Coefficient** Set to **Heat Transfer Coefficient applied by Co-Simulation**.
Environmental Flux	**Thermal Specification** Set to **Environment**. **Ambient Temperature** Set to **Reference Temperature applied by Co-Simulation**. **Heat Transfer Coefficient** Set to **Heat Transfer Coefficient applied by Co-Simulation**.

- *Unmanaged*
  You are responsible for setting all boundary conditions manually.

[Imported Field Function]

Displays the field functions that are imported with this field. When the co-simulation starts running, any field imported from the partner simulation becomes available as a field function named `Co-Simulation: Imported <field_name>`.

Partitioned Coupling Model Reference

The **Partitioned Coupling** model exposes additional settings to treat the fields at the fluid-structure interface.

Model Name	**Partitioned Coupling**	
Provided By	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Partitioned Coupling**	
Requires	One of the STAR-CCM+ co-simulation models. See STAR-CCM+ Co-Simulation Models Reference.	
Activates	Link Conditions and Values	**FSI Under-Relaxation Option** (fluid simulation, implicit coupling only) **FSI Displacement Under-Relaxation** (fluid simulation) **FSI Fluid Load Control Option** (solid simulation) See Co-Simulation Link Conditions and Values.
	Solvers	**Fixed-Point Iterative** solver. See Fixed-Point Iterative Solver.
	Monitors	**FSI Displacement**
	Field Functions	**Mapped Morpher Displacement**, **Mapped Displacement At Iteration**, **FSI Under-Relaxation Factor**. See Field Functions.

Co-Simulation Link Conditions and Values

When you activate the **Partitioned Coupling** model, STAR-CCM+ adds additional conditions and values under the relevant **Co-Simulations** > **Link** node. These additional controls allow you to specify the FSI coupling algorithm settings.

For more information on other conditions and values, see STAR-CCM+ Link Reference.

Fluid Simulation

The following settings are only available for the fluid simulation:

FSI Under-Relaxation Option (Only for Implicit Coupling)

In the fluid simulation, specifies the method that STAR-CCM+ uses to under-relax the displacement imported from the solid simulation. The available options are:

Method	Activated Physics Values
Constant STAR-CCM+ applies a constant under-relaxation factor to the solid displacements, starting from the second iteration of each time-step. At the first iteration of each time-step, no under-relaxation factor is applied.	**FSI Displacement Under-Relaxation** Specifies the *Under-Relaxation Factor* used to under-relax the displacement of the structure.

Aitkens

STAR-CCM+ calculates the displacement under-relaxation factor adaptively based on the displacement residuals, according to Aitken's method. The under-relaxation factor is applied at each iteration within a time-step. This method helps to accelerate convergence of the fixed-point algorithm.

Aitkens Urf Bounds

Specifies lower and upper values for the under-relaxation factor calculated using Aitken's method.

FSI Displacement Under-Relaxation

This value specifies the *Under-Relaxation Factor* used to under-relax the displacement of the structure. For explicit coupling, this factor is applied to the solid displacement field at the beginning of each time-step. For implicit coupling, STAR-CCM+ calculates the displacement under-relaxation factor based on the settings specified for the **FSI Under-Relaxation Option** condition.

Solid Simulation

The following settings are only available for the solid simulation:

FSI Fluid Load Control Option

Specifies the method for applying the fluid load on the solid structure. The available methods are:

Method	Activated Physics Values
None Directly applies the imported fluid loads on the solid structure.	None
Linear Ramp & Clipping Allows you to ramp and limit the imported fluid loads.	**FSI Fluid Load Linear Ramp & Clipping** Allows you to specify the following properties: *Time Ramping Parameters*: you can specify an initial and ending time, [t_{start}, t_{end}] s, to apply the imported fluid traction progressively, from zero to the actual value. *Traction Clipping Parameters*: to reflect the actual operating conditions, you can limit the traction values imported from the fluid simulation. For the imported traction load, you can specify the maximum and minimum pressure, as well as the peak shear stress magnitude, [p_{max}, p_{min}, $shear_{max}$] Pa.

Fixed-Point Iterative Solver

The **Fixed-Point Iterative** solver uses the fixed-point iterative method to under-relax the solid displacement at the fluid-structure interface. The solver has the following child node:

Displacement Convergence Tolerance

Sets the convergence tolerance to evaluate convergence during each time-step. When convergence is achieved in a time-step, a message is displayed in the *Output* window. Specify a value in the range [0, 1].

Field Functions

Mapped Morpher Displacement

Displacement that the morpher uses to morph the fluid mesh.

Mapped Displacement At Iteration

Imported solid displacement, before the under-relaxation is applied.

FSI Under-Relaxation Factor

Under-relaxation factor calculated during the time-step.

Uncoupled Mesh Entities Treatment Model Reference

The **Uncoupled Mesh Entities Treatment** model allows you to set independent boundary conditions for the faces, on the coupled boundary mesh, that are not successfully matched with faces on the partner simulation mesh.

Model Name	**Uncoupled Mesh Entities Treatment**	
Provided By	**[physics continuum]** > **Models** > *Optional Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Uncoupled Mesh Entities Treatment**	
Requires	Physics Models: *Space* *Time*: one of **Steady**, **Implicit Unsteady** *Material*: any *Optional Models*: **Co-Simulation** *Co-Simulation Models*: one of **STAR-CCM+ Steady Coupling**, **STAR-CCM+ Explicit Coupling**, **STAR-CCM+ Implicit Coupling**	
Activates	Boundaries	**Boundary [Uncoupled]**. See Boundaries.
	Zone Values	**Uncoupling Mapper**. See Data Mappers.
	Data Mappers	**Default Uncoupling Surface Co-Simulation Mapper**. See Data Mappers.

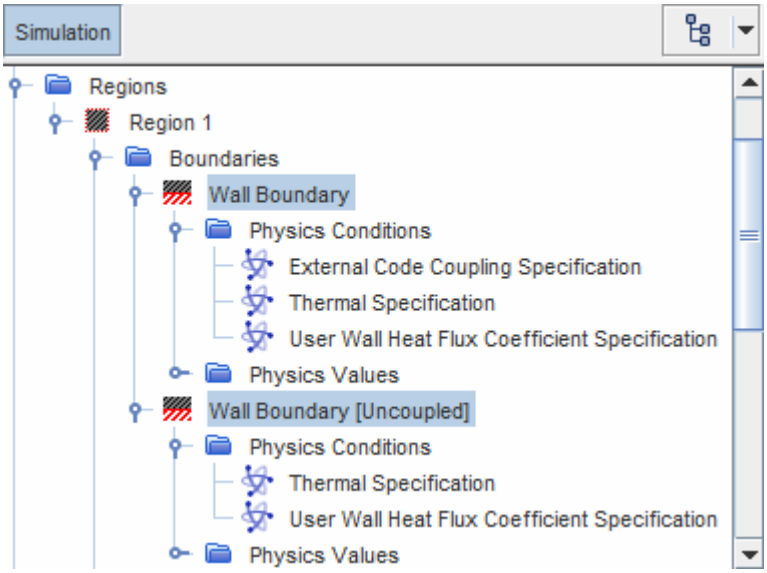
This model is not compatible with motion models other than **Stationary**.

When this model is active, STAR-CCM+ automatically sets the **Mapping Option** to **Map After Import**. See Mapping Option.

Boundaries

Boundary [Uncoupled]

When you assign a boundary to a co-simulation zone, STAR-CCM+ creates an additional boundary named `<Original Boundary Name> [Uncoupled]`. This boundary contains the coupled boundary faces that are not successfully mapped to faces on the partner mesh and, therefore, do not participate in the co-simulation. Use this boundary to define boundary conditions for the faces that fail mapping.



Data Mappers

Default Uncoupling Surface Co-Simulation Mapper

STAR-CCM+ automatically creates this data mapper when you assign a boundary to a co-simulation zone. This data mapper identifies any unmatched faces based on specified proximity and normal direction criteria. The mapper assigns the faces that meet the specified criteria to the original coupled boundary, and assigns the faces that do not meet the criteria to the [Uncoupled] boundary.

STAR-CCM+ uses this data mapper only for splitting the original boundary into matched and unmatched faces. As this mapper is not used for mapping solution data, STAR-CCM+ ignores any settings other than the **Proximity Constraint** and **Normal Direction Constraint**.

Co-Simulation Zone Values

Uncoupling Mapper

Allows you to select the data mapper that identifies matched and unmatched faces.

LMS Amesim Co-Simulation

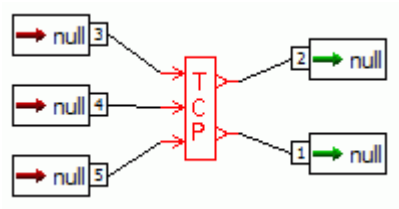
STAR-CCM+ allows for two-way co-simulation with LMS Imagine.Lab Amesim, which is used for modeling dynamic systems.

In a co-simulation between STAR-CCM+ and LMS Amesim, the two codes exchange data interactively while the simulations are running. For example, STAR-CCM+ can calculate the fluid force acting on the surface of a check valve, while LMS Amesim solves for the displacement of the valve in response to the fluid loads.

For more information on LMS Amesim, visit the official website. For a list of all supported versions of LMS Amesim, refer to the section *External Packages Support* of the STAR-CCM+ Release Notes.

Co-Simulation Interface and Exchanged Fields

LMS Amesim communicates with STAR-CCM+ through TCP/IP sockets using a TCP element (the `dynamic_cosim_socket` element). A TCP element contains input and output ports, which exchange data with STAR-CCM+.



The ports on the left side of the element send data to STAR-CCM+, whereas the ports on the right side of the element receive data from STAR-CCM+. STAR-CCM+ applies the fields imported from LMS Amesim to the specified coupled boundaries. STAR-CCM+ must export single-value scalars to LMS Amesim.

STAR-CCM+ allows you to specify the exchanged fields using either:

- Predefined fields—STAR-CCM+ and LMS Amesim can exchange mass flow and pressure fields through co-simulation zones. With this method, you can define boundary conditions using either a managed or unmanaged approach. In the managed approach, STAR-CCM+ automatically defines the boundary conditions at the coupled boundaries, based on the specified imported fields. In the unmanaged approach, you set the boundary conditions manually using the field functions that store the imported data. STAR-CCM+ automatically calculates the mass flow, or pressure, at the coupled boundaries and sends single-value fields to LMS Amesim.
- Auxiliary fields—STAR-CCM+ and LMS Amesim can exchange any single-value scalar field functions. With this method, you set all boundary conditions manually using the field functions that store the imported data. When you specify the exported fields, you are responsible for converting any 3D fields to single-value fields, for example, preparing force, mass flow, or surface average reports.

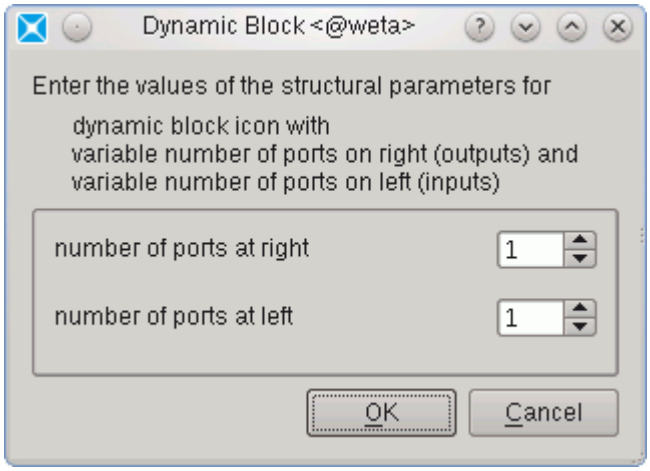
Contents:

Specifying Co-Simulation Settings in LMS Amesim
Specifying Co-Simulation Settings in STAR-CCM+
Coupling STAR-CCM+ to LMS Amesim
LMS Amesim Co-Simulation Model Reference

Specifying Co-Simulation Settings in LMS Amesim

To couple LMS Amesim with STAR-CCM+, you add a TCP element to the LMS Amesim system and you configure it to act as a server.

1. Add a TCP element (the `dynamic_cosim_socket` element) to the LMS Amesim system, with the required number of input and output ports:



   The **number of ports at left** option specifies the number of ports that export data to STAR-CCM+. The **number of ports at right** option specifies the number of ports that import data from STAR-CCM+. For example, if you want to transfer two scalar values from STAR-CCM+ to LMS Amesim, and transfer three scalar values from LMS Amesim to STAR-CCM+, you require a total of five ports on the TCP element: three ports on the left side and two ports on the right side.

Set the TCP element to act as a server. When the TCP element is set as a server, you can set the time-step for the LMS Amesim solver.

2. Click to switch to parameter mode.

   The appropriate license for your LMS Amesim system is checked out.
3. Double-click the `dynamic_cosim_socket` element.

   The *Change Parameters* dialog appears.
4. Set *mode* to **server**.

   A new property, *sample time*, appears in the dialog.
5. Set *sample time* to the desired time-step for your case.
6. Click **Close**.

After configuring the TCP element to act as a server, you can modify the run parameters for the LMS Amesim solver. You can adjust the time-step and total time of the simulation. It is good practice to set the total run time of the LMS Amesim solver to a larger value than the total run time set in STAR-CCM+. When coupled to LMS Amesim, STAR-CCM+ keeps two steps of LMS Amesim data and interpolates between them.

7. Click to switch to simulation mode.
8. Click to set the run parameters.

   The *Run Parameters* dialog appears.
9. Set *Start time* to correspond with the coupling time in STAR-CCM+.

   In STAR-CCM+, the coupling time is given by the value of the **Couple from Step** option multiplied by the simulation time-step. See Couple from Step.
10. Set *Final time* to a value greater than the total simulation time in STAR-CCM+.

    This action is required to make sure that LMS Amesim does not finish solving before STAR-CCM+, due to numerical precision issues.
11. Set *Print interval* to match the sample time-step that you set in the TCP element.

    This property is used for post-processing in LMS Amesim.
12. Click **OK**.

Specifying Co-Simulation Settings in STAR-CCM+

Set up the STAR-CCM+ simulation for co-simulation with LMS Amesim.

1. Activate the following physics models on the relevant fluid continuum:

Group	Physics Model
Space	any
Time	**Implicit Unsteady**
Material	any
Optional Models	**Co-Simulation**
Co-Simulation Models	**Amesim**
Amesim Coupling Models	**Amesim Explicit Coupling** (selected automatically)

   You can select further models to meet individual simulation requirements.

STAR-CCM+ computes pressure relative to the specified reference pressure for the physics continuum. STAR-CCM+ exports this relative pressure and assumes that imported pressures are similarly defined. To ensure consistent pressure exchange between STAR-CCM+ and LMS Amesim:

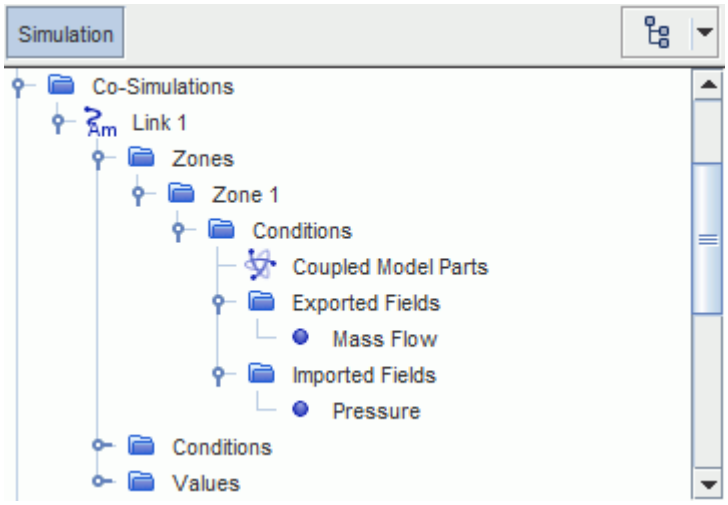
2. Select the **[Physics Continuum]** > **Reference Values** > **Reference Pressure** node and set *Value* to the value of the reference pressure in LMS Amesim.

Decide which method you want to use for exchanging data between the two codes. You can exchange predefined fields (mass flow and pressure) using co-simulation zones. With this method, you can instruct STAR-CCM+ to automatically set the boundary conditions based on the imported data. Alternatively, you can exchange any scalar fields using auxiliary fields. With this method, you manually define boundary conditions using the imported data.

To exchange predefined fields:

3. Right-click the **Co-Simulations** > **Link 1** > **Zones** node and select **New**. Repeat the step to create a co-simulation zone for each boundary that you want to couple with LMS Amesim.
4. To couple a boundary with LMS Amesim, assign it to a co-simulation zone:
   a) Expand the boundary node of interest, **Regions** > **[region]** > **Boundaries** > **[boundary]**.
   b) Select the **Physics Conditions** > **External Code Coupling Specification** node.
   c) Set *External Code Coupling Specification* to a co-simulation zone.
5. Repeat the previous steps as required, to specify all coupled boundaries.

   When you assign a boundary to a co-simulation zone, STAR-CCM+ automatically adds mass flow and pressure fields to the **[Zone]** > **Conditions** > **Exported Fields** and **Imported Fields** nodes.



Depending on the boundary type, mass flow and pressure are selected for either import or export.

To deactivate import or export of a field, delete the relevant node under the **Exported Fields** or **Imported Fields** node.

By default, the *Field Treatment* property of predefined imported fields is set to **Managed**. This option instructs STAR-CCM+ to automatically define boundary conditions using the imported fields.

- If you wish to manipulate the imported fields before applying them to the boundaries, you can change the *Field Treatment* to **Unmanaged**. With this setting, you are responsible for defining all boundaries conditions manually.

To exchange fields other than mass flow and pressure, you use auxiliary fields. You can add auxiliary fields using the **Exported Fields** and **Imported Fields** nodes under either the **[Zone]** > **Conditions** node or the **Link 1** > **Conditions** node. The functionality and workflow is the same.

To export an auxiliary field:

6. Right-click the **Exported Fields** node and select **Add Auxiliary Field** > **Scalar**.
7. Select the **Exported Fields** > **Scalar Auxiliary Field** node and specify the field function that you wish to export to LMS Amesim. Make sure that you select a single-value field function.

   To convert 3D fields to 1D fields, you can use the field functions associated with reports. In STAR-CCM+, all reports automatically generate a field function that returns the value of the report at each iteration. For example, you can define a **Force** report to calculate the shear forces acting on a surface and send this report to LMS Amesim by selecting the field function associated with the report.

Some reports, for example Force reports that include Shear Force, are only available after the flow solution is stepped at least once. If you use such reports in exchanged field specifications:

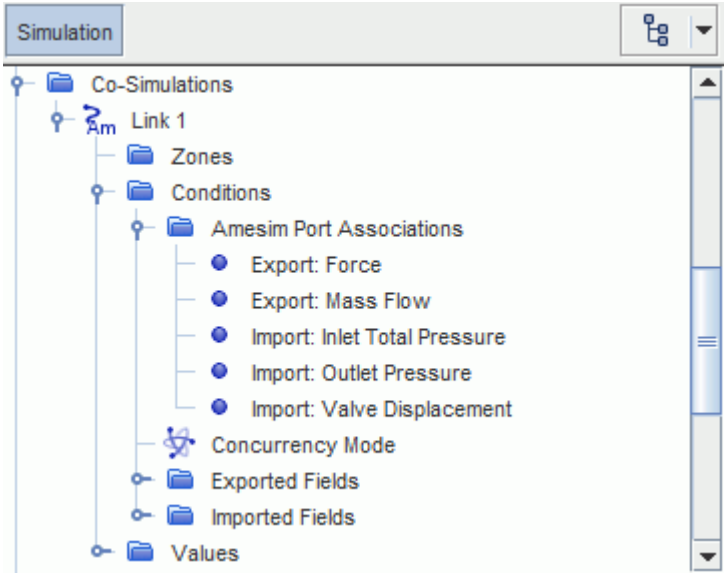
8. Select the **Link** > **Values** > **Couple From Step** node and set *Couple From Step* to `1`.

   For more information, see LMS Amesim Link Reference.

To import an auxiliary field:

9. Right-click the **Imported Fields** node and select **Add Auxiliary Field** > **Scalar**.

For each import and export specification, STAR-CCM+ adds a node under the **Link** > **Values** > **Amesim Port Associations** node.



You use these nodes to associate each imported and exported field with the corresponding LMS Amesim port:

10. Select the **Amesim Port Associations** > **[Field Specification]** node and specify the *Amesim Port*.

    STAR-CCM+ creates two field functions for each field specification:

    `Amesim Port #`, where `#` is the LMS Amesim port number. This field function contains the data imported from, or exported to this LMS Amesim port during the current time-step.
    `Amesim Port # Previous Time Step`, which contains the data imported from, or exported to this LMS Amesim port during the previous time-step.

Imported auxiliary fields require you to set up the boundary conditions manually. To apply the field imported from an LMS Amesim port to the coupled STAR-CCM+ boundary:

11. Expand the relevant **[Region]** > **Boundaries** > **[Boundary]** node.
12. Select the **Physics Values** > **[scalar quantity]** that you want to set using the imported data and set *Method* to **Field Function**.
13. Set *Field Function* to the relevant `Amesim Port #` field function.
14. Complete the simulation set up.

    For a list of all available properties, see LMS Amesim Link Reference.

Coupling STAR-CCM+ to LMS Amesim

Couple STAR-CCM+ with LMS Amesim and run the co-simulation.

1. Set the time-step in both STAR-CCM+ and LMS Amesim.

   It is good practice to set the total run time in LMS Amesim to a larger value than the value set in STAR-CCM+.

The LMS Amesim solver must be already running for STAR-CCM+ to couple with it. To start the LMS Amesim solver:

2. In LMS Amesim, click (**Start a simulation**).

   When an LMS Amesim system contains the `dynamic_cosim_socket` element, the LMS Amesim solver waits for the STAR-CCM+ solver to run before doing any of its own calculations.

To couple STAR-CCM+ and LMS Amesim:

3. In STAR-CCM+, select the **Co-Simulations** > **Link** > **Values** > **Connection Settings** node and specify the name of the machine where LMS Amesim is running and the communication port.
4. Right-click the **Link** node and select **Connect**.

   STAR-CCM+ couples to LMS Amesim. A message summarizing the coupling details appears in the *Output* window.

LMS Amesim Co-Simulation Model Reference

The **Amesim Explicit Coupling** model allows you to couple STAR-CCM+ and LMS Amesim in a two-way co-simulation analysis.

Provided By	**[physics continuum]** > **Models** > *Co-Simulation Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Amesim Explicit Coupling**	
Requires	Physics Models: *Space*: any *Time*: **Implicit Unsteady** *Material*: any *Optional Models*: **Co-Simulation** *Co-Simulation Models*: **Amesim**	
Activates	Boundary Inputs	**External Code Coupling Specification**. See Boundary Settings.
	Solvers	**Amesim Coupling**.
	Other Nodes	**Co-Simulations** > **Link 1**. See LMS Amesim Link Reference.

Boundary Settings

Only available for boundaries of type **Velocity Inlet**, **Mass Flow Inlet**, and **Pressure Outlet**.

External Code Coupling Specification

Allows you to assign the boundary to a co-simulation zone for coupling. During the co-simulation, the STAR-CCM+ boundary exchanges data with the corresponding coupled surface in the partner simulation, according to the co-simulation zone specifications.

Contents:

LMS Amesim Link Reference
Co-Simulation Zones Reference
Exchanged Fields Reference

LMS Amesim Link Reference

In STAR-CCM+, you specify the settings for co-simulation in a co-simulation link under the **Co-Simulations** node.

A co-simulation link node, **Co-Simulations** > **Link 1**, appears in the simulation when you activate the **Amesim** co-simulation model. This node allows you to specify co-simulation global settings, such as the exchanged fields and connection settings, as well as to define and set up co-simulation zones.

For details on the co-simulation zone settings, see Co-Simulation Zones Reference.

Link Properties

Co-Simulation Type

Displays which partner code is coupled to STAR-CCM+ in the co-simulation analysis represented by this link. Read-only.

Link Conditions and Values

You specify the general settings of the co-simulation under the **Link** > **Conditions** and **Link** > **Values** nodes.

Amesim

Displays the connection status.

Connection Settings

Allows you to specify the machine and port where LMS Amesim is running.

Port Number—specifies the port to use when communicating with LMS Amesim. The default port for LMS Amesim co-simulation is `60000`.
Hostname—specifies the name of the machine where LMS Amesim is running. If LMS Amesim is on the same machine as the STAR-CCM+ installation, you can leave this property as `localhost`.

Couple From Step

Allows you to delay the transfer of data to the LMS Amesim solver until a defined time-step.

Specifically, this option specifies how many time-steps STAR-CCM+ runs without sending data to LMS Amesim. As the count starts from zero, the time at which coupling starts is given by the **Couple From Step** value multiplied by the simulation time-step.

This feature allows the STAR-CCM+ solver to reach a level of stability while receiving data from LMS Amesim, before sending data back to LMS Amesim. See Coupling STAR-CCM+ to LMS Amesim.

Amesim Port Associations

Allows you to associate STAR-CCM+ field specifications and LMS Amesim ports. For a field specification, you specify the corresponding LMS Amesim port using the *Associated ID* property.

STAR-CCM+ adds two field functions for each TCP port. These field functions are stored in the **Tools** > **Field Functions** node. The `Amesim Port #` field functions correspond to data from the current time-step, while the `Amesim Port # Previous Time Step` field functions correspond to data from the previous time-step.

Concurrency Mode

Specifies whether the STAR-CCM+ and LMS Amesim times are shifted.

Concurrent—STAR-CCM+ and LMS Amesim start at the same time and run simultaneously.
Concurrent with time shift—STAR-CCM+ and LMS Amesim run simultaneously, but their times are shifted. Specifically, LMS Amesim runs for a time interval equal to the coupling time-step, before STAR-CCM+ starts running.

Exported Fields, Imported Fields

Allow you to specify the exported and imported fields using auxiliary fields. See Exchanged Fields Reference.

Link Right-Click Actions

Disconnect

Terminates the connection between STAR-CCM+ and LMS Amesim.

Connect

Starts the connection, coupling the STAR-CCM+ simulation with the running partner simulation that you specified.

Co-Simulation Zones Reference

Co-simulation zones allow you to couple STAR-CCM+ boundaries with corresponding LMS Amesim ports.

Zones Manager Properties

Co-Simulation Zones

Displays the number of co-simulation zones defined for this link. Read-only.

Zones Manager Right-Click Actions

New

Adds a co-simulation zone under the `Zones` node.

Zone Properties

Index

Specifies a unique index which identifies a zone. Read-only.

Zone Type

Specifies what type of STAR-CCM+ objects are assigned to the co-simulation zone for coupling. In co-simulation with LMS Amesim the available option is **Surface Zone**.

Zone Conditions and Values

The conditions and values of a co-simulation zone allow you to specify the coupled boundaries and the fields that are exchanged during the co-simulation.

Coupled Model Parts

Displays the STAR-CCM+ boundaries that exchange data with LMS Amesim through this co-simulation zone. This property is read-only. To assign a boundary to a co-simulation zone, use the **External Code Coupling Specification** condition for the boundary.

Exported Fields, Imported Fields

Allow you to specify the fields that are exchanged during the co-simulation using either predefined fields or auxiliary fields. For more information, see Exchanged Fields Reference.

Exchanged Fields Reference

You specify exported and imported fields using predefined fields or auxiliary fields.

Exported/Imported Fields Manager Right-Click Actions

Add Flow Field

Only available in co-simulation zones.

Allows you to select a predefined flow field for export or import. The available fields are:

- **Mass Flow**: available as exported field for pressure outlet boundaries, and as imported field for velocity inlet and mass flow inlet boundaries.
- **Pressure**: available as exported field for velocity inlet and mass flow inlet boundaries, and as imported field for pressure outlet boundaries.

  STAR-CCM+ exchanges relative pressure. For consistency, the reference pressure specified in STAR-CCM+ must match the reference pressure in LMS Amesim.

Add Auxiliary Field

Allows you to add a scalar auxiliary field, for export or import of any single-value field function. See Specifying Co-Simulation Settings in STAR-CCM+.

[Exported Field] Properties

The fields under the **Exported Fields** node expose the following properties:

Index

Identifies the exported field with a unique number (read-only).

Target Field Location

Allows you to select the location, on the partner simulation mesh, where STAR-CCM+ maps the exported fields. For fields exported to LMS Amesim, the only option is **Vertex**.

[Exported Field Function]

Allows you to select the field functions that are exported to the partner simulation. For each exported field, STAR-CCM+ also adds two additional field functions:

- `Amesim Port #`, which corresponds to the data exported to LMS Amesim port `#` at the current time-step.
- `Amesim Port # Previous Step`, which corresponds to the data exported to LMS Amesim port `#` at the previous time-step

[Imported Field] Properties

Index

Identifies the imported field with a unique number (read-only).

Target Field Location

Displays the location on the mesh, where STAR-CCM+ maps the imported fields (read-only).

Field Treatment

- *Managed*:
  Only available for predefined fields. This option is not available for auxiliary fields. STAR-CCM+ automatically applies the imported field to the coupled boundaries, setting the relevant boundary conditions and values as follows:

Imported Field	Wall Boundary Conditions and Values
Mass Flow	Depending on the boundary type, either: **Mass Flow Rate** Set to **Coupled to 1D**. or: **Velocity Magnitude** Set to **Coupled to 1D**.
Pressure	**Pressure** Set to **Coupled to 1D**.

- *Unmanaged*
  You are responsible for setting all boundary conditions manually, using the field functions that store the imported data.

[Imported Field Function]

Displays the field functions that store the data imported with this field. For each imported field, STAR-CCM+ adds two field functions:

- `Amesim Port #`, which corresponds to the data imported from LMS Amesim port # at the current time-step.
- `Amesim Port # Previous Step`, which corresponds to the data imported from LMS Amesim port # during the previous time-step

Abaqus Co-Simulation

The co-simulation technique can be used to solve complex fluid-structure interaction (FSI) problems by coupling Abaqus and STAR-CCM+. Mechanical and/or thermal fields can be exchanged across the fluid-structure interface.

Abaqus solves the structural domain and STAR-CCM+ solves the fluid domain. The two domains communicate through the SIMULIA Co-Simulation Engine (CSE).

Most of the features in STAR-CCM+ can be used in a co-simulation, including compressible and incompressible flow, laminar and turbulent flow, heat transfer, and free-surface, multi-species, and multiphase driven flows. The implicit unsteady solver is enabled for use with Abaqus co-simulation. In mechanical co-simulation analyses, the morpher updates the STAR-CCM+ mesh in response to the displacements imported from Abaqus.

Most of the features in Abaqus are also available for use in a co-simulation, including nonlinear materials, nonlinear geometric effects, and contact. The following Abaqus procedures support co-simulation: static stress/displacement, dynamic (implicit and explicit), heat transfer (steady-state and transient), coupled temperature-displacement, coupled thermal-electrical, and piezoelectric analysis. The interface domain can be modeled with most continuum, shell, and membrane elements.

Applications that this FSI capability does not target include:

- Frequency domain FSI problems, including vibro-acoustic and aero-acoustic. Such problems are treated more effectively using frequency domain methods and are not suited for the partitioned solution approach. The Abaqus acoustic capability is applicable for such problems. The Harmonic Balance method (a frequency domain analysis) in STAR-CCM+ is not supported with co-simulation.
- Structures that are modeled with truss, beam, and any special-purpose elements that are wetted by the fluid.

  Such discretization has an inconsistent geometric idealization with the fluid domain. If such structural elements exist in the model, they must be linked to other structural elements that unambiguously define the actual wetted surface. Imported and exported data is then mapped to/from this wetted surface to the corresponding surface in STAR-CCM+.
- Problems involving injection molding, casting, material solidification, and superplastic forming.

  If the fluid-structure interface changes during the simulation, these applications are not suited for the FSI approach. The Abaqus Interface for Moldflow can be applicable (see the Abaqus Interface for Moldflow User’s Guide, for more information).
- Problems involving rupture, penetration, and fragmentation.

  In these applications, the structural deformation can cause a single fluid domain to separate into multiple domains.

- Problems involving flow in porous media where the intention is to account for coupled effects on the fine-scale internal structure of the media (specifically, where both the fluid and solid domain overlap).

  There is no prohibition on the use of the lumped parameter porous resistance model available for regions in STAR-CCM+, but this model only has an impact on the fluid domain that STAR-CCM+ computes. Similarly, you can use the coupled pore fluid diffusion/stress procedure in Abaqus/Standard to model the porous media for the solid domain.

This chapter describes how to use the co-simulation technique with STAR-CCM+ and Abaqus to solve FSI problems. For general information on using Abaqus, refer to the Abaqus Analysis User's Guide.

Coupling Scheme

You can couple STAR-CCM+ with Abaqus using an explicit or implicit coupling scheme. The explicit scheme is appropriate for applications where the structure and the fluid are weakly coupled, so that the effect of the fluid on the structure is negligible compared to the effect of the structure on the fluid. The implicit coupling scheme allows STAR-CCM+ and Abaqus to exchange data more than once per time-step, and is appropriate for applications where the structure and the fluid are strongly coupled.

Contents:

General Requirements
Co-Simulation Interface and Exchanged Fields
General Workflow for Co-Simulation with Abaqus
Specifying Co-Simulation Settings in Abaqus
Specifying Co-Simulation Settings in STAR-CCM+
Running the Co-Simulation and Terminating Abaqus
Restarting from a Previous Abaqus Simulation
Remeshing in STAR-CCM+
Post-Processing the Co-Simulation Results
Allocating Cores to Abaqus in Batch Queuing Systems
Abaqus Co-Simulation Models Reference
Abaqus Link Reference
STAR-CCM+ Properties and Equivalent Abaqus Keywords
Additional Considerations

General Requirements

This section describes the general requirements for running a co-simulation with Abaqus.

Before you start, make sure that you have access to an Abaqus installation. For a list of all supported versions of Abaqus, refer to the section *External Packages Support* of the STAR-CCM+ Release Notes.

Licensing

STAR-CCM+ and Abaqus are separately licensed products, and licenses are required from both SIMULIA and Siemens PLM Software. In addition, a separate license is required for the SIMULIA Co-Simulation Engine. Information on installing and managing Abaqus licenses can be found under Licensing on the Support page at www.3ds.com/products-services/simulia. Information on installing and managing STAR-CCM+ licenses can be found in the Setting Up Licensing section of the STAR-CCM+ Installation Guide. Contact your local SIMULIA or Siemens PLM Software support office for assistance in determining licensing requirements.

SIMULIA Co-Simulation Engine

The SIMULIA Co-Simulation Engine is a software component responsible for communication between Abaqus and STAR-CCM+. It is distributed and installed with the Abaqus software as dynamic link libraries.

For STAR-CCM+ to communicate with Abaqus during the co-simulation, it must have access to these dynamic link libraries. To find the libraries, STAR-CCM+ works through the following three methods, in order, until it has found them:

1. STAR-CCM+ uses the path that the `SIMULIA_CSE_LIBPATH` environment variable sets. If this environment variable exists, and contains a valid path to dynamic link libraries, STAR-CCM+ uses these libraries. Set this path when working with libraries that have been downloaded separately to the main Abaqus installer.
2. STAR-CCM+ locates the libraries relative to the executable path or launch command that is given for Abaqus. The Executable Name property of the **Co-Simulations** > **[Link 1]** > **Values** > **Abaqus Execution** node takes either an Abaqus command or full path to the Abaqus executable. STAR-CCM+ can use this setting to locate the libraries. This setting is sufficient when the default dynamic link libraries contained within the Abaqus installation are used.
3. STAR-CCM+ searches the paths that are set in the `LD_LIBRARY_PATH` environment variable on Linux, or the `PATH` environment variable on Windows. Including any paths added to the program environment using the `-ldlibpath` option on the `starccm+` command.

On rare occasions, it is necessary to obtain updated versions of the dynamic link libraries from the SIMULIA download area. Instructions on downloading libraries separately can be found in the document, “SIMULIA Co-Simulation Engine dynamic libraries for fluid-structure interaction analyses using Abaqus and STAR-CCM+”. This document is located in the Dassault Systems Knowledge Base at: www.3ds.com/support/knowledge-base.

On installing these libraries, set the environment variable `SIMULIA_CSE_LIBPATH` to the path where the dynamic link libraries were installed:

- On Windows platforms:

```
> set SIMULIA_CSE_LIBPATH=<dynamic_libs_dir>\lib
```

- On Unix/Linux platforms:
  - For Korn and Bash shells:

```
% export SIMULIA_CSE_LIBPATH=<dynamic_libs_dir>/lib
```

  - For C and Tcsh shells:

```
% setenv SIMULIA_CSE_LIBPATH <dynamic_libs_dir>/lib
```

where `dynamic_libs_dir` is the path to the directory containing the SIMULIA Co-Simulation Engine dynamic link libraries. Add these settings to your environment or profile.

Co-Simulation Interface and Exchanged Fields

Co-Simulation with Abaqus allows for thermal and mechanical coupling at the fluid-structure interface. In thermal co-simulation analyses, STAR-CCM+ exports boundary heat flux to Abaqus and imports temperature data. In mechanical co-simulation analyses, STAR-CCM+ exports traction data (pressure and wall shear stress) to Abaqus and imports nodal displacements. Co-Simulation with Abaqus allows for implicit or explicit coupling.

Fluid-Structure Interface

During the co-simulation, STAR-CCM+ and Abaqus exchange fields across a shared interface, that you specify by defining co-simulation zones in STAR-CCM+. A co-simulation zone specifies a set of coupled boundaries and their Abaqus counterpart. A co-simulation zone also specifies which fields are exchanged between the coupled model parts specified for that zone.

In a co-simulation with Abaqus, you can exchange data between STAR-CCM+ boundaries and Abaqus surfaces (surface-to-surface coupling), or between STAR-CCM+ boundaries and Abaqus nodes (surface-to-point coupling).

In surface-to-point coupling, STAR-CCM+ maps boundary loads onto the coupled Abaqus node. This feature is intended for coupling STAR-CCM+ with a rigid body in Abaqus. In mechanical co-simulation analyses, the displacement of the Abaqus node in response to the applied loads completely defines the displacement of the rigid body Abaqus model. For thermal co-simulation analyses, the coupled Abaqus node must have temperature specifications. For example, an isothermal rigid body in Abaqus is specified as `*RIGID BODY, ISOTHERMAL=yes`.

Note: Surface-to-point coupling is only available when you select STAR-CCM+ as the mapper code. For more information, see Specifying Co-Simulation Settings in STAR-CCM+.

The coupled STAR-CCM+ boundaries can be wall boundaries, baffle boundaries, and porous baffle boundaries. In most cases, you can use wall boundaries. The use of baffle boundaries or porous baffle boundaries is recommended for modeling the thermal resistance or porosity of a thin structure. In both cases, thermal resistance and porosity are modeled within STAR-CCM+, whereas Abaqus models the structure response to the pressure field sent by STAR-CCM+.

Exchanged Fields

The tables below identify the fields that can be exchanged between Abaqus and STAR-CCM+ during a co-simulation. The first table defines the correspondence between the Abaqus fields and the STAR-CCM+ field functions as well as the dimensionality of the fields.

Field	Abaqus		STAR-CCM+	
Field Name	**Abaqus Field ID**	**Dimensionality**	**Field Function Name***	**Dimensionality**
Displacement, Rotation	U, UR	L	Displacement	L

Field	Abaqus		STAR-CCM+	
Velocity, Rotational Velocity	V, VR	LT^{-1}, T^{-1}	Velocity	LT^{-1}
Traction Vector, Total Moment	CF, CM	F, FL	Wall Shear Stress, Pressure	FL^{-2}
Temperature	NT	θ	Temperature	θ
Heat Flux	CFL	JT^{-1}	Boundary Heat Flux	$JT^{-1}L^{-2}$
Film Properties (heat transfer coefficient, ambient temperature)	CFILM	$JT^{-1}L^{-2}\theta^{-1}$ θ	Boundary Heat Flux, Local Heat Transfer Reference Temperature or Specified Y+ Heat Transfer Reference Temperature	$JT^{-1}L^{-2}$ θ

*STAR-CCM+ creates new field functions to store the imported or exported data. The name of these field functions is the name of the field functions selected for import/export, preceded by `Co-Simulation: Imported/Exported`. For example, **Co-Simulation: Imported Temperature**, or **Co-Simulation: Exported Pressure**.

The following table lists the fields which are imported (incoming) or exported (outgoing) by Abaqus for each available Abaqus analysis procedure type.

Abaqus Procedure	Incoming Fields			Outgoing Fields		
	CF, **CM**	**CFL**	**CFILM**	**U**, **UR**	**V**, **VR**	**NT**
Static	●	-	-	●	-	-
Dynamic (Implicit)	●	-	-	●	●	-
Dynamic (Explicit)	●	-	-	●	●	-
Heat Transfer	-	●	●	-	-	●
Coupled Temperature-Displacement	●	●	●	●	-	●

General Workflow for Co-Simulation with Abaqus

To set up a co-simulation with Abaqus:

1. Develop a standalone STAR-CCM+ fluid simulation with the following considerations:
   - For mechanical co-simulation involving surface displacements, apply provisional fixed wall boundary conditions to the FSI boundary of interest. Alternatively, use a morphing motion model with a prescribed displacement or grid velocity that mimics the actual displacement.
   - For thermal co-simulation analyses, apply provisional temperature boundary conditions to the wall boundary of interest.
2. Develop a standalone Abaqus structural model with the following considerations:
   - Identify the fluid-structure interfaces using surface definitions as required.
   - Verify the model using provisional pressure and/or heat flux loads at the fluid-structure interface. Apply pressure/heat flux load magnitudes that are reasonable and similar to the expected fluid loads. You can specify the assumed pressure/heat flux loads using surface-based loading options (`*DSLOAD`, `*DSFLUX`, and `*SFILM`).
3. Run each stand-alone simulation to convergence before attempting co-simulation. If either stand-alone model is improperly defined or has numerical problems, you are unlikely to obtain a solution for a co-simulation.
4. Remove the provisional loads from each standalone model.
5. Modify the Abaqus input file for co-simulation. See Specifying Co-Simulation Settings in Abaqus.
6. Prepare the STAR-CCM+ model for co-simulation. See Specifying Co-Simulation Settings in STAR-CCM+.
7. Launch and execute the co-simulation from STAR-CCM+. See Running the Co-Simulation and Terminating Abaqus.

Specifying Co-Simulation Settings in Abaqus

Add co-simulation commands to the Abaqus input file, to specify the co-simulation step, the coupled Abaqus surfaces or nodes, and the fields that are exchanged during the co-simulation.

You can specify the co-simulation settings by either modifying the Abaqus input (*.inp) file manually, or by using the *Write Abaqus Keywords* option in STAR-CCM+. This option instructs STAR-CCM+ to automatically add the co-simulation definitions to the Abaqus input (*.inp) file, based on the co-simulation set up in the STAR-CCM+ simulation.

To manually specify co-simulation settings in the Abaqus input (*.inp) file:

1. Open the Abaqus input (*.inp) file with a text editor of your choice.
2. Include co-simulation to the analysis step, by adding the following options in the history section of the input file:

   ```
   *CO-SIMULATION, NAME=<name>, PROGRAM=MULTIPHYSICS
   ```

   Communication between Abaqus and STAR-CCM+ initiates during the co-simulation step and terminates when either program ends the co-simulation. Abaqus terminates the co-simulation event when it reaches the end of the analysis step or when the analysis cannot proceed any further; for example, due to convergence problems.

Define co-simulation regions to specify the Abaqus surfaces or nodes that couple with STAR-CCM+ and the fields exchanged across the co-simulation interface.

3. In the Abaqus `.inp` file, include separate `*CO-SIMULATION REGION` definitions to specify the fields that are imported and exported by Abaqus across the coupled surface/node. For each co-simulation region, set *TYPE* to either `SURFACE` or `NODE` and specify the name of the Abaqus coupled surface/node:
   - For coupling between STAR-CCM+ boundaries and Abaqus surfaces, include the following `*CO-SIMULATION REGION` definitions:

     ```
     *CO-SIMULATION REGION, TYPE=SURFACE, IMPORT
     <surface>, <field_identifier>
     <surface>, <field_identifier>
     ```

     ```
     *CO-SIMULATION REGION, TYPE=SURFACE, EXPORT
     <surface>, <field_identifier>
     <surface>, <field_identifier>
     ```

     where `<surface>` is an element-based surface (see "Element-based surface definition," Section 2.3.2 of the Abaqus Analysis User's Guide). In the first co-simulation region definition, `<field_identifier>` designates the field that Abaqus imports from STAR-CCM+ during the co-simulation. In the second co-simulation region definition, `<field_identifier>` designates the field that Abaqus exports to STAR-CCM+ during the co-simulation.

Note: For each coupling zone defined in the STAR-CCM+ simulation, specify a single unique surface name. This surface can be defined as the union of multiple disconnected surfaces. The same surface name can be specified on multiple lines when defining the imported/exported fields.

- For coupling between STAR-CCM+ boundaries and Abaqus nodes, include the following `*CO-SIMULATION REGION` definitions:

```
*CO-SIMULATION REGION, TYPE=NODE, IMPORT
<node>, <field_identifier>
```

```
*CO-SIMULATION REGION, TYPE=NODE, EXPORT
<node>, <field_identifier>
```

where `<node>` is a node set, containing a single node, defined on the Abaqus model and the `<field_identifier>` designates the field that is imported or exported during the co-simulation.

Note: For each coupling zone defined in the STAR-CCM+ simulation, specify a single unique node name. Coupling between STAR-CCM+ boundaries and an Abaqus node is only available when you select STAR-CCM+ to be the mapper code.

For more information on the fields that are available for exchange, see Co-Simulation Interface and Exchanged Fields.

Optionally, you can write restart frames during the Abaqus analysis so that you can restart the co-simulation in case of termination of the analysis due to a power outage, etc. The time points in which Abaqus saves a restart frame must be synchronized with the Auto Save settings in STAR-CCM+ to provide a consistent state from which to restart the co-simulation.

4. In the co-simulation step definition, add the following keyword to write `n` equally spaced (in time) restart frames to the Abaqus restart file during the co-simulation:

```
*RESTART, WRITE, NUMBER INTERVAL=<n>
```

5. Additionally, to save only the latest restart frame on the restart file (analogous to setting Max Autosaved Files to 1 in STAR-CCM+) use:

```
RESTART, WRITE, NUMBER INTERVAL=<n>, OVERLAY
```

Specifying Co-Simulation Settings in STAR-CCM+

Specify the general settings for co-simulation with Abaqus in the STAR-CCM+ simulation.

1. On the physics continuum that couples with Abaqus, activate the following physics models:

Group	Physics Model
Optional Models	**Co-Simulation**
Co-Simulation Models	**Abaqus**
Abaqus Coupling Models	Select one of the following: **Abaqus Explicit Coupling** **Abaqus Implicit Coupling** For weakly coupled applications, where the effect of the structure on the fluid is much larger than the effect of the fluid on the structure, choose the explicit scheme. This method allows the simulations to exchange data not more than once per time-step. To maintain stability with this scheme, choose a sufficiently small coupling step. For strongly coupled applications, choose the implicit scheme, which allows the simulations to exchange data more than once per time-step, and therefore allows you to use larger time-steps than the explicit scheme.

2. Specify an appropriate simulation time-step. The time-step size determines the time that it takes to solve a case, the stability of the solution, and its ability to capture all physical effects in the case.
   - For steady-state cases with static loads, such as a deformed wing under fluid loads, the effect of time-step on accuracy is minimal. Use time-steps larger than the fluid or structural time scale for the best results.
   - For cases with impact loading, such as fluid loads from sloshing, choose a time-step to resolve the peak load.
   - For cases with oscillations or vibrations, such as releasing a pre-stressed cantilever, set a time-step of one hundredth (1/100) of the expected period of oscillations to capture the behavior of the system adequately.

   To test if the time-step is sufficiently small for accuracy, run a simulation with a time-step of half the original value, and observe if there is a change in results.

Specify the options for launching and running the Abaqus job:

3. Select the **Co-Simulations** > **[Link 1]** > **Values** > **Abaqus Execution** node and set the *Current Job Name*, *Input File*, and *Executable Name* properties as required.

   For more information, see Abaqus Link Reference.

Define whether STAR-CCM+ and Abaqus run simultaneously or sequentially. For implicit coupling, only sequential runs are possible.

4. Select the **[Link 1]** > **Conditions** > **Time-Marching Sequence** node.

- To run both codes simultaneously, set *Option* to **Concurrent**. This option is the least stable option. However, if the computation time per time-step is equal and the partition of processors is balanced, then running the codes simultaneously is faster than running sequentially—assuming that resources are available to run each code on separate hosts.
- To run the codes sequentially during the co-simulation, set *Option* to either **Abaqus Leads** or **STAR-CCM+ Leads**. The stability and computational costs for both these options are identical. Choose the leading code based on the physics that is being solved.

  The leading code should be solving the physics which is least sensitive to the initial prediction of the state at the end of the time-step. In mechanical co-simulation analyses, the guidelines is for the solid mechanics to lead. In this case, the solid uses the data from the previous time-step as initial prediction of the fluid loads at the end of the current time-step. It is not recommended to set the fluid mechanics to lead, as this requires a prediction of the displacement of the solid, which can be accelerating. It is also preferable for the solid to lead because the fluid mesh will be in sync with the solid mesh instead of lagging by an iteration. This is particularly important for explicit coupling.

  In thermal co-simulation analyses, it is best for the fluid mechanics to lead. In this case, the fluid uses the data from the previous time-step as initial prediction for the surface temperature at the end of the current time-step. It is not recommended to set the solid mechanics to lead, as this requires a prediction of the fluid heat transfer coefficient and ambient temperature, and the fluid may require several interactions before it is sufficiently converged to provide a reasonable estimate.

For more information, see Abaqus Link Reference.

Optionally, if you wish STAR-CCM+ to automatically add co-simulation definitions to the Abaqus `*.inp` file, based on the co-simulation settings in STAR-CCM+:

- Select the **[Link 1]** > **Conditions** > **Write Abaqus Keywords** node and set *Option* to **Yes**.

  Use the **[Link 1]** > **Values** > **Abaqus Keywords Step Number** node to specify the step in the Abaqus input file where the co-simulation keywords are inserted.

Specify the method by which the coupling time interval is determined:

5. Select the **[Link 1]** > **Conditions** > **Coupling Negotiation Method** and set *Option* to the desired method.

   In some co-simulations cases, it can be expensive to run STAR-CCM+ and Abaqus at the smallest time scale of the two codes. The user-defined coupling negotiation method allows you to specify different coupling time intervals for each code. This coupling negotiation method is useful for loosely coupled problems. For more information, see Abaqus Link Reference.

When using the **Abaqus Implicit Coupling** model, specify the number of inner iterations between data exchanges:

- Select the **[Link 1]** > **Conditions** > **Field Exchange Controls** and set *Inner Iterations per Exchange* to the desired number of iterations.

For implicit coupling, you can also modify the under-relaxation factor for the imported fields. For more information, see Abaqus Link Reference.

To specify the units for data that is imported from and exported to Abaqus:

6. Expand the **[Link 1]** > **Values** > **External Code Units Manager** node and set the units as required.

Specify whether the mapping of data at the fluid-structure interface is performed by Abaqus or STAR-CCM+:

7. Select the **[Link 1]** > **Conditions** > **Mapping Option** node and choose the desired mapper code.

For more information, see Abaqus Link Reference.

8. Complete the simulation set-up by specifying the coupled boundaries and the fields that are exchanged with Abaqus. See Specifying the Coupled Boundaries and Specifying Exported and Imported Fields.

Contents:

Specifying the Coupled Boundaries
Specifying Exported and Imported Fields

Specifying the Coupled Boundaries

Set up co-simulation zones to couple STAR-CCM+ boundaries with their partner Abaqus surfaces or nodes.

You can define multiple co-simulation zones to apply different conditions on different sets of coupled boundaries. The fields that are exchanged between the codes, and the mapper settings, are set independently for each co-simulation zone.

By default, the **[Link 1]** > **Zones** node contains one co-simulation zone. To create additional zones:

- Right-click the **Zones** node and select **New**.

To couple a STAR-CCM+ boundary with an Abaqus surface, or node:

1. Expand the **Regions** > **[Region]** > **Boundaries** node.
2. Expand the boundary that you want to couple with Abaqus.
3. Select the **Physics Conditions** > **External Code Coupling Specification** node and assign the boundary to a co-simulation zone through the *External Code Coupling Specification* property.

   STAR-CCM+ automatically adds the boundary under the corresponding **[Zone]** > **Conditions** > **Coupled Model Parts** node.

To specify whether the boundary couples with a surface or a node on the Abaqus model:

4. Expand the **Co-simulations** > **[Link 1]** > **Zones** > **[Zone]** node.
5. Select the **Conditions** > **Partner Zone Type** node.
   - To couple with an Abaqus surface, set *Option* to **Surface Zone**.
   - To couple with an Abaqus node, set *Option* to **Point Zone**.
6. Specify the partner Abaqus surface/node:
7. Select the **[Zone]** > **Values** > **Abaqus *CO-SIMULATION REGION Name** node and set *Set Name* to the name of the Abaqus surface/node, including the prefix `ASSEMBLY_` when there is no Assembly defined in the Abaqus model.
   For example, `ASSEMBLY_plate`.

   If *Write Abaqus Keywords* is on, STAR-CCM+ uses the specified name to define the Abaqus co-simulation region in the Abaqus input file.

   For information on how to define co-simulation regions in Abaqus, see Specifying Co-Simulation Settings in Abaqus.
8. If you wish to couple other STAR-CCM+ boundaries with Abaqus using different co-simulation settings, for example different import/export specification, create a new co-simulation zone and repeat the previous steps to specify the coupled boundaries.

To verify that the STAR-CCM+ coupled boundaries and the Abaqus coupled surface/node are co-located, import the Abaqus model and inspect the fluid-structure interface in a scene.

For information on how to specify the exchanged fields for a co-simulation zone, see Specifying Exported and Imported Fields.

Specifying Exported and Imported Fields

For each co-simulation zone, specify the fields that are exchanged between Abaqus and STAR-CCM+ during the co-simulation.

The field functions that you specify for import/export in STAR-CCM+ must match with the exported and imported field types that you set in the Abaqus input file. Fields that are set for export in Abaqus must be set for import in STAR-CCM+. Likewise, fields that are set for export in STAR-CCM+ must be set for import in Abaqus. See Specifying Co-Simulation Settings in Abaqus.

STAR-CCM+ automatically sets up the co-simulation zones for thermal and mechanical coupling based on the simulation settings.

When the physics continuum that contains an Abaqus co-simulation model also contains an energy model for the fluid, STAR-CCM+ automatically adds imported and exported thermal fields to the co-simulation zones. In thermal co-simulation analyses, STAR-CCM+ solves for the heat flux at the coupled boundaries, whereas Abaqus solves for temperature.

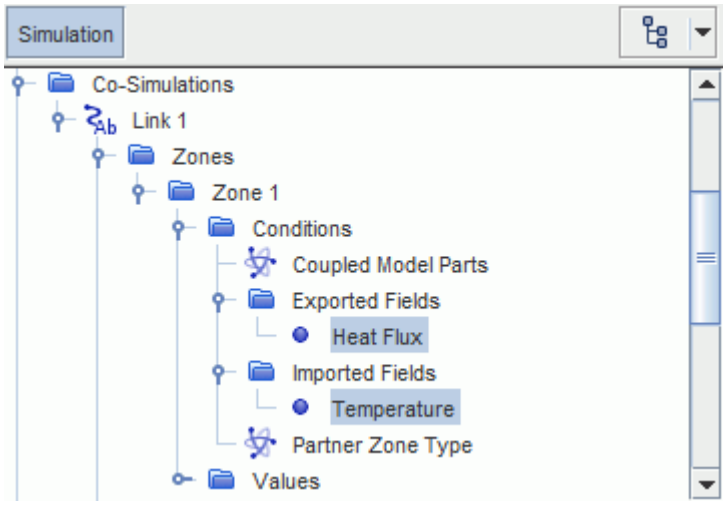
To specify thermal settings:

1. Make sure that you assign the relevant boundaries to the zone and that you activate an *Energy* model in the fluid continuum.
2. Expand the relevant **[Zone]** > **Conditions** > **Exported Fields** node.

When the physics continuum contains a *Flow* model, STAR-CCM+ automatically adds **Pressure** to the exported fields. In thermal co-simulation, exporting pressure is not required. To deactivate pressure export:

3. Right-click the **Pressure** node and select **Delete**.

   The **Exported Fields** and **Imported Fields** nodes are now set so that STAR-CCM+ exports boundary heat flux and imports temperature.



4. Optionally, you can export boundary heat flux with reference temperature data:
   a) Right-click the **Exported Fields** node and select **Add Thermal Field** > **Reference Temperature**.
   b) Select the **Reference Temperature** node and set *Field Function* to either:
      - **Local Heat Transfer Reference Temperature**, suitable for simulations that involve radiation.
      - **Specified Y+ Heat Transfer Reference Temperature**, suitable for most FSI applications.
5. You can customize the export settings of each exported field node by modifying the node properties.

   For more information, see [Exported Field] Properties.

By default, STAR-CCM+ automatically sets the boundary conditions using the imported fields:

Node	Property	Setting
[coupled boundary] > **Physics Conditions** > **Thermal Specification**	*Condition*	**Temperature**
[coupled boundary] > **Physics Values** > **Static temperature**	*Method*	**Temperature applied by Co-Simulation**

6. If you wish to manipulate the imported temperature field (which is stored in the **Co-Simulation: Imported Temperature** field function) before it is applied to the coupled boundaries, select the **Imported Fields** > **Temperature** node and set *Field Treatment* to **Unmanaged**.

   With this method, you are responsible for setting all boundary conditions manually.

When you activate a *Flow* model and assign morphing motion to the fluid region, STAR-CCM+ automatically adds imported and exported mechanical fields to the co-simulation zones. In mechanical co-simulation analyses, STAR-CCM+ solves for the fluid loads, whereas Abaqus solves for the solid displacement in response to the applied loads. STAR-CCM+ uses the morpher to move the fluid boundaries in response to the imported displacements.

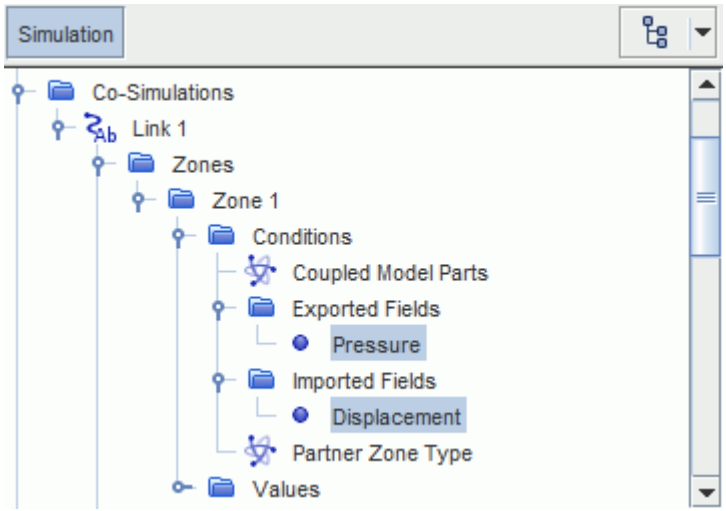
To activate the morpher:

7. Right-click the **Tools** > **Motions** node and select **New** > **Morphing**.
8. Select the **Regions** > **[Co-simulation region]** > **Physics Values** > **Motion Specification** node and set *Motion* to **Morphing**.

To specify mechanical settings:

9. Expand the relevant **[Zone]** > **Conditions** node.

   The **Exported Fields** and **Imported Fields** nodes are automatically set so that STAR-CCM+ exports pressure and imports nodal displacements:



10. To export traction (pressure and wall shear stress), instead of pressure only:
    a) Delete the **Exported Fields** > **Pressure** node.
    b) Right-click the **Exported Fields** node and select **Add Mechanical Field** > **Traction**.
11. You can customize the export settings of each exported field node by modifying the node properties.

    For more information, see [Exported Field] Properties.

By default, STAR-CCM+ automatically sets the boundary conditions using the imported fields:

Node	Property	Setting
[coupled boundary] > Physics Conditions		
└ **Morpher Specification**	*Specification*	**Displacement**
└ **Morpher Displacement Specification**	*Specification*	**Total**
[coupled boundary] > Physics Values > Morpher Total Linear Displacement	*Method*	**Displacements applied by Co-Simulation**

12. If you wish to manipulate the imported displacement field (which is stored in the **Co-Simulation: Imported Displacement** field function) before it is applied to the coupled boundaries, select the **Imported Fields** > **Displacement** node and set *Field Treatment* to **Unmanaged**.

    With this method, you are responsible for setting all boundary conditions manually.

Mechanical analyses can exhibit numerical instabilities which result from high inertia of added-mass, high compliance of the structure, mismatch in initial conditions, or other sources. In such cases, adjust the loading conditions by either progressively increasing the exported traction load, or by limiting its value. This practice generally allows for a larger coupling step size while maintaining numerical stability, although a large coupling step size can result in loss of accuracy.

13. Select the **[Zone]** > **Values** > **Exported Traction Field Controls** node and adjust the loading conditions as appropriate.

    For more information, see Exported Traction Field Controls.

14. Optionally, you can deactivate data export, or import, to run a one-way coupled analysis:
    - To deactivate data export, delete all nodes under the **Exported Fields** node.
    - To deactivate data import, delete all nodes under the **Imported Fields** node.

If you activate the *Write Abaqus Keywords* option, STAR-CCM+ automatically writes the fields definition keywords in the Abaqus input file, according to the import/export conditions specified for the co-simulation zones.

If you do not activate the *Write Abaqus Keywords* option, also specify the imported/exported fields in the Abaqus input file. For more details, see Specifying Co-Simulation Settings in Abaqus.

Running the Co-Simulation and Terminating Abaqus

This section describes the steps for running the co-simulation case.

It is assumed that the model is loaded and prepared for a co-simulation as described in previous sections.

Running the Co-Simulation

Run the co-simulation by clicking the (**Run/Restart**) button in the **Solution** toolbar, or by selecting **Solution** > **Run**. STAR-CCM+ launches Abaqus automatically. The co-simulation progress can be monitored in the *Output* window, and STAR-CCM+ reports output for each data exchange process.

Terminating Abaqus from STAR-CCM+

If Abaqus fails to terminate automatically at the end of the co-simulation, or if you discover that the co-simulation is being conducted using the wrong set of parameters, you can manually terminate the co-simulation. To manually terminate the co-simulation, right-click the **Co-Simulations** > **[Link 1]** node and select **Terminate**.

Restarting from a Previous Abaqus Simulation

An Abaqus restart must be performed when continuing from a previous Abaqus analysis, whether the analysis:

- terminated due to an abort, for example in case of convergence failure in the Abaqus solver or a power failure
- completed successfully (either a co-simulation analysis or an Abaqus stand-alone analysis to pre-load the structure)
- terminated under STAR-CCM+ control when the fluid mesh was remeshed in the middle of an analysis. See Remeshing in STAR-CCM+ .

In order to restart, both the STAR-CCM+ and Abaqus models must be in the same configuration at the point in time of the restart. Significant differences in configuration (such as a mismatch in the FSI boundary mesh positions) results in mapping errors (reported in the STAR-CCM+ log window) or mapping inaccuracies. A good practice is to do both of the following steps when running an analysis:

- Save restart frames at evenly spaced points in the Abaqus analysis. Do this using the `*RESTART, WRITE, NUMBER INTERVAL=<n>` command in the original Abaqus `.inp` file. See Specifying Co-Simulation Settings in Abaqus.
- Synchronize the auto-save feature in STAR-CCM+ to save at the same time intervals.

For analyses that were terminated before the end of the co-simulation analysis this practice ensures that there is a matching restart frame in Abaqus corresponding to a saved `.sim` file in STAR-CCM+.

The mapping configuration must be chosen to suit the type of restart. For co-simulations where both the fluid and the structure are run together in all preceding analyses, set the mapper to use the original coordinates of the coupled FSI boundary. Set this option under the relevant **[Zone]** > **Values** > **Mapper Settings** > **Mapper Configuration** node. In certain cases, there is a mismatch in the configuration that is represented by the original coordinates of each mesh. Examples of such cases are:

- where the fluid analysis is remeshed
- the Abaqus analysis is restarted from a stand-alone pre-loading step in Abaqus

In these cases, set **Mapper Configuration** to **Current Coordinates**.

Define a restart from a point at which the models are in equilibrium. This condition is automatically satisfied when both analyses are started at the same point in time for the restart. If Abaqus is first run stand-alone to pre-load the structure and/or STAR-CCM+ is run with a fixed FSI boundary to develop an initial flow field, traction exported from STAR-CCM+, and loads/boundary conditions applied in Abaqus, may need to be ramped to allow a smooth transition to an equilibrium state at the fluid-structure interface.

To set up a restart analysis:

1. Create an Abaqus `.inp` file with `*RESTART, READ, ...`, corresponding to the restart frame you are restarting from. If you wish to restart from a time point other than the time of completion of the previous step, include the `END STEP` parameter in the `*RESTART` definition. Also, include a new `*STEP` definition with the time duration of the new analysis and any additional structural loading or boundary conditions. Check the Abaqus `.msg` (Abaqus/Standard) or `.sta` (Abaqus/Explicit) files of the previous analysis for information about the restart frames that were written.

2. In STAR-CCM+, select the **[Link 1]** > **Conditions** > **External Code Restart** node and set *Restart Option* to **New Step**.

   A node called **Abaqus Old Job Name** appears under the **[Link 1]** > **Values** node.
3. Specify a new name for the Abaqus job and the new input file through the **Abaqus Execution** properties, *Current Job Name* and *Input File*.
4. Specify the old Abaqus job name using the **[Link 1]** > **Values** > **Abaqus Old Job Name** node.

   The Abaqus restart files from the analysis defined by *Old Job Name* are read at the start of the co-simulation and must be present in the same directory as the *Input File* of the restart analysis.
5. Set the **Mapper Settings** > **Reference Configuration** to **Current Coordinates** if the original configurations of the structure and fluid do not match.
6. Set the STAR-CCM+ stopping time to the current time in STAR-CCM+ plus the Abaqus restart step time. The co-simulation time corresponds to the Abaqus step time and starts from zero each time a co-simulation is restarted.
7. Save the `.sim` file to a new name and click **Run** in STAR-CCM+.

Remeshing in STAR-CCM+

If the deformation of the mesh during the analysis requires the fluid to be remeshed before continuing, the choice of the Mapper Code will define the workflow:

- If *Mapper Code* is set to **STAR-CCM+**, then the analysis can be continued without any intervention.
- If *Mapper Code* is set to **Abaqus**, then the co-simulation will be terminated and must be restarted using the **Current Coordinates** option for the mapper *Reference Configuration*. See Restarting From a Previous Abaqus Simulation.

Post-Processing the Co-Simulation Results

The structural response and interface response for FSI simulations can be post-processed in Abaqus/CAE. The fluid behavior and interface solution can be viewed in STAR-CCM+. Combined fluid and structural solution post-processing can be done using third-party software.

Analyzing with STAR-CCM+

In STAR-CCM+ you can view imported data using the corresponding field function. For example, you can visualize the field functions `Co-Simulation: Imported Displacement`, `Co-Simulation: Imported Velocity`, and `Co-Simulation: Imported Temperature`, in scenes and reports for the coupled boundary. Generated images from the scenes can be exported for use in creating animations using the methods that are described in Creating Images for an Animation. For general information on post-processing in STAR-CCM+, refer to the Analyzing section of the User Guide.

Output Analysis in Abaqus

The structural response and the co-simulation traction vectors (CF), as well as the co-simulation heat flux vectors (CFL) for thermal coupling, can be visualized in Abaqus/CAE. The output file for the co-simulation is located in the directory containing the Abaqus input file. The output file is named `<current job name>.odb`, where `<current job name>` corresponds to the *Current Job Name* property specified in the co-simulation session properties. Details on how to choose the data that is included in the `*.odb` file can be found in "Output to the output database", Section 4.1.3 of the Abaqus Analysis User's Guide.

Imported co-simulation traction and heat flux vectors can be displayed by requesting CF and CFL as nodal field output. The following example illustrates an output request for nodal displacements, nodal traction vector, nodal temperature (NT), and heat flux vectors imported from STAR-CCM+, and computed material point stresses.

Input File Usage:

```
*OUTPUT, FIELD
*NODE OUPUT
 U, CF, NT, CFL
*ELEMENT OUTPUT
 S
```

Allocating Cores to Abaqus in Batch Queuing Systems

When running both STAR-CCM+ and Abaqus as parallel jobs on batch queuing systems, STAR-CCM+ allows you to set the number of cores to allocate to Abaqus in the STAR-CCM+ launch command. It does this through a second parameter on the `-np` option, that is:

```
-np [starccm+_cores],[abaqus_cores]
```

To run an Abaqus Co-Simulation on a batch queuing system:

1. Edit the launch command in the run script and allocate the desired number of cores for each process. In the following example, STAR-CCM+ is allocated 24 cores, and Abaqus is allocated 8 cores:

   ```
   [INSTALL_DIR]/star/bin/starccm+ -np 24,8 -batchsystem <system> -
   batch <macro_file>.java
   ```

   The Abaqus value (`8`) that is defined in the above argument automatically overwrites the value of *Number of CPUs* in the Execution Properties. See Specifying Co-Simulation Settings in Abaqus.

   Note: The STAR-CCM+ CPU Binding feature for parallel jobs can prevent a co-simulation from running in parallel as expected. You are advised to deactivate this option when you launch more than one process per core. To deactivate CPU Binding, launch STAR-CCM+ with the `-cpubind off` argument. In general, deactivating CPU Binding decreases performance. See CPU Binding.

2. Submit the run script to the batch management system.

Once the run script starts, STAR-CCM+ generates a machine file from the hosts that the queuing system allocates. STAR-CCM+ uses the number of cores defined by the first value in the `-np` argument. When STAR-CCM+ subsequently launches Abaqus, it allocates to Abaqus the number of cores defined by the second value in the `-np` argument using hosts not already allocated to run STAR-CCM+ .

The total number of cores requested from the queuing system determines whether STAR-CCM+ and Abaqus are run on separate cores or if they run on the same cores.

- To run each code on separate cores, request the sum of the cores designated on the `-np` argument from the queuing system.
- You can run each code on the same cores for cases where the codes are not running concurrently, (that is, the **Time-Marching Sequence** *Option* is set to **STAR-CCM+ Leads** or **Abaqus Leads**). The memory footprint of each code must fit within the local memory of each host. If these conditions apply, then ask the queuing system for the largest of the two numbers appended to the `-np` argument. The two codes then use the same cores but run in sequence (so that STAR-CCM+ sleeps while Abaqus runs, and vice versa).

Abaqus Co-Simulation Models Reference

The Abaqus co-simulation model allows you to couple STAR-CCM+ and Abaqus in a co-simulation analysis. This type of co-simulation supports implicit or explicit coupling.

Model Names	**Abaqus Explicit Coupling**	Simulations exchange data once per time-step.
	Abaqus Implicit Coupling	Simulations exchange data more than once per time-step.
Provided By	**[physics continuum]** > **Models** > *Abaqus Coupling Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **Abaqus Explicit Coupling**	
Requires	Physics Models: *Space*: **Three Dimensional** *Time*: **Implicit Unsteady** *Material*: any *Optional Models*: **Co-Simulation** *Co-Simulation Models*: **Abaqus**	
Activates	Boundary Inputs	**External Code Coupling Specification**. See Boundary Settings.
	Solvers	**Abaqus Co-Simulation** See Abaqus Co-Simulation Solver Properties.
	Field Functions	**Incremental Nodal Displacement**, **Co-Simulation Exchange**, **Co-Simulation Map Face Error**, **Co-Simulation Map Vertex Error**. See Field Functions.
	Other Nodes	**Co-Simulations** > **[Link 1]**. See Abaqus Link Reference.

Boundary Settings

Available for boundaries of type **Wall**.

External Code Coupling Specification

Allows you to assign the boundary to a co-simulation zone for coupling. During the co-simulation, the STAR-CCM+ boundary exchanges data with the corresponding coupled surface in the partner simulation, according to the co-simulation zone specifications.

Abaqus Co-Simulation Solver Properties

Verbosity

Allows you to choose how much information STAR-CCM+ reports in the *Output* window while the simulation is running. This property is useful for debugging problems when they occur. The available options are **None** , **Low** , and **High** .

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Temporary Storage Retained

When activated, STAR-CCM+ retains additional field data that the solver generates during an iteration. The particular data retained depends on the solver, and becomes available as field functions during subsequent iterations. Deactivated by default.

Field Functions

Incremental Nodal Displacement

Represents the increment of the imported Abaqus displacement over a time-step, that is, the difference between the displacement imported from Abaqus at the current time-step, and the displacement imported from Abaqus during the previous time-step.

You can visualize the **Incremental Nodal Displacement** on the STAR-CCM+ fluid region, or part, that is coupled with Abaqus.

Co-Simulation Exchange

Represents the number of exchanges between STAR-CCM+ and Abaqus.

Co-Simulation Map Face Error

Indicates error when mapping field data onto faces.

Co-Simulation Map Vertex Error

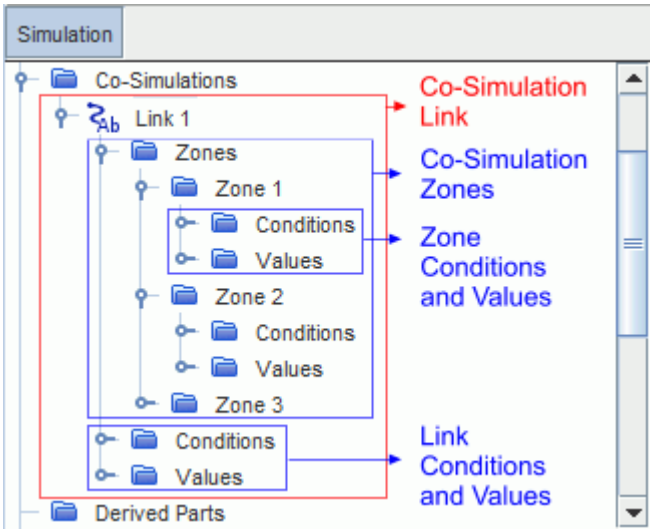
Indicates error when mapping field data onto vertices.

Abaqus Link Reference

In STAR-CCM+, you specify the settings for co-simulation with Abaqus in a co-simulation link under the **Co-Simulations** node.

A co-simulation link, **Co-Simulations** > **[Link 1]**, appears in the simulation tree when you activate the Abaqus co-simulation model. This node allows you to specify co-simulation global settings, such as the connection method and concurrency mode, as well as to define and set up co-simulation zones, which define sets of boundaries, where fields are exchanged with the partner simulation.

For details on the co-simulation zone settings, see Co-Simulation Zones Reference.



[Link 1] Properties

Co-Simulation Type

Displays which partner code is coupled to STAR-CCM+ in the co-simulation analysis represented by this link. Read-only.

[Link 1] Conditions and Values

You specify the general settings of the co-simulation under the **[Link 1]** > **Conditions** and **[Link 1]** > **Values** nodes.

Abaqus Execution

Allows you to specify the options for launching and running the Abaqus job. The available properties are:

Current Job Name -- Defines the current arbitrary job name for the Abaqus Co-Simulation output files. When setting up a co-simulation restart, the *Current Job Name* must be different from the *Old Job Name*. See Restarting from a Previous Abaqus Simulation.

Input File -- Specifies the Abaqus input file for the co-simulation. You can enter either the full path to the file or a path relative to the STAR-CCM+ simulation file.

Executable Name -- Specifies the command to run Abaqus. If you set up a shortcut to run Abaqus in batch, you can enter the shortcut. Otherwise, specify the path to the Abaqus executable, including the executable name.

Additional Command Line Options -- Specifies more command line options for use with the Abaqus executable. Multiple options must be space-separated. Make sure that the command options are compatible with the co-simulation analysis mode. For example, entering `-double both -output_precision full -user usersub.f` would run Abaqus/Explicit in double precision, output results to the output database (`*.odb`) file in double precision, and specify a user subroutine to be used in the analysis.

Number of CPUs -- Specifies the number of processors that Abaqus uses when the co-simulation is run. Must be entered as an integer. If the number of processes for Abaqus is entered using the command-line argument, `-np`, this property is ignored.

Host List -- Specifies the hosts and CPUs per host on which Abaqus is run (for example, host1:2,host2:4).

Remote Shell -- Specifies the type of shell that is used to log in to the remote machine. It is recommended that you use the default, `rsh`, for unsecured networks (usually within a firewall). The other supported option is `ssh`, and must be used when all connections are secure (for example, when `rsh` is blocked). If using `ssh`, make sure that you set it up so that it does not prompt for a password.

Remote User -- Specifies the username for the remote shell login. If this option is left blank, the local username is used.

External Code Units Manager

To specify the units for data that is imported from and exported to Abaqus, use the **[Link 1]** > **Values** > **External Code Units Manager** node. For imported fields, the unit specifications tell STAR-CCM+ the units of the incoming data. For exported fields, the unit specifications ensure that data sent to Abaqus is in the correct units. In each case, these settings depend on which consistent unit system was used when defining the Abaqus model. These settings do not affect the units that are used in STAR-CCM+ calculations.

If a unit used in the Abaqus model is not a pre-defined unit in the STAR-CCM+ units manager, you must create the unit before it appears and can be selected in the child nodes of the **External Code Units Manager** node.

Coupling Negotiation Method

Specifies the method to determine the time interval between data exchanges. For all but the **User Defined** option, the time interval between exchanges is the same in both simulations. The available methods are:

Option	Corresponding Node
Constant The simulations exchange data at a specified time interval.	**Constant Coupling Time Interval** Specifies the coupling time interval.

Min The simulations exchange data at a time interval equal to the smallest of the time-steps (either the current simulation time-step or the partner simulation time-step).	None
Max The simulations exchange data at a time interval equal to the largest of the time-steps (either the current simulation time-step or the partner simulation time-step).	None
Import The simulations exchange data at a time interval equal to the partner simulation time-step.	None
Export The simulations exchange data at a time interval equal to the current simulation time-step.	None
User Defined The simulations exchange data at specified independent time intervals. This option is useful if the simulations are running at different time scales.	**User-Defined Negotiation Method Controls** The available properties are: *STAR-CCM+ Coupling Time Interval*: Specifies the coupling time interval for the STAR-CCM+ simulation. *Abaqus Coupling Time Step*: Specifies the coupling time interval for the Abaqus analysis. *Exported Field Calculation*: Specifies the method that is used to calculate the field values that are exported to Abaqus. Choose **Average**, to average the field values over all time-steps between exchanges. Choose **Instantaneous** to export the field values as they appear at the time of exchange.

External Code Restart

Allows you to specify resume files for co-simulation restart. For more information, see Restarting from a Previous Abaqus Simulation.

Restart Option	Corresponding Child Node
None The simulation is not set up for co-simulation restart.	None

New Step

Defines a co-simulation restart step.

Abaqus Old Job Name

Specifies the name of the old Abaqus job used for restart.

The Abaqus restart files from the analysis defined by *Old Job Name* are read at the start of the co-simulation and must be present in the same directory as the *Input File* of the restart analysis.

Mapping Option

Specifies whether STAR-CCM+ or Abaqus maps the exchanged data. When coupling STAR-CCM+ boundaries with Abaqus surfaces (surface-to-surface coupling), you can select either STAR-CCM+ or Abaqus as the mapper code. However, coupling between STAR-CCM+ boundaries and a node on the Abaqus model (surface-to-point coupling) is only available when you select STAR-CCM+ as the mapper code. The available options are:

Map Both Directions: STAR-CCM+ maps the imported solution data before the solver run starts and the exported solution data after the solver stops. STAR-CCM+ maps its solution onto the Abaqus mesh before exchanging data with Abaqus. Abaqus exchanges its solution data with STAR-CCM+, which then maps the Abaqus solution onto the STAR-CCM+ mesh.

Partner Maps Both Directions: Abaqus maps the imported solution data before the solver run starts and the exported solution data after the solver completes.

Time-Marching Sequence

Defines whether STAR-CCM+ and Abaqus run simultaneously or sequentially. For implicit coupling, only sequential runs are possible. For this node, you can set *Option* to:

Abaqus Leads: sets Abaqus and STAR-CCM+ to run sequentially with Abaqus leading the co-simulation.

STAR-CCM+ Leads: sets Abaqus and STAR-CCM+ to run sequentially with STAR-CCM+ leading the co-simulation.

Concurrent: sets Abaqus and STAR-CCM+ to run simultaneously during the co-simulation. Only available if the coupling scheme is set to explicit.

Write Abaqus Keywords

When activated, instructs STAR-CCM+ to automatically add co-simulation definitions to the Abaqus `*.inp` file, based on the co-simulation set up in STAR-CCM+. STAR-CCM+ overwrites any existing `*CO-SIMULATION` keywords defined in the `*.inp` file. When deactivated, you must manually add the co-simulation parameters to the `*.inp` file.

***Field Exchange Controls* (Only for Abaqus Implicit Coupling)**

Allows you to specify the field exchange settings for implicit coupling. The available properties are:

Inner Iterations Per Exchange -- Specifies the number of inner iterations between data exchanges within a time-step.

Minimum Number of Exchanges -- Sets the lower bound of the total number of exchanges in a time-step.

Field Convergence Tolerance -- Specifies the tolerance for assessing convergence of the displacement field imported from Abaqus. Within a time-step, the exchanges between STAR-CCM+ and Abaqus terminate when the displacement residual, which represents the change of a norm of the imported

displacement field, reaches the specified tolerance, or when the maximum number of inner iterations, set in the stopping criteria, is reached.

Imported Field URF Type -- To accelerate convergence of the implicit loop, you can modify the under-relaxation factor for the imported data, which affects the solution update at each iteration. Setting the *Imported Field URF Type* to **Constant**, applies a fixed under-relaxation factor. Within a time-step, STAR-CCM+ applies an under-relaxation factor equal to 1 at the first data exchange, and an under-relaxation factor equal to the specified *Imported Field URF* at following data exchanges. When you set *Imported Field URF Type* to **Adaptive**, STAR-CCM+ optimizes the solution update by controlling the under-relaxation factor based on the residual. Within a time step, STAR-CCM+ applies an under-relaxation factor equal to 1 at the first data exchange, an under-relaxation factor equal to the specified *Imported Field URF* at the second data exchange, and an under-relaxation factor between the specified *Minimum Adaptive URF* and *Maximum Adaptive URF* at following data exchanges.

Minimum Adaptive URF -- Sets the minimum adaptive under-relaxation value for the imported data.

Maximum Adaptive URF -- Sets the maximum adaptive under-relaxation value for the imported data.

Imported Field URF -- Sets the constant under-relaxation value or initial value for the Adaptive under-relaxation type.

[Link 1] Right-Click Actions

Terminate

Closes the connection between STAR-CCM+ and Abaqus.

Contents:

Co-Simulation Zones Reference
Exchanged Fields Reference
Exported Traction Field Controls

See Also:

Restarting from a Previous Abaqus Simulation

Co-Simulation Zones Reference

In STAR-CCM+, a co-simulation zone defines a set of coupled boundaries that exchange data with a specified Abaqus surface or node.

You can define multiple co-simulation zones to apply different conditions on different sets of coupled boundaries. The fields that are exchanged between the codes, and the mapper settings, are set independently for each co-simulation zone.

A coupling zone can be set up for surface-to-surface coupling, to exchange data between STAR-CCM+ boundaries and Abaqus surfaces, or surface-to-point coupling, for data exchange between STAR-CCM+ boundaries and a node on the Abaqus model.

Zones Manager Properties

Co-Simulation Zones

Displays the number of co-simulation zones defined for this link. Read-only.

Zones Manager Right-Click Actions

New

Adds a co-simulation zone under the `Zones` node.

Zone Properties

Index

Specifies a unique index which identifies a zone. A zone is connected with the zone of same index in the partner simulation. Read-only.

Zone Type

Specifies what type of STAR-CCM+ objects are assigned to the co-simulation zone for coupling. The available option is **Surface Zone**.

Zone Conditions and Values

The conditions and values of a co-simulation zone allow you to specify the coupled boundaries, the corresponding Abaqus co-simulation region, and the fields that are exchanged during the co-simulation.

Abaqus *CO-SIMULATION REGION NAME

Specifies the name of the Abaqus surface, or node, that couples with the STAR-CCM+ boundaries assigned to this zone.

Mapper Settings

Allows you to specify the settings that control mapping of data between the fluid and the structural mesh. A separate mapping operation is done for each type of exchanged field (displacement and temperature versus traction and heat flux) and each direction of mapping (imported versus exported). For fields exported to Abaqus, the fluid boundary acts as a source and the structure surface acts as a target. For fields imported from Abaqus, the structure surface acts as a source and the fluid boundary acts as a target.

The mappers make an initial search using a reference configuration. The position and extent of the coupled fluid boundary and structure surface must match at the beginning of the co-simulation. Possible causes of a mismatch in position are: discretization differences, an error identifying the units that Abaqus uses, or the use of an incorrect reference configuration in the initial search. Each code reports initial (target) search failures for its coupled boundary/surface separately (in the *Output* window in STAR-CCM+ and the `.msg`/`.sta` files in Abaqus).

The available mapping controls are:

Reference Configuration -- Specifies whether the mappers make the initial search using the **Current Coordinates** or the **Original Coordinates** as reference configuration. If you mesh both the fluid and the structure using the same geometry, their original configurations match, and you can set the reference configuration to the original coordinates for the first and subsequent (restart) co-simulations. If you mesh

either the fluid or the structure using different configurations, use the current coordinates for the initial search. An example of this latter scenario is when the structure undergoes some deformation for pre-loading, and the fluid mesh is generated based on the deformed structure.

Proximity Tolerance -- In surface-to-surface coupling, this option specifies the relative search distance to use in the initial search. If STAR-CCM+ mappers are used, the distance is relative to the maximum face size on the coupled boundary source surface. If Abaqus mappers are used, the distance is relative to the average face size on the coupled boundary source surface. This option is not used in surface-to-point coupling, where all the vertices and faces of the STAR-CCM+ coupled boundaries are included in the mapping operation. You can increase the proximity tolerance to resolve small mismatches. However, arbitrarily increasing the proximity tolerance can mask errors in the model setup by allowing the search to find neighbors that are not in close proximity. Matching neighbors that are not in close proximity leads to incorrect results.

Check Normals -- In surface-to-surface coupling, this option checks that the STAR-CCM+ boundary and Abaqus surface are oriented toward one another in the initial search in the mapper. This option is not used in surface-to-point coupling. In cases where initial search failures occur due to discretization differences in areas of large curvature variation, you can deactivate the normal check.

Mapping Rotations -- In surface-to-surface coupling, when you activate this option the mappers add a rotational correction to the displacement mapped on the target surface if this is offset from the source surface. When the source surface undergoes large rotations, activate this option to preserve the shape of the fluid surface as the structure rotates. Deactivate this option when the structure in Abaqus is modeled using shell elements and the deformation causes the physical shell surface to self-intersect during the analysis. In surface-to-point mapping, this option controls whether rotations of the Abaqus node are also imported, and resultant moments exported.

Coupled Model Parts

Displays the boundaries on the STAR-CCM+ model that exchange data with Abaqus, through this co-simulation zone.

This property is read-only. To assign a STAR-CCM+ boundary or region to a co-simulation zone, use the **External Code Coupling Specification** boundary condition.

Exported Fields, Imported Fields

Allow you to specify the fields that are exchanged with Abaqus. For more information, see Exchanged Fields Reference. When you export traction fields, you can adjust the exported traction using the **Values** > **Exported Traction Field Controls** node. See Exported Traction Field Controls.

Partner Zone Type

Specifies whether the boundaries that are assigned to this zone couple with an Abaqus surface or node.

Exchanged Fields Reference

In each co-simulation zone, you specify exported and imported fields using predefined thermal and mechanical fields. The exported and imported fields must match the fields imported and exported by Abaqus.

Exported/Imported Fields Manager Right-Click Actions

Add Thermal Field

Allows you to select a predefined thermal field for export or import. This option becomes available when you activate an *Energy* model on the relevant physics continuum.

- **Heat Flux**: available as exported field.
- **Reference Temperature**: available as additional exported field. Requires the **Heat Flux** exported field.
- **Temperature**: available as imported field.

Add Mechanical Field

Allows you to select a predefined mechanical field for export or import. This option is available for the **Exported Fields** node when you activate a *Flow* model in the relevant physics continuum, and for the **Imported Fields** node when you assign a **Morphing** motion to the fluid region.

- **Pressure**: available as exported field.
- **Traction**: available as exported field.
- **Displacement**: available as imported field.
- **Velocity**: available as additional imported field. Requires the **Displacement** imported field.

  Used for stabilization. If not imported, STAR-CCM+ calculates the velocity from the imported displacement data.

[Exported Field] Properties

The fields under the **Exported Fields** node expose the following properties:

Index

Identifies the exported field with a unique number (read-only).

Target Field Location

Allows you to select the location, on the partner simulation mesh, where STAR-CCM+ maps the exported fields.

[Exported Field Function]

Allows you to select the field function that is exported with this field specification. STAR-CCM+ stores the exported data in an additional field function named `Co-Simulation: Exported <field_name>`. For example, `Co-Simulation: Exported Velocity`.

[Imported Field] Properties

Index

Identifies the imported field with a unique number (read-only).

Target Field Location

Displays the location on the mesh, where STAR-CCM+ maps the imported field (read-only).

Field Treatment

- *Managed*:
  STAR-CCM+ automatically applies the imported field to the coupled boundaries, setting the relevant boundary conditions and values as follows:

Imported Field	Wall Boundary Conditions and Values
Displacement	**Morpher Specification** Set to **Displacement**. **Morpher Displacement Specification** Set to **Total**. **Morpher Total Linear Displacement** Set to **Displacement applied by Co-Simulation**.
Temperature	**Thermal Specification** Set to **Temperature**. **Static Temperature** Set to **Temperature applied by Co-Simulation**.

- *Unmanaged*
  You are responsible for setting all boundary conditions manually.

[Imported Field Function]

Displays the field function that is imported with this field specification. When the co-simulation starts running, any field imported from the partner simulation becomes available as a field function named `Co-Simulation: Imported <field_name>`.

Exported Traction Field Controls

Mechanical co-simulations can exhibit numerical instabilities due to high inertia of added-mass, high compliance of the structure, mismatch in the initial conditions, or other sources. In such cases, adjust the exported traction fields by settings the **[Zone]** > **[Values]** > **Exported Traction Field Controls**.

Pressure Ramping Time Scale

Sets the time scale for the pressure ramping time parameters. The available options are:

Co-Simulation Time -- Employs pressure ramping that is based on co-simulation time.

STAR-CCM+ Time -- Employs pressure ramping that is based on STAR-CCM+ time.

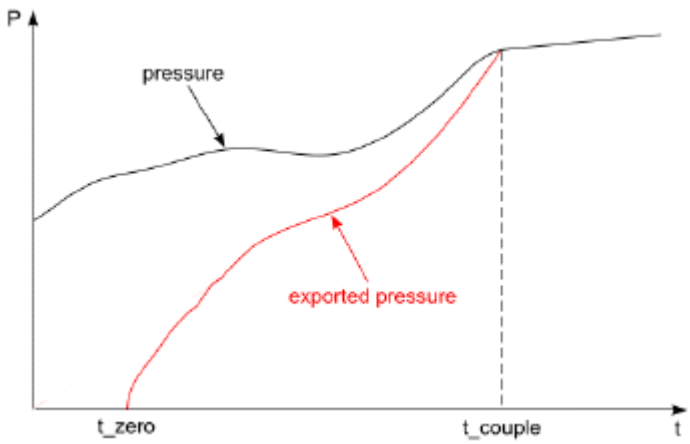
Pressure Ramping Parameters

Progressively increases the pressure component of the traction load that is exported to Abaqus. Ramping the pressure loads is particularly useful at the start of a co-simulation, where the structure is initially unloaded and an initial flow field in the fluid was calculated in a standalone analysis with a rigid structure.

The pressure ramping parameters are specified as `[t_zero, t_couple]`. During $t < \mathbf{t_zero}$, the STAR-CCM+ exported pressure and shear stress are zero. Between $\mathbf{t_zero} < t < \mathbf{t_couple}$ the exported

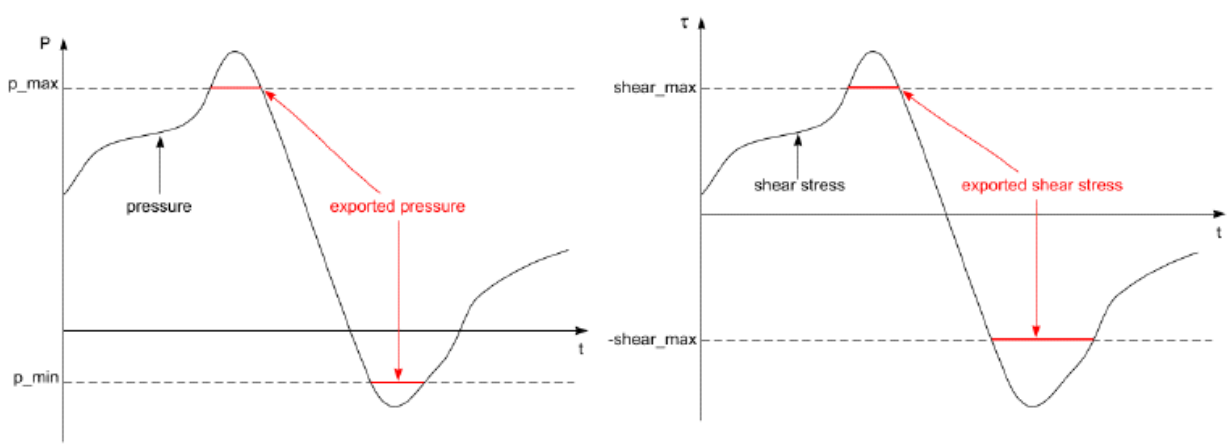
pressure value is smoothly blended so that by $t \geq$ **t_couple** the exported pressure equals the actual pressure STAR-CCM+ computed. In the same period, the exported shear stress vector is the same direction as the shear stress vector STAR-CCM+ computed, but the magnitude is smoothly blended. Similar to the exported pressure, at $t \geq$ **t_couple** the exported shear stress equals the shear stress STAR-CCM+ computed.

If you set *Pressure Ramping Time Scale* to **Co-Simulation Time**, set **t_zero** to `0.0`. If you choose the **STAR-CCM+ Time**, set **t_zero** to the physical time where the pressure is zero.



Traction Clipping Parameters

Allows you to limit the traction loads in the fluid domain to reflect the actual operating conditions. The traction clipping parameters are specified as `[p_max, p_min, shear_max]`, where **p_max** and **p_min** are the maximum and minimum values for the exported pressures, and **shear_max** is the maximum shear stress magnitude. These values are constant throughout the co-simulation. When setting the pressure parameters **p_max** must be greater than **p_min** and both values can be negative. For **shear_max**, the direction of the vector is kept the same and just the magnitude is clipped, as shown in the following.



An example of when these limiting values are useful is in simulating multiphase flow of gas and liquid in a pressurized tank. Consider a case where the known peak operating pressure in the tank is 70 bar. Rather

than export a computed pressure that is potentially unrealistic or can result in an unstable co-simulation (especially in the early stages of the co-simulation), you can enforce the peak operating pressure (**p_max**).

Apply Traction Field Multiplier

When activated, STAR-CCM+ applies the traction field multiplier that you specify for the *Traction Field Multiplier* property.

Traction Field Multiplier

Specifies a scalar field function to be multiplied with the traction field before export to Abaqus.

A *Traction Field Multiplier* can be specified to define a spatially and/or time varying function to modify the traction exported to Abaqus. The multiplier is applied before the *Pressure Ramping Parameters* or *Traction Clipping Parameters* are applied. An existing or user-defined scalar field function can be used as the multiplier. Some uses of a multiplier include applying a factor of safety to the load in a one-way coupled analysis or masking pressures in regions deemed to be undergoing contact.

STAR-CCM+ Properties and Equivalent Abaqus Keywords

The following table summarizes the STAR-CCM+ field functions and equivalent Abaqus keywords for defining the physical quantities that are transferred between the two codes.

STAR-CCM+		Abaqus
Property	**Value**	**Keyword**
Exported Field Functions	**Pressure and Wall Shear Stress**	`*CO-SIMULATION REGION, TYPE=SURFACE, IMPORT <Abaqus surface>, CF` (or `CM`)
	Boundary Heat Flux	`*CO-SIMULATION REGION, TYPE=SURFACE, IMPORT <Abaqus surface>, CFL`
	Boundary Heat Flux and either Specified Y+ Heat Transfer Reference Temperature or Local Heat Transfer Reference Temperature	`*CO-SIMULATION REGION, TYPE=SURFACE, IMPORT <Abaqus surface>, CFILM`
Imported Field Functions	**Co-Simulation: Imported Displacement**	`*CO-SIMULATION REGION, TYPE=SURFACE, EXPORT <Abaqus surface>, U` (or `UR`)
	Co-Simulation: Imported Velocity	`*CO-SIMULATION REGION, TYPE=SURFACE, EXPORT <Abaqus surface>, V` (or `VR`)
	Co-Simulation: Imported Temperature	`*CO-SIMULATION REGION, TYPE=SURFACE, EXPORT <Abaqus surface>, NT`
Coupled Boundaries	**STAR-CCM+ boundary**	`Specify the co-located Abaqus surface in the *CO-SIMULATION REGION keyword (as above).`

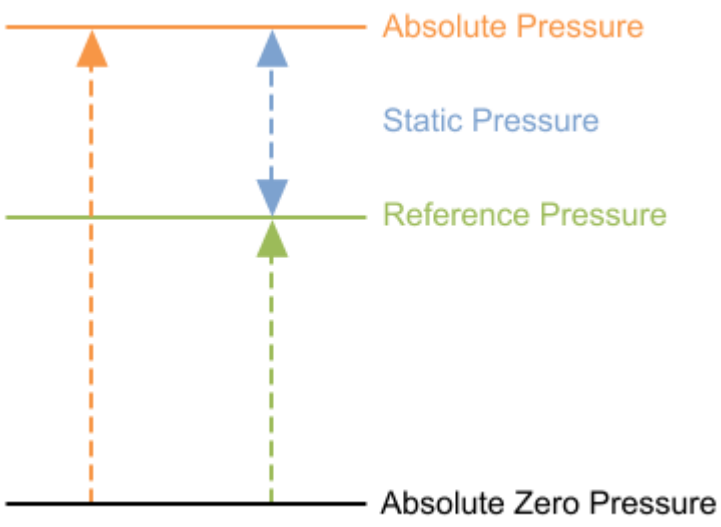
Additional Considerations

This chapter includes more technical information for performing co-simulations using Abaqus and STAR-CCM+.

- Exporting pressure loads
- Model dimensions and coordinate systems
- Co-location of the Abaqus and STAR-CCM+ models
- Supported Abaqus elements
- Supported STAR-CCM+ cells
- Known limitations

Exporting Pressure Loads

For co-simulation, export the pressure load on a wall boundary using one of two field functions: `Static Pressure` or `Absolute Pressure`. The `Absolute Pressure` in STAR-CCM+ is the absolute static pressure, which is the sum of the static pressure and the reference pressure. The `Static Pressure` is equal to the absolute pressure minus the reference pressure, and is often termed the "gauge" pressure in literature. The following diagram illustrates the relationship between these values.



Note: Use consistent applications of pressure in both Abaqus and STAR-CCM+. In Abaqus, the reference pressure is the pressure at which the structure is considered to be unloaded, with no displacements. It is the variation of the absolute static pressure with respect to the reference pressure that causes deformations and strains in the structure.

As an example of how to ensure consistency, consider a cylinder that is loaded on the inside diameter (ID) and the outside diameter (OD). STAR-CCM+ models the fluid on the inside of the cylinder. Assume that the OD of the cylinder is subject to 1 bar (atmospheric conditions), and that the STAR-CCM+ reference pressure

is 1 bar (which is the default value). If STAR-CCM+ is set to export the `Static Pressure` as the load on the ID, then, in Abaqus, load the OD of the cylinder with zero load to be consistent. In effect, this means that no load is required in Abaqus at the OD. If, however, STAR-CCM+ is set to export the `Absolute Pressure` as the load on the ID of the cylinder, then, in Abaqus, load the OD with 1 bar (that is, the same value as the reference pressure in STAR-CCM+).

If the absolute static pressure within the cylinder is 10 bar when the fluid velocity is zero, it is convenient for the STAR-CCM+ reference pressure to be set to 10 bar. With the 10-bar reference pressure, if you export `Static Pressure` as the load on the ID, in Abaqus the OD must be loaded with a pressure of -9 bar (that is, a tension load) to be consistent. If, however, you choose to export `Absolute Pressure` from STAR-CCM+, load the OD in the Abaqus with a pressure of 1 bar.

This procedure of loading the OD of the cylinder with the pressure that will give the correct net loading on the structure is correct only for elements where through-thickness behavior is not considered (for example, shell elements). If solid elements are used to model the cylinder, then applying fictitious loads to the OD will produce the correct bulk loading, but incorrect behavior in the walls of the cylinder. In this case, absolute pressure should be exported to Abaqus for the ID and the ambient pressure should be applied on the OD within the Abaqus model.

Model Dimensions and Coordinate Systems

Co-simulation between two-dimensional and axisymmetric models is not supported. Represent two-dimensional models as a three-dimensional slice of unit thickness, or wedge element, with the appropriate symmetry boundary conditions applied.

Vector fields are transferred according to Abaqus and STAR-CCM+ conventions. The first component represents the quantity along the x-axis, the second component represents the quantity along the y-axis, and the third component represents the quantity along the z-axis. All fields are transferred in the global coordinate system (the Laboratory Coordinate System in STAR-CCM+). If a local coordinate system is applied in the Abaqus model, the incoming and outgoing fields are transformed appropriately.

Co-Location of the Abaqus and STAR-CCM+ Models

Co-locate the Abaqus and STAR-CCM+ models. Place the fluid-structure interface regions of both models in the same geometric positions in space. Ensure that the coupled wall boundaries in STAR-CCM+ are not larger or smaller in extent than the corresponding interface in the Abaqus model.

Supported Abaqus Elements

Abaqus supports first-order and second-order continuum, shell, and membrane elements. Modified tetrahedral elements are not supported due to mapping limitations.

Supported STAR-CCM+ Cells

STAR-CCM+ supports tetrahedral and trimmed mesh cell types, as well as cells of arbitrary polyhedral type in 3D for co-simulation with Abaqus.

Known Limitations

This section lists the known limitations of co-simulation using Abaqus and STAR-CCM+.

- Run Abaqus and STAR-CCM+ on binary compatible platforms.

- The coupled boundaries must be of the same spatial extents - the coupled boundaries cannot be larger or smaller in extent than their counterpart in either code.
- If coupling both sides of a double-sided structure, a separate surface must be created for each side of the structure and assigned to a separate coupling zone in the fluid domain.

GT-SUITE Co-Simulation

STAR-CCM+ allows for two-way co-simulation with GT-SUITE, a multiphysics simulation tool from Gamma Technologies. GT-SUITE provides a modeling framework for the design and analysis of engineering systems, such as engines, vehicles, and thermal-hydraulic networks.

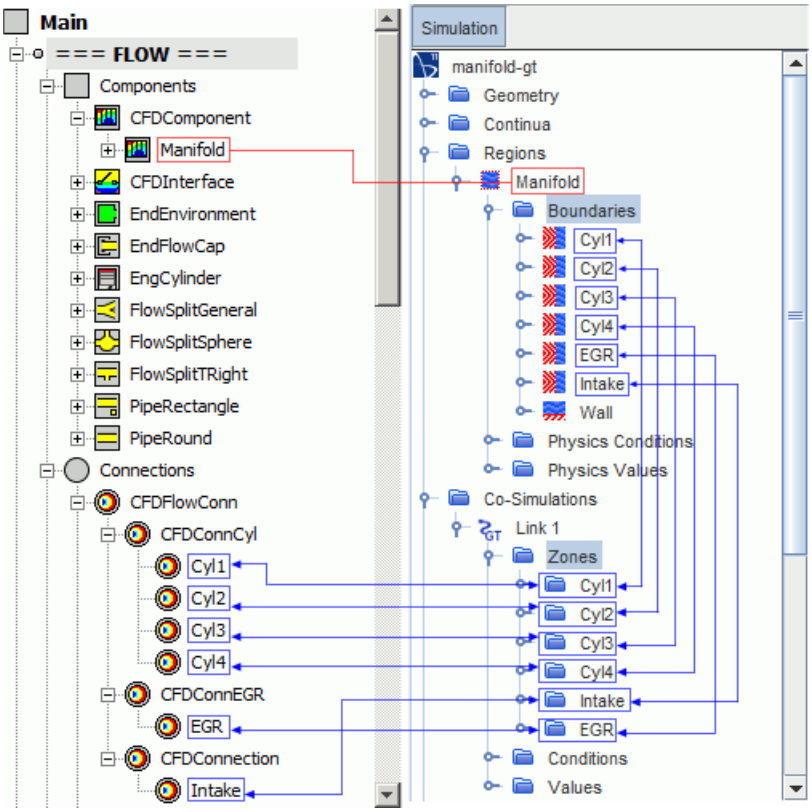
During the co-simulation, GT-SUITE drives the overall one-dimensional system simulation, while STAR-CCM+ models the flow within components that require a three-dimensional analysis. For example, you can use STAR-CCM+ to model an intake manifold that forms part of an engine analysis designed in GT-SUITE.

Co-Simulation Interface

In GT-SUITE, you define the co-simulation interface using the following objects:

- **CFD Component**—represents the component that is modeled in 3D using STAR-CCM+.
- **CFD Interfaces**—mark the 1D/3D interfaces between GT-SUITE and STAR-CCM+.
- **CFD Connections**—define the physical conditions and values at the 1D/3D interfaces.

In STAR-CCM+, you couple the relevant boundaries with GT-SUITE using co-simulation zones. Each co-simulation zone uniquely associates a GT-SUITE CFD connection with a STAR-CCM+ boundary.



Co-simulation zones and CFD connections must match in number and name.

Exchanged Fields

Co-simulation supports coupling between CFD connections (either **CFDFlowConn** or **CFDFlowConnBeta** connections) and STAR-CCM+ boundaries of type **Mass Flow Inlet**, **Velocity Inlet**, or **Pressure Outlet**.

The fields that are exchanged between the CFD connection and the coupled STAR-CCM+ boundary depend on:

- the STAR-CCM+ boundary type
- the GT-SUITE CFD Interface Condition that you define for the CFD connection

STAR-CCM+ Boundary Type	GT-SUITE CFD Connection	GT-SUITE CFD Interface Condition Imposed by GT	Imported from GT-SUITE	Exported to GT-SUITE
Velocity Inlet	**CFDFlowConn**	Flow (velocity or mass flow rate)	Velocity	Static Pressure
	CFDFlowConnBeta	Flow (CFD imposes GT pressure): Velocity		
Mass Flow Inlet	**CFDFlowConn**	Flow (velocity or mass flow rate)	Velocity Density	Static Pressure

STAR-CCM+ Boundary Type	GT-SUITE CFD Connection	GT-SUITE CFD Interface Condition Imposed by GT	Imported from GT-SUITE	Exported to GT-SUITE
	CFDFlowConnBeta	Flow (CFD imposes GT pressure): Velocity		
		Flow (CFD imposes GT pressure): Mass Flow Rate	Mass	
Pressure Outlet	**CFDFlowConn**	Pressure	Depending on the flow direction, either: • Static Pressure, when flow is leaving the STAR-CCM+ boundary • Total Pressure, when flow is entering the boundary	Velocity
	CFDFlowConnBeta	Pressure (CFD imposes GT velocity)		
		Pressure (CFD imposes GT mass flow)		Mass Flow Rate (via file)

At a **Mass Flow Inlet** boundary, STAR-CCM+ applies the mass flow rate that is calculated from the imported fields (either velocity and density, or mass).

Depending on the flow direction, STAR-CCM+ also imports or exports the following fields, regardless of the type of coupled boundary and CFD connection:

Flow Direction	Imported from GT-SUITE	Exported to GT-SUITE
Entering the STAR-CCM+ boundary	Temperature Mass Fractions	--
Leaving the STAR-CCM+ boundary	--	Temperature Mass Fractions Passive Scalars

At a **Mass Flow Inlet** boundary, STAR-CCM+ applies the total temperature that is calculated from the imported static temperature and velocity. At a **Velocity Inlet** boundary, STAR-CCM+ directly applies the static temperature imported from GT-SUITE.

GT-SUITE and STAR-CCM+ do not exchange turbulence values. Turbulence values are specified in STAR-CCM+ at the boundary level, in the usual manner.

Before sending data back to GT-SUITE, STAR-CCM+ averages the computed fields across sampling volumes adjacent to the coupled boundaries, to convert the three-dimensional fields to one-dimensional values. Volume averaging is important when significant three-dimensional effects occur near the coupled boundary. The co-simulation provides optimum results when the flow in the sampling volumes is approximately one-dimensional.

GT-SUITE co-simulation uses an explicit coupling scheme, that is, data exchange occurs once per time-step.

Multi-Component Gases: Coupled Material Components

In co-simulation with GT-SUITE you can model single-component gases and liquids, and multi-component gases. When you model multi-component gases, GT-SUITE defines the components of the fluid mixture in the co-simulation. These components include unburned species and burned species, which result from the products of reaction between the unburned species. The combustion products consist of 13 components with varying composition:

prod_ar: Argon
prod_co: Carbon Monoxide
prod_co2: Carbon Dioxide
prod_h: Monatomic Hydrogen
prod_h2: Hydrogen
prod_h2o: Water
prod_n: Monatomic Nitrogen
prod_n2: Nitrogen
prod_no: Nitric Oxide
prod_o: Monatomic Oxygen
prod_o2: Oxygen
prod_oh: Hydroxide
prod_so2: Sulphur Dioxide

You associate the GT-SUITE species with corresponding mixture components or passive scalars in STAR-CCM+. Mixture components represent GT-SUITE species that influence the bulk flow, that is, species that have non-trace quantities. Passive scalars represent GT-SUITE species that do not influence the bulk flow, that is, species that have trace quantities.

The species mass fractions are exchanged at the co-simulation interface, as described in the previous section.

GT-SUITE Coupling Libraries

In order to communicate with GT-SUITE, STAR-CCM+ requires the location of the GT-SUITE coupling library. The GT-SUITE installation provides library files for exchanging data using either single or double precision:

Platform	Precision	Library File Location
Linux 32-bit	Single	`[GTIHOME]/[GT-SUITE_version]/GTsuite/bin/linux_x86/coupling/libgtlink.so`
	Double	`[GTIHOME]/[GT-SUITE_version]/GTsuite/bin/linux_x86/coupling/libgtlink_dp.so`
Linux 64-bit	Single	`[GTIHOME]/[GT-SUITE_version]/GTsuite/bin/linux_x86_64/coupling/libgtlink.so`
	Double	`[GTIHOME]/[GT-SUITE_version]/GTsuite/bin/linux_x86_64/coupling/libgtlink_dp.so`

Platform	Precision	Library File Location
Windows 32-bit	Single	`[GTIHOME]/[GT-SUITE_version]/GTsuite/bin/win32/coupling/gtlink.dll`
	Double	`[GTIHOME]/[GT-SUITE_version]/GTsuite/bin/win32/coupling/gtlink_dp.dll`
Windows 64-bit	Single	`[GTIHOME]/[GT-SUITE_version]/GTsuite/bin/win64/coupling/gtlink.dll`
	Double	`[GTIHOME]/[GT-SUITE_version]/GTsuite/bin/win64/coupling/gtlink_dp.dll`

where `[GTIHOME]` is the path to the GT-SUITE installation and `[GT-SUITE_version]` is the version of GT-SUITE.

Contents:

Preparing the GT-SUITE Simulation for Co-Simulation
Preparing the STAR-CCM+ Simulation for Co-Simulation
Coupling to GT-SUITE
Setting Up STAR-CCM+ and GT-SUITE for a Restart Run
GT-SUITE Co-Simulation Model Reference

Preparing the GT-SUITE Simulation for Co-Simulation

Starting from a fully set up GT-SUITE simulation, you define the co-simulation interface by adding CFD objects to the model map.

The following instructions outline the general workflow and may vary based on the version of GT-SUITE that you are running.

- For a step-by-step example using GT-SUITE v2017, refer to the tutorial GT-SUITE Co-Simulation: 1D Coupling.
- For more information on setting up a simulation in GT-SUITE, refer to the GT-SUITE User Guide.

Before you define co-simulation settings:

1. Launch GT-ISE and load the GT-SUITE simulation that you want to couple with STAR-CCM+.
2. Run the standalone GT-SUITE analysis until converged.

   This practice allows you to:

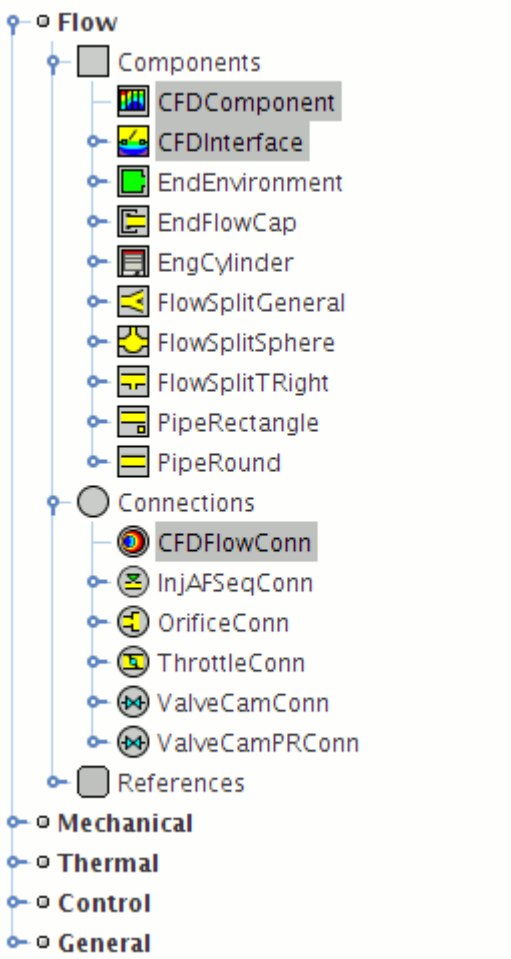
   - Check that the GT-SUITE simulation is set up correctly.
   - Determine the number of cycles that are required to achieve convergence.
   - Use the results from the standalone GT-SUITE analysis as initial conditions in STAR-CCM+, in order to speed up convergence of the coupled problem.
3. Analyze the results in GT-POST, making note of the average values for pressure, temperature, and, when modeling multi-component gases, species mass fractions.

To prepare the simulation for coupling, you add CFD objects to the GT-SUITE model. CFD objects allow you to define the interfaces with STAR-CCM+ and to control the co-simulation settings. To create CFD objects, you require the corresponding templates:

4. In the *Templates and Search* ribbon, click **Template Library**.

5. In the *Template Library* tree, expand the **Flow** > **General Flow** node.
6. Drag and drop the following templates from the template library tree onto the project tree:
   - **Components** > **CFDComponent**
   - **Components** > **CFDInterface**
   - either:
     - **Connections** > **CFDFlowConn**
     - **Connections** > **CFDFlowConnBeta**

       The **CFDFlowConnBeta** object exposes additional controls for setting pressure relaxation and tracking mass continuity at the interface.



Contents:

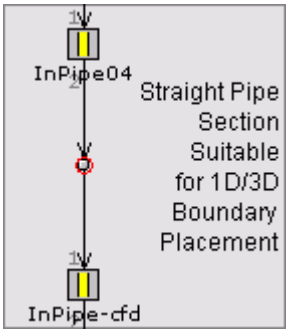
Defining CFD Objects
Defining a CFD Species Map in GT-SUITE

Defining CFD Objects

Using the CFD templates, add the required CFD objects to the project tree.

Add CFD interfaces to the model map, to represent the STAR-CCM+ boundaries:

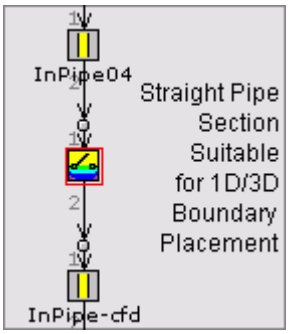
1. In the model map, locate the components of type **'Pipe*'** that you want to couple with STAR-CCM+.
2. For each pipe section, replace the default 1D orifices with 1D/3D **CFDInterface** objects using the following procedure:
   a) Delete the default 1D orifices.



b) Drag and drop the **Components** > **CFDInterface** > **def** node from the project tree onto the model map, then place it at the original orifice position.

c) Right-click the CFDInterface on the model map and select **Start New Link**.

d) Use the **Start New Link** tool to join the CFDInterface to the adjacent pipes, making sure that the arrows point in the direction of the flow.



CFD interfaces serve as 1D/3D switches. When GT-SUITE and STAR-CCM+ are uncoupled, GT-SUITE solves the flow in the one-dimensional components. When coupled, the GT-SUITE solver supplies boundary conditions to STAR-CCM+ and no longer solves the flow in the one-dimensional components.

To define the characteristics of the 1D/3D interfaces, add CFD connections to the model map. To create a CFD connection:

3. In the project tree, right-click either the **Connections** > **CFDFlowConn** node or the **Connections** > **CFDFlowConnBeta** node and select **Add Object...**

   CFDFlowConnBeta connections allow you to control pressure relaxation and mass continuity at the interface.

4. Define the properties of the CFD connection as follows:

Property/Attribute	Setting
CFD Domain Sampling Length	Specify the length of the volume-averaging zone, across which STAR-CCM+ averages the three-dimensional fields, before sending values to GT-SUITE. The specified length is rounded up or down to cover the nearest complete mesh cell.

Property/Attribute	Setting
	Make sure that the sampling volume length covers an area of the simulation where there is one-dimensional flow. You can visualize the sampling volumes in STAR-CCM+ using the **Average Zone Flag** field function in a scalar scene. See Preparing the STAR-CCM+ Simulation for Co-Simulation.
CFD Boundary Cross-section (be precise)	Specify the cross-section of the coupled STAR-CCM+ boundary. In general, GT-SUITE allows for pipes with circular or rectangular cross-sections. For cross-sections that do not meet these requirements, calculate the effective diameter in STAR-CCM+ by using a **Sum** report of the *Area* field function at the coupled boundary.
Species Map Override	**def**

5. Define the remaining properties based on the type of connection:
   - For a **CFDFlowConn** object, set *Interface Conditions Imposed by GT at the CFD Boundary* to one of the following:
     - To couple with a STAR-CCM+ velocity inlet or mass flow inlet, select **Flow (velocity or mass flow rate)**
     - To couple with a STAR-CCM+ pressure outlet, select **Pressure**

     This choice determines the fields exchanged during the co-simulation. See Boundary Types and Exchanged Fields.
   - For a **CFDFlowConnBeta** object, set *CFD Interface Condition Imposed by GT:* to one of the following:
     - To couple with a STAR-CCM+ velocity inlet, select **Flow (CFD imposes GT pressure): Velocity**.
     - To couple with a STAR-CCM+ mass flow inlet, select either:
       - **Flow (CFD imposes GT pressure): Velocity**
       - **Flow (CFD imposes GT pressure): Mass Flow Rate**
     - To couple with a STAR-CCM+ pressure outlet, select either:
       - **Pressure (CFD imposes GT velocity)**
       - **Pressure (CFD imposes GT mass flow)**—when you choose this option, GT-SUITE must import the mass flow rate data through a file. For instructions, see Importing Mass Flow Rate from a File.

     This choice determines the fields exchanged during the co-simulation. For more information, see Boundary Types and Exchanged Fields.
6. Define any remaining properties as appropriate to your case.

Add a CFD component to the model map to represent the STAR-CCM+ region:

7. In the project tree, right-click the **Components** > **CFDComponent** node and select **Add Object...**
8. In the *Create Object* dialog:

a) Set *Object Name* to the name of the coupled STAR-CCM+ region.

b) In the *Simulation Phases* tab, set *Engine Cycles or Time Before CFD (or "cfdreload") Started* to the number of pre-cycles that GT-SUITE runs independently before coupling with STAR-CCM+.

Pre-cycles improve stability in GT-SUITE, providing more accurate boundary conditions from GT-SUITE and therefore reducing the time to reach a converged solution.

9. Complete the set up by defining the GT-SUITE species. See Defining a CFD Species Map in GT-SUITE.

Contents:

Importing Mass Flow Rate from a File

Importing Mass Flow Rate from a File

When you couple a **CFDFlowConnBeta** connection of type **Pressure (CFD imposes GT mass flow)** with a STAR-CCM+ pressure outlet, STAR-CCM+ sends mass flow rate data using a file.

To import mass flow rate from STAR-CCM+:

1. Instruct STAR-CCM+ to export mass flow rate to a file at each time-step (for instructions, see Exporting Mass Flow Rate to a File).
2. In the *Templates and Search* ribbon, click **Template Library**.

3. In the *Template Library* tree, expand the **General** > **External Model Links** > **References** node.
4. Drag and drop the **CoSimInputFile** node onto the project tree.
5. In the project tree, right-click the **General** > **References** > **CoSimInputFile** node and select **Add Object...**
6. In the *Template: CoSimInputFile* dialog:

   a) Specify the *Object Name*.

   b) Click the *File* tab and:

      1. Set *File Name* to `[Plot Name]_[Base Filename]_*.csv`, where `[Plot Name]` and `[Base Filename]` are defined in STAR-CCM+. See Exporting Mass Flow Rate to a File.
      2. Set *Input File Directory* to the directory where STAR-CCM+ exports the mass flow rate files. See Exporting Mass Flow Rate to a File.

   c) Click the *Read Specification* tab and activate **Bottom #Lines to Shift Block Up**. Make sure that the corresponding *Object Value* is set to zero.

   d) For the attribute *Block Line 1*:

      1. Set *Word #1* to **ignore/ign**.
      2. Set *Word #2* to `1`.

7. In the project tree, right-click the relevant **CFDFlowConnBeta** connection node and select **Edit Properties...**
8. In the dialog, set the following properties:

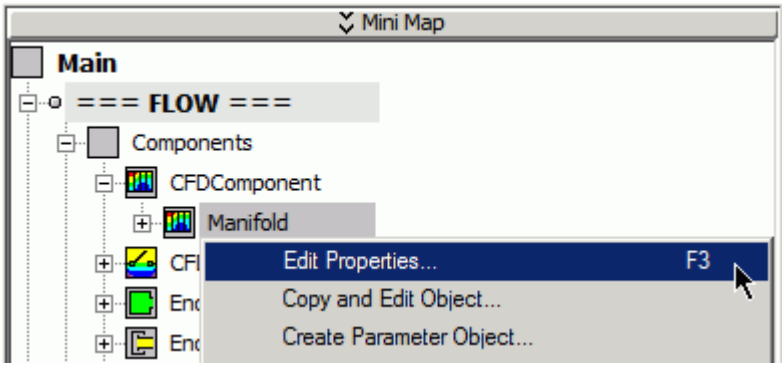
Property	Setting
CFD Interface Condition Imposed by GT:	**Pressure (CFD imposes GT mass flow)**
File Containing CFD Species Flow Rates (timestep updated)	Select the **CoSimInputFile** object that you just created.

Defining a CFD Species Map in GT-SUITE

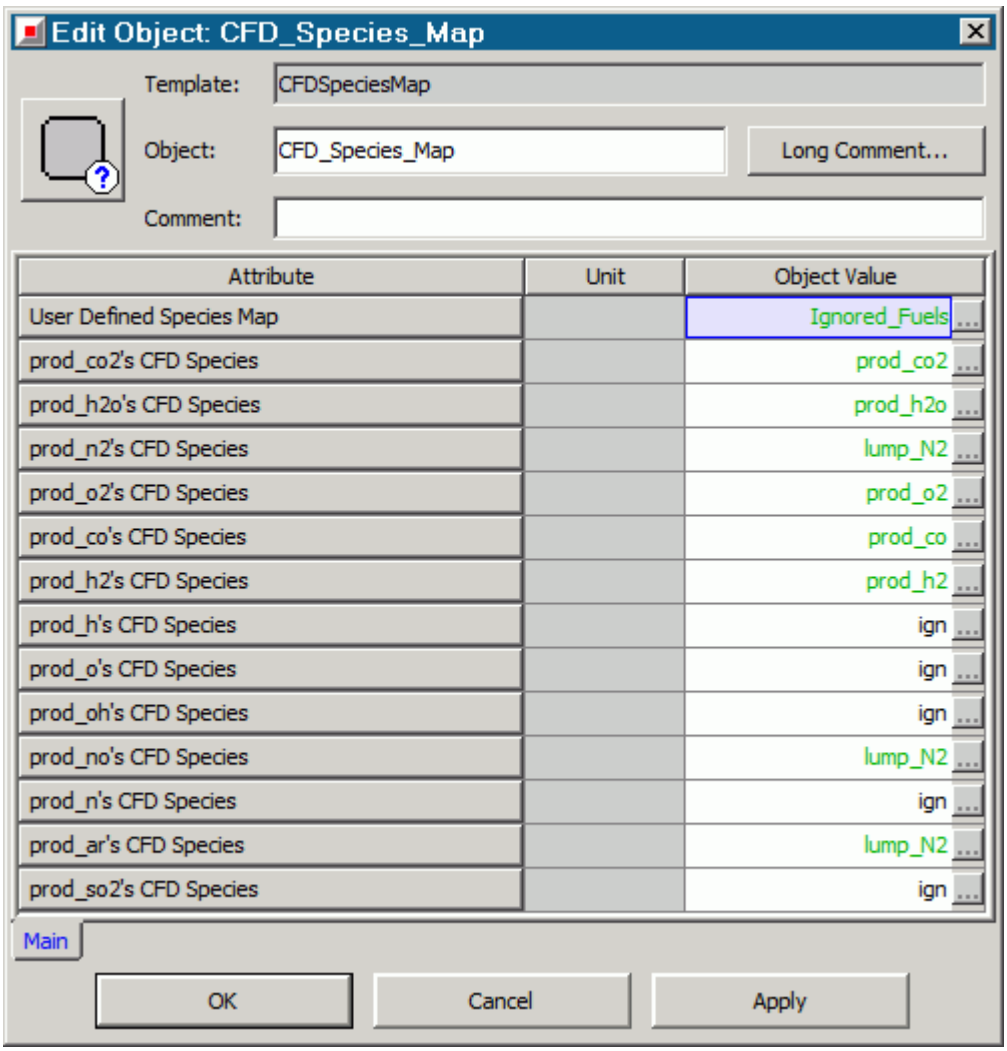
CFD species maps determine how the gas components are transferred between STAR-CCM+ and GT-SUITE.

To define a CFD species map:

1. Right-click a CFD component, and select **Edit Properties...**



2. In the *Edit Object* dialog, click the ellipsis (...) next to the *Species Mapping Object* attribute.
3. In the *Value Selector* dialog, select **CFDSpeciesMap** from the *Create New Object* group box and click **OK**.
4. In the *Create Object* dialog, set *Object* to the name of the species map.
5. Set *User Defined Species Map* to one of the following options:
   - `def` — Uniquely associates all GT-SUITE species with a STAR-CCM+ gas component or passive scalar.
   - **CFDDefinedSpeciesMap** — The user-defined **CFDDefinedSpeciesMap** reference object defines the association of one or more GT-SUITE species with a STAR-CCM+ gas component or passive scalar. For example, you can map all harmful gases to a STAR-CCM+ passive scalar called Pollutants.
6. Set each CFD species to one of the following options:
   - `ign` — Ignores the CFD species and does not send it to STAR-CCM+. The mass fraction of this species is redistributed between the other species to maintain mass conservation. This option is only recommended for trace components.
   - **CFDSpecies** — Associates the GT-SUITE species with the user-defined **CFDSpecies** in STAR-CCM+.



Contents:

Defining a CFDDefinedSpeciesMap
Defining a CFDSpecies

Defining a CFDDefinedSpeciesMap

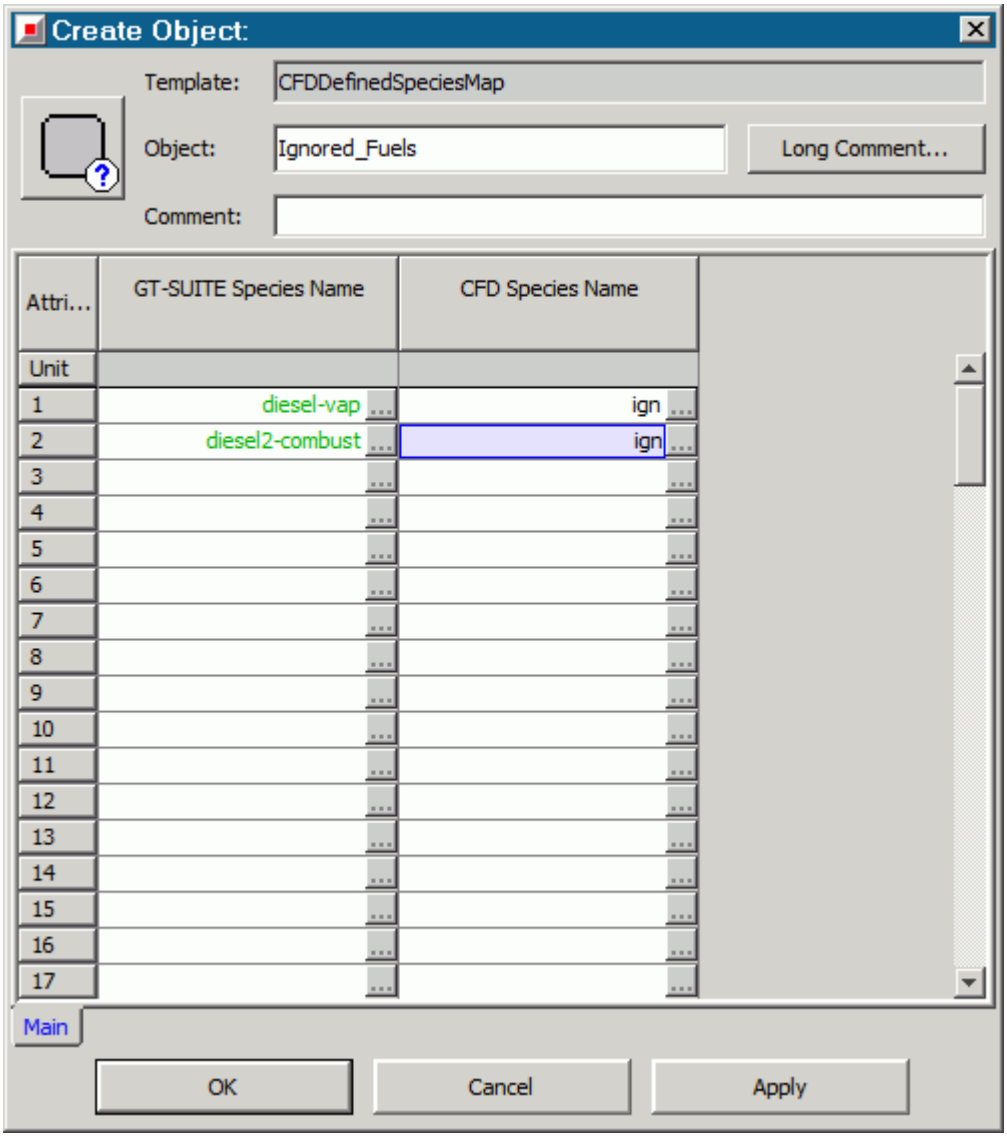
A CFDDefinedSpeciesMap is similar to a CFDSpeciesMap, but you can map multiple GT-SUITE species to a single STAR-CCM+ species. You can also define the mapping for the intake charge species.

The following steps assume that you are defining a CFD species map, as detailed in the previous section.

To define a CFDDefinedSpeciesMap:

1. In the *Edit Object* dialog of the CFDSpeciesMap, click the ellipsis (...) next to the *User Defined Species Map* attribute.
2. In the *Value Selector* dialog, select CFDDefinedSpeciesMap from the *Create New Object* group box and click **OK**.

3. In the *Create Object* dialog, set *Object* to the name of the CFDDefinedSpeciesMap.
4. Set each entry in the *GT-SUITE Species Name* column to the GT-SUITE species that you want to include in the map.
5. Set each entry in the *CFD Species Name* column to one of the following options:
   - `ign` — Ignores the CFD species and does not send it to STAR-CCM+. The mass fraction of this species is redistributed between the other species to maintain mass conservation. This option is only recommended for trace components or species that are not used in the GT-SUITE case that you are analyzing.
   - **CFDSpecies** — Associates the GT-SUITE species with the user-defined CFDSpecies in STAR-CCM+. See Defining a CFDSpecies.



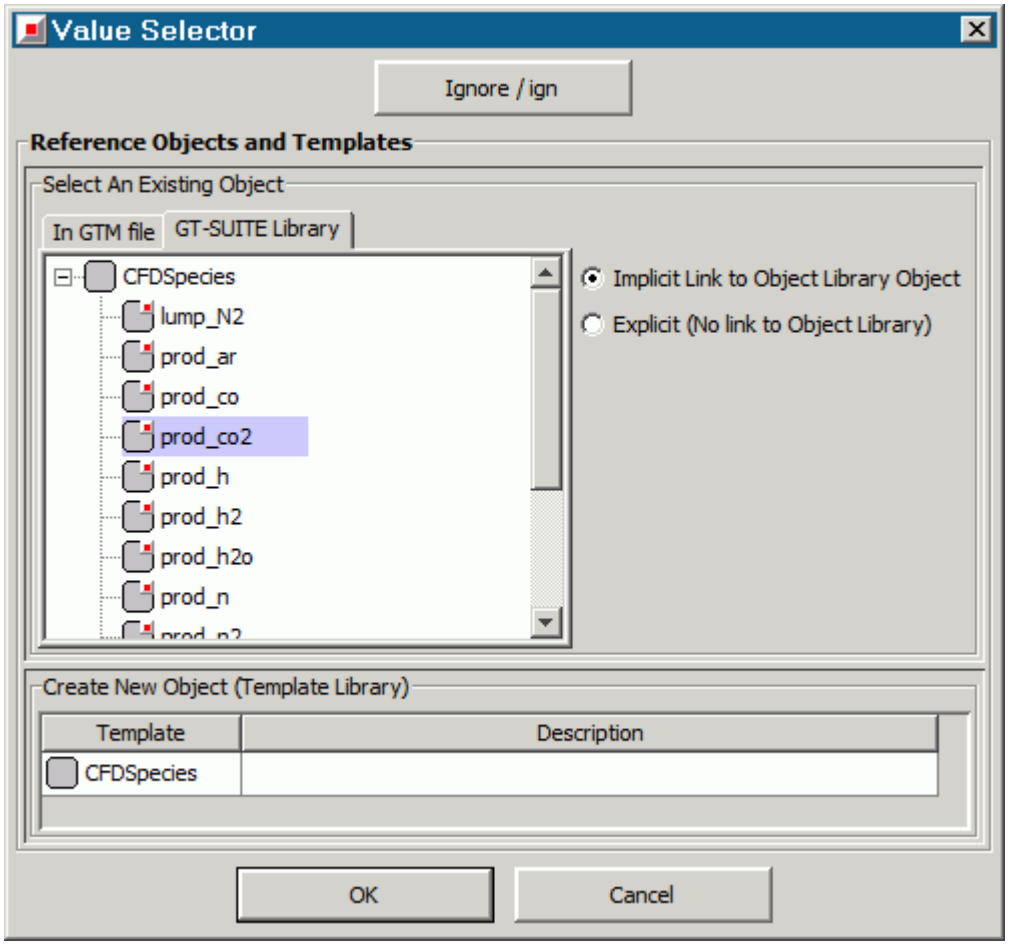
Defining a CFDSpecies

A CFDSpecies is the species that GT-SUITE passed to STAR-CCM+ during a coupled simulation. GT-SUITE writes the material properties for these species when you run the simulation preprocessing.

The following steps assume that you are defining a CFD species map, as detailed in the previous section.

To define a CFDSpecies:

1. In the *Edit Object* dialog of the CFDSpeciesMap, click the ellipsis (...) next to a CFD species.
2. In the *Value Selector*, select the relevant species from the *GT-SUITE Library* tab and click **OK**.



Preparing the STAR-CCM+ Simulation for Co-Simulation

Set up a STAR-CCM+ simulation for co-simulation with GT-SUITE.

1. Activate the following physics models on the relevant fluid continuum:

Group	Physics Model
Space	**Three Dimensional**
Time	**Implicit Unsteady**
Material	**Gas**, **Liquid**, or **Multi-Component Gas** (**Non-Reacting**)
Flow	**Coupled Flow** or **Segregated Flow**
Enabled Models	**Gradients** (selected automatically)
Equation of State	**Ideal Gas**
Energy	any
Viscous Regime	any
Optional Models	**Co-Simulation**
Co-Simulation Models	**GT-SUITE**
Optional Models	**Passive Scalar** **Note:** This model can be neglected if you represent all species of a multi-component gas with gas components.

   You can select further models from the *Optional Models* group to meet individual simulation requirements.

Simulation preprocessing in GT-SUITE generates a material database file, `<GT-POWER_case_name>.chemkin_GT.dbs`. To use the same material properties as GT-SUITE, import this file into STAR-CCM+:

2. Right-click the **Tools** > **Material Databases** node and select **New Material Database**.
3. Right-click the new material database and select **Import Properties**.
4. In the *Open* dialog, select the `<GT-POWER_case_name>.chemkin_GT.dbs` file and click **OK**.
5. In the *Import Properties* dialog, select **Import properties for all materials** and click **OK**.

When modeling multi-component gases, define a mixture component or passive scalar for each species defined in GT-SUITE. In general, use gas components to represent GT-SUITE species that have non-trace quantities, and passive scalars to represent GT-SUITE species that have trace quantities. A mixture component influences the physical properties of the simulation, thus providing the most accurate and realistic solution. A passive scalar has no effect on the physical properties of the simulation, which reduces processing times. If you use passive scalars for all species, specify a background fluid in the region.

To add a mixture component to the simulation:

6. Expand the **[physics continuum]** > **Models** node.
7. Right-click the **Multi-Component Gas** > **Gas Components** node and select **Select Mixture Components...**
8. In the *Select Mixture Components* dialog, select gas components from the **GT-POWER Materials** database, then click **Apply**.

Note: Although the “fuel-combust” species is a liquid, you can represent it with a gas in the **Multi-Component Gas** model. This modeling approach does not affect results, as the gas only represents the species in STAR-CCM+, with the physics values received from GT-SUITE.

To add a passive scalar to the simulation:

9. Right-click the **Passive Scalars** node and select **New**.

To reduce the numerical round-off, set the reference pressure to the average pressure in GT-SUITE. You can identify the average pressure in GT-SUITE by running a standalone GT-SUITE analysis and using GT-POST to analyze the results.

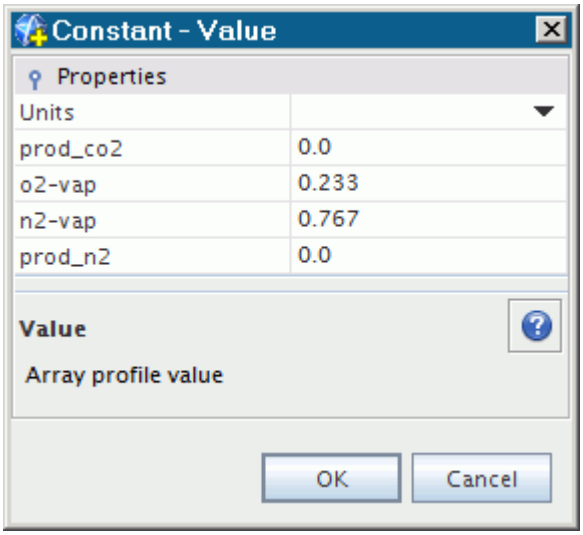
To set the reference pressure:

10. Select the **[physics continuum]** > **Reference Values** > **Reference Pressure** node and set the *Value* property.

Note: All pressures in STAR-CCM+ are defined relative to the reference pressure. In GT-SUITE, all pressures are absolute. Take these differences into consideration when defining pressure values.

For best initialization, set the STAR-CCM+ initial conditions to the average values from GT-SUITE, which you can obtain by running a standalone GT-SUITE analysis and using GT-POST to analyze the results:

11. Expand the **[physics continuum]** > **Initial Conditions** node, and specify the initial conditions as follows:
    a) As the initial pressure in STAR-CCM+ is relative to the reference pressure, set the **Pressure** value to the difference between the average pressure from GT-SUITE and the STAR-CCM+ reference pressure.
    b) As the initial temperature in STAR-CCM+ is an absolute value, set the **Static Temperature** value to be the same as the average temperature from GT-SUITE.
    c) When you model multi-component gases, set the species mass fractions to the average values from GT-SUITE, by setting the **Species Mass Fraction** value. Click ... (**Custom Editor**) to enter the mass fractions in an array.



Specify boundary types in the usual manner. The coupled boundary types determine the fields exchanged between GT-SUITE and STAR-CCM+ during the co-simulation, as explained in Exchanged Fields.

12. Specify the boundary types. For the boundary that couple with GT-SUITE, set the *Type* to either **Mass Flow Inlet**, **Velocity Inlet**, or **Pressure Outlet**.

    The recommended practice is to set the boundary type to mass flow inlet for all coupled boundaries—regardless of whether they are inlets or outlets. If you use a mass flow inlet, STAR-CCM+ uses the boundary area, normal velocity, and density from GT-SUITE to determine the mass flow rate at the boundary. Small changes in density are reflected in the mass flow rate across the boundary.

    For coupled boundaries that have large negative flows (such as flows that are leaving the STAR-CCM+ domain), use coupled velocity inlet boundaries. Using coupled mass flow inlet boundaries can sometimes lead to instabilities for some models. If you use a velocity inlet, STAR-CCM+ takes the normal velocity from GT-SUITE and applies it uniformly at the coupled boundaries.

    Note: Setting all the STAR-CCM+ coupled boundaries to mass flow or velocity inlets does not have the same negative numerical repercussions that would occur in a standalone CFD model. The absolute pressure of the CFD model drives the GT-SUITE solution and sets the mass flux or velocity at the boundary.

13. Specify boundary conditions in the usual manner. On the coupled boundaries, only specify turbulence settings, as all the other values are set by GT-SUITE.
14. Complete the set up by specifying co-simulation settings, including the coupled boundaries, coupled species, and connection settings. For instructions, refer to the section, Specifying Co-Simulation Settings.

Contents:

Specifying Co-Simulation Settings
Exporting Mass Flow Rate to a File

Specifying Co-Simulation Settings

Before connecting to GT-SUITE, you couple the GT-SUITE CFD connections with the corresponding STAR-CCM+ objects. You also specify co-simulation global settings, such as the connection method and the location of the GT-SUITE case file.

You specify all the co-simulation settings within the **Co-Simulations** > **Link 1** node, which appears in the simulation tree when you activate a co-simulation model. For information on all the properties available for this node and its child nodes, refer to GT-SUITE Link Reference.

In STAR-CCM+, you define co-simulation zones to couple GT-SUITE CFD connections with the corresponding STAR-CCM+ boundaries:

1. Create a co-simulation zone for each CFD connection defined in GT-SUITE.

   STAR-CCM+ automatically adds a co-simulation zone when you activate the **GT-SUITE Co-Simulation** model. To create additional zones, right-click the **Link 1** > **Zones** node and select **New**.

STAR-CCM+ matches co-simulation zones and CFD connections by name. To successfully couple STAR-CCM+ with GT-SUITE:

2. Rename each co-simulation zone to the name of the corresponding GT-SUITE CFD connection.

Use the co-simulation zones that you created to couple STAR-CCM+ boundaries with GT-SUITE. To couple a boundary with a GT-SUITE CFD connection:

3. Expand the relevant boundary node, **[Fluid Region]** > **Boundaries** > **[Boundary]**.
4. Select the **Physics Conditions** > **External Code Coupling Specification** node.
5. Set *External Code Coupling Specification* to the co-simulation zone associated with the CFD connection of interest.

   When you assign a boundary to a co-simulation zone, the *Coupled Model Parts* property for the zone automatically updates to display the name coupled boundary.
6. Repeat the previous steps as required, to specify all coupled boundaries.

The imported and exported fields depend on the type of coupled boundary and CFD connection. To specify the exchanged fields in STAR-CCM+:

7. Expand the relevant **Link 1** > **Zones** > **[Zone]** > **Conditions** node.
8. Select the **Field Export Option** and **Field Import Option** nodes in turn and specify the exported and imported fields based on the associated CFD connection, (see Exchanged Fields).
   For example, to couple a mass flow inlet with a **CFDFlowConnBeta** of type **Flow (CFD imposes GT pressure): Mass Flow Rate**:

1. Set **Field Export Option** to **Pressure**.
2. Set **Field Import Option** to **Mass**.

For information on all available combinations, refer to the table in Exchanged Fields.

For multi-component gases, associate the GT-SUITE species with corresponding gas components or passive scalars in STAR-CCM+:

9. Right-click the **Link 1** > **Values** > **Partner Material Components** node and select **New Partner Material Component**.
10. Repeat the previous step to create a material component node for each of the species defined in GT-SUITE.
11. Rename each material component node to the name of the corresponding GT-SUITE species.
12. Select each of the **Partner Material Components** > **[Material Component]** node and set *Coupled STAR-CCM+ Mixture Component/Passive Scalar* to the corresponding gas component or passive scalar.

Note: Make sure that each GT-SUITE species is coupled with a unique mixture component or passive scalar. If you assign two GT-SUITE species to the same gas component, the solver stops with an error message before the first iteration.

Exporting Mass Flow Rate to a File

When you couple a pressure outlet boundary with a **CFDFlowConnBeta** connection, STAR-CCM+ can export either velocity or mass flow rate. To export mass flow rate, you save mass flow data to files, which you then provide to GT-SUITE.

To couple a pressure outlet boundary with a **CFDFlowConnBeta** connection of type **Pressure (CFD imposes GT mass flow)**, prepare a mass flow report in STAR-CCM+:

1. Right-click the **Reports** node and select **New Report** > **Surface Integral**.
2. Select the **Reports** > **[Surface Integral]** node and set the following properties:

Property	Setting
Field Function	**Mass Flux**
Parts	Select the coupled pressure outlet boundary
Representation	**Volume Mesh**

3. Right-click the **[Surface Integral]** report node and select **Create Monitor and Plot from Report**.

Instruct STAR-CCM+ to export the plot data to a file at each time-step:

4. Select the **Plots** > **[Surface Integral Monitor Plot]** > **Update** node and set the following properties:

Property	Setting
File Format	**csv**
Save to File	**Activated**
Output Directory	Specify the directory where STAR-CCM+ saves the files.

Property	Setting
Base Filename	Specify a base name for the files. STAR-CCM+ saves the files as `[Output Directory]/[PlotName]_[Base Filename]_[time-step#].csv`.
Trigger	**Time Step**

For information on importing the file in GT-SUITE, see Importing Mass Flow Rate from a File.

Coupling to GT-SUITE

Specify the settings for launching and connecting to GT-SUITE, then initiate coupling.

1. Expand the **Co-Simulations** > **Link 1** node.
2. Select the **Conditions** > **GT-SUITE Version** node and select the version of GT-SUITE that you are running.
3. Select the **Conditions** > **Partner Filesystem** node and specify the location of the GT-SUITE model file:
   - When STAR-CCM+ and GT-SUITE run on the same machine, set *Option* to **Local Filesystem**.
   - When STAR-CCM+ and GT-SUITE run on different machines, set *Option* to **Remote Filesystem**.

     For remote locations, the GT-SUITE model file must be located in the directory from which you execute the GT-SUITE command, `gtlink`.
4. Select the **Values** > **GT-SUITE Model File** node and set *GT-SUITE Input File* to either the `.gtm` or `.dat` model file, as follows:
   - When STAR-CCM+ and GT-SUITE run on the same machine, click ... (**Custom Editor**) and use the dialog to select the file in the local filesystem.
   - When STAR-CCM+ and GT-SUITE run on different machines, specify the full name of the model file, including the extension (either `.gtm` or `.dat`), without including any folder path.

To initiate the communication with GT-SUITE, STAR-CCM+ requires the location of the GT-SUITE coupling libraries (see GT-SUITE Coupling Libraries):

5. Select the **Conditions** > **Load Partner Library Option** node and set *Option* to one of the following:
   - When STAR-CCM+ and GT-SUITE run on the same machine:
     1. Select **Use environment variable**.
     2. Select the **Values** > **GTIHOME Environment Variable** and set *GTIHOME* to the path to the GT-SUITE installation. For more information, see Load Partner Library Option.

        STAR-CCM+ looks for the library file in the default location (see GT-SUITE Coupling Libraries) within the specified GT-SUITE installation. At the default location, GT-SUITE provides library files for double or single precision coupling.
     3. Select the **Conditions** > **Coupling Precision Option** node and set *Option* to either **Double** or **Single**, as required.
   - When STAR-CCM+ and GT-SUITE run on different machines:
     1. Select **Specify library file**.
     2. Copy the required single or double precision library file (for information on the default location, see GT-SUITE Coupling Libraries) to the machine where STAR-CCM+ is running.
     3. Select the **Values** > **GT-SUITE Library** node and set *GT-SUITE Library* to the library file that you copied.

Specify the method for launching GT-SUITE:

6. Select the **Conditions** > **Launch Partner Option** node and set *Option* to one of the following:

- To launch GT-SUITE manually, select **Do Not Launch**. If GT-SUITE runs on a remote host, this is the only suitable option.
- To instruct STAR-CCM+ to automatically launch GT-SUITE:
  1. Select **Launch Default Application**.
  2. Select the **Values** > **GTIHOME Environment Variable** node and set *GTIHOME* to the path to the GT-SUITE installation.
  3. Select the **Values** > **GTISOFT_LICENSE_FILE Environment Variable** node and set *GTISOFT_LICENSE_FILE* to the path to the directory containing the license file.

  For more information, see Launch Partner Option.

When STAR-CCM+ launches GT-SUITE (that is, **Launch Partner Option** is set to **Launch Default Application**), STAR-CCM+ automatically uses the launching details for connection. When you launch GT-SUITE manually (that is, **Launch Partner Option** is set to **Do Not Launch**), the coupling library requires the host and port on which GT-SUITE is running. The host and port information is stored in a connection file, `gtlink.prm`. To create this file:

7. Select the **Conditions** > **Connect Option** node and set *Option* to **Partner Code Creates Connection File**.

   When launched, GT-SUITE automatically creates the required `gtlink.prm` connection file in the directory containing the `gtlink` command.
8. Copy the `gtlink.prm` file to the directory containing the STAR-CCM+ simulation file.

Note: Versions of GT-SUITE older than v2017 do not create the connection file automatically. In this case, you can instruct STAR-CCM+ to create the connection file in the simulation directory, based on the host and port that you specify using the **Values** > **Partner Host:Port** node. For more information, see Connect Option.

9. To run the co-simulation, click **Run (Ctrl+R)**.

   STAR-CCM+ automatically loads the coupling library and, based on the specified launch and connect methods, connects, or launches and connects to, to the specified GT-SUITE server.

   On connection, STAR-CCM+ matches the co-simulation zones with the corresponding GT-SUITE CFD connections.

   STAR-CCM+ also allows you to manually load or unload the coupling library, and manually control the launch and connect steps. For more information, see Link Right-Click Actions.

To terminate the connection:

10. Right-click the **Co-Simulations** > **Link** node and select **Terminate**.
11. To restart the co-simulation:
    - If the STAR-CCM+ session is active, select **Solution** > **Clear Solution**. You can then start GT-SUITE using the previous method.
    - If you close STAR-CCM+, perform a restart-run by modifying the GT-SUITE model file to restart from a user-defined time-step. Restart the STAR-CCM+ simulation from an autosave file.

Contents:

Running a Co-Simulation with GT-SUITE on Parallel Hosts

Running a Co-Simulation with GT-SUITE on Parallel Hosts

If you intend to run on parallel hosts (a multiple node cluster or several machines over a network), the procedure for running the co-simulation must meet certain criteria. The following examples demonstrate coupling over parallel hosts using two machines or nodes; A is the master host and B is the slave host.

Running GT-SUITE on the Master Host

The easiest way to run a coupled simulation is to allow STAR-CCM+ to launch the GT-SUITE server automatically on the master host. However, the master host must have access to the GT-SUITE installation.

To run the GT-SUITE server on the master host, A:

1. Launch a parallel simulation using the procedure in Using a Remote Multi-Core Workstation, ensuring that the master host is the first entry in the table. Alternatively, from the command line:

   ```
   starccm+ -server -on A:x,B:y -np x+y
   ```

   where:

   - `x` – is the number of processes on A
   - `y` – is the number of processes on B

Running GT-SUITE on a Slave Host

If the master host does not have access to a GT-SUITE installation, run the GT-SUITE server on one of the slave hosts. This scenario requires a different coupling procedure.

To run the GT-SUITE server on the slave host, B:

1. Launch a parallel simulation using the procedure in Using a Remote Multi-Core Workstation, ensuring that the master host is the first entry in the table. Alternatively, from the command line:

   ```
   starccm+ -server -on A:x,B:y -np x+y
   ```

   where:

   - `x` – is the number of processes on A
   - `y` – is the number of processes on B

Setting Up STAR-CCM+ and GT-SUITE for a Restart Run

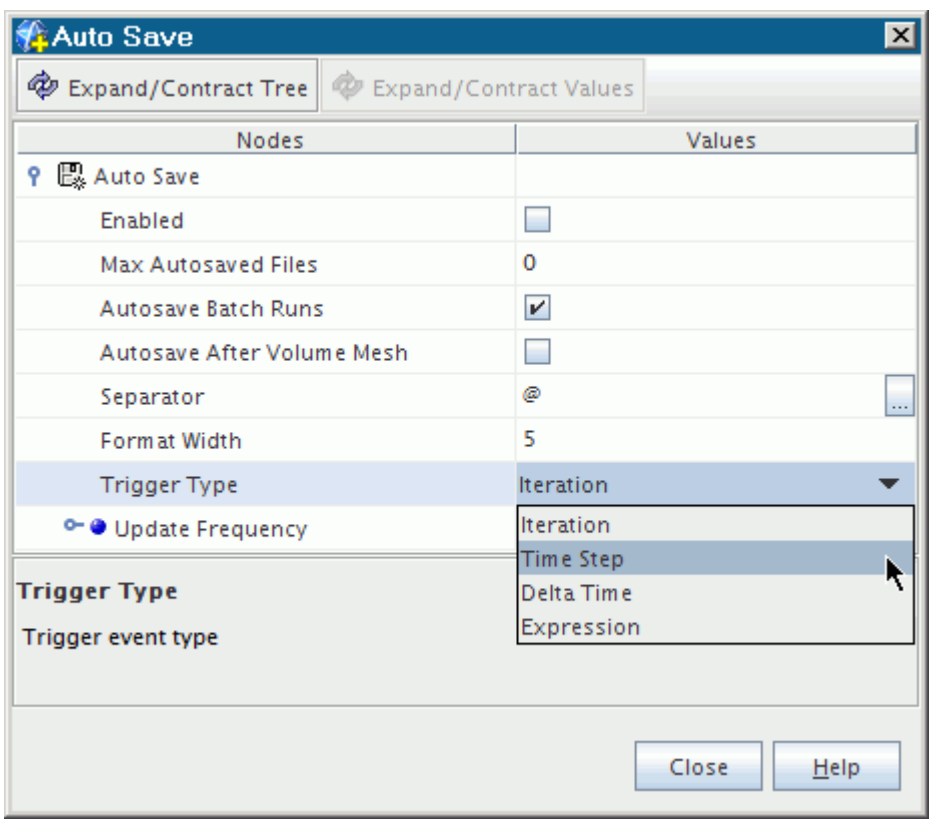
On occasion, it is necessary to stop the solver prematurely, or continue running an existing analysis for more time-steps. To meet these requirements, GT-SUITE co-simulation provides facilities to perform a restart run. This section details this procedure for a case that has been stopped before completion.

To continue an analysis for more time-steps, the following steps can also be followed, but increase the STAR-CCM+ stopping criteria to extend the duration of the simulation.

Before running the solver, enable the **Auto Save** feature in STAR-CCM+ to save the simulation at regular intervals. The autosaved files provide an exact time-step from which you can restart the solver, thus ensuring the two codes begin at the same point in the analysis. The autosave file has a name that corresponds to the current time-step, for example, `casename@01000.sim` is the saved simulation at time-step 1000.

To enable **Auto Save**:

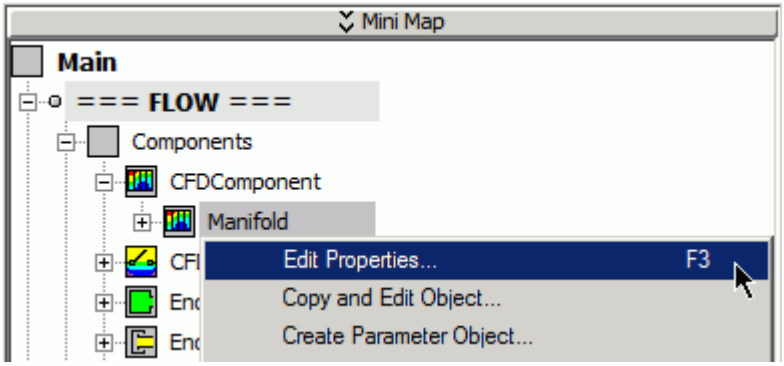
1. From the menu bar, select **File** > **Auto Save...**.
2. In the *Auto Save* dialog, set *Trigger Type* to **Time Step**, as the restart time-step must be specified in the GT-SUITE case.



3. Set *Frequency* and *Max Autosaved Files* as required.
4. Start the GT-SUITE solver and obtain a solution in the usual manner.

To restart the simulation in GT-SUITE, modify the case setup so that the solver restarts from a user-specified time-step:

5. Right-click a CFD component, and select **Edit Properties...**



6. In the *Edit Object* dialog, select the *Advanced* tab.
7. Set *Read to CFD Time Step Number* to `def`.
8. Save the GT-SUITE case file and create a `.dat` file, by selecting **Run** > **Run-Simulation Utilities** > **Create .dat file** from the menu bar.

To restart the simulation in STAR-CCM+:

9. Load the autosaved file into STAR-CCM+.
   - If you perform a restart from a different directory, make sure that you copy the relevant files from the previous working directory (that is, the `*.cfdreload`, `*.sim`, `*.gtm`, and `*.gdx` files).
10. Run the solvers in the usual manner.

    The STAR-CCM+ solver restarts from the last time-step and the GT-SUITE solver restarts from the user-specified time-step.

See Also:

Automatically Saving at Regular Intervals

GT-SUITE Co-Simulation Model Reference

The **GT-SUITE** co-simulation model allows you to couple STAR-CCM+ and GT-SUITE in a two-way co-simulation analysis.

Provided By	**[physics continuum]** > **Models** > *Co-Simulation Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **GT-SUITE**	
Requires	Physics Models: *Space*: **Three Dimensional** *Time*: **Implicit Unsteady** *Material*: one of **Gas**, **Liquid**, **Multi-Component Gas** (**Non-reacting**) *Flow*: any *Optional Models*: **Co-Simulation**	
Activates	Boundary Inputs	**External Code Coupling Specification**. See Boundary Settings.
	Solvers	**GT-SUITE Coupling Solver**. See GT-SUITE Coupling Solver Properties.
	Field Functions	**Average Zone Flag**. See Field Functions.
	Other Nodes	**Co-Simulations** > **Link 1**. See GT-SUITE Link Reference.

Boundary Settings

Only available for boundaries of type **Velocity Inlet**, **Mass Flow Inlet**, and **Pressure Outlet**.

External Code Coupling Specification

Allows you to assign the boundary to a co-simulation zone for coupling. During the co-simulation, the STAR-CCM+ boundary exchanges data with the corresponding coupled surface in the partner simulation, according to the co-simulation zone specifications.

GT-SUITE Coupling Solver Properties

Solver Frozen

When activated, the solver does not update any quantity during an iteration. Use this for debugging purposes, or to keep one set of equations fixed while you solve another set. Deactivated by default.

Field Functions

Average Zone Flag

Before sending data to GT-SUITE, STAR-CCM+ averages the computed fields across sampling volumes adjacent to the coupled boundaries, to provide transition from the three dimensional to the one dimensional solution. This field function allows you to visualize the sampling volumes in a scalar scene. The displayed field function values have no significance and simply distinguish the individual zones.

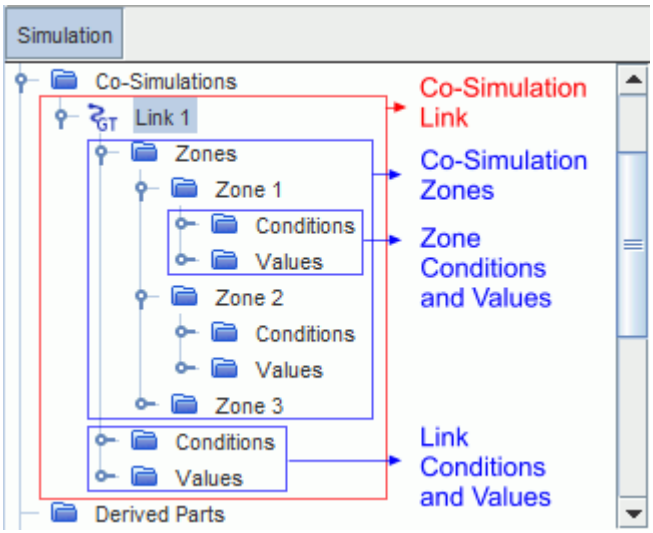
Contents:

GT-SUITE Link Reference

GT-SUITE Link Reference

In STAR-CCM+, you specify the settings for co-simulation with GT-SUITE in a co-simulation link under the **Co-Simulations** node.

A co-simulation link node, **Co-Simulations** > **Link 1**, becomes available in the simulation when you activate the **GT-SUITE Co-Simulation** model. This node allows you to specify co-simulation global settings, such as the connection method, as well as to define and set up co-simulation zones, which couple the STAR-CCM+ boundaries with the corresponding GT-SUITE CFD Connections.



This section describes the global settings available for co-simulation with GT-SUITE. For details on the co-simulation zone settings, see Co-Simulation Zones Reference.

Link Properties

Co-Simulation Type

Displays which partner code is coupled to STAR-CCM+ in the co-simulation analysis represented by this link. Read-only.

Link Conditions and Values

You specify the general settings of the co-simulation under the **Link** > **Conditions** and **Link** > **Values** nodes.

GT-SUITE Version

Allows you to specify which version of GT-SUITE you couple with STAR-CCM+. For a list of all supported versions of GT-SUITE, refer to the section *External Packages Support* of the STAR-CCM+ Release Notes.

Load Partner Library Option

Allows you to specify the location of the required GT-SUITE coupling library file (see GT-SUITE Coupling Libraries).

Method	Corresponding Condition/Value Nodes
Specify Library File Allows you to manually specify the path to the GT-SUITE library file. This option is suitable for cases where the location of the library file is not the default. See Coupling to GT-SUITE.	**GT-SUITE Library** Specifies the path and name of the GT-SUITE library file. You can type the full path and name manually, or you can click ... (**Custom Editor**) and navigate to the file using the file selection dialog.
Use environment variable Instructs STAR-CCM+ to load the library from the default location (see GT-SUITE Coupling Libraries). This option requires you to save the path to the GT-SUITE installation in the environment variable, `$GTIHOME`.	**GTIHOME** Specifies the path to the GT-SUITE installation. You can type the full path manually, or you can click ... (**Custom Editor**) and navigate to the installation directory using the file selection dialog. This setting overwrites any existing `$GTIHOME` environment variable, for the current co-simulation run. When restoring a simulation containing the **GTIHOME** setting, STAR-CCM+ uses the `$GTIHOME` environment variable, if this is set on the system. If not set, STAR-CCM+ uses the **GTIHOME** setting that is found in the simulation. **Coupling Precision Option** Allows you to choose between the **Single** precision library file or the **Double** precision library file that are provided in the GT-SUITE installation. See GT-SUITE Coupling Libraries.

Launch Partner Option

Allows you to specify which method STAR-CCM+ uses to launch GT-SUITE.

Method	Corresponding Value Nodes
Launch Default Application Automatically launches GT-SUITE on co-simulation run. This option requires you to specify the path to the GT-SUITE installation and license file using the environment variables `$GTIHOME` and `$GTISOFT_LICENSE_FILE`.	The following settings allow you to overwrite the `$GTIHOME` and `$GTISOFT_LICENSE_FILE` environment variables defined on your system, for the current co-simulation run: **GTIHOME** Specifies the path to the GT-SUITE installation. You can type the full path manually, or you can click ... (**Custom Editor**) and navigate to the installation directory using the file selection dialog.

	GTISOFT_LICENSE_FILE Specifies the GT-SUITE license file. When restoring a simulation that contains these setting, STAR-CCM+ uses the environment variables, if these are set on the system. If not set, STAR-CCM+ uses the settings that are found in the simulation.
Do Not Launch Requires you to manually launch GT-SUITE.	None

Connect Option

Allows you to specify which method STAR-CCM+ uses to connect to GT-SUITE.

When STAR-CCM+ launches GT-SUITE (that is, **Launch Partner Option** is set to **Launch Default Application**), STAR-CCM+ automatically uses the launching details for connection.

When you launch GT-SUITE manually (that is, **Launch Partner Option** is set to **Do Not Launch**), the coupling library requires the host and port on which GT-SUITE is running. The host and port information is stored in a connection file, `gtlink.prm`, which can be created using the following methods:

Method	Corresponding Value Nodes
Partner Code Creates Connection File STAR-CCM+ does not create the connection file. From v2017, when you launch GT-SUITE in server mode using the `gtlink` command, GT-SUITE creates the required `gtlink.prm` connection file in the directory from which you instantiate `gtlink`. To use this file for connection, copy it to the directory containing the STAR-CCM+ simulation file.	None
STAR-CCM+ Creates Connection File On co-simulation run, STAR-CCM+ automatically writes the specified host and port details to a file and saves the file in the simulation directory as `gtlink.prm`.	**Partner Host:Port** Specifies the host and port where GT-SUITE is running. Provide these details in the format `[Host Name]:[Port Number]` **GT-SUITE Partner Connection File** Displays the name of the connection file created by STAR-CCM+ (read-only).

Partner Filesystem

Specifies whether the GT-SUITE files are located on a local or remote file system. See GT-SUITE Model File.

GT-SUITE

Displays the status of the connection between STAR-CCM+ and GT-SUITE and the GT-SUITE library that STAR-CCM+ uses to communicate with GT-SUITE.

GT-SUITE Model File

Specifies the location and name of the GT-SUITE case file. You can provide either the `.gtm` file or the `.dat` file. When providing the `.dat` file, make sure it is up-to-date. See Coupling to GT-SUITE.

Partner Material Components

For multi-component gases, allows you to couple the fluid components defined in GT-SUITE with gas components and passive scalars defined in STAR-CCM+.

The **[Material Component]** nodes must match the species defined in GT-SUITE in number and name. See Specifying Co-Simulation Settings.

Link Right-Click Actions

Load Partner Library

Loads the specified GT-SUITE library file. See Load Partner Library Option.

Launch and Connect

Launches the specified partner simulation and starts the connection, coupling the STAR-CCM+ simulation with the partner simulation.

Connect

Starts the connection, coupling the STAR-CCM+ simulation with the running partner simulation that you specified.

Terminate

Terminates the connection between STAR-CCM+ and GT-SUITE.

Unload Partner Library

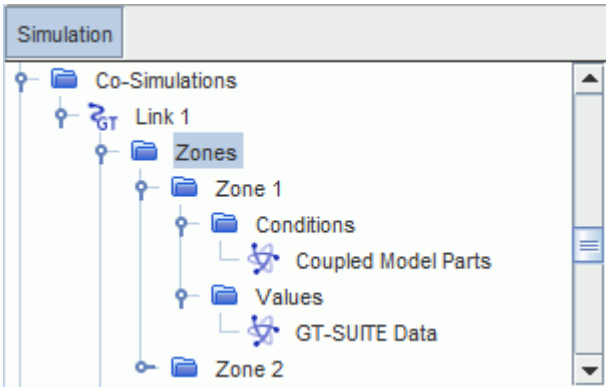
Unloads any GT-SUITE library file that was previously loaded.

Contents:

Co-Simulation Zones Reference

Co-Simulation Zones Reference

Co-simulation zones allow you to couple STAR-CCM+ boundaries with the partner GT-SUITE CFD connections.



Zones Manager Properties

Co-Simulation Zones

Displays the number of co-simulation zones defined for this link. Read-only.

Zones Manager Right-Click Actions

New

Adds a co-simulation zone under the `Zones` node. For coupling to be successful, the name of a co-simulation zone must match the name of the corresponding GT-SUITE CFD connection.

Zone Properties

Index

Displays a unique identifier for this zone. Read-only.

Zone Type

Specifies what type of STAR-CCM+ objects are assigned to this co-simulation zone. The available option is **Surface Zone**, which allows you to couple STAR-CCM+ boundaries to their surface counterparts in the partner simulation (surface-to-surface coupling).

Zone Conditions and Values

Coupled Model Parts

Displays the parts on the STAR-CCM+ model that exchange data with the partner simulation, through this co-simulation zone. As this type of co-simulation only supports surface-to-surface coupling, the coupled model parts can only be boundaries.

This property is read-only. To assign a STAR-CCM+ boundary to a co-simulation zone, use the **External Code Coupling Specification** boundary condition.

Field Export Option, Field Import Option

Specifies the fields that STAR-CCM+ exchanges with GT-SUITE. The available options depend on the type of boundary that is assigned to the zone. When multiple options are available, choose the option based on the type of GT-SUITE CFD connection that is coupled with the boundary (see Exchanged Fields).

Boundary Type	Field Export Option	Field Import Option
Velocity Inlet	Pressure	Velocity
Mass Flow Inlet	Pressure	Velocity and Density
		Mass
Pressure Outlet	Velocity	Pressure
	Mass Flow Rate (via file)	

GT-SUITE Zone Data

Displays the settings defined for the coupled CFD connection in GT-SUITE. As these settings are defined in GT-SUITE, all the properties are read-only:

Area: displays the CFD cross-section that is specified for the coupled CFD connection in GT-SUITE (read-only).
ID: displays the ID number that is given to the CFD connection within the GT-SUITE case (read-only). This index is not used to match the co-simulation zone with the corresponding CFD connection. Co-simulation zones and CFD connections are matched by name.
Sampling Length: displays the length of the sampling volume across which STAR-CCM+ averages the computed fields before sending them to GT-SUITE. The sampling volume length is defined in GT-SUITE (read-only).

WAVE Co-Simulation

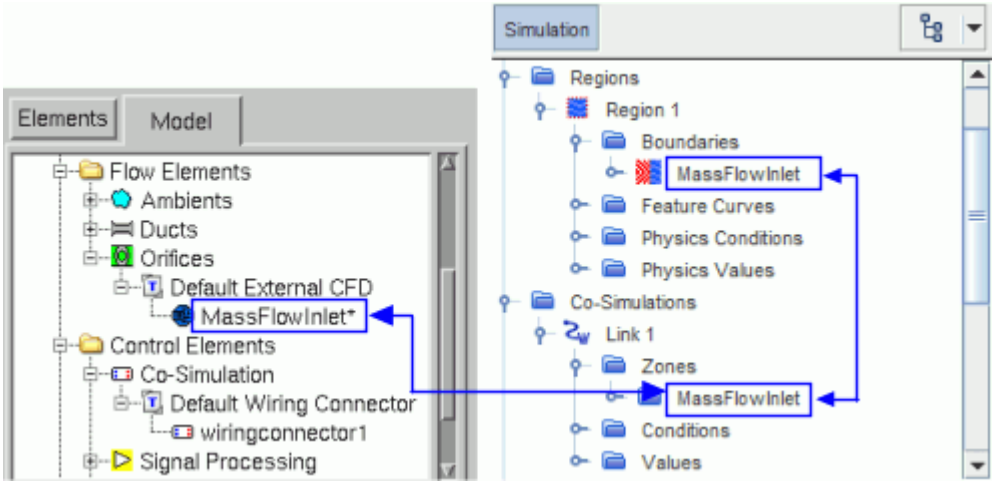
STAR-CCM+ allows for two-way co-simulation with WAVE, a 1D engine and gas dynamics simulation tool from *Ricardo Software*.

For more information on WAVE, visit the official website.

In a co-simulation between STAR-CCM+ and WAVE, WAVE computes a full engine-power analysis and allocates a portion of the flow to STAR-CCM+ for solving in 3D. Co-simulation reduces the computational times in both STAR-CCM+ and WAVE. STAR-CCM+ receives time-accurate boundary conditions from WAVE, and WAVE receives space-accurate boundary conditions from STAR-CCM+.

Co-Simulation Interface

In STAR-CCM+, you define co-simulation zones that couple the CFD orifices defined in WAVE with corresponding STAR-CCM+ mass flow inlet boundaries. Each co-simulation zone uniquely couples a WAVE CFD Orifice with a STAR-CCM+ boundary. Co-simulation zones and CFD orifices must match in number and name.



During the co-simulation, WAVE sends the computed flow quantities and species mass fractions to STAR-CCM+, which then applies them uniformly at the coupled boundary. Before sending data back to WAVE, STAR-CCM+ averages the computed fields in order to provide one-dimensional values.

You can also exchange additional single-value data, for example shaft speed, torque, and flow rate, by associating STAR-CCM+ auxiliary field specifications with WAVE control links.

WAVE co-simulation uses an explicit coupling scheme, that is, data is exchanged once per time-step.

Exchanged Fields

Co-simulation zones allow coupling between WAVE orifices and STAR-CCM+ mass flow inlets. STAR-CCM+ automatically sets the boundary conditions based on the imported fields.

For exchange of single-value data using STAR-CCM+ auxiliary fields and WAVE control links, you manually set the relevant boundary conditions using the field functions that store the imported data.

Option	Fields Sent to WAVE	Fields Received from WAVE
Co-simulation zones	• Density • Temperature • Pressure • Mass Flow • Species Mass Fractions	• Density • Temperature • Pressure • Mass Flow • Species Mass Fractions
Single-value data exchange	Auxiliary fields representing single-value data	Control link data

Coupled Species

WAVE uses combustion models to determine the products of reaction between fuel and air. WAVE uses five active scalars in modeling the combustion process:

- **air**—Air, with gas composition aN2, bO2, cCO2, dH2O (defined in WAVE pre-processing).
- **burnedAir**—Products of combustion, with gas composition aN2, bO2, cCO2, dH2O.
- **burnedFuel**—Remaining products of combustion, with gas composition eCO, fNO, gOH, hN, iO, jH, kH2.
- **fuelLiquid**—Fuel in liquid form, with molecular composition CCHHOONN (specified in WAVE pre-processing).
- **fuelVapour**—Fuel in vapor form, with molecular composition CCHHOONN (specified in WAVE pre-processing).

In a co-simulation with WAVE, you couple the WAVE active scalars with corresponding mixture components or passive scalars defined in STAR-CCM+. The mass fractions are exchanged at the co-simulation interface.

Additionally, you can couple passive scalars defined in WAVE with mixture components or passive scalars defined in STAR-CCM+.

Contents:

Preparing the WAVE Simulation for Co-Simulation
Preparing the STAR-CCM+ Simulation for Co-Simulation
Coupling STAR-CCM+ to WAVE
WAVE Co-Simulation Model Reference

Preparing the WAVE Simulation for Co-Simulation

Modify a WAVE simulation for co-simulation with STAR-CCM+.

The following instructions assume that you start from a fully set up WAVE simulation. Also, some of the following instructions may vary based on the version of WAVE that you are running. For more information on setting up a case in WAVE, refer to the WAVE User Guide.

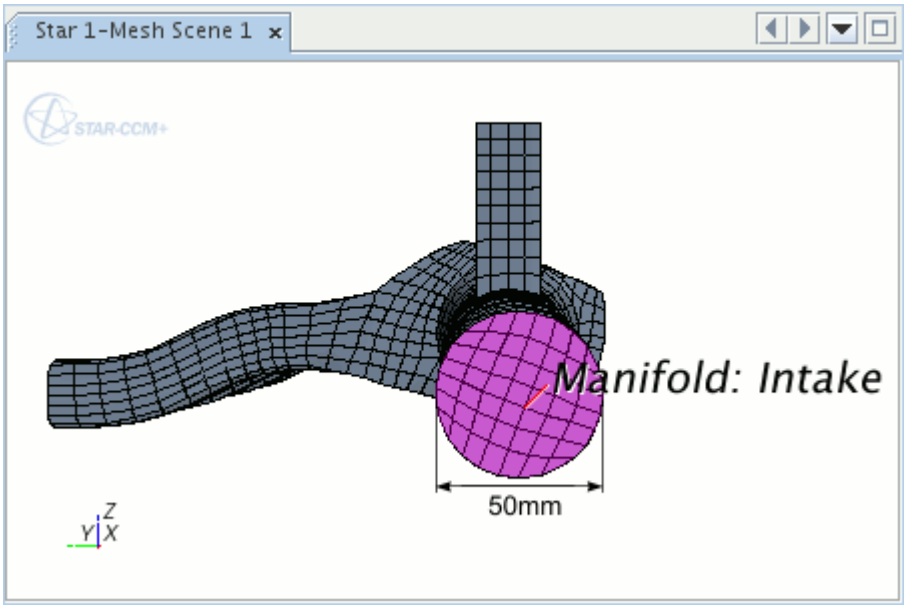
Before specifying co-simulation settings, it is good practice to run a standalone WAVE analysis. You can use the results from the standalone WAVE analysis as initial conditions in STAR-CCM+.

1. Run the WAVE simulation independently until converged.
2. Using a text editor, open the `.out` file that is associated with the WAVE simulation. This file is typically located in the same directory as the WAVE simulation file.
3. Note the average pressure and temperature for the adjacent duct to the CFD inlet.
   For example:

```
-------------------------------------------------------------------------------
  F I N A L   O U T P U T   O F   D U C T S
-------------------------------------------------------------------------------
         Duct        Junction  TWALL   TAV    PAV     PMAX    PMIN     UMAX
                                [K]    [K]   [bar]    [bar]   [bar]    [m/s]

        duct1            amb1  300.0  300.0  1.000   1.000   1.000     0.1
                       Intake  300.0  300.0  1.000   1.000   1.000     0.1

       duct10       cylinder2  400.0  595.8  1.000   1.052   0.947    57.2
                        orif6  400.0  564.5  1.001   1.049   0.956    15.4

       duct11            cyl1  300.0  329.9  0.172   0.368   0.146     0.0
                        orif1  300.0  388.5  0.172   0.365   0.146     4.9
```

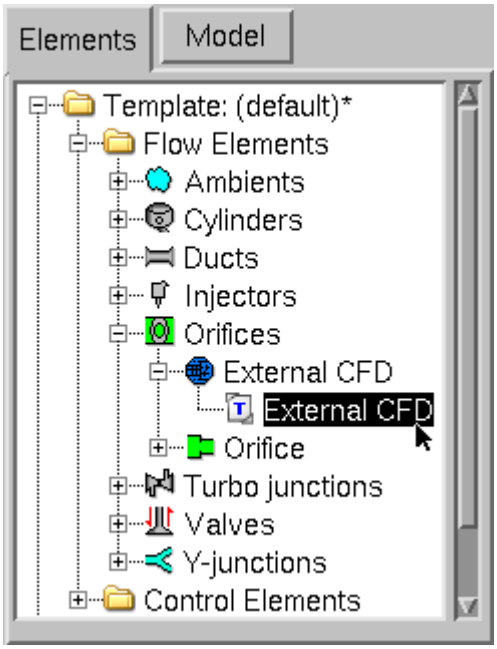
You can now modify the WAVE simulation for co-simulation with STAR-CCM+.

For each CFD orifice in WAVE, specify the diameter of the coupled zone in STAR-CCM+. If the zone cross-section is non-circular, use the effective diameter, that is, the diameter of an equivalent circle with the same cross-sectional area as the zone. In the example below, the diameter at the manifold intake is 50 mm.



To set the CFD orifice diameter in WAVE:

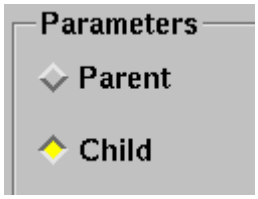
4. Double-click **External CFD Orifice**.



5. In the *External CFD Orifice* dialog, set the value in the **Diameter** textbox to match the zone diameter in STAR-CCM+.

Specify WAVE as the child process. This parameter allows STAR-CCM+ to start the WAVE solver when obtaining a solution:

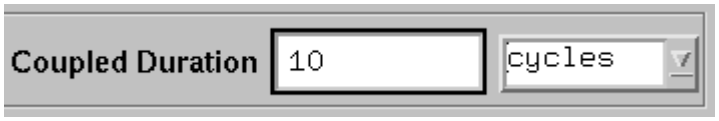
6. From the WAVE menu bar, select **Model** > **External** > **CFD...**
7. From the *Parameters* group, select **Child**.



Specify the number of pre-cycles that WAVE runs independently before sending data to STAR-CCM+. Pre-cycles improve stability in WAVE, providing more accurate boundary conditions to STAR-CCM+ and, therefore, reducing the time to reach a converged solution.

8. From the WAVE menu bar, select **Model** > **External** > **CFD...**
9. From the *Process Control* group, select **Postponed**.

10. Specify the number of coupled cycles in the *Coupled Duration* textbox.



This value is subtracted from the total number of simulation cycles, to determine the number of pre-cycles.

The WAVE solver runs independently until it reaches the number of pre-cycles. When the number of pre-cycles is met, WAVE starts sending data to STAR-CCM+.

The transfer of data to WAVE from STAR-CCM+ can be delayed until a defined time-step. This facility allows the STAR-CCM+ solver to reach a level of stability while receiving data from WAVE, before sending data back to WAVE. To set a transfer delay:

11. From the WAVE menu bar, select **Model** > **External** > **CFD...**
12. Enter the number of time-steps in the *One Way Steps* textbox.

Preparing the STAR-CCM+ Simulation for Co-Simulation

Set up a STAR-CCM+ simulation for co-simulation with WAVE.

1. Before you set up the physics continua, prepare the required geometry, regions, interfaces, and mesh, as appropriate to your analysis.

   For more information on these general operations, see STAR-CCM+ Simulation Process.

2. In the co-simulation physics continuum, activate the following physics models:

Group	Physics Model
Space	**Three Dimensional**
Time	one of **Implicit Unsteady**, **PISO Unsteady**
Material	**Multi-Component Gas**
Reaction Regime	**Non-Reacting**
Flow	any
Enabled Models	**Gradients** (selected automatically)
Equation of State	any
Energy	any
Viscous Regime	any
Optional Models	**Co-Simulation**
Co-Simulation Models	**WAVE**
Optional Models	**Passive Scalar** **Note:** This model can be neglected if you represent all species of a multi-component gas with gas components.

   You can select further models to meet specific analysis requirements.

3. Define a mixture component or passive scalar for each active scalar defined in WAVE.

   The choice between using a mixture component or passive scalar to represent a WAVE active scalar depends on user requirements. A mixture component influences the physical properties of the mixture, providing the most accurate and realistic solution. A passive scalar has no effect on the physical properties of the mixture and so reduces compute time. If passive scalars are used for all species, a background fluid must be present in the region.

   To add a mixture component to the simulation:

   a) Expand the **Continua** > **[Physics Continuum]** > **Models** node.

   b) Right-click the **Multi-Component Gas** > **Gas Components** node and select **Select Mixture Components...**

   c) In the *Select Mixture Components* dialog, select the gas components, then click **Apply**.

Note: Although the WAVE *fuelLiquid* species is a liquid, it can be represented by a gas in the **Multi-Component Gas** model. This does not affect results, as the gas simply represents the species in STAR-CCM+, with the physics values received from WAVE.

To add a passive scalar to the simulation:

a) Right-click the **Passive Scalars** node and select **New**.

If you wish to couple WAVE passive scalars with STAR-CCM+, use the same procedure to define the required mixture components or passive scalars in STAR-CCM+.

To reduce the numerical round-off, set the reference pressure to the average pressure in WAVE. You can identify the average pressure in WAVE by running a standalone WAVE analysis. You can then read the average pressure in the CFD component in the relevant WAVE `.out` file (see Preparing the WAVE Simulation for Co-Simulation).

To set the reference pressure:

4. Select the **[Physics Continuum]** > **Reference Values** > **Reference Pressure** node and set *Value* to the average pressure in WAVE.

Note: All pressures in STAR-CCM+ are defined relative to the reference pressure. In WAVE, all pressures are absolute. Take these differences into consideration when defining pressure values.

The initial conditions in STAR-CCM+ are used to reduce the time that it takes for the solution to converge. For best initialization, set the STAR-CCM+ initial conditions to the average values from WAVE, which you can obtain by running a standalone WAVE analysis.

5. Expand the **[Physics Continuum]** > **Initial Conditions** node and specify the initial conditions as follows:
   a) As the initial pressure in STAR-CCM+ is relative to the reference pressure, set the **Pressure** value to the difference between the average pressure from WAVE and the STAR-CCM+ reference pressure.
   b) As the initial temperature in STAR-CCM+ is an absolute value, set the **Static Temperature** value to be the same as the average temperature from WAVE.
   c) Set the species mass fractions to the average values from WAVE, by setting the **Species Mass Fraction** value. Click ... (**Custom Editor**) to enter the mass fractions in an array. Initial mass fractions must sum to the value of 1.
   As an example, if coupling to a component on the intake side of an engine, it would be intuitive to initialize the region with air. Hence, the mass fractions would be as follows:

Mixture Component	Value
air	1.0
fuelVapour	0.0
fuelLiquid	0.0

Contents:

Exchanging Predefined Fields at Co-Simulation Zones
Exchanging Single-Value Data

Exchanging Predefined Fields at Co-Simulation Zones

To couple STAR-CCM+ mass flow inlet boundaries and WAVE CFD orifices, you define co-simulation zones. When coupling STAR-CCM+ and WAVE through co-simulation zones, STAR-CCM+ automatically defines the exchanged fields and boundary conditions.

For information on the predefined fields exchanged at co-simulation zones, see Exchanged Fields.

To couple a mass flow inlet with a WAVE CFD orifice:

1. Right-click the **Co-Simulations** > **[Link 1]** > **Zones** node and select **New**.

   STAR-CCM+ adds a co-simulation zone, **[Zone 1]**, under the **Zones** node.

Assign the relevant mass flow inlet boundary to the co-simulation zone:

2. Expand the relevant **Regions** > **[Region]** > **Boundaries** > **[Mass Flow Inlet Boundary]** node.
3. Select the **Physics Conditions** > **External Code Coupling Specification** node and set *External Code Coupling Specification* to **[Link 1]: [Zone 1]**.

   For the coupled boundaries, STAR-CCM+ automatically sets the boundary conditions based on the fields imported from WAVE, except turbulence. When modeling turbulent flow, define the **Turbulence Specification** at the coupled boundaries as appropriate to your analysis.
4. Select the **Link 1** > **Zones** > **[Zone 1]** > **Values** > **Fluid Link** node and set *Fluid Link ID* to the *Interface Number* of the associated **CFD External** orifice (defined in WAVE).
5. Define the boundary conditions at the boundaries that are not coupled to WAVE in the usual manner.

Couple each of the WAVE active scalars with the corresponding STAR-CCM+ mixture component or passive scalar.

6. Right-click the **Link 1** > **Values** > **Coupled Species** node and select **New Coupled Species**.
7. Repeat the previous step to create a coupled species node for each of the active scalars defined in WAVE.
8. Rename each coupled species node to the name of the corresponding WAVE active scalar.
9. For each **Coupled Species** > **[Species]** node, set *Mixture Component/Passive Scalar* to the relevant STAR-CCM+ mixture component or passive scalar.

   All the active scalars defined in WAVE must be coupled to either STAR-CCM+ mixture components or passive scalars. Similarly, all the mixture components defined in STAR-CCM+ must be coupled to either an active scalar or a passive scalar in WAVE.

   Optionally, you can use the same procedure to couple WAVE passive scalars with STAR-CCM+ mixture components or passive scalars. You can leave passive scalars in either WAVE or STAR-CCM+ as uncoupled.

Exchanging Single-Value Data

You can exchange single-value data between STAR-CCM+ and WAVE by associating STAR-CCM+ auxiliary field specifications with WAVE control links.

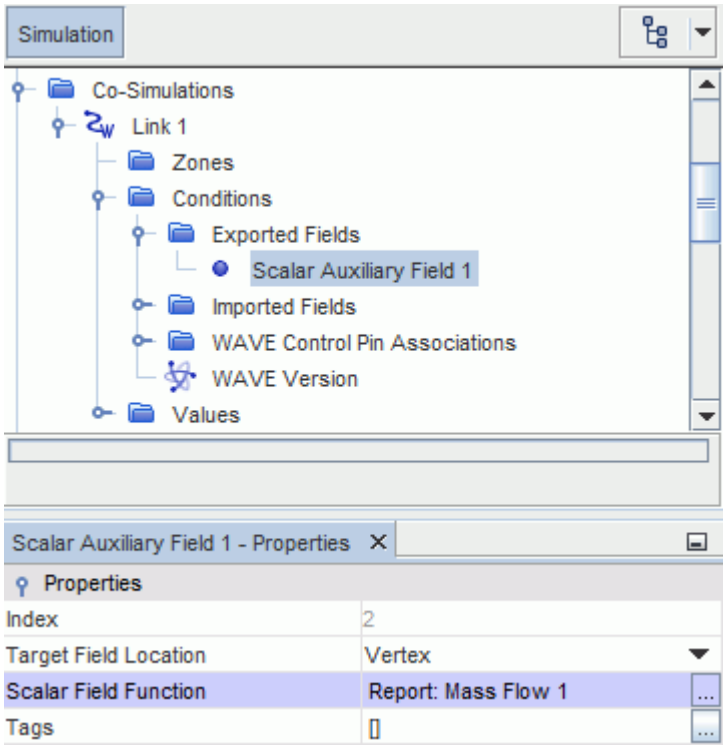
In each code, you specify the data that is exported to the partner code. To export single-value data from STAR-CCM+ to WAVE:

1. Expand the **Co-Simulations** > **[Link 1]** > **Conditions** node.

2. Right-click the **Exported Fields** node and select **Add Auxiliary Field** > **Scalar**.

   STAR-CCM+ adds an auxiliary field specification under the **Exported Fields** node.

3. Select the **[Scalar Auxiliary Field]** specification node and select the *Scalar Field Function* that you want to send to WAVE. The selected field function must represent a single value. For example, you can select the field function associated with a report.

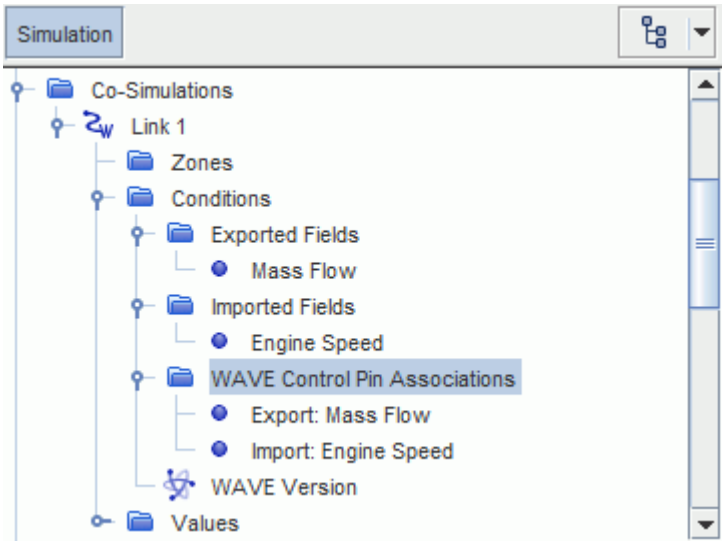


Similarly, STAR-CCM+ can import single-value data from WAVE control links. When the two codes connect, STAR-CCM+ automatically adds auxiliary field import specifications based on the export settings specified in WAVE. Alternatively, to manually create import specifications:

- Right-click the **Imported Fields** node and select **Add Auxiliary Field** > **Scalar**.

Once you define export (and, optionally, import) specifications in STAR-CCM+, you associate them with the corresponding WAVE control links using pin associations. STAR-CCM+ automatically adds pin associations for each export or import field specification.

Pin association nodes have the same name of the corresponding auxiliary field specifications, preceded by either `Export:` or `Import:`.



To specify the WAVE control links:

4. For each export specification, select the relevant **Conditions** > **WAVE Control Pin Associations** > **[Export: Scalar Auxiliary Field]** node.
5. Set *Associated ID* to the pin number of the WAVE control link that receives the data.
6. For each import specification, STAR-CCM+ automatically defines the association based on the export settings specified in WAVE. Alternatively, you can manually set the *Associated ID* for each **[Import: Scalar Auxiliary Field]** node.

   When the two codes connect, STAR-CCM+ checks that the settings are consistent. STAR-CCM+ deletes any import specification that is not associated with an export control link. For export specifications that are in conflict with export control links, STAR-CCM+ modifies the control pin associations, in order to restore a valid association setup.

STAR-CCM+ stores the imported data in field functions that are named after the corresponding WAVE control link, `[WAVE Control Pin #]`, where # is the control link pin number.

7. Use the imported data field functions to set boundary conditions in STAR-CCM+.

Coupling STAR-CCM+ to WAVE

Specify the WAVE launching details and connect to the WAVE simulation.

To couple to WAVE, STAR-CCM+ requires the path to the WAVE co-simulation libraries, the path and name of the WAVE simulation file, and the path to the WAVE executable.

1. Expand the **Co-Simulations** > **[Link 1]** node.
2. Select the **Conditions** > **WAVE Version** node and specify the version of WAVE that you are running.

   For more information, see WAVE Version.
3. Expand the **[Link 1]** > **Values** node.
4. Select the **Launch Command Line** node and set *Command Line* to the command that you use to launch WAVE.
5. Select the **RSimlink Libraries Folder** node and specify the path to the folder containing the co-simulation RSimlink libraries.

   See RSimlink Libraries.
6. Select the **WAVE Input File** node and specify the path and name of the WAVE simulation file.
7. To connect to the WAVE simulation, right-click the **[Link 1]** node and select **Connect**.
8. To run the co-simulation, click (**Run**) in the STAR-CCM+ simulation.

   STAR-CCM+ automatically disconnects from WAVE at the end of the co-simulation analysis.

   If you manually stop the co-simulation, you can disconnect from WAVE by right-clicking the **[Link 1]** node and selecting **Disconnect**.

WAVE Co-Simulation Model Reference

The **WAVE** co-simulation model allows you to couple STAR-CCM+ and WAVE in a two-way co-simulation analysis.

Provided By	**[physics continuum]** > **Models** > *Co-Simulation Models*	
Example Node Path	**Continua** > **Physics 1** > **Models** > **WAVE**	
Requires	Physics Models: *Space*: **Three Dimensional** *Time*: one of **Implicit Unsteady**, **PISO Unsteady** *Material*: **Multi-Component Gas** (**Non-reacting**) *Flow*: any *Optional Models*: **Co-Simulation**	
Activates	Boundary Inputs	**External Code Coupling Specification**. See Boundary Settings.
	Solvers	**WAVE Co-Simulation**.
	Other Nodes	**Co-Simulations** > **Link 1**. See WAVE Link Reference.

Boundary Settings

Only available for boundaries of type **Mass Flow Inlet**.

External Code Coupling Specification

Allows you to assign the boundary to a co-simulation zone for coupling. During the co-simulation, the STAR-CCM+ boundary exchanges data with the corresponding coupled surface in the partner simulation, according to the co-simulation zone specifications.

Contents:

WAVE Link Reference

WAVE Link Reference

In STAR-CCM+, you specify the settings for co-simulation with WAVE in a co-simulation link under the **Co-Simulations** node.

A co-simulation link node, **Co-Simulations** > **Link 1**, appears in the simulation when you activate the **WAVE** co-simulation model. This node allows you to define co-simulation global settings, co-simulation zones, and single-value data exchange specifications.

This section describes the global settings available for co-simulation with WAVE. For details on the co-simulation zone settings, see Co-Simulation Zones Reference.

Link Properties

Co-Simulation Type

Displays which partner code is coupled to STAR-CCM+ in the co-simulation analysis represented by this link. Read-only.

Link Conditions and Values

You specify the general settings of the co-simulation under the **Link** > **Conditions** and **Link** > **Values** nodes.

Exported Fields, Imported Fields

These nodes allow you to specify any single-value data for import or export. See Exchanging Single-Value Data .

WAVE Control Pin Associations

Allows you to associate STAR-CCM+ auxiliary field specifications and WAVE control links. For a field specification, you specify the corresponding WAVE control link using the *Associated ID* property.

STAR-CCM+ stores the imported field data in field functions that appear under the **Tools** > **Field Functions** node. For each imported field specification, STAR-CCM+ adds a field function named `Wave Control Pin #`, where `#` is the *Associated ID*.

WAVE Version

Allows you to specify which version of WAVE you couple with STAR-CCM+. You can specify a version of WAVE different from the ones suggested by selecting **Custom**. Specify the custom version using the corresponding **Values** > **Custom Code Version** node.

For a list of all supported versions of WAVE, refer to the section *External Packages Support* of the STAR-CCM+ Release Notes.

Coupled Species

Allows you to create **Species** nodes to couple the fluid components defined in WAVE with the gas components and passive scalars defined in STAR-CCM+. See Coupled Species Reference.

Launch Command Line

Specifies the command to launch WAVE.

RSimlink Libraries Folder

Specifies the path to the folder containing the **WAVE** co-simulation libraries. The libraries consists of two files:

On Linux, `librsl.so` and `libru.so`.
On Windows, `rsl.dll` and `ru.dll`.

WAVE

Displays the status of the connection between STAR-CCM+ and WAVE.

WAVE Input File

Specifies the location and name of the WAVE simulation file. The WAVE input file and WAVE installation must be located on the same filesystem.

Link Right-Click Actions

Connect

Starts the connection, coupling the STAR-CCM+ simulation with the running partner simulation that you specified.

Disconnect

Terminates the connection between STAR-CCM+ and WAVE.

Contents:

Co-Simulation Zones Reference
Coupled Species Reference

Co-Simulation Zones Reference

Co-simulation zones allow you to couple STAR-CCM+ boundaries with the partner WAVE CFD orifices.

Zones Manager Properties

Co-Simulation Zones

Displays the number of co-simulation zones defined for this link. Read-only.

Zones Manager Right-Click Actions

New

Adds a co-simulation zone under the `Zones` node. For coupling to be successful, the name of a co-simulation zone must match the name of the corresponding WAVE CFD orifices.

Zone Properties

Index

Displays a unique identifier for this zone. Read-only.

Zone Type

Specifies what type of STAR-CCM+ objects are assigned to this co-simulation zone. The available option is **Surface Zone**, which allows you to couple STAR-CCM+ boundaries to their surface counterparts in the partner simulation (surface-to-surface coupling).

Zone Conditions and Values

Coupled Model Parts

Displays the parts on the STAR-CCM+ model that exchange data with the partner simulation through this co-simulation zone. As this type of co-simulation only supports surface-to-surface coupling, the coupled model parts can only be boundaries.

This property is read-only. To assign a STAR-CCM+ boundary to a co-simulation zone, use the **External Code Coupling Specification** boundary condition.

Fluid Link

Displays the settings that define the coupled WAVE CFD orifice:

Fluid Link ID: specifies the *Interface Number* of the coupled **External CFD** orifice defined in WAVE.
WAVE Boundary Area: displays the area of the CFD orifice that is specified in WAVE (read-only).
WAVE Discretization Length: displays the length of the sampling volume across which STAR-CCM+ averages the computed fields before sending them to WAVE. The sampling volume length is defined in WAVE (read-only).
WAVE Inside Area: read-only.

Coupled Species Reference

When you model multi-component gases, the fluid components defined in WAVE must have corresponding components in STAR-CCM+. You define this association under the **Coupled Species** node.

Coupled Species Manager Right-Click Actions

New Coupled Species

Creates a **[Species]** node, that you use to couple WAVE fluid components with the corresponding STAR-CCM+ fluid components. The number of coupled species must match the number of WAVE species.

Species Properties

Mixture Component/Passive Scalar

Allows you to couple a STAR-CCM+ gas component or passive scalar with a WAVE species. The name of the **Species** node must match the name of the species defined in WAVE.

RELAP Coupling

The RELAP Coupling feature provides the ability for two-way transfer of data between three-dimensional STAR-CCM+, and one-dimensional RELAP. For example, the natural convection currents in coolant fluid can be analyzed in detail in STAR-CCM+, with RELAP providing the boundary conditions.

This provides increased detail in specified areas of the RELAP case, by adding a three-dimensional element to a one-dimensional simulation. Similarly, processing times can be reduced in a STAR-CCM+ simulation, by utilizing the speed of one-dimensional codes.

Note: RELAP is only available on Linux. STAR-CCM+ can run on Windows or Linux and connect to RELAP running on Linux.

Contents:

What is RELAP Coupling?
What is a RELAP Zone or Flow Zone?
Preparing the RELAP Case
Setting Up the STAR-CCM+ Simulation
Coupling STAR-CCM+ to RELAP

What is RELAP Coupling?

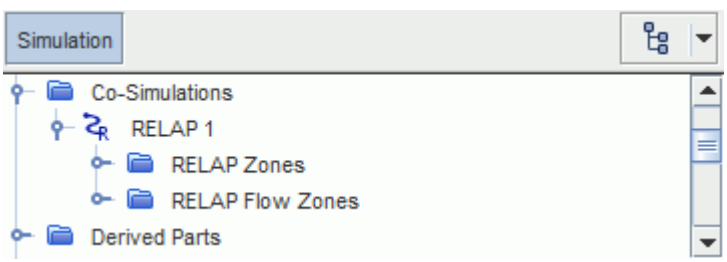
RELAP coupling provides a means of data exchange between three-dimensional STAR-CCM+ and one-dimensional RELAP. This allows for a compromise between the speed of one-dimensional code, with the detail of three-dimensional code. The feature provides optimum results when coupling occurs at a flow zone where the flow is expected to be largely one-dimensional. This recommendation does not apply to zones, as they apply thermal data to a wall boundary.

When running the solver, the flow or thermal data received from RELAP are applied at the coupled STAR-CCM+ boundary. If there is outflow at the boundary, the temperature is averaged at the flow zone, and sent back to RELAP. The STAR-CCM+ solver typically uses a smaller time step than RELAP. Data is exchanged at each RELAP time step, and the boundary conditions for STAR-CCM+ are interpolated between these time steps.

The following table summarizes how STAR-CCM+ and RELAP boundaries couple to one another:

STAR-CCM+	RELAP
Wall	Heat Structure
Mass Flow	Time Dependent Volume and Junction

The RELAP node, located within the **Co-Simulations** node, becomes available when the **RELAP Coupling** model is activated. It holds subnodes, which are populated with zones and flow zones, once the simulation is coupled to RELAP.

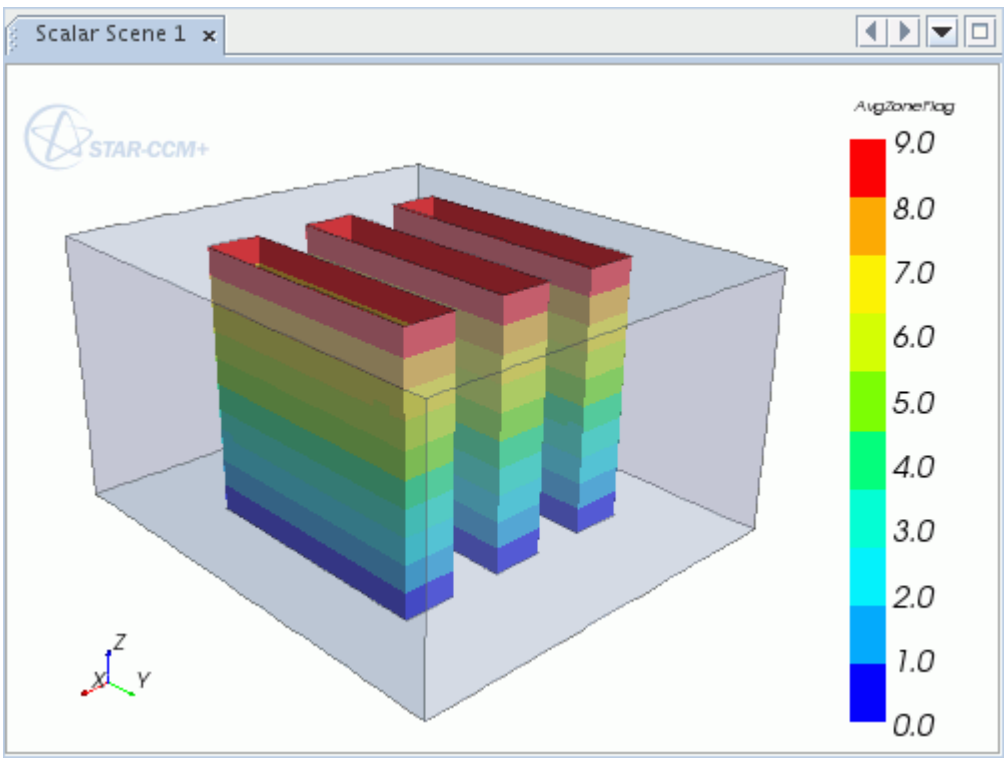


What is a RELAP Zone or Flow Zone?

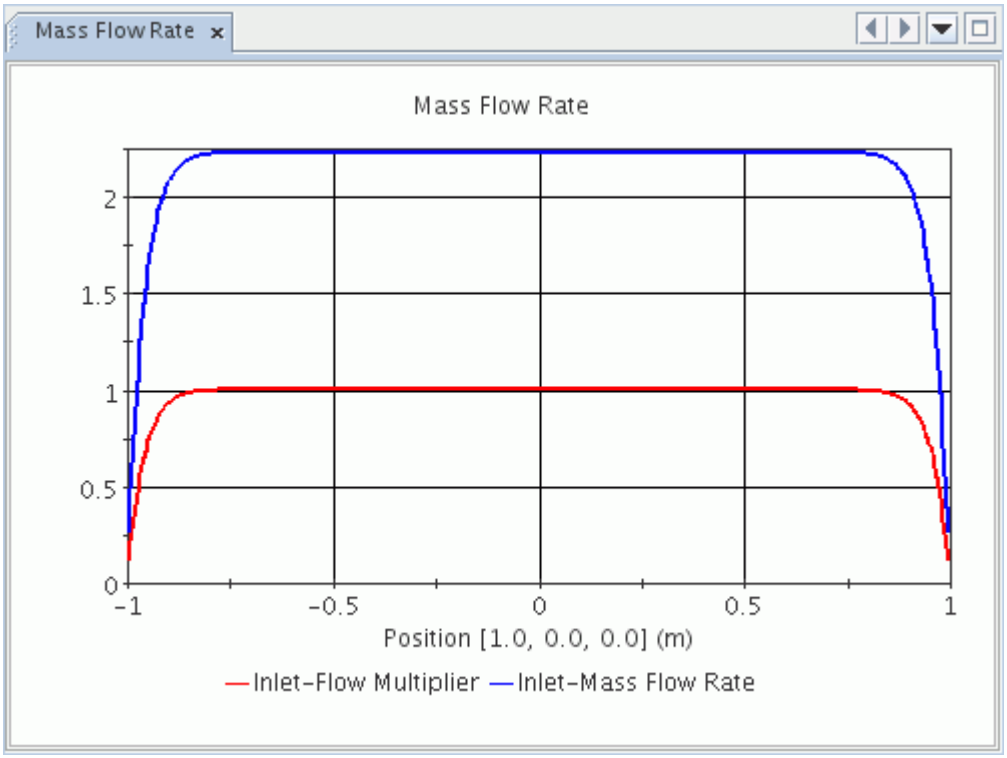
A RELAP zone or flow zone represents the equivalent of a STAR-CCM+ boundary in RELAP. Each RELAP zone and flow zone must be assigned to a coupled boundary in the STAR-CCM+ simulation.

A RELAP zone is a specified wall boundary in the CFD region, which provides a means of exchanging thermal data between STAR-CCM+ and RELAP. Similarly, a RELAP flow zone is a mass flow boundary, which provides a means of exchanging flow data between STAR-CCM+ and RELAP. The physics values at the zones or flow zones are averaged, providing a single value used to determine the equivalent boundary conditions in the one-dimensional code.

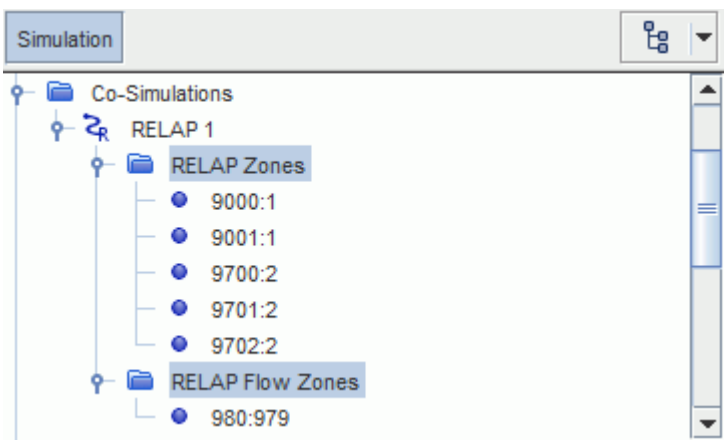
RELAP zones transfer thermal data based upon the axial discretization in RELAP. This means that multiple constant values from RELAP are applied at the wall boundary. The direction and distribution of the one-dimensional discretization is determined by the Orientation property. The RELAP discretization on the coupled boundary can be visualized as a scalar displayer using the **AvgZoneFlag** field function.



RELAP flow zones transfer flow data, with a uniform value applied at the STAR-CCM+ boundary. This value can be multiplied by a flow multiplier to modify the received value. A field function or table can also be used to multiply the received value by a variable scale factor. For example, this can be used with cell positions or wall distance to model the effect of a boundary layer at the flow boundary.



The **RELAP Zones** and **RELAP Flow Zones** nodes hold the individual zone nodes, which become available when RELAP Coupling is activated and started.



Preparing the RELAP Case

In order to couple RELAP to STAR-CCM+, you must create a `.control` file that specifies the port number and which parts of the RELAP model are to be coupled to STAR-CCM+.

This file must have the same case name as the RELAP case input file and be located in the same directory. For example, if the RELAP case input file is `casename.i`, then the file required by STAR-CCM+ must be called `casename.i.control`.

The `.control` file must follow a certain format, with an example and explanation below:

```
PORT 5555
NUMCOUPLED,5,1,0
DEBUGOUTPUT
PRESBC TDV 980 JUN 979
FLUXBC HS 9000 SIDE 1
FLUXBC HS 9001 SIDE 1
TEMPBC HS 9700 SIDE 2
TEMPBC HS 9701 SIDE 2
TEMPBC HS 9702 SIDE 2
```

`PORT 5555`	Set the port number for communication, which must match the RELAP node property.
`NUMCOUPLED,5,1,0`	Specifies the number of zones, followed by the number of flow zones. In the example, five thermal boundaries (zones) and one flow boundary (flow zone).
`DEBUGOUTPUT`	Makes RELAP write debug output when running
`PRESBC TDV 980 JUN 979`	Define a pressure boundary condition at a specified time dependant volume ID (e.g. 980) and a junction ID (e.g. 979). This will be represented by a flow zone in STAR-CCM+.
`FLUXBC HS 9000 SIDE 1`	Define a heat flux boundary condition at a specified heat structure ID (e.g. 9000) and side (e.g. 1). This will be represented by a zone in STAR-CCM+ that sends temperature data and receives heat flux.
`TEMPBC HS 9700 SIDE 2`	Define a temperature boundary condition at a specified heat structure ID (e.g. 9702) and side (e.g. 2). This will be represented by a zone in STAR-CCM+ that sends heat flux data and receives temperature.

Setting Up the STAR-CCM+ Simulation

The following section details the process of setting up a STAR-CCM+ simulation for coupling with RELAP.

Contents:

Activating the RELAP Co-Simulation Model
Setting the STAR-CCM+ Boundary

Activating the RELAP Co-Simulation Model

The **RELAP** co-simulation model is selected from within the *Physics Model Selection* dialog on the relevant continuum.

To select the **RELAP** model, the following models are required:

1. Select **Three Dimensional** from the *Space* group box.
2. Select **Implicit Unsteady** from the *Time* group box.
3. Select any material from the *Material* group box, except **Solid**.
   - If you select **Multi-Component Gas** or **Multi-Component Liquid**, select **Non-Reacting** from the *Reaction Regime* group box.
   - If you select **Eulerian Multiphase**, select **Volume of Fluid (VOF)** from the *Multiphase Flow Model* group box.
4. Select either **Segregated Flow** or **Coupled Flow** from the *Flow* group box, depending on the required solver type.

   The Segregated solver is recommended in most cases.
5. Select **Co-Simulation** from the *Optional Models* group box.
6. Select **RELAP** from the *Co-Simulation Models* group box.

To fully define the physics continua used in a RELAP coupled simulation, the following additional models must also be activated:

7. Either a **Segregated Energy** or **Coupled Energy** model from the *Optional Models* group.
8. Any model from the *Equation of State* group box.
9. **Turbulent** from the *Viscous Regime* group box.
10. Any turbulence model from the *Reynolds-Averaged Turbulence* group, depending on simulation requirements.

    Further models can be selected from the *Optional Physics Models* group, to meet individual simulation requirements.

Setting the STAR-CCM+ Boundary

The boundary type determines the physics values exchanged between STAR-CCM+ and RELAP. RELAP coupling is capable of transferring thermal data at walls and flow data at inlets and outlets.

The first step in setting the STAR-CCM+ boundary is to specify the boundary as coupled. For thermal data, you must specify whether the STAR-CCM+ wall boundary will send temperature and receive heat flux, or vice versa. For flow data, there is an option to multiply the received data by a scale factor.

Contents:

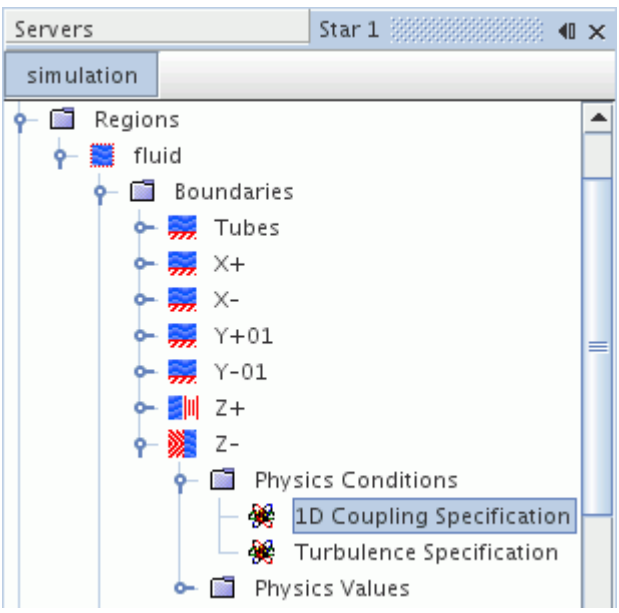
1D Coupling Specification
Coupling a STAR-CCM+ Wall Boundary to a RELAP Heat Structure
Coupling a STAR-CCM+ Mass Flow Boundary to a RELAP Time Dependant Volume and Junction
Flow Multiplier

1D Coupling Specification

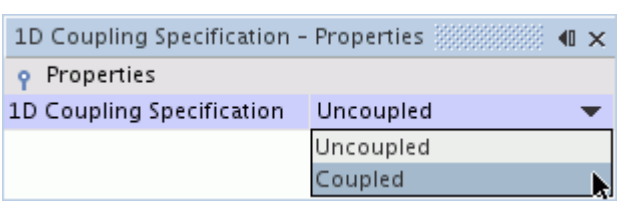
The *1D Coupling Specification* property determines whether the boundary in question is coupled to RELAP.

To set a boundary as coupled:

1. Select the **1D Coupling Specification** node, within the **Physics Conditions** node.



2. In the *Properties* window, set *1D Coupling Specification* to **Coupled**.



Coupling a STAR-CCM+ Wall Boundary to a RELAP Heat Structure

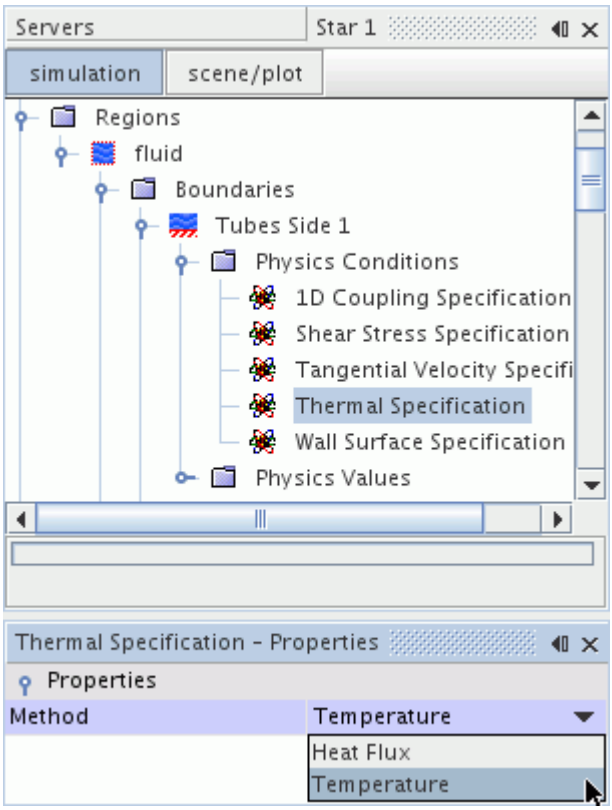
To transfer thermal data, a STAR-CCM+ wall must be assigned to a RELAP heat structure at a zone.

There are then two coupling options available:

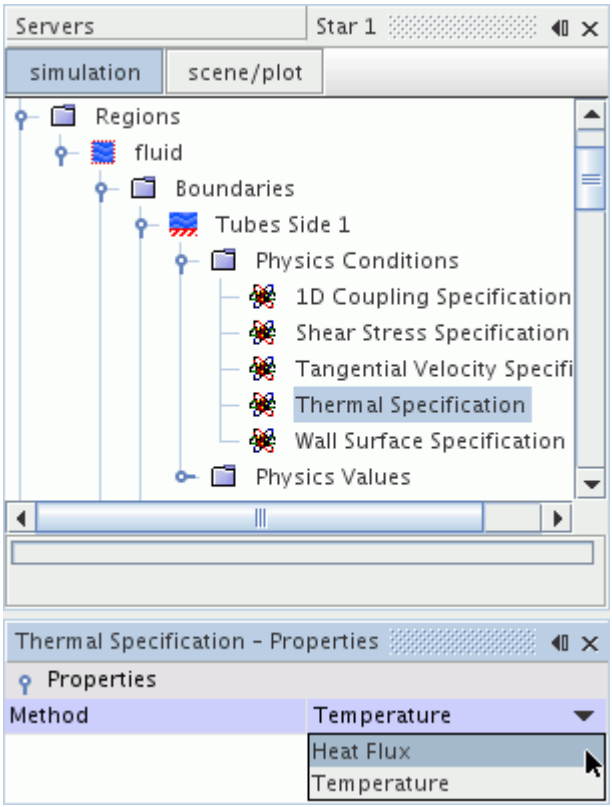
- STAR-CCM+ receives temperature and sends heat flux.
- STAR-CCM+ receives heat flux and sends temperature.

To couple a STAR-CCM+ wall boundary to a RELAP heat structure:

1. Select the STAR-CCM+ boundary **Physics Conditions** > **Thermal Specification** node.
   - To set STAR-CCM+ to receive temperature data, set *Method* to **Temperature**.



   - To set STAR-CCM+ to receive heat flux data, set *Method* to **Heat Flux**.



2. In the `casename.i.control` file (see Preparing the RELAP Case):
   - If you set STAR-CCM+ to receive temperature, set RELAP to receive heat flux by specifying the `FLUXBC` boundary condition.
   - If you set STAR-CCM+ to receive heat flux, set RELAP to receive temperature by specifying the `TEMPBC` boundary condition.

Coupling a STAR-CCM+ Mass Flow Boundary to a RELAP Time Dependant Volume and Junction

To transfer flow data, a STAR-CCM+ mass flow boundary must be assigned to both a RELAP time dependant volume and a junction, at a flow zone.

A summary of the data transferred at a flow zone is included below:

- Pressure sent from STAR-CCM+ to the RELAP time dependant volume.
- Mass flux sent from the RELAP junction to STAR-CCM+.
- If there is inflow into STAR-CCM+, temperature is sent from the RELAP time dependant volume to STAR-CCM+.
- If there is outflow from STAR-CCM+, temperature is sent from STAR-CCM+ to the RELAP time dependant volume.

The majority of boundary conditions at a flow zone are governed by RELAP, except turbulence. Therefore, the boundary is set as coupled, with turbulence specification required.

Flow Multiplier

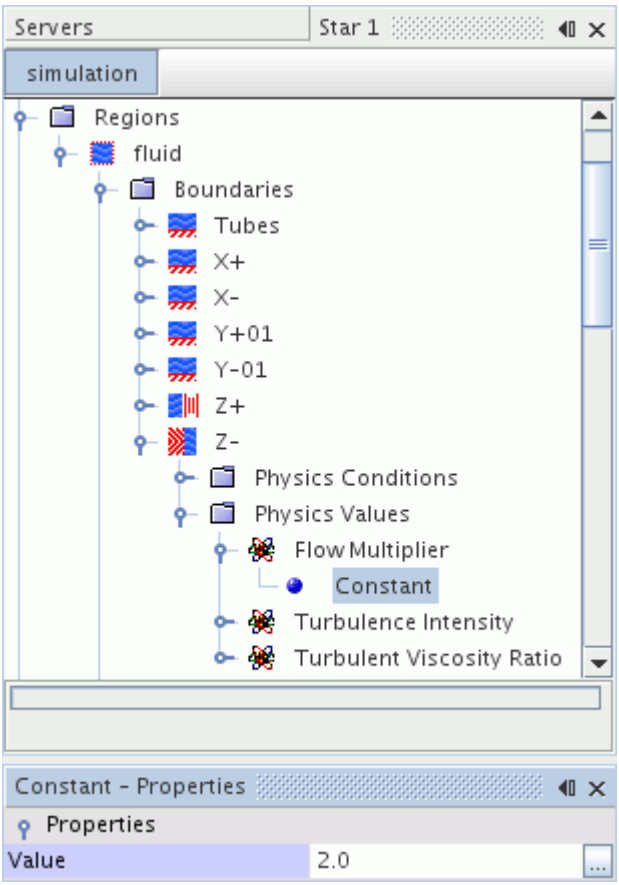
The *Flow Multiplier* allows you to multiply the data received from RELAP at the coupled boundary with a scale factor. For example, if the RELAP simulation modeled one tube, but the STAR-CCM+ simulation models two tubes, you could use a multiplier of 2 to double the received flow data at the coupled boundary.

To set a flow multiplier:

1. Select the **Physics Values** > **Flow Multiplier** node and set the *Method*.

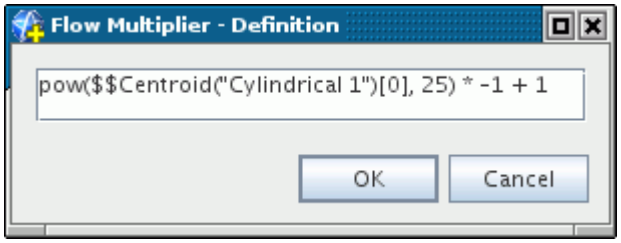
   The *Method* menu allows you to select **Constant**, **Field Function**, **Tabular** or **User Code** input.

2. Select the sub-node of the **Flow Multiplier** node and enter the required input in the *Properties* window. The example below shows use of the **Constant** option to double the received data.

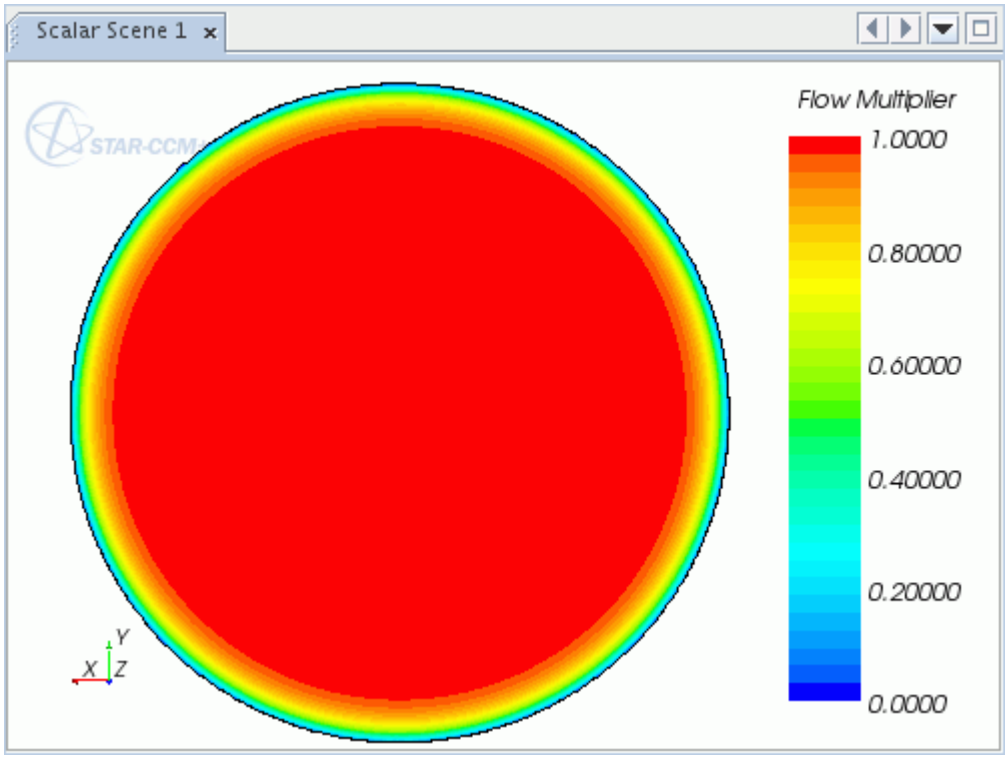


The main use of this property is to take the one-dimensional, uniform flow data, and provide a parabolic profile at the coupled boundary, which takes the boundary layer at the wall into consideration. As an example, you can use a field function to multiply the data by a varying scale factor dependant on cell position.

The following field function uses the cell centroid in the radial direction, relative to a cylindrical coordinate system located at the centre of a pipe. This value is put to the power of 25 to give a parabolic shape, representative of an arbitrary boundary layer. The profile is then flipped and transposed so that the majority of the flow is given a flow multiplier of approximately 1.



This example profile is illustrated below by displaying the flow multiplier field function on the inlet of a cylindrical pipe:



A similar effect can also be achieved on a per-cell basis by using tabular data with reference to cell IDs.

Coupling STAR-CCM+ to RELAP

This section details the process for coupling to RELAP.

Contents:

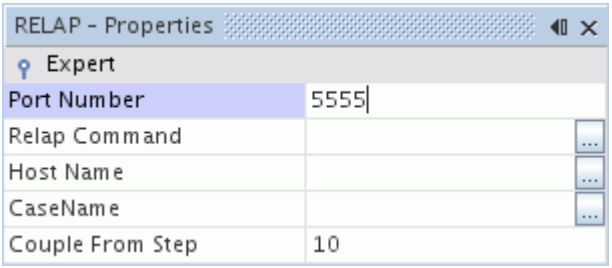
Setting the Port Number
Setting the RELAP command
Setting the Host Name
Setting the Case Name
Starting and Coupling to RELAP
Setting the Couple From Step Option
Assigning the RELAP Zones and Flow Zones
Setting RELAP Zone Properties

Setting the Port Number

The *Port Number* property must be set to the same port number specified in the `casename.i.control` file. This is the port at which RELAP and STAR-CCM+ will communicate.

To set the port number:

1. Select the **Co-Simulations** > **RELAP** node.
2. In the *Properties* window, set the *Port Number*.

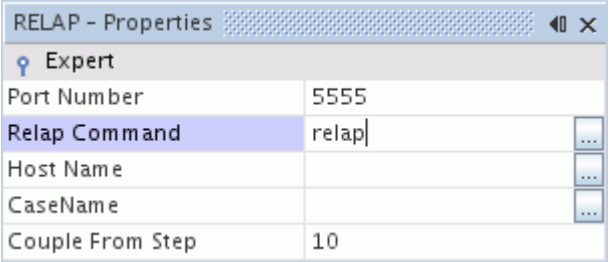


Setting the RELAP command

The *Relap Command* property must be set to the command that is used to execute RELAP in a Linux shell. This can also be set as the absolute path to the RELAP executable, or as a relative path from the current working directory.

To set the command to launch RELAP: Then,

1. Select the **Co-Simulations** > **RELAP** node.
2. In the *Properties* window, set *Relap Command*.

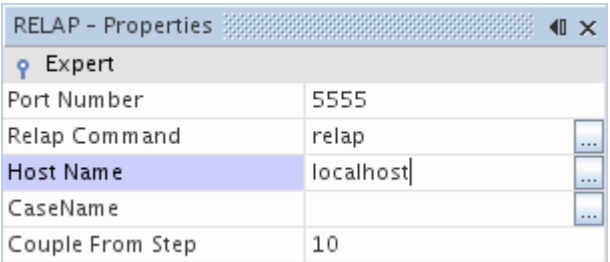


Setting the Host Name

The *Host Name* property must be set to the name of the machine where RELAP will be launched. If this is the same machine as the STAR-CCM+ client, the property should be set to `localhost`. This will allow STAR-CCM+ to launch RELAP automatically. If the RELAP machine is different to the STAR-CCM+ machine, the exact name of the machine on the network should be used. In this case, RELAP must be started on the host manually.

To set the host name:

1. Select the **Co-Simulations** > **RELAP** node.
2. In the *Properties* window, set *Host Name*.

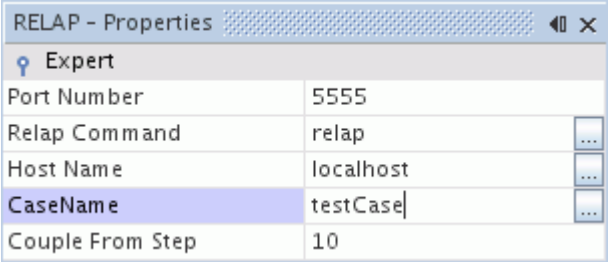


Setting the Case Name

The *Case Name* property must be set to the case name of the RELAP case file, (i.e. the same as `casename` in the `casename.i.control` file).

To set the case name:

1. Select the **Co-Simulations** > **RELAP** node.
2. In the *Properties* window, set the *Case Name*.



Starting and Coupling to RELAP

Once the simulation is fully defined, the STAR-CCM+ simulation can be coupled to RELAP. When coupling is started, RELAP reads the `casename.i.control` file and sends the zones and flow zones to STAR-CCM+. The zones and flow zones can then be assigned to coupled boundaries.

To start and couple to RELAP:

1. Right-click the **Co-Simulations** > **RELAP** node and select **Start RELAP**.

   When complete, the *Output* window will display `RELAP Connection Initialized`. There will also be a summary of the RELAP zones added to the simulation.

RELAP can be stopped at any time. To stop RELAP:

2. Click **Solution** > **Clear Solution**.

   This will break the connection between STAR-CCM+ and RELAP. RELAP can then be restarted using the methods mentioned above, which will reset the coupling and take into account any changes made to the RELAP case. Closing STAR-CCM+ will also close RELAP.

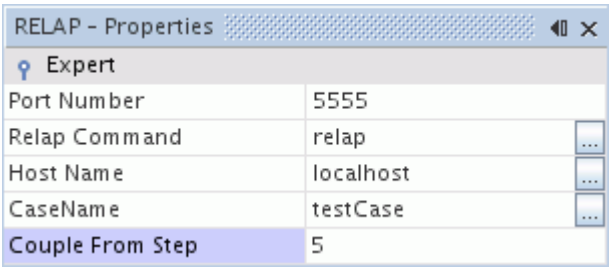
Setting the Couple From Step Option

The *Couple From Step* option allows the transfer of data from STAR-CCM+ to RELAP to be delayed until a defined time step. This allows the STAR-CCM+ solver to reach a level of stability whilst receiving data from RELAP, before sending data back to RELAP. Note that, although data is not transferred to RELAP, the codes are both advancing at the same time. During this period of one-way transfer, the RELAP analysis is running with constant boundary conditions at the coupled boundary.

Using the *Couple From Step* option is not typically recommended. It is only required if the solver diverges during the initial stages, and you are unable to obtain a solution.

To set the *Couple From Step* option:

1. Select the **Co-Simulations** > **RELAP** node.
2. In the *Properties* window, set *Couple From Step*.



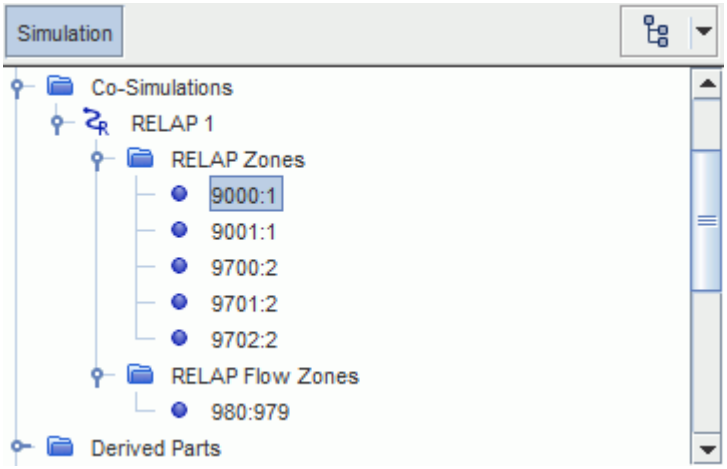
Assigning the RELAP Zones and Flow Zones

All RELAP zones and flow zones must be assigned to a STAR-CCM+ boundary, with the correct boundary type and conditions. This provides a means of identifying the location of data exchange between the one-dimensional and three-dimensional code.

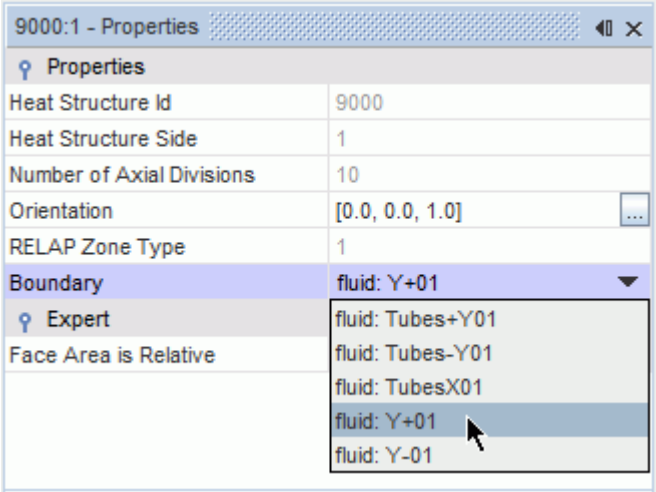
The node names are based upon data received from RELAP. Zones are named “Heat Structure ID : Side ID” (e.g. 9000:1) and flow zones are named “Time Dependant Volume ID : Junction ID” (e.g. 980:979). This allows you to identify the part of the RELAP model to which the node refers, so that the appropriate boundary can be assigned.

To assign a zone or flow zone:

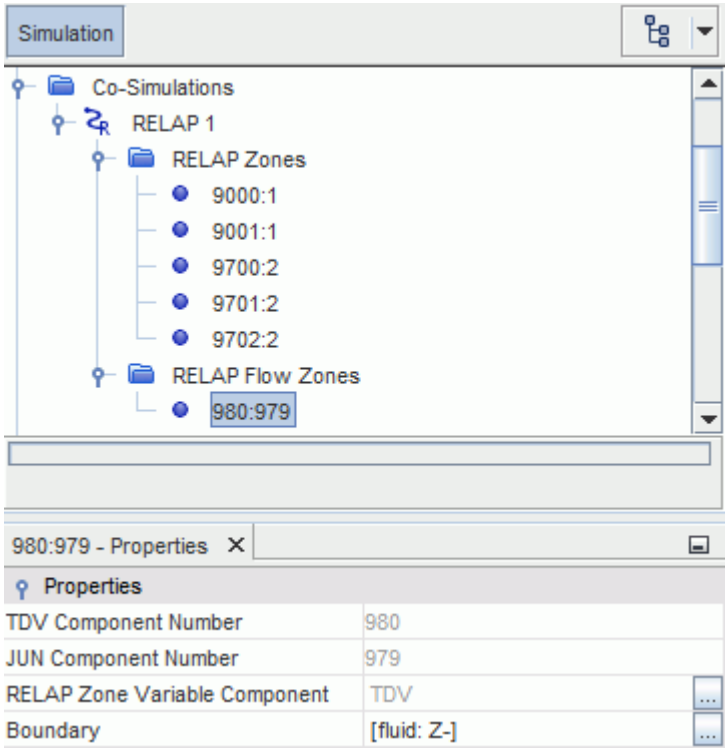
1. Select a sub-node of the **RELAP Zones** or **RELAP Flow Zones** node.



2. In the *Properties* window, assign the boundary using the *Boundary* drop-down menu.



For comparison, the image below shows boundary assignment in the **RELAP Flow Zone** properties window.

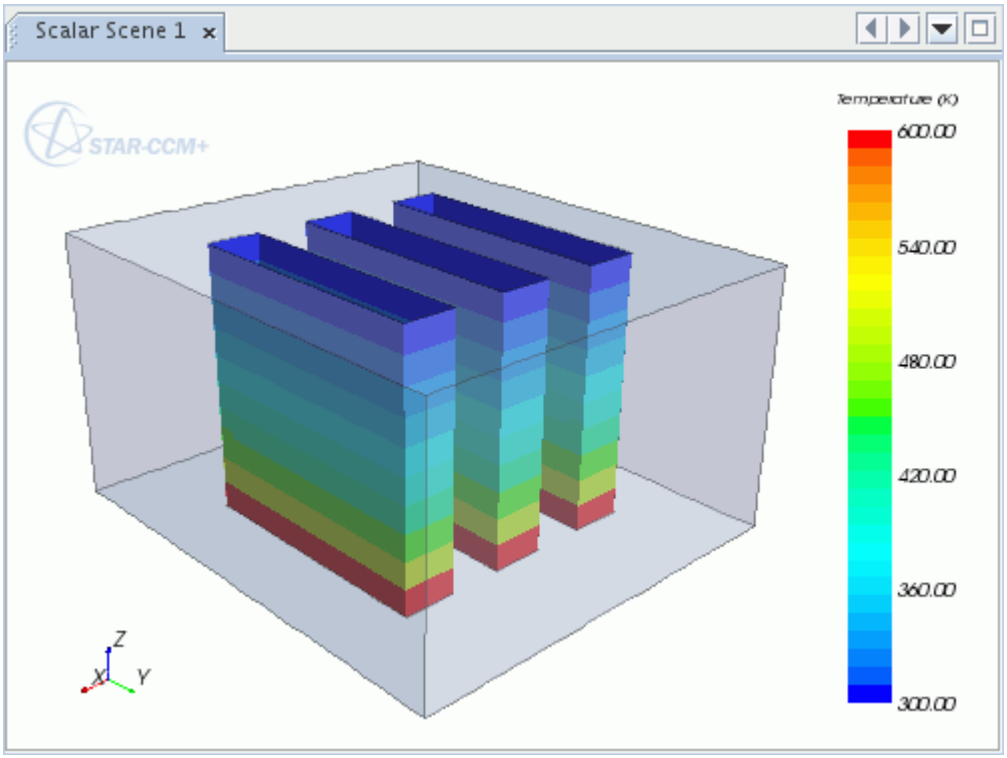


Setting RELAP Zone Properties

The RELAP zone properties provide two user-defined settings: *Orientation* and *Face Area is Relative*.

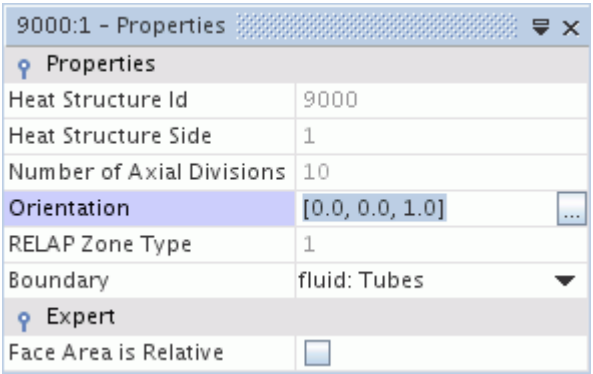
The *Orientation* property determines the direction of the axial discretization from RELAP. The axial discretization allows STAR-CCM+ to receive a different temperature or heat flux value from each part of the discretization in RELAP, which provides a gradient along the wall in the specified direction. For example, the

following image shows a temperature distribution on the coupled boundary as a result of the RELAP discretization.



To set the orientation of the axial discretization:

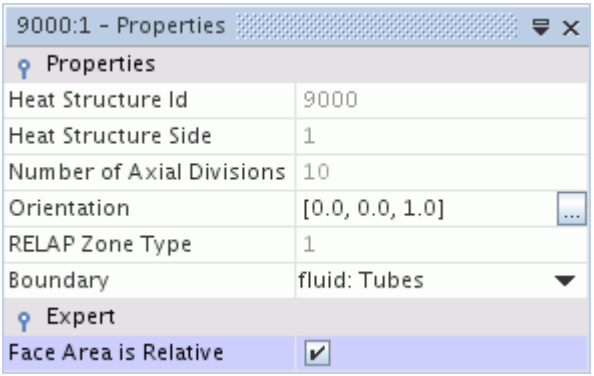
1. Select a RELAP zone within the **RELAP Zones** node and set *Orientation* with the x, y and z components of a direction vector.



In some cases, the surface area of the STAR-CCM+ wall boundary may not match the area of the RELAP heat structure. In this instance, you should activate the *Face Area is Relative* option so that the STAR-CCM+ wall boundary is considered relative to the RELAP heat structure. This will apply the necessary scale factor to the values received at the coupled boundary.

To activate the *Face Area is Relative* option:

2. Select a RELAP zone within the **RELAP Zones** node and activate *Face Area is Relative*.

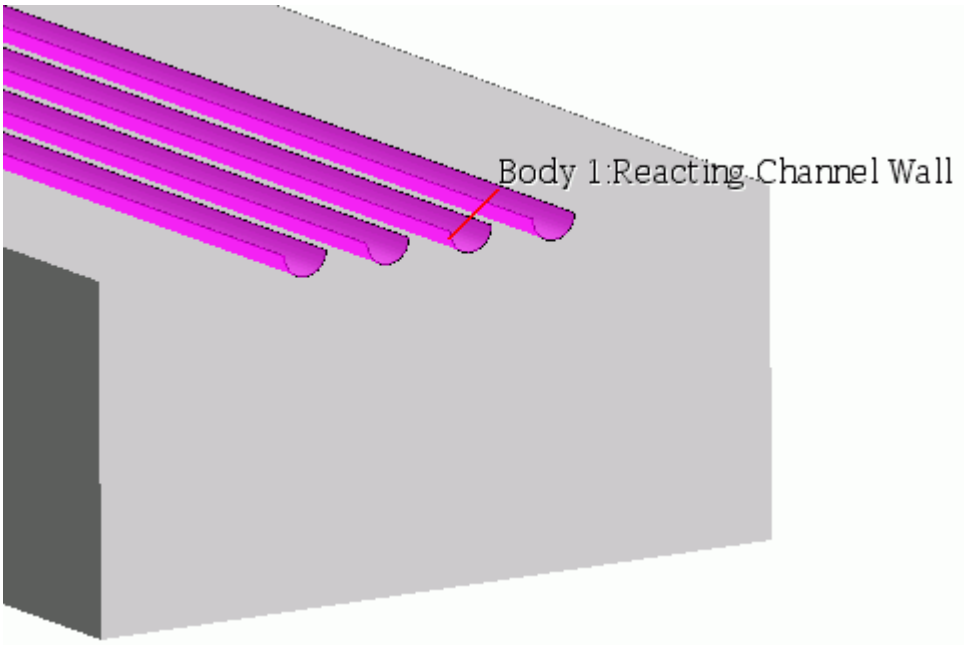

9000:1 - Properties
Properties
Heat Structure Id
9000
Heat Structure Side
1
Number of Axial Divisions
10
Orientation
[0.0, 0.0, 1.0]
RELAP Zone Type
1
Boundary
fluid: Tubes
Expert
Face Area is Relative

Reacting Channel Coupling

The Reacting Channel Coupling feature provides the ability for exchange of data between three-dimensional STAR-CCM+ and the one-dimensional Plug Flow Reactor in DARS-CFD. You can use the **Reacting Channel** co-simulation model in STAR-CCM+ to solve reacting flow simulations in tubular heat exchangers with channels that are long and thin.

When the **Reacting Channel** model is selected along with the **Co-Simulation** model, a **Co-simulations** > **Reacting Channel Co-Simulation** node appears.

Several geometrical channels can represent one reacting channel as long as all of the geometrical channels have identical geometries of the same orientation. For geometrical channels that differ in length, diameter, or orientation, define each different group of geometrical channels as a separate reacting channel.



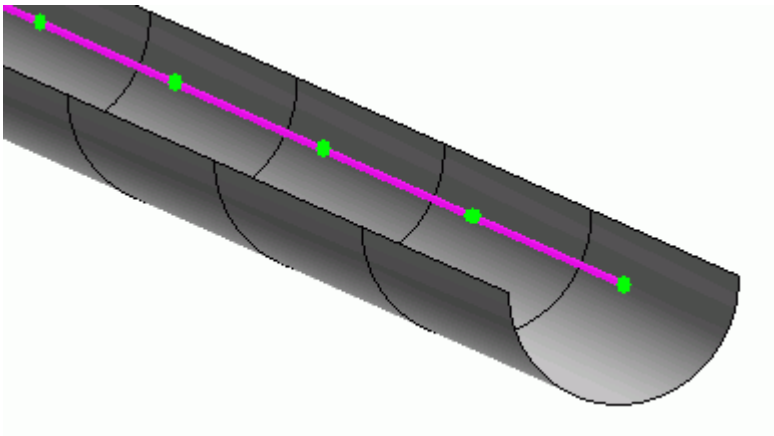
This functionality is useful for simulating chemical processes such as steam reforming and firebox combustion.

STAR-CCM+ meshes and solves the volume of the outer flow. The reacting channels are solved using a Plug Flow Reactor (PFR) approximation. The PFR is part of the DARS-CFD package. The reacting channels are not meshed and are considered 1D. However, the outer flow volume must account for the presence of the reacting channels in its geometry.

The coupling happens on the reacting channel wall boundary between the outer flow in STAR-CCM+ and the reacting channel in the PFR. The temperature is exchanged from the outer flow to the reacting channel and

heat flux is returned back from the reacting channel to the outer flow. Although reacting channels appear as 3D features in a geometry, they become 1D representations when defined as reacting channels in the Reacting Channel Coupling model.

Since the reacting channels are one-dimensional and have no mesh, data is exchanged between axial points, whose frequency you specify, along a vector through the centre of the reacting channel and the surface boundary of the outer flow, as represented by the diagram below:



Note: It is imperative that all geometrical channels that are defined within a single reacting channel, are the same length and diameter with the same properties.

Reactions that occur in the reacting channel can consume or create heat energy. At specified time intervals, STAR-CCM+ sends the temperature along the geometrical channel wall to the Plug Flow Reactor (PFR). The PFR then solves for the reacting flow, determines the temperature within the reacting flow, and calculates the heat flux at each axial point in the reacting flow channel. The PFR then sends this data back to the outer flow in STAR-CCM+ . See Eqn. (1405).

Contents:

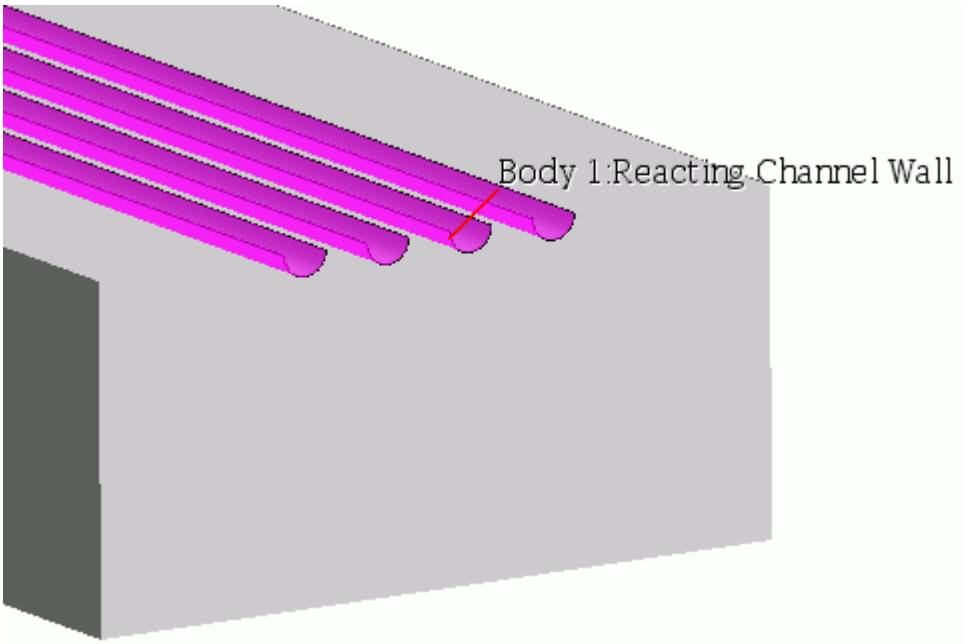
Modeling Reacting Channels
Reacting Channel Coupling Formulation
Reacting Channel Coupling Nomenclature

Modeling Reacting Channels

The **Reacting Channel** co-simulation model in STAR-CCM+ allows you to exchange data between reacting channels that use the DARS-CFD one-dimensional Plug Flow Reactor and three-dimensional STAR-CCM+.

Before setting up reacting channels, make sure that you have a fluid domain with a three-dimensional mesh that contains unmeshed voids that represent the reacting channels.

When using the reacting channel coupling model, set up at least one reacting channel co-simulation zone. Each reacting channel can represent several geometrical channels.



To activate the **Reacting Channel** model:

1. Set up a steady state three-dimensional flow simulation using either the segregated or coupled flow solvers. Include an energy model in the selection.
2. Open the *Physics Model Selection* dialog. See Using the Physics Model Selection Dialog.
3. Deactivate **Auto-select recommended models** then select the specific models that you require for the simulation, including the following models:

Group Box	Model
Optional Models	**Co-Simulation**
Co-Simulation Models	**Reacting Channel**

4. Click **Close**.

   STAR-CCM+ adds the **Co-Simulations** node and a single **Reacting Channel Co-Simulation**.

Within one simulation, it is possible to create more than one reacting channel co-simulation. Separate reacting channel co-simulations are necessary for each reacting channel mechanism that is required. Each reacting

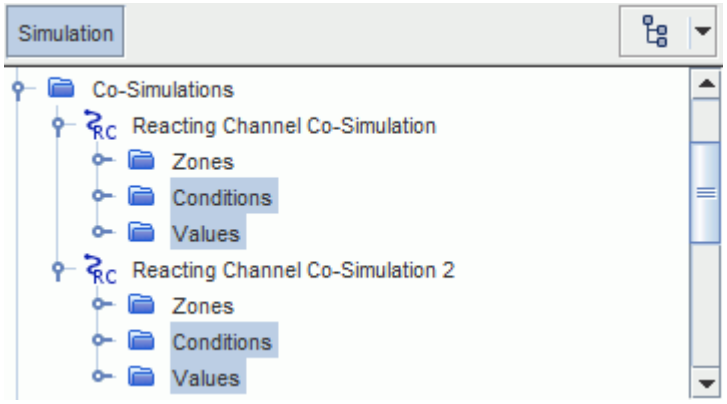
channel co-simulation must contain at least one reacting channel zone. You can specify reaction conditions and values for each reacting channel co-simulation and also define further conditions and values that are specific to each reacting channel zone.

5. To specify more than one reacting channel mechanism, right-click the **Co-Simulations** node and select **New**.

   A **Reacting Channel Co-Simulation [2]** node appears.

6. Create a **Reacting Channel Co-Simulation** node for each reacting channel mechanism that you want to use.

7. Specify the Conditions and Values for each Reacting Channel Co-Simulation.



   In particular, for each reacting channel co-simulation, import the reacting channel mechanism.

   a) Select the **Reacting Channel Co-Simulation** > **Conditions** > **PFR Library Input Option** node and specify the *Input Type* as **Binary** or **ASCII**, depending on the format of the input file.

      See Reacting Channel Co-Simulation Conditions and Values. To set up a complex chemistry definition, see DARS-CFD Properties.

   b) Expand the **Reacting Channel Co-Simulation** > **Values** > **PFR Library** node.

   c) In the *Properties* window, click the Custom Editor for *PFR Input* and in the dialog that appears, select the reaction mechanism file then click **OK**.

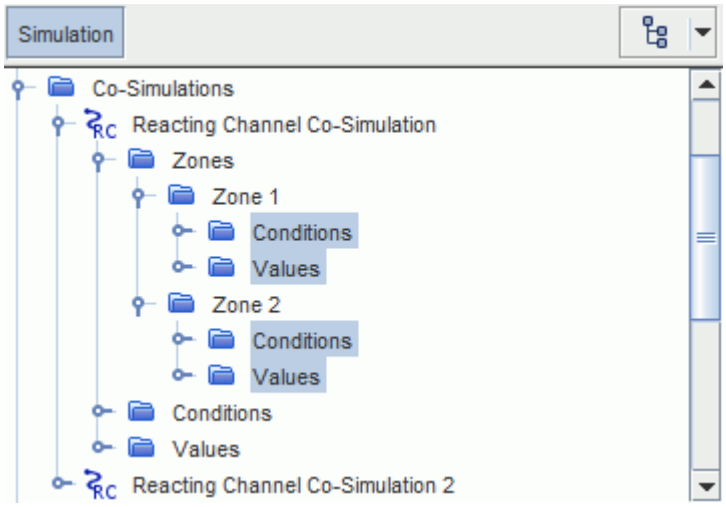
   d) Select the **Reacting Channel Co-Simulation** > **Values** > **Coupling Strategy** node and specify the parameters that are required for coupling.

The **Zones** node contains a **Zone 1** node by default. To create more reacting channel zones:

8. Right-click the **Reacting Channel Co-Simulation** > **Zones** node and select **New**.

   A **Zone [2]** node appears.

9. Specify the Conditions and Values for each Reacting Channel Co-Simulation Zone.



In particular:

a) Select the **[Zone]** > **Conditions** > **Reacting Channel Type** node and specify the *Correlation Type* as **Pipe** or **Packed Bed**.

b) Select the **[Zone]** > **Conditions** > **Heat Transfer Coefficient Correlation** node and choose the appropriate *Correlation Type*.

c) Set other **[Zone]** > **Conditions** as necessary.

See Reacting Channel Co-Simulation Zones.

Each zone represents one reacting channel and contains a set of conditions and values which are applied to the boundaries that you specify for that reacting channel zone. To specify the boundaries for a reacting channel zone:

10. Group the boundaries for each reacting channel zone:
    a) Right-click the **Regions** > **[Region]** > **Boundaries** node and select **New Group**.
    b) Rename the **New Group** node appropriately.
    c) Expand the **Boundaries** node and multi-select the boundaries for one reacting channel zone.
    d) Drag the selected boundaries on to the **[New Group]** node.

    You can use the **[New Group]** node to edit the properties, conditions, and values of all boundaries that it contains.
11. Right-click the **[New Group]** node and select **Edit**.
12. In the *Multiple Objects* dialog, expand the **Physics Conditions** > **External Code Coupling Specification** node and set *External Code Coupling Specification* to the Reacting Channel Co-Simulation Zone for the selected boundaries.
13. Group the remaining boundaries as necessary and select an appropriate zone for the **External Code Coupling Specification** for each group of boundaries.

Contents:

Reacting Channel Co-Simulation Zones
Reacting Channel Co-Simulation Conditions and Values

Reacting Channel Co-Simulation Zones

When the Reacting Channel model is selected, a set of specific conditions and values are available to specify for each zone within the **Co-Simulations** > **Reacting Channel Co-Simulation** > **Zones** folders.

For each reacting channel Zone, there is a set of conditions and values that you can specify.

The following Conditions are available to specify for each reacting channel zone:

- Coupled Model Parts
- Heat Transfer Coefficient Correlation
- Pipe Friction Correlation
- Reacting Channel Heat Transfer Coefficient Option
- Reacting Channel Inlet Viscosity Option
- Reacting Channel Pipe Friction Factor Option
- Reacting Channel Type
- Reacting Channel Wall Temperature Option

Coupled Model Parts

Boundaries	Displays a read only value which contains the boundaries that are specified for that reacting channel zone.
Regions	Displays a read only value which contains the regions that are specified for that reacting channel zone.

Heat Transfer Coefficient Correlation

Correlation Type	Allows you to specify the correlation type for the heat transfer coefficient.	
	Gnielinski (Pipe)	Determines the heat transfer coefficient using Eqn. (1392)
	Leva/Grummer (Packed Bed)	Determines the heat transfer coefficient using Eqn. (1396)
	Beek (Packed Bed)	Determines the heat transfer coefficient using Eqn. (1398)
	DeWasch/Froment (Packed Bed)	Determines the heat transfer coefficient using Eqn. (1399)

Pipe Friction Correlation

Correlation Type	Allows you to specify the correlation type for the pipe friction coefficient.	
	Blasius (Pipe)	Determines the pipe friction coefficient using Eqn. (1400)
	Filonenko (Pipe)	Determines the pipe friction coefficient using Eqn. (1401)

	Ergun (Packed Bed)	Determines the pipe friction coefficient using Eqn. (1403) which is valid for simulations with low $\frac{\mathrm{Re}}{1-\chi}$ values.
	Hicks (Packed Bed)	Determines the pipe friction coefficient using Eqn. (1402) which is valid for simulations with high $\frac{\mathrm{Re}}{1-\chi}$ values.
	Ergun/Hicks (Packed Bed)	Determines the pipe friction coefficient using Ergun correlation (Eqn. (1403)) when $\frac{\mathrm{Re}}{1-\chi} < 300$, Hicks correlation (Eqn. (1402)) when $\frac{\mathrm{Re}}{1-\chi} > 500$, and linear interpolation in between.

Reacting Channel Heat Transfer Coefficient Option

Reacting Channel Heat Transfer Coefficient Option	Allows you to specify the method for determining the Heat Transfer Coefficient, h_f which is used to determine the convective heat transfer source $\dot{Q}_C$ in Eqn. (1391).	
	Constant	Adds a **Values** > **Heat Transfer Coefficient** node which allows you to specify the Heat Transfer Coefficient, h_f as a constant array value.
	Uniform	Allows you to specify the Heat Transfer Coefficient using values which are taken directly from the reacting channel Inlet values.
	Non-Uniform	Determines the heat transfer coefficient using data from each axial point.
	Tabular	Adds a **Values** > **Heat Transfer Coefficient** node which allows you to specify the Heat Transfer Coefficient using tabular data which you can import.

Reacting Channel Inlet Viscosity Option

Option	Allows you to specify how the viscosity, μ, of the components entering the reacting channel is determined at the reacting channel inlet.	
	Constant	Adds a **Values** > **Inlet Viscosity** node which allows you to set a value for the gas Viscosity at the inlet.
	Uniform	Allows you to specify the gas Viscosity at the inlet using values which are taken directly from the reacting channel Inlet values.

Reacting Channel Pipe Friction Factor Option

Option	Allows you to specify a multiplying factor for the pipe friction, f which is used to calculate the Nusselt number Eqn. (1394) for cases with turbulent flow.	
	Constant	Allows you to specify a constant value for the **Values** > **Pipe Friction Factor** node.
	Tabular	Allows you to specify the **Values** > **Pipe Friction Factor** using tabular data that you can import.

Reacting Channel Type

Correlation Type	Allows you to specify the correlation type for the reacting channel.	
	Pipe	Specifies the pipes as empty reacting channels – which contain no material other than the gas that is in them.
	Packed Bed	Specifies the pipes as fixed bed / packed bed reacting channels. This option is suitable for modeling applications such as steam cracking and methane steam reforming.

Reacting Channel Wall Temperature Option

Option	Allows you specify how the Reacting Channel Wall Temperature, T_W, Eqn. (1391), is determined at the wall boundary.	
	Uniform	Determines the temperature using data that is averaged from all axial points throughout the reacting channel zone.
	Non-Uniform	Determines the temperature using data from each axial point throughout the reacting channel zone.

The following Values are available to specify for each reacting channel zone:

- **Heat Transfer Coefficient**

  Available when the reacting channel correlation type is set to **Pipe**. The convective heat transfer coefficient, h_f, is available to specify as:

  - a constant value when **Reacting Channel Heat Transfer Coefficient Option** is set to **Constant**
  - tabular data when **Reacting Channel Heat Transfer Coefficient Option** is set to **Tabular**

  This value is used to determine the convective heat transfer source, $\dot{Q}_C$, in Eqn. (1391)

- **Inlet Area**

Specify the area of one geometrical channel in a reacting channel zone (each geometrical channel within a reacting channel zone is expected to be of equivalent dimensions). This value is used to calculate the channel diameter D_C which is used to determine the convective heat transfer coefficient of the reacting channel, h_f in Eqn. (1392).

- **Inlet Mass Fractions**

  Allows you to specify the initial mass fraction profile of the components entering the reacting channel.

- **Inlet Pressure**

  Allows you to specify the pressure at the reacting channel inlet.

- **Inlet Temperature**

  Allows you to specify the reacting channel bulk temperature, T,

- **Inlet Velocity**

  Allows you to specify the velocity of the components entering the reacting channel inlet.

- **Inlet Viscosity**

  Available to specify when the **Reacting Channel Inlet Viscosity Option** is set to **Constant**. Allows you to specify the viscosity, μ , in Eqn. (1395) of the components entering the reacting channel inlet.

- **Packed Bed Heat Transfer Factor**

  Available when the reacting channel correlation type is set to **Packed Bed**. Allows you to specify the heat transfer factor, f_{htg} , which is used to calculate the heat transfer correlation for the packed bed reacting channel in Eqn. (1396) and Eqn. (1398).

- **Packed Bed Particle Diameter**

  Available when the reacting channel correlation type is set to **Packed Bed**. Allows you to specify the equivalent packed bed particle diameter, D_p , which is used to calculate the heat transfer correlation for the packed bed reacting channel in Eqn. (1396) and Eqn. (1398).

- **Packed Bed Porosity**

  Available when the reacting channel correlation type is set to **Packed Bed**. Allows you to specify the porosity, χ , of the material within the packed bed reacting channel.

- **Pipe Friction Factor**

  The pipe friction factor, f, is available to specify as:

  - a constant value when **Reacting Channel Pipe Friction Factor** is set to **Constant**.
  - tabular data when **Reacting Channel Pipe Friction Factor** is set to **Tabular**.

  This value is used to determine the Nusselt number for cases with turbulent flow, in Eqn. (1394), which is used to calculate the convective heat transfer coefficient, h_f, in Eqn. (1392).

- **Pipe Length**

  Allows you to specify the length of one geometrical channel in the reacting channel zone.

- **Solver Settings**

  Allows you to set the reacting channel solver settings for the zone.

  Properties:

Absolute Tolerance

Allows you to specify an absolute tolerance for the solver.

Relative Tolerance

Allows you to specify a relative tolerance for the Solver.

Minimum Time Step Size

Sets the minimum time step size of the solver.

Maximum Time Step Size

Sets the maximum time step size of the solver.

Limit Rates

When activated, the rates within the solver are limited to the physical solution only.

Expert Properties:

Maximum BDF Order

Specifies the maximum order of the backward differentiation formula (BDF) method which is used to solve the stiff differential equations by numerically integrating the ordinary differential equations (ODE).

Maximum Newton Steps

Specifies the maximum number of Newton steps that are taken to solve the non-linear equations at each timestep when the implicit backward differentiation formula (BDF) method is used.

- **Underrelaxation Factor**

  Allows you to set an under-relaxation factor, α, which governs the extent to which the old solution is supplanted by the newly computed solution. The default value (1.0) means that no under-relaxation is used

- **Wall Discretization**

  Allows STAR-CCM+ to receive a different temperature or heat flux value from each axial division that you specify which provides a gradient along the reacting channel in the specified orientation.

Reacting Channel Co-Simulation Conditions and Values

When simulating reacting channel co-simulation, you can specify the reacting channel conditions and values when the **Reacting Channel Co-Simulation** model is selected.

The **Reacting Channel Co-Simulation** node contains three sub-nodes which allow you to define the reacting channel zones, conditions, and values. Selecting the **Reacting Channel Co-Simulation** node also displays the following read-only property:

Co-Simulation Type	Displays the read-only value **ReactingChannel Co-Simulation**.

The following **Reacting Channel Co-Simulation** > **Conditions** sub-nodes are available:

PFR Library Input Option

Input Type	Allows you to specify the type of table that imports the reacting channel plug flow reactor data.	
	ASCII	Set *Input type* to **ASCII** when the data that you import is in ASCII format.
	Binary	Set *Input type* to **Binary** when the data that you import is in Binary format. Creates an extra Conditions sub-node called **PFR User Coding Option**.

PFR User Coding Option

User Coding	Only available to set when the Input *Type* is set to **Binary**. Allows you to modify the reaction rates or species source terms that are imported as binary data using the Dars-CFD FORTRAN files.	
	None	Does not use user coding.
	Calculate Source Term	Calculates the species source terms with the Dars-CFD FORTRAN file: SetDARSGasSrcTerms.f
	Modify Reaction Rates	Modifies the species reaction rates with the Dars-CFD FORTRAN file: ModDARSGasReacRates.f
	Modify Source Term	Modifies the species source terms with the Dars-CFD FORTRAN file ModDARSGasSrcTerms.f

Reacting Channel Species

PFR Species (read only)	Allows you to see all species that are involved in this mechanism in the reacting channel – reactants and products.

The following **Reacting Channel Co-Simulation** > **Values** sub-nodes are available:

Coupling Strategy

Couple After Step	Specifies the time-step after which data is transferred from the Plug Flow Reactor (PFR)to STAR-CCM+. Delaying the data transfer until a specified time step allows the solver to reach a level of stability while receiving data from the PFR, before sending data back to the PFR . Set to 0 as default.
Coupling Frequency	Specifies the interval of time steps that occur between each transfer of data from the PFR to STAR-CCM+. Set to 5 by default.

PFR Library

Selecting the **Reacting Channel Co-Simulation** > **Values** > **PFR Library** node allows you to set the following properties:

PFR Input	Specifies the location of the PFR files that are used for the *PFR Library Input Option* Condition.
Files Loaded	Appears as activated and read-only when PFR files are loaded.
Library Loaded	Appears as a read-only option which appears activated when a PFR library is loaded.

Right-clicking the **Reacting Channel Co-Simulation** > **Values** > **PFR Library**node provides the following options:

Export Complex Chemistry Definition Input	Allows you to save the complex chemistry definition for the reacting channel as a .txt file by default.
Delete Complex Chemistry Case	Deletes the complex chemistry definition from the reacting channel.

Reacting Channel Coupling Formulation

The Reacting Channel Coupling feature couples a 3D STAR-CCM+ simulation with the 1D Plug Flow Reactor in DARS-CFD.

Reacting Channel Plug Flow Reactor Formulation

The flow in the reacting channel is simulated using the one-dimensional Plug Flow Reactor (PFR) from DARS-CFD. At any axial point within the reacting channel, the flow variables are radially uniform. For a given reaction, the axial diffusion of heat and species are consistent and are therefore ignored. The evolution of species and temperatures in the PFR are represented as follows:

- Species:

$$\frac{\mathrm{d}Y_k}{\mathrm{dt}} = \frac{\dot{\omega}_k}{\rho} \tag{1389}$$

- Temperature:

$$\frac{\mathrm{dT}}{\mathrm{dt}} = \frac{\sum_k \dot{\omega} h_k + \dot{Q}_C}{\rho c_p} \tag{1390}$$

In the reacting channel, the convective heat transfer source, $\dot{Q}_C$, is determined using:

$$\dot{Q}_C = h_f\left(\mathrm{T_W} - \mathrm{T}\right) \tag{1391}$$

$\mathrm{T_W}$, the reacting channel wall temperature, is averaged from the 3D outer flow temperature field on the resolved reacting channel wall.

The convective heat transfer coefficient, h_f, of the reacting channel is determined depending on the type of heat transfer correlation that is specified:

- For open pipes which use the Gnielinski heat transfer correlation:

$$h_f = \frac{NuK_C}{D_C} \tag{1392}$$

  Nu is the Nusselt number which is determined according to fully developed pipe flow empirical correlations:

  - For laminar flow where Re < 3000:

$$Nu = 3.66 \tag{1393}$$

  - For turbulent flow where Re > 3000, Nu is determined using Gnielinski's correlation for turbulent flow:

$$Nu = \frac{(f/2)\,(\mathrm{Re} - 1000)\mathrm{Pr}}{1 + 12.7\sqrt{(f/2)}\left(\mathrm{Pr}^{2/3} - 1\right)} \tag{1394}$$

  Where Pr is the Prandtl number which is determined by:

$$\mathrm{Pr} = \frac{\mu\, c_p}{K_c} \tag{1395}$$

and f is the Fanning friction factor.

- For packed bed pipes that use the Leva / Grummer heat transfer correlation:

$$h_f = f_{htg} 0.813 \frac{K_C}{2r_{in}} \exp\left(-3\frac{D_p}{r_{in}}\right) \mathrm{Re}^{0.9} \tag{1396}$$

where D_p is the equivalent packed bed particle diameter, f_{htg} is the heat transfer factor that is defined, and

$$\mathrm{Re} = \frac{D_p G_s}{\mu} \tag{1397}$$

- For packed bed pipes that use the Beek heat transfer correlation:

$$h_f = f_{htg} \frac{K_C}{D_p}\left(2.58\,\mathrm{Re}^{1/3}\mathrm{Pr}^{1/3} + 0.094\mathrm{Re}^{4/5}\mathrm{Pr}^{0.4}\right) \tag{1398}$$

Where f_{htg} is the heat transfer factor that is defined and the Prandtl number, Pr, is determined by Eqn. (1395).

- For packed bed pipes that use the De Wasch / Froment heat transfer correlation:

$$h_f = \frac{K_C}{D_p}\left(h_f^0 + 0.033\mathrm{Pr}\,\mathrm{Re}\right) \tag{1399}$$

For smooth, cylindrical, reacting channels, the pipe friction factor f, is determined depending on the type of friction factor correlation that is specified:

- For open pipes that use the Blasius friction factor correlation:

$$f = 0.0791\mathrm{Re}^{-0.25} \tag{1400}$$

- For open pipes that use the Filonenko friction factor correlation:

$$f = \frac{0.25}{(0.79\log(\mathrm{Re}) - 1.64)^2} \tag{1401}$$

- For packed bed pipes that use the Hicks friction factor correlation:

$$f = 6.8\frac{(1-\chi)^{1.2}}{\chi^3}\mathrm{Re}^{-0.2} \tag{1402}$$

- For packed bed pipes that use the Ergun friction factor correlation:

$$f = \frac{(1-\chi)}{\chi}\left[1.75 + \frac{150(1-\chi)}{\mathrm{Re}}\right] \tag{1403}$$

The equations in the PFR are solved using a stiff Ordinary Differential Equation (ODE) solver with time-steps that are based on the grid size of the reacting channel and the local reacting channel velocity.

The pressure drop is calculated as:

$$\frac{\partial\left(\rho v^2 + p\right)}{\partial t} = -\rho v^3 \frac{f}{D} \tag{1404}$$

where f is the friction factor and D is the characteristic length of the system. The characteristic length of the system is either:

- for open pipes, the hydraulic radius $\frac{D_H}{2}$

- for packed beds, the packed bed representative particle diameter

Outer Flow STAR-CCM+ Formulation

The outer flow, which you can define as reacting or non-reacting, is simulated using three-dimensional STAR-CCM+. You can mesh the three-dimensional outer flow geometry as long as the geometrical channels within the geometry that represent the reacting channel remain unmeshed.

The heat flux boundary condition at the reacting channel wall, $\dot{Q}_{3D}$, is calculated as an under-relaxation of the heat that the outer flow gains from the reacting channel. This heat flux is determined using:

$$\dot{Q}_{3D} = -a\dot{Q}_C - \left(1 - a\right)\dot{Q}_C^0 \tag{1405}$$

When the solution for the simulation converges, the heat flux from the reacting channel is equal to the flux gain from the outer flow.

Reacting Channel Coupling Nomenclature

The following nomenclature is used in the formulation for reacting channel co-simulation mechanisms.

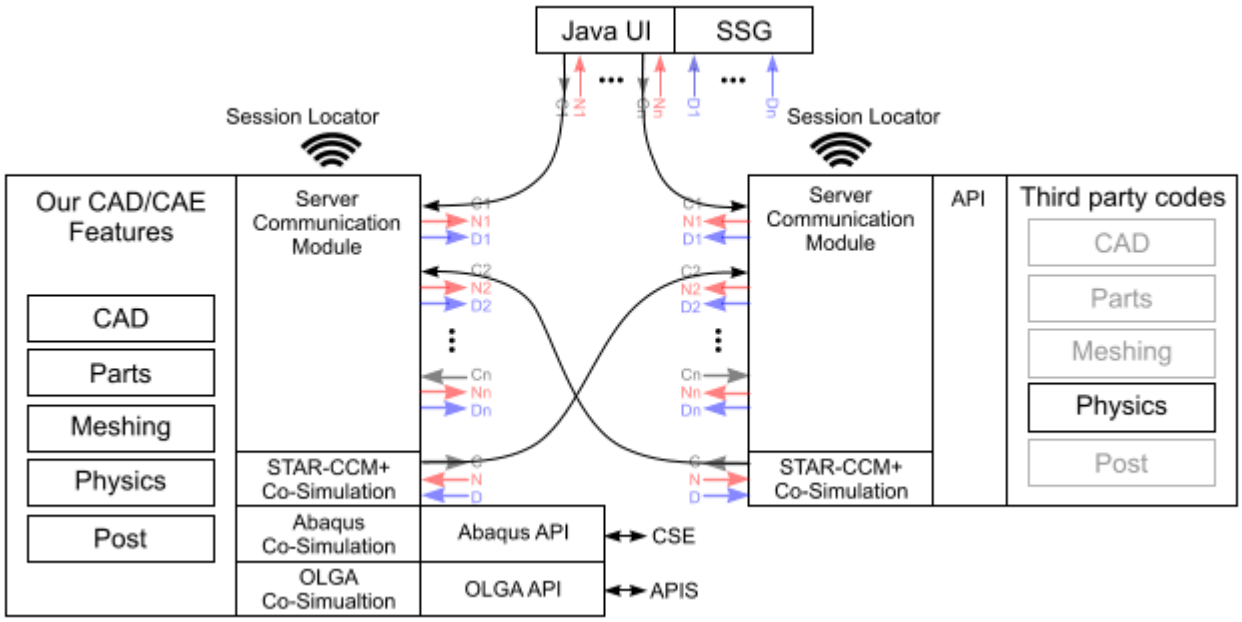
a	Under-relaxation factor, specified as a value in a Reacting Channel Co-Simulation Zone.
c_p	Specific heat, $(\mathrm{J/kg-K})$
χ	Porosity
D_C	Channel diameter, (m), calculated from the inlet area that you specify as a value in a zone.
D_p	Equivalent Packed Bed Particle Diameter, which you can specify as a value in a zone.
f	Pipe Friction Factor, which you can specify as a condition for each zone. See Reacting Channel Pipe Friction Factor Option
f_{htg}	Heat Transfer Factor, which you can specify as a value for a zone within a packed bed reacting channel.
h_f	Convective heat transfer coefficient of the reacting channel, specified as a condition in a Reacting Channel Co-Simulation Zone.
$h_k = k^{th}$	Total enthalpy, $(\mathrm{J/kg})$
K_C	Gas phase thermal conductivity, $(\mathrm{W/m-K})$
μ	Gas viscosity, which you can specify as a condition for a zone. See Reacting Channel Inlet Viscosity Option
Nu	Nusselt number, determined according to fully developed pipe flow empirical calculations. See Reacting Channel Coupling Formulation
$\dot{\omega}$	Net source term, $(\mathrm{kg/m^3-s})$
$\dot{\omega}_k = k^{th}$	Net molar reaction rate, $(\mathrm{kg/m^3-s})$
Pr	Prandtl number as determined by Eqn. (1395)
$\dot{Q}_C$	Convective heat transfer source, $(\mathrm{W/m^2})$, calculated by Eqn. (1391).
$\dot{Q}_C^0$	Heat flux from previous iteration, $(\mathrm{W/m^2})$.

$\dot{Q}_{3D}$	Heat flux boundary condition at the reacting channel wall, $(\mathrm{W/m^2})$
ρ	Mixture density, $(\mathrm{kg/m^3})$
T	Reacting channel bulk temperature, (K), specified as a value in a Reacting Channel Co-Simulation Zone.
$\mathrm{T_W}$	Reacting channel wall temperature, (K), which is averaged from the 3D outer flow temperature field on the resolved reacting channel wall.
$W_k = k^{th}$	Molecular weight, $(\mathrm{kg/kmol})$
$Y_k = k^{th}$	Species mass fraction, specified as a value in a Reacting Channel Co-Simulation Zone.

Co-Simulation API

STAR-CCM+ provides an Application Programming Interface (API) that allows you to write partner code that exchanges data with a running STAR-CCM+ simulation. As the partner code can be part of any other application that supports linking to a flat C library, the co-simulation API extends the range of co-simulation scenarios possible with STAR-CCM+.

When initialized, the Co-Simulation API launches a companion STAR-CCM+ API server alongside the partner software. During co-simulation, the companion STAR-CCM+ API server communicates with other STAR-CCM+ servers using the same protocols as a co-simulation between two STAR-CCM+ simulations. Partner software makes calls into the Co-Simulation API library to receive data from, and transfer data to, the companion STAR-CCM+ API server.



Each release of the Co-Simulation API is identified with a unique number, that is, the API version. Typically, new versions of the Co-Simulation API are released less frequently than new versions of STAR-CCM+. To facilitate the transition to newer API versions and to prolong backwards compatibility, each STAR-CCM+ release supports multiple versions of the Co-Simulation API. You are advised to keep your program up-to-date with the latest version of the Co-Simulation API, as older API versions may be retired in future STAR-CCM+ releases.

The following sections describe the latest version of the API. For more information on older API versions, refer to the User Guide of previous STAR-CCM+ releases.

Contents:

Co-Simulation API Overview
Preparing Your Code for Co-Simulation with STAR-CCM+
Running a Co-Simulation between Partner Code and STAR-CCM+

Co-Simulation API Overview

The following sections provide information on how to couple your code with STAR-CCM+ through the STAR-CCM+ Co-Simulation API. For details on the available API functions and options, refer to the Co-Simulation API Online Help. You can access the Online Help from the STAR-CCM+ interface by selecting **Help** > **Co-simulation API**.

Example Program

The STAR-CCM+ installation contains an example C++ program, SpindleValve, which couples with STAR-CCM+ using the latest Co-Simulation API version. You are advised to study the example code before attempting to create your own program. The SpindleValve example program is located at:

- **Windows:** `[INSTALLDIR]\doc\startutorialsdata\coupling\data\CoSimAPI_Code`
- **Linux:** `[INSTALLDIR]/doc/startutorialsdata/coupling/data/CoSimAPI_Code`

The Co-Simulation API: Spindle Valve tutorial demonstrates how to couple the SpindleValve example program and STAR-CCM+ in a co-simulation analysis.

API Library

The Co-Simulation API library is located at:

- **Windows:** `[INSTALLDIR]\star\lib\[OS]\[COMPILER]\lib\StarccmplusCoSimulationApi.dll`
- **Linux:** `[INSTALLDIR]/star/lib/[OS]/[COMPILER]/lib/StarccmplusCoSimulationApi.so`

The `LibraryLoader.h` and `LibraryLoader.cpp` files included in the SpindleValve example demonstrate how to wrap the API library for dynamic loading.

Header Files

For each supported API version, STAR-CCM+ provides a header file that contains the function suites which define that version of the API. The header files are located in the STAR-CCM+ installation, at:

- **Windows:** `[INSTALLDIR]\star\include`
- **Linux:** `[INSTALLDIR]/star/include`

In the online help, you can access an HTML version of the header files under the **Files** tab.

The header files corresponding to the latest version of the API are:

`StarccmplusCosimulationApiV6.h`	Contains the function suites that define the Co-Simulation API v6. Your program only needs to include this header file. The `StarccmplusCosimulationApiStruct.h` and `StarccmplusCosimulationApiConstants.h` files are automatically included.
`StarccmplusCosimulationApiStruct.h`	Contains the primary access function for the function suite API.

`StarccmplusCosimulationApiConstants.h`	Contains constants used by the function suite API.

Note: The folder contains additional header files corresponding to deprecated versions of the API, namely, `StarccmplusCosimulationApi.h`, `StarccmplusCosimulationApiV3.h`, and `StarccmplusCosimulationApiV4.h`. Note that your program can include only one of the provided header files.

API Structure and Function Suites

The Co-Simulation API consists of a set of functions that provide an interface for communication between your program and the STAR-CCM+ API server. These functions are grouped into function suites.

There are two types of function suite. The first type, which includes the main API Suite, the Factory Suite, and the Properties Suite, provides an interface for your program to call into the STAR-CCM+ API server. The second type, which includes the Handler Suite, provides an interface for the STAR-CCM+ API server to call back into your program. All function suites are contained within the header file, `StarccmplusCosimulationApiV#.h`, where `#` is the API version number.

Each function suite is identified by the API version with which the function suite is released and a unique name. This framework allows versioning by providing function suite definitions for each API version. Each new API version may extend an existing function suite, create a new function suite, or retire an old function suite.

Before you can access the function suites, you make a call into the Co-Simulation API library to get a pointer to the API structure.

API Structure

The API structure, which is contained in `StarccmplusCosimulationApiStruct.h`, contains pointers to the API library functions that allow your program to:

- verify the API version
- retrieve the STAR-CCM+ function suites by API version and name
- specify your handler function suite by API version and name

For details on all the available functions and options, go to the `Online Help > Classes > StarccmplusCoSimulationApiStruct` page.

Main API Suite

The main function suite, `StarccmplusApiSuiteV#`, where # is the API version number, contains pointers to the API library functions that allow your program to:

- initialize or terminate the API library
- set the time-step
- set the coupling time interval (explicit coupling) or the number of inner iterations per exchange (implicit coupling)
- instruct your program to wait for data from STAR-CCM+
- notify STAR-CCM+ that the mesh or field data are ready for export
- notify STAR-CCM+ that the current processing loop is complete

For details on all the available functions and options, go to the `Online Help > Classes > StarccmplusApiSuiteV6` page.

Factory Suite

The factory suite, `StarccmplusFactorySuiteV#`, where # is the API version number, contains pointers to the API library functions that allow your program to communicate with, and instantiate objects on, the STAR-CCM+ API server. Specifically, this function suite allows you to:

- create physics continua
- create regions and boundaries
- add a co-simulation
- create co-simulation zones
- assign regions and boundaries to co-simulation zones
- provide a mesh
- import/export field data

For details on all the available functions and options, go to the `Online Help > Classes > StarccmplusFactorySuiteV6` page.

Properties Suite

To transfer data between your program and the STAR-CCM+ API server, you define intermediate data store structures called properties containers. A properties container has its own ID of type `int`. API calls that create properties containers return the container ID. Your program must retain container IDs for all future operations that involve the same properties container.

The properties suite, `StarccmplusPropertiesSuiteV#`, where # is the API version number, contains pointers to the API library functions that allow your program to:

- create a properties container
- fill a properties container
- add or remove properties (integers, strings, arrays, arrays of arrays, etc.) to a properties container
- extract properties from a properties container

For details on all the available functions and options, go to the `Online Help > Classes > StarccmplusPropertiesSuiteV6` page.

Handler Suite

The handler suite, `StarccmplusHandlerSuiteV#`, where # is the API version number, contains pointers to the functions that allow STAR-CCM+ to call back into your program.

For details, go to the `Online Help > Classes > StarccmplusHandlerSuiteV3` page (the Handler Suite has not changed since v3 of the API).

Helper Scripts

The STAR-CCM+ installation contains two helper scripts, `buildapi` and `starapi`. These scripts allow you to compile and launch the example program, SpindleValve. See Example Program.

You can also use these scripts to compile and launch your own program. However, you are advised to compile and launch your program using your own build system, based on requirements.

The helper scripts are located at:

- **Windows:**

  ```
  [INSTALLDIR]\star\bin\buildapi.bat
  [INSTALLDIR]\star\bin\starapi.bat
  ```

- **Linux:**

  ```
  [INSTALLDIR]/star/bin/buildapi
  [INSTALLDIR]/star/bin/starapi
  ```

buildapi

The `buildapi` helper script provides a build system for compiling your program. The script works with the following compilers:

- gcc for C and Intel Fortran 11 on Linux
- Intel C++ Compiler and Intel Fortran Compiler on Windows

Note: The script is tested with the compilers that are used to build STAR-CCM+. For information, refer to the Installation Guide, section *Certified Platforms for STAR-CCM+ v12.06*.

To use the script to compile your code, use the following command:

- **Windows:**

  ```
  > call "[INTEL_C++_INSTALLDIR]\bin\compilervars.bat" intel64
  > [INSTALLDIR]\star\bin\buildapi.bat [<verbosity>] <executable
  name> [<source files>]
  ```

- **Linux:**

  ```
  % [INSTALLDIR]/star/bin/buildapi [<verbosity>] <executable name>
  [<source files>]
  ```

where:

- `[<verbosity>]` specifies the verbose mode.

  The available options are:

  - `-v` for low verbosity
  - `-vv` for medium verbosity
  - `-vvv` for high verbosity

- `<executable name>` specifies the target name of the final executable.
- `[<source files>]` specifies the source code files or includes source file directories. If you do not specify this argument, the compiler will compile all the files with extension `.cpp`, `.c`, or `.f`, `.f90` that are located in the current folder.

Note: If you experience problems running the script, open it in a text editor and check that all required paths are resolved by the environment you are working with.

starapi

The `starapi` script launches your program and sets the environment that allows your program to load the co-simulation library and launch the STAR-CCM+ API server. To launch your program from a command console, use the following command:

- **Windows:**

```
> [INSTALLADIR]\star\bin\starapi.bat [<verbosity>] <executable>
[<executable arguments>]
```

- **Linux:**

```
% [INSTALLDIR]/star/bin/starapi [<verbosity>] <executable>
[<executable arguments>]
```

where:

- `[<verbosity>]` specifies the verbose mode.

  The available options are:

  - `-v` for low verbosity
  - `-vv` for medium verbosity
  - `-vvv` for high verbosity
- `<executable>` specifies the executable path and name.
- `[<executable arguments>]` contains additional parameters required by the partner program.

The API server behaves like any other STAR-CCM+ servers, that is, you can connect to the API server from a STAR-CCM+ client. Visualizing the API server in a STAR-CCM+ client can be useful for debugging purposes. However, make sure that you do not make any changes to the API server through the STAR-CCM+ client. Any changes to the API server must come from your program through the Co-Simulation API.

Note: Launching your program from a command console checks out a STAR-CCM+ license. For information on how to run a co-simulation between your program and STAR-CCM+ using a single license, see Running a Co-Simulation between Partner Code and STAR-CCM+.

Preparing Your Code for Co-Simulation with STAR-CCM+

Your program couples with STAR-CCM+ through the STAR-CCM+ API server, which is a STAR-CCM+ simulation that represents your program. You set up and communicate with the STAR-CCM+ API server through the Co-Simulation API library. The STAR-CCM+ API server couples with other STAR-CCM+ simulations using STAR-CCM+ to STAR-CCM+ Co-Simulation.

To set up and communicate with the STAR-CCM+ API server, prepare your code following the steps below. Each step includes a code example that uses the latest version of the API.

1. Include one of the Co-Simulation API header files in your program.

   See Header Files. API versions based on the function suite structure are not suitable for coupling with versions of STAR-CCM+ prior to v11.04.

2. For dynamic loading of the Co-Simulation API library, also include the `LibraryLoader.h` and `LibraryLoader.cpp` files that are provided with the SpindleValve example program.

   - If your program is written in C++, you can directly use the `LibraryLoader.h` and `LibraryLoader.cpp` files in your program.
   - If your program is written in Fortran or C, you can use these files with a C wrapper.

   See Example Program and API Library.

3. Define pointers to the API structure and the function suites:

```
StarccmplusCoSimulationApiStruct const * apiStruct;
StarccmplusApiSuiteV6 const * apiSuite;
StarccmplusFactorySuiteV6 const * factorySuite;
StarccmplusPropertiesSuiteV6 const * propertiesSuite;
```

4. Define the `messageHandler` and `fillContainerHandler` callback functions that STAR-CCM+ uses to call back into your program:

```
void messageHandler(StarccmplusMessageType const msgType, char
const* message)
{see SpindleValve program for example message handler};

void fillContainerHandler(int containerIndex, char const*
requirement)
{see SpindleValve program for example fill container handler};

StarccmplusHandlerSuiteV6 handlerSuite ={
  messageHandler,
  fillContainerHandler};
```

5. Load the Co-Simulation API library.

```
apiLib.load();
```

6. Get a pointer to the API structure defined in `StarccmplusCoSimulationApiStruct.h`.

```
apiLib.getApiStruct(&apiStruct)
```

7. Verify that the Co-Simulation API version in use is supported.

   The STAR-CCM+ version that is launched is a user choice, that is, your program cannot use the API to load specific versions of STAR-CCM+. Typically, your program verifies whether the version of STAR-CCM+ that the user launches is compatible with the API version in use, and issues an error when it is not compatible.

```
apiStruct->isApiVersionSupported(
  API_VERSION,
  &versionsArray,
  &versionsArraySize);
```

   The API queries STAR-CCM+ for a list of supported API versions, then matches the input `API_VERSION` against this list:

   - If there is no match, the API call returns 1. In this case, unless your program can handle falling back to a different version of the API, you cannot couple with this version of STAR-CCM+.
   - If there is a match, the API call returns 0. In this case, your program can couple with this version of STAR-CCM+.

   Although you cannot use the API to load specific versions of STAR-CCM+, you can take into account the STAR-CCM+ version in order to determine the capabilities available in STAR-CCM+ (for example, to determine whether implicit coupling is available).

8. Get pointers to the function suites.

```
apiStruct->setExternalCodeFunctions(
  API_VERSION,
  StarccmplusCoSimulationHandlerSuiteName,
  &handlerSuite);
```

```
apiStruct->getStarccmplusFunctions(
  API_VERSION,
  StarccmplusCoSimulationApiSuiteName,
  (void const * *)&apiSuite);
```

```
apiStruct->getStarccmplusFunctions(
  API_VERSION,
  StarccmplusCoSimulationFactorySuiteName,
  (void const * *)&factorySuite);
```

```
apiStruct->getStarccmplusFunctions(
  API_VERSION,
  StarccmplusCoSimulationPropertiesSuiteName,
  (void const * *)&propertiesSuite);
```

   After these calls, you make all subsequent calls using the function suites.

9. Initialize the library.

```
apiSuite->initialize(argc,argv);
```

10. Set up the STAR-CCM+ API Server:
    a) Create a physics continuum on the STAR-CCM+ API Server. The continuum requires you to specify a material model, a time discretization model, and the co-simulation coupling scheme.

```
int continuumId = factorySuite->createContinuum(
  "Solid Continuum",
  StarccmplusSingleComponentSolid,
  StarccmplusSingleSurfacePolyMesh,
  StarccmplusImplicitUnsteady,
  StarccmplusExplicitCoupling,
  propertiesSuite->create());
```

    b) Create a region on the STAR-CCM+ API server. As the region requires you to specify a continuum ID, make sure that you create the physics continuum first.

```
int regionId = factorySuite->createRegion(
  continuumId,
  "Spindle Ball",
  StarccmplusSurfaceRegion,
  StarccmplusSolidRegion);
```

    c) Create the region boundaries. As you create boundaries within regions, make sure that you create a region first.

```
int boundaryId = factorySuite->createBoundary(
  regionId,
  "Spindle Ball Surface",
  StarccmplusWallBoundary);
```

    d) Add a co-simulation link.

```
int coSimulationId = factorySuite->createCoSimulation(
  continuumId,
  "Link 1",
  propertiesSuite->create();
```

    e) Create a co-simulation zone of appropriate type (either surface, for surface-to-surface coupling, or volume, for volume-to-volume coupling). As you create co-simulation zones within the co-simulation link, you require the co-simulation link ID.

```
int zoneId = factorySuite->createCoSimulationZone(
  coSimulationId,
  "zoneName",
  StarccmplusSurfaceZone,
  propertiesSuite->create();
```

f) To couple a model part (either boundary or region) with STAR-CCM+, assign it to a co-simulation zone. Assign coupled boundaries to surface zones and coupled regions to volume zones.

```
factorySuite->addModelPartToZone(
zoneId,
boundaryId);
```

g) Provide a mesh to the STAR-CCM+ API server that conforms to the STAR-CCM+ polyhedral mesh format (see Polyhedral Mesh Reference). To register the mesh, you require the region ID.

```
int meshId = factorySuite->registerOutgoingMesh(
  regionId,
  StarccmplusSurfacePolyMesh);
```

h) Register data fields that your program sends to STAR-CCM+ through the co-simulation zone.

```
int displId = factorySuite->registerOutgoingField(
  zoneId,
  StarccmplusVertexFieldLocation,
  StarccmplusDisplacementSpecification,
  StarccmplusDisplacements,
  propertiesSuite->create());
```

i) Register data fields that your program receives from STAR-CCM+ through the co-simulation zone.

```
int pressureId = factorySuite->registerIncomingField(
  zoneId,
  StarccmplusFaceFieldLocation,
  StarccmplusPressureSpecification,
  StarccmplusPressure,
  propertiesSuite->create());
```

11. Notify STAR-CCM+ that the mesh is ready for use:

```
apiSuite->notifyOutgoingMeshesReady();
```

12. Specify the solver settings and run loop. Include the solver time-step and either the coupling time (for explicit coupling) or the number of inner iterations per exchange (for implicit coupling).

a) Set the solver time-step.

```
apiSuite->setTimeStep(t, dt);
```

b) Specify the data exchange frequency:

Explicit Loop	Implicit Loop
Specify the coupling time, which must be equal to, or greater than, the solver target time (current time + time-step). `apiSuite->setCouplingTime(couplingTime);`	Set the number of inner iterations per exchange. `apiSuite->setNumberOfInnerIterationsPerExchange(`

Explicit Loop	Implicit Loop
	`inner_iterations_per_exchange);`

c) Wait for incoming fields.

```
apiSuite->waitForIncomingFields();
```

d) If the return value of the waitForIncomingFields call is non-zero, a data exchange occurred. In this case, retrieve the registered incoming fields.

```
propertiesSuite->addDouble(
  pressureId,"Time",t);

propertiesSuite->fill(pressureId,"FieldValues");

propertiesSuite->getDoubleArray(
  pressureId, "Pressure", &pressField[0], pressField.size());
```

e) Solve for partner solution and prepare the solution data for export.

f) At every iteration or time-step, notify STAR-CCM+ that outgoing fields are ready.

```
apiSuite->notifyOutgoingFieldsReady();
```

g) At the end of a solution loop, notify STAR-CCM+ that the time-step is complete.

```
apiSuite->notifyTimeStepComplete();
```

h) Repeat the solution loop until done.

13. When complete, finalize the Co-Simulation API library.

```
apiSuite->finalize();
```

14. Terminate the API library.

```
apiLib.close();
```

Contents:

Polyhedral Mesh Reference
Exchanged Fields Reference

Polyhedral Mesh Reference

Co-simulation between STAR-CCM+ and partner code through the STAR-CCM+ API allows for surface-to-surface and volume-to-volume coupling.

For more information on these coupling options, see Co-Simulation Interface and Exchanged Fields.

When the STAR-CCM+ API asks your program to fill the mesh, provide the data identified in the sections below.

Surface Mesh

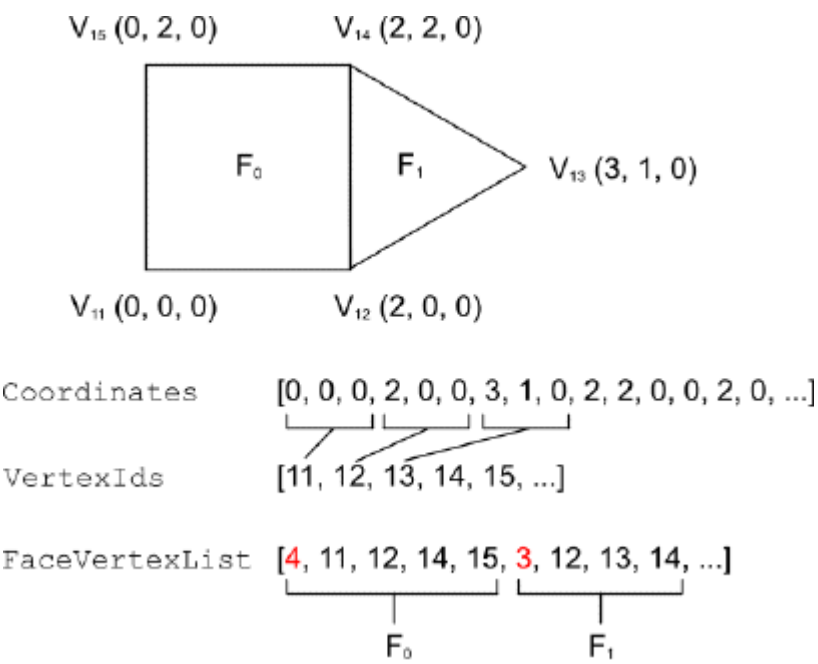
For surface-to-surface coupling, provide the following data:

String Identifier	Data Type	Length
`NumberOfFaces`	Integer	Single value
`NumberOfVertices`	Integer	Single value
`NumberOfDimensions`	Integer	Single value
`Coordinates`	Double Array	`NumberOfVertices * NumberOfDimensions`
`VertexIds`	Integer Array	`NumberOfVertices`
`FaceVertexList`	Integer Array	`Sum over all faces(Face Vertex Count + 1)`

The `FaceVertexList` is a ragged array whose sequence follows the pattern:

```
[<Num Vertices on Face 0>, V0, V1, V2,..., <Num Vertices on Face 1>,
V0, V1, V2, ...]
```

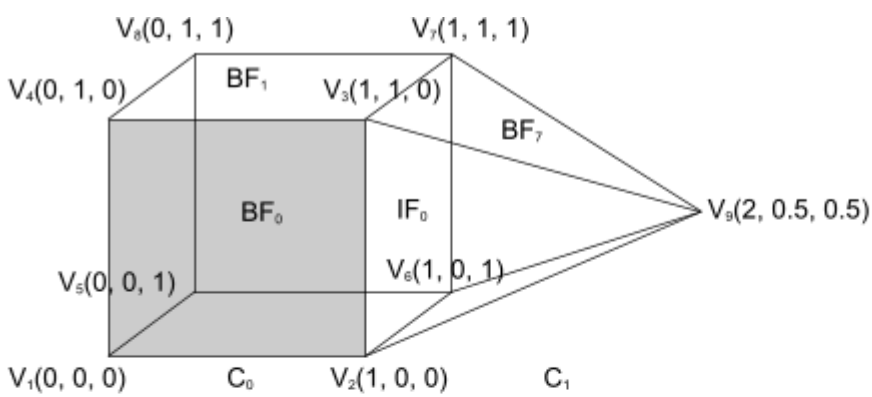
The diagram below illustrates the relationship between the arrays. After constructing the mesh, your program should call the API function `notifyOutgoingMeshesReady()`.



Volume Mesh

For volume-to-volume coupling, provide the following data:

String Identifier	Data Type	Length
`NumberOfCells`	Integer	Single value
`NumberOfVertices`	Integer	Single value
`NumberOfInteriorFaces`	Integer	Single value
`NumberOfBoundaryFaces`	Integer	Single value
`NumberOfDimensions`	Integer	Single value
`Coordinates`	Double Array	`NumberOfVertices * NumberOfDimensions`
`VertexIds`	Integer Array	`NumberOfVertices`
`InteriorFaceCellIndexList`	Integer Array	`NumberOfInteriorFaces * 2`
`InteriorFaceVertexList`	Integer Array	`Sum over all interior faces (Face Vertex Count + 1)`
`BoundaryFaceCellIndexList`	Integer Array	`NumberOfBoundaryFaces * 2`
`BoundaryFaceVertexList`	Integer Array	`Sum over all boundary faces (Face Vertex Count + 1)`



`InteriorFaceVertexList:` [4, 2, 3, 7, 6] (IF_0)

`InteriorFaceCellIndexList:` [C_0, C_1] (IF_0)

`BoundaryFaceVertexList:` [4, 1, 2, 3, 4, ... , 3, 3, 9, 7, ...] (BF_0, BF_7)

`BoundaryFaceCellIndexList:` [C_0, `nullIndex`, C_0, `nullIndex`, ... , C_1, `nullIndex`, ...] (BF_0, BF_1, BF_7)

where `nullIndex = max unsigned int`.

Exchanged Fields Reference

The Co-Simulation API provides functions for outgoing and incoming fields. Registration functions instruct the API to prepare for exchanging particular fields.

The following code shows an example registration for outgoing fields. This example uses the Property Suite to assign a string (in this case, the field name) to the registered field:

```
int propertiesId = propertiesSuite->create();

propertiesSuite->addString(propertiesId, "FieldName", "Displacement
Field");

int displacementId = factorySuite->registerOutgoingField(
  zoneId,
  StarccmplusVertexFieldLocation,
  StarccmplusDisplacementSpecification,
  StarccmplusDisplacements,
  propertiesId);
```

Example registration for incoming fields:

```
int pressureId = factorySuite->registerIncomingField(
  zoneId,
  StarccmplusFaceFieldLocation,
  StarccmplusPressureSpecification,
  StarccmplusPressure,
  propertiesSuite->create());
```

Use the following table in registration functions for data fields:

StarccmplusSpecification	StarccmplusFieldType	StarccmplusFieldLocation
StarccmplusPressureSpecification	StarccmplusPressure	StarccmplusFaceFieldLocation
StarccmplusTractionSpecification	StarccmplusPressure AND StarccmplusWallShearStress	StarccmplusFaceFieldLocation
StarccmplusDisplacementSpecification	StarccmplusDisplacements	StarccmplusVertexFieldLocation
StarccmplusTemperatureSpecification	StarccmplusTemperature	StarccmplusFaceFieldLocation
StarccmplusConvectiveFluxSpecification	StarccmplusHeatTransferCoefficient AND StarccmplusReferenceTemperature	StarccmplusFaceFieldLocation
StarccmplusEnvironmentalFluxSpecification	tarccmplusHeatTransferCoefficient AND StarccmplusReferenceTemperature	StarccmplusFaceFieldLocation

StarccmplusSpecification	StarccmplusFieldType	StarccmplusFieldLocation
StarccmplusScalarAuxiliarySpecification	StarccmplusScalarAuxiliaryField	StarccmplusFaceFieldLocation OR StarccmplusVertexFieldLocation
StarccmplusVectorAuxiliarySpecification	StarccmplusVectorAuxiliaryField	StarccmplusFaceFieldLocation OR StarccmplusVertexFieldLocation

Use the `StarccmplusSpecification` to say what quantity your program is going to exchange (incoming or outgoing). Use the `StarccmplusFieldType` to define the type of data that the codes exchange for that quantity.

StarccmplusFieldType	Array Type	Array Length	Dimension	Units
`StarccmplusPressure`	1 double per face	`NumberOfFaces`	Pressure	Pa
`StarccmplusWallShearStress`	3 doubles per face	`NumberOfFaces * 3`	Pressure	Pa
`StarccmplusDisplacements`	3 doubles per vertex	`NumberOfVertices * 3`	Length	m
`StarccmplusTemperature`	1 double per face	`NumberOfFaces`	Temperature	K
`StarccmplusHeatTransferCoefficient`	1 double per face	`NumberOfFaces`	Power/ Length^2-Temperature	W/m^2K
`StarccmplusReferenceTemperature`	1 double per face	`NumberOfFaces`	Temperature	K
`StarccmplusScalarAuxiliaryField`	1 double per face or cell	`NumberOfFaces` or `NumberOfCells`		
`StarccmplusVectorAuxiliaryField`	3 double per face or cell	`NumberOfFaces` or `NumberOfCells`		

Running a Co-Simulation between Partner Code and STAR-CCM+

Compile your code then launch your program from STAR-CCM+ for co-simulation.

1. Before coupling to a STAR-CCM+ simulation, compile your code. You can use your own build system or the `buildapi` helper script that is provided in the STAR-CCM+ installation. To use the `buildapi` script:
   a) In a terminal/command prompt, go to the folder that contains the code files.
   b) From this location, run the `buildapi` helper script:
      - On Windows, set the Intel C++ Compiler environment before running the `buildapi.bat` script:

        ```
        > call "[INTEL_C++_INSTALLDIR]\bin\compilervars.bat" intel64
        > [INSTALLDIR]\star\bin\buildapi.bat <executable name>
        ```

      - On Linux, make sure that you use g++ 4.3.4 or greater before running the `buildapi` script:

        ```
        % [INSTALLDIR]/star/bin/buildapi <executable name>
        ```

      where `<executable name>` specifies a name for the output executable file.

      As the argument for the source files is not specified, this command compiles all the `.cpp` files that are located in the project folder and creates an executable output file that is used to launch the partner program. See the section, Helper Scripts, for more information as well as additional arguments that you can use with this script.
2. Launch the STAR-CCM+ simulation that you want to couple with your program.
3. Set up the simulation for STAR-CCM+ to STAR-CCM+ Co-Simulation. For more information, see STAR-CCM+ Co-Simulation Workflow.
4. In the STAR-CCM+ simulation, select the **Co-Simulations** > **[Link 1]** > **Conditions** > **Launch Partner Option** node and set *Method* to **Launch Application via Command Line**.
5. Select the **[Link 1]** > **Values** > **Launch Command Line** node. For the *Command Line* property, enter:
   - On Windows:

     ```
     > [INSTALLDIR]\star\bin\starapi.bat <executable>
     ```

   - On Linux:

     ```
     % [INSTALLDIR]/star/bin/starapi <executable>
     ```

   where `<executable>` specifies the path to the executable file created when compiling your code. See the section, Helper Scripts, for more information about the `starapi` script as well as additional options that you can use with this script.
6. Run the co-simulation from the STAR-CCM+ simulation.

Note: Solution stepping is not allowed for co-simulation with external code through the Co-Simulation API.

When you run the co-simulation, the `starapi` helper script launches your program and the associated STAR-CCM+ API server. The STAR-CCM+ simulation connects to the running program through the STAR-CCM+ API server and the co-simulation analysis starts.

Note: When using the **Launch Application via Command Line** method, the STAR-CCM+ simulation and the STAR-CCM+ API server share the same STAR-CCM+ license, meaning that you require a single license to run the co-simulation. On the other hand, if you launch the partner program separately (for example running the starapi script in a command console) and then connect the STAR-CCM+ simulation to the running program by specifying the partner program host and port, you require two STAR-CCM+ licenses.

Data Mappers

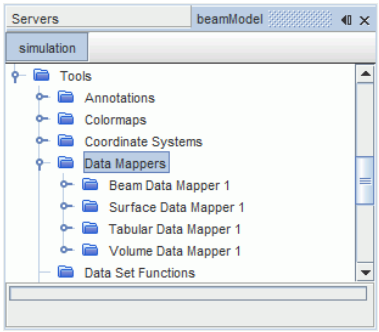
In STAR-CCM+, data mappers allow you to interpolate field data available either on a source mesh or on a source table to a target mesh representation, so that the mapped fields can be used for applying boundary or volume conditions on the target mesh.

Data mapping is an essential step in explicit coupling for multiphysics applications. For example, in fluid structure interaction problems, the solid and the fluid exchange fluid forces and solid displacements across a shared surface. If the solid and the fluid have different meshes, you use surface-to-surface mapping to map the fields between the different representations of the shared surface. In an explicit thermo-structural analysis, you can use different meshes for the conjugate heat transfer and the structural analyses, for example, a finite volume (FV) mesh for the CHT analysis and a finite element mesh (FE) for the structural analysis. In this case, you use volume-to-volume mapping to interpolate the temperature field from the FV mesh to the FE mesh, and use the mapped thermal loads as boundary conditions for the structural analysis.

Data mappers allow you to:

- map data to/from surfaces and volumes defined on the following discrete representations:
  - STAR-CCM+ finite volume (FV) or finite element (FE) meshes
  - finite element (FE) meshes imported from third-party CAE packages
- map data to/from FE beams imported from third-party CAE packages
- map tabular data onto a target mesh:
  - allows you to define boundary conditions on the target mesh using the interpolated fields, instead of the original tabular data
  - intended for cases where the tabular values are much coarser than the target mesh representation

STAR-CCM+ offers the following types of mappers:

	**Surface Data Mapper**: interpolates data from source surfaces to target surfaces, volumes, or beams. Surface-to-beam mapping is intended for FSI applications, as it allows you to map pressure and shear stress data, defined on the source surfaces mesh, to the equivalent forces and moments on the nodes of an imported CAE beam.
	Volume Data Mapper: interpolates data from source volumes to target surfaces or volumes.
	Beam Data Mapper: interpolates data from a single source beam to multiple target surfaces. Multiple beams require the creation of multiple mappers.

	Beam-to-surface mapping is intended for FSI applications, as it allows you to map nodal displacements and rotations, defined on an imported beam model, onto the vertices of the wetted STAR-CCM+ surfaces. The mapped beam data can be used by the morpher to update the mesh of the wetted surfaces.
	Tabular Data Mapper: interpolates data from an XYZ table to target surfaces, volumes, or beams.

There are various ways to create a data mapper in an existing simulation:

- Manual setup -- For more information on how to set up mapping operations manually, refer to the workflow and reference sections, Mapping Fields Using Data Mappers and Data Mappers Reference.
- Invoking mapping actions on imported models -- A data mapper is automatically created and listed under **Tools** > **Data Mappers** whenever a map action is invoked on the **[Imported Model]** node. The properties of the mapper are automatically filled based on the selections in the *Map* dialog. The created mapper can be modified or reused as desired. See Using File-Based Coupling.
- Setting up a simulation for co-simulation with a third party code -- A data mapper is automatically created and listed under **Tools** > **Data Mappers**.

Caution: Data mappers can cause inconsistent reporting if you use them with field function monitors or with update events other than iteration, time-step, or delta time.

Contents:

Mapping Fields Using Data Mappers
Data Mappers Reference
Data Mapper Formulation
Examples of Using Data Mappers

Mapping Fields Using Data Mappers

The following workflow demonstrates how to manually create, set up and execute mapping operations, to interpolate field data between different mesh representations or between tabular data and target meshes.

For data mapping between different meshes, STAR-CCM+ offers alternative methods to the manual set up. For example, you can map field data between STAR-CCM+ and imported CAE model meshes using the imported CAE models framework, that automatically creates data mappers based on the simulation settings. For more information, see Using File-Based Coupling. You can also exchange data between two STAR-CCM+ simulations, or between a STAR-CCM+ simulation and a third-party code, using the co-simulation framework. See Coupling with CAE Products.

The first step in the workflow is to create a data mapper and to specify source settings. Depending on whether your source data is available on a source mesh, or in tabular form, follow the guidelines outlined in one of the following sections:

- Mapping Data from Surface Volume or Beam Sources
- Mapping Tabular Data

The steps for setting up the target mesh, defining constraint checks, and executing the mapper are the same in both cases. As this is a general workflow, refer to the reference sections for a complete list of available properties and options.

Note: When mapping fields onto STAR-CCM+ volumes or surfaces, you can either use a parts-based approach, where you map the data onto geometry parts or part surfaces, or a region-based approach, where you map the data onto regions or boundaries. When you visualize the mapped data within scenes and reports, make sure that you display the data onto the STAR-CCM+ objects you mapped the data to.

For a practical example of mapping fields between a STAR-CCM+ finite element mesh and a STAR-CCM+ finite volume mesh, refer to the Thermal Strain: Exhaust Manifold tutorial.

Contents:

Mapping Data from Surface, Volume, or Beam Sources
Mapping Tabular Data
Specifying the Target Mesh and Interpolation Method
Defining Constraint Checks
Executing the Mapping Operation

Mapping Data from Surface, Volume, or Beam Sources

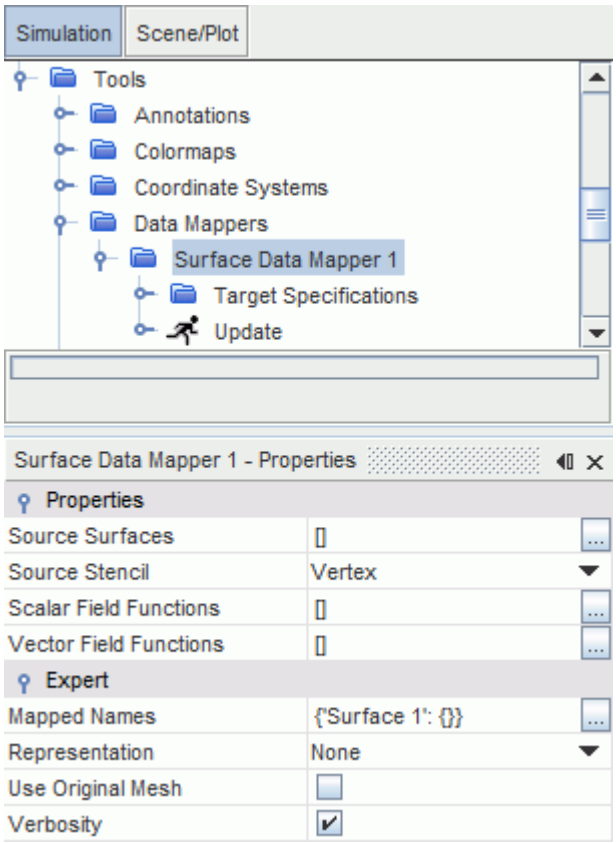
The steps for creating a data mapper to interpolate fields between different mesh representations are outlined in this section.

To create a data mapper:

1. Right-click the **Tools** > **Data Mappers** node, select **New Data Mapper** and choose the mapper type based on the type of source:

- To map data from surfaces of a STAR-CCM+ 3D mesh or of an imported CAE mesh, choose a **Surface Data Mapper**.

  STAR-CCM+ treats 2D regions as surfaces. Therefore, to map data from a STAR-CCM+ 2D region, choose a **Surface Data Mapper**.
- To map data from volumes of a STAR-CCM+ 3D mesh or of an imported CAE mesh, choose a **Volume Data Mapper**.
- To map data from an imported CAE beam, choose a **Beam Data Mapper**. This option is intended for FSI applications and allows you to map nodal displacements and rotations, defined on an imported CAE beam, to the wetted STAR-CCM+ surfaces.



2. Depending on whether you created a Surface, Volume, or Beam Data Mapper, specify the source by setting the *Source Surfaces*, *Source Volumes* or *Source Beams* property.

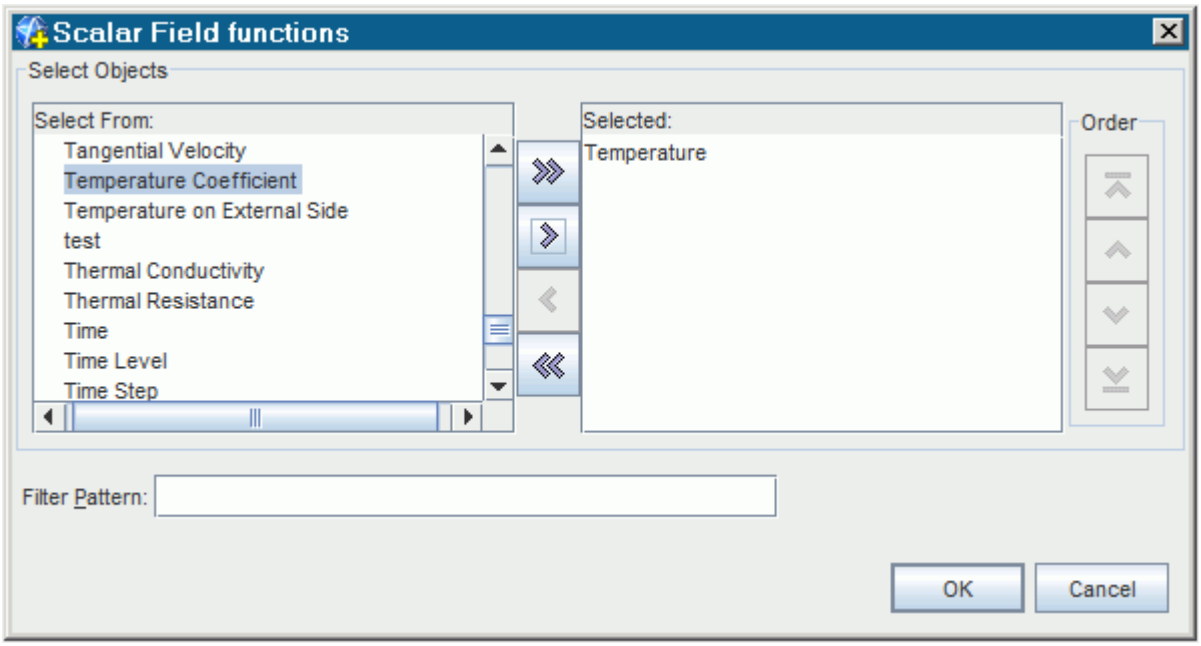
   Source surfaces can either be 2D regions, region boundaries, or part surfaces in STAR-CCM+, or surfaces defined on an imported CAE model. Source volumes can either be STAR-CCM+ 3D regions or parts, or volumes defined on an imported CAE model. You can specify multiple source surfaces or volumes.

   For a beam mapper, specify a single source beam. Additional source beams require additional mappers. A source beam can only be an imported CAE beam model.
3. Set the *Source Stencil* property to specify the location where the solution data is available on the source mesh.

For finite volume (FV) source meshes, the source stencil is typically either a cell (for source volumes) or a face (for source surfaces). For finite element (FE) source meshes, the source stencil is typically a vertex. For more information, see Surface, Volume, Beam, and Tabular Data Mapper Reference.

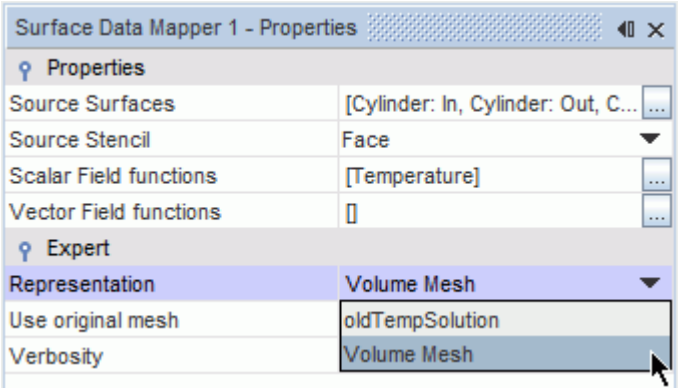
Specify at least one scalar or vector field for mapping. The data mapper maps the specified fields from the source to the target mesh when you execute the mapping operation.

4. Select the *Scalar* or *Vector Field Functions* property. Click ... (**Custom Editor**) and use the selection dialog to select one or more fields from the list of field functions that are available for mapping.



5. If the source mesh is a STAR-CCM+ mesh, select the representation to get source data from.

   You can use source data defined on the volume mesh representation, or defined on any representations available in a simulation history file.



If the source mesh is subject to any motion other than Stationary, specify the reference mesh for mapping:

6. Activate *Use Original Mesh* to use the original mesh as the reference mesh for mapping. Deactivate this option to use the current mesh as the reference mesh for mapping.

When you execute a data mapper, the mapped fields become available as field functions, with the default name `Mapped<Target_Stencil><Field_Name>`. To specify alternative names for the output field functions:

7. Select *Mapped Names*. Click ... (**Custom Editor**) and use the dialog to specify field function names for each of the specified fields.
8. Follow the steps outlined in the next sections to specify the target mesh and other mapping options, and to execute the data mapper.

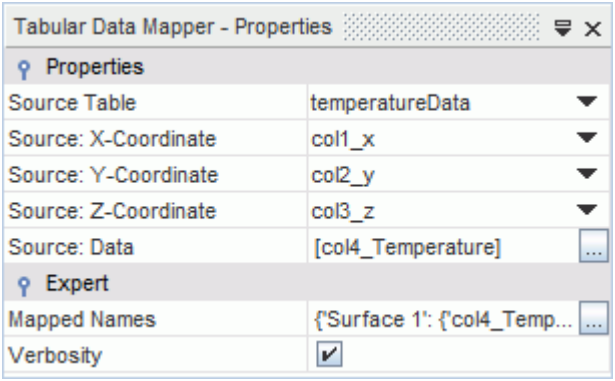
Mapping Tabular Data

In STAR-CCM+, you can define boundary or initial conditions using tabular data. If the table data is much coarser than the target mesh, create a tabular data mapper to interpolate the tabular data to the mesh.

The table can be an XYZ table created in STAR-CCM+, or a file table that you imported in STAR-CCM+. The XYZ table must have at least four named columns. Three of the columns specify the x, y, and z location of a value and the fourth column specifies the scalar value to be mapped at that location. You can specify more fields by adding additional columns.

1. Create the XYZ table with the source data in STAR-CCM+, or import a file table in STAR-CCM+ using the **Tools** > **Tables** node.

   For more information, see Creating a Table and Reading a Table Data File.
2. Right click the **Data Mappers** node and select **New Data Mapper** > **Tabular Data Mapper**.
3. Use the tabular data mapper properties to specify the table and which table columns represent the field values, and the x, y, and z coordinates that define the location of each field value.



When you execute a data mapper, the mapped fields become available as field functions, with the default name `Mapped<Target_Stencil><Field_Name>`. To specify alternative names for the output field functions:

4. Select *Mapped Names*. Click ... (**Custom Editor**) and use the dialog to specify field function names for each of the specified fields.

For more information on the available properties, see Surface, Volume, Beam, and Tabular Data Mapper Reference.

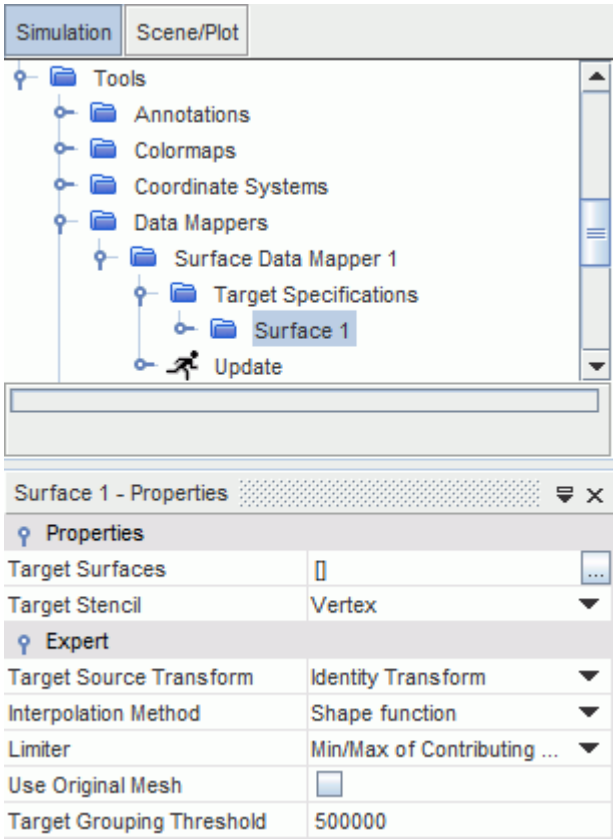
5. Follow the steps outlined in the next sections to specify the target mesh and other mapping options, and to execute the data mapper.

Specifying the Target Mesh and Interpolation Method

Set up mapping targets to specify the target mesh, the interpolation method, and other mapping options. A mapping operation can have one or more targets of type **Surface**, **Volume**, or **Beam**.

To add a target:

1. Right-click the **Target Specifications** node, select **New Group** and choose the type of target:
   - To map the source data to surfaces of a STAR-CCM+ mesh or of an imported CAE mesh, choose a **Surface** target.
   - To map the source data to volumes of a STAR-CCM+ mesh or of an imported CAE mesh, choose a **Volume** target.
   - To map pressure and shear stress, defined on one or more surfaces, to nodal forces and moments on an imported CAE beam, choose a **Beam** target.



2. Depending on whether you created a Surface, Volume, or Beam target, specify the target mesh using the *Target Surfaces*, *Target Volumes* or *Target Beams* property.

Target surfaces can either be 2D regions, region boundaries, or part surfaces in STAR-CCM+, or surfaces defined on an imported CAE model. Target volumes can either be STAR-CCM+ regions or parts, or volumes defined on an imported CAE model. You can specify multiple target surfaces or volumes at once.

For a beam target, specify a single target beam. Additional target beams require additional mappers. A target beam can only be an imported CAE beam model.

3. Set the *Target Stencil* property to specify the location on the target mesh where the data mapper interpolates the source data.

   For finite volume (FV) target meshes, the target stencil is typically either a cell (for target volumes) or a face (for target surfaces). For finite element (FE) target meshes, the target stencil is typically a vertex. When you map nodal rotations and displacements of an imported beam to target STAR-CCM+ surfaces, set the target surfaces stencil to **Vertex**. See Target Reference.

4. Specify the *Interpolation Method* that the mapper uses to interpolate data from the source to the target mesh.

   For more information on the available interpolation schemes, see Target Reference, and Data Mapper Formulation. For surface-to-surface coupling, the Exact and Approximate Imprinting schemes are globally conservative methods, but do not strictly satisfy local conservation. They result in more accurate solutions than the least squares methods when mapping from a fine to a coarser mesh. The least squares method is more accurate locally and is a better choice in cases where the target and source surfaces have comparable mesh densities. In terms of performance, the exact imprinting method is computationally more expensive than the least squares scheme, while the approximate method is of comparable efficiency to the least squares.

5. Specify the transform to project targets on the sources.

   For more information on the available transforms, see Target Source Transform Reference.

   Additional expert properties are available. For the complete list of available options, refer to the section, Target Reference.

Defining Constraint Checks

Optionally, enable constraint checks to specify criteria for mapping. Two constraint checks are available: the **Proximity Constraint** check, and the **Normal Direction Constraint** check, which is only available for surface-to-surface mapping. When you enable a constraint check, the mapper only maps field data to target points that pass the constraint check.

To enable constraint checks:

1. Select the relevant constraint checks, either **[Target]** > **Proximity Constraint**, or **[Target]** > **Normal Direction Constraint**, or both, and activate *Enable*.
2. When using a **Proximity Constraint** check, specify either an absolute or relative proximity tolerance. When using a **Normal Direction Constraint** check, specify the lower limit for the angle between the source and target normals.

   For more information, see Constraint Checks Reference.

Mappers do not interpolate data to target points that fail the proximity or normalcy checks. At these points, you can specify constant alternate values for the scalar or vector fields that you specified for mapping. To assign alternate values to vector fields:

3. Select the **[Target]** > **Alternate Value Specification** node.

4. Click to the right of *Vector Alternate Values*.
5. Enter the three components of each function as a comma-separated list contained in round brackets. List all the functions as a comma-separated list in the same order as the selected field functions. Place the entire list in square brackets.
   For example, to set the alternate value of Velocity to (1,0,0) and the Wall Shear Stress to (0,0,0):
   - If the *Selected Vector Fields* are listed as **[Velocity, Wall Shear Stress]**, then in the *Vector Alternate Values* type `[(1,0,0),(0,0,0)]`

To specify alternate values for scalar fields:

6. Click to the right of *Scalar Alternate Values*.
7. Enter the constant values for all the functions as a comma-separated list in the same order as the selected field functions. Place the entire list in square brackets.
   For example, to set the alternate value of Local Heat Transfer Coefficient to 0 and the Local Heat Transfer Reference Temperature to 300:
   - If the *Selected Scalar Fields* are listed as **[Local Heat Transfer Coefficient, Local Heat Transfer Reference Temperature]**, then in the *Scalar Alternate Values* type `[0.0,300]`.

   For more information, see Alternate Value Specification Reference.

Executing the Mapping Operation

Execute the mapping operation manually, or set up automatic mapping at a specified frequency during the simulation run. Optionally, pre-register the mapped fields so that you can access the corresponding field functions before mapping occurs.

To pre-register the mapped fields as field functions:

1. Right-click the **Data Mappers** node and select **Pre-Register Mapped Fields**.

   STAR-CCM+ generates empty field functions for each field that you specified for mapping in any data mappers. When you execute the mapping operation, STAR-CCM+ populates the empty field functions with the mapped data. This option allows you to set up all scenes and boundary conditions that use the mapped data field functions, before mapping occurs.

To set up automatic mapping:

2. Select the **[Data Mapper]** > **Update** node and activate *Enabled*.
3. Set *Trigger* to either **Iteration** or **Time-Step**, depending on whether you want to specify the mapping frequency using iterations or time-steps.
4. Select the **Update** > **Iteration Frequency** or **Time-Step Frequency** node, and specify the frequency at which mapping occurs.
5. Use the remaining properties to instruct the mapper to start/stop mapping data after a specified number of iterations, or time-steps.

   When you run the simulation, STAR-CCM+ executes the mapping operation based on the update specifications for the mapper.

Alternatively, you can execute the mapping operation manually:

6. Right-click the data mapper node and select **Map Data**.

If the *Verbosity* expert property is enabled, statistics of the neighborhood search and mapping are presented in the *Output Window* at the end of the mapping process.

If the solution on the source parts is updated, you can re-execute the mapper to update the mapped fields.

You can visualize the mapped fields in a scene, or use the mapped fields to define boundary conditions, by selecting the corresponding field functions.

Data Mappers Reference

Data mappers allow you to map data from source meshes or tables to target mesh representations.

The following sections provide information on all the properties and options available for data mappers.

You can set up data mappers under the **Tools** > **Data Mappers** node. The options available when you right-click the **Data Mappers** node allow you to create new mappers and to perform actions on all the existing data mappers at once:

Action Menu

New Data Mapper	Creates a data mapper. Choose the mapper type based on the type of source from which you want to map data.	
	Surface Data Mapper	Interpolates data from source surfaces to target surfaces, volumes, or beams.
	Volume Data Mapper	Interpolates data from source volumes to target surfaces or volumes.
	Beam Data Mapper	Interpolates data from a single source beam to a single or multiple target surfaces.
	Tabular Data Mapper	Interpolates data from an XYZ table to target surfaces, volumes, or beams.
Clear Mapped Fields	This action deletes any mapped fields and corresponding mapped data field functions generated by the mappers. If a mapped data field function is used by other objects in the simulation, such as scenes or reports, make sure that you remove any dependencies before using the **Clear Mapped Fields** option. If you do not remove the dependencies, any data associated with the mapped field function will be deleted but the mapped field function will not be removed from the simulation tree.	
Pre-Register Mapped Fields	By default, mapped fields become available as field functions when you execute the corresponding mapping operation. Use this option to pre-register the mapped fields, that you specified for any data mappers, as field functions, so that they are immediately available to other simulation objects such as reports or scenes.	

Contents:

Surface, Volume, Beam, and Tabular Data Mapper Reference
Target Reference
Constraint Checks Reference
Field Functions Reference
Specific CAE Properties

Surface, Volume, Beam, and Tabular Data Mapper Reference

This section provides information on the actions and properties available for all types of mappers. The data mapper properties include source settings and mapped field specifications.

Action Menu

Map Data	Executes the mapping operation.
Delete	Deletes the data mapper, without deleting any mapped data. To delete mapped data, first use the **Clear Mapped Fields** action, available for the parent **Data Mappers** node.

Surface, Volume, and Beam Data Mapper Properties

[Source]	Specifies the source from which to map data. The name of this property changes to one of the following, based on the type of data mapper:	
	Source Surfaces	For Surface Data Mappers, you can specify one or more surfaces from which to map data. Source surfaces can either be 2D regions, region boundaries, part surfaces, or surfaces defined on an imported CAE model.
	Source Volumes	For Volume Data Mappers, you can specify one or more volumes from which to map data. Source volumes can either be STAR-CCM+ regions or parts, or volumes defined on an imported CAE model.
	Source Beams	For Beam Data Mappers, you can specify a single beam from which to map data. The source beam can only be an imported CAE beam.
Source Stencil	Specifies the location where the solution data is available on the source mesh. If the solution available on the source mesh was computed using finite volume (FV) techniques, the source stencil is most likely either a cell or a face. If the solution available on the source mesh was computed using finite elements (FE) techniques, the source stencil is typically a vertex. Choosing the wrong combination of stencils reduces the integrity of the mapping operation.	
	Face	Instructs the data mapper to map field data available at the source mesh faces. Typically, this stencil is appropriate for surface data mappers that map data from surfaces of a finite volume mesh, for example boundaries or part surfaces of a STAR-CCM+ FV model.
	Cell	Instructs the data mapper to map field data available at the source mesh cell centroids. Typically, this stencil is appropriate for volume data mappers that map data from volumes (cells) of a finite volume mesh, for example regions or parts of a STAR-CCM+ FV model.
	Vertex	Instructs the data mapper to map field data available at the source mesh vertices. Typically, this stencil is appropriate for surface, volume, and beam data mappers that map data from a finite element source mesh. For example, part surfaces, parts, or regions of STAR-CCM+ FE models, surfaces or volumes of imported FE models, imported beams.

Scalar Field Functions	Specify scalar and/or vector fields for mapping. The data mapper interpolates the specified fields, available at the source faces, cells, or vertices as specified in the *Source Stencil* property, to the target mesh.
Vector Field Functions	
Mapped Names	The fields that you specify for mapping become available as field functions when you execute the data mapper or the **Pre-Register Mapped Fields** option. See Data Mappers Reference. The mapped data field functions have the default name, `Mapped<Target_stencil><Field_name>`. Using this property, you can specify/edit an alternative name for each of the field functions generated by the data mapper.
Representation	For STAR-CCM+ source meshes, this property allows you to select the representation to get source data from, which could be either the **Volume Mesh** representation or any representations available in a simulation history file. See Recording And Viewing Solutions. This property has no effect when the source mesh is an imported CAE model.
Use original mesh	When activated, instructs the data mapper to use the original mesh as the reference mesh for mapping. When deactivated, the data mapper uses the current mesh as the reference mesh for mapping. See Mapping Between Structure And Fluid Grids.
Verbosity	Controls the level of verbosity of the selected data mapper. When activated, the *Output* window displays additional details on the mapping process, such as the percentage of target source faces, cells, or vertices that is mapped successfully.

Tabular Data Mapper Properties

Source Table	Allows you to provide source data in tabular form. You can only select XYZ tables created in STAR-CCM+ (see Creating a Table), or file tables imported from external files (see Reading a Table Data File). The table must have at least four named columns. Three of the columns specify an x, y, and z location, and the fourth column specifies the value of a certain field at that location. Additional columns can specify the values of other scalar fields at the prescribed locations.
Source: X-Coordinate	Allows you to select the source table column that contains the X-coordinates.
Source: Y-Coordinate	Allows you to select the source table column that contains the Y-coordinates.
Source: Z-Coordinate	Allows you to select the source table column that contains the Z-coordinates.
Source: Data	Allows you to specify the source table column that contains the values of the field to map. You can select multiple columns to specify multiple fields for mapping.
Mapped Names	Allows you to specify/edit names for each of the output field functions generated by the data mapper for each of the *Source(s): Data* set.
Verbosity	Controls the level of verbosity of the selected data mapper. When activated, the *Output* window displays additional details on the mapping process, such as the percentage of target source faces, cells, or vertices that is mapped successfully.

Target Reference

This section provides information on the properties available for all types of mapping targets. A mapping operation can have several targets for each source.

Target Specifications Reference

Action Menu

New Group	Adds a new target of one of the following types.	
	Surface	Allows you to interpolate source data to target surfaces.
	Volume	Allows you to interpolate source data to target volumes.
	Beam	Allows you to interpolate source data to a target beam.

Surface, Volume, and Beam Target Reference

Properties

[Target]	Specifies the target to map data to. The name of this property changes to one of the following, based on the type of target:	
	Target Surfaces	For surface targets, you can specify one or more surfaces to map data to. The target surfaces can either be 2D regions, region boundaries, part surfaces, or surfaces defined on an imported CAE model.
	Target Volumes	For volume targets, you can specify one or more volumes to map data to. The target volumes can either be STAR-CCM+ regions or parts, or volumes defined on an imported CAE model.
	Target Beams	For beam targets, you can specify a single beam to map data to. The target beam can only be an imported CAE beam.
Target Stencil	Specifies the location on the target mesh where the mapper maps the solution data. When mapping to finite volume (FV) meshes, the target stencil is most likely either a cell or a face. When mapping to finite element (FE) meshes, the target stencil is typically a vertex. Choosing the wrong combination of stencils reduces the integrity of the mapping operation.	
	Face	The mapper interpolates the source data to the target mesh faces. Typically, this stencil is appropriate for target surfaces on a finite volume mesh, for example boundaries or part surfaces of a STAR-CCM+ FV model. This option is only available for surface targets.
	Cell	The mapper interpolates the source data to the target mesh cell centroids. Typically, this stencil is appropriate for target

		volumes with finite volume mesh, for example regions or parts of a STAR-CCM+ FV model. This option is only available for volume targets.
	Vertex	The mapper interpolates the source data to the target mesh vertices. Typically, this stencil is appropriate for target surfaces, volumes, and beams with finite element mesh. For example, part surfaces, parts, or regions of STAR-CCM + FE models, surfaces or volumes of imported FE models, imported beams. This option is available for all types of targets and is the only option available for beam targets. **Note:** When mapping beam data to surfaces, set the target surfaces stencil to **Vertex**, as the morpher uses the surface nodal displacements to update the mesh.
Use Original Mesh	When activated, instructs the data mapper to use the original mesh as the reference mesh for mapping. When deactivated, the data mapper uses the current mesh as the reference mesh for mapping. See Mapping Between Structure And Fluid Grids.	
Target Source Transform	Specifies the transform to project targets on the sources.	
	Identity Transform	Uses the original target points. This transform is available when mapping from any type of source to any type of target.
	Simple Transform	Rotates, translates, or scales the target points. This transform is available for surface-to-surface and volume-to-volume mapping. This option adds a subnode called **Simple Transform**. See Target Source Transform Reference.
	Normal Projection Transform	Projects the points in the normal direction onto the source surface This transform is available for surface-to-volume mapping.
	Helical Projection Transform	Projects the points onto the source surface using a helical path. This transform is available for surface-to-volume mapping. This option adds a subnode called **Helical Projection Transform**. See Target Source Transform Reference.
Boundary Map Option (only for volume targets)	For volume targets, this option determines how the mapper maps data from the source volumes to the boundary faces of the target volumes.	
	Map from Boundaries of Source Volumes	Use if the boundaries of the target and source volume coincide (even though the meshes are different).
	Map from Interior of Source Volumes	Use if the target volume is only a portion of a larger source volume.

		Note: For STAR-CCM+ Co-simulation, this option is the only option supported.
Limiter	If this option is beyond the range of the contributing stencil values, it specifies how the mapped value is limited.	
	Min/Max of Contributing Stencil	Limits the mapped value based on the values of the points chosen for interpolation.
	Min/Max of Contributing Sources	Limits the mapped values to the minimum/maximum of the source surface/volume. This option allows more extrapolation as compared to the first option.
	No Limiting. Allow Extrapolation	Allows extrapolation and does not limit the mapped value.
Interpolation Method	Specifies the data interpolation scheme. Each method uses reciprocity between connected faces to ensure two-way flows of data and enhances stability and robustness.	
	Shape Function	The interpolation stencil is determined using shape functions, which are special relations between vertices. This method is available for vertex source surface data location.
	Least Squares	The interpolation stencil includes the nearest neighbor and its immediate neighbors, weighted by the distance between the source and target data locations. If the source and target are coincident, the resultant weight at that location is one. This method is available for any source and target surface and volume data locations.
	Nearest Neighbor	The interpolation stencil includes only the nearest neighbor. The weight of the nearest neighbor in the stencil is 1, and is therefore the only value included in the interpolation. This method is available for any source and target surface and volume data locations.
	Exact Imprint	Uses a maximum radius of the target face to find the source faces to include in the interpolation stencil. The maximum radius is the distance from the target face centroid to one of its vertices. The target and source faces are intersected and the weights are the areas of the subfaces that match up between the target and sources. This method is available when mapping between surfaces. Since exact imprinting is computationally intensive to find exact intersections, this method relies on a proximity check to determine the stencil.
	Approximate Imprint	As with the exact imprint method, except that the weights include all source faces that are found within the maximum radius, and do not depend on the intersecting process. This method is available when mapping between surfaces.

Target Grouping Threshold (only for surface targets)	Data mappers group multiple target surfaces to perform mapping operations. The *Target Grouping Threshold* option allows you to specify the number of target surface faces per process that are searched by the mapper. A large input results in fewer searches, but may increase the memory usage.

Contents:

Target Source Transform Reference

Target Source Transform Reference

The target source transform is set up to define how the target entity is transformed into the source entity.

Helical Projection Transform Properties

Pitch	Specifies the pitch of the helix, which is the distance along the axis of one complete turn of the helix.
Helical axis	Specifies the axis of rotation of the helix.
Helical Axis Origin	Specifies the origin of the helix.

Simple Transform Properties

Rotation Angle (deg)	Specifies the angle of rotation about the rotation axis in degrees.
Scale	Specifies the scaling vector.
Rotation Origin	Specifies the axis of rotation origin.
Rotation Axis	Specifies the axis of rotation.
Translation	Specifies the translation vector.

Constraint Checks Reference

Set up constraint checks to specify criteria for mapping. If you enable a constraint check, the mapper only maps field data to target points that pass the constraint check. You can specify alternate field values at target points that fail the constraint checks.

Proximity Constraint Check Reference

The proximity check is relevant if the surfaces are physically separated from each other or have some mismatch to avoid mapping data. Such a situation is possible if the source parts do not completely represent some of the target parts. This scenario is often encountered in solid/fluid coupling, especially with external packages. For co-located surfaces, a proximity check is not required. When the proximity constraint is activated, the distance between parts is compared to a tolerance you define. If the distance is greater than this tolerance, an alternate value is used. Activate and set the proximity constraint in the properties of the **Target Specifications** > **[Target]** > **Proximity Constraint** node.

Note: Be aware of the general mesh sizes on parts while choosing this tolerance. If the mesh sizes vary extensively among the various selected parts, choose appropriate subsets of the parts for mapping.

Properties

Enable	When activated, the mapper checks if the target point and its nearest source point are within a certain distance. The mapper only maps field data to target points that pass the proximity check. You can specify alternate field values at target points that fail the proximity check. See Alternate Value Specification Reference.
Use Absolute Tolerance	When activated, the mapper checks if the target point and its nearest source point are within the value that you specified for the *Absolute Proximity Tolerance*. When deactivated, the mapper checks if the target point and its nearest source point are within the value specified for the *Relative Proximity Tolerance*, multiplied by the longest element edge-length of the target surface mesh.
Relative Proximity Tolerance	Specifies a multiplication factor for checking proximity. The tolerance is calculated by multiplying this factor by the longest element edge-length of the target surface mesh. For example, if the maximum target edge length is 0.05 mm and the source and target meshes are 0.1 mm apart, then a factor of 2 is required to enable mapping. The mapper uses the value when you activate *Enable*, and deactivate *Use Absolute Tolerance*.
Absolute Proximity Tolerance	Specifies a value for the tolerance for checking proximity. The mapper uses the value when you activate both *Enable* and *Use Absolute Tolerance*.

Normal Direction Constraint Check Reference

The normalcy check is relevant for surface-to-surface mapping. This check is intended to ensure that the surface normal at the target point faces the surface normal at the closest source point. Such a condition is true for solid-fluid interfaces, for instance. Also, for thin solids, this check avoids wrongly identifying the matching surface.

The normalcy tolerance is the lower limit of the angle between surface normals at the target point and the corresponding source point. While ideally this angle is 180 degrees, account for differences in surface meshes on either side and also for differences in the modeling itself. A tolerance of 90 degrees or greater means that the normals face each other. Enable and set the normal direction constraint in the properties of the **Target Specifications** > **Surface** > **Normal Direction Constraint** node.

Properties

Enable	When activated, the mapper checks if the angle between surface normals at the target point and the corresponding source point is greater than the value specified within *Minimum angle (degrees)*. This check only applies to surface-to-surface mapping. The mapper only maps field data to target points that pass the check. You can specify alternate field values at target points that fail the normal direction constraint check. See Alternate Value Specification Reference.

Minimum angle (degrees)	Specifies the lower limit for the angle between source and target normals.

Contents:

Alternate Value Specification Reference

Alternate Value Specification Reference

Data is not mapped to target points that fail the proximity or normalcy checks. Choose to assign constant alternate values for each field to these points. Specify the alternate value in the properties of the **Target Specifications** > **Surface/Volume** > **Alternate Value Specification** node. The field functions you selected for mapping populate the relevant property fields of the **Alternate Values Specification** node.

Note: Be careful while specifying alternate values. In STAR-CCM+, the order of the values is tied to the order in which the selected fields are listed. This rule applies to both scalar and vector alternate values.

Properties

Selected Scalar Fields	This read-only property lists the scalar fields for mapping.	
Scalar Alternate Values	Enter an alternate value for each of the chosen scalar fields to be assigned to target points that are not mapped. Enter the constant values for all the functions as a comma-separated list in the same order as the selected field functions. Place the entire list in square brackets. This property is relevant only if the proximity check (or normalcy check for the surface mapper) option is activated. See Defining Constraint Checks.	
Selected Vector Fields	This read-only property lists the scalar fields for mapping.	
Vector Alternate Values	Enter an alternate value in vector notation for each of the chosen vector fields to be assigned to target points that are not mapped. Enter the three components of each function as a comma-separated list contained in round brackets. List all the functions as a comma-separated list in the same order as the selected field functions. Place the entire list in square brackets. This property is relevant only if the *Proximity Check* (or *Normalcy Check* for the surface mapper) option is activated. See Defining Constraint Checks.	
Estimate Alternate Values	Controls whether to estimate an alternate value for unmapped targets. These options only appear for target surfaces.	
	Do Not Estimate Alternate Values	The specified alternate values are used.
	Estimate For All Unmapped Patches	An alternate value is estimated by averaging the surrounding mapped targets. If this fails, the specified alternate values are used.

	Max. Patch Face Area	An alternate value is estimated by averaging the surrounding mapped targets, limited to a maximum patch area. A **Maximum Unmapped Patch Area** subnode is created. If this mapping fails, the specified alternate values are used.
	Max. Patch Face Count	An alternate value is estimated by averaging the surrounding mapped targets, limited to a specified number of surrounding faces. A **Maximum Unmapped Face Count** subnode is created. If this fails, the specified alternate values are used.

Field Functions Reference

When you execute a data mapper, STAR-CCM+ creates field functions that allow you to access the mapped data and to determine whether the mapping process was successful.

Mapped Flag, Mapped Vertex Flag

These field functions allow you to determine where, on the target mesh, the mapping process was successful. The functions also show you where the process failed. When you execute a data mapper, STAR-CCM+ creates either one of these field functions, depending on whether the target stencil is **Cell**/**Face** (**Mapped Flag**), or **Vertex** (**Mapped Vertex Flag**).

To use these field functions, plot the relevant function on the target mesh in a scalar scene. At each cell, face, or vertex, the field function assumes a discrete value that identifies the success or failure of the mapping process for that face, cell, or vertex:

`0`—Successfully mapped.
`1`—Failed proximity constraint check.
`2`—Failed normal direction constraint check.
`3`—Failed imprinting.
`4`—Failed mapping. Applied estimated alternate values.
`5`—Not mapped. This value is only relevant for data mappers using geometry parts as targets. When the targets are geometry parts, some cells/faces on the corresponding region are not used in the mapping process.

Based on the mapping operation outcome, you can choose appropriate values for the tolerances.

When you execute more than one mapping operation, either from the same data mapper or from different data mappers, STAR-CCM+ updates the field function values to represent the latest mapping process.

[Mapped Data]

STAR-CCM+ stores the mapped fields in field functions that are created when you execute the data mapper or the **Pre-Register Mapped Fields** action. Mapped data field functions have the default name, `mapped<Target_stencil><Field_name>`, or a custom name specified using the *Mapped Names* property for the mapper. See Data Mappers Reference.

Specific CAE Properties

Data mappers are created manually or by invoking a map-action on surfaces or volumes of imported models. The map-action dialog has special properties depending on the type of surfaces and the data type chosen for mapping.

Heat Transfer Export Options Reference

In coupling with CAE packages for Conjugate Heat Transfer applications, the CAE model is imported into the STAR-CCM+ simulation. Typically, the Local Heat Transfer Coefficient (HTC) is mapped from the STAR-CCM+ fluid boundary to the corresponding solid surface from the imported CAE model. As a fluid temperature is required to calculate the heat flux, the Local Heat Transfer Reference Temperature is mapped and exported along with HTC as well. The map-action dialog also allows you to specify a constant bulk temperature as the reference temperature. The dialog also allows you to choose the cell-center temperature in case radiation is being solved in both the solid and the fluid. The surface data mapper displays a **Heat Transfer Export Options** subnode when appropriate.

The **Heat Transfer Export Options** node is displayed if you choose all of the following:

- STAR-CCM+ boundaries as a source surfaces
- Imported Surfaces as target surfaces
- The source and target stencils set at Face
- Local Heat Transfer Coefficient included among the Scalar field functions for mapping.

The HTC export options can be specified by selecting the node in the object tree and working with the properties.

You cannot activate both the *Use Near Wall Cell Temperature* and *Use Specified Bulk Temperature* options at the same time. However, if neither one is activated, the field function for Local Heat Transfer Reference Temperature is automatically appended to the list of scalar field functions for mapping.

Properties

Use Near Wall Cell Temperature	Accounts for the inclusion of radiation in the simulation. When activated, STAR-CCM+ exports cell center temperature as the reference temperature. This is applicable in case radiation is being solved in both the STAR-CCM+ and CAE packages.
Use Specified Bulk Temperature	Controls the export of the bulk temperature. When activated, STAR-CCM+ exports a constant bulk temperature as the reference temperature.
Bulk Temperature	Specifies a value for bulk temperature.

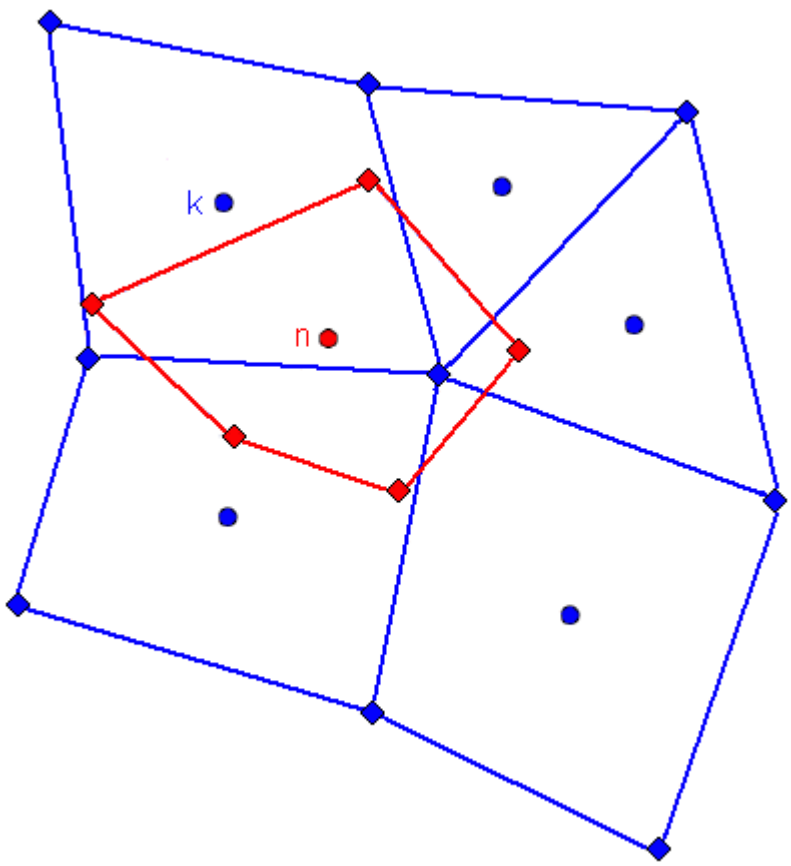
Data Mapper Formulation

Data mappers use various interpolation schemes to transfer source data to target meshes. The schemes vary depending on the choice of source stencil.

The following sections describe the various interpolation schemes. The shape function scheme is illustrated in the context of vertex to face mapping and all the other schemes are illustrated in the context of face to face mapping.

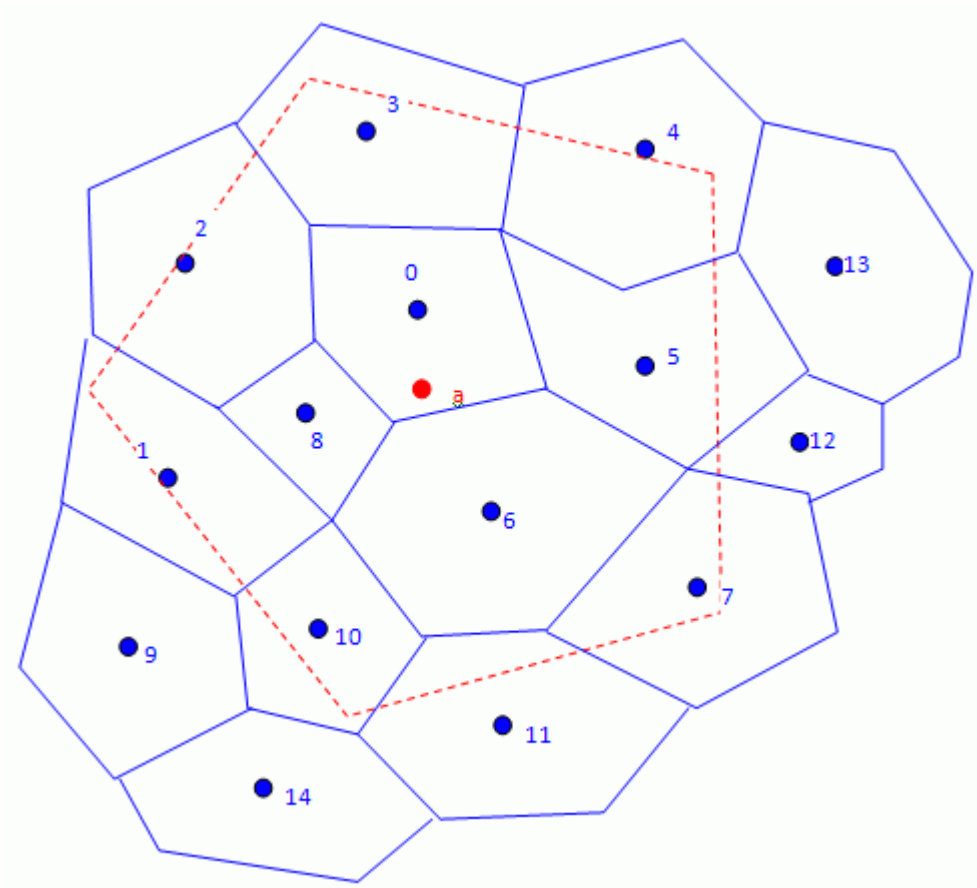
Nearest Neighbor

In the figure below, consider mapping data from faces of the red background mesh to faces of the blue mesh using a nearest neighbor scheme. **Face k** is the closest point to the target **Face n**. The data from **Face k** is assigned to **Face n** as the mapped value.



Least Squares Interpolation

In the figure below, consider mapping data from faces of the blue background mesh to faces of the red mesh using a least squares scheme. **Face 0** is the closest source to the target **Face a**. The neighbors of **Face 0** are defined as any face that shares at least one vertex with **Face 0**. These neighbors are included as part of the interpolation stencil. In the example below the neighbors include faces **0**, **2**, **3**, **4**, **5**, **6**, and **8**.



STAR-CCM+ uses the second-order terms of a Taylor series expansion as a “cost function” to approximate the error of the temperature distribution at point a. It then minimizes the cost function in Eqn. (1406) to solve for the temperature at the target point a.

$$J = \sum_{n \in F(c)} \left(T + \left(\mathbf{x}_n - \mathbf{x}\right) \cdot \nabla T - T_n\right)^2 \tag{1406}$$

$$\nabla T = \begin{cases} \dfrac{\partial T}{\partial t_1}\mathbf{t}_1 + \dfrac{\partial T}{\partial t_2}\mathbf{t}_2 & \text{for surfaces} \\ \dfrac{\partial T}{\partial x}\mathbf{t}_x + \dfrac{\partial T}{\partial y}\mathbf{t}_y + \dfrac{\partial T}{\partial z}\mathbf{t}_z & \text{for volumes} \end{cases} \tag{1407}$$

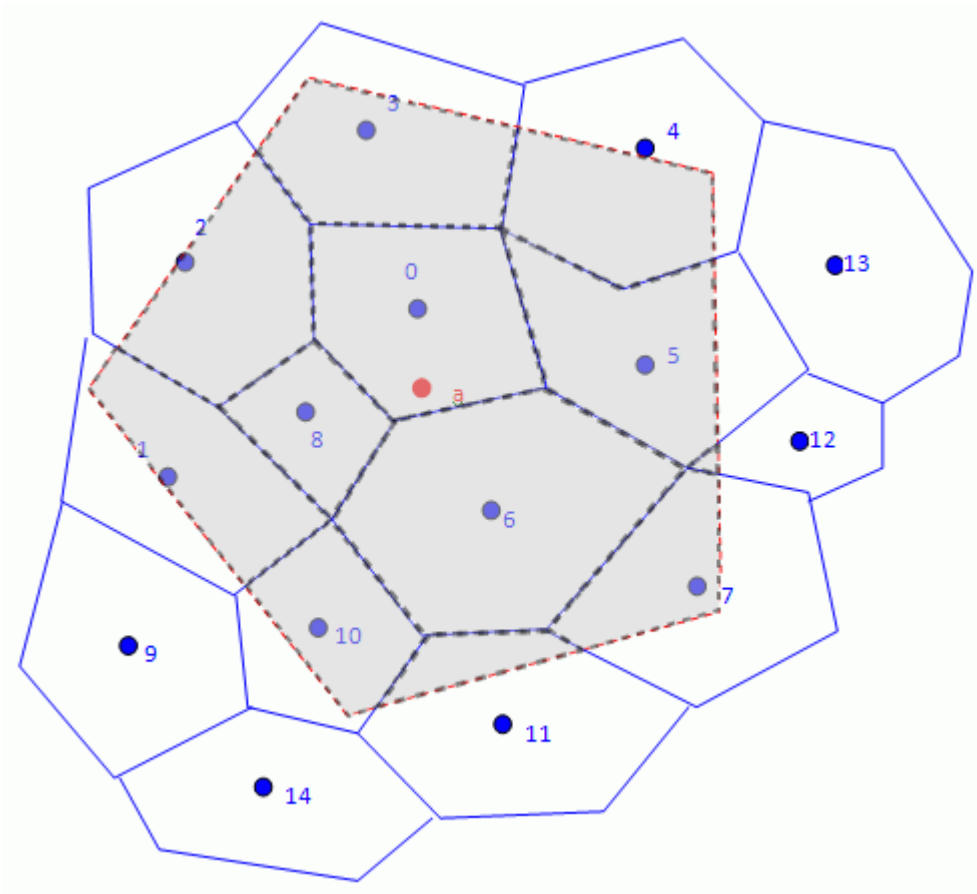
where $\mathbf{t}_1$ and $\mathbf{t}_2$ and are surface tangents.

The drawback of a neighbor-based least squares scheme is that some of the blue faces (for example 1, 10, 11, 7) that the target face imprints upon are not included in the stencil.

Note: The least squares method is applicable for all element types and for source and vertex source stencils.

Exact Imprinting Interpolation

In the figure below, consider mapping data from faces of the blue background mesh to faces of the red mesh using the exact imprinting scheme. In this method, the target red face is imprinted on the background blue mesh. This results in several facelets that are the intersection of the background mesh (faces **0**, **1**, **2**, **3**, **4**, **5**, **6**, **7**, **8**, **10**, **11**) with the target face.



The normalized areas of these facelets are used as interpolation weights in evaluating the mapped value at face-center a. This method is preferable when:

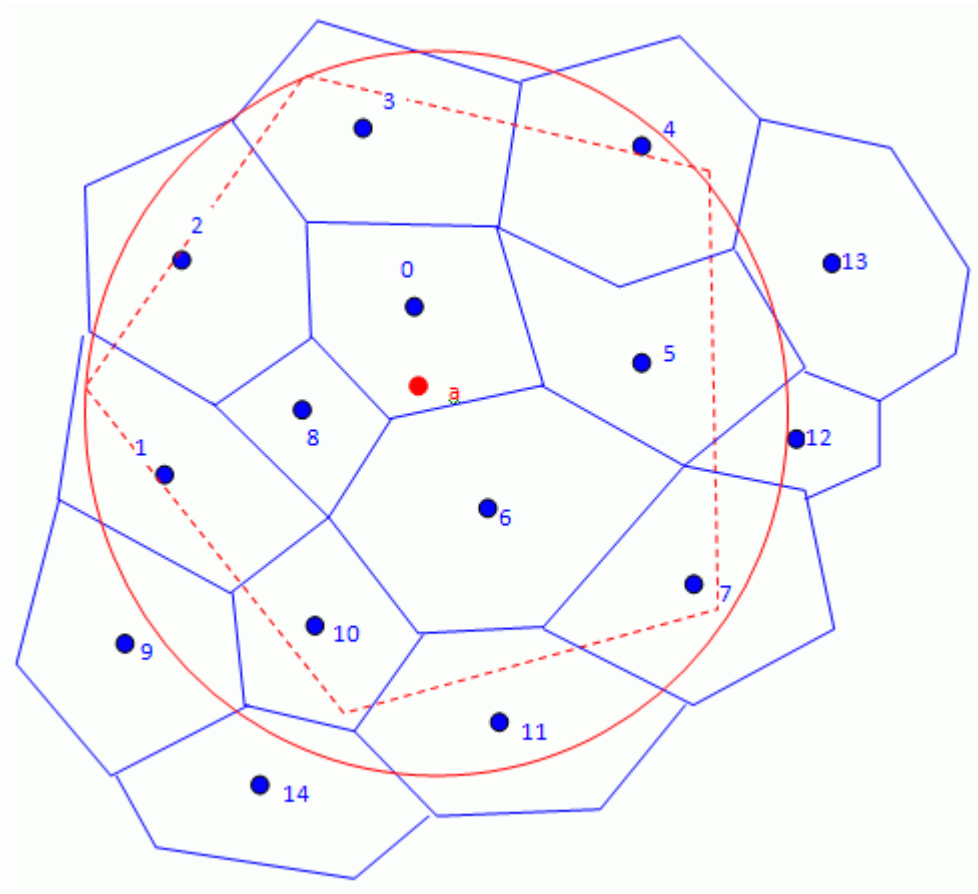
- a flux-preserving mapping is desired
- the source mesh is finer than the target mesh and the exact imprint does not fail to map.

Unlike the least squares scheme, all the source faces that the target face imprints upon are included in the stencil.

Mapping with the exact imprint method can fail when surface curvature is inadequately resolved in either the source or target mesh.

Approximate Imprinting Method

In the figure below, consider mapping data from faces of the blue background mesh to the red face using an approximate imprinting method. This method is a variation of exact imprinting. Since exact imprinting is computationally intensive to find exact intersections, this method relies on a proximity check to determine the stencil. The stencil includes all source faces that lie within a circle that is centered at the target face-centroid and passes through the farthest vertex of the target face. In the figure, the circle includes faces **0**, **1**, **2**, **3**, **4**, **5**, **6**, **7**, **8**, **9**, **10**, **11**, **12**, **13** and **14** as part of the stencil.

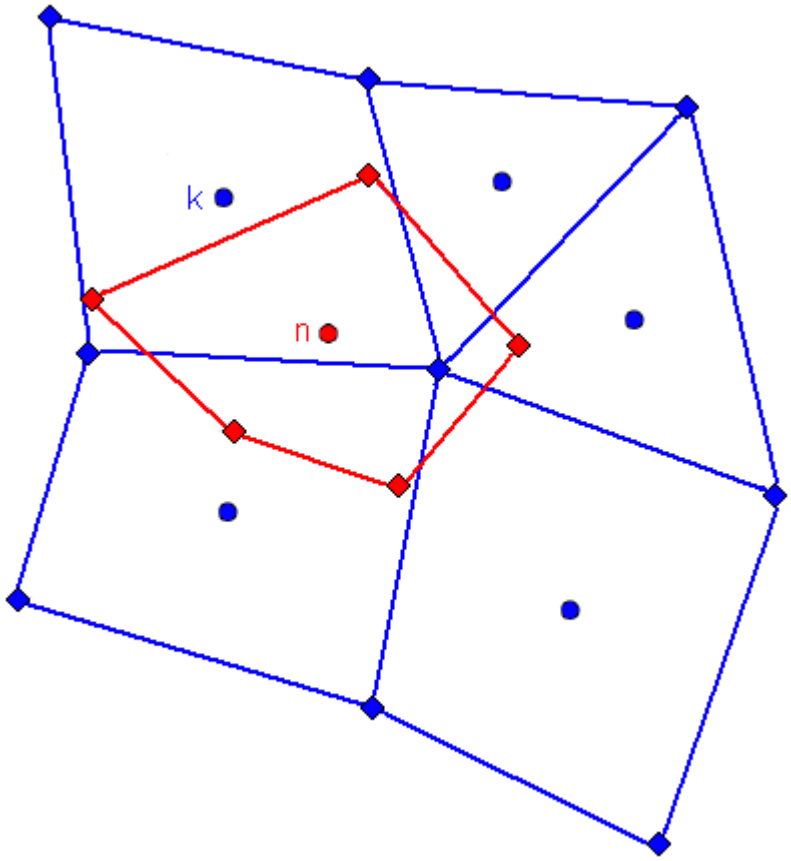


Exact imprinting would have eliminated faces **9**, **13** and **14** from the stencil because they do not intersect the target face. As for exact imprinting, the normalized face areas are used as interpolation weights in this case.

Although the approximate imprinting method is more robust than the exact imprinting method, it does not preserve flux quantities. The approximate method has comparable accuracy to the exact method when the source mesh is much finer than the target mesh.

Shape Function Interpolation

In the figure, consider mapping data from vertices of the blue source mesh to faces of the red target mesh using shape function interpolation. This scenario is typical when mapping data from a finite-element mesh to a finite-volume mesh. The face centroid n lies within the element k. The shape functions of element k are used to interpolate data to the point n.



The general form of shape function interpolant is:

$$\mathrm{T}_n = \sum_{l \in N(k)} \mathrm{T}_l N_l\left(\xi_n, \eta_n, \chi_n\right) \tag{1408}$$

where N is the shape function for the specific element type and (ξ_n, η_n, χ_n) are the computational coordinates of the point n in the element k

For a four-node isoparametric quadrilateral in 2D, the shape function for each node is:

$$N_1 = \frac{(1-\xi)(1-\eta)}{4} \tag{1409}$$

$$N_2 = \frac{(1+\xi)(1-\eta)}{4} \tag{1410}$$

$$N_3 = \frac{(1+\xi)(1+\eta)}{4} \tag{1411}$$

$$N_4 = \frac{(1-\xi)(1+\eta)}{4} \tag{1412}$$

For a three-node isoparametric triangle in 3D, the shape function at each node is:

$$N_1(\xi, \eta) = 1 - \xi - \eta \tag{1413}$$

$$N_2(\xi,\eta) = \xi \tag{1414}$$

$$N_3(\xi,\eta) = \eta \tag{1415}$$

Note: The shape function scheme is applicable only for triangle and quadrilateral elements in 2D and tetrahedral and hexahedral elements in 3D.

When mapping with shape functions from a FEA mesh to a STAR-CCM+ mesh, STAR-CCM+ recognizes the element type and uses the relevant shape function for interpolation. STAR-CCM+ supports higher order elements with mid-side nodes for Abaqus and other FEA codes.

When projecting points from STAR-CCM+ to the FEA mesh, however, STAR-CCM+ only uses corner nodes for the parametric map. Thus the mapper projects to Finite Element triangle faces as if they are flat. Projecting to Finite Element quad faces accounts for warping if the corner nodes do not lie in one plane.

Examples of Using Data Mappers

This section provides two examples of multiphysics applications where you can use data mapping for explicit coupling.

- Explicit coupling within STAR-CCM+ (Conjugate Heat Transfer) -- Thermal analysis of a solid with flow-thermal analysis of fluid, for instance in under-hood analysis. The CFD in the fluid region can be solved in STAR-CCM+ while the solid region is analyzed either in STAR-CCM+ or in an external CAE package.
- Explicit coupling between STAR-CCM+ and Abaqus (Fluid Structure Interaction) -- Stress analysis of a solid coupled with CFD in fluid, for instance analyzing the flow around an F1 wing. In this case, STAR-CCM+ solves the CFD and the structural analysis is either done using STAR-CCM+ or an external CAE package.

Explicit Coupling Within STAR-CCM+

Consider a conjugate heat transfer (CHT) application in which both the solid and fluid regions are created and meshed in STAR-CCM+. Instead of creating an interface as is traditionally done for implicit coupling, you can solve the fluid and solid regions separately as follows:

1. Set up physics continua by selecting all the appropriate models to solve for flow and energy in the fluid and heat transfer in the solid.

   The idea is to solve solid and fluid separately, but couple them by exchanging boundary data at the common boundary.
2. Create two surface data mappers from the **Tools** > **Data Mappers** node, one for mapping from solid to fluid (S2F) and the other for mapping from fluid to solid (F2S).

   In CHT applications, the data for exchange includes:
   - **Local Heat Transfer Coefficient**, and **Local Heat Transfer Reference Temperature** from fluid to solid
   - Surface **Temperature** from solid to fluid
3. For the S2F mapper, choose the solid boundary as the source surface and the fluid boundary as the target surface, and select face stencils for both source and targets. In addition, choose **Temperature** as a scalar field for mapping.
4. For the F2S mapper, choose the fluid boundary to be the source and the solid boundary to be the target, and again do face-to-face mapping. The scalar fields in this case are htc and Tref.

   The mappers are now set up to map data.
5. Map once to create Mapped field functions.
6. Use these Mapped field functions as boundary conditions:
   - For the solid boundary, choose **Convection** as the *Thermal Specification* and choose **Field Function** as the *Method* to apply heat transfer coefficient and ambient temperature. Then select the **MappedLocalHeatTransferCoefficient** and **MappedLocalHeatTransferReferenceTemperature** as the respective field functions.
   - For the fluid boundary, choose to apply the Dirichlet boundary condition using the **Field Function** method. In this case choose **MappedTemperature** as the applied field function.

   When the solver iterates, these fields are applied as boundary conditions.

A java macro can be created to take the appropriate number of steps before performing a map operation. Perform a map operation to update the Mapped field functions. The coupling frequency can be determined based on the nature of the problem.

Explicit Coupling Between STAR-CCM+ and Abaqus

Consider an FSI analysis in which the solid region is modeled in Abaqus and the fluid region is modeled in STAR-CCM+. The fluid model is created and meshed in STAR-CCM+ as usual. The solid model is created in Abaqus.

1. Set up the fluid physics continuum and specify appropriate solver properties. Also set up the morpher to account for solid displacement, as required for an FSI analysis.
2. After setting up the Abaqus model, create an `.inp` file as well as a database file (`.odb`) with initial nodal displacements.
3. Import the Abaqus `.inp` file into the simulation.
4. Identify the interfacing surface pairs.
5. Create two surface data mappers from the **Tools** > **Data Mappers** node, one for mapping from solid to fluid (S2F) and the other for mapping from fluid to solid (F2S).
6. In FSI cases, the exchanged data are:
   - Pressure and Wall Shear Stress from fluid to solid
   - Surface Displacement from solid to fluid
7. For the S2F mapper, choose the Abaqus surface as the source surface and the fluid boundary as the target surface, and choose the vertex source stencil and the face target stencil. In addition, choose **Incremental Surface Displacement** as a mapped vector field (this field is created when nodal displacements are imported from Abaqus).
8. For the F2S mapper, choose the fluid boundary to be the source and the solid boundary to be the target, and choose face stencils for both source and targets. Choose **Pressure** as a scalar field and **Wall Shear Stress** as a vector field for mapping.

   The mappers are now set up to map data.
9. Map using the S2F mapper to create Mapped field functions.
10. Use these Mapped field functions as boundary conditions:
    - For the fluid boundary, choose **Displacement** as the Morpher specification. The displacement is applied using the mapped field function **MappedIncrementalSurfaceDisplacement**.
    - For the Abaqus imported surface, **MappedPressure** and **MappedWallShearStress** are exported to a separate `.inp` file and applied as loads on the Abaqus model. The details of using Abaqus are beyond the scope of this section.
11. You can create a java macro to control the frequency of coupling depending on the nature of the problem. However, each time the coupling happens, perform the following steps:
    - Import new displacement data from Abaqus.
    - Map data to the STAR-CCM+ boundary.
    - Map data from the fluid boundary to the imported surface.
    - Export the mapped data to the `.inp` file.
    - Apply the load on the Abaqus model.

The STAR-CCM+ Component for Isight

Your STAR-CCM+ installation contains a Java component that can be used within the Isight process automation suite from SIMULIA. The component is located:

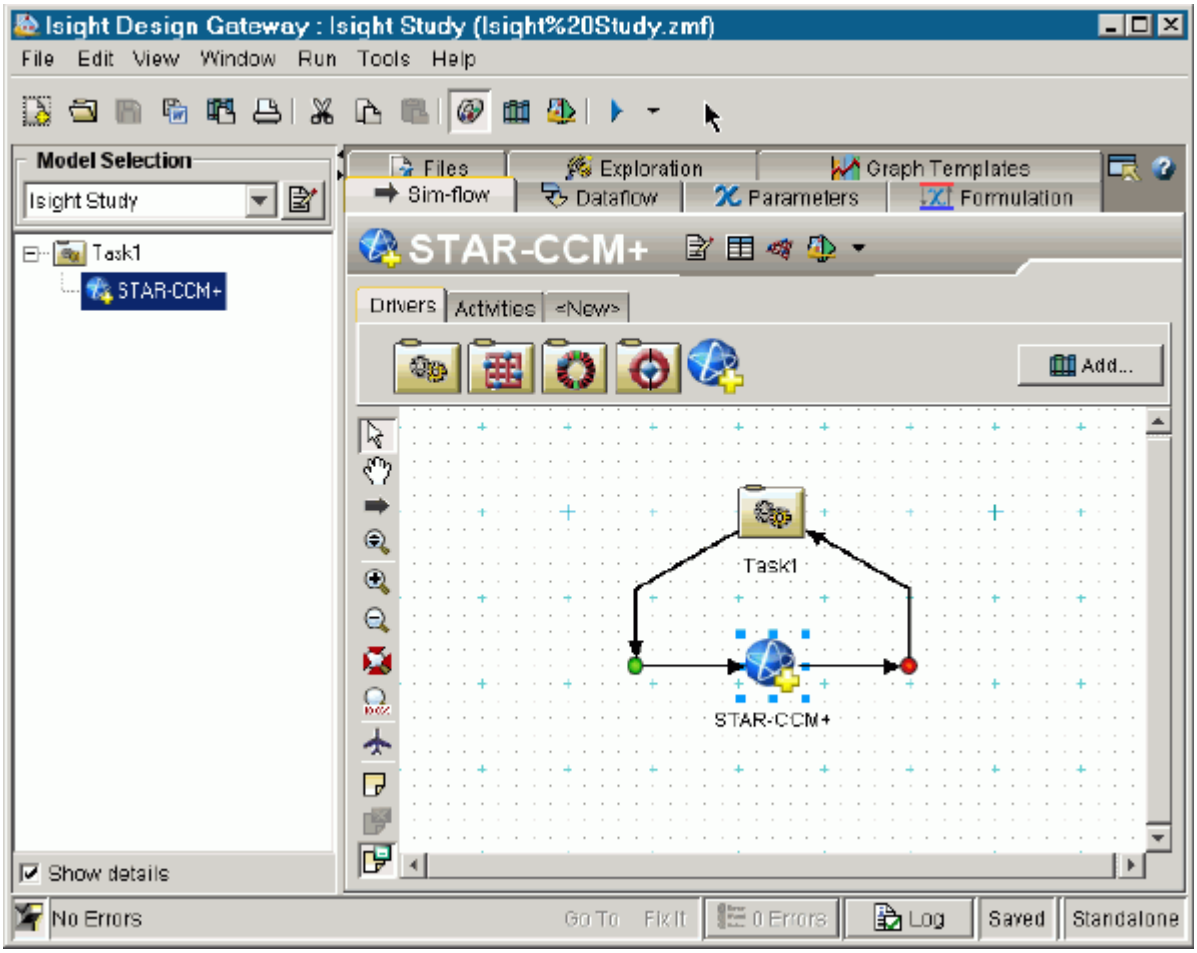
Linux:

```
[INSTALL_DIR]/isight/component/6.02/starccm.jar
```

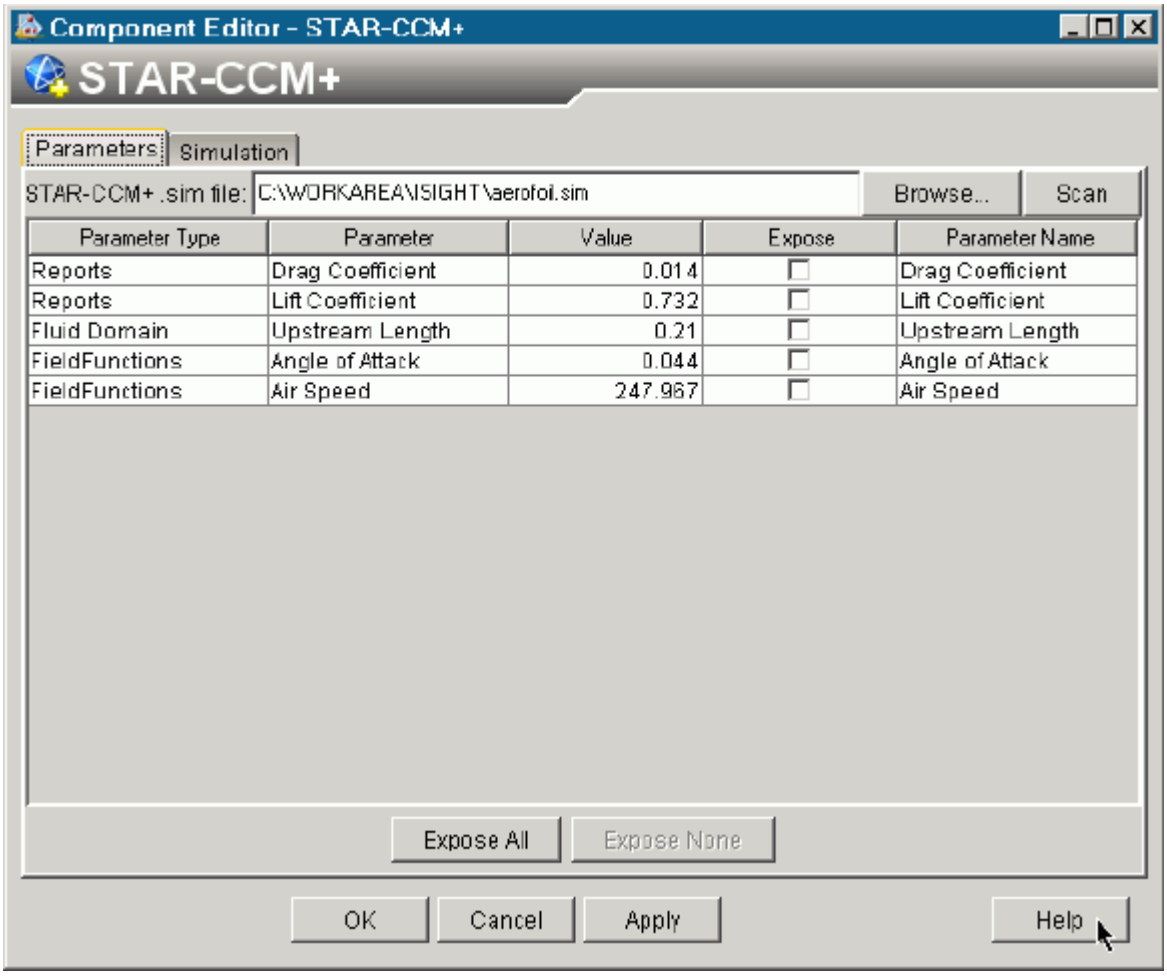
Windows:

```
[INSTALL_DIR]\isight\component\6.02\starccm.jar
```

Add this component to your Isight Design Gateway using the Isight Library. It becomes available for selection in the Isight interface when it is successfully published.



Documentation for the component can be accessed by clicking **Help** on the *Component Editor* dialog.


Component Editor - STAR-CCM+
STAR-CCM+
Parameters
Simulation
STAR-CCM+ .sim file: C:\WORKAREA\ISIGHT\aerofoi.sim
Browse...
Scan
Parameter Type | Parameter | Value | Expose | Parameter Name
Reports | Drag Coefficient | 0.014 | Drag Coefficient
Reports | Lift Coefficient | 0.732 | Lift Coefficient
Fluid Domain | Upstream Length | 0.21 | Upstream Length
FieldFunctions | Angle of Attack | 0.044 | Angle of Attack
FieldFunctions | Air Speed | 247.967 | Air Speed
Expose All
Expose None
OK
Cancel
Apply
Help

Add-ons

Enter a short description of your concept here (optional).

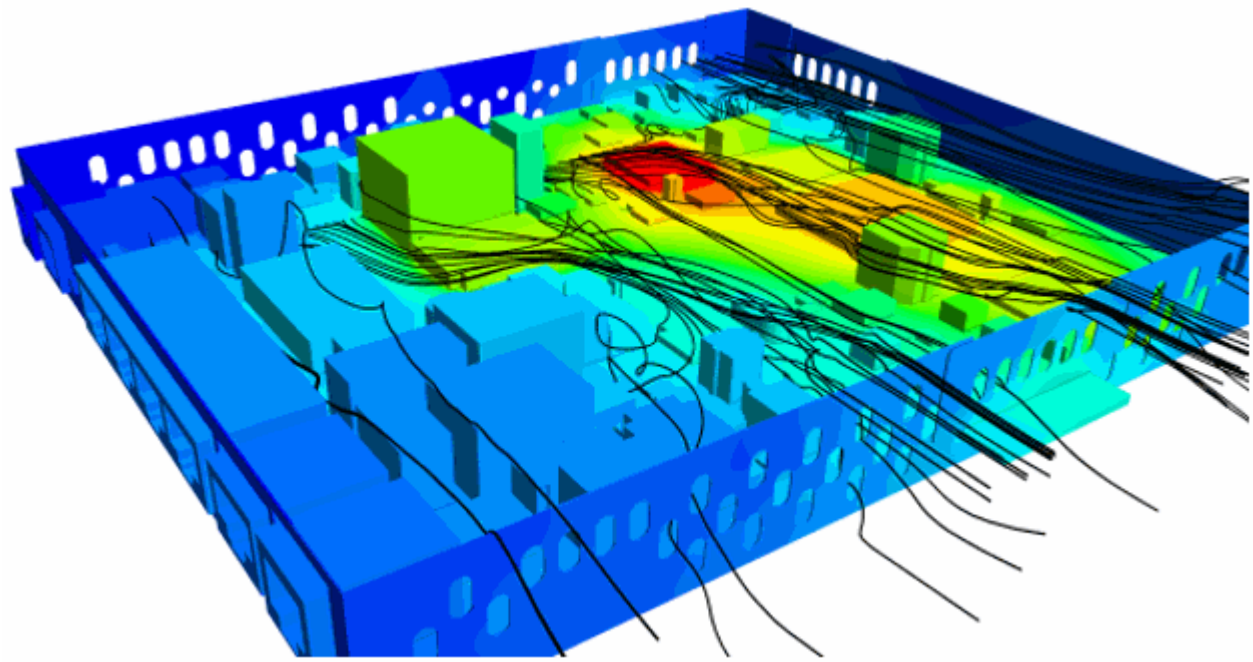
This is the start of your concept.

Contents:

Electronics Cooling Toolset
Optimate User Guide
Optimate Tutorials
STAR-ICE
Casting Simulation with STAR-Cast

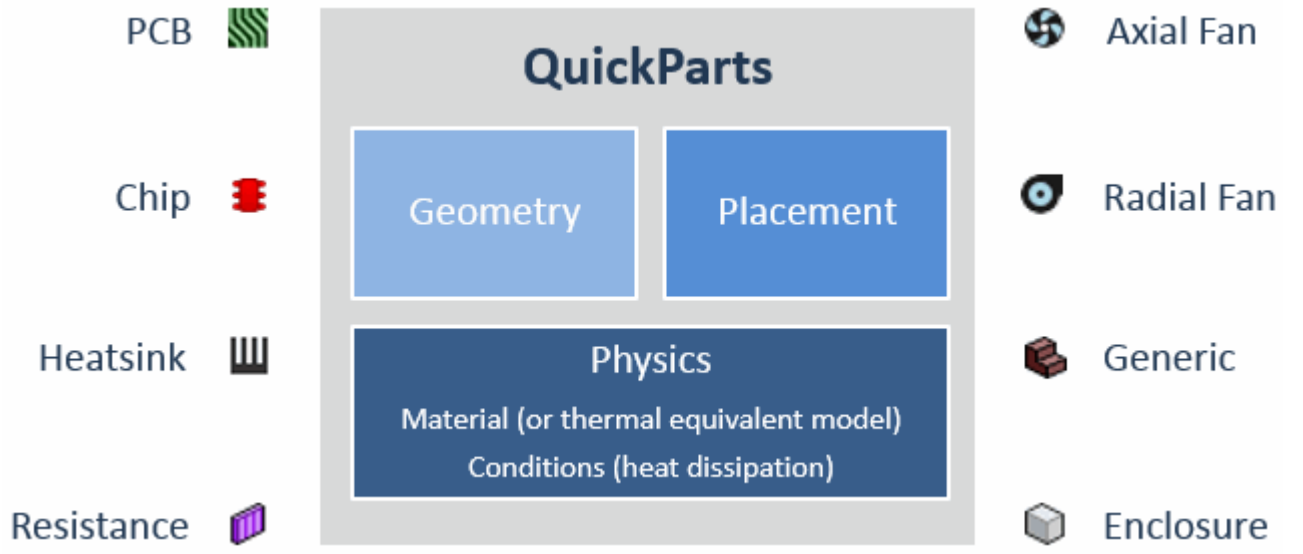
Electronics Cooling Toolset

The Electronics Cooling Toolset provides a simplified approach within STAR-CCM+ to perform simulations of thermal management for electronics devices—from design to solution analysis.



The Electronics Cooling Toolset simplifies and speeds up the simulation setup of typical electronics cooling problems in STAR-CCM+ by accessing a subset of features that are relevant to a thermal management simulation.

A key feature of the Electronics Cooling Toolset is the provision of QuickParts. QuickParts allow you to define the geometry, placement, and physics of components in an efficient and compact way. QuickParts are available for the most common components of an electronics device, such as PCBs, chips, heatsinks, or fans. Geometry templates enable you to build these components without requiring an external CAD tool. Additionally, interactive dimensioning and positioning features reduce the time spent on component design and layout.



For chips, the Electronics Cooling Toolset provides a compact thermal modeling approach. Compact thermal models (CTMs) represent the critical heat flow paths within chip packages using thermal resistor networks. CTMs do not resolve the internal volume of chip packages and thereby reduce computational costs for complex devices.

The user interface of the Electronics Cooling Toolset consists of several ordered nodes that you work through sequentially. Each node represents a specific stage in the workflow of setting up the simulation. In the background, the Electronics Cooling Toolset defines the models and parameters that are required for the analysis in the STAR-CCM+ simulation tree.

You do not require a detailed knowledge of CFD or STAR-CCM+ in order to use the tool. However, you still have access to the full functionality of STAR-CCM+ at any stage if you want to modify your model or account for more complex scenarios.

Note: The Electronics Cooling Toolset is part of the STAR-CCM+ installation and does not require any specific licensing.

Inside the Electronics Cooling Toolset

The main features of the Electronics Cooling Toolset are:

Geometry

- Geometry generation that is based on simple shape templates
- Import of IDF (Intermediate Data Format) files
- Import of CAD models from other CAD packages
- Transfer of CAD data from the STAR-CCM+ CAD modeler

Physics

- Time: Steady-state calculations
- Solution domains: multi-domain (fluid and solid)
- Heat transfer modes: conduction, convection (forced and buoyancy driven), and radiation (wavelength independent Surface-to-Surface (S2S) radiative heat transfer)

- Flow regime: laminar or turbulent (Realizable K-Epsilon Two-Layer All Y+)
- Equation of state: constant density or ideal gas

Conditions

- Environment: ambient boundary conditions
- Thermal specification: specification of material or specification of layers (PCB)
- Compact thermal models: resistor networks for chips
- Heat dissipation: volumetric heat sources

Mesh

- Volume Mesh: Trimmed cell mesher generates predominantly hexahedral cells
- Prism Layers: prismatic cells on walls for boundary flow accuracy
- Cell size: automatic calculation of target and minimum cell size that is based on overall model size
- Mesh Refinement: integrated features for mesh refinement to detect and capture complex geometries
- Local mesh controls: cell size customization on surfaces and within volumetric zones

Solve

- Segregated solver for flow and energy
- Solver under-relaxation
- Live monitoring of solution data

Results

- Visualization of geometry, scalar, and vector quantities
- Access to solution data through streamlines, isosurfaces, and plane sections
- Residuals and XY plots
- Summary reports

Contents:

Accessing the Electronics Cooling Toolset
Setting up an Electronics Cooling Simulation
Electronics Cooling Toolset Reference
Resistor Networks Formulation
Bibliography

Accessing the Electronics Cooling Toolset

The Electronics Cooling Toolset is accessible by clicking a customized toolbar button in STAR-CCM+.

To access the Electronics Cooling Toolset:

1. Open STAR-CCM+ and start a new simulation.
2. In the STAR-CCM+ menu, select **Window** > **Toolbars** > **Customize.**
3. In the *Customize Toolbars* dialog, under the **Wizards** node, select **Launch Electronics Cooling App**.
4. Drag the (**Launch Electronics Cooling App**) icon onto one of the toolbars at the top of the STAR-CCM+ workspace and drop it at an appropriate position.
5. In the *Customize Toolbars* dialog, click **Close**.

   The STAR-CCM+ GUI now displays the (**Launch Electronics Cooling App**) button.

The Electronics Cooling Toolset provides an inherent file compatibility with STAR-CCM+—all setup and solution data are stored within the `.sim` file.

Objective	Procedure
Launching the Electronics Cooling Toolset	1. Launch STAR-CCM+ and click (**Create a New Simulation**). 2. In the *Create a New Simulation* dialog, set the startup options to suit your requirements (see Interactive Startup Reference) and click **OK**. 3. In the STAR-CCM+ toolbar, click (**Launch Electronics Cooling App**). 4. The Electronics Cooling Toolset dialog appears in which you can set up an electronics cooling simulation.
Loading an existing electronics cooling simulation	1. Launch STAR-CCM+ and click (**Load Simulation...**). 2. In the *Load Simulation* dialog, click **Browse...**. 3. In the *Open* dialog, navigate to the working directory, select the appropriate `.sim` file, and click **Open**. 4. Complete the *Load Simulation* dialog according to the properties described in Interactive Startup Reference and click **OK**. 5. In the STAR-CCM+ toolbar, click (**Launch Electronics Cooling App**).
Saving an electronics cooling simulation	You can save an electronics cooling simulation at any stage.

Objective	Procedure
	1. Click (**Save**). The Save dialog that appears allows you to save a `.sim` file.

Note: In this context, unless otherwise stated, the term "electronics cooling simulation" refers to a simulation that is set up with the Electronics Cooling Toolset.

Setting up an Electronics Cooling Simulation

The following steps outline the recommended workflow to analyse electronic devices using the Electronics Cooling Toolset in STAR-CCM+.

To set up a thermal management simulation of your electronics device, work through the nodes of the Electronics Cooling Toolset from top to bottom. The workflow to follow is:

1. Defining the Setup
2. Defining Parameters
3. Creating and Positioning the Components
4. Adding an Enclosure and Fans
5. Extracting Fluid Volumes
6. Setting Boundary and Interface Conditions
7. Setting Up the Mesh
8. Preparing the Post-Processing
9. Running the Simulation
10. Analyzing the Results

For practical examples of electronics cooling analysis using the Electronics Cooling Toolset, refer to the following tutorials:

- Electronics Cooling Toolset: Natural Convection JEDEC
- Electronics Cooling Toolset: Graphics Card Cooling

Contents:

Defining the Setup
Defining Parameters
Creating and Positioning the Components
Adding an Enclosure and Fans
Extracting Fluid Volumes
Setting Boundary and Interface Conditions
Setting Up the Mesh
Preparing the Post-Processing
Running the Simulation
Analyzing the Results

Defining the Setup

Follow these steps to define the physics, units, and solver settings for an electronics cooling simulation.

1. Click (**Launch Electronics Cooling App**).

The Electronics Cooling Toolset appears in a separate tab within STAR-CCM+. An empty *Electronics Cooling* scene opens for visualization.

To define the physics of your electronics cooling simulation:

2. Right-click the **Setup** > **Physics** node and select **Edit**.
3. In the *Edit* dialog, set:
   - Ambient Conditions
   - Solution Physics

   See Setup Reference—Physics.
4. In the *Materials* group box, define the various solid, gas, and liquid materials that are present within your electronics device, see Defining Materials.
5. Click **Close**.

To define the preferred units for your electronics cooling simulation:

6. Right-click the **Setup** > **Units** node and select **Edit**.

   The *Edit* dialog displays the default engineering units for the various fundamental and derived dimensions.
7. Set the units to suit your simulation. See Setup Reference—Units.
8. Click **Close**.

To define the solver settings for your electronics cooling simulation:

9. Right-click the **Setup** > **Solver** node and select **Edit**.
10. In the *Edit* dialog, set the following **Solver Settings**:
    - Stopping Criteria
    - Segregated Flow Under-Relaxation Factors
    - Segregated Energy Under-Relaxation Factors

    See Setup Reference—Solver.
11. Click **Close**.

The physics, units, and solver settings are now defined. In the background, the selected conditions and parameters are set in the STAR-CCM+ simulation tree.

Contents:

Defining Materials

Defining Materials

Follow these steps to set up the various materials that are present within your electronics device.

To define the solid materials:

1. Right-click the **Setup** > **Physics** node and select **Edit**.
2. In the *Materials* group box, next to the **Solid Sim Materials** box, click **Select...**.
3. In the *Select Materials* dialog, click **Add Material**.

4. From the database of standard solid materials, select the materials to suit your components and click **OK**.

To create an identical copy of a solid material:

5. In the list of **Solid Simulation Materials**, select the material of interest and click **Duplicate**.

To edit the properties of a solid material:

6. In the list of **Solid Simulation Materials**, select the relevant material.
7. Set the **Material Name** to suit your requirements.
8. Specify the material properties using one of the following methods:
   - For a constant material property, set **Method** to **Constant** and enter the respective value in the **Value** box.
   - For a material property that is a function of temperature, set **Method** to **Polynomial in T** and click the respective **Value** box. In the *Polynomial Input* dialog, set the Polynomial in T Properties.
   - For an anisotropic material property, set **Method** to **Anisotropic**. You specify the components of the anisotropy tensor when setting up the relevant QuickParts, see PCB Reference and Generic Reference.

   Note: Not all methods are available for each material property. For example, the Anisotropic method is only available for the Thermal Conductivity property.

9. Click **Apply** and **Close**.

To reduce the time when setting up the solid components, select the material that characterizes most components:

10. Next to the **Default Solid** box, click **Select...**.
11. From the list of Solid Sim Materials, select the relevant material and click **OK**.

    Note: The **Default Solid** property does not support Solid Sim Materials with **Thermal Conductivity** > **Method** set to **Anisotropic**.

To define the gas material:

12. In the *Material* group box, next to the **Default Gas** box, click **Select...**.
13. In the *Select Material* dialog, next to the **Selected Gas Material** box, click **Select...**.
14. From the database of standard gas materials, select a suitable gas, such as Air, and click **OK**.
15. Set the **Material Name** and material properties to suit your requirements.
16. Click **Apply** and **Close**.

A similar procedure is followed to define the **Default Liquid** material.

Defining Parameters

Follow these steps to define global parameters. A global parameter is a scalar physical quantity that can be defined once and used in multiple Construction Geometries, QuickParts, or QuickPart Surfaces.

One application of a global parameter is in defining a key physical characteristic as a reference value. Other values can point directly to this reference value or be defined in terms of it. For example, in your electronics

device, the number of fins for a heatsink could be set once as a global parameter, and then used in numerous other heatsinks.

Similarly, you can define a global constant like the mathematical value pi (3.14...). By defining a global parameter, you can use it repeatedly without entering the value again.

Another important use is in design exploration—global parameters can act as controls that you adjust over multiple design simulations.

Note: Only SI units are available for global parameters.

To set up a global parameter:

1. Right-click the **Parameters** node and select **Edit**.
2. In the *Edit* dialog, click **Add a new parameter**.
3. In the group-box, set the following properties:
   - Parameter
   - Dimensions
   - Value

   For more information, see Parameters Reference.
4. Click **Add**.

   The parameter is now available for other nodes in your electronics cooling simulation.
5. Repeat step 2 to 4 to create additional parameters.

To edit a parameter:

6. Select the text-box of the parameter property whose value you want to change and set the value to suit your requirements.

To delete a parameter:

7. Click ▬ next to the parameter entry.
8. Click **Close**.

To apply a global parameter, follow a procedure similar to the example outlined below:

9. Right-click a node that stores a scalar physical quantity, such as **QuickParts** > **Heatsink**, and select **Edit** (see Creating and Positioning the Components).
10. In the *Edit* dialog, set the value of the respective scalar quantity to `$[Parameter]`, where [Parameter] is the name of the global parameter.
11. Click **Apply** and **Close**.

Creating and Positioning the Components

Follow these steps to define the geometry, placement, and physics of the components in your electronics device.

When setting up the various components of your device—such as PCBs, chips, heatsinks, or resistances—consider the following recommendations:

- Simplify or eliminate components away from the primary heat flow path.

- Eliminate any parts such as fasteners that are not critical to the heat flow path.
- Fill any holes that are created by eliminating irrelevant parts, and any irrelevant gaps.
- Simplify components such as connectors, threaded stand-offs, and transformers.
- For sheet metal parts, remove the rounds resulting from bends, and fill or remove small features that are created by bend reliefs and the gaps resulting from rip corners.
- Eliminate interferences.

Start with creating PCBs and associated electronic components:

1. If you have IDF (Intermediate Data Format) files that contain the layout definitions of the board and its components, import the files:
   a) Right-click the **QuickParts** node and select **Import IDF file**.
   b) In the *Open* dialog, navigate to the folder that contains the IDF board file.

      The IDF board file includes information about the outline of the board, its thickness, and the positions of the components on the board.
   c) Select the file and click **Open**.

      The IDF library file must be in the same directory to load the complete circuit board. If no library file is found, you are asked to locate it manually. The IDF library file includes information about the outline of the components and their thickness.
   d) In the *Import IDF Options* dialog, set the desired import options (see Import IDF Options Panel) and click **OK**.

      A composite QuickPart—named after the imported file—appears in the tree. The composite contains PCB and Generic QuickParts representing the board and its components, respectively.
2. To create PCBs and its components directly:
   a) Right-click the QuickParts node and select **New QuickPart** > **PCB** > **[Geometry]**, where [Geometry] is one of the built-in shape templates for the PCB.
      Example: **New QuickPart** > **PCB** > **Rectangular**.

      The initial position of the local origin of the new QuickPart corresponds to the origin of the Laboratory Coordinate System.
   b) To create a component, such as a chip, on a PCB, in the *Graphics* window, select the face of the PCB on which you want to place the chip.
   c) Right-click and select **New QuickPart** > **Chip** > **[Geometry]**.

      The position of the mouse pointer when selecting the face determines the initial position of the local origin of the new chip.

Add additional components such as chips, heatsinks, fans, resistances, or other generic elements:

3. Create QuickParts of the relevant types from within the tree or interactively as described in step 2. Creating QuickParts interactively in the *Graphics* window usually reduces the time for placing new QuickParts on previously created components.

When you create a new QuickPart, an *Edit* dialog automatically appears that displays the QuickPart properties.

4. To define the QuickPart properties:
   a) Enter the **Name** of the QuickPart under which it is stored in the tree.
   b) In the *Geometry* section, set the shape-specific properties.

Example: For a rectangular PCB, you set **X**, **Y**, and **Z** dimensions and the **Local Origin** of the geometry.

Alternatively, dimension the geometry interactively and in reference to other objects. See Dimensioning Objects Interactively.

c) In the *Placement* section, set the **Location**, **Rotation Axis Vector**, **Angle**, and **Coordinate System** properties.

Alternatively, place the geometry interactively and in reference to other objects. See Placing Objects Interactively.

d) To preview your geometry and placement settings, click **Update** at the bottom of the dialog.

e) In the *Physics* section, set the QuickPart specific physical properties—like **Material** or **Heat Source**—to suit the represented component.

f) For a Chip QuickPart, you can mimic the performance of the respective chip package by using a Compact Thermal Model (CTM):

1. In the *Physics* section, set **Thermal Specification** to a resistor network that suits the heat flow paths within the chip package.

   Example: For a leaded package, the package temperature is affected not only by the heat transfer through the top and bottom surfaces but also by the heat flow path through the sides. To consider heat transfer through all sides of the package, set **Thermal Specification** to **Star Network**. See Chip Reference—Resistor Network Types.

2. Set the resistor network properties as required.

g) Click **OK**.

For more information on QuickPart specific properties, see QuickParts Reference.

5. To change QuickPart properties after creation:

a) Right-click the QuickPart node and select **Edit**.

The *Edit* dialog re-launches.

If the built-in geometry templates for QuickParts are not suitable for the components of your device, you can create Custom QuickParts whose geometries are based on Construction Geometries that you specify. Custom QuickParts are particularly useful for defining advanced components, such as cooling loops.

6. To create a Custom QuickPart:

a) Create a Construction Geometry that defines the shape and position of the component. See Creating Construction Geometries.

b) Create a new QuickPart from within the tree or interactively as described in step 2. For [Geometry] select **Custom**.
Example: For the tubes of a cooling loop, select **Generic** > **Custom**.

c) In the *Select Construction Geometry* dialog, select the created Construction Geometry, set the QuickPart options to suit your requirements, and click **OK**. See Select Construction Geometry Dialog.

d) To define the physics properties, right-click the QuickPart node and select **Edit**.

e) In the *Edit* dialog, set the properties as required and click **Apply** and **Close**.

Usually, complex devices are composed of different electronic modules. Organize the components of your device using composites:

7. To create a composite:

a) Using <Ctrl>, multi-select the relevant QuickParts.

b) Right-click and select **Composite**.

A new composite part (assembly) node appears.

8. To move a QuickPart to a previously created assembly:

a) Select the QuickPart and drag and drop it onto the composite part node.

Contents:

Dimensioning and Placing Objects Interactively
Creating Construction Geometries

Dimensioning and Placing Objects Interactively

Follow theses steps to interactively set length dimensions or define placement properties of a Construction Geometry or QuickPart.

Note: In this context, unless otherwise stated, the term "object" refers to a Construction Geometry or QuickPart.

Dimensioning and positioning objects interactively in the *Graphics* window reduces setup time and allows you to refer to dimensions and positions of other objects.

The following interactive dimensioning feature is available when editing an object:

- Setting a length dimension

The following interactive positioning features are available:

- Changing the local origin of an object
- Creating a reference coordinate system in the center of a selected face
- Defining the position and orientation of an object using the Positioning Tool
- Placing an object with respect to another object in the *Graphics* window
- Centering an object on a face of another object

For detailed information about the various interactive features, see Interactive Right-Click Actions and Interactive Right-Click Actions in Editing Mode.

Dimensioning

To interactively set a length dimension of an object:

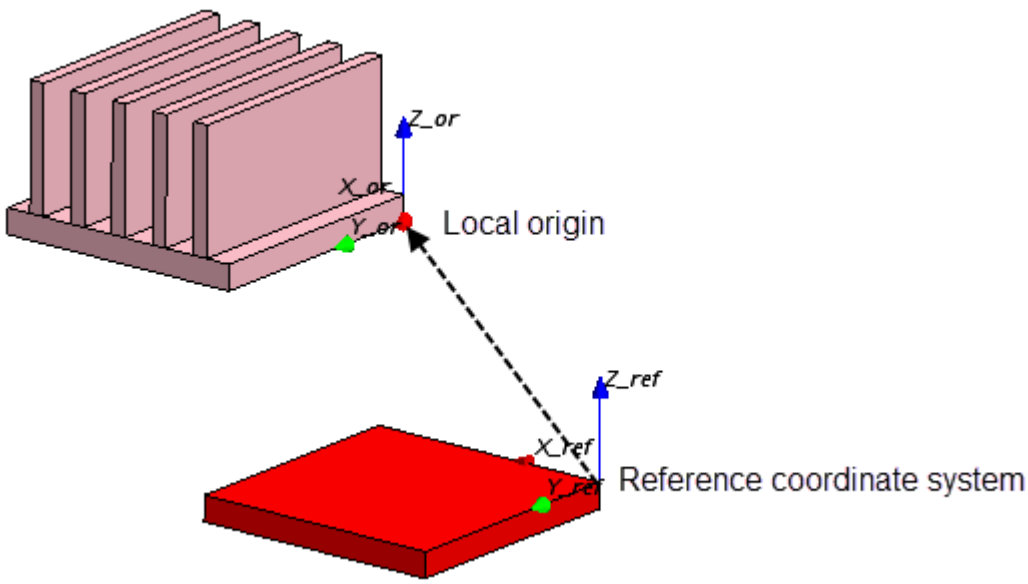
1. Right-click the object node and select **Edit**.
2. In the *Graphics* window, select either one or two entities as follows:
   - Select a single `Linear Edge` when the length of that edge is the length that you want to apply to an edge of the object chosen in the tree.
   - Select two entities when you want to apply the distance between them to a dimension of the chosen object. You can choose any two entities of type `Vertex`, `Linear Edge`, or `Planar Face`.
3. Right-click and select **Set Dimension** > **[Length Dimension]**, where [Length Dimension] is a length dimension of the object being edited.

The object geometry updates automatically. In the *Edit* dialog, the respective geometry property displays the value that is calculated from the selected entities.

Placing

The placement of an object is determined by the location and orientation of the object's local coordinate system with respect to a reference coordinate system. See X_or, Y_or, Z_or and X_ref, Y_ref, Z_ref in the screenshot below.

The local coordinate system of an object is placed at the object's local origin. In some situations it helps to move the object's local origin to a specific geometric entity to simplify the placement of the object. Creating a new reference coordinate system can further simplify placing the object.



To change the location of the local origin:

4. In the *Graphics* window, select a suitable geometric entity of the object.

   The supported geometric entities are `Vertex`, `Linear Edge`, `Curved Edge`, and `Planar Face`.

5. Right-click and select one of the following specific options:
   - **Set Origin Reference At Vertex**
   - **Set Origin Reference To Center Of Linear Edge**
   - **Set Origin Reference To Center Of Curve**
   - **Set Origin Reference To Center Of Face**

   The object's local coordinate system moves to the desired position.

To create a reference coordinate system in the center of a specific face:

6. In the *Graphics* window, select the respective face.
7. Right-click and select **Create Face Coordinate System**.

   A new reference coordinate system is created, where the origin of the coordinate system is centered on the selected face. The coordinate system is now available as reference coordinate system for any object within your simulation.

To define the position and orientation of an object using the Positioning Tool:

8. Launch the **Positioning Tool** using one of the following options:
   - When creating or editing the object, in the *Edit* dialog, click the **Position** button.
   - Right-click the object node and select **Position**.
   - In the *Graphics* window, select a face of the object, right-click, and select **Position...**.
9. In the *Edit* dialog, choose one of the following positioning options:
   - Align
   - Rotate
   - Translate
10. Set the properties of the selected positioning mode as required. See Positioning Tool Reference.

    You can either set the properties manually or use the blue boxes to set the properties interactively. To set a property interactively:

    a) Click the blue box. The color of the box becomes dark blue.

    b) In the *Graphics* window, select a suitable geometric entity.

    c) To remove an entity from the box, right-click the box and select **Remove**. Alternatively, click anywhere in the empty space of the *Graphics* window.
11. Click **Apply**.
12. To leave the Positioning Tool, click **Return**.

To place an object with respect to another object in the *Graphics* window:

13. Right-click the geometry object node and select **Edit**.
14. In the *Graphics* window, using <Ctrl>, multi-select a suitable geometric entity of the object being edited and another entity of the reference object. The supported geometric entities are `Vertex`, `Linear Edge`, and `Planar Face`.
15. Right-click and select **Position** > **[Option]**, where [Option] is one of the following placement modes:
    - Set Distance
    - Snap
    - Set Angle
    - Align
    - Align and Snap
    - AntiAlign
    - AntiAlign and Snap

    The position and orientation of the object is updated accordingly.

To center an object on the planar face of another object:

16. Right-click the node of the object being centered and select **Edit**.
17. In the *Graphics* window, select the desired face of the other object.
18. Right-click and select **Move Object Being Edited to Center of Face**.

    The object moves to a position where the object's local origin corresponds to the center of the face.

Creating Construction Geometries

Follow these steps to build arbitrary solid objects or to build sheet objects for splitting QuickPart Surfaces.

Within the Electronics Cooling Toolset you can create Construction Geometries to describe sheet or solid objects. To create Construction Geometries, the following options are available:

- Building geometry based on simple shape templates
- Importing geometry from another CAD package
- Creating geometry within the 3D-CAD modeler and transferring it to the Electronics Cooling Toolset

To operate on created or imported geometrical shapes, the Electronics Cooling Toolset supports the following options:

- Applying boolean operations
- Generating patterns of objects

To build a geometry based on simple shape templates:

1. Right-click the **Construction Geometry** node and select **New Construction Geometry** > **[Type]**, where [Type] is one of the following:
   - Block
   - Sphere
   - Cylinder
   - Sheet Body Rectangle
   - Sheet Body Circle

   Alternatively, you can create a Construction Geometry on the surface of another geometry object that is visible in the *Graphics* window. Usually, creating a Construction Geometry interactively reduces the time for placing the Construction Geometry on a previously created geometry object:

   1. In the *Graphics* window, select a suitable face of the object.
   2. Right-click and select **New Construction Geometry** > **[Type]**.

   The position of the mouse pointer when selecting the face determines the position of the initial local origin of the new Construction Geometry.

An *Edit* dialog appears that displays the name, geometry, and placement properties of the Construction Geometry.

To define the properties of the Construction Geometry:

2. For **Name**, enter a description of the Construction Geometry under which it is stored in the tree.
3. In the *Geometry* and *Placement* sections, define the shape and position of the geometry as required. See Construction Geometries Shape Templates. Alternatively, dimension and place the Construction Geometry interactively and in reference to other geometrical objects. See Dimensioning and Placing Objects Interactively.
4. If you want to preview your geometry and placement settings, click **Update** at the bottom of the dialog.
5. Click **OK**.

To import a geometry from another CAD package:

6. Right-click the **Construction Geometry** node and select **Import CAD model**

7. In the *Open* dialog, navigate to the folder that contains the CAD model.
8. Select the CAD file and click **Open**.

   For more information, see Imported Construction Geometries.

To create a geometry within 3D-CAD and transfer it to the Electronics Cooling Toolset:

9. Click the *Simulation* tab.

   The STAR-CCM+ tree appears.
10. Right-click the **Geometry** > **3D-CAD Models** > **Electronics Cooling 3D-CAD 1** node and select **Edit...**.
11. Build the respective geometry in 3D-CAD, see Using 3D-CAD.
12. When you are done with creating the geometry, click **Close 3D-CAD**.
13. Click the *Electronics Cooling* tab.
14. Right-click the **Construction Geometry** node and select **Transfer CAD Bodies.**

    For more information, see Transferred Construction Geometries.

To edit the properties of a Construction Geometry:

15. Right-click the respective Construction Geometry node and select **Edit**.

    The *Edit* dialog re-launches that allows you to change the name, geometry, and placement properties as required.

To apply a boolean operation on a previously created geometry:

16. Right-click the node of the respective Construction Geometry and select **Boolean Construction Geometry**.
17. In the *Edit* dialog, set the geometry and placement properties as required. See Construction Geometries Operations—Boolean Construction Geometry.
18. Click **OK**.

To generate a pattern of identical geometries:

19. Right-click the node of the Construction Object that you want to multiply and select **Pattern Construction Geometry**.
20. In the *Edit* dialog, set the properties to suit the required pattern. See Construction Geometries Operations—Pattern Construction Geometry
21. Click **OK**.

Adding an Enclosure and Fans

Follow these steps to create an enclosure for your electronics device and to fit fans in the enclosure walls.

If you want to perform an internal flow analysis and consider heat transfer effects through the housing of your device, create an Enclosure QuickPart:

1. From within the tree or interactively, create a new Enclosure QuickPart using **New QuickPart** > **Enclosure** > **[Geometry]**.
2. In the *Edit* dialog, set the enclosure properties to suit your requirements. See Enclosure Reference.
3. Click **OK**.

To cut out in- and outflow openings, you specify the profiles of the openings using Sheet Construction Geometries:

4. In the Graphics window, select the relevant enclosure wall.
5. Right-click and select **New Construction Geometry** > **[Type]**, where [Type] is one of the following:
   - Sheet Body Circle
   - Sheet Body Rectangle

   For more information, see Creating Construction Geometries.
6. Right-click the respective enclosure node and select **Edit Interactions**.
7. Within the *Edit* dialog, in the *Extrude Cut Through First Segment* group box, set **Sheet Construction Geometry** to the previously created sheet geometries.

   For more information, see Interactions Reference—Extrude Cut Through First Segment.
8. Click **Apply** and **Close**.

If you want to remove a wall from a rectangular enclosure:

9. In the *Graphics* window, select the respective enclosure wall.
10. Right-click and select **Create Open Enclosure Face**.

    The wall disappears from the enclosure geometry.
11. To restore previously removed walls of an enclosure:
    a) Select any face of the enclosure.
    b) Right-click and select **Restore All Open Enclosure Faces.**

To fit an axial fan in an enclosure wall:

12. In the *Graphics* window, select the wall of interest.
13. On the enclosure wall, create an **Axial Fan** QuickPart. See steps 6–12.
14. Right-click and select **New QuickPart** > **Axial Fan** > **[Geometry]** , where [Geometry] can be **Templated** or **Custom**.
15. In the *Edit* dialog, set the fan properties to suit your requirements. See Axial Fan Reference.

    Alternatively, dimension and place the axial fan interactively and in reference to other objects, such as the enclosure.
16. Click **OK**.
17. Right-click the enclosure node and select **Edit Interactions**.
18. Within the *Edit* dialog, in the *Subtract* group box, set **Solid Quick Objects** to the Axial Fan QuickPart.

    For more information, see Interactions Reference—Subtract.
19. Click **Apply** and **Close**.

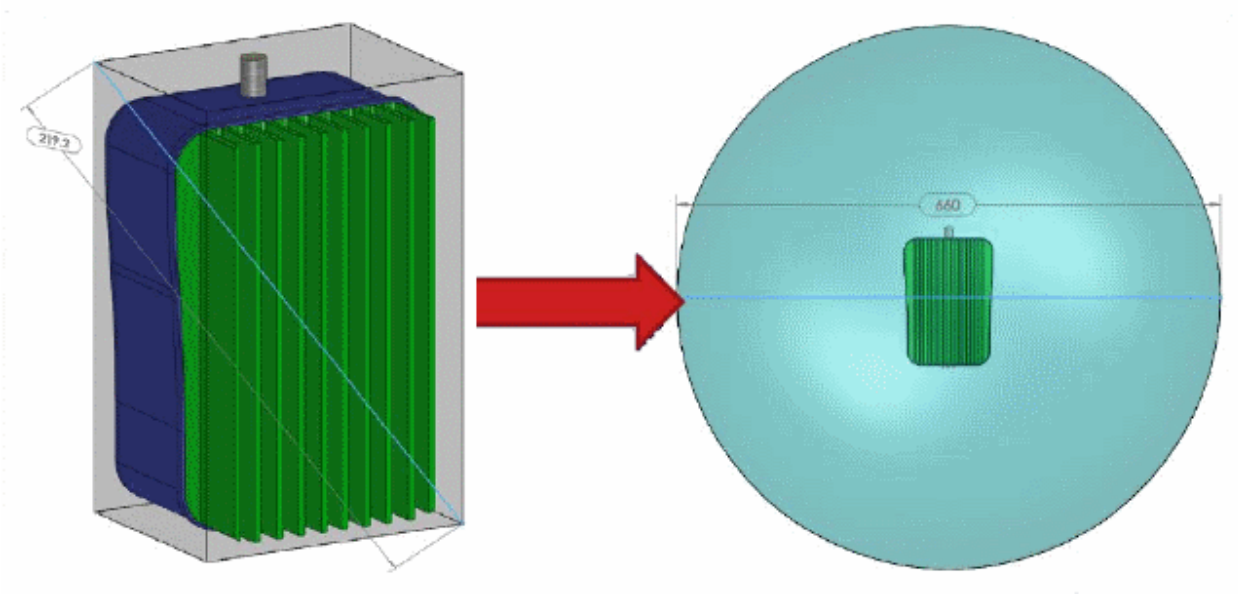
    An opening is cut in the enclosure wall that corresponds to the fan geometry.

Extracting Fluid Volumes

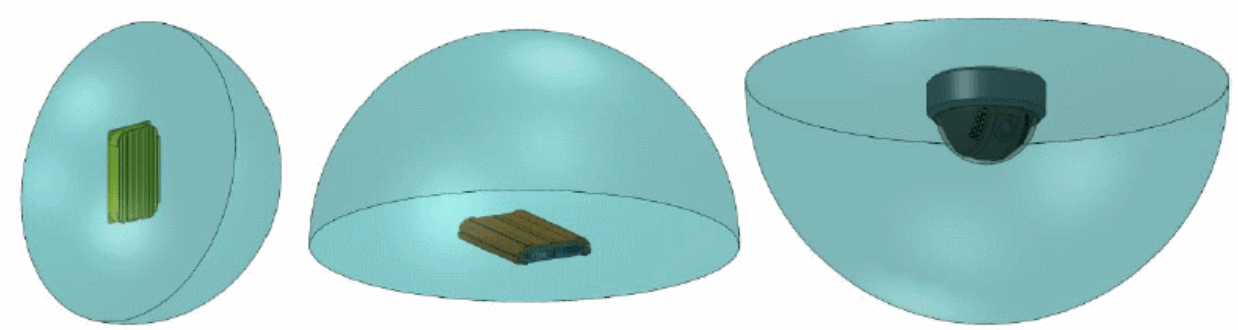
Follow these steps to define air and coolant domains for an electronics cooling simulation.

STAR-CCM+ allows you to model internal flow, external flow, or both.

- For internal air or coolant flow, you extract the void between the components that enclose the relevant domain.
- For external flow, you must first create a shape to let the fluid move around the electronics device. Typically, you use a sphere or hemisphere whose outer surface represents the far-field conditions of the flow. Make the diameter of the sphere or hemisphere three to five times the length of the longest diagonal of the model.



If the device is mounted on a wall, table, or ceiling, you define a hemisphere.



For an external fluid domain, you create a suitable Construction Geometry that describes the volume of interest. For an internal fluid domain, you can use a combination of QuickParts and a Construction Geometry to define the volume of interest.

1. If necessary, create the appropriate Construction Geometry using one of the following methods:

Geometry of External Air Domain	Procedure
Sphere	Create a Sphere Construction Geometry. See Creating Construction Geometries.

Hemisphere	Create a sphere and a block geometry—where one block plane cuts through the center of the sphere—and subtract the block from the sphere: 1. Create a Sphere Construction Geometry and a Block Construction Geometry. See Creating Construction Geometries. 2. Right-click the **Construction Geometry** > **Sphere** node and select **Boolean Construction Object**. 3. Within the in-place dialog, in the *Geometry* section, set **To Subtract** to the created Block Construction Geometry. 4. Click **OK**.

2. Right-click the **QuickParts** node and select **Create Extract Volume**.

   The *Extract Volume* in-place dialog appears.
3. Within the **Geometry** section, in the *Extract Volume Domain Defined By* group box, activate the appropriate option:
   - **Construction Geometry**

     Use the selected construction geometry to create a fluid analysis domain. You use this option to set up and extract an external fluid domain.
   - **QuickParts + Const Geometry**

     Use the selected QuickParts and (optionally) the selected construction geometry to create a fluid analysis domain. You use this option to extract an internal volume of an electronics device.

Select the objects that enclose the volume of interest. The union of all the components of the selected construction geometry and the selected QuickParts defines the boundary of the volume that is extracted.

4. Click the **Construction Geometry** or **QuickParts + Const Geometry** box (whichever is appropriate) and do the following:
   - For an internal air domain, select the QuickParts that enclose the air, such as an Enclosure QuickPart.
   - For a coolant domain, select the Generic QuickParts and the construction geometry that represent the tubes of the cooling loop.
5. (Applies to **QuickParts + Const Geometry** only) For open enclosures or cooling loops that contain entry and exit holes, set the **Inlet/Outlet Edges or Faces**:
   a) In the *Graphics* window, select the edges or faces that describe the locations of the holes. See Extract Volume Reference.

      The selected entities are marked in green.

   b) To remove an entity from the list of **Inlet/Outlet Edges or Faces**, right-click the respective entity and select **Remove**. Alternatively, click the respective entity in the *Graphics* window while pressing the `<Ctrl>` key. To delete all selected entities from the list, right-click anywhere in the **Inlet/Outlet Edges or Faces** box and select **Remove All** or click anywhere in the empty space of the *Graphics* window.
6. Within the **Physics** section, set the material of the extracted volume:
   - For an internal air domain, select **Gas**.
   - For a coolant domain, select **Liquid**.
7. Click **OK** and **Return**.

An **Extract Volume** QuickPart appears in the tree.

Setting Boundary and Interface Conditions

Follow these steps to define the physical conditions on the surface of a QuickPart.

Most QuickParts have only one QuickPart Surface by default. If you want to set different boundary or interface conditions on specific faces of a QuickPart—like inflow, outflow, and wall conditions on the surface of an extracted air volume—you must split the default surface first. See Splitting a QuickPart Surface.

For a problem to be well-posed, boundary conditions must be specified on all solution domain boundaries. Interface conditions describe the physical effects between contacting QuickParts.

To set boundary and interface conditions on a specific QuickPart Surface:

1. Right-click the respective QuickPart Surface node and select **Edit**.
2. In the *Edit* dialog, set the boundary and interface properties to suit your simulation.

   See QuickPart Surfaces Reference.
3. To visualize the faces that correspond to the boundary or interfaces of the QuickPart Surface, click **Highlight** in the respective group box.
4. Click **Apply** and **Close**.

Contents:

Splitting a QuickPart Surface

Splitting a QuickPart Surface

Follow these steps to split a QuickPart Surface.

Most QuickParts have only one QuickPart Surface, by default. You can split a QuickPart Surface in one of the following ways:

- Creating a new QuickPart Surface using the underlying face information.

  Example: A block-shaped chip has only one QuickPart Surface, by default, but it has six faces—one for each side of the block (this information is stored as part of the geometry). Each face can be used to create a new QuickPart Surface.
- Creating a new QuickPart Surface from a modified face information.

  Example: To create a circular QuickPart Surface on a block, you must first modify the underlying face information by imprinting a circular profile on the relevant side surface. This action creates a circular face on the respective side of the block. The circular face can then be used to create a new circular QuickPart Surface.

To create a new QuickPart Surface from a specific face:

1. In the *Graphics* window, select the face.

   To select multiple faces, hold down the `<Ctrl>` key.
2. Right-click and select **Create QuickPart Surface**.
3. In the *Rename* dialog, enter the **Name** of the new QuickPart Surface and click **OK**.

The new QuickPart Surface appears under the QuickPart node.

To modify the face information of a QuickPart:

4. In the *Graphics* window, select the face of interest.
5. Right-click and select **New Construction Geometry** > **[Type]**, where [Type] is one of the following:
   - Sheet Body Circle
   - Sheet Body Rectangle

   For more information, see Creating Construction Geometries.
6. Right-click the respective QuickPart node and select **Edit Interactions**.
7. Within the *Edit* dialog, in the *Imprint* box, set **Sheet Construction Geometry** to the previously created sheet geometry.

   For more information, see Interactions Reference.
8. Click **Apply** and **Close**.

   The profile is imprinted and the face information is modified accordingly. You can now use the resulting faces to create a new QuickPart Surface.

Setting Up the Mesh

Follow these steps to define the mesh of an electronics cooling simulation.

Before you set up the mesh, check your model for intersecting QuickParts:

1. Right-click the **QuickParts** node and select **Check Interference**.
2. If the *Output* window lists any intersecting QuickParts, correct the QuickParts:
   a) Right-click the respective QuickPart node and select **Edit**.
   b) In the *Edit* dialog, modify the geometry and placement properties as required.
   c) Click **Apply** and **Close**.

To set up the mesh:

3. Right-click the **Mesh** node and select **Edit Mesh Settings**.

   The warning triangle on the **Mesh** icon indicates that you must still generate the mesh (after defining the settings and any custom controls).
4. In the *Edit* tab, define the global mesh and boundary layer settings to suit your electronics cooling simulation. See Mesh Reference. If you are uncertain of any settings, maintain the default settings. They are suitable in many cases.
5. To commit changes to the default mesh settings, click **Apply**.
6. Click **Close**.
7. If you want to customize the mesh on specific faces of a QuickPart Surface, separate those faces into their own QuickPart Surface. See Splitting a QuickPart Surface.
8. To set customized mesh settings on a QuickPart Surface:
   a) Right-click the **Mesh** > **Custom Controls** node and select **New** > **Surface Control**.
   b) In the *Edit* dialog, set **Surfaces** to the respective QuickPart Surface and change the mesh settings as required. See Mesh Reference—Surface Control.

   c) Click **Apply** and **Close**.

9. If you want to customize the mesh within a specific volumetric zone, define a geometry that encloses the zone of interest and set up a Volume Control:
   a) Create an arbitrary three-dimensional volume shape. See Creating Construction Geometries.
   b) Right-click the **Mesh** > **Custom Controls** node and select **New** > **Volume Control**.
   c) In the *Edit* dialog, set **Parts** to the created Construction Geometry and change the mesh settings as required. See Mesh Reference—Volume Control.
   d) Click **Apply** and **Close**.

To generate the mesh:

10. In the *Mesh Generation* toolbar (see Toolbar Reference), click (**Generate Volume Mesh**)
    The progress of the mesh generation is displayed in the *Output* window. Once the progress is complete, the warning triangle on the **Mesh** icon disappears.

Note: When you change mesh settings after the meshing progress is complete, the warning triangle indicates that the mesh is no longer up to date and requires re-execution.

Preparing the Post-Processing

The Electronics Cooling Toolset allows you to analyze the solution while the simulation is running, as well as when it completes.

Field functions provide access to raw simulation data, and to data computed from the raw data. See Field Functions Reference. These field functions can be analyzed using three general strategies:

- Visualizing the data in geometry, scalar, or vector scenes. Scenes are the basic building blocks that build the visualization display in the *Graphics* window. Solution data can be displayed on QuickPart Surfaces or on drawables, which are: plane sections, isosurfaces, or streamlines. See Scenes Reference.
- Plotting sets of data that are extracted from the solution in XY plots. XY plots are two-dimensional graphs that display scalar values from the set of available solution variables. For an XY plot, you choose derived data items—point probes or line probes—from which the solution data is extracted. See Results Reference.
- Creating reports of specific quantities. A report displays various statistical functions of solution data—maximum, minimum, average, and total values. The report item can be a QuickPart, a QuickPart Surface, or a derived data item. See Results Reference.

Note: If you create scenes or plots before you run the simulation, they are updated continuously as the solution progresses.

Before creating scenes, plots, or reports for post-processing, be clear on what quantities you want to analyze, and define the objects from which you want to extract the quantities.

To visualize the solution in a scene:

1. If existing QuickPart Surfaces are not sufficient for you to use in visualizing data, create additional drawable items:

a) Right-click the **Scenes** > **Drawable** node and select **New** > **[Drawable]**, where [Drawable] is one of the following drawable items:
   - **Section**
   - **IsoSurface**
   - **Streamlines**

b) In the *Edit* dialog, set the properties of the respective drawable item as required. See Drawables Reference.

c) Click **OK**.

2. If you want to add a text label to a scene:
   a) Right-click the **Scenes** > **Scene Annotations** node and select **New**.
   b) In the *Edit* dialog, set **Name** and **Annotation Text**. See Scene Annotations Reference.
   c) Click **OK**.
3. Create a scene:
   a) Right-click the **Scenes** node and select **New**.
   b) In the *Edit* dialog, click the **Drawable Items** box and select the QuickPart Surfaces, Drawables, and Annotations that you want to display. Set the other scene properties to suit your analysis, see Scenes Reference.
   c) Click **OK**.
4. If you want to render a scene visible or invisible in the visualization display, right-click the respective scene node and select **Toggle Visibility**.

To plot sets of data in an XY plot:

5. Define the points from which you want to extract solution data:
   a) Right-click the **Results** > **Derived Data** node:
      1. To extract data from evenly spaced points along a line, select **New** > **Quick Line Probe**.
      2. To extract data from a single point location, select **New** > **Quick Point Probe**.
   b) In the *Edit* dialog, set the properties of the respective derived data item (see Results Reference—Derived Data). Alternatively, you can position the probe interactively in the *Graphics* window (see Using the Line Tool or Using the Point Tool).
   c) Click **OK**.
6. Create a plot:
   a) Right-click the **Results** > **Plots** node and select **New**.
   b) In the in-place dialog, set **Derived Data Items** to the respective point or line probes and set the properties of the plot as required. See Results Reference—Plots.
7. Right-click the plot node and select **Open.**

   The plot is displayed in the *Graphics* window.

To report specific quantities:

8. If the quantity of interest is sourced from discrete points, create Quick Point Probes or Quick Line Probes as described in step 5.
9. Create a report:
   a) Right-click the **Results** > **Reports** node and select **Create Post Report**.

b) In the *Edit* dialog, set the **Name**, **Report Field**, and **Report Items** as required (see Results Reference—Reports).

c) Click **OK**.

Running the Simulation

Follow these steps to run the simulation.

Before running the simulation, check your simulation setup:

1. Right-click the **Electronics** node at the root of the object tree and select **Summarize...**.

   The *Output* window displays the following information:

   - Total Power
   - Gravity Direction
   - Number of QuickParts
   - Number of Mesh Cells
   - Number of Iterations
   - QuickPart Type Count

   See Electronics Cooling Toolset Reference.

To run the simulation:

2. Click (**Run**) in the *Solution* toolbar. See Toolbar Reference.

   A *Residuals* plot is created automatically and shows the progress of the solver. While the simulation is running, you can click the *Electronics Cooling 1* tab at top of the Graphics window to visualize the solution.

After the run is finished, save the simulation:

3. In the *System* toolbar, click (**Save**).

Analyzing the Results

Follow these steps to view, plot and report the results of the converged simulation.

After the run is finished, analyze the simulation results:

- Examine the scene displays
- Investigate the plots
- Analyze the values of the post reports

To examine the scene displays:

1. Click the *Electronics Cooling 1* tab at top of the *Graphics* window.
2. To render a scene visible or invisible in the visualization display, right-click the scene node and select **Toggle Visibility**.
3. If you want to create additional scenes, follow the steps to visualize the solution in a scene.

4. To save the scenes that are visible in the visualization display to a scene image file:
   a) Right-click the background of the visualization display and select **Hardcopy...**.
   b) Use the *Save To File* dialog to save the visualization display. For details, see Saving Scene Image Files.
   c) Click **Save**.
5. If a scene contains drawable items, that are sections, isosurfaces, or streamlines, you can animate theses items. To animate a drawable item:
   a) Right-click the drawable item node and select **Edit**.
   b) Within the *Edit* dialog, in the *Animation Settings* box, activate **Turn On Animation**.
   c) Click **OK**.
   d) To control the animation, use the *Animation* toolbar buttons (see Toolbar Reference). See Controlling the Movement of the Animation.
   e) To record a segment of the animation onto separate files—either a commonly used type of video file, `.avi`, or a set of frames in `.png` graphics format—see Recording the Animation onto Separate Files.

Investigate the plots:

6. To open a plot, right-click the plot node and select **Open**.

   If you closed the Residuals plot after the run and want to restore it again, right-click the **Plots** node and select **Open Residuals Plot**.
7. If you want to create additional plots, follow the steps to plot sets of data in an XY plot..
8. To save a plot to a plot image file:
   a) Open the plot.
   b) Right-click the plot display and select **Hardcopy...**.
   c) Use the *Save To File* dialog to save the plot display. For details, see Saving a Plot Image File.
   d) Click **Save**.
9. To tabulate the data sets of a plot, right-click the plot display and select **Tabulate...**.

   The data sets appear in a spreadsheet-like dialog. The header includes the names of the derived data items and the plotted scalars.
10. To export the data sets of a plot to a `.csv` file:
    a) Tabulate the plot as described in step 9. In the *Tabular Data* dialog, click **Export**. Alternatively, you can directly export the plot by right-clicking the plot display and selecting **Export...**.
    b) In the *Save* dialog, enter the file name and click **Save**.

Analyze the values of the post reports:

11. To extract the data of a post report, right-click the report node and select **Tabulate**.

    You can extract data of multiple reports by using <CTRL> to multi-select the nodes.

    For information on the displayed data, see Results Reference-Tabulate.
12. To tabulate all reports, right-click the **Reports** node and select **Tabulate All**.
13. To export the report values to a `.csv` file:
    a) Tabulate the report values as described in step 11 or 12 and, in the *Tabular Data* dialog, click **Export**.

Alternatively, you can directly export the report values by right-clicking the report node and selecting **Export...**.

You can export data of multiple reports by using <CTRL> to multi-select the nodes.

b) In the *Save* dialog, enter the file name and click **Save**.

14. To export all report data to a `.csv` file, right-click the **Reports** node and select **Export All**.

Electronics Cooling Toolset Reference

The following sections provide definitions and settings for the various objects within the Electronics Cooling Toolset.

Electronics Manager Right-Click Actions

The following actions are available when you right-click the **Electronics** node at the root of the object tree:

Summarize...

Displays the following summary of simulation data in the Output window:

- **Total Power**: sum of all specified heat sources.
- **Gravity Direction**: display of gravity vector.
- **Number of QuickParts**: total number of all QuickPart objects.
- **Number of Mesh Cells**: number of cells for the generated volume mesh.
- **Number of Iterations**: current iteration number.
- **QuickPart Type Count**: lists the number of QuickParts for each available QuickPart type.

Collapse Tree

Shrinks the tree back to its default size.

Update

Updates the simulation setup. Use this action if your model is out of date and you encounter a problem regarding a CAD data or QuickPart update.

Common Object Right-Click Actions

The following action is available when you right-click an object node in the tree:

Edit

Allows you to edit the properties of the selected object. Applying this action to several QuickParts of the same type enables you to multi-edit the physics. Hold down the <Ctrl> key when selecting the parts.

The following actions are available to all object nodes except for the **Setup** nodes:

Delete

Removes the selected object from the tree. Deleting the resulting part of a Construction Geometries Operation reverses the operation and restores the input parts. When you delete a QuickPart Surface, the corresponding faces are transferred back to the default surface.

Rename...

Allows you to change the name of the object under which it is stored in the tree.

Set New Tag...

Creates a new tag and adds it to the selected object. Tags can be used to organize objects in the model. If you have a large simulation with many objects in the tree, you can add a custom tree for quick access to the objects that are associated with a tag. See Using Custom Simulation Trees.

Set Tags...

Allows you to set a new or previously created tag to the selected object. For more information, see Set New Tag.

Contents:

Setup
Parameters
Construction Geometries
QuickParts
Placement
Interactive Right-Click Actions
Mesh
Scenes
Results
Field Functions

Setup

The **Setup** manager node allows you to define the physics, default units, and solver settings for an electronics cooling simulation.

Physics

For the gas domain, you can choose from the following flow and energy scenarios:

- **Forced Convection**—the fluid movement results only from external sources (for example a fan).

  You solve for flow and/or energy within a turbulent or laminar flow regime. The density of the fluid phase is assumed to be constant throughout the domain.
- **Natural Convection**—as Forced Convection, but includes gravity effects and radiation.

  This scenario allows you to model buoyancy due to small density differences that are caused by temperature gradients.
- **Custom**—as Natural Convection, but you can choose to use the ideal gas model which allows you to model flows with large variations of density due to temperature and pressure variations.

For a liquid domain, such as a cooling loop, the default scenario is **Forced Convection**.

The **Setup** > **Physics** node allows you to set the following properties that are common to all fluid domains and flow scenarios:

Physics—Common Properties	
Ambient Conditions	Specifies the physical conditions of the environment. **Temperature** Specifies the temperature of the environment. This temperature is used as default value for every fixed temperature setting, for example, for a fixed temperature boundary condition on a wall. **Note:** If you want to change a specific fixed temperature setting to another temperature value, delete the $DefaultAmbientTemperature entry and enter the required value.
Solution Physics	Specifies the physical effects that you want to consider in the simulation. **Solve for** Defines the equations that you want to solve. • **Flow**: When activated, solves the Navier-Stokes equations for continuity and momentum (see Fluid Flow). Selecting this option allows you to evaluate the flow field in the fluid domains. • **Energy**: When activated, solves the energy equation (see Fluid Flow). Selecting this option enables you to analyze the temperature distribution within the fluid and solid domains. **Pressure** Specifies the reference pressure for the gas domain and liquid domain, respectively. The reference pressure is simply a device that is used to reduce the numerical roundoff error in the numerical calculations involving pressure. This reduction in numerical error is necessary since the differences in pressure are important, and these differences can be small relative to the absolute value of the pressure (for example, in fully incompressible, or very low-Mach number flows). By subtracting a suitable constant reference pressure, a working pressure is obtained that is less prone to roundoff errors. For constant density flows, the actual value of the reference pressure has no relevance to the calculations. However, when using the ideal gas model, the reference pressure is used as in Eqn. (1466). Set the reference pressure to some ambient value such that the working pressures are small. The default value is 101325.0 Pa. **Flow Regime** Defines the flow pattern within the gas and liquid domain, respectively. • **Turbulent**: Solves for a flow that is in a state of continuous instability and exhibits irregular, small-scale, high-frequency fluctuations.

Physics—Common Properties	
	Turbulent flow is modeled using a K-Epsilon approach to provide closure to the Reynolds-Averaged Navier-Stokes (RANS) equations. For more detailed information on the turbulence model that is used, see Realizable Two-Layer K-Epsilon Model. • **Laminar**: Solves for a well-ordered flow that is free of macroscopic, non-repeating fluctuations. Numerical instabilities can arise from simulating laminar flows at a Reynolds number (the ratio of viscous to inertial forces) that is too large. These instabilities can impede convergence. Therefore, a laminar flow simulation is only appropriate if you already know that the Reynolds number of the problem is sufficiently low.
Materials	Specifies the materials that are present within the electronic device. **Solid Sim Materials** Specifies the materials of the different solid components. **Default Solid** Specifies the default material that is used for a new solid QuickPart. To reduce the time to set up the solid components, select the material that characterizes most components. **Note:** Usually, most materials that are present within an electronics device are isotropic. For this reason, Solid Sim Materials with anisotropic thermal conductivity are not supported as Default Solid. **Default Gas** Specifies the material of the gas domain. **Default Liquid** Specifies the material of the liquid domain.

The following properties are available for the gas domain. The availability of the properties depends on the flow scenario that you choose in the **Solve** drop-down menu within the **Solution Physics** box:

Solution Physics—Gases		Natural Convection	Custom
Gas Model	Defines the equation of state for the density of the gaseous phase. The following options are available: • **Constant Density**: The density of the gaseous phase is assumed to be invariant throughout the domain.		✓

Solution Physics—Gases		Natural Convection	Custom
	• **Ideal Gas**: Uses the ideal gas law to express density as a function of temperature and pressure.		
Gravity Effects	Allows you to account for the effect of gravitational acceleration in the gaseous phase due to density variations. For more information, see Modeling Gravity. When activated, the following options are available: • **X**, **Y**, or **Z**: Sets the direction in which gravity acts to the respective axis of the Laboratory coordinate system. • **Custom**: Set the components of the gravity vector to the specified **X**, **Y**, and **Z** values.	✔	✔
Radiation	Allows you to include radiative heat transfer between diffuse surfaces independently of wavelength. The medium that fills the space between the surfaces is non-participating. That is, it does not absorb, emit, or scatter radiation. For more information, see Modeling Surface-to-Surface Radiation and The Gray Thermal Radiation Model. When activated, the following properties are available: **Radiation Temperature** Specifies the temperature that is effectively radiated from the environment. The thermal environment is modeled as a black body with unit emissivity. **Default Surface Emissivity** Specifies the default surface emissivity that is set on every new QuickPartSurface. Set the emissivity that applies to most of the surfaces.	✔	✔

Units

The **Setup** > **Units** node allows you to set the preferred units for the various fundamental and derived dimensions (like temperature or mass flow rate):

Units Properties	
Dimension	Specifies the physical dimension of the unit.
Preferred Units	Specifies the preferred unit for the dimension. For each dimensional combination in use in the simulation, the Electronics Cooling Toolset marks exactly one unit to be the preferred unit for that dimension. For example the preferred unit for quantities with the dimension of length might be m and the preferred unit for quantities with dimensions of length per time might be m/s. The

Units Properties	
	preferred units are the ones that are used by default for any new physical quantity. You can change the preferred unit for an individual dimensional combination. Although the preferred unit is used by default, you can override it on specific settings. For example, you might have the preferred unit for dimensions of velocity to be m/s, but for a specific velocity boundary condition you could change the units to km/h.

Units Actions	
(**Add a new unit**)	Adds a user-defined unit. The following properties characterize a user-defined unit: **Description** Specifies the full name of the unit. **Conversion** Specifies the conversion factor that converts the quantity from this unit to SI units. **Offset** Specifies the offset factor used to convert the quantity from this unit to SI units (for example for Celsius and Fahrenheit temperatures).
(**Edit Unit**)	Allows you to edit the properties of a user-defined unit.
(**Delete Unit**)	Deletes a user-defined unit.

Solver

The **Setup** > **Solver** node allows you to set the following stopping criteria and numerical parameters for the solver:

Solver Settings	
Stopping Criteria	Defines how long the solution runs. The stopping decision is based on the number of steps that the solver executes, including any steps that are executed in a previous session. If you clear the solution, the counter resets to zero. **Maximum Steps** Specifies the maximum number of steps to take before the criterion is satisfied. The default is 1000 steps.
Segregated Flow	Controls the solution update and thus the convergence of the solvers that compute the intermediate velocity field and the update of the pressure field. **Velocity Under-Relaxation Factor, Pressure Under-Relaxation Factor** At each iteration, these properties govern the extent to which the newly computed solution supplants the old solution. For the theoretical background, see Finite Volume Discretization.

Solver Settings	
	The default values of 0.7 for velocity and 0.3 for pressure are conservative, that is they lead to a converged solution in most cases. For further guidelines, see Setting Under-Relaxation Factors for Steady State Computations.
Segregated Energy	Controls the solution update of the solver that computes the temperature field in the fluid and solid domains. **Fluid Under-Relaxation Factor, Solid Under-Relaxation Factor** In order to promote convergence, these properties under-relax changes of the solution during the iterative process, see Finite Volume Discretization. The default value for both fluid energy and solid energy is 1.0 (that is no under-relaxation). These values are progressive parameters that produce a fast solution convergence for a wide range of physical situations. Sometimes it is necessary to decrease the fluid energy relaxation to 0.9 and the solid energy relaxation to 0.99 to achieve a stable solution.

Parameters

The **Parameters** node allows you to define, edit, and delete global parameters, which you can use in multiple Construction Geometries, QuickParts, or QuickPart Surfaces.

Parameters Properties	
Parameter	Specifies the name of the global parameter, such as `MyNumberOfFins` for a scalar parameter.
Dimensions	Specifies the dimension of the global parameter. The dimension that you set determines the SI unit in the **Value** text-box. For example, **Temperature** gives you the unit K. **Dimensionless** gives you no unit.
Value	Specifies the value of the global parameter.

Parameters Actions	
Add a new parameter	Adds a new global parameter.
(**Delete**)	Deletes a global parameter.

Construction Geometries

The **Construction Geometry** manager node allows you to create and edit geometry objects.

Two types of **Construction Geometries** ( ) are available: two-dimensional and three-dimensional.

Two-dimensional Construction Geometries, called Sheet Construction Geometries, are geometries that are composed from surfaces that have zero-thickness. A Sheet Construction Geometry allows you to determine

the shape and position of a hole through a QuickPart. Furthermore, you can imprint the profile of a two-dimensional Construction Object on a QuickPart to split the respective surface. For more information, see Interactions.

Three-dimensional Construction Geometries describe closed bodies that have a finite volume. These Construction Geometries allow you to model electric components that are geometrically more complex than the prescribed templates of the respective QuickPart. Once such a Construction Geometry is created it can be used to define the shape of any Custom QuickPart.

To create a Construction Geometry within the Electronics Cooling Toolset, the following options are available:

- Create a geometry that is based on a simple shape template.
- Import a geometry from another CAD package.
- Create a geometry within 3D-CAD and transfer it into the Electronics Cooling Toolset.

To define the placement of a Construction Geometry, the following options are available:

- Specifying the Placement properties

  You define the location and orientation of the Construction Geometry with respect to a selected coordinate system.
- Using the Positioning Tool

  The Positioning Tool allows you to place Construction Geometries in reference to each other.

Furthermore, the Electronics Cooling Toolset offers you two possibilities to operate on Construction Geometries:

- Apply Boolean operations—unite and subtract—on previously created Construction Geometries.
- Generate a pattern of identical geometries from a seed Construction Geometry.

Common Construction Geometry Properties

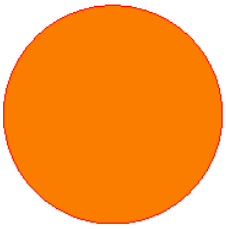
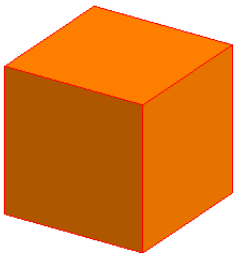
The following properties are common to all types of Construction Geometries:

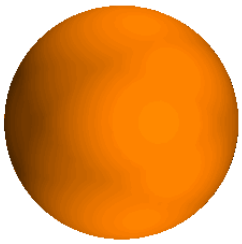
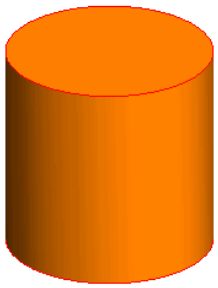
Common Properties	
Name	Specifies the name of the Construction Geometry under which it is stored in the tree.
Placement	See Placement properties.

Construction Geometries Shape Templates

The Electronics Cooling Toolset allows you to create Construction Geometries that are based on simple shape templates. The following geometry properties characterize the different two- and three-dimensional Construction Geometry shape templates:

Shape Template	Geometry Properties
Sheet Body Rectangle	**Length** Specifies the length of the rectangle.

Shape Template	Geometry Properties
	Width Specifies the width of the rectangle. **Local Origin** Specifies the position of the rectangle's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. It is used as reference for placing the geometry in the simulation domain. The following options are available: • **Corner 1**: Xmin/Ymin/Zmin corner of the rectangle. • **Center**: Center of the rectangle. • **Custom**: This value appears when you set the local origin interactively within the *Graphics* window.
Sheet Body Circle 	**Radius** Specifies the radius of the circle. **Note:** The **Local Origin** of a circular sheet body is the center of the circular face, by default.
Block 	**X, Y, Z** Specifies the dimensions of the block in x-, y-, and z-direction, in the geometry's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. **Local Origin** Specifies the position of the block's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. It is used as reference for placing the geometry in the simulation domain. The following options are available: • **Bottom Center**: Center of the Zmin face. • **Center**: Center of the block. • **Bottom Corner 1**: Xmin/Ymin/Zmin corner of the block. • **Bottom Corner 2**: Xmin/Ymax/Zmin corner of the block. • **Custom**: This value appears when you set the local origin interactively within the *Graphics* window. **Calculate Size from QuickParts** Allows you to calculate the dimensions of the block based on the maximum and minimum coordinates of particular QuickParts that you specify. You can modify the size of the bounding block by entering offsets in negative and positive coordinate directions.

Shape Template	Geometry Properties
	This option is particularly useful for defining a volumetric zone for mesh customization around specific QuickParts.
Sphere 	**Radius** Specifies the radius of the sphere. **Calculate Size from QuickParts** Allows you to calculate the radius of the sphere as half the maximum distance between particular QuickParts that you specify. You can extend the size of the bounding sphere by adding an offset to the radius. This option is particularly useful for defining a volumetric zone for mesh customization around specific QuickParts. **Note:** The **Local Origin** of a sphere is the centroid of the sphere, by default.
Cylinder 	**Height** Specifies the length of the cylinder from bottom to top. **Base Radius** Specifies the radius of the cylinder base. **Tapered Cylinder** When activated, the **Top Radius** specifies the radius on top of the cylinder; otherwise the cylinder has a constant radius throughout. **Local Origin** Specifies the position of the block's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. It is used as reference for placing the geometry in the simulation domain. The following options are available: • **Bottom Center**: Center of the Zmin face. • **Center**: Center of the cylinder. • **Custom**: This value appears when you set the local origin interactively within the *Graphics* window.

Imported and Transferred Construction Geometries

To model components of complex geometry, the Electronics Cooling Toolset allows both importing geometry from another CAD package and transferring CAD bodies from the 3D-CAD modeler within STAR-CCM+.

The import process supports geometries that are geometrically and topologically valid from the Parasolid geometry kernel point of view. For information on the CAD file formats that are supported for import, see Supported CAD file formats.

The transfer process allows you to send CAD bodies from the 3D-CAD modeler to the Electronics Cooling Toolset. Launching the Electronics Cooling Toolset automatically creates the 3D-CAD

model **Electronics Cooling 3D-CAD 1** that contains the geometries of all electric components that you create within the Electronics Cooling Toolset. By directly entering this model, you can use the capabilities of the 3D-CAD modeler to create advanced geometries. For details on how to build a geometry in 3D-CAD, see Using 3D-CAD.

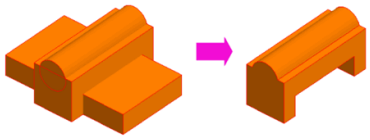
The following geometry properties are common to both imported and transferred Construction Geometries:

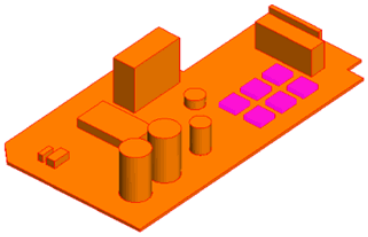
Source of Construction Geometry	Geometry Properties
Another CAD package 3D-CAD modeler in STAR-CCM+	**Local Origin** Specifies the position of the geometry's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. It is used as reference for placing the geometry in the simulation domain. The following options are available: • **Center of Bounding Box**: Center of the box that bounds the imported geometry. • **Bottom (ZMin) Center of Bounding Box**: Center of the Zmin face of the bounding box. • **Bounding Box: +-X/+-Y/+-Z**: A corner of the bounding box. For example, Bounding Box: +X/-Y/+Z sets the local origin to the corner that corresponds to the maximum X coordinate, the minimum Y coordinate and the maximum Z coordinate. • **Custom**: This value appears when you set the local origin interactively within the *Graphics* window.

Construction Geometries Operations

The Electronics Cooling Toolset allows you to modify or multiply previously created Construction Geometries. The **Boolean Construction Geometry** action performs a unite or a subtract operation on selected Construction Geometries. The **Pattern Construction Geometry** action generates a series of identical objects from a seed Construction Geometry, which follow a regular formation along straight lines.

The following geometry properties describe the various types of operations:

Operation	Geometry Properties
Boolean Construction Geometry 	**To Unite** Specifies the Construction Geometries that the unite operation is performed on. It unites all input objects that are in contact with or overlap each other. **To Subtract** Specifies the Construction Geometries to remove from the target object. The specified Construction Geometry is consumed in the operation.

Operation	Geometry Properties
Pattern Construction Geometry 	**Seed Construction Object** Specifies the Construction Geometry to be reproduced multiple times using a linear pattern. **Direction (n)** Defines the parameters for one of the two directions in the linear pattern: **Direction(n)** Specifies the direction in which the pattern extends. The **X**, **Y**, and **Z** components of the direction vector are defined with respect to the selected **Coordinate System**. **Distance** Specifies the gap between each object instance. **Number of Instances** Specifies how many times the body is reproduced in each direction. The value must be a whole number. If you enter a decimal value, it is rounded down to a whole number.

Note: The **Local Origin** of the resulting geometry is placed in the origin of the Laboratory coordinate system by default (see X_or, Y_or, Z_or in the *Graphics* window).

Select Construction Geometry Dialog

Using the *Select Construction Geometry* dialog, you can map Construction Geometries to their corresponding Custom QuickParts. The dialog is split into two main sections. The top section shows all the Construction Geometries in the simulation that are not already mapped to a QuickPart. The bottom section contains the following options for setting the QuickParts and QuickPart Surfaces:

QuickPart Options	
Use Construction Geometry Name	When **On**, names the Custom QuickPart after the name of the selected Construction Geometry. For an imported or transferred Construction Geometry, this is the name of the respective CAD body. When **Off**, names the Custom QuickPart after the QuickPart type, such as **Chip 2**.
Create QuickPart Surface for Named Faces	When **On**, creates a QuickPart Surface for each face that is named in the selected Construction Geometry. This option is only relevant for imported and transferred Construction Geometries, where the surface of the respective CAD body is composed of several named faces.

QuickPart Options	
	When **Off**, creates one **Default** QuickPart Surface for all faces of the selected Construction Geometry.

Construction Geometry Manager Right-Click Actions

The following actions are available when you right-click the **Construction Geometry** manager node.

Import CAD Model

Imports geometry from another CAD package.

New Construction Geometry > [Shape Template]

Creates a new Construction Object of the selected shape template using a graphical editor.

Transfer Cad Bodies

Transfers CAD bodies from the 3D-CAD modeler into the Electronics Cooling module.

Show Only Unused Construction Geometry

Hides all Construction Geometries from the *Graphics* window that are used to define the geometry of a Custom QuickPart.

Check Interference

Checks if Construction Geometries are intersecting each other. Intersecting Construction Geometries are listed in the *Output* window.

Construction Geometries Common Right-Click Actions

The following specific actions are available when you right-click a particular Construction Geometry:

Boolean Construction Geometry

Allows you to perform Boolean unite and subtract operations on the selected Construction Geometry. The result is a new Construction Geometry that is added to the Construction Geometry manager node.

Note: This action is not available for Sheet Construction Geometries.

Pattern Construction Geometry

Allows you to perform a linear pattern operation on the selected Construction Geometry. The result is a new Construction Geometry that contains multiple identical geometries that follow a regular formation along straight lines. You can define the pattern by one or two linear directions extending from the original Construction Geometry. The number of bodies and the distance between them determine the extent of the pattern.

Hide

Hides the selected Construction Geometry from the *Graphics* window. This action is useful for gaining a clear view of other objects. You can restore hidden objects again by using the Show action.

Show

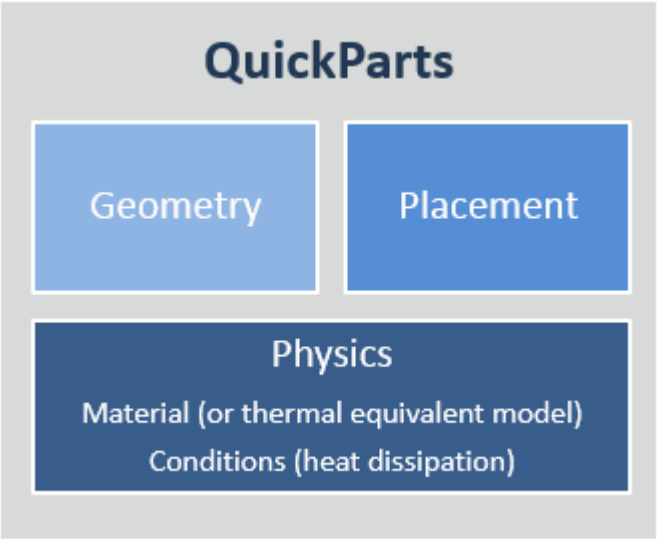
Restores a hidden Construction Geometry.

Position

Launches the Positioning Tool that allows you to interactively place the selected Construction Geometry in reference to other Construction Geometries or QuickParts.

QuickParts

QuickParts describe the components of an electronics device and incorporate geometry, placement, and physics settings.



QuickParts are classified according to their role in an electronics simulation. The following types are available:

- Heatsink
- Chip
- PCB
- Enclosure
- Axial Fan
- Radial Fan
- Resistance
- Generic

To describe the geometry of a specific QuickPart type the following approaches are available:

- Using templated geometries

  Templated QuickParts follow a prescribed geometry template. For instance, a templated chip can be defined as rectangular block or as cylindrical block. Likewise, a templated heatsink can be of type Plate

Fin, Pin Fin, Cross Cut, or Compact. Each of these geometrical types follows a different but prescribed geometry template.

- Using customized geometries

  Custom QuickParts allow you to use arbitrary geometries that you obtain from Construction Geometries.

Furthermore, the Electronics Cooling Toolset allows the import of IDF (Intermediate Data Format) files. IDF files contain design information about circuit boards and electronic components. The import process reads these files and builds the circuit board using PCB and Generic QuickParts.

To define the placement of a QuickPart, the following options are available:

- Specifying the Placement properties

  You define the location and orientation of the QuickPart with respect to a selected coordinate system.

- Using the Positioning Tool

  The Positioning Tool allows you to place QuickParts in reference to each other.

The physics settings of a QuickPart incorporate material and heat dissipation properties. For Chip QuickParts, compact thermal models allow you to define the heat flow paths within a chip package without resolving the internal volume of the chip.

QuickParts Manager Right-Click Actions

The following actions are available when you right-click the **QuickParts** manager node.

Import IDF file

Allows you to import IDF files that contain the layout definitions of the board and its components. You can filter the components that you want to import using the Import IDF Options Panel.

New QuickPart > [Type] > [Geometry]

Creates a new QuickPart of the selected type and geometry description, using a graphical editor in the *Graphics* window.

Paste QuickParts

Pastes the copied QuickParts from the clipboard. If you select this action when you right-click in the *Graphics* window, the QuickParts are pasted at the selected location. If you select this action when you right-click the **QuickParts** manager node, the QuickParts are pasted in the tree and appear in the *Graphics* window at the origin location (0,0,0). In both cases, the relative orientations and distances between the QuickParts are retained as in the original, and you can then move the QuickParts to the appropriate locations.

Create Extract Volume

Finds enclosed volumes within selected QuickParts or Construction Objects and generates a new part to fill the volume. See Extract Volume.

Check Interference

Checks if QuickParts are intersecting each other. Intersecting QuickParts are listed in the *Output* window.

QuickParts Common Right-Click Actions

The following specific actions are available when you right-click a particular QuickPart:

Highlight

Highlights the selected QuickPart in the *Graphics* window.

Composite

Groups selected QuickParts into an assembly for easier organization. To select several parts at once, hold down the <Ctrl> key when selecting the parts.

Hide

Hides the selected QuickPart from the *Graphics* window. This action is useful for gaining a clear view of other components. You can restore hidden parts again by using the Show action.

Show

Restores a hidden QuickPart.

Edit Interactions

Allows you to define interactions with other QuickParts and Construction Objects, see Interactions.

Position

Launches the Positioning Tool that allows you to interactively place the selected QuickPart in reference to other QuickParts or Construction Objects.

Copy QuickParts

Copies the selected QuickParts to the clipboard. The **Paste QuickParts** action becomes available when you right-click the **QuickParts** manager node or right-click in the *Graphics* window.

Contents:

Import IDF Options Panel
Heatsink
Chip
PCB
Enclosure
Axial Fan
Radial Fan
Resistance
Generic
Interactions
Extract Volume
QuickPart Surfaces

Import IDF Options Panel

The *Import IDF Options* panel allows you to specify what components of the IDF file to import into the Electronics Cooling Toolset.

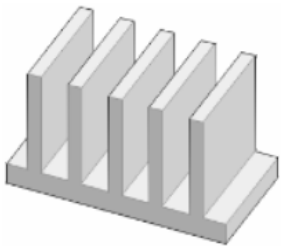
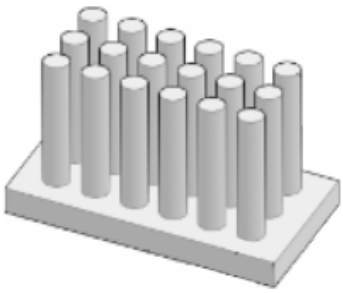
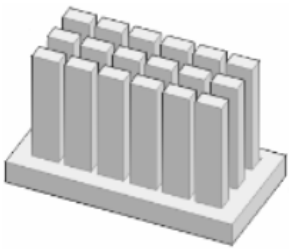
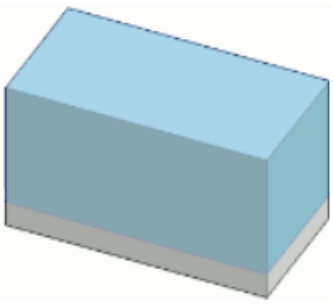
The following filter options are available:

Reference Designator Filter	Excludes components based on their reference designator. A reference designator is a component identifier that is specified in the IDF library file. The **Select** button opens an object selector panel that lists the various reference designators that are available in the file. By default, no components are filtered out.
Property Value Filter	Excludes the components that are smaller than the specified **Property Size** from the import. The size of the component is defined as the maximum edge in the bounding box that encloses the 2D profile of the component. The **Add Filter** adds the filter to the **Current Filter** list.
Geometry Feature Filter	Excludes holes that are smaller than the specified **Hole Radius** from the import. The **Add Filter** adds the filter to the **Current Filter** list.
Current Filter	Contains a list of active filters. The **Delete Filter** button removes selected filters from the list.
IDF 2D Sketch (Filters Applied)	Displays the front and back of the PCB and its components after filtering. Changing the filters dynamically updates the preview.

Heatsink

Heat sinks cool electronic components by transferring the generated heat into the surrounding fluid.

The Electronics Cooling Toolset supports the following **Heatsink** QuickParts:

Plate Fin	Pin Fin	Cross Cut	Compact
			

While Plate Fin, Pin Fin, and Cross Cut heat sinks resolve the geometry of a real heat sink using simple shape templates, a Compact heat sink enables you to model the behaviour of a heat sink without considering geometric details. The Compact heat sink uses a Porous Medium approach for the fin region to capture the effects on the overall fluid flow and heat transfer. With this approach, you do not have to define a fine mesh around the fins or pins of a heat sink.

If none of the built-in heat sinks are suitable, you can define a Custom heat sink whose geometry is based on Construction Geometries that you specify.

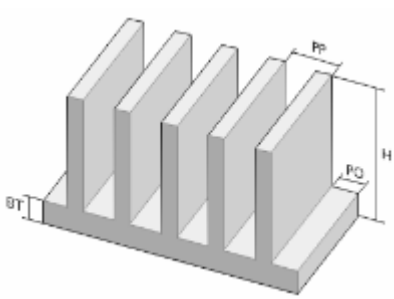
Heat Sink—Common Properties

The following properties are common to all types of heat sinks:

Name	Specifies the name of the heat sink under which it is stored in the tree.
Geometry	**Local Origin** Specifies the position of the heat sink's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. It is used for placing the heat sink in the simulation domain. The following options are available: • **A**: Center of the bottom (Zmin) face. • **B**: Xmin/Ymin corner of the bottom (Zmin) face. • **C**: Xmin/Ymax corner of the bottom (Zmin) face. • **Custom**: This value appears when you set the local origin interactively within the *Graphics* window, see Placing Object Interactively.
Placement	See Placement properties.

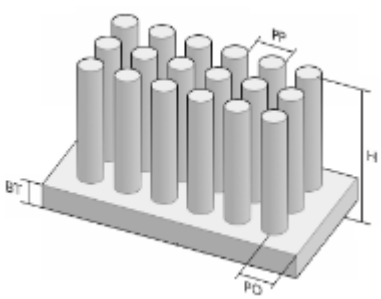
The following geometry and physics properties characterize the different types of heat sinks—if you specify geometry parameters that lead to an invalid body, the parameters that require correction are highlighted in red:

Plate Fin—Specific Properties

Geometry	
	X, Y Specify the dimensions in x- and y-direction, in the heat sink's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. **Height - (H)** Specifies the height from base to top of the fins.

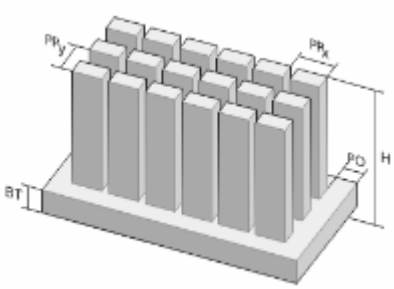
	Base Thickness - (BT) Specifies the thickness of the heat sink's base. **Fin Offset - (PO)** Specifies the offset of the plate fins from the base's sides. **Fin Thickness** Specifies the thickness of each plate fin. **Number of Fins** Specifies the number of plate fins that are placed on the heat sink's base.
Physics	**Fins Bonded to Base** Controls if the fins of the heat sink are bonded to or extruded from the base. When **Off**, a **Material** property is available to specify the overall material of the extruded heat sink. When **On**, a **Base Material**, and a **Fin Material** property specify the material of the heat sink's base and fin respectively. A **Contact Resistance** property sets the contact resistance between the fins and the heat sink's base. You can choose the materials from the Solid Sim Materials that you define in the Setup panel.

Pin Fin—Specific Properties

Geometry	
	X, Y Specifies the dimensions in x- and y-direction, in the heat sink's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. **Height - (H)** Specifies the height from base to top of the fins. **Base Thickness - (BT)** Specifies the thickness of the heat sink's base.

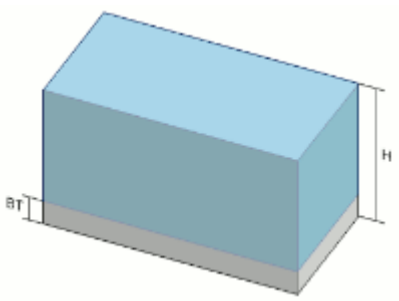
	Min Pin Offset - (PO) Specifies the minimum offset of the pin fins from the base's sides. **Pin Radius** Specifies the radius of each pin fin. **Pin Pin Spacing** Controls the spacing between the pin fins. The following options are available: • **Pin-Pin Distance**: Sets the spacing based on the distance between the pin fins within one line. This option activates the **Pitch - (PP)** property that specifies the distance from one pin fin to the next. • **Number of Pins**: Sets the spacing based on the number of pin fins within one line. This option activates the **Count X** and **Count Y** properties that specify the number of pin fins in x- and y-direction, respectively, in the heat sink's local coordinate system. **Rows Staggered** Controls if the pin fins are arranged in-line or staggered.
Physics	**Material** Defines the material of the heat sink. You can choose the material from the Solid Sim Materials that you define in the Setup panel.

Cross Cut—Specific Properties

Geometry	
	X, Y Specifies the dimensions in x- and y-direction, in the heat sink's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. **Height - (H)** Specifies the height from base to top of the fins. **Base Thickness - (BT)** Specifies the thickness of the heat sink's base.

	Min Pin Offset - (PO) Specifies the minimum offset of the pin fins from the base's sides. **Pin Length** Specifies the length of each pin fin. **Pin Width** Specifies the width of each pin fin. **Pin Pin Spacing** Controls the spacing between the pin fins. The following options are available: • **Pin-Pin Distance**: Sets the spacing based on the distance between the pin fins within one line. This option activates the **Pitch - (PP)** property that specifies the distance from one pin fin to the next. • **Number of Pins**: Sets the spacing based on the number of pin fins within one line. This option activates the **Count X** and **Count Y** properties that specify the number of pin fins in x- and y-direction, respectively, in the heat sink's local coordinate system. **Rows Staggered** Controls if the pin fins are arranged in-line or staggered.
Physics	**Material** Defines the material of the heat sink. You can choose the material from the Solid Sim Materials that you define in the Setup panel.

Compact—Specific Properties

Geometry	
	X, Y Specifies the dimensions in x- and y-direction, in the heat sink's local coordinate system. See X_or, Y_or, Z_or in the visualization display. **Height - (H)** Specifies the height from base to top of the porous block.

	Optional Base Controls if the heat sink has a base or not. When **Present**, a **Base Thickness - (BT)** property specifies the thickness of the base. For the physics, you specify a **Base Material** and a **Contact Resistance** property.
Physics	**Base Material** Specifies the material of the heat sink's base. You can choose from the Solid Sim Materials that you define in the Setup panel. **Fin Properties** Specifies the properties of a porous medium that is used to mimic the effect of actual fins on the overall flow and heat transfer. Each parameter requires an **XXComponent**, a **YY Component**, and a **ZZ Component** that represent the diagonal components of a Principal Tensor Profile. The following parameters are available: **Porous Inertial Resistance** Specifies the inertial resistance caused by the modeled fins, see Porous Media Drag Model Region Settings. **Porous Viscous Resistance** Specifies the viscous resistance caused by the modeled fins, see Porous Media Drag Model Region Settings. **Solid Thermal Conductivity** Specifies the thermal conductivity of the modeled fins' material, see Solid Thermal Conductivity. **Porosity** Specifies the ratio of open volume to total volume within the heat sink. **Contact Resistance** Specifies the contact resistance between the fins and the heat sink's base.

Custom—Specific Properties

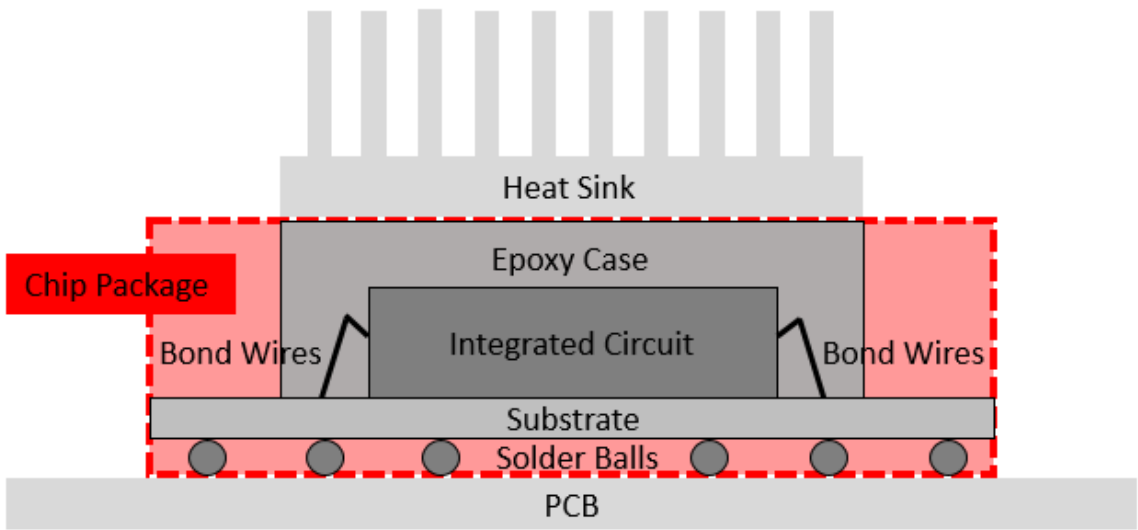
Geometry	**Base** Specifies the Construction Geometries that define the geometry of the heat sink's base. **Fins** Specifies the Construction Geometries that define the geometry of the fins.

Physics	**Fins Bonded to Base** Controls if the fins of the heat sink are bonded to or extruded from the base. When **Off**, a **Material** property is available to specify the overall material of the extruded heat sink. When **On**, a **Base Material**, and a **Fin Material** property specify the material of the heat sink's base and fin respectively. A **Contact Resistance** property sets the contact resistance between the fins and the heat sink's base. You can choose the materials from the Solid Sim Materials that you define in the Setup panel.

Chip

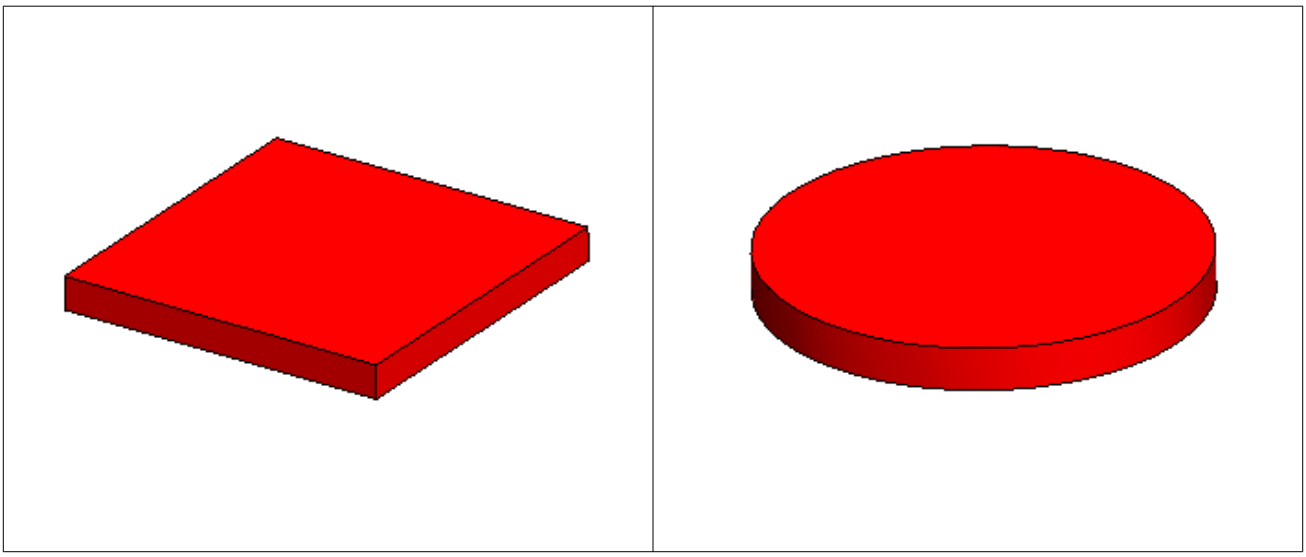
Chips are modules made of semiconductor material on which integrated circuits are embedded. Chips generate excess heat and thus require cooling to improve reliability and avoid premature failure.

The following image displays the various elements of a chip package:



To represent the geometry of a chip package, the Electronics Cooling Toolset supports the following **Chip** QuickParts:

Block	Cylinder

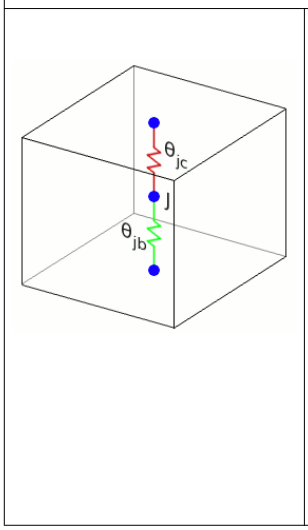
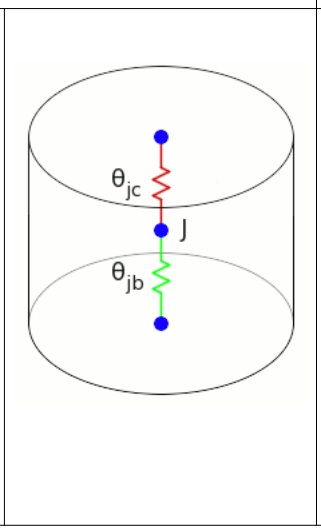
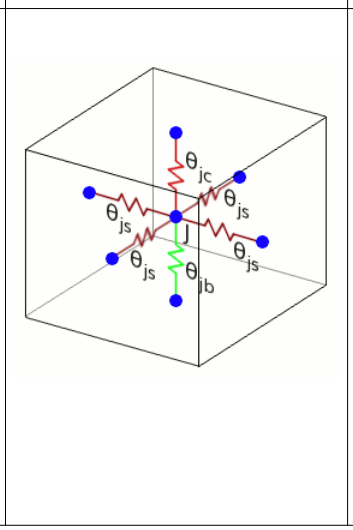
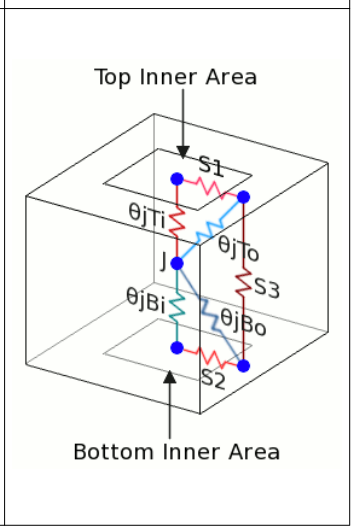


If the built-in shape templates are not suitable, you can define a Custom chip whose geometry is based on a Construction Geometry that you specify.

To represent the physics of a chip package, two thermal modeling approaches are available:

- Solid model: represents the chip package as a single solid part of a specified material. A total heat source models the excess heat of the package.
- Compact thermal model (CTM) ([191], [192]): a simplified component model that mimics the performance of the chip package within the system-level analysis. It represents the critical heat flow paths within the package using a thermal resistor network. Each network node corresponds to a surface of the package or a region within the package and is associated with a single temperature.

A CTM of a chip package is more accurate than a solid model. As a CTM does not require a volume mesh within the package, it is also computationally less expensive. On the other hand, the resistor network of a CTM requires that you have access to detailed information about the heat flow paths within the package. The following types of resistor networks are available:

Two Resistor		Star Network	Delphi Network
			

The various network types differ in the number of nodes and the connections between them. Each resistor network owns the following three nodes:

- Junction node—that represents the integrated circuit.
- Case node—that describes the top surface of the package.
- Board node—that is considered as the contact between the package and the PCB.

A Star Network adds additional nodes to the side surfaces. A Delphi Network splits the top surface and the contacting area between the package and the PCB in an inner and an outer area, respectively, and represents them by separate nodes.

Two resistor and Delphi networks are suitable for packages where the heat transfer through the sides can be ignored. For example, within a typical BGA (ball grid array) package most of the heat flows through the solderballs to the PCB at the bottom and/or to the heatsink on top. A two resistor network assumes uniform temperatures on the top and bottom surfaces, whereas a Delphi network relaxes this assumption by allowing different temperature in the inner and outer areas of the surfaces.

Star networks are suitable for packages where the heat transfer through the sides can not be neglected. For example, within a leaded package the heat flow path through the sides significantly affects the package temperature. The available star network assumes a uniform temperature across the side surfaces.

Within the Electronics Cooling Module, each network node—apart from the junction node—corresponds to a QuickPart Surface of constant and uniform temperature. Surfaces without a network node are modeled as adiabatic. For more information, see Resistor Networks Formulation.

Chip—Properties

The following properties are common to all types of chip packages:

Name	Specifies the name of the chip under which it is stored in the tree.
Geometry	**Local Origin** Specifies the location of the chip's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. It is used for placing the chip in the simulation domain. The following options are available: • **Bottom Center**: Center of the Zmin face. • **Bottom Corner 1** (block chips only): Xmin/Ymin/Zmin corner of the block. • **Bottom Corner 2** (block chips only): Xmin/Ymax/Zmin corner of the block. • **Center**: Center of the geometry. • **Custom**: This value appears when you set the local origin interactively within the *Graphics* window, see Placing Object Interactively.
Placement	See Placement properties.
Physics	**Material Specification** Controls the thermal modeling approach for the chip package. The following options are available:

	• **Default Material Specification**: Models the chip package as a solid part. You select the **Material** of the solid part from the Solid Sim Materials that you define in the **Setup** panel. The **Heat Source** value specifies the total heat generated by the package. This option is available for templated and customized chip geometries. • Two Resistor, Star Network, or Delphi Network: Models a chip package using the CTM approach. These options are available for templated chip geometries only. The Star Network and the Delphi Network options are further restricted to block geometries. This option is available for templated and customized chip geometries.

The following properties characterize the different geometrical types:

Block—Specific Properties

Geometry	**X, Y, Z** Specifies the size of the block in x-, y, and z-direction, in the chip's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window.

Cylinder—Specific Properties

Geometry	**Radius** Specifies the radius of the cylinder. **Height** Specifies the extent of the cylinder from bottom to top.

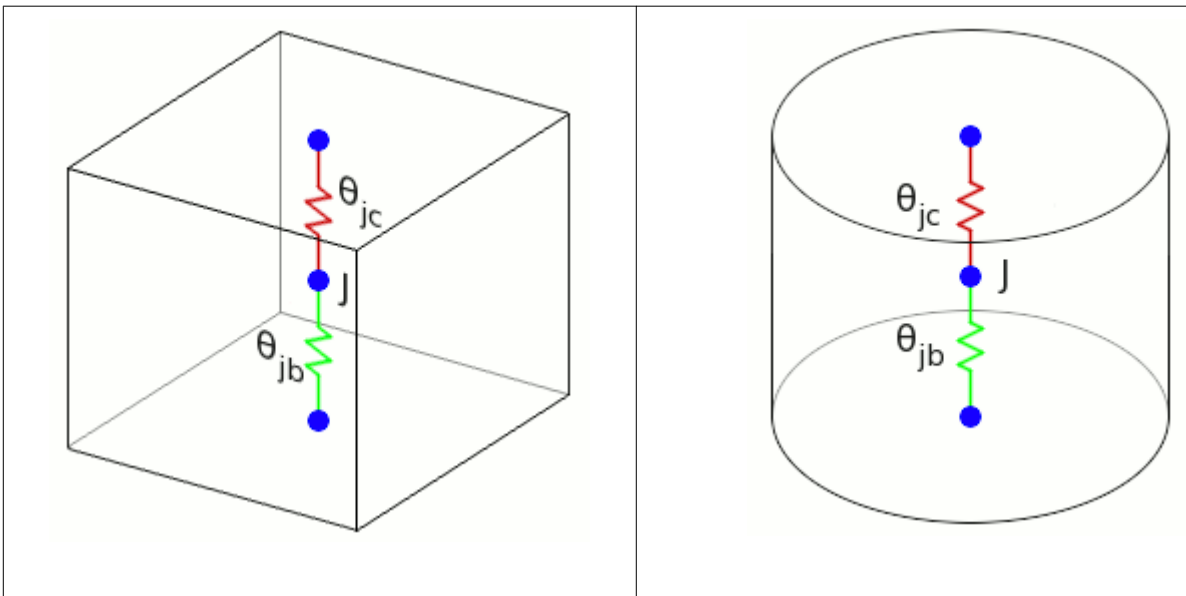
Custom—Specific Properties

Geometry	**Selected Construction Geometry** Specifies the Construction Geometry that describes the geometry of the chip. If you select multiple Construction Geometries when creating a Custom chip, the Electronics Cooling Toolset automatically creates multiple chips that own the same physics.

The following tables display the properties of the available resistor networks:

Note: Request the required input values from the package supplier or derive them from JEDEC-standard thermal tests. Alternatively, you can extract the values from a detailed thermal model of the package using a statistical optimization process that minimizes errors in junction and surface temperatures for a variety of environmental conditions.

Two Resistor



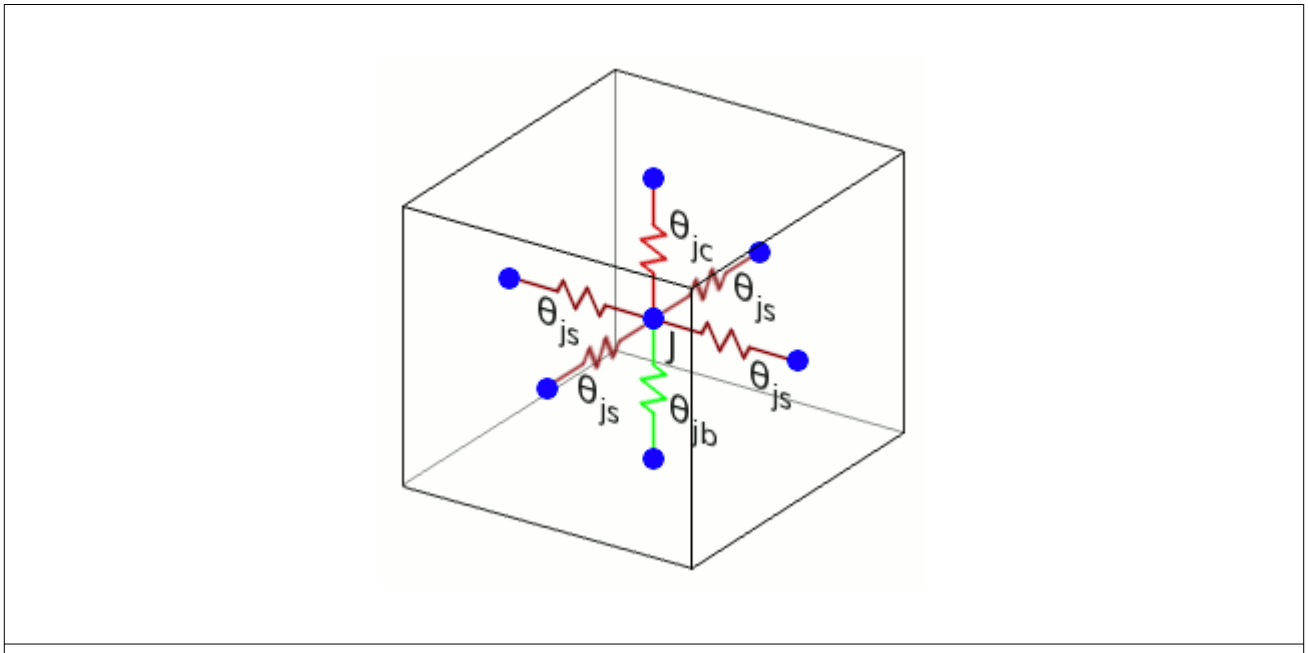
θjc, θjb

Specify the thermal resistance between the respective network nodes, see diagram above.

Heat Source

The total heating power of the integrated circuit that is applied at the junction node J.

Star Network



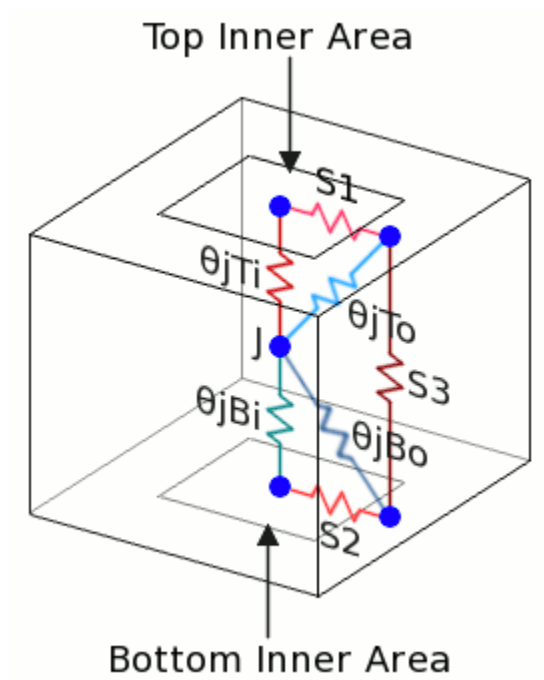
θjc, θjb, θjs

Specify the thermal resistance between the respective network nodes, see diagram above.

Heat Source

Specifies total heating power of the integrated circuit that is applied at the junction node J.

Delphi Network



% Top inner area, % Bottom inner area

Specify the top inner and bottom inner area ratios, respectively.

θjTi, θjTo, θjBi, θjBo, **S1, S2, S3**

Specify the thermal resistance between the respective network nodes, see diagram above.

Heat Source

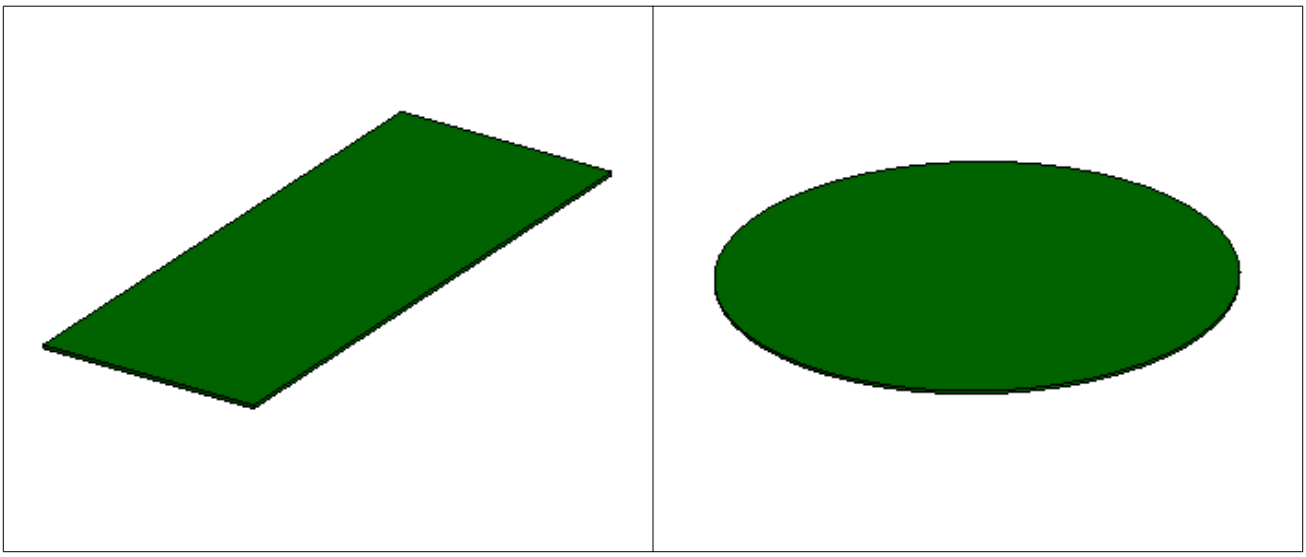
Specifies total heating power of the integrated circuit that is applied at the junction node J.

PCB

Printed circuit boards (PCBs) route signals and power between electronic components. They typically assemble conductive tracks and dielectric laminates in multiple layers.

The Electronics Cooling Toolset supports the following geometry templates for **PCB** QuickParts:

Rectangular	Circular



If none of the built-in PCBs are suitable, you can define a Custom PCB whose geometry is based on a Construction Geometry that you specify.

If you have IDF files that contain board and its components' layout definitions, you can import the files (see Import IFD file action). The Electronics Cooling Toolset automatically creates a PCB QuickPart for the board and Generic QuickParts for the components.

The thermal properties of a PCB can be specified as material dependent or layer dependent. The effect of conductive tracks and non-conductive substrate on the heat transfer is best described with layer dependent properties. The specific values depend on the number, thickness, and metal fractions of the trace layers.

PCB—Properties

The following properties are common to all types of PCBs:

Name	Specifies the name of the PCB under which it is stored in the tree.
Geometry	**Local Origin** Specifies the position of the PCB's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. It is used for placing the PCB in the simulation domain. The following options are available: • **Bottom Center**: Center of the Zmin face. • **Bottom Center of Range Box** (IDF import only): Center of the bottom face of the bounding box. • **Bottom Corner 1** (rectangular PCB only): Xmin/Ymin/Zmin corner of the rectangle. • **Bottom Corner 2** (rectangular PCB only): Xmin/Ymax/Zmin corner of the rectangle. • **Center**: Center of the geometry. • **Center of Range Box** (IDF import only): Center of the box that bounds the imported geometry. • **Custom**: This value appears when you set the local origin interactively within the *Graphics* window, see Placing Object Interactively. • **IDF Default Reference** (IDF import only): As specified in the IDF file.

Placement	See Placement properties.
Physics	**Heat Source** Specifies the total heat generated by the PCB. **Thermal Specification** Controls if the thermal properties of the PCB are modeled as material dependent or layer dependent. The following options are available: • **Specify Material**: Allows you to set a uniform **Material** throughout the PCB. You can choose the material from the Solid Sim Materials that you define in the Setup panel. If the Thermal Conductivity property of the respective material uses the Anisotropic method, you set the components of the **Thermal Conductivity** tensor using the principal tensor method. See Setting Values Using a Principal Tensor. The **XX Component**, **YY Component**, and **ZZ Component** are defined with respect to the axis of the QuickPart's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. • **Specify By Layer**: Allows you to set the **Number of Layers**, the **Metal Material**—that is the material of the conductive tracks and layers—, and the **Dielectric Material**—that is the laminate material. A table editor enables you to specify the thickness and metal fraction for each layer. For a Custom PCB, you need to set a **PCB Normal** to control the orientation of the layers.

The following geometry properties characterize the different types of PCBs:

Rectangular—Specific Properties

Geometry	**X, Y** Specify the dimensions in x- and y-direction, in the PCB's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. **Thickness** Specifies the extent of the rectangular PCB from bottom to top.

Circular—Specific Properties

Geometry	**Radius** Specifies the radius of the circular outline. **Thickness** Specifies the length that the circular PCB extends from bottom to top.

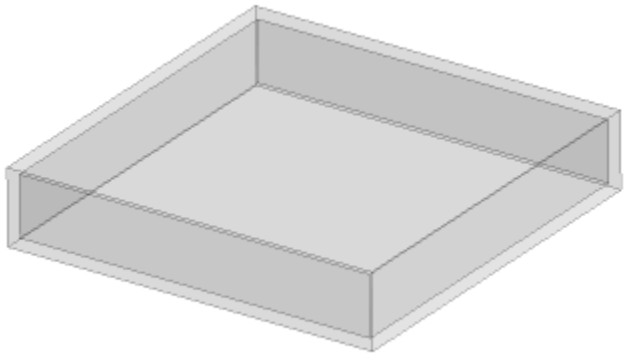
Custom—Specific Properties

Geometry	**Selected Construction Geometry** Specifies the Construction Geometry that describes the geometry of the PCB. If you select multiple Construction Geometries when creating a Custom PCB, the Electronics Cooling Toolset automatically creates multiple PCBs that own the same physics properties.

Enclosure

An enclosure houses the components of an electronics device and usually bounds the domain for the flow analysis.

The Electronics Cooling Toolset supports the following geometry template for an **Enclosure** QuickPart:

Rectangular


If the built-in enclosure is not suitable, you can define a Custom enclosure whose geometry is based on a Construction Geometry that you specify.

Once an enclosure is created, you can fit openings and fans in the enclosure walls, see Creating and Positioning the Components.

Enclosure—Properties

The following properties are common to Rectangular and Custom enclosures:

Name	Specifies the name of the enclosure under which it is stored in the tree.

Geometry	**Local Origin** Specifies the position of the enclosure's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. It is used as reference for placing the enclosure in the simulation domain. The following options are available: • **Bottom Center**: Center of the Zmin face. • **Bottom Corner 1**: Xmin/Ymin/Zmin corner of the enclosure. • **Bottom Corner 2**: Xmin/Ymax/Zmin corner of the enclosure. • **Center**: Center of the geometry. • **Custom**: This value appears when you set the local origin interactively within the *Graphics* window, see Placing Object Interactively.
Placement	See Placement properties.
Physics	**Material** Specifies the material of the enclosure. You can choose from the Solid Sim Materials that you define in the Setup panel. **Heat Source** Specifies the total heat that the enclosure generates.

The following geometry properties characterize the different enclosure types:

Rectangular—Specific Properties

Geometry	**X, Y, Z** Specifies the size of the enclosure in x-, y-, and z-direction, in the enclosure's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. **Thickness** Specifies the sheeting thickness. **Shell Outward** When activated, **X**, **Y**, and **Z** define the size of the enclosure's inner volume. The total size of the enclosure is increased in all directions by the specified **Thickness** value. **Calculate Size from Selected QuickParts** Allows you to create an enclosure that includes specific QuickParts. The extent of the block in the respective coordinate direction is calculated based on the maximum distance between two of the selected QuickParts in that coordinate direction. You can extend the size of the block by adding an offset in negative and positive coordinate direction.

Custom—Specific Properties

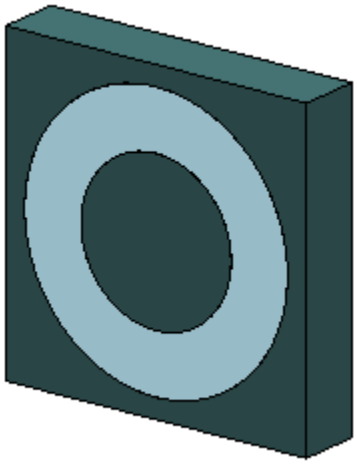
Geometry	**Selected Construction Geometry** Specifies the Construction Geometry that describes the geometry of the enclosure. If you select multiple Construction Geometries when creating a Custom enclosure, the Electronics Cooling Toolset automatically creates multiple enclosures that own the same physics properties.

Axial Fan

Fans cool electronic devices by drawing cool air over the components' heat sinks. In axial fans, the gas flow is parallel to the fan's axis. They generally produce low pressures with high flow rates.

The Electronics Cooling Toolset supports an **Axial Fan** QuickPart that consists of a fan hub, a housing, and a fan volume. The fan volume represents the volume swept out by the fan blades and does not consider the individual blades of the fan.

The geometry of most axial fans can be simplified to follow the available templated fan type:

Templated


If the built-in axial fan is not suitable, you can define a Custom axial fan whose geometry is based on Construction Geometries that you specify.

An Axial Fan QuickPart imposes a pressure jump upon the flow that passes through the fan volume. The pressure rise depends on the fan performance curve that you specify. A fan performance curve describes the pressure rise across the fan—that is the static pressure measured downstream of the fan minus the total pressure that is measured upstream of the fan—as a function of the volumetric flow rate.

An optional fan hub heat source allows you to model the heat that is generated by the electric motor of the fan.

Note: The material of the fan volume is the Default Gas material that you specify in the Setup panel. Axial Fan QuickParts are currently not supported for liquid domains.

Axial Fan—Properties

The following properties are common to Templated and Custom axial fans:

Name	Specifies the name of the heat sink under which it is stored in the tree.
Geometry	**Local Origin** Specifies the position of the axial fan's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. It is used for placing the fan in the simulation domain. The following options are available: • **A**: Center of front (Zmax) face. • **B**: Center of the fan geometry. • **C**: Center of back (Zmin) face. • **Custom**: This value appears when you set the local origin interactively within the *Graphics* window, see Placing Object Interactively. **Note:** For a Custom axial fan, the local origin of the geometry is placed in the origin of the Laboratory coordinate system, by default.
Placement	See Placement properties.
Physics	**Housing Material, Hub Material** Specify the material of the fan's housing and hub, respectively. You can choose the materials from the list of Solid Sim Materials that you define in the Setup panel. **Hub Heat Source** Specifies the total heat generated by the electric motor that is integrated in the fan hub. **Fan Surface** Controls if the pressure jump is imposed on the **Front Surface** or on the **Back Surface** of the fan volume. To check the respective surface, see the origin of the black arrow within the *Graphics* window. **Reverse Direction** Allows you to reverse the direction of the fan flow to show from the specified **Fan Surface** into the fan volume. To check the flow direction, see the black arrow within the *Graphics* window. **Fan Curve Type** Controls the specification of the fan performance curve. The following options are available:

	• **Polynomial**: Defines the fan curve as a linear function. The pressure rise across the Fan Surface is calculated as $p(\dot{v}) = -\frac{p_{\max}}{\dot{v}_{\max}}\dot{v} + p_{\max}$ where $p_{\max}$ and $\dot{v}_{\max}$ are the specified **Maximum Pressure** and **Maximum Flow**, respectively. • **Table**: Defines the fan curve as a table of pressure rise versus volumetric flow rate as indicated by the imported **Fan Curve File** (`*.csv`). After the import, select the appropriate **Pressure Units** and **Volume Flow Units**. The pressure rise is calculated by linear interpolation between the data points in the table. Outside the intervals, the nearest limit is used. The specified fan performance curve must correspond to the actual fan operating rotation rate and temperature. If your data correspond to some standard rotation rate and temperature, you must modify the fan performance curve to suit your conditions using established fan laws: $\Delta P_2 = \Delta P_1 \times (\omega_2/\omega_1)^2 \times (T_1/T_2)$ $\psi_2 = \psi_1(\omega_2/\omega_1)$ where ω is the fan rotation rate (rpm), T is the fluid temperature (K), and ψ is the volumetric flow rate.

The following geometry properties characterize the different types of axial fans:

Templated—Specific Properties

Geometry	**Housing Width** Specifies the extent of the housing in x- and y-direction, in the axial fan's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. **Housing Depth** Specifies the length that the housing extends from back to front. **Fan Radius** Specifies the outer radius of the fan volume. **Hub Radius** Specifies the radius of the fan hub.

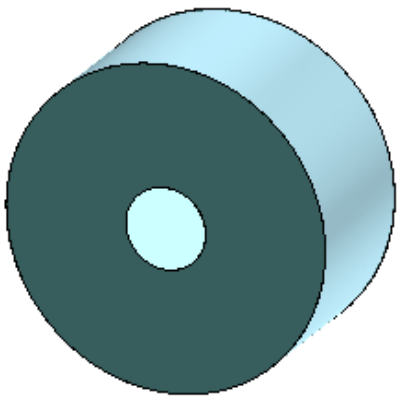
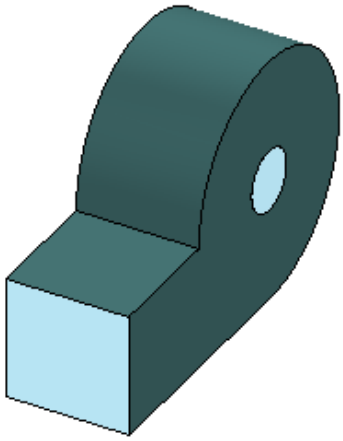
Custom—Specific Properties

Geometry	**Selected Housing, Selected Fluid, Selected Hub** Specify the Construction Geometries that describe the housing, the fan volume and the fan hub, respectively. **Select Front Faces, Select Back faces** Specify the front surface and the back surface of the **Selected Fluid**, respectively, that are required for the **Fan Surface** specification.

Radial Fan

Fans cool electronic devices by drawing cool air over the components' heat sinks. In radial fans, the gas flow is perpendicular to the fan's axis. They generally produce high pressures and low flow rates.

The Electronics Cooling Toolset supports two types of **Radial Fan** QuickParts:

Templated (for impeller fans)	TemplatedWithVolute (for centrifugal fans)
	

If none of the built-in radial fans are suitable, you can define a Custom radial fan whose geometry is based on a Construction Geometry that you specify.

A Radial Fan QuickPart physically separates the inflow and outflow surfaces in space—the volume within the fan is not modeled. Fluxes that cross the inflow surface are transformed and applied to the outflow surface. The pressure rise that the Radial Fan QuickPart imposes across the two surfaces of the contacting gas QuickPart depends on the fan performance curve that you specify. A fan performance curve describes the pressure rise across the fan—that is the static pressure measured downstream of the fan minus the total pressure that is measured upstream of the fan—as a function of the volumetric flow rate.

An optional heat source allows you to model the heat that is generated in operation of the fan.

Note: Radial Fan QuickParts are currently not supported for liquid domains.

Radial Fan—Properties

The following properties are common to all types of radial fans:

Name	Specifies the name of the heat sink under which it is stored in the tree.
Geometry	**Local Origin** Specifies the position of the radial fan's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. It is used for placing the fan in the simulation domain. The following options are available: • **A**: Center of the Zmin face. • **B**: Center of the fan geometry. • **C**: Center of the Zmax face. • **Custom**: This value appears when you set the local origin interactively within the *Graphics* window, see Placing Object Interactively.
Placement	See Placement properties.
Physics	**Blower Heat Generation Rate** Specifies the total heat generated by the fan. **Fan Curve Type** Controls the specification of the fan performance curve. The following options are available: • **Polynomial**: Defines the fan curve as a linear function. The pressure rise across the Fan Surface is calculated as $p(\dot{v}) = -\frac{p_{\max}}{\dot{v}_{\max}}\dot{v} + p_{\max}$ where $p_{\max}$ and $\dot{v}_{\max}$ are the specified **Maximum Pressure** and **Maximum Flow**, respectively. • **Table**: Defines the fan curve as a table of pressure rise versus volumetric flow rate as indicated by the imported **Fan Curve File** (`*.csv`). After the import, select the appropriate **Pressure Units** and **Volume Flow Units**. The pressure rise is calculated by linear interpolation between the data points in the table. Outside the intervals, the nearest limit is used. The specified fan performance curve must correspond to the actual fan operating rotation rate and temperature. If your data correspond to some standard rotation rate and temperature, you must modify the fan performance curve to suit your conditions using established fan laws: $\Delta P_2 = \Delta P_1 \times (\omega_2/\omega_1)^2 \times (T_1/T_2)$ $\psi_2 = \psi_1(\omega_2/\omega_1)$

	where ω is the fan rotation rate (rpm), T is the fluid temperature (K), and ψ is the volumetric flow rate.

The following properties characterize the different types of radial fans:

Templated—Specific Properties

Geometry	**Fan Radius** Specifies the radius of the fan volume. **Inlet Radius** Specifies the radius of the inflow surface. **Width** Specifies the extent of the fan volume in z-direction, in the radial fan's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. **Inlet Type** Specifies the location of the inflow surface. The following options are available: • **Front**: Zmax side of the fan. • **Back**: Zmin side of the fan. • **Dual**: Each side of the fan (Zmin and Zmax).

TemplatedWithVolute—Specific Properties

Geometry	**Fan Radius, Inlet Radius, Width, Inlet Type** As for Templated radial fan. **Volute Length** Specifies the length of the volute. **Outlet Width** Specifies the width of the outflow surface.

Custom—Specific Properties

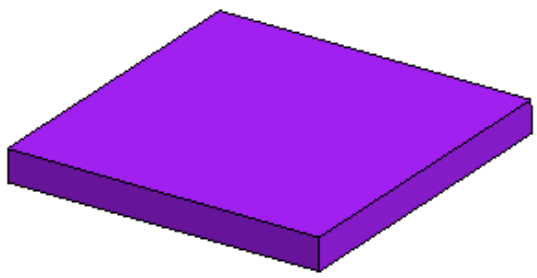
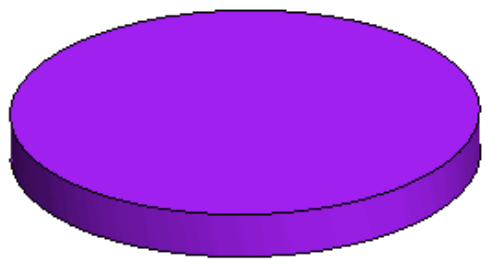
Geometry	**Selected Construction Geometry** Specifies the Construction Geometry that describes the fan volume. **Select Inlet Faces, Select Outlet faces** Specify the inflow surfaces and the outflow surfaces of the **Selected Construction Geometry** to determine the flow direction through the fan.

Resistance

A Resistance describes any electronic component that permits the passage of gas and generates heat.

The **Resistance** QuickPart does not resolve the internal structure of the component but uses a Porous Medium approach to describe the effect of the component on the overall flow and heat transfer. With this approach, you can model complex power supplies without considering geometric details. It is particularly useful for modeling PCB racks.

The Electronics Cooling Toolset supports the following simple shape templates:

Block	Cylinder
	

If none of the built-in shape templates are suitable, you can define a Custom resistance whose geometry is based on a Construction Geometry that you specify.

Note: The fluid that is contained in a Resistance QuickPart is the Default Gas material that you specify in the Setup panel. Resistance QuickParts are currently not supported in liquid domains.

Resistance—Properties

The following properties are common to all resistance types:

Name	Specifies the name of the resistance under which it is stored in the tree.
Geometry	**Local Origin** Specifies the position of the resistance's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. It is used for placing the resistance in the simulation domain. The following options are available: • **Bottom Center**: Center of the Zmin face. • **Bottom Corner 1** (block resistance only): Xmin/Ymin/Zmin corner of the block. • **Bottom Corner 2** (block resistance only): Xmin/Ymax/Zmin corner of the block.

	• **Center**: Center of the geometry. • **Custom**: This value appears when you set the local origin interactively within the Graphics window, see Placing Object Interactively.
Placement	See Placement properties.
Physics	Specifies the properties of a porous medium that is used to mimic the effect of the actual component on the overall flow and heat transfer. The **Porous Inertial Resistance**, the **Porous Viscous Resistance**, and the **Solid Thermal Conductivity** require the input of an **XX Component**, a **YY Component**, and a **ZZ Component** that represent the diagonal components of a Principal Tensor Profile in the local coordinate system of the resistance. See X_or, Y_or, Z_or in the *Graphics* window. **Porous Inertial Resistance, Porous Viscous Resistance** Specifies the inertial and porous resistance of modeled component, respectively. For more detail, see Superficial Velocity Formulation. **Porosity** Specifies the ratio of open volume to total volume within the modeled component. **Solid Thermal Conductivity** Specifies the thermal conductivity of the modeled component. For more detail, see Solid Thermal Conductivity. **Heat Source** Specifies the total heat that the modeled component generates.

The following properties characterize the various resistance types:

Block—Specific Properties

Geometry	**X, Y, Z** Specifies the total size of the block in x-, y- and z-direction, in the resistance's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window.

Cylinder—Specific Properties

Geometry	**Radius** Specifies the radius of the cylinder. **Height** Specifies the extent of the cylinder from bottom to top.

Custom—Specific Properties

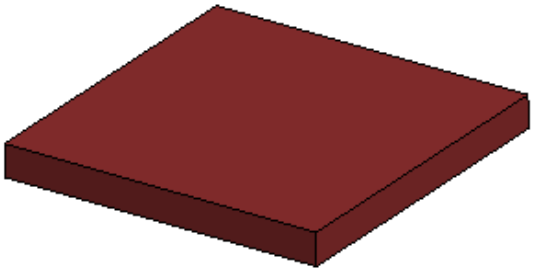
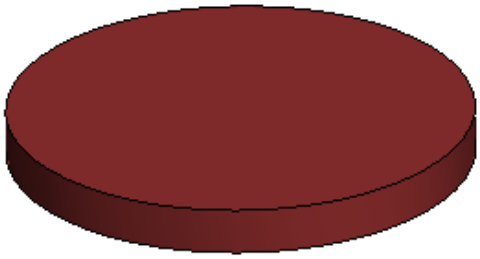
Geometry	**Selected Construction Geometry** Specifies the Construction Geometry that describes the resistance. If you select multiple Construction Geometries when creating a Custom resistance, the Electronics Cooling Toolset automatically creates multiple Custom resistance QuickParts that own the same physics.

Generic

A generic QuickPart describes any component that does not correlate to one of the predetermined QuickPart categories.

A **Generic** QuickPart can be of solid, gas, or liquid material and allows for the specification of a heat source.

The Electronics Cooling Toolset supports the following simple shape templates:

Block	Cylinder
	

If none of the built-in generics are suitable, you can define a Custom generic whose geometry is based on a Construction Geometry that you specify.

If you have IDF files that contain board and its components' layout definitions, you can import the files (see Import IFD file action). The Electronics Cooling Toolset automatically creates a PCB QuickPart for the board and Generic QuickParts for the components.

Generic—Properties

The following properties are common to all types of generics:

Name	Specifies the name of the generic under which it is stored in the tree.

Geometry	**Local Origin** Specifies the position of the generic's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. It is used for placing the generic in the simulation domain. The following options are available: • **Bottom Center**: Center of the Zmin face. • **Bottom Center of Range Box** (IDF import only): Center of the bottom face of the bounding box. • **Bottom Corner 1** (block generic only): Xmin/Ymin/Zmin corner of the block. • **Bottom Corner 2** (block generic only): Xmin/Ymax/Zmin corner of the block. • **Center**: Center of the geometry. • **Center of Range Box** (IDF import only): Center of the box that bounds the imported geometry. • **Custom**: This value appears when you set the local origin interactively within the Graphics window, see Placing Object Interactively. • **IDF Default Reference** (IDF import only): As specified in the IDF file.
Placement	See Placement properties.
Physics	**Heat Source** Specifies the total heat that the modeled component generates. **Material** Specifies the material of the modeled component. The following options are available: • **Solid**: Allows you to choose a material from the Solid Sim Materials that you define in the Setup panel. If the Thermal Conductivity property of the respective material uses the Anisotropic method, you set the components of the **Thermal Conductivity** tensor using the principal tensor method. See Setting Values Using a Principal Tensor. The **XX Component**, **YY Component**, and **ZZ Component** are defined with respect to the axis of the QuickPart's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. • **Gas**: Sets the Default Gas material. • **Liquid**: Sets the Default Liquid Material.

The following geometry properties characterize the various types of generics:

Block—Specific Properties

Geometry	**X, Y, Z** Specify the dimensions in x-, y-, and z-direction, in the generic's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window.

Cylinder—Specific Properties

Geometry	**Radius** Specifies the radius of the cylinder. **Height** Specifies the extent of the cylinder from bottom to top.

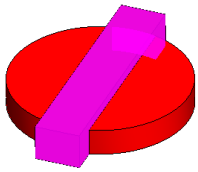
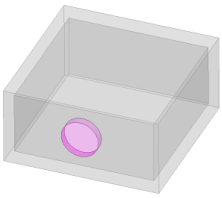
Custom—Specific Properties

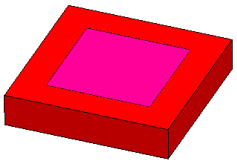
Geometry	**Selected Construction Geometry** Specifies the Construction Geometry that describes the geometry of the generic. If you select multiple Construction Geometries when creating a Custom generic, the Electronics Cooling Toolset automatically creates multiple generics that own the same physics properties.

Interactions

Within the Electronics Cooling Toolset, an interaction allows you to remove volume from a QuickPart, to cut holes through QuickPart segments, or to split a QuickPart Surface.

The following properties define the behaviour of an interaction:

Interaction		Example
Subtract	Allows you to remove the volume of one or more objects from the selected QuickPart. The subtracted objects are retained. Each object must share some of its volume with the QuickPart, otherwise the QuickPart remains unchanged. **Solid Quick Objects** Specifies the objects that remove material from the QuickPart. You can choose from QuickParts and Construction Geometries except for Sheet Construction Geometries.	
Extrude Cut Through First Segment	Allows you to project one or more Sheet Construction Geometries in normal direction to cut holes through the selected QuickPart. The three-dimensional objects that are formed by projecting the sheet objects are subtracted from the first segment of the QuickPart. This is particularly useful for cutting openings in the enclosure of an electronic device.	

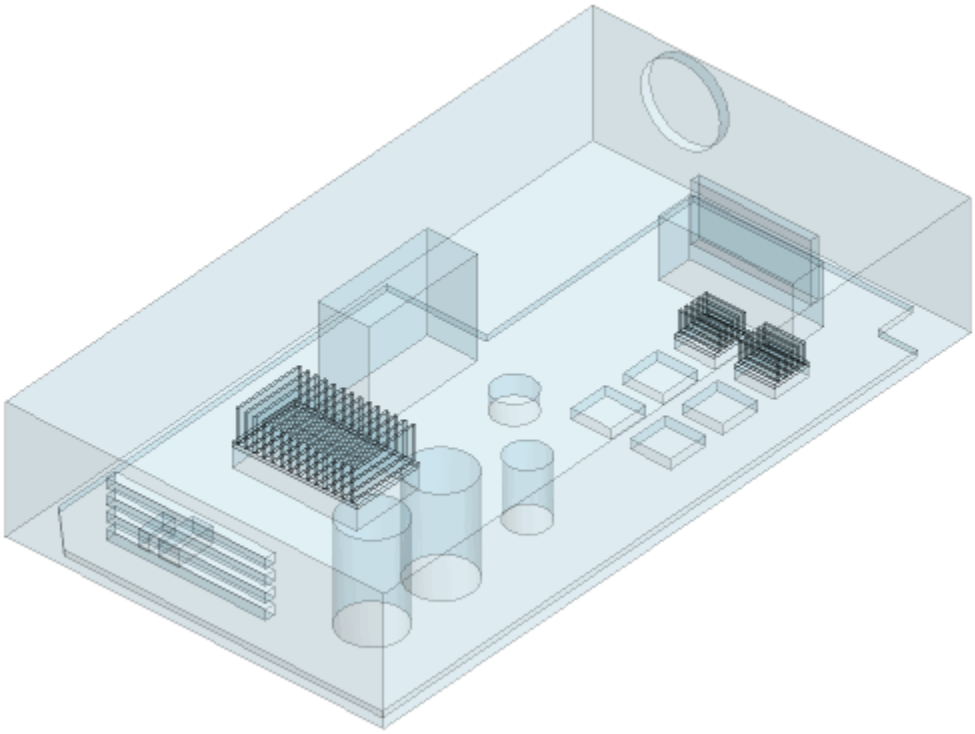
Interaction		Example
	Sheet Construction Geometry Specifies the geometries that determine the shape and position of the holes. You can choose from any Sheet Construction Geometry or pattern of Sheet Construction Geometries.	
Imprint	Allows you to imprint the profile of one or more Sheet Construction Geometries onto the surface of the selected QuickPart. The surface is split where the new lines are added. The QuickPart and the Sheet Construction Geometries must overlap by less than `1.0E-8 m`, or have coincident faces, otherwise the QuickPart remains unchanged. This property enables you to create new QuickPart Surfaces on which you can specify custom mesh settings or physics conditions. **Sheet Construction Geometry** Specifies the objects that imprint their profile. You can choose from any Sheet Construction Geometry or pattern of Sheet Construction Geometries.	

Extract Volume

An **Extracted Volume** part describes the domain that is enclosed within selected QuickParts or Construction Geometries.

The process of finding the enclosed volume can be thought of as capping all openings in the model, and filling the void in between. You identify the location of the openings by selecting specific faces and/or edges. This action is useful for creating the fluid domains for the flow analysis.

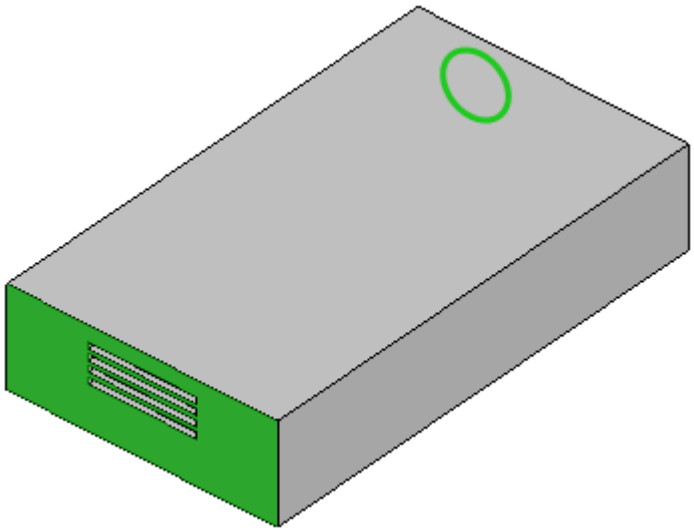
Extracted Volume



Extracted Volume—Properties

The following properties describe an extracted volume:

Geometry	Defines the bounds that enclose the volume. **Extract Volume Domain Defined By:** Controls if the volume of interest is bounded by a **Construction Geometry** or by **QuickParts**. The **Construction Geometry** or **QuickParts** that you specify describe the bounding geometry—the extraction procedure automatically recognizes the components that are located within the volume. For QuickParts that contain entry and exit holes—like open enclosures—you specify **Inlet/Outlet Edges or Faces** that you select interactively in the *Graphics* window. When you select edges, you specify the exact locations of the holes. When you select faces, the Electronics Cooling Module analyzes it for the presence of any holes. The following example displays a valid edge/face selection for extracting the volume within and open enclosure:

	 The holes must meet the following criteria in order to carry out the volume extraction successfully: • The edge loop of each hole must lie on a single face, or cross a maximum of two connected faces. • The edges of each hole must lie on the same plane. The extraction process fails if material can leak out of the model through any unselected holes.
Physics	Defines the material of the extracted volume. **Gas, Solid, Liquid** Controls if the modeled component is of gaseous, solid, or liquid material. Selecting the Solid option allows you to choose a **Material** from the Solid Sim Materials that you define in the Setup panel. The material of a gaseous or liquid domain corresponds to the Default Gas or Default Liquid material, respectively.

Extracted Volume—Right-Click Actions

The following specific actions are available when you right-click an extracted volume QuickPart:

Suppress

Suppresses the execution of the volume extraction.

By default, the volume extraction is executed whenever you create or edit a QuickPart in the model of your electronics device. The computational costs for a volume extraction increases with the number of QuickParts in your model. Therefore, this action is useful to improve performance when creating or editing QuickParts in a model with many QuickParts.

Note: Suppressing an extracted volume can invalidate QuickParts that depend on the extracted volume, and cause warning dialogs to appear if the CAD engine fails to reconstruct the model.

Unsuppress

Reverses the **Suppress** action.

QuickPart Surfaces

Every QuickPart is composed of QuickPart Surfaces. By defining the boundary and interface types, and setting associated values, you can control how the QuickPart interacts with its surroundings.

A QuickPart can be in contact with the environment and/or in contact with other QuickParts. The Electronics Cooling Toolset allows you to set the physics conditions for either situation within a single QuickPart Surface:

- **Boundary Type**: Controls the default conditions for faces of the QuickPart Surface that are in contact with the open environment.
- **Interface Type**: Controls the default conditions for faces that are in contact with other QuickParts. Interfaces allow solution quantities like mass, momentum, or energy to pass from one QuickPart to another.

The Electronics Cooling Toolset provides several boundary types to cover various physical situations:

- Wall
- Velocity Inlet
- Stagnation Inlet
- Pressure Outlet

The following interface types are available depending on the state of the contacting QuickParts:

	Solid||Solid	Gas||Gas	Solid||Gas	Porous||Gas	Liquid||Gas
		Liquid||Liquid	Solid||Liquid		Liquid||Porous
		Porous||Porous	Solid||Porous		
Contact Interface	✔		✔		
Internal Interface	✔	✔		✔	
Baffle Interface		✔		✔	✔
Porous Baffle Interface		✔		✔	
Fan Interface		✔		✔	

Note: Most QuickParts have only one QuickPart Surface by default. To set different boundary types on specific surfaces of a QuickPart—like inflow, outflow, and wall conditions for the air domain—you must split the default surface. See Splitting a QuickPart Surface. When you delete a previously created QuickPart Surface from the tree, the corresponding faces are transferred back to the default surface.

The following properties characterize the different boundary and interface types:

Wall—Properties

A wall boundary represents a flow impermeable boundary. For more information, see Wall.

Conditions	**Thermal Specification** Controls the thermal conditions at the wall. The following options are available: • **Heat Source**: Applies a user-defined total heat source as indicated by the specified **Heat Flux** value. • **Temperature**: Sets the boundary temperature to the specified **Static Temperature** value. • **Convection**: Permits convection heat transfer on the outside of the domain. To account for ambient conditions, you specify **Ambient Temperature** and **Heat Transfer Coefficient** values. • **Environment**: Permits both convection and radiation heat transfer between the external side of the QuickPart and the environment. You specify values for the **Ambient Temperature**, the **Heat Transfer Coefficient**, and the **Thermal Resistance**. The emissivity on the external side of the QuickPart corresponds to the Default Surface Emissivity that you define in the Setup panel. For more information, see Setting an Environment Boundary Condition for Heat Transfer **Wall Surface Specification (only for a turbulent flow regime)** Controls the finish of the surface. The following options are available: • **Smooth**: Regular wall treatment for an even wall. • **Rough**: Modifies the wall treatment to incorporate roughness as indicated by the specified **Roughness Height**. You obtain the value of the equivalent sand-grain roughness height that is appropriate for your model either from the literature or empirically. To determine from an experiment: • Measure the friction factor for the Reynolds number of your case in the experiment. • Plot your measurements on a Moody diagram. • Deduce the equivalent sand-grain roughness height from the Moody diagram curve. For more information, see Wall Treatment for Rough Walls.
Values	**Surface Emissivity (only if radiation is included)** Specifies the ratio of the power that the solid QuickPart emits to the power it would emit as a black body at the same temperature. Additional values are available depending on the conditions that you specify (see above).

Velocity Inlet—Properties

A velocity inlet boundary represents an inflow boundary that allows you to set a specific velocity magnitude. The flow direction is normal to the boundary face. For more information, see Velocity Inlet.

Surface Emissivity (only if radiation is included)

As for Wall boundary.

Static Temperature

Specifies the static temperature of the inflowing fluid.

Velocity Magnitude

Specifies the magnitude of the inflow velocity. The flow direction is normal to the boundary face.

Stagnation Inlet—Properties

A stagnation inlet boundary represents an inflow boundary that enables you to set the conditions of an imaginary plenum, far upstream, in which the flow is completely at rest. For more information, see Stagnation Inlet.

Conditions	**Stagnation Inlet Option** Controls the pressure conditions at the inlet. The following options are available: • **None**: Applies the default conditions. • **Pressure Jump**: Imposes a pressure jump on the inlet. The available **Pressure Jump Options** are: ◦ **None**: Does not apply a pressure jump. ◦ **Fan**: Imposes a pressure jump that is obtained from a fan performance curve that you specify. A fan performance curve describes the pressure rise across the fan—that is the static pressure measured downstream of the fan minus the total pressure that is measured upstream of the fan—as a function of the volumetric flow rate. The following **Fan Curve Type** options are available: ▪ **Polynomial**: Defines the fan curve as a linear function. The pressure rise across the boundary is calculated as $p(\dot{v}) = -\frac{p_{\max}}{\dot{v}_{\max}}\dot{v} + p_{\max}$ where $p_{\max}$ and $\dot{v}_{\max}$ are the specified **Maximum Pressure** and **Maximum Flow**, respectively. ▪ **Table**: Defines the fan curve as a table of pressure rise versus volumetric flow rate as indicated by the imported **Fan Curve File** (`*.csv`). After the import, select the appropriate **Pressure Units** and **Volume Flow Units**. The pressure rise is calculated by linear interpolation between the data points in the table. Outside the intervals, the nearest limit is used.

	The specified fan performance curve must correspond to the actual fan operating rotation rate and temperature. If your data correspond to some standard rotation rate and temperature, scale the fan performance curve as required using the following established fan laws: $$\Delta P_2 = \Delta P_1 \times \left(\omega_2/\omega_1\right)^2 \times \left(T_1/T_2\right)$$ $$\psi_2 = \psi_1\left(\omega_2/\omega_1\right)$$ where ω is the fan rotation rate (rpm), T is the fluid temperature (K), and ψ is the volumetric flow rate. ◦ **Loss Coefficient**: Computes the pressure loss as $0.5K\rho\|v\|^2$ where K is the specified **Pressure Loss Coefficient** value.
Values	**Total Pressure** Specifies the total pressure upstream of the simulation domain. **Surface Emissivity (only if radiation is included)** As for Wall boundary. **Total Temperature** Specifies the total temperature of the upstream plenum. Additional values are available depending on the conditions that you specify (see above).

Pressure Outlet—Properties

A pressure outlet boundary represents an outflow boundary that requires the specification of the working pressure across the outlet. For more information, see Pressure Outlet.

Conditions	**Pressure Outlet Option** Controls the pressure conditions at the outlet. The following options are available: • **None**: Applies the default conditions. • **Pressure Jump**: Imposes a pressure jump on the outlet. The available **Pressure Jump Options** are: ◦ **None**: Does not apply a pressure jump on the boundary. ◦ **Fan**: as for **Stagnation Inlet** boundary.
Values	**Surface Emissivity (only if radiation is included)** As for Wall boundary. **Static Temperature** Specifies the temperature of the fluid in the case of backflow into the domain. **Pressure** Specifies the working pressure.

	Additional values are available depending on the conditions that you specify (see above).

Contact Interface—Properties

A contact interface joins two solid QuickParts or a solid and a fluid QuickPart and permits conjugate heat transfer between them.

Conditions	**Thermal Specification** Controls the thermal conditions at the interface. The following options are available: • **Conjugate Heat Transfer**: The temperature at the interface is determined from the heat that is transferred through the interface. Optionally, you can specify a **Contact Resistance**, that is the resistance to conduction through the interface. The relation between the heat flux across the contact interface, the contact resistance R, and the temperature difference ΔT is given by $\dot{q}'' = \frac{\Delta T}{R}$. Furthermore, you can provide an energy source for the interface. The specified **Heat Source** value defines the total rate of heat transfer through the interface. • **Specified Temperature**: The temperature at the interface is the same for both parent surfaces as indicated by the specified **Static Temperature** value. **Wall Surface Specification (only for a turbulent flow regime)** As for Wall boundary.
Values	**Surface Emissivity (only if radiation is included)** As for Wall boundary. Additional values are available depending on the conditions that you specify (see above).

Internal Interface—Properties

An internal interface joins two QuickParts of the same material together and allows the transfer of flow and/or energy. You do not set any conditions or values.

Baffle Interface—Properties

A baffle interface represents one or more thin sheets of impermeable, conducting materials between fluid QuickParts. For more information, see Baffle Interface Modeling.

Conditions	**Wall Surface Specification (only for a turbulent flow regime)** As for Contact Interface.
Values	**Thermal Resistance** Specifies the resistance to conduction through the baffle. To calculate the thermal resistance R of a baffle, you can use the following formula:

	$\frac{1}{R} = \sum_i^n \frac{k_i}{\Delta x_i}$ where k_i and Δx_i are the conductivity and thickness, respectively, of each of the n modeled layers comprising the baffle. **Heat Source** Specifies the heat transferred through the baffle. **Surface Emissivity (only if radiation is included)** As for Wall boundary. Additional values are available depending on the conditions that you specify (see above).

Porous Baffle Interface—Properties

A porous baffle interface represents a porous membrane between two fluid QuickParts through which the fluid passes and experiences a pressure drop. Porous baffles can be used to model perforated plates, thin screens, and wire screens. For more information, see Porous Baffle Interface—Modeling.

Conditions	**Wall Surface Specification (only for a turbulent flow regime)** As for Contact Interface.
Values	**Thermal Resistance** As for Baffle Interface. **Heat Source** As for Baffle Interface. **Surface Emissivity (only if radiation is included)** As for Wall boundary. **Porous Inertial Resistance** Specifies the value of the coefficient α in Eqn. (81). **Porous Viscous Resistance** Specifies the value of the coefficient β in Eqn. (81).

Note: A porous baffle is a "lumped" model; when modeling a baffle, you determine the bulk effects that the baffle has on the flow. If you want to compute the flow coming out from the small openings within the baffle, you must model the actual geometry and resolve it with a fine mesh.

Fan Interface—Properties

A fan interface models an axial fan between two fluid QuickParts as a zero-thickness interface which imposes a pressure jump upon the passing flow. Modeling a fan interface requires the specification of a fan performance curve that corresponds to the physical fan that you are modeling.

Conditions	**Fan Curve Type** Controls the specification of the fan performance curve. The following options are available: • **Polynomial**: As for Stagnation Inlet boundary. • **Table**: As for Stagnation Inlet boundary.
Values	The values that are available depend on the conditions that you specify (see above).

Note: The fan interface is a simple model that trades accuracy for speed. To represent the three-dimensional volume of an axial fan, see Axial Fan QuickPart.

Placement

The Electronics Cooling Toolset offers various positioning capabilities for Construction Geometries and QuickParts.

To define the placement of a geometry, the following options are available:

- Specifying the **Placement** properties

  You define the location and orientation of the geometry with respect to a selected coordinate system.
- Using the **Positioning Tool**

  The Positioning Tool allows you to place geometries in reference to each other.

The following placement properties are common to all types of Construction Geometries and QuickParts:

Placement	**Location** Specifies the position vector (**X**, **Y**, **Z**) of the geometry's **Local Origin**, in the selected **Coordinate System**. The **Local Origin** is a *Geometry* property of the respective Construction Geometry or QuickPart. **Note:** For a Custom QuickPart, the local origin corresponds to the local origin of the underlying Construction Geometry, by default. You can change the local origin of a Construction Object or QuickPart interactively within the *Graphics* window, see Placing Objects Interactively. **Rotation Axis Vector** Specifies the direction vector (**X**, **Y**, **Z**) of the axis about which you rotate the geometry, in the selected **Coordinate System**. The axis position, or the origin of the axis, is the geometry's **Local Origin**.

	Angle Specifies the angle of rotation. **Coordinate System** Specifies the reference coordinate system for the geometry's **Location** and **Rotation Axis Vector**. See X_ref, Y_ref, Z_ref in the *Graphics* window. **Position** Launches the Positioning Tool, that allows you to interactively place the geometry in reference to other geometries.

Contents:

Positioning Tool

Positioning Tool

The **Positioning Tool** allows you to place Construction Geometries and QuickParts in reference to each other.

The following options are available:

- Align—turns and moves an object until it is aligned with another object.
- Rotate—turns an object around a specified axis.
- Translate—moves an object from one position to another.

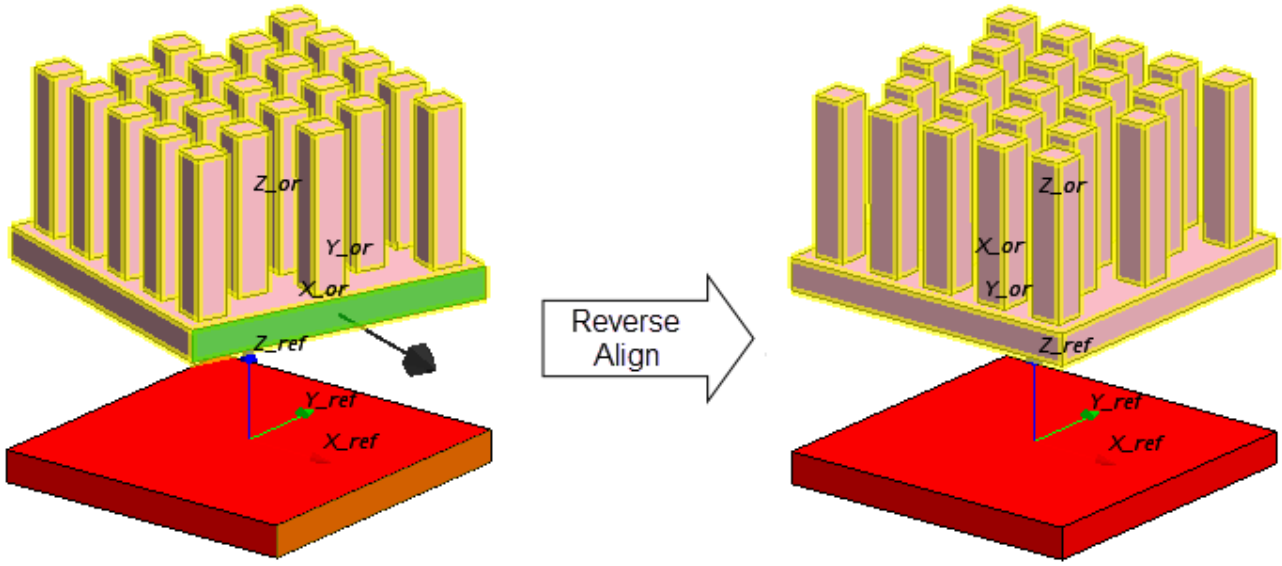
The following properties characterize the different positioning options:

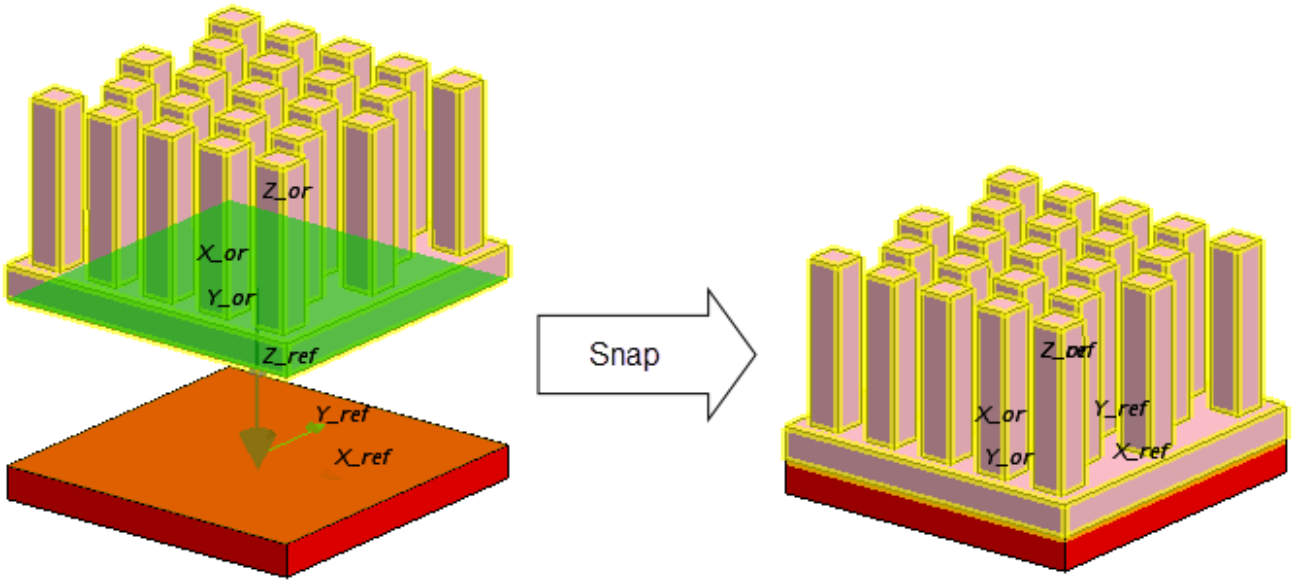
Align—Properties

Target Objects	Specifies the Construction Geometries or QuickParts you want to align.
Align By	Controls the alignment mode. The following options are available: • **Face, Edge, Vertex** Aligns the **Target Objects** with a geometric entity—face, edge, or vertex—of another QuickPart. **Reference**: Specifies a geometric entity of the reference QuickPart. **Target**: Specifies a geometric entity of the **Target Objects**. You select the reference and target entities interactively in the *Graphics* window (highlighted in ochre and green, respectively). After the selection, the calculated **Angle** between the selected entities is displayed. **Align**: Rotates the Target Objects about its local origin (see X_ref, Y_ref, Z_ref in the *Graphics* window) to end up in a position where the selected geometric entities are parallel to each other. The vector normals of the selected entities point in the same direction. **Reverse Align**: As **Align** with the vector normals of the selected entities pointing in opposite directions.

	For aligned geometric entities, **Snap** translates the Target Objects to end up in a position where the selected geometric entities are in contact. **Distance** displays the magnitude of the translation vector. • **X-Axis**, or **Y-Axis**, or **Z-Axis**: Aligns the Target Objects with the x-, y-, or z-axis of a **Coordinate System** that you specify. **Set Reference Axis Face**: Specifies a face of the **Target Objects**. **Coordinate System**: Specifies the laboratory coordinate system or a local coordinate system of another QuickPart. The Target Objects are rotated about its local origin (see X_ref, Y_ref, Z_ref in the *Graphics* window) to end up in a position where the respective axis is normal to the selected face.

Example: Positioning of a heat sink on a block-shaped chip.



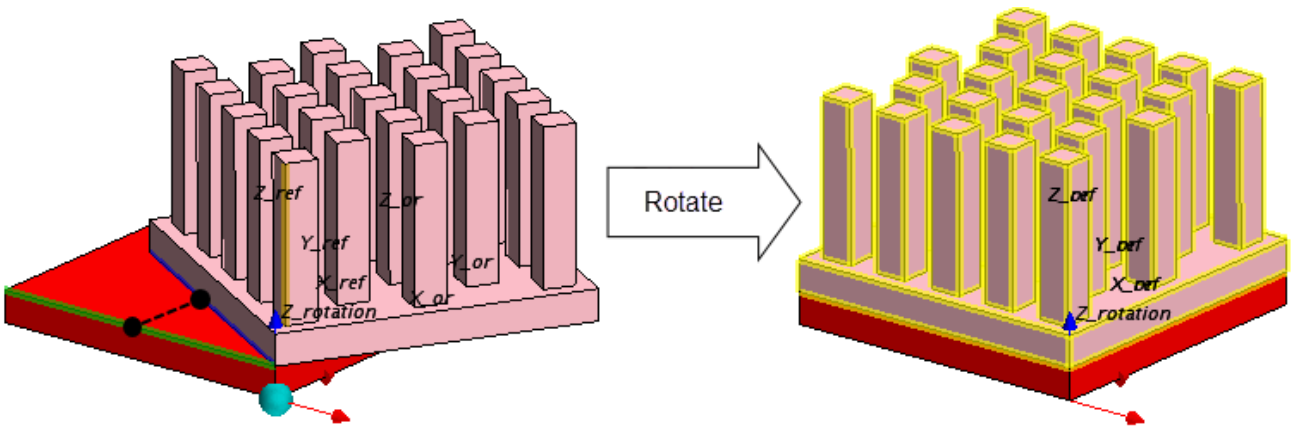


Rotate—Properties

Target Objects	Allows you to select the Construction Geometries or QuickParts you want to rotate.
Rotation Vector	Defines the direction vector of the axis of rotation. The following options are available: • Enter the **X**, **Y**, and **Z** components of the direction vector, in the selected **Coordinate System**. • Determine the direction vector interactively within the *Graphics* window by selecting one of the following entities (highlighted in orange): ◦ `Edge`: sets the rotation vector collinear to the edge. ◦ `Face`: sets the rotation vector equal to the face normal.
Rotation Position	Defines the origin of the axis of rotation. The following options are available: • Enter the **X**, **Y**, and **Z** components of the origin, in the selected **Coordinate System**. • Determine the origin interactively within the *Graphics* window by selecting one of the following entities (highlighted in pale blue): ◦ `Face`: sets the origin to the center of the face. ◦ `Edge`: sets the origin to the midpoint of the edge. ◦ `Vertex`: sets the origin to the vertex location.
Angle	Specifies the angle of rotation. The following options are available: • Enter the angle value directly. • Determine the angle interactively within the *Graphics* window. The rotation angle is calculated **From** one point **To** another, where the point locations depend on the selected entities (highlighted in dark blue and green, respectively): ◦ `Face`: sets the point to the center of the face.

	◦ `Edge`: sets the point to the midpoint of the edge. ◦ `Vertex`: sets the point to the vertex location.
Coordinate System	Defines the coordinate system for the rotation. You can define the rotation in the **Laboratory** coordinate system or in the local coordinate systems of another QuickPart. See X_ref, Y_ref, Z_ref in the *Graphics* window.

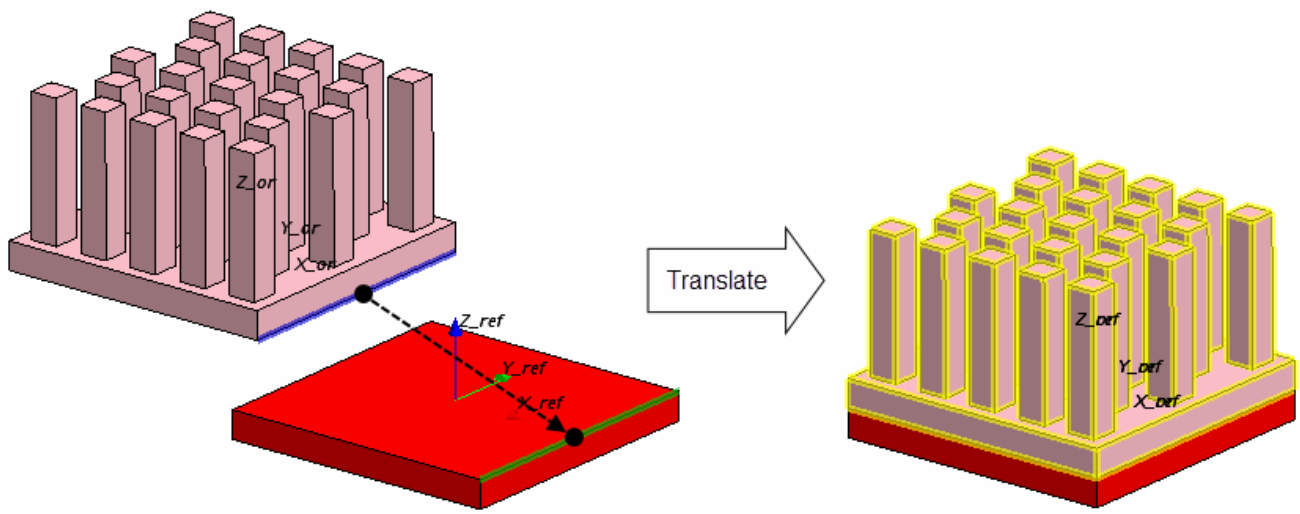
Example: Positioning of a heat sink on a block-shaped chip.



Translate—Properties

Target Objects	Allows you to select the Construction Geometries or QuickParts you want to translate.
Translation Vector	Defines the direction and distance of translation. The following options are available: • Enter the **X**, **Y**, and **Z** components of the translation vector, in the selected **Coordinate System**. • Determine the translation vector interactively within the *Graphics* window. The direction and distance is calculated **From** one point **To** another, where the point locations depend on the selected entities (highlighted in dark blue and green, respectively): ◦ `Face`: sets the point to the center of the face. ◦ `Edge`: sets the point to the midpoint of the edge. ◦ `Vertex`: sets the point to the vertex location.
Coordinate System	Defines the coordinate system for the translation. You can define the translation in the **Laboratory** coordinate system or in the local coordinate systems of another QuickPart. See X_ref, Y_ref, Z_ref in the *Graphics* window.

Example: Positioning of a heat sink on a block-shaped chip.



Interactive Right-Click Actions

The Electronics Cooling Toolset offers various interactive dimensioning and positioning capabilities for Construction Geometries and QuickParts.

The following right-click actions are available for the different geometric entities that are displayed in the *Graphics* window:

Selected Geometric Entity		Common Right-Click Actions
None		**Restore Hidden Quick Objects** Restores all previously hidden Construction Geometries and/or QuickParts. **Restore Hidden Quick CAD Faces** Restores all previously hidden faces of Construction Geometries and/or QuickParts. **Hardcopy...** Allows you to save the visualization display to an image file. For more information, see Hardcopy Properties.
Vertex		**Set Origin Reference At Vertex** Sets the local origin of the object that owns the vertex to the location of the selected vertex. The value of the object's **Local Origin** is set to **Custom**. The components of the **Location Vector** are updated to reflect the vertex position.
Edge	Linear	**Set Origin Reference To Center Of Linear Edge** Sets the local origin of the object that owns the edge to the center of the selected linear edge. The value of the object's **Local Origin** is set to

Selected Geometric Entity		Common Right-Click Actions
		Custom. The components of the **Location Vector** are updated to reflect the midpoint position.
	Curved	**Set Origin Reference To Center Of Curve** Sets the local origin of the object that owns the circular curve to the center of the circle. The value of the object's **Local Origin** is set to **Custom**. The components of the **Location Vector** are updated to reflect the position.
Face	Planar	**Set Origin Reference To Center Of Face** Sets the local origin of the object that owns the face to the center of the face. The value of the object's **Local Origin** is set to **Custom**. The components of the **Location Vector** are updated to reflect the position. **Create Face Coordinate System** Creates a new local coordinate system, where the origin of the coordinate system is centered on the selected face. Local coordinate systems are useful to place Construction Geometry or QuickPart in reference to each other. **Rename Face Coordinate System** Allows you to rename a previously created face coordinate system. **Hide** Hides the selected face from the visualization display. This action is useful for gaining a clear view of other objects. You can restore hidden objects by right-clicking in the empty background of the visualization display and selecting **Restore Hidden Quick CAD Faces**.
		New QuickPart > [Category] > [Type] Creates a new QuickPart of the respective category and type on the selected face. For information on the different categories and types of QuickPart and their properties, see QuickParts. **New Construction Geometry > [Type]** Creates a new templated Construction Geometry of the respective type on the selected face. For information on the different types of Construction Geometries and their properties, see Construction Geometries Shape Templates. **Note:** The position of the mouse pointer when selecting the face determines the position of the local origin of the new QuickPart or Construction Geometry, respectively.

Selected Geometric Entity		Common Right-Click Actions
	`Curved`	**Hide** As for `Planar Face`.
`Any`		**Rename** Allows you to change the name of the respective Construction Geometry or QuickPart under which it is stored in the tree. **Edit** Allows you to change the geometrical and physical properties of the respective QuickPart or Construction Geometry. **Hide** Hides the respective Construction Geometry or QuickPart from the visualization display. This action is useful for gaining a clear view of other objects. You can restore a specific hidden object by right-clicking on the respective entry in the tree and selecting **Show**. To restore all hidden objects click in the empty background of the visualization display and select **Restore Hidden Quick Objects**. **Position...** Launches the Positioning Tool that allows you to place the respective Construction Geometry or QuickPart in reference to other objects. **Paste QuickParts** Pastes the copied QuickParts from the clipboard. The QuickParts are pasted at the selected location in the *Graphics* window, replacing the selected item. If you select this action when you right-click the **QuickParts** manager node, the QuickParts are pasted in the tree and appear in the *Graphics* window at the origin location (0,0,0). In both cases, the relative orientations and distances between the QuickParts are retained as in the original, and you can then move the QuickParts to the appropriate locations.

Selected Geometric Entity	Specific Right-Click Actions for QuickParts
`Face`	**Edit** Allows you to change the properties of the respective QuickPart Surface. **Create QuickPart Surface** Creates a new QuickPart Surface that corresponds to the selected face. You can select several faces at once by holding down the <Ctrl> key when selecting the faces. The new QuickPart Surface is stored in the tree under the specified **Name**.

Selected Geometric Entity	Specific Right-Click Actions for QuickParts
	Create Open Enclosure Face Opens an enclosure by removing the wall that corresponds to the selected face. You can restore removed walls by right-clicking on any face of the respective enclosure and selecting **Restore All Open Enclosure Faces**. **Note:** This action is only available for rectangular enclosures.
Any	**Copy QuickParts** Copies the selected QuickParts to the clipboard. The **Paste QuickParts** action becomes available when you right-click the **QuickParts** manager node or right-click in the *Graphics* window.

Contents:

Interactive Right-Click Actions in Editing Mode

Interactive Right-Click Actions in Editing Mode

Additional right-click actions are available while editing a Construction Geometry or a QuickPart. They allow you to place and dimension the object being edited interactively and in reference to other objects.

The following actions are available for the various geometric entities in editing mode:

Selected Geometric Entity		Right-Click Actions
None		**Restore Hidden Quick CAD Faces** Restores all previously hidden faces of Construction Geometry and/or QuickParts. **Hardcopy** Allows you to save the *Graphics* window to an image file. For more information, see Hardcopy Properties.
Vertex		**Set Origin Reference At Vertex** Sets the local origin of the object being edited to the location of the selected vertex. The value of the object's **Local Origin** is set to **Custom**. The components of the **Location Vector** are updated to reflect the vertex position.
Edge	Linear	**Set Origin Reference To Center Of Linear Edge** Sets the local origin of the object being edited to the center of the selected linear edge. The value of the object's **Local Origin** is set to **Custom**. The components of the **Location Vector** are updated to reflect the midpoint position.

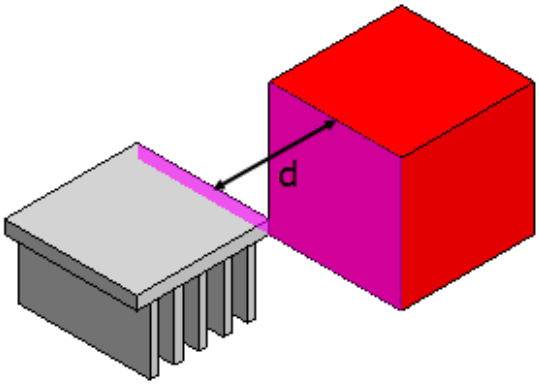
Selected Geometric Entity		Right-Click Actions
		Set Dimension > [Length Dimension] Allows you to set any length dimension of the object being edited to the length of the selected edge.
	`Curved`	**Set Origin Reference To Center Of Curve** Sets the local origin of the object that owns the circular curve to the center of the circle. The value of the object's **Local Origin** is set to **Custom**. The components of the **Location Vector** are updated to reflect the position.
`Face`	`Planar`	**Hide** Hides the selected face from the *Graphics* window. This action is useful for gaining a clear view of other objects. You can restore hidden objects again by right-clicking in the empty background of the scene and selecting **Restore Hidden Quick CAD Faces**. **Set Origin Reference To Center Of Face** Sets the local origin of the object being edited to the center of the planar face. The value of the object's **Local Origin** is set to **Custom**. The components of the **Location Vector** are updated to reflect the position. **Move Object Being Edited to Center of Face** Translates the object being edited. The translation vector is calculated as the vector pointing from the objects local origin to the center of the selected planar face.
	`Curved`	**Hide** As for `Planar Face`.
Any two of the following entities: • `Vertex` • `Linear Edge` • `Planar Face` **Note:** To select two geometric entities hold down the <Ctrl> key when selecting the entities.		**Hide** As for `Planar Face`. **Note:** This action is only available if one of the selected entities is a face. **Position > [Option]** See Positioning an Object. **Note:** This action is only available if one of the selected entities belongs to the object being edited. **Set Dimension > [Length Dimension]** Allows you to set any length dimension of the object being edited to the distance between the selected entities:

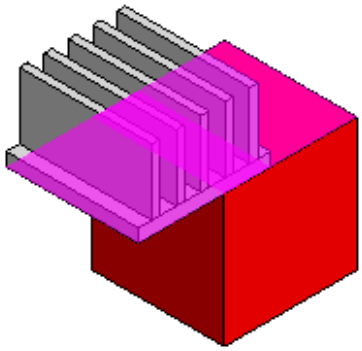
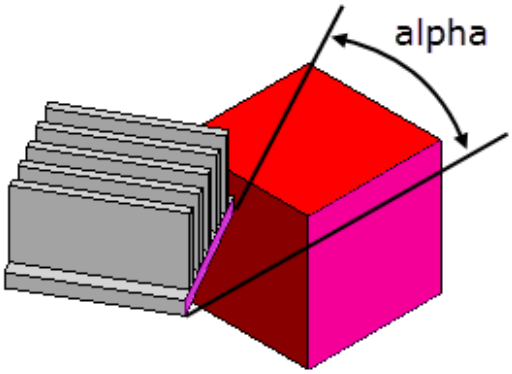
Selected Geometric Entity	Right-Click Actions
	• `Two Vertices`: distance between the vertices • `Any other two entities`: perpendicular distance between the entities **Note:** This action is only available for a selection of faces and/or edges if the faces and/or edges are parallel to each other.

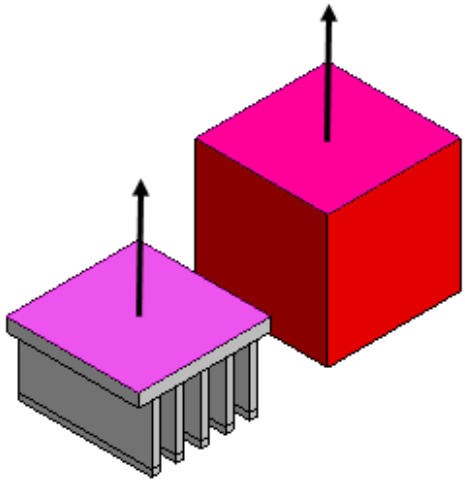
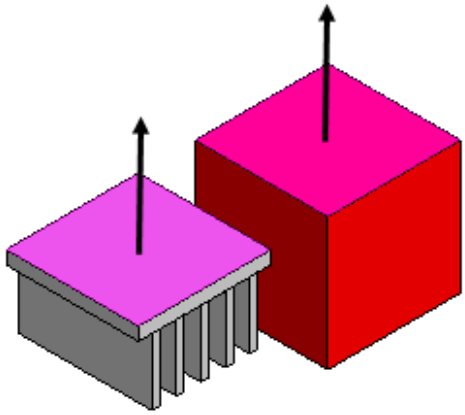
Positioning an Object

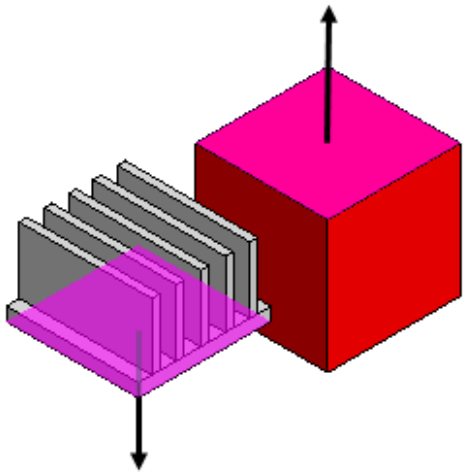
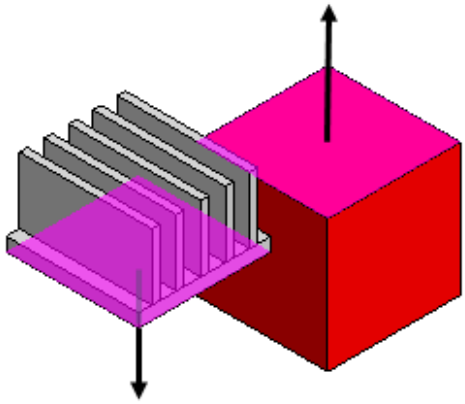
The **Position > [Option]** action allows you to position the object being edited in relation to another object.

The following options are available:

Options	Example
Note: For some combinations of `Vertex`, `Linear Edge` and `Planar Face` only a subset of options is available.	
Set Distance Translates the object being edited to end up in a position where the selected geometric entities are at a specified **Distance** (d) from each other.	
Snap Translates the object being edited to end up in a position where the selected geometric entities are in contact.	

Options	Example
	
Set Angle Rotates the object being edited about its local origin to end up in a position where the selected geometric entities form a specified **Angle** (alpha).	
Align Rotates the object being edited about its local origin to end up in a position where the selected geometric entities are parallel to each other. The vector normals of the selected entities point in the same direction.	

Options	Example
	
Align and Snap Combines **Align** and **Snap** in one action	
AntiAlign Rotates the object being edited about its local origin to end up in a position where the selected geometric entities are parallel to each other. The vector normals of the selected entities point in opposite directions.	

Options	Example
	
AntiAlign and Snap Combines **AntiAlign** and **Snap** in one action.	

Mesh

The **Mesh** manager node allows you to define and control the volumetric discretization of the computational domain.

The Electronics Cooling Toolset generates predominantly hexahedral cells in the volume mesh with prism shaped cells next to the walls. The prismatic cells are created in the fluid domain only and allow the solver to accurately capture near-wall flow and heat transfer. An integrated mesh refinement procedure automatically refines the mesh to capture complex features. The refinement takes place on surfaces that are in close proximity to one another, strongly curved, or relatively small when compared to the rest of the surfaces.

The Electronics Cooling Toolset provides three types of mesh settings:

- Global settings—apply to all QuickParts and QuickPart Surfaces.
- Surface Controls—are valid for specific QuickPart Surfaces and override the global settings.
- Volume Controls—are valid for the volume contained within a QuickPart or a Construction Geometry and override the global settings.

The Custom Control—Surface Control or Volume Control—that produces the smallest cell size takes precedence.

Mesh Settings

The following properties define the global characteristics of the mesh:

Automatically compute Base and Minimum Sizes	Controls the definition of the base size and the minimum cell size. By default the Electronics Cooling Toolset calculates these values automatically, where the calculation is based on the geometric scales of your electronics device. Deactivating this option allows you specify cell sizes directly.
Base Size	Specifies a characteristic dimension of the model. The minimum cell size and the fluid boundary layer thickness can be set as a percentage of this value. It also defines the target cell size on the surfaces.
Minimum Size	Specifies the lower limit of cell size. Decreasing this value allows more refinement in areas where the mesher is trying to capture complex features. However, an extensive refinement increases the CPU time and size of the mesh.
Mesh Refinement	Controls the refinement of the mesh in order to accurately capture complex geometrical features. **Surface Proximity** When activated, allows mesh refinement on surfaces that are close to one another. It determines the cell size as the distance between surfaces across a gap divided by the specified value. Increasing this value produces more cells in gaps. **Pts around Circle** When activated, allows mesh refinement on curved surfaces. The specified value determines the number of cells that are used to define a 360 degree cylindrical surface. Increasing the value increases the relative refinement of cells next to curved surfaces. For components that have a relatively small diameter when compared to the rest of the mesh region, decrease this value to avoid over-refinement. **Edge Proximity** When activated, allows mesh refinement on relatively small surfaces. It determines the cell size as the distance between opposite perimeter edges across a surface divided by the specified value.

	Increasing this value produces more cells on small surfaces. **Note:** If a QuickPart is composed of one QuickPart Surface only, applying edge proximity has no effect. You must split the QuickPart Surface first to get perimeter edges accordingly. See Splitting a QuickPart Surface.
Fluid Boundary Layer	Controls the prismatic mesh for the fluid boundary layer. **Total Thickness** Specifies the total overall thickness of the prism layer. **Number of Layers** Specifies the number of cell layers that are generated within the prism layer. **Stretching** Specifies the target growth rate of successive prism layers away from the wall.

Custom Mesh Settings

The following properties define the customized characteristics of the mesh:

Properties		Surface Control	Volume Control
Name	Specifies the name of the Custom Control under which it is stored in the tree.	✔	✔
Surfaces	Specifies the QuickPart Surfaces that define the surfaces of mesh customization.	✔	
Parts	Specifies the Construction Geometries that define the volumetric zone of mesh customization.		✔
Target Size	Specifies the target cell size in the volumetric zone of mesh customization.	✔	✔
Minimum Size	As for Mesh Settings.	✔	
Fluid Boundary Layer	As for Mesh Settings.	✔	✔

Options

The following options control the specification of various properties within the global and customized settings:

Percentage of Base	Specifies the reference value as percentage of the base size that is defined within the **Edit Mesh Settings** dialog. The advantage of a relative size specification is that changing the base size scales both global and object specific sizes.
Absolute Size	Specifies the reference value as absolute in any of the allowable length units.
Use Parent Value	Applies the global value to the mesh setting as defined in the **Edit Mesh Settings** dialog.
Specify Custom Value	Applies a customized value to the mesh setting.
Disable	Does not apply prismatic cell layers to the surface.

Mesh Right-Click Actions

The following action is available when you right-click the **Mesh** manager node:

Edit Mesh Settings

Allows you to specify the global Mesh Settings.

The following action is available when you right-click the **Mesh** > **Custom Controls** node:

New > [Type]

Creates a new custom control. The following types are available:

- Surface Control
- Volume Control

A separate *Edit* tab opens and allows you to set the Custom Mesh Settings of the respective custom control.

The following specific actions are available when you right-click a particular custom control:

Highlight

Highlights the respective surfaces or parts of the selected custom control in the *Graphics* window.

Scenes

The **Scenes** manager node allows you to visualize the geometry, mesh, and solution data that are contained in the simulation.

Scenes are the basic building blocks that build the visualization display. A pre-defined default scene displays the geometry of all Construction Geometries and QuickParts that you create.

A scene can display one of the following information:

- Geometry
- Simulation results—a scalar or a vector field

The respective information can be displayed on QuickParts Surfaces and/or on Drawables. Three different types of Drawables are available:

- Section—a plane which cuts through the solution domain.
- IsoSurface—a surface over which a scalar field has a constant value.
- Streamlines—curves that are tangent to a selected vector field.

Optionally, you can choose to display surface or volume mesh grids. Furthermore, Scene Annotations enable you to add customized text labels to the scene display.

The following properties define the characteristics of the pre-defined default scene:

Default Scene	
Name	Specifies the name of the scene under which it is stored in the tree.
Display Mesh	Controls the appearance of the mesh grid. Displaying the grid lets you examine the density and cell structure of various parts of the model, and thereby make judgments about accuracy. The following options are available: • **No Mesh**: Hides the grid except for the outlines. • **Surface Mesh**: Shows the surface grid. • **Volume Mesh**: Shows the volume grid.
Opacity	Specifies the transparency of the parts. The settings range from 0.0 (invisible) to 1.0 (opaque). This property allows to show an overlay of different parts of the model.

The following properties define the characteristics of a scene:

Scene	
Name	As for Default Scene.
Drawable Items	Specifies the items that are displayed in the scene. You can choose from **QuickPart Surfaces**, Drawables, and Scene Annotations.
Display Mesh	As for Default Scene.
Opacity	As for Default Scene.
Displayer Mode	Controls how the selected Drawables are displayed. The following options are available: • **Geometry**: Displays the outline and surface. • **Scalar**: Displays a scalar field using contours. This option activates the **Field Function**, **Color Map**, and **Contour Style** properties. • **Vector**: Displays glyphs that represent a specific vector field. This option activates the **Field Function** and **Color Map** properties.
Field Function	Specifies the scalar or vector field to show, such as temperature or velocity. See Field Functions. This property becomes available when you set the **Displayer Mode** to **Scalar** or **Vector**.

Scene	
Color Map	Specifies the colors to indicate values from lowest to highest. For the list of pre-defined choices, see Pre-Defined Colormaps. This property becomes available when you set the **Displayer Mode** to **Scalar** or **Vector**.
Contour Style	Controls how the scalar values, represented by colors, are separated from each other. This property becomes available when you set the **Displayer Mode** to **Scalar**. The following options are available: • **Automatic**: Automatically selects the most appropriate contour style, depending on the specified scalar function and parts. • **Filled**: Fills each cell face of the surfaces that are displayed with the scalar value stored at the center of its corresponding cell. Scenes using this style have a stepped appearance. • **Smooth Filled**: Displays vertex values rather than cell centroid-based values. The scalar field appears smoother when compared to the filled contour style. See Understanding Smoothed Values. • **Smooth Filled + Lines**: Adds lines to separate the colored values in the Smooth Filled style. • **Line**: Displays isolines of the vertex-based scalar quantity that is colored with the value of the scalar. • **Smooth Blended**: Displays vertex values, similar to the Smooth Filled option. However, it is less computationally expensive than the smooth filled option due to the technique used to blend colors across the cell surfaces.

The following action is available when you right-click the **Scene** manager node:

New

Creates a new scene. A separate *Edit* tab opens up and allows you to set the properties of the scene.

The following specific actions are available when you right-click a particular scene:

Toggle Visibility

Renders the selected scene visible or invisible in the *Graphics* window.

Drawables

The following properties characterize the different types of Drawables:

Section	
Name	Specifies the name of the section under which it is stored in the tree.
Plane Parameters	Controls the placement of the section. You can position the section either interactively in the *Graphics* window (see Implicit Plane Tool) or by manually setting the following properties: **Origin** Specifies a Cartesian position vector that defines the origin of the plane.

Section	
	Normal Specifies a Cartesian position vector that defines the second point through which the plane normal passes. The first point is the origin of the Laboratory Coordinate System.
Animation Settings	Controls if the section can be animated in the respective scene. **Turn On Animation** When activated, the section can sweep in its normal direction through the solution domain. See Animation Toolbar Reference.

IsoSurface	
Name	Specifies the name of the isosurface under which it is stored in the tree.
Isosurface Field	Specifies the scalar field upon which the isosurface is based.
Mode	Controls if a single isosurface or multiple isosurfaces are created. The following options are available: • **Single**: Creates one isosurface corresponding to a scalar value. You set a **Single** value. • **Multiple (range)**: Creates multiple isosurfaces corresponding to a range of equally spaced scalar values. You set the **Number of Isosurfaces**, that is the total number of equally spaced isosurfaces in the range, and the **Multiple (range)**, that is the lowest and highest values in the range of isosurfaces. • **Multiple (arbitrary)**: Creates multiple isosurfaces corresponding to a set of scalar values that you specify with numeric entries. You set the **Isovalues**, that is a list of scalar values for the multiple isosurfaces, directly. Alternatively, you can set the number of isovalues and click **Auto Generate Values** to select the isovalues automatically. You can also click **Query** to select a scalar range automatically.
Animation Settings	Controls if the isosurface can be animated in the respective scene. **Turn On Animation** When activated, the isosurface can sweep through the values of the scalar field upon which the isosurface is based. See Animation Toolbar Reference.

Streamlines	
Name	Specifies the name of the streamlines under which it is stored in the tree.
Streamlines Field	Specifies the vector field upon which the streamlines are based.
Integration Direction	Controls the direction of streamline integration from the seed location.

Streamlines	
	Forward The integration proceeds in the forward direction (for instance with the flow if velocity is the Streamlines Field). **Backward** The integration proceeds in the backward direction (for instance against the flow if the velocity is the Streamlines Field). **Both** The integration proceeds both forward and backward from the seed location.
Snap to Part	Controls if the streamline seed points are snapped to a specific surface.
Seed Mode	Controls how the streamline seed points are clustered and defined. The following options are available: • **Point Seed**: Uses a single point or a cloud of points as the seed. The center of the point cloud is specified by the vector components of the **Seed Position** with respect to the selected **Coordinate System**. Alternatively, you can activate the **Display Tool** and position the point interactively (see Using the Point Tool). The **Seed Radius** specifies the radius of the point cloud. It should remain at the default value of zero if you use only one point. The **Number of Points** specifies the resolution of the point cloud. • **Line Seed**: Uses a line of points as the seed. You can define the line either by activating the **Display Tool** and drawing the line interactively (see Using the Line Tool) or by setting the end points **Line Point 1** and **Line Point 2** directly with respect to the selected **Coordinate System**. The **Resolution** specifies the number of evenly spaced points along the line. • **Part Seed**: Uses an existing QuickPart Surface as the seed. The **Seed Part** specifies the surfaces to be used as the seed. The **Part U-Resolution** and **Part V-Resolution** specify the maximum number of points extracted from the part in the u- and v-direction, respectively. The u- and v-direction are part of the computation of an average plane that is based on the specified Seed Part. The u-direction is the vector in this plane that is most closely aligned with the global Cartesian x-axis. If the average plane is an x-y plane (perpendicular to the z-axis), u corresponds exactly to global x. If the average plane is skewed from x-y, the u-axis points toward the global x-axis. The y-axis is then the cross-product of the computed x-axis and the normal. For the x-y plane, this y-axis is exactly in the y-direction.
Animation Settings	Controls if the streamline can be animated in the respective scene. **Turn On Animation** When activated, the streamlines can be animated with a series of moving objects that are called tracers. Each tracer consists of a particle, or head, and a tail. Streamline tracers can move along the streamline paths. See Animation Toolbar Reference.

The following action is available when you right-click the **Drawables** manager node:

New > [Type]

Creates a new Drawable. The following types are available:

- Section
- IsoSurface
- Streamlines

A separate *Edit* tab opens up and allows you to set the properties of the respective Drawable.

The following specific actions are available when you right-click a particular Drawable:

Hide

Hides the selected Drawable from the *Graphics* window. This action is useful for gaining a clear view of other objects. You can restore hidden Drawables again by using the Show action.

Show

Restores a hidden Drawable.

Scene Annotations

The following properties define a scene annotation:

Scene Annotation	
Name	Specifies the name of the scene annotation under which it is stored in the tree.
Annotation Text	Specifies the text that appears in the scene display.

The following action is available when you right-click the **Scene Annotation** manager node:

New

Creates a new scene annotation. A separate *Edit* tab opens up and allows you to set the properties of the annotation.

Results

The **Results** manager node allows you to compute and display summaries of solution data.

Derived Data objects enable you to take samples of solution data at certain positions based on specific coordinates or intervals. The following types of derived data are available:

- Point probe—defines a single point location.
- Line probe—consists of evenly spaced points along a line.

The data value that is extracted for a particular point equals the cell value or surface value—whichever is closest in Eucledian distance to the point. If a point lies outside of the geometry, it extracts the surface cell value of the face that is closest to the point.

The Electronics Cooling Module offers two possibilities to analyze these data:

- Plots—extract solution data from Derived Data objects and display them in two-dimensional graphs.
- Reports—apply statistical functions to solution data, such as maximum, minimum, and average functions. The solution data can be extracted from Derived Data objects or from QuickParts.

Derived Data

The following properties are common to all Derived Data objects:

Common Properties	
Name	Specifies the name of the Derived Data object under which it is stored in the tree.
QuickParts	Specifies the QuickParts that are used as the source for extracting solution data.

You can position point and line probes interactively in the *Graphics* window, see Using the Point Tool and Using the Line Tool, or set the following properties manually:

Edit [Point/Line] Probe Properties		Point Probe	Line Probe
Snap To Part	Allows you to select a part along which you can position the points (this action is essentially snapping to the bounding box of the part).	✓	✓
Point	Specifies the location of the point. The position vector is specified with respect to the selected **Coordinate System**.	✓	
Point 1 and **Point 2**	Specifies the location of the line's end points. The position vectors are specified with respect to the selected **Coordinate System**.		✓
Lock	Allows you to fix the position of any coordinate or any combination of them.	✓	
Resolution	Specifies the number of evenly spaced segments along the line. The actual number of points in the probe is one greater than the number of segments.		✓
Coordinate System	Specifies the coordinate system for positioning the points.	✓	✓

The following action is available when you right-click the **Derived Data** manager node:

New > [Type]

Allows you to create a new point probe or a new line probe. A separate Edit tab opens up that displays the customizable properties of the probe.

Plots

The following properties define the characteristics of a plot:

Plot Properties	
Name	Specifies the name of the plot under which it is stored in the tree.

Plot Properties	
Derived Data Items	Specifies the Derived Data objects from which the data are extracted.
Y-Axis Fields	Specifies the specific scalar field function to be plotted as the data value for the y-axis. See Scalar Field Functions.
X Axis Type	Specifies the type of data to use for the x-axis of the plot. The following options are available: • **Direction**: Plots position information along the axes. Selecting this option activates the **Direction** property. • **Scalar**: Selects a solution variable to be plotted along the axes. Selecting this option activates the **X-Axis Scalar** property.
Direction	Since derived data exist in three-dimensional space while each axis of a plot represents only a single dimension, the three-dimensional coordinates of the points being plotted must be reduced to one. This objective is accomplished by specifying a direction vector to which the position of sample locations is projected. The following options are available: • **X-Axis**, **Y-Axis**, **Z-Axis**: Project the sample locations to the x-,y-, or z-axis of the global coordinate system. • **Custom Axis**: Specify a **Custom Direction** vector to which the position of sample locations is projected. For example, in order to plot X coordinates of the selected derived data items, you would specify a direction vector of [1,0,0].
X-Axis Scalar	Specifies the scalar field function to be plotted along the x-axis of the plot. See Scalar Field Functions.

The following action is available when you right-click the **Plots** manager node:

New

Creates a new plot. A separate Edit tab allows you to set the properties of the plot.

The following specific action is available when you right-click a particular plot:

Open

Displays the plot in the *Graphics* window or makes it active if it is already open.

Reports

The following properties define the characteristics of a report:

Report Properties	
Name	Specifies the name of the report under which it is stored in the tree.
Report Field	Specifies the scalar field function upon which the report is based. See Scalar Field Functions.
Report Items	Specifies the items from which the solution data are extracted. You can chose from QuickParts, QuickPart Surfaces, and Derived Data objects.

The following actions are available when you right-click the **Reports** manager node

Create Post Report

Creates a new report. A separate Edit tab allows you to set the properties of the report.

Tabulate All

Displays all reports in a spreadsheet-like dialog. See Tabulate.

Export All

Allows you to save all reported data in a `.csv` file.

The following specific actions are available when you right-click a particular report:

Tabulate

Displays the following properties and quantitative values for the selected report in a spreadsheet-like dialog:

Tabular Data	
QP/QPS	Displays the respective report item—a QuickPart,a QuickPart Surface, or a Derived Data item.
Field	Displays the respective report field.
Minimum	Returns the minimum field value within the respective report item.
Maximum	Returns the maximum field value within the respective report item.
Average	For a QuickPart, returns the volume averaged field value computed as: $$\text{Volume Average} \equiv \frac{1}{V}\int \phi \, dV = \frac{\sum_c \phi_c V_c}{\sum_c V_c}$$ where ϕ_c is the cell value of the selected scalar field and V_c is the cell volume. For a QuickPart Surface, returns the surface averaged field value computes as: $$\text{Surface Average} \equiv \frac{1}{a}\int \phi \, da = \frac{\sum_f \phi_f A_f}{\sum_f A_f}$$ where ϕ_c is the face value of the selected scalar field and A_f is the face area.
Total	Returns the sum field value over the faces of a QuickPart Surface. This quantity is only available for the following report fields: • Boundary Heat Transfer • Mass Flow Rate

It is possible to tabulate several reports at once by holding down the <Ctrl> key when selecting the objects.

Export

Allows you to save the reported data in a `.csv` file.

Field Functions

The following field functions are made available to the electronics cooling simulation depending on the physics setup:

Scalar Field Functions

Boundary Emissivity (requires radiation)

The gray emissivity at a surface.

Boundary Emissivity on External Side (requires radiation)

The gray emissivity at the surface for the outward facing side. This field function only available on outer wall surfaces if the Thermal Specification is set to Environment.

Boundary Heat Flux

The magnitude of the heat flux vector normal to the surface. It is defined as $(\dot{\mathbf{q}}'' \cdot \mathbf{a})/|\mathbf{a}|$.

Boundary Heat Flux on External Side

The magnitude of the heat flux vector normal to the surface for the outward facing side. This field function is only available on outer wall surfaces if the Thermal Specification is set to Environment.

Boundary Heat Transfer

The heat transferred at the surface, defined as $\dot{\mathbf{q}}'' \cdot \mathbf{a}$.

Boundary Irradiation (requires radiation)

The irradiation at the surface. The irradiation is the radiative heat flux incident on a surface.

Boundary Irradiation on External Side (requires radiation)

The irradiation at the surface for the outward facing side. This field function is only available on outer wall surfaces if the Thermal Specification is set to Environment.

Boundary Radiation Heat Flux (requires radiation)

The net radiative heat flux to the surface (absorption minus emission). A positive value indicates a net absorption by the boundary.

Boundary Radiation Heat Flux on External Side (requires radiation)

The net radiative heat flux for the outward facing side. This field function is only available on outer wall surfaces if the Thermal Specification is set to Environment.

Cell Quality

Cell quality is a function that is determined from the relative geometric distribution of the cell centroids of the face neighbor cells and of the orientation of the cell faces. Generally, flat cells with highly non-orthogonal faces have a low cell quality. A cell with a quality of 1.0 is considered perfect. A degenerate cell has a cell quality approaching 0. Depending on the physics that you select, the cell quality of a cell can be fairly low and still provide a valid solution. However, poor cell quality is likely to affect both the robustness and accuracy of the solution.

Density

The material density defined by the method that you chose for the Setup.

Effective Viscosity

The sum of the laminar and turbulent viscosities $\mu + \mu_t$.

External Ambient Temperature

The surface ambient temperature that you specify on a wall surface. This field function is only available on outer wall surfaces if the Thermal Specification is set to Convection or Environment.

External Heat Transfer Coefficient

The surface external heat transfer coefficient that you specify on a wall surface. This field function is only available on outer wall surfaces if the Thermal Specification is set to Convection or Environment.

Heat Transfer Coefficient

The wall boundary heat transfer coefficient defined by:

$$h = \left(\dot{\mathbf{q}}''_{\text{conduction}} \cdot \mathbf{a}\right) / \left[|\mathbf{a}|\left(T_{\text{ref}} - T_f\right)\right]$$

where T_f is the wall boundary temperature and T_{ref} is the specified reference temperature in the field function.

Mass Flow Rate

The rate at which mass flows through an open surface.

Static Pressure

The spherical part of the stress tensor acting in fluids, which is the same as the actual thermodynamic pressure of the fluid. When the gravity model is active, it is related to the working pressure, (which becomes the piezometric pressure), by Eqn. (1649) for variable density flows and by Eqn. (1653) for constant density flows.

Temperature

The temperature field.

Thermal Conductivity

The thermal conductivity material property according to the method that you defined for the physics. This field function is a scalar value and therefore not populated in regions where thermal conductivity is defined by an anisotropic tensor.

Total Pressure

The total pressure (gauge) is the pressure that is obtained from isentropically bringing the flow to rest. For an ideal gas with constant specific heat, it is defined as

$$P_t = P_{\text{abs}}\left[\left(1 + \frac{(\gamma - 1)}{2} M^2\right)^{\gamma/(\gamma-1)}\right] - P_{\text{ref}}$$

where P_{abs} , P_{ref} , M and γ are the absolute pressure, reference pressure, Mach number, and ratio of specific heats, respectively. For alternate equations of state, or a non-constant specific heat, P_t is obtained by integrating $\mathrm{d}\, h = \mathrm{d}\, P/\rho$ from static conditions to total conditions.

Turbulent Dissipation Rate

The transported variable ε in the K-Epsilon Turbulence model.

Turbulent Kinetic Energy

The transported variable k in the K-Epsilon Turbulence model.

Turbulent Viscosity

The turbulent viscosity μ_t in the K-Epsilon Turbulence model. The K-Epsilon Turbulence model belongs to the family of Eddy Viscosity Models and therefore uses the concept of a turbulent viscosity to model the Reynolds stress tensor as a function of mean flow quantities.

Velocity: Component n (where n = 0, 1, 2)

The velocity component in x, y, or z direction respectively.

Velocity: Magnitude

The magnitude of the **Velocity** vector.

Vorticity: Magnitude

The magnitude of the **Vorticity** vector.

Wall Shear Stress: Magnitude

The magnitude of the **Wall Shear Stress** vector.

Vector Field Functions

Area

The vector of the face area components, that is the area magnitude multiplied by the unit normal. You can also run the Area field function on an unmeshed surface.

Velocity

The velocity vector field.

Vorticity

The vector variable with components ζ_x , ζ_y , ζ_z defined as:

$$\zeta_x = \frac{\partial w}{\partial y} - \frac{\partial v}{\partial z}, \qquad \zeta_y = \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x}, \qquad \zeta_z = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$$

where u , v and w are velocity components in the x, y and z directions. $\nabla\times\ \mathbf{v}$ is the curl of the velocity field.

Wall Shear Stress

The wall shear stress vector, which is defined at a wall face as $\vec{\tau}_w = (-\mathbf{T}\cdot\mathbf{a}/|\mathrm{a}|)$, where $\mathbf{T}$ is the stress tensor and $\mathbf{a}$ is the **Area** vector.

Resistor Networks Formulation

For all network types that are available within the Electronics Cooling Toolset, the junction node temperature T_J is obtained by solving the following equation:

$$\sum_i \left(\frac{T_J - T_i}{R_{Ji}} \right) = S_J \tag{1416}$$

where:

- T_i is the temperature of the thermal network node i that is connected to the junction node.
- R_{Ji} is the thermal resistance between the junction node and the connected node i.
- S_J is the heat source at the junction node.

At interfaces between thermal networks and Quick Parts, face temperatures T_f are calculated as:

$$-\mathbf{q}''_f \cdot \mathbb{d}\mathbf{a}_f + \sum_i \left(\frac{T_f - T_i}{R_{fi}} \right) = 0 \tag{1417}$$

where:

- $\mathbf{q}''_f$ is the heat flux out of surface face f.
- $\mathbf{a}_f$ is the area vector of surface face f.
- T_i is the temperature of the thermal network node i that is connected to the respective interface node.
- R_{fi} is the thermal resistance between the interface node and the connected node i.

If a chip surface is not represented by a network node, then the faces of the contacting Quick Part are modeled as adiabatic:

$$\mathbf{q}''_f = 0 \tag{1418}$$

Bibliography

[191] *JEDEC Solid State Technology Association, JEDEC Standard JESD15-1, "Compact Thermal Model Overview", October 2008.*

[192] *Bar-Cohen, A. and Krueger, W.B., "Thermal Characterization of Chip Packages - Evolutionary Development of Compact Models" , Thirteenth Annual Semiconductor Thermal Measurement and Management Symposium - SEMITHERM XIII, Austin, TX, January 28-30, 1997 IEEE, pp. 180-197.*

Optimate User Guide

Optimate is an add-on for STAR-CCM+ that allows you to perform design exploration studies (which includes design sweeps, optimization, design of experiments, and robustness analyses) within the STAR-CCM+ environment.

Optimate is installed alongside STAR-CCM+ and is accessed from the toolbar. Optimate uses technology from Red Cedar Technology® to execute, monitor, and post-process the design exploration studies. The SHERPA algorithm is used for the Optimization and Pareto Optimization modes of Optimate+.

Note: Optimate is designed for use when STAR-CCM+ is the only analysis tool in the automation or optimization process. If you want to use an additional tool, you must use an external optimization package. In this case, Optimate can help create the necessary batch scripting to run STAR-CCM+.

Contents:

Capabilities of Optimate and Optimate+
How Does Optimate Work?
Optimate Run Modes
Optimate Variables
Optimate Responses
The Optimate Workflow
Saving and Loading Optimate Projects
Post-Processing Optimate Results
Customizing the Optimate Workflow Using Java Macros
Using Design Sets
Running Optimate Jobs
Monitoring Designs
Exporting Scenes from Optimate to STAR-View+
Running a CAD Robustness Study
Optimate Licensing
Coupling Optimate to External Optimization Codes
Optimate Options
File Descriptions
Troubleshooting

See Also:

The SHERPA Algorithm

Capabilities of Optimate and Optimate+

There are two variants of Optimate: Optimate and Optimate+.

Optimate provides a fully automated process to gain insight into a product's behavior. Optimate+ provides a fully automated process to find a set of optimum designs that match engineering objectives. The following table provides a list of supported features in each variant:

Functionality	Optimate	Optimate+
Design Sweeps	Yes	Yes
Design of Experiments	Yes	Yes
Stochastic Analysis	Yes	Yes
Optimization using SHERPA	No	Yes
Pareto Optimization using MO-SHERPA	No	Yes

How Does Optimate Work?

Optimate is a plug-in that works together with STAR-CCM+ to help you understand your design space.

Every Optimate study is based on a STAR-CCM+ simulation, so you must first set up your case in STAR-CCM+ before launching Optimate. In Optimate, you choose the type of study to perform. You then create variables from STAR-CCM+ simulation objects that Optimate can modify throughout the course of the study. Then define the aims and goals for your design exploration and set up the run parameters.

When you run an Optimate study, the STAR-CCM+ server for the baseline simulation shuts down to free up the license. The STAR-CCM+ solver launches in batch mode and starts solving the first job in the study. You control the number of jobs that Optimate runs in a study, as well as how many jobs run at the same time, in the *Assembly* panel. When running a particular job, Optimate modifies the simulation settings in STAR-CCM+ (which can include changes to the geometry design, or changes to the physics conditions) and starts recording the outputs. Optimate checks the running jobs to see if they are complete. The frequency with which Optimate checks the jobs is specified in the *Assembly* tab. When a job completes, Optimate records the outputs.

For each analysis, Optimate writes out certain files. See File Descriptions.

Optimate Run Modes

You can perform a number of studies with Optimate.

Design Sweeps

Design Sweeps are used for design exploration. You can use Design Sweeps mode to replicate test results, create aerodynamic loads databases, create compressor maps, or with any application where you know the configuration for each job before running the analysis. In Design Sweeps mode, you provide the variables (which are stored in a `.csv` file) before starting the analysis. No optimization takes place in Design Sweeps mode.

Design of Experiments

A Design of Experiments study is used for local exploration around a particular design. In this type of analysis, you supply the variable values and Optimate uses Design of Experiments principles to create a response surface. In this way you can assess the sensitivities of design factors and their interaction with each other.

For each DoE, you can select one of the following sensitivity and screening methods:

- *2 Level Full Factorial Runs* — A two-level full factorial DoE studies the various factors, and all possible combinations of the factors, at a specified number of levels. From this type of study, you can gain insight into the sensitivity of each factor and every combination of factors to the design responses. A two-level full factorial study examines all the factors at two levels: low and high. The number of runs that are required for a 2-level full factorial study is 2^n, where *n* is the number of DoE factors.
- *3 Level Full Factorial Runs* — A three-level full factorial DoE study is conceptually similar to a two-level full factorial study, with the main difference being that all factors and combinations of factors are evaluated at three levels: low, mid, and high. The number of runs that are required for a three-level full factorial study is 3^n, where *n* is the number of DoE factors.

For each DoE, you can select one of the following response surface modelling methods:

- *Latin Hypercube* — The Latin Hypercube sampling option is an efficient method to generate a response surface for the design space. You can use this generated response surface for higher-level DoE (Design of Experiments) study types, like two- or three-level full factorial studies, since running these studies on a response surface is computationally inexpensive. The minimum number of evaluations to fit a quadratic surface is:

  $$\frac{(n+1)\cdot(n+2)}{2}$$

  where `n` is the number of DoE factors. You are recommended to evaluate at least two to four more evaluations than the minimum, up to a maximum of two times the minimum.
- *Using Existing Designs* — Allows you to create a response surface that fits a design set specified by you. The designs are specified through a `.csv` file which you can export through HEEDS Post. To export a set of designs from HEEDS Post, open the design table and select **Save Plot Data**.

Stochastic Analysis

A Stochastic Analysis study is used to assess the impact of minor variations in the input values on a particular design. These variations could come from installation or manufacturing tolerances. In a Stochastic Analysis,

Optimate applies a specified probability distribution to the inputs. The resulting distribution of the outputs helps you assess the robustness of the design. You can also specify constraints on design responses to obtain a prediction of failure rate.

Optimization Using SHERPA

In an Optimization study, the optimization algorithm selects the input values to meet the specific goals of the analysis. There are two types of optimization studies: Single-Objective and Multi-Objective.

In Single-Objective optimization (**Weighted sum of all objectives** run mode), typical goals include minimizing pressure drop, maximizing outlet uniformity, minimizing component temperatures, and minimizing stress. You define the range of permissible values for each variable and the search algorithm chooses any value within the bounds during the analysis. Single-Objective Optimizations use the SHERPA search algorithm by Red Cedar Technology®. See The SHERPA Algorithm.

In Multi-Objective optimization (**Multiple objective tradeoff study (Pareto front)** run mode), you optimize based on two competing objectives. Pareto Optimization is a formulation for investigating multi-objective optimization problems. This type of optimization is suited to cases where there are two objectives which are competitive in nature. In such cases, there is no single optimum design but rather there is a curve along which all designs are optimum in one objective for a given value in the other objective. This behavior is called non-dominated sorting. The resulting curve is referred to as a Pareto Front and defines the optimum trade-off relationship between two objectives. In Pareto Optimization, you are not required to define the weighting for the competing objectives before running the analysis. The study returns a set of all optimal designs that lie on the Pareto Front. Multi-Objective Optimizations use the MO-SHERPA search algorithm by Red Cedar Technology®.

When performing an optimization study, you must define at least one objective. If you define multiple objectives, a linear weighting is used to combine the objectives. Because the weighting of possibly competing objectives is defined a priori, running an optimization analysis returns a single best design.

Contents:

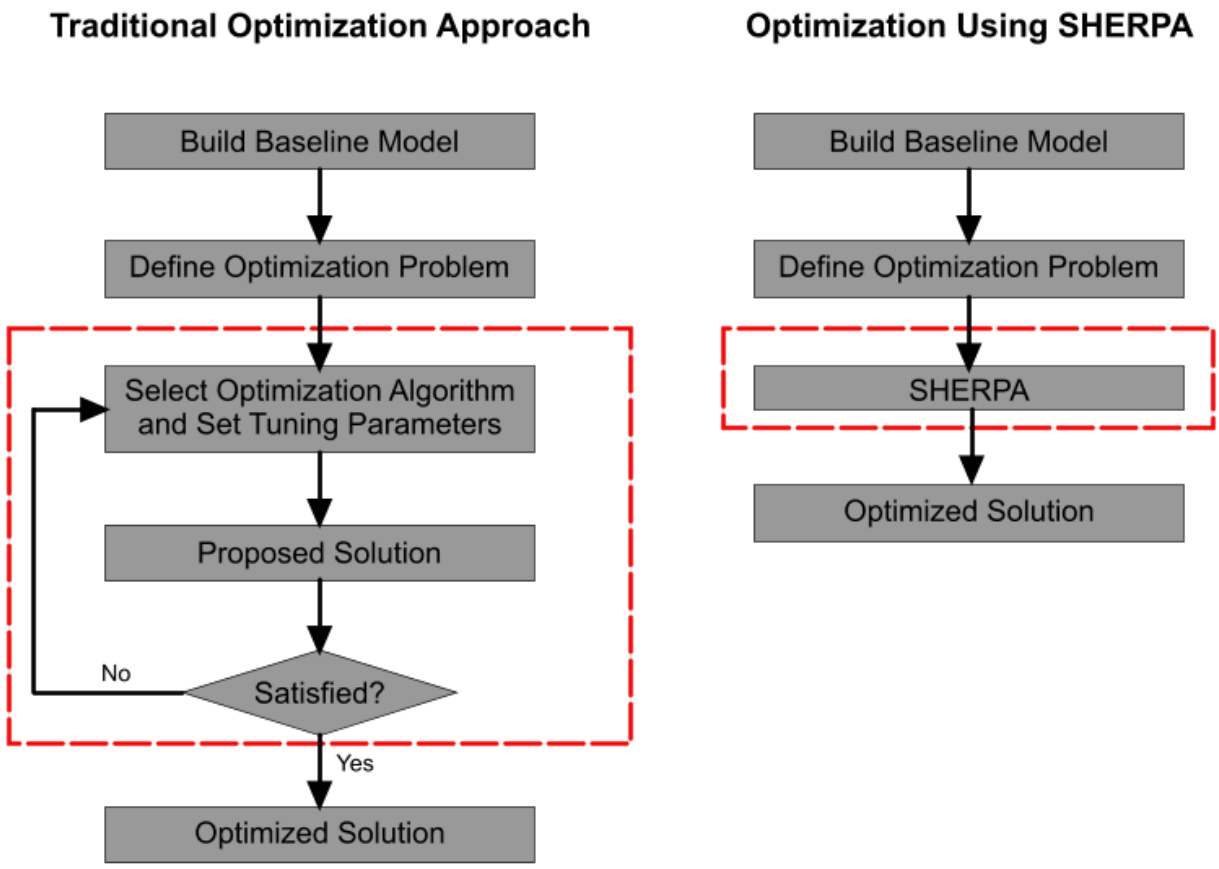
The SHERPA Algorithm

The SHERPA Algorithm

SHERPA is a unique search algorithm that is developed by Red Cedar Technology®.

During a single search, SHERPA (which stands for Simultaneous Hybrid Exploration that is Robust, Progressive, and Adaptive), uses multiple search methods simultaneously rather than sequentially. This approach uses the best attributes of each search method. If a particular search method is deemed to be ineffective, SHERPA reduces its participation.

A comparison of a traditional optimization approach and of the SHERPA approach is displayed below.



While running the SHERPA algorithm, a combination of global and local search methods are used. At any given time, the number of different methods that are used can range between two and ten. Unlike traditional optimization algorithms that require you to tune parameters manually, the tuning parameters in each method that SHERPA uses are modified automatically during the search according to knowledge gained about the nature of the design space. As SHERPA learns more about the design space, it determines when and to what extent to use each search method.

The general advantages of using SHERPA are:

- You do not need to spend time understanding the design space before choosing a suitable algorithm. SHERPA learns about the design space.
- You do not need to have any expertise in optimization algorithms and applications. SHERPA makes all the decisions about which method to use and how to tune them.
- You can define realistic problems that are based on real engineering or business costs and benefits, without being constrained by the limitations of a particular search method. With SHERPA, your analysis can contain much broader goals and can include a larger number of variables compared to traditional approaches.

Optimate Variables

Optimate variables are the values from the baseline STAR-CCM+ simulation that Optimate can vary for design exploration studies. The types of STAR-CCM+ simulation objects that can be selected as Optimate variables are listed in this section.

Design Parameters

In 3D-CAD models, the design parameters that you expose are available in Optimate. Changing a design parameter pushes the modified geometry though the mesh pipeline so that it is remeshed. Make sure that range for the variables allows for the geometry to remesh successfully. If a change to a design parameter causes the geometry to become invalid, the Optimate analysis fails to run.

You can create variables from the following 3D-CAD design parameters:

- Scalar quantities — values such as dimension distances, radii, extrusion lengths, taper angles.
- Translation components — Optimate attempts to translate a coordinate system with the same name as the current design parameter. This feature is useful if you want to rotate a component after the translation. It is important to create the translation variables first, before any rotation variables.

Design parameters that originate from CAD Clients are also supported.

Parts for Swapping

You can create part variables using parts that are imported directly into the **Parts** node in the source simulation file. To use parts as variables, you must supply at least two CAD files in a specified directory. If swapping a part causes a mesh operation to return an invalid result, the Optimate analysis fails to run. If you select a NASTRAN (`.nas`) file, the simulation preferred length unit is selected as the import unit.

Motions

Motion variables are created from defined motions in the simulation file. If you are using reference frames for boundary conditions, variables are not created. You must create a motion and use the automatically created reference frame as the boundary condition for the region. You can create variables from the following motion types:

- Rotation rate
- Translation velocity

Regions

Region variables are created from regions that are stored within the source simulation file. Selecting a region populates the *Regions Physics Values* property with the physics values of that region.

Regions Physics Values

You can create variables from physics values that are stored in the *Regions* property within the source simulation file. Only the physics values of regions whose *Method* property is set to **Constant** are supported.

The following region physics values are supported:

- Volumetric heat source

Mesh Values

You can perform mesh dependency studies by varying the base size in your mesh continuum (regions-based meshing) or your automated mesh operation (parts-based meshing).

Boundaries

Boundary variables are created from boundary data that is stored within the source simulation file. Selecting a boundary as a variable populates the *Boundary Physics Values* property with the physics values of that boundary.

Boundary Physics Values

You can create variables from boundary physics values. Only the physics values of boundaries whose *Method* property is set to **Constant** are supported.

The following boundary physics values are supported:

- Velocity magnitude
- Mass flow rate
- Static temperature
- Total temperature
- Static pressure
- Total pressure
- Mach number
- Flow angles
- Mass fractions
- Wall roughness
- Turbulence scalars
- Thermal scalars

Flow Angles

If you want to vary the inlet flow angles, Optimate can modify a local Cartesian coordinate system labeled `wind`. To use this feature, create a coordinate system in the source simulation file and name it `wind`. Also make sure that you align this coordinate system so that the flow with zero pitch angle, zero roll angle, and zero yaw angle is aligned along the positive X axis. Pitch, yaw , and roll are defined as rotations about the Y, Z, and X axis respectively. All reports that remain relative to the *wind* coordinate system must reference the *wind* coordinate system.

Scalar User Field Functions

You can create variables from user field function that are stored within the source simulation file. Only scalar field functions whose *Definition* property contain numeric values are supported. Examples of allowable definitions are:

- `10`
- `-1.325`

Optimate Responses

To analyze how different designs affect the simulation results, Optimate records data from selected STAR-CCM+ reports at the end of each study.

Optimate records data from the reports that are selected as responses. Once you select a STAR-CCM+ report as an Optimate response, you can use it to set an engineering objective. For example, a response that records pressure data can be used by Optimate to explore different designs aimed to maximize/minimize pressure.

In Optimization studies, objectives drive the Optimate study, whereas in other study types the objectives only affect the order in which the recorded responses are listed.

The following requirements must be satisfied for the study to complete successfully:

- For all studies, you must record data from at least one report.
- For optimization cases, you must select at least one objective.
- For Pareto optimization cases, you must select at least two objectives.
- For DoE studies, you must have at least two responses.

The Optimate Workflow

This section contains the recommended workflow to set up a study in Optimate.

The recommended process to follow when performing a study in Optimate is provided below:

1. Identify the goals of your study.
2. Create a case in STAR-CCM+, focusing on what you want to achieve in Optimate.

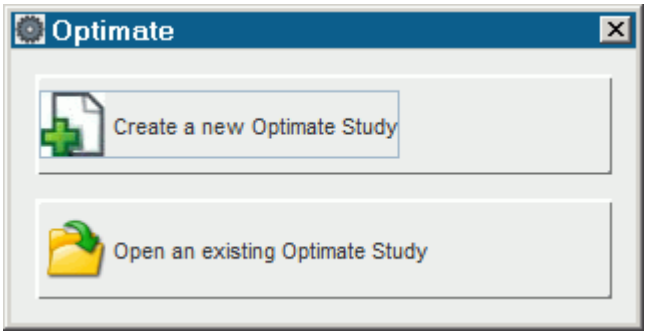
   This stage involves setting up your case in STAR-CCM+ with a suitable mesh and physics setting.

   Create the reports that you want Optimate to extract data from.

   Finally, make sure that you create the relevant scenes that you want Optimate to use for post-processing.
3. Launch Optimate:

   a) From the STAR-CCM+ user interface, click (**Start Optimate**).

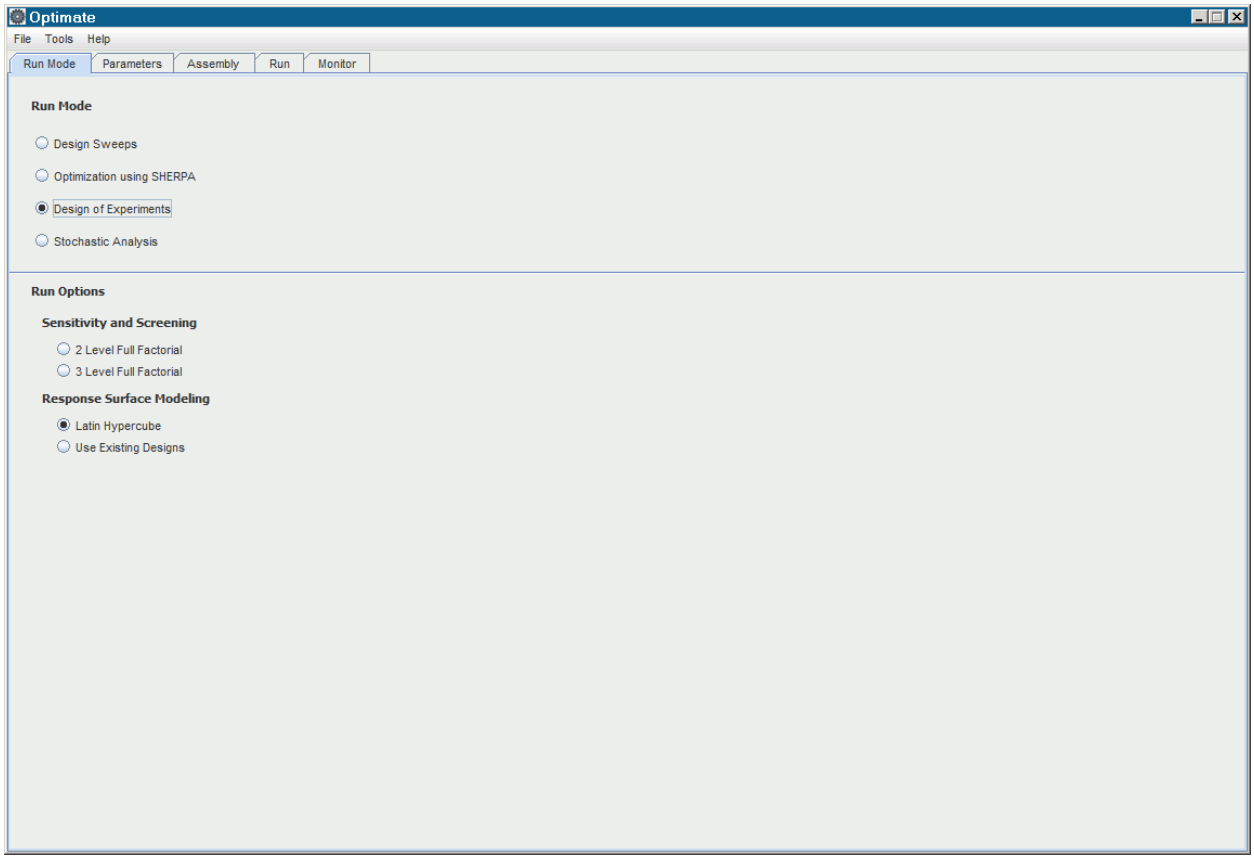
   The *Optimate* dialog appears.



   b) Select the option that is relevant to your situation.
4. Select the type of study that you want Optimate to run:

   a) In the *Run Mode* tab, set *Run Mode* to the type of run that you want to perform in Optimate. See Optimate Run Modes.

   b) For each Run Mode, Optimate presents a set of **Run Options** in the lower part of the panel. See Run Mode Options Reference.



5. Create the variables that you want Optimate to modify in the study.

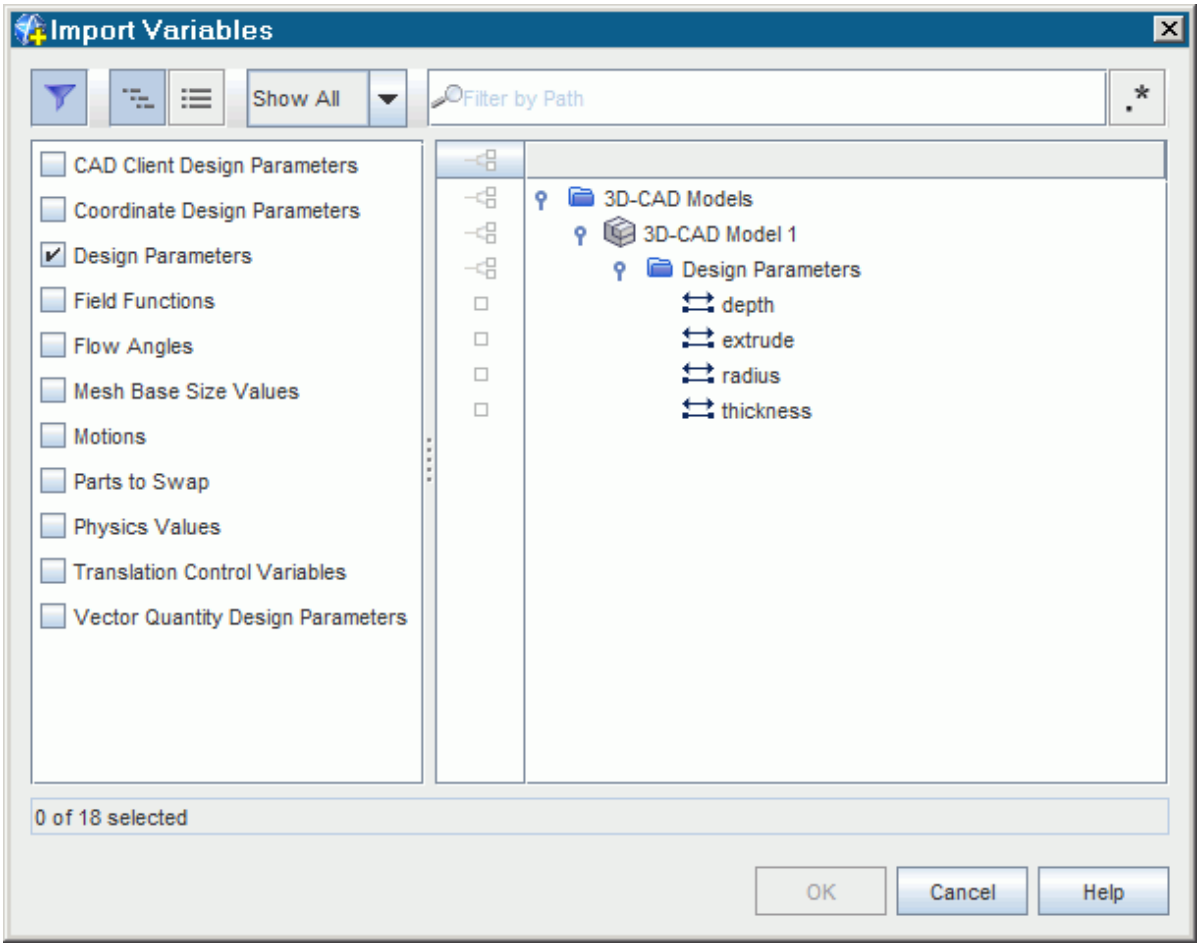
   Optimate variables are the values from the baseline STAR-CCM+ simulation that Optimate can vary for design exploration studies.

   To create a variable:

   a) In the *Parameters* tab, click:

   b) In the *Import Variables* dialog, select the desired variable and click **OK**.

      In the *Import Variables* dialog, there are pre-defined filters for the various categories of Optimate variables. To access these filters, click the filter icon.



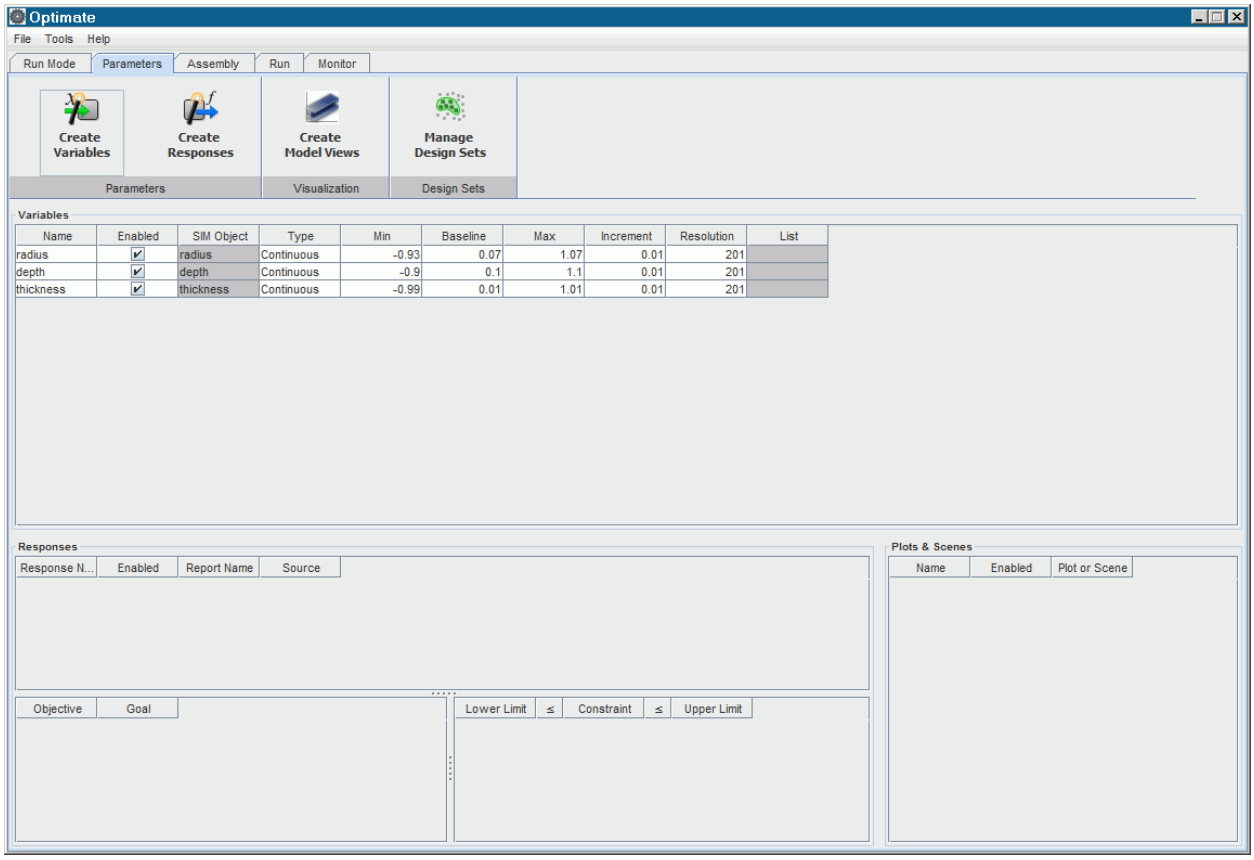
You can select multiple variables at once. For a list of supported variables, see Optimate Variables.

The selected variable is added to the *Variables* group-box. All active variables are displayed in this group-box.

c) Set the variable properties based on your analysis requirements. See Variables Reference.

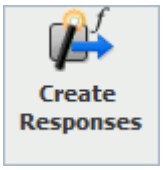
Note: When performing a **Design Sweeps** study or a **Design of Experiments** study, you can define the Optimate variables using tabular data from a `.csv` file. See Using Design Sets.

d) Repeat these steps for all other active variables in Optimate.



6. To analyze how different designs affect the simulation results, select the STAR-CCM+ reports that you want Optimate to record at the end of each analysis:

   a) In the *Parameters* tab, click:



   b) In the *Import Reports* dialog, select the reports that you want Optimate to extract data from and click **OK**.

   You can select multiple reports at once. If you want to extract data from all the available reports, right-click on a blank area in the *Import Reports* dialog and choose **Select All**.

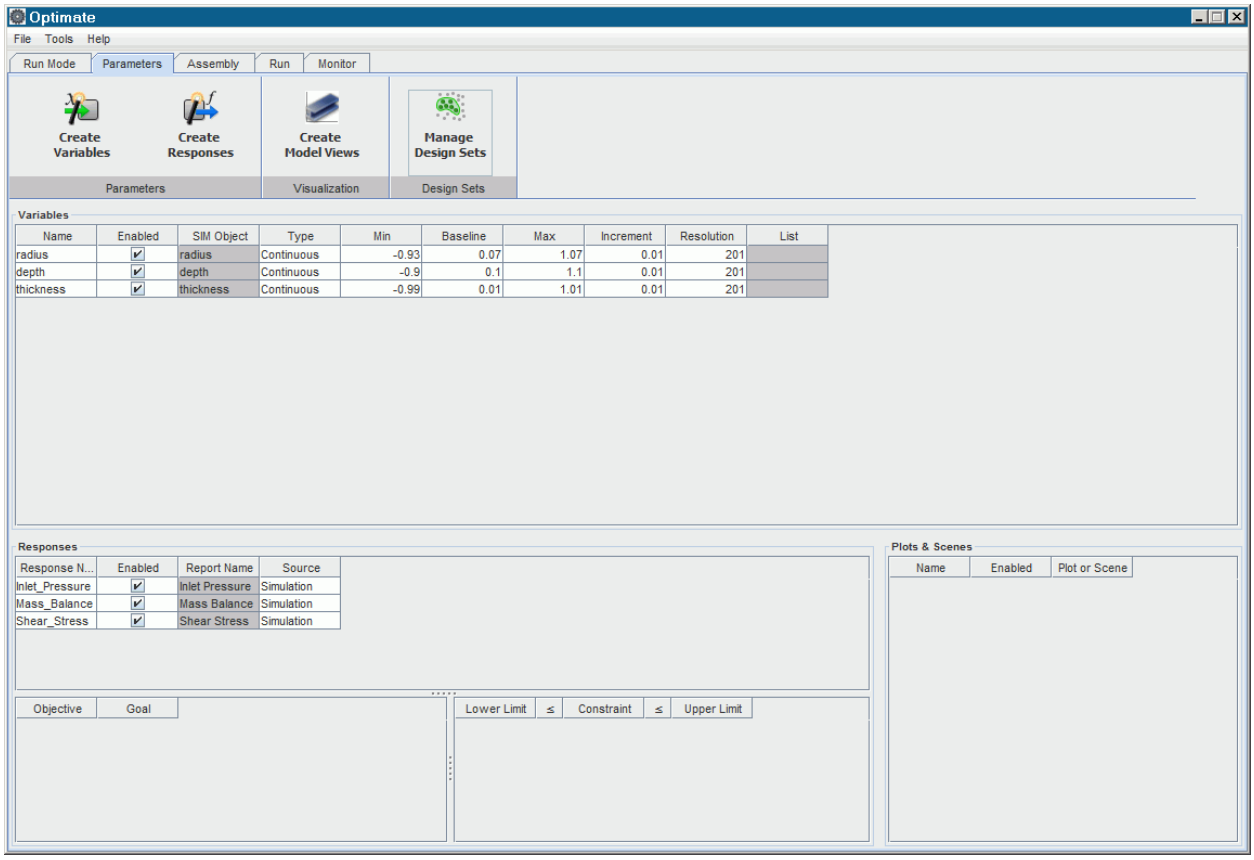
   Note: You must select at least one report.

   The selected reports are added to the *Responses* group-box. All responses are displayed here.

   c) For each response, set the available properties based on your analysis requirements.

   Some of the properties are set through the *Responses Properties* dialog which you can access when you right-click a response and select **Properties**.

   For a list of all the available properties, see Responses Reference.



7. When performing an Optimization study, define the goals of the analysis by setting the relevant objectives. For information on Optimization studies, see Optimization Using SHERPA . To set an objective:
   a) Right-click the response of interest and select **Set as objective**.

      The objective appears at the bottom of the *Parameters* tab.
   b) Set the study goal to either minimize or maximize by setting the objective *Goal* property.

Objective	Goal
InletPressure	Minimize

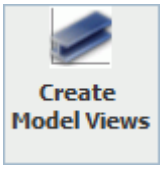
      You can set additional properties through the *Objective Properties* dialog that appears when you right-click an objective and select **Properties**.

      For a list of all the available properties, see Objective Properties.
8. Optionally, you can constrain the report value to determine whether a design is feasible or not. To create a constraint:
   a) Right-click the response of interest and select **Set as constraint**.

      The constraint appears at the bottom of the *Parameters* tab.
   b) Set the constraint properties based on your study requirements. See Constraint Properties.

      You can set additional properties through the *Constraint Properties* dialog that appears when you right-click a constraint and select **Properties**.
9. Select the plots and scenes that you want Optimate to create representations for:

a) In the *Parameters* tab, click:

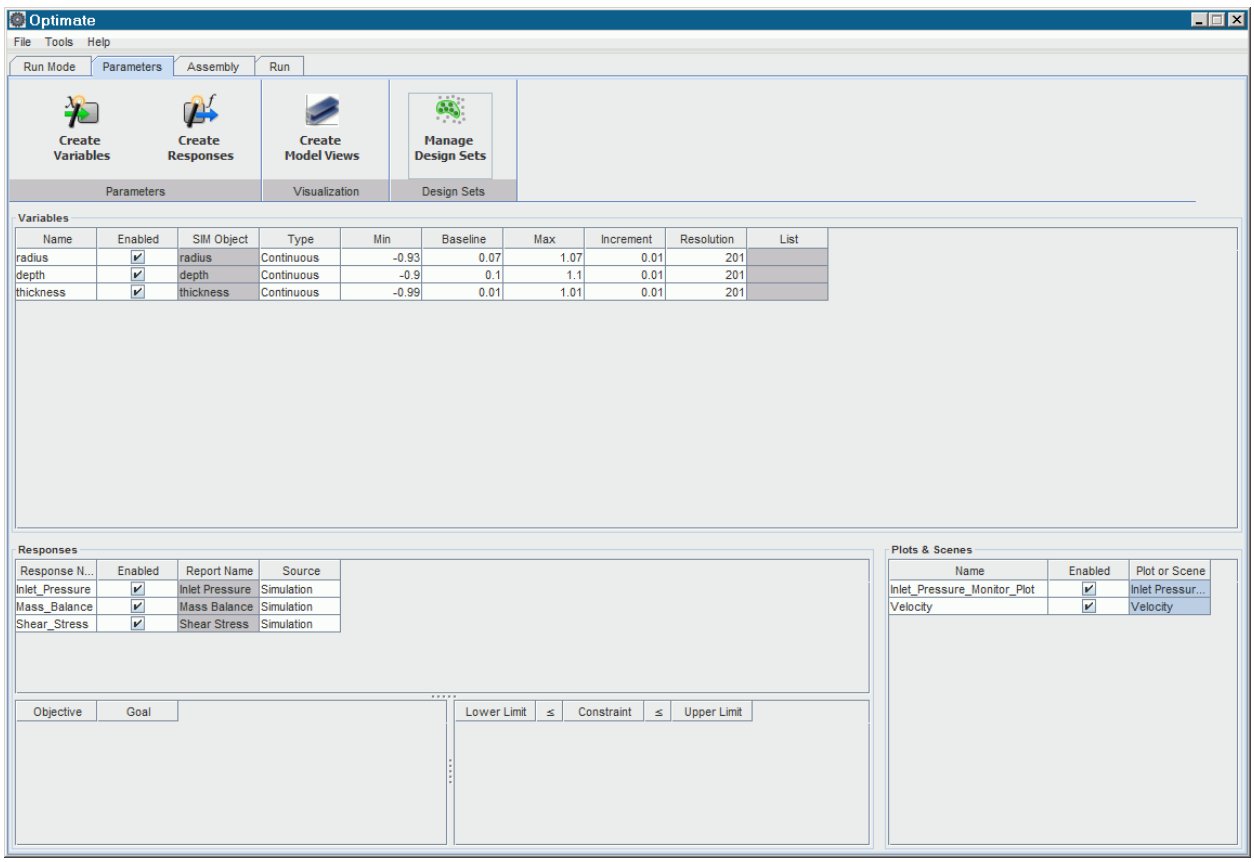


b) In the *Import Plots and Scenes* dialog, select the plots and scenes that you want Optimate to create representations for and click **OK**.

You can select multiple objects at once. If you want to create representations for all the available plots and scenes, right-click on a blank area in the *Import Plots and Scenes* dialog and select **Select All**.

The selected plots and scenes are added to the *Plots & Scenes* group-box, at the bottom right of the *Parameters* tab. All plots and scenes are displayed here.

c) For each plot and scene, set the available properties based on your study requirements. See Plots and Scenes Reference.



10. Define the run settings for the Optimate study:

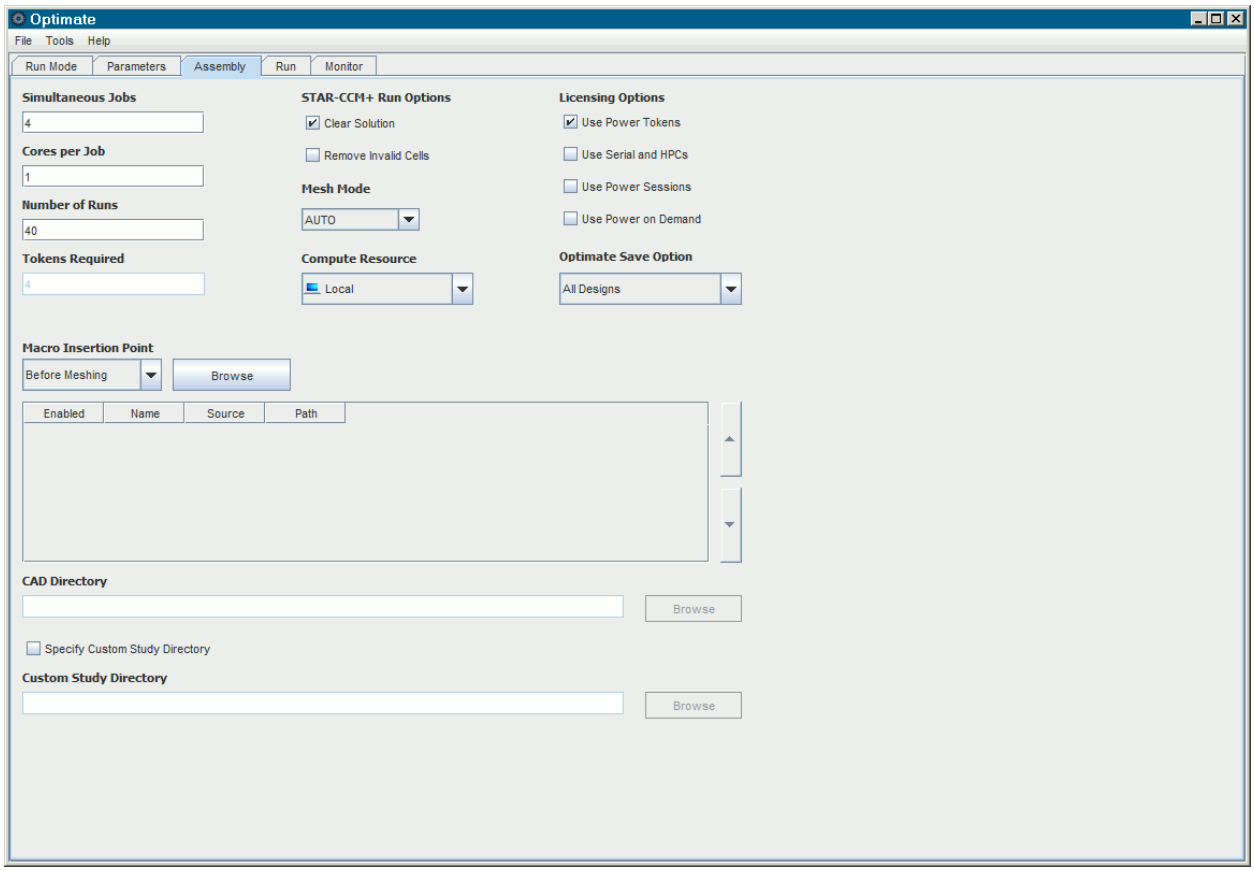
a) In the *Assembly* tab, select the number of cores to run the study on, as well as how many jobs to run at the same time. See Job-Related Options.

b) Define the STAR-CCM+ run options. See STAR-CCM+ Run Options.

c) Define the licensing options. See Licensing Options.

d) Select a compute resource to use for the study. Compute resources contain the stored properties of clusters and remote machines. See Creating a Compute Resource.

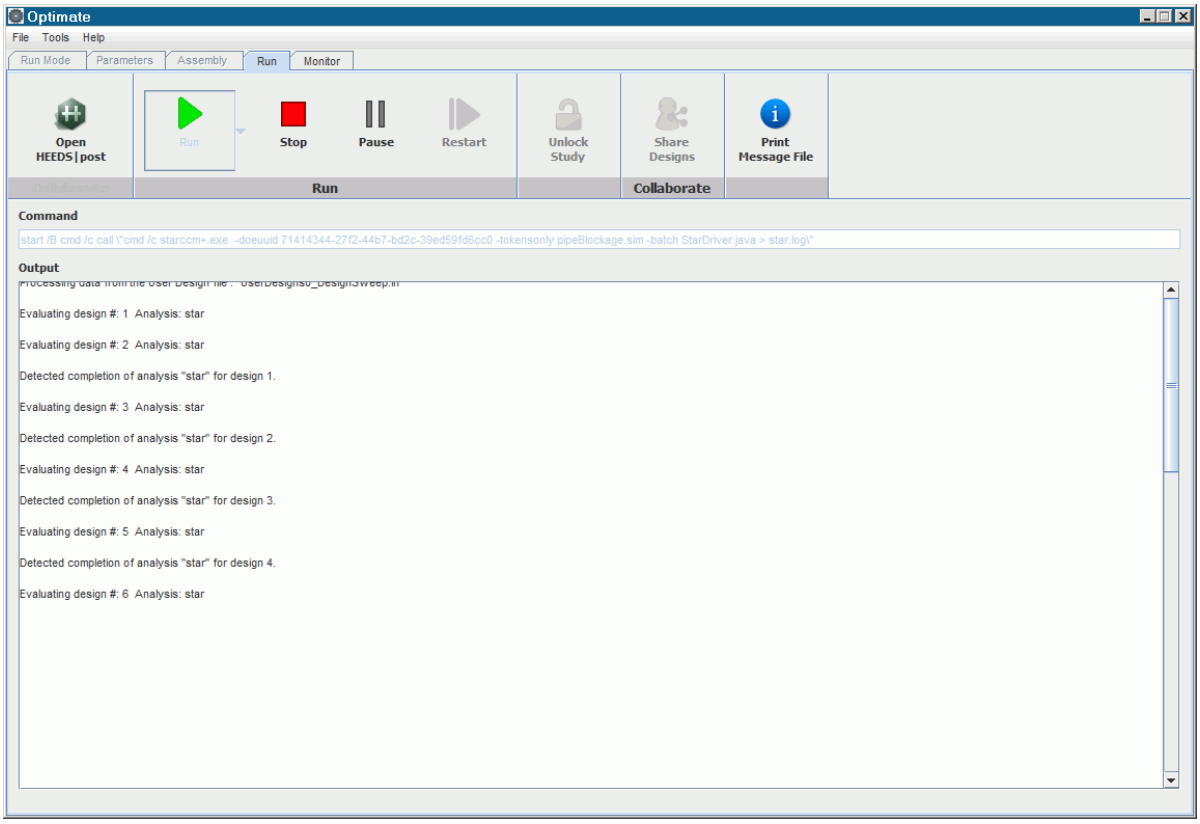
You can use multiple compute resources together in a compute set if they are able to execute the same command to launch STAR-CCM+. See Using Multiple Compute Resources Together.



11. Run the Optimate study:

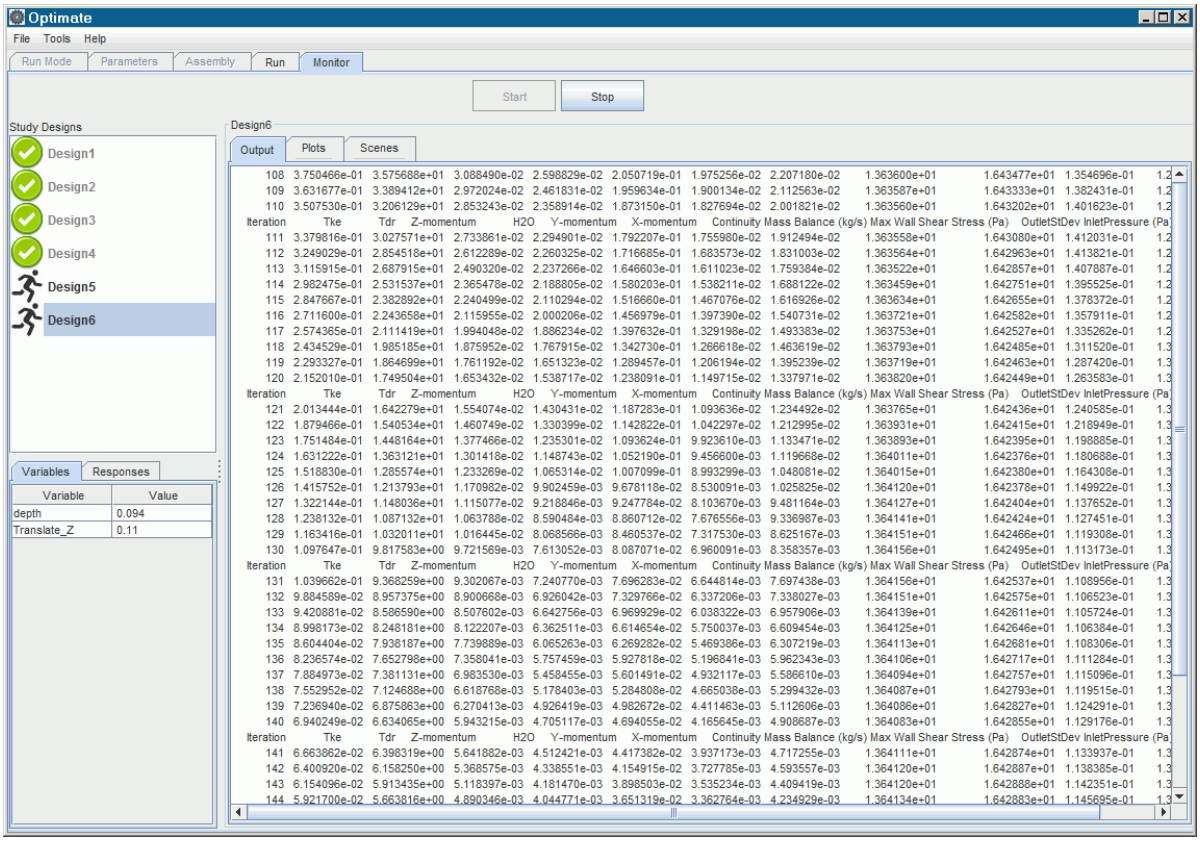
a) In the *Run* tab, from the ribbon, click (**Run**).

The current STAR-CCM+ server closes and the jobs are submitted in the background. The output from Optimate is displayed in the *Output* window. The STAR-CCM+ servers for each job are visible in the STAR-CCM+ server list. All necessary files for this study are created. A pop up window appears while the study is building.



b) To monitor the status of the designs in the study, switch to the *Monitor* tab and click **Start**.

While monitoring the study you can connect to individual designs and look at the simulation data. See Monitoring Designs.



c) Launch HEEDS Post to post-process the results from the study, return to the *Run* tab and click (**Open HEEDS|Post**).

12. Post-process the study results in HEEDS Post.

Contents:

Run Mode Options Reference
Variables Reference
Responses Reference
Scenes and Plots Reference
Common Menu Items
Assembly Reference
Run Reference

Run Mode Options Reference

The run options for the various run modes are provided below.

Design Sweeps Run Options

Use Design Set	Activate this option to define variable values using a `.csv` file. For more information, see Using Design Sets.

Design of Experiments Run Options

Sensitivity and Screening	
2 Level Full Factorial	Carries out a two-level full factorial DoE (Design of Experiments) study. See Design of Experiments: 2 Level Full Factorial Runs.
3 Level Full Factorial	Carries out a three-level full factorial DoE (Design of Experiments) study. See Design of Experiments: 3 Level Full Factorial Runs.
Response Surface Modeling	
Latin Hypercube	Uses the Latin Hypercube sampling option to generate a response surface for the design space. See Design of Experiments; Latin Hypercube.
Use Existing Designs	Allows you to create a response surface that fits a design set specified by you. The designs are specified through a `.csv` file which you can export through HEEDS Post. To export a set of designs from HEEDS Post, open the design table and select **Save Plot Data**. See Design of Experiments: Using Existing Designs.

Optimization Using SHERPA Run Options

Weighted sum of all objectives	Specifies a study where all objectives are combined into a single performance function that is defined as: $$\sum_{i=1}^{N_{obj}} \frac{LinWt_i \cdot Sign_i \cdot Obj_i}{Norm_i} - \sum_{j=1}^{N_{con}} \frac{QuadWt_j \cdot ConViol_j^2}{Norm_j^2}$$ where: • N_{obj} is the number of objectives in the optimization study. • $LinWt_i$ is the linear weight for the ith objective. • $Sign_i$ is the sign for the ith objective. The value is -1 for objectives being minimized and +1 for objectives being maximized. • Obj_i is the response value for the ith objective for that design. • $Norm_i$ is the normalizing value for the ith objective. • N_{con} is the number of constraints in the optimization study. • $QuadWt_j$ is the quadratic weight for the jth constraint. This value is 10000.0. • $ConViol_j$ is the amount by which the jth constraint is violated. This value is 0.0 if the constraint is met. • $Norm_j$ is the normalizing value for the jth constraint. Single objective studies use the SHERPA algorithm. See Optimization Using SHERPA.
Multiple objective tradeoff study (Pareto front)	Specifies a study where each objective is independently considered by the algorithm. Multi-objective studies use the MO-SHERPA algorithm. See Optimization Using SHERPA.

	Archive size	Specifies the number of designs in a single cycle of a Pareto optimization study. This value also determines the number of rank 1 designs the algorithm should try to find during the search in order to resolve the Pareto front. This value must be evenly divisible by four and greater than or equal to 12. This value should also be greater than or equal to the number of variables in the study. As an example, a Pareto optimization study with 8 variables would have an archive size of 12, while one with 13 variables would have an archive size of 16.
Random seed	Specifies the seed value for the optimization study. This feature is useful if you want to reproduce the same results from an optimization study. Click to randomise the seed value. The seed value is stored in the Optimate project file.	

Variables Reference

Adjust the variable properties according to the type of study you want to perform.

Optimate can access a number of simulation objects in STAR-CCM+ and make changes to their properties or settings. For more information on what STAR-CCM+ objects can be selected as Optimate variables, see Optimate Variables.

Variable Common Properties

The following properties are available for all run modes.

Name	Specifies the name of the Optimate variable. By default, *Name* is set to the name of the corresponding STAR-CCM+ object. Changing the Optimate variable *Name* does not change the STAR-CCM+ object that is used to define the variable.	
Enabled	Allows you to disable and re-enable a variable. You can permanently delete a variable using the variable menu options.	
	Activated	The variable is active in the study.
	Deactivated	The variable is not used by Optimate.
Sim Object	Displays the corresponding STAR-CCM+ object (Read-only).	

The following properties are available for all run modes except Stochastic Analyses and analyses that use design sets. For more information, see Using Design Sets.

Type	Allows you to specify whether the variable value is continuous, discrete, or constant.	
	Continuous	Allows Optimate to set the variable to any value within a specified range. You specify the range by setting the *Min* and *Max* properties.
	Discrete	Instructs Optimate to set the variable to the values that you specify in the *List* property.

	Constant	Instructs Optimate to set the variable to the *Baseline* value, without exploring other designs.
Min	Specifies the minimum value that Optimate can set the selected variable.	
Baseline	Specifies the value of the corresponding STAR-CCM+ object in the baseline simulation. The distribution that is used to define the variables is centered in this value. The baseline is also the first design run in the study. See Running the Baseline Design.	
Max	Specifies the maximum value that Optimate can set the selected variable.	
Increment	Displays the increment size. When set, *Resolution* is automatically updated. This property is not available for Design of Experiments analyses.	
Resolution	Specifies the number of different values to look at for the selected variable. When set, *Increment* is automatically updated. This property is not available for Design of Experiments analyses.	
List	Specifies the list of values, separated with commas or spaces, that Optimate can use for the variable. This property is active when you set *Type* to **Discrete**.	

Variable Properties for Design of Experiments Analyses

For Design of Experiments analyses, the following properties are also available.

Mode	Specifies whether the property values are absolute, or relative to the baseline value.	
	Value	All property values are specified as absolute values.
	Percent	All property values are specified as a percentage of the *Baseline* value.
Low	Defines the lower value that Optimate uses for the selected variable. This property is only available when using the **2 Level Full Factorial** and **3 Level Full Factorial** run options.	
Mid	Defines the mid value that Optimate uses for the selected variable. This property is only available when using the **3 Level Full Factorial** run option.	
High	Defines the upper value that Optimate uses for the selected variable. This property is only available when using the **2 Level Full Factorial** and **3 Level Full Factorial** run options.	

Variable Properties for Stochastic Analyses

In Stochastic Analyses, you choose the type of distribution you want to define the variable values.

Mode	Specifies whether the property values are absolute, or relative to the baseline value.	
	Value	All property values are specified as absolute values.

	Percent	All property values are specified as a percentage of the *Baseline* value for Stochastic studies, and *Mid* for 2 level and 3 level full factorial DoE studies.
Baseline	Specifies the value of the corresponding STAR-CCM+ object in the baseline simulation. The distribution that is used to define the variables is centered in this value. The baseline is also the first design run in the study. See Running the Baseline Design.	
Distribution	Specifies the type of distribution that is used to define the variable values.	
	Normal	Specifies a Normal distribution centered in the *Baseline* value. The standard deviation is specified through the *St. Deviation* property.
	Triangular	Specifies a Triangular distribution centered in the *Baseline* value. The lower and upper limits of the distribution are specified through the *Min Variation* and *Max Variation* properties, respectively.
	Uniform	Specifies a uniform distribution. The maximum variation, in either direction, from the *Baseline* value, is specified through the *Max Variation* property.
St. Deviation	Specifies the standard deviation of the Normal distribution. This property is only available when *Distribution* is set to **Normal**.	
Min Variation	Specifies the lower limit for the Triangular distribution. This property is only available when *Distribution* is set to **Triangular**.	
Max Variation	For a Uniform distribution, specifies the maximum variation, in either direction, from the baseline value. For a Triangular distribution, specifies the distribution upper limit.	

Responses Reference

Adjust the response properties according to the type of study you want to perform.

Responses Properties

Response Name	Specifies the name of the Optimate response. By default, *Response Name* is set to the name of the corresponding STAR-CCM+ report.
Enabled	Allows you to disable and re-enable a response. You can permanently delete a response using the response menu. See Common Menu Items.

	Activated	The response is active in the study.
	Deactivated	The response is not used by Optimate.
Report Name	Displays the corresponding STAR-CCM+ report (Read-only).	
Source	Specifies whether data is imported from the simulation or from an Optimate RSM (Response Surface Model) file.	
	Simulation	The response source is the STAR-CCM+ simulation.
	RSM	The response source is a specified Optimate RSM (Response Surface Model) file. You specify the file through the **Import RSM Data** property in the *Response Properties* dialog.

Response Properties Dialog

The following properties are available when you right-click a response in the *Responses* group box and select **Properties**.

Minimum	Allows you to set a minimum value for the selected response. Designs that return response values below the *Minimum* are classed as error designs.
Maximum	Allows you to set a maximum value for the selected response. Designs that return response values above the *Maximum* are classed as error designs.
Import RSM Data	Allows you to specify a RSM (Response Surface Model) file that defines the response source. This option is only active when *Source* is set to **RSM**. Clicking **Import RSM Data** brings up the *Open* dialog, that allows you to select the desired RSM file.

Response Menu Items

You access the following options by right-clicking a response in the *Responses* group box. Common menu items are also available, see Common Menu Items.

Set as objective	Instructs Optimate to record the value of the response and to evaluate how the response value compares to previous designs. See Responses. The response is added to the objectives at the bottom of the *Parameters* tab. For more information on the properties available for objectives, see Objective Properties.
Set as constraint	Instructs Optimate to record the value of the response and to determine whether this design is feasible or not. The response is added to the constraints at the bottom of the *Parameters* tab. For more information on the properties available for constraints, see Constraint Properties.
Properties	Allows you to access additional response properties, that can be set using the *Response Properties* dialog.

Objectives Properties

Goal	Sets the analysis goal for the selected response.

	Minimize	Specifies that the goal is to minimize the value in the selected response.
	Maximize	Specifies that the goal is to maximize the value in the selected response.

Objective Properties Dialog

The following properties are available when you right-click an objective and select **Properties**.

Linear Weight	Determines how the objectives are combined to evaluate the performance of each design. Only use this property if there are multiple objectives in a single objective optimization study.
Baseline Normalization	Sets the normalization values automatically for each variable by using the baseline run.
Normalization Value	Specifies the normalization. Only use this property if there are multiple objectives. Set this property so that all normalized objectives are of a similar order of magnitude.

Constraint Properties

Lower Limit	Specifies a minimum value for the response. Any designs with a response value below this value is classed as infeasible.
Upper Limit	Specifies a maximum value for the response. Any designs with a response value above this value is classed as infeasible.

Constraint Properties Dialog

The following properties are available when you right-click a constraint and select **Properties**.

Upper bound normalization factor	
Use upper bound value	Instructs Optimate to use the value specified in the *Upper Limit* property.
Define	Defines a normalization factor for the upper bound.
Lower bound normalization factor	
Use lower bound value	Instructs Optimate to use the value specified in the *Lower Limit* property.
Define	Defines a normalization factor for the lower bound.

Scenes and Plots Reference

Optimate can record data from scenes and plots that are stored within the source simulation file. Optimate records the selected plots/scenes at the end of each analysis.

Scenes and Plots Properties

Name	Specifies the name of the scene/plot in Optimate. By default, *Name* is set to the name of the corresponding STAR-CCM+ plot/scene.	
Enabled	Allows you to disable and re-enable an active plot/scene.	
	Activated	The plot/scene is active.
	Deactivated	The plot/scene is not used by Optimate.
Plot or Scene	Displays the corresponding STAR-CCM+ plot/scene (Read-only).	

For common menu items, that you access when right-clicking a scene/plot, see Common Menu Items.

Common Menu Items

You access the following options by right-clicking an object in the *Parameters* tab.

The following menu items are available when you right-click a variable, response, plot, or scene.

Toggle Enabled	Activate/Deactivate the *Enabled* property for the selected objects.
Delete	Allows you to delete the selected objects from the current study.
Re-map to STAR-CCM+	Allows you to assign a variable, response, plot, or scene to a different STAR-CCM+ object.

The following menu item is available when you right-click a response, objective, constraint, plot, or scene.

Properties	Allows you to access additional properties using the object *Properties* dialog.

Assembly Reference

The Assembly panel contains the settings that control how the STAR-CCM+ jobs run.

Job-Related Options

These options control how the Optimate jobs are run, and provide information on the job.

Simultaneous Jobs	Specifies how many jobs are run at the same time. When running an Optimization study, this value must not be greater than the value specified in *Number of Runs* divided by 10. If computational and license resources permit, set this value to 1/5 to 1/7 of the value set in *Number of Runs*.
Cores per Job	Specifies how many cores to run with each STAR-CCM+ job.
Number of Runs	Specifies the total number of cases to run for the project. For Design Sweeps, the number of runs is determined by the number of variables and how many values are used for each variable. For single-objective Optimization studies, run at least 30 to 35 cases. As a guideline, set this value to 10 times the number of variables in your study. For multi-objective Optimization studies, run at least 40 cases, which is the default value. As a

	guideline, for multi-objective Optimization studies, set this value to 8 to 12 times the archive size. If the archive size is larger than 20, the number of runs value is multiplied by 2 to give 40 runs. The SHERPA algorithm works best when it can learn as many times as possible. For two-level full factorial or three-level full factorial runs, the number of runs is determined by the algorithm and is populated automatically.	
Tokens Required	Displays how many Power Tokens are required to run a project with the current settings. This value is determined automatically by multiplying the number of simultaneous jobs by the number of cores per job. If you are using a cluster, this value does not reflect the correct number of tokens you require unless you set *Cores per STAR-CCM+ Job* to reflect the command you provide in the queue configuration window. This value is only relevant if you activate **Use Power Tokens** under Licensing Options. See Licensing Options.	
Optimate Save Option	Specifies what designs to save.	
	All Designs	Instructs Optimate to retain all design directories.
	All Best Designs	Instructs the Optimate solver to remove all design directories except for those that are identified as the best design during the history of the project run. For example, the first run of an optimization project is always identified as the first best design, however, as subsequent jobs run, the Optimate solver identifies improved designs and the first design is no longer classed as the best design. Activating this option retains the directory for the first design because it was identified at one point as the best design. This option is only available when running optimization studies.
	Last Best Design	Instructs the Optimate solver to remove all design directories except for the one that is identified as the best design (based on the specified objectives). This option is only available when running optimization studies.
	None	Instructs Optimate to remove all design directories after the outputs of a completed simulation (report values, plots, and scenes) are extracted.
Macro Insertion Point	Specifies the insertion point for custom Java macros into the `StarDriver.java` macro that is used to execute each STAR-CCM+ simulation. The custom macros are listed a panel below this option. This list updates according to what option you select for *Macro Insertion Point*. See Customizing the Workflow Using Java Macros and File Descriptions.	
	Before Meshing	Displays the custom macros that are selected to be inserted immediately before executing the meshing pipeline.
	After Meshing	Displays the custom macros that are selected to be inserted after generating the simulation mesh, but before running the solver.
	After Running	Displays the custom macros that are selected to be inserted immediately after the solver has been completed, and before any results (reports, scenes, and plots) are returned to the Optimate solver.

STAR-CCM+ Run Options

The STAR-CCM+ run options specify how STAR-CCM+ jobs are submitted, executed, and saved.

Mesh Mode	Controls whether the design is remeshed.	
	AUTO	Checks the variables and decides whether a remesh is required. If the variables make a direct change to the geometry or a mesh parameter, a remesh is required.
	FORCE	Forces a remesh. Sometimes you have a field function or macro that affects the geometry or mesh, however Optimate does not detect that a remesh is required. If you know that your design requires a remesh, use this option to force the remesh.
	NONE	Does not remesh the geometry.
Remove Invalid Cells	Instructs STAR-CCM+ to remove invalid cells on each individual job that is submitted. It is recommended that you activate this option if the mesh settings in the source simulation file occasionally generate invalid meshes. Invalid meshes are more likely to occur for projects where widely varying geometries are generated and when you use surface wrapping on complex geometry.	
Clear Solution	Specifies whether to clear the solution in STAR-CCM+ before running each study. Activate this option if you want to clear the current solution before running each job. Deactivate this option if you want to use the solution contained in your simulation file as a starting point for the next job, which can significantly reduce the number of iterations that are required to reach a converged solution. If you are not clearing the solution, make sure that you have appropriate stopping criteria set up for the study.	
Compute Resource	Specifies the compute resource to use for the study. A compute resource contains the properties of a cluster or remote machine. See Creating a Compute Resource. A compute set allows you to use multiple compute resources together if they are able to execute the same command to launch STAR-CCM+. See Using Multiple Compute Resources Together.	

Licensing Options

The Optimate licensing options control how licenses are checked out for the study. During run time, Optimate jobs require an Optimate or Optimate+ license in addition to the necessary licenses for starting the STAR-CCM+ jobs. You can use existing STAR-CCM+ license features in addition to Power Tokens. Power Token licensing offers the flexibility of launching multiple simultaneous designs while only requiring a single `ccmpsuite` feature license.

Use Power Tokens	Instructs Optimate to check out Power Tokens for each STAR-CCM+ job based on the number of cores each job will be run on. This option is the default licensing scheme for Optimate. When this option is activated, Optimate checks out a single `ccmpsuite` feature license for the duration of the study. Note that if you are using Power Tokens in addition to other license types, such as serial and HPC, or Power Sessions, this `ccmpsuite` feature license is additional to any other feature licenses that are required by individual designs being run using these other licensing schemes.

Use Serial and HPCs	Allows Optimate to check out STAR-CCM+ session licenses and HPC licenses for each STAR-CCM+ job.
Use Power Sessions	Allows Optimate to check out Power Session licenses for each STAR-CCM+ job.
Use Power on Demand	Allows Optimate to check out a Power Session license from the CD-adapco Power-on-Demand server. Activating this option brings up a dialog where you are prompted for a Power-on-Demand key. The license server is set to `1999@flex-cd-adapco.com`. You can change the server port and address by modifying the `Process1.in` file after building the project. See File Descriptions.

Directory Options

The directory options specify the location where the study files are stored. You can also specify the location where you have stored your custom macros and CAD geometry files.

CAD Directory	Specifies the relative path to the directory that contains the CAD parts. This option is only available if you have created parts variables in the *Variables* tab and want to swap parts during the project. By default, the directory is set to the same directory as the source simulation file. If your CAD parts are stored in a different directory, update the path to point to the correct directory. You can specify the directory by browsing the local file system or manually if the run directory is not currently mounted. Relative paths are acceptable when manually setting this directory. Relative paths are acceptable when manually setting this directory.
Specify Custom Study Directory	Activating this option allows you to specify a custom directory where the study files are stored.
Custom Study Directory	Specifies the custom directory for the study.

Run Reference

The Run panel contains the options to run Optimate.

Open HEEDS|Post	Launches HEEDS Post for post-processing the Optimate study results. See Post-Processing Optimate Results.
Run	Runs the Optimate study.
Run Baseline	Runs a single design using the baseline values. See Running the Baseline Design.
Run CAD Robustness	Runs a CAD Robustness study to check that a valid geometry can be generated with the specified variables and study settings. This study identifies how frequently the CAD geometries fail to generate. The CAD Robustness study creates a DoE (Latin hypercube) of the CAD parameters, where the number of designs in the study is determined by the number of variables and the sampling intensity. Running the study updates the CAD parameters and checks the resulting geometries for validity—no meshing or solving is performed.

	The study runs on your local machine—even if you set the compute resource for the study to a cluster or a remote machine. The results of the study are saved in a sub-directory (`CAD_Robustness`) of the study's directory. Scenes that are listed as model views are saved displaying the geometry representation. See Running a CAD Robustness Study.
Stop	Stops the Optimate study. Stopping an Optimate study does not kill the STAR-CCM+ jobs that are already running.
Pause/Resume	Pauses the Optimate study. Resumes the study from the paused stated.
Restart	Restarts the Optimate study.
Unlock Study	Unlocks the Optimate study to allow you to modify the study parameters. Each study is locked when you run the Optimate solver. See Modifying Project Settings After Running a Job.
Share Designs	Allows you to share your own geometry designs with the Optimate solver. See Sharing Designs.
Print Message File	Prints the contents of the Optimate solver message file, `Agent_Group_1.mes`, to the Optimate Output window. This file contains information about the Optimate job and is useful for debugging jobs that fail to start.
Command	Displays the command that Optimate submits when you run the solver. The information displayed in this window is read-only.
Output	Displays information about the Optimate project. Information about the project is displayed in this window when you build the project and run the Optimate solver.

Saving and Loading Optimate Projects

The Optimate study settings are stored in an external file, called the Optimate project (`.optm`) file.

You can save or load the Optimate project file.

Saving the Optimate Project

To save an Optimate project:

- In Optimate, select **File** > **Save**.
- In the *Save* dialog, enter the name and location where you want to save the project.
- Click **Save**.

To save a copy of the project:

- In Optimate, select **File** > **Save As**.
- In the *Save* dialog, enter the new name and location.
- Click **Save**.

Loading an Optimate Project

To load an Optimate project:

- Load STAR-CCM+ and launch Optimate.
- In the *Load Study?* dialog, click **Open an Existing Optimate Study**.

Post-Processing Optimate Results

Once a series of jobs have completed, you can post-process the results in HEEDS Post. If you want to have a look at a particular design in more detail, it is recommended that you perform the post-processing in STAR-CCM+.

STAR-CCM+

If you choose to save all simulation files during your Optimate analysis, then the individual simulation files are stored in the following directories:

```
star_0/DesignN/star/
```

Where `N` is the design ID for each job. Each simulation file is appended with a three digit *design ID* so that you can copy them into the same directory for archiving, but can still reference the ID for post-processing in HEEDS Post. For more information on how to save all simulation files, see Assembly Panel.

HEEDS Post

HEEDS Post is a tool for post-processing series of jobs. You can launch HEEDS Post from within the Optimate environment.

To launch HEEDS Post:

- Launch Optimate and load your project file.
- In the *Run* panel, click (**Open HEEDS|post**).

  The active study in Optimate is loaded into HEEDS Post.

If you move the study to a new folder, you must reset the association in HEEDS Post.

To re-associate the study folder:

- Right-click the study name in the tree, select **Find Study Folder**.
- Navigate to the study folder and click **Accept**.

Launching HEEDS Post Using a Shortcut

If you use HEEDS Post often, you can create a shortcut to the HEEDS Post executable and launch it directly with the shortcut, without having to launch Optimate first.

To create a shortcut to HEEDS Post:

- Navigate to `[INSTALL_DIR]\optimate\[VERSION]\Win64\`.
- Right-click the `HEEDSPost.exe` executable and select **Send to** > **Desktop (create shortcut)**.

  A shortcut appears on your desktop.
- Double-click the shortcut to launch HEEDS Post.
- In HEEDS Post, select **File** > **Open** and select the desired Optimate project file (`*.optm`).

Customizing the Optimate Workflow Using Java Macros

You can customize the Optimate workflow using Java macros to perform additional actions in the STAR-CCM+ simulation.

When you run the Optimate solver, it runs through a set of default steps which are defined in the `StarDriver.java` macro. You can expand the workflow by adding custom macros to the Optimate study. This modifies the `StarDriver.java` macro to reflect your custom workflow. There are three areas where you can expand the workflow: before meshing, after meshing, and after running.

One example of how you can customize the workflow is to change the run settings and then re-run the solver. You can record a macro in STAR-CCM+, where you modify the run settings, for example the mach number, extend the number of iterations in the stopping criteria, and then re-run the solver. Using this technique you can perform two runs with different mach numbers for a single design. You then insert this macro into the `StarDriver.java` macro after running the solver. When Optimate reaches this stage, it performs the custom steps and then proceeds with the remaining default workflow.

To customize the workflow:

1. In STAR-CCM+, record the steps that you want to include in the custom workflow.
2. In Optimate, set up your study parameters.
3. In the *Assembly* tab, set *Macro Insertion Point* to the option that is relevant to your situation:
   - **Before Meshing** — select this option if you want to perform the actions immediately before executing the meshing pipeline.
   - **After Meshing** — select this option if you want to perform the actions immediately after executing the meshing pipeline.
   - **After Running** — select this option if you want to perform the actions immediately after the STAR-CCM+ solver completes, and before any results are returned to the Optimate solver.
4. Click **Browse** next to *Macros Insertion Point*.
5. In the *Select STAR-CCM+ Macros to be Inserted* dialog navigate to the directory where you stored the custom macros and select them.

   The macros are added to the macro insertion portion of the panel. Only macros for the selected insertion point are displayed. To view the macros from the other insertion points, switch to the appropriate *Macro Insertion Point*. If you have multiple macros in the list, you can reorder them using the up and down arrows.

The custom macros are added to the relevant part of the `StarDriver.java` macro. When you run the Optimate study, the custom steps are run at the appropriate stage of the workflow.

Using Design Sets

When performing a Design Sweeps study or a Design of Experiments study, you can define the Optimate variables using tabular data. You can either enter the tabular data manually or import a `.csv` file.

An example format for the `.csv` file data is provided below:

```
InletPressure, OutletStDev, depth, TranslateZ
1.1737733459e+02, 4.4361197948e-01, 4.0000000000e-02, 0
1.0233065796e+03, 1.3013350964e-01, 8.8000000000e-02, 2.0000000000e-03
```

Note: The names of the columns must match the names of the variables that you want to define using this file.

To instruct Optimate to use `.csv` file data to define variables:

1. In the *Run Mode* tab:
   - If *Run Mode* is set to **Design Sweeps**, activate the **Use Design Set** run option.
   - If *Run Mode* is set to **Design of Experiments**, activate the **Use Existing Design Set** run option.
2. In the *Parameters* tab, create the variables that you want to define. See Optimate Workflow.
3. In the *Parameters* tab, click **Manage Design Sets**.

   The *Select a design set for the study* dialog appears.

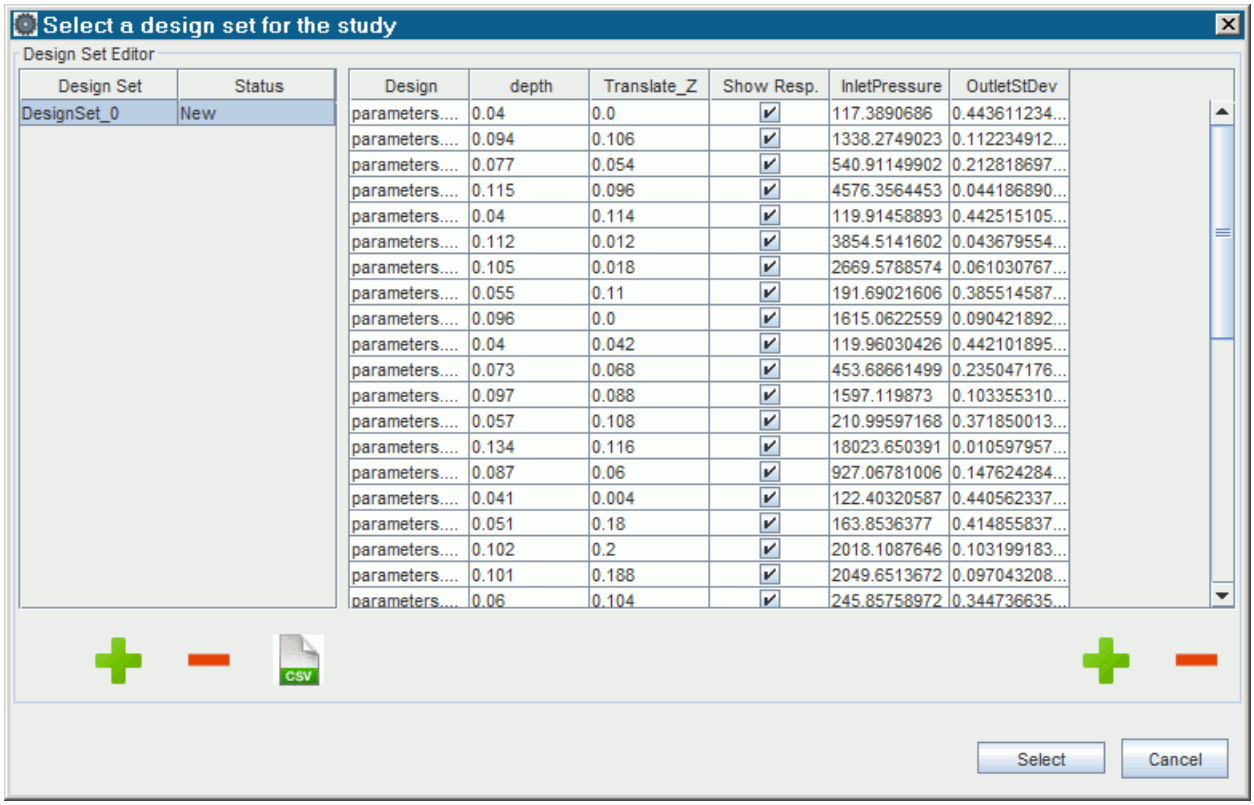
4. In the dialog, click (**Add Design Set**) to create a design set.

For the design set that you created, you can enter designs manually or you can import designs from a .csv file. To enter designs manually:

5. Click (**Add Design**) and enter the appropriate values for each variable.
   You can add further designs using the same technique.

To import designs as part of .csv file:

6. Click (**Import from CSV**).
7. In the *Open* dialog, navigate to the desired `.csv` file. The first line of the `.csv` file must contain the name of the columns.
   If you do not wish to import specific lines of your `.csv` file, skip them by entering the corresponding row numbers, separated by a space, in the *Lines to skip* property.
8. In the *Open* dialog, click **Import**.



9. Click **Select**.

   All the variables are updated to use data from the design set.

Running Optimate Jobs

You can run Optimate jobs locally, on a remote machine, interactively, or in batch mode.

Contents:

Optimate Environment Variables
Running Local Jobs in Batch
Running Optimate Studies on a Cluster
Modifying Project Settings After Running a Job
Sharing Designs
Running the Baseline Design

Optimate Environment Variables

Optimate requires certain environment variables in order to run correctly.

When you run Optimate from the interface, these variables are automatically created for you. If you run Optimate from the command prompt, you must create and set these variables manually.

Optimate Environment Variables for Linux Platforms

On Linux platforms, set the following variables:

- `PYTHONHOME=[STAR-CCM+_INSTALL_DIR]/optimate/[VERSION]/Python27`
- `HEEDS_ROOT=[STAR-CCM+_INSTALL_DIR]/optimate/[VERSION]/LX64/solver`
- `LD_LIBRARY_PATH=[STAR-CCM+_INSTALL_DIR]/optimate/[VERSION]/LX64:[STAR-CCM+_INSTALL_DIR]/optimate/[VERSION]/LX64/solver`
- `PATH=$PATH:[STAR-CCM+_INSTALL_DIR]/optimate/[VERSION]/LX64`

Optimate Environment Variables for Windows Platforms

On Windows platforms, set the following variables:

- `PYTHONHOME=[STAR-CCM+_INSTALL_DIR]\optimate\[VERSION]\Python27`
- `HEEDS_ROOT=[STAR-CCM+_INSTALL_DIR]\optimate\[VERSION]\Win64\solver`
- `PATH=%PATH%;[STAR-CCM+_INSTALL_DIR]\optimate\[VERSION]\Win64`

Running Local Jobs in Batch

You can run submit jobs in batch by executing the Optimate solver from the command prompt.

To run a job in batch:

1. Make sure that the environment variables that Optimate requires are set. See Optimate Environment Variables.

2. Open a command prompt.
3. Change to the directory that contains the baseline STAR-CCM+ simulation file and the study files.
4. Enter the following command:
   - For Windows:

     ```
     > [STAR-CCM+_INSTALL_DIR]\optimate\[VERSION]\Win64\solver
     \Optimate.exe i=Agent_Group_1.in overwrite skipEvalCheck
     ```

   - For Linux:

     ```
     % [STAR-CCM+_INSTALL_DIR]/optimate/[VERSION]/LX64/solver/
     Optimate i=Agent_Group_1.in overwrite skipEvalCheck
     ```

5. If the previous Optimate job did not exit correctly, delete the lock file named `star0.lock`.
6. Change the extension of `UserDesigns0_initialze` file from `.in_backup_at_[n]_evals` to `.in`.
7. Execute the command again.

Contents:

Resuming Jobs
Extending Jobs

Resuming Jobs

If a job aborts for any reason, you can resume it where it picks up from the last completed design.

You can only resume a job from the command prompt and not via the Optimate interface.

To resume a job:

1. Make sure that the environment variables that Optimate requires are set. See Optimate Environment Variables.
2. Type the following in the command prompt:

   ```
   [STAR-CCM+_INSTALL_DIR]/optimate/Ver2014.06/LX64/solver/Optimate
   i=Agent_Group_1.in useHistory skipEvalCheck
   ```

   The study continues from the last completed design.

Extending Jobs

If a job completes but further designs are needed to achieve the desired results, you can extend the job.

You can only resume a job from the command prompt and not via the Optimate interface.

To extend a job:

1. Locate the `Agent_Group_1.in` file in your study directory.
2. Open `Agent_Group_1.in` in a text editor of your choice.

3. Locate the `*STUDY` tag and modify the `mode` property:

```
mode: extend
```

4. Locate the `*RUN_OPTIONS` tag and modify the `overwriteResults` property:

```
overwiteResults: no
```

5. Locate the `*AGENT` tag and modify the `numEvalsTotal` and `numEvals` properties. For example, if your previous job contains 20 designs but you require 10 additional designs, set the following values:

```
numEvalsTotal: 30
```

```
numEvals: 10
```

6. Save the file and close it.
7. Make sure that the environment variables that Optimate requires are set. See Optimate Environment Variables.
8. In your command prompt, enter the following command to extend the job:

```
[STAR-CCM+_INSTALL_DIR]\optimate\Ver2014.06\Win64\solver
\Optimate.exe i=Agent_Group_1.in
```

Running Optimate Studies on a Cluster

There are two methods that you can use to run an Optimate study on a cluster. In both methods, the STAR-CCM+ solver runs on the cluster. The difference is where the Optimate solver executable is run.

Optimate requires a queuing system to be present on the cluster in order to run. Queuing systems offer you the ability to run Optimate on a flexible number of cores. Machine files are not supported at this time.

Running the Optimate Solver on a Cluster

With direct execution, you run the Optimate solver executable on the cluster head node, or any compute node which can submit jobs to other nodes. This method is useful if your local machine restarts or shuts down during the night, or if you do not have access to a password-free SSH tunnel to a cluster. Optimate is installed together with STAR-CCM+ on the cluster, meaning that no additional installation or configuration is required to run an Optimate job on a cluster.

Running the Optimate Solver on the Local Machine using Remote Execution

With remote execution, you run Optimate on the local workstation and transfer data to and from STAR-CCM+ as it runs in the remote environment. Use remote execution when you have large cases that require STAR-CCM+ to run on a remote workstation or cluster.

To perform a remote execution, make sure that you satisfy the following requirements:

- You must have a password-free SSH tunnel set up between the local machine and the cluster. You must be able to directly access the head node (or a compute node that can submit jobscan on a cluster. For detailed instructions on how to set this up, contact your dedicated support engineer.

- You cannot restart or shut down the local machine during the study.
- All files must exist on the local machine so that they can be transferred over to the remote machine as necessary. Consequently, the script that launches Optimate on the remote workstation must also be in the local directory where you build the project.

Contents:

Creating the Run Script for Submitting Jobs to a Linux Cluster
Creating a Remote Profile
Creating a Compute Resource
Using Multiple Compute Resources Together
Running the Optimate Solver on a Linux Cluster
Submitting Optimate Jobs to a Linux Cluster with Remote Execution
Running the Optimate Solver on a Windows Cluster
Submitting Optimate Jobs to a Windows Cluster with Remote Execution
Manage Compute Resources Panel Reference

Creating the Run Script for Submitting Jobs to a Linux Cluster

When setting up a generic queue case, Optimate creates an additional file named `optimate.txt`. This file contains an ID flag that must be passed to STAR-CCM+ to access the Power Token licensing scheme. You must incorporate this ID into the run script that is used to call STAR-CCM+.

An example of a c-shell run script is provided below:

```
#!/bin/tcsh
set UUID=`cat optimate.txt`
starccm+ -doeuuid $UUID -batch StarDriver.java -np 2 test.sim >>&
test.log
```

Where:

- `set UUID=`cat optimate.txt`` creates the variable with the licensing information that is required to access the Power Token licensing scheme.
- `-doeuuid $UUID` is the command line option that passes the variable to STAR-CCM+.
- `-batch StarDriver.java` is the batch script to run STAR-CCM+.

By default, the run script that Optimate calls is:

```
./ccmprun.csh
```

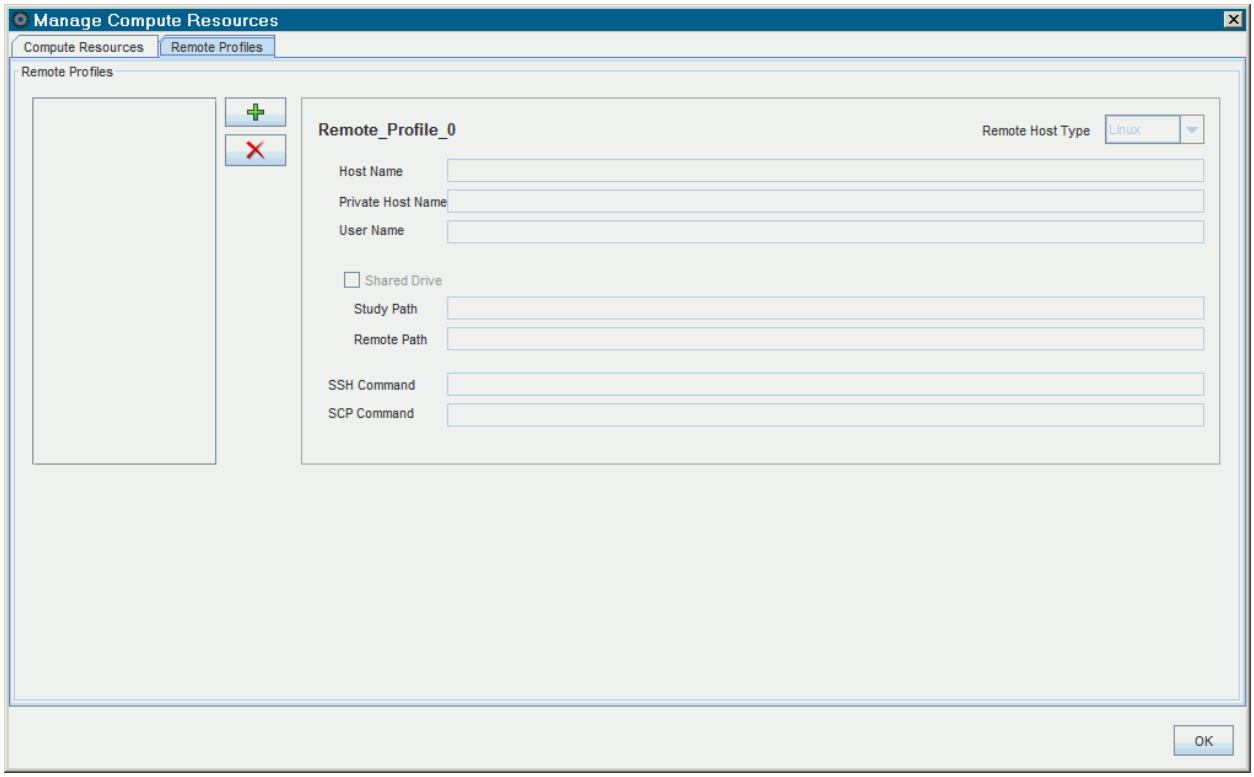
It is likely that you will append this script to include machine information that is related to the specific cluster you want to use. Also note that the command that executes STAR-CCM+ is set as a background process. This is necessary for the Optimate solver to function correctly.

Creating a Remote Profile

Before performing a remote execution, you must first create a remote profile. This profile stores all the information about the remote machine.

To create a remote profile:

1. In the Optimate environment, select **Tools** > **Manage Compute Resources**.
2. In the *Manage Compute Resources* dialog, click the **Remote Profiles** tab.



3. Click .
   A new entry is added to the group box on the left side of the panel.
4. To rename the remote profile, double-click the entry in the group box, enter a suitable name, and press <ENTER>.
5. Define the profile parameters. See Remote Profiles Panel.
6. Click **OK**.

Once you create a remote profile, it is stored so that you can use it for future studies.

Contents:

Remote Profiles Panel Reference

Remote Profiles Panel Reference

You can manage the remote host profiles from this panel.

Remote Host Profiles	Contains a list of active remote host profiles.

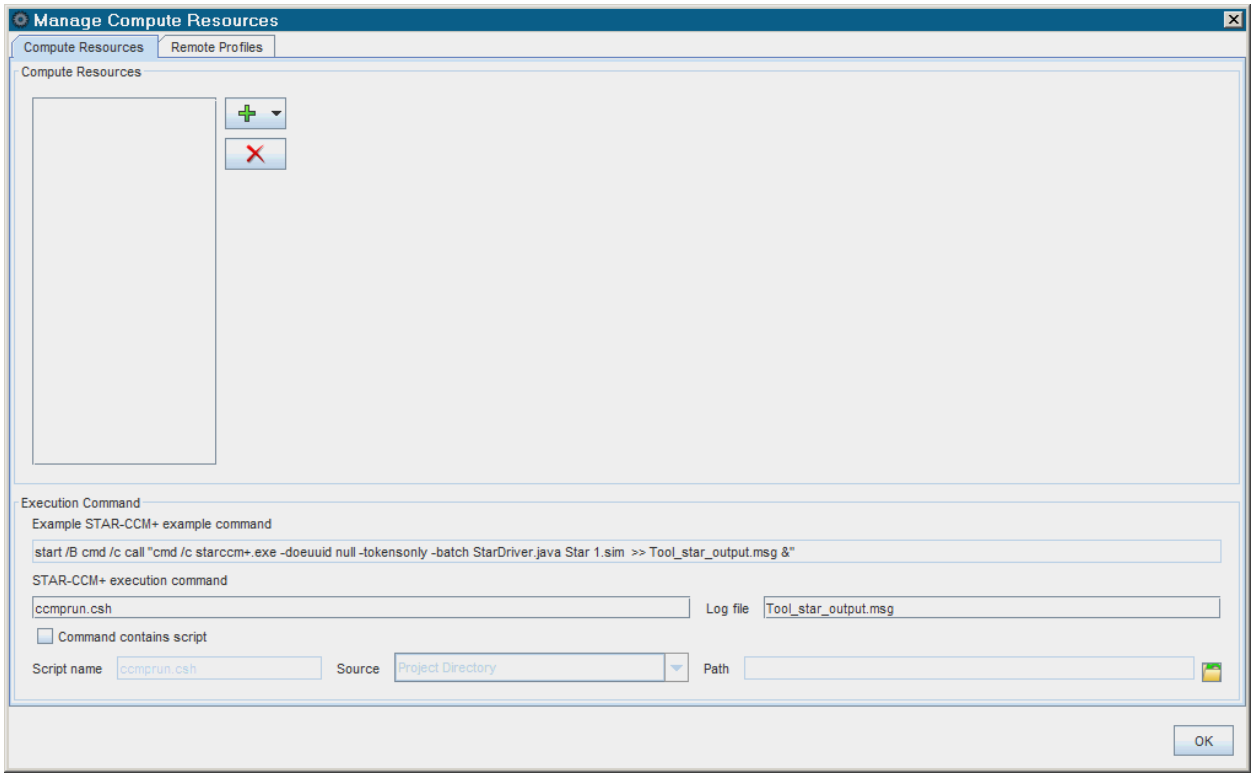
Remote Host Type	Specifies the operating system of the remote machine.
Profile Name	To specify the name to store for the remote profile in Optimate, double-click the remote profile in profile group box and enter a suitable name.
Host Name	Specifies the name of the remote host. If you require the domain name to connect from your local machine, include it here.
Private Host Name	Specifies the private host name with which to start a proxy server on the head node of the cluster. If a server starts on a compute node of a cluster, and you want to connect to this node through your local workstation, you can use the server on the compute node to start a proxy server to connect to the head node of the cluster. If the cluster head node is on a separate network to the compute nodes, you cannot start the proxy server using just the public host name (*Remote Host Name*). The private host name allows you to start the proxy server in this type of scenario.
User Name	Specifies the user name that is required to log into the remote machine. A password-free SSH must be set up to this user account on the remote machine.
Shared Drive	Specifies whether your local workstation shares the same drive with your remote machine. Activate this option if you can access the same files from both local and remote machines.
Study Path	When your local workstation does not share the same drive with your remote machine, that is, when you deactivate *Shared Drive*, Optimate makes a copy of the study directory on the remote machine. You can specify the location where Optimate includes the copy of the study directory, by providing an absolute path.
Remote Path	When your local workstation shares the same drive with your remote machine, that is, when you activate *Shared Drive*, this property allows you to specify the path that the remote machine uses to access the project on the shared drive.
SSH Command	The command that is used to SSH from the local machine to the remote machine.
SCP Command	The command that is used to SCP (copy files) from the local machine to the remote machine.

Creating a Compute Resource

Define and store the properties of a cluster or remote machine in a Compute Resource.

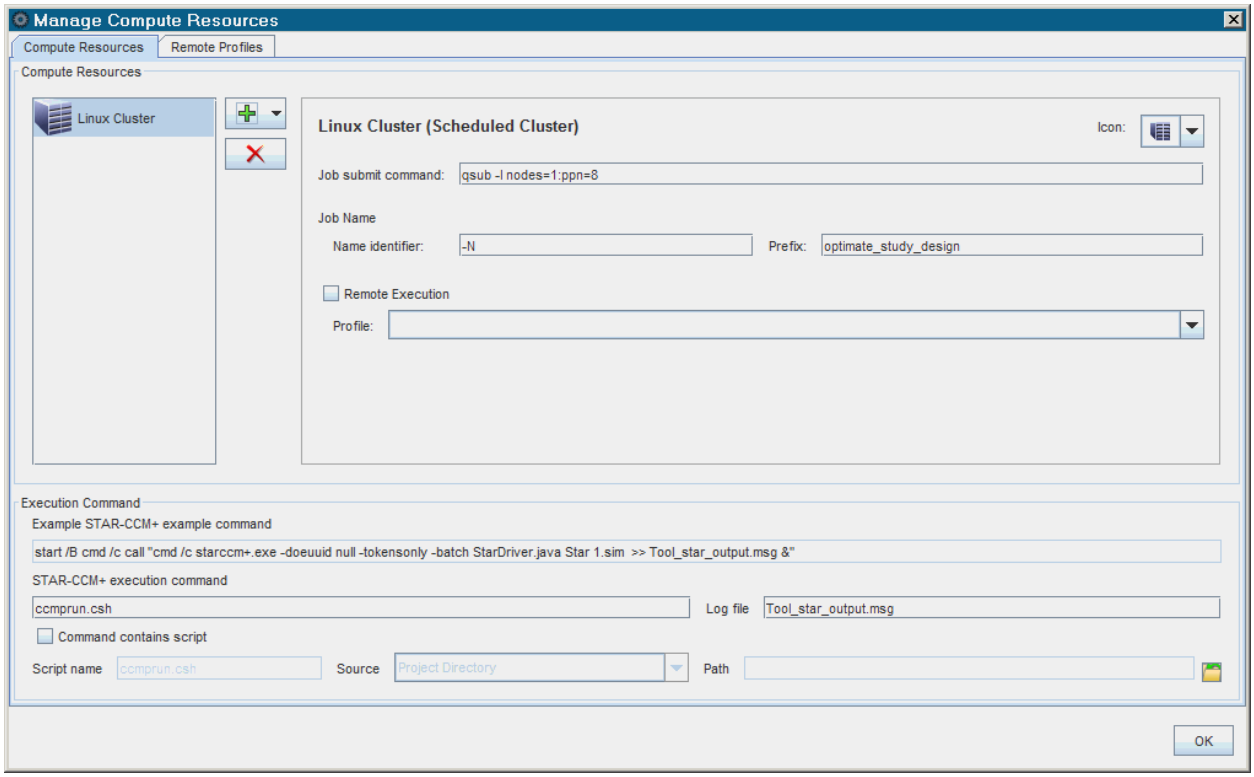
To create a Compute Resource:

1. In the Optimate environment, select **Tools** > **Manage Compute Resources**.
2. In the *Manage Compute Resources* dialog, make sure that the **Compute Resources** tab is displayed.



3. Click and select the type of resource that you want to add:
   - **Direct Cluster** — Resource to run using direct parallel execution, such as a local machine or workstation.
   - **Scheduled Cluster** — Resource to run using a generic scheduler, such as a Linux cluster.
   - **Microsoft HPC Cluster** — Resource to run using a Windows HPC cluster.
   - **Compute Set** — Resource that groups defined compute resources together so that you can use them in conjunction. See Using Multiple Compute Resources Together.

   A compute resource entry is added to the group box on the left side of the window. Properties that are specific to each compute resource are contained in the group box on the right side of the window. Properties that apply to all compute resources are contained in the *Execution Command* group box at the bottom of the window.



To remove an entry, select the resource from the group box, and click .

4. Double-click the compute resource entry, enter a suitable name, and press <ENTER>.
5. To customize the icon that appears next to the compute resource, expand the drop-down list next to the **Icon** property in the right side of the dialog, and select a suitable icon.

6. Set the properties for the compute resource. See Manage Compute Resources Panel Reference.
7. Click **OK**.

The compute resources are defined and stored for future use.

Using Multiple Compute Resources Together

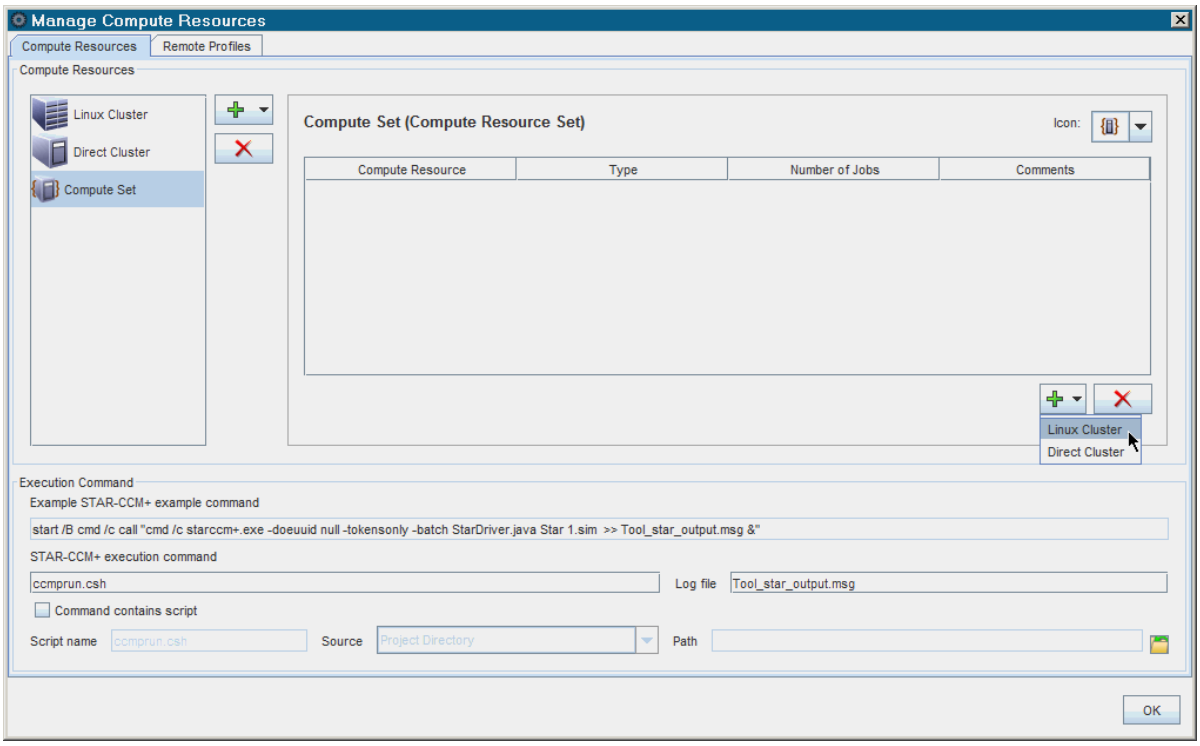
Group multiple Compute Resources into a Compute Set and use them in conjunction.

All Compute Resources in the Compute Set must be able to execute the same command to launch STAR-CCM+. Consequently, Compute Sets are not suitable for using Windows workstations with Linux workstations.

To create a Compute Set:

1. Create the Compute Resources that you want to add to the Compute Set. See Creating a Compute Resource.
2. In the *Manage Compute Resources* dialog, click and select **Compute Set**.
3. In the *Compute Set (Compute Resource Set)* group box:

   a) Click and select an existing Compute Resource that you want to add to the set.



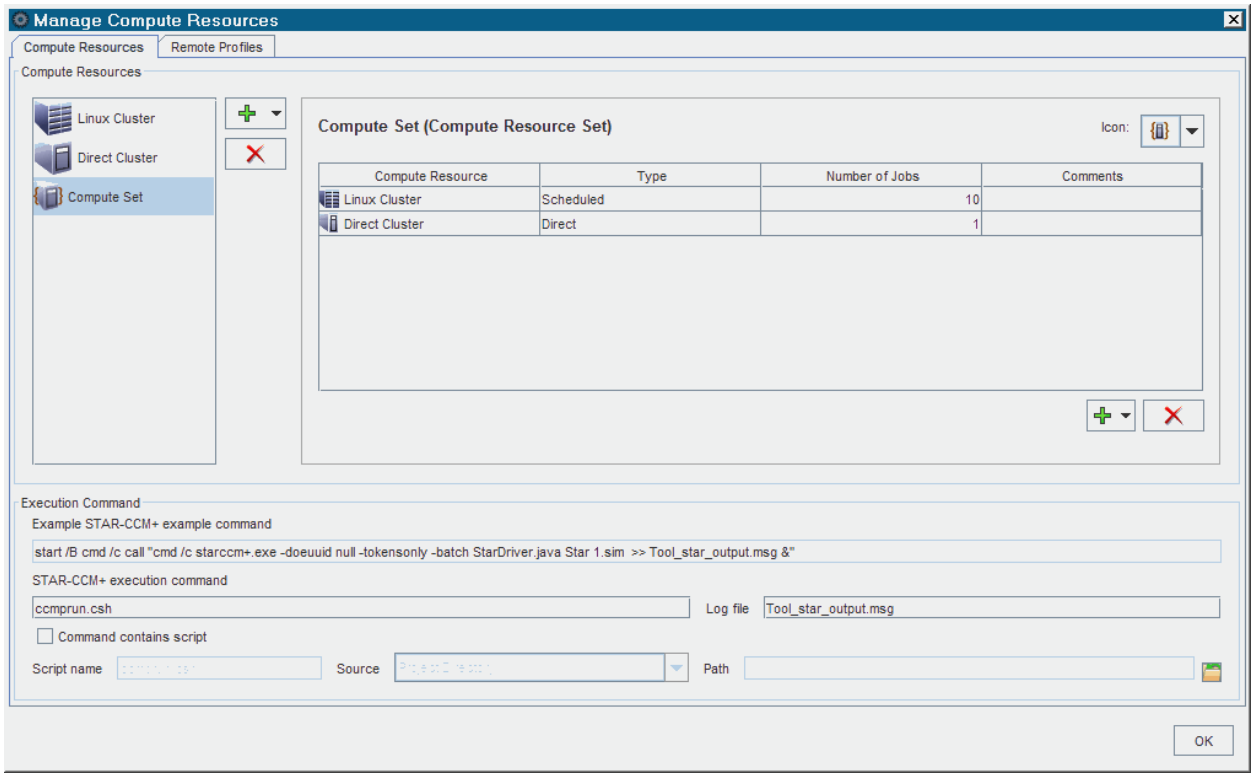
   b) Add the other Compute Resources to the Compute Set.

   c) Under the **Number of Jobs** column, specify the number of jobs that you want to run on each Compute Resource.

   d) Under the **Comments** column, enter any comments that are relevant to each Compute Resource.

   e) Set the execution command for the Compute Set. See Manage Compute Resources Panel Reference: Executing a Command with Scripts.

   The *Execution Command* group box contains parameters that are common to all Compute Resources.



4. Click **OK**.

Running the Optimate Solver on a Linux Cluster

To run an Optimate study on a cluster with a generic queuing system:

1. On your local machine, launch STAR-CCM+ and set up your baseline simulation.
2. On your local machine, launch Optimate and set up your study. See The Optimate Workflow.

   Note: Do not run the Optimate solver from the user interface.

3. If you do not already have a Direct Cluster compute resource, create one and configure the STAR-CCM+ execution command.

   See Creating a Compute Resource and Manage Compute Resources Panel Reference: Using Direct Parallel Execution.
4. In Optimate, click the *Assembly* tab and set **Compute Resource** to the Direct Cluster compute resource that you want to use for the study.
5. Copy your run script into the study folder that you copied over to your run directory on the cluster.

   Make sure that you place the script in the folder that contains the study files, `Agent_Group_1.in`, `Agent_Group_1.mes`, etc.
6. Transfer the study folder and simulation file to your run directory on the cluster.
7. On the cluster, set the variables that Optimate requires. See Optimate Environment Variables

8. From the study directory on the cluster, submit the job:

```
[STAR-CCM+_INSTALL_DIR]/optimate/[VERSION]/LX64/solver/Optimate
i=Agent_Group_1.in overwrite skipEvalCheck
```

   Make sure that you enter the full path to the Optimate solver executable.

To stop a job running in batch mode:

9. In the terminal where you launched the job, press **<Esc>** and follow the instructions that appear in the pop up dialog.

Submitting Optimate Jobs to a Linux Cluster with Remote Execution

To submit an Optimate job to a cluster with a generic queuing system using remote execution:

1. On your local machine, launch STAR-CCM+ and set up your baseline simulation.
2. On your local machine, launch Optimate and set up your study variables and outputs. See The Optimate Workflow.
3. If you do not already have a remote profile, create one. See Creating a Remote Profile.
4. If you do not already have a Scheduled Cluster compute resource, create one. See Creating a Compute Resource.
   a) Activate the **Remote Execution** option and select your profile. See Manage Compute Resources Panel Reference: Remote Execution
   b) Configure the remaining parameters for the queuing system. See Manage Compute Resources Panel Reference: Using the Scheduled Cluster
5. In Optimate, click the *Assembly* tab and set **Compute Resource** to the Scheduled Cluster compute resource that you want to use for the study.
6. If you want to use custom macros in your study, place these macros in the location on the local machine where Optimate is running and select the macros from that location. Optimate then copies the files from the local machine to the execution directory for each design on the cluster.

   Make sure that the path displayed in the *Path* column points to the directory that contains the macros on the local machine.
7. Place the run script in the location on the local machine where Optimate is running and update the *Source* property in the *Manage Compute Resources* panel accordingly. See Manage Compute Resources Panel Reference: Executing a Command With Scripts.

   When setting up a generic queue case, Optimate creates an additional file named `optimate.txt`. This file contains an ID flag that must be passed to STAR-CCM+ to access the Power Token licensing scheme. Make sure that the run script incorporates this ID. See Creating the Run Script for Submitting Jobs to a Linux Cluster.
8. Click the *Run* tab and run the Optimate study.

   The study folder is generated on your local machine and the study files are sent to the run folder on the cluster. The STAR-CCM+ simulation files remain on the cluster, however the image files are sent back to your local machine where you can view the results and perform post-processing in real-time. The *Output* window displays information about the run, where you can keep track of the jobs locally.

Running the Optimate Solver on a Windows Cluster

To run an Optimate study on a cluster with a Microsoft queuing system:

1. On your local machine, launch STAR-CCM+ and set up your baseline simulation.
2. On your local machine, launch Optimate and set up your study. See The Optimate Workflow.

   Do not run the Optimate solver from the user interface.
3. If you do not already have a Microsoft HPC Cluster compute resource, create one and configure the STAR-CCM+ execution command.

   See Creating a Compute Resource and Manage Compute Resources Panel Reference: Using the Microsoft HPC Cluster.
4. In Optimate, click the *Assembly* tab and set **Compute Resource** to the Microsoft HPC Cluster compute resource that you want to use for the study.
5. Copy your run script from the cluster into the study folder that you copied over to your run directory on the cluster.

   Make sure that you place the script in the folder that contains the study files, `Agent_Group_1.in`, `Agent_Group_1.mes`, etc.
6. Transfer the study folder and simulation file to your run directory on the cluster.
7. On the cluster, set the variables that Optimate requires. See Optimate Environment Variables.
8. From the study directory, submit the job:

   ```
   [STAR-CCM+_INSTALL_DIR]\optimate\[VERSION]\Win64\solver
   \Optimate.exe i=Agent_Group_1.in overwrite skipEvalCheck
   ```

   Make sure that enter the full path to the Optimate solver executable.

To stop a job running in batch mode:

9. In the command prompt where you launched the job, press **<Esc>** and follow the instructions that appear in the pop up dialog.

Submitting Optimate Jobs to a Windows Cluster with Remote Execution

To submit an Optimate job to a cluster with a Microsoft queuing system using remote execution:

1. On your local machine, launch STAR-CCM+ and set up your baseline simulation.
2. On your local machine, launch Optimate and set up your study variables and outputs. See The Optimate Workflow.
3. Create a remote profile. See Creating a Remote Profile.
4. If you do not already have a Microsoft HPC Cluster compute resource, create one. See Creating a Compute Resource.
   a) Activate the **Remote Execution** option and select your profile. See Manage Compute Resources Panel Reference: Remote Execution.
   b) Configure the remaining parameters for the queuing system. See Manage Compute Resources Panel Reference: Using the Microsoft HPC Cluster.
5. In Optimate, click the *Assembly* tab and set **Compute Resource** to the Microsoft HPC Cluster compute resource that you want to use for the study.

6. If you want to use custom macros in your study, place these macros in the location on the local machine where Optimate is running and select the macros from that location. Optimate then copies the files from the local machine to the execution directory for each design on the cluster.

   Make sure that the path displayed in the *Path* column points to the directory that contains the macros on the local machine.
7. Place the run script in the location on the local machine where Optimate is running and update the *Source* property in the *Manage Compute Resources* panel accordingly. See Manage Compute Resources Panel Reference: Executing a Command With Scripts.
8. Click the *Run* tab and run the Optimate study.

   The study folder is generated on your local machine and the study files are sent to the run folder on the cluster. The STAR-CCM+ simulation files remain on the cluster, however the image files are sent back to your local machine where you can view the results and perform post-processing in real-time. The *Output* window displays information about the run, where you can keep track of the jobs locally.

Manage Compute Resources Panel Reference

Sets up the commands to run on a generic queuing system. The default values reflect the commands to run on a Linux cluster running OpenPBS. Modify the values to reflect the configuration that you plan on running.

There are two methods available.

Using Direct Parallel Execution

The options to run using direct parallel execution, Direct Cluster compute resource, are displayed below.

STAR-CCM+ execution command	Specifies the STAR-CCM+ command or the batch script that is used to launch STAR-CCM+ on the cluster. It is important to provide the shared path of the executable for the full installation of STAR-CCM+.
	This can be the command you use in a command prompt to submit to a job scheduler:
	`starccm+ -np 3 -batch StarDriver.java simulation.sim`
	Alternatively, you can use a scheduling tool and a script to submit a job. As an example, using OpenPBS you enter:
	`qsub -l nodes=1:ppn=8 ccmprun.csh`

Using the Scheduled Cluster

The options to run using a Scheduled Cluster, Linux Cluster compute resource, are displayed below.

Job submit command	Command that is used to submit each job on the cluster. The default value is set up for a cluster running OpenPBS where each job is submitted on one node running four cores per node.

	Note: Modifying the number of nodes per job and cores per node does not affect the number of Power Tokens displayed in the assembly tab. However, it does affect the number of Power Tokens that are required to run.
Job name identifier	Flag that specifies the name of a job being submitted to the queue. The default value is set up for a cluster running OpenPBS.
Job name prefix	Specifies the prefix to use for the job.
STAR-CCM+ execution command	Command that is given for each submitted job. By default each job is started by executing the contents of the shell script `./ccmprun.csh`. You must create this shell script for a specific machine. See Creating the Run Script for Submitting Jobs to a Linux Cluster.

Using the Microsoft HPC Cluster

The options to run using a Microsoft HPC cluster, Windows Cluster compute resource, are displayed below.

Job submit command	System command to put a job in the queue. You can use the default value, `job submit` in almost all cases.
Job name identifier	Command line argument to identify the name for each submitted job. You can use the default value, `/jobname:`, in almost all cases.
Job name prefix	The prefix of each STAR-CCM+ job that is launched. The full name of the job is this prefix, appended by the evaluation number that corresponds to that job.
STAR-CCM+ execution command	The STAR-CCM+ command or the batch script that is used to launch STAR-CCM+ on the cluster. It is important to provide the shared path of the executable for the full installation of STAR-CCM+.

Executing a Command with Scripts

If your execution command contains a script, set the options below.

Command contains script	Indicates that the execution command contains a run script. If part of the execution command is a run script, activate this option.	
Script name	Specifies the script name as used in the execution command.	
Source	Specifies the location of the script file. Choose one of the options below	
	Project Directory	Specifies that the script file is located inside the Optimate project directory. The project directory is one directory above the study directory. If you have not specified a custom study directory, the project directory contains the simulation file. If you are using a custom study directory, then the simulation is copied to the study directory, and the project directory is still one directory above the study directory.
	Study Directory	Specifies that the script file is located inside the Optimate study directory.

	Specified Path	Specifies that the script file is located inside the directory specified in the *Path* property.
Path	Specifies the full path to the directory containing the script file. This property is active when you set *Source* to **Specified Path**.	

STAR-CCM+ Log File

If you are running on a cluster, and you are using a script, then the script is responsible for deciding where to place the output from STAR-CCM+. If you want to use the Monitor functionality, specify the name of the log file in the *Log file* field. Make sure that you specify the correct log file name, otherwise Optimate recognizes successful designs as errors. If you do not specify the name of the log file, the study still runs, but you cannot connect to and monitor the progress of each design. Only the hourglass icon will be visible next to each design in the *Study Designs* group box.

See Monitoring Designs and Design Status Reference.

Remote Execution

The options to run using remote execution are displayed below.

Remote Execution	Toggles the remote execution command. When activated, select your profile from the **Profile** drop-down menu.
Profile	Specifies the profile to use when performing a remote execution. Profiles that you create in the *Remote Host Profiles Manager* dialog are displayed here. See Remote Profiles Panel Reference.

Modifying Project Settings After Running a Job

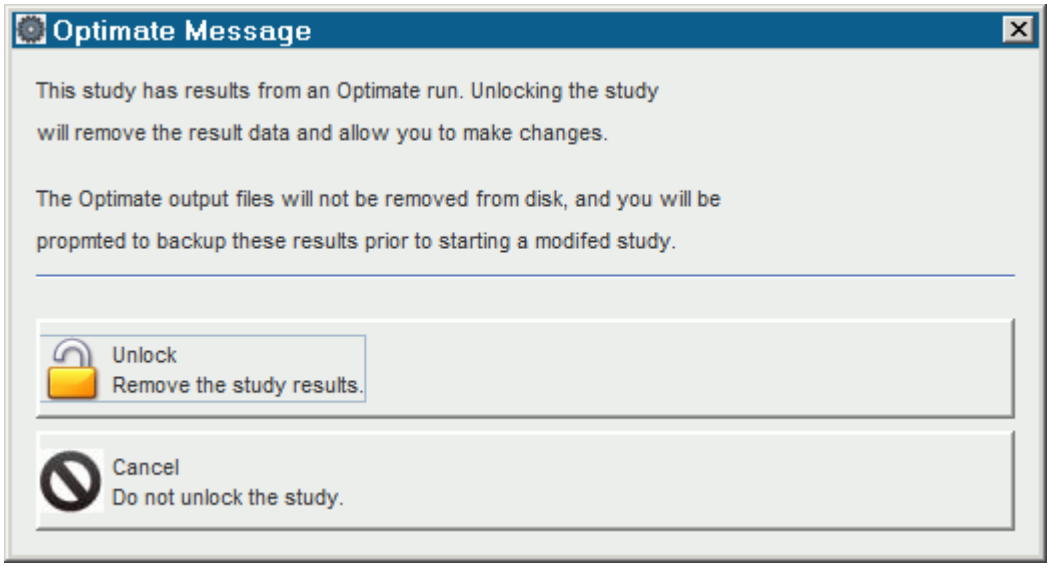
After running a study, the Optimate project settings are locked until you reload the project.

You can modify the study settings by unlocking the study. Unlocking a study removes the results from the current session, but not from the disk.

To unlock a project:

- From the *Run* tab, click (**Unlock Study**).

  The *Optimate Message* dialog appears.



- Click **Unlock**.

The results for this session are removed and you can modify the study settings. If you run the solver in this session, you are prompted to back up the old results before running the modified study.

Sharing Designs

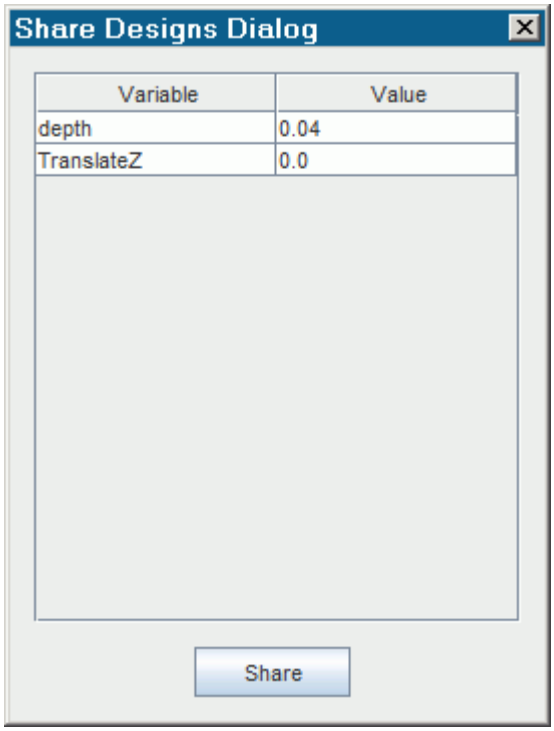
You can share your own design with the study within Optimate.

This option allows you to insert your own design into the study. You can insert a design at any time into the study during the run. You can then study the current set of results and make a best guess at a good design. Optimate then evaluates this design and uses the results for the rest of the run.

To share a design:

- From the *Run* tab, click (**Share Design**).

  The *Share Designs Dialog* appears.



- Set the desired values for each variable and click **Share**.

This design runs after the set of current evaluations complete.

Running the Baseline Design

Optimate can run a single evaluation known as the baseline. The baseline design represents the first design that is executed when you run the study. This design is defined by the baseline values of all the variables in the *Parameters* tab.

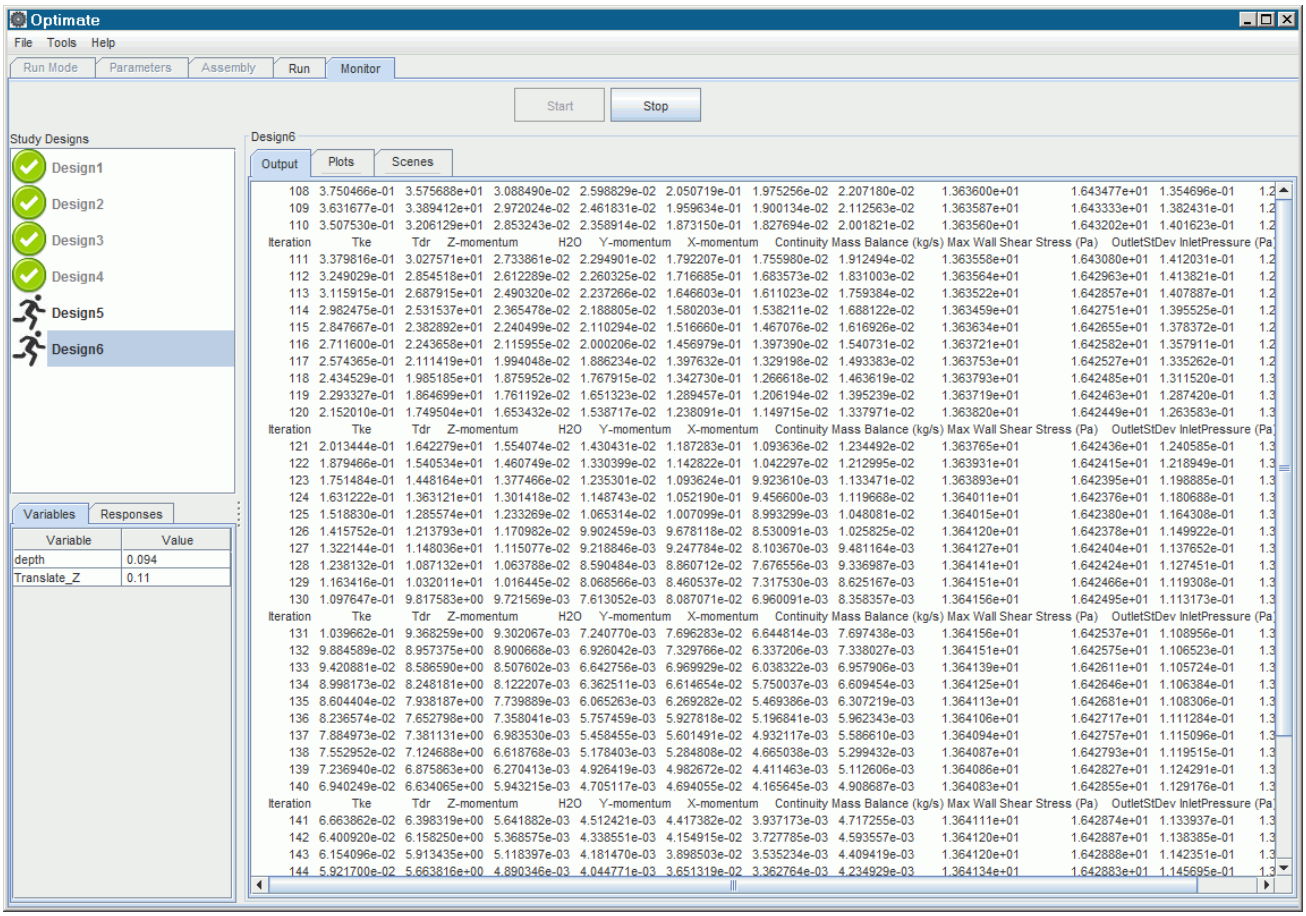
To run the baseline design:

1. From the *Run* tab, click **Run** > **Run Baseline**.

Monitoring Designs

While the study is running, you can monitor the progress of each design.

With the monitor function, you can track all designs in a study. For example, you can see which designs have finished and, of these finished designs, which designs are feasible and which designs are not feasible. You can also see which designs are currently running and connect to them. Once connected, you can view the content of the *Output* window, the plots, and the scenes of that particular simulation.



To monitor the designs in a study:

1. Run the study.
2. Click the *Monitor* tab.
3. Click **Start**.

   The *Study Designs* group box populates with the designs. A status icon appears next to each design to indicate at what stage the design is at. See Design Status Reference.
4. Select a design in the *Study Designs* group box.

While the study is running, you can view the values of the variables that are being analyzed in the selected design in the *Variables* tab. Once the design completes, you can view the response data in the *Responses* tab.

When the design completes, you can view the log file by right-clicking and selecting **Edit Log**.

Connecting to Designs

While a study is running, you can connect to active designs and view the output, plots, and scenes from that design:

1. Right-click a design that is running and select **Connect**.

   When connected, the *Output*, *Plots*, and *Scenes* tab appear in the workspace.
2. Click the desired tab to view the data from the simulation.

   For scenes and plots, click the relevant tab and select the desired plot or scene that you want to display.

   The live data from the simulation is displayed in the relevant tab.

   Plots, and scenes in particular, contain heavy data that you may not want to transfer over the network for an extended period. You can choose to stop displaying live data from scenes and plots by clicking on the relevant tab and selecting **None**.

Stopping the Monitor

If you no longer want to monitor the designs, you can choose to stop the monitor without affecting the run. Monitoring designs while the study is running can slow down the process due to the extra data that has to be sent to the client. If you are not actively using this feature, you are advised to stop monitoring the designs.

To stop monitoring the study:

- Click **Stop**.

  The monitoring of the designs stops. You can click **Start** to start monitoring the designs once more.

Contents:

Design Status Reference
Manually Marking Designs as Errors or Converged

Design Status Reference

While monitoring jobs, the status of each design in the study is displayed.

Status	Description
	Indicates that the design is running. You can connect to running designs to view the live simulation data live. See Connecting to Designs.
	Indicates that the design has completed successfully. For completed designs, you can view the data for responses that you select in Optimate. Click the *Responses* tab in the **Monitor** panel.

Status	Description
	Indicates that the design is recognized as an error.
	Indicates that the design is being processed. Processing occurs when the design is submitted and the STAR-CCM+ server is loading, or while the design is in the queue and is waiting to start. Processing also occurs when a design has completed but Optimate has not yet extracted the results.

Manually Marking Designs as Errors or Converged

While connected to an active design, you can choose to mark the design as an error or as converged, based on your observations in the output, plots, and scenes.

While monitoring a design, if you notice that the simulation has clearly converged, or that something is clearly wrong, you can mark this design as either converged or as an error. The selected design is stopped and tagged appropriately. The study then moves on to the next design.

To mark a design as either converged or as an error:

1. Connect to a design. See Connecting to Designs.
2. Mark the design:
   - To mark a design as an error, click the scene tab and select **Mark as error**.
   - To mark a design as converged, click the scene tab and select **Mark as converged**.

   The status of the design updates to reflect the decision. See Design Status Reference

Exporting Scenes from Optimate to STAR-View+

You can export scene data from Optimate as STAR-View+ format.

To export scenes as STAR-View+ format:

1. Set up the variables, responses, and plots and scenes for the Optimate study. See The Optimate Workflow.
2. In the *Parameters* tab, right-click the scene you want to export as a STAR-View+ format and select **Properties**.
3. In the *Model View Properties* dialog:
   a) Set the image resolution.
   b) Set the zoom level for the image.
   c) Activate the **Export to STAR-View+** option.
   d) Click **OK**.
4. Click **Tools** > **Options**.
5. In the *Options* dialog:
   a) Set **STAR-View+ Export Mode** to the desired mode. See STAR-View+ Export Mode.
   b) Click **Close**.
6. Finish setting up the Optimate study and run the study.

Running a CAD Robustness Study

The CAD robustness study checks whether the CAD geometries that are created from the parameters you specify in the Optimate study are valid.

To run a CAD robustness test:

- In Optimate, select **Tools** > **Options**.
- In the *Options* dialog, set the parameters for the CAD Robustness Study. See CAD Robustness Study Options.
- From the *Run* tab, click **Run** > **CAD Robustness**.

  Optimate samples the design space using a Latin Hypercube sampling technique.

After you run a CAD Robustness analysis, you can post-process the analysis:

- Launch HEEDS Post.
- In HEEDS Post, locate the objective called **CAD Robustness Report**.

  Any design with a value of 0 for this report means that the design was unable to regenerate the CAD correctly.

Note: If a geometry manages to generate successfully, this does not guarantee that it can be meshed successfully.

Optimate Licensing

Optimate and Optimate+ are add-on products for STAR-CCM+.

By default, all Optimate and Optimate+ sessions use Power Tokens (`DOEtoken` feature licenses). Power Tokens are a unique and flexible approach to STAR-CCM+ licensing. See **User Guide** > **Using STAR-CCM+** > **STAR-CCM+ Licensing** > **Power Tokens**.

The following licenses are required to run an Optimate study within STAR-CCM+:

- 1 STAR-CCM+ session license (`ccmpsuite` feature license). Used for the duration of your Optimate study.
- 1 Optimate (`doesuite`) or Optimate+ (`doesuiteplus`) session license. Used for the Optimate solver.
- 1 license for each concurrent configuration. This may be a Power Token (`doetoken`) or STAR-CCM+ session license (`ccmpsuite`)
- You can run each configuration in parallel. Each process requires a Power Token (`doetoken`) or STAR-HPC (`hpcdomains`) license.

You can control the order the licenses are checked out. You can also use Power Tokens for Pre- and Post-Processing sessions in STAR-CCM+.

Coupling Optimate to External Optimization Codes

Coupling Optimate to an external optimization code allows the external code to drive the design exploration study.

Optimate creates the input and output files that you use with the external optimization code, as well as the macro that is used to run STAR-CCM+ with the values defined in these input and output files.

To couple Optimate to an external optimization code:

1. Load your baseline simulation file in STAR-CCM+.
2. Launch Optimate and either create or load a study.
3. From the **Tools** menu, select **Options**.
4. In the *Options* dialog, activate the **Use External Optimizer** option and click **Save and Close**.

   The *Run Mode* and *Run* tabs are grayed out in the Optimate environment. Only the tabs that are relevant to this mode of Optimate are displayed.
5. Click the *Parameters* tab and set up the variables, responses, plots, and scenes that you want to transfer between STAR-CCM+ and the external optimization code.

   Do not set any other values, such as *Min* and *Max*, or any constraints or objectives for your responses. These items are not required when coupling to external optimization codes and are managed by the external code. Any values that you set here are ignored.
6. Click the *Assembly* tab and set the options in the *STAR-CCM+ Run Options* group. See Assembly Panel Reference.

   Only the options that are relevant to STAR-CCM+ are valid.
7. Click **Build Study**.

   Optimate creates the input (`simulation_name.in`), output (`simulation_name.out`), and macro (`StarDriver.java`) files in the study folder. The input and output files are text files that you can incorporate into your external optimization code. The macro file tells STAR-CCM+ what data is the input and what data is the output.
8. Launch the external optimization code and incorporate the input and output files.

Optimate Options

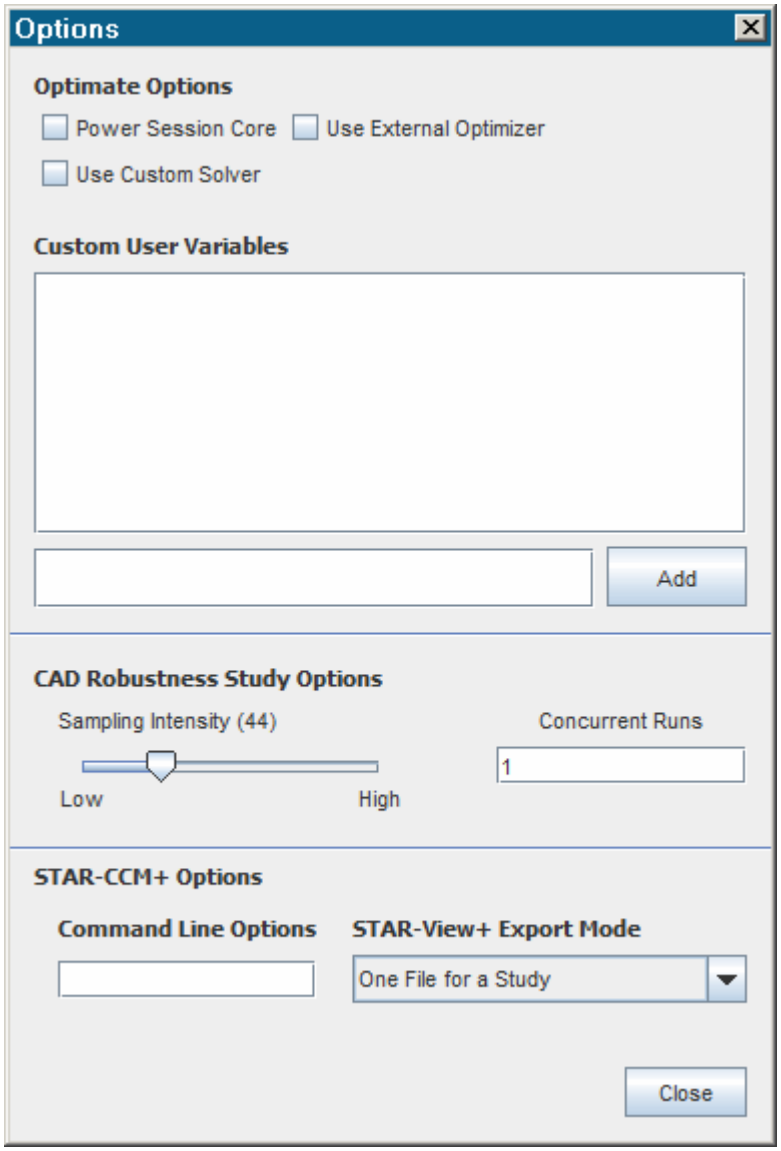
You can configure the behavior of Optimate through the Options dialog.

From the Options dialog, you can instruct Optimate to use an external solver, specify the study options fo the CAD robustness study, and add STAR-CCM+ command line options.

To access the the dialog:

1. In Optimate, click **Tools** > **Options**.

   The *Options* dialog appears.



2. Modify the options. See Optimate Options Panel.
3. Click **Save and Close**.

Contents:

Optimate Options Panel

Optimate Options Panel

You can control the behavior of Optimate through the Options panel.

Optimate Options

The Optimate options affect the behavior of Optimate.

Power Session Core	Determines which STAR-CCM+ license is used as the study license if you have activated the **Use Power Token** option in the *Assembly* tab. See Licensing Options. By default, Optimate tries to check out a `ccmpsuite` feature license for the duration of the study. See Optimate Licensing. If you don't have any `ccmpsuite` feature licenses, or know that this won't be available, activate this option to use a power session license in its place.
Use External Optimizer	Allows an external code to drive the design exploration study. Optimate writes out the macro to automate the process, in addition to two text files which define the inputs and outputs of the study. You are responsible for linking these files with the external tool. See Coupling Optimate to External Optimization Codes.
Use Custom Solver	Allows you to choose a different version of the Optimate solver. In the dialog that appears, navigate to the directory that contains the desired version of the Optimate solver executable and select it. When you next run an Optimate study, this version of the solver will be used.

Custom User Variables

Allows you to link in variable classes to Optimate. For example, if you want to create a variable out of a turbulence model coefficient, something which you cannot do natively through Optimate, you can create a custom user variable using the API in Optimate.

To add a custom user variable, in the text field below the *Custom User Variables* group box, type the fully qualified class path to the class that you wrote and click **Add**. If the entry is accepted, this means that Optimate recognizes entry as a variable. When you add a variable in the *Parameters* tab, custom variables are shown (including filter type) in the *Import Variables* dialog. See Optimate Workflow.

CAD Robustness Study Options

The CAD Robustness Study Options affect the behavior of the CAD robustness study.

Sampling Intensity	Controls the number of samples that are run for a CAD robustness study. You control the number of samples with the slider. The higher the sampling intensity, the greater the number of samples that are picked throughout the design space.
Concurrent Runs	Controls the number of CAD robustness evaluations that are run in parallel.

STAR-CCM+ Options

Command Line Options	Specify additional command line arguments for launching each STAR-CCM+ evaluation. By default, Optimate uses power tokens before using HPC's for each additional CPU withing an evaluation.	
STAR-View+ Export Mode	Specifies the data that is included in the exported STAR-View+ scene. To set up the scene export, see Exporting Scenes from Optimate to STAR-View+.	
	One File for a Study	Generates a single STAR-View+ scene for the study. The scene file is stored in the `<STUDY_FOLDER>/star_0/` folder.
	One File for a Design	Generates a STAR-View+ scene for each design. The scene files are located in the `<STUDY_FOLDER>/POST_0/Design<x>/star/` folders.
	Study File and Design File	Generates a single STAR-View+ scene for the study as well as a STAR-View+ scene for the individual designs.

File Descriptions

File descriptions are provided below to help you debug cases that are not executing as expected.

Build Projects

When you build an Optimate project, the following files are found in the simulation directory.

File	Description
`Agent_Group_1.in`	Contains information about the run mode and the number of runs to be completed.
`Assembly.in`	Contains information about the run mode.
`Definitions.in`	Contains information about variables that are changed at run time. Here, you can find the name of the variable, the minimum value, the maximum value, and the baseline value.
`M_in_star_{SIM FILE NAME}_optimate.in`	Specifies how Optimate modifies the input file for each run such that STAR-CCM+ runs use the updated variable values.
`Performance0.in`	Contains information about the outputs, objectives, and constraints for each project, and tells Optimate where it can find these values after each STAR-CCM+ job completes.
`{SIM FILE NAME}_optimate.in`	Contains the variable names and values that are read by STAR-CCM+ at run time. This file is copied to the design directory at run time. Optimate modifies this file with new values. The STAR-CCM+ batch script reads this file and then updates the fields in STAR-CCM+ before running the analysis.
`{SIM FILE NAME}_optimate.out`	Written by each STAR-CCM+ job at completion and contains the values of the outputs that are read by Optimate. If the STAR-CCM+ analysis does not complete successfully, this file is not written and Optimate marks the analysis as a failure.
`Process1.in`	Contains the command to start each STAR-CCM+ analysis, either via a queuing system or directly. Also contains information about how often to check the project for completion and the named files to be copied into each design directory.
`Representation0.in`	Contains information about the resolution of each variable.
`StarDriver.java`	Batch script that each STAR-CCM+ job executes. It is responsible for reading the input file, adjusting parameters in STAR-CCM+, remeshing if necessary, running the analysis, and exporting the results.
`UserDesigns0_DesignSweep.in_backup_at_0_evals`	Contains the value of each variable to be used for each STAR-CCM+ job. This file is only written for analyses that are run using Design Sweeps mode. See Design Exploration.

Running and Completed Projects

The following files are found in the project directory.

File	Description
`Agent_Group_1.mes`	Output from the Optimate process. Contains information about when each STAR-CCM+ job is started and completed. If there are any errors parsing the other input files, you can find a detailed message here.

`star0.gph`
`star0.hst`
`star0.res`
`star0.rpt`
	Post-processing files that are loaded into HEEDS Post. If you run a project on a computer that does not have HEEDS Post, you must copy these four files and the `star0.hds` file to the machine with HEEDS Post.

`star_0/`	Directory that is created at run time and contains all the design directories.

Troubleshooting

You can find information about failed jobs in the files written by Optimate and STAR-CCM+.

Message File

During an Optimate run, Optimate writes to the `Agent_Group_1.mes` file. This file is located in the same directory as the original simulation file. Errors parsing the project files written by Optimate are stored in this location.

A list of commonly encountered errors with suggested solutions is provided below:

Type of Error	Suggested Solution
Licensing errors	• Check the `CDLMD_LICENSE_FILE` variable • Make sure that the license server is running
Errors executing	• Check the command in the `Process1.in` file

STAR-CCM+ Log File

During an Optimate run, Optimate creates directories for each STAR-CCM+ job. The directory for each run are located in the following directory:

`star_0/DesignN/star/`

where `N` is the design ID of each job, starting at one and ending with the number of jobs to be run.

If you activate the **Save all Simulation Files** option in the *Assembly* Panel, these folders are not deleted and persist after the run completes. If you deactivate this option, the directories only persist for the duration of the run.

If any job fails, it is placed in the directory:

`star_0/DesignN-ERROR/star/`

The above directories contain all the input files, outputs files, a log file, and the simulation file for each run. If STAR-CCM+ fails, an error message is added to the `star.log` file.

A list of commonly encountered errors with suggested solutions is provided below:

Type of Error	Suggested Solution
Errors getting appropriate licenses	• Change the number of simultaneous jobs in the `Process1.in` file • Change the number of cores per STAR-CCM+ job in the `Process1.in` file • Check the `CDLMD_LICENSE_FILE` variable • Make sure that the license server is running
Errors modifying the 3D-CAD model	• Change values to avoid invalid geometry
Errors meshing	• Check the mesh settings in the original simulation file

Type of Error	Suggested Solution
Floating point errors	• Review the solver settings • Review the initial conditions from the original simulation file
Errors importing CAD files	• Check the `directoryCAD` variable in the `StarDriver.java` file
Errors exporting scenes	• Check the `directoryCAD` variable in the `StarDriver.java` file
Errors saving simulation file	• Check the `directoryPOST` variable in `StarDriver.java` file
Incompatible version errors	• Modify the `path` environmental variable such that the correct version of STAR-CCM+ is listed first in the path

HEEDS Post

If HEEDS Post fails to open or does not work as expected, you can take a look at the HEEDS Post log file for further information. The log file is called `HEEDSPost.log` and is created in the home directory.

Optimate Is Running But No Designs Are Launched

This issue is typically caused by an incorrect or inappropriate STAR-CCM+ command line option in the **Tools** > **Options** command line textbox.

To troubleshoot issues like these:

1. Navigate to the `star_0/Design<N>/star/` folder.
2. Open `Tool_star_output.msg` in a text editor of your choice.

   This file contains the output from the command prompt or shell that was used to launch the STAR-CCM+ process. Typically, when a design fails to launch, it includes a message from STAR-CCM+ that indicates the command line arguments cannot be interpreted. This file also provides a reference to the acceptable command line arguments.
3. In Optimate, go to **Tools** > **Options** and check the contents of the command line.
4. Make sure that any additional command line arguments conform to the guide that is provided in the `Tool_star_output.msg` file.

Optimate Tutorials

This section contains tutorials for Optimate and Optimate+.

Contents:

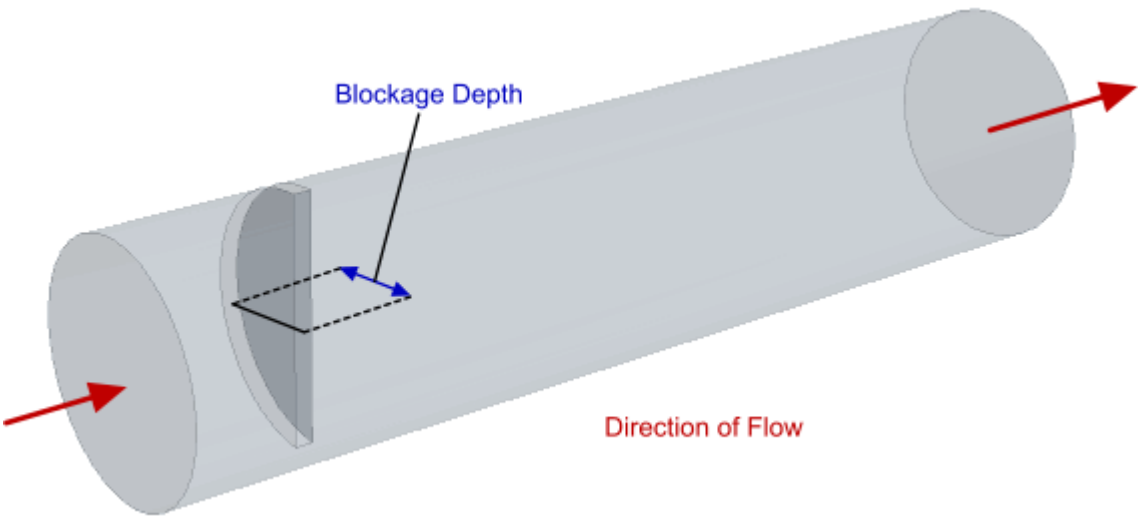
Design Sweeps Tutorial
Design of Experiments Tutorial
Pareto Optimization Tutorial
Use Existing Designs Tutorial
Optimization with Macro Tutorial

Design Sweeps Tutorial

This tutorial demonstrates how to use Optimate to setup a Design Sweeps study.

In this type of study, you describe all of the inputs for each case for STAR-CCM+ to run. This type of study is applicable when recreating test results or creating loads databases.

In this tutorial, you vary the depth of a blockage in a pipe to see how it affects the flow. The depth of the blockage is a design parameter that Optimate can modify during the study.



Contents:

Prerequisites
Starting Optimate
Selecting the Study Type
Creating the Variables
Selecting Data to Post-Process
Defining the Run Settings
Running the Study
Post-Processing

Prerequisites

This tutorial assumes that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Techniques	Associated Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+

Techniques	Associated Tutorial
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

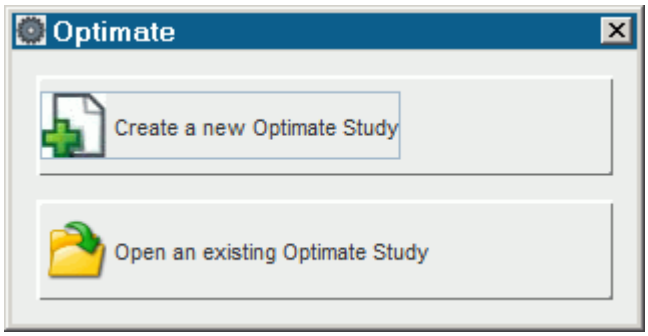
This tutorial requires an Optimate+ license in order to use the **Optimization using SHERPA** run mode.

Starting Optimate

Load the starting simulation file and launch Optimate.

To launch Optimate for the study:

1. In your workspace, create a folder named `designSweeps`.
2. Navigate to `[INSTALL_DIR]/doc/startutorialsdata/optimate/data`
3. Copy `pipeBlockage.sim` to your workspace.
4. Launch STAR-CCM+.
5. Select **File** > **Load Simulation...**
6. In the *Load Simulation* dialog, click **Browse...**.
7. In the *Open* dialog, select `pipeBlockage.sim` from your workspace and click **Open**.
8. In the *Load Simulation* dialog, click **OK**.
9. From the STAR-CCM+ toolbar, click (**Start Optimate**).
   The *Optimate* dialog appears.



10. Click **Create a new Optimate Study**.
    The Optimate workspace appears.

Selecting the Study Type

To select the type of study to perform:

1. In the Run Mode tab, make sure that *Run Mode* is set to **Design Sweeps**.
2. Save the Optimate study file by selecting **File** > **Save As** from the menu.
3. In the *Save* dialog, save the file as `designSweeps.optm`.

Creating the Variables

Create the variables for your study. In this tutorial, you want to vary the depth of the blockage in the pipe from 0.08 meters to 0.11 meters.

1. In the *Parameters* tab, click **Create Variables**.
2. In the *Import Variables* dialog, expand the **3D-CAD Models** > **3D-CAD Model 1** > **Design Parameters** node.
3. Select **depth** and click **OK**.

   The design parameter, **depth**, is added to the *Variables* group box and is identified as a variable that Optimate can modify.
4. Set the minimum and maximum size for the chosen variable, as well as the number of increments to use:
5. Set *Min* to `0.08`.

   The units that are used are taken from the simulation file. In this case, the units are meters.
6. Set *Max* to `0.11`.
7. Set *Resolution* to `7`.

   This property instructs Optimate to execute seven STAR-CCM+ runs for this variable. The values are equally spaced starting at 0.08 meters and ending at 0.11 meters.

Selecting Data to Post-Process

Select what data you want Optimate to save during the analysis. This data is used for post-processing in a later step.

1. In the *Parameters* tab, click **Create Responses**.
2. In the *Import Reports* dialog, select all reports and click **OK**.

   All reports from the simulation file are added to the *Responses* box. Optimate will save data from these reports during the analysis.
3. In the *Responses* group box, right-click the **InletPressure** response and select **Set as objective**.

   The **InletPressure** objective appears at the bottom of the *Parameters* tab.
4. For the **InletPressure** objective, make sure that *Goal* is set to **Minimize**.

   Even though there will be no optimization for this study, the results are sorted based on the performance of each case, ranging from the case with the smallest inlet pressure to the case with the largest inlet pressure.

Set Optimate to export data from the relevant scenes and plots after each STAR-CCM+ run:

5. Click **Create Model Views**.
6. In the *Import Plots and Scenes* dialog, select **Plots** > **Residuals** and **Scenes** > **Velocity**, then click **OK**.

   The **Residuals** plot and the **Velocity** scene are added to the *Plots and Scenes* group box.

Defining the Run Settings

Specify the run settings for the solver. To reduce the time it takes to complete the study, you can run multiple jobs at the same time.

1. Click the *Assembly* tab.

2. Set *Simultaneous Jobs* to `2`.

   This property instructs Optimate to run two jobs at the same time. Note that your machine must have at least two cores spare. If you have more cores, and sufficient tokens, you can increase this value or increase the number of cores that are used per job through the *Cores per Job* property. The total number of tokens required to run your study is displayed under the *Tokens Required* property.
3. Activate the **Clear Solution** option.

   This option instructs STAR-CCM+ to clear the solution before each new case is run.
4. Make sure that *Optimate Save Option* is set to **All Designs**.

   This option instructs Optimate to save all design iterations.
5. Save the study.

Running the Study

The study is now ready to be run. To start the job on your local machine:

1. Click the *Run* tab.
2. Click (**Run**).

   All necessary files for this study are created.

Post-Processing

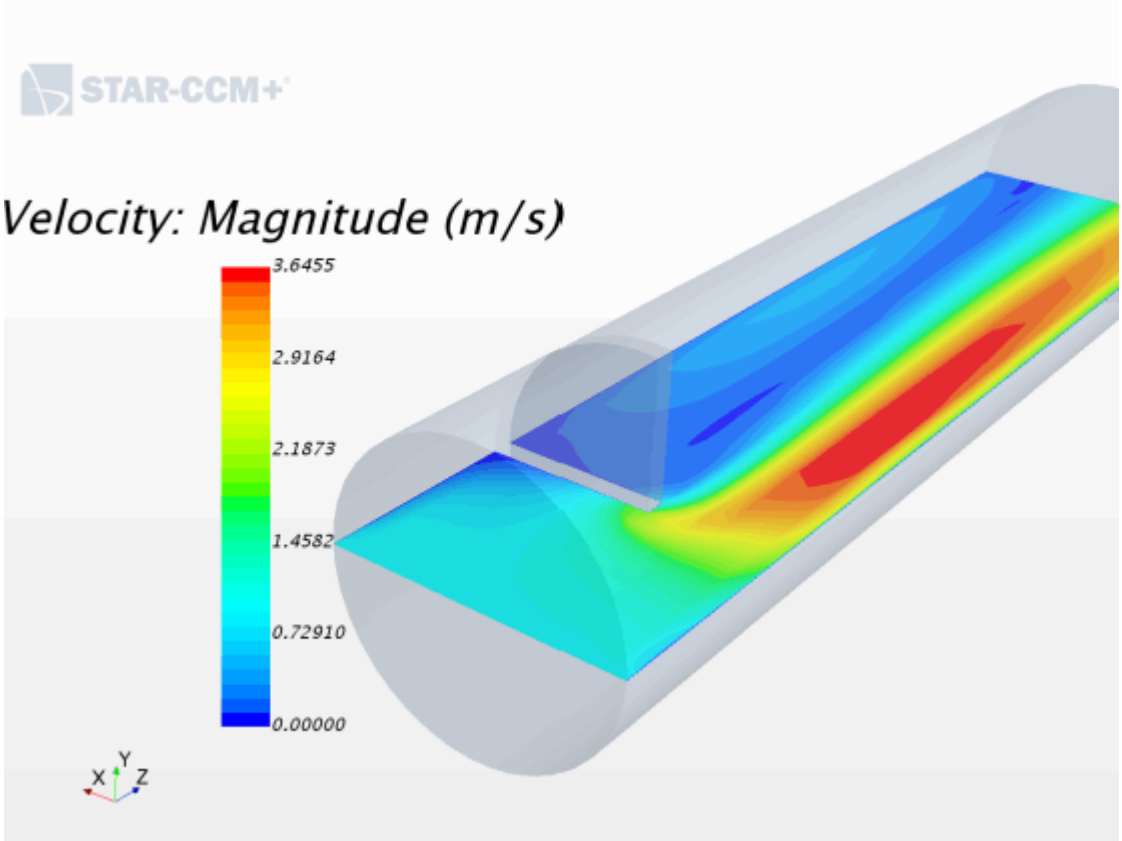
Post-process the results in HEEDS Post.

Create a Model View plot to visualize the different designs:

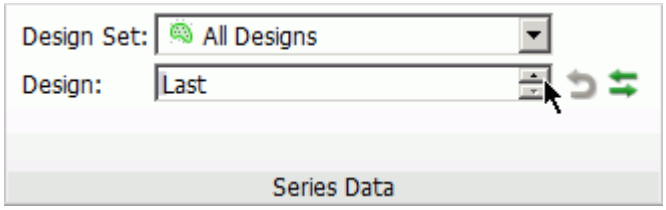
1. Click (**Open HEEDS|Post**) to launch HEEDS Post.
2. From the HEEDS Post ribbon, in the **General Plots** panel, click (**Model View**).

   The *Create New Plot Wizard* dialog appears.
3. From the *Available files* group box, select **Velocity-scene.png**.
4. Click **Finish**.

   A velocity scene is added to the HEEDS Post environment.



5. Select **Model_6** from the **Plots** folder,
6. In the HEEDS Post ribbon, in the *Series Data* panel, cycle through the designs to see how the changes to the design affect the flow in the pipe.

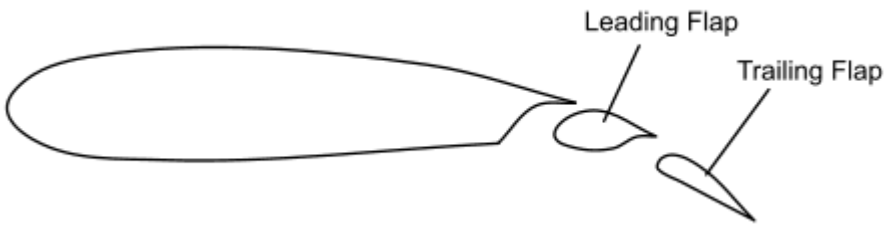


Design of Experiments Tutorial

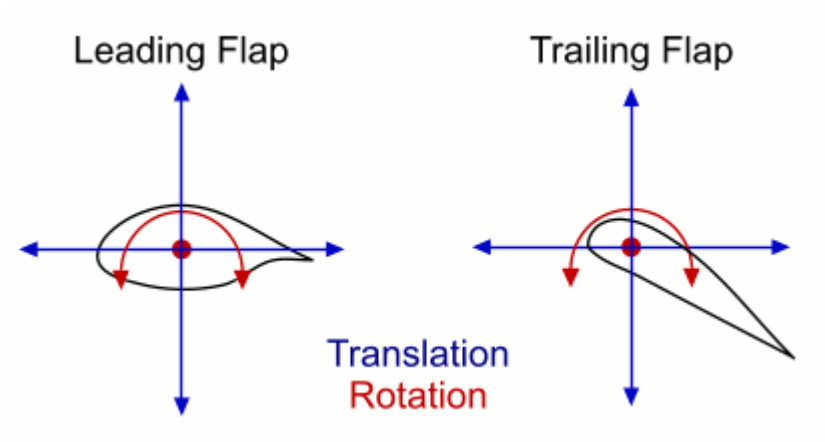
This tutorial demonstrates how to use Optimate to set up a Design of Experiments study.

In this tutorial, you first create a response surface that is representative of the physical problem and then proceed to carrying out a full factorial Design of Experiments study on the generated response surface. The aim of this study is to gain sensitivity information about the problem.

The geometry in this tutorial consists of an aircraft wing with a double-slotted flap system.



Optimate can modify the X and Y translation of each flap, as well as the rotation of the flap.



Contents:

Prerequisites
Part 1: Creating a Response Surface
Part 2: Running a Full Factorial Design of Experiments Study

Prerequisites

This tutorial assumes that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Techniques	Associated Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Part 1: Creating a Response Surface

In the first part of this tutorial you create the response surface.

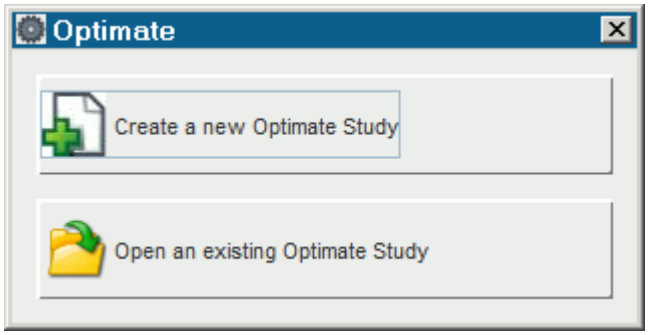
Contents:

Starting Optimate
Selecting the Study Type
Creating the Variables
Selecting Data to Post-Process
Defining the Run Settings
Running the Study
Post-Processing

Starting Optimate

Load the starting simulation file and launch Optimate. To launch Optimate for the study:

1. In your workspace, create a folder named `DoE`.
2. Navigate to `[INSTALL_DIR]/doc/startutorialsdata/optimate/data`.
3. Copy `airfoil.sim` to your workspace.
4. Launch STAR-CCM+.
5. Select **File** > **Load Simulation...**
6. In the *Load Simulation* dialog, click **Browse...**
7. In the *Open* dialog, select `airfoil.sim` from your workspace and click **Open**.
8. In the *Load Simulation* dialog, click **OK**.
9. From the STAR-CCM+ toolbar, click (**Start Optimate**).
   The *Optimate* dialog appears.



10. Click **Create a new Optimate Study**.

    The Optimate workspace appears.

Selecting the Study Type

To select the type of study to perform:

1. In the *Run Mode* tab, set *Run Mode* to **Design of Experiments**.
2. In the *Run Options* panel, make sure that *Response Surface Modeling* is set to **Latin Hypercube**.
3. Save the Optimate study by selecting **File** > **Save As** from the menu.
4. In the *Save* dialog, save the file as `responseSurface.optm`.

Creating the Variables

Specify the design parameters for your study. In this tutorial, you want to vary the position of the leading and trailing flaps of the wing.

1. Click the *Parameters* tab.
2. Click **Create Variables**.
3. In the *Import Variables* dialog, expand the **3D-CAD Models** > **3D-CAD Model 1** > **Design Parameters** node.
4. Select **LeadingFlapRotateAngle**, **TrailingFlapRotateAngle**, and **TranslateTrailingFlap** and click **OK**.

   The selected variables are added to the *Variables* group box. The X, Y, and Z components of the **TranslateTrailingFlap** parameter are imported as separate variables.

As the geometry is two-dimensional, remove the Z component variable:

5. Resize the **Name** column to display the full name of the variables.
6. Right-click the **TranslateTrailingFlap_Z** variable and select **Delete**.

Set the minimum and maximum size for the variables:

7. Select the **LeadingFlapRotateAngle** variable and set *Min* to `-4.0` and *Max* to `5.0`.
8. For the remaining variables, set the following values:

Variable	Minimum	Maximum
TrailingFlapRotateAngle	`-15.0`	`25.0`
TranslateTrailingFlap_X	`-0.02`	`0.02`
TranslateTrailingFlap_Y	`-0.01`	`0.01`

Selecting Data to Post-Process

For this study, you examine the sensitivity to all of the available outputs.

1. In the *Parameters* tab, click **Create Responses**.
2. In the *Import Reports* dialog, select **Cl** and **Cd** and click **OK**.

   The selected reports are added to the *Responses* group box.

3. Click **Create Model Views**.
4. In the *Import Plots and Scenes* dialog:
   a) Expand the **Plots** node and select **Lift and Drag** and **Residuals**.
   b) Expand the **Scenes** node and select **Scalar-Mach**.
   c) Click **OK**.

   The selected plots and scenes are added to the *Plots & Scenes* group box.

Defining the Run Settings

Specify the run settings for the solver. To reduce the time it takes to complete the study, you can run multiple jobs at the same time.

1. Click the *Assembly* tab.
2. Set *Simultaneous Jobs* to `2`.

   This property instructs Optimate to run two jobs at the same time. Note that your machine must have at least two cores spare. If you have more cores, and sufficient tokens, you can increase this value or increase the number of cores that are used per job through the *Cores per Job* property. The total number of tokens required to run your study is displayed under the *Tokens Required* property.
3. Activate the **Clear Solution** option.

   This option instructs STAR-CCM+ to clear the solution before it runs each new case.
4. Set *Number of Runs* to `20`.

   This option instructs STAR-CCM+ to run 20 different cases. The minimum number of evaluations for 4 factors is (4+1)*(4+2)/2, which is 15. To ensure a good fit, however, you can run 20 evaluations.
5. Save the study.

Running the Study

The study is now ready to be run. To start the job on your local machine:

1. Click the *Run* tab.
2. Click (**Run**).

   All necessary files for this study are created.

Post-Processing

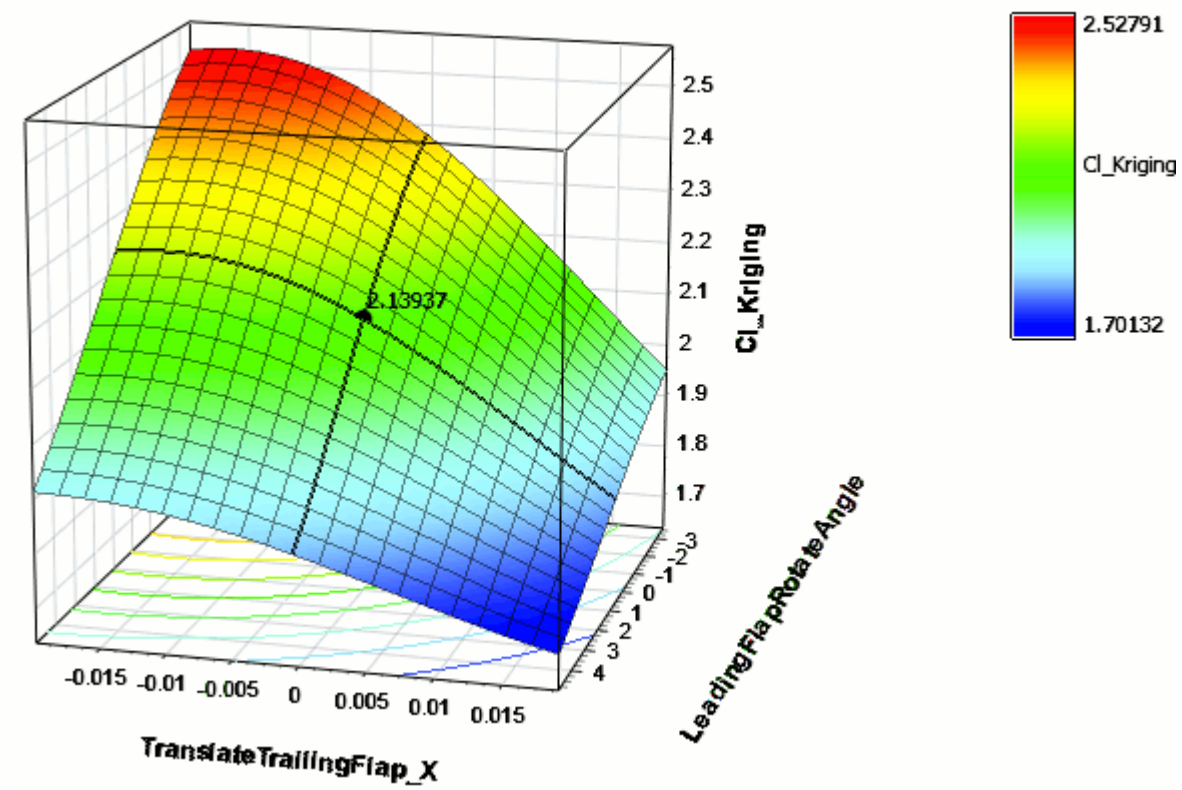
Visualize the 3D response surface in HEEDS Post. The Latin Hypercube technique is based on a random sampling method to explore the design space. Consequently, the results that you obtain may differ to those displayed in this tutorial.

1. Click (**Open HEEDS|Post**) to launch HEEDS Post.
2. From the HEEDS Post ribbon, in the **DOE Plots** panel, click (**3D Response Surface**).

   The *Create New Plot Wizard* dialog appears.
3. Set *X-Axis Parameter* to **LeadingFlapRotateAngle**.

4. Set *Y-Axis Parameter* to **TranslateTrailingFlap_X**.
5. Select the coefficient of lift with the linear fit model, **Cl_Kriging**, as the surrogate.
6. Click **Finish**.

   The 3D response surface is created.



You can now use the created response surface for further DoE or Stochastic analyses. Close HEEDS Post, saving the results, but leave Optimate running.

Part 2: Running a Full Factorial Design of Experiments Study

Use the response surface that you generated in part 1 to run a full factorial Design of Experiments study.

Contents:

Modifying the Study
Running the 2 Level Full Factorial Study
Post Processing the Full Factorial Study

Modifying the Study

Modify the study to run a 2 level full factorial study. After running a study, Optimate locks the study parameters. You can unlock the study to modify the parameters.

1. In Optimate, in the *Run* tab, click (**Unlock Study**).

   In the dialog that appears, click (**Unlock**).

   The study unlocks and you can now modify the study parameters.
2. Click the *Run Mode* tab and set *Sensitivity and Screening* to **2 Level Full Factorial**.
3. Click the *Parameters* tab and make sure that the variables are as defined in Part 1.

   The variables and variable ranges must match those in the response surface.
4. In the *Responses* group box, select **Cl** and set *Source* to **RSM**.
5. Right-click the **Cl** response and select **Properties**.
6. In the *Response Properties* dialog, click **Import RSM Data**.
7. In the *Open* dialog, navigate to the `airfoil_responseSurface` folder, select `Cl-Kriging-0.mdl` and click **Import**.
8. In the *Response Properties* dialog, click **OK**.
9. Using the same technique, import the RSM data for the **Cd** response.

   The corresponding RSM data is `Cd-Kriging-0.mdl`.
10. Click the *Assembly* tab.

    Simultaneous Jobs and *Number of Runs* are automatically defined to match the requirements of a two-level full factorial run.
11. Save the study as `responseSurface_FullFactorial.optm`.

Running the 2 Level Full Factorial Study

The 2 Level Full Factorial study is now ready to be run. To start the job on your local machine:

1. Click the *Run* tab.
2. Click (**Run**) and select **Run**.

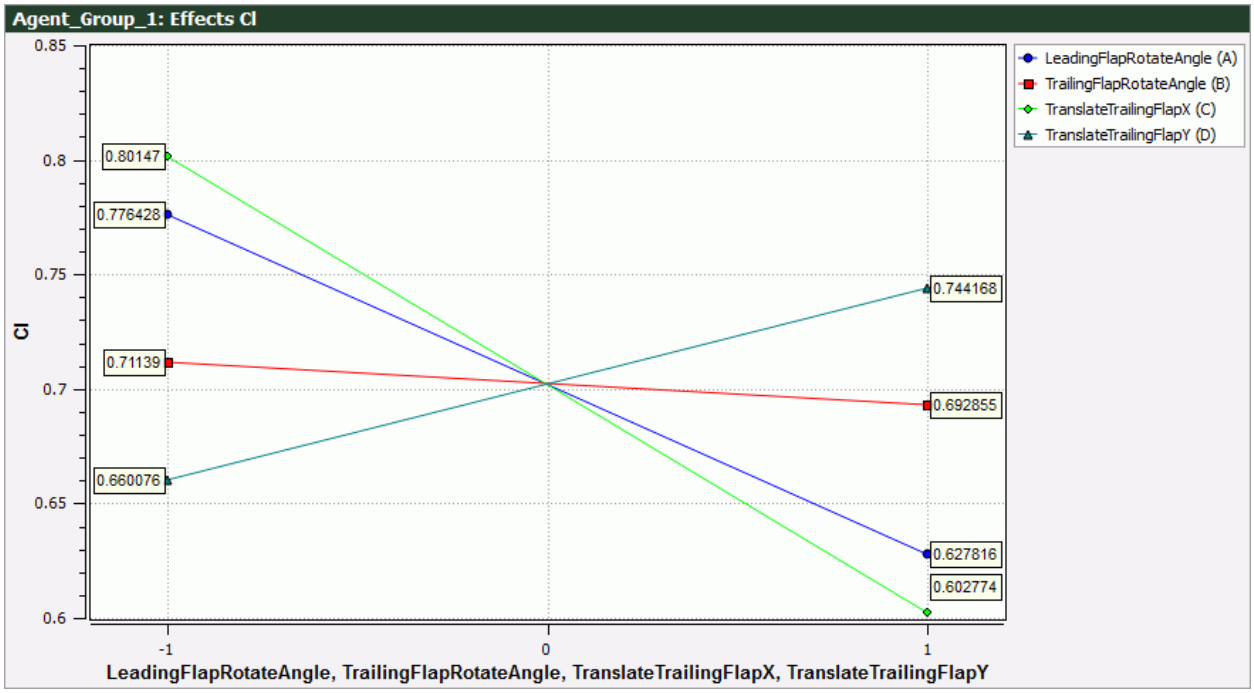
   The output from Optimate is displayed in the *Output* window. The STAR-CCM+ servers for each job are visible in the server list.

Post Processing the Full Factorial Study

Post-process the results of the full factorial study. The Latin Hypercube technique is based on a random sampling method to explore the design space. Consequently, the results that you obtain may differ to those displayed in this tutorial.

1. Click (**Open HEEDS|Post**) to launch HEEDS Post.
2. In HEEDS Post, select the **Plots** > **Effects Cl** node.

   This graph shows the effect of each factor on Cl, as the factors go from their low to high values. The higher the slope of a specific curve, the more effect that factor has on Cl.



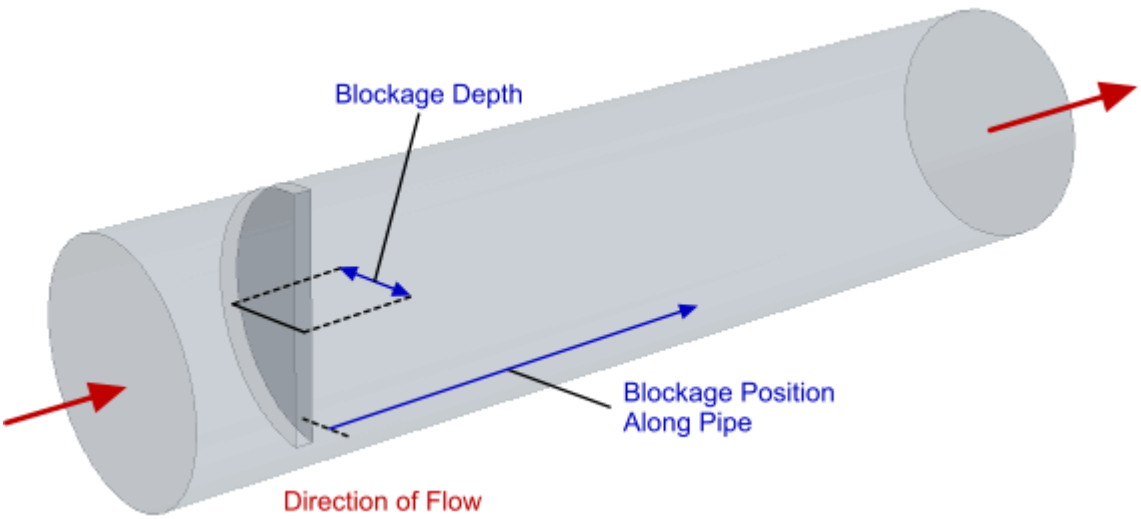
Similar graphs are automatically generated for Cd.

Pareto Optimization Tutorial

This tutorial demonstrates how to use Optimate to set up a Pareto Optimization study.

In this type of study, you define the range for the geometry modifications and competing objectives. The MO-SHERPA algorithm then finds the set of designs that are on the Pareto front.

In this tutorial, Optimate varies the depth of a blockage in a pipe, as well as the position of the blockage along the pipe, with the aim of minimizing the pressure at the inlet.



Contents:

Prerequisites
Starting Optimate
Selecting the Study Type
Creating the Variables
Selecting Data to Post-Process
Defining the Run Settings
Running the Study
Post-Processing

Prerequisites

This tutorial assumes that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Techniques	Associated Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+

Techniques	Associated Tutorial
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

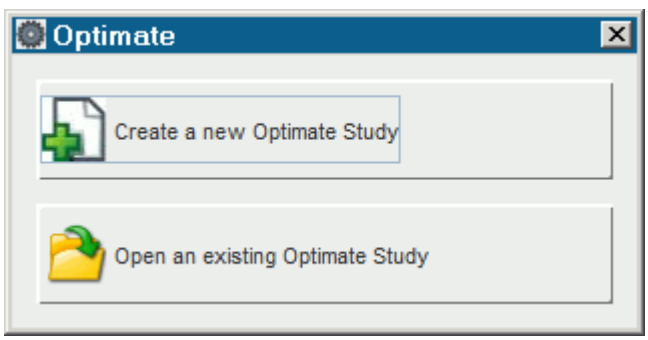
This tutorial requires an Optimate+ license in order to use the **Optimization using SHERPA** run mode.

Starting Optimate

Load the starting simulation file and launch Optimate.

To launch Optimate for the study:

1. In your workspace, create a folder named `paretoOptimization`.
2. Navigate to `[INSTALL_DIR]/doc/startutorialsdata/optimate/data`.
3. Copy `pipeMixingBlockage.sim` to your workspace.
4. Launch STAR-CCM+.
5. Select **File** > **Load Simulation...**
6. In the *Load Simulation* dialog, click **Browse...**
7. In the *Open* dialog, select `pipeMixingBlockage.sim` from your workspace and click **Open**.
8. In the *Load Simulation* dialog, click **OK**.
9. From the STAR-CCM+ toolbar, click (**Start Optimate**).
   The *Optimate* dialog appears.



10. Click **Create a new Optimate Study**.
    The Optimate workspace appears.

Selecting the Study Type

To select the type of study to perform:

1. In the *Run Mode* tab, set *Run Mode* to **Optimization using SHERPA**.
2. Set *Run Options* to **Multiple objective tradeoff study (Pareto front)**.
   Leave the *Archive size* set to `12`.
3. Set *Random seed* to `0.5`.
   The seed value specifies the starting value that Optimate uses for the optimization.

4. Save the Optimate study file by selecting **File** > **Save As** from the menu.
5. In the *Save* dialog, save the file as `paretoOptimization.optm`.

Creating the Variables

Specify the design parameter for your study. In this tutorial, you want to vary the depth of the blockage in the pipe as well as the position of the blockage along the pipe.

1. In the *Parameters* tab, click **Create Variables**.
2. In the *Import Variables* dialog, expand the **3D-CAD Models** > **3D-CAD Model 1** > **Design Parameters** node.
3. Select **depth** and click **OK**.

   The design parameter, **depth**, is added to the *Variables* group box and is identified as a variable that Optimate can modify.
4. Set the minimum and maximum size for the chosen variable, as well as the number of increments to use:
5. Set *Min* to `0.04`.

   The units that are used are taken from the simulation file. In this case, the units are meters.
6. Set *Max* to `0.14`.
7. Set *Increment* to `0.0025`.
8. Using the same technique, import the **Translate** design parameter as a variable.

   The X, Y, and Z components of the **Translate** parameter are imported as separate variables.
9. Delete the **Translate_X** and **Translate_Y** variables.
10. For the **Translate_Z** variable, set the following properties:

Variable	Min	Max	Increment
Translate_Z	`0.0`	`0.2`	`0.0025`

Selecting Data to Post-Process

Select what data you want Optimate to save during the analysis. This data is used for post-processing in a later step.

1. In the *Parameters* tab, click **Create Responses**.
2. In the **Import Reports** dialog, select **InletPressure** and **OutletStDev**, then click **OK**.

   The selected reports are added to the *Responses* group box.
3. In the *Responses* group box, multi-select **InletPressure** and **OutletStDev**.
4. Right-click one of the selected items and select **Set as objective**.

   The selected reports are added as objectives at the bottom of the *Parameters* tab.
5. For both objectives, make sure that *Goal* is set to **Minimize**.

Make sure that the normalization values for each variable are set by the baseline simulation:

6. Right-click the **InletPressure** objective and select **Properties**.
7. In the *Objective Properties* dialog, make sure that *Baseline Normalization* is activated.

8. Repeat the previous two steps for the **OutletStDev** objective.

Set Optimate to export data from the relevant scenes and plots after each STAR-CCM+ run:

9. Click **Create Model Views**.
10. In the *Import Plots and Scenes* dialog, select **Plots** > **Residuals** and **Scenes** > **Velocity**, then click **OK**.

    The **Residuals** plot and the **Velocity** scene are added to the *Plots and Scenes* group box.

Defining the Run Settings

Specify the run settings for the solver. For Pareto optimization studies, you require a minimum of 35 jobs. This values is automatically set in the *Number of Runs* parameter. To reduce the time it takes to complete the study, you can run multiple jobs at the same time.

1. Click the *Assembly* tab.
2. Set *Simultaneous Jobs* to `2`.

   This property instructs Optimate to run two jobs at the same time. Note that your machine must have at least two cores spare. If you have more cores, and sufficient tokens, you can increase this value or increase the number of cores that are used per job through the *Cores per Job* property. The total number of tokens required to run your study is displayed under the *Tokens Required* property.
3. Set *Number of Runs* to `40`.

   The recommended number of runs is 8 to 12 times the *Archive Size*. For this tutorial, the *Archive Size* is `12`, making the recommended number of runs in the region of 96 and 144. To reduce the time it takes to run this tutorial, however, reduce the number of runs to 40.
4. Activate the **Clear Solution** option.

   This option instructs STAR-CCM+ to clear the solution before it runs each new case.
5. Save the study.

Running the Study

The study is now ready to be run. To start the job on your local machine:

1. Click the *Run* tab.
2. Click (**Run**).

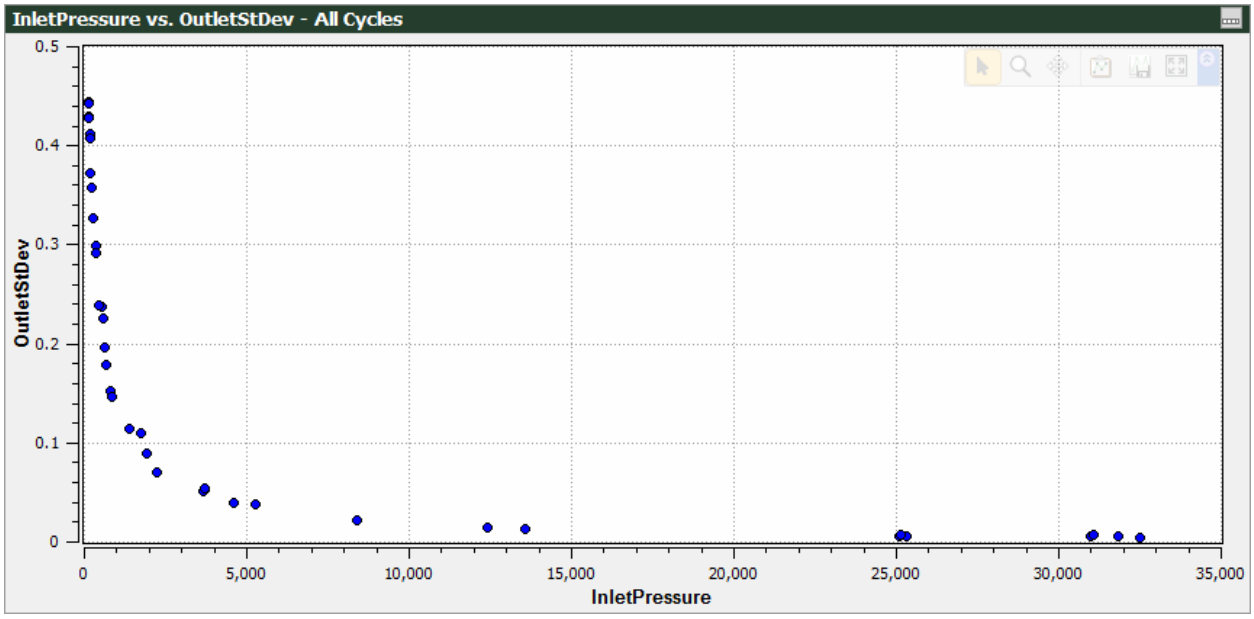
   All necessary files for this study are created.

Post-Processing

For this analysis, the Pareto front is automatically plotted as **InletPressure vs. OutletStDev**. The results that you obtain may differ slightly to those displayed in this tutorial.

1. Click (**Open HEEDS|Post**) to launch HEEDS Post.
2. In HEEDS Post, click the **Plots** > **InletPressure vs. OutletStDev** node to display the Pareto front.

   Set *Cycle* to **All** to display data from all cycles.



Create a 2D relation plot to show the effect of the inputs on the outputs.

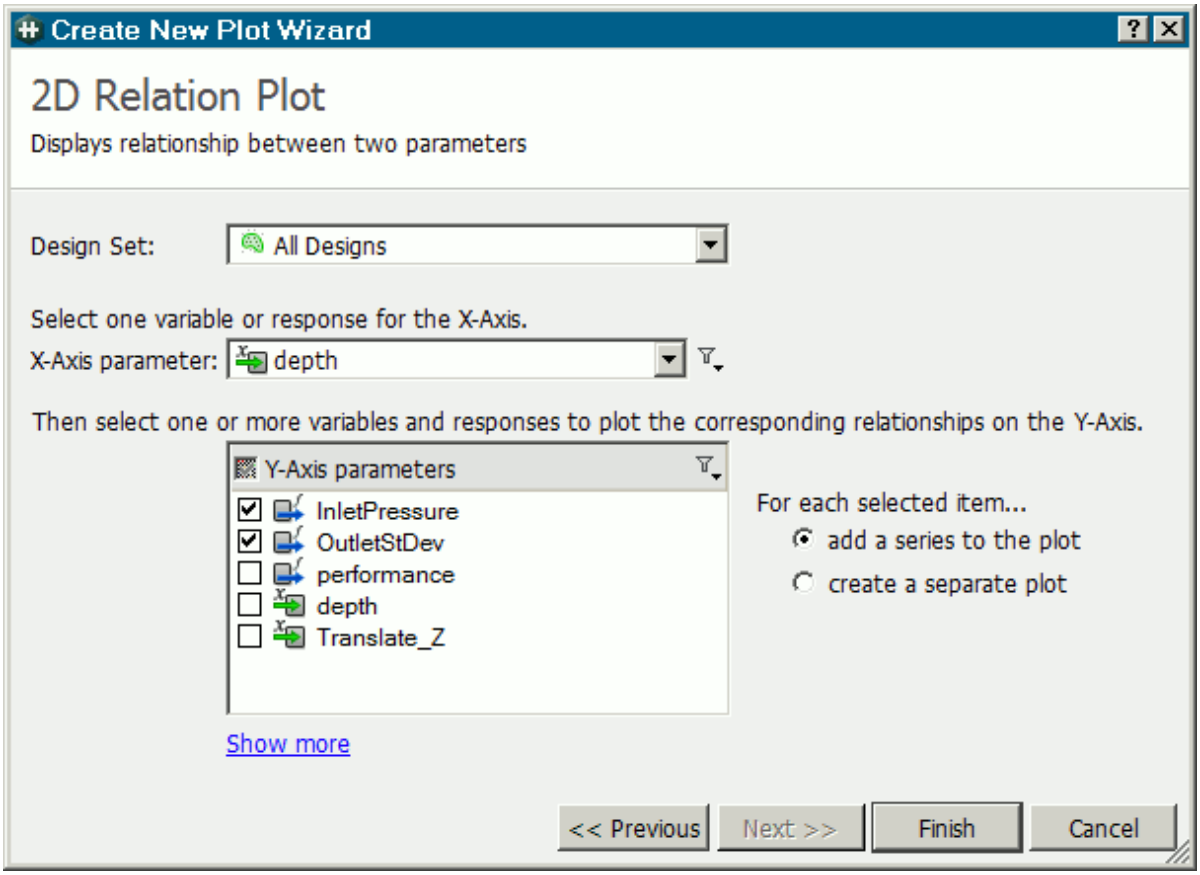
3. From the HEEDS Post ribbon, click the arrow next to the **Create Plot** icon and select **2D Relation**.

   The *Create New Plot Wizard* dialog appears.

4. In the *Create New Plot Wizard* dialog, set:

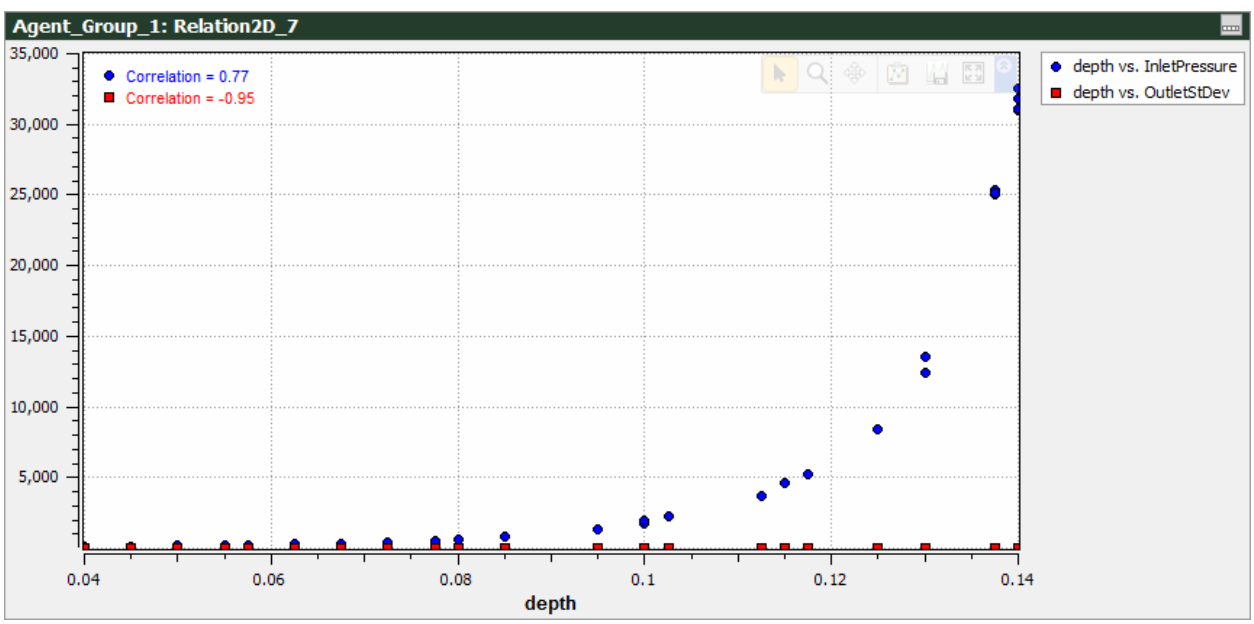
   a) *X-Axis Parameter* to **depth**.

   b) *Y-Axis Parameter* to **InletPressure** and **OutletStDev**.

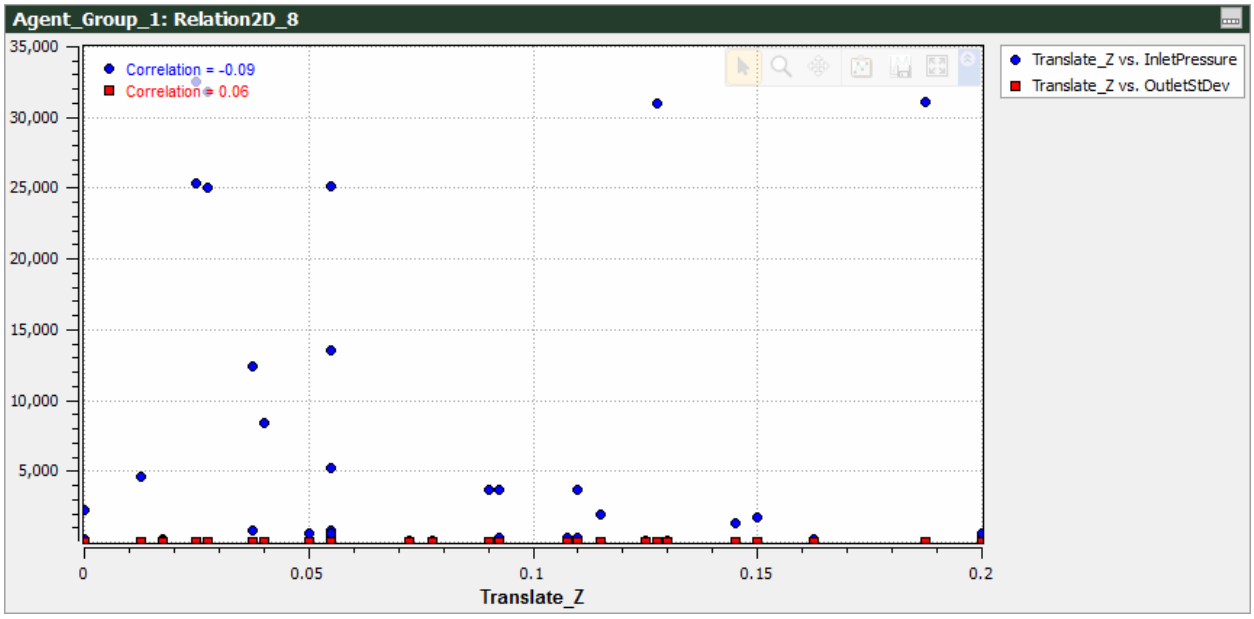


c) Click **Finish**.

The 2D relation plot is created.



5. Repeat the same process, using **TranslateZ** as the *X-Axis Parameter*.



6. Click **File** > **Save** to save the results in HEEDS Post.

Use Existing Designs Tutorial

This tutorial demonstrates how to use Optimate to setup a Design of Experiments tutorial using existing designs.

In this tutorial, you first create a response surface representative of the physical problem, and then conduct a full factorial DoE on the generated response surface to gain sensitivity information about the problem.

Contents:

Prerequisites
Preparing An Existing Designs File
Selecting the Study Type
Using Tabular Data to Define Variables and Responses
Running the Study
Post-Processing

Prerequisites

This tutorial assumes that you are already familiar with certain techniques in STAR-CCM+ and Optimate.

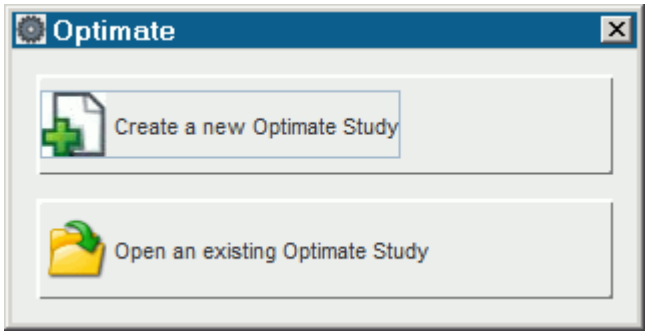
Before attempting this tutorial, make sure that you are familiar with the following techniques:

Techniques	Associated Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+
Running a Pareto optimization study in Optimate	Pareto Optimization Tutorial

This tutorial makes use of the result files from the Pareto Optimization Tutorial. Make sure that you have completed that tutorial first before attempting the current tutorial.

Preparing An Existing Designs File

1. In your workspace, create a folder named `existingDesigns`.
2. Navigate to the `paretoOptimization` folder that you created in the Pareto Optimization tutorial.
3. Copy over the `paretoOptimization.optm` and `pipeMixingBlockage.sim` files, as well as the `pipeMixingBlockage_paretoOptimization` folder, into your `existingDesigns` folder.
4. Launch STAR-CCM+ and load `pipeMixingBlockage.sim`.
5. From the STAR-CCM+ toolbar, click (**Start Optimate**).
   The *Optimate* dialog appears.



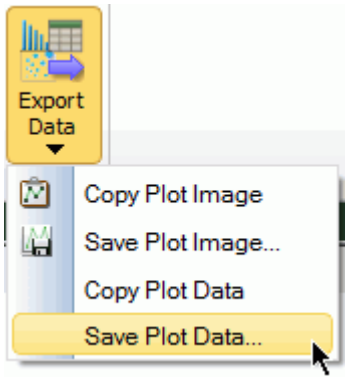
6. Click **Open an Existing Optimate Study**.
7. In the *Select Optimate Study File* dialog, select `paretoOptimization.optm` and click **Open**.

   The Optimate workspace appears and the settings that are contained in the project file are loaded.

8. In the *Run* tab, click (**Open HEEDS|Post**) to launch HEEDS Post.
9. In HEEDS Post, select the **Plots** > **Design Table** node.

   The design table is displayed.

10. From the ribbon, in the **Tools** panel, click **Export Data** > **Save Plot Data...**.



11. In the dialog that appears, save the design data to your workspace as `designTableData.csv`.
12. Close HEEDS Post without saving the changes.

Selecting the Study Type

Modify the run mode so that the project uses the data from the exported design table.

1. Click the *Run Mode* tab.
2. Set *Run Mode* to **Design of Experiments**.
3. Set *Response Surface Modeling* to **Use Existing Designs**.
4. Save the study as `designOfExperiments_existingDesigns.optm`.

Using Tabular Data to Define Variables and Responses

Use csv tabular data to define all the parameters in this study.

1. Click the *Parameters* tab.
2. At the bottom of the tab, delete the **InletPressure** and **OutletStDev** objectives that you defined in the previous tutorial. Do not delete any of the variables or responses.

Create a design set with the data that you exported in a previous section. The data will be used to define the existing variables and responses.

3. Click **Manage Design Sets**.
4. In the *Select a design set for the study* dialog, click (**Add Design Set**) then (**Import from CSV**).
5. In the *Open* dialog, select `designTableData.csv` and set *Lines to skip* to `1`, then click **Import**.
6. Click **Select**.

   All the variables and responses are updated to use data from the design set.

Running the Study

The study is now ready to be run. To start the job on your local machine:

1. Click the *Run* tab.
2. Click (**Run**).

   All necessary files for this study are created.

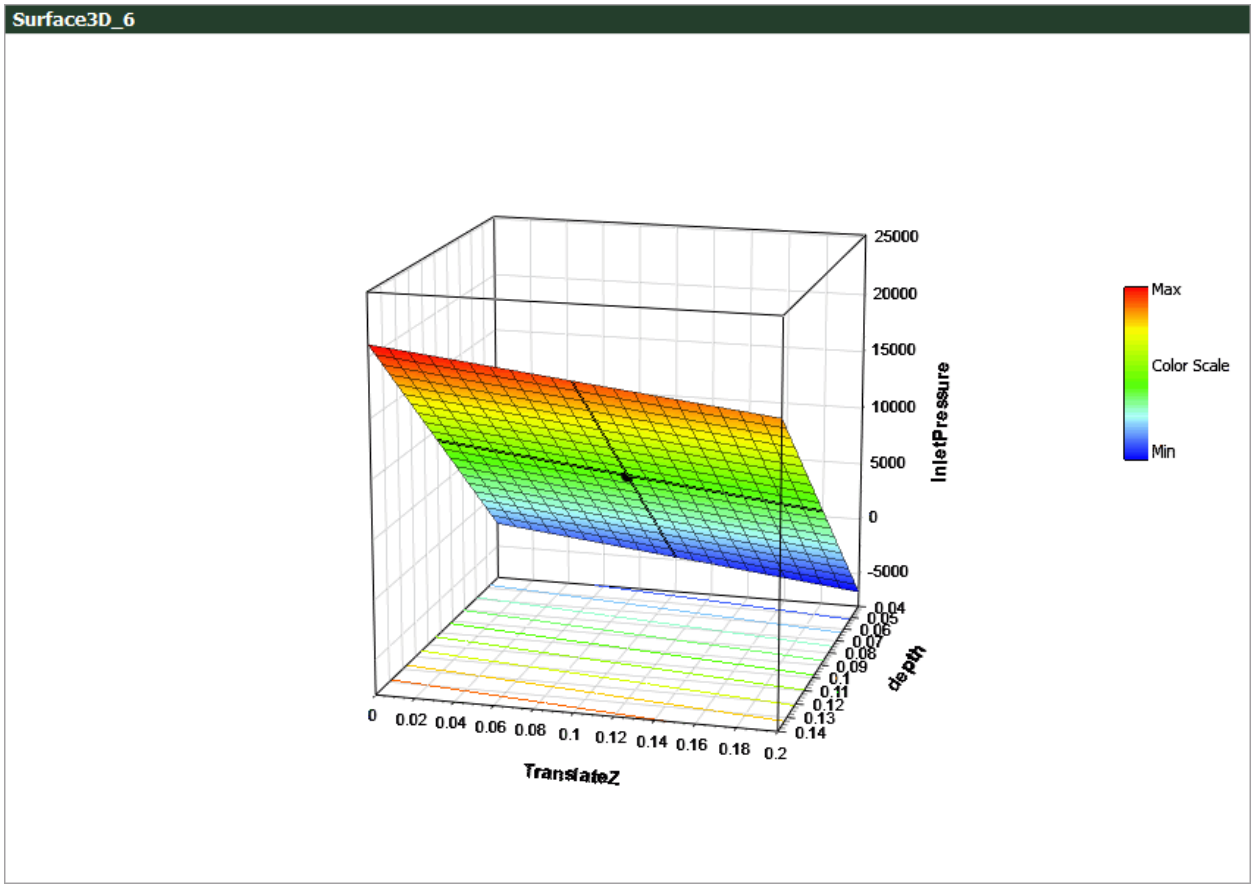
Post-Processing

Visualize the 3D response surface in HEEDS Post.

1. Click (**Open HEEDS|Post**) to launch HEEDS Post.
2. From the HEEDS Post ribbon, in the **DOE Plots** panel, click (**3D Response Surface**).

   The *Create New Plot Wizard* dialog appears.
3. Set *X-Axis Parameter* to **depth**.
4. Set *Y-Axis Parameter* to **TranslateZ**.
5. Select the inlet pressure **InletPressure_Linear** as the response.
6. Click **Finish**.

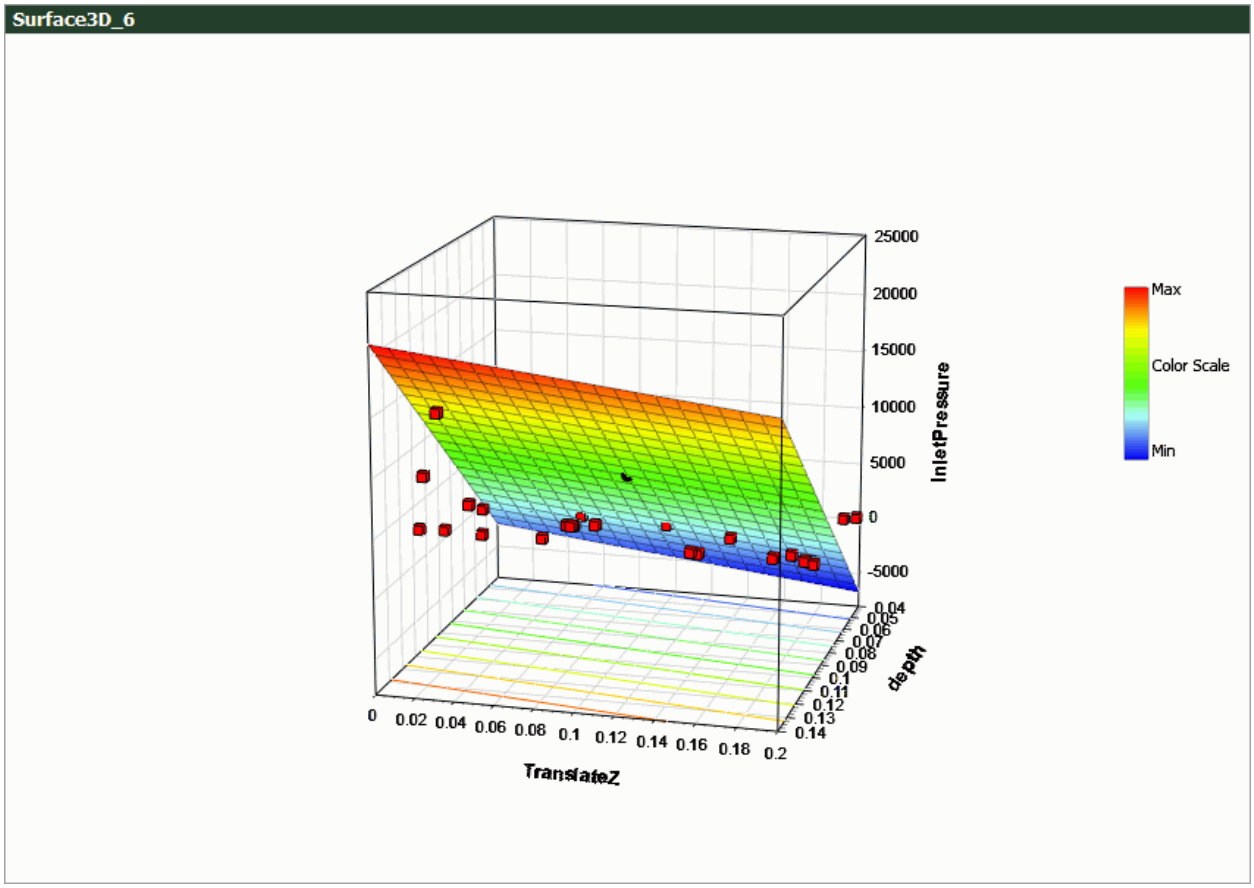
   The 3D response surface is created.



To view the discrete design data overlaid on the surface:

7. From the HEEDS Post ribbon, in the *Series Data* panel, click (**Add New Series**).
8. In the *Add Series to Plot* dialog, set *Type* to **Design data**.
9. Set *Design Set* to **All Designs**.
10. Click **OK**.

    The design data is overlaid on the 3D response surface.


Surface3D_6
25000
20000
15000
10000
5000
0
-5000
InletPressure
0.04
0.05
0.06
0.07
0.08
0.09
0.1
0.11
0.12
0.13
0.14
depth
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2
TranslateZ
Max
Color Scale
Min

Optimization with Macro Tutorial

This tutorial demonstrates how to use Optimate to set up an optimization on a 2D simulation.

As part of the tutorial, you insert a macro into the project to change the run settings and then re-run the solver. You insert the macro into the study so that it runs after the solver has completed running. The macro modifies the mach number from 0.125 to 0.1, extends the number of iterations in the stopping criteria to 1200, and then re-runs the solver. Using this technique, you can perform two runs with different mach numbers for a single design.

Contents:

Prerequisites
Starting Optimate
Inserting the Macro Into the Study
Running the Study
Post-Processing

Prerequisites

This tutorial assumes that you are already familiar with certain techniques in STAR-CCM+ and Optimate.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Techniques	Associated Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+
Recording and editing Java macros	Simple Java Macros: Post-Processing Objects
The Optimate+ workflow	Design Sweeps Pareto Optimization

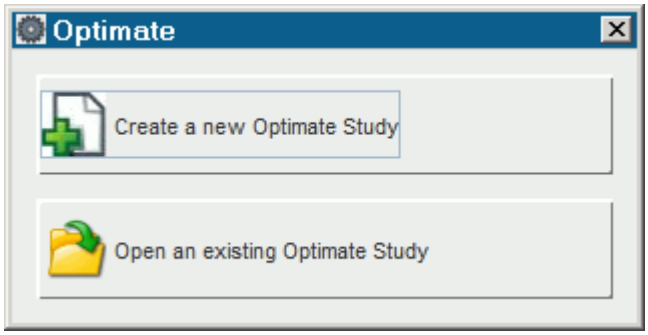
This tutorial requires an Optimate+ license in order to use the **Optimization using SHEPRA** run mode.

Starting Optimate

1. In your workspace, create a folder named `optimizationWithMacro`.
2. Navigate to `[INSTALL_DIR]/doc/startutorialsdata/optimate/data`
3. Copy `airfoil.sim`, `macroOptimization.optm`, and `changeRunSettings.java` to your `optimizationWithMacro` folder.
4. Launch STAR-CCM+.
5. Select **File** > **Load Simulation...**
6. In the *Load Simulation* dialog, click **Browse...**

7. In the *Open* dialog, select `airfoil.sim` and click **Open**.
8. In the *Load Simulation* dialog, click **OK**.
9. From the STAR-CCM+ toolbar, click (**Start Optimate**).

   The *Optimate* dialog appears.



10. Click **Open an Existing Optimate Study**.
11. In the *Select Optimate Study File* dialog, select `macroOptimization.optm` and click **Open**.

    The Optimate workspace appears and the settings that are contained in the project file are loaded.
12. Examine the *Run Mode* and *Parameters* panels to familiarize yourself with the case setup.

Inserting the Macro Into the Study

Insert the Java macro into the workflow so that it changes the mach number and extends the number of iterations before running the solver again.

1. Click the *Assembly* tab.
2. Set *Macro Insertion Point* to **After Running**.

   This property runs the macro after the solver finishes running a design, but before moving on to the next design.
3. Click **Browse** next to *Macro Insertion Point*.
4. In the *Select STAR-CCM+ Macros to be Inserted* dialog, navigate to the `optimizationWithMacro` folder.
5. Select `changeRunSettings.java` and click **Select**.

   The path to this Java macro is automatically added to the *Path* column of the macro insertion panel.
6. Save the study as `macroOptimization_run.optm`.

Running the Study

The study is now ready to be run. To start the job on your local machine:

1. Click the *Run* tab.
2. Click (**Run**).

   All necessary files for this study are created.

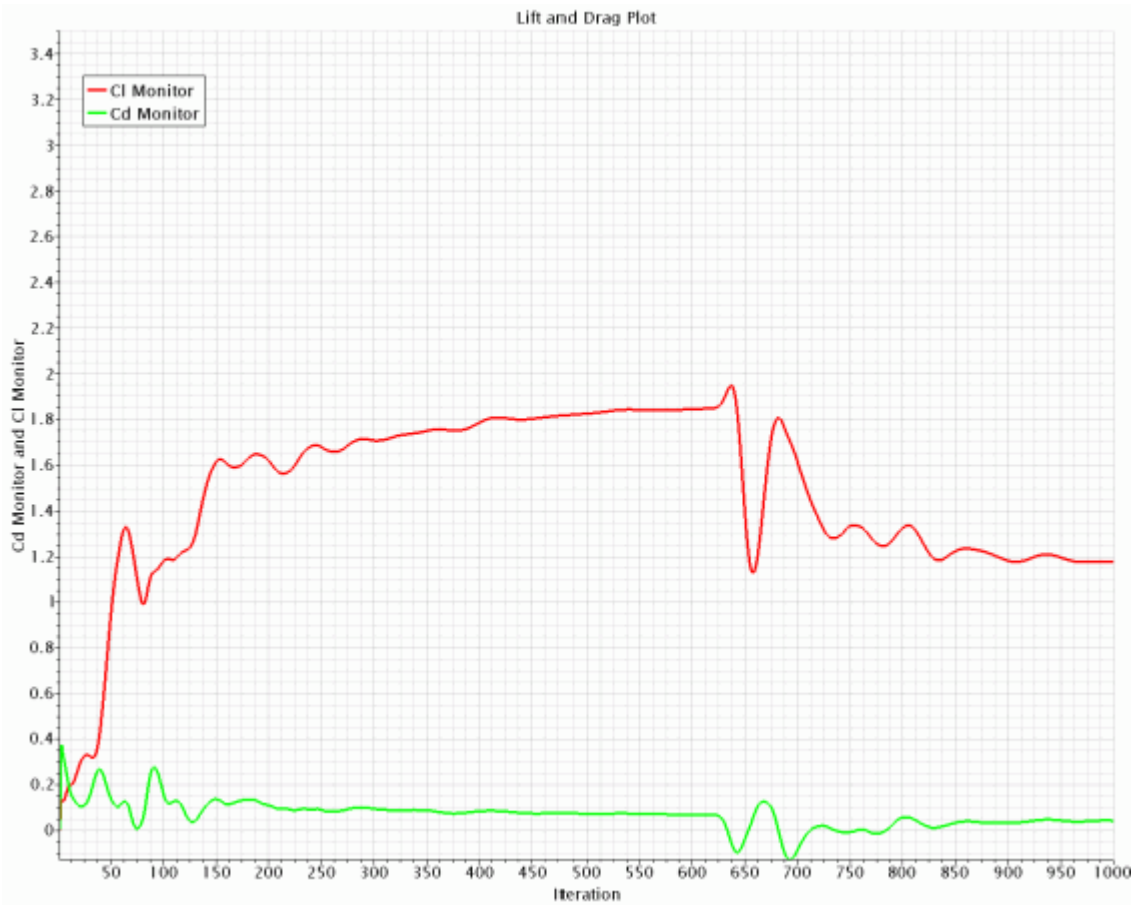
Post-Processing

Check the lift and drag plot to see how these values are affected by the change in mach number.

To access the solution data:

1. Click (**Open HEEDS|Post**) to launch HEEDS Post.
2. In the object tree on the left of the screen, expand the **Plots** folder and select the **star Image** node.
3. From the HEEDS|Post ribbon, in **File** panel, select **Lift_and_Drag-plot.png**.

The lift and drag plot is displayed.



STAR-ICE

STAR-ICE provides a simplified approach within STAR-CCM+ to perform transient moving-mesh simulations of unfired internal combustion engines (ICE). STAR-ICE accesses a subset of STAR-CCM+ features that are relevant to cold-flow simulations

A key feature of STAR-ICE is the provision of Engine Parts. Engine Parts allow you to define the geometry, physics, and motion of your engine components in an efficient and compact way. Engine Parts are available for cylinders and for valves.

STAR-ICE is capable of creating meshes for two-, three-, four-, or five-valve cylinders. The motion of the mesh is modeled using a morph-map approach. This approach redistributes the mesh vertices in response to the movement of the piston and the valves. Based on specific mesh quality criteria, STAR-ICE automatically re-meshes the engine geometry and maps the solution onto the new mesh. The interpolation method conserves mass, momentum, and total energy over the engine volume.

The STAR-ICE user interface contains several ordered nodes that you work through sequentially. In the background, STAR-ICE defines the models and parameters that are required for the analysis in the STAR-CCM+ simulation tree.

You do not require a detailed knowledge of CFD or STAR-CCM+ in order to use the tool. However, you still have access to the full functionality of STAR-CCM+ at any stage if you want to modify your model or account for more complex scenarios.

STAR-ICE is part of the STAR-CCM+ installation. However, you require specific licensing to launch STAR-ICE.

For a practical example of an unfired ICE simulation using STAR-ICE, refer to the following tutorial:

- Gasoline Engine

Inside STAR-ICE

The main features of STAR-ICE are:

Geometry

- Import CAD data

Physics

- Time: transient calculations
- Time-step: constant or stepped table
- Solution domain: single-domain (gas)
- Material: nitrogen (specification of molecular weight)
- Heat transfer modes: conduction and convection (forced and buoyancy driven)
- Flow regime: turbulent (Realizable K-Epsilon Two-Layer All Y+)

- Equation of state: ideal gas

Conditions

- Geometry: full or half-engine (with symmetry)
- Initialization: import tabular pressure and temperature data
- Flow boundaries: pressure outlets using tabular pressure and temperature data
- Wall boundaries: thermal and wall surface specification

Motion

- Cylinder motion: calculates piston position from engine geometry and operating conditions
- Valves motion: import valve lift curves

Mesh

- Volume mesh: trimmed cell mesher (predominantly hexahedral cells)
- Prism layers: prismatic cells on walls for boundary flow accuracy
- Cell size: based on best practices
- Mesh refinement zones for chamfer and valves

Solve

- Segregated solver for flow and energy
- SIMPLE algorithm for pressure-velocity coupling
- Solver under-relaxation
- Live monitoring of solution data

Results

- Visualization of geometry, mesh, and solution data
- Access to solution data through plane sections that are positioned and oriented based on the engine geometry
- Plots for characteristics of the in-cylinder flow, such as swirl, x-tumble, and y-tumble

Note: Currently, STAR-ICE does not support any combustion models.

Contents:

ICE Mechanism
Geometry Requirements
Valves Motion
Piston Motion
Mesh Strategy
Solution Analysis
STAR-ICE Workflow
STAR-ICE Troubleshooting
STAR-ICE Reference

ICE Mechanism

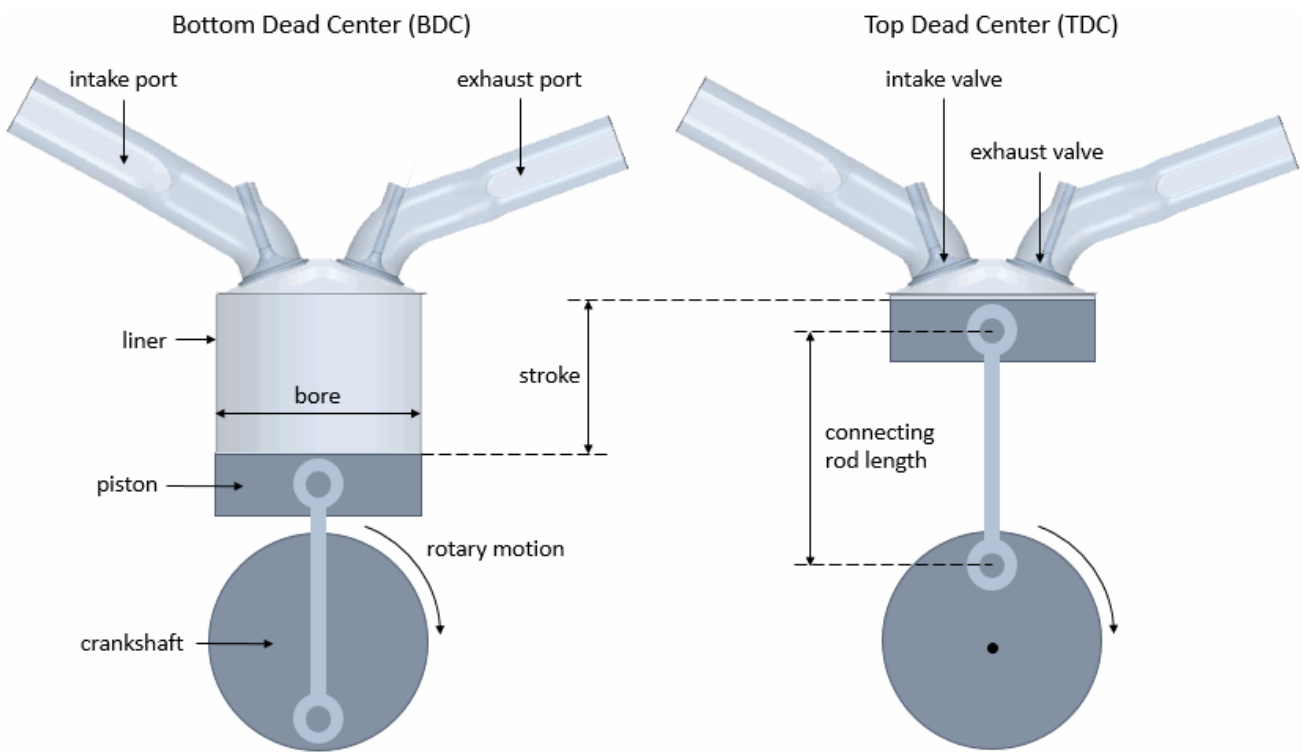
The development and improvement of internal combustion engines comes from a systematic analysis of the in-cylinder flow. STAR-ICE is capable of simulating the in-cylinder flow of reciprocating internal combustion engines that operate on a four-stroke cycle at steady speed.

Note: Currently, STAR-ICE does not support any combustion models.

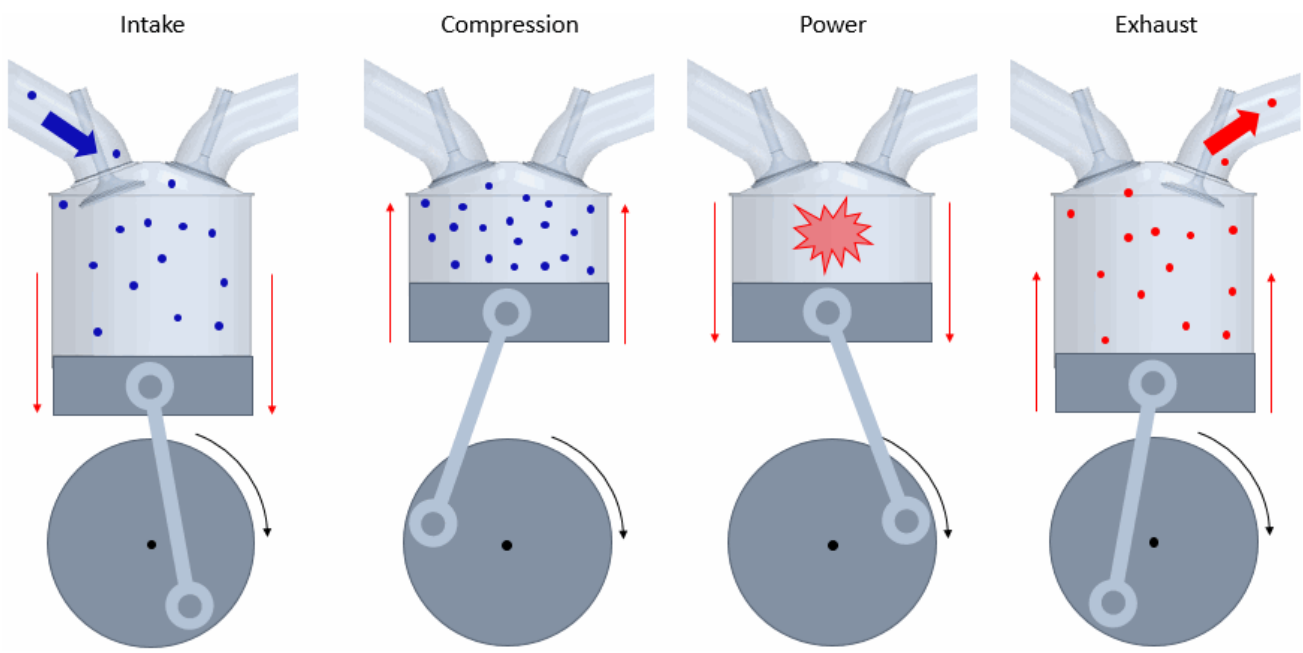
In a reciprocating internal combustion engine, pistons move up and down to transform the expansion of gases due to combustion into rotational motion. A crank mechanism transmits the linear motion of the pistons through a connecting rod to the rotating shaft. Intake valves regulate the inflow of air-fuel mixture through the intake port; exhaust valves similarly regulate the outflow of burned gases through the exhaust port.

The position at which the piston is at its lowest point is called Bottom Dead Center (BDC). When the piston is at its highest point, it is at Top Dead Center (TDC).

The following image displays a schematic diagram of a reciprocating internal combustion engine:



The bore represents the diameter of the engine cylinder; the engine stroke specifies the distance that the piston travels from BDC to TDC. A four-stroke engine is an engine where the piston goes through four strokes while turning the crankshaft twice. The following image displays the strokes of a four-stroke cycle:



Intake

During the intake stroke, the piston moves from TDC to BDC drawing an air-fuel mixture through the intake port into the cylinder. The intake valves are in their open positions.

Compression

In the compression stroke, the piston returns from BDC to TDC compressing the air-fuel mixture. The intake and exhaust valves are closed. At the end of the compression stroke, the crankshaft has completed a complete 360 degree revolution.

Power

The power stroke starts with the ignition of the air-fuel mixture. The expansion of the mixture pushes the piston from TDC to BDC, which forces the crank to rotate. The intake and exhaust valves are closed.

Exhaust

During the Exhaust stroke, the piston returns from BDC to TDC. The exhaust valves are open to release the burned gases through the exhaust port.

At the end of a four-stroke cycle, the crankshaft has completed two complete revolutions (720 degree).

Geometry Requirements

For the automated setup of an internal combustion engine in STAR-ICE, the geometry that you import must meet certain criteria. These criteria concern the file format as well as the positions and names of bodies and faces.

File Formats

STAR-ICE requires that you import the internal combustion geometry from a CAD-based format.

The supported file formats are:

- .x_t/.x_b - Parasolid Transmit files
- .iges/.igs - IGES files
- .step/.stp - STEP files
- .vda - VDA files
- .prt/.asm - Pro/ENGINEER files
- .model/.exp/.session - CATIA V4 files
- .catpart/.catproduct - CATIA V5 files
- .sldprt/.sldasm - SolidWorks files
- .prt - NX files
- .par/.asm - Solid Edge files
- .ipt/.iam - Inventor files
- .3dm - Rhinoceros 3D Model files
- .jt - JT Open B-rep files

CAD Bodies

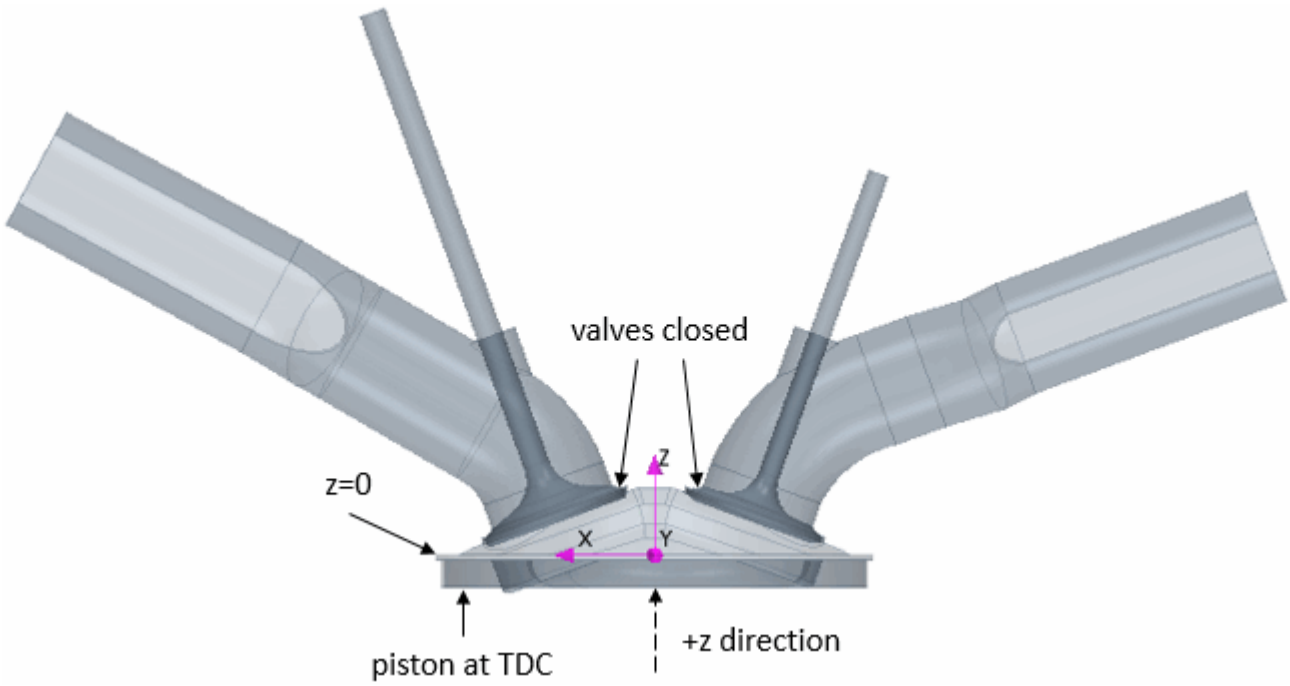
The import process supports geometries that are geometrically and topologically valid from the Parasolid kernel point of view.

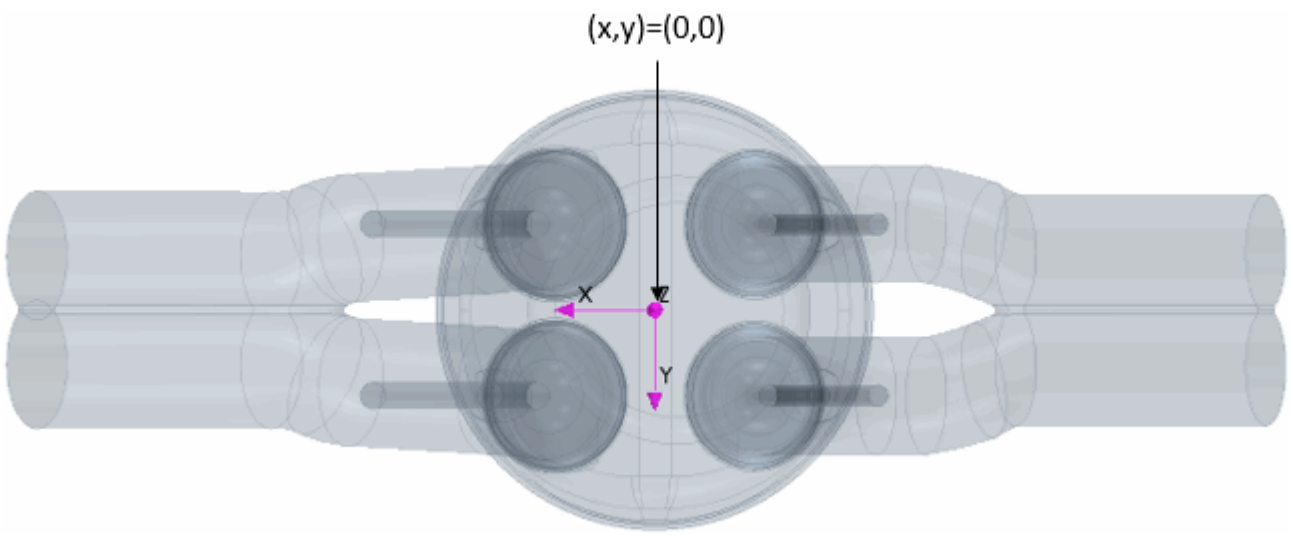
STAR-ICE requires the following CAD bodies:

- One CAD body for the wetted surface of a single cylinder.
- One CAD body for each solid valve. STAR-ICE allows a maximum of three intake valves and two exhaust valves.

The following alignment and orientation of the cylinder and valve geometries are required:

- The cylinder must be positioned so that the piston motion is along the z-axis, with the compression stroke moving the piston in the positive z-direction.
- The x, y coordinates of the cylinder centre must be at x=0, y=0
- The firedeck z coordinate must be at z=0.
- The piston must be at the TDC position.
- All valves must be at their closed position.





Additionally, in your CAD package, you must define the engine geometry in mm. The export tolerance must be 5.0E-5 mm or smaller.

To enable STAR-ICE to recognize the engine bodies, you must name the bodies prior to CAD import in a certain manner. STAR-ICE uses name-based Object Filters that assign the bodies to the respective Engine Parts, automatically. The following table displays the filters and the associated Engine Parts for the CAD bodies that you import:

CAD Body			

Filter: CAD Body Name Contains	`engine`	`intake`
		AND
		`valve`
		AND
		a number (1, 2, or 3)
			`exhaust`
			AND
			`valve`
			AND
			a number (1 or 2)

CAD Body Name Examples	`engine-fluid`	`intake-valve 1` `intake-valve 2`	`exhaust-valve 1` `exhaust-valve 2`
Engine Part	**Cylinder**	**Intake Valve 1** **Intake Valve 2**	**Exhaust Valve 1** **Exhaust Valve 2**

CAD Faces

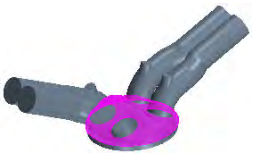
For the cylinder wetted surface and the valves, you must name certain faces prior to CAD import. STAR-ICE uses the face names to assign the faces to the respective Engine Part Surfaces. The default STAR-ICE filters work as follows:

Cylinder

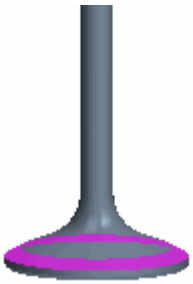
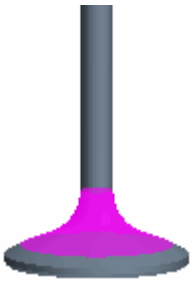
CAD Face				

Filter: CAD Face Name Contains	`intake`
	AND
	`port`
	AND NOT
	`inlet`
		`exhaust`
		AND
		`port`
		AND NOT
		`outlet`
			`inlet`	`outlet`

CAD Face Name Example	`intake-port`	`exhaust-port`	`intake-port inlet`	`exhaust-port outlet`
Cylinder Part Surface	**Intake Port**	**Exhaust Port**	**Inlet**	**Outlet**

CAD Face				-
Filter: CAD Face Name Contains	`liner`	`piston`	`head`	unnamed
CAD Face Name Example	`cylinder liner`	`cylinder piston`	`cylinder head`	-
Cylinder Part Surface	**Liner**	**Piston**	**head**	**Default : Gases**

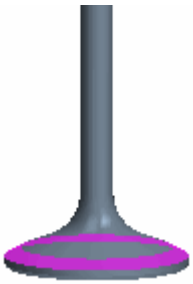
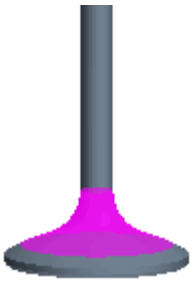
Intake Valve [n]

CAD Face				

Filter: CAD Face Name Contains	`intake`
	AND
	`valve`
	AND
	`head`
	AND
	a number (1, 2, or 3)
		`intake`
		AND
		`valve`
		AND
		`face`
		AND
		a number (1, 2, or 3)
			`intake`
			AND
			`valve`
			AND
			`stem`
			AND
			a number (1, 2, or 3)
				unnamed

CAD Face Name Example	`intake-valve 1 head`	`intake-valve 1 face`	`intake-valve 1 stem`	-
Intake/ Exhaust Valve Part Surface	**Head**	**Face**	**Stem**	**Default : Solids**

Exhaust Valve [n]

CAD Face				

Filter: CAD Face Name Contains	`exhaust`
	AND
	`valve`
	AND
	`head`
	AND
	a number (1 or 2)
		`exhaust`
		AND
		`valve`
		AND
		`face`
		AND
		a number (1 or 2)
			`exhaust`
			AND
			`valve`
			AND
			`stem`
			AND
			a number (1 or 2)
				unnamed

CAD Face Name Example	`exhaust-valve 1 head`	`exhaust-valve 1 face`	`exhaust-valve 1 stem`	-
Intake/ Exhaust Valve Part Surface	**Head**	**Face**	**Stem**	**Default : Solids**

Note: The applied filters are case sensitive. Only use lower case letters for the body and face names.

Valves Motion

STAR-ICE models the motion of the valves using valve lift curves.

A valve lift curve describes the offset of a valve from its closed position as a function of crank angle. For each valve, STAR-ICE requires a valve lift curve in tabular form, where the first column lists the crank angle position in deg and the second column lists the according valve lift in mm.

The following table gives an example of a valve lift table:

```
107,0.0000
108,0.0037
109,0.0112
110,0.0187
111,0.0262
112,0.0337
113,0.0412
114,0.0487
115,0.0562
116,0.0637
117,0.0712
118,0.0787
119,0.0862
120,0.0937
121,0.1012
122,0.1087
123,0.1162
124,0.1238
125,0.1317
...
```

The table must provide data for the duration of the valve lift only and begin and end at zero lift. For the remainder of the cycle, STAR-ICE automatically applies zero lift. When necessary, STAR-ICE automatically applies the appropriate offsets to the crank angle data so that multiple cycles can be handled automatically.

STAR-ICE also allows you to modify the imported tabular data and preview the modified valve lift curve interactively.

The valve lift is calculated as:

$$VL = interpolateTable\left(CA_{cycle} \cdot DM \cdot PA - CA + DM \cdot PA\right) \cdot LM - L \tag{1419}$$

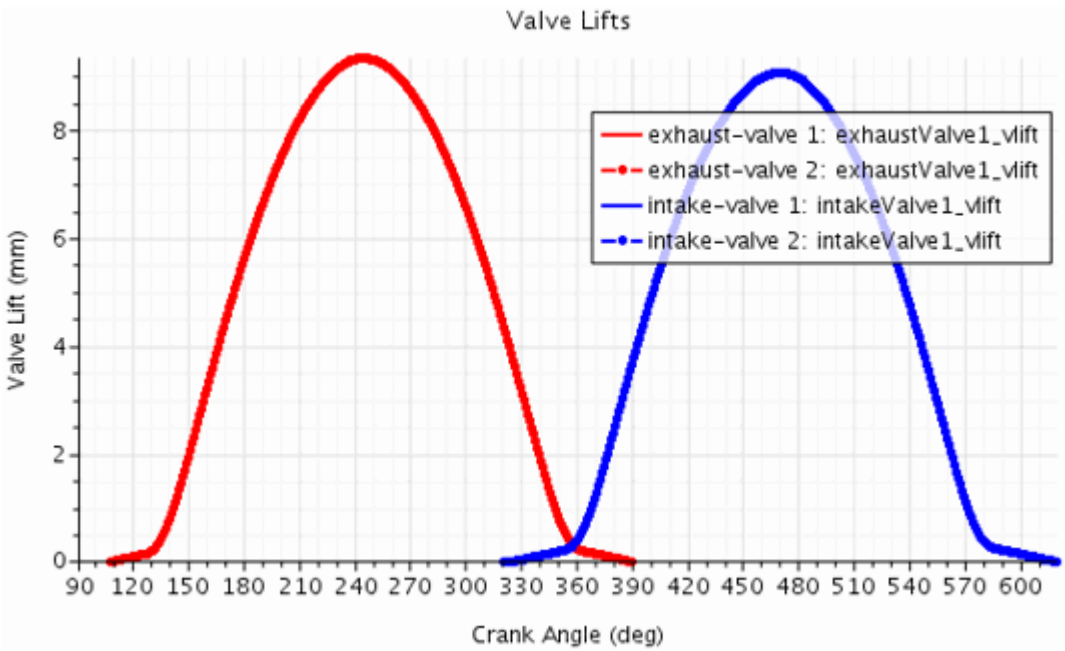
where:

- $interpolateTable()$ linearly interpolates the tabular data.
- CA_{cycle} is the cycle wrapped crank angle in deg (repeats 0 to 720).
- DM is the specified **Duration Mutliplier**.
- PA is the specified **Profile Anchor**.

- CA is the specified **Cycle Anchor**.
- LM is the specified **Lift Multiplier**.
- L is the specified **Lash**.

For information, on the specified values, see Engine Reference—Valve Dialog.

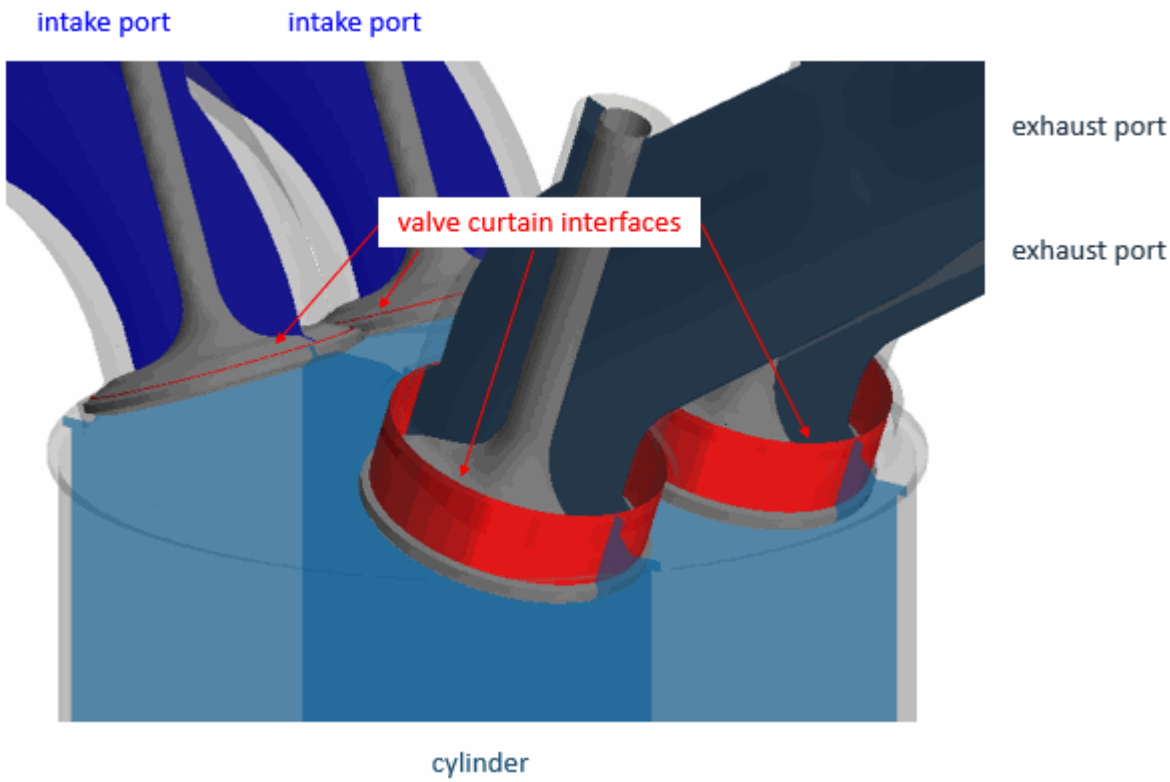
The following image shows an example of valve lift curves for a four-valve cylinder:



Valves Opening and Closing

STAR-ICE controls the opening and closing of the valves using *valve curtain interfaces*. A valve curtain interface is a curved surface area that is located on the Face surface of a valve and separates the cylinder volume from the respective port volume.

The following image shows the valve curtain interfaces for a four-valve cylinder:



STAR-ICE captures the valve opening and closing events at a default valve lift of 0.1 mm. As long as the valve lift exceeds that value, the valve curtain interface allows mass, momentum, and energy to pass from one volume to another. When the valve lift drops to that value or below, the valve curtain interface becomes an impermeable baffle.

Piston Motion

To model the motion of the piston, STAR-ICE requires specific information about the engine geometry.

Using these geometry properties, STAR-ICE moves the piston along the negative z-axis, where the offset from TDC position is calculated as:

$$Piston_z = PPO + S/2 \cdot \cos\left(CA_{cycle}\right) + CRL \cdot \sqrt{1 - \left(\frac{S/2 \cdot \sin\left(CA_{cycle}\right) + PO}{CRL}\right)^2} - \sqrt{(CRL + S/2)^2 - PO^2} \qquad (1420)$$

and:

- PPO is the specified **Piston Position Offset**.
- S is the specified **Stroke**.
- CA_{cycle} is the cycle wrapped crank angle in deg (repeats 0 to 720).
- CRL is the specified **Connecting Rod Length**.
- PO is the specified **Pin Offset**. See Piston Motion Modification.

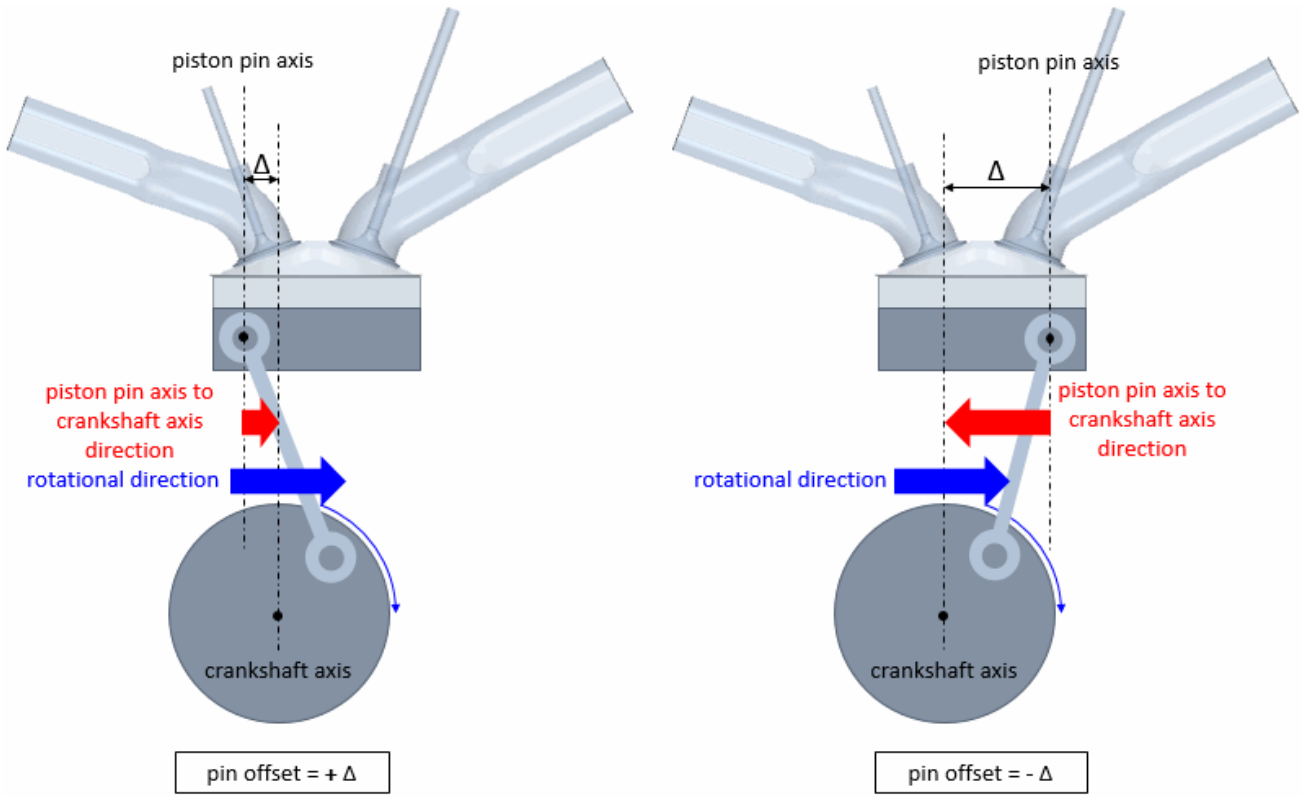
For information, on the specified values, see Engine Reference—Cylinder Dialog.

Piston Motion Modification

STAR-ICE allows you to simulate engines with piston pins and crankshafts that are offset from the cylinder axis. An offset piston pin or crankshaft has the effect of advancing or retarding the piston TDC and BDC position relative to the crank angle.

The pin offset in STAR-ICE is defined as the distance between the piston pin axis and the crankshaft axis. The sign of the pin offset depends on the rotational direction at the upper half of the crank and on the direction from the piston pin axis to the crankshaft axis.

The following image illustrates the convention that is used for the pin offset sign:



If the two directions are the same, then the pin offset has a positive value. If the directions are reverse to each other, then the pin offset sign is negative.

Mesh Strategy

STAR-ICE generates a trimmed mesh with automatic volumetric refinements to account for various flow characteristics within the flow domain.

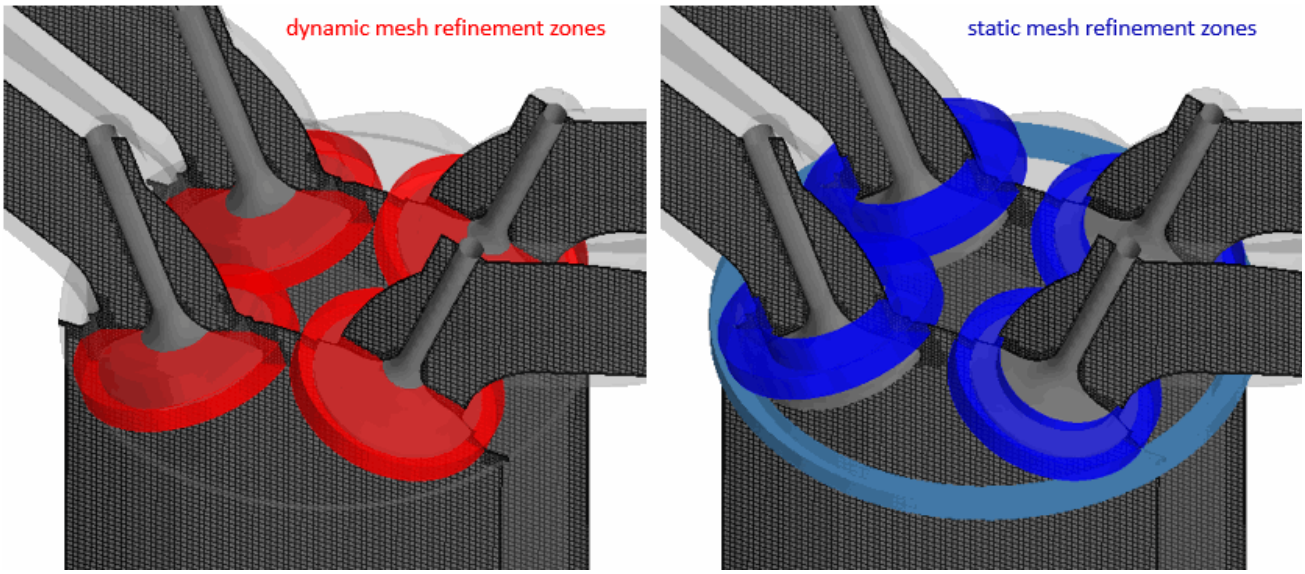
The trimmed cell mesher generates predominantly hexahedral cells. To improve the accuracy of the flow solution near the walls, STAR-ICE generates orthogonal prismatic cells next to wall surfaces.

Additionally, STAR-ICE uses volume shapes to increase the mesh density in specific zones. There are two types of mesh refinement zones:

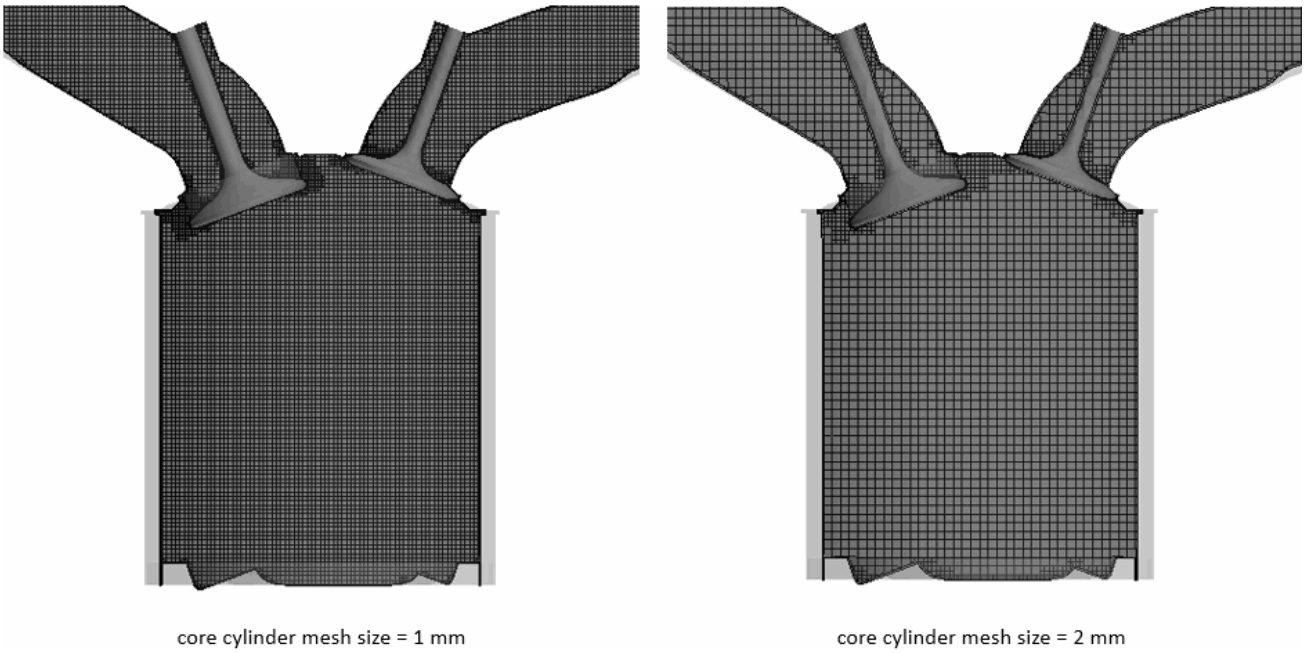
- Dynamic zones follow the moving valves to ensure a good mesh quality around the valves for all crank angle positions.
- Static zones are fixed for all crank angle positions and include the cylinder seat area and the cylinder chamfer.

As the valves have an high impact on the in-cylinder flow, an accurate representation of these areas is important. Fine meshing is applied to avoid poor fluid cell quality at the valve opening and closing times. For the cylinder chamfer, a fine mesh is applied to accurately capture the geometry of the chamfer and to avoid poor fluid cell quality at the time the piston approaches TDC.

The following images display the dynamic and static mesh refinement zones, respectively:



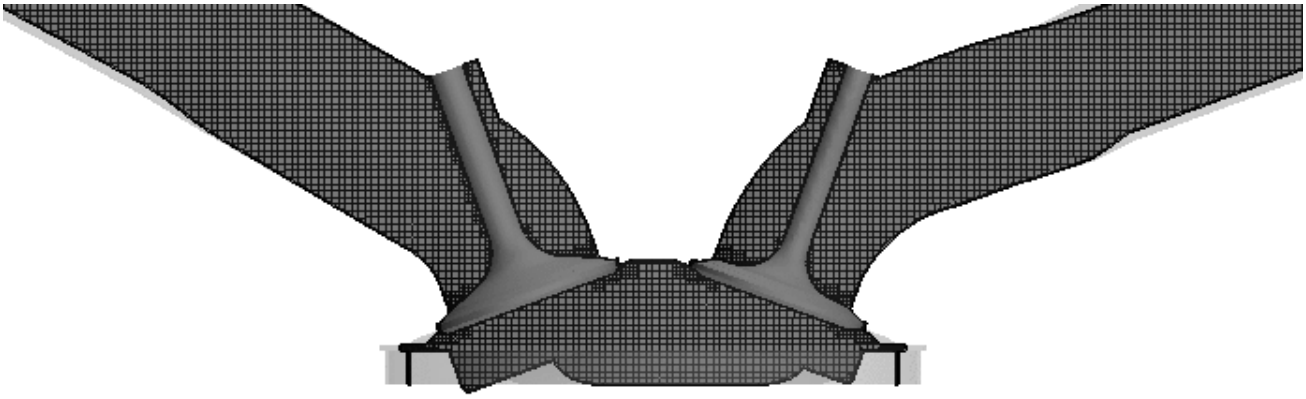
The default mesh settings are based on best practices to ensure a high level of accuracy in the solution with a minimum cell count. However, STAR-ICE allows you to control specific mesh characteristics, such as the cell size in the cylinder core. The following images display the mesh for a gasoline engine using a core cylinder mesh size of 1 mm (default) and 2 mm, respectively:



The mesh size in the refinement zones depends on the core cylinder mesh size. By default, it is half the core cylinder mesh size.

In the simulation of IC engines, the motion of the piston and valves leads to a deformation of the computational grid. STAR-ICE models this grid deformation using the Morpher method, which redistributes the mesh vertices in response to the movement of the piston and valves. As cumulative deformation of the mesh decreases the quality of results, STAR-ICE re-meshes the engine geometry based on specific criteria. For example, STAR-ICE re-meshes the engine when the cell quality drops below a predefined value. The mesh refinement zones dynamically follow the moving valves to ensure a good mesh quality around the valves for all crank angle positions.

The following animation displays the mesh deformation and re-meshing procedure:



Solution Analysis

STAR-ICE allows you to analyse the solution of an ICE simulation while the simulation is running as well as when the simulation completes.

For this reason, you generally create and open post-processing objects before running the simulation

STAR-ICE provides two categories of post-processing object:

- Plots
- Scenes

For both categories, several predefined objects are available that display specific quantities.

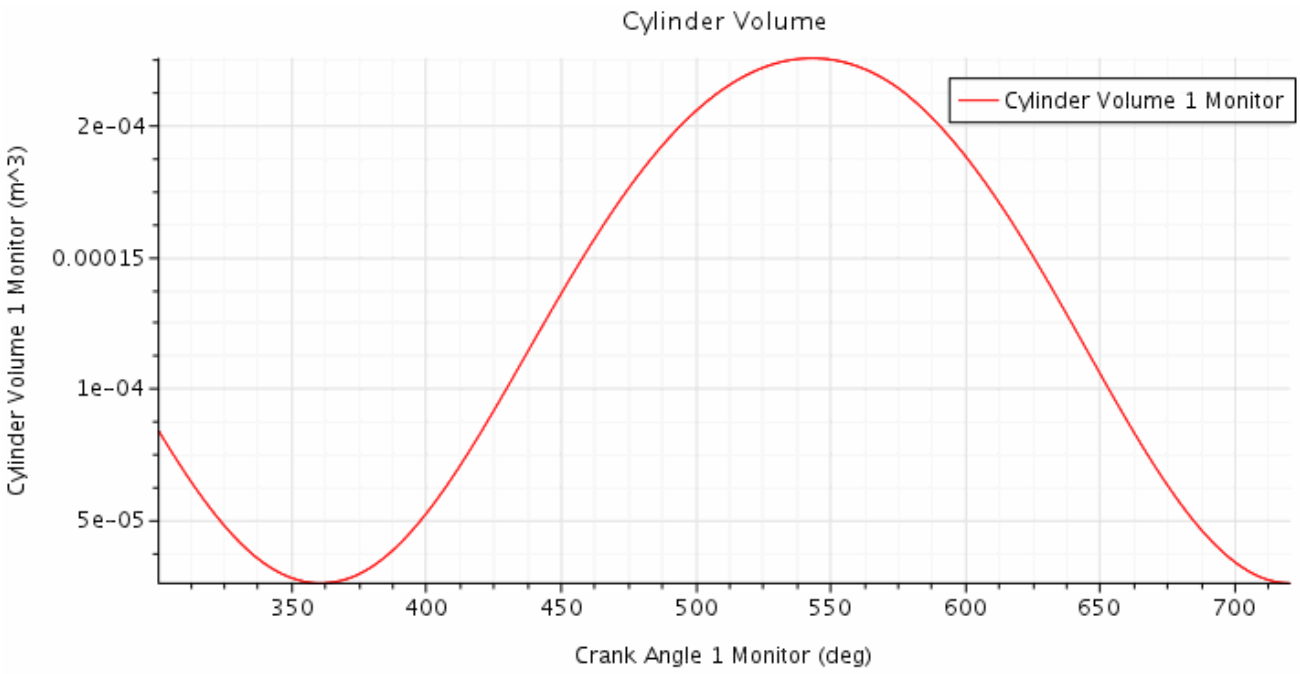
Plots

Plots in STAR-ICE display specific values collected by report-based monitors as functions of crank angle.

The following plots are available:

- **Cylinder Volume**—plots the volume of the cylinder.
- **Cylinder Mass**—plots the mass of the gas in the cylinder, where the mass is calculated as volume integral of density over the cylinder.
- **Cylinder Pressure**—plots the volume average of the absolute pressure inside the cylinder.
- **Cylinder Temperature**—plots the mass average of the temperature inside the cylinder.
- **Cylinder Turbulence Kinetic Energy**—plots the mass average of the turbulent kinetic energy inside the cylinder.
- **Swirl**—plots the swirl inside the cylinder.
- **Tumble-X**—plots the cross-tumble inside the cylinder.
- **Tumble-Y**—plots the main-tumble inside the cylinder.

For example, the following image displays a **Cylinder Volume** plot:



Note: In the context of plots, the term "cylinder" refers to the volume of the cylinder without the ports. This volume is limited by the valve curtain interfaces.

Scenes

Scenes allow you to visualize the geometry, mesh, and solution data that are contained in the simulation.

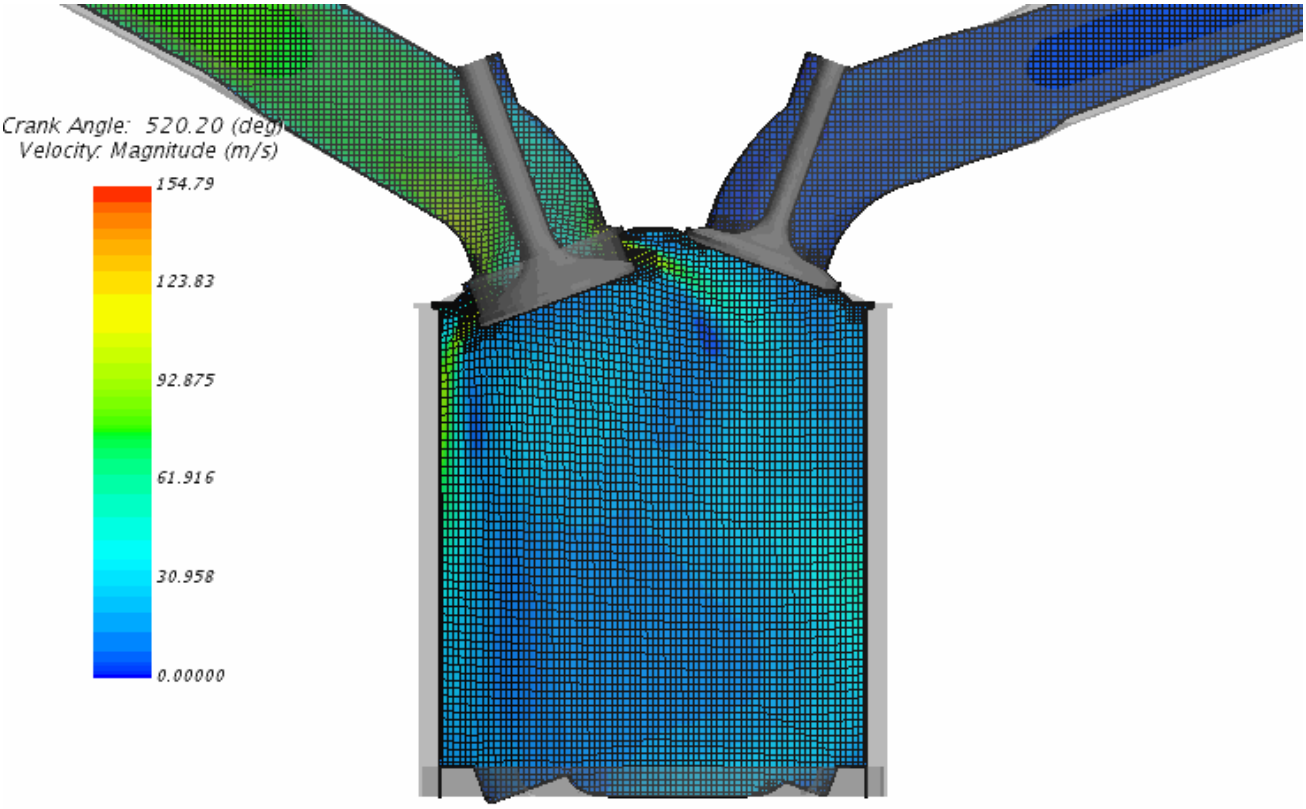
Scenes encapsulate the complete representation of components that are required to generate an image or an animation. Within a scene, displayers bring together parts and other attributes (such as scalars and surface properties) to produce graphics primitives which are displayed in the scene. For more information, see Visualizing the Solution.

STAR-ICE provides a number of predefined scenes. These scenes allow you to examine specific solution fields from which you can draw conclusions about the performance of your engine.

The following predefined scenes are available:

- **Absolute Pressure 1**
- **Temperature 1**
- **Turbulent Kinetic Energy 1**
- **Velocity Magnitude 1**

For example, the following image displays a Velocity Magnitude 1 scene:

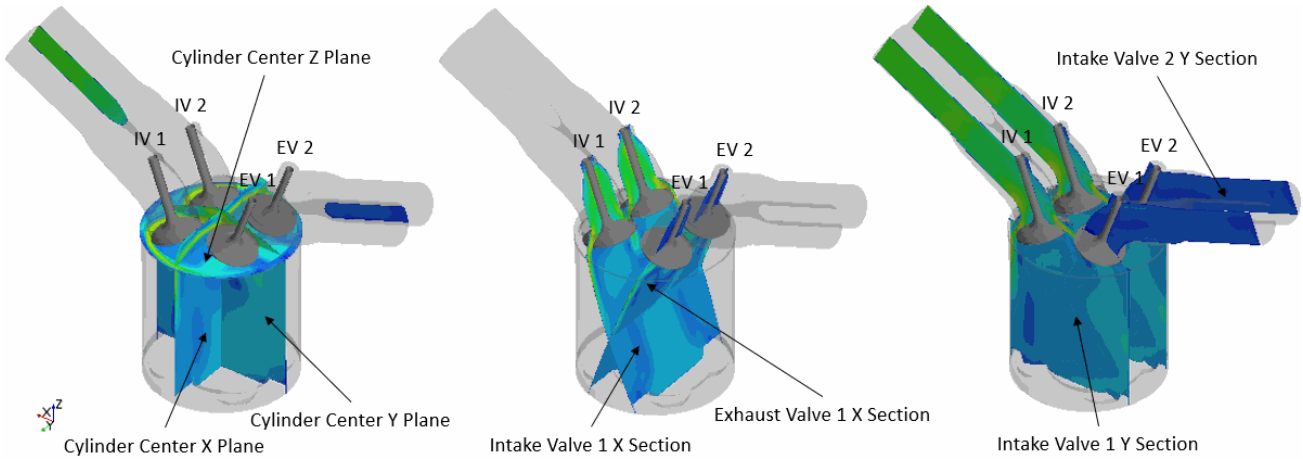


By default, every scene displays the mesh and its respective solution field on a plane that cuts through one of the intake valves and one of the exhaust valves and is oriented normal to the Y-direction. However, STAR-ICE provides additional plane sections, which allow you to access solution data in other areas of the engine domain.

The following plane sections are available:

- **Cylinder Center X Plane**
- **Cylinder Center Y Plane**
- **Cylinder Center Z Plane**
- **Exhaust Valve 1 X Section**
- **Intake Valve 1 X Section**
- **Intake Valve 1 Y Section**
- **Intake Valve 2 Y Section**

The following image shows the location and orientation of the different plane sections for an engine with two intake valves and exhaust valves, respectively:



Additionally, STAR-ICE allows you to display scalar fields on cylinder and valve surfaces.

STAR-ICE Workflow

To set up, run, and post-process an unfired engine cycle problem in STAR-ICE, you work through the nodes from top to bottom.

The main steps required for a STAR-ICE simulation are:

1. Accessing STAR-ICE
2. Importing the Geometry
3. Defining the Setup
4. Defining Parameters
5. Setting up the Engine
6. Setting up the Mesh
7. Preparing the Post-Processing
8. Running the Simulation
9. Analyzing the Results

Contents:

Accessing STAR-ICE
Importing the Geometry
Defining the Solvers Setup
Defining Parameters
Setting up the Engine
Setting Up the Mesh
Preparing for Post-Processing
Running the Simulation
Analyzing the Results
Scripting the Simulation

Accessing STAR-ICE

STAR-ICE is an additional tool whose icon you must add to the STAR-CCM+ interface. After adding the icon, you can launch the tool.

STAR-ICE requires a STAR-ICE add-on license (`enablestarice`). Alternatively, you can use a `proice` feature license together with an `enablereactivemodels` feature license.

To activate the STAR-ICE add-on license in STAR-CCM+:

1. Open STAR-CCM+ and start a new simulation.

   To reduce the time it takes the solvers and meshers in STAR-ICE to run an ICE simulation, you are advised to start the simulation in parallel on a multi-core machine. For more information, see Launching STAR-CCM+.
2. In the STAR-CCM+ menu, select **Tools** > **Add-ons**.

3. In the *Add-on Licensing* dialog, activate *STAR-ICE*.
4. Click **OK**.

To automatically check out the STAR-ICE add-on license each time you start or load a simulation:

5. In the STAR-CCM+ menu, select **Tools** > **Options**.
6. In the *Services* dialog, select **Licensing** and activate **Reserve STAR-ICE License**.
7. Click **Close**.

To access STAR-ICE:

8. In the STAR-CCM+ menu, select **Window** > **Toolbars** > **Customize**.
9. In the *Customize Toolbars* dialog, under the **Wizards** node, select **Open STAR-ICE**.
10. Drag the (**Open STAR-ICE**) icon onto one of the toolbars at the top of the STAR-CCM+ workspace and release it.
11. In the *Customize Toolbars* dialog, click **Close**.

    The STAR-CCM+ GUI now displays the (**Open STAR-ICE**) button.
12. Click (**Open STAR-ICE**).

    The STAR-ICE dialog appears in which you set up your ICE simulation.

Contents:

Saving, Closing, and Loading a simulation in STAR-ICE

Saving, Closing, and Loading a simulation in STAR-ICE

STAR-ICE is fully integrated within STAR-CCM+—all setup and solution data are stored within the `.sim` file.

Note: In this context, unless otherwise stated, the term "ICE simulation" refers to a simulation that is set up with STAR-ICE.

Objective	Procedure
Saving an ICE simulation	1. Click (**Save**). 2. In the *Save* dialog, enter the name and location where you want to save the file. 3. Click **Save**. You can save an ICE simulation at any stage.
Closing an ICE simulation	1. Click the **x** button next to the STAR-ICE simulation in the *Explorer* pane.
Loading an ICE simulation	1. Launch STAR-CCM+ and click (**Load Simulation...**). 2. In the *Load Simulation* dialog, click **Browse...**.

Objective	Procedure
	3. In the *Open* dialog, navigate to the working directory, select the appropriate `.sim` file, and click **Open**. 4. Complete the *Load Simulation* dialog according to the properties described in Interactive Startup Reference and click **OK**. 5. In the STAR-CCM+ toolbar, click (**Open STAR-ICE**).

Importing the Geometry

The setup of an internal combustion engine starts from a geometry in CAD format.

Before you import the CAD geometry, make sure that you have the bodies and faces prepared as described in Geometry Requirements.

To import the CAD geometry:

1. Right-click the **Geometry** node and select **Import CAD Model**.
2. In the *Open* dialog, navigate to the folder that contains the geometry file.
3. Select the file and click **Open**.

   For more information, see Geometry Reference—Geometry Right-Click Actions.

   The geometry objects, which correspond to the imported CAD bodies, appear in the tree. The *Graphics* window updates and displays the geometry objects in orange color. In the bottom left corner of the *Graphics* window, the **Crank Angle** annotation displays a value of 0, which corresponds to the TDC position of the imported geometry.

Defining the Solvers Setup

Follow these steps to define the solver settings, the auto-save settings, the operating conditions, and the lift closure tolerance for the valve motion.

To edit the solvers setup:

1. Right-click the **Setup** > **Solvers** node and select **Edit**.
2. In the *Engine Solver Settings* group box, set the following properties:
   - The type of time-step control, such as **Crank Angle Step**
   - **Molecular Weight of Gas**
   - **Segregated Flow** under-relaxation factors
   - **Segregated Energy** under-relaxation factors

   See Setup Reference—Engine Solver Settings.

To save separate copies of the simulation file automatically at regular intervals during the run:

3. In the *Auto-Save Settings* group-box, activate **Enabled** and set the following properties:.

- The trigger for the auto-save operation, such as **Number of Time Steps**
- **Number of Files to Keep**

See Setup Reference—Auto-Save Settings

4. In the *Operating Conditions* group-box, set the following properties:
   - **RPM**
   - **Start Angle**
   - **Duration**
   - **Reference Pressure**

   See Setup Reference—Operating Conditions.
5. In the *Valve Motion* group-box, set **Lift Closure Tolerance** to the valve lift that corresponds to a closed position.

   See Setup Reference—Valve Motion.
6. Click **Apply**, then **Close**.

The solvers settings are now defined. In the background, the selected conditions and parameters are set in the STAR-CCM+ simulation tree.

Defining Parameters

A global parameter is a scalar physical quantity that can be defined once and used to set multiple properties in your simulation.

One application of a global parameter is in defining a key physical characteristic as a reference value. Other values can point directly to this reference value or be defined in terms of it. For example, in your internal combustion engine, the lash for some of the valve lift curves could be set once as a global parameter, and then be used by the respective valve lift curves.

Another important use of global parameters is the integration of STAR-ICE with Design Manager—global parameters can act as controls that you adjust over multiple design simulations within a design study.

Note: Only SI units are available for global parameters.

To set up a global parameter:

1. Right-click the **Parameters** node and select **Edit**.

   The *Edit* dialog gives you a summary of some pre-defined global Parameters and their default values. You are advised to not use these pre-defined Parameters as they will be removed in future versions.
2. In the *Edit* dialog, click **Add a new parameter**.
3. In the group-box, set the following properties:
   - **Parameter**
   - **Dimensions**
   - **Value**

   For more information, see Parameters Reference.
4. Click **Add**.

   The parameter is now available for other properties in your ICE simulation.

To edit a parameter:

5. Select the text-box of the parameter property whose value you want to change and set the value to suit your requirements.

To delete a parameter:

6. Click ▬ next to the parameter entry.
7. Click **Close**.

To apply a global parameter, follow a procedure similar to the example outlined below:

8. Right-click a node that stores a scalar physical quantity, such as **Engine** > **Exhaust Valve 1**, and select **Edit**.
9. In the *Edit* dialog, set the value of the respective scalar quantity, such as *Lash*, to `$[Parameter]`, where [Parameter] is the name of the global parameter that you created.
10. Click **Apply**, then **Close**.

Setting up the Engine

STAR-ICE automatically creates Engine Parts based on your inputs. Along with these parts, you define the physics and motions that apply to the engine.

1. Right-click the **Engine** node and select **Create Cylinder and Valves**.
2. Within the *Engine Parts Creation* dialog, in the *Engine Configuration* group-box, set the number of **Intake Valve(s)** and **Exhaust Valve(s)** of the imported engine geometry, respectively.
3. If the engine geometry allows you to use symmetry conditions, you can include only half the geometry, thus reducing the cell count and computational cost of the simulation:
   a) In the *Engine Symmetry* group-box, activate **Half Model**.
   b) For **Symmetry Plane**, select the side of the geometry to keep.

   For more information, see Engine Reference—Engine Parts Creation Dialog
4. Click **OK**.

   STAR-ICE creates the respective Engine Parts:

   - **Cylinder**
   - **Exhaust Valve [n]**
   - **Intake Valve [n]**

   The *Graphics* window updates to display the Engine Parts (in pale blue color) at the **Start Angle** that you specified in Defining the Setup.

   These Engine Parts allow you to set the physics for the cylinder and the valves:

   - Set Initial Conditions
   - Set up the Piston
   - Set up the Valve Motions
   - Set Boundary Conditions

Contents:

Setting Initial Conditions
Setting up the Piston
Setting up the Valve Motions
Setting Boundary Conditions

Setting Initial Conditions

You specify the initial absolute pressure and temperature conditions in the cylinder and ports using tables that describe these values as functions of crank angle. STAR-ICE evaluates the initial values at the specified start angle using linear interpolation between the data points. For the initial turbulence quantities, you specify constant values.

For the initial absolute pressure and temperature conditions in the cylinder and the ports, use relevant and verified engine data if available. Such data can be derived from experimental results or a 1-D simulation. If no such data is available, initialize the engine components as follows:

- Cylinder: Use the exhaust boundary conditions. This recommendation assumes that the simulation starts close to the end of the exhaust phase.
- Intake port: Use the intake boundary conditions.
- Exhaust port: Use the exhaust boundary conditions.

The supported file formats are `.csv`, `.txt`, `.dat`.

You can describe the initial absolute pressure and temperature values in a single table or in separate tables.

Turbulence is generated around the valves during the intake process and hence the solution is not a strong function of initial turbulence values. The best way of initializing turbulence parameters is to utilise the results of a 3D simulation of the gas exchange process that runs over multiple cycles. If such a simulation is not available, you can try a strategy based on the rpm, stroke, and bore values, plus the following assumptions:

- Calculate the mean piston velocity from $v^p_{mean} = 2 \cdot stroke \cdot rpm/60$
- Estimate the velocity fluctuation from $v' = 0.5 \cdot v^p_{mean}$ (it has been found that a factor of 0.5 gives results that are close to the 3D simulation).
- Estimate the turbulence kinetic energy from $k_{init} = \frac{3}{2} \cdot (v')^2$
- Specify the turbulence length scale as $l_{init} = 0.025 \cdot bore$
- Calculate the turbulence dissipation rate from $\varepsilon_{init} = C_\mu^{3/4} \cdot \frac{k_{init}^{3/2}}{l_{init}}$, where $C_\mu = 0.09$.

To set initial conditions:

1. Right-click the **Engine** > **Cylinder** node and select **Edit**.
2. Within the *Edit* dialog, in the *Initial Conditions* tab, set the **Turbulence Specification** for the cylinder and the ports, such as **K + Epsilon**.

To set initial conditions for the cylinder:

3. In the *Cylinder* group-box, set the initial **Pressure** using tabular data:

a) Next to the **File** drop-down menu, click **Import**.

b) In the *Import Table* dialog, navigate to the folder that contains the table.

c) Select the file and click **Open**.

In the *Graphics* window, a plot opens that displays the absolute pressure in the cylinder as a function of crank angle.

d) Set the following table properties:

- **Crank Angle Column**
- **Absolute Pressure Column**
- **Crank Angle Units**
- **Pressure Units**

4. Set the initial **Temperature** data:

a) Depending on whether the temperature data are described in the same or in a separate table as the pressure data:

- If the temperature data are described in the same table, from the **File** drop-down menu below the **Temperature** property, select the previously imported table.
- If the temperature data are described in a separate table, import the table as described for the pressure table using the **Import** button below the **Temperature** property.

The plot in the *Graphics* window now displays the absolute pressure and temperature curves for the cylinder.

b) Set the following table properties:

- **Crank Angle Column**
- **Temperature Column**
- **Crank Angle Units**
- **Temperature Units**

5. To determine the absolute pressure and temperature values that STAR-ICE uses for initialization, hover the mouse over the respective curve and stop at the specified crank angle.

The coordinates are displayed next to the pointer.

6. Set the initial values for the turbulence quantities, such as **Turbulent Dissipation Rate** and **Turbulent Kinetic Energy**.

To set initial conditions for the ports, repeat Steps 3 - 6 for the *Intake Port [n]* and *Exhaust Port [n]* group boxes, where [n] specifies the number of the respective port.

For more information, see Engine Reference—Initial Conditions.

7. Click **Apply**, then **Close**.

Setting up the Piston

Here, you set the engine geometry properties that impact the piston motion and the cylinder displacement and compression ratio.

To set the geometry properties of the engine:

1. Right-click the **Engine** > **Cylinder** node and select **Edit**.

2. the *Edit* dialog, click the *Motion* tab.
3. In the *Engine Geometry* group-box, set the following properties:
   - **Bore**—STAR-ICE calculates this value from the imported geometry. If the value is not calculated precisely, replace it with the exact value.
   - **Stroke**
   - **Connecting Rod Length**
   - **Pin Offset**

   See Engine Reference—Cylinder Dialog: Motion.

To check if your setup is correct, you can calculate the displacement and compression ratio of the cylinder and compare the results to measurements.

To calculate the displacement and compression ratio:

4. Click **Apply**, then **Update Compression Ratio**.

   STAR-ICE updates the geometry in the background to calculate the cylinder volume at TDC position with the valves closed. This value is required for the compression ratio calculation.

   The **Displacement** and **Compression Ratio** values for the cylinder are displayed.
5. If the displayed values do not match your measurements, check your geometry properties again and correct them accordingly.

If the piston of your imported geometry is not at TDC position, do the following:

6. In the *Piston Motion* group-box, set **Piston Position Offset** to the offset of the imported piston geometry from TDC.
7. Click **Apply**, then **Close**.

The *Engine Geometry* properties (except for **Bore**) and the **Piston Position Offset** define the motion of the piston. When you run the simulation, STAR-ICE uses these properties to calculate the piston position for the respective crank angle and updates the geometry accordingly. However, as long as you do not start the simulation, the *Graphics* window displays the piston position using the default values. In the bottom left corner, the red-colored starred text indicates that the displayed piston is not at its correct position for the specified start angle. Additionally the warning triangle ( ) on the **Engine** > **Cylinder** node indicates that this Engine Part requires an update.

8. If you want to visualize the correct piston position for the **Start Angle** that you specified in Defining the Setup, click **Update STAR-ICE**.

Setting up the Valve Motions

For the motion of the valves, you import tables that describe the valve lifts in mm as functions of degree crank angle. If the imported table values do not represent the real valve lifts, you can correct the valve lift curves here.

To define the motion for an exhaust valve:

1. Right-click the respective **Engine** > **Exhaust Valve [n]** node and select **Edit**.
2. In the *Edit* dialog, next to the **Valve Lift File** drop-down menu, click **Import**.
3. In the *Import Valve Lift Table* dialog, navigate to the valve lift table.

4. Select the table and click **Open**.

   In the *Graphics* window, a *Valve Lifts* plot opens, which displays the valve lift as a function of crank angle.
5. To correct the valve lift curve , set the following properties:
   - **Lash**
   - **Cycle Anchor**
   - **Profile Anchor**
   - **Lift Multiplier**
   - **Duration Multiplier**

   See Engine Reference—Valve Dialog.

   The *Valve Lifts* plot updates to display the corrected curve.
6. Click **Apply**, then **Close**.

The imported valve lift table and the specified properties define the motion of the valve. When you run the simulation, STAR-ICE uses these data to calculate the valve position for the respective crank angle and updates the geometry accordingly. However, as long as you do not start the simulation, the *Graphics* window displays the valve in its closed position. In the bottom left corner, the red-colored starred text indicates that the displayed valve is not at its correct position for the specified start angle. Additionally the warning triangle ( ) on the **Engine** > **Exhaust Valve [n]** node indicates that this Engine Part requires an update.

7. If you want to visualize the correct valve position for the **Start Angle** that you specified in Defining the Setup, click **Update STAR-ICE**.

To define the motion for an intake valve:

8. Right-click the respective **Engine** > **Intake Valve [n]** node and select **Edit**.
9. Define the valve lift using the same procedure as for an exhaust valve.

   The *Valve Lifts* plot displays the lift curves for all valves.

Setting Boundary Conditions

Boundary conditions define the physical conditions on the surface of the cylinder and valves.

By default, the gas enters through the intake port inlet and exits through the exhaust port outlet, where both boundaries are modeled as pressure outlets. This setting results in the specification of an absolute pressure inflow condition and a static pressure outflow condition. You should therefore make sure that the values that you specify correspond to absolute pressure.

The inlet and outlet boundary conditions can have significant effect on the flow predictions. Therefore, if available, use relevant and verified engine data in tabular form to set both absolute pressure and temperature as functions of crank angle. The supported file formats are `.csv`, `.txt`, `.dat`.

You can describe the absolute pressure and temperature values in a single table or in separate tables. If no engine data are available for temperature, you can set a constant value instead.

Engine data for turbulence boundary values are usually not available, but reasonable settings are 10% for the turbulence intensity and 10% of the hydraulic diameter for the length scale.

Wall boundaries are modeled as smooth walls with a constant temperature of 500 K, by default. However, STAR-ICE also allows you to model rough walls. For the thermal specification, you can also specify the temperature on the wall as a function of crank angle or apply adiabatic conditions.

To set the boundary conditions for flow boundaries:

1. Right-click the **Engine** > **Cylinder** > **Inlet [n]** node and select **Edit**.
2. Within the *Edit* dialog, in the *Boundary* group box, set the **Turbulence Specification** for the boundary, such as **Intensity + Length Scale**.
3. Set the temperature depending on whether engine data for the temperature are available or not:

Temperature Engine Data	Procedure
Available	1. In the *Static Temperature* group-box, next to the **File** drop-down menu, click **Import**. 2. In the *Import Table* dialog, navigate to the folder that contains the table. 3. Select the file and click **Open**. 4. In the *Graphics* window, a plot opens that displays the static temperature as a function of crank angle. 5. Set the following table properties: • **Crank Angle Column** • **Temperature Column** • **Crank Angle Units** • **Temperature Units**
Not available	1. In the *Static Temperature* group-box, from the top drop-down menu, select **Constant**. The default constant temperature is 500 K. 2. To set a specific temperature value, delete the `$DefaultAmbientTemperature` entry and set another value.

4. To set the pressure:
   a) Depending on whether the pressure data are described in the same or in a separate table as the temperature data:
      - If the pressure data are described in the same table, in the *Pressure* group-box, select the previously imported table from the **File** drop-down .
      - If the temperature data are described in a separate table, import the table as described for the temperature table using the **Import** button in the *Pressure* group-box.

      The plot in the *Graphics* window now displays the temperature and pressure curves.
   b) Set the following table properties:
      - **Crank Angle Column**
      - **Absolute Pressure Column**
      - **Crank Angle Units**
      - **Pressure Units**

5. Set the boundary values for the turbulence quantities, such as **Turbulence Intensity** and **Turbulent Length Scale**.
6. Click **Apply**, then **Close**.
7. To set the boundary conditions at the outlet, repeat these steps for the **Engine** > **Cylinder** > **Outlet [n]** node.

To change the default boundary conditions at a wall:

8. If you want to set different boundary conditions on specific faces of the Engine Part Surfaces **Cylinder** > **Default : Gases** or **Intake/Exhaust Valve [n]** > **Default : Solids**, separate those faces into their own Engine Part Surface first.

   See Splitting an Engine Part Surface.
9. Edit the **Engine** > **[engine part]** > **[wall]** node, where [engine part] is the respective Engine Part, such as **Cylinder**, and [wall] is the Engine Part Surface for which you want to change the wall boundary conditions, such as **Liner**.
10. In the *Edit* dialog, set the following boundary conditions:
    - **Thermal Specification**
    - **Wall Surface Specification**
11. Set the boundary values depending on the specified conditions.

    For example, if you set **Thermal Specification** to **Temperature** and **Wall Surface Specification** to **Rough**, you set values for the **Static Temperature** and the **Roughness Height**.

    See Engine Part Surface Reference.

Contents:

Splitting an Engine Part Surface

Splitting an Engine Part Surface

In STAR-ICE, you can split the Engine Part Surfaces **Default : Solids** and **Default : Gas** for use in defining specific boundary conditions or customizing the mesh.

You can split these Engine Part Surfaces in one of the following ways:

- Creating a new Engine Part Surface using the underlying face information.

  Example: A block has only one surface, by default, but it has six faces—one for each side of the block (this information is stored as part of the geometry). Each face can be used to create a new Engine Part Surface.
- Creating a new Engine Part Surface from modified face information.

  Example: To create a circular Engine Part Surface on a block, you must first modify the underlying face information by imprinting a circular profile on the relevant side surface. This action creates a circular face on the respective side of the block. The circular face can then be used to create a new circular Engine Part Surface.

To create a new Engine Part Surface from a specific face:

1. In the *Graphics* window, select the face of the respective **Default : Solids** or **Default : Gases** surface.

   To select multiple faces, press and hold down the `<Ctrl>` key.

2. In the STAR-ICE tree, right-click and select **Create QuickPart Surface**.
3. In the *Rename* dialog, enter the **Name** of the new Engine Part Surface and click **OK**.

   The new Engine Part Surface appears under the Engine Part node.

To modify the face information of an Engine Part Surface:

4. In the *Graphics* window, select the face of interest.
5. Right-click and select **New Construction Geometry** > **[Type]**, where [Type] is one of the following:
   - Sheet Body Circle
   - Sheet Body Rectangle

   For more information, see Creating Geometries.
6. Right-click the respective Engine Part node and select **Edit Interactions**.
7. Within the *Edit* dialog, in the *Imprint* box, set **Sheet Construction Geometry** to the previously created sheet geometry.

   For more information, see Interactions Reference.
8. Click **Apply**, then **Close**.

   The profile is imprinted and the face information is modified accordingly. You can now use the resulting faces to create a new Engine Part Surface.

Setting Up the Mesh

The default mesh settings are based on best practices to ensure a high level of accuracy in the solution with a minimum cell count.

For that reason, in most cases, you can maintain the default settings. However, if you encounter convergence problems due to insufficient mesh resolution or if you want to increase the accuracy, you can edit the mesh settings.

To edit the mesh settings:

1. Right-click the **Mesh** node and select **Edit Mesh Settings**.

   The warning triangle ( ) on the **Mesh** node indicates that you must still generate the mesh (after defining the settings and any custom controls).
2. In the *Edit* tab, define the following global mesh and boundary layer settings to suit your requirements:
   - **Core Cylinder Mesh Size**
   - **Normalized Valve Curtain Interface Position**
   - **Minimum Size**
   - **Fluid Boundary Layer**

   See Mesh Reference.

   For example, decreasing the **Core Cylinder Mesh Size** decreases the target cell size in the core of the cylinder, which increases the mesh resolution and with it the accuracy of the solution. However, this increase in the accuracy of the solution also comes with an increase of computational cost.
3. To commit changes to the default mesh settings, click **Apply**.

During the running simulation, STAR-ICE re-meshes the geometry based on certain re-mesh criteria to ensure a good cell quality for all crank angles. These re-mesh criteria are also based on best practices and are suitable in most cases. However, you can change the criteria if you wish.

To edit the re-mesh criteria:

4. In the *Re-mesh Criteria* group-box, set the following properties:
   - **Min Cell Quality**
   - **Max Piston Stretching**
   - **Max Valve Stretching**
   - **Max Valve Displacement**
   - **Piston Tolerance**
   - **Min Cell Quality Reference**. This property displays the minimum cell quality after the last re-mesh. DO NOT modify this value.

   See Mesh Reference—Re-mesh Criteria.
5. Click **Apply**, then **Close**.
6. If you want to customize the mesh on specific faces of the Engine Part Surfaces **Cylinder** > **Default : Gases** or **Intake/Exhaust Valve [n]** > **Default : Solids**, first separate those faces into their own Engine Part Surface. See Splitting an Engine Part Surface.
7. To set customized mesh settings on a specific Engine Part Surface:
   a) Right-click the **Mesh** > **Custom Controls** node and select **New** > **Surface Control**.
   b) In the *Edit* dialog, set **Surfaces** to the respective Engine Part Surface and change the mesh settings as required. See Mesh Reference—Surface Control.
   c) Click **Apply**, then **Close**.

STAR-ICE defines mesh refinement zones around the chamfer and the valves, which dynamically follow the moving valves. You can create additional refinement zones to accurately capture flow phenomena in specific areas of the cylinder or the ports. However, in contrast to the predefined mesh refinement zones, Volumetric Controls that you create can not move with the valves or the piston

8. If you want to customize the mesh within an additional volumetric zone, define a geometry that encloses the zone of interest and set up a Volume Control:
   a) Create an arbitrary three-dimensional volume shape. See Creating Geometries.
   b) Right-click the **Mesh** > **Custom Controls** node and select **New** > **Volume Control**.
   c) In the *Edit* dialog, set **Parts** to the created Construction Geometry and change the mesh settings as required. See Mesh Reference—Volume Control.
   d) Click **Apply** and **Close**.

STAR-ICE automatically generates the mesh as part of the simulation run. However, if you want to generate and check the mesh before running the simulation, you can do so.

To generate the mesh manually:

9. In the *Mesh Generation* toolbar (see Toolbar Reference), click (**Generate Volume Mesh**)

   The progress of the mesh generation is displayed in the *Output* window. Once the progress is complete, the warning triangle on the **Mesh** icon disappears.

Note: When you change mesh settings after the meshing progress is complete, the warning triangle indicates that the mesh is no longer up to date and requires re-execution.

Contents:

Creating Geometries

Creating Geometries

In STAR-ICE, you can create additional geometries for use in splitting existing surfaces within the engine geometry, or for defining mesh refinement zones.

To create geometries, the following options are available:

- Build the geometry based on simple shape templates
- Import the geometry from another CAD package
- Create the geometry within the 3D-CAD modeler and transfer it to STAR-ICE

To operate on created or imported geometrical shapes, STAR-ICE supports the following options:

- Apply boolean operations
- Generate patterns of objects

To build a geometry based on simple shape templates:

1. Right-click the **Geometry** node and select **New Construction Geometry** > **[Type]**, where [Type] is one of the following:
   - Block
   - Sphere
   - Cylinder
   - Sheet Body Rectangle
   - Sheet Body Circle

   Alternatively, you can create a Construction Geometry on the surface of another geometry object that is visible in the *Graphics* window. Usually, creating a Construction Geometry interactively reduces the time for placing the Construction Geometry on a previously created geometry object:

   1. In the *Graphics* window, select a suitable face of the object.
   2. Right-click and select **New Construction Geometry** > **[Type]**.

   The position of the mouse pointer when selecting the face determines the position of the initial local origin of the new Construction Geometry.

An *Edit* dialog appears that displays the name, geometry, and placement properties of the Construction Geometry.

To define the properties of the Construction Geometry:

2. For **Name**, enter a description of the Construction Geometry under which it is stored in the tree.
3. In the *Geometry* and *Placement* sections, define the shape and position of the geometry as required. See Shape Templates Reference.
4. If you want to preview your geometry and placement settings, click **Update** at the bottom of the dialog.

5. Click **OK**.

To import a geometry from another CAD package:

6. Right-click the **Geometry** node and select **Import CAD model**
7. In the *Open* dialog, navigate to the folder that contains the CAD model.
8. Select the CAD file and click **Open**.

To create a geometry within 3D-CAD and transfer it to STAR-ICE:

9. Click the *Simulation* tab.

   The STAR-CCM+ tree appears.
10. Right-click the **Geometry** > **3D-CAD Models** > **STAR-ICE 3D-CAD 1** node and select **Edit...**.
11. Build the respective geometry in 3D-CAD, see Using 3D-CAD.
12. When you are done with creating the geometry, click **Close 3D-CAD**.
13. Click the *STAR-ICE* tab.
14. Right-click the **Geometry** node and select **Transfer CAD Bodies.**
15. In the *Select Cad Bodies For Transfer* dialog, select the respective CAD body and click **OK**.

To edit the properties of a Construction Geometry:

16. Right-click the respective Construction Geometry node and select **Edit**.

    The *Edit* dialog launches that allows you to change the name, geometry, and placement properties as required.

To apply a boolean operation on a previously created geometry:

17. Right-click the node of the respective Construction Geometry and select **Boolean Construction Geometry**.
18. In the *Edit* dialog, set the geometry and placement properties as required. See Boolean Operations Reference.
19. Click **OK**.

To generate a pattern of identical geometries:

20. Right-click the node of the Construction Object that you want to multiply and select **Pattern Construction Geometry**.
21. In the *Edit* dialog, set the properties to suit the required pattern. See Boolean Operations Reference.
22. Click **OK**.

Preparing for Post-Processing

STAR-ICE provides several plots and scenes by default. You can analyze the scenes and plots while the simulation is running, as well as when it completes.

If you open and prepare the scenes and plots before you run the simulation, they are updated continuously as the solution progresses:

- Open a plot
- Open a scene

- Display the solution on additional plane sections in a scene
- Display the solution on Engine Part Surfaces in a scene
- Create a new scene
- Write a scene to image files at regular intervals

To open a plot:

1. Right-click the **Results** > **Plots** > **[plot]** node and select **Show**, where [plot] is the plot of interest, such as **Cylinder Mass**.

   The plot opens in the *Graphics* window.

To open a scene:

2. Right-click the **Results** > **Scenes** > **[scene]** node and select **Open**, where [scene] is the scene you want to display, such as **Velocity Magnitude 1**.

   The scene opens in the *Graphics* window and displays the mesh and the respective scalar field on a section that cuts through the engine domain.

3. If you want to hide the mesh, click (**Hide Al Meshes**).

   Clicking the button again restores the mesh on the plane section.

STAR-ICE provides several plane sections, which allow you to access solution data on various areas in the engine domain.

To display the scalar field on additional plane sections:

4. Right-click the **[scene]** > **Displayers** > **[scene] Derived Parts Scalar 1** > **Parts** > **[plane section]** node and select **Show**, where [plane section] is the plane section of interest, such as **Cylinder Center X Plane**.
5. To hide a plane section from a displayer, right-click the [plane section] node of interest and select **Hide**.

STAR-ICE also allows you to display scalar fields on cylinder and valve surfaces:

6. Depending on the Engine Part Surfaces on which you want to display a scalar field, select one of the following nodes:
   - For valve surfaces, select the respective **[scene]** > **Displayers** > **Intake/Exhaust Valve [n] Displayer 1** > **Scalar Field** node.
   - For the piston surface, select the **[scene]** > **Displayers** > **Cylinder Piston Surface Displayer 1** > **Scalar Field** node.
   - For cylinder surfaces, apart from the piston surface, select the **[scene]** > **Displayers** > **Cylinder Displayer 1** > **Scalar Field** node.
7. Set *Function* to the scalar field that you want to display, such as **Temperature**.

   The scene displays the selected scalar field on the surfaces of interest. For cylinder surfaces, including the piston surface, the scalar field is displayed with reduced opacity to reveal internal structures.

To obtain a view of internal structures, you can render a displayer visible or invisible in the *Graphics* window:

8. Right-click the respective **[scene]** > **Displayers** > **[displayer]** node and select **Toggle Visibility**.

If you require additional scenes to analyse your simulation results, you can create new scenes. Before creating scenes, be clear on what quantities you want to analyze and from which objects you want to extract the quantities.

To create a new scene:

9. Right-click one of the pre-defined scenes and select **Copy**.
10. Right-click the **Scenes** manager node and select **Paste**.
11. Modify the scene displayers to suit your analysis requirements, see Using Displayers.

To write a scene to image files at regular intervals:

12. Follow the steps described in Creating Images for an Animation.

For more information, see Visualizing the Solution.

The setup of your ICE simulation is now complete.

13. Save the simulation .

Running the Simulation

The simulation is now ready to run.

To decrease the time required for completing an ICE simulation, you are advised to run STAR-ICE simulations in parallel on multi-core machines.

To run the simulation:

1. Depending on whether you want to run the simulation from the GUI or from a batch client, do one of the following:

Method	Procedure
Running directly from the GUI	1. Right-click the **Engine** node and select **Run Solution**. **Note:** DO NOT click (**Run**) in the *Solution* toolbar. See Toolbar Reference. Clicking this button runs the simulation for one time-step only.
Running from a batch client For more information, see Running the Client in Batch.	1. Save the following java code to a macro file (`.java`) and name it runSTARICE:

```
// STAR-CCM+ macro: runSTARICE.java
package macro;

import java.util.*;

import star.starice.*;
import star.common.*;
import star.base.neo.*;
import star.quickobjects.*;

public class runSTARICE extends StarMacro {

  public void execute() {
    execute0();
```

```
  }

  private void execute0() {

    Simulation simulation_0 = getActiveSimulation();

    simulation_0.loadStarIce("StarIce");

    StarIceEngine starIceEngine_0 =
simulation_0.get(StarIceEngine.class);

    StarIcePhysicsManager starIcePhysicsManager_0 =
starIceEngine_0.getStarIcePhysicsManager();

    Units units_0 = ((Units)
simulation_0.getUnitsManager().getObject("K"));


starIcePhysicsManager_0.setDefaultAmbientTemperature(500.0,
units_0);

    QuickSimulationIterator quickSimulationIterator_0 =
starIceEngine_0.getQuickSimulationIterator();

    quickSimulationIterator_0.run();
  }
}
```

2. Move the macro file to your working directory, which contains the STAR-ICE simulation file (`.sim`).
3. Open a command prompt and enter the following command line:

   ```
   [STAR-CCM+_INSTALL_DIR]\star\bin\starccm+ -batch
   [PARALLEL_OPTIONS] [WORKING_DIR]\runSTARICE.java [WORKING_DIR]
   \[SIMULATION].sim
   ```

   where:

   - `[STAR-CCM+_INSTALL_DIR]` represents the path to the STAR-CCM+ installation directory.
   - `[PARALLEL_OPTIONS]` represents optional arguments for running the simulation in parallel. See Parallel Options.
   - `[WORKING_DIR]` represents the path to your working directory.
   - `[SIMULATION]` represents the name of the simulation file.

   This command line only applies on Windows. For Linux, you must use forward slashes.

STAR-ICE updates the geometry to represent your settings and starts the meshing process. After the mesh is complete, the simulation starts. During the run, STAR-ICE re-meshes the geometry automatically when it meets one of the default re-mesh criteria.

Analyzing the Results

After the run is finished, you analyze the simulation results.

View and plot the results of the converged simulation:

- Investigate the plots
- Examine the scenes

To investigate the plots:

1. To open a plot, right-click the plot node and select **Show**.
2. To save a plot to a plot image file:
   a) Open the plot.
   b) In the *Graphics* window, right-click the plot display and select **Hardcopy...**.
   c) Use the *Save to File* dialog to save the plot display. For details, see Saving a Plot Image File.
   d) Click **Save**.
3. To tabulate the data sets of a plot, in the *Graphics* window, right-click the plot display and select **Tabulate**.

   The data sets appear in a spreadsheet-like dialog.
4. To export the data sets of a plot to a `.csv` file:
   a) In the *Tabular Data* dialog, click **Export**.

      Alternatively, you can directly export the plot by right-clicking the plot display in the *Graphics* window and selecting **Export...**.
   b) In the *Save* dialog, enter the file name and click **Save**.

To examine the scenes:

5. If you wrote a scene to image files at regular intervals, click through the saved files to visualize the solution field at different crank angles. The base name of each image file is combined with the number of the update to ensure that each file is unique. The saved image files can be animated with a media program of your choice.
6. To save a scene to a scene image file after the run:
   a) In STAR-ICE, select the scene node and select **Open**.
   b) In the *Graphics* window, right-click the background of the visualization display and select **Hardcopy...**.
   c) Use the Save to File dialog to save the scene display. For details, see Saving Scene Image Files.
   d) Click **Save**.

Scripting the Simulation

You can script the process of setting up and running a simulation in STAR-ICE.

Similar to STAR-CCM+ (see Scripting the Application), scripting in STAR-ICE is particularly useful when you want to automate repetitive tasks or when you want to perform certain actions within the project when running in batch (locally or on a cluster).

Example: You want to run the same analysis for different engine geometries. In that case, for the first geometry, you record all the steps that you perform in STAR-ICE for setting up and running the analysis. To analyse another geometry, you can then import the geometry into a new STAR-ICE project and run the macro.

To script an ICE simulation:

1. Before recording a macro, be clear on how you want to play the macro.
   - If you want to play the macro from STAR-CCM+ or when running in batch, record the step when clicking (**Open STAR-ICE**).
   - If you want to run the macro from STAR-ICE, do not record clicking (**Open STAR-ICE**).
2. In STAR-ICE, record the steps that you want to include in the macro. See Recording a Macro.
3. Play the macro. See Playing the Macro.

STAR-ICE Troubleshooting

If you encounter problems when setting up or running an ICE simulation, check these common problem scenarios for advice on how to proceed. Also consult the Knowledge Base on the customer success portal.

Errors about Invalid CAD

When creating Engine Parts from imported Geometry parts, various errors about invalid CAD data can occur. In most cases, these errors indicate problems with the imported CAD model.

- Check that all bodies are valid from the Parasolid kernel point of view.
- In your CAD tool, ensure that the tolerance is set to `5.0E-5 mm` or smaller when exporting the CAD geometry.
- Try to increase or decrease the Normalized Valve Curtain Interface Position in order to accurately reflect the valve contact.
- When importing the CAD geometry, try to deactivate the Remove Redundant Topology option.
- Contact your support engineer.

Negative Volume Cells

In most cases, negative volume cells during the run occur due to an inappropriate time-step size or due to re-mesh criteria that are not sufficiently sensitive.

- Reduce the Time-Step Size near the moment of valve closure. For example, use a time step table that allows you to define a variable time-step size.
- Apply relevant re-mesh criteria. For example, increase the Min Cell Quality.
- Try to increase or decrease the Normalized Valve Curtain Interface Position in order to accurately reflect the valve contact.

Solution Divergence

Residuals that show a solution divergence indicate a problem in the overall setup or an inappropriate time step size.

- Make sure that the overall setup is correct. Some things to look for are:
  - Reasonable initial and boundary conditions.
  - Correct units in the input tables.
  - Correct pressure specification—STAR-ICE requires Absolute Pressure values for initialization and at flow boundaries.
- Reduce the Time-Step Size.
- Reduce the Under-Relaxation Factors.

STAR-ICE Reference

STAR-ICE provides specific right-click actions and properties for the various objects in the object-tree.

STAR-ICE Right-Click Actions

Object	Right-Click Action
STAR-ICE	**Summarize...** Displays a summary of simulation data in the *Output* window. **Note:** This right-click action is currently not supported—no simulation data are listed. **Collapse Tree** Shrinks the tree back to its default size.

Common Object Right-Click Actions

The following right-click actions are common to many object nodes in the object tree:

Edit...

Allows you to edit the properties of the selected object.

Copy

Copies the object and selected properties onto the clipboard.

Paste

Replaces the properties of the current object by pasting the ones from the clipboard.

Delete

Removes the selected object from the tree. Deleting the resulting part of a Boolean Operation reverses the operation and restores the input parts. When you delete an Engine Part Surface, the corresponding faces are transferred back to the default surface.

Highlight

Highlights an object in the *Graphics* window.

Rename...

Allows you to change the name of the object under which it is stored in the object tree.

Set New Tag...

Creates a new tag and adds it to the selected object. Tags can be used to organize objects in the model. See Tagging Simulation Objects.

Set Tags...

Allows you to set a new or previously created tag to the selected object. See **Set New Tag**.

For more information, see Working with the Object Tree.

Common Object Properties

Tags

Specifies the tags for the selected object. See **Set New Tag**.

Contents:

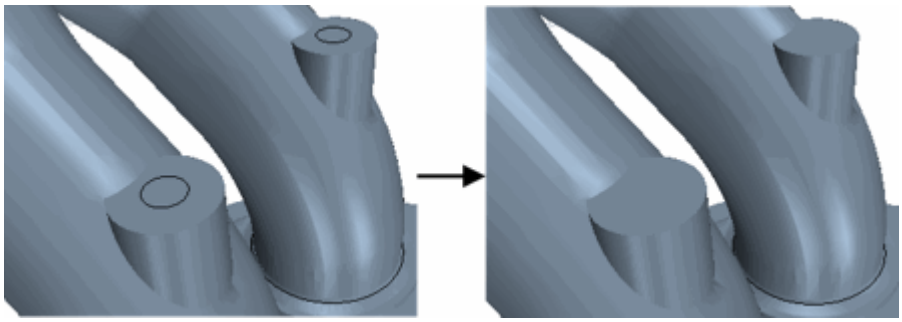
Geometry
Setup
Parameters
Engine
Mesh
Graphics Window—Interactive Right-Click Actions

Geometry

The following right-click actions and dialogs are available for geometries in STAR-ICE.

Geometry Right-Click Actions

Object	Right-Click Action
Geometry	**Import CAD Model** Opens a dialog that allows you to import a CAD geometry. For information on the supported file formats, see Geometry Requirements. The following import options are available: • **Repair Invalid Bodies**: When **On**, STAR-CCM+ attempts to heal errors with the underlying Parasolid geometry and reports on the number of invalid bodies that are successfully repaired. For more information, see Repair Invalid Bodies. • **Remove Redundant Topology**: When **On**, STAR-CCM+ removes redundant edges and vertices. The following image shows a CAD geometry before and after removing redundant topology:

Object	Right-Click Action
	 **New Construction Geometry > [construction geometry]** Creates a new geometry using a graphical editor. The following shape templates are available: • Block • Sphere • Cylinder • Sheet Body Rectangle • Sheet Body Circle For more information, see Shape Templates. **Transfer Cad Bodies** Transfers CAD bodies from the CAD modeler in STAR-CCM+ to STAR-ICE. Launching STAR-ICE automatically creates the 3D-CAD model **STAR-ICE 3D-CAD 1** in STAR-CCM+. This 3D-CAD model contains all Geometry Parts that you import or create in STAR-ICE. By directly entering this model, you can use the capabilities of the 3D-CAD modeler to create advanced geometries. For details on how to build a geometry in 3D-CAD, see Using 3D-CAD. **Show Only Unused Construction Geometry** Hides all geometries from the *Graphics* window that are used to define the engine. **Check Interference** Checks if geometries are intersecting each other. Intersecting geometries are listed in the *Output* window.
∟ **[geometry]**	**Boolean Construction Geometry** Allows you to perform Boolean unite and subtract operations on the selected geometry. The result is a new geometry that is added to the **Geometry** node. See Boolean Operations.

Object	Right-Click Action
	Pattern Construction Geometry Allows you to perform a linear pattern operation on the selected geometry. The result is a new geometry that contains multiple identical geometries that follow a regular formation along straight lines. You can define the pattern by one or by two linear directions extending from the original geometry. The number of bodies and the distance between them determine the extent of the pattern. See Boolean Operations. **Hide** Hides the selected geometry from the *Graphics* window. This action is useful for gaining a clear view of other objects. You can restore hidden objects again by using the **Show** action. **Show** Restores a hidden geometry. **Edit** Opens a dialog that allows you to set geometry properties. See Geometry Dialog. **Position** Launches the Positioning Tool that allows you to interactively place the selected geometry in reference to other geometries.

Geometry Dialog

Source of Geometry	Property
Shape template (Block, Sphere, Cylinder, Sheet Body Rectangle, Sheet Body Circle)	See Shape Templates.
Imported from another CAD package	*Name* Specifies the name of the geometry under which it is stored in the tree. *Geometry* **Local Origin** Specifies the position of the geometry's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. It is used as reference for placing the geometry in the simulation domain. The following options are available:

Source of Geometry	Property
	• **Center of Bounding Box**: Center of the box that bounds the imported geometry. • **Bottom (ZMin) Center of Bounding Box**: Center of the Zmin face of the bounding box. • **Bounding Box: +-X/+-Y/+-Z**: A corner of the bounding box. For example, Bounding Box: +X/-Y/+Z sets the local origin to the corner that corresponds to the maximum X coordinate, the minimum Y coordinate and the maximum Z coordinate. • **Custom**: This value appears when you set the local origin interactively within the *Graphics* window. *Placement* See Placement properties.

Contents:

Shape Templates
Boolean Operations
Placement

Shape Templates

STAR-ICE allows you to create geometries that are based on simple shape templates.

Common Properties

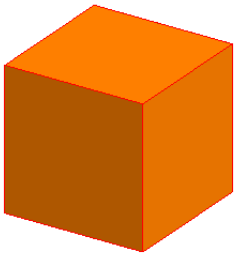
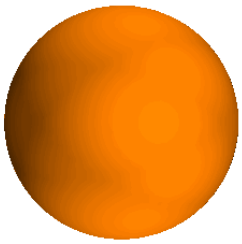
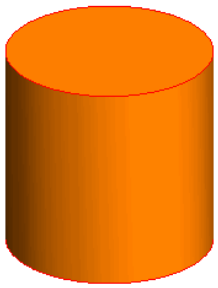
The following properties are common to all types of shape template geometries:

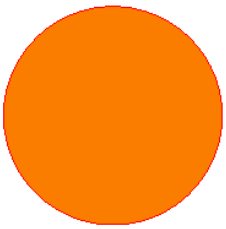
Common Properties	
Name	Specifies the name of the geometry under which it is stored in the tree.
Placement	See Placement properties.

Shape Template Properties

The following **Geometry** properties characterize the different shape templates:

Shape Template	Property
Block	**X, Y, Z** Specifies the dimensions of the block in x-, y-, and z-direction, in the geometry's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window.

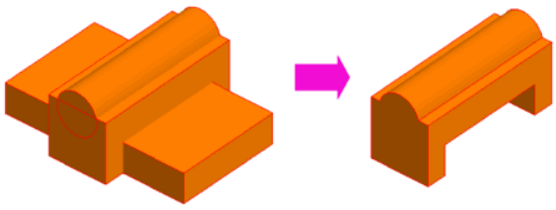
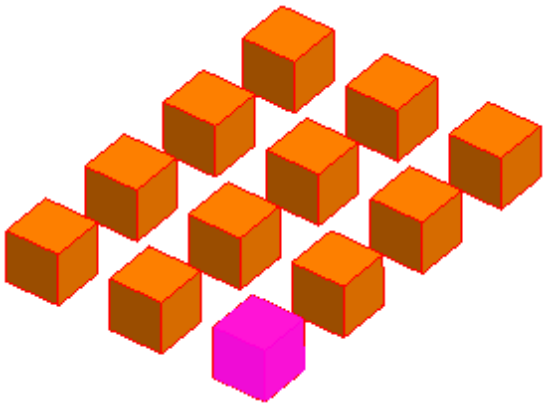
Shape Template	Property
	**Local Origin** Specifies the position of the block's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. It is used as reference for placing the geometry in the simulation domain. The following options are available: • **Bottom Center**: Center of the Zmin face. • **Center**: Center of the block. • **Bottom Corner 1**: Xmin/Ymin/Zmin corner of the block. • **Bottom Corner 2**: Xmin/Ymax/Zmin corner of the block. • **Custom**: This value appears when you set the local origin interactively within the *Graphics* window. **Calculate Size from QuickParts** Allows you to calculate the dimensions of the block based on the maximum and minimum coordinates of particular engine parts that you specify. You can modify the size of the bounding block by entering offsets in negative and positive coordinate directions. This option is particularly useful for defining a volumetric zone for mesh customization around specific engine parts.
Sphere 	**Radius** Specifies the radius of the sphere. **Calculate Size from QuickParts** Allows you to calculate the radius of the sphere as half the maximum distance between particular engine parts that you specify. You can extend the size of the bounding sphere by adding an offset to the radius. This option is particularly useful for defining a volumetric zone for mesh customization around specific engine parts. **Note:** The **Local Origin** of a sphere is the centroid of the sphere, by default.
Cylinder 	**Height** Specifies the length of the cylinder from bottom to top. **Base Radius** Specifies the radius of the cylinder base. **Tapered Cylinder** When activated, the **Top Radius** specifies the radius on top of the cylinder; otherwise the cylinder has a constant radius throughout.

Shape Template	Property
	Local Origin Specifies the position of the block's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. It is used as reference for placing the geometry in the simulation domain. The following options are available: • **Bottom Center**: Center of the Zmin face. • **Center**: Center of the cylinder. • **Custom**: This value appears when you set the local origin interactively within the *Graphics* window.
Sheet Body Rectangle 	**Length** Specifies the length of the rectangle. **Width** Specifies the width of the rectangle. **Local Origin** Specifies the position of the rectangle's local coordinate system. See X_or, Y_or, Z_or in the *Graphics* window. It is used as reference for placing the geometry in the simulation domain. The following options are available: • **Corner 1**: Xmin/Ymin/Zmin corner of the rectangle. • **Center**: Center of the rectangle. • **Custom**: This value appears when you set the local origin interactively within the *Graphics* window.
Sheet Body Circle 	**Radius** Specifies the radius of the circle. **Note:** The **Local Origin** of a circular sheet body is the center of the circular face, by default.

Boolean Operations

STAR-ICE allows you to unite, subtract, and pattern geometries.

The following **Geometry** properties describe the various types of operations:

Operation	Property
Boolean Construction Geometry 	**To Unite** Specifies the geometries that the unite operation is performed on. It unites all input objects that are in contact with or overlap each other. **To Subtract** Specifies the geometries to remove from the target object. The specified geometry is consumed in the operation.
Pattern Construction Geometry 	**Seed Construction Object** Specifies the geometry to be reproduced multiple times using a linear pattern. **Direction (n)** Defines the parameters for one of the two directions in the linear pattern: **Direction(n)** Specifies the direction in which the pattern extends. The **X**, **Y**, and **Z** components of the direction vector are defined with respect to the selected **Coordinate System**. **Distance** Specifies the gap between each object instance. **Number of Instances** Specifies how many times the body is reproduced in each direction. The value must be a whole number. If you enter a decimal value, it is rounded down to a whole number.

Note: The **Local Origin** of the resulting geometry is placed in the origin of the Laboratory coordinate system by default (see X_or, Y_or, Z_or in the *Graphics* window).

Placement

STAR-ICE offers various positioning capabilities for geometries.

The following options are available:

- Specifying the **Placement** properties

  You define the location and orientation of the geometry with respect to a selected coordinate system.

- Using the Positioning Tool

  The Positioning Tool allows you to place geometries in reference to each other.

The following **Placement** properties are common to all types of geometries:

Location

Specifies the position vector (**X**, **Y**, **Z**) of the geometry's **Local Origin**, in the selected **Coordinate System**.

The **Local Origin** is a *Geometry* property of the respective geometry.

Rotation Axis Vector

Specifies the direction vector (**X**, **Y**, **Z**) of the axis about which you rotate the geometry, in the selected **Coordinate System**. The axis position, or the origin of the axis, is the geometry's **Local Origin**.

Angle

Specifies the angle of rotation.

Coordinate System

Specifies the reference coordinate system for the geometry's **Location** and **Rotation Axis Vector**. See X_ref, Y_ref, Z_ref in the *Graphics* window.

Position

Launches the Positioning Tool, which allows you to interactively place the geometry in reference to other geometries.

Contents:

Positioning Tool

Positioning Tool

The **Positioning Tool** allows you to place geometries in reference to each other.

The following options are available:

- Align—turns and moves an object until it is aligned with another object.
- Rotate—turns an object around a specified axis.
- Translate—moves an object from one position to another.

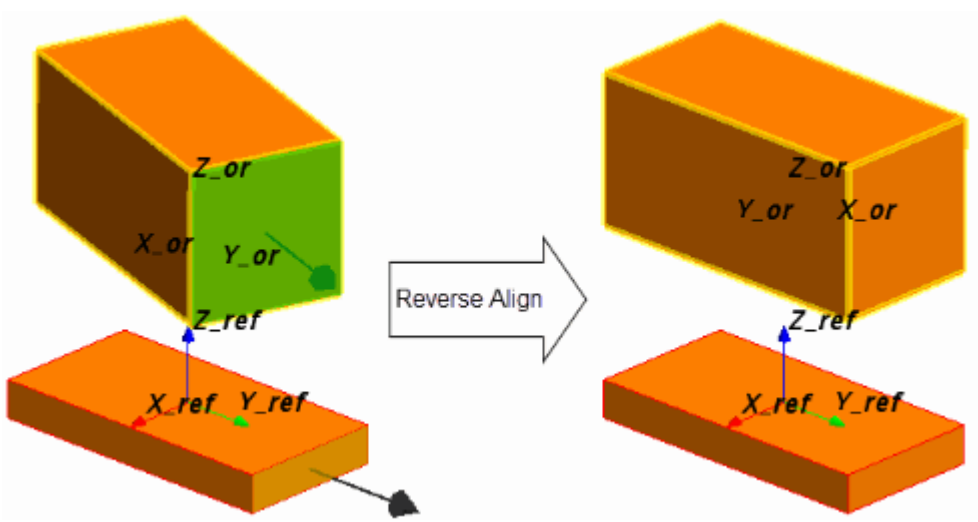
The following properties characterize the different positioning options:

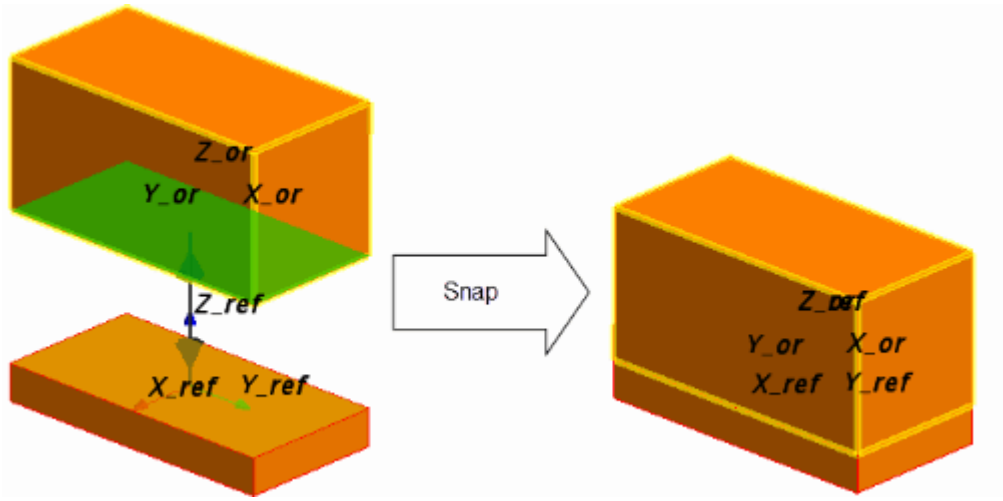
Align—Properties

Target Objects	Specifies the geometry you want to align.
Align By	Controls the alignment mode. The following options are available: • **Face, Edge, Vertex** Aligns the **Target Objects** with a geometric entity—face, edge, or vertex—of another geometry.

	Reference: Specifies a geometric entity of the reference geometry. **Target**: Specifies a geometric entity of the **Target Objects**. You select the reference and target entities interactively in the *Graphics* window (highlighted in ochre and green, respectively). After the selection, the calculated **Angle** between the selected entities is displayed. **Align**: Rotates the **Target Objects** about its local origin (see X_ref, Y_ref, Z_ref in the *Graphics* window) to end up in a position where the selected geometric entities are parallel to each other. The vector normals of the selected entities point in the same direction. **Reverse Align**: As **Align** with the vector normals of the selected entities pointing in opposite directions. For aligned geometric entities, **Snap** translates the **Target Objects** to end up in a position where the selected geometric entities are in contact. **Distance** displays the magnitude of the translation vector. • **X-Axis**, or **Y-Axis**, or **Z-Axis**: Aligns the **Target Objects** with the x-, y-, or z-axis of a **Coordinate System** that you specify. **Set Reference Axis Face**: Specifies a face of the **Target Objects**. **Coordinate System**: Specifies the laboratory coordinate system or a local coordinate system of another geometry. The **Target Objects** are rotated about its local origin (see X_ref, Y_ref, Z_ref in the *Graphics* window) to end up in a position where the respective axis is normal to the selected face.

Example: Positioning of a block on another block.



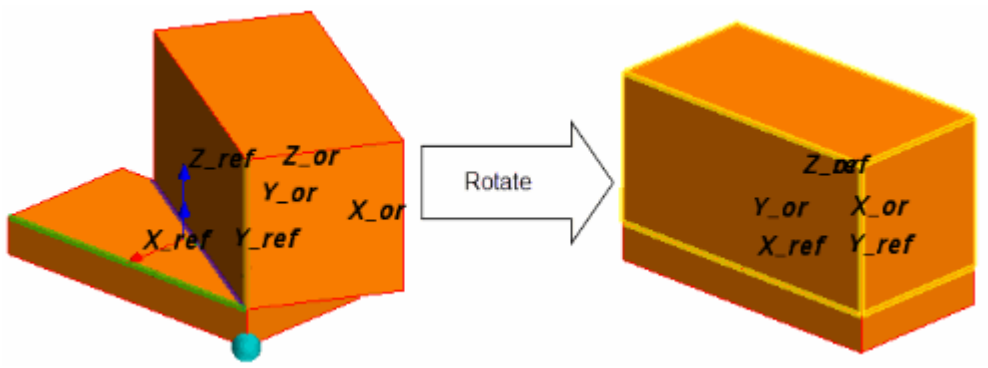


Rotate—Properties

Target Objects	Specifies the geometry you want to rotate.
Rotation Vector	Defines the direction vector of the axis of rotation. You can: • Enter the **X**, **Y**, and **Z** components of the direction vector, in the selected **Coordinate System**. • Determine the direction vector interactively within the *Graphics* window by selecting one of the following entities (highlighted in orange): ◦ `Edge`: sets the rotation vector collinear to the edge. ◦ `Face`: sets the rotation vector equal to the face normal.
Rotation Position	Defines the origin of the axis of rotation. You can: • Enter the **X**, **Y**, and **Z** components of the origin, in the selected **Coordinate System**. • Determine the origin interactively within the *Graphics* window by selecting one of the following entities (highlighted in pale blue): ◦ `Face`: sets the origin to the center of the face. ◦ `Edge`: sets the origin to the midpoint of the edge. ◦ `Vertex`: sets the origin to the vertex location.
Angle	Specifies the angle of rotation. You can: • Enter the angle value directly. • Determine the angle interactively within the *Graphics* window. The rotation angle is calculated **From** one point **To** another, where the point locations depend on the selected entities (highlighted in dark blue and green, respectively): ◦ `Face`: sets the point to the center of the face. ◦ `Edge`: sets the point to the midpoint of the edge. ◦ `Vertex`: sets the point to the vertex location.

Coordinate System	Defines the coordinate system for the rotation. You can define the rotation in the **Laboratory** coordinate system or in the local coordinate systems of another geometry. See X_ref, Y_ref, Z_ref in the *Graphics* window.

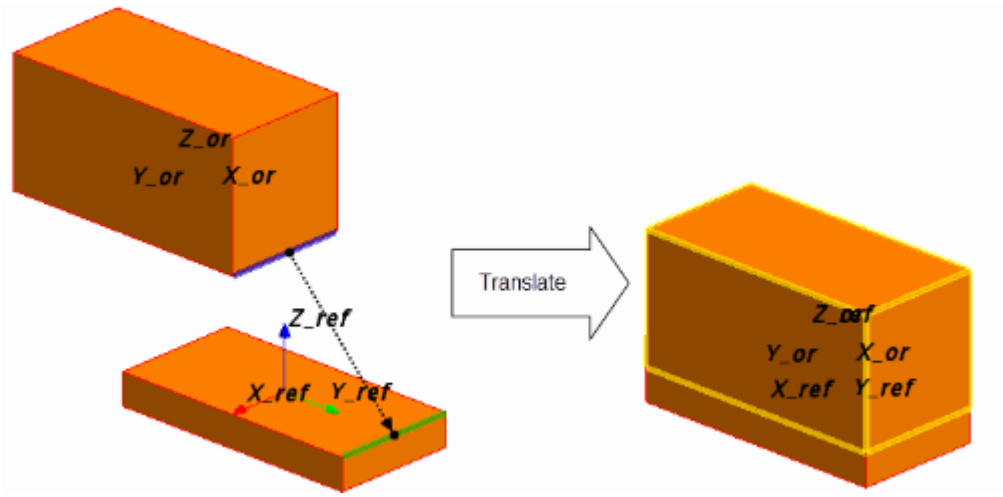
Example: Positioning of a block on another block.



Translate—Properties

Target Objects	Specifies the geometry you want to translate.
Translation Vector	Defines the direction and distance of translation. You can: • Enter the **X**, **Y**, and **Z** components of the translation vector, in the selected **Coordinate System**. • Determine the translation vector interactively within the *Graphics* window. The direction and distance is calculated **From** one point **To** another, where the point locations depend on the selected entities (highlighted in dark blue and green, respectively): ◦ `Face`: sets the point to the center of the face. ◦ `Edge`: sets the point to the midpoint of the edge. ◦ `Vertex`: sets the point to the vertex location.
Coordinate System	Defines the coordinate system for the translation. You can define the translation in the **Laboratory** coordinate system or in the local coordinate systems of another geometry. See X_ref, Y_ref, Z_ref in the *Graphics* window.

Example: Positioning of a block on another block.



Setup

The following right-click actions and dialogs are available for the setup in STAR-ICE.

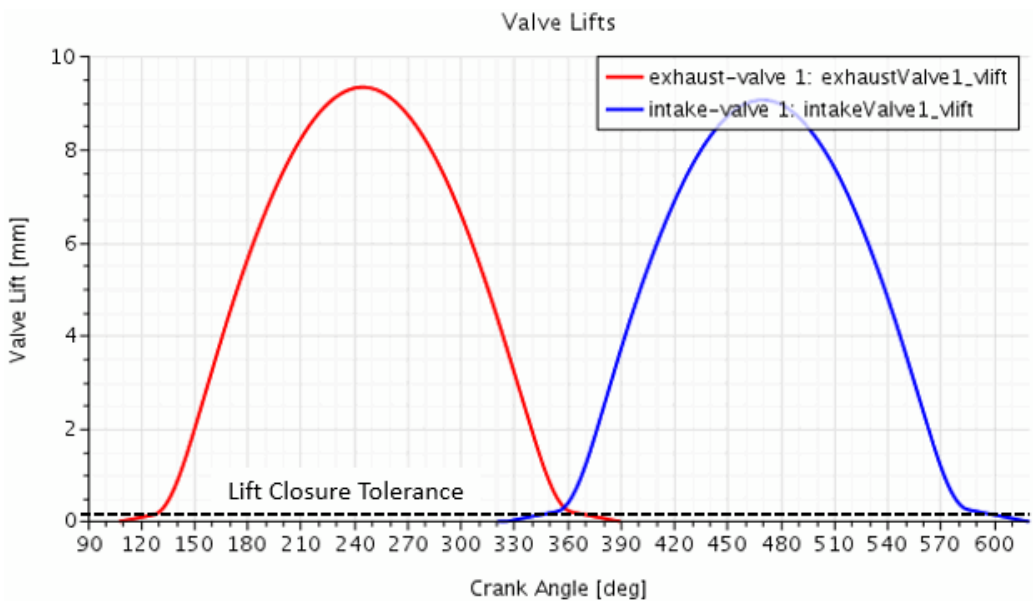
Setup Right-Click Actions

Object	Right-Click Action
Setup	None.
∟ **Solvers**	**Edit** Opens a dialog that allows you to set the properties of the solvers. See Solvers Dialog.

Solvers Dialog

Engine Solver Settings	
Unsteady	Controls the time-step specification and the molecular weight of the gas. **Time Step** Allows you to set a constant time-step size in seconds. **Crank Angle Step** Allows you to set a constant time-step size in crank angle degrees. **Stepped Table** Allows you to set the time-step size in crank angle degrees as a function of crank angle. You Import the tabular values from a `.csv` file, which contains comma-separated or tab-separated values. STAR-ICE interpolates the values using the step method. **Example**: Decrease of time-step size during opening and closing of the valves.

Engine Solver Settings	

	Suppose the graph below shows the valve lift curves for an exhaust valve and an intake valve:
	 
	
	With a **Lift Closure Tolerance** of 0.1 mm, you can consider the exhaust valves opening at 130 CA and closing at 360 CA. The intake valves open at 355 CA and close at 385 CA.
	The following table reduces the time-step size of 0.1 CA to 0.05 CA during opening/closing of the valves and keeps the reduced time-step for 1 CA after opening/closing:
	`CA,TimeStep(CrankAngleStep)`
	`0,0.1`
	`129.9,0.05`
	`131.0,0.1`
	`354.9,0.05`
	`356.0,0.1`
	`359.9,0.05`
	`361.0,0.1`
	`584.9,0.05`
	`586.0,0.1`
	Molecular Weight of Gas
	Specifies the molecular weight of the gas as used by the ideal gas law, see M in Eqn. (1467).

Segregated Flow	Controls the solution update and thus the convergence of the solvers that compute the intermediate velocity field and the update of the pressure field.

Engine Solver Settings	
	Velocity Under-Relaxation Factor, Pressure Under-Relaxation Factor At each iteration, these properties govern the extent to which the newly computed solution supplants the old solution. For the theoretical background, see Eqn. (1625). The default values of 0.8 for velocity and 0.2 for pressure are conservative, that is they lead to a converged solution in most cases. For further guidelines, see Setting Under-Relaxation Factors for Transient Simulations.
Segregated Energy	Controls the solution update of the solver that computes the temperature field in the fluid and solid domains. **Fluid Under-Relaxation Factor, Solid Under-Relaxation Factor** In order to promote convergence, these properties under-relax changes of the solution during the iterative process. See Eqn. (1625). The default values are 0.9 for fluid energy and 0.99 for solid energy. These values achieve a stable solution in most cases.

Auto-Save Settings	
Enabled	When **On**, STAR-ICE saves a separate copy of the simulation file automatically at regular intervals during the run. This option is also useful for recovering from computer system crashes. The following options control how STAR-ICE triggers the auto save operation: **Delta Crank Angle** Saves the simulation every time the crank angle passes the specified delta value. **Delta Time** Saves the simulation every time the simulation time passes the specified delta value. **Number of Time Steps** Saves the simulation every time the time step passes the specified value. By default, when STAR-ICE saves copies of the simulation file, it places an @ character to separate the base file name and the number of the trigger event. STAR-ICE saves these copies in the same directory as the original simulation file.
Number of Files to Keep	When the specified number of auto-saved simulation files is reached, the next auto-save operation deletes the oldest copy of the file. When set to `0`, instructs STAR-ICE to keep overwriting the original simulation file.

Operating Conditions	
RPM	Specifies the engine angular velocity in revolutions per minute.
Start Angle	Specifies the crank angle at which the simulation begins.

Operating Conditions	
	In general, you run an analysis over multiple engine cycles to reduce the effect of initial conditions on the final results. If your processing time is limited, you are advised to start the analysis some crank angles before the intake valve opens—for example at 300 deg crank angle. At this stage, you can assume that the flow in the cylinder is relatively steady compared to the intake and combustion phases.
Duration	Specifies the duration of the simulation in terms of crank angle interval. Therefore, the stopping crank angle is the **Start Angle** plus the **Duration**.
Reference Pressure	Specifies the reference pressure for the gas domain. The reference pressure is simply a device that is used to reduce the numerical roundoff error in the numerical calculations involving pressure. This reduction in numerical error is necessary since the differences in pressure are important, and these differences can be small relative to the absolute value of the pressure (for example, in fully incompressible, or very low-Mach number flows). By subtracting a suitable constant reference pressure, a working pressure is obtained that is less prone to roundoff errors. The reference pressure is used as in Eqn. (1466) .Set the reference pressure to some ambient value such that the working pressures are small. The default value is 1.0 bar.

Valve Motion	
Lift Closure Tolerance	Specifies the valve lift that corresponds to a closed valve. When the valve lift drops to this value or below, STAR-ICE prevents flow past the valve. To prevent flow, STAR-ICE sets the valve curtain interface as an impermeable baffle.
[valve] Offset	Specifies the valve lift of the imported valve geometries.

Parameters

The following right-click actions and dialogs are available for parameters in STAR-ICE.

Parameters Right-Click Actions

Object	Right-Click Action
Parameters	**Edit** Opens a dialog that allows you to define, edit, and delete global parameters. See Parameters Dialog. By default, STAR-ICE provides a parameter for each scalar quantity that you define in the STAR-ICE simulation.

Parameters Dialog

Parameters Properties	
Parameter	Specifies the name of the global parameter, such as `MyValveLash` for a scalar parameter.

Parameters Properties	
Dimensions	Specifies the dimension of the global parameter. The dimension that you set determines the SI unit in the **Value** text-box. For example, **Temperature** gives you the unit K. **Dimensionless** gives you no unit.
Value	Specifies the value of the global parameter.

Parameters Actions	
Add a new parameter	Adds a new global parameter.
(**Delete**)	Deletes a global parameter.

Engine

The following right-click actions and dialogs are available for Engine Parts in STAR-ICE.

Engine Right-Click Actions

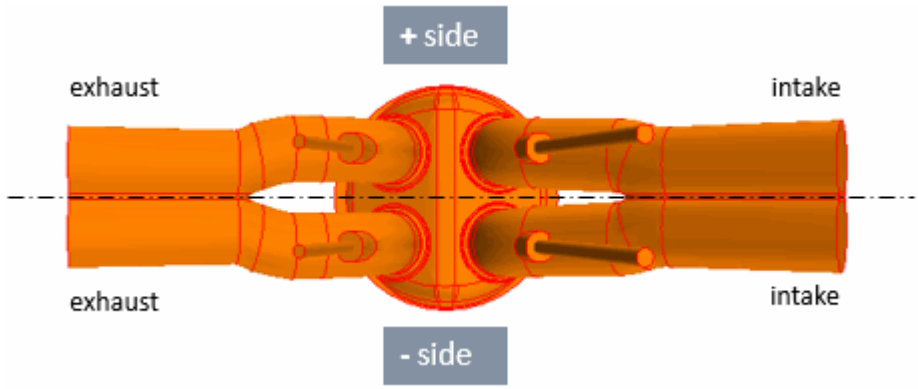
Object	Right-Click Action
Engine	**Create Cylinder and Valves** Opens a dialog that allows you to specify the engine configuration and engine symmetry. See Engine Parts Creation Dialog. Using the settings that you specify in the dialog and the filters described in [ref_link], STAR-ICE assigns the geometry to the respective Engine Parts and the geometry surfaces to the respective Engine Part Surfaces. **Run Solution** Runs the simulation as specified in the Solvers Dialog.
∟ **[engine part]**	**Hide** Hides the selected Engine Part from the *Graphics* window. This action is useful for gaining a clear view of other objects. You can restore hidden objects again by using the **Show** action. **Show** Restores a hidden Engine Part. **Edit** For **Cylinder**, opens the Cylinder Dialog, which allows you to specify the initial conditions and the piston motion of the cylinder. For **Exhaust Valve [n]** / **Intake Valve [n]**, opens the Valve Dialog, which allows you to specify the motion of the valves.

Object	Right-Click Action
	Edit Interactions Allows you to define interactions with other Engine Parts and geometries, see Interactions. **Position** Launches the Positioning Tool that allows you to interactively place the Engine Part in reference to other objects.
└ [engine part surface]	**Hide** Hides the selected Engine Part Surface from the *Graphics* window. This action is useful for gaining a clear view of other objects. You can restore hidden objects again by using the **Show** action. **Show** Restores a hidden Engine Part Surface. **Edit** Opens a dialog that allows you to specify the physical conditions on the boundary. See Engine Part Surfaces. **Dependencies** Displays dependency relationships of the surface with other objects. See Object Dependencies.

Engine Parts Creation Dialog

Engine Configuration	
Intake Valve(s)	Specifies the number of intake and exhaust valves that you import.
Exhaust Valve(s)	

Engine Symmetry	
Half Model	When **On**, creates a half model from a symmetric geometry. **Symmetry Plane** Specifies the symmetry of the imported geometry. • **+ side**: Keeps the + side of the geometry • **- side**: Keeps the - side of the geometry

Engine Symmetry	
	

Cylinder Dialog

Name

Specifies the name of the cylinder under which it is stored in the tree.

Initial Conditions	
Turbulence Specification	Controls how you define the initial turbulence profile for the cylinder and the ports. The following options are available: • **K + Epsilon** • **Intensity + Length Scale** • **Intensity + Viscosita Ratio** The corresponding value nodes become available in the **Cylinder** and **Intake / Exhaust Port [n]** group boxes. For more information, see K-Epsilon Initial Conditions Reference.
Cylinder **Intake Port [n]** **Exhaust Port [n]**	**Pressure / Temperature** Specifies the initial absolute pressure / temperature in form of a **Table**. The table must define the absolute pressure / temperature as a function of crank angle. STAR-ICE evaluates the respective value at the specified Start Angle using linear interpolation between the data points: • **File**: Sets the file that contains the absolute pressure / temperature data. **Import** allows you to import a table of one of the following file formats: ***.csv**, ***.txt**, or ***.dat**. • **Crank Angle Column**: Sets the header of the column that contains the crank angle values. • **Absolute Pressure Column / Temperature Column**: Sets the header of the column that contains the absolute pressure / temperature values.

Initial Conditions	
	• **Crank Angle Units**: Sets the units of the imported crank angle values—**deg** or **radian**. • **Pressure / Temperature Units**: Sets the units of the imported absolute pressure / temperature values. For pressure: **Pa**, **bar**, or **psi**. For temperature: **C**, **F**, **K**, or **R**. If no table headers are available, STAR-ICE numbers the columns starting from zero, such as **column0**. **Note:** When you change the units of a table after having clicked the **Apply** button, STAR-ICE converts the table values to the new units.

Motion	
Engine Geometry	**Bore** Specifies the diameter of the engine cylinder, see B in Eqn. (1421) and Eqn. (1422). STAR-ICE calculates this value from the imported geometry. If the value is not calculated precisely, you can overwrite it with the exact value. **Stroke** Specifies the distance that the piston travels from BDC to TDC, see S in [eqn_link], Eqn. (1421), and Eqn. (1422). **Connecting Rod Length** Specifies the length of the link that connects the piston to the crank shaft, see CRL in [eqn_link]. **Pin Offset** Specifies the pin offset, see PO in [eqn_link].
	Displacement Displays the volume that the cylinder displaces when the piston moves from TDC to BDC. It is calculated as: $$D = \frac{\pi}{4} B^2 S$$ (1421) where: • B is the specified **Bore**. • S is the specified **Stroke**. **Compression Ratio** Displays the ratio of the cylinder volume at BTD to the cylinder volume at TDC.

Motion	
	It is calculated as: $$CR = \frac{\frac{\pi}{4}B^2S + V_{TDC}}{V_{TDC}}$$ (1422) where: • B is the specified **Bore**. • S is the specified **Stroke**. • V_{TDC} is the cylinder volume at TDC with the valves closed. **Note:** Currently, STAR-ICE can calculate the correct compression ratio only if you create the Engine Parts with a Start Angle of 0 deg crank angle (see Setup Reference). **Update Compression Ratio** Recalculates the compression ratio. This action updates the geometry in the background to calculate V_{TDC} in [eqn_link].
Adjust Piston Position	**Piston Position Offset** Specifies the offset of the imported piston geometry from TDC, see PPO in [eqn_link].

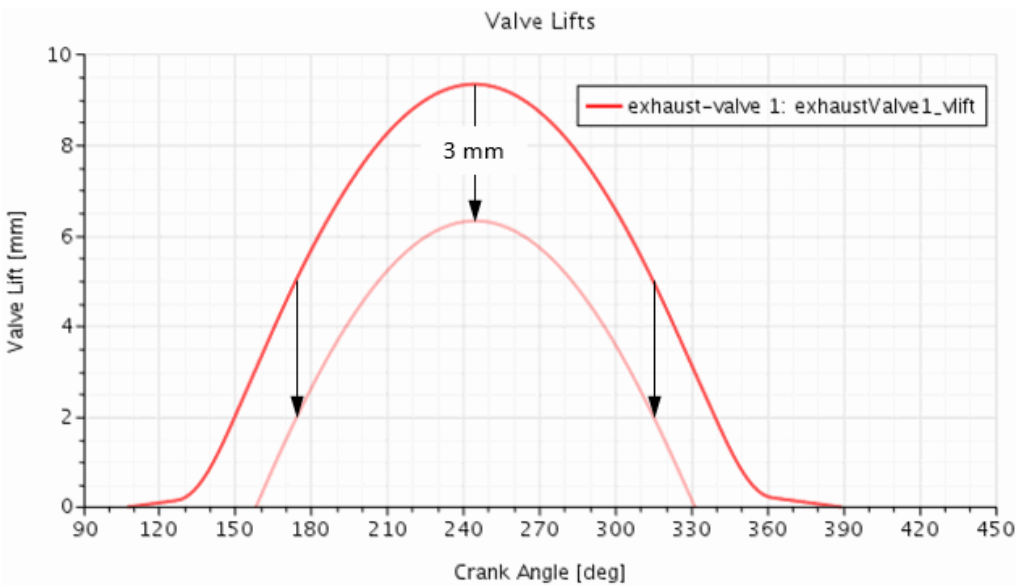
Valve Dialog

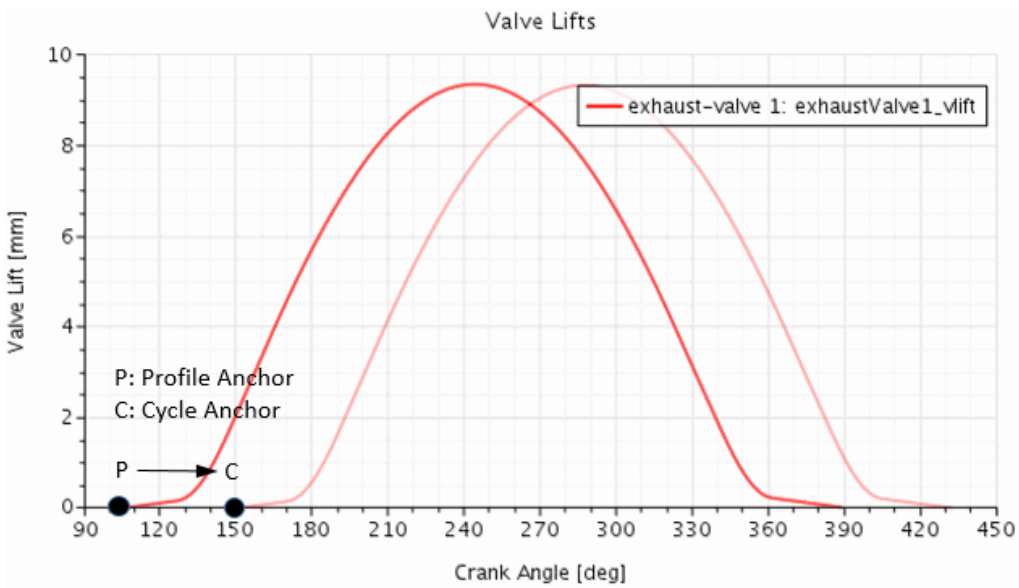
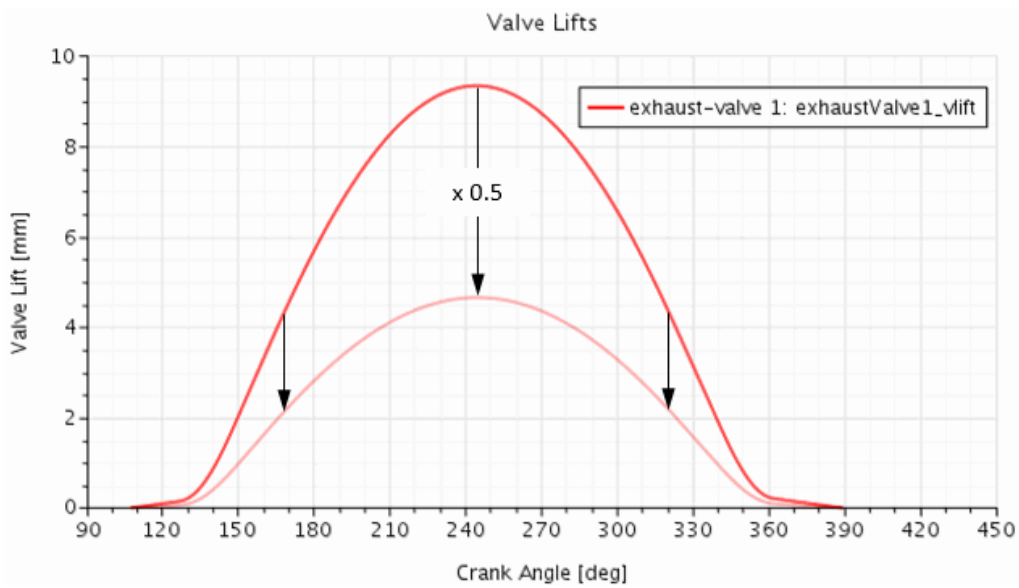
The valve dialog allows you to import tabular data for valve lift curves and modify them according to your needs.

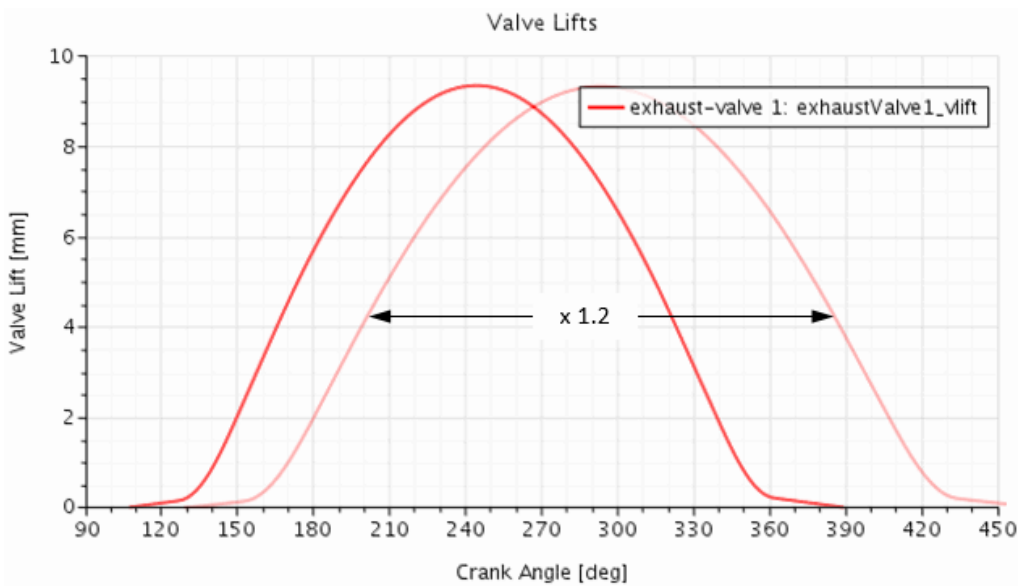
Name

Specifies the name of the valve under which it is stored in the tree.

Motion	
Valve Lift File	Specifies the table that describes the valve lift in mm as a function of crank angle in deg. **Import** allows you to import a table of one of the following file formats: ***.csv**, ***.txt**, or ***.dat**. **Note:** The table must not contain any headers. **Crank Angle Units** Displays the units for the imported crank angle values. Currently, only **deg** is supported.

Motion	
	Lift Units Displays the units for the imported valve lift values. Currently, only **mm** is supported.
Lash	Defines the play resulting from loose connections between the rocker arm and the valve stem. The valve lift curve moves down by the specified value. **Example**: Setting lash to `3.0 mm`. 
Profile Anchor / **Cycle Anchor**	The **Profile Anchor** sets an angular position in the defined lift profile. The **Cycle Anchor** sets the angular position in the engine cycle to which you want to move the profile anchor. **Example**: Setting the profile anchor to `107.0 deg` and the cycle anchor to `150.0 deg`.

Motion	
	
Lift Multiplier	Modifies the actual lift of the valve (y-axis) **Example**: Setting lift multiplier to `0.5`. 
Duration Multiplier	Modifies the duration of the valve (x-axis). **Example**: Setting duration multiplier to `1.2`.

Motion	
	

Contents:

Engine Part Surface
Interactions

Engine Part Surface

STAR-ICE provides several boundary types to cover various physical situations.

Cylinder Part Surface Properties

Name

Specifies the name of the Cylinder Part Surface under which it is stored in the tree.

Boundary

Group-box that allows you to set the physical conditions for the faces of the Cylinder Part Surface that are in contact with the open environment.

The following **Types** are available:

- **Velocity Inlet**
- **Pressure Outlet**
- **Stagnation Inlet**
- **Symmetry Plane**
- **Wall**

The following tables list the surface properties depending on the selected type:

Velocity Inlet	
A velocity inlet boundary represents an inflow boundary that allows you to set a specific velocity magnitude. The flow direction is normal to the boundary face. For more information, see Velocity Inlet.	
Conditions	**Turbulence Specification** Controls how you define the turbulence profile at the flow boundary. The following options are available: • **K + Epsilon** • **Intensity + Length Scale** • **Intensity + Viscosita Ratio** For more information on these options and their corresponding value nodes, see K-Epsilon Boundaries Reference.
Values	**Static Temperature** Specifies the static temperature of the inflowing fluid. The following options are available: • **Constant**: Sets the temperature to the specified value. The default setting **$DefaultAmbientTemperature** equals 500 K. • **Table**: Specifies the temperature in form of a table that defines the temperature as a function of crank angle. STAR-ICE uses linear interpolation between the data points. • **File**: Sets the file that contains the temperature data. **Import** allows you to import a table of one of the following file formats: ***.csv**, ***.txt**, or ***.dat**. • **Crank Angle Column**: Sets the header of the column that contains the crank angle values. • **Temperature Column**: Sets the header of the column that contains the temperature values. • **Crank Angle Units**: Sets the units of the imported crank angle values—**deg** or **radian**. • **Temperature Units**: Sets the units of the imported temperature values— **C**, **F**, **K**, or **R**. If no table headers are available, STAR-ICE numbers the columns starting from zero, such as **column0**. **Note:** When you change the units of a table after having clicked the **Apply** button, STAR-ICE converts the table values to the new units. **Velocity Magnitude** Specifies the constant magnitude of the inflow velocity. The flow direction is normal to the boundary face.

Velocity Inlet	
	Additional values are available depending on the conditions that you specify (see above).

Pressure Outlet	
A pressure outlet boundary represents an outflow boundary that requires the specification of the working pressure across the outlet. For more information, see Pressure Outlet.	
Conditions	**Pressure Outlet Option** Controls the pressure conditions at the outlet. The following options are available: • **None**: Applies the default conditions. • **Pressure Jump**: Imposes a pressure jump on the outlet. The available **Pressure Jump Options** are: ◦ **None**: Does not apply a pressure jump on the boundary. ◦ **Fan**: Imposes a pressure jump that is obtained from a fan performance curve that you specify. A fan performance curve describes the pressure rise across the fan—that is the static pressure measured downstream of the fan minus the total pressure that is measured upstream of the fan—as a function of the volumetric flow rate. The following **Fan Curve Type** options are available: ▪ **Polynomial**: Defines the fan curve as a linear function. The pressure rise across the boundary is calculated as $p(\dot{v}) = -\frac{p_{\max}}{\dot{v}_{\max}}\dot{v} + p_{\max}$ where $p_{\max}$ and $\dot{v}_{\max}$ are the specified **Maximum Pressure** and **Maximum Flow**, respectively. ▪ **Table**: Defines the fan curve as a table of pressure rise versus volumetric flow rate as indicated by the imported **Fan Curve File** (`*.csv`). After the import, select the appropriate **Pressure Units** and **Volume Flow Units**. The pressure rise is calculated by linear interpolation between the data points in the table. Outside the intervals, the nearest limit is used. The specified fan performance curve must correspond to the actual fan operating rotation rate and temperature. If your data correspond to some standard rotation rate and temperature, scale the fan performance curve as required using the following established fan laws: $\Delta P_2 = \Delta P_1 \times (\omega_2/\omega_1)^2 \times (T_1/T_2)$ $\psi_2 = \psi_1(\omega_2/\omega_1)$

Pressure Outlet	
	where ω is the fan rotation rate (rpm), T is the fluid temperature (K), and ψ is the volumetric flow rate. **Turbulence Specification** As for Velocity Inlet.
Values	**Static Temperature** As for Velocity Inlet. **Pressure** Specifies the absolute pressure at the flow boundary in form of a **Table**. The table must define the pressure as a function of crank angle. STAR-ICE uses linear interpolation between the data points. • **File**: Sets the file that contains the pressure data. **Import** allows you to import a table of one of the following file formats: ***.csv**, ***.txt**, or ***.dat**. • **Crank Angle Column**: Sets the header of the column that contains the crank angle values. • **Pressure Column**: Sets the header of the column that contains the pressure values. • **Crank Angle Units**: Sets the units of the imported crank angle values— **deg** or **radian**. • **Pressure Units**: Sets the units of the imported pressure values— **Pa**, **bar**, or **psi**. If no table headers are available, STAR-ICE numbers the columns starting from zero, such as **column0**. **Note:** When you change the units of a table after having clicked the **Apply** button, STAR-ICE converts the table values to the new units. Additional values are available depending on the conditions that you specify (see above).

Stagnation Inlet	
A stagnation inlet boundary represents an inflow boundary that enables you to set the conditions of an imaginary plenum, far upstream, in which the flow is completely at rest. For more information, see Stagnation Inlet.	
Conditions	**Stagnation Inlet Option** Controls the pressure conditions at the inlet. The following options are available:

Stagnation Inlet	
	• **None**: Applies the default conditions. • **Pressure Jump**: Imposes a pressure jump on the inlet. The available **Pressure Jump Options** are: ◦ **None**: As for **Pressure Outlet Option** at Pressure Outlet. ◦ **Fan**: As for **Pressure Outlet Option** at Pressure Outlet. ◦ **Loss Coefficient**: Computes the pressure loss as $0.5K\rho\|\mathrm{v}\|^2$ where K is the specified **Pressure Loss Coefficient** value. **Turbulence Specification** As for Velocity Inlet.
Values	**Total Pressure** Specifies the total pressure upstream of the simulation domain in the form of a **Table**. For the table properties, see **Pressure** at Pressure Outlet. **Total Temperature** Specifies the total temperature of the upstream plenum. For the available options, see **Static Temperature** at Velocity Inlet. Additional values are available depending on the conditions that you specify (see above).

Symmetry Plane	
A symmetry plane boundary represents an imaginary plane of symmetry in the simulation. The solution that is obtained with a symmetry plane boundary is identical to the solution that would be obtained by mirroring the mesh about the symmetry plane (in half the resulting domain). For more information, see Symmetry Plane.	
Conditions	None
Values	None

Wall	
A wall boundary represents a flow impermeable boundary. For more information, see Wall.	
Conditions	**Thermal Specification** Controls the thermal conditions at the wall. The following options are available: • **Temperature**: Sets the boundary temperature to the specified **Static Temperature** value. The default setting **$DefaultAmbientTemperature** equals 500 K.

Wall	
	• **Adiabatic**: Does not permit heat transfer across the boundary. **Wall Surface Specification** Controls the finish of the surface. The following options are available: • **Smooth**: Regular wall treatment for an even wall. • **Rough**: Modifies the wall treatment to incorporate roughness as indicated by the specified **Roughness Height**. You obtain the value of the equivalent sand-grain roughness height that is appropriate for your model either from the literature or empirically. To determine from an experiment: • Measure the friction factor for the Reynolds number of your case in the experiment. • Plot your measurements on a Moody diagram. • Deduce the equivalent sand-grain roughness height from the Moody diagram curve. For more information, see Wall Treatment for Rough Walls.
Values	The available values depend on the conditions that you specify (see above).

Interface Solid\|\|Gas (for Default : Gases only)

Group-box that contains the physical conditions for the faces of the **Default : Gases** surface that are in contact with a Valve Part Surface.

Note: DO NOT use this group-box. It will be removed in future releases. Instead, use the **Interface Solid||Gas** group box of the Valve Part Surfaces.

Valve Part Surface Properties

Name

Specifies the name of the Valve Part Surface under which it is stored in the tree.

Boundary

Group-box that allows you to set the physical conditions for the faces of the Valve Part Surface that are in contact with the open environment.

For Valve Surfaces, only **Wall** boundaries are supported.

The following table lists the surface properties:

Wall	
A wall boundary represents a flow impermeable boundary. For more information, see Wall.	
Conditions	None.
Values	None.

Interface Solid||Gas

Group-box that contains the physical conditions for the faces of the Valve Part Surface that are in contact with a Cylinder Part Surface.

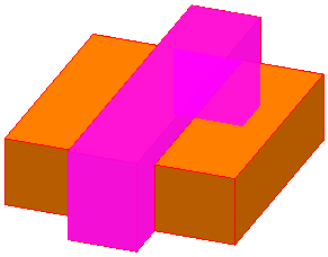
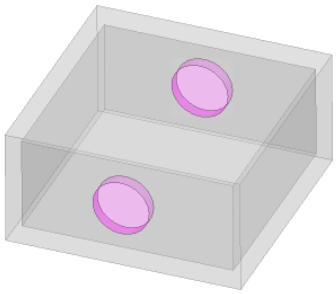
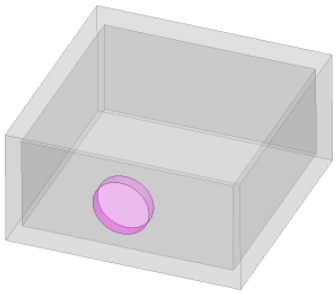
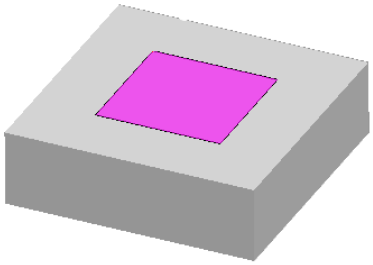
Contact Interface	
Conditions	**Thermal Specification** Controls the thermal conditions at the wall. The following options are available: • **Temperature**: Sets the boundary temperature to the specified **Static Temperature** value. • **Adiabatic**: Does not permit heat transfer across the boundary. **Wall Surface Specification** Controls the finish of the surface. The following options are available: • **Smooth**: Regular wall treatment for an even wall. • **Rough**: Modifies the wall treatment to incorporate roughness as indicated by the specified **Roughness Height**. You obtain the value of the equivalent sand-grain roughness height that is appropriate for your model either from the literature or empirically. To determine from an experiment: • Measure the friction factor for the Reynolds number of your case in the experiment. • Plot your measurements on a Moody diagram. • Deduce the equivalent sand-grain roughness height from the Moody diagram curve. For more information, see Wall Treatment for Rough Walls.
Values	The available values depend on the conditions that you specify (see above).

Interactions

Within STAR-ICE, an interaction allows you to remove volume from an Engine Part, to cut holes through Engine Parts, or to split an Engine Part Surface.

The following properties define the behaviour of an interaction:

Interaction		Example
Subtract	Allows you to remove the volume of one or more objects from the selected Engine Part. The subtracted objects are retained. Each object must share some of its volume with the Engine Part, otherwise the Engine Part remains unchanged.	

Interaction		Example
	Solid Quick Objects Specifies the objects that remove material from the Engine Part. You can choose from Engine Parts and geometries except for sheet geometries (such as Sheet Body Rectangle).	
Extrude Cut Through All	Allows you to project one or more sheet geometries in the normal direction so that they cut holes through the selected Engine Part. The three-dimensional objects that are formed by projecting the sheet geometries are subtracted from the Engine Part. **Sheet Construction Geometry** Specifies the geometries that determine the shape and position of the holes. You can choose from any sheet geometry or pattern of sheet geometries.	
Extrude Cut Through First Segment	Allows you to project one or more sheet geometries in the normal direction so that they cut holes through the selected Engine Part. The three-dimensional objects that are formed by projecting the sheet geometries are subtracted from the first segment of the Engine Part. **Sheet Construction Geometry** Specifies the geometries that determine the shape and position of the holes. You can choose from any sheet geometry or pattern of sheet geometries.	
Imprint	Allows you to imprint the profile of one or more sheet geometries onto the surface of the selected Engine Part. The surface is split where the new lines are added. The Engine Part and the sheet geometries must overlap by less than `1.0E-8 m`, or have coincident faces, as otherwise the Engine Part remains unchanged. This property allows you to create new Engine Part Surfaces on which you can specify custom mesh settings or physics conditions.	

Interaction		Example
	Sheet Construction Geometry Specifies the objects that imprint their profile. You can choose from any sheet geometry or pattern of sheet geometries.	

Mesh

The following right-click actions and dialogs are available for meshes in STAR-ICE.

Mesh Right-Click Actions

Object	Right-Click Action
Mesh	**Edit Mesh Settings** Opens a dialog that allows you to specify the global characteristics of the mesh. See Mesh Settings Dialog. **Generate Surface Mesh** Re-meshes the initial surface to provide a quality discretized mesh that is suitable for CFD. **Generate Volume Mesh** Generates a volume mesh by cutting a hexahedral template mesh.
└ Custom Controls	**New > [custom control]** Creates a new custom control. The following types are available: • Surface Control • Volume Control A separate *Edit* tab opens and allows you to set the mesh settings of the respective custom control. Custom Mesh Settings Dialog.
└ [custom control]	**Highlight** Highlights the respective surfaces or parts of the selected custom control in the *Graphics* window. **Edit Control** Allows you to edit the custom mesh settings.

Mesh Settings Dialog

Mesh Properties	
Core Cylinder Mesh Size	Specifies the target cell size in the core of the cylinder. The minimum cell size and the fluid boundary layer thickness can be set as a percentage of this value.
Minimum Size	Specifies the lower limit of cell size. Decreasing this value allows more refinement in areas where the mesher is trying to capture complex features.
Normalized Valve Curtain Interface Position	Specifies the location of the valve curtain as a percentage of the valve's Face radial distance. The valve curtain separates the cylinder volume from the port volumes. This property supports values between 25 and 75. Only modify this value if you encounter one of the following problems: • CAD errors, such as invalid CAD • Mesh errors, such as negative volume cells • Bad convergence
Fluid Boundary Layer	Controls the prism layer mesh that is intended to capture the fluid boundary layer across the surface. **Total Thickness** Specifies the total overall thickness of the prism layer. **Number of Layers** Specifies the number of cell layers that are generated within the prism layer. **Stretching** Specifies the target growth rate of successive prism layers away from the wall.
Re-mesh Criteria	Controls the criteria for re-meshing the engine. **Min Cell Quality** Specifies the minimum Cell Quality. STAR-ICE re-meshes the engine if any cell in the mesh drops below this value. **Max Piston Stretching** Piston stretching is defined as the ratio of the current piston position to the piston position after the last re-mesh. The piston position is defined as the minimum z-position of the Liner surface in the Laboratory coordinate system. The Laboratory coordinate system is placed on the firedeck with its z-axis pointing up. STAR-ICE re-meshes the engine if the piston stretching exceeds the specified value in either direction (up or down). This criterium becomes active as soon as the piston passes the specified **Piston Tolerance**.

Mesh Properties	
	Max Valve Stretching Valve stretching is defined as the ratio of the current valve lift to the valve lift after the last re-mesh. STAR-ICE re-meshes the engine if the valve stretching exceeds the specified value in either direction. **Max Valve Displacement** Valve displacement is defined as the distance between the current valve lift and the valve lift after the last re-mesh. STAR-ICE re-meshes the engine if one of the valves passes a distance that exceeds the specified value in either direction. **Piston Tolerance** Specifies the distance that the piston is allowed to move away from TDC position without considering the **Max Piston Stretching** criterion. This property allows you to ignore the **Max Piston Stretching** criterion for small position changes away from TDC. **Note:** The value is negative because the piston moves in negative z-direction. **Min Cell Quality Reference** Displays the minimum cell quality after the last re-mesh. This value is calculated automatically. **Note:** DO NOT modify this value.

Custom Mesh Settings Dialog

Custom Mesh Properties		Surface Control	Volume Control
Name	Specifies the name of the custom control under which it is stored in the tree.	✔	✔
Surfaces	Specifies the Engine Part Surfaces to which the custom properties apply.	✔	
Parts	Specifies the Construction Geometries in which the custom properties apply.		✔
Target Size	Specifies the custom target cell size.	✔	✔
Minimum Size	As in Mesh Settings Dialog.	✔	
Fluid Boundary Layer	As in Mesh Settings Dialog.	✔	✔

Property Options

The following options control the specification of various properties within the global and customized settings:

Percentage of Base	Specifies the reference value as percentage of the base size that is defined within the **Edit Mesh Settings** dialog. The advantage of a relative size specification is that changing the base size scales both global and object specific sizes.
Absolute Value	Specifies the reference value as absolute in any of the allowable length units.
Use Parent Value	Applies the global value to the mesh setting as defined in the **Edit Mesh Settings** dialog.
Specify Custom Value	Applies a customized value to the mesh setting.
Disable	Does not apply prismatic cell layers to the surface.

Graphics Window—Interactive Right-Click Actions

STAR-ICE offers various interactive dimensioning and positioning capabilities for geometries and Engine Parts.

The following right-click actions are available for the different geometric entities that are displayed in the *Graphics* window:

Selected Geometric Entity		Common Right-Click Actions
None		**Restore Hidden Quick Objects** Restores all previously hidden geometries and/or Engine Parts. **Restore Hidden Quick Cad Faces** Restores all previously hidden faces of geometries and/or Engine Parts. **Hardcopy...** Allows you to save the visualization display to an image file. For more information, see Hardcopy Properties.
Face	Planar	**Create Cylinder and Valves** Opens a dialog that allows you to specify the engine configuration and engine symmetry. See Engine Parts Creation Dialog. Using the settings that you specify in the dialog and the filters described in [ref_link], STAR-ICE assigns the geometry to the respective Engine Parts and the geometry surfaces to the respective Engine Part Surfaces.

Selected Geometric Entity		Common Right-Click Actions
		Create Face Coordinate System Creates a new local coordinate system, where the origin of the coordinate system is centered on the selected face. Local coordinate systems are useful to place geometries or Engine Parts in reference to each other. **Hide** Hides the selected face from the visualization display. This action is useful for gaining a clear view of other objects. You can restore hidden objects by right-clicking in the empty background of the visualization display and selecting **Restore Hidden Quick CAD Faces**.
		New Construction Geometry > [Type] Creates a new templated geometry of the respective type on the selected face. **Note:** The position of the mouse pointer when selecting the face determines the position of the local origin of the new geometry.
	`Curved`	**Hide** As for `Planar Face`.
`Any`		**Rename** Allows you to change the name of the respective geometry or Engine Part under which it is stored in the tree. **Edit** Allows you to change the geometrical and physical properties of the respective geometry or Engine Part. **Hide** Hides the respective geometry or Engine Part from the visualization display. This action is useful for gaining a clear view of other objects. You can restore a specific hidden object by right-clicking on the respective entry in the tree and selecting **Show**. To restore all hidden objects click in the empty background of the visualization display and select **Restore Hidden Quick Objects**. **Position...** Launches the Positioning Tool that allows you to place the respective geometry or Engine Part in reference to other objects.

Selected Geometric Entity	Specific Right-Click Actions for Engine Parts
Face	**Edit** Allows you to change the properties of the respective Engine Part Surface. **Create QuickPart Surface** Creates a new Engine Part Surface that corresponds to the selected face. You can select several faces at once by holding down the <Ctrl> key when selecting the faces. The new Engine Part Surface is stored in the tree under the specified **Name**. This option is only available for unnamed faces.

Contents:

Interactive Right-Click Actions in Editing Mode

Interactive Right-Click Actions in Editing Mode

Additional right-click actions are available while editing a geometry or an Engine Part. They allow you to place and dimension the object being edited interactively and in reference to other objects.

The following actions are available for the various geometric entities in editing mode:

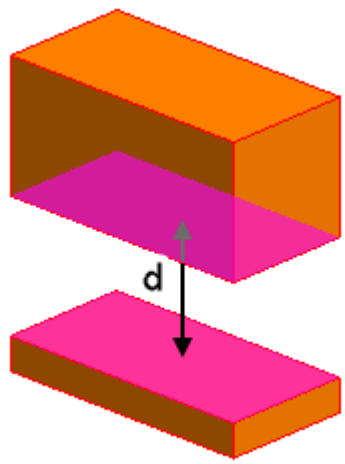
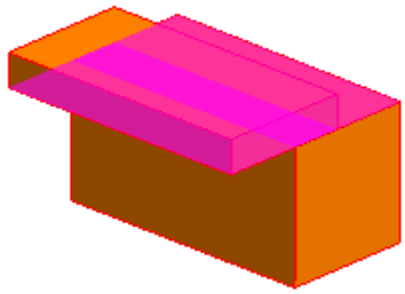
Selected Geometric Entity		Right-Click Actions
None		**Restore Hidden Quick Objects** Restores all previously hidden geometries and/or Engine Parts. **Restore Hidden Quick Cad Faces** Restores all previously hidden faces of geometries and/or Engine Parts. **Hardcopy** Allows you to save the *Graphics* window to an image file. For more information, see Hardcopy Properties.
Vertex		**Set Origin Reference At Vertex** Sets the local origin of the object being edited to the location of the selected vertex. The value of the object's **Local Origin** is set to **Custom**. The components of the **Location Vector** are updated to reflect the vertex position.
Edge	Linear	**Set Origin Reference To Center Of Linear Edge** Sets the local origin of the object being edited to the center of the selected linear edge. The value of the object's **Local Origin** is set to **Custom**. The components of the **Location Vector** are updated to reflect the midpoint position.

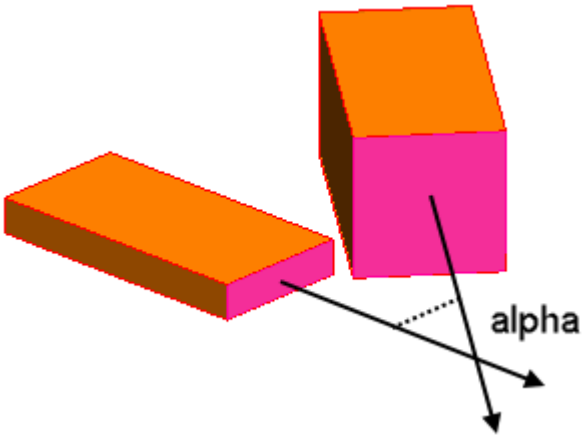
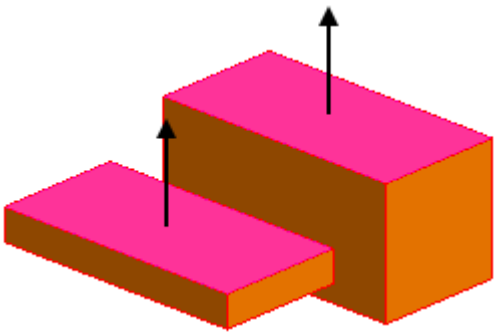
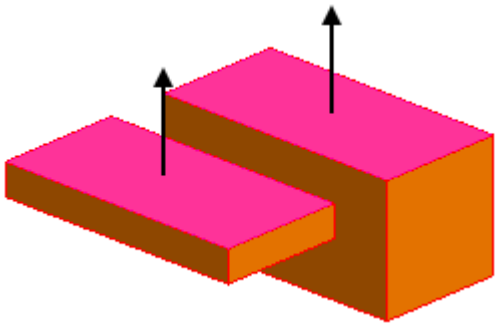
Selected Geometric Entity		Right-Click Actions
		Set Dimension > [Length Dimension] Allows you to set any length dimension of the object being edited to the length of the selected edge.
	`Curved`	**Set Origin Reference To Center Of Curve** Sets the local origin of the object that owns the circular curve to the center of the circle. The value of the object's **Local Origin** is set to **Custom**. The components of the **Location Vector** are updated to reflect the position.
`Face`	`Planar`	**Set Origin Reference To Center Of Face** Sets the local origin of the object being edited to the center of the planar face. The value of the object's **Local Origin** is set to **Custom**. The components of the **Location Vector** are updated to reflect the position. **Hide** Hides the selected face from the *Graphics* window. This action is useful for gaining a clear view of other objects. You can restore hidden objects again by right-clicking in the empty background of the scene and selecting **Restore Hidden Quick CAD Faces**. **Move Object Being Edited to Center of Face** Translates the object being edited. The translation vector is calculated as the vector pointing from the objects local origin to the center of the selected planar face.
	`Curved`	**Hide** As for `Planar Face`.
Any two of the following entities: • `Vertex` • `Linear Edge` • `Planar Face` **Note:** To select two geometric entities hold down the <Ctrl> key when selecting the entities.		**Hide** As for `Planar Face`. **Note:** This action is only available if one of the selected entities is a face. **Position > [Option]** **Note:** This action is only available if one of the selected entities belongs to the object being edited. **Set Dimension > [Length Dimension]** Allows you to set any length dimension of the object being edited to the distance between the selected entities:

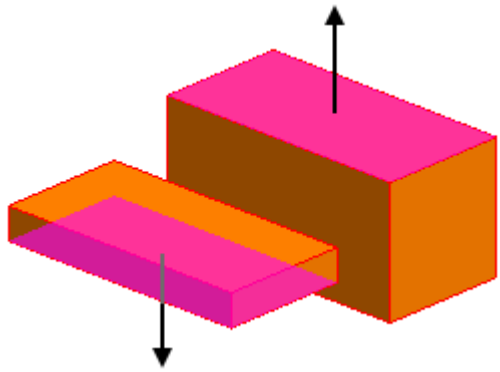
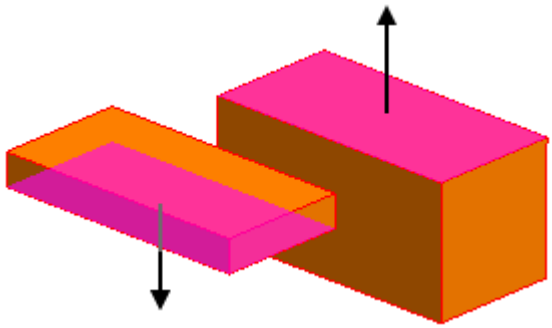
Selected Geometric Entity	Right-Click Actions
	• `Two Vertices`: distance between the vertices • `Any other two entities`: perpendicular distance between the entities **Note:** This action is only available for a selection of faces and/or edges if the faces and/or edges are parallel to each other.

Position > [Option]

Allows you to position the object being edited in relation to another object. The following options are available:

Options	Example
Set Distance Translates the object being edited to a position where the selected geometric entities are at a specified **Distance** (d) from each other.	
Snap Translates the object being edited to a position where the selected geometric entities are in contact.	
Set Angle Rotates the object being edited about its local origin to a position where the selected geometric entities form a specified **Angle** (alpha).	

Options	Example
	
Align Rotates the object being edited about its local origin to a position where the selected geometric entities are parallel to each other. The vector normals of the selected entities point in the same direction.	
Align and Snap Combines **Align** and **Snap** in one action	
AntiAlign Rotates the object being edited about its local origin to a position where the selected geometric entities are parallel to each other. The vector normals of the selected entities point in opposite directions.	

Options	Example
	
AntiAlign and Snap Combines **AntiAlign** and **Snap** in one action.	

Note: For some combinations of `Vertex`, `Linear Edge` and `Planar Face` only a subset of options is available.

Casting Simulation with STAR-Cast

The Casting Module, also referred to as STAR-Cast, is a casting computational software tool. STAR-Cast was jointly developed by CD-adapco (now part of Siemens PLM Software) and ACCESS. ACCESS is an independent research center that is associated with the Technical University of Aachen (RWTH) in Germany.

Casting simulations with STAR-Cast are performed as multiphase mould filling flows with solidification. STAR-Cast can be used as a design tool to assess the filling, solidification and cool-down process. It allows for defect prediction such as topological macro porosity prediction, micro porosity prediction, shrinkage due to phase change, trapped gas during filling as well as cold shots or misruns. Casting simulation enables the design of gating and runner systems before start of production.

Contents:

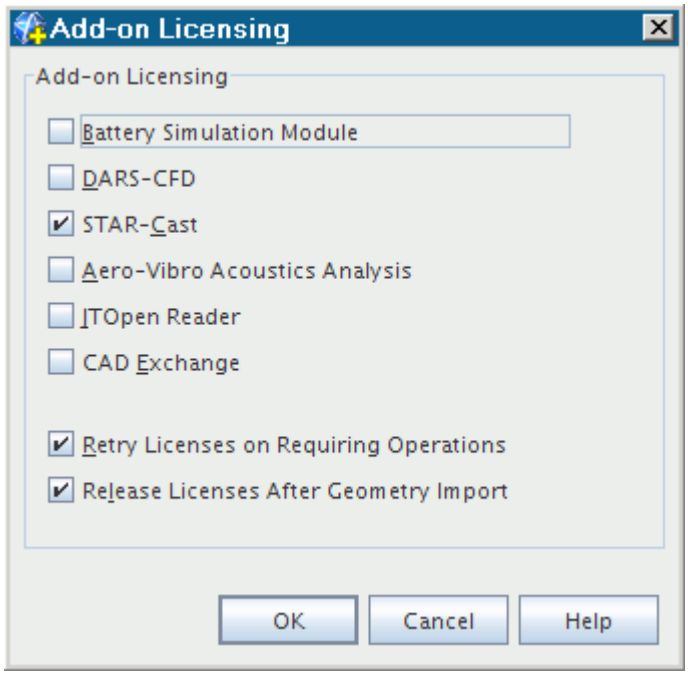
About the Casting Module
Casting Workflow
Setting Up a Casting Simulation
Selecting the Physics Models for the Mold
Selecting the Pure Thermal Model
Selecting the Variable Density Compensation Model
Working With the Casting Material Database
Using Imported Casting Materials and Properties in STAR-CCM+
Using the Macro Porosity Model to Detect Shrinkage-Related Defects
Defect Prediction Using Criteria Functions
Automated Time Step Control
Artificial Compressibility
The Casting User Interface
Casting Field Functions
Casting Bibliography

About the Casting Module

In order to use the Casting Module in STAR-CCM+, you require a STAR-Cast license.

To check whether the Casting Module is available in STAR-CCM+:

1. In the STAR-CCM+ menu bar, select **Tools** > **Add-ons**.
2. In the *Add-on Licensing* dialog, verify that STAR-Cast is activated.



3. Click **OK** to close the *Add-on Licensing* dialog.

Casting Workflow

A casting simulation is a transient multi-phase simulation using the VOF model. You perform a conjugate heat transfer simulation between the solidifying melt and the solid mold. Several steps are required to set up such a simulation.

There are two options to set up a casting simulation:

- You can use the Casting User Interface (CUI) to guide you step-by-step through the setup.

  See The Casting User Interface
- You can set up the casting simulation using the STAR-CCM+ simulation tree.

The process for setting up a casting simulation in the STAR-CCM+ simulation tree is outlined below:

1. Create a multi-region volume mesh with one region for the melt and one region for the mold.
   - If you set up an investment casting simulation, use the Offset Mesher to create shell molds.

     See STAR-Cast: Investment Casting.
2. Select the physics models for the melt .
   - If you run a pure thermal simulation, select the pure thermal model.
3. Select the physics models for the mold.
   - If you run a pure thermal simulation, select the variable density compensation model.
4. Use the Casting Material Database Editor to select the casting materials that you want to use in your simulation, and export the selected collection of materials to STAR-CCM+.

   See Working With the Casting Material Database.
5. Use the imported casting materials in physics continua and assign the appropriate material properties as boundary values.

   See Using Imported Casting Materials and Properties in STAR-CCM+.
6. For the physics continuum of the melt, define the Eulerian phases for the liquid melt and gas.

   The liquid phase must be a single-component liquid and have a temperature-dependent density, such as **User Defined EOS** or **Polynomial Density**.

   See Defining Eulerian Phases.
7. For the Eulerian melt phase, select appropriate melting and solidification models. For pure thermal simulations, do not select the mushy zone or the slurry viscosity models.

   See Modeling Melting and Solidification
8. If you perform a sand casting simulation, select the wall porosity model for the gas phase. Wall porosity accounts for degassing that occurs when melt enters the mold and air is pushed into the sand mold.

   See Working with Wall Porosity.
9. Use the macro porosity model to detect shrinkage-related defects in a cast part.

   See Using the Macro Porosity Model to Detect Shrinkage-Related Defects.
   - If you run a pure thermal simulation, select the **Macro Porosity (pure thermal)** model.
10. If necessary, use the available criteria functions to post-process the simulation results.

See Defect Prediction Using Criteria Functions.

11. To apply automated time-step control, select the segregated time-step model.

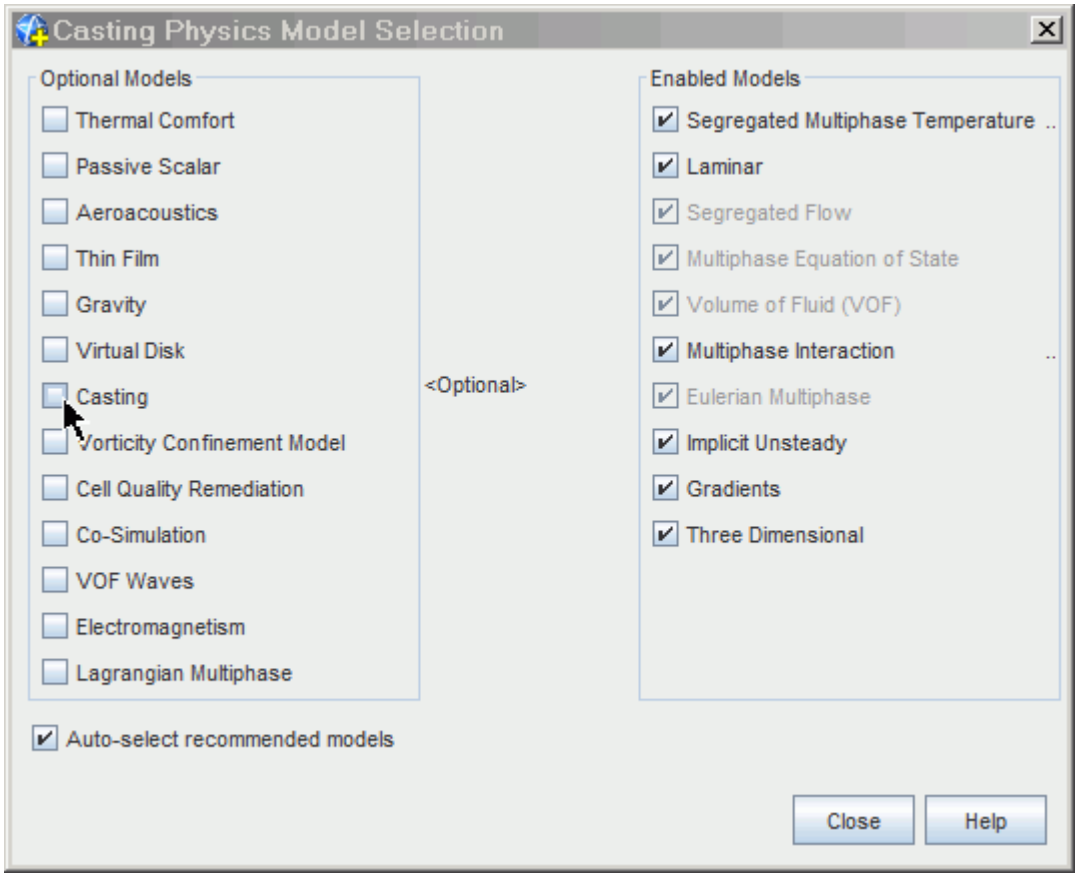
    See Automated Time-Step Control

Selecting the Physics Models for the Melt

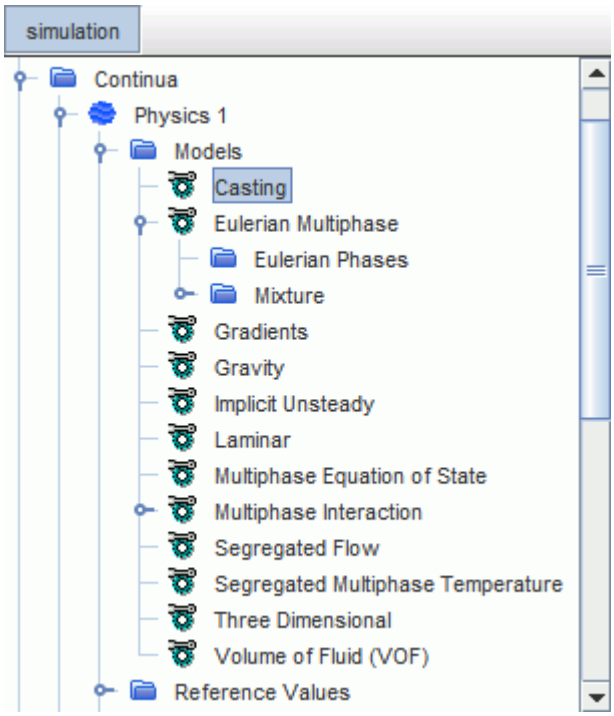
You select the multiphase models, a laminar flow regime, and the casting model.

1. Start STAR-CCM+ and create the appropriate physics continuum.
2. In the *Physics Model Selection* dialog, select the following models:

Group Box	Model
Time	**Implicit Unsteady**
Material	**Eulerian Multiphase**
	Multiphase Interaction (Selected automatically)
Eulerian Multiphase Model	**Volume of Fluid (VOF)**
	Gradients (Selected automatically)
	Multiphase Equation of State (Selected automatically)
	Segregated Flow (Selected automatically)
Viscous Regime	**Laminar**
Optional Models	**Segregated Multiphase Temperature**
	Casting



The **Casting** model appears under the **Models** node in the simulation tree.



The **Casting** node also appears under the **Tools** node in the simulation tree. This node contains the **Material Database Editor** node, which lets you access the Casting Material Database Editor.

See Accessing the Casting Material Database Editor.

Selecting the Physics Models for the Mold

The mold is a solid material. To investigate the heat transfer in the mold, you activate the solution of the energy equation in the solid. Optionally, you can take radiation into account.

To select the physics models for the mold:

1. For the physics continuum of the mold, select the following models:

Group Box	Model
Time	**Implicit Unsteady**
Material	**Solid**
Optional Models	**Segregated Solid Energy**
Equation of State	**User Defined EOS**
Optional Models	**Radiation**
Radiation	**Surface-to-Surface Radiation**
Radiation Spectrum	**Gray Thermal Radiation**

Selecting the Pure Thermal Model

A pure thermal simulation simulates only the solidification process of the melt. The solution of the flow equations is de-activated and only the solution of the energy equation is activated. This simulation type lets you analyze the behavior of the solidifying melt with respect to temperature without any flow.

During a solidification process, the total mass of a solidifying part does not change. For a pure thermal simulation, the solidifying part cannot move as a consequence of shrinking. If the density is not constant for a pure thermal simulation, for example, if density depends strongly on temperature, any changes in density result in a change of the system mass. To remedy this behavior, STAR-CCM+ applies a corrective energy source term when the pure thermal model is activated.

Before you begin, the following physics models must be selected:

- Implicit Unsteady
- Casting
- Segregated Multiphase Temperature
- Multiphase Equation of State

To select the pure thermal model:

1. Open the *Physics Model Selection* dialog of the physics continuum of the melt.
2. From the *Optional Models* group box, select **Pure Thermal**.

Selecting the Variable Density Compensation Model

You use the variable density compensation model for any solid region with a non-constant solid density. The variable density compensation model applies the corrective energy source term. The corrective energy source is a source term that compensates the changes of the system mass in the energy transport equation due to shrinkage of a solidifying part with temperature-dependent density.

Before you begin, the following models must be activated:

- Segregated Solid Energy
- Implicit Unsteady
- Casting
- User-Defined EOS or Polynomial EOS

To select the variable density compensation model:

1. Open the *Physics Model Selection* dialog of the physics continuum of the solid mold.
2. From the *Optional Models* group box, select **Variable Density Compensation**.

Working With the Casting Material Database

The accuracy of a casting simulation depends on the quality and completeness of the material data that you use. Therefore, a database that contains certified and qualified property data for casting is essential for high quality casting simulations.

To meet these requirements, STAR-CCM+ provides the Casting Material Database. This database is a component of STAR-Cast.

The Casting Material Database contains three classes of material data:

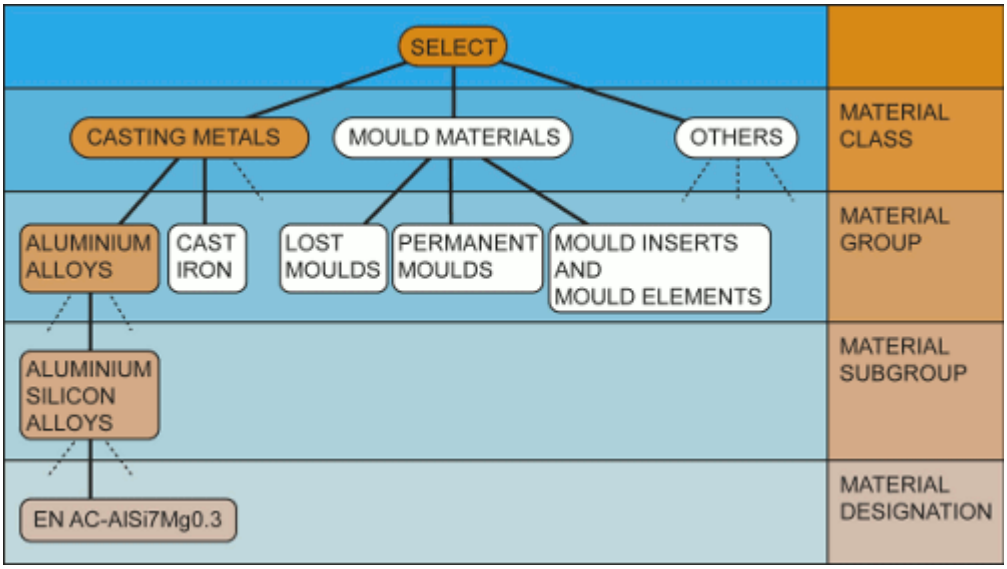
- Common data: data accessible to the licensee. This data is provided by ACCESS and includes datasets for the most frequently posed problems.
- Private data: private and confidential customer data that is provided by a person in authority within your company.

  See Working with Private Material Data.
- User-defined data: data provided by and accessible only to you. This data class adds the flexibility to create custom data from different sources.

  See Working with User-Defined Material Data.

The Casting Material Tree Structure

In the Casting Material Database, casting materials are categorized according to the tree structure shown below:



Materials are classified into three main groups, or material classes, as follows:

- Casting metals
- Mold materials

- Others

Each class is divided into a material group, and a material subgroup. The material designations are at a fourth level.

Using the Casting Material Database

You access the Casting Material Database with the Casting Material Database Editor.

See Accessing the Casting Material Database Editor.

The Casting Material Database Editor lets you do the following:

- Select a casting material from the Casting Material Database.

  See Selecting a Casting Material.
- View the physical properties of a selected casting material.

  See Viewing Available Material Properties for a Casting Material.
- Viewing the datasets that belong to a selected casting material.

  You can extract graphs and material data information from datasets, and integrate this information into casting simulation reports.

  See Viewing and Exporting Datasets.
- Compare the physical properties of different casting materials.

  See Comparing the Physical Properties of Different Materials.
- Export selected casting material data to STAR-CCM+.

  Material data that is exported to STAR-CCM+ can be included in your casting simulation.

  See Exporting Casting Material Data to STAR-CCM+.
- Create and modify private and user-defined casting material data.

  See Working with Private Material Data and Working with User-Defined Material Data.

Contents:

Accessing the Casting Material Database Editor
Selecting a Casting Material
Selecting a Material Designation Using the Drop-down Menus
Selecting a Material Designation Using the Icon Lists
Selecting a Material Designation Using a String Search
Selecting a Material Designation Using Its Dataset Number
Selecting a Material Designation Using the Search History
Viewing Details of the Selected Material Designation
Viewing Available Material Properties for a Casting Material
Viewing and Exporting Datasets
Comparing the Physical Properties of Different Materials
Exporting Casting Material Data to STAR-CCM+
Working with Private Data
Working with User-Defined Data

Working with the STAR-Cast Mat Database Builder

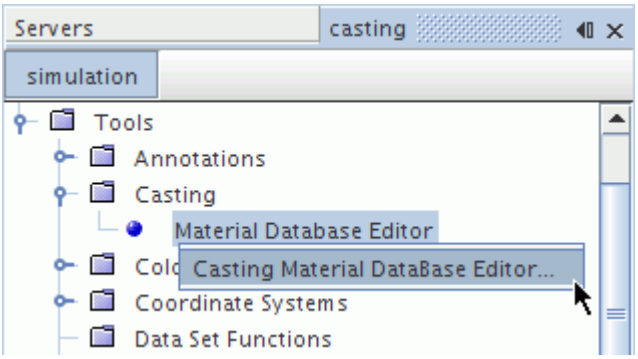
Accessing the Casting Material Database Editor

The Casting Material Database Editor, also referred to as STAR-Cast mat, allows you to search for material data, view it, and export it to STAR-CCM+.

The material data that you export to STAR-CCM+ can be included in your casting simulation.

To open the Casting Material Database Editor:

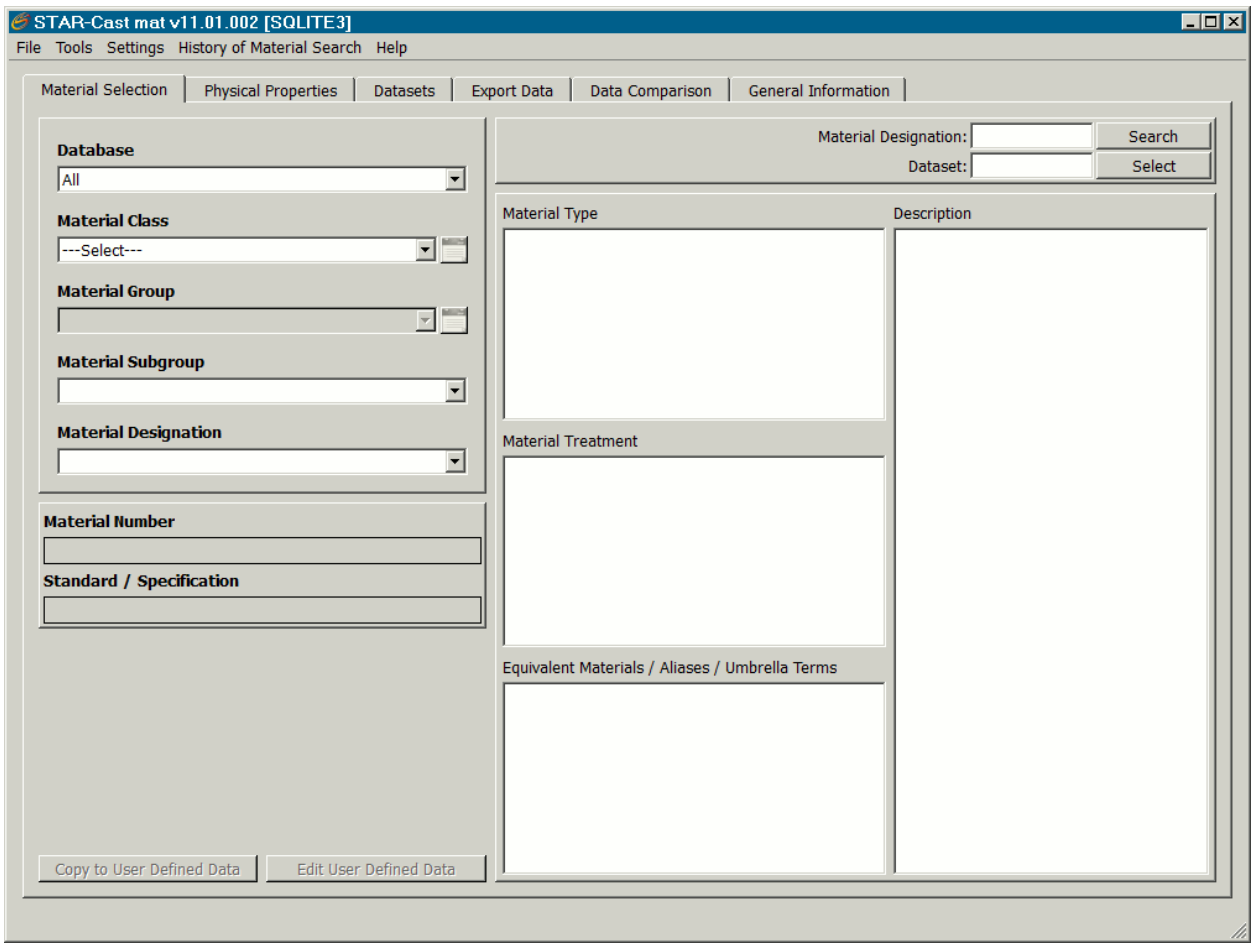
1. Activate the casting model.
2. Right-click the **Tools** > **Casting** > **Material Database Editor** node and select **Casting Material Database Editor**.



When you open the Casting Material Database Editor for the first time, the following disclaimer notice appears.

3. Read the provisions governing database rights and the disclaimer of liability carefully.
4. If you agree with the provisions, select the agreement radio button and click **OK**.

   The Casting Material Database Editor appears.



You cannot access the STAR-CCM+ workspace while the Casting Material Database Editor is open.

The Casting Material Database Editor closes automatically when you export a collection of materials directly to STAR-CCM+.

To close the Casting Material Database Editor manually:

5. Select **File** > **Exit**.

Contents:

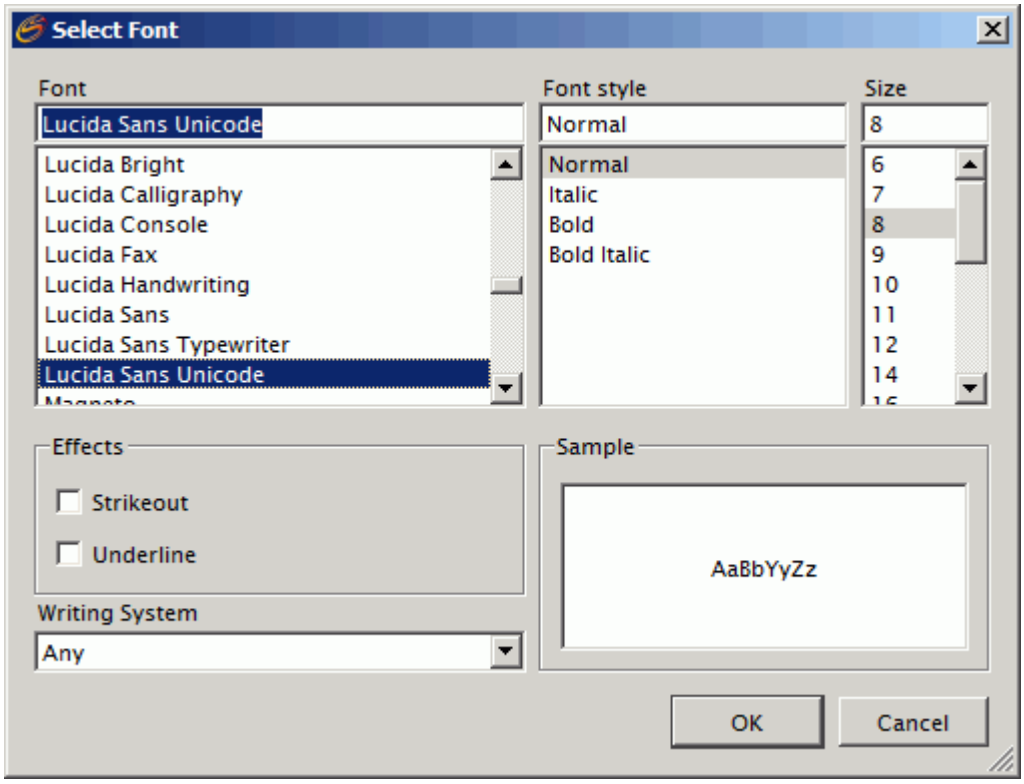
Changing the Appearance of the Casting Material Database Editor
Getting Information on the Casting Material Database

Changing the Appearance of the Casting Material Database Editor

You can change the appearance of the Casting Material Database Editor to suit your requirements. You can resize the window and set the text style, size, and font. You can also restore the default settings when necessary.

To specify the font to use in the Casting Material Database Editor:

1. Select **Settings** > **Font**.
2. In the *Select Font* dialog, make the appropriate changes to the font settings.



3. To confirm the new settings, click **OK**.

To restore the Casting Material Database Editor to the default font settings and window size:

- Select **Settings** > **Reset GUI**.

Getting Information on the Casting Material Database

The Casting Material Database includes online help information.

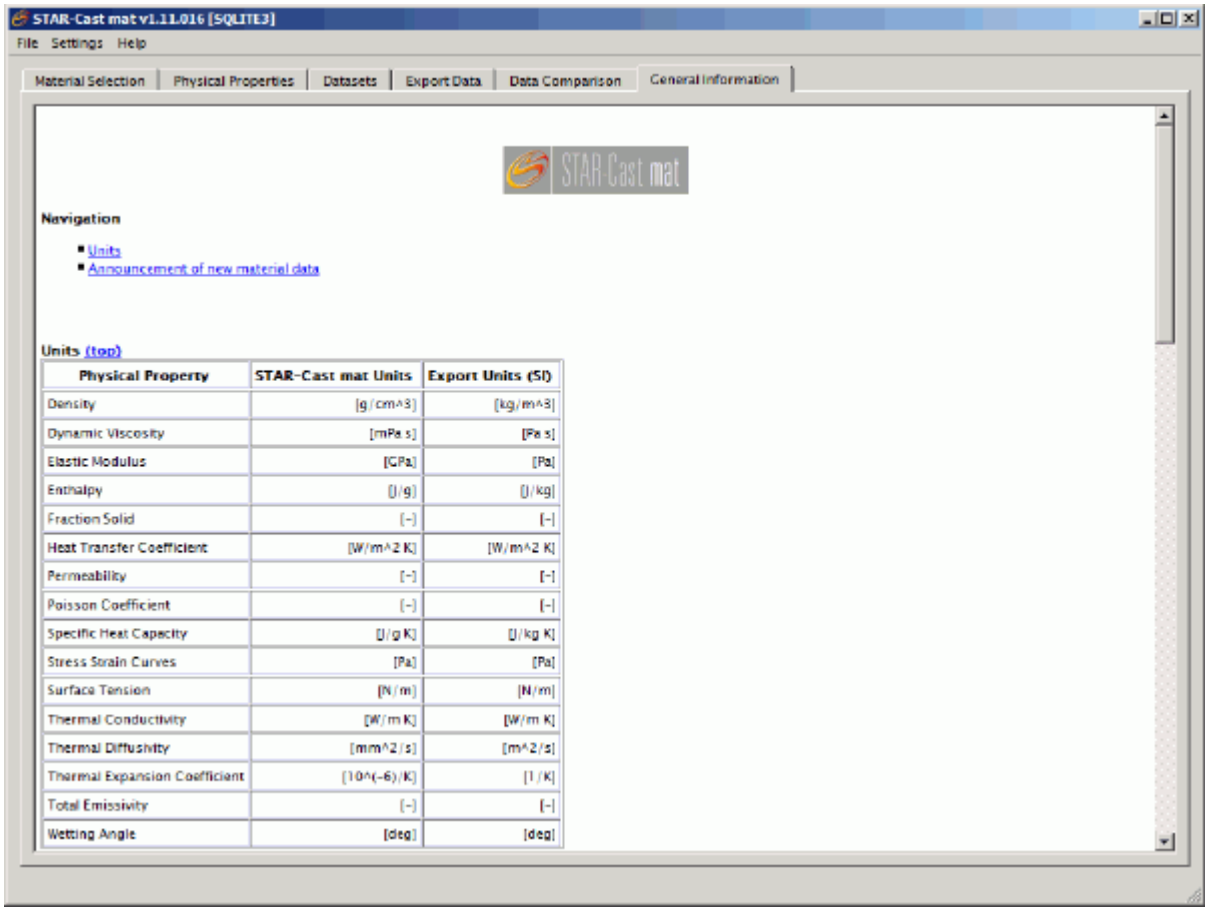
To access the Casting Material Database online help, do either of the following:

- Select **Help** > **Help**.
- Press F1.

To get product information specific to the Casting Material Database, do either of the following:

- Select **Help** > **About**.
- Click the *General Information* tab.

  The *General Information* panel displays the units that the Casting Material Database uses, and lists the new material designations that have been added to the database recently.



Physical Property	STAR-Cast mat Units	Export Units (SI)
Density	[g/cm^3]	[kg/m^3]
Dynamic Viscosity	[mPa s]	[Pa s]
Elastic Modulus	[GPa]	[Pa]
Enthalpy	[J/g]	[J/kg]
Fraction Solid	[-]	[-]
Heat Transfer Coefficient	[W/m^2 K]	[W/m^2 K]
Permeability	[-]	[-]
Poisson Coefficient	[-]	[-]
Specific Heat Capacity	[J/g K]	[J/kg K]
Stress Strain Curves	[Pa]	[Pa]
Surface Tension	[N/m]	[N/m]
Thermal Conductivity	[W/m K]	[W/m K]
Thermal Diffusivity	[mm^2/s]	[m^2/s]
Thermal Expansion Coefficient	[10^(-6)/K]	[1/K]
Total Emissivity	[-]	[-]
Wetting Angle	[deg]	[deg]

Selecting a Casting Material

You select the casting materials that you want to use from the Casting Material Database Editor. By default, common and private data is displayed, but user-defined data is hidden.

To display user-defined data:

In the Casting Material Database Editor, on the Settings menu, activate **Show User Defined Data**.

For more information, see Working with User-Defined Material Data.

All material designations, including user-defined ones, are displayed.

To select a casting material:

1. In the Casting Material Database Editor, click the *Material Selection* tab.
2. In the *Material Selection* panel, select the appropriate material designation by doing any of the following:
   - Choose the appropriate items from the four drop-down menus that reflect the tree structure.

     See Selecting a Material Designation Using the Drop-down Menus.
   - Select the appropriate material by using the icon lists.

     See Selecting a Material Designation Using the Icon Lists.
   - Run a string search for the material designation.

See Selecting a Material Designation Using a String Search.

- Enter a unique dataset number.

  See Selecting a Material Designation Using Its Dataset Number.

- Select a recently searched material designation from the search history.

  See Selecting a Material Designation Using the Search History.

Details of the selected material are displayed in the *Material Selection* panel.

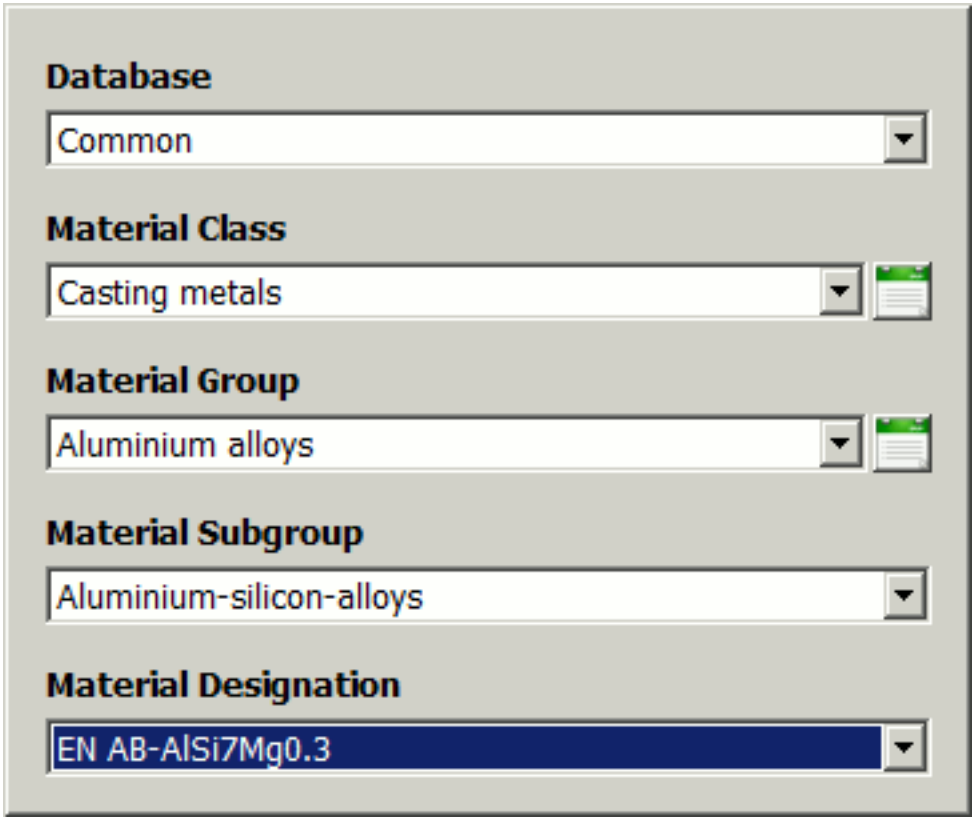
See Viewing Details of the Selected Material Designation.

You can also select materials from the Physical Properties panel. However, to do so, you must have previously selected another material designation.

See Selecting a Material Designation from a Material Property.

Selecting a Material Designation Using the Drop-down Menus

You can select a material designation from the four drop-down menus that reflect the tree structure.



To select a material designation:

1. Choose the database from the Database drop down menu.
2. Choose the material class from the Material Class drop down menu.
3. Select the appropriate material group from the Material Group menu.

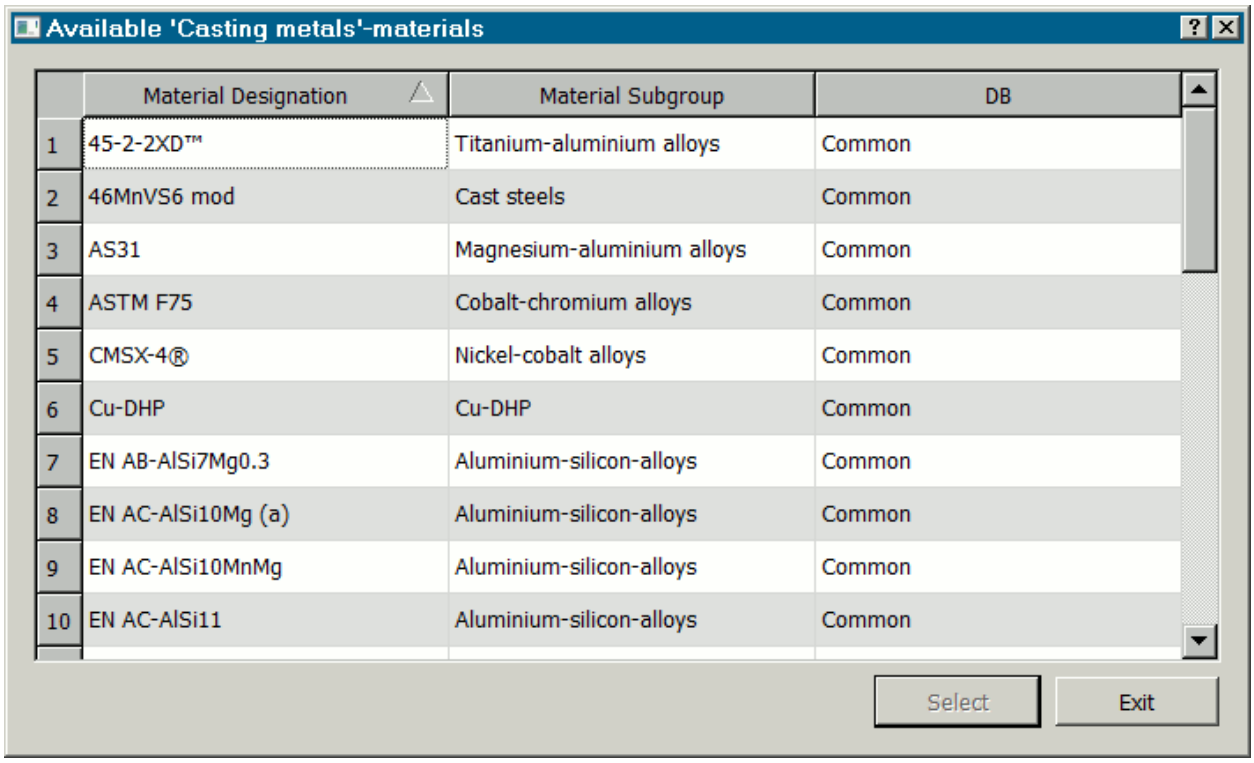
4. Select the appropriate material subgroup from the Material Subgroup menu.
5. Select a material designation from the Material Designation menu.

Selecting a Material Designation Using the Icon Lists

You can select a material designation from the icon lists that are available to the right of the Material Class and Material Group drop-down menus.

To select a material designation from an icon list:

1. Click the appropriate icon list button.
2. In the *Available Materials* dialog, click the material designation that you want to use.



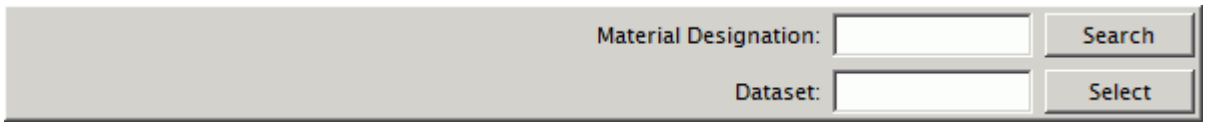
	Material Designation	Material Subgroup	DB
1	45-2-2XD™	Titanium-aluminium alloys	Common
2	46MnVS6 mod	Cast steels	Common
3	AS31	Magnesium-aluminium alloys	Common
4	ASTM F75	Cobalt-chromium alloys	Common
5	CMSX-4®	Nickel-cobalt alloys	Common
6	Cu-DHP	Cu-DHP	Common
7	EN AB-AlSi7Mg0.3	Aluminium-silicon-alloys	Common
8	EN AC-AlSi10Mg (a)	Aluminium-silicon-alloys	Common
9	EN AC-AlSi10MnMg	Aluminium-silicon-alloys	Common
10	EN AC-AlSi11	Aluminium-silicon-alloys	Common

3. Click **Select**.

Selecting a Material Designation Using a String Search

You can select a material designation using a string search.

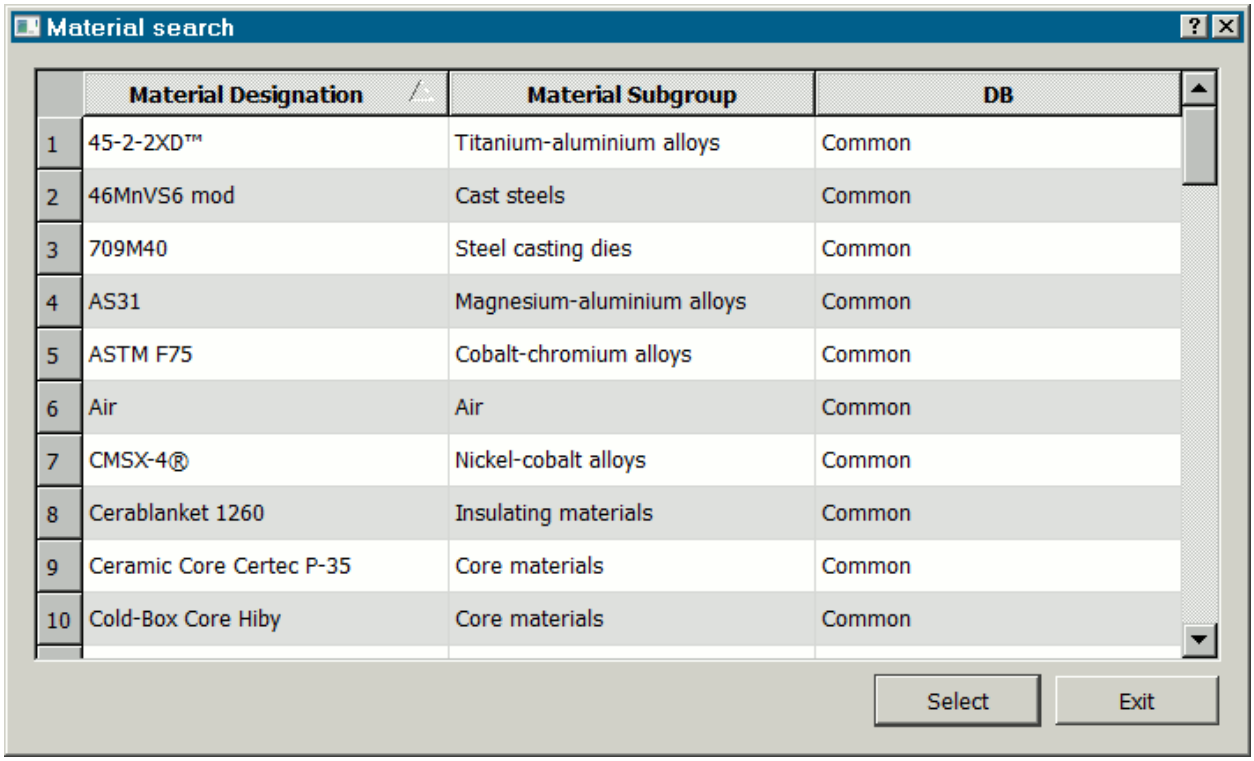
1. In the Material Designation box, enter the name of the material that you want (or a related string).



If you want to see the complete set of material designations that are available in the Casting Material Database, leave the *Material Designation* box empty.

2. Click **Search**.

   The *Material Search* dialog appears.



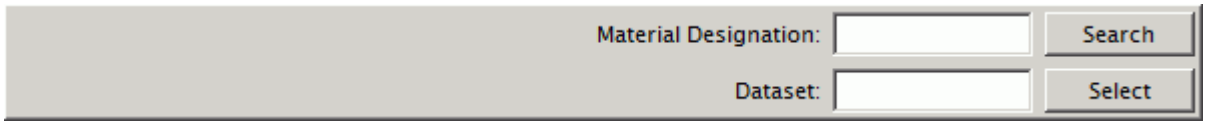
	Material Designation	Material Subgroup	DB
1	45-2-2XD™	Titanium-aluminium alloys	Common
2	46MnVS6 mod	Cast steels	Common
3	709M40	Steel casting dies	Common
4	AS31	Magnesium-aluminium alloys	Common
5	ASTM F75	Cobalt-chromium alloys	Common
6	Air	Air	Common
7	CMSX-4®	Nickel-cobalt alloys	Common
8	Cerablanket 1260	Insulating materials	Common
9	Ceramic Core Certec P-35	Core materials	Common
10	Cold-Box Core Hiby	Core materials	Common

3. In the *Material Search* dialog, click the material designation that you want.
4. Click **Select**.

Selecting a Material Designation Using Its Dataset Number

You can select a material designation by specifying the dataset number of one of its physical properties:

1. In the Dataset box, enter the appropriate dataset number.



2. Click **Select**.

   The Datasets tab appears, displaying details of the selected dataset. The corresponding material is automatically selected in the *Material Selection* tab.

See Viewing the Details of a Dataset.

Selecting a Material Designation Using the Search History

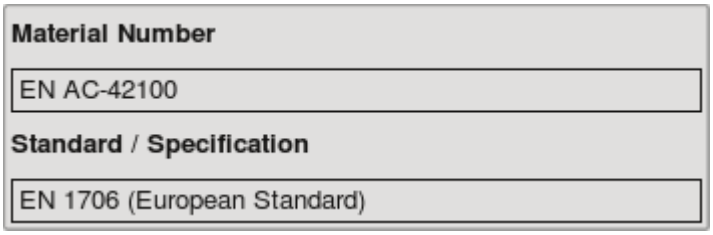
You can select a recently searched material designation from the search history.

To select a material designation from the search history:

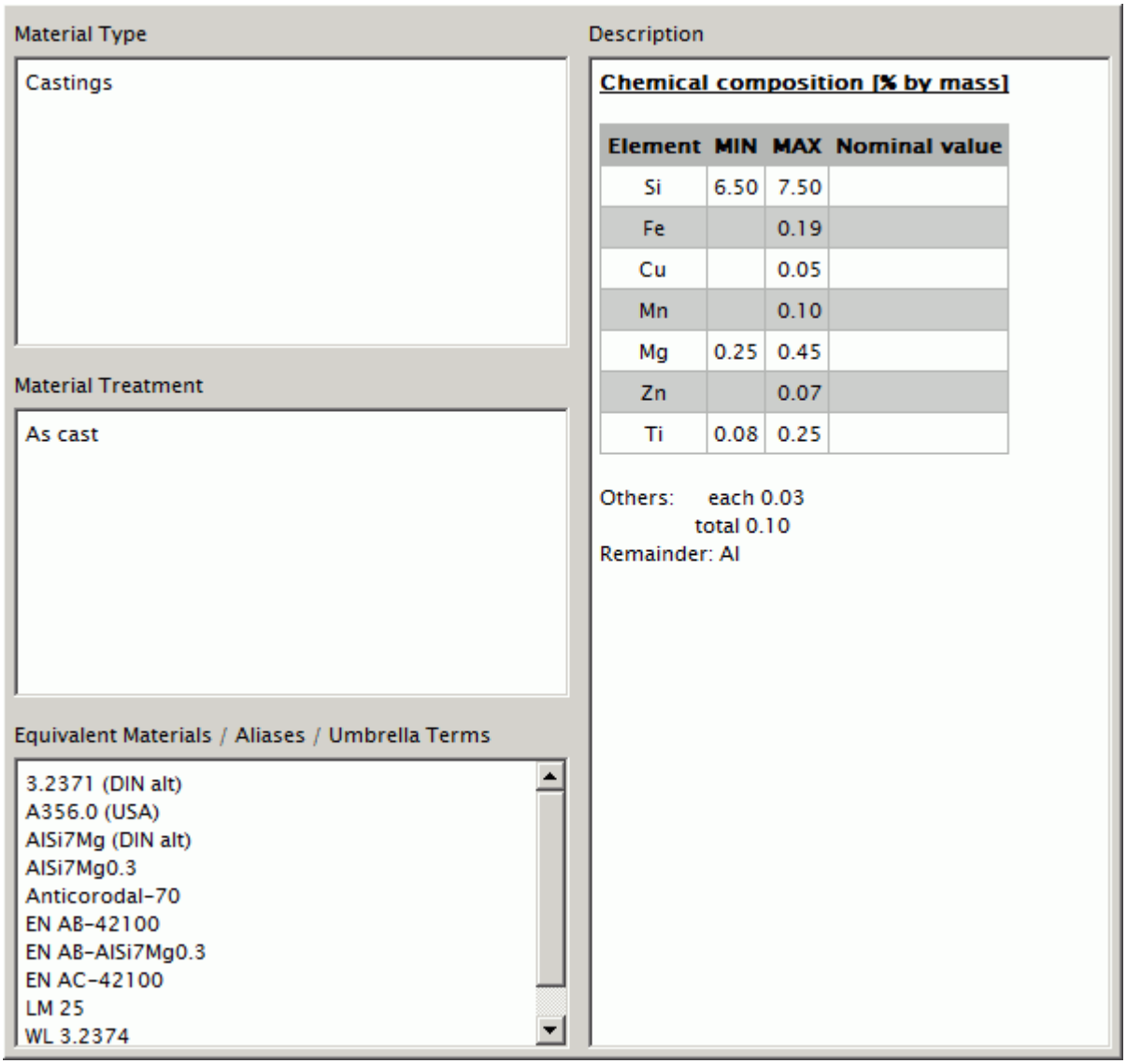
1. Select **History of Material Search** from the Casting Material Database Editor menu.
   A list displays all material designations that you have searched for recently.
2. Click the material designation that you want to use.

Viewing Details of the Selected Material Designation

After you have selected a material designation, the material number and the standard (for example, European, American, British) or a company-defined specification are displayed below the material drop-down menus in the *Material Selection* panel.



The *Material Selection* panel contains information text boxes that display general information about the selected material designation.



- **Material Type:** Nature of the material (as applicable) to which the Standard/Specification refers, or the specimens to which the measurements correspond.

  For example: castings, alloyed ingots for remelting, and alumina-based shell.
- **Material Treatment:** Describes how the material has been processed.

  For example: heat-treated and as cast condition.
- **Equivalent Materials/Aliases/Umbrella Terms:** A list of equivalent materials.

  For example, alloys with similar composition or specifications, and aliases for the material designation in the Casting Material Database dialog.
- **Description:** The composition or specification of the material (as applicable) and other information about the composition and manufacturing or special properties.

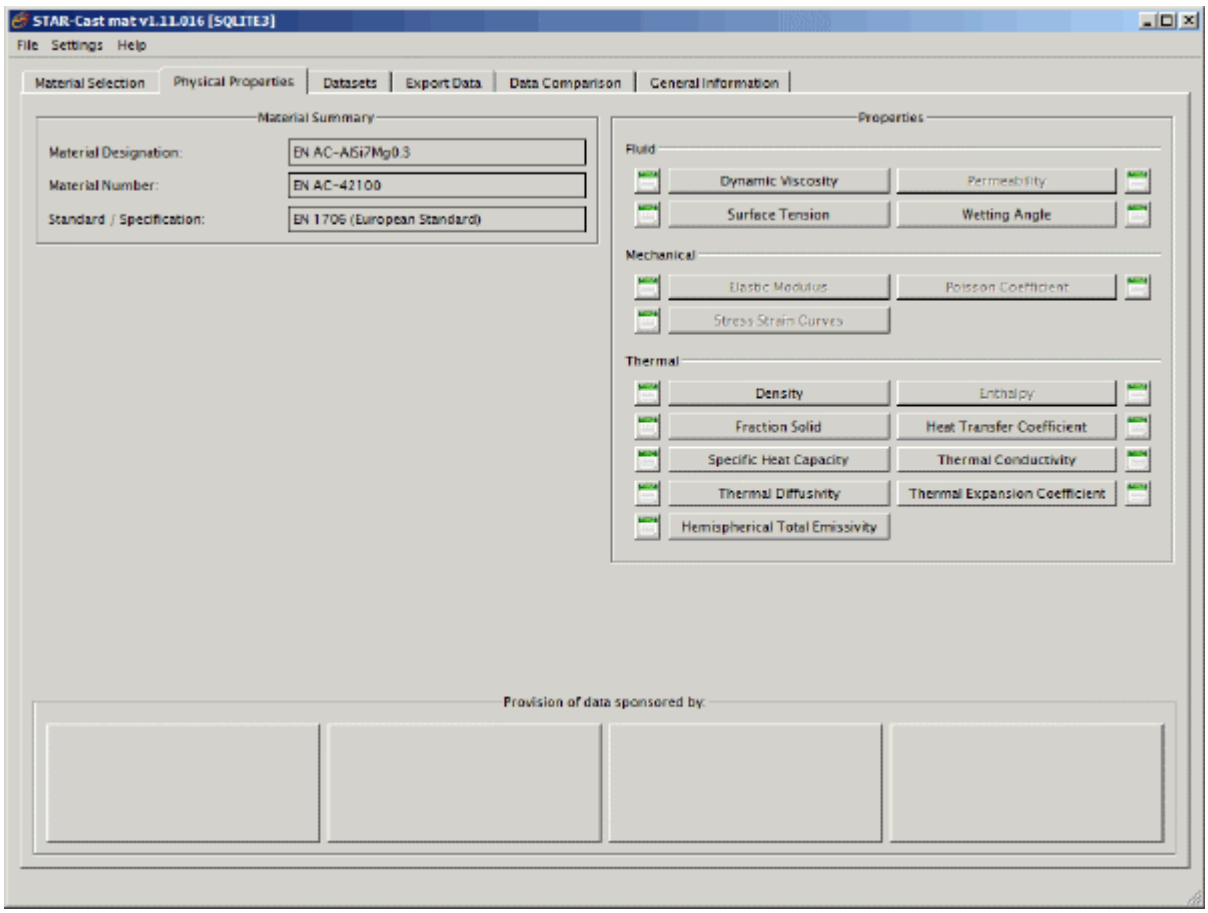
  For example, porosity.

Viewing Available Material Properties for a Casting Material

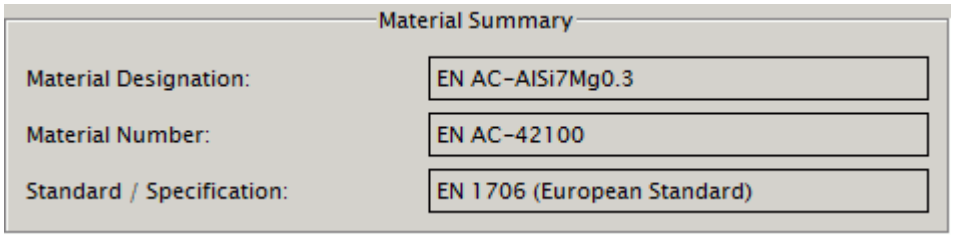
The *Physical Properties* panel of the Casting Material Database Editor lets you view the available material properties for the selected material designation.

To view the material properties for a casting material:

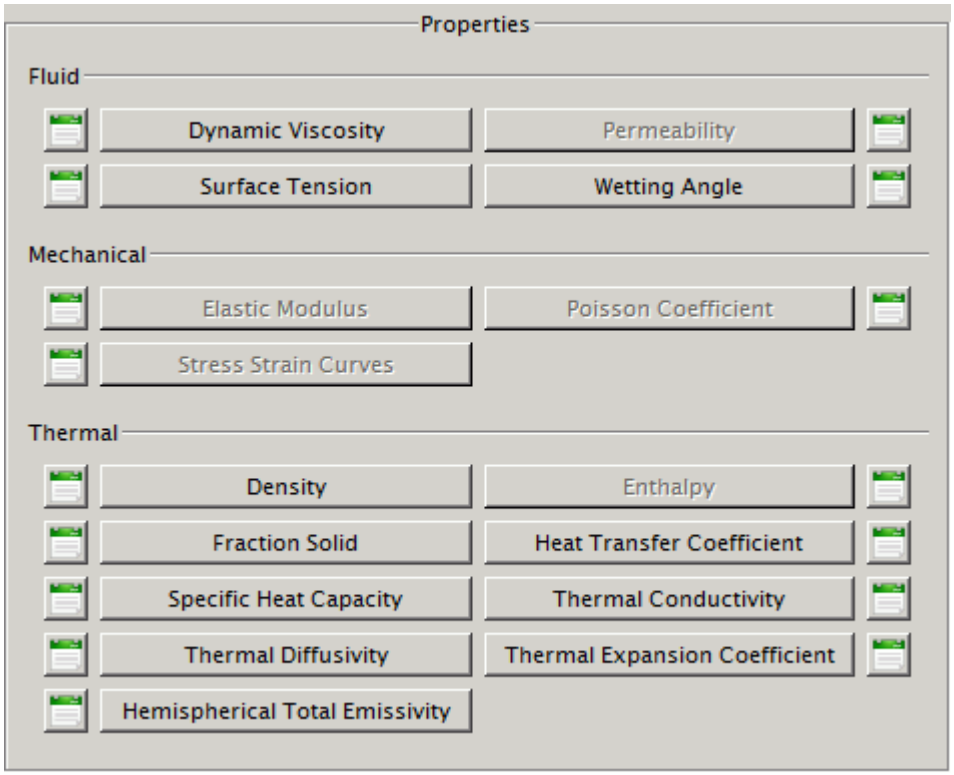
1. In the Casting Material Database Editor, click the *Physical Properties* tab.



If a material designation is selected, the *Material Summary* group box displays the Material Designation, the Material Number (as applicable), and the Standard/Specification (as applicable).



The *Properties* group box lists physical properties under three headings: Fluid, Mechanical, and Thermal.



2. In the *Properties* group box, click the property that you want to view.

The *Datasets* panel opens, and displays dataset information for the selected property.

See Viewing the Details of a Dataset.

If data for a property is not available, the corresponding property button is grayed out.

Contents:

Selecting a Material Designation from a Material Property

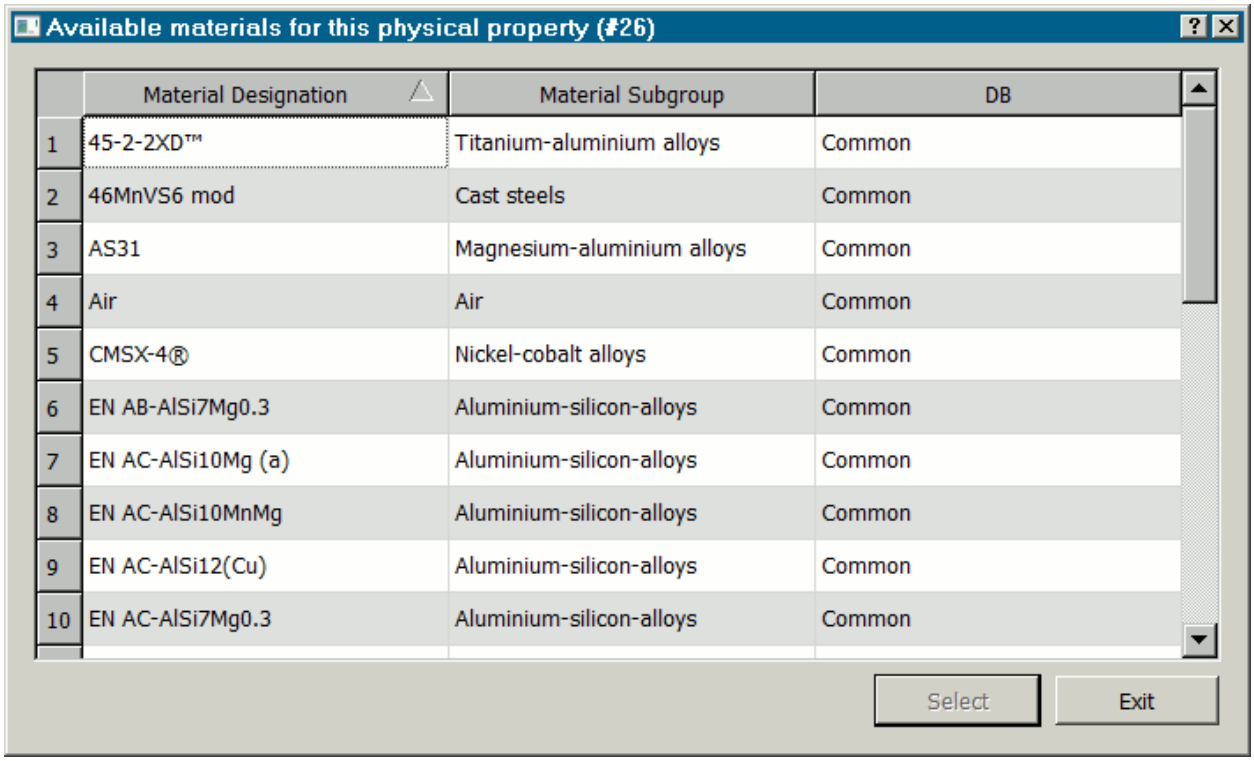
Selecting a Material Designation from a Material Property

When a material designation is loaded in the *Physical Properties* panel, you can select a different material designation that has a particular property.

To select another material designation from a material property:

1. Click the icon list button that is next to the appropriate property button.

   A dialog listing all of the available materials for the corresponding property appears.



2. Click the material designation that you want to select.
3. Click **Select**.

   Details of the selected material designation are displayed in the *Material Summary* group box.

Viewing and Exporting Datasets

A dataset is a set of data values that represents the variation of a physical property with temperature. In the Casting Material Database, each dataset is assigned a unique number and is associated with a particular casting material designation. A particular physical property of a casting material designation can have two or more datasets that are associated with it.

You can view the details of a particular dataset, or view the details of all of the datasets that are associated with a particular property of a material.

See Viewing the Details of a Dataset.

To see the physical property data in graphical form, you can view and compare the dataset curves.

See Viewing Dataset Curves.

If two or more datasets are available for a particular property, you can select the dataset that is most appropriate for your simulation.

See Selecting a Dataset to Include in an Exported Material.

The units of measurement for a physical property in the Casting Material Database are expressed in the system of units that is the most commonly used in the literature. During the data export, the units are automatically converted according to the requirements of STAR-CCM+.

Contents:

Viewing the Details of a Dataset
Viewing Dataset Curves
Selecting a Dataset to Include in an Exported Material

Viewing the Details of a Dataset

The Casting Material Database Editor can display detailed information on a particular dataset.

To view the details of a particular dataset:

- In the Casting Material Database Editor, do one of the following:
  - In the *Datasets* tab, in the Dataset box, enter the appropriate dataset number.
  - In the *Material Selection* tab, in the Dataset box, enter the appropriate dataset number.
- Click **Select**.

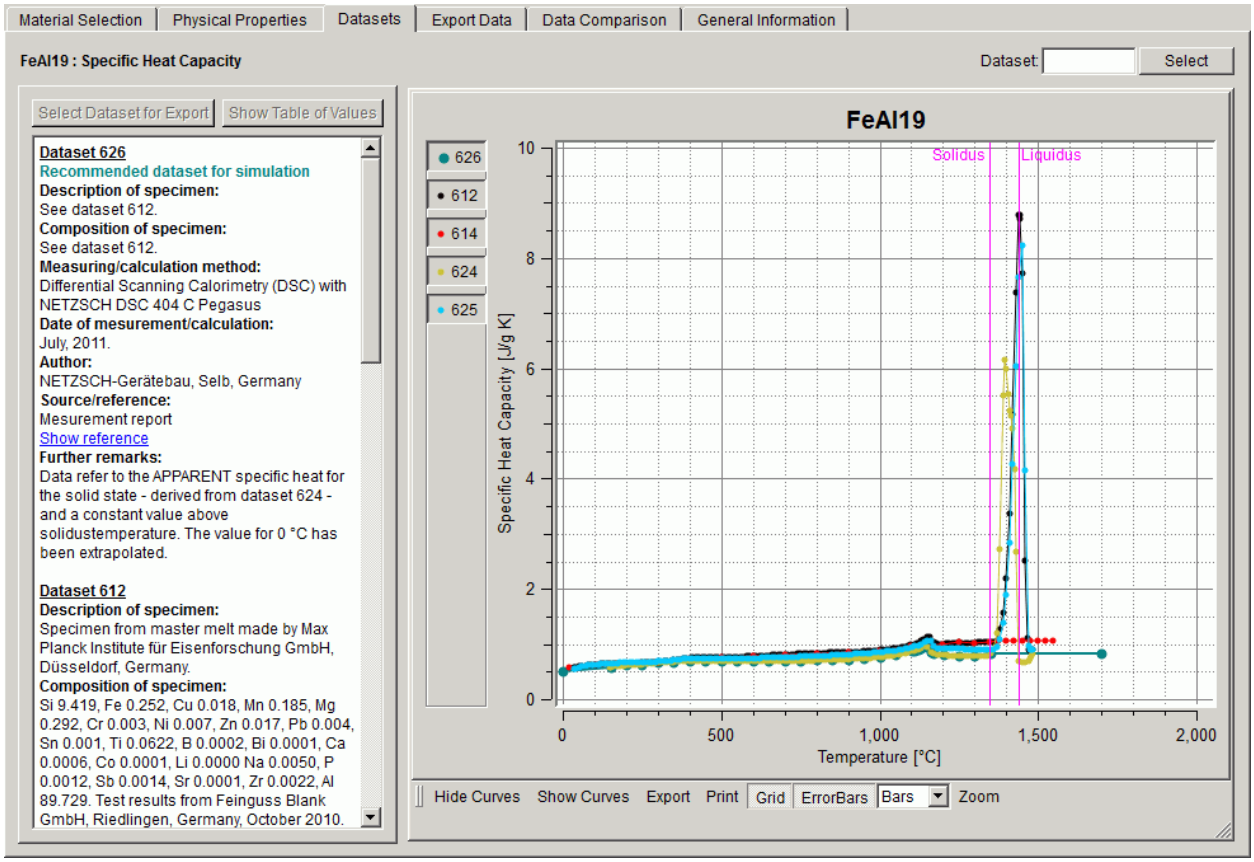
  The selected dataset is displayed. The material with which the dataset is associated is also selected and is displayed in the *Material Selection* tab.

To view the details of all of the datasets that are associated with a particular property of a casting material:

- Select the appropriate casting material.

  See Selecting a Casting Material.
- In the *Physical Properties* tab, in the *Properties* group box, click the appropriate physical property.

  The *Datasets* panel appears, displaying details of all of the datasets that are associated with the specified casting material and physical property.



Viewing Detailed Information about a Dataset

You can view a description of each dataset. You can also view the raw data for a dataset, including the error tolerances on each value.

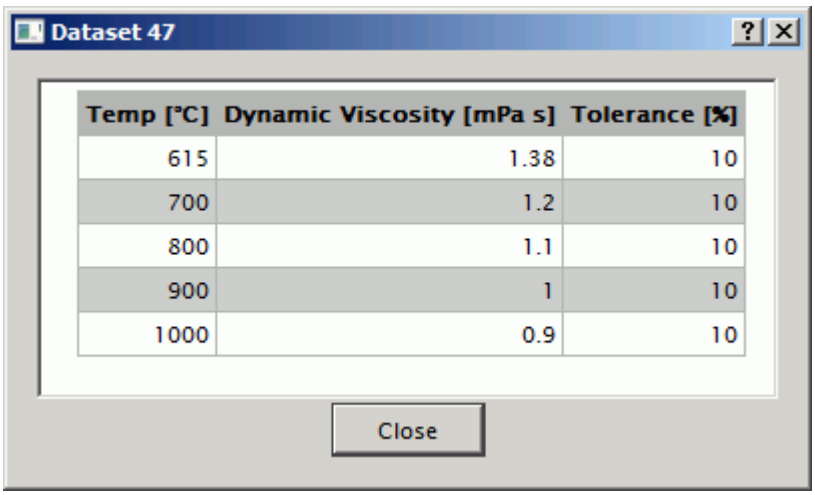
The text box at the left of the *Datasets* panel contains information on each selected dataset. This information includes a description of the dataset and details of its origin.

To view the raw data for a dataset:

- In the vertical bar to the left of the graph, select the dataset that you want.
- Ensure that all other datasets are deselected.

  If two or more datasets are selected, the **Show Table of Values** button remains grayed out.
- Click **Show Table of Values**.

The raw values of the dataset, including the error tolerance on each value, are displayed in tabular form.



Temp [°C]	Dynamic Viscosity [mPa s]	Tolerance [%]
615	1.38	10
700	1.2	10
800	1.1	10
900	1	10
1000	0.9	10

Viewing Dataset Curves

To see the temperature dependence of the physical property in graphical form, you can view the dataset curves. The symbol and ID number of each available dataset is shown in the vertical bar to the left of the physical property axis. You can select the datasets to display in the graph.

To display particular datasets:

- In the vertical bar, activate the appropriate dataset numbers.

To remove particular datasets from the display:

- Deactivate the appropriate dataset numbers.

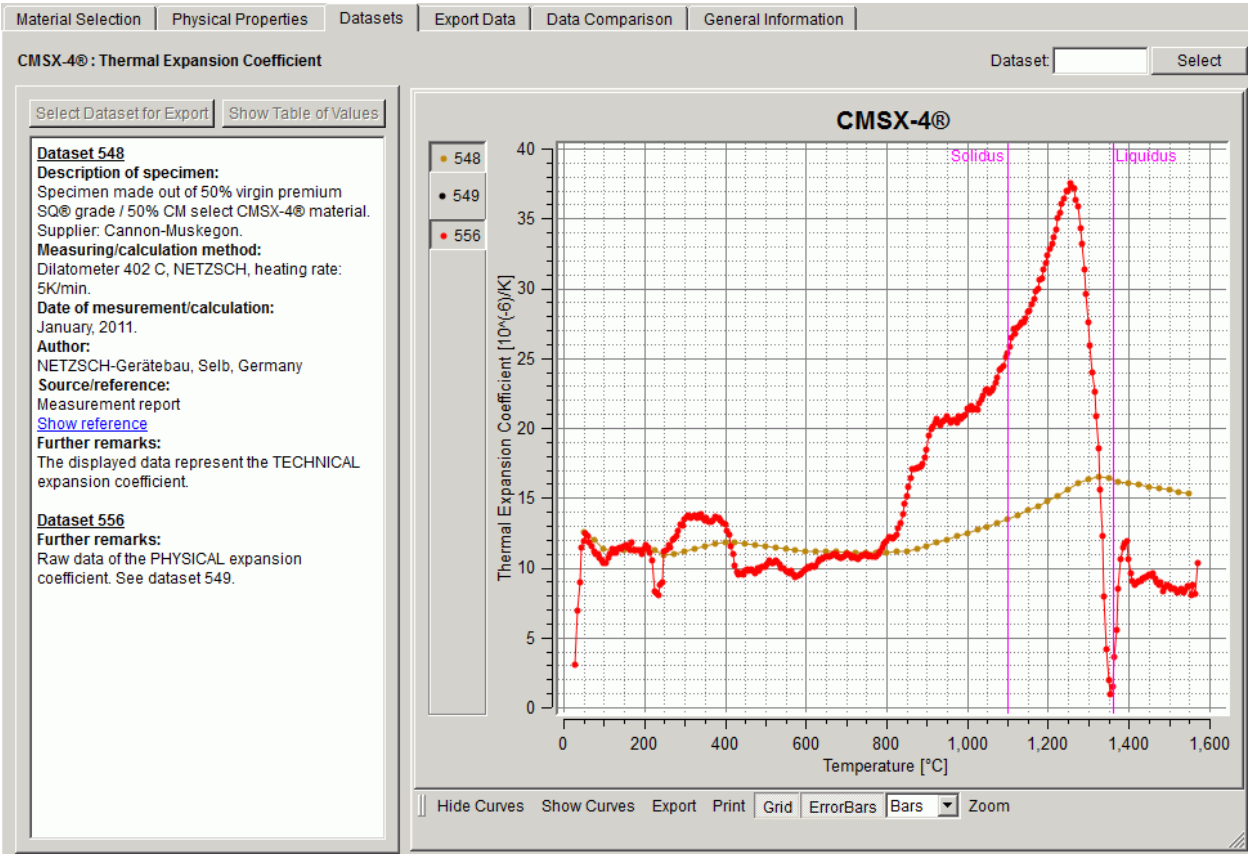
To display all of the available datasets:

- In the lower menu bar, click **Show Curves**.

If you want to deactivate all of the datasets:

- Click **Hide Curves**.

  The selected datasets are displayed.



Ordinates represent the physical property to which the plotted data relates. The abscissa denotes temperature. The graph displays liquidus and solidus temperatures (depending on the range of temperatures plotted), and latent heat information. A symbol on the graph represents each value in the dataset.

To determine the value of a particular dataset point:

- Hover the mouse over the appropriate dataset point.

  Its coordinates are displayed next to the pointer.

Changing the Appearance of a Graph

You can use the following options to change the appearance of the graph.

Grid	Toggles the background grid display.
ErrorBars	Toggles the display of error bars on the data points. The error bars reflect the tolerance that is specified on the data values. In the drop-down list, select the type of error bars to use: **Bars** **Tube**
Zoom	Lets you zoom in on a particular area of the graph. To select the area of interest, click one corner of the area and then drag the rectangle over the appropriate range. To reset the range to the default value, deactivate **Zoom**.

Saving and Printing a Graph

You can save a graph as a PDF file that you can use later, for example, to include in a report. You can also print the graph directly to your printer. The generated output includes the dataset symbols and ID numbers to the left of the property axis.

Export	Saves the graph as a PDF file, using the default name `myPlot.pdf`. The dataset information is saved automatically in a separate PDF file, using the same file name with _Refs appended.
Print	Prints the graph and the dataset information.

Selecting a Dataset to Include in an Exported Material

If two or more datasets are available for a material property, you can choose which dataset to use in your simulation. When you export the material as part of a material collection, the selected dataset is used to define the material property.

If a particular dataset is recommended, it is selected for export in the *Export Data* panel by default. If you want to use a different dataset, or if no dataset is selected, you can select the appropriate dataset.

To select a dataset for export:

1. In the vertical bar to the left of the graph, select the dataset that you want.
2. Ensure that all other datasets are deselected.

   If two or more datasets are selected, the **Select Dataset for Export** button remains grayed out.

3. Click **Select Dataset for Export**.

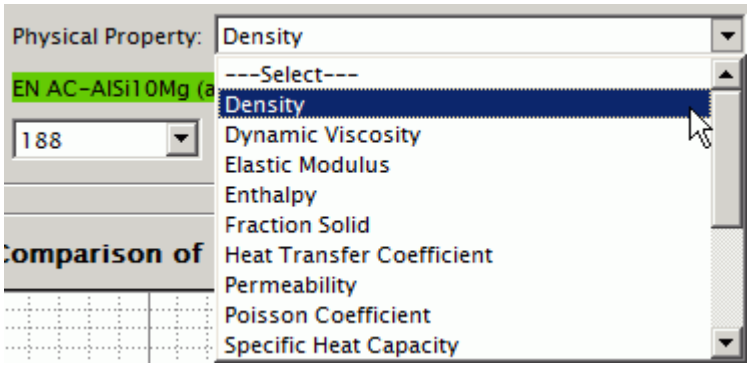
   The selected dataset number is shown in the *Selected Datasets* group box on the *Export Data* tab.

Comparing the Physical Properties of Different Materials

The *Data Comparison* panel lets you compare a single physical property for up to five different materials. This feature lets you compare the datasets that are available for particular materials, and select the materials and datasets that are the most appropriate to your simulation.

To compare a particular physical property for different materials:

1. In the Casting Material Database Editor, click the *Data Comparison* tab.
2. In the Physical Property box, click the down arrow and then select the physical property that you want to compare.

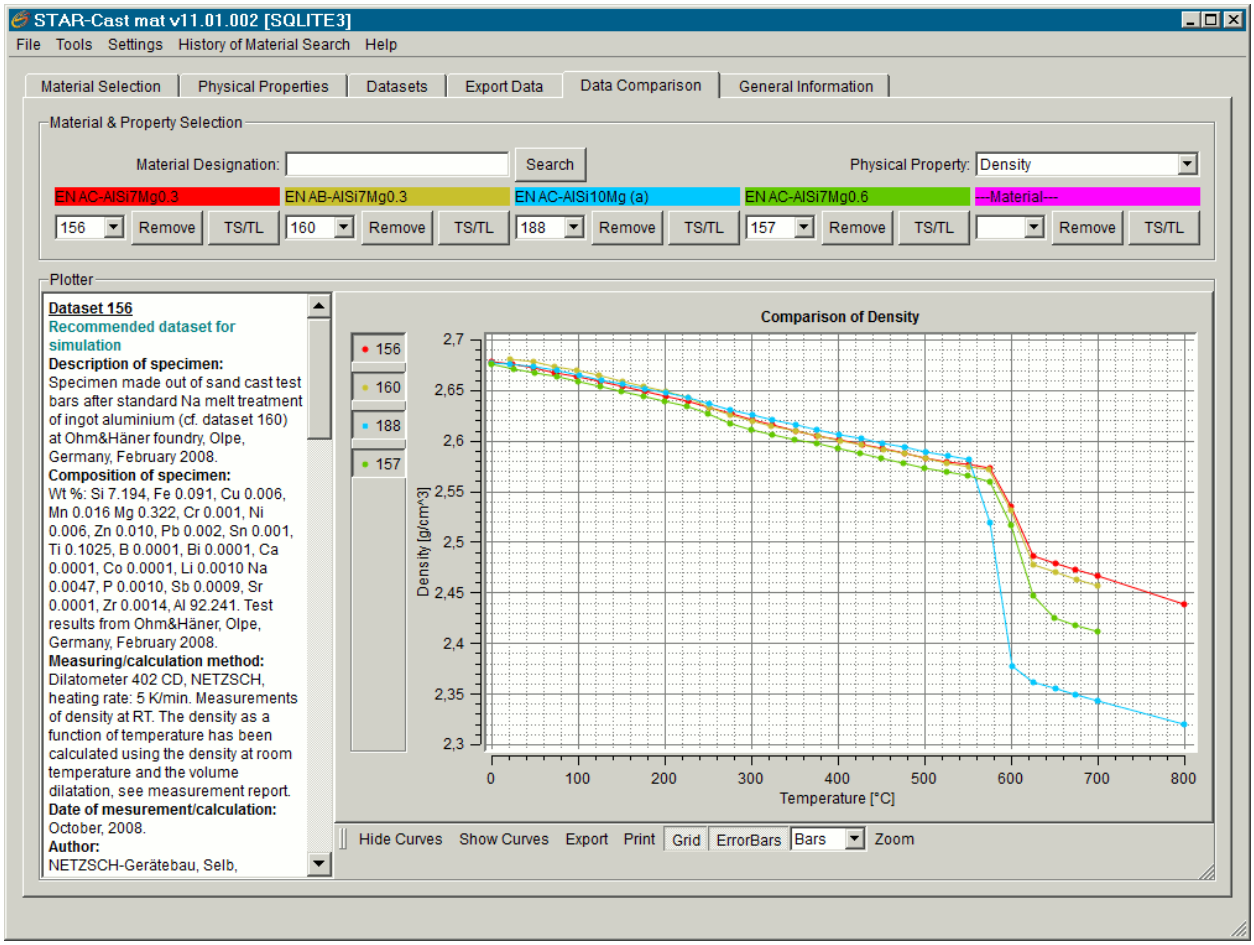


3. Select the materials that you want to compare:
   a) In the Material Designation box, enter all or part of the material name.

      If you want to list all available materials, leave the Material Designation box empty.
   b) Click **Search** to display the list of matched materials.
   c) In the *Material Search* dialog, select the materials that you want to compare.

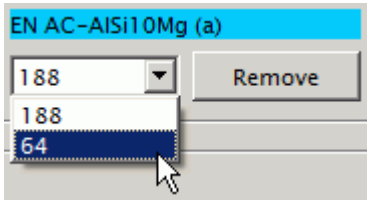
      You can select up to five materials. If you attempt to select any more, the **Select** button is grayed out.
   d) Click **Select**.

   The selected materials are added to the comparison slots on the *Data Comparison* panel.



If a material designation has two or more datasets that are associated with the selected physical property, the recommended curve and its documentation are shown by default.

4. If you want to select another dataset, click the drop-down list under the material designation and select the appropriate dataset number.



5. If you want to display solidus and liquidus temperatures, click **TS/TL** below the material designation.

   The temperature range, if available, is displayed.

You can remove or replace a material designation:

6. Click **Remove** below the material designation.

   The material designation field is cleared and the associated dataset is removed.

7. If you want to display a different material, click **Search** and then select the appropriate material designation.

You can use the graphical display features to view and compare the dataset curves. You can also save a graph as a PDF file to use later, or print a graph directly to your printer.

For more information, see Viewing Dataset Curves.

Exporting Casting Material Data to STAR-CCM+

The *Export Data* panel lets you prepare the collection of materials for export to STAR-CCM+.

You can export the collection of materials directly to STAR-CCM+, or save it as a file that you can import later.

Contents:

Preparing a Collection of Materials for Export
Exporting the Collection of Materials

Preparing a Collection of Materials for Export

In order to export material data from the Casting Material Database to STAR-CCM+, you prepare the appropriate collection of materials.

To add a material to the collection for each material that you want to export:

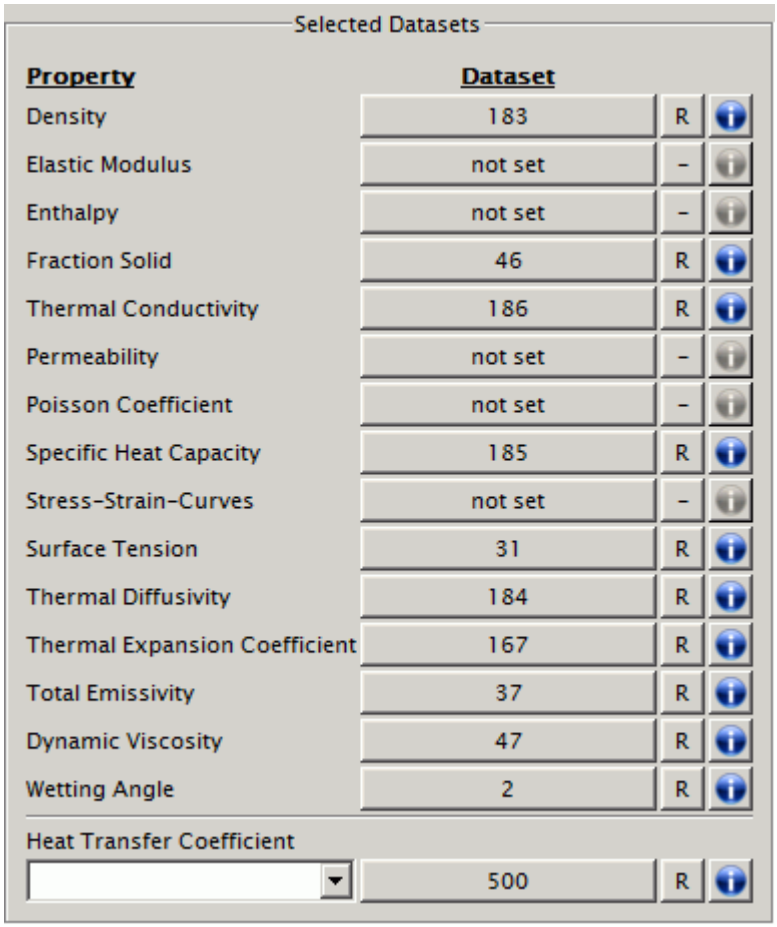
- In the Casting Material Database Editor, in the *Material Selection* tab, select the material designation.

  See Selecting a Casting Material.
- Review the datasets that are associated with the material and, if necessary, select the datasets that you want to use.

  If two or more datasets are available for a particular property, you can select the appropriate dataset in the Datasets panel.

  See Selecting a Dataset to Include in an Exported Material.
- Click the *Export Data* tab.

  The *Selected Datasets* group box shows the datasets that are currently selected for the material.



The letter **R** to the right of each dataset denotes that the recommended dataset is selected for that particular material property.

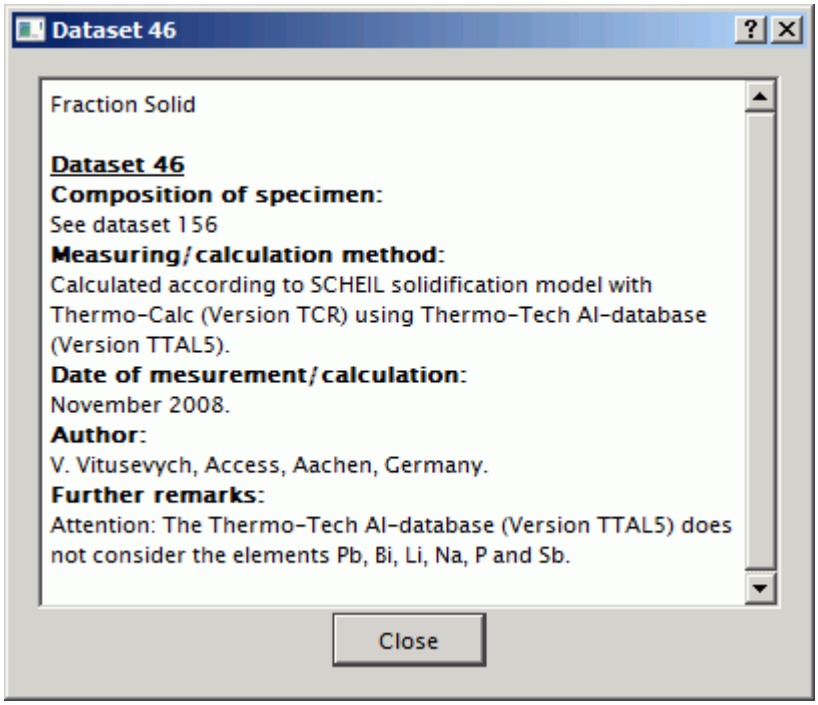
- Verify that the selected datasets are appropriate for your simulation.

  If alternative datasets are available for a particular property, you can change the selection in the Datasets panel.

  See Selecting a Dataset to Include in an Exported Material.

- If you want to see summary information for a selected dataset, click the blue button to the right of the dataset.

  The dataset documentation summary is displayed in the *Dataset* dialog.



- To add the current material designation (labeled at the top of the *Export Data* panel) to the collection of materials for export, click **Add Material to Collection**.

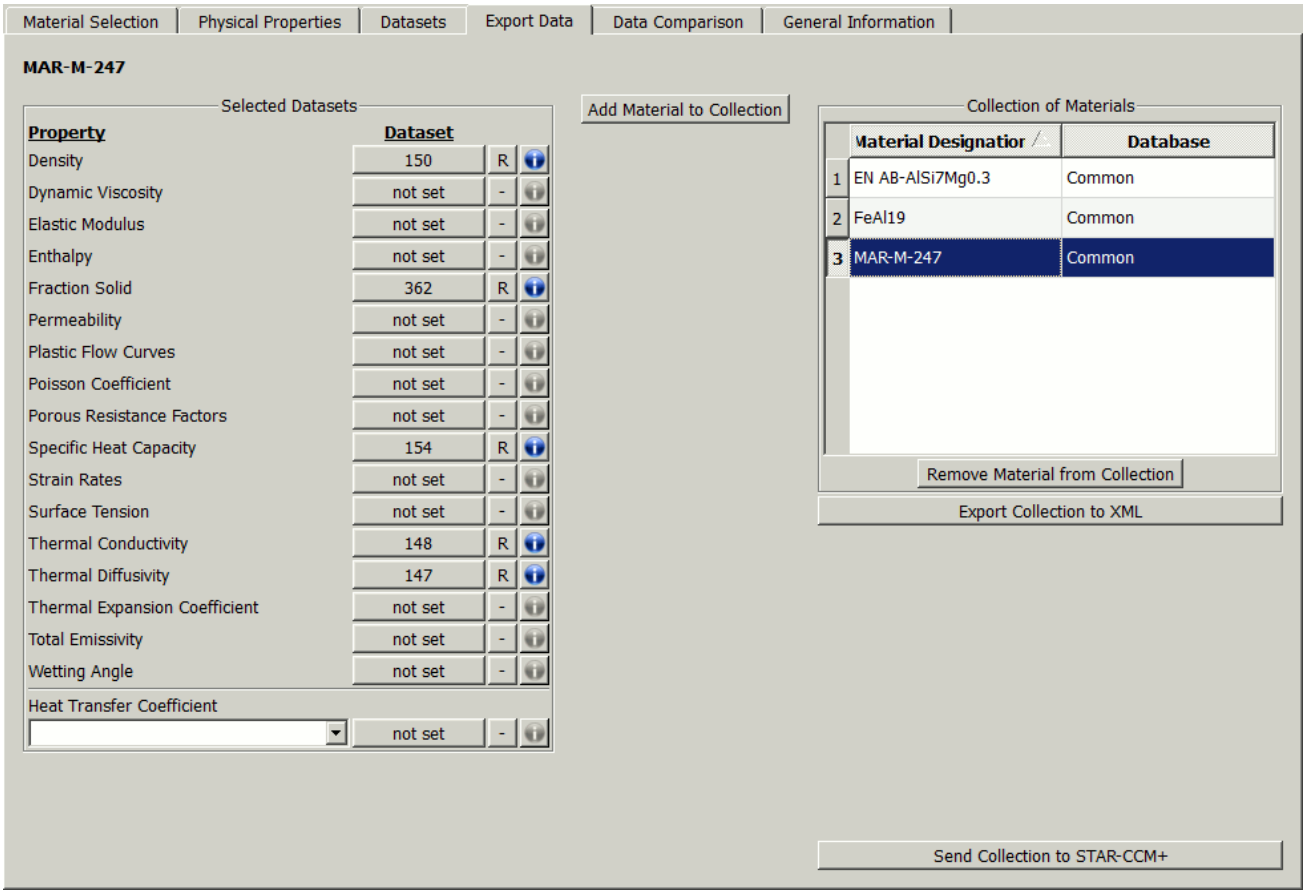
  The selected material is added to the Collection of Materials list.

To remove a material from the collection:

- In the Collection of Materials list, select the material designation that you want to remove.
- Click **Remove Material from Collection**.

Exporting the Collection of Materials

When you have prepared the appropriate collection of materials, you can export the materials and the selected physical properties from the Casting Material Database to STAR-CCM+.

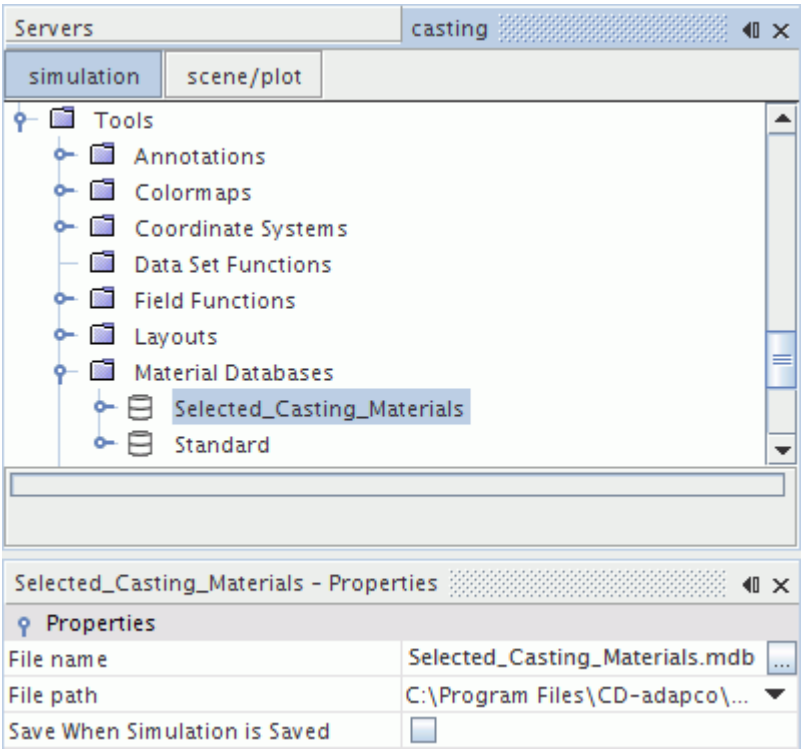


To export the collection of materials directly to STAR-CCM+:

- Click **Send Collection to** STAR-CCM+.

  The Casting Material Database Editor closes. The material collection is saved as an XML file with a predefined name (ExportToStarCCM_YYMMhhmmss.xml (Yr,Month,hr,min,sec)) under the current working directory.

  STAR-CCM+ automatically reads the exported material database and loads it under the **Tools** > **Material Databases** > **Selected_Casting_Materials** node.



To export the collection of materials to an .xml file that you can import later:

- Click **Export Collection to XML**.
- In the *Save File* dialog, specify the appropriate file name and location and then click **Save**.

  The Casting Material Database Editor remains open. You can import the exported file into the STAR-CCM+ Material Database later.

  See Using Imported Casting Materials and Properties in STAR-CCM+.

Working with Private Data

Within the Casting Material Database, a private material database allows you to work with confidential material data belonging to an individual owner. A private material database can be made available by a person in authority within your company.

If you are a person in authority within your company, you can create and edit private material data and datasets with the STAR-Cast Mat Database Builder. Private data are stored in the database file `MDB_Private.splite3` located in the `scmat/11.02.003/win64/database` subdirectory of your STAR-CCM+ installation directory. Before you start working on private material data, make sure that you have write permission on the database and the directory in which the database resides.

To work on private material data using the STAR-Cast Mat Database Builder:

1. In the Casting Material Database Editor, on the Tools menu, select **Private DB Editor**.

   The STAR-Cast Mat Database Builder launches in a separate window.
2. Create or edit the relevant material data and datasets.

For information on how to work with the STAR-Cast Mat Database Builder, see Working with the STAR-Cast Mat Database Builder.

Working with User-Defined Data

Within the Casting Material Database, a user-defined material database allows you to run casting simulations with your own material data.

You can create and edit user-defined material data and store them in the local database file `MDB_Scratch.sqlite3` within your home directory.

By default, common and private data is displayed, but user-defined data is hidden.

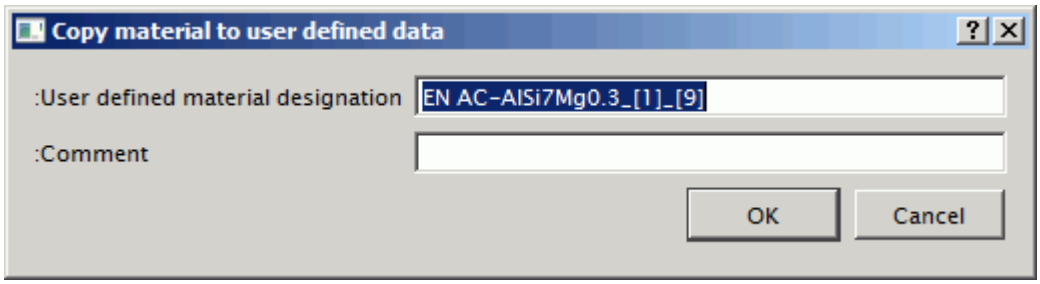
To display user-defined material data:

1. In the Casting Material Database Editor, on the Settings menu, activate **Show User Defined Data**.
2. In the *Material Selection* panel, launch the *Material Search* dialog by running an empty string search.

   All material designations, including user-defined ones, are displayed.

To create user-defined material data:

3. In the Casting Material Database Editor, select an existing material designation whose properties values are close to your user-defined properties.
4. In the *Material Selection* panel, click **Copy to User Defined Data**.
5. In the *Copy material to user-defined data* dialog, in the User-defined material designation box, enter the appropriate name.



   Alternatively, you can leave the default name composed of the original material designation from which the data have been copied and some postfix notation.
6. Optional: Enter a comment in the Comment box.
7. Click OK to confirm the new details.

   The user-defined material designation is added to the material subgroup that the original material designation belongs to.

To edit user-defined material data:

8. In the Casting Material Database Editor, select the user-defined material designation that you want to modify.
9. In the *Material Selection* panel, click **Edit User Defined Data**.

   The STAR-Cast Mat Database Builder launches in a separate window.

10. Edit the relevant material data and datasets.

    For information on how to work with the STAR-Cast Mat Database Builder, see Working with the STAR-Cast Mat Database Builder.

Working with the STAR-Cast Mat Database Builder

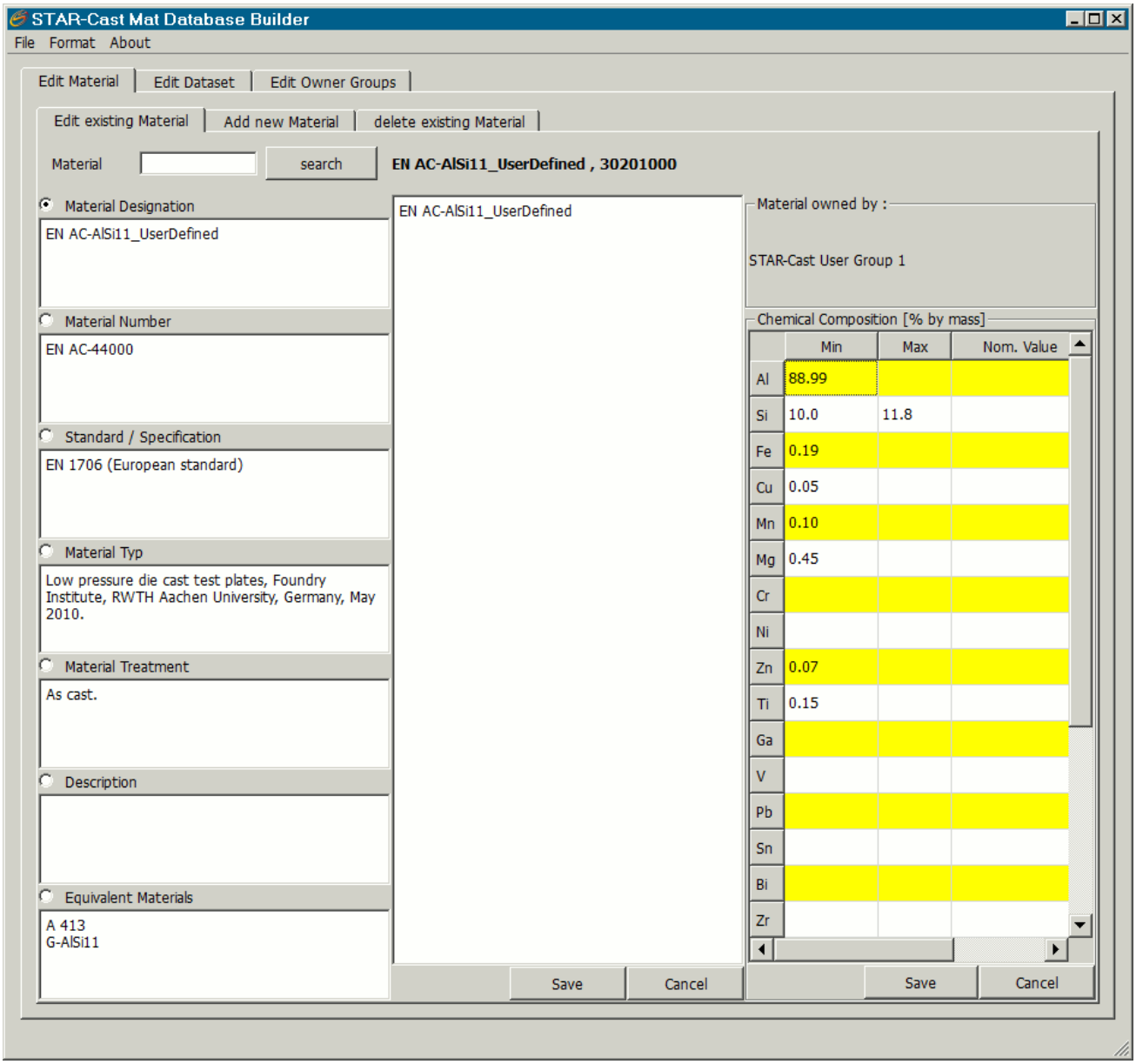
You can create and edit private and user-defined material data using the STAR-Cast Mat Database Builder.

There are two options to launch the STAR-Cast Mat Database Builder from the Database Editor, depending on the type of material data you want to work on:

- For user-defined material data, select a user-defined material designation (see Selecting a Casting Material) in the *Material Selection* panel and click **Edit User Defined Data**.
- For private material data, select **Tools** > **Private DB Editor**.

The Star-Cast Mat Database Builder allows you to create and edit:

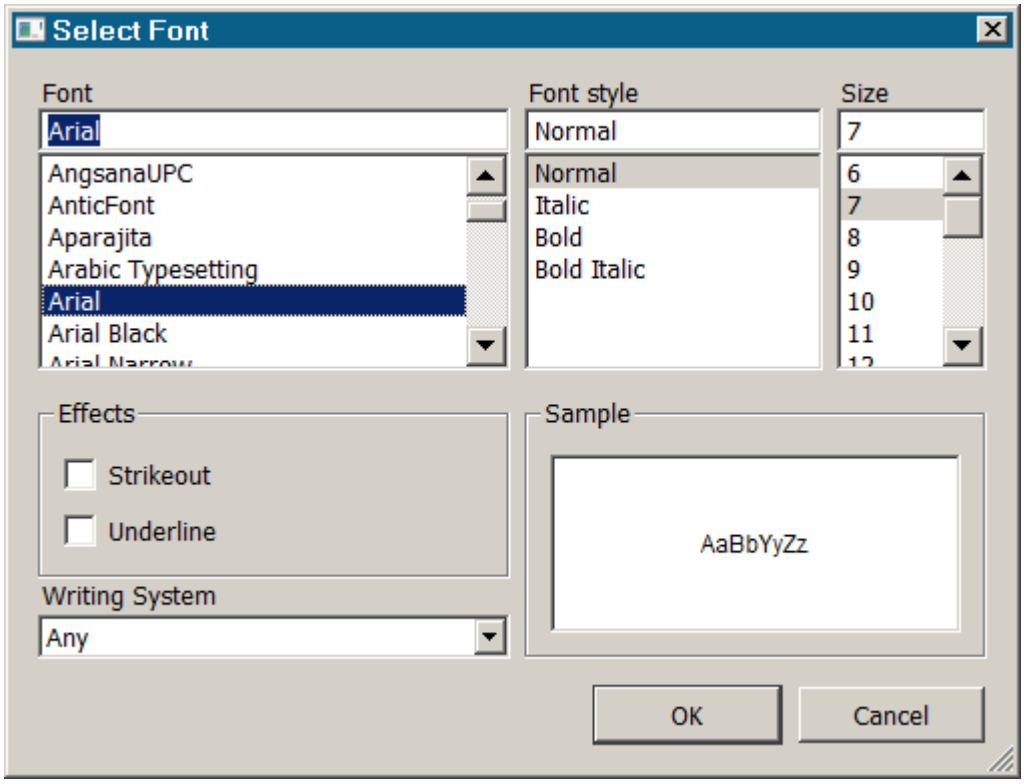
- Material data, see Creating and Editing Materials.
- Datasets, see Creating and Editing Datasets.
- Owner groups, see Creating and Editing Owner Groups.



You can control the appearance of the STAR-Cast Mat Database Builder to suit your requirements. You can resize the window and set the font, font style, and size.

To change the font:

1. Select **Format** > **Font**.
2. In the *Select Font* dialog, make the appropriate changes to the font settings.



3. To confirm the new settings, click **OK**.

Contents:

Creating and Editing Materials
Creating and Editing Datasets
Creating and Editing Owner Groups
STAR-Cast Mat Database Builder Reference

Creating and Editing Materials

You can work with private and user-defined material data in the *Edit Material* tab of the STAR-Cast Mat Database Builder.

Within this tab you can:

- Edit a material.
- Add a new material.
- Delete a material.

To edit an existing material:

1. Select the *Edit existing Material* subtab of the *Edit Material* tab.
2. In the **Material** textbox, enter the name of the material (or a related string) that you want to edit and click **search**.

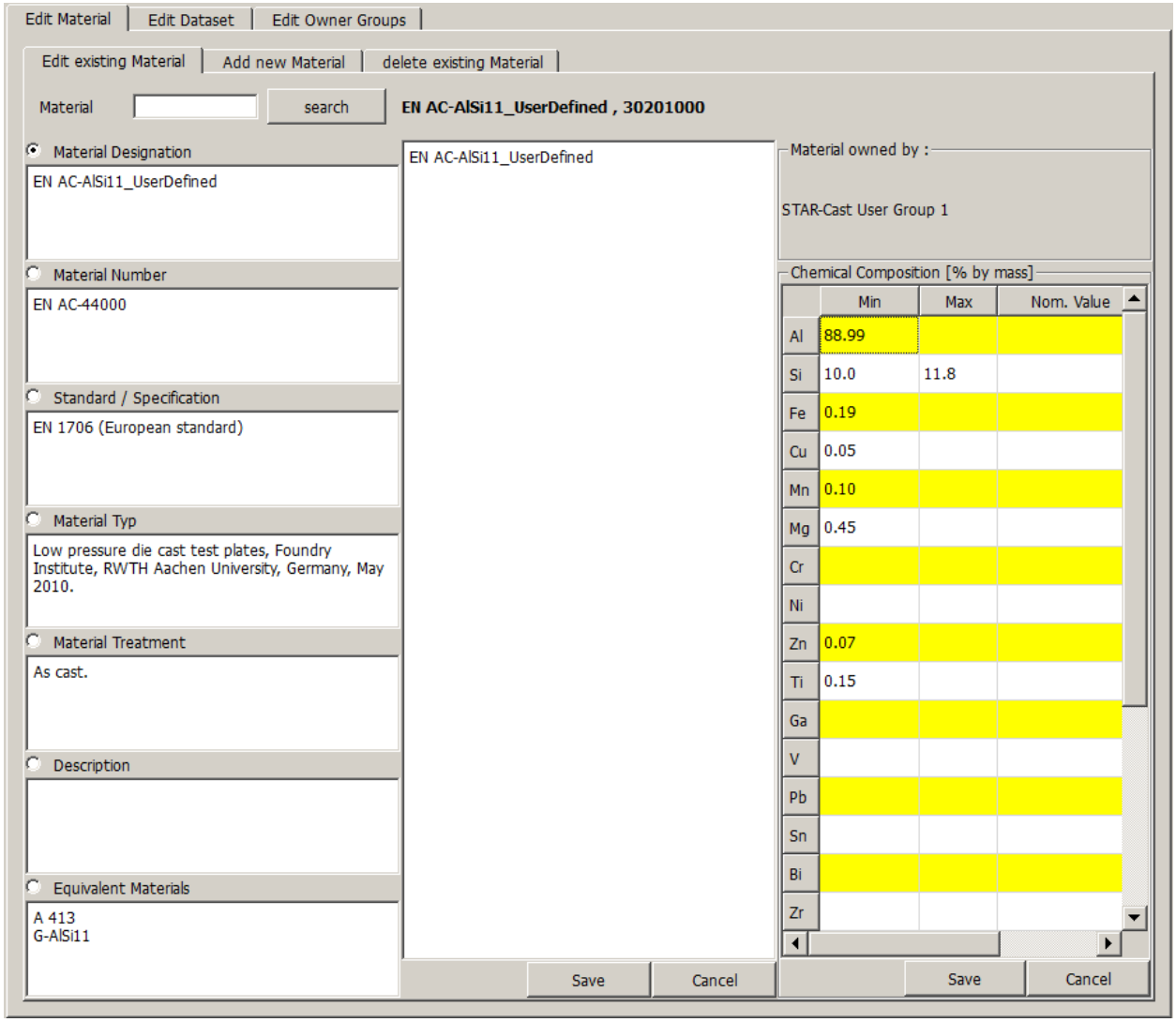
If you want to see the complete set of material designations that are available, leave the **Material** textbox empty.

3. In the **Material Search** dialog, choose the corresponding material designation and click **Select**.

   The material designation and the respective database ID are displayed.

4. To change one of the general information of the selected material on the left hand-side of the tab (see Edit Material Tab Reference), activate the relevant checkbox. Edit the information as required within the editing box in the center of the tab, and click **Save**.

5. To change the chemical composition of the material, select the relevant cell of the **Chemical Composition [% by mass]** table (see Edit Material Tab Reference), modify the value, and click **Save**.

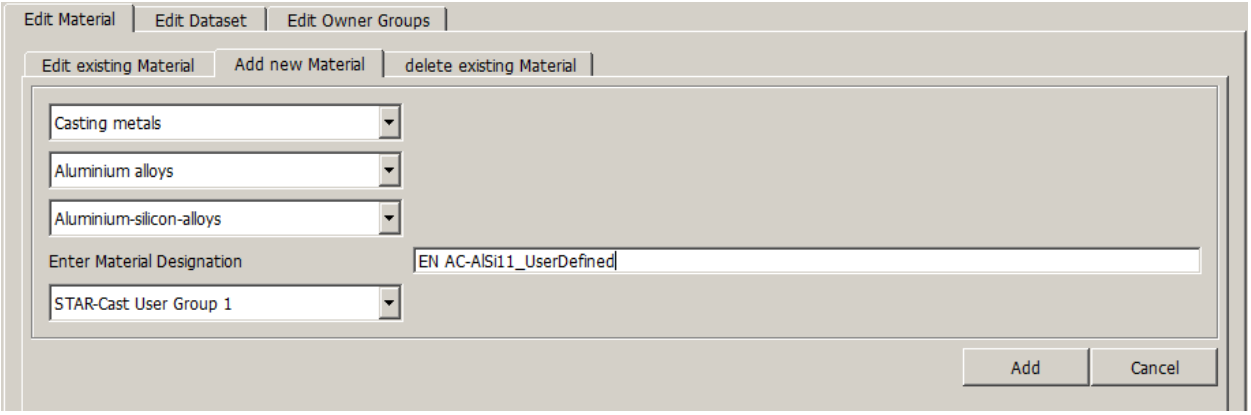


To add a new material designation:

6. Select the *Add new Material* subtab of the *Edit Material* tab.
7. Choose the material class from the **Select Material Class...** drop-down menu.
8. Choose the appropriate material group from the **Select Material Group...** drop-down menu.
9. Choose the appropriate material subgroup from the **Select Material Sub-Group...** drop-down menu.

10. Enter the designation of your material in the **Enter Material Designation** textbox.
11. Choose the owner of the material from the **Select Owner...** drop-down menu.

    Every new material that you add requires the declaration of a material owner. If no owner group is available, the STAR-Cast Mat Database Builder prompts you to create one first. For information on how to create an owner group, see Creating and Editing Owner Groups.



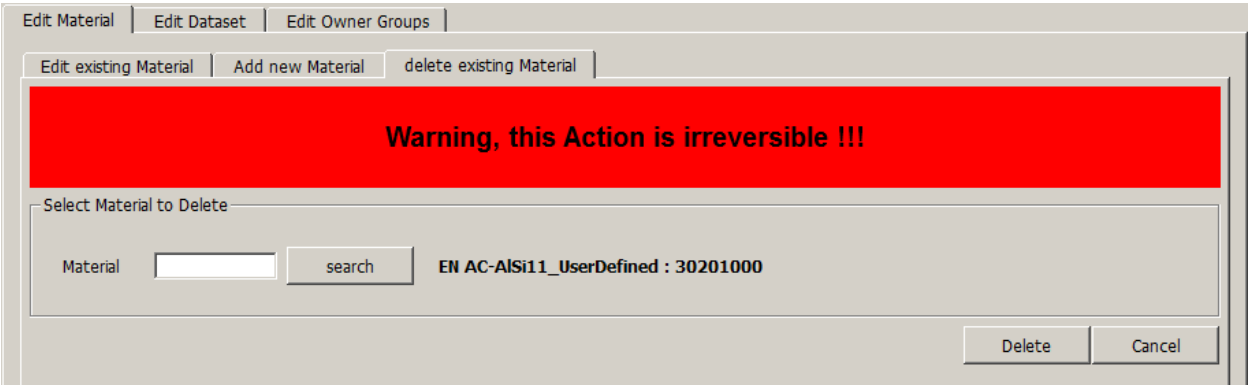
12. Click **Add** to save the new material designation to the database.

To delete a material:

13. Select the *delete existing Material* subtab of the *Edit Material* tab.
14. Within the **Select Material to Delete** group box, in the **Material** textbox, enter the name of the material (or a related string) that you want to delete and click **search**.

    If you want to see the complete set of material designations that are available, leave the **Material** textbox empty.
15. In the **Material Search** dialog, choose the corresponding material designation and click **Select**.

    The material designation and the respective database ID are displayed.



16. Click **Delete** to remove the material and all its associated datasets irreversibly from the database.

Creating and Editing Datasets

You can work with private and user-defined datasets in the *Edit Dataset* tab of the STAR-Cast Mat Database Builder.

Within this tab you can:

- Edit a dataset.
- Add a new dataset.
- Delete a dataset.
- Extrapolate, reduce, and smooth a dataset.

To edit an existing dataset:

1. Select the *Edit existing Dataset* subtab of the *Edit Dataset* tab.
2. In the **Dataset** textbox, enter the number of the dataset you want to edit and click **search**.
3. To change one of the general information of the selected dataset on the left hand-side of the tab (see Edit Dataset Panel Reference), activate the relevant checkbox, and do one of the following:

Option	Description
Edit information directly.	1. Edit information within the editing box in the center of the tab. 2. Click **Save**.
Copy information from another dataset within the same database.	1. Enter the relevant dataset number in the **copy selected from** text box. 2. Click **Copy**.

4. To mark the dataset as being recommended for the respective material property, click **Set** next to the **Recommended** option. The background color of the **Recommended** option switches from red to green.

   This option is only available for private material datasets.

   When **Set**, the background color of the **Recommended** option switches from red to green.
5. To provide source, reference, or specimen files for the dataset, do one of the following:

Option	Description
Add source or reference files.	1. Next to the **Source /Reference** option click **Set**. 2. In the **Select File** dialog, browse to the relevant file. The selected files are then copied to the `scmat/11.02.003/win64/resources/references` subdirectory of your STAR-CCM+ installation directory.
Add specimen files.	1. Next to the **Specimen** option click **Set**. 2. In the **Select File** dialog, browse to the relevant file. The selected files are then copied to the `scmat/11.02.003/win64/resources/specimen` subdirectory of your STAR-CCM+ installation directory.

These options are only available for private material datasets. You need to have write permission on the directories to add files.

When **Set**, the background color of the selected option switches from red to green.

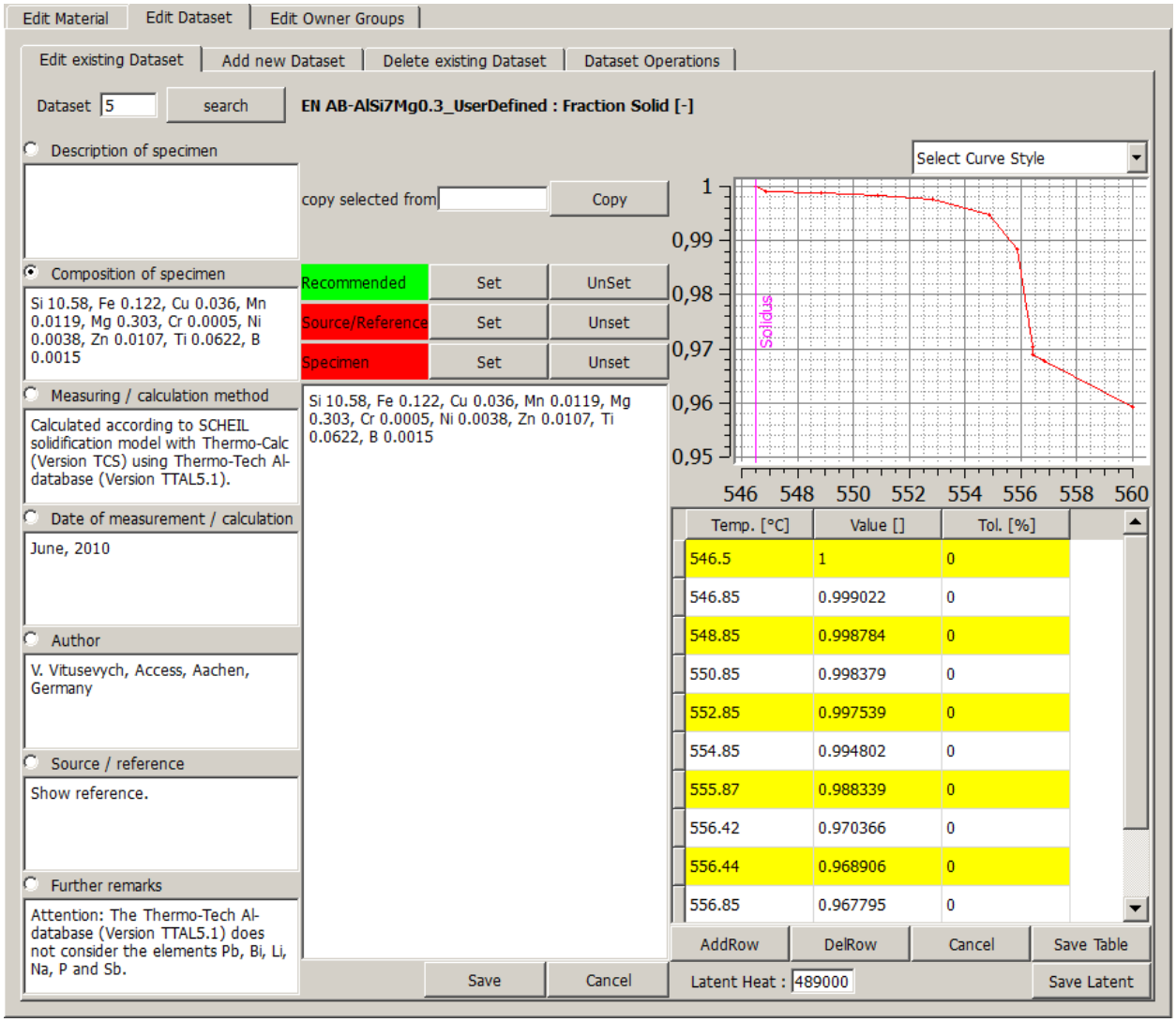
6. To modify the actual dataset, select the relevant cell within the dataset table (see Edit Dataset Panel Reference), modify the values as required, and click **Save Table**.

   Use the **AddRow** (add row) and **DelRow** (delete row) buttons to edit the table dimension.

   The database plot is updated accordingly.

7. If the physical property that owns the dataset is `Fraction Solid`, enter the value of latent heat in J/g within the **Latent Heat** textbox and click **Save Latent**.

8. To control the style of the dataset curve, choose the appropriate style (see Edit Dataset Panel Reference) from the **Select Curve Style** drop-down menu.



To add a new dataset to a specified material property:

9. Select the *Add new Dataset* subtab of the *Edit Dataset* tab.

10. Within the **Select Material** group box, in the **Material** textbox, enter the name of the material (or a related string) and click **search**.

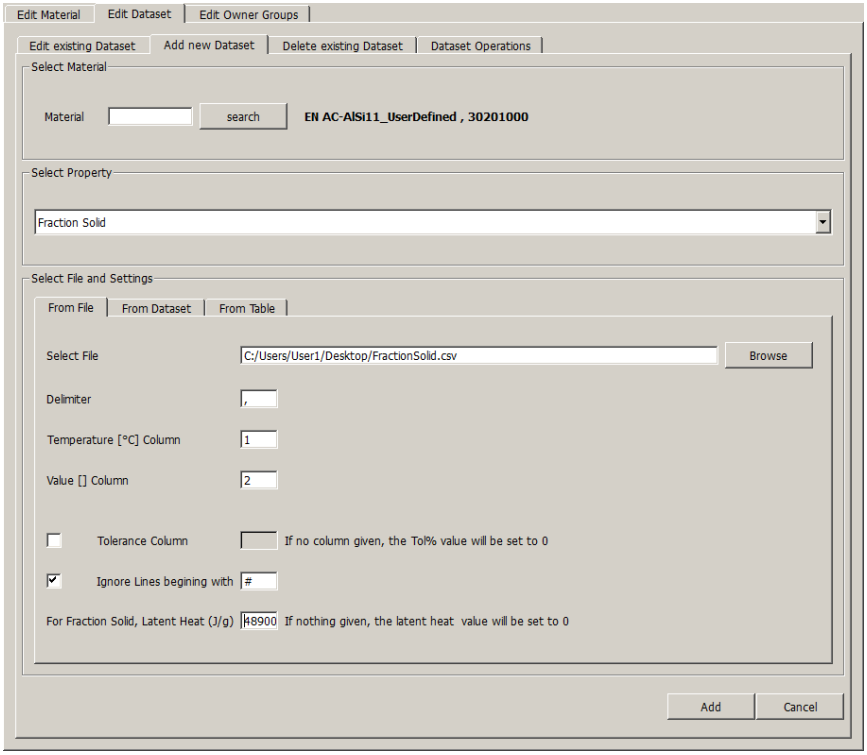
    If you want to see the complete set of material designations that are available, leave the **Material** textbox empty.

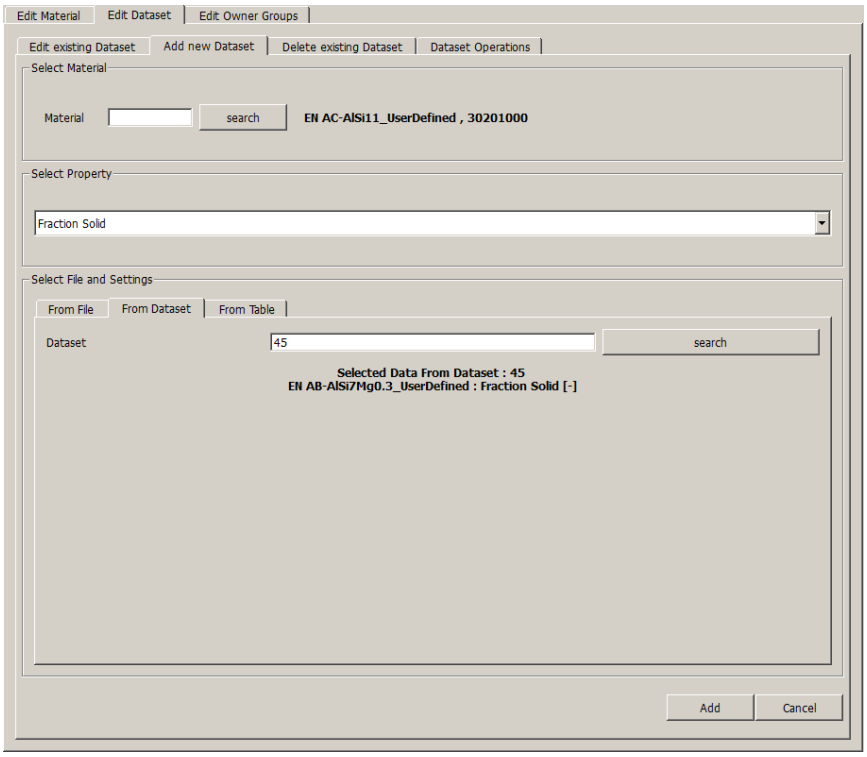
11. In the **Material Search** dialog, choose the corresponding material designation and click **Select**.

    The material designation and the respective database ID are displayed.

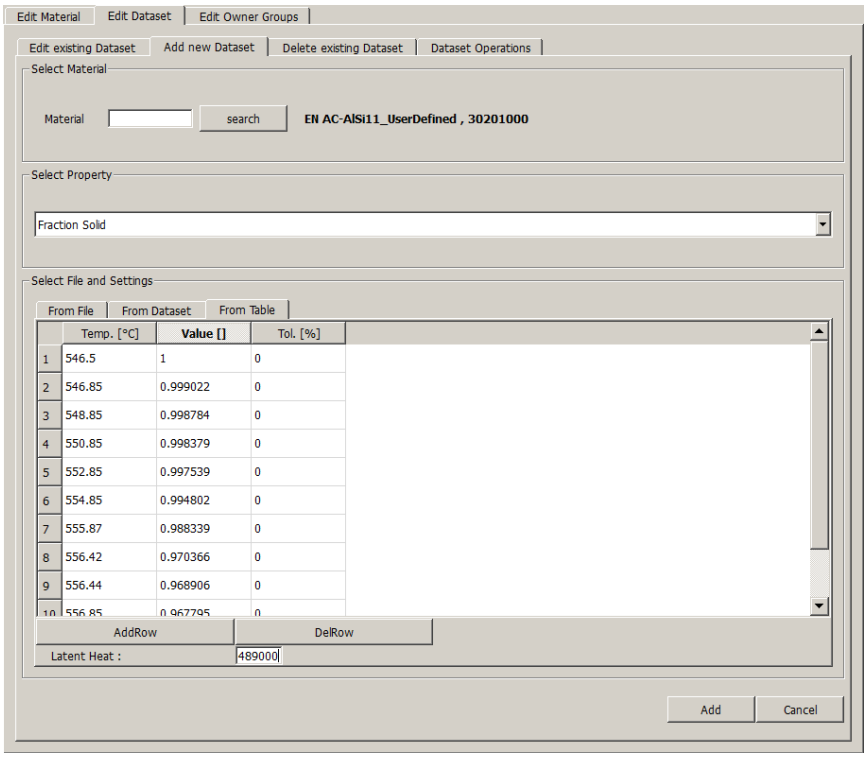
12. Within the **Select Property** group box, In the **Select...** drop-down list, select the appropriate physical property.

13. To specify the source of the new dataset, within the **Select File and Settings** group box, do one of the following:

Option	Description
Add a new dataset from a dataset input file.	1. Select the *From File* subtab. 2. Enter the required information, see Edit Dataset Panel Reference. 
Add a new dataset by duplicating the data from a specified dataset of the same database.	1. Select the *From Dataset* subtab. 2. In the **Dataset** textbox, enter the number of the respective dataset and click **search**. The material and physical property that own the specified dataset is displayed.



Create a new dataset table.	1. Select the *From Table* subtab. 2. Click the **AddRow** (add row) and **DelRow** (delete row) buttons to dimension the table. 3. Select the relevant cell within the dataset table, see Edit Dataset Panel Reference, and set the value as required.

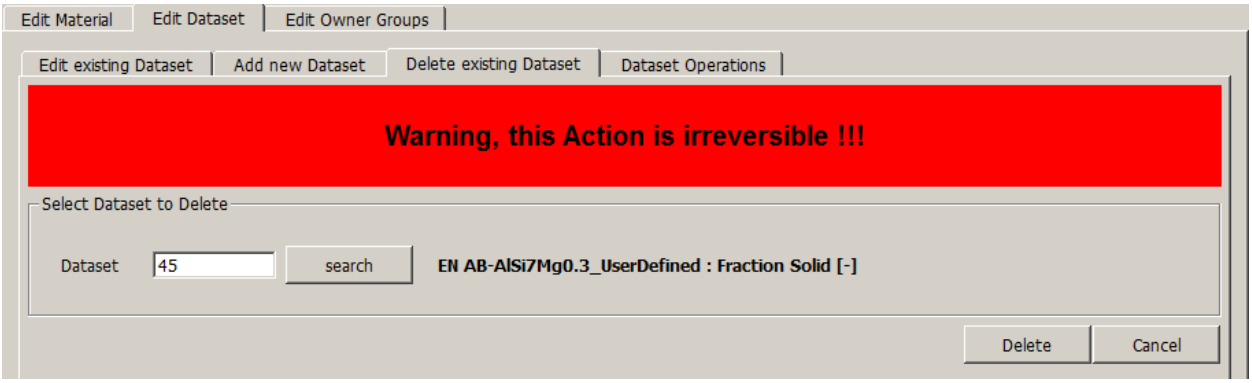


14. Click **Add** to save the new dataset to the database.

To delete an existing dataset:

15. Select the *Delete existing Dataset* subtab of the *Edit Dataset* tab.
16. Within the **Select Dataset to Delete** group box, in the **Dataset** textbox, enter the number of the dataset that you want to delete and click **search**.

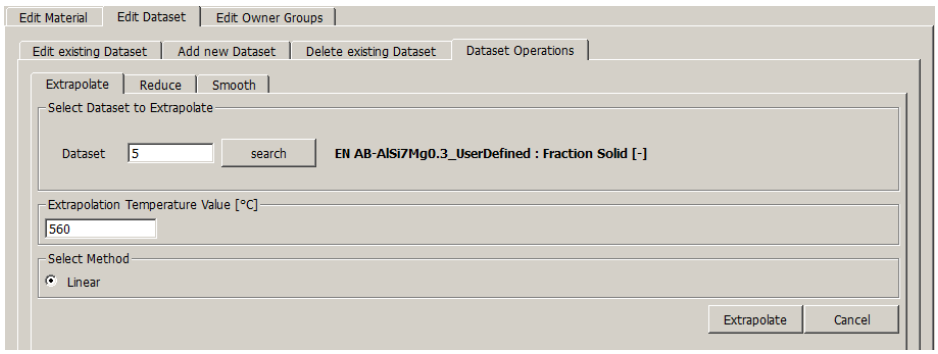
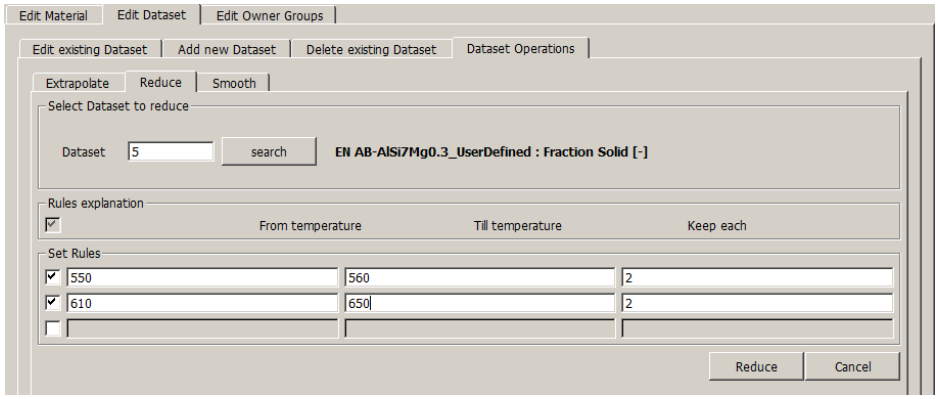
    The material and physical property that own the specified dataset is displayed.



17. Click **Delete** to remove the dataset irreversibly from the database.

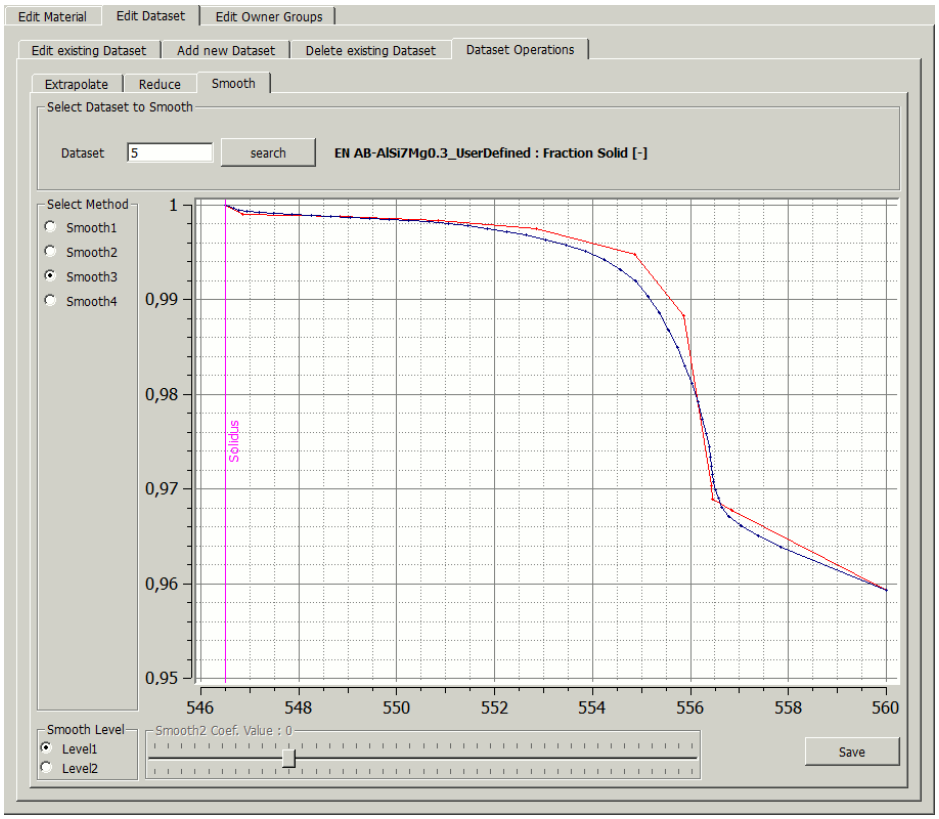
To operate on datasets:

18. Select the *Dataset Operations* subtab of the *Edit Dataset* tab and do one of the following:

Option	Description
Extrapolate a dataset to a temperature value outside its temperature range.	1. Select the *Extrapolate* subtab. 2. Within the **Select Dataset to Extrapolate** group box, in the **Dataset** textbox, enter the number of the dataset and click **search**. The material and physical property that own the dataset is displayed. 3. Within the **Extrapolation Temperature Value [°C]** group box, enter the required temperature value. 4. In the **Select Method** group box, activate the **Linear** extrapolation method.  5. Click **Extrapolate** to calculate the value at the specified temperature and add the data to the dataset.
Reduce the amount of data within a dataset.	1. Select the *Reduce* subtab. 2. Within the **Select Dataset to reduce** group box, in the **Dataset** textbox, enter the number of the dataset and click **search**. The material and physical property that own the dataset is displayed. 3. In the **Set Rules** group box, add up to three rules to define the data reduction. Each rule allows you to define a temperature interval in which the amount of data is reduced, see Edit Dataset Panel Reference.  4. Click **Reduce** to delete the redundant data and save the reduced dataset to the database.

Smooth the curve of a dataset.

1. Select the *Smooth* subtab.
2. Within the **Select Dataset to Smooth** group box, in the **Dataset** textbox, enter the number of the dataset and click **search**. The material and physical property that own the dataset is displayed and the dataset curve is plotted in the graphics box.
3. In the **Select Method** group box, choose the appropriate method that you want to use for smoothing the dataset, see Edit Dataset Panel Reference. The profile is updated to give you a preview of the smoothed dataset.
4. In the **Smooth Level** group box, choose the smoothing level as required, see Edit Dataset Panel Reference. The curve is updated accordingly.



5. Click **Save** to confirm the settings and save the smoothed dataset to the database.

Creating and Editing Owner Groups

You can work with owner groups for private and user-defined material data in the *Edit Owner Groups* tab of the STAR-Cast Mat Database Builder.

Each material within the STAR-Cast Mat Database Builder that is private or user-defined needs to be assigned to a specific owner group. Within this tab you can:

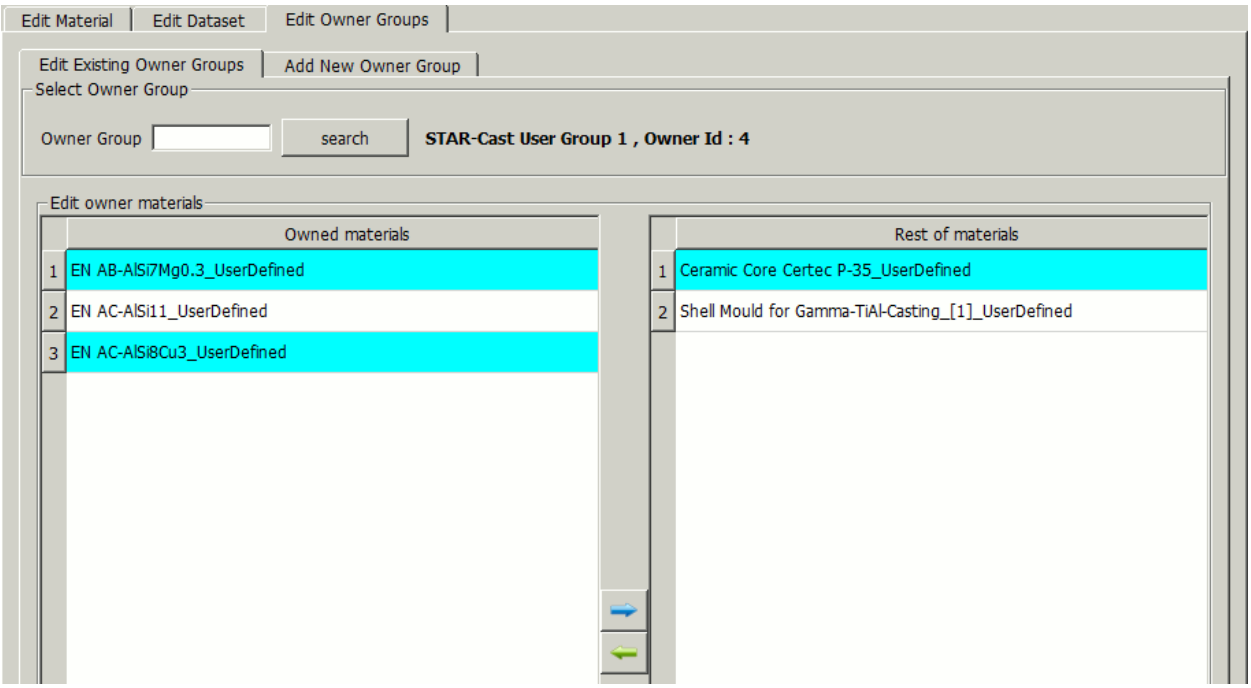
- Edit an owner group.
- Add a new owner group.

To edit the list of materials that are assigned to an owner group:

1. Select the *Edit Existing Owner Groups* subtab of the *Edit Owner Groups* tab.
2. Within the **Select Owner Group** group box, in the **Owner Group** textbox, enter the name of the owner group (or a related string) you want to edit and click **search**.

   If you want to see the complete set of owners that are available, leave the **Owner Group** textbox empty.
3. In the **Owner Search** dialog, choose the corresponding owner designation and click **Select**.

   The materials that are assigned to the owner are displayed in the **Owned materials** box.



4. To modify the list of owner materials, do one of the following:

Option	Description
Remove owned materials.	Select the relevant material designation in the **Owned materials** list and click the blue arrow button.
Add materials.	Select the relevant material designation in the **Rest of materials** list and click the green arrow button.

   The material is transferred to the respective target box.

To add a new owner group:

5. Select the *Add New Owner Group* subtab of the *Edit Owner Groups* tab.
6. Enter the designation of the owner group in the **Enter Owner Group Name** textbox and click **Add**.

The new owner is added to the database.

STAR-Cast Mat Database Builder Reference

The STAR-CAST Material Database Builder allows you to control material data, datasets, and owner groups of your private or user-defined material database. The following sections give you an overview of the corresponding tabs within the database builder and their properties.

Edit Material Tab

Edit existing Material	Subtab to edit private or user-defined material data.	
	Material	Search for material to edit.
	Material Designation	Designation of the material.
	Material Number	Number of the material.
	Standard/ Specification	Standard of the material (for example, European, American, British) or a company-defined specification.
	Material Type	Nature of the material (as applicable) to which the Standard/Specification refers, or the specimens to which the measurements correspond. For example: castings, alloyed ingots for remelting, and alumina-based shell.
	Material Treatment	Describes how the material has been processed. For example: heat-treated and as cast condition.
	Description	The composition or specification of the material (as applicable) and other information about the composition and manufacturing or special properties. For example, porosity.
	Equivalent Materials	A list of equivalent materials. For example, alloys with similar composition or specifications, and aliases for the material designation in the Casting Material Database dialog.
	Material owned by	Name of the material owner group. For example: STAR-Cast User 1 For information on how to add and edit owner groups, see Creating and Editing Owner Groups.

	Chemical Composition [% by mass]	Table that lists the chemical composition of the material in % by mass.
		Min Minimum percentage of the element in the chemical composition by mass. **Max** Maximum percentage of the element on the chemical composition by mass. **Nom. Value** Nominal value of the element in the chemical composition.
Add new Material	Subtab to add new private or user-defined material data.	
	Select Material Class...	Choice of material class for the new material.
	Select Material Group...	Choice of material group for the new material.
	Select Material Sub-Group...	Choice of material subgroup for the new material.
	Enter Material Designation	Designation for the new material.
	Select Owner...	Choice of owner for the material from the *Select Owner...* drop down menu. Every new material that you add requires the declaration of a material owner. If no owner group is available, the STAR-Cast Mat Database Builder prompts you to create one first. For information on how to create an owner group, see Creating and Editing Owner Groups.
delete existing Material	Subtab to delete private or user-defined material data.	
	Select Material to Delete	Controls the material that you want to delete.
		Material Search for the relevant material.

Edit Dataset Panel

Edit existing Dataset	Subtab to edit private or user-defined material datasets.	
	Description of specimen	Description of the sample of material for examination. For example: specimen made out of sand cast test bars after standard Na melt treatment of ingot aluminium.

	Composition of specimen	Chemical composition of specimen in % by mass. For example: Wt. %: Si 9.419, Fe 0.252, and so on.
	Measuring / calculation method	Description of the measuring or calculation method. For example: the density as a function of temperature has been calculated using the density at room temperature and the volume dilatation.
	Date of measurement / calculation	Date when the dataset was measured or calculated. For example: January, 2016
	Author	Name of the person or institute that measured or calculated the data.
	Source / reference	The composition or specification of the material (as applicable) and other information about the composition and manufacturing or special properties. For example: measurement report.
	Further remarks	Any additional information that is worth being mentioned. For example: the first and the last point have been linearly extrapolated from 21°C to 0°C and from 700°C to 800°C.
	copy selected from	Search for dataset within the same database whose selected information you want to duplicate.
	Recommended	When **Set**, the dataset is marked as recommended for the respective physical property.
	Source / Reference	When **Set**, you can specify the path to source or reference files. The files are then copied to the `scmat/11.02.003/win64/resources/references` subdirectory of your STAR-CCM+ installation directory.
	Specimen	When **Set**, you can specify the path to specimen files. The files are then copied to the `scmat/11.02.003/win64/resources/specimen` subdirectory of your STAR-CCM+ installation directory.
	Select Curve Style	Controls the style of the dataset curve.
		Lines Displays the dataset as a lined curve. **Dots** Displays the dataset as dots.
	Temp. [°C]	Temperature of the point of measurement or calculation.
	Value []	Value of the physical property at the respective temperature.
	Tol. [%]	Tolerance of the measured or calculated property at the respective temperature.
	Latent Heat	Value of latent heat in J/g (only if the physical property is `Fraction Solid`).

Add new Dataset	Subtab to add new private or user-defined material datasets.	
	Select Material	Controls the material whose physical property you want to describe by a new dataset.
		Material Search for the relevant material.
	Select Property	Controls the physical property that you want to describe by a new dataset.
	Select File and Settings	Controls the source of the new dataset.
		From File Subtab to load a new dataset from a specified input file, which contains tabular values. • **Select File**: Database input file in ASCII format, such as .csv. • **Delimiter**: Delimiter that is used to separate columns in the database input file. • **Temperature [°C] Column**: Temperature column number. • **Value [] Column**: Physical property column number. • **Tolerance Column**: Tolerance column number. If no column is given, the Tol% value is set to 0. • **Ignore Lines begining with**: Character that indicates the beginning of a comment in the database input file. Activate this option and then specify the appropriate comment character in the text box. • **For Fraction Solid, Latent Heat (J/g)**: Value of latent heat in J/g (only for `Fraction Solid` as selected property). *From Dataset* Subtab to create a dataset that duplicates a dataset within the same database. • **Dataset**: Search for the dataset to duplicate. *From Table* Subtab to create a new dataset table. • **Temp. [°C]**: Temperature of the point of measurement or calculation. • **Value []**: Value of the physical property at the respective temperature. • **Tol. [%]**: Tolerance of the measured or calculated property value at the respective temperature. • **Latent Heat**: Value of latent heat in J/g (only for `Fraction Solid` as selected property).
Delete Existing Dataset	Subtab to delete private or user-defined material datasets.	
	Select Dataset to Delete	Controls the datasets that you want to delete.

		Dataset Search for the relevant dataset.
Dataset Operations	Subtab to operate on private or user-defined material datasets.	
	Extrapolate	Subtab to extrapolate a dataset to a temperature value outside its temperature range.
		Select Dataset to Extrapolate Controls which dataset you want to extrapolate. • **Dataset**: Search for the relevant dataset. **Extrapolation Temperature Value [°C]** Temperature value outside the temperature range of the dataset to extrapolate the physical property to. **Select Method** Controls which method to use for extrapolation of the selected physical property. • **Linear**: Linear extrapolation to the specified temperature value.
	Reduce	Subtab to reduce the amount of data within a dataset.
		Select Dataset to reduce Controls which dataset you want to reduce. • **Dataset**: Search for the relevant dataset. **Rules explanation** Controls the rules of the dataset reduction. • **From temperature**: Start of temperature interval. • **Till temperature**: End of temperature interval. • **Keep each**: Frequency of data to keep within the dataset. For example: 5 means to keep every fifth value within the interval. **Set Rules** Controls up to three rules to define the data reduction. Each rule allows you to define a temperature interval in which the amount of data is reduced.
	Smooth	Subtab to smooth the curve of a dataset.
		Select Dataset to Smooth Controls the dataset that you want to smooth. • **Dataset**: Search for the relevant dataset.

		Select Method Controls the method that you want to use for smoothing the dataset. • **Smooth1**: Use Bézier curve smoothing. • **Smooth2**: Adapt position of the middle control points with the **Smooth2 Coef. Value** followed by Bézier curve smoothing. • **Smooth3**: Use moving average followed by Bézier curve smoothing. • **Smooth4**: Use average of neighboring points for middle control points followed by Bézier curve smoothing. **Smooth Level** Controls the level of smoothing. • **Level1**: Use quadratic Bézier curve. • **Level2**: Use cubic Bézier curve. **Smooth2 Coef. Value** Value to control the middle point for **Smooth2**.

Edit Owner Groups Panel

Edit Existing Owner Groups	Subtab to edit existing owner groups for private or user-defined material data.	
	Select Owner Group	Search for owner group whose corresponding material designations you want to edit.
		Owner Group Search for owner group to edit.
	Edit owner materials	Controls the material designations that belong to the specified owner group.
		Owned materials Material designations that belong to the specified owner group. **Rest of materials** Rest of material designations within the same database that do not belong to the specified owner group.
Add New Owner Group	Subtab to add new owner groups for private or user-defined material data.	
	Enter Owner Group Name	Designation of new owner group.

Using Imported Casting Materials and Properties in STAR-CCM+

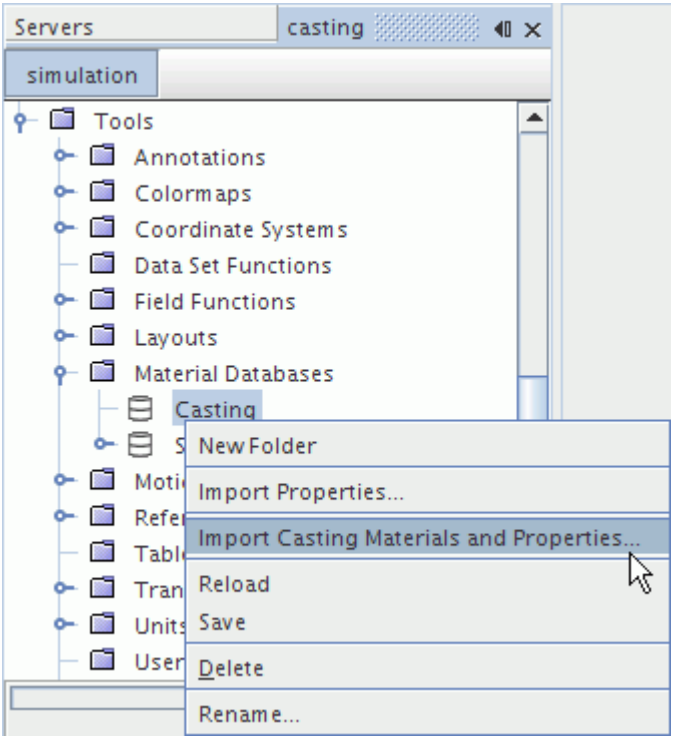
You can import casting materials and properties into the STAR-CCM+ Material Database. You can then use the materials in a physics continuum, and use the material properties as boundary values.

Importing Casting Materials into the STAR-CCM+ Material Database

You import the casting material data from an XML file that is created in the Casting Material Database Editor.

For more information, see Exporting Casting Material Data to STAR-CCM+.

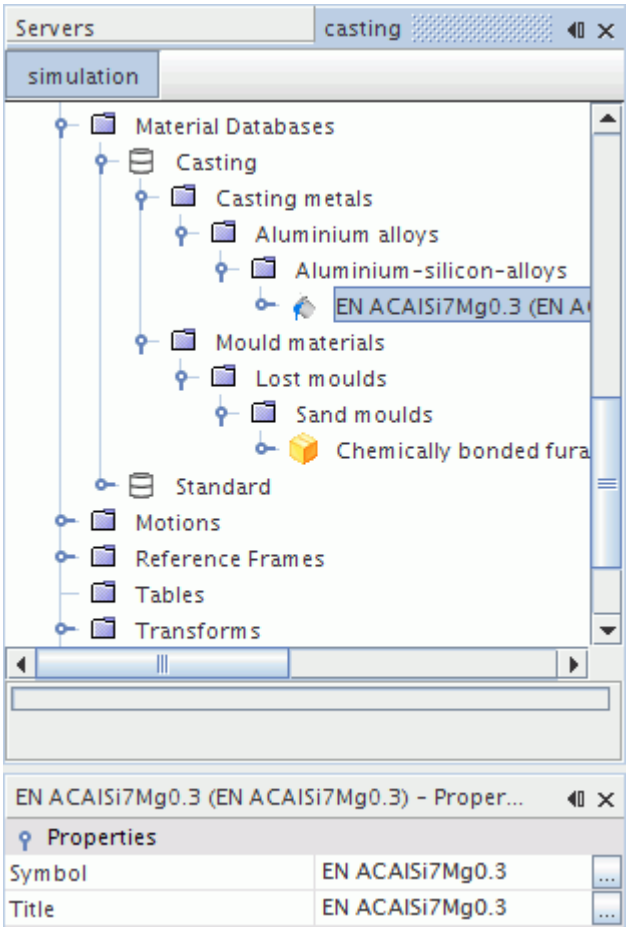
1. If necessary, create a new Material Database in STAR-CCM+.
2. Right-click the material database to which you want to add casting materials, and select **Import Casting Materials and Properties**.



3. In the *Open* dialog, select the appropriate XML file.

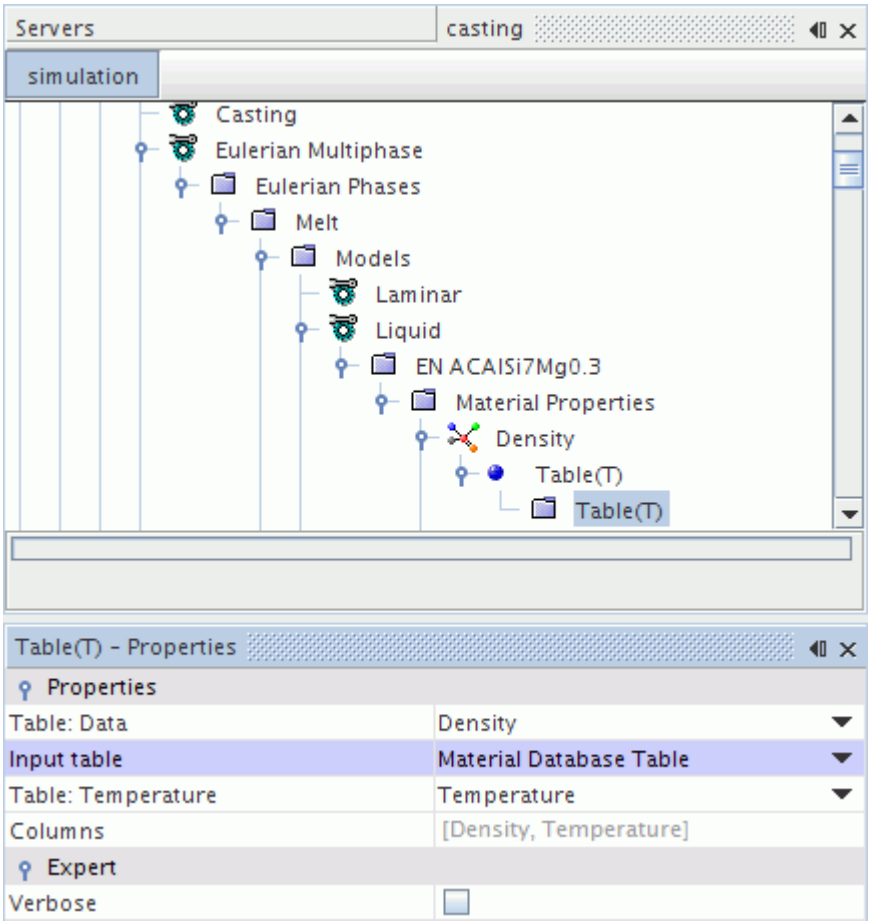
The casting materials are imported into the material database and appear under a tree structure representing the material hierarchy of the Casting Material Database Editor.

For example, a material designation for an aluminum silicon alloy appears under **Casting metals** > **Aluminium alloys** > **Aluminium-silicon-alloys**.



Using Imported Physical Properties in Physics Continua

Once the selected casting materials are imported into STAR-CCM+, they can be selected as single components, phases or within multi-component mixtures in the Physics Continua.



When the **Table(T)** method is chosen for an imported physical property, the *Input Table* property defaults to Material Database Table provided the imported material has tabulated data in the STAR-CCM+ material database. Otherwise it is set to None.

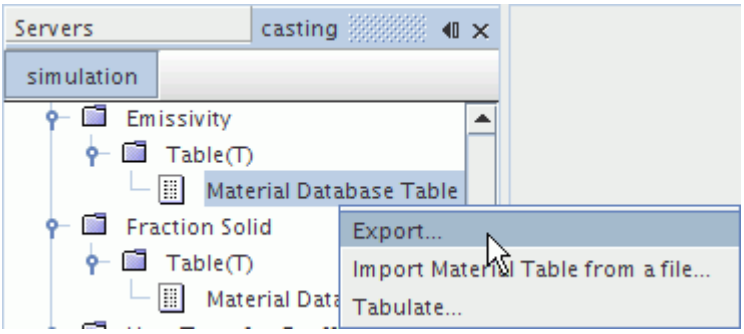
Using Boundary Values Exported from the Material Database Editor

Some material properties from the Casting Material Database Editor are defined as boundary values by STAR-CCM+. These properties include:

- Emissivity
- Heat Transfer Coefficient
- Wetting Angle (Contact Angle)

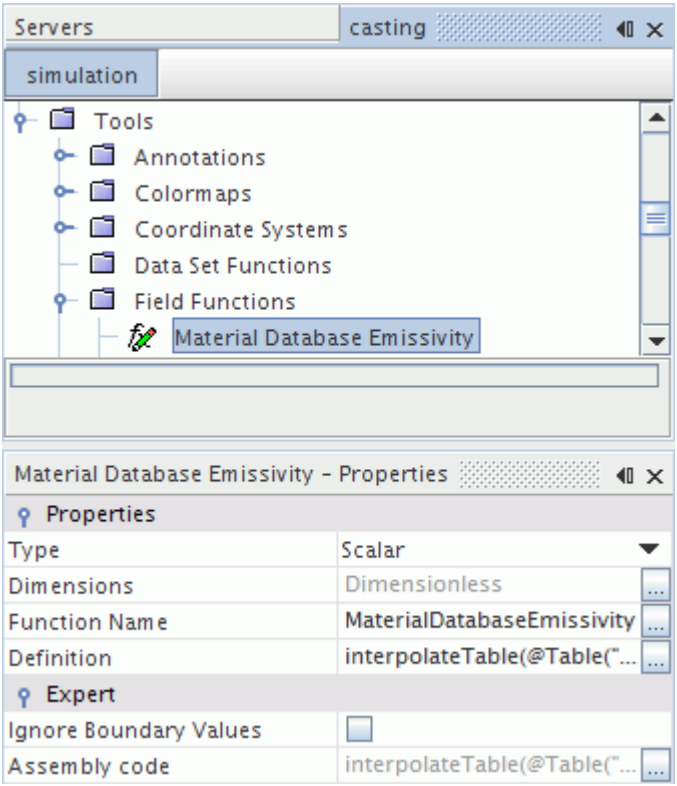
To use these properties in STAR-CCM+:

1. Export the material data to a `.csv` file.

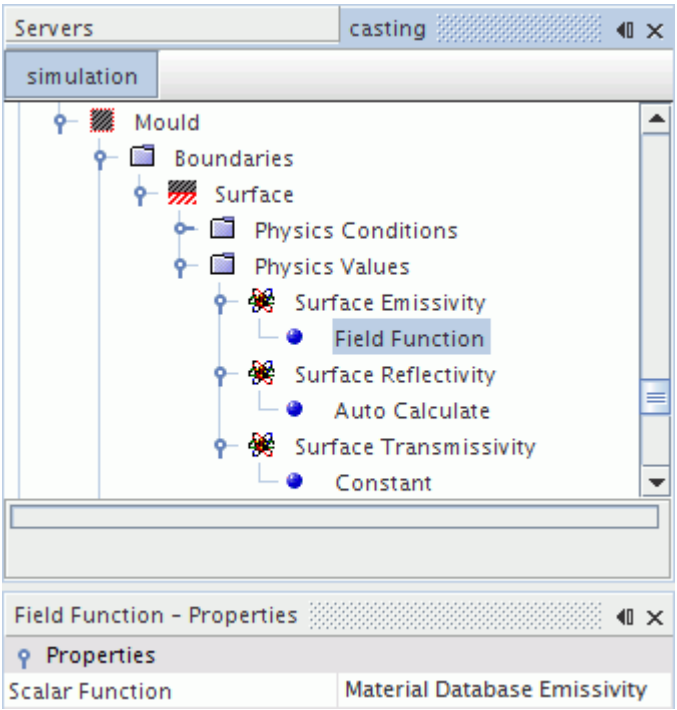


2. Create a table using the exported `.csv` file.
3. Create a field function and use the table interpolation syntax to generate a scalar expression for the material property.

   For casting properties, the <scalar_expression> is $Temperature.



4. Use this field function to specify the boundary value profile.


Servers
casting
simulation
Mould
Boundaries
Surface
Physics Conditions
Physics Values
Surface Emissivity
Field Function
Surface Reflectivity
Auto Calculate
Surface Transmissivity
Constant
Field Function - Properties
Properties
Scalar Function
Material Database Emissivity

Detecting Shrinkage-Related Defects

You can use the Macro Porosity model to detect shrinkage-related defects in a cast part. When isolated liquid melt regions occur during the solidification process of a cast part, shrinkage cannot be compensated for by supplying extra melt, and pores form.

The Macro Porosity model is available in two versions:

- **Macro Porosity (fully coupled)**

  Used in a full VOF simulation that models flow, pressure, and thermal effects.

  This model uses an auxiliary void phase to model the shrinkage-related cavities. The cavities can be vacuum bubbles, or they can be filled with gas that is dissolved in the liquid phase (hydrogen, for example). In either situation, you specify the appropriate void pressure. The shrinkage is compensated for through the gaseous void phase such that the liquid mass remains constant.
- **Macro Porosity (pure thermal)**

  Used with a pure thermal simulation to model only the solidification process of a liquid with temperature-dependent density. The solution of the flow equations is deactivated and only thermal effects are modeled.

  This model applies only when the solidifying melt is fully enclosed by either wall boundaries or by flow-stopped cells. The shrinkage or expansion is compensated for by phase replacement in the appropriate cells to ensure that the mass is conserved.

  - To model shrinkage, the liquid phase is replaced with void phase.
  - To model expansion, the void phase is replaced with liquid phase.

    If no void phase is available in the liquid zone cells, expansion cannot be compensated for, and a mass conservation error can result.

Note: Simulations that use parallel processing can have minor mass distribution errors. These errors are introduced by the multiple partitions of the liquid zone: the phase replacement at each time-step is performed entirely in one partition (when the amount of melt that is replaced on the partition is sufficient to compensate for the shrinkage of the liquid zone).

Phase replacement is performed in the partition that contains the cell with the global maximum value of pore probability. If two or more partitions contain cells with the same maximum value, the cell that has the lowest volume fraction of void determines the partition in which the phase replacement is performed. If cells with the same maximum value of pore probability and the same volume fraction of void exist in two or more partitions, the partition with the lowest index is chosen for phase replacement. In these cases, the phase replacement is performed in a single partition rather than being spread evenly across all of the appropriate partitions. The melt mass is conserved perfectly in every case, but the mass distribution can be uneven in parallel simulations.

The two Macro Porosity models are alternatives; you cannot use both in the same simulation.

To set up shrinkage-related defect detection:

Prerequisite: a VOF Multiphase simulation with the appropriate models selected. An Energy model must be selected. The **Macro Porosity (pure thermal)** model also requires the **Pure Thermal** model selected.

See Casting Workflow and Macro Porosity Model Reference.

1. In the simulation, under the **Eulerian Multiphase** > **Eulerian Phases** node, create a gas phase for the pores/void.

   The gas phase must be a single-component gas. For the **Macro Porosity (fully coupled)** model only, the gas must be compressible. The **Macro Porosity (pure thermal)** model has no constraints on the equation of state of the void phase.

   See Selecting Phase Models.
2. Set the material properties of the gaseous void phase and ensure that the following requirements are met:
   - $\mathrm{d}\,\rho / \mathrm{d}\,p$ must be larger than zero.
   - This void phase must have a much lower density than the liquid phase.

     As the ratio of liquid to void phase density gets closer to 1, more fluid is required to be converted from liquid to void phase to achieve a certain pressure increase in a cell.
3. Define the macro porosity model:
   a. Create a phase interaction and activate the **VOF-VOF Phase Interaction** model.
   b. Select the **VOF-VOF Phase Interaction** node and set the *Primary Phase* to the liquid melt phase and set the *Secondary Phase* to the gaseous void phase.
   c. Reopen the *Phase Interaction Model Selection* dialog and, in the *Optional Models* group box, select the appropriate model: **Macro Porosity (fully coupled)** or **Macro Porosity (pure thermal)**.
   d. To set the macro porosity properties, edit the [*Phase Interaction*] node:
      - For the **Macro Porosity (fully coupled)** model, set the *Void Pressure* and *Scaling Factor*.

        See Macro Porosity (fully coupled) Model Properties.
      - For the **Macro Porosity (pure thermal)** model, set the **Solid Cell Indicator** and **Pore Probability Indicator**.

        Pores can develop only in the liquid melt, not the solidified melt. The **Solid Cell Indicator** gives information about which cells are filled with liquid and which are filled with solid. The **Pore Probability Indicator** tells you how likely it is that pores form.

        See Macro Porosity (pure thermal) Child Nodes.

Post-Processing

The Macro Porosity (pure thermal) model keeps the mass of the primary phase constant. However, if you create a report accumulating the mass, the report shows an error in mass conservation at the end of each time-step. This error is because the mass is corrected before the first iteration of each time-step. The temperature change during the time-step can alter the density, and therefore change the phase mass within the time-step. To check the result of phase replacement directly, you need to run the simulation for one time-step with the energy solver frozen. This keeps the temperature field unchanged from the time when the phase replacement is performed until the end of the time-step when the report is evaluated.

Contents:

Macro Porosity Model Reference
Variation of the Scaling Factor

Macro Porosity Formulation

Macro Porosity Model Reference

The Macro Porosity model is used to detect shrinkage-related defects in a solidifying liquid with temperature-dependent density. The Macro Porosity (fully coupled) model is used in a full VOF simulation to model flow, pressure, and thermal effects. The Macro Porosity (pure thermal) model is used with a pure thermal simulation to model only the solidification process of the melt.

Macro Porosity Model Reference

Model Names	**Macro Porosity (fully coupled)**	
	Macro Porosity (pure thermal)	
Theory	See Macro Porosity Formulation. Applies to the Macro Porosity (fully coupled) model only.	
Provided By	**[physics continuum]** > **Models** > **Multiphase Interaction** > **Phase Interactions** > **[phase interaction]** > **Models** > *Optional Models*	
Example Node Path	**[physics continuum]** > **Models** > **Multiphase Interaction** > **Phase Interactions** > **[phase interaction]** > **Models** > **Macro Porosity (fully coupled)**	
Requires	A VOF Multiphase simulation with the following models selected: *Space*: **Axisymmetric**, **Two Dimensional**, or **Three Dimensional** *Time*: **Implicit Unsteady** *Material*: **Eulerian Multiphase** *Eulerian Multiphase Model*: **Volume of Fluid (VOF)** *Energy*: **Segregated Fluid Isothermal** or **Segregated Multiphase Temperature** *Optional Models*: **Pure Thermal** (required for the Macro Porosity (pure thermal) model only) A minimum of two VOF phases: a liquid phase (for the melt) and a gas phase (for the pores/void). The liquid phase must be a single-component liquid and have a temperature-dependent density, such as **User Defined EOS** or **Polynomial Density**. The gas phase must be a single-component gas. For the Macro Porosity (fully coupled) model only, the gas must be compressible. A phase interaction with the following setup: The VOF-VOF Phase Interaction model activated, with the Primary Phase set to the liquid phase and the Secondary Phase set to the gas phase. *Phase Interaction Topology*: **VOF-VOF Phase Interaction** *Optional Models*: **Macro Porosity (fully coupled)** or **Macro Porosity (pure thermal)**	
Properties	See Macro Porosity (pure thermal) Model Properties and Macro Porosity (fully coupled) Model Properties.	
Activates	Model Controls (child nodes)	For the Macro Porosity (pure thermal) model: **Solid Cell Indicator** **Pore Probability Indicator** See Macro Porosity (pure thermal) Child Nodes.

	Field Functions	See Field Functions.

Macro Porosity (pure thermal) Model Properties

The Macro Porosity (pure thermal) model has the following property:

Verbose

When activated, writes an output before the first iteration of each time-step. This output details the amount of the primary phase (melt) that is replaced by the secondary phase (void).

Macro Porosity (fully coupled) Model Properties

The Macro Porosity (fully coupled) model has the following properties:

Void Pressure

The pressure p_{void} in Eqn. (1424) below which a gaseous phase void is assumed to outgas from the liquid phase. The default value is 50000 Pa.

Under-Relaxation Factor

At each iteration, this property governs the extent to which the newly computed void phase fraction supplants the old solution. The default value is 1.0.

Scaling Factor

The scaling factor in the Macro Porosity (fully coupled) model is a multiplicative factor to the source rate of the void phase. This value is s in Eqn. (1430). Limited to $s \geq 0$.

See Variation of the Scaling Factor.

Macro Porosity (pure thermal) Child Nodes

The Macro Porosity (pure thermal) model has the following child nodes:

Solid Cell Indicator

Identifies the cells that are considered solid and the cells that are fluid. Macro porosity can occur only for liquid zones that are fully enclosed by either wall boundaries or solid cells. A value of zero indicates a fluid cell, a non-zero value indicates a solid cell. A **Liquid Zone Index** value is assigned to each liquid cell, indicating the enclosed liquid zone to which the cell belongs. Shrinkage, and consequently the target amount of melt that is replaced by void to keep the melt mass constant, is computed for each of the cell clumps that have the same **Liquid Zone Index**.

If you choose the Field Function method, the **Flow Stop Flag** field function is set automatically. This is a built-in field function that is available with the Melting-Solidification model. See Melting-Solidification Model Field Functions.

The default setting is a constant value of 0.

Pore Probability Indicator

Identifies the cells within which melt is replaced with void in cases of shrinkage, in order of priority. For each group of cells that have the same **Liquid Zone Index**, the replacement of the target amount of melt with void is performed first in the cell that has the maximum value for the pore probability.

For expansion, the inverse value of the pore probability is used to determine the cells in which the void phase is replaced with liquid phase.

For example, when the **Gravity Potential** method (the default setting) is used for the pore probability profile:

- To model shrinkage, the liquid phase is replaced in the upper-most cells of the liquid zone that contain liquid phase.
- To model expansion, the void phase is replaced with liquid phase in the lowest cells of the liquid zone that contain void phase.

  If no void phase is available in the liquid zone, the mass change of the liquid zone cannot be compensated for. A mass conservation error can result.

Method	Description
Gravity Potential	The phase replacement takes place in the highest (with respect to the gravity vector) cells of each enclosed liquid region. After the phase replacement, the melt volume is the same as it was at the beginning of the previous time-step. If no gravity model is selected for the simulation, it is considered a zero-gravity simulation and the pore probability is uniformly zero.
Pressure Field	The phase replacement takes place in the cells of each enclosed liquid region where the absolute pressure is minimal. After the phase replacement, the melt volume is the same as it was at the beginning of the previous time-step.

Field Functions

The following primitive field functions are available to the simulation when the **Macro Porosity (pure thermal)** model is used.

Contiguous Stopped Region Index

A domain-wide unique index of contiguous stopped regions. This is used to indicate that a cell belongs to a certain liquid zone enclosed by either flow-stopped cells or boundaries.

Liquid Zone Index of <phase interaction>

Index of the contiguous liquid zone. A Liquid Zone Index value is assigned to each liquid cell, indicating the enclosed liquid zone to which the cell belongs. Shrinkage, and the appropriate phase replacement, is computed for each of the zones that have the same Liquid Zone Index.

Phase Replacement Volume Fraction Limit of <phase interaction>

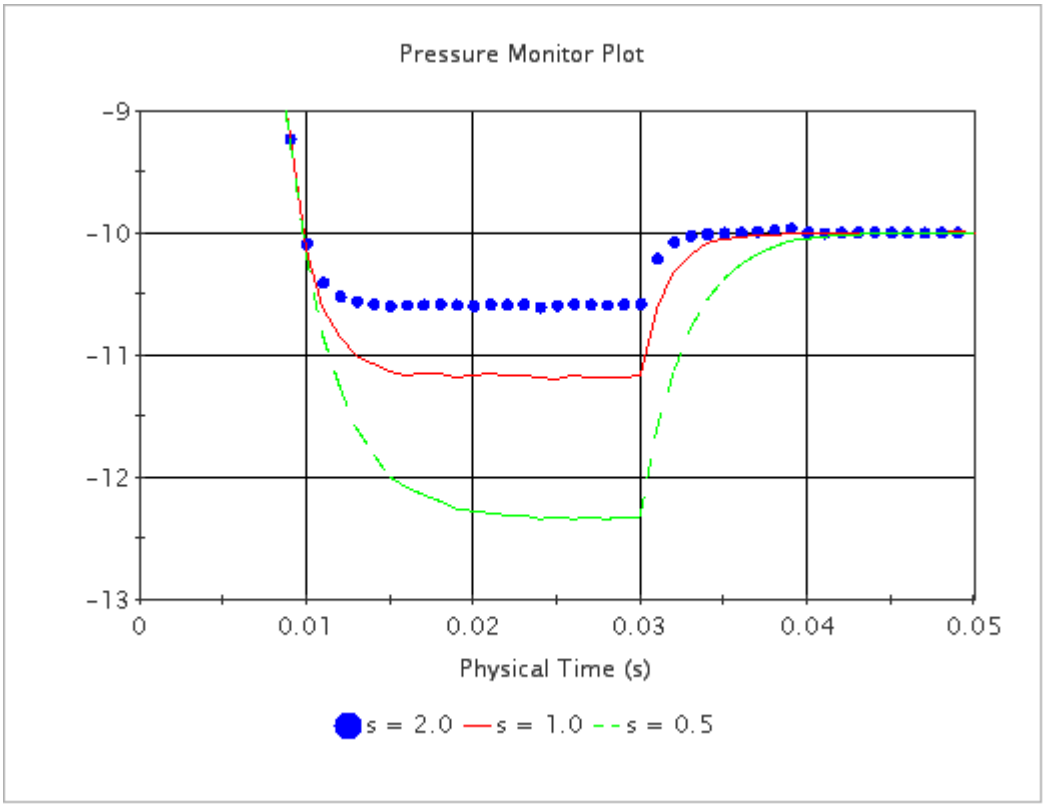
Positive values indicate shrinkage, negative values indicate expansion.

Variation of the Scaling Factor

The scaling factor s in the **Macro Porosity (fully coupled)** model is a multiplicative factor to the source rate of the void phase.

Consider a liquid region that is enclosed by flow-stopped cells and cooled by a volumetric heat source. The temperature decrease causes shrinkage that the void gas compensates for such that the mixture density remains constant. The pressure asymptotically approaches a constant value slightly below the target void pressure depending on the cooling rate and the scaling factor. After reaching a constant pressure level, the volumetric heat source is deactivated, and the pressure asymptotically approaches the specified void pressure.

The scaling factor influences the minimum pressure and the time at which it is reached. The lower the source rate, the longer it takes for the void phase to compensate for the pressure decrease. Hence, the largest scaling factor—2.0 in the following plot—yields the largest minimum pressure value.



Macro Porosity Formulation

When the static pressure in a cell falls below the user-specified void pressure, a fraction of the liquid phase is converted to the gaseous void phase. The total fluid mass is kept constant.

This formulation applies to the **Macro Porosity (fully coupled)** model only. The **Macro Porosity (pure thermal)** model is a simple model that attempts to keep the mass of the primary phase constant for each liquid zone.

Macro Porosity (fully coupled) Model Criterion

At the same pressure, a certain amount of a light gas fills a larger volume than the same mass of a heavy liquid. Since the cell volume is constant, the pressure is increased. This mechanism is used to keep the pressure from decreasing significantly below the target void pressure.

The equation of state of an ideal gas, the void phase void, is:

$$p = \rho \frac{R_u}{M_{\mathrm{void}}} T \tag{1423}$$

with the universal gas constant R_u = 8314.4621 J/kmol K, and M_{void} is the molecular weight in kg/kmol.

Mass and cell volume are constant, and pressure increases during the phase conversion. Therefore, the following relation is fulfilled even for the minimum temperature in any cell during the simulation that is run at $p = p_{\mathrm{void}}$:

$$p_{\mathrm{void}} \leq \rho_l \frac{R_u}{M_{\mathrm{void}}} T_{\mathrm{min}} \tag{1424}$$

Hence, the Macro Porosity (fully coupled) model can operate correctly when the molecular weight of the ideal gas void phase is:

$$M_{\mathrm{void}} \leq \frac{\rho_l R_u T_{\mathrm{min}}}{p_{\mathrm{void}}} \tag{1425}$$

However, this criterion is easily met for common casting material properties:

$M_{\mathrm{air}} = 28.9664 \leq 44898.1 = \frac{2700 \cdot 8314.4621 \cdot 100}{50000}$

Void Phase Source

The rate at which the liquid phase is converted to the void phase without pressure decreasing below p_{void} must be determined.

The relative volume change of the cell that leads to a pressure increase up to p_{void} is computed. The result is the fraction of the cell volume that must be filled with void gas at $p = p_{\mathrm{void}}$ and $T = T_{\mathrm{cell}}$.

$\Delta\, p$ is the difference between the user-specified void pressure p_{void} and the static pressure p in the cell:

$$\Delta\, p = p_{\mathrm{void}} - p \tag{1426}$$

This pressure difference causes a change in mixture density in the cell, which is approximated with the derivative of mixture density with respect to pressure $\partial \rho / \partial p$. This approximation assumes that $\partial \rho / \partial p$ is constant for small pressure deviations:

$$\Delta\, \rho \approx \Delta\, p \frac{\partial \rho}{\partial p} \tag{1427}$$

The relation between a density change and the related volume change for constant fluid mass in a cell is:

$$
\begin{aligned}
m &= \rho V \\
\rho V &= (\rho + \Delta\rho)(V + \Delta V) \\
\frac{\rho V}{\rho + \Delta\rho} - V &= \Delta V \\
\frac{\rho}{\rho + \Delta\rho} - 1 &= \frac{\Delta V}{V} \\
-\frac{\Delta\,\rho}{\rho + \Delta\,\rho} &= \frac{\Delta\,V}{V}
\end{aligned}
\tag{1428}
$$

Eqn. (1428) gives the fraction of the cell volume that must be filled with the void phase to compensate (inverted sign) the volume change of the mixture when the pressure is increased again to void pressure:

$$
\frac{\Delta\,V}{V} = \frac{\Delta\,\rho}{\rho + \Delta\,\rho} \tag{1429}
$$

A VOF-phase source is specified in STAR-CCM+ in the units $\frac{\text{Volume}}{\text{Volume-Time}} = \frac{1}{\text{s}}$. The relative volume change applied per unit time (second) and an under-relaxation factor gives the rate at which the void fluid is converted from the liquid phase to the gaseous phase.

$$
\begin{aligned}
\left.\frac{\Delta\,V}{V}\right|_{\text{void}} &= \max\left[0, \frac{\Delta\,\rho}{\rho + \Delta\,\rho} s\right] \\
&= \max\left[0, \frac{(p_{\text{void}} - p)\frac{\partial\rho}{\partial p}}{\rho + (p_{\text{void}} - p)\frac{\partial\rho}{\partial p}} s\right]
\end{aligned}
\tag{1430}
$$

Values without index are cell values of the mixture. The scaling factor s is limited to values $s \geq 0$. It allows an adaptation to the specific case where larger rates are required to increase the minimum pressure.

The rate is limited to positive values to prevent emitted gas being dissolved again.

Void Phase Source Linearization

The linearization of the void phase source term is the derivative of Eqn. (1430) with respect to pressure (assuming $\partial\rho / \partial p$ is constant):

$$
\frac{\partial}{\partial p}\left(\left.\frac{\Delta\,V}{V}\right|_{\text{void}}\right) = \frac{\partial}{\partial p}\left[\frac{(p_{\text{void}} - p)\frac{\partial\rho}{\partial p}}{\rho + (p_{\text{void}} - p)\frac{\partial\rho}{\partial p}} s\right] = -\frac{p\frac{\partial\rho}{\partial p}}{\left[\rho + (p_{\text{void}} - p)\frac{\partial\rho}{\partial p}\right]^2} s \tag{1431}
$$

Macro Porosity Nomenclature

The terms that are used in the Macro Porosity (fully coupled) model formulation are defined below.

ρ	Density
M_{void}	Molecular weight of the void phase
R_u	Universal gas constant
T	Temperature
V	Volume

p	Pressure
s	Scaling factor

Defect Prediction Using Criteria Functions

Criteria functions are post-processing items to help predict casting defects like macro- or micro-porosity and to estimate relevant microstructural quantities such as primary and secondary dendrite arm spacings.

They are scalar and vector quantities available as field functions when the following models are selected:

- Criteria Functions model

  Select this model for any criteria function to become available.
- Primary Dendrite Arm Spacing model

  Select the Criteria Functions model before this model becomes available.
- Secondary Dendrite Arm Spacing models

  Select the Criteria Functions model before this model becomes available.
- Dimensionless Niyama model

  You must select the Criteria Functions model and one of the Secondary Dendrite Arm Spacing models before this model becomes available. You are advised to use the Secondary Dendrite Arm Spacing (CR) model (see Secondary Dendrite Arm Spacing (CR)) in combination with the Dimensionless Niyama model. The original formulation of the Dimensionless Niyama criterion [193] uses the secondary dendrite arm spacing computed from the cooling rate.

Criteria functions can be plotted in a scene for visualization where the distribution of these quantities allows conclusions about casting defects.

Some criteria functions are collected over the complete duration of the simulation and contain the complete solution history. Therefore, whenever any property of the Criteria Functions model or any property relevant to the melting/solidification process is modified, the simulation must be restarted from scratch. No solution history file or transient post file is required to evaluate Criteria Functions: all the required data is stored in the `.sim` file.

The main point of interest is to evaluate the criteria functions at the critical temperature when the macroscopic flow stops due to solidification. Liquid melt cannot be further supplied to compensate for shrinkage effects. However, this critical temperature is not known exactly because the macroscopic stop of flow depends on the casting process and the alloy. Therefore, a set of temperatures close to the estimated critical temperature is specified in the range between the solidus and liquidus temperature. The criteria functions are evaluated at the several specified critical temperatures.

Contents:

Setting Up a Criteria Function
Criteria Functions Definitions
Criteria Functions Properties

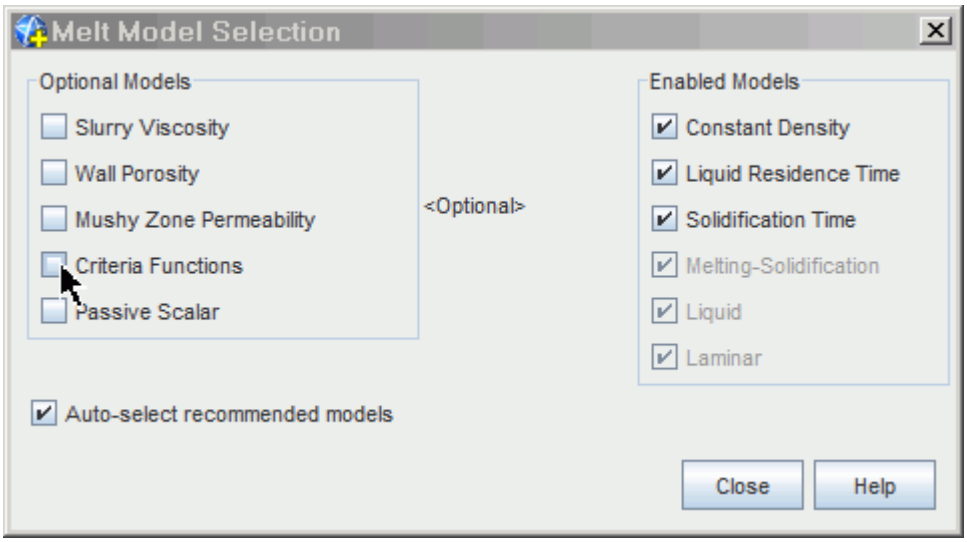
Setting Up a Criteria Function

You select the particular criteria function that you require as a phase interaction model.

To set up a Criteria Function:

1. Select the Casting model.
2. Select **Multiphase Interaction** from the *Optional Models* group box.
3. In the simulation tree, under the **Models** > **Eulerian Multiphase** node, create the Eulerian phase.
4. Under the Eulerian phase, open the *Phase Model Selection* dialog and select the following models:
   a) Select **Liquid** from the *Material* group box.
   b) Select **Melting-Solidification** from the *Optional Models* group box.
   c) Select **Solidification Time** from the *Optional Models* group box.

      You must select the Solidification Time model first for the Criteria Functions model to become available.
   d) Optionally, select **Liquid Residence Time** from the *Optional Models* group box.
   e) Select **Criteria Functions** from the *Optional Models* group box.

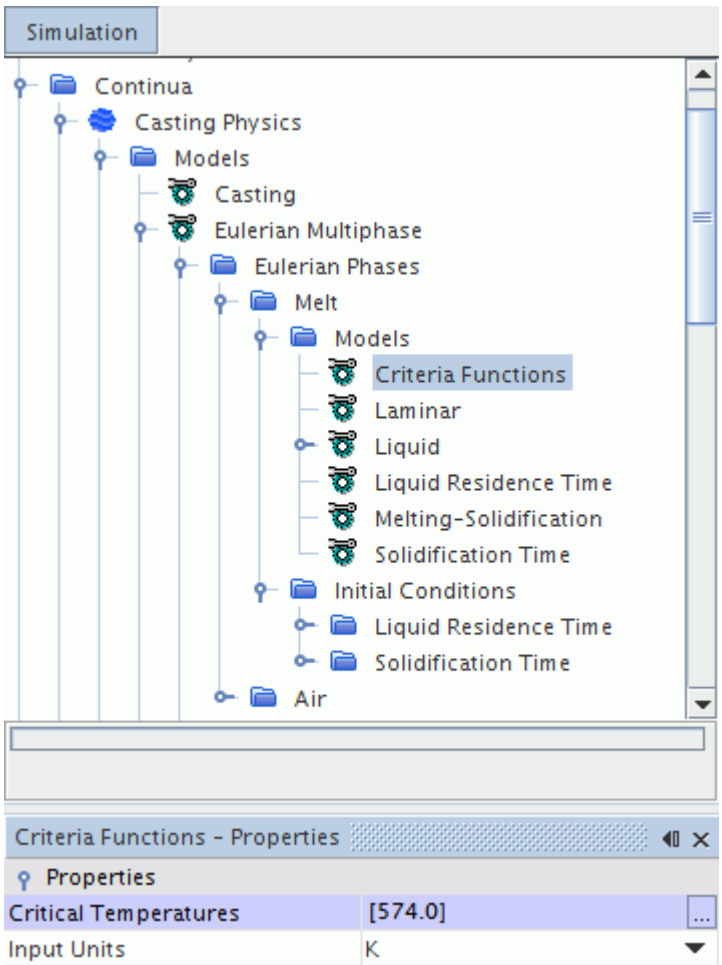


5. In the *Phase Model Selection* dialog, within *Optional Models*, select the particular criteria function that you require:
   1. Select **Primary Dendrite Arm Spacing**.
   2. Select **Secondary Dendrite Arm Spacing** and then select one of the following sub-models:
      - **Secondary Dendrite Arm Spacing (CR)**
      - **Secondary Dendrite Arm Spacing (LST)**
   3. Select **Dimensionless Niyama Criterion**.

   Note: You are advised to select the Secondary Dendrite Arm Spacing (CR) model in combination with the Dimensionless Niyama criterion, because only this model complies with the original formulation of the Dimensionless Niyama model [193].

6. Click **Close**.

   The **Criteria Functions** node, the **Liquid Residence Time** node, and the **Solidification node** are added to the Eulerian phase models in the simulation tree. The **Initial Conditions** node is added to the Eulerian phase.



7. Select each child node of the **Initial Conditions** node in turn and set the initial conditions for:
   - Liquid Residence Time
   - Solidification Time

Criteria Functions Definitions

The following Criteria Functions are available:

Contents:

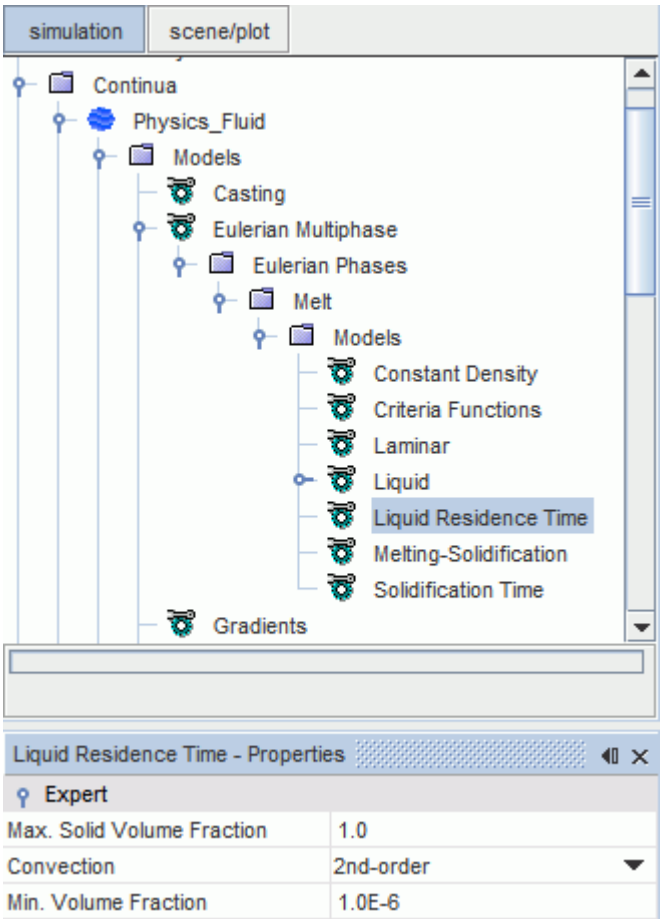
Liquid Residence Time
Solidification Time
Local Solidification Time
Cooling Rate
Isotherm Speed
Temperature Gradient
Mean Cooling Rate
Solidification Velocity

G/v
Niyama Criterion
Dimensionless Niyama Criterion
Primary Dendrite Arm Spacing
Secondary Dendrite Arm Spacing (LST)
Secondary Dendrite Arm Spacing (CR)

Liquid Residence Time

The fluid enters the computational domain in a liquid state with a temperature above the solidus temperature. The Liquid Residence Time is the time between the fluid entering the computational domain and the moment that the temperature decreases below the solidus temperature.

This time is computed for every VOF phase for which you activate the Liquid Residence Time model (Setting Up a Criteria Function).



Liquid Residence Time Properties

Max. Solid Volume Fraction	The residence time is accumulated only for solid volume fractions below the specified value.

Convection	The convection scheme to use. For guidance on selecting a convection scheme, see Convective Flux.	
	1st-order	First-order convection scheme.
	2nd-order	Second-order convection scheme.
Min. Volume Fraction	The residence time is accumulated only for liquid volume fractions above the specified value.	

Liquid Residence Time is transported only by convection. The initial value is zero everywhere. A source term adding the time step to the residence time at every time step is applied to every cell in which the temperature of the VOF phase is above solidus temperature.

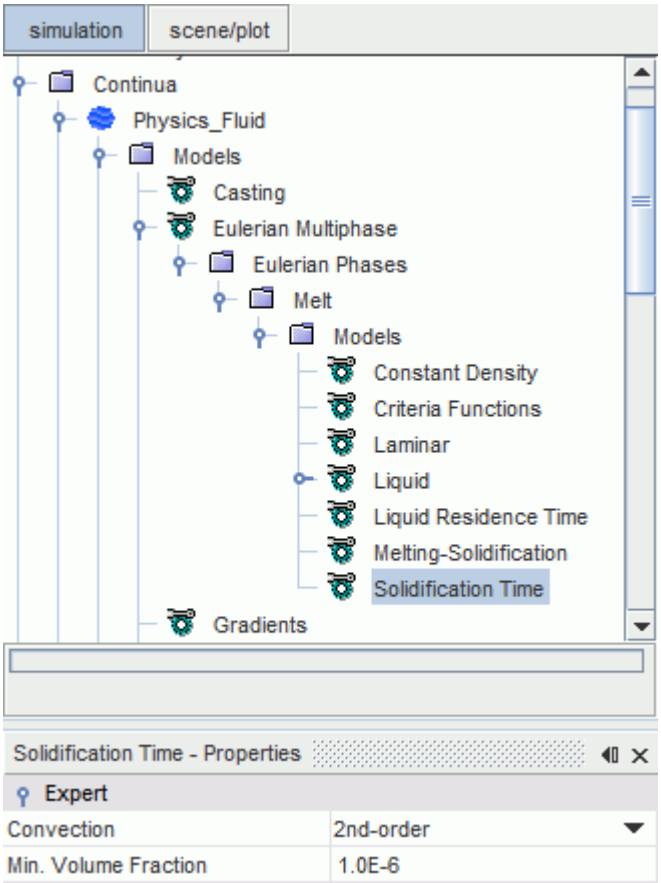
The Liquid Residence Time is zero for every cell with a phase volume fraction below the minimum volume fraction.

Note: The computation of the Liquid Residence Time is optional, because no other Criteria Function depends on this value.

Solidification Time

The Solidification Time is the time that is taken to cool down the VOF phase from liquidus temperature to solidus temperature.

This time is computed for every VOF phase for which you activate the Solidification Time model (see Setting Up a Criteria Function). The appropriate critical temperatures must be specified in the Criteria Functions model.



Solidification Time Properties

Convection	The convection scheme to use. For guidance on selecting a convection scheme, Convective Flux.	
	1st-order	First-order convection scheme.
	2nd-order	Second-order convection scheme.
Min. Volume Fraction	The residence time is accumulated only for liquid volume fractions above the specified value.	

Solidification time is transported only by convection. The initial value is zero everywhere. In every cell in which the VOF phase is between the liquidus and solidus temperatures, a source term that adds the time step to the solidification time is applied at every time step. The scalar value equals the accumulated time that the VOF phase spent within the computational domain in a partly solidified state.

The value of the passive scalar is reset to zero when the phase remelts. That is, when the temperature increases above liquidus temperature after being in the interval between liquidus and solidus temperature for more than one iteration. Hence, the Solidification Time represents only the time that the phase spent between liquidus and solidus temperature for the last time. It is zero for every cell where the temperature is above liquidus temperature or the volume fraction of the phase is below the minimum volume fraction.

Large solidification times indicate hot spots in casting; these spots are commonly due to porosity defects. On the other hand, a very short solidification time for cast iron indicates the presence of white solidified areas in casting.

This quantity is used to compute the Local Solidification Time.

Local Solidification Time

The scalar quantity Local Solidification Time is collected for all specified Criteria Temperatures below the liquidus temperature. The Local Solidification Time is the value of the Solidification Time passive scalar in the cell at the moment the temperature decreases below the respective Criteria Temperature.

The field is initialized with zero and values are modified in the iteration when the temperature falls below the respective Criteria Temperature.

The Local Solidification Time $t|_{T_{crit}}$ measures how long the fluid was in a partially solidified state before finally solidifying in the cell. Short local solidification times indicate a fine microstructure, and thus enhanced mechanical properties for aluminum and steel alloys.

This quantity is used for the computation of:

- (spatial) Local Solidification Time Gradient $\nabla t|_{T_{crit}}$ — not available as a field function but required for the computation of Isotherm Speed.
- Isotherm Speed
- Mean Cooling Rate—requires only Local Solidification Time for $T_{crit} = T_{\text{solidus}}$.
- G/v—requires Local Solidification Time Gradient.
- Secondary Dendrite Arm Spacing (LST)

Cooling Rate

The scalar quantity Cooling Rate is the time derivative of temperature. The cooling rate is computed for all specified Criteria Temperatures.

Values are initially zero and are only overwritten each iteration when the temperature falls below the respective Criteria Temperature.

$$\dot{T}\Big|_{T_{crit}} = \frac{\partial T}{\partial t}\Big|_{T_{crit}} \tag{1432}$$

Analyzing the cooling rate in the temperature interval of interest can provide information on local microstructure properties, and thus on local mechanical properties.

For example, a high cooling rate in the solidification interval of an aluminum alloy indicates a fine microstructure: this fine microstructure corresponds to better tensile strength in these casting areas. On the other hand, cooling rates in the solidification interval close to the eutectic temperature for cast iron indicate that cementite can appear.

This quantity is used for the computation of:

- Solidification Velocity

- Niyama Criterion

Isotherm Speed

The scalar quantity Isotherm Speed is an approximation of the propagation velocity of the isotherm at the respective Criteria Temperature.

$$v_{iso}\Big|_{T_{crit}} = \frac{\nabla t|_{T_{crit}}}{\left|\nabla t|_{T_{crit}}\right|^2} \tag{1433}$$

It is computed for all specified Criteria Temperatures.

Temperature Gradient

The vector quantity Temperature Gradient $\Delta\, T|_{T_{crit}}$ is the spatial derivative of temperature. It is computed for all specified Criteria Temperatures.

Analyzing the temperature gradient in the temperature interval of interest can provide information on local microstructure properties, and thus on local mechanical properties.

For example, a high temperature gradient in the solidification interval of an aluminum alloy indicates a fine microstructure: this fine microstructure corresponds to good tensile strength in these casting areas. On the other hand, high temperature gradients in the solidification interval close to the eutectic temperature for cast iron indicate that cementite can appear.

This quantity is used for the computation of:

- Solidification Velocity
- G/v
- Niyama Criterion
- Primary Dendrite Arm Spacing

Mean Cooling Rate

The scalar quantity Mean Cooling Rate is the average cooling rate over the entire solidification process.

$$\dot{\bar{T}} = \frac{T_{\text{solidus}} - T_{\text{liquidus}}}{t|_{T_{\text{solidus}}}} \tag{1434}$$

Solidification Velocity

The vector quantity Solidification Velocity is an approximation for the velocity of the solidification front. This quantity is computed for all specified Criteria Temperatures.

$$v_s\Big|_{T_{crit}} = \frac{\dot{T}\Big|_{T_{crit}} \nabla T|_{T_{crit}}}{\left|\nabla T|_{T_{crit}}\right|^2} \tag{1435}$$

The solidification velocity is important for the analysis of directional solidification of single crystal castings. Inhomogeneity of the solidification velocity pattern can indicate areas in casting that contain crystals of different orientation.

This quantity is used for the computation of Primary Dendrite Arm Spacing.

G/v

The scalar quantity G/v is the ratio between the temperature gradient G and the solidification velocity v and is an important measure for the resulting micro structure.

This quantity is approximately the scalar product of the Temperature Gradient and the gradient of the Local Solidification Time, and is computed for all Criteria Temperatures below liquidus temperature:

$$G/v|_{T_{crit}} = \nabla T\,|_{T_{crit}} \cdot \nabla t|_{T_{crit}} \tag{1436}$$

This criteria function is used in the analysis of interface stability and is thus used to predict the solidification type: planar, cellular, or dendritic. High G/v ratios increase the probability for planar interface solidification, and lower G/v ratios leads to cellular or dendritic solidification.

Niyama Criterion

The scalar quantity Niyama Criterion estimates the porosity probability from the ratio between the shrinkage rate and the feeding distance.

The shrinkage rate is assumed to be proportional to the Solidification Velocity and the feeding distance is assumed to be proportional to the Temperature GradientGradient. The criterion is computed for all Critical Temperatures.

$$K_{\text{Niyama}} = \frac{\nabla T|_{T_{crit}}}{\sqrt{-\dot{T}|_{T_{crit}}}} \tag{1437}$$

The Niyama Criterion predicts central porosity areas that appear as a result of melt contraction in the last stages of solidification. This criterion applies to low-carbon steel alloys. The critical value of this criterion, below which shrinkage defects form, is determined experimentally.

Dimensionless Niyama Criterion

You can use the dimensionless Niyama criterion to directly predict the amount of shrinkage porosity during solidification of metal alloy castings.

This criterion does not require any threshold value below which shrinkage porosity forms. You can use this criterion for pure thermal simulations without computing pressure or velocity fields. The criterion is computed for all Critical Temperatures. The implementation is based on [193].

Solid Fraction Curve Integral

First, the fraction solid curve is integrated:

$$I\left(\alpha^*_{l,cr}\right) = \int_{\alpha^*_{l,cr}}^{1} 180 \cdot \frac{\left(1-\alpha^*_l\right)^2}{\alpha^{*2}_l} \cdot \frac{\partial}{\partial \alpha^*_l}\left(T^*\right) \cdot \mathrm{d}\,\alpha^*_l \tag{1438}$$

where α^*_l is the relative liquid volume fraction. α^*_s is computed from the relative solid volume fraction α^*_s as follows:

$\alpha^*_l = 1-\alpha^*_s$

$\alpha^*_{l,cr}$ is the critical liquid volume fraction at which the melt pressure decreases to the critical pressure value and porosity begins to form. T^* is the normalized temperature and is defined as:

$$T^* = \frac{T-T_{solidus}}{T_{liquidus}-T_{solidus}} \tag{1439}$$

The integral in Eqn. (1438) is evaluated numerically using at least the user-specified *Min. Number of Integration Points (*see Dimensionless Niyama Criterion Properties). The integral is calculated only once when the Dimensionless Niyama model is activated or when the Fraction Solid Curve material property changes.

If the fraction solid curve is specified as a function of pressure, a pressure of 1 atm is applied for its evaluation. STAR-CCM+ assumes that the fraction solid curve is measured at this pressure.

Computation of the Dimensionless Niyama Criterion

At the end of every time step, the value of the Dimensionless Niyama criterion is computed as:

$$Ny^* = \lambda_2 Ny \sqrt{\frac{\Delta P_{cr}}{\mu_{liq}\beta\Delta T_f}} \tag{1440}$$

where:

ΔP_{cr}	Critical pressure drop
μ_{liq}	Dynamic viscosity at liquidus temperature
β	Solidification contraction
$\Delta T_f = T_{liq}-T_{sol}$	Freezing temperature range
λ_2	Secondary Dendrite Arm Spacing from Eqn. (1447)

The critical pressure drop is defined as:

$$\Delta P_{cr} = P_{liq}-P_{cr} \tag{1441}$$

where P_{liq} is the pressure at liquidus temperature. P_{liq} is the sum of the ambient pressure of the system and the local head pressure. P_{cr} is the critical pressure and is calculated by considering the mechanical equilibrium that is necessary for a stable pore to exist. Here, P_{cr} is an adjustable parameter.

The solidification contraction is defined as:

$$\beta = \frac{\rho_{sol} - \rho_{liq}}{\rho_{liq}} \tag{1442}$$

The liquid and solid densities, ρ_{liq} and ρ_{sol}, are considered constant during solidification.

The original formulation of the Dimensionless Niyama criterion [193] uses the secondary dendrite arm spacing computed from the cooling rate (see Secondary Dendrite Arm Spacing (CR)).

Computation of the Solid Fraction Curve Integral Target Value

The target value of the solid fraction integral is given by:

$$I\left(\alpha^*_{l,cr}\right) = Ny^{*2} \tag{1443}$$

Computation of the Pore Volume Fraction

Knowing the target value of the solid fraction integral, the critical liquid fraction $\alpha^*_{l,cr}$ can be calculated. $\alpha^*_{l,cr}$ is the integration limit of Eqn. (1438).

Computation of the Absolute Pore Volume Fraction

The pore volume fraction is:

$$\alpha_p = \beta' \alpha^*_{l,cr} = \frac{\beta}{\beta + 1} \alpha^*_{l,cr} = \frac{\rho_{sol} - \rho_{liq}}{\rho_{sol}} \alpha^*_{l,cr} \tag{1444}$$

Dimensionless Niyama Criterion Properties

Dynamic Viscosity @ T_liquidus	The dynamic viscosity at liquidus temperature for the melt material.
Critical Pressure Drop	Defined as the difference between the pressure at liquidus temperature and the critical pressure (see Eqn. (1441)).
Solidification Contraction	β in Eqn. (1442).
Min. Number of Integration Points	The minimum number of integration points used to numerically integrate Eqn. (1438).

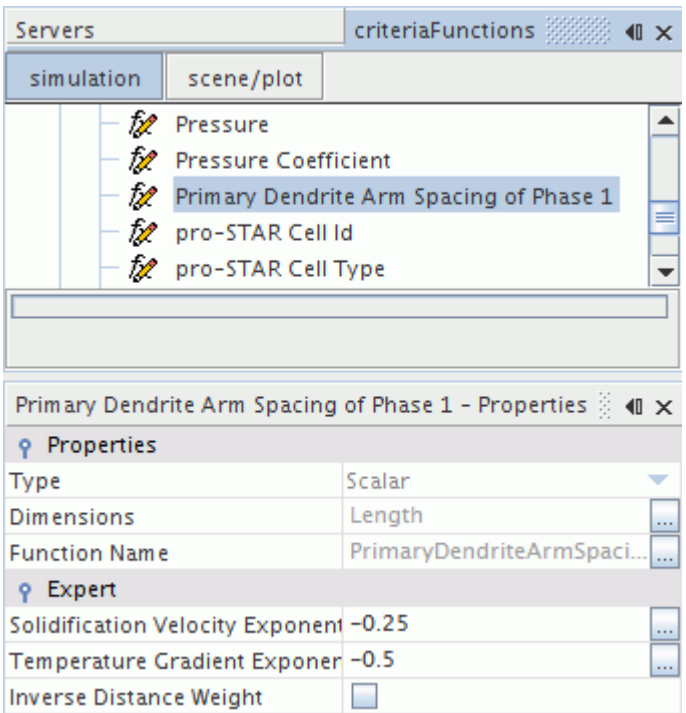
Primary Dendrite Arm Spacing

The scalar quantity primary dendrite arm spacing (also referred to as PDAS or DAS) is the distance between the axes of the dendrites. This quantity is computed only for liquidus temperature.

It is possible to calculate the primary dendrite arm spacing from the Temperature Gradient and the Solidification Velocity. This calculation uses the material property Primary Dendrite Arm Spacing Coefficient K_1 and the field function properties Solidification Velocity Exponent a and Temperature Gradient Exponent b.

$$\lambda_1 = K_1\left(\left|v_s|_{T_{\text{liquidus}}}\right|\right)^a\left(\left|\nabla T|_{T_{\text{liquidus}}}\right|\right)^b \tag{1445}$$

Since both exponents are derived from theory, they are field function properties and not material properties. By contrast, the exponent that is used for the computation of the Secondary Dendrite Arm Spacing (LST) is a material property.



The primary dendrite arm spacing is used to estimate the local mechanical properties in columnar solidification. The finer the primary dendrite arm spacing is, the higher the mechanical properties.

Primary Dendrite Arm Spacing Properties

Coefficient	K_1 in Eqn. (1445).
Solidification Velocity Exponent	a in Eqn. (1445).
Temperature Gradient Exponent	b in Eqn. (1445).

Secondary Dendrite Arm Spacing (LST)

The scalar quantity Secondary Dendrite Arm Spacing (LST) (also referred to as SDAS-LST) is the distance between the arms of dendrites. This quantity is computed for all specified Criteria Temperatures.

The secondary dendrite arm spacing (LST) can be estimated from the Local Solidification Time using the material properties Secondary Dendrite Arm Spacing Coefficient K_2 and Secondary Dendrite Arm Spacing Exponent c.

$$\lambda_{2,LST}\Big|_{T_{crit}} = K_{2,LST}\left(t_{T_{crit}}\right)^c \tag{1446}$$

The secondary dendrite arm spacing is used to estimate local mechanical properties in both columnar and equiaxed solidification. The finer the secondary dendrite arm spacing is, the higher the mechanical properties.

Secondary Dendrite Arm Spacing (LST) Properties

Coefficient	$K_{2,LST}$ in Eqn. (1446).
Exponent	c in Eqn. (1446).

Secondary Dendrite Arm Spacing (CR)

The scalar quantity Secondary Dendrite Arm Spacing (CR) (also referred to as SDAS-CR) is the distance between the arms of dendrites. This quantity is computed for all specified Criteria Temperatures.

The secondary dendrite arm spacing (CR) can be estimated from the Cooling Rateusing the material properties Secondary Dendrite Arm Spacing Coefficient K_2 and Secondary Dendrite Arm Spacing Exponent c.

$$\lambda_{2,CR}\Big|_{T_{crit}} = K_{2,CR}\left(\dot{T}|_{T_{crit}}\right)^c \tag{1447}$$

Secondary Dendrite Arm Spacing (CR) Properties

Coefficient	$K_{2,CR}$ in Eqn. (1447).
Exponent	c in Eqn. (1447).

Criteria Functions Properties

You specify the critical temperatures between the solidus and liquidus temperatures at which Criteria Functions are evaluated.

Critical Temperatures	The array of critical temperatures between the solidus and liquidus temperatures at which Criteria Functions are evaluated. When you specify one or more critical temperatures, the Solidification Time function is activated. You can enter any number of critical temperatures.
Input Units	The unit for the Critical Temperatures array.

Automated Time-Step Control

To adjust the time-step automatically during the run, you can use the Segregated Time Step model. The Segregated Time Step model has the potential to improve the quality and run-time of your simulation results by automatically controlling the time-step size to help attain a specified temporal resolution. This adjustment is particularly useful for cases with large variations of flow topology or large varying time scales of the physics.

When you activate the Segregated Time Step model, the available methods are listed automatically. A Time Step method is available if the quantity on which its condition is based is available: that is, the corresponding physics model is selected. The Segregated Time Step model is not available within a solid continuum.

For each available Time Step method, the corresponding node is added under the **Segregated Time Step** node. The Time Step methods are:

Flow	Uses a time-step condition based on the Courant number weighted by the rate of change of the absolute pressure. This method is available when the Segregated Flow model is enabled for the physics continuum.
VOF	Uses a time-step condition based on the Courant number weighted by the rate of change of the volume fraction. This method is available when the Volume of Fluid (VOF) model is enabled for the physics continuum.
Energy	Uses a time-step condition based on the Courant number weighted by the rate of change of the total energy. This method is available only when the Segregated Multiphase Temperature model is enabled for the physics continuum.
Melting-Solidification	Uses a time-step condition based on the propagation velocity of the melting-solidification interface. This method is available only when the Melting-Solidification model is enabled for at least one VOF phase.

When the simulation is run, the Segregated Time Step model evaluates each of the available Time Step methods to determine a time-step change factor for each method. The Segregated Time Step model then uses the lowest of those change factors to calculate a new time-step size based on the previous time-step size. The new time-step size must lie within the specified upper and lower bounds, and the maximum time-step change factors for increasing or decreasing time-step sizes are also taken into consideration. The upper bound is set in the Implicit Unsteady solver; the lower bound and the maximum time-step change factors are set in the Segregated Time Step model. See Setting up the Segregated Time Step Model.

Each available Time Step method uses its own Segregated Time Step Condition as the fundamental quantity to determine whether the specified limit is satisfied and therefore determine the direction in which the new time-step size can be changed. The time-step condition can undergo a volume-weighted smoothing procedure prior to the evaluation of the time step-size change factor.

You can deactivate an available Time Step method by setting the *Max Condition Limit* value to zero. This setting prevents the Time Step method from providing a valid time-step change factor. However, any data remains available for post-processing purposes.

The Segregated Time Step Condition variable of each available Time Step method is made available as a field function under the name *Time Step Condition [method name]*.

Contents:

Setting up the Casting Time Step Model
Time Step Control Properties

Setting up the Segregated Time Step Model

The Segregated Time Step model is available only when the Casting, Implicit Unsteady, and VOF models are selected.

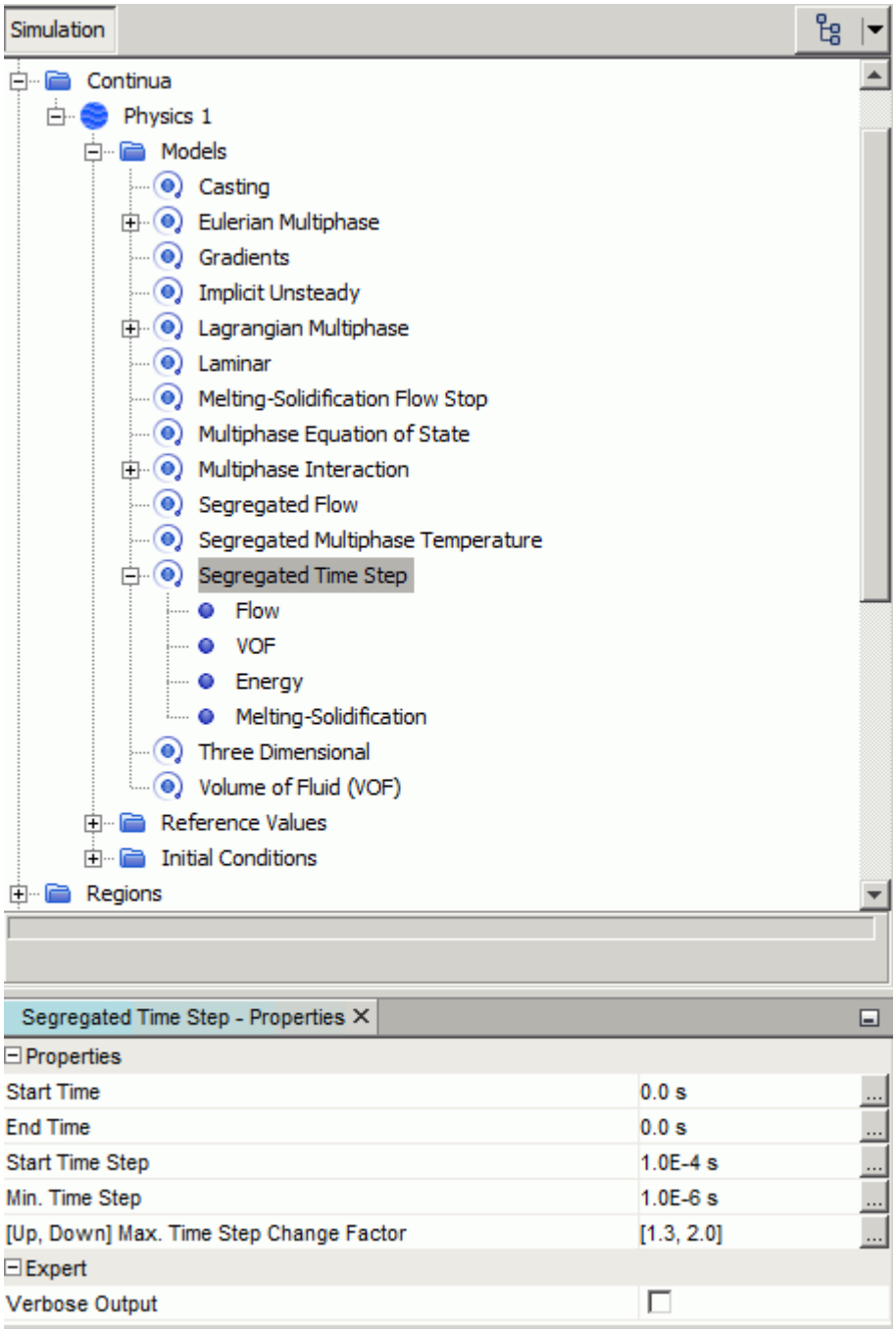
See Selecting the Casting Model.

To set up the Segregated Time Step model:

1. Right-click the **Continua** > **Physics 1** > **Models** node and choose **Select models...**.

   The *Model Selection* dialog appears.
2. In the *Optional Models* group box, activate **Segregated Time Step**.
3. Click **Close**.

   The **Segregated Time Step** node is added to the simulation tree.

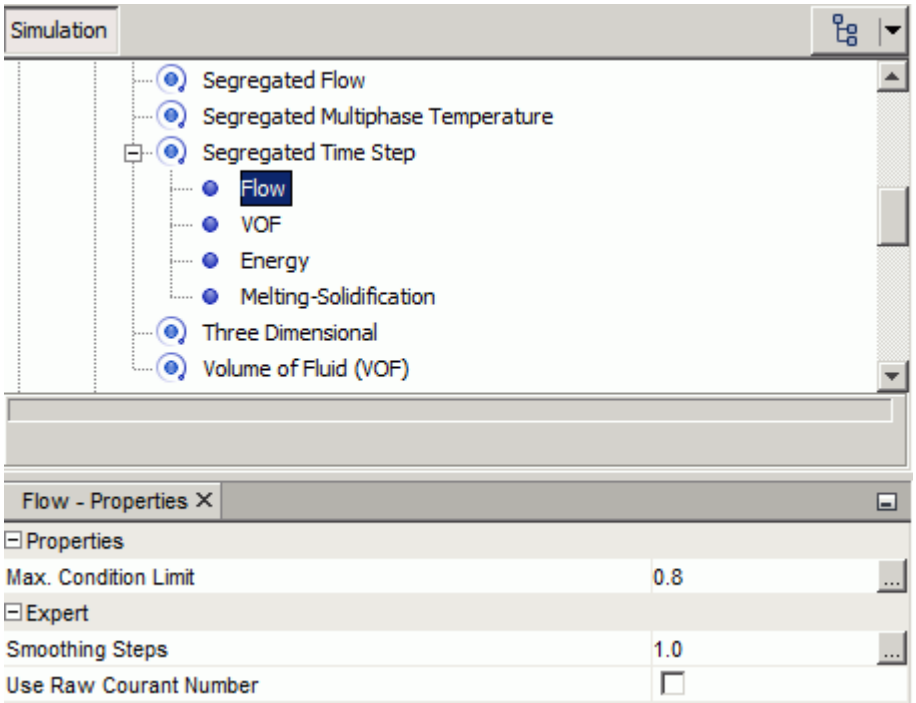


4. Select the **Segregated Time Step** node and set the relevant properties.

   See Segregated Time Step Control Properties.

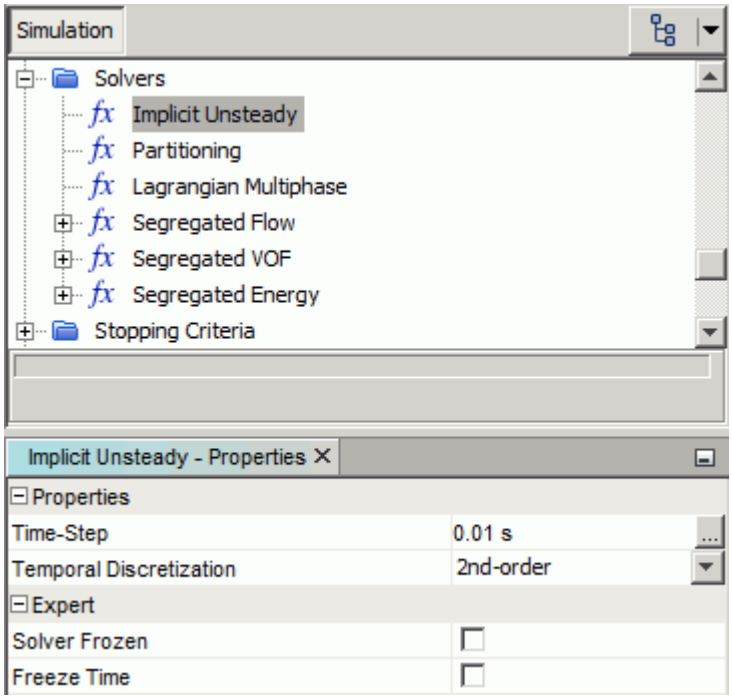
5. For each available Segregated Time Step method, select the corresponding node and set the relevant properties.

   See Segregated Time Step Method Properties.



In the Implicit Unsteady solver, specify the maximum time-step. This value is the upper time-step size limit for the automatic time-step computation.

6. Select the **Solvers** > **Implicit Unsteady** node and set *Time-Step* to the maximum time-step value.



Time Step Control Properties

You can set the Time Step Control properties.

Segregated Time Step Control Properties

Start Time	The time at which automatic time-step control becomes active.
End Time	The time at which automatic time-step control is deactivated. This end time is met precisely by the automatic time-step control. *End Time* is only active if the specified value is larger than the specified *Start Time.*
Start Time Step	The size of the first time-step after solution initialization if the *Start Time* is set to `0.0`.
Min. Time Step	The lower limit on the time-step size. Exceptionally, the time-step size can be reduced below this value in order to meet the *End Time* target. The upper limit on the time-step size is specified in the Implicit Unsteady solver.
[Up, Down] Max. Time Step Change Factor	The maximum ratio of the current targeted time-step to the previous time-step. You set this ratio to avoid large jumps in the time-step size. The first value applies to an increasing time-step size; the second value applies to a decreasing time-step size. If the maximum time-step change factor is exceeded, the targeted time-step size is cut off at the maximum value permitted by the change factor.
Verbose Output	Additional information appears in the output window during the simulation.

Segregated Time Step Method Properties

The Segregated Time Step methods use the same set of properties, described below. However, the Flow method has one extra property.

Max Condition Limit	The limit for the maximum (smoothed) time step condition. Specifying a value of 0.0 deactivates the method.
Smoothing Steps	The number of smoothing steps that are applied to the time step condition. STAR-CCM+ applies a volume-weighted smoothing algorithm which provides a spatially-balanced distribution of the time-step condition field. This smoothing effectively evens out large peaks.
Use Raw Courant Number	Does not perform unsteady term weighting, but instead considers the raw Courant Number. This property applies to the Flow method only.

Artificial Compressibility

Typically, for casting simulations, the gas phase is considered to be compressible and the ideal gas law is applied. The melt is a liquid and has a constant density. Therefore, the multiphase mixture is not compressible throughout the entire computational domain. This inhomogeneity results in regions with finite and infinite speeds of sound, which can lead to numerical instabilities and convergence problems.

To overcome such robustness issues caused by combining a constant density VOF melt phase with an ideal gas VOF gas phase, the artificial compressibility method can be applied. The artificial compressibility for the VOF melt phase ensures a compressible mixture in the entire fluid domain.

There are two options for defining the artificial compressibility of the melt:

- Pressure Dependent Density
- Pressure and Temperature Dependent Density

Pressure Dependent Density

If the melt density does not depend on temperature, such as for isothermal simulations, the density is defined as:

$$\rho_{\mathrm{Melt}} = \max\left(\rho_{\mathrm{Melt}_{min}}, \rho_{\mathrm{Melt}_{ref}} + \frac{d}{dp}\left(\rho_{\mathrm{Melt}}\right) \cdot p\right) \tag{1448}$$

where:

Symbol	Description	Value
ρ_{Melt}	Density of the melt	
$\rho_{\mathrm{Melt}_{min}}$	Minimum melt density	$1kg/m^3$
$\rho_{\mathrm{Melt}_{ref}}$	Reference melt density that is linearly interpolated from the temperature-dependent density table of the Casting Material Database at a temperature that closely matches the Casting Temperature.	
$\frac{d}{dp}\left(\rho_{\mathrm{Melt}}\right)$	Pressure derivative of the melt density	$10^{-5}Pa$
p	Relative pressure	

and

$$\frac{\partial}{\partial T}\left(\rho_{\mathrm{Melt}}\right) = 0$$

Pressure and Temperature Dependent Density

If the melt density depends on both pressure and temperature, the density is given as:

$$\rho_{\mathrm{Melt}} = \max\left(\rho_{\mathrm{Melt}_{min}}, \rho_{\mathrm{Melt}}(T) + \frac{d}{dp}\left(\rho_{\mathrm{Melt}}\right) \cdot p\right) \tag{1449}$$

where:

Symbol	Description	Value
ρ_{Melt}	Density of the melt	
$\rho_{\mathrm{Melt}_{min}}$	Minimum melt density	$1kg/m^3$
$\rho_{\mathrm{Melt}}(T)$	Temperature-dependent density of the melt that is linearly interpolated from the Casting Material database.	
$\frac{d}{dp}\left(\rho_{\mathrm{Melt}}\right)$	Pressure derivative of the melt density	$10^{-5}Pa$
p	Relative pressure	

and

$$\frac{\partial}{\partial T}\left(\rho_{\mathrm{Melt}}\right)) \neq 0$$

is computed from the tabular melt density values of the Casting Material Database.

The Casting User Interface

The Casting User Interface (CUI) is a dedicated user interface specifically designed to guide you through the setup of casting simulations. It is intuitive to use and easy to learn.

The Casting User Interface lets you set up:

- Physical models
- Material models
- Initial conditions
- Boundary conditions
- Interfaces
- Visualization and reports

The currently supported casting processes are:

- Investment casting
- Sand casting
- Permanent die casting

When you have prepared the geometry and completed the meshing, you can start using the CUI. When you use the CUI to set up the casting simulation, the STAR-CCM+ simulation tree is populated and updated in the background. Making a selection or setting in the CUI can result in one or more actions in the simulation tree.

It is strongly recommended that you perform the entire casting setup in the CUI. Do not toggle between the simulation tree and the CUI. The CUI updates settings in the simulation tree. However, selections and entries from the simulation tree are not transferred back to the CUI.

Saving and Restoring the Casting User Interface Settings

After you have made model selections and settings in the CUI and have saved the simulation file, an XML file is automatically saved to the local working directory where the simulation file resides. The naming convention of the XML file is `<simfilename>.cast.xml` where simfilename is replaced with the name of the sim file. The XML file contains all of the settings that you have specified in the CUI.

The next time that you open the simulation file `<simfilename>.sim` and start the CUI, the existing XML file <simfilename>.cast.xml is read. The CUI opens with the previously stored settings.

Current Limitations

- The table editor only supports a maximum of 20 rows.
- It is not possible to change the units of the input values for tables.
- Variable values in the form of field functions are not available.

Contents:

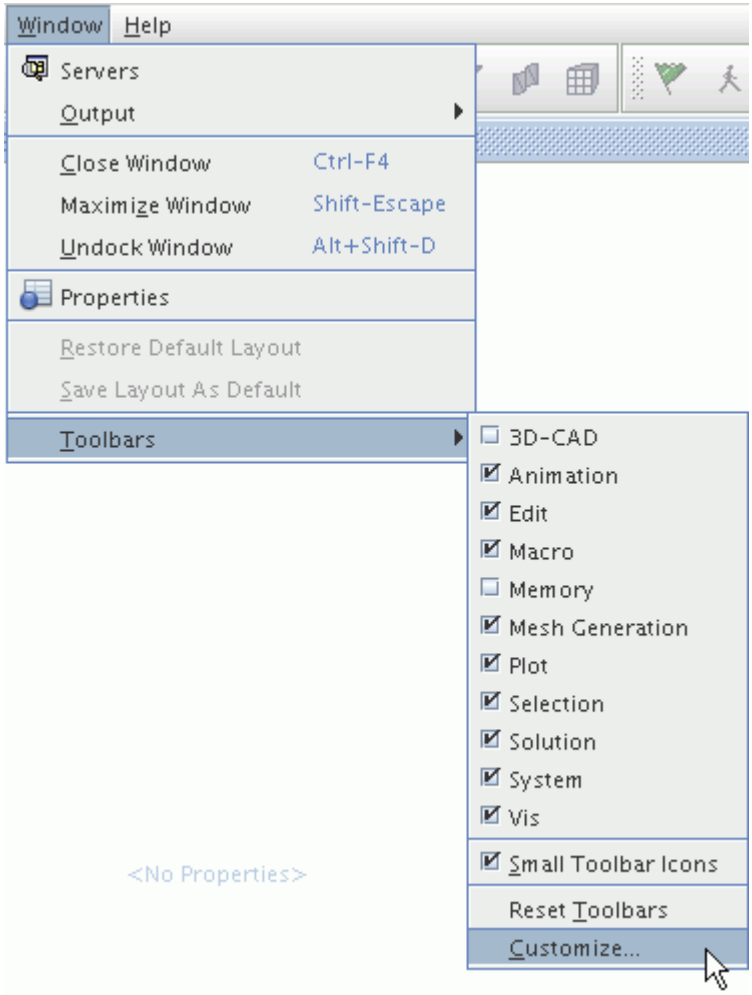
Accessing the Casting User Interface
Casting User Interface Reference

Accessing the Casting User Interface

You must add the Casting User Interface icon to the STAR-CCM+ toolbars.

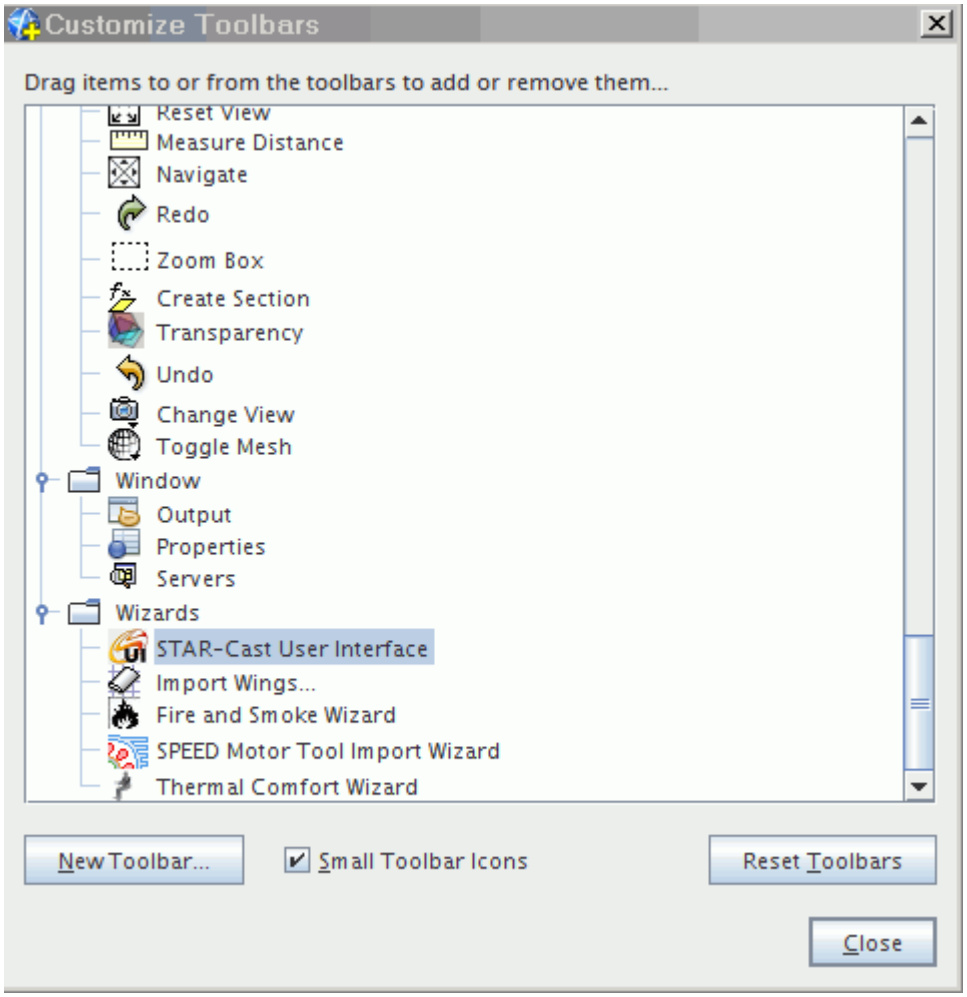
To make the Casting User Interface (CUI) accessible in STAR-CCM+:

1. In the STAR-CCM+ menu, select **Window** > **Toolbars** > **Customize.**



2. Create a new toolbar to contain the CUI icon.
   a) In the *Customize Toolbars* dialog, click **New Toolbar...**
   b) In the *New Toolbar* dialog, enter an appropriate name for the new toolbar, for example, Casting.
   c) Click **OK**.

      The new toolbar appears on the right next to the existing toolbars at the top of the STAR-CCM+ workspace. This can take a few seconds.
3. In the *Customize Toolbars* dialog, under the **Wizards** node, select **STAR-Cast User Interface**.



4. Drag the STAR-Cast User Interface icon onto the newly created casting toolbar at the top of the STAR-CCM+ workspace and drop it at the appropriate position.

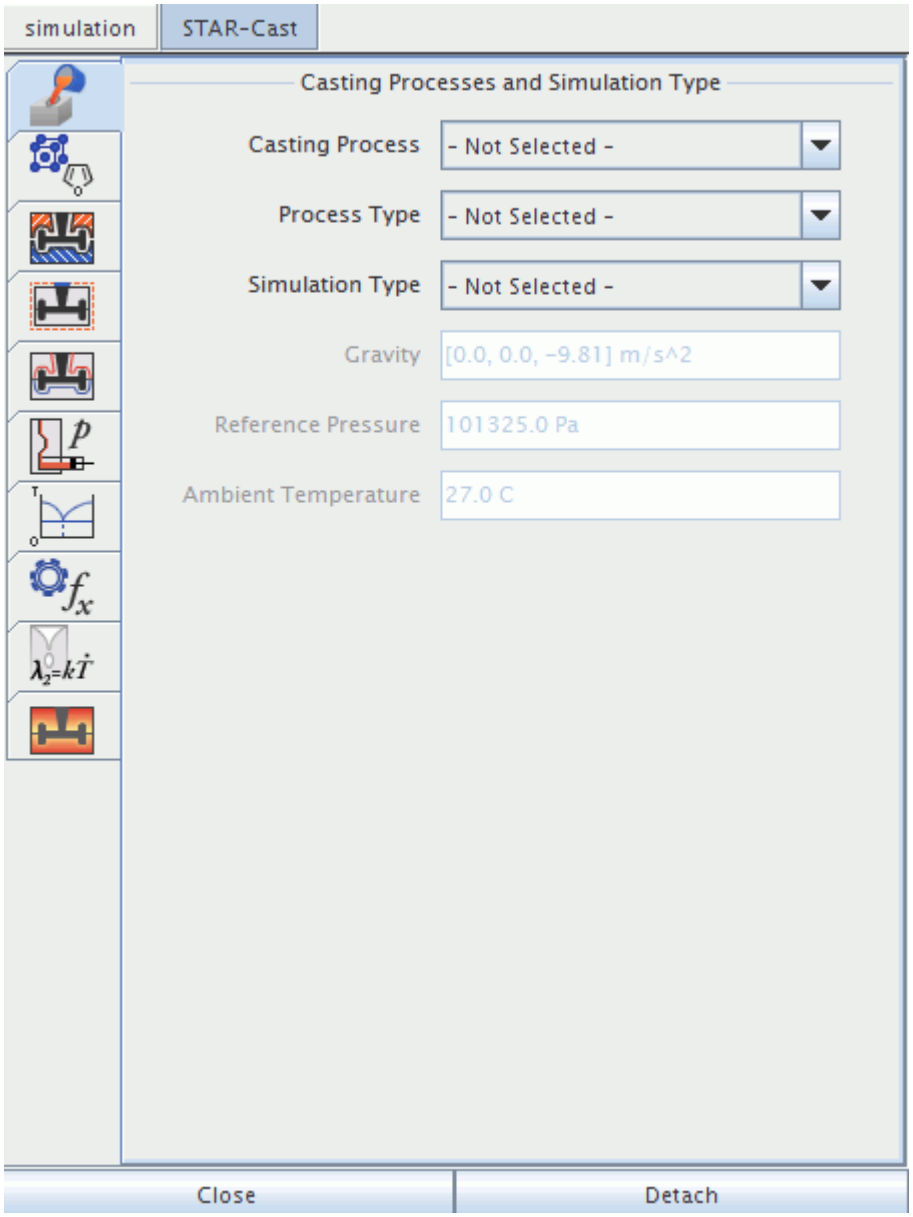
   The icon is added to the toolbar at the specified location.

5. In the *Customize Toolbars* dialog, click **Close**.

To open the Casting User Interface:

- Click the STAR-Cast User Interface icon in the **Casting** toolbar.

  The CUI opens with the *Casting Processes and Simulation Type* panel as an embedded panel in the *simulation* window covering the simulation tree.



To detach the CUI from the simulation tree, click the **Detach** button. This docks the CUI to the right-hand side of the STAR-CCM+ workspace. Once the CUI is detached, an **Attach** button appears that allows you to re-attach the CUI to the simulation tree. Use the **Close** button to close the CUI.

Casting User Interface Reference

The Casting User Interface consists of eleven different panels. Each panel covers a step in the setup of a casting simulation.

You walk through the panels from top to bottom. The order is important because the CUI updates and adjusts the possible settings depending on previous selections. This behavior is reflected by the look of the icons and tabs: if a required selection was not made in a previous panel, the icons and/or the tabs remain grayed out.

In the panels, you enter the settings required for the casting simulation. For most panels, the procedure is:

- Specify the conditions for a particular setup step, for example, the boundaries, in the *Settings* tab.

  A condition is a set of properties that can be saved under a specific name.
- Assign the saved condition to the item to be defined in the *Selections* tab, for example, to the boundary surface.

The advantage of this two-step approach is that different conditions can be saved for later use and can easily be exchanged in the *Selections* tab.

Contents:

The Condition Settings/Import Export Panel
The Casting Processes and Simulation Type Panel
The Material Data Panel
The Regions Panel
The Boundaries Panel
The Interfaces Panel
The Process Parameters Panel
The Physical Models Panel
The Solvers Panel
The Criteria Functions Panel
The Post Processing Panel

The Condition Settings/Import Export Panel

In the *Condition Settings Import/Export* panel, you can store or retrieve condition settings that you make in the CUI. These settings are stored in the `cast.xml` file. Use the *Condition Filter* section to select either all or certain conditions for export or import.

- To import condition settings from a `cast.xml` file, click **Import**.
- To export condition settings to a `cast.xml` file, click **Export**.

Condition Filter	The following condition settings can be selected for export or import: • Regions • Interfaces • Physical Models • Stopping Criteria • Boundaries • Process Parameters • Solver Settings
Conditions	Displays all the conditions that were either defined manually during the current setup and/or conditions that were imported from the `cast.xml` file.

The Casting Processes and Simulation Type Panel

In order to use the Casting User Interface, you must select the casting process, the process type, and the simulation type.

You must make the first three selections:

- Casting Process
- Process Type
- Simulation Type

The selections that you make here affect the options that are available in the following panels. Not all of the *Process Type* options listed in the following table are available for all *Casting Process* options.

Casting Process	Selects the casting process to be used.	
	Investment	Sets up an investment casting process.
	Sand	Sets up a sand casting process.
	Permanent Die	Sets up a permanent die casting process.
Process Type	Selects the process type to be used for the previously selected casting process. Note that not all process type options are available for each casting process. Which process type option becomes available depends on the selection that you have made under *Casting Process*. This option invokes the creation of a physics continuum for the melt in the simulation tree. Various physics models are activated, including: • Casting • VOF Multiphase The Cell Quality Remediation model is also activated. Selects the implicit unsteady segregated solver for a transient simulation. The flow is laminar for all process types.	
	Bridgman	Sets up a Bridgman type casting process. Activates a translational motion model.
	Centrifugal	Activates a rotational motion model.
	Gravity	Sets up a gravity type casting process.
	High Pressure Die	Sets up a high pressure die casting process. Activates a morphing motion model.

	Tilt	Activates a rotational motion model.
Simulation Type	Selects the simulation type to be used for the casting process.	
	Filling and Solidification	Sets up a simulation including the pouring and the solidification process of the melt. Activates the segregated energy solver.
	Isothermal Filling	The solidification process of the melt is not considered. Energy is not solved for. Note that the *Isothermal Filling* option is not available for the *Bridgman* process type option.
	Pure Thermal	Sets up a thermal solidification analysis of the melt. Activates the segregated energy solver.
Gravity	Specification of the gravity vector.	
Reference Pressure	Specification of the reference pressure value.	
Casting Temperature	For the *Simulation Type* **Isothermal Filling,**the *Casting Temperature* sets the continuum temperature of the **Melt** continuum. The simulation is performed at this temperature level. This means that the thermodynamic properties of the melt and the gas at this temperature level are applied. To obtain the constant material properties for the melt at this temperature, the thermodynamic properties provided in the Casting Material Database are evaluated. For the *Simulation Type* **Filling and Solidification**, the *Casting Temperature* is the default inflow boundary temperature of the melt for the boundary types **Velocity Inlet Boundary**, **Mass flow Inlet Boundary**, and **Pressure Outlet Boundary**. This value can be overwritten by respective entries in the *Boundaries* tab. For the *Simulation Type* **Pure Thermal**, the *Casting Temperature* is the default initial temperature of the melt. This value can be overwritten by respective entries in the *Region* tab.	
Ambient Temperature	Effective radiation temperature of the environment. This temperature is needed for the gray thermal radiation model. If *Simulation Type* is **Isothermal Filling**, this option is deactivated.	

The Material Data Panel

In the *Material Data* panel you select the materials used for the casting simulation. The selection encompasses materials for the melt as well as materials for the solid molds.

Use the ACCESS casting material database to select the materials.

To select casting materials:

- Click **Launch Database...**

  This opens the casting material database. For more information, Working With the Casting Material Database.

The selected materials are displayed in the scroll list of the CUI in the Material Data Panel. The selected materials also automatically go to the **Tools** > **Material Databases** node of the simulation tree. A new material database called **Selected_Casting_Material.mdb** is created under that node.

You can import previously selected materials data from an existing casting material XML file, which has been created by the ACCESS casting material database.

To import an existing casting material database XML file:

- Click **Import XML...** and select the file that you would like to import.

The Regions Panel

In the *Regions* panel, you set the initial conditions of the simulation and assign the previously selected materials to the regions. The regions exist already as they are created during the geometry preparation and the mesh generation process.

The *Regions* panel consists of two tabs: the *Initial Conditions* and the *Selections* tab. You assign the initial conditions that you define in the *Initial Conditions* tab to the existing regions in the *Selections* tab.

Regions Initial Conditions Tab

In the *Initial Conditions* tab, you define the initial conditions for either the fluid or the solid regions and save them under a user-defined name. These initial conditions can then be assigned to a region in the *Regions* table in the *Selections* tab.

Initial Conditions	Lists the initial conditions that are already defined.
Name	User-defined name under which the initial condition is saved.
Type	Specifies whether the initial condition is related to a fluid or a solid region.

Initial Conditions Properties

Properties	Specifies the variable for which the initial condition is to be set.	
	Temperature	Sets the initial temperature. The units are C for degrees Celsius, K for kelvins, and F for Fahrenheit.
	Velocity	Sets the initial velocity components with respect to the laboratory coordinate system in m/s.

	Volume Fraction	Sets the initial volume fraction, that is the fraction of the local cell volume which a phase occupies. A value of 0.0 means that nothing of that phase is contained in the cell whereas a value of 1.0 means that the cell is completely filled with the phase.
Options	The properties can be specified in different formats.	
	Constant	Sets a constant value of the property. You specify this value in the corresponding *Values* cell of the *Properties* table.
	Gas	Sets the volume fraction of the gas phase to 1.0 in all cells of the solution domain. The volume fraction of the melt phase is set to 0.0.
	Melt	Sets the volume fraction of the melt phase to 1.0 in all cells of the solution domain. The volume fraction of the gas phase is set to 0.0.
	Melt Under X-Axis	Sets the melt volume fraction to 1.0 in all cells whose position in x-direction (with respect to the laboratory coordinate system) is below a specified value. You specify this value in the corresponding *Values* cell of the *Properties* table. The melt volume fraction of the remaining cells is set to 0.0.
	Melt Under Y-Axis	See **Melt Under X-Axis**, but use the y-direction instead.
	Melt Under Z-Axis	See **Melt Under X-Axis**, but use the z-direction instead.
	Table (Time)	Sets the property value in the form of a table as a function of time.
	Table (XYZ)	Sets the property value in the form of a table as a function of coordinates with respect to the laboratory coordinate system.

Table Options	If either Table (Time) or Table (XYZ) has been selected, the *Values* cell is greyed out and a table editor appears. You can use the units selector to set the units of the variable. For **Temperature**, the table values adjust when the unit changes. The table can be input in two different ways.	
	Tabulate	Enter the tabular values manually in the table editor.
	Import .csv	Import the tabular values from a file, which contains comma separated values.
	Export .csv	The tabular values can be exported to a file containing comma separated values.

Note: If you selected **Isothermal Filling** as *Simulation Type* in the *Casting Processes and Simulation Type* panel, only the *Temperature* option set to **Constant** is valid. The other two options **Table (Time)**and **Table (XYZ)**can be selected, but they have no effect on the simulation.

Regions Selections Tab

Regions	Contains the regions that have been created during the geometry preparation and the meshing process.
Materials	Selection of the previously selected materials from the casting material database. Selecting a liquid casting material invokes: • Creation of the corresponding liquid material within the Eulerian phase of the melt continuum previously created. • Assigning of melt continuum to the selected region in the table. • Activation of **Density Limiting** for the ideal gas model of the Eulerian gas phase. The *Cut-off pressure* is set to a value of 20% of the specified reference pressure. • Activation of Artificial Compressibility for the melt phase. For more information, see Artificial Compressibility. The type of artificial compressibility depends on the *Simulation Type* that is set in the *Casting Processes and Simulation Type* panel. For **Isothermal Filling**, Pressure Dependent Density is applied. For the **Pure Thermal** and **Filling and Solidification**, Pressure and Temperature Dependent Density is applied. • Activation of the Surface Tension model (see Surface Tension Formulation). • Activation of the Interface Momentum Dissipation model (see Using the Interface Momentum Dissipation Model). Selecting a solid mold material invokes:

	• Creation of a solid physics continuum containing the solid mold material. • Grey thermal radiation model is activated. Assigning of solid mold continuum to the selected region in the table.
Conditions	Assigns the initial conditions to the selected region. The available conditions must be set before in the *Settings* tab. Depending on the **Type** selection that you have made in the *Settings* tab, only a suitable selection of either a fluid or a solid initial condition is available.

The Boundaries Panel

In the *Boundaries* panel, you specify the boundary conditions.

The panel consists of two tabs: the *Settings* tab and the *Selections* tab. First, you define the boundary conditions in the *Settings* tab. Then, you can assign these conditions to the actual boundary surfaces in the model.

Boundaries Settings Tab

Boundary Conditions	Lists the boundary conditions that are already defined.	
Name	User-defined name under which the boundary condition is saved.	
Type	Specifies the type of boundary condition to be used. There are different boundary condition types available depending on whether it is a fluid or a solid boundary. Apply boundary conditions termed *Fluid* to boundary surfaces encompassing a fluid. Apply boundary conditions termed *Solid* to boundary surfaces encompassing a solid. The boundary conditions have different properties	
	Adiabatic Fluid Wall Boundary	Boundary, which is impenetrable for fluid flow and energy.
	Adiabatic Solid Wall Boundary	Boundary, which is impenetrable for energy.
	Fixed Temperature Fluid Wall Boundary	Boundary, on which a fixed temperature is specified.
	Fixed Temperature Solid Wall Boundary	Boundary, on which a fixed temperature is specified.
	Fluid Wall Boundary	Convective boundary, for which an ambient temperature outside of the fluid domain is specified. The heat flux through the wall and the temperature at the wall are computed by use of the specified heat transfer coefficient (W/m2 s).

		The wall porosity model can be applied at this boundary type.
	Heat Flux to Fluid Wall Boundary	Boundary, on which a heat flux in W/m2 is specified. The wall porosity model can be applied at this boundary type.
	Heat Flux to Solid Wall Boundary	Boundary, on which a heat flux in W/m2 is specified. The wall porosity model can be applied at this boundary type.
	Mass Flow Inlet Boundary	Boundary, for which the inflow mass flow rate (kg/s) is specified. The inflow temperature and the volume fraction (dimensionless values between 0 and 1) are also specified.
	Pressure Outlet Boundary	Outflow boundary for which the pressure is specified. Specified temperature and volume fraction are applied under inflow conditions, that is when recirculation occurs.
	Solid Wall Boundary	Convective boundary, for which an ambient temperature outside of the solid domain is specified. The heat flux through the wall and the temperature at the wall are computed by use of the specified heat transfer coefficient (W/m2 s). The emitted radiation can be specified through the emissivity parameter (dimensionless value between 0 and 1).
	Symmetry Plane Boundary	This boundary condition can be used at symmetry planes of geometries. It has no properties.
	Velocity Inlet Boundary	Specify the normal inlet velocity components (m/s).

Boundary Conditions Properties

Properties		
	Contact Angle	The contact angle in degrees between the wall and the melt. For

		details on the definition of the contact angle, see What Is a Contact Angle?.
	Porous Inertia Coefficient	Coefficient of the wall porosity model. It corresponds to the α coefficient in Eqn. (848). See Wall Porosity Formulation for details on the definition of the porous inertia coefficient.
	Porous Viscous Coefficient	Coefficient of the wall porosity model (m/s). It corresponds to the β coefficient in Eqn. (848). See Wall Porosity Formulation for details on the definition of the porous inertia coefficient.
	Slip Condition	This property has two options: **No**. A no-slip wall is applied. The fluid velocity at the wall equals the wall velocity. The wall can be at rest or you can specify a tangential wall velocity in the form of a velocity vector with respect to the laboratory coordinate system. **Yes**. A slip wall is applied. The fluid has a different velocity than the wall velocity.
	Temperature	Temperature at the boundary.
	Emissivity	Ratio of the power that a body emits to the power it would emit as a black body at the same temperature (dimensionless). Emissivity is a parameter of the grey thermal radiation model.
	Ambient Temperature	Assumed temperature outside of the computational domain.
	Heat Transfer Coefficient	Heat transfer coefficient at the wall (W/m2s) used to compute the temperature at the wall.
	Heat Flux	Heat flux at the boundary (W/m2)
	Mass Flow Rate	Fluid mass flow rate (kg/s).
	Volume Fraction	The volume fraction of a phase is the fraction of the local cell volume

		which that phase occupies. A value of 0.0 means that nothing of that phase is contained in the cell whereas a value of 1.0 means that the cell is completely filled with the phase.
Options	The properties can be specified in different formats.	
	Constant	Sets a constant value of the property. Specify this value in the corresponding *Values* cell of the *Properties* table.
	Table (Time)	Sets the property value in the form of a table as a function of time.
	Table (XYZ)	Sets the property value in the form of a table as a function of coordinates with respect to the laboratory coordinate system.
	Material DBS	Takes the property value from the Casting Material Database.
	Gas	Sets the volume fraction of the gas phase to 1.0 and the melt volume fraction to 0.0.
	Melt	Sets the volume fraction of the melt phase to 1.0 and the gas volume fraction to 0.0.
Table Options	If either **Table (Time)** or **Table (XYZ)** is selected, the *Values* cell is greyed out and a table editor appears. You can use the units selector to set the units of the variable. For **Temperature**, the table values adjust when the unit changes. The table can be input in two different ways.	
	Tabulate	Enter the tabular values manually in the table editor.
	Import .csv	Import the tabular values from a file, which contains comma separated values.
	Export .csv	The tabular values can be exported to a file containing comma separated values.

Boundaries Selections Tab

Boundaries	Contains the boundary surfaces that have been created during the meshing process.

Conditions	Assigns the boundary conditions to the selected boundary surface. The available conditions must have been set before in the *Settings* tab. Depending on the **Type** selection that you have made in the *Settings* tab, only a suitable selection of either a fluid or a solid boundary condition is available. After a selection has been made, the corresponding properties are shown at the bottom of the panel.
Type	Specifies the type of boundary condition to use. There are different boundary condition types available depending on whether it is a fluid or a solid boundary.

The Interfaces Panel

In the *Interfaces* panel, you specify the interfaces between the fluid and the solid regions. At the interfaces, variable values between fluid and solid regions are exchanged in the course of the iterations of the solvers.

First, you define the interface conditions in the *Settings* tab. Then, you assign these conditions to the actual interface surfaces in the model in the *Selections* tab. The properties of the assigned conditions are then shown in the Properties table of the *Selections* tab.

Note that the settings made here in the *Interfaces* panel of the CUI, invoke simulation tree settings both on the interface nodes on the regions level as well as on the **Interface** node itself.

Interfaces Settings Tab

Conditions	Lists the interface conditions that are already defined.	
Name	User-defined name under which the interface condition is saved.	
Type	Specifies the type of interface condition to use. There are different interface condition types available. The interface conditions have different properties.	
	Fluid to Solid	Defines a solid-fluid interface.
	Solid to Solid	Defines a solid-solid interface.
	Flow Through	Defines an internal interface that permits fluid and energy flow through it.

Interfaces Properties

Properties		
	Contact Angle	The contact angle in degrees between the interface and the melt. For details on the definition of the contact angle, see What Is a Contact Angle?.

	Porous Inertia Coefficient	The coefficient of the wall porosity model. It corresponds to the α coefficient in Eqn. (848). See Wall Porosity Formulation for details on the definition of the porous inertia coefficient.
	Porous Viscous Coefficient	The coefficient of the wall porosity model (m/s). It corresponds to the β coefficient in Eqn. (848). See Wall Porosity Formulation for details on the definition of the porous inertia coefficient.
	Slip Condition	This property has two options: **No**. A no-slip condition is applied at the interface. The fluid velocity at the interface equals the interface velocity. The interface can be at rest or you can specify a tangential interface velocity in the form of a velocity vector with respect to the laboratory coordinate system. **Yes**. A slip condition is applied at the interface. The fluid has a different velocity than the interface velocity.
	Ambient Temperature	Ambient temperature for use with convection condition thermal specification.
	Heat Transfer Coefficient	Heat transfer coefficient at the interface (W/m2K) used to compute the temperature at the interface. In the simulation tree, this value is transformed into a contact resistance, which is the inverse of the heat transfer coefficient in m2K/W.
Options	The properties can be specified in different formats.	
	Constant	Sets a constant value of the property. Specify this value in the corresponding *Values* cell of the *Properties* table.

	Table (Time)	Sets the property value in the form of a table as a function of time.
	Table (XYZ)	Sets the property value in the form of a table as a function of coordinates with respect to the laboratory coordinate system.
	Table (Temp.)	Set the property value in the form of a table as a function of temperature.
	Material DBS	Takes the property value from the Casting Material Database.
Table Options	If either **Table (Time)** or **Table (XYZ)** has been selected, the *Values* cell is greyed out and a table editor appears. You can use the units selector to set the units of the variable. For **Temperature**, the table values adjust when the unit changes. The table can be input in two different ways.	
	Tabulate	Enter the tabular values manually in the table editor.
	Import .csv	Import the tabular values from a file, which contains comma separated values.
	Export .csv	The tabular values can be exported to a file containing comma separated values.

Interfaces Selections Tab

Interfaces	Contains the interfaces that have been created during the meshing process.
Conditions	Assigns the interface conditions to the selected interface. The available conditions must have been set before in the *Settings* tab. Depending on the **Type** selection that you have made in the *Settings* tab, only a suitable selection of the interface types is available. For example, at a fluid-solid interface you are only able to select an interface condition of the type **Fluid to Solid**. After a selection has been made, the corresponding properties are shown at the bottom of the panel.
Type	Specifies the type of interface to be used. There are different interface condition types available depending on whether it is a solid-fluid interface, a solid-solid interface or a flow-through (internal) interface.

The Process Parameters Panel

In the *Process Parameters* panel, you specify process parameters, which are specific to the process type previously selected in the *Casting Process and Simulation Type* panel.

The panel consists of two tabs: the *Settings* tab and the *Selections* tab. First, you define the process parameters in the *Settings* tab. Then, you assign the process parameter conditions to the selected regions in the *Selections* tab.

Process Parameters Settings Tab

Conditions	Lists the process parameter conditions that are already defined.	
Name	User-defined name under which the process parameter condition is saved.	
Type	Specifies the type of process parameter condition to be used. There are different process parameter condition types available. The process parameter conditions have different properties	
	Bridgman Casting	Directed solidification and single crystal casting process where the cast part moves out of an oven. Activates a translational motion model.
	Tilt Casting	To fill the mold, the entire apparatus is tilted according to a tilting curve. Activates a rotational motion model.
	Centrifugal Casting	To fill the mold, the entire apparatus or the mold only is rotated according to a rotational curve. Activates a rotational motion model.
	High Pressure Die Casting	To fill the mold, the piston head is morphed according to a specified **Piston Direction** and **Piston Speed**. When the pressure on the piston head exceeds the specified **Switching Pressure**, the **Piston Speed** is modified to keep the pressure value equal to the specified **Holding Pressure**. Activates a morphing motion model.

Process Parameters Properties

Properties	Process parameter properties.	
	Translational Velocity	The translational velocity for the Bridgman casting process. This is specified in the form of a vector with respect to the laboratory coordinate system.
	Axis Direction	The direction of the z-axis of a new local coordinate system around which the rotation is performed. This is defined with respect to the laboratory coordinate system.
	Axis Origin	The origin of the new local coordinate system for the rotation with respect to the laboratory coordinate system.
	Rotational Rate	Specify the rotation rate in rad/s.
	Gas Treatment	The gas treatment for the high pressure die casting process.
	Holding Pressure	Maximum pressure on the piston head boundary that is maintained after the **Switching Pressure** is exceeded to compensate for material shrinkage.
	Switching Pressure	Maximum pressure on the piston head boundary at which the **Piston Speed** is modified to keep the pressure value equal to the specified **Holding Pressure**.
	Piston Direction	The axis direction for the piston head motion with respect to the laboratory coordinate system.
	Piston Speed	The speed at which the piston head moves to fill the mold.
Options	The properties can be specified in different formats.	
	Constant	Sets a constant value of the property. Specify this value in the corresponding *Values* cell of the *Properties* table.
	Table	Sets the property value in the form of a table as a function of time.

	Normal	Sets the **Ideal Gas** model for air.
	Simple	Sets the **Constant Density** model for air. Source terms within the cavity keep the relative pressure below 0 Pa and allow air to leave and to enter the domain based on the pressure distribution.
	X-axis, **Y-axis**, **Z-axis**	Sets the **Piston Direction** to the x-, y-, or z-axis of the laboratory coordinate system.
	Table (Time)	Sets the **Piston Speed** value in the form of a table as a function of time.
	Table (Position)	Sets the **Piston Speed** value in the form of a table as a function of the distance between the piston head and its starting position.
Table Options	If **Table**, **Table (Time)**, or **Table (Position)** is selected, the *Values* cell is greyed out and a table editor appears. The table can be input in three different ways.	
	Tabulate	Enter the tabular values manually in the table editor.
	Import .csv	Import the tabular values from a file, which contains comma separated values.
	Export .csv	The tabular values can be exported to a file containing comma separated values.

Process Parameters Selections Tab

Depending on the *Process Type* selection made in the *Casting Process and Simulation Type* panel, the tab appears differently. If the *Process Type* is **Gravity**, the *Selections* tab is empty and no assignments are required. If the *Process Type* is either **Bridgman**, **Centrifugal**, or **Tilt**, a region selection dialog appears that allows selection of the regions to which the process parameters are applied. For a **High Pressure Die** process a piston head selection dialog allows selection of the boundary to which the process parameters are applied.

The **Conditions** selector allows selection of the process parameter conditions that have been previously defined in the *Settings* tab. The assigned properties are shown at the bottom of the panel.

The Physical Models Panel

In the *Physical Models* panel, you define and select the parameters and options for optional physical models as part of the casting simulation.

These optional models are:

- Melting and Solidification Flow Stop Model

  See Flow Stop Mass Compensation
- Mushy Zone Permeability Model

  See Mushy Zone Permeability Model
- Slurry Viscosity

  See Slurry Viscosity Model
- Surface Tension

  See Surface Tension
- Macro Porosity Model

  See Using the Macro Porosity Model to Detect Shrinkage-Related Defects
- Contact Time

  See Using the Contact Time Model

The *Physical Models* panel consists of two tabs: the *Settings* tab and the *Selections* tab. First, you define the physical model in the *Settings* tab. Then, you assign the physical models conditions to the selected physical models in the *Selections* tab.

Physical Models Settings Tab

Conditions	Lists the physical models conditions that are defined.	
Name	User-defined name under which the physical model condition is saved.	
Type	Specifies the type of physical model condition to be used. There are different process physical model types available. The physical model conditions have different properties	
	Melting and Solidification	Selects the Melting and Solidification Flow Stop model.
	Mushy Zone	Selects the Mushy Zone Permeability model.
	Slurry Viscosity	Selects the Slurry Viscosity Model .
	Liquid Residence Time	Selects the Liquid Residence Time criteria function (Liquid Residence Time).
	Macro Porosity	Selects the Macro Porosity model (Using the Macro Porosity Model to Detect Shrinkage-Related Defects).
	Surface Tension	Selects surface tension. Once you combine this model with the **Melting and Solidification**

		model, the surface tension coefficient σ is modified to approach zero when the melt solidifies, that is when the relative solid fraction α_s^* reaches the flow stop solid fraction α_{stop}. The effective surface tension coefficient σ^* is then calculated as: $$\sigma^* = \sigma\left(0.5 - \frac{\arctan\left[1000\left(\alpha_s^* - \alpha_{stop}\right)\right]}{\pi}\right)$$
	Contact Time	Selects the Contact Time model. Determines the contact time between a gas and a melt phase only in the liquid state. Once the solid volume fraction of the melt reaches the **Solid Volume Fraction Limit** in the simulation, the contact time stops accumulating.

Physical Models Properties

Properties		
	Flow Stop Relative Solid Fraction	Defines the cell solid volume fraction (values between 0 and 1) at which the fluid loses its ability to flow. However, the flow is stopped in the cell only when the total amount of solidified fluid exceeds the Flow Stop Solid Fraction that is specified for the Melting and Solidification Flow Stop model. This property is deactivated for values greater than 1. Default value 1.1.
	Flow Stop Solid Fraction	Specifies when a cell is blocked for the flow. Set the solid fraction to a value between 0 and 1. A typical value is 0.9.
	Max. Viscous Resistance	Sets a maximum level of resistance (kg/m3 s). This setting applies only to the cells with a solid

		volume fraction greater than zero, in all regions to which the respective physics continuum is assigned.
	Use Forchheimer's Term	When activated, STAR-CCM+ includes the contribution of the Forchheimer's term to the flow resistance of the Carman-Kozeny mushy zone model (see Eqn. (2888).
	Secondary Dendrite Arm Spacing	Is used for the calculation of the Carman-Kozeny Shape Factor (m). The shape factor c_s is a function of the secondary dendrite arm spacing d) and a shape constant $c : c_s = c/d^2$. The default value for c is 180.
	Critical Solid Fraction	The critical solid volume fraction α^*_{cr} (see Eqn. (2890)). The default value is 0.27 (for Aluminum alloy A201).
	Max. Solid Volume Fraction	The residence time is accumulated only for solid volume fractions below the specified value.
	Surface Tension	Specifies the value of the surface tension of the melt in N/m.
	Void Pressure	The pressure p_{void} in Eqn. (1424) below which a vapour bubble forms with the liquid phase. The default value is 10000 Pa.
	Solid Volume Fraction Limit	Sets the value of the solid volume fraction at which contact time stops accumulating. Below this value, contact time is accumulated.
Options	The properties can be specified in different formats.	
	Constant	Sets a constant value of the property. You specify this value in the corresponding *Values* cell of the *Properties* table.
	Material DBS	Takes the property value from the Casting Material Database.

	Table (Temp.)	Sets the property value in the form of a table as a function of temperature.
	Use Switching	Toggles the switching option for the Mushy Zone Permeability model on and off. If switching is activated, specify the non-dimensional constant s (see Eqn. (2890)). The default value is 100 because the shape is rather insensitive for values from 80 through 120.

Physical Models Selections Tab

Physical Models	Contains the physical models available.
Conditions	Assigns the physical model conditions to the selected physical model. The available conditions must be set before in the *Settings* tab. Only the relevant physical model types are available, corresponding to the **Type** selection that you made in the *Settings* tab. For example, for the **Melting-Solidification** model you are only able to select a physical model condition of the type **Melting-Solidification**. After a selection has been made, it is shown at the bottom of the panel.
Model	Lists the selected physical model. The properties, options and values are shown underneath.

The Solvers Panel

In the *Solvers* panel, you specify the numerical parameters and stopping criteria for the casting simulation.

The panel consists of three tabs: the *Solver Settings*, the *Stopping Criteria* tab, and the *Process Phases* tab. First, you define the solver settings conditions in the *Solver Settings* tab as well as the Stopping Criteria conditions in the *Stopping Criteria* tab. Then, you assign both of these conditions to a process phase in the *Process Phases* tab.

Solver Settings Tab

Conditions	Scroll list where the defined solver settings conditions are listed by name.
Name	User-defined name under which the solver settings condition is saved.

Solver Settings Properties

Within the *Solver Settings* tab you specify the conditions and values for the following numerical solver parameters:

Time Steps

Controls how you define the time-step size in s. The available methods are:

Automatic	Applies automatic solution control that drives the time-step size according to a range of conditions, which are: **Flow** Monitors the rate of change of absolute pressure. **Volume of Fluid** Monitors the local rate of change of the volume fraction. **Energy** Monitors the local rate of change of the total energy. **Melting/Solidification** Monitors the transport of the phase change interface. For more information, see Automated Time Step Control. The automatic solution control itself as well as the conditions that you activate (**Condition On**) require the setting of additional properties, see Time Step Control Properties.
Constant	Sets a constant value of the time-step size.
Table	Sets the time-step size in the form of a table as a function of time. You can input the table in two different ways using the table editor: **Tabulate** Enter the tabular values manually in the table editor. **Import .csv** Import the tabular values from a file, which contains comma-separated values. **Export .csv** allows you to export tabular values to a file containing comma-separated values.

Under-Relaxation Factors

Control the extent to which the newly computed solution supplants the old solution. The **Solver Off** option toggles the solution of the respective equation on and off.

Pressure URF

Controls the solution update for the pressure solver.

Volume of Fluid URF

Controls the solution update for the Segregated Volume Fraction solver.

Fluid Energy URF

Controls the solution update for the segregated energy solver in the fluid material.

Solid Energy URF

Controls the solution update for the segregated energy solver in the solid material.

Passive Scalar URF

Controls the solution update for the passive scalar solver.

Melting/Solidification URF

Controls the solution update for the Melting and Solidification.

Mushy Zone URF

Controls the solution update for the Mush Zone Permeability Model.

Slurry Viscosity URF

Controls the solution update for the Slurry Viscosity Model.

Stopping Criteria Tab

Conditions	Lists the stopping criteria conditions that are already defined.
Name	User-defined name under which the stopping criteria condition is saved.

Stopping Criteria Properties

Maximum Physical Time	Specify the maximum physical time for the process phase in s.	
Maximum Inner Iteration	Specify the maximum number of inner iterations for each unsteady time-step.	
Minimum Inner Iteration	Specify the minimum number of inner iterations for each unsteady time-step.	
Mass Error Rate	Stopping criterion for the inner iterations of the activated solvers based on the relative mass error rate. Activating this stopping criterion lets STAR-CCM+ automatically determine the number of inner iterations that are needed to reach a minimum accuracy limit for mass conservation. For a definition of the relative mass error rate, see Casting Field Functions.	
	Off	The stopping criterion is deactivated.
	Minimum	The stopping criterion is activated. Provides a minimum limit for the relative mass error rate. You choose the minimum limit, for example, by considering the variation in weight of the casted parts. If the weight varies, for instance, by 3 % and the total physical simulation time is, for example, 8 s, a value like 3%/8s = 0.375 %/s is appropriate for this example.
Energy Criterion	Stopping criterion for the inner iterations of the energy solver based on an asymptotic limit for the energy equation residual.	
	Off	The stopping criterion is deactivated.

	Asymptotic	Activates the stopping criterion and lets STAR-CCM+ automatically determine the number of inner iterations that are needed to reach an asymptotic limit for the energy equation residual.	
		Min-Max Limit	The maximum difference between the energy equation residual samples.
		Number of Samples	The number of samples to be considered for the asymptotic limit.

Process Phases Tab

Process Phase	Contains the process phases. You need to define a process phase first by entering a user-defined name in the **Process Phase Name** text field and clicking **Insert**. A process phase is used to divide the casting simulation into several distinct physical time-dependent phases such as, for example, the filling phase and the solidification phase. Each process phase can have different requirements regarding the time-step size and the numerical solver parameters since the physical processes being simulated in each phase work in different time-scales. For example, the time-step size for the solidification process can be significantly larger than for the filling process phase. When a filling process phase switches to a solidification process phase in the course of the simulation, the CUI executes the following STAR-CCM+ actions in the background: • Stop the simulation at the user-defined point in time where the filling process phase switches to the solidification process phase. • Generate an internal table (X,Y,Z) for temperature, VOF melt, solidification time (liquid residence time if activated) and pressure. • Extract values. • Clear the solution for all melt physics regions. • Set velocity 0,0,0 as initial condition. • Assign XyzInternalTable values for temperature, VOF melt, solidification time (liquid residence time) and pressure. • Continue to run. To make sure that the CUI identifies the solidification phase correctly, the solvers for pressure, velocity, and volume of fluid must be turned off by activating the **Solver Off** option in the *Solver Settings* tab.
Solver Settings	Assigns the solver settings conditions to the selected process phase. The available conditions must have been set before in the *Solver Settings* tab. After a selection has been made, it is shown at the bottom of the panel.
Stopping Criteria	Assigns the stopping criteria conditions to the selected process phase. The available conditions must have been set before in the *Stopping*

	Criteria tab. After a selection has been made, it is shown at the bottom of the panel.
Process Phase Name	Specify a user-defined name for the process phase that you want to define.

You can use a Java macro to start the simulation using the specified process phases. Save the process phase settings to a Java macro by clicking **Export...** Start the simulation by clicking **Run**. This plays the previously saved process phase Java macro.

The Criteria Functions Panel

In the *Criteria Functions* panel, you specify the critical temperatures at which the criteria functions are evaluated.

For a complete list of the available criteria functions and their definitions, Criteria Functions Definitions.

Criteria Temperatures	Lists the critical temperatures that you have specified. To define critical temperatures, enter a temperature value in the **Temperature** text field and click **Add**.
Units	The critical temperature units. Currently, degrees Celsius is the only option available.
Temperature	A critical temperature between the solidus and liquidus temperatures at which Criteria Functions are evaluated.

The Post Processing Panel

In the *Post Processing* panel, you specify which results of the casting simulation you want to visualize and report during the run.

The *Post Processing* panel consists of three tabs: the *Global* tab, the *Probes* tab, and the *Transient* tab.

Post Processing Global Tab

Cavity Plausibility Checks	Activating an option invokes the creation of one or several reports, monitors and plots in the simulation tree.	
	Velocity	Creates a maximum and minimum velocity report, monitor and plot for all melt regions.
	Pressure	Creates a maximum and minimum pressure report, monitor and plot for all fluid regions.
	Temperature	Creates a maximum and minimum temperature report, monitor and plot for all fluid regions.

Filling/Solidification Progress		
	Melt (Mass)	Creates a field function, report, monitor, and plot to analyze the volume integral of the melt mass (kg/m3) as a function of time (s).
	Solidified (Mass)	Creates a threshold, report, monitor, and plot to analyze the volume integral of the solidified melt mass (kg) as a function of time (s).
	Volume Ratio (Melt/Cavity)	Creates a field function, report, monitor, and plot to analyze the ratio of melt volume to total volume of the fluid domain.
	Volume Ratio (Solid/Liquid)	Creates a field function, threshold, report, monitor, and plot to analyze the ratio of solidified melt volume to melt volume.
Scalar Scenes	Activating an option invokes the creation of a scalar scene.	
	Air	Creates a scalar scene visualizing the temperature field of air as a contour plot.
	Melt	Creates a scalar scene visualizing the temperature field of melt as a contour plot.
	Solidified	Creates a scalar scene visualizing the temperature field of solidified melt as a contour plot.

Post Processing Probes Tab

Probes	Lists the probes that are already defined.	
Name	User-defined name under which the probe is saved.	
Type	Specifies the type of probe.	
	Point	A point probe lets you extract variable values at a point within the computational domain.
	Line	A line probe lets you extract variable values at several points along a line within the computational domain.
Properties	Specifies the input parameters of the different probe types.	

	Origin	Specify the coordinates of the point within the computational domain where the point probe is located.
	Point 1/2	Specify the coordinates of the two points connecting the line on which the line probe point lie.
	Resolution	Specify the number of probe points along the probe line.
X-Y Plots	Creates a graph of the selected variable value at the probe over time.	
	Name	Specify a name for the graph.
	Scalar	Specify the scalar variable that you want to monitor in the graph.
	Probes	Select a previously defined probe at which you want to monitor the selected **Scalar**.

Post Processing Transient Tab

Automatic Backup	Activate the **Enable** button to define automatic backup of sim files during the simulation.	
	Trigger Type	Specify whether you want the files saved based on iteration, time step, or time interval.
	Number of Backup Files	Specify the maximum number of backup sim files to be saved.
	Start	Specify the starting iteration, time step, or time interval.
	Number of Iterations	Specify the iteration frequency by which the backup sim files are saved.
	Update Frequency	Specify the time step frequency by which the backup sim files are saved.
	Delta Time	Specify the time interval by which the backup sim files are saved.
	Enable Stop	When activated, you can specify from which point the update event stops updating. Depending on the selected **Trigger Type,**you specify iteration, time step, or time interval.

Post Process	Activate the **Enable** button to define creation of a transient post-processing simh file.	
	Trigger Type	Specify whether you want the visualization data saved based on iteration, time step or time interval.
	Start	Specify the starting iteration, time step, or time interval.
	Number of Iterations	Specify the iteration frequency by which the visualization data are saved to the simh file.
	Update Frequency	Specify the time step frequency by which the visualization data are saved to the simh file
	Delta time	Specify the time interval by which the visualization data are saved to the simh file
	Enable Stop	When activated, you can specify from which point the update event stops updating. Depending on the selected **Trigger Type,**you specify iteration, time step, or time interval.
Variables of Interest	Select the variables that you want to store to the simh file for visualization.	
	Scalar Functions	Selection of scalar field functions.
	Vector Functions	Selection of vector field functions.

Casting Field Functions

The following primitive field functions are available to the casting simulation.

Relative Mass Error Rate	The relative mass conservation error rate for all liquid phases in the computational domain. The relative mass error conservation rate is defined as: $\varepsilon = \left\lvert \left(m_{\text{target}} - m_{actual}\right) / \left(m_{\text{target}} \cdot \Delta t\right) \cdot 100 \right\rvert$, where m_{target} is the target liquid mass that should be in the domain, m_{actual} is the liquid mass that actually is in the domain, and Δt is the current time step. The target liquid mass is given by: $m_{\text{target}} = m_0 + \sum \dot{m}_{\text{Bound}} \cdot \Delta t$ where m_0 is the mass computed from the previous time step and $\dot{m}_{\text{Bound}}$ are the mass fluxes through the boundaries. The CUI uses this field function to create a stopping criterion to automatically determine the number of inner iterations (see Stopping Criteria Properties).
Fluid Side Temperature	The temperature of the fluid-side at a direct Contact Interface or an indirect Mapped Contact Interface between a fluid and a solid region. Useful to evaluate a thermal contact resistance at the interfaces based on the temperature of the melt. The thermal contact resistance can take into account a possible gap formation due to volume shrinkage during solidification. Since the temperatures between melt and mold at the interface usually differ in the range of several 100 K, the regular (averaged) temperature field is impractical. If this field function is evaluated at boundaries other than (mapped) contact interface boundaries between a fluid and a solid region, the regular temperature field is returned.
Solid Side Temperature	The temperature of the solid-side at a direct Contact Interface or an indirect Mapped Contact Interface between a fluid and a solid region. This is the opposite field function to the Fluid-Side Temperature . If this field function is evaluated at boundaries other than (mapped) contact interface boundaries between a fluid and a solid region, the regular temperature field is returned.

Contents:

Criteria Functions Field Functions Reference
Automated Time-Step Control Field Functions Reference

Criteria Functions Field Functions Reference

Some field functions are made available to the simulation when the Criteria Functions model is activated.

The following primitive field function is available when the *Temporary Storage Retained* expert property of the Segregated Volume Fraction solver is activated.

Source Rate of Void<Phase>	The VOF phase source for the auxiliary void phase.

The following primitive field functions are available to the simulation when the Criteria Functions model is activated.

Cooling Rate of <Phase> @ <Critical Temperature>	The Cooling Rate of the melting-solidifying phase at the critical temperature.
Secondary Dendrite Arm Spacing of <Phase> @ <Critical Temperature>	The Secondary Dendrite Arm Spacing (LST) of the melting-solidifying phase at the critical temperature.
Primary Dendrite Arm Spacing of <Phase>	The Primary Dendrite Arm Spacing of the melting-solidifying phase.
G/v of <Phase> @ <Critical Temperature>	The G/v ratio of the melting-solidifying phase at the critical temperature.
Isotherm Speed of <Phase> @ <Critical Temperature>	The Isotherm Speed of the melting-solidifying phase at the critical temperature.
LiquidResidenceTime<Phase>	The Liquid Residence Time of the melting-solidifying phase.
Local Solidification Time of <Phase> @ <Critical Temperature>	The Local Solidification Time of the melting-solidifying phase at the critical temperature.
Mean Cooling Rate of <Phase>	The Mean Cooling Rate of the melting-solidifying phase.
Niyama Criterion of <Phase> @ <Critical Temperature>	The Niyama Criterion of the melting-solidifying phase. (The field function **Niyama Criterion of <Phase>** in earlier releases now corresponds to **Niyama Criterion of <Phase> @T_solidus**.)
Solidification Velocity of <Phase> @ <Critical Temperature>	The Solidification Velocity of the melting-solidifying phase at the critical temperature.
SolidificationTime<Phase>	The Solidification Time of the melting-solidifying phase.
Temperature Gradient of <Phase> @ <Critical Temperature>	The Temperature Gradient of the melting-solidifying phase at the critical temperature.
Dimensionless Niyama of <Phase> @ <Critical Temperature>	The Dimensionless Niyama Criterion of the melting-solidifying phase at the critical temperature and the solidus temperature.

Pore Volume Fraction of <Phase> @ <Critical Temperature>	The absolute pore volume fraction as given by Eqn. (1444) for the critical temperature and the solidus temperature.

Automated Time-Step Control Field Functions Reference

The following primitive field functions are available to the simulation when time step controls of the Segregated Time Step model are activated.

You can plot these field functions per iteration by using maximum reports or expression reports. These field functions let you detect if a time-step limitation due to the segregated time step model takes place and which segregated time step model property dominates the current time-step.

<CFL Condition> Max CFL Ratio of <Time Step Control>	Defined as (Calculated maximum CFL number of the current time-step)/(User-specified *Target Max CFL number*). The *Target Max CFL number* is specified in the CFL condition child node. The max CFL ratio tends to converge towards 1. The higher the max CFL ratio, the more dominant the maximum CFL condition is in determining the current time-step size.
<CFL Condition> CFL Ratio of <Time Step Control>	Defined as (Calculated mean CFL number of the current time-step)/(User-specified *Target Mean CFL number*). The *Target Mean CFL number* is specified in the CFL condition child node. The mean CFL ratio tends to converge towards 1. The higher the mean CFL ratio, the more dominant the mean CFL condition is in determining the current time-step size.
Min Time Step Size Limit Active of <Time Step Control>	Flag that indicates whether the determined time-step size is limited by the *Min. Time Step Size* of that time step control. A value of 0 indicates that the current time-step is not limited; a value of 1 indicates that it is limited.
Max Time Step Size Limit Active	Flag that indicates whether the determined time-step size is limited by the maximum *Time-Step* as specified under **Solvers** > **Implicit Unsteady**.

Casting Bibliography

[193] *K.D. Carlson, C. Beckerman, "Prediction of Shrinkage Pore Volume Fraction Using a Dimensionless Niyama Criterion", Metallurgical and Materials Transactions A, 40A, 2009, pp. 163-175*

Theory

STAR-CCM+ simulations are built on numerical algorithms that solve relevant laws of physics according to the conditions that the simulation defines. So that you can identify the physical laws, and understand something of the methods by which they are solved, STAR-CCM+ comes with a Theory Guide.

This Theory Guide is being delivered in installments, the first of which was released with STAR-CCM+ v11.06. To identify sections that are not yet complete, a pair of '~' symbols are placed around the section titles. These sections are targeted for delivery in future releases of STAR-CCM+. For those sections, you can find underlying theory in formulation topics within the User Guide.

Contents:

Fundamental Equations
Fluid Flow
Numerical Flow Solution
~Moving Reference Frames~
Turbulence
~Transition~
Heat Transfer
Porous Media
Species and Passive Scalars
Eulerian Multiphase Flow
Lagrangian Multiphase Flow
Reacting Flow
Electrochemistry
Electromagnetism
Solid Mechanics
Aeroacoustics
Finite Element Method
~Mesh Motion~
Rotating Flow
~Harmonic Balance~
Adjoint
Vector and Tensor Algebra

Fundamental Equations

STAR-CCM+ models a range of physics phenomena including fluid mechanics, solid mechanics, heat transfer, electromagnetism, and chemical reactions. On a macroscopic scale, where the typical lengths are much greater than the inter-atomic distances, the discrete structure of matter can be neglected and materials can be modeled as continua. The mathematical models that describe the physics of continua are derived from fundamental laws that express conservation principles.

The conservation laws for a continuum can be expressed using an Eulerian approach or a Lagrangian approach. In the Eulerian approach, a given *volume* represents a portion of space where material can flow through. In the Lagrangian approach, a given *volume* represents a portion of material in the body, so that an observer follows the material as it moves through space.

STAR-CCM+ employs both Lagrangian and Eulerian descriptions, whichever is most convenient for modeling a particular field of physics. For dispersed phases, STAR-CCM+ offers a choice, as it implements both Eulerian and Lagrangian descriptions to describe similar phenomena.

Most of the physics represented in STAR-CCM+ stem from a core set of fundamental laws. These fundamental laws are presented here in differential form, for an infinitesimal control volume. Further sections of the theory guide develop these laws into the numerical solution techniques that STAR-CCM+ employs.

Mechanics

Continuum mechanics studies the behavior of continua in response to mechanical forces. The fundamental laws that govern the mechanics of fluids and solids are the conservation of mass, linear momentum, angular momentum, and energy.

Conservation of Mass

The balance of mass through a control volume is expressed by the continuity equation:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{1450}$$

where ρ is the density, that is, the mass per unit volume, and $\mathbf{v}$ is the continuum velocity.

Conservation of Linear Momentum

The time rate of change of linear momentum is equal to the resultant force acting on the continuum:

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v}) = \nabla \cdot \boldsymbol{\sigma} + \mathbf{f}_{\mathrm{b}} \tag{1451}$$

where $\otimes$ denotes the Kronecker product, $\mathbf{f}_{\mathrm{b}}$ is the resultant of the body forces (such as gravity and centrifugal forces) per unit volume acting on the continuum, and $\boldsymbol{\sigma}$ is the stress tensor. For a fluid, the

stress tensor is often written as sum of normal stresses and shear stresses, $\boldsymbol{\sigma} = -p\mathbf{I} + \mathbf{T}$, where p is the pressure and $\mathbf{T}$ is the viscous stress tensor, giving:

$$\frac{\partial(\rho\mathbf{v})}{\partial t} + \nabla\cdot(\rho\mathbf{v}\otimes\mathbf{v}) = -\nabla\cdot(p\mathbf{I}) + \nabla\cdot\mathbf{T} + \mathbf{f}_{\mathrm{b}} \tag{1452}$$

Conservation of Angular Momentum

Conservation of angular momentum requires that the stress tensor is symmetric:

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^{\mathrm{T}} \tag{1453}$$

Conservation of Energy

When the first law of thermodynamics is applied to the control volume, the conservation of energy can be written as:

$$\frac{\partial(\rho E)}{\partial t} + \nabla\cdot\left(\rho E\mathbf{v}\right) = \mathbf{f}_{\mathrm{b}}\cdot\mathbf{v} + \nabla\cdot(\mathbf{v}\cdot\boldsymbol{\sigma}) - \nabla\cdot\mathbf{q} + S_E \tag{1454}$$

where E is the total energy per unit mass, $\mathbf{q}$ is the heat flux, and S_E is an energy source per unit volume.

Electromagnetism

Electromagnetism studies the behavior of continua in response to electromagnetic fields. The fundamental laws that describe the electromagnetic behavior of a continuum are Maxwell's equations and the conservation of electric charge.

Maxwell's Equations

Maxwell's equations can be written as:

$$\frac{\partial\mathbf{B}}{\partial t} + \nabla\times\mathbf{E} = 0 \tag{1455}$$

$$\frac{\partial\mathbf{D}}{\partial t} - \nabla\times\mathbf{H} = -\mathbf{J} \tag{1456}$$

$$\nabla\cdot\mathbf{D} = \rho \tag{1457}$$

$$\nabla\cdot\mathbf{B} = 0 \tag{1458}$$

where ρ is the electric charge density, $\mathbf{J}$ is the electric current density, $\mathbf{E}$ is the electric field, $\mathbf{D}$ is the electric flux density, $\mathbf{H}$ is the magnetic field, and $\mathbf{B}$ is the magnetic flux density.

Conservation of Electric Charge

The conservation of charge within a control volume is given by the continuity equation:

$$\nabla\cdot\mathbf{J} + \frac{\partial\rho}{\partial t} = 0 \tag{1459}$$

STAR-CCM+ also models chemical reactions, radiation, and a variety of other phenomena. The mathematical models that describe these phenomena are discussed in specific sections.

Constitutive Laws

In most cases, the partial differential equations of a mathematical model are not a closed set, that is, the number of unknown quantities exceeds the number of equations. To provide closure, additional equations are added to the mathematical models. These additional equations, called constitutive laws, depend on the material under consideration.

Discretization and Solution

STAR-CCM+ uses discretization methods to convert the continuous system of equations to a set of discrete algebraic equations, which can be solved using numerical techniques.

Discretization methods follow a common procedure:

- the continuous domain is divided into a finite number of subdomains (cells/elements)
- the unknowns are stored at specific locations of the mesh (vertices, cell centroids, face centroids, or edges)
- an integral or weak form of the differential equations is employed for spatial discretization. After discretizing the time derivative, a coupled system of algebraic equations (nonlinear, in general) results that needs to be solved at each time-step

Depending on the mathematical model, STAR-CCM+ discretizes the continuous equations using either the Finite Volume or the Finite Element method.

Fluid Flow

STAR-CCM+ can simulate internal and external fluid flow across a wide range of flow regimes, and for a variety of fluid types. It solves the conservation equations for mass, momentum, and energy for general incompressible and compressible fluid flows.

Equations of Fluid Flow in the Absolute Velocity Formulation

Consider a reference frame that rotates with constant angular velocity $\boldsymbol{\omega}$ around an axis and translates with a translation velocity v_t relative to a stationary (laboratory) reference frame.

The velocity relative to the moving reference frame (relative velocity) $\mathbf{v}_r$ transforms from the velocity in the stationary laboratory reference frame (absolute velocity) $\mathbf{v}$ by:

$$\mathbf{v}_r = \mathbf{v} - \mathbf{v}_g = \mathbf{v} - \left(\mathbf{v}_t + \boldsymbol{\omega} \times \mathbf{r}\right) \tag{1460}$$

where:

$\mathbf{v}_g$: the reference frame velocity relative to the inertial laboratory frame

$\mathbf{r}$: position vector from the origin of the moving reference frame to a point in the computational domain.

With Eqn. (1460), and integrating over a finite control volume, the governing equations of fluid flow can be written as:

Continuity Equation

$$\frac{\partial}{\partial t}\left(\int_V \rho dV\right) + \oint_A \rho \mathbf{v}_r \cdot d\mathbf{a} = \int_V S_u dV \tag{1461}$$

Momentum Equation

$$\frac{\partial}{\partial t}\left(\int_V \rho \mathbf{v} dV\right) + \oint_A \rho \mathbf{v}_r \otimes \mathbf{v} \cdot d\mathbf{a} = -\oint_A p\mathbf{I} \cdot d\mathbf{a} + \oint_A \mathbf{T} \cdot d\mathbf{a} - \int_V \rho\left[\boldsymbol{\omega} \times \left(\mathbf{v} - \mathbf{v}_t\right)\right] dV + \int_V \mathbf{f}_b dV \tag{1462}$$

Energy Equation

$$\frac{\partial}{\partial t}\int_V \rho E dV + \oint_A \left[\rho H \mathbf{v}_r + \mathbf{v}_g p\right] \cdot d\mathbf{a} = -\oint_A \dot{\mathbf{q}}'' \cdot d\mathbf{a} + \oint_A \mathbf{T} \cdot \mathbf{v} d\mathbf{a} + \int_V \mathbf{f}_b \cdot \mathbf{v} dV + \int_V S_E dV \tag{1463}$$

Constitutive Relations

In order to solve the momentum equations for the velocity field, closure must be provided between the stress tensor and the velocity field of the fluid. A large number of constitutive equations exist that incorporate different material properties of the fluid such as viscosity, first and second normal stress coefficient. In

addition, the equations of state, which are also constitutive relations, are required for the closure of the system of equations that is described above.

Contents:

Equations of State
Newtonian Fluids
Non-Newtonian Fluids
Boundary Conditions
Gravity
Fluid Flow Bibliography

Equations of State

The equations of state are constitutive relations that describe the relation between the density and the internal energy to the two basic thermodynamic variables pressure and temperature.

STAR-CCM+ provides the following options:

Constant Density

$$\rho = \rho_0 \tag{1464}$$

where ρ_0 is a constant.

Polynomial Density

For gases, liquids, and solids, density can be specified as a function of temperature T in the form of a polynomial. The temperature range can be subdivided into intervals, each with its own polynomial function:

$$\rho = \sum_{i=0}^{n} a_i \cdot T^{i-1} \tag{1465}$$

where a is a polynomial coefficient.

Ideal Gas

The ideal gas law expresses density as a function of temperature and pressure:

$$\rho = \frac{p}{RT} \tag{1466}$$

where:

p is the pressure
R is the specific gas constant

and:

$$R = R_u / M \tag{1467}$$

where:

R_u is the universal gas constant [8314.4621 J/kmol K]
M is the molecular weight

Real Gas

At high pressure and low temperature, the p-v-T behavior of real gases deviates from that predicted by the ideal gas equation. This change of behavior results from the molecules of the gas taking up a significant portion of the total volume as the gas density increases. In addition, intermolecular attractive forces become increasingly important.

Van der Waals

The van der Waals equation is given by:

$$\left(p + a/v^2\right)\ \left(v - b\right) = R_u\ T \tag{1468}$$

Van der Waals replaced specific volume in the ideal gas relation, $pv = R_u T$, with $(v - b)$ to account for the volume that the particles of the gas occupy, while replacing the pressure with the term $\left(p + a/v^2\right)$. The constant b is the co-volume of the particles, and the constant a is a measure of the attractive forces.

The constants a and b are evaluated based on the experimental observation that, in p-v space, the critical isotherm $T = T_c = \text{constant}$ has zero slope and an inflection point at the critical state, $p = p_c$ and $v = v_c$. Thus the following can be written:

$$\frac{dp}{dv}|T_c = 0 \tag{1469}$$

and:

$$\frac{d^2p}{dv^2}|T_c = 0 \tag{1470}$$

which leads to the following expressions for a and b:

$$a = \frac{27}{64}\frac{{R_u}^2 T_c^2}{p_c}, \qquad b = \frac{1}{8}\frac{R_u T_c}{p_c} \tag{1471}$$

These equations allow a and b to be determined from experimentally measured values of critical pressure and temperature, p_c and T_c.

Peng Robinson

The Peng-Robinson equation is given by:

$$p = \frac{R_u T}{(v - b)} - \frac{a\ \alpha\left(T_r\right)}{\left(v^2 + 2bv - b^2\right)} \tag{1472}$$

where v is the specific volume.
The function $\alpha\left(T_r\right)$ is given by:

$$\alpha\left(T_r\right) = \left[1 + \left(0.37464 + 1.54226\ \omega - 0.2699\ \omega^2\right)\left(1 - {T_r}^{0.5}\right)\right]^2 \tag{1473}$$

where $T_r = T/T_c$ is the reduced temperature and ω is the acentric factor of the gas.

The constants are found to be:

$$a = \frac{0.4572 R_u^2 T_c^{2.5}}{p_c}, \quad b = \frac{0.0778 R_u T_c}{p_c} \tag{1474}$$

Redlich-Kwong

The Redlich-Kwong equation is given by:

$$p = \frac{R_u T}{(v - b)} - \frac{a}{\left[T^{\ 0.5} v(v + b)\right]} \tag{1475}$$

and

$$a = \frac{0.4275 R_u^2 T_c^{2.5}}{p_c}, \qquad b = \frac{0.0867 R_u T_c}{p_c} \tag{1476}$$

Soave-Redlich-Kwong

Soave modified the Redlich-Kwong equation, replacing the $T^{0.5}$ power in the denominator of Eqn. (1476) with a different temperature-dependent expression [236]. This modification improves predictions for liquid densities and vapor-liquid equilibrium, and is:

$$p = \frac{R_u T}{v-b} - \frac{a\,\alpha\left(T_r\right)}{v\,\left(v+b\right)} \tag{1477}$$

The function $\alpha\left(T_r\right)$ is:

$$\alpha\left(T_r\right) = \left[1 + \left(0.48 + 1.574\omega - 0.176\omega^2\right)\left(1 - T_r^{0.5}\right)\right]^2 \tag{1478}$$

and

$$a = \frac{0.42748 R_u^2 T_c^{2.5}}{p_c}, \qquad b = \frac{0.08664 R_u T_c}{p_c} \tag{1479}$$

Modified Soave-Redlich-Kwong

Graboski and Daubert modified the Soave-Redlich-Kwong model to improve its behavior for phase equilibrium calculations that involve non-polar molecules [211] and [212]. All equations remain identical to the Soave-Redlich-Kwong model, except the function $\alpha\left(T_r\right)$ which changes to:

$$\alpha\left(T_r\right) = \left[1 + \left(0.48503 + 1.55171\ \omega - 0.15613\ \omega^2\right)\left(1 - T_r^{0.5}\right)\right]^2 \tag{1480}$$

Equilibrium Air

For air at high temperatures, the effects of molecular dissociation, internal energy excitation, and ionization become significant.

For situations where the time scale of these effects is much shorter than the time scale of the flow, these effects can be treated in an equilibrium manner. In this case, the properties of the flow, such as the density, specific heat, and transport properties, become expressions of two thermodynamic variables. These expressions typically take the forms of curve fits that account for the chemical reactions and energy modes present at high temperature. In particular, the curve fits of Gupta and others [213] were used. These curve fits specify the compressibility factor, specific heat, enthalpy, viscosity, and thermal conductivity as a function of temperature and pressure. The curve fits are valid for temperatures less than 30,000 K and for pressures between 10^{-4} and 10^{2} atmospheres.

The equilibrium air equation of state provides the compressiblity factor as a field function. In the context of this model, the compressibility factor represents the ratio of the undissociated molecular weight to the mean molecular weight of air [213]:

$$Z\left(T, P\right) = \frac{M_o}{\overline{M}} \tag{1481}$$

where M_o represents the molecular weight of undissociated air and $\overline{M}$ represents the mean molecular weight of air at the given temperature and pressure. Because the mean molecular weight is a function of

the composition, the compressibility factor can be used as a measure of the degree of dissociation in the air, with higher values signifying a large degree of dissociation and ionization.

Thermal Non-Equilibrium

For a gas, the internal energy is composed of translational, rotational, vibrational, and electronic modes. In equilibrium, these energy modes can be adequately described by a single characteristic temperature. For thermal non-equilibrium, the translational and rotational energy modes are described by one temperature, while the vibrational and electronic energy modes are described by an additional vibrational-electronic temperature. The vibrational-electronic temperature is determined by solving an additional energy equation describing the conservation of vibrational-electronic energy. Gnoffo and others [210] give this equation, simplified to non-ionizing flows by Lockwood [223]):

$$\frac{\partial \rho e_{ve}}{\partial t} + \nabla \cdot \left(\rho e_{ve} \mathbf{U}\right) = \nabla \cdot \left(k_{ve} \nabla T\right) + \nabla \cdot \left(\rho \sum_s h_{ve,s} D_s \nabla y_s\right) + \sum_{s=\mathrm{mol}} \rho_s \frac{\left(e^*_{v,s} - e_{v,s}\right)}{<\tau_s>} + \sum_{s=\mathrm{mol}} \dot{\omega}_s e_{ve,s} \tag{1482}$$

The exchange of vibrational-electronic energy and translational-rotational energy is modeled through a relaxation term. Gnoffo and others [210] present this relaxation, and provide a simplification which allows the calculation to include fewer species-dependent parameters:

$$e^*_{v,s} - e_{v,s} \approx C^s_{v,v}\left(T - T_v\right) \tag{1483}$$

$$\sum_{s=\mathrm{mol}} \frac{\rho_s C^s_{v,v}}{\langle \tau_s \rangle} \approx \frac{\rho\, C^s_{v,v}}{\overline{\tau}_v} \tag{1484}$$

To describe the relationship between vibrational-electronic enthalpy and temperature, the vibrational specific heat must be specified. This vibrational specific heat is specified by simultaneously setting the total specific heat and the translational-rotational specific heat. For this model, the translational-rotational energy modes are assumed fully excited $(F-E)$, giving a constant translational-rotational specific heat $\left(C_{p,tr}\right)$. With these two values specified, the vibrational specific heat and enthalpy are given by [210]:

$$C_{p,ve}\left(T_{ve}\right) = C_{p,\mathrm{total}}\left(T_{ve}\right) - C^{F-E}_{p,tr} \tag{1485}$$

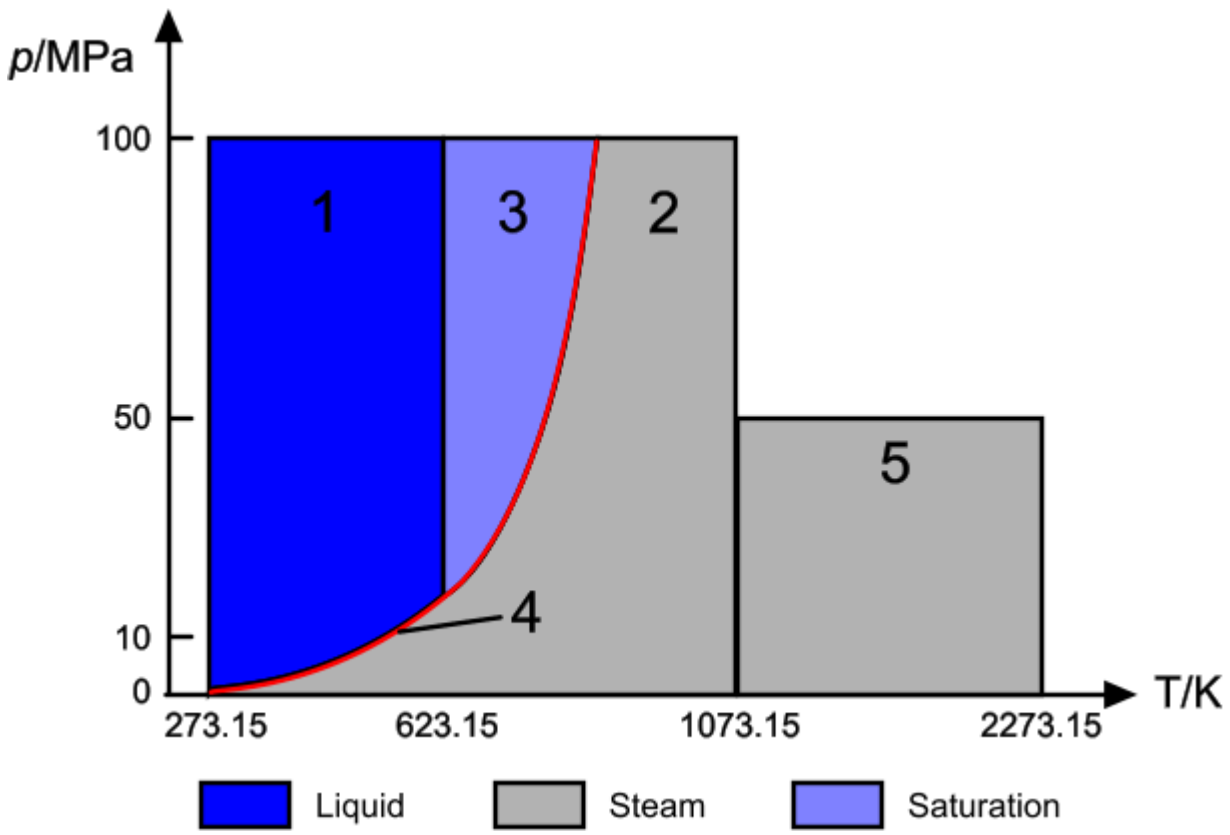
$$h_{ve}\left(T_{ve}\right) = h_{\mathrm{total}}\left(T_{ve}\right) - C^{F-E}_{p,tr}\left(T_{ve} - T_{\mathrm{ref}}\right) - h_o \tag{1486}$$

IAPWS-IF97

The models of IAPWS-IF97 (International Association for the Properties of Water and Steam, Industrial Formulation 1997) allow you to run simulations of liquid water or gaseous steam.

These models include calculations of density and other thermodynamic properties. The IAPWS-IF97 provides fundamental polynomial equations for the specific Gibbs free energy, $g(p,T)$. Specific volume, internal energy, entropy, enthalpy, heat capacity, and speed of sound are derived from the fundamental equation using appropriate combinations of the dimensionless Gibbs free energy and its derivatives [217].

Based on the Gibbs free energy, the IAPWS-IF97 models in STAR-CCM+ are only valid within certain ranges of temperature and pressure. These ranges are represented in a diagram of regions [217] which are essentially the phases of water and steam.



The following table lists the components of the region diagram and how they work with IAPWS-IF97 models in STAR-CCM+.

Component	Ranges of Validity		STAR-CCM+ Models
Region 1 (liquid)	273.15 K ≤ T ≤ 623.15 K	p ≤ 100 MPa	IAPWS-IF97 (Water)
Region 2 (steam)	273.15 K ≤ T ≤ 1073.15 K	p ≤ 100 MPa	IAPWS-IF97 (Steam)
Region 3 (saturation)	This phase region is not supported in STAR-CCM+.		
Boundary 4	This boundary lies between liquid and saturation on one side, and steam on the other.		
Region 5 (steam)	1073.15 K < T ≤ 2273.15 K	p ≤ 50 MPa	IAPWS-IF97 (Steam)

Newtonian Fluids

A Newtonian fluid is described by an explicit constitutive equation that relates the viscous stress tensor $\mathbf{T}$ to the velocity field through a constant viscosity. The relation between the shear stress and the shear rate is linear.

The viscous stress tensor in Eqn. (1452) is not constant, but a variable function of the velocity field for a particular fluid. Typically, in a constitutive relation, the velocity field is expressed in the form of the rate of deformation tensor:

$$\mathbf{D} = \frac{1}{2}\left(\nabla\mathbf{v} + \left(\nabla\mathbf{v}\right)^T\right) \tag{1487}$$

Newtonian fluid is the simplest mathematical model to describe the viscous behavior of many liquids and gases such as water and air.

The stress tensor is given by:

$$\mathbf{T} = 2\mu\mathbf{D} - \frac{2}{3}\left(\nabla\cdot\mathbf{v}\right)\mathbf{I} \tag{1488}$$

where μ is the constant dynamic viscosity of the fluid and $\mathbf{D}$ is the rate of deformation (strain) tensor given by Eqn. (1487). For incompressible flows, the second term in Eqn. (1488) is zero due to the continuity equation.

Temperature-Dependent Newtonian Viscosity

For gases, the viscosity can be a function of temperature.

Sutherland's Law

The Sutherland equation is a widely used approximation for modeling the dependence of gas viscosity on temperature:

$$\frac{\mu}{\mu_0} = \left(\frac{T}{T_0}\right)^{\frac{3}{2}}\left(\frac{T_0 + S}{T + S}\right) \tag{1489}$$

Here, S is an effective temperature, called the Sutherland constant, and μ_0 is the reference viscosity at the reference temperature $T_0 = 273\,\mathrm{K}$

Power Law

For dilute gases, the temperature-dependent viscosity can be approximated by a power-law:

$$\frac{\mu}{\mu_0} = \left(\frac{T}{T_0}\right)^n \tag{1490}$$

where T_0 is the reference temperature, μ_0 is the reference viscosity and n is the Power law exponent.

Non-Newtonian Fluids

For many industrial fluids, it is not sufficient to assume a linearly viscous fluid behavior due to their complex microstructure. These fluids exhibit non-Newtonian behavior.

Although any fluid with a viscosity that is not constant is considered non-Newtonian, not all non-Newtonian fluids behave in the same way—there are different types. The types that STAR-CCM+ can model are described as follows:

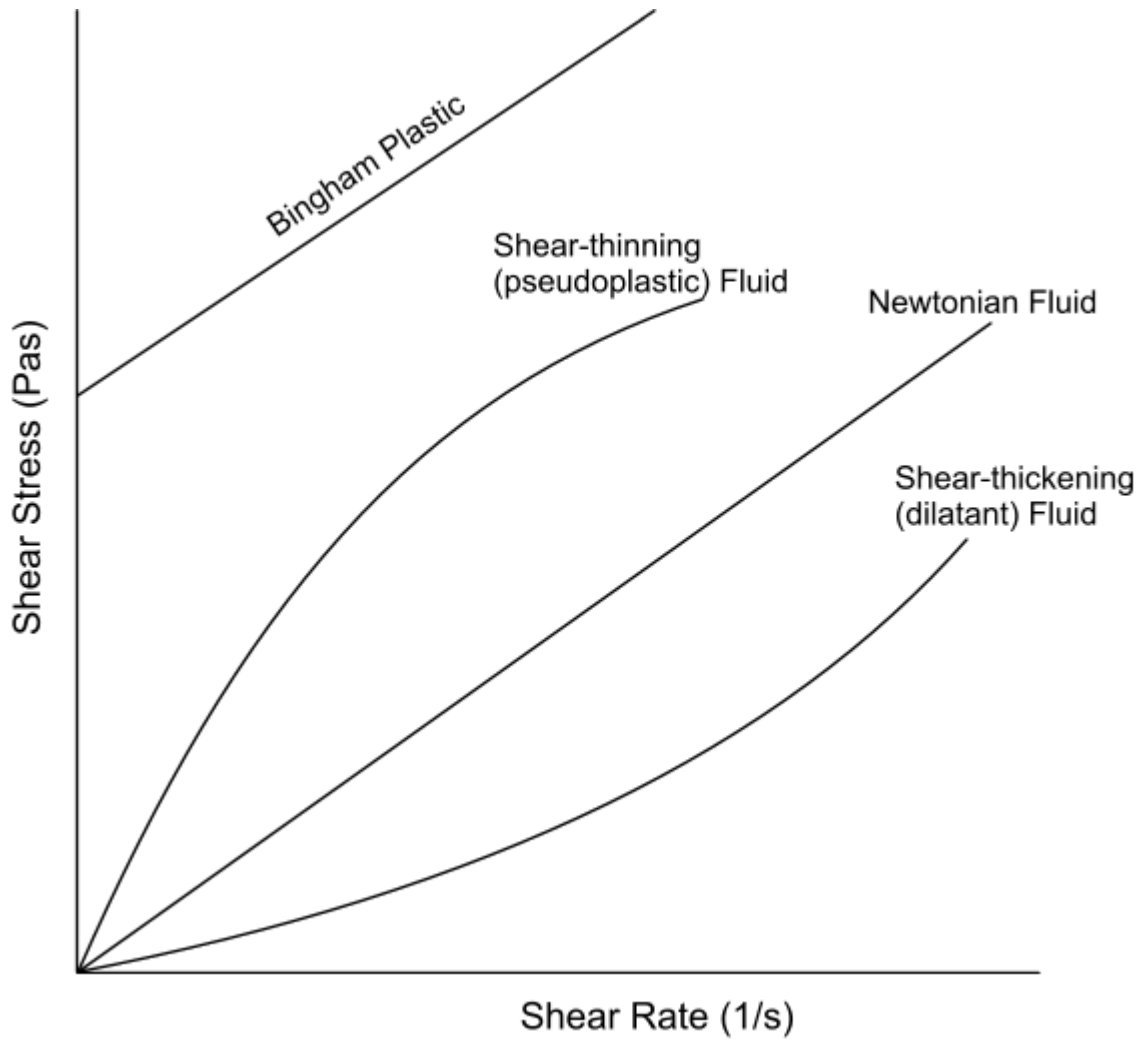
Generalized Newtonian

A Generalized Newtonian fluid is the simplest mathematical description of a non-Newtonian fluid. The constant viscosity in Eqn. (1488) is replaced by an apparent viscosity $\mu = \mu(\dot{\gamma}, T)$ that is a function of shear-rate and/or temperature.

Viscoplastic

Viscoplastic fluids fall within the same category as Generalized Newtonian fluids, but with the addition of a yield stress. Viscoplastic fluids only start to flow if the deformation exceeds a critical shear stress value.

The following diagram shows the shear stress as a function of shear rate for different types of fluids. A Newtonian fluid gives a straight line through the origin where the slope is given by the constant viscosity. For the shear-thinning (pseudoplastic) fluid, the shear stress gives a concave downward curve with increasing shear rate, which corresponds to a decrease in viscosity as the shear rate increases. The shear-thickening (dilatant) fluid shows the opposite behavior. The shear stress gives a concave upward curve with increasing shear rate, which corresponds to an increase in viscosity as the shear rate increases. The viscoplastic fluid, commonly termed Bingham plastic, must exceed the yield stress before it can move.



Viscoelastic

Viscoelastic fluids exhibit both viscous and elastic behavior when undergoing deformation. Their stresses not only depend on the current motion of the fluid, but also on the history of the motion. The deformation that the fluid experienced in the past influences the current stresses. In contrast to a purely elastic material, the influence of the past deformation decreases over time. This effect can be quantified by use of the so-called relaxation time.

Each viscoelastic model provides a constitutive equation which solves the full stress tensor $\mathbf{T}(\mathbf{D})$. Most complex fluids are slightly viscoelastic. However, if the shear viscosity is the most important attribute and elastic effects are small, they can often be treated as Generalized Newtonian fluids. Typical viscoelastic fluids are: polymer melts, rubbers, and bread dough. Weakly viscoelastic fluids include shampoos, paints, and polymer solutions.

Multiphase

Some multiphase fluids are non-Newtonian when the mixture viscosity depends on the volume fraction of the dispersed phase $\mu = \mu(\phi)$. Multiphase non-Newtonian fluids can be modeled with separate phases or as a single phase mixture (assuming that the volume fraction remains constant). These multiphase non-

Newtonian fluids are often viscoplastic and include emulsions, suspensions, and biological fluids (blood, joint fluid).

The following table provides an overview of the available non-Newtonian fluids in STAR-CCM+ along with their discretization methods and characteristics:

Approach	Discretization	Shear normal stresses	Turbulence
Generalized Newtonian	FV	No	Yes
Generalized Newtonian	FE	No	No
Viscoelastic	FE	Yes	No
Multiphase	FV	Yes	No

The finite element (FE) solver is tailored to solve viscoelastic models. Simulations that involve Generalized Newtonian fluids using the finite volume (FV) solver can make use of most of the turbulence models available to Newtonian fluids.

Contents:

Generalized Newtonian Fluid
Viscoplastic Fluid
Viscoelastic Fluids
Time-Temperature Superposition
Film Casting

Generalized Newtonian Fluid

The generalized Newtonian fluid model is an explicit constitutive equation that relates the stress tensor to the velocity field through a variable viscosity. This model does not account for elastic effects.

For a generalized Newtonian fluid, the stress tensor is given by:

$$\mathbf{T}(\dot{\gamma}) = 2\mu(\dot{\gamma})\mathbf{D} \tag{1491}$$

where $\dot{\gamma}$ is the shear rate.

The shear rate is calculated from the second invariant of the rate of deformation (strain) tensor:

$$\dot{\gamma} = \sqrt{(2\mathbf{D}:\mathbf{D})} \tag{1492}$$

In STAR-CCM+, the following generalized Newtonian fluid models are implemented, that is, shear-rate dependent viscosity functions:

Power Law

The simplest model to describe a shear-rate dependent viscosity behavior is the power law.

The value of the power law exponent n determines the class of the fluid:

$$\mu(\dot{\gamma}, T) = k\dot{\gamma}^{n-1} \tag{1493}$$

- $n = 1 \quad \Rightarrow$ Newtonian fluid

- $n > 1$ ⇒ Shear-thickening (dilatant) fluid
- $n < 1$ ⇒ Shear-thinning (pseudo-plastic) fluid

Cross Fluid

The Cross Fluid is defined from [201] as:

$$\mu(\dot{\gamma}, T) = \mu_\infty + \frac{\mu_0 - \mu_\infty}{1 + \left(\frac{\dot{\gamma}}{\dot{\gamma}_c}\right)^m} \tag{1494}$$

where:

m is the Cross rate constant
μ_0 is the zero-shear viscosity, the viscosity at the Newtonian plateau
μ_∞ is the infinite-shear viscosity
$\dot{\gamma}_c$ is the critical shear strain-rate at which shear-thinning starts

Newtonian behavior is regained for $m = 0$. As m increases towards unity, the degree of shear-thinning becomes stronger.

Carreau-Yasuda Fluid

The non-Newtonian Generalized Carreau-Yasuda Fluid for viscosity comes from [199] and [244]. It is defined as:

$$\mu(\dot{\gamma}, T) = \mu_\infty + (\mu_0 - \mu_\infty)\left(1 + (\lambda\dot{\gamma})^a\right)^{(n-1)/a} \tag{1495}$$

where:

n power constant
λ relaxation time constant
a parameter to control shear-thinning. When $a = 2$, the original Carreau model is regained.

Viscoplastic Fluid

Viscoplastic fluids are materials that exhibit a yield stress: below the yield stress, there is no deformation of the fluid and it behaves like a rigid solid. When the yield stress is exceeded, the material flows like a fluid. These materials are also called Bingham plastics after Bingham, who first described these types of materials.

Such flow behavior can be observed for slurries and suspensions, certain polymer solutions, heavy oils, cosmetic creams, liquid chocolate, and some pastes.

The constitutive equation can be written as:

$$\mathbf{T} = \mathbf{T}_0 + 2\mu(\dot{\gamma})\mathbf{D} \tag{1496}$$

where $\mathbf{T}_0$ is the yield stress. For $\mathbf{T} > \mathbf{T}_0$, the material flows; for $\mathbf{T} < \mathbf{T}_0$, the material remains solid.

STAR-CCM+ provides the Herschel-Bulkley model for viscoplastic fluids.

Herschel-Bulkley

The Herschel-Bulkley model for Bingham plastics is commonly used to model materials like concrete, mud, and toothpaste.

$$\mu(\dot{\gamma}) = \begin{cases} a_T \cdot \mu_0 & \text{, if } \dot{\gamma} < \dfrac{\tau_0}{\mu_0} \\ a_T \cdot \dfrac{\tau_0 + k\left[a_T\left(\dot{\gamma} - \frac{\tau_0}{\mu_0}\right)\right]^n}{a_T\dot{\gamma}} & \text{, if } \dot{\gamma} > \dfrac{\tau_0}{\mu_0} \end{cases} \tag{1497}$$

where:

k is the consistency factor
n is the power law exponent
τ_0 is the yield stress threshold
μ_0 is the yielding viscosity
μ_{min} is the minimum viscosity limit
μ_{max} is the maximum viscosity limit
$\dot{\gamma}$ is the shear rate
a_T is the temperature shift factor

For low shear rates ($\dot{\gamma} < \tau_0/\mu_0$), the material behaves like a highly viscous fluid with yielding viscosity μ_0. Once the yield stress threshold is passed with increasing shear rate ($\dot{\gamma} > \tau_0/\mu_0$), the viscosity is given by the power law expression. In both cases, the viscosity is bounded by μ_{min} and μ_{max}.

The method reduces to the Ostwald de Waele law, (or the standard power law), when the yield stress threshold is set to zero: $\tau_0 = 0$.

Viscoelastic Fluids

Viscoelastic fluids are materials that show both viscous behavior of a fluid and elastic behavior of a solid.

Viscoelastic fluids can exhibit qualitatively different phenomena from Newtonian fluid flow. These flow phenomena are due to the normal stresses that exist within viscoelastic fluids. Such well-known effects are:

- Weissenberg effect or rod-climbing effect. Consider a rotating rod whose end is immersed into a viscoelastic fluid, for example bread dough. The rotation of the rod generates elastic forces and a consequent stretching of the fluid. This stretching results in a positive normal force and the fluid rises up the rod. By contrast, in Newtonian liquids inertial forces dominate and the fluid moves away from the rod. The Weissenberg effect is important in industrial mixing processes.
- Die swell or extrudate swell. Consider extrusion of a polymeric liquid through a die. On exiting the die, the viscoelastic fluid swells, that is, it increases in shape and volume. This phenomenon is related to the normal stresses of the fluid inside the pipe. When the fluid exits the die, these normal stresses are relieved and the jet expands in the transverse direction.

Modeling viscoelastic fluids requires complex constitutive equations for the relation between the stress tensor and the velocity field. STAR-CCM+ provides several non-linear viscoelastic models of the differential type. The constitutive equations for the tensor components form a system of non-linear partial differential equations that are discretized using the finite element method and solved using a direct solver, the viscous flow solver.

The viscous flow solver assumes fully developed velocity and stress at inlets and outlets. For an internal flow, fully developed velocity means that all velocity vectors are parallel to each other and perpendicular to the inlet and outlet cross sections. Fully developed stress means that the stress components are calculated based on the conditions considered for the fully developed velocity at inlet or outlet. With these conditions, the velocity and stress components in a pipe or channel with uniform cross section are independent of the streamwise direction. Notice that this last condition cannot be satisfied for three-dimensional flows of viscoelastic fluids with non-zero second normal stress difference due to the formation of secondary flow.

For 3D flows, the presence of the second normal stress brings about the secondary motion in the pipe. The streamlines are no longer perfectly parallel, the fluid gains velocity in the third direction, and there is no fully developed flow. As a result, fully developed conditions are not relevant for the three-dimensional flows of viscoelastic models with second normal stress. This remains true for three dimensional flows with the periodic condition as well.

The Oldroyd-B model, the Phan-Thien-Tanner model, and the Giesekus model can be used in various modes to fit the relaxation modulus accurately and to predict the shear thinning as well as normal stresses and extensional viscosity. The stresses of all of these models are split into a polymeric (viscoelastic) part $\mathbf{T}^p$ and a Newtonian solvent part $\mathbf{T}^s$:

$$\mathbf{T} = \mathbf{T}^s + \mathbf{T}^p \tag{1498}$$

where:

$$\mathbf{T}^s = 2\mu_s\mathbf{D} \tag{1499}$$

The solvent viscosity μ_s can be constant or dependent on the shear-rate. Any of the shear-rate dependent viscosity functions of Eqn. (1493) to Eqn. (1495) can be applied. $\mathbf{D}$ is the rate-of-deformation tensor; see Eqn. (1487).

To provide an estimate of the shear resistance of the viscoelastic material with respect to the local shear rate in the system, the viscoelastic equivalent viscosity is introduced:

$$\mu_{equiv} = \frac{II_T}{II_D} = \frac{(tr[\mathbf{T}])^2 - tr\left(\mathbf{T}^2\right)}{(tr[\mathbf{D}])^2 - tr\left(\mathbf{D}^2\right)} \tag{1500}$$

The total viscosity is given by:

$$\mu_{total} = \mu_s + \mu_{equiv} \tag{1501}$$

Oldroyd-B Model

The Oldroyd-B model [230] is mostly used to describe the rheological characteristics of polymer liquids composed at low concentration and moderate shear rates.

The Oldroyd-B model can be considered as a linear superposition of the Upper Convected Maxwell model stress for the polymeric part $\mathbf{T}^p$ and a Newtonian solvent contribution $\mathbf{T}^s$.

$$\mathbf{T}^p + \lambda\overset{\nabla}{\mathbf{T}}{}^p = 2\mu_0\mathbf{D} \tag{1502}$$

where

$$\overset{\triangledown}{\mathbf{T}} = \frac{\partial}{\partial t}\mathbf{T} + \mathbf{v}\cdot \triangledown \mathbf{T} - (\triangledown \mathbf{v})^T \cdot \mathbf{T} - \mathbf{T}\cdot \triangledown \mathbf{v} \tag{1503}$$

is the upper-convected derivative of the stress tensor $\mathbf{T}$, μ_0 is the zero-shear rate (model) dynamic viscosity, and λ is the (modal) relaxation time.

Linear Phan-Thien-Tanner Model

$$\left[1 + \frac{\epsilon\lambda}{\mu_0} tr(\mathbf{T}^p)\right]\mathbf{T}^p + \lambda\left[\overset{\triangledown}{\mathbf{T}}^p + \xi(\mathbf{T}^p \cdot \mathbf{D} + \mathbf{D}\cdot\mathbf{T}^p)\right] = 2\mu_0\mathbf{D} \tag{1504}$$

where ϵ is the Phan-Thien-Tanner parameter and ξ is a model parameter that gives a combination of the upper-convected and the lower-convected derivatives of the stress tensor, $0 \leq \xi \leq 2$.

When ϵ is zero, the model reduces to the Johnson-Segalman model:

$$\mathbf{T}^p + \lambda\left[\overset{\triangledown}{\mathbf{T}}^p + \xi(\mathbf{T}^p \cdot \mathbf{D} + \mathbf{D}\cdot\mathbf{T}^p)\right] = 2\mu_0\mathbf{D}$$

Exponential Phan-Thien Tanner Model

$$\left[\exp\left(\frac{\epsilon\lambda}{\mu_0} tr(\mathbf{T}^p)\right)\right]\mathbf{T}^p + \lambda\left[\overset{\triangledown}{\mathbf{T}}^p + \xi(\mathbf{T}^p \cdot \mathbf{D} + \mathbf{D}\cdot\mathbf{T}^p)\right] = 2\mu_0\mathbf{D} \tag{1505}$$

where ϵ is the Phan-Thien-Tanner parameter and ξ is a model parameter, as in the linear version of the model, but with an exponential factor in place of the first term. This model also reduces to the Johnson-Segalman model when ϵ is zero.

Giesekus-Leonov Model

$$\mathbf{T}^p + \alpha\frac{\lambda}{\mu_0}\mathbf{T}^{p2} + \lambda\overset{\triangledown}{\mathbf{T}}^p = 2\mu_0\mathbf{D} \tag{1506}$$

where α is the Giesekus-Leonov parameter or mobility factor, $0 \leq \alpha \leq 0.5$.

eXtended Pom-Pom Model

The single-equation eXtended Pom-Pom model (XPP) is used for flows of branched polymers in complex cases involving both extension and shear flow. It cannot be derived from the general model.

$$f(\Lambda, \mathbf{T}^p)\mathbf{T}^p + \lambda_b\overset{\triangledown}{\mathbf{T}}^p + G(f(\Lambda, \mathbf{T}^p) - 1)\mathbf{I} + \frac{\alpha}{G}(\mathbf{T}^p \cdot \mathbf{T}^p) = 2\mu_0\mathbf{D} \tag{1507}$$

The function $f(\Lambda, \mathbf{T}^p)$ is given by

$$f(\Lambda, \mathbf{T}^p) = \frac{2}{\epsilon}e^{\frac{2}{Q}(\Lambda - 1)}\left(1 - \frac{1}{\Lambda^2}\right) + \frac{1}{\Lambda^2}\left(1 - \frac{\alpha\ tr(\mathbf{T}^p \cdot \mathbf{T}^p)}{3\,G^2}\right) \tag{1508}$$

where:

$\Lambda = \sqrt{1 + \frac{tr(\mathbf{T}^p)}{3G}}$ is the stretch factor for the material

$\epsilon = \frac{\lambda_s}{\lambda_b}$, where λ_s and λ_b are the relaxation times for the backbone stretch and backbone tube orientation of the polymer
G is the shear modulus
α is the anisotropy parameter
Q is the number of arms on the polymer molecule

All viscoelastic constitutive models are implemented as multi-mode models with:

$$\mathbf{T}^p = \sum_{i=1}^{N} \mathbf{T}_i \tag{1509}$$

and N is the number of modes.

Time-Temperature Superposition

The time-temperature superposition concept can be used to determine the temperature-dependent viscosity of polymeric liquids from a known viscosity at a reference temperature.

If the flow is non-isothermal, the temperature dependence of viscosity can be accounted for by using a shift factor a_T.

When the simulation is isothermal, $a_T = 1$. For polymers:

$$a_T = \frac{\mu_T}{\mu_{T_0}}$$

where μ is the viscosity at temperature T and T_0.

The shift factor can be calculated by the Arrhenius model, the Nahme model, or the Williams-Landel-Ferry (WLF) model.

For the Arrhenius model:

$$\log\left(a_T\right) = \frac{E_a}{R}\left(\frac{1}{T} - \frac{1}{T_0}\right) \tag{1510}$$

For the Nahme model:

$$\log\left(a_T\right) = -\frac{E_a}{RT_0^2}\left(T - T_0\right) \tag{1511}$$

where:

E_a universal gas constant
R activation energy
T_0 reference temperature in K

For the Williams-Landel-Ferry (WLF) model:

$$\log\left(a_T\right) = \frac{C_1\left(T - T_0\right)}{C_2 - \left(T - T_0\right)} \tag{1512}$$

where C_1 and C_2 are positive constants that depend on the material and the reference temperature. If the reference temperature is taken as the glass transition temperature T_g then C_1 and C_2 are universal for a given material. For any polymer, approximate values are $C_1 \approx 15° C$ and $C_2 \approx 50° C$.

Newtonian Liquids

For non-isothermal simulations, the shift factor modifies the viscosity as follows:

$$\mu\left(T\right) = a_T \mu_{ref} \tag{1513}$$

where μ_{ref} is the user-specified reference viscosity.

Generalized Newtonian Fluids

Power Law

The simplest model to describe a shear-rate dependent viscosity behavior is the power law.

The value of the power law exponent n determines the class of the fluid:

$$\mu\left(\dot{\gamma}, T\right) = a_T \cdot k\dot{\gamma}^{n-1} \tag{1514}$$

- $n = 1$ ⇒ Newtonian fluid
- $n > 1$ ⇒ Shear-thickening (dilatant) fluid
- $n < 1$ ⇒ Shear-thinning (pseudo-plastic) fluid

Cross Fluid

The Cross Fluid is defined from [201] as:

$$\mu\left(\dot{\gamma}, T\right) = a_T\left(\mu_\infty + \frac{\mu_0 - \mu_\infty}{1 + \left(\frac{a_T\dot{\gamma}}{\dot{\gamma}_c}\right)^m}\right) \tag{1515}$$

where:

m is the Cross rate constant
μ_0 is the zero-shear viscosity, the viscosity at the Newtonian plateau
μ_∞ is the infinite-shear viscosity
$\dot{\gamma}_c$ is the critical shear strain-rate at which shear-thinning starts

Newtonian behavior is regained for $m = 0$. As m increases towards unity, the degree of shear-thinning becomes stronger.

Carreau-Yasuda Fluid

The non-Newtonian Generalized Carreau-Yasuda Fluid for viscosity comes from [199] and [244]. It is defined as:

$$\mu(\dot{\gamma}, T) = a_T\left(\mu_\infty + \left(\mu_0 - \mu_\infty\right)\left(1 + \left(\lambda a_T \dot{\gamma}\right)^a\right)^{(n-1)/a}\right) \tag{1516}$$

where:

n power constant
λ relaxation time constant
a parameter to control shear-thinning. When $a = 2$, the original Carreau model is regained.

Viscoelastic Fluids

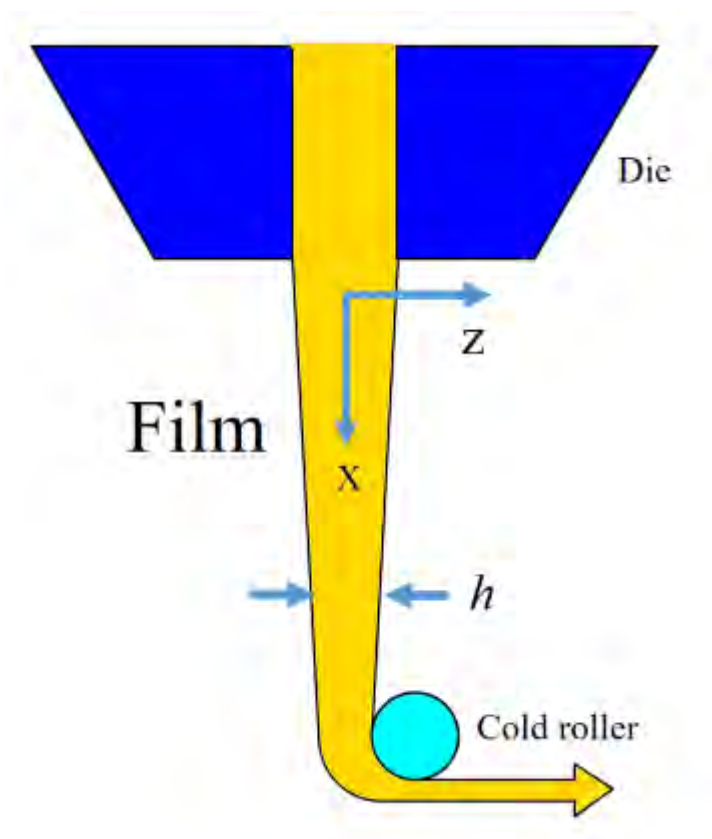
For viscoelastic fluids, both the polymer viscosity μ_0 and the relaxation time constant λ in Eqn. (1502) to Eqn. (1507) are functions of temperature, proportional to the temperature shift factor a_T:

$$\mu\left(T\right) = a_T \mu_{ref} \tag{1517}$$

$$\lambda\left(T\right) = a_T \lambda_{ref} \tag{1518}$$

Film Casting

Film casting is the process of extruding a thin film of molten polymer, then stretching it over a cold roller, to reduce the thickness of the film to the desired value.



The film casting model assumes the thickness of the film is negligible compared to the length and width, and that the variation of the *x* and *y* velocity components in the thickness direction *z* can be neglected. The dynamics of the film are those of an incompressible viscoelastic fluid in a steady state.

The momentum equation for the film is:

$$\rho h\left(\mathbf{v} \cdot \nabla_s \mathbf{v}\right) = \nabla_s \cdot \left(h\left[-\tau_{zz}\mathbf{I} + \tau\right]\right) \tag{1519}$$

where:

- ρ is the density of the polymer.
- h is the film thickness.
- $\mathbf{v}$ is the film velocity.
- ∇_s is the surface gradient, $\nabla_s = \left(\mathbf{I} - \mathbf{e}_z\mathbf{e}_z\right) \cdot \nabla$ where $\mathbf{e}_z$ is the unit normal vector in the *z* direction and $\mathbf{I}$ is the identity tensor.
- $\boldsymbol{\tau}$ is the total hydrodynamic stress tensor, $\boldsymbol{\tau} = 2\mu_s\mathbf{D} + \mathbf{T}$ where μ_s is the solvent viscosity, $\mathbf{D}$ is the strain rate tensor, and $\mathbf{T}$ is the viscoelastic stress tensor.

Since mass is locally conserved in the film, the mass equation for the film is:

$$\nabla_s \cdot h\mathbf{v} = 0 \tag{1520}$$

Assuming no stress in the *z* direction, the energy equation for the film is:

$$\rho C_p \mathbf{v} \cdot \nabla_s T = k\nabla_s^2 T - \frac{q_c + q_r}{h} + \boldsymbol{\tau} : \nabla_s \mathbf{v} \tag{1521}$$

where:

- C_p is the specific heat capacity of the film.
- k is the thermal conductivity of the film.
- q_c and q_r are the convective and the radiation heat flux through the film.

Boundary Conditions

To complete the mathematical model, you specify conditions on the solution domain boundary.

STAR-CCM+ provides a variety of boundary conditions that cover many practical flow situations. Boundary conditions enable the flow to enter or to leave the domain. The flow can be confined or blocked by a wall. By using symmetry or periodic boundary conditions, you can simplify the geometry and reduce the runtime of the simulation.

Contents:

Stagnation Inlet
Velocity Inlet
Mass Flow Inlet
~Pressure Outlet~
~Outlet~
~Symmetry Plane~
Wall
~Free-Stream~

Stagnation Inlet

The stagnation inlet boundary is an inflow condition at which you prescribe the total values for pressure and temperature as well as the flow direction. The stagnation conditions refer to the conditions in an imaginary plenum, far upstream, in which the flow is completely at rest. For incompressible flows, Bernoulli's equation is used to relate total pressure, static pressure, and velocity magnitude. For compressible ideal-gas flows, isentropic relations are used, and characteristic variables help determine the boundary properties of the flow.

Boundary Inputs

For a stagnation inlet boundary, you specify the following variables:

Inputs	Incompressible Equation of State	Compressible Equation of State
supersonic static pressure P_s^{sup}		✓
total pressure P_t	✓	✓
total temperature T_t	✓	✓
inflow direction $\vec{\theta}$	✓	✓

The total pressure P_t and the total temperature T_t are specified with respect to a reference frame. The reference frame is the laboratory, the region, or the local reference frame. The flow direction is either given as normal to the boundary, as individual angle components, or directly as flow direction angles.

Computed Values

For a stagnation inlet boundary, STAR-CCM+ computes the following values:

static pressure P_s

velocity $\mathbf{v}$, computed at the boundary face

static temperature T_s

Compressible and Non-Isothermal Equation of State

The computation of the velocity magnitude differs depending on whether an ideal gas or a non-ideal gas equation of state is used.

- **Ideal Gas**
  STAR-CCM+ computes the negative Riemann invariant by using extrapolated values from the interior of the domain:

$$C_{ext}^{-} = \mathbf{v}^{ext} \cdot \mathbf{n} - 2\frac{c^{ext}}{\gamma - 1} \tag{1522}$$

where:

$\mathbf{v}^{ext}$	boundary velocity that is extrapolated from the interior of the domain
$\mathbf{n}$	normal vector
c^{ext}	velocity of sound that is extrapolated from the interior of the domain
$\gamma = \frac{C_p}{C_v}$	specific heat ratio

Equalizing:

$$C^{-} = \mathbf{v} \cdot \mathbf{n} - 2\frac{c}{\gamma - 1} \equiv C_{ext}^{-} \tag{1523}$$

Substituting the following Eqn. (1524) to Eqn. (1526) into Eqn. (1523) and assuming a perfect gas $\left(C_p = const.\right)$:

$$\mathbf{v} \cdot \mathbf{n} = |\mathbf{v}| \cdot \left(\vec{\theta} \cdot \mathbf{n}\right) \tag{1524}$$

$$C_p T_t = C_p T_s + \frac{|\mathbf{v}|^2}{2} \tag{1525}$$

$$c = \sqrt{\gamma R T_s} = \sqrt{\left(\gamma - 1\right) C_p T_s} = \sqrt{\left(\gamma - 1\right) C_p \left(T_t - \frac{|\mathbf{v}|^2}{2C_p}\right)} \tag{1526}$$

results in a quadratic equation in boundary velocity magnitude $|\mathbf{v}|$ that is solved for $|\mathbf{v}|$.

- **Non-Ideal Gas**
  For a non-ideal gas equation of state, the boundary velocity magnitude is extrapolated from the cell layer adjacent to the boundary in the interior of the domain:

$$|\mathbf{v}| = |\mathbf{v}|^{ext} \tag{1527}$$

For both types of compressible equations of state— the ideal gas law and the non-ideal gas law—the onward procedure for calculating the boundary values is the same.

By using the specified inflow direction, the boundary velocity vector is given by:

$$\mathbf{v} = |\mathbf{v}| \cdot \vec{\theta} \tag{1528}$$

The relation for static enthalpy:

$$H\left(T_s, P_s\right) = H\left(T_t, P_t\right) - \frac{|\mathbf{v}|^2}{2} \tag{1529}$$

Together with the isentropic relation:

$$S\left(T_s, P_s\right) = S\left(T_t, P_t\right) \tag{1530}$$

are solved for static temperature T_s and static pressure P_s, where S is the fluid entropy. For a compressible ideal gas, the isentropic relation is a simple expression that can readily be solved for T_s and P_s.

When local supersonic conditions occur at the boundary, all boundary values are computed from the specified input values, including the supersonic static pressure P_s^{sup}.

The fluid density ρ and the total enthalpy H_t at the boundary face are updated from the equation of state by using the now known boundary values for static pressure P_s, velocity $\mathbf{v}$, and the static temperature T_s:

$$\begin{aligned} \rho &= \rho\left(P_s, T_s\right) \\ H_t &= H_s\left(P_s, T_s\right) + \frac{|\mathbf{v}|^2}{2} \end{aligned} \tag{1531}$$

Incompressible and/or Isothermal Equation of State

The boundary pressure is calculated from the Bernoulli equation:

$$P_s = P_t - \frac{\rho}{2}|\mathbf{v}|^2 \tag{1532}$$

where the boundary velocity magnitude is extrapolated from the interior of the domain:

$$|\mathbf{v}| = |\mathbf{v}|^{ext} \tag{1533}$$

The boundary velocity vector is then obtained by multiplying the velocity magnitude with the specified inflow direction as given by Eqn. (1528).

Incompressible and Non-Isothermal Equation of State

In addition to computing the static pressure and the velocity vector at the boundary according to Eqn. (1532) and Eqn. (1533), for a non-isothermal simulation, the static temperature is calculated as:

$$T_s = T_t - \frac{|\mathbf{v}|^2}{2C_p} \tag{1534}$$

The density at the boundary face is then updated:

$$\rho = \rho\left(T_s\right) \tag{1535}$$

The static and total enthalpies are given by:

$$H_s = H_s(P_s, T_s)$$
$$H_t = H_s + \frac{|\mathbf{v}|^2}{2} \tag{1536}$$

Velocity Inlet

The velocity inlet boundary is usually used as an inflow condition where the distributions of velocity and fluid properties are known. At this inflow boundary, the conditions are prescribed and used to calculate the inlet volume flux as well as the fluxes of momentum and energy.

Boundary Inputs

For a velocity inlet boundary, you specify the following variables:

Inputs	Incompressible Equation of State	Compressible Equation of State (use only for subsonic compressible flow)
velocity $\mathbf{v}_{spec}$	✓	✓
static temperature $T_{s,spec}$	✓	✓

The velocity $\mathbf{v}_{spec}$ is specified with respect to a reference frame, which can be the laboratory, the region, or the local reference frame. The flow direction is either given as normal to the boundary, as individual angle components, or directly as flow direction angles.

Computed Values

The velocity inlet boundary condition is valid for incompressible and (subsonic) compressible flows. Boundary face values for velocity and static temperature are user-specified directly. The static pressure P_s is extrapolated to the boundary face from the interior of the domain.

Incompressible and Compressible Equation of State

The procedure for updating the boundary face values is as follows:

The boundary face velocity is assigned the user-specified velocity at the boundary:

$$\mathbf{v} = \mathbf{v}_{spec} \tag{1537}$$

The boundary face volume flux is computed as:

$$F = \mathbf{v} \cdot \mathbf{a} - G \tag{1538}$$

where:

$\mathbf{a}$	face unit vector
G	face grid flux

The value of the computed boundary face volume flux represents the flow direction. A positive volume flux corresponds to outflow and a negative volume flux corresponds to inflow across a boundary.

For non-isothermal cases, the temperature at the boundary face T_s is updated based on flow direction across the boundary:

$$\begin{aligned} T_s &= T_s^{ext} \quad , \quad \text{if} \quad F > 0 \\ T_s &= T_{s,spec} , \quad \text{if} \quad F < 0 \end{aligned} \tag{1539}$$

where T_s^{ext} is the static temperature that is extrapolated from the interior of the domain and $T_{s,spec}$ is the static temperature that is prescribed at the boundary.

The static pressure P_s is extrapolated from the static pressure at the interior of the domain:

$$P_s = P_s^{ext} \tag{1540}$$

The density ρ and the total enthalpy H_t are calculated as:

$$\begin{aligned} \rho &= \rho\left(P_s, T_s\right) \\ H_t &= H_s\left(P_s, T_s\right) + \frac{\mathbf{v}^2}{2} \end{aligned} \tag{1541}$$

Mass Flow Inlet

You can use the mass flow inlet boundary to specify the known mass flow rate or the mass flux (mass flow rate per unit area) at a boundary. This boundary condition is designed primarily for specifying mass flow inward across the boundary, but also works for specifying mass flowing outward. Mass flow inlet is available for both compressible and incompressible flow regimes.

Boundary Inputs

For a mass flow inlet boundary, you specify the following variables:

Inputs	Incompressible	Compressible
total mass flow rate $\dot{m}_{\text{total}}$ or mass flux $\dot{m}''$	✓	✓
supersonic static pressure P_s^{sup}		✓
total temperature T_t	✓	✓
inflow direction $\vec{\theta}$	✓	✓

You specify the inflow direction as normal to the boundary, as individual angle components, or directly as flow direction angles. Flow conditions are specified relative to the flow condition's reference frame, which can be either the laboratory, the region, or the local reference frame.

Computed Values

For a mass flow inlet boundary, STAR-CCM+ computes the following values:

velocity $\mathbf{v}$, computed at the boundary face
static pressure P_s

static temperature T_s

When you specify the total mass flow rate for the inlet, $\dot{m}_{\text{total}}$, STAR-CCM+ distributes it over all faces of the boundary and calculates a uniform mass flow rate on each face of the boundary:

$$\dot{m}_f = \dot{m}_{\text{total}} \frac{\left|\mathbf{a}_f\right|}{\sum_f \left|\mathbf{a}_f\right|} \tag{1542}$$

where a_f is the boundary face area vector.

When you specify mass flux, $\dot{m}''$, the mass flow rate at each face is calculated as:

$$\dot{m}_f = \dot{m}'' \cdot \left|\mathbf{a}_f\right| \tag{1543}$$

The static pressure at the boundary is extrapolated from the interior of the domain:

$$P_s = P_s^{ext} \tag{1544}$$

Compressible Equation of State

When the inflow is supersonic, the static pressure is updated with the user-specified supersonic static pressure at the boundary:

$$P_s = P_s^{\text{sup}} \tag{1545}$$

The static temperature at the boundary is calculated based on flow conditions:

- Under inflow conditions, that is, mass flow rate $\dot{m}_f \geq 0$, the static temperature at the boundary, T_s, is computed by using the total enthalpy (temperature) relation:

$$H_t = H_s + \frac{|v|^2}{2} \quad , \quad C_p T_t = C_p T_s + \frac{|v|^2}{2} \tag{1546}$$

  where H_t is the total enthalpy, H_s is the static enthalpy, and C_p is the specific heat.

  This yields a quadratic equation that is solved for T_s:

$$T_s = T_t - \frac{(\rho\, v)^2}{2C_p} \cdot \left(\frac{RT_s}{P_s}\right)^2 \tag{1547}$$

  Where:

$$\rho\, v = \frac{\dot{m}_f}{\left|\vec{\theta} \cdot \mathbf{a}_f\right|} \tag{1548}$$

- Under outflow conditions, that is, mass flux $\dot{m}_f < 0$, T_s is extrapolated from the interior of the domain:

$$T_s = T_s^{ext} \tag{1549}$$

The fluid density is updated from the equation of state by using the now known boundary values for static pressure P_s and static temperature T_s:

$$\rho = \rho\left(P_s, T_s\right) \tag{1550}$$

The velocity vector at the boundary face is calculated as:

$$\mathbf{v} = \frac{\dot{m}_f}{\rho\left(\vec{\theta} \cdot \mathbf{a}_f\right)}\vec{\theta} \tag{1551}$$

With the velocity from Eqn. (1551), the total enthalpy at the boundary is given by:

$$H_t = H_s\left(P_s, T_s\right) + \frac{|\mathbf{v}|^2}{2} \tag{1552}$$

where the static enthalpy H_s is calculated based on the equation of state.

Incompressible Equation of State

The density at the boundary is updated from the equation of state by using the static temperature from the previous iteration:

$$\rho = \rho\left(P_s, T_s^*\right) \tag{1553}$$

Inserting the density from Eqn. (1553), the velocity vector is calculated from the mass flux as:

$$\mathbf{v} = \frac{\dot{m}_f}{\rho\left(\vec{\theta} \cdot \mathbf{a}_f\right)}\vec{\theta} \tag{1554}$$

Then, the static temperature at the boundary is recomputed based on flow conditions:

- Under inflow conditions, that is, mass flow rate $\dot{m}_f \geq 0$:

$$T_s = T_t - \frac{|\mathbf{v}|^2}{2C_p} \tag{1555}$$

- Under outflow conditions, that is, mass flow rate $\dot{m}_f < 0$, T_s is extrapolated from the interior of the domain:

$$T_s = T_s^{ext} \tag{1556}$$

The total enthalpy is given by:

$$H_t = H_s\left(P_s, T_s\right) + \frac{|\mathbf{v}|^2}{2} \tag{1557}$$

where the static enthalpy H_s is calculated based on the equation of state.

~Pressure Outlet~

For this release, see *User Guide* > *Modeling Physics* > *Modeling Flow and Energy* > *Boundary Conditions Formulation* > *Pressure Outlet*.

Theory guide content is planned for a future release.

~Outlet~

For this release, see *User Guide* > *Modeling Physics* > *Modeling Flow and Energy* > *Boundary Conditions Formulation* > *Outlet*.

Theory guide content is planned for a future release.

~Symmetry Plane~

For this release, see *User Guide* > *Modeling Physics* > *Modeling Flow and Energy* > *Boundary Conditions Formulation* > *Symmetry Plane*.

Theory guide content is planned for a future release.

Wall

A wall boundary represents an impermeable surface that confines fluid or solid regions.

You can use the wall boundary conditions to model different types of walls. For viscous flows, the no-slip condition is applied by default. The no-slip conditions means that the fluid sticks to the wall and moves with the same velocity as the wall. Thus, for a stationary wall, the fluid has zero velocity at the wall. You can also model the motion of a wall by specifying a tangential velocity component. Depending on the geometry, you can model a translational or rotational motion of the wall in the plane of the boundary. For the no-slip condition, this means that the fluid moves at the same velocity as the specified tangential velocity component. Alternatively, the wall can be modelled as a slip wall representing an impenetrable but traction-free surface.

In the case of turbulent flow, different near-wall treatments to compute shear are employed depending on the turbulence model. These different approaches are algebraic wall functions, two-layer models, or low Reynolds number models.

When you take heat transfer to or from a wall into account, that is, you solve for temperature in a fluid and/or a solid, different thermal specifications are available that determine the heat flux-temperature relationship at the wall.

Boundary Inputs

For a wall boundary, depending on your thermal and tangential velocity specification, you specify the following variables:

Inputs
(relative) velocity $\mathbf{v}_{\text{spec}}$
(relative) wall rotation $\boldsymbol{\omega}$
static temperature T_{spec}

In the case of heat transfer, you specify additional boundary input variables.

The shear stress specification determines whether you model a no-slip wall or a slip wall. The tangential velocity specification sets the velocity of the wall, which in turn influences the computation of the fluid velocity.

The tangential velocity of a wall boundary is specified relative to the reference frame, which can be either the laboratory, the region, or the local reference frame.

Computed Values

For a wall boundary, STAR-CCM+ computes the following values:

Wall pressure P_s, computed at the boundary face

Wall fluid velocity $\mathbf{v}$, computed at the boundary face

Static temperature at the wall T_s, computed at the boundary face

The static pressure at the wall is linearly extrapolated from the interior of the domain:

$$P_s = P_s^{\text{ext}} \tag{1558}$$

No-Slip Wall

On a no-slip wall, the fluid moves at the same velocity as the wall. In other words, the fluid velocity relative to the wall velocity is zero. Depending on the tangential velocity specification, the fluid velocity on the wall is computed differently:

- **Fixed**

$$\mathbf{v} = 0 \tag{1559}$$

- **Vector**

$$\mathbf{v} = \mathbf{v}_{\text{spec}} \tag{1560}$$

- **Rotation Rate**

$$\mathbf{v} = \Omega \times \mathbf{r} \tag{1561}$$

where Ω is the rotational speed vector that is given by:

$$\Omega = \boldsymbol{\omega} \times \mathbf{a} \tag{1562}$$

where $\boldsymbol{\omega}$ is the user-specified angular velocity in rad/s and $\mathbf{a}$ is the axis vector.

$\mathbf{r}$ is the position vector that is defined as:

$$\mathbf{r} = \mathbf{r}_f - \mathbf{r}_o \tag{1563}$$

where $\mathbf{r}_f$ is the location of the boundary face and $\mathbf{r}_o$ is the location of the rotation axis origin.

The boundary fluid velocity is then corrected to match the normal grid flux at the boundary:

$$\mathbf{v} = \mathbf{v} - \left(\mathbf{v} \cdot \mathbf{a}_f - G\right)\frac{\mathbf{a}_f}{\left\|\mathbf{a}_f\right\|^2} \tag{1564}$$

This correction ensures consistency between the normal component of wall fluid velocity and the grid-flux.

Slip Wall

On a slip wall, the fluid slides along the wall without any shear forces. The fluid velocity does not correspond to the wall velocity and is computed as:

$$\mathbf{v} = \mathbf{v}^{\text{ext}} - \left(\mathbf{v}^{\text{ext}} \cdot \mathbf{a}_f - G\right)\frac{\mathbf{a}_f}{\left\|\mathbf{a}_f\right\|^2} \tag{1565}$$

where $\mathbf{v}^{\text{ext}}$ is the fluid velocity extrapolated from the interior of the domain.

The computation of the wall fluid temperature depends on the thermal specification at the wall:

Adiabatic

An adiabatic wall does not permit heat transfer across the boundary. The fluid temperature at the wall:

$$T_s = T_s^{\text{ext}} \tag{1566}$$

as well as density and total enthalpy:

$$\begin{aligned} \rho &= \rho^{\text{ext}} \\ H_t &= H_t^{\text{ext}} \end{aligned} \tag{1567}$$

are extrapolated from the interior of the domain.

Temperature

The temperature at the wall can be set to a user-specified temperature:

$$T_s = T_{\text{spec}} \tag{1568}$$

Convection

The convection wall boundary takes into account a convective heat flux from the environment to the external side of the boundary. The static fluid temperature at the wall T_s is computed as part of the solution of the governing fluid flow and energy equations.

Environment

The environment option at a wall boundary caters for modeling both convective and radiative heat transfer between the environment and the external side of the boundary. The fluid temperature at the wall T_s is computed as part of the solution of the governing fluid flow and energy equations.

Heat Flux

For this thermal specification, you specify the heat flux density at a wall. The fluid temperature at the wall T_s is computed as part of the solution of the governing fluid flow and energy equations.

~Free-Stream~

For this release, see *User Guide* > *Modeling Physics* > *Modeling Flow and Energy* > *Boundary Conditions Formulation* > *Free-Stream*.

Theory guide content is planned for a future release.

Gravity

Forces due to gravity often play an important role in the momentum balance for fluids and solids. The body forces due to gravity are added to the momentum equations as a source term.

$$\mathbf{f}_g = \rho \mathbf{g} \tag{1569}$$

where g is the gravity vector.

When the fluid density ρ varies with temperature, the gravity vector acts on the density and induces a natural convection flow.

The buoyancy forces due to a non-uniform density field are considered significant when the ratio of the Grashof (Gr) and the Reynolds (Re) numbers approaches or exceeds unity:

$$\frac{Gr}{\mathrm{Re}^2} = \frac{\rho \beta \Delta T |g| L}{\rho v^2} \tag{1570}$$

where β is the thermal expansion coefficient.

For natural convection flows, you can use the Boussinesq model to simulate buoyancy effects instead of using a temperature-dependent density when the temperature differences are small. The density used in the Boussinesq model is constant in all solved equations and only the buoyancy source term varies.

Working Pressure

When the Gravity model is active, the STAR-CCM+ working pressure becomes the piezometric pressure, in which the reference altitude (which you set) is taken into account. The piezometric pressure is given as:

$$p_{piezo} = p_{static} - \rho_{ref}\, \mathbf{g} \cdot \left(\mathbf{x}\text{-}\mathbf{x}_0\right) \tag{1571}$$

For both variable and constant density flows it is necessary to set the reference altitude $\mathbf{x}_0$, which is used in the calculation of piezometric pressure as shown above. The reference altitude is the point at which the piezometric and static pressure are considered equal.

When setting initial or boundary pressures in simulations for which the Gravity model is active, you are setting the piezometric pressure value. STAR-CCM+ calculates the corresponding static pressure.

Variable Density Flows

For variable density flows, the buoyancy source term is:

$$\mathbf{f}_g = \left(\rho - \rho_{\mathrm{ref}}\right)\mathbf{g} \tag{1572}$$

where ρ_{ref} is the reference density.

For VOF multiphase flows the density variation could be very significant (several orders of magnitude), and result in numerical instabilities. To obtain a robust and stable numerical solution in such cases, set the reference density to zero or to the density of the lightest phase.

Boussinesq Model

You can enhance convergence of natural convection simulations by using the Boussinesq model. If the constant density ρ_{ref} is used in all terms of the governing equations by eliminating ρ through the Boussinesq approximation $\rho = \rho_{ref}(1 - \beta\Delta T)$ from the buoyancy term, the buoyancy source term is approximated as:

$$\mathbf{f}_g = \rho_{ref}\ \mathbf{g}\ \beta\left(T_{\mathrm{ref}} - T\right) \tag{1573}$$

where T_{ref} is the operating temperature.

This approximation is valid only for small temperature differences and consequently for small density variations. The Boussinesq approximation is valid if $\beta\left(T - T_0\right) \ll 1$.

Fluid Flow Bibliography

[194] *Bingham, E.C. 1916 '"An Investigation of the Laws of Plastic Flow" .US Bureau of Standards Bulletin, 13, pp. 309-353.*

[195] *Bird, R.B., Armstrong, R.C, Hassager, Ole. 1987. Dynamics of Polymeric Liquids, I-II. John Wiley & Sons, New York.*

[196] *Bird, R.B., Stewart, W.E., and Lightfoot, E.N. 1966. "Transport Phenomena". John Wiley & Sons, New York.*

[197] *Baltussen, M.G.H.M., et al. 2010. "Anisotropy parameter restrictions for the eXtended Pom-Pom model", J. Non-Newtonian Fluid Mech., 165, pp. 1047-1054.*

[198] *Burton,T., Jenkins, N., Sharpe, D., Bossanyi, E. 2011. "Wind Energy Handbook", 2nd ed., Wiley.*

[199] *Carreau, P.J. 1972. "Rheological equations from molecular network theories", J. Rheol., 16(1), pp. 99-127.*

[200] *Chima, R.V., and Liou, M.-S. 2003. "Comparison of the AUSM+ and H-CUSP Schemes for Turbomachinery Applications", NASA/TM-2003-212457.*

[201] *Cross, M.M. 1965. "Rheology of non-Newtonian fluids: A new flow equation for pseudoplastic systems", J. Colloid Sci., 20(5), pp. 417-437.*

[202] *Demirdzic, I., Lilek, Z., and Peric, M. 1993. "A collocated finite volume method for predicting flows at all speeds", Int. J. for Numerical Methods in Fluids, 16, pp. 1029-1050.*

[203] *Demirdzic, I. and Muzaferija, S. 1995. "Numerical method for coupled fluid flow, heat transfer and stress analysis using unstructured moving meshes with cells of arbitrary topology", Comput. Methods Appl. Mech. Eng., pp. 1-21.*

[204] *Escudier, M.P., Poole, R.J., Dales, C., Nouar, C., Desaubry, C., Gresham, L. and Pullum, L. 2005. Observations of asymmetrical flow behaviour in transitional pipe flow of yield-stress and other shear thinning liquids J. Non-Newtonian Fluid Mechanics, 127, pp. 143-155.*

[205] *Ferziger, J.H. and Peric, M. 2002. "Computational Methods for Fluid Dynamics", 3rd rev. ed., Springer-Verlag, Berlin.*

[206] *Ferry, John D., 1980. Viscoelastic Properties of Polymers, 3rd edition, Wiley.*

[207] *Giles, M.B. 1988. "Non-reflecting boundary conditions for the Euler equations", CFDL Report 88-1, MIT Dept. of Aero. and Astro.*

[208] *Giles, M.B. 1990. "Non-reflecting boundary conditions for Euler equation calculations", AIAA Journal, 28(12):2050-2058.*

[209] *Giles, M.B. 1991 “UNSFLO: A numerical method for unsteady flow in turbomachinery”, Gas Turbine Laboratory Report GTL 205, MIT Dept. of Aero. and Astro.*

[210] *Gnoffo, P. A., Gupta, R. N., and Shinn, J. L. 1989. “Conservation Equations and Physical Models for Hypersonic Air Flows in Thermal and Chemical Nonequilibrium”, NASA TP-2867.*

[211] *Graboski, M. S. and Daubert, T. E. 1978. “A Modified Soave Equation of State for Phase Equilibrium Calculations: Hydrocarbon Systems”, Ind. Eng. Chem. Process Des. Dev., 17(4), pp. 443-448.*

[212] *Graboski, M. S. and Daubert, T. E. 1978. “A Modified Soave Equation of State for Phase Equilibrium Calculations: Systems Containing CO2, H2S, N2, and CO”, Ind. Eng. Chem. Process Des. Dev., 17(4), pp. 448-454.*

[213] *Gupta, R. N., Lee, K. P., Thompson, R. A., and Yos, J. M. 1991. “Calculations and Curve Fits of Thermodynamic and Transport Properties for Equilibrium Air to 30000 K”, NASA STI/Recon Technical Report N, 92, 11285.*

[214] *Hansen, M. 2007. “Aerodynamics of Wind Turbines, 2nd ed. Routledge.*

[215] *Hosangadi, A., Sachdev, J. , Sankaran, V. 2012. "Improved Flux Formulations for Unsteady Low Mach Number Flows". ICCFD7 proceedings.*

[216] *Huang, H., and Ayoub, J. 2006. “Applicability of the Forchheimer Equation for Non-Darcy Flow in Porous Media”. SPE Journal, March 2008, pp. 112-122.*

[217] *International Association for the Properties of Water and Steam. 2007. Revised Release on the IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam, IAPWS Secretariat, Lucerne, Switzerland. (available at www.iapws.org).*

[218] *Issa, R. I. , Gosman, A. D., and Watkins, A. P., 1986. "The Computation of Compressible and Incompressible Recirculating Flows by a Non-Iterative Implicit Scheme," J. Computational Physics, 62(1).*

[219] *Jameson, A., Schmidt, W. and Turkel, E. 1981. “Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes”, AIAA Paper 81-1259.*

[220] *Legat, V., Marchal, J.M. 1992. "Predictions of threedimensional general shape extrudates by an implicit iterative scheme," Int. J. for Numerical Methods in Fluids, 14, 609-625.*

[221] *Liou, M.-S. 1996. “A Sequel to AUSM: AUSM+”, J. Comput. Phys., 129: 364-382.*

[222] *Liou, M.-S. 2001. “Ten Years in the Making--AUSM-Family”, AIAA 2001-2521.*

[223] *Lockwood, B. A. 2012. “Gradient-based Approaches for Sensitivity Analysis and Uncertainty Quantification within Hypersonic Flows”, Ph.D. Dissertation, University of Wyoming.*

[224] *Loehmer, R. 2009. “On Limiters for Minimal Vorticity Dissipation”, AIAA 2009-135.*

[225] *Lynn, N. and Steinhoff, J. 2007. “Large Reynolds Number Turbulence Modeling with Vorticity Confinement”, AIAA 2007-3965.*

[226] *Makkonen, L. 2000. “Models for the growth of rime, glaze, icicles and wet snow on structures.” Phil. Trans. R. Soc. Lond. A, 358, pp. 2913-2939.*

[227] *Manwell, J.F., McGowan, J., and Rogers, A. 2009. “Wind Energy Explained: Theory, Design and Application”, 2nd ed. Wiley.*

[228] *Mathur, S.R. and Murthy, J.Y. 1997. “Pressure-based method for unstructured meshes”, Numerical Heat Transfer, Part B: Fundamentals, 31(2), pp. 195-214.*

[229] *Mathur, S.R. and Murthy, J.Y. 1997. “Pressure boundary conditions for incompressible flow using unstructured meshes”, Numerical Heat Transfer, Part B: Fundamentals, 32(3), pp. 283-298.*

[230] *Oldroyd, J. 1950. " On the Formulation of Rheological Equations of State" , Proceedings of the Royal Society of London, Series A, Mathematical and Physical Sciences, 200, pp. 523-541.*

[231] *Peric, M., Kressler, R., and Scheuerer, G. 1988. “Comparison of finite-volume numerical methods with staggered and colocated grids”, Computers & Fluids, 16(4), pp. 389-403.*

[232] *Poling, B. E., Prausnitz, J. M., and O'Connell, J. P., “The Properties of Gases & Liquids”, 5th Ed., McGraw-Hill, 2001.*

[233] *Rajagopalan, D., Armstrong, R.C., Brown, R.A. 1990. "Finite element methods for calculation of steady viscoelastic flow using constitutive equations with a Newtonian viscosity", J. Non-Newtonian Fluid Mech., 36, pp 159-192.*

[234] *Reid, R. C., Prausnitz, J. M., and Poling, B. E. 1987. “The Properties of Gases & Liquids”, 4th Ed., McGraw-Hill.*

[235] *Saxer, A.P. 1992. “A numerical analysis of three-dimensional inviscid rotor/stator interactions using non-reflecting boundary conditions”, Ph.D. Thesis. MIT Dept. of Aero. and Astro.*

[236] *Soave, G. 1972. “Equilibrium Constants from a Modified Redlich–Kwong Equation of State”, Chem. Eng. Sci., 27, pp. 1197-1203.*

[237] *Steinhoff, J., Lynn, N. and Wang, L. 2005. “Computation of High Reynolds Number Flows Using Vorticity Confinement: I. Formulation”, UTSI Preprint.*

[238] *Turkel, E. 1987. “Preconditioned methods for solving the incompressible and low speed compressible equations”, Journal of Computational Physics, 72, pp. 277-298.*

[239] *Verbeeten, W.M.H., Peters, G.W.M., and Baaijens, F.P.T. 2002. "Viscoelastic analysis of complex polymer melt flows using the eXtended Pom-Pom model", J. Non-Newtonian Fluid Mech., 108, pp. 301-326.*

[240] *Weiss, J.M., Maruszewski, J.P., and Smith, W.A. 1999. “Implicit solution of preconditioned Navier-Stokes equations using algebraic multigrid”, AIAA Journal, 37(1), pp. 29-36.*

[241] *Weiss, J.M., and Smith, W.A. 1995. “Preconditioning applied to variable and constant density flows”, AIAA Journal, 33(11), pp. 2050-2057.*

[242] *Rajagopalan, R.G. 1993. “Three Dimensional Analysis of a Rotor in Forward Flight”, Journal of the American Helicopter Society, Vol. 38.*

[243] *Yang, Z. et al. 2000. “Recent Improvements to a Hybrid Method for Rotors in Forward Flight”, AIAA Conference Reno.*

[244] *Yasuda, K. 1979. Investigation of the analogies between viscometric and linear viscoelastic properties of polystyrene fluids. PhD Thesis, Dept. of Chemical Engineering, Massachusetts Institute of Technology.*

Numerical Flow Solution

In the finite volume method, the solution domain is subdivided into a finite number of small control volumes, corresponding to the cells of a computational grid.

Discrete versions of the integral form of the conservation equations are applied to each control volume. The objective is to obtain a set of linear algebraic equations, with the total number of unknowns in each equation system corresponding to the number of cells in the grid. The dependent variable values are stored at the cell center, that is, STAR-CCM+ uses a co-located variable arrangement.

Contents:

Finite Volume Discretization
Finite Element Method for Viscoelastic Fluid Flow
Numerical Flow Solution Bibliography

Finite Volume Discretization

Generally, numerical methods, and this includes the finite-volume method, transform the mathematical model into a system of algebraic equations. This transformation involves discretizing the governing equations in space and time. The resulting linear equations are then solved with an algebraic multigrid solver.

For unsteady problems, the physical time interval to be analyzed is subdivided into an arbitrary number of sub-intervals that are called time-steps.

General Transport Equation

When the appropriate constitutive relations are introduced into the conservation equations a closed set of equations is obtained. All conservation equations can be written in terms of a generic transport equation. By integrating the generic transport equation over a control volume V and applying Gauss's divergence theorem, the following integral form of the transport equation is obtained:

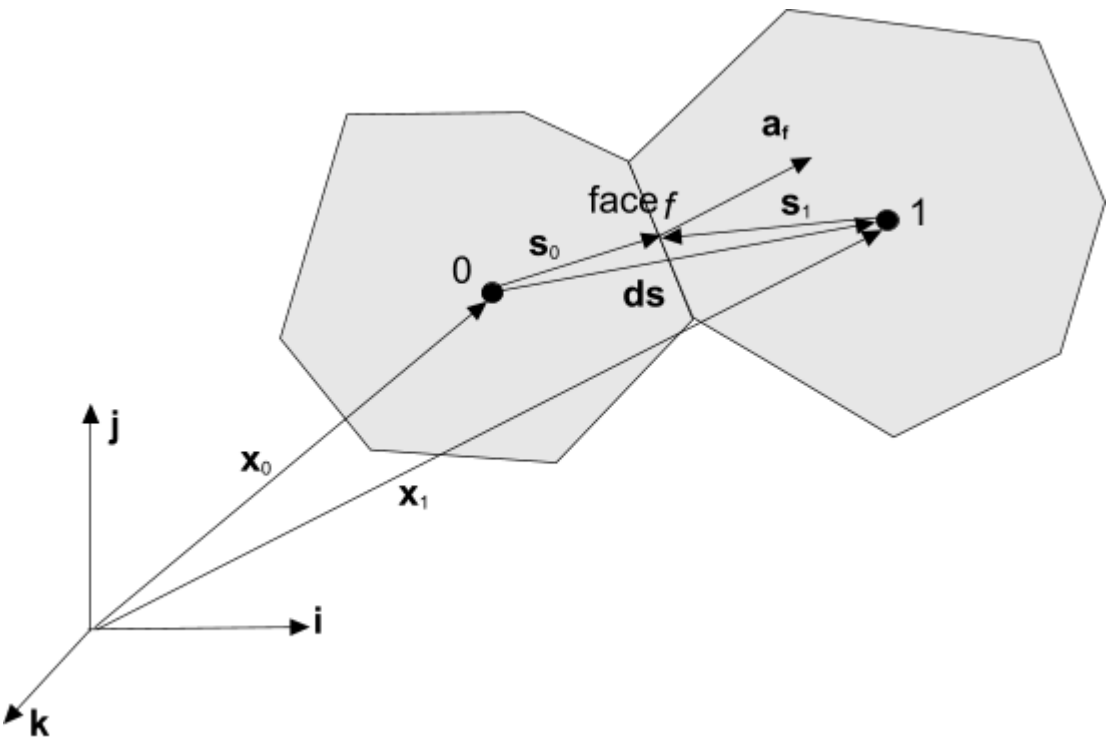
$$\underbrace{\frac{d}{dt}\int_V \rho\phi dV}_{\text{Transient Term}} + \underbrace{\int_A \rho\mathbf{v}\phi \cdot d\mathbf{a}}_{\text{Convective Flux}} = \underbrace{\int_A \Gamma\nabla\phi d\mathbf{a}}_{\text{Diffusive Flux}} + \underbrace{\int_V S_\phi dV}_{\text{Source Term}} \tag{1574}$$

where ϕ represents the transport of a scalar property, A is the surface area of the control volume and $d\mathbf{a}$ denotes the surface vector. By setting ϕ, for example, equal to $1, u, v, w, E,$ or H, Y_i and selecting appropriate values for the diffusion coefficient Γ and source terms, special forms of the partial differential equations for mass, momentum, energy, and species conservation are obtained.

Eqn. (1574) has four distinct terms:

- The transient term, which signifies the time rate of change of fluid property ϕ inside the control volume.
- The convective flux, which expresses the net rate of decrease of fluid property ϕ across the control volume boundaries due to convection.
- The diffusive flux, which corresponds to the net rate of increase of fluid property ϕ across the control volume boundaries due to diffusion.
- The source term, which expresses the generation/destruction of fluid property ϕ inside the control volume.

The following diagram shows two polyhedral computational cells that illustrate the discretization of the generic transport equation:



Surface Integrals

The surface integrals in Eqn. (1574) are evaluated on two levels using quadrature approximations:

- The integral is expressed in terms of variable values at one location on the cell face.

  STAR-CCM+ employs the second-order midpoint rule, that is, the integral is evaluated as the product of the value at the cell face center and the cell face area:

  $$\int_A \mathbf{J}^{\phi} \cdot d\mathbf{a} \approx \sum_f \mathbf{J}_f^{\phi} \cdot \mathbf{a}_f$$

  where $\mathbf{J}^{\phi}$ is either the convective or the diffusive flux of fluid property ϕ, $\mathbf{a}_f$ is the surface area vector of face f of the cell, and $\sum_f$ is the sum over all cell faces of the cell.

- The values at the cell face center are not known and are approximated through interpolation in terms of the values at the cell centers. These interpolation schemes, that is, discretization schemes are described under Convective Flux and Diffusive Flux.

Volume Integrals

The source term in Eqn. (1574) requires integration over the volume of the cell. STAR-CCM+ approximates the volume integral by the product of the mean value of the source term at the cell center and the volume of the cell. This approximation is second-order accurate.

$$\int_V S_{\phi} dV \approx S_{\phi_0} V_0 \tag{1575}$$

To achieve second-order accuracy of these approximations, the cell face center is the area weight center and the cell center is the volume center.

Applying the integration approximations to Eqn. (1574) yields the following semi-discrete transport equation:

$$\frac{d}{dt}\left(\rho\phi V\right)_0 + \sum_f \left[\rho\phi\left(\mathbf{v}\cdot\mathbf{a}\right)\right]_f = \sum_f \left(\Gamma\nabla\phi\cdot\mathbf{a}\right)_f + \left(S_\phi V\right)_0 \tag{1576}$$

Convective Flux

The discretized convective term at a face can be rearranged as follows:

$$\left(\phi\rho\,\mathbf{v}\cdot\mathbf{a}\right)_f = \left(\dot{m}\phi\right)_f = \dot{m}_f\phi_f \tag{1577}$$

where $\dot{m}_f$ is the mass flow rate at the face.

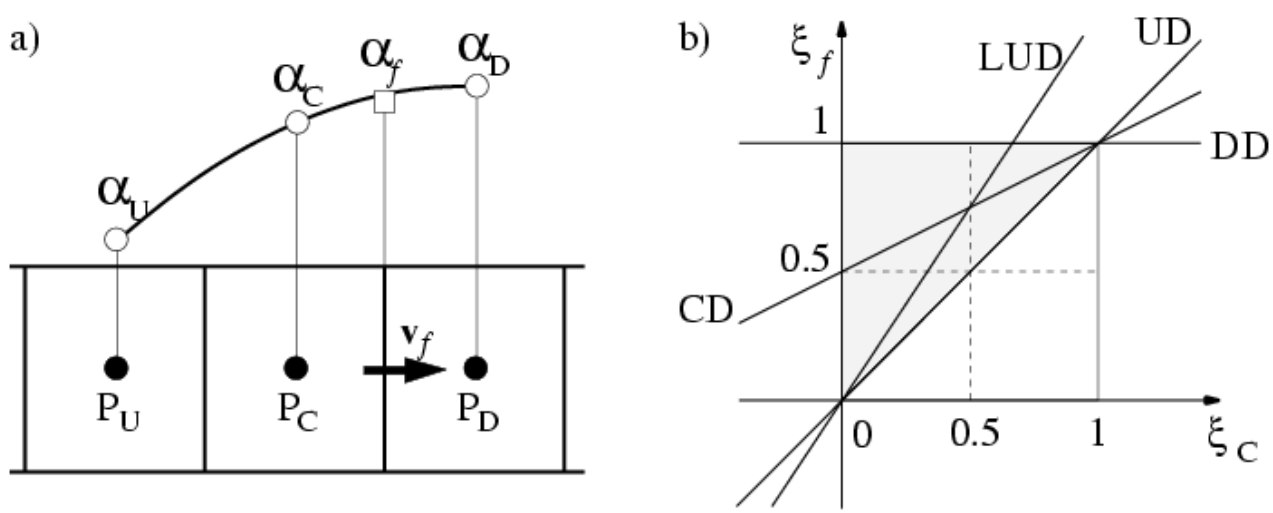
Eqn. (1577) requires the values of fluid property ϕ_f at the face. The manner in which the fluid property face value ϕ_f is computed from the cell values has a profound effect on the stability and accuracy of the numerical scheme.

The Normalized Variable Diagram

The normalized variable diagram (NVD) is useful for analyzing boundedness properties of convective discretization schemes.

Part (a) of the figure below shows three cells in the vicinity of a cell face f, across which the velocity $\mathbf{v}_f$ is known. The nodal variable values are labeled α_D, α_C and α_U, representing the downwind, central, and upwind positions relative to each other.

The convection boundedness criterion is shown in the NVD diagram as Part (b) of the diagram:



The normalized variable $\xi(\mathbf{r},t)$ in the vicinity of face f is defined as:

$$\xi\left(\mathbf{r},t\right) = \frac{\alpha(\mathbf{r},t)-\alpha_U}{\alpha_D-\alpha_U} \tag{1578}$$

The normalized face value:

$$\xi_f = \frac{\alpha_f-\alpha_U}{\alpha_D-\alpha_U} \tag{1579}$$

calculated by any differencing scheme that uses only nodal values of α at points U, C, and D, can be written in the form:

$$\xi_f = \xi_f(\xi_C)$$

where:

$$\xi_C = \frac{\alpha_C - \alpha_U}{\alpha_D - \alpha_U} \tag{1580}$$

To avoid non-physical oscillations in the solution, α_C (and consequently α_f) has to be locally bounded between α_U and α_D, meaning:

$$\alpha_U \leq \alpha_C \leq \alpha_D \tag{1581}$$

or

$$\alpha_D \leq \alpha_C \leq \alpha_U \tag{1582}$$

If this criterion is satisfied at every point in the domain, the entire solution is free of non-physical oscillations.

The boundedness criterion for convection differencing schemes can be presented in the NVD diagram, showing ξ_f as a function of ξ_C, as the hatched region in part (b) of the above Figure (including the line $\xi_f = \xi_C$). It can also be expressed through the following conditions:

- For $0 \leq \xi_C \leq 1$ the bounded region lies above the line $\xi_f = \xi_C$ and below $\xi_f = 1$.
- For $\xi_C < 0$ and $\xi_C > 1$, ξ_f is equal to ξ_C.

NVD is concerned with convective transport alone. If sources or sinks are present, the conditions that are given in Eqn. (1581) and Eqn. (1582) can be violated. The importance of the boundedness criterion is especially clear in the case of variables which have physical bounds; for example, the phase volume fraction cannot become negative, or larger than unity.

First-Order Upwind

The first-order upwind differencing scheme (FOU) approximates the cell face value ϕ_f in the form of a step function. The approximation is of first-order accuracy. Depending on the flow direction, the convective flux is computed as:

$$(\dot{m}\phi)_f = \begin{cases} \dot{m}_f \phi_0 & \text{for } \dot{m}_f \geq 0 \\ \dot{m}_f \phi_1 & \text{for } \dot{m}_f < 0 \end{cases} \tag{1583}$$

where a positive mass flux is leaving the cell and a negative mass flux is entering the cell. This discretization scheme is based on transport properties of the flow: due to convection, ϕ is transported only downstream. If the streamlines of the flow are aligned with the mesh lines, the resulting approximations are good. However, if the streamlines form an angle with the mesh lines, the UDS is inaccurate. These errors are called artificial or numerical diffusion. The advantage of UDS is that it is unconditionally bounded, which is stabilizing and helps the solver achieve robust convergence. It is fairly common practice to obtain an initial solution using the first-order convection and then switch to the second-order one to obtain a more accurate, final converged solution.

Second-Order Upwind

For a second-order upwind (SOU) scheme, the convective flux is computed as:

$$\left(\dot{m}\phi\right)_f = \begin{cases} \dot{m}_f\phi_{f,0} & \text{for } \dot{m}_f \geq 0 \\ \dot{m}_f\phi_{f,1} & \text{for } \dot{m}_f < 0 \end{cases} \tag{1584}$$

where the face values $\phi_{f,0}$ and $\phi_{f,1}$, are linearly interpolated from the cell center values on either side of the face:

$$\begin{aligned} \phi_{f,0} &= \phi_0 + \mathbf{s}_0 \cdot \left(\nabla\phi\right)_{r,0} \\ \phi_{f,1} &= \phi_1 + \mathbf{s}_1 \cdot \left(\nabla\phi\right)_{r,1} \end{aligned}$$

where:

$$\begin{aligned} \mathbf{s}_0 &= \mathbf{x}_f - \mathbf{x}_0 \\ \mathbf{s}_1 &= \mathbf{x}_f - \mathbf{x}_1 \end{aligned}$$

The gradients $(\nabla\phi)_{r,0}$ and $(\nabla\phi)_{r,1}$ of cells 0 and 1 are calculated according to Eqn. (1616) as limited reconstruction gradients.

Central-Differencing

The central-differencing scheme (CDS) approximates the cell face center value by linear interpolation between the two nearest neighbouring cell center values.

For a central-differencing scheme, the convective flux is computed as:

$$\left(\dot{m}\phi\right)_f = \dot{m}_f\left[f\phi_0 + \left(1-f\right)\phi_1\right] \tag{1585}$$

where the linear interpolation factor is defined as:

f is related to the mesh stretching. For a uniform mesh, f has a value of 0.5

Central-differencing is formally second-order accurate. However it is prone to dispersive error and is beset with stability problems for most steady-state situations. The dispersive errors make it problematic for discretizing positive-definite quantities (such as temperature or turbulent kinetic energy) where overshoots cannot be tolerated.

A significant advantage of central-differencing over second-order upwind is that, when used to discretize velocity (not a positive definite quantity) it preserves turbulent kinetic energy. Therefore, it is a useful scheme in large eddy simulation (LES), where upwind schemes cause turbulent kinetic energy to decay unnaturally fast.

Bounded Central-Differencing

For a bounded central-differencing scheme as described in [257] and [280], the convective flux is computed as:

$$\left(\dot{m}\phi\right)_f = \begin{cases} \dot{m}\phi_{FOU} & \text{for} \quad \xi < 0 \text{ or } 1 < \xi \\ \dot{m}\left(\sigma\phi_{CD} + \left(1-\sigma\right)\phi_{SOU}\right) & \text{for} \quad 0 \leq \xi \leq 1 \end{cases} \tag{1586}$$

where:

ϕ_{FOU}	Cell-face center value obtained through first-order upwind interpolation

ϕ_{SOU}	Cell-face center value obtained through second-order upwind interpolation
ϕ_{CD}	Cell-face center value obtained through central-differencing interpolation
ξ	Normalized-Variable Diagram (NVD) value that is computed based on local conditions

A smooth and monotone function of ξ is:

$$\sigma = \sigma(\xi) \tag{1587}$$

which satisfies $\sigma(0) = 0$ and $\sigma(\xi) = 1$ for $\xi_{ubf} \leq \xi$. ξ_{ubf} is called the upwind blending factor whose value ensures a proper balance between the accuracy and robustness of the scheme. Thus, smaller values of ξ_{ubf} provide more accuracy, while larger values increase the robustness of the scheme.

To maintain boundedness, the bounded central-differencing scheme turns into a first-order upwind scheme when the convection boundedness criterion is not satisfied, for example when $\xi < 0 \text{ or } 1 < \xi$. The central-differencing scheme on the contrary, which is formally a second-order accurate scheme, does not do that. Thus, the bounded central-differencing scheme can be more dissipative than the central-differencing scheme—especially on coarser meshes.

Overall, the bounded central-differencing scheme provides a good compromise between accuracy (much improved, when compared with the second-order upwind scheme) and robustness (due to its warranted boundedness). It is the recommended scheme for LES of complex turbulent flows.

Hybrid MUSCL 3rd-Order/Central-Differencing

The scheme is valid for both steady and unsteady simulations, and has one model parameter σ, which is used to control the numerical dissipation in the scheme. As with the bounded central-differencing scheme, this scheme uses a Normalized-Variable Diagram (NVD) value ξ to ensure the boundedness of the scheme by switching to the first-order scheme in regions of non-smooth flows.

When smooth local flow conditions are detected, the scheme is constructed as a blend between a MUSCL 3rd-order upwind and a 3rd-order central-differencing reconstruction scheme.

The convective flux is computed as:

$$(\dot{m}\phi)_f = \begin{cases} \dot{m}\phi_{FOU} & \text{for} \quad \xi < 0 \quad \text{or} \quad \xi > 1 \\ \dot{m}\left(\sigma\phi_{MUSCL3} + (1-\sigma)\phi_{CD3}\right) & \text{for} \quad 0 \leq \xi \leq 1 \end{cases} \tag{1588}$$

The blending factor σ is user-controlled and must be decided based on the physical problem or model.

The MUSCL 3rd-order upwind reconstructed value ϕ_{muscl3} is carefully limited at high speeds so as to not affect the formal order of accuracy while preventing spurious oscillations. The high speed limiting is based on WENO (Weighted Essentially Non-Oscillatory) principles and is only applied to the quadratic part of the face-value reconstruction. WENO-based averaging is computed based on three different stencils.

Note: The ϕ_{muscl3} accuracy is reduced to second order in the regions next to the strong shocks.

The switch between the unlimited and the WENO-limited ϕ_{muscl3} is done based on a Mach-number-based flux-limiter. For incompressible simulations, ϕ_{muscl3} is *unlimited* (the flux limiter is 1), thus only relying on the use of the NVD value to ensure the boundedness of the scheme. Overall, the Hybrid MUSCL 3rd-

order/CD scheme provides improved (reduced) dissipation when compared with both second-order and Bounded Central-Differencing (BCD) schemes. At the same time, it is robust (due to its boundedness) and capable of simulating steady and unsteady flows from incompressible to high-speed compressible regimes.

Hybrid Second-Order Upwind/Central-Differencing

The second-order upwind is the most accurate scheme for steady-state calculations, while the kinetic energy-preserving properties of the central-differencing scheme make it appropriate for large eddy simulation (LES). In this scheme, both approaches are blended as follows::

$$\left(\dot{m}\phi\right)_f = \begin{cases} \dot{m}_f\left\{\sigma\phi_{f,0} + \left(1-\sigma\right)\left[f\phi_{f,0} + \left(1-f\right)\phi_{f,1}\right]\right\} & \text{for } \dot{m}_f \geq 0 \\ \dot{m}_f\left\{\sigma\phi_{f,1} + \left(1-\sigma\right)\left[f\phi_{f,0} + \left(1-f\right)\phi_{f,1}\right]\right\} & \text{for } \dot{m}_f < 0 \end{cases} \tag{1589}$$

where σ is a blending factor that is chosen as appropriate for the flow regime. The face values $\phi_{f,0}$ and $\phi_{f,1}$, are linearly interpolated from the cell values on either side of the face using the reconstruction gradients, as with the second-order upwind scheme.

Hybrid Second-Order Upwind/Bounded-Central

The following hybrid second-order upwind/bounded central scheme is the default for detached eddy simulation (DES) of complex turbulent flows:

$$\left(\dot{m}\phi\right)_f = \dot{m}\left(\sigma\phi_{SOU} + \left(1-\sigma\right)\phi_{BCD}\right) \tag{1590}$$

where at face f, the values of ϕ are ϕ_{SOU} for the second-order upwind scheme and ϕ_{BCD} for the bounded central-differencing scheme. σ is a blending factor that is chosen as appropriate for the flow regime.

Diffusive Flux

The diffusive flux in Eqn. (1574) through internal cell faces of a cell is discretized as:

$$D_f = \left(\Gamma\nabla\phi\cdot\mathbf{a}\right)_f \tag{1591}$$

where Γ is the face diffusivity, $\nabla\phi$ is the gradient of fluid property ϕ, and $\mathbf{a}$ is the surface area vector.

To obtain an accurate second-order expression for an interior face gradient that implicitly involves the cell values ϕ_0 and ϕ_1, the following decomposition is used:

$$\nabla\phi_f = \left(\phi_1-\phi_0\right)\boldsymbol{\alpha} + \overline{\nabla\phi} - \left(\overline{\nabla\phi}\cdot\mathbf{ds}\right)\boldsymbol{\alpha} \tag{1592}$$

where:

$$\begin{aligned} \boldsymbol{\alpha} &= \frac{\mathbf{a}}{\mathbf{a}\cdot\mathbf{ds}} \\ \mathbf{ds} &= \mathbf{x}_1-\mathbf{x}_0 \\ \overline{\nabla\phi} &= \frac{\left(\nabla\phi_0 + \nabla\phi_1\right)}{2} \end{aligned}$$

The diffusion flux at an interior face can then be written as:

$$D_f = \Gamma_f\nabla\phi_f\cdot\mathbf{a} = \Gamma_f\left[\left(\phi_1-\phi_0\right)\vec{\alpha}\cdot\mathbf{a} + \overline{\nabla\phi}\cdot\mathbf{a} - \left(\overline{\nabla\phi}\cdot\mathbf{ds}\right)\vec{\alpha}\cdot\mathbf{a}\right] \tag{1593}$$

where Γ_f is a harmonic average of the cell values.

A similar decomposition is used at a boundary face:

$$D_f = \Gamma_f \nabla \phi_f \cdot \mathbf{a} = \Gamma_f \left[\left(\phi_f - \phi_0 \right) \vec{\alpha} \cdot \mathbf{a} + \nabla \phi_0 \cdot \mathbf{a} - \left(\nabla \phi_0 \cdot \mathbf{ds} \right) \vec{\alpha} \cdot \mathbf{a} \right] \tag{1594}$$

where:

$$\vec{\alpha} = \frac{\mathbf{a}}{\mathbf{a} \cdot \mathbf{ds}} \tag{1595}$$

$$\mathbf{ds} = \mathbf{x}_f - \mathbf{x}_0 \tag{1596}$$

The second and third terms in Eqn. (1593) represent the secondary gradient (or cross-diffusion) contribution. They are essential for maintaining accuracy on non-orthogonal meshes.

The formulation that is presented above assumes that the centroids of cells 0 and 1 lie on opposing sides of the face. It is further assumed that their location is consistent with the convention that the face area vector points out of cell 0. To prevent non-physical solutions, take care to use meshes for which the angle between $\mathbf{a}$ and $\mathbf{ds}$ is not greater than 90 degrees.

To ensure that a valid mesh is being used, a diagnostic tool that is provided in STAR-CCM+ allows this angle (termed skewness angle) to be computed in degrees and stored in adjacent cells. The value that is stored in each cell represents the largest skewness angle for each face of the cell.

As a further diagnostic tool that can be employed to check whether the secondary gradient contribution is destabilizing, these terms can be optionally omitted, in which case Eqn. (1593) reduces to:

$$D_f \approx \Gamma_f \left[\left(\phi_1 - \phi_0 \right) \vec{\alpha} \cdot \mathbf{a} \right] \tag{1597}$$

Similar to interior faces, the second and third terms in Eqn. (1594) represent the boundary secondary gradient (or cross-diffusion) contribution.

As with the interior, these boundary secondary gradients can be optionally neglected as a further diagnostic tool, in which case Eqn. (1594) reduces to:

$$D_f \approx \Gamma \left[\left(\phi_f - \phi_0 \right) \vec{\alpha} \cdot \mathbf{a} \right] \tag{1598}$$

Transient Term

For transient simulations, time is an additional coordinate. Generally, numerical procedures require that this coordinate is discretized in addition to a spatial discretization, that is, the total time interval is subdivided into time-steps. The solution of the governing equations is obtained at different time-levels whereby the solution at time t requires the solutions from previous time-levels. Time integration schemes are distinguished between the number of time-levels they use for integration and on which time-level the fluxes and sources are integrated.

Implicit Time Integration

The Euler implicit scheme, a first-order temporal scheme, approximates the transient term in Eqn. (1574) using the solution at the current time level, $n+1$ and the one from the previous time level, n:

$$\frac{d}{dt} \left(\rho \phi V \right)_0 = \frac{(\rho \phi V)_0{}^{n+1} - (\rho \phi V)_0{}^{n}}{\Delta t} \tag{1599}$$

A second-order temporal scheme discretization of the unsteady term uses the solution at the current time level, $n+1$, as well as the solutions from the previous two time levels, n and $n-1$, as follows:

$$\frac{d}{dt}\left(\rho\phi V\right)_0 = \frac{\left\{\left(\alpha^2-1\right)\left[(\rho\phi V)_0^{n+1}-(\rho\phi V)_0^n\right]+\left[(\rho\phi V)_0^{n-1}-(\rho\phi V)_0^n\right]\right\}}{\alpha(\alpha-1)\Delta t^{n+1}} \tag{1600}$$

where:

$$\begin{aligned}\alpha &= 1+\frac{\Delta t^{n+1}}{\Delta t^n}\\ \Delta t^{n+1} &= t^{n+1}-t^n\\ \Delta t^n &= t^n-t^{n-1}\end{aligned}$$

On the first time-step of a second-order temporal simulation, a first-order discretization is used since only two time levels are available.

Gradients

In addition to variable values, variable gradients are required at cell centers and at cell-face centers for:

- construction of variable values at the cell faces
- secondary gradients calculation for diffusion terms
- pressure gradients calculation for pressure-velocity coupling
- strain-rate and rotation-rate calculations for turbulence models

In STAR-CCM+, the steps that are involved in computing gradients are:

1. Computing the (unlimited) reconstruction gradients. Here, unlimited means that the gradients do not prohibit the reconstructed field variables on the cell faces from exceeding the minimum and maximum values of the neighboring cells. The following methods are available:
   - Hybrid Gauss-Least Squares Method
   - Green-Gauss Method
     - the weighted least-squares method is used for pressure
     - the Gauss method is used for all variables other than pressure
2. Limiting the reconstruction gradients. The limited reconstruction gradients are used to determine scalar values at the cell faces. These scalar values are used in computing flux integrals.
3. Computing the cell gradients from the limited reconstruction gradients. This step is only required for the Green-Gauss method. For the Hybrid Gauss-LSQ method, the unlimited LSQ-based gradients are linear-exact and are used as cell gradients in the diffusive fluxes. This approach is more accurate for cell gradient calculations than the Green-Gauss method.

Hybrid Gauss-Least Squares Method

The unlimited (denoted by superscript u) reconstruction (denoted by subscript r) gradients in cell-0 are computed using the following hybrid formula:

$$\left(\nabla\phi\right)_r^u = \sum_f \left(\phi_n-\phi_0\right)w_f^0 \tag{1601}$$

with

$$w_f^0 = \beta w_f^{lsq} + \left(1-\beta\right) w_f^G \tag{1602}$$

$$w_f^{lsq} = \left[\sum_f \frac{\mathbf{ds}\otimes\mathbf{ds}}{\mathbf{ds}^2}\right]^{-1} \frac{[\mathbf{ds}]}{\mathbf{ds}^2} \tag{1603}$$

$$\mathbf{ds} = \mathbf{x}_n - \mathbf{x}_0 \tag{1604}$$

$$w_f^G = \frac{A_f}{V_0 + V_n} \tag{1605}$$

where $\mathbf{x}_0$ and $\mathbf{x}_n$ are the centroids of cell-0 and that of the neighbor cell n, addressed through face f. ϕ_0 and ϕ_n are data values in cell-0 and its neighbor, A_f is the face f area-vector, V_0 and V_n are the respective cell volumes, and β is the geometric Gauss-LSQ gradient blending factor field function.

The Gauss/LSQ gradient blending factor is used to determine how the two gradient method options are used in calculating the gradient. The following conditions apply to the bending factor in gradient calculations:

- Blending factor of 1 equates to using LSQ method
- Blending factor of 0 equates to using Green-Gauss method
- Blending factor from 0 through 1 equates to using a combination of LSQ and Green-Gauss, with:

$$\nabla\phi = \left(\beta\right)\mathbf{LSQ_grad}\left(\phi\right) + \left(1-\beta\right)\mathbf{GG_grad}\left(\phi\right) \tag{1606}$$

where β is the cell value of the gradient factor, $\mathbf{LSQ_grad}(\phi)$ is the LSQ computed gradient of ϕ and $\mathbf{GG_grad}(\phi)$ is the Green-Gauss computed gradient of ϕ.

If the Green-Gauss method is selected for gradient calculations, the Gauss-LSQ gradient blending factor is not used.

The cell value of the blending factor is computed as the minimum of the following:

- Blending value is 1 if the least-squares tensor eigenvalues ratio of the cell is larger than the specified minimum value (set using Least-Squares Quality Criterion property)
- Blending value is 1 if the tangent of the skewness angle is smaller than the Normalized Flat Cells Curvature Factor times the aspect ratio of the cell. It is less than 1, with a rapid decrease to zero otherwise (set using the Flat Cell Curvature Criterion property)
- Blending value is 1 if the skewness angle is smaller than the Maximum safe (Positive) skewness angle. Blending value is 0 if the skewness angle is larger than the Minimum unsafe (Positive) skewness angle. Otherwise, the blending value is from 0 through 1 with a linear variation, (Cell Skewness Criterion)
- Blending value is 0 if it is a Chevron-cell and 1 otherwise (Chevron-Cell Criterion)

The Gauss/LSQ gradient blending factor only applies for the Hybrid LSQ-Gauss method.

Weighted Least Squares Method

The unlimited reconstruction gradients in cell-0 are computed using the following weighted least squares formula:

$$\left(\nabla\phi\right)_r^u = \left[\sum_f \frac{\mathbf{ds}\otimes\mathbf{ds}}{\mathbf{ds}\cdot\mathbf{ds}}\right]^{-1} \left[\sum_f \frac{\left(\phi_0-\phi_n\right)\mathbf{ds}}{\mathbf{ds}\cdot\mathbf{ds}}\right] \tag{1607}$$

where

$$\mathbf{ds} = \mathbf{x}_n - \mathbf{x}_0 \tag{1608}$$

with $\mathbf{x}_0$ and $\mathbf{x}_n$ representing the centroids of cell-0 and that of the neighbor cell that is addressed through face f, and ϕ_0 and ϕ_n representing the data values in cell-0 and its neighbor.

Gauss Method

From the divergence theorem of Gauss:

$$\int_V \nabla\phi dV = \int_A \phi d\mathbf{a} \tag{1609}$$

Written in discrete form, this divergence theorem can be used to compute the unlimited reconstruction gradients:

$$\left(\nabla\phi\right)_r^u = \frac{1}{V_0}\sum_f \phi_f \mathbf{a}_f \tag{1610}$$

where the face value is approximated by the arithmetic average of the adjacent cell values:

$$\phi_f = \frac{\phi_0 + \phi_1}{2} \tag{1611}$$

Cell Gradients

Cell gradients are computed using the reconstruction gradients. When using the Hybrid Gauss-LSQ method, the cell gradients are a copy of the reconstruction gradients. For the Green-Gauss Method, the improved estimates of the face values are obtained from the reconstruction gradients. They can in turn be used to obtain better estimates of the cell gradients using the divergence theorem of Gauss:

$$\nabla\phi = \frac{1}{V_0}\sum_f \phi_f \mathbf{a}_f \tag{1612}$$

where the face value is approximated by the arithmetic average of the face values reconstructed from the adjacent cell values:

$$\phi_f = \frac{\phi_{f,0} + \phi_{f,1}}{2} \tag{1613}$$

where, as before:

$$\phi_{f,0} = \phi_0 + \mathbf{s}_0 \cdot \left(\nabla\phi\right)_{r,0} \tag{1614}$$

$$\phi_{f,1} = \phi_1 + \mathbf{s}_1 \cdot \left(\nabla\phi\right)_{r,1} \tag{1615}$$

Gradient Limiters

The problem with simply reconstructing face values from the unlimited reconstruction gradients is that the reconstructed face values can fall outside the range of cell values found in neighboring cells (connected through faces). For this reason, STAR-CCM+ finds the minimum and maximum bounds of the neighboring cell values and uses these to limit the reconstruction gradients.

The following gradient limiters are available:

- Venkatakrishnan
- Min-Mod
- Modified Venkatakrishnan

The face value ($\phi_{f,0}$) reconstructed from the cell-0 value at any face centroid is given by:

$$\phi_{f,0} = \phi_0 + \mathbf{s}_0 \cdot \left(\nabla\phi\right)_{r,0} \tag{1616}$$

where $\mathbf{s}_0 = \mathbf{x}_f - \mathbf{x}_0$, $\mathbf{x}_f$ and $\mathbf{x}_0$ are the face and cell centroids, respectively, and $\nabla\phi_{r,0}$ is the reconstructed face value.

For each cell-0, a limited reconstruction gradient is required, such that the reconstructed face value does not exceed the maximum and minimum of the neighboring cell centroid values, including the value in cell-0. A scale factor α is defined that expresses the ratio of the limited and unlimited values, that is:

$$\left(\nabla\phi\right)_{r,0} = \alpha\left(\nabla\phi\right)^u_{r,0} \tag{1617}$$

For each cell-0, the quantities are defined:

$$\begin{aligned} \phi_0^{\max} &= \max\left(\phi_0, \phi_{\text{neighbors}}\right) \\ \phi_0^{\min} &= \min\left(\phi_0, \phi_{\text{neighbors}}\right) \end{aligned}$$

where $\phi_{\text{neighbors}}$ represents the cell value in each neighbor that has a common face with cell-0.

These quantities can also be defined:

$$\begin{aligned} \Delta_{\max} &= \phi_0^{\max} - \phi_0 \\ \Delta_{\min} &= \phi_0^{\min} - \phi_0 \end{aligned}$$

For each face f of cell-0, Δ_f is defined as:

$$\Delta_f = \phi_{f,0} - \phi_0 = \mathbf{s}_0 \cdot \left(\nabla\phi\right)^u_{r,0} \tag{1618}$$

Now defining:

$$r_f = \begin{cases} \dfrac{\Delta_f}{\Delta_{\max}} & \text{for } \Delta_f > 0 \\ \dfrac{\Delta_f}{\Delta_{\min}} & \text{for } \Delta_f \leq 0 \end{cases} \tag{1619}$$

Venkatakrishnan

The Venkatakrishnan [307] limiter gives for the face:

$$\alpha_f = \frac{2r_f + 1}{r_f\left(2r_f + 1\right) + 1} \tag{1620}$$

Min-Mod

The Min-Mod limiter gives for the face:

$$\alpha_f = \min\left(1, 1/r_f\right) \tag{1621}$$

Modified Vekatakrishnan

The Modified Venkatakrishnan limiter gives for the face:

$$\alpha_f = \frac{r_f(2r_f + 1) + 1}{r_f(r_f(2r_f + 1) + 1) + 1} \tag{1622}$$

The Venkatakrishnan limiter is the default option. In certain cases, the Min-Mod and the modified Venkatakrishnan limiter can be less dissipative than the default option, particular in unsteady simulations for aeroacoustics or when using LES/DES for turbulence modeling.

The cell value of α is given by:

$$\alpha = \min\left(\alpha_f\right) \tag{1623}$$

Total Variation Bounded Gradient

If gradients are excessively limited, convergence can slow or stall. This problem arises most often with the gradients of near-constant fields, or when the limiter cycles between 0 and 1 without settling on a convergence value.

To avoid this problem, Total Variation Bounded gradient limiting can be applied. Reconstructed values vary by an additional difference of

$$\delta = \psi \cdot \max\left(\Delta_{\max} - \Delta_{\min}\right) \tag{1624}$$

where ψ is a fraction between 0 and 1, and $\max\left(\Delta_{\max} - \Delta_{\min}\right)$ is the largest difference between local maximum and local minimum anywhere in the simulation.

The default value of ψ is 0.05. Larger values of ψ mean that gradients are limited less. This can improve accuracy, especially for smooth flow field gradient reconstruction. However, excessively large values can lead to loss of solver robustness and lower accuracy, by producing large over- or under-shoots at flow field discontinuities.

TVB Gradient Limiting is useful for high-fidelity simulations, such as aeroacoustics and turbulence simulations using Large Eddy Simulation (LES) or Detached Eddy Simulation (DES).

Boundary Conditions

The foregoing expressions for the evaluation of the convective and the diffusive fluxes above are valid for all interior cell faces. On faces that coincide with boundaries of the computational domain, boundary conditions are applied. The integrals over the boundary surface are expressed as a function of the known boundary data and cell values from the interior.

In the case of Dirichlet boundary conditions:

- the convective fluxes are calculated by replacing ϕ_f in Eqn. (1577) by the boundary value ϕ_b
- the diffusive fluxes are calculated by replacing ϕ_1 by the boundary value ϕ_b and by replacing $\overline{\nabla\phi}$ by $\nabla\phi_0$ in Eqn. (1593)

In the case of Neumann boundary conditions, the diffusive fluxes are calculated directly with the variable values at the boundary being obtained from the discretized gradient approximations.

Algebraic System of Equations

With $\Delta\phi_p = \phi_p^{k+1} - \phi_p^k$, the algebraic system for the transported variable ϕ is written in *delta* form:

$$\frac{a_p}{\omega}\,\Delta\,\phi_p + \sum_n a_n\,\Delta\,\phi_n = r \tag{1625}$$

where ϕ is the transported variable at iteration $k+1$ (current iteration) or k (previous iteration), respectively. The summation is over all the neighbors n of cell p. The right-hand side r is the residual. The coefficients a_p and a_n are obtained directly from the discretized terms. ω is an under-relaxation factor.

The right-hand side:

$$r = -\left[\frac{d}{dt}(\rho\phi V) + \sum_f (\rho\phi\mathbf{v}\cdot\mathbf{a})_f - \sum_f (\Gamma\nabla\phi\cdot\mathbf{a})_f - S_\phi V\right] = 0 \tag{1626}$$

is termed the residual, and represents the discretized form of the original equation (Eqn. (1576)) at iteration k. By definition, then, the residual becomes zero when the discretized equation is satisfied exactly.

Due to the non-linear nature of the problem, an iterative solution is required. There are two levels of iteration; an outer iteration loop controlling the solution update and an inner loop governing the iterative solution of the linear system. Since the outer iterations are repeated multiple times, it is sufficient to solve the linear system only approximately at each outer iteration. The iterative solution of the linear system is accomplished using Algebraic Multigrid (AMG).

Contents:

Segregated Flow Solver
Coupled Flow Solver
Algebraic Multigrid

Segregated Flow Solver

The segregated flow solver solves the integral conservation equations of mass and momentum in a sequential manner. The non-linear governing equations are solved iteratively one after the other for the solution variables such as u, v, w, p.

The segregated solver employs a pressure-velocity coupling algorithm where the mass conservation constraint on the velocity field is fulfilled by solving a pressure-correction equation. The pressure-correction equation is constructed from the continuity equation and the momentum equations such that a predicted velocity field is sought that fulfills the continuity equation, which is achieved by correcting the pressure. This method is also called a predictor-corrector approach. Pressure as a variable is obtained from the pressure-correction equation.

STAR-CCM+ implements two pressure-velocity coupling algorithms:

- SIMPLE
- PISO

The segregated solver has its roots in constant-density flows. Although it can handle mildly compressible flows and low Rayleigh number natural convection, it is not suitable for shock-capturing, high Mach number, and high Rayleigh number applications.

To a large extent, the methods that are presented are based on [249], [259], [260], [288], [289], and [292].

Discretization of the Momentum Equation

The momentum equations are discretized in a similar way to the scalar transport equation as described in Generic Transport Equation by setting $\phi = u, v, w$:

$$\frac{\partial}{\partial t}(\rho \mathbf{v} V)_0 + \sum_f \left[\rho \mathbf{v}\mathbf{v} \cdot \mathbf{a}\right]_f = -\sum_f \left(p\,\mathbf{I} \cdot \mathbf{a}\right)_f + \sum_f \mathbf{T} \cdot \mathbf{a} \tag{1627}$$

The convective and diffusive fluxes are discretized as described in Convective Flux and Diffusive Flux. The discretization of the pressure gradient requires special attention.

Face Pressure Evaluation

Pressure appears as pressure gradient in the momentum equations. In order to compute the pressure gradient term in Eqn. (1627), the pressure must be evaluated at each face.

$$p_f = \frac{\overline{a_0} p_{f0} + \overline{a_1} p_{f1}}{\overline{a_0} + \overline{a_1}} \tag{1628}$$

where $\overline{a_0}$ and $\overline{a_1}$ are the average of the momentum coefficients for all components of momentum for cells 0 and 1, respectively. p_{f0} and p_{f1} are interpolated from cell values and reconstruction gradients according to Eqn. (1616).

When the *delta-V dissipation* expert property is turned on, a dissipation term is incorporated into the face pressure evaluation, as follows:

$$p_f = \frac{\overline{a_0} p_{f0} + \overline{a_1} p_{f1}}{\overline{a_0} + \overline{a_1}} + \frac{2\overline{a_0 a_1}}{\overline{a_0} + \overline{a_1}} \cdot \frac{\left(\mathbf{v}_{f0} - \mathbf{v}_{f1}\right) \cdot \mathbf{a}}{4(\mathbf{a} \cdot \mathbf{a})} \tag{1629}$$

where $\mathbf{v}_{f0}$ and $\mathbf{v}_{f1}$ are also evaluated according to Eqn. (1616).

At boundaries, the cell 1 contribution comes from a fictitious “ghost cell” in which the velocity has been reflected about the boundary face. Therefore Eqn. (1628) becomes:

$$p_f = p_{f0} \tag{1630}$$

With *delta-V dissipation* on, Eqn. (1629) becomes:

$$p_f = p_{f0} + \overline{a_0} \cdot \frac{\mathbf{v}_{f0} \cdot \mathbf{a}}{2(\mathbf{a} \cdot \mathbf{a})} \tag{1631}$$

at a boundary.

Pressure-Correction Equation

The pressure-velocity coupling is achieved by re-writing the continuity equation in terms of a mass flux correction $\dot{m}'_f$:

$$\sum_f \dot{m}_f = \sum_f \left(\dot{m}_f^* + \dot{m}'_f\right) = 0 \tag{1632}$$

The uncorrected face mass flux $\dot{m}_f^*$ is computed after the discrete momentum equations have been solved. The momentum equations are solved with a guessed pressure field $p*$, which does not satify continuity. The mass flow correction $\dot{m}'_f$ is required to satisfy continuity.

The uncorrected mass flux at an interior face can be written in terms of the cell variables as follows:

$$\dot{m}_f^* = \rho_f \mathbf{a} \cdot \left(\frac{\mathbf{v}_0^* + \mathbf{v}_1^*}{2}\right) - \Upsilon_f \tag{1633}$$

where $\mathbf{v}_0^*$ and $\mathbf{v}_1^*$ are the cell velocities after the discrete momentum equations have been solved. Linear interpolation of the velocities at the cell centers to the face results in unphysical checkerboarding of pressure. To prevent this checkerboarding, the Rhie-and-Chow dissipationΥ_f is introduced:

$$\Upsilon_f = Q_f\left(p_1 - p_0 - \overline{\nabla p_f^*} \cdot d\mathbf{s}\right) \tag{1634}$$

where:

$$Q_f = \rho_f \left(\frac{V_0 + V_1}{\overline{a}_0 + \overline{a}_1}\right) \boldsymbol{\alpha} \cdot \mathbf{a} \tag{1635}$$

where:

V_0, V_1	Cell volumes for cell 0 and cell 1
$\overline{a}_0, \overline{a}_1$	Average of the momentum coefficients for all components of momentum for cells 0 and 1
p_0	Pressure for cell 0
p_1	Pressure for cell 1
$\overline{\nabla p_f^*}$	Volume-weighted average of the cell gradients of pressure, computed using a volume-based interpolation between the gradient values of the two cells

Unsteady Dissipation Correction

For either coefficient $\overline{a} = \overline{a}_0 \text{ or } \overline{a}_1$, when unsteady dissipation corrections are enabled, the coefficients $\overline{a}$ are modified based on an *Acoustic CFL*.

- Let *Acoustic CFL* be $CFL_a = c\frac{\Delta t}{\Delta x}$ where c is the speed of sound, Δt is the simulation time-step, and Δx is the local grid size.
- Let limCFL be the Limit Acoustic CFL value specified by the user.

Then:

- If $\mathrm{CFL} > \mathrm{limCFL}$, $\overline{a}$ is used without modification.
- Otherwise, $\overline{a}$ is replaced with $\overline{a}'$, as follows:

$$\overline{a}' = \overline{a}_u\left(1 + \frac{\mathrm{CFL}_r\overline{a}_{uns}}{\max\left(\overline{a}_u, 1 \times 10^{-20}\right)}\right)$$

where:

- $\overline{a}_u = \overline{a} - \overline{a}_{uns}$ and
  $\overline{a}_{uns} = \frac{3}{2}\rho\frac{\mathrm{Vol}}{\Delta t}$
- $\mathrm{CFL}_r = \min\left(1.0, \frac{\mathrm{CFL}}{\mathrm{limCFL}}\right)$

p_L and p_R are the pressures values computed by linear reconstruction at two symmetric points $(\mathbf{x}_L, \mathbf{x}_R)$ with regard to the face-centroid, in the direction of face normal.

$\mathbf{x}_L = \mathbf{x}_f - d_s\mathbf{n}$ and $\mathbf{x}_R = \mathbf{x}_f + d_s\mathbf{n}$

where:

- $\mathbf{n}$ is the unit vector associated with area vector $\mathbf{a}_f$.
- $\mathbf{x}_f$ is the position vector of the face centroid.
- d_s is a scalar distance, where:
  - If $\max\left(\frac{V_1}{V_0}, \frac{V_0}{V_1}\right) > 10$, then $d_s = \frac{0.5}{|\boldsymbol{\alpha}|}$.
  - Otherwise, $d_s = \min(d_{s0}, d_{s1})$, where:
    - $d_{s0} = d\mathbf{x}_0 \cdot \mathbf{n}$ (the projection of $d\mathbf{x}_0 = \mathbf{x}_f - \mathbf{x}_0$ onto the face area normal)
    - $d_{s1} = d\mathbf{x}_1 \cdot \mathbf{n}$ (the projection of $d\mathbf{x}_1 = \mathbf{x}_f - \mathbf{x}_1$ onto the face area normal).

Density Correction

For compressible flow, the density correction ρ' is introduced:

$$\begin{aligned}\dot{m}_f &= (\rho + \rho')_f\left(\mathrm{v}^*_{fn} + \mathrm{v}'_{fn}\right)|\mathbf{a}| \\ &= \left(\rho_f v^*_{fn} + \rho'_f v^*_{fn} + \rho_f \mathrm{v}'_{fn} + \rho'_f \mathrm{v}'_{fn}\right)|\mathbf{a}|\end{aligned} \tag{1636}$$

where the subscript “fn” denotes the face-normal component.

$$\rho_f \mathrm{v}'_{fn}|\mathbf{a}| \equiv -Q_f\left(p'_1 - p'_0\right) \tag{1637}$$

where p'_1 and p'_0 are the cell pressure corrections, and:

$$\rho'_f \mathrm{v}^*_{fn}|\mathbf{a}| = \frac{\dot{m}^*_f}{\rho_f}\left(\frac{\partial\rho}{\partial p}\right)_T \mathrm{p}'_{\mathrm{upwind}} \tag{1638}$$

where:

$$\mathrm{p}'_{\mathrm{upwind}} = \begin{cases} p'_0 & \text{for } \dot{m}^*_f > 0 \\ p'_1 & \text{for } \dot{m}^*_f < 0 \end{cases} \tag{1639}$$

Since you can choose to neglect the second-order term $\rho'_f v_{fn}'$, these equations can be combined to define the mass flow correction as:

$$\dot{m}'_f = Q_f\left(p'_0 - p'_1\right) + \frac{\dot{m}_f^*}{\rho_f}\left(\frac{\partial \rho}{\partial p}\right)_T \mathrm{p}'_{\mathrm{upwind}} \tag{1640}$$

Discrete Pressure Correction Equation

The discrete pressure correction equation is obtained from Eqn. (1632) and Eqn. (1640) and is written in coefficient form as:

$$p'_p + \sum_n a_n p'_n = r \tag{1641}$$

The residual r is the net mass flow into the cell:

$$r = -\sum_n \dot{m}_f^* \tag{1642}$$

SIMPLE Algorithm

The SIMPLE algorithm is used to control the overall solution. This algorithm is summarized as follows:

1. Set the boundary conditions.
2. Computed the reconstruction gradients of velocity and pressure.
3. Compute the velocity and pressure gradients.
4. Solve the discretized momentum equation. This creates the intermediate velocity field $\mathbf{v}^*$.
5. Compute the uncorrected mass fluxes at faces $\dot{m}_f^*$.
6. Solve the pressure correction equation. This produces cell values for the pressure correction p'.
7. Update the pressure field:

$$p^{n+1} = p^n + \omega p' \tag{1643}$$

   where ω is the under-relaxation factor for pressure.
8. Update the boundary pressure corrections p'_b.
9. Correct the face mass fluxes:

$$\dot{m}_f^{n+1} = \dot{m}_f^* + \dot{m}'_f \tag{1644}$$

10. Correct the cell velocities:

$$\mathbf{v}_p^{n+1} = \mathbf{v}_p^* - \frac{V \nabla \mathrm{p}'}{\mathbf{a}'^{\mathrm{v}}_p} \tag{1645}$$

   where $\nabla p'$ is the gradient of the pressure corrections, $\mathbf{a}'^{\mathrm{v}}_p$ is the vector of central coefficients for the discretized velocity equation, and V is the cell volume.
11. Update density due to pressure changes.
12. Free all temporary storage.

PISO Algorithm

The PISO algorithm is suited for transient cases where the convective Courant number is small.

Comparing PISO with SIMPLE:

- PISO is faster than SIMPLE at short time-steps, though both algorithms have the same level of temporal accuracy. [276]
- PISO becomes unstable at long time-steps, when the combined CFL rises much above 10, while SIMPLE remains stable.
- As time-step size increases, SIMPLE loses temporal accuracy of transient solutions. However SIMPLE can still obtain accurate steady state solutions, if they exist, by using large time-step size.

Therefore, use SIMPLE for cases where large time-step sizes can be used, such as in achieving steady state solution, but use PISO where the problem demands use of small time-step size and greater temporal accuracy.

The PISO algorithm shares many features with the SIMPLE algorithm. The algorithm is summarized as follows:

1. Set the boundary conditions.
2. Compute the reconstruction gradients of velocity and pressure.
3. Compute the velocity and pressure gradients.
4. Solve the discretized momentum equation. This action creates the intermediate velocity field v^*.
5. Compute the uncorrected mass fluxes at faces $\dot{m}_f^*$.
6. Construct the pressure correction equation and test for convergence of the PISO algorithm. If PISO has converged, then set the last iteration flag to true.
7. Solve the pressure correction equation. This action produces cell values for the pressure correction p'.
8. Update the pressure field:

$$p^{n+1} = p^n + \omega p' \tag{1646}$$

   where ω is the under-relaxation factor for pressure. If this iteration is the last, ω is equal to one.
9. Update the boundary pressure corrections ${p'}_b$.
10. Correct the face mass fluxes:

$$\dot{m}_f^{n+1} = \dot{m}_f^* + {\dot{m}'}_f \tag{1647}$$

11. Correct the cell velocities:

$$v_p^{n+1} = v_p^* - \frac{V\omega\nabla p'}{a_p^v} \tag{1648}$$

   where $\nabla p'$ is the gradient of the pressure corrections, a_p^v is the vector of central coefficients for the discretized linear system representing the velocity equation, and V is the cell volume.
12. Update density due to pressure changes.
13. Solve any PISO-coupled equations such as for Species or Energy.
14. Recalculate density from the equation of state (except for the last corrector).
15. If this iteration is the last, then go to step 18.
16. Explicitly update the momentum equation with corrections due to the neighbor cells.
17. Go to step 5 and continue until convergence is reached.
18. Free all temporary storage.

Coupled Flow Solver

With the coupled solver, the conservation equations for continuity, momentum, energy, and species are solved in a coupled manner, that is, they are solved simultaneously as a vector of equations. The velocity field is obtained from the momentum equations. From the continuity equation, the pressure is calculated and the density is evaluated from the equation of state.

The coupled system of equations is solved by either the implicit or the explicit time-integration scheme.

Governing Equations in Vector Form

The system of governing equations in Cartesian integral form for an arbitrary control volume V with differential surface area $d\mathbf{a}$ can be written as:

$$\frac{\partial}{\partial t}\int_V \mathbf{W}dV + \oint \left[\mathbf{F} - \mathbf{G}\right] \cdot d\mathbf{a} = \int_V \mathbf{H}dV \tag{1649}$$

The vectors are defined as:

$$\mathbf{W} = \begin{bmatrix} \rho \\ \rho\mathbf{v} \\ \rho E \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho\mathbf{v} \\ \rho\mathbf{v}\mathbf{v} + p\mathbf{I} \\ \rho\mathbf{v}H + p\mathbf{v} \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} 0 \\ \mathbf{T} \\ \mathbf{T}\cdot\mathbf{v} + \dot{q}'' \end{bmatrix} \tag{1650}$$

where:

ρ	Density of the fluid
$\mathbf{v}$	Velocity of the fluid
E	Total energy per unit mass
p	Pressure
$\mathbf{T}$	Viscous stress tensor
$\dot{q}''$	Heat flux vector
$\mathbf{H}$	Vector of body forces

Total energy is related to the total enthalpy H by:

$$E = H - p/\rho \tag{1651}$$

where:

$$H = h + |\mathbf{v}|^2/2 \tag{1652}$$

and $h = C_p T$

where C_p is the specific heat capacity and T is the temperature.

For low Mach number and generally incompressible flows, the system of equations can become numerically stiff, which in turn can lead to convergence problems. To remedy this problem, preconditioning is introduced.

Preconditioned Equations

To provide efficient solution of both compressible and incompressible flows at all speeds, a preconditioning matrix Γ is incorporated into Eqn. (1649) by multiplication with the transient term [306]:

$$\Gamma \frac{\partial}{\partial t} \int_V \mathbf{Q} \, \mathrm{d}\, V + \oint \left(\mathbf{F} - \mathbf{G} \right) \cdot \, \mathrm{d}\, \mathbf{a} = \int_V \mathbf{H} \, \mathrm{d}\, V \tag{1653}$$

where:

$$\Gamma = \begin{bmatrix} \theta & 0 & \rho_T \\ \theta \mathbf{v} & \rho \mathbf{I} & \rho_T \mathbf{v} \\ \theta H - \delta & \rho \mathbf{v} & \rho_T H + \rho \, C_p \end{bmatrix} \tag{1654}$$

and:

$\rho_T = \frac{\partial \rho}{\partial T}\Big|_p$

ρ_T is the derivative of density with respect to temperature at constant pressure.

$$\text{For ideal gases: } \delta = 1 \;\; \rho_T = -\frac{p}{RT}$$
$$\text{For incompressible fluids: } \delta = 0 \qquad \rho_T = 0$$

For δ = 1, this matrix becomes a member of Turkel's family of preconditioners [304].

The parameter θ is defined as:

$$\theta = \frac{1}{U_r^2} - \frac{\rho_T}{\rho C_p} \tag{1655}$$

The reference velocity U_r appearing in Eqn. (1655) is chosen such that the eigenvalues of the system remain well-conditioned with respect to the convective and diffusive time scales [306]. This is accomplished by limiting U_r such that it does not go below either the local convection velocity or the diffusion velocity. An additional limitation on U_r considers local pressure differences and is designed to increase numerical stability in stagnation regions by prohibiting amplification of pressure perturbations there. Thus, the restriction on U_r becomes:

$$U_r = \min \left[\max \left(|\mathrm{v}|, \frac{\nu}{\Delta x}, \varepsilon \sqrt{\frac{\delta p}{\rho}}, U_{r,\min} \right), U_{r,\max} \right] \tag{1656}$$

where Δx is the inter-cell length scale over which the diffusion occurs and δp is the pressure difference between adjacent cells. For compressible flows, the maximum reference velocity is limited to the local speed of sound, c. In this case, the parameter $U_{r,\max}$ in Eqn. (1656) is replaced by c, and the value prescribed for $U_{r,\max}$ is not considered. The scaling parameter ε appearing in Eqn. (1656) is set to 2.

The non-preconditioned Navier-Stokes equations are recovered exactly from Eqn. (1649) by setting:

$$1/U_r^2 = \rho_p = \left.\frac{\partial \rho}{\partial p}\right|_T$$

In this case, Γ reduces to the Jacobian $\partial \mathbf{W} / \partial \mathbf{Q}$.

Although Eqn. (1653) is conservative for steady-state flows, it is not strictly conservative for time-dependent flows [304]. A dual time-stepping method is used to recover a time-accurate solution of the preconditioned equations for unsteady flows with implicit time stepping.

Applying Eqn. (1653) to a cell-centered control volume for cell-0, the following discretized system is obtained:

$$V_0 \Gamma_0 \frac{\partial \mathbf{Q}_0}{\partial t} + \sum_f \left(\mathbf{f}_f - \mathbf{g}_f\right) \cdot \mathbf{a} = \mathbf{h} V_0 \tag{1657}$$

where the summation is over the total number of faces defining cell-0. $\mathbf{f}_f$ is the inviscid (convective) flux and $\mathbf{g}_f$ is the viscous (diffusive) flux through face f. V_0 is the volume of cell-0 and $\mathbf{\Gamma}_0$ is the preconditioning matrix that is evaluated in cell-0.

Inviscid Fluxes

Based on upwind-differencing, there are two schemes available for discretizing the inviscid fluxes in Eqn. (1657)

Weiss-Smith Preconditioned Roe's Scheme

By default, the inviscid fluxes appearing in Eqn. (1657) are evaluated by using the Weiss-Smith preconditioned Roe's flux-difference splitting scheme as described in [305] and [306]. This scheme acknowledges that $\mathbf{f}_f$ contains characteristic information propagating through the domain with speed and direction according to the eigenvalues of the system. By splitting $\mathbf{f}_f$ into parts, where each part contains information traveling in a particular direction (that is, characteristic information), and upwind differencing the split fluxes in a manner consistent with their corresponding eigenvalues, the following expression for the value of the flux at each face is obtained:

$$\mathbf{f}_f = \frac{1}{2}\left(\mathbf{f}_0 + \mathbf{f}_1\right) - \frac{1}{2}\Gamma |\mathbf{A}| \, \Delta \, \mathbf{Q} \tag{1658}$$

where the subscripts "0" and "1" refer to the cells on either side of face-f, Γ is preconditioning matrix, and $\Delta\mathbf{Q}$ is:

$$\Delta \, \mathbf{Q} = \left(\mathbf{Q}_1^r - \mathbf{Q}_0^r\right) \tag{1659}$$

where $\mathbf{Q}_0^r$ and $\mathbf{Q}_1^r$ are the solution vectors from cell-0 and cell-1 interpolated to the face using reconstruction gradients.

The matrix $|\mathbf{A}|$ is defined as:

$$|\mathbf{A}| = \mathbf{M} \, |\Lambda| \, \mathbf{M}^{-1} \tag{1660}$$

where Λ is the diagonal matrix of eigenvalues and $\mathbf{M}$ is the modal matrix that diagonalizes $\Gamma^{-1}(\partial \mathbf{f} / \partial \mathbf{Q})$.

By using these reconstructed solution vectors, the discretization scheme becomes formally second-order accurate. In this form, Eqn. (1658) can be viewed as a second-order central-difference plus an added matrix dissipation, combining the pressure and velocity differential dissipations. This added matrix dissipation term is not only responsible for producing an upwinding of the convected variables, and of

pressure and flux velocity in supersonic flow. It also provides the pressure-velocity coupling that is required for stability and efficient convergence of low-speed and incompressible flows.

The mass-flux pressure dissipation term is modified to provide improved numerical dissipation for the entire range of Strouhal numbers, from Strouhal number zero (steady state) to high Strouhal numbers.

This modification affects the pressure differences term of the mass-flux dissipation. It is inspired by the method proposed by Hosangadi et al. [271], but with the following differences:

- It improves the accuracy of simulation for unsteady flows that require both sound and advected quantities (like total enthalpy and entropy).
- It allows you to specify a limit to the allowed amount of dissipation correction, for more robust simulation. The Unsteady Preconditioning Max Factor sets this limit. This factor has a range from 0 (no correction) to 1 (full correction).
- It bases the mass-flux dissipation corrections on local mesh and flow conditions.
- It does not require you to provide a reference length scale.

Liou's AUSM+ Scheme

Inviscid fluxes can be evaluated using Liou's AUSM+ flux-vector splitting scheme, as described in [255], [281] and other references. This scheme is also based on the upwind concept, which is applied consistently on the two physically distinct parts of the inviscid flux: convective and pressure. Thus, the numerical flux is written as:

$$\mathbf{f_f} = \mathbf{f}^{(c)} + \mathbf{P} = \mathbf{m_i}^{+}\left(1, u, v, ..., H\right)_0^T + \mathbf{m_i}^{-}\left(1, u, v, ..., H\right)_1^T + \mathbf{P}_i \tag{1661}$$

Here $\mathbf{m_i}$ is the mass flux, $\mathbf{m_i}^{+}$ is defined as $0.5\left(\mathbf{m_i} + \left|\mathbf{m_i}\right|\right)$, $\mathbf{m_i}^{-}$ is defined as $0.5\left(\mathbf{m_i} - \left|\mathbf{m_i}\right|\right)$, and $\mathbf{P}_i$ is the pressure flux. The mass flux and pressure flux are computed based on local flow characteristics, emphasizing on correct information propagation inside the fluid for convective and acoustic processes.

The mass-flux pressure dissipation term is modified to provide improved numerical dissipation for the entire range of Strouhal numbers, from Strouhal number zero (steady state) to high Strouhal numbers.

This modification follows the method proposed by Hosangadi et al. [271], but with the following differences:

- It bases the mass-flux dissipation corrections on local mesh and flow conditions.
- It does not require you to provide a reference length scale.

Steady Solution Algorithms

For steady-state simulations, the coupled system of equations Eqn. (1657) is discretized in time and time-stepping is performed until a quasi steady-state solution obtained. STAR-CCM+ provides an explicit and an implicit procedure to perform the time-integration.

Explicit Time-Stepping

An explicit multi-stage time-stepping scheme [277] is used to discretize the time-derivative in Eqn. (1653). The solution is advanced from time t to time $t + \Delta t$ with an m-stage Runge-Kutta scheme:

$$\begin{aligned} \mathbf{Q}^{(0)} &= \mathbf{Q}_t \\ \mathbf{Q}^{(i)} &= \mathbf{Q}^{(0)} - \alpha_i \Delta t \Gamma^{-1} \mathbf{R}^{(i-1)} \\ \mathbf{Q}_{t+\Delta t} &= \mathbf{Q}^{(m)} \end{aligned} \tag{1662}$$

where $i = 1, 2, ..., m$ is the stage counter for the m-stage scheme and α_i is the multi-stage coefficient for the ith stage. The residual $\mathbf{R}^{(i)}$ is computed from the intermediate solution $\mathbf{Q}^{(i)}$ and, for Eqn. (1657), is:

$$\mathbf{R}^{(i)} = \frac{1}{V}\sum_f \left\{\mathbf{f}\left(\mathbf{Q}^{(i)}\right) - \mathbf{g}\left(\mathbf{Q}^{(i)}\right)\right\} \tag{1663}$$

Residual Smoothing

Residual smoothing is a mechanism for increasing the explicit time-step size by removing the high-wave number oscillations from the residuals. The residuals for cell i are filtered through a Laplacian smoothing operator:

$$\bar{\mathbf{r}}_i = \mathbf{r}_i + \varepsilon \sum_{\text{neighbors}} \left(\bar{\mathbf{r}}_j - \bar{\mathbf{r}}_i\right) \tag{1664}$$

where the subscript j refers to the neighboring cells.

This equation is implemented using a Jacobi iteration as follows:

$$\bar{\mathbf{r}}_i^m = \frac{\mathrm{r}_i^0 + \varepsilon \sum\limits_{\text{neighbors}} \bar{\mathbf{r}}_j^{m-1}}{1 + \varepsilon \sum\limits_{\text{neighbors}} 1} \tag{1665}$$

where $\mathbf{r}_i^0$ represents the original (unsmoothed) residuals and $\bar{\mathbf{r}}_i^m$ represents the smoothed residuals after iteration m. The default number of smoothing operations is 2 with the under-relaxation factor ε = 0.5. This is typically sufficient to allow the Courant number to be doubled.

Implicit Time-Stepping

The coupled system of equations Eqn. (1649) is discretized in time by use of an Euler implicit scheme in conjunction with a Newton-type linearization of the fluxes .

$$\left[\mathbf{D}_i + \sum_j^{\mathrm{N_{faces}}} \mathrm{S}_{j,k}\right] \Delta\, \mathbf{Q} = -\mathbf{R}_i^n \tag{1666}$$

where $\Delta\, \mathbf{Q} \equiv \mathbf{Q}^{n+1} - \mathbf{Q}^n$. The center and off-diagonal coefficient matrices, D_i and $\mathrm{S}_{j,k}$, and the residual vector $\mathbf{r}_i^n$ are:

$$\mathrm{D}_i = \frac{V}{\Delta\, t}\Gamma + \sum_j^{\mathrm{N_{faces}}} \mathrm{S}_{j,k} \tag{1667}$$

$$\mathrm{S}_{j,k} = \frac{\partial \mathbf{F}_j}{\partial \mathbf{Q}_k} - \frac{\partial \mathbf{G}_j}{\partial \mathbf{Q}_k} \tag{1668}$$

$$\mathbf{R}_i^n = \sum_j^{\mathrm{Nfaces}} \left(\mathbf{F}_j - \mathbf{G}_j\right)^n \tag{1669}$$

Transient Solution Algorithms

For transient simulations, STAR-CCM+ provides explicit and implicit time-stepping schemes. The implicit time-stepping procedure is also known as dual-time formulation.

Explicit Time-Stepping

In STAR-CCM+, explicit time-stepping for transient simulations corresponds to the explicit time-stepping scheme that is described in Explicit Time-Stepping for steady-state simulations.

The physical time-step $\Delta\, t$ for unsteady simulations is applied uniformly to all cells in the domain.

The time-step $\Delta\, t$, appearing in Eqn. (1662) for explicit, unsteady calculations, is computed by consideration of the Courant number and Von Neumann stability conditions:

$$\Delta\, t = \min_{\forall x}\left(\frac{\mathrm{CFL}\, V(\mathbf{x})}{\lambda_{\max}(\mathbf{x})}, \frac{VNN\, \Delta\, x^2(\mathbf{x})}{\nu(\mathbf{x})}\right) \tag{1670}$$

The chosen time-step is the minimum local time-step over all the fluid cells in the model, and is equal to the minimum value of the pseudo-time step $\Delta\tau$ (Eqn. (1673)) in the domain.

Implicit Time-Stepping (Dual Time-Stepping)

Since time-derivative preconditioning destroys the time accuracy of the governing equations, solving for unsteady flows requires a dual time-stepping, with inner iterations in pseudo-time. A preconditioned pseudo-time derivative term is introduced into conservation Eqn. (1649) as follows:

$$\frac{\partial}{\partial t}\int_V \mathbf{W}\, \mathrm{d}\, V + \Gamma\frac{\partial}{\partial \tau}\int_V \mathbf{Q}\, \mathrm{d}\, V + \oint \left[\mathbf{F} - \mathbf{G}\right] \cdot d\mathbf{a} = \int_V \mathbf{H}\, \mathrm{d}\, V \tag{1671}$$

where t denotes the physical time-step which is applied uniformly to all cells in the domain. τ is a locally varying pseudo-time used in the time-marching procedure. As inner iterations progress to convergence, the second term of Eqn. (1671) vanishes and Eqn. (1649) is recovered.

The pseudo-time derivative is driven to zero with the following iterative algorithm:

$$\begin{gathered}\mathbf{Q}^{(0)} = \mathbf{Q}_n \\ \left[\Gamma + \frac{3}{2}\frac{\Delta\tau}{\Delta t}\frac{\partial \mathbf{W}}{\partial \mathbf{Q}}\right]\Delta\mathbf{Q} = \Delta\,\tau\left\{\mathbf{R}^{(i-1)} + \frac{1}{2\Delta t}\left(3\mathbf{W}^{(i-1)} - 4\mathbf{W}^{(n)} + \mathbf{W}^{(n-1)}\right)\right\} \\ \mathbf{Q}_{n+1} = \mathbf{Q}^{(m)}\end{gathered} \tag{1672}$$

where i is the inner iteration counter, n represents any given physical time level, and $\Delta\mathbf{Q} \equiv \mathbf{Q}^{(i)} - \mathbf{Q}^{(0)}$. Throughout the pseudo-time iterations, $\mathbf{W}^{(n)}$ and $\mathbf{W}^{(n-1)}$ are held constant and $\mathbf{W}^{(i-1)}$ is computed from $\mathbf{Q}^{(i-1)}$. When inner iterations converge, the solution at the next physical time level $\mathbf{W}^{(i+1)}$ is given by $\mathbf{W}\left(\mathbf{Q}_{\tau+\Delta\tau}\right)$. For each inner iteration, the linear system (block coupled for the explicit scheme or fully coupled for the implicit scheme) is solved to compute $\Delta\mathbf{Q}$.
The local pseudo time-step $\Delta\tau$ associated with the iterations of all steady simulations, or within a time-step of a transient simulation, is computed considering the Courant number and Von Neumann stability conditions.

$$\Delta\tau = \min\left(\frac{\mathrm{CFL}\, V(\mathbf{x})}{\lambda_{\max}(\mathbf{x})}, \frac{VNN\, \Delta\, x^2(\mathbf{x})}{\nu(\mathbf{x})}\right) \tag{1673}$$

where CFL is the user-specified Courant number (dimensionless), V is the cell volume (m^3), VNN is the Von Neumann number ($VNN \approx 1$), $\Delta\, x$ is a characteristic cell length scale (m) and ν is kinematic viscosity (m^2/s). $\lambda_{\max}$ is the maximum eigenvalue of the system as follows:

$$\lambda_{\max} = |\,\mathbf{u}\cdot\mathbf{a}\,| + c|\mathbf{a}| \tag{1674}$$

where $\mathbf{u}$ is velocity and c the speed of sound.

The maximum eigenvalue is determined from the pre-conditioned system, Eqn. (1653). The field function Convective Courant Number provides the local convective Courant number for a given time-step.

For implicit unsteady computations, Eqn. (1667), you specify the time-step.

Continuity Convergence Accelerator Algorithm

The Continuity Convergence Accelerator pressure-correction discretization uses the Coupled Flow numerical dissipation, not the Rhie-Chow dissipation of the Segregated solver. Choosing the Coupled Flow numerical dissipation formulation over the Rhie-Chow formulation prevents some of the known issues of the Segregated solver, especially in high-speed regimes, such as wiggles at flow discontinuities due to improper numerical dissipation.

When activated, the Continuity Convergence Accelerator algorithm adds a post-update conditioning step to help satisfy better the continuity equation.

The Coupled Implicit solver algorithm with Continuity Convergence Accelerator is:

1. As part of the normal Coupled Implicit iteration update, solve the discretized coupled flow equations, and perform the normal update to obtain an intermediate flow field state p^*, v^*...
2. Compute the new cell mass fluxes and mass imbalances for this intermediate state, m_f^*.
3. Discretize and solve the Continuity Convergence Accelerator pressure-correction equation with the new mass imbalances in the right-hand side:

$$a_p \mathrm{p}'_p + \sum_n a_n \mathrm{p}'_n = -\sum_n m_f^* \tag{1675}$$

   to produce cell values of the pressure correction p'. The subscript n denotes all neighbors of cell p with faces f.
4. Update cell pressures using the under-relaxation factor ω:

$$p^{n+1} = p^* + \omega p' \tag{1676}$$

5. Update the boundary pressure corrections p'_b, and then pressure p_b.
6. Correct the face mass fluxes:

$$m_f^{n+1} = m_f^* + \mathrm{m}' \tag{1677}$$

   where m' is computed using Eqn. (1640).

   If the *Enhanced Mass-Imbalance Calculations* expert property is activated, the mass imbalances are recomputed using either Eqn. (1658) or Eqn. (1661).
7. Correct the cell velocities

$$v^{n+1} = v^* - f(\omega)\frac{V\,\Delta\,\mathrm{p}'}{A_p v} \tag{1678}$$

   where:

   - $f(\omega)$ is a non-linear function of ω
   - $\Delta\,\mathrm{p}'$ is the gradient of the pressure corrections

- A_p is the vector of central coefficients for the original coupled equations discretized system representing the velocity equations
- V is the cell volume

8. Update density, total enthalpy, and other variables with the pressure and velocity changes.

Algebraic Multigrid

Conventional iterative solution algorithms such as Jacobi, Gauss-Seidel, or ILU (Incomplete Lower-Upper) converge significantly slower with increasing mesh sizes. This slow convergence in turn leads to a quadratic increase in computational time. To accelerate solver convergence, STAR-CCM+ employs the Algebraic Multigrid (AMG) method. The concept of multigrid methods is based on the fact that an iterative solution algorithm reduces efficiently the numerical error components whose wave lengths correspond to the cell size (high-frequency errors). The long-wavelength (low-frequency) errors, however, are reduced rather slowly with such a method.

Multigrid methods reduce the low-frequency errors through an iterative process on a hierarchy of successively coarsened linear systems. Algebraic multigrid derives a coarse level system without reference to the underlying grid geometry. The coarse-grid equations are derived from arithmetic combinations of the fine-grid coefficients. After several iterations, a multigrid algorithm transfers the computation from a fine linear system to a coarse linear system. These iterations are also called smoothing iterations, because the error function is then smooth, that is, free of high-frequency components of the error. Due to the transfer of the solution process to a coarser linear system, the errors are now higher-frequency with respect to the cell size and can be reduced efficiently. To reduce the error of the fine linear system solution on a coarser linear system, a defect equation is defined.

Multigrid algorithms apply the following steps:

- Agglomerate cells to form coarse grid levels.
- Transfer the residual from a fine level to a a coarser level (known as restriction).
- Transfer the correction from a coarse level back to a finer level (known as prolongation).

STAR-CCM+ provides two methods for accelerating the linear solver, the preconditioned conjugate-gradient method for incompressible flows and the bi-conjugate gradient stabilized method for all others.

Multigrid Cycling

Multigrid techniques enable a basic iteration scheme such as Gauss-Seidel to be greatly accelerated by using simple correction sweeps on a sequence of coarse meshes. The strategy for visiting the coarse levels can have a significant impact on the efficiency of the algorithm. Two cycling strategies are available in the AMG solver that is used in STAR-CCM+—fixed and flexible:

- Fixed cycles—A complete multigrid cycle represents the recursive application of a single cycle that is composed of the following steps:
  1. (Pre)smooth
  2. Restrict
  3. Cycle anew
  4. Prolongate

5. (Post)smooth

These steps are applied to a sequence of successively coarser grids or equation sets. Smooth means to apply any number of iterative relaxation sweeps to the equations on the current fine level to compute a new set of corrections. Restrict is to transfer the existing residuals down to the next coarsest level where a new cycle is applied. Following that the resultant corrections are prolongated—transferred back up to the current fine level—where smoothing is again applied.

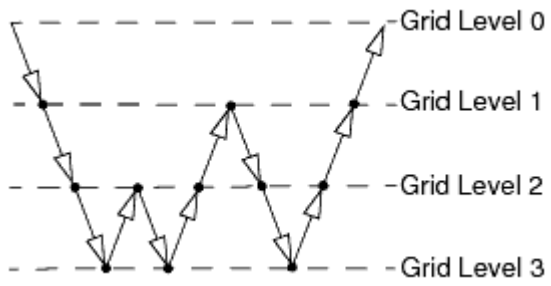
The three types of fixed cycles are:

- F fixed cycles
- V fixed cycles
- W fixed cycles

- Flexible cycles—This type of cycle is a more economical cycling strategy for linear systems that are not stiff. Rather than using all multigrid levels in a regular pattern, the residuals are monitored after every sweep on a given grid level. If the residual rate of reduction exceeds a given threshold, the solution continues on a coarser level. If the residual on a given level is reduced more than a specified tolerance, the solution moves to a finer level. A limit is further imposed on the number of sweeps that are allowed at any level.

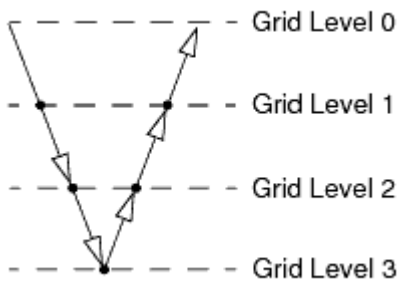
F Cycle

The F cycle is a variant of the W cycle. Illustrated below, this cycle involves fewer coarse-level sweeps than the W cycle, but still more than the V cycle.



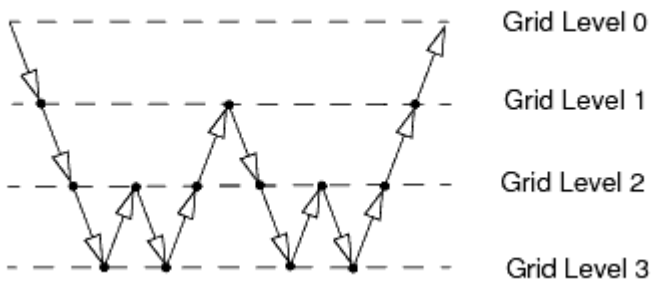
V Cycle

The V cycle is the simplest type of fixed cycle, and only has two legs. In the first leg, one performs a number of relaxation sweeps on the finest level and transfers the residuals to the next level. Then, the operation is repeated on successively coarse levels until the coarsest level is reached. A coarse “grid” generally contains only a few “cells”. After finishing the sweeps on the coarsest level, the solution is used to correct the solution on the next finer level. Some relaxation sweeps are performed on that level before repeating the process until the finest level is reached. This procedure is illustrated below.



W Cycle

For stiff systems, the V cycle sometimes is not sufficient, and more coarse iterations are advantageous. The W cycle increases the number of coarse relaxation sweeps as illustrated below.



AMG Solver

The result of the discretization approach that is described above is a linear system:

$$\mathbf{A}\,\mathbf{x} = \mathbf{b} \tag{1679}$$

representing the algebraic equations for each computational cell. The matrix $\mathbf{A}$ represents the coefficients of the linear system (for example, the coefficients a_p and a_n on the left-hand side of Eqn. (1625), the vector $\mathbf{x}$ represents the unknowns ($\Delta\,\phi$ in Eqn. (1625)) in each cell, and the vector $\mathbf{b}$ represents the residuals (see Eqn. (1626)) from each cell.

The resultant linear system of equations is solved using either the Jacobi, the Gauss-Seidel, or the Incomplete Lower Upper decomposition relaxation scheme (smoother).

Jacobi and Gauss-Seidel

The general principle behind iterative methods is that, given an approximate solution $\mathbf{x}^k$, a better approximation $\mathbf{x}^{k+1}$ is sought, and then the process is repeated.

If the error at iteration k is defined as:

$$\mathbf{e}^k = \mathbf{x} - \mathbf{x}^k \tag{1680}$$

where $\mathbf{x}$ represents the exact solution and the residual as:

$$\mathbf{r}^k = \mathbf{b} - \mathbf{A}\mathbf{x}^k \tag{1681}$$

it follows that:

$$\mathbf{A}\mathbf{e}^k = \mathbf{r}^k \tag{1682}$$

Therefore, continuing the iteration until the residual is driven to a small value, drives the error a small value.

The most basic iterative methods are Jacobi and Gauss-Seidel iteration. These methods involve visiting each cell in sequence, and updating the value of x_i in each cell i using the coefficients of its n neighbor cells as follows:

$$x_i = \frac{1}{A_{i,i}}\left(b - \sum\nolimits_{\text{neighbors } n} A_{i,n} x_n\right) \tag{1683}$$

The difference between Jacobi and Gauss-Seidel iteration is subtle: Jacobi uses the “old” values of x_n, while Gauss-Seidel uses the available values that have been updated, resulting in better convergence.

Incomplete Lower Upper (ILU)

The ILU (Incomplete Lower-Upper) method, Saad [298], solves iteratively for $\mathbf{x}^{k+1}$ using:

$$\widetilde{\mathbf{A}}\left(\mathbf{x}^{k+1} - \mathbf{x}^k\right) = b - \mathbf{A}\mathbf{x}^k \tag{1684}$$

where

$$\widetilde{\mathbf{A}} = \mathbf{LU} = \left(\mathbf{D} + \mathbf{L}_A\right)\left(\mathbf{I} + \mathbf{D}^{-1}\mathbf{U}_A\right) \tag{1685}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{L}_A$ and $\mathbf{U}_A$ are the lower and upper triangular matrices of the original matrix $\mathbf{A}$, and $\mathbf{D}$ is a diagonal matrix that is calculated using the MILU(0) method.

The ILU method is more computationally expensive but more robust than the Gauss-Seidel method. The robustness of the ILU method means that it can handle group sizes of 8 or more in three dimensions; the Gauss-Seidel method can only handle group sizes of 4. Use the ILU method for complex problems, and to improve convergence and parallel scalability. For three-dimensional problems, use it with a group size of 8.

Acceleration Methods

Preconditioned Conjugate-Gradient Method

To accelerate the convergence of the iterative solution of the linear system in the Pressure solver, a conjugate-gradient method is available that uses the AMG solver as the preconditioner. Use this method for incompressible flows (Constant Density model) with the Segregated Flow model.

To solve the linear system $\mathbf{A}\,\mathbf{x} = \mathbf{b}$, the preconditioned conjugate-gradient (PCG) takes the following form:

$$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0 \tag{1686}$$

$$\mathbf{z}_0 \cong \text{AMG fixed cycle}\left(\mathbf{A}, \mathbf{r}_0\right) \tag{1687}$$

$$\mathbf{p}_0 = \mathbf{z}_0 \tag{1688}$$

$$\mathbf{k} = 0 \tag{1689}$$

Each iteration involves the calculations in Eqn. (1690) through Eqn. (1696):

$$\alpha_k = \frac{\left(\mathbf{r}_k^T \mathbf{z}_k\right)}{\left(\mathbf{p}_k^T \mathrm{A} \mathbf{p}_k\right)} \tag{1690}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \tag{1691}$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{A} \mathbf{p}_k \tag{1692}$$

If $\left|\mathbf{r}_{k+1}\right| < \varepsilon$, then the iteration ends.

$$\mathbf{z}_{k+1} \cong \text{AMG fixed cycle } \left(\mathbf{A}, \mathbf{r}_{k+1}\right) \tag{1693}$$

$$\beta_k = \frac{\left(\mathbf{r}_k^+ \mathbf{z}_{k+1}\right)}{\left(\mathbf{r}_k \mathbf{z}_k\right)} \tag{1694}$$

$$\mathbf{p}_{k+1} = \mathbf{z}_{k+1} + \beta_k \mathbf{r}_k \tag{1695}$$

$$k = k + 1 \tag{1696}$$

The solution of the linear system is $\mathbf{x}_{k+1}$.

A necessary condition for the convergence of the PCG algorithm is that the matrix $\mathbf{A}$ (which here is also the preconditioner) is symmetric positive-definite (SPD). The matrix of the linear system of the Pressure solver in pure incompressible cases satisfies this condition.

Eqn. (1687) and Eqn. (1693) in the iteration process are preconditioner solver calculations, which involve one AMG fixed-cycle loop to approximate the auxiliary linear systems:

$$\mathbf{A} \mathbf{z}_k = \mathbf{r}_k \qquad \mathrm{k} = 0, 1, \ldots \tag{1697}$$

As implemented, the PCG accelerates convergence and improves the parallel scalability of the Pressure solver. For normal cases (no non-linear instability), it is expected that the PCG does not change significantly the overall non-linear convergence of STAR-CCM+. The PCG can provide robust convergence for those cases where numerical instability results from a lack of incomplete convergence of the linear system of the Pressure solver.

Bi-Conjugate Gradient Stabilized Method

The Bi-Conjugate Gradient Stabilized (BiCGStab) method has been developed to solve nonsymmetric (general) linear systems. STAR-CCM+ implements an AMG-preconditioned BiCGStab algorithm to improve the robustness and the speed of convergence of linear systems that the solvers produce.

To solve the general linear system $\mathbf{A}\,\mathbf{x} = \mathbf{b}$, the preconditioned BiCGStab takes the following steps:

$$\mathbf{r}_0 = \mathbf{b} - \mathbf{A} \mathbf{x}_0 \tag{1698}$$

$$\widetilde{\mathrm{r}} = \mathbf{r}_0 \tag{1699}$$

$$i = 0 \tag{1700}$$

Each iteration i involves calculations of Eqn. (1701) through Eqn. (1714):

$$\rho_{i-1} = \widetilde{\mathrm{r}}^T \mathbf{r}_{i-1} \tag{1701}$$

If $\rho_{i-1} = 0$, the method fails.

If $i = 1$:

$$\mathbf{p}_i = \mathbf{r}_{i-1} \tag{1702}$$

Otherwise, as per Eqn. (1703) and Eqn. (1704):

$$\beta_{i-1} = \left(\frac{\rho_{i-1}}{\rho_{i-2}}\right)\left(\frac{\alpha_{i-1}}{\omega_{i-1}}\right) \tag{1703}$$

$$\mathbf{p}_i = \mathbf{r}_{i-1} + \beta_{i-1}\left(\mathbf{p}_{i-1} - \omega_{i-1}\mathbf{v}_{i-1}\right) \tag{1704}$$

$$\hat{\mathrm{p}} = \text{AMG fixed cycle}\left(\mathbf{A}, \mathbf{p}_i\right) \tag{1705}$$

$$\mathbf{v}_i = \mathbf{A} \cdot \hat{\mathrm{p}} \tag{1706}$$

$$\alpha_i = \frac{\rho_{i-1}}{\left(\tilde{\mathrm{r}}^T \cdot \mathbf{v}_i\right)} \tag{1707}$$

$$\mathbf{S} = \mathbf{r}_{i-1} - \left(\alpha_i \cdot \mathbf{v}_i\right) \tag{1708}$$

If $(|\mathbf{S}| < \varepsilon)$, then as per Eqn. (1709):

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha_i \hat{\mathrm{p}} \tag{1709}$$

and the iteration ends.

Otherwise the iteration continues as follows:

$$\hat{\mathrm{S}} = \text{AMG fixed cycle}\left(\mathbf{A}, \mathbf{S}\right) \tag{1710}$$

$$\mathbf{t} = \mathrm{A} \cdot \hat{\mathrm{S}} \tag{1711}$$

$$\omega_i = \frac{\mathbf{t}^T \cdot \mathbf{S}}{\mathbf{t}^T \cdot \mathbf{t}} \tag{1712}$$

$$\mathrm{x}_i = \mathbf{x}_{i-1} + \alpha_i \hat{\mathrm{p}} + \omega_i \mathbf{S} \tag{1713}$$

$$\mathbf{r}_i = \mathbf{S} - \omega_i \mathbf{t} \tag{1714}$$

If $\left|\mathbf{r}_i\right| < \varepsilon$, then the iteration ends.

If $\omega_i \neq 0$, then the iteration continues.

The solution of the linear system is $\mathbf{x}_i$.

Eqn. (1705) and Eqn. (1710) in the iteration process are the preconditioner solver calculations, which involve one AMG fixed-cycle loop to obtain an approximate solution of:

$$\mathbf{A} \cdot \hat{\mathrm{p}} = \mathbf{p}_i \quad \text{and} \quad \mathrm{A} \cdot \hat{\mathrm{S}} = \mathrm{S} \quad \text{respectively} \tag{1715}$$

As with the Preconditioned Conjugate-Gradient method, BiCGStab can improve parallel scalability and improve robustness for those cases where numerical instability was due to lack of sufficient convergence of the linear system.

Finite Element Method for Viscoelastic Fluid Flow

In STAR-CCM+, the numerical algorithm for capturing the complex flow field of non-Newtonian viscoelastic fluids is based on the finite element method. The discretization of a closed-form non-linear differential constitutive equation is described exemplarily by means of the Oldroyd-B model and can be readily extended to other differential constitutive equations.

The governing set of equations consisting of continuity equation, momentum equations, and constitutive equation is non-dimensionalized. The coupled system of equations is discretized by using a particular mixed finite element method, which employs three stabilization techniques. These stabilization techniques address issues with the stress-velocity coupling, the non-linear convection terms, and the incompressibility constraint of the fluid.

In steady-state and under isothermal conditions, the governing equations for an incompressible viscoelastic fluid flow can be written in non-dimensional form:

Continuity equation

$$\nabla \cdot \mathbf{v} = 0 \tag{1716}$$

Momentum equation

$$\mathrm{Re}(\mathbf{v} \cdot \nabla \mathbf{v}) = -\nabla p + \nabla \cdot \mathbf{T} \tag{1717}$$

Constitutive Equation

To describe the numerical procedure, the viscoelastic constitutive equation is represented by the Oldroyd B model:

$$\mathrm{Wi}\overset{\triangledown}{\mathbf{T}}_i^p + \mathbf{T}_i^p = 2\zeta_i \mathbf{D} \tag{1718}$$

$$\mathbf{T} = 2\zeta_s \mathbf{D} + \sum_{i=1}^{N} \mathbf{T}_i^p \tag{1719}$$

where:

$\mathbf{v}$	non-dimensional velocity
p	non-dimensional pressure
$\mathbf{T}$	non-dimensional viscous stress tensor
$\mathrm{Re} = \frac{\rho v_c l_c}{\mu_c}$	Reynolds number
v_c	characteristic velocity
l_c	characteristic length
μ_c	characteristic dynamic viscosity

$\mathrm{Wi}_i = \frac{\lambda_i v_c}{l_c}$	Weissenberg number that is associated with the ith mode
λ_i	relaxation time of the ith mode
$\zeta_i = \mu_i / \mu_c$	dimensionless viscosity
μ_i	viscosity that is associated with the ith mode
μ_s	solvent viscosity

The above equations are non-dimensionalized by $v \sim v_c, l \sim l_c, p, \mathbf{T} \sim \mu_c v_c / l_c$.

In this form, the above system of equations yields a double saddle-point problem, because the velocity and pressure do not appear in the momentum and continuity equations explicitly. Although the velocity appears explicitly only in the momentum equation through the solvent viscosity term, the contribution of this term is often negligible for viscoelastic fluids. To overcome this difficulty, the Discrete-Elastic-Viscous-Split-Stress (DEVSS) method proposed by Guénette and Fortin [767] is used. The method defines an additional tensorial unknown $\mathbf{d}$:

$$\mathbf{T} \rightarrow 2\zeta_s \mathbf{D}(\mathbf{v}) + \sum_{i=1}^{N} \mathbf{T}_i^p - 2\alpha \mathbf{d} + 2\alpha \mathbf{D}(\mathbf{v}) \tag{1720}$$

where α is an arbitrary constant.

This substitution results in the following $(\mathbf{v}, p, \mathbf{T}_i^p, \mathbf{d})$ formulation:

$$\nabla \cdot \mathbf{v} = 0 \tag{1721}$$

$$\mathrm{Re}(\mathbf{v} \cdot \nabla \mathbf{v}) + \nabla p - (2\zeta_s + 2\alpha) \nabla \cdot \mathbf{D}(\mathbf{v}) - \sum_{i=1}^{N} \nabla \cdot \mathbf{T}_i^p + 2\alpha \nabla \cdot \mathbf{d} = 0 \tag{1722}$$

$$\mathrm{Wi}_i \mathbf{T}_i^p + \mathbf{T}_i^p - 2\xi_i \mathbf{D}(\mathbf{v}) = 0 \tag{1723}$$

$$\mathbf{D}(\mathbf{v}) - \mathbf{d} = 0 \tag{1724}$$

Variational Formulation

To obtain the variational formulation, the above set of equations is multiplied with the weight functions $(\widetilde{\mathrm{p}}_h, \widetilde{\mathbf{v}}_h, \widetilde{\mathbf{T}}_h, \widetilde{\mathbf{d}}_h)$ and then integrated over the volume domain Ω with the enclosing boundary Γ and normal $\mathbf{n}_\Gamma$.

By using the standard Galerkin approach for the system of $(p_h, \mathbf{v}_h, \mathbf{T}_{i,h}^p, \mathbf{d}_h)$ and applying the divergence theorem, the system of equations is given by:

$$\int_\Omega \nabla \cdot \mathbf{v}_h \quad \widetilde{p}_h \quad d\Omega = 0, \qquad \forall \widetilde{p}_h \in H_0^1(\Omega) \tag{1725}$$

$$\mathrm{Re}\int_{\Omega}(\mathbf{v}_h\cdot\nabla\mathbf{v}_h)\cdot\widetilde{\mathbf{v}_h}\quad d\Omega-\int_{\Omega}p_h\nabla\cdot\widetilde{\mathbf{v}}_h\quad d\Omega+(2\zeta_s+2\alpha)\int_{\Omega}\mathbf{D}(\mathbf{v}_h):\mathbf{D}(\widetilde{\mathbf{v}}_h)\quad d\Omega \tag{1726}$$
$$+\sum_{m=1}^{N}\int_{\Omega}\mathbf{T}_i^p:\mathbf{D}(\widetilde{\mathbf{v}}_h)\quad d\Omega-2\alpha\int_{\Omega}\mathbf{d}:\mathbf{D}(\widetilde{\mathbf{v}}_h)\quad d\Omega-\int_{\Gamma}\mathbf{n}_{\Gamma}$$
$$\cdot(-p_h\mathbf{I}+2\zeta_s\mathbf{D}(\mathbf{v}_h)+\sum_{m=1}^{N}\mathbf{T}_i^p+2\alpha(\mathbf{D}(\mathbf{v}_h)-\mathbf{d}))\cdot\widetilde{\mathbf{v}}_h\quad d\Gamma=0,\qquad\forall\widetilde{\mathbf{v}}_h\in(H_0^1(\Omega))^3,$$

$$\mathrm{We}_i\int_{\Omega}\overset{\nabla}{\mathbf{T}}{}_{i,h}^p:\varphi_h\quad d\Omega+\int_{\Omega}\mathbf{T}_{i,h}^p:\varphi_h\quad d\Omega-2\zeta^m\int_{\Omega}\mathbf{D}(\mathbf{v}_h):\varphi_h\quad d\Omega=0,\qquad\forall\varphi_h\in(H_0^1(\Omega))^6 \tag{1727}$$

$$\int_{\Omega}\mathbf{D}(\mathbf{v}_h):\Psi_h\quad d\Omega-\int_{\Omega}\mathbf{d}_h:\Psi_h\quad d\Omega=0,\qquad\forall\Psi_h\in(H_0^1(\Omega))^6 \tag{1728}$$

where $H_0^k(\Omega)$ denotes the Sobolev space consisting of square integrable functions for which all derivatives up to order k are square integrable, with the magnitude zero at the boundaries with essential boundary conditions and $\mathbf{n}_{\Gamma}$ is the unit normal vector pointing outward to the boundary Γ enclosing the domain Ω.

The continuous space domain is discretized into a finite number of elements, which are interconnected at the vertices.

Passive Scalars in Viscoelastic Fluid Flow

Passive scalars are treated as components of the viscous fluid that can be advected by the flow field and diffused according to Fick's law, but occur in concentrations so small that their presence does not alter the flow field. With multiple passive scalars, there is no interaction between the components. The distribution of each passive scalar in the system is described by an advection-diffusion equation:

$$Pe_i\left(\frac{\partial C_i}{\partial t}+\mathbf{v}\cdot\nabla C_i\right)=\nabla^2 C_i \tag{1729}$$

where:

- C_i is the non-dimensional distribution of the ith species in the system.
- Pe_i is the Péclet number, $Pe_i=v_c l_c/D_i$.
- D_i is the molecular diffusivity of the ith species that (for small molecule species) is obtained by the Stokes-Einstein relation.
- t is non-dimensional physical time scaled with the advection time scale $t_c\sim a/v_c$.
- $\mathbf{v}$ is the non-dimensional velocity vector.

This equation is one-way coupled to the hydrodynamic equations.

To obtain a variational formulation for passive scalars, multiply Eqn. (1729) by the weight function $\widetilde{C}_i$ and then integrate over the volume domain Ω. This gives:

$$\int_{\Omega}Pe_i\widetilde{C}_i\frac{\partial C_i}{\partial t}d\Omega+\int_{\Omega}Pe_i\widetilde{C}_i(\mathbf{v}\cdot\nabla C_i)d\Omega-\int_{\Omega}\widetilde{C}_i\nabla^2 C_i d\Omega=0,\quad\forall\widetilde{C}_i\in H_0^1(\Omega) \tag{1730}$$

Integrate the last term by parts and use the divergence theorem to derive:

$$\int_{\Omega} Pe_i \widetilde{C}_i \frac{\partial C_i}{\partial t} d\Omega + \int_{\Omega} Pe_i \widetilde{C}_i (\mathbf{v} \cdot \nabla C_i) d\Omega + \int_{\Omega} \nabla \widetilde{C}_i \cdot \nabla C_i d\Omega - \int_{\Gamma} \widetilde{C}_i (\mathbf{n}_{\Gamma} \cdot \nabla C_i) d\Omega = 0, \quad \forall \widetilde{C}_i \in H_0^1(\Omega) \tag{1731}$$

The passive scalar equation does not modify the flow field, but it cannot be solved independently, since a velocity field must be known in order to solve the above discretized equation simultaneously with the discretized hydrodynamic equations.

Streamline Upwind Petrov-Galerkin

STAR-CCM+ uses the standard streamline upwind Petrov-Galerkin method (SUPG), which is a stabilization technique to prevent oscillations in convection-dominated flow problems in the finite element method.

One difficulty in numerically solving viscoelastic fluid flow is the existence of non-linear and non-Hermitian convective terms in the momentum and constitutive equations. The convective terms that are associated with the momentum equation are usually small in polymeric flows, because the characteristic viscosity μ_c is high and therefore, the Reynolds number Re is small. However, the effect of the non-linear term in the constitutive equation is dominant at high Weissenberg numbers Wi.

There is also a convective term in the passive scalar equation, so the equation must be stabilized for high values of Pe_i.

The general form of this consistent stabilization technique relies on adding of quadratures for each element in the following form to the weak form:

$$\beta_{\mathrm{SUPG}} \int_{\Omega^e} \tau_{\mathrm{SUPG}} \quad \mathcal{P} \cdot \mathcal{R} \quad d\Omega \tag{1732}$$

where β_{SUPG} is a parameter (but usually equal to one for convenience), $\mathcal{P}$ is a specific operator applied to the weight function, $\mathcal{R}$ is the residual of the partial differential equation and τ_{SUPG} is the SUPG stabilization parameter (also called intrinsic time). In the SUPG stabilization technique, these functions for the momentum equation are defined as [780] [781] [782]:

$$\tau_{\mathrm{SUPG}} = \left(\left(\frac{2||\mathbf{v}_h||}{h} \right)^2 + \left(\frac{4\mu_s}{\rho h^2} \right)^2 \right)^{-\frac{1}{2}} \tag{1733}$$

$$\mathcal{P} = \mathbf{v}_h \cdot \nabla \mathbf{u}_h \tag{1734}$$

$$\mathcal{R} = \mathrm{Re}(\mathbf{v}_h \cdot \nabla \mathbf{v}_h) + \nabla p_h - \sum_{m=1}^{N} \nabla \cdot \mathbf{T}_{i,h}^p - 2\zeta_s \mathbf{D}(\mathbf{v}_h) \tag{1735}$$

For the constitutive equation, they are given by:

$$\tau_{\mathrm{SUPG}} = \frac{h}{2||\mathbf{v}_h||} \tag{1736}$$

$$\mathcal{P} = \mathbf{v}_h \cdot \nabla \boldsymbol{\varphi}_h \tag{1737}$$

$$\mathcal{R} = \mathrm{Wi}_i \overset{\nabla}{\mathbf{T}}_i^p + \mathbf{T}_i^p - 2\zeta_i \mathbf{D}_{\mathbf{v}} \tag{1738}$$

where h denotes the characteristic size of the element.

For passive scalar fluid components, τ_{SUPG}, $\mathcal{P}$, and $\mathcal{R}$ are given by:

$$\tau_{\mathrm{SUPG}} = \frac{h}{2\left|\left|\mathbf{v}_h\right|\right|} \tag{1739}$$

$$\mathcal{P} = \mathbf{v}_h \cdot \nabla \widetilde{C}_i \tag{1740}$$

$$\mathcal{R} = Pe_i\left(\frac{\partial C_i}{\partial t} + \mathbf{v} \cdot \nabla C_i\right) - \nabla^2 C_i \tag{1741}$$

Pressure Stabilized Petrov-Galerkin

The algebraic system of equations for the nodal values of unknown pressure and velocity vectors in the Galerkin formulation is governed by a partitioned matrix with a zero submatrix on the diagonal. The stability and convergence of this algebraic system relies on satisfying the Ladyzenskaja-Babuska-Brezzi (LBB) compatibility condition [771]. STAR-CCM+ employs the pressure stabilized Petrov-Galerkin (PSPG) method [768] [769] [781] that circumvents this requirement.

PSPG is a numerical technique to stabilize the formulation of the Galerkin finite element of an incompressible fluid. Similar to SUPG, the method adds quadratures, which are associated with each element, into the continuity equation:

$$\beta_{\mathrm{PSPG}} \int_{\Omega^e} \tau_{\mathrm{PSPG}} \left(\nabla q_h \cdot \mathcal{R}\right) d\Omega \tag{1742}$$

With:

$$\tau_{\mathrm{PSPG}} = \frac{\rho h^2}{12\mu_s} \tag{1743}$$

where $\mathcal{R}$ is defined above and β_{PSPG} is the Petrov-Galerkin parameter, used to control numerical stabilization of incompressibility. The default value is 1.0. The PSPG formulation permits the use of mixed elements, e.g. Q1-Q1 and Q2-Q2, with equal-order interpolations for the velocity and the pressure.

For a general treatment of finite element method, see Finite Element Method.

Numerical Flow Solution Bibliography

[245] *Baaijens, F. P. T. 1998. "Mixed finite element methods for viscoelastic flow analysis: a review, J. Non-Newton Fluid, 79, pp. 361-385.*

[246] *Babuska, I. 1973. "The finite element method with Lagrangian multipliers", J. Rheol., 20, pp. 179-192.*

[247] *Babuska, I. 1971. " Error-bounds for finite element method" , Numerische Mathematik, 16, pp. 322-333.*

[248] *Bird, R.B., Armstrong, R.C, Hassager, Ole. 1987. Dynamics of Polymeric Liquids, I-II. John Wiley & Sons, New York.*

[249] *Bird, R.B., Stewart, W.E., and Lightfoot, E.N. 1966. “Transport Phenomena”. John Wiley & Sons, New York.*

[250] *Brezzi, F. 1974. " On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers", ESAIM: Mathematical Modelling and Numerical Analysis, 8, pp. 129-151.*

[251] *Brooks, A. N. and Hughes, T. J. R. 1982. " Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations" , Comput. Methods in Appl. Mech. Eng., 32, pp. 199-259.*

[252] *Burton,T., Jenkins, N., Sharpe, D., Bossanyi, E. 2011. “Wind Energy Handbook”, 2nd ed., Wiley.*

[253] *Bush, M. B., Tanner, R. I. and Phan-Thien, N.1985. " A boundary element investigation of extrudate swell", J. Non-Newton. Fluid, 18, pp. 143-162.*

[254] *Bush, M. B. and Tanner, R. I. 1983. " Numerical solution of viscous flows using integral equation methods", Int. J. Numer. Meth. Fl., 3, pp. 71-92.*

[255] *Chima, R.V., and Liou, M.-S. 2003. “Comparison of the AUSM+ and H-CUSP Schemes for Turbomachinery Applications”, NASA/TM-2003-212457.*

[256] *Crochet, M. J. and Walters, K. and Davies, A. R., 1984. “Numerical simulation of non-Newtonian flow”, Elsevier Amsterdam ; New York.*

[257] *Darwish M.S. and Moukalled F. 1994. “Normalized Variable and Space Formulation Methodology for High-Resolution Schemes”, Num. Heat Trans., Part B, 26 pp. 79-107.*

[258] *Demirdzic, I., Lilek, Z., and Peric, M. 1993. “A collocated finite volume method for predicting flows at all speeds”, Int. J. for Numerical Methods in Fluids, 16, pp. 1029-1050.*

[259] *Demirdzic, I. and Muzaferija, S. 1995. “Numerical method for coupled fluid flow, heat transfer and stress analysis using unstructured moving meshes with cells of arbitrary topology”, Comput. Methods Appl. Mech. Eng., pp. 1-21.*

[260] *Ferziger, J.H. and Peric, M. 2002. “Computational Methods for Fluid Dynamics”, 3rd rev. ed., Springer-Verlag, Berlin.*

[261] *Giles, M.B. 1988. “Non-reflecting boundary conditions for the Euler equations”, CFDL Report 88-1, MIT Dept. of Aero. and Astro.*

[262] *Giles, M.B. 1990. “Non-reflecting boundary conditions for Euler equation calculations”, AIAA Journal, 28(12):2050-2058.*

[263] *Giles, M.B. 1991 “UNSFLO: A numerical method for unsteady flow in turbomachinery”, Gas Turbine Laboratory Report GTL 205, MIT Dept. of Aero. and Astro.*

[264] *Giraud, L., d'Humières, D. and Lallemand, P. 1998. “A lattice Boltzmann model for Jeffreys viscoelastic fluid”, Eur. Phys. Lett., 42, pp. 625*

[265] *Gnoffo, P. A., Gupta, R. N., and Shinn, J. L. 1989. “Conservation Equations and Physical Models for Hypersonic Air Flows in Thermal and Chemical Nonequilibrium”, NASA TP-2867.*

[266] *Graboski, M. S. and Daubert, T. E. 1978. “A Modified Soave Equation of State for Phase Equilibrium Calculations: Hydrocarbon Systems”, Ind. Eng. Chem. Process Des. Dev., 17(4), pp. 443-448.*

[267] *Graboski, M. S. and Daubert, T. E. 1978. “A Modified Soave Equation of State for Phase Equilibrium Calculations: Systems Containing CO2, H2S, N2, and CO”, Ind. Eng. Chem. Process Des. Dev., 17(4), pp. 448-454.*

[268] *Guénette, R. and Fortin, M. 1995. " A new mixed finite element method for computing viscoelastic flows", J. Non-Newton. Fluid, 60, pp. 27-52.*

[269] *Gupta, R. N., Lee, K. P., Thompson, R. A., and Yos, J. M. 1991. “Calculations and Curve Fits of Thermodynamic and Transport Properties for Equilibrium Air to 30000 K”, NASA STI/Recon Technical Report N, 92, 11285.*

[270] *Hansen, M. 2007. “Aerodynamics of Wind Turbines, 2nd ed. Routledge.*

[271] *Hosangadi, A., Sachdev, J. , Sankaran, V. 2012. "Improved Flux Formulations for Unsteady Low Mach Number Flows". ICCFD7 proceedings.*

[272] *Huang, H., and Ayoub, J. 2006. “Applicability of the Forchheimer Equation for Non-Darcy Flow in Porous Media”. SPE Journal, March 2008, pp. 112-122.*

[273] *Hughes, T. J. R.,Franca, L. P. and Balestra, M. 1986. " A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuska-Brezzi condition: a stable Petrov-Galerkin formulation of the stokes problem accommodating equal-order interpolations", Comput. Methods in Appl. Mech. Eng., 59, pp. 85-99.*

[274] *Hughes, T. J. R. and Franca, L. P. 1987. " A new finite element formulation for computational fluid dynamics: VII. The stokes problem with various well-posed boundary conditions: Symmetric formulations that converge for all velocity/pressure spaces", Comput. Methods in Appl. Mech. Eng., 65, pp. 85-96.*

[275] *International Association for the Properties of Water and Steam. 2007. Revised Release on the IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam, IAPWS Secretariat, Lucerne, Switzerland. (available at www.iapws.org).*

[276] *Issa, R. I. , Gosman, A. D., and Watkins, A. P., 1986. "The Computation of Compressible and Incompressible Recirculating Flows by a Non-Iterative Implicit Scheme," J. Computational Physics, 62(1).*

[277] *Jameson, A., Schmidt, W. and Turkel, E. 1981. “Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes”, AIAA Paper 81-1259.*

[278] *Ladyzhenskaya, O. A. 1963. " The Mathematical Theory of Viscous Incompressible Flow" , Gordon and Breach.*

[279] *Legat, V., Marchal, J.M. 1992. "Predictions of threedimensional general shape extrudates by an implicit iterative scheme," Int. J. for Numerical Methods in Fluids, 14, 609-625.*

[280] *Leonard, B. P. 1991. “The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection”, Comp. Methods in Applied Mechanics and Engineering, 88 pp. 17-74.*

[281] *Liou, M.-S. 1996. “A Sequel to AUSM: AUSM+”, J. Comput. Phys., 129: 364-382.*

[282] *Liou, M.-S. 2001. “Ten Years in the Making--AUSM-Family”, AIAA 2001-2521.*

[283] *Lockwood, B. A. 2012. “Gradient-based Approaches for Sensitivity Analysis and Uncertainty Quantification within Hypersonic Flows”, Ph.D. Dissertation, University of Wyoming.*

[284] *Loehmer, R. 2009. “On Limiters for Minimal Vorticity Dissipation”, AIAA 2009-135.*

[285] *Lynn, N. and Steinhoff, J. 2007. “Large Reynolds Number Turbulence Modeling with Vorticity Confinement”, AIAA 2007-3965.*

[286] *Makkonen, L. 2000. “Models for the growth of rime, glaze, icicles and wet snow on structures.” Phil. Trans. R. Soc. Lond. A, 358, pp. 2913-2939.*

[287] *Manwell, J.F., McGowan, J., and Rogers, A. 2009. “Wind Energy Explained: Theory, Design and Application”, 2nd ed. Wiley.*

[288] *Mathur, S.R. and Murthy, J.Y. 1997. “Pressure-based method for unstructured meshes”, Numerical Heat Transfer, Part B: Fundamentals, 31(2), pp. 195-214.*

[289] *Mathur, S.R. and Murthy, J.Y. 1997. “Pressure boundary conditions for incompressible flow using unstructured meshes”, Numerical Heat Transfer, Part B: Fundamentals, 32(3), pp. 283-298.*

[290] *Murthy, J. Y. 2002. “Numerical Methods in Heat, Mass and Momentum Transfer”, Draft Notes, School of Mechanical Engineering, Purdue University.*

[291] *Oliveira, P. J., Pinho, F. T. and Pinto, G. A., G. 1998. “Numerical simulation of non-linear elastic flows with a general collocated finite-volume method”, J. Non-Newton Fluid, 79, pp. 1-43.*

[292] *Peric, M., Kressler, R., and Scheuerer, G. 1988. “Comparison of finite-volume numerical methods with staggered and colocated grids”, Computers & Fluids, 16(4), pp. 389-403.*

[293] *Poling, B. E., Prausnitz, J. M., and O'Connell, J. P., “The Properties of Gases & Liquids”, 5th Ed., McGraw-Hill, 2001.*

[294] *Rajagopalan, D., Armstrong, R.C., Brown, R.A. 1990. "Finite element methods for calculation of steady viscoelastic flow using constitutive equations with a Newtonian viscosity", J. Non-Newtonian Fluid Mech., 36, pp 159-192.*

[295] *Rajagopalan, R.G. 1993. “Three Dimensional Analysis of a Rotor in Forward Flight”, Journal of the American Helicopter Society, Vol. 38.*

[296] *Reid, R. C., Prausnitz, J. M., and Poling, B. E. 1987. “The Properties of Gases & Liquids”, 4th Ed., McGraw-Hill.*

[297] *Saxer, A.P. 1992. “A numerical analysis of three-dimensional inviscid rotor/stator interactions using non-reflecting boundary conditions”, Ph.D. Thesis. MIT Dept. of Aero. and Astro.*

[298] *Saad, Y. 2003. “Iterative Methods for Sparse Linear Systems”, 2nd ed., Society for Industrial and Applied Mathematics*

[299] *Shakib, F., Hughes, T. J. R. and Johan, Z. 1991. " Second World Congress on Computational Mechanics A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier-Stokes equations" , Comput. Methods in Appl. Mech. Eng., 89, pp. 141-219.*

[300] *Shakib Tezduyar_Rev, Tezduyar_PSPG*

[301] *Soave, G. 1972. “Equilibrium Constants from a Modified Redlich–Kwong Equation of State”, Chem. Eng. Sci., 27, pp. 1197-1203.*

[302] *Steinhoff, J., Lynn, N. and Wang, L. 2005. “Computation of High Reynolds Number Flows Using Vorticity Confinement: I. Formulation”, UTSI Preprint.*

[303] *Tezduyar, T. E. and Osawa, Y. 2000. " Finite element stabilization parameters computed from element matrices and vectors" , Comput. Methods in Appl. Mech. Eng., 190, pp. 411-430.*

[304] *Turkel, E. 1987. “Preconditioned methods for solving the incompressible and low speed compressible equations”, Journal of Computational Physics, 72, pp. 277-298.*

[305] *Weiss, J.M., Maruszewski, J.P., and Smith, W.A. 1999. “Implicit solution of preconditioned Navier-Stokes equations using algebraic multigrid”, AIAA Journal, 37(1), pp. 29-36.*

[306] *Weiss, J.M., and Smith, W.A. 1995. “Preconditioning applied to variable and constant density flows”, AIAA Journal, 33(11), pp. 2050-2057.*

[307] *Venkatakrishnan, V. 1994.“On the Convergence of Limiters and Convergence to Steady State Solutions”, AIAA-93-0880.*

[308] *Xue, S.-C.,Phan-Thien, N. and Tanner, R. I. 1995. “Numerical study of secondary flows of viscoelastic fluid in straight pipes by an implicit finite volume method”, J. Non-Newton Fluid, 59, pp. 191-213.*

[309] *Yang, Z. et al. 2000. “Recent Improvements to a Hybrid Method for Rotors in Forward Flight”, AIAA Conference Reno.*

~Moving Reference Frames~

For this release, see *User Guide* > *Using Tools* > *Reference Frames* > *Governing Equations in Moving Reference Frames*.

Theory guide content planned for future release.

Turbulence

Most fluid flows of engineering interest are characterized by irregularly fluctuating flow quantities. Often these fluctuations are at such small scales and high frequencies that resolving them in time and space comes at excessive computational costs. Instead of solving for the exact governing equations of turbulent flows (Direct Numerical Simulation), it is less expensive to solve for averaged or filtered quantities and approximate the impact of the small fluctuating structures. Turbulence models provide different approaches for modeling these structures.

The turbulence models that are implemented in STAR-CCM+ can be subdivided into two categories:

- Reynolds-Averaged Navier-Stokes turbulence models
- Scale-resolving simulations

It is generally recognised that all existing turbulence models are approximate representations of the physical phenomena of turbulence. The degree of approximation in a given model depends on the nature of the flow to which it is applied. The characterization of the circumstances which give rise to 'good' and 'bad' performance must be based mainly on experience.

Contents:

Velocity Gradient and Invariants
Non-Dimensional Quantities
Reynolds-Averaged Navier-Stokes (RANS) Turbulence Models
Scale-Resolving Simulations
Turbulent Time and Length Scales
Turbulence Bibliography

Velocity Gradient and Invariants

In the context of turbulence modelling, it is useful to define a number of quantities (scalars and tensors) related to the velocity gradient.

One quantity related to the velocity gradient is the modulus of the mean strain rate tensor S. It is defined as:

$$S = |\mathbf{S}| = \sqrt{2\mathbf{S}:\mathbf{S}^T} = \sqrt{2\mathbf{S}:\mathbf{S}} \tag{1744}$$

The mean strain rate tensor $\mathbf{S}$ is given by:

$$\mathbf{S} = \frac{1}{2}\left(\nabla\overline{\mathbf{v}} + \nabla\overline{\mathbf{v}}^T\right) \tag{1745}$$

where $\overline{\mathbf{v}}$ is the mean velocity.

The modulus of the mean vorticity tensor is defined in a similar way as:

$$W = |\mathbf{W}| = \sqrt{2\mathbf{W}:\mathbf{W}^T} \tag{1746}$$

where:

$$\mathbf{W} = \frac{1}{2}\left(\nabla\overline{\mathbf{v}} - \nabla\overline{\mathbf{v}}^T\right) \tag{1747}$$

For a rotating frame of reference, the absolute vorticity tensor is defined as:

$$\widetilde{\mathbf{W}} = \mathbf{W} + \mathbf{E}\boldsymbol{\omega}^f \tag{1748}$$

where $\mathbf{E}$ is the three-dimensional permutation tensor and ω^f is the frame rotation rate.

Non-Dimensional Quantities

The most common non-dimensional quantity used in turbulence modeling is the Reynolds number, which is the ratio of inertial over viscous forces. The following definitions are used throughout the theory of turbulence modeling:

Wall-distance Reynolds number:

$$Re_d = \frac{\sqrt{k}d}{\nu} \tag{1749}$$

where:

- k is the turbulent kinetic energy.
- d is the distance to the wall.
- ν is the kinematic viscosity.

Turbulent Reynolds number:

$$Re_t = \frac{k^2}{\nu\varepsilon} \tag{1750}$$

or:

$$Re_t = \frac{k}{\nu\omega} \tag{1751}$$

where:

- ε is the turbulent dissipation rate.
- ω is the specific dissipation rate.

Kolmogorov Reynolds number:

$$Re_\varepsilon = \frac{(\nu\varepsilon)^{1/4}d}{\nu} \tag{1752}$$

Another useful number is the Rayleigh number, defined as:

$$Ra = \frac{\left\|g_i\right\|\beta\Delta T L_{ref}^3}{\nu\alpha} \tag{1753}$$

where:

- $\left\|g_i\right\|$ is the norm of the gravity vector.
- β is the thermal expansion coefficient.
- ΔT is the temperature difference (typically between fluid and containing walls).
- L_{ref} is the characteristic length.
- α is the thermal diffusivity.

α is usually defined as $\alpha = \frac{\nu}{Pr}$, with Pr the Prandtl number defined as:

$$Pr = \frac{C_p \mu}{\kappa} \tag{1754}$$

where:

- C_p is the specific heat.
- μ is the dynamic viscosity.
- κ is the thermal conductivity.

Reynolds-Averaged Navier-Stokes (RANS) Turbulence Models

RANS turbulence models provide closure relations for the Reynolds-Averaged Navier-Stokes equations, that govern the transport of the mean flow quantities.

To obtain the Reynolds-Averaged Navier-Stokes equations, each solution variable ϕ in the instantaneous Navier-Stokes equations is decomposed into its mean, or averaged, value $\overline{\phi}$ and its fluctuating component ϕ':

$$\phi = \overline{\phi} + \phi' \tag{1755}$$

where ϕ represents velocity components, pressure, energy, or species concentration.

The averaging process may be thought of as time-averaging for steady-state situations and ensemble averaging for repeatable transient situations. Inserting the decomposed solution variables into the Navier-Stokes equations results in equations for the mean quantities.

The mean mass and momentum transport equations can be written as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \overline{\mathbf{v}}) = 0 \tag{1756}$$

$$\frac{\partial}{\partial t}\left(\rho \overline{\mathbf{v}}\right) + \nabla \cdot (\rho \overline{\mathbf{v}} \otimes \overline{\mathbf{v}}) = -\nabla \cdot \overline{p}\mathbf{I} + \nabla \cdot \left(\mathbf{T} + \mathbf{T}_t\right) + \mathbf{f}_b \tag{1757}$$

where:

- ρ is the density.
- $\overline{\mathbf{v}}$ and $\overline{p}$ are the mean velocity and pressure respectively.
- $\mathbf{I}$ is the identity tensor.
- $\mathbf{T}$ is the viscous stress tensor.
- $\mathbf{f}_b$ is the resultant of the body forces (such as gravity and centrifugal forces).

These equations are essentially identical to the original Navier-Stokes equations (see Eqn. (1450) and Eqn. (1452)), except that an additional term now appears in the momentum transport equation. This additional term is a tensor quantity, known as the Reynolds stress tensor, which has the following definition:

$$\mathbf{T}_t = -\rho \begin{pmatrix} \overline{u'u'} & \overline{u'v'} & \overline{u'w'} \\ \overline{u'v'} & \overline{v'v'} & \overline{v'w'} \\ \overline{u'w'} & \overline{v'w'} & \overline{w'w'} \end{pmatrix} \tag{1758}$$

The challenge is thus to model $\mathbf{T}_t$ in terms of the mean flow quantities, and hence provide closure of the governing equations. Two basic approaches are used in STAR-CCM+:

- Eddy viscosity models
- Reynolds stress transport models

Contents:

Eddy Viscosity Models

Reynolds Stress Transport (RST) Models
RANS Boundary, Region, and Initial Conditions
~RANS Wall Treatment~

Eddy Viscosity Models

Eddy viscosity models are based on the analogy between the molecular gradient-diffusion process and turbulent motion.

The concept of a turbulent eddy viscosity μ_t makes it possible to model the Reynolds stress tensor as a function of mean flow quantities. The most common model is known as the Boussinesq approximation:

$$\mathbf{T}_t = 2\mu_t \mathbf{S} - \frac{2}{3}(\mu_t \nabla \cdot \bar{\mathbf{v}})\mathbf{I} \tag{1759}$$

where:

- $\mathbf{S}$ is the mean strain rate tensor given by Eqn. (1745).
- $\bar{\mathbf{v}}$ is the mean velocity.
- $\mathbf{I}$ is the identity tensor.

While some simpler models rely on the concept of mixing length to model the turbulent viscosity in terms of mean flow quantities (similar to the Smagorinsky Subgrid Scale model used in Large Eddy Simulation (LES)), the eddy viscosity models in STAR-CCM+ solve additional transport equations for scalar quantities that enable the turbulent viscosity μ_t to be derived. These include the following turbulence models:

- Spalart-Allmaras model
- K-Epsilon model
- K-Omega model
- Elliptic Blending model
- V2F model

The assumption that the Reynolds stress tensor is linearly proportional to the mean strain rate does not consider anisotropy of turbulence. In order to account for this turbulence anisotropy, some two-equation models are provided with an option to extend the linear approximation to include non-linear constitutive relations.

Contents:

Spalart-Allmaras Model
K-Epsilon Model
K-Omega Model
Elliptic Blending Model
V2F Model
Curvature Correction
Turbulent Viscosity User Scaling

Spalart-Allmaras Model

The Spalart-Allmaras turbulence model solves a transport equation for the modified diffusivity $\tilde{\nu}$ in order to determine the turbulent eddy viscosity.

This approach is in contrast to many of the early one-equation models that solved an equation for the transport of turbulent kinetic energy and required an algebraic prescription of a length scale. (See [314].)

The original model was developed primarily for the aerospace industry, and has the advantage of being readily implemented in an unstructured CFD solver (unlike the more traditional aerospace models, such as those of Baldwin-Lomax [310] and Johnson-King [312]). This advantage has resulted in its popularity increasing as the use of unstructured CFD methods has grown more widespread in the aerospace industry.

The authors of the original Spalart-Allmaras turbulence model presented results for attached boundary layers and flows with mild separation (such as flow past a wing). It is reasonable to expect that these cases are the types of flows for which the model yields the best results. Wilcox [315] presents free-shear flow spreading rates for the model. While acceptable results are obtained for wake, mixing layer and radial jet flows, the predicted spreading rates for plane and round jets are inaccurate. Therefore, Wilcox concludes that the model is not suited to applications involving jet-like free-shear regions. It is also likely to be less suited to flows involving complex recirculation and body forces (such as buoyancy) than two-equation models such as K-Epsilon and K-Omega or Reynolds Stress Transport.

Model Variants

Two variants of the Spalart-Allmaras model are implemented in STAR-CCM+:

Model Variant	Abbreviation
Standard Spalart-Allmaras	SSA
High-Reynolds Number Spalart-Allmaras	HRe SA

Standard Spalart-Allmaras

In its standard form, the Spalart-Allmaras model is a low-Reynolds number model, meaning that it is applied without wall functions.

According to the formulation of the model, the entire turbulent boundary layer, including the viscous sublayer, can be accurately resolved and the model can be applied on fine meshes (small values of y^+). In STAR-CCM+, this model is also available with an all-y^+ wall treatment.

High-Reynolds Number Spalart-Allmaras

In the derivation of the model, the authors present a high-Reynolds number version in which the viscous damping within the buffer layer and viscous sublayer is not included.

This form of the model, analogous to a K-Epsilon model with wall functions, is also available in STAR-CCM+. It is only suited to coarse, wall-function-type meshes where the y^+ values generally exceed about 30. The standard model with an all-y^+ wall treatment can give similar results to this model on coarse meshes.

Relation for Turbulent Viscosity

The turbulent eddy viscosity μ_t is calculated as:

$$\mu_t = \rho f_{v1} \tilde{\nu} \tag{1760}$$

where:

- ρ is the density.
- f_{v1} is a Damping Function.

Transport Equation

The transport equation for the modified diffusivity $\tilde{\nu}$ is:

$$\frac{\partial}{\partial t}\left(\rho\tilde{\nu}\right) + \nabla\cdot\left(\rho\tilde{\nu}\overline{\mathbf{v}}\right) = \frac{1}{\sigma_{\tilde{\nu}}}\nabla\cdot\left[\left(\mu+\rho\tilde{\nu}\right)\nabla\tilde{\nu}\right] + P_{\tilde{\nu}} + S_{\tilde{\nu}} \tag{1761}$$

where:

- $\overline{\mathbf{v}}$ is the mean velocity.
- $\sigma_{\tilde{\nu}}$ is a Model Coefficient.
- μ is the dynamic viscosity.
- $P_{\tilde{\nu}}$ is the Production Term.
- $S_{\tilde{\nu}}$ is the user-specified source term.

Production Term

The production term $P_{\tilde{\nu}}$ is defined as:

$$P_{\tilde{\nu}} = D_{\tilde{\nu}} + G_{\tilde{\nu}} - \Upsilon_{\tilde{\nu}} \tag{1762}$$

where the contributions to the production term are:

	Description	Formulation	where:
$D_{\tilde{\nu}}$	Non-conservative diffusion	$\frac{C_{b2}}{\sigma_{\tilde{\nu}}}\rho\left(\nabla\tilde{\nu}\cdot\nabla\tilde{\nu}\right)$ (1763)	• C_{b2} is a Model Coefficient.
$G_{\tilde{\nu}}$	Turbulent production	$\rho\left(1-f_{t2}\right)C_{b1}f_{r1}\tilde{S}\tilde{\nu}$ (1764)	• f_{t2} is a Damping Function. • C_{b1} is a Model Coefficient. • f_{r1} is the rotation function given by Eqn. (1911). • $\tilde{S}$ is the Deformation Parameter.
$\Upsilon_{\tilde{\nu}}$	Turbulent dissipation	$\rho\left(C_{w1}f_w - \frac{C_{b1}}{\kappa^2}f_{t2}\right)\left(\frac{\tilde{\nu}}{d}\right)^2$ (1765)	• $C_{w1} = \frac{C_{b1}}{\kappa^2} + \frac{1+C_{b2}}{\sigma_{\tilde{\nu}}}$

	Description	Formulation	where:
			• $f_w = g\left(\frac{1+c_{w3}^6}{g^6+c_{w3}^6}\right)^{1/6}$ where: ◦ $g = r + C_{w2}\left(r^6 - r\right)$ ◦ $r = \min\left(\frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}, 10\right)$ ◦ C_{w2} and C_{w3} are Model Coefficients. • κ is the von Karman constant, see Model Coefficients. • d is the distance to the wall.

Deformation Parameter

The deformation parameter $\tilde{S}$ is calculated as:

$$\tilde{S} = \hat{S} + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2} \tag{1766}$$

where:

- f_{v2} is a Damping Function.

Two alternatives are provided for evaluation of the scalar deformation $\hat{S}$. The original model uses the magnitude of the vorticity:

$$\hat{S} = W \tag{1767}$$

where:

- W is given by Eqn. (1746).

An alternative suggested in Dacles-Mariani and others [311] combines the strain rate and vorticity tensor magnitudes as follows:

$$\hat{S} = W + C_{\text{prod}} \min\left[0, S - W\right] \tag{1768}$$

where:

- C_{prod} is a Model Coefficient.
- S is given by Eqn. (1744).

Damping Functions

For turbulence models that resolve the viscous- and buffer-layer, damping functions mimic the decrease of turbulent mixing near the walls.

Model Variant	f_{v1}	f_{v2}	f_{t2}
SSA	$\frac{\chi^3}{\chi^3 + C_{v1}^3}$ (1769)	$1 - \frac{\chi}{1 + \chi f_{v1}}$ (1770)	$1.1\exp\left(-2\chi^2\right)$ (1771)
HRe SA	1	0	0

where:

- $\chi = \frac{\tilde{\nu}}{\nu}$
- C_{v1} is a Model Coefficient.

Model Coefficients

C_{b1}	C_{b2}	C_{w2}	C_{w3}
0.1355	0.622	0.3	2

κ	$\sigma_{\tilde{\nu}}$	C_{v1}	C_{prod}
0.41	2/3	7.1	2

K-Epsilon Model

The K-Epsilon turbulence model is a two-equation model that solves transport equations for the turbulent kinetic energy k and the turbulent dissipation rate ε in order to determine the turbulent eddy viscosity.

Various forms of the K-Epsilon model have been in use for a number of decades, and it has become the most widely used model for industrial applications. Since the inception of the K-Epsilon model, there have been countless attempts to improve it. The most significant improvements have been incorporated into STAR-CCM+.

High-Reynolds Number Approach

The original K-Epsilon turbulence model by Jones and Launder [319] was applied with wall functions. This high-Reynolds number approach was later modified to take into account the blocking effects of the wall (viscous and buffer layer) using a low-Reynolds number approach and a two-layer approach.

Low-Reynolds Number Approach

The most common approach is to apply damping functions to some or all of the coefficients in the model (C_μ, $C_{\varepsilon 1}$, and $C_{\varepsilon 2}$). These damping functions modulate the coefficients as functions of a turbulence Reynolds number, often also incorporating the wall distance. Dozens of models incorporating damping functions have been proposed in the literature, and two have been incorporated into STAR-CCM+:

- Standard K-Epsilon Low-Re model
- AKN K-Epsilon Low-Re model

Another low-Reynolds number K-Epsilon model implemented in STAR-CCM+ is the V2F Model. This model, however, does not use wall distance or damping functions; it is claimed to be valid throughout the flow domain while automatically modeling the region close to solid surfaces.

Two-Layer Approach

The two-layer approach, first suggested by Rodi [324], is an alternative to the low-Reynolds number approach that allows the K-Epsilon model to be applied in the viscous-affected layer (including the viscous sub-layer and the buffer layer).

In this approach, the computation is divided into two layers. In the layer next to the wall, the turbulent dissipation rate ε and the turbulent viscosity μ_t are specified as functions of wall distance. The values of ε specified in the near-wall layer are blended smoothly with the values computed from solving the transport equation far from the wall. The equation for the turbulent kinetic energy is solved across the entire flow domain. This explicit specification of ε and μ_t is arguably no less empirical than the damping function approach, and the results are often as good or better.

Several types of two-layer formulations have been proposed, and three have been implemented in STAR-CCM+, two for shear-driven flows and one for buoyancy-driven flows:

- Shear Driven (Wolfstein)
- Shear Driven (Norris-Reynolds)
- Buoyancy Driven (Xu)

In STAR-CCM+, the two-layer formulations work with either low-Reynolds number type meshes $y^+ \sim 1$ or wall-function type meshes $y^+ > 30$.

Model Variants

Six variants of the K-Epsilon model are implemented in STAR-CCM+:

Model Variant	Abbreviation
Standard K-Epsilon	SKE
Standard K-Epsilon Two-Layer	SKE 2L
Standard K-Epsilon Low-Re	SKE LRe
AKN K-Epsilon Low-Re	AKN KE LRe
Realizable K-Epsilon	RKE
Realizable K-Epsilon Two-layer	RKE 2L

Standard K-Epsilon

The Standard K-Epsilon model is a *de facto* standard version of the two-equation model that involves transport equations for the turbulent kinetic energy k and its dissipation rate ε. The transport equations are of the form suggested by Jones and Launder [319], with coefficients suggested by Launder and Sharma [321]. Some additional terms have been added to the model in STAR-CCM+ to account for effects such as buoyancy and compressibility. An optional non-linear constitutive relation is also provided.

Standard K-Epsilon Two-Layer

The Standard K-Epsilon Two-Layer model combines the Standard K-Epsilon model with the two-layer approach.

The coefficients in the models are identical, but the model gains the added flexibility of an all-y^+ wall treatment.

Standard K-Epsilon Low-Re

The Standard K-Epsilon Low-Reynolds Number model combines the Standard K-Epsilon model with the low-Reynolds number approach.

The low-Reynolds number model by Lien and others is dubbed the “Standard Low-Reynolds Number K-Epsilon Model” because it has identical coefficients to the Standard K-Epsilon model, but it provides more damping functions. These functions enable it to be applied in the viscous-affected regions near walls. This model is recommended for natural convection problems, particularly if the Yap correction is invoked. (See [322].)

AKN K-Epsilon Low-Re

The Abe-Kondoh-Nagano K-Epsilon Low-Reynolds Number model [316] has different coefficients than the Standard K-Epsilon model, and uses different damping functions than the Standard K-Epsilon Low-Reynolds Number model.

This model is known to work well for a wide range of complex flows. It is a good choice for applications such as compact heat exchangers, where the Reynolds numbers are low but the flow is relatively complex.

Realizable K-Epsilon

The Realizable K-Epsilon model contains a new transport equation for the turbulent dissipation rate ε [326]. Also, a critical coefficient of the model C_μ is expressed as a function of mean flow and turbulence properties, rather than assumed to be constant as in the standard model. This procedure lets the model satisfy certain mathematical constraints on the normal stresses consistent with the physics of turbulence (realizability). The concept of a variable C_μ is also consistent with experimental observations in boundary layers.

This model is substantially better than the Standard K-Epsilon model for many applications, and can generally be relied upon to give answers that are at least as accurate. Both the standard and realizable models are available in STAR-CCM+ with the option of using a two-layer approach, which enables them to be used with fine meshes that resolve the viscous sublayer.

Realizable K-Epsilon Two-Layer

The Realizable Two-Layer K-Epsilon model combines the Realizable K-Epsilon model with the two-layer approach.

The coefficients in the models are identical, but the model gains the added flexibility of an all-y^+ wall treatment.

Relation for Turbulent Viscosity

The turbulent eddy viscosity μ_t is calculated as:

$$\mu_t = \rho C_\mu f_\mu kT \tag{1772}$$

where:

- ρ is the density.
- C_μ is a Model Coefficient.
- f_μ is a Damping Function.
- T is the turbulent time scale.

The turbulent time scale is calculated as:

Model Variant	T	T with Realizable Scale Option
SKE	$\max\left(T_e, C_t\sqrt{\frac{\nu}{\varepsilon}}\right)$ (1773)	$\max\left(\min\left(T_e, \frac{C_T}{C_\mu f_\mu S}\right), C_t\sqrt{\frac{\nu}{\varepsilon}}\right)$ (1774)
SKE 2L		
SKE LRe		
AKN KE LRe		
RKE	T_e	-
RKE 2L		

where:

- $T_e = \frac{k}{\varepsilon}$ is the large-eddy time scale.
- C_t and C_T are Model Coefficients.
- ν is the kinematic viscosity.
- S is given by Eqn. (1744).

Transport Equations

The transport equations for the kinetic energy k and the turbulent dissipation rate ε are:

$$\frac{\partial}{\partial t}\left(\rho k\right) + \nabla\cdot\left(\rho k\overline{\mathbf{v}}\right) = \nabla\cdot\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\nabla k\right] + P_k - \rho\left(\varepsilon - \varepsilon_0\right) + S_k \tag{1775}$$

$$\frac{\partial}{\partial t}\left(\rho \varepsilon\right) + \nabla\cdot\left(\rho \varepsilon\overline{\mathbf{v}}\right) = \nabla\cdot\left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\nabla \varepsilon\right] + \frac{1}{T_e}C_{\varepsilon 1}P_\varepsilon - C_{\varepsilon 2}f_2\rho\left(\frac{\varepsilon}{T_e} - \frac{\varepsilon_0}{T_0}\right) + S_\varepsilon \tag{1776}$$

where:

- $\overline{\mathbf{v}}$ is the mean velocity.
- μ is the dynamic viscosity.
- σ_k, σ_ε, $C_{\varepsilon 1}$, and $C_{\varepsilon 2}$ are Model Coefficients.
- P_k and P_ε are Production Terms.
- f_2 is a Damping Function.
- S_k and S_ε are the user-specified source terms.

ε_0 is the ambient turbulence value in the source terms that counteracts turbulence decay [327]. The possibility to impose an ambient source term also leads to the definition of a specific time-scale T_0 that is defined as:

$$T_0 = \max\left(\frac{k_0}{\varepsilon_0}, C_t\sqrt{\frac{\nu}{\varepsilon_0}}\right) \tag{1777}$$

where:

- C_t is a Model Coefficient

Production Terms

The formulation of the production terms P_k and P_ε depends on the K-Epsilon model variant:

Model Variant	P_k	P_ε
SKE	$G_k + G_{nl} + G_b + \Upsilon_M$ (1778)	$G_k + G_{nl} + C_{\varepsilon 3} G_b$ (1779)
SKE 2L	$G_k + G_{nl} + G_b + \Upsilon_M$ (1780)	$G_k + G_{nl} + C_{\varepsilon 3} G_b + \frac{\rho}{C_{\varepsilon 1}} \Upsilon_y$ (1781)
SKE LRe	$G_k + G_b + \Upsilon_M$ (1782)	$G_k + G_{nl} + G' + C_{\varepsilon 3} G_b + \frac{\rho}{C_{\varepsilon 1}} \Upsilon_y$ (1783)
RKE	$f_c G_k + G_b + \Upsilon_M$ (1784)	$f_c Sk + C_{\varepsilon 3} G_b$ (1785)
RKE 2L	$f_c G_k + G_b + \Upsilon_M$ (1786)	$f_c Sk + C_{\varepsilon 3} G_b$ (1787)
AKN KE LRe	$G_k + G_b + \Upsilon_M$ (1788)	$G_k + C_{\varepsilon 3} G_b$ (1789)

where:

- $C_{\varepsilon 3}$ is a Model Coefficient.
- f_c is the curvature correction factor given by Eqn. (1912).
- the contributions to the production terms are:

	Description	Formulation	where:
G_k	Turbulent production	$\mu_t S^2 - \frac{2}{3}\rho k \nabla \cdot \bar{\mathbf{v}} - \frac{2}{3}\mu_t \left(\nabla \cdot \bar{\mathbf{v}}\right)^2$ (1790)	-
G_b	Buoyancy production	$\beta \frac{\mu_t}{Pr_t}\left(\nabla \bar{T} \cdot \mathbf{g}\right)$ (1791)	• β is the coefficient of thermal expansion. For constant density flows using the Boussinesq approximation, β is user-specified.

	Description	Formulation	where:
			For ideal gases, β is given by $\beta = -\frac{1}{\rho}\frac{\partial \rho}{\partial \overline{T}}$. • Pr_t is the turbulent Prandtl number. • $\overline{T}$ is the mean temperature. • $\mathbf{g}$ is the gravitational vector.
G_{nl}	"Non-linear" production	$\nabla \cdot \overline{\mathbf{v}}:\left(\mathbf{T}_{t,NL}\right)$ (1792)	• $\mathbf{T}_{t,NL}$ is the non-linear contribution to the Constitutive Relationship.
G'	Additional production	$Df_2\left(G_k + 2\mu\frac{k}{d^2}\right)\exp\left(-ERe_d^2\right)$ (1793)	• D and E are Model Coefficients. • f_2 is a Damping Function. • d is the distance to the wall. • Re_d is given by Eqn. (1749).
Υ_M	Compressibility modification (Sarkar et al. [325])	$\frac{C_M k \varepsilon}{c^2}$ (1794)	• C_M is a Model Coefficient. • c is the speed of sound.
Υ_y	Yap correction [330]	$C_w\frac{\varepsilon^2}{k}\max\left[\left(\frac{l}{l_\varepsilon}-1\right)\left(\frac{l}{l_\varepsilon}\right)^2, 0\right]$ (1795)	• C_w is a Model Coefficient. • l and l_ε are length scales defined as $l = \frac{k^{\frac{3}{2}}}{\varepsilon}$ and $l_\varepsilon = C_l d$, where C_l is a Model Coefficient.

Damping Functions

For turbulence models that resolve the viscous- and buffer-layer (SKE LRe, AKN KE LRe), damping functions mimic the decrease of turbulent mixing near the walls. For the RKE and RKE 2L models, damping functions enforce realizability.

Model Variant	f_2	f_μ
SKE	1	1
SKE 2L		
SKE LRe	$1 - C\exp\left(-Re_t^2\right)$ (1796)	$1-\exp\left[-\left(C_{d0}\sqrt{Re_d} + C_{d1}Re_d + C_{d2}Re_d^2\right)\right]$ (1797)

Model Variant	f_2	f_μ
AKN KE LRe	$\left[1-\exp\left(-\frac{Re_\varepsilon}{3.1}\right)\right]^2\left\{1-0.3\exp\left[-\left(\frac{Re_t}{6.5}\right)^2\right]\right\}$ (1798)	$\left[1-\exp\left(-\frac{Re_\varepsilon}{14}\right)\right]^2\left\{1+\frac{5}{Re_t^{3/4}}\exp\left[-\left(\frac{Re_t}{200}\right)^2\right]\right\}$ (1799)
RKE	$\frac{k}{k+\sqrt{\nu\varepsilon}}$ (1800)	1
RKE 2L		

where:

- C, C_{d0}, C_{d1}, and C_{d2} are Model Coefficients.
- Re_ε, Re_t, and Re_d are given by Eqn. (1752), Eqn. (1750), and Eqn. (1749), respectively.
- $\mathbf{S}$ and $\mathbf{W}$ are given by Eqn. (1745) and Eqn. (1747), respectively.
- $\mathbf{S}^* = \mathbf{S} - \frac{1}{3}\mathrm{tr}(\mathbf{S})\mathbf{I}$

Model Coefficients

Coefficient	SKE / SKE 2L	SKE LRe	AKN KE LRe	RKE / RKE 2L
C	-	0.3	-	-
C_{d0}	-	0.091	-	-
C_{d1}	-	0.0042	-	-
C_{d2}	-	0.00011	-	-
C_l	2.55	2.55	2.55	-
C_M (Sarkar)	2	2	2	2
C_t	1	1	1	1
C_T	0.6	0.6	0.6	-
C_w	0.83	0.83	0.83	-
$C_{\varepsilon 1}$	1.44	1.44	1.5	$\max\left(0.43, \frac{\eta}{5+\eta}\right)$ where: $\eta = \frac{Sk}{\varepsilon}$
$C_{\varepsilon 2}$	1.92	1.92	1.9	1.9

Coefficient	SKE / SKE 2L	SKE LRe	AKN KE LRe	RKE / RKE 2L
C_μ	0.09	0.09	0.09	$\frac{1}{4+\sqrt{6}\cos\left[\frac{1}{3}\mathrm{acos}\left(\sqrt{6}\frac{S^3}{\sqrt{S:S}^3}\right)\right]\sqrt{\mathbf{S}:\mathrm{S}+\mathbf{W}:\mathbf{W}}\frac{k}{\varepsilon}}$
D	-	1	-	-
E	-	0.00375	-	-
σ_ε	1.3	1.3	1.4	1.2
σ_k	1	1	1.4	1

$C_{\varepsilon 3}$ **(all model variants)**

The available literature is not clear as to the specification of this coefficient. By default, it is computed according to [318] as:

$$C_{\varepsilon 3} = \tanh\frac{|\mathbf{v}_b|}{|\mathbf{u}_b|} \tag{1801}$$

where $\mathbf{v}_b$ and $\mathbf{u}_b$ are the velocity components parallel and perpendicular to the gravitational vector $\mathbf{g}$.

This formulation tends to set the coefficient to zero outside natural convection boundary layers.

Alternatively, $C_{\varepsilon 3}$ can be taken as constant everywhere, or specified depending on the buoyancy production term G_b as follows:

$$C_{\varepsilon 3} = \begin{cases} 1 & \text{for } G_b \geq 0 \\ 0 & \text{for } G_b < 0 \end{cases} \tag{1802}$$

Contents:

Two-Layer Models
Constitutive Relations

Two-Layer Models

A two-layer model solves for k but prescribes ε algebraically with distance from the wall in the viscosity dominated near-wall flow regions. In STAR-CCM+, this approach is available with both the Standard and the Realizable K-Epsilon models.

For the two-layer models the dissipation rate near the wall is simply prescribed as:

$$\varepsilon = \frac{k^{3/2}}{l_\varepsilon} \tag{1803}$$

where:

- l_ε is a length scale function that is calculated depending on the Model Variant.

As per the suggestion of Jongen [320], the following wall-proximity indicator is used to combine the two-layer formulation with the full two-equation model:

$$\lambda = \frac{1}{2}\left[1 + \tanh\left(\frac{\mathrm{Re}_d - \mathrm{Re}_y^*}{A}\right)\right] \tag{1804}$$

where:

- Re_d is the wall-distance Reynolds number given by Eqn. (1749).
- Re_y^* defines the limit of applicability of the two-layer formulation and is a Model Coefficient.
- A determines the width of the wall-proximity indicator and is defined such that the value of λ is within 1% of its far-field value for a given variation of $\Delta\ \mathrm{Re}_y$:

$$A = \frac{\left|\Delta\ \mathrm{Re}_y\right|}{\operatorname{atanh} 0.98} \tag{1805}$$

  where $\Delta\ \mathrm{Re}_y$ is a Model Coefficient.

The turbulent viscosity from the K-Epsilon model $\mu_t\big|_{k-\epsilon}$ is then blended with the two-layer value as follows:

$$\mu_t = \lambda\mu_t\big|_{k-\epsilon} + (1-\lambda)\mu\left(\frac{\mu_t}{\mu}\right)_{\mathrm{2layer}} \tag{1806}$$

where:

- $\left(\frac{\mu_t}{\mu}\right)_{\mathrm{2layer}}$ is calculated depending on the Model Variant.

Model Variants

In STAR-CCM+ three variants of the two-layer formulation are available:

Model Variant	l_ε	$\left(\frac{\mu_t}{\mu}\right)_{\mathrm{2layer}}$
Wolfstein [328]	$c_l d\left[1-\exp\left(-\frac{\mathrm{Re}_d}{2c_l}\right)\right]$ (1807)	$0.42\mathrm{Re}_d C_\mu^{1/4}\left[1-\exp\left(-\frac{\mathrm{Re}_d}{70}\right)\right]$ (1808)
Norris-Reynolds [323]	$c_l d\frac{\mathrm{Re}_d}{\mathrm{Re}_d + 5.3}$ (1809)	$0.42\mathrm{Re}_d C_\mu^{1/4}\left[1-\exp\left(-\frac{\mathrm{Re}_d}{50.5}\right)\right]$ (1810)
Xu [329] (specifically for natural convection)	$\frac{8.8d}{1 + 10/y_v{}^* + 5.15\times 10^{-2} y_v{}^*}\frac{1}{\sqrt{v^2}/k}$ (1811)	$\frac{0.544 y_v{}^*}{1 + 5.025\times 10^{-4} y_v{}^{*1.65}}$ (1812)

where:

- $c_l = 0.42 C_\mu^{-3/4}$.
- C_μ is given in K-Epsilon Model—Model Coefficients.

- d is the distance to the wall.
- $y_v{}^* = \mathrm{Re}_d\sqrt{\frac{\overline{v^2}}{k}}$

  where:

  $\frac{\overline{v^2}}{k} = 7.19 \times 10^{-3}\mathrm{Re}_d - 4.33 \times 10^{-5}\mathrm{Re}_d^2 + 8.8 \times 10^{-8}\mathrm{Re}_d^3.$

Model Coefficients

$\mathrm{Re}_y{}^*$	$\Delta\ \mathrm{Re}_y$
60	10

Constitutive Relations

Constitutive relations describe the relation between the Reynolds stress tensor and the mean strain rate used in the Boussinesq approximation.

By default, the Boussinesq approximation implies a linear constitutive relation. Non-linear constitutive relations [322] account for anisotropy of turbulence by adding non-linear functions of the strain and rotation tensors. Two non-linear constitutive relations are available for the Standard K-Epsilon model variants (SKE, SKE 2L, and SKE LRe) in STAR-CCM+:

Constitutive Relation	Formulation
Quadratic	$$\mathbf{T}_{t,NL} = -4\mu_t\frac{k}{\varepsilon}\left\{C_1\left[\mathbf{S}\cdot\mathbf{S}-\frac{1}{3}\mathbf{I}(\mathbf{S}:\mathbf{S})\right] + C_2(\mathbf{W}\cdot\mathbf{S}+\mathbf{S}\cdot\mathbf{W}^T) + C_3\left[\mathbf{W}\cdot\mathbf{W}^T-\frac{1}{3}\mathbf{I}(\mathbf{W}:\mathbf{W}^T)\right]\right\}$$ (1813) where: $C_1 = \frac{C_{NL1}}{(C_{NL6}+C_{NL7}\overline{S}^3)C_\mu}$ $C_2 = \frac{C_{NL2}}{(C_{NL6}+C_{NL7}\overline{S}^3)C_\mu}$ $C_3 = \frac{C_{NL3}}{(C_{NL6}+C_{NL7}\overline{S}^3)C_\mu}$
Cubic	$$\mathbf{T}_{t,NL} = \mathbf{T}_{t,quad}-8\mu_t\frac{k^2}{\varepsilon^2}\left\{C_4\left[(\mathbf{S}\cdot\mathbf{S})\cdot\mathbf{W}+\mathbf{W}^T\cdot(\mathbf{S}\cdot\mathbf{S})\right] + C_5(\mathbf{S}:\mathbf{S}-\mathbf{W}:\mathbf{W}^T)\left[\mathbf{S}-\frac{1}{3}\mathrm{tr}(\mathbf{S})\mathbf{I}\right]\right\}$$ (1814) where $\mathbf{T}_{t,quad}$ is given by Eqn. (1813) and: $C_4 = C_{NL4}C_\mu{}^2$

Constitutive Relation	Formulation
	$C_5 = C_{NL5} C_\mu{}^2$

where:

- $\mathbf{S}$ and $\mathrm{tr}(\mathbf{S})$ are given by Eqn. (1745) and Eqn. (4155), respectively.
- $\mathbf{I}$ is the identity tensor.
- $\mathbf{W}$ is given by Eqn. (1747).
- C_{NL1}, C_{NL2}, C_{NL3}, C_{NL4}, C_{NL5}, C_{NL6}, and C_{NL7} are Model Coefficients.

The coefficient C_μ is given by:

$$C_\mu = \frac{C_{a0}}{C_{a1} + C_{a2}\overline{S} + C_{a3}\overline{W}} \tag{1815}$$

where:

- C_{a0}, C_{a1}, C_{a2}, and C_{a3} are Model Coefficients.
- $\overline{S} = \frac{k}{\varepsilon}\sqrt{2\mathbf{S}:\mathbf{S}}$
- $\overline{W} = \frac{k}{\varepsilon}\sqrt{2\mathbf{W}:\mathbf{W}}$

When non-linear constitutive relations are used, the variable coefficient C_μ (Eqn. (1815)) replaces the constant value of C_μ in the relation for the turbulent viscosity μ_t, see Eqn. (1772).

For the momentum equation Eqn. (1462), the stress tensor $\mathbf{T}$ is defined as:

$$\mathbf{T} \equiv \mathbf{T}_t + \mathbf{T}_{t,NL} \tag{1816}$$

where $\mathbf{T}_t$ is given by Eqn. (1759).

Model Coefficients

C_{NL1}	C_{NL2}	C_{NL3}	C_{NL4}	C_{NL5}	C_{NL6}	C_{NL7}
0.75	3.75	4.75	-10	-2	1000	1

C_{a0}	C_{a1}	C_{a2}	C_{a3}
0.667	1.25	1	0.9

K-Omega Model

The K-Omega turbulence model is a two-equation model that solves transport equations for the turbulent kinetic energy k and the specific dissipation rate ω—the dissipation rate per unit turbulent kinetic energy ($\omega \propto \varepsilon/k$)—in order to determine the turbulent eddy viscosity.

The book by D.C. Wilcox [339] is the most comprehensive reference on the K-Omega model, discussing the origin of the model, comparing it to other models, and presenting the latest version of the model. As the originator of the K-Omega model, Wilcox claims the superiority of his model over the K-Epsilon model, and the superiority of the Omega transport equation over other scale equations.

One reported advantage of the K-Omega model over the K-Epsilon model is its improved performance for boundary layers under adverse pressure gradients. Perhaps the most significant advantage, however, is that it may be applied throughout the boundary layer, including the viscous-dominated region, without further modification. Furthermore, the standard K-Omega model can be used in this mode without requiring the computation of wall distance.

The biggest disadvantage of the K-Omega model, in its original form, is that boundary layer computations are sensitive to the values of ω in the free-stream. This translates into extreme sensitivity to inlet boundary conditions for internal flows, a problem that does not exist for the K-Epsilon models. The K-Omega model variants included in STAR-CCM+ have been modified in an attempt to address this shortcoming.

Model Variants

Two variants of the K-Omega model are implemented in STAR-CCM+:

Model Variant	Abbreviation
Standard K-Omega	SKO
SST K-Omega	SSTKO

Standard K-Omega

Wilcox revised his original model in 1998, and then in 2006, to account for several perceived deficiencies in the original version (1988). These revisions include:

- A revised set of model coefficients
- Two corrections to account for sensitivity to free-stream/inlet conditions, both based on products of ∇k and $\nabla \omega$. The first model, introduced in 1998, is a modification of the turbulent kinetic energy equation. The more recent revision [340] is the introduction of a cross-diffusion term that is similar to that used in the SST K-Omega model.
- A correction to improve the free-shear-flow spreading rates of the model
- A compressibility correction
- Low-Reynolds number corrections that allow the K-Omega model to be better applied in the prediction of low-Reynolds number and transitional flows

However, the validation results published in Wilcox's book are typically for two-dimensional, primarily parabolic, flows. Until further validations for complex flows are widely published, the corrections should be used with caution. Therefore, each correction is included as an option in STAR-CCM+.

SST K-Omega

The problem of sensitivity to free-stream/inlet conditions was addressed by Menter [334], who recognized that the ε transport equation from the Standard K-Epsilon model could be transformed into an ω transport equation by variable substitution.

The transformed equation looks similar to the one in the Standard K-Omega model, but adds an additional non-conservative cross-diffusion term containing the dot product $\nabla k \cdot \nabla \omega$. Inclusion of this term in the ω

transport equation potentially makes the K-Omega model give identical results to the K-Epsilon model. Menter suggested using a blending function (which includes functions of wall distance) that would include the cross-diffusion term far from walls, but not near the wall. This approach effectively blends a K-Epsilon model in the far-field with a K-Omega model near the wall. Purists may object that the blending function crossover location is arbitrary, and could obscure some critical feature of the turbulence. Nevertheless, the fact remains that this approach cures the biggest drawback to applying the K-Omega model to practical flow simulations.

Menter also introduced a modification to the linear constitutive equation and named the model containing this modification the SST (shear-stress transport) K-Omega model. However, the linear relation between the Reynolds stresses and the mean strain rate tends to strongly under predict the anisotropy of turbulence. Turbulence is anisotropic in most complex flows—for example in strong swirl, streamline curvature, shear layer, or boundary layer flows. The anisotropy of the Reynolds stresses not only affects the flow field but also the turbulent transport of scalars (temperature, concentration, passive scalar).

To account for anisotropy of turbulence, STAR-CCM+ offers the following non-linear constitutive relations:

- Quadratic—as suggested by Spalart [336].
- Cubic—as proposed by Wallin and Johansson [338] and further improved by Hellsten [333]

The cubic constitutive relation is derived from a Reynolds stress transport model and, thus, represents an Explicit Algebraic Reynolds Stress Model (EARSM). Reynolds stress transport models perform much better than eddy viscosity models in anisotropic turbulence, but are often unstable when used in complex flows. Therefore, the cubic model is a good compromise between the two.

The SST model has seen fairly wide application in the aerospace industry, where viscous flows are typically resolved, and turbulence models are applied throughout the boundary layer.

Relation for Turbulent Viscosity

The turbulent eddy viscosity μ_t is calculated as:

$$\mu_t = \rho k T \tag{1817}$$

where:

- ρ is the density.
- T is the turbulent time scale.

The turbulent time scale is calculated using Durbin's realizability constraint [332] as:

Model Variant	T	T with Realizable Scale Option
SKO	$\frac{\alpha^*}{\omega}$ (1818)	$\min\left(\frac{\alpha^*}{\omega}, \frac{C_T}{\sqrt{3}S}\right)$ (1819)
SSTKO	$\min\left(\frac{\alpha^*}{\omega}, \frac{\mathrm{a}_1}{SF_2}\right)$ (1820)	$\min\left(\frac{1}{\max(\omega/\alpha^*, (SF_2)/\mathrm{a}_1)}, \frac{C_T}{\sqrt{3}S}\right)$ (1821)

where:

- α^*, a_1, and C_T are Model Coefficients.

- S is given by Eqn. (1744).

F_2 is a blending function calculated as:

$$F_2 = \tanh\left(\left(\max\left(\frac{2\sqrt{k}}{\beta^*\omega d}, \frac{500\nu}{d^2\omega}\right)\right)^2\right) \tag{1822}$$

where:

- β^* is a Model Coefficient.
- d is the distance to the wall.

For SSTKO, Menter's original model [334] uses the modulus of the vorticity tensor rather than S. This slight modification extends the applicability of the model beyond aerodynamic applications. Furthermore, Durbin's realizability constraint is used instead of Menter's proposal—that turbulent production be limited to some multiple of the dissipation.

Transport Equations

The transport equations for the kinetic energy k and the specific dissipation rate ω are:

$$\frac{\partial}{\partial t}(\rho k) + \nabla\cdot(\rho k\overline{\mathbf{v}}) = \nabla\cdot\left[\left(\mu + \sigma_k\mu_t\right)\nabla k\right] + P_k - \rho\beta^* f_{\beta^*}\left(\omega k - \omega_0 k_0\right) + S_k \tag{1823}$$

$$\frac{\partial}{\partial t}(\rho\omega) + \nabla\cdot(\rho\omega\overline{\mathbf{v}}) = \nabla\cdot\left[\left(\mu + \sigma_\omega\mu_t\right)\nabla\omega\right] + P_\omega - \rho\beta f_\beta\left(\omega^2 - \omega_0^2\right) + S_\omega \tag{1824}$$

where:

- $\overline{\mathbf{v}}$ is the mean velocity.
- μ is the dynamic viscosity.
- σ_k, and σ_ω, $C_{\varepsilon 1}$, and $C_{\varepsilon 2}$ are Model Coefficients.
- P_k and P_ω are Production Terms.
- f_{β^*} is the free-shear modification factor.
- f_β is the vortex-stretching modification factor.
- S_k and S_ω are the user-specified source terms.
- k_0 and ω_0 are the ambient turbulence values that counteract turbulence decay [337].

Production Terms

The formulation of the production terms P_k and P_ω depends on the K-Omega model variant:

Model Variant	P_k	P_ω
SKO	$G_k + G_b$	G_ω
SSTKO	$G_k + G_{nl} + G_b$	$G_\omega + D_\omega$

where:

- the contributions to the production terms are:

	Description	Formulation	where:
G_k	Turbulent production	$\mu_t f_c S^2 - \frac{2}{3}\rho k \nabla\cdot \overline{\mathbf{v}} - \frac{2}{3}\mu_t \left(\nabla\cdot \overline{\mathbf{v}}\right)^2$ (1825)	• f_c is the curvature correction factor given by Eqn. (1912).
G_b	Buoyancy production	$\beta \frac{\mu_t}{Pr_t}\left(\nabla\overline{T}\cdot \mathbf{g}\right)$ (1826)	• β is the coefficient of thermal expansion. For constant density flows using the Boussinesq approximation, β is user-specified. For ideal gases, β is given by $\beta = -\frac{1}{\rho}\frac{\partial\rho}{\partial\overline{T}}$ • Pr_t is the turbulent Prandtl number. • $\overline{T}$ is the mean temperature. • $\mathbf{g}$ is the gravitational vector.
G_{nl}	"Non-linear" production	$\nabla\cdot\overline{\mathbf{v}}:\left(\mathbf{T}_{t,NL}\right)$ (1827)	• $\mathbf{T}_{t,NL}$ is a non-linear Constitutive Relation.
G_ω	Specific dissipation production	SKO: $\rho\alpha\left[\left(\alpha^* S^2 - \frac{2}{3}\left(\nabla\cdot\overline{\mathbf{v}}\right)^2\right) - \frac{2}{3}\omega\nabla\cdot\overline{\mathbf{v}}\right]$ (1828) SSTKO: $\rho\gamma\left[\left(S^2 - \frac{2}{3}\left(\nabla\cdot\overline{\mathbf{v}}\right)^2\right) - \frac{2}{3}\omega\nabla\cdot\overline{\mathbf{v}}\right]$ (1829)	• α, α^*, and γ are Model Coefficients.
D_ω	Cross-diffusion term	$2\rho\left(1-F_1\right)\sigma_{\omega_2}\frac{1}{\omega}\nabla k\cdot\nabla\omega$ (1830)	• F_1 is given by Eqn. (1839). • σ_{ω_2} is a Model Coefficient.

Model Coefficients

Coefficient	SKO	SSTKO
a_1	-	0.31
α	0.52	-
α^*	1	$F_1\alpha_1^* + \left(1-F_1\right)\alpha_2^*$ (1831)
α_1^*	-	1
α_2^*	-	1
β	0.072	$F_1\beta_1 + \left(1-F_1\right)\beta_2$

Coefficient	SKO	SSTKO
		(1832)
β_1	-	0.075
β_2	-	0.0828
β^*	0.09	$F_1\beta_1^* + \left(1-F_1\right)\beta_2^*$ (1833)
β_1^*	-	0.09
β_2^*	-	0.09
γ	-	$F_1\gamma_1 + \left(1-F_1\right)\gamma_2$ (1834)
γ_1	-	$\frac{\beta_1}{\beta^*} - \sigma_{\omega_1}\frac{\kappa^2}{\sqrt{\beta^*}}$ (1835)
γ_2	-	$\frac{\beta_2}{\beta^*} - \sigma_{\omega_2}\frac{\kappa^2}{\sqrt{\beta^*}}$ (1836)
κ	-	0.41
σ_k	0.5	$F_1\sigma_{k_1} + \left(1-F_1\right)\sigma_{k_2}$ (1837)
σ_{k_1}	-	0.85
σ_{k_2}	-	1
σ_ω	0.5	$F_1\sigma_{\omega_1} + \left(1-F_1\right)\sigma_{\omega_2}$ (1838)
σ_{ω_1}	-	0.5
σ_{ω_2}	-	0.856
C_T	0.6	0.6

The blending function F_1 combines the near-wall contribution of a coefficient with its value far away from the wall and is defined as:

$$F_1 = \tanh\left(\left[\min\left(\max\left(\frac{\sqrt{k}}{0.09\omega d}, \frac{500\nu}{d^2\omega}\right), \frac{2k}{d^2\mathrm{CD}_{k\omega}}\right)\right]^4\right) \tag{1839}$$

where:

- d is the distance to the wall.
- ν is the kinematic viscosity.
- $\mathrm{CD}_{k\omega} = \max\left(\frac{1}{\omega}\nabla k \cdot \nabla\omega, 10^{-20}\right)$ (cross-diffusion coefficient)

Contents:

Free-Shear Modification
Vortex Stretching Modification
Low Reynolds Number Modification
Constitutive Relations
Compressibility Correction

Free-Shear Modification

For flows that are dominated by free-shear layers (such as jets or mixing layers) or when free-stream turbulence is high, the Standard K-Omega model tends to underpredict the production of ω. Free-shear modifications are designed to counteract this effect.

The following free-shear modifications are available in STAR-CCM+:

Free-Shear Modification	Formulation
Dissipation Limiter	f_{β^*} in Eqn. (1823) is defined as follows: $$f_{\beta^*} = \begin{cases} 1 & \text{for } \chi_k \leq 0 \\ \dfrac{1 + 680\chi_k^2}{1 + 400\chi_k^2} & \text{for } \chi_k > 0 \end{cases} \quad (1840)$$ where: $$\chi_k = \frac{\nabla k \cdot \nabla\omega}{\omega^3}$$
Cross-diffusion Limiter	The following term is added to the specific dissipation rate equation Eqn. (1824) as an additional source term: $$CD_\omega = \sigma_d \frac{\rho}{\omega}\frac{\partial k}{\partial y}\frac{\partial \omega}{\partial y} \quad (1841)$$ where: $$\sigma_d = \begin{cases} 0, \dfrac{\partial k}{\partial y}\dfrac{\partial \omega}{\partial y} & \leq 0 \\ \dfrac{1}{8}, \dfrac{\partial k}{\partial y}\dfrac{\partial \omega}{\partial y} & > 0 \end{cases}$$ Additionally, the function f_{β^*} is set to unity.
None	f_{β^*} is set to unity.

Vortex Stretching Modification

Almost all two-equation models tend to predict a larger spreading rate for round jets than for plane jets, while the opposite effect is observed in experiments. The vortex-stretching modification is designed to overcome this round-jet/plane-jet anomaly [339]. In STAR-CCM+, this modification is implemented for the Standard K-Omega model and is defined as follows:

Vortex-Stretching Modification	Formulation
Activated	f_β is defined as follows: $$f_\beta = \frac{1 + 70\chi_\omega}{1 + 80\chi_\omega} \quad (1842)$$ where: $$\chi_\omega = \frac{(\mathbf{W} \cdot \mathbf{W}):\mathbf{S}}{(\beta^* \omega)^3}$$ and: • $\mathbf{S}$ and $\mathbf{W}$ are given by Eqn. (1745) and Eqn. (1747), respectively. • β^* is a Model Coefficient. If the cross-diffusion limiter is activated, f_β is defined as: $$f_\beta = \frac{1 + 85\chi_\omega}{1 + 100\chi_\omega} \quad (1843)$$
Deactivated	f_β is set to unity.

The vortex-stretching modification is not available for the SST K-Omega model—f_β is set to unity.

Low Reynolds Number Modification

The low Reynolds number modification can be used to account for low Reynolds number and transitional effects.

If the low Reynolds number modification is activated, the following K-Omega Model Coefficients are replaced:

Model Variant	Model Coefficients	Replaced By:
SKO	α	$$\frac{\alpha}{\alpha^*} \cdot \frac{1/9 + Re_t/Re_\omega}{1 + Re_t/Re_\omega} \quad (1844)$$
	α^*	$$\frac{\beta/3 + Re_t/Re_k}{1 + Re_t/Re_k} \quad (1845)$$

Model Variant	Model Coefficients	Replaced By:
	β^*	$\beta^* \cdot \frac{4/15 + \left(Re_t/Re_\beta\right)^4}{1 + \left(Re_t/Re_\beta\right)^4}$ (1846)
SSTKO	α_1^*	$\frac{\beta/3 + Re_t/Re_k}{1 + Re_t/Re_k}$ (1847)
	β_1^*	$\beta_2^* \cdot \frac{4/15 + \left(Re_t/Re_\beta\right)^4}{1 + \left(Re_t/Re_\beta\right)^4}$ (1848)
	γ_1	$\gamma_1 \cdot \frac{1/9 + Re_t/Re_\omega}{1 + Re_t/Re_\omega} \cdot \frac{1}{\alpha^*}$ (1849)

where:

- Re_t is given by Eqn. (1751).
- Re_ω, Re_k, and Re_β are Model Coefficients.

Model Coefficients

Re_ω	Re_k	Re_β
2.95	6	8

Constitutive Relations

Constitutive relations describe the relation between the Reynolds stress tensor and the mean strain rate that is used in the Boussinesq approximation.

By default, the Boussinesq approximation implies a linear constitutive relation. Non-linear constitutive relations account for anisotropy of turbulence by adding non-linear functions of the strain and vorticity tensors. For the SST (Menter) K-Omega model in STAR-CCM+, quadratic and cubic constitutive relations are available. The cubic constitutive relation is derived from a Reynolds stress model and thus represents an Explicit Algebraic Reynolds Stress Model (EARSM):

Constitutive Relation	Formulation
Quadratic [336]	$\mathbf{T}_{t,NL} = -2\mu_t 0.04645(\mathbf{O} \cdot \mathbf{S} - \mathbf{S} \cdot \mathbf{O})$ (1850)

Constitutive Relation	Formulation
Cubic [333], [338]	$$\mathbf{T}_{t,NL} = -\rho k\left[\beta_3\left(\mathbf{W}^* \cdot \mathbf{W}^* - \frac{1}{3}II_\Omega \mathbf{I}\right) + \beta_4\left(\mathbf{S}^* \cdot \mathbf{W}^* - \mathbf{W}^* \cdot \mathbf{S}^*\right) + \beta_6\left(\mathbf{S}^* \cdot \mathbf{W}^* \cdot \mathbf{W}^* + \mathbf{W}^* \cdot \mathbf{W}^* \cdot \mathbf{S}^* - II_\Omega \mathbf{S}^* - \frac{2}{3}IV\mathbf{I}\right)\right] \quad (1851)$$

where:

- ρ is the density.
- $\mathbf{I}$ is the identity tensor.
- β_3, β_4, and β_6 are Model Coefficients.

$\mathbf{S}$ is the strain rate tensor given by Eqn. (1745) and $\mathbf{S}^*$ is given by:

$$\mathbf{S}^* = \frac{1}{\beta^* \omega}\mathbf{S} \tag{1852}$$

where β^* is given in K-Omega Model—Model Coefficients.

$\mathbf{W}$ is the vorticity tensor given by Eqn. (1747), $\mathbf{W}^*$ is given by:

$$\mathbf{W}^* = \frac{1}{\beta^* \omega}\mathbf{W} \tag{1853}$$

and:

$$\mathbf{O} = \frac{\mathbf{W}}{\sqrt{(\mathbf{S} - \mathbf{W})(\mathbf{S} - \mathbf{W})}} \tag{1854}$$

$$II_\Omega = \mathbf{W}^* : \mathbf{W}^{*\mathrm{T}} \tag{1855}$$

$$IV = \mathbf{S}^* \cdot \mathrm{W}^* \cdot \mathrm{W}^* \tag{1856}$$

Relation for Turbulent Viscosity

When applying the linear or the quadratic constitutive relation, the turbulent viscosity for the SST (Menter) K-Omega model is given by Eqn. (1817). For the cubic constitutive relation, μ_t is computed as:

$$\mu_t = \frac{C_\mu}{\beta^*}\frac{\rho k}{\omega} \tag{1857}$$

The coefficient C_μ is obtained from:

$$C_\mu = -\frac{1}{2}\left(\beta_1 + II_\Omega \beta_6\right) \tag{1858}$$

where β_1 and β_6 are Model Coefficients.

Model Coefficients

The coefficients β_i are given by:

$$\beta_1 = -\frac{N\left(2N^2 - 7II_\Omega\right)}{Q} \tag{1859}$$

$$\beta_3 = -\frac{12IV}{NQ} \tag{1860}$$

$$\beta_4 = -\frac{2\left(N^2 - 2II_\Omega\right)}{Q} \tag{1861}$$

$$\beta_6 = -\frac{6N}{Q} \tag{1862}$$

$$Q = \frac{5}{6}\left(N^2 - 2II_\Omega\right)\left(2N^2 - II_\Omega\right) \tag{1863}$$

where N is given by:

$$N = \begin{cases} \frac{A_3'}{3} + \sqrt[3]{P_1 + \sqrt{P_2}} + sign\left(P_1 - \sqrt{P_2}\right)\sqrt[3]{\left|P_1 - \sqrt{P_2}\right|} & \text{for } P_2 \geq 0 \\ \frac{A_3'}{3} + 2\sqrt[6]{P_1^2 - P_2}\cos\left[\frac{1}{3}\cos^{-1}\left(\frac{P_1}{\sqrt{P_1^2 - P_2}}\right)\right] & \text{for } P_2 < 0 \end{cases} \tag{1864}$$

where:

$$P_1 = A_3'\left(\frac{{A_3'}^2}{27} + \frac{9}{20}II_S - \frac{2}{3}II_\Omega\right) \tag{1865}$$

$$P_2 = P_1^2 - \left(\frac{{A_3'}^2}{9} + \frac{9}{10}II_S - \frac{2}{3}II_\Omega\right)^3 \tag{1866}$$

$$A_3' = \frac{9}{5} + \frac{9}{4}2.2\max\left(1 + \beta_1^{eq}II_S, 0\right) \tag{1867}$$

$$\beta_1^{eq} = -\frac{6}{5}\left(\frac{\frac{81}{20}}{\left(\frac{81}{20}\right)^2 - 2II_\Omega}\right) \tag{1868}$$

$$II_S = \mathbf{S}^*\mathbf{:}\mathbf{S}^{*\mathrm{T}} \tag{1869}$$

Compressibility Correction

Most turbulence models are derived based on incompressible flow assumptions. However, at high Mach number density variations affect the structures of turbulent flow and corrections must be introduced.

The turbulence dissipation must be modified to account for dilatation-induced dissipation. In STAR-CCM+, the compressibility correction describes this dilatation-dissipation as a function of the turbulent Mach number M_t [335].

If compressibility correction is enabled, the coefficients β and β^* are replaced by:

$$\beta_{\mathrm{comp}} = \beta - \beta^*\xi^*F\left(M_t\right) \tag{1870}$$

$$\beta^*_{\text{comp}} = \beta^*\left[1 + \xi^* F\left(M_t\right)\right] \tag{1871}$$

where:

- β^* is either given by the constant K-Omega Model Coefficient or, if the low Reynolds number correction is used, by the Replaced K-Omega Model Coefficient.
- ξ^* is a Model Coefficient.

$F\left(M_t\right)$ is the compressibility function defined as:

$$F\left(M_t\right) = \max\left[0, M_t^2 - \left(\frac{1}{4}\right)^2\right] \tag{1872}$$

where:

- ${M_t}^2 = \frac{2k}{c^2}$, where c is the speed of sound.

Model Coefficient

ξ^*
1.5

Elliptic Blending Model

The Elliptic Blending turbulence model solves transport equations for the turbulent kinetic energy k, the turbulent dissipation rate ε, the normalized (reduced) wall-normal stress component $\varphi = \overline{\vartheta^2}/k$, and the elliptic blending factor α in order to determine the turbulent eddy viscosity.

The concept of elliptic relaxation was proposed by Durbin [342] for Reynolds-stress models. The initial model required the solution of six additional transport equations, but this number was later reduced to a single additional equation. The model was later simplified by Manceau and Hanjalić [345] to make it more industry-friendly.

Model Variants

Two variants of the Elliptic Blending model are implemented in STAR-CCM+:

Model Variant	Abbreviation
Standard Elliptic Blending	EBS
Lag Elliptic Blending	EBL

Standard Elliptic Blending

The elliptic relaxation model led to the development of some two-equation eddy viscosity models, starting with the $\overline{\vartheta^2} - f$ model. One of the most robust models, according to its authors, is the $BL{-}\overline{\vartheta^2}/k$ model of Billard and Laurence [341]. Important modifications to this model make it truly robust for complex flow geometries.

The benefits of using the Billard - Laurence model are:

- An improvement on the existing realizable $k-\varepsilon$ model in terms of accuracy, especially in the near-wall region.
- An improvement in terms of stability over the SST $k-\omega$ model.

Lag Elliptic Blending

The Lag Elliptic Blending model combines the Standard Elliptic Blending model with the stress-strain lag concept originally proposed by Revell et al. [347].

In flow regions where non-equilibrium effects result in a misalignment of the principal components of the stress and strain-rate tensors, linear eddy viscosity models tend to over-predict the production of k. To overcome this effect, the Lag Elliptic Blending model incorporates the angle between these components. Additional terms model the effects of anisotropy—similar to non-linear constitutive relations—and curvatures and rotational effects—similar to curvature correction. These terms are directly embedded in the transport equation for the reduced stress function φ [346].

The Lag Elliptic Blending model provides a good predictive capability for separated or unsteady flow (such as vortex shedding), or when the flow is subject to rotation or strong streamline curvature.

Relation for Turbulent Viscosity

The turbulent eddy viscosity μ_t is calculated as:

$$\mu_t = \rho C_\mu \varphi k \min\left(T, \frac{C_T}{\sqrt{3} C_\mu \varphi S}\right) \tag{1873}$$

where:

- ρ is the density.
- C_μ and C_T are Model Coefficients.
- T is the turbulent time scale.
- S is given by Eqn. (1744).

The turbulent time scale is calculated as:

$$T = \sqrt{T_e^2 + C_t^2 \frac{\nu}{\varepsilon}} \tag{1874}$$

where:

- $T_e = \frac{k}{\varepsilon}$ is the large-eddy time scale.
- C_t is a Model Coefficient.
- ν is the kinematic viscosity.

Transport Equations

The transport equations for the four variables k, ε, φ, and α are:

$$\frac{\partial}{\partial t}(\rho k) + \nabla\cdot(\rho k \bar{\mathbf{v}}) = \nabla\cdot\left[\left(\frac{\mu}{2} + \frac{\mu_t}{\sigma_k}\right)\nabla k\right] + P_k - \rho(\varepsilon - \varepsilon_0) + S_k \tag{1875}$$

$$\frac{\partial}{\partial t}(\rho\varepsilon) + \nabla\cdot(\rho\varepsilon\overline{\mathbf{v}}) = \nabla\cdot\left[\left(\frac{\mu}{2}+\frac{\mu_t}{\sigma_\varepsilon}\right)\nabla\varepsilon\right] + \frac{1}{T_e}C_{\varepsilon1}P_\varepsilon - C^*_{\varepsilon2}\left(\frac{\varepsilon}{T_e}-\frac{\varepsilon_0}{T_0}\right) + S_\varepsilon \tag{1876}$$

$$\frac{\partial}{\partial t}(\rho\varphi) + \nabla\cdot(\rho\varphi\overline{\mathbf{v}}) = \nabla\cdot\left[\left(\frac{\mu}{2}+\frac{\mu_t}{\sigma_\varphi}\right)\nabla\varphi\right] + P_\varphi + S_\varphi \tag{1877}$$

$$\nabla\cdot\left(L^2\nabla\alpha\right) = \alpha - 1 \tag{1878}$$

where:

- $\overline{\mathbf{v}}$ is the mean velocity.
- μ is the dynamic viscosity.
- P_k, P_ε, and P_φ are Production Terms.
- $C_{\varepsilon1}$, $C^*_{\varepsilon2}$, σ_k, σ_ε, and σ_φ are Model Coefficients.
- S_k, S_ε, and S_φ are the user-specified source terms.

L is the turbulent length scale calculated as:

$$L = C_L\sqrt{\frac{k^3}{\varepsilon^2} + C_\eta{}^2\sqrt{\frac{\nu^3}{\varepsilon}}} \tag{1879}$$

where C_L and C_η are Model Coefficients.

ε_0 is the ambient turbulence value in the source terms that counteracts turbulence decay [327]. The possibility to impose an ambient source term also leads to the definition of a specific time-scale T_0 that is defined as:

$$T_0 = \max\left(\frac{k_0}{\varepsilon_0}, C_t\sqrt{\frac{\nu}{\varepsilon_0}}\right) \tag{1880}$$

Production Terms

The formulation of the production terms P_k, P_ε, and P_φ depends on the Elliptic Blending model variant:

Model Variant	P_k	P_ε
EBS	$G_k + G_b + \Upsilon_M$ (1881)	$G_k + C_{\varepsilon3}G_b + \frac{1}{C_{\varepsilon1}}E$ (1882)
EBL		

Model Variant	P_φ
EBS	$-\frac{\varphi}{k}(G_k + G_b) + \rho(1-\alpha^3)f_w + \rho\alpha^3 f_h$ (1883) where: $f_w = -\frac{1}{2}\frac{\varphi}{T_e}$

Model Variant	P_φ
	$f_h = -\frac{1}{T}\left(C_1 - 1 + C_2\frac{G_k + G_b}{\rho\varepsilon}\right)\left(\varphi - \frac{2}{3}\right)$
EBL	$$-(2 - C_{\varepsilon 1})\frac{\varphi}{k}(G_k + G_b) + \rho(1 - \alpha^3)f_w + \rho\alpha^3 f_h$$ (1884) where: $f_w = -\left(C_{\varepsilon 2} - 1 + 5 - \frac{1}{C_\mu}\right)\frac{\varphi}{T_e}$ $f_h = -\frac{1}{T_e}\left(C_1 + C_{\varepsilon 2} - 2 + C_1^*\frac{G_k + G_b}{\rho\varepsilon}\right)\varphi + \frac{C_{P3}}{T_e} + \frac{C_3^*}{\sqrt{2}}\varphi S + \frac{1}{S^2 T_e}\left[\frac{2}{C_\mu}(1 - C_4)\mathbf{AS} - \frac{2}{C_\mu}(1 - C_5)\mathbf{A}\widehat{\mathbf{W}}\right]:\mathbf{S}$ $\mathbf{A}$ is the Reynolds-stress anisotropy tensor that is defined as: $\mathbf{A} = -2\frac{\mu_t}{\rho k}\left[\mathbf{S} + \frac{2 - 2C_5}{C_1 + C_1^* + 1}\frac{2}{\sqrt{(\mathbf{S} + \widehat{\mathbf{W}}):(\mathbf{S} + \widehat{\mathbf{W}})}}\left(\mathbf{S}\widehat{\mathbf{W}} - \widehat{\mathbf{W}}\mathbf{S}\right)\right]$ where $\mathbf{S}$ is given by Eqn. (1745). $\widehat{\mathbf{W}}$ is the modified absolute vorticity tensor given by: $\widehat{\mathbf{W}} = \widetilde{\mathbf{W}} - \mathbf{W}^{\mathbf{s}}$ where $\widetilde{\mathbf{W}}$ is given by Eqn. (1748) and $\mathbf{W}^{\mathbf{s}}$ is the Spalart-Shur tensor given by: $\mathbf{W}^{\mathbf{s}} = \frac{1}{S^2}\left(\mathbf{S}\frac{D\mathbf{S}}{Dt} - \frac{D\mathbf{S}}{Dt}\mathbf{S}\right)$

where:

- C_1, C_1^*, C_2, C_3^*, C_4, C_5, $C_{\varepsilon 2}$, $C_{\varepsilon 3}$, C_μ, and C_{P3} are Model Coefficients.
- the contributions to the production terms are:

	Description	Formulation	Where:
G_k	Turbulent production	$\mu_t S^2 - \frac{2}{3}\rho k\,\nabla\cdot\overline{\mathbf{v}} - \frac{2}{3}\mu_t\left(\nabla\cdot\overline{\mathbf{v}}\right)^2$ (1885)	-
G_b	Buoyancy production	$\beta\frac{\mu_t}{Pr_t}\left(\nabla\overline{T}\cdot\mathbf{g}\right)$ (1886)	• β is the coefficient of thermal expansion. For constant density flows using the Boussinesq approximation, β is user-specified. For ideal gases, β is given by $\beta = -\frac{1}{\rho}\frac{\partial\rho}{\partial\overline{T}}$. • Pr_t is the turbulent Prandtl number.

	Description	Formulation	Where:
			• $\overline{T}$ is the mean temperature. • $\mathbf{g}$ is the gravitational vector.
Υ_M	Compressibility modification (Sarkar et al. [325])	$\frac{C_M\, k\, \varepsilon}{c^2}$ (1887)	• C_M is a Model Coefficient. • c is the speed of sound.
E	Additional production	$C_k(1-\alpha)^3 \nu \mu_t \frac{k}{\varepsilon}\left[\nabla\cdot\left(\lVert 2\mathbf{Sn}\rVert\mathbf{n}\right)\right]^2$ (1888)	• C_k is a Model Coefficient. • $\mathbf{n}$ represents the wall normal direction.

Model Coefficients

Coefficient	EBS	EBL
C_1	1.7	1.7
C_1^*	-	0.9
C_2	0.9	-
C_3	-	0.8
C_3^*	-	0.65
C_4	-	0.625
C_5	-	0.2
$C_{\varepsilon 1}$	1.44	1.44
$C_{\varepsilon 2}$	1.83	1.9
$C_{\varepsilon 2}^*$	See Free-Stream Option.	$C_{\varepsilon 2}$
$C_{\varepsilon 3}$	See K-Epsilon Model—Model Coefficients.	
C_k	2.3	2.3
C_L	0.164	0.164
C_M (Sarkar)	2	2
C_μ	0.22	0.22
C_η	75	75

Coefficient	EBS	EBL
C_{P3}	-	See Damping Function.
C_T	1	1
C_t	4	4
σ_ε	1.5	1.2
σ_φ	1	1
σ_k	1	1

Free-Stream Option

One important difference between the Standard Elliptic Blending model and other two-equation models is the definition of the coefficient of the destruction term $C^*_{\varepsilon 2}$. This term is defined as a function of the turbulent kinetic energy gradient, to reduce the $C_{\varepsilon 2}$ value in the defect layer, where the free-stream is starting to dominate. However, for some external aerodynamic cases, this formulation can cause instabilities in the free-stream. For such cases, a constant $C_{\varepsilon 2}$ value is recommended:

Free-stream Option	$C^*_{\varepsilon 2}$
Variable C2e Option	$C_{\varepsilon 2} + \alpha^3 (1 - C_{\varepsilon 2}) \tanh\left[\left\lvert \frac{\partial}{\partial x_j}\left(\frac{\mu_t}{\rho \sigma_k}\frac{\partial k}{\partial x_j}\right)\varepsilon^{-1}\right\rvert^{3/2}\right]$ (1889)
Off	$C_{\varepsilon 2}$ (1890)

Damping Function

For turbulence models that resolve the viscous- and buffer-layer, damping functions mimic the decrease of turbulent mixing near the walls. For the Lag Elliptic Blending model, the kinematic blocking of the wall is implemented within the Model Coefficient C_{P3} as:

$$C_{P3} = \frac{f_\mu}{C_\mu}\left(\frac{2}{3} - \frac{C_3}{2}\right) \tag{1891}$$

where C_3 is a Model Coefficient and the damping function f_μ is defined as:

$$f_\mu = \frac{\eta + \alpha^3}{\max(\eta, 1.87)} \tag{1892}$$

and time-scale ratio η is calculated as:

$$\eta = ST_e \tag{1893}$$

Changes From Base Model

The main changes from the original model proposed by Billard and Laurence [341] are:

- Removal of an additional cross-diffusion term in the φ transport equation, for stability reasons.
- The constant in the definition of $C^*_{\varepsilon 2}$, which acts on its decay in the defect layer, has been increased from 0.4 to 1.
- The extra-dissipation term E, which accounts for viscous wall-effects, was moved from the k-equation back to the ε-equation. This change is for stability reasons and to avoid unwanted relaminarization effects in some scenarios.

V2F Model

The V2F turbulence model solves transport equations for two more turbulence quantities in addition to k and ε, namely the wall-normal stress component $\overline{\vartheta^2} = \overline{\mathrm{v}'\mathrm{v}'}$ and the elliptic relaxation parameter f, in order to determine the turbulent eddy viscosity.

The V2F model ([348], [350], [349]) is designed to handle wall effects in turbulent boundary layers and to accommodate non-local effects. It is known to capture the near-wall turbulence effects more accurately, which is crucial for the accurate prediction of heat transfer, skin friction and flow separation.

Relation for Turbulent Viscosity

The turbulent eddy viscosity μ_t is calculated as:

$$\mu_t = \rho \ \min\left(C_\mu kT, C_{\mu_{\vartheta^2}} \vartheta^2 T_s\right) \tag{1894}$$

where:

- ρ is the density.
- C_μ and $C_{\mu_{\vartheta^2}}$ are Model Coefficients.
- T and T_s are turbulent time scale and limited time scale, respectively.

The time scales are calculated using:

T	T_s	T with Realizable Scale Option
$\max\left(T_e, C_t\sqrt{\frac{\nu}{\varepsilon}}\right)$ (1895)		$\max\left(\min\left(T_e, \frac{0.6k}{\sqrt{3}C_{\mu_{\vartheta^2}}\overline{\vartheta^2}S}\right), C_t\sqrt{\frac{\nu}{\varepsilon}}\right)$ (1896)

where:

- $T_e = \frac{k}{\varepsilon}$ is the large-eddy time scale.
- C_t is a Model Coefficient.

- ν is the kinematic viscosity.
- S is given by Eqn. (1744).

Transport Equations

The transport equations for the four variables k, ε, $\overline{\vartheta^2}$, and f are:

$$\frac{\partial}{\partial t}(\rho k) + \nabla\cdot(\rho k\overline{\mathbf{v}}) = \nabla\cdot\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\nabla k\right] + P_k - \rho\left(\left(\varepsilon - \varepsilon_0\right)\right) + S_k \tag{1897}$$

$$\frac{\partial}{\partial t}(\rho\varepsilon) + \nabla\cdot(\rho\varepsilon\overline{\mathbf{v}}) = \nabla\cdot\left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\nabla\varepsilon\right] + \frac{1}{T_e}C_{\varepsilon_1}P_\varepsilon - C_{\varepsilon_2}\rho\left(\frac{\varepsilon}{T_e} - \frac{\varepsilon_0}{T_0}\right) + S_\varepsilon \tag{1898}$$

$$\frac{\partial}{\partial t}\left(\rho\overline{\vartheta^2}\right) + \nabla\cdot\left(\rho\overline{\vartheta^2}\overline{\mathbf{v}}\right) = \nabla\cdot\left[\left(\mu + \frac{\mu_t}{\sigma_{\overline{\vartheta^2}}}\right)\nabla\overline{\vartheta^2}\right] + P_{\overline{\vartheta^2}} - \frac{6\rho\overline{\vartheta^2}\varepsilon}{k} \tag{1899}$$

$$\nabla\cdot\left(L^2\nabla f\right) - f + P_f = 0 \tag{1900}$$

where:

- $\overline{\mathbf{v}}$ is the mean velocity.
- μ is the dynamic viscosity.
- P_k, P_ε, $P_{\overline{\vartheta^2}}$, and P_f are Production Terms.
- $C_{\varepsilon 1}$, $C_{\varepsilon 2}$, σ_k, σ_ε, and $\sigma_{\overline{\vartheta^2}}$ are Model Coefficients.
- S_k and S_ε are the user-specified source terms.

ε_0 is the ambient turbulence value in the source terms that counteracts turbulence decay [327]. The possibility to impose an ambient source term also leads to the definition of a specific time-scale T_0 that is defined as:

$$T_0 = \max\left(\frac{k_0}{\varepsilon_0}, C_t\sqrt{\frac{\nu}{\varepsilon_0}}\right) \tag{1901}$$

The turbulent length scale L is calculated as:

L	L with Realizable Scale Option
$C_L \max\left(\frac{k^{3/2}}{\varepsilon}, C_\eta\left(\frac{\nu^3}{\varepsilon}\right)^{1/4}\right)$ (1902)	$C_L \max\left(\min\left(\frac{k^{3/2}}{\varepsilon}, \frac{k^{3/2}}{\sqrt{3}C_{\mu_{\vartheta^2}}\overline{\vartheta^2}S}\right), C_\eta\left(\frac{\nu^3}{\varepsilon}\right)^{1/4}\right)$ (1903)

where:

- C_L and C_η are Model Coefficients.

Production Terms

The production terms P_k, P_ε, $P_{\overline{\vartheta^2}}$, and P_f are calculated as:

P_k	P_ε
$G_k + G_b + \Upsilon_M$ (1904)	$\left(1 + A\sqrt{\frac{k}{\overline{\vartheta^2}}}\right)G_k + C_{\varepsilon 3}G_b$ (1905)

$P_{\overline{\vartheta^2}}$	P_f
$\rho\min\left\{kf, -\frac{1}{T}\left[\left(C_1 - 6\right)\overline{\vartheta^2} - \frac{2k}{3}\left(C_1 - 1\right)\right] + C_2 G_k\right\}$ (1906)	$\frac{1}{T_s}\left(C_1 - 1\right)\left(\frac{2}{3} - \frac{\overline{\vartheta^2}}{k}\right) + C_2\frac{C_\mu \overline{\vartheta^2} S^2}{\varepsilon} + \frac{5\overline{\vartheta^2}}{kT_s}$ (1907)

where:

- A, $C_{\varepsilon 3}$, C_1, and C_2, are Model Coefficients.
- the contributions to the production terms are:

	Description	Formulation	Where:
G_k	Turbulent production	$\mu_t S^2 - \frac{2}{3}\rho k \nabla\mathbf{v} - \frac{2}{3}\mu_t\left(\nabla\mathbf{v}\right)^2$ (1908)	-
G_b	Buoyancy production	$\beta\frac{\mu_t}{Pr_t}\left(\nabla\overline{T}\cdot\mathbf{g}\right)$ (1909)	• β is the coefficient of thermal expansion. For constant density flows using the Boussinesq approximation, you specify β. For ideal gases, the following relation is used: $\beta = -\frac{1}{\rho}\frac{\partial\rho}{\partial\overline{T}}$ • Pr_t is the turbulent Prandtl number. • $\overline{T}$ is the mean temperature. • $\mathbf{g}$ is the gravitational vector.
Υ_M	Compressibility modification (Sarkar et al. [325])	$\frac{C_M k \varepsilon}{c^2}$ (1910)	• C_M is a Model Coefficient. • c is the speed of sound.

Model Coefficients

C_1	C_2	$C_{\varepsilon 1}$	$C_{\varepsilon 2}$	$C_{\varepsilon 3}$	C_μ	$C_{\mu_{\vartheta^2}}$	C_L	C_M (Sarkar)	C_t	C_η
1.4	0.3	1.4	1.9	See K-Epsilon Model—Model Coefficients.	0.09	0.22	0.23	2	6	70

σ_k	σ_ε	$\sigma_{\overline{\vartheta^2}}$
1	1.3	1

A
0.045

Curvature Correction

Rotation and by association streamline curvature have a strong influence on the evolution of turbulent quantities. Such sensitivity is absent in one- and two-equation eddy viscosity models, for pure derivation equations. Different methods have been proposed to account for this deficiency. Three different curvature correction (CC) formulations are used in STAR-CCM+, depending on the turbulence model being considered.

CC for Spalart-Allmaras models

Shur and others [313] proposed a correction term for system-rotation and streamline curvature effects. The model acts directly on the eddy viscosity by including a rotation function f_{r1} in the production term $P_{\tilde{\nu}}$:

$$f_{r1} = \left(1 + c_{r1}\right) \frac{2r^*}{1 + r^*}\left[1 - c_{r3}\tan^{-1}\left(c_{r2}\tilde{r}\right)\right] - c_{r1} \tag{1911}$$

where:

- c_{r1}, c_{r2}, and c_{r3} are Model Coefficients.
- $r^* = |\mathbf{S}|/|\widetilde{\mathbf{W}}|$
- $r = 2\left(\widetilde{\mathbf{W}}\mathbf{S}\right):\left(\frac{D\mathbf{S}}{Dt}\right)\frac{1}{D^4}$
- $D^2 = 0.5\left(|\mathbf{S}|^2 + |\widetilde{\mathbf{W}}|^2\right)$

$\mathbf{S}$ and $\widetilde{\mathbf{W}}$ are given by Eqn. (1745) and Eqn. (1748), respectively.

Model Coefficients

c_{r1}	c_{r2}	c_{r3}
1	12	1

CC for K-Epsilon and K-Omega models

The transport equation for the turbulent kinetic energy is insensitive, by construction, to stabilizing and destabilizing effects usually associated with strong (streamline) curvature and frame-rotation. These effects can be incorporated by using a curvature correction factor, which alters the turbulent kinetic energy production term according to the local rotation and vorticity rates.

The curvature correction factor f_c [331] is calculated as:

$$f_c = \min\left(C_{\max}, \frac{1}{C_{r_1}(|\eta|-\eta) + \sqrt{1-\min\left(C_{r_2}, 0.99\right)}}\right) \tag{1912}$$

where:

- $C_{\max}$, C_{r_1}, and C_{r_2} are Model Coefficients.
- $\eta = T^2(\mathbf{S}:\mathbf{S} - \mathbf{W}:\mathbf{W})$

The time-scale T is limited in order to have the correct near-wall asymptotic behavior:

$$T = \max\left(T_1, T_3\right)$$

where:

Model	T_1	T_2	T_3
K-Epsilon models	$\frac{\varepsilon}{k}$	$6\sqrt{\frac{\nu}{\varepsilon}}$	$\left(T_1^{1.625} T_2\right)^{\frac{1}{1.625+1}}$
K-Omega models	$\frac{1}{\beta^* \omega}$	$6\sqrt{\frac{\nu}{\beta^* k \omega}}$	

The strain-rate tensor $\mathbf{S}$ is as defined in Eqn. (1745), and the absolute rotation-rate tensor $\mathbf{W}$ is calculated as:

$$\mathbf{W} = \mathbf{W}^l + \mathbf{W}^f + \left(C_{\mathrm{ct}} - 1\right)\mathbf{W}^S \tag{1913}$$

where C_{ct} is a Model Coefficient and:

$$\mathbf{W}^l = \frac{1}{2}\left(\nabla\overline{\mathbf{v}} - \nabla\overline{\mathbf{v}}^T\right) \tag{1914}$$

$$\mathbf{W}^f = \mathbf{E} \cdot \boldsymbol{\omega}^f \tag{1915}$$

$$\mathbf{W}^S = \mathbf{W}^f - \mathbf{E} \cdot \left(\mathbf{A}^{-1} \cdot \boldsymbol{\omega}\right) \tag{1916}$$

$\mathbf{W}^l$, $\mathbf{W}^f$, and $\mathbf{W}^S$ are the contributions due to the vorticity computed in the local frame of reference, the rotating frame of reference (defined by the frame rotation vector $\boldsymbol{\omega}^f$), and the streamline curvature, respectively.

$\mathbf{E}$ is the Levi-Civita tensor (also know as the three-dimensional permutation tensor).

For the last contribution, two additional terms are needed:

$$\mathbf{A} = \mathbf{I} - \frac{3\mathbf{S}^2}{2\mathbf{S}:\mathbf{S}} \quad \text{and} \quad \boldsymbol{\omega} = \mathbf{E} \cdot \frac{\mathbf{S} \cdot (D_t\mathbf{S}) - (D_t\mathbf{S}) \cdot \mathbf{S}}{2\mathbf{S}:\mathbf{S}} \tag{1917}$$

where $D_t\mathbf{S}$ is the strain-rate total derivative tensor. For more details on the derivation of this method, see [331].

If curvature correction is not activated, f_c is set to unity.

Model Coefficients

c_{r1}	c_{r2}	C_{ct}	$C_{\max}$ (Upper Limit)
0.04645	0.25	2	1.25

Turbulent Viscosity User Scaling

The **Turbulent Viscosity User Scaling** model allows you to scale the turbulent viscosity for both K-Epsilon and K-Omega turbulence models.

When the model is activated, the value of the turbulent viscosity μ_t is multiplied with a custom scaling factor a:

$$\mu_{t,scaled} = a\mu_t \tag{1918}$$

Reynolds Stress Transport (RST) Models

Reynolds Stress Transport (RST) models, also known as second-moment closure models, directly calculate the components of the specific Reynolds stress tensor $\mathbf{R} = -\mathbf{T}_t/\rho$ by solving their governing transport equations.

RST models have the potential to predict complex flows more accurately than eddy viscosity models because the transport equations for the Reynolds stresses naturally account for the effects of turbulence anisotropy, streamline curvature, swirl rotation and high strain rates.

The starting point for the development of an RST model is generally the exact differential transport equation for the Reynolds stresses, which is derived by multiplying the instantaneous Navier-Stokes equations by a fluctuating property and Reynolds-averaging their product (see [357] for this exact equation). In the resulting equations, the transient, convective and molecular diffusion terms do not require modeling. The terms remaining to be modeled are the turbulent diffusion term, the dissipation term and, perhaps the greatest challenge, the pressure-strain term. Appropriate models for these terms have received much attention during the past few decades, some of which are reviewed in [358].

To model the pressure-strain term, the following approaches are implemented in STAR-CCM+:

- Linear Pressure-Strain model
- Quadratic Pressure-Strain model

The Linear Pressure-Strain model can be used with a high-y^+ wall treatment, but it is also available with a two-layer formulation, which makes it applicable right down to the wall.

The Quadratic Pressure-Strain model and its low-Reynolds number variant, the Elliptic Blending model, are based on the most recent and precise formulation of the pressure-strain term. Thus, they are likely to be the

models of choice in a majority of cases, the Elliptic Blending model being preferred as it also contains specific treatments for all-y^+ meshes. The Quadratic Pressure-Strain model can only be used with a high-y^+ wall treatment (that is, using wall functions) without resolving the viscous-affected near-wall region.

Contents:

Reynolds Stress Transport Equation
Linear Pressure-Strain Model
Quadratic Pressure-Strain Model
Elliptic Blending Model

Reynolds Stress Transport Equation

The transport equation for the specific Reynolds stress tensor $\mathbf{R}$ is:

$$\frac{\partial}{\partial t}(\rho\mathbf{R}) + \nabla\cdot(\rho\mathbf{R}\overline{\mathbf{v}}) = \nabla\cdot\mathbf{D} + \mathbf{P} + \mathbf{G} - \frac{2}{3}\rho\mathbf{I}\Upsilon_M + \underline{\phi} + \underline{\varepsilon} + \mathbf{S_R} \tag{1919}$$

where:

- ρ is the density.
- $\overline{\mathbf{v}}$ is the mean velocity.
- $\mathbf{D}$ is the Reynolds Stress Diffusion.
- $\mathbf{P}$ is the Turbulent Production.
- $\mathbf{G}$ is the Buoyancy Production.
- $\mathbf{I}$ is the identity tensor.
- γ_M is the Dilatation Dissipation.
- $\underline{\phi}$ is the pressure strain tensor.
- $\underline{\varepsilon}$ is the turbulent dissipation rate tensor.
- $\mathbf{S_R}$ is the user-specified source.

For the Linear Pressure Strain and Quadratic Pressure Strain models, the dissipation is simply:

$$\underline{\varepsilon} = \frac{2}{3}\varepsilon\mathbf{I} \tag{1920}$$

Seven equations must be solved (as opposed to the two equations of a K-Epsilon or a K-Omega model): six equations for the Reynolds stresses (symmetric tensor) and one equation for the isotropic turbulent dissipation ε (see Eqn. (1776) (SKE)).

Reynolds Stress Diffusion

Two different models for the Reynolds-stress diffusion are available. By default, following [356], a simple isotropic form of the turbulent diffusion is adopted, such that:

$$\mathbf{D} = \left(\mu + \frac{\mu_t}{\sigma_k}\right)\nabla\mathbf{R} \tag{1921}$$

where:

- μ is the dynamic viscosity.
- σ_k is a Model Coefficient.

The turbulent viscosity $\mu_t$is computed as:

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon} \tag{1922}$$

where C_μ is a Model Coefficient. The turbulent kinetic energy k is defined as:

$$k = \frac{1}{2}\mathrm{tr}\left(\mathbf{R}\right) \tag{1923}$$

where $\mathrm{tr}(\mathbf{R})$ is the trace of the Reynolds stress tensor (see Eqn. (4155)).

Alternatively, the Reynolds-stress diffusion can be written according to Daly and Harlow [351] in the form:

$$\mathbf{D} = \mu \nabla \mathbf{R} + C_s \frac{k}{\varepsilon}\left(\mathbf{R} \cdot \nabla\right)\mathbf{R} \tag{1924}$$

where C_s is a Model Coefficient.

This model is also known as the Generalized Gradient Diffusion Hypothesis, GGDH.

Turbulent Production

The turbulent production is obtained directly, without recourse to modeling as follows:

$$\mathbf{P} = -\rho\left(\mathbf{R} \cdot \nabla \overline{\mathbf{v}}^T + \nabla \overline{\mathbf{v}} \cdot \mathbf{R}^T\right) = -\rho\left(\mathbf{R} \cdot \nabla \overline{\mathbf{v}}^T + \nabla \overline{\mathbf{v}} \cdot \mathbf{R}\right) \tag{1925}$$

Buoyancy Production

For constant-density flows using the Boussinesq approximation, the buoyancy production is modeled as:

$$\mathbf{G} = \beta \frac{\mu_t}{Pr_t}\left(\nabla \overline{T} \cdot \mathbf{g}\right) \tag{1926}$$

where:

- β is the coefficient of thermal expansion.
- $\mathbf{g}$ is the gravitational vector.
- Pr_t is the turbulent Prandtl number.
- $\overline{T}$ is the mean temperature.

For flows with varying density:

$$\mathbf{G} = \frac{\mu_t}{\rho Pr_t}\left(\nabla \rho \cdot \mathbf{g}\right) \tag{1927}$$

Turbulent Dissipation Rate

The isotropic turbulent dissipation rate is obtained from a transport equation analogous to the K-Epsilon model (and with identical boundary conditions):

$$\frac{\partial}{\partial t}(\rho\varepsilon) + \nabla\cdot(\rho\varepsilon\overline{\mathbf{v}}) = \nabla\cdot\left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\nabla\varepsilon\right] + \frac{\varepsilon}{k}\left[C_{\varepsilon 1}\left(\frac{1}{2}\mathrm{tr}(\mathbf{P}) + \frac{1}{2}C_{\varepsilon 3}\mathrm{tr}(\mathbf{G})\right) - C_{\varepsilon 2}\rho\varepsilon\right] \tag{1928}$$

where:

- $C_{\varepsilon 1}$ and $C_{\varepsilon 2}$ are Model Coefficients.
- $C_{\varepsilon 3}$ is determined as in the Standard K-Epsilon model (see K-Epsilon Model Coefficients).

Dilatation Dissipation Rate

The dilatation dissipation Υ_M is modeled as in the K-Epsilon model, using the model of Sarkar et al. [357]:

$$\Upsilon_M = \frac{C_M k\varepsilon}{c^2} \tag{1929}$$

where:

- C_M is a Model Coefficient.
- c is the speed of sound.

Model Coefficients

Coefficient	Linear Pressure Strain	Linear Pressure Strain Two Layer	Quadratic Pressure Strain	Elliptic Blending
σ_k	1	0.82	1	1
σ_ε	1.22	1	1.22	1.15
C_M (Sarkar)	2	2	2	2
C_s	0.2	0.2	0.2	0.21
$C_{\varepsilon 1}$	1.44	1.44	1.44	1.44
$C_{\varepsilon 2}$	1.92	1.92	1.83	1.83
C_μ	0.065536	0.09	0.098596	0.07

Linear Pressure-Strain Model

The linear model of Gibson and Launder [352] for the pressure-strain term comprises five terms; these are the rapid term, the slow (return-to-isotropy) term, and their respective wall-reflection terms:

$$\underline{\phi} = \underline{\phi}_s + \underline{\phi}_r + \underline{\phi}_{r,b} + \underline{\phi}_{1w} + \underline{\phi}_{2w} \tag{1930}$$

The slow pressure-strain term $\underline{\phi}_s$ is modeled as:

$$\underline{\phi}_s = -C_1 \rho \frac{\varepsilon}{k}\left(\mathbf{R} - \frac{2}{3}k\mathbf{I}\right) \tag{1931}$$

where:

- C_1 is a Model Coefficient.
- ρ is the density.
- k is the turbulent kinetic energy given by Eqn. (1923).
- $\mathbf{I}$ is the identity tensor.

The rapid pressure-strain term $\underline{\phi}_r$ is modeled as:

$$\underline{\phi}_r = -C_2\left[\mathbf{P} - \frac{1}{3}\mathbf{I}\,\mathrm{tr}\left(\mathbf{P}\right)\right] \tag{1932}$$

where:

- C_2 is a Model Coefficient.
- $\mathbf{P}$ is given by Eqn. (1925).

The buoyancy contribution $\underline{\phi}_{r,b}$ is modeled as:

$$\underline{\phi}_{r,b} = -C_3\left[\mathbf{G} - \frac{1}{3}\mathbf{I}\,\mathrm{tr}\left(\mathbf{G}\right)\right] \tag{1933}$$

where:

- C_3 is a Model Coefficient.
- $\mathbf{G}$ is given by Eqn. (1926).

The slow wall-reflection term $\underline{\phi}_{1w}$ is modeled as:

$$\underline{\phi}_{1w} = \rho C_{1w} \frac{\varepsilon}{k}\left[\left(\mathbf{R}:\mathbf{N}\right)\mathbf{I} - \frac{3}{2}\left(\mathbf{R}\cdot\mathbf{N} + \mathbf{N}\cdot\mathbf{R}\right)\right] f_w \tag{1934}$$

where:

- C_{1w} is a Model Coefficient.
- $\mathbf{N} = \mathbf{n} \otimes \mathbf{n}$, where the “wall-normal unit vector” $\mathbf{n}$ is defined as the normalized gradient of the wall distance.

f_w is calculated as:

$$f_w = \min\left(\frac{k^{3/2}}{C_l \varepsilon d}, f_w^{\max}\right) \tag{1935}$$

where:

- C_l and $f_w^{\max}$ are Model Coefficients.

- d is the distance to the wall.

The rapid wall-reflection term $\underline{\phi}_{2w}$ is modeled as:

$$\underline{\phi}_{2w} = C_{2w}\left[\left(\underline{\phi}_r:\mathbf{N}\right)\mathbf{I} - \frac{3}{2}\left(\underline{\phi}_r \cdot \mathbf{N} + \mathbf{N} \cdot \underline{\phi}_r\right)\right] f_w \tag{1936}$$

where C_{2w} is a Model Coefficient.

Model Coefficients

C_1	C_2	C_{1w}	C_{2w}	C_3	C_l	$f_w^{\max}$
1.8	0.6	0.5	0.3	0.5	2.5	1

Contents:

Linear Pressure-Strain Two-Layer Model

Linear Pressure-Strain Two-Layer Model

The advantage of the linear pressure-strain model is that it lends itself to being incorporated into a two-layer formulation, which can be used to resolve the viscous sublayer for low-Reynolds number type applications.

In the linear pressure-strain model approach, suggested by Rodi [324], the computation is divided into two layers. In the layer adjacent to the wall, the turbulent dissipation rate ε and the turbulent viscosity μ_t are specified as functions of wall distance. The values of ε specified in the near-wall layer are blended smoothly with the values computed from solving the transport equation far from the wall. The formulation is identical to the two-layer formulation used in the K-Epsilon models.

Model Coefficients

As proposed by Launder and Shima [355], the two-layer model expresses the following model coefficients of the Linear Pressure-Strain Model in terms of the turbulent Reynolds number and anisotropy tensor:

C_1	C_2	C_{1w}	C_{2w}
$1 + 2.58\, a\, a_2^{1/4}\left\{1 - \exp\left[\left(-\frac{Re_t}{150}\right)^2\right]\right\}$	$0.75\sqrt{a}$	$-\frac{2}{3}C_1 + 1.67$	$\max\left(\frac{4C_2 - 1}{6C_2}, 0\right)$

The parameter a and the tensor invariants a_2 and a_3 are defined as:

$$a = 1 - \frac{9}{8}\left(a_2 - a_3\right) \tag{1937}$$

$$a_2 = \mathbf{A}:\mathbf{A} \tag{1938}$$

$$a_3 = A_{ik}A_{kj}A_{ji} \tag{1939}$$

where the anisotropy tensor $\mathbf{A}$ is defined as:

$$\mathbf{A} = \frac{\mathbf{R}}{k} - \frac{2}{3}\mathbf{I} \tag{1940}$$

Re_t is the turbulent Reynolds number given by Eqn. (1750).

Quadratic Pressure-Strain Model

The quadratic pressure-strain model of Sarkar, Speziale and Gatski [358] uses a higher-order truncation of the expansion for the pressure-strain term:

$$\begin{aligned} \underline{\phi} = &-\left[C_{s1}\rho\varepsilon + C_{r4}\mathrm{tr}(\mathbf{P}+\mathbf{G})\right]\mathbf{A} \\ &+ C_{s2}\rho\varepsilon\left(\mathbf{A}\cdot\mathbf{A} - \frac{1}{3}\mathbf{A}{:}\mathbf{A}\,\mathbf{I}\right) + \left(C_{r3} - C_{r3}^{*}\sqrt{\mathbf{A}{:}\mathbf{A}}\right)\rho k\,\mathbf{S} \\ &+ C_{r1}\rho k\left(\mathbf{A}\cdot\mathbf{S} + \mathbf{S}\cdot\mathbf{A} - \frac{2}{3}\mathbf{A}{:}\mathbf{S}\,\mathbf{I}\right) + C_{r2}\rho k\left(\mathbf{A}\cdot\mathbf{W}^{T} + \mathbf{W}\cdot\mathbf{A}\right) \end{aligned} \tag{1941}$$

where:

- ρ is the density.
- k is the turbulent kinetic energy given by Eqn. (1923).
- $\mathbf{P}$ is given by Eqn. (1925).
- $\mathbf{G}$ is given by Eqn. (1926).
- $\mathbf{A}$ is given by Eqn. (1940).
- $\mathbf{S}$ is given by Eqn. (1745).
- $\mathbf{W}$ is given by Eqn. (1747) .
- $\mathbf{I}$ is the identity tensor.
- C_{s1}, C_{r4}, C_{s2}, C_{r3}, C_{r3}^{*}, C_{r1} ,and C_{r2} are Model Coefficients.

Model Coefficients

C_{s1}	C_{s2}
1.7	1.05

C_{r1}	C_{r2}	C_{r3}	C_{r3}^{*}	C_{r4}
0.625	0.2	0.8	0.65	0.9

Elliptic Blending Model

The Elliptic Blending RST model of Manceau and Hanjalić [345] is a low-Reynolds number model that is based on a inhomogeneous near-wall formulation of the quasi-linear quadratic pressure-strain term. The blending function is used to blend the viscous sub-layer with the log-layer formulation of the pressure-strain term. This approach requires the solution of an elliptic equation for the blending parameter α. The version of the model implemented in STAR-CCM+ was revised by Lardeau and Manceau [354].

The pressure-strain model of Manceau and Hanjalić is based on a blending of near-wall and quadratic pressure-strain models for the pressure-strain and dissipation:

$$\underline{\phi} - \underline{\varepsilon} = \left(1 - \alpha^3\right)\left(\underline{\phi}^w - \underline{\varepsilon}^w\right) + \alpha^3\left(\underline{\phi}^h - \underline{\varepsilon}^h\right) \tag{1942}$$

where the blending parameter α is the solution of the elliptic equation:

$$\alpha - L^2\nabla^2\alpha = 1 \tag{1943}$$

In the outer region, the quasi-linear version of the Sarkar, Speziale, and Gatski [358] model is used:

$$\begin{aligned} \underline{\phi}^h = &-\left[C_1\rho\varepsilon + C_{1s}\mathrm{tr}(\mathbf{P} + \mathbf{G})\right]\mathbf{A} + \left(C_3 - C_{3s}\sqrt{\mathbf{A}:\mathbf{A}}\right)\rho k\mathbf{S} \\ + \quad & C_4\rho k\left(\mathbf{S}\cdot\mathbf{A} + \mathbf{A}\cdot\mathbf{S} - \frac{2}{3}\mathbf{A}:\mathbf{S}\mathbf{I}\right) + C_5\rho k\left(\mathbf{W}\cdot\mathbf{A} + \mathbf{A}\cdot\mathbf{W}^T\right) \end{aligned} \tag{1944}$$

where:

- ρ is the density.
- k is the turbulent kinetic energy given by Eqn. (1923).
- C_1, C_{1s}, C_3, C_{3s}, C_4, and C_5 are Model Coefficients.
- $\mathbf{P}$ is given by Eqn. (1925).
- $\mathbf{G}$ is given by Eqn. (1926).
- $\mathbf{A}$ is given by Eqn. (1940).
- $\mathbf{S}$ is given by Eqn. (1745).
- $\mathbf{W}$ is given by Eqn. (1747).
- $\mathbf{I}$ is the identity tensor.

In the near-wall layer:

$$\underline{\phi}^w = -5\rho\frac{\varepsilon}{k}\left[\mathbf{R}\cdot\mathbf{N} + \mathbf{R}\cdot\mathbf{N} - \frac{1}{2}\mathbf{R}:\mathbf{N}(\mathbf{N} + \mathbf{I})\right] \tag{1945}$$

where $\mathbf{N} = \mathbf{n} \otimes \mathbf{n}$ is computed from the wall-normal vector $\mathbf{n}$, defined as:

$$\mathbf{n} = \frac{\nabla\alpha}{\sqrt{\nabla\alpha\cdot\nabla\alpha}} \tag{1946}$$

For the dissipation rate, the formulations for the outer region and the near-wall layer are:

$$\underline{\varepsilon}^h = \frac{2}{3}\varepsilon\mathbf{I} \tag{1947}$$

$$\underline{\varepsilon}^w = \mathbf{R}\frac{\varepsilon}{k} \tag{1948}$$

The turbulent length-scale L is defined as:

$$L = C_l\max\left(\frac{k^{3/2}}{\varepsilon}, C_\eta\frac{\nu^{3/4}}{\varepsilon^{1/4}}\right) \tag{1949}$$

where:

- C_l and C_η are Model Coefficients.
- ν is the kinematic viscosity.

For the Elliptic Blending model, the eddy viscosity μ_t (Eqn. (1922)) is redefined as:

$$\mu_t = \rho C_\mu k T \tag{1950}$$

where:

- C_μ is a Model Coefficient.
- T is the turbulent time-scale calculated as:

$$T = \max\left(\frac{k}{\varepsilon}, C_t\sqrt{\frac{\nu}{\varepsilon}}\right) \tag{1951}$$

  where C_t is a Model Coefficient.

An additional source term is also added to the transport equation for ε (Eqn. (1776)), in order to reproduce the correct near-wall behavior of the dissipation rate:

$$E = A_1 \nu \left(\mathbf{R}:\mathbf{N}\right)\frac{k}{\varepsilon}\left(1-\alpha^3\right)\left[\nabla\left(\left\|\mathbf{S}.\mathbf{n}\right\|\mathbf{n}\right)\right]^2 \tag{1952}$$

where A_1 is a Model Coefficient.

Model Coefficients

A_1	C_1	C_{1_s}	C_3	C_{3_s}	C_4	C_5
0.115	1.7	0.9	0.8	0.65	0.625	0.2

C_l	C_t	C_η
0.133	6	80

RANS Boundary, Region, and Initial Conditions

Non-Flow Boundaries

At wall, symmetry plane, and axis boundaries, the transported variables are defined as follows:

RANS Model	Transported Variable	Wall	Symmetry	Axis
Spalart-Allmaras models	$\tilde{\nu}$	0	$\partial\tilde{\nu}/\partial n\|_{sym} = 0$	$\partial\tilde{\nu}/\partial n\|_{axis} = 0$
K-Epsilon models	k	$\partial k/\partial n\|_{wall} = 0$	$\partial k/\partial n\|_{sym} = 0$	$\partial k/\partial n\|_{axis} = 0$
	ε	Wall cell value is specified according to the appropriate method in the Wall Treatment.	$\partial \varepsilon/\partial n\|_{sym} = 0$	$\partial \varepsilon/\partial n\|_{axis} = 0$

RANS Model	Transported Variable	Wall	Symmetry	Axis
(Lag) Elliptic Blending model	k	$\partial k/\partial n\|_{wall} = 0$	$\partial k/\partial n\|_{sym} = 0$	$\partial k/\partial n\|_{axis} = 0$
	ε	Wall cell value is specified according to the appropriate method in the Wall Treatment.	$\partial \varepsilon/\partial n\|_{sym} = 0$	$\partial \varepsilon/\partial n\|_{axis} = 0$
	φ	$\partial \varphi/\partial n\|_{wall} = 0$	$\partial \varphi/\partial n\|_{sym} = 0$	$\partial \varphi/\partial n\|_{axis} = 0$
	α	0	$\partial \alpha/\partial n\|_{sym} = 0$	$\partial \alpha/\partial n\|_{axis} = 0$
V2F model	k	$\partial k/\partial n\|_{wall} = 0$	$\partial k/\partial n\|_{sym} = 0$	$\partial k/\partial n\|_{axis} = 0$
	ε	Wall cell value is specified according to the appropriate method in the Wall Treatment.	$\partial \varepsilon/\partial n\|_{sym} = 0$	$\partial \varepsilon/\partial n\|_{axis} = 0$
	$\overline{\vartheta^2}$	0	$\partial \overline{\vartheta^2}/\partial n\Big\|_{sym} = 0$	$\partial \overline{\vartheta^2}/\partial n\Big\|_{axis} = 0$
	f	0	$\partial f/\partial n\|_{sym} = 0$	$\partial f/\partial n\|_{axis} = 0$
K-Omega models	k	$\partial k/\partial n\|_{wall} = 0$	$\partial k/\partial n\|_{sym} = 0$	$\partial k/\partial n\|_{axis} = 0$
	ω	Wall cell value is specified according to the appropriate method in the Wall Treatment.	$\partial \omega/\partial n\|_{sym} = 0$	$\partial \omega/\partial n\|_{axis} = 0$
Reynolds Stress model	$\mathbf{R}$	$\partial \mathbf{R}/\partial n\|_{wall} = 0$ In addition, a method that is developed by Hadzic [353] is used to impose the value of production of each stress component in accordance with a wall function approach. See Wall Treatment.	See Reynolds Stress Model—Symmetry.	See Reynolds Stress Model—Axis.

Reynolds Stress Model—Symmetry

On a symmetry boundary, the procedure for setting the specific Reynolds stress tensor at the face $\mathbf{R}_f$ is as follows. First, the tensor at the adjacent cell center $\mathbf{R}$ is rotated such that it is parallel to the symmetry plane to obtain a new tensor termed $\mathbf{R}'$. This then enables the symmetry boundary face values of the tensor in this rotated coordinate system to be explicitly set as follows:

$$\mathbf{R}'_f = \begin{bmatrix} R'_{11} & 0 & R'_{13} \\ 0 & R'_{22} & 0 \\ R'_{31} & 0 & R'_{33} \end{bmatrix}_c \tag{1953}$$

The values of $\mathbf{R}_f$ are then computed by rotating the $\mathbf{R}'_f$ tensor back to the original orientation of $\mathbf{R}$—this is made easy by the fact that the inverse of the rotation tensor is simply its transpose, since it is an orthogonal tensor.

Reynolds Stress Model—Axis

On an axis boundary, the procedure for setting the specific Reynolds stress tensor at the face $\mathbf{R}_f$ is similar to the symmetry plane. In this case, however, the face values of the tensor in the coordinate system parallel to the axis are as follows:

$$\mathbf{R}'_f = \begin{bmatrix} R'_{11} & 0 & 0 \\ 0 & R'_{22} & 0 \\ 0 & 0 & R'_{33} \end{bmatrix}_c \tag{1954}$$

Flow Boundaries, Region, and Initial Conditions

When defining values for flow boundaries, region, and initial conditions, you have three choices for specifying the transported variables:

- Specify the values of the transported variables directly.
- Have the values of the transported variables derived from a specified turbulence intensity I and length scale L.
- Have the values of the transported variables derived from a specified turbulence intensity I and turbulent viscosity ratio μ_t/μ.

RANS Model	Direct Specification of transported variables	Derivation of transported variables from I and L	Derivation of transported variables from I and μ_t/μ
Spalart-Allmaras model	$\tilde{\nu}$	$C_\mu^{0.25}\sqrt{\frac{3}{2}}I\mathrm{v}L$ (1955)	$\frac{\mu_t}{\mu} = \frac{\rho\tilde{\nu}f_{v1}}{\mu}$ (1956)
K-Epsilon model	k	$\frac{3}{2}(I\mathrm{v})^2$ (1957)	$\frac{3}{2}(I\mathrm{v})^2$ (1958)
	ε	$\frac{C_\mu^{3/4}k^{3/2}}{L}$ (1959)	$\frac{\rho C_\mu k^2}{(\mu_t/\mu)\,\mu}$ (1960)
(Lag) Elliptic Blending model	k	As for K-Epsilon model.	
	ε		
	φ	φ	φ
	NOTE: In porous regions, the value of φ is set to $2/3$.		
V2F model	k	As for K-Epsilon model.	
	ε		
	$\overline{\vartheta^2}$	$\frac{2}{3}k$ (1961)	$\frac{2}{3}k$ (1962)
	NOTE: In porous regions, the value of $\overline{\vartheta^2}$ is set to the specified initial condition value.		

RANS Model	Direct Specification of transported variables	Derivation of transported variables from I and L	Derivation of transported variables from I and μ_t/μ
	f (initial conditions only)	0.2 (initial conditions only)	0.2 (initial conditions only)
	NOTE: In porous regions, the value of f is set to the specified initial condition value. For flow boundaries, f is set to the value of the boundary-next cell.		
K-Omega model	k	As for K-Epsilon model.	
	ω	$\frac{\sqrt{k}}{L\beta^{*1/4}}$ (1963)	$\frac{\rho k}{(\mu_t/\mu)\mu}$ (1964)
Reynolds Stress model	$\mathbf{R}$	$\frac{2}{3}(I\mathrm{v})^2 I$ (1965)	$(I\mathrm{v})^2\mathbf{I}$ (1966)
	ε	$\frac{C_\mu{}^{3/4}\left[\frac{1}{2}tr\mathbf{R}\right]^{3/2}}{L}$ (1967)	$\frac{C_\mu\left[\frac{1}{2}tr\mathbf{R}\right]^2}{(\mu_t/\mu)\mu}$ (1968)

where:

- $\mathbf{v}$ is the local velocity magnitude.
- C_μ and β^* are model coefficients.
- f_{v1} is a Damping Function.

For the Reynolds stress model, a fourth choice allows you to enter the dissipation rate ε directly but have the Reynolds stresses $\mathbf{R}$ derived from the turbulent kinetic energy k using the relation:

$$\mathbf{R} = \frac{2}{3}k\mathbf{I} \tag{1969}$$

where $\mathbf{I}$ is the identity matrix.

Boundary Layer Initialization

When convergence is slow, or when the initialization of the boundary layer is the main consideration (as in internal flows), you can initialize flow in the near-wall region only using the local relative velocity u_r, the wall-parallel non-dimensional velocity u^+ and the y^+ value:

RANS Model	Transported Variable	Calculation
K-Epsilon model	k	$\sqrt{\left(\frac{1}{du^+/dy^+} - 1\right)\frac{\varepsilon\kappa\nu}{f_u}}$ (1970)

RANS Model	Transported Variable	Calculation
	ε	$\frac{1}{\max\left(17,\ y^{+}\right)}\frac{u_r^4}{\kappa\nu}$ (1971)
(Lag) Elliptic Blending model	k	As for K-Epsilon model.
	ε	
	φ	As specified for region initialization.
V2F model	k	As for K-Epsilon model.
	ε	
	$\overline{\vartheta^2}$	As specified for region initialization.
	f	
K-Omega model	k	As for K-Epsilon model.
	ω	$\frac{\varepsilon}{\beta^* f_{\beta^*} k}$ (1972) where k and ε are calculated as for K-Epsilon model.
Reynolds Stress model	$\mathbf{R}$	$\frac{2}{3}k\mathbf{I}$ (1973) where k is calculated as for K-Epsilon model.
	ε	As for K-Epsilon model.

where:

- κ is the von Karman constant.
- $f_u = \exp\left(-\frac{1.0674 \cdot 10^{-3}}{4} y^{+3}\right)$
- β^* is a Model Coefficient.
- f_{β^*} is the Free-Shear Modification factor.

~RANS Wall Treatment~

For this release, *User Guide* > *Modeling Physics* > *Modeling Turbulence* > *Turbulence Formulation* > *Wall Laws Formulation*.

Theory guide content is planned for a future release.

Scale-Resolving Simulations

In contrast to RANS models, scale-resolving simulations resolve the large scales of turbulence and model small-scale motions.

Two approaches are available in STAR-CCM+:

- Large Eddy Simulation (LES)
- Detached Eddy Simulation (DES)

A Large Eddy Simulation resolves turbulent structures in space everywhere in the flow domain down to the grid limit, where subgrid models approximate the impact of the subgrid structures on the flow field. In order to resolve the crucial turbulent structures near the wall, this approach requires an excessively high mesh resolution in the wall boundary layer—not only in the direction normal to the wall but also in the flow direction. As a result of the high computational costs that go along with the high cell count, LES is used mainly for academic applications or for flows with low Reynolds numbers.

A Detached Eddy Simulation (DES) is a Hybrid LES-RANS approach that resolves turbulent structures in the core flow and covers the wall boundary layers by a RANS model. Thereby, DES avoids the expensive mesh requirements of LES.

Contents:

Large Eddy Simulation (LES)
Detached Eddy Simulation (DES)
Boundary and Initial Conditions for Scale-Resolving Simulations

Large Eddy Simulation (LES)

Large Eddy Simulation (LES) is an inherently transient technique in which the large scales of the turbulence are directly resolved everywhere in the flow domain, and the small-scale motions are modeled.

One justification for the LES technique is that by modeling “less” of the turbulence, and explicitly solving for more of it, the error in the turbulence modeling assumptions is not as consequential. Furthermore, it is hypothesized that the smaller eddies are self-similar and thus lend themselves to simpler and more universal models. The downside of the approach is the computational expense, which, although less than direct numerical simulation, is still nonetheless excessive.

In contrast to the RANS equations, the equations that are solved for LES are obtained by a spatial filtering rather than an averaging process. Each solution variable ϕ is decomposed into a filtered value $\widetilde{\phi}$ and a sub-filtered, or sub-grid, value ϕ':

$$\phi = \widetilde{\phi} + \phi' \tag{1974}$$

where ϕ represents velocity components, pressure, energy, or species concentration.

The spatial filtering removes the smaller eddies—associated with higher frequencies—and thereby reduces the range of scales that must be resolved. LES filtering can be either explicit or implicit. Explicit filtering

applies a filter function (such as box or Gaussian) to the discretized Navier-Stokes equations. The filtering of the generic instantaneous flow variable $\phi(t,x)$ is defined as:

$$\widetilde{\phi}(t,x) = \iiint_{-\infty}^{\infty} G\left(x - x', \Delta\right)\phi(t,x')dx' \tag{1975}$$

where $G(\mathbf{x},\Delta)$ is the filter function characterized by a filter width $\Delta = \left(\Delta_x \Delta_y \Delta_z\right)^{1/3}$.

In STAR-CCM+, implicit filtering is used. For this approach, the computational grid determines the scales of the eddies that are filtered out. Implicit filtering takes full advantage of the grid resolution and is, in general, computationally less expensive than explicit filtering.

Inserting the decomposed solution variables into the Navier-Stokes equations results in equations for the filtered quantities. The filtered mass and momentum transport equations can be written as:

$$\frac{\partial \rho}{\partial t} + \nabla\cdot\left(\rho\widetilde{\mathbf{v}}\right) = 0 \tag{1976}$$

$$\frac{\partial}{\partial t}\left(\rho\widetilde{\mathbf{v}}\right) + \nabla\cdot\left(\rho\widetilde{\mathbf{v}}\otimes\widetilde{\mathbf{v}}\right) = -\nabla\cdot\widetilde{p}\mathbf{I} + \nabla\cdot\left(\mathbf{T} + \mathbf{T}_t\right) + \mathbf{f}_b \tag{1977}$$

where:

- ρ is the density.
- $\widetilde{\mathbf{v}}$ and $\widetilde{p}$ are the filtered velocity and pressure respectively.
- $\mathbf{I}$ is the identity tensor.
- $\mathbf{T}$ is the stress tensor.
- $\mathbf{f}_b$ is the resultant of the body forces (such as gravity and centrifugal forces).

The filtered equations are rearranged into a form that looks identical to the unsteady RANS equations. However, the turbulent stress tensor $\mathbf{T}_t$ now represents the subgrid scale stresses. These stresses result from the interaction between the larger, resolved eddies and the smaller, unresolved eddies and are modeled using the Boussinesq approximation as follows:

$$\mathbf{T}_t = 2\mu_t\mathbf{S} - \frac{2}{3}\left(\mu_t\nabla\cdot\widetilde{\mathbf{v}} + \rho k\right)\mathbf{I} \tag{1978}$$

where:

- $\mathbf{S}$ is the strain rate tensor given by Eqn. (1745) and computed from the resolved velocity field $\widetilde{\mathbf{v}}$.
- k is the subgrid scale turbulent kinetic energy.

The subgrid scale turbulent viscosity μ_t must be described by a subgrid scale model that accounts for the effects of small eddies on the resolved flow. Currently, three subgrid scale models are available in STAR-CCM+:

1. Smagorinsky Subgrid Scale model
2. Dynamic Smagorinsky Subgrid Scale model
3. WALE Subgrid Scale model

The subgrid scale turbulent kinetic energy k is a physical by-product of the STAR-CCM+ flow solution. Within the computation of the subgrid turbulent viscosity is an implicit level of subgrid kinetic energy k_{SGS}.

Contents:

Smagorinsky Subgrid Scale Model
Dynamic Smagorinsky Subgrid Scale Model
WALE Subgrid Scale Model
~LES Wall Treatment~

Smagorinsky Subgrid Scale Model

The Smagorinsky Subgrid Scale model is the original and most basic subgrid scale model, upon which many more advanced models are based. This model uses a mixing length hypothesis to model the subgrid scale stresses.

One shortcoming of the Smagorinsky Subgrid Scale model is that it contains a model coefficient C_s that is not universal and depends on the local flow conditions. Additionally, a damping function is needed for proper results in wall-bounded flows. The Van Driest damping function is a non-local operation, requiring the use of a KD-tree data structure to communicate wall information to the interior. In simulations with very complex geometries, this KD-tree must be broadcast to each parallel node, resulting in a memory and communication overhead. An update policy can be used to improve the communication overhead, but not the memory overhead. Therefore, this model should be avoided for large-scale computations on complex geometries. It is, however, useful for reference simulations, since it is the original algebraic subgrid scale model formulation.

Smagorinsky Subgrid Scale Viscosity

The Smagorinsky Subgrid Scale model [364] provides the following mixing-length type formula for the subgrid scale viscosity:

$$\mu_t = \rho \Delta^2 S \tag{1979}$$

where:

- ρ is the density.
- Δ is the length scale or grid filter width.
- S is given by Eqn. (1744) and computed from the resolved velocity field $\tilde{\mathbf{v}}$.

The length scale Δ is directly related to the cell volume V and the wall distance d as follows:

$$\Delta = \begin{cases} f_v C_s V^{1/3} & \text{if length scale limit not applied} \\ f_v \min\left(\kappa d, C_s V^{1/3}\right) & \text{if length scale limit is applied} \end{cases} \tag{1980}$$

where:

- f_v is the Van Driest damping function.
- C_s is a Model Coefficient.
- κ is the von Karman constant (see Model Coefficients).

The Van Driest damping function f_v computed as follows:

$$f_v = \begin{cases} 1 & \text{no damping} \\ 1-\exp\left(-\frac{y^+}{A}\right) & \text{standard} \\ \sqrt{1-\exp\left(-\frac{y^+}{A}\right)^3} & \text{modified} \end{cases} \tag{1981}$$

where A is a Model Coefficient.

y^+ is the dimensionless wall distance defined as:

$$y^+ = \frac{u^* d}{\nu} \tag{1982}$$

where:

- u^* is the friction velocity computed from the instantaneous wall shear stress at the nearest wall face.
- d is the distance to the wall.
- ν is the kinematic viscosity.

The modified version of the Van Driest damping function [366] is attributed to Piomelli and co-workers [360].

Model Coefficients

C_s	κ	A
0.1	0.41	25

The model coefficient C_s is not universal. Typical values of C_s reported in the literature range from 0.165 for homogeneous isotropic decaying turbulence to 0.07 for channel flows. The default value of $C_s = 0.1$ is a compromise between these two extremes.

Dynamic Smagorinsky Subgrid Scale Model

The Dynamic Smagorinsky Subgrid Scale model has the same basic form as the Smagorinsky model, however instead of using a single user-defined C_s coefficient, the model computes a local time-varying coefficient by test-filtering the flow field at a length scale greater than the grid length scale. This dynamic variation of the constant gives the model its name, and allows it to compute the correct result for wall-bounded flows without the use of damping functions.

Dynamic Smagorinsky Subgrid Scale Viscosity

The calculation of the turbulent viscosity in the Dynamic Smagorinsky SGS model ([361], [362]) follows the same equation as for the standard Smagorinsky SGS model:

$$\mu_t = \rho \Delta^2 S \tag{1983}$$

where:

- ρ is the density.
- S is given by Eqn. (1744) and computed from the resolved velocity field $\tilde{\mathbf{v}}$.

However, the parameter Δ^2 is computed dynamically as a function of a number of test-filtered variables:

$$\Delta^2 = C_s^2 V^{2/3} \tag{1984}$$

where V is the cell volume.

If $\widetilde{\phi}$ is the grid-filtered LES variable in a cell, then the corresponding test-filtered value $\widehat{\widetilde{\phi}}$ is given as:

$$\widehat{\widetilde{\phi}} = \frac{1}{\sum_{n=0}^{N} V_n} \sum_{n=1}^{N} \widetilde{\phi}_n V_n \tag{1985}$$

where the subscripts denote cell number, cell 0 is the current cell, and cells 1 to N are the face neighbors of cell 0.

Continuing with the superscript "~" notation to denote test-filtered variables, the dynamic parameter C_s^2 is computed as follows:

$$L_{ij} = \widetilde{u_i}\widehat{\widetilde{u_j}} - \widehat{\widetilde{u_i}}\widehat{\widetilde{u_j}} \tag{1986}$$

$$M_{ij} = 2\widetilde{L}^2\left(\left|\widetilde{S}\right|\widehat{\widetilde{S}}_{ij} - \frac{\widehat{L}^2}{\widetilde{L}^2}\left|\widehat{\widetilde{S}}\right|\widehat{\widetilde{S}}_{ij}\right) \tag{1987}$$

where:

- $\widetilde{L}$ is the grid filter length.
- $\widehat{L}/\widetilde{L}$ is the filter width ratio and a Model Coefficients.

The tensor M_{ij} arises during the derivation of the model by subtracting the Smagorinsky subgrid stress tensor at the grid filter scale from the same tensor at the test filter scale. Finally, C_s^2 is computed as:

$$C_s^2 = \frac{\langle\ L_{ij}M_{ij}\ \rangle}{\langle\ M_{ij}M_{ij}\ \rangle} \tag{1988}$$

where the brackets <> denote averaging. Since, in most simulations, there is no homogeneous direction over which to average, the same approach is used as for the filtering operation. This averaging is optional; you can activate or deactivate it in the *Average Parameter* expert property of the model in the GUI.

To increase stability in the calculation, the values of C_s^2 are clipped using minimum and maximum values. See Model Coefficients.

Model Coefficients

$\widehat{L}/\widetilde{L}$	$C_{s\,\mathrm{min}}^2$	$C_{s\,\mathrm{max}}^2$
2	0	1000000

WALE Subgrid Scale Model

The WALE (Wall-Adapting Local-Eddy Viscosity) Subgrid Scale model is a more modern subgrid scale model that uses a novel form of the velocity gradient tensor in its formulation.

Similar to the Smagorinsky Subgrid Scale model, it suffers from the limitation that the model coefficient C_w (see Eqn. (1990)) is not universal. But validations using STAR-CCM+ have shown that the WALE model is seemingly less sensitive to the value of this coefficient than the Smagorinsky model. Another advantage of the WALE model is that it does not require any form of near-wall damping—it automatically gives accurate scaling at walls.

WALE Subgrid Scale Viscosity

The WALE Subgrid Scale model [363] provides the following mixing-length type formula for the subgrid scale viscosity:

$$\mu_t = \rho \Delta^2 S_w \tag{1989}$$

where:

- ρ is the density.
- Δ is the length scale or grid filter width.
- S_w is a deformation parameter.

The length scale Δ is defined in terms of the cell volume V as:

$$\Delta = \begin{cases} C_w V^{1/3} & \text{if length scale limit is not applied} \\ \min\left(\kappa d, C_w V^{1/3}\right) & \text{if length scale limit is applied} \end{cases} \tag{1990}$$

where:

- C_w is a Model Coefficient.
- κ is the von Karman constant. See Model Coefficients.

The deformation parameter S_w is defined as:

$$S_w = \frac{\mathbf{S}_d{:}\mathbf{S}_d{}^{3/2}}{\mathbf{S}_d{:}\mathbf{S}_d{}^{5/4} + \mathbf{S}{:}\mathbf{S}^{5/2}} \tag{1991}$$

where:

- $\mathbf{S}$ is given by Eqn. (1745) and computed from the resolved velocity field $\tilde{\mathbf{v}}$..

The tensor $\mathbf{S}_d$ is defined as:

$$\mathbf{S}_d = \frac{1}{2}\left[\nabla\mathbf{v} \cdot \nabla\mathbf{v} + \left(\nabla\mathbf{v} \cdot \nabla\mathbf{v}\right)^T\right] - \frac{1}{3}\mathrm{tr}\left(\nabla\mathbf{v} \cdot \nabla\mathbf{v}\right)\mathbf{I} \tag{1992}$$

where $\mathbf{I}$ is the identity tensor.

Model Coefficients

C_w	κ
0.544	0.41

The model coefficient C_w is not universal. Typical values of C_w reported in the literature range from 0.5 for homogeneous isotropic decaying turbulence to 0.325 for channel flows. Validations using STAR-CCM+ have shown that the default value of $C_w = 0.544$ works well for both homogeneous isotropic decaying turbulence and for channel flows.

~LES Wall Treatment~

Theory guide content is planned for a future release.

Detached Eddy Simulation (DES)

Detached Eddy Simulation (DES) is a hybrid modeling approach that combines features of Reynolds-Averaged (RANS) simulation in some parts of the flow and Large Eddy Simulation (LES) in others.

The unsteady RANS equations are applicable to transient situations where the unsteadiness is either imposed, such as by a time-varying boundary condition, or is inherent, such as the vortex shedding in a massively separated flow. In the latter case, transient simulations often yield better results than attempting use a steady-state approach. However, successful unsteady RANS simulations require that the time scales of the turbulence be disparate from the mean-flow unsteadiness. Furthermore, the limitations of the turbulence model may preclude good unsteady results.

DES turbulence models are set up so that boundary layers and irrotational flow regions are solved using a base RANS closure model. However, the turbulence model is intrinsically modified so that, if the grid is fine enough, it will emulate a basic LES subgrid scale model in detached flow regions. In this way, one gets the best of both worlds: a RANS simulation in the boundary layers and an LES in the unsteady separated regions. For more background on DES, see [371].

STAR-CCM+ provides the DES modeling approach for three different RANS models:

- Spalart-Allmaras DES
- Elliptic Blending K-Epsilon DES
- SST K-Omega DES

The following DES variants are implemented:

- Delayed Detached Eddy Simulation (DDES)
- Improved Delayed Detached Eddy Simulation (IDDES)

The DDES model incorporates a delay factor that enhances the ability of the model to distinguish between LES and RANS regions on meshes where spatial refinement could give rise to ambiguous behavior.

For IDDES, the subgrid length-scale includes a dependence on the wall distance. This approach allows RANS to be used in a much thinner near-wall region, in which the wall distance is much smaller than the boundary-

layer thickness. IDDES was introduced in order to provide some WMLES (Wall-modelled LES) capabilities to the DES formulation.

While DES holds great promise for certain types of simulations, it must be cautioned that this technology is not the answer to all turbulence modeling problems. The creation of suitable grids is something of an art, and an unsteady simulation is required, with statistics gathered to obtain a mean field as in a LES.

Flow Simulation Methodology

The momentum equations for the RANS averaged velocity $\overline{\mathbf{v}}$ and the LES filtered velocity $\widetilde{\mathbf{v}}$ are defined as:

$$\frac{\partial}{\partial t}\left(\rho\overline{\mathbf{v}}\right) + \nabla\cdot\left(\rho\overline{\mathbf{v}}\otimes\overline{\mathbf{v}}\right) = -\nabla\cdot\overline{p}\mathbf{I} + \nabla\cdot\left(\mathbf{T} + \mathbf{T}_t^{RANS}\right) + \mathbf{f}_b \tag{1993}$$

$$\frac{\partial}{\partial t}\left(\rho\widetilde{\mathbf{v}}\right) + \nabla\cdot\left(\rho\widetilde{\mathbf{v}}\otimes\widetilde{\mathbf{v}}\right) = -\nabla\cdot\widetilde{p}\mathbf{I} + \nabla\cdot\left(\mathbf{T} + \mathbf{T}_t^{LES}\right) + \mathbf{f}_b \tag{1994}$$

For a RANS model, the Reynolds-stress tensor $\mathbf{T}_t^{RANS}$ is a function of a time-and length-scale

$$\mathbf{T}_t^{RANS} = f(\nabla\cdot\overline{\mathbf{v}}, k, \varepsilon) \tag{1995}$$

and for LES,

$$\mathbf{T}_t^{LES} = f(\nabla\cdot\widetilde{\mathbf{v}}, \Delta) \tag{1996}$$

In Eqn. (1996), the most crucial role is played by the filter width, or length-scale, which is used not only for the definition of the subgrid-scale eddy viscosity, but is also used in many models for the definition of the transition region.

The similarities of Eqn. (1993) and Eqn. (1994) suggest that a unified approach can be used to solve for a quantity $\widehat{u_i}$, that is dependant on the local definition:

$$\frac{\partial}{\partial t}\left(\rho\widehat{\mathbf{v}}\right) + \nabla\cdot\left(\rho\widehat{\mathbf{v}}\otimes\widehat{\mathbf{v}}\right) = -\nabla\cdot\widehat{p}\mathbf{I} + \nabla\cdot\left(\mathbf{T} + \mathbf{T}_t^{model}\right) + \mathbf{f}_b \tag{1997}$$

where the modeled stress tensor $\mathbf{T}_t^{model}$ is defined as:

$$\mathbf{T}_t^{model} = f_\Delta\left(\frac{\Delta}{l_k}\right)\mathbf{T}_t^{RANS} \tag{1998}$$

where:

- Δ is the local measure of the grid size.
- l_k is the turbulent length-scale.
- f_Δ is a damping function that depends on the DES variant (DDES or IDDES) .

Contents:

Spalart-Allmaras DES
Elliptic Blending DES
SST K-Omega DES
Function for Hybrid Upwind Scheme

Spalart-Allmaras DES

The Spalart-Allmaras Detached Eddy model combines features of the Standard Spalart-Allmaras RANS model in the boundary layers with a large eddy simulation (LES) in unsteady separated regions.

The DES formulation ([371], [368]) is obtained by replacing the distance to the nearest wall d in the turbulent dissipation term $\Upsilon_{\tilde{\nu}}$ (see Eqn. (1762)) by $\tilde{d}$. The definition of $\tilde{d}$ depends on the DES approach used.

DDES Formulation

To prevent a premature onset of the LES mode within the boundary layer of ambiguous meshes, Spalart and others [372] have introduced a correction to the definition of the dissipation length scale $\tilde{d}$ that depends on the value of the turbulent eddy viscosity and velocity gradient. The new approach has been named Delayed DES (DDES).

For DDES, the length scale $\tilde{d}$ is computed as:

$$\tilde{d} = d - f_d \max\left(0, d - \psi C_{\mathrm{DES}} \Delta\right) \tag{1999}$$

and:

- C_{DES} is a Model Coefficient.
- Δ is the largest distance between the cell center under consideration and the cell centers of the neighboring cells.

The function f_d is [372]:

$$f_d = 1 - \tanh\left[\left(8 r_d\right)^3\right] \tag{2000}$$

where:

$$r_d = \frac{\tilde{\nu}}{\sqrt{\nabla \mathbf{v} : \nabla \mathbf{v}^T} \kappa^2 d^2} \tag{2001}$$

and κ is the von Karman constant. See Spalart-Allmaras Model—Model Coefficients.

The optional low-Reynolds number correction function ψ is designed to prevent activation of the low-Reynolds number terms when in “LES” mode [372]. It is defined as:

$$\psi^2 = \min\left\{100, \frac{1 - C_{b1}\left(\left[f_{t2} + \left(1 - f_{t2}\right) f_{v2}\right] / f_w^* \kappa^2 C_{w1}\right)}{\left(1 - f_{t2}\right) f_{v1}}\right\} \tag{2002}$$

where:

- C_{b1} and C_{w1} are given in Spalart-Allmaras Model—Model Coefficients.
- f_{t2} is a damping function defined as appropriate for the DES model. See DES Damping Function.
- f_{v1} and f_{v2} are given in Spalart-Allmaras Model—Damping Functions.

The coefficient f_w^* is a computed iteratively as a nonlinear function of the other model coefficients, as described in [372]. It has a value of 0.424 when the standard model coefficients are used.

IDDES Formulation

For the IDDES formulation of Shur and others [370], two more functions are introduced to the calculation of the modified wall distance $\tilde{d}$ to add Wall-Modeled LES (WMLES) capability: a blending function f_B and the so-called "elevating" function f_e.

The length scale $\tilde{d}$ is computed as:

$$\tilde{d} = \widetilde{f_d}(1 + f_e)d + (1 - \widetilde{f_d})\psi C_{\mathrm{DES}}\,\Delta_{\mathrm{IDDES}} \tag{2003}$$

where:

$$f_B = \min\left[2\exp\left(-9\alpha^2\right), 1\right] \tag{2004}$$

$$\alpha = 0.25 - \frac{d}{\Delta} \tag{2005}$$

$$f_e = \max\left[\left(f_{e1} - 1\right), 0\right]\psi f_{e2} \tag{2006}$$

$$f_{e1} = \begin{cases} 2\exp\left(-11.09\alpha^2\right) & \text{if } \alpha \geq 0 \\ 2\exp\left(-9\alpha^2\right) & \text{if } \alpha < 0 \end{cases} \tag{2007}$$

$$f_{e2} = 1 - \max\left(f_t, f_l\right) \tag{2008}$$

$$f_t = \tanh\left[\left(C_t^2 r_{dt}\right)^3\right] \tag{2009}$$

$$f_l = \tanh\left[\left(C_l^2 r_{dl}\right)^{10}\right] \tag{2010}$$

$$r_{dt} = \frac{\nu_t}{\sqrt{\nabla\mathbf{v} : \nabla\mathbf{v}^T \kappa^2 d^2}} \tag{2011}$$

$$r_{dl} = \frac{\nu}{\sqrt{\nabla\mathbf{v} : \nabla\mathbf{v}^T \kappa^2 d^2}} \tag{2012}$$

where C_t and C_l are Model Coefficients.

The introduction of the low-Reynolds number correction function ψ in the elevating function f_e is unrelated to the low-Reynolds number correction role of this function in the LES mode of DDES, and is purely empirical.

The WMLES and DDES branches of the model are combined using a modified version of the DDES f_d function as follows:

$$\tilde{f}_d = \max\left(\left(1 - f_{dt}\right), f_B\right) \tag{2013}$$

$$f_{dt} = 1 - \tanh\left[\left(C_{dt} r_{dt}\right)^3\right] \tag{2014}$$

where C_{dt} is a Model Coefficient.

The IDDES model also uses an altered version of the mesh length scale Δ_{IDDES}, computed as:

$$\Delta_{\mathrm{IDDES}} = \min\left(\max\left(0.15d, 0.15\Delta, \Delta_{\min}\right), \Delta\right) \tag{2015}$$

where $\Delta_{\min}$ is the smallest distance between the cell center under consideration and the cell centers of the neighboring cells.

DES Damping Function

The DDES and IDDES versions of the Spalart-Allmaras model contain a modification to the f_{t2} damping function (see Spalart-Allmaras Model—Damping Functions).

f_{t2} is defined as recommended by Squires and others [373] as:

$$f_{t2} = C_{t3}\exp\left(-C_{t4}\chi^2\right) \tag{2016}$$

This modification does not affect the predictions of fully turbulent flows, but prevents spurious upstream propagation of the eddy viscosity into attached laminar regions.

DES Model Coefficients

C_{dt}	C_{DES}	C_t	C_l
8	0.65	1.63	3.55

Elliptic Blending DES

The Elliptic Blending Detached Eddy model combines features of the Elliptic Blending RANS model in the boundary layers with a large eddy simulation (LES) in unsteady separated regions.

For the Elliptic Blending model, the DDES approach is implemented in STAR-CCM+. The formulation is obtained by modifying the turbulent dissipation rate ε in the transport equation for the turbulent kinetic energy.

DDES Formulation

The turbulent dissipation rate ε in Eqn. (1875) is replaced by $\tilde{\varepsilon}$, where:

$$\tilde{\varepsilon} = \frac{k^{3/2}}{\tilde{d}} \tag{2017}$$

where k is the turbulent kinetic energy and $\tilde{d}$ is defined as:

$$\tilde{d} = \frac{k^{3/2}}{\varepsilon} - f_d \max\left(0, \frac{k^{3/2}}{\varepsilon} - \psi C_{\mathrm{DES}}\Delta\right) \tag{2018}$$

where:

- f_d is given by Eqn. (2000).
- C_{DES} is a Model Coefficient.
- $\Delta = \max\left(\Delta_x, \Delta_y, \Delta_z\right)$ is the filter width, where Δ_x, Δ_y, and Δ_z are cell characteristic mesh spacing in all spacial directions.

The correction factor ψ is calculated as:

$$\psi = \left[\frac{C_{\varepsilon 1}}{C_{\varepsilon 2} C_{\mu} \varphi}\right]^{3/4} \tag{2019}$$

where $C_{\varepsilon 1}$, $C_{\varepsilon 2}$, and C_{μ} are given in Elliptic Blending Model—Model Coefficients.

DES Model Coefficient

C_{DES}
0.2

SST K-Omega DES

The SST K-Omega Detached Eddy model combines features of the SST K-Omega RANS model in the boundary layers with a large eddy simulation (LES) in unsteady separated regions.

The DES formulation of the SST K-Omega model is obtained by modifying the dissipation term in the transport equation for the turbulent kinetic energy.

DDES Formulation

Based on the work by Menter and Kuntz [369], the specific dissipation rate ω in Eqn. (1823) is replaced by $\widetilde{\omega}$, where:

$$\widetilde{\omega} = \omega \phi \tag{2020}$$

and ϕ is defined as:

$$\phi = \max\left(\frac{l_t}{C_{\mathrm{DES}} \Delta} F, 1\right) \tag{2021}$$

where:

- $l_t = \frac{\sqrt{k}}{\beta^* \omega}$
- Δ is the largest distance between the cell center under consideration and the cell centers of the neighboring cells
- $F = 1 - F_2$, where F_2 is the blending function given by Eqn. (1822).

For $\phi = 1$, the RANS solution is recovered, and for $\phi > 1$, the solution tends towards the LES definition.

The model coefficient C_{DES}, which blends the values obtained from independent calibration of the K-Epsilon and K-Omega branches of the K-Omega SST model [374], is evaluated as follows:

$$C_{\mathrm{DES}} = C_{\mathrm{DES},k\text{-}\omega} F_1 + C_{\mathrm{DES},\,k\text{-}\varepsilon}\left(1 - F_1\right) \tag{2022}$$

where:

- $C_{\mathrm{DES},k\text{-}\omega}$ and $C_{\mathrm{DES},k\text{-}\varepsilon}$ are Model Coefficients.
- F_1 is defined in Eqn. (1839).

IDDES Formulation

For the IDDES formulation of Shur et al. [370], the specific dissipation rate ω in Eqn. (1823) is replaced by $\widetilde{\omega}$ defined as:

$$\widetilde{\omega} = \frac{\sqrt{k}}{l_{\mathrm{HYBRID}}\beta^* f_{\beta^*}} \tag{2023}$$

where:

- β^* is given in K-Omega Model—Model Coefficients.
- f_{β^*} is the free-shear modification factor.

and:

$$l_{\mathrm{HYBRID}} = \widetilde{f}_d\left(1 + f_e\right)l_t + \left(1 - \widetilde{f}_d\right)C_{\mathrm{DES}}\Delta_{\mathrm{IDDES}} \tag{2024}$$

Two more functions are introduced to the length scale calculation to add Wall-Modeled LES (WMLES) capability, a blending function f_B and the so-called "elevating" function f_e:

$$f_B = \min\left[2\exp\left(-9\alpha^2\right), 1\right] \tag{2025}$$

$$\alpha = 0.25 - \frac{d}{\Delta} \tag{2026}$$

$$f_e = \max\left[\left(f_{e1} - 1\right), 0\right]\psi f_{e2} \tag{2027}$$

$$f_{e1} = \left\{\begin{pmatrix} 2\exp\left(-11.09\alpha^2\right) & \text{if } \alpha \geq 0 \\ 2\exp\left(-9\alpha^2\right) & \text{if } \alpha < 0 \end{pmatrix}\right. \tag{2028}$$

$$f_{e2} = 1 - \max\left(f_t, f_l\right) \tag{2029}$$

$$f_t = \tanh\left[\left(C_t^2 r_{dt}\right)^3\right] \tag{2030}$$

$$f_l = \tanh\left[\left(C_l^2 r_{dl}\right)^{10}\right] \tag{2031}$$

$$r_{dt} = \frac{\nu_t}{\sqrt{\nabla\mathbf{v} : \nabla\mathbf{v}^T}\kappa^2 d^2} \tag{2032}$$

$$r_{dl} = \frac{\nu}{\sqrt{\nabla\mathbf{v} : \nabla\mathbf{v}^T}\kappa^2 d^2} \tag{2033}$$

where:

- C_t and C_l are Model Coefficients.
- ν is the kinematic viscosity.
- $\nu_t = \frac{\mu_t}{\rho}$, where μ_t is the turbulent eddy viscosity.
- κ is the von Karman constant. See K-Omega Model—Model Coefficients.
- d is the distance to the wall.

The introduction of the low-Reynolds number correction function ψ in the formulation of f_e is unrelated to the low-Reynolds number correction role of this function in the LES mode of DDES, and is purely empirical.

The WMLES and DDES branches of the model are combined using a modified version of the DDES f_d function as follows:

$$\widetilde{f}_d = \max\left(\left(1-f_{dt}\right), f_B\right) \tag{2034}$$

where:

$$f_{dt} = 1-\tanh\left[\left(C_{dt}r_{dt}\right)^3\right] \tag{2035}$$

and C_{dt} is a Model Coefficient.

The IDDES model also uses an altered version of the mesh length scale Δ_{IDDES}, computed as:

$$\Delta_{\mathrm{IDDES}} = \min\left(\max\left(0.15d, 0.15\Delta, \Delta_{\min}\right), \Delta\right) \tag{2036}$$

where $\Delta_{\min}$ is the smallest distance between the cell center under consideration and the cell centers of the neighboring cells.

DES Model Coefficients

$C_{\mathrm{DES},k\text{-}\omega}$	$C_{\mathrm{DES},k\text{-}\varepsilon}$	C_{dt}	C_l	C_t
0.78	0.61	20	5	1.87

Function for Hybrid Upwind Scheme

Recognizing the superiority of an upwind scheme for RANS computations and the benefits of central differencing for LES, Travin and others [374] proposed the following blending coefficient for DES:

$$\sigma = \tanh\left\{\left[\min\left(A, 10\right)\right]^3\right\} \tag{2037}$$

where:

$$A = \max\left(\frac{l_{\mathrm{ratio}}}{\max\left(g, 10^{-20}\right)}-0.5, 0\right) \tag{2038}$$

$$l_{\mathrm{ratio}} = C_{\mathrm{DES}}\Delta\sqrt{\frac{\rho C_\mu^{1.5}\max\left(\widehat{S}, C_{\mathrm{desT}}/\tau\right)}{\mu+\mu_t}} \tag{2039}$$

$$g = \tanh\left\{\left[\min\left(B, 10\right)\right]^4\right\} \tag{2040}$$

$$B = \frac{2W\max(S, W)}{\max\left(\widehat{S}^2, 10^{-20}\right)} \tag{2041}$$

$$\widehat{S}^2 = \frac{W^2+S^2}{2} \tag{2042}$$

with the modulus of the mean vorticity tensor W and the modulus of the mean strain rate tensor S defined according to Eqn. (1746) and Eqn. (1744) respectively.

The local convective scale, τ is defined as:

$$\tau = \frac{\Delta}{|\mathbf{v}|} \tag{2043}$$

C_{desT} is a Model Coefficient and C_{DES} is given in Spalart-Allmaras DES—Model Coefficients.

Model Coefficient

C_{desT}
1

Boundary and Initial Conditions for Scale-Resolving Simulations

The specification of realistic inflow boundary and initial conditions is of paramount importance in Large Eddy and Detached Eddy Simulations of spatially developing flows.

As an inadequate amount of information introduces sources of error, a number of methods have been proposed for the specification of initialization and inflow, with varying degrees of complexity and computational cost.

The simplest approach consists of superimposing Gaussian noise to the mean velocity. However, this approach has the disadvantage of producing a decorrelated velocity field that is incapable of sustaining the growth of coherent turbulent structures.

In STAR-CCM+, the Synthetic Eddy Method (SEM) is implemented, which automatically provides turbulent eddies across inflow boundaries and initializes the flow domain with a perturbated flow field.

Contents:

Synthetic Eddy Method (SEM)

Synthetic Eddy Method (SEM)

The Synthetic Eddy Method (SEM) is a simple, yet effective, approach for generating the instantaneous velocity field that is required for LES and DES.

The Synthetic Eddy Method, proposed by Jarrin et al. [377], retains the conceptual basis of the vortex method [375], but is more flexible and virtually mesh-independent.

Within the SEM framework, the turbulent flow field is seen as a superposition of spinning eddies, whose spin (clockwise or counterclockwise), and position, are drawn from a normalized uniform distribution. The size of the eddies is the characteristic scale of turbulence. As eddies must span at least two cells in order to produce a correlated signal, the minimum mesh-spacing imposes a limit on this turbulence scale.

A user-assigned Reynolds stress tensor provides the correlation function that the method requires. If this stress tensor is not known, the correlation function uses only normal components that are directly derived from a given turbulence intensity (assumption of isotropy).

The synthetic eddies generated at the inflow are convected and recycled in the computational domain with the mean inflow velocity.

The turbulent structures that SEM produces are only an approximation of real turbulence, and must be allowed to develop the proper correlations. The correlations develop naturally as the eddies are convected downstream. To allow this behavior to occur, it is important to leave sufficient distance between the inflow boundary and the region of interest.

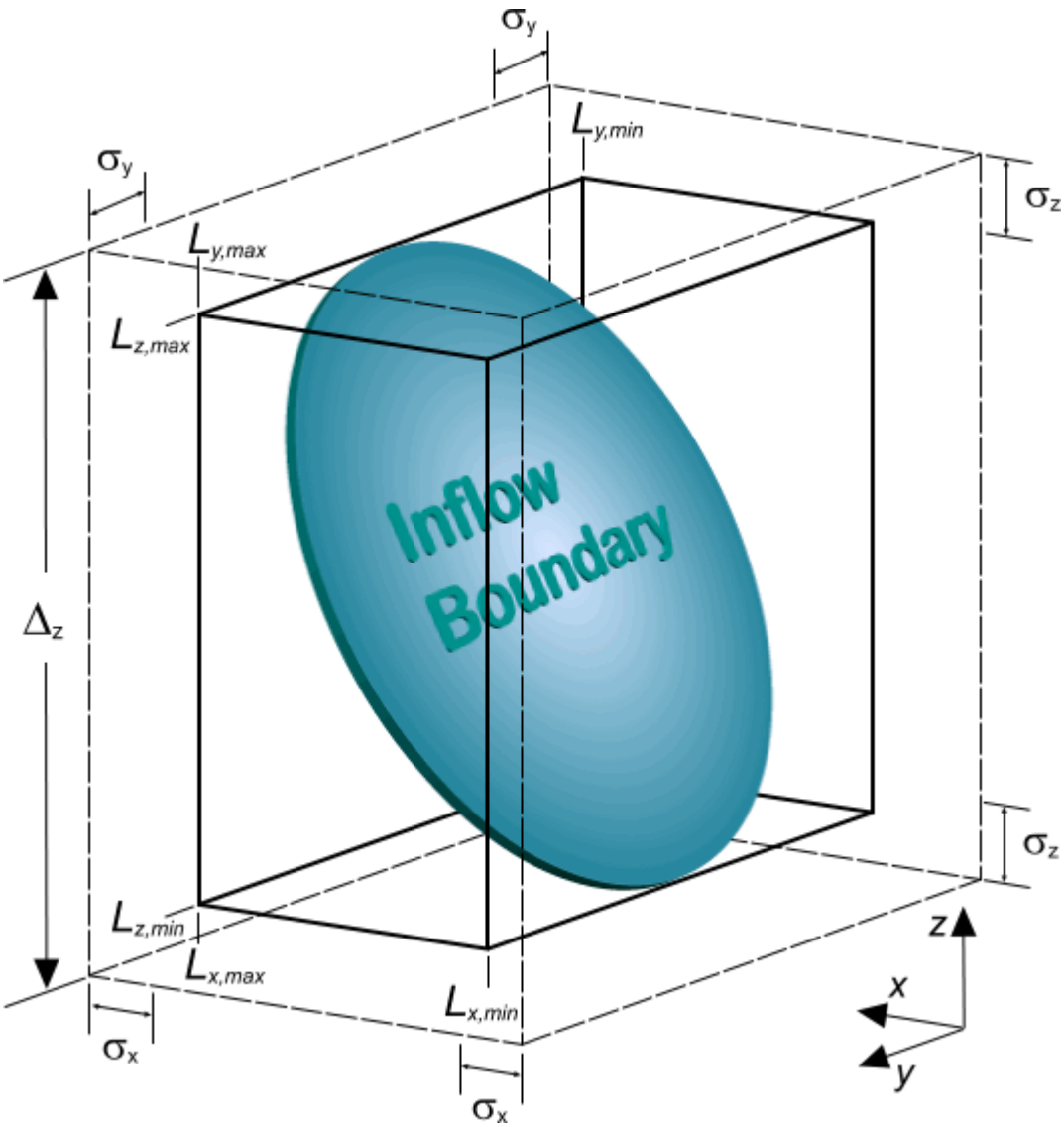
The fluctuating velocity signal is generated on an interval

$$\left[L_{x,min}, L_{x,max}\right] \times \left[L_{y,min}, L_{y,max}\right] \times \left[L_{z,min}, L_{z,max}\right] \tag{2044}$$

The signal uses a shape in:

$$\left[-\sigma_x, +\sigma_x\right] \times \left[-\sigma_y, +\sigma_y\right] \times \left[-\sigma_z, +\sigma_z\right] \tag{2045}$$

The interval limits are based on the geometric limits of the inflow section, as shown in the figure below, and σ_i is the mean eddy length scale in coordinate direction i.



The shape function $\mathbf{f}(\mathbf{x})$ of the eddies satisfies the following normalization property:

$$\frac{1}{\Delta_x \Delta_y \Delta_z} \int_{-\Delta_x/2}^{+\Delta_x/2} \int_{-\Delta_y/2}^{+\Delta_y/2} \int_{-\Delta_z/2}^{+\Delta_z/2} \left\| \mathbf{f}(\mathbf{x}) \right\| \, \mathrm{d}\, \mathbf{x} = \mathbf{1} \tag{2046}$$

where:

$$\Delta_i = L_{i,max} - L_{i,min} + 2\sigma_i \tag{2047}$$

and i is one of the three coordinate directions. Any function satisfying Eqn. (2046) is suitable to be taken as a shape function. In STAR-CCM+, the following tent function is used to represent the influence the k-th eddy exerts at position $\mathbf{x}_k$ on a point $\mathbf{x}$ on the inflow surface:

$$f(x_i) = \max\left[0, 1 - \frac{|x_i - x_{i,k}|}{\sigma_{i,k}}\right] \tag{2048}$$

where:

- x_i is the coordinate of point $\mathbf{x}$ in the i-th direction.
- $x_{i,k}$ is the coordinate of the center of eddy k in the i-th direction.

Each eddy therefore has a position $\mathbf{x}_k$, a length scale $\boldsymbol{\sigma}_k$, and a rotation (spin) direction which is given as:

$$\varepsilon_{k,i} = \pm 1 \tag{2049}$$

As shown in [376] the sign of the rotation is chosen randomly from a uniform distribution to ensure that the generated velocity signal satisfies:

$$\overline{v_i' v_j'} = \delta_{ij} \tag{2050}$$

The overbar denotes ensemble averaging and δ_{ij} is the Kronecker delta.

The eddy turbulent length scale $\boldsymbol{\sigma}_k$ can, in principle, be a function of space and vary according to the coordinate direction. In the current implementation, however, the turbulent length scale is assumed to be isotropic: while it may vary according to position in space, it must have the same value in each coordinate direction at any one point. The reason for this is that preliminary studies have shown that, for increasing values of anisotropy in the turbulent length scale, there is a progressively larger departure from the analytical normalization properties of the shape function. This results in distortion of the instantaneous velocity signal, which fails to recover the prescribed moments. This area is currently the subject of ongoing research.

The coordinates of the centers of eddies are chosen randomly in a volume $\Delta_x \times \Delta_y \times \Delta_z$, where Eqn. (2047) provides each dimension. The number of eddies, N, is obtained from the ratio of this volume to the average eddy volume, computed as $\overline{\sigma}^3$, where $\overline{\sigma}$ is the average eddy length scale.

Invoking the assumption of frozen turbulence, the position of each eddy k is advanced in each coordinate direction i according to:

$$x_{i,k}^{new} = x_{i,k}^{old} + \Delta t V_i \tag{2051}$$

where Δt is the time interval.

The convective speed V is obtained as an average over the inflow area of the mean inflow velocity profile. The fluctuating velocity signal is therefore obtained as:

$$v_i'(\mathbf{x}, t) = \frac{1}{\sqrt{N}} \sum_{k=1}^{N} \varepsilon_{i,k}\, f_k(\boldsymbol{\sigma}_k, \mathbf{x}, t) \tag{2052}$$

The instantaneous velocity field is computed as:

$$v_i(\mathbf{x}, t) = V_i(\mathbf{x}, t) + \underbrace{a_{ij} v_i'(\mathbf{x}, t)}_{v_i''} \tag{2053}$$

The factors a_{ij} are obtained from the Cholesky decomposition of the Reynolds stress tensor as:

$$\begin{pmatrix} \sqrt{R_{11}} & 0 & 0 \\ R_{21}/a_{11} & \sqrt{R_{22}-a_{21}^2} & 0 \\ R_{31}-a_{11} & \left(R_{32}-a_{21}a_{31}\right)/a_{22} & \sqrt{R_{33}-a_{31}^2-a_{32}^2} \end{pmatrix} \tag{2054}$$

where:

$$R_{ij} = \overline{v_i'' v_j''} \tag{2055}$$

The assumption of isotropy is invoked when turbulence intensity is provided at the inlet, rather than the individual components of the stress tensor. In this case, the normal components of the Reynolds stress tensor are calculated from the intensity and mean inflow velocity according to the following relations:

$$I = \sqrt{\frac{\mathrm{v}''^2}{V^2}}$$

$$k = \frac{3}{2}\mathrm{v}''^2$$

$$V^2 = V_i V_i$$

which results in:

$$R_{ii} = I^2 V^2 \tag{2056}$$

Turbulent Time and Length Scales

Certain aeroacoustics, combustion, Eulerian, and Lagrangian models require the computation of turbulent time and length scales. The Kolmogorov and Taylor scales provide information about the local limits for LES/DES meshes.

The following turbulent time and length scales are provided to these models depending on the respective turbulence model:

Turbulence Model	Turbulent Time Scale τ	Turbulent Length Scale l	Kolmogorov Length Scale η	Kolmogorov Time Scale τ_η	Taylor Micro Scale λ
Spalart-Allmaras	$(1-f_{v2})\min\left(\frac{2}{\tilde{S}}, \frac{\rho\kappa d^2}{\mu_t}\right)$ (2057)	$\frac{1}{2}(1-f_{v2})\sqrt{\tau\frac{\mu_t}{\rho C_\mu}}$ (2058)	-	-	-
K-Epsilon	$\frac{k}{\varepsilon}$ (2059)	$\tau k^{1/2}$ (2060)	$\left(\frac{\nu^3}{\varepsilon}\right)^{1/4}$ (2061)	$\left(\frac{\nu}{\varepsilon}\right)^{1/2}$ (2062)	$\sqrt{10\nu\frac{k}{\varepsilon}}$ (2063)
K-Omega	$(\beta^*\omega)^{-1}$ (2064)	$\tau k^{1/2}$ (2065)	$\left(\frac{\nu^3}{\beta^*\omega k}\right)^{1/4}$ (2066)	$\left(\frac{\nu}{\beta^*\omega k}\right)^{1/2}$ (2067)	$\sqrt{10\nu\frac{1}{\beta^*\omega}}$ (2068)
RST	-	-	$\left(\frac{\nu^3}{\varepsilon}\right)^{1/4}$ (2069)	$\left(\frac{\nu}{\varepsilon}\right)^{1/2}$ (2070)	$\sqrt{10\nu\frac{k}{\varepsilon}}$ (2071)
LES	$\frac{C_t}{S}$ (2072)	$\tau\sqrt{C_t\frac{\mu_t}{\rho}S}$ (2073)	-	-	-
DES	As for the respective RANS model.		-	-	-

where:

- f_{v2} is a Spalart-Allmaras Damping Function.
- $\tilde{S}$ is given by Eqn. (1766).
- ρ is the density.
- κ is the von Karman constant. See Spalart-Allmaras Model—Model Coefficients.
- d is the distance to the wall.
- μ_t is the turbulent eddy viscosity.

- C_μ is a Model Coefficient.
- k is the turbulent kinetic energy.
- ε is the turbulent dissipation rate.
- ν is the kinematic viscosity.
- β^* is given in K-Omega Model—Model Coefficients.
- ω is the specific dissipation rate.
- C_t is a Model Coefficient and must be greater than $(\sqrt{3})/2$ to satisfy realizability constraints [367].
- S is given by Eqn. (1744) and computed from the resolved velocity field $\tilde{\mathbf{v}}$.

For LES, the relation for τ is obtained from the ratio of the subgrid scale turbulent kinetic energy k_{SGS} and its dissipation rate ε_{SGS}. When computing this ratio, the subgrid scale turbulent kinetic energy is modeled as:

$$k_{SGS} = C_t \frac{\mu_t}{\rho} S \tag{2074}$$

The dissipation rate ε_{SGS} is modeled from equilibrium considerations as:

$$\varepsilon_{SGS} = \frac{\mu_t}{\rho} S^2 \tag{2075}$$

The turbulent lenght scale satisfies $l = \sqrt{k_{SGS}}\tau$.

Model Coefficients

C_t	C_μ
3.5	0.09

Turbulence Bibliography

Spalart-Allmaras Turbulence

[310] *Baldwin, B.S. and Lomax, H. 1978. "Thin-Layer Approximation and Algebraic Model for Separated Turbulent Flows", AIAA-78-0257.*

[311] *Dacles-Mariani, J., Zilliac, G., Chow, J.S., and Bradshaw, P. 1995. "Numerical/experimental study of a wingtip vortex in the near field", AIAA Journal, 33(9), pp. 1561-1568.*

[312] *Johnson, D.A. and King, L.S. 1985. "A Mathematically Simple Turbulence Closure Model for Attached and Separated Turbulent Boundary Layers", AIAA Journal, 23(11), pp. 1684-1692*

[313] *Shur, M. L., Strelets, M. K., Travin, A. K., and Spalart, P. R. 2000. "Turbulence Modeling in Rotating and Curved Channels: Assessing the Spalart-Shur Correction", AIAA Journal, 38(5), pp. 784-792.*

[314] *Spalart, P.R., and Allmaras, S.R. 1992. "A one-equation turbulence model for aerodynamic flows", AIAA-92-0439.*

[315] *Wilcox, D.C. 1998. "Turbulence Modeling for CFD". 2nd edition, DCW Industries, Inc.*

K-Epsilon Turbulence

[316] *Abe, K., Kondoh, T., and Nagano, Y. 1994. "A new turbulence model for predicting fluid flow and heat transfer in separating and reattaching flows --- 1. Flow field calculations", Int. J. Heat Mass Transfer, 37, pp. 139-151.*

[317] *Durbin, P.A. 1991. "Near-wall turbulence closure modeling without damping functions", Theoretical and Computational Fluid Dynamics, 3, pp. 1-13.*

[318] *Henkes, R.A.W.M., van der Flugt, F.F., and Hoogendoorn, C.J. 1991. "Natural convection in a square cavity calculated with low-Reynolds number turbulence models", Int. J. Heat Mass Transfer, 34, pp. 1543-1557.*

[319] *Jones, W.P., and Launder, B.E. 1972. "The Prediction of Laminarization with a Two-Equation Model of Turbulence", Int. J. Heat and Mass Transfer, 15, pp. 301-314.*

[320] *Jongen, T. 1998. "Simulation and Modeling of Turbulent Incompressible Flows", Ph.D. Thesis, Lausanne EPFL.*

[321] *Launder, B.E., and Sharma, B.I. 1974. "Application of the Energy Dissipation Model of Turbulence to the Calculation of Flow Near a Spinning Disc", Letter in Heat and Mass Transfer, vol. 1, no. 2, pp 131-138.*

[322] *Lien, F.S., Chen, W.L., and Leschziner, M.A. 1996. "Low-Reynolds number eddy-viscosity modelling based on non-linear stress-strain/vorticity relations", Proc. 3rd Symp. on Engineering Turbulence Modelling and Measurements, 27-29 May, Crete, Greece.*

[323] *Norris, L.H., and Reynolds, W.C. 1975. “Turbulent channel flow with a moving wavy boundary”, Report No. FM–10, Department of Mechanical Engineering, Stanford University, USA.*

[324] *Rodi, W. 1991. “Experience with Two-Layer Models Combining the k-e Model with a One-Equation Model Near the Wall”, 29th Aerospace Sciences Meeting, January 7-10, Reno, NV, AIAA 91-0216.*

[325] *Sarkar, S., and Balakrishnan, L. 1990. “Application of a Reynolds-stress turbulence model to the compressible shear layer”, ICASE Report 90-18, NASA CR 182002.*

[326] *Shih, T.-H., Liou, W.W., Shabbir, A., Yang, Z. and Zhu, J. 1994. “A New k-ε Eddy Viscosity Model for High Reynolds Number Turbulent Flows -- Model Development and Validation”, NASA TM 106721.*

[327] *Spalart, P.R. and Rumsey, C.L. 2007. “Effective Inflow Conditions for Turbulence Models in Aerodynamic Calculations”, AIAA Journal, 45(10), pp. 2544-2553.*

[328] *Wolfstein, M. 1969. “The velocity and temperature distribution in one-dimensional flow with turbulence augmentation and pressure gradient”, Int. J. Heat Mass Transfer, 12, pp. 301-318.*

[329] *Xu, W., Chen, Q., and Nieuwstadt, F.T.M. 1998. “A new turbulence model for near-wall natural convection”, Int. J. Heat Mass Transfer, 41, pp. 3161-3176.*

[330] *Yap, C. 1987. “Turbulent heat and momentum transfer in recirculating and impinging flows”, Ph.D. Thesis, Faculty of Technology, University of Manchester, U.K.*

K-Omega Turbulence

[331] *Arolla, S. K., and Durbin, P. A. 2012. “Modeling rotation and curvature effects within scalar eddy viscosity model framework”, publication pending.*

[332] *Durbin, P.A. 1996. “On the k-e stagnation point anomaly”, Int. J. Heat and Fluid Flow 17, pp. 89-90.*

[333] *Hellsten, A. 2005. "New Advanced k-omega Turbulence Model for High-Lift Aerodynamics", AIAA Journal, Vol.43, No. 9, pp. 1857-1869.*

[334] *Menter, F.R. 1994. “Two-equation eddy-viscosity turbulence modeling for engineering applications”, AIAA Journal, 32(8), pp. 1598-1605.*

[335] *Sarkar, S., Erlebacher, G., Hussaini, M. Y., and Kreiss, H. O., "The analysis and modelling of dilatational terms in compressible turbulence", Journal of Fluid Mechanics , Vol.227, pp. 473-493.*

[336] *Spalart, P.R. 2000. "Strategies for turbulence modelling and simulations", Int. J. of Heat and Fluid Flow, Vol.21(3), pp. 252-263.*

[337] *Spalart, P.R. and Rumsey, C.L. 2007. “Effective Inflow Conditions for Turbulence Models in Aerodynamic Calculations”, AIAA Journal, 45(10), pp. 2544-2553.*

[338] *Wallin, S. and Johansson, A. V. 2000. "An Explicit Algebraic Reynolds Stress Model for Incompressible and Compressible Turbulent Flows," J. Fluid Mechanics, Vol. 403, pp. 89-132.*

[339] *Wilcox, D.C. 1998. “Turbulence Modeling for CFD”, 2nd edition, DCW Industries, Inc.*

[340] *Wilcox, D.C. 2008. “Formulation of the k-ω turbulence model revisited”, AIAA Journal, 46(11), pp. 2823-2838.*

Elliptic Blending Turbulence

[341] *Billard, F. and Laurence, D. 2012. “A robust $k-\varepsilon$-v^2/k elliptic blending turbulence model applied to near-wall, separated and buoyant flows”, International Journal of Heat and Fluid Flow 33 (1), pp. 45–58.*

[342] *Durbin, P.A. 1993. “A Reynolds stress model for near-wall turbulence”, Journal of Fluid Mechanics 249 (1), pp 465–498.*

[343] *Durbin, P. A., Medic, G., Seo, J. M., Eaton, J. K., and Song, S. 2001. "Rough Wall Modification of 2-layer k-epsilon" , ASME Journal of Fluids Engineering, 123, pp. 16-21.*

[344] *Knopp, T., Eisfeld, B., and Calvo, J. B., 2009. "A new extension for k-omega turbulence models to account for wall roughness", International Journal of Head and Fluid Flow, 30, pp. 54-65.*

[345] *Manceau, R. and Hanjalić, K. 2002. “Elliptic blending model: A new near-wall Reynolds-stress turbulence closure” Physics of Fluids 14, pp 744.*

Lag Elliptic Blending Turbulence

[346] *Lardeau, S., Billard, F. 2016. "Development of an elliptic-blending lag model for industrial applications", 54th AIAA Aerospace Sciences Meeting, p.1600.*

[347] *Revell, A.J., Benhamadouche, S., Craft, T., Laurence, D. 2006. " A stress-strain lag eddy viscosity model for unsteady mean flow." International Journal of Heat and Fluid Flow 27 (5), pp. 821-830.*

V2F Turbulence

[348] *Davidson, L., Nielsen, P.V., and Sveningsson, A. 2003. “Modifications of the $\overline{v^2}-f$ Model for Computing the Flow in a 3D Wall Jet”, Turbulence, Heat and Mass Transfer, 4, pp. 577-584.*

[349] *Durbin, P. A. 1996, “On the k-e stagnation point anomaly”, Int. J. Heat and Fluid Flow 17, 89-90.*

[350] *Lien, L. S., Kalitzin, G., Durbin, P. A. 1998. “RANS modeling for compressible and transitional flows”, Center for Turbulence Research - Proceedings of the Summer Program 1998*

Reynolds Stress Transport Turbulence

[351] *Daly, B.J. and Harlow, F.H. 1970. “Transport equations of turbulence”, Physics of Fluids 13, pp. 2634-2649.*

[352] *Gibson, M.M. and Launder, B.E. 1978. “Ground effects on pressure fluctuations in the atmospheric boundary layer”, J. Fluid Mech., 86(3), pp. 491-511.*

[353] *Hadzic, I. 1999. “Second-moment closure modeling of transitional and unsteady turbulent flows”, doctoral thesis, Technical University of Delft.*

[354] *Lardeau, S. and Manceau, R. 2014. “Computations of Complex Flow Configurations Using a Modified Elliptic-Blending Reynolds-Stress Model”, Proc. of the 10th Eng. Turb. Modelling and Measurement Conf., Marbella, Spain.*

[355] *Launder, B.E. and Shima, N. 1989. “Second Moment Closure for the Near-Wall Sublayer: Development and Application”, AIAA J. 27(10), October, pp. 1319-1325.*

[356] *Lien, F.S. and Leschziner, M.A. 1994. “Assessment of turbulence-transport models including non-linear RNG eddy-viscosity formulation and second-moment closure for flow over a backward-facing step”, Computers Fluids, 23(8), pp. 983-1004.*

[357] *Sarkar, S., and Balakrishnan, L. 1990. “Application of a Reynolds-stress turbulence model to the compressible shear layer”, ICASE Report 90-18, NASA CR 182002.*

[358] *Speziale, C.G., Sarkar, S., and Gatski, T.B. 1991. “Modelling the pressure-strain correlation of turbulence: an invariant dynamical systems approach”, J. Fluid Mech., 227, pp. 245-272.*

Large Eddy Simulation

[359] *Addad, Y., Gaitonde, U., Laurence, D., and Rolfo, S. 2008. " Optimal Unstructured Meshing for Large Eddy Simulations" , Quality and Reliability of Large-Edy Simulations (12), pp. 93-103, Springer.*

[360] *Balaras, E., Benocci, C., and Piomelli, U. 1996. “Two-Layer Approximate Boundary Conditions for Large-Eddy Simulations,” AIAA Journal 34(6), pp. 1111-1119.*

[361] *Germano, M., Piomelli, U., Moin, P., and Cabot, W. H. 1991. “A Dynamic Subgrid-Scale Eddy Viscosity Model”, Physics of Fluids A, 3(7), pp. 1760-1765.*

[362] *Lilly, D. K. 1992. “A proposed modification of the Germano subgrid-scale closure method”, Physics of Fluids A 4(3), pp. 633-635.*

[363] *Nicoud, F. and Ducros, F., 1999. “Subgrid-Scale Stress Modelling Based on the Square of the Velocity Gradient Tensor,” Flow, Turbulence and Combustion, 62, pp. 183-200.*

[364] *Smagorinsky, J. 1963. “General Circulation Experiments with the Primitive Equations: Part I, The Basic Experiment,” Monthly Weather Review, 91, pp. 99-164.*

[365] *Tucker, P. G. 2016. "Advanced Computational Fluid Dynamics", Cambridge University Press*

[366] *Van Driest, E. R. 1956. “On Turbulent Flow Near a Wall,” J. Aero. Sci., pp. 1007-1011.*

[367] *Vreman, B., Geurts, B., and Kuerten, H. 1994. “Realizability conditions for the turbulent stress tensor in Large Eddy Simulation,” J. Fluid Mech., 278, pp. 351–362.*

Detached Eddy Simulation

[368] *Constantinescu, G.S., Pacheco, R., and Squires, K.D. 2002. "Detached-eddy simulation of flow over a sphere", AIAA Aerospace Sciences Meeting 2002, Reno, Nevada, USA, Paper AIAA 2002-0425.*

[369] *Menter, F. R. and Kuntz, M. 2002. "Adaptation of Eddy Viscosity Turbulence Models to Unsteady Separated Flows Behind Vehicles", in The Aerodynamics of Heavy Vehicles: Trucks, Buses and Trains, Springer, Asilomar, CA.*

[370] *Shur, M.L., Spalart, P.R., Strelets, M.Kh., and Travin, A.K. 2008. "A hybrid RANS-LES approach with delayed-DES and wall-modelled LES capabilities", International J. Heat and Fluid Flow, 29(6), pp. 1638-1649.*

[371] *Spalart, P.R., Jou., W.H., Strelets, M. and Allmaras, S.R. 1997. "Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach", 1st AFOSR Int. Conf. on DNS/LES Aug, 4-8 1997. In: Advances in DNS/LES, C. Liu and Z. Liu Eds., Greyden Press, Columbus, OH.*

[372] *Spalart, P.R., Deck, S., Shur, M.L., Squires, K.D., Strelets. M. and Travin, A. 2006. "A new version of detached eddy simulation, resistant to ambiguous grid densities", Theor. Comput. Fluid Dynamics, 20, pp. 181-195.*

[373] *Squires, K. D., Forsythe, J. R., Strang, W. Z., Wurtsler, K. E., Tomaro, R.F., Grismer, M. J. and Spalart, P.R. 2002. "Progress on detached-eddy simulation of massively separated flows", AIAA 2002-1021.*

[374] *Travin, A., Shur, M. L., Strelets, M. Kh., and Spalart, P. R. 2002. "Physical and numerical upgrades in the Detached-Eddy Simulation of complex turbulent flows", in Advances in LES of Complex Flows, R. Friedrich and W. Rodi Eds., Kluwer Academic Press, pp. 239-254.*

SEM

[375] *Benamadouche, S., Jarrin, N., Addad, Y., and Laurence, D. 2006. "Synthetic turbulent inflow conditions based on a vortex-method for large eddy simulation", Progress in Computational Fluid Dynamics, 6, pp. 50-57.*

[376] *Jarrin, N. 2008. "Synthetic inflow boundary conditions for the numerical simulation of turbulence", Ph.D. Thesis, University of Manchester (UK).*

[377] *Jarrin, N., Benamadouche, S., Laurence, D., and Prosser, R. 2006. "A synthetic-eddy-method for generating inflow conditions for large eddy simulations". International Journal of Heat and Fluid Flow, 27, pp. 585-593.*

Temperature Flux

[378] *Kenjeres, S., Gunarjo, S.B., and Hanjalic, K. 2005. "Contribution to elliptic relaxation modelling of turbulent natural and mixed convection", Int. J. Heat and Fluid Flow, 26(4), pp. 569–586.*

~Transition~

For this release, see *User Guide* > *Modeling Physics* > *Modeling Transition* > *Transition Formulation*.

Theory guide content is planned for a future release.

Heat Transfer

Heat transfer is the exchange of thermal energy between media at different temperatures. Heat transfers from locations of high temperature to locations of low temperature in order to reach an equilibrium state. The three main mechanisms of heat transfer are: conduction, convection, and radiation.

In STAR-CCM+, heat transfer can be calculated within a fluid (single or multi-component), between different fluid streams, between a fluid and a solid, and within a solid. The coupled heat transfer within a fluid and an adjoining solid is called conjugate heat transfer. For a conjugate heat transfer analysis, the energy equation is solved throughout the fluid and the solid solution domain with an efficient implicit thermal coupling at the fluid/solid interface. All other conservation equations are solved within the fluid only.

Energy Equation in Fluids

STAR-CCM+ implements the energy equation in the following integral form:

$$\frac{\partial}{\partial t}\int_V \rho E dV + \oint_A \rho H \mathbf{v}\cdot d\mathbf{a} = -\underbrace{\oint_A \dot{\mathbf{q}}''\cdot d\mathbf{a}}_{\text{Conduction}} + \underbrace{\oint_A \mathbf{T}\cdot\mathbf{v} d\mathbf{a}}_{\text{Viscous Work}} + \int_V \mathbf{f}_b\cdot\mathbf{v} dV + \int_V S_u dV \tag{2076}$$

where:

E is the total energy
H is the total enthalpy
$\dot{\mathbf{q}}''$ is the heat flux vector
$\mathbf{T}$ is the viscous stress tensor
$\mathbf{v}$ is the velocity vector
$\mathbf{f}_b$ is the body force vector representing the combined body forces of rotation, gravity, and the fan mode
S_u contributes energy source terms, such as radiation sources, interphase energy sources, or energy sources due to chemical reactions. User-defined volumetric sources also enter through this term.

Total energy is related to the total enthalpy H by:

$$E = H - p/\rho \tag{2077}$$

where:

$$H = h + |\mathbf{v}|^2/2 \tag{2078}$$

and h is the static enthalpy.

Energy Equation in Solids

The governing equation for energy transport within a solid is given as follows:

$$\frac{d}{dt}\int_V \rho C_p T dV + \oint_A \rho C_p T \mathbf{v}_s \cdot d\mathbf{a} = -\oint_A \dot{\mathbf{q}}'' \cdot d\mathbf{a} + \int_V S_u dV \tag{2079}$$

where:

ρ is the solid density
C_p is the specific heat
T is the temperature
$\dot{\mathbf{q}}''$ is the heat flux vector
$\mathbf{v}_s$ is the solid convective velocity (only applicable when the region is a pure body of rotation)
S_u is a user-defined volumetric heat source within the solid

Contents:

Conduction
Convection
Thermal Radiation
Solar Radiation
Heat Transfer Bibliography

Conduction

Heat conduction is the transfer of internal energy through interactions between the microscopic particles that compose a conducting substance. Energy flows in the direction of high temperature to low temperature as those particles in high temperature regions are more energetic than their low temperature neighbors.

Conduction takes place in all forms of matter: solids, liquids, gases, and plasmas. In solids, heat conduction is occurs due to the vibrations of molecules in a lattice in conjunction with diffusion of free electrons. In gases and liquids, heat conduction is due to collisions between molecules and the diffusion. Conduction is greater in solids because of the relatively close spatial relationships between atoms, whereas fluids and gases are characterized by larger distances between atoms.

The law of heat conduction, also known as Fourier's law, provides a means of calculating the local heat flux. This law can be stated in two equivalent forms:

- The integral form, which considers the amount of energy flowing into or out of a body as a whole.
- The differential form, which considers the flow rates or fluxes of energy locally:

$$\dot{\mathbf{q}}'' = -k\nabla T \tag{2080}$$

  In this expression, $\dot{\mathbf{q}}''$ $\left[W/m^2\right]$ is the local heat flux vector, k $[W/mK]$ is the thermal conductivity of the material, and ∇T $[K/m]$is the temperature gradient. The thermal conductivity of the material can be a strong function of temperature, pressure (for gases), spatial location, and can exhibit anisotropic behavior (that is, it varies with orientation and is represented by a second-order tensor). The negative sign in the expression is required because k is greater than zero, yet heat always moves from a high temperature region to a low temperature region.

Thermal conductivities vary from material to material. They are largest for metallic solids, smaller for nonmetallic solids, very small for liquids, and extremely small for gases.

The major objective of performing a simulation that includes conductive heat transfer is to determine the temperature field in a medium that satisfies the boundary conditions. The First Law of Thermodynamics (that is, Conservation of Energy) and Fourier's law can be used to derive the Heat Diffusion Equation (a second-order partial differential equation in temperature, or enthalpy, or energy) from which the temperature field can be calculated. This temperature solution can be steady (a function of space only and not changing with time), or unsteady (the temperature field varies in both space and time as, for example, in metal quenching applications).

Convection

Convective heat transfer is the transfer of thermal energy by the combined effects of random molecular motion (diffusion) within the fluid, and the overall movement of the fluid from one place to another.. In engineering practice, it is used to provide specific temperature changes.

In engineering practice, convection heat transfer between a fluid in motion and a bounding surface is of particular interest. Near the surface, where fluid velocity is zero, heat transfer occurs by diffusion only. Moving away from the surface, heat is transported downstream by the fluid motion and passes into the bulk fluid flow.

Convection is usually described as being either natural, where natural buoyancy forces drive the bulk fluid motion, or forced, where an external driving force moves the fluid.

Convective heat transfer at a surface is governed by Newton's law of cooling:

$$\dot{q}_s'' = h\left(T_s - T_{ref}\right) \tag{2081}$$

where:

$\dot{q}_s''$ is the local surface heat flux (that is, power per unit area)

h is the local convective heat transfer coefficient

T_s is the surface temperature

T_{ref} is a characteristic temperature of the fluid moving over the surface

Newton's law of cooling expresses a linear relationship between the local surface heat flux and the difference between the local surface and fluid temperatures. This linear relationship is only an approximation: in reality the relationship can be strongly nonlinear. Because flow conditions can vary from point to point on the surface, both q_s'' and h can vary as a function of space and time.

Natural Convection

In a gravitational field, natural convection occurs due to temperature differences which affect the density, and thus the relative buoyancy, of the fluid. Denser components fall, while lighter (less dense) components rise, leading to bulk fluid movement.

Natural convection is more likely and more rapid when any of the following conditions apply:

- a greater variation in density between two mixing fluids
- a larger acceleration due to gravity that drives the convection
- a larger distance through the convecting medium.

Natural convection is less likely and/or less rapid when any of the following conditions apply:

- a more rapid diffusion (which acts to diffuse away the thermal gradient that causes the convection)
- a more viscous (sticky) fluid.

To quantify when natural convection is possible, the dimensionless Rayleigh number is defined and can be viewed as the ratio of buoyant and viscous forces times the ratio of momentum and thermal diffusivities.

Natural convection becomes possible when the Rayleigh number exceeds 10^5.

Forced Convection

In forced convection, fluid movement results from external sources (for example, a fan, pump, or the action of a propeller), and is typically used to increase the rate of heat transfer at a surface in cooling or heating applications.

In cases of mixed convection (that is, natural convection and forced convection occurring together) you can determine how much is due to forced convection and how much is due to natural convection. The relative magnitudes of the Rayleigh (Ra), Prandtl (Pr), and Reynolds (Re) numbers indicate which form of convection dominates:

If $\frac{Ra}{Pr\mathrm{Re}^2} \gg 1$, natural convection dominates, forced convection is neglected.

If $\frac{Ra}{Pr\mathrm{Re}^2} \ll 1$, forced convection dominates, natural convection is neglected.

If $\frac{Ra}{Pr\mathrm{Re}^2} \simeq 1$, mixed convection occurs, neither forced or free convection dominate, so both are taken into account.

Convective Heat Transfer Coefficients

In Eqn. (2081), $\dot{q}_s''$, T_s, and T_{ref} are fundamental in nature, while the heat transfer coefficient is a constant of proportionality that relates the three fundamental parameters. While $\dot{q}_s''$ and T_s are unambiguous, there is some latitude in the choice of the fluid temperature T_{ref}. Depending upon the choice of T_{ref}, the heat transfer coefficient is different in order to satisfy Eqn. (2081). Therefore, heat transfer coefficients cannot be defined without also defining a reference temperature: there is an infinite number of heat transfer coefficient and reference temperature combinations that give the same surface heat flux.

In STAR-CCM+, the conceptual centerpiece for modeling convective heat transfer at the wall for turbulent flows stems from the standard wall functions (SWF). Consider the expression for the local surface heat flux:

$$q_s'' = \frac{\rho_f(y_c)C_{p,f}(y_c)u_\tau}{T^+(y^+(y_c))}(T_s - T_c) \tag{2082}$$

You can then equate the terms from Eqn. (2081) and Eqn. (2082) and define an h in terms of the local flow conditions.

In this equation:

ρ_f is the fluid density

$C_{p,f}$ is the fluid-specific heat capacity

u_τ is a velocity scale that is based on the wall shear stress

T^+ is the dimensionless temperature

$y^+ = (u_\tau y_c)/\upsilon_f$ is a Reynolds number

y_c and T_c are the normal distance and temperature of the near-wall cell, respectively

The Standard Wall Functions (SWFs) are a set of semi-empirical functions that are used to satisfy the flow physics in the near-wall region (the boundary layer). They give relationships for T^+ and u_τ in terms of the

laminar and turbulent Prandtl numbers, the dimensionless near-wall flow velocity, and the turbulent kinetic energy.

Four heat transfer coefficients are defined in STAR-CCM+:

- Local Heat Transfer Coefficient
- Specified y+ Heat Transfer Coefficient
- Heat Transfer Coefficient
- Virtual Local Heat Transfer Coefficient

These heat transfer coefficients are all grounded to varying degrees in the standard wall function treatment. The local heat transfer coefficient is the one that is used internally within the STAR-CCM+ solver. The other three heat transfer coefficients are computed during post-processing. In the case, of multi-phase flow, volume-fraction-weighted versions of these heat transfer coefficients and reference temperatures are defined for use with co-simulation or coupling to other CAE codes.

Local Heat Transfer Coefficient

In STAR-CCM+, the local surface heat transfer coefficient is always calculated (and used internally) in the code using the Standard Wall Functions and Eqn. (2084). However, there can be times when you want to view or export heat transfer coefficients that are defined in other ways. For example, to compare the results to previous work or to use with other applications. In general, you can accomplish this by post-processing the simulation results and defining the one-to-one linear mapping (based on equating the surface heat fluxes):

$$h_{user} = h\left(\frac{T_s - T_c}{T_s - T_{ref}}\right) \tag{2083}$$

In this way, you can define your own h and T_{ref} pair, where h, T_c, and T_s are known from the simulation, and you pick T_{ref} to suit your requirements.

If you equate the Local Heat Transfer Coefficient and Local Reference Temperature terms from Eqn. (2081) and Eqn. (2082):

$$h_1 = \frac{\rho_f(y_c)C_{p,f}(y_c)u_\tau}{T^+(y^+(y_c))} \tag{2084}$$

and $T_{ref,1} = T_c$

you have the pair (h_1 and $T_{ref,1}$) which can be viewed or exported.

Specified y+ Heat Transfer Coefficient

The first built-in post-processing option in STAR-CCM+ calculates the heat transfer coefficient as in Eqn. (2084), but at a user-specified y^+ value instead of the value that is associated with the near-wall cell:

$$h_2 = \frac{\rho_f(y_c)C_{p,f}(y_c)u_\tau}{T^+(y^+{}_{user})} \tag{2085}$$

Then, using the surface heat flux as calculated in Eqn. (2082), you solve Eqn. (2081) for the reference temperature:

$$T_{ref,2} = T_s - \frac{q_s''}{h_2} \tag{2086}$$

The result is the pair (h_2 and $T_{ref,2}$) which can be viewed or exported.

This definition of the reference temperature does not guarantee that $T_{ref,2}$ lies within the minimum and maximum temperature range of the local fluid region. If $T_{ref,2}$ lies outside this temperature range, then either the minimum or maximum temperature is used in its place, and the heat transfer coefficient is modified accordingly.

Heat Transfer Coefficient

The second built-in post-processing option in STAR-CCM+ calculates the surface heat flux using Eqn. (2082), and then recasts Eqn. (2081) as:

$$h_3 = \frac{q_s''}{T_s - T_{ref,3}} \tag{2087}$$

The heat transfer coefficient is calculated using a user-defined reference temperature ($T_{ref,3}$).

The result is the pair (h_3 and $T_{ref,3}$) which can be viewed or exported.

Virtual Local Heat Transfer Coefficient

The third built-in post-processing option in STAR-CCM+ does not need to solve the energy equation in the fluid.

Instead, this option uses the Local Heat Transfer Coefficient (Eqn. (2084)), which is calculated using only flow field information. You specify the specific heat, the molecular Prandtl number, and the turbulent Prandtl number as reference values for this heat transfer coefficient.

The result is the pair (h_1 and $T_{ref,4}$) which can be viewed or exported.

Thermal Radiation

Thermal radiation is the emission of electromagnetic waves from all matter that has a temperature greater than absolute zero, and represents a conversion of thermal energy into electromagnetic energy. The thermal motion of charged particles in matter results in charge-acceleration and dipole oscillation. This behavior drives the electrodynamic generation of coupled electric and magnetic fields, which cause the emission of thermal radiation.

Electromagnetic radiation, or light, does not require the presence of matter to propagate, and travels the fastest in a vacuum. All forms of matter emit radiation. For gases and some semi-transparent solids (for example, glass and salt crystals at elevated temperatures) emission is a volumetric phenomenon (that is, emission is an integrated effect of local emission throughout the volume). These volumetric effects are modeled with the Radiative Transport Equation (RTE) for Participating Media Eqn. (2117). For most solids and liquids, radiation that is emitted from the interior molecules is strongly absorbed by adjoining molecules. Therefore, because all of the emitted radiation originates from molecules near the surface, it can be considered as a surface phenomenon. If these surfaces form the boundaries of a participating medium, the effects are considered as boundary conditions for solution of the RTE. In the case where these surfaces surround a medium with relatively little volumetric emission, absorption, and scattering, the exchange of energy can be solved for using the surface-to-surface solver (see S2S (Surface-to-Surface) Radiation).

For most heat transfer applications, thermal radiation can be treated as unpolarized due to multiple reflections and scattering and incoherent (waves or photons are usually out of phase). The length scale for transport is usually much larger than the wavelength of radiation, so the limiting description of geometric optics (that is, the wavelength approaches zero) can be applied. Radiation can then be described in terms of rays carrying energy in specific directions.

The maximum flux at which radiation can be emitted from a surface is given by the Stefan-Boltzmann law:

$$\dot{q}_{bb}'' = \sigma T_s^{\,4} \tag{2088}$$

In this expression, $\dot{q}_{bb}''$ is the local surface heat flux, T_s is the temperature of the surface, and σ is the Stefan-Boltzmann constant $\sigma = 5.67 \times 10^{-8} W/m^2K^4$. Such a surface is called an ideal radiator or black body.

The heat flux that is emitted by a real surface is less than the heat flux of an ideal radiator, and is given by:

$$\mathrm{q}_e'' = \varepsilon q_{bb}'' = \varepsilon\sigma T_s^{\,4} \tag{2089}$$

where $0 \leq \varepsilon \leq 1$ is the surface emissivity (that is, the relative ability of a surface to emit energy by radiation).

If radiation is incident upon a surface, a portion is absorbed, a portion is reflected, and a portion is transmitted through the material:

$$\mathrm{q}_{inc}'' = \mathrm{q}_{abs}'' + \mathrm{q}_{ref}'' + \mathrm{q}_{trans}'' = \alpha \mathrm{q}_{inc}'' + \rho \mathrm{q}_{inc}'' + \tau \mathrm{q}_{inc}'' \tag{2090}$$

The constants in this expression are:

$0 \leq \alpha \leq 1$	The surface absorptivity: the fraction of incident radiation that gets absorbed.

$0 \leq \rho \leq 1$	The reflectivity: the fraction of incident radiation that gets reflected away from the surface.
$0 \leq \tau \leq 1$	The transmissivity: the fraction of incident radiation that gets transmitted through the material.

Absorptivity, reflectivity, and emissivity depend on the surface temperature, surface roughness, emission angle, and wavelength of the radiation (the radiation is not monochromatic, but comprises a continuous dispersion of photon energies).

Contents:

S2S (Surface-to-Surface) Radiation
Participating Media Radiation
Radiative Properties

S2S (Surface-to-Surface) Radiation

The Surface-to-Surface (S2S) radiation model in STAR-CCM+ allows you to analyze thermal radiative heat transfer between surfaces of arbitrary complexity that form enclosed spaces.

The medium that fills the space between the surfaces is non-participating, that is, it does not absorb, emit, or scatter radiation. The radiation properties and the thermal boundary conditions that are imposed on each surface define uniquely the amount of radiation that a surface receives and emits. The surface properties are quantified in terms of emissivity, reflectivity, transmissivity, and radiation temperature. These properties do not depend on direction, however, they can depend on radiation wavelength with the multiband radiation model. The S2S Radiation model is based on enclosure theory [379], [382].

The net radiant flux on each surface is a function of the surface properties, the thermal boundary conditions that are imposed on that surface, and is calculated so that radiation is balanced. The radiation balance is enforced on the entire closed set of surfaces, by considering each surface and how it exchanges radiation with all other surfaces.

The calculation approach consists of the following main elements:

- The boundaries enclosing the computational domain and internal boundaries are spatially subdivided into contiguous, non-overlapping patches.
- From the center of each patch, a specified number of beam is emitted over the enclosing hemisphere with solid angles that are discretized using an angular quadrature. Each beam is traced through the computational domain until it intercepts an opposing patch, thus defining a pair of patches that exchange radiative energy.
- The radiation energy transfer to or from each patch is then calculated from the radiation transport equation and the boundary conditions.

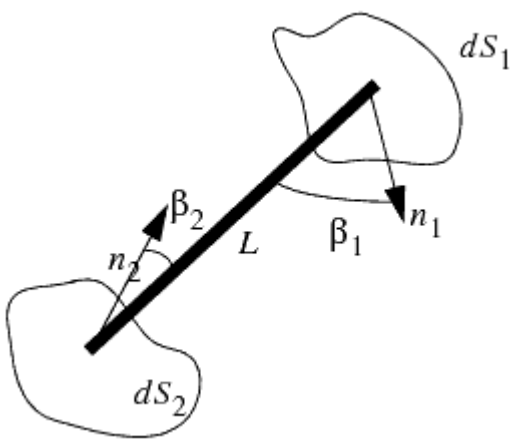
Surface Patches and View (Interaction) Factors

The surfaces in the model can have either diffuse or specular surface properties. The surface emissivity is assumed diffuse, but the reflectivity can be either diffuse or specular, and all transmissivity is specular. Specular properties are considered in the calculation of the view factors. Hence, the view factors are not merely geometric properties, but also include the influence of the specular surface properties.

The amount of radiation that is exchanged between surfaces depends on the (diffuse) emission from the surfaces and the position and orientation of the surfaces relative to each other. The amount of radiation also depends on the presence of any specularly transmissive or reflective surfaces that can alter the transfer between the surfaces. The latter two influences—geometry and specular properties—are accounted for by the view (or interaction) factors which, by definition, represent the "fraction of uniform diffuse radiation leaving a surface that directly reaches another surface" [379].

The following Eqn. (2091) to Eqn. (2096) define the view factor relationships for gray and diffuse surfaces.

The following diagram shows the radiation exchange between two elemental surfaces:



The total amount of radiation power that dS_1 emits and dS_2 receives is:

$$P_{1-2} = \mathrm{i}'_1 dS_1 \cos(\beta_1)\left(\frac{dS_2 \cos(\beta_2)}{L^2}\right) \tag{2091}$$

where:

P_{1-2} is the total power leaving surface 1 and incident on surface 2 [W]

i'_1 is the total intensity leaving surface 1, where intensity is defined as the "radiative energy passing through an area per unit solid angle, per unit of the area projected normal to the direction of passage, and per unit time" [379] [$\frac{W}{m^2 sr}$]

β is the angle between the surface normal and a line joining two surfaces [rad]

L is the distance

By taking the fraction of the power that dS_2 receives over the total radiation power that is emitted from dS_1 $(P_1 = \pi i'_1 dS_1)$, the view factor for the two surfaces is obtained as:

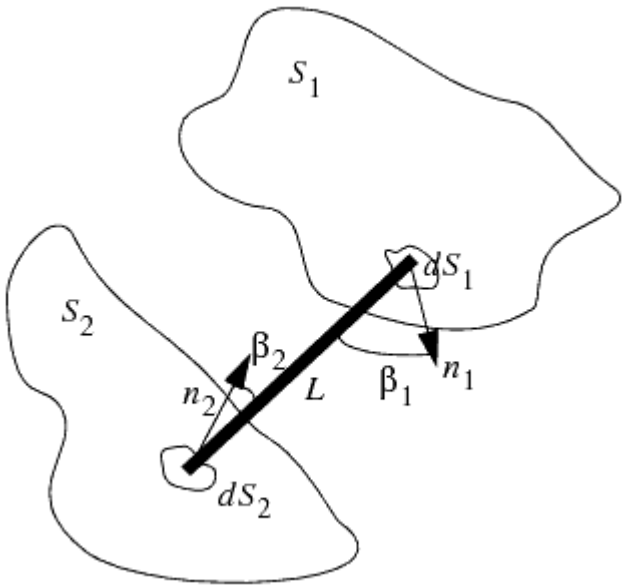
$$dF_{1-2} = \frac{P_{1-2}}{\pi i'_1 dS_1} \tag{2092}$$

Applying definition Eqn. (2092) to two black surfaces i and j in thermal equilibrium at the same temperature (no net energy exchange) yields the requirement that view factors must satisfy the reciprocity relation:

$$dF_{i-j}dS_i = dF_{j-1}dS_j \tag{2093}$$

For diffuse radiation, relationship Eqn. (2093) applies to any two surfaces, whether black or not.

These definitions can be extended to surfaces of finite size, for which the view factors are obtained using integration over constituent elemental surfaces. The following diagram shows the radiation exchange between two finite surfaces:



For the (S1, S2) pair:

$$F_{1-2} = \frac{\int_{S_1}\int_{S_2} i'_1 \frac{\cos(\beta_1)\cos(\beta_2)}{\pi L^2} dS_1 dS_2}{\int_{S_1} i'_1 dS_1} \tag{2094}$$

and (assuming i' is constant over element 1):

$$F_{1-2} = \frac{1}{S_1}\int_{S_1}\int_{S_2} \frac{\cos(\beta_1)\cos(\beta_2)}{\pi L^2} dS_1 dS_2 \tag{2095}$$

Likewise:

$$F_{2-1} = \frac{1}{S_2}\int_{S_2}\int_{S_1} \frac{\cos(\beta_1)\cos(\beta_2)}{\pi L^2} dS_1 dS_2 \quad \text{or} \quad F_{2-1} = F_{1-2}\frac{S_1}{S_2} \tag{2096}$$

In STAR-CCM+, the boundary surfaces are discretized into smaller elements called patches. These patches are sets of contiguous boundary cell faces, and view factors are calculated for each patch pair. By definition, patches do not straddle boundaries and are therefore at most as large as an entire boundary or as small as a

boundary cell face. The emissive power (and therefore the radiation intensity) and radiation properties are assumed to be uniform over the surface of each patch.

The view factors as in Eqns. Eqn. (2095) and Eqn. (2096) above are purely topological quantities, dependent only on the geometry of the two surfaces. However, the presence of partially transmissive and/or specularly reflective surfaces in STAR-CCM+ can introduce a further dependence on other surfaces and their respective surface properties. As a result, the collection of view factors is, in general, dependent on the surface geometry and the respective transmissivity and specular reflectivity of those surfaces. For a fixed geometry and fixed surface transmissivity and specular reflectivity, the view factors must be obtained only once, usually at the start of a simulation.

The double integral is approximated using a ray tracing approach. Collections of polygons represent surfaces, and the solid angle is discretized using an angular quadrature. For each surface or patch, a predefined number of rays is traced through the computational domain starting at the patch centroid. The directions and weights are based on discretization of an ideal hemisphere over the patch.

A ray shot from one patch can do any of the following:

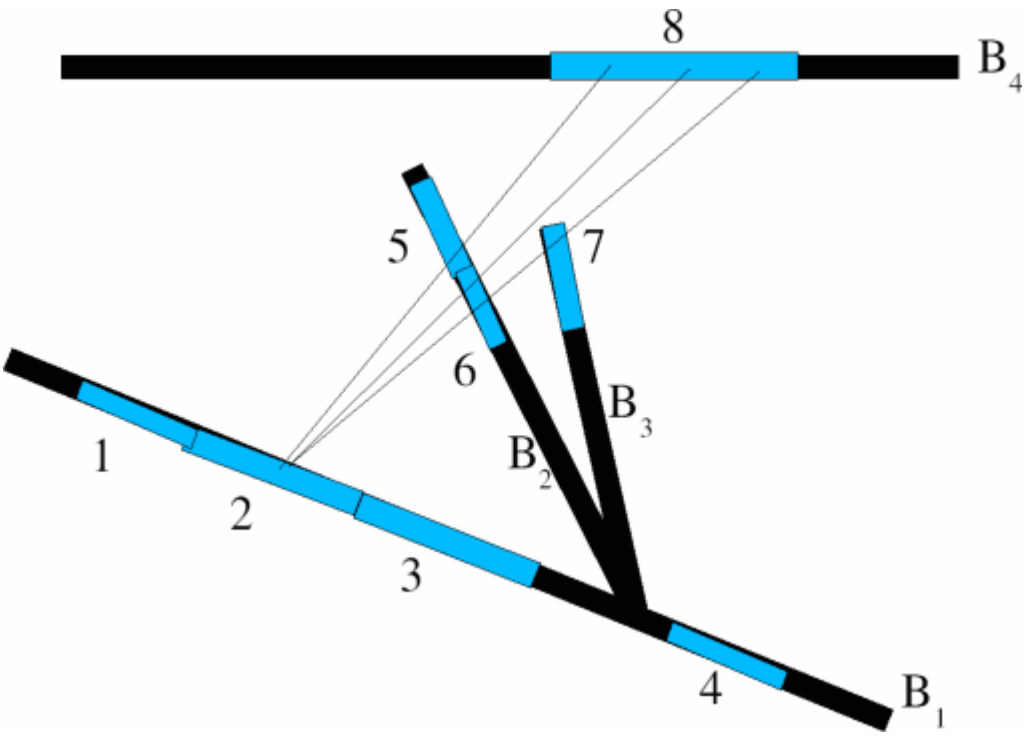
- Hit an opaque patch.
- Pass through a semi-transparent patch (with reduced strength proportional to the transmissivity of the boundary to which the hit patch belongs).
- Reflect from a specularly reflective patch.
- Pass through a transparent boundary or an opening and be dispersed in the environment.

In all these cases a view factor is generated between the shooting patch and, respectively, the opaque patch, the semi-transparent patch, the specularly reflective patch, or the environment.

This method is fairly simple and cost-effective compared to other methods (hemicube, hemisphere, or Monte Carlo). There are some analogies to the Unit Sphere Method [380]. After the beam tracking has finished, the array of view factors obtained satisfies conservation but not reciprocity, so a post-correction procedure is necessary to enforce reciprocity. Conservation is also enforced, since as soon as reciprocity is imposed conservation is lost.

The general definitions above describe the direct, unimpeded interaction between two isolated surfaces. In reality, when more surfaces are involved, radiation energy issuing from a patch can reach another patch after having gone through one or more non-opaque surfaces. The radiation energy can also undergo reflection by one or more specularly reflective surfaces.

The S2S radiation model makes it possible to model specular transmission, where radiation emitted from a surface is transmitted through another surface along its original direction. This situation is schematized below for boundaries B_1 to B_4 and selected patches:



Computing, for example, the irradiation from patch 2 on B_1 onto patch 8 on B_4, involves summing up the following: the energy which reaches 8 through 5 directly, the energy which reaches 8 through 6 directly, and the energy which reaches 8 through both 6 and 7. Each time a radiation ray is transmitted through a surface, the transmissivity of that surface attenuates its intensity. Therefore, the only way to account properly for all the possible paths is to account for them and the transmissivities at the time when the view factors are calculated. The calculation uses integration over small, discretized solid angles.

The influence of specularly reflective surfaces is similarly considered during the ray tracing process. Specularly reflective surfaces change the direction of rays and potentially the ray strength when the surface is less than fully reflective. Furthermore, surfaces can be both specularly reflective and transmissive. In this special case, each incident ray is split into two child rays -- one transmitted through the surface and one reflected from the surface. The strengths of the resulting two child rays are determined from the strength of the parent ray and by the respective specular transmissivity and reflectivity.

When transmissive and/or specularly reflective surfaces are present, the view factors for a patch pair are global view factors. These global view factors already account for all possible radiation interactions in the space between the two patches. The global view factors no longer are purely topological quantities, and must be recalculated every time the transmissivity or specular reflectivity of a surface in the domain is changed.

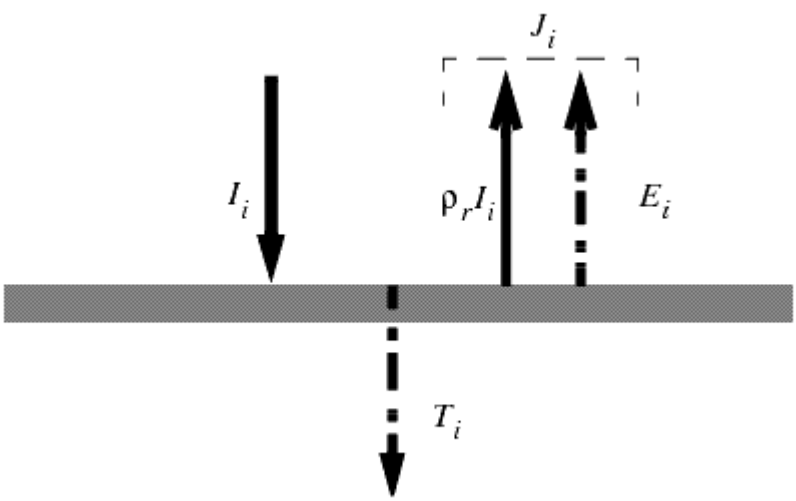
Environmental Patch

In addition to the patches on physical boundaries, a virtual patch is created when the view factors calculator detects non-opaque external surfaces. The environmental patch is internal to the code and makes it possible to collect and account for all the energy radiated to and from the environment around the regions. By definition, the environmental patch temperature is the radiation temperature of the thermal environment.

Radiation Equilibrium

After the surfaces have been decomposed into patches and view factors have been calculated for the patch pairs, radiant fluxes on each patch are obtained. The radiant fluxes are obtained by enforcing radiation equilibrium on the entire closed set of surfaces for each radiation spectrum or spectral band independently.

For a given radiation spectrum or spectral band, λ, radiation contributions on each patch can be expressed in terms of the incident, emitted, and transmitted radiation, as illustrated below for a generic patch i. The following diagram shows a schematic representation of radiation fluxes on patch i:



For radiation to balance over the entire closed set of surfaces, all radiation received by each patch must come from the other patches. Summing over all the patches, the total radiation incident on patch i is then:

$$Q_{I,i,\lambda} = S_i I_{i,\lambda} = \sum_{j=1}^{N_p} \frac{F_{\text{j-i},\lambda}}{\left(1-\tau_{i,\lambda}-\rho_{s,i,\lambda}\right)} J_{d,j,\lambda} S_j + \frac{F_{\text{e-i},\lambda}}{\left(1-\tau_{i,\lambda}-\rho_{s,i,\lambda}\right)} J_{e,\lambda} S_e \tag{2097}$$

where:

S_i, S_j are the surface areas of patches i and j, respectively

I_i is the irradiation or incident radiant flux on patch $i[\frac{W}{m^2}]$

N_p is the total number of patches

$F_{\text{j-i},\lambda}$ is the view factor from patch j to i resulting from the approximation of Eqns. Eqn. (2091) to Eqn. (2096) using ray tracing

$J_{d,j,\lambda}$ is the diffuse component of the radiant flux leaving patch j, sometimes termed the radiosity

$F_{e\text{-i},\lambda}$ is the view factor from the environmental patch to patch i

$\tau_{i,\lambda}$ is the transmissivity

$\rho_{s,i,\lambda}$ is the specular component of reflectivity

The view factors include the influence of any transmissive or specularly reflective surfaces. Correspondingly, the view factors are effectively normalized using the diffuse portion of the surface properties at patch i: $\left(1-\tau_{i,\lambda}-\rho_{s,i,\lambda}\right)$.

The general reciprocity relationship for view factors between surfaces that can be transmissive and/or specularly reflective is:

$$F_{\text{i-j},\lambda} = F_{\text{i-j},\lambda}\left[\frac{S_i\left(1-\tau_{i,\lambda}-\rho_{s,i,\lambda}\right)}{S_j\left(1-\tau_{j,\lambda}-\rho_{s,j,\lambda}\right)}\right] \tag{2098}$$

By introducing the reciprocity relationship into Eqn. (2097), the incoming radiation flux becomes:

$$I_{i,\lambda} = \sum_{j=1}^{N_p} \frac{F_{\text{i-j},\lambda}}{1-\tau_{j,\lambda}-\rho_{s,j,\lambda}} J_{d,j,\lambda} + \frac{F_{\text{i-e},\lambda}}{1-\tau_{e,\lambda}-\rho_{s,e,\lambda}} J_{e,\lambda} \tag{2099}$$

where $J_{d,j,\lambda}$is the diffuse radiosity:

$$J_{d,j,\lambda} = \rho_{d,j,\lambda} I_{j,\lambda} + E_{j,\lambda} + E_{d,j,\lambda} \tag{2100}$$

where:

$E_{j,\lambda}$ is the emission based on Planck's Law

$E_{d,j,\lambda}$ is the user-specified non-Planck emission

After introducing the effective radiosity as:

$$J_{i,\lambda}^{eff} = \frac{J_{d,i,\lambda}}{1-\tau_{i,\lambda}-\rho_{s,i,\lambda}} \tag{2101}$$

the irradiation can be written as:

$$I_{i,\lambda} = \sum_{j=1}^{N_p} F_{\text{i-j},\lambda} J_{j,\lambda}^{eff} + F_{\text{i-e},\lambda} J_{e,\lambda}^{eff} \tag{2102}$$

The total diffuse radiation that patch i emits is the sum of the diffusely reflected and emitted components:

$$J_{d,i,\lambda} = \rho_{d,i,\lambda} I_{i,\lambda} + E_{i,\lambda} + E_{d,i,\lambda} \tag{2103}$$

Combining Eqns. Eqn. (2102) and Eqn. (2103) for each patch yields an equation for $J_{i,\lambda}^{eff}$:

$$\begin{aligned}&\left(1-\tau_{i,\lambda}-\rho_{i,\lambda}^{s}\right)\left(1-F_{\text{i-i},\lambda}\right)J^{\text{eff}}{}_{i,\lambda}-\rho_{d,i,\lambda}\sum_{(j=1),(j\neq i)}^{N_p}\left[F_{\text{i-j},\lambda}J^{\text{eff}}{}_{j,\lambda}\right]\\&= E_{i,\lambda}+E_{d,i,\lambda}+\rho_{d,i,\lambda}\left[F_{\text{i-e},\lambda}\left(E_{\text{env},\lambda}+J_{d,\text{source},\lambda}\right)+J_{\text{dir,source},\lambda}\right]\end{aligned} \tag{2104}$$

where:

$J_{d,\text{source},e}$ is a specified external diffuse radiative flux source, for example solar radiation

$J_{\text{dir,source},i,\lambda}$ is a specified external collimated flux source

To close the system of equations, the emitted energy and environment radiosity on the right-hand side of Eqn. (2104) can be further specified. These terms are defined below for an arbitrary spectrum or band. As

such, the definitions are applicable to either the gray or multiband spectrum models. Simplifications for the gray spectrum model (gray thermal radiation) are noted where appropriate.

The surface emissive power for a given spectrum or band is the product of the emissivity and the blackbody emissive power for that spectrum or band:

$$E_{i,\lambda} = \varepsilon_{i,\lambda} \Delta_{\left(\lambda_1 \rightarrow \lambda_2, T_i\right)} \sigma T_i^4 \tag{2105}$$

where:

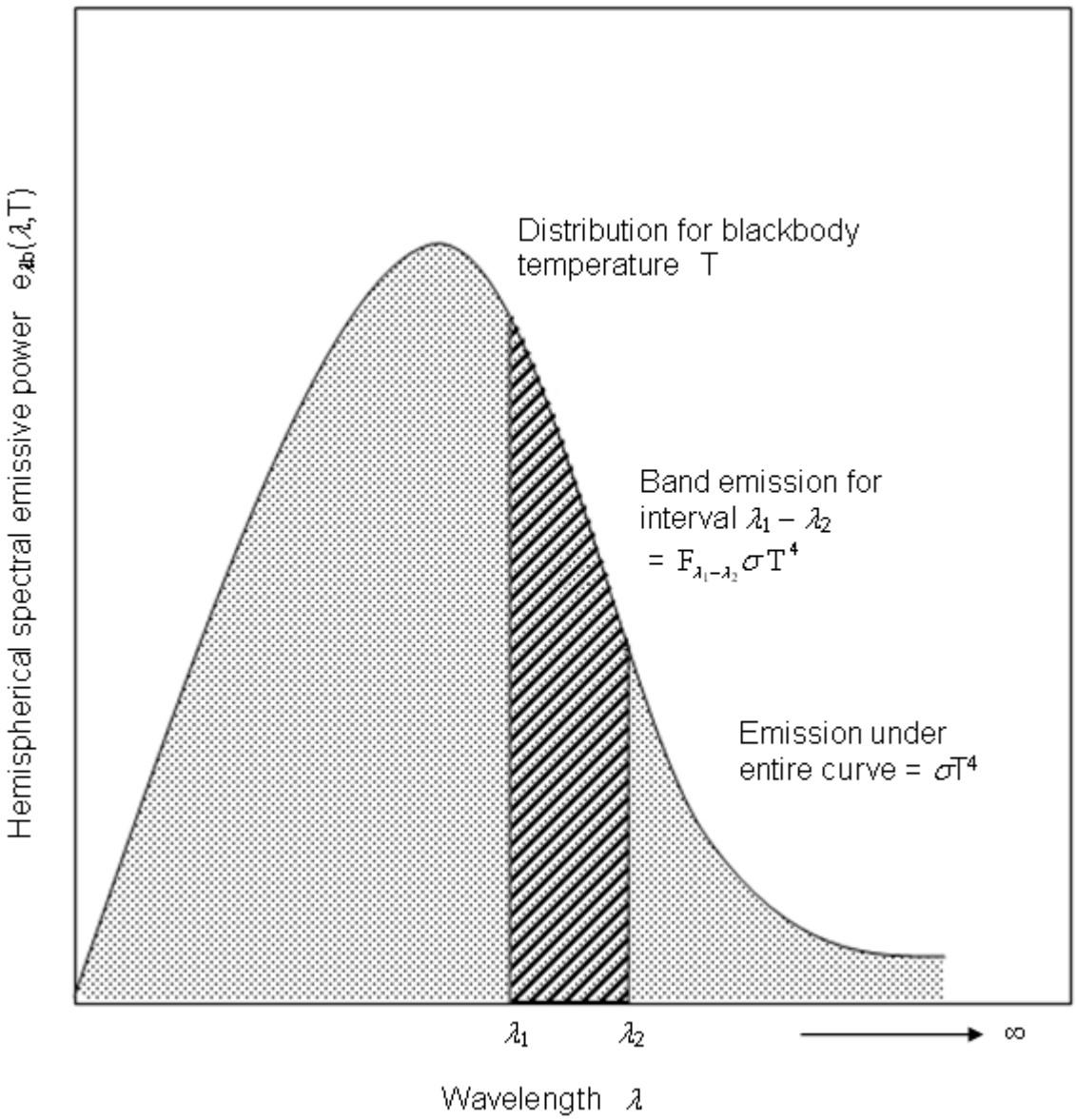
$\Delta_{\left(\lambda_1 \rightarrow \lambda_2, T_i\right)}$ is the fraction of total blackbody emission for the spectrum or band

λ_1 and λ_2 represent the lower and upper wavelength bounds of the spectrum or band

$\varepsilon_{i,\lambda}$ is the emissivity

σ is the Stefan-Boltzmann constant [$\frac{W}{m^2 K^4}$]

The following graph shows the fractional blackbody emission for a band:



For gray radiation, the wavelength bounds are effectively zero and infinity and this factor is simply unity. For multiband radiation, this fraction is a function of the surface temperature and wavelength bounds:

$$\Delta_{(\lambda_1 \to \lambda_2, T)} = \Delta_{(0 \to \lambda_2, T)} - \Delta_{(0 \to \lambda_1, T)} = f(\lambda_2 T) - f(\lambda_1 T) \tag{2106}$$

The fractional blackbody emission function can be evaluated by either integrating the Planck distribution or by interpolating from a table that is based on this integration [379].

The environment is modeled as a black body with unity emissivity and zero reflectivity and transmissivity. Thus, the effective radiosity of the environment is:

$$J_{e,\lambda}^{eff} = \Delta_{(\lambda_1 \to \lambda_2, T_e)} \sigma T_e^4 \tag{2107}$$

As before, the fraction of total blackbody emission for a given spectrum or band is dependent on the extent of the spectrum or band and the environment temperature. For Gray Thermal Radiation, the fraction reduces to unity.

If solar loads are considered, two more sources can be present—direct solar radiation and diffuse solar radiation. For purposes of apportioning these loads by spectrum or band, the spectral distribution of the solar emissions is assumed to be represented with a blackbody at the average surface temperature of the sun. The direct and diffuse solar fluxes augment the effective radiosity of the environment:

$$F_{\text{i-e},\lambda} J_{e,\lambda}^{eff} = F_{\text{i-e},\lambda}\left[\Delta_{\left(\lambda_1 \to \lambda_2, T_e\right)} \sigma T_e^4 + \Delta_{\left(\lambda_1 \to \lambda_2, T_{sun}\right)} J_{d,\text{solar},e}\right] + J_{\text{dir,solar},i,\lambda} \tag{2108}$$

This patch-based quantity is dependent on the specified sun position, and the specified direct solar flux, and the bounds of the given spectral waveband (used for apportioning the total direct flux). This quantity is also dependent on the specified spectrum/band transmissivities and the model geometry.

Substituting Eqns. Eqn. (2105) and Eqn. (2108) into Eqn. (2104) and expanding the i to i terms out of the sum yields the general system of equations for the effective radiosity:

$$\begin{aligned}\left(1-\tau_{i,\lambda}-\rho_{s,i,\lambda}\right)\left(1-F_{\text{i-i},\lambda}\right) J_{i,\lambda}^{eff} - \rho_{d,i,\lambda} \sum_{j=1, j \neq i}^{N_p} F_{\text{i-j},\lambda} J_{j,\lambda}^{eff} \\ = \varepsilon_{i,\lambda} \Delta_{\left(\lambda_1 \to \lambda_2, T_i\right)} \sigma T_i^4 + E_{d,i,\lambda} \\ + \rho_{d,i,\lambda}\left\{F_{\text{i-e},\lambda}\left[\Delta_{\left(\lambda_1 \to \lambda_2, T_e\right)} \sigma T_e^4 + \Delta_{\left(\lambda_1 \to \lambda_2, T_{sun}\right)} J_{d,\text{solar},e}\right] + J_{\text{dir,solar},i,\lambda}\right\}\end{aligned} \tag{2109}$$

For the gray spectrum model, only one system of equations exists, and that system represents the full thermal spectrum. For the multiband spectrum model, a separate system exists for each spectral band. In the absence of solar radiation, the solar load terms are omitted from Eqn. (2109).

Numerical Solution

The patch values for emissive power, reflectivity, and transmissivity are obtained by surface-averaging the cell-face values for all boundary faces that constitute each patch:

$$E_{i,\lambda} = \varepsilon_{i,\lambda}\Delta_{\left(\lambda_1 \to \lambda_2, T_i\right)}\sigma T_i^4 = \frac{\sum_{f_1^{(i)}}^{f_N^{(i)}} \epsilon_{f_k,\lambda}\Delta_{\left(\lambda_1 \to \lambda_2, T_{f_k}\right)} T_{f_k}^4 S_{f_k}}{\sum_{f_1^{(i)}}^{f_N^{(i)}} S_{f_k}}$$

$$\rho_{i,\lambda} = \frac{\sum_{f_1^{(i)}}^{f_N^{(i)}} \rho_{f_k,\lambda} S_{f_k}}{\sum_{f_1^{(i)}}^{f_N^{(i)}} S_{f_k}} \tag{2110}$$

$$\tau_{i,\lambda} = \frac{\sum_{f_1^{(i)}}^{f_N^{(i)}} \tau_{f_k,\lambda} S_{f_k}}{\sum_{f_1^{(i)}}^{f_N^{(i)}} S_{f_k}}$$

Surface-averaging the face emissive power rather than averaging the face emissivity and temperature independently is required to conserve energy. Given the emissive powers and properties for the entire set of patches, the set of radiation balance equations, Eqn. (2109), is solved in matrix form by the S2S solver and its AMG Linear Solver for each spectrum or band.

Based on the updated values of the effective radiosities, the patch irradiation fluxes are obtained directly from Eqn. (2102). The contribution to the irradiation from the environment in Eqn. (2102) is based on either Eqn. (2107) or Eqn. (2108) depending on whether solar loads are considered. Radiosities are then extracted from the effective radiosities using Eqn. (2101). In the presence of solar radiation, these fluxes include the influence of the solar loads.

The radiative fluxes on the faces are then derived from the patch fluxes. The irradiation (incident flux) at each face f is equal to the irradiation on the associated patch i. The face radiosity is a combination of local face emissive power and the reflected irradiation:

$$I_{f,\lambda} = I_{i,\lambda} \tag{2111}$$

$$J_{f,\lambda} = E_{f,\lambda} + \rho_{f,\lambda} I_{i,\lambda} \tag{2112}$$

The reflectivity in the above equation is the full reflectivity (diffuse + specular). The use of the local face emissive power rather than the patch emissive power is critical to ensuring local energy balances when the net radiative flux is later derived from the radiosity.

The full-thermal-spectrum fluxes are calculated by summing the fluxes from the individual spectra or bands:

$$I_f = \sum_{\lambda \in specta} I_{f,\lambda} \tag{2113}$$

$$J_f = \sum_{\lambda \in spectra} J_{f,\lambda} \tag{2114}$$

$$\left(q_r\right)_f = \sum_{\lambda \in spectra} \left(q_r\right)_{f,\lambda} = \sum_{\lambda \in specta} \left[\left(1-\tau_{f,\lambda}\right)_I - J_{f,\lambda}\right] \tag{2115}$$

For the gray spectrum model, the summation operation is trivial; the gray fluxes are the total fluxes. For the multiband spectrum model, the summation extends over all spectral bands. Eqn. (2115) defines the total or full-spectrum net radiative heat flux to the boundary (total absorption minus total emission). A positive value indicates net absorption by the boundary.

Mean Radiant Temperature (MRT)

The MRT is a useful variable for thermal comfort analysis as it provides “the uniform temperature of an imaginary enclosure in which radiant heat transfer from the human body equals the radiant heat transfer in the actual nonuniform enclosure.” See [383].

The MRT is derived by equating the actual heat flux with the idealized heat flux (that from an imaginary room at a uniform temperature and with unity emissivity). By substituting the expressions for the equated heat fluxes and rearranging, the MRT can be expressed in terms of the irradiation:

$$\left(T_{MRT}\right)_f = \left(\frac{I_f}{\sigma}\right)^{\frac{1}{4}} \tag{2116}$$

The irradiation or incident heat flux in the above expression is the result of the surface-to-surface solution. This formulation considers the potential influences of solar radiation, radiation from the thermal environment, transmission through non-opaque surfaces, and emission from non-black surfaces.

Participating Media Radiation

STAR-CCM+ accounts for participating media effects by using the Discrete Ordinate Method (DOM). The DOM is a numerical model for describing radiative heat transfer in participating media.

The DOM simulates thermal radiation exchange between diffuse/specular surfaces forming a closed set. The surface radiative properties are quantified in terms of emissivity, specular and diffuse reflectivity, transmissivity, and radiation temperature. These properties do not depend on direction. For the multiband spectrum model, the surface properties can depend on wavelength.

The medium that fills the space between the surfaces can also absorb, emit, or scatter radiation. Therefore, the amount of radiation that each surface receives and emits depends on this effect, as well as the optical properties of the surface and the thermal boundary conditions that are imposed on it.

As radiation travels through a medium, the intervening material absorbs and increments its radiant intensity I in the Ω direction. The radiative transfer equation (RTE) governs this process. This equation can be written in terms of radiant intensity for a specific wavelength λ:

$$\frac{\mathrm{d}\, I_\lambda}{\mathrm{d}\, s} = -\beta_\lambda I_\lambda + k_{a\lambda} I_{b\lambda} + \frac{k_{s\lambda}}{4\pi}\int_{4\pi} I_\lambda(\Omega)\ \mathrm{d}\ (\Omega) + k_{pa\lambda} I_{pb\lambda} + \frac{k_{ps\lambda}}{4\pi}\int_{4\pi} I_\lambda(\Omega)\ \mathrm{d}\ (\Omega) \tag{2117}$$

where:

I_λ	radiative intensity at wavelength λ $[W/\left(m^2 sr\, m^{-1}\right)]$
$k_{a\lambda}$	absorption coefficient at wavelength λ $\left(m^{-1}\right)$
$k_{s\lambda}$	scattering coefficient at wavelength λ $\left(m^{-1}\right)$

$I_{b\lambda}$	black body intensity at wavelength λ
Ω	solid angle
$k_{pa\lambda}$	particle absorption coefficient at wavelength λ $\left(m^{-1}\right)$
$k_{ps\lambda}$	particle scattering coefficient at wavelength λ $\left(m^{-1}\right)$
$I_{pb\lambda}$	particle black body intensity at wavelength λ and current particle temperature.
β_λ	extinction coefficient, which is defined as $\beta_\lambda = k_{a\lambda} + k_{s\lambda} + k_{pa\lambda} + k_{ps\lambda}$
s	distance in the Ω direction

The black body intensity is given by:

$$I_{b\lambda} = \frac{2C_1}{\lambda^5\left(e^{C_2/\lambda T} - 1\right)} \tag{2118}$$

where $C_1 = 0.595522 \times 10^{-16}\ Wm^2/s$ and $C_2 = 0.01439\ mK$.

The in-scattering component is assumed to be isotropic.

When the medium has several components, the absorption coefficient $k_{a\lambda}$ is given by:

$$k_{a\lambda} = \sum_i k_{a\lambda i} \tag{2119}$$

where i indexes the participating components, such as CO2, H2O, or soot.

When the absorption and scattering coefficients of the medium are independent of wavelength, the medium is called gray. In that case, the RTE can be integrated over wavelength (or, equivalently, wavenumber) to produce a wavelength-independent equation.

The particle radiation is valid for both gray and multiband radiation for continuous media. However, for the case of multiband (spectral) radiation, the particle properties are still gray. The surface and continuum properties can be spectral, but the particle properties are gray (the absorption efficiencies are independent of wavelength). Particle scattering is isotropic.

Boundary Conditions: Reflection, Refraction, and Radiosity

The boundary condition that is applied to the RTE for each wavelength λ is:

$$I_\lambda(\mathbf{s}) = \varepsilon_{w\lambda} I_{b\lambda} + \frac{E^d_{w\lambda}}{\pi} + \frac{\rho^d_{w\lambda}}{\pi_{eff}} \int_{(\mathbf{n}\cdot\mathbf{s}' < 0)} I_\lambda(\mathbf{s}')|\mathbf{n}\cdot\mathbf{s}'|\ \mathrm{d}\ \Omega + \rho^s_{w\lambda} I_\lambda(\mathbf{s}^\mathbf{s}) + \tau_{w\lambda}(\mathbf{s}^\mathbf{i}) I_\lambda(\mathbf{s}^\mathbf{i}) \tag{2120}$$

where:

$\varepsilon_{w\lambda}$ is the diffuse emissivity

$\frac{E^d_{w\lambda}}{\pi}$ is the diffuse non-Planck emission term

$\rho_{w\lambda}^{d}$ is the diffuse reflectivity

$\pi_{eff} = \sum_{\mathbf{s_i} \cdot \mathbf{n_i}} w_i(\mathbf{s_i} \cdot \mathbf{n_i})$ is the half moment for the ordinate set chosen

$\rho_{w\lambda}^{s}$ is the specular reflectivity

$\tau_{w\lambda}$ is the transmissivity. Only interfaces can be partially transmissive; all other external boundaries are modeled as opaque

$\mathbf{s^s} = \mathbf{s} - 2(\mathbf{s} \cdot \mathbf{n})\mathbf{n}$ is the incoming ray direction as determined by the laws of reflection

$\mathbf{s^i}$ is the incoming ray direction as determined by the laws of refraction or Snell's law

s' is the unit vector along the distance coordinate leaving the boundary

$\mathbf{n}$ is the surface normal

The w subscript indicates the state at the boundary. Surface normal points *toward* the surface. Radiative properties such as specular reflectivity and transmissivity at the boundary become direction-dependent for refraction.

The boundary radiosity is:

$$J_w = \int_{\mathbf{n} \cdot \mathbf{s} < 0} \epsilon_w I_{bw} |\mathbf{n} \cdot \mathbf{s}| d\Omega + \int_{\mathbf{n} \cdot \mathbf{s} > 0} \rho_w^d(\mathbf{s}') I(\mathbf{s}') |\mathbf{n} \cdot \mathbf{s}'| d\Omega + \int_{\mathbf{n} \cdot \mathbf{s^s} > 0} \rho_w^{\mathbf{s}}(\mathbf{s^s}) I(\mathbf{s^s}) \left| \mathbf{n} \cdot \mathbf{s^s} \right| d\Omega \tag{2121}$$

Boundary radiosity on external boundaries does not include contributions due to transmission from the environment. However, for interface boundaries, radiosity includes contributions due to transmission from adjacent regions. This is done to keep DOM consistent with S2S. For DOM, radiosity is a post-processing parameter.

Radiant Heat Flux

The radiant heat flux in a particular direction $\mathbf{q}_r$ is calculated using the integration of the radiant intensity over all solid angles and over the wavelength spectrum:

$$\mathbf{q}_r(r) = \int_0^\infty \int_{4\pi} I_\lambda(\mathbf{s}) s d\Omega d\lambda \tag{2122}$$

The radiation solution is coupled to the fluid dynamic solution through the divergence of the radiative heat flux. This term exchanges energy between the fluid and the radiant energy field. Given the intensity field, the divergence of the heat flux is computed as:

$$\nabla \cdot \dot{\mathbf{q}}_r = \int_0^\infty k_{a\lambda} \left(4\pi I_{b\lambda} - \int_{4\pi} I_\lambda d\Omega \right) d\lambda \tag{2123}$$

Refractive Index and Blackbody Intensity

Local blackbody intensity is augmented by the square of the absolute refractive index n:

$$I_b = \frac{n^2 \sigma T^4}{\pi} \tag{2124}$$

This equation applies both within the medium and at boundaries. In the medium, blackbody intensity is multiplied by absorption coefficient, while at boundaries, it is multiplied by boundary emissivity.

When radiation is transmitted or refracted from one medium to another, the intensity changes by:

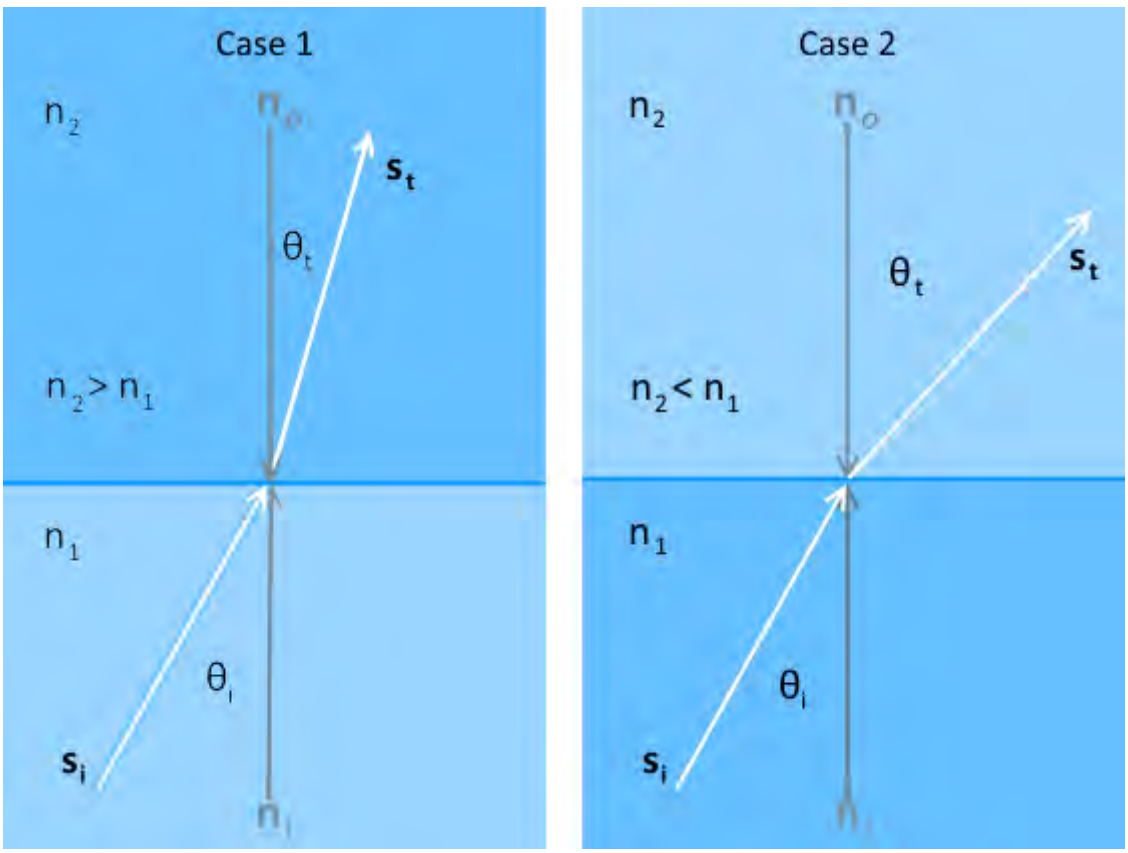
$$\frac{I_2}{n_2^2} = \frac{I_1 \tau_1}{n_1^2}$$

where subscript 1 indicates values on the incoming side of the boundary and subscript 2, for the outgoing side. The increase of intensity does not violate the emission maximum given by Planck's Law. Only the local intensity rises, not the global intensity for the whole medium.

Refraction, Reflection, and Critical Angle

Consider an interface boundary between two continua with refractive indices n_1 and n_2, with a light ray passing from the first to the second. There are two possible cases:

- Case 1: the ray passes from low to high refractive index, $n_2 > n_1$
- Case 2: the ray passes from high to low refractive index, $n_2 < n_1$



n_i and n_o are the inner and outer unit normal vectors

s_i and s_t are the unit direction vectors of the incident and transmitted rays

θ_i and θ_t are the angles of incidence and transmission

The angles are related to the indices of refraction by Snell's law:

$$n_1 \sin(\theta_i) = n_2 \sin(\theta_t)$$

In Case 1, the transmitted ray moves closer to the normal on the outer side, so all incident rays are transmitted through the surface.

In Case 2, the transmitted ray moves further from the normal on the outer side. As θ_i increases, θ_t eventually reaches 90° to the normal and is parallel to the surface. Beyond this point, the ray is reflected back into the first medium. The angle at which this change occurs is called the "critical angle" θ_c.

$$\theta_c = \sin^{-1}\left(\frac{n_2}{n_1}\right) \tag{2125}$$

The parameter calculated and stored for critical angle checking during run time is:

$$\cos^2(\theta_c) = 1 - \left(\frac{n_2}{n_1}\right)^2$$

For Case 1, because $n_2 > n_1$, the righthand side of this equation is negative, meaning there is no critical angle and no internal reflection.

For Case 2, because, $n_2 < n_1$, the righthand side of this equation is between 0 and 1, meaning a critical angle and internal reflection exist.

The vectors of the incident and transmitted rays are related as follows:

$$\mathbf{s}_t = \frac{n_1}{n_2}\left(\mathbf{s}_i - (\mathbf{s}_i \cdot \mathbf{n}_i)\mathbf{n}_i\right) - \sqrt{1 - \sin^2(\theta_t)}\ \ \mathbf{n}_o \tag{2126}$$

where:

$$\sin^2(\theta_t) = \left(\frac{n_1}{n_2}\right)^2 \left(1 - (\mathbf{s}_i \cdot \mathbf{n}_i)^2\right)$$

Correction Factors for Reflection and Refraction

Since the outgoing direction calculated by the laws of reflection and refraction does not always line up with the preset ordinate directions, STAR-CCM+ uses a nearest neighbor ordinate direction. This modeling creates slight misalignment in the direction of outgoing energy, which requires a correction factor to conserve total energy over that boundary face integrated over the hemisphere.

The correction factor for specular reflection is:

$$f^{refl} = \frac{\int_{\mathbf{n}\cdot\mathbf{s}>0} \rho_w^s(\mathbf{s}^i) I(\mathbf{s}^i) |\mathbf{n}\cdot\mathbf{s}| d\Omega}{\int_{\mathbf{n}\cdot\mathbf{s}<0} \rho_w^s(\mathbf{s}_{quad}^i) I(\mathbf{s}_{quad}^i) |\mathbf{n}\cdot\mathbf{s}| d\Omega} \tag{2127}$$

where:

ρ_w^s is specular reflectivity. For boundaries without refraction, this is a constant spatially and directionally, and drops out of the integral

$\mathbf{s}^i$ is the incoming direction

$\mathbf{s}_{quad}^i$ is the incoming direction related to each of the outgoing quadrature directions, found using the laws of reflection and nearest neighbor quadrature

For semi-transparent boundaries with refraction, specular reflectivity is not always constant over the hemisphere. If the critical angle θ_c is less than 90°, specular reflectivity outside that cone of incidence is equal to specular reflectivity inside plus transmissivity, $\rho_w^s + \tau_w$.

The correction factor for refraction transmission is:

$$f^{\text{refrac}} = \frac{\int_{\mathbf{n}\cdot\mathbf{s}>0} \tau_w(\mathbf{s}^r) I(\mathbf{s}^r) |\mathbf{n}_1 \cdot \mathbf{s}^r| d\Omega}{\int_{\mathbf{n}\cdot\mathbf{s}<o} \tau_w(\mathbf{s}_{quad}^r) I(\mathbf{s}_{quad}^r) |\mathbf{n}_2 \cdot \mathbf{s}| d\Omega} \tag{2128}$$

where:

$\mathbf{s}^r$ is the incoming direction of transmission

$\mathbf{s}_{quad}^r$ is the incoming direction of transmission related to each of the outgoing quadrature directions, found using the laws of reflection and nearest neighbor quadrature

To find $\mathbf{s}_{quad}^r$, first calculate a theoretical incident ray $\mathbf{i}$ using:

$$\mathbf{i} = \frac{n_2}{n_1}\left(\mathbf{s}_t - (\mathbf{s}_t \cdot \mathbf{n}_o)\mathbf{n}_o\right) - \sqrt{1 - \sin^2(\theta_t)}\ \ \mathbf{n}_i$$

with

$$\sin^2(\theta_i) = \left(\frac{n_2}{n_1}\right)^2 \left(1 - (\mathbf{s}_t \cdot \mathbf{n}_o)^2\right)$$

This theoretical incident ray $\mathbf{i}$ is then used to find the nearest quadrature direction within the critical angle θ_c.

For the transmissivity τ_w, if the critical angle θ_c is less than 90°, transmissivity is zero outside the cone of incidence.

The outgoing intensity is then given by Eqn. (2120) with the DOM direction correction included:

$$I_\lambda(\mathbf{s}) = \varepsilon_{w\lambda} I_{b\lambda} + \frac{\rho_{w\lambda}^d}{\pi_{eff}} \int_{\mathbf{n}\cdot\mathbf{s}'<0} I_\lambda(\mathbf{s}') |\mathbf{n}\cdot\mathbf{s}'| d\Omega + f^{\text{refl}} \rho_{w\lambda}^s(\mathbf{s}_{quad}^i) I_\lambda(\mathbf{s}_{quad}^i) + f^{\text{refrac}} \tau_{w\lambda}(\mathbf{s}_{quad}^i) I_\lambda(\mathbf{s}_{quad}^r) \tag{2129}$$

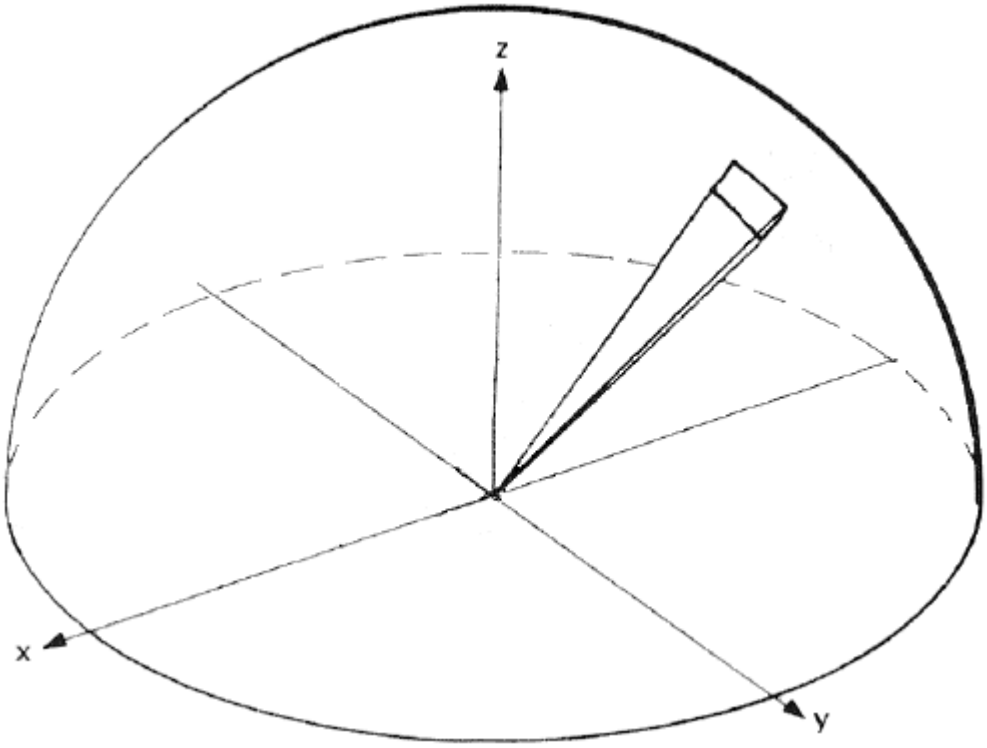
Contents:

Discrete Ordinate Method Numerical Solution

Discrete Ordinate Method Numerical Solution

The discrete ordinates method solves field equations for radiation intensity that is associated with a fixed direction s, represented by a discrete solid angle.

The following diagram illustrates a discrete solid angle as part of a hemisphere:



A solid angle is the three-dimensional analog of an ordinary angle and is measured in steradians [sr]. One steradian is the solid angle that is subtended at the center of a sphere by an area on the surface of the sphere equal to the radius squared. As such, the solid angle of a complete sphere is 4π steradians.

The discrete ordinates calculation requires specification of the number of solid angles or “ordinates” into which the sphere surrounding any particular point is discretized. Increasing the number of ordinates leads to greater accuracy. A detailed description of the method can be found in Modest [380] and Siegel and Howell [391]. As a result of Eqn. (2117), the form of these ordinate equations (for each wavelength band) is:

$$s_i \cdot \nabla I_{i\Delta\lambda} = -\beta_{\Delta\lambda}\ I_{i\Delta\lambda} + k_{a\Delta\lambda}\ I_{b\Delta\lambda} + \frac{k_{s\Delta\lambda}}{4\pi}\sum_{j=1}^{n} w_j\ I_{j\Delta\lambda} + \overline{k}_{p\Delta\lambda}\ I_{p\Delta\lambda} + \frac{\overline{k}_{p\Delta\lambda}}{4\pi}\sum_{j=1}^{n} w_j\ I_{j\Delta\lambda} \tag{2130}$$

In the above equation, $\Delta\lambda$ represents a wavelength band from λ_m to λ_n. The black body emission in this band is:

$$I_{b\Delta\lambda} = \int_0^{\lambda_n} I_{b\lambda}\ \mathrm{d}\ \lambda - \int_0^{\lambda_m} I_{b\lambda}\ \mathrm{d}\ \lambda \tag{2131}$$

The equivalent particle absorption and scattering coefficients are given in [380] as:

$$\overline{k}_{pa\Delta\lambda} = \sum_i Q_{a,i} N_i \frac{\pi d_i^2}{4} \tag{2132}$$

$$\overline{k}_{ps\Delta\lambda} = \sum_i Q_{s,i} N_i \frac{\pi d_i^2}{4} \tag{2133}$$

where i indicates the parcel index number.

The transport equation for each ordinate direction is discretized and solved independently, using the same discretization and iterative solution practices as for the other transport equations. For this reason, global iterations are necessary to include the isotropic in-scattering terms in the RTE and to compute wall boundary conditions. The angularly discretized boundary conditions take the form:

$$I_{i\Delta\lambda,w} = \varepsilon_{w\Delta\lambda} I_{b\Delta\lambda} + \frac{\rho^{d}_{w\Delta\lambda}}{\pi_{\text{eff}}} \sum\nolimits_{(\mathbf{n}\cdot\mathbf{s}'<0)} I_{j\Delta\lambda} \left|\mathbf{n}\cdot\mathbf{s}'_{j}\right|_{w} + f\rho^{s}_{w\Delta\lambda} I_{\Delta\lambda}\left(\mathbf{s}^{\mathbf{s}}\right) + \tau_{w\Delta\lambda} I_{i\Delta\lambda} \tag{2134}$$

where π_{eff} is the half moment for the chosen ordinate set, which is defined as:

$$\pi_{\text{eff}} = \sum\nolimits_{\mathbf{n}\cdot\mathbf{s}'>0} w_j \left(\mathbf{n}\cdot\mathbf{s}'_{j}\right)_{w} \tag{2135}$$

and f is the correction factor that is used when the direction of specular reflection is not aligned with any discrete ordinate direction. The diffuse reflection term represents summation over incoming (incident) ordinate directions.

Convergence is measured using the infinity norm of the normalized difference in the incident energy from successive space-angle sweeps.

The radiation solution provides a source term to the fluid dynamic energy equation as given by Eqn. (2123). The discretized form of this source term at any cell is:

$$-\nabla\cdot\mathbf{q}_r = \sum_{\lambda} k_{a\Delta\lambda}\left(\sum_{j=1}^{n} w_j I_{j\Delta\lambda} - 4\pi I_{b\Delta\lambda}\right) \tag{2136}$$

The net absorption rate of radiative energy by all particles within a given cell is:

$$-\nabla\cdot\mathbf{q}_{pr} = \sum_{\lambda} \overline{k}_{pa\Delta\lambda}\left(\sum_{j=1}^{n} w_j I_{j\Delta\lambda} - 4\pi I_{pb\Delta\lambda}\right) \tag{2137}$$

For Lagrangian particle transport, at convergence this quantity exactly balances the net radiative energy that all particles in the cell absorb.

Discrete Ordinate Quadrature

The ordinate set that is required to solve Eqn. (2130) with boundary conditions (Eqn. (2134)) is obtained using the discrete ordinate quadrature scheme. General principles for DOM quadrature schemes [388] are:

- All discrete ordinates, $\Omega_m = \left(\zeta_m, \eta_m, \mu_m\right)$, where $\left(\zeta_m, \eta_m, \mu_m\right)$ are the direction cosines, must be placed on the unit sphere, that is:

$$\left(\zeta_m^2 + \eta_m^2 + \mu_m^2\right) = 1 \tag{2138}$$

- The weights that are associated with each ordinate direction must be all positive. This requirement ensures that the error of the quadrature scheme is minimized.
- The number of photons must be preserved which is equivalent to satisfy:

$$\sum_{m=1}^{M} w_m\left(\mathbf{n}\overline{\Omega_m}\right) = 0 \tag{2139}$$

where w_m is the weight that is associated with the ordinates and $\mathbf{n}$ is an arbitrary unit vector.

- The quadrature must be invariant to any rotation of the arrangement of the discrete directions around the center of the unit sphere.

Although the first three criteria can be easily fulfilled, the fourth is impossible to be satisfied for any arbitrary rotation by a finite number of ordinates. Hence all ordinate sets must satisfy at least zeroth, first and second moments. For accuracy in radiation calculations, the ordinate set with weights must also satisfy the first moment over a half range along the three principal directions. This expression is:

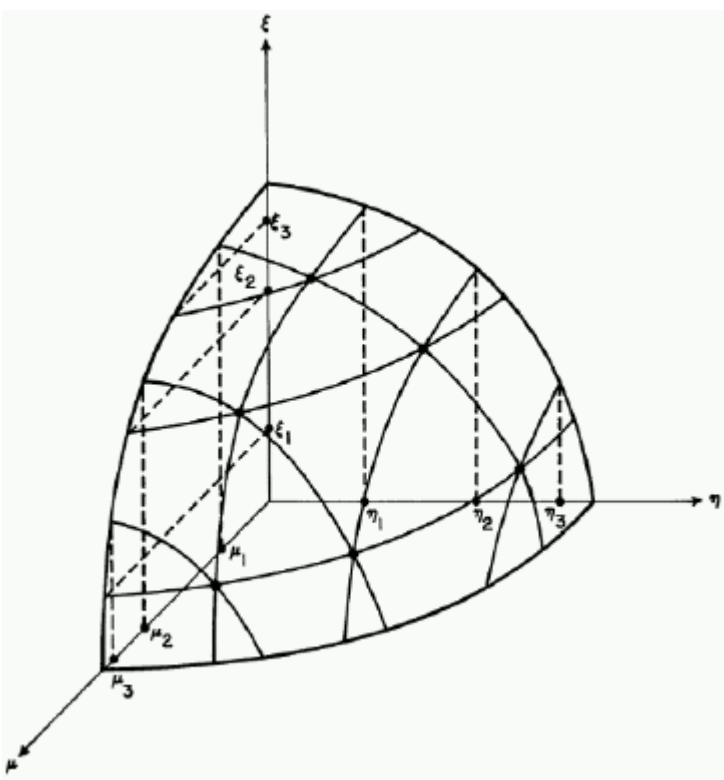
$$\sum_{\mathbf{n}\overline{\Omega_m} > 0} w_m\left(\mathbf{n}\overline{\Omega_m}\right) = \pi \tag{2140}$$

This additional requirement comes from the following:

- Radiative intensity can have directional discontinuity at a wall.
- Important radiative heat fluxes at the walls are evaluated through a first moment of intensity over a half range of 2π.

Level Symmetric (Sn) Quadrature

Level symmetric, also known as Sn, quadrature is a quadrature scheme where the discrete ordinates are arranged on the latitudes on the unit sphere. The corresponding latitudes have the same distance in the x-direction, y-direction, and z-direction and therefore this quadrature scheme is called Level Symmetric quadrature. A typical arrangement is illustrated in the figure (symmetric point arrangement for n = 6) [389].



The rotational invariance to the principal octahedron is satisfied implicitly because of the arrangement on latitudes. Due to the latitude-based arrangement, Sn quadrature can be used in solution of discrete ordinate equations both in three-dimensional and axisymmetric two-dimensional problems. Currently only Sn quadratures are available in STAR-CCM+. Sn quadratures can be formulated to satisfy either odd or even moments. To find the Sn formulation that satisfies a particular moment, see the following table. Sn(O) is the odd formulation and Sn(E) is the even formulation.

	Type	
Order	Sn(O)	Sn(E)
2	0, 1	0, 2
4	0, 1, 2	0, 2, 4
6	0, 1, 2, 3	0, 2, 4, 6
8	0, 1, 2, 3, 4	0, 2, 4, 6, 8
12	0, 1, 2, 3, 4, 5, 6	0, 2, 4, 6, 8, 10, 12
16	0, 1, 2, 3, 4, 5, 6, 7, 8	0, 2, 4, 6, 8, 10, 12, 14, 16

All Sn quadratures ($N > 2$) with odd formulation satisfy the primary requirements of satisfying zeroth, first and second moments whereas the even formulation does not. However odd formulation of Sn incurs negative weights for ($N \geq 10$) which are, in general, not suitable due to inaccuracy. The even formulation of Sn does not have that problem except for N = 10. Hence, in STAR-CCM+ odd formulation is available for N = 2, 4, 6, and 8, and even formulation is available for N = 12 and 16. All of the above available Sn quadratures in STAR-CCM+ satisfy the half moment criteria along principal directions.

Radiative Properties

The solution of the radiative transport equation (RTE) requires the definition of radiative properties for interfaces and surfaces, along with each if participating media are considered.

Spectral Modeling of Thermal Radiation

STAR-CCM+ uses the thermal radiation model to define how thermal radiation properties are modeled—either as wavelength independent (gray) or as wavelength dependent (spectral). The choice pertains uniformly to all radiation within the regions of the simulation and in the surrounding environment. The thermal radiation model is responsible for managing the radiative properties and the spectrum-dependent solution variables in the continuum.

The gray thermal radiation model accounts for diffuse and specular radiation independent of wavelength. The radiation properties of the media and the surrounding surfaces are considered the same for all wavelengths.

The multiband thermal radiation model lets you simulate diffuse and specular radiation that is dependent on wavelength or frequency. With this model, wavelength bands specify radiation properties of the media and surrounding surfaces. The number and extents of bands are chosen to describe property variations with respect to wavelength. During the solution process, the governing radiation equations are solved for each defined band using the specified properties. The total radiation solution is then obtained by integrating (summing) the solutions from each of the individual bands. Since the governing equations are solved for each

band, the radiation solution time generally increases linearly with the number of bands. As a consequence, a trade-off often exists between accuracy and computational time.

Radiative Properties of Surfaces

Surface properties are defined at the boundary level within STAR-CCM+.

Absorptivity

Absorption is a phenomenon that describes energy being removed from the incident radiative energy by the material or the boundary. Absorptivity is dimensionless and is defined as:

$$\alpha_\lambda = \frac{q_r^{abs}}{I_{f,\lambda}} \tag{2141}$$

where q_r^{abs} is the radiative energy that is absorbed and $I_{f,\lambda}$ is the total radiative energy incident on a boundary. From Kirchhoff's law, the spectral directional emissivity is equal to the spectral directional absorptivity. In STAR-CCM+, directional properties are not used. Therefore, Kirchhoff's law is implemented as:

$$\epsilon_\lambda = \alpha_\lambda \tag{2142}$$

where ϵ_λ is the spectral emissivity and α_λ is the spectral absorptivity. For a gray model, the properties are independent of both direction and wavelength, and emissivity equals absorptivity for the entire spectrum.

Transmissivity

Transmissivity τ_λ is dimensionless and is defined as:

$$\tau_\lambda = \frac{\dot{q}_r^{trans}}{I_{f,\lambda}} \tag{2143}$$

where $\dot{q}_r^{trans}$ is the transmitted radiant energy and $I_{f,\lambda}$ is the incident radiant energy at a given surface.

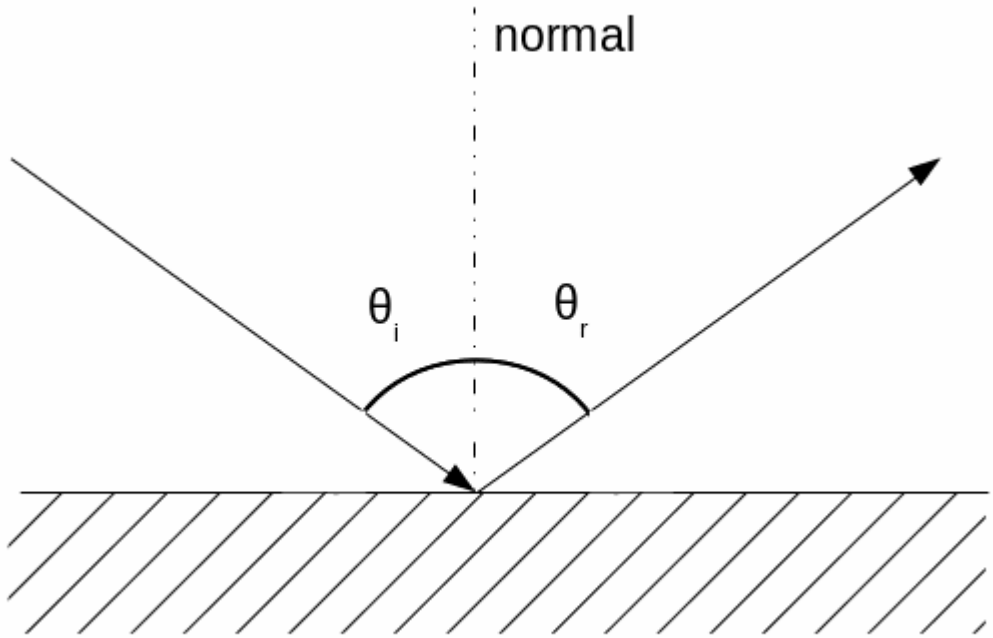
Reflectivity

Reflectivity ρ_λ is defined as:

$$\rho_\lambda = \frac{\dot{q}_r^{refl}}{I_{f,\lambda}} \tag{2144}$$

where $\dot{q}_r^{refl}$ is the reflected radiant energy.

Surface reflectivity can be partially or fully specular (as opposed to diffuse). A perfectly specular surface is one where the direction of incoming light is perfectly reflected across the surface normal. The angle of incidence θ_i is equal to the angle of reflection $\theta_r$as is shown in the following diagram:



Reflection specularity is the specular fraction of the surface reflectivity. A value of 0 denotes that all of the reflection is diffuse, whereas a value of 1 denotes that all of the reflection is specular. Values between 0 and 1 indicate partial specularity. Specular reflectivity is then determined from the product of reflection specularity and surface reflectivity. A special case is that of a mirrored surface that is represented by setting the surface reflectivity and the reflection specularity to 1. This setting means that the surface is completely reflective and that all of the reflection is specular. In this case, the specular reflectivity is also equal to 1. Alternatively, specular reflective surfaces an also be partially transmissive, for example, plate glass or particularly low-emissivity glas and/or they can be absorptive such as high-gloss-painted surface. In such cases, the surface reflectivity is less than 1 with surface transmissivity and/or surface emissivity making up the difference to satisfy Kirchhoff's law.

Radiative Properties of Participating Media

In STAR-CCM+, the media can absorb, emit, scatter isotropically, and refract. The absorption coefficient controls absorption/emission, the scattering coefficient controls scattering, and the refractive index controls local emission as well as the direction of radiation at the interface between two media. For these coefficients, you can specify the values directly or use the built-in Weighted Sum of Gray Gases method.

As radiation travels through a medium, the intervening material absorbs and increments its radiant intensity I in the Ω direction. The radiative transfer equation (RTE) Eqn. (2117) governs this process.

Absorption Coefficient

Absorption is a phenomenon that describes energy being removed from the incident radiative energy by the material. This phenomenon is described using the absorption coefficient k. Incoming intensity is linearly decremented by k along a line of sight according to:

$$\frac{dI_{\lambda,abs}}{ds} = -kI_{\lambda} \tag{2145}$$

In local thermodynamic equilibrium conditions, Kirchhoff's law applies. The volumetric emission is proportional to the absorption coefficient and the blackbody function. The local intensity along a line of sight is augmented following:

$$\frac{dI_{\lambda,em}}{ds} = kI_{b,\lambda} \tag{2146}$$

Scattering Coefficient

The scattering coefficient is the property of a medium that describes the amount of scattering of thermal radiation per unit path length within the medium. Scattering in STAR-CCM+ is isotropic for both continuum and particle radiation.

Refractive Index

The refractive index governs local emission and the direction as is described in Refractive Index and Blackbody Intensity and Refraction, Reflection, and Critical Angle.

In STAR-CCM+, refraction is only considered at the interface between two media. Therefore, refraction is specified for each medium.

When the absorption and the scattering coefficients as well as the refractive index of the medium are independent of wavelength, the medium is called gray. In that case, the RTE can be integrated over wavelength (or, equivalently, wavenumber) to produce a wavelength-independent equation. If the properties vary across the spectrum, a multiband thermal model can be used to model the spectral variability.

Weighted Sum of Gray Gases Method (WSGGM)

STAR-CCM+ employs the weighted sum of gray gases method (WSGGM) to model radiative heat transfer properties in participating media that are gas mixtures consisting of $\mathrm{CO_2}$ and $\mathrm{H_2O}$. WSGGM is widely used in the simulation of air-fuel combustion processes.

Combustion product gases do not scatter radiation significantly, but are strong selective emitters and absorbers of radiation. $\mathrm{CO_2}$ and $\mathrm{H_2O}$ gases dominate cloud emission and absorption among the combustion gas products. These species require that the spectral variation of radiative properties is accounted for.

The weighted sum of gray gases (WSGG) method for calculating the absorption coefficient is a reasonable compromise between the oversimplified user-provided gray gas model and a complete spectral property model which takes into account detailed absorption bands. This method uses the Hottel charts [385] and sum-of-gray-gases models [386] for a mixture containing $\mathrm{CO_2}$ and/or $\mathrm{H_2O}$ gases only. Several curve-fitted sum-of-gray-gases expressions are available [384]. In the WSGG method, the medium is assumed to consist of different fractions of gray gases with different absorption coefficients.

The relationship between the total gray gas absorption coefficient κ and the total absorptivity α for a gray medium is the Bouguer-Lambert law:

$$\alpha = 1 - e^{-\kappa s} \tag{2147}$$

where s is the optical path length (OPL).

The total absorptivity is approximated with a weighted-sum of several gray gases as:

$$\alpha \approx \sum_{k=0}^{K} a_k \left(1 - e^{-\kappa_k s}\right) \tag{2148}$$

where K is the total number of gray gases, a_k is the weight factor and κ_k is the absorption coefficient of each gray gas. Based on the quality of the curve-fit of Hottel charts [385] and sum-of-gray-gases models [386], typically $K = 3$ gives satisfactory accuracy. STAR-CCM+ always uses this value of K.

After computing α from Eqn. (2148), the total gray gas absorption coefficient κ is obtained using the Bouger-Lambert Law (Eqn. (2147)). κ is then used to solve the RTE as k_a.

The OPL is a user-provided quantity. The OPL can be used as a tuning parameter to get the expected and reasonable distribution for the absorption coefficient of the gas medium. The OPL value can range from the cell dimensions to the dimension of the entire computational domain, depending on the optical thickness of the combustion region. For instance, if the optical thickness of the combustion medium is high then the cell thickness can be used to define the OPL. However, for an optically thin medium, use a value that corresponds to the reactor dimension.

The OPL is related to optical thickness by:

$$\theta = s\kappa \tag{2149}$$

where θ is the optical thickness, s is the OPL, and κ is the absorption coefficient.

Once a value of the OPL is prescribed, the WSGG method captures local variation of the absorption coefficient (which is due to variation in the gas temperature and partial pressures of combustion gases).

For arbitrary geometries, the OPL can be approximated as [387]:

$$s = 3.6 \ \ (\text{volume}/\text{surface area}) \tag{2150}$$

where the volume is the total volume of the gas and the surface area is its total surface area.

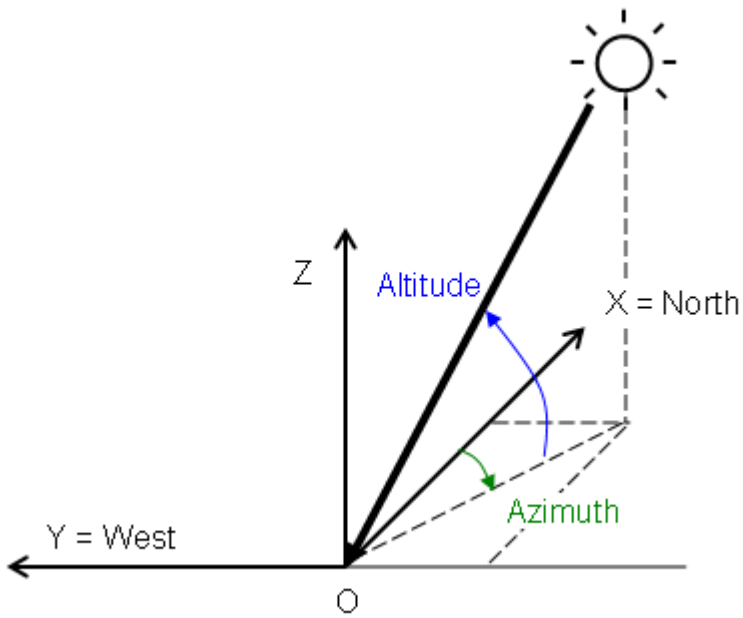
Solar Radiation

Thermal radiative exchange with the solar environment includes both direct and diffuse components.

The direct solar flux is the irradiation from the sun incident directly on the regions without interference from the atmosphere or other objects. It is modeled as a collimated source as given by $J_{\mathrm{dir,\,source},\,\lambda}$ in Eqn. (2104).

Diffuse solar radiation results from the atmospheric scattering, and reflection and transmission from objects such as the ground. It is modeled as a diffuse source $J_{\mathrm{d,\,source},\,\lambda}$ in Eqn. (2104).

The coordinate system for the orientation of the direct solar flux is defined in the following figure:



The X-axis is aligned with the local magnetic north direction and the Y-axis is aligned with the local west direction—that is, the XY plane is the local earth-surface plane. The Z-axis then points skyward since STAR-CCM+ employs right-handed coordinate systems.

User Input Range for the Solar Calculator

The solar calculator is a set of equations that calculate the astronomical relationship between sun, earth, and the atmosphere. It is developed by the National Renewable Energy Laboratory (NREL) [392]. This formulation presents the equations for solar position and flux calculations in the solar calculator. The user inputs for various parameters must lie within the ranges that are shown in the following table .

All angles that are shown in this formulation are in degrees.

Variable		Range
Year		1950-2050
Month		1-12

Variable		Range
Day		1-31 (depending on the month)
Hour		0 – 23
Minute		0-59
Time zone		-12.0 -+12.0 (in hours)
Latitude		- 90.0-+90.0 (in degrees)
Longitude		-180.0-+180.0 (in degrees)
Fraction for diffuse flux		0.0-1.0
Sunshine factor		0.0-1.0

Calculation of Sun Direction

Solar altitude, azimuthal angles, and related geometric parameters can be calculated as:

- Day angle [393]:

$$\alpha_d = \frac{360°}{365}\left(d_{year} - 1\right) \tag{2151}$$

where d_{year} is the day of the year.

- Earth radius vector [394]:

$$r_E = 1.000110 + 0.034221\cos\left(\alpha_d\right) + 0.001280\sin\left(\alpha_d\right) + 0.000719\cos\left(2\alpha_d\right) + 0.000077\sin\left(2\alpha_d\right) \tag{2152}$$

Coordinated Universal Time (UTC) in hours:

$$UTC = hour + \frac{minute}{60} - timeZone \tag{2153}$$

Julian day [395]:

$$d_{jul} = 32916.5 + 365\left(year - 1949\right) + \frac{year - 1949}{4} + d_{year} + \frac{UTC}{24} \tag{2154}$$

Time that is used in calculation of elliptical coordinate [395]:

$$t_{ec} = d_{jul} - 51545.0 \tag{2155}$$

Mean longitude [395]:

$$L_m = 280.460 + 0.9856474t_{ec} \qquad \left(0° \le L_m \le 360°\right) \tag{2156}$$

Mean anomaly [395]:

$$A_m = 357.528 + 0.9856003t_{ec} \qquad \left(0° \le A_m \le 360°\right) \tag{2157}$$

Elliptic longitude [395]:

$$L_e = L_m + 1.915\sin\left(A_m\right) + 0.020\sin\left(2A_m\right) \qquad \left(0° \le L_e \le 360°\right) \tag{2158}$$

Obliquity of the elliptic [395]:

$$O_e = 23.439 - \left(4 \times 10^7\right) t_{ec} \tag{2159}$$

Declination [395]:

$$D_c = \sin^{-1}\left\{\sin\left(O_e\right)\sin\left(L_e\right)\right\} \tag{2160}$$

Right ascension [395]:

$$R_a = \tan^{-1}\left\{\frac{\cos\left(O_e\right)\sin\left(L_e\right)}{\cos\left(L_e\right)}\right\} \qquad \left(0° \le R_a \le 360°\right) \tag{2161}$$

Greenwich mean sidereal time [395]:

$$t_{gmt} = 6.697375 + 0.0657098242 t_{ec} + UTC \qquad \left(0 \le t_{gmt} \le 24\right) \tag{2162}$$

Local mean sidereal time [395]:

$$L_{mst} = 15 t_{gmt} + L_o \qquad \left(0° \le L_{mst} \le 360°\right) \tag{2163}$$

where L_o is the user-defined longitude of the geographical location.

Hour angle [395]:

$$\beta_h = L_{mst} - R_a \qquad \left(-180° \le \beta_h \le 180°\right) \tag{2164}$$

Solar zenith Z and altitude angles without refraction [395]:

$$Z = \cos^{-1}\left\{\sin\left(D_c\right)\sin\left(L_a\right) + \cos\left(D_c\right)\cos\left(L_a\right)\cos\left(\beta_h\right)\right\} \qquad \left(0° \le Z \le 99°\right) \tag{2165}$$

where L_a is the user-defined latitude of the geographical location. Z is limited to 9 degrees below the horizon to compensate for refraction.

Altitude:

$$\theta = 90° - Z \tag{2166}$$

Solar azimuthal angle [395]:

$$\phi = 180° - \cos^{-1}\left\{\frac{\sin\left(\theta\right)\sin\left(L_a\right) - \sin\left(D_c\right)}{\cos\left(\theta\right)\cos\left(L_a\right)}\right\} \qquad \text{if} \quad \beta_h > 0°, \quad \phi = 360° - \phi \tag{2167}$$

Application of Refraction Correction

The refraction correction [396] is applied as follows:

cf is the correction factor due to atmospheric refraction.

$$cf = 0 \qquad \text{if } \theta > 85° \tag{2168}$$

$$cf = \frac{58.1}{\tan(\theta)} - \frac{0.07}{\tan^{3}(\theta)} + \frac{0.000086}{\tan^{5}(\theta)} \qquad \text{if } 5° \leq \theta \leq 85° \tag{2169}$$

$$cf = 1735 + \theta\left[-518.2 + \theta\left\{103.4 + \theta\left(-12.79 + 0.711\right)\right\}\right] \quad \text{if } -0.575° \leq \theta < 5° \tag{2170}$$

$$cf = \frac{20.774}{\tan(\theta)} \qquad \text{if } \theta < -0.575° \tag{2171}$$

$$cf = \left(\frac{cf}{3600}\right)\left(\frac{283P}{1013(273 + T)}\right) \tag{2172}$$

where P equals 1013 millibars atmospheric pressure and T equals $15°\mathrm{C}$ mean atmospheric temperature.

Refraction corrected altitude:

$$\theta_r = \theta + cf \qquad \left(-9° \leq \theta_r \leq 90°\right) \tag{2173}$$

Refraction corrected zenith:

$$Z_r = 90 - \theta_r \tag{2174}$$

Calculation of Solar Fluxes

Total solar flux:

$$q = C_s r_E\left(\cos\left(Z_r\right)\right) f_s \qquad \text{if } \cos\left(Z_r\right) > 0 \tag{2175}$$

$$q = 0 \qquad \text{if } \cos\left(Z_r\right) \leq 0 \tag{2176}$$

where C_s is the solar constant, equal to 1376.0 W/m^2, and f_s is the sunshine factor.

Diffuse flux:

$$q_{diff} = q f_{diff} \tag{2177}$$

where f_{diff} is the diffuse factor.

Direct flux:

$$q_{dir} = q\left(1.0 - f_{diff}\right) \tag{2178}$$

Heat Transfer Bibliography

[379] *Siegel, R., and Howell, J.R. 1992. "Thermal Radiation Heat Transfer", Third Edition, Hemisphere Publishing Co.*

[380] *Modest, Michael F. 2003. "Radiative Heat Transfer", Second Edition, Academic Press.*

[381] *Taylor, R.P. and Luck, R. 1995. "Comparison of Reciprocity and Closure Enforcement Methods for Radiation View Factors", J Thermophysics and Heat Transfer, 9:4, pp. 660-666.*

[382] *Holman, J.P. 2001. "Heat Transfer", Eighth SI Metric Edition, McGraw Hill.*

[383] *ASHRAE 1997. "ASHRAE Handbook-Fundamentals", Atlanta: American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc.*

[384] *Coppalle, A. and Vervisch, P. 1983. "The total emissivities of high-temperature flames," Combustion and Flame, 49(101), pp. 101-108.*

[385] *Hottel, H. C. 1954. "Radiant-Heat Transmission," Heat Transmission, 3rd Ed., W. H. McAdams (Ed.), McGraw-Hill, NY.*

[386] *Hottel, H. C. and Sarofim, A. F. 1964. Radiative Transfer, McGraw-Hill, NY.*

[387] *Incropera, F. P. and DeWitt, D. P. 1990. Fundamentals of Heat and Mass Transfer, 3rd Ed., Wiley, NY.*

[388] *Koch, R. and Becker, R. 2003. "Evaluation of quadrature schemes for the discrete ordinate method", Proceedings of Eurotherm 73 on Computational Thermal Radiation in Participating Media, pp. 15-17.*

[389] *Lathrop, K. D. and Carlson, B. G. 1965. "Discrete ordinates angular quadrature of the neutron transport equation", Report LA-3186, Los Alamos Scientific Laboratory.*

[390] *Modest, M.F. 1993. "Radiative Heat Transfer." McGraw-Hill, New York.*

[391] *Siegel, R. and Howell, J.R. 1992. "Thermal Radiation Heat Transfer", Third Edition, Hemisphere Publishing Co., Washington.*

[392] *SOLPOS – Solar Calculator, National Renewable Energy Laboratory, http://rredc.nrel.gov/solar/codesandalgorithms/solpos/.*

[393] *Iqbal, M. 1983. An Introduction to Solar Radiation, Academic Press, NY.*

[394] *Spencer, J. W. 1971. "Fourier series representation of the position of the sun", Search 2(5), p. 172.*

[395] *Michalsky, J. 1988. "The Astronomical Almanac's algorithm for approximate solar position (1950-2050)", Solar Energy 40(3), pp. 227-235.*

[396] *Zimmerman, J. C. 1981. “Sun-pointing programs and their accuracy”, SAND81-0761, Experimental Systems Operation Division 4721, Sandia National Laboratories, Albuquerque, NM.*

Porous Media

Porous media are continua that contain both fluid and fine-scale solid structures, for example: packed-bed chemical reactors, filters, radiators, honeycomb structures, or fibrous materials. The solid geometrical structures are too fine to be individually meshed and fully resolved by a computational grid.

STAR-CCM+ provides two approaches to model the effects of the porous medium on the flow:

- One way of modeling flow in a porous medium is to introduce source terms into the momentum transport equations to approximate the pressure losses. This approach is usually termed as superficial velocity formulation. In STAR-CCM+, it is called porous region modeling.
- The second modeling approach is more general in that it accounts for the increase in physical velocity when flow enters the porous medium. This approach is usually termed physical velocity formulation. In STAR-CCM+, it is called phasic porous media modeling. The phasic porous media model provides the ability to have multiple solid phases in the porous medium, for each of which a separate energy equation can be solved.

Both modeling approaches use the porosity and add appropriate source terms to the governing equations. The porosity is defined as the ratio of the volume V_f that is occupied by the fluid and the total volume V of a cell:

$$\chi = \frac{V_f}{V} \tag{2179}$$

When flow enters a porous medium, the physical flow velocity rises at a given point, because there is less open area available to the flow. Porous region modeling does not compute this increase in physical velocity, instead basing the approach on a calculated superficial velocity. The superficial velocity is related to the physical velocity as:

$$\mathbf{v}_s = \chi \cdot \mathbf{v} \tag{2180}$$

The superficial velocity is the same in the fluid region and in the porous region. The superficial velocity is an artificial flow velocity that assumes that only fluid passes the cross-sectional area and neglects the solid portion of the porous medium.

Note: For multiphase segregated flow simulations only, a blending is used to ensure a smooth transition from resistance-dominated to inertia/viscosity dominated regimes.

Diffusion in porous media depends on the porosity of the medium and on the tortuosity τ. Tortuosity is defined as the ratio between the actual (convoluted) path length through the porous medium from point to point and the straight-line distance between the same two points. Tortuosity reduces the rate of diffusion in a porous medium, so that the effective diffusion coefficient is:

$$D_{eff} = \frac{\chi}{\tau} D \tag{2181}$$

where D is the diffusion coefficient of the fluid. Tortuosity can be a factor in cases involving species transport, passive scalars, Li-ion transport, and electrical conductivity in the context of electromagnetics.

Turbulence in Porous Media

The turbulence transport equations are not solved for in the porous regions. The effect of a porous region on turbulent flow depends on its internal structure. Where turbulence is present, the turbulence scales are dominated by the geometric structure of the porous medium. As STAR-CCM+ does not predict these turbulence scales directly in the porous region, you must specify them directly in the form of either:

- turbulent kinetic energy and dissipation rate
- turbulence intensity and length scale
- turbulence intensity and viscosity ratio

Turbulence quantities in the fluid leaving the porous region are constructed from the user-defined values; they are not transported from the upstream side of the porous region.

Contents:

Superficial Velocity Formulation
Physical Velocity Formulation
Porous Baffle
Porous Media Bibliography

See Also:

Species Transport
Passive Scalars Transport
Lithium-Ion Model Reactions
Constitutive Relations

Superficial Velocity Formulation

STAR-CCM+ models the pressure losses in the porous region by introducing momentum sinks in the momentum equations. STAR-CCM+ includes the user-specified volume porosity in the unsteady terms and in the calculation of effective thermal conductivity.

Continuity Equation

The continuity equation that governs the mass balance in a porous medium then reads:

$$\frac{\partial}{\partial t}\left(\int_V \rho\chi\right)dV + \oint_A \rho\mathbf{v}_s \cdot d\mathbf{a} = \int_V S_u dV \tag{2182}$$

where $\mathbf{v}_s$ is the superficial velocity, ρ is the fluid density, $\mathbf{a}$ is the surface area vector, and S_u is a user-defined mass source or sink.

Momentum Equation

The momentum equation contains an additional source term that accounts for the resistance to the flow imparted by the porous medium. The porous medium resistance force is defined in terms of the superficial velocity $\mathbf{v}_s$ and the porous resistance tensor $\mathbf{P}$:

$$\mathbf{f}_p = -\mathbf{P} \cdot \mathbf{v}_s \tag{2183}$$

The porous resistance tensor consists of two components:

$$\mathbf{P} = \mathbf{P}_v + \mathbf{P}_i\left|\mathbf{v}_s\right| \tag{2184}$$

where $\mathbf{P}_v$ is the viscous (linear) and $\mathbf{P}_i$ is the inertial (quadratic) resistance tensor. This source term represents a momentum sink that creates a pressure drop.

The momentum balance for a porous medium is then given by:

$$\begin{aligned}\frac{\partial}{\partial t}\left(\int_V \rho\mathbf{v}_s\right)dV + \oint_A \rho\mathbf{v}_s \otimes \mathbf{v_s} \cdot d\mathbf{a} \; = \\ -\oint_A p\mathbf{I} \cdot d\mathbf{a} + \oint_A \mathbf{T} \cdot d\mathbf{a} + \int_V \mathbf{f}_b\, dV + \int_V \mathbf{f}_p\, dV\end{aligned} \tag{2185}$$

where $\mathbf{f}_b$ is the body force vector comprising all other volumetric body forces, $\mathbf{I}$ is the identity matrix, and $\mathbf{f}_p$ is the porous resistance tensor given by Eqn. (2183).

Energy Equation

The energy equation for a porous medium is given by:

$$\begin{aligned}\frac{\partial}{\partial t}\int_V \left(\left(\chi(\rho E)_{\text{fluid}} + \left(1-\chi\right)\left(\rho C_p\right)_{\text{solid}}\, T\right)dV\right)dV + \oint_A \rho H\mathbf{v}_s \cdot d\mathbf{a} = \\ -\oint_A \dot{\mathbf{q}}'' \cdot d\mathbf{a} + \oint_A \mathbf{T} \cdot \mathbf{v}_s d\mathbf{a} + \int_V \mathbf{f}_b \cdot \mathbf{v}_s dV + \int_V S_u dV\end{aligned} \tag{2186}$$

where H is total enthalpy and the heat flux $\dot{\mathbf{q}}''$ is defined as:

$$\dot{\mathbf{q}}'' = -k_{\text{eff}} \nabla T \tag{2187}$$

and

$$k_{\text{eff}} = \chi k_{\text{fluid}} + \left(1-\chi\right) k_{\text{solid}} \tag{2188}$$

where k_{eff} is the effective thermal conductivity of the porous medium, k_{fluid} is the thermal conductivity of the fluid, and k_{solid} is the thermal conductivity of the solid.

Influence of Porous Resistance on Temperature for Incompressible Flow

When you simulate incompressible flow through a porous medium and you solve for energy, a change in temperature proportional to the user-specified porous resistance takes place.

Consider the following example: a one-dimensional steady incompressible flow of a fluid with constant C_v (specific heat at constant volume) through a porous medium with a constant cross-section.

Mass conservation requires that the velocity v is constant. Momentum conservation reduces to a balance between the pressure gradient dp/dx and the porous resistance force f_p:

$$f_p = \frac{dp}{dx} \tag{2189}$$

The energy equation reduces to:

$$\frac{dH}{dx} = 0 \tag{2190}$$

and since total enthalpy is $H = C_v T + \frac{p}{\rho} + \frac{v^2}{2}$, the equation becomes:

$$\frac{dT}{dx} = \frac{-1}{\rho c_v}\frac{dp}{dx} = \frac{-f_p}{\rho c_v} \tag{2191}$$

Hence, it is normal to expect a temperature increase or decrease for an incompressible flow through a porous medium.

Physical Velocity Formulation

The porous media model in STAR-CCM+ uses the physical velocity formulation. This formulation accounts for the increase in velocity when the flow enters the porous medium.

In the equations below, velocity refers to physical velocity. With regard to energy, the porous media model in STAR-CCM+ provides either non-equilibrium or equilibrium modeling. The non-equilibrium model does not assume that fluid and solid phase are at the same temperature. It solves separate energy transport equations for the fluid and the solid phases. The equilibrium model, on the other hand, assumes that the fluid and solid phases are at the same temperature. It solves a single energy transport equation.

Continuity Equation

$$\frac{\partial(\chi\rho)}{\partial t} + \nabla \cdot (\chi\rho\mathbf{v}) = 0 \tag{2192}$$

where:

- χ is the porosity
- $\mathbf{v}$ is the physical velocity

Momentum Equation

$$\frac{\partial(\chi\rho\mathbf{v})}{\partial t} + \nabla \cdot (\chi\rho\mathbf{v}\mathbf{v}) = -\chi\nabla p + \nabla \cdot (\chi\mathbf{T}) - \chi\mathbf{P}_v\mathbf{v} - \chi\mathbf{P}_i|\mathbf{v}|\mathbf{v} \tag{2193}$$

where:

- $\mathbf{P}_i$ is the inertial resistance tensor
- $\mathbf{P}_v$ is the viscous resistance tensor

Thermal Non-Equilibrium Energy Equations

$$\frac{\partial\left(\chi\rho_{\text{fluid}}E_{\text{fluid}}\right)}{\partial t} + \nabla \cdot \left(\chi\rho_{\text{fluid}}H_{\text{fluid}}\mathbf{v}\right) = -\nabla \cdot \left(\chi\mathbf{q}_{\text{fluid}}\right) + \nabla \cdot (\chi\mathbf{T} \cdot \mathbf{v}) + ah(T_{\text{fluid}} - T_{\text{solid}}) \tag{2194}$$

where:

- E_{fluid} is the total energy of the fluid
- H_{fluid} is the total enthalpy of the fluid
- a is the interaction area density
- h is the heat transfer coefficient
- $\mathbf{T}$ is the stress tensor

$$\frac{\partial\left((1-\chi)\rho_s E_s\right)}{\partial t} = -\nabla \cdot ((1-\chi)\mathbf{q}) + ah(T_s - T_f) \tag{2195}$$

where:

- E_{solid} is the total energy of the solid
- $\mathbf{q}$ is the heat flux through conduction

Thermal Equilibrium Energy Equations

$$\frac{\partial(\rho \mathrm{E})_{\mathrm{eff}}}{\partial t} + \nabla \cdot \left(\chi \rho_{\mathrm{fluid}} H_{\mathrm{fluid}} \mathbf{v}\right) = \nabla \cdot \left(k_{\mathrm{eff}} \nabla T_{\mathrm{fluid}}\right) + \nabla \cdot \left(\chi \mathbf{T}_{\mathrm{fluid}} \cdot \mathbf{v}\right) \tag{2196}$$

$$(\rho E)_{\mathrm{eff}} = \chi \rho_{\mathrm{fluid}} E_{\mathrm{fluid}} + (1-\chi) \rho_{\mathrm{solid}} C_{p,\mathrm{solid}} T_{\mathrm{fluid}} \tag{2197}$$

$$k_{\mathrm{eff}} = \chi k_{\mathrm{fluid}} + (1-\chi) k_{\mathrm{solid}} \tag{2198}$$

where:

- $C_{p,\mathrm{solid}}$ is the specific heat capacity of the solid
- k_{eff} is an effective thermal conductivity that is calculated from the thermal conductivities of the solid k_{solid} and the fluid k_{fluid}.

Conditions at the Porous Interface

Across the porous interface, the physical velocity is discontinuous.

The following table lists the porous interface conditions for various variables:

Variable	Porous Interface Condition
Physical velocity	Discontinuous
Superficial velocity	Continous
Static pressure	Discontinuous
Total pressure	Continuous

Porous Baffle

There are situations when the porous medium thickness is small compared to the other dimensions, for example, the flow through a perforated plate or a wire screen. As a result, the flow resistance and heat transfer can be assumed to be one-dimensional. The treatment of the flow resistance is similar to porous media.

For porous baffles, the pressure drop Δp is specified as:

$$\Delta p = -\rho\left(P_i\left|v_{s,n}\right| + P_v\right)v_{s,n} \tag{2199}$$

where:

$v_{s,n}$ is the superficial velocity normal to the surface
P_i is the porous inertial resistance
P_v is the porous viscous resistance

The coefficients P_i and P_v effectively define the baffle permeability for one-dimensional flow. The method assumes that the direction of the flow is unchanged as it passes through the baffle.

Through the porous baffle, heat conduction can be taken into account with the following assumptions:

- conduction is one-dimensional
- thermal inertia is negligible
- the effective baffle area for heat conduction is given by $(1-\chi)\cdot \text{solid surface area}$

The total heat transfer across the porous baffle is computed as a linear combination of the heat transfer due to convection and diffusion in the fluid, and the conduction through the baffle. The porosity χ provides the linear weighting factor. This calculation is given by:

$$\dot{q}'' = (1-\chi)\frac{\Delta T}{\left(\frac{1}{h_1} + R_{\text{th}} + \frac{1}{h_2}\right)}|\mathbf{a}| + \chi(k\nabla T)\cdot\mathbf{a} - \dot{m}H \tag{2200}$$

where:

χ is the surface porosity that is defined as the ratio of the area of the holes to the area of the surface
ΔT is the temperature difference in the fluid on either side of the baffle
h_1, h_2 are the heat transfer coefficients on either side of the baffle
R_{th} is the thermal resistance per unit area
k is the thermal conductivity of the baffle
$\dot{m}$ is the mass flow rate
H is the fluid enthalpy

For heat transfer analysis, the porous baffle offers a thermal contact resistance that can be defined as an effective resistance or consisting of multiple layers. Each layer represents a different material with different thermal conductivities. The thermal resistance is calculated as:

$$\frac{1}{R_{\text{th}}} = \sum_{i=1}^{n}\frac{k_i}{\Delta x_i} \tag{2201}$$

where k_i and Δx_i are the conductivity and thickness, respectively, of each of the n layers comprising the baffle.

Porous Media Bibliography

[397] *Huang, H., and Ayoub, J., March 2008. "Applicability of the Forchheimer Equation for Non-Darcy Flow in Porous Media". SPE Journal, pp. 112-122.*

Species and Passive Scalars

STAR-CCM+ models the transport, the mixing, and the chemical reactions of multi-component liquids or multi-component gases by solving conservation equations for scalar variables that represent the mass fraction of each species in the mixture. This conservation equation allows for convection, diffusion, and optional source effects, and is solved in addition to the global mass continuity equation.

For a multi-component gas or liquid mixture that is modeled through species, the mixture composition influences the fluid's bulk material properties such as density, molecular transport properties, absorption, and so on depending on the mass fraction.

You can also model additional scalar components in your simulation that do not influence the material properties of the fluid by using passive scalars. A similar conservation equation considering convection, diffusion, and optional sources is solved to track the distribution of the passive scalar. You can use passive scalars as tracers in the fluid flow that do not have any mass or volume.

Contents:

Species Transport
Multi-Component Mixture Material Properties
Passive Scalars Transport

Species Transport

A particular species is defined as one component in a multi-component gas mixture or liquid mixture.

The transport equation for the mass fraction Y_i of a species is given by:

$$\frac{d}{dt}\int_V \rho Y_i \, dV + \oint_A \rho Y_i \mathbf{v} \cdot d\mathbf{a} = \oint_A \left[J_i + \frac{\mu_t}{\sigma_t} \nabla Y_i \right] \cdot d\mathbf{a} + \int_V S_{Y_i} dV \tag{2202}$$

where:

- t is time
- V is volume
- $\mathbf{a}$ is the area vector
- i is the component index.
- ρ is the overall density.
- $\mathbf{v}$ is the velocity.
- μ_t is the turbulent dynamic viscosity.
- σ_t is the turbulent Schmidt number
- S_{Y_i} is a user specified region source term, or a source term resulting from reactions, for species i.
- J_i is the laminar (or molecular) diffusive flux.

Turbulent diffusion is accounted for by the term $\frac{\mu_t}{\sigma_t}$. Laminar diffusion is defined by J_i.

Fick's Law

When multi-component diffusion is deactivated, Fick's Law is used to calculate the diffusive flux J_i, given by:

$$J_i = \rho D_{i,m} \nabla Y_i \tag{2203}$$

where $D_{i,m}$ is the molecular diffusivity of component i in the mixture.

Multi-Component Diffusion

When multi-component diffusion is activated, the scalar diffusivity of Fick's Law is replaced by a matrix, and the diffusive flux for component i becomes a function of the mass fractions for all components:

$$J_i = \rho \sum_{j=1}^{N} D_{i,j} \nabla Y_j \tag{2204}$$

$D_{i,j}$ represents the multi-component diffusion coefficients which are calculated using the Maxwell-Stefan equations. These equations define the diffusive flux implicitly as a function of mole fraction gradients, which are given as:

$$\nabla X_i = \frac{M_w}{\rho} \sum_{j=1, j \neq i}^{N} \frac{X_i J_j}{M_j \mathfrak{D}_{i,j}} - \frac{X_j J_i}{M_i \mathfrak{D}_{i,j}} \tag{2205}$$

$\mathfrak{D}_{i,j}$ represents the binary diffusion coefficient between components i and j. Writing these equations in matrix form gives:

$$[B] \nabla Y = [A] J \tag{2206}$$

where $[B]$ represents the mapping from mass fraction gradients to mole fraction gradients and $[A]$ represents the Maxwell-Stefan equations. By inverting $[A]$ and multiplying by the matrix $[B]$, the multi-component diffusion coefficients are calculated:

$$D_{i,j} = \rho \sum_{k=1}^{N} A_{i,k}{}^{-1} B_{k,j} \tag{2207}$$

Soret Effect

When the Soret Effect option is activated, an additional term is added to the diffusive flux J_i for each component. In the case that multi-component diffusion is deactivated, the expression for the diffusive flux becomes [32]:

$$J_i = \rho D_{i,m} \nabla Y_i + \rho \frac{D_{i,t}}{T} \nabla T \tag{2208}$$

and when multi-component diffusion is activated, the expression becomes [31]:

$$J_i = \rho \left(\sum_{j=1}^{N} D_{i,j} \nabla Y_j \right) + \rho \frac{D_{i,t}}{T} \nabla T \tag{2209}$$

where $D_{i,t}$ is the thermal diffusion coefficient for component i.

Thermal Diffusion

For the Warnatz Model, the thermal diffusion coefficient $D_{i,t}$ in Eqn. (2208) or Eqn. (2209) is computed based on the thermal diffusion ratio. The thermal diffusion ratio between component i and component j is given as [33]:

$$K_{t,ij} = \frac{15}{2} \frac{\left(2A^*_{i,j} + 5\right)\left(6C^*_{i,j} - 5\right)}{A^*_{i,j}\left(16A^*_{i,j} - 12B^*_{i,j} + 55\right)} \frac{M_i - M_j}{M_i + M_j} X_i X_j \tag{2210}$$

and:

$$\begin{aligned} A^*_{i,j} &= \frac{1}{2} \frac{\Omega^{(2,2)}_{i,j}}{\Omega^{(1,1)}_{i,j}} \\ B^*_{i,j} &= \frac{1}{3} \frac{5\Omega^{(1,2)}_{i,j} - \Omega^{(1,3)}_{i,j}}{\Omega^{(1,1)}_{i,j}} \\ C^*_{i,j} &= \frac{1}{3} \frac{\Omega^{(1,2)}_{i,j}}{\Omega^{(1,1)}_{i,j}} \end{aligned} \tag{2211}$$

where:

- $\Omega_{i,j}$ represents collision integrals between component i and component j.

- M_i is the molecular weight of component i

Using the thermal diffusion ratio, the thermal diffusion coefficient for component i is given as:

$$D_{i,t} = D_{i,m} \frac{M_i}{M_w} \sum_{j=1}^{N} K_{t,ij} \tag{2212}$$

where $D_{i,m}$ is the effective molecular diffusivity of component i into the mixture and M_w is the mean molecular weight of the mixture.

For N components, STAR-CCM+ solves transport equations for all species and ensures that all mass fractions sum to 1.

Species Transport in Porous Media

In porous media, the diffusive flux is modified by the porosity χ and tortuosity τ of the media.

For Fick's Law:

$$J_i = \frac{\chi}{\tau} \rho D_{i,m} \nabla Y_i$$

For multi-component diffusion:

$$J_i = \rho \frac{\chi}{\tau} \sum_{j=1}^{N} D_{i,j} \nabla Y_j$$

For the Soret effect:

$$J_i = \rho \frac{\chi}{\tau} D_{i,m} \nabla Y_i + \rho \frac{\chi}{\tau} \frac{D_{i,t}}{T} \nabla T$$

See Also:

Porous Media

Multi-Component Mixture Material Properties

When simulating reacting or non-reacting species transport in terms of a multi-component gas mixture or a multi-component liquid mixture, the material properties of the mixture (not the material properties of the mixture components) are usually calculated as functions of mass fraction or mole fraction of the mixture components.

Mass-Weighted Mixture

The mass-weighted mixture method calculates a given mixture property by mass-weighting the component property values.

For example, mixture property ϕ_{mix} is:

$$\phi_{mix} = \sum_{i=1}^{N} Y_i \phi_i \tag{2213}$$

where Y_i and ϕ_i are the mass fraction and property values of mixture component i and N is the total number of components in the mixture.

Binary Diffusion Coefficient Method for Molecular Diffusivity

This method uses specified binary diffusion coefficients between the components of the mixture to calculate a molecular diffusivity for each component that is diffusing into the mixture as a whole.

For a mixture containing N components, this method requires the specification of $(N(N-1))/2$ diffusion coefficients.

This specification enforces the fact that self-diffusion is zero and the binary diffusion coefficients are symmetric. An array profile specifies these binary diffusion coefficients with all the standard methods available for specifying the individual profiles. The diffusivity of each component is calculated as:

$$D_{i,m} = \frac{1 - X_i}{\sum_{j\, j \neq i} \frac{X_j}{\mathfrak{D}_{i,j}}} \tag{2214}$$

where:

- X_i, X_j are the mole fractions of component i and j, respectively.
- $D_{i,m}$ is the molecular diffusivity of component i in the mixture.
- $\mathfrak{D}_{i,j}$ is the binary diffusion coefficient of components i and j

Schmidt Number

The Schmidt number function is available for specifying molecular diffusivity of components in a mixture.

The Schmidt number is a non-dimensional parameter that is defined as:

$$\mathrm{Sc}_i = \frac{\nu}{D_{i,m}} \tag{2215}$$

where ν is the kinematic viscosity.

Kinetic Theory Method for Molecular Diffusivity

This method provides the molecular diffusivity of components in a mixture $D_{i,m}$ using kinetic theory. $D_{i,m}$ is calculated as in Eqn. (2214).

From gas kinetic theory, the expression for the binary diffusion coefficient $\mathfrak{D}_{i,j}$ is based on Chapman-Enskog [26]:

$$\mathfrak{D}_{i,j} = \frac{2.66 \times 10^{-7} T^{3/2}}{p M_{i,j}^{1/2} \sigma_{i,j}^2 \Omega(T^*)} \tag{2216}$$

and:

$$M_{i,j} = \frac{2 M_i M_j}{M_i + M_j} \tag{2217}$$

where:

- M_i, M_j are the molecular weights of component i and j, respectively.
- $\sigma_{i,j}$ is the Lennard-Jones characteristic length, the collision diameter for the pair of component i and component j. $\sigma_{i,j}$ is a function of the Lennard-Jones collision diameter material property, σ_i, that is specified for each component i of the mixture. $\sigma_{i,j}$ is different for polar and non-polar molecules (gases).
- p is the (absolute) static pressure
- $\Omega(T^*)$ is a collision integral that is a function of the reduced temperature, T^*. T^* is defined similarly to the reduced temperature for the viscosity calculation:

$$T^* = \frac{kT}{\varepsilon_{i,j}} \tag{2218}$$

where:

- k is the Boltzmann constant $= 1.3806503 \times 10^{-23}$ m^2 kg s^{-2} K^{-1}
- T is temperature
- $\varepsilon_{i,j}$ is the characteristic Lennard-Jones energy for the pair of component i and component j. $\varepsilon_{i,j}$ is a function of the Lennard-Jones potential energy material property, ε_i, that is specified for each component i of the mixture. $\varepsilon_{i,j}$ is different for polar and non-polar molecules (gases).

Lewis Number

The Lewis number Le is defined as the ratio of thermal diffusion to mass diffusion D_m.

The Lewis number is expressed as follows:

$$\mathrm{Le} \equiv \frac{k/\rho C_p}{D_m} \tag{2219}$$

where ρ, C_p, and k are the density, specific heat, and thermal conductivity of the mixture, respectively.

Given values for L_e, D_m, ρ, and C_p, thermal conductivity k can be calculated as follows:

$$k = L_e D_m \rho C_p \tag{2220}$$

Mixture Method for Critical Temperature and Pressure

The mixture method calculates the critical temperature and pressure of a mixture.

The mixture method calculates the critical temperature $T_{c,m}$ of a mixture as:

$$T_{c,m} = \frac{\sum_i T_{c,i} Y_i / W_i}{\sum_i Y_i / W_i} \tag{2221}$$

where W_i is the molecular weight, Y_i the mass fraction and $T_{c,i}$ the critical temperature of species i. The summation is over all species in the mixture. The critical pressure $P_{c,m}$ of a mixture is similarly calculated as:

$$P_{c,m} = \frac{\sum_i P_{c,i} Y_i / W_i}{\sum_i Y_i / W_i} \tag{2222}$$

Mixture Method for Surface Tension

When the mixture method is used for surface tension, the mixture surface tension is:

$$\sigma_m{}^r = \sum_i X_i \sigma_i{}^r \tag{2223}$$

where σ_i is the surface tension of component i, X_i is its mole fraction, and r is the exponent property of the mixing law.

Mixture Method for Molecular Weight

When the mixture method is used for molecular weight, the mixing law is:

$$W_m = \frac{1}{\sum_i Y_i / W_i} \tag{2224}$$

where W_i is the molecular weight of species i, Y_i is the mass fraction of species i, and the summation is over all species in the mixture.

Mathur-Saxena Averaging Method for Dynamic Viscosity and Thermal Conductivity

The Mathur-Saxena averaging methods for dynamic viscosity and thermal conductivity are specifications of the multi-component Mathur-Saxena Averaging Property method <Φ> where Φ is a material property. This method uses the following formula to compute the property on the mixture level, using values that are given for the individual mixture components (X_i is the mole fraction of component i):

$$\Phi = \frac{1}{2}\left(\sum_1^n X_i \Phi_i + \left(\sum_1^n \frac{X_i}{\Phi_i}\right)^{-1}\right) \tag{2225}$$

Passive Scalars Transport

Passive scalars are called passive, because they do not affect the physical properties of the simulation. You can use more than one passive scalar in a simulation.

The following situations are examples where passive scalars can be useful:

- Choosing the best position for a sensor while designing a part with multiple inlet pipes and an exhaust pipe.

  You can use the passive scalars to check the scalar values at discrete points in the computational domain. Then you can determine which of the scalars sends the strongest signal to the sensor.
- Analyzing the mixing of two fluid streams that have the same properties.

  The fluid is represented as single-phase, but you can use multiple passive scalars to examine the effect of mixing.
- Tracing how smoke or any other vapor would convect and diffuse in a room or any fluid domain.

  More specifically, you can model the dissolution of a gas in a liquid.

The transport equation for the passive scalar component ϕ_j is:

$$\frac{\partial}{\partial t}\int_V \rho\phi_i dV + \oint_A \rho\phi_i \mathbf{v}\cdot d\mathbf{a} = \oint_A \mathbf{j}_i \cdot d\mathbf{a} + \int_V S_{\phi_i} dV \tag{2226}$$

where:

- i is the component index.
- $\mathbf{j}_i$ is the diffusion flux.
- S_{ϕ_i} is a source term for passive scalar component i.
- ϕ_i is assumed to be positive-definite.

Linear Eddy-Diffusivity

In the linear eddy-diffusivity model, the diffusion flux is:

$$\mathbf{j}_i = \left(\frac{\mu}{\sigma} + \frac{\mu_t}{\sigma_t}\right)\nabla\phi_i$$

where:

- μ is the viscosity.
- μ_t is the turbulent viscosity.
- σ is molecular Schmidt number—a material property.
- σ_t is the turbulent Schmidt number.

Generalized Gradient Diffusion Hypothesis (GGDH)

The Generalized Gradient Diffusion Hypothesis model (GGDH) provides improved prediction for passive scalar diffusion in highly anisotropic turbulent flows. This hypothesis models the passive scalar diffusion flux as a tensor quantity proportional to the Reynolds stresses:

$$\mathbf{j}_i = \frac{\mu}{\sigma}\nabla\phi_i + \frac{C_s}{\sigma_t}\frac{k}{\epsilon}\mathbf{R}\cdot\nabla\phi_i \tag{2227}$$

where:

- ϕ_i is the value of the passive scalar component i.
- $C_s = 0.2$ is a model constant.
- k is kinetic energy.

  ϵ is the dissipation rate.
- $\mathbf{R}$ is the Reynolds stress tensor.

Passive Diffusion in Porous Media

In porous media, the passive scalar diffusion flux $\mathbf{j}_i$ is multiplied by the porosity/tortuosity ratio χ/τ.

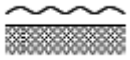
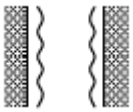
See Also:

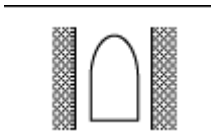
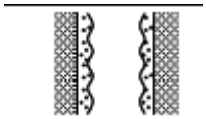
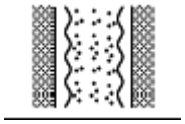
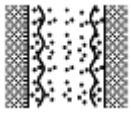
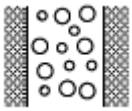
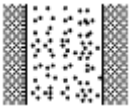
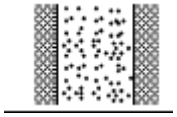
Porous Media

Eulerian Multiphase Flow

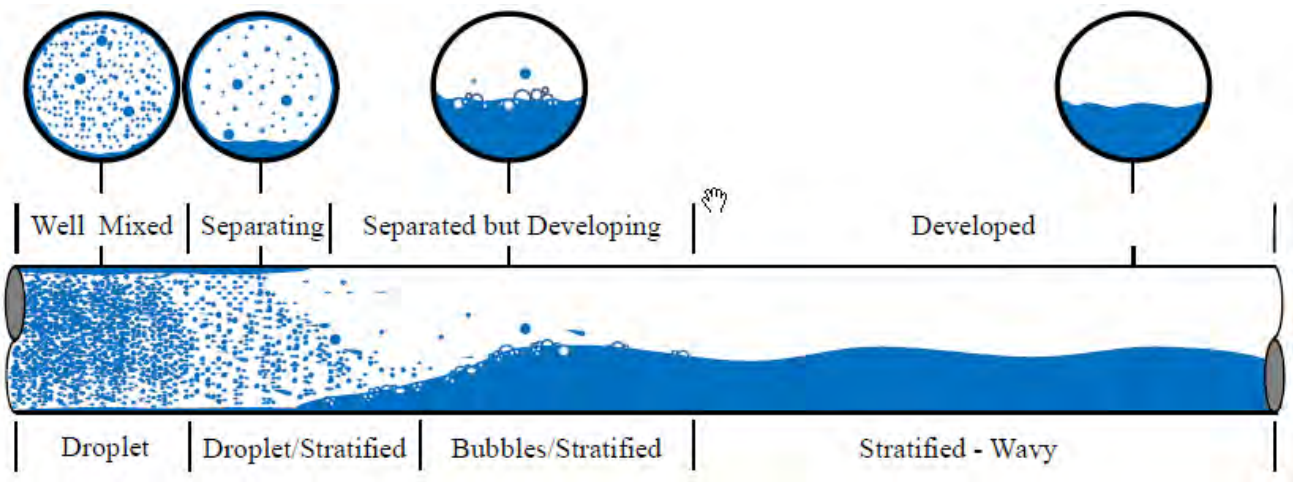
Multiphase flows, where several fluids flow in the domain of interest, play an important role in variety of industrial applications. In general, we associate phases with gases, liquids or solids and as such some simple examples of multiphase flows are: air bubbles rising in a glass of water, sand particles carried by wind, rain drops in air. The definition of *phase* can be generalized and applied to other fluid characteristics such as size, shape, density, and temperature.

Numerical simulations of such flows need to handle additional complexity in comparison to simulating single phase flow due to the presence of interface across which there is a jump in fluid-fluid properties and the exchange of mass, momentum and heat between the phases occurs as well. For modelling considerations, such flows can be classified based on increasing spatial scales of interface between the phases, into dispersed (bubbly flow, droplet flow), mixed / intermittent (slug flow, churn flow) and separated / stratified (film flow, annular flow, horizontal stratified flow).

Type	Flow Regime	Geometry	Characteristics	Examples
Stratified flows	Film Flow		• Liquid film in gas • Gas film in liquid	Film condensation Film boiling
	Annular Flow		• Liquid core and gas film • Gas core and liquid film	• Film boiling • Boilers
	Jet Flow		Gas pocket in liquid	• Atomization • Jet Condenser
	Free surface		Tank filled with liquid and gas above	• Marine towing tank and waves • Sloshing
Mixed or transitional flows	Slug flow		Gas pocket in liquid	Sodium boiling in forced convection

Type	Flow Regime	Geometry	Characteristics	Examples
				
	Bubbly annular flow		Gas bubbles in liquid film with gas core	Evaporators with wall nucleation
	Droplet annular flow		Gas core with droplets and liquid film	Steam generator
	Bubbly droplet annular flow		Gas core with droplet and liquid film with gas bubbles	Boiling nuclear reactor channel
Dispersed flows	Bubbly flow		Gas bubbles in liqud	Chemical reactors
	Droplet flow		Liquid droplets in gas	Spray cooling
	Particulate flow		Solid particles in gas or liquid	Transportation of powder

In several practical applications, mostly more than one flow regimes exist within the domain. The following diagram shows an example of such flow:



The volume tracking / interface capturing methods such as VOF [474] or level-set methods [497] are well suited for simulation of stratified multiphase flows which are characterized by large scale interface. It must be noted that the size of computational cell is critical in classifying the interface as large scale or a small scale. Since the volume tracking / interface capturing methods rely on resolving the interface completely, they are prohibitively expensive for simulating multi-scale flows where modelling of dispersed regime accurately is equally important.

The two fluid model [424] [440], referred to as multiphase segregated flow model in STAR-CCM+, on the other hand, is successfully used for modelling dispersed flows as well as multi-scale flow situations where modeling of dispersed flow along with separated flow accurately is equally important.

Contents:

Multiphase Segregated Flow
Volume of Fluid Method
VOF Bibliography
Fluid Film

Multiphase Segregated Flow

The two-fluid model that is referred to as multiphase segregated flow model in STAR-CCM+, and whose initial development was driven largely by nuclear industry, is generally used for modeling dispersed flows.

The multiphase segregated flow model treats each phase as inter-penetrating continua. It solves transport equations for mass, momentum and energy of each phase while all the phases share a common pressure field. Eulerian averaging [441] [511] [425] of the transport equations results in additional interaction between the phases. These interactions require models for closure like drag force, lift force, interphase heat transfer, and so on.

Volume Fraction

The share of the flow domain that is occupied by each phase is given by its volume fraction.

The volume of a phase i is given by:

$$V_i = \int_V \alpha_i \, dV \tag{2228}$$

where α_i is the volume fraction of phase i. The volume fractions of each Eulerian phase must fulfill the requirement:

$$\sum_{i=1}^{n} \alpha_i = 1 \tag{2229}$$

where n is the number of all phases.

Continuity Equation

The conservation of mass for a generic phase i is given by:

$$\frac{\partial}{\partial t}\int_V \alpha_i \rho_i \, dV + \oint_A \alpha_i \rho_i \mathbf{v}_i \cdot d\mathbf{a} = \int_V \sum_{j \neq i} \left(m_{ij} - m_{ji}\right) dV + \int_V S_i^{\alpha} dV \tag{2230}$$

where:

- α_i is the volume fraction
- ρ_i is the density
- $\mathbf{v}_i$ is the velocity
- m_{ij} is the mass transfer rate to phase i, from phase j ($m_{ij} \geq 0$)
- m_{ji} is the mass transfer rate to phase j, from phase i ($m_{ji} \geq 0$)
- S_i^{α} is a user-defined phase mass source term

Momentum Equation

The momentum balance for the generic phase i reads:

$$\frac{\partial}{\partial t}\int_V \alpha_i\rho_i\mathbf{v}_i\,dV + \oint_A \alpha_i\rho_i\mathbf{v}_i\otimes\mathbf{v}_i\cdot d\mathbf{a} = -\int_V \alpha_i\nabla p\;dV$$
$$+\int_V \alpha_i\rho_i\mathbf{g}\;dV + \oint_A \left[\alpha_i\left(\mathbf{T}_i+\mathbf{T}_i^t\right)\right]\cdot d\mathbf{a} + \int_V \mathbf{M}_i\,dV \tag{2231}$$
$$+\int_V \left(\mathbf{F}_{int}\right)_i dV + \int_V \mathbf{S}_i^\alpha dV + \int_V \sum_{i=1}^{n}\left(m_{ij}\mathbf{v}_j - m_{ji}\mathbf{v}_i\right)d\,V$$

where:

- p is the pressure, assumed to be equal in all phases
- $\mathbf{g}$ is the gravity vector
- $\mathbf{T}_i$ and $\mathbf{T}_i^t$ are the molecular and turbulent stresses, respectively
- $\mathbf{M}_i$ is the interphase momentum transfer per unit volume
- $\left(\mathbf{F}_{int}\right)_i$ represents internal forces (such as the solid pressure force between particles or a user-defined potential force)
- $\mathbf{S}_i^\alpha$ is the phase momentum source term
- m_{ij} is the mass transfer rate from phase j to phase i.
- m_{ji} is the mass transfer rate from phase i to phase j.

The interphase momentum transfer represents the sum of all the forces the phases exert on one another and satisfies the equation:

$$\sum_i M_i = 0 \tag{2232}$$

Energy Equation

Energy is conserved for a Eulerian phase according to the following equation:

$$\frac{\partial}{\partial t}\int_V \alpha_i\rho_i E_i\,dV + \oint_A \alpha_i\rho_i H_i\mathbf{v}_i\cdot d\mathbf{a} + \oint_A \alpha_i p\,d\mathbf{a} = \oint_A \alpha_i k_{eff,i}\nabla T_i\,d\mathbf{a} + \oint_A \mathbf{T}_i\cdot\mathbf{v}_i\,d\mathbf{a}$$
$$+\int_V \mathbf{f}_i\cdot\mathbf{v}_i\,dV + \int_V \sum_{j\neq i} Q_{ij}dV + \int_V \sum_{(ij)} Q_i^{(ij)}dV + \int_V S_{u,i}dV + \int_V \sum_{j\neq i}\left(m_{ij}-m_{ji}\right)h_i\left(T_{ij}\right)dV \tag{2233}$$

where:

- E_i is the total energy
- H_i is the total enthalpy
- $\mathbf{T}_i$ is the viscous stress tensor
- α_i, ρ_i and $\mathbf{v}_i$ are the volume fraction, density, and velocity of phase i, as in Eqn. (2230).
- T_i is the temperature.
- $k_{eff,i}$ is the effective thermal conductivity, from Eqn. (2234).
- $\mathbf{f}_i$ is the body force vector.

- Q_{ij} is the interphase heat transfer rate to phase i from phase j. It refers to the heat transfer between phases due to a temperature difference between the phases.
- $Q_i^{(ij)}$ is the heat transfer rate to phase i from phase pair interface (ij). It refers to the heat transfer phenomena such as evaporation, condensation, and boiling where heat transfer occurs between the phase and the interface.
- $S_{u,i}$ is the energy source.
- $h_i\left(T_{ij}\right)$ is the phase i enthalpy that is evaluated at the interface temperature T_{ij}.

The effective thermal conductivity k_{eff} is:

$$k_{eff,i} = k_i + \frac{\mu_{t,i} C_{p,i}}{\sigma_{t,i}} \tag{2234}$$

where:

- k_i is the thermal conductivity.
- $\mu_{t,i}$ is the turbulent viscosity.
- $C_{p,i}$ is the specific heat.
- $\sigma_{t,i}$ is the turbulent thermal diffusion Prandtl number.

Now, total energy, E, is related to the total enthalpy, H, by

$$E_i = H_i - \frac{p}{\rho_i} \tag{2235}$$

where:

$$H_i = h_i + \frac{\left|\mathbf{v}_i\right|^2}{2} \tag{2236}$$

and

$$h_i\left(T_i\right) = h_i^{REF} + \int_{T_i^{REF}}^{T_i} C_{p,i}\left(\mathrm{T}'\right)\ \mathrm{d}\ \mathrm{T}' \tag{2237}$$

where h_i^{REF} is the heat of formation for phase i and T_i^{REF} is the standard state temperature for phase i.

When heat is transferred only by thermal diffusion between phases, the interphase heat transfer rates satisfy $Q_{ij} = -Q_{ji}$. Where heat and mass are only transferred by phase change (bulk boiling or condensation), the heat transfer rates between each phase i,j, and a phase pair interface (ij) satisfy a heat balance:

$$Q_i^{(ij)} + Q_j^{(ij)} + \left(m_{ij} - m_{ji}\right)\Delta h_{ij} = 0 \tag{2238}$$

where Δh_{ij} is the heat required to produce phase i from phase j. This is a signed quantity so that $\Delta h_{ij} = -\Delta h_{ji}$. The heat of phase change is defined from the phase enthalpy definitions that are evaluated at the interface temperature T_{ij}:

$$\Delta h_{ij} = h_j\left(T_{ij}\right) - h_i\left(T_{ij}\right) \tag{2239}$$

Contents:

Phase Interaction Topology
Interaction Length Scale
Interaction Area Density
Momentum Transfer
Heat and Mass Transfer
Size Distribution
Large Scale Interfaces
Particulate Flows
Rheology of Emulsions and Suspensions
Multiphase Turbulence
Boundary Conditions
Multiphase Segregated Flow Bibliography

Phase Interaction Topology

The two-fluid model provides a flexible framework where dispersed, stratified or mixed two-phase flow can be modeled by using appropriate closure relations. For a given pair of phases, the multiphase segregated flow model in STAR-CCM+ can model different types of phase topology.

Continuous-Dispersed

The continuous-dispersed phase interaction topology provides the means to model the interaction of a phase which is dispersed within another phase throughout the domain at all times.

In the following equations, the subscript c denotes the *continuous phase* and the subscript d stands for *dispersed phase*. The subscript ij describes an interaction between phases i and j where i and j can be continuous or dispersed.

Multiple Flow Regimes

The multiple flow regime topology is an extended topology definition where all types of multiphase flows, that is, dispersed (for example bubble, droplet, and particle flows) and stratified (for example free-surface flows) flows can be modelled. An example of such a multiple flow regime topology is shown in the diagram in the previous section.

In several practical situations, it is desirable to simulate both dispersed and separated two-phase flow. To achieve this combined simulation, the flow topology must be classified into dispersed and separated two-phase flow. STAR-CCM+ employs a volume-fraction based criterion to classify the flow regime into dispersed and separated two-phase flow.

As a first step, the co-existence of dispersed and separated two-phase flow entails that the pair of phases must be a set of primary (p) and secondary (s) phase rather than a pair of continuous and dispersed phase. Based on this definition, for a pair of primary and secondary phase, three flow regimes are defined based on the volume fraction of the secondary phase:

Classification Criterion	Flow Regime	Flow Topology
$0 < \alpha_s < \alpha_{fr}$	First (dispersed) regime	Secondary phase is dispersed in primary phase.
$\alpha_{fr} < \alpha_s < \alpha_{sr}$	Interface regime	Both phases are separated.
$\alpha_{sr} < \alpha_s < 1.0$	Second (dispersed) regime	Primary phase is dispersed in secondary phase

where:

- α_s is the volume fraction of the secondary phase
- α_{fr} is the terminus of the first regime. It is the value of α_s at which the first regime transits to the interface regime. The default value is taken as 0.3 based on literature [411].
- α_{sr} is the onset of the second regime. It corresponds to the value of α_s at which the interface regime transits to the second regime. The default value is taken as 0.7 based on literature [411].

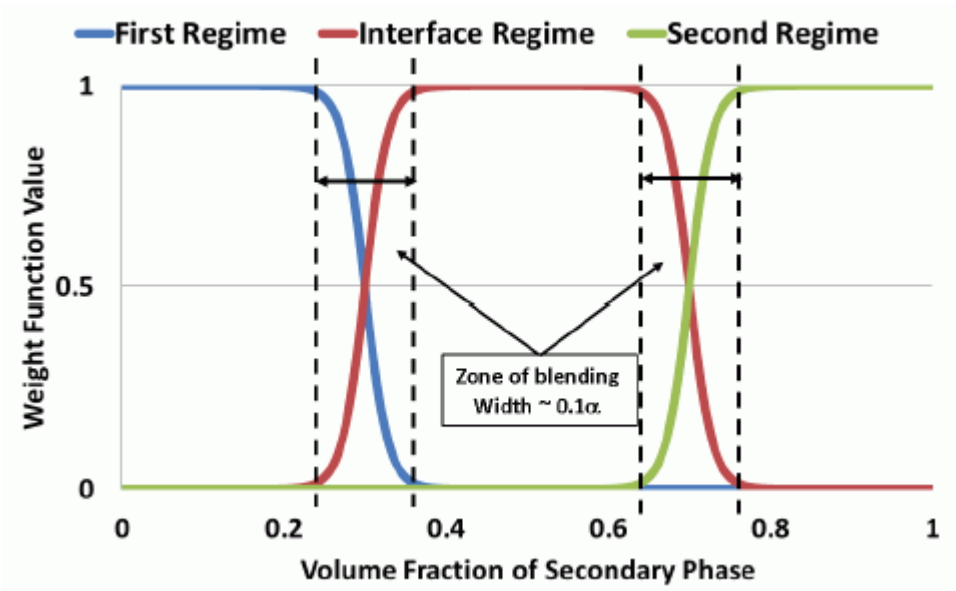
To understand this regime definition more clearly, consider a system of water as primary phase and air as secondary phase. In this scenario, the multiple flow regime model treats bubbly flow in the first dispersed regime, droplet flow in the second dispersed regime, and separated water-air flow in the large scale interface regime.

To enforce the classification of flow regime in each computational cell, the total linearized drag A and the heat transfer coefficient h are calculated by use of the weighted sum of the interaction of each flow topology regime as follows:

$$A = \sum_{r = \mathrm{fr, ir, sr}} W_r A_r \tag{2240}$$

$$h = \sum_{r = \mathrm{fr, ir, sr}} W_r h_r \tag{2241}$$

where W_r is the weight function. The weight functions depending on the volume fraction of the secondary phase are shown in the following plot:



Interaction Length Scale

For continuous-dispersed phase interactions, the interface length scale is taken to be an effective mean diameter of the dispersed phase particles. Where particles are not spherical, this is absorbed as a correction factor into interphase transfer models such as for bubble drag and lift force.

In general, the Sauter mean diameter is preferred since this gives the correct mean interfacial area where there is a distribution of particle sizes. If the S-Gamma model or the AMUSIG model for predicting particle size distribution is used, you can specify the interaction length scale through the Sauter mean diameter.

The interaction length scale, l_{cd} in a continuous-dispersed phase interaction, can be provided as a constant or a field function. For boiling flows, when boiling mass transfer is active, the Kurul Podowski length scale is provided as an alternative.

For the first and second regimes of a multiple flow regime topology, the two interaction length scales l_{fr} and l_{sr} refer to typical bubble and droplet diameters, respectively.

Interaction Area Density

The interaction area density specifies the interfacial area available for momentum, heat, and mass transfer between each pair of phases in an interaction.

Heat and mass transfer models use the interaction area density directly.

For Eulerian multiphase flow, drag models use one quarter of the interfacial area as an estimate of the projected area. Any correction factors, such as for non-spherical particle shapes or particle crowding, are assumed to be covered in correlations for the drag, heat, and mass transfer coefficients. There are two built-in models for interaction area density in STAR-CCM+: spherical particle and symmetric.

Spherical Particle Interaction Area Density

This is based on the surface area of a spherical particle, and is not applicable at high particle volume fractions. For this model, the interaction area density a_{cd} between a continuous phase (subscript c) and a dispersed phase (subscript d) is:

$$a_{cd} = \frac{6\alpha_d}{l_{cd}} \tag{2242}$$

where α_d is the volume fraction for the dispersed phase in the interacting phase pair and l_{cd} is the interaction length scale for the pair.

Symmetric Particle Interaction Area Density

The availability of the continuous phase modifies the symmetric particle interaction area density, and so can be used for a wide range of volume fractions:

$$a_{cd} = \frac{6\alpha_c\alpha_d}{l_{cd}} \tag{2243}$$

where:

- α_d is the volume fraction for the dispersed phase in the interacting phase pair
- α_c is the volume fraction for the continuous phase in the pair
- l_{cd} is the interaction length scale for the pair.

Momentum Transfer

The momentum transfer between phases represents the sum of all the forces that the phases exert on one another.

In STAR-CCM+, the momentum transfer term in the momentum equation Eqn. (2231) includes the following contributions:

$$\mathbf{M}_i = \sum_{j \neq i}\left(\mathbf{F}_{ij}^{\mathrm{D}} + \mathbf{F}_{ij}{}^{\mathrm{VM}} + \mathbf{F}_{ij}{}^{\mathrm{L}} + \mathbf{F}_{ij}^{\mathrm{TD}} + \mathbf{F}_{ij}^{\mathrm{WL}}\right) \tag{2244}$$

where:

- $\mathbf{F}_{ij}^{\mathrm{D}}$ is the drag force
- $\mathbf{F}_{ij}^{\mathrm{VM}}$ is the virtual mass force
- $\mathbf{F}_{ij}^{\mathrm{L}}$ is the lift force
- $\mathbf{F}_{ij}^{\mathrm{TD}}$ is the turbulent dispersion force
- $\mathbf{F}_{ij}^{\mathrm{WL}}$ is the wall lubrication force

The forces are defined as force per cell volume, on phase i, due to phase j. It follows that:

$$\mathbf{F}_{ji} = -\mathbf{F}_{ij} \tag{2245}$$

Contents:

Drag
Virtual Mass
Lift
Turbulent Dispersion
Wall Lubrication

Drag

For general applications, the inter-phase drag force is computed as a function of the drag coefficient.

For a continuous-dispersed phase interaction, the force acting on the dispersed phase i due to the drag of phase j is implemented as:

$$\mathbf{F}_{ij}{}^{D} = A_D \mathbf{v}_r \tag{2246}$$

where A_D is the linearized drag coefficient and $\mathbf{v}_r$ is the relative velocity between the phases i and j. A_D acts as a linear multiplier of the relative velocity that is defined as:

$$\mathbf{v}_r = \mathbf{v}_j - \mathbf{v}_i \tag{2247}$$

For a multiple flow regime phase interaction, the total drag force is calculated as:

$$\mathbf{F}_{ij}^{D} = \sum_{t\,=\,fr,\,ir,\,\mathrm{sr}} W_t A_{D,t} \mathbf{v}_r \tag{2248}$$

where W_t is the weight function.

The calculation of the linearized drag coefficient is analogous to the continuous-dispersed phase interaction. For the first regime, the primary phase is considered as the continuous phase and the secondary phase as the dispersed phase. For the second regime, the secondary phase is considered as the continuous phase and the primary phase as the dispersed phase.

The linearized drag coefficient is related to the standard engineering definition of the drag coefficient C_D for particles by:

$$A_D = C_D \frac{1}{2} \rho_c |\mathbf{v}_r| \left(\frac{a_{cd}}{4} \right) \tag{2249}$$

where a_{cd} is the interfacial area density. The factor $\frac{a_{cd}}{4}$ represents the projected area of the equivalent spherical particle. Corrections for the shape of non-spherical particles can be absorbed into the model for the drag coefficient C_D.

The standard drag coefficient C_D is composed of two factors:

$$D_D = f_D C_{\mathrm{D}\infty} \tag{2250}$$

where $C_{\mathrm{D}\infty}$ is the single-particle drag coefficient, which is determined from the measurement of terminal velocity of a particle in an unbounded stagnant continuous phase. f_D is a drag correction factor for the shape

of non-spherical particles, which is determined from the measurement of terminal velocity for multiple particles of the same size traveling together at a given concentration. This correction can cover a number of physical effects that can be difficult to separate in practice, such as hindering due to increased turbulence interaction, or co-operative motion due to swarming or coalescence.

Force Balance Under Terminal Velocity Conditions

Under terminal velocity conditions, the pressure buoyancy and drag forces are in equilibrium and the two-phase momentum conservation equations for the continuous and dispersed phases reduce to the result:

$$V^2 = \frac{8\alpha_c\alpha_d}{a}\frac{\Delta\rho}{\rho_c}\frac{g}{C_D} \tag{2251}$$

where:

$\Delta\rho = |\rho_c - \rho_d|$

a is the interfacial area density for the two-phase interaction.

There are two alternative standard models for interfacial area density in STAR-CCM+, which leads to:

$$V^2 = \begin{cases} \frac{4}{3}\frac{\Delta\rho}{\rho_c}\frac{gd}{C_D}\alpha_c & \text{for Spherical Particle Interfacial Area} \\ \frac{4}{3}\frac{\Delta\rho}{\rho_c}\frac{gd}{C_D} & \text{for Symmetric Interfacial Area} \end{cases} \tag{2252}$$

Eqn. (2252) is a key equation for the calibration of single-particle drag models and multi-particle correction factors from experimental measurements of terminal velocity V as a function of particle size d, and continuous phase volume fraction α_c.

The results of such a two-phase calibration are applied in the multiphase context, and in flows where particles are not traveling at terminal velocity. The terminal velocity is replaced with the computed relative velocity $|\mathbf{v}_r|$ when determining the drag coefficient C_D and drag correction factor ${f_{ij}}^D$ for each phase interaction ij.

Role of Interfacial Area Density Method in Force Balance

The Spherical Particle Interfacial Area method is geometrically exact for spherical particles of a given diameter. Any corrections for non-spherical shape can be absorbed into the drag coefficient. Regardless of particle shape, the term $\alpha_c\Delta\rho g d$ in the force balance correctly reflects the change in effective buoyancy force with concentration due to the reducing difference in density between the equivalent-volume spherical particle and the two-phase mixture as α_c falls.

The Symmetric Interfacial Area method is used where robustness is required across a wide range of volume fraction, so that the maximum packing of spherical particles can be exceeded. If an additional factor of $1/\alpha_c$ is absorbed into the drag correction factor that is used with the Symmetric Interfacial Area, both models give the same result, up to the maximum packing point. This adjustment is done automatically for the standard drag correction factor models that are implemented in STAR-CCM+. However, include the adjustment explicitly when using the Symmetric Interfacial Area Density for drag correction models from field functions or exponents.

Special Considerations for Bubble Drag Coefficients

The drag laws for bubbles which allow for reduced drag due to internal circulation includes Tomiyama/Pure, Tomiyama/Moderately Contaminated, Bozzano-Dente, and the Wang drag law. All of these are implemented in STAR-CCM+ as a function of (Re, Eo, Mo).

These laws have been formulated in the literature to reproduce the correct drag coefficient at terminal velocity. However, the laws, can result in unphysically low estimates of drag coefficient for small bubbles traveling at high slip velocity.

In STAR-CCM+, this problem is avoided by imposing a lower limit on the bubble drag coefficient. This limit is based on the lowest value of drag coefficient observed for a bubble of any size when it is traveling at its terminal velocity. This point corresponds to the slip velocity just before the bubble starts to distort from its spherical shape. It corresponds to a Weber number of about 2, so this is a physically meaningful constraint to apply at all bubble speeds.

Drag Correction Factors

The objective of drag correction methods is to reproduce experimental terminal velocities correctly as a function of particle concentration.

The traditional approach of Richardson and Zaki [483] is widely applicable for spherical particles of a given size. Drag correction for bubble columns is more complex because the bubble size can change as a function of concentration. In traditional approaches (for example, a model for the churn turbulent regime [442]), bubble drag correction has implicitly included the effects of turbulence interactions and of coalescence and breakup. A more recent trend is to isolate the effect of concentration alone on the terminal velocity of bubbles of a specified size (for example Simmonet et al. [492]).

To consider all these possibilities, define correction factors for particle size, terminal velocity, and drag coefficient. Let:

$$f_d = \frac{d}{d_\infty} \tag{2253}$$

$$f_V = \frac{V}{V_\infty} \tag{2254}$$

$$f_D = \frac{C_D}{C_{D\infty}} \tag{2255}$$

where the subscript ∞ indicates conditions for a single particle in an unbounded stagnant fluid.

Setting $\alpha_c = 1$ in the force balance Eqn. (2252), gives the force balance for a single particle traveling at terminal velocity in an unbounded medium:

$$V_\infty{}^2 = \frac{4}{3}\frac{\Delta\rho}{\rho_c}\frac{g d_\infty}{C_{D\infty}} \tag{2256}$$

This result applies for either of two interaction area density models. Then taking the ratio of Eqn. (2252) over Eqn. (2256) leads to the following relation between the three correction factors:

$$f_V{}^2 = \begin{cases} \dfrac{\alpha_c f_d}{f_D} \text{ for Spherical Particle Interfacial Area} \\ \dfrac{f_d}{f_D} \text{ for Symmetric Interfacial Area} \end{cases} \tag{2257}$$

The STAR-CCM+ model does not include the size correction factor f_d when computing f_V for two reasons:

- Modern computational methods focus on drag and drag correction for particles of a specified size, and particle size distribution is considered as a separate model.
- Size change is only relevant for drag correction of large bubbles whose drag coefficient is independent of Re.

Working with drag correction for particles of a specified size, the above relationship reduces to:

$$f_V{}^2 = \begin{cases} \dfrac{\alpha_c}{f_D} \text{ for Spherical Particle Interfacial Area} \\ \dfrac{1}{f_D} \text{ for Symmetric Interfacial Area} \end{cases} \tag{2258}$$

In STAR-CCM+ the relationship is used to:

- implement a drag correction factor f_D for models that are specified in literature as using a velocity correction factor f_V.
- obtain a velocity correction factor f_V for models that are specified in the literature as using drag correction factor f_D. The velocity correction factor is required to evaluate the drag coefficient for a single particle.

This last point is important in reproducing terminal velocity where the drag coefficient $C_{D\infty}$ depends on the particle Reynolds number. Evaluate the single-particle drag coefficient at the single particle terminal velocity V_∞ not at the terminal velocity V for which you are trying to estimate the drag coefficient in a multi-particle system. Apply the velocity correction factor f_V to the slip Reynolds number:

$$C_{D\infty} \equiv C_D(Re_\infty) = C_D\left(\frac{Re}{f_V}\right) \tag{2259}$$

Output Variables

To help you follow these drag computations, the following fields are available for inspection for each phase pair in a STAR-CCM+ simulation:

- The particle Eotvos number, Eo. This field only applies where the drag model depends on surface tension.
- The particle slip velocity Reynolds number, Re
- The single particle Reynolds number, Re_∞
- The single particle drag coefficient, $C_{D\infty}$
- The drag correction factor, f_D

Fixed Exponent Regimes

Where correction factors can be modeled as the continuous phase volume fraction raised to some constant power, that is:

$$f_V = \frac{V}{V_\infty} = \alpha_c^n \tag{2260}$$

$$f_D = \frac{C_D(Re, \alpha_c)}{C_D(Re_\infty, 1)} = \alpha_c^{n_D} \tag{2261}$$

Then Eqn. (2261) leads to a simple relation between these exponents:

$$n_D = \begin{cases} 1-2n \text{ for Spherical Particle Interfacial Area} \\ -2n \text{ for Symmetric Interfacial Area} \end{cases} \tag{2262}$$

For example, under Stokesian conditions ($Re_\infty < 0.1$), the constant Richardson Zaki [483] exponent of $n = 4.65$ for velocity correction leads to a drag correction exponent of:

$$n_D = -8.3 \tag{2263}$$

when using the Spherical Particle Interfacial Area, or:

$$n_D = -9.3 \tag{2264}$$

when using the Symmetric Interfacial Area.

Contents:

Fluid-Fluid Drag Coefficients
Fluid-Fluid Drag Corrections

Fluid-Fluid Drag Coefficients

STAR-CCM+ provides various drag coefficient methods.

Schiller-Naumann Drag Coefficient for Spherical Particles

STAR-CCM+ selects the Schiller-Naumann drag law internally according to the type of continuous fluid.

For **Newtonian fluids**, the drag coefficient of spherical, rigid particles is computed based on the correlation of Schiller and Naumann [491]:

$$C_D = \begin{cases} \frac{24}{\mathrm{Re}_d}\left(1 + 0.15\mathrm{Re}_d^{0.687}\right) & 0 < \mathrm{Re}_d \leq 1000 \\ 0.44 & \mathrm{Re}_d > 1000 \end{cases} \tag{2265}$$

The dispersed phase Reynolds number, Re_d is defined as:

$$\mathrm{Re}_d = \frac{\rho_c |\mathbf{v}_r| l}{\mu_c} \tag{2266}$$

where:

- ρ_c is the density of the continuous phase
- μ_c is the dynamic viscosity of the continuous phase
- l is interaction length scale or bubble size.

Schiller-Naumann with Non-Newtonian Herschel–Bulkley Continuous Phase

If the continuous phase is **non-Newtonian**, the Schiller-Naumann drag is modified. From, [401], the drag coefficient of spherical, rigid particles in a Herschel–Bulkley (or Power Law) continuous phase is:

$$C_D = \frac{24}{\mathrm{Re}_{HB}}\left(1 + 0.15\mathrm{Re}_{HB}^{0.687}\right) \tag{2267}$$

The dispersed phase Reynolds number, Re_{HB} is defined as:

$$\mathrm{Re}_{HB} = \frac{\mathrm{Re}_{PL}}{1 + \frac{7\pi}{24}\mathrm{Bi}_{HB}} \tag{2268}$$

The particle Bingham number, Bi_{HB}, is defined as:

$$\mathrm{Bi}_{HB} = \frac{\tau_0}{K}\left(\frac{d}{|\mathbf{v}_r|}\right)^n \tag{2269}$$

The Power-Law Reynolds number is:

$$\mathrm{Re}_{PL} = \frac{\rho_c|\mathbf{v}_r|^{2-n}d^n}{K} \tag{2270}$$

where:

- τ_0 is the yield stress
- K is the consistency factor
- n is the power-law component of the model
- $|\mathbf{v}_r|$ is the relative slip velocity between phases
- d is the particle diameter.

Schiller-Naumann with Non-Newtonian Generalized Carreau-Yasuda Continuous Phase

If the continuous phase is **non-Newtonian**, the drag coefficient of spherical, rigid particles in a Carreau-Yasuda continuous phase is:

$$C_D = \frac{24}{\mathrm{Re}_{HB}}\left(1 + 0.15\mathrm{Re}_0^{0.687}\right)\left(1 + 0.65(n-1)\lambda^{0.20}\right) \tag{2271}$$

where n and λ are parameters in the Carreau model [412].

Re_0 is the zero-shear particle Reynolds number that uses the zero-shear viscosity of the fluid:

$$\mathrm{Re}_0 = \frac{\rho_c|\mathbf{v}_r|d}{\mu_0} \tag{2272}$$

where:

- μ_0 is the zero shear viscosity of the continuous phase
- d is the interaction length scale or bubble size

- ρ_c is the density of the continuous phase
- $|\mathbf{v}_r|$ is the magnitude of the relative velocity between the phases.

Rusche-Issa Drag Coefficient

The Rusche-Issa drag coefficient applies when the dispersed phase volume fraction becomes high. In order to account for the presence of a larger number of particles that increase drag, Rusche and Issa created a drag correction term that ranges from one at zero dispersed phase volume fraction and increases exponentially [485].

$$f(\alpha_d) = e^{K_1 \alpha_d} + \alpha_d^{K_2} \tag{2273}$$

The Rusche-Issa drag law uses a linearized Schiller-Naumann form combined with the Rusche-Issa correction.

Rusche [484] studied a series of experiments from academic literature for dispersed phases that were gas, liquid, and solid particles, and found correlations to fit the drag coefficient [486]. The averages of the experiments give the parameters that are summarized below:

Dispersed Phase	Gas Bubbles	Liquid Droplets	Solid Particles
K1	3.64	2.10	2.68
K2	0.864	0.249	0.430

Symmetric Drag Coefficient

The symmetric drag coefficient provides the linearized coefficient as:

$$A_{cd}^D = \frac{3}{4} C_D \frac{\alpha_c \alpha_d (\alpha_c \rho_c + \alpha_d \rho_d)}{(\alpha_c l_c + \alpha_d l_d)} |\mathbf{v}_r| \tag{2274}$$

where:

l_c is the interaction length scale

l_d is the inverted topology length scale

$$C_D = \begin{cases} \frac{24}{Re_{cd}} \left(1 + 0.15 Re_{cd}^{0.687}\right) & 0 < Re_{cd} \leq 1000 \\ 0.44 & Re_{cd} > 1000 \end{cases} \tag{2275}$$

with,

$$Re_{cd} = \frac{(\alpha_c \rho_c + \alpha_d \rho_d)(\alpha_c l_c + \alpha_d l_d)|\mathbf{v}_r|}{(\alpha_c \mu_c + \alpha_d \mu_d)(\alpha_c + \alpha_d)} \tag{2276}$$

Bozzano-Dente Drag Coefficient for Bubbles

This correlation covers a wide range of bubbly flow regimes and accounts for different bubble shapes with a single expression. It has been developed for the case of a single bubble rising in a liquid column under the action of gravity. It takes both Eotvos number and Morton number into account, so can be applied to highly viscous systems such as glycerol, molten glass and some oils.

It can also be used for low pressure and high pressure water systems.

Note: For small bubbles, this correlation gives a different asymptote (Re/48) for the drag coefficient when compared to the values normally associated with contaminated or pure water systems (Re/24 or Re/16).

For this model, from Bozzano and Dente [408], the drag coefficient is:

$$C_D = f\left(\frac{a}{R_0}\right)^2 \tag{2277}$$

In which f is a friction factor and $(a/R_0)^2$ is a deformation factor. These quantities are derived as

$$f = \left(\frac{48}{\mathrm{Re}}\right)\frac{1+12\mathrm{Mo}^{1/3}}{1+36\mathrm{Mo}^{1/3}} + 0.9\frac{\mathrm{Eo}^{3/2}}{1.4\left(1+30\mathrm{Mo}^{1/6}\right)+\mathrm{Eo}^{3/2}} \tag{2278}$$

and

$$\left(\frac{a}{R_0}\right)^2 \cong \frac{10\left(1+1.3\mathrm{Mo}^{1/6}\right)+3.1\mathrm{Eo}}{10\left(1+1.3\mathrm{Mo}^{1/6}\right)+\mathrm{Eo}} \tag{2279}$$

Where Re is the Reynolds number, Mo is the Morton number, and Eo is the Eotvos number.

$$\mathrm{Re} = \frac{\rho_c|\mathbf{v}_r|l}{\mu_c} \tag{2280}$$

$$\mathrm{Eo} = \frac{|\rho_c-\rho_d|gl^2}{\sigma} \tag{2281}$$

$$\mathrm{Mo} = \frac{|\rho_c-\rho_d|g\mu_c^4}{\rho_c\rho_d\sigma^3} \tag{2282}$$

- ρ_c and μ_c are the density and dynamic viscosity of the continuous phase
- ρ_d is the density of the dispersed phase
- $|\mathbf{v}_r|$ is the magnitude of the relative velocity between the phases
- l is interaction length scale or bubble size
- g is gravitational acceleration
- σ is the surface tension.

Hamard and Rybczynski Drag Coefficient for Droplets

The Hamard and Rybczynski drag model is used for viscous Newtonian fluid droplets dispersed in a second immiscible viscous Newtonian fluid. From [412], the drag coefficient is:

$$C_D = \frac{24}{\mathrm{Re}}Y \tag{2283}$$

where:

Re is the particle Reynolds number

where:

$$Y = \frac{2+3X_E}{3+3X_E} \tag{2284}$$

$$X_E = \frac{\mu_d}{\mu_c} \tag{2285}$$

where μ_c is the dynamic viscosity of the continuous phase and μ_d is the dynamic viscosity of the dispersed phase.

For gas bubbles, $X_E \ll 1$ and $Y \rightarrow 2/3$. For solid particles $Y \rightarrow 1$.

The Hamard and Rybczynski drag model is not valid for high Reynolds numbers, where the Schiller-Naumann model is expected to be more accurate.

Tomiyama Drag Coefficient for Bubbles

The Tomiyama correlation is suited to systems for a wide range of conditions, as shown by the range of Eotvos (Eo), Morton (Mo), and Reynolds (Re) numbers for which it is valid:

$$\begin{gathered} 10^{-2} < Eo < 10^{3} \\ 10^{-14} < Mo < 10^{7} \\ 10^{-3} < Re < 10^{5} \end{gathered} \tag{2286}$$

The drag coefficient that is evaluated for a single bubble, which is based upon three contamination states, is from [503]:

- Pure:

$$C_D = \max\left[\min\left(\frac{16}{\mathrm{Re}}\left(1 + 0.15\mathrm{Re}^{0.687}\right), \frac{48}{\mathrm{Re}}\right), \frac{8\mathrm{Eo}}{3(\mathrm{Eo}+4)}\right] \tag{2287}$$

- Moderate:

$$C_D = \max\left[\min\left(\frac{24}{\mathrm{Re}}\left(1 + 0.15\mathrm{Re}^{0.687}\right), \frac{72}{\mathrm{Re}}\right), \frac{8\mathrm{Eo}}{3(\mathrm{Eo}+4)}\right] \tag{2288}$$

- Contaminated:

$$C_D = \max\left[\left(\frac{24}{\mathrm{Re}}\left(1 + 0.15\mathrm{Re}^{0.687}\right)\right), \frac{8\mathrm{Eo}}{3(\mathrm{Eo}+4)}\right] \tag{2289}$$

where Re is the Reynolds number and Eo is the Eotvos number.

$$\mathrm{Re} = \frac{\rho_c |\mathbf{v}_r| l}{\mu_c} \tag{2290}$$

$$\mathrm{Eo} = \frac{|\rho_c - \rho_d| g l^2}{\sigma} \tag{2291}$$

- ρ_c is the density of the continuous phase
- μ_c is the dynamic viscosity of the continuous phase
- ρ_d is the density of the dispersed phase
- $|\mathbf{v}_r|$ is the magnitude of the relative velocity between the phases
- l is interaction length scale or bubble size
- g is gravitational acceleration

- σ is the surface tension.

The Tomiyama drag coefficient was used by Tomiyama et al. [504] while evaluating their lift coefficients. Therefore, use the Tomiyama lift coefficient with the Tomiyama drag coefficient.

Wang Drag Coefficient for Bubbles

This model is for concentrations of air bubbles in water at near atmospheric pressure. Use with care for other conditions.

Bubbles deform and depart from the spherical shape. Use the empirical correlation of Wang et al. [512], to obtain more realistic values. The correlation was derived using curve-fitting measurements for a single bubble rising in water:

$$C_D = e^{\left[a + b\ln\mathrm{Re}_d + c\left(\ln\mathrm{Re}_d\right)^2\right]} \tag{2292}$$

Where Re_d is the dispersed phase Reynolds number.

The original coefficients are given in the table below.

$\mathbf{Re}_d$	**a**	**b**	**c**
$\mathbf{Re}_d \leq 1$	$\ln 24$	-1	0
$1 < \mathbf{Re}_d \leq 450$	2.699467	-0.33581596	-0.07135617
$450 < \mathbf{Re}_d \leq 4000$	-51.77171	13.1670725	-0.8235592
$\mathbf{Re}_d > 4000$	$\ln(8/3)$	0	0

The STAR-CCM+ implementation has been modified from Wang's original method [512] so that the drag coefficient is continuous as particle Reynolds number changes. This is important to minimize the residual achievable in applications with variable particle diameter, and the modified method gives reliable convergence with normal values of relaxation factors.

The modifications are:

- The breakpoint between the first two Re regimes in the above table has been moved from Re=1 to Re=2.197115.
- The last two Re regimes in the above table have been replaced using:

$$C_D = \frac{8}{3}\frac{\mathrm{Eo}_{AW}}{\mathrm{Eo}_{AW} + 4} \tag{2293}$$

For close equivalence with the original Wang curve fit, the Eotvos number is here computed from bubble diameter d using water and air properties that are fixed at a temperature of 293K.

$$\mathrm{Eo}_{AW} = \frac{\left(\rho_W - \rho_A\right)gd^2}{\sigma_{AW}} \tag{2294}$$

- ρ_W is 998.20 kg/m^3.
- ρ_A is 1.2069 kg/m^3.
- g is 9.81 m/s^2.

- σ_{AW} is 0.072736 N/m.

Štrubelj and Tiselj Interface Drag Coefficient

For the interface regime, the method of Štrubelj and Tiselj [496] is adapted to calculate the interface linearized drag coefficient:

$$A_{D,ir} = f_{ir}\alpha_p\alpha_s\rho_m \;\; \rho_m = \alpha_p\rho_p + \alpha_s\rho_s \;\; f_{ir} = 1/t_{sc} \tag{2295}$$

- The default value of the relaxation time scale t_{sc} is taken as 0.01 seconds.
- Reduction of relaxation time scale helps in instantaneous equalizing of velocities of both of the phases as it leads to increased drag in the interface regime.
- By increasing the drag in the interface regime, the slip in that region can be reduced.

Blended Interface Drag Coefficient

The blended method calculates the interface regime drag by blending the drags of the dispersed regime on either side of the interface regime. This method is useful when a clear separated interface is not defined in the interface regime.

Within the interface regime, the Linearized drag (A_d) is calculated as:

$$A_{d,ir} = \alpha_p A_{d,fr} + \alpha_s A_{d,sr} \tag{2296}$$

where the subscripts fr, ir and sr are for the first regime, the interface regime, and the second regime respectively.

Fluid-Fluid Drag Corrections

Drag correction describes how concentration modifies the single-particle drag coefficient model in a multi-particle system.

Richardson Zaki Drag Correction

This is a drag correction method for spherical particles with a wide range of applicability. According to this model, the effect of increased concentration on spherical particles is that of hindering. The terminal velocity decreases faster than would be expected purely from the reduction of buoyancy force of the two-phase mixture on the particles.

This model can also cover droplets that are small enough to retain their spherical shape and whose internal viscosity is high enough for internal circulation to be negligible. This model also applies to bubbles whose surfaces are immobile due to surface contamination, as in the case of air bubbles in regular tap water.

As originally specified, the Richardson Zaki model [483] for terminal velocity correction for spherical particles is:

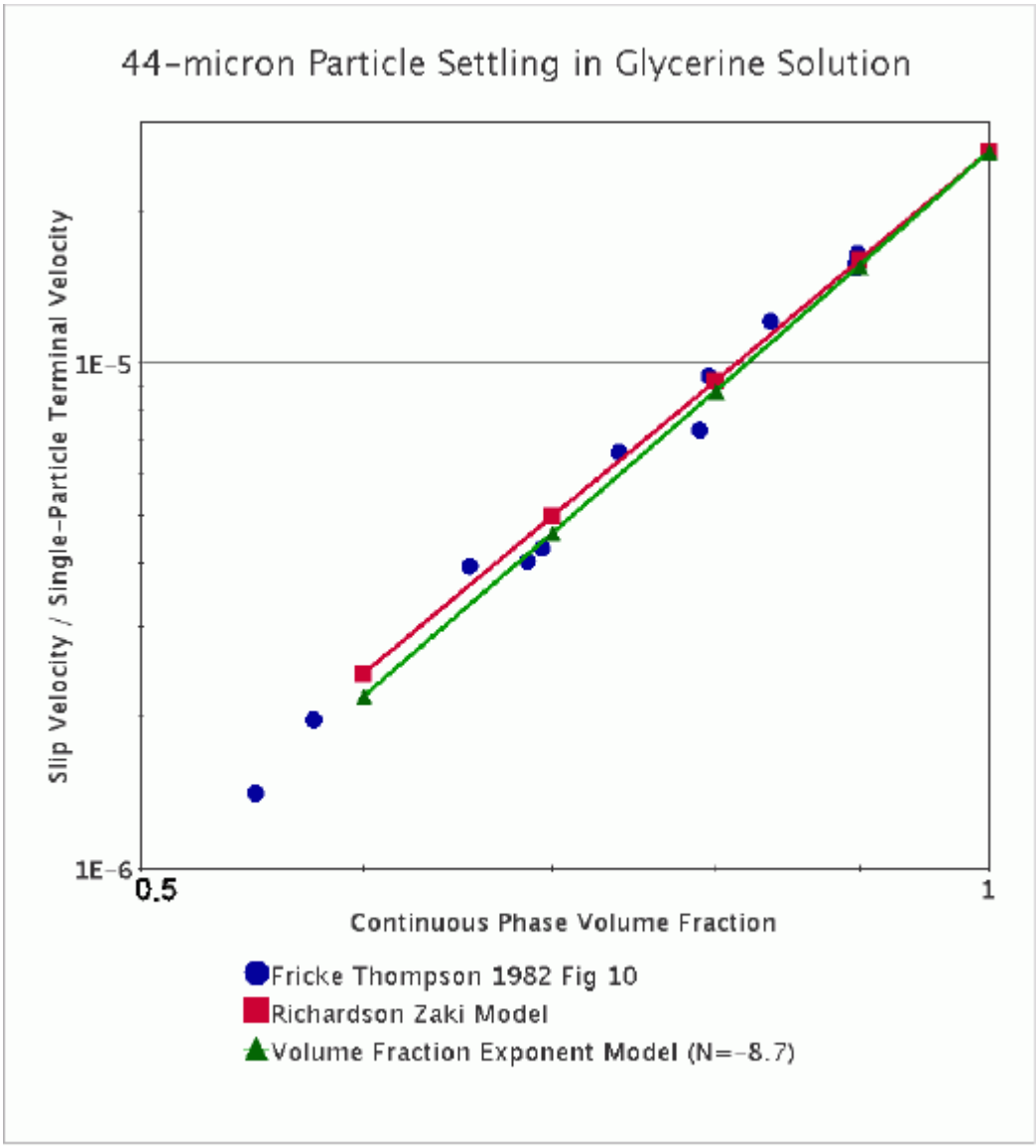
$$\frac{V}{V_\infty} = \alpha_c^n \tag{2297}$$

Range	n for small tubes	n for large tubes
$\mathrm{Re}_\infty < 0.2$	4.65	4.65
$0.2 < \mathrm{Re}_\infty < 1$	$\left(4.35 + 17.5\frac{d}{D}\right)Re_\infty^{-0.03}$	$4.35Re_\infty^{-0.03}$

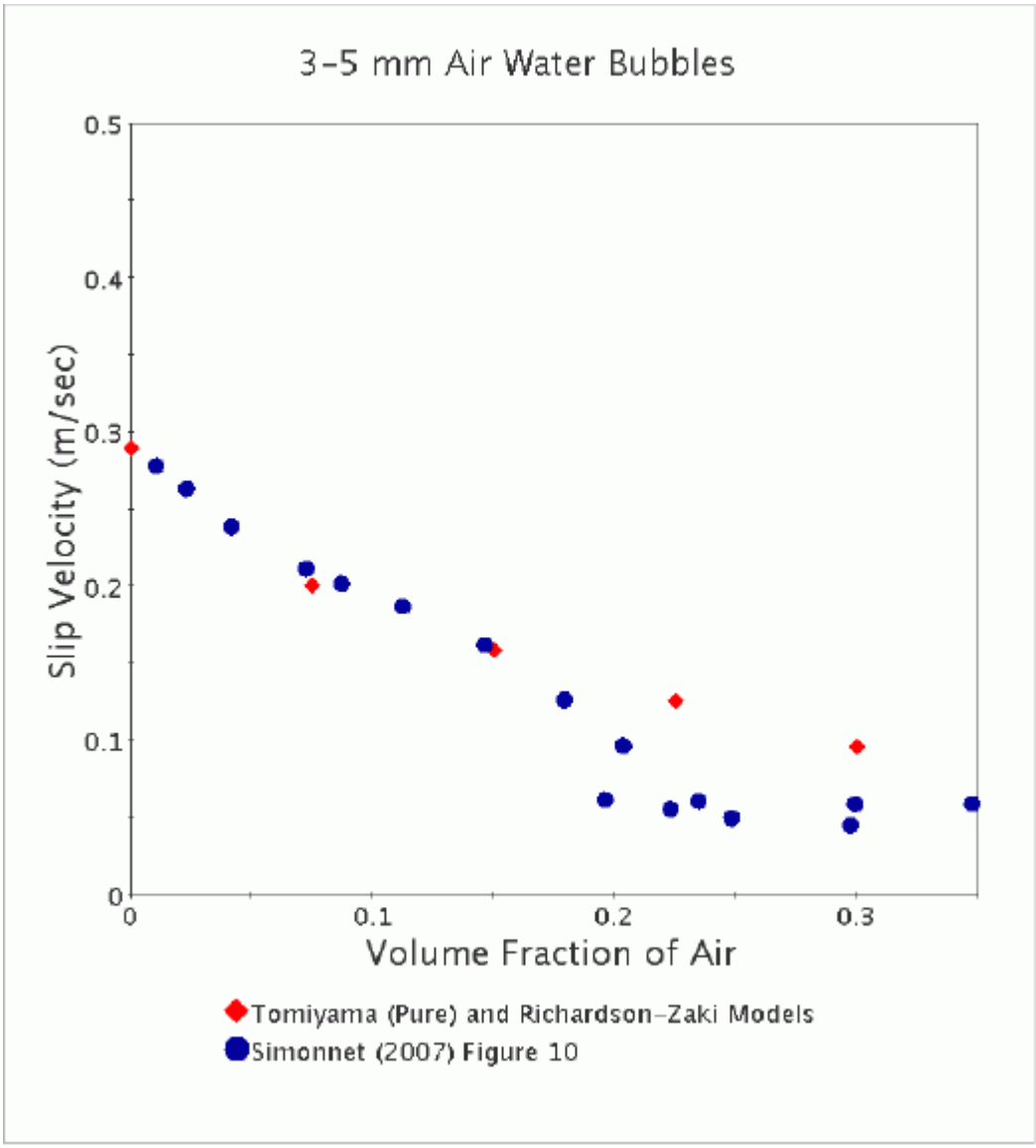
Range	n for small tubes	n for large tubes
$1 < \mathrm{Re}_\infty < 200$	$\left(4.45 + 18\frac{d}{D}\right)Re_\infty^{-0.1}$	$4.45Re_\infty^{-0.1}$
$200 < \mathrm{Re}_\infty < 500$	$4.45Re_\infty^{-0.1}$	$4.45Re_\infty^{-0.1}$
$\mathrm{Re}_\infty > 500$	2.39	2.39

n is the Richardson Zaki exponent, D is tube diameter, and d is particle diameter.

The two figures below illustrate the wide range of applicability of the Richardson Zaki method. The first shows the terminal velocity correction factor that is computed in a Stokesian particle settling test ($\mathrm{Re}_\infty < 0.2$) against the data of Fricke and Thompson [432]. A suspension of 44-micron resin particles in a glycerine-water mixture settles out at a rate that particle concentration affects. The rate is well-predicted either by the Richardson Zaki method, or by the Volume Fraction Exponent method using exponent n_D =-8.7.



The second figure shows the terminal velocity of 3-5 mm air bubbles rising in water against the data of Simonnet [492] under conditions $\mathrm{Re}_{\infty} > 500$. Even though the bubbles are ellipsoidal in this size range, agreement is good.



The original Richardson Zaki model is adapted for computation by:

- Omitting the small tube term as this would generate an unacceptable discontinuity between Reynolds number ranges
- Using min() and max() functions to make sure exact continuity between Reynolds number ranges
- Allowing you to update or adapt the original calibration constants.
- Preventing the drag correction reaching implausibly high or low levels as the continuous phase volume fraction vanishes. Set a maximum packing $\alpha_{d,max}$ for the sum of the dispersed phase volume fractions, so that the strength of the drag correction does not change further beyond this concentration.

$$f_V = max\left(\alpha_c, 1-\alpha_{d,max}\right) \tag{2298}$$

$$n = max\left\{\begin{matrix} A \\ min\left\{\begin{matrix} \\ min\left(BRe_\infty^{n_1}, CRe_\infty^{n_2}\right) \\ \\ D \end{matrix}\right. \end{matrix}\right. \tag{2299}$$

$$Re_\infty = \frac{Re}{f_V} \tag{2300}$$

$$f_D = \begin{cases} \frac{\alpha_c}{f_V^2} \text{ Spherical Particle Interfacial area} \\ \frac{1}{f_V^2} \text{ Symmetric Interfacial area} \end{cases} \tag{2301}$$

The single-particle Reynolds number Re_∞ is applied both to the single-particle drag law as well as to the Richardson Zaki correction itself. This is essential to reproduce terminal velocity correctly in regimes where drag coefficient is a strong function of Reynolds number.

This implementation gives identical results for drag coefficient and terminal velocity regardless of which of the two interfacial area models is used, at least up to the maximum packing point.

The default calibration is applicable for spherical particles, but you can modify it:

A	B	C	D	n_1	n_2	$\alpha_{d,max}$
4.65	4.35	4.45	2.39	-0.03	-0.1	0.66

Lockett Kirkpatrick Drag Correction

Lockett and Kirkpatrick [463] investigated the effect of bubble concentration on the terminal velocity of 5-mm air bubbles in water. They achieved stable bubbly flow of up to 66% void fraction by using a down flow. They applied the Richardson Zaki model to spherical particles to terminal velocity data for 5-mm air bubbles in water. The model assumed that both the drag coefficient and drag correction were in the Reynolds number independent range. For this range, where Re > 500, the Richardson Zaki exponent, n, is 2.39. They found that a better fit to the data could be achieved by applying a further empirical factor to take bubble deformation into account.

In a figure further below, the Lockett Kirkpatrick method is compared with the Simonnet et al. [492] method for predicting Simonnet's terminal velocity data for 7-10 mm bubbles. The agreement is good up to 15% void fraction, after which Simonnet found a swarming behavior not observed in the Lockett Kirkpatrick data from 5-mm bubbles.

The original Lockett Kirkpatrick model is:

$$\frac{V}{V_\infty} = \alpha_c^{2.39}\left[1 + 2.55\left(1-\alpha_c\right)^3\right] \tag{2302}$$

This Lockett Kirkpatrick model is adapted for computation by limiting the correction for extreme values of α_c through a maximum packing criterion $\alpha_{d,max}$. $\alpha_{d,max}$ has a default value of 0.66.

$$\widetilde{\alpha_c} = max\left(\alpha_c, 1-\alpha_{d,max}\right) \tag{2303}$$

$$f_V = (\tilde{\alpha}_c)^{2.39}\left[1 + 2.55\left(1-\tilde{\alpha}_c\right)^3\right] \tag{2304}$$

$$f_D = \begin{cases} \dfrac{\alpha}{f_V^2} \text{ Spherical Particle Interfacial area} \\ \dfrac{1}{f_V^2} \text{ Symmetric Interfacial area} \end{cases} \tag{2305}$$

This method gives identical results for drag coefficient and terminal velocity regardless of the interfacial area model that is used, at least up to the maximum packing point.

Simonnet Drag Correction

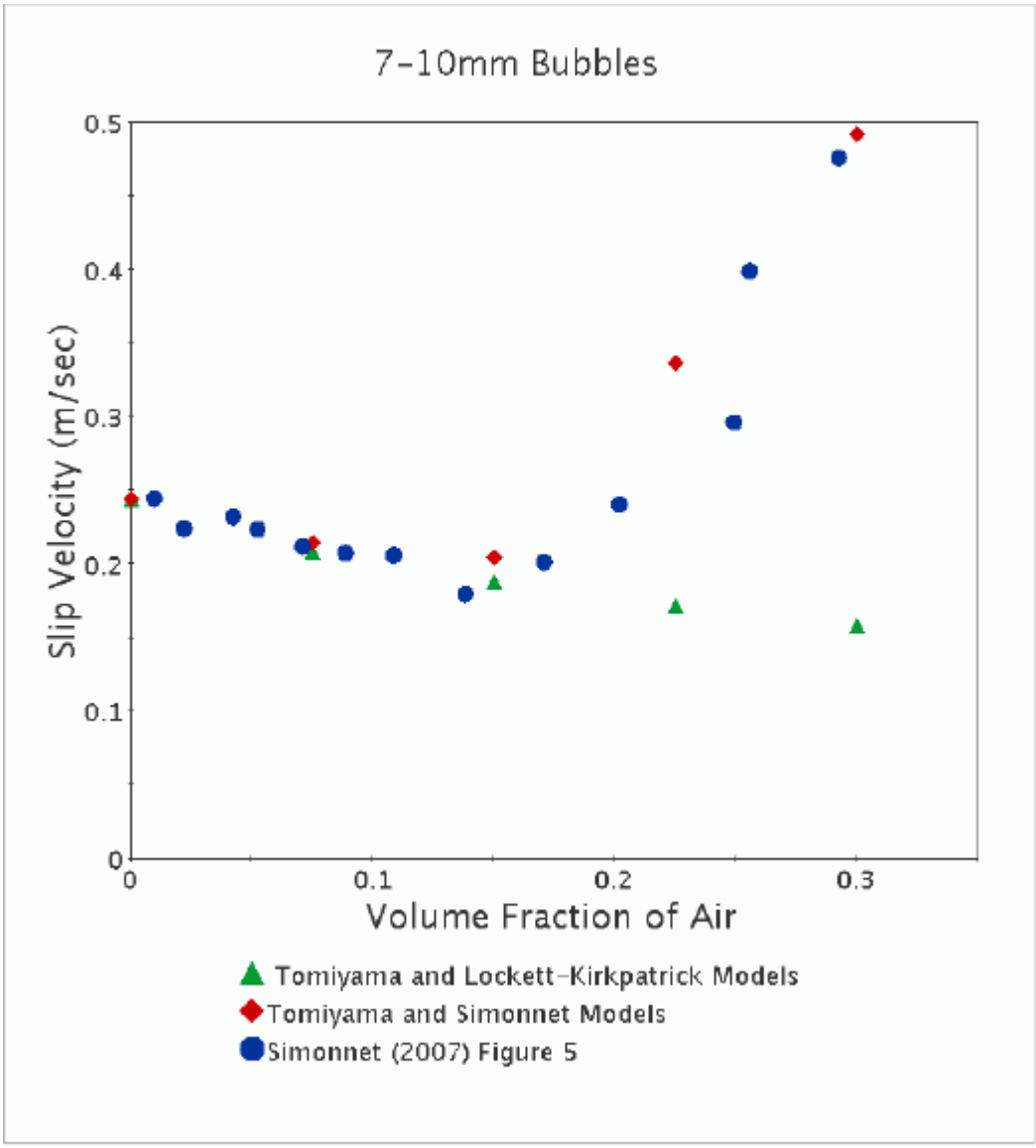
This is a drag correction for air / water bubbles in the 7-10 mm size range (Eotvos number 6.6-13.4) and 0-30% void fraction range. Effects are hindering in the range 0-15% and swarming in the range 15-30%.

Simonnet and others [492] measured terminal velocities, taking great care to work with uniform size distributions. Their recommended model for drag correction factor in the 7-10 mm range is:

$$\frac{C_D}{C_{D\infty}} = \alpha_c\left[\alpha_c^{25} + \left(4.8\frac{1-\alpha_c}{\alpha_c}\right)^{25}\right]^{-2/25} \tag{2306}$$

However, they also found that for smaller bubbles (2-6 mm) only hindering behavior occurs.

Terminal velocity computed using Simonnet's hindering/swarming method and Lockett and Kirkpatrick's [463] hindering-only methods are compared below against Simonnet's own data for 7-10 mm bubbles in air-water [492]. Both methods are a good fit to the data up to 15% void fraction when swarming behavior begins.



The Simonnet model is adapted for computation by making the correction go to zero smoothly and linearly as the continuous phase volume fraction vanishes, so that it can be used safely beyond the void fraction range of Simonnet's data.

$$f_D = s f_{D1} + (1 - s) f_{D2} \tag{2307}$$

$$f_{D1} = \alpha_c \left[\alpha_c^{25} + \left(4.8 \frac{1 - \alpha_c}{\alpha_c} \right)^{25} \right]^{-2/25} \tag{2308}$$

$$f_{D2} = 0.25 \alpha_c \tag{2309}$$

$$s = 0.5 \left\{ 1 + \tanh \left(\frac{\alpha_c - 0.7058}{0.05} \right) \right\} \tag{2310}$$

Spherical Particle Interfacial Area Density and Symmetric Interfacial Area Density models give the same results for drag coefficient and terminal velocity when the Simonnet drag correction model is used in its range of applicability of 0-30% void fraction.

Volume Fraction Exponent Drag Correction

This drag correction model assumes that the drag correction factor is of the form:

$$f_D = \alpha_c^{n_D} \tag{2311}$$

where n_D is some constant power.

This method complements the other drag correction methods by covering several regimes of practical interest, such as

- Small spherical particles settling under Stokesian drag condition.
- Large spherical particles under turbulent wake drag conditions.
- Spherical cap bubbles swarming under a churn-turbulent flow regime.

Regime	Reference	n	n_D (Spherical particle Interfacial Area)	n_D (Symmetric Interfacial Area)
small spherical particles $\mathrm{Re}_\infty < 0.2$	[483]	+4.65	-8.3	-9.3
large spherical particles $\mathrm{Re}_\infty > 500$	[483]	+2.39	-3.78	-4.78
spherical cap bubbles, churn-turbulent conditions	[442]	-0.5	+2	+1

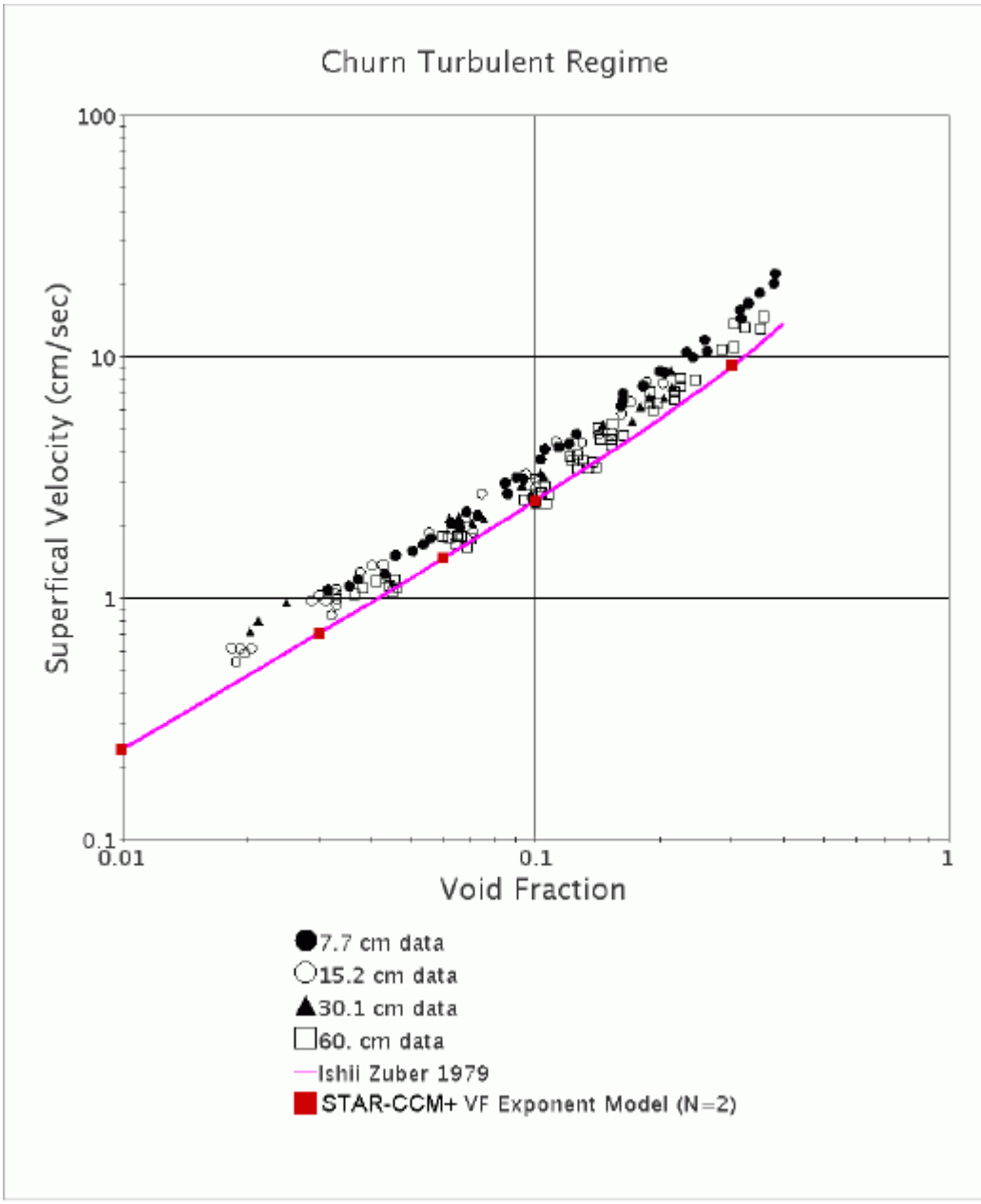
The first of these three regimes has already been shown in the figure in the Richardson Zaki section on the dilute particle settling under Stokesian conditions. The settling velocity is a strong function of concentration. Using a drag correction exponent of $n_D = -8.7$ is equivalent to using the velocity correction exponent recommended by Fricke and Thompson, $n = 4.85$, which provides a good fit to their own data [432].

The drag correction method in the churn-turbulent regime is taken from Ishii and Zuber [442] and is relevant for large spherical cap bubbles in bubble columns. In such columns, the bubble size is controlled by coalescence and breakup, and is unrelated to the initial size of the bubbles that are introduced at the inlet.

Ishii and Zuber suggested that bubbles in this regime are characterized by a Weber number of 8 and a drag coefficient of 8/3. Both of these numbers are defined in terms of their drift flux model velocity. This leads to a model predicting pure swarming behavior. Bubble equivalent diameter, terminal velocity, and drag coefficient vary as continuous phase volume fraction raised to the power of -1/2, -3/4 and +2 respectively.

Ishii and Zuber found that this model under-predicts Akita and Yoshida data for superficial velocity for air bubbles in dilute sodium sulfite solution that is presented in Figure 16 of Ishii and Zuber [442]. However it does show the correct trend of superficial velocity against bubble concentration and is a reasonable approximation for the data from the bubble column with the largest diameter of 60 cm.

In the verification calculation reproduced below, the equivalent bubble diameter varies from 11 mm to 13 mm as the void fraction increases from 1% to 30%. The terminal velocity increases from 23 cm/sec to 30 cm/sec, as expected from the relevant exponents.



The constant exponent method is adapted for computation by limiting the correction for extreme values of α_c. This is done through a maximum packing criterion $\alpha_{d,max}$, which has a default value of 0.66:

$$\tilde{\alpha}_c = max\left(\alpha_c, 1-\alpha_{d,max}\right) \tag{2312}$$

$$f_D = \left(\tilde{\alpha}_c\right)^{n_D} \tag{2313}$$

$$f_V = \begin{cases} \sqrt{\frac{\alpha}{f_D}} \text{ Spherical Particle Interfacial Area} \\ \sqrt{\frac{1}{f_D}} \text{ Symmetric Interfacial Area} \end{cases} \tag{2314}$$

This velocity correction factor f_V is used to compute the single-particle terminal velocity correctly where the drag coefficient depends on Reynolds number. However, this Reynolds number adjustment is not relevant in cases where the single particle drag coefficient is not a function of Reynolds number. For example, in the Ishii Zuber model for large bubbles in the churn turbulent regime.

Virtual Mass

The inertia of surrounding fluid influences the acceleration of a particle that is submerged in the flow. Inviscid flow theory allows this effect to be represented as a “virtual mass” or “added mass” equal to a constant multiplied by the mass of fluid that the particle displaces.

Including this “virtual mass force” can make accelerating flows more realistic [453]. For example, the virtual mass term can influence the trajectory of bubbles in a steady but swirling flow. Because it narrows the range of response timescales across different phases, the virtual mass term can also help non-accelerating flows to converge, by making them less sensitive to momentum or pressure relaxation factors.

A two-phase formulation for the virtual mass force is from Auton et al. [402]. This formulation can also be used in the multiphase context where there is one continuous phase and several dilute, dispersed phases. For non-dilute systems, it can still give a useful first approximation, especially if the virtual mass coefficient includes a correction for the concentration of particles.

The virtual mass force acting on phase i due to acceleration relative to phase j is:

$$\mathbf{F}_{ij}^{VM} = C_{VM}\rho_c\alpha_d\left(\mathbf{a}_j - \mathbf{a}_i\right) \tag{2315}$$

where:

c is the continuous phase in the phase interaction ij.

d is the dispersed phase in the phase interaction ij.

C_{VM} is the virtual mass coefficient for interaction ij.

$\mathbf{a}_i, \mathbf{a}_j$ are the acceleration of phase i and j, respectively.

The acceleration term for virtual mass is based on the rate of change of velocity of constant mass particles, which in the stationary coordinate frame becomes the material derivative:

$$\mathbf{a}_i = \left(\frac{D\,\mathbf{v}}{Dt}\right)_i \tag{2316}$$

However, for applications where the particle mass can increase or decrease due to interphase mass transfer, $\mathbf{a}_i$ must be generalized. This is accomplished by starting with the momentum balance for the phase that is contained within a constant control volume. The momentum fluxes across phase-to-phase interfaces can be identified as similar terms to the momentum fluxes across intersections between the phase and the outer surfaces of the control volume. This leads to the following definition of the total inertial term for phase i in reaction to all other applied forces:

$$\rho_i \alpha_i \mathbf{a}_i = \frac{\partial}{\partial t} \rho_i \alpha_i \mathbf{v}_i + \nabla \cdot \left(\rho_i \alpha_i \mathbf{v}_i \mathbf{v}_i \right) - \sum_i m_{ij} \mathbf{v}_{ji} \tag{2317}$$

where:

m_{ij} is the interphase mass transfer rate per volume from phase i to phase j.

$\mathbf{v}_{ij}$ is the interphase value of velocity.

The following closure satisfies conservation of momentum in the sense that the momentum flux leaving one phase balances the momentum flux entering the other phase:

$$\mathbf{v}_{ij} = \begin{cases} \mathbf{v}_j & m_{ij} > 0 \\ \mathbf{v}_i & m_{ij} < 0 \end{cases} \tag{2318}$$

The full acceleration term, including momentum fluxes due to interphase mass transfer, then becomes:

$$\mathbf{a}_i = \left(\frac{D\,\mathbf{v}}{Dt} \right)_i + \frac{1}{\rho_i \alpha_i} \sum_l m_{ij} \left(\mathbf{v}_i - \mathbf{v}_{ij} \right) \tag{2319}$$

Spherical Particle Virtual Mass Coefficient

From inviscid flow theory [453], the virtual mass coefficient for a spherical particle accelerating in an unbounded three-dimensional fluid is:

$$C_{VM,\,\mathrm{sphere}} = 0.5 \tag{2320}$$

This function is limited for stable solutions across a wide range of volume fraction by specifying a minimum free stream fraction γ_{min} with a default value of zero.

$$C_{VM} = \left[\left(C_{VM,\,\mathrm{sphere}} \right)^{-2} + C_{VM,\,\max} \left(\alpha_c, \gamma_{min} \right)^{-2} \right]^{-1/2} \tag{2321}$$

Zuber Virtual Mass Coefficient

Lamb [453] provided an inviscid flow solution for a case of an inner sphere accelerating across the center of an outer stationary sphere. Zuber [520] used this solution to estimate the virtual mass coefficient for an infinite array of particles. The cube of the sphere diameter ratio represents the volume fraction.

The result in terms of the volume fraction of a single dispersed phase is:

$$C_{VM,\,\mathrm{Zuber}} = 0.5 \frac{1 + 2\alpha_d}{1 - \alpha_d} \tag{2322}$$

This function is adapted for multiphase computation by replacing α_d by the sum of dispersed phase volume fractions:

$$\sum \alpha_d = 1 - \alpha_c \tag{2323}$$

and then applying a limiter for robustness as $\alpha_c \to 0$:

$$C_{VM,\,\mathrm{Zuber}} = 0.5 \frac{3 - 2\alpha_c}{\alpha_c} \tag{2324}$$

$$C_{VM} = \left[C_{VM}^{-2} + C_{VM,\,\max} \left(\alpha_c, \gamma_{min} \right)^{-2} \right]^{-(1/2)} \tag{2325}$$

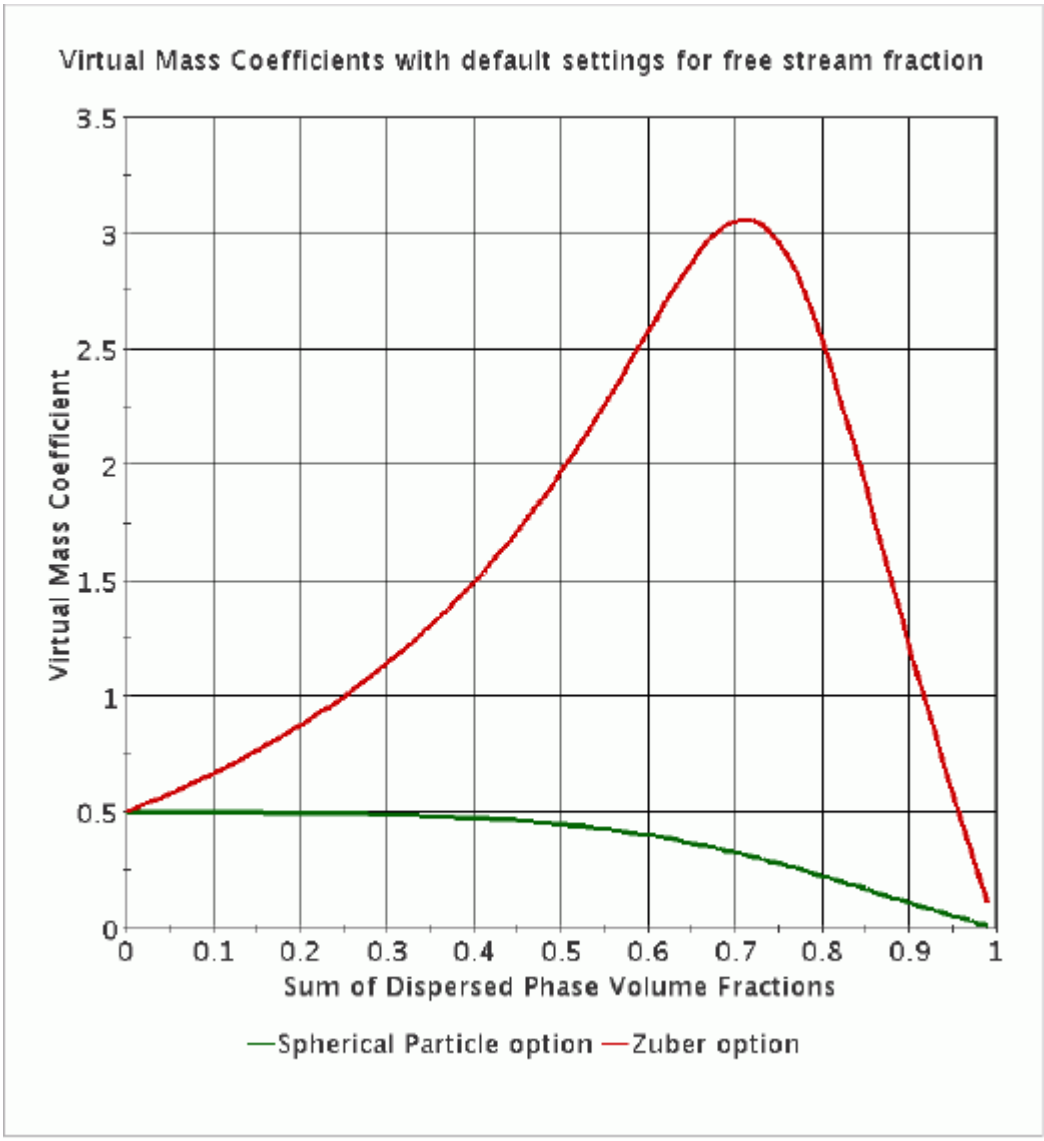
The default value of the notional “free stream” fraction is $\gamma_{min} = -10$. This allows the Zuber virtual mass coefficient to represent the effect of total particle concentration on the virtual mass force in the useful range $0 < \sum \alpha_d < 0.66$ before limiting is applied.

In a simulation, the selected virtual mass coefficient can be plotted as a field.

The default option is the spherical particle method that is based on a standard coefficient of 0.5 for spherical particles. For stability, the coefficient is reduced at high dispersed phase concentrations.

The Zuber option is suitable for improved accuracy in modeling the effect of increasing particle concentration, or multiple dispersed phases, in accelerating flows.

The virtual mass coefficients that are produced by these two methods are compared in the figure below, each using their own default value for free stream fraction γ_{min} as a robustness limit.



Virtual Mass Terms in Rotating Frame

The Virtual Mass Force term that is defined above is modified as follows when used in multiple reference frame (MRF) applications. This modification is based on an absolute velocity formulation, which solves directly for velocities in the inertial frame, I, using momentum equations that are derived in the rotating frame, R.

$$\mathbf{F}_{ij}{}^{VM} = C_{VM}\rho_c\alpha_d\left[\mathbf{a}_j{}^{IR} - \mathbf{a}_i{}^{IR}\right] + C_{VM}\rho_c\alpha_d\omega \times \left[\mathbf{v}_j{}^{I} - \mathbf{v}_i{}^{I}\right] \tag{2326}$$

where ω is the angular velocity of the rotating frame, and the phase-k acceleration in the rotating frame is defined in terms of the absolute velocity by:

$$\mathbf{a}_k{}^{IR} = \left[\frac{D}{Dt}\mathbf{v}_k{}^{I}\right]_k^{R} + \frac{1}{\rho_k\alpha_k}\sum_l m_{kl}\left(\mathbf{v}_k{}^{I} - \mathbf{v}_{kl}{}^{I}\right) \tag{2327}$$

$$\left[\frac{D}{Dt}\phi\right]_k^{R} = \frac{\partial\phi}{\partial t} + \mathbf{v}_k{}^{R}\cdot\nabla\phi \tag{2328}$$

Virtual Mass Stress

The terms above describe the virtual mass effects of mean flow acceleration such as occur in developing flows or due to swirl. However, virtual mass effects are also present in the local instantaneous response of particles to turbulent fluctuations. The resulting contribution to the Reynolds-averaged phase momentum equations can be described by a further virtual mass stress term.

The instantaneous virtual mass force (seen by the dispersed phase) reads:

$$\mathbf{F}_{vms} = \rho_c C_{VM}\alpha_d\left(\frac{\partial\mathbf{v}_c}{\partial t} + \mathbf{v}_c\cdot\nabla\mathbf{v}_c - \frac{\partial\mathbf{v}_d}{\partial t} - \mathbf{v}_d\cdot\nabla\mathbf{v}_d\right) \tag{2329}$$

where subscripts c and d stand for continuous and dispersed phases, respectively.

Reynolds averaging of Eqn. (2329) yields several terms. While acceleration of the mean flow is accounted by the Virtual Mass Force, the fluctuation covariances, given by:

$$\langle\ \alpha_d\mathbf{v}_d''\cdot\nabla\mathbf{v}_d''\ \rangle\ ,\ \langle\ \alpha_c\mathbf{v}_c''\cdot\nabla\mathbf{v}_c''\ \rangle \tag{2330}$$

yield, after some algebra:

$$\langle\ \mathbf{F}_{vms}\ \rangle = \rho_c C_{VM}\overline{\alpha}_d\left(\frac{1}{\overline{\alpha}_c}\nabla\cdot\overline{\alpha}_c\sigma_c - \frac{1}{\overline{\alpha}_d}\nabla\cdot\overline{\alpha}_d\sigma_d\right) \tag{2331}$$

where σ_c and σ_d are the Reynolds Stresses of the corresponding phases, and $\langle\ \mathbf{F}_{vms}\ \rangle$ is the Virtual Mass Stress term.

Virtual Mass Coefficient Limit

In the pursuit of treating the virtual mass force as an objective interphase force, Cook and Harlow [414] formulated an alternative virtual mass term by collapsing a three-field (dispersed, entrained continuous and bulk continuous phases) model into two-fields.

The Cook and Harlow formulation becomes singular when the virtual mass associated with the dispersed phase exceeds the total mass of the continuous phase. This is not necessarily a physical constraint and it does not apply to other virtual mass formulations, such as those based on Auton et al. [402]. Nevertheless the

Cook and Harlow concept of an entrained fluid fraction and a free stream fluid fraction can be used as a starting point when considering the maximum particle concentrations for which the virtual mass force is a relevant correction to the multiphase momentum equations. It is assumed that there is one continuous phase and that the same model for virtual mass coefficient C_{VM} is used for all dispersed phases. The fraction of "free stream" fluid, that is the fraction that is not entrained on the dispersed phases as "added mass" is:

$$\gamma = \frac{\alpha_c - C^{VM} \sum \alpha_d}{\alpha_c} \tag{2332}$$

Then an upper bound on the virtual mass coefficient is provided by a plausible assumption that the fraction of free fluid γ is not less than some minimum γ_{min}:

$$C_{VM,\max}(\alpha_c, \gamma_{min}) = \frac{\alpha_c}{1-\alpha_c}(1-\gamma_{min}) \tag{2333}$$

Choosing $\gamma_{min} = 0$ gives a useful operating range of continuous phase volume fraction $1 \leq \alpha_c \leq 0.5$ for which a typical virtual mass coefficient of $C_{VM} = 0.5$ can be applied before the constraint starts to limit the coefficient. Beyond this range, the constraint makes the virtual mass force go to zero as the continuous phase volume fraction goes to zero, which helps maintain stability.

The virtual mass coefficient is based on the kinetic energy content in the fluid field, and it is not always helpful to consider it only in terms of entrained mass. Strong virtual mass effects, exceeding the Cook and Harlow limit $\gamma_{min} = 0$, can indeed be realistic in close-packed particles [520] or in slug flows [440]. The physical interpretation of such high virtual mass coefficients is that a relative acceleration difference of the particle can cause a fluid backflow through confined gaps between particles, or between particles and walls, at local peak velocities many times higher than the mean relative slip velocity between phases. Under these conditions, relax the constraint by using negative values of γ_{min} to see how far accuracy can be improved without the solution becoming unstable.

Lift

When the continuous phase flow field is non-uniform or swirling, particles experience a lift force perpendicular to the relative velocity.

This force, F_{ij}^L, is derived by Auton et al. [402] as:

$$\mathbf{F}_{ij}^L = C_{L,\text{effective}} \alpha_d \rho_c \left[\mathbf{v}_r \times (\nabla \times \mathbf{v}_c)\right] \tag{2334}$$

where $C_{L,effective}$ is calculated from the Lift Coefficient (C_L) and

the Lift-Correction (f_l) as:

$f_l C_L$ for Spherical Particle Interaction Area Density model

$f_l \alpha_c C_L$ for Symmetric Interaction Area Density model

You can set the lift coefficient, C_L, but it is set to $C_L = 0.25$ by default, following Lance et al. [454]. Other models that take account of bubble size and distortion can be found in the literature. You can implement these models through field functions for the lift coefficient.

For a multiple flow regime phase interaction, the lift force is calculated only for the first regime and the second regime. The weight function of the interface regime is taken as zero.

The lift force is calculated as:

$$\mathbf{F}_{ij}^{L} = W_{fr} C_{L,fr} \alpha_s \rho_p \boldsymbol{u}_r \times \left(\nabla \times \boldsymbol{u}_p \right) + W_{sr} C_{L,sr} \alpha_p \rho_s \boldsymbol{u}_r \times \left(\nabla \times \boldsymbol{u}_s \right) \tag{2335}$$

where:

- $C_{L,fr}$ is the lift coefficient considering the secondary phase dispersed in the primary phase.
- $C_{L,sr}$ is the lift coefficient considering the primary phase dispersed in the secondary phase.

Tomiyama Lift Coefficient

The Tomiyama lift coefficient has been evaluated based on experimental trajectories of single bubbles in a high-viscosity system. However, it is worth noting that Tomiyama et al. [504] found that the Tomiyama lift coefficient proposed also yields similar values to experimental data for a small bubble in a low viscosity, air-water system.

The lift coefficient that is proposed by Tomiyama et al. [504] is:

$$C_L = \begin{cases} 0.288\tanh\left(0.121\max\left[\mathrm{Re}, 7.374\right]\right) & \mathrm{Eo}_d < 4 \\ 0.00105\mathrm{Eo}_d^3 - 0.0159\mathrm{Eo}_d^2 - 0.0204\mathrm{Eo}_d + 0.474 & 4 \le \mathrm{Eo}_d \le 10 \\ -0.27 & 10 < \mathrm{Eo}_d \end{cases} \tag{2336}$$

Where Re is the Reynolds number (see Eqn. (2290)) and Eo_d is the modified Eotvos number that is based on the maximum horizontal dimension of a bubble.

The empirical relations between Eo_d and Eo is:

$$\mathrm{Eo}_d = \mathrm{Eo} \times \mathrm{E}^{-2/3} \tag{2337}$$

Where E is the empirical correlation for the bubble aspect ratio of spheroidal bubbles in a fully contaminated system and is:

$$\mathrm{E} = \frac{1}{1 + 0.163\mathrm{Eo}^{0.757}} \tag{2338}$$

Note: The Tomiyama [503] drag coefficient was used by Tomiyama et al. [504] while evaluating their lift coefficients. Therefore, use the Tomiyama lift coefficient with the Tomiyama drag coefficient.

Lift Correction

A strong correlation between the lift correction and drag correction has been observed and, in light of limited literature available for lift correction, a perfect correlation of lift correction with drag correction is approximated.

The following options are available:

- Drag Correlated Lift Coefficient Correction method

This method uses the Drag Coefficient Correction as an approximation for the Lift Coefficient Correction.

This approximation is useful for cases when swarming is modeled.

- Field Function

Turbulent Dispersion

The effect of turbulence in redistribution of non-uniformities in phase concentration is modeled by an additional turbulent dispersion force in the phase momentum equations. This term arises naturally when Reynolds averaging is applied to the instantaneous drag force.

For a simple derivation and further references, see Contribution of Drag to Turbulent Dispersion. With suitable closures, the term has the following form:

$$\mathbf{F}_{ij}^{TD} = A_D \mathbf{v}_{TD}$$
$$\mathbf{v}_{TD} = \mathbf{D}_{TD} \cdot \left\{ \frac{\nabla \alpha_d}{\alpha_d} - \frac{\nabla \alpha_c}{\alpha_c} \right\} \tag{2339}$$

where:

- $\mathbf{F}_{ij}^{TD}$ is the force per volume applied to the continuous phase i momentum equation due to dispersed phase j.
- $\mathbf{v}_{TD}$ is a relative drift velocity due to the use of volume-fraction weighted definitions of phase velocity.

  Alternatively, $\mathbf{v}_{TD}$ can be written:

$$\mathbf{v}_{TD} = \mathbf{D}_{TD} \cdot \left\{ \nabla \ln(\alpha_d) - \nabla \ln(\alpha_c) \right\} \tag{2340}$$

The logarithmic form is easier to discretize on a collocated grid. Use it for simulating flows where separation or boiling/condensation processes can produce volume fractions differing by many orders of magnitude in adjacent cells.

The tensor diffusivity coefficient $\mathbf{D}_{TD}$ is usually approximated isotropically from the continuous phase turbulent diffusivity:

$$\mathbf{D}_{TD} = C_0 \frac{\nu_c^t}{\sigma_\alpha} \mathbf{I} \tag{2341}$$

where:

- ν_c^t is the continuous phase turbulent kinematic viscosity.
- σ_α is the turbulent Prandtl number for volume fraction. This is discussed in more detail in the following section.
- The constant C_0 is not part of the model derivation, but has a default value 1 and you can vary it to test the sensitivity of the solution to this term.
- $\mathbf{I}$ is the identity matrix.

When a Reynolds stress tensor $\mathbf{R}_c$ is modeled for the continuous phase, then an anisotropic approximation is also available:

$$\mathbf{D}_{TD} = C_0 \frac{3}{2} \frac{C_\mu}{\sigma_\alpha} \frac{k_c}{\varepsilon_c} \mathbf{R}_c \tag{2342}$$

Instead of choosing a model for the tensor diffusivity, $\mathbf{D}_{TD}$, the essential difference between alternative dispersion coefficient models can be encapsulated as an effective Turbulent Prandtl Number, σ_α. This number signifies the ratio of momentum diffusivity over the volume fraction diffusivity due to continuous phase velocity fluctuations.

Note: For turbulent diffusion of energy (temperature) this ratio is known as Turbulent Prandtl number while for turbulent diffusion of concentration this ratio is known as Turbulent Schmidt number. Since there is no convention for general variables, the preference here is to name the quantity as Turbulent Prandtl number.

There are four modeling options for the Turbulent Prandtl number:

- Constant, with default value 1 as for a passive scalar
- User-specified expression
- Tchen turbulent dispersion coefficient, recommended for small particles or bubbles
- Inertial turbulent dispersion coefficient, recommended for larger particles or bubbles

For a multiple flow regime phase interaction, the turbulence dispersion force is calculated only for the first regime and the second regime. The weight function of the interface regime is taken as zero.

The turbulent dispersion force is calculated as:

$$\boldsymbol{F}^{TD} = \left(W_{fr} \frac{A_{D,fr} \upsilon_p^t}{\sigma_p} + W_{sr} \frac{A_{D,sr} \upsilon_s^t}{\sigma_s} \right) \left(\frac{\nabla \alpha_s}{\alpha_s} - \frac{\nabla \alpha_p}{\alpha_p} \right) \tag{2343}$$

where:

- υ_p^t, υ_s^t are the primary and secondary phase turbulent kinematic viscosity, respectively.
- σ_p, σ_s are the primary and secondary turbulent Prandtl number, respectively.

Tchen Turbulent Dispersion Coefficient

Thai-Van et al. [501] describe a model for effective turbulent diffusivity of particles. This is based on Simonin's extension of Tchen analysis of particle behavior in steady homogeneous turbulence.

Expressed as an effective Turbulent Prandtl number for the Turbulent Dispersion Force term, this model becomes:

$$\sigma_\alpha = \sigma_0 \sqrt{1 + C_\beta \xi^2} \frac{1 + \eta}{b + \eta} \tag{2344}$$

where:

- σ_0 is the unmodified Turbulent Prandtl number, assumed to be unity to represent basic passive diffusivity.
- C_β is a calibration coefficient for the crossing-trajectories effect.

- ξ is the particle slip velocity scaled by turbulent fluctuation velocity.
- η is the particle-eddy interaction time, scaled by particle relaxation time.
- b is the ratio of the coefficients of the continuous/disperse acceleration terms in the equation of motion for a particle.

For further details, see Tchen Closures.

The general effect of high crossing speed ($\xi \gg 1$) in this model is to reduce correlation and dispersion through an increased effective Turbulent Prandtl number.

For bubbly flows (phase densities $\rho_d \ll \rho_c$) with suitable interaction times, the effect of virtual mass in this model could be to increase dispersion by up to a factor of 1.5.

Inertial Turbulent Dispersion Coefficient

If the particles are larger than the inner turbulent scale, they hold a substantial share of the fluctuating kinetic energy and particle-particle interaction is an important energy redistribution mechanism. Gas-particle flows with high particle concentrations are best modeled as granular flows, but for more dilute particle concentrations (α_d between about 1% to 10%), it is often possible to obtain useful results with an inertial correction to the turbulent dispersion.

The effective Turbulent Prandtl number for inertial dispersion of heavy particles is modeled as:

$$\sigma_\alpha = \frac{\sigma_0}{\left(1 + \frac{\tau_R}{\tau_T}\right)\left(1 + \frac{\rho_c C_{VM}}{\rho_d + \rho_c C_{VM}}\right)} \tag{2345}$$

where:

- σ_0 is the unmodified Turbulent Prandtl, assumed to be unity to represent basic passive diffusivity.
- τ_R is the particle relaxation time:

$$\tau_R = \frac{\rho_d \alpha_d}{A^D_{cd}}\left(1 + \frac{\rho_c}{\rho_d} C_{VM}\right) \tag{2346}$$

- τ_T is an eddy-turnover time:

$$\tau_T = C_\mu \frac{k_c}{\varepsilon_c} \tag{2347}$$

- C_{VM} is a virtual mass coefficient for particle response, which is taken from the Virtual Mass Force model if that has been activated for simulation of an accelerating mean flow; otherwise it is assumed to have a constant value of 0.5.

The first factor in the denominator of Eqn. (2345) is accounting for the increase of the longer particle relaxation times ($\tau_R \gg \tau_T$), while the second one originates from the virtual mass force (see Contribution of Virtual Mass to Turbulent Dispersion for details). The general effect of the model is to increase dispersion through a reduced effective Turbulent Prandtl number.

Wall Lubrication

Antal et al. proposed that bubbles rising close to a wall experience a force from the fluid flowing asymmetrically around the bubble. This force prevents the bubbles from touching the wall.

The force per unit volume, $\mathbf{F}_{WL}$, experienced by a dispersed phase at a distance, y_w, from a wall is:

$$\mathbf{F}_{ij}^{WL} = -C_{WL}(y_w)\alpha_d\rho_c\frac{\left|\mathbf{v}_{r,\ \parallel}\right|^2}{d_p}\mathbf{n} \tag{2348}$$

where:

α_d is the bubble volume fraction.

d_p is the diameter of the bubble.

C_{WL} is a function of inverse length. Its value decreases rapidly with increasing distance, y_w.

$\mathbf{n}$ is the outward facing unit normal at the nearest point on the wall, so that the force is pointing inwards to prevent the bubbles from contacting the wall.

The velocity scale in this model is based on the slip velocity component that is parallel to the wall:

$$\mathbf{v}_{r,\ \parallel} = \mathbf{v}_r - (\mathbf{v}_r \cdot \mathbf{n})\mathbf{n} \tag{2349}$$

The Antal model [398] is implemented in STAR-CCM+ using two calibration coefficients, C_{w1} and C_{w2}:

$$C_{WL}(y_w) = max\left\{C_{w1} + \left(\frac{C_{w2}}{y_w}\right)d, 0\right\} \tag{2350}$$

The coefficients are chosen such that C_{w1} becomes zero a few bubble diameters away from the wall. That is, when:

$$\frac{y_w}{d} > \frac{-C_{w2}}{C_{w1}} \tag{2351}$$

In practice, this means that the force is zero at five bubble diameters. The default value for C_{w1} is -0.01 and the default value for C_{w2} is 0.05.

STAR-CCM+ also allows you to implement alternative models for the wall lubrication force by defining field functions for C_{WL}.

Heat and Mass Transfer

Contents:

Thermal Diffusion
Evaporation and Condensation
Mass Transfer Coefficients

Bulk Boiling
Wall Boiling
Dissolution

Thermal Diffusion

The heat transfer by thermal diffusion between phase i and phase j is modeled as:

$$Q_{ij} = -Q_{ji} = h^{(ij)} a_{ij} \left(T_j - T_i \right) \tag{2352}$$

where:

- Q_{ij} is the interphase heat transfer rate to phase i from phase j
- $h^{(ij)}$ is the mean surface average heat transfer coefficient
- a_{ij} is the interfacial area per unit volume

If c and d represent the continuous and dispersed phases in the phase interaction pair (ij), the mean heat transfer coefficient is computed as a function of the continuous phase thermal conductivity, the Nusselt number, and the interaction length scale:

$$h^{(ij)} = \frac{k_c Nu}{l_{cd}} \tag{2353}$$

Ranz-Marshall Correlation

According to Ranz-Marshall [481], the Nusselt number is computed as:

In STAR-CCM+ the Nusselt number is:

$$Nu = 2 + 0.6 Re_d^{0.5} Pr_c^{0.3} \tag{2354}$$

where:

- Pr_c is the continuous phase Prandtl number
- Re_d is the dispersed phase Reynolds number

For a multiple flow regime phase interaction, the calculation of the interphase heat transfer is analogous to drag.

The total source term for the energy equation is calculated as:

$$S_T = \sum_{t = fr,\ ir,\ sr} W_t h_{T,\ t} T_r \quad T_r = T_s - T_p \tag{2355}$$

The interphase heat transfer coefficient (h_T) for the first regime is calculated in the same way as for the continuous-dispersed phase interaction. The primary phase is considered as the continuous phase and the secondary phase is considered as the dispersed phase. The interphase heat transfer coefficient for the second regime is calculated considering the secondary phase as the continuous phase and the primary phase as the dispersed phase.

The Nusselt number, Nu, is also calculated in the same manner:

$$h_{T,\ fr} = \frac{k_p Nu_{fr} a_{i,\ fr}}{l_{i,\ fr}} \tag{2356}$$

$$h_{T,\ sr} = \frac{k_s Nu_{sr} a_{i,\ sr}}{l_{i,\ sr}} \tag{2357}$$

For the interface regime, the following approach is used:

$$h_{T,\ ir} = \frac{k_m Nu_{ir} a_{i,\ ir}}{l_{i,\ ir}} \tag{2358}$$

$k_m = \alpha_p k_p + \alpha_s k_s$

$a_{i,ir} = 0.5\left|\nabla\alpha_p\right|\left|\nabla\alpha_s\right| / \left(\left|\nabla\alpha_p\right| + \left|\nabla\alpha_s\right|\right)$

$l_{i,\ ir} = \sqrt[3]{0.5\Delta\forall}$

This definition is consistent with the form of the interphase heat transfer coefficient, while providing the freedom to control the slip in temperature by setting Nu_{ir}.

Evaporation and Condensation

The Single Component and Multicomponent Droplet Evaporation Mass Transfer Rate models calculate the mass transfer rate from single and multi-component droplets to a multi-component gas. The Multicomponent model assumes that the liquid drops are internally homogeneous and that the liquid behaves like an ideal mixture.

Mass Transfer

Mass transfer (which can be evaporation or condensation) occurs when the system departs from the equilibrium condition:

$$Y_i^{eq} = K_i(P,T)\, X_i^{eq} \tag{2359}$$

where:

- Y_i^{eq} is the interface equilibrium mole fraction (gas side) of the evaporating species i
- K_i is the equilibrium coefficient for the species i. This coefficient is normally a function of pressure and temperature.
- X_i^{eq} is the interface equilibrium mole fraction (liquid side) of the evaporating species i

For the single-component case, this expression Eqn. (2359) simplifies to:

$$Y_{eq} = K(P,T) \tag{2360}$$

Evaporation occurs when the gas bulk mole fraction of the evaporating component is below the equilibrium value Y_{eq}, otherwise condensation occurs.

Using the model from Spalding [494], the total mass transfer rate m_i is calculated according to:

$$\dot{m}_i = -\epsilon_i g^* A_s \ln(1+B) \tag{2361}$$

where:

- B is the Spalding transfer number.

  This value can be either positive or negative. If B is negative, condensation (treated as ”reverse evaporation”) occurs.
- g^* is the mass transfer conductance (in the limit $B \rightarrow 0$)
- A_s is the area of the droplet surface.
- ϵ_i is the fractional mass transfer rate, which has the property:

  $\sum_T \epsilon_i = 1$

  with the summation over the T transferred components.

For the single-component case, Eqn. (2361) simplifies to:

$$\dot{m}_p = -g^* A_s \ln(1+B)$$

The expression for B and g^* depends on the evaporation mode. Two evaporation modes are implemented: the vapor diffusion limited evaporation mode and the heat transfer limited evaporation mode. STAR-CCM+ applies the appropriate evaporation mode automatically according to the value of $\sum_T Y_{i,eq}$ where $Y_{i,eq}$ is the equilibrium value of transferred species i on the vapor side of the liquid-vapor interface (see Eqn. (2361)).

Vapor Diffusion Limited Evaporation

The vapor diffusion limited evaporation mode occurs when the droplet surface is not saturated (that is, $\sum_T Y_{i,eq} < 1$) and the evaporation rate depends on the rate at which the vapor species can diffuse away from the droplet. Under this condition, the fractional mass transfer rate and the mass transfer number are calculated as:

$$\epsilon_i = \frac{Y_{i,m}(1+B) - Y_i}{B} \tag{2362}$$

$$B = \frac{\sum_T Y_{i,m} - \sum_T Y_i}{1 - \sum_T Y_{i,m}} \tag{2363}$$

where:

- Y_i is the bulk gas vapor mass fraction of the evaporating component i.
- $Y_{i,m}$ is the equilibrium vapor mass fraction:

$$Y_{i,m} = Y_{i,eq} \frac{W_i}{W_s} \tag{2364}$$

where:

- W_i is the molecular weight of the vapor at the drop surface.
- W_s is the molecular weight of the gas mixture at the drop surface.

For the single-component case, the expression for the mass transfer number, Eqn. (2363), simplifies to:

$$B = \frac{Y_{v,s} - Y_v}{1 - Y_{v,s}} \tag{2365}$$

The conductance g^* in this evaporation mode is:

$$g^* = \frac{\rho D_v Sh_p}{D_p} \tag{2366}$$

where:

- D_v is the molecular diffusivity of the vapor
- D_p is the drop diameter
- Sh_p is the Sherwood number.

Heat Transfer Limited Evaporation

The heat transfer limited evaporation mode is active when the vapor at the droplet surface is saturated (that is, $\sum_T Y_{i,eq} \geq 1$). This evaporation mode is also used when there are no inert species in the gas.

Under this condition, the volatile components vaporize in proportion to their mass fraction according to ([493]):

$$\epsilon_i = \frac{Y_{ip}}{\sum_T Y_{ip}} \tag{2367}$$

It is assumed that heat does not penetrate the droplet and that the evaporation rate results from the balance between heat transfer and latent heat due to evaporation. This mode requires thermal non-equilibrium between the phases for mass transfer to occur.

The transfer number is:

$$B = \frac{C_p\left(T_g - T_d\right)}{\sum_T \epsilon_i L_i} \tag{2368}$$

where:

- T_g is the gas temperature
- T_d is the drop temperature
- c_p is the specific heat of the gas
- L_i is the latent heat of vaporization of species i, which is calculated as the difference between the vapor and liquid enthalpies at the drop temperature.

For the single-component case, the expression for the transfer number, Eqn. (2368), simplifies to:

$$B = \frac{c_p\left(T_g - T_d\right)}{L} \tag{2369}$$

The conductance is:

$$g^* = \frac{kNu}{c_p D_p} \tag{2370}$$

where:

- Nu is the Nusselt number
- k is the thermal conductivity of the gas.

Heat Transfer

The droplet evaporation mass transfer rate model assumes thermal equilibrium inside the drops: the model neglects temperature gradients inside the drop, so the interface temperature is assumed to be the same as the inner drop temperature. The energy balance at the drop interface is:

$$q_L - q_G = \sum_T \dot{m}_i L_i \tag{2371}$$

where:

- q_L is the heat that is transferred by the liquid
- $q_G = h\left(T_g - T_d\right)$ is the heat that is transferred by gas to the interface
- h is the heat transfer coefficient
- L_i is the latent heat of vaporization of species i.

The value of q_L is derived from Eqn. (2371) after the calculation of $\dot{m}_i$ and q_G.

Multi-Component Mass Transfer

If a phase is chosen to be a multi-component gas, liquid, or solid (particle), then the segregated species solver can be selected for that phase.

This solver allows multiple species transport equations to be solved:

$$\frac{\delta \rho_j \alpha_j Y_{j,i}}{\delta t} + \nabla \cdot \left(\rho_j \alpha_j Y_{j,i} \mathbf{v}_j\right) = \nabla \cdot \left(\rho_j \alpha_j D_j \nabla Y_{j,i}\right) + S_{j,i} + m'_{j,i} \tag{2372}$$

where:

- ρ_j, α_j and $\mathbf{v}_j$ are the density, volume fraction and velocity of phase j
- $Y_{j,i}$ is the mass fraction of species i in phase j, that is, the fraction of the total mass of that phase
- D_j is the mass diffusivity; combining molecular and turbulent diffusivity
- $S_{j,i}$ is a general mass source
- $m'_{j,i}$ is used here to refer specifically to the transfer of species from one phase to another.

Note: If there is a single phase with α_j equal to 1 and no interphase mass transfer, this equation reverts to the species transport equation for a single phase.

Similar to a single phase, boundary, and initial conditions are required for all species for all multi-component phases.

Mass Transfer Coefficients

The mass transfer coefficient k is found using Sherwood numbers for each phase:

$$k = Sh\frac{D}{l} \tag{2373}$$

where D is the species diffusivity within a phase and l is the phase interaction length scale.

A single Sherwood number is assumed to apply for the whole phase. The Sherwood number for a dispersed phase or continuous phase can be specified by a constant or by a field function.

Ranz-Marshall Method

The Sherwood number for a continuous phase can also be calculated using a Ranz-Marshall method:

$$Sh = 2 + 0.6Re^{1/2}Sc^{1/3} \tag{2374}$$

where:

- Re is the phase pair Reynolds Number
- $\mathrm{Sc} = \frac{\mu}{\rho \mathrm{D}}$ is the Schmidt number

Bulk Boiling

The bulk boiling model in STAR-CCM+ is an essential part of subcooled boiling. It models boiling and condensation in the interior of the flow.

The rate of mass transfer between the phases depends only on the heat transfer rate from the liquid and vapor to a saturated interface between these two phases. It is important that the interfacial area density is provided by a well-defined characterization of bubbles by their size and number density.

This thermally driven model is not suitable for describing boiling without bubbles (homogeneous nucleation), evaporation (concentration-driven multi-component mass transfer), nor cavitation (inertially limited mass transfer). It is also not suitable when the resulting interphase mass transfer rate occurs on time scales that are much shorter than flow time scales. This scenario is unlikely to be solvable with a segregated solver algorithm.

The heat transfer from the phase-change interface to each of the two phases on either side of the interface is modeled as:

$$Q_i^{(ij)} = h_i^{(ij)} a_{ij}\left(T_{ij} - T_i\right) \tag{2375}$$

$$Q_j^{(ij)} = h_j^{(ij)} a_{ij}\left(T_{ij} - T_j\right) \tag{2376}$$

where:

- T_i and T_j are the bulk temperatures of the continuous/primary and dispersed/secondary phases, respectively
- T_{ij} is the interface temperature

The interface temperature is often defined as a constant saturation temperature for a particular system pressure. In STAR-CCM+, a field function can optionally be used for the interface temperature, so that depth can be accounted for, or overpressure due to surface tension in micron-sized bubbles.

The heat transfer coefficient for energy supply from the continuous phase is:

$$h_i^{(ij)} = Nu_i k_i / l_{ij} \tag{2377}$$

where Nu_i is the continuous phase Nusselt number and l_{ij} is the interaction length scale for the phase pair, usually the bubble size.

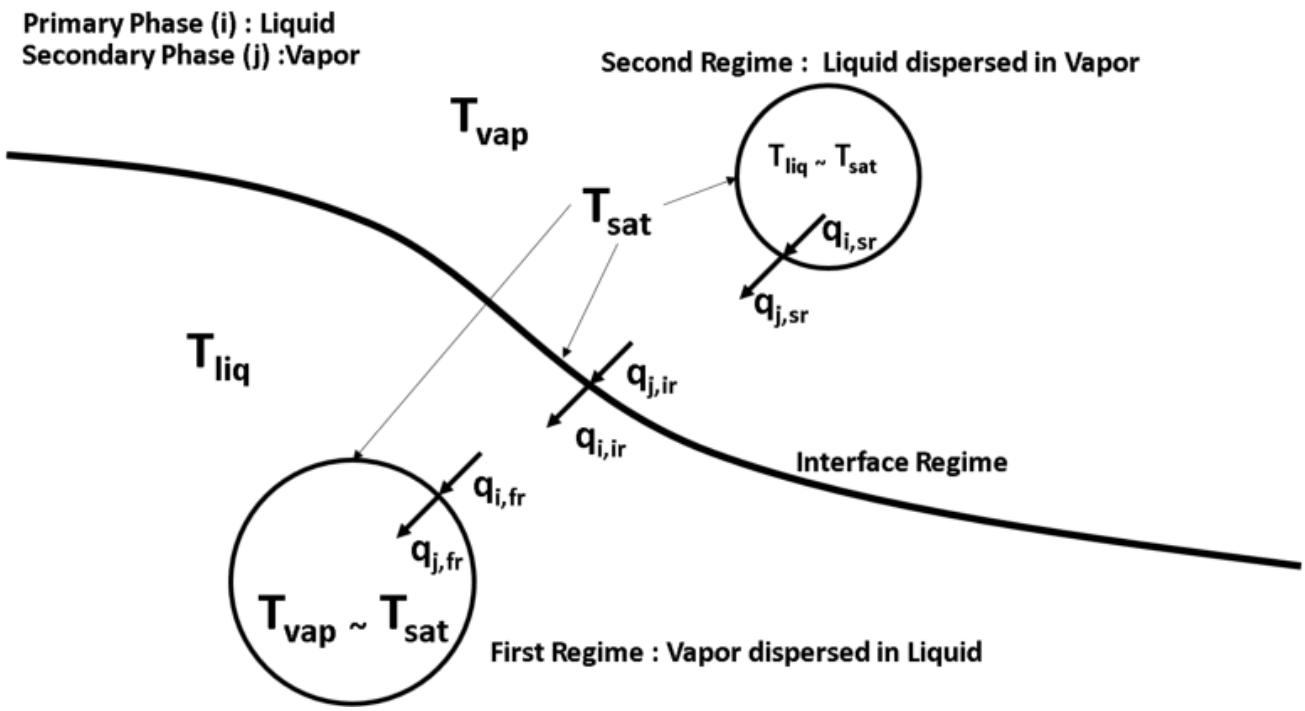
Similarly, for the dispersed phase, the heat transfer coefficient for energy supply is:

$$h_j^{(ij)} = Nu_j k_j / l_{ij} \tag{2378}$$

where Nu_j is the dispersed phase Nusselt number.

The Nusselt number of the continuous phase is calculated using the Eqn. (2354). This correlation is not available for the dispersed phase Nusselt number since convective processes are often defined poorly inside the particle.

In a multiple flow regime phase interaction, a typical scenario for bulk boiling/condensation is shown below:



The heat transfer is calculated for a total of six situations for a pair of primary phase (i) and secondary phase (j). That is, for each regime, the heat transfer from a phase to the interface is calculated. The heat-transfer coefficient for the energy supply from the primary phase and secondary phase is the weighted sum of the heat-transfer coefficient at the interface for the first regime, the interface regime, and the second regime:

$$h_i^{(ij)} a_{ij} = \sum_{r\,=\,fr,\,ir,\,sr} W_r \left(h_{i,r}^{(ij)} a_{ij,r} \right) \tag{2379}$$

$$h_j^{(ij)} a_{ij} = \sum_{r\,=\,fr,\,ir,\,sr} W_r \left(h_{j,r}^{(ij)} a_{ij,r} \right) \tag{2380}$$

In the above equations, the primary and secondary side heat transfer coefficient for different regimes is calculated as:

- For the primary (liquid) side:
  - first regime, $h_{i,fr} = Nu_{i,fr}\frac{k_i}{l_{fr}}$
  - interface regime $h_{i,ir} = Nu_{i,ir}\frac{k_i}{l_{ir}}$
  - second regime $h_{i,sr} = Nu_{i,sr}\frac{k_i}{l_{sr}}$
- For the secondary (vapor) side:
  - first regime, $h_{j,fr} = Nu_{j,fr}\frac{k_j}{l_{fr}}$
  - interface regime $h_{j,ir} = Nu_{j,ir}\frac{k_j}{l_{ir}}$
  - second regime $h_{j,sr} = Nu_{j,sr}\frac{k_j}{l_{sr}}$

Note: In the first and second regime, there can be specific situations when high convective currents are not expected within the spherical body. In these situations, the bubbles and droplets can be treated as lumped bodies. To ensure that there is a uniform temperature within the spherical body (that is, the vapor side for the first regime and the liquid side for the second regime), the Nusselt number is calculated using the corresponding fluid properties as $\frac{h_{internal} \cdot l}{k}$, where $h_{internal} = \frac{\rho \cdot cp \cdot l}{\Delta t_s}$. Δt_s is the relaxation time-scale to ensure rapid equalization of temperature, and can be set to a low value, for example, $\Delta t_s = 1e^{-5}s$.

Hughes and Duffey Correlation

The liquid side heat transfer coefficient in the interface regime is calculated using the surface renewal theory that is introduced by Hughes and Duffey ([439]). The Nusselt number predicted by this theory is:

$$Nu_{i,ir} = \frac{2}{\sqrt{\pi}}\mathrm{Re}_t\mathrm{Pr}_i^{1/2} \tag{2381}$$

where:

- $\mathrm{Re}_t = l_t u_t / \nu_l$ is the turbulent Reynolds number
  - l_t is the turbulent length scale
  - u_t is the turbulent velocity scale
  - ν_l is the liquid kinematic viscosity.
- P_{rl} is the liquid Prandtl number.

This method is available only when the primary phase is a turbulent model.

The interface mass flux is computed from the heat balance:

$$m^{(ij)} = -\frac{Q_i^{(ij)} + Q_i^{(ij)}}{\Delta h_{ij}} \tag{2382}$$

where:

- $m^{(ij)}$ is the signed mass transfer rate per unit interfacial area for the interaction from phase j to phase i
- Δh_{ij} is the signed enthalpy input to create phase j from phase i

Distinguish interphase mass transfer source and sink terms appearing in various phase transport equations according to sign. This technique allows you to import or export different interface values of the transported variable:

$$m_{ij} = max\left(m^{(ij)}, 0\right) \tag{2383}$$

$$m_{ji} = max\left(-m^{(ij)}, 0\right) \tag{2384}$$

so that the signed flux $m^{(ij)}$ is equivalent to $m_{ij} - m_{ji}$.

Contents:

Multi-Component Bulk Boiling

Multi-Component Bulk Boiling

The multi-component boiling mass transfer model calculates the molar flux and the temperature at the phase interface. These terms are needed for the calculation of mass transfer, momentum transfer, and energy transfer between the phases.

Consider a stationary coordinate reference frame and define $\mathbf{v}_i$ as the velocity of component *i* in this frame. The molar flux of species i is expressed as:

$$\mathbf{N}_i = c_i \mathbf{v}_i \left[\frac{\mathrm{mol}}{\mathrm{m}^2\mathrm{s}}\right] \tag{2385}$$

Sum over all the species to get the total molar flux:

$$\mathbf{N}_t = \sum_{i=1}^{n} \mathbf{N}_i = c_t \mathbf{v} \tag{2386}$$

where the molar average velocity $\mathbf{v}$ is defined as:

$$\mathbf{v} = \sum_{i=1}^{n} x_i \mathbf{v}_i \tag{2387}$$

The diffusion flux, which is the flux of component *i* relative to the flux of the mixture, can now be defined. Choose $\mathbf{v}$ as the mixture reference velocity and therefore define the molar diffusion flux as:

$$\mathbf{J}_i = c_i\left(\mathbf{v}_i - \mathbf{v}\right) \tag{2388}$$

with:

$$\sum_{i=1}^{n} \mathbf{J}_i = 0 \tag{2389}$$

The molar flux $\mathbf{N}_i$ is related to the molar diffusion flux by:

$$\mathbf{N}_i = \mathbf{J}_i + c_i\mathbf{v} = \mathbf{J}_i + x_i\mathbf{N}_t \tag{2390}$$

Film Model

The assumptions of the Film Model can be summarized as follows:

- All the resistance to mass transfer is localized in a thin film that is adjacent to the phase boundary.
- Mass transfer occurs inside this film by steady-state molecular diffusion alone.
- Mass transfer through the film occurs in the direction normal to the interface (that is, composition gradients along the interface are neglected).
- There are no composition gradients outside the film.

Diffusion Flux Equation

With the previous assumptions, the species concentration gradients across the film can be calculated using finite differences and the equation for the diffusion flux becomes:

$$J_i = N_i - x_i N_t = c_t \sum_{j=1}^{n-1} k_{ij}^* \, \Delta \, x_j \tag{2391}$$

where we express the finite flux mass transfer coefficient matrix as:

$$[k^*] = [k][\Xi] \tag{2392}$$

with $[k]$ and $[\Xi]$ being the low-flux mass transfer coefficient matrix and the finite flux correction coefficient matrix respectively.

Mass Transfer Coefficient in the Zero Flux Limit

The low-flux mass transfer coefficient matrix $[k]$ is defined as:

$$\left[k\right] = \frac{1}{l}[B]^{-1} \tag{2393}$$

$$B_{ii} = \frac{x_i}{D_{in}} + \sum_{\substack{k=1 \\ k \neq i}}^{n} \frac{x_k}{D_{ij}} \tag{2394}$$

$$B_{ij} = -x_i\left(\frac{1}{D_{ij}} - \frac{1}{D_{in}}\right) \tag{2395}$$

The length scale l that appears in Eqn. (2393) is derived from the film theory and is the thickness of the film layer. This quantity can be calculated from the Sherwood number, since:

$$\mathrm{Sh} = \frac{kd}{D} = \frac{d}{l} \tag{2396}$$

from which $l = \frac{d}{\mathrm{Sh}}$ can be obtained.

The resulting film thickness is a function of flow conditions, geometry, and fluid properties. In STAR-CCM+, only one characteristic length d and one Sherwood number Sh can be specified for each phase interaction. Therefore, the same value of l is calculated and applied for all of the transferred components.

Mass Transfer Correction Coefficient Matrix

The form of the correction coefficients depends on the mass transfer model that is adopted. For the two fluid film model, following the convention of positive flux from the vapor to the liquid, the correction matrices are:

$$[\Xi]^V = [\Phi]^V\left[\exp[\Phi]^V - [I]\right]^{-1} \tag{2397}$$

and

$$[\Xi]^L = [\Phi]^L\exp[\Phi]^L\left[\exp[\Phi]^L - [I]\right]^{-1} \tag{2398}$$

for the vapor and the liquid respectively, where:

$$\Phi_{ii} = \frac{N_i}{c_t\kappa_{in}} + \sum_{\substack{k=1\\k\neq i}}^{n}\frac{N_k}{c_t\kappa_{ik}} \tag{2399}$$

$$\Phi_{ij} = -N_i\left(\frac{1}{c_t\kappa_{ij}} - \frac{1}{c_t\kappa_{in}}\right) \tag{2400}$$

Interface Energy Balance

Continuity of the energy flux across the vapor - liquid interface is imposed:

$$E^V = E^L = E \tag{2401}$$

where E^V and E^L are the normal components of the energy flux at the interface. Eqn. (2401) can be rewritten as:

$$q^V + \sum_{i=1}^{n} N_i^V\overline{H}_i^V\left(T^I\right) = q^L + \sum_{i=1}^{n} N_i^L\overline{H}_i^L\left(T^I\right) \tag{2402}$$

with the heat fluxes given by:

$$q^V = h_V{}^*\left(T^V - T^I\right) \tag{2403}$$

$$q^L = h_L{}^*\left(T^I - T^L\right) \tag{2404}$$

The finite flux heat transfer coefficient in the vapor phase is calculated according to the film model as:

$$h_V{}^* = h^V\left(\Phi_H^V\right)/\left(\exp\Phi_H^V - 1\right) = \Xi_H h^V \tag{2405}$$

where h^V is the zero-flux heat transfer coefficient, $\Xi_H = \left(\Phi_H^V\right)/\left(\exp\Phi_H^V - 1\right)$ is the Ackermann correction factor and

$$\Phi_H = \sum_{i=1}^{n} N_i W_i c_{pi}/h \tag{2406}$$

Similarly, for the liquid:

$$h_L{}^* = h^L\left(\Phi_H^L\right)\exp\Phi_H^L/\left(\exp\Phi_H^L - 1\right) = \Xi_H^L h^L \tag{2407}$$

with $\Xi_H^L = \left(\Phi_H^L\right)\exp\Phi_H^L/\left(\exp\Phi_H^L - 1\right)$.

The high flux correction for h^L is expected to be close to unity due to the high value of the liquid phase heat transfer coefficient.

Species Equilibrium at the Interface

The composition of the phases on either side of the interface between the two phases are assumed to be in instant local equilibrium with each other and with the interface temperature and the system pressure:

$$y_i^I = K_i^0\left(P, T^I\right)x_i^I \tag{2408}$$

where ideal behavior of the mixtures is assumed. The mole fractions also obey the constraints:

$$\sum x_i^I = 1 \tag{2409}$$

$$\sum y_i^I = 1 \tag{2410}$$

Note that a solution of these interface state relations is possible only if at least one of the vapor-liquid equilibrium coefficients K_i^0 is less than unity while at least one of the K_i^0 is greater than unity.

Equations of the Point Solver

Consider two mixtures x and y divided by an interface. The superscripts L and V are used for bulk quantities in the liquid and in the vapor respectively, while I indicates interface quantities. The interface is considered as a surface with no resistance to mass transfer and where equilibrium conditions prevail at any time. It is assumed that no chemical reactions occur at the interface or in the film.

When the interface is not stationary, the fluxes $\mathbf{N}_i$ refer to the interface and not to the laboratory fixed frame of reference (see [500] for more details). It is assumed that the mass transfer takes place in a direction normal to the interface:

$$N_i^x = N_i^y = N_i \;\; i = 1, 2, ..., n \tag{2411}$$

$$i = 1n \tag{2412}$$

where N_i^x is the normal component of $\mathbf{N}_i$ and is directed from the bulk phase x to the interface I while N_i^y is directed from the interface I to the bulk phase y.

In the general case, there can be m species in the liquid and n in the gas, not all of them volatile or condensable. Since there are k couples between the volatile and condensable species, then $m-k$ species are non-volatile and $n-k$ species are non-condensable.

Thus, the variables to calculate are m interface mole fractions x_i^I, n interface mole fractions y_i^I, k interface molar fluxes N_i, and the interface temperature T^I.

Therefore, the total number of variables is $m+n+k+1$ and the equations of the model are:

- $m-1$ equations from the interface molar fluxes (liquid side)

$$N_i - x_i^I N_t = c_t^L \sum_{i=1}^{n-1} k_{il}^{*L}\left(x_j^I - x_j\right) \quad i = 1,\ k+1,\ \ldots,\ m-1 \tag{2413}$$

- $n-1$ equations from the interface molar fluxes (vapor side)

$$N_i - y_i^I N_t = c_t^V \sum_{i=1}^{n-1} k_{ij}^{*V}\left(y_j - y_j^I\right) \quad i = 1,\ k+1,\ \ldots,\ n-1 \tag{2414}$$

- 1 equation from the energy balance at the interface

$$h_{V*}\left(T^V - T^I\right) + \sum_{i=1}^{n} N_i^V \overline{H}_i^V W_i\left(T^I\right) = h_{L*}\left(T^I - T^L\right) + \sum_{i=1}^{m} N_i^L \overline{H}_i^L W_i\left(T^I\right) \tag{2415}$$

- k equations from the equilibrium relations at the interface

$$y_i^I = K_i x_i^I \quad i = 1, 2, \ldots, k \tag{2416}$$

- 1 equation from the constraint on mole fractions at the interface (liquid side)

$$\sum_{i=1}^{m} x_i^I = 1 \tag{2417}$$

- 1 equation from the constraint on mole fractions at the interface (vapor side)

$$\sum_{i=1}^{n} y_i^I = 1 \tag{2418}$$

The number of equations is, therefore, $m+n+k+1$ as the number of variables. If species i is not volatile or non-condensable, then $N_i = 0$ in Eqn. (2413) and Eqn. (2414). If the liquid is a pure substance ($m = 1$), there are no equations of the form Eqn. (2413) and Eqn. (2416) is always satisfied.

Likewise, if the vapor is a pure substance ($n = 1$) then there are no equations of the form Eqn. (2414) and Eqn. (2417) is always satisfied.

Eqn. (2413) to Eqn. (2417) are a system of non-linear algebraic equations that are solved for each cell of the computational domain with a Newton solver.

Reduced Model for Dilute Dispersions

When the Maxwell-Stefan diffusion model is not active, the assumption of dilute dispersion is made, that is $x_i \approx 0$ for $i = 1, 2, \ldots, n-1$ and $x_n \approx 1$. With this assumption, only the interactions between each disperse species and the last species in the list (the solvent) are accounted for. The interactions between the disperse species are neglected. Consequently, Eqn. (2413) and Eqn. (2414) reduce to:

$$N_i - x_i^I N_t = c_t^L \kappa_{in}^L \frac{\Phi_{ii}^L e^{\Phi_{ii}^L}}{e^{\Phi_{ii}^L} - 1}\left(x_i^I - x_i\right) \tag{2419}$$

and

$$N_i - y_i^I N_t = c_t^V \kappa_{in}^V \frac{\Phi_{ii}^V}{e^{\Phi_{ii}^L} - 1} \left(y_i - y_i^I \right) \tag{2420}$$

respectively, with $\kappa_{in} = \frac{D_{in}}{l}$ and $\Phi_{ii} = \frac{N_t}{c_t \kappa_{in}}$

The other equations are not affected by the assumption of dilute dispersion.

Nomenclature

The terms that are used in the equations of the Multi-component Boiling Mass Transfer model are defined in the following table.

c_i	Molar density of $i = \rho_i / M_i$	$\left[\frac{\mathrm{mol}}{\mathrm{m}^3}\right]$
c_t	Mixture molar density $= \sum_{i=1}^{n} c_i$	$\left[\frac{\mathrm{mol}}{\mathrm{m}^3}\right]$
c_p	Specific heat	$\left[\frac{\mathrm{J}}{\mathrm{kgK}}\right]$
d	Characteristic length scale of the system	m
$\lvert D \rvert$	Matrix of Fick diffusion coefficients	$\left[\frac{\mathrm{m}^2}{\mathrm{s}}\right]$
$\lvert \mathrm{D} \rvert$	Matrix of Maxwell-Stefan diffusion coefficients	$\left[\frac{\mathrm{m}^2}{\mathrm{s}}\right]$
h	Heat transfer coefficient	$\left[\frac{\mathrm{W}}{\mathrm{m}^2\mathrm{K}}\right]$
H	Specific enthalpy	$\left[\frac{\mathrm{J}}{\mathrm{kg}}\right]$
$\lvert \mathbf{I} \rvert$	Unit tensor	
$\mathbf{J}_i$	Molar diffusion flux $= c_i(\mathbf{v}_i - \mathbf{v})$	$\left[\frac{\mathrm{mol}}{\mathrm{m}^2\mathrm{s}}\right]$
l	Film thickness	m
M_i	Molar mass of component i	$\left[\frac{\mathrm{kg}}{\mathrm{mol}}\right]$
$\mathbf{N}_i$	Molar flux of component i	$\left[\frac{\mathrm{mol}}{\mathrm{m}^2\mathrm{s}}\right]$
Sh	Sherwood number	
$\mathbf{v}$	Molar averaged referenced velocity $= \sum_{i=1}^{n} x_i \mathbf{v}_i$	$\left[\frac{\mathrm{m}}{\mathrm{s}}\right]$
x_i, y_i	Bulk mole fraction of component i (liquid side, vapor side)	

x_i^I , y_i^I	Interface mole fraction of component i (liquid side, vapor side)	
W_i	Molecular weight of i	$\left[\frac{\mathrm{g}}{\mathrm{mol}}\right]$
μ_i	Molar chemical potential of i	$\left[\frac{\mathrm{J}}{\mathrm{mol}}\right]$
ρ_i	Mass density of $i = c_i M_i$	$\left[\frac{\mathrm{kg}}{\mathrm{m}^3}\right]$
ρ_t	Mixture mass density $= \sum_{i=1}^{n} \rho_i$	$\left[\frac{\mathrm{kg}}{\mathrm{m}^3}\right]$
ω_i	Mass fraction of $i = \rho_i / \rho_t$	
$[\Phi]$	Matrix of mass transfer rate factors	
$[\Xi]$	Matrix of high flux correction factors	

Wall Boiling

The wall boiling model is aimed at forced-flow, subcooled, boiling applications. In subcooled boiling, local boiling occurs on a heated surface despite the mean temperature of the liquid being less than the saturation temperature. The degree of liquid subcooling is expressed either as a temperature difference or an enthalpy difference below the saturation value.

Provided that the flow regime near the wall remains close to bubbly flow, a standard subcooled boiling model can be pushed towards zero subcooling conditions (that is, saturated boiling). The model can also be pushed towards higher heat fluxes approaching departure from nucleate boiling (DNB) conditions.

By definition, the phases in subcooled boiling are not in thermal equilibrium and each phase requires its own energy equation. The phases in subcooled boiling require their own equation of motion, as the bubble moves away from the boiling surface while cool liquid moves to take its place. The condensation rate of vapor in the bulk subcooled liquid is a part of the wall boiling process and decides when net vapor generation first begins on the wall, and how the vapor is distributed downstream of this point.

Within this formulation, some model properties are referred to as “standard values”. These values are parameters that were carefully chosen by the original authors. Do not change unless you are an expert with experimental support. Some models are implemented specifically for water systems, so adjust the highlighted model properties if a working fluid other than water is used.

The subcooled boiling model of Kurul and Podowski [451], with contributions by many other workers, assumes that all the heat flux from the wall is transferred to the liquid next to the wall, and is then partitioned into three components:

$$\dot{q}''_w = \dot{q}''_{conv} + \dot{q}''_{evap} + \dot{q}''_{quench} \tag{2421}$$

These components of heat flux are described below:

- $\dot{q}''_{conv}$ is the convective heat flux given by Eqn. (2423), which describes the removal of heat by single-phase turbulent convection on those parts of the wall that are not affected by boiling. In applications with fixed wall heat flux, this term defines the point at which the wall first exceeds saturation temperature.
- $\dot{q}''_{evap}$ is the evaporative heat flux given by Eqn. (2425), which describes the power that is used to produce bubbles from nucleation to departure. This term is a strong function of wall superheat, $T_w - T_{sat}$. In fixed heat-flux applications, once boiling has started, this term is responsible for maintaining a wall temperature that is slightly higher than the saturation temperature.
- $\dot{q}''_{quench}$ is the quenching heat flux given by Eqn. (2427), which describes the enhancement of heat transfer, due to the replacement of a departing bubble by an influx of cooler liquid farther away from the wall. Bubble-induced quenching is also known in literature as liquid agitation or pumping. Quenching is less important when the liquid is close to saturation temperature, but in highly subcooled flows, $T_l \ll T_{sat}$, it makes an important contribution.

Vapor Contribution

Subcooled boiling models start with the assumption that only the liquid phase is in contact with the wall. However, when the vapor volume fraction next to the wall becomes high enough, liquid access to the wall and its capability to remove heat from the wall are both restricted. This can occur unintentionally during a subcooled boiling simulation, as it passes through various intermediate states from initial conditions towards a converged solution.

Alternatively, the vapor contribution is relevant when studying the transition to DNB (Departure from Nucleate Boiling) conditions as wall heat flux is further increased.

When liquid access is restricted, vapor heat transfer removes some fraction of the wall heat flux and causes an increase in wall temperature. The heat fluxes can be represented as:

$$\dot{q}''_w = \left(\dot{q}''_{conv} + \dot{q}''_{evap} + \dot{q}''_{quench}\right)\left(1 - K_{dry}\right) + K_{dry}\dot{q}''_{dry} \tag{2422}$$

where:

- $\dot{q}''_{dry}$ is the vapor contribution to convective heat flux (see Eqn. (2424)), based on single-phase turbulent convection by the vapor.
- K_{dry} is the wall contact area fraction for the vapor (see Eqn. (2434)), based on either an expression for K_{dry}, or a transition volume fraction representing the start of wall dryout.

Convective Heat Flux

The turbulent convection fluxes are based on single-phase wall treatments.

For liquid contact with the wall, this flux is:

$$\dot{q}''_{conv} = \frac{\rho_l c_{pl} u_l^*}{t_l^+}\left(T_w - T_l\right) \tag{2423}$$

For vapor contact with the wall, this flux is:

$$\dot{q}''_{dry} = \frac{\rho_g c_{pg} u_g^*}{t_g^+}\left(T_w - T_g\right) \tag{2424}$$

The volume fraction does not modify the convection contributions as the contributions are in non-boiling heat transfer, where volume fraction represents the wall contact area fraction for each phase (compare this with Eqn. (2424)).

In the subcooled boiling model, it is assumed that the liquid is in full contact with the wall and is superheated first before vapor can be generated. However, if the wall begins to dry out, the factor, K_{dry}, can be specified to represent the wall contact area fraction for the vapor.

Evaporative Heat Flux

An evaporative heat flux is constructed for the following nucleate boiling mechanism:

- Bubbles nucleate and grow on suitably sized cavities in the wall.
- Bubbles reach a critical size, at which point the forces holding the bubble to the wall are no longer balanced.

The evaporative heat flux is:

$$\dot{q}''_{evap} = n'' f \left(\frac{\pi d_w^3}{6} \right) \rho_g h_{lg} \tag{2425}$$

where:

- n'' is the nucleation site number density
- f is the bubble departure frequency
- d_w is the bubble departure diameter
- ρ_g is the vapor phase density
- h_{lg} is the latent heat

The evaporative heat flux, $\dot{q}''_{evap}$, is the rate of conversion of liquid to vapor per unit wall area:

$$\dot{q}''_{evap} = \frac{\dot{q}''_{evap}}{h_{lg}} \tag{2426}$$

Bubble Induced Quenching Heat Flux

When a bubble detaches from the wall, the space it occupied is replaced by cooler liquid. Quenching heat transfer is the component of heat flux that is used in heating this replacement liquid, as modeled by Del Valle and Kenning [418].

The bubble induced quenching heat flux is implemented using the temperature difference between the wall and the liquid next to the wall, T_l, together with a slowly varying correction factor, s_{quench}:

$$q''_{quench} = h_{quench} s_{quench} \left(T_{wall} - T_l \right) \tag{2427}$$

The correction factor is defined as

$$s_{quench} = \frac{T_{wall} - T_{quench}}{T_{wall} - T_l} = \frac{T^+\left(y_{quench}\right)}{T^+\left(y_c\right)} \tag{2428}$$

where:

- h_{quench} is the quenching heat transfer coefficient

- T_{wall} is the wall temperature
- T_{quench} is the temperature at which liquid is brought to the wall by the departure of a bubble
- $T^{+}(y)$ is the dimensionless temperature profile that is formed by scaling wall heat flux to the liquid using liquid wall turbulence scales.
- y_c is the distance from the wall to the nearest cell center.
- y_{quench} is the representative distance from the wall at which cold liquid is drawn by the departure of a bubble. You specify this distance in terms of bubble departure diameters or of liquid wall turbulence scales.

Overall Energy Balance at the Wall

If other phases are present in the flow, they are assumed not to contact the wall and, therefore, have no influence on the wall boiling process.

Only the liquid phase l, and possibly the vapor phase g, are assumed to be in contact with the wall, as the wall contact area fraction K_{dry} indicates.

The combined models for subcooled boiling with possible dryout (see Eqn. (2422)) fits into the multiphase energy balance at the wall (see Eqn. (2773)) as follows:

$$\dot{q}''_{w} = \dot{q}''_{wl} + \dot{q}''_{w\mathrm{lg}} + \dot{q}''_{wg} \tag{2429}$$

where:

- $\dot{q}''_{wl}$ is the heat flux between the wall and the liquid phase:

$$\dot{q}''_{wl} = \left(1-K_{dry}\right)\left[\left(1-K_{quench}\right)\dot{q}''_{conv} + \dot{q}''_{quench}\right] \tag{2430}$$

- $\dot{q}''_{w\mathrm{lg}}$ is the heat flux between the wall and liquid-gas interface:

$$\dot{q}''_{w\mathrm{lg}} = \left(1-K_{dry}\right)\dot{q}''_{evap} \tag{2431}$$

- $\dot{q}''_{wg}$ is the heat flux between the wall and gas phase:

$$\dot{q}''_{wg} = K_{dry}\dot{q}''_{dry} \tag{2432}$$

You can override any of the standard models for $\dot{q}''_{wl}$, $\dot{q}''_{w\mathrm{lg}}$, and $\dot{q}''_{wg}$ with your own models.

Contents:

Wall Dryout Area Fraction
Nucleation Site Number Density
Bubble Departure Diameter
Bubble Departure Frequency
Bubble Induced Quenching Heat Transfer Coefficient

Wall Dryout Area Fraction

Weisman and Pei [514] postulated that, in wall boiling, a bubbly layer exists near the wall and, at DNB (Departure from Nucleate Boiling), this layer reaches a critical bubble concentration. At this concentration, liquid transport is no longer sufficient to remove the heat flux that is applied at the wall. Weisman-Pei found that the maximum value of average volume fraction was 0.82 in a bubbly layer 5.5 departure diameters thick.

A basic application of this DNB criterion to a CFD calculation, is that:

- The CFD computation replaces Weisman-Pei's analysis of radial transport.
- The computed distribution of bubble volume fraction, averaged over some prescribed bubbly layer thickness, reaches some critical value.

Such a computation would not expect to use the same coefficients as the original Weisman-Pei model. Also, reassess these coefficients if alternative models are selected for the bubble departure diameter.

In STAR-CCM+, there are three options for specifying the bubbly layer thickness for the dryout criterion:

- Wall Cell—the layer thickness is the width of the grid cell next to the wall
- Fixed Number of Diameters—the layer thickness is a factor of the number of bubble departure diameters
- Fixed Yplus—the layer thickness is a factor of the wall turbulence scales for the vapor phase

An output field function, **Bubbly Layer Volume Fraction** is computed on the wall. This can be easily plotted for comparison on different grids and also used in user expressions for modeling wall dryout.

There is no justification for assuming a logarithmic profile exists for bubble volume fraction, $\alpha_g(y)$, near the wall when approaching DNB conditions. Instead, the distribution is approximated by a one-term expansion about the first cell center at distance y_c from the wall, using the wall-normal component of the volume fraction derivative, $\alpha_g{}'$. This distribution can then be averaged over a layer thickness δ.

$$\alpha_\delta = \frac{1}{\delta}\int_0^\delta \mathrm{d}y\left\{\alpha_g(y_c) + \alpha_g{}'(y_c)(y-y_c)\right\} = \alpha_g(y_c) + \alpha_g{}'(y_c)(\delta/2-y_c) \tag{2433}$$

The simplest model for local wall dryout is that heat transfer from the wall to the vapor begins when the vapor volume fraction exceeds a specified value. At the same time, heat transfer from the wall to the liquid convection and liquid evaporation declines. This is expressed as an effective wall contact area fraction for the vapor:

$$K_{dry} = \begin{cases} 0 & \alpha_\delta \le \alpha_{dry} \\ \mathrm{f}(\beta) & \alpha_\delta > \alpha_{dry} \end{cases} \tag{2434}$$

where:

- α_{dry} is the wall dryout break-point or the volume fraction at which dryout begins
- α_δ is the vapor volume fraction averaged over the bubbly layer thickness.
- $\mathrm{f}(\beta)$ is the function:

$$\mathrm{f}(\beta) = \beta^2(3-2\beta) \tag{2435}$$

Which transitions symmetrically and smoothly from 0 through 1 over $0 \le \beta \le 1$, where β is the ratio:

$$\beta = \frac{\alpha_\delta - \alpha_{dry}}{1-\alpha_{dry}} \tag{2436}$$

Which varies from 0 through 1 for $\alpha_{dry} \leq \alpha_{\delta} \leq 1$,

α_{δ} is the vapor volume fraction averaged over the bubbly layer thickness.

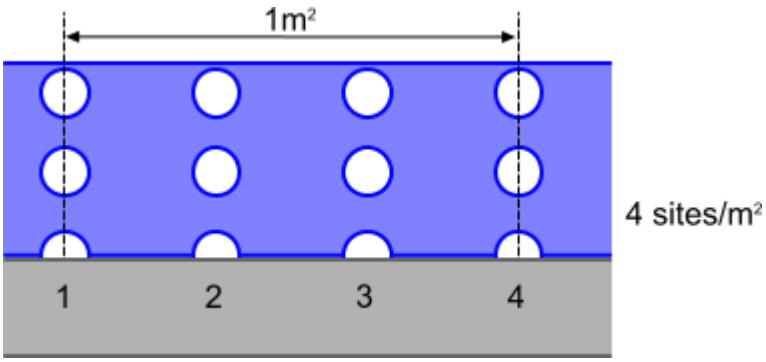
The default value for α_{dry} is 0.9. This value prevents instabilities in the computation due to unintentional dryout, as it reduces the heat flux that is directed to the liquid as the amount of liquid reduces.

Weisman and Pei [514] geometrically derived a maximum vapor volume fraction of 0.82 for use in their critical heat flux model. Adjust the wall dryout break-point, α_{dry}, when studying the transition to DNB conditions.

Nucleation Site Number Density

The nucleation site number density determines the number of locations on the heated surface where bubbles form, per unit area.

This is the leading factor determining the evaporation rate in a mechanistic model of subcooled boiling (see Eqn. (2421)). The diagram below illustrates this parameter:



A nucleation site is a gas-filled cavity in the wall. In practice, there is a range of cavity sizes. A larger wall superheat allows bubble growth in more cavities of smaller size.

Two standard submodels have been implemented for the nucleation site number density:

- Lemmert Chawla – A correlation against wall superheat, and is often used together with the **Tolubinsky Kostanchuk** model for bubble departure diameter.
- Hibiki Ishii – A mechanistic model for active nucleation site number density as a function of critical cavity size and contact angle. It is designed for use with the Kocamustafaogullari model for bubble departure diameter, and is applicable up to 198 bar.

The Lemmert Chawla model is the default option, as its fixed power for superheat dependence means that it is easier to obtain initial solutions. The more recent Hibiki Ishii model has a wider range of applicability, but it can be difficult to reach a converged solution in high flux cases.

Under the 45 bar, 0.57 MW/m^2 conditions of the first test in Bartolomei and Chanturiya [404], these two methods (each used with its recommended departure diameter model) can produce comparable results for axial void and wall superheat. However, the departure diameter that is associated with the Hibiki Ishii and Kocamustafaogullari solution is much smaller.

At higher pressures, and higher heat fluxes, the Hibiki Ishii model has an increasingly stronger dependence on wall superheat. This model often predicts a much smaller wall superheat than the Lemmert Chawla model for the same heat flux.

Lemmert Chawla

The original Lemmert Chawla model [477][455] for nucleation site number density is:

$$\frac{n''}{\left[m^{-2}\right]} = \left(m\Delta T_{sup}\right)^{p} \tag{2437}$$

where:

- n'' is the nucleation site density
- m is a calibration constant, with a default value of 185.0
- p is a superheat exponent, with a default value of 1.805
- ΔT_{sup} is the wall superheat, which is calculated as:

$$\Delta T_{sup} = \min\left(\max\left(T_{wall} - T_{sat}, 0\right), \Delta T_{max}\right) \tag{2438}$$

where:

ΔT_{max} is the maximum superheat applied to the **Lemmert Chawla** model.

STAR-CCM+ uses a modified form of the Lemmert Chawla equations. This modification allows that the superheat exponent does not have to be constant, but can itself increase significantly with wall superheat.

$$\frac{n''}{n_0''} = \left(\frac{\Delta T_{sup}}{\Delta T_0}\right)^{A + B\left(\Delta T / \Delta T_0\right)} \tag{2439}$$

where:

- A and B are calibration constants
- n_0'' is the reference nucleation site number density, this value is from (185.0/K)1.805.
- ΔT_0 is the reference wall superheat

You can adjust the Lemmert Chawla Method so that it is suitable for use with the Kocamustafaogullari bubble departure diameters by comparing it with the Hibiki-Ishii Method, at a given operating pressure, and for a useful range of wall superheats. For example, the table below shows a fit of the Lemmert Chawla method to the properties of water assuming a contact angle of 0.722 radians:

P (bar)	Superheat range (K)	n_0 (m-2)	ΔT_0 (K)	A	B	Fit Accuracy
45	1 - 10	138,100	1	1.348	0.35	7%
70	1 - 7	302,300	1	1.566	0.58	10%
150	1 - 3	5,665,000	1	3.095	1.7	0.1%

Using a fitted Lemmert Chawla method instead of using the Hibiki Ishii method gives similar results. The main advantage of such a fit is that it indicates the sensitivity of either method to wall superheat as an

explicit superheat exponent. For example, using the calibrations above, the table below shows that, even at 45 bar, the nucleation site number density can vary as strongly as $n'' = f\left(\Delta T^5\right)$:

P (bar)	Superheat range (K)	Exponent range
45	1 - 10	1.7 - 4.8
70	1 - 7	2.2 - 5.6
150	1 - 3	4.8 - 8.2

Hibiki Ishii

For high wall superheats, the Hibiki Ishii method can produce extreme values of wall heat fluxes. To improve robustness, a limit is applied to the amount of superheat that can be applied:

$$\Delta T_{sup} = \min\left(\max\left(T_{wall} - T_{sat}, 0\right), \Delta T_{max}\right) \tag{2440}$$

The default value for the maximum superheat ΔT_{max} is 25K. Do not allow this limit to affect the converged solution. If necessary, adjust the limit during the calculation.

Also, use system reference pressure in place of local liquid pressure to improve robustness.

$$P_l \approx P_{REF} \tag{2441}$$

Following Hibiki and Ishii [436], the temperature of the vapor next to the wall is identified with the wall temperature.

$$T_g \approx T_{sat} + \Delta T_{sup} \tag{2442}$$

For the contact angle between the working fluid and the surface during boiling, θ, Hibiki and Ishii used a value at room temperature. They suggest that a temperature dependence of contact angle is implicitly included in the calibration of the pressure effects function $\mathrm{f}\left(\rho^+\right)$.

The nucleation site number density is then modeled as:

$$n'' = \overline{n''}\left(1 - \exp\left[-\frac{\theta^2}{8\mu^2}\right]\right)\left(\exp\left[f\left(\rho^+\right)\frac{\lambda'}{R_c}\right] - 1\right) \tag{2443}$$

where:

- $\overline{n''}$ is the average cavity density, with a default value of 4.74E5 sites/m^{-2}
- θ is the wall contact angle, with a default value of 0.722 radians (41.37°), which is suitable for water systems
- μ is the wall contact angle scale, with a standard value of 0.722 radians (41.37°)
- λ' is the cavity length scale, with default value 2.50E-6 m
- $\mathrm{f}\left(\rho^+\right)$ is the logarithmic, non-dimensional density function representing the effect of pressure
- R_c is the critical cavity radius

The critical cavity radius is:

$$R_c = \frac{2\sigma\left[1 + \left(\rho_g/\rho_l\right)\right]/P_l}{\exp\left[h_{\lg}\left(T_g - T_{sat}\right)/\left(RT_gT_{sat}\right)\right] - 1} \tag{2444}$$

where:

- σ is the surface tension
- ρ_g is the vapor phase density
- ρ_l is the liquid phase density
- P_l is the liquid pressure
- h_{lg} is the latent heat
- R is the specific gas constant for a vapor of given molecular weight.

The logarithmic, non-dimensional density function is:

$$\mathrm{f}(\rho^+) = C_0 + C_1\rho^+ + C_2\rho^{+2} + C_3\rho^{+3} \tag{2445}$$

where:

$$\rho^+ = \log_{10}(\rho^*) \tag{2446}$$

and:

$$\rho^* = \frac{\rho_l - \rho_g}{\rho_g} \tag{2447}$$

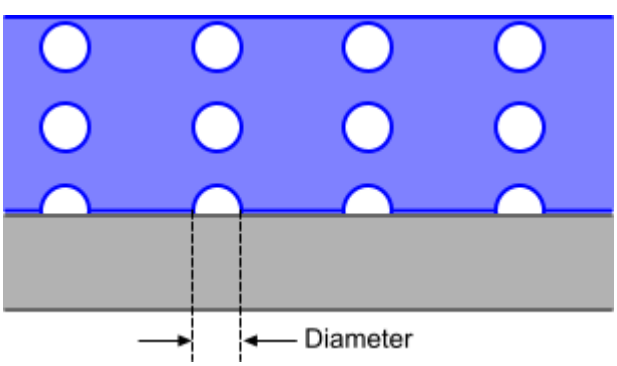
The standard values for the density function constants are:

$$\begin{aligned} C_0 &= -0.01064 \\ C_1 &= 0.48246 \\ C_2 &= -0.22712 \\ C_3 &= 0.05468 \end{aligned} \tag{2448}$$

Bubble Departure Diameter

The bubble departure diameter determines the diameter of the bubble at the instant it leaves the nucleation site.

This is the second of the three factors determining the evaporation rate in subcooled boiling (see Eqn. (2421)). The diagram below illustrates this parameter:



There are two standard methods that are implemented for calculating the bubble departure diameter:

- Tolubinsky Kostanchuk – Correlation against liquid subcooling.
- Kocamustafaogullari – Correlation that is based on bubble force balance at departure, and calibrated using steam-water data between 0.067 and 141.87 bar.

You can investigate various combinations of standard and your own submodels, but the following two combinations can both give reasonable estimates for evaporation rates:

- The Tolubinsky Kostanchuk Bubble Departure Diameter model that is used with the Lemmert Chawla Nucleation Site Number Density model.
- The Kocamustafaogullari Bubble Departure Diameter model that is used with the Hibiki Ishii Nucleation Site Number Density model.

The default option is to use Tolubinsky Kostanchuk for departure diameter, as appropriate for the default submodel for nucleation site number density.

Tolubinsky Kostanchuk Method

The Tolubinsky Kostanchuk model [502] correlates bubble departure diameter against liquid subcooling:

$$d_w = d_0 \exp\left[-\frac{\Delta T_{sub}}{\Delta T_0}\right] \tag{2449}$$

where:

- d_0 is the reference diameter with default value 0.0006 m
- ΔT_0 is the reference subcooling with default value 45
- ΔT_{sub} is the subcooling of the liquid next to the wall:

$$\Delta T_{sub} = T_{sat} - T_l \tag{2450}$$

Kocamustafaogullari Method

As noted in Hibiki and Ishii [436], it is usual to use a room temperature value for contact angle between the working fluid and surface. The actual temperature dependence is covered elsewhere in the overall model calibration.

The Kocamustafaogullari model [447] provides a correlation for departure diameter that is based on a force balance between gravity and surface tension, modified by a density ratio, which makes it useful for a wide range of pressures. This model is calculated as follows:

$$d_w = d_1 \theta \left(\frac{\sigma}{g\Delta\rho}\right)^{0.5} \left(\frac{\Delta\rho}{\rho_g}\right)^{0.9} \tag{2451}$$

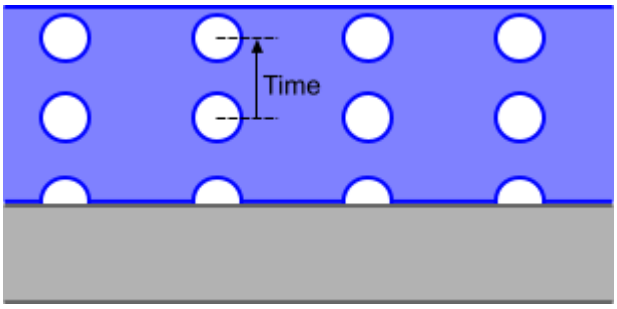
where:

- d_1 is a calibration constant, with a default value of 1.5126×10^{-3} m/radian (2.64×10^{-5} m/degree)
- θ is a wall contact angle for the working fluid, with a default value of 0.722 radians (41.36°), which is suitable for water systems
- σ is the surface tension
- g is the acceleration due to gravity
- $\Delta\rho$ is the difference in density between the liquid and gas phase
- ρ_g is the vapor phase density

Bubble Departure Frequency

The bubble departure frequency determines how many bubbles leave a nucleation site per second.

This is the last of three key factors determining the evaporation rate in subcooled boiling (see Eqn. (2421)). The diagram below illustrates this parameter:



Cole Model

The standard model that is implemented for bubble departure frequency is the Cole model [413]. This uses a typical bubble rise velocity (estimated using unit drag coefficient) as the velocity scale, and bubble diameter d_w as the length scale:

$$f = \sqrt{\frac{4}{3}\frac{g\left(\rho_l - \rho_g\right)}{d_w \rho_l}} \tag{2452}$$

where:

- g is acceleration due to gravity
- ρ_l is the liquid phase density

Bubble Induced Quenching Heat Transfer Coefficient

When a bubble leaves the heated surface, cooler liquid fills the space that it occupied. The heat transfer during this process is known as quenching heat transfer. The quenching heat Transfer coefficient is used to calculate the quenching heat flux.

Del Valle Kenning

When using the Del Valle Kenning model, the quenching heat transfer coefficient is:

$$h_{quench} = 2K_{quench} f \sqrt{\frac{\rho_l Cp_l k_l t_w}{\pi}} \tag{2453}$$

where:

- K_{quench} is the bubble influence wall area fraction given by Eqn. (2422)
- f is the bubble departure frequency
- ρ_l is the liquid density

- Cp_l liquid specific heat
- k_l is the liquid conductivity
- t_w is the time elapsed between bubble departure and the nucleation of the next bubble:

$$t_w = \frac{C_w}{f'} \tag{2454}$$

where:

C_w is the wait coefficient. The default value is 0.8. This value comes from an assumption by Kurul and Podowski [451] that quenching occurs between the departure of one bubble and the nucleation of next. This period is 80% of the departure cycle.

The bubble influence wall area fraction estimates the fraction of the wall area that is affected by the sweep of liquid inflow beneath a departing bubble. When an individual bubble departs from the wall with a diameter d_w, subcooled liquid flows in to fill the space underneath the detached bubble. The wall area influenced by this quenching flow is larger than the basic footprint of the bubble $\pi d_w^2/4$.

The standard model that is implemented for the bubble influence wall area fraction follows Kurul Podowski [451]:

$$K_{quench} = F_A \frac{\pi d_w^2}{4} n'' \tag{2455}$$

where:

- F_A is an area coefficient for scaling between the nucleation site area density and the wall area fraction the bubble-induced quenching influences.
- n'' is the nucleation site number density.

The default value for F_A is 2.0 and is chosen from experience with the main 45-bar test case in Bartolomei and Chanturiya [404].

Dissolution

The Dissolution Mass Transfer Rate model calculates the mass of a component that is transferred from one multi-component phase to another.

The rate of this mass transfer is assumed to be slow and independent of the rate of transfers between any other species. Diffusion driven by the concentration gradient is assumed to be the mechanism that moves species between the phase bulk region and the phase boundary. This movement can occur in either direction.

Gas-Liquid Interactions

For gas-liquid interactions, the equilibrium between species at the interface is defined as:

$$X^*_{gas,i} = K_i^0 X^*_{liq,i} \tag{2456}$$

where $X^*_{gas,i}$ and $X^*_{liq,i}$ are the equilibrium mole fractions of component i at the gas side and liquid side of the interface respectively, and K_i^0 is the equilibrium constant. With this equilibrium definition, the value of K_i^0 is independent of the choice of the continuous and dispersed phase.

The equilibrium constant can be specified using the following techniques:

Raoult's Law

Raoult's law states that the equilibrium partial pressure of a component above the surface of a liquid is equal to the vapor pressure of a pure sample of the component multiplied by the mole fraction of the component in the liquid. Hence Raoult's law gives an equilibrium constant:

$$K_i^0 = \frac{p_{vap}}{p_{abs}} \tag{2457}$$

Raoult's Law is appropriate to volatile liquid species evaporating into or out of a gas.

Henry's Law

For cases where only small amounts of a species are able to dissolve into a liquid, for instance when a gas such as oxygen dissolves in trace amounts into water, Raoult's law no longer holds. In this case Henry's law can be used. Henry's law states that the equilibrium amount of a gas species that dissolves in a liquid is directly proportional to the partial pressure of that species in the gas:

$$K_i^0 = \frac{H_i^0}{p_{abs}} \tag{2458}$$

where H_i^0 is Henry's coefficient and is unique to any gas and liquid solvent pair. The default value of 4.2E9 Pa is appropriate for oxygen dissolving in water at room temperature.

The rate of transfer of the species from one phase to the other is governed by the rate at which the species can diffuse between the phase boundary and the interior of the phase.

For gas-liquid interactions, the species mass balance at the interface is:

$$\dot{m}_{l,i} + \dot{m}_{g,i} = 0 \tag{2459}$$

$$\dot{m}_{l,i} = k_l a_{lg} \rho_l \left(Y^*_{l,i} - Y_{l,i} \right) \tag{2460}$$

$$\dot{m}_{g,i} = k_g a_{lg} \rho_g \left(Y^*_{g,i} - Y_{g,i} \right) \tag{2461}$$

where:

- The subscript l refers to the liquid phase and g to the gas
- k is a mass transfer coefficient for either phase
- a_{lg} is the interfacial area density
- ρ is the phase density
- $Y_{l,i}$ and $Y_{g,i}$ are the mass fractions of the two phases
- Y^* refers to the equilibrium mass fraction at the interface of the two phases.

Liquid-Liquid Interactions

For liquid-liquid interactions, the equilibrium between species at the interface is defined as:

$$X^*_{cont,i} = K_i^0 X^*_{disp,i} \tag{2462}$$

Where $X^*_{cont,i}$ and $X^*_{disp,i}$ are the continuous phase and dispersed phase equilibrium mole fractions at the interface respectively. In this case, the value of the equilibrium constant K_i^0 depends on the choice of the continuous and dispersed phase. In particular, if the continuous and disperse phases are swapped, the value of K_i^0 (either constant or field function) becomes $1/K_i^0$.

The equilibrium constant can be specified using the following techniques

- Constant value
- Field function

For liquid-liquid interactions, the species mass balance at the interface is:

$$\dot{m}_{c,i} + \dot{m}_{d,i} = 0 \tag{2463}$$

$$\dot{m}_{c,i} = k_c a_{cd} \rho_c \left(Y^*_{c,i} - Y_{c,i}\right) \tag{2464}$$

$$\dot{m}_{d,i} = k_d a_{cd} \rho_d \left(Y^*_{d,i} - Y_{d,i}\right) \tag{2465}$$

where the subscript c refers to the continuous phase and d to the dispersed phase. The meaning of the symbols is the same as in the liquid-gas interaction case above.

Size Distribution

In order to account for a size distribution of the dispersed phase, the mass and momentum transport equations have to be combined with a population balance equation (PBE). The central object of a PBE is the particle number density $n\left(d_p\right)$, which gives the number of particles (droplets or bubbles) whose diameters range from d_p to $d_p + d\left(d_p\right)$.

In its general form, the PBE reads:

$$\frac{\partial n\left(d_p\right)}{\partial t} + \nabla \cdot \left(\mathbf{v} n\left(d_p\right)\right) = B - D \tag{2466}$$

where B and D are *birth* and *death* terms due to beakup, coalescence, nucleation, dissolution, and so on [480].

The two most popular numerical methods for solving the PBE are the method of moments and the method of classes. Instead of solving for the number density, the method of moments solves for a (usually small) number of its moments:

$$S_\gamma = \int d_p^\gamma n\left(d_p\right) d\left(d_p\right) \tag{2467}$$

In STAR-CCM+, the method of moments is implemented under the name S-Gamma method. This method uses only one or two moments of the size distribution. The S-Gamma method is a computationally efficient tools that is suitable for engineering estimates.

The method of classes implies that the polydisperse phase is split into a number of size classes. Each class has its own mass and momentum balance. The standard method of classes, which is also known as multiple size-group (MUSIG) method [459], prescribes the sizes of the size-groups a priory. The sizes are constant throughout the computational domain, which makes the method time-consuming. In STAR-CCM+, the partitioning of the particles ensemble into size classes is adaptive in space and time [510], therefore the name adaptive multiple size-group (AMUSIG) method. The AMUSIG method lets you work with a smaller number of size-groups than other class methods.

Contents:

S-Gamma
Adaptive Multiple Size-Group (AMUSIG)

S-Gamma

The S-Gamma model is a method of moments that solves for the zeroth and the second moments of the number density $n(d_p)$. One transport equation is solved for each moment. Particles change size due to breakup and coalescence as they interact with each other.

The S-Gamma model (see Lo and Rao [460] and Lo and Zhang [462]) assumes that the particle size distribution conforms to a pre-defined log-normal shape function. The log-normal particle size distribution (probability density function) is represented by the mean particle diameter and its variance:

$$P(d_p) = \frac{1}{d_p \sigma \sqrt{2\pi}} \exp\left(\frac{(\ln d_p - \mu)^2}{2\sigma^2}\right) \tag{2468}$$

where:

- μ is the mean logarithm of the particle diameter
- σ is the standard deviation of the particle diameter

The S-Gamma model defines a volumetric conserved quantity S_γ that is related to the moment M_γ of the particle size distribution:

$$S_\gamma = nM_\gamma = n\int_0^\infty d_p^\gamma P(d_p)\, d(d_p) \tag{2469}$$

where:

- $\gamma = 0, 1, 2, 3$ is the order of the moment
- n is the number of particles per unit volume.
- d_p is the particle diameter
- $P(d)$ is the probability density function of particle diameter [460], [462].

STAR-CCM+ solves a transport equation for each of the three moments of $P(d)$:

- Zeroth moment: the particle number density n
- Second moment: related to the interfacial area density
- Third moment: related to the dispersed phase volume fraction

The zeroth moment is given by:

$$S_0 = \int n(d_p) d(d_p) \tag{2470}$$

The second moment reads:

$$S_2 = \int d_p^2 n(d_p) d(d_p) \tag{2471}$$

The choice of moments is dictated by physical reasons. The zeroth moment S_0 corresponds to the total number of particles. Assuming that the particles are spherical, the interfacial area can be calculated as:

$$A = \pi \int d_p^2 n(d_p)\, d(d_p) = \pi S_2 \tag{2472}$$

The volume fraction of the dispersed phase can be calculated as:

$$\alpha = \frac{\pi}{6} \int d_p^3 n(d_p) d(d_p) = \frac{\pi}{6} S_3 \tag{2473}$$

As the third moment is based on the dispersed phase volume fraction, it is not calculated by the S-Gamma method, but it is imported from the multiphase segregated flow solver. The multiphase segregated flow solver solves for the volume fraction of the dispersed phase.

Contents:

S-Gamma Breakup and Coalescence
Moments and Mean Particle Diameters
Transport Equations

S-Gamma Breakup and Coalescence

The S-Gamma equations (for S_0 and S_2) are solved with source terms added to model coalescence and breakup of bubbles and other particle types.

Both these processes conserve the total volume of the dispersed phase. No extra term is required in the volume fraction equation which is used together with S_0 and S_2 to give the mean diameter.

Breakup

The source terms for breakup are found by integrating the effect of a bubble or droplet of size d breaking up over all diameters. The general form for the breakup is from Lo and Zhang [462]:

$$\frac{d}{dt} S_\gamma = \int_{d_{lo}}^{d_{hi}} nP(d) \frac{\left(N(d)^{1-\gamma/3} - 1\right)}{\tau(d)} d^\gamma d(d) \tag{2474}$$

Where:

- N is the number of fragments, or child bubbles, produced when a bubble breaks down.
- τ is the timescale for breakup.
- d_{lo} and d_{hi} are the upper and lower limits of a particular breakup regime.

The implementation in STAR-CCM+ considers both viscous and inertial breakup regimes. For laminar flow, the lower limit for viscous breakup is the critical diameter d_{cr} and the upper limit is infinite. For turbulent flow, the upper limit is the Kolmogorov length scale L_k; if d_{cr} is greater than L_k then viscous breakup plays no role. For inertial breakup the integral is performed from the greater of L_k and d_{cr} to infinity. By writing $N(d)$ and $\tau(d)$ in the form:

$$N(d) = \mathrm{N}' d^b \tag{2475}$$

$$\tau(d) = \tau' d^a \tag{2476}$$

The integral can be written in the form:

$$\frac{d}{dt} S_\gamma = \frac{\mathrm{N}'^{1-\gamma/3}}{\tau'} \int_{\mathrm{d}_{lo}}^{\mathrm{d}_{hi}} nP(d) d^{\delta_1} d(d) - \frac{1}{\tau'} \int_{\mathrm{d}_{lo}}^{\mathrm{d}_{hi}} nP(d) d^{\delta_2} d(d) \tag{2477}$$

Where $\delta_1 = \gamma + b - b\gamma/3 - a$ and $\delta_2 = \gamma - a$. This form of the equation is from Hill [437].

Viscous Breakup

For viscous breakup, the critical diameter is given as [462], [437]:

$$d_{cr} = \frac{2\sigma\Omega_{cr}}{\mu_c \dot{\gamma}} \tag{2478}$$

Where:

- μ_c is the viscosity of the continuous phase.
- σ is the surface tension.
- $\dot{\gamma}$ is the local shear rate.
- Ω_{cr} is the critical capillary number [437].

The fragmentation number is assumed to be binary [462], that is:

$$N(d) = 2 \tag{2479}$$

The breakup timescale is from [462]:

$$\tau(d) = \frac{\mu_c d}{\sigma} f(\lambda) \tag{2480}$$

Where:

- μ_c is the continuous phase dynamic viscosity.

- σ is the surface tension.
- $\log f(\lambda) = p_0 + p_1 \log(\lambda) + p_2 (\log(\lambda))^2$ and $\lambda = \mu_d / \mu_c$ and the constants are given in [437].

Inertial Breakup

For inertial breakup, the critical bubble diameter and related values are calculated as follows [462], [437]:

$$d_{cr} = \left(1 + C_\alpha \alpha_d\right)\left(\frac{2\sigma We_{cr}}{\rho_c}\right)^{3/5} \varepsilon^{-2/5} \tag{2481}$$

$$\tau(d) = 2\pi k_{br}\left(\frac{3\rho_d + 2\rho_c}{192\sigma}\right)^{1/2} d^{3/2} \tag{2482}$$

$$N(d) = 2 \tag{2483}$$

Where:

- ρ_d and ρ_c are the dispersed and continuous phase densities.
- ε is the continuous phase turbulent dissipation rate.
- We_{cr} is the critical Weber number which you define with a default value of 0.25.
- C_α and k_{br} are constants with default values of 4.6 and 0.2, respectively.

Coalescence

Performing the double integral required for an exact evaluation of the coalescence term is not practicable. A simplified source term is developed based on an equivalent diameter d_{eq} for each moment, which here is taken to be equal to $d_{3\gamma}$.

The source term that is given in [462] is:

$$\frac{d}{dt} S_\gamma = F_{cl}\left(2^{\gamma/3} - 2\right)\left(\frac{6\alpha}{\pi}\right)^2 k_{coll} u_{rel} P_{cl}\left(d_{eq}\right) d_{eq}^{\gamma-4} \tag{2484}$$

Where:

- F_{cl} is the calibration coefficient.
- k_{coll} is a collision rate.
- u_{rel} is the typical phase velocity difference.
- $P_{cl}\left(d_{eq}\right)$ is the probability of collision leading to coalescence.

To split the source term into a collision rate (in collisions per unit volume per unit time) and a coalescence probability, either of which can be replaced with functions you define. This formulation has been rearranged slightly to give:

$$\frac{\partial}{\partial t} S_\gamma = F_{cl}\left(2^{\gamma/3} - 2\right) K_{coll} P_{cl}\left(d_{eq}\right) d_{eq}^{\gamma} \tag{2485}$$

Where K_{coll} is the collision rate and is:

$$K_{coll} = k_{coll} u_{rel} d_{eq}^2 \left(\frac{6\alpha}{\pi d_{eq}^3} \right)^2 \tag{2486}$$

For a turbulent flow, viscous and inertial effects lead to coalescence and a source term of the form that is shown above is evaluated for each. By successively setting γ to 0 and 2, the source terms for the zeroth and second moment S-Gamma transport equations are obtained.

Viscous Coalescence

For viscous coalescence, the collision rate is [462]:

$$K_{coll,v} = \left(\frac{8\pi}{3} \right)^{1/2} \left(\dot{\gamma} d_{eq} \right) d_{eq}^2 \left(\frac{6\alpha}{\pi d_{eq}^3} \right)^2 \tag{2487}$$

With $\dot{\gamma}$ being the shear rate. For a laminar flow, the shear rate is calculated using the velocity field. For a turbulent field, the shear rate is $\dot{\gamma} = \sqrt{\varepsilon \rho_c / \mu_c}$ [460] which uses the turbulent dissipation rate, density, and viscosity from the continuous phase. The probability is from [462]:

$$P_v\left(d_{eq}\right) = \exp\left(-t_d \dot{\gamma}\right) \tag{2488}$$

Physically, this is comparing the time that the bubbles interact with the time required for the film initially to separate and then drain away. Four models for drainage modes are from [460]. Starting from the model that gives the slowest drainage time (and hence the lowest coalescence rate) toward the fastest, these models are called:

- Fully Immobile:

$$t_d = \frac{3\mu_c d_{eq}^2 F_i}{64\pi\sigma^2 h_{cr}^2} \tag{2489}$$

- Partially Mobile Interface (Short Collision time):

$$t_d = \frac{3}{2} \left(\frac{F_i d_{eq}^2 \sqrt{\mu_d \rho_d}}{32\pi\sigma^2 h_{cr}} \right) \tag{2490}$$

- Partially Mobile Interface (Quasi-Steady state):

$$t_d = \frac{\pi \mu_d \sqrt{F_i}}{2h_{cr}} \left(\frac{d_{eq}}{4\pi\sigma} \right)^{3/2} \tag{2491}$$

- Fully Mobile:

$$t_d = \frac{3\mu_c d_{eq}}{4\sigma} \ln\left(\frac{d_{eq}}{8h_{cr}} \right) \tag{2492}$$

Where $F_i = (3/2)\pi\mu_c \dot{\gamma} d_{eq}^2$.

The critical film thickness is:

$$h_{cr} = \left(\frac{A_H d_{eq}}{24\pi\sigma}\right)^{1/3} \tag{2493}$$

Where:

- A_H is the Hamaker constant and its default value is 5×10^{-21}.
- σ is the surface tension.

Reference [462] provides all equations except the fully immobile expression.

Inertial Coalescence

For inertial coalescence, the collision rate and probability are from [462]:

$$K_{coll,i} = \left(\frac{2\pi}{15}\right)^{1/2}\left(\varepsilon d_{eq}\right)^{1/3} d_{eq}^2 \left(\frac{6\alpha}{\pi d_{eq}^3}\right)^2 \tag{2494}$$

$$P\left(d_{eq}\right) = \frac{\Phi_{max}}{\pi}\left(1 - \frac{k_{cl,2}^2\left(We - We_0\right)^2}{\Phi_{max}^2}\right)^{1/2} \tag{2495}$$

Φ_{max}, the maximum phase difference is:

$$\Phi_{max} = \left(2h_0^2\rho_c\sigma\right) / \left(We_0\mu_d^2 d_{eq}\right) \tag{2496}$$

Also:

$$We_0 = 0.8We_{cr} \tag{2497}$$

$$h_0 = 8.3h_{cr} \tag{2498}$$

You can set We_{cr} and $k_{cl,2}$, the latter having a default value of 12.7.

Moments and Mean Particle Diameters

Knowing the zeroth moment S_0, the second moment S_2, and the volume fraction α, and assuming a log-normal distribution , all other parameters of the distribution can be calculated. The relations below can be helpful in characterizing the size distribution.

The particle size distribution moments and the particle diameters are related through the following expressions:

- **Sauter Mean Diameter**:

$$d_{32} = \frac{6\alpha}{\pi S_2} \tag{2499}$$

- **Three-zero diameter**:

$$d_{30} = \sqrt[3]{\frac{6\alpha}{\pi S_0}} \tag{2500}$$

$$d_{10} = \frac{S_1}{S_0} = \exp\left(\mu + \frac{\sigma^2}{2}\right), \quad d_{20} = \frac{S_2}{S_0} = \exp\left(2\left(\mu + \sigma^2\right)\right) \tag{2501}$$

$$\alpha = \frac{\pi}{6} S_3 = S_0 \frac{\pi}{6} \exp\left(3\mu + \frac{9\sigma^2}{2}\right) \tag{2502}$$

$$\mu = \frac{3}{2}\ln\left(\frac{S_2}{S_0}\right) - \frac{2}{3}\ln\left(\frac{6\alpha}{\pi S_0}\right) = \frac{1}{2}\ln\frac{\left(d_{30}\right)^5}{\left(d_{32}\right)^3} \tag{2503}$$

$$\sigma^2 = \frac{2}{3}\ln\left(\frac{6\alpha}{\pi S_0}\right) - \ln\left(\frac{S_2}{S_0}\right) = \ln\frac{d_{32}}{d_{30}} \tag{2504}$$

$$\begin{gathered} d_{30} = \sqrt[3]{\frac{6\alpha}{\pi S_0}} \\ d_{10} = \frac{S_2}{S_0}\exp\left(-\left(\mu + \frac{3}{2}\sigma^2\right)\right) = \frac{\left(d_{30}\right)^2}{d_{32}} \\ d_{32} = \frac{6\alpha}{\pi S_2} \end{gathered} \tag{2505}$$

The realizability condition $\sigma^2 \geq 0$ is equivalent to $d_{32} \geq d_{30}$ and implies that:

$$\frac{S_2}{S_0} \leq \left(\frac{6\alpha}{\pi S_0}\right)^{2/3} \tag{2506}$$

Transport Equations

The transport equation for volume fraction (and hence S_3) is solved by the multiphase segregated flow solver and not by the S-Gamma method. If you are not interested in the full details of the size distribution, you can solve for S_2 only. This is called one-equation mode. In two-equation mode, the equations for S_2 and S_0 are both solved.

Variable density does not play a role in the conservation of the number density of particles.

Transport Equation for the Zeroth Moment (Particle Number Density)

The instantaneous transport equation for S_0 is given by [460], [462]:

$$\frac{\partial S_0}{\partial t} + \nabla \cdot \left(S_0 \mathbf{v}_d\right) = s_{br} + s_{cl} + s_{wb} \tag{2507}$$

where:

- $\mathbf{v}_d$ is the dispersed phase velocity that is calculated by the multiphase velocity solver
- s_{br} is a source term that represents the effect of breakup

- s_{cl} is a source term that represents the effect of coalescence
- s_{wb} is a source term that accounts for bubbles that are created on walls. Applies only at wall boundaries when wall boiling is active.

s_{wb} is given as:

$$s_{wb} = \frac{dS_0}{dt} = \dot{n}A \tag{2508}$$

where:

- A is the wall area of a cell
- $\dot{n}$ is the local bubble creation rate

Reynolds averaging of Eqn. (2507) yields:

$$\frac{\partial \overline{S}_0}{\partial t} + \nabla \cdot \left(\langle S_0 \mathbf{v}_d \rangle\right) = \langle s_{br} \rangle + \langle s_{cl} \rangle + \langle s_{wb} \rangle \tag{2509}$$

For any scalar ϕ (for example, temperature, concentration, kinetic energy of turbulence), the Reynolds-averaged flux is modeled as:

$$\langle \alpha \rho \phi \mathbf{v} \rangle = \overline{\alpha} \rho \langle \mathbf{v} \rangle \overline{\phi} + \left\langle \alpha \rho \phi' \mathbf{v}' \right\rangle \approx \overline{\alpha} \rho \langle \mathbf{v} \rangle \langle \phi \rangle - \overline{\alpha} \rho D_T \nabla \overline{\phi} \tag{2510}$$

where D_T is the coefficient of turbulent diffusion and α is the volume fraction. This value is calculated from the turbulent viscosity, density, and the turbulent Prandtl number as:

$$D_T = \frac{\nu_T}{\rho \mathrm{Pr}} \tag{2511}$$

Rewriting S_0 as $\alpha\rho\frac{S_0}{\alpha\rho}$, the Reynolds-averaged transport equation for S_0 is obtained as:

$$\frac{\partial \overline{S}_0}{\partial t} + \nabla \cdot \left(\overline{S}_0 \langle \mathbf{v}_d \rangle - \rho \overline{\alpha}_d D_T \nabla \frac{\overline{S}_0}{\rho R A \alpha_d}\right) = \langle s_{br} \rangle + \langle s_{cl} \rangle + \langle s_{wb} \rangle \tag{2512}$$

Transport Equation for the Second Moment

For S_2, Wei and Morel [513] give an equation for the interfacial area a_d of the dispersed phase in the form:

$$\frac{\partial a_d}{\partial t} + \nabla \cdot \left(a_d \mathbf{v}_d\right) = s_{a,br} + s_{a,cl} + s_{a,m} - \frac{2a_d}{3\rho}\left(\frac{\partial \rho}{\partial t} + \mathbf{v}_d \nabla \rho\right) \tag{2513}$$

$s_{a,br}$ and $s_{a,cl}$ represent the source of area due to breakup and coalescence respectively, and $s_{a,m}$ represents the source of area due to mass transfer between phases. The final term on the right-hand side represents the effect of variable density on interfacial area. By multiplying through by $\rho^{2/3}$, performing some algebraic manipulations, and finally dividing by π to convert to S_2, it can be shown that the transport equation for S_2 is:

$$\frac{\partial \rho^{2/3} S_2}{\partial t} + \nabla \cdot \left(\rho^{2/3} S_2 \mathbf{v}_d\right) = s_{br} + s_{cl} + s_m \tag{2514}$$

Therefore, the Reynolds-averaged version of the transport equation is:

$$\frac{\partial \rho^{2/3}\overline{S}_2}{\partial t} + \nabla \cdot \left(\rho^{2/3}\overline{S}_2 \langle \mathbf{v}_d \rangle - \rho \overline{\alpha}_d D_T \nabla \frac{\overline{S}_2}{\rho^{2/3}\overline{\alpha}_d} \right) = \langle s_{br} \rangle + \langle s_{cl} \rangle + \langle s_{wb} \rangle \tag{2515}$$

The source term for S-Gamma due to the mass transfer reads:

$$\dot{S}_2 = 4G\sum n_i d_i = 4GS_0 d_{10} \tag{2516}$$

But $\dot{M} = \rho \pi S_2 G$, and therefore:

$$G = \frac{\dot{M}}{\rho \pi S_2} \tag{2517}$$

where:

- $\dot{M}$ is the interphase mass transfer rate
- G is the interphase mass flux

Combining these two formulae with d_{10}, we obtain:

$$\dot{S}_2 = \frac{4\dot{M}}{\pi \rho} \left(\frac{\pi S_0}{6\alpha} \right)^{1/3} \tag{2518}$$

Adaptive Multiple Size-Group (AMUSIG)

The adaptive multiple size-group (AMUSIG) model predicts the size distribution of droplets or bubbles in a dispersed flow regime of a multiphase flow.

The collection of droplets or bubbles is modeled using a number of groups of representative computational particles. Each group consists of identical particles, has its own size, and optionally can have its own velocity. A population balance is solved for each size-group in each cell. Breakup and coalescence models provide the principal contributions to the birth and death source terms for each balance. Other relevant effects include particle diffusion, acceleration, separation, and phase density changes. The solution to the balance equation gives the local particle number density for the group. The particle diameter is computed from the number density and the volume fraction for the group.

The method is locally adaptive so that each size-group carries approximately the same share of the phase mass or volume. This method is efficient and benefits fully from the computational expense of multi-speed modeling. Such modeling is necessary when analysing strong interactions between the size distribution and the flow. However, adaptation also means that the particle size that is associated with a group can vary gradually from cell to cell. Take care when you are comparing or integrating group results from different parts of the domain.

Multi-Speed AMUSIG

You use multi-speed AMUSIG for modeling flow-size interactions in full. Mass and momentum equations are solved individually for each particle size-group. This option is more expensive, but is justified where there is strong interaction between the flow and the local size distribution. For example, this option is relevant for flows that separate particles by size. It is also relevant for flows where the interaction of buoyancy, turbulence, coalescence, and breakup cause different secondary flow patterns for different particle sizes, and therefore further promote overall turbulence and mixing.

In order to account for the multidisperse nature of the flow, the dispersed phase is split into M size-groups. From the modeling point of view, each group can be considered a separate phase in every aspect but the name. Each group moves with its own velocity and exchanges mass, momentum, and energy with the other groups and with the continuous phase [459].

The Reynolds-averaged mass conservation equation for the ith group reads:

$$\frac{\partial \rho_p \overline{\alpha}_i}{\partial t} + \nabla \cdot \left(\rho_p \overline{\alpha}_i \langle \mathbf{v}_i \rangle\right) = m_{ij} - m_{ji} \tag{2519}$$

where:

- ρ_p is the density of the dispersed phase.
- $\overline{\alpha}_i$ is the Reynolds-averaged volume fraction of the ith group.
- m_{ij} and m_{ji} are the (averaged) mass fluxes from the jth group to the ith group, and from the ith group to the jth group, respectively.
- $\langle \mathbf{v}_i \rangle$ is the phase-averaged velocity of the group [430]:

$$\left\langle \mathbf{v}_i \right\rangle = \frac{\langle \alpha_i v_i \rangle}{\overline{\alpha}_i} \tag{2520}$$

Where:

- α_i is the instantaneous value of volume fraction.
- v_i is the instantaneous value of velocity.
- Angular brackets mean Reynolds averaging.

The Reynolds-averaged momentum conservation equation for the ith group reads:

$$\frac{\partial \rho_p \overline{\alpha}_i \langle \mathbf{v}_i \rangle}{\partial t} + \nabla \cdot \left(\rho_p \overline{\alpha}_i \langle \mathbf{v}_i \rangle \langle \mathbf{v}_i \rangle\right) = -\overline{\alpha}_i \nabla P - \nabla \cdot \tau_i + \left\langle \mathbf{F}_i \right\rangle + m_{ij} \left\langle \mathbf{v}_j \right\rangle - m_{ji} \left\langle \mathbf{v}_i \right\rangle \tag{2521}$$

Where:

- τ_i is the Reynolds stress.
- $\mathbf{F}_i$ is the interaction force between the continuous phase and the ith group.

In order to calculate the phase-interaction forces, the size of the particles in the ith group has to be specified. When a constant size for the group is prescribed, the MUSIG method is obtained [459]. If the particle size distribution varies significantly across the system, the fixed discretization in the size space is not efficient from the numerical point of view. To track the size distribution adaptively, Eqn. (2519) and Eqn. (2521) are augmented by an equation for the number density of the ith group.

Having the mean volume fraction and number density, the volume of the particles in the ith group can be postulated as:

$$\overline{v}_i = \frac{\overline{\alpha}_i}{\overline{n}_i} \tag{2522}$$

The Reynolds averaged equation for the number density reads:

$$\frac{\partial \overline{n}_i}{\partial t} + \nabla \cdot \overline{n}_i \{\langle \mathbf{v}_i \rangle + D_T \nabla (\ln \overline{\alpha}_i - \ln \overline{n}_i)\} = \langle S_i \rangle \tag{2523}$$

Where:

- $\langle S_i \rangle$ is a source term that is provided by the AMUSIG model.
- D_T is the number density diffusion coefficient.

  This value is calculated from the kinematic turbulent viscosity and the Number Density Turbulent Prandtl number, as:

  $D_T = \nu_t / \sigma_n$.

If the volume of the group $(\overline{\alpha}_i / \overline{n}_i)$ is constant, the term $\nabla (\ln \overline{\alpha}_i - \ln \overline{n}_i) = 0$, and so the diffusive velocity in Eqn. (2523) vanishes.

Single-Speed AMUSIG

You use single-speed AMUSIG for modeling flow-size interactions in the mean. Mass and momentum equations are solved at the phase level, and all of the particle size groups are assumed to be convected at phase velocity. This option allows STAR-CCM+ to model the effect of the flow and turbulence on the size distribution. The predicted mean particle size is passed back to the flow calculation.

The single-speed AMUSIG model is a simplified version of the multi-speed model. This model assumes that all size-groups are moving with the same velocity. The mass and momentum equations are solved at the phase level and then translated to the size-group level as:

$$\begin{aligned} \overline{\alpha}_i &= \frac{\overline{\alpha}_{phase}}{M} \\ \mathbf{v}_i &= \mathbf{v}_{phase} \end{aligned} \tag{2524}$$

The only equation that is solved at the size-group level is Eqn. (2523).

Population Balance Algorithm

In order to close the model that is described by Eqn. (2519), Eqn. (2521), Eqn. (2522), and Eqn. (2523) the AMUSIG population balance algorithm specifies m_{ij}, m_{ji} and $\langle S_i \rangle$.

There are five physical and numerical factors that contribute to m_{ij}, m_{ji} and $\langle S_i \rangle$:

- **Breakup**

  The probability of a particle breaking up is given by the corresponding Breakage Rate. When particle breakup occurs, the fragments are distributed according to the Daughter Particles Size distribution. The mass and number density of the fragments are distributed among the other size groups. However, unlike the method of classes, breakup of the smallest size-group is allowed.
- **Coalescence**

  The probability of two particles colliding is given by the corresponding Collision Rate. When two particles collide, the probability that the particles coalesce is given by the Coalescence Efficiency. The mass and number density of the resulting particle are distributed among other size-groups. Once again, due to the

adaptive nature of the method, the biggest size-group can coalesce. Coalescence between particles within the same size-group is also allowed.

- **Inlet**

  The AMUSIG model requires the inlet boundary conditions to be specified at the phase level. This specification is required because the sizes of the particles at an inlet are not necessarily equal to the sizes of the particles in the nearby cells. The fresh particles that are coming through the inlet are mixing with the particles that are already present in the stream. The mixing is performed using the same procedure as coalescence and breakup: the mass and number density of the incoming particles are distributed among the existing size-groups.

- **Redistribution**

  Redistribution is a counter-intuitive term that has no physical counterpart, but plays a central role in the AMUSIG method. When there is a strong breakup, you would intuitively assume that the volume fraction of the large size-groups would decrease, while the smallest size-groups would gain more mass. However, if you plot the volume fraction of the different size-groups, you can see that they are nearly equal in each cell. Instead, the diameters of all the size-groups decrease: this reduction is due to the redistribution. The redistribution algorithm does not require any user input, but its effect is apparent and has a strong impact on the performance of the AMUSIG method.

- **Number density diffusion**

  It is known that the diffusive flux in Eqn. (2523) produces so-called spurious dissipation proportional to $D_T\left|\nabla\overline{v}_i\right|^2$ ([429]). The spurious dissipation implies that the predicted size distribution is less broad than it should be — the standard deviation of the size distribution is "dissipated" by the turbulent diffusion. In order to avoid the spurious dissipation, the AMUSIG model has a special treatment to the number density diffusion term.

  However, this term sometimes leads to instability, so the Size Distribution Diffusion Limiter (λ_D) property is available to experienced users. For more information, see Adaptive Multiple Size-Group Model Properties.

Contents:

AMUSIG Breakup and Coalescence
Mean Particle Diameters
Initial and Boundary Particle Size Distributions

AMUSIG Breakup and Coalescence

For laminar flows, coalescence and breakup are driven by the shear rate of the continuous phase. For turbulent flows, coalescence and breakup are driven by the turbulence scales of the continuous phase. The breakup and coalescence rates have some sensitivity to the turbulence models that are selected for the continuous phase and for particle-induced turbulence. To obtain comparable results with different turbulence models, you may need to recalibrate the models.

The STAR-CCM+ breakup and coalescence models for AMUSIG are not suitable for particulate flow. However, you can use AMUSIG to model the size distribution of solid particle phases in a particulate flow: use your own understanding and knowledge of coalescence and breakup phenomena for the application of interest to calibrate power law coalescence and breakup models as appropriate.

A set of models is available for describing breakup and coalescence processes for different applications:

- The model of Luo ([466]) is provided for turbulent coalescence.
- The model of Tsouris and Tavlarides ([467]) and the model of Martinez-Bazan ([467]) are provided for break-up in turbulent liquid/liquid and bubbly flows.
- Shear-induced breakup and coalescence models are provided for laminar flows.
- Constant and power-law models are also implemented for testing and demonstration purposes, and to provide some user-specified modeling.

The Coalescence Rate model has submodels for Collision Rate and Coalescence Efficiency. The Breakup Rate model has submodels for Breakage Rate and Fragments Particle Size Distribution, which defines a removal term for particles of a given size. All of the rate models have an overall calibration constant which you can adjust to suit your requirements. The Critical Weber number is an adjustable parameter that is relevant for the breakup rate models.

A single size-group is sufficient for modeling hydrostatic expansion of dilute bubbles rising in a deep tank or column. However, for a meaningful collision/breakup study, a minimum of three groups is recommended. The adaptivity of the AMUSIG model allows you to investigate significant and rapid changes of particle size with a few groups. Increasing the number of groups should lead to resolution-independent results, in a manner similar to mesh refinement. Computational cost sets the practical upper limit on the number of size-groups.

Laminar and Turbulent Breakup Models

The breakup process is modeled in two parts, a breakage rate and a fragments particle size distribution. The breakup rate has a dimension of *time*$^{-1}$, so the probability that a single particle is broken within the time span dt is (breakup rate) dt. The fragment size distribution is dimensionless. It determines how many fragments are created, and the distribution of the fragments.

Power Law Breakage Rate

This model applies to both turbulent breakup and laminar breakup. The power law breakage rate model describes how quickly a particle of a given size can be broken. This model is a generic model with adjustable parameters for the breakage rate multiplier K_B of number density at some particle size d scaled by a characteristic diameter d_0.

$$K_B(d) = C\left(\frac{d}{d_0}\right)^a \tag{2525}$$

Since the product of K_B and number density is a rate in events per second per m^3, the calibration constant C has dimensions of per second.

This breakage rate enters the conservation equations as any other breakage rate:

$$\frac{Dn}{Dt} = -\left(\text{breakage rate}\right) * n \tag{2526}$$

There are three turbulent breakup models available, Tsoures and Tavlarides, Martines-Bazan ([467]), and Coulaloglou and Eskin. All three models have a similar structure for the breakup rate:

$$\text{Breakup Rate} = C_g \frac{\left(\varepsilon_f d_p\right)^{1/3}}{d_p} f\left(\text{We}_{\text{cr}}/\text{We}\right) \tag{2527}$$

where:

- ε_f is the turbulent dissipation rate of the continuous phase.
- d_p is the diameter of the particle.
- We is the Weber number.
- $\mathrm{We}_{\mathrm{cr}}$ is the critical Weber number. It determines the size of the particle that can survive the given intensity of turbulence. A high value of $\mathrm{We}_{\mathrm{cr}}$ implies that the breakup is weak.
- C_g is a calibration constant that can be used to adjust the breakup time scale.

All three models assume that a high Weber number implies a higher breakup rate. The Weber number is the ratio of the dynamic pressure that tries to deform and to eventually destroy the bubble or droplet, and the capillary pressure that stabilizes the bubble and preserves its spherical shape.

For turbulent flows, the Weber number is given by:

$$\mathrm{We} = \frac{\rho_f \left(\epsilon_f d_p\right)^{2/3} d_p}{\sigma} \left(\frac{\rho_p}{\rho_f}\right)^{1/3} \tag{2528}$$

The correction factor $\left(\frac{\rho_p}{\rho_f}\right)^{1/3}$ implies that bubbles can survive stronger turbulence than droplets [457]. This correction can overestimate the diameter of the bubbles. However, you can increase the breakup rate by decreasing the critical Weber number.

The difference between the two models is in the function f in Eqn. (2527).

Martinez-Bazan Breakage Rate

For the function f in Eqn. (2527), the Martinez-Bazan model assumes a square root shape. The Martinez-Bazan model implies that there is a minimum diameter below which no breakup occurs.

Tsouris and Tavlarides Breakage Rate

For the function f in Eqn. (2527), the Tsouris and Tavlarides model uses an exponent. The Tsouris and Tavlarides model predicts that any droplet can be broken (there is no minimum diameter), but the breakup probability decreases exponentially with droplet diameter.

Coulagloglou and Eskin

The Coulaloglou and Eskin model predicts a broader size distribution than the other two models.

In laminar flow, the dynamic pressure is applied by mean shear, not turbulence, so the equation for the Weber number reads:

$$\mathrm{We} = \frac{\left(\dot{\gamma} d_p\right)^2 d_p}{\sigma} \tag{2529}$$

where $\dot{\gamma}$ is the mean shear rate and σ is surface tension.

Coalescence Models

The coalescence process is modeled in two parts, a collision rate and a coalescence efficiency.

Turbulent Collision Rate

Applies to turbulent coalescence only. The coalescence rate is a product of collision rate and coalescence efficiency. The turbulent collision rate is calculated as the probability of collision of two spheres with diameters d_1 and d_2 moving with random relative velocity $v = \left(\epsilon_f \frac{d_1 + d_2}{2}\right)^{1/3}$.

The model has no fitting parameters, apart from those in the coalescence efficiency submodels.

Luo Coalescence Efficiency

The Luo coalescence efficiency model ([466]) assumes that the coalescence happens when the contact time $t_{contact}$ is bigger than $t_{rupture}$ — the time that is needed to break the liquid film separating two bubbles:

$$\text{Coalescence Efficiency} \sim \exp\left(-C\frac{t_{rupture}}{t_{contact}}\right) \tag{2530}$$

where C is the only calibration constant of the model. High C means that there is no coalescence; the default value is $C = 1$.

Film Drainage Coalescence Efficiency

The Film Drainage coalescence efficiency model is a generic model that is compatible with any collision rate model.

$$\text{Coalescence Efficiency} \sim \exp\left(-C\frac{\mathrm{We}^m}{\mathrm{Re}^n}\right) \tag{2531}$$

where We is the collision Weber number:

$$\mathrm{We} = \frac{\rho U_c^2 d_{eff}}{\sigma} \tag{2532}$$

and Re is the collision Reynolds number:

$$\mathrm{Re} = \frac{\rho U_c d_{eff}}{\mu} \tag{2533}$$

The velocity of collision U_c is provided by the collision rate model. d_{eff} is the effective diameter of the colliding droplets or bubbles.

Laminar Collision Rate

For the laminar collision rate, the relative velocity of two spheres is due to the mean shear rate:

$$v = \dot{S}\frac{d_1 + d_2}{2} \tag{2534}$$

Enhanced Power Law Collision Rate

This collision rate model applies to both Turbulent Coalescence and Laminar Coalescence.

Enhanced Power Law Collision Rate with its adjustable parameters allows investigation of alternative coagulation kernels K between particles of two sizes d_1 and d_2, scaled by characteristic diameter d_0.

$$K(d_1, d_2) = C\left[\left(\frac{d_1}{d_0}\right)^a + \left(\frac{d_2}{d_0}\right)^a\right]^b \left[\left(\frac{d_1}{d_0}\right)^c + \left(\frac{d_2}{d_0}\right)^c\right]^d \tag{2535}$$

Since the product of the kernel with the number densities of the two colliding sizes is a rate in events per second per m^3, the calibration constant C has dimensions of m^3 per second.

Mean Particle Diameters

For many practical applications, the size distribution is too detailed a description of the dispersed phase, and more integral characteristics are needed. The main output from the population balance calculation is the mean particle diameter. There are different definitions of the mean diameter depending on the weight function of the averaging.

Differences in values between these diameters are indicators of non-uniformity in the distribution. Variance can be quantified from these differences.

The general form of the formula for the mean diameter, based on moments of the number density, is:

$$d_{pq} = \sqrt[(p-q)]{\frac{\sum n_i d_i^p}{\sum n_i d_i^q}} \tag{2536}$$

Choosing appropriate values for p and q allows you to obtain the various alternative definitions for diameter. The particular and most important types of mean diameter that are implemented in the AMUSIG model are:

Surface Mean Diameter (commonly known as the Sauter mean diameter)

$$d_{32} = \frac{\sum n_i d_i^3}{\sum n_i d_i^2} \tag{2537}$$

For the AMUSIG model, the mean diameter d_{32} is labeled as Surface Mean Diameter to avoid a name clash with the existing usage in the S-Gamma model.

Volume Mean Diameter

$$d_{43} = \frac{\sum n_i d_i^4}{\sum n_i d_i^3} \tag{2538}$$

Volume-Based Diameter

$$d_{30} = \sqrt[3]{\frac{\sum n_i d_i^3}{\sum n_i}} \tag{2539}$$

Initial and Boundary Particle Size Distributions

STAR-CCM+ samples the initial conditions once, and then attempts to approximate the cumulative distribution for each method with a few groups. It resamples the inflow conditions at every iteration, and can contribute to different size-groups depending on the conditions inside the converging flow next to the boundary.

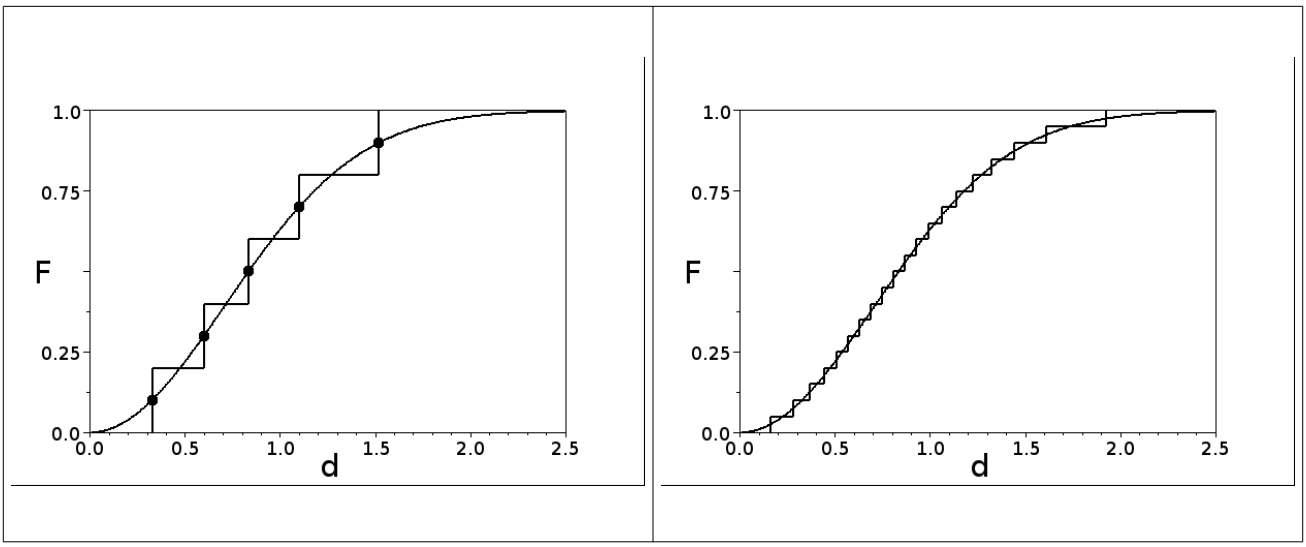
The AMUSIG method is volume-based, that is, at each point all size-groups have approximately the same volume fractions. The initial and boundary conditions are also formulated in terms of volume distributions. The volume fraction of all the particles (droplets, bubbles) that have a diameter less than or equal to d is:

$$\alpha_d F(d) \tag{2540}$$

where α_d is the volume fraction of the dispersed phase, and $F(d)$ is the cumulative size distribution (CSD).

The AMUSIG method represents the dispersed phase as an ensemble of the size-groups, so the CSD of the method has a staircase shape, as shown below.

Rosin-Rammler distribution and its approximation by the AMUSIG method: 5 size-groups (left), and 20 size-groups (right).



The position of an increment of F corresponds to the group diameter of the corresponding size-group; all increments of F have the same height.

In order to approximate a continuous distribution by the AMUSIG method, it is assumed that the middle of each increment (the dots in the 5 size-groups figure, above left) belongs to the curve. For M size-groups, the diameter of the ith size-group is calculated as the solution of the following equation:

$$F(d_i) = \frac{2i-1}{2M},\ i = 1, \ldots, M \tag{2541}$$

The quality of the approximation increases with the number of size-groups. However, you can verify that even a small number of groups (for example, $M = 5$) captures the main characteristics of the distribution, for example, d_{43}, d_{32}.

The distributions that are used in the AMUSIG method are described below.

Uniform distribution

This distribution assumes that the particles are uniformly distributed between a minimum size and a maximum size:

$$F(d) = \begin{cases} 0 & , \quad d < d_{\min} \\ \dfrac{(d - d_{\min})}{(d_{\max} - d_{\min})} & , \quad d_{\min} \le d \le d_{\max} \\ 1 & , \quad d > d_{\max} \end{cases} \tag{2542}$$

where $d_{\min}$ is the minimum diameter and $d_{\max}$ is the maximum diameter.

The uniform distribution is recommended in most cases, because it allows for easy control of the size distribution. In many cases, the initial (inlet) distribution is known only approximately and has a minor effect on the solution, so specifying the maximum size and the minimum size is sufficient.

The mean diameters of the uniform distribution are:

$$d_{pq} = \sqrt[p-q]{\frac{(q-2)\left(d_{\max}^{p-2} - d_{\min}^{p-2}\right)}{(p-2)\left(d_{\max}^{q-2} - d_{\min}^{q-2}\right)}} \tag{2543}$$

In particular:

$$d_{43} = \frac{d_{\max} + d_{\min}}{2}, \; d_{32} = \frac{d_{\max} + d_{\min}}{\ln(d_{\max}) - \ln(d_{\min})} \tag{2544}$$

Rosin-Rammler distribution

The Rosin-Rammler distribution, also known as the Weibull distribution, is:

$$F(d) = 1 - \exp\left(-\left[\frac{d}{\lambda}\right]^k\right) \tag{2545}$$

where k is the shape parameter and λ is the size parameter.

This distribution is widely used in particle size analysis of comminution processes. In the particular case $k = 1$, it becomes an exponential distribution. The parameter λ has the dimension of length and specifies the characteristic size, while the parameter k is dimensionless and controls the spread of the distribution (a higher k implies a narrower distribution).

The mean diameters of the Rosin-Rammler distribution are:

$$d_{pq} = \lambda \sqrt[p-q]{\frac{\Gamma\left(\frac{p-1}{k}\right)}{\Gamma\left(\frac{q-1}{k}\right)}} \tag{2546}$$

where Γ is the Euler gamma function. It can be seen that d_{pq} is determined only for $p > 1, \; q > 1$. In particular, d_{10} and d_{30} are not determined for the Rosin-Rammler distribution.

Log-normal distribution

The log-normal distribution implies that the logarithm of diameters has a normal distribution with mean μ and standard deviation σ. The formula for the CSD reads:

$$F(d) = \frac{1}{2} + \frac{1}{2}\mathrm{erf}\left(\frac{\ln d - \mu}{\sqrt{2}\sigma}\right) \tag{2547}$$

The mean diameters of the log-normal distribution are:

$$d_{pq} = \exp\left(\mu + \frac{1}{2}(p + q - 6)\sigma^2\right) \tag{2548}$$

The equation above provides an easy way to fit the log-normal distribution to the experimental data. Knowing two different diameters, for example d_{pq} and d_{st}, the following equations are obtained:

$$\frac{d_{pq}}{d_{st}} = \exp\left(\frac{1}{2}(p+q-s-t)\sigma^2\right),\ \sigma^2 = 2\frac{\ln\left(\frac{d_{pq}}{d_{st}}\right)}{p+q-s-t} \tag{2549}$$

Substitution of σ^2 into the previous equation yields (after some algebra):

$$\mu = \frac{6-s-t}{p+q-s-t}\ln\left(d_{pq}\right) + \frac{p+q-6}{p+q-s-t}\ln\left(d_{st}\right) \tag{2550}$$

Large Scale Interfaces

The large scale interface detection model is implemented within the multiple flow regime framework to enable the detection of a group of cells that contain a large interface.

Interface Detection

This model is based on the LIM method of Coste [415].

The interface detection for a pair of primary (p) and secondary (s) phases is achieved in two steps:

1. Identifying the cells that contain the interface to derive a raw interface.
2. Reducing the raw interface to a one-cell-thick interface.

Step 1: Identifying the cells that contain the interface

A sharp gradient of volume fraction for both the phases is calculated as:

$$\nabla\alpha_c = \frac{1}{\Delta v}\sum_f \alpha_{c,f}\Delta s_f \tag{2551}$$

where:

- Subscript c is for the phase.
- Subscript f is for the cell-face.
- Δv represents the volume of the cell.
- Δs represents the surface area of the cell.

To evaluate Eqn. (2551), the volume-fraction value at the cell-face for neighboring cells A and B is calculated as follows:

$$\begin{aligned} &\text{if } \alpha_A + \alpha_B < 1 && \frac{2}{\alpha_f} = \frac{1}{\alpha_A} + \frac{1}{\alpha_B} \\ &\text{if } \alpha_A + \alpha_B = 1 && \alpha_f = 0.5 \\ &\text{if } \alpha_A + \alpha_B > 1 && \frac{2}{1-\alpha_f} = \frac{1}{1-\alpha_A} + \frac{1}{1-\alpha_B} \end{aligned} \tag{2552}$$

This selective use of harmonic averaging is necessary to achieve a sharp gradient.

Once the sharp gradient of the volume fraction has been calculated for each phase of the phase interaction, the sharp gradient of the volume fraction for the phase interaction (ps) is calculated by taking the harmonic average of the absolute value of the individual components (i) of phase as:

$$\nabla\alpha_{ps} = \partial_{x_i}\alpha_{ps} = \frac{2\left|\partial_{x_i}\alpha_p\right|\left|\partial_{x_i}\alpha_s\right|}{\left|\partial_{x_i}\alpha_p\right| + \left|\partial_{x_i}\alpha_s\right|} \tag{2553}$$

This sharp gradient of the volume fraction for the phase interaction is compared to a maximum value of the sharp gradient of the volume fraction in each direction, which is calculated as:

$$\left(\nabla\alpha_{ps}\right)_{\max} = \frac{0.5}{\Delta v}\Sigma_f \Delta s_f \tag{2554}$$

The interface exists in a cell when at least one component satisfies:

$$\partial_{x_i}\alpha_{ps} > \frac{\left(\partial_{x_i}\alpha_{ps}\right)_{\max}}{C_1} \tag{2555}$$

All of the cells that satisfy Eqn. (2555) are considered to be a raw representation of the interface, which is then refined to obtain a single cell thick interface. The thickness of the raw interface can be adjusted by modifying the primary criteria C_1, whose default value is taken as $C_1 = 5$.

Step 2: Constructing a one-cell-thick interface

To reduce the raw interface to a one-cell-thick interface, it is necessary to select between two neighboring cells that belong to the raw interface and are also parallel to the interface. For each pair of such cells (A, B) with $\overrightarrow{AB}$ representing the line joining the centroids of the two cells, if the following condition is satisfied, the cell in which the volume fraction of the primary phase is less is removed from the raw interface to obtain the final interface:

$$\left|\left(\nabla\alpha_{ps,A} + \nabla\alpha_{ps,B}\right)\cdot\overrightarrow{AB}\right| > C_2\left\|\nabla\alpha_{ps,A} + \nabla\alpha_{ps,B}\right\|\left\|\overrightarrow{AB}\right\| \tag{2556}$$

where:

- $|x|$ is the absolute value of a scalar x.
- $\|\vec{x}\|$ is the magnitude of vector $\vec{x}$.

Adaptive Interface Sharpening (ADIS) Scheme for Volume Fraction

The Multiple Flow Regime model lets you model dispersed and segregated two-phase flows within a single framework. If an interface sharpening scheme for volume fraction is used for the entire domain, the result is good in the regions where a sharp interface exists naturally but can exhibit artificial sharpening in naturally smooth regions. Similarly, if a standard TVD scheme is used for the entire domain, the result can show artificial diffusion in sharp regions while preserving naturally smooth regions.

The Adaptive Interface Sharpening (ADIS) scheme addresses this problem by using an interface sharpening scheme (the High-Resolution Interface Capturing (HRIC) scheme) in the vicinity of a large scale interface and using the standard TVD scheme (a first order or second order scheme) away from the sharp interface.

The Large Scale Interface Detection model provides a "Large Interface Marker Band" which represents the large interface. To avoid sharp changes that are due to the use of different advection schemes, a blending function is built upon this Large Interface Marker Band. The blending function goes from 1 to 0 across a specified number of cells. You specify the number of cells in the *Number of Cell Layers for Smoothing* property of the Adaptive Interface Sharpening model. The default value is 2 cells.

The face value of the volume fraction is calculated as:

$$\alpha_f = f_{blending}\alpha_{f^{HRIC}} + \left(1 - f_{blending}\right)\alpha_{f^{TVD}} \tag{2557}$$

Contents:

Turbulence Damping
Surface Tension

Turbulence Damping

In the vicinity of a large scale interface, a boundary layer develops and the flow is close to laminar. Turbulence close to the interface needs to be damped properly depending on the local flow structure.

Following Egorov [469] and Lo & Tomasello [461], a symmetric damping procedure is used to model the interface as a moving wall for both of the phases in the interface region.

Currently, the interface turbulence damping model is available only with the $k - \omega$ turbulence model. The interface turbulence damping model uses low Re treatment to estimate the specific dissipation rate near the large scale interface:

$$\omega_i = B\frac{6\mu_i}{\beta\rho_i\Delta n^2} \tag{2558}$$

where the subscript i is for the ith phase.

The normal distance from the interface, Δn is calculated from the cell volume as $\Delta n = \sqrt[3]{\Delta \mathrm{V}}$.

When the aspect ratio of the interface is too high, the turbulence damping constant B must account for that factor.

To enforce ω_i as a boundary condition, a source term is added to the specific dissipation rate equation of the phases as:

$$S_\omega = \alpha \mathrm{A} \cdot \Delta n \beta \rho_i \left(\omega_i\right)^2 \tag{2559}$$

where $A = |\nabla\alpha|$ is the interfacial area density, to ensure that the source term is added only in the vicinity of the large scale interface.

Surface Tension

Immiscibility of two fluids is a result of strong cohesion forces between their molecules and depends on the nature of the fluids. The experimentally determined surface tension coefficient σ expresses the ease with which the fluids can be mixed.

The surface tension force is an interfacial force, which is modeled as a volumetric force using the Continuum Surface Force (CSF) approach of Brackbill and others [522].

For more information, see Surface Tension Formulation.

Strubelj and others [495] proposed an extension of the Continuum Surface Force (CSF) model for a two-fluid model. The surface tension force is split among the phases occupying the cell as:

$$\mathbf{f}_{\sigma,i} = \beta_i \mathbf{f}_\sigma \tag{2560}$$

where the subscript i indicates the ith phase and β_i denotes the splitting factor of the surface tension force. The pressure gradient within the two-fluid model is the summation of the momentum equations:

$$\sum_i \alpha_i \nabla p = \nabla p = \sum_i \beta_i \mathbf{f}_\sigma \tag{2561}$$

At kinetic equilibrium, the pressure gradient is equal to surface tension force, thus:

$$\sum_i \beta_i = 1 \tag{2562}$$

Following Strubelj, the following form of surface tension force is implemented:

$$\mathbf{f}_{\sigma,i} = \alpha_i \sigma \kappa \nabla \alpha_1 \tag{2563}$$

Particulate Flows

Particulate flows describe multiphase flow regimes that deal with gas-solid or liquid-solid flows.

Particle-Laden Flows

Particle-laden flow represents flow of discrete solid particles in a continuous gas. Examples for this type of flow regime are pneumatic transport of powders.

Granular flow represents flow of discrete solid particles in a continuous fluid. In case of high particle loading, interactions between particles that impose additional stresses cannot be neglected, and it is required to model these by use of the granular flow model. Examples for this type of flow regime are fluidized beds or pneumatic conveyors.

Slurry Flows

Slurry flow represents the flow of discrete solid particles in liquids. Examples are cement.

Contents:

Fluid-Solid Drag Coefficients
Granular Flow
Erosion
Crystallization

Fluid-Solid Drag Coefficients

Arastoopour Drag Model

The Arastoopour Drag Model is appropriate for solid dispersed-phase applications such as fluidized beds, where particle loadings are high.

The Arastroopour drag coefficient [399] is:

$$A_D = \left(0.336 + \frac{17.3}{Re_d}\right)\frac{\alpha_d \rho_d |\mathbf{v}_r|}{l_{cd}}\alpha_c^n \tag{2564}$$

where the exponent n is set to -2.8 by default.

Gidaspow Drag Model

The Gidaspow Drag Model is appropriate for solid dispersed-phase applications such as fluidized beds where particle loadings are high. The most widely used formulae are the Ergun equation for regions of high solid particle concentrations and a modified Stokes law for regions of lower concentration [407], [516], [487].

A transition volume fraction, α_{tr}, is used as the basis for switching between these formulae as follows:

$$A_D = \begin{cases} \dfrac{150\alpha_d^2\mu_c}{\alpha_c l_{cd}^2} + \dfrac{1.75\alpha_d\rho_c|\mathbf{v}_r|}{l_{cd}} & \alpha_d \geq \alpha_{tr} \\ \dfrac{3}{4}C_D\dfrac{\alpha_d\rho_c}{l_{cd}}|\mathbf{v}_r|\alpha_c^n & \alpha_d < \alpha_{tr} \end{cases} \tag{2565}$$

The drag coefficient C_D in Eqn. (2565) is from the correlation of Schiller and Naumann Eqn. (2265).

Even below the transition volume fraction, $\alpha_d \leq \alpha_{tr}$, the inter-particle force affects the particle velocity. This hindered settling is accounted for by the correction factor α_c^n in Eqn. (2565).

Specify both the value of the transition volume fraction (by default α_{tr} = 0.2) and the hindered-settling exponent (by default n = -1.65).

The Gidaspow drag coefficient method reduces to the Wen and Yu correlation when the transition volume fraction is set to 1.0.

Syamlal O'Brien Drag Model

The Syamlal O'Brien drag coefficient is based on measurements of terminal velocity in settling beds. The drag coefficient defines fluid-solid momentum exchange as a function of the particle phase fraction, particle Reynolds number, and a terminal velocity expression.

At the minimum fluidization condition, the force balance yields the following equation:

$$\frac{A_D}{\alpha_g}\cdot\left(v_{mf}\right) = \left(1-\alpha_g\right)\cdot\left(\rho_s - \rho_g\right)\cdot g \tag{2566}$$

where:

$$A_D = \frac{3}{4}\frac{\alpha_s \alpha_g \rho_g}{d_p} C_D \left|\mathbf{v}_{mf}\right| \tag{2567}$$

You require an accurate value of C_D for the given minimum fluidization velocity, v_{mf}, such that Eqn. (2566) is satisfied.

Eqn. (2566) can be recast in the following form:

$$\frac{3}{4} C_{D_s} \mathrm{Re}_{ts}^2 = \frac{3}{4} C_D \mathrm{Re}_t^2 = Ar \tag{2568}$$

The above relation holds for a single particle as well as a cluster of particles. C_{D_s} is the drag coefficient for a single particle. Re_{ts} is the Reynolds number based on terminal settling velocity of single particle. The quantities without the s subscript are those for a group of particles. Ar, the non-dimensional Archimedes number, is a function of the known quantities and is given by:

$$Ar = \frac{\rho_g}{\mu_g^2} d_p^3 \left(\rho_p - \rho_g\right) g \tag{2569}$$

Define V_r as the ratio of single particle terminal velocity to that of a cluster of particles:

$$V_r = \frac{\mathrm{Re}_t}{\mathrm{Re}_{ts}} \tag{2570}$$

Substituting for V_r in Eqn. (2568), yields the following relation between single and multi-particle drag coefficients:

$$C_D = \frac{C_{D_s}}{V_r^2} \tag{2571}$$

The single particle drag law employed by Syamlal-O'Brien is that given by Del Valle [418]:

$$C_{D_s} = \left(0.63 + \frac{4.8}{\sqrt{\mathrm{Re}_{ts}}}\right)^2 \tag{2572}$$

To complete the modeling, Syamlal-O'Brien choose an analytical formulation for V_r:

$$V_r = 0.5\left[A - 0.06 Re_t + \sqrt{\left(0.06 Re_t\right)^2 + 0.12 Re_t (2B - A) + A^2}\right] \tag{2573}$$

where

$$A = \epsilon^{4.14} \tag{2574}$$

and

$$B = \begin{cases} C_2 \epsilon^{1.28} & \epsilon < \alpha_{tr} \\ \epsilon^{C_1} & \epsilon \geq \alpha_{tr} \end{cases} \tag{2575}$$

where:

α_{tr} is the transition volume fraction, which is set to 0.85 by default.

C_1 and C_2 are model constants, which are set to 2.65 and 0.8 respectively [428].

To ensure continuity of B, C_1 and C_2 have the following relationship:

$$C_1 = 1.28 + \frac{\log C_2}{\log 0.85} \tag{2576}$$

Adjustment of Syamlal-O'Brien Drag Law

From the known quantities, the Archimedes number can be estimated using Eqn. (2568). Substituting Eqn. (2572) in Eqn. (2568) gives an explicit formula for Re_{ts}:

$$\mathrm{Re}_{ts} = \left[\frac{\sqrt{4.8^2 + 2.52\sqrt{\frac{4Ar}{3}}} - 4.8}{1.26}\right]^2 \tag{2577}$$

For fluidized beds where information about the minimum fluidization velocity U_{mf} is available, the multi-particle Reynolds number can be calculated as follows:

$$\mathrm{Re}_t = \frac{\rho_g U_{mf} d_p}{\epsilon_{mf}\mu_f} \tag{2578}$$

The minimum fluidization velocity is specified as the superficial velocity, so it is divided by the gas volume fraction at the minimum fluidization condition, ϵ_{mf}. Typically we assume that $\epsilon_{mf} = 0.4$.

For small particles or for small Reynolds numbers ($\mathrm{Re}_t < 20$), the following expression obtained from the first term of the Ergun equation can be used to estimate ϵ_{mf}:

$$U_{mf} = \frac{d_p^2(\rho_p - \rho_g)\, g}{150\mu_g} \frac{\epsilon_{mf}^3}{1 - \epsilon_{mf}} \tag{2579}$$

Knowing Re_{ts} and Re_t, V_r can be estimated using Eqn. (2570). Note that during the tuning process, the void fraction, ϵ, used in Eqn. (2574) and Eqn. (2575) should be ϵ_{mf}.

Finally, C_2, can be estimated from Eqn. (2573) and Eqn. (2575).

The parameter C_1 is obtained from C_2 using Eqn. (2576).

For more information and an example case, see the technical report Study of Gas-Solid Fluidized Bed - Tuning Syamlal-O'Brien Drag Model in the Technical Library on the Steve Portal.

Granular Flow

Particle-particle interactions can be significant in solid particle flows and therefore require their own set of models.

STAR-CCM+ provides the granular pressure model for gas-solid flows such as fluidized beds. The model introduces the concept of the *packing limit*. Below this limit, particles are loosely packed and inter-particle

collisions determine their motion; these collisions are handled as part of a granular temperature model. Above the packing limit, friction between particles dominates their motion and a frictional stress model is used.

In the Multiphase Segregated Flow model, all phases are treated as an interpenetrating continua (fluid-like), so there is a viscous term for each phase. For solid particles, the viscous term is modeled using granular viscosity when the granular model is enabled. However, in the absence of the granular model, a small value is used for viscosity (via pseudo-solid viscosity) to close the viscous term. The pseudo-solid viscosity is the laminar (molecular) viscosity of the continuous solid-phase, and is used for calculating the viscous stress term in the momentum equations. It is activated only for laminar cases.

Contents:

Granular Stress
Granular Temperature
Granular Boundary Conditions
Granular Drag Coefficient

Granular Stress

Particle-particle interactions can be significant in solid particle flows. The particle-particle interaction forces are considered as internal forces within phases.

STAR-CCM+ provides two models to account for the solid particle-particle interaction forces. These models apply exclusively to the dispersed phase.

Solid Pressure Force

A compaction, or 'solid pressure', force is provided as one model of particle-particle interactions within a solid particle flow.

It is implemented as the following internal force term in the momentum equation for a solid particle phase:

$$\left(\mathbf{F}_{int}\right)_i = -\frac{\alpha_i}{\Sigma_{j=1}^{M}\alpha_{p,j}}\left[e^{A\left(\alpha_{p,\max} - \sum_{j=1}^{M}\alpha_{p,j}\right)}\right]\nabla\left(\sum_{j=1}^{M}\alpha_{p,j}\right) \tag{2580}$$

where:

$\alpha_{p,max}$ is the maximum packing limit

$\alpha_{p,j}$ is the volume fraction of the jth particle phase

M is the number of particle phases

A is the model constant, set to -600.0 by default

When activated during a multiphase flow simulation, the solid pressure force comes into action only when the solid volume fraction in any part of the domain approaches $\alpha_{p,max}$. The force acts to resist the formation of unphysical solid packing fractions by increasing exponentially as the solid volume fraction approaches the maximum packing limit. This model is useful to study a multiphase flow containing solids without needing to consider the effects of solid stress due to friction and/or collision.

The maximum packing limit is set to 0.624 by default, representative of rigid spherical solid particles.

Granular Pressure

The granular pressure model provides a way to estimate the stresses in the granular medium. The model is applicable to gas-solid flows and is typically used for fluidized bed applications.

Fluidized bed reactors are widely used in combustion, catalytic cracking and various other chemical and metallurgical processes. The particle motion can be classified into two regimes - kinetic and frictional. In the kinetic regime, the particle motion is governed by collisions between particles. By analogy with gas kinetic theory, and after assuming distribution and collision properties for the particles, a kinetic theory of granular pressures can be derived for single particle flows [434], [465].

In the frictional regime, where the particle motion is contact-dominated, empirical formulations that are borrowed from soil mechanics are employed. It is possible to work with multiple particle phases (that is, particles of different sizes) when using the Granular Pressure Model. Schaeffer, [490] described the stresses in the frictional regime that is based on a simplified version of the critical state theory. Johnson and Jackson, [444] also developed constitutive relations for stresses in the frictional regime.

A full stress tensor is then obtained that describes the particle-particle interaction terms in the solid phase momentum equation:

$$\left(F_{\mathrm{int}}\right)_p = \nabla\cdot \overline{S}_p \tag{2581}$$

where the granular stress tensor is defined as:

$$\overline{S}_p = \left[-P_p + \left(\xi_p - \frac{2}{3}\mu_p\right)\nabla\cdot \mathbf{v}_p\right]\mathbf{I} \tag{2582}$$

where:

- P_p is the solid pressure.
- μ_p is the effective granular viscosity.
- I is the identity tensor.
- The subscript p,is used to denote the particle phase.

ξ_p is the bulk viscosity. If you activate only the frictional regime, ξ_p is set to 0. If you activate only the kinetic regime, ξ_p is computed using Eqn. (2613). If you activate both regimes, ξ_p is computed based on the frictional model you select.

Kinetic Regime

The kinetic regime is characterized by particle volume fractions below the maximum packing limit, $\alpha_{p,max}$. In the kinetic regime, the particle motion is governed by collisions between particles. The granular temperature, θ_p, is a measure of the fluctuating kinetic energy of particles:

$$\theta_p = \frac{1}{3}\langle \mathrm{v}'_{pi}\cdot \mathrm{v}'_{pi}\rangle \tag{2583}$$

where v'_{pi} is the random fluctuating velocity of the particle phase and $\langle\ \ \rangle$ denotes ensemble averaging.

The probability of particle collisions is characterized by the particle radial distribution function, g_0. The radial distribution function represents the spatial distribution of the particles, and therefore their proximity, and can be related to the dispersed-phase volume fraction. The form in [422] has been implemented:

$$g_0 = \frac{3}{5}\left[1-\left(\frac{\alpha_p}{\alpha_{p,max}}\right)^{1/3}\right]^{-1} \tag{2584}$$

This function becomes undefined when α_p approaches the maximum packing limit, $\alpha_{p,max}$. In this limit, Eqn. (2584) is supplemented by the equation that is proposed in [508] for $\alpha_p > \alpha_{crit}$:

$$g_0 = 1.08 \times 10^3 + 1.08 \times 10^6\left(\alpha_p - \alpha_{crit}\right) + 1.08 \times 10^9\left(\alpha_p - \alpha_{crit}\right)^2 + 1.08 \times 10^{12}\left(\alpha_p - \alpha_{crit}\right)^3 \tag{2585}$$

where:

$$\alpha_{crit} = \alpha_{p,max} - 0.001 \tag{2586}$$

The energy that is dissipated in particle collisions is characterized by the coefficient of restitution, e. You can specify e (by default e = 0.9).

Mixture Maximum Packing Fraction

Yu and Standish [518] proposed an empirical correlation to estimate a maximum packing limit for a mixture of multiple particles. There is no limit to the number of particles in the mixture for this correlation.

The cumulative particle volume fraction, $\alpha_{p,sum}$, for a mixture of M particles is:

$$\alpha_{p,sum} = \sum_{n=1}^{M} \alpha_n \tag{2587}$$

The mixture maximum packing fraction is:

$$\alpha_{p,max}^{m} = min\left\{\frac{\alpha_{i,max}}{1-\sum_{j=1}^{i}\left(1-\frac{\alpha_{i,max}}{p_{ij}}\right)\left(\frac{cx_i}{X_{ij}}\right)-\sum_{j=i+1}^{M}\left(1-\frac{\alpha_{i,max}}{p_{ij}}\right)\left(\frac{cx_i}{X_{ij}}\right)}\right\} \tag{2588}$$

where i=1, 2, ... , M are the particle phases in the simulation arranged in order from coarsest to finest.

$$cx_i = \frac{\alpha_i}{\alpha_{p,sum}} \tag{2589}$$

$$X_{ij} = \begin{cases} \dfrac{1-r_{ij}^2}{2-\alpha_{i,max}} & j < i \\ 1-\dfrac{1-r_{ij}^2}{2-\alpha_{i,max}} & j \geq i \end{cases} \tag{2590}$$

$$r_{ij} = \begin{cases} \dfrac{d_{p,i}}{d_{p,j}} & j \leq i \\ \dfrac{d_{p,j}}{d_{p,i}} & j > i \end{cases} \tag{2591}$$

$$p_{ij} = \begin{cases} \alpha_{i,max} + \alpha_{i,max}\left(1-\alpha_{i,max}\right)\left(1-2.35r_{ij} + 1.35r_{ij}^2\right) & r_{ij} \leq 0.741 \\ \alpha_{i,max} & r_{ij} > 0.741 \end{cases} \tag{2592}$$

Frictional Regime

The following models are available in the frictional regime:

- Modified Johnson
- Schaeffer

Modified Johnson Frictional Model

The Modified Johnson [444] model gets activated in regions where the cumulative particle volume fraction, $\alpha_{p,sum}$, exceeds the specified minimum frictional volume fraction packing limit, $\alpha_{p,min}^f$. The solid pressure in this regime is given by the frictional solid pressure, P_p^f:

$$P_p^f = \begin{cases} \frac{\alpha_p}{\alpha_{p,sum}} Fr \left(\frac{\left(\alpha_{p,sum} - \alpha_{p,min}^f\right)^r}{\left(\alpha_{p,max}^m - \alpha_{p,sum}\right)^s} \right) & \alpha_{p,sum} > \alpha_{p,min}^f \\ 0 & \alpha_{p,sum} \leq \alpha_{p,min}^f \end{cases} \tag{2593}$$

where:

$\alpha_{p,max}^m$ is the mixture maximum packing fraction

Fr, r, and s are user-specified empirical constants

The effective granular viscosity in this regime, μ_p^f, is given by the frictional viscosity:

$$\mu_p^f = \begin{cases} min\left(\frac{P_p^f \sin\phi}{\sqrt{2\mathbf{D_p}\mathbf{D_p} + \frac{\theta_p}{d_p^2}}}, \mu_{p,max} \right) & \alpha_{p,sum} > \alpha_{p,min}^f \\ 0 & \alpha_{p,sum} \leq \alpha_{p,min}^f \end{cases} \tag{2594}$$

where:

- ϕ is the angle of internal friction. The recommended value is 28.5° when using the Johnson model.
- $\mathbf{D_p}$ is the strain rate tensor for the particle phase.
- $\mu_{p,max}$ is the user-defined Maximum Solid Viscosity limit applied to μ_p.

The term involving granular temperature, θ_p is ignored when the kinetic regime is not activated.

In the frictional regime, the bulk viscosity ξ_p is set to 0 where the cumulative particle volume fraction exceeds the minimum frictional volume fraction $\alpha_{p,min}^f$, and is computed using Eqn. (2613) everywhere else.

Schaeffer Frictional Model

The Schaeffer [490] model gets activated in regions where the cumulative particle volume fraction, $\alpha_{p,sum}$, exceeds the specified mixture maximum packing fraction, $\alpha_{p,max}^{m}$. The solid pressure in this regime is given by the frictional solid pressure, P_p^f:

$$P_p^f = \begin{cases} \frac{\alpha_p}{\alpha_{p,sum}}\left(10^{25}\left(\alpha_{p,sum}-\alpha_{p,max}^{m}\right)^{10}\right) & \alpha_{p,sum} > \alpha_{p,max}^{m} \\ 0 & \alpha_{p,sum} \leq \alpha_{p,max}^{m} \end{cases} \tag{2595}$$

The effective granular viscosity in this regime, μ_p^f, is given by the frictional viscosity:

$$\mu_p^f = \begin{cases} min\left(\frac{P_p^f \sin\phi}{\sqrt{4I_{2D}}}, \mu_{p,max}\right) & \alpha_{p,sum} > \alpha_{p,max}^{m} \\ 0 & \alpha_{p,sum} \leq \alpha_{p,max}^{m} \end{cases} \tag{2596}$$

where:

- ϕ is the angle of internal friction and is set to 25° by default.
- $\mu_{p,max}$ is the user-defined Maximum Solid Viscosity limit applied to μ_p.
- I_{2D} is the second invariant of the deviator of the strain rate tensor:

$$I_{2D} = \frac{1}{6}\left[\left(D_{p11}-D_{p22}\right)^2 + \left(D_{p22}-D_{p33}\right)^2 + \left(D_{p33}-D_{p11}\right)^2\right] + D_{p12}^2 + D_{p23}^2 + D_{p31}^2 \tag{2597}$$

In the frictional regime, where both regime and frictional are specified, the bulk viscosity, ξ_p, is set to 0 above the mixture minimum packing fraction, $\alpha_{p,max}^{m}$, and is computed using Eqn. (2613) everywhere else.

Granular Temperature

The granular temperature, θ_p, can be estimated by solving the conservation equation for granular energy.

The conservation equation for the fluctuating granular kinetic energy is given as:

$$\frac{3}{2}\left[\frac{\partial}{\partial t}\int_V \alpha_p\rho_p\theta_p\, dV + \oint_A \alpha_p\rho_p\theta_p\mathbf{v}_p \cdot d\mathbf{a}\right] = \oint_A \tau_p{:}\nabla\mathbf{v}_p \cdot d\mathbf{a} + \oint_A \kappa_{eff}\nabla\theta_p \cdot d\mathbf{a} - \int_V \gamma\, dV - \int_V J_p\, dV + \int_V \alpha_p\rho_p\epsilon_p\, dV \tag{2598}$$

The first term in Eqn. (2598) (above) is the granular energy production. Wall treatment is used to compute a modified value of granular energy production in the first cell adjacent to the wall.

The effective granular diffusion coefficient, κ_{eff}, is:

$$\kappa_{eff} = \kappa + \frac{3\mu_{pt}}{2\sigma_{pt}} \tag{2599}$$

where:

- κ is the granular diffusion coefficient
- μ_{pt} is the turbulent viscosity of the granular phase
- σ_{pt} is the turbulent granular diffusion Prandtl number. The default value is 1.

The granular diffusion coefficient, κ, comes in two forms:

- The Gidaspow form:

$$\kappa = \frac{150\rho_p d_p\sqrt{\pi\theta_p}}{384(1+e)g_0}\left[\frac{\alpha_p}{\alpha_{p,sum}} + \frac{12}{5}\alpha_p(1+e)g_0 + \frac{36}{25}\alpha_p\alpha_{p,sum}(1+e)^2 {g_0}^2\right] + 2\alpha_p\alpha_{p,sum}\rho_p d_p(1+e)g_0\sqrt{\frac{\theta_p}{\pi}} \tag{2600}$$

- The Syamlal form:

$$\kappa = \frac{15\alpha_p\rho_p d_p\sqrt{\pi\theta_p}}{4(41-33\eta)}\left[1 + \frac{12}{5}\eta^2(4\eta-3)\alpha_{p,sum}g_0 + \frac{16}{15\pi}(41-33\eta)\eta\alpha_{p,sum}g_0\right] \tag{2601}$$

where:

- ρ_p is the particle density.
- d_p is the particle diameter. The interaction length scale is set as equal to the particle diameter.
- $\eta = \frac{1}{2}(1+e)$

The dissipation of granular energy, γ, is:

$$\gamma = \frac{12(1-e^2)g_0}{d_p\sqrt{\pi}}\alpha_p\alpha_{p,\mathrm{sum}}\rho_p{\theta_p}^{3/2} \tag{2602}$$

The dissipation of granular energy due to inter-phase drag, J_p, is:

$$J_p = A_D\left[3\theta_p - \frac{A_D d_p(v_g - v_p)^2}{4\alpha_p\rho_p g_0\sqrt{\pi\theta_p}}\right] \tag{2603}$$

where A_D is the interphase momentum transfer coefficient.

The first term is attributed to the work of Gidaspow [434] and is always included. The second term is attributed to the work of Louge and others [464] and is included only if the *Cross-Correlation Term* property of the Granular Energy Transfer model is set to **Louge**.

The last term in Eqn. (2598), $\alpha_p\rho_p\epsilon_p$, is a source term that is due to the dissipation of granular phase turbulent kinetic energy. This term is present only if the granular phase is modeled using the Turbulent viscous regime.

Algebraic Model for Granular Temperature

By assuming local equilibrium, where energy dissipation and production balance each other, the energy equation can be reduced to a quadratic equation which is solved directly for θ_p. The equilibrium form has been implemented and the granular temperature is computed as:

$$\sqrt{\theta_p} = \frac{-K_{1p}\alpha_p D_{pii} + \sqrt{K_{1p}^2\alpha_p^2 D_{pii}^2 + 4K_{4p}\alpha_p\left[K_{2p}D_{pii}^2 + 2K_{3p}\left(D_{pij}D_{pij}\right)\right]}}{2\alpha_p K_{4p}} \tag{2604}$$

where:

$$K_{1p} = 2\left(1+e\right)\rho_p g_0 \tag{2605}$$

$$K_{2p} = \frac{4d_p\rho_p(1+e)_\alpha g_0}{3\sqrt{\pi}} - \frac{2}{3}K_{3p} \tag{2606}$$

$$K_{3p} = \frac{d_p\rho_p}{2}\left\{\frac{\sqrt{\pi}}{3(3-e)}\left[0.5\left(3e+1\right) + 0.4\left(1+e\right)\left(3e-1\right)\alpha_p g_0\right] + \frac{8\alpha_p g_0(1+e)}{5\sqrt{\pi}}\right\} \tag{2607}$$

$$K_{4p} = \frac{12\left(1-e^2\right)\rho_p g_0}{d_p\sqrt{\pi}} \tag{2608}$$

D_{pij} is the strain rate tensor:

$$D_{pij} = \frac{1}{2}\left(\frac{\partial v_{pi}}{\partial x_j} + \frac{\partial v_{pj}}{\partial x_i}\right) \tag{2609}$$

where ρ_p is the particle density and d_p is the particle diameter. The interaction length scale is equal to the particle diameter.

Expressions for the solid pressure, bulk viscosity, and effective granular viscosity are obtained as functions of the particle granular temperature, θ_p.

The minimum granular temperature in the domain is set to $1 \times 10^{-10} m^2/s^2$ by default but you can change the value if necessary. To estimate the granular temperature at boundaries, either extrapolate the granular temperature at the face cell or use a user-defined value. The specified granular temperature value is used to calculate the solid pressure at the boundaries, which has a significant influence on results.

The solid pressure is composed of collisional and kinetic contributions, [509]:

$$P_p = P_p^C + P_p^K \tag{2610}$$

$$P_p^C = 2g_0\rho_p\alpha_p^2\theta_p\left(1+e\right) \tag{2611}$$

$$P_p^K = \rho_p\alpha_p\theta_p \tag{2612}$$

The bulk viscosity is [490]:

$$\xi_p = \frac{4}{3}\alpha_p^2\rho_p d_p g_0\left(1+e\right)\sqrt{\frac{\theta_p}{\pi}} \tag{2613}$$

The effective granular viscosity μ_p is composed of collisional, kinetic, and frictional contributions:

$$\mu_p = \mu_p^C + \mu_p^K + \mu_p^f \tag{2614}$$

A user-defined Maximum Solid Viscosity limit is applied to μ_p.

$$\mu_p^C = \frac{4}{5}\alpha_p^2\rho_p d_p g_0\left(1+e\right)\sqrt{\frac{\theta_p}{\pi}} \tag{2615}$$

You can select between two forms of the particle kinetic viscosity:

- The form from Gidaspow [434] is:

$$\mu_p^K = \frac{10\rho_p d_p\sqrt{\pi\theta_p}}{96(1+e)g_0}\left[1+\frac{4}{5}g_0\alpha_p\left(1+e\right)\right]^2 \tag{2616}$$

- The form from Syamlal [405] is:

$$\mu_p^K = \frac{\alpha_p\rho_p d_p\sqrt{\pi\theta_p}}{6(3-e)}\left[0.5\left(3e+1\right)+\frac{2}{5}\left(1+e\right)\left(3e-1\right)g_0\alpha_p\right] \tag{2617}$$

The Syamlal version predicts lower viscosity at particle volume fractions below 0.3 [509]. You are advised to use the corresponding linearized drag law (Gidaspow or Syamlal) in accordance with the version of particle kinetic viscosity selected.

Granular Boundary Conditions

The granular boundary conditions have been implemented in STAR-CCM+ such that the two boundary conditions, shear stress specification and granular temperature specification, can be considered independent.

For the shear-stress (of the particle phase), the following condition is enforced when the partial-slip option is used [444]:

$$\frac{v_{sl}\cdot\tau_p\cdot\vec{n}}{\left|v_{sl}\right|}+\frac{\sqrt{3\theta_p}\pi\phi\rho_p\alpha_p g_0\left|v_{sl}\right|}{6\alpha_{p,max}}=0 \tag{2618}$$

where:

- The first term on the LHS is the component of particle stress at the wall in the direction of the slip velocity, v_{sl}.
- $\vec{n}$ is the unit normal at the boundary.
- ϕ is the specularity coefficient. This value varies between 0 for perfectly specular collisions and 1 for perfectly diffuse collisions.

For the granular temperature specification, the following condition is enforced when the Johnson-Jackson option is used [444]:

$$\kappa\nabla\theta_p\cdot\vec{n}+\frac{\pi\left(1-e_w{}^2\right)\rho_p\alpha_p g_0\theta_p\sqrt{3\theta_p}}{4\alpha_{p,max}}-\frac{\sqrt{3\theta_p}\pi\phi\rho_p\alpha_p g_0\left|v_{sl}\right|^2}{6\alpha_{p,max}}=0 \tag{2619}$$

where:

- κ is the granular diffusion coefficient.
- e_w is the coefficient of restitution for collisions between particles and the wall.

The third term on the LHS appears only when the Partial-Slip option is used for the shear stress specification.

Boundary Conditions for k and ϵ

When an appropriate turbulence model is active, the boundary conditions for k and ϵ are implemented in the following form [430]:

$$\mu_p \frac{dk_p}{dy} + D_w k_p = 0 \tag{2620}$$

$$\mu_p \frac{d\epsilon_p}{dy} + D_w \epsilon_p = 0 \tag{2621}$$

This formulation is along the same lines as that for the particle momentum Johnson-Jackson boundary condition:

$$\mu_p \frac{dv_p}{dy} + D_w v_p = 0 \tag{2622}$$

Granular Drag Coefficient

To model the drag between particle phases, the Gera-Syamlal drag coefficient is available.

Gera-Syamlal Granular Drag Force Model

The formulation for the Gera-Syamlal [405], [433] drag force model is:

$$A_D = 3(1+e)\left(\frac{\pi}{2} + c_f \frac{\pi^2}{8}\right)\frac{\left(d_i + d_j\right)^2}{2\pi\left(\rho_i d_i^{\,3} + \rho_j d_j^{\,3}\right)}\rho_i \rho_j g_0 |\mathbf{v}_r| \tag{2623}$$

where e is the coefficient of restitution for the phase interaction, c_f is the coefficient of friction for this interaction. Default value for c_f is 0.

Erosion

The erosion model simulates two mechanisms of erosion: impact wear and abrasive wear.

Impact wear predicts the rate of erosion due to direct impact of particles on the eroded surface. Abrasive wear predicts the rate of erosion due to abrasion of the surface by the particles.

The erosion model uses the particle phase information in the cell next to the wall to determine the following quantities:

- Particle volume fraction
- Granular temperature
- Particle velocity

The particle volume fraction and granular temperature are used to determine $\dot{m}_p$, the mass flow rate of particles impacting per unit area of the wall per unit time. The particle velocity adjacent to the wall is used to correlate $\dot{m}_p$ with the mass loss rate of wall material.

Impact Wear Erosion

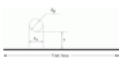
The impact erosion rate is defined as the mass of wall material eroded per unit area per unit time:

$$E_f = \dot{m}_p e_r \tag{2624}$$

where:

- $\dot{m}_p$ is the mass flow rate of particles per unit area impacting on the face
- e_r is the erosion ratio

The diagram below is a schematic view of a single particle approaching a wall:



where:

- d_p is the particle diameter
- a_c is the average boundary area per particle
- s is the average distance between the boundary and the surface of an adjacent particle

For a unit area, the mass flow rate of particles hitting the area is given by:

$$\dot{m}_p = M_p C_p N_p \tag{2625}$$

where:

$M_p = \rho_p \frac{\pi}{6} d_p^3$ (2626)	Mass of a single particle
$C_p = \frac{\sqrt{3\theta_p}}{s}$ (2627)	Collision frequency of each particle
$N_p = \frac{1}{a_c}$ (2628)	Number of particles adjacent to unit area
$s = d_p \left[\left(\frac{\alpha_{\max}}{\alpha_p} \right)^{\frac{1}{3}} - 1 \right]$ (2629)	Average distance between the boundary and the surface of an adjacent particle
$a_c = d_p^2 \left(\frac{\alpha_{\max}}{\alpha_p} \right)^{\frac{2}{3}}$ (2630)	Average boundary area per particle
α_p	Particle volume fraction
$\alpha_{\max}$	Maximum packing volume fraction

Substituting for these quantities and simplifying Eqn. (2624) leads to the final form for erosion rate:

$$E_f = \frac{\pi}{6} \rho_p \sqrt{3\theta_p} \frac{\alpha_p g_0}{\alpha_{\max}} e_r \tag{2631}$$

where:

- ρ_p is the particle density
- θ_p is the particle granular temperature
- g_0 is the particle radial distribution function

When using multiple particle phases, α_p is replaced by α_{cumul}, the cumulative particle volume fraction when evaluating the contribution due to each particle phase.

The erosion ratio, e_r, in Eulerian multiphase simulations uses the same erosion correlations as in Lagrangian and DEM multiphase simulations:

- Ahlert Correlation
- DNV Correlation
- Neilson-Gilchrist Correlation
- Oka Correlation

Abrasive Wear Erosion

The abrasive wear rate, E_w, is defined as the mass of wall material eroded due to abrasion per unit area per unit time.

The mass of wall material eroded due to abrasion is given by the Archard formulation as below.

$$m_w = aFs \tag{2632}$$

where:

- a is the abrasive wear coefficient

  The default value of 1E-2 kg/J is the high end of the recommended range of values, 1E-8 to 1E-2, mild wear to severe wear.
- F is the normal force
- s is the sliding distance.

Thus, E_w is:

$$E_w = aPU_s \tag{2633}$$

where:

- P is the normal pressure exerted by particles at the wall
- U_s is their sliding velocity at the wall.

The normal pressure, P, is:

$$P = \dot{m}_p U_n \tag{2634}$$

where:

- $\dot{m}_p$ is the mass flow rate of particles per unit area impacting on the face
- U_n is the normal velocity of particle phase at the wall.

The formulation for $\dot{m}_p$ is described in Eqn. (2625).

Thus the final form of Abrasive Wear rate in Eulerian multiphase is:

$$E_w = a\frac{\pi}{6}\rho_p\sqrt{3\theta_p}\frac{\alpha_{cumul}g_0}{\alpha_{\max}}U_nU_s \tag{2635}$$

Crystallization

A Solute Crystallization model and a Melt Crystallization model are implemented in STAR-CCM+.

Nucleation

The simplest form of spatially homogeneous population balance equation (PBE) reads:

$$\partial_t n = -\partial_L(2Gn) + \dot{N}_0\delta(L - L_0) \tag{2636}$$

where:

- $n(L)$ is the particle size distribution (PSD), the number density of particles of diameter L.
- $G = \frac{1}{2}\partial_t L$ is the growth rate accounting for the phase change at the surface of a particle.
- $\dot{N}_0$ is the nucleation rate.
- L_0 is the nucleus diameter.

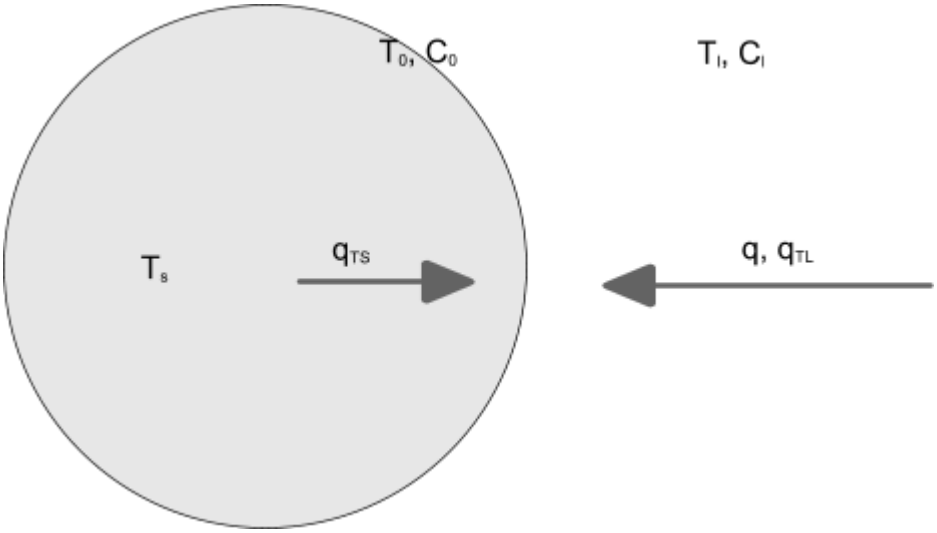
Concentration of the solute in the solution, C_l, is governed by the advection-diffusion-reaction equation:

$$\partial_t\rho_l\alpha_lC_l + \nabla\cdot\rho_l\alpha_l(C_lu_l - D_T\nabla C_l) = -S_c \tag{2637}$$

where the sink term, S_c, is given by integration of the surface growth rate over entire particle-liquid interface and creation of new particles:

$$S_c = \pi\int\rho_s n(L)L^2Gd(L) + \rho_s\dot{N}_0\frac{\pi}{6}L_0^3 \tag{2638}$$

The schematic diagram below shows the mass and energy balance for a particle:



The interior of the particle has a temperature T_s. At the interface, the concentration of the solute is C_0 and its corresponding temperature is T_0. In the liquid, the temperature is T_l and the concentration of the solute is C_l. The mass flux of the solute to the surface, q, is balanced by the crystal growth rate, $\rho_s G$. Heat is transported to the surface from the liquid and from the interior of the particle: the corresponding fluxes are q_{TL} and q_{TS}.

The mass and energy balance read:

$$q = \rho_s G \tag{2639}$$

$$q_{TL} + q_{TS} = -qH_c \tag{2640}$$

where H_c is the enthalpy of crystallization.

Generally speaking, Eqn. (2639) and Eqn. (2640) are appropriate to almost every mass transfer process (for example, boiling, heterogeneous combustion). However, numerical considerations and user convenience require further differentiation of the models.

A distinction can be made between concentration-driven (solute) crystallization and temperature-driven (melt) crystallization (Mullin, [473]).

Solute Crystallization

In a solute crystallization process, a compound (the solute) is dissolved in another liquid (the solvent). When the temperature drops, or an anti-solvent is added, the concentration of the solute becomes higher than the saturation concentration and crystallization starts.

The surface growth rate G is a function of the supersaturation (Mullin, [473]; Borissova, [406]):

$$G = k_g (\Delta C)^{m_g} \tag{2641}$$

where k_g (the prefactor) and m_g (the exponent) are empirical parameters.

The saturation concentration $C_{sat}(T)$ is a function of the temperature. The supersaturation delta $\Delta C = C_0 - C_{sat}$ is the difference between the solute concentration in the liquid film around the particle and the saturation concentration.

Usually, dissolution is a faster process than crystallization (Mullin, [473]), so the parameters of Eqn. (2641) are demarcated:

$$k_g = \begin{cases} k_g^+, \ C_{sat} \geq 0 \\ k_g^-, \ C_{sat} < 0 \end{cases} \tag{2642}$$

$$m_g = \begin{cases} m_g^+, \ C_{sat} \geq 0 \\ m_g^-, \ C_{sat} < 0 \end{cases} \tag{2643}$$

The phase change is balanced by corresponding mass transfer to the film; the mass balance is:

$$q = \rho_s G = \rho_l Sh \frac{D}{L}\left(C_l - C_0\right) \tag{2644}$$

where:

- q is the mass flux
- Sh is the Sherwood number
- D is the coefficient of molecular diffusion
- ρ_s and ρ_l are densities of the solid and liquid phases, respectively.

The energy balance for the film includes both internal and external heat transfer, as follows:

$$Nu \frac{\lambda_l}{L}\left(T_l - T_0\right) = Fo \frac{\lambda_s}{L}\left(T_0 - T_s\right) - qH_c \tag{2645}$$

where:

- Nu is the Nusselt number
- Fo is the Fourier number
- λ_l and T_l are the heat conductivity and temperature of the liquid phase
- λ_s and T_s are the heat conductivity and temperature of the solid phase.

The STAR-CCM+ implementation follows that of other models (for example, evaporation) and assumes that $Fo = \infty$, that is $T_0 = T_s$.

Melt Crystallization

Unlike its solute counterpart, melt crystallization can occur in a single-component liquid (for example, ice formation in water). Admixtures that are dissolved in the liquid are also important. Firstly, admixtures lower the saturation temperature. For example, the freezing temperature of sea water is lower than the freezing temperature of fresh water. Secondly, the composition of the crystals is different from the composition of the melt. For example, melt crystallization processes are important in the food industry and in water desalination (Rahman, [479], Casenave, [410]).

The growth rate is given by the following formula (Mullin, [473]):

$$G = \begin{cases} k_g^+ (\Delta T)^{m_g^+} C_0, & \Delta T \geq 0 \\ k_g^- (\Delta T)^{m_g^-}, & \Delta T < 0 \end{cases} \tag{2646}$$

where $\Delta T = T_0 - T_{sat}$ is temperature supersaturation. As for solute crystallization, the melting (dissolution) is usually faster than the crystal growth, so the parameters of Eqn. (2646) are split into two sets.

The growth depends on the concentration of the freezing component on the surface of the crystal, the melting is concentration-independent because the crystal is assumed to be single-component. An additional simplification is made in the current implementation of the melt crystallization model: it is assumed that $C_0 = C_l$. This assumption (Casenave, [410]) is justified for high mass transfer rates and (or) slow growth rates.

The mass and energy balances for melt crystallization are:

$$q = \rho_s G \tag{2647}$$

$$q_{TS} = Nu \frac{\lambda_l}{L} (T_s - T_l) - qH_c \tag{2648}$$

Armenante-Kirwan Correlations

The standard Ranz-Marshall correlations for the Nusselt number and the Sherwood number assume that the heat and mass transfer are amplified by mean interphase slip velocity. For small particles, the mean slip is small, but experiments suggest that the transfer is enhanced by turbulence. A model for mass transfer to micro-particles has been proposed by Armenante and Kirwan ([400]).

A number of similar models have been reported in the literature (Pangarkar, [476]). However, given the uncertainty in the experimental data, STAR-CCM+ implements a new model with user-controlled parameters:

$$Nu = 2.0 + \alpha \mathrm{Re}_T^{\beta} \mathrm{Pr}^{\gamma} \left(\frac{\Delta \rho}{\rho_l} \right)^{\delta} \tag{2649}$$

$$Sh = 2.0 + \alpha \mathrm{Re}_T^{\beta} Sc^{\gamma} \left(\frac{\Delta \rho}{\rho_l} \right)^{\delta} \tag{2650}$$

where the turbulent Reynolds number, Re_T, is:

$$\mathrm{Re}_T = \frac{\epsilon^{1/3} L^{4/3}}{\nu} \tag{2651}$$

where ϵ is the power input per unit mass of fluid.

The default values of the coefficients are $\alpha = 0.52$, $\beta = 0.52$, $\gamma = 0.333$, $\delta = 0.0$.

These values can be user-specified. See Setting the Sherwood Number and Setting the Nusselt Number.

Rheology of Emulsions and Suspensions

A suspension is a heterogeneous mixture of dispersed solid particles in a liquid, for example, pastes and clays. An emulsion, on the other hand, is a mixture of two or more liquids where one liquid is dispersed in the other such as oil in water. The presence of particles suspended in a Newtonian liquid is known to lead to non-

Newtonian behavior, as the mixture viscosity becomes increasingly dependent on the volume fraction of the dispersed phase.

Krieger and Dougherty [450] developed relative (mixture) viscosity models to describe the non-Newtonian behaviour of suspensions of particles in a liquid. For these models, the relative viscosity depends only upon the volume fraction of the particles. As the concentration of particles increases, an exponential increase in viscosity occurs as the particles move towards contact causing moments of hydrodynamic lubrication forces to become large. Eventually jamming occurs at the critical volume fraction or maximum packing volume fraction. The maximum packing volume fraction is defined as the physically largest volume the particles can occupy when they are packed (usually assumed as close random packing). Maximum packing volume fraction is an empirical parameter. Experiments show that the most compact way to pack hard perfect spheres randomly gives a maximum packing volume fraction of approximately 0.64.

Similar relative viscosity models were also created for emulsions by Pal and Rhodes [475].

In suspension rheology, the dimensionless relative viscosity is used to describe the mixture viscosity. The relative mixture viscosity prescribes how the mixture viscosity increases as more particles interact with each other, along with hydrodynamic forces that are present in the fluid between the particles. As the maximum packing of the particles approaches and the mixture jams, the relative viscosity tends to infinity, essentially encapsulating a yield stress behaviour.

Generally, the relative viscosity is defined as:

$$\eta_r = \frac{\eta}{\mu_c} \tag{2652}$$

where:

- η is the mixture viscosity
- μ_c is the continuous phase (Newtonian) viscosity.

The non-Newtonian behaviour of suspensions and emulsions in terms of the relative viscosity influences the flow through the stress tensor $\mathbf{T}_i$ in the momentum equation Eqn. (2231).

Constitutive Equations

For a **Newtonian fluid**, the stress tensor for a generic phase i is given by (see also Eqn. (1488)):

$$\mathbf{T}_{i,\,\mathrm{Newtonian}} = 2\alpha_i\mu_i\left(\mathbf{D}_i - \frac{1}{3}(\nabla\cdot\mathbf{v}_i)\mathbf{I}\right) \tag{2653}$$

where:

- α_i is the volume fraction
- μ_i is the (Newtonian) viscosity of phase i
- $\mathbf{D}_i$ is the rate-of-deformation tensor.

For a suspension, the **continuous liquid stress tensor** ($i = c$) takes into account an extra stress tensor $\mathbf{T}_{c,\,\mathrm{extra}}$ in addition to the Newtonian stress of the phase. Therefore, the total stress tensor for the continuous phase is:

$$\mathbf{T}_{c,\,\mathrm{total}} = \mathbf{T}_{c,\,\mathrm{Newtonian}} + \mathbf{T}_{c,\,\mathrm{extra}} \tag{2654}$$

The extra stress tensor for the continuous phase is defined as:

$$\mathbf{T}_{c,\,\mathrm{extra}} = 2\alpha_c\mu_c(\eta_r(\phi) - 1)\mathbf{D}_c \tag{2655}$$

where ϕ is the volume fraction of the dispersed phase, that is, of the suspended particles. Note that in suspension rheology, the volume fraction of the dispersed phase is generally denoted by ϕ instead of α_d. Therefore, ϕ is used in all following equations of this chapter. Eqn. (2655) contains a shear stress contribution from the relative viscosity from which one is subtracted to give only the extra-stress due to the presence of the particles in the mixture.

The **dispersed particle phase stress tensor** $(i = d)$ $\mathbf{T}_{d,\,\mathrm{extra}}$ has a split contribution from shear along with the normal stress contribution and the osmotic pressure Π:.

$$\mathbf{T}_{d,\,\mathrm{extra}} = 2\phi\mu_c(\eta_r(\phi) - 1)\mathbf{D}_d - \mu_c\eta_n(\phi)\dot{\gamma}_d\mathbf{Q} - \Pi\mathbf{I} \tag{2656}$$

where:

- η_n is the normal relative viscosity
- $\mathbf{Q}$ is the anisotropy tensor
- $\dot{\gamma}_d$ is the shear-rate of the dispersed phase
- Π is the osmotic pressure.

Generally, emulsions use the same stress equation as suspensions. Since for emulsions, the dispersed phase does not consist of hard spheres as for suspensions, maximum packing can be higher before jamming occurs. Jamming occurs at a critical packing fraction, also called the maximum packing fraction. For emulsions, the relative viscosity inverts at volume fractions above the inversion volume fraction:

$$\eta_r(\phi) \rightarrow \eta_r(1 - \phi) \quad \text{for} \;\; \phi \geq \phi_{inv} \tag{2657}$$

where:

- ϕ is the packing volume fraction
- ϕ_{inv} is the inversion volume fraction.

STAR-CCM+ provides several methods to model the relative viscosity η_r, the normal relative viscosity η_n, and the osmotic pressure Π in Eqn. (2656). One of the earliest from the study of suspensions is the Krieger-Dougherty model [450].

Krieger-Dougherty Model

The Krieger-Dougherty model was based on the assumption that crowding plays a vital role in the origin of non-Newtonian flow behavior in rigid-sphere suspensions. Measurements on latex systems did not detect non-Newtonian behavior at concentrations below 20% by volume of suspended polymer. The Krieger-Dougherty model takes into account the interactions between neighboring spherical particles. The resulting flow equation compares well with experimental viscometric data gathered from tests using synthetic latexes and solutions of polymer particles.

In STAR-CCM+, the Krieger-Dougherty model has the form:

$$\eta_r = \left(1 - \frac{\phi}{\phi_m}\right)^{-[\eta]\phi_m} \tag{2658}$$

where:

- $[\eta]$ is the intrinsic viscosity. For spherical particles, $[\eta] = 2.5$.
- ϕ is the volume fraction.
- ϕ_m is the maximum critical packing fraction. For hard spheres, $\phi_m \simeq 0.645$.

For the Krieger-Dougherty model, the normal relative viscosity is zero. Therefore, the second term in Eqn. (2656) vanishes for this model.

Morris and Boulay Model

The Morris and Boulay model was based on an examination of the role of normal stresses in causing particle migration and macroscopic spatial variation of the particle volume fraction ϕ in a mixture of rigid, neutrally buoyant spherical particles that are suspended in a Newtonian fluid.

The entire ϕ dependence of the compressive shear-induced normal stresses is captured by a normal relative viscosity $\eta_n(\phi)$. This quantity vanishes (as ϕ^2) at $\phi = 0$ and diverges at maximum packing, in the same fashion as does the shear viscosity $\eta_s(\phi)$. Anisotropy of the normal stresses arising from the presence of the particles is modeled as independent of ϕ.

In STAR-CCM+, the relative shear viscosity of the Morris and Boulay model [472] is defined as:

$$\eta_r(\phi) = 1 + 2.5\phi\left(1-\frac{\phi}{\phi_m}\right)^{-1} + K_s\left(\frac{\phi}{\phi_m}\right)^2\left(1-\frac{\phi}{\phi_m}\right)^{-2} \tag{2659}$$

where:

- K_s is the contact contribution
- ϕ_m is the maximum packing fraction.

The Morris and Boulay model was derived from experiments in pipes and Couette devices where the flow becomes aligned in a particular direction that depends on the geometry. The normal stress was shown to be anisotropic. The normal relative viscosity leads to migration of the particles in different directions.

For shear flow in the x direction, the anisotropy tensor reads:

$$Q = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_2 \end{bmatrix} \tag{2660}$$

where the anisotropy parameters are:

- λ_1 = 1.0
- λ_2 = 0.8
- λ_3 = 0.5

An isotropic version gives equal weighting to all directions by setting the anisotropy tensor equal to the identity tensor, [471].

$$Q = I \tag{2661}$$

This form is valid for 3D simulations, but the contributions are no longer anisotropic.

The normal relative viscosity of the Morris and Boulay model is defined as:

$$\eta_n(\phi) = K_n\left(\frac{\phi}{\phi_m}\right)^2\left(1-\frac{\phi}{\phi_m}\right)^{-2} \tag{2662}$$

where:

- K_n is the normal contact contribution
- ϕ_m is the maximum packing fraction.

This is equivalent to a particle pressure. It leads to migration of particles towards regions of low shear-rate.

Shear Thinning Model

The shear thinning model blends a zero-shear and infinite-shear Morris and Boulay relative viscosity model [472] with a Carreau Generalized Newtonian model. The shear thinning model is not experimentally verified. You are advised to use it with caution and only if the physics is known to deviate from the Morris and Boulay model.

The zero-shear relative viscosity is:

$$\eta_{r0}(\phi) = 1 + 2.5\phi\left(1-\frac{\phi}{\phi_{m0}}\right)^{-1} + K_{s0}\left(\frac{\phi}{\phi_{m0}}\right)^2\left(1-\frac{\phi}{\phi_{m0}}\right)^{-2} \tag{2663}$$

where

- K_{s0} is the contact parameter at zero shear rate
- ϕ_{m0} is the maximum packing at zero shear rate.

The infinite-shear relative viscosity is:

$$\eta_{r\infty}(\phi) = 1 + 2.5\phi\left(1-\frac{\phi}{\phi_{m\infty}}\right)^{-1} + K_{s\infty}\left(\frac{\phi}{\phi_{m\infty}}\right)^2\left(1-\frac{\phi}{\phi_{m\infty}}\right)^{-2} \tag{2664}$$

where:

- ϕ is the dispersed phase volume fraction
- $K_{s\infty}$ is the contact parameter at infinite shear rate
- $\phi_{m\infty}$ is the maximum packing at infinite shear rate.

Blended with a Carreau model:

$$\eta_r(\dot{\gamma},\phi) = \frac{\tau_y}{\dot{\gamma}} + \eta_{r\infty}(\phi) + \left(\eta_{r0}(\phi) - \eta_{r\infty}(\phi)\right)\left(1 + (\lambda\dot{\gamma})^a\right)^{\frac{(n-1)}{a}} \tag{2665}$$

where:

- τ_y is the yield stress
- λ is the relaxation time
- $a = 2$ is a parameter to control shear thinning
- n is the power constant.

Following the same approach as for the shear relative viscosity, the Morris and Boulay normal relative viscosity equation is applied to the zero and infinite shear-rate limits:

$$\eta_{n0}(\phi) = K_{n0}\left(\frac{\phi}{\phi_{m0}}\right)^2\left(1-\frac{\phi}{\phi_{m0}}\right)^{-2} \tag{2666}$$

where K_{n0} is the normal contact parameter at zero shear rate.

and:

$$\eta_{n\infty}(\phi) = K_{n\infty}\left(\frac{\phi}{\phi_{m\infty}}\right)^2\left(1-\frac{\phi}{\phi_{m\infty}}\right)^{-2} \tag{2667}$$

where $K_{\mathrm{n}\infty}$ is the normal contact parameter at infinite shear rate.

Again, the normal viscosity limits are blended to give the normal relative viscosity:

$$\eta_n(\dot{\gamma},\phi) = \frac{\tau_y}{\dot{\gamma}} + \eta_{n\infty}(\phi) + \left(\eta_{n0}(\phi) - \eta_{n\infty}(\phi)\right)\left(1+(\lambda\dot{\gamma})^a\right)^{\frac{(n-1)}{a}} \tag{2668}$$

Yield Stress

For **mono-disperse emulsions**, the yield stress in Eqn. (2665) and Eqn. (2668) is [468]:

$$\tau_y = \tau_{y0}\left(\frac{\sigma}{a}\right)\phi_{eff}\left(\phi_{eff}-\phi_m\right)^2 \tag{2669}$$

where:

- τ_{y0} is a constant prefactor. The default value is 0.5, as used in [468].
- σ is the inter-facial tension on the surface of the droplets.
- a is the radius of the droplets.
- ϕ_{eff} is the effective volume fraction, accounting for the presence of the thin film between droplets.
- ϕ_m is the critical packing fraction (for which $\phi_c = 0.645$ for randomly close-packed hard spheres).

Emulsions can have a much larger maximum packing fraction than particle suspensions due to the deformability of liquid droplets.

For **suspensions** of solid particles, the yield stress develops below maximum packing, due to the formation of a network of particles in contact with one another. Several studies [[417]], [456] have revealed a power-law behavior relating yield stress to volume fraction with a variable power index (from 2 to 4). The following model also fits much of the data available for suspension yield stress:

$$\tau_y = \tau_{y0}\left(\frac{\phi}{\phi-\phi_{my}}\right)^2 \tag{2670}$$

where:

- τ_{y0} is a constant prefactor of order unity
- ϕ_{my} is the yield maximum packing.

Osmotic Pressure

Osmotic pressure in Eqn. (2656) is due to the particle-excluded volume and can be derived from the stress due to interparticle forces. Π_o is the osmotic pressure of the particles given by Mewis and Wagner [470]:

$$\Pi_o = nk_B T\left(\frac{1+\phi+\phi^2-\phi^3}{(1-\phi)^3}\right) \tag{2671}$$

where:

- n is the particle density
- k_B is the Boltzmann constant
- T is temperature.

STAR-CCM+ employs the hard sphere osmotic pressure, which defines a step function when the suspended particles come into contact:

$$\Pi_{\text{hs}} = \begin{cases} \Pi_o(\phi), & \text{if } \phi < \phi_m \\ \Pi_{\text{max}}, & \text{if } \phi \geq \phi_m \end{cases} \tag{2672}$$

where $\Pi_o(\phi)$ is given by Eqn. (2671) and Π_{max} is the user-specified maximum osmotic pressure.

When the dispersed phase volume fraction is above maximum packing, the osmotic pressure is given a large maximum osmotic pressure value to represent a hard sphere interaction that repels the particles from packing beyond the maximum packing limit.

For an emulsion the hard sphere repulsion reduces to the inverted osmotic pressure above the inversion volume fraction:

$$\Pi_{\text{hs}} = \begin{cases} \Pi_o(\phi), & \text{if } \phi < \phi_m \\ \Pi_{\text{max}}, & \text{if } \phi_m \leq \phi \leq \phi_i \\ \Pi_o(1-\phi), & \text{if } \phi > \phi_i \end{cases} \tag{2673}$$

Suspension and Emulsion Drag

Drag models for particles in a liquid are functions of the single particle Reynolds number. In turn, the particle Reynolds number is a function of the continuous liquid dynamic viscosity. Numerous experimental studies [412] correlate drag models with parameters of single phase Generalized Newtonian models. STAR-CCM+ employs a modified Schiller-Naumann model where the continuous phase dynamic viscosity μ_c is replaced by $\mu_c \cdot \eta_r$. This combined viscosity reflects the viscosity of the mixture rather than the viscosity of the dilute continous phase.

The inversion volume fraction ϕ_i is defined within the emulsion drag model.

Multiphase Turbulence

STAR-CCM+ models turbulence effects in the continuous and in the dispersed phase.

The presence of a dispersed phase can modify the turbulence of a continuous phase in a number of ways. Shear production is modified through the effect of buoyancy, drag, and other forces on velocity profiles. The

passage of the dispersed phase particles generates velocity fluctuations in the continuous phase. The wake of the dispersed phase particles is a source of turbulence. The interaction of velocity fluctuations and volume fraction fluctuations can generate further turbulence enhancement or damping terms.

For multiphase segregated flow, flow turbulence can be modeled by Reynolds-Averaged Navier Stokes (RANS) turbulence models, such as the $k-\epsilon$ models, the $k-\omega$ models, and the Reynolds stress models.

As an alternative to the RANS turbulence models, for bubbly flow, Large Eddy Simulation (LES) can be used.

Contents:

Continuous Phase Turbulence
Dispersed Phase Turbulence
Turbulence Interaction
Phase Averaging and Closures

Continuous Phase Turbulence

For the continuous phase, STAR-CCM+ solves two modified transport equations for the turbulence quantities. The modifications include the volume fractions of each phase and additional source terms that account for the effect of the dispersed phase on the continuous phase turbulence field.

Multiphase turbulence transport is presented in terms of a generic turbulence scalar equation for multiphase flows:

$$\frac{\partial}{\partial t}\int_V \alpha_i\rho_i\phi_i\,dV + \oint_A \alpha_i\rho_i\phi_i\bar{\mathbf{v}}_i\cdot d\mathbf{a} = \oint_A \alpha_i\left(\mu_i + \frac{\mu_{t,i}}{\sigma_\phi}\right)\nabla\phi_i\cdot d\mathbf{a} + \int_V \alpha_i P_{\phi,i}\,dV - \int_V \alpha_i D_{\phi,i}\,dV \tag{2674}$$
$$+\int_V \alpha_i S_{\phi,i,u}dV + \int_V S_{\phi,c,\mathrm{PIT}}dV + \sum_{i\neq j}\int_V \alpha_i S_{\phi,ij}dV + \int_V \sum_{i\neq j}\left(m_{ij}\phi_j - m_{ji}\phi_i\right)dV$$

where:

- the subscript i denotes the different phases.
- ϕ represents the transported scalar variable. Depending on the turbulence model, ϕ is replaced by the turbulent kinetic energy k, dissipation rate ϵ, specific dissipation rate ω, or the components of the specific Reynolds stress tensor $\mathbf{R}$.
- α_i is the volume fraction of phase i.
- $\overline{v}_i$ is the mean velocity of phase i.
- μ_i is the dynamic viscosity.
- $\mu_{t,i}$ is the turbulent viscosity.
- σ_ϕ is the turbulent Prandtl number.
- $P_{\phi,i}$ is the production term.
- $D_{\phi,i}$ is the dissipation term.
- $S_{\phi,i,u}$ is the user-defined source term.

- $S_{\phi,c,\mathrm{PIT}}$ is the source term that is due to particle induced turbulence. This source term enters only the continuous phase transport equation.
- $S_{\phi,ij}$ is the source term due turbulence transfer between phases. This source term enters the transport equation only if the interphase turbulence model is enabled and only for the pair of phases for which the model is enabled.
- m_{ij} is the mass transfer rate from phase j to phase i.
- m_{ji} is the mass transfer rate from phase i to phase j.

The production and dissipation terms vary according to the selected turbulence model. For more information on RANS turbulence modeling, see also Reynolds-Averaged Navier-Stokes (RANS) Turbulence Models.

These transport equations are always solved for the continuous phase. They are also solved for the dispersed phase when you do not use a turbulent response model.

Dispersed Phase Turbulence

For the dispersed phase, STAR-CCM+ provides two main alternatives for modeling turbulence.

These alternatives are:

- The turbulence of the dispersed phase is correlated to the continuous phase by using semi-empirical models. The correlation is provided by the **response function** that is defined as the ratio of the dispersed phase velocity fluctuation to that of the continuous phase. This influence of the dispersed phase on the continuous phase enters the continuous phase turbulence transport equations in the form of additional source terms.
- Full turbulence models, such as $k-\varepsilon$ model or the Reynolds Stress model, can also be used for the dispersed phase. Running full turbulence models for the dispersed phase can often give a comparable solution to turbulence response models. However, this result can happen for no better reason than similarity of flow conditions with the continuous phase together with a weak dependence of the solution on Reynolds number. Use it either for mixed regimes (continuous and dispersed topology is inverted in different parts of the same flow) or for heavy particles (where a large share of the kinetic energy is held in the dispersed phase).

  Two modified transport equations for the turbulence quantities, similar to the ones for the continuous phase, are solved. The turbulence interaction between the phases is provided through the additional source terms in each equation.

Contents:

Response Models

Response Models

Where there is a clearly-defined dispersed phase, then turbulent response provides a well-defined model for dispersed phase effective turbulent diffusivity and standard wall functions.

The following equations are again presented in terms of the $k-\varepsilon$ model. The turbulent stress is modeled by using the eddy-viscosity concept as given by Eqn. (1759) with the single-phase turbulent viscosity given by:

$$\mu^t = \rho C_\mu \frac{k^2}{\varepsilon} \tag{2675}$$

where:

C_μ is a model coefficient in the $k-\varepsilon$ model. The modified $k-\varepsilon$ equations are solved for the continuous phase and the turbulence of the dispersed phase is correlated to that of the continuous phase. The correlation is provided by the response function C_t:

$$C_t = \frac{|\mathbf{v}_d{}'|}{|\mathbf{v}_c{}'|} \tag{2676}$$

where $\mathbf{v}_d{}'$ is the velocity fluctuation of the dispersed phase and $\mathbf{v}_c{}'$ is the velocity fluctuation of the continuous phase.

If the continuous phase uses the $k-\varepsilon$ model, then the dispersed-phase turbulent kinetic energy k_d is:

$$k_d = C_t^2 k_c \tag{2677}$$

With these definitions, the dispersed phase turbulent eddy viscosity is:

$$\mu_d^t = \frac{\rho_d}{\rho_c} C_t^2 \mu_c^t \tag{2678}$$

where:

- ρ_d is the dispersed phase density
- ρ_c is the continuous phase density

Issa

The Issa model provides C_t for a bubbly flow. It estimates C_t to be up to 3 for very small volume fractions of dispersed phase, but rapidly decreasing to unity for dispersed phase volume fractions greater than about 5%.

The Issa turbulence response model is defined as a correlation for the turbulence response coefficient C_t with a volume fraction correction [485]:

$$C_t(\alpha_d) = 1 + \left(C_t{}^* - 1\right)^{-f(\alpha_d)} \tag{2679}$$

where:

$$f(\alpha_d) = 180\alpha_d - 4.71 \times 10^3 \alpha_d^2 + 4.26 \times 10^4 \alpha_d^3 \tag{2680}$$

$C_t{}^*$ is computed from the Issa model [512]:

$$C_t{}^* = \frac{3+\beta}{1+\beta+2(\rho_d/\rho_c)} \tag{2681}$$

β is defined as:

$$\beta = \frac{2A_{ij}^D l_e^2}{\alpha_d \mu_c Re_t} \tag{2682}$$

and:

$$l_e = C_\mu \frac{k_c^{3/2}}{\varepsilon_c} \tag{2683}$$

$$Re_t = \frac{|\mathbf{v}_c'| l_e}{\mathbf{v}_c} \tag{2684}$$

$$|\mathbf{v}_c'| = \sqrt{\frac{2}{3} k_c} \tag{2685}$$

Tchen

The Tchen model provides a dispersed phase turbulent diffusivity for gas flows laden with heavy particles, as specified by Thai-Van et al. [501]. However, this reference remarks that this model can also be sufficient for bubbly flows because of the weak influence of the eddy diffusivity of the bubbly phase.

From Thai-Van et al. [501], the turbulence diffusivity for turbulent gas flows laden with heavy particles can be modeled as:

$$\nu_d{}^t = \tau_I \frac{1}{3} q_{cd} + \frac{1}{2} \tau_R \frac{2}{3} q_d{}^2 \tag{2686}$$

where:

- q_{cd} is the dispersed-phase average of the product of dispersed and continuous phase velocity fluctuations
- q_d is half the dispersed-phase average of the product of dispersed and dispersed phase velocity fluctuations

For more details on the closure of these correlations, see Tchen Closures.

Turbulence Interaction

Contents:

Particle Induced Turbulence
Turbulence Transfer between Phases

Particle Induced Turbulence

Optionally, STAR-CCM+ takes into account the influence of the dispersed phase on the turbulence of the continuous phase. For the Troshko Hassan, the Gosman, and the Tchen models, this influence enters the modified turbulence transport equations in the form of source terms. The optional models that represent these source terms are called particle-induced turbulence models. The Sato model on the other hand considers the turbulence effects of the dispersed phase on the continuous phase in the form of an enhanced effective viscosity.

$k-\varepsilon$ Turbulence

Troshko Hassan

The Troshko-Hassan model begins by assuming that the whole of the work that is done by the drag force is an unconditionally positive production of continuous-phase pseudo-turbulence. Another feature of this model is that it assumes that this strong turbulence source is dissipated locally using a particle relaxation time scale.

The Troshko and Hassan model describes bubble induced turbulence effects [505] by providing source terms to the continuous K-Epsilon model. It uses the Virtual Mass phase interaction model. If the Virtual Mass phase interaction model is not selected, then the virtual mass coefficient defaults to the value of a spherical bubble:

$$C_{VM} = \frac{1}{2} \tag{2687}$$

The source term is derived by considering that the correlation between velocity fluctuation at the interface and interfacial force density is exactly equal to the work of interfacial force density per unit time. For bubbly flows, the drag force dominates the interfacial force. The original form from Troshko and Hassan [505] is:

$$S_{k_c} = \mathbf{v}_r \cdot F_{ij}^D = \frac{3}{4}\frac{C_D}{l_{cd}}\alpha_d \rho_c |\mathbf{v}_r|^3 \tag{2688}$$

where:

C_D is the bubble drag coefficient.

In STAR-CCM+, Eqn. (2685) is rewritten in terms of the linearized drag coefficient, A_D, to account for multi-particle effects:

$$S_{k_c} = A_D |\mathbf{v}_r|^2 \tag{2689}$$

The dissipation term uses the energy source term that decays with a characteristic time, the Bubble Pseudo-Turbulence Dissipation Relaxation (BPTDR) time, t_b, and is scaled with a calibration constant C_3:

$$S_{\varepsilon_c} = \frac{C_3 S_{k_c}}{t_b} \tag{2690}$$

For the data that was analyzed, Troshko and Hassan found $C_3 = 0.45$. The bubble pseudo-turbulence dissipation relaxation time is:

$$t_b = \frac{2C_{VM}d}{3C_D|\mathbf{v}_r|} \tag{2691}$$

Assuming that $\rho_d \ll \rho_c$ for bubbly flow and again substituting C_D with A_D gives:

$$t_b = \frac{\rho_c C_{VM}\alpha_d}{2A_D} \tag{2692}$$

where C_{VM} is the virtual mass coefficient.

If this model is used with the Reynolds Stress equations, its contribution is assumed to be isotropic:

$$\mathbf{S}_{Rc} = \frac{2}{3}S_{kc}\mathbf{I} \tag{2693}$$

Gosman

The Troshko and Hassan model describes bubble induced turbulence effects [505] by providing source terms to the continuous K-Epsilon model. It uses the Virtual Mass phase interaction model. If the Virtual Mass phase interaction model is not selected, then the virtual mass coefficient defaults to the value of a spherical bubble:

$$C_{VM} = \frac{1}{2} \tag{2694}$$

The source term is derived by considering that the correlation between velocity fluctuation at the interface and interfacial force density is exactly equal to the work of interfacial force density per unit time. For bubbly flows, the drag force dominates the interfacial force. The original form from Troshko and Hassan [505] is:

$$S_{k_c} = \mathbf{v}_r \cdot F_{ij}^D = \frac{3}{4}\frac{C_D}{d}\alpha_d \rho_c |\mathbf{v}_r|^3 \tag{2695}$$

where:

C_D is the bubble drag coefficient.

In STAR-CCM+, Eqn. (2685) is rewritten in terms of the linearized drag coefficient, A_D, to account for multi-particle effects:

$$S_{k_c} = A_D |\mathbf{v}_r|^2 \tag{2696}$$

The dissipation term uses the energy source term that decays with a characteristic time, the Bubble Pseudo-Turbulence Dissipation Relaxation (BPTDR) time, t_b, and is scaled with a calibration constant C_3:

$$S_{\varepsilon_c} = \frac{C_3 S_{k_c}}{t_b} \tag{2697}$$

For the data that was analyzed, Troshko and Hassan found $C_3 = 0.45$. The bubble pseudo-turbulence dissipation relaxation time is:

$$t_b = \frac{2C_{VM}d}{3C_D|\mathbf{v}_r|} \tag{2698}$$

Assuming that $\rho_d \ll \rho_c$ for bubbly flow and again substituting C_D with A_D gives:

$$t_b = \frac{\rho_c C_{VM}\alpha_d}{2A_D} \tag{2699}$$

where C_{VM} is the virtual mass coefficient.

If this model is used with the Reynolds Stress equations, its contribution is assumed to be isotropic:

$$\mathbf{S}_{Rc} = \frac{2}{3}S_{kc}\mathbf{I} \tag{2700}$$

Tchen

Reynolds averaging of continuous-phase velocity fluctuation times the drag source term from the fluctuating part of the continuous-phase momentum equation results in the following source term for the continuous-phase turbulent kinetic energy:

$$S_{kc} = C_0 A_D \left(q_{cd} - 2{q_c}^2 - \mathbf{v}_{TD} \cdot \mathbf{v}_r \right) \tag{2701}$$

where:

- q_{cd} is the dispersed-phase average of the product of dispersed and continuous phase velocity fluctuations.
- q_c is half the dispersed-phase average of the product of continuous and continuous phase velocity fluctuations.
- Constant C_0 is not part of the model derivation, but defaults to 1. You can adjust it to test the sensitivity of the solution to this term.

See Particle Induced Turbulence for the origin of this term, but also keep in mind that:

- Only the contribution of the drag force has been considered in this derivation.
- The closure that is used for the drag force is good under Stokesian conditions but only approximate if ${A_{cd}}^D$ represents a non-Stokesian drag law.

Taken together, $q_{cd} - 2{q_c}^2$ represents an exchange between cross-phase covariance and continuous-phase turbulent kinetic energy.

${A_{cd}}^D \left(\mathbf{v}_{TD} \cdot \mathbf{v}_r \right)$ represents the work that is done by the turbulent dispersion force against mean slip velocity. See Particle Induced Turbulence.

The corresponding contribution to the Reynolds Stress equations is assumed to be isotropic:

$$\mathbf{S}_{Rc} = \frac{2}{3} S_{kc} \mathbf{I} \tag{2702}$$

Following Elghobashi and Abou-Arab [426], the dominant dissipation time scale is assumed to be the same as for shear-induced turbulence:

$$S_{\varepsilon c} = C_{\varepsilon 3} \frac{\varepsilon_c}{k_c} \max\left(S_{kc}, 0 \right) \tag{2703}$$

A negative turbulent energy source term is not allowed to contribute to the turbulent dissipation equation.

Parameter $C_{\varepsilon 3}$ has a default value of 1.44, and you can adjust it independently of constants for other terms.

The correlations q_{cd} and ${q_c}^2$ are closed using Tchen theory as described in Tchen Closures:

$$\begin{gathered} q_{cd} = 2 \left[\frac{b + \eta}{1 + \eta} \right] k_c \\ {q_c}^2 = k_c \end{gathered} \tag{2704}$$

where:

- η is the particle-eddy interaction time, which is scaled by particle relaxation time.
- b is the ratio of the coefficients of the continuous/disperse phase acceleration terms in the particle equation of motion:

$$b = \frac{1 + C_{VM}}{\frac{\rho_d}{\rho_c} + C_{VM}} \tag{2705}$$

After closure, the transfer term $q_{cd} - 2{q_c}^2$ becomes proportional to $(b-1)$.

So for gas-particle flow, ρ_d » ρ_c, where coefficient b is small, the transfer term becomes a sink of continuous phase turbulence kinetic energy.

On the other hand, for bubbly flow, ρ_d « ρ_c, where coefficient b tends to 3/2 if C_{VM} is about 1/2, the transfer term becomes a source of continuous phase turbulence kinetic energy.

Sato

The Sato model is the simplest and earliest model for particle-induced mixing, and is a robust alternative to later source-based models for particle induced turbulence. This model is applicable when modeling a dilute bubbly flow.

In a dilute bubbly flow, the shear-induced turbulence and bubble-induced fluctuations operate at different scales. Figure 11 in Lance and Bataille (1991) [454] gives an example of grid turbulence where bubble-induced fluctuations are dominant but the decay rate of the shear-induced part remains unaffected. This situation is not well-described by the more advanced models that solve transport equations for the combined turbulence energy with one length scale equation. However, the Sato model assumes the shear-induced part is totally independent while the bubble-induced part is always in local equilibrium and does not need a transport equation.

It acts through enhanced effective viscosity of the continuous phase:

$$\nu_{eff} = \nu + \nu_{turb} + \nu_{Sato} \tag{2706}$$

where:

- ν is the continuous phase kinematic viscosity
- ν_{turb} is the turbulent diffusivity
- ν_{Sato} is the Sato bubble-induced viscosity.

The Sato bubble-induced viscosity is given by:

$$\nu_{Sato} = k f_d f_B \alpha_{gas} \frac{d_b}{2} U_{slip} \tag{2707}$$

where:

- k is a model calibration constant
- f_d is the van Driest damping factor
- f_B is a bubble diameter shape correction factor
- α_{gas} is the dispersed phase volume fraction
- d_b is the bubble diameter
- U_{slip} is the slip velocity between phases.

The van Driest [507] damping factor is:

$$f_d = \left(1 - \exp\left(-\frac{y^+}{A^+}\right)\right)^2 \tag{2708}$$

where A^+ is a dimensionless constant.

The bubble diameter shape correction factor is:

$$f_B = \begin{matrix} 4y/d_b(1 - y/d_b) & y < d_b/2 \\ 1 & y > d_b/2 \end{matrix} \tag{2709}$$

where y is the distance from the wall.

This superposition of turbulent and bubble-induced viscosities is partly justified by the measurements of turbulent intensity at different volume fractions (seen in Figure 11 of Lance and Bataille [454]). The effect of volume fraction on the decay of grid-generated turbulence suggests that the effect of bubble-induced fluctuations is purely additive. However, this is not strictly true, particularly in a flow with significant shear production and wall effects. This model should be used with caution, particularly at low Reynolds numbers where the bubble-induced part may be much larger than the shear-induced part.

$k - \omega$ Turbulence

The interaction of dispersed phase particles with turbulence in the continuous phase has traditionally been modeled in terms of the turbulence dissipation rate, ϵ. This is because within the inertial subrange between the largest energy-containing eddies and the smallest viscous eddies, ϵ describes the relationship between length scale and velocity scale, which allows the magnitude of velocity fluctuations around a particle of given size to be estimated.

For the $k - \omega$ model, these ϵ-based models are reused by computing an ϵ field consistent with the $k - \omega$ model. However, different turbulence models can have different interpretations of k and ϵ required to successfully reproduce measured velocity profiles and shear stress, and so may compute different values of k and ϵ, particularly near walls. Therefore it is to be expected that multiphase models that are known to physically depend on ϵ, such as the rate of particle break-up, may need some recalibration when the continuous phase turbulence model is switched between $k - \epsilon$ and $k - \omega$.

Alternative models for Particle Induced Turbulence differ in two aspects, which must be chosen and calibrated consistently. The first is the strength and derivation of the source term for turbulent kinetic energy, S_k. The second is the choice of the dissipation time scale for the corresponding contribution to the turbulent dissipation equation, S_ϵ. The manipulation of these two terms to derive the source S_ω for the ω equation is outlined below.

Using rate of change notation, $\dot{k}$, as a shorthand for the continuous-phase balance equation for k, and so on, the contributions of particle-induced turbulence to the three equations are:

$$\dot{k} = S_k \tag{2710}$$

$$\dot{\epsilon} = S_\epsilon \tag{2711}$$

$$\dot{\omega} = S_\omega \tag{2712}$$

Assuming $\epsilon = \beta^* \omega k$ gives:

$$S_\omega = \omega\left\{\frac{\dot{\epsilon}}{\epsilon} - \frac{\dot{k}}{k}\right\} = \frac{1}{k}\left\{\frac{S_\epsilon}{\beta^*} - \omega S_k\right\} \tag{2713}$$

This result also leads to a useful insight into the behavior of the turbulent timescale, $\frac{k}{\epsilon} = \frac{1}{\beta^* \omega}$, in alternative models whenever the particle-induced turbulence term becomes significant.

Since the Gosman model uses $\frac{k}{\epsilon}$ as the dissipation time scale for particle induced turbulence (neglecting the turbulent dispersion dot product term which is small in fully developed flows), it gives a precisely zero contribution to the ω equation:

$$S_\epsilon = \frac{\epsilon}{k} S_k \tag{2714}$$

$$S_\omega = 0 \tag{2715}$$

The Tchen model starts with the same dissipation timescale, $\frac{k}{\epsilon}$, as Gosman, but with a calibration constant $C_{\epsilon 3}$, so that S_ϵ is not immediately in local equilibrium with $\frac{\epsilon}{k} S_k$.

Then there is an opportunity for particle-induced turbulence to shorten $\frac{k}{\epsilon}$ when using the default calibration $C_{\epsilon 3} = 1.44$ and the net source S_k happens to be positive, as is likely in fully developed flow, and to lengthen it in other circumstances.

$$S_\epsilon = C_{\epsilon 3} \frac{\epsilon}{k} \max(S_k, 0) \tag{2716}$$

$$S_\omega = \frac{\omega}{k} \left\{ C_{\epsilon 3} \max(S_k, 0) - S_k \right\} \tag{2717}$$

The Troshko-Hassan model uses a dissipation time scale t_b proportional to bubble diameter over slip velocity:

$$S_\epsilon = C_{\epsilon 3} \frac{S_k}{t_b} \tag{2718}$$

$$S_\omega = \frac{S_k}{k} \left\{ \frac{C_{\epsilon 3}}{\beta^* t_b} - \omega \right\} \tag{2719}$$

This model gives a clear bubble-related asymptote for the turbulent time scale wherever the bubble-induced term becomes dominant, for example, far from walls:

$$\frac{k}{\epsilon} \rightarrow \frac{t_b}{C_{\epsilon 3}} \tag{2720}$$

Large Eddy Simulation

The Particle Induced Turbulence model contributes to the turbulent viscosity according to:

$$\mu_t = \mu_{SGS} + \mu_{PIT} \tag{2721}$$

Here $\mu_{SGS} = \rho_c \Delta^2 S$ is the SGS contribution to the turbulent viscosity with ρ_c being the density of the continuous phase, Δ the length scale, and S a function of the strain rate tensor. The latter two are defined individually for each of the SGS models as in Eqn. (1980), Eqn. (1984), and Eqn. (1990) for Δ, and Eqn. (1979), Eqn. (1983), and Eqn. (1989) for S.

On the other hand:

$$\mu_{PIT} = \Delta \rho_c \alpha_d |v_r| \tag{2722}$$

is defined as in Lakehal [452], where α_d is the dispersed phase volume fraction and v_r is the phase relative velocity.

Turbulence Transfer between Phases

The interphase turbulence transfer model accounts for turbulence interactions between the continuous and the dispersed phase due to drag. Note that the model is applicable only for a continuous gas phase with a solid particle dispersed phase. The model adds source terms to the turbulent kinetic energy and the dissipation equation for both the continuous and the dispersed phase. The unclosed or modeled term (within the source terms) is the cross-correlation between fluid and particle fluctuating velocities evaluated along the particle trajectory.

These source terms are a result of the work of the drag force on the fluctuating components of velocity and are formulated according to Fox [430].

This model is applicable for gas-particle flows carrying fine particles (that is, Geldart A type particles, having a size between 20 and 100 μm, and a density typically less than 1.4g/cm^3). A typical application for this model is a fluidized catalytic cracking (FCC) system.

The interphase turbulence transfer model is available only when the dispersed phase is a solid particle phase. STAR-CCM+ restricts particle phases to use only the standard $k-\epsilon$ two-layer turbulence model, so this formulation applies only to the scenario where both the continuous phase and the dispersed phase use the $k-\epsilon$ model. S_{kc} and S_{kd} are the source terms for the continuous phase and dispersed phase turbulent kinetic energy equations respectively:

$$S_{kc} = C_0 2A_{cd}\left[\beta_{cd}\sqrt{k_c k_d} - k_c\right] \tag{2723}$$

$$S_{kd} = C_0 2A_{cd}\left[\beta_{cd}\sqrt{k_c k_d} - k_d\right] \tag{2724}$$

$S_{\epsilon c}$ and $S_{\epsilon d}$ are the source terms for the continuous and dispersed phase turbulence dissipation rate equations respectively:

$$S_{\epsilon c} = C_{\epsilon 3} C_0 2A_{cd}\left[\beta_{cd}\sqrt{\epsilon_c \epsilon_d} - \epsilon_c\right] \tag{2725}$$

$$S_{\epsilon d} = C_{\epsilon 3} C_0 2A_{cd}\left[\beta_{cd}\sqrt{\epsilon_c \epsilon_d} - \epsilon_d\right] \tag{2726}$$

where:

- C_0 is a constant that lets you calibrate the model. (This constant is not part of the original model formulation.)
- $C_{\epsilon 3}$ defaults to 1.54.
- A_{cd} is the linearized drag coefficient.
- β_{cd} is the velocity cross-correlation coefficient and defaults to 0.7.

  In STAR-CCM+, β_{cd} can be set as a constant, specified with a field function, or specified with the Zaichik method:

- $k_k = \frac{1}{2}\langle \mathbf{v}_k'' \cdot \mathbf{v}_k'' \rangle_k$

where $\mathbf{v}_k$ is the instantaneous velocity of phase k.

- $\varepsilon_k = \frac{\mu_k}{\rho_k} \langle 2S_k" \cdot S_k" \rangle_k$

  where $S_k" = \frac{1}{2}\left\{\nabla \mathbf{v}_k" + \left(\nabla \mathbf{v}_k"\right)^T\right\}$

Zaichik Method

Zaichik [519] provides a model for the unclosed velocity cross-correlation coefficient term β_{cd} that is valid for arbitrary values of particle-to-fluid density and the ratio between particle size and fluid turbulence length scale:

$$\beta_{cd} = \frac{\left(1 - A\right) f_{u\alpha} + A}{\sqrt{\left(1 - A^2\right) f_{u\alpha} + A^2}} \tag{2727}$$

with:

$$A = \frac{\left(1 + C_{VM}\right)\rho_c}{\rho_d + C_{VM}\rho_c} \tag{2728}$$

and:

$$f_{u\alpha} = \frac{2\Omega_\alpha + Z^2}{2\Omega_\alpha + 2\Omega_\alpha^2 + Z^2} \tag{2729}$$

where:

- C_{VM} is the virtual mass coefficient
- $f_{u\alpha}$ is the response coefficient
- Ω_α is the particle inertia parameter
- Z is the ratio between the Taylor differential and the Lagrangian integral timescales of the continuous phase

The particle inertia parameter is defined as:

$$\Omega_\alpha = \frac{\tau_p}{T_{Lp}} \tag{2730}$$

with:

$$\tau_p = \frac{\alpha_d \rho_d}{A_{cd}}\left(1 + C_{VM}\frac{\rho_c}{\rho_d}\right) \tag{2731}$$

where τ_p is the particle response time. T_{Lp} is the eddy particle interaction timescale and is given by:

$$T_{Lp} = \frac{T_{Lp}^l + 2T_{Lp}^n}{3} \tag{2732}$$

with:

$$T_{Lp}^l = \frac{T_L}{\sqrt{1 + C_\beta \zeta^2}}$$

$$T_{Lp}^{n} = \frac{T_L}{\sqrt{1 + 4C_\beta \zeta^2}}$$

$$T_L = \frac{2\tau_k(\mathrm{Re}_\lambda + C_1)}{\sqrt{15}C_{0\infty}},\ C_{0\infty} = 7,\ C_1 = 32$$

where:

- T_{Lp}^{l} and T_{Lp}^{n} are the Lagrangian integral timescales of fluid velocities viewed by particles, respectively, in the longitudinal and normal directions to the relative velocity vector
- T_L is the Lagrangian integral timescale
- ζ is the particle slip velocity, scaled by turbulent fluctuation velocity
- C_β is a calibration coefficient for the crossing-trajectories effect. Following Deutsch and Simonin [419], it defaults to 0.45.

The scaled particle slip velocity is given by:

$$\zeta = \frac{|v_r|}{\sqrt{\frac{2}{3}k_c}}$$

where v_r is the relative velocity between phases.

The ratio between the Taylor differential and the Lagrangian integral timescales of the continuous phase is:

$$Z = \frac{\tau_T}{T_L}$$

with:

$$\tau_T = \tau_k\left(\frac{2\mathrm{Re}_\lambda}{\sqrt{15}a_0}\right)^{1/2}$$

where:

- τ_T is the Taylor differential timescale
- τ_k is the Kolmogorov timescale
- Re_λ is the Taylor-scale Reynolds number

The Kolmogorov timescale is defined as:

$$\tau_k = \sqrt{\frac{\nu_c}{\epsilon_c}}$$

The Taylor-scale Reynolds number is given by:

$$\mathrm{Re}_\lambda = \sqrt{\frac{20k_c^2}{3\nu_c\epsilon_c}}$$

where ν_c is

$$a_0 = \frac{a_{01} + a_{0\infty}\mathrm{Re}_\lambda}{a_{02} + \mathrm{Re}_\lambda},\ a_{01} = 11,\ a_{02} = 205,\ a_{0\infty} = 7$$

Phase Averaging and Closures

The use of phase averaging generally reduces the number of closures that are needed to model turbulent multiphase flow. In particular, using phase-averaged velocity for convection fluxes eliminates the need to include a turbulent diffusion term in phase mass conservation equations.

$$\phi = \overline{\phi} + \phi' = \langle \phi \rangle_k + \phi'' \tag{2733}$$

where:

- $\overline{\phi}$ is the Reynolds average of the instantaneous variable ϕ.
- $\langle \phi \rangle_k$ is the phase average of the instantaneous variable ϕ formed by a Reynolds average that is weighted with the volume fraction for phase k:

$$\langle \phi \rangle_k = \frac{\overline{\alpha_k \phi}}{\overline{\alpha_k}} \tag{2734}$$

where α_k is the instantaneous volume fraction of phase k.

Elsewhere in the User Guide, α_k is used for the Reynolds average volume fraction, $\overline{\alpha_k}$.

The following Eulerian multiphase turbulence equations are derived in terms of phase-averaged turbulence stresses, kinetic energy, and dissipation [435]:

$$\mathbf{R}_k = \langle \mathbf{v}_k'' \mathbf{v}_k'' \rangle_k \tag{2735}$$

$$k_k = \frac{1}{2} \langle \mathbf{v}_k'' \cdot \mathbf{v}_k'' \rangle_k \tag{2736}$$

$$\varepsilon_k = \frac{\mu_k}{\rho_k} \langle 2S_k'' \cdot S_k'' \rangle_k \tag{2737}$$

$$S_k'' = \frac{1}{2}\left\{\nabla \mathbf{v}_k'' + \left(\nabla \mathbf{v}_k''\right)^T\right\} \tag{2738}$$

where $\mathbf{v}_k$ is the instantaneous velocity of phase k.

Elsewhere in the User Guide, $\mathbf{v}_k$ is used for the phase average velocity, $\langle \mathbf{v}_k \rangle_k$.

Boussinesq Closure

Using the eddy-diffusivity or Boussinesq hypothesis, the Reynolds flux of any scalar φ due to turbulent velocity fluctuation $\mathbf{v}'_c$ in the continuous phase can be modeled as:

$$\overline{\phi' \mathbf{v}'_c} = \overline{\phi' \mathbf{v}_c} = \overline{\phi \mathbf{v}'_c} \approx -\frac{\nu_c^t}{\sigma_\phi} \nabla \overline{\phi} \tag{2739}$$

where σ_ϕ is the ratio of turbulent diffusivities of momentum and ϕ.

Tchen Closures

Thai-Van et al. [501] summarises closures for in-phase and cross-phase turbulent diffusivities that are based on the work of by Simonin and others (for example see Deutch and Simonin [420]) in extending the results of Tchen theory for small particles in steady homogeneous turbulence (for example see Hinze [438]).

These closures are used in STAR-CCM+ for the Tchen options for turbulent dispersion force, for dispersed-phase turbulence, and for particle induced turbulence. They provide models for the following turbulent diffusivities:

Continuous-phase turbulent diffusion:

$$\nu_c{}^t = \frac{2}{3} q_c{}^2 \tau_T \tag{2740}$$

Interphase turbulent dispersion:

$$\mathbf{D}_{cd} = \tau_I \frac{1}{3} q_{cd} \mathbf{I} \tag{2741}$$

Dispersed phase turbulent diffusion:

$$\nu_d{}^t = \tau_I \frac{1}{3} q_{cd} + \frac{1}{2} \tau_R \frac{2}{3} q_d{}^2 \tag{2742}$$

constructed using these correlations:

- $q_c{}^2 = \frac{1}{2} \langle \mathbf{v}_c{}'' \cdot \mathbf{v}_c{}'' \rangle_c$
- $q_{cd} = \langle \mathbf{v}_c{}'' \cdot \mathbf{v}_d{}'' \rangle_d$
- $q_d{}^2 = \frac{1}{2} \langle \mathbf{v}_d{}'' \cdot \mathbf{v}_d{}'' \rangle_d$

and these time scales:

- τ_T is the eddy turnover time.
- τ_I is the particle-eddy interaction time.
- τ_R is the particle relaxation time.

According to Thai-Van et al., Eqn. (2742) was originally derived for turbulent gas flows laden with heavy particles. They commented that an eddy-viscosity closure for a bubbly phase could also be derived but had a negligible effect on predictions.

If a turbulence model is used for the continuous phase, then $q_c{}^2$ is readily available as the continuous-phase turbulent kinetic energy k_c, and $\nu_c{}^t$ corresponds to the turbulent diffusivity provided by a Standard K-Epsilon turbulence model.

From Thai-Van et al., the other correlations are estimated using:

$$q_{cd} = 2 q_c{}^2 \left[\frac{b + \eta}{1 + \eta} \right] \tag{2743}$$

$$q_d{}^2 = q_c{}^2 \left[\frac{b^2 + \eta}{1 + \eta} \right] \tag{2744}$$

$$\tau_I = \frac{\tau_T}{\sigma_0}\frac{1}{\sqrt{1 + C_\beta \xi^2}} \tag{2745}$$

$$\tau_T = \frac{3}{2}C_\mu \frac{k_c}{\varepsilon_c} \tag{2746}$$

$$\tau_R = \tau_D\left(1 + \frac{\rho_c}{\rho_d}C_{VM}\right) \tag{2747}$$

where:

- τ_D is a basic particle timescale. It is used here more generally to represent the inverse of the linearized drag coefficient. For more information on this coefficient, see Contribution of Drag to Turbulent Dispersion.
- b is the ratio of the coefficients of the continuous/dispersed phase acceleration terms in the particle equation of motion:

$$b = \frac{1 + C_{VM}}{\frac{\rho_d}{\rho_c} + C_{VM}} \tag{2748}$$

- C_{VM} is a virtual mass coefficient for particle response, with a default value of 0.5 if a virtual mass force model has not been specified for the mean flow.
- η is the particle-eddy interaction time, which is scaled by particle relaxation time:

$$\eta = \frac{\tau_I}{\tau_R} \tag{2749}$$

- C_β is a calibration coefficient for the cross-trajectories effect. Following Deutsch and Simonin [420], choose it as 0.45 for the direction parallel to the mean relative velocity, and as 1.8 for orthogonal directions. The default value is 1.8.
- ξ is the particle slip velocity, scaled by turbulent fluctuation velocity:

$$\xi = \frac{|\mathbf{v}_r|}{\sqrt{\frac{2}{3}k_c}} \tag{2750}$$

Although Thai-Van et al. include a correction for fluctuations in this slip velocity estimate, this correction is neglected in STAR-CCM+.

Contribution of Drag to Turbulent Dispersion

Under Stokesian (viscous drag) conditions, the instantaneous drag force on a solid spherical particle of fixed diameter d and density ρ_d, due to velocity relative to a fluid of viscosity μ_c, is:

$$\frac{\rho_d \alpha_d}{\tau_D}(\mathbf{v}_d - \mathbf{v}_c) \tag{2751}$$

$$\tau_D = \frac{\rho_d d^2}{18\mu_c} = \frac{d^2}{18\nu_c}\left(\frac{\rho_d}{\rho_c}\right) \tag{2752}$$

The Reynolds average of the instantaneous drag force can be written as the sum of a mean drag force term and a turbulent dispersion term:

$$\overline{\frac{\rho_d \alpha_d}{\tau_D}\left(\mathbf{v}_d - \mathbf{v}_c\right)} = \frac{\rho_d}{\tau_D}\overline{\alpha_d}\left(\mathbf{v}_r + \mathbf{v}_{TD}\right) \tag{2753}$$

$$\mathbf{v}_r = \langle \mathbf{v}_d \rangle_d - \langle \mathbf{v}_c \rangle_c \tag{2754}$$

$$\mathbf{v}_{TD} = \langle \mathbf{v}_c \rangle_c - \langle \mathbf{v}_c \rangle_d \tag{2755}$$

Using the Boussinesq closure to model the last term:

$$\mathbf{v}_{TD} = \frac{\overline{\alpha_c \mathbf{v}_c}}{\overline{\alpha_c}} - \frac{\overline{\alpha_d \mathbf{v}_c}}{\overline{\alpha_d}} = \frac{\overline{\alpha_c \mathbf{v}_c{}'}}{\overline{\alpha_c}} - \frac{\overline{\alpha_d \mathbf{v}_c{}'}}{\overline{\alpha_d}} \approx -\frac{\nu_c{}^t}{\sigma_\alpha}\left(\frac{\nabla\overline{\alpha_c}}{\overline{\alpha_c}} - \frac{\nabla\overline{\alpha_d}}{\overline{\alpha_d}}\right) \tag{2756}$$

For non-Stokesian drag laws, and for other interfacial area models, the result of Reynolds averaging no longer has a straightforward closure, but the above result is still used as an approximation.

For small particles, it is possible to compensate partly for the non-Stokesian drag laws. This compensation is done by including a fluctuating contribution to the mean particle relative velocity used in the drag coefficient calculation (for example see Thai-Van et al.). However, this practice is not followed here, since more generally it is difficult to say whether particle-eddy interaction reduces or enhances mean slip [403].

For the general case, replace the Stokesian coefficient

$$\frac{\rho_d}{\tau_D}\overline{\alpha_d} \tag{2757}$$

with the mean linearized drag coefficient $A_{cd}{}^D$ evaluated using a mean slip velocity $\mathbf{v}_r$ and mean interfacial area a_{cd}''':

$$A_{cd}{}^D = \frac{a_{cd}'''}{8}\rho_c C_D \left|\mathbf{v}_r\right| \tag{2758}$$

to give these mean drag and turbulent dispersion forces:

$$\mathbf{F}_{cd}{}^D = A_{cd}{}^D \mathbf{v}_r \tag{2759}$$

$$\mathbf{F}_{cd}{}^{TD} = A_{cd}{}^D \mathbf{v}_{TD} \tag{2760}$$

This form of turbulent dispersion force and its closure has been known for some time. For example, see Deutsch and Simonin [420] or Gosman et al. [435]. It has also recently been reconciled with alternative approaches in Drew [423] and Burns et al. [409].

Contribution of Virtual Mass to Turbulent Dispersion

The instantaneous virtual mass force reads:

$$\rho_c C_{VM} \alpha_d \left(\mathbf{a}_d - \mathbf{a}_c\right) \tag{2761}$$

where $\mathbf{a}$ is the acceleration of the corresponding phase. Averaging the above equation yields:

$$\rho_c C_{VM} \overline{\alpha_d} \left(\langle \mathbf{a}_d \rangle_d - \langle \mathbf{a}_c \rangle_d \right) = \rho_c C_{VM} \overline{\alpha_d} \left(\langle \mathbf{a}_d \rangle_d - \langle \mathbf{a}_c \rangle_c \right) + \rho_c C_{VM} \overline{\alpha_d} \left(\frac{\langle \alpha_c' \mathbf{a}_c \rangle}{\overline{\alpha_c}} - \frac{\langle \alpha_d' \mathbf{a}_c \rangle}{\overline{\alpha_d}} \right) \tag{2762}$$

The first term is for the inertia of the mean flow. The second one contributes to the turbulent dispersion force. It can be modeled using the Boussinesq closure, assuming particle relaxation time τ_R as an estimate of the acceleration time scale:

$$\frac{\langle \alpha_c' \mathbf{a}_c \rangle}{\overline{\alpha_c}} - \frac{\langle \alpha_d' \mathbf{a}_c \rangle}{\overline{\alpha_d}} \approx -\frac{\nu_c^{\ t}}{\tau_R \sigma_\alpha} \left(\frac{\nabla \overline{\alpha_c}}{\overline{\alpha_c}} - \frac{\nabla \overline{\alpha_d}}{\overline{\alpha_d}} \right) \tag{2763}$$

$$\tau_R = \frac{\rho_d \overline{\alpha_d}}{A_{cd}^{\ D}} \left(1 + \frac{\rho_c}{\rho_d} C_{VM} \right) \tag{2764}$$

Combining the contributions from drag and virtual mass terms:

$$A_{cd}^{\ D} \mathbf{v}_{TD} - \rho_c C_{VM} \overline{\alpha_d} \frac{\nu_c^{\ t}}{\tau_R \sigma_\alpha} \left(\frac{\nabla \overline{\alpha_c}}{\overline{\alpha_c}} - \frac{\nabla \overline{\alpha_d}}{\overline{\alpha_d}} \right) \Bigg) = \left(1 + \frac{\rho_c C_{VM}}{\rho_d + \rho_c C_{VM}} \right) A_{cd}^{\ D} \mathbf{v}_{TD} \tag{2765}$$

For bubbly flows ($\rho_c \gg \rho_d$), including a contribution from the virtual mass can double the overall turbulent dispersion coefficient.

While virtual mass is always important in modeling particle response in a turbulent flow, the modeling of the mean virtual mass force is optional, as not all applications include an accelerating mean flow.

Particle Induced Turbulence

This section outlines the derivation of the source term for the continuous-phase fluctuation kinetic energy equation. Out of all possible interphase forces, the drag force is assumed to make the dominant contribution to this term.

Phase averaging of the kinetic energy of the instantaneous flow gives the following composition for the kinetic energy of the velocity fluctuations in the continuous-phase:

$$k_c = \frac{1}{2} \langle \mathbf{v}_c'' \cdot \mathbf{v}_c'' \rangle_c = \frac{1}{2} \left\{ \langle \mathbf{v}_c \cdot \mathbf{v}_c \rangle_c - \langle \mathbf{v}_c \rangle_c \cdot \langle \mathbf{v}_c \rangle_c \right\} \tag{2766}$$

where:

- $\frac{1}{2} \langle \mathbf{v}_c'' \cdot \mathbf{v}_c'' \rangle_c$ is the mean kinetic energy held in the instantaneous flow. The balance equation can be derived by Reynolds averaging the scalar product of the instantaneous phase velocity $\mathbf{v}_c$ with the instantaneous phase momentum equation.
- $\frac{1}{2} \langle \mathbf{v}_c \rangle_c \cdot \langle \mathbf{v}_c \rangle_c$ is the kinetic energy that is held in the mean flow. The equation can be derived from the scalar product of the phase mean velocity $\langle \mathbf{v}_c \rangle_c$ with the Reynolds averaged phase momentum equation.

The equation for k_c can be derived by subtracting these two equations. Then an instantaneous force term, assumed proportional to slip velocity $\mathbf{v}_d - \mathbf{v}_c$, and to disperse phase volume fraction α_d,

$$\mathbf{f}_c = \frac{\rho_d \alpha_d}{\tau_D}\left(\mathbf{v}_d - \mathbf{v}_c\right) \tag{2767}$$

makes the following contribution to the equation for k_c:

$$S_{kc} = \overline{\mathbf{v}_c \cdot \mathbf{f}_c} - \langle \mathbf{v}_c \rangle \cdot \overline{\mathbf{f}_c} = \overline{\mathbf{v}_c" \cdot \mathbf{f}_c} \tag{2768}$$

$$= \frac{\rho_d}{\tau_D}\overline{\alpha_d}\left\{ \langle \mathbf{v}_c" \cdot \mathbf{v}_d" \rangle_d - \langle \mathbf{v}_c" \cdot \mathbf{v}_c" \rangle_d + \langle \mathbf{v}_c" \rangle_d \cdot \left[\langle \mathbf{v}_d \rangle_d - \langle \mathbf{v}_c \rangle_c \right] \right\} \tag{2769}$$

$$= A_{cd}{}^D \left\{ \langle \mathbf{v}_c" \cdot \mathbf{v}_d" \rangle_d - \langle \mathbf{v}_c" \cdot \mathbf{v}_c" \rangle_d - \mathbf{v}_{TD} \cdot \mathbf{v}_r \right\} \tag{2770}$$

The final results are obtained by using the drag linearization coefficient $A_{cd}{}^D$ and dispersion velocity $\mathbf{v}_{TD}$ from the earlier section Contribution of Drag to Turbulent Dispersion, together with this identity:

$$\mathbf{v}_{TD} = \langle \mathbf{v}_c \rangle_c - \langle \mathbf{v}_c \rangle_d = \frac{\overline{\left[\langle \mathbf{v}_c \rangle_c - \mathbf{v}_c \right] \alpha_d}}{\overline{\alpha_d}} = - \langle \mathbf{v}_c" \rangle_d \tag{2771}$$

S_{kc} can now be modeled either with the help of a turbulent response model or with the help of the Tchen closures, when taken together with this further approximation:

$$\langle \mathbf{v}_c \cdot \mathbf{v}_c \rangle_d \approx \langle \mathbf{v}_c \cdot \mathbf{v}_c \rangle_c = 2k_c \tag{2772}$$

The first two terms in Eqn. (2770) represent an equilibrium or transfer between the covariance of dispersed-continuous velocity fluctuation and the turbulent kinetic energy of the continuous phase.

The last term can be viewed as the dot product of turbulent dispersion force with the mean slip velocity (or alternatively, of the mean drag force with the turbulent dispersion velocity). Its role depends on the alignment of these two vectors.

The dot product can often be small or zero in a fully developed and converged pipe flow. In a horizontal pipe flow case, this small value is because the magnitude of the slip velocity is likely to be small under fully developed conditions. In a vertical pipe flow case, it is because although the slip is not small, it is orthogonal to any gradients of volume fraction under fully developed conditions.

However, the term can be positive or negative, and large, under developing conditions or during convergence. It makes a positive contribution to continuous phase turbulent kinetic energy when mean drag force and turbulent dispersion force are acting in opposite directions. For example, it contributes when particles are falling, or when bubbles are rising, against their own concentration gradients.

Boundary Conditions

Thermal

For Eulerian multiphase flows, it is assumed that the same local wall temperature T_w is experienced by all phases in contact with the wall. This useful closure, together with an energy balance at the wall, allows the most common thermal boundary conditions to be specified for the wall, rather than for each individual phase.

The energy balance for the wall is:

$$\dot{q}''_w = \sum_i \dot{q}''_{wi} + \sum_{(ij)} \dot{q}''_{w(ij)} \tag{2773}$$

where:

- $\dot{q}''_w$ is the external heat flux from the environment to the wall
- $\dot{q}''_{wi}$ is the heat flux from the wall to the phase i fluid next to the wall
- $\dot{q}''_{w(ij)}$ is the heat flux from the wall to a phase interaction (ij) interface next to the wall

External Heat Flux

The four specifications available for modeling the external heat flux q''_w are as follows:

- **Adiabatic**: $\dot{q}''_w$ is zero and T_w is computed.
- **Heat Flux**: $\dot{q}''_w$ is specified and T_w is computed.
- **Temperature**: T_w is specified and $\dot{q}''_w$ is computed.
- **Convection**: Ambient temperature T_∞ and external heat transfer coefficient h_∞ are specified, with both q''_w and T_w computed.

  The external heat flux for the convection option is:

$$\dot{q}''_w = h_\infty \left(T_\infty - T_w\right) \tag{2774}$$

Phase Heat Flux

The standard model for internal heat flux from each phase i to the wall $\dot{q}''_{wi}$ is based on single-phase wall treatments within each phase together with an assumption that each phase has a local wall contact area proportional to the volume fraction next to the wall.

For turbulent flow, these two models combine into:

$$\dot{q}''_{wi} = \alpha_i \frac{\rho_i c_{pi} u_i^*}{t_i^+} \left(T_w - T_i\right) \tag{2775}$$

where:

- α_i is the volume fraction of phase i
- u_i^* is the turbulent wall velocity scale of phase i
- t_i^+ is defined by the thermal wall law for phase i

For laminar flow, this relationship is:

$$\dot{q}''_{wi} = \frac{-\dot{q}''_{wi} \cdot \mathbf{a}}{|\mathbf{a}|} = \frac{\alpha_i k_i \nabla T_i \cdot \mathbf{a}}{|\mathbf{a}|} \tag{2776}$$

where $\mathbf{a}$ is the outward facing wall face area vector.

You can also override the standard relationship for phase heat flux, by specifying three coefficients A_i, B_i and C_i:

$$-\dot{q}''_{wi} = A_i + B_i T_i + C_i T_w \tag{2777}$$

This allows various opportunities for alternative heat flux relationships, wall contact models, or linearization methods.

Note: The sign convention, which is based on the discretization of fluxes using outward facing areas. Since either side of this formula is defined to be positive when heat flows from the fluid to the wall, the coefficient of the fluid temperature, B_i, is always positive or zero. The coefficient of the wall temperature, C_i, is always negative or zero.

The internal coefficients from single-phase wall treatments are available as field functions for constructing multiphase coefficients. For example, the standard turbulent or laminar wall contact model is equivalent to specifying the coefficients:

$$\begin{aligned} A_i &= \alpha_i A_{INTERNAL,i} \\ B_i &= \alpha_i B_{INTERNAL,i} \\ C_i &= \alpha_i C_{INTERNAL,i} \end{aligned} \tag{2778}$$

Phase Interaction Heat Flux

This term is absent from the wall heat balance, unless the phase interaction (ij) includes phase change, in which case this heat flux drives the rate of mass transfer at the wall by:

$$\dot{q}''_{w(ij)} = -\dot{m}''_{w(ij)} h_{(ij)} \tag{2779}$$

where:

- $\dot{m}''_{w(ij)}$ is the rate of interphase mass transfer from phase j to phase i per unit wall area
- $h_{(ij)}$ is the enthalpy of phase change from phase i to phase j.

Standard relationships for phase interaction heat fluxes are described in the Wall Boiling section.

Override a standard relationship for phase interaction heat flux, by specifying three coefficients $A_{(ij)}$, $B_{(ij)}$, and $C_{(ij)}$:

$$-\dot{q}''_{w(ij)} = A_{(ij)} + B_{(ij)} T_{(ij)} + C_{(ij)} T_w \tag{2780}$$

where:

- $T_{(ij)}$ is the temperature of the interface between phases i and j, which is typically specified as the saturation temperature

For example, an evaporation heat flux that depends on some power of the wall superheat, such as:

$$\dot{q}''_{w(ij)} = K \max\left(T_w - T_{(ij)}, 0\right)^n \tag{2781}$$

Can be linearized by differentiation with respect to each temperature:

$$\dot{q}''_{w(ij)}\left(T_{(ij)}, T_w\right) = \dot{q}''_{w(ij)}\left(T_{(ij)}{}^*, T_w{}^*\right) + \left(T_{(ij)} - T_{(ij)}{}^*\right)\left[\frac{\partial}{\partial T_{(ij)}} \dot{q}''_{w(ij)}\right]^* + \left(T_w - T_w{}^*\right)\left[\frac{\partial}{\partial T_w} \dot{q}''_{w(ij)}\right]^* \tag{2782}$$

where an asterisk signifies a value at the previous iteration.

Collecting terms in $T_{(ij)}$ and T_w leads to the implementation:

$$\begin{aligned} A_{ij} &= (n-1)K \max\left(T_w{}^* - T_{(ij)}{}^*, 0\right)^n \\ B_{ij} &= -nK \max\left(T_w{}^* - T_{(ij)}{}^*, 0\right)^{n-1} \\ C_{ij} &= -B_{(ij)} \end{aligned} \tag{2783}$$

Permeability

A wall boundary that has been set to phase permeable for a particular phase allows a phase mass flux approaching the boundary to leave the system at this boundary.

The mass flux for phase i is given by:

$$\dot{m}_i = \rho_i \mathbf{v}_i \cdot \mathbf{a} \alpha_i \gamma_i \tag{2784}$$

where:

- m_i is the mass flux leaving the system
- ρ_i is the cell-centered density
- $\mathbf{v}_i$ is the cell-centred velocity
- α_i is the cell-centered volume fraction
- $\mathbf{a}$ is the cell wall area vector (normal to the wall in the outward facing direction)
- γ_i is the wall permeability factor

The wall permeability takes a value from 0 through 1 to control the amount of the flux incident on the surface that is removed from the flow. Any component of the wall velocity parallel to the wall is the same as for a standard wall (zero in this case for a no-slip wall). When a permeable wall phase condition is activated, the mass flux that is calculated at the wall cell is used in all the other transport equations.

Multiphase Segregated Flow Bibliography

[398] *Antal, S. P., Lahey, R. T., and Flaherty, J. E. 1991. “Analysis of phase distribution in fully developed laminar bubbly two-phase flow”, Int. J. Multiphase Flow, 17, pp. 635 – 652.*

[399] *Arastoopour, H., Pakdel, P., and Adewumi, M. 1990. “Hydrodynamic analysis of dilute gas-solids flow in a vertical pipe”, Powder Technology, 62, pp. 163 – 170.*

[400] *Armenante, P. M., Kirwan, D. J., 1989. "Mass transfer to microparticles in agitated systems", Chemical Engineering Science, 44, pp. 2781 – 2796.*

[401] *Atapattu, D.D., Chhabra, R.P., and Uhlherr, P.H.T. 1995. “Creeping sphere motion in Herschel-Bulkley fluids: Flow field and drag”, Journal of Non-Newtonian Fluid Mechanics, 59, pp. 245 – 265.*

[402] *Auton, T.R., Hunt, J.C.R., and Prud’homme, M. 1988. “The force exerted on a body in inviscid unsteady non-uniform rotational flow”, J. Fluid Mech., 197, pp. 241 – 257.*

[403] *Balachandar S. and Eaton J. K. 2010. “Turbulent dispersed multiphase flow”, Annu. Rev. Fluid Mech., 42, pp 111 – 133.*

[404] *Bartolomei G.C. and Chanturiya V.M. 1967. “Experimental study of true void fraction when boiling subcooled water in vertical tubes”, Thermal Engineering, 14, pp 123 – 128.*

[405] *Benyahia, S., Syamlal, M. and O’Brien, T.J. Accesses Jan 2012. “Summary of MFIX equations 2012-1”, from URL https://mfix.netl.doe.gov/documentation/MFIXEquations2012-1.pdf*

[406] *Borissova, A., 2009. "General systems modeling of multi-phase batch crystallization from solution", Chemical Engineering and Processing: Process Intensification 48, 268 – 278.*

[407] *Bouillard, J., Lyczkowski, R., and Gidaspow, D. 1989. “Porosity Distributions in a Fluidized Bed with an Immersed Obstacle”, AIChE J 35(6), pp. 908 – 922.*

[408] *Bozzano, G., and Dente, M. 2001. “Shape and terminal velocity of single bubble motion: a novel approach”, Computers and Chemical Engineering, 25, pp. 571 – 576.*

[409] *Burns, A. D., Frank, T., Hamill, I., and Shi, J-M. 2004. “The Favre averaged drag model for turbulent dispersion in Eulerian multi-phase flows”, technical report, ICMF.*

[410] *Casenave, C., Dochain, D., Alvarez, G., Arellano, M., Benkhelifa, H., Leducq, D., 2014. "Model identification and reduction for the control of an ice cream crystallization process", Chemical Engineering Science 119, 274 - 287.*

[411] *Cerne, G., Petelinb, S., Tiselja, I., 2001, "Coupling of the Interface Tracking and the Two-Fluid Models for the Simulation of Incompressible Two-Phase Flow" Journal of Computational Physics, Volume 171, Issue 2, pp 776 – 804*

[412] *Chhabra, R.P., 1992. “Bubbles, Drops and Particles in Non-Newtonian Fluids”, CRC Press.*

[413] *Cole R. 1960. “A photographic study of pool boiling in the region of the critical heat flux”, AIChE J., 6, pp. 533 – 542.*

[414] *Cook, T.L. and Harlow, F.L. 1984. “Virtual mass in multiphase flow”, Int. J. Multiphase Flow, 10, pp. 691 – 696.*

[415] *Coste, P. 2013, "A Large Interface Model for Two-Phase CFD", Nuclear Engineering and Design, 255, pp. 38 – 50.*

[416] *Coulaloglou, C.A. and Tavlarides, L.L., 1977, "Description of interaction processes in agitated liquid-liquid dispersions", Chemical Engineering Science, 32, pp. 1289 – 1297.*

[417] *Darbouret, M., Cournil, M. and Herri, J. M. 2005. "Rheological study of an hydrate slurry for air conditioning application", Fifth International Conference on Gas Hydrates (ICGH 5), Trondheim, Norway, page 2005.*

[418] *Del Valle M.V.H. and Kenning D.B.R. 1985. “Subcooled flow boiling at high heat flux”, Int. J. Heat Mass Transfer, 28, pp. 1907 – 1920.*

[419] *Deutsch, E., and Simonin, O. 1991. "Large eddy simulation applied to the modelling of particulate transport coefficients in turbulent two-phase flows", Proc. of the Eighth Symposium on Turbulent Shear Flows, Munich, Germany, pp 10-1-1 to 10-1-6.*

[420] *Deutsch,E., and Simonin, O. 1991. "Large eddy simulation applied to the motion of particles in stationary homogeneous fluid turbulence", ASME FED, 110:35 – 42.*

[421] *Di Felice, R. 1994. "The voidage functions for fluid-particle interaction system" Intl. J. Multiphase Flow, 20(1):153 – 159.*

[422] *Ding, J. and Gidaspow, D. 1990. "A bubbling fluidisation model using kinetic theory of granular flow", AIChE Journal, 36(4), pp. 523 – 538.*

[423] *Drew, D. A., 2001. "A turbulent dispersion model for particles or bubbles", J. Engineering Mathematics, 41:259 – 274, 2001.*

[424] *Drew, D.A., 1983. "Mathematical Modeling Of Two-Phase Flow", Annual Review Of Fluid Mechanics, 15, pp. 261-291.*

[425] *Drew, D. A. and Passman, S. "Theory Of Multicomponent Fluids", Springer, New York, 1998.*

[426] *Elghobashi, S. E., and Abou-Arab, T. W., 1983. "A two-equation turbulence model for two-phase flows", Phys. Fluids, 26(4):931 – 938.*

[427] *Eskin, D., Taylor, S. D., and Yang, D., 2017. "Modeling of droplet dispersion in a turbulent Taylor-Couette flow", Chemical Engineering Science 161, pp. 36 – 47.*

[428] *Esmaili, E. and Mahinpey, N. 2011. "Adjustment of drag coefficient correlations in three dimensional cfd simulation of gas-solid bubbling fluidized bed", Advances in Engineering Software, 42, pp. 375 – 386.*

[429] *Fox, R. O., 2003. "Computational models for turbulent reacting flows" Cambridge University Press.*

[430] *Fox, R.O. 2014. "On multiphase turbulence models for collisional fluid-particle flows", J. Fluid Mech. Vol. 742. pp 368 – 424.*

[431] *Fox, R. "On the relationship between Lagrangian micromixing models and computational fluid dynamics" Chemical Engineering and Processing. 37, 521535.*

[432] *Frick, A.M. and Thompson, B.E. 1982. "Settling Velocities in Batch Sedimentation", Oak Ridge National Laboratory, ORNL/MIT358*

[433] *D. Gera, M. Syamlal, and T.J. O'Brien, 2002. Int. J. Multiphase flow, 30(4):419 – 428.*

[434] *Gidaspow, G. 1994. "Multiphase Flow and Fluidization - Continuum and Kinetic Theory Descriptions", Academic Press.*

[435] *Gosman, A.D., Issa, R.I., Lekakou, C., Looney, M.K., and Politis, S. 1992. "Multidimensional modelling of turbulent two-phase flows in stirred vessels", AIChE Journal, 38(12), pp. 1946 – 1956.*

[436] *Hibiki T. and Ishii M., 2003. “Active nucleation site number density in boiling systems”, Int. J. Heat Mass Transfer, 46, pp. 2587 – 2601.*

[437] *Hill, D.P. 1998. “The Computer Simulation of Dispersed Two Phase Flows”, Ph.D. dissertation, Imperial College, London.*

[438] *Hinze, J. O., 1959. Turbulence.*

[439] *Hughes E. D. and Duffey R. B., 1991. "Direct contact condensation and momentum-transfer in turbulent separated flows", Int. J. Multiphase Flow, 17, 599 – 619.*

[440] *Ishii, M. and Mishima, K. 1984. “Two-fluid model and hydrodynamic constitutive relations”, Nuclear Engineering Design, 82, pp. 107 – 126.*

[441] *Ishii, M., "Thermo-Fluid Dynamic, Theory Of Two-Phase", Eyrolles, 1975.*

[442] *Ishii, M. and Zuber, N. 1979. “Drag Coefficient and Relative Velocity in Bubbly, Droplet and Particulate Flows,” AiChe. Journal, 25, pp. 843 – 855.*

[443] *Issa, R. I., and Oliveira, P. J. 1994. “Numerical prediction of phase separation in two-phase flow through T-junctions”, Computers Fluids, 23(2), pp. 347 – 372.*

[444] *Johnson, P.C., and Jackson, R. 1987. “Frictional-collisional constitutive relations for granular materials, with application to plane shearing”, J. Fluid Mech., 176, pp. 67 – 93.*

[445] *Kamp, A., Chesters, A., Colin, C., and Fabre, J., 2001. "Bubble coalescence in turbulent flows: A mechanistic model for turbulence-induced coalescence applied to microgravity bubbly pipe flow", International Journal of Multiphase Flow, 27, pp. 1363 – 1396.*

[446] *Karnik, A., Ajay, M.R., and Tandon, M. 2013. "Numerical investigation of the hydrodynamics of cylindrical fluidized bed" The 14th Intl. Conf. on Fluidization - From Fundamentals to Products*

[447] *Kocamustafaogullari G., 1983. “Pressure dependence of bubble departure diameter for water”, Int. Comm. Heat Mass Transfer, 10, pp. 501 – 509.*

[448] *Koch, D. L. 1990. "Kinetic Theory for a Monodisperse Gas-Solid Suspension" Phys. Fluids A, 2, 1711.*

[449] *Koehler, S. A., Hilgenfeldt, S., Stone, H. A., 2000. "A generalized view of foam drainage: experiment and theory", Langmuir 16, 6327 – 6341.*

[450] *Krieger, I.M., and Dougherty, T.J. 1959. “A mechanism for non-Newtonian flow in suspensions of rigid spheres” Transactions of the Society of Rheology, 3, pp. 137 – 152.*

[451] *Kurul, N. and Podowski, M.Z. 1990. “Multidimensional effects in sub-cooled boiling”, Proceedings of the Ninth Heat Transfer Conference, Jerusalem.*

[452] *Lakehal, M., Smith. 2002. "Large-eddy simulation of bubbly turbulent shear flows" Journal of Turbulence, 3(25).*

[453] *Lamb, H., 1945. Hydrodynamics, Dover Publications, New York, NY, USA.*

[454] *Lance, M., and Bateille, J., 1991. “Turbulence in the liquid phase of a uniform bubbly air-water flow”, J. Fluid Mech., 222, pp. 95 – 118.*

[455] *Lemmert, M. and Chawla, J.M. 1977. “Influence of flow velocity on surface boiling heat transfer coefficient”, Heat Transfer in Boiling, E. Hahne and U. Grigull, Eds., Academic Press and Hemisphere, New York, NY, USA.*

[456] *Leong, Y. K., Scales, P. J., Healy, T. W. and Boger, D. V. 1995. "Effect of particle size on colloidal zirconia rheology at the isoelectric point", J. Am. Ceram. Soc., 78(8):2209.*

[457] *Levich, V. G., 1962. "Physiochemical hydrodynamics", Prentice-Hall.*

[458] *Liao, Y., and Lucas, D., 2010. "A literature review on mechanisms and models for the coalescence process of fluid particles", Chemical Engineering Science, 65, pp. 2851 –2864.*

[459] *Lo S., 1996. "Application of the MUSIG model to bubbly flows", AEAT-1096, AEA Technology, 230: 8216 – 8246.*

[460] *Lo, S. and Rao, P. 2007. “Modelling of droplet breakup and coalescence in an oil-water pipeline”, 6th Int. Conf. Multiphase Flow, ICMF 2007, Leipzig, Germany.*

[461] *Lo, S., and Tomasello, A., 2010, "Recent progress in CFD modelling of multiphase flow in horizontal and near-horizontal pipes", 7th North America Multiphase Technology, Banff, Canada.*

[462] *Lo, S. and Zhang, D. 2009. “Modelling of Break-up and Coalescence in Bubbly Two-Phase Flows”, J. Comp. Multiphase Flows, 1(1), pp. 23 – 38.*

[463] *Lockett, M.J. and Kirkpatrick R.D. 1975. “Ideal Bubbly Flow and Actual Flow in Bubble Columns”, Trans. Inst. Chem. Engrs, 53, pp. 267 – 273.*

[464] *Louge, M. Y., Mastorakos, E., and Jenkins, J. T., 1991. "The Role of Particle Collisions in Pneumatic Transport" J. Fluid Mech., 231, 345.*

[465] *Lun, C.K.K. Savage, D.J. Jefferey, S.B. and Chepurniy, N. 1984. “Kinetic theories for granular flow: inelastic particles in Couette flow and slightly inelastic particles in a general flowfield”, Journal of Fluid Mechanics, 140, pp. 223 – 256.*

[466] *Luo, H., 1993. "Coalescence, breakup and liquid circulation in bubble column reactors", PhD thesis, Norges Tekniske Hoegskole, Trondheim.*

[467] *Martinez-Bazan, C., Rodriguez-Rodriguez, J., Deane, G. B., Montanes, J. L., Lasheras, J. C., 2010. "Considerations on bubble fragmentation models" J. Fluid Mech., 661, pp 159 – 177.*

[468] *Mason, T. G. 1999. "New fundamental concepts in emulsion rheology", Current Opinion in Colloid and Interface Science, 4:231 – 238.*

[469] *Menter, F.R., and Egorov, Y., 2004 "Re-visiting the turbulent scale equation", Proc. IUTAM Symp. One Hundred Years of Boundary Layer Research, Göttingen, Springer.*

[470] *Mewis, J., and Wagner, N. J. 2012. "Colloidal Suspension Rheology", Cambridge University Press.*

[471] *Miller, R.M., Singh, J.P., and Morris, J.F. 2009. "Suspension flow modeling for general geometries", Chemical Engineering Science, 64 (22) pp. 4597 – 4610*

[472] *Morris, J.F. and Boulay, F. 1999. "Curvilinear flows of noncolloidal suspensions: The role of normal stresses", Journal of Rheology, 43, p. 1213.*

[473] *Mullin, J., 2001. "Crystallization" (Fourth Edition). Butterworth-Heinemann, Oxford.*

[474] *C. W. Hirt and B. D. Nichols. 1981. "Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries", J. Comput. Phys., 39, pp. 201-225.*

[475] *Pal, R. and Rhodes, E. 1989. "Viscosity/concentration relationships for emulsions", J. Rheol., 33, p. 1021.*

[476] *Pangarkar, V. G., Yawalkar, A. A., Sharma, M. M., Beenackers, A. A. C. M., 2002. "Particle-liquid mass transfer coefficient in two-/three-phase stirred tank reactors", Industrial & Engineering Chemistry Research 41, 4141 - 4167.*

[477] *Podowski, M.Z. and Podowski, R.M. 2009. “Mechanistic Multidimensional Modeling of Forced Convection Boiling Heat Transfer”, Science and Technology of Nuclear Installations, Article ID 387020, 10 pages.*

[478] *Politis, S., Issa, R. I., Looney, M. K., Gosman, A. D., Lekakou, C. 1992. “Multidimensional modeling of turbulent two-phase flows in stirred vessels“, AIChE Journal, 38(12), pp. 1946 – 1956.*

[479] *Rahman, M. S., Ahmed, M., Chen, X. D., 2006. "Freezing-melting process and desalination: I. Review of the state-of-the-art", Separation & Purification Reviews 35, 59 – 96.*

[480] *Ramkrishna, D. 2000. "Population balances".Academic Press.*

[481] *Ranz, W.E., and Marshall, W.R. 1952. “Evaporation from drops--Part I and II”, Chemical Engineering Progress, 48(3), p. 141*

[482] *Richards, L. A., 1931. "Capillary conduction of liquids through porous mediums", Journal of Applied Physics 1, 318 – 333.*

[483] *Richardson, J, and Zaki, W. 1954. “Sedimentation and Fluidisation: Part 1”, Trans. Inst. Chem. Engrs, 32, pp. 35 – 53.*

[484] *Rusche, H., 2002. Computational fluid dynamics of dispersed two-phase flows at high phase fractions", Technical report, PhD Thesis, Imperial College, University of London.*

[485] *Rusche, H., Behzadi, A. and Issa, R. 2004. “Modelling of dispersed bubble and droplet flow at high phase fractions”, Chemical Engineering Science, 59, pp. 759 – 770.*

[486] *Rusche, H., and Issa, R. 2000. "The effect of voidage on the drag force on particles, droplets and bubbles in dispersed two-phase flow", Japanese European Two-phase Flow Meeting, Tsukuba, Japan.*

[487] *Samuelsberg, A. and Hjertager, B.H. 1996. “Computational Modeling of Gas/Particle Flow in a Riser”, AIChE J, 42(6), p. 1536.*

[488] *Sato, Sadatomi, Sekoguchi. 1981. “Momentum and heat transfer in two-phase bubble flow-I”, Int Journal Multiphase Flow, 7, pp. 167 – 177.*

[489] *Sato, Sadatomi, Sekoguchi. 1981. “Momentum and heat transfer in two-phase bubble flow-II”, Int Journal Multiphase Flow, 7, pp. 169 – 190.*

[490] *Schaeffer, G. 1987. “Instability in the evolution equations describing incompressible granular flow”, J. Diff. Equ., 66, pp. 19 – 50.*

[491] *Schiller, L., and Naumann, A. 1933. “Uber die grundlegenden Berechnungen bei der Schwerkraftaufbereitung”, VDI Zeits, 77(12), pp. 318 – 320.*

[492] *Simonnet M., Gentric, G., Olms, E. and Midoux, N. 2007. “Experimentation determination of the drag coefficient in a swarm of bubbles”, Chem. Eng. Sci., 62, pp. 858 – 866.*

[493] *Sirignano, W. A. 1999. "Fluid dynamics and transport of droplets and sprays", Cambridge University Press, Cambridge*

[494] *Spalding, D. B. 1960. "A standard formulation of the steady convective mass transfer problem", Int. J. Heat Mass Transfer, 1, 192 – 207*

[495] *Strubelj, L., Tiselj, I., and Mavko, B. 2009. "Simulations of free surface flows with implementation of surface tension and interface sharpening in the two-fluid model" International Journal of Heat and Fluid Flow, 30 (4), pp. 741 – 750.*

[496] *Strubelj, L., and Tiselj, I. 2010, "Two-fluid model with interface sharpening", International Journal for Numerical Methods in Engineering Vol 85, Issue 5.*

[497] *Sussman, M., Samereka, P. and Osher, S. 1994. "A level set approach for computing solutions to incompressible two-phase flows", J. Comput. Phys. 114, pp. 146-159.*

[498] *Syamlal, M. and O’Brien, T.J. April 1987. "The derivation of a drag coefficient formula from velocity-voidage correlations" Technical Note, U.S. Department of energy, Office of Fossil Energy, NETL, Morgantown, WV.*

[499] *Taghipour, F. Ellis, N. and Wong, C. 2005. "Experimental and computational study of gas-solid fluidized bed hydrodynamics" Chemical Engineering Science, 60(24):6857 – 6867.*

[500] *Taylor R., Krishna R., “Multicomponent Mass Transfer”, Wiley, 1993*

[501] *Thai-Van, D., Minier, J. P., Simonin, O., Freydier, P., and Olive, J., 1994. "Multidimensional two-fluid model computation of turbulent dispersed two-phase flows", ASME FED, 185:277 – 291.*

[502] *Tolubinsky V.I. and Kostanchuk D.M. 1970. “Vapour bubbles growth rate and heat transfer intensity at subcooled water boiling”, Heat Transfer 1970, Preprints of papers presented at the 4th International Heat Transfer Conference, Paris, 5, Paper No. B-2.8*

[503] *Tomiyama, A., Kataoka, I., Zun, I., and Sakaguchi, T. 1998. “Drag Coefficients of single bubbles under normal and micro gravity conditions”, JSME International Journal, Series B, 41(2), pp. 472 – 479.*

[504] *Tomiyama, A., Tamai, H., Zun, I., and Hosokawa, S. 2002. “Transverse migration of single bubbles in simple shear flows”, Chemical Engineering Science, 57, pp. 1849 – 1858.*

[505] *Troshko, A. A. and Hassan, Y. A. 2001. “A two-equation turbulence model of turbulent bubbly flows”, International Journal of Multiphase Flow, 27 pp. 1965 – 2000.*

[506] *Tsouris, C. and Tavlarides, L. L. 1994. "Breakage and coalescence model for drops in turbulent dispersions", AIChE Journal, 49 pp. 395 – 406.*

[507] *van Driest. 1955. “On Turbulent Flow Near a Wall”, presented at the 1955 Heat Transfer and Fluid Mechanics Institute, Los Angeles, June 23 – 25, 1955.*

[508] *van Wachem, B., Schouten, J.C., Krishna, R. and van den Bleek, C.M. 1998. “Eulerian simulations of bubbling behaviour in gas-solid fluidised beds”, Computers Chem. Eng., 22, Suppl. pp. S299 – S306.*

[509] *van Wachem, B., Schouten, J.C., van den Bleek, C.M., Krishna, R. and Sinclair, J. L. 2001. “Comparative analysis of CFD models of dense gas-solid systems”, AIChE Journal, 47(5) pp. 1035 – 1051.*

[510] *Vikhansky, A. and Splawski, A. 2015. "Adaptive multiply size group method for CFD-population balance modelling of polydisperse flows", The Canadian Journal of Chemical Engineering, Volume 93, Issue 8, Pages: 1327 – 1334.*

[511] *Wallis, G. B., "One-Dimensional Two-Phase Flow", Mcgraw-Hill, New York, 1969.*

[512] *Wang, D.M. 1994. “Modelling of bubbly flow in a sudden pipe expansion”, Technical Report II-34, BRITE/EuRam Project BE-4098.*

[513] *Wei, Y. and Morel, C. 2004. “Volumetric interfacial area prediction in upward bubbly two-phase flow”, Int. J. Heat and Mass Transfer, 47, pp. 307 – 328.*

[514] *Weisman J. and Pei B.S. 1983. “Prediction of critical heat flux in flow boiling at low qualities”, Int. J. Heat Mass Transfer, 26, No. 10, pp. 1463 – 1477*

[515] *Wen, C.Y. and Yu, Y.H. 1966. "Mechanics of fluidization" Chemical Engg. Prog. Symposium Series, 62(2):100 – 111.*

[516] *Witt, P.J. and Perry, J.H. 1996. “A Study in Multiphase Modelling of Fluidised Beds”, Computational Techniques and Applications: CTAC95, R. May & A. Easton Eds., World Scientific, pp. 787 – 794.*

[517] *Woo, X. Y., Tan, R. B. H., Braatz, R. D., 2009. "Modeling and computational fluid dynamics-population balance equation-micromixing simulation of impinging jet crystallizers", Crystal Growth & Design 9, 156-164.*

[518] *Yu, A. B., and Standish, N. 1987. “Porosity calculations of multi-component mixtures of spherical particles”, Powder Technology, 52, pp. 233 – 241.*

[519] *Zaichik, L.I., Simonin, O., and Alipchenkov, V.M. 2010. "Turbulent collision rates of arbitrary-density particles", Intl. J. Heat and Mass Transfer. 53. pp. 1613 – 1620.*

[520] *Zuber, N. 1964. “On the dispersed two-phase flow in the laminar flow regime”, Chemical Engineering Science, 19, pp. 897 – 917.*

Volume of Fluid Method

The Volume of Fluid (VOF) multiphase model implementation in STAR-CCM+ belongs to the family of interface-capturing methods that predict the distribution and the movement of the interface of immiscible phases. This modeling approach assumes that the mesh resolution is sufficient to resolve the position and the shape of the interface between the phases.

The distribution of phases and the position of the interface are described by the fields of phase volume fraction α_i. The volume fraction of phase i is defined as:

$$\alpha_i = \frac{V_i}{V} \tag{2785}$$

where V_i is the volume of phase i in the cell and V is the volume of the cell. The volume fractions of all phases in a cell must sum up to one:

$$\sum_{i=1}^{N} \alpha_i = 1 \tag{2786}$$

where N is the total number of phases.

Depending on the value of the volume fraction, the presence of different phases or fluids in a cell can be distinguished:

- $\alpha_i = 0$—the cell is completely void of phase i
- $\alpha_i = 1$—the cell is completely filled with phase i
- $0 < \alpha_i < 1$—values between the two limits indicate the presence of an interface between phases

The material properties that are calculated in the cells containing the interface depend on the material properties of the constituent fluids. The fluids that are present in the same interface-containing cell are treated as a mixture:

$$\rho = \sum_i \rho_i \alpha_i \tag{2787}$$

$$\mu = \sum_i \mu_i \alpha_i \tag{2788}$$

$$C_p = \sum_i \frac{\left(C_p\right)_i \rho_i}{\rho} \alpha_i \tag{2789}$$

where ρ_i is the density, μ_i is the dynamic viscosity, and $\left(C_p\right)_i$ is the specific heat of phase i.

Volume Fraction Transport Equation

The distribution of phase i is driven by the phase mass conservation equation:

$$\frac{\partial}{\partial t} \int_V \alpha_i dV + \oint_A \alpha_i \mathbf{v} \cdot d\mathbf{a} = \int_V \left(S_{\alpha_i} - \frac{\alpha_i}{\rho_i} \frac{D\rho_i}{Dt} \right) dV \tag{2790}$$

where $\mathbf{a}$ is the surface area vector, $\mathbf{v}$ is the velocity, S_{α_i} is a user-defined source term of phase i, and $D\rho_i/Dt$ is the material or Lagrangian derivative of the phase densities ρ_i.

Continuity Equation

The total mass conservation equation for all phases is given by:

$$\frac{\partial}{\partial t}\left(\int_V \rho dV\right) + \oint_A \rho \mathrm{v} \cdot d\mathbf{a} = \int_V S dV \tag{2791}$$

where S is a mass source term that is related to the phase source term as follows:

$$S = \sum_i S_{\alpha_i} \cdot \rho_i \tag{2792}$$

The dependency on the volume fractions of the constituent phases of the fluid mixture is accounted for through the density, which is given by Eqn. (2787).

Momentum Equation

$$\frac{\partial}{\partial t}\left(\int_V \rho \mathbf{v} dV\right) + \oint_A \rho \mathbf{v} \otimes \mathbf{v} \cdot d\mathbf{a} = -\oint_A p\mathbf{I} \cdot d\mathbf{a} + \oint_A \mathbf{T} \cdot d\mathbf{a} + \int_V \mathbf{f}_b dV \tag{2793}$$

where:

- p is the pressure
- $\mathbf{I}$ is the unity tensor
- $\mathbf{T}$ is the stress tensor
- $\mathbf{f}_b$ is the vector of body forces

Energy Equation

$$\frac{\partial}{\partial t}\int_V \rho E dV + \oint_A \left[\rho \mathrm{H} \mathbf{v} + p\right] \cdot d\mathbf{a} = -\oint_A \dot{\mathbf{q}}'' \cdot d\mathbf{a} + \oint_A \mathbf{T} \cdot \mathbf{v} d\mathbf{a} + \int_V \mathbf{f}_b \cdot \mathbf{v} dV + \int_V S_E dV \tag{2794}$$

where:

- E is the total energy
- H is the total enthalpy
- $\dot{\mathbf{q}}''$ is the heat flux vector
- S_E is a user-defined energy source term

Contents:

High-Resolution Interface Capturing (HRIC)
Surface Tension
Free Surface Waves
Evaporation and Condensation
Boiling
Cavitation and Gas Dissolution
Flash Boiling
Melting and Solidification

High-Resolution Interface Capturing (HRIC)

An important quality of a system of immiscible phases (for example, air and water) is that the fluids always remain separated by a sharp interface. The High-Resolution Interface Capturing (HRIC) scheme is designed to mimic the convective transport of immiscible fluid components, resulting in a scheme that is suited for tracking sharp interfaces.

A simple higher-order scheme (for example, central differencing scheme (CDS) or second-order upwind scheme (SOU)) would fail in approximating large spatial variations of phase volume fractions, which are best represented by the Heaviside unit step function.

The HRIC scheme is based on the normalized variable diagram (NVD). The normalized face value ξ_f is calculated as:

$$\xi_f = \begin{cases} \xi_C & \text{if} \quad \xi_C < 0 \\ 2\xi_C & \text{if} \quad 0 \le \xi_C < 0.5 \\ 1 & \text{if} \quad 0.5 \le \xi_C < 1 \\ \xi_C & \text{if} \quad 1 \le \xi_C \end{cases} \tag{2795}$$

The calculated value of ξ_f is further corrected according to the local Courant number:

$$\mathrm{Co} = \frac{\mathbf{v} \cdot \mathbf{a}_f}{V_{P_C}} \delta t \tag{2796}$$

This correction is to take into account the availability criterion. This criterion says that the amount of one fluid that is convected across a cell face during a time-step is always less than or equal to the amount available in the donor cell. This correction is made according to the expressions:

$$\xi_f{}^* = \begin{cases} \xi_f & \text{if} \quad Co < Co_l \\ \xi_C + \left(\xi_f - \xi_C\right) \dfrac{Co_u - Co}{Co_u - Co_l} & \text{if} \quad Co_l \le Co < Co_u \\ \xi_C & \text{if} \quad Co_u \le Co \end{cases} \tag{2797}$$

and it plays a role only in transient simulations. Default values for Co_l and Co_u are 0.5 and 1. They are introduced to control blending of HRIC and FOU schemes depending on the Courant number. For values of $Co < Co_l$ HRIC is used, for $Co_l < Co < Co_u$ a blend of HRIC and FOU is used, and for $Co_u < Co$ FOU is used.

The blending is introduced in the case when a large time variation of the free surface shape is present, and the time-step is too large to resolve details of it. It brings stability and robustness to the scheme. In this case, smaller values of Co_u and Co_l help to promote convergence. Smaller values activate FOU sooner, and the calculation is more stable. At the same time, the interface is more “smudged” in regions with large Co numbers.

For problems which have a steady-state solution, the values of Co_u and Co_l must be large, such that HRIC is used irrespective of the time-step selected. Any number larger than the maximum CFL is appropriate.

The front sharpening aspect of the downwind scheme is just what is needed when the interface is perpendicular to the flow. However, when the interface is parallel to the flow direction, the downwind scheme

tends to wrinkle it. The final correction of ξ_f is based on the angle θ between the normal to the interface $\mathbf{n}_i$ and the cell-face surface vector $\mathbf{a}_f$, as shown below:

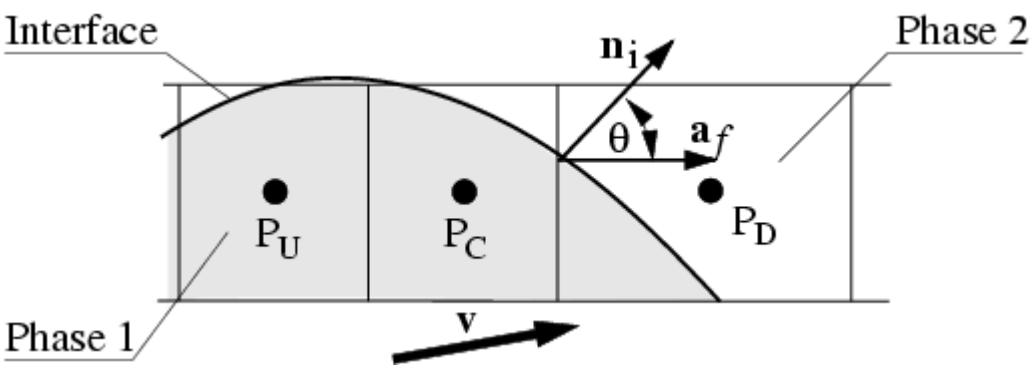
$$\xi_f{}^{**} = \xi_f{}^{*}\left(\cos\theta\right)^{C_\theta} + \xi_C\left(1-\left(\cos\theta\right)^{C_\theta}\right) \tag{2798}$$

The default value of the angle factor C_θ is 0.05. If the free surface is not smooth and not following the grid lines, increase its value.

The cell-face value ξ_f is now calculated as:

$$\alpha_f^{HRIC} = \xi_f{}^{**}\left(\alpha_D - \alpha_U\right) + \alpha_U \tag{2799}$$

The following diagram shows the interface between two fluids and the notation used.



Surface Tension

The surface tension force is a tensile force tangential to the interface separating two fluids. It works to keep the fluid molecules at the free surface in contact with the rest of the fluid.

The interfacial surface force is modeled as a volumetric force using the continuum surface force (CSF) approach of Brackbill and others [522].

The magnitude of the surface tension force depends mainly on the nature of the fluid pair and on temperature. For a curved interface, the surface tension force $\mathbf{f}_\sigma$ can be resolved into two components:

$$\mathbf{f}_\sigma = \mathbf{f}_{\sigma,n} + \mathbf{f}_{\sigma,t} \tag{2800}$$

where:

$$\mathbf{f}_{\sigma,n} = \sigma K \mathbf{n} \tag{2801}$$

and:

$$\mathbf{f}_{\sigma,t} = \frac{\partial \sigma}{\partial t} t \tag{2802}$$

where:

- σ is the surface tension coefficient.
- $\mathbf{n}$ is the unit vector normal to the free surface and directed from liquid to gas.
- t is the unit vector in the tangential direction to the free surface.
- K is the mean curvature of the free surface.

The surface tension force is calculated according to the continuum surface force (CSF) model. This model uses the smooth field of the phase volume fraction α_i to calculate a vector normal to the interface:

$$\mathbf{n} = \nabla\alpha_i \tag{2803}$$

The curvature of the interface can therefore be expressed in terms of the divergence of the unit normal vector $\mathbf{n}$:

$$\kappa = -\nabla\cdot\frac{\nabla\alpha_i}{|\nabla\alpha_i|} \tag{2804}$$

Now the normal component of the surface tension force $\mathbf{f}_{\sigma,n}$ can be expressed as:

$$\mathbf{f}_{\sigma,n} = -\sigma\,\nabla\cdot\left(\frac{\nabla\alpha_i}{|\nabla\alpha_i|}\right)\nabla\alpha_i \tag{2805}$$

When the surface tension coefficient varies along the surface, which can be due to temperature differences, the tangential part does not vanish. In this case, Marangoni or Bénard convection can develop tangential to the free surface. The tangential force is evaluated as:

$$\mathbf{f}_{\sigma,t} = \left(\nabla\sigma\right)_t|\nabla\alpha_i| \tag{2806}$$

where $(\nabla\sigma)_t$ is the gradient of the surface tension coefficient in the tangential direction.

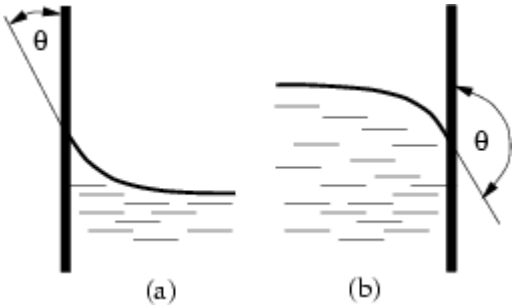
Contents:

Contact Angle

Contact Angle

The contact angle θ describes the influence of a solid wall on the surface tension force.

The contact angle is measured at the triple line, which is the line where the wall and both fluids are in contact. In STAR-CCM+, for an interaction between phases, you set one fluid to be the primary phase and the other as the secondary phase. Values of contact angle smaller than $90°$ mean that the primary phase is wetting the wall, as illustrated in part (a) of the following diagram:



When contact angle also depends on the triple line velocity, it is called dynamic contact angle. Some simple models are based on a step function that uses a single advancing and receding angle, while other models use a smoothed step function. These models work well for inertia-dominated flows because the dynamic contact angle does not depend strongly on the velocity.

STAR-CCM+ implements the Kistler correlation, which works well for capillary-dominated flows.

The dimensionless capillary number Ca is defined as:

$$Ca = \frac{V\mu}{\sigma} \tag{2807}$$

where:

- V is the triple line characteristic velocity
- μ is the dynamic viscosity of the primary phase (usually a liquid phase)
- σ is the surface tension force

The triple line velocity is defined as:

$$V = -\left(\mathbf{V} \cdot \hat{\mathbf{n}}_t\right) \tag{2808}$$

where:

- $\mathbf{V}$ is the relative velocity of the fluid and the corresponding wall at the triple line.
- $\hat{\mathbf{n}}_t$ is the unit vector in the tangential direction pointing in the same direction as the volume fraction gradient of the primary phase ($\nabla\alpha$)

The dynamic contact angle models attempt to account for the dependency of the advancing and receding contact angles on the triple line velocity.

Kistler Correlation

STAR-CCM+ implements the Kistler correlation [559], which is an empirical dynamic contact angle correlation based on the capillary number Ca and utilizing the Hoffman function. The Kistler contact angle is defined as:

$$\theta_k = f_{Hoff}\left(Ca + f_{Hoff}^{-1}\left(\theta_s\right)\right) \tag{2809}$$

where f_{Hoff} is the Hoffman function:

$$f_{Hoff}(x) = \cos^{-1}\left(1 - 2\tanh\left(5.16\left(\frac{x}{1 + 1.31x^{0.99}}\right)^{0.706}\right)\right) \tag{2810}$$

with f_{Hoff}^{-1} being its inverse. STAR-CCM+ uses a slightly modified definition of the Hoffman function to achieve a closed form for its inverse function. In the modified Hoffman function, the $x^{0.99}$ term in the denominator is changed to a simple x. The closed form of the inverse function is valid for contact angle values in the range 0–176 degrees. Advancing or receding contact angles that are larger than the upper bound are limited to 176 degrees.

The static contact angle θ_s is inserted into Eqn. (2809) as either the static advancing or receding contact angle, depending on the sign of the capillary number. θ_s is specified as the *Advancing Contact Angle* or *Receding Contact Angle* in the Contact Angle Properties.

To enhance the stability of the method that is implemented in STAR-CCM+, a range for the equilibrium capillary number (Ca_{eq}) can be defined. Within the specified range $-Ca_{eq} < Ca < Ca_{eq}$, the resulting dynamic contact angle is blended with the equilibrium contact angle (θ_e) as a weighted average:

$$\theta_d = f\theta_e + (1 - f)\theta_k \tag{2811}$$

While θ_e is a user-specified value, the factor f is determined within the Ca_{eq} range as:

$$f = 0.5 + 0.5\cos\left(\frac{Ca}{Ca_{eq}}\pi\right) \tag{2812}$$

Free Surface Waves

STAR-CCM+ provides the VOF Waves model to simulate surface gravity waves on the interface between a liquid and a gas, typically the atmosphere. Application examples for VOF waves are the investigation of flow around ships or wave impact on offshore structures.

The concept of a steadily-progressing periodic wave train is a convenient model that is used in coastal and ocean engineering applications to give fluid velocities, pressures, and surface elevations caused by waves. Assuming that the waves are propagating steadily without change (the steady wave problem), the wave train can be uniquely specified and solved in terms of three physical length scales: water depth, wave length, and wave height.

Flat Wave

A flat wave represents a calm plane of water.

First Order Wave

A first order wave is modeled with a first order approximation to the Stokes theory of waves. This approximation generates waves that have a regular periodic sinusoidal profile.

The equation for horizontal velocity is:

$$v_h = a\omega\cos\left(\mathbf{K}\cdot\mathbf{x} - \omega t\right)e^{Kz} \tag{2813}$$

The equation for vertical velocity is:

$$v_v = a\omega\sin\left(\mathbf{K}\cdot\mathbf{x} - \omega t\right)e^{Kz} \tag{2814}$$

The equation for surface elevation is:

$$\eta = a\cos\left(\mathbf{K} \cdot \mathbf{x} - \omega t\right) \tag{2815}$$

where:

- a is the wave amplitude
- ω is the wave frequency
- $\mathbf{K}$ is the wave vector
- K is the magnitude of the wave vector
- z is the vertical distance from the mean water level.

The wave period T is defined as:

$$T = \frac{2\pi}{\omega} \tag{2816}$$

The wavelength λ is defined as:

$$\lambda = \frac{2\pi}{K} \tag{2817}$$

The dispersion relation (between wave period T and wave length λ) for first order waves in finite water depth d is:

$$T = \left[\frac{g}{2\pi\lambda}\tanh\left(\frac{2\pi d}{\lambda}\right)\right]^{-1/2} \tag{2818}$$

Whereas for infinite water depth, the dispersion relation is:

$$\lambda = \frac{gT^2}{2\pi} \tag{2819}$$

The wave shape is independent of depth.

Fifth Order Wave

A fifth order wave is modeled with a fifth order approximation to the Stokes theory of waves. This wave more closely resembles a real wave than a wave that is generated by the first order method. The wave profile and the wave phase velocity depend on the water depth, wave height, and current.

The fifth order VOF waves are based on work by Fenton [138].

The Ursell number U_R is defined as [135]:

$$U_R = \frac{H\lambda^2}{d^3} \tag{2820}$$

where H, is the wave height λ is the wavelength and d is the depth of the water.

This wave theory is valid only for Ursell numbers less than 30.

Superposition Wave

A superposition wave is a linear superposition of different partial first order waves. It can be used to simulate more complex wave phenomena, such as a cross sea or spectral waves. A cross sea is a sea state with two wave systems traveling at oblique angles. This state can occur when water waves from one weather system continue despite a shift in wind.

Cnoidal Waves

A cnoidal wave is a nonlinear and exact periodic wave solution of the Korteweg–de Vries equation, which describes the propagation of waves over a flat bed. Korteweg & de Vries [142] obtained periodic solutions which they termed ”cnoidal” because the surface elevation is proportional to the square of the Jacobian elliptic function $\mathrm{cn}()$. Cnoidal waves are used to describe surface gravity waves that have a long wavelength when compared to the fluid depth. The cnoidal solution displays the long flat troughs and narrow crests that are observed in natural shallow water waves. As the wavelength tends to infinity, the cnoidal solution describes a solitary wave.

The cnoidal solution presents results in terms of expansions in wave height. The parameter that is used is the wave height relative to the trough depth of fluid, H/h, which is denoted by ϵ.

The series is expressed as a power series in ϵ/m rather than ϵ, [137]. As m can be less than 1, it is better to monitor the magnitude of ϵ/m than to have a power series in ϵ with coefficients which are polynomials in $1/m$.

The full solution to third order is presented. This solution is more applicable to shorter and reduced-amplitude waves, where the parameter m is less than 0.96.

The symbol cn is used to denote $\mathrm{cn}(\alpha X/h|m) = \mathrm{cn}(\alpha(x-ct)/h|m)$.

Surface elevation calculation

$$\begin{aligned}\frac{\eta}{h} &= 1 + \left(\frac{\epsilon}{m}\right) m\,\mathrm{cn}^2 + \left(\frac{\epsilon}{m}\right)^2\left(-\frac{3}{4}m^2\mathrm{cn}^2 + \frac{3}{4}m^2\mathrm{cn}^4\right) \\ &+ \left(\frac{\epsilon}{m}\right)^3\left(\left(-\frac{61}{80}m^2 + \frac{111}{80}m^3\right)\mathrm{cn}^2 + \left(\frac{61}{80}m^2 - \frac{53}{20}m^3\right)\mathrm{cn}^4 + \frac{101}{80}m^3\mathrm{cn}^6\right)\end{aligned} \tag{2821}$$

Horizontal fluid velocity in the frame of the wave calculation

Variation in the x-direction is assumed to be relatively slow and can be expressed in terms of a scaled dimensionless variable $\alpha x/h$. α is a small quantity which expresses the relative slowness of variation in the x-direction, and h is the trough depth.

The series are expressed in terms of α^2 (specifically $\delta = 4\alpha^2/3$) [139]. The results are accurate even for high waves.

$$\begin{aligned}\frac{U}{\sqrt{gh}} &= -1 + \delta\left(\frac{1}{2} - m + m\mathrm{cn}^2\right) \\ &+ \delta^2\left(-\frac{19}{40} + \frac{79}{40}m - \frac{79}{40}m^2 + \mathrm{cn}^2\left(-\frac{3}{2}m + 3m^2\right) - m^2\mathrm{cn}^4\right. \\ &\left.+ \left(\frac{Y}{h}\right)^2\left(-\frac{3}{4}m + \frac{3}{4}m^2 + \mathrm{cn}^2\left(\frac{3}{2}m - 3m^2\right) + \frac{9}{4}m^2\mathrm{cn}^4\right)\right) \\ &+ \delta^3\left(\frac{55}{112} - \frac{3471}{1120}m + \frac{7113}{1120}m^2 - \frac{2371}{560}m^3 + \mathrm{cn}^2\left(\frac{71}{40}m - \frac{339}{40}m^2 + \frac{339}{40}m^3\right)\right. \\ &+ \mathrm{cn}^4\left(\frac{27}{10}m^2 - \frac{27}{5}m^3\right) + \frac{6}{5}m^4\mathrm{cn}^6 + \left(\frac{Y}{h}\right)^2\left(\frac{9}{8}m - \frac{27}{8}m^2 + \frac{9}{4}m^3\right. \\ &+ \mathrm{cn}^2\left(-\frac{9}{4}m + \frac{27}{2}m^2 - \frac{27}{2}m^3\right) + \mathrm{cn}^4\left(-\frac{75}{8}m^2 + \frac{75}{4}m^3\right) - \frac{15}{2}m^3\mathrm{cn}^6\Big) \\ &+ \left(\frac{Y}{h}\right)^4\left(-\frac{3}{16}m + \frac{9}{16}m^2 - \frac{3}{8}m^3 + \mathrm{cn}^2\left(\frac{3}{8}m - \frac{51}{16}m^2 + \frac{51}{16}m^3\right)\right. \\ &\left.\left.+ \mathrm{cn}^4\left(\frac{45}{16}m^2 - \frac{45}{8}m^3\right) + \frac{45}{16}m^3\mathrm{cn}^6\right)\right)\end{aligned} \tag{2822}$$

The leading term -1 arises because the wave is considered to be traveling in the positive x-direction. The fluid is flowing under the wave in the negative x-direction, relative to the wave, with velocities of the order of the wave speed.

Vertical fluid velocity calculation

The vertical fluid velocity can be obtained from Eqn. (920) by using the mass conservation equation $\partial U/\partial X + \partial V/\partial Y = 0$, and $d(\mathrm{cn}(\theta|m))/d\theta = -\mathrm{sn}(\theta|m)\mathrm{dn}(\theta|m)$. Each term that contains $(Y/h)^i \mathrm{cn}^j(\alpha X/h|m)$, for $j > 0$, is replaced by $\alpha \mathrm{sn}(\,)\mathrm{dn}(\,)\left(\frac{j}{i+1}\right) \times (Y/h)^{i+1}\mathrm{cn}^{j-1}(\,)$. Hence, if Eqn. (920) is written as:

$$\frac{U}{\sqrt{gh}} = -1 + \sum_{i=1}^{5} \delta^i \sum_{j=0}^{i-1} \left(\frac{Y}{h}\right)^{2j} \sum_{k=0}^{i} \mathrm{cn}^{2k}(\,)\Phi_{ijk} \tag{2823}$$

where each coefficient Φ_{ijk} is a polynomial of degree i in the parameter m, then the vertical velocity component is:

$$\frac{V}{\sqrt{gh}} = 2\alpha \mathrm{cn}(\,)\mathrm{sn}(\,)\mathrm{dn}(\,) \sum_{i=1}^{5} \delta^i \sum_{j=0}^{i-1} \left(\frac{Y}{h}\right)^{2j+1} \sum_{k=1}^{i} \mathrm{cn}^{2(k-1)}(\,)\frac{k}{2j+1}\Phi_{ijk} \tag{2824}$$

Irregular Waves

An irregular wave models wind seas (a short-term sea state) with a wave spectrum, that is, the power spectral density function of the vertical sea surface displacement. An irregular wave can use either the Pierson-Moskowitz spectrum or the JONSWAP spectrum [136]. Both spectra describe wind sea conditions that often occur for the most severe sea states.

The Pierson-Moskowitz spectrum $S_{PM}(\omega)$ is defined as:

$$S_{PM}(\omega) = \frac{5}{16}\left(H_S^2 \omega_p^4\right)\omega^{-5}\exp\left(-\frac{5}{4}\left(\frac{\omega}{\omega_p}\right)^{-4}\right) \tag{2825}$$

where $\omega_p = (2\pi)/\left(T_p\right)$ represents the angular spectral peak frequency.

The JONSWAP spectrum $S_J(\omega)$ is expressed as:

$$S_J(\omega) = A_\gamma S_{PM}(\omega)\gamma^{\exp\left(-0.5\left(\frac{\omega-\omega_p}{\sigma\omega_p}\right)^2\right)} \tag{2826}$$

where:

- $S_{PM}(\omega)$ is the Pierson-Moskowitz spectrum
- γ is the non-dimensional peak shape parameter
- $A_\gamma = 1 - 0.287\ln(\gamma)$ is a normalizing factor
- σ is the spectral width parameter:

  σ_a for $\omega \le \omega_p$

  σ_b for $\omega > \omega_p$.

Contents:

Suppression of Wave Reflections at Boundaries

Suppression of Wave Reflections at Boundaries

Wave Damping

Damping of waves is possible by introducing resistance to vertical motion. STAR-CCM+ implements the method by Choi and Yoon [134], adding a resistance term to the equation for w-velocity:

$$S_z^{\mathrm{d}} = \rho\left(f_1 + f_2|w|\right)\frac{e^{\kappa}-1}{e^{1}-1}w \tag{2827}$$

with:

$$\kappa = \left(\frac{x-x_{\mathrm{sd}}}{x_{\mathrm{ed}}-x_{\mathrm{sd}}}\right)^{n_d} \tag{2828}$$

where:

- x_{sd} is the starting point for wave damping (propagation in the x-direction)
- x_{ed} is the end point for wave damping (boundary)
- f_1, f_2 and n_d are parameters of the damping model
- w is the vertical velocity component.

Wave Forcing

Following the approach of Kim and others ([141]), STAR-CCM+ allows coupling of the 3D flow simulation with a theoretical solution that is specified by VOF waves.

Forcing the solution of the discretized Navier-Stokes equations towards another solution (such as a theoretical solution or simplified numerical solution) over a specified distance reduces the computing effort by using a reduced-size solution domain. This forcing also eliminates problems that are associated with reflections of surface waves at boundaries, owing to the damping feature of the gradual forcing.

Wave forcing applies only to Momentum. No Phase sources or Turbulence sources are added. Wave forcing is achieved by adding a source term to the transport (momentum) equations of the form:

$$q_\phi = -\gamma\rho(\phi - \phi^*) \tag{2829}$$

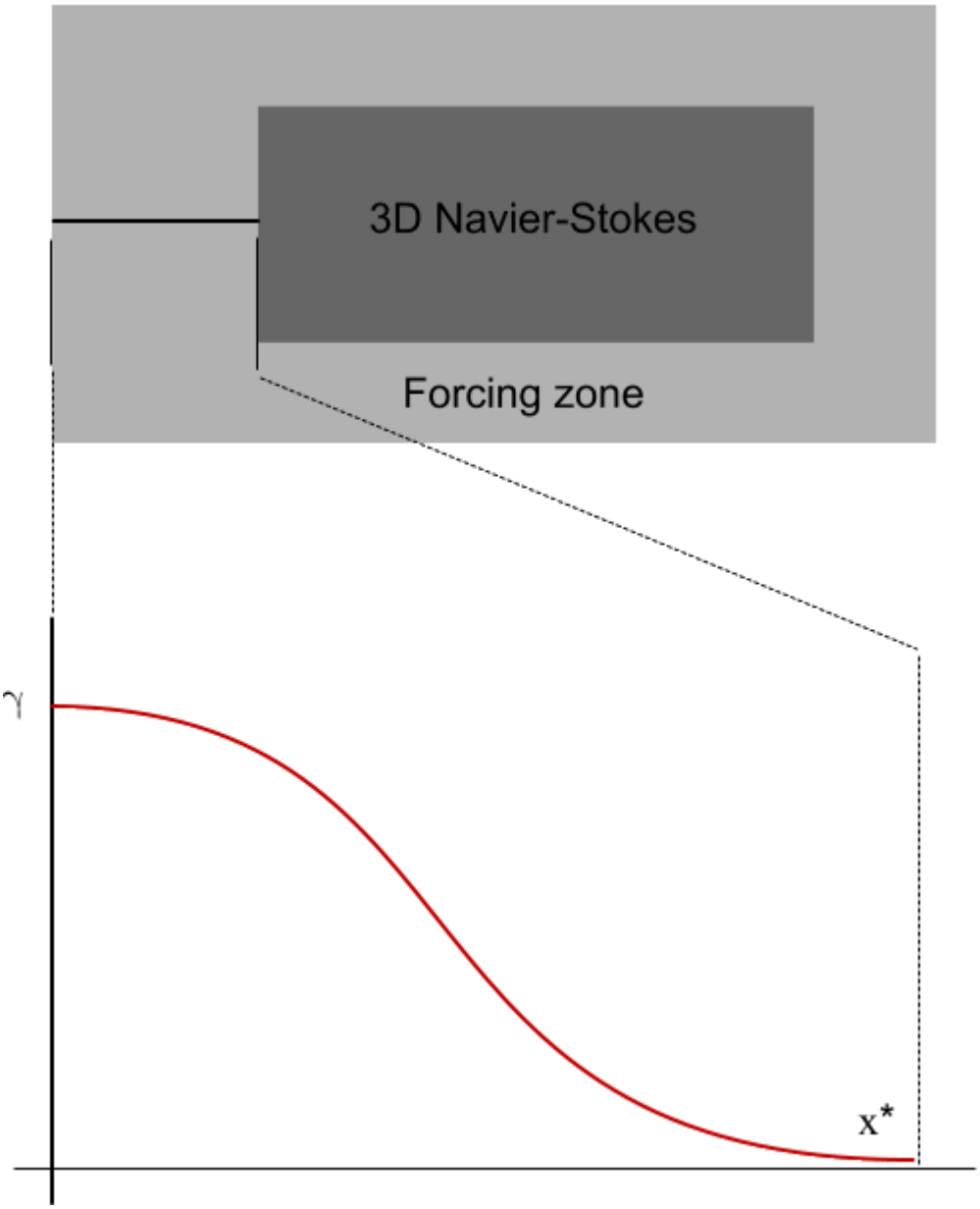
where:

- γ is the forcing coefficient
- ρ is the fluid density
- ϕ is the current solution of the transport equation
- ϕ^* is the value towards which the solution is forced.

When the solution needs to be fixed to a particular value of ϕ^*, a large number is used for γ. The remaining parts of the discretized equation then become negligible.

The source term from Eqn. (927) is applied with a variable forcing coefficient over a specified forcing zone. For example, the figure below is a schematic view of two overlapping solution domains. The 3D Navier-Stokes equations are solved in one domain, within the shaded zones. No forcing is applied within the inner zone (3D Navier-Stokes), but within the outer zone (Forcing zone) the forcing source term is activated along the solution

domain boundaries. The width of the forcing zone can be different for different boundaries. The optimal width of the forcing zone depends on the problem that you are modeling.



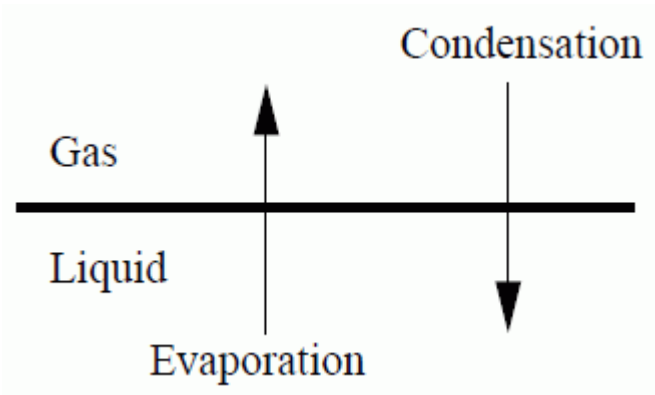
The forcing coefficient varies smoothly from zero at the inner edge of the forcing zone to the maximum value γ at the boundary (the outer edge of the forcing zone).

$$\gamma = -\gamma_0 \cos^2(\pi x^*/2) \tag{2830}$$

The volume coupling can also be either one-way or two-way, as the forcing zones usually do not coincide.

Evaporation and Condensation

Evaporation is a form of vaporization, the phase change from liquid to gas, which occurs when liquid molecules or atoms with high enough kinetic energy escape through the interface between the two phases. Condensation is the reverse process of vaporization and is the phase change from gas to liquid.



In STAR-CCM+, evaporation and condensation deal with evaporation (without boiling) and condensation taking place at the free surface interface between a liquid and a gas phase. Both phases are modeled as multi-component mixtures that can include non-interacting (inert) species [547], [546]. Evaporation and condensation are treated as hydrodynamically limited, that is, the phases are considered to be in equilibrium at the interface and the driving force is the diffusion of species. Raoult's law describes the phase equilibrium.

The evaporation rate (kg/s m^2) for a particular component i is given by:

$$\dot{m}_i = -\frac{\rho_g D_{g,i} \left.\frac{\partial Y_{g,i}}{\partial n}\right|_s}{1 - \sum_{j=1}^{N_v} Y_{g,j}^s} \tag{2831}$$

where:

- ρ_g is the density of the gas phase
- Y^s is the component mass fraction at the surface
- $D_{g,i}$ is the diffusion coefficient
- N_v is the total number of components undergoing evaporation or condensation
- Subscript g denotes a gas phase variable

The vapor pressure p_i of a component is:

$$p_i = a_i p_i^* \tag{2832}$$

where p_i^* is the vapor pressure of the pure component i and a_i is the activity that depends on the liquid mixture. This expression can be written as:

$$p_i = \gamma_i X_{L,i} p_i^* \tag{2833}$$

where γ_i is the activity coefficient that accounts for deviations from ideal behavior of a mixture.

Raoult's law states that $\gamma_i \approx 1$, an approximation that is used here. Given the vapor pressures of the components, their molar fractions at the interface can be derived from:

$$X^s_{g,i} = \frac{p_i}{p} \tag{2834}$$

where p is the pressure. The conversion to interface vapor mass fraction is done through:

$$Y^s_{g,i} = \frac{X^s_i W_i}{\sum_{j=1}^{N_v} X^s_j W_j + \sum_{j=1}^{N_{g,p}} X^s_j W_j} \tag{2835}$$

where $N_{g,p}$ is the number of inert components in the gas phase. While the molar fraction of the evaporation components is known, it is unknown for the inert species (which do not undergo evaporation or condensation). A good approximation is the introduction of a background molar weight calculated a short distance away from the interface:

$$W_{bg} = \frac{\sum_{j=1}^{N_{g,p}} X_j W_j}{\sum_{j=1}^{N_{g,p}} X_j} \tag{2836}$$

The interfacial background molar fraction is:

$$X^s_{bg} = 1 - \sum_{j=1}^{N_v} X^s_j \tag{2837}$$

Therefore the interfacial vapor mass fraction can be approximated as:

$$Y^s_{g,i} \approx \frac{X^s_i W_i}{\sum_{j=1}^{N_v} X^s_j W_j + X^s_{bg} W_{bg}} \tag{2838}$$

The numerical solution of Eqn. (2831) begins from the definition of the following scalar function $Y_{g,m}$, for every interacting component:

$$Y_{g,m} = \alpha_l Y^s_{g,i} + \alpha_g Y_{g,i} \tag{2839}$$

where α_l is the liquid volume fraction and α_g is the gas volume fraction. The approximated evaporation rate $\dot{M}'_{i,c}$ for one cell can be obtained as:

$$\dot{M}'_{i,c} \approx -\frac{\rho_g D_{g,i} \nabla Y_{g,m} \nabla \alpha_l V_c}{1 - \sum_{j=1}^{N_v} Y^s_{g,j}} \tag{2840}$$

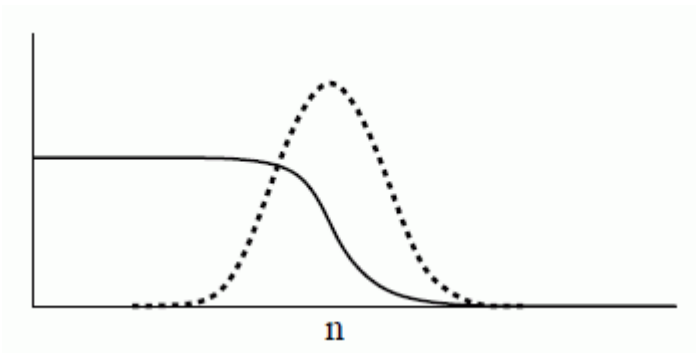
using the fact that the integral of $\nabla \alpha_l$, the gradient of liquid fraction, over a certain volume expresses the area of the interface it contains [522], [533].

Comparing Eqn. (2831) and Eqn. (2840), it is seen that $\nabla Y_{g,m}$ expresses the effect of $\nabla Y_{g,i}$. A lowest order analysis shows that the relative error of the evaporation rate, in quasi-steady state, behaves as:

$$\varepsilon_{rel} \approx \frac{1}{H|\nabla\alpha|} \tag{2841}$$

where H expresses the thickness of the boundary layer above the interface.

As such, the accuracy depends on how well the location of the interface is determined in comparison with the boundary layer. When the interface is not smeared, $|\nabla\alpha|^{-1}$, the reciprocal of the magnitude of the gradient of volume fraction, is a measure of the mesh size. This condition highlights the need for a sufficiently fine grid, to model the transport of species away from and towards the interface accurately. Similar considerations are valid for gradients of the temperature and pressure near the interface.



The sources for species, phases, and continuity are, based on Eqn. (2840), distributed over those cells that make up the interface, that is with non-vanishing gradient of the volume fraction, as shown in the above graph depicting the typical behavior of α_l (solid line) and $-\left|\nabla\alpha_l\right|$ (dashed line) at a free surface interface, along the surface normal coordinate n.

Boiling

Boiling is a rapid vaporization of a liquid. It typically takes place when a liquid is heated to the boiling point (saturation temperature of the liquid). Its saturation vapor pressure then becomes equal to or larger than the pressure of the surrounding liquid.

Wall Boiling

When a liquid is in contact with a wall that is maintained at temperatures T_{wall} above the saturation temperature of the liquid, boiling eventually occurs at that liquid-solid interface. In this case, boiling occurs in three characteristic stages:

- Nucleate boiling involves a creation and growth of vapor bubbles on a heated surface, which rise from discrete points on a surface. The temperature of the surface is only slightly above the saturation temperature of the liquid. In general, the number of nucleation sites increases with an increasing surface temperature. An increased surface roughness can create more nucleation sites, while an exceptionally smooth surface can result in superheating.
- Film boiling occurs when the critical heat flux is exceeded and a continuous vapor film covers the heated surface. The vapor layer has a lower thermal conductivity so the vapor layer typically “insulates” the surface.

- Transition boiling occurs at surface temperatures between the maximum attainable in nucleate and the minimum attainable in film boiling. It is an intermediate, unstable form of boiling with elements of both types.

There are two distinct choices for modeling boiling in STAR-CCM+, namely the Rohsenow and the Transition Boiling models. The former uses the Rohsenow nucleate boiling model applicable for boiling at relatively low solid temperatures. A film boiling model is also integrated into the Rohsenow implementation in STAR-CCM+ to facilitate its use with boiling at high solid temperatures. The Transition Boiling model has expressions for nucleate and transition boiling.

Rohsenow Boiling Model

The empirical correlation that Rohsenow [601] presented is used to calculate the surface heat flux due to boiling:

$$q_{bw} = \mu_l h_{lat} \sqrt{\frac{g(\rho_l - \rho_v)}{\sigma}} \left(\frac{C_{p_l}(T_w - T_{sat})}{C_{qw} h_{lat} Pr_l^{n_p}} \right)^{3.03} \tag{2842}$$

In this equation, μ_l, h_{lat}, C_{p_l}, ρ_l, and Pr_l are the dynamic viscosity, latent heat, specific heat, density, and Prandtl number of the liquid phase, n_p is the Prandtl number exponent (1.73 by default), g is gravity, and ρ_v is the vapor density, σ is the surface tension coefficient at the liquid-vapor interface, T_w is the wall temperature, T_{sat} is the saturation temperature, and C_{qw} is an empirical coefficient varying with the liquid-surface combination.

The vapor mass generation rate $\dot{m}_{ew}$ over the area that nucleation sites cover is:

$$\dot{m}_{ew} = \frac{C_{ew} q_{bw}}{h_{lat}} \tag{2843}$$

where C_{ew} is a model constant stating how much of the boiling heat flux is used for creation of vapor bubbles.

If the Rohsenow correlation is applied outside its range of applicability (for example, to the film boiling regime), unrealistically high heat fluxes could result. Fluid temperatures can become higher than the near wall temperature. This behavior stems from the fact that the Rohsenow correlation does not depend on the fluid temperature; heat enters the domain irrespective of the fluid temperature.

To prevent this condition, the heat flux that the correlation calculates is multiplied by:

$$max\left[0, min\left(\left(\frac{T_w - T}{Tw - T_{sat}}\right), 1\right)\right] \tag{2844}$$

where T is the fluid temperature near the heated wall. Thus, if $T < T_{sat}$, the Rohsenow correlation is used directly, and if $T > T_w$ the boiling heat flux is zero. For fluid temperatures which are between the wall and the saturation temperature, only a fraction of the heat flux predicted by the correlation is used.

Film Boiling Model

At a sufficiently high wall temperature, a layer of vapor is created above the heated wall. As a result the liquid is no longer in contact with the heated wall. The evaporation occurs at the vapor liquid interface instead of at the heated wall. The vapor film that is created between the liquid and the heated wall acts as

an insulator, which slows down the heat transfer process considerably. This process is known as film boiling.

The film boiling model assumes that the film thickness is resolved in one or more cells. That is, the vapor occupies one or more cells close to the heated wall. In this case, the expressions for the wall heat flux are the same as in the case of single-phase flows. If it is not possible to have a mesh which resolves the film thickness, you specify the value of the vapor volume fraction $\alpha_{filmBoiling}$, which indicates when the film boiling starts. For values of the vapor volume fraction smaller than $\alpha_{filmBoiling}$ the nucleate boiling is assumed, and for larger values it is assumed that the film boiling takes place. If the film is resolved and the cell next to the heated wall is filled with vapor, the results do not depend on values of $\alpha_{filmBoiling}$ from 0 through 1.

The film-boiling vapor volume fraction $\alpha_{filmBoiling}$ is the value of volume fraction of the vapor phase which indicates full transition to film boiling.

If the VOF boiling model is active, you can specify $\alpha_{filmBoiling}$ for each wall boundary. A value of $\alpha_{filmBoiling}$ larger than 1 implies that there is no film boiling along these wall boundaries.

For values of volume fraction less than $\alpha_{filmBoiling}$, a transition function is:

$$f\left(\alpha_{vapor}\right) = 1 - \min\left(1, \alpha_{vapor}/\alpha_{filmBoiling}\right)^8 \tag{2845}$$

This function controls the fraction of boiling at the wall that is considered to be nucleate boiling. Here α_{vapor} is the vapor volume fraction at the wall.

Transition Boiling Model

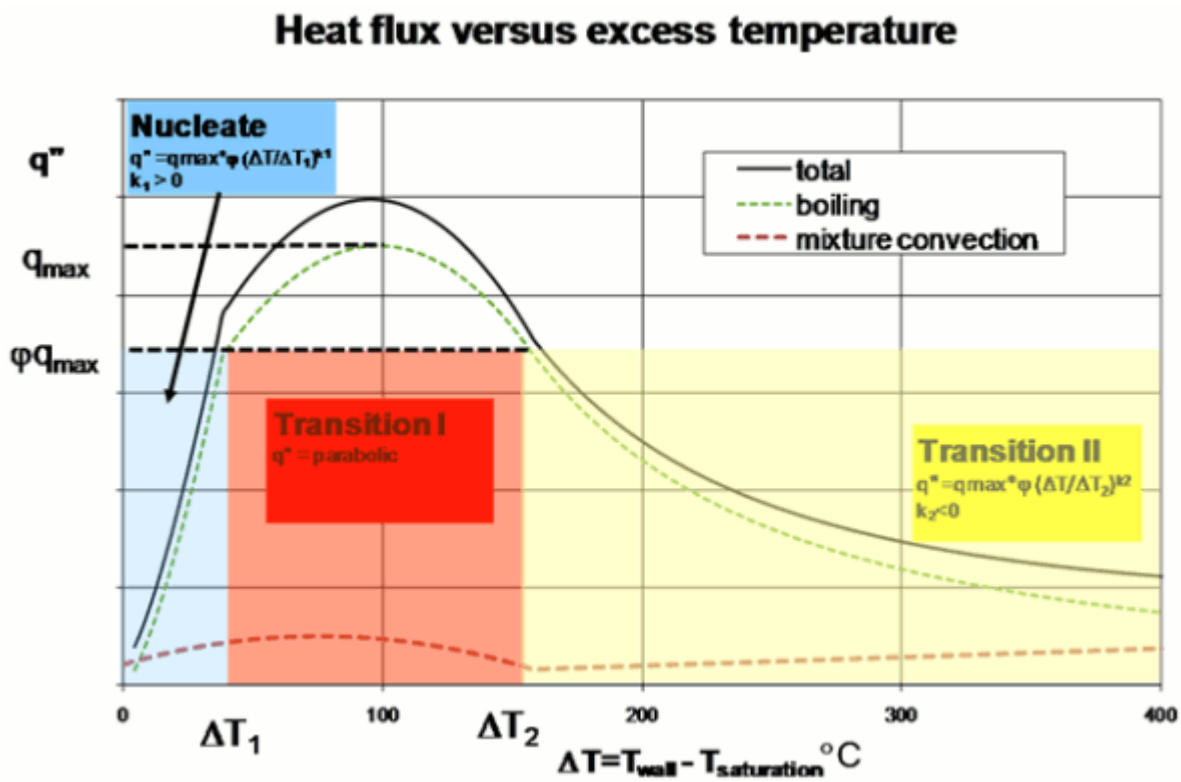
You can specify the constants of this boiling model. The equations are as follows:

$$q_{boiling}\left(\Delta T\right) = q_{\max} S \phi \left(\frac{\Delta T}{\Delta T_1}\right)^{K_1} \qquad 0 \le \Delta T \le \Delta T_1 \tag{2846}$$

$$q_{boiling}\left(\Delta T\right) = q_{\max} S \left[1 - 4\left(1-\phi\right)\left(\frac{\Delta T - \Delta T_{\max}}{\Delta T_2 - \Delta T_1}\right)^2\right] \qquad \Delta T_1 \le \Delta T \le \Delta T_2 \tag{2847}$$

$$q_{boiling}\left(\Delta T\right) = q_{\max} S \phi \left(\frac{\Delta T}{\Delta T_2}\right)^{-K_2} \qquad \Delta T_2 \le \Delta T \tag{2848}$$

The first of the above equations simulates nucleate boiling, while the remaining two simulate transition I and transition II boiling. The three regimes are shown graphically in the following figure.



There are five empirical constants in the above model, namely, K_1, K_2, ΔT_1, ΔT_2, and $q_{\max}$, all of which are positive. ΔT_{max} is $\frac{1}{2}(\Delta T_1 + \Delta T_2)$.The constant ϕ is set equal to 0.75. Further, ΔT_2 must be greater than ΔT_1. (In the above expressions K_2 is positive even though it is a negative exponent because the sign is accounted for in the implementation.) The scale factor S in the above expressions is not treated as a sixth constant. It allows you to scale $q_{\max}$ up or down for a given boundary. All the constants are likely to be application-specific. Hence, it is recommended that default values for the constants be tuned with credible measurements. The default values are from fits of boiling experiments of Ellion [529] using an apparatus with a particular geometry.

As mentioned in the previous section, $\alpha_{filmBoiling}$ is the metric used to toggle between the nucleate and film boiling regimes in the Rohsenow Boiling Model. However, in the Transition Boiling Model, $\alpha_{filmBoiling}$ is not used; instead ΔT_1 and ΔT_2 are the metrics that switch between the three expressions (or regimes).

The total heat flux at the wall is the sum of the heat fluxes due to boiling, convection of the liquid-vapor mixture and thermal radiation. In the above equations, $q_{\max}$ is the maximum of the boiling heat flux component; it is not the critical heat flux which is a maximum of the total heat flux. The point on the curve corresponding to the minimum of the total heat flux is the Leidenfrost point and it occurs in the Transition II boiling regime. At the Leidenfrost point, vapor completely covers the surface and heat transfer is primarily driven by mixture convection and radiation with a negligible boiling contribution.

In the CFD implementation of the above boiling model, the boiling heat flux contribution to the total heat flux is considered to be zero when there is no liquid to boil in any cell that is next to the wall. Further, if the fluid temperature T is above the saturation temperature T_{sat} the boiling heat flux is calculated according to Eqn. (2844).

Bulk Boiling

The heat transfer to the interface between the phases is used to compute the rate of evaporation or condensation $\dot{m}_{ec}$. The following are assumed:

- The vapor bubbles are on the saturation temperature T_{sat}
- The temperature of the liquid T_l is approximated by the temperature of the liquid-vapor mixture T
- All the heat flux from the liquid to the interface is used in mass transfer, that is, evaporation or condensation:

$$\dot{m}_{ec} = \frac{\mathrm{HTCxArea}\left(T - T_{sat}\right)}{h_{lat}} \tag{2849}$$

where h_{lat} is the latent heat of vaporization. $\mathrm{HTCxArea}$ is the heat transfer coefficient between the vapor bubbles and the surrounding liquid, which is multiplied by the specific contact area (contact area per unit volume) between the two.

Cavitation and Gas Dissolution

When the static pressure at a particular location within a liquid falls below the saturation vapor pressure of the liquid, the liquid undergoes a phase change called cavitation. This phase change creates cavitation bubbles that are filled with the vapor of the liquid. Liquid can dissolve a gas up to a certain maximal concentration (saturation). When there are changes in the pressure or temperature (or both), the liquid can become over-saturated or under-saturated. In the over-saturated case, gas bubbles will form and grow in the liquid; in the under-saturated case the gas bubbles will shrink and disappear. For a multi-component gas phase, the bubbles of free gas are a mixture of those components that come out of the solution. The presence of dissolved and free gas in a liquid affects other properties, such as the rates of cavitation.

STAR-CCM+ uses the same homogeneous seed-based approach for both cavitation and gas dissolution models. Although these zero-dimensional approaches do not capture all of the physical phenomena present, they are proven methods for use in industrial design studies.

Seed-based mass transfer models are used for gas dissolution and cavitation. The seed-based models are based on [545]. The two interacting phases are denoted as l and v, with mass transfer positive from l to v.

In order to be able to account for the effects of liquid purity (that is, the content of solid particles and dissolved gas), it is assumed that nucleation sites (seeds) are present in liquid. The following modeling assumptions are made:

- Seeds are spherical and uniformly distributed in liquid, as characterized by the number n_0 (the number of seeds per unit volume of liquid).
- All seeds initially have the same radius.

It is assumed that the number of seeds in a control volume is proportional to the amount of liquid, so there is no need to follow each seed and track its motion in space.

Under the above assumptions, the number of seeds within a control volume at any time is:

$$N = n_0 \alpha_l V \tag{2850}$$

where α_l is the volume fraction of liquid within a control volume of volume V.

The total vapor volume inside the control volume equals:

$$V_v = NV_b \tag{2851}$$

where V_b is the volume of one bubble, and:

$$V_b = \frac{4}{3}\pi R^3 \tag{2852}$$

with R being the local bubble radius.

The vapor volume fraction α_v can now be expressed in terms of n_0, R, and volume fraction of liquid, α_l, following the definition:

$$\alpha_v = \frac{V_v}{V} = \frac{NV_b}{V} = \frac{4}{3}\pi R^3 n_0 \alpha_l \tag{2853}$$

When the volume fraction of vapor is known, the local radius of bubbles can be computed from this expression:

$$R = \left(\frac{3\alpha_v}{4\pi n_0 \alpha_l}\right)^{1/3} \tag{2854}$$

The modeling framework that is outlined above is used to define the rate of vapor production or consumption in the mass conservation equation. The rate of bubble radius change is the key parameter, as vapor production results in bubble growth and vapor consumption results in bubble collapse. Another parameter is the amount of liquid in the control volume. Obviously, any increase of the vapor mass is accompanied by a corresponding decrease in the liquid mass.

The vapor bubbles are moving with the flow. Therefore, the rate at which vapor is created at any particular time is approximately the rate at which the volume of bubbles present in the control volume at that time is changing. The change of volume of an individual bubble is:

$$\frac{dV_b}{dt} = 4\pi R^2 \frac{dR}{dt} = 4\pi R^2 v_r \tag{2855}$$

where $v_r = \mathrm{d}R/\mathrm{d}t$ denotes the bubble growth velocity.

The rate of growth (or collapse) of all bubbles present in a control volume represents the source term in the equation for vapor volume fraction, that is:

$$Q_\mathrm{v} = N\frac{dV_b}{dt} = 4\pi n_0 \left(1 - \alpha_\mathrm{v}\right) V R^r v_r \tag{2856}$$

The above source term represents the contribution from cavitation only; vapor can also be produced by boiling. For example, the same flow can have vapor generation at a hot wall due to boiling and vapor generation at a sharp corner due to locally low pressure and cavitation. In such cases, the source terms that result from different models must be added up.

The volume fraction of the vapour can be expressed as:

$$\alpha_v = \frac{V_v}{V} = \frac{V_v}{V_v + V_l} \tag{2857}$$

where V is the control volume, and V_v and V_l denote the respective parts of the volume that the vapour and the liquid phases occupy.

Within the control volume, the vapour phase is assumed to be present in the form of bubbles. Each bubble has the same radius R. The number density of seeds is defined as n_0, which corresponds to the number of bubbles per unit volume.

Therefore the following relationship holds between the vapour and liquid phases:

$$V_v = n_0 V_l \frac{4}{3}\pi R^3 \tag{2858}$$

The volume fraction of gas, α_v, becomes:

$$\alpha_v = \frac{n_0 \frac{4}{3}\pi R^3}{1 + n_0 \frac{4}{3}\pi R^3} \tag{2859}$$

The vapour bubble radius can be expressed as:

$$R^3 = \frac{\alpha_v}{n_0 \frac{4}{3}\pi\left(1-\alpha_v\right)} \tag{2860}$$

The seed diameter $D_0 = 2R_0$ (which is user-specified) provides a lower limit R_{min} for the bubble size.

The mass transfer rate per unit volume can be written as:

$$\dot{m} = n_0 \alpha_l 4\pi \rho_v R^2 v_r \tag{2861}$$

For multi-component cavitation, the overall cavitation rate is broken down into rates for the individual components. Unresolved interfacial properties also affect this overall rate. A reasonable assumption is to use the following:

$$\dot{m}_{cav,i} = \dot{m}_{cav}\left(\left(X_{i,l,\infty} p^*_{sat,i}\right) / \left(\sum_i^{N_c} X_{i,l,\infty} p^*_{sat,i}\right)\right) \tag{2862}$$

This expression represents a weighting that is based on the individual fugacity of each component, in the context of ideal gases and ideal liquids.

The liquid volume disappears with a volume rate:

$$\dot{V}_l = -\frac{\dot{m}}{\rho_l} \tag{2863}$$

Vapour volume is generated with a volume rate:

$$\dot{V}_g = \frac{\dot{m}}{\rho_v} \tag{2864}$$

The remaining unknown factor is v_r giving the rate at which the bubble radius changes.

The following models for the bubble radius change rate are provided:

- The Rayleigh–Plesset cavitation model

- The Schnerr-Sauer cavitation model
- The Gas Dissolution model

If the phases are multi-component, the rate model breaks down the global mass transfer rate $\dot{m}$ in rates $\dot{m}_i$ for each individual component. These component mass transfer rates appear as source terms in the species transport equations.

Full Rayleigh–Plesset Cavitation Model

The Full Rayleigh–Plesset model includes the influence of bubble growth acceleration, as well as viscous and surface tension effects.

The bubble growth velocity v_r is determined using the Rayleigh-Plesset equation:

$$R\frac{dv_r}{dt} + \frac{3}{2}v_r^2 = \frac{p_{sat} - p}{\rho_l} - \frac{2\sigma}{\rho_l R} - 4\frac{\mu_l}{\rho_l R}v_r \tag{2865}$$

where:

- p_{sat} is the saturation pressure for given temperature
- p is the local pressure in the surrounding liquid
- σ is the surface tension
- ρ_l is the liquid density

The bubble-growth velocity that is needed for the source term of the vapor volume fraction equation can be computed without neglecting any of the terms in the Rayleigh-Plesset equation.

For multi-component cavitation, the term p_{sat} in the formulation is replaced with $p_{sat,m}$.

$$p_{sat,m} = \sum_i^{N_c} X_{i,l,s} p^*_{sat,i} \tag{2866}$$

This expression is Raoult's Law, with $p^*_{sat,i}$ indicating the saturation pressure of the pure component. The interfacial molar fraction of the liquid components is approximately their value in the bulk, so $X_{i,l,s} \approx X_{i,l,\infty}$. This formulation neglects the effect of other gases that are present in the bubbles (for example, as a consequence of gas dissociation).

Schnerr-Sauer Cavitation Model

The Schnerr-Sauer cavitation model is based on a reduced Rayleigh–Plesset (RP) equation and neglects the influence of bubble growth acceleration, viscous effects, and surface tension effects. The Schnerr-Sauer cavitation model lets you scale the bubble growth rate and collapse rate for both single-component materials and multi-component materials.

The cavitation bubble growth rate is estimated using the inertia controlled growth model [545]:

$${v_r}^2 = \frac{2}{3}\left(\frac{p_{sat} - p}{\rho_l}\right) \tag{2867}$$

where p_{sat} is the saturation pressure corresponding to the temperature at the bubble surface, p is the pressure of the surrounding liquid and ρ_l is the liquid density. Eqn. (2867) is a simplification of the more general Rayleigh-Plesset equation which accounts for the inertia, viscous effects, and surface tension effects.

Research results suggest that for most practical applications, it is not necessary to account for the viscous and surface tension effects [523].

For multi-component cavitation, the term p_{sat} in the formulation is replaced with $p_{sat,m}$.

Gas Dissolution Model

For single-component bubbles, the approximated Epstein-Plesset [530], [538], [541] formulation is assumed to hold:

$$v_r = \frac{D_{i,l}\left(c_{i,l,\infty} - c_{i,l,s}\right)}{\rho_g R} \tag{2868}$$

Due to the small fraction of $c_{i,l}$ present, this formulation can be written as:

$$v_r = \frac{D_{i,l}\rho_l\left(Y_{i,l,\infty} - Y_{i,l,s}\right)}{\rho_g R} \tag{2869}$$

To determine the saturation mass fraction, Henry's law is used, in the following form:

$$X_{i,g,s} = \frac{H_i}{p^{abs}} X_{i,l,s} \tag{2870}$$

or

$$X_{i,l,s} = \frac{p^{abs}}{H_i} X_{i,g,s} \tag{2871}$$

where H_i is the Henry's law constant.

This expression is similar to Raoult's law, but is valid in the limit of $X_{i,l,s} \to 0$, whereas Raoult's law holds in the opposite limit.

In single-component gas dissolution, $X_{i,g,s} = 1$.

To convert molar fraction to mass fraction, use the assumption:

$$\sum_j^N X_{j,l} W_j \approx \sum_j^{N_{sv}} X_{j,l} W_j \tag{2872}$$

with:

$$\sum_j^{N_{sv}} X_j \approx 1 \tag{2873}$$

where N_{sv} is the number of solvent components. The dissolved species barely contribute to the molar mass of the liquid mixture, due to their low concentration. Therefore, a good approximation is:

$$Y_{i,l,s} \approx \left(X_{i,l,s} W_i\right) / \left(\sum_j^N X_{j,l,\infty} W_j\right) \tag{2874}$$

Eqn. (2869) determines the bubble radius change rate. In the single-component case, the partial pressure of this component is equal to the total gas pressure. The extension to multi-component is readily achieved by assuming ideal gas behavior. Each component contributes to the pressure according to its molar fraction in the free gas, and the species distribution within the bubble maintains a constant profile. In that case, $X_{i,g,s}$ becomes a transported quantity that is influenced by the gas dissolution rates.

The bubble radius rate change is computed as:

$$\frac{dR}{dt} = \sum_{i}^{N_{sl}} \frac{D_{i,l}\rho_l\left(Y_{i,l,\infty} - Y_{i,l,s}\right)}{\rho_g R} = \sum_{i}^{N_{sl}} \left(\frac{dR}{dt}\right)_i \tag{2875}$$

Flash Boiling

Flash boiling is a phase change in a high temperature liquid that is depressurized below its vapor pressure. This phenomenon occurs in thermal non-equilibrium and therefore requires a finite rate model.

Homogeneous Relaxation Model

The Homogeneous Relaxation Model (HRM) is used for modeling vaporization mechanisms ranging from cavitation to flash boiling. This model is based on a finite rate equation for the rate of change of the vapor mass fraction with an empirical time-scale formulation.

The rate of change of vapor mass fraction is:

$$\frac{dY}{dt} = \frac{\overline{Y} - Y}{\theta} \tag{2876}$$

where Y is the vapour mass fraction, $\overline{Y}$ is the equilibrium mass fraction, and θ is the relaxation time scale.

θ is calculated from an empirical correlation. The most widely used empirical correlations is:

$$\theta = \theta_0 {\alpha_v}^{-0.54} \psi^{-1.76} \tag{2877}$$

where θ_0 is a modelling constant with a value equal to 3.84×10^{-7}, α_v is the vapor volume fraction, and ψ is a non-dimensional pressure that is defined as:

$$\psi = \frac{P_{sat} - P}{P_{crit} - P_{sat}} \tag{2878}$$

Eqn. (2876) is used to calculate the source terms in the vapour and the liquid volume fraction equations.

Melting and Solidification

Melting is the process which changes the state of matter from solid to liquid. It takes place when a solid substance is heated to the melting temperature. The opposite process is called solidification.

For pure substances, for example water, the melting and solidification usually take place at one temperature, that is, the liquidus and the solidus are at the same temperature. The melting-solidification model that is implemented in STAR-CCM+ does not track the liquid-solid interface explicitly. Instead, the model uses an enthalpy formulation to determine the distribution of the solidified portion of the liquid-solid phase.

For melting-solidification, the enthalpy of the liquid-solid phase ${h_{\mathrm{ls}}}^*$ includes the latent heat of fusion h_{fusion}:

$${h_{\mathrm{ls}}}^* = h_{\mathrm{ls}} + \left(1 - {\alpha^*}_s\right) h_{fusion} \tag{2879}$$

where h_{ls} is the sensible enthalpy.

The relative solid volume fraction $\alpha^*{}_s$ is defined as the portion of the volume of the liquid-solid phase that is in the solid state. In the enthalpy model, the relative solid volume fraction $\alpha^*{}_s$ is a function of temperature:

$$\alpha^*{}_s = \begin{cases} 1 & if \quad T^* < 0 \\ f(T^*) & if \quad 0 < T^* < 1 \\ 0 & if \quad 1 < T^* \end{cases} \tag{2880}$$

where T^* is the normalized temperature that is defined as:

$$T^* = \frac{T - T_{\text{solidus}}}{T_{\text{liquidus}} - T_{\text{solidus}}} \tag{2881}$$

The function $f(T^*)$ is called the fraction solid curve. For a linear dependence between $\alpha^*{}_s$ and T^*, the solidification path is defined as:

$$f\left(T^*\right) = 1 - T^* \tag{2882}$$

The solidification path can also be specified as a table, in which case linear dependencies are assumed between the points in the table. The values of temperature in the table have to increase monotonically as the values of the relative solid fraction decrease monotonically.

If melting and solidification take place at one temperature (that is, $T_{\text{solidus}} = T_{\text{liquidus}}$), a linear solidification path is assumed and a small temperature interval of 0.002 K is introduced automatically. The physical properties of the liquid-solid phase can be constant, or can be a function of temperature and the relative solid volume fraction.

Contents:

Flow Resistance due to Solidification

Flow Resistance due to Solidification

The melting and solidification model simulates only the effect of phase change on the energy equation; it does not modify the momentum equations. A further model is required to account for the change in material properties from the liquid state to the solid state.

Slurry Viscosity Model

In the early stages of solidification, small crystals nucleate and grow in the melt. These crystals move with the liquid but do not interact with each other. The mixture of the solid and liquid phases is called a slurry. However, the presence of the solid crystals in the liquid increases the viscosity of the two-phase mixture, which increases the flow resistance. This increased resistance is simulated using the Slurry Viscosity model.

The Slurry Viscosity model is used for low values of the solid volume fraction, $\alpha^*{}_s \leq \alpha^*_{\text{cr}}$, where the critical solid fraction α_{cr} defines the upper limit of applicability.

The method of Metzner ([536]) expresses the variation in viscosity of the mixture as a function of the solid volume fraction $\alpha^*{}_s$:

$$\mu^* = \mu_l \left[1 - \left(\frac{\alpha^* \cdot F_\mu(\alpha^*_s)}{A} \right) \right]^{-2} \tag{2883}$$

where:

- μ^* is the effective viscosity of the solid-liquid mixture
- μ_l is the dynamic viscosity of the liquid
- F_μ is a switching function that is given by Eqn. (2889)
- A is a crystal constant which depends on the aspect ratio and surface roughness of the crystal.

Mushy Zone Permeability Model

For high solid volume fractions, equiaxed grains grow and start to agglomerate and to form dendritic regions. Dendrites also grow from cooled solid surfaces or the solidification front into the melt. A partially solidified stationary region permeated with dendrites is called a “mushy zone”.

A common engineering application for this model is high-pressure die casting where melt is pressed through ducts with wall temperatures below solidus temperature at high velocities. The melt solidifies at the walls and dendrites grow from the solidification front into the melt. This partially solidified region causes an additional pressure loss in the duct that influences the melt mass flow. This pressure loss is simulated using the Mushy Zone Permeability model.

The *Carman-Kozeny Mushy Zone Permeability* method provides a mushy zone for the flow resistance, eliminating the need to set up porous media. In addition, the method updates the velocity field in the melting and solidification process. The flow resistance is active only in cells that have a solid volume fraction greater than zero and is applied to all regions in the physics continuum.

The flow resistance in a mushy zone can be modeled similar to an isotropic porous medium. Therefore, to quantify the permeability, the Carman-Kozeny equation [525] for flow through a porous medium is used:

$$K = \frac{(1-\alpha_s^*)^3}{\alpha_s^{*2} F_K c_s} \tag{2884}$$

where:

- $K\left[m^2\right]$ is the permeability
- α^*_s is the relative solid volume fraction
- F_K is a non-dimensional switching function that is given by Eqn. (2890).
- c_s is the shape factor:

$$c_s = c/d^2 \tag{2885}$$

where c is a shape constant with a default value of 180 and d is the secondary dendrite arm spacing.

The contribution of the mushy zone permeability to the momentum equation Eqn. (2793) enters the porous medium resistance force $\mathbf{f}_p$:

$$\mathbf{f}_p = -\mathbf{P}_{\text{v, Mushy Zone}} \cdot \mathbf{v} \tag{2886}$$

where $\mathbf{P}$ is the porous resistance tensor, which is given by:

$$\mathbf{P}_{\text{MushyZone}} = \left(\frac{\mu}{K} + \frac{C_F \rho_l \alpha_l^*}{\sqrt{K}} \left\|\mathbf{v}\right\|\right) \tag{2887}$$

$\mathbf{P}_{\text{v}}$ and $\mathbf{P}_{\text{i}}$ are the viscous (linear) and the inertial (quadratic) resistance tensors, respectively.

The mushy zone permeability contributes to $\mathbf{P}_{\text{v}}$ through:

$$\mathbf{P}_{\text{v, Mushy Zone}} = \left(\frac{\mu}{K} + \frac{C_F \rho_l \alpha^*{}_l}{\sqrt{K}} \left\|\mathbf{v}\right\|\right) \tag{2888}$$

where:

- $\frac{\mu}{K}$ is Darcy's term
- $\frac{C_F \rho_l \alpha^*{}_l}{\sqrt{K}} \left\|\mathbf{v}\right\|$ is Forchheimer's term

where:

- μ is the dynamic viscosity
- C_F is the Ergun coefficient
- ρ_l is the liquid phase density
- α_l^* is the liquid fraction that is given by $\alpha_l^* = \left(1 - \alpha_s^*\right)$.

$\mathbf{P}_{\text{v,MushyZone}}$ can be considered a drag coefficient. The drag coefficient is given by the Darcy's and Forchheimer's terms, which are often referred to as the "viscous-drag" and "form-drag" terms.

The Forchheimer's term is optional and its contribution to the mushy zone permeability can be deactivated. The Forchheimer's term must be considered when relatively high interdendritic fluid-flow velocities are expected. Such high velocities can be important during the intensification phase in die or squeeze casting.

Switching Function for Metzner Slurry Viscosity and Carman-Kozeny Mushy Zone Permeability Model

The Metzner slurry viscosity model and the Carman-Kozeny mushy zone permeability model are typically used in combination. Both models cover different ranges of solidification state. The applicability of each model depends on the extent to which the liquid has solidified.

The state of solidification is expressed using the relative solid volume fraction $\alpha^*{}_s$. The critical relative solid fraction α_{cr}^* separates the applicability ranges of the two models.

$\alpha_s^* \leq \alpha_{\text{cr}}^*$	Metzner Slurry Viscosity Model
$\alpha_{\text{cr}}^* < \alpha_s^*$	Carman-Kozeny Mushy Zone Permeability Model

For the Metzner slurry viscosity model and the Carman-Kozeny mushy zone model, a switching function ensures that each model is applied within its appropriate range of solid volume fraction. The switching function is applied as a multiplicative factor to the respective model equation (see Eqn. (2883) and Eqn. (2884)).

The switching function for the slurry viscosity model is:

$$F_\mu\left(\alpha^*{}_s\right) = 0.5 - \frac{\arctan\left[s\left(\alpha^*{}_s - \alpha^*{}_{\text{cr}}\right)\right]}{\pi} \tag{2889}$$

where s is a switching function constant.

The switching function for the mushy zone permeability model is given by:

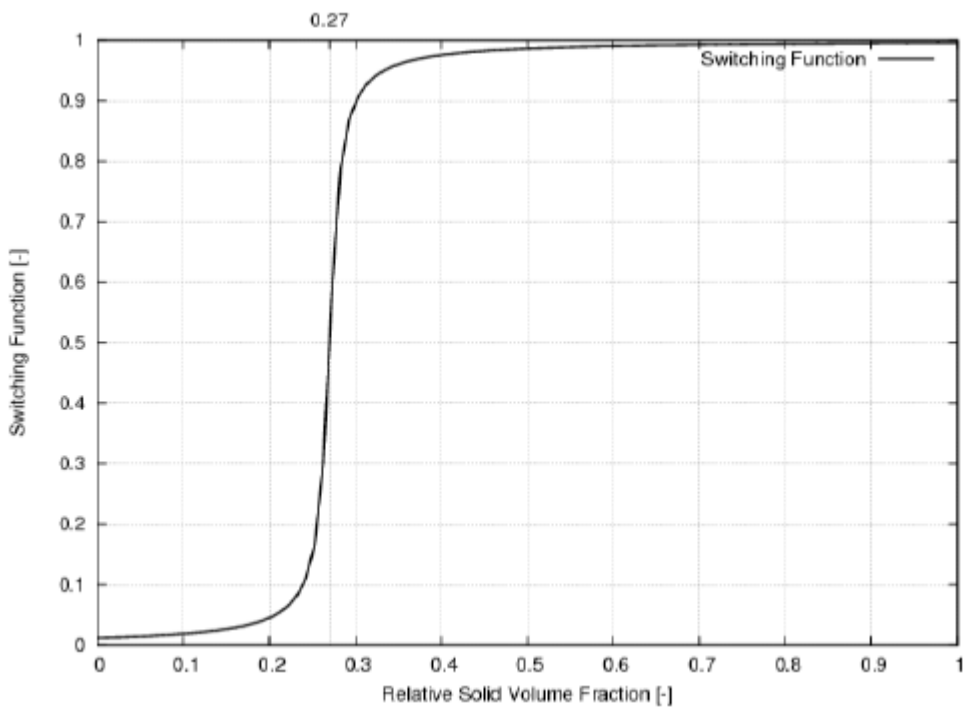
$$F_K(\alpha^*{}_s) = 1 - F_\mu(\alpha^*{}_s) = 0.5 + \frac{\arctan[s(\alpha^*{}_s - \alpha^*{}_{\mathrm{cr}})]}{\pi} \tag{2890}$$

The switching function has a smooth transition near the critical relative solid volume fraction $\alpha^*{}_{cr}$(see the figure below) where solid grains start to agglomerate to a significant extent:

The default value of the non-dimensional constant s is 100 because the shape is rather insensitive for values from 80 through 120. The default value for the critical solid volume fraction $\alpha^*{}_{cr}$ is 0.27 according to Baeckerud and others [521] for Aluminum alloy A201.

Oldenburger and Spera [542] initially suggested this switching function. The switching function that is implemented in STAR-CCM+ is a slightly modified formulation that is based on the model of Chang and Stefanescu [527].

The following figure depicts the switching function F_K for critical solid volume fraction $\alpha^*_{cr} = 0.27$ (default value).



VOF Bibliography

[521] *Baeckerud, L., Chai, G. and Tamminen, J. 1992. Solidification Characteristics of Aluminum Alloys 2, American Foundry Society, Inc. Des Plaines, Illinois.*

[522] *Brackbill, J. U., Kothe, D. B., and Zemach, C. 1992. A Continuum method for modeling surface tension, J. Comp. Physics, 100, pp. 335-354.*

[523] *Brennen, C.E., Cavitation and Bubble Dynamics, Oxford University Press, 1995.*

[524] *Butterworth, D. and Hewitt, G.F. 1974. “Two-Phase Flow and Heat Transfer”, Oxford University Press, Oxford.*

[525] *Carman, P. C. 1937. “Fluid Flow Through Granular Beds”, Transactions of the Institution of Chemical Engineers, 15, pp. 150-166.*

[526] *Chan, S.H., Cho, D.H., Kocamustafaogullari, G. 1983. "Melting and solidification with internal radiative transfer - a generalized phase change model", International Journal of Heat and Mass Transfer, 26(4), pp. 621-633.*

[527] *Chang, S. and Stefanescu, D. M. 1996. “A Model for Macrosegregation and Its Application to Al-Cu Castings”, Metallurgical and Materials Transactions, 27A, pp. 2708-2721.*

[528] *Criscione, A., 2011. "Numerical investigation of impacting water drops in air crossflow". In: 24th European Conference on Liquid Atomization and Spray Systems, Estoril, Portugal.*

[529] *Ellion, M.E. 1953. “A study of the mechanism of boiling heat transfer”, Ph.D. Dissertation, California Institute of Technology.*

[530] *Epstein, P., Plesset, M., On the stability of gas bubbles in liquid-gas solutions, J. Chemical Physics, Vol. 18, No. 11 (1985).*

[531] *Fujimoto, H. 2007, "Three-dimensional numerical analysis of the deformation behavior of droplets impinging onto a solid substrate". Int. J. Multiphase Flow, 33: p. 317-332.*

[532] *Gopalakrishnan, S. and Schmidt, David P., 2009. "A Computational Study of Flashing Flow in Fuel Injector Nozzles", SAE Int. Journal of Engines. 1(1), pp. 160−170.*

[533] *Hardt, S., and Wondra, F. Evaporation model for interfacial flows based on a continuum-field representation of the source terms. Journal of Computational Physics, 227(10):5871–5895, 2008.*

[534] *Landau, L.D., Lifshitz, E.M., Fluid Mechanics, Second Edition: Volume 6 (Course of Theoretical Physics), 1984.*

[535] *Meredith, K.V., Heather, A., de Vries, J. & Xin, Y., A numerical model for partially-wetted flow of thin liquid films. Computational Methods in Multiphase Flow, VI, pp. 239-250, 2011.*

[536] *Metzner, A. B. 1985. “Rheology of Suspensions in Polymeric Liquids”, Journal of Rheology, 29(6), pp. 739-775.*

[537] *Muzaferija, S. and Peric, M. 1999. Computation of free surface flows using interface-tracking and interface-capturing methods, Chap. 2 in O. Mahrenholtz and M. Markiewicz (eds.), Nonlinear Water Wave Interaction, Computational Mechanics Publications, WIT Press, Southampton.*

[538] *Muzaferija, S., Peric, M., Simulation of evolution of dissolved gas in liquids, Internal document.*

[539] *Nemec, T., Marsik, F., The Classical Multicomponent Nucleation Theory for Cavitation in Water with Dissolved Gases, Proc. 7th Int. Symp. on Cavitation, Paper No. 120, (2009)*

[540] *Neroorkar, K.D., Gopalakrishnan, S., Schmidt, David P., and Grover, R.O. Jr., 2011. "Simulation of FlashBoiling in Pressure Swirl Injectors", Atomization and Sprays, 21(2), pp. 179−188.*

[541] *Ni, B., Pieprzak, J., Transportation and transformation of air bubbles in aerated oil through an engine lubrication system, SAE-Paper 2004-01-2915.*

[542] *Oldenburg, C. M., and Spera, F. J. 1992. “Hybrid Model for Solidification and Convection”, Numerical Heat Transfer, Part B, 21, pp. 217-229.*

[543] *Rohsenow, W.M. 1952. “A Method of Correlation Heat Transfer Data for Surface Boiling of Liquid”, Trans. ASME, 74, 969.*

[544] *Saiz Jabardo, J.M., Fockink da Silva, E., Ribatski, G. and de Barros, S.F. 2004. “Evaluation of the Rohsenow Correlation Through Experimental Pool Boiling of Halocarbon Refrigerants on Cylindrical Surfaces”, J. of the Braz. Soc. of Mech. Sci. & Eng, XXVI, No. 2.*

[545] *Sauer, J., Instationaer kavitierende Stroemungen - Ein neues Modell, basierend auf Front Capturing VOF und Blasendynamik, Dissertation, Universitaet Kalrsruhe, 2000.*

[546] *Sazhin, S.S. Advanced models of fuel droplet heating and evaporation. Progress in Energy and Combustion Science, 32:162–214, 2006.*

[547] *Schlottke, J., and Weigand, B. Direct numerical simulation of evaporating droplets. Journal of Computational Physics, 227(10):5215–5237, 2008.*

[548] *Sikalo, S. 2005, "Dynamic contact angle of spreading droplets: Experiments and simulations". In: Physics of Fluids, 17, 062103.*

[549] *Singhal, A.K., Athavale, M.M., Li, H., Jiang, Y., Mathematical Basis and Validation of the Full Cavitation Model, J. Fluids Engineering, Vol. 124 (2002).*

[550] *Swaminathan, C.R. and Voller, V.R. 1992 “A general enthalpy method for modeling solidification processes”, Met. Trans. B, 22(B), pp. 651-664.*

[551] *Teskeredzic, A., Demirdzic, I. and Muzaferija, S. 2002. “Numerical method for heat transfer, fluid flow, and stress analysis in phase-change problems”, Numerical Heat Transfer, Part B, 42, pp. 437-459.*

[552] *Voller, V.R. 1997 “An overview of numerical methods for solving phase change problems”, in: Advances in Numerical Heat Transfer, W.J. Minkowycz and E.M. Sparrow Eds., Taylor & Francis, pp. 341-378.*

[553] *Yokoi, K. 2009, "Numerical studies of the influence of the dynamic contact angle on a droplet impacting on a dry surface". In: Physics of Fluids, 21, 072102.*

Fluid Film

The fluid film model in STAR-CCM+ is used for modeling the distribution and transport of a thin layer of liquid —a fluid film— on solid surfaces. Application areas include vehicle rainwater management, selective catalytic reduction (SCR), and lubrication.

In STAR-CCM+, a fluid film can form and accumulate due to several mechanisms:

- Droplet impingement, which requires interaction with other multiphase modeling approaches such as Lagrangian multiphase, discrete multiphase (DMP), and multiphase segregated flow.
- Resolved fluid film simulation where a liquid film or pool of liquid that is modeled using the Volume-of-Fluid multiphase model transforms into liquid film that is modeled using the fluid film model
- Condensation from the gas phase
- Inflow
- Initial film thickness specification
- User source

The fluid film can reduce in thickness or be removed through the following mechanisms:

- Wave stripping where droplets are stripped due to instabilities in waves in the fluid film. This mechanism requires interaction with the Lagrangian multiphase model.
- Edge stripping where droplets are stripped when the fluid film flows over a sharp edge. This mechanism requires interaction with the Lagrangian multiphase model.
- Resolved fluid film simulation
- Evaporation
- Boiling

- Outflow

The fluid film model solves transport equations for mass, momentum, energy, species, and volume fraction. The model assumes that the film is thin enough for the laminar boundary layer approximation to apply. If turbulence occurs within the film, the assumption is violated. It further assumes a parabolic velocity profile across the film.

Continuity Equation

The equation that governs the film mass conservation is as follows:

$$\frac{d}{dt}\int_V \rho_f dV + \int_A \rho_f \mathbf{v}_f \cdot d\mathbf{a} = \int_V \frac{S_u}{h_f} dV \tag{2891}$$

This equation is used to calculate the film thickness h_f. It is specific in the sense that the volume V and the surface A are functions of the film thickness and its spatial distribution.

Here, ρ_f is the film density, $\mathbf{v}_f$ is the film velocity, and the subscript f denotes the fluid film values.

The quantity s_u is the mass source/sink per unit area. Contributions to the mass source or sink come from droplet impingement, film stripping, mass transfer, or user-defined sources.

Momentum Equation

The momentum conservation equation is:

$$\frac{d}{dt}\int_V \rho_f \mathbf{v}_f dV + \int_A \rho_f \mathbf{v}_f \otimes \mathbf{v}_f \cdot d\mathbf{a} = \int_A \mathbf{T}_f \cdot d\mathbf{a} - \int_A p_f d\mathbf{a} + \int_V \left(\rho_f \mathbf{f}_b + \frac{\mathbf{s}_u}{h_f}\right) dV \tag{2892}$$

where $\mathbf{s}_u$ is the momentum source corresponding to the mass source S_u, p_f is the pressure, $\mathbf{f}_b$ is the body force (for example, gravity, or the pseudo-force in a moving reference frame), and $\mathbf{T}_f$ is the viscous stress tensor within the film. The kinematic and dynamic conditions at the interface between the film and the surrounding fluid (film free surface) are satisfied:

$$\left(\mathbf{v}_f\right)_{\text{int}} = (\mathbf{v})_{\text{int}} \tag{2893}$$

$$\left(\mathbf{T}_f \cdot d\mathbf{a} + p_f d\mathbf{a}\right)_{\text{int}} = (\mathbf{T} \cdot d\mathbf{a} + p d\mathbf{a})_{\text{int}} \tag{2894}$$

The quantities without the subscript f are associated with the surrounding fluid. Assuming that the normal components of the viscous and convective terms are negligible, the pressure distribution within the fluid film is obtained from Eqn. (2892) as:

$$p_f(\xi) = p_{\text{int}} - \mathbf{s}_m \cdot \mathbf{n} - \rho_f \mathbf{f}_b \cdot \mathbf{n}\left(h_f - \xi\right) + \int_\xi^{h_f} \frac{d}{dt}\left(\rho_f \mathbf{v}_f \cdot \mathbf{n}\right) \mathrm{d}\,\xi \tag{2895}$$

where $\mathbf{n}$ is the wall surface unit vector pointing towards the film, and ξ is the local coordinate normal to the wall. Eqn. (2895) assumes that the force $\mathbf{s}_u$ is applied at the film free surface.

Energy Equation

The energy conservation equation is:

$$\frac{d}{dt}\int_V \rho_f E_f dV + \int_A \left[\rho_f H_f \mathbf{v}_f \cdot d\mathbf{a} = \int_A \mathbf{q}''_f \cdot d\mathbf{a} + \int_A \mathbf{T}_f \cdot \mathbf{v}_f d\mathbf{a} + \int_V \mathbf{f}_b \cdot \mathbf{v}_f dV + \int_V \frac{S_E}{h_f} dV \tag{2896}$$

where E_f is the film total energy, H_f is the film total enthalpy, $\mathbf{q}''_f$ is the film heat flux, $\mathbf{T}_f$ is the fluid viscous stress tensor, and $\mathbf{f}_b$ is the body force. S_E is the energy source/sink term per unit film area. Contributions to the energy source or sink come from droplet impingement, film stripping, or user-defined sources.

Species Mass Conservation Equation

For a multi-component film, the species mass conservation equation of species i is solved:

$$\frac{d}{dt}\int_V \rho_f Y_{i,f} dV + \int_A \rho_f \mathbf{v}_f Y_{i,f} \cdot d\mathbf{a} = \int_A \left(\frac{\mu_f}{\sigma}\right) \nabla Y_{i,f} \cdot d\mathbf{a} + \int_V \frac{S_{u,i}}{h_f} dV \tag{2897}$$

where $Y_{i,f}$ is the mass fraction of species i, and σ is the molecular Schmidt number. The quantity $S_{m,i}$ is the mass source or sink for species i.

Volume Fraction

The volume that the fluid film occupies is subtracted from the volume of the gas phase in the cells adjacent to the film. The volume fraction of the film, in the neighboring gas cell, is computed as:

$$\alpha_f = \min\left(\frac{V_f}{V}, \alpha_{f,\max}\right) \tag{2898}$$

where V_f is the film volume (the area multiplied by the film height), V is the volume of the neighboring cell, and $\alpha_{f,\max}$ is the maximum volume fraction.

Contents:

Surface Tension
Impingement
Wave Stripping
Edge Stripping
Evaporation and Condensation
Boiling
Melting and Solidification
Resolved Fluid Film
Fluid Film Bibliography

Surface Tension

At the interface between a fluid film and the gas phase, surface tension is caused by the greater attraction of fluid molecules to each other than to the gas molecules. The net effect is an inward force at the interface that causes the fluid film to behave as if its surface were covered with a stretched elastic membrane: that is, the surface is under tension. This tension is expressed using the experimentally determined surface tension coefficient σ.

The surface tension formulation that is employed in the fluid film phase is based on the model that is introduced in [535]. There, the surface tension effects in the fluid film momentum transport are separated into two terms: capillary pressure and contact line force.

The capillary pressure, p_σ, is expressed as:

$$p_\sigma = -a\sigma\left(\nabla_s^2\delta\right) \tag{2899}$$

where:

- σ is the surface tension coefficient.
- $\nabla_s^2\delta$ approximates the curvature of the liquid surface with δ being the fluid film thickness.
- a is a non-dimensional scale factor.

This force term has units of N/m^2 and acts orthogonal to the bottom surface. This relationship is valid only for surfaces with slight curvature [534].

The contact line force is calculated as:

$$\tau_\sigma = b\sigma(1-\cos(\theta))\nabla w \tag{2900}$$

where:

- θ describes the contact angle as a fundamental parameter of the wetting behavior of the film.
- w is defined as 1 wherever $\delta > \delta_{\min}$ (a user-specified minimum value) and 0 elsewhere.
- b is an empirical parameter that is used to calibrate the model with respect to experimental results.

This force term has units of N/m^2 and acts tangential to the bottom surface.

The resulting surface tension force that is contributed to the momentum transport equation of the liquid film is:

$$\mathbf{f}_\sigma = p_\sigma + \tau_\sigma \tag{2901}$$

Impingement

Liquid droplets can impinge on a dry wall to form a fluid film or they can impinge on an existing fluid film. In both cases, the mass, the momentum, and the energy of the droplets are transferred to the film.

For the Bai-Gosman and the Bai-Onera wall impingement models, the formation of fluid film from droplets depends on different impingement regimes. The droplets can exhibit behaviours such as adherence, rebound, spread, or splash. Depending on droplet behaviour, only part of droplet mass, momentum, and energy might be transferred to the fluid film.

Inertial Impingement

The inertial impingement model uses the inertia of the dispersed droplets that approach the wall to compute the mass flux of the droplets that impinge on the wall to form a fluid film. Impingement occurs on the walls at which a shell interface has been created. The inertial impingement model is only applicable when the dispersed phase is calculated by the multiphase segregated flow model.

The mass flux due to impingement is:

$$m_{\text{imp}} = \gamma\rho_{\text{imp}}\mathbf{v}_{\text{imp}} \cdot \mathbf{a}\alpha_{\text{imp}} \tag{2902}$$

where:

- m_{imp} is the mass flux leaving the system.

- γ is the impingement efficiency.
- ρ_{imp} is the cell-centered density.
- $\mathbf{v}_{\text{imp}}$ is the cell-centered velocity.
- $\mathbf{a}$ is the cell wall area vector (normal to the wall in the outward facing direction).
- α_{imp} is the cell-centered volume fraction.

The impingement efficiency specifies the fraction of the flux incident on the surface that impinges on the fluid film. The impingement efficiency can be a value from 0 through 1.

The kinematic and dynamic conditions at the interface between the fluid film and the surrounding multiphase fluid (film free surface) are satisfied. The impinging mass flux that is calculated at the interface cell is used in all of the other transport equations.

The liquid phase impinging on the fluid-film interface must have the exact same material components as the fluid-film phase.

Wave Stripping

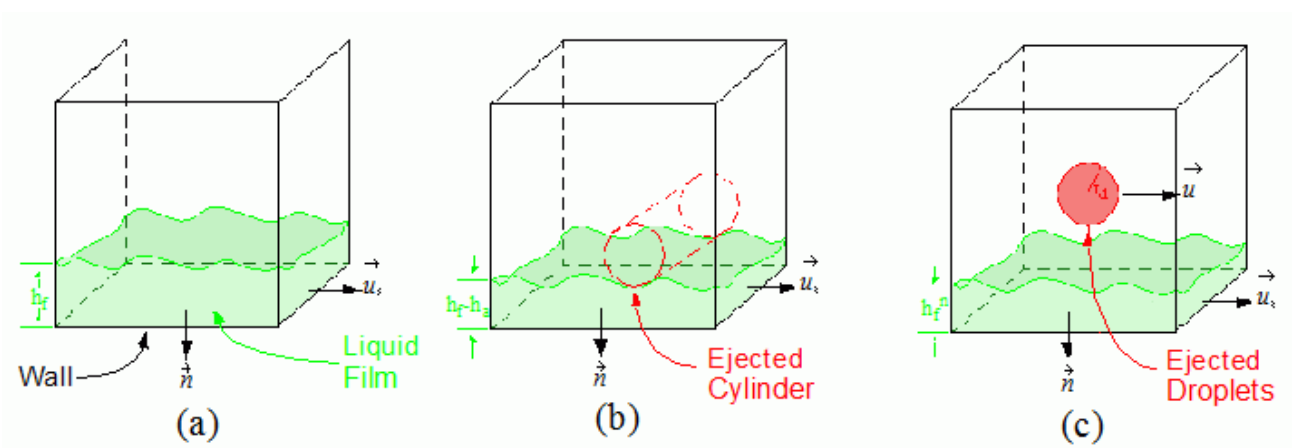
Depending on flow conditions and the geometry of the wall on which the film resides, the film can form droplets and this process is called film stripping.

Two stripping mechanisms are modeled:

- Stripping due to wave instability generated by an adjacent fluid flow
- Stripping due to body-force induced instability (gravity, film acceleration)

Adjacent Flow and Body-Force Induced Instability

Due to external forces such as gravity or shearing at the interface between a fluid film and its surrounding fluid, the film surface can develop instabilities and start shedding droplets. The model described here is based on the original documentation of Hervé Foucart [556]. It postulates that stripping at the surface occurs in a three-stage process, as represented schematically in the following illustration of the liquid film stripping mechanism.



First, waves develop at the liquid-gas interface. As these waves grow, they become unstable and form cylinders of liquid which are ejected into the surrounding gas. The cylinders then break up further forming spherical droplets. The model is closed by computing the radius of the child droplets as a function of the most unstable wavelength characterizing the surface instability.

It is accepted that for surface stripping:

- Ejection occurs when critical conditions are satisfied, that is a minimum height is required for droplet ejection.
- The ejected fluid volume forms a cylinder, as a result of the Kelvin-Helmholtz instability.
- The cylinder then breaks into droplets due to Rayleigh-Taylor instabilities.

Having characterized surface stripping as a function of wavelength, the stripping model relationships are now closed by computing the wavelength for which surface break-up is most likely to occur. This wavelength is the most unstable, and is calculated as the resonance wavelength using the dispersion equation. Solving the dispersion equation for surface waves, the resonance wavelength is obtained as:

$$\lambda_{\text{res}} = \frac{2\pi}{\rho_f \mathbf{f}_b \cdot \mathbf{n}}\left(\frac{1}{3}\rho\tilde{\mathbf{v}}_r^2 - \sqrt{\left(\frac{1}{3}\rho\tilde{\mathbf{v}}_r^2\right)^2 - \rho_f \mathbf{f}_b \cdot \mathbf{n}\sigma}\right) \tag{2903}$$

where $\tilde{\mathbf{v}}_r^2$ is the square of relative velocity between the film and the surrounding fluid $\left(\mathbf{v}_f - \mathbf{v}\right) \cdot \left(\mathbf{v}_f - \mathbf{v}\right)$, σ is the surface tension, and $\mathbf{f}_b$ is the body force acting on film which also includes the inertial force. The minimum height necessary for droplet ejection is then:

$$h_{\text{min}} = c_H \frac{\lambda_{\text{res}}}{2\pi} \tag{2904}$$

The film height to strip, that is the surface wave amplitude, is:

$$h_a = \left[\frac{3}{4}\left(\frac{2}{3.78}\right)^3\right]^2 \pi\lambda_{\text{res}} \tag{2905}$$

The resulting droplet diameter is:

$$D_d = c_D\sqrt{\frac{\lambda_{\text{res}} h_a}{\pi}} \tag{2906}$$

Note: For wave stripping into a dispersed phase, Eqn. (2906) is not used. The droplet diameter is specified manually rather than being computed.

The default setting for c_D is 3.78. The number 3.78 in Eqn. (2905) arises from Rayleigh theory. It means that the diameter of each droplet that is generated from the cylinder is 3.78 times the diameter of that cylinder.

All droplets initially have the same velocity as the film and are placed at a point between the cell center and the boundary face center, depending on the radius of the ejected droplet.

Although no consideration of time scale is included in the original stripping model, the time scale can be defined in terms of a linear growth rate as:

$$t_b = \frac{\lambda_{\text{res}}\left(\rho_f + \rho\right)}{\left|\mathbf{v}_r\right|\sqrt{(1 - \text{We})\rho_f \rho}} \tag{2907}$$

where We is the Weber number. In Eqn. (2907), $\mathrm{We} = 2/3$, which is when the most unstable mode occurs.

Using this time scale, stripping is expected to occur n_b times during a time-step Δt:

$$n_b = c_B \frac{\Delta t}{t_b} \tag{2908}$$

Free-Stream Velocity

The wave stripping model requires a measure of the free-stream velocity ($\mathbf{v}_r$ as used in Eqn. (2903)).

STAR-CCM+ calculates two measures of the free-stream velocity and uses the highest.

- One measure is the gas velocity in the nearest gas cell. The disadvantage with this measure is that the value depends on the mesh size, and the approximation becomes less valid as the mesh size decreases.
- A second measure is a mesh-independent estimate of the free-stream velocity based on the shear velocity. This second measure is derived below.

At the free surface, on the gas side, the shear (or frictional) velocity is defined as:

$$\mathbf{v}_* = -\sqrt{\frac{\tau}{\rho}}\mathbf{n}_\tau \tag{2909}$$

where $\mathbf{n}_\tau$ is the unit vector in the direction of the shear stress τ.

It is assumed that the free-stream gas velocity can be reasonably approximated as:

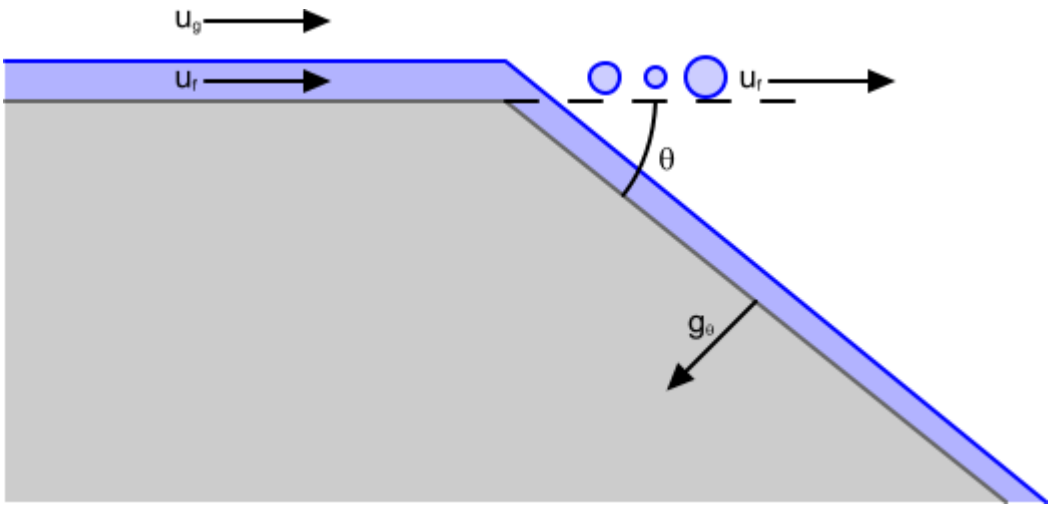
$$\mathbf{v}_r \approx (\mathbf{v})_{\mathrm{int}} + K_{*,\infty}\mathbf{v}_* = \mathbf{v} \tag{2910}$$

where $(\mathbf{v})_{\mathrm{int}}$ is the interfacial velocity and $K_{*,\infty}$ is a constant that relates the shear velocity to the free-stream velocity. This constant is specified as the *Gas Velocity Scaling Factor* in the Wave Stripping model properties.

Edge Stripping

Edge stripping models the break-up of the liquid film over a sharp edge.

Maroteaux et al. [561] proposed a model for the break-up of liquid films over a sharp edge. This model is based on the Rayleigh-Taylor instability and contains the break-up criterion and resulting droplet diameter distribution. However, Gubaidullin [558] has raised questions about the validity of the breakup criterion. Therefore, an alternative criterion is adopted, proposed by Friedrich et al. [557], that is based on a force balance. The droplet diameter distribution still follows Maroteaux.



The model is only applied at edges where the corner angle (θ in the diagram above) is greater than the user-defined minimum corner angle, θ_{min}. If a boundary cell has multiple sharp edges, stripping occurs over all of them in a weighted manner.

The force ratio FR, which is essentially the ratio of the film momentum flux to the surface tension and gravity force, is:

$$FR = \frac{We_f}{1 + \frac{1}{\sin\theta} + Bo_f \frac{L_b}{h_f \sin\theta}} \tag{2911}$$

where We_f is the film Weber number, Bo_f is the Bond number, L_b is the break-up length and h_f is the film thickness.

The film Weber number is:

$$We_f = \frac{\rho_f {v_f}^2 h_f}{\sigma} \tag{2912}$$

where ρ_f is film density, v_f is film velocity projected orthogonal to the stripping edge, h_f is film thickness and σ is surface tension coefficient.

The film Bond number is:

$$Bo_f = \frac{\rho_f g_\theta {h_f}^2}{\sigma} \tag{2913}$$

where g_θ is the component of acceleration normal to the downstream wall.

Arai and Hashimoto [554] give the break-up length as:

$$L_b = 0.0388 {h_f}^{0.5} {Re_f}^{0.6} {We_{rel}}^{-0.5} \tag{2914}$$

where Re_f is the film Reynolds number and We_{rel} is the relative Weber number.

The film Reynolds number is:

$$Re_f = \frac{\rho_f v_f h_f}{\mu_f} \tag{2915}$$

where μ_f is the film dynamic viscosity.

The relative Weber number is:

$$We_{rel} = \frac{h_f \rho \left(v_g - v_f\right)^2}{2\sigma} \tag{2916}$$

where ρ is the gas density and v_g is the gas velocity projected orthogonally to the stripping edge.

The break-up is deemed to occur when $FR > FR_C$, where FR_C is the user-defined critical force ratio, which is proposed in [557]. The critical force ratio default value is 1. Of the fluid crossing the edge, only a fraction x_S separates from the film. This fraction is approximated using the formula below:

$$x_S = \begin{cases} 0 & FR \leq FR_C \\ 0.44\left(FR - FR_C\right) & FR_c < FR \leq \left(FR_C + 1.6\right) \\ 0.057\left(FR - FR_C - 1.6\right) + 0.704 & \left(FR_C + 1.6\right) < FR \leq \left(FR_c + 6.792\right) \\ 1 & \left(FR_C + 6.792\right) < FR \end{cases} \tag{2917}$$

which is based on the experimental data in [557].

Note: For edge stripping into a dispersed phase, Eqn. (2918) to Eqn. (2923) are not used. The droplet diameter is specified manually rather than being computed.

For droplet formation, the model that is presented in [561] is adopted, which estimates the parent droplet diameter D_d as:

$$D_d = c_1 \sqrt{\frac{\lambda h_f}{\pi}} \tag{2918}$$

where c_1 is a user-defined droplet diameter scale factor. This factor is set to 3.78 by default. The wave length λ is calculated from the most unstable wavenumber k, which maximizes the growth rate:

$$\omega = -\left(\frac{\sigma - \left(\rho_f - \rho_g\right) a / k^2}{2\mu_f h_f}\right)\left(\frac{\left(kh_f\right)\sinh\left(kh_f\right)\cosh\left(kh_f\right) - k^2 h_f^2}{\cosh^2\left(kh_f\right) + k^2 h_f^2}\right) \tag{2919}$$

where the acceleration a is computed from:

$$a = \frac{v_f^2 \theta}{h_f(\pi + \theta)} \tag{2920}$$

and λ is related to k by $\lambda = 2\pi / k$. You can choose to provide an upper limit for the parent droplet diameter D_d by activating the Limit parent droplet diameter property and supplying a value for the maximum diameter.

According to the model, the cumulative droplet size distribution $F(D)$ is a Rosin-Rammler distribution:

$$F(D) = 1 - e^{-(D/X)^q} \tag{2921}$$

where q is a user-defined parameter, which is set to 1.5 by default, and:

$$X = \frac{D_d}{(3\ln 10)^{1/q}} \tag{2922}$$

There are two options for determining the droplet diameter:

- Generate droplets with a deterministic diameter, namely the average of the size distribution that is:

$$D = X\Gamma\left(1 + \frac{1}{q}\right) \tag{2923}$$

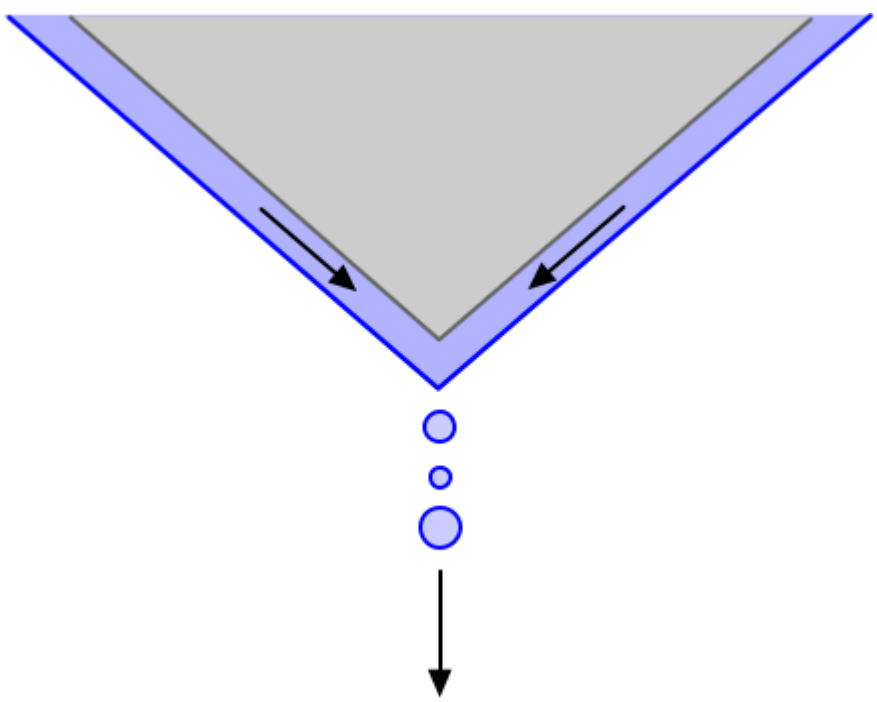
where $\Gamma(x)$ represents the Gamma function.

- Generate droplets with a randomized diameter D according to Eqn. (2921).

The droplets are ejected at the film velocity speeds in terms of both magnitude and direction.

Edge Stripping with Flow from Both Sides

For a case where flow comes toward an edge from both sides, the flows are combined into one stream of Lagrangian droplets that conserves mass, momentum, species, and energy. For example, if both flows have a different temperature, the droplets get a temperature somewhere in between. For opposing flows, the stripping fraction x_S (Eqn. (2917)) is also set to 1 for both sides (which is full stripping). Droplet diameter is computed based on the film properties of the side with the biggest flux. The following diagram illustrates opposing-flow edge stripping.



Evaporation and Condensation

STAR-CCM+ models evaporation of the fluid film into a gas phase and condensation from a gas phase into the fluid film.

The species mass flux for every component i is conserved at the interface between the gas and the fluid film, such that:

$$\rho Y_i\left(v-\dot{h}\right)-\rho D_i\frac{dY_i}{dy}=\rho_f Y_{f,i}\left(v_f-\dot{h}_f\right)-\rho_f D_{f,i}\frac{dY_i}{\mathrm{dy}}\bigg|_f \tag{2924}$$

where, evaluated at the interface:

- ρ and ρ_f are the gas and liquid film densities
- Y_i and $Y_{f,i}$ are the mass fractions for the gas and liquid film
- v and v_f are the normal velocity components for the gas and liquid film
- D_i and $D_{f,i}$ are the gas and liquid film molecular diffusion coefficients
- $\dot{h}$ is the rate of change of film thickness

The total mass flux is also conserved:

$$\rho\left(v-\dot{h}\right)=\rho_f\left(v_f-\dot{h}\right) \tag{2925}$$

Combining Eqn. (2924) and Eqn. (2925):

$$\rho_f\left(v_f-\dot{h}\right)\left(Y_i-Y_{\mathrm{f,i}}\right)+\rho_f D_{f,i}\frac{\mathrm{d}\,Y_i}{\mathrm{d}\,y}\bigg|_f-\rho D_i\frac{\mathrm{d}\,Y_i}{\mathrm{d}\,y}=0 \tag{2926}$$

The evaporation rate is defined as:

$$\dot{m}_v=-\rho_f\dot{h} \tag{2927}$$

with v_f considered to be zero from here on. The summation of Eqn. (2926) over all liquid film component N_L becomes:

$$\left[1-\sum_j^{N_L}Y_j\right]\dot{m}_v=-\sum_j^{N_L}\rho D_i\frac{dY_i}{dy} \tag{2928}$$

and since

$$-Y_j\dot{m}_v=-\rho D_i\frac{dY_i}{dy} \tag{2929}$$

for all $N_{P,L}$ species i in the film that are inert, the summation can be limited to the N_v interacting components. The total evaporation rate can therefore be expressed as:

$$\dot{m}_v=-\frac{\sum_j^{N_v}\rho D_i\frac{\mathrm{d}\,Y_i}{\mathrm{d}\,y}}{1-\sum_j^{N_v}Y_j} \tag{2930}$$

Note that Eqn. (2930) is valid only below saturation conditions. That is:

$$\sum_{j}^{N_v} Y_j < 1 \tag{2931}$$

At saturation conditions, the algebra that is used to derive Eqn. (2930) is not correct.

The normal derivative is treated through the species transfer coefficients $s_{t,i}$ and the Spalding transfer number B such that:

$$\dot{m}_v = -\frac{\sum_j^{N_v} s_{t,j}\left(Y_{c,j} - Y_j\right)}{1 - \sum_j^{N_v} Y_j} \cdot \frac{\ln(1+B)}{B} \tag{2932}$$

where the subscript c indicates a cell value and B is defined as:

$$B = \frac{\sum_j^{N_v} Y_j - \sum_j^{N_v} Y_\infty}{1 - \sum_j^{N_v} Y_j} \tag{2933}$$

where it is assumed that $Y_\infty \approx Y_c$

When the gas phase is a VOF phase, $\dot{m}_v$ is multiplied by the volume fraction of the VOF phase.

The interfacial gas mass fraction Y_j is required to evaluate Eqn. (2932). Y_j is related to Y_j through Raoult's law, as follows:

Consider the vapor pressure p_i of a component:

$$p_i = a_i p^*_i \tag{2934}$$

where p^*_i is the vapor pressure of the pure component i and a_i is the activity, which is dependent on the liquid mixture. This expression can be written as:

$$p_i = \gamma_i X_{f,i} p^*_i\left(T_s\right) \tag{2935}$$

Raoult's law states that $\gamma_i \approx 1$, an approximation that is followed here. Given the components vapor pressures, their molar fractions at the interface can be derived from:

$$X_i = \frac{p_i}{p} \tag{2936}$$

where p is the ambient pressure. The conversion to interface mass fraction is obtained using:

$$Y_i = \frac{X_i W_i}{\sum_j^{N_v} X_j W_j + \sum_j^{N_{P,G}} X_j W_j} \tag{2937}$$

While X_i is known for the evaporation components, it is unknown for the inert species. A good approximation is the introduction of a background molar weight, which is calculated at a small distance away from the interface:

$$W_{bg} = \frac{\sum_j^{N_{P,G}} X_j W_j}{\sum_j^{N_{P,G}} X_j} \tag{2938}$$

The interfacial background molar fraction is:

$$X_{bg} = 1 - \sum_j^{N_v} X_j \tag{2939}$$

The interfacial vapor mass fraction can then be approximated as:

$$Y_i \approx \frac{X_i W_i}{X_{bg} W_{bg} + \sum_j^{N_v} X_j W_j} \tag{2940}$$

Eqn. (2935) assumes knowledge of $X_{f,i}$. Presently, the normal species profile in the film is approximated as a constant, such that the interfacial molar fractions follow from the associated cell center values:

$$X_{f,i} = X_{f_c,i} \tag{2941}$$

The component evaporation rate can be expressed as:

$$\dot{m}_{v,\,i} = Y_i \dot{m}_v - \rho D_i \frac{\mathrm{d}\, Y_i}{\mathrm{d}\, y} \tag{2942}$$

which is valid under all conditions.

The interfacial heat flux conservation is expressed as:

$$k_g \frac{dT}{dy} - k_l \frac{dT}{dy}\bigg|_f - \dot{Q}_v = 0 \tag{2943}$$

where k denotes thermal conductivity and

$$\dot{Q}_v = \sum_i^{N_v} \Delta H_i^{vap} \dot{m}_{v,i} \tag{2944}$$

which is a condition for the interfacial surface temperature T_s.

Note that a combination of Eqn. (2942) and Eqn. (2944) leads to an expression for the total evaporation rate which is valid at all conditions:

$$\dot{m}_v = \frac{\dot{Q}_v + \sum_i^{N_v} \Delta H_i^{vap} \rho D_i \frac{dY_i}{dy}}{\sum_i^{N_v} \Delta H_i^{vap} Y_i} \tag{2945}$$

Boiling

Boiling is considered when the temperature of the fluid film is above the boiling temperature and the boiling property is activated. Below the boiling temperature, the same equilibrium conditions as for evaporation are assumed, but the vapor mass rates are determined differently.

Bulk Boiling

The heat transfer to the interface between the phases is used to compute the mass rate of evaporation when film cells are boiling. It is assumed that:

- The vapor bubbles are at the saturation temperature.
- The temperature of the liquid is approximately the temperature of the mixture.
- All the heat flux from the liquid to the interface is used in mass transfer due to evaporation.

$$\dot{m}_v = \frac{\mathrm{HTCxArea}\left(T - T_{sat}\right)}{\Delta H^{vap}} \tag{2946}$$

$\mathrm{HTCxArea}$ is the product of the heat transfer coefficient between the vapor bubbles and the surrounding liquid, and the specific contact area (contact area per unit volume) between the two. ΔH^{vap} is the latent heat of vaporization of the mixture and computes as:

$$\Delta H^{vap} = \sum_i Y_i \, \Delta H_i^{vap} \tag{2947}$$

where Y_i is the mass fraction and ΔH_i^{vap} is the latent heat of pure component i. This latent heat is the difference between the total enthalpy of its liquid phase $H_{i,f}$ and its gas phase H_i.

$$\Delta H_i^{vap} = H_i - H_{i,f} \tag{2948}$$

Wall Boiling

The empirical standard pool boiling correlation [564] is used to model liquid film boiling at a hot wall. The film starts to boil when the wall temperature exceeds the boiling temperature of the liquid. The heat flux passing from the wall to the film increases in proportion to the cubic power of $\Delta T_W = T_W - T_{sat}$ as:

$$q_{boil} = \frac{\mu_f \, \Delta H^{vap}}{\mathrm{Pr}_f^{\,n}} \sqrt{\frac{g\left(\rho_f - \rho\right)}{\sigma}} \left(\frac{C_{p,f} \Delta T_W}{c_{sf} \Delta H^{vap}} \right)^3 \tag{2949}$$

where:

- the viscosity μ_f, the Prandtl number $\mathrm{Pr}_f^{\,n}$, and the specific heat $C_{p,f}$ are evaluated for the liquid phase.
- ρ_f and ρ denote the densities of the liquid and gas phases.
- g and σ denote the gravity acceleration and surface tension coefficient respectively.
- c_{sf} and n characterize the fluid wall interface.

  For example, for water-brass contact, the values are $c_{sf} = 0.006$ and $n = 3$.

- T_{sat} represents the saturation temperature of the mixture and is computed from the temperature-dependent saturation pressure of its components by applying Dalton's law.

The heat flux increases as T_W increases but does not exceed a critical heat flux that is:

$$q_{max} = c_{max}\rho^{1/2}\,\Delta\,H^{vap}\left[g\left(\rho_f - \rho\right)\sigma\right]^{1/4} \tag{2950}$$

where $c_{max} = 0.15$. After q_{max} has been attained, the heat flux decreases to its minimum at the Leidenfrost temperature T_L:

$$q_{min} = c_{min}\rho_g\,\Delta\,H^{vap}\left[g\left(\rho_f - \rho\right)\sigma\right]^{1/4}\left(\rho_f + \rho\right)^{-1/2} \tag{2951}$$

where $c_{min} = 0.09$.

The wall temperature of maximal heat flux $T_S = T_{sat} + \Delta\,T_S$ can be obtained from equations for q_{max} and q_{boil} resulting in

$$\Delta\,T_S = \frac{c_{sf}c_{max}^{1/3}\mathrm{Pr}_f^{\,n/3}\sigma^{1/4}\rho^{1/6}\,\Delta\,H^{vap}}{C_{p,f}\mu_f^{\,1/3}g^{1/12}\left(\rho_f - \rho\right)^{1/12}} \tag{2952}$$

Based on this correlation, the heat flux from the wall is computed as follows:

If $\Delta\,T_W < \Delta\,T_S$, then

$$q_W = q_{boil} \tag{2953}$$

Else, if $\Delta\,T_W < c_s\,\Delta\,T_S$, then

$$q_W = q_{max} \tag{2954}$$

Else, if $\Delta\,T_W < \Delta\,T_L$, then

$$q_W = q_{min} + \frac{\Delta\,T_L - \Delta\,T_W}{\Delta\,T_L - c_s\,\Delta\,T_S}\left(q_{max} - q_{min}\right) \tag{2955}$$

Else,

$$q_W = q_{min}\frac{\Delta\,T_W}{\Delta\,T_L} \tag{2956}$$

$\Delta\,T_L = T_L - T_{sat}$ and c_s is assumed to be 1.2.

The component mass rates of vaporization are then computed as

$$\dot{m}_{v,i} = c_e Y_i \frac{q_W}{\Delta H^{vap}} \tag{2957}$$

where c_e is the wall boiling evaporation factor that determines how much of the boiling heat flux generates vapor. A value smaller than one means that a portion of the boiling heat flux does not generate vapor but heats up the film, for example:

- when heat is supplied near the wall to bubbles which then move away into the liquid film and condensate again inside the film
- when evaporation takes place at the base of a bubble and condensation at its tip.

Melting and Solidification

STAR-CCM+ models melting and solidification of the fluid film.

The melting-solidification model does not track the liquid-solid interface explicitly. Instead, the model uses an enthalpy formulation to determine the distribution of the solidified portion of the liquid-solid film.

The enthalpy of the liquid-solid film ${h_{\mathrm{ls}}}^*$ includes the latent heat of fusion h_{fusion}:

$$ {h_{\mathrm{ls}}}^* = h_{\mathrm{ls}} + \left(1 - {\alpha^*}_s\right)_h \tag{2958} $$

where h_{ls} is the sensible enthalpy. The relative solid volume fraction ${\alpha^*}_s$ is defined as the portion of the volume of the liquid-solid film which the solid state occupies. In the enthalpy model, the relative solid volume fraction ${\alpha^*}_s$ is a function of temperature:

$$ {\alpha^*}_s = \begin{cases} 1 & \text{if} \quad T^* < 0 \\ f(T^*) & \text{if} \quad 0 < T^* < 1 \\ 0 & \text{if} \quad 1 < T^* \end{cases} \tag{2959} $$

where T^* is the normalized temperature that is defined as:

$$ T^* = \frac{T - T_{\mathrm{solidus}}}{T_{\mathrm{liquidus}} - T_{\mathrm{solidus}}} \tag{2960} $$

The function $f(T^*)$ is called the fraction solid curve. For a linear dependence between ${\alpha^*}_s$ and T^*, the solidification path is defined as:

$$ f(T^*) = 1 - T^* \tag{2961} $$

The solidification path can also be specified as a table, in which case linear dependencies are assumed between the points in the table. The values of temperature in the table have to increase (decrease) monotonically as the values of the relative solid fraction decrease (increase) monotonically.

If melting and solidification take place at one temperature (that is, $T_{\mathrm{solidus}} = T_{\mathrm{liquidus}}$), a linear solidification path is assumed and a small temperature interval of 0.002 K is automatically introduced.

Solidified Mass Removal and Morphing

STAR-CCM+ provides an option to remove the solidified mass from the film. To accurately represent the geometry in combination with the solidified layer that forms on it, you use morphing motion to adjust the surface of the solidified volume when the solidified mass is removed. The removed solid mass determines the distance the morpher displaces the boundary to model the thickness of the solidified layer. This option is typically used to model icing.

During every time step, a solid film thickness increment $\Delta\ h_s$ is computed, updated at every inner iteration during the segregated process. This value represents the increase in solidified thickness during the time step. $\Delta\ h_s$ is then used to construct the sources and sinks into the various transport equations.

The updates (at every inner iteration) are computed as:

$$\Delta\ h_s{}' = \begin{cases} -\dfrac{\Delta\ H(T) - \Delta\ H_{liq}}{\Delta\ H_{fus}}\left(h + \Delta\ h_s\right) & \text{if} \quad T \geq T_{liq} \\ h \cdot \alpha s & \text{otherwise} \end{cases} \tag{2962}$$

with $\Delta\ h_s{}'$ the correction to $\Delta\ h_s$. This expression implies that (within each time step) the amount of solidified mass increases when the temperature is below the liquidus temperature, but is reduced if the temperature is above the liquidus temperature.However, the present implementation has a lower limit:

$\Delta\ h_s \geq 0$

and therefore this feature only allows mass to solidify; it does not allow the reverse process of melting. Once the solidified mass has been removed, it leaves the system and cannot be retrieved.

The solid film thickness increment can be related to a mass sink $\frac{kg}{m^2 s}$:

$$\dot{m} = \frac{\rho_{liq}\ \Delta\ h_s}{\Delta\ t} \tag{2963}$$

which is used as a base for corresponding sources in the continuity, momentum, energy, and species transport equations in the film.

For each time step, the solidification is computed as follows:

- A mass sink is applied to the liquid film, corresponding to how much liquid is turning into solid mass.
- This mass sink is used to calculate the velocity of the solid front (that is, the velocity that the interface between solid and liquid is moving), also taking into account the density of the solid mass.
- This velocity is used to calculate the solid thickness increment $\Delta\ h_s$ for that time step; simply $\Delta\ h_s = V \cdot dt$.

Typically, this solidified mass accumulates on the wall: it remains in place (it does not flow any further) and deforms the surface as seen by the gas and the remaining film. This effect is modeled by feeding the displacement $\Delta\ h_s$ into the morpher.

For each time step, the displacement of every face is:

$$\overline{d}_{\text{face}} = \Delta h_s \mathbf{a} \tag{2964}$$

where $\mathbf{a}$ is a unit vector, normal to the film surface and pointing into the gas (that is, the growth is assumed to happen in the direction normal to the surface). The displacement $\overline{d}_{\text{face}}$ is then interpolated to displacements $\overline{d}_{\text{vert}}$ on the vertices.

The solid time-step factor value K acts as a multiplier for the solid front velocity, that is, $V = K \cdot V$. Consequently, within each time step, the solid thickness increment is increased by a factor K. This can be interpreted as a form of solid multiplier (that is, each 1kg/s of solidifying liquid is turned into K kg/s of solid). This can be considered as a mass source, although there are no transport equations solved for the solid: the

solid is removed from the simulation and its effect is included only through the morphing of the boundary. The total physical time should be reduced by a factor K to get the same physical result.

When the vertices are displaced according to the algorithm that is described above, the displacement can lead to roughness in the deformation, forming a zig-zag pattern. This roughness can cause instability issues in the fluid film transport.

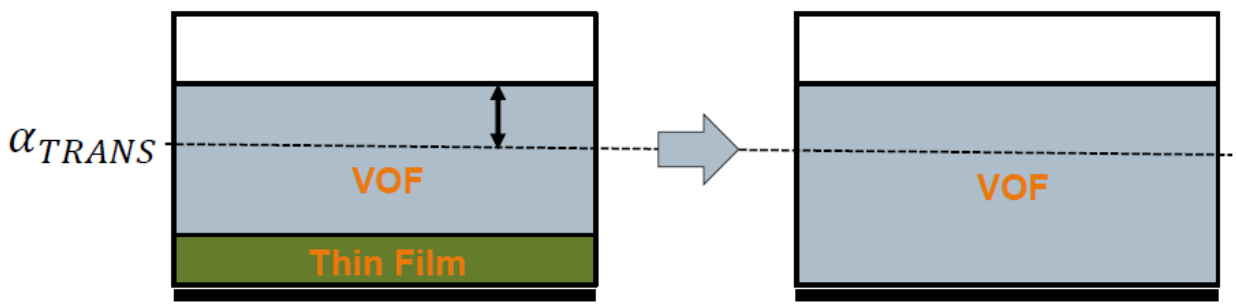
To resolve this issue, an optional smoother is available. The smoother recomputes the displacement as a weighted average, including the displacements at the nearest neighbors. This smoothing can be repeated multiple times.

Resolved Fluid Film

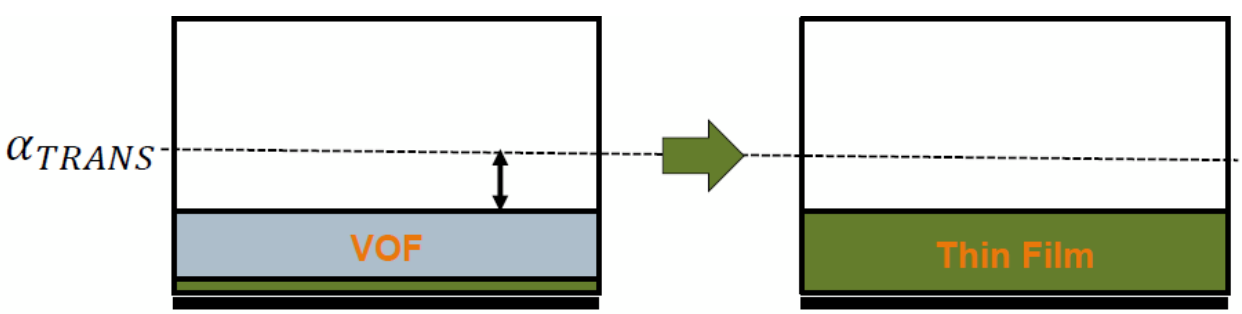
The Resolved Fluid Film model automatically uses the phase representation (fluid film or VOF) that is most suitable for the local flow conditions and mesh resolution.

The transition between the fluid film and VOF phase models is based on the volume fraction of liquid (the total of the fluid film phase and the VOF phase combined) at an interface and the transition volume fraction α_{TRANS} that is specified for the interface:

- Fluid film transitions into the VOF phase when the total volume fraction in the neighbor volume cell is greater than the specified transition volume fraction α_{TRANS}.



- The VOF phase transitions into fluid film when the total volume fraction is less than the specified transition volume fraction α_{TRANS}.



At every iteration, STAR-CCM+ determines the most appropriate physics model by considering the volume fraction:

$$\alpha_h = \frac{V_{VOF} + V_{film}}{V_{cell}} \tag{2965}$$

where:

- V_{VOF} is the volume of the VOF liquid phase in the cell.
- V_{film} is the volume of fluid film in the cell.
- V_{cell} is the volume of the cell.

Values of α_{TRANS} from 0 through 1 give the expected mesh-dependent behavior. If $\alpha_h > \alpha_{TRANS}$, the fluid film is modeled as VOF; otherwise, it is modeled as fluid film. By default, $\alpha_{TRANS} = 0.5$.

If $\alpha_{TRANS} \leq 0$, the liquid is modeled as VOF only. If $\alpha_{TRANS} \geq 1$, the fluid film model is used.

The transfer of mass between the fluid film and the VOF liquid phase is:

$$\dot{m} = \left(\alpha_h - \alpha_{TRANS}\right) \cdot \frac{\widetilde{m}}{\tau} \tag{2966}$$

where:

- $\dot{m}$ is the mass transfer rate in kg/s. This value is positive for fluid film → VOF mass transfer.
- $\widetilde{m}$ is the mass of liquid that is available for transfer.
- τ is the time scale. This value is of the order of the time step size.

Fluid Film Bibliography

[554] *Arai, T., Hashimoto, H., 1985. Disintegration of a thin liquid sheet in a concurrent gas stream, Proceedings of the Third International Conference on Liquid atomization and Spray Systems, London*

[555] *Bai, C. and Gosman, A.D. 1996. “Mathematical modeling of wall films formed by impinging sprays”, SAE Technical Paper Series 960626.*

[556] *Foucart, H. Private Communication.*

[557] *Friedrich, M. A., Lan, H., Drallmeier, J. A., Armaly, B. F. 2007. A separation criterion with experimental validation for shear-driven films in separated flows, Accepted for publication by ASME J. Fluids Engineering.*

[558] *Gubaidullin, A. 2007. Comments on Liquid film atomization on wall edges separation criterion and droplets formation model, J. Fluids Engineering, 129, pp. 665-666.*

[559] *Kistler, S. F. 1993, "Hydrodynamics of wetting", Wettability, pp. 311.*

[560] *Ledoux, M. 1993. Modelisation des Sprays, Cours de DEA.*

[561] *Maroteaux, F., Llory, D., Le Coz, J-F., Habchi, C. 2002. Liquid film atomization on wall edges separation criterion and droplets formation model, J. Fluids Engineering, 124, pp. 565-575.*

[562] *Sirignano, W.A. 1999. Fluid Dynamics and Transport of Droplets and Sprays, Cambridge University Press, New York.*

[563] *Torres, D.J., O'Rourke, P.J., and Amsden, A.A. 2003. "Efficient multicomponent fuel algorithm", Combust. Theory Modelling, 7, p. 67.*

[564] *White, F.M. 1988. "Heat and Mass Transfer". Addison Wesley.*

Lagrangian Multiphase Flow

Multiphase flows are found in a wide variety of industrial processes, some examples of which are internal combustion engines, liquid, or solid fueled combustors, spray driers, cyclone dust separators, and chemical reactors. Multiphase in this context refers to one thermodynamic phase, be it a solid, a liquid, or a gas, interacting with another distinct phase.

In many of these multiphase processes, there are large separations in physical scales between the phases which make a calculation of a full size device impractical or impossible. For example, in a typical heavy duty diesel engine there can be approximately 10^7 distinct particles with a wide range of diameters, temperatures, and compositions, making a full Eulerian description at the very least impractical—the particles have diameters on the order of micrometers and the cylinder bore is in the order of ten centimeters. Processes which have a large number of diameter classes or a broad distribution of states (such as temperature) can often be more easily modeled using a Lagrangian frame of reference. The Lagrange multiphase flow models in STAR-CCM+ are designed to simulate and track the flow path of dispersed particles in a continuous phase including the associated heat and mass transfer phenomena such as droplet evaporation.

Lagrangian numerical methods can be used in conjunction with Eulerian numerical methods to describe these situations where individual particle dynamics impact the solution on the scale that is resolved in the Eulerian field. Typically, a Lagrangian reference frame is used to describe the evolution of individual particles as they traverse the domain. The equations of change are written following an individual particle. Particles are not resolved on the Eulerian field, they are subgrid and the interaction between the phases is modeled.

In general, the continuous phase drives the motion of the dispersed phase. The continuous phase itself is affected by the dispersed phase, as particles occupy volume, and can exchange momentum, heat, and mass with the continuous phase. The strength of the interactions depends on the size, density, and number of dispersed particles.

If the flow is laminar, each particle that is released from a point follows a smooth unique trajectory, that is, the motion of the particle is deterministic. If particles are released to a turbulent carrier flow, each of them has their own random path due to interaction with the fluctuating turbulent velocity field.

If the dispersed phase is volatile, soluble, or reactive, mass transfer occurs between the phases. These effects are accompanied by interphase heat transfer that can also arise due to the interphase temperature differences. Interphase mass transfer causes size changes in the dispersed particles. Even if the initial size distribution is uniform (monodispersed), these effects produce a variable-size (polydispersed) population. The size changes can also be produced by fluid-dynamic forces acting on the dispersed particles causing them to break up into smaller particles. Collisions between particles can also produce the opposite effect, that is, size increase due to coalescence or agglomeration. When dispersed particles hit a wall, the result can be bouncing or shattering, depending on the impact conditions.

Applications with high volume fractions of particles, such as fluidized beds, packed beds, or pneumatic transport, the effect of inter-particle interaction is important. For these applications, STAR-CCM+ offers the discrete element method (DEM) that is incorporated into the Lagrangian multiphase framework. DEM is

specifically designed to model granular material. In general, DEM particles are solid particles that experience additional contact forces, which enter the Lagrangian momentum equations. In addition to the linear momentum equation, the conservation of angular momentum must be fulfilled for a DEM particle.

STAR-CCM+ provides a modeling framework that is able to encompass all of the above phenomena. The framework is based on a Lagrangian-Eulerian approach [576] where the conservation equations of mass, momentum, and energy for the dispersed phase are written for each individual particle in Lagrangian form. This approach allows to calculate the trajectory of each individual particle. The governing equations for the continuous phase are expressed in Eulerian form and they are modified to take into account the presence of the dispersed phase. For flows involving a small number of dispersed particles, it is possible to solve a set of the Lagrangian equations for every particle. However, if the number of particles is large, a statistical approach is more practical. In this statistical approach, the total number of particles is represented by a smaller number of computational parcels. Each of these parcels represents a group (cluster) of particles that share the same properties. The number of particles in a parcel must be large enough so that the properties of the full population of particles are well represented. Also, the number of parcels that are used in the simulation must be large enough to represent the properties of the full population of particles.

STAR-CCM+ distinguishes between different types of particles: material particles, massless particles, multi-component particles, and DEM particles. The material particles are the most general Lagrangian multiphase dispersed phase that have mass and volume and are governed by the Lagrangian conservation laws of mass, momentum, and energy. Massless particles, on the other hand, do not represent a material, but rather massless virtual particles. These virtual particles move with the continuous phase without influencing each other or the continuous phase. Massless particles are useful to visualize turbulent diffusion of the continuous phase, the residence time of fluid in the continuous phase, or the mixing of fluid streams in the continuous phase. Multi-component particles are liquid particles (droplets) that are composed of two or more liquid materials that can evaporate to their respective gaseous counterparts. For multi-component material particles, the mass conservation equation is solved for each component of the droplet. The overall mass conservation is satisfied by summing up the component mass conservation equations. DEM particles are available with DEM only and they are restricted to solid particles. DEM particles can be single or multi-component solid or multi-component coal. The shapes of the DEM particles can be defined in several ways:

- single spherical particles
- larger sperical particles that represent multiple smaller particles
- non-spherical particles with fixed bonds between the particle components
- non-spherical particles with breakable bonds between the particle components

Contents:

Particle Equations of Motion
Particle Heat and Mass Transfer
Two-Way Coupling with the Continuous Phase
Particle Injection
Sprays and Droplets
Particle Porosity
Discrete Element Method
Erosion
Passive Scalar Interactions
Solution Methodology

Lagrangian Multiphase Bibliography

Particle Equations of Motion

The conservation equation of momentum for a particle is written in the Lagrangian framework. The change in momentum is balanced by surface and body forces that act on the particle.

The equation of conservation of (linear) momentum for a material or DEM particle of mass m_p is given by:

$$m_p \frac{d\mathbf{v}_p}{dt} = \mathbf{F}_s + \mathbf{F}_b \tag{2967}$$

where $\mathbf{v}_p$ denotes the instantaneous particle velocity, $\mathbf{F}_s$ is the resultant of the forces that act on the surface of the particle, and $\mathbf{F}_b$ is the resultant of the body forces. These forces in turn are decomposed into:

$$\mathbf{F}_s = \mathbf{F}_d + \mathbf{F}_p + \mathbf{F}_{vm} \tag{2968}$$

and

$$\mathbf{F}_b = \mathbf{F}_g + \mathbf{F}_{MRF} + \mathbf{F}_u + \mathbf{F}_c + \mathbf{F}_{\mathrm{Co}} \tag{2969}$$

where:

$\mathbf{F}_d$	drag force
$\mathbf{F}_p$	pressure gradient force
$\mathbf{F}_{vm}$	virtual mass force
$\mathbf{F}_g$	gravity force
$\mathbf{F}_u$	user-defined body force
$\mathbf{F}_c$	contact force (DEM modeling only)
$\mathbf{F}_{\mathrm{Co}}$	Coulomb force

The resultant of the surface forces $\mathbf{F}_s$ represents the momentum transfer from the continuous phase to the particle. When using the two-way coupling modeling approach (see Interaction with the Continuous Phase), $\mathbf{F}_s$ is accumulated over all the parcels and applied in the continuous phase momentum equation.

In DEM modeling, the contact force $\mathbf{F}_c$ represents inter-particle and particle-boundary interaction. This force is accumulated over the contacts that a particle has and it is applied as a body force:

$$\mathbf{F}_c = \sum_{contacts} \mathbf{F}_{cm} \tag{2970}$$

where $\mathbf{F}_{cm}$ is a contact model force. STAR-CCM+ provides several contact force models that describe the contact force $\mathbf{F}_{cm}$. For more information, see Contact Force.

Additionally, DEM particles have orientations and therefore, their angular momentum must also be conserved:

$$\mathbf{I}_p \frac{d\boldsymbol{\omega}_p}{dt} = \mathbf{M}_b + \mathbf{M}_c \tag{2971}$$

where:

$\mathbf{I}_p$	particle moment of inertia, which is described by a second-order tensor
$\boldsymbol{\omega}_p$	particle angular velocity
$\mathbf{M}_b$	drag torque, that is, moment that acts on the particle due to rotational drag(Drag Torque)
$\mathbf{M}_c$	total moment from contact forces, both from rolling resistance models ($\mathbf{M}_{cm}$) and any other contact forces ($\mathbf{r}_c \times \mathbf{F}_{cm}$)

$$\mathbf{M}_c = \sum_{contacts} \left(\mathbf{r}_c \times \mathbf{F}_{cm} + \mathbf{M}_{cm}\right) \tag{2972}$$

where:

$\mathbf{r}_c$	position vector from the particle center of gravity to the contact point
$\mathbf{M}_{cm}$	moment that acts on the particle from rolling resistance (Rolling Resistance)

Drag Force

The drag force in Eqn. (2968) is defined as:

$$\mathbf{F}_d = \frac{1}{2} C_d \rho A_p \left|\mathbf{v}_s\right| \mathbf{v}_s \tag{2973}$$

where:

C_d	drag coefficient of the particle
ρ	density of the continuous phase
$\mathbf{v}_s = \mathbf{v} - \mathbf{v}_p$	the particle slip velocity with $\mathbf{v}$ being the instantaneous velocity of the continuous phase
A_p	projected area of the particle

Eqn. (2973) can also be written as:

$$\mathbf{F}_d = \frac{m_p \mathbf{v}_s}{\tau_v} \tag{2974}$$

where τ_v is the momentum relaxation time-scale:

$$\tau_v = \frac{2m_p}{C_d \rho A_p |\mathbf{v}_s|} \tag{2975}$$

Drag Coefficient

The drag coefficient C_d in Eqn. (2973) is a function of the small-scale flow features around the individual particles.

These features are impractical to resolve spatially, and so the usual practice is to obtain the drag coefficient from correlations, typically derived from experiment or theoretical studies. These correlations depend on the nature of the dispersed phase. Droplets, bubbles, and solid particles have different correlations. The shape of the particle, the presence of inter-phase mass and energy transfer, and so on, affect these correlations.

STAR-CCM+ provides the following methods for defining the drag coefficient:

- Schiller-Naumann correlation
- Liu dynamic drag coefficient
- Haider and Levenspiel

The following drag force coefficients are available with the DEM drag force model:

- Clift
- Di Felice
- Gidaspow

Schiller-Naumann Correlation

The Schiller-Naumann correlation [604] is suitable for spherical solid particles, liquid droplets, and small-diameter bubbles. It is formulated as:

$$C_d = \begin{cases} \frac{24}{\mathrm{Re}_p}\left(1 + 0.15\mathrm{Re}_p^{0.687}\right) & \mathrm{Re}_p \leq 10^3 \\ 0.44 & \mathrm{Re}_p > 10^3 \end{cases} \tag{2976}$$

where Re_p is the particle Reynolds number that is defined as:

$$\mathrm{Re}_p \equiv \frac{\rho |\mathbf{v}_s| D_p}{\mu} \tag{2977}$$

where D_p is the particle diameter and μ is the dynamic viscosity. This correlation is available only when the continuous phase is viscous.

Liu Dynamic Drag Coefficient

The Liu dynamic drag coefficient [585] accounts for the dependence of the drag of a liquid droplet on its distortion under the action of aerodynamic forces. As a basis, it uses the following expression for the drag coefficient of an undistorted sphere:

$$C_{d,sphere} = \begin{cases} \frac{24}{\mathrm{Re}_p}\left(1 + \frac{1}{6}\mathrm{Re}_p^{2/3}\right) & \mathrm{Re}_p \leq 1000 \\ 0.424 & \mathrm{Re}_p > 1000 \end{cases} \tag{2978}$$

As the distortion of the droplet increases, its shape is assumed to become a disk whose axis is aligned with the slip velocity. This shape increases the drag on the droplet. The Liu drag coefficient models this effect by noting that the high Reynolds number limit of the drag coefficient of a disk is 1.54. It then assumes that the disk drag is 1.54/0.424 higher than the sphere drag at all Reynolds numbers, and that the drag of intermediate shapes can be interpolated between those two extremes. So:

$$C_d = C_{d,sphere}\left(1 + 2.632y\right) \tag{2979}$$

The interpolation factor y is 0 for a sphere and 1 for a disk. It is identified as the TAB distortion, Eqn. (3101), and hence the Liu drag coefficient requires the TAB distortion model to calculate this quantity.

Clift Method

This method is available only with cylindrical particles. The drag coefficient C_d depends on the particle Reynolds number and the aspect ratio of the cylinder. The Clift drag coefficient is given as:

- For long cylindrical particles (aspect ratio ≥ 1):

$$\begin{aligned}
\mathrm{Re}_p \le 5&: \quad C_d = 9.689\ \mathrm{Re}_p{}^{-0.78}\left(1 + 0.147\ \mathrm{Re}_p{}^{0.82}\right)\\
5 < \mathrm{Re}_p \le 40&: C_d = 9.689\ \mathrm{Re}_p{}^{-0.78}\left(1 + 0.227\ \mathrm{Re}_p{}^{0.55}\right)\\
40 < \mathrm{Re}_p \le 400&: \ C_d = 9.689\ \mathrm{Re}_p{}^{-0.78}\left(1 + 0.0.838\ \mathrm{Re}_p{}^{0.82}\right)\\
400 < \mathrm{Re}_p&: \quad C_d = 0.99\ \gamma^{-.012}E^{-0.08}
\end{aligned} \tag{2980}$$

  where $\gamma = \rho_p/\rho_f$ is the ratio of particle density to fluid density and $E = h/d$ is the aspect ratio.

- For flattened cylindrical particles (aspect ratio < 1):

$$\begin{aligned}
\mathrm{Re}_p \le 0.01&: \ C_d = \frac{64}{\pi\ \mathrm{Re}_p}\left(1 + \frac{\mathrm{Re}_p}{2\pi}\right)\\
0.01 < \mathrm{Re}_p \le 1.5&: \ C_d = \frac{64}{\pi\ \mathrm{Re}_p}\left(1 + 10^x\right)\\
\left(x = -0.883 + 0.906\ \log\left(\mathrm{Re}_p\right) - 0.025\left(\log\mathrm{Re}_p\right)^2\right)&\\
1.5 < \mathrm{Re}_p \le 133&: \ C_d = \frac{64}{\pi\ \mathrm{Re}_p}\left(1 + 0.138\ \mathrm{Re}_p^{0.792}\right)\\
133 < \mathrm{Re}_p&: \ C_d = 1.17
\end{aligned} \tag{2981}$$

Di Felice Drag Coefficient Method

This drag method introduces an extra term in the fluid drag force expression (Eqn. (2973)) to account for the presence of other particles around a particle. The Di Felice drag coefficient is given as:

$$C_d = \left(0.63 + \frac{4.80}{\sqrt{\epsilon_i \mathrm{Re}_p}}\right)^2 \epsilon^{2-\zeta} \tag{2982}$$

where Re_p defines the particle Reynolds Number, ϵ_i is the void fraction around a particle, and:

$$\zeta = 3.7 - 0.65\exp\left[-0.5\left(1.5 - \log\left[\epsilon\ \mathrm{Re}_p\right]\right)^2\right] \tag{2983}$$

The term $\epsilon^{2-\zeta}$ accounts for the effect of enhanced drag on a particle, due to the presence of other particles around it.

Gidaspow Drag Coefficient Method

The Gidaspow model is useful in systems with regions of high particle density, for example, in fluidized beds. The model covers a wide range of void fractions by combining the Wen-Yu and the Ergun equations [623].

STAR-CCM+ incorporates the equations in drag coefficient form as follows:

$$C_d = \frac{4}{3}\left(150\frac{1-\epsilon}{\epsilon\ \mathrm{Re}_p} + 1.75\right) \quad \text{if } \epsilon < \epsilon_{min} \tag{2984}$$

Otherwise:

$$C_d = \begin{cases} \dfrac{\left(14 + 3.6 \cdot\ \mathrm{Re}_p^{0.687}\right)}{\epsilon\,\mathrm{Re}_p}\epsilon^{1-w} & \text{if } \ \epsilon\,\mathrm{Re}_p \le 1000 \\ 0.44 & \text{if } \ \epsilon\,\mathrm{Re}_p > 1000 \end{cases} \tag{2985}$$

where:

- ϵ is the void fraction.
- ϵ_{min} is the cutoff void fraction.
- w is the Wen-Yu exponent.

Haider and Levenspiel Drag Coefficient Method

The drag coefficient C_d depends on the particle Reynolds number and the particle sphericity:

$$C_d = \frac{24}{\mathrm{Re}_p}\left(1 + A\mathrm{Re}_p^B\right) + \frac{C}{\left(1 + \frac{D}{\mathrm{Re}_p}\right)} \tag{2986}$$

where ϕ is the particle sphericity and:

$$\begin{aligned} A &= 8.1716e^{-4.0665\phi} \\ B &= 0.0964 + 0.5565\phi \\ C &= 73.690e^{-5.0746\phi} \\ D &= 5.3780e^{6.2122\phi} \end{aligned} \tag{2987}$$

Pressure Gradient Force

The pressure gradient force is as defined as:

$$\mathbf{F}_p = -V_p \nabla p_{static} \tag{2988}$$

where V_p is the volume of the particle and ∇p_{static} is the gradient of the static pressure in the continuous phase.

Virtual Mass Force

The virtual mass force [569] is:

$$\mathbf{F}_{vm} = C_{vm}\rho V_p\left(\frac{D\mathbf{v}}{Dt} - \frac{d\mathbf{v}_p}{dt}\right) \tag{2989}$$

where C_{vm} is the virtual mass coefficient. The default value of 0.5 for this coefficient is for a sphere in a uniform, inviscid, incompressible flow [587]. The operator D/Dt is the material (substantive) derivative.

Drag Torque

For DEM particles, the drag torque $\mathbf{M}_b$ (see Eqn. (2971)) reduces the difference in the rotational velocities between a particle and the fluid in which it is immersed.

The drag torque $\mathbf{M}_b$ applied to a DEM particle is defined as [119]:

$$\mathbf{M}_b = \frac{\rho}{2}\left(\frac{D}{2}\right)^5 C_R|\mathbf{\Omega}|\mathbf{\Omega} \tag{2990}$$

where:

C_R	rotational drag coefficient
$\mathbf{\Omega}$	relative angular velocity of the particle to the fluid

The relative angular velocity of the particle to the fluid, or slip-rotation, is given by:

$$\mathbf{\Omega} = \frac{1}{2}\nabla \times \mathbf{v} - \boldsymbol{\omega}_p \tag{2991}$$

where:

$\mathbf{v}$	fluid velocity
$\boldsymbol{\omega}_p$	angular velocity of the particle

The rotational drag coefficient C_R in Eqn. (2990) is defined as [119]:

$$C_R = \begin{cases} \dfrac{12.9}{\mathrm{Re}_R{}^{0.5}} + \dfrac{128.4}{\mathrm{Re}_R}, & 32 \le \mathrm{Re}_R < 1000 \\ \dfrac{64\pi}{\mathrm{Re}_R}, & \mathrm{Re}_R < 32 \end{cases} \tag{2992}$$

Re_R is the rotational Reynolds number, defined as:

$$\mathrm{Re}_R = \frac{\rho D^2|\mathbf{\Omega}|}{\mu} \tag{2993}$$

Although Sommerfeld restricts the rotational Reynolds number to values below 1000, the STAR-CCM+ places no upper limit on it.

Lift Forces

Lift forces in DEM simulations can arise from particle spin, particle shear, or both. In this context, lift forces refer to mean forces normal to the particle velocity; they are not necessarily forces in the upward direction. Lift forces in statistical Lagrangian simulations can arise from particle shear only.

Particle Spin Lift Force

This force applies to a spinning particle moving relative to a fluid. The force is given by:

$$\mathbf{F}_{LR} = \frac{\rho\pi}{8} D^2 C_{LR} |\mathbf{v}_s| \frac{\mathbf{\Omega} \times \mathbf{v}_s}{|\mathbf{\Omega}|} \tag{2994}$$

Sommerfeld [119] gives the coefficient of rotational lift C_{LR} as:

$$C_{LR} = 0.45 + \left(\frac{\mathrm{Re}_R}{\mathrm{Re}_p} - 0.45\right) e^{-0.5684 \mathrm{Re}_R{}^{0.4} \mathrm{Re}_p{}^{0.3}} \tag{2995}$$

Sommerfeld reports this formula for C_{LR} as valid for particle Reynolds numbers below 140. You can also define C_{LR} using a field function.

Particle Shear Lift Force

This force applies to a particle moving relative to a fluid where there is a velocity gradient in the fluid orthogonal to the relative motion.

Saffman [633] gives the shear lift force as:

$$\mathbf{F}_{LS} = 1.615 D^2 (\rho\mu)^{0.5} \left| \frac{\delta \mathbf{v}}{\delta y} \right| \mathbf{v}_S \tag{2996}$$

The y direction is the direction of the velocity gradient.

The three-dimensional version of this equation is:

$$\mathbf{F}_{LS} = C_{LS} \frac{\rho\pi}{8} D^3 (\mathbf{v}_S \times \boldsymbol{\omega}) \tag{2997}$$

where:

- $\boldsymbol{\omega} = \nabla \times \mathbf{v}$, the curl of the fluid velocity.
- C_{LS} is the shear lift coefficient.

You can define C_{LS} as a field function or use either of two published definitions:

- Saffman [633] provides a definition that recovers the original Saffman asymptotic solution for low Reynolds numbers:

$$C_{LS} = \frac{4.1126}{\mathrm{Re}_S{}^{0.5}} \tag{2998}$$

- Sommerfeld [119] provides a definition for a broader range of Reynolds numbers:

$$C_{LS} = \frac{4.1126}{\mathrm{Re}_S{}^{0.5}} f\mathrm{Re}_p, \mathrm{Re}_S \tag{2999}$$

where:

$$f\mathrm{Re}_p, \mathrm{Re}_S = \begin{cases} \left(1 - 0.3314\beta^{0.5}\right) e^{-0.1\mathrm{Re}_p} + 0.3314\beta^{0.5}, & \left(\mathrm{Re}_p \le 40\right) \\ 0.0524\left(\beta \mathrm{Re}_p\right)^{0.5}, & \left(\mathrm{Re}_p > 40\right) \end{cases} \tag{3000}$$

and:

$$\beta = 0.5 \frac{\mathrm{Re}_S}{\mathrm{Re}_p} \tag{3001}$$

where:

Re_S is the Reynolds number for shear flow:

$$\mathrm{Re}_S = \frac{\rho D^2 |\boldsymbol{\omega}|}{\mu}$$

Gravity Force

The gravity force is defined as:

$$\mathbf{F}_g = m_p \mathbf{g} \tag{3002}$$

where $\mathbf{g}$ is the gravitational acceleration vector.

Forces in Moving Reference Frames

A rotating reference frame introduces a body force:

$$F_{MRF} = m_p \left[\boldsymbol{\omega} \times \left(\boldsymbol{\omega} \times \mathbf{r} \right) + 2 \left(\boldsymbol{\omega} \times \mathbf{v}_p \right) \right] \tag{3003}$$

where $\boldsymbol{\omega}$ is the angular velocity vector of the rotating reference frame and $\mathbf{r}$ is the distance vector to the axis of rotation.

Coulomb Force

Particles can enter the domain bearing electrical charges or, under some circumstances, can acquire electric charges after entering, resulting in an additional force acting on the particles. Charged particles appear in many applications, such as electrostatic spray painting and electrostatic precipitation in air pollution control.

A charged particle with a charge q in an electric field $\mathbf{E}$ experiences a force. This force is called Coulomb force and is given by:

$$\mathbf{F}_{\mathrm{Co}} = q\,\mathbf{E} \tag{3004}$$

Field Charging

Field Charging models the charging of particles by an electric field that comes from a unipolar charge distribution in the carrier phase. It is the dominant mechanism of particle charging for particles larger than 1 micron.

Field charging occurs when unipolar ions, traveling along an electric field, bombard a particle and lose their charge to it, until the charge on the particle reaches a saturation level.

The saturation charge of a spherical particle exposed to an electric field is:

$$Q_s = \frac{3\pi\epsilon_c\epsilon_r}{\epsilon_r + 2} |E| d_p^2 \tag{3005}$$

where:

- ϵ_c is the permittivity of the medium.
- ϵ_r is the relative permittivity of the particle.
- E is the strength of the electric field, calculated from Maxwell's equations.
- d_p is the diameter of the particle.

The total surface charge on the particle Q_p evolves as:

$$\frac{dQ_p}{dt} = \frac{Q_s}{\tau_q}\left(1 - \frac{Q_p}{Q_s}\right)^2 \tag{3006}$$

where the time constant τ_q is:

$$\tau_q = \frac{4\epsilon_c}{\rho K} \tag{3007}$$

- ρ is the charge density.
- K is the ion mobility.

User-Defined Body Force

The user-defined body force in Eqn. (2969) is defined as:

$$\mathbf{F}_u = V_p \mathbf{f}_u \tag{3008}$$

where $\mathbf{f}_u$ is a user-specified body force per unit volume.

Particle Position and Velocity

The variables describing the state of the particle change as a function of time and space due to the forces enumerated in the previous sections. In the Lagrangian framework, the state is evolved on a particle by particle basis while the Eulerian fields are frozen. The particle position and particle velocity are obtained by numerically integrating over time the particle momentum equation Eqn. (2967) and the position equation:

$$\frac{dr_p}{dt} = \mathbf{v}_p \tag{3009}$$

where $r_p(t)$ is the instantaneous position vector. A particle trajectory is the locus of points obtained by the integration of the position equation.

In the statistical Lagrangian approach, parcels of individual particles are tracked. The term parcel is used to denote a set of particles which all have the same state and evolve in the same way. A single integration of the state is applied to all the particles in the parcel, and any exchange with the Eulerian phase is multiplied by the number of particles that the parcel represents.

In subsequent sections, a quantity which is identical for a parcel and its particles, such as velocity in the preceding paragraph, is described as a “particle” quantity and given the subscript p. The term “parcel” and the subscript π are reserved for quantities which are exclusively defined on parcels.

Contents:

Turbulent Dispersion

Turbulent Dispersion

A particle in turbulent flow experiences a randomly-varying velocity field to which it responds according to its inertia. This behaviour is modeled by a stochastic approach that includes the effect of instantaneous velocity fluctuations on the particle.

Following Gosman and Ioannides [579], a particle is assumed to pass through a sequence of turbulent eddies as it traverses a turbulent flow field. Here, an eddy is a local disturbance to the Reynolds-averaged velocity field. The particle remains in the eddy until either the eddy time-scale τ_e is exceeded, or the separation between the particle and the eddy exceeds the length scale of the eddy, l_e. In practice, an eddy transit time, τ_c serves in place of the latter.

The particle experiences an instantaneous fluid velocity $\mathbf{v}$ in each eddy, which is

$$\mathbf{v} = \overline{\mathbf{v}} + \mathbf{v}' \tag{3010}$$

where $\overline{\mathbf{v}}$ is the local Reynolds-averaged velocity and $\mathbf{v}'$ is the eddy velocity fluctuation, unique to each particle. The latter is a normal (Gaussian) deviate with zero mean value and a standard deviation that comes from the eddy velocity scale

$$u_e = \frac{l_t}{\tau_t}\sqrt{\frac{2}{3}} \tag{3011}$$

The turbulence model provides the length and time-scales of the turbulence, l_t and τ_t. For example, their ratio is $\sqrt{k}$ with a K-Epsilon or K-Omega turbulence model.

Once generated, a single realization of $\mathbf{v}'$ continues to apply to a particle until its eddy interaction time is exceeded:

$$\tau_I = \min\left(\tau_e, \tau_c\right) \tag{3012}$$

Thus, $\mathbf{v}$ as given by Eqn. (3010) is used to evaluate the particle slip velocity and feeds through to other models, for example, using the particle Reynolds number, Eqn. (2977).

The eddy time-scale measures the lifetime of the eddy: the maximum interval over which a single realization of $\mathbf{v}'$ remains valid. It can be related to the diffusion of a passive scalar according to the underlying turbulence model, giving

$$\tau_e = \frac{2\mu_t}{\rho u_e^2} \tag{3013}$$

Particles can interact with the eddy for less than the eddy time-scale when they have non-zero slip velocity, which precludes massless particles by definition. With non-zero slip velocity, it is possible for the particle to “cross” the eddy, and hence escape it. The eddy transit time is defined only for material particles when the drag force is activated, in which case it is estimated from

$$\tau_c = \begin{cases} \infty & \tau_v \le \dfrac{l_e}{|\mathbf{v}_s|} \\ -\tau_v \ln\left(1 - \dfrac{l_e}{\tau_v |\mathbf{v}_s|}\right) & \tau_v > \dfrac{l_e}{|\mathbf{v}_s|} \end{cases} \tag{3014}$$

where τ_v is the momentum relaxation time-scale, Eqn. (2975), and l_e is the eddy length scale $u_e \tau_e$.

Particle Heat and Mass Transfer

If the dispersed phase is volatile, soluble, or reactive, mass transfer occurs between the phases. This mass transfer is accompanied by interphase heat transfer. Heat transfer can also take place because of the interphase temperature differences. Interphase mass transfer causes size changes in the dispersed material particles.

Particle Mass Balance

The equation of conservation of mass of a material particle is given by:

$$\frac{dm_p}{dt} = \dot{m}_p \tag{3015}$$

where m_p is the mass of the particle and $\dot{m}_p$ the rate of mass transfer to the particle. The latter is zero unless mass transfer occurs, such as evaporation.

If the two-way coupling is active, $\dot{m}_p$ is accumulated over all the parcels and applied in the continuous phase continuity equation. See Two-Way Coupling with the Continuous Phase.

Droplet Evaporation

Quasi-steady single-component droplet evaporation assumes droplets to be internally homogeneous, consisting of a single liquid component.

The rate of change of droplet mass due to quasi-steady evaporation $\dot{m}_p$ can be written as [610]:

$$\dot{m}_p = -g^* A_s \ln\left(1 + B\right) \tag{3016}$$

where B is the Spalding transfer number and g^* is the mass transfer conductance (to be precise, in the limit $B \rightarrow 0$).

Multi-component droplet evaporation also assumes droplets to be internally homogeneous, consisting of an ideal mixture of liquid components, some of which are transferred. This means that they can vaporize. There can also be inert components in both the droplet and the gas.

For multi-component droplet evaporation, the rate of change of mass of each transferred component due to quasi-steady evaporation $\dot{m}_{pi}$ can be written as:

$$\dot{m}_{pi} = -\varepsilon_i g^* A_s \ln\left(1 + B\right) \tag{3017}$$

where the index i refers to each component of the mixture of components and ε_i is the fractional mass transfer rate. The fractional mass transfer rate has the property:

$$\sum_T \varepsilon_i = 1 \tag{3018}$$

with the summation over the T transferred components. Effectively, the transfer number represents the driving force for evaporation, which is a function of thermodynamic conditions of the liquid and vapor. Conductance, on the other hand, represents geometrical and mechanical effects, such as the size and velocity of the droplet. Expressions for these quantities depend on the mode of evaporation, of which there are three:

- Super-Critical
- Heat Transfer Limited Evaporation
- Vapor Diffusion Limited Evaporation

The conductance is positive by definition. The transfer number can be positive or negative, with the latter implying condensation, that is, condensation is treated as “reverse evaporation”.

Evaporation Mode	Single-Component Droplets	Multi-Component Droplets
Super-Critical This mode is active when the droplet temperature exceeds the critical temperature.	$T_p \geq T_c$ where T_p is the particle temperature and T_c is the critical temperature of the material.	$T_p \geq \max\left(T_{c,i}\right)$ where $T_{c,i}$ is the critical temperature of the transferred components.
Heat Transfer Limited Evaporation This mode is active when the vapor at the droplet surface is saturated and subcritical. The evaporation rate results from the balance between heat transfer from the vapor and latent heat transfer due to evaporation.	The condition for a saturated vapor is that the surface equilibrium mole fraction of the vapor exceeds unity: $X_{v,s} = \frac{p_{sat}\left(T_p\right)}{p} \geq 1$ in which $p_{sat}\left(T_p\right)$ is the vapor saturation pressure at the droplet surface temperature. Under this condition, the transfer number is $B = \frac{c_p\left(T - T_p\right)}{L}$ where L is the latent heat of vaporization.	The first condition for a saturated vapor is: $\sum_T X_{is} \geq 1$ (3019) where X_{is} is the equilibrium mole fraction of transferred component i at the droplet surface. X_{is} is evaluated using Raoult’s law: $X_{is} = \frac{p_{sat}^i\left(T_p\right)}{p} X_{ip}$ in which $p_{sat}^i\left(T_p\right)$ is the component saturation pressure that is evaluated at the droplet surface temperature, and X_{ip} is the component mole fraction in the droplet. When the condition in Eqn. (3019) is satisfied, the volatile components vaporize in proportion to their mass fractions, that is: $\varepsilon_i = \frac{Y_{ip}}{\sum_T Y_{ip}}$ The second condition for a saturated vapor is: $\sum_T Y_i = 1$ where Y_i is the mass fraction of component i in the “free stream”, that is the cell containing the droplet.

Evaporation Mode	Single-Component Droplets	Multi-Component Droplets
		The transfer number is $$B = \frac{c_p\left(T - T_p\right)}{\sum_T \varepsilon_i L_i}$$ where L_i is the component latent heat of vaporization.
	Finally, the conductance is $$g^* = \frac{k\mathrm{Nu}_p}{c_p D_p}$$ where Nu_p is the Nusselt number.	
Vapor Diffusion Limited Evaporation This mode is active when the vapor at the droplet surface is subcritical and not saturated, that is, when neither of the other two modes is active. The evaporation rate then depends on the rate at which vapor can diffuse away from the droplet.	Under this condition, the transfer number is [609] $$B = \frac{Y_{v,s} - Y_v}{1 - Y_{v,s}}$$ (3020) where Y_v is the vapor mass fraction in the “free stream”, that is, cell, and $Y_{v,s}$ is the surface equilibrium vapor mass fraction $$Y_{v,s} = X_{v,s}\frac{W_v}{W_s}$$ (3021) The molecular weights W_v and W_s are of the vapor and gas mixture at the droplet surface respectively.	From [609], the fractional mass transfer rate is: $$\varepsilon_i = \frac{Y_{is}(1 + B) - Y_i}{B}$$ The transfer number is: $$B = \frac{\sum_T Y_{is} - \sum_T Y_i}{1 - \sum_T Y_{is}}$$ where Y_i is the mass fraction of component i in the “free stream”, that is the cell containing the droplet. Y_{is} is its surface equilibrium mass fraction: $$Y_{is} = X_{is}\frac{W_i}{W_s}$$
	Finally, the conductance is: $$g^* = \frac{\rho D_v Sh_p}{D_p}$$ (3022) where D_v is the molecular diffusivity of the vapor and Sh_p the Sherwood number.	

Sherwood Number

The Sherwood number is effectively a dimensionless mass transfer conductance.

The Sherwood number is related to the actual mass transfer conductance through Eqn. (3022). As with the drag coefficient and heat transfer coefficient, the Sherwood number must be defined using a correlation.

STAR-CCM+ provides the Ranz-Marshall correlation for defining the Sherwood number.

Ranz-Marshall Correlation

The Ranz-Marshall correlation [596] is suitable for spherical particles up to $\mathrm{Re}_p \simeq 5000$. It is formulated as

$$Sh_p = 2\left(1 + 0.3\mathrm{Re}_p^{1/2} Sc^{1/3}\right) \tag{3023}$$

in which Sc is the Schmidt number of the continuous phase. This correlation is available only when the continuous phase is viscous.

Particle Energy Balance

A material particle is assumed to be internally homogeneous which, from a thermal point of view, implies a low Biot number, for example, less than approximately 0.1.

The generic form of the equation of conservation of energy consistent with this assumption is

$$m_p c_p \frac{dT_p}{dt} = Q_t + Q_{rad} + Q_s \tag{3024}$$

Here, Q_t represents the rate of convective heat transfer to the particle from the continuous phase, Q_{rad} represents the rate of radiative heat transfer, and Q_s other heat sources.

Convective Heat Transfer

The convective heat transfer term is formulated as

$$Q_t = fhA_s\left(T - T_p\right) \tag{3025}$$

where A_s is the particle surface area, and h is the heat transfer coefficient. The factor f is a mass transfer correction, for which the formulation of El Wakil and others [578] is used

$$f = \frac{z}{e^z - 1} \tag{3026}$$

with

$$z = \frac{-\dot{m}_p c_p}{hA_s} \tag{3027}$$

In the limit $\dot{m}_p \to 0$, $f \to 1$.

The heat transfer to the particle is always active in the energy equation. It can also be written

$$Q_t = \frac{f m_p c_p \left(T - T_p\right)}{\tau_T} \tag{3028}$$

where τ_T is the thermal relaxation time-scale:

$$\tau_T = \frac{m_p c_p}{hA_s} \tag{3029}$$

If the Two-Way Coupling model is active, Q_t is accumulated over all the parcels and applied in the continuous phase energy equation.

Heat Transfer Coefficient

As with the drag coefficient, the heat transfer coefficient h must be defined using a correlation.

The heat transfer coefficient is often given in terms of the particle Nusselt number:

$$\mathrm{Nu}_p \equiv \frac{hD_p}{k} \tag{3030}$$

where k is the thermal conductivity of the continuous phase.

STAR-CCM+ provides the Ranz-Marshall correlation for defining the heat transfer coefficient.

Ranz-Marshall Correlation

The Ranz-Marshall correlation [596] is suitable for spherical particles up to $\mathrm{Re}_p \simeq 5000$. It is formulated as:

$$\mathrm{Nu}_p = 2\left(1 + 0.3\mathrm{Re}_p^{1/2}\mathrm{Pr}^{1/3}\right) \tag{3031}$$

in which Pr is the Prandtl number of the continuous phase. This correlation is available only when the continuous phase is viscous.

Radiative Heat Transfer

If the Particle Radiation model is active, heat transfer due to radiation is included in the energy equation. The radiative heat transfer, Q_{rad}, to the particle is defined as:

$$Q_{rad} = \frac{A_s}{4}Q_{a,p}\left(G - 4\sigma T_p^4\right) \tag{3032}$$

where A_s is the particle surface area, $Q_{a,p}$ is the absorption efficiency of the particle, G is the incident radiative heat flux, and σ is the Stefan-Boltzmann constant.

Other Sources of Heat Transfer

The other terms in the energy equation are the user-defined energy source and the latent heat transfer:

$$Q_s = V_p q_u + \dot{m}_p L_{eff} \tag{3033}$$

where q_u is the user energy source (per unit volume) and L_{eff} is the effective latent heat of transferred material, which depends on the mass transfer model and the composition of the transferred material. The user energy source is activated in the energy equation by the User-Defined Energy Source model.

Two-Way Coupling with the Continuous Phase

In STAR-CCM+, you can simulate the interactions between the dispersed phase and the continuous phase as *one-way coupling* or *two-way coupling*. The coupling refers to the way that momentum, heat, and mass are exchanged between the phases.

With one-way coupling, only the continuous phase influences the dispersed phase, but not in the reverse direction. With two-way coupling, the effects of the dispersed phase on the continuous phase such as displacement, interphase momentum, mass, and heat transfer are taken into account.

The displacement of the continuous phase by the dispersed phase is accounted for through the volume fraction. The volume fraction of a Lagrangian phase is the fraction of the local cell volume which that phase occupies. It is calculated for both discrete material particles and DEM particles.

The volume fraction ϕ_c of all Lagrangian phases for which the two-way coupling is active is accumulated. This volume is seen as a void by the continuous phase: the volume available for the continuous phase decreases by this fraction. In practice, ϕ_c is under-relaxed to promote stability:

$$\phi_c = \alpha(\sum\phi) + (1-\alpha)\phi_{c,\ old} \tag{3034}$$

where α is an under-relaxation factor of the two-way coupling solver and $\sum\phi$ is the sum of the volume fractions of the relevant Lagrangian phases. The value of ϕ_c which is contributed to the continuous phase void fraction is limited to $min\left(\phi_c, \phi_{c,max}\right)$, again to promote stability.

The Lagrangian dispersed phase equations, when integrated over a cell, yield the changes in the momentum, mass, and energy of each particle between its entry and exit. The sum of these changes for all particles crossing the volume provides the net momentum, mass, and energy that is exchanged with the continuous phase. These net exchanges enter the continuous phase equations as source terms. For massless particles, the source terms are not computed as these particles do not influence the continuous phase.

Momentum Transfer

The rate of momentum transfer to a single particle from the continuous phase is $\mathbf{F}_s + \dot{m}_p\mathbf{v}_p$, where $\mathbf{F}_s$ is the force acting on surface of the particle, defined in Eqn. (2968), and $\dot{m}_p$ the rate of mass transfer to the particle.

In unsteady simulations, the rate of momentum transfer from all particles in a cell c to the continuous phase is:

$$\mathbf{S_v} = -\frac{1}{\Delta t}\sum_{\pi}\left(\int_t^{t+\Delta t}\int_{V_c}\delta\left(\mathbf{r}-\mathbf{r}_\pi\right)n_\pi\left(\mathbf{F}_s + \dot{m}_p\mathbf{v}_p\right)dV\,dt\right) \tag{3035}$$

where the volume integral is over the cell; the Dirac delta function filters out parcels which are not in the cell. The summation is over all parcels for which two-way coupling is active. The discrete form of Eqn. (3035) is:

$$\mathbf{S_v} = -\frac{1}{\Delta t}\sum_{\pi}\sum_{\delta t_p} n_\pi\left(\mathbf{F}_s + \dot{m}_p\mathbf{v}_p\right)\delta t_p \tag{3036}$$

where the second summation is over all δt_p for which parcel π is in cell c.

In steady simulations, the rate of momentum transfer from all particles in a cell c to the continuous phase is

$$\mathbf{S_v} = -\sum_{\pi}\left(\int_0^{\infty}\int_{V_c}\delta(\mathbf{r}-\mathbf{r}_{\pi})\dot{n}_{\pi}\left(\mathbf{F}_s + \dot{m}_p\mathbf{v}_p\right)dV\,dt\right) \tag{3037}$$

or, in discrete form,

$$\mathbf{S_v} = -\sum_{\pi}\sum_{\delta t_p}\dot{n}_{\pi}\left(\mathbf{F}_s + \dot{m}_p\mathbf{v}_p\right)\delta t_p \tag{3038}$$

The field $\mathbf{S_v}$ is applied in the continuous phase momentum equation.

Mass Transfer

The rate of mass transfer to a single particle from the continuous phase is $\dot{m}_p$.

In unsteady simulations, the rate of mass transfer from all particles in a cell c to the continuous phase is

$$S_m = -\frac{1}{\Delta t}\sum_{\pi}\left(\int_t^{t+\Delta t}\int_{V_c}\delta(\mathbf{r}-\mathbf{r}_{\pi})n_{\pi}\dot{m}_p\,dV\,dt\right) \tag{3039}$$

where the volume integral is over the cell; the Dirac delta function filters out parcels which are not in the cell. The summation is over all parcels for which two-way coupling is active. The discrete form of Eqn. (3039) is

$$S_m = -\frac{1}{\Delta t}\sum_{\pi}\sum_{\delta t_p}n_{\pi}\dot{m}_p\delta t_p \tag{3040}$$

where the second summation is over all δt_p for which parcel π is in cell c.

In steady simulations, the rate of mass transfer from all particles in a cell c to the continuous phase is

$$S_m = -\sum_{\pi}\left(\int_0^{\infty}\int_{V_c}\delta(\mathbf{r}-\mathbf{r}_{\pi})\dot{n}_{\pi}\dot{m}_p\,dV\,dt\right) \tag{3041}$$

or, in discrete form,

$$S_m = -\sum_{\pi}\sum_{\delta t_p}\dot{n}_{\pi}\dot{m}_p\delta t_p \tag{3042}$$

The field S_m is applied in the continuous phase continuity equation.

Energy Transfer

The rate of total energy transfer to a single particle from the continuous phase is

$$Q_t + \mathbf{F}_s\cdot\mathbf{v}_p + \frac{1}{2}\dot{m}_p\mathbf{v}_p^2 + \dot{m}_p h \tag{3043}$$

where Q_t is the surface heat transfer defined in Eqn. (3025), $\mathbf{F}_s$ is the surface force that is defined in Eqn. (2968), $\dot{m}_p$ the rate of mass transfer to the particle and h the enthalpy of transferred material.

In unsteady simulations, the rate of energy transfer from all particles in a cell c to the continuous phase is

$$S_E = -\frac{1}{\Delta t}\sum_{\pi}\left(\int_t^{t+\Delta t}\int_{V_c}\delta(\mathbf{r}-\mathbf{r}_\pi)n_\pi\left(Q_t + \mathbf{F}_s\cdot\mathbf{v}_p + \frac{1}{2}\dot{m}_p\mathbf{v}_p^2 + \dot{m}_p h\right)dV\,dt\right) \tag{3044}$$

where the volume integral is over the cell; the Dirac delta function filters out parcels which are not in the cell. The summation is over all parcels for which two-way coupling is active. The discrete form of Eqn. (3044) is

$$S_E = -\frac{1}{\Delta t}\sum_{\pi}\sum_{\delta t_p} n_\pi\left(Q_t + \mathbf{F}_s\cdot\mathbf{v}_p + \frac{1}{2}\dot{m}_p\mathbf{v}_p^2 + \dot{m}_p h\right)\delta t_p \tag{3045}$$

where the second summation is over all δt_p for which parcel π is in cell c.

In steady simulations, the rate of energy transfer from all particles in a cell c to the continuous phase is

$$S_E = -\sum_{\pi}\left(\int_0^{\infty}\int_{V_c}\delta(\mathbf{r}-\mathbf{r}_\pi)\dot{n}_\pi\left(Q_t + \mathbf{F}_s\cdot\mathbf{v}_p + \frac{1}{2}\dot{m}_p\mathbf{v}_p^2 + \dot{m}_p h\right)dV\,dt\right) \tag{3046}$$

or, in discrete form,

$$S_E = -\sum_{\pi}\sum_{\delta t_p}\dot{n}_\pi\left(Q_t + \mathbf{F}_s\cdot\mathbf{v}_p + \frac{1}{2}\dot{m}_p\mathbf{v}_p^2 + \dot{m}_p h\right)\delta t_p \tag{3047}$$

The field S_E is applied in the continuous phase energy equation.

Particle Injection

Particles enter the computational domain through injectors at one or more discrete locations. An injector defines the size and the velocity vector distribution of the particles, and for unsteady simulations, the frequency. For heat and mass transfer effects, particle temperature and composition are also specified.

STAR-CCM+ provides several types of injectors. For some of them, a statistical particle size distribution is specified.

Particle Size Distributions

A range of particle sizes that are generated by an injector are represented statistically by a particle size distribution.

In STAR-CCM+, a particle size distribution is quantified using a cumulative distribution function (CDF) $F(D)$. Here, *size* means diameter or mass, depending on the particle size specification. The symbol D is used for either. The flow rate specification of the injector determines the precise definition of the cumulative distribution function:

- when the mass flow rate is specified, the cumulative distribution function $F_m(D)$ gives the fraction of the mass flow rate of the injector with a size smaller than D. Hence $F(D_M) = 0.5$ defines D_M as the mass-median size.
- when the volume flow rate is specified, the cumulative distribution function $F_V(D)$ gives the fraction of the volume flow rate of the injector with a size smaller than D. Hence $F(D_M) = 0.5$ defines D_M as the volume-median size.
- when the particle flow rate is specified, the cumulative distribution function $F_N(D)$ gives the fraction of the particle flow rate of the injector with a size smaller than D. Hence $F(D_M) = 0.5$ defines D_M as the number-median size.

This definition provides three essential properties of a cumulative distribution function:

$$\begin{gathered} \lim_{D \to 0} F(D) = 0 \\ \lim_{D \to \infty} F(D) = 1 \\ \frac{dF(D)}{dD} \geq 0 \end{gathered} \tag{3048}$$

STAR-CCM+ provides four internal cumulative distribution functions:

- Log-Normal
- Normal
- Nukiyama-Tanazawa
- Rosin-Rammler

In addition, the function is tabulated as pairs of $(D, F(D))$ values. Make sure that these values respect the properties in Eqn. (3048), with the additional constraint that CDF values must be strictly increasing—duplicate values are not permitted.

Given a cumulative distribution function, STAR-CCM+ generates particle sizes by dividing the distribution into ranges of equal mass, volume, or number (depending on how the flow rate is prescribed). The number of ranges is the number of parcel streams for the injector.

Log-Normal

The log-normal size distribution [584] is a normal (Gaussian) distribution which uses the logarithm of the particle size as the independent variable. Its cumulative distribution function is

$$F(D) = \frac{1}{2}\left[1 + erf\left(\frac{\ln D - \ln \overline{D}}{\sigma\sqrt{2}}\right)\right] \tag{3049}$$

in which the parameters are the standard deviation σ and the mean size $\overline{D}$.

Normal

The normal size distribution is a normal (Gaussian) distribution which uses the particle size as the independent variable. Its cumulative distribution function is

$$F(D) = \frac{1}{2}\left[1 + erf\left(\frac{D - \overline{D}}{\sigma\sqrt{2}}\right)\right] \tag{3050}$$

in which the parameters are the standard deviation σ and the mean size $\overline{D}$.

Nukiyama-Tanazawa

The Nukiyama-Tanazawa size distribution is a probability density function that is based on empirical studies of liquid droplet size in spray injectors. The size probability distribution function is:

$$F(D) = P(a,\ b) \tag{3051}$$

where P is defined using the complete ($\Gamma(a)$) and lower incomplete ($\gamma(a,b)$) gamma functions:

- $P\left(a,b\right) = \frac{\gamma(a,b)}{\Gamma(a)} = \frac{\int_0^b e^{-t} t^{a-1} dt}{\int_0^\infty e^{-t} t^{a-1} dt}$
- $a = \frac{\alpha + 1}{\beta}$
- $b = BD^\beta$
  where:

$$B = \left(\frac{1}{\overline{D}} \frac{\Gamma\left(\frac{\alpha+4}{\beta}\right)}{\Gamma\left(\frac{\alpha+3}{\beta}\right)}\right)^\beta$$

$\overline{D}$ is the mean size. The values of α and β can be set in the model. By default, $\alpha = 2$ and $\beta = 1$.

Rosin-Rammler

The Rosin-Rammler distribution [584] was developed to describe the volume distribution of particles as a function of their diameter, $F_V\left(D_p\right)$. In STAR-CCM+, it is extended to be a generic size distribution, with cumulative distribution function

$$F(D) = 1 - \exp\left[-\left(\frac{D}{D_{ref}}\right)^q\right] \tag{3052}$$

in which the parameters are the exponent q and the reference size D_{ref}. This form identifies the Rosin-Rammler distribution as a Weibull distribution.

As noted above, the Rosin-Rammler distribution is a cumulative mass, volume, or number distribution, depending on the flow rate specification of the injector.

Sprays and Droplets

The term *spray modeling* refers to a subset of discrete phase particle modeling that describes the breakup of a continuous liquid into droplets. A number of industrial processes rely on turning quantities of liquid into vapor. One of the most common examples is liquid fueled combustion.

Liquid fuels are convenient to transport, store, and a wide variety of fuels are stable in liquid form. To burn such stable liquid fuels in an engine, intimate mixing must occur between the fuel and the oxidizer in a gaseous state. This gaseous mixing entails turning the liquid into a vapor and promoting mixing of the resulting gases.

There are two subsets of spray modeling: primary atomization and secondary atomization. Primary atomization makes assumptions about the physics inside the nozzle and computes an initial drop size in the region near the nozzle. Secondary atomization models droplets that traverse the domain where they become hydrodynamically unstable and break into smaller droplets. Droplets can interact with each other directly through collision, creating larger droplets and changing the dynamics of the spray plume.

Realistic simulations of spray driven combustion systems involve not only detailed simulation of droplet dynamics, but also mass transfer from vaporization, interaction with the (nearly always turbulent) Eulerian flow, chemical reactions, and resulting large temperature gradients as well as radiation to the surroundings. Liquid fuels are burned in a wide variety of devices, each with particular regimes with respect to spray modeling. Thus, a single spray model is unlikely to be universally applicable to all conditions and experimental validation of a combustion system is still a necessity.

Contents:

Primary Atomization
Secondary Breakup
Collision and Coalescence
Wall Impingement

Primary Atomization

Primary atomization refers to the process of forcing liquid through a small orifice at a high pressure, resulting in a fine spray of liquid droplets. The injection velocity, that is, the velocity of the liquid as it exits the nozzle and enters the computational domain, is one of the most important parameters in a spray calculation. It strongly influences the atomization and break-up process, the spray penetration, the inter-phase transfer processes, and also the inter-droplet and droplet-wall interactions.

The nozzle type injector contains capabilities for modeling the behavior of special-purpose atomizers that are used for combustion processes. The injection process that is modeled includes the flow in the nozzle hole with the exit velocity and the atomization. The atomization creates a liquid droplet spray that is represented by computational parcels. The parcel properties such as velocity, size, temperature, and density are calculated by the atomization models.

Depending on the internal construction of the injector, the liquid at the exit of an injector nozzle can be in the form of a thin liquid sheet or a jet. STAR-CCM+ currently provides two models that simulate the atomization process of the liquid exiting the nozzle:

- LISA model for the breakup of a thin liquid sheet that is produced with a pressure-swirl atomizer
- Huh model for the breakup of a liquid jet created through a simple hole-type nozzle

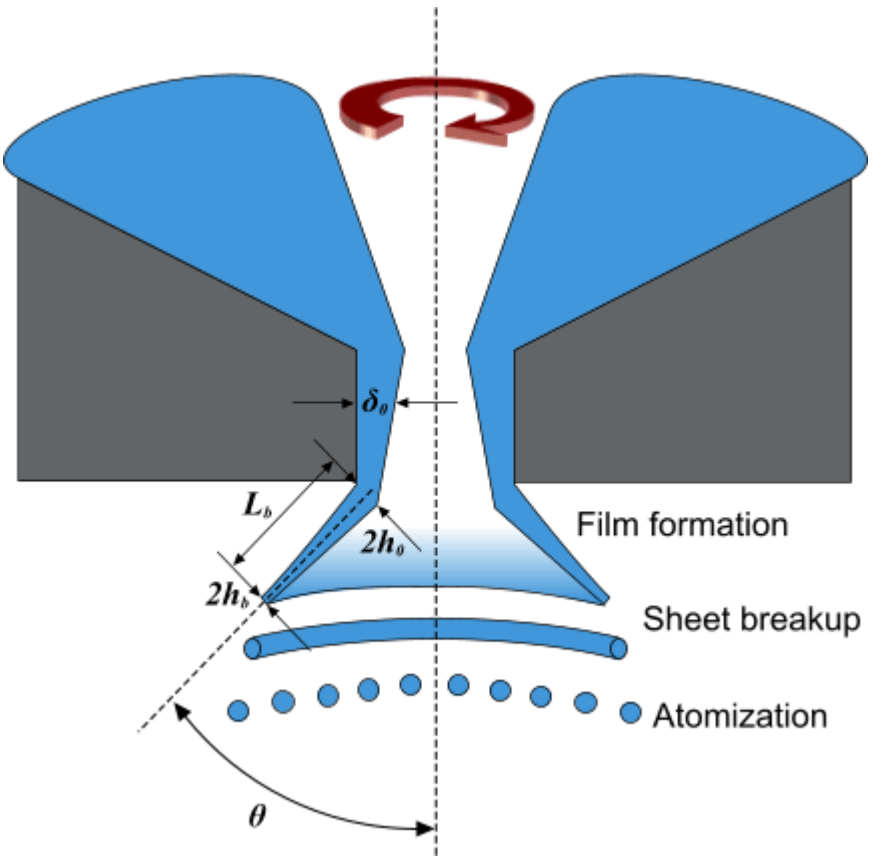
Both models require two basic parameters: the orifice diameter at the nozzle exit and the injector axis.

LISA Model

The Linearized Instability Sheet Atomization (LISA) model is a primary atomization model for the thin liquid sheet at the nozzle exit that is created by the action of a pressure-swirl atomizer.

The pressure-swirl atomizer, also commonly known as a simplex atomizer, is widely used for liquid-fuel injection in gas turbines, oil furnaces, and direct-injection spark-ignited automobile engines.

In this atomizer type, a swirling motion is imparted to the injected liquid through nozzles that are known as swirl ports. The liquid accelerates as it flows through the swirl ports into a central swirl chamber. Under the action of centrifugal force, it spreads out in the form of a conical sheet and develops a hollow air core. It emerges from the orifice as an unstable thinning liquid sheet, then breaking up into ligaments and then droplets. A sketch of a hollow-cone pressure swirl injector is shown in the figure below.



In the illustration above, three stages in the atomization process are identified: film formation, sheet breakup, and atomization. δ_o is the initial film thickness in the nozzle; $2h_0$ is the initial thickness of the film as it exits the nozzle, and $2h_b$ is the thickness of the film at the point of breakup. θ is the spray angle, which is defined as $0.5 \times (\text{inner cone angle} + \text{outer cone angle})$.

The LISA model combines the sheet breakup and atomization stage into one, so that there are two stages modeled overall.

The LISA model that is implemented in STAR-CCM+ is based primarily on the work of Senecal and others [607] and Schmidt and others [605].

Assumptions that are made in the LISA model are:

- A hollow-cone spray forms.
- The model only calculates a mean droplet diameter, and generates droplets at the injector position by assuming a Rosin-Rammler size distribution. The droplets are distributed uniformly in the spray cone.
- The primary breakup model does not operate in the spray region. The breakup length L_b is assumed to be small, and the injection flow rate is steady. The atomized particles appear at the injection point immediately.
- Most numerical models assume that the slip velocity between the liquid sheet and the surrounding air is equal to the absolute velocity of the liquid near the injector ($v_s \equiv V_f$).

Film Formation

The centrifugal motion of the liquid within the injector creates a liquid film surrounding an air core. The thickness and velocity of the liquid film are calculated based on the input parameters for the pressure swirl injector. The initial sheet thickness within the injector, δ_0, is

$$\dot{m}_f = \pi \rho u_f \delta_0 \left(d_0 - \delta_o\right) \tag{3053}$$

where $\dot{m}_f$ is the mass flowrate through the injector (specified by you), ρ is the density of the continuous phase, and d_0 is the injector exit diameter.

The axial component u_f of the film velocity $\mathbf{v}_f$ at the injector exit is difficult to calculate from first principles. Instead, the approach from Han [580] is used. In this approach, the injector exit velocity profile is assumed to be uniform, and the absolute film velocity magnitude V_f is related to the injection pressure head Δp by:

$$V_f = \left|\mathbf{v}_f\right| = k_v \sqrt{\frac{2\Delta p}{\rho}} \tag{3054}$$

where $\mathbf{v}_f = u_f \mathbf{i} + v_f \mathbf{j}$

The value of the velocity coefficient k_v is obtained [580] as

$$k_v = max\left[0.7, \frac{4\dot{m}}{\pi d_0^2 \rho_f \cos\theta} \sqrt{\frac{\rho_f}{2\Delta p}}\right] \tag{3055}$$

The constant axial component is then

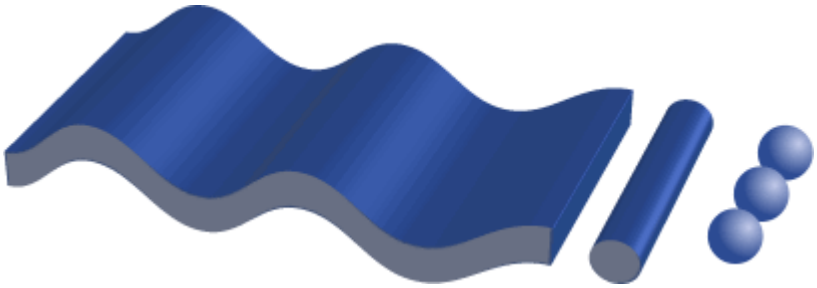
$$u_f = V_f \cos\theta \tag{3056}$$

where θ is the spray angle.

Sheet Breakup and Atomization

This section is only a brief summary of the breakup and atomization stage: details can be obtained from the paper by Senecal and others [607]. The model represents the film exiting the injector as a two-dimensional, viscous, incompressible liquid sheet of thickness $2h$ moving through a quiescent gas medium. The model

also includes the effects of surrounding gas, liquid viscosity, and surface tension on the breakup of the liquid sheet. This phenomenon is depicted in the figure below.



Fluctuating velocities and pressure for both the liquid and the surrounding gas are produced by imposing a spectrum of infinitesimal wave-type disturbances on the initially steady motion. These disturbances, which are defined within a coordinate system that moves with the sheet, have the form:

$$\eta = \eta_0 e^{(ikx + \omega t)} \tag{3057}$$

In this equation, η_0 is the initial wave amplitude, $k = 2\pi/\lambda$ is the wavenumber, and $\omega = \omega_r + i\omega_i$ is the complex growth rate with i being the imaginary unit ($i^2 = -1$). The most unstable disturbance has the largest value of ω_r, denoted by Ω_s, and is assumed to be responsible for breakup. It is therefore desired to obtain a dispersion relation $\omega = \omega(k)$ from which the most unstable disturbance can be deduced. It has been shown that two solutions or modes exist that satisfy the conservation equations and their associated boundary conditions at the upper and lower interfaces for the disturbed liquid sheet. For the first solution, called the sinuous mode, the waves on the upper and lower boundaries are in phase. For the second solution, called the varicose mode, the waves are π radians out of phase. Numerous authors (for example, Squire [611], Rangel and Sirignano [595], and Senecal and others [607]) have shown that the sinuous mode is dominant for low velocities and low gas to liquid density ratios. In addition, the modes become indistinguishable for high velocity flows.

As derived in Senecal and others [607], the dispersion relation for the sinuous mode is

$$\omega^2\left[\tanh(kh) + \Gamma\right] + \omega\left[4v_f k^2 \tanh\left((kh) + 2i\Gamma k V_s\right)\right] + 4v_f^2 k^4 \tanh\left(kh\right) - 4v_f^2 k^3 \xi \tanh(\xi h) - \Gamma V_s^2 k^2 + \frac{\sigma k^3}{\rho_f} = 0 \tag{3058}$$

where V_s is the magnitude of the relative velocity between the liquid film and the gas, $\Gamma = \rho/\rho_f$, $\xi = k^2 + \omega/v_f$, and σ is the surface tension.

With some simplifications, the growth rate for the sinuous mode is

$$\omega_r = -\frac{2v_f k^2 \tanh\left(kh\right)}{\tanh(kh) + \Gamma} + \frac{\sqrt{4v_f^2 k^4 \tanh^2\left(kh\right) - \Gamma^2 V_s^2 k^2 - \left[\tanh(kh) + \Gamma\right]\left(-\Gamma V_s^2 k^2 + \sigma k^3/\rho_f\right)}}{\tanh(kh) + \Gamma} \tag{3059}$$

This expression can be simplified further depending on whether the disturbance waves being formed are long or short. The distinction between short or long waves is made by calculating the Weber number for the gas phase, $We_g = \rho {V_s}^2 h/\sigma$ (based on the relative liquid velocity, sheet half-thickness, and gas density).

Short waves are formed for $We_g > 27/16$, corresponding to high-speed flows with $\rho/\rho_f \ll 1$; otherwise long waves (corresponding to low-speed flows) are formed. Most modern injectors produce short waves.

Short Waves

For short waves the simplified expression for the growth rate is

$$\omega_r = -2\nu_f k^2 + \sqrt{4{\nu_f}^2 k^4 + \Gamma {V_s}^2 k^2 - \frac{\sigma k^3}{\rho_f}} \tag{3060}$$

Here, $k = K_s$ such that ω_r has its maximum value (denoted by Ω_s). For this condition, the sheet breaks up at a length and time that are given, respectively, by

$$\begin{aligned} L_b &= V_f \tau_b \\ \tau_b &= \frac{\zeta_b}{\Omega_s} \end{aligned} \tag{3061}$$

where $\zeta_b = \ln(\eta_b/\eta_0)$ is the sheet breakup empirical constant that you define (default value equal to 12). The ratio η_b/η_0 is the ratio of the surface disturbance at sheet breakup to the initial disturbance. The sheet half-thickness h_b [595] at the breakup is

$$h_b = \frac{h_0(d_0 - \delta_0)}{2L_b \sin\theta + d_0 - \delta_0} \tag{3062}$$

where

$$h_0 = 0.5\delta_0 \cos\theta \tag{3063}$$

The diameter of the ligament at sheet breakup is

$$d_L = \sqrt{\frac{16h_b}{K_s}} \tag{3064}$$

Long Waves

Viscosity has a minor effect on wave growth in the long wave regime. In this case, K_s is directly obtained from an inviscid analysis as:

$$K_s = \frac{\rho_g {V_s}^2}{2\sigma} \tag{3065}$$

For the inviscid case:

$$\Omega_s = \sqrt{\frac{\Gamma {V_s}^2 {K_s}^2 - \frac{\sigma {K_s}^3}{\rho_f}}{K_s h_b}} \tag{3066}$$

$$L_b = V_s \tau_b = \frac{V_s \zeta_b}{\Omega_s} \tag{3067}$$

Eqn. (3062), Eqn. (3066) and Eqn. (3067), determine the sheet half-thickness h_b at the sheet breakup. The diameter of the ligament at sheet breakup is:

$$d_L = \sqrt{\frac{8h_b}{K_s}} \tag{3068}$$

Droplet Diameter Distribution

After calculating d_L, the mean diameter d_D for the Rosin-Rammler distribution (with a spread parameter of 3.5 and a specified dispersion angle) is

$$d_D = 1.88 d_L \left(1 + 3Oh\right)^{1/6} \tag{3069}$$

where Oh, the Ohnesorge number, is

$$Oh = \frac{\mu_f}{\left(\rho_f \sigma d_L\right)^{1/2}} \tag{3070}$$

Huh Model

Huh's model is based on the premise, supported by order-of-magnitude estimates, that the two most important mechanisms in spray atomization are the gas inertia and the internal turbulence stresses generated in the nozzle.

The Huh model divides atomization into two stages:

1. The turbulence that is generated in the nozzle hole produces initial perturbations on the jet surface when it exits the hole.
2. Once the perturbations have reached a certain level, they grow exponentially by pressure forces. These forces are induced through interaction with the surrounding fluid (surface wave growth), until these perturbations become detached from the jet surface as droplets.

The model estimates the initial perturbations from an analysis of the flow through the hole and then uses established wave growth theory, together with other hypotheses, to represent the atomization process.

Nozzle Turbulence and Its Decay

The average turbulence kinetic energy k_a and its dissipation rate ϵ_a at the hole exit are calculated from the following relations, which are derived from force and energy balances:

$$\begin{aligned} k_a &= \frac{U^2}{8\frac{L}{D}}\left(\frac{1}{c_d^2} - K_c - 1\right) \\ \epsilon_a &= \frac{K_\epsilon U^3}{2L}\left(\frac{1}{c_d^2} - K_c - 1\right) \end{aligned} \tag{3071}$$

where:

- U is the average injection velocity over the time period of injection.
- L is the nozzle length, the distance from the primary atomization point to the point of secondary breakup.
- D is the nozzle diameter.
- K_c is the form loss coefficient, default value 0.45. It is a proportionality constant for the pressure loss associated with the sharpness of the nozzle entrance corner.

$$K_c = \frac{2\Delta p_f}{\rho U^2}$$

where:

- Δp_f is the pressure loss across the nozzle corner.
- ρ is the density of the liquid.

- K_ϵ is an empirical coefficient, default value 0.5, proportional to the initial dissipation rate of turbulent kinetic energy at the nozzle exit.
- c_d is the nozzle discharge coefficient, default value is 0.6. It is a proportionality constant for the rate of volume flow through the nozzle.

$$c_d = \frac{\dot{Q}}{A_0\sqrt{\frac{2\Delta p}{\rho}}}$$

where:

- A_0 is the nozzle cross-section.
- $\dot{Q}$ is the rate of volume flow.
- Δp is the pressure drop across the nozzle.

If:

$$\left(\frac{1}{c_d^2} - K_c - 1\right) \leq 0$$

Eqn. (3071) simplifies to:

$$\begin{aligned} k_a &= \frac{1.5}{400}U^2 \\ \epsilon_a &= C_\mu^{3/4}\frac{k_a^{3/2}}{0.01\,L} \end{aligned} \tag{3072}$$

The corresponding turbulence length scale L_t^o and time scale τ_t^o are:

$$\begin{aligned} L_t^o &= C_\mu^{3/4}\frac{k_a^{3/2}}{\epsilon_a} \\ \tau_t^o &= C_\mu^{3/4}\frac{k_a}{\epsilon_a} \end{aligned} \tag{3073}$$

where C_μ is the coefficient of the $k-\epsilon$ model. Each injected (parent) droplet is assigned these scales as the initial values. Thereafter the turbulence within it decays with time. The following relations describe this time dependence:

$$L_t(t) = L_t^o\left(1 + C_{a1}\frac{t}{\tau_t^o}\right)^{C_{a2}} \tag{3074}$$

$$\tau_t(t) = \tau_t^o\left(1 + C_{a1}\frac{t}{\tau_t^o}\right) \tag{3075}$$

where C_{a1} (default value 0.92) and C_{a2} (default value 0.4565) are model coefficients.

Surface Wave Growth

The second stage of the atomization process is the interaction between liquid jet and surrounding gas field. This interaction increases the amplitude of the perturbation with wave growth rate $\mathrm{Re}[\omega]$ until these perturbations become detached from the jet surface ($\mathrm{Re}[\omega]$ is the real part of the complex wave celerity ω. The value of $\mathrm{Re}[\omega]$ is equal to the reciprocal value of the wave growth time scale, τ_W). These detached parts create the so-called 'secondary droplets'. The perturbation amplitude obeys the dispersion equation, which is derived by Taylor [613]:

$$\left(\omega + 2\nu\kappa^2\right)^2 + \frac{\sigma\kappa^3}{\rho_d} - 4\nu^2\kappa^3\left(\kappa^2 + \frac{\omega}{\nu}\right)^{1/2} + (\omega + iU\kappa)^2\frac{\rho}{\rho_d} = 0 \tag{3076}$$

where:

- σ is the coefficient of surface tension.
- ν is the liquid kinematic viscosity.
- $\kappa = 2\pi/L_W$ is the wavenumber.
  - $L_W = L_A/C_2$ is the wavelength of surface perturbation. C_2 is a model coefficient, default value 0.5.
  - $L_A = C_1 L_t$ is the atomization length scale. C_1 is a model coefficient, default value 2.0.
- ρ is the gas density.
- i is the imaginary unit ($i^2 = -1$).

The time scale of atomization τ_A is taken to be a linear combination of the turbulent time scale τ_t and the wave growth time scale τ_W:

$$\tau_A = C_3\tau_t + C_4\tau_W = \tau_{spn} + \tau_{exp} \tag{3077}$$

where:

- $\tau_{spn} = C_3\tau_t$ is the spontaneous time scale; C_3 is a model coefficient, default value 1.0.
- $\tau_{exp} = C_4\tau_W$ is the exponential time scale; C_4 is a model coefficient, default value 1.5.

Droplet Size Distribution

This model assumes that droplets leave the nozzle with diameter D_d then break up into secondary droplets. The break-up rate for droplets is:

$$\frac{dD_d}{dt} = -\frac{2L_A}{\tau_A K_A} \tag{3078}$$

where:

- D_d is the diameter of the parent droplet.
- K_A is a model constant to control the break-up rate, default value 0.1.

This initial break-up process stops when the spontaneous time scale becomes greater than the exponential scale.

The diameter of the secondary droplets that are formed from parent droplet break-up is estimated from the following probability density function:

$$f(x) = C\frac{\Phi(x)}{\tau_A(x)} \tag{3079}$$

where:

- x is droplet size.
- C is a normalization constant.
- $\Phi(x)$ is a dimensionless turbulence energy spectrum (assuming isotropic turbulence), estimated as:

$$\Phi(x) = \frac{\left(\kappa(x)/\kappa_e\right)^2}{\left(1 + \left(\kappa(x)/\kappa_e\right)^2\right)^{11/6}} \tag{3080}$$

  where $\kappa_e = 2\pi/L_e$, and $L_e = L_t/0.75$ with L_t being the turbulence length scale.

The distribution function for droplet diameters, $F(x)$, is then:

$$F(x) = \int_{x_{\min}}^{x} f(x)\, dx \tag{3081}$$

$x_{\min}$ is the minimum droplet size, which is calculated from Kelvin-Helmholtz instability theory as:

$$x_{\min} = 2\pi\frac{\sigma_d\left(\rho_d + \rho\right)}{U^2\rho_d\rho} \tag{3082}$$

where:

- σ_d is the surface tension of the droplet.
- ρ_d is the density of the droplet.
- ρ is the density of the gas.

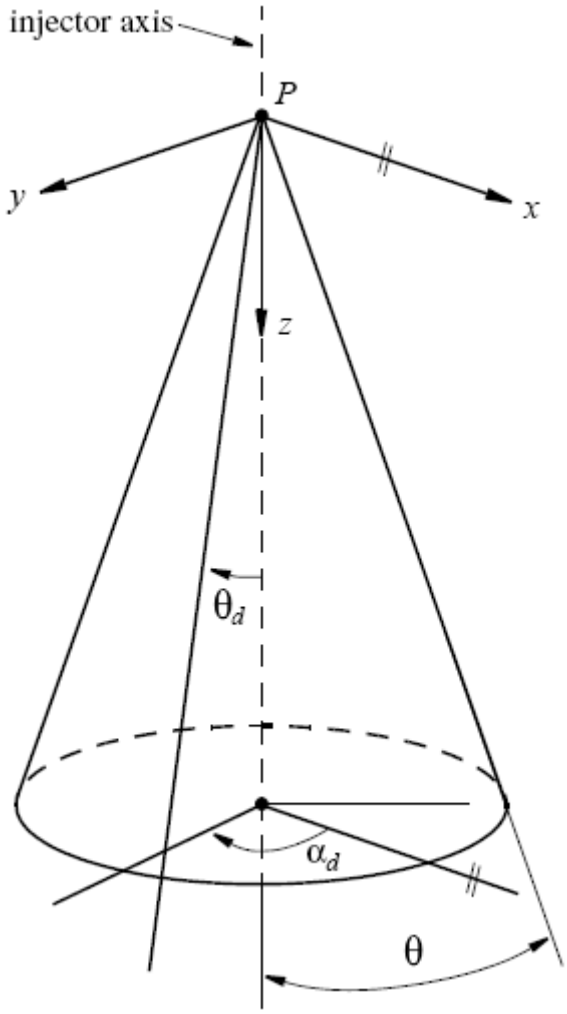
Droplet Velocity Distribution

The initial velocity for each droplet is characterized by two angles: the polar angle θ_d and the azimuthal angle α_d. The polar angle is based on the assumption of equal probability of velocity direction between the user-specified inner cone angle θ_i and outer cone angle θ_o.

$$\theta_d = \theta_i + X_r\left(\theta_o - \theta_i\right) \tag{3083}$$

where X_r is a random number in the range [0, 1]. The azimuthal angle is taken as the user-specified swirl angle. Assuming that the magnitude of the droplet velocity at nozzle exit is the liquid film velocity V_f, the components of droplet velocity are calculated as:

$$v_{d.x} = V_f \sin\left(\theta_d\right)\ \sin\left(\alpha_d\right)$$
$$v_{d.y} = V_f \sin\left(\theta_d\right)\ \cos\left(\alpha_d\right)$$
$$v_{d.z} = V_f \cos\left(\theta_d\right)$$



Secondary Breakup

Liquid droplets sometimes break up under the action of non-uniform surface forces that are induced by their motion relative to the continuous phase. This phenomenon known as secondary breakup. STAR-CCM+ provides a set of models that are used for the break-up of liquid droplets in a gaseous stream.

The response of droplets to non-uniform surface forces is to deform, with the deformation resisted by surface tension and viscous forces inside the droplet. Characteristic measures of this behavior are the Weber and Ohnesorge numbers:

$$\mathrm{We} = \frac{\rho_g |\mathbf{v}_s|^2 D_p}{\sigma} \tag{3084}$$

$$\mathrm{Oh} = \frac{\mu_l}{\sqrt{\rho_l D_p \sigma}} \tag{3085}$$

Depending on the magnitude of these forces, various breakup regimes have been identified, each characterized by the shape of the deforming droplets. An example categorization (after Stiesch [612]) for low Oh is given below.

Category		Weber Number
Vibrational breakup		~ 12
Bag breakup		< 20
Bag/ streamer breakup		< 50
Stripping breakup		< 100
Catastrophic breakup		> 100

Detailed modeling of even one breakup regime is difficult. At most, the goal of secondary breakup models is to predict when breakup occurs and what diameters result from it.

Contents:

KHRT Breakup Model
Reitz-Diwakar Breakup Model
TAB Distortion and Breakup Model
SSD Breakup Model

KHRT Breakup Model

The KHRT Breakup model combines two submodels, one based on Kelvin-Helmholtz (KH) theory and one based on Rayleigh-Taylor (RT) theory. Both breakup submodels consider the growth of instabilities on a droplet and provide expressions for their wavelength and frequency.

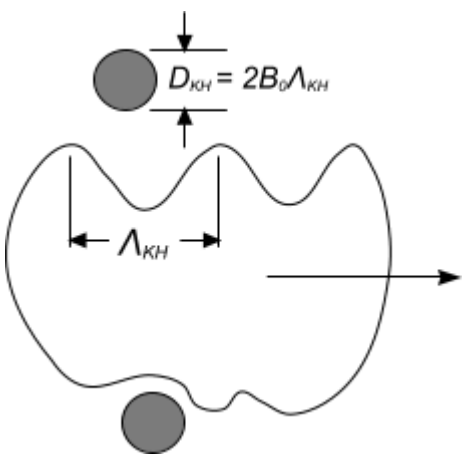
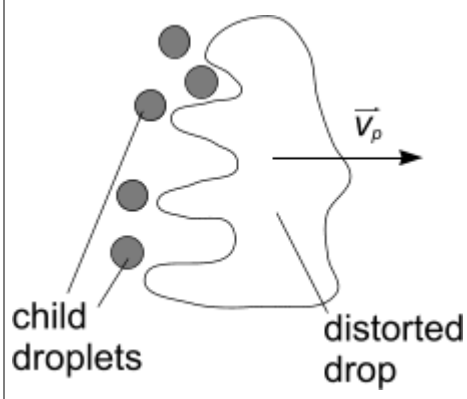
Kelvin-Helmholtz instabilities are due to the slip velocity of the droplet, which eventually shears small child droplets off the parent [597], corresponding to the stripping regime. Rayleigh-Taylor instabilities are due to the acceleration of the droplet and tend to shatter the droplet completely [594], corresponding to the catastrophic regime. The KH and RT submodels compete: instabilities due to both can grow simultaneously; if they grow for long enough, breakup occurs due to the RT instabilities. Otherwise KH breakup occurs.

Because there is a tendency for the KH submodel to generate a bi-modal size distribution, the smaller child droplets are accumulated into a new parcel; the parent parcel retains the larger parent droplets. However, to prevent too many child parcels being created, these parcels are only released when their mass reaches a predefined proportion of the original mass of the parent. Child parcels are given a lateral velocity proportional to the kinetic energy of the droplet oscillation at the instant of breakup. This velocity tends to generate a spreading effect from an injector which, to some extent, makes a cone injector unnecessary.

The KHRT model was developed as a replacement for the TAB model [594]. Although the latter can still be preferred at low Weber numbers, the KHRT model is preferred at higher Weber numbers. The principal difficulty in using the model is tuning the coefficients to the application under consideration. This difficulty reflects the complexity of the physics and is not due to a flaw in the model.

Kelvin-Helmholtz instabilities are postulated as the cause of stripping breakup.

The model schematics and key equations are illustrated below.

KH Model	RT Model
	
Child drop diameter $= 2B_0\Lambda_{KH}$	Child drop diameter $= C_3\Lambda_{RT}$
Breakup time constant $= \frac{1.894B_1D_p}{\Lambda_{KH}\Omega_{KH}}$	Breakup time constant $= \frac{C_\tau}{\Omega_{RT}}$

Reitz [597] proposed expressions for the wavelength and growth-rate of these instabilities:

$$\Lambda_{KH} = \frac{4.51D_p\left(1 + 0.45\mathrm{Oh}_r^{0.5}\right)\left(1 + 0.4\mathrm{Ta}_r^{0.7}\right)}{\left(1 + 0.865\mathrm{We}_r^{1.67}\right)^{0.6}} \tag{3086}$$

$$\Omega_{KH} = \frac{0.34 + 0.385\mathrm{We}_r^{1.5}}{\left(1 + \mathrm{Oh}_r\right)\left(1 + 1.4\mathrm{Ta}_r^{0.6}\right)}\sqrt{\frac{4\pi\sigma}{3m_p}} \tag{3087}$$

where $\mathrm{We}_r = 0.5\mathrm{We}$ is the Weber number, which is based on the droplet radius,

$$\mathrm{Oh}_r = \mu_l\sqrt{\frac{2}{\rho_l D_p \sigma}} \tag{3088}$$

is the Ohnesorge number based on the droplet radius and $\mathrm{Ta}_r = \mathrm{Oh}_r\sqrt{\mathrm{We}_r}$ is the Taylor number that is based on the droplet radius. The wavelength and growth-rate in turn are used to formulate a characteristic time-scale and length-scale for breakup caused by these instabilities:

$$\tau_{KH} = \frac{1.894 B_1 D_p}{\Lambda_{KH}\Omega_{KH}} \tag{3089}$$

$$D_{KH} = 2B_0\Lambda_{KH} \tag{3090}$$

The coefficient B_0 takes the value 0.61. The coefficient B_1 can be tuned to account for uncertainties in, for example, the droplet initial conditions. A wide range of values have been reported in the literature, from 1.73 to 60 or higher.

If $D_p > D_{KH}$, breakup due to KH instabilities is possible, in which case the parent droplet diameter decreases according to:

$$\frac{dD_p}{dt} = \frac{D_{KH} - D_p}{\tau_{KH}} \tag{3091}$$

However, this is allowed to occur only if breakup due to RT instabilities (see below) does not occur.

The mass shed during KH breakup is accumulated [594] until it reaches a user-specified proportion of the parent parcel's original mass, default 3%. One or more child parcels are created, with child droplets having diameter D_{KH}. The diameter of the droplets in the parent parcel remains unchanged; the number of droplets is calculated to conserve mass. Child parcels are also given a velocity normal to the original velocity of the parent parcel:

$$v_\perp = A_1\Lambda_{KH}\Omega_{KH} \tag{3092}$$

The normal velocity coefficient A_1 can be tuned to produce a known spray angle; alternatively the spray angle may be specified directly at a cone injector, in which case A_1 should be set to zero.

Rayleigh-Taylor instabilities are postulated as the cause of "catastrophic" breakup. Following Senecal et al [607], STAR-CCM+ incorporates the effect of viscosity on these instabilities. The relationship between their wavenumber and growth-rate as a function of the droplet acceleration a is:

$$\omega_{RT} = -k_{RT}^2\left(\frac{\mu_l + \mu_g}{\rho_l + \rho_g}\right) + \sqrt{k_{RT}\left(\frac{\rho_l - \rho_g}{\rho_l + \rho_g}\right)a - \frac{k_{RT}^3\sigma}{\rho_l + \rho_g} + k_{RT}^4\left(\frac{\mu_l + \mu_g}{\rho_l + \rho_g}\right)^2} \tag{3093}$$

Eqn. (3093) is solved numerically to give the wavelength Λ_{RT} corresponding to the maximum growth-rate Ω_{RT}. This is assumed to be the critical mode. In the limit of negligible viscosity, this method reproduces the inviscid RT formulation that is used by [594], for example. The justification for including viscosity is that its effects can be significant; implementations which neglect it typically require additional modifications such as a user-defined breakup length [600].

The wavelength and growth-rate in turn are used to formulate a characteristic time-scale and length-scale for breakup that is caused by these instabilities:

$$\tau_{RT} = \frac{C_\tau}{\Omega_{RT}} \tag{3094}$$

$$D_{RT} = C_3 \Lambda_{RT} \tag{3095}$$

The coefficient C_τ takes the default value 1. The coefficient C_3 takes the default value 0.1, with values in the range 0.1-5.33 being reported in the literature.

If $D_p > D_{RT}$, a critical Rayleigh-Taylor instability is assumed to be growing on the droplet. If the instability persists for a duration τ_{RT}, a breakup event occurs. The parent droplets shatter completely, producing child droplets with new diameter D_{RT}.

Reitz-Diwakar Breakup Model

The Reitz-Diwakar breakup model is based on observed length- and time-scales of droplet breakup.

The Reitz-Diwakar breakup model, [598] and [599], assumes that breakup occurs in one of two possible modes:

- “bag” breakup, in which the non-uniform pressure field around the droplet causes it to expand in the low-pressure wake region and eventually disintegrate when surface tension forces are overcome
- “stripping” breakup, in which liquid is sheared or stripped from the droplet surface

In either case, [598] and [599], theoretical studies provide a criterion for the onset of break-up and concurrently an estimate of the stable droplet diameter D_s and the characteristic time-scale τ_b of the breakup process. The droplet diameter decreases according to:

$$\frac{dD_p}{dt} = \frac{D_s - D_p}{\tau_b} \tag{3096}$$

in which D_s is the stable diameter and τ_b the breakup time-scale. Both D_s and τ_b depend on the active breakup regime.

Bag Breakup

The first breakup regime, bag breakup, is caused by the droplet expanding into the low-pressure wake region behind it, and eventually disintegrating when surface tension forces are overcome. It is assumed to be possible if

$$We > We_{crit} \tag{3097}$$

with $We_{crit} = 12$ the default. The stable diameter is obtained by making Eqn. (3097) an equality. The characteristic time-scale for this breakup regime is

$$\tau_b = \frac{C_{b2} D_p}{4} \sqrt{\frac{\rho_l D_p}{\sigma}} \tag{3098}$$

Stripping Breakup

The second breakup regime, stripping breakup, is caused by liquid being sheared or stripped from the droplet surface. It is assumed to be possible if Eqn. (3097) is satisfied and additionally

$$We > \max\left(2C_{s1} Re^{0.5}, We_{crit}\right) \tag{3099}$$

The default value for C_{s1} is 0.5. The stable diameter is obtained by making Eqn. (3099) an equality. The characteristic time-scale for this breakup regime is

$$\tau_b = \frac{C_{s2}}{2}\sqrt{\frac{\rho_l}{\rho}}\frac{D_p}{|\mathbf{v}_s|} \tag{3100}$$

The default value for C_{s2} is 20. When both Eqn. (3097) and Eqn. (3099) are satisfied, the regime with the smallest characteristic time-scale dominates.

TAB Distortion and Breakup Model

The Taylor analogy breakup (TAB) model is based on Taylor's analogy. The analogy represents a distorting droplet as a damped spring-mass system; it considers only the fundamental mode of oscillation of the droplet. The displacement and velocity of the mass in the spring-mass system correspond to representative distortion and rate of distortion quantities for the droplet.

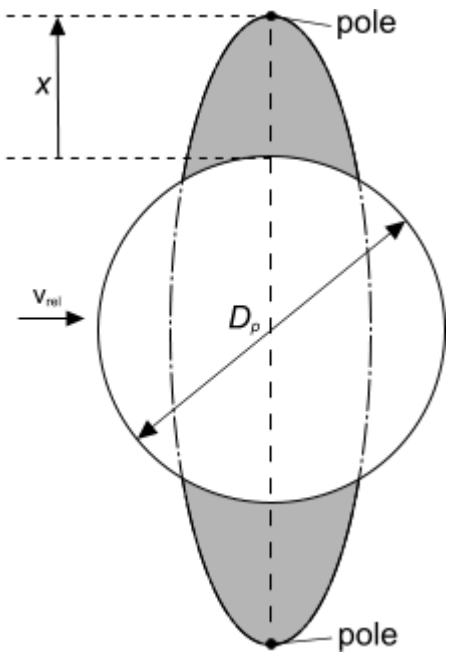
The TAB model equations govern the droplet oscillation and distortion [591]. When the droplet oscillations reach a critical value, breakup replaces the parent particles with child particles whose diameter is chosen from a Rosin-Rammler distribution. By default, the breakup event creates no parcels—the original parcel is retained, but the particle diameter changes to the new child value. New parcels can optionally be created however, in which case the droplet diameter in each is chosen independently. This feature can be useful in modeling the generation of a distribution of droplet sizes through breakup. Parcels that are involved in a breakup event are given a lateral velocity proportional to the kinetic energy of the droplet oscillation at the instant of breakup. This velocity tends to generate a spreading effect from an injector which, to some extent, makes a cone injector unnecessary.

Despite being based on a single mode of oscillation in the vibrational regime, the TAB model reproduces the same characteristic time-scales in low and high Weber number limits as the Reitz-Diwakar model. Typically, however, the TAB model is used at low Weber numbers; hollow-cone gasoline sprays are an example of a preferred application for this model [570]. Outside its range of validity, the model tends to underpredict droplet sizes.

Droplet Distortion

The goal of the TAB distortion model is to calculate the instantaneous displacement x of the droplet equator from its equilibrium position.

This displacement is illustrated in the following diagram (after Baumgarten [570]).



For simplicity of analysis, this displacement normalizes to the droplet diameter

$$y = \frac{2x}{C_b D_p} \tag{3101}$$

in which C_b is an empirical constant. The Taylor analogy between a distorting droplet and a damped spring-mass system [613] gives a second-order ordinary differential equation governing y

$$\frac{\rho_l D_p^2 \dot{y}}{4} = \frac{C_k \sigma}{D_p}\left(\frac{We}{We_{crit}} - 2y\right) - C_d \mu_l \dot{y} \tag{3102}$$

where σ and μ_l are respectively the surface tension and viscosity of the droplet and We its Weber number, Eqn. (3084). C_d, C_k and We_{crit} are empirical constants with default values 5, 8 and 12 respectively. C_b does not appear in Eqn. (3102): it is not a necessary parameter to determine y. Eqn. (3102) is integrated for each droplet from injection, with initial conditions $y = \dot{y} = 0$.

Droplet Breakup

The TAB breakup model relies on the TAB distortion model to calculate the normalized droplet distortion y.

When y reaches unity (Eqn. (3101)), the droplet is assumed to be critically distorted; a breakup event is then triggered. The Sauter mean diameter D_{32} of the post-breakup droplets is related to the diameter D_p of the parent droplet through an energy balance [591], giving

$$\frac{D_p}{D_{32}} = 1 + \frac{K C_b^2 C_k}{5} + \frac{3 C_b^2 m_p}{4\sigma\pi}\left(\frac{K}{5} - \frac{C_v^2}{6}\right)\dot{y}^2 \tag{3103}$$

Here, C_k and C_b are inherited from the TAB distortion model, with C_b taking the default value 0.5. K is the ratio of the total energy in distortion and oscillation to the energy in the fundamental mode, default 10/3. In STAR-CCM+, the diameter of child droplets is obtained by sampling a Rosin-Rammler distribution with the

Sauter mean diameter given by Eqn. (3103) and a user-specified exponent (spread). This is done separately for the parent parcel and, optionally, each child parcel.

Post-breakup parcels are also given a velocity normal to the original velocity of the parent parcel. This is proportional to their rate of distortion at the instant of breakup $\dot{y}_b$

$$v_\perp = \frac{1}{2} C_v C_b D_p \dot{y}_b \tag{3104}$$

In both Eqn. (3103) and Eqn. (3104), C_v is the normal velocity coefficient. This coefficient can be tuned to produce a known spray angle; alternatively the spray angle may be specified directly at a cone injector, in which case C_v should be set to zero.

SSD Breakup Model

In the Stochastic Secondary Droplet (SSD) breakup model, breakup is a random process unrelated to original droplet size. This results in a log normal distribution of droplet diameters in the long time limit.

The particles interact with the continuous fluid phase as they travel. Particles break up when they meet three criteria:

- The particle radius is larger than the critical radius, r_c.
- The Weber number is larger than a critical Weber number, We_{cr}.
- The particle's accumulated residence time is larger than the breakup time. τ_{bu} (Eqn. (3106)). The model tracks parcels as they cross the domain and compares the residence time with a locally evaluated breakup time. The residence time is only calculated on parcels with a radius larger than a critical radius and a Weber number larger than a critical Weber number.

The first two criteria are related according to:

$$r_c = \frac{\mathrm{We}_{cr} \sigma_l}{\rho_g {v_s}^2} \tag{3105}$$

where:

- r_c is the critical radius
- We_{cr} is the critical Weber number specified in the interface. The default value is 6.
- σ_l is the surface tension of the liquid.
- ρ_g is the density of the continuous phase.
- v_s, slip velocity, is the relative velocity of the particle through the continuous phase.

Particles cease to exist when they break up into child particles. In turn, these child particles can also break up, if they meet the criteria above. The higher the critical Weber number, the larger the particles when breakup stops.

Particle break up only if they have a residence time greater than the breakup time. The breakup time scale is defined as:

$$\tau_{\mathrm{bu}} = B_1\sqrt{\frac{\rho_l}{\rho_g}}\frac{r_p}{v_s} \tag{3106}$$

where:

- τ_{bu} is the breakup time.
- B_1 is a parameter regulating speed of breakup, set in the interface ; larger values mean breakup occurs later. The default value is $\sqrt{3}$.
- ρ_l is the density of the liquid.
- ρ_g is the density of the continuous phase.
- r_p is the radius of the particle.
- v_s, slip velocity, is the relative velocity of the droplet through the continuous phase.

When breakup happens, the parent parcel is replaced by a number of child parcels, each containing a number of identical particles. The particle diameters in a given parcel follow a log-normal distribution:

$$\Phi(r) = \frac{1}{2}\left(1 + \mathrm{erf}\left[\frac{\log\left(\frac{r}{r_0}\right) - \langle\, \xi \,\rangle}{\sqrt{2\,\langle\, \xi^2 \,\rangle}}\right]\right) \tag{3107}$$

where:

- $\Phi(r)$ is the breakup distribution function, the probability density as a function of r.
- r is the radius of the child droplet.
- r_0 is the radius of the parent droplet.
- $\langle\, \xi \,\rangle$ is the first moment of $\Phi(r)$, equal to a constant, K_0, set in the interface with default value -0.1. More negative values of K_0 give a smaller mean distribution of child diameters. Positive values of K_0 give larger mean child diameters.
- $\langle\, \xi^2 \,\rangle$ is the second moment of $\Phi(r)$,

$$\langle\, \xi^2 \,\rangle = K_1 \ln\left(\frac{\mathrm{We}}{\mathrm{We}_{\mathrm{cr}}}\right) \tag{3108}$$

  where:

  - K_1 is a constant set in the interface, default value 0.1. K_1 must be positive.
  - We is the Weber number of the particle.
  - $\mathrm{We}_{\mathrm{cr}}$ is the critical Weber number, a constant set in the interface, default value 6.
  - $\langle\, \xi^2 \,\rangle$ is the variance of the distribution, and K_1 is directly proportional to it. The larger they are, the broader the distribution of diameters.

Child parcels receive the velocity of the parent parcel multiplied by a factor, $\left|\mathbf{W}_{\mathrm{bu}}\right|$:

$$\left|\mathbf{W}_{\mathrm{bu}}\right| = K_3 r_{\mathrm{p,\,c}} v = \frac{K_3 r_{\mathrm{p,\,c}}}{\tau_{\mathrm{bu}}} \tag{3109}$$

where:

- K_3 is a constant set in the interface, *Normal Velocity Coefficient*, default value 0.1, limited to values between 0 and 1. The larger K_3 is, the more rapid the radial diffusion of the spray. Use a value of 0 if the spray angle is known and fixed at the injector.
- $r_{\mathrm{p,\,c}}$ is the radius of the child particle.
- $\nu = \frac{1}{\tau_{\mathrm{bu}}}$ is the breakup frequency, the reciprocal of the breakup time.

This model differs from other breakup models in three ways:

- The parent drop no longer exists after the breakup event. (The particle numbering system re-numbers after the breakup, so a particle before the event with the same number as a particle after the event is not the same particle.)
- Because the model works with a user-specified particle count, you seldom have very large particle counts that are difficult to handle numerically.
- The breakup distribution function, Φr in Eqn. (3107), is a function of local conditions and depends only weakly on the diameter of the parent. Though the distribution is limited to the range between the current radius and half the critical radius, the function depends more on the Weber number, which is a strong function of local flow.

Collision and Coalescence

After a collision is detected between two droplets, different collision outcomes are possible: bounce, permanent coalescence, and separation (grazing collision). However, this implementation ignores the formation of satellite droplets which may occur at high collision Weber numbers.

Collision outcomes are described by three non-dimensional parameters:

- the collision Weber number We_{coll}

$$We_{\mathrm{coll}} = \frac{\rho_l \left(v_{i,j}\right)^2 \left(r_1 + r_2\right)}{\sigma} \tag{3110}$$

  The relative velocity $v_{i,\,j}$ is

$$v_{i,\,j} = v_1 - v_2 \tag{3111}$$

  where v_1 and v_2 are the velocities of the colliding drops, and r_1 and r_2 are their corresponding radii.

- the impact parameter B

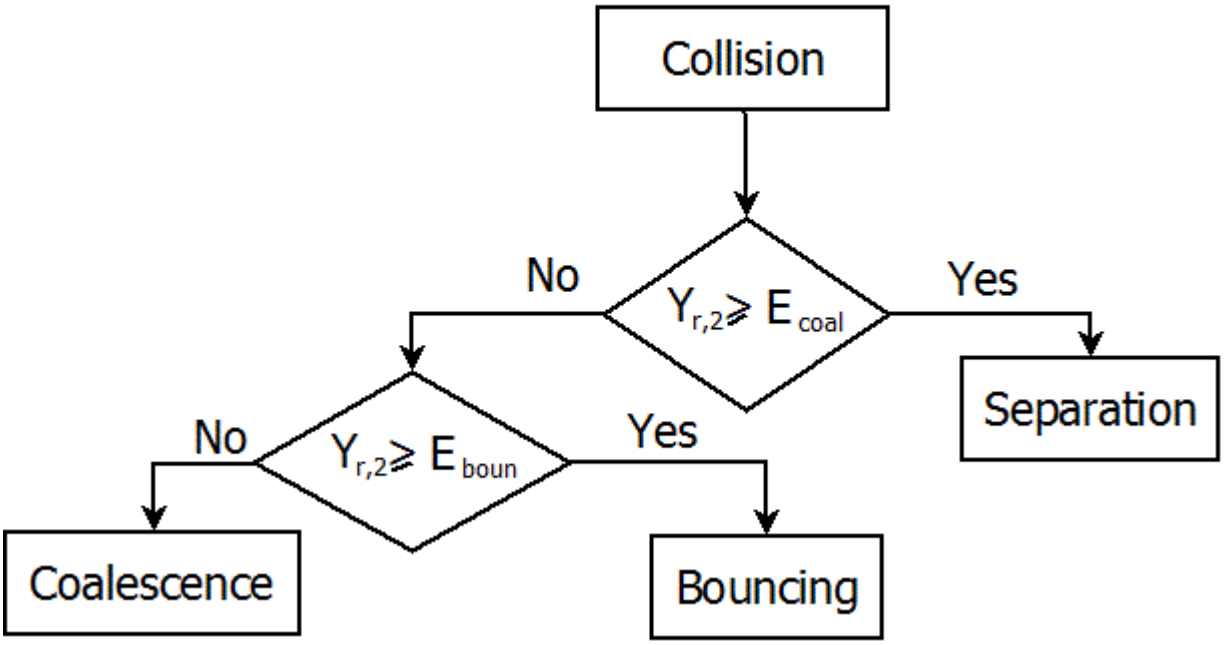
$$B = \frac{b}{r_1 + r_2} \tag{3112}$$

  where b is calculated by taking the distance from the center of one drop to the relative velocity vector that is placed on the center of the other drop. When $B = 0$, the collision is head-on. When $B = 1$, the drops barely graze each other.

- the drop size ratio γ

$$\gamma = \frac{r_2}{r_1} \quad , r_2 > r_1 \tag{3113}$$

The overall collision dynamics algorithm is described in the flowchart [583].



$Y_{r,2}$ is a random number in the range [0,1].

The coalescence efficiency E_{coal} is defined as the probability of coalescence once a collision has occurred. O'Rourke and Bracco [589] employ the approach of Brazier-Smith et al. [575] to obtain

$$E_{\text{coal}} = \min\left[1.0, \frac{2.4f(\gamma)}{We}\right] \tag{3114}$$

where

$$f(\gamma) = \gamma^3 - 2.4\gamma^2 + 2.7\gamma \tag{3115}$$

The bounce efficiency E_{boun} is

$$E_{\text{boun}} = \min\left[1.0, \left(\frac{We}{2.4f(\gamma)}\right)^{1/3}\right] \tag{3116}$$

If coalescence occurs, the number of coalescences m that each droplet undergoes is determined by finding the value of m for which

$$\sum_{n=0}^{m-1} P_n < xx < \sum_{n=0}^{m} P_n \tag{3117}$$

where P_n is the probability of n collisions given by Eqn. (3127).

When coalescence occurs, the number of droplets from the less populous parcel is subtracted from the more populous parcel. If the outcome is grazing collision or bounce, only one collision is calculated per droplet. The post-collision velocity, temperature, and mass of the droplets are calculated based on the conservation of mass, momentum, and energy.

STAR-CCM+ uses a host cell approach to detect collisions. Two parcels can collide only if they are in the same cell. STAR-CCM+ uses the faster of the following collision algorithms to detect collisions in any particular cell:

- the NTC (No Time Counter) collision detection algorithm [606]
- the O'Rourke algorithm [590]

NTC Algorithm

If a cell contains N droplets, the expected number of collisions in the cell over a time interval is given by summing the probability of all possible collisions

$$M_{\text{coll}} = \frac{1}{2}\sum_{i=1}^{N_p} q_i \sum_{j=1}^{N_p} q_j \frac{v_{i,j}\sigma_{i,j}\,\Delta t}{V} \tag{3118}$$

where:

- $v_{i,j}$ is the relative velocity between two colliding parcels
- $\sigma_{i,j}$ is the collision cross section of the two drops defined as

$$\sigma_{i,j} = \pi\left(r_i + r_j\right)^2 \tag{3119}$$

- Δt is the time-step size
- V is the cell volume
- N_p is the number of parcels in a cell
- q_i is the number of droplets in parcel i.

The factor of one-half is a result of symmetry. This summation can be modified by pulling a constant factor outside:

$$M_{\text{coll}} = \frac{(qv\sigma)_{\max}\,\Delta t}{2V}\sum_{i=1}^{N_p} q_i \sum_{j=1}^{N_p} \frac{q_j v_{i,j}\sigma_{i,j}}{(qv\sigma)_{\max}} \tag{3120}$$

The value of $(qv\sigma)_{\max}$ is used for scaling the selection probability of a collision. The chosen value is sufficiently large so that the following restriction holds

$$\frac{q_j v_{i,j}\sigma_{i,j}}{(qv\sigma)_{\max}} < 1 \tag{3121}$$

It is assumed that a representative subsample of parcels may be randomly selected from the set of parcels in the cell. This statistical approximation allows a constant multiplier to reduce the limits of summation as follows:

$$M_{\text{coll}} = \sum_{i=1}^{\sqrt{M_{\text{cand}}}} q_i \sum_{j=1}^{\sqrt{M_{\text{cand}}}} \frac{q_j v_{i,j} \sigma_{i,j}}{(qv\sigma)_{\max}} \tag{3122}$$

where

$$M_{\text{cand}} = \frac{N_p^{\ 2}{}_{(qv\sigma)} \Delta t}{2V} \tag{3123}$$

The overall cost is proportional to the product of the limits of the summation, namely M_{cand}. The value of M_{cand} is linearly proportional to N_p, because q goes as $1/N_p$. The double summation is evaluated using an acceptance-rejection scheme. The number of candidate pairs (M_{cand}) is selected randomly from the cell population.

After a pair has been selected, a random number from a uniform distribution (from 0 to 1) is used to determine if the candidate pair actually collides. A new random number is chosen for every pair, even for pairs within the same cell. A collision takes place between parcels i and j if the deviate r satisfies the inequality

$$r < \frac{q_g v_{i,j} \sigma_{i,j}}{(qv\sigma)_{\max}} \tag{3124}$$

The variable q_g represents the greater number of droplets between q_i and q_j. If the collision is accepted, then q_l, the lesser number of droplets, actually participates in the collision. This distinction is important in the case of droplet coalescence, where one parcel of droplets absorbs the other. Whenever employed for a given cell, the NTC algorithm is linearly proportional to N_p.

If the spray is so dense that $M_{\text{cand}} > N_p^{\ 2}/2$, then direct calculation of collisions using the O'Rourke approach is more efficient than the NTC algorithm for this cell. STAR-CCM+ then switches to the O'Rourke algorithm automatically.

O'Rourke Algorithm

The O'Rourke collision algorithm is a direct technique because it considers all possible collision partners.

The probability of any droplet colliding with any other droplet is given by

$$p_{i,j} = \frac{v_{i,j} \sigma_{i,j} \Delta t}{V} \tag{3125}$$

The mean expected number of collisions between a droplet in parcel i and the droplets in parcel j is given by

$$\overline{\mu} = q_j \frac{v_{i,j} \sigma_{i,j} \Delta t}{V} \tag{3126}$$

The number of collisions is determined by sampling from a Poisson distribution with a mean of $\overline{\mu}$

$$P_n = \frac{\overline{\mu}}{n!} e^{-\overline{\mu}} \tag{3127}$$

Eqn. (3127) gives the probability of n collisions. P_0 gives the probability of zero collisions. Thus, whenever employed for a given cell, the O'Rourke approach is proportional to $N_p^{\ 2}$.

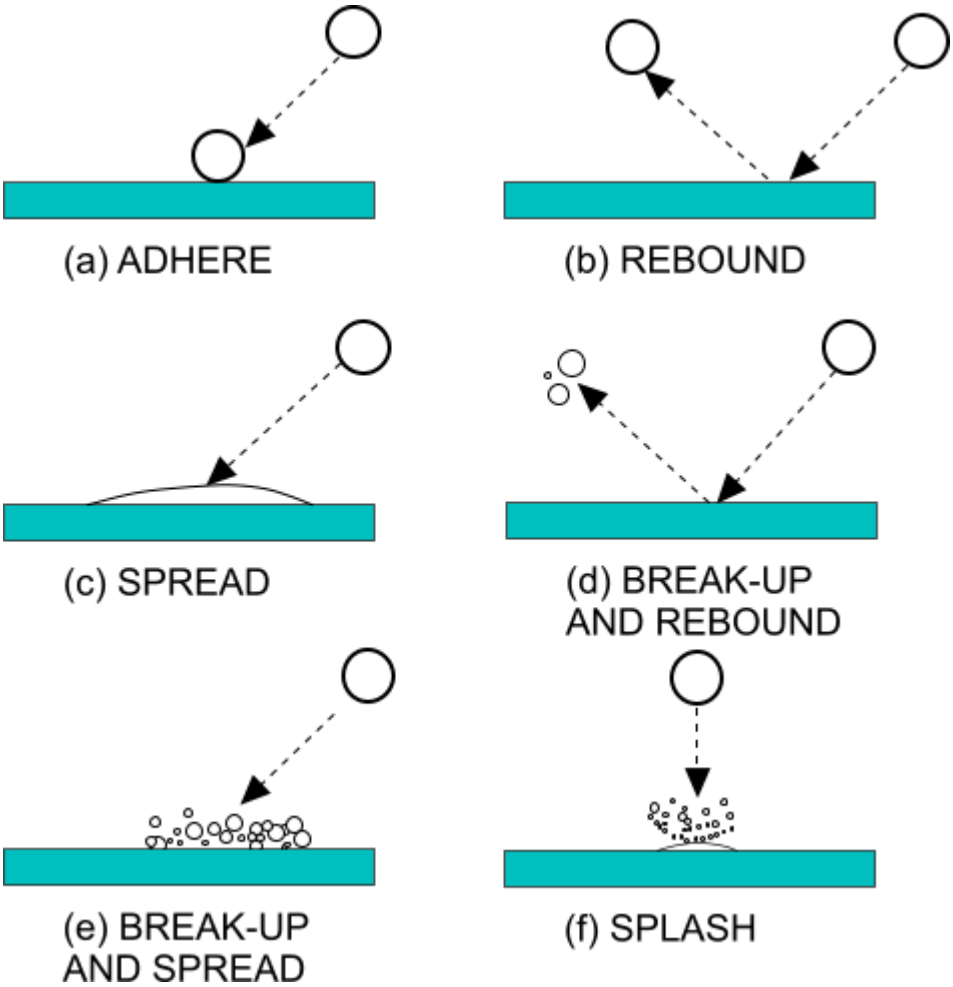
Wall Impingement

The impact of particles on rigid solid surfaces produce a variety of effects. These effects depend on the size, the velocity, the material of the impacting particles and they depend on the nature of the surface. A liquid droplet, for example, can stick, bounce, or splash. The liquid that is deposited on the wall can retain the droplet form or it can merge into a fluid film.

Bai-Gosman Wall Impingement

The Bai-Gosman model aims to predict the outcomes of liquid droplets impacting an impermeable boundary ("wall") or fluid film.

The Bai-Gosman model, originally described in [571], [572], and [573], but extended for Siemens PLM Software, categorizes possible outcomes as being in one of six possible regimes, illustrated in the following diagram.



For fluid film boundaries, momentum and kinetic energy lost by the droplets are gained by the film.

The choice of regime for a given impingement event is made using four parameters:

- The incident Weber number

$$We_I = \frac{\rho_p \mathbf{v}_{r,n}^2 D_p}{\sigma} \tag{3128}$$

in which $\mathbf{v}_{r,n} = \left(\mathbf{v}_p - \mathbf{v}_w\right) \cdot \mathbf{n}_w$ is the normal component of the particle velocity relative to the wall. $\mathbf{n}_w$ is the unit vector normal to the boundary.

- The Laplace number

$$\mathrm{La} = \frac{\rho_p \sigma D_p}{\mu_p^2} \tag{3129}$$

- The boundary temperature T_w (provided the Energy model is active in the physics continuum).
- The wall state; a boundary can be either wet or dry. For fluid film boundaries, this is determined by the presence or absence of a fluid film. For other boundaries, this is determined by the user-selected Wall State property.

  For a dry wall state:

  - At Weber numbers below We_c and wall temperatures below T_{12}, impinging droplets spread out on the wall.
  - At higher Weber numbers or wall temperatures, droplets can spread, break up and spread, break up and rebound, or rebound.
  - Above We_c, impinging droplets splash.

  For a wet wall state:

  - At Weber numbers below 2 and wall temperatures below T_{12}, impinging droplets adhere to the wall.
  - At higher Weber numbers or wall temperatures, droplets can spread, break up and spread, break up and rebound, or rebound.
  - Above We_c, impinging droplets splash.

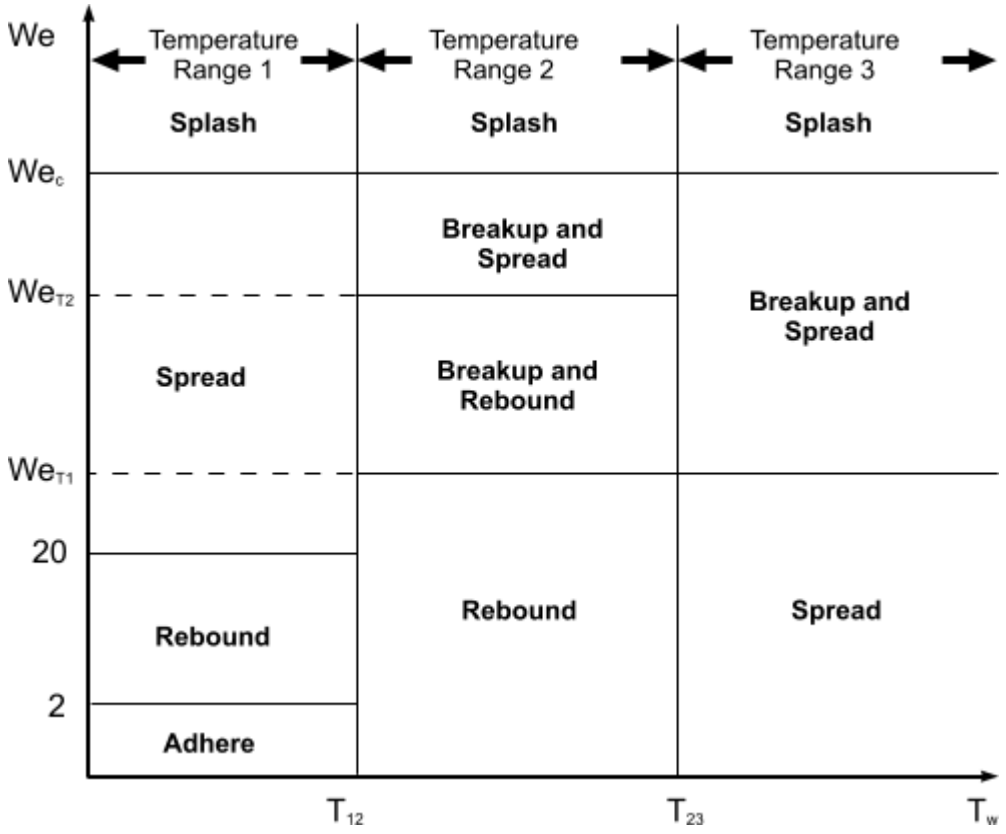
Leidenfrost effect occurs when the temperature is high enough, at T_{12}, reducing friction and heat transfer. It peaks at T_{23}.

Regime Transition Criteria

The idealized map of impingement regimes is shown below, in which there are three temperature ranges, separated by two transition temperatures:

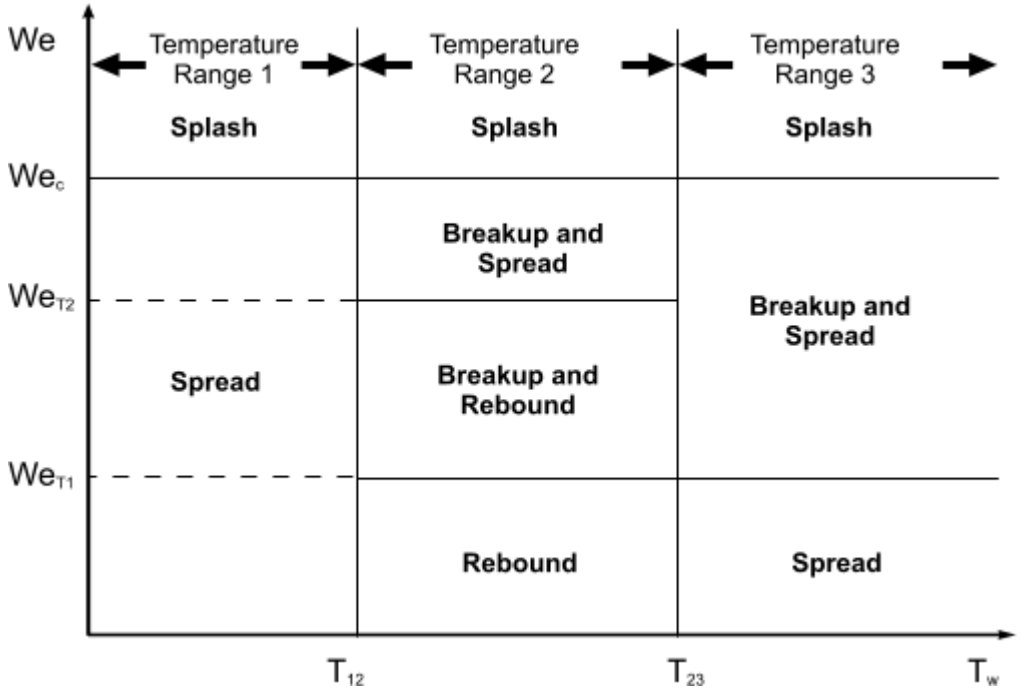
- T_{12}, separating range 1 from range 2, is expected to be approximately the boiling temperature of the droplet.
- T_{23}, separating range 2 from range 3, is expected to be approximately the Leidenfrost temperature of the droplet.

The transition criteria for wetted walls are given below for the three wall temperature ranges.



The Weber numbers 2 and 20, shown in the preceding diagram, can be adjusted in the expert properties of this model.

The transition criteria for dry walls are given below for the three wall temperature ranges.



Temperature Range 1

The first temperature range is expressed as:

$$T_w \leq T_{12} \tag{3130}$$

It also applies when the Energy model is not active in the physics continuum. The model recognizes the following regimes in this temperature range:

- Adhere

$$We_I \leq 2 \tag{3131}$$

in which the impinging droplet adheres to the boundary in nearly spherical form. If the boundary is a fluid film, the droplets transfer their mass, momentum, and kinetic energy to the film.

In STAR-CCM+, it is possible to select the specific boundary interaction mode to control the outcome of a particle satisfying the adhere condition. The default in unsteady is Stick, while in steady the default is Escape.

- Rebound

$$We_I \leq 20 \tag{3132}$$

in which the impinging droplet bounces off the boundary after the impact. This is functionally identical to the Rebound boundary interaction mode with restitution coefficients

$$e_n = 0.993 - 1.76\theta + 1.56\theta^2 - 0.49\theta^3 \tag{3133}$$

$$e_t = \frac{5}{7} \tag{3134}$$

in which θ is the droplet incidence angle that is measured from the boundary.

- Spread

$$We_I \leq We_c \tag{3135}$$

in which the droplet merges with an existing film on a wetted boundary. If the boundary is a fluid film, the droplets transfer their mass, momentum, and kinetic energy to the film. The limiting Weber number We_c is given by

$$\mathrm{We}_c = A\mathrm{La}^{-0.18} \tag{3136}$$

For smooth walls or wet rough walls

$$A = A_w \tag{3137}$$

where A_w is the user-defined splash onset coefficient with a default value, 1320.

For dry, rough walls A is interpolated from the table below

Roughness **µm**	0	0.05	0.14	0.84	3.1	12.0	∞
$\mathbf{A/A_w}$	3.9879	3.9879	3.4348	1.9955	1.5576	1.0015	1

In the current version of STAR-CCM+, the spread regime is implemented as rebound with restitution coefficients

$$e_n = 0 \tag{3138}$$

$$e_t = 1 \tag{3139}$$

- Splash

$$We_I > We_c \tag{3140}$$

in which the impinging droplet breaks up into smaller droplets, some of which are reflected from the boundary. Residue may remain stuck to the wall. Each impinging parcel, of mass m_π, produces a user-specified number of secondary parcels which are randomly reflected from the wall within an ejection cone. The total mass of the secondary parcels, m_s, is determined by the ratio

$$r_m = \frac{m_s}{m_\pi} = 0.2 + c_{ws}X_r \tag{3141}$$

where Xr is a random number that is distributed uniformly in the range [0 — 1]. c_{ws} is a constant, for wet walls

$$c_{ws} = 0.75 \tag{3142}$$

and for dry walls

$$c_{ws} = 0.6 \tag{3143}$$

The total number of secondary droplets that are created per splash is given by

$$N_s = a_0\left(\frac{We_I}{We_m} - 1\right) \tag{3144}$$

with a_0 a user-specified coefficient (default 5) and We_m a Weber number which, in this temperature range, is equal to We_c. The actual number of secondary droplets that are created by a splash event tends to N_s as the number of secondary parcels tends to infinity. This is ensured by randomly selecting the diameters and number of secondary droplets per parcel from a Rosin-Rammler distribution with user-specified exponent q and reference diameter

$$D_{ref} = \left(\frac{r_m}{N_s\Gamma(1 + 3/q)}\right)^{1/3} D_p \tag{3145}$$

in which $\Gamma(x)$ is the gamma function. The default exponent, 1, produces a Chi-squared distribution.

The velocity of the secondary droplets is calculated from two contributions; one due to the normal incident component $\mathbf{v}_n$ and another due to the tangential incident component $\mathbf{v}_t$. They are denoted $\mathbf{v}'_n$ and $\mathbf{v}'_t$ respectively.

The magnitude of $\mathbf{v}'_n$ for each secondary parcel is calculated from the overall conservation of energy as described in [573]. The direction of $\mathbf{v}'_n$ is chosen by randomly selecting an azimuthal angle in the range [0°, 360°] and an ejection angle from the wall in a user-specified range, default [5°, 50°].

If there is insufficient energy to provide the proposed secondary parcels with both surface energy and kinetic energy, the splashing is abandoned. Instead, the impinging droplet simply spreads.

The $\mathbf{v}_t$ component contributes directly to the secondary droplet velocity as

$$\mathbf{v}'_t = c_f \mathbf{v}_t \tag{3146}$$

where cf c_f is the wall friction coefficient which is estimated to be in the range 0.6 to 0.8; its default value is set to 0.7.

Finally, the parent parcel sticks to the wall. Its mass is reduced by a factor $(1-r_m)$, while the number of droplets which it represents remains unchanged.

Temperature Range 2

The second temperature range is expressed as:

$$T_{12} < T_w \leq T_{23} \tag{3147}$$

In this range, there is no contact or deposition due to an intervening vapor film. The regimes are determined by two characteristic Weber numbers We_{T1} and We_{T2} as follows:

- Rebound:

$$\mathrm{We}_I \leq \mathrm{We}_{T1} \tag{3148}$$

- Break-up and Rebound:

$$\mathrm{We}_{T1} < \mathrm{We}_I \leq \mathrm{We}_{T2} \tag{3149}$$

  in which the incident droplets shatter and rebound from the wall. This regime is implemented as a variant of the splash regime in the first temperature range except that no residue is left on the wall ($r_m = 1$) and $\mathrm{We}_m = \mathrm{We}_{T1}$.

- Break-up and Spread:

$$\mathrm{We}_{T2} < \mathrm{We}_I \leq \mathrm{We}_c \tag{3150}$$

  in which the incident droplets shatter and spread along the wall. This regime is implemented as a variant of the splash regime in the first temperature range except that no residue is left on the wall ($r_m = 1$), $\mathrm{We}_m = \mathrm{We}_{T2}$ and the ejection angle is zero.

- Splash:

$$\mathrm{We}_I > \mathrm{We}_c \tag{3151}$$

  which is identical to the splash regime in the first temperature range except that no residue is left on the wall ($r_m = 1$).

Following [571], default values of 30 and 80 for water are used for We_{T1} and We_{T2} respectively.

Temperature Range 3

The third temperature range is expressed as:

$$T_w > T_{23} \tag{3152}$$

.

Again, wall contact is prevented by a vapor film and the droplets are elevated from and move tangentially to the surface. The regimes are as follows

- Spread:

$$\mathrm{We}_I \leq \mathrm{We}_{T1} \tag{3153}$$

- Break-up and Spread:

$$\mathrm{We}_{T1} < \mathrm{We}_I \leq \mathrm{We}_c \tag{3154}$$

  in which the incident droplets shatter and spread along the wall. This regime is implemented as a variant of the splash regime in the first temperature range except that no residue is left on the wall ($r_m = 1$), $\mathrm{We}_m = \mathrm{We}_{T1}$ and the ejection angle is zero.

- Splash:

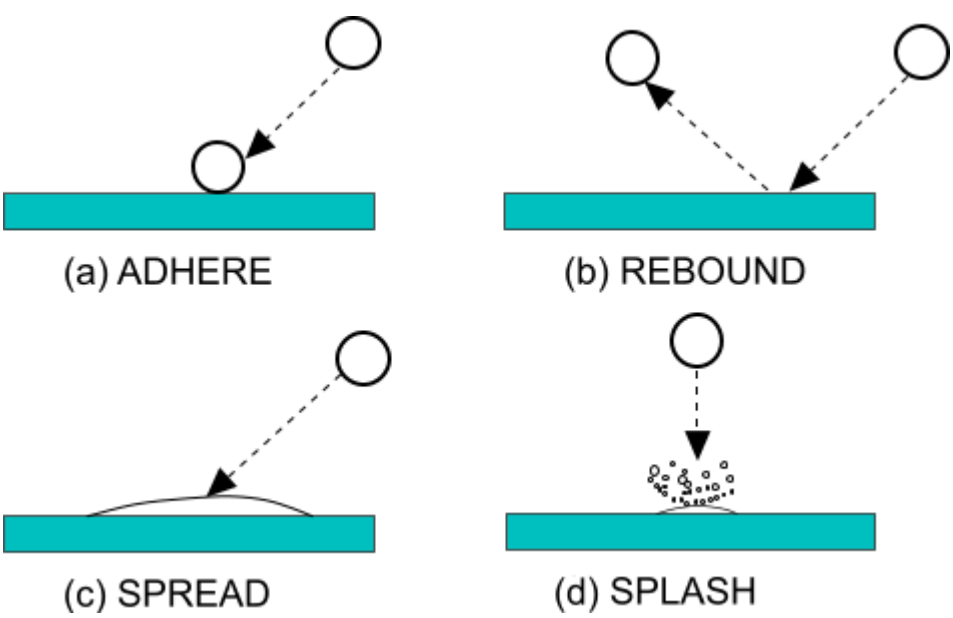
$$\mathrm{We}_I > \mathrm{We}_c \tag{3155}$$

  which is identical to the splash regime in the first temperature range except that no residue is left on the wall ($r_m = 1$).

Bai-ONERA Wall Impingement

The Bai-ONERA model is a development from the Bai-Gosman model. The model is designed to achieve a smooth transition to recover the Bai-Gosman model for situations where the wall temperature is lower than the droplet saturation temperature. It predicts the outcomes of droplets impacting on walls, with special attention to the results when the wall temperature is between the boiling point and the Leidenfrost point for the liquid.

The Bai-ONERA model categorizes possible outcomes as being in one of four possible regimes, illustrated in the following diagram.



For fluid film boundaries (wet walls), the momentum and kinetic energy that are lost by the droplets transfer to the film.

The choice of regime for a given impingement event is made using three parameters:

- A dimensionless temperature T^* defined as:

$$T^* = \frac{T_w - T_s}{T_L - T_s}$$

where:

- T_w is the wall temperature
- T_s is the saturation temperature
- T_L is the Leidenfrost temperature

- The droplet Weber number We that is based on the normal component of relative velocities between the wall and the droplet (see Eqn. (3128)).
- The Ohnesorge number Oh defined as:

$$Oh = \frac{\mu_p}{\sqrt{\rho_p \sigma D_p}} = \frac{1}{\sqrt{La}}$$

where La is the Laplace number (see Eqn. (3129)).

The Weber and Ohnesorge numbers combine into a single dimensionless parameter K defined as

$$K = We \;\; Oh^{-0.4} \tag{3156}$$

Regime Transition Criteria

The maps of impingement regimes, which are shown below, are divided into three temperature ranges, which are separated by two transition temperatures:

- T_{12}, separating range 1 from range 2, is expected to be approximately equal to the saturation temperature T_s.
- T_{23}, separating range 2 from range 3, is expected to be approximately equal to the Leidenfrost temperature T_L.

The transition between splashing and spreading is the function K_s:

$$K_s(T^*) = \begin{cases} K_0 & ; \quad T^* \le 0, \quad T_w \le T_{12} \\ K_0 + T^*(K_1 - K_0) & ; \quad 0 < T^* < 1, \quad T_{12} < T_w < T_{23} \\ K_1 & ; \quad 1 \le T^*, \quad T_{23} \le T_w \end{cases} \tag{3157}$$

The transition between spreading and rebound is the function K_r:

$$K_r(T^*) = \begin{cases} 0 & ; \quad T^* \le 0, \quad T_w \le T_{12} \\ K_1 \cdot (T^*)^{\gamma} & ; \quad 0 < T^* < 1, \quad T_{12} < T_w < T_{23} \\ K_1 & ; \quad 1 \le T^*, \quad T_{23} \le T_w \end{cases} \tag{3158}$$

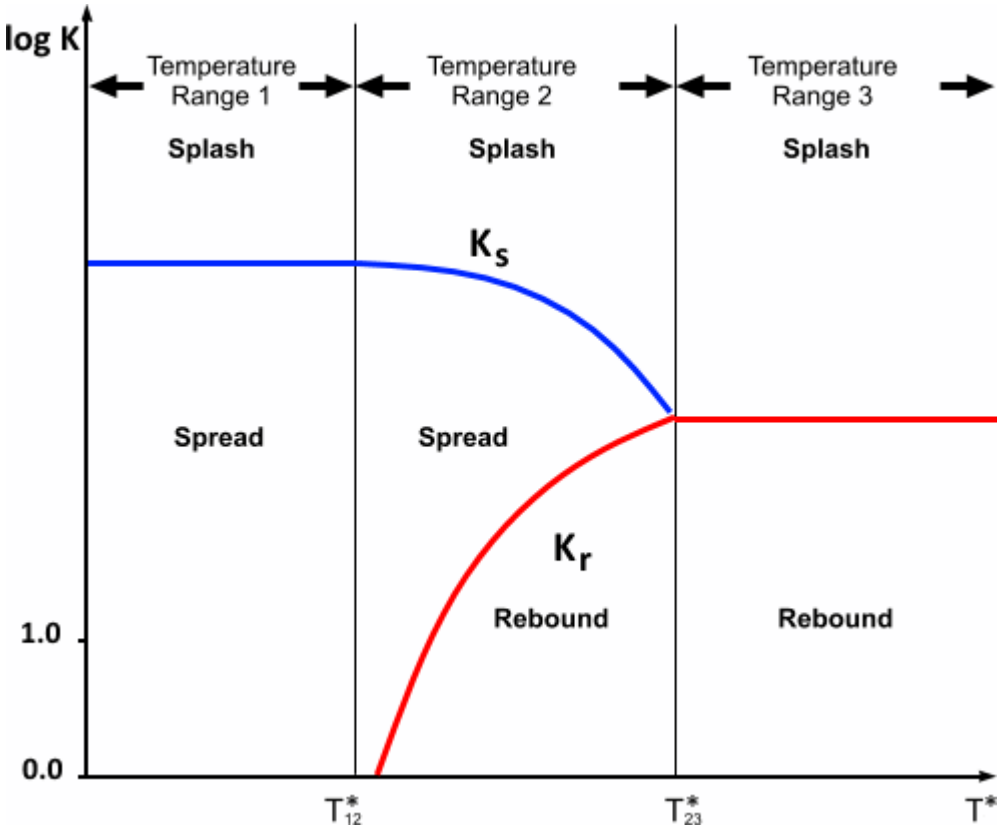
K_0, K_1, and γ are model constants, which you can set in the model properties.

Smooth Dry Wall

For a smooth dry wall:

- $K_0 = 3000.$

- $K_1 = 450.$
- $\gamma = 3.$

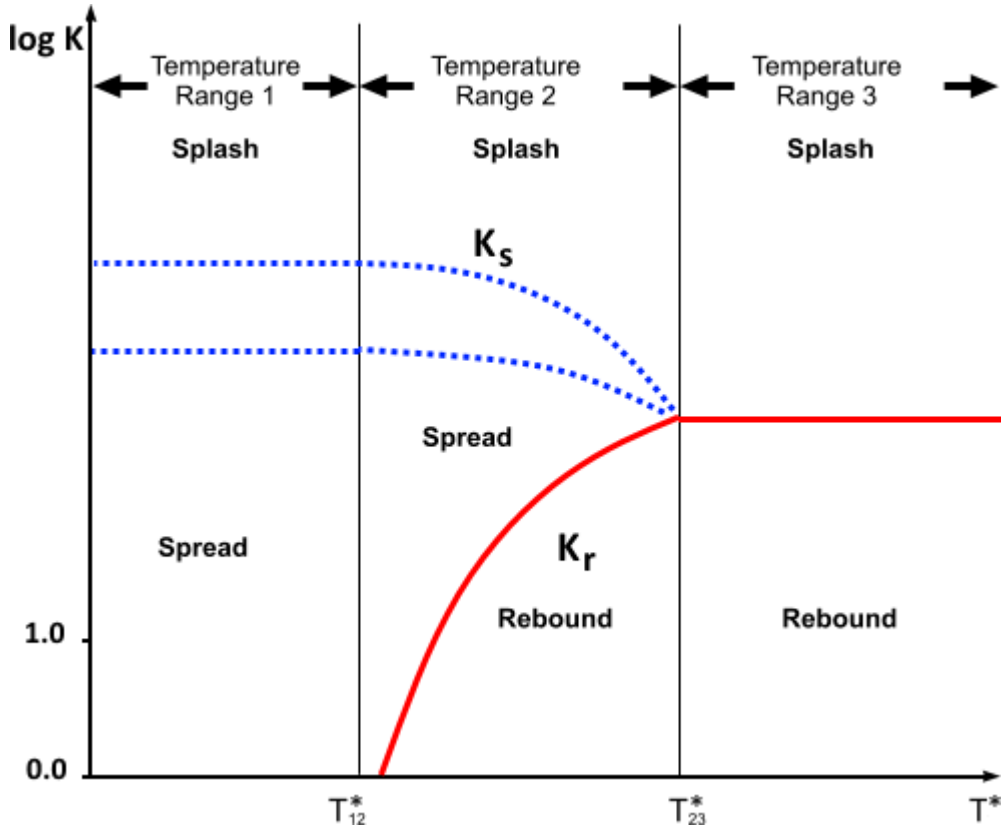


Rough Dry Wall

For a rough dry wall in temperature range 1, $K_0 = We_A \ Oh^{-0.4}$, where $We_A = A \ La^{-0.18}$. La is the Laplace number and the coefficient A is taken from the following table:

Wall roughness	0.05 µm	0.14 µm	0.84 µm	3.1 µm	12.0 µm
Coefficient *A*	5264	4534	2634	2056	1322

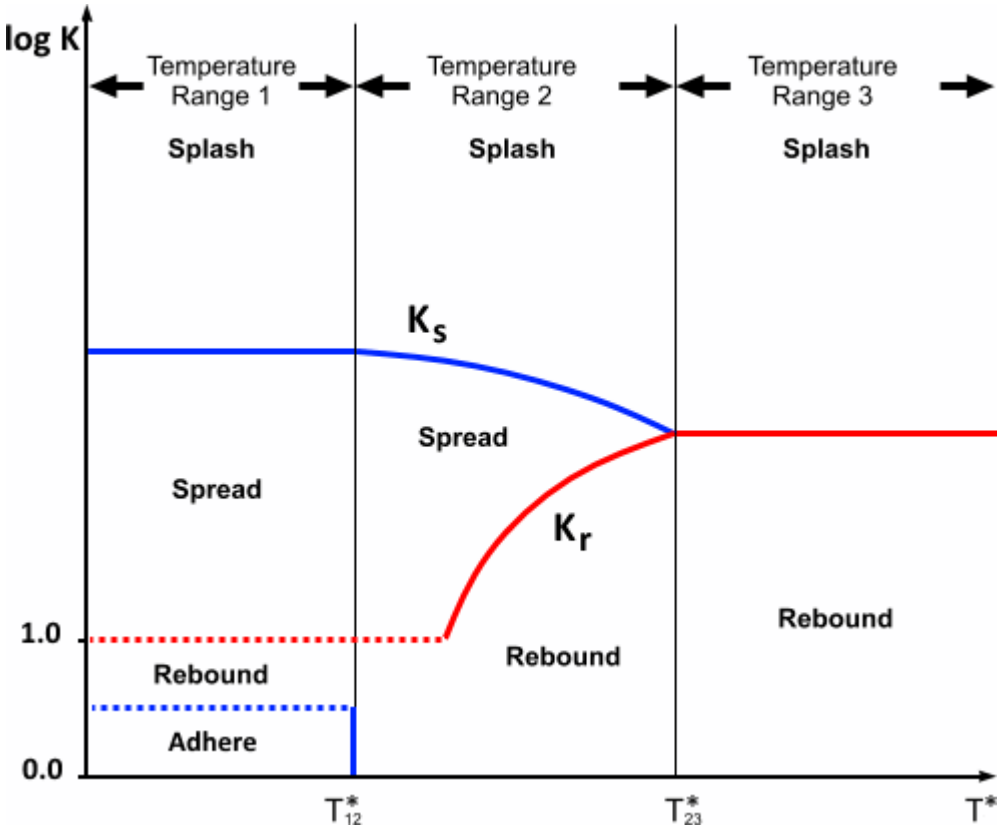
As a result, the transition K_s lies between the two dotted lines in the following figure, the exact position depending on the wall roughness.



Wet Wall

For a wet wall in temperature range 1, $K_r \leq 20.0\ La^{0.20}$ separates the rebound and the spread regimes, which are shown by the dotted line in the following figure. The actual location of the line depends on the Laplace number La of the given droplet. The dotted line extends into temperature range 2 until it merges with the K_r function defined in Eqn. (3158).

In addition, to maintain compatibility with the Bai-Gosman model, the Bai-ONERA model includes an adhere regime, for droplets with low impacting Weber number in temperature range 1.



The Splash Ratio

The splash ratio r_s is the ratio between the splashed and incident droplet masses:

$$r_s = \frac{m_s}{m_i} \tag{3159}$$

where m_s is the splashed mass and m_i is the incident mass.

The droplet mass that is spread on the surface is $m_{spr} = m_i - m_s = (1 - r_s)m_i$.

To adjust r_s to reflect the fact that there is no spreading above the Leidenfrost temperature, introduce a correction factor $f(T^*)$ with the following characteristics:

- $m_{spr} = (1 - r_s)m_i f(T^*)$.
- f is monotonic.
- $f = 1$ at $T^* = 0 \quad (= T^*_{12})$.
- $f = 0$ at $T^* = 1 \quad (= T^*_{23})$.

Define f as:

$$f(T^*) = \begin{cases} 1.0 & ; \quad T^* \le 0, \quad T \le T_{12} \\ 1 - (T^*)^n & ; \quad 0 < T^* < 1, \quad T_{12} < T < T_{23} \\ 0.0 & ; \quad 1 \le T^*, \quad T_{23} \le T \end{cases} \tag{3160}$$

where n is the *Smrf* property of the Bai-ONERA model, controlling the shape of variation.

The amount of splashed mass is then:

$$m_s = m_i\left[1 - \left(1 - r_s\right)f(T^*)\right] \tag{3161}$$

Satoh Wall Impingement Model

The Satoh model aims to predict the outcomes of oil droplets impacting solid boundaries, particularly in oil-mist separators.

In the Satoh model [603], droplets impinging on boundaries are assumed to either rebound or spread depending on their diameter, D and velocity normal to the wall, Vp.

An intermediate region where a combination of rebounding and spreading occurs is defined between two lines, Initial Spread Line (ISL) and Full Spread Line (FSL). These lines both follow power law curves.

- Initial Spread Line

$$Vp_{ISL} = a_{ISL}\left(\frac{D}{D_{ref}}\right)^{n_{ISL}} \tag{3162}$$

- Full Spread Line

$$Vp_{FSL} = a_{FSL}\left(\frac{D}{D_{ref}}\right)^{n_{FSL}} \tag{3163}$$

where D_{ref} is the reference diameter, a is the pre-exponent multiplier and n is the exponent value.

Rebound

Rebound mode is available only at walls, baffles, and contact interfaces. Rebounding particles remain active in the simulation; the mode is distinguished by its treatment of the particle velocity. The rebound velocity relative to the wall velocity is determined by the impingement velocity and user-defined restitution coefficients:

$$\left(\mathbf{v}_p - \mathbf{v}_w\right)^R = e_t\left(\mathbf{v}_p - \mathbf{v}_w\right)_t^I - e_n\left(\mathbf{v}_p - \mathbf{v}_w\right)_n^I \tag{3164}$$

The superscripts R and I denote rebound and impingement respectively; the subscripts n and t denote wall-normal and tangential respectively. Since the left-hand side of Eqn. (3164) can be split into orthogonal n and t components, it can be split into two equations:

$$\left(\mathbf{v}_p - \mathbf{v}_w\right)_t^R = e_t\left(\mathbf{v}_p - \mathbf{v}_w\right)_t^I \tag{3165}$$

$$\left(\mathbf{v}_p - \mathbf{v}_w\right)_n^R = -e_n\left(\mathbf{v}_p - \mathbf{v}_w\right)_n^I \tag{3166}$$

which serve to emphasize the definition of the restitution coefficients as the constants of proportionality between impingement and rebound velocities. Both coefficients may range from 0 to 1. The latter, “perfect” elastic rebound, is the default.

The tangential velocity of a wall boundary is zero unless a value is explicitly prescribed through a wall sliding option. In other words, it is non-zero only at no-slip walls.

Impingement Heat Transfer

The Impingement Heat Transfer model is designed to predict the transfer of heat between a wall and a stream of droplets impinging on it. In a simulation with two-way coupling, the droplets can change the temperature of the wall.

This temperature change can happen even in cases where the wall is above the Leidenfrost temperature. Heat can be transferred between droplets and wall through the gas phase, albeit more slowly than without the Leidenfrost effect. The droplets can then cool the wall below the Leidenfrost point, so that liquid films can form.

The model makes the following assumptions:

- The area through which energy flows from the drop to the wall face is no larger than the area of the wall face.
- For a wall boundary condition where the temperature is fixed (one-way coupling), the particle surface temperature over a dynamically calculated impingement area is fixed. The particle temperature is assumed to be at the droplet centroid a distance r from the wall.
- The distortion of the droplet is neglected and a diffuse spray is assumed.

At high mass loadings (seen with volume fractions larger than 0.3), these assumptions can break down, resulting in odd behavior in the gas phase. This behavior includes, but is not limited to, spurious temperature fluctuations, unphysical particle temperatures, and general difficulty converging.

The heat penetration coefficient for the solid wall, b_w, in equation Eqn. (3172) is multiplied by the HPC Multiplier. This can be used to model a coating or resistance in the wall in transient simulations using constant, field function or table methods.

The energy equation for the droplet or particle is:

$$mc_p \frac{dT_p}{dt} = \dot{Q}_c + \dot{Q}_r + \dot{Q}_l + \dot{Q}_s + \dot{Q}_{wp} \tag{3167}$$

where:

- $\dot{Q}_c$ is the rate of heat transfer from convection.
- $\dot{Q}_r$ is the rate of heat transfer from radiation.
- $\dot{Q}_l$ is the rate of heat transfer from mass transfer.
- $\dot{Q}_s$ is the rate of heat transfer from user-defined heat sources.
- $\dot{Q}_{wp}$ is the rate of heat transfer from wall conduction.

Heat that is transferred from the wall to the drop is described by the Wruck correlation [615]:

$$\dot{Q}_{wp} = A_{cont} \frac{2\sqrt{t_{cont}}}{\sqrt{\pi}\,\Delta\, t_p} \frac{b_w b_p}{b_w + b_p} \left(T_w - T_p\right) \tag{3168}$$

where terms with a w or a p refer to the wall or particle, respectively, and:

- A_{cont} is the effective contact area:

$$A_{cont} = \frac{\pi}{4} {D_{eff}}^2 \tag{3169}$$

where D_{eff} is an effective diameter, time-averaged over the droplet contact time (Akao et al. [566]):

$$D_{eff} = D_0 0.61 We_p^{0.38} \tag{3170}$$

where D_0 is the droplet diameter and We_p is the droplet Weber number.

- t_{cont} is the contact time between particle and wall:

$$t_{cont} = \begin{cases} \frac{\pi}{4}\sqrt{\frac{\rho_p D_0^3}{\sigma_p}} & K < 40 \\ \sqrt{\frac{\pi}{2}}\left(\frac{\rho_p D_0^5}{\sigma_p u_r^2}\right)^{0.25} & K \geq 40 \end{cases} \tag{3171}$$

where:

ρ_p is the particle density.

σ_p is the particle surface tension.

u_r is the velocity of the particle relative to the wall.

K is a dimensionless number proportional to the impact energy. Birkhold et al. [574] define this number as:

$K = Ca^{5/4} La^{3/4}$

where Ca is the Capillary number and La is the Laplace number.

- Δt_p is the particle time-step.
- b_w and b_p are the heat penetration coefficients as evaluated with the materials of the wall or the particle.

$$b_w = c_h\sqrt{\kappa_w \rho_w \left(C_p\right)_w} \ , \quad b_p = \sqrt{\kappa_p \rho_p \left(C_p\right)_p} \tag{3172}$$

where:

  - c_h is the *HPC Multiplier*, default value `1.0`, used to represent coating or other thermal resistance.
  - κ_w and κ_p are the thermal conductivities of the wall and the particle.
  - ρ_w and ρ_p are the densities of the wall and the particle.
  - $\left(C_p\right)_w$ and $\left(C_p\right)_p$ are the specific heats of the wall and the particle.

When there is no conducting solid, the wall temperature is used to calculate the heat flux with the Wruck correlation. Since there is no material specified for the wall, b_w is set equal to b_p, which makes the harmonic average $\frac{1}{2}b_p$.

- T_w and T_p are the temperatures of the wall and the particle.

Eqn. (3168) can be written as:

$$\dot{Q}_{wp} = A_p h_{wp}\left(T_w - T_p\right) \tag{3173}$$

where:

- A_p is the cross-sectional area of the particle.
- h_{wp} is:

$$h_{wp} = \frac{2A_{cont}\sqrt{t_{cont}}\left(b_w b_p\right)}{A_p\sqrt{\pi}\left(b_w + b_p\right)\,\Delta\, t_p} \tag{3174}$$

Film Damping

Film Damping mode allows the motion of solid particles to be damped immediately before and after wall impingement as they pass through a fluid film. This mode is only available for solid particles on fluid film boundaries.

In this situation, the equation of conservation of momentum for a material particle is assumed to reduce to:

$$m_p \frac{d\mathbf{v}_p}{dt} = \mathbf{F}_d + \mathbf{F}_{vm} \tag{3175}$$

where:

- $\mathbf{F}_d$ is the drag force
- $\mathbf{F}_{vm}$ is the virtual mass force

Drag Force

The equation for drag force can be used in its standard form. As the particles are small, spherical, and solid the drag coefficient of the particle, C_d can be calculated using the Schiller-Naumann correlation.

Virtual Mass Force

The equation for virtual mass force is simplified using two assumptions:

- The virtual mass coefficient, C_{vm} is equal to 0.5
- The acceleration of the fluid film is negligible compared to the acceleration at the particle:

$$\left|\frac{D\mathbf{v}}{Dt}\right| \ll \left|\frac{d\mathbf{v}_p}{dt}\right| \tag{3176}$$

Using these assumptions the virtual mass force becomes:

$$\mathbf{F}_{vm} = -0.5\rho V_p \frac{d\mathbf{v}_p}{dt} \tag{3177}$$

Droplet-to-Film Transition Criteria

The Droplet Film Transition model determines whether droplets impinging on a wall form a liquid film on the wall.

The Droplet Film Transition model calculates either the coverage ratio or the equivalent film thickness, depending on which criterion is selected. The coverage ratio criterion is generally more suited to internal combustion simulations with high Weber numbers. The equivalent film thickness criterion is more generally applicable.

Coverage Ratio

The coverage ratio is the fraction of the cell wall that is covered by impinging droplets. On impact, a droplet is taken to assume a cylindrical form with a spreading diameter D_s:

$$D_s = 0.613 D_d We_d^{0.39} \tag{3178}$$

where:

- D_d is the droplet diameter.
- We_d is the Weber number for the droplet.

The area that is covered by the droplets is then A_s:

$$A_s = 0.25 N \pi D_s^2 \tag{3179}$$

where N is the droplet number density. All droplets merge into a liquid film when A_s reaches the value that is given for the *Coverage Ratio* property.

The coverage ratio is then $\gamma_c = A_s / A_c$ where A_c is the area of the cell wall.

Equivalent Film Thickness

The equivalent film thickness is δe:

$$\delta e = \frac{V_d}{A_c} \tag{3180}$$

where V_d is the total volume of all the droplets on the cell wall. All droplets merge into a liquid film when δe reaches the value that is given for the *Equivalent Film Thickness* property.

Particle Porosity

STAR-CCM+ also provides the optional Lagrangian model, Particle Porosity, which allows you to define how the porosity of particles which burn internally changes through the course of a particle reaction.

The porosity, θ, of a particle determines how the diameter and surface area of the particle changes throughout a reaction. When the surface area of a particle changes, the overall reaction rate is affected. Using the Particle Porosity model, you can control how the size of the reacting particle varies by specifying a value for the exponent, α which is related to the diameter and mass of particles by:

$$\frac{d_p}{d_{p\,0}} = \left[\frac{m_p}{m_{p\,0}}\right]^{\alpha/3} \tag{3181}$$

where $0 \leq \alpha \leq 1$

- d_p= particle diameter
- m_p= particle mass
- 0= initial state of the particle

A value for α of 1.0 implies that the particle diameter reduces proportional to the cube root of particle mass. However, a value of 0.0 implies that the particle diameter, and therefore volume, remains the same and particle porosity, θ, increases inversely to particle mass. For coal/char combustion setups, a value of 0.75 works well and for devolatilization reactions, a value of 0.0 works well.

The exponent, α, is used to calculate porosity:

$$\theta = 1 - \left[\frac{m_p}{m_{p0}}\right]^{1-\alpha} \tag{3182}$$

then, particle volume:

$$V_p = \frac{m_p}{(1-\theta)\,\rho_{material}} \tag{3183}$$

and, particle diameter:

$$d_p = \left[\frac{6V_p}{\pi}\right]^{1/3} \tag{3184}$$

where the apparent density is:

$$\rho_{apparent} = (1-\theta)\,\rho_{material} \tag{3185}$$

The diameter of the particle is then used to calculate the surface area of the particle, A_p, which is used by the combustion solver. See Particle Porosity Model.

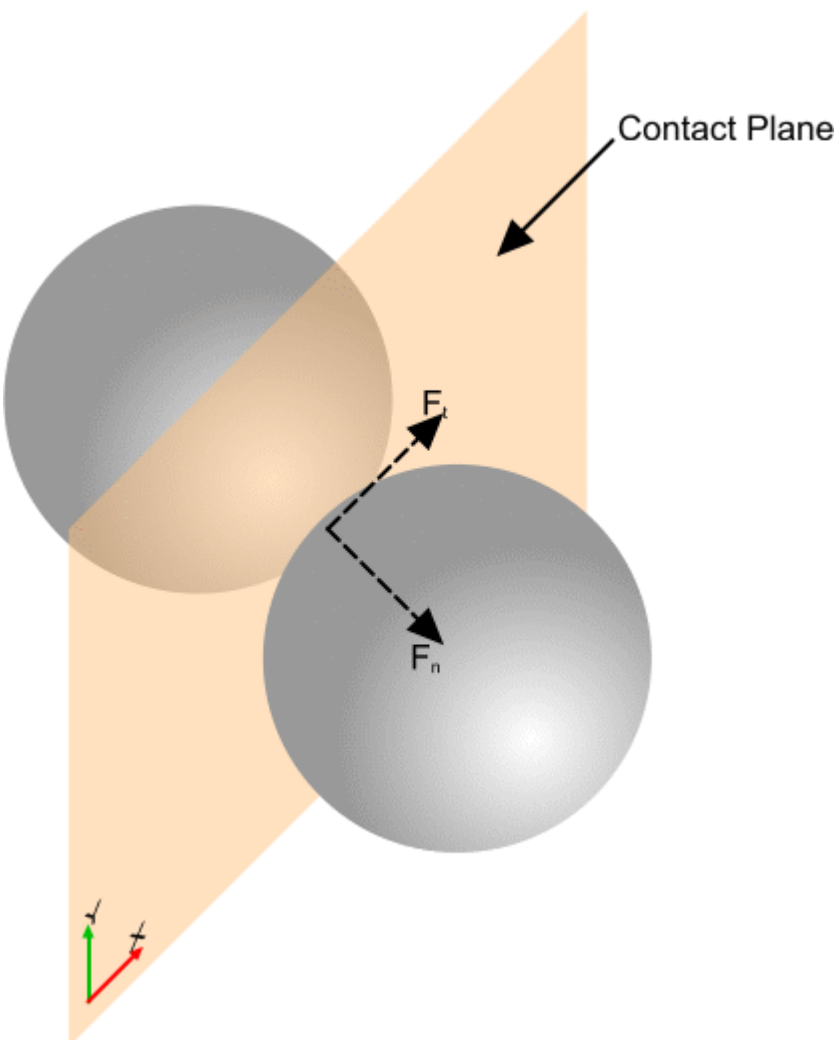
Discrete Element Method

The discrete element method (DEM) is designed for modeling the granular flow of materials such as sand, food particles, powders, capsules, and slurries. These flows are characterized by a high particle density where the inter-particle interaction is of importance.

Established by Cundall and Strack [620], the DEM model is an extension of the Lagrangian modeling methodology to include dense particle flows. The distinct characteristic of DEM is that inter-particle contact forces are included in the equations of motion. These forces cannot be ignored for highly loaded flows, that is, many interacting particles.

The DEM particles can assume different shapes and volumes. To represent a DEM particle, the simplest shape is a sphere. STAR-CCM+ uses multiple sphere shapes to represent non-spherical particles.

STAR-CCM+ uses a classical mechanics method to model DEM, and is based on soft-particle formulation where particles are allowed to develop an overlap. The calculated contact force is proportional to the overlap, as well as to the particle material and geometric properties. This contact force enters the Lagrangian momentum equation Eqn. (2969) as F_c.



To obtain the DEM particle trajectory, Eqn. (2969) is integrated over time using a characteristic DEM particle times scale.

Discrete Element Method Particle Times Scale

The maximum time-step that is allowed for a DEM particle is constrained by the assumption that the force acting on a particle is only affected by the particle's immediate neighbors during a single time-step duration.

The time-step is therefore limited by the time it takes the Rayleigh wave to propagate across the surface of the sphere to the opposite pole [626]:

$$\tau_1 = \pi \frac{R_{min}}{V_{Rayleigh}} \tag{3186}$$

where R_{min} is the minimal sphere radius.

The Rayleigh wave velocity depends on material properties, and the exact value is obtained as a solution to a secular equation [631]. STAR-CCM+ uses an approximation of the solution that provides sufficient precision without incurring high computational cost [628], [629].

Besides the wave propagation, more time-step limiting criteria are applied to moving particles. The duration of impact of two perfectly elastic spheres assuming the Hertz contact theory was derived by Timoshenko [635] to be:

$$\tau_2 = 2.94\left(\frac{5\sqrt{2}\pi\rho}{4}\frac{1-\nu^2}{E}\right)^{\frac{2}{5}}\frac{R}{\sqrt[5]{v_{impact}}} \tag{3187}$$

In order to resolve the collisions a minimum of 10 time-steps is required for collision resolution.

The final restriction on a DEM particle time-step is geometric. It is based on the assumption that particles must not move too far within the time-step. This condition prevents missing contacts between DEM particles, as well as particles and walls. Therefore each particle is constrained such that it takes at least 10 time-steps for the particle to move the full length of the radius.

$$\tau_3 = \left(\frac{R}{v_{particle}}\right) \tag{3188}$$

The final particle time-step is determined as a minimum of τ_1, τ_2, τ_3. In practice τ_1 is typically the limiting factor while τ_2 and τ_3 only constrain particles that are moving fast, or the Young's modulus of the material is set low in order to accelerate the simulation.

Artificial Viscosity

The artificial viscosity model is used to correct the results in cases where high impact velocities cause the numerical models to over-predict the overlap between the colliding objects. The energy that is dissipated by artificial viscosity represents unresolved processes of damage, deformation, and entropy.

The name "Artificial Viscosity" refers only to the similarity between the formulation of the Artificial Viscosity model and the formulations for viscosity. It does not refer to any physical effect resembling real viscosity.

This model introduces a force during compression (while the particles are overlapping and their centers are moving nearer) based on the von Neumann-Richtmyer artificial viscosity [638]:

$$F_v = -f\left(Q_2|v_n| - Q_1 c\right)\frac{m_{eq}}{R_{eq}}v_n \tag{3189}$$

where:

- F_v is the artificial viscosity force.
- v_n is the component of the velocity in the contact normal direction.
- m_{eq} is the equivalent (harmonic mean) mass of the particles.
- R_{eq} is the equivalent (harmonic mean) radius of the particles.
- c is the speed of sound in the material. It is estimated from the equivalent mass and radius, and from the contact stiffness for the specific contact model in use.
- Q_1 is the linear coefficient.
- Q_2 is the quadratic coefficient.

The factor f is a linear ramp function that is scaled to the particle sizes as:

$$f = \begin{cases} 0, & r/R_{eq} < p_{\text{lower}} \\ \dfrac{r/R_{eq} - p_{\text{lower}}}{p_{\text{upper}} - p_{\text{lower}}}, & p_{\text{upper}} \geq r/R_{eq} \geq p_{\text{lower}} \\ 1, & r/R_{eq} > p_{\text{upper}} \end{cases} \tag{3190}$$

where:

- r is the penetration, the overlap distance between particles or between particle and wall.
- p_{upper} and p_{lower} are the upper and lower ramp limits. They are defined in terms of the ratio of the penetration r to the equivalent particle radius R_{eq}.

This ramp function f allows you to use artificial viscosity without affecting typical low penetration contacts.

Contents:

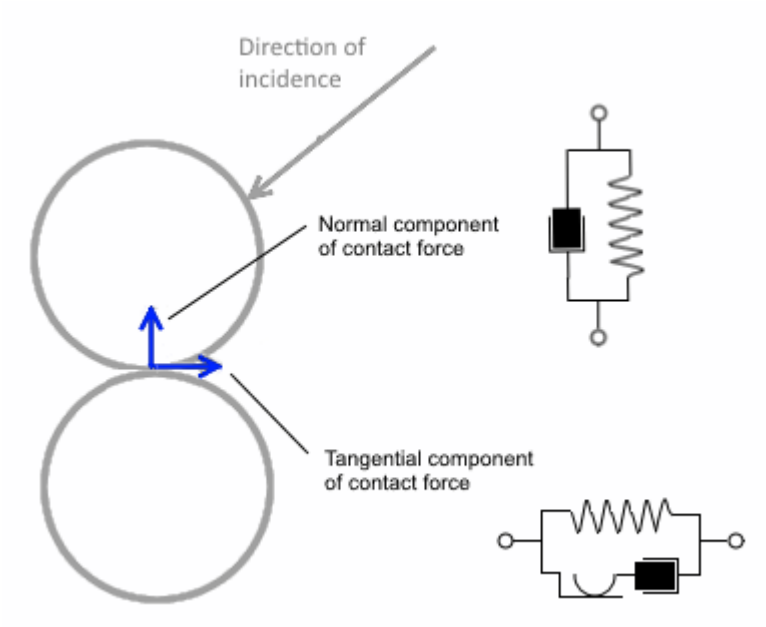
Contact Force
Particle Clustering and Breakup
Heat Transfer

Contact Force

Contact force formulation in DEM is typically a variant of the spring-dashpot model. The spring generates repulsive force pushing particles apart and the dashpot represents viscous damping and allows simulation of collision types other than perfectly elastic.

The forces at the point of contact are modeled as a pair of spring-dashpot oscillators. A parallel linear spring-dashpot model represents the normal force and a parallel linear spring-dashpot in series with a slider

represents the tangential direction of force with, respect to the contact plane normal vector. In both, the spring accounts for the elastic part of the response and the dashpot accounts for energy dissipation during collision.



STAR-CCM+ provides three contact models:

- Hertz Mindlin
- Linear Spring
- Walton Braun

Hertz-Mindlin No-Slip Contact Model

The Hertz-Mindlin contact model is a variant of the non-linear spring-dashpot contact model that is based on the Hertz-Mindlin contact theory [626], [622]. The forces between two spheres, A and B, are described by the following set of equations.

$$\mathbf{F}_{\text{contact}} = \mathbf{F}_n + \mathbf{F}_t \tag{3191}$$

where $\mathbf{F}_n$ is the normal and $\mathbf{F}_t$ is a tangential force component.

The normal direction is defined by the following:

- Normal force:

$$\mathbf{F}_n = -K_n d_n - N_n v_n \tag{3192}$$

- Normal spring stiffness:

$$K_n = \frac{4}{3} E_{eq} \sqrt{d_n R_{eq}} \tag{3193}$$

- Normal damping:

$$N_n = \sqrt{\left(5 K_n M_{eq}\right)} N_{n\ \text{damp}} \tag{3194}$$

where $N_{n\,\mathrm{damp}}$ is the normal damping coefficient as given in Eqn. (3198).

The tangential direction is defined by the following:

- The tangential force is defined by $-K_t d_t - N_t v_t$ if $|K_t d_t| < |K_n d_n| C_{fs}$, where C_{fs} is a static friction coefficient. Otherwise:

$$\mathbf{F}_t = \frac{|K_n d_n| C_{fs} d_t}{|d_t|} \tag{3195}$$

- Tangential spring stiffness:

$$K_t = 8 G_{eq} \sqrt{d_n R_{eq}} \tag{3196}$$

- Tangential damping:

$$N_t = \sqrt{\left(5 K_t M_{eq}\right)} N_{t\,\mathrm{damp}} \tag{3197}$$

where $N_{t\,\mathrm{damp}}$ is the tangential damping coefficient as given in Eqn. (3199).

$$N_{n\ \mathrm{damp}} = \frac{-\ln\left(C_{n\,\mathrm{rest}}\right)}{\sqrt{\pi^2 + \ln\left(C_{n\,\mathrm{rest}}\right)^2}} \tag{3198}$$

$$N_{t\ \mathrm{damp}} = \frac{-\ln\left(C_{t\ \mathrm{rest}}\right)}{\sqrt{\pi^2 + \ln\left(C_{t\ \mathrm{rest}}\right)^2}} \tag{3199}$$

Here, $C_{n\ \mathrm{rest}}$ and $C_{t\ \mathrm{rest}}$ are the normal and tangential coefficients of restitution, model parameters set by the user. If $C_{n\ \mathrm{rest}} = 0$, then $N_{n\,\mathrm{damp}} = 1$, and if $C_{t\ \mathrm{rest}} = 0$, then $N_{t\,\mathrm{damp}} = 1$.

The equivalent radius is defined as:

$$R_{eq} = \frac{1}{\frac{1}{R_A} + \frac{1}{R_B}} \tag{3200}$$

The equivalent particle mass is:

$$M_{eq} = \frac{1}{\frac{1}{M_A} + \frac{1}{M_B}} \tag{3201}$$

The equivalent Young's modulus is expressed as:

$$E_{eq} = \frac{1}{\frac{1-\nu_A^2}{E_A} + \frac{1-\nu_B^2}{E_B}} \tag{3202}$$

The equivalent shear modulus is:

$$G_{eq} = \frac{1}{\frac{2\left(2-\nu_A\right)\left(1+\nu_A\right)}{E_A} + \frac{2\left(2-\nu_B\right)\left(1+\nu_B\right)}{E_B}} \tag{3203}$$

where:

- M_A and M_B are masses of sphere A and sphere B.
- d_n and d_t are overlaps in the normal and tangential directions at the contact point.
- R_A and R_B are the radii of the spheres.
- E_A and E_B are the Young's modulus of the spheres.
- ν_A and ν_B are the Poisson's ratios.
- v_n and v_t are the normal and tangential velocity components of the relative sphere surface velocity at the contact point.

For particle-wall collisions, the formulas stay the same, but the wall radius and mass are assumed to be $R_{wall} = \infty$ and $M_{wall} = \infty$, so the equivalent radius is reduced to $R_{eq} = R_{particle}$ and equivalent mass to $M_{wall} = M_{particle}$.

Di Renzo and Di Maio [622] proposed several formulations for detailed treatment of tangential micro-slip and tangential force computation. STAR-CCM+ uses the formulation that Tsuji proposed [637], where the tangential force is assumed to be non-linear but the detail micro-slip tracking is replaced by analytical expression. The resulting code is computationally efficient while still matching experimental data.

Linear Spring Contact Model

This model is based on the work of Cundall and Strack [620]. The normal and tangential forces $\mathbf{F}_n$ and $\mathbf{F}_t$ are defined as they are in Eqn. (3192) and Eqn. (3195), but parameters K_n, K_t, N_n, and N_t have the following definitions:

- K_n is the normal spring constant.
- K_t is the tangential spring constant.
- $N_n = 2N_{n\ \mathrm{damp}}\sqrt{K_n M_{eq}}$ where:
  - $N_{n\ \mathrm{damp}}$ is the normal damping coefficient as in Eqn. (3198).
  - M_{eq} is the equivalent particle mass as in Eqn. (3201).
- $N_t = 2N_{t\ \mathrm{damp}}\sqrt{K_t M_{eq}}$ where $N_{t\ \mathrm{damp}}$ is the tangential damping coefficient as in Eqn. (3199).

When the Linear Spring model uses the particle material based method, the spring stiffness is estimated based on the assumption that the linear model is a linearized Hertz model, in which the maximum normal contact force is:

$$K_n \delta_{max} = \frac{4}{3} E_{eq} \sqrt{R_{eq} \delta_{max}} \delta_{max} \tag{3204}$$

where:

- δ_{max} is the maximum overlap.
- E_{eq} is the equivalent Young's modulus defined in Eqn. (3202).
- R_{eq} is the equivalent radius defined in Eqn. (3200).

To calculate the spring constants, let:

$$\lambda_{max} = \frac{\delta_{max}}{R_{eq}} \tag{3205}$$

This gives the normal spring constant K_n as:

$$K_n = \frac{4}{3}\sqrt{\lambda_{max}} E_{eq} R_{eq} \tag{3206}$$

The tangential spring constant K_t is:

$$K_t = 8\sqrt{\lambda_{max}} G_{eq} R_{eq} \tag{3207}$$

where G_{eq} is the equivalent shear modulus defined in Eqn. (3203).

Walton Braun Hysteretic Contact Model

The Walton Braun hysteretic plastic-elastic contact system is characterized by the following relations:

$$\mathbf{F}_n = -K_1 \delta \mathbf{n} \tag{3208}$$

when the contact is loading, and:

$$\mathbf{F}_n = -K_2 \left(\delta - \delta_{\text{deformation}}\right) \mathbf{n} \tag{3209}$$

during unloading, where:

$$K_1 = 1.6 \pi R Y_0 \tag{3210}$$

$$K_2 = \frac{1}{E_f} K_1 \tag{3211}$$

$$Y_0 = E \cdot \text{YieldStressFraction} \tag{3212}$$

$$\delta_{\text{deformation}} = \delta_{\max}\left(1 - E_f\right) \tag{3213}$$

E is the material Young modulus, δ is the contact normal overlap, $\mathbf{n}$ is the contact normal vector, and Yield Stress Fraction is a user-controllable model property defining the onset of plastic deformation. E_f is a user-controllable energy fraction that defines the amount of energy that is recovered during the unloading and $\delta_{\max}$ is the maximum overlap that is reached during the contact loading.

Contact Cycle

When particle contact is formed, the state of the contact is tracked at each time-step as follows:

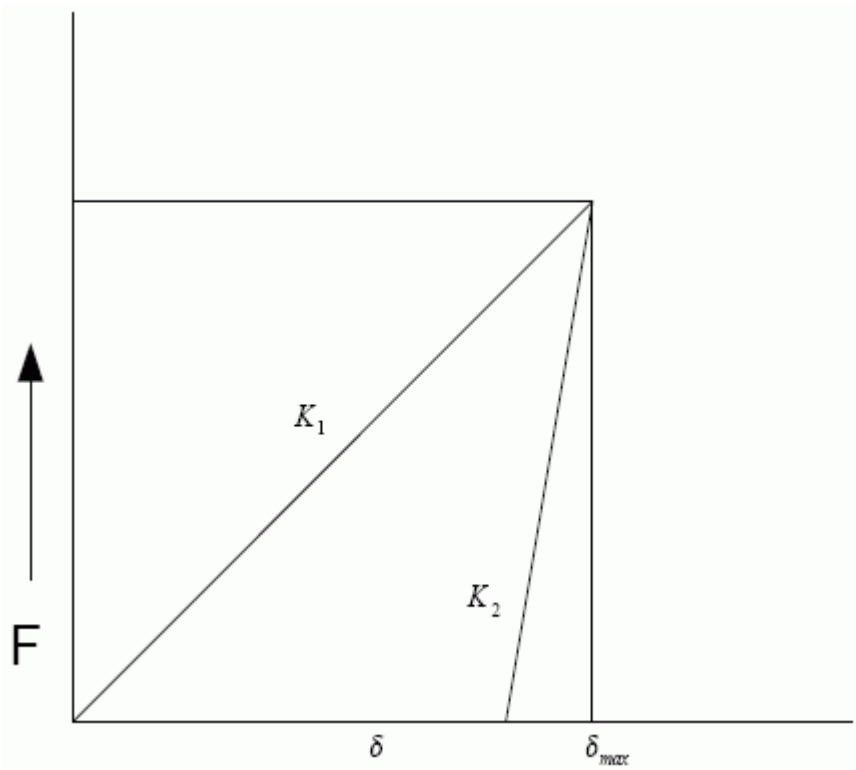
- Contact is considered loading when $\mathbf{v}_r \cdot \mathbf{n} \geq 0$ and $\delta > \delta_{\max}$, where $\mathbf{v}_r$ is relative surface velocity.

  When contact is loading, the force is calculated using K_1 stiffness and $\delta_{\max}$ is updated to a value of δ.
- If the contact is not loading, it is either unloading ($\mathbf{v}_r \cdot \mathbf{n} < 0$) or reloading ($\delta \leq \delta_{\max}$). During unloading, the K_2 stiffness is used to compute the normal force and the value of $\delta_{\max}$ is not updated.

- If the particles separate completely, the contact tracking information is removed and subsequent contact is treated as new.

Geometric Interpretation

The loading/unloading cycle can be geometrically represented as follows:



Contents:

Linear Cohesion
Rolling Resistance

Linear Cohesion

Cohesion modeling facilitates simulation of inter-molecular attraction forces (Van der Waals forces) between particle surfaces. For some classes of simulations, such as dry powders, these forces significantly affect the outcome and cannot be ignored.

This variant of cohesion modeling can use either of two models that are described in published literature: the Johnson-Kendall-Roberts (JKR) model [626] or the Derjaguin-Muller-Toporov (DMT) model [617].

Cohesion force between two spherical particles is expressed in both cases as:

$$F_{cohesion} = R_{min} W \pi F \tag{3214}$$

where R_{min} is the minimal radius of surfaces in contact, W is the work of cohesion $\left[J/m^2\right]$, and F is a multiplication model blending factor with values as follows:

- $\frac{3}{2}$ for the JKR model
- 2 for the DMT model

The principal difference between JKR and DMT model formulation is the surface area where the cohesion force is acting. In the JKR model, this area is limited to direct contact, while in the DMT model it also incorporates the “neck” area in immediate contact proximity.

Rolling Resistance

The following rolling resistance methods are applicable to DEM particles:

- Force Proportional Method
- Constant Torque Method
- Displacement Damping Method

If you are using the Hertz-Mindlin, Walton-Braun, or Bonded Particle contact models together with the Rolling Resistance model, the Rolling Resistance model applies a rolling friction coefficient to the contact models.

Force Proportional Method

This method only supplies a coefficient of rolling resistance, μ_r, also called the coefficient of rolling friction. It can take on any positive real value. The default value is 0.001.

Constant Torque Method

The Constant Torque method applies a torque to resist rolling, calculated as:

$$M_r = \frac{-\omega_{rel}}{|\omega_{rel}|}\mu_r R_{eq} F_n \tag{3215}$$

where:

- ω_{rel} is the relative rotation, given as $\omega_{rel} = \omega_i - \omega_j$, where:
  - ω_i is the rotation of the particle for which the resistance moment M_r is being calculated.
  - ω_j is the rotation of the other particle in the contact.
- μ_r is the coefficient of rolling friction, also called coefficient of rolling resistance. The default value is 0.1.
- R_{eq} is the particle equivalent radius, that is, the harmonic mean of the radii of the two particles.
- F_n is the magnitude of the contact force.

This method also uses a “Moment Limiter.” This function limits the size of the angular momentum of the particle, measured around the axis of moment. This maximum is 0.08 of the moment that negates the lesser of the two rotations ω_i and ω_j in the next time-step.

This limit can control specious rotations in particle compacts. These rotations can arise from a large number of contacts or from high compact pressure.

Displacement Damping Method

The Displacement Damping method combines a viscous damping resistance with a spring-like resisting torque generated by elastic energy stored in the particle as it tries to turn around a finite contact area.

The spring resistance torque at time t is:

$$M^k{}_{r,t+\delta t} = M^k{}_{r,t} - k_r \omega_{rel} \Delta t \tag{3216}$$

where $\left|M^k{}_{r,t}\right| \leq k_r \theta_r{}^{max}$ and $k_r = \left(\mu_r R_{eq} F_n\right) / \theta_r{}^{max}$, and where:

- $\theta_r{}^{max}$ is the maximum angular displacement before rolling.
- ω_{rel} is the relative rotation, given as $\omega_{rel} = \omega_i - \omega_j$, where:
  - ω_i is the rotation of the particle for which the resistance moment M_r is being calculated.
  - ω_j is the rotation of the other particle in the contact.
- μ_r is the coefficient of rolling friction, also called coefficient of rolling resistance. The default value is 0.1.
- R_{eq} is the particle equivalent radius, that is, the harmonic mean of the radii of the two particles.
- F_n is the magnitude of the contact force.

You can activate the viscous damping torque or leave it off. When active, it is:

$$M_r{}^d = \begin{cases} -\nu C^{\text{crit}} \omega_{\text{rel}} & \text{if } \left(\left|M^k{}_{r,t}\right| \leq k_r \theta_r{}^{max}\right) \\ -f \nu C^{\text{crit}} \omega_{\text{rel}} & \text{if } \left(\left|M^k{}_{r,t}\right| > k_r \theta_r{}^{max}\right) \end{cases} \tag{3217}$$

where:

- ν is the damping factor parameter.
- f is a damping factor multiplier
- C^{crit} is the critical viscous damping constant, computed as:

  $C_r{}^{\text{crit}} = 2\sqrt{I_r k_r}$

  where I_r is the equivalent moment of inertia for the relative rotational vibration mode about the contact point.

Particle Clustering and Breakup

Particle bonding is used to extend the particle model to more complex (compliant) shapes or structures. You use the particle clumps model to create groups of particles that represent a complex shape. The particle clumps model invokes the bonded particles model where the individual particles retain their separate identities and are linked by elastically compliant bonds. The addition of a damage model, allows the fracture and fragmentation of the clump with the constituent particles separating from the clump.

The composite particle model, on the other hand, treats the ensemble as a single rigid body that does not separate during the simulation.

Parallel bonds use the same mechanism as the clumped particles, however, the bonds are formed in an interval during the simulation as particles collide. This forming of bonds can be part of an initialization phase where, for example, particles are packed into a volume and an initial configuration is formed. The mechanism is also used to represent a process, for example, a tableting process during the simulation. The result is to form a compliant structure with damage as an option.

Particle Bonding Model

The particle bonding model is used by both the parallel bonds model and the bonded particle (clumps) model. The parallel bonds model uses particle bonding to describe interactions between the constituent particles, but also models elastic collisions between particles, based on the elastic contact models chosen by the user. The bonded particle (clumps) model uses only the particle bonding model.

The particle bonding model simulates cumulative damage to bonds. When an interaction producing some damage but falls short of failure, the amount of that damage is reflected in the reduced strength of the bond.

The parallel bond forces are represented in the following governing equations [630].

Force and torque on a particle due to parallel bonds is:

$$\mathbf{F}_i = F_n \mathbf{n}_i + F_s \mathbf{t}_i \tag{3218}$$

$$\mathbf{M}_i = M_n \mathbf{n}_i + M_s \mathbf{t}_i \tag{3219}$$

where the terms F_n, M_n denote the normal and F_s, M_s denote the shear components of forces and torques with respect to the contact plane, and:

$$\Delta F_n = k_n A \Delta U_n \tag{3220}$$

$$\Delta F_s = -k_s A \Delta U_s \tag{3221}$$

$$\Delta M_n = -k_s J \Delta \Omega_n \tag{3222}$$

$$\Delta M_s = -k_n I \Delta \Omega_s \tag{3223}$$

where:

$$A = \pi R^2 \tag{3224}$$

$$I = \frac{1}{4}\pi R^4 \tag{3225}$$

$$J = \frac{1}{2}\pi R^4 \tag{3226}$$

$$k_n = \frac{E}{R_a + R_b} \tag{3227}$$

$$k_s = 6\left(\frac{G}{R_a + R_b}\right) \tag{3228}$$

$$\Delta U = V_{\text{relative}} \Delta t \tag{3229}$$

where ΔU_n is the normal component of ΔU, and ΔU_s is the tangential component of ΔU.

$$\Delta \Omega = \omega_{\text{relative}} \Delta t \tag{3230}$$

where $\Delta \Omega_n$ is the normal component of $\Delta \Omega$, and $\Delta \Omega_s$ is the tangential component of $\Delta \Omega$.

The parallel bond can be treated as brittle and disintegrative when maximum shear and tensile stresses exceed prescribed limit values. The stress limit values are calculated based on beam theory to be:

$$\delta_{\max} = \frac{-F_n}{A} + \frac{|M_s| R}{I} \tag{3231}$$

$$\sigma_{\max} = \frac{|F_s|}{A} + \frac{|M_n| R}{J} \tag{3232}$$

where:

$\delta_{\max}$	maximum tensile stress that is calculated in the bond
$\sigma_{\max}$	maximum shear stress that is calculated in the bond

If either value exceeds the predefined maximum, the bond is removed and the forces are further calculated only based on the granular interaction of particles.

Simple Failure Model

The simple failure model breaks a bond in the simulation if tensile stress or shear stress on the bond exceed one of the user-specified maximum values δ_m and σ_m.

The bond fails, if:

$$\delta_{\max} > \delta_m \quad \text{or} \quad \sigma_{\max} > \sigma_m$$

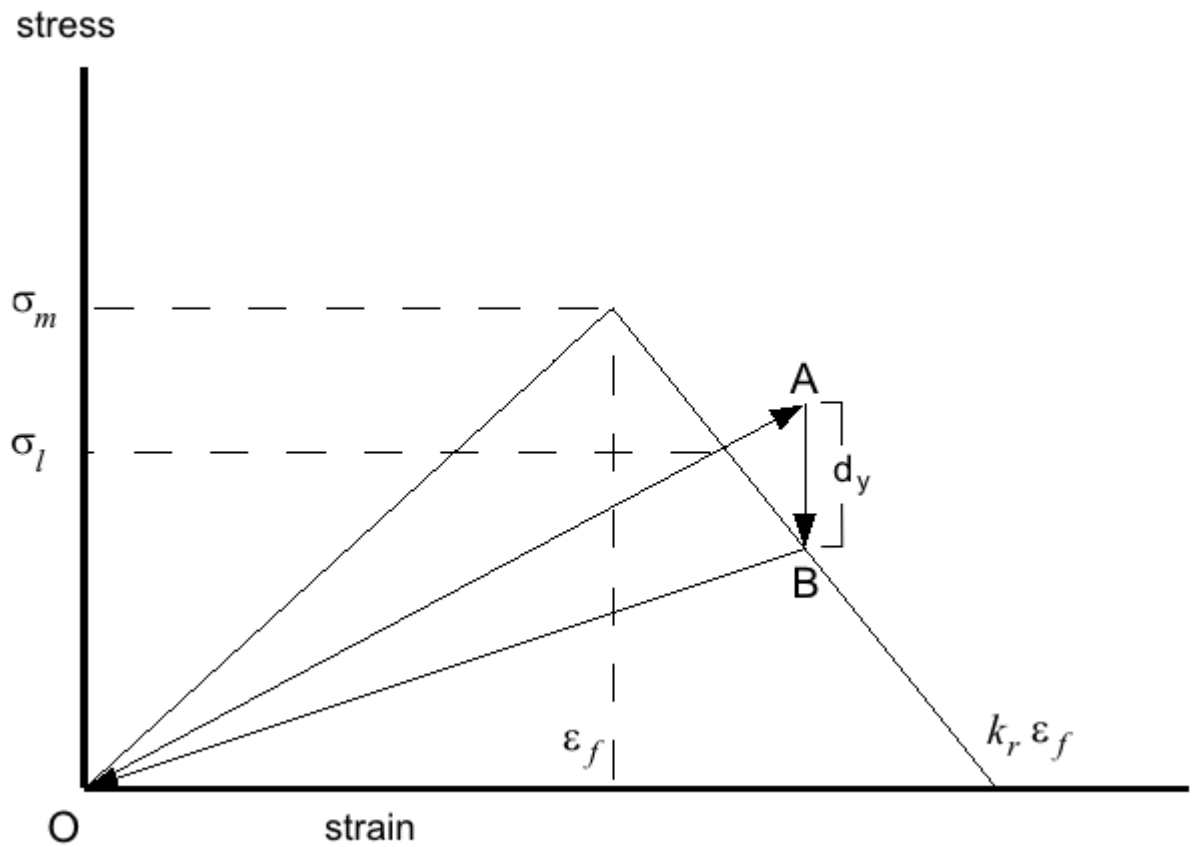
where:

δ_m	maximum tensile stress that is allowed without failure of the bond
σ_m	maximum shear stress that is allowed without failure of the bond

Constant Rate Damage Model

The constant rate damage model simulates the weakening and breaking of bonds between particles in a clump, resulting from collisions.

The constant rate damage model follows the stress-strain curve shown in the following figure:



The point (ε_f, σ_m) is the point of strain ε_f at failure and maximum stress σ_m induced by that strain. Once conditions reach the point (ε_f, σ_m), the model calculates the damage required to return the stress to the failure surface, defined by the line from (ε_f, σ_m) to ($k_r\,\varepsilon_f$, 0), where k_r is the fracture softening modulus, a model parameter with default value 2 at constant strain. For stability, k_r should be greater than 1. [627]

If d_i is the damage at time-step i, the damage at time-step $i+1$ is:

$$d_{i+1} = \min\left(d_y,\ d_v,\ 1\right) \tag{3233}$$

where:

d_v is the amount of damage produced by the fastest possible speed of crack growth:

$$d_v = d_i + 2\sqrt{d_i}\frac{C_r}{r_b}\Delta t + \left(\frac{C_r}{r_b}\right)^2 \Delta t^2 \tag{3234}$$

- C_r is the Rayleigh wave speed.
- r_b is the bond radius.
- Δt is the size of the time-step.

$$d_y = \frac{k_r}{k_r - 1}\left(1 - \left(1 - d_i\right)\frac{\sigma_m}{\sigma_o}\right) \tag{3235}$$

where:

k_r	fracture softening modulus, a model parameter, default value 2
σ_m	maximum shear stress that is allowed without failure of the bond, determined by the **Tensile Strength Method** property of the Constant Rate Damage model
σ_o	maximum tensile stress for the bond [618], [636]

The maximum tensile stress for the bond is calculated as:

$$\sigma_o = \sqrt{\max\left(0,\ \delta_{\max}\right)^2 + \sigma_{\max}} \tag{3236}$$

where:

$\delta_{\max}$	maximum tensile stress that is calculated in the bond
$\sigma_{\max}$	maximum shear stress that is calculated in the bond

If $\sigma_o > \sigma_l$, damage is evolving:

$$\sigma_l = \sigma_m \frac{k_r\left(1 - d_i\right)}{\left(k_r - 1\right)\left(1 - d_i\right) + 1} \tag{3237}$$

As a bond is damaged, its stiffness is modified. When the bond is in tension, the stiffness of the bond, which is represented by the parameters k_n and k_s in Eqn. (3221) to Eqn. (3223), is reduced by a multiplicative factor. This multiplicative factor is $\left(1 - d_i\right)$. The factor represents the reduced area that remains intact in the damaged bond.

When the bond is in compression, no reduction of stiffness occurs, because the direct contact of a fracture surface transmits the normal load. The resultant friction together with surface asperities work to transmit the shear force.

Heat Transfer

When particles make contact with other particles or with the wall, heat is transferred through conduction. The rate of heat transfer is determined by the temperature difference, the effective contact area and the thermal conductivity of the materials.

Conduction

Two bodies are assumed to exchange heat through conduction when they make physical contact. The two bodies can be two particles or a particle and a wall.

There are four cases:

- Particle-particle (see [619]):

  The heat exchange q_{ij} from particle i to particle j is:

$$q_{ij} = 4r_c k\left(T_j - T_i\right) \tag{3238}$$

  where:

  - r_c is the contact area radius.
  - k is the equivalent thermal conductivity of the two particles, where

    $\frac{1}{k} = \frac{1}{k_i} + \frac{1}{k_j}$

    where k_i and k_j are the thermal conductivities for particles i and j.
  - T_i and T_j are the temperatures of particle i and particle j.
- Particle-wall, adiabatic wall: heat exchange is zero.
- Particle-wall, fixed wall temperature:

  The heat exchange q_{iw} from the wall to particle i is:

$$q_{iw} = k\left(T_w - T_i\right) \tag{3239}$$

  where:

  - k is the equivalent thermal conductivity, equal to k_i.
  - T_i and T_w are the temperatures of particle i and the wall.
- Particle-wall, fixed wall heat flux:

  The heat exchange q_{iw} from the wall to particle i is:

$$q_{iw} = A_c q_w \tag{3240}$$

  where:

  - A_c is the contact area between the particle and the wall.
  - q_w is the heat flux density of the wall.

If the simulation includes a continuous phase, its energy state is unaffected by particle-particle and particle-wall heat conduction unless you activate the Two-Way Coupling model. With this model active, the Lagrangian energy source term is added to the energy equation of the continuous phase.

The linearized wall heat flux, q_w, is expressed as:

$$q_w = a + bT_c + cT_w + dT_w{}^4 \tag{3241}$$

where:

- T_c and T_w are the cell and wall temperatures, respectively.
- a, b, c, and d are wall heat flux coefficients.

The contribution of DEM particle-wall conductive heat flux to the wall heat flux can be made by modifying wall heat flux coefficients a and c as follows:

For fixed wall heat flux conditions, only a needs to be modified:

$$a = \sum \frac{A_c}{A} q_w \tag{3242}$$

where:

- A_c is the particle-wall contact area.
- A the boundary face area.

For fixed wall temperature conditions, a and c are modified as:

$$\begin{aligned} a &= \sum 4 r_c k T_p \\ c &= \sum 4 r_c k \end{aligned} \tag{3243}$$

where:

- k is the particle thermal conductivity.
- r_c the contact radius.
- T_p the particle temperature.

Impact Heat Model

The impact heat model is a widely used approximate model. It calculates the rate of heat production that results from friction and damping in DEM particles using a linear formulation.

The rate of heat generation is:

$$q_r = c_t F_t v_t + c_n F_n v_n \tag{3244}$$

where:

- c_t and c_n are, respectively, the fractions of frictional work and damping work that are converted to heat.
- F_t and F_n are, respectively, the frictional (tangential) forces and damping (normal) forces on the particle.
- v_t and v_n are, respectively, the relative tangential and normal impact velocities.

The values for c_t and c_n must be estimated for the individual setup being simulated. The other factors are derived from other inputs in the DEM models. For examples, see Iwasaki [625] and Rojek [632].

Erosion

In STAR-CCM+, erosion modeling predicts the rate of erosion from particle impact on solid walls. Erosion rate is defined as the mass of wall material removed per unit area per unit time.

In calculating the erosion rate, STAR-CCM+ accumulates the damage that each particle impact causes. This calculation is done by selecting a correlation for the erosion ratio, that is, the mass of wall material eroded per unit mass of impinging particles. Erosion damage depends on both the material of the walls and the material of the particles. Any correlation for the erosion ratio is specific to a particular wall material, particle material, and carrier fluid. It can also be specific to a particular range of flow conditions.

STAR-CCM+ provides erosion models for both Lagrangian discrete material particles (droplets, bubbles, solid particles) and DEM particles. For the Lagrangian discrete material particles, erosion due to impact is modeled; for DEM particles, erosion due abrasion is modeled additionally, that is, erosion is the sum of impact erosion and abrasive erosion.

The following correlations for the erosion ratio are available:

- Ahlert
- DNV
- Neilson-Gilchrist
- Oka
- Archard

The Ahlert, DNV, Neilson-Gilchrist, and Oka correlations describe impact wear, from direct impact of particles on the eroded surface. The Archard correlation describes erosion due to scouring, where particles strike tangentially or at low angles.

Impact Erosion

The erosion rate is defined as the mass of wall material eroded per unit area per unit time. It is calculated on wall, baffle, and contact boundary faces by accumulating the damage that each particle impact does on the face:

$$E_f = \frac{1}{A_f} \Sigma_{\pi(f)} \dot{m}_\pi e_r \tag{3245}$$

where:

A_f is the area of the face

$\dot{m}_\pi$ is the mass flow rate of particles in parcel π impacting on the face

e_r is the erosion ratio

The summation is over all parcels which strike the face in this iteration or time-step. The erosion rate therefore depends on the flow (whether and how particles impact on the wall) and the chosen method for the erosion ratio e_r.

Abrasive Erosion

Abrasive erosion is available for DEM only. Abrasive erosion is due to scouring, where particles scrape tangentially along walls or strike at low angles. The abrasive erosion rate is calculated as:

$$E_f = \frac{1}{A_f dt} \sum_{\pi(f)} e_r \tag{3246}$$

where:

dt is the fluid time-step
e_r is the erosion ratio that is calculated with the Archard correlation

Erosion Correlations

Ahlert Correlation

The Ahlert correlation [565] for the erosion ratio is:

$$e_r = KF_s f(\alpha) \left(\frac{u_{\text{rel}}}{u_{\text{ref}}}\right)^n \tag{3247}$$

where:

K is a material-dependent constant
F_s is a factor to account for the shape of the particles
$f(\alpha)$ is a function expressing the dependency on the particle incidence angle
v_{rel} is the magnitude of the relative velocity of the particle with respect to the wall, $|\mathbf{v}_{\text{rel}}|, (\mathbf{v}_{\text{rel}} = \mathbf{v}_{\text{particle}} - \mathbf{v}_{\text{wall}})$
v_{ref} is a constant reference velocity
n a constant exponent

The shape coefficient F_s is reported to take the value 1 for angular particles, 0.53 for semi-rounded particles and 0.2 for fully rounded particles.

The angle function $f(\alpha)$ is split into two ranges. Below the user-specified transition angle α_0 it is a polynomial in α, the incidence angle in radians. Above the transition angle, $f(\alpha)$ follows a trigonometric relationship

$$f(\alpha) = x\cos^2\alpha \sin \omega\alpha + y\sin^2\alpha + z \tag{3248}$$

The constants w, x and y are user-defined, whereas z is calculated internally by requiring that $f(\alpha)$ is continuous at α_0.

The default coefficients for the Ahlert correlation are for liquid-borne semi-rounded sand particles eroding aluminum, as given by McLaury and others [586].

DNV Correlation

The DNV correlation [581] for the erosion ratio is:

$$e_r = Kf(\theta) \left(\frac{v_{\text{rel}}}{v_{\text{ref}}}\right)^n \tag{3249}$$

in which:

- K is a material-dependent constant, with a default value of 2.0E-9, for steel.
- $f(\theta)$ is a function expressing the dependency on the particle incidence angle. The default form of the function is:

$$f\theta = A\theta + B\theta^2 + C\theta^3 + D\theta^4 + E\theta^5 + F\theta^6 + G\theta^7 + H\theta^8 \tag{3250}$$

where the values of the coefficients are:

$$\begin{aligned} A &= 9.37 & E &= 170.137 \\ B &= -42.295 & F &= -98.398 \\ C &= 110.864 & G &= 31.211 \\ D &= -175.804 & H &= -4.170 \end{aligned}$$

θ is in radians and has a range of 0 to $\frac{\pi}{2}$.

- v_{rel} is the magnitude of the relative velocity of the particle with respect to the wall, $|\mathbf{v}_{\text{rel}}|, (\mathbf{v}_{\text{rel}} = \mathbf{v}_{\text{particle}} - \mathbf{v}_{\text{wall}})$.
- v_{ref} is a constant reference velocity. The default value is 1 m/s.
- n a constant exponent. The default value is 2.6.

The default coefficients for the DNV correlation are for air-borne sand eroding carbon steel, and are taken from [581]. The DNV correlation does not explicitly depend on particle diameter. However, the correlation was derived from experimental data observed for a mean particle diameter of 225 μm.

Neilson-Gilchrist Correlation

The Neilson-Gilchrist [588] correlation for the erosion ratio is:

$$e_r = e_{rC} + e_{rD} \tag{3251}$$

in which e_{rC} and e_{rD} represent contributions from cutting and deformation respectively. The cutting erosion is modeled as a function of the incidence angle α

$$e_{rC} = \begin{cases} \dfrac{v_{\text{rel}}^2 \cos^2\alpha \sin\frac{\pi\alpha}{2\alpha_0}}{2\varepsilon_C} & \alpha < \alpha_0 \\ \dfrac{v_{\text{rel}}^2 \cos^2\alpha}{2\varepsilon_C} & \alpha \geq \alpha_0 \end{cases} \tag{3252}$$

where:

- α_0 is the user-specified transition angle.
- ε_C is the user-specified cutting coefficient.
- v_{rel} is the magnitude of the relative velocity of the particle with respect to the wall, $|\mathbf{v}_{\text{rel}}|, (\mathbf{v}_{\text{rel}} = \mathbf{v}_{\text{particle}} - \mathbf{v}_{\text{wall}})$.

The deformation erosion is similarly

$$e_{rD} = \frac{max\left(u_{\mathrm{rel}}\sin\alpha - K, 0\right)^2}{2\varepsilon_D} \tag{3253}$$

with ε_D the deformation coefficient and K the cut-off velocity, below which no deformation erosion occurs.

The default coefficients for the Neilson-Gilchrist correlation are for liquid-borne sand eroding AISI 4130 steel, and are taken from Wallace and others [614].

Oka Correlation

The Oka correlation [592] and [593] for the erosion ratio is:

$$e_r = e_{90} g\left(\alpha\right)\left(\frac{v_{\mathrm{rel}}}{v_{\mathrm{ref}}}\right)^{k_2}\left(\frac{D_p}{D_{\mathrm{ref}}}\right)^{k_3} \tag{3254}$$

where:

- The angle function $g(\alpha)$ is defined as:

$$g\left(\alpha\right) = \left(\sin\alpha\right)^{n_1}\left(1 + H_v\left(1 - \sin\alpha\right)\right)^{n_2} \tag{3255}$$

with n_1, n_2 and H_v user-specified constants. Oka and others identify the value of H_v as the Vickers hardness of the eroded material in units of GPa.

- u_{rel} is the magnitude of the relative velocity of the particle with respect to the wall, $\left|\mathbf{v}_{\mathrm{rel}}\right|, \left(\mathbf{v}_{\mathrm{rel}} = \mathbf{v}_{\mathrm{particle}} - \mathbf{v}_{\mathrm{wall}}\right)$.
- u_{ref} is the user-specified reference velocity.
- D_{ref} is the user-specified reference diameter.
- k_2 and k_3 are user-specified exponents.

By inspection of Eqn. (3254) and Eqn. (3255), e_{90} is revealed to be the reference erosion ratio at $u_{\mathrm{rel}} = u_{\mathrm{ref}}$, $D_p = D_{\mathrm{ref}}$ and $\alpha = 90°$. In STAR-CCM+ the value of e_{90} is calibrated by equating values for erosion ratio from the Oka and DNV models at 90º impact angle, (DNV reference particle size), $D_{ref} = 326\mu m$ (Oka reference particle size), and 104 m/s. This equation gives:

$$e_{90} = e_{r,DNV@90}\left(D_{\mathrm{ref}}/D_p\right)^{k_3}$$

The purported strength of the Oka model is that the coefficients for a particular combination of eroded material and eroding material can be derived from more fundamental coefficients. These coefficients are specific to either the eroded material or the eroding material. Hence, for example, the fundamental coefficients for sand can serve as a basis for both sand-steel erosion and sand-aluminum erosion. The fundamental coefficients for the eroding material, in turn, are shown to be derivable from measurable properties of the eroding material such as its Vickers hardness.

STAR-CCM+ bases the Oka correlation on the coefficients in Eqn. (3254) and Eqn. (3255); for more information on relating these coefficients to fundamental coefficients, see [592] and [593]. The default coefficients are for air-borne sand eroding 0.25% carbon steel, and are taken from [592] and [593], except for e_{90} which is calibrated using the DNV correlation.

Archard Correlation

The Archard correlation [616] for the Abrasive Rate erosion ratio is:

$$e_r = aFs \tag{3256}$$

where:

- a is the Abrasive Wear Coefficient, with a default value of 0.01 kg/J. The default value of 1E-2 is the high end of the recommended range of values, 1E-8 to 1E-2, mild wear to severe wear. (See Archard Properties.)
- F is the normal force.
- s is the sliding distance.

Passive Scalar Interactions

There are two forms of interaction between passive scalar values: volume weighted and area weighted.

Volume Weighted Interaction

Volume weighted interactions model diffusion, with distinct forward and backward rates. This interaction conserves the density weighted passive scalar.

The volume weighted interaction models the mass flow rate as:

$$J = k\left(\rho_1\varphi_1 - \rho_2\varphi_2\right)A_p \tag{3257}$$

where:

- J is the mass flow rate.
- k is the interaction coefficient set under the Phase Interaction node.
- ρ_1 is the density of the Lagrangian particle.
- ρ_2 is the density of the Eulerian fluid.
- φ_1 is the value of the passive scalar for the Lagrangian particle.
- φ_2 is the value of the passive scalar for the Eulerian fluid.
- A_p is the surface area of the particle.

The source term for a typical Lagrangian particle in the parcel is:

$$S_{\varphi_1} = \left(-J\right)/V_p \tag{3258}$$

where:

- S_{φ_1} is the value of the passive scalar source for the particle.
- V_p is the volume of the particle.

The source term for the Eulerian cell is:

$$S_{\varphi_2} = n_p \cdot J/V_c \tag{3259}$$

where:

- n_p is the number of particles in the parcel.
- S_{φ_2} is the value of the passive scalar source for the fluid.
- V_c is the volume of the cell.

Area Weighted Interaction

Area weighted interactions model adsorption/desorption.

The area weighted interaction models the mass flow rate as:

$$J = \left(k_{1 \to 2}\rho_1\varphi_1 - k_{2 \to 1}\rho_2\varphi_2\right)A_p \tag{3260}$$

where:

- J is the mass flow rate.
- $k_{1 \to 2}$ and $k_{2 \to 1}$ are the interaction coefficients set under the Phase Interaction node.
- ρ_1 is the density of the Lagrangian particle.
- ρ_2 is the density of the Eulerian fluid.
- φ_1 is the value of the passive scalar for the Lagrangian particle.
- φ_2 is the value of the passive scalar for the Eulerian fluid.
- A_p is the surface area of the particle.

The source term for a typical Lagrangian particle in the parcel is:

$$S_{\varphi_1} = \left(-J\right)/A_p \tag{3261}$$

where:

- S_{φ_1} is the value of the passive scalar source for the particle.
- A_p is the surface area of the particle.

The source term for the Eulerian cell is:

$$S_{\varphi_2} = n_p \cdot J/V_c \tag{3262}$$

where:

- n_p is the number of particles in the parcel.
- S_{φ_2} is the value of the passive scalar source for the fluid.
- V_c is the volume of the cell.

Passive Scalar Transfer for DEM Particles

The Passive Scalar Transfer model describes how a passive scalar rubs off one DEM particle onto another. It is primarily intended for describing how a coating of negligible mass is shared among colliding particles.

The Area Weighted and Volume Weighted transfer methods are both described by this equation:

$$\dot{m} = k\left(C_1\rho_1 n_1 - C_2\rho_2 n_2\right)A_c \tag{3263}$$

where:

- $\dot{m}$ is the flow of passive scalar between phases 1 and 2, regarded as analogous to a mass flux.
- k is the value of the user-set property *Coefficient* in the **Area Weighted** or **Volume Weighted** node.
- C_1 and C_2 are the values of the passive scalar on phase 1 and phase 2.
- ρ_1 and ρ_2 are the particle densities of phase 1 and phase 2.

- n_1 and n_2 are the parcel particle counts for phase 1 and phase 2.
- A_c is the area that is computed from the harmonic mean of the radii of the two particles.

In Volume Weighted transfer, the flow is divided by the volume of the respective particles, while in Area Weighted transfer, the flow is divided by the surface areas. Compare these methods to the Passive Scalar Interactions under Lagrangian Multiphase Formulation.

In the Increment transfer method, a fixed amount of passive scalar (the increment, which is given by *Coefficient*) is added to each particle every time they make contact. This method does not model transfer of any physical property, but can be used in cases such as counting the number of contacts a particle undergoes.

Solution Methodology

This section describes the methodology that is used to solve the system of equations that the Lagrangian Multiphase and the Lagrangian Phase models generate.

The solution of equations in a Lagrangian framework implies the time-marching of ordinary differential equations (ODEs). The set of equations depends on the models that are invoked for the Lagrangian phase in question; the one common element is an ODE for the position of each parcel, Eqn. (2967).

This section covers:

- Unsteady Procedure
- Steady Procedure
- Time-Step Controls
- Two-Grid Procedure

Unsteady Procedure

A time-step in an unsteady simulation advances the solution from time t to time $t + \Delta t$. A single iteration of the unsteady solver for the Lagrangian Multiphase model involves:

1. Recovering the Lagrangian Multiphase solution at time t.
2. Advancing the Lagrangian Multiphase solution to time $t + \Delta t$ by time-marching. A local time-step is used for each parcel. The time-step is adjusted dynamically according to the time-step controls in force.

These two steps are executed in each iteration of the unsteady solver, before the flow solver. If the Two-Way Coupling model is active, source terms that the Lagrangian phase models compute are stored for subsequent application in their respective transport equations.

In an explicit unsteady case, this procedure means that the Lagrangian Multiphase solution is obtained using the continuous phase solution from the previous time-step.

Steady Procedure

An iteration in a steady simulation updates the steady solution. For the Lagrangian Multiphase model, the steady solution consists of

- Tracks, generated by the Track File model.
- Cell fields such as volume fraction and sources.
- Boundary fields such as incident mass flux.

A steady state is attained when these entities become invariant with further iteration. Despite this steady characteristic of the solution, it results from time-marching, where the time variable is the residence time of the parcels. The procedure in each iteration is:

- Deactivate the Lagrangian Multiphase solution.
- Generate the Lagrangian Multiphase solution by time-marching each parcel until it has left the computational domain, or has been removed, or until the user-specified maximum residence time is reached. A local time-step is used for each parcel, which is adjusted dynamically according to the time-step controls in force.

These two steps are executed in each iteration of the steady solver, before the flow solver.

Time-Step Controls

The local time-step δt_p controls the accuracy of the Lagrangian Multiphase solution. This time-step is used to time-march each parcel. The time-step is calculated dynamically using user-defined bounds and model time-scales.

You can use the Parcel Time Step field function to obtain the local time-step in field function expressions that are evaluated during substepping, for example in a passive scalar source. STAR-CCM+ can use the **Parcel Time Step** field function for analysis or visualization when temporary storage is turned on or when the Lagrangian Multiphase Solver is running.

User-Defined Bounds

Three parameters are available for you to control the local time-step δt_p. Smaller δt_p can increase accuracy, until temporal error becomes negligible. The three parameters are:

- The physical time-step Δt in unsteady simulations. Clearly $\delta t_p \leq \Delta t$.
- The maximum Courant number Co_{max}, which implies:

$$\delta t_p \leq \frac{Co_{max} \Delta x}{max\left(|\mathbf{v}|, \left|\mathbf{v}_p\right|\right)} \tag{3264}$$

where Δx is a characteristic length-scale of the cell containing the parcel.

- The minimum Courant number Co_{min}, which implies:

$$\delta t_p \geq \frac{Co_{min} \Delta x}{max\left(|\mathbf{v}|, \left|\mathbf{v}_p\right|\right)} \tag{3265}$$

Model Time-Scales

Models can also propose constraints for δt_p. Among the model time-scales are:

- The momentum relaxation time-scale, Eqn. (2975), provided by the Drag Force model.
- The thermal relaxation time-scale, Eqn. (3029), provided by the Lagrangian Energy model.
- The residual of the eddy interaction time, Eqn. (3012), provided by the Turbulent Dispersion model.
- The mass transfer time-scale $m_p / \dot{m}_p$ provided by the Droplet Evaporation model.

Two-Grid Procedure

Two-way coupling assumes that the fluid cell volume is large compared to the particle size. This assumption fails in cases where cells become small due to the demands of flow modeling.

To ensure correct coupling when cell sizes are small, STAR-CCM+ provides a two-grid procedure for coupling that clusters groups of contiguous cells together to create a virtual grid of larger cells. The simulation uses these larger cells for calculating parcel interactions with the fluid phase. After calculating interactions, the simulation distributes volume fraction contribution, momentum and energy source terms, and other transferred quantities evenly across the component cells.

You control the size of the large-scale cells with the *Scale* property of the **Cell Cluster** node. The cells of the large-scale grid are not always convex, but are approximately cubical in most cases. See Source Smoothing Method.

Lagrangian Multiphase Bibliography

[565] *Ahlert, K. 1994. “Effects of particle impingement angle and surface wetting on solid particle erosion of AISI 1018 steel”, MS Thesis,where: University of Tulsa, USA.*

[566] *Akao, J., Akaki, K., Mori, S., and Moriyama, A. 1980. “Deformation behaviors of a liquid droplet impinging on to a hot metal surface”, Transactions of the Iron and Steel Institute of Japan, 20, pp. 737-743.*

[567] *Apte, S.V., Mahesh, K., Gorokhovski, M., Moin, P. 2009. "Stochastic modeling of atomizing spray in a complex swirl injector using large eddy simulation", Proceedings of the Combustion Institute*

[568] *Apte, S. V., Gorokhovski, M. and Moin, P. September 2003. "LES of of atomizing spray with stochastic modeling of secondary breakup", International Journal of Multiphase Flow, Volume 29, Issue 9. pp. 1503-1522*

[569] *Auton, T.R., Hunt, J.C.R., and Prud’Homme, M. 1988. “The force exerted on a body in inviscid unsteady non-uniform rotational flow”, J. Fluid Mech., 197, pp. 241–257.*

[570] *Baumgarten, C. 2006. “Mixture Formation in Internal Combustion Engines”, Springer-Verlag.*

[571] *Bai, C. 1996. “Modeling of spray impingement processes”, Ph.D Thesis, University of London.*

[572] *Bai, C., and Gosman, A.D. 1995. “Development of methodology for spray impingement simulation”, SAE Technical Paper Series 950283.*

[573] *Bai, C.X., Rusche, H., and Gosman, A.D. 2002. “Modeling of gasoline spray impingement”, Atomization & Sprays, 12 (1-3), pp. 1–27.*

[574] *Birkhold, F., Meinast, U., Wassermann, P., and Deutschmann, O. 2006. "Analysis of the Injection of Urea-water-solution for Automotive SCR DeNOx-Systems: Modelling of Two-phase Flow and Spray-Wall Interaction", SAE-2006-01-0643.*

[575] *Brazier-Smith, P., Jennings, S., and Latham, J. 1972. “The Interaction of Falling Rain Drops: Coalescence” Proceedings of the Royal Society of London A, 326, pp. 393–408.*

[576] *Crowe, C.T., Sommerfeld, M., and Tsuji, Y. 1998. Multiphase Flows with Droplets and Particles. CRC Press, Boca Raton.*

[577] *Eckhause, J.E., Reitz, R.D. 1995. “Modeling heat transfer to impinging fuel sprays in direct-injection engines”, Atomization and Sprays, 5, pp. 213–242.*

[578] *El Wakil, M.M., Ueyhara, O.A., and Myers, P.S. 1954. “A theoretical investigation of the heating-up period of injected fuel droplets vaporizing in air”, NACA Tech. Note 3179.*

[579] *Gosman, A.D., and Ioannides, E. 1983. “Aspects of computer simulation of liquid-fueled combustors”, AIAA, J. Energy, 7(6), pp. 482–490.*

[580] *Han, Z., Parish, S., Farrell, P.V., and Reitz, R.D. 1997. "Modeling Atomization Processes of Pressure-Swirl Hollow-Cone Fuel Sprays", Atomization and Sprays, 7(6) pp. 663–684.*

[581] *Haugen, K., Kvernvold, O., Ronold, A., and Sandberg, R. 1995. "Sand erosion of wear-resistant materials: Erosion in choke valves", Wear, 186-187, pp. 179–188.*

[582] *Incopera, F.P., and DeWitt, D.P. 1996. Fundamentals of Heat and Mass Transfer. 4th Edition, John Wiley & Sons*

[583] *Kralj, C. 1995. "Numerical Simulation of Diesel Flow Processes", Ph.D. Thesis, Imperial College, London.*

[584] *Lefebvre, A.H. 1989. Atomization and Sprays, Taylor & Francis, Bristol PA*

[585] *Liu, A.B., Mather, D., and Reitz, R.D. 1993. "Modeling the Effects of Drop Drag and Breakup on Fuel Sprays", SAE Paper 930072.*

[586] *McLaury, B.S., Shirazi, S.A., Shadley, J.R., and Rybicki, E.F. 1996. "Modeling erosion in chokes", ASME FED conference, 236(1), pp. 773–781.*

[587] *Milne-Thompson, L.M. 1968. Theoretical Hydrodynamics, 5th edition, McMillan & Co., New York.*

[588] *Neilson, J.H., and Gilchrist, A. 1968. "Erosion by a stream of solid particles", Wear, 11, pp. 111–122.*

[589] *O'Rourke, P.J., and Bracco, F. 1980. "Modeling of Drop Interactions in Thick Sprays and a Comparison with Experiments" Proceedings of the Institution of Mechanical Engineers, 9, pp. 101–106.*

[590] *O'Rourke, P.J. 1981. "Collective Drop Effects on Vaporizing Liquid Sprays", Department of Mechanical and Aerospace Engineering, Princeton University.*

[591] *O'Rourke, P.J., and Amsden, A.A. 1987. "The TAB Method for Numerical Calculation of Spray Droplet Breakup", SAE Paper 872089.*

[592] *Oka, Y.I, Okamura, K., and Yoshida, T. 2005. "Practical estimation of erosion damage caused by solid particle impact. Part 1: Effect of impact parameters on a predictive equation", Wear, 259, pp. 95–101.*

[593] *Oka, Y.I., and Yoshida, T. 2005. "Practical estimation of erosion damage caused by solid particle impact. Part 2: Mechanical properties of materials directly associated with erosion damage", Wear, 259, pp. 102–109.*

[594] *Patterson, M.A., and Reitz, R.D. 1998. "Modeling the Effects of Fuel Spray Characteristics on Diesel Engine Combustion and Emission", SAE Paper 980131.*

[595] *Rangel, R.H., and Sirignano, W.A. 1991. "The linear and nonlinear shear instability of a fluid sheet", Phys. Fluids A., 3 p. 2392.*

[596] *Ranz,W.E., and Marshall,W.R. 1952. “Evaporation from drops Parts I and II”, Chem. Eng. Prog., 48(3), p. 141.*

[597] *Reitz, R.D. 1987. “Modeling Atomization Processes in High-Pressure Vaporizing Sprays”, Atomisation and Spray Technology, 3, pp. 309–337.*

[598] *Reitz, R.D., and Diwakar, R. 1986. “Effect of Drop Breakup on Fuel Sprays”, SAE Paper 860469.*

[599] *Reitz, R.D., and Diwakar, R. 1987. “Structure of High-Pressure Fuel Sprays”, SAE Paper 870598.*

[600] *Ricart, L.M., Reitz, R.D., Dec, J.M. 1997. “Comparisons of Predicted Diesel Spray Liquid Penetration and Vapor Fuel Distributions with In-Cylinder Optical Measurements”, Paper No. 97-ICE-52, ICE-Vol. 29-2, 1997 Fall Technical Conference, ASME.*

[601] *Rohsenow, W.M. 1952. ‘A method of correlating heat transfer data for surface boiling liquids’, Transactions of the ASME, 74, p. 969.*

[602] *Rosa, N.G., Villedieu, P., Dewitte, J., and Lavergne, G., 2006. "A New Droplet-Wall Interaction Model", Paper ID ICLASS06-167, ICLASS-2006.*

[603] *Satoh, K., Kawai, T., Ishikawa, M., and Matsuoka, T., 2000. “Development of Method for Predicting Efficiency of Oil Mist Separators”, SAE Technical Paper Series, 2000-01-1234*

[604] *Schiller, L., and Naumann, A. 1933 “Ueber die grundlegenden Berechnungen bei der Schwerkraftaufbereitung”, VDI Zeits., 77(12), pp. 318–320.*

[605] *Schmidt, D.P., Nouar, I., Senecal, P.K., Rutland, C.J., Martin, J.K., Reitz, R.D., and Hoffman, J.A. 1999. “Pressure-Swirl Atomization in the Near Field”, SAE Technical Paper Series, 1999-01-0496.*

[606] *Schmidt, D.P., and Rutland, C.J. 2000. “A New Droplet Collision Algorithm”, J Comput Phys, 164, pp. 62–80.*

[607] *Senecal, P.K., Richards, K.J., Pomraning, E., Yang, T., Dai, M.Z., McDavid, R.M., Patterson, M.A., Hou, S., and Shethaji, T. 2007. “A New Parallel Cut-Cell Cartesian CFD Code for Rapid Grid Generation Applied to In-Cylinder Diesel Engine Simulations”, SAE Paper 2007-01-0159.*

[608] *Senecal, P.K., Schmidt, D.P., Nouar, I., Rutland, C.J., and Reitz, R.D. 1999.”Modeling High Speed Viscous Liquid Sheet Atomization”, Int. J. Multiphase Flow, 25, pp. 1073–97.*

[609] *Sirignano, W.A. 1999. Fluid dynamics and transport of droplets and sprays, Cambridge University Press, Cambridge.*

[610] *Spalding, D.B. 1960. “A standard formulation of the steady convective mass transfer problem”, Int. J. Heat Mass Transfer, 1, pp. 192-207.*

[611] *Squire, H.B. 1953. “Investigation of the instability of a moving liquid film”, Brit. J. Appl. Phys., 4 p. 167.*

[612] *Stiesch, G. 2003. “Modeling Engine Spray and Combustion Processes”, Springer (Berlin).*

[613] *Taylor, G. I. 1963. "Generation of ripples by wind blowing over a viscous fluid," Scientific Papers of Sir G. I. Taylor (Ed. G. K. Batchelor), 3.*

[614] *Wallace, M.S., Peters, J.S., Scanlon, T.J., Dempster, W.M., McCulloch, S., and Ogilvie, J.B. 2000. "CFD-based erosion modeling of multi-orifice choke valves", ASME, Paper FEDSM2000-11244.*

[615] *Wruck, N. 1998. "Transientes Seiden von Tropfen beim Wandaufprall", Ph.D. Thesis, RWTH Aachen.*

DEM

[616] *Archard, J. F., and Hirst, W. 1956. "The wear of metals under unlubricated conditions", Proceedings of the Royals Society of London. Series A, Mathematical and Physical Sciences. 236(1206), pp 397-410.*

[617] *Butt, H-J., Graf, K. and Kappl, M. 2006. Physics and Chemistry of Interfaces, Wiley-VCH.*

[618] *Camacho, G.T. and M. Ortiz, 1996, "Computation Modelling of Impact Damage in Brittle Materials," Int'l J. Solids Structures, 33(20-22) pp 2899-2938.*

[619] *Cheng, G. J., Yu, A. B. and Zulli, P. 1999. "Evaluation of effective thermal conductivity from the structure of packed bed", Chemical Engineering Science, 54, pp. 4199-4209.*

[620] *Cundall, P. A., Strack, O. D. L. 1979. "A discrete numerical model for granular assemblies", Geotechnique, 29, pp. 47-65.*

[621] *Di Felice, R. 1994. "The voidage function for fluid-particle interaction systems", Int. J. Multiphase Flow, 20, pp. 153-159.*

[622] *Di Renzo, Alberto and Di Maio, Francesco Paolo. 2004. "Comparison of contact-force models for the simulation of collisions in DEM-based granular flow codes", Chemical Engineering Science 59 pp. 525-541.*

[623] *Gidaspow, D. 1994. Multiphase Flow and Fluidization-Continuum and Kinetic Theory Descriptions, Academic Press.*

[624] *Haider, A. and Levenspiel, O. 1989. "Drag coefficient and terminal velocity of spherical and non-spherical particles", Powder Technology, 77, pp. 143-152.*

[625] *Iwasaki, T., Munetake, S., Takaomi, K.. 2001. "Analysis of collision energy of bead media in a high-speed elliptical rotor-type poweder mixer using the discrete element method," Powder Technology 121 pp. 239-248.*

[626] *Johnson, K.L. 1987. Contact Mechanics, Cambridge University Press, Cambridge.*

[627] *Kosteski, L., R.B. D'Ambra and I. Iturrioz, 2012. "Crack propagation in elastic solids using the truss-like discrete element method," Int. J. Fract., 174, pp. 139-161.*

[628] *Pham, Chi Vinh, Ogden, R.W. 2004. “On formulas for the Rayleigh wave speed”, Wave Motion, 39 pp. 191-197.*

[629] *Pichugin, A.V. 2008. “Approximation of the Rayleigh wave speed”.*

[630] *Potyondy, D.O, and Cundall, P.A. 2004. “A bonded-particle model for rock”, Int. J. Rock Mechanics & Mining Sciences 41 pp. 1329–1364.*

[631] *Rayleigh, L. 1885. “On waves propagated along the plane surface of an elastic solid”, Proc. Lond. Math. Soc. 17.*

[632] *Rojek, J. 2014. "Discrete element thermomechanical modelling of rock cutting with valuation of tool wear," Comp. Part. Mech. 1:71-84.*

[633] *Saffman, P.G. 1965. "The lift on a small sphere in a slow shear flow," J. Fluid Mech. 22 385-400. (See also Saffman, P.G. 1968. Corrigendum to: “The lift on a small sphere in a slow shear flow,“ J. Fluid Mech. 31 624.)*

[634] *Sommerfeld, M. 2000. "Theoretical and Experimental Modelling of Particulate Flows," Technical Report Lecture Series 2000-06, von Karman Institute for Fluid Dynamics pp 20-23.*

[635] *Timoshenko, S. 1951. Theory of Elasticity, McGraw-Hill Book Company.*

[636] *Trent, B.C., L.G. Margolin, P.A. Cundall, and Gaffney, E.D., 1986. "The Micro-Mechanics of Cemented Granular Material," Second Int'l. Conf. On Constitutive Laws for Engineering Materials, Tucson, Arizona.*

[637] *Tsuji, Y., Tanaka, T., Ishida, T. 1992. “Lagrangian numerical simulation of plug flow of cohesionless particles in horizontal pipe.” Powder Technology 71:3 pp. 239-250.*

[638] *Von Neumann, J., and Richtmyer, R. D. 1950. "A method for the numerical calculation of hydrodynamic shocks", J. Appl. Phys. 21, 232–237.*

[639] *Walton, O.R. 1993. “Numerical simulation of inclined chute flows of monodisperse inelastic, frictional spheres”, Mech. Mater. 16 239–247.*

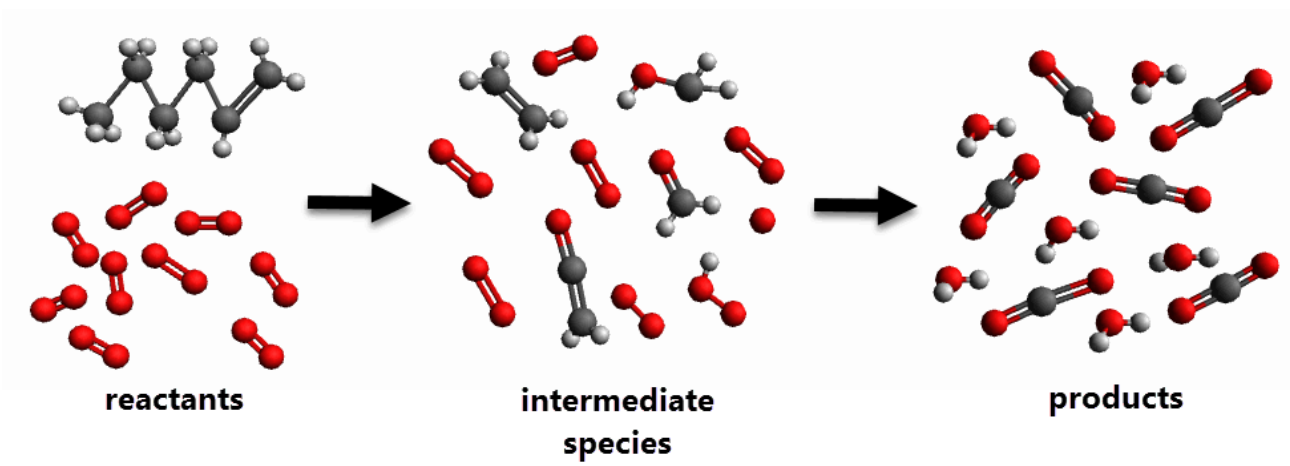
[640] *Walton, O.R. 1993. “Numerical simulation of inelastic, frictional particle-particle interactions”, in Particulate Two-Phase Flow, M.C. Roco, Ed., Butterworth–Heinemann, Stoneham, MA, pp. 884–911.*

[641] *Walton, O.R. and Braun, R.L. 1986. “Viscosity, granular-temperature, and stress calculations for shearing assemblies of inelastic, frictional disks”, J. Rheology 30:5, pp. 949–980.*

Reacting Flow

In reacting flows, chemical species mix with each other and react when conditions allow. To model these flows, STAR-CCM+ couples species and energy transport equations with the chemistry solvers that compute the source terms.

For example, the following diagram represents a reaction between a hydrocarbon (hexene) and oxygen. Often many intermediate reactions occur before the final products are created.



Examples of reacting flow systems include:

- combustion in gas turbine combustors
- surface reactions in aftertreatment catalysts
- liquid-liquid reactions in mixing vessels
- polymerization in a tubular reactor

During combustion of hydrocarbon fuels, for example, tens of thousands of reactions occur between thousands of species. Researchers develop reaction mechanisms that represent these real-life combustion scenarios in a sequence of elementary reactions. Reaction mechanisms contain anywhere from one to thousands of reactions. These reactions often have a wide variety of timescales—anywhere between 10^{-9}s and 10^{2}s.

As an example, a simplified global reaction mechanism for propane combustion can be defined as three irreversible reactions:

$$C_3H_8 + 1.5O_2 \rightarrow 3CO + 4H_2$$

$$CO + 0.5O_2 \rightarrow CO_2$$

$$H_2 + 0.5O_2 \rightarrow H_2O$$

The reaction mechanism that is defined above consists of six species and three irreversible reactions. The Species transport model solves for the conservation of each of the species in this system (as described in Eqn. (2202)). The source terms that are required to close the equations are provided by the chemistry solver by utilizing the kinetic rate information that is described in the reaction mechanism. No additional chemistry-related source terms are required for the energy equation as STAR-CCM+ solves the energy equation in chemical-thermal enthalpy form.

Modeling Approaches

STAR-CCM+ categorizes reacting flow models into two types—reacting species transport and flamelet.

- In reacting species transport models, the conservation equations—which include the chemical source term— are solved for all species in a mechanism.

  The reacting species transport modeling approach is suitable for reacting flow systems in which the mixing timescale, τ_{mix}, is shorter than the reaction timescale, τ_{kin}, for one or more species. This approach is useful for simulating slowly forming pollutants, part-load conditions in combustors, and transient combustion such as explosions, ignition, and extinction.
- In flamelet models, the reacting flow system is parameterized by a limited number of flamelet variables which describe the thermochemical state in the CFD cell. Instead of solving the conservation equations for all of the species, the conservation equations are solved for the limited number of flamelet variables. The number and type of flamelet variables for which the conservation equations are solved depend on the specific flamelet model. The flamelet approach has a reduced computational cost compared to the reacting species transport approach because a reduced number of transport equations are solved and the chemistry in the reaction mechanism is solved before the CFD simulation.

  The flamelet modeling approach is suitable for reacting flow systems in which the reaction timescale is shorter than the mixing timescale. This approach is useful for steady-burning furnaces and burners, and full-load conditions in combustors.

In reacting flow simulations, the continuity equation and the conservation equation for flow are solved according to Eqn. (1461) and Eqn. (1462) respectively.

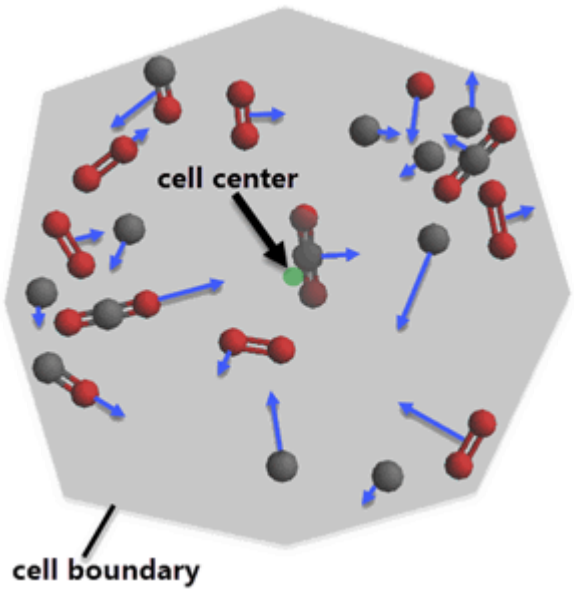
The chemical source term, S_i, accounts for:

- For reacting species transport models, the conservation equations for all species are solved using the mass fraction Y_i of species i.
- For flamelet models, the conservation equations are not solved for the species mass fractions. Instead, conservation equations are solved for the relevant flamelet variables:
  - mixture fraction Z
  - mixture fraction variance Z_{var}
  - *unnormalized progress variable y
  - *unnormalized progress variable variance y_{var}

  (*When using the FGM model.)

  The chemical source term is also included in each progress variable conservation equation.

In STAR-CCM+, the main difference between each of the reacting flow models is how the chemical source termS_i is formulated. Chemical reactions are exponentially dependent on temperature and successful collisions of reactants. The reaction rates depend on local instantaneous species and temperature distributions—which are not resolved in RANS and LES calculations. Therefore, different models are available to account for the fact that the average rate of reaction that is calculated for a cell does not accurately reflect the rate of reaction at specific locations within the cell. For example, differences in temperature within the cell result in species having differences in the kinetic energy that they have available for reacting.



When the average reaction rate is affected by the turbulence as well as the chemistry, this phenomenon is called turbulence-chemistry interactions. Different models are used to account for these turbulence-chemistry interaction effects. Species transport models account for this effect through turbulent diffusion between cells. Time scale models can also be applied, where the time scale of the laminar chemical source term (the reaction rate calculated from average quantities) is combined with a time scale for turbulence. In flamelet models, the turbulence-chemistry interaction effects are handled through assumed shape probability density functions (PDFs) for the flamelet variables—which consider the effect of local species and temperature distributions. In addition, flame propagation models can be used to specifically model the movement of a flame front as function of turbulence and chemical state.

Contents:

Basic Definitions
Governing Equations for Reacting Flows
Chemical Kinetics
Reactions—Species Transport
Gas Phase Combustion—Flamelet Models
Fluid-Fluid Interphase Reactions
Surface Chemistry
Ignitors
NOx

Soot
Particle Reactions
Polymerization
Reacting Flow Nomenclature
Reacting Flow Bibliography

Basic Definitions

Some of the specific terms that are used throughout the reacting flow theory content are summarized here.

Basic Definitions

Cell Residence Time

The cell residence time τ_{res} is the time step over which chemistry in a cell is evaluated. For unsteady flows this is the specified time-step, and for steady flows, this is the time that a fluid element spends in the cell—calculated as the mass in the cell divided by the mass flux in the cell.

$$\tau_{res} = \frac{m}{\dot{m}} \tag{3266}$$

Combustion Progress Variable

Unnormalized Progress Variable

The unnormalized progress variable y is used to track the progress of combustion. It is calculated by STAR-CCM+ based either on chemical enthalpy or species weights W_i—depending on the option that is chosen when generating the flamelet table.

$$y = \Sigma\left(W_i Y_i\right) \tag{3267}$$

Unnormalized Progress Variable Variance

The unnormalized progress variable variance, denoted as y_{var}, is calculated using either a transport equation or by an algebraic relationship.

$$y_{\mathrm{var}} = \overline{(y - \overline{y})^2} \tag{3268}$$

Normalized Progress Variable

The normalized progress variable is denoted as c. When $c = 0$, combustion has not yet commenced. When $c = 1$, combustion is complete. STAR-CCM+ calculates the normalized progress variable by considering the proportions of products that are formed—such as CO and $\mathrm{CO_2}$ during hydrocarbon combustion, or $\mathrm{H_2O}$ during hydrogen combustion.

$$c = \frac{y - y_{unburnt}}{y_{burnt} - y_{unburnt}} \tag{3269}$$

Equivalence Ratio

The equivalence ratio Φ relates the mass fraction Y of fuel to oxidizer in an unburned mixture to that of a stoichiometric mixture. It is calculated from the amount of carbon, hydrogen and oxygen in the mixture:

$$\Phi = \frac{Y_{(C,H)u}/Y_{(O)u}}{\left(Y_{(C,H)u}/Y_{(O)u}\right)_{stoich}} \tag{3270}$$

$$= \frac{m_{(C,H),u}/m_{(O),u}}{\left(m_{(C,H),u}/m_{(O),u}\right)_{stoich}} \tag{3271}$$

$$= \frac{n_{(C,H),u}/n_{(O),u}}{\left(n_{(C,H),u}/n_{(O),u}\right)_{stoich}} \tag{3272}$$

where u denotes unburnt.

Flamelet

A flamelet refers to a thin laminar flame which can also be referred to in combustion simulations as a fine structure.

Mass Fraction

The mass fraction Y_i of species i is given by the fraction of the total mass of all molecules of that species in a mixture m_i to the total mass of all molecules of all species in the mixture, m:

$$Y_i = \frac{m_i}{m} \tag{3273}$$

where:

$$m = \sum_i m_i \tag{3274}$$

and m_i is calculated as:

$$m_i = \rho_i V \tag{3275}$$

where V is the volume.

Mean Molecular Weight

The mean molecular weight of a mixture is defined as:

$$\overline{W} = \frac{m}{n} \tag{3276}$$

Combining Eqn. (3279), Eqn. (3281), Eqn. (3273), and Eqn. (3276) gives an expression for the relationship between mole and mass fractions.

$$Y_i = \frac{W_i}{\overline{W}} X_i \tag{3277}$$

Mixture Fraction

The mixture fraction Z is defined as the elemental composition that originated from the fuel stream. For example, consider a hydrocarbon molecule CnHm burning in air. The mixture fraction is then the elemental mass fraction of C and H, regardless of the state of reaction. That is, the C and H atoms can be in unburnt fuel, CnHm, or in burnt products of H_2O and CO_2.

The mixture fraction ranges from 0 for pure oxidizer streams, to 1 for pure fuel streams.

In the following equations, $(fuel)$ and $(oxid)$ denote the two streams (fuel and oxidiser), u and b denote unburned and burned respectively, and F and O denote fuel and oxidizer. If inert gases like argon (Ar) are present in the streams, the mixture fraction is calculated with respect to mass fractions of fuel and oxidizer

in the two streams. Let $Y_{F,(fuel)}$ be the mass fraction of fuel in $\dot{m}_{(fuel)}$ and $Y_{O,(oxid)}$ the mass fraction of oxidizer in $\dot{m}_{(oxid)}$. The fuel mass fraction in the mixture is proportional to the fuel mass fraction in $\dot{m}_{(fuel)}$.

$$Y_{F,u} = Y_{F,(fuel)} Z \tag{3278}$$

Similarly, the oxidizer mass fraction in the mixture is proportional to the oxidizer mass fraction in $\dot{m}_s$.

Mole Fraction

The mole fraction X_i of species i, is given by the fraction of moles which that species occupies in the mixture n_i to the total number of moles of all species in the mixture, n_s:

$$X_i = \frac{n_i}{n_s} \tag{3279}$$

where:

$$n_s = \sum_i n_i \tag{3280}$$

Molecular Weight

The molecular weight of species i is given by the sum of the atomic weight of all elements e in the species, weighted with the number of elements e in the given species i.

$$W_i = \sum_e a_{ie} W_e \tag{3281}$$

As an example, molecular hydrogen H2, which consists of two hydrogen atoms, has a molecular weight of twice the molecular weight of atomic hydrogen, H.

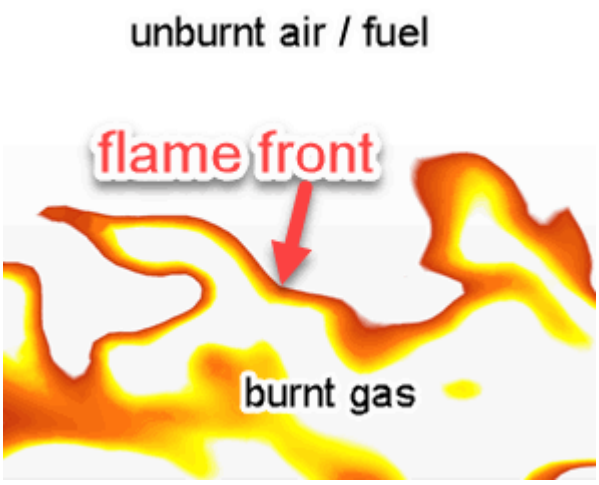
The molecular weight can also be calculated from a given mixture of species:

$$W_i = \frac{m_i}{n_i} \tag{3282}$$

where n_i is the number of moles of species i and m_i is the mass of species i.

Reaction Front (Flame Front)

The location at which reactions occur in a flame is known as the flame front, or reaction front. To determine the position of the flame front, often the concentration of OH (an intermediate species) is monitored.



Scalar Dissipation Rate

The scalar dissipation rate, χ, is the rate at which fluctuations in scalar variables (that are caused by turbulence) are dissipated. The scalar dissipation rate is given by:

$$\chi = \frac{D_f Z_{\mathrm{var}}}{\tau_{turb}} \tag{3283}$$

where D is the effective diffusivity and τ_{turb} is the turbulent timescale.

Flame temperatures depend strongly on the scalar dissipation rate—the maximum flame temperature decreases with an increasing scalar dissipation rate. Therefore, high values of scalar dissipation rate can cause extinction. When simulating a turbulent non-premixed flame, the scalar dissipation rate is highest near the inlet nozzle of a combustor (where the velocity is high). Extinction of the flame near the nozzle can cause lift-off of turbulent flames.

Stoichiometric Coefficient

υ_{ij} is the stoichiometric coefficient of species i which defines how many moles of the species take part in reaction j. For example, for a system of two reactions,

$$CH_4 + 2O_2 \rightarrow CO_2 + 2H_2O \tag{3284}$$

$$C + O_2 \rightarrow CO_2 \tag{3285}$$

the stoichiometric coefficients are:

Species	Reaction 1	Reaction 2
CH_4	1	0
O_2	2	1
C	0	1
CO_2	1	1

H_2O	2	0

In the STAR-CCM+ GUI, all stoichiometric coefficients are entered as positive numbers.

For example, reaction 1 (above) shows that for every one mole of CH_4 that is consumed, there are two moles of H_2O produced.

Governing Equations for Reacting Flows

In reacting flow simulations, the continuity equation and the conservation equation for flow are solved according to Eqn. (1461) and Eqn. (1462) respectively.

The conservation equation for energy is solved, with the addition of the source term that accounts for the heat that is released in the reaction:

Differential form:

$$\frac{\partial(\rho E)}{\partial t} + \nabla\cdot\left(\rho E\mathbf{v}\right) = \mathbf{f}_{\mathrm{b}}\cdot\mathbf{v} + \nabla\cdot(\mathbf{v}\cdot\boldsymbol{\sigma}) - \nabla\cdot\mathbf{q} + S_E \tag{3286}$$

Integral form:

$$\frac{\partial}{\partial t}\int_V \rho E dV + \oint_A \left[\rho H\mathbf{v}_r + \mathbf{v}_g p\right]\cdot d\mathbf{a} = -\oint_A \dot{\mathbf{q}}''\cdot d\mathbf{a} + \oint_A \mathbf{T}\cdot\mathbf{v}d\mathbf{a} + \int_V \mathbf{f}_b\cdot\mathbf{v}dV + \int_V S_E dV \tag{3287}$$

where S_E is the source term (which includes the source term from the reactions).

Additionally, conservation equations for species mass fractions (Y_i) are solved:

Differential form:

$$\frac{\partial\rho Y_i}{\partial t} + \nabla\cdot\left(\rho U Y_i\right) = \nabla\cdot\left(J_i + \frac{\mu_t}{\sigma_t}\nabla Y_i\right) + S_{Y_i} \tag{3288}$$

Integral form:

$$\frac{\partial}{\partial t}\left(\int_{\widetilde{V}} \rho Y_i\right)d\widetilde{V} + \oint_A \rho Y_i\left(\mathbf{v}-\mathbf{v}_g\right)\cdot d\widetilde{\mathbf{a}} = \oint_A \left[J_i + \frac{\mu_t}{\sigma_t}\nabla Y_i\right]\cdot d\widetilde{\mathbf{a}} + \int_{\widetilde{V}} S_{Y_i}\,\mathrm{d}\,\widetilde{V} \tag{3289}$$

where S_{Y_i} is the mass fraction source term from reactions.

Chemical Kinetics

Chemical kinetics is concerned with the mechanisms and rates of reaction of chemical species at the molecular level.

When modeling reacting species transport, providing a source term to the transport equations requires calculation of the reaction rates (chemical kinetics) for all reactions in the chemical mechanism.

When modeling flamelets, the reactions are pre-calculated, whereas reacting species transport calculates the rates during run-time.

Any elementary chemical reaction can be represented by the following general equation.

$$\upsilon'_A A + \upsilon'_B B + \ldots \underset{k_b}{\overset{k_f}{\rightleftharpoons}} \upsilon''_C C + \upsilon''_D D + \ldots \tag{3290}$$

Where the stoichiometric coefficients υ with superscript ' indicate a reactant and '' indicate a product. A net stoichiometric coefficient υ_{ij} gives the total number of moles of species i that are produced or consumed by the reaction, j:

$$\upsilon_{ij} = \upsilon''_i - \upsilon'_i \tag{3291}$$

Reactions can proceed forwards where reactants become products, or backwards where products breakup to form the reactants. k_f is the rate constant of the forwards reaction and k_b is the rate constant of the backwards reaction. k is the overall rate constant which accounts for both k_f and k_b.

$$k = k_f - k_b \tag{3292}$$

Reaction Rate

The reaction rate for elementary reactions is calculated as follows.

Taking for example one reaction from a reaction mechanism:

$$2\mathrm{CO} + \mathrm{O}_2 \overset{k}{\leftrightarrow} 2\mathrm{CO}_2 \tag{3293}$$

there are two reactant species and one product species. In order for the reaction to proceed, the reactant molecules must collide with energy that is equal to or greater than the activation energy E_a. The rate r_k at which this overall reaction proceeds is also dependant upon the rate constant k, and the concentration of the two reactants to the power of their stoichiometric coefficients.

In equation Eqn. (3293), there are twice as many CO molecules as O_2 molecules.

The rate equation is therefore written as:

$$r_k = k_f \ [\mathrm{CO}]^2 k_f \left[\mathrm{O}_2\right] - k_b \left[\mathrm{CO}_2\right]^2 \tag{3294}$$

For a general reaction, Eqn. (3290), this is represented as:

$$r_k = k_f [A]^{\upsilon'_A} [B]^{\upsilon'_B} \ldots - k_b [C]^{\upsilon''_C} [D]^{\upsilon''_D} \ldots \tag{3295}$$

The chemical formula of a species in square brackets represents the concentration of that species in the reaction mixture.

Since reactants are consumed during reactions, the rate of the reaction can also be expressed as a function of its concentration over time, such as CO in Eqn. (3293):

$$\frac{\mathrm{d}[\mathrm{CO}]}{\mathrm{d}t} = -k_f \cdot [\mathrm{CO}]^2\left[\mathrm{O}_2\right] + k_b\left[\mathrm{CO}_2\right]^2 \tag{3296}$$

For a general one-step reaction, Eqn. (3290), this is represented as:

$$\frac{\mathrm{d}[A]}{\mathrm{d}t} = \upsilon_A\left[k_f \cdot [A]^{\upsilon'_A}[B]^{\upsilon'_B}\ldots - k_b[C]^{\upsilon''_C}[D]^{\upsilon''_D}\ldots\right] \tag{3297}$$

where

$$\upsilon_A = \upsilon''_A - \upsilon'_A \tag{3298}$$

Each reaction proceeds at its own rate hence, at a given location, it has a unique time scale that depends on the concentrations of different species at that location, the temperature, and pressure.

The production (or consumption) rate r_i of species i, depends on the reaction rates of the reactions which produce and consume the species, and on the concentration of all species which participate in those reactions. The total net rate is therefore a sum of Eqn. (3297) over all reactions in which species i participates. In the equation below, j (=1, 2, … , N_j) denotes reactions and i denotes species.

$$r_i = \frac{d[A]}{dt} = \sum_{j=1}^{N_j} \upsilon_{ij} r_j \tag{3299}$$

$$r_j = k_{f,j}\prod_i [A]^{\upsilon'_{j,i}} - k_{b,j}\prod_i [A]^{\upsilon''_{j,i}} \tag{3300}$$

where the rate coefficients, k_f for the forwards reaction and k_b for the backwards reaction, are determined by Eqn. (3301).

Rate Constant

The rate constant k is most often determined using the Arrhenius equation, in which the rate has a non-linear dependance on temperature:

$$k = AT^{\beta}\exp\left(-\frac{E_a}{R_u T}\right) \tag{3301}$$

where R_u is the universal gas constant and the following values are empirical factors that are supplied by the chemical mechanism:

- the temperature exponent β
- the activation energy E_a
- the frequency factor, or pre-exponential factor A

Third Body Reactions

Third Body Efficiencies

Some reactions require a “third body” in order to proceed, as in:

$$H + O_2 + M \leftrightarrow HO_2 + M \tag{3302}$$

where M represents the third body. Third bodies act as a form of catalyst. Typically, any species in the mixture can act as a third body. Third bodies affect the rate at which the reaction proceeds.

The expression for the reaction rate for a reaction which requires a third body is as follows:

$$r_j = \left(\sum_{i=1}^{N}\left(\alpha_{ij}\right)\left[X_i\right]\right)\left(k_{f,j}\prod_{i=1}^{N}\left[X_i\right]^{v'_{ij}} - k_{b,j}\prod_{i=1}^{N}\left[X_i\right]^{v''_{ij}}\right) \tag{3303}$$

for species $(i = 1, \ldots, N)$:

- α_{ij} represents the collision efficiency of the third body
- $k_{f,j}$ is the forward rate constant of reaction j
- $k_{b,j}$ is the backwards rate constant of reaction of reaction j

When $\alpha_{ij} = 1$, all species in the mixture contribute equally as third bodies and the total concentration of the mixture is:

$$\left[M\right] = \sum_{i=1}^{N}\left[X_i\right] \tag{3304}$$

However, when $\alpha_{ij} \neq 1$, the concentration of the effective third body in the reaction is:

$$\left[M\right] = \sum_{i=1}^{N}\left(\alpha_{ij}\right)\left[X_i\right] \tag{3305}$$

where α_{ij} is specified for the third body species in the reaction mechanism.

Pressure-Dependent Chemical Reaction Rates

Pressure dependent chemical reactions (fall-off reactions) have a first-order rate at low pressure and become zero-order as the concentration of the third-body species increases. In this type of reaction, the rate is affected by pressure as well as temperature, STAR-CCM+ calculates the rate constants using the pre-exponential factor A that is provided by the reaction mechanism:

$$k = k_\infty\left(\frac{P_r}{1 + P_r}\right)F \tag{3306}$$

where the reduced pressure P_r is,

$$P_r = \frac{k_0\left[M\right]}{k_\infty} \tag{3307}$$

where $[M]$ is the concentration of the species M with third-body efficiencies.

The low pressure limit k_0 is,

$$k_0 = A_0 T^{\beta 0}\exp\left(-\frac{E_0}{R_u T}\right) \tag{3308}$$

and the high pressure limit k_∞ is,

$$k_\infty = A_\infty T^{\beta\infty}\exp\left(-\frac{E_\infty}{R_u T}\right) \tag{3309}$$

The blending (fall off) function, F, is calculated differently, depending on the approach that you use to specify the pressure-dependent parameters.

- Using the Lindemann method, F is calculated as unity, $F = 1$
- Using the Troe method, F is calculated as:

$$\log F = \left[1 + \left[\frac{\log P_r + c}{n - d(\log P_r + c)}\right]^2\right]^{-1} \log F_{cent} \tag{3310}$$

where:

$$F_{cent} = (1 - \alpha)\exp\left(-\frac{T}{T***}\right) + \alpha\exp\left(-\frac{T}{T*}\right) + \exp\left(-\frac{T**}{T}\right) \tag{3311}$$

- $c = -0.4 - 0.67\log F_{cent}$
- $n = 0.75 - 1.27\log F_{cent}$
- $d = 0.14$
- α, T, $T*$, $T**$, and $T***$ are constants that you specify within the imported chemical mechanism

- Using the (SRI) method, F is calculated as:

$$F = d\left[a\exp\left(\frac{-b}{T}\right) + \exp\left(\frac{-T}{c}\right)\right]^X T^e \tag{3312}$$

where a, b, c, d, and e are parameters that you specify within the imported chemical mechanism. When d and e are not specified in the mechanism, they are set to 1 and 0 respectively, by default.

$$X = \frac{1}{1 + \log^2 P_r} \tag{3313}$$

For both the Troe and SRI methods, F approaches unity for both high and low pressure limits.

Chemically-Activated Bimolecular Reaction Rates

In chemically-activated bimolecular reactions, the reaction rate decreases (falls off) as the pressure increases, due to the collisional stabilization of reaction intermediates. The forward reaction rate constant is proportional to the low-pressure rate constant:

$$k = k_0\left(\frac{1}{1 + P_r}\right)F \tag{3314}$$

The blending function F is also defined using the Troe, Lindemann, or SRI method.

Pressure Dependence Through Logarithmic Interpolation (PLOG)

When using the PLOG method to describe the pressure dependence of a reaction rate, the reaction rate is calculated by the Arrhenius expression Eqn. (3301) using rate parameters (β, E_a, and A) that are modified for different pressures.

Rate parameters are given for a discrete set of (at least two) different pressures within the pressure range of interest.

- When the pressure P in a simulation is between given pressures (for example, between P_i and P_{i+1}), the rate parameter is calculated by a linear interpolation of $\ln k$ as a function of $\ln P$:

$$\ln k = \ln k_i + \left(\ln k_{i+1} - \ln k_i\right) \frac{\ln P - \ln P_i}{\ln P_{i+1} - \ln P_i} \tag{3315}$$

- When the pressure within a simulation is higher or lower than any of the given pressures, the rate parameters for the highest or lowest pressures, respectively, are used.

Chemical Equilibrium

When the forwards and backwards reaction rates are equal, the reaction is said to have reached equilibrium and no net reaction is observed. The proportions of reactants and products are constant as long as the pressure (for gas-phase reactions), temperature, and concentrations of species are constant.

You can determine the amount of each species that is present at equilibrium using the equilibrium constant, K.

For a general reaction, Eqn. (3290), the equilibrium constant is defined as:

$$K = \frac{(a_C)^{v''_C}(a_D)^{v''_D}\ldots}{(a_A)^{v'_A}(a_B)^{v'_B}\ldots} \tag{3316}$$

where a is the activity of the respective species. The activity of a gas-phase species, for example species A, is defined as:

$$a_A = \frac{[A]}{c^0} \tag{3317}$$

where c^0 is a reference concentration.

At chemical equilibrium:

$$k_f[A]^{v'_A}[B]^{v'_B}\ldots = +k_b[C]^{v''_C}[D]^{v''_D}\ldots \tag{3318}$$

which can also be expressed as:

$$\frac{[C]^{v''_C}[D]^{v''_D}\ldots}{[A]^{v'_A}[B]^{v'_B}\ldots} = \frac{k_f}{k_b} \tag{3319}$$

For reactions in which the concentrations of species is known at equilibrium, the equilibrium constant with respect to concentration K_c can be determined by:

$$K_c = \frac{[C]^{v''_C}[D]^{v''_D}}{[A]^{v'_A}[B]^{v'_B}} \tag{3320}$$

which means:

$$K_c = \frac{k_f}{k_b} \tag{3321}$$

the equilibrium constant with respect to concentration can also be determined from thermodynamic data using Helmholtz free energy, A:

$$K_c = \exp\left(\frac{\Delta_R \overline{A}^0}{R_u T}\right) \tag{3322}$$

or for the equilibrium constant with respect to pressure, using Gibbs free energy, G:

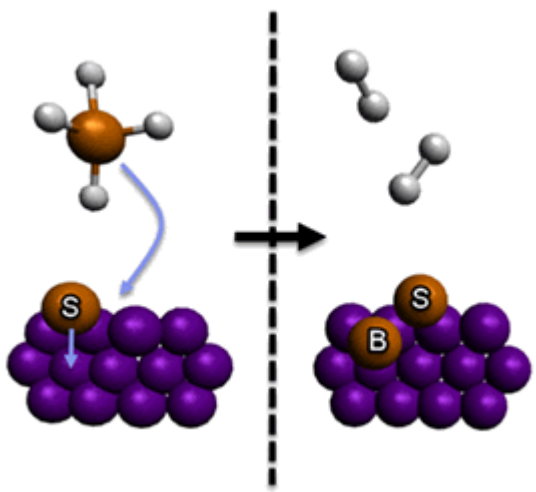
$$K_p = \exp\left(\frac{\Delta_R \overline{G}^0}{R_u T}\right) \tag{3323}$$

Surface Chemistry

There are three possible processes that can occur during surface chemistry reactions:

- Gas-phase molecules colliding with surface molecules and becoming adsorbed. The surface site (the reactive molecules of the surface) becomes part of the bulk and the previous gas-phase molecules form the new surface on top of the bulk.
- Surface reactions between molecules on the surface, or one molecule on the surface and one in the gas, reacting to form new surface, gas phase, or bulk molecules.
- Desorption of a surface molecule—the molecule entering the gas phase after a reaction, or simply by overcoming the bonding energy.

For example, in the diagram below, a gas molecule is adsorbed onto a surface next to a previously adsorbed surface species. The gas molecule is broken up into its constituent atoms. Some of the adsorbed atoms react to form smaller gas molecules which leave the surface. One atom from the gas molecule remains adsorbed as a surface species, and the previously adsorbed surface species becomes a bulk species.



The reaction in the diagram above can be represented by:

$$\mathrm{SiH_{4(G)}} + \mathrm{Si_{(S)}} \rightarrow \mathrm{Si_{(S)}} + \mathrm{Si_{(B)}} + 2\mathrm{H_{2(G)}}$$

Bulk species are not considered reactive but can end up on a surface site if the surface molecules previously covering the bulk species are desorbed or take part in a reaction.

With the Surface-Gas Interaction model, three types of species must be defined:

- Species in the gas phase, G
- Species on the surface at the gas-surface interface, S. Surface reactions can be described as using either open site formalism or atomic site formalism.
- Species in the solid layer—bulk species, B.

The general surface reaction form for reaction j is:

$$\sum_{i=1}^{N_g} \mathrm{g}'_{i,j} G_i + \sum_{i=1}^{N_s} \mathrm{s}'_{i,j} S_i + \sum_{i=1}^{N_b} \mathrm{b}'_{i,j} B_i = \sum_{i=1}^{N_g} g_{i,j''} G_i + \sum_{i=1}^{N_s} s_{i,j''} S_i + \sum_{i=1}^{N_b} b_{i,j''} B_i \tag{3324}$$

where:

- G_i, S_i, and B_i are gas, site, and bulk species.
- N_g, N_s, and N_b are the numbers of gas, site, and bulk species.
- $\mathrm{g}'_{i,j}$, $\mathrm{s}'_{i,j}$, and $\mathrm{b}'_{i,j}$ are stoichiometry coefficients for gas, site, and bulk reactants.
- $g_{i,j''}$, $s_{i,j''}$, and $b_{i,j''}$ are stoichiometry coefficients for gas, site, and bulk products.

The rate of the jth reaction is then:

$$r_j = k_{f,j} \prod_{i=1}^{N_g} \left[G_i\right]_{\mathrm{wall}}^{p_{g(i,j)}} \prod_{i=1}^{N_s} \left[S_i\right]_{\mathrm{wall}}^{p_{s(i,j)}} \tag{3325}$$

where p_g and p_s are rate exponents for reactants, and $[*]_{\mathrm{wall}}$ represents molar concentration on the wall. The forward rate constant $k_{f,j}$ for reaction j is given by an Arrhenius expression. Which, for the simplest type of the reaction, is given by Eqn. (3301) (where k is $k_{f,j}$).

The concentration of surface species, $[S_i]$ (in moles/m²), is determined for the n'th site type, by the site density Γ_n, the fraction of sites that are covered by the i'th species $Z_{S_i,n}$, and the number of sites that a species covers $\sigma_{S_i,n}$.

$$\left[S_i\right] = \sum_n \frac{Z_{S_i,n}\Gamma_n}{\sigma_{S_i,n}} \tag{3326}$$

The sum runs over all different site types.

It is assumed that the reaction rate does not depend on the concentrations of the bulk species.

The net molar rate of change per surface area (production or consumption) of each species i is given by:

$$\hat{r}_{i,\mathrm{gas}} = \sum_{j=1}^{N_{reac}} \left(g_{i,j''} - \mathrm{g}'_{i,j''}\right) r_j \tag{3327}$$

$$\hat{r}_{i,\mathrm{site}} = \sum_{j=1}^{N_{reac}} \left(s_{i,j''} - \mathrm{s}'_{i,j}\right) r_j \tag{3328}$$

$$\hat{r}_{i,\mathrm{bulk}} = \sum_{j=1}^{N_{reac}} \left(b_{i,j''} - \mathrm{b}'_{i,j}\right) r_j \tag{3329}$$

For some surface reactions, the reaction rate is influenced by the surface coverage characteristics of the species that are involved. The arrhenius rate expression, Eqn. (3301), can be modified to create a corrected rate constant as follows:

- **Site Coverage Correction Rate Expression**

  Some surface reactions are characterized by the surface site coverage z_i of species i.

$$k_{f,j} = A_j T^{\beta_j} \exp\left(-\frac{E_a}{R_u T}\right) \prod_{i=1}^{N_s} 10^{\eta_{i,j}} z_i^{\mu_{i,j}} \exp\left(-\frac{\varepsilon_{i,j} Z_i}{R_u T}\right) \tag{3330}$$

where $\eta_{i,j}$, $\mu_{i,j}$, and $\varepsilon_{i,j}$ are the three surface coverage parameters that you can specify for species i and reaction j on the reacting surface. In imported reaction mechanism files, surface coverage modification reactions are indicated by the keyword, `COV`. See Reaction Mechanism Formats.

- **Sticking Coefficient Correction Rate Expression**

  Forward surface reaction rates can be expressed using sticking coefficients or sticking probabilities instead of reaction rate coefficients. Sticking coefficients describe the probability for a species to stick to a surface. The reaction can include one gas phase species and any number of surface species, the reaction rate is proportional to the coverage of all involved surface species. The sticking coefficient s_c is calculated as:

$$s_c = \min\left[1, A_j T^{\beta_j} e^{\frac{-E_{a,j}}{R_u T}}\right] \tag{3331}$$

  and the rate expression is modified as:

$$k_{f,j} = s_c \frac{\prod_{i=1}^{N_s} \sigma_i^{\upsilon'_{i,j}}}{(\Gamma_{tot})^m} \sqrt{\frac{R_u T}{2\pi W_i}} \tag{3332}$$

  Γ_{tot} is the total surface site concentration and m is the sum of all the surface reactant stoichiometric coefficients. W_i is the molecular weight of the gas-phase species i, and σ_i is the number of sites that each surface species occupies—$\upsilon'_{i,j}$ is the reaction order for that species.

- **Motz-Wise Correction Rate Expression**

  The rate expression for the forward reaction using the Motz-Wise correction factor is:

$$k_{f,j} = \left(\frac{s_c}{1 - s_c/2}\right) \frac{\prod_{i=1}^{N_s} \sigma_i^{\upsilon'_{i,j}}}{(\Gamma_{tot})^m} \sqrt{\frac{R_u T}{2\pi W_i}} \tag{3333}$$

  where s_c is calculated by Eqn. (3331).

- **Bohm Correction Rate Expression**

  When simulating reactions which involve positive ions, you can modify the rate constant to include a Bohm velocity correction $\sqrt{\frac{R_u T_e}{W_{ion}}}$ which considers the probability for the reaction to occur:

$$k_{f,j,Bohm} = A_j T^{\beta_j} e^{-E_{a,j}/RT} \frac{\prod_{i=1}^{N_s} \sigma_i^{\upsilon'_{i,j}}}{(\Gamma_{tot})^m} \sqrt{\frac{R_u T_e}{W_{ion}}} \tag{3334}$$

  where T_e is the electron temperature and W_{ion} is the molecular weight of the positive ion.

Particle Chemistry

For reactions of particles, the rate constant k is determined using the Arrhenius equation Eqn. (3301). Since particle chemistry reaction rates consider diffusion alongside kinetics, these calculations are described in the Particle Reactions section of the Theory Guide.

Reactions—Species Transport

Using reacting species transport, all species in the mechanism are convected and diffused through the computational domain, and react using the prescribed chemical mechanism.

The following reacting species transport models are available in STAR-CCM+:

- Complex Chemistry: Useful when chemical kinetic phenomena are important, for example calculation of slowly forming species such as CO or for prediction of ignition and extinction. Suitable for handling many reactions since an ODE solver is used to integrate the chemical source terms for all species over the CFD time-step through adaptive time-step subcycling in each cell.
- Eddy Break-Up (EBU): Useful when only the flame position is of interest. The reaction is limited by the single turbulent mixing time-scale, so EBU is only suitable for mechanisms with a few reaction steps.
- Eddy Contact Micromixing (ECM): Useful for diffusion-limited reactions in liquid-liquid chemistry. The reaction is limited by the single turbulent mixing time-scale, so ECM is only suitable for mechanisms with a few reaction steps.

Contents:

Complex Chemistry
Eddy Break-Up
Thickened Flame
Eddy-Contact Micromixing

Complex Chemistry

In STAR-CCM+, detailed Complex Chemistry is solved using a stiff ODE (Ordinary Differential Equation) solver—either CVODE or DARS-CFD—to integrate the chemical source terms. To consider the turbulence effects on combustion, you can also select either the Laminar Flame Concept (LFC) model or the Eddy Dissipation Concept (EDC) model.

The Complex Chemistry model is suitable for introducing detailed chemistry information to the CFD simulation. This model can solve thousands of reactions among hundreds of species—hence the term complex chemistry. Since an ODE solver is used to integrate the chemical source terms, the Complex Chemistry model can handle stiff reaction systems (reaction systems with a wide range of reaction time scales).

Reaction System

The Complex Chemistry model requires detailed reaction mechanism information about species, reactions, thermodynamics, and transport properties. These details are supplied by complex chemistry definition files that are imported in the Chemkin format using the Complex Chemistry model.

Source Term Definition

The general species transport equation is formulated as follows:

$$\frac{\partial}{\partial t}\rho Y_i + \frac{\partial}{\partial x_j}\left(\rho u_j Y_i + F_{k,j}\right) = \omega_i \tag{3335}$$

where $F_{k,j}$ is the diffusion flux component and source term ω_i is the rate of production of species i.

The operator splitting algorithm takes advantage of the different time scales involved for the chemical reactions and the flow field. The time integration of the chemical state (species mass fractions Y_i and temperature T) is performed in two steps:

- At the beginning of each time-step, the chemical state is integrated in each CFD cell from state $\left(Y_i, T, p\right)^n$ to $\left(Y_i, T, p\right)^*$, accounting only for the chemical source terms:

  $$Y_i^* = Y_i + \int_0^\tau r_k\left(\mathbf{Y}, T, p\right) dt \tag{3336}$$

  where Y_i^* is the mass fraction at the end of a time integration τ with a stiff ODE solver, as calculated in Eqn. (3336). r_k is the reaction rate from Eqn. (3295), $\mathbf{Y}$ is the mass fraction vector, and T is temperature.

  The system of Eqn. (3336) is solved using the stiff ODE solver—either CVODE or DARS-CFD.
- The species transport equation is solved with an explicit reaction source term ω_i for the i'th species given as:

  $$\omega_i = \rho f\left(\frac{Y_i^* - Y_i}{\tau}\right) \tag{3337}$$

  where ρ is the density, f is the mean reaction rate multiplier, Y_i denotes the current mass fraction in the cell. The time integration τ is specified for unsteady simulations, and for steady simulations is taken as the residence time τ_{res} as defined in Eqn. (3266).

For unsteady simulations, it is essential to maintain a low Courant number to ensure that any errors in the operator splitting scheme are small.

Both the Laminar Flame Concept (LFC) and Eddy Dissipation Concept (EDC) consider the turbulence effects on combustion implicitly through the increased turbulent diffusivity that is provided by the turbulence model.

In the case of premixed flames, this increased diffusivity results in a flame thickness that is larger than the laminar flame thickness, and a turbulent flame propagation speed that is faster than the laminar flame speed. The flame thickness and speed is controlled through the turbulent Schmidt and Prandtl numbers (set these equal to each other).

In addition to this implicit turbulence effect through the turbulent diffusivity, the EDC model also considers the "turbulence-chemistry interaction" in the species chemical sources. The mean species source term in the species transport equation for both LFC and TFC is modeled as Eqn. (3337).

- For the LFC model, the mean reaction rate multiplier f in Eqn. (3337) is 1, and the time scale, τ in Eqn. (3337) is the residence time in the cell, which is calculated as Eqn. (3266) multiplied by the CFL Scaling Factor F_{CFL}, which has a default value of 1.
- For the EDC model, the mean reaction rate multiplier f in Eqn. (3337) is calculated as:

$$f = \left(\left[C_l \left(\frac{\upsilon \tau_{turb}}{L^2} \right)^{0.25} \right]^{-3} - 1 \right)^{-1} \tag{3338}$$

where C_l is the fine structure length constant with a default value of 2.1377, υ is the kinematic viscosity, τ_{turb} is the turbulent time-scale, and L is the turbulent length scale.

For the EDC model, the time scale τ in Eqn. (3337), is calculated as:

$$\tau = F_{CFL}\, \tau_{turb} \tag{3339}$$

where τ_{turb} is the turbulent time-scale which is defined by the turbulence model that is used, and F_{CFL} is the CFL Scaling Factor

Eqn. (3336) and Eqn. (3337) represent the average rate of change of the species in the cell over the time-step τ. The LFC time-step is approximately the residence time in the cell Eqn. (3266). The EDC time-step in Eqn. (3339) is close to the Kolmogorov time-scale τ_k (as described in the section, Turbulence Transfer between Phases), which is a model for the time-scale of the smallest turbulent eddy. Since the Kolmogorov time is typically less than the residence time, and the EDC mean reaction rate multiplier f in Eqn. (3338) is less than 1, EDC typically models the effect of turbulence as a reduction in the mean reaction rate compared to LFC.

- The Equilibrium Time Scale method assumes the chemical composition relaxes to the local equilibrium composition at a time scale that is determined by flow and chemistry time scales. The equilibrium composition is computed using ISAT.

  The chemistry source term is computed based on the specified fuel species:

$$\omega_i = \rho \frac{Y_i - Y_i^{eq}}{\tau_{char}} \tag{3340}$$

  where:

  - Y_i is the mass fraction for species i.
  - Y_i^{eq} is the local instantaneous thermodynamic equilibrium mass fraction for species i.
  - τ_{char} is a characteristic time scale.

$$\tau_{char} = \tau_{flow} + \tau_{chem} \tag{3341}$$

    where:

    - $\tau_{flow} = k/\left(\varepsilon c_{flow}\right)$, where:
      - ε is turbulent dissipation rate.
      - c_{flow} is a turbulent rate constant with a default value of 4.

      For laminar flow, $\tau_{flow} = \Delta/\left(U c_{flow}\right)$, where:

      - Δ is the grid size.
      - U is the local velocity.
    - $\tau_{chem} = c_{chem} min_f \left(\frac{Y_f}{r_f} \right)$, where:

- c_{chem} is a model constant with a default value of 2.
- f is an index for a user-specified list of fuel species.
- min_f is taking the minimum of all the user-specified fuel species.
- Y_f is the fuel species mass fraction in a CFD cell.
- r_f is the net reaction rate of species f.

Solution Procedure

In STAR-CCM+, the reacting species transport equations are solved using the ODE solver with the operator splitting algorithm to find an average reaction rate to remove stiffness. When running with the Steady State model, an artificial chemistry time step is introduced which is based on convection and diffusion fluxes in that cell.

The time step for integration of the chemistry in a steady state simulation τ for each cell p is determined by:

$$\tau = \frac{V_p \rho_p F_{CFL}}{A_p} \tag{3342}$$

$$= \frac{m}{\dot{m}} F_{CFL} \tag{3343}$$

$$= \tau_{res} F_{CFL} \tag{3344}$$

which is a function of the cell volume V_p, density ρ_p, CFL scaling factor F_{CFL}, and central coefficient A_p which results from discretization and linearization. Eqn. (3342) is approximately the residence time Eqn. (3266) in the cell multiplied by the CFL scaling factor F_{CFL}.

The Complex Chemistry model uses either the CVODE solver or the DARS-CFD solver to integrate the stiff chemistry over a time-step.

The evolution of mean species mass fractions and temperature in a given CFD cell is calculated by integrating at the cell conditions using Jacobian matrices to solve the ordinary differential equation:

$$Y_i = Y_i\left(t_0\right) + \int_0^{\tau_{res}} \dot{\omega}_t \, dt \tag{3345}$$

The transport equations that are solved for a reacting flow contain the chemical source term, ω.

$$\frac{d\left(Y_i\right)}{dt} = \dot{\omega}_n\left(Y_i, T, p\right) \tag{3346}$$

Where Y_i is the mass fraction of species i. The Jacobian J_ω is defined as:

$$\mathrm{J}_\omega = \frac{\partial \omega}{\partial Y} \tag{3347}$$

For n species, the Jacobian matrix is written as:

$$J_\omega = \begin{bmatrix} \frac{\partial \omega_1}{\partial Y_1} & \frac{\partial \omega_1}{\partial Y_2} & \cdots & \frac{\partial \omega_1}{\partial Y_n} \\ \frac{\partial \omega_2}{\partial Y_1} & \frac{\partial \omega_2}{\partial Y_2} & \cdots & \frac{\partial \omega_2}{\partial Y_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \omega_n}{\partial Y_1} & \frac{\partial \omega_n}{\partial Y_2} & \cdots & \frac{\partial \omega_n}{\partial Y_n} \end{bmatrix} \tag{3348}$$

A numerical Jacobian describes a Jacobian that is derived using the finite differences method for each species in the system. Since the chemical source term, ω, is calculated repeatedly as the ODE solver iterates, the cost of solving the ODE is expensive.

An analytical Jacobian is calculated with the chain rule to obtain an analytical expression for J_ω. STAR-CCM+ uses the same analytical Jacobian expression for every mechanism, however, the expression uses different coefficients that are determined by the mechanism.

Since this process is computationally expensive, ISAT, Clustering, or Dynamic Mechanism Reduction are used to reduce the expense and reach a converged solution faster.

ISAT

In-Situ Adaptive Tabulation (ISAT) dynamically tabulates functions that are computationally expensive to evaluate, and interpolates many times after tabulation [662]. The ISAT method approximates the solution of the ODE (Eqn. (3336)) for a given initial condition through:

- Thermodynamic variable (temperature or total enthalpy).
- Composition variable (mass fraction or molar number).
- Possibly pressure and time step (depending on the type of the simulation).

These values constitute an independent variable or a query point (ϕ). The solution f of the ODE equation (Eqn. (3336)) is called reaction mapping.

The ISAT method starts with an empty table and gradually adds points to it. To solve the first ODE equation for ϕ_0 (a first query), ISAT does the following:

- It calls the ODE solver (DARS-CFD in this case) to calculate f_0.
- It creates an entry and adds it to the table.

Every table entry consists of:

- An independent variable ϕ
- A reaction mapping variable f
- A mapping gradient matrix $A = \frac{\partial f}{\partial \phi}$
- An Ellipsoid Of Accuracy (EOA), derived by a singular value decomposition of the mapping gradient.

For all subsequent queries, ISAT does the following for query point ϕ_n:

- It traverses the table and finds a suitable point to check for the possible approximation.
- For the suitable point ϕ_t it checks the value: $f_q = f_t + A_t(\phi_n - \phi_t)$

The following two scenarios can happen:

- Either f_q lies in the EOA, which means that the value f_q approximates, within the user-specified tolerance, the true value of $f(\phi_n))$. In other words, query ϕ_n is retrieved.
- Or f_q lies outside of the EOA. In this case, the true value of $f(\phi_n))$ is calculated and the difference between the two values is deduced. If this difference is smaller than the user-defined tolerance, this means that the EOA at point ϕ_n is too strict. In this case, the ellipsoid is grown to accept point f_q. If the difference is higher than the specified tolerance, the point is added to the table.

Points in the table are stored in the leaves of the binary space partition tree. This tree divides the region in the$n$-dimensional space (n being the dimension of the query point) currently occupied by the points into smaller subregions for faster search. When a new query point is added to the tree, an original leaf becomes now two leafs, split by a hyperplane. One new leaf contains the old point, which lies on one side of the hyperplane. The second new leaf, which contains the new query point, lies on the other side of the hyperplane.

Therefore the following properties characterize the ISAT method:

- Number of retrieves: the number of queries for which the reaction mapping was approximated
- Number of grows: the number of queries for which the reaction mapping was calculated but the point was not added to the table because the EOA of the old point in the table was grown
- Number of adds: the number of queries for which the reaction mapping was calculated and the point was added to the table

Growing of the EOAs can be deactivated through a property within the ISAT approximation method. Since EOA is only an approximation of the actual region of accuracy, growing of the EOA can introduce errors in the approximation table. Furthermore, it is possible that adding a point to the table will cover more future queries than growing the existing EOA.

Clustering

Clustering reduces the computational expense of complex chemistry calculations by averaging together cells with similar chemical compositions, integrating the reduced ODE set, and then interpolating the clusters back to the cells.

A low-dimensional, uniformly spaced clustering grid is formed from a few composition variables. By default, STAR-CCM+ uses two clustering variables for unsteady simulations—equivalence ratio and temperature. For steady-state simulations, the log of the chemical time scale is also used as an additional clustering variable.

The clustering tolerance determines the distance of the clustering composition variable between the clustering grid points. For relative tolerances ε_{rel} the number of grid points for the clustering composition variable ϕ is:

$$\frac{1}{\varepsilon_{rel}} \tag{3349}$$

For absolute tolerances ε_{abs}, the number of grid points for the clustering composition variable ϕ is:

$$\frac{\phi_{\max} - \phi_{\min}}{\varepsilon_{abs}} \tag{3350}$$

where $\phi_{\max}$ and $\phi_{\min}$ represent the maximum and minimum values of ϕ in the computational domain.

Following Babajimopolous [647], the clustering equivalence ratio is defined as:

$$\phi = \frac{2C^* + \frac{H^*}{2}}{O^*} \tag{3351}$$

where:

- C^* is the number of C atoms in all species except CO_2, divided by the species molecular weight.
- H^* is the number of H atoms in all species except H_2O, divided by the species molecular weight.
- O^* is the number of O atoms in all species except CO_2 and H_2O, divided by the species molecular weight.

In addition to the default clustering variables, you can select additional clustering variables such as mixture fraction and species. The clustering mixture fraction is defined as the elemental mass fraction of Hydrogen and Oxygen atoms in the mixture.

Within each clustering grid point, species mass fractions, energy, pressure—and for steady-state, the chemistry time-step—are cell mass-weighted averaged to form the cluster mean. This composition (Y_i^* in Eqn. (3336)) is integrated with the selected ODE solver—as in Eqn. (3336). The source term is then applied to all cells in the cluster.

Dynamic Mechanism Reduction

Dynamic Mechanism Reduction allows the CVODE solver in STAR-CCM+ to solve for a reduced number of species in a full chemical mechanism. The Directed Relation Graph (DRG) algorithm [652] considers the reaction rates of all species in a cell at each time step and removes species from the mechanism if they do not change substantially over the course of that time step.

When deciding if a species can be removed from the full chemical mechanism, the DRG algorithm considers how significant an influence that species has on the rate of production of other species.

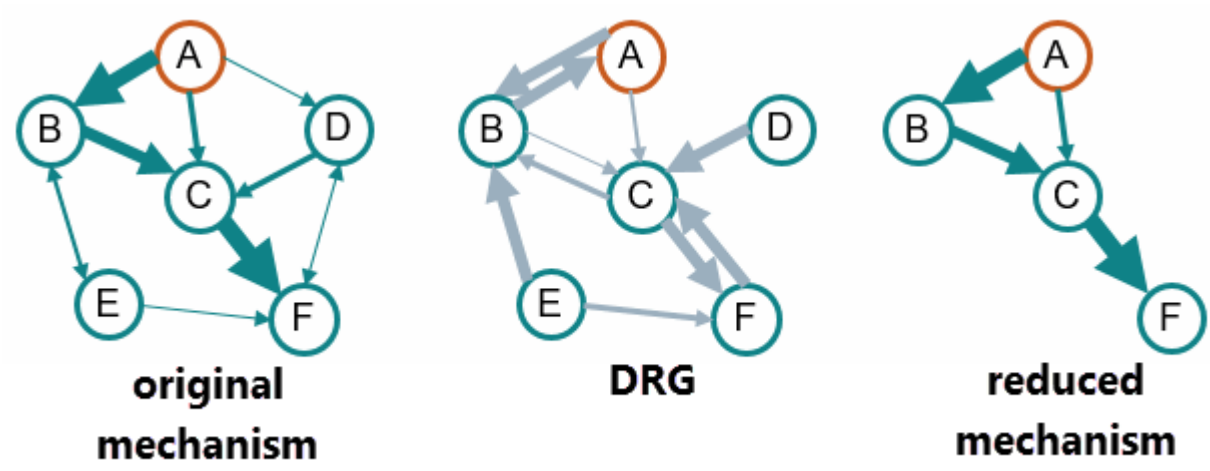
For example, if species A is identified as a target species to be retained in the mechanism, the effect of removing species B on the rate of production of species A is determined as follows:

$$r_{AB} = \frac{\max_i \left| \nu_{A,i} \omega_i \delta_{B,i} \right|}{\max_i \left| \nu_{A,i} \omega_i \right|} \tag{3352}$$

- if the ith elementary reaction involves species B, $\delta_{B,i} = 1$
- if the ith elementary reaction does not involve species B, $\delta_{B,i} = 0$

When the normalized contribution of species B to the production rate of species A (r_{AB}) is greater than the error tolerance ε that is specified, the DRG algorithm also retains species B. Species A and B are said to be strongly coupled and removal of species B would impact the production rate of species A significantly.

Each node (A to F in the diagram below) in a DRG represents a species in the full chemical mechanism.



All arrows in the diagram above represent directed edges—these are edges in the DRG that exist between species with a dependence. The arrows show the direction of dependence of one species on another—the width indicates the strength of the dependence. Since A is dependant on B and C, and C is dependant on F, the group of species A, B, C, and F must be maintained to correctly calculate A. The remaining species can be removed as they do not have a significant effect on the production rate of A.

Eddy Break-Up

The Eddy Break-Up model is intended for modeling reacting flow with fast chemistry where the reaction rate is determined by the rate at which turbulence can mix the reactants and heat.

Reaction System

EBU models characterize the reacting flow system by a specified number of species and chemical reactions. The standard EBU model assumes that the reaction rate is determined solely by the turbulent mixing time scale. However, the hybrid kinetics and combined time-scale versions of the EBU model assume that the rate of reaction is also affected by the kinetic reaction rate from finite rate kinetics.

Source Term Definition

The kinetic reaction source term for each species is obtained from multiplying the rate from Eqn. (3299) by the molecular weight W_i. The species source term S_i is modeled as a function of chemical reaction rates and characteristic time scales. The specific formulation of the source term and time scales depend on the EBU model selection.

Standard EBU

The Standard EBU model assumes that when species are mixed, they are immediately burnt. The chemical source term is calculated from the mixing time scale.

The reaction rate is modeled through an expression that accounts for the turbulent micromixing process. For a reaction of the form Eqn. (3290), the rate of fuel depletion is assumed to be:

$$r_F = -\frac{\rho}{W_F}\left(\frac{1}{\tau_{turb}}\right)A_{ebu}\min\left[Y_F, \frac{Y_O}{s_O}\right] \quad \text{moles}/\left(\text{m}^3\text{s}\right) \tag{3353}$$

or, when the EBU coefficients option, **Use Products for Rate**, is activated:

$$r_F = r_{F,mix} = -\frac{\rho}{W_F}\left(\frac{1}{\tau_{turb}}\right)A_{ebu}\min\left[Y_F, \frac{Y_O}{s_O}, B_{ebu}\left(\frac{Y_{P1}}{s_{P1}} + \frac{Y_{P2}}{s_{P2}} + \ldots + \frac{Y_{Pj}}{s_{Pj}}\right)\right] \tag{3354}$$

In the above equation,

$$s_O \equiv \upsilon_O m_O / \upsilon_F W_F \tag{3355}$$

$$s_{Pi} \equiv \left|\upsilon_{Pi}\right| m_{Pi} / \upsilon_F W_F \tag{3356}$$

when using the K-Epsilon model:

$$\tau_{turb} = \frac{k}{\varepsilon} \tag{3357}$$

where k is the turbulent kinetic energy, and ε is its dissipation rate. W_F is the molecular weight of fuel in a cell. The min operator on the right-hand side of Eqn. (3353) indicates that the concentration of the limiting reactant is used to determine a mass fraction scale when calculating the reactant consumption rate. Eqn. (3353) essentially states that the integrated micromixing rate is proportional to the mean (macroscopic) concentration of the limiting reactant, divided by the time scale of the large eddies. Eqn. (3354) is an optional modification to Eqn. (3353) for premixed flames in which fuel and oxidizer are already mixed at the molecular scale. The reaction rate is determined by the rate at which the products are mixed with reactants.

Hybrid EBU

The Hybrid EBU model assumes that the minimum value of mixing and chemical kinetic time scale is rate-limiting and calculates the source term using Eqn. (3299) multiplied by the molecular weight W_i of species i.

This model is expressed as:

$$r_i = -\min\left(\left|r_{i,kin}\left(\rho, Y_1, Y_2, Y_3, \ldots, Y_N, \text{T}\right)\right|, \left|r_{i,mix}\right|\right) \tag{3358}$$

Combined Time-Scale EBU

The Combined Time-Scale EBU model combines the mixing time scale and chemical time scale to calculate the source term using Eqn. (3337). The turbulent mixing time scale τ_{mix} used in the standard EBU model, decreases with decreasing distance from solid surfaces. This leads to over-prediction of reaction rates in near-wall regions. To alleviate this problem, the mixing time scale is augmented by a time scale that is derived from the chemical reaction rate from finite-rate kinetics, Eqn. (3299), defined as:

$$\tau_{kin} = \frac{\rho Y_F}{m_F}\left(\frac{1}{\left|r_{i,kin}\left(\overline{\rho}, \overline{Y}_1, \overline{Y}_2, \overline{Y}_3, \ldots, \overline{Y}_N, \overline{T}\right)\right|}\right) \tag{3359}$$

As a result, the time scale, τ, that is used to calculate reaction rate (Eqn. (3353) or Eqn. (3354)) is calculated by:

$$\tau = \tau_{mix} + \tau_{kin} \tag{3360}$$

Therefore, the reaction rate that is predicted by the combined time-scale variant is smaller than that from the standard EBU model.

Kinetics Only EBU

The Kinetics Only EBU model assumes that the reaction rate is dictated by finite-rate chemical kinetics. Therefore, the source term is calculated by Eqn. (3337) using the reaction rate that is calculated by Eqn. (3299).

Solution Procedure

The Eddy Break up model uses the source term in Eqn. (3337) directly in the transport equations—with no subcycling. Therefore, this model is suitable for a mechanism with a limited number of reactions that are not stiff.

For transient (unsteady) simulations, the chemistry is calculated over the duration of the time-step. For steady state simulations, the chemistry is calculated over the cell residence time (chemistry time step) τ_{res} in Eqn. (3266).

If source term limiting is activated, the reaction rate r_F in Eqn. (3353) or Eqn. (3354) is limited to make sure that the amount of reactants in a cell is less than or equal to the amount of reactants that were in the cell before the reaction started:

$$r_{F,\lim} = \min\left(r_F, \frac{m\ \ Y_F}{\tau_{res}}\right) \tag{3361}$$

Where m is the total mass in the cell.

Limiting source terms prevents the reaction rate from driving the fuel mass fraction Y_F and oxidizer mass fraction Y_O to a negative value, which is not physically possible.

Thickened Flame

The Thickened Flame model artificially thickens thin flames to resolve them on the CFD mesh.

Reaction System

In a large-eddy (LES) simulation of turbulent reactive flows, the subgrid-scale filtered balance equation for chemical species has the form Eqn. (3362).

In practice, the computational mesh is too coarse to resolve the actual flame, which is usually around 0.1 to 1.0 mm thick. To capture the correct flame speed, about fifteen cells are required within the flame.

One approach to overcome this difficulty is known as the Thickened Flame model. The thickness of the computed flame front structure is artificially increased to the size which can be resolved on a LES filter, while kinetic constants are adjusted to keep the laminar flame speed unchanged.

Source Term Definition

The artificial thickness can be achieved by multiplying species and heat diffusion coefficients by a factor F, and decreasing the reaction rate by the same factor F. This is based on the well-established theories of laminar premixed flame that laminar flame speed s_L^0 and the laminar flame thickness δ_L^0 scale as $s_L^0 \alpha \sqrt{D\dot{\omega}}$ and $\delta_L^0 \alpha \left(D / s_L^0\right)$, where D is the diffusion coefficient and $\dot{\omega}$ the reaction rate. For gas turbines to resolve the flame on a LES grid, F needs to be approximately 10 to 100.

The thickening procedure above reduced the Damkohler number D_α to D_α/F, which makes the flame less sensitive to the small turbulent motions, that is, eddies smaller than $F/\ \Delta_l^0$. This subgrid scale effect can be incorporated in the modeling by using an efficiency factor E_F evaluated from the subgrid scale wrinkling factor model. Then the underestimation of the flame front wrinkling by the thickened flame approach can be corrected by increasing the flame speed by the factor E_F.

While the thickening factor F can be constant, dynamic forms of the thickened flame model have been proposed. There the thickening factor F is dynamically determined based on time and spatial location. Usually it takes large values inside the reaction zone where flame structure is small, while it is set to be unity away from the flame front. A sensor Ω is usually used to detect the presence of the reaction zone.

The resulting species and enthalpy equation is:

$$\frac{\partial(\rho)Y_i}{\partial t} + \frac{\partial(\rho)Y_i u_i}{\partial x_i} = \frac{\partial}{\partial x_i}\left[E_F F\left(D + \frac{\mu_t}{\sigma_k}\right)\frac{\partial Y_i}{\partial x_i}\right] + \frac{E_F(\dot{\omega}_i)}{F} \tag{3362}$$

There are two methods of computing the dynamic thickening factor F for the Thickened Flame model:

- Arrhenius expression
- Reaction progress variable expression

Arrhenius Expression

For a simple one–step chemistry scheme:

$$\gamma_F F + \gamma_O O = \gamma_P P \tag{3363}$$

The sensor Ω is formulated as:

$$\Omega = Y_F^{\gamma_F} Y_O^{\gamma_O} \exp\left(-\Gamma \frac{T_a}{T}\right) \tag{3364}$$

Γ is a modeling parameter $(\Gamma < 1)$.

The thickening factor F [695] takes this form:

$$F = 1 + \left(F_{max} - 1\right)\tanh\left(\beta \frac{\Omega}{\Omega_{max}}\right) \tag{3365}$$

Reaction Progress Variable Expression

The Arrhenius expression is only valid for one-step chemistry and is hard to extend to multi-step mechanisms. Durand and Polifke [694] proposed a different expression in which the sensor Ω is related to the reaction progress variable. When c is close to 0 or 1, it is relatively far from the flame front, so the flame thickening process is not needed and the thickening factor F is close to 1, while the thickening factor takes the maximum value F_{max} at the location where $c = 1$. It is computed using the following form:

$$F = 1 + \left(F_{max} - 1\right)\Omega(c) \tag{3366}$$

$$\Omega(c) = 16\left[c(1-c)\right]^2 \tag{3367}$$

$$c = 1 - Y_F/\left(Y_F(z)\right) \tag{3368}$$

Here the progress variable is evaluated based on the fuel mass fraction Y_F in the domain and the unburnt fuel mass fraction $Y_F(Z)$ that is solely dependent on the mixture fraction Z.

In addition, the effective diffusivity D_{loc} is also represented in terms of the sensor function $\Omega(c)$ as follows:

$$D_{loc} = DEF + \left(1-\Omega(c)\right)\frac{\mu_t}{\sigma_k} \tag{3369}$$

and the species transport equation is modified to:

$$\frac{\partial(\rho)Y_i}{\partial t} + \frac{\partial(\rho)Y_i u_i}{\partial x_i} = \frac{\partial}{\partial x_i}\left[E_F F D_{loc}\frac{\partial Y_i}{\partial x_i}\right] + \frac{E_F(\dot{\omega}_i)}{F} \tag{3370}$$

Solution Procedure

Efficiency Function

There are two methods of computing the efficiency function E_F for the Thickened Flame model:

- **Wrinkling Factor Ratio Model**
  A dimensionless wrinkling factor Ξ is defined as the ratio of the flame surface to its projection in the direction of propagation. Based on DNS data and spectral analysis, Colin et al. [693] proposed the following expression for modeling the wrinkling factor:

$$\Xi = 1 + \beta\frac{\mathrm{u}'_{\Delta_e}}{s_L^0}\Gamma\left(\frac{\Delta_e}{\delta_L^0}, \frac{\mathrm{u}'_{\Delta_e}}{s_L^0}\right) \tag{3371}$$

  Here u'_{Δ_e} is the local subgrid scale turbulent velocity, s_L^0 is the laminar flame speed, Δ_e is the local filter size, and δ_L^0 is the laminar flame thickness.

  β is computed as:

$$\beta = \frac{2\ln 2}{3c_{ms}\left(Re_t^{1/2}-1\right)}, c_{ms} = 0.28, Re_t = \frac{\mathrm{u}'l_t}{\mu} \tag{3372}$$

  where Re_t is the turbulent Reynolds number.

  The function Γ represents the integration of the effective strain rate that is induced by all scales affected due to artificial thickening, and is estimated as:

$$\Gamma\left(\frac{\Delta_e}{\delta_L^0}, \frac{\mathrm{u}'_{\Delta_e}}{s_L^0}\right) = 0.75\exp\left[-1.2\left(\frac{\mathrm{u}'_{\Delta_e}}{s_L^0}\right)^{-0.3}\right]\left(\frac{\Delta_e}{\delta_L^0}\right)^{2/3} \tag{3373}$$

  Finally, the efficiency function takes the following form as defined by the ratio between the wrinkling factor Ξ, of laminar flame $\left(\delta_L = \delta_L^0\right)$ to thickened flame $\left(\delta_L = \delta_L^l\right)$:

$$E = \frac{\Xi|_{\delta_L = \delta_L^0}}{\Xi|_{\delta_L = \delta_L^l}} \geq 1 \tag{3374}$$

- **Power-Law Flame-Wrinkling Model**
  Charlette et al. [692] proposed a formulation for estimating the efficiency function which relates the flame surface area to a cutoff length and limits the wrinkling at the smallest length-scales of the flame. Based on the asymptotic analysis, the efficiency function is evaluated using a power-law expression:

$$E = \left(1 + \min\left[\frac{\Delta_e}{\delta_L^0}, \Gamma \frac{u'_{\Delta_e}}{s_L^0}\right]\right)^{\alpha} \tag{3375}$$

where Γ is defined as:

$$\Gamma\left(\frac{\Delta_e}{\delta_L^0}, \frac{u'_{\Delta_e}}{s_L^0}, Re_\Delta\right) = \left[\left(\left(f_u^{-a} + f_\Delta^{-a}\right)^{-1/a}\right)^{-b} + f_{Re}^{-b}\right]^{-1/b} \tag{3376}$$

$$f_u = 4\left(\frac{27C_k}{110}\right)\left(\frac{18C_k}{55}\right)\left(\frac{u'_{\Delta_e}}{s_L^0}\right)^2 \tag{3377}$$

$$f_\Delta = \left[\frac{27C_k\pi^{4/3}}{110} \times \left(\left(\frac{\Delta_e}{\delta_L^0}\right)^{4/3} - 1\right)\right]^{1/2} \tag{3378}$$

$$f_{Re} = \left[\frac{9}{55}\exp\left(-\frac{3}{2}C_k\pi^{4/3}Re_\Delta^{-1}\right)\right]^{1/2} \times Re_\Delta^{1/2} \tag{3379}$$

The constants a, b, and C_k control the sharpness of the transitions between the asymptotic behaviors. The suggested values are $b = 1.4$, $C_k = 1.5$:

$$a = 0.6 + 0.2\exp\left[-0.1\left(u'_{\Delta_e}/s_L^0\right)\right] - 0.20\exp\left[-0.01\left(\Delta_e/\delta_L^0\right)\right], Re_\Delta = 4\frac{\Delta_e}{\delta_L^0}\frac{u'_{\Delta_e}}{s_L^0} \tag{3380}$$

Currently, $\alpha = 0.5$ is used, resulting in the non-dynamic formulation.

Laminar Flame Thickness

The Laminar Flame Speed Relation method computes the laminar flame thickness as follows, [691]:

$$\delta_L^0 = 2\frac{\mu_b}{\Pr \rho_u s_L^0} \tag{3381}$$

Here $\Pr$ is the laminar Prandtl number of the burnt gas, assumed to be constant and the default value is 0.9. μ_b is the molecular viscosity of the burnt gas calculated from Sutherland's law:

$$\mu_b = \frac{\mu_1 T_b^{1.5}}{\mu_2 + T_b} \tag{3382}$$

where μ_1 and μ_2 are coefficients with the default values of 1.457e-6$\mathrm{kgm}^{-2}K^{-1.5}$ and 110K. T_b is the burnt gas temperature.

Eddy-Contact Micromixing

Liquid-liquid reactions can be modeled using Eddy Contact Micromixing in which the reaction rate is determined by an expression that accounts for the molecular micromixing process. Liquids generally have

small molecular diffusivity, and reaction is limited by the rate that reactants can diffuse to the liquid-liquid interface. The ECM model models this effect by increasing the time-scale for reaction.

Reaction System

The Eddy-Contact Micromixing model uses a reaction system similar to that of the Eddy Break-Up model. See Eddy Break-Up: Reaction System.

Source Term Definition

The source terms for Eddy-Contact Micromixing are calculated from Eqn. (3337) using reaction rates that are calculated differently depending which of the following methods are used.

For a reaction of the form Eqn. (3290):

- With standard eddy contact micromixing (convection limited), the rate of fuel depletion is assumed to be Eqn. (3353).

  In which the source term, S_O is Eqn. (3355).

  The mixing time scale τ_{turb} in Eqn. (3353) is calculated by one of two options:

  - **Kolmogorov Method**

$$\frac{1}{\tau_{turb}} = A\sqrt{\frac{\varepsilon}{\nu}} \tag{3383}$$

    Where ε is the turbulent dissipation rate, τ_{turb} is the time scale, ν is the kinematic viscosity, and A is a constant (mixing coefficient) with a default value of 0.04.

  - **Classical Scalar Dissipation**

$$\frac{1}{\tau_{turb}} = A\frac{2 + Sc^{-1}}{2}\left[\frac{1}{2}\frac{k}{\varepsilon} + \frac{1}{2}\ln\left(Sc\right)\sqrt{\frac{\nu}{\varepsilon}}\right]^{-1} \tag{3384}$$

    Where k is the turbulent kinetic energy, A is a constant (mixing coefficient) with a default value of 1.0, and Sc is the Schmidt number. Unlike gases, for liquids the molecular mixing is slow, which means that $Sc \gg 1$.

- With no micromixing (Kinetics Only), the rate of fuel depletion is assumed to be dictated by finite-rate chemical kinetics. Therefore, the source term is calculated by Eqn. (3337) using the reaction rate that is calculated by Eqn. (3299).

Solution Procedure

The Eddy-Contact Micromixing model solves liquid-liquid reactions in the same way as Eddy Break-Up reactions are solved. See Eddy Break-Up: Solution Procedure.

Gas Phase Combustion—Flamelet Models

To reduce computational expense in combustion simulations, STAR-CCM+ can precompute reactions for representative scenarios and tabulate the relevant quantities. A turbulent flame can be approximated as an ensemble of laminar flamelets.

The term flamelet is used to describe a basic 0D or 1D laminar flame geometry. When flamelets are calculated, the detailed thermo-chemistry of the temperature and species within the flame is parameterized by two or more variables.

Each flamelet model in STAR-CCM+ makes a different assumption about the composition of the flamelets:

- The Chemical Equilibrium (CE) model assumes that the turbulent flow is in chemical equilibrium, hence chemistry is very fast compared to flow and mixing. This corresponds to the zero strain of the Steady Laminar Flamelet (SLF) model, or the burnt state of the Flamelet Generated Manifold (FGM) model. The thermo-chemical state is parameterized by mixture fraction Z and enthalpy h.
- The SLF model embeds steady diffusion flamelets in the turbulent flow. The assumption is that chemistry is fast compared to flow and mixing, and hence the flame is thin. The thermo-chemical state is parameterized by mixture fraction Z, and the scalar dissipation rate χ.
- The FGM model assumes that the thermo-chemical states in a turbulent flame are similar to those in a laminar flame. The thermo-chemical state is be parameterized by mixture fraction Z, reaction progress c, and enthalpy h. Since there is only one reacting variable (reaction progress, c), only one chemical time scale is represented—of the heat release reactions.

The CE model and the SLF model are limited to modeling fast chemistry without ignition and extinction, however, the FGM model can be used in any combustor.

For premixed and partially-premixed systems, there are flame fronts which move at a turbulent flame speed. Therefore, for the flamelet models that cannot capture the flame front, STAR-CCM+ provides additional flame propogation models.

Contents:

Mixture Fraction
Progress Variable
Flamelet Tables
Chemical Equilibrium
Steady Laminar Flamelet
Flamelet Generated Manifold
Flame Propagation

Mixture Fraction

The mixture fraction is the elemental mass fraction that originated from the fuel stream. Mixture fraction is a conserved scalar that is 1 for the fuel stream and 0 for the oxidizer stream.

For the flamelet methods, STAR-CCM+ solves a transport equation for mixture fraction and uses it to obtain species mass fractions.

Mixture Fraction

The mixture fraction Z is calculated using:

$$Z = \frac{m_f}{m_f + m_{ox}} \tag{3385}$$

where m_f is the total mass of all elements that originate from the fuel stream at any spatial location and m_{ox} is the total mass of all elements that originate from the oxidizer stream.

For simulations with a third stream that participates in a reaction m_{sec} the mixture fraction of the fuel stream is defined as:

$$Z = \frac{m_f}{m_f + m_{ox} + m_{\text{sec}}} \tag{3386}$$

and the mixture fraction of the third stream is defined as:

$$Z_{\text{sec}} = \frac{m_{\text{sec}}}{m_f + m_{ox} + m_{\text{sec}}} \tag{3387}$$

If an inert (unreactive) stream is also included, the mixture fraction of the inert stream Z_i is defined as:

$$Z_i = \frac{m_i}{m_f + m_{ox} + m_i + m_{\text{sec}}} \tag{3388}$$

When a transport equation is solved for the mixture fraction Z, the mean mixture fraction within a cell is used $\overline{Z}$, since the mixture fraction at the cell centre is not necessarily representative of the entire cell volume. Z is transported by convection and diffusion and there is no production when there is no phase change. For steady-state flows, there is no accumulation, so the transient term is zero.

$$\frac{\partial}{\partial t}\left(\rho Z\right) + \nabla \bullet \left[\rho V - \left(\rho D_f + \frac{\mu_t}{Sc_Z}\right)\nabla Z\right] = \dot{\omega}_{spray} \tag{3389}$$

To calculate averaged values of reaction state-space variables where the turbulent fluctuations at any spatial location need to be accounted for, an equation is derived for the second moment (mixture fraction variance) Z_{var}.

Mixture Fraction Variance

The equation that is derived for the mixture fraction variance is:

$$\frac{\partial}{\partial t}\left(\rho Z_{\text{var}}\right) + \nabla\cdot\left(\rho \mathbf{v} Z_{\text{var}} - \frac{\mu_t}{Sc_g}\nabla Z_{\text{var}}\right) = \frac{2\mu_t}{Sc_g}(\nabla Z)^2 - C_d \rho \frac{\varepsilon}{k} Z_{\text{var}} \tag{3390}$$

where the mixture fraction variance Z_{var} of each scaled mixture fraction is calculated as:

$$Z_{\text{var}} = \overline{\left(Z - \overline{Z}\right)^2} \tag{3391}$$

Sc_g is the turbulent Schmidt number for the mixture fraction variance and C_d is a scalar dissipation constant.

Inert Stream

The presence of an inert stream impacts the mixture fraction properties since the inert stream dilutes the overall mixture. When transport equations are solved for the mixture fraction in the presence of an inert stream, the inert fraction Z_I and the active fuel stream fraction Z are considered for each inlet. This transport equation is given by:

$$\frac{\partial}{\partial t}\left(\rho Z_I\right) + \nabla \bullet \left[\rho V - \left(\rho D_f + \frac{\mu_t}{Sc_g}\right)\nabla Z_I\right] = \dot{\omega}_{spray} \tag{3392}$$

further explanation regarding the flamelet table lookup is detailed below in the Tabulated Method section.

Progress Variable

Within a combustion system, combustion takes place in a thin layer that divides the whole domain between unburnt and fully burnt mixtures. By defining a quantity that represents this distinction in each computational cell, it is possible to solve a transport equation that tracks the progress of transition from the unburnt to fully burnt states.

The progress variable c indicates whether the chemical state within a cell represents the unburnt conditions ($c = 0$), burnt conditions ($c = 1$), or something in-between ($0 < c < 1$).

A transport equation for c is defined as:

$$\frac{\partial}{\partial t}(\rho c) + \nabla\cdot\left(\rho \mathbf{v} c\right) - \nabla\cdot\left(\Gamma_c \nabla c\right) = \dot{\omega}_c \tag{3393}$$

where ρ is the mixture density and Γ_c is a diffusion coefficient which includes laminar and turbulent contributions.

The source term $\dot{\omega}_c$ is calculated depending on the context within which c is solved. Other sections (specific to each combustion model) provide these definitions.

Flamelet Tables

To reduce the time that STAR-CCM+ spends in computing reaction outcomes during a simulation, STAR-CCM+ generates lookup tables before the simulation starts. These lookup tables provide reaction products and mixture properties for a chosen range of states.

Independent Variables

Within the flamelet tables, independent variables define the dimensions on which the table is constructed. The method by which the chemistry is computed determines the set of independent variables.

For each independent variable, STAR-CCM+ accepts a range of points at which to compute chemical states, along with conditions of temperature and pressure. The total number of states stored in the table therefore becomes

$$N_{states} = \prod_i \left(n_i\right) \tag{3394}$$

where n_i is the number of points in each dimension, i.

Reaction outcomes are calculated using the method that applies to the chosen model. For example, when FGM is chosen, STAR-CCM+ solves the system of ODEs that apply to the reaction mechanism. Currently, reactions are computed for a single pressure and at the specified initial temperature.

For each of the methods, the mixture fraction independent variable is plotted on the Z (mixture fraction) grid of the flamelet table. The Mixture Fraction Grid Stretch Factor allows you to define how the points are distributed along the Z grid. A value of 1.0 distributes the points uniformly from $Z = 0$ to the stoichiometric mixture fraction Z_{st}. Below 1.0, points are moved towards $Z = 0$, and above 1.0, points are clustered around each side of Z_{st}. The clustering algorithm expands grids points away from Zst using a geometric sequence. That is, in the direction away from Z_{st}, each subsequent cell length is multiplied by this factor.

When using adaptive gridding, STAR-CCM+ starts to generate the flamelet table using an initial coarse mesh that is created with the pre-determined grid-stretch. STAR-CCM+ then inserts a new table point midway between these existing points. If the difference between linear interpolation of the two existing points and the newly inserted point is less than the normalized tolerance, the newly inserted point is discarded. If not, the process is repeated until the maximum number of grid points are reached. The normalized tolerance is the tolerance that you specify, multiplied by the maximum value of the variable in this table, minus the minimum value of the variable in the table. The algorithm is applied to temperature and all of the post-processing species that you specify.

Table entries are generated for the instantaneous chemical states. In reality, turbulence effects are continually at work within the fluid system at timescales smaller than the simulation time-step. To account for the turbulence effects, a probability density function is assumed for the dependent quantities and then sampled when integrating these quantities across the entire simulation time-step. The general form of the integration step is represented as:

$$\widetilde{Q} = \int_{1}^{0} \left(Q\left(Z\right) P\left(Z\right) dZ \right) \tag{3395}$$

When using the Chemical Equilibrium model or the Steady Laminar Flamelet model, the temperature is taken directly from the flamelet table. For the Steady Laminar Flamelet model, temperature is taken from the enthalpy. The mixture properties values (such as dynamic viscosity, thermal conductivity, and diffusivity) for each of the flamelet models can be stored in the flamelet tables.

When the transport data is imported along with the chemical and thermodynamic data for flamelet generation, STAR-CCM+ tabulates the molecular transport properties for Dynamic Viscosity, Molecular Diffusivity, and Thermal Conductivity. Based on the properties of each species, kinetic theory is used to calculate the dynamic viscosity, molecular diffusivity, and thermal conductivity of each species. Mass fraction weighted averaging is then used to calculate the mixture diffusivity, while Mathur-Saxena averaging is used to calculate the mixture viscosity and thermal conductivity. However, if any of the properties that are required to use kinetic theory are missing from the `tran.dat` file, values are taken from the STAR-CCM+ general database for each species instead. These tabulated values are taken from the flamelet table when the respective Material Properties nodes are set to Flamelet Table—necessary for simulations where laminar transport is important (in particular, LES). By default, constant molecular transport properties are used, which are applicable to most simulations in which turbulent transport is much greater than laminar transport (in particular, RANS).

Inert Streams

When an inert stream is present, STAR-CCM+ solves a transport equation for the mean enthalpy, h, and the temperature is calculated by solving:

$$h = \left(1 - Y_I\right)h_f + Y_I h_I \tag{3396}$$

where h_f is the enthalpy of the flamelet species and h_I is the enthalpy of the inert species. In the absence of an inert stream, the temperature is taken directly from the flamelet table—except when using the Steady Laminar Flamelet model, which determines temperature from the enthalpy. As well as storing energy, the flamelet tables store relevant mixture properties.

In the presence of an inert stream, the flamelet table lookup is scaled as:

$$\frac{\overline{Y_f}}{1 - Y_I} \tag{3397}$$

If $Y_I = 0$, all species fractions are given from the flamelet table as they would be without an inert stream. However, if $Y_I \neq 0$, the species fraction in the presence of an inert stream is given by:

$$Y_i = \left(1 - Y_I\right)Y_{i,p} + Y_I Y_{i,I} \tag{3398}$$

where $Y_I + Y_f \leq 1$

If the sum of the fuel and inert mass fractions is less than one for an inlet, the remaining mass fraction for that inlet is attributed to an oxidiser stream.

The Inert Stream model assumes that the inert species and the reacting flamelet species are in thermal equilibrium.

Chemical Equilibrium

In some reacting systems, the reaction occurs almost instantaneously when the necessary species are present in the correct proportions. For these systems, the atomic concentrations and initial temperature at a location determine the instantaneous value of any scalar at that spatial location and time.

The Chemical Equilibrium model assumes that what is mixed has reacted and reached chemical equilibrium. The model assumes that all species diffuse at the same rate, which is reasonable for turbulent flows where turbulent diffusivity is much greater than molecular diffusivity.

In addition to tracking the mixture fraction and its variance, when the Chemical Equilibrium model is active, STAR-CCM+ additionally solves for heat loss ratio.

Heat Loss Ratio

The heat loss ratio γ is used to indicate the amount of heat loss.

For non-adiabatic systems, STAR-CCM+ uses heat loss ratio γ to account for the loss or gain of heat which arises due to situations such as wall heat transfer, radiation, spray evaporation etc. The amount of heat loss or gain represents the departure of the system from its adiabatic state. The flamelet table is additionally parameterized by heat loss ratio—which accounts for the non-adiabatic effects in the CFD domain. Heat loss ratio in the flamelet table represents the normalized enthalpy difference between the cell enthalpy and its adiabatic state which is a function of mixture fraction

$$\gamma = \frac{h_{ad} - h}{h_{sens}} \tag{3399}$$

where h_{ad} is the adiabatic enthalpy, h is the cell enthalpy and h_{sens} is the sensible enthalpy. h_{ad} and h_{sens} are a function of mixture fraction f.

Temperature is stored in the table and then retrieved based on the table dimensions.

Tabulation for Chemical Equilibrium

The independent variables are:

- Mixture Fraction Z
- Mixture Fraction Variance Z_{var}
- Heat Loss Ratio (when using the Non-Adiabatic model) γ

When the boundary condition values are given for Z and the streams are specified for $Z = 1$ (fuel stream) and $Z = 0$ (oxidizer stream), the value of any conserved scalar ϕ (concentration for a specific element) is calculated at any spatial location according to:

$$\phi = Z\phi_f + \left(1 - Z\right)\phi_{ox} \tag{3400}$$

where ϕ_f and ϕ_{ox} represent the values of the conserved scalars in the fuel and oxidizer streams respectively. Therefore, when Z is known, the concentration of any given element is also known at that location. When the elemental concentrations are known, STAR-CCM+ passes them along with the initial temperature to an equilibrium routine which yields:

- the mass fractions of all species
- the density
- the temperature at that point (using a Gibbs free-energy minimization technique)

Integration

The averaged value of any scalar in a turbulent flow field is the instantaneous value that is integrated over the joint PPDF of Z and h:

$$\overline{\phi}\left(\dot{Z}, \overline{h}\right) = \int \phi\left(Z, h\right) P\left(Z, h\right) \mathrm{d}Z\, dh \tag{3401}$$

STAR-CCM+ makes a statistical independence assumption for $P(Z, h)$ so that:

$$P\left(Z, h\right) = P\left(Z\right) P\left(h\right) \tag{3402}$$

and further assumes a Dirac delta function for $P(h)$. In other words, the mean value of enthalpy is used in the calculation of:

$$\overline{\phi}\left(\dot{Z}, \overline{h}\right) = \int \phi\left(Z, \overline{h}\right) P\left(Z\right) \mathrm{d}Z \tag{3403}$$

The integrals in Eqn. (3401) and Eqn. (3403) are pre-computed. When the value of an integral is needed during the calculation, a simple interpolation is all that is needed to provide the correct value.

The non-adiabatic PPDF model can work with Lagrangian evaporation models. In that case, a source term resulting from the evaporation of the Lagrangian phase is added to the right-hand side of the transport equation for mixture fraction Z (Eqn. (3389)).

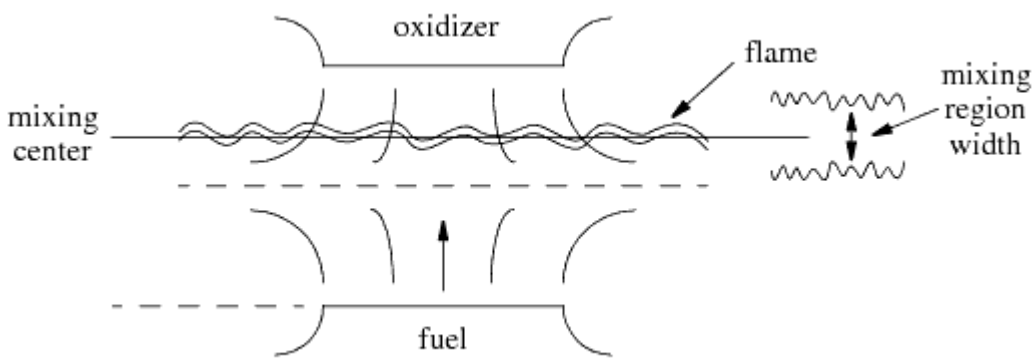
Steady Laminar Flamelet

In combustion theory, a turbulent flame can be considered as a composition of many thin laminar flamelet structures that are locally one-dimensional. In the tabulated approach, STAR-CCM+ solves for reaction products using the one-dimensional model and stores these in the lookup table.

This model assumes that the thickness of the reaction zone is smaller than the smallest turbulence length scale, the Kolmogorov length scale. Therefore, the local instantaneous reaction zone structure is considered to be the same as in a quasi-steady one-dimensional laminar flamelet.

The main effect of the turbulent velocity field on the structure of the laminar flamelet is through the stretch of the reaction zone—described by the scalar dissipation rate (or strain rate). At low values of strain rate, laminar flamelet structures approach the equilibrium state; at high strain rates, flame extinction can occur. Through solving for the quasi-steady, one-dimensional laminar flames under different strain rates and with detailed reaction mechanisms, non- equilibrium and finite-rate chemistry effects can be introduced into the turbulent combustion modeling.

STAR-CCM+ solves for an idealized axisymmetric counterflow diffusion flame, as depicted in the following diagram:



Scalar Dissipation Rate

In the context of the Steady Laminar Flamelet model, the scalar dissipation rate, χ, is the rate at which turbulence-generated fluctuations in mixture fraction are dissipated. In STAR-CCM+, the scalar dissipation rate is defined as:

$$\chi = \frac{D_f Z_{\mathrm{var}}}{\tau_{turb}} \tag{3404}$$

where τ_{turb} is the turbulent timescale and D is the effective diffusivity—which includes laminar and turbulent effects.

Flame temperatures depend strongly on the scalar dissipation rate—the maximum flame temperature decreases with an increasing scalar dissipation rate. Therefore, high values of scalar dissipation rate can cause extinction.

The scalar dissipation is zero at equilibrium. To determine the scalar dissipation rate, STAR-CCM+ starts solving with the initial scalar dissipation rate that is specified and continues increasing this value according

to the scalar dissipation multiplier until the temperature starts to drop—which indicates that the flame has extinguished, or until the maximum scalar dissipation rate is reached.

When simulating a turbulent non-premixed flame, the scalar dissipation rate is highest near the inlet nozzle of a combustor (where the velocity is high). Extinction of the flame near the nozzle can cause lift-off of turbulent flames.

Tabulation for Steady Laminar Flamelet

The independent variables are:

- Mixture Fraction
- Mixture Fraction Variance

The mixture fraction can replace the space coordinate in the one-dimensional system that is shown above, to yield the following equations [663]:

$$\frac{\partial Y_i}{\partial t} = R_{i,kin} + D_i \chi \frac{\partial^2 Y_i}{\partial Z^2} \tag{3405}$$

$$\frac{\partial T}{\partial t} = -\sum_{i=1}^{N} \frac{h_i R_{i,kin}}{C_p} + \alpha\chi\left(\frac{\partial^2 T}{\partial Z^2}\right) \tag{3406}$$

where Y_i is the mass fraction, T is the temperature, α is the thermal diffusivity, C_p the specific heat, and h_i the enthalpy of species i.

The steady laminar flamelet model solves for the balance between diffusion and reaction along the one-dimensional flamelet as a function of the scalar dissipation rate and the mixture fraction. STAR-CCM+ generates the flamelet table by solving Eqn. (3405) and Eqn. (3406) for different values of χ. Results are tabulated as functions of mixture fraction along the line. Therefore, any variable of interest, ϕ, can be expressed as:

$$\phi = \phi(Z, \chi) \tag{3407}$$

Integration

The integration of the Steady Laminar Flamelet model is similar to that of the Chemical Equilibrium model. The difference being that, given Eqn. (3407) for any state-space variable, a joint Probability Density Function of Z and χ are required. That is, we integrate over the χ space as well as the Z space:

$$\phi(Z, \chi) = \iint \phi(Z, \chi)\, P(Z, \chi)\, dZ d\chi \tag{3408}$$

Flamelet Generated Manifold

The Flamelet Generated Manifold model is different from the Chemical Equilibrium (CE) model or Steady Laminar Flamelet (SLF) model since a progress variable which represents the reaction progress is included in the flamelet manifold. This helps to simulate flames in which local ignition or extinction occurs—which is a major distinction from CE or SLF models.

In the current implementation, a constant pressure reactor is used to generate the manifolds. The table has the following dimensions: mixture fraction, heat loss ratio (same definition as in CE), and progress variable before doing the integration.

Unnormalized Progress Variable

The FGM model solves a transport equation for an unnormalized progress variable, y, which is based either on chemical enthalpy or species weights, depending on the option that is chosen when generating the FGM table.

The unnormalized progress variable is defined as:

$$y = \Sigma\left(W_k Y_k\right) \tag{3409}$$

where W_k is the kth species weight and Y_k is the mass fraction of the kth species. When y is based on chemical enthalpy, W_k in Eqn. (3409) is replaced by the enthalpy of the kth species, h_k.

The y transport equation can be written:

$$\frac{\partial \rho y}{\partial t} + \nabla \cdot \left(\rho \mathbf{u} y\right) - \nabla \cdot \left(\Gamma_y \nabla_y\right) = \dot{\omega}_y \tag{3410}$$

where Γ_y is the diffusivity which is computed from the material properties:

$$\Gamma_y = \rho D_{lam} + \frac{\mu_t}{Sc_t} \tag{3411}$$

- D_{lam} is the laminar diffusivity.
- μ_t is the turbulent viscosity.
- Sc_t is the turbulent Schmidt number.

The source term is taken from the precomputed FGM table.

A progress variable is then defined as:

$$c = \frac{y - y_{c=0}}{y_{c=1} - y_{c=0}} \tag{3412}$$

Where:

- y is the unnormalized progress variable of the mixture.
- $y_{c=0}$ is the unnormalized progress variable at the initial state.
- $y_{c=1}$ is the unnormalized progress variable at the equilibrium state.

Progress Variable Variance

There are two options available to compute the Progress Variable Variance:

- Algebraic Relationship

  This option computes the progress variable variance using:

$$y_{var} = c_v * \Delta^2 * \left(\nabla y\right)^2 \tag{3413}$$

where,

c_v= model constant

Δ= mesh size

- Transport Equation

This option computes the progress variable variance using a transport equation:

$$\frac{\partial}{\partial t}\left(\rho\chi y_{var}\right) + \nabla \bullet \left(\rho U y_{var} - \frac{\mu_t}{Sc_g}\nabla y_{var}\right) = \frac{2\mu_t}{Sc_g} tr\left(\nabla y : \nabla y\right) - C_d \rho \frac{\varepsilon}{k} y_{var} \tag{3414}$$

where

$$y_{var} = \overline{\left(y - \overline{y}\right)^2} \tag{3415}$$

Sc_g= turbulent Schmidt number for the variance

C_d = dissipation constant

Progress Variable Source

Different source terms are used to solve the transport equations, depending on the option that is selected for the progress variable source.

- Flamelet Generated Manifold (FGM) kinetic rate

  The source term, $\dot{\omega}_y$, is calculated from the FGM table based on Eqn. (3410). Where the unnormalized progress variable y is solved through the transport equation and the progress variable is computed based on Eqn. (3412).
- Coherent Flame Model (CFM)

  The source term, $\dot{\omega}_{y,\ cfm}$, is calculated based on the CFM flame propagation method Eqn. (3419).
- Turbulent Flame Closure (TFC)

  The source term, $\dot{\omega}_{y,\ tfc}$, is calculated based on the TFC flame propagation method Eqn. (3420).

Tabulation for FGM

The independent variables are:

- Mixture Fraction Z
- Heat Loss Ratio γ
- Progress Variable c

During the tabulation process, the following equations are solved for each Z and γ.

$$\begin{cases} \dfrac{\partial Y_i}{\partial t} = R_{i,kin} \\ \dfrac{\partial T}{\partial t} = -\sum\limits_{i=1}^{N} \dfrac{h_i R_{i,kin}}{C_p} \end{cases} \tag{3416}$$

Each point at physical time instant t is converted to the corresponding progress variable c.

Integration for FGM

After integration, the FGM table has five independent variables:

- Mixture Fraction $\overline{Z}$
- Heat Loss Ratio $\overline{\gamma}$
- Progress Variable $\overline{c}$

- Mixture Fraction Variance $\overline{Z}_{\mathrm{var}}$
- Progress Variable Variance $\overline{c}_{\mathrm{var}}$

For each point, $\overline{Z}$, $\overline{\gamma}$, $\overline{c}$, $\overline{Z}_{\mathrm{var}}$, and $\overline{c}_{\mathrm{var}}$, the averaged quantity $\widetilde{Q}$ is determined by:

$$\begin{aligned}
&\widetilde{Q}\left(\widetilde{Z},\widetilde{c},\widetilde{\gamma},\ \widetilde{Z}_{\mathrm{var}},\ \widetilde{c}_{\mathrm{var}}\right) \\
&= \int_0^1\int_0^1\int_{-1}^1 Q\left(Z,c,\gamma\right)P\left(Z,c,\gamma\right)dZdcd\gamma \\
&= \int_0^1\int_0^1 Q\left(Z,c,\widetilde{\gamma}\right)P\left(Z,c\right)dZdc \\
&= \int_0^1\int_0^1 Q\left(Z,c,\widetilde{\gamma}\right)P_c\left(c\,\middle|\,Z\right)P_Z\left(Z\right)dZdc
\end{aligned} \tag{3417}$$

here, $P_Z(Z)$ is a beta pdf that is parameterized as Z and Z_{var}. $P_c(c\,|\,Z)$ is a beta pdf that is parameterized as c and c_{var}.

Flame Propagation

There are two flame propagation models in STAR-CCM+ that are used to calculate the flame movement in space, for premixed and partially-premixed systems, through calculation of the turbulent flame speed.

- Coherent Flame Model (CFM)

  The CFM model assumes that combustion occurs in the flamelet region. The mean turbulent reaction rate can be expressed as the product of the flame surface density by the consumption rate of fuel per unit of flame area.
- Turbulent Flame-speed Closure (TFC)

  The TFC combustion model assumes that in a premixed combustion system, the reaction takes place in a thin layer which separates the whole domain to unburnt and fully burnt mixtures. The mean reaction rate is closed using the turbulent flame speed.

Both of these flame position models solve for the flame position by transporting a reaction progress variable to obtain the reaction progress:

$$\frac{\partial \rho y}{\partial t} + \nabla\cdot\left(\rho u y\right) - \nabla\cdot\left(\Gamma_y \nabla y\right) = \dot{\omega}_{y,\ tfc}\left(\mathrm{or}\quad \dot{\omega}_{y,\ cfm}\right) \tag{3418}$$

When the TFC/CFM model is used alongside the Chemical Equilibrium (CE) model or Steady Laminar Flamelet (SLF) model, y is taken as the fuel mass fraction. However, when the TFC/CFM model is selected with the Flamelet Generated Manifold (FGM) model, y is the unnormalized progress variable which is defined by the FGM model in Eqn. (3409).

The source terms are calculated as:

CFM

$$\dot{\omega}_{y,\ cfm} = -\left(\rho_u S_u \Sigma\right) \tag{3419}$$

where:

- ρ_u is the unburnt density
- S_u is the laminar flame speed
- Σ is the flame area density—which is defined as the flame area per unit volume

There are two options for the laminar flame speed:

- Meghalchi
- Gulder

TFC

$$\dot{\omega}_{y,\ tfc} = A\,\rho_u S_t |\nabla y| \tag{3420}$$

where:

- A is the TFC rate coefficient
- ρ_u is the unburnt density
- S_t is the turbulent flame speed

There are three options for the turbulent flame speed:

- Zimont
- Peters
- User-defined (a scalar profile)

In a partially premixed flame, the averaged value for any quantity Q at location $\underline{x}$ and time t, $\widetilde{Q}(\underline{x},t)$, can be calculated with the following form:

$$\widetilde{Q}(\underline{x},t) = \int_0^1 \int_0^1 Q(Z,c) P(Z,c;\underline{x},t)\ \mathrm{d}\,Z\ \mathrm{d}\,c \tag{3421}$$

Here $P(Z,c;\underline{x},t)$ is the joint pdf of the mixture fraction Z and progress variable c.

The joint pdf can be decomposed to a conditional pdf $P_z(Z\,|\,c;\underline{x},t)$ and a marginal pdf $P_c(c;\underline{x},t)$:

$$P(Z,c;\underline{x},t) = P_z(Z\,|\,c;\underline{x},t) P_c(c;\underline{x},t) \tag{3422}$$

Using the classical flamelet approximation of a bimodal form:

$$P_c(c;\underline{x},t) = \tilde{c}\delta(1-c) + (1-\tilde{c})\delta(c) \tag{3423}$$

where δ is a δ function, we can get:

$$\widetilde{Q}(\underline{x},t) = (1-\tilde{c})\widetilde{Q}^u(\underline{x},t) + \tilde{c}\widetilde{Q}^b(\underline{x},t) \tag{3424}$$

where $\widetilde{Q}^u(\underline{x},t)$ is the unburnt state and $\widetilde{Q}^b(\underline{x},t)$ is the burned state. In this model, the burned state is from the Chemical Equilibrium or Steady Laminar Flamelet chemistry model.

Flame Area Density

The equation for σ is:

$$\frac{\partial(\rho\sigma)}{\partial t} + \nabla \cdot \left(\rho u \sigma\right) - \nabla \cdot \left(\rho \Gamma_\sigma \nabla \sigma\right) = S_\Sigma + \nabla \cdot \left(\Gamma_\sigma \sigma \nabla \rho\right) \tag{3425}$$

The source term S_Σ includes the flame area production by stretch and destruction by fuel consumption and is calculated using the following equation:

$$S_\Sigma = \alpha K_t \Sigma - \beta \frac{\rho_u Y_{ft} S_u \left(1 + a\sqrt{k}/S_u\right)}{\rho Y_f} \Sigma^2 \tag{3426}$$

α, β and a are model parameters. Σ is the flame area density, which is defined as the flame area per unit volume. σ is the flame area per unit mass.

$$\Sigma = \rho\sigma \tag{3427}$$

where ρ is the gas density.

The method for K_t calculation is described in the Net Flame Stretch K_t Calculation section.

Net Flame Stretch K_t Calculation

The methods that are developed in [650] are used. K_t is tabulated with two parameters:

$$\Gamma_K = \frac{K_t}{\varepsilon/k} = f\left(\frac{u'}{S_u}, \frac{I_l}{\delta_l}\right) \tag{3428}$$

u' is the turbulence intensity that is obtained from the following equation:

$$u' = \sqrt{2/3k} \tag{3429}$$

I_l is the integral length scale that is defined as:

$$I_l = \frac{C_\mu^{3/4}}{\kappa} \cdot \frac{k^{3/2}}{\varepsilon} \tag{3430}$$

where C_μ and κ are the turbulent viscosity coefficient and von Karman constant of the K-Epsilon model, respectively.

δ_l is the thermal laminar flame thickness that is calculated as follows:

$$\delta_l = 2\frac{\mu_b}{Pr \rho_u S_u} \tag{3431}$$

In this expression, Pr is the laminar Prandtl number of the burnt gas and assumed to be 0.9. μ_b is the molecular viscosity of the burnt gas that is calculated from Sutherland's law:

$$\mu_b = \frac{\mu_1 T_b^{1.5}}{\mu_2 + T_b} \tag{3432}$$

Here, μ_1 and μ_2 are constants with the values of 1.457e-6 and 110, respectively. T_b is the burnt gas temperature. Γ_k has the following form:

$$\Gamma_k = \Gamma_p - B \cdot \Gamma_q \tag{3433}$$

Γ_p and Γ_q are the flame production and quench due to the stretch; they follow the empirical correlation in [650]. B is a model parameter.

Contents:

Flame Speed Calculations

Flame Speed Calculations

Laminar Flame Speed

STAR-CCM+ incorporates two correlations for calculating the unstrained laminar flame speed. As the correlation by Metghalchi and Keck is only valid for equivalence ratios ϕ that are close to 1, the correlation by Gülder is recommended for most general applications.

Metghalchi

The correlations that is proposed by Metghalchi and Keck [664] is calculated as follows:

$$S_u = S_{u0}\left(\frac{T_u}{T_0}\right)^{\alpha}\left(\frac{p}{p_0}\right)^{\beta} \qquad \mathrm{m/s} \tag{3434}$$

where p is the pressure, T is the temperature, the subscripts 0 and u denote reference and unburnt gas properties, respectively, and S is the laminar flame speed. The reference laminar flame speed S_{u0} and the exponents α and β depend on the equivalence ratio ϕ of the fuel. If $0.4 \leq p \leq 50$ atm, $300 \leq T \leq 700$ K, and $0.8 \leq \phi \leq 1.5$, the authors claim that the laminar flame speed is within 10% of the measured data. $\phi \geq 0.65$ is recommended because:

- None of the combustion models can accurately predict burning in lean mixtures.
- The laminar flame speed correlation function progressively diverges from the experimentally observed data in lean mixtures.

Gülder

The second laminar flame speed correlation, which Gülder proposed [665], is calculated as follows:

$$S_u = ZW\phi^{\eta}\exp\left[-\xi(\phi - 1.075)^2\right]\left(\frac{T_u}{T_0}\right)^{\alpha}\left(\frac{P}{P_0}\right)^{\beta} \tag{3435}$$

where Z, W, η, ξ, α and β are fuel-dependent constants that are defined in the table below:

Fuel	Z	W	η	ξ	α	β	
						$\phi < 1$	$\phi > 1$
Methane	1	0.422	0.15	5.18	2.00	–0.5	–0.5
Propane	1	0.446	0.12	4.95	1.77	–0.2	–0.2
Methanol	1	0.492	0.25	5.11	1.75	$-0.2/\sqrt{\phi}$	-0.2ϕ

Fuel	Z	W	η	ξ	α	β	
Ethanol	1	0.465	0.25	6.34	1.75	$-0.17/\sqrt{\phi}$	-0.17ϕ
Iso-octane	1	0.4658	–0.326	4.48	1.56	–0.22	–0.22

Turbulent Flame Speed

Zimont

Zimont used the following correlation for the turbulent flame speed [689]:

$$S_T = 0.5\, G\left(\mathrm{u}'\right)^{3/4} S_L^{1/2} \alpha_u^{-1/4} I_l^{1/4} \tag{3436}$$

Here, u' is the turbulent velocity, S_L is the laminar flame speed (see Laminar Flame Thickness), α_u is the unburnt thermal diffusivity of the unburnt mixture, and I_l is the integral turbulent length scale.

The stretch factor G takes the stretch effect into account by representing the probability of unquenched flamelets which is obtained by integrating the log-normal distribution of the turbulent dissipation rate:

$$G = \frac{1}{2} erfc\left[-\frac{1}{\sqrt{2\sigma}}\left(\ln\frac{\varepsilon_{cr}}{\varepsilon} + \frac{\sigma}{2}\right)\right] \tag{3437}$$

where $erfc$ is a complementary error function and σ is the standard deviation of the distribution of ε computed with the following equation:

$$\sigma = \mu_{str} \ln\left(I_l/\eta\right) \tag{3438}$$

where I_l is the integral turbulent length scale, η is the Kolmogorov micro-scale, and μ_{str} is an empirical model coefficient with a default value of 0.28.

ε_{cr} is the turbulent dissipation rate at the critical strain rate g_{cr}:

$$\varepsilon_{cr} = 15 \nu g_{cr}^2 \tag{3439}$$

where ν is the kinematic viscosity of the fluid.

A high value for g_{cr} suggests no occurrence of the flame stretch. One method to compute g_{cr} is to assume that it is proportional to the chemical time scale:

$$g_{cr} = \frac{B S_L^2}{\alpha_u} \tag{3440}$$

Peters

The Peters correlation [690] for turbulent flame speed has the following form:

$$s_T = s_L\left(1 + \sigma_t\right) \tag{3441}$$

where:

$$\sigma_t = -AB + \sqrt{(AB)^2 + C\frac{\mathrm{u}' I_l}{s_L \delta_L^0}}$$
$$A = \frac{A_4 B_3^2}{2B_1}$$
$$B = \frac{I_l}{\delta_L^0} c_{ew}$$
$$C = A_4 B_3^2 \tag{3442}$$

where δ_L^0 is the laminar flame thickness, and c_{ew} is Ewald's corrector which has a default value of 1.0.

A_1, A_4, B_1, and B_3 are model constants with default settings of 0.37, 0.78, 2.0, and 1.0, respectively.

Note: A wall effect constant is used to model the quenching of the flame at walls by multiplying the constant when computing the turbulent flame speed. By modifying the constant from 0 to 1, the wall quenching effect on the flame in the vicinity of wall boundaries can be adjusted from fully extinguished to no effect.

Fluid-Fluid Interphase Reactions

For Eulerian Multiphase (EMP) and Volume of Fluid Method (VOF) reacting flows, the user-defined reaction rate is calculated in the form:

$$\omega_i = k_i \Pi \left[X_j\right]^{r_j} a_{cd} \tag{3443}$$

based on the following user inputs:

- k_i, the user reaction coefficient for reaction i.
- The units of k_i. The unit of $k_i \Pi \left[X_j\right]^{r_j}$ must take the form [Quantity/Length²/s]—as an option in the GUI.
- r_j, the rate exponent, which is specified for each reactant and product.

X_j is the species concentration and a_{cd} is the interaction area density, see Interaction Area Density. Since a_{cd} is applied internally, if for some reason the user-defined reaction rate does not include a_{cd}, then k_i should be divided by `$InteractionAreaDensityPhaseInteraction1` or similar.

Surface Chemistry

In this generalized surface chemistry implementation, gas species mass fractions at the reacting surfaces are assumed to be the same as the gas species mass fractions at the center of the cell volume.

As the goal of the surface reaction is to compute concentrations of gas and site species at the wall, there is the following system of equations to solve:

$$\frac{\partial \rho Y_{i,\mathrm{wall}}}{\partial t} = \gamma_w \widehat{R}_{i,\mathrm{gas}}, i = 1, 2, 3, ..., N_g \tag{3444}$$

$$\frac{\partial \rho S_{i,\mathrm{wall}}}{\partial t} = \gamma_w \widehat{R}_{i,\mathrm{site}}, i = 1, 2, 3, ..., N_s \tag{3445}$$

where γ_w is the surface washcoat factor, $Y_{i,\mathrm{wall}}$ is the mass fraction of the gas species i at the wall, and $S_{i,\mathrm{wall}}$ is the site species concentration, which is given as:

$$S_{i,\mathrm{wall}} = \rho_{\mathrm{site}} z_i \tag{3446}$$

where ρ_{site} is the site density which specifies the number of active sites that exist on the surface. z_i is the surface site fraction which specifies the fraction of the total number of sites that species, i, covers. You can set the Surface Initial Site Fraction Profiles at boundaries and plot the surface site fractions of each species to monitor these as the reaction proceeds. The sum of all z_i values for all species is equal to one.

Together, Eqn. (3444) and Eqn. (3445) form a system of ODEs that are solved with the CVODE package.

Integrating the Gas Species Solution Back to STAR-CCM+

The solution of this surface reaction model is the composition of the gas and surface species on the wall, $Y^*_{i,\mathrm{wall}}$, and $S^*_{i,\mathrm{wall}}$. The surface site species fraction is updated directly. For the gas species, the source term of the species transport equation is updated with the addition of averaged source term, Ω:

$$\Omega_i = \frac{\left(Y^*_{i,\mathrm{wall}} - Y_{i,\mathrm{wall}}\right)}{dt} i = 1, 2, 3, ..., N_g \tag{3447}$$

This term contributes to the region source term S_{Y_i} from Eqn. (2202).

$Y^*_{i,\mathrm{wall}}$ is the gas-phase mass fraction at the wall, at the end of the surface chemistry integration, for time-step dt.

Surface Chemistry with Complex Chemistry

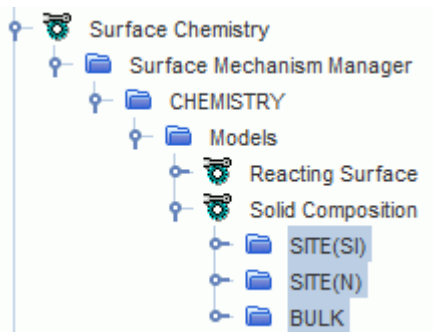
With the surface chemistry model, three types of species are defined.

- Species in the gas phase with mass fractions denoted with $Y_i^{(g)}$
- Species on the surface at the gas-surface interface with site fractions denoted as $Z_i^{(s)}$. Some sites are "open" (denoted as a generic species $OS^{(s)}$) when they do not contain any species, and they can be used to model absorption from the gas phase.

- Species in the solid layer—bulk species with their mole fractions denoted as $W_i^{(b)}$

Limitations to surface chemistry modeling:

- You can only apply surface chemistry to wall and baffle boundary conditions or to porous regions.
- By default, mesh movement is not implemented to account for surface growth or shrinking. However, it is possible to use morphing motion and specify surface site and bulk deposition rates as input using the Surface Chemistry physics model.



It is assumed that the gas species mass fractions at the reacting surfaces are the same as the gas species mass fractions at the center of the cell volume. Then species and energy transport equations for the gas phase have one more source term ($\omega_k^{(s)}$ in the species transport equation and $src_h(s)$ in the enthalpy equation) corresponding to surface reactions:

$$\frac{\partial}{\partial t}\rho\chi Y_k^{(g)} + \frac{\partial}{\partial x_j}\left(\rho u_j Y_k^{(g)} + D_{k,j}\right) = \omega_k^{(g)} + \omega_k^{(s)} \tag{3448}$$

and

$$\frac{\partial}{\partial t}\left(\rho\chi E\right) + \frac{\partial}{\partial x_j}\left(\rho u_j H + Dh_j\right) = s + src_h(s) \tag{3449}$$

It is assumed that no convection or diffusion of surface or bulk species is present. Therefore, the governing transport equations for surface and bulk species are the following:

$$\frac{\partial}{\partial t}Z_k^{(s)} = S_{Z_k^{(s)}} \tag{3450}$$

and

$$\frac{\partial}{\partial t}W_k^{(b)} = S_{W_k^{(b)}} \tag{3451}$$

These equations are solved simultaneously.

Output for the gas phase uses the operator splitting algorithm to transform the gas species transport equation from the PDE to ODE form. Output for the surface phase consists of the new molar fraction of bulk species and site fraction of surface site species. However, there are two more source terms that are related to the surface/gas interaction—$\omega_k^{(s)}$ and $src_h(s)$ in the equations above. Unlike the source terms from the gas phase, these source terms are not linearized.

Operator Splitting Algorithm

This algorithm requires mean values for the source terms over the time-interval dt.

$$\omega_k^{(s)} = \rho \frac{{Y^*}_k^{(g)} - Y_k^{(g)}}{dt} \tag{3452}$$

$$src_h(s) = \rho \frac{\sum_{k=1,g} {Y^*}_k^{(g)} h_k^* - Y_k^{(g)} h_k}{dt} \tag{3453}$$

Index g denotes that summation is performed over the gas species. The h_k is the partial enthalpy of the gas species k. Unlike in the case of gas phase only, the source terms of species do not sum to 0 so this summation term has to be added to the mass conservation equation.

Ignitors

When an ignitor is active, mesh cells which are contained within the ignitor volume shapes have their properties changed as specified by the ignitor.

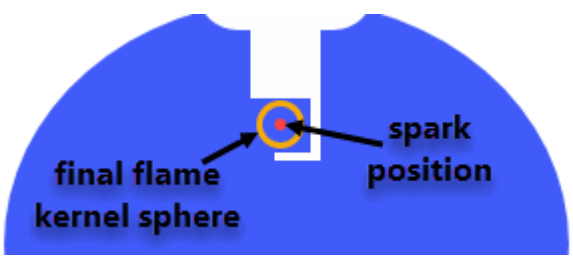
Subgrid Spark Ignition Model

During spark ignition, a high-voltage electrical current creates a plasma arc between electrodes which provides enough energy to ignite a fuel/air mixture and creates a hot gas-ball (flame kernel). The flame propagates outwards from the site of ignition.

There are two main difficulties with spark modeling. The first is the complex physics of the initial arc creation, laminar flame propagation, and the transition to turbulence. The second issue is that the initial spark kernel is very small, and commonly not well resolved on the CFD grid. The Subgrid Spark Ignition model attempts to mitigate these difficulties by solving a 0D equation for the spark radius until the spark is resolved on the CFD mesh.

In STAR-CCM+, a modeled ODE equation for the burnt gas mass radius r_b is solved. The burnt volume from the 0D solution is then averaged onto a fixed volume sphere (final flame kernel sphere) within the CFD mesh with a radius of about five grid cells.

The reaction progress within the flame kernel sphere increases uniformly from unburnt to burnt as the 0D spark radius increases to the final flame kernel sphere radius. After which, the flame is resolved on the CFD grid.



Arc Breakdown and Initial Flame Kernel

The arc breakdown creates a small kernel of initial temperature T_i and spark radius $r_b(0)$—which is logged in the *Output* window in metres, for example: `Kernel Radius for spark #1 = 0.00058087`

The initial spark radius (burnt gas mass radius) $r_b(0)$ is given by [687]:

$$r_b(0) = \sqrt{\frac{\gamma - 1}{\gamma} \frac{E_{bd}}{p d_{gap} \pi \left(\frac{1 - T_u}{T_b(0)} \right)}} \tag{3454}$$

γ is the ratio of specific heats, p is the gas pressure in bar, d_{gap} is the inter-electrodes distance (1mm), T_u is the unburnt gas temperature, $T_b(0)$ is the initial temperature, and E_{bd} is the breakdown energy, given by [685], [686]:

$$E_{bd} = \frac{\left(4.3 + \frac{136p}{T_u} + \frac{324p}{T_u d_{gap}}\right)^2}{C^2 d_{gap}} \tag{3455}$$

in which $C = 245\left[kV/(Jmm)^{1/2}\right]$.

The initial temperature $T_b(0)$ is given by [687]:

$$T_b(0) = T_u\left[\frac{1}{\gamma}\left(\frac{T_{bd}}{T_u} - 1\right) + 1\right] \tag{3456}$$

T_{bd} is the breakdown temperature—which is set at 60,000K.

Ignition Flame Kernel Expansion

As the flame propagates outwards from the flame kernel, the burnt gas mass m_b evolves according to:

$$\frac{dm_b}{dt} = \rho_u S_l A_k \Xi \tag{3457}$$

Where ρ_u is the unburnt gas density, S_l is the laminar flame speed, A_k is the flame kernel surface area, and Ξ is the turbulence wrinkling factor [688]:

$$\Xi \coloneqq \frac{S_t}{S_l} = 1 + A\left(\frac{u'}{S_l}\right)^{5/6} \tag{3458}$$

Where S_t is the turbulent flame speed, u' is the velocity fluctuation, and A is the turbulent flame speed coefficient.

Therefore, the ODE equation for the burnt radius is:

$$\frac{dr_b}{dt} = \frac{\rho_u}{\rho_b} S_l \Xi = \frac{T_b}{T_u} S_l \Xi = (\tau + 1) S_t \tag{3459}$$

Where T_b is the temperature in the ignition kernel—which is assumed to be equal to the equilibrium temperature. τ is the expansion ratio:

$$\tau \coloneqq \frac{T_b}{T_u} - 1 \tag{3460}$$

Progress Variable Distribution

The fixed/final radius of the flame kernel sphere $r_b(final)$ is defined as:

$$r_b(final) = \max\left(5r_b(0), 5\Delta x, 0.1L_t\right) \tag{3461}$$

Where $r_b(0)$ is the initial spark radius in metres, Δx is the average linear cell size, and L_t is defined as:

$$L_t = \frac{k^{3/2}}{\varepsilon} \tag{3462}$$

The progress variable for ignition is:

$$c(r_b) = \left(\frac{r_b}{r_b(final)}\right)^3 \tag{3463}$$

NOx

There are three sources of NOx emissions; thermal, prompt, and fuel. The transport equation that is used to solve for NOx is the same for all NOx models, with the addition of reacting source terms to account for each NOx source.

NOx concentration is usually low compared to other species in combustion systems. As a result, it is generally agreed that NOx chemistry has negligible influence and can be decoupled from the main combustion and flow field calculations.

$$\rho\frac{\partial Y_{NO_X}}{\partial t} + \nabla\cdot\left[\rho\ \mathbf{v}\,Y_{NO_X} - \left[\rho D_f + \frac{\mu_t}{\sigma_t}\right]\nabla Y_{NO_X}\right] = \dot{\omega}_{NO_{X,\ therm}} + \dot{\omega}_{NO_{X,\ prompt}} + \dot{\omega}_{NO_{X,\ fuel}} \tag{3464}$$

NOx Thermal

The NOx Thermal model solves the transport equation for NO with the thermal source term $\dot{\omega}_{NO_{X,\ therm}}$

Formation of nitric oxide during combustion of hydrocarbons is temperature-dependent for the NOx Thermal model. It is produced by the reaction of atmospheric nitrogen with oxygen at elevated temperatures.

Thermal NOx is formed by high temperature oxidation of atmospheric nitrogen. For thermal nitric oxide, the principal reactions are generally recognized to be those proposed by the following three extended Zeldovich mechanisms:

$$N_2 + O \underset{k_{-1}}{\overset{k_1}{\Leftrightarrow}} NO + N \tag{3465}$$

$$N + O_2 \underset{k_{-2}}{\overset{k_2}{\Leftrightarrow}} NO + O \tag{3466}$$

$$N + OH \underset{k_{-3}}{\overset{k_3}{\Leftrightarrow}} NO + H \tag{3467}$$

The rate constants for these reactions have been measured in numerous experimental studies, and the data obtained from these studies has been critically evaluated by Baulch et al. [667]. The expressions for rate coefficients of reactions are given below (in units of $\mathrm{m}^3(\mathrm{kmol})^{-1}s^{-1}$):

$$k_1 = 1.8 \times 10^{11}\mathrm{e}^{-\left(\frac{38370}{T}\right)} \tag{3468}$$

$$k_{-1} = 3.8 \times 10^{10}\mathrm{e}^{-\left(\frac{425}{T}\right)} \tag{3469}$$

$$k_2 = 1.8 \times 10^7 T \mathrm{e}^{-\left(\frac{4680}{T}\right)} \tag{3470}$$

$$k_{-2} = 3.8 \times 10^6 T \mathrm{e}^{-\left(\frac{20820}{T}\right)} \tag{3471}$$

$$k_3 = 7.1 \times 10^{10} \mathrm{e}^{-\left(\frac{450}{T}\right)} \tag{3472}$$

$$k_{-3} = 1.7 \times 10^{11} \mathrm{e}^{-\left(\frac{24560}{T}\right)} \tag{3473}$$

where k_1, k_2 and k_3 are the forward rate constants and k_{-1}, k_{-2} and k_{-3} are the backward rate constants for NOx reactions, respectively. The rate of formation of NOx is significant only at high temperatures since the thermal fixation of nitrogen requires the breaking of a strong N2 bond. This effect is represented by the high activation energy of the first reaction of the mechanism, which makes this reaction the rate-limiting step of the Zeldovich mechanism.

For the Complex Chemistry or Eddy Break-Up combustion models, if the concentrations of radicals O, OH and H are calculated by the combustion model, they are used in the NOx model. Otherwise, certain assumptions have to be made to obtain their values. The concentration of radical H is assumed to be zero. Two approaches are being used to solve the O-concentration:

$$[O] = k_p [O_2]^{1/2} \tag{3474}$$

- **Equilibrium Approach**

  where:

$$k_p = 1.255 \times 10^4 T^{-1/2} \mathrm{e}^{-\left(\frac{31090}{T}\right)} \tag{3475}$$

- **Partial Equilibrium Approach**

  This approach considers a third-body reaction, which leads to a different expression for k_p:

$$k_p = 1.16 \times T^{1/2} \mathrm{e}^{-\left(\frac{27123}{T}\right)} \tag{3476}$$

  The partial equilibrium approach generally predicts higher O-concentrations than the equilibrium approach.

There are also two approaches being used to solve the OH-concentration:

- **Exclusion Approach**

  In this approach, the third reaction in the extended Zeldovich mechanism is assumed to be negligible through the following observation:

$$k_2 [O_2]_{\mathrm{eq}} >> k_3 [OH]_{\mathrm{eq}} \tag{3477}$$

- **Partial Equilibrium Approach**

  With this approach, the OH-concentration is given as:

$$[OH] = 6.7331 * T^{-0.57} e^{-\left(\frac{4595}{T}\right)} [O]^{1/2} [H_2O]^{1/2} \tag{3478}$$

The net rate of formation of NO is given by:

$$\frac{d[NO]}{dt} = k_1[O][N2] + k_2[N][O2] + k_3[N][OH] - k_{-1}[NO][N] - k_{-2}[NO][O] - k_{-3}[NO][H] \tag{3479}$$

where $[X]$ is molar concentration of X (in units of $\mathrm{kmol\,m^{-3}s^{-1}}$). The activation energy for oxidation of N is small, hence a quasi-steady state can be established:

$$\begin{aligned} \frac{d[N]}{dt} &= 0 = k_1[O][N2] - k_2[N][O2] - k_3[N][OH] \\ &- k_{-1}[NO][N] + k_{-2}[NO][O] + k_{-3}[NO][H] \\ &\Rightarrow [N] = \frac{k_1[O][N2] + k_{-2}[NO][O] + k_{-3}[NO][H]}{k_{-1}[NO] + k_2[O2] + k_3[OH]} \end{aligned} \tag{3480}$$

Based on this assumption, the instantaneous rate of formation of NOx is (in units of $\mathrm{kmol\,m^{-3}s^{-1}}$):

$$\begin{aligned} \frac{d[NO]}{dt} &= k_1[O][N2] - k_{-2}[NO][O] - k_{-3}[NO][H] \\ &+ \frac{k_1[O][N2] + k_{-2}[NO][O] + k_{-3}[NO][H]}{k_{-1}[NO] + k_2[O2] + k_3[OH]} \left(k_2[O2] + k_3[OH] - k_{-1}[NO]\right) \end{aligned} \tag{3481}$$

which can be simplified further to the expression:

$$\begin{aligned} \frac{d[NO]}{dt} &= k_1[O][N2] - k_{-2}[NO][O] - k_{-3}[NO][H] + \frac{k_1[O][N2]\left(k_2[O2] + k_3[OH]\right)}{k_{-1}[NO] + k_2[O2] + k_3[OH]} \\ &+ \frac{k_{-1}[NO]^2\left(k_{-2}[O] + k_{-3}[H]\right)}{k_{-1}[NO] + k_2[O2] + k_3[OH]} \end{aligned} \tag{3482}$$

Therefore, the instantaneous rate of thermal NOx production can be expressed as:

$$\dot{\omega}_{NO_{X,\,therm}} = A - B[NO] + \frac{A\frac{C}{D}}{[NO] + \frac{C}{D}} - [NO]^2 \frac{B}{[NO] + \frac{C}{D}} \tag{3483}$$

where:

$$\begin{aligned} A &= k_1[O][N2] \\ B &= k_{-2}[O] - k_{-3}[H] \\ C &= k_2[O2] + k_3[OH] \\ D &= k_{-1} \end{aligned} \tag{3484}$$

When modeling NOx with flamelet models, the values for the coefficients A, B, C, and D are stored in the flamelet table. The flamelet model can then calculate the flamelets source term without calculating for each species.

If both the NOx Thermal and NOx Prompt models are activated, then the net production rate of NOx is the sum of the thermal and prompt NOx.

NOx Prompt

The formation of prompt NOx involves complex reactions with many intermediate species such as CH, CH2, HCN, CN, and others. These complex mechanisms are simplified to a global rate.

For most hydrocarbon fuels, the production rate of prompt NOx is taken from the global kinetics parameters derived by De Soete [668] and represented by the following equations:

$$\dot{\omega}_{NO_X,\, prompt} = k_r Mw_{NO_X} \tag{3485}$$

Where Mw_{NO_X} is the molecular weight of NOx and k_r is the rate constant given by:

$$k_r = fA\left[\mathbf{O}_2\right]^a\left[\mathbf{N}_2\right]\left[\mathbf{Fuel}\right]\mathrm{e}^{-\left(\frac{E_A}{R_u T}\right)} \tag{3486}$$

E_A is the activation energy. A is the pre-exponential factor that is given by:

$$A = 6.4\ e^{6\left(R_u T/p\right)^{a+1}} \tag{3487}$$

f is the correction factor that is used to change the value of the pre-exponential factor—for a particular hydrocarbon whose kinetics data is known to you. The value of a varies with the oxygen mole fraction X_{O2} as follows:

$X_{O2} \leq 4.1 \times 10^{-3}$	$a = 1$
$4.1 \times 10^{-3} < X_{O2} \leq 1.11 \times 10^{-2}$	$a = -3.95 - 0.9 \ln X_{O2}$
$1.11 \times 10^{-2} < X_{O2} < 0.03$	$a = 0.35 - 0.1 \ln X_{O2}$
$X_{O2} \geq 0.03$	$a = 0$

If both the NOx Thermal and NOx Prompt models are activated, then the net production rate of NOx is the sum of the thermal and prompt NOx.

Fuel NOx

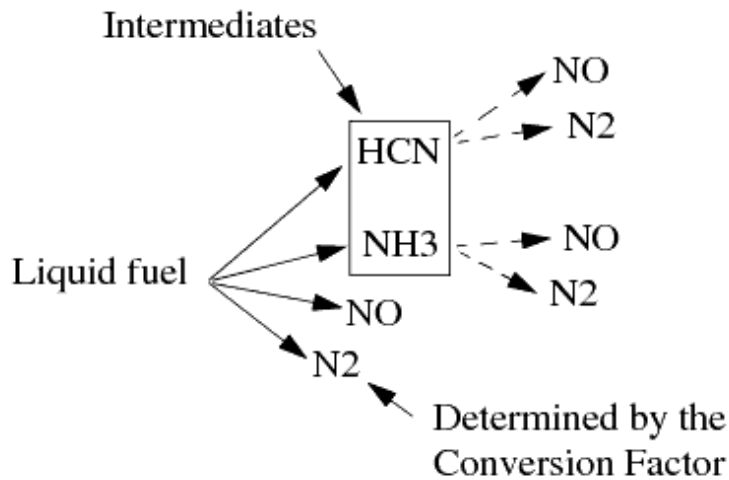
The Fuel NOx model solves three transport equations for NO, HCN, and NH3. Each of these transport equation includes the source terms $\dot{\omega}_{NO}$, $\dot{\omega}_{HCN}$, and $\dot{\omega}_{NH3}$. These source terms are calculated as:

$$\begin{aligned} \dot{\omega}_{HCN} &= \dot{\omega}_{HCN,\ \text{produced}} + \dot{\omega}_{HCN,\ \text{consumed}} \\ \dot{\omega}_{NH3} &= \dot{\omega}_{NH3,\ \text{produced}} + \dot{\omega}_{NH3,\ \text{consumed}} \\ \dot{\omega}_{NO} &= \dot{\omega}_{NO,\ \text{produced}} + \dot{\omega}_{NO,\ \text{consumed}} \end{aligned} \tag{3488}$$

When using the Lagrangian phase models—Liquid Fuel NOx or Coal Fuel NOx—these source terms are derived as shown below.

Liquid Fuel NOx

For liquid fuel, the only source of nitrogen is the evaporating fuel. The intermediates formed can be HCN or NH3. The fuel can also form NO through other pathways. There is also a possibility that N2 is formed as one of the intermediate species. The following figure shows the possible routes.



For multi-component droplets such as $C_xH_yN_z + H_2O$, you identify the fuel species. Only the fuel $C_xH_yN_z$ source is needed.

$$\dot{\omega}_{HCN,\,\text{produced}} = \frac{(\dot{\omega}_{\text{fuel}})(Y_N)(CF)(Z_{HCN})(Mw_{HCN})}{(Mw_N)(V)}$$
$$\dot{\omega}_{NH3,\,\text{produced}} = \frac{(\dot{\omega}_{\text{fuel}})(Y_N)(CF)(Z_{NH3})(Mw_{NH3})}{(Mw_N)(V)} \tag{3489}$$

where $\dot{\omega}_{\text{fuel}}$ is the rate of evaporation of liquid fuel (kg/s), V is the cell volume, and CF is a correction factor for fuel NOx.

$\dot{\omega}_{NO,\,\text{produced}}$ consists of NO produced from the fuel directly and/or NO produced by the oxidation of HCN/NH3:

$$\dot{\omega}_{NO,\,\text{produced}} = \dot{\omega}_{NO,\,\text{direct}} + \dot{\omega}_{NO,\,\text{oxidation}} \tag{3490}$$

$$\dot{\omega}_{NO,\,\text{direct}} = \frac{(\dot{\omega}_{\text{fuel}})(Y_N)(CF)(Z_{NO})(Mw_{NO})}{(Mw_N)(V)} \tag{3491}$$

There are four types of reactions:

$$\begin{aligned} HCN + O2\ (R_{1-HCN})\ \ leads\ \ to\ \ NO_{production} \\ HCN\ +\ NO\ (R_{2-HCN})\ \ leads\ \ to\ \ NO_{consumption} \\ NH3\ +\ O2\ (R_{1-NH3})\ \ leads\ \ to\ \ NO_{production} \\ NH3\ +\ NO\ (R_{2-NH3})\ \ leads\ \ to\ \ NO_{consumption} \end{aligned} \tag{3492}$$

The kinetics rates are obtained from De Soete [668].

For intermediate species HCN:

$$R_{1-HCN} = A_{1-HCN} X_{HCN} X_{O2}^{a} e^{-E_{1,HCN/(RT)}}$$
$$R_{2-HCN} = A_{2-HCN} X_{HCN} X_{NO}^{a} e^{-E_{2,HCN/(RT)}} \quad (3493)$$

where:

$R_1, R_2 =$ conversion rates of $HCN\left(S^{-1}\right)$

$X =$ mole fractions

$A_{1,\ HCN} = 1.0 \times 10^{10}\left(S^{-1}\right)$

$A_{2,\ HCN} = 3.0 \times 10^{12}\left(S^{-1}\right)$

$E_{1,\ HCN} = 280451.95\ \mathrm{J/gmol}$

$E_{2,\ HCN} = 251151\ \mathrm{J/gmol}$

$X_{O2} \leq 4.1 \times 10^{-3}$	$a = 1$
$4.1 \times 10^{-3} < X_{O2} \leq 1.11 \times 10^{-2}$	$a = -3.95 - 0.9 \ln X_{O2}$
$1.11 \times 10^{-2} < X_{O2} < 0.03$	$a = 0.35 - 0.1 \ln X_{O2}$
$X_{O2} \geq 0.03$	$a = 0$

For intermediate species NH3:

$$R_{1-NH3} = A_{1,NH3} X_{NH3} X_{O2}^{a} e^{-E_{1,NH3/(RT)}}$$
$$R_{2-NH3} = A_{2,NH3} X_{NH3} X_{NO}^{a} e^{-E_{2,NH3/(RT)}} \quad (3494)$$

$R_1, R_2 =$ conversion rates of $NH3\left(S^{-1}\right)$

$A_{1,NH3} = 4.0 \times 10^{6}\left(S^{-1}\right)$

$A_{2,NH3} = 1.8 \times 10^{8}\left(S^{-1}\right)$

$E_{1,NH3} = 133947.2\ \mathrm{J/gmol}$

$E_{2,NH3} = 113017.95\ \mathrm{J/gmol}$

From Eqn. (3488):

$$\dot{\omega}_{HCN,\,\mathrm{consumed}} = \dot{\omega}_{HCN-1} + \dot{\omega}_{HCN-2}$$
$$= -\left(R_{1,\ HCN} + R_{2,\ HCN}\right)\rho \frac{Mw_{\ HCN}}{Mw_{\mathrm{mix}}} \quad (3495)$$

$$\dot{\omega}_{NH3,\,\mathrm{consumed}} = \dot{\omega}_{NH3-1} + \dot{\omega}_{NH3-2}$$
$$= -\left(R_{1,NH3} + R_{2,NH3}\right)\rho \frac{Mw_{NH3}}{Mw_{\mathrm{mix}}} \quad (3496)$$

From Eqns. Eqn. (3490) and Eqn. (3488):

$$\dot{\omega}_{NO,\ \mathrm{oxidation}} = \left(R_{1,\ HCN} + R_{1,\ NH3}\right)\rho \frac{Mw_{\ NO}}{Mw_{\ \mathrm{mix}}} \quad (3497)$$

$$\dot{\omega}_{NO,\ \mathrm{consumed}} = \left(R_{2,\ HCN} + R_{2,\ NH3}\right)\rho\frac{Mw_{\ NO}}{Mw_{\mathrm{mix}}} \tag{3498}$$

So putting together all equations:

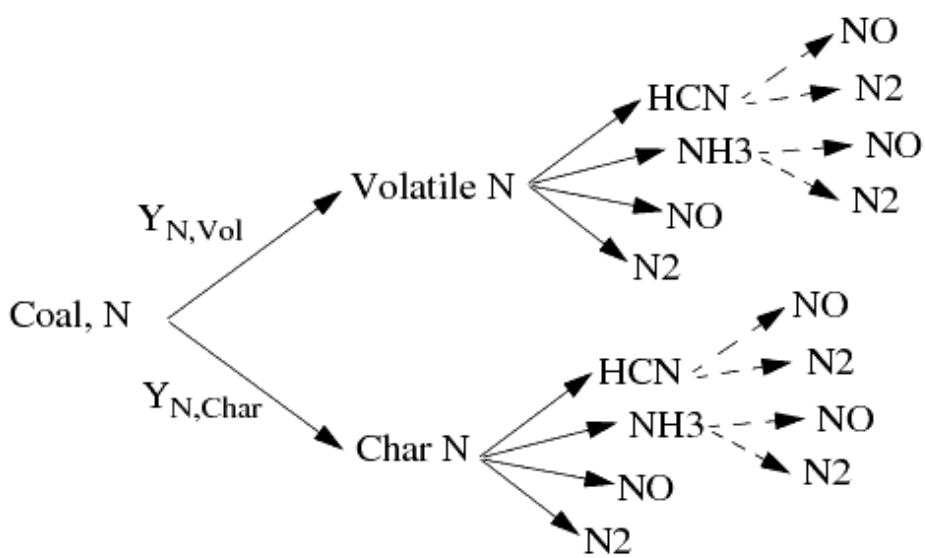
$$\dot{\omega}_{HCN} = \frac{\left(\dot{\omega}_{\mathrm{fuel}}\right)\left(Y_N\right)(CF)\left(Z_{HCN}\right)\left(Mw_{HCN}\right)}{\left(Mw_N\right)(V)} - \frac{\left(R_{1,HCN} + R_{2,HCN}\right)\rho Mw_{\ HCN}}{Mw_{\mathrm{mix}}} \tag{3499}$$

$$\dot{\omega}_{NH3} = \frac{\left(\dot{\omega}_{\mathrm{fuel}}\right)\left(Y_N\right)(CF)\left(Z_{NH3}\right)\left(Mw_{NH3}\right)}{\left(Mw_N\right)(V)} - \frac{\left(R_{1,NH3} + R_{2,NH3}\right)\rho Mw_{\ HCN}}{Mw_{\mathrm{mix}}} \tag{3500}$$

$$\dot{\omega}_{NO} = \frac{\left(\dot{\omega}_{\mathrm{fuel}}\right)\left(Y_N\right)(CF)\left(Z_{NO}\right)\left(Mw_{NO}\right)}{\left(Mw_N\right)(V)} + \frac{\left(R_{1,HCN} + R_{1,NH3} - R_{2,HCN} - R_{2,NH3}\right)\rho Mw_{\ NO}}{Mw_{\mathrm{mix}}} \tag{3501}$$

Coal NOx

Unlike the liquid fuel NOx, coal nitrogen is divided into volatile bound N and char bound N.



$$\dot{\omega}_{HCN,\ \mathrm{produced}} = \frac{\left(\dot{\omega}_{\mathrm{devol}}\right)\left(Y_{N,\ \mathrm{vol}}\right)(CF)\left(Z_{HCN}\right)\left(Mw_{HCN}\right)}{\left(Mw_N\right)(V)} + \frac{\left(\dot{\omega}_{\mathrm{char\text{-}oxi}}\right)\left(Y_{N,\ \mathrm{char}}\right)(CF)\left(Z_{HCN}\right)\left(Mw_{HCN}\right)}{\left(Mw_N\right)(V)} \tag{3502}$$

$\dot{\omega}_{HCN,\ \mathrm{consumed}} = -\left(R_{1,HCN} + R_{2,HCN}\right)\rho\frac{Mw_{HCN}}{Mw_{\mathrm{mix}}}$ (as in Eqn. (3495))

$$\dot{\omega}_{NH3,\,\mathrm{produced}} = \frac{(\dot{\omega}_{\mathrm{devol}})(Y_{N,\,\mathrm{vol}})(CF)(Z_{NH3})(Mw_{HCN})}{(Mw_N)(V)} + \frac{(\dot{\omega}_{\mathrm{char\text{-}oxi}})(Y_{N,\,\mathrm{char}})(CF)(Z_{NH3})(Mw_{HCN})}{(Mw_N)(V)} \tag{3503}$$

$\dot{\omega}_{NH3,\,\mathrm{consumed}} = -(R_{1,NH3} + R_{2,NH3})\rho\frac{Mw_{NH3}}{Mw_{\mathrm{mix}}}$ (as in Eqn. (3496))

Note that the values for CF, Z_{HCN}, Z_{NH3} and Z_{NO} can be specified differently for char and volatile.

$$\dot{\omega}_{NO,\,\mathrm{direct}} = \frac{(\dot{\omega}_{\mathrm{devol}})(Y_{N,\,\mathrm{vol}})(CF)(Z_{NO})(Mw_{NO})}{(Mw_N)(V)} + \frac{(\dot{\omega}_{\mathrm{char\text{-}oxi}})(Y_{N,\,\mathrm{char}})(CF)(Z_{NO})(Mw_{NO})}{(Mw_N)(V)} \tag{3504}$$

The values for $\dot{\omega}_{NO,\,\mathrm{oxidation}}$ and $\dot{\omega}_{NO,\,\mathrm{consumed}}$ are the same as in Eqn. (3497) and Eqn. (3498). There is some restriction in choosing values for the mass fraction of char $Y_{N,\,\mathrm{char}}$ and the mass fraction of coal $Y_{N,\,\mathrm{vol}}$—they are not independent of each other. You need ultimate and proximate analysis. Ultimate analysis indicates the percentage of elemental N in coal, that is, the mass fraction of N Y_N in dry ash free (daf) coal:

$$Y_N = Y_{N,\,\mathrm{vol}} + Y_{N,\,char} \tag{3505}$$

Proximate analysis provides the mass fractions of vol and char—along with other coal components. Always make sure that Eqn. (3505) is satisfied.

Soot

The Soot Moments and Soot Two-Equation models solve different transport equations for soot.

The formation and emission of carbonaceous particles is a process that is often observed during the combustion of hydrocarbons. These particulates, called soot, are a carbon-based material comprising mainly of polycyclic aromatic hydrocarbons (PAH's) and are identified in flames and fires as a yellow luminescence. In gas turbines, internal combustion engines, and other practical combustion devices, the formation of soot is mostly a product of incomplete combustion. Apart from the resulting loss in combustion efficiency, a particularly serious effect of soot formation is the health hazard. A typical diesel engine can emit soot at a rate of 0.2 to 0.6 g/km—driving the demand to produce cleaner engines. However, some situations require the presence or generation of soot—for example, in the production of automobile tyres where carbon black is needed. In furnaces for industrial application or in heat generators, the intermediate formation of soot is required to augment the heat transfer by radiation. The soot then has to be oxidised before the exhaust is released into the environment. These processes require a detailed knowledge of the different mechanisms leading to soot formation.

It is widely accepted that the formation of soot is a complex process which consists of

- fuel pyrolysis and oxidation reactions
- formation of polycyclic aromatic hydrocarbons
- the inception of first particles
- the growth of soot particles due to reaction with gas phase species
- the coagulation of particles
- the oxidation of soot particles and intermediates

Soot Moments

The Soot Moments emission model uses Method of Moments formulation.

In the Method of Moments formulation, the r-th of up to four moments of the soot particle size distribution function (PSDF) is identified as M_r.

The zero-th moment is related to the mean number density, whereas the first moment is related to the mean volume fraction of the soot particles. The second and third moments are related to the variance and skewness, respectively.

The transport equation for the r-th moment is:

$$\frac{\partial}{\partial t}\left(\rho M_{mr}\right) + \frac{\partial}{\partial x_j}\left(\rho u_j M_{mr} - \left(\frac{\mu}{\sigma_t} + \frac{\mu_t}{\sigma_t}\right)\frac{\partial M_{mr}}{\partial x_j}\right) = \dot{\omega}_{M_r} \tag{3506}$$

where $M_{mr}\left(= M_r/\rho\right)$ is the mass-based moment, σ_t is the turbulent Schmidt number for Soot (specified by you as the soot property), and $\dot{\omega}_{M_r}$ is the average source term.

Soot Moment Source

The general soot moment source term $\dot{\omega}_{M_r}$ is given by:

$$\dot{\omega}_{M_r} = \frac{dM_r}{dt} \tag{3507}$$

where:

$$\frac{dM_r}{dt} = R_r + G_r + W_r \tag{3508}$$

where R_r, G_r, and W_r are the instantaneous rates of particle nucleation, coagulation, and heterogeneous surface reaction terms (that is, surface growth, oxidation, and PAH condensation), respectively.

Based on the moments, the Soot characteristics are computed as follows:

- Soot Number Density $\left[1/\mathrm{m}^3\right]$:

$$N = M_0 N_A \tag{3509}$$

  where M_0 is the zero-th moment and N_A is Avogadro's number.

- Soot Volume Fraction:

$$f_v = M_1 \frac{Mw_{soot}}{\rho_{soot}} \tag{3510}$$

- Soot Mass Density $\left[\mathrm{kg}/\mathrm{m}^3\right]$:

$$M = M_1 Mw_{soot} \tag{3511}$$

- Soot Mean Diameter:

$$d = \frac{M_{1/3}}{M_0}\left(\frac{6Mw_{soot}}{\pi\rho_{soot}N_A}\right)^{1/3} \tag{3512}$$

- Soot Surface Density:

$$S = \pi N_A M_{2/3}\left(\frac{6Mw_{soot}}{\pi\rho_{soot}N_A}\right)^{1/3} \tag{3513}$$

Nucleation Source

Particle nucleation (or particle inception) is an important first step in process of soot formation. Particle nucleation results in generation of the smallest size soot particles. Nucleation is usually modeled as coalescence of two PAH species, called as soot precursors, into a soot particle. The nucleation rate for the moments is given as described by Frenklach et al. [675]:

$$R_0 = \varepsilon\sqrt{\frac{4\pi k_B T}{m_C n_{C_{PAH}}}}\, d_{PAH}^2 N_{PAH}^2 \tag{3514}$$

$$R_r = 2n_{C_{PAH}} R_{r-1} \qquad r = 1, 2, \ldots \tag{3515}$$

where:

- $n_{C_{PAH}}$ is the number of carbon atoms per PAH molecule. Assuming that the PAH species for inception is Pyrene (A4), $n_{C_{PAH}} = 16$.

- m_C is the mass of a carbon atom (12.0).
- N_{PAH} is the number density of the PAH molecules which is obtained from the molar concentration of the PAH species.
- d_{PAH} is the size of the PAH molecule in metres—which is set to 7.11 e^{-10}.

If the chemical mechanism used does not include A4 species, STAR-CCM+ allows using Acetylene (C_2H_2) as a soot precursor for nucleation. The expression for nucleation when using C_2H_2 as a soot precursor is given by:

$$R_r = 2\frac{N_A}{C_{\min}}k_N\left(T\right)\left[C_2H_2\right] \qquad r = 0 \tag{3516}$$

For $r = 1, 2, 3$:

$$R_r = 2C_{\min}R_{r-1} \tag{3517}$$

where:

- N_A is Avogadro's number.
- $C_{\min}$ is the number of carbon molecules in the smallest PAH which coagulates into a dimer to form the incipient soot particle—set as 10.

The expression for k_N is given by:

$$k_N = 0.63 \times 10^4 \exp\left(\frac{-21100}{R_u T}\right) \tag{3518}$$

PAH Condensation Source

Some PAH (A4) molecules can condense on the soot particles. The source term due to PAH condensation is modeled as described in Frenklach and Wang [677].

$$W_r^{cond} = 2.2\sqrt{\frac{\pi k_b T}{2m_C}}\sum_{k=0}^{r-1}\binom{r}{k}\left(C_h^2 M_{r-k+\frac{1}{2}}^{PAH} M_k + 2C_h C_s M_{r-k}^{PAH} M_{k+\frac{1}{3}} + C_s^2 M_{r-k-\frac{1}{2}}^{PAH} M_{k+\frac{2}{3}}\right. \tag{3519}$$

$$r = 1, 2, \ldots$$

where:

- d_a is the collision diameter of PAH. Since only one class of PAH (Pyrene, A4) is assumed to condense onto the soot surface, the PAH moments in the expression above can be reduced to:

$$M_r^{PAH} = m_{PAH}^r N_{PAH} \tag{3520}$$

Coagulation Source

G_r is evaluated by a double interpolation after inserting a mass-equivalent expression of the collision frequency factor. A general outline is presented here that is taken from Frenklach [676]. First, a grid function of order l in general form is defined as:

$$f_l^{(x,y)} = \sum_{i=1}^{\infty}\sum_{j=1}^{\infty}\left(m_i + m_j\right)^l m_i^x m_j^y\left(m_i^{\frac{1}{3}} + m_i^{\frac{1}{3}}\right)^2 N_i N_j \qquad l = 0, 1, 2, \ldots \tag{3521}$$

Then the G_r can be represented in terms of grid functions as:

$$G_0 = \frac{1}{2} K'_{fm} M_0^2 f_{\frac{1}{2}}^{(0,0)} \tag{3522}$$

$$G_r = \frac{1}{2} K'_{fm} M_0^2 \sum_{k=1}^{r-1} \binom{r}{k} f_{\frac{1}{2}}^{(k,r-k)} \qquad r = 2, 3, \ldots \tag{3523}$$

Fractional-ordered grid functions can be obtained by Lagrange interpolation of integer ordered grid functions. For example:

$$f_{\frac{1}{2}} = f_0^{\frac{3}{8}} f_1^{\frac{3}{4}} f_2^{-\frac{1}{8}} \tag{3524}$$

$f_l(l = 0, 1, 2, \ldots)$ can be evaluated in terms of fractional-order moments through interpolation. For example, upon simplification:

$$f_1 = 2\mu_{\frac{13}{6}} \mu_{\frac{1}{2}} + 4\mu_{\frac{11}{6}} \mu_{\frac{5}{6}} + 2\mu_{\frac{3}{2}} \mu_{\frac{7}{6}} \tag{3525}$$

where μ_r and fractional-order reduced moment μ is obtained by interpolating whole-ordered moments. The coefficient K'_{fm} is given by:

$$K'_{fm} = \varepsilon \sqrt{\frac{6k_B T}{\rho_s}} \left(\frac{3m_C}{4\pi\rho_s} \right) \tag{3526}$$

Surface Growth and Oxidation

Heterogeneous surface reactions, such as surface growth and oxidation, contribute to soot particle growth. It is generally accepted that acetylene (C_2H_2) is the main participating gas-phase species, and the surface hydrogen-abstraction-C_2H_2-addition (HACA, Table below) mechanism usually is invoked.

The soot surface growth is modeled using the HACA mechanism (Appel et al. [671]), summarized below:

Reaction Number	Reaction
1	$\mathrm{C_{soot} - H + H \rightleftharpoons C_{soot}\bullet + H_2}$
2	$\mathrm{C_{soot} - H + OH \rightleftharpoons C_{soot}\bullet + H_2O}$
3	$\mathrm{C_{soot}\bullet + H \rightarrow C_{soot} - H}$
4a	$\mathrm{C_{soot}\bullet + C_2H_2 \rightarrow C_{soot} - H + H}$
4b	$\mathrm{C_{soot}\bullet + C_2H_2 \rightarrow C_{soot}\bullet + H}$
5	$\mathrm{C_{soot}\bullet + O_2 \rightarrow 2CO + products}$
6	$\mathrm{C_{soot}\bullet + OH \rightarrow CO + products}$

In these reactions, $\mathrm{C_{soot} - H}$ represents a reactive site on the surface of the soot particle, and $\mathrm{C_{soot}\bullet}$ denotes the corresponding radical. The reaction rates are computed as Eqn. (3301).

The soot kinetic model parameters are given as (Appel et al. [671]):

k_i	A_i $cm^3mol^{-1}s^{-1}$	β_i	E_i kcal/mol
k_1	4.2×10^{13}	0.0	13.0
k_{-1}	3.9×10^{12}	0.0	11.0
k_2	1.0×10^{10}	0.734	1.43
k_{-2}	3.68×10^{8}	1.139	17.1
k_3	2.0×10^{13}	0.0	0.0
k_{4a}	8.0×10^{7}	1.56	3.8
k_5	2.2×10^{12}	0.0	7.5
k_6	$\gamma_{OH=0.13}$	0.0	0.0

The reactions with the gas-phase species are modeled as:

$$W_r^{C_2H_2} = k_{4a}\left[C_2H_2\right]\alpha\chi_{Csoot} \bullet \pi C_s^2 M_0 \sum_{l=0}^{r-1}\binom{r}{l}\mu_{l+\frac{2}{3}}(2)^{r-l} \tag{3527}$$

$$W_r^{O_2} = k_5[O_2]\alpha\chi_{Csoot} \bullet \pi C_s^2 M_0 \sum_{l=0}^{r-1}\binom{r}{l}\mu_{l+\frac{2}{3}}(-2)^{r-l} \tag{3528}$$

$$W_r^{OH} = \gamma_{OH}[OH]\sqrt{\frac{\pi k_B T}{2m_{OH}}}N_A C_s^2 M_0 \sum_{l=0}^{r-1}\binom{r}{l}\mu_{l+\frac{2}{3}}(-1)^{r-l} \tag{3529}$$

where:

- α is the steric factor—the fraction of the soot surface sites available for reaction.
- m_{OH} is the mass of the OH radical.
- γ_{OH} is the collision efficiency for the OH radical.

The rate of the surface reaction depends on the particle surface area and the number of active reaction (radical) sites available. The number density of active radical sites can be obtained from the total number density of all radical sites on the soot surface using a steady-state approximation. The total number density of all radical sites is estimated to be 2.5×10^{15} cm^{-2}.

$$C_s = \left(\frac{6m_C}{\pi\rho_s}\right)^{\frac{1}{3}} \tag{3530}$$

$\chi_{Csoot\bullet}$ is the number density of the surface radicals, given by:

$$\chi_{Csoot\bullet} = \chi_{Csoot-H}\frac{k_1[H] + k_2[OH]}{k_{-1}[H_2] + k_{-2}[H_2O] + k_3[H] + k_{4a}[C_2H_2] + k_5[O_2]} \tag{3531}$$

where $\chi_{Csoot\bullet}$ is the nominal number density of the soot radical sites, for which a value of 2.3×10^{19} m^{-2} is prescribed. In the present work, the reaction pathway 4a from the table is considered, and 4b is ignored.

The parameter α is difficult to quantify from a theoretical point of view and can have a marked influence on the soot levels. Steric factor α is defined as the fraction of the reactive sites ('armchair' sites) on the surface

of the soot particle that are available for surface growth or oxidation reactions. The value of α ranges from zero to unity.

Various values and functional forms for α are found in literature. STAR-CCM+ provides three ways to compute α: a constant value, a premixed temperature-based correlation, and as a user-defined scalar.

The premixed temperature-based correlation is taken from Appel et al. [671] who fitted the correlation for α, where α was a function of the local temperature and the average soot particle size, quantified by the reduced soot moment μ_1. In their work, α was calculated as:

$$\alpha = \tanh\left(\frac{a}{\log \mu_1} + b\right) \tag{3532}$$

where the fitted parameters are:

$a = 3.0 \times (12.65 - 0.00563T)$

$b = -1.38 + 1.02 \times 10^{-2}T$

Soot Two-Equation

The soot two-equation model developed here is based on the modeling framework provided by [674] and [684].

This model is a semi-empirical soot model where the overall soot emission has been modeled with four key physical processes:

- Nucleation
- Coagulation
- Surface-growth
- Oxidation

This model is particularly useful for non-premixed turbulent jet flames.

Although soot number density and mass density are the primary variables of interest, for numerical convenience, scaled variables are preferred for the transport equations:

- Scaled soot number density (N) [/m3]

  Solving for normalized number density $\varnothing_N$, the soot number density is expressed as:

$$\varnothing_N = \frac{N}{\rho N_A} \tag{3533}$$

- Scaled soot mass density (M) [kg/m3]

  The soot mass density is expressed as:

$$M = \rho \varnothing_M \tag{3534}$$

The generic transport equation for scalar $\varnothing$ is of the form:

$$\frac{\partial \rho \varnothing}{\partial t} + \frac{\partial \rho u_j \varnothing}{\partial x_j} = \frac{\partial}{\partial x_j}\left(D \frac{\partial \varnothing}{\partial x_j}\right) + \dot{\omega}_{\varnothing} \tag{3535}$$

For scaled soot number density, $\varnothing = N$

For scaled soot mass density, $\varnothing = \varnothing_M$

The soot number density and mass density are Soot Two-Equation Source terms which are expressed as:

$$\dot{\omega}_{\varnothing N} = \frac{d\rho\varnothing_N}{dt} = \frac{1}{N_A}\frac{dN}{dt} \tag{3536}$$

$$\dot{\omega}_{\varnothing M} = \frac{d\rho\varnothing_M}{dt} = \frac{dM}{dt} \tag{3537}$$

Physical processes

The source terms $\dot{\omega}_{\varnothing N}$ and $\dot{\omega}_{\varnothing M}$ are split into four physical processes:

- Nucleation

  One nucleation submodel is based on polycyclic aromatic hydrocarbons (PAH), and may be expressed as:

$$\begin{aligned}\left(\frac{dN}{dt}\right)_{nu} &= 8 \cdot c_2 \frac{N_A}{M_P}\left[\rho^2\left(\frac{Y_{C_2H_2}}{Mw_{C_2H_2}}\right)^2 \frac{Y_{C_6H_5} Mw_{H_2}}{Mw_{C_6H_5} Y_{H_2}}\right] e^{-\frac{4378}{T}} \\ &+ 8 \cdot c_3 \frac{N_A}{M_P}\left[\rho^2 \frac{Y_{C_2H_2} Y_{C_6H_6} Y_{C_6H_5} Mw_{H_2}}{Mw_{C_2H_2} Mw_{C_6H_6} Mw_{C_6H_5} Y_{H_2}}\right] e^{-\frac{6390}{T}}\end{aligned} \tag{3538}$$

  where c_2 and c_3 are model constants with the following values:

  - $c_2 = 127 \times 10^{8.88}$
  - $c_3 = 178 \times 10^{9.50}$

  See [684].

  Alternatively, an acetylene-based nucleation submodel can be used which requires the soot precursor C2H2. If C2H2 is specified in the chemical mechanism or species list—and also listed as a species for tabulation when using a flamelet model, C2H2 concentration is provided by the combustion model. When C2H2 is not provided by the combustion model, the C2H2 concentration is calculated empirically from the mixture fraction.

  The nucleation rate for soot number density is expressed as:

$$\left(\frac{dN}{dt}\right)_{nu} = C_1 N_A \rho \frac{Y_{C2H2}}{Mw_{C2H2}} e^{\frac{-21000}{T}} \tag{3539}$$

  The nucleation rate for mass density is expressed as:

$$\left(\frac{dM}{dt}\right)_{nu} = \frac{M_p}{N_A}\left(\frac{dN}{dt}\right)_{nu} \tag{3540}$$

  $M_p = 1200 \frac{kg}{kmol}$

  $N_A = 6.023 \times 10^{26}/kmol$

  $C_1 = 54/s$

- Coagulation

$$\left(\frac{dN}{dt}\right)_{co} = -\left(\frac{24R_u}{\rho_{soot}N_A}\right)^{1/2}\left(\frac{6}{\pi\rho_{soot}}\right)^{1/6}T^{1/2}M^{1/6}N^{11/6} \tag{3541}$$

$\rho_{soot} = 2000\frac{kg}{m^3}$

R is the universal gas constant.

$\left(\frac{dM}{dt}\right)_{co} = 0$

- Soot surface growth (requires C2H2)

$\left(\frac{dN}{dt}\right)_{sg} = 0$

$$\left(\frac{dM}{dt}\right)_{sg} = C_4\rho\frac{Y_{C2H2}}{M_{W,\ C2H2}}e^{\frac{-12100}{T}}\left[(\pi N)^{1/3}\left(\frac{6M}{\rho_{soot}}\right)^{2/3}\right] \tag{3542}$$

$C_4 = 9000.6\frac{kg.\mathrm{m}}{kmol.\mathrm{s}}$

- Soot oxidation (requires O2 and OH)

$\left(\frac{dN}{dt}\right)_{ox} = 0$

$$\left(\frac{dM}{dt}\right)_{ox} = -C_5\ \rho\,\eta\frac{Y_{OH}}{Mw_{OH}}T^{\frac{1}{2}}\left[(\pi N)^{\frac{1}{3}}\left(\frac{6M}{\rho_{soot}}\right)^{\frac{2}{3}}\right] - C_6\ \rho\frac{Y_{O2}}{Mw_{O2}}T^{\frac{1}{2}}\ e^{\frac{-19778}{T}}\left[(\pi N)^{\frac{1}{3}}\left(\frac{6M}{\rho_{soot}}\right)^{\frac{2}{3}}\right] \tag{3543}$$

$\eta = 0.13$

$C_5 = 105.81\frac{kg.\mathrm{m}}{kmol.\mathrm{K}^{1/2}s}$

$C_6 = 8903.51\frac{kg.\mathrm{m}}{kmol.\mathrm{K}^{1/2}s}$

The total source terms for number density and mass are:

$$\frac{dN}{dt} = \left(\frac{dN}{dt}\right)_{nu} + \left(\frac{dN}{dt}\right)_{co} + \left(\frac{dN}{dt}\right)_{sg} + \left(\frac{dN}{dt}\right)_{ox} \tag{3544}$$

$$\frac{dM}{dt} = \left(\frac{dM}{dt}\right)_{nu} + \left(\frac{dM}{dt}\right)_{co} + \left(\frac{dM}{dt}\right)_{sg} + \left(\frac{dM}{dt}\right)_{ox} \tag{3545}$$

Equations Eqn. (3544) and Eqn. (3545) are substituted in Eqn. (3536) and Eqn. (3537) to obtain the source terms for the transport equations.

The two-equation model developed here allows you to specify your own rates for the four physical processes (Nucleation, Coagulation, Surface growth and Oxidation).

The generalized framework for the computation of source term for $\varnothing = \varnothing_N, \varnothing_M$ is:

$$\left(\dot{\omega}_{\varnothing}\right)_{total} = A_1 + A_2\dot{\omega}_{\varnothing} \tag{3546}$$

A_2 is the scale factor that needs to be specified for all four processes, and A_1 is the user-specified rate. If A_1=0 and A_2=1, the total source term is computed from equations Eqn. (3539) to Eqn. (3543). By setting A_2 =0 you can disable the STAR-CCM+ inbuilt source computation. You can then implement your own source

term (throughA_1) by specifying the rates for each of the physical processes. When you opt to specify your own rates, Jacobians that you can specify values for are also included as additional options for better numerical stability.

Species Options

By default, the user-specified rate option is disabled and all of the scaling factors are set to 1.0. The complete description of the physical processes requires specification of the mass fraction of a few species. These species may or may not be the part of the reaction set up. So choices are provided for you to compute the species concentrations. These options are available for O, OH, and C2H2.

Fluid Streams

Based on the fluid stream definition, an algebraic expression is used to compute the local mixture fraction with Bilger's mixture fraction definition [673]:

$$Z = \frac{\frac{2\left(Z_C - Z_{C,ox}\right)}{Mw_C} + \frac{\left(Z_H - Z_{H,ox}\right)}{2Mw_H} - \frac{\left(Z_O - Z_{O,ox}\right)}{Mw_O}}{\frac{2\left(Z_{C,f} - Z_{C,ox}\right)}{Mw_C} + \frac{\left(Z_{H,f} - Z_{H,ox}\right)}{2Mw_H} - \frac{\left(Z_{O,f} - Z_{O,ox}\right)}{Mw_O}} \tag{3547}$$

z refers to the elemental mass fraction. The C, H, and O subscripts refer to elemental carbon, hydrogen, and oxygen. The f and ox subscripts refer to the fuel and oxidizer streams.

Soot Absorption Coefficient

When the participating radiation model is active, soot can contribute to the overall gas absorption coefficient for the cell. You can set the absorption coefficient as a constant (default value zero), as a field function, or as the Planck Mean Absorption Coefficient k_a in units of 1/m.

$$k_a = \frac{C_0 f_v T}{0.0037535} \tag{3548}$$

where:

- C_0 is a constant you can set, with a default value of 4.9.
- f_v is the soot volume fraction from Eqn. (3510).
- T is the temperature.

When accounting for all participating media, the sum of the absorption coefficient is calculated by Eqn. (2119).

Particle Reactions

Discrete-phase (Lagrangian, DEM) particles, such as multi-component solid particles or coal particles, can react with the surrounding gases to form either gaseous or solid products. STAR-CCM+ provides multiple models which allow you to simulate different types of particle reactions.

Particle Chemistry

Particle Devolatilization

Devolatilization is a gasification process where a solid is converted to volatile matter and a residue.

For each devolatilization reaction, only one solid component can be chosen as a reactant. Any number of products can be chosen as long as there is at least one solid and one gas-phase product in each reaction. The most general representation of a devolatilization reaction is:

$$A_{(s)} \rightarrow B1_{(g)} + ... + Bn_{(g)} + C1_{(s)} + ... + Cn_{(s)} \tag{3549}$$

First Order Rate Formulation

In the First Order Rate method, the reaction rate coefficient k_{devol} (1/s) is of the Arrhenius form given by:

$$k_{devol} = AT^{\beta} e^{\left(\frac{-E_a}{R_u T}\right)} \tag{3550}$$

The rate (kg/s) of consumption of the devolatilizing solid component is given by:

$$\frac{dm_i}{dt} = -k_{devol} \alpha_i m \tag{3551}$$

In the User Reaction Rate method, you specify the reaction rate (kg/s) directly:

$$\frac{dm_i}{dt} = -r_{user\ rate} \tag{3552}$$

Particle Combustion

First-Order Combined Rate Formulation

In the First-Order Rate method, the reaction rate coefficient k_{char} (m/s) is of the Arrhenius form given by:

$$k_{char} = AT^{\beta} e^{\left(\frac{-E_a}{R_u T}\right)} \tag{3553}$$

The rate of consumption of the solid component is dictated by the combined effect of the Arrhenius rate and gas-reactant diffusion rate to the particle surface given by [669]:

$$\frac{dm_i}{dt} = -\frac{k_m k_{char}}{k_{char} + k_m} \phi C_g M_w A_p \tag{3554}$$

where:

$$k_m = \frac{(Sh)(D_m)}{d} \tag{3555}$$

Half-Order Combined Rate Formulation

In the Half-Order Rate method, the reaction rate coefficient is of the Arrhenius form given by Eqn. (3301).

The rate of consumption of the solid component is dictated by the combined effect of the Arrhenius rate and gas-reactant diffusion to the particle surface given by:

$$\frac{dm_i}{dt} = k_m C_g M_w \phi A_p - \frac{k_m r^2}{\left(M_w \phi A_p\right) k_{char}^2} \tag{3556}$$

In the User Reaction Rate method, you specify the reaction rate directly:

$$\frac{dm_i}{dt} = -r_{user\ rate} \tag{3557}$$

Coal Combustion

A coal particle consists of four components: raw coal, char, ash, and moisture. Ash represents the chemically inert part of the particle. Char is the residue that is left when the volatile matter is released from the raw coal, plus that portion of the particle that cannot undergo devolatilization.

Coal Moisture Evaporation

The moisture content of the coal particle is assumed to coat the particle. Moisture evaporation occurs rapidly before other mass transfer processes take place.

The continuity equation for the moisture component in particle p is as follows:

$$\frac{d\alpha_{wp} m_p}{dt} = -r_{wp} \tag{3558}$$

The moisture evaporation rate r_{wp} required in Eqn. (3558) is obtained by solving for both heat and mass transfer. The formulation for this evaporation rate is identical to that of the Quasi-Steady Single-Component Droplet Evaporation model. In coal combustion, the heat transfer rate commonly controls the evaporation rate.

Raw Coal Devolatilization

Using this mechanism is a reasonable compromise between a simple one-step mechanism, which does not account for the effect of particle history on volatile yield, and a more complex, multi-step mechanism which increases computational time.

The continuity equation for the raw coal component in particle p is described as:

$$\frac{dm_{cp}}{dt} = -r_{cp} \tag{3559}$$

The net rate of raw coal consumption is given by:

$$r_{cp} = k_{pn} \alpha_{cp} m_p \tag{3560}$$

For the User-Defined model, you specify r_{cp} as the User Devolatilization Rate.

The raw coal, that is, the dry, ash-free portion of the coal, undergoes devolatilization to form volatiles and char by one or more reactions. The nth reaction is represented by:

$$\left(raw\ coal\right)_p \xrightarrow{k_{pn}} YY_{pn}\left(volatile\ matter\right)_g + \left(1-YY_{pn}\right)\left(char\right)_p \tag{3561}$$

The volatile matter consists of combustible hydrocarbons and oxygen, and may also contain inorganic species (for example, N, Cl, and S). This is represented by the CoalVolatile component in the gas phase.

The kinetic rate of production of volatiles from the nth reaction (Eqn. (3561)) is given by:

$$r_{vpn} = k_{pn} YY_{pn} \alpha_{cp} m_p \tag{3562}$$

where the reaction rate constant k_{pn} is given by:

$$k_{pn} = A_{pn} e^{\left(-\frac{E_{pn}}{R_u T_p}\right)} \tag{3563}$$

The two-step mechanism [648] is recommended for computing the devolatilization rate.

Char Oxidation

Raw coal devolatilization produces char according to Eqn. (3561). The continuity equation for the char component in particle p is as follows:

$$\frac{dm_{hp}}{dt} = -r_{hp} \tag{3564}$$

Char reacts heterogeneously, after diffusion of the reactants (that is, O2, CO2 and H2O) to the particle surface, by one or more reactions of the form:

$$\left(char\right)_l + \left(oxidizer\right)_l \xrightarrow{k_{pl}} \left(products\right)_l \tag{3565}$$

Depending on the char oxidation model that is selected, the oxidation reaction is assumed to be first-order or half-order with respect to the oxidizer concentration. Three oxidizers are currently considered for char reactions: O_2, H_2O and CO_2. These correspond to the following reactions:

$$\varphi C + O_2 \rightarrow \left(2\varphi - 2\right)CO + \left(2-\varphi\right)CO_2 \tag{3566}$$

$$C + H_2O \rightarrow CO + H_2 \tag{3567}$$

$$C + CO_2 \rightarrow 2CO \tag{3568}$$

First-Order Char Oxidation

A simple mass balance results in an expression for the rate of formation of char from particle p of the nth devolatilization reaction as shown below:

$$r_{hpn} = r_{vpn}\left(1-YY_{pn}\right)/YY_{pn} \tag{3569}$$

The net rate of raw coal consumption is the rate of char oxidation plus volatile formation.

Char is assumed to react heterogeneously with gaseous oxidizers that diffuse to the particle surface from the bulk gas phase. Two rate-limiting steps are considered for this process:

- Gas phase diffusion
- Heterogeneous reaction

The surface reaction is assumed to be first-order in the surface concentration of the oxidizer. The Arrhenius expression for the rate constant is then given by:

$$k_{pl} = A_{pl} T_p^n e^{\left(-\frac{E_{pl}}{R_u T_p}\right)} \tag{3570}$$

The overall combined char reaction rate is calculated from the following implicit equation [669]:

$$r_{hpl} = \frac{A_p^2 M_{hp} M_g \varphi_l k_{cpl} k_{pl} \zeta_{pl} C_{olg} C_g}{M_g A_p C_g \left(\zeta_{pl} k_{pl} + k_{cpl}\right) + r_p} \tag{3571}$$

The mass transfer coefficient without the correction from blowing, k_{cpl}, is computed from the Sherwood number, which in turn is based on the Ranz-Marshall correlation. The value of particle pore area coefficient, ζ_{pl}, has been chosen as 1.0. Since the char oxidation reaction is zero-order with respect to char concentration, heterogeneous ignition is permitted.

Half-Order Char Oxidation

Char is assumed to react heterogeneously with gaseous oxidizers that diffuse to the particle surface from the bulk gas phase. Two rate-limiting steps are considered for this process:

- Gas phase diffusion
- Half-Order Heterogeneous reaction

The surface reaction is assumed to be half-order in the surface concentration of the oxidizer. The Arrhenius expression for the rate constant is then given by:

$$k_{pl} = A_{pl} T_p^n e^{\left(-\frac{E_{pl}}{R_u T_p}\right)} \tag{3572}$$

The overall half-order combined char reaction rate for reaction l is calculated from the following implicit equation:

$$r_{hpl} = \phi_l k_{cpl} A_p M_{hp} \left[C_{olg} - \left(\frac{r_{hpl}}{\phi_l k_{pl} A_p M_{hp} \zeta_{pl}} \right)^2 \right] \tag{3573}$$

The mass transfer coefficient without the correction from blowing, k_{cpl}, is computed from the Sherwood number, which in turn is based on the Ranz-Marshall correlation. The value of particle pore area coefficient, ζ_{pl}, has been chosen as 1.0.

Effect of Particle Porosity

STAR-CCM+ also provides the optional Lagrangian model, Particle Porosity, which allows you to define how the porosity of particles which burn internally changes through the course of a coal combustion reaction. The porosity of a particle, θ, determines how the diameter and surface area of the particle changes throughout a reaction. When the surface area of a particle changes, the overall reaction rate is affected. Using the Particle

Porosity model, you can control how the reacting particle varies by specifying the exponent α which is used to determine porosity according to Eqn. (3182). The porosity is then used to calculate the particle volume, Eqn. (3183), which is used to calculate the particle diameter, Eqn. (3184), using the apparent density Eqn. (3185). Particle diameter, d_p, is then used to calculate the surface area of the particle, A_p.

Polymerization

Polymerization is a chemical process in which small molecules combine to form larger molecules that contain repeating structural units of the original molecules.

Typically, in a solvent, monomer molecules combine together in the presence of an initiator together in a chemical reaction to form polymer chains or 3D networks. Polymerization involves many reaction steps such as initiation, propagation, branching, termination. Many radicals of different chain lengths are formed. The final product is a mixture containing polymer molecules of varying lengths and structure.

This feature is based on the Method of Moments Model for polymerization. Primary properties like chain length distribution of polymer molecules are obtained by classical method of moments.

STAR-CCM+ solves for the scalar transport of three Live polymers, λ_0, λ_1, and λ_2, and three Dead polymers, μ_0, μ_1, and μ_2. In addition to the source terms described below, STAR-CCM+ allows you to add moment sources that you define. STAR-CCM+ also solves for the scalar transport of the Initiator, Monomer, Radical, Solvent, and Modifier.

The source terms of the moment transport equations depend on the sub processes of polymerization:

- Initiator Decomposition

$$I_i \xrightarrow{k_{di}} 2R^{\cdot}, \qquad i = 1, 2, \ldots, N_i \tag{3574}$$

- Chain Initiation

$$R^{\cdot} + M \xrightarrow{k_I} R_1 \tag{3575}$$

where,

the initiator consumption rate is found using:

$$r_{Ii} = -k_{di}\left[I_i\right], i = 1 \tag{3576}$$

and the primary radical formation rate is found using:

$$r_{R^{\cdot}} = \sum_{i=1}^{N_d} 2 f_i k_{di}\left[I_i\right] - k_I\left[R^{\cdot}\right]\left[M\right] \tag{3577}$$

where,

f_i = initiator efficiency

- Propagation

$$R_x + M \xrightarrow{k_p} R_{x+1} \tag{3578}$$

where the monomer consumption rate is found using:

$$r_M = -k_p[M]\lambda_0 \tag{3579}$$

- Chain transfer to monomer

$$R_x + M \xrightarrow{k_{tm}} P_x + R_1 \tag{3580}$$

- Chain transfer to solvent

$$R_x + S \xrightarrow{k_{ts}} P_x + R_1 \tag{3581}$$

where the solvent consumption rate is found using:

$$r_S = -k_{ts}[S]\lambda_0 \tag{3582}$$

- Chain transfer to polymer

$$R_x + P_y \xrightarrow{k_{ts}} R_y + P_x \tag{3583}$$

- β-Scission

$$R_x + P_y \xrightarrow{k_\beta} P_x + R_z + P^{=}_{y-z} \tag{3584}$$

- Reaction with terminal double bond

$$R_x + P_y^{=} \xrightarrow{k_{db}} R_{x+y} \tag{3585}$$

- Termination by disproportionation

$$R_x + R_y \xrightarrow{k_{td}} P_x + P_y \tag{3586}$$

- Termination by combination

$$R_x + R_y \xrightarrow{k_{tc}} P_{x+y}. \tag{3587}$$

STAR-CCM+ solves scalar transport equations for moments as follows:

- For Live Polymers

  λ_0 is solved using r_{λ_0} as the source term

$$\lambda_0 = \sum_{n=1}^{\infty} R_n \tag{3588}$$

where r_{λ_0} is defined by:

$$r_{\lambda_0} = \sum_{i=1}^{N_d} 2fk_{d_i}[I_i] - (k_{tc} + k_{td})\lambda_0^2 \tag{3589}$$

λ_1 is solved using r_{λ_1} as the source term

$$\lambda_1 = \sum_{n=1}^{\infty} nR_n \tag{3590}$$

where r_{λ_1} is defined by:

$$r_{\lambda_1} = \sum_{i=1}^{N_d} 2fk_{d_i}[I_i] - \left(k_{tm}[M] + k_{ts}[S]\right)\lambda_1 + \tag{3591}$$

$$k_p[M]\lambda_0 - \left(k_{tc} + k_{td}\right)\lambda_0\lambda_1 + k_{tp}\left(\lambda_0\mu_2 - \lambda_1\mu_1\right) +$$

$$k_{db}\lambda_0\mu_1 + k_\beta\left(\frac{1}{2}\lambda_0\mu_2 - \lambda_1\mu_1\right)$$

λ_2 is solved using r_{λ_2} as the source term

$$\lambda_2 = \sum_{n=1}^{\infty} n^2R_n \tag{3592}$$

where r_{λ_2} is defined by:

$$r_{\lambda_2} = \sum_{i=1}^{N_d} 2fk_{d_i}[I_i] - \left(k_{tm}[M] + k_{ts}[S]\right)\lambda_2 + \tag{3593}$$

$$2k_p[M]\lambda_1 - \left(k_{tc} + k_{td}\right)\lambda_0\lambda_2 + k_{tp}\left(\lambda_0\mu_3 - \lambda_2\mu_1\right)$$

$$+ k_{db}\left(2\lambda_1\mu_1 + \lambda_0\mu_2\right) + k_\beta\left(\frac{1}{3}\lambda_0\mu_3 - \frac{1}{2}\lambda_0\mu_2 + \frac{1}{6}\lambda_0\mu_1 - \lambda_2\mu_1\right)$$

- For Dead Polymers

  μ_0 is solved using r_{μ_0} as the source term

$$\mu_0 = \sum_{n=1}^{\infty} P_n \tag{3594}$$

  where r_{μ_0} is defined by:

$$r_{\mu_0} = \left(k_{tm}[M] + k_{ts}[S]\right)\lambda_0 + \left(\frac{1}{2}k_{tc} + k_{td}\right)\lambda_0^2 - k_{db}\lambda_0\mu_0 + k_\beta\lambda_0\mu_1 \tag{3595}$$

μ_1 is solved using r_{μ_1} as the source term

$$\mu_1 = \sum_{n=1}^{\infty} nP_n \tag{3596}$$

where r_{μ_1} is defined by:

$$r_{\mu_1} = \left(k_{tm}[M] + k_{ts}[S]\right)\lambda_1 + \left(k_{tc} + k_{td}\right)\lambda_0\lambda_1 - k_{tp}\left(\lambda_0\mu_2 - \lambda_1\mu_1\right) - k_{db}\lambda_0\mu_1 + k_\beta\left(\lambda_1\mu_1 - \frac{1}{2}\lambda_0\mu_2\right) \tag{3597}$$

μ_2 is solved using r_{μ_2} as the source term

$$\mu_2 = \sum_{n=1}^{\infty} n^2 P_n \tag{3598}$$

where r_{μ_2} is defined by:

$$r_{\mu_2} = \left(k_{tm}[M] + k_{ts}[S]\right)\lambda_2 + \left(k_{tc} + k_{td}\right)\lambda_0\lambda_2 + k_{tc}\lambda_1^2 - k_{tp}\left(\lambda_0\mu_3 - \lambda_2\mu_1\right) - k_{db}\lambda_0\mu_2 + k_\beta\left[\lambda_2\mu_1 - \lambda_0\left(\frac{2}{3}\mu_3 + \frac{1}{2}\mu_2 - \frac{1}{6}\mu_1\right)\right] \tag{3599}$$

The closure of the higher moment, μ_3, is adopted from Pladis and Kiparissides [670].

In addition to the source terms and their jacobians for polymer scalars and polymer moments that are mentioned above, you can also include user-defined rates and jacobians for polymer scalar and moments sources.

Based on the specified Heat of Propagation and Heat of Initiation, appropriate source terms are added to the energy equation.

By solving the transport of the above six Moment equations, you can obtain information about the polymer molecular structure, such as Polydispersity and the Molecular Weight Distribution of the polymer. STAR-CCM+ provides post-processing field functions for Polydispersity Index, Polymer Number-Averaged, and Mass-Averaged Chain Lengths. All Live and Dead Moments are also available as post-processing field functions.

Polydispersity Index is given by:

$$\frac{\mu_2\mu_0}{\mu_1^2} \tag{3600}$$

Polymer Number-Averaged Chain Length (NACL) is given by:

$$\frac{\mu_1}{\mu_0} \tag{3601}$$

Polymer Mass-Averaged Chain Length (MACL) is given by:

$$\frac{\mu_2}{\mu_1} \tag{3602}$$

Reacting Flow Nomenclature

a_i	Activity of species i calculated by Eqn. (3317).
A	Helmholtz free energy
A_j	Pre-exponential factor $\left[\left(\left[\mathrm{kmol}/\left(\mathrm{m}^3\right)\right]^{\left(1-P_j\right)}\right)/\mathrm{s}\right]$ The units for the pre-exponent A_j change according to the order of the reaction (sum of p_{ij}) and the temperature dependency. For a bi-molecular reaction where n = 0, A is of unit [cm3/mole/s].
A_p	Surface area of the particle $\left(\mathrm{m}^2\right)$
A_{pl}	Arrhenius pre-exponential factor for char oxidation reaction l ($m/(s-K^n)$ where n is the order of the reaction)
A_{pn}	Arrhenius pre-exponential factor for devolatilization reaction n (1/s)
α	Coefficient imported in TROE mechanism / Mass fraction of coal component
α_i	Mass fraction of ith component in particle
α_{ji}	Collision efficiency of the third body in a reaction
b	Regress variable
β	Temperature exponent / an Arrhenius parameter
c	Normalized progress variable
C_g	Overall gas concentration (kmol/m3)
C_{olg}	Concentration of oxidizer (kmol/m3)
C_p	Heat capacity at constant pressure
C_V	Heat capacity at constant volume
d	Particle diameter / Soot Mean Diameter $\left(\frac{6M}{\pi N \rho_{soot}}\right)^{\frac{1}{3}}$ (m)
D_m	Diffusion coefficient $\left(\mathrm{m}^2/\mathrm{s}\right)$
E_a	Activation energy $[\mathrm{J}/\mathrm{kmol}]$

$\exp\left(-E_{aj}/RT\right)$	$[\text{dimensionless}]$
f	Initiator frequency
f_i	Initiator efficiency
$\mathrm{f_v}$	Soot volume fraction $\left(\frac{M}{\rho_{soot}}\right)$
F	Blending function / thickening factor
F_{CFL}	CFL scaling factor
G	Gibbs free energy
γ	Heat loss ratio (enthalpy)
h	Enthalpy
H	Enthalpy
I	Initiator
$\emptyset_N$	Soot scaled number density
$\emptyset_M$	Soot scaled mass density
k	Reaction rate coefficient (units vary: Char Oxidation (m/s) and Devolatilization (1/s))
k_{cpl}	Mass transfer coefficient
k_m	Mass transfer coefficient (m/s)
K	Equilibrium constant
K_a	Planck Mean Absorption Coefficient
K_f	User-defined reaction coefficient $\left[\left(\left[\mathrm{kmol/m^3}\right]^{\left(1-N_j\right)}\right)/\mathrm{s}\right]$
λ_i	Live polymer with ith moment
m	Total mass in a CFD cell
$\dot{m}$	Mass flux in a CFD cell
m_i	Mass of ith component in particle / total mass of species i in a mixture
m_p	Particle mass
M	Soot mass density
M	Monomer / mean molecular mass

M_W	Molecular Weight, kg
$\overline{M_W}$	Mean molecular weight of a mixture, kg
$M_{W,soot}$	Soot Molecular Weight (default value = 24 kg/kmol)
$M_{W,g}$	Molecular weight of the gas mixture
$M_{W,hp}$	Char molecular weight (kg/kmol)
M_r	r -th Moment
μ_i	Dead polymer with ith moment / chemical potential of species i
μ_t	Turbulent viscosity
n_i	Number of moles of species i in a mixture
N	Soot number density
N_A	Avogadro Number (6.02214179^{26} $1/\mathrm{kmol}$)
N_j	Sum of all ν_i for reaction j $[\mathrm{dimensionless}]$
ζ_{pl}	Particle pore area coefficient
P	Pressure / dead polymer
$P(z,b;\underline{x},t)$	Joint pdf of the mixture fraction z and regress variable b
$P_b(b;\underline{x},t)$	A marginal pdf
P_j	Sum of all p_{ij} for reaction j $[\mathrm{dimensionless}]$
P_n	Dead polymer with n monomer units
P_r	Reduced pressure in pressure dependant reactions
$P_z(z\,\vert\,b;\underline{x},t)$	A conditional pdf on b
ϕ	Ratio of stoichiometric coefficients of solid and gas reactants
Φ	Equivalence ratio
Ψ	Stoichiometric mass ratio
$\widetilde{Q}$	Favre average of quantity Q
$\widetilde{Q}^u(\underline{x},t)$	Favre averaged quantity at unburnt state
$\widetilde{Q}^b(\underline{x},t)$	Favre averaged quantity at burned state
r	Reaction rate (kg/s)

$\widehat{r_i}$	Net molar rate of change (production or consumption) of species i (kg/s)
r_k	Reaction rate / rate of consumption of solid reactant (kg/s)
r_{vpn}	Production rate of volatiles for particle p and nth devolatilization step (kg/s)
r_{hpl}	Overall combined char reaction rate (kg/s)
r_{hpn}	Production rate of char for particle p and nth devolatilization step (kg/s)
r_p	Total reaction rate of coal particle p (kg/s)
r_{wp}	Moisture evaporation rate (kg/s)
R	Live Polymer
$R^{\cdot}$	Radical
$\mathbf{R}$	Reaction $[\mathrm{dimensionless}]$
R_j	Final reaction rate [kmol/(s m3)]
R_n	Live polymer with n monomer units
R_u	Universal gas constant 8314.4621 $[\mathrm{J/kmol\ K}]$
ρ	Density $\left[\mathrm{kg/m^3}\right]$
ρ_{soot}	Soot Density (default value = 1800 $\mathrm{kg/m^3}$)
$\left(\rho Y_i\right)/M_i$	Molar concentration $\left[\mathrm{kmol/m^3}\right]$
s_c	Sticking coefficient for surface reactions
S	Solvent (polymerization) / Soot Surface Density (1/m) / Entropy
Sc	Turbulent Schmidt number
S_E	Source term
S_i	Chemical source term
Sh	Sherwood number
t	Physical time
T	Temperature $[\mathrm{K}]$
T^{β_j}	Temperature to the exponent β_j $[\mathrm{dimensionless}]$
T_p	Particle temperature
τ	Time scale $[\mathrm{s}]$

τ_{mix}	Turbulent mixing time scale $[\mathrm{s}]$
τ_{kin}	Time scale that is derived from chemical reaction rate from finite-rate kinetics $[\mathrm{s}]$
τ_{res}	Residence time in the cell $[\mathrm{s}]$
τ_{turb}	Turbulent time scale $[\mathrm{s}]$
Γ_{tot}	Total surface site concentration [mole/m²]
U	Total internal energy $[J]$
V	Volume
υ_{ij}	Stoichiometric coefficient of reacting species i in reaction j
φ_l	Stoichiometric coefficient for the char reaction
$\underline{x}$	Position vector
X_i	Mole fraction of species i $[\mathrm{dimensionless}]$
y	Unnormalized progress variable
$\widetilde{y}$	Progress variable variance
$\mathbf{Y}$	Mass fraction vector
YY_{pn}	Mass stoichiometric coefficient for particle devolatilization reaction, n
Y_i	Mass fraction of species i $[\mathrm{dimensionless}]$
Ω_i	Source term of species i $\left[\mathrm{kg}/\left(\mathrm{s\ m}^3\right)\right]$
ω_i	Species source term
ω_k	Reaction source term
Z	Mixture fraction
$\widetilde{Z}$	Mixture fraction variance

Particle Chemistry Subscripts

w	moisture component
c	raw coal component
h	char component
v	volatile matter
n	devolatilization step
l	char oxidation reaction

g	gas phase
p	particle phase

Reacting Flow Bibliography

Adiabatic PPDF Bibliography

[642] *Jones, W.P. 1980. 'Prediction methods for turbulent flames', in "Prediction Methods for Turbulent Flow" (Ed. W. Kollman). Hemisphere, Washington, D.C., pp. 1-45.*

[643] *Peters, N. 1986. 'Laminar flamelet concepts in turbulent combustion', 21st Symp. (Int.) on Combustion, The Combustion Institute, pp. 1231-1250.*

[644] *Pitsch, H., Desjardins, O., Balarac, G. and Ihme, M. 2008. "Large-Eddy Simulation of Turbulent Reacting Flows", Prog. Aerospace Sci., pp. 466–478.*

[645] *Veynante D. and Vervisch L. 2002. 'Turbulent Combustion Modeling' in 'Progress in Energy and Combustion Science', Vol. 28, pp. 248-252.*

Analytical Jacobian Libraries

[646] *Lu, T. 2012. "Computational Tools for Diagnostics and Reduction of Detailed Chemical Kinetics", 2012 Princeton-CEFRC Summer School on Combustion, Princeton University.*

Clustering

[647] *Babajimopoulos, 1., Assanis, D. N., Flowers, D. L., Aceves, S. M., Hessel, R. P., "A fully coupled computational fluid dynamics and multi-zone model with detailed chemicalkinetics for the simulation of premixed charge compression ignition engines", Int. Journal of Engine Research, 2005, vol 6 no 5*

Coal

[648] *Kobayashi, H., Howard, J.B., and Sarofim, A.F. 1977. "Coal Devolatilization at High Temperatures", 18th Symposium (International) on Combustion, The Combustion Institute, Pittsburgh, PA, p. 411.*

[649] *Smoot, D.J., and Smith, P.J. 1985. "Coal Combustion and Gasification", The Plenum Chemical Engineering series, New York.*

Coherent Flame

[650] *Meneveau, C., and Poinsot, T. 1991 'Stretching and quenching of flamelets in premixed turbulent combustion', Combust. Flame, 86, pp. 311-332.*

CVODE

[651] *S.B. Pope, V. Hiremath, S.R. Lantz, Z. Ren and L. Lu (2012) ISAT-CK7: "A Fortran 90 library to accelerate the implementation of combustion chemistry"*

http://tcg.mae.cornell.edu/ISATCK7

Dynamic Mechanism Reduction

[652] *T. Lu, C. Law, Proc. Combust. Inst. 30 (2005) 1333-1341.*

EBU

[653] *B. F. Magnussen and B. H. Hjertager."On mathematical models of turbulent combustion with special emphasis on soot formation and combustion". In 16th Symp. (Int'l.) on Combustion. The Combustion Institute. 1976.*

Eddy Contact Micromixing

[654] *Forney, L. J., and Nafia, N. 2000. “Eddy Contact Model: CFD Simulations of Liquid Reactions in Nearly Homogeneous Turbulence”, Chem. Eng. Sci., 55(24), pp. 6049-6058.*

[655] *Hjertager, L.K., Hjertager, B.H., Solberg, T., 2002. “CFD modelling of fast chemical reactions in turbulent liquid flows”, Computers and Chem. Eng. 26 pp. 507–515.*

Eddy Dissipation Concept

[656] *B. F. Magnussen, "The Eddy Dissipation Concept a Bridge between Science and Technology," in ECCOSMAS Thematic Conference on Computational Combustion, June 21-24, 2005. Lisbon, Portugal.*

[657] *B.F. Magnussen and B.H. Hjertager 'On the structure of turbulence and a generalized eddy dissipation concept for chemical reaction in turbulent flow', 19th AIAA Aerospace Meeting, St. Louis, USA, 1981.*

[658] *I. S. Ertesvag and B. Magnussen, "The Eddy Dissipation Turbulence Energy Cascade Model," Combustion Science and Technology, vol.159, pp.213-235, 2000.*

FGM

[659] *J.A. van Oijen, L.P.H. de Geoy. Modelling of premixed laminar flames using flamelet-generated manifolds. Combust. Sci. Technol., 161:113, 2000.*

[660] *H. Lehtiniemi, F. Mauß, M. Balthasar, and I. Magnusson. Modeling diesel spray ignition using detailed chemistry with a progress variable approach. Combust. Sci. and Tech. 178 (10-11) (2006) 1977-1997.*

[661] *G. Goldin, Y. Zhang, "A GENERALIZED FGM PROGRESS VARIABLE WEIGHT OPTIMIZATION USING HEEDS" , Proceedings of ASME Turbo Expo 2017: Turbomachinery Technical Conference and Exposition GT2017-64446, June 26-30, 2017, Charlotte, NC, USA.*

ISAT / Partially Stirred Reactor

[662] *Pope, S.B. 1997. “Computationally efficient implementation of combustion chemistry using in situ adaptive tabulation”, Combust. Theory and Modelling, 1, pp. 41-63.*

Laminar Flame Concept

[663] *R. J. Kee, F. M. Rupley, and J. A. Miller, "Chemkin-II: A Fortran Chemical Kinetics Package for the Analysis of Gas-Phase Chemical Kinetics" Sandia National Laboratories Report SAND89- 8009, (1990).*

Laminar Flame Speed

[664] *Metghalchi, M. and Keck, J.C. 1982 'Burning velocities of mixtures of air with methanol, isooctane, and indelene at high pressure and temperature', Combust. Flame, 48, pp. 191-210.*

[665] *Gülder, Ö.L. 1991 'Turbulent premixed flame propagation models for different combustion regimes', Proceedings of The Combustion Institute, Vol.23, pp. 743-750.*

Non-Adiabatic PPDF Bibliography

[666] *Smoot and P. Smith, Coal Combustion and Gasification, The Plenum Chemical Engineering Series, 1985.*

NOx

[667] *Baulch, D. L., Drysdall, D. D., Horne, D. G., and Lloyd, A. C. 1973. Evaluated Kinetic Data for High Temperature Reactions, vols. 1-3. Butterworth.*

[668] *De Soete, G. G. 1975. "Overall Reaction Rates of NO and N2 Formation from Fuel Nitrogen", in 15th Symp. (Intl.) on Combustion, pp. 1093-1102. The Combustion Institute.*

Particle Reactions

[669] *Smoot, D.J. and Smith, P.J. 1985. "Coal Combustion and Gasification", The Plenum Chemical Engineering series, New York.*

Polymerization

[670] *Pladis and Kiparissides, Chem Eng Sc, 53, No.18, pp 3315, 1998*

Soot

[671] *Appel, J., H. Bockhorn, and M. Frenklach (2000) "Kinetic Modeling of Soot Formation with Detailed Chemistry and Physics: Laminar Premixed Flames of C2 Hydrocarbons," Combustion and Flame, 122, pp. 122-136*

[672] *Balthasar, M., Mauss, F., and Wang, H. 2002. "A computational study of particle thermal ionization and its effect on soot mass growth in laminar premixed flames", Comb. & Flame, 129, pp. 204-216.*

[673] *Bilger, R.W. 1988. "The structure of turbulent nonpremixed flames", Proc. Combust. Inst. 23, pp. 475–488.*

[674] *Brookes, S. J. and Moss, J. B. 1999. "Predictions of soot and thermal radiation in confined turbulent jet diffusion flames," Comb. & Flame, 116, pp. 486-503.*

[675] *Brown, N., K. Revzan, and M. Frenklach (1998) "Detailed Kinetic Modeling of Soot Formation in Ethylene/Air Mixtures Reacting in a Perfectly Stirred Reactor," Proc. Combust. Inst., 27, pp. 1573 - 1580.*

[676] *Frenklach, M. (2002) "Reaction Mechanism of Soot Formation in Flames," Phys. Chem. Chem. Phys., 4, pp. 2028 - 2037.*

[677] *Frenklach, M. and H. Wang (1994) "Detailed Mechanism and Modeling of Soot Particle Formation," in Soot Formation in Combustion: Mechanisms and Models (H. Bockhorn, ed.), Springer-Verlag, New York.*

[678] *Kazakov, A. and Frenklach, M. 1998. “Dynamic Modeling of Soot Particle Coagulation and Aggregation: Implementation With the Method of Moments and Application to High-Pressure Laminar Premixed Flames”, Comb. & Flame, 114, pp. 484-501.*

[679] *Kazakov, A., Wang, H. and Frenklach, M. 1995. “Detailed Modeling of Soot Formation in Laminar Premixed Ethylene Flames at a Pressure of 10 Bar”, Comb. & Flame, 100, pp. 111-120.*

[680] *Lindstedt, R. P. and S. A. Louloudi (2005) "Joint-scalar transported PDF modeling of soot formation and oxidation," Proc. Combust. Inst., 30, pp. 775 - 783.*

[681] *Mauss, F., Trilken, B., Breitbach, H., and Peters, N. 1994. “Soot Formation in Partially Premixed Diffusion Flames at Atmospheric Pressure”, Soot Formation in Combustion: Mechanisms and Models, H. Bockhorn ed., Springer Verlag, pp. 325-349.*

[682] *Modest, Michael F., 2003. Radiative Heat Transfer, Academic Press.*

[683] *Nakov, G., Mauss, F., Wenzel, P., Steiner, R. et al., "Soot Simulation under Diesel Engine Conditions Using a Flamelet Approach," SAE Int. J. Engines 2(2):89-104, 2010.*

[684] *Wen, Z., Yun, S., Thomson, M. J., and Lightstone, M. F. 2003. "Modeling soot formation in turbulent kerosene/air jet diffusion flames," Comb. & Flame, 135, pp. 323-340.*

Spark Ignitor

[685] *JM Duclos, O.Colin, “Arc and Kernel Tracking Ignition Model for 3D Spark Ignition Engine Calculations” COMODIA 2001, IFP 343-350*

[686] *S.Falfari, GM Bianchi “Development of an Ignition Model for SI Engines Simulation”, SAE paper 2007-01-0148*

[687] *J.Song, M.Sunwoo “A Modeling and Experimental Study of Initial Flame Kernel Development and Propagation in SI Engines”, SAE paper 2000-01-0960*

[688] *StarCD V4.28 Methodology Guide*

TFC

[689] *Zimont, V.L., Polifke, W., Bettelini, M., and Weisenstein, W. 1998. "An Efficient Computational Model for Premixed Turbulent Combustion at High Reynolds Numbers Based on a Turbulent Flame Speed Closure", ASME J. Eng. Gas Turbines and Power, 120(3), pp. 526-532.*

[690] *Peters, N. 2000. Turbulent Combustion, Cambridge University Press, Cambridge.*

Thickened Flame

[691] *Blint, R. 1986. "The relationship of the laminar flame width to flame speed", Combust. Sci. and Tech., 49:79.*

[692] *Charlette, F., Meneveau, C. and Veynante, D. 2002. "Wrinkling Model for LES of Premixed Turbulent Combustion Part I: Non–Dynamic Formulation and Initial Tests", Combust. Flame, 131:159.*

[693] *Colin, O., Ducros, F., Veynante, D. and Poinsot, T. 2000. "A Thickened Flame Model for Large Eddy Simulation of Turbulent Premixed Combustion", Phys. Fluids, 12:1843.*

[694] *Durand, L. and Polifke, W. 2007. "Implementation of the Thickened Flame Model for Large Eddy Simulation of Turbulent Premixed Combustion in a Commercial Solver", ASME Paper No. GT2007-28188.*

[695] *Legier, J.P., Poinsot, T. and Veynante, D. 2000. "Dynamically thickened flame LES model for premixed and non-premixed turbulent combustion", Proceedings of the Summer Program 2000, Center for Turbulence Research, pp. 157-168.*

Electrochemistry

Electrochemistry is the discipline investigating relationships between electrical currents and chemical composition change in general.

Well-known electrochemical phenomena, research fields, and applications include the initially discovered galvanism (muscle stimulation through electric currents), electrophysiology (neuroscience), and corrosion. Also, electrochemical material removal methods such as etching and electrochemical machining, material deposition methods like electrodeposition, galvanization and synthesis, and last but not least, fuel cells and batteries.

A variety of approaches can be used for electrochemical modelling. Simple reaction models which assume uniform species concentrations can predict reaction rates solely depending on the electric potential situation. More detailed models solve transport equations for the different kinds of species that are involved in the reactions, and consider their contributions to the rate of the reactions. Equal for all modelling variants, the transport of both ions and charge-neutral reaction components must be modelled before any chemical conversion can take place—regardless if the conversion is of chemical or electriochemical nature. This fundamental circumstance is represented throughout the structure of this theory section.

STAR-CCM+ implements electrochemistry models widely generically and based on first principles, to assert their applicability independent of any specific application. For certain applications where tailored solutions provide better results, like the lithium ion battery model, specific solutions exist as well.

You can use the electrochemistry models in STAR-CCM+ to determine the evolution of electrochemical reaction rates, currents, species concentrations, and fluxes, or to investigate derived parameters like the ionic space charge density.

Contents:

Transport of Electrochemical Reaction Components
Conversion of Electrochemical Reaction Components
Model Simplifications / Engineering Approaches
Electrochemistry Bibliography
Electrochemistry Nomenclature

Transport of Electrochemical Reaction Components

Before reactions can be defined and parameterized for any reactive flow, the chemical composition of a fluid mixture needs to be known. This holds equally true for electrochemical reactions.

Electrochemical reactions in STAR-CCM+ can convert reaction components from both regular species, and electrochemical species mixtures.

Species

The term, species, refers to any component of a chemical reaction. For example, in the hydrogen oxidation reaction:

$$\mathrm{H_2} + \frac{1}{2}\mathrm{O_2} \rightarrow \mathrm{H_2O} \tag{3603}$$

there are three species: two reactant species—hydrogen ($\mathrm{H_2}$) and oxygen ($\mathrm{O_2}$), and one product species—water ($\mathrm{H_2O}$). In STAR-CCM+, the electrically neutral species that you define using the material models, such as Liquid or Multi-Component Gas, are known as species.

Electrochemical Species

Electrically charged species, such as ions, or charged particles consisting of several molecules, are modeled using the electrochemical species model. Ions are atoms, groups of atoms, or molecules, that are charged because they contain a greater or lesser number of electrons than protons. If the number of electrons orbiting a nucleus is greater than the number of protons in that nucleus, this particle is referred to as a negatively charged ion. Conversely, if the number of electrons orbiting a nucleus is less than the number of protons, this particle is referred to as a positively charged ion.

When considering electrochemical reactions, the term, electrochemical species, refers to charged species such as ions, that participate in an electrochemical reaction. For example, in the reaction:

$$\mathrm{H^+} + \mathrm{OH^-} \rightarrow \mathrm{H_2O} \tag{3604}$$

there are two electrochemical species, the hydrogen ion ($\mathrm{H^+}$, proton) and the hydroxide ion ($\mathrm{OH^-}$).

Being charged matter, electrochemical species are affected by Coulomb forces and are therefore transported by electric fields. This process is called migration. As such, electrochemical species modelling can be useful even in the absence of electrochemical reactions, when species shall be separated, or pumped without moving parts interacting with the fluid.

The mass balance equations that are listed in the following sections determine the spatial distribution of reaction components within an electrochemically reacting system. These balances are dependent on relations that describe the electric potential and velocity, and usually require electroneutrality to be preserved. The velocity is generally determined by Fluid Flow. The governing equations for electric potential can vary depending on the existence of significant electrical current densities—as outlined in the following sections.

Contents:

Electrochemical Species

Segregated Species

Electrochemical Species

Electrochemical species are used to describe the electrically charged components of a fluid mixture, both gaseous or liquid. As such, they describe the ionized electrolytes within a non-ionized solvent, or the ions that exist within a plasma.

Electrolytic Solutions

According to Newman [161], electrolytes can be described as either ideal or dilute solutions, moderately dilute solutions, and concentrated solutions. The following sections outline the differences between these, and the models which are used in STAR-CCM+ to describe them.

Transport mechanisms

Electrochemical species are transported by three dominant transport mechanisms: advection, diffusion, and migration (sometimes also referred to as drift flux). While advection describes transport that is induced by velocities, diffusion describes the transport that is due to gradients in species concentrations. Migration, or drift flux, describes transport that is due to electric potential gradients, and is often related to the concept of transference. While in general all three transport mechanisms occur simultaneously, many systems lend themselves to simplifications which ease the numerical solution process. Also, these transport models can be described using different methodologies, this leads to slight variations in equation sets in the following sections.

The electric potentials that are necessary to describe migration fluxes are determined using Eqn. (3702). In general, the frequency of many electrochemical and plasma currents is sufficiently low such that magnetic effects can be ignored and the electrostatic potential model Eqn. (3724) can be used. Furthermore, many systems that require electroneutral mixtures allow for an incompressible description in the evolution equation for electric potential, making use of ohms law. For these cases, the electrodynamic potential model Eqn. (3727) can be used.

In any case, STAR-CCM+ allows you the freedom to set the physical coupling between electrochemical species and electric potential.

Contents:

Dilute Solutions
Concentrated Solutions
Coulomb Forces

Dilute Solutions

Dilute electrolytic solutions consist of minor quantities of ionic species which exist in a non-ionized solvent, excessing the ions. Transport models for such solutions model the transport primarily independent of the interactions between different kinds of ions, that is, only dependent on the concentration and electric potential gradients of the electrochemical species under consideration (diffusion, migration).

Following the dilute assumption, electrochemical species mixtures which are considered by the electrochemical species and solid ion models in STAR-CCM+ will not affect the density of a mixture.

Coupling Electrochemical Species and Electrodynamic Potential Models

The equations in this section are primarily solved when modeling applications which enforce electroneutrality within the solution, such as batteries, corrosion, etching, and electrochemical deposition. For this case, the electrodynamic potential model provides the electric potential in the transport equation for electrochemical species.

The transport equation for the molar concentrations c_i of the electrochemical species i, can be written as:

Continuous form:

$$\frac{\partial}{\partial t}\left(c_i\chi\right) = -\nabla \cdot \mathbf{N}_i \tag{3605}$$

$$\mathbf{N}_i = -D_i\chi^\beta\nabla c_i - z_iFu_ic_i\chi^\beta\nabla\phi + c_i\mathbf{v} \tag{3606}$$

Finite Volume or integral form:

$$\frac{\partial}{\partial t}\int_V \chi c_i \mathrm{d}V + \oint_A c_i \mathrm{v} \cdot \mathrm{da} = \oint_A D_i\chi^\beta\nabla c_i \cdot \mathrm{da} + \oint_A z_iFu_ic_i\chi^\beta\nabla\phi \cdot \mathrm{da} + \int_V S_{c_i}\mathrm{d}V \tag{3607}$$

here, χ^β represents the porosity and N_i represents the molar concentration flux of species component i.

The convective term $c_i\mathbf{v}$ uses the superficial velocity for porous regions, which is advection velocity that is multiplied by porosity.

When the Electrochemical Species model is active, the Electrodynamic Potential model solves a transport equation for the ionic space charge density ρ_{ion}:

Continuous form:

$$\frac{\partial}{\partial t}\left(\rho_{ion}\chi\right) = -\nabla \cdot \mathbf{J} = 0 \tag{3608}$$

$$\mathbf{J} = -\kappa_i\chi^\beta\nabla\phi - F\sum_i z_iD_i\chi^\beta\nabla c_i + \rho_{ion}\mathbf{v} \tag{3609}$$

The first term on the right-hand side here represents the electrical current due to migration, while the second term represents the diffusion current density. Assuming electroneutrality ($\rho_{ion} = 0$), the third term vanishes and the finite volume, or integral form reads:

$$0 = \int_A \kappa_i\chi^\beta\nabla\phi \cdot \mathrm{da} + F\sum_i z_i\int_A D_i\chi^\beta\nabla c_i \cdot \mathrm{da} + \int_V S_{\rho_{ion}}\mathrm{d}V \tag{3610}$$

where $S_{\rho_{ion}}$ represents user-defined sources. Substituting expressions for κ_i, the ionic conductivity of species i:

$$\kappa_i = F^2\sum_i z_i^2u_ic_i \tag{3611}$$

the Nernst-Einstein relation,

$$u_i = \frac{D_i}{R_u T} \tag{3612}$$

and the relation between the ionic space charge density and molar concentrations,

$$\rho_{ion} = F\sum_i \; z_i c_i \tag{3613}$$

the following flux expressions are derived for the molar concentration flux of species component i and the electrical current density $\mathbf{J}$:

$$\mathbf{N}_i = -D_i \, \chi^\beta \nabla c_i - z_i F \frac{D_i}{R_u T} \chi^\beta c_i \nabla \phi + c_i \mathbf{v} \tag{3614}$$

$$\mathbf{J} = -F^2 \sum_i \; z^2{}_i \frac{D_i}{R_u T} \chi^\beta c_i \nabla \phi - F\sum_i \; z_i D_i \chi^\beta \nabla c_i + F\sum_i \; z_i c_i \mathbf{v} \tag{3615}$$

Comparison of Eqn. (3614) and Eqn. (3615) produces a relation between the molar species fluxes and the electric current densities:

$$\mathbf{J} = F\sum_i \; z_i \mathbf{N}_i \tag{3616}$$

Due to the electroneutrality ($\rho_{ion} = 0$), that is generally assumed in liquid electrolytes, the convective term in Eqn. (3609) is zero.

Coupling Electrochemical Species and Electrostatic Potential Models

When electroneutrality cannot be assumed, such as in ionic winds, or in certain kinds of plasmas, the Electrochemical Species and Electrostatic Potential models need to be coupled. The electrostatics potential equation,

$$\nabla \cdot \mathbf{E} = \rho_{ion} / \varepsilon_0 \tag{3617}$$

replaces Eqn. (3608) to determine the electric potential.

The continuous form is written as:

$$\oint_A \; \varepsilon_0 \nabla \phi \cdot \mathrm{da} = \int_V \; \rho_{ion} \mathrm{d}V + \int_V \; S_{Estat} \mathrm{d}V \tag{3618}$$

Here, ρ_{ion} is non-zero and is computed from Eqn. (3613).

The transport equations for the electrochemical species are then often expressed in terms of a species number density:

$$\frac{\partial}{\partial t}\left(n_P \chi\right) = -\nabla \cdot \mathbf{N}_P \tag{3619}$$

$$\mathbf{N}_P = -n_P \mathrm{K}_P \chi^\beta \nabla \phi - D_P \chi^\beta \nabla n_P + n_P \mathbf{v} \tag{3620}$$

See Cagnoni et al. (2013) [152].

The species number density n_P is equal to the molar concentration of the ionic species c multiplied by Avogadro's number. Therefore, number density represents the number of ionic species particles per volume.

Often, only a single ionic species with a unity charge number ($z = 1$) is considered.

For a single ionic species i, the relation:

$$q\, n_P = \rho_{ion} = qN_A c = Fc \tag{3621}$$

(by definition of Faraday's constant) is used to rewrite Eqn. (3619) and Eqn. (3620) as a transport equation for charge density:

$$\frac{\partial}{\partial t}\left(\rho_{ion}\chi\right) = -\nabla \cdot \mathbf{J} \tag{3622}$$

$$\mathbf{J} = -\rho_{ion}\mathrm{K}\chi^{\beta}\nabla\phi - D\chi^{\beta}\nabla\rho_{ion} + \rho_{ion}\mathbf{v} \tag{3623}$$

as in Neimarlija et al. (2009) [159].

You can also obtain Eqn. (3622) and Eqn. (3623) by multiplying Eqn. (3605) and Eqn. (3606) with Faraday's constant F, and charge number z_i, applying Eqn. (3613), and substituting ionic mobility as:

$$\mathrm{K}_i = Fz_i u_i \tag{3624}$$

Although K_i and u_i are commonly termed ion mobility, their dimension and units differ by the dimension of Faraday's constant ($\left[\mathrm{m}^2/\mathrm{V\,s}\right]$ vs $\left[\mathrm{m}^2\mathrm{kmol}/\mathrm{J\,s}\right]$).

Concentrated Solutions

Concentrated solution theory is considered crucial for the modelling of Lithium-Ion batteries, where electrolytes need to satisfy lowest -resistivity requirements and thus cannot be in the infinitely dilute regime. The Li-Ion Battery Cell model in STAR-CCM+ is based on concentrated solution theory.

According to Newman [161], dilute solution theory can be extended to include interactions between all species in an electrolyte. This involves:

- electrochemical potentials
- activity coefficient variations
- variation of ionic diffusion coefficients with concentration

Moderately dilute solution theory accounts for electrochemical potentials and activity coefficients, whereas concentrated solution theory accounts additionally for concentration-dependent diffusion coefficients. The basis of concentrated solution theory is given by Stefan-Maxwell diffusion.

Similar to set-ups that use the electrochemical species model, the Li-Ion Battery Cell model interacts with electric potentials, that are provided here by the Li-Ion Electric Potential model. The species transport is described by the Li-Ion Concentration model, which basically assumes an electroneutral electrolyte—consisting of a solvent that embeds a binary salt which decomposes into a cation and an anion. As a consequence, only a single Lithium/Salt Concentration scalar is solved for, which is understood to be the concentration of either the salt, the cation, the anion, or the intercalated Lithium (when in the solid electrodes).

In the solid electrodes and current collectors, the electric current density is defined using Ohm's law. A simple species diffusion transport equation is solved for lithium (Lithium/Salt Concentration) transport within the electrode phase.

$$\mathbf{J} = \sigma \mathbf{E} \tag{3625}$$

where $\mathbf{E} = -\nabla\phi$.

Within the electrolyte phase, the electric current density $\mathbf{J}$ is due to diffusion and migration of ions (salt). Using concentrated solution theory [163], $\mathbf{J}$ can be written as

$$\mathbf{J} = \kappa\mathbf{E} - \frac{\kappa}{F}\left(\frac{s_+}{n\nu_+} + \frac{{t_+}^0}{z_+\nu_+} - \frac{s_0 c}{n c_0}\right)\nabla\mu_e \tag{3626}$$

where the first term of the right-hand side corresponds to migration, and the second term to diffusion. The effective electrical conductivity κ in the liquid phase accounts for an eventual non-unity porosity χ and tortuosity τ within the separator.

$$\kappa = \frac{\chi}{\tau}\kappa_0 \tag{3627}$$

Assuming a binary lithium based electrolyte to parametrize Eqn. (3626) (see [163] and [155]):

$$\mathbf{J} = \kappa\mathbf{E} + \frac{2R_u T\kappa}{F}\left(1 - {t_+}^0\right)\left[1 + \frac{\mathrm{d}\left(\ln f_\pm\right)}{\mathrm{d}\left(\ln c\right)}\right]\nabla\left(\ln c\right) \tag{3628}$$

or

$$\mathbf{J} = \kappa\mathbf{E} + \kappa_\mathrm{d}\nabla\left(\ln c\right) \tag{3629}$$

where $\kappa_\mathrm{d} = \frac{2R_u T\kappa}{F}\left(1 - {t_+}^0\right)\left[1 + \frac{\mathrm{d}\left(\ln f_\pm\right)}{\mathrm{d}\left(\ln c\right)}\right]$.

STAR-CCM+ uses this electrical current density definition for the electrolyte that is used in the Li-Ion Battery Cell Model.

Due to charge conservation, the electric current density field must be divergence free:

$$\nabla \cdot \mathbf{J} = 0 \tag{3630}$$

Applying Eqn. (3625) and Eqn. (3629) (the electric current density definition) to Eqn. (3630), integration over a control volume, and subsequent application of Gauss's theorem yields the transport equations for electric potential:

$$\oint_S \sigma\nabla\phi \cdot \mathrm{d}\,\mathbf{s} = 0 \tag{3631}$$

within the (solid) electrode and current collector phases, and:

$$\oint_S \kappa\nabla\phi \cdot \mathrm{d}\,\mathbf{s} = \oint_S \kappa_\mathrm{d}\nabla\left(\ln c\right) \cdot \mathrm{d}\,\mathbf{s} \tag{3632}$$

within the (liquid) electrolyte phase.

A simple species diffusion transport equation is solved for lithium (Lithium/Salt Concentration) transport within the electrode phase:

$$\frac{\partial}{\partial t}\int_V c \, \mathrm{d}\, V = \oint_S D \nabla c \cdot \, \mathrm{d}\, \mathbf{s} \tag{3633}$$

where D is a constant that is defined by Eqn. (3635).

An additional transport term for lithium cation migration is considered in the (liquid) electrolyte phase, so that:

$$\frac{\partial}{\partial t}\int_V \chi c \, \mathrm{d}\, V = \oint_S D \nabla c \cdot \, \mathrm{d}\, \mathbf{s} - \int_V \frac{\mathbf{J} \cdot \nabla t_+^0}{F} \, \mathrm{d}\, V \tag{3634}$$

The Effective Lithium/Salt Diffusivity Coefficient D accounts for non-unity values of porosity χ and tortuosity τ within the separator, and considers the *Concentrated Solution Correction to Salt Diffusion Coefficient*, which is written as:

$$D = D_0 \frac{\chi}{\tau}\left[1 - \frac{d\left(\ln c_0\right)}{d\left(\ln c\right)}\right] \tag{3635}$$

Convective transport in the electrolyte is not modeled.

Electric Potential Interfacing Condition

Conservation of the electric charge that is exchanged across the Solid Electrolyte Interface (SEI) gives the interface condition for the Boundary Specific Electric Current:

$$j_{n,s} = -j_{n,l} \tag{3636}$$

where $j_{n,s}$ is the Boundary Specific Electric Current on the electrode (solid):

$$j_{n,s} = \mathbf{J}_s \cdot \mathbf{n}_s \tag{3637}$$

and $J_{n,l}$ is the Boundary Specific Electric Current on the electrolyte (liquid):

$$j_{n,l} = \mathbf{J}_s \cdot \mathbf{n}_l \tag{3638}$$

The Boundary Specific Electric Current is the SEI normal component of the Electric Current Density.

The normal vector always points outward of the considered electrode or electrolyte domain:

$$\mathbf{n}_s = -\mathbf{n}_l \tag{3639}$$

Using the definition for the specific electric current (Eqn. (3636)), the potential boundary condition on the electrode side is:

$$\left.\frac{\partial \phi}{\partial n}\right|_{\mathrm{SEI}} = -\frac{j_{n,s}}{\sigma} \tag{3640}$$

Using Eqn. (3640), the potential boundary condition on the electrolyte side is:

$$\left.\frac{\partial \phi}{\partial n}\right|_{\mathrm{SEI}} = -\frac{j_{n,l}}{\kappa} + \frac{\kappa_d}{\kappa}\frac{\partial\left(\ln c_1\right)}{\partial n} \tag{3641}$$

Concentration Interfacing Condition

By conservation of the flux of Lithium $\mathbf{N}_+$ across the SEI,

$$\mathbf{N}_+ \cdot \mathbf{n}_s = -\mathbf{N}_+ \cdot \mathbf{n}_l \tag{3642}$$

On the electrolyte side, the flux of lithium ions is expressed as

$$\mathbf{N}_+ = \frac{\mathbf{J}}{F} = -D\ \Delta\ c\frac{1}{1-t_+{}^0} \tag{3643}$$

Taking the inner product of Eqn. (3643) with the SEI normal vector $\mathbf{n}_l$, the interfacing condition reads:

$$\left.\frac{\partial c}{\partial n}\right|_{\mathrm{SEI}} = -\frac{j_{n,l}}{FD}\left(1-t_+{}^0\right) \tag{3644}$$

On the solid electrode side, the lithium non-ionic flux is expressed as:

$$\mathbf{N}_+ = \frac{\mathbf{J}}{F} = -D\ \Delta\ c \tag{3645}$$

Taking the inner product of Eqn. (3645) with $\mathbf{n}_s$, the concentration interfacing condition becomes:

$$\left.\frac{\partial c}{\partial n}\right|_{\mathrm{SEI}} = -\frac{j_{n,s}}{FD} \tag{3646}$$

Coulomb Forces

Once the electrochemical species model is selected in STAR-CCM+, the coulomb force model can be selected to include the forces acting on the flow field due to the electrical charge carriers.

The Coulomb force that is exerted on ionic species in an electric field leads to a volumetric force term which is written as:

$$\mathrm{f}_{Coulomb} = \rho_{ion}\mathrm{E} \tag{3647}$$

See Cagnoni et al., (2013) [152].

The ionic space charge density here is computed from Eqn. (3613) rather than considering a single ionic species as in Eqn. (3621).

Segregated Species

While widely used in electrochemical modelling, segregated species transport is not specific to electrochemistry but also to mixing and combustion problems.

For more details, see Species Transport.

Conversion of Electrochemical Reaction Components

Once appropriate transport models describe the spatial distributions of species and electrons in a flow field, these potential reaction components can be used to parameterize reactions.

Both electrically neutral and electrically charged species can represent reactants or products in electrochemical reactions or in charge neutral reactions (see Chemical Kinetics, Bulk Ion Chemical Reactions and Sorption).

Electrochemical reactions are strictly heterogeneous—they always involve reactants and products from at least two distinct phases.

In iron corrosion for example, solid iron can react with electrons from the same solid phase to form dissolved iron cations in the adjacent solution phase. Reactions in fuel cells occur at three-phase boundaries. This tendency preferentially also holds true for batteries where electrons are provided from a solid conductive aid phase to convert lithium—that is intercalated in an active material phase—into dissolved lithium cations in an electrolyte phase.

In STAR-CCM+, electrochemical reactions are modelled on the two-dimensional surfaces of solid electrodes, however, besides electrochemical reactions, the reactants and products can undergo further reactions of chemical nature within the fluid volume of an electrolyte. The following sections cover both the electrochemical reactions, as well as sorption and bulk chemical reactions of electrochemical species.

Contents:

Electrochemical Surface Reactions (Heterogeneous)
Lithium-Ion Model Reactions
Surface Effects in Ionic Winds
Sorption
Bulk Ion Chemical Reactions (Homogeneous)

Electrochemical Surface Reactions (Heterogeneous)

Electrochemical surface reactions either produce electrons (anodic) or consume electrons (cathodic). Any number of these reactions can occur at any given time at the interface between a metal and an electrolyte.

A general form of electrochemical surface reactions is represented as:

$$j_{n,s} = f(\eta)\ ;\ \begin{Bmatrix} j_{n,s} > 0 \\ \eta > 0 \end{Bmatrix} \equiv \text{anodic and} \begin{Bmatrix} j_{n,s} < 0 \\ \eta < 0 \end{Bmatrix} \equiv \text{cathodic} \tag{3648}$$

The electrochemical reaction rate $j_{n,s}$ is given in terms of the specific electric current that is exerted by the reaction. The boundary species electrochemical reaction flux of a reaction component i is:

$$N_{n,s,i} = \frac{\nu_i}{F\nu_e} j_{n,s} \tag{3649}$$

where ν_i is the stoichiometric coefficient of the reaction component, ν_e is the stoichiometric coefficient of the electron, and F is Faraday's constant.

Reaction formulations provide functional forms which specify the dependence of the specific reaction current density, that is the electrochemical reaction rate $j_{n,s}$, on the surface overpotential η:

$$\eta = \phi_s - \phi_l - U_{eq} - j_{n,s}R \tag{3650}$$

where ϕ_s represents the solid (electric) potential, ϕ_l represents the potential of the liquid solution relative to a reference electrode, $j_{n,s}$ is the boundary-specific electric current, and R is the resistance that is specified at the boundary or interface.

ϕ_l is generally obtained from the Electrodynamic Potential model. For electrochemically reacting boundaries, ϕ_s is specified in the physics values at each boundary, whereas it is also determined from the Electrodynamic Potential model in the case of an interface.

The electric current is affected when additional electrical resistance is specified on a boundary or interface. The additional voltage drop is calculated according to:

$$\Delta\phi = j_{n,s}R \tag{3651}$$

where ϕ is the electric potential.

The driving force behind these reactions is the difference in Gibbs free energy $\Delta G°$ between the reactants and the products—which equates to the equilibrium potential U_{eq}. For the electrochemical reaction:

$$\left(s_a A \leftrightarrow s_b B\right) + ne^- \tag{3652}$$

$$U_{corr}(volts) = \frac{\Delta G°}{nF} \tag{3653}$$

Where the Faraday constant F=96,485,258.799 C/kmol and n is the stoichiometric coefficient of the electrons.

The potential difference U at the interface is due to the difference in potential of the metal and the electrolyte.

In STAR-CCM+, the potential difference is defined as:

$$U = \phi_s - \phi_l \tag{3654}$$

Where ϕ_s is the potential of the solid and ϕ_l is the potential of the liquid electrolyte.

The overpotential definition is equivalent to that of the overpotential which is defined for the electrodynamic potential model in Eqn. (3734). Where:

- ϕ_s is equivalent to ϕ_1

  This value represents the potential of the boundary on the solid region side of the interface which corresponds to the second boundary that you select when defining an interface.
- ϕ_l is equivalent to ϕ_0

  This value represents the potential of the boundary on the electrolyte region side of the interface which corresponds to the first boundary that you select when defining an interface.

In Eqn. (3734), ϕ_E represents the equilibrium potential U_{eq}.

If the potential difference at the interface is larger than the equilibrium potential, $U > U_{eq}$, the reaction is anodic. If the potential difference is less than the corrosion equilibrium potential, $U < U_{eq}$, the reaction is cathodic. When the equilibrium potential is zero, the anodic and cathodic reactions balance, and the net flux is zero.

Anodic reactions (that produce electrons) are related to positive overpotential. Cathodic reactions (that consume electrons) are related to negative overpotential. In these reactions, the current density i is positive for anodic reactions and negative for cathodic reactions.

The flow of charged particles determines the current (electrons in the metal and ions in the electrolyte).

The current density can be related to the electrochemical potential of a reacting metal using one of the following reaction formulation property methods:

- Butler-Volmer
- Tabular
- Tafel
- Tafel Slope (log 10)
- Transport Limited Tafel Slope (log 10)
- Nernst Equilibrium Potential

In each of these reaction formulation property methods, temperature T is essential to parameterize the rates of electrochemical reactions. In particular, temperature influences the equilibrium potential U_{eq} when it is specified using the Nernst Equilibrium Potential method. When either the gas, liquid, multicomponent gas, or multicomponent liquid model is selected, the temperature is determined by the energy models that are selected. If no energy model is selected, STAR-CCM+ uses a constant temperature of 293.15K. However, when one of the Eulerian Multiphase models is selected, the temperature is calculated as follows:

- When the Eulerian Multiphase Mixture model or Volume of Fluid (VOF) model is selected, the default energy models calculate one temperature for the combined mixture of phases.
- When the Multiphase Segregated Flow model is selected, you can select the Phase Coupled Energy model—this energy model solves for temperature in each phase. The Electrochemical Reactions model then calculates a weighted average temperature T_{echem}:

$$T_{echem} = \frac{\Sigma_i\left(\alpha_i \rho_i c_{p,\ i} T_i\right)}{\Sigma_i\left(\alpha_i \rho_i c_{p,\ i}\right)} \tag{3655}$$

  where for each phase, i:

  - α_i is the volume fraction.
  - ρ_i is the density.
  - $c_{p,\ i}$ is the heat capacity.

Butler-Volmer

When a reaction is reversible, both anodic and cathodic Tafel equations are combined to form the Butler-Volmer equation:

$$j = j_o\left\{\exp\left(\alpha_a \frac{F}{R_u T}\eta\right) - \exp\left(-\alpha_c \frac{F}{R_u T}\eta\right)\right\} \tag{3656}$$

Where α_a and α_c are respectively the anodic and cathodic charge transfer coefficients.

α_a and α_c consider the effective charge transfer coefficient—they are the intrinsic transfer coefficients that are multiplied by the number of electrons, n, that are involved in the reaction.

The exchange current density and equilibrium potential are often dependent on other variables such as the concentrations of reactants and products, temperature, and surface contaminants.

Using Butler-Volmer kinetics the specific reaction current is expressed as:

$$j_{n,s} = j_0 \prod_i \left(\frac{c_i}{c_{i,ref}} \right)^{\gamma_i} \left(e^{\frac{\alpha_a F \eta}{R_u T}} - e^{\frac{-\alpha_c F \eta}{R_u T}} \right) \tag{3657}$$

where

$$e^{\frac{\alpha_a F \eta}{R_u T}} \tag{3658}$$

accounts for anodic currents that are directed from the solid towards the liquid and

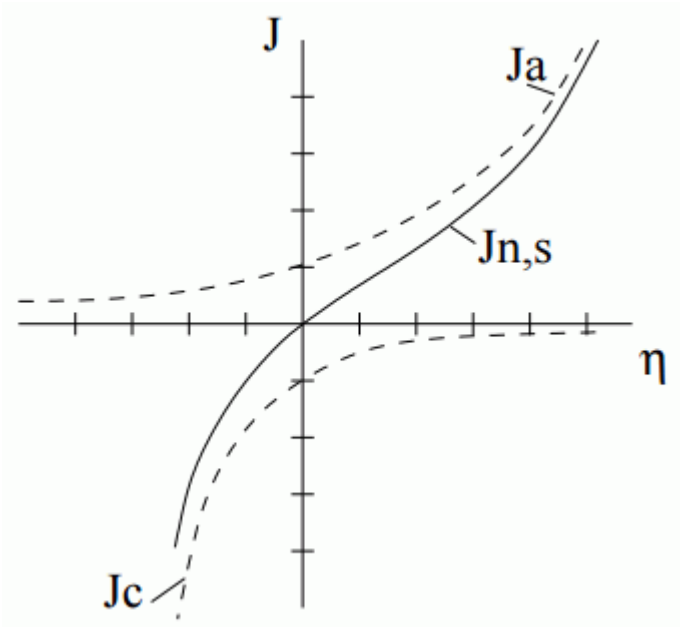
$$e^{\frac{-\alpha_c F \eta}{R_u T}} \tag{3659}$$

accounts for cathodic currents that are directed from the liquid towards the solid.

The anodic and cathodic apparent transfer coefficients, α_a, and α_c, are defined using scalar profiles.

The product term $\prod_i \left(c_i / c_{i,\,ref} \right)^{\gamma_i}$, multiplies the molar concentrations of electrochemical species that are taking part in the reaction, to the power of the rate exponent γ_i. This term is only considered when the Electrochemical Species model is selected. You can specify γ_i for each reactant and product in the reaction. $c_{i,\,ref}$ represents a reference concentration for species i. In STAR-CCM+, the reference concentration for species i is implicitly assumed to be $c_{i,\,ref}$= 1 kmol / m³.

The following diagram shows the dependence of the specific reaction current $j_{n,s}$ on the surface overpotential η for the Butler-Volmer method. Splitting the specific reaction current into anodic and cathodic contributions illustrates the nature of Tafel reactions.



Tabular

The Tabular Polarization Curve method allows you to simulate electrochemical reactions using tabular data that defines the current and electrochemical potential at the interface between the metal and the electrolyte. The Tabular method lets you import polarization curves when modeling species that do not follow the same pattern as generalized by the Butler-Volmer equation.

When modeling corrosion, STAR-CCM+ uses the polarization curves to calculate the corrosion potential for each metal when the current is zero.

When this method is selected, you set the boundary conditions at the interface between the metal and the electrolyte manually. See Tabular Polarization Curve Properties.

Tafel

Butler-Volmer kinetics accounts for both anodic reactions in which free electrons are produced in the solid phase, and cathodic reactions in which free electrons are consumed. In some scenarios, if one of the apparent transfer coefficients is small or operates solely in either positive or negative surface overpotential conditions, reactions can exhibit currents in only anodic or cathodic directions. In such cases, either j_a or j_c is neglected and the negligible reaction term is disregarded. The specific reaction current $j_{n,s}$ then reads:

$$j_{n,s} = \pm j_0 \cdot \prod_i \left(\frac{c_i}{c_{i,\,ref}} \right)^{\gamma_i} \cdot e^{\frac{\pm\alpha \cdot F\eta}{R_u T}} \tag{3660}$$

Since the apparent transfer coefficient α is no longer specified as anodic or cathodic, when using the Electrochemical Species model, the reaction setup determines the direction of the current ($\pm$).

- When electrons are specified as reactants, the current is cathodic ($-\alpha$)
- When electrons are specified as products, the current is anodic ($+\alpha$)

In the absence of the Electrochemical Species model, the direction of the current is specified in the Properties of the **Reaction Formulation** > **Tafel** node, see Reaction Formulation Properties.

Tafel Slope (log 10)

The Tafel equation is often laid out and parameterized as it is in the Butler-Volmer section, with both exponent variables independent of electric potential. In practice, often these exponent variables are measured as one combined quantity, β, and called the Tafel Slope:

$$\beta = \ln\left(10\right)\frac{R_u T}{\alpha F} = \pm 2.303\frac{R_u T}{\alpha F} \tag{3661}$$

so the specific reaction current is expressed as:

$$j_{n,s} = \pm j_0 \cdot \prod_i \left(\frac{c_i}{c_{i,\,ref}}\right)^{\gamma_i} \cdot 10^{\left(\eta\,\beta^{-1}\right)} \tag{3662}$$

When specifying the y-intercept value for the specific exchange current density, $\log_{10}\left(j_0\right)$, the specific reaction current is expressed as:

$$\log_{10}\left(\left|j_{n,s}\right|\right) = \log_{10}\left(j_0\right) + \log_{10}\left(\prod_i \left(\frac{c_i}{c_{i,\,ref}}\right)^{\gamma_i}\right) + \frac{\eta}{\beta} \tag{3663}$$

You can specify the combined term β as a scalar profile.

Since β is not specified as anodic or cathodic, the Electrochemical Species model determines the direction of the current ($\pm$). In the absence of the Electrochemical Species model, the direction of the current ($\pm$) is set in the Properties of the **Reaction Formulation** > **Tafel Slope (log 10)** node, see Reaction Formulation Properties.

Transport Limited Tafel Slope (log 10)

The Transport Limited Tafel Slope method approximates effects that are not initially accounted for in the simulation setup. When a positive limiting current $j_{\lim}$ is specified, the specific reaction current that is provided by the Tafel method (Eqn. (3660)) is limited:

$$j_{n,s} = \frac{1}{\frac{1}{j_{\lim}} + \frac{1}{j_{tafel}}} \tag{3664}$$

This method of modeling is empirical and is not based on first principles.

Nernst Equilibrium Potential

When the equilibrium potential U_{eq} is specified using the Nernst Equilibrium Potential method, U_{eq} is computed as:

$$U_{eq} = E^0 - \frac{R_u T}{nF}\ln\left[\prod_i c_i^{\nu_i}\right] \tag{3665}$$

where ν_i represents the stoichiometric coefficient of electrochemical species i in the reaction. ν_i is positive for anodic reactions / cathodic products, and negative for cathodic reactants / anodic products. E^0 is the Nernst Standard Potential at the reference molar concentration of unity. In STAR-CCM+, the reference concentration for species i is implicitly assumed to be $c_{i,\ ref}$= 1 kmol / m³. To specify alternative reference concentrations, you can define the exchange current density j_0 using the field function:

```
j_0=j_0,ref*prod(1/pow(c_i,ref,stoichiometric_coefficient))
```

where `stoichiometric_coefficient` is ν_i.

Note: c_i represents only the electrochemical species concentrations—the concentrations of the solvent are not considered.

Electrochemical Reaction Heating

You can use the Electrochemical Reaction Heating model to account for heat contributions which are due to reversible and irreversible electrochemical processes. This feature is important for applications such as electroplating or solid oxide fuel cells which create high electrochemical currents.

When using the Electrochemical Reaction Heating model, STAR-CCM+ calculates the heat contributions $\dot{q}_A$ that are released from the reacting surface interface according to:

$$\dot{q}_A = \sum_i j_{n,s,i}\ \eta_i + \ j_{n,s,i} T \frac{dU_{eq}}{dT} \tag{3666}$$

where, for the Butler-Volmer method, the irreversible heat contribution is always positive, $j_{n,\ s,\ i} \cdot \eta_i > 0\ \forall\ j_{n,\ s,\ i}, \eta_i$.

$j_{n,\ s,\ i}$ represents the boundary specific electric current density on the solid for species i, η_i is the surface overpotential for species i, and $j_{n,s,i} T \frac{dU_{eq}}{dT}$ represents the reversible heat contribution.

Note: If the Tafel method is selected and operates at negative overpotentials that are outside the valid range of approximation, it is possible that negative irreversible heat release rates are predicted as a consequence.

Lithium-Ion Model Reactions

The lithium ion battery model inherently assumes an electroneutral, binary mixture of reaction components.

With only a single scalar modelled for all chemical components, a single reaction is modelled for the lithium intercalation:

$$\mathrm{Li}^+ + e^- + \mathrm{Site} \rightleftharpoons \mathrm{Li} \tag{3667}$$

This reaction rate expression still accounts for concentration variations of different reaction components, and as such is different from the formulations in the previous section (Electrochemical Surface Reactions (Heterogeneous)).

Butler-Volmer Relationship

The Butler-Volmer relationship is used to compute the solid electrolyte interface (SEI) normal component of the electric current density $j_{n,s} = -j_{n,l}$, written as:

$$j_{n,s} = j_0 \left(e^{\frac{\alpha_a F \eta}{R_u T}} - e^{\frac{-\alpha_c F \eta}{R_u T}} \right) + C \frac{\partial}{\partial t} \left(\phi_s - \phi_l \right) \tag{3668}$$

$$j_0 = FK \left(\frac{c_s}{c_{s,\max}} \right)^{\alpha_1} \left(1 - \frac{c_s}{c_{s,\max}} \right)^{\alpha_2} \left(\frac{c_l}{c_{l,\mathrm{ref}}} \right)^{\alpha_3} \tag{3669}$$

where:

- c_l is the lithium ion/salt concentration
- c_s is the combined concentration of intercalated lithium in the solid
- $c_{l,\mathrm{ref}}$ is a molar reference concentration of lithium cations in the electrolyte

The first term in Eqn. (3669) represents the solid lithium concentration in the reaction, the last term represents the lithium ions in the electrolyte, and the intermediate term represents vacancies in the intercalation material.

The SEI surface overpotential η is computed as

$$\eta = \phi_s - \phi_l - U_{\mathrm{eq}} - j_{n,s} R_{\mathrm{SEI}} \tag{3670}$$

The rate coefficient K depends on temperature according to:

$$K = K_0 e^{\frac{-E_a}{R_u T}} \tag{3671}$$

Similarly, the electrical resistance of the SEI is dependent on temperature:

$$R_{\mathrm{SEI}} = R_0 e^{\frac{-E_{\mathrm{SEI}}}{R_u T}} \tag{3672}$$

See Also:

Porous Media

Surface Effects in Ionic Winds

The following formulation describes the boundary conditions that are present in ionic wind simulations.

Collecting Electrode

Assuming an infinitely fast neutralization of ions at the collecting electrode boundary, zero space charge density ρ is applied together with zero electric potential ϕ, as a boundary condition. The collecting electrode is assumed to be grounded.

Discharge Electrode

The visual critical electric field E_{SC}, which is set as a boundary condition, can be computed by:

$$E_{SC} = A\delta + B\sqrt{\frac{\delta}{a}} \tag{3673}$$

where:

- E_{SC} is the critical electric field intensity at the start of the corona
- A and B are empirical constants for air and a negative corona—which usually have values of $32.3 \times 10^5\ \mathrm{V/m}$ and $0.846 \times 10^5\ \mathrm{V/m^{1/2}}$ respectively
- δ is the relative density of air which is assumed to be $\delta = 1$
- a represents the local radius of the curvature of the electrode

Note: On symmetry plane boundaries, dependent variables are assumed to have a zero gradient in the direction normal to the boundary.

Dielectric Boundaries

At dielectric boundaries, such as plastic housings of ionic wind generators, where no neutralization of ions is assumed to occur, you can set a zero flux of ions by setting both the specific electric flux and the specified flux of the electrochemical species to zero.

Sorption

While electrochemical problems predominantly require the modelling of the electrochemical (charge converting heterogeneous reactions), the concentrations of reactants are also significantly changed by non charge-transferring processes such as sorption.

Sorption is a general term for the mechanisms of adsorption and absorption. Adsorption describes the process of molecules being bound to a surface by means of adhesion. Absorption refers to molecules which enter a bulk volume—in electrochemical applications, this is most commonly an electrolyte that is modelled using a solid physics continuum in STAR-CCM+.

In the same way as in the section, Electrochemical Surface Reactions (Heterogeneous), a sorption process can be annotated in the form of a reaction. For the example of water and steam/vapour at an interface, this sorption process can be written as:

$$\mathrm{H_2O^{(fluid)}} \rightleftharpoons \mathrm{H_2O^{(sorbed)}} \tag{3674}$$

The surface sorption rate r_{surf} of the process can be modelled dependent on the imbalance from equilibrium.

$$r_{surf} = r_{surf,0}\left(K_{\mathrm{eq}} c_{\mathrm{fluid}} - c_{\mathrm{sorbed}}\right) \tag{3675}$$

where $r_{surf,0}$ is the sorption rate constant and K_{eq} is an equilibrium constant, which allows for discontinuous concentrations at interfaces, like in Henry's law.

The surface rates determine the surface flux of chemical species i , which can be given as:

$$\mathrm{N}_{n,s,i} = \nu_i r_s \tag{3676}$$

Different models can be used to parameterize Eqn. (3675). STAR-CCM+ offers the following methods:

Springer

To allow for the most general representation, STAR-CCM+ models the reaction rate in terms of chemical activities rather than concentrations to account for deviations from ideal solution behavior. As a consequence, Eqn. (3675) becomes:

$$r_{surf} = r_{surf,0}\left(C_{\mathrm{eq}} - a_{sorbed}\right) \tag{3677}$$

where C_{eq} is an equilibrium-related term.

The equilibrium concentration term for the fluid species is computed as:

$$C_{\mathrm{eq}} = \xi \lambda_{\mathrm{eq}} \tag{3678}$$

where ξ is the equilibrium concentration constant of the reaction.

λ_{eq} is the dimensionless solute content at equilibrium and is usually computed in terms of chemical activities a:

$$a = \begin{cases} \left(\dfrac{c\mathrm{RT}}{P_{\mathrm{sat}}}\right)^{\gamma} & \text{for gas phase} \\ c^{\gamma} & \text{for liquid/intercalated phase} \end{cases} \tag{3679}$$

where γ is the rate exponent.

The Springer reaction formulation [162] is tailored to model water transport in PEM fuel cells. The dimensionless solute content at equilibrium λ_{eq} becomes a dimensionless water content at equilibrium, and is expressed as:

$$\lambda_{\mathrm{eq}} = \begin{cases} Aa^3 + Ba^2 + Ca + D & a \le 1 \\ \lambda_{\mathrm{eq}}\big|_{a\,=\,1}(1-s) + 16.8s & 0 < s \le 1 \end{cases} \tag{3680}$$

where s represents the liquid volume fraction. Currently, the feature that is implemented in STAR-CCM+ cannot account for liquid volume fractions; the dimensionless water content at equilibrium thus is cropped at $a = 1$, which corresponds to a saturated gas. A, B, C, and D are coefficients that you can specify. The default coefficient values in STAR-CCM+ are from Springer [162] at 30°C for water.

Wu Li Berg

Similar to the springer formulation, the Wu Li Berg reaction formulation [164] is tailored to water transport in PEM fuel cells. As a main difference, the dimensionless water content at equilibrium is formulated with λ_{eq} defined as:

$$\lambda_{\mathrm{eq}} = 0.3 + 6a(1 - \tanh(a - 0.5)) + 3.9\sqrt{a}\left(1 + \tanh\left(\frac{a - 0.89}{0.23}\right)\right) \tag{3681}$$

If the reactant activity a exceeds 1.0, the activity is limited to a value of 1.0.

Henry

Different from the above fuel cell related expressions, Henry's law is not formulated in terms of reference concentrations and dimensionless solute contents—the surface sorption rate is expressed directly as:

$$r_s = r_{s,0}\left(Ha_r - a_p\right) \tag{3682}$$

where H is Henry's constant, a_r is the activity of the reactant species, and a_p is the activity of the product species.

Bulk Ion Chemical Reactions (Homogeneous)

For certain types of electrochemistry problems, there are bulk chemical reactions that occur on a very short timescale such that they are assumed to reach equilibrium conditions instantaneously.

Instead of specifying forward and reverse rate coefficients, the effect of the reaction is expressed in an equilibrium constant that is defined as:

$$\ln K_{eq} = \ln \left[\frac{k_f}{k_r}\right] = \sum_i v_i \ln c_i^{eq} \tag{3683}$$

where k_f is the forward reaction rate, k_r is the reverse reaction rate, c_i^{eq} is the equilibrium molar concentration of a reactant or product in the reaction, and v_i is the stoichiometric coefficient of a reactant or product in the reaction (positive for a reactant and negative for a product). K_{eq} is the equilibrium constant:

$$K_{eq} = \prod_i c_i^{v_i} \tag{3684}$$

The equilibrium constant is related to the Gibbs Free Energy change for a given reaction by:

$$\Delta G = -R_u T \ln K_{eq} \tag{3685}$$

The Bulk Ion Chemical Reactions model uses the Coupled Electrochemical Species Solver to solve for electrochemical species concentrations.

For each electrochemical species, the following source term is used which pushes the concentrations of the electrochemical species towards equilibrium:

$$\frac{c_i^{eq} - c_i}{\tau} \tag{3686}$$

where τ is the characteristic timescale for equilibrium. In this feature, residence time is used as the time-step for steady problems. The residence time is computed as an average of the residence times of each electrochemical species component. You can visualize this value using the field function, Electrochemical Species Residence Time.

Model Simplifications / Engineering Approaches

While electrochemical simulations can account for the most detailed first-principle physics, most engineering problems can also be successfully diagnosed based on simplified setups which warrant both understandability and a rapid simulation time.

Current Distribution Methodology

Electrochemical engineers and scientists categorize setups according to the so-called current distribution concept of a setup. The three main aspects here lie in the consideration or neglect of:

1. electrical transport losses due to resistance in the electrolyte solution or in metal phases
2. activation overpotentials
3. concentration dependent reaction rates and overpotentials

Primary Current Distribution

Primary current distribution setups solely account for electrical transport losses due to resistance (1.), but (2.) and (3.) are neglected. These setups restrain from solving species concentrations and only need to resolve electric potentials. When temperatures are considered fixed as well, only one partial differential equation (PDE) needs to be solved for the electric potential, and setups can be simulated using only minor computational resources. Also, the computation is sped-up due to the absence of coupling effects between electric potentials and species concentrations. In STAR-CCM+, such setups are approximated using only the Electrodynamic Potential model in conjunction with the Electrochemical Reactions model—where maximum specific exchange currents are applied to minimize activation overpotentials.

Neglecting activation overpotentials can be seen equivocal to assuming an infinite electrochemical reaction rate, where all significant losses occur in the electrolyte and/or in the metal transport of charge carriers. Note a certain contradiction when treating reactions that are significantly faster than species transport—real reactions with a chemical equilibrium at high progress rates are likely to suffer from reactant depletion, decreasing their rate. Concentration effects are however not considered in primary current distribution simulations, rendering them a non-optimal setup for such simulations.

The methodology nevertheless serves well when reactions are impeded by the metal / solution transport, as in many (anti-) corrosion setups. Despite being a computationally performant modeling technique, you are recommended to use secondary current distribution with STAR-CCM+.

Secondary Current Distribution

Secondary current distribution setups build on the foundation of primary current distribution, but additionally account for activation overpotentials (2.). With the same STAR-CCM+ model set as primary current distribution setups, but with more realistic parameters set for the specific exchange currents, this treatment is equally performant when considering computational expense, but delivers more realistic results and is hence recommendable.

One main limitation of secondary current distribution setups still lies in not resolving species concentrations. As such, a secondary treatment basically assumes a perfectly mixed electrolyte. On the positive side, a certain level of saturation effects is considered in electrochemical reaction rates, usually leading to decreased, more realistic extremal values. Prefer using this treatment when electrochemical reactions do not lead to

significant changes in molar concentrations of reactants, for example when reactants are in excess (such as salt in seawater), and the timescales of the reaction are relatively large in comparison to the convective and diffusive flow timescales.

Tertiary Current Distribution

The most detailed modelling is represented by tertiary current distribution modelling where reaction rates and overpotentials are concentration dependent. In this scenario, all of (1.) (2.) and (3.) are considered.

While providing the best quality and highest fidelity of the results, this scenario requires the highest computational effort, and the highest demand on parameterizing the simulation model.

For the former computational effort, consider that primary and secondary current distributions only need to resolve the electric potential field, and reactions at the boundary—only one partial differential equation needs to be solved for the electrodynamic potential there. For isothermal tertiary current setups, one partial differential equation will need to be solved for each species; these PDEs depend on the flow velocities and as such, another three equations have to be solved for the velocity components, and one for pressure. This is all in addition to the PDE for the electric potential, pushing the computational requirements to solving at least 5+n partial differential equations rather than a single one for primary and secondary current distributions. At this point, solving for an additional PDE for temperature variations is generally recommended as it is a relatively minor cost of the overall simulation.

	Primary	Secondary	Tertiary
Considerations	Electrical Transport Losses	Electrical Transport Losses Activation Overpotential	Electrical Transport Losses Activation Overpotential Concentration Dependence
Relevant STAR-CCM+ Models	**Electrodynamic Potential Model** **Electrochemical Reactions**	**Electrodynamic Potential Model** **Electrochemical Reactions**	**Electrodynamic Potential Model** **Electrochemical Reactions** either: • **Electrochemical Species** • **Multi-Component Gas** • **Multi-Component Liquid**
Number of Partial Differential Equations Solved (isothermal)	1	1	5 + number of species
Accessible Solution Quantities	Electric Potential ϕ Electrical Currents j	Electric Potential ϕ Electrical Currents j Surface Overpotential η	Electric Potential ϕ Electrical Currents j Surface Overpotential η

	Primary	Secondary	Tertiary
			Molar Concentrations c_i Velocity v

Not only is the computational resource considerable for these simulations, the parameterization requirements for computational engineers are also considerable. Reactions need to be known and parameterized to account for their dependence on concentrations. Simple measurements like cyclic voltammetry will not suffice to parameterize the setup but detailed knowledge is required about the reaction mechanisms involved in the process, involving not only stoichiometries but also and rate constants and exchange current densities for each single reaction.

Detailed parameterization of models provides superior results when reactant concentrations significantly influence the rate of reactions—which is usually the case when the reaction rate is not infinite, but time scales are equal or smaller than the ones of diffusion / convection, that is when reactants cannot be fed fast enough to the reaction site, or products cannot be removed fast enough to sustain the reaction at unchanged rates. This is the case for many technical applications like fuel cells or batteries in critical load scenarios, or some wet-etching simulations where manufacturing speed is crucial for achieving low production costs.

Temperature Effects

Electrochemical reaction rates are clearly temperature dependent. Engineering problems may be situated at a specific temperature, however, where solving for temperature as a variable quantity might not be necessary, and the isothermal energy model could be used in order to save computational resource.

An example is represented when looking at the corrosion of a water pipeline that lies buried in permafrost ground. Temperature may be controlled by heating elements, but when investigating corrosion rates, this temperature may be considered as fixed.

Some applications may require solving for temperature. In oil processing pipelines, metals may be subjected to both spatially and temporally varying temperatures. If electrochemical reaction rates are still low, for example when considering unwanted corrosion effects, the electrochemical heating may still be neglected as it only contributes negligibly to the overall temperature.

Neglecting temperature effects has significant runtime impact when looking at primary and secondary current distribution setups, where only a single PDE needs to be solved then. Solving for temperature in tertiary setups will represent an additional PDE to the 5+n existing PDEs, as such, neglect of solving for temperature will only yield minimal savings, and is not recommended in favor of accurate results.

Electrochemistry Bibliography

[696] *Cagnoni, Davide; Agostini, Francesco; Christen, Thomas; Parolini, Nicola; Stevanović, Ivica; de Falco, Carlo. "Multiphysics simulation of corona discharge induced ionic wind". Journal of Applied Physics, Volume 114, Issue 23, pp. 233301-233301-10 (2013).*

[697] *O. R. J. G. a. J. M. V. Costa-Nunes, "Comparison of the performance of Cu–CeO 2–YSZ and Ni–YSZ composite SOFC anodes with H 2, CO, and syngas," Journal of power sources, vol. 141, no. 2, pp. 241-249, 2005.*

[698] *Douglas P. Riemer, Mark E. Orazem, Application of boundary element models to predict effectiveness of coupons for accessing cathodic protection of buried structures. Corrosion (2000) Vol. 56, No. 8, pp. 794-800.*

[699] *Doyle, C.M. 1995. "Design and Simulation of Lithium Rechargeable Batteries." PhD Thesis, Department of Chemical Engineering, University of California, Berkeley.*

[700] *Georgiadou, Maria and Richard Alkire. "Anisotropic Chemical Pattern Etching of Copper Foil." J. Electrochem Soc (1994): 679-689.*

[701] *K.L. Gering, Prediction of electrolyte viscosity for aqueous and non-aqueoussystems: Results from a molecular model based on ion solvation and a chemicalphysics framework, Electrochimica Acta 51 (2006) 3125.*

[702] *Krupa, Matthew, et al. "Cathodic Protection of Highly Complex and Shielded Components." Corrosion. 2006.*

[703] *Neimarlija N. Demirdžić I. Muzaferija S. "Finite volume method for calculation of electrostatic fields in electrostatic precipitators". Journal of Electrostatics, Volume 67, Issue 1, p. 37-47 (2009).*

[704] *Nels E. Jewell-Larsen, Sergey V. Karpov, Igor A. Krichtafovitch, Vivi Jayanty, Chih-Peng Hsu, Alexander V. Mamishev. "Modeling of corona-induced electrohydrodynamic flow with COMSOL multiphysics". Proc ESA Annual Meeting on Electrostatic 2008, Paper E1.*

[705] *Newman, J. and Thomas-Alyea, K. E. 2004. Electrochemical Systems., 3rd edn. J. Wiley. ISBN: 978-0-471-47756-3.*

[706] *Springer, T.E., Zawodzinski and Gottesfeld, S. 1991. " Polymer Electrolyte Fuel Cell Model", J. Electrochem. Soc., 138(8), pp. 2334-2342.*

[707] *Spotnitz, R. 2006. Porous Electrode Theory, ECS Short Course: Battery Design and Simulation.*

[708] *Wu, H., Li, X. and Berg, P. 2009. " On the modeling of water transport in polymer electrolyte membrane fuel cells" , Electrochimica Acta, 54(27), pp. 6913-6927.*

[709] *Zou, Y., J. Wang, and Y. Y. Zheng. "Electrochemical techniques for determining corrosion rate of rusted steel in seawater." Corrosion Science **53.1** (2011): 208-216.*

Electrochemistry Nomenclature

The following is a guide to the symbols used in this section.

Scalars

α_1	Electrode reaction order [-]
α_2	Vacancies reaction order [-]
α_3	Electrolyte reaction order [-]
α_a	Anodic transfer coefficient [-]
α_c	Cathodic transfer coefficient [-]
$\Delta G°$	Gibbs free energy of reactants and products [J/mol]
η	SEI surface overpotential [V]
η_i	Surface overpotential of species i [V]
κ	Ionic conductivity [S/m] or [A/V m]
μ_e	Chemical potential of electrolyte [J/mol]
ν_+	Number of cations into which a molecule of electrolyte dissociates [-]
$\overline{c_+}$	Uniform/Volume average Lithium/Salt concentration in positive electrode [kmol/m^3]
$\overline{c_-}$	Uniform/Volume average Lithium/Salt concentration in negative electrode [kmol/m^3]
$\overline{x_{init}}$	Volume average initial stoichiometry of negative electrode active materials [-]
$\overline{x}$	Volume average stoichiometry of negative electrode active materials [-]
$\overline{y_{init}}$	Volume average initial stoichiometry of positive electrode active materials [-]
$\overline{y}$	Volume average stoichiometry of positive electrode active materials [-]
ϕ	Electric potential [V]
ϕ_s	Electric potential of the solid [V]
ϕ_l	Electric potential of the liquid electrolyte [V]
ϕ_{corr}	Corrosion potential [V]

ϕ_s^{corr}	Corrosion potential of the solid [V]
ϕ_l^{corr}	Corrosion potential of the liquid electrolyte [V]
σ	Effective electrical conductivity [S/m] or [A/V m]
σ_0	Electrical conductivity [S/m] or [A/V m]
τ	Tortuosity, non-unity only in separator [-]
τ_{eq}	The characteristic timescale for equilibrium.
C	Double-layer capacitance [F/m^2] [C/V m^2]
c	Molar concentration of a single electrolyte [kmol/m^3]
c_0	Molar concentration of solvent [kmol/m^3]
c_i	Molar concentration of species i [kmol/m^3]
c_i^{eq}	Equilibrium molar concentration of a reactant or procuct [kmol/m^3]
c_l	Molar concentration of lithium cations in electrolyte (liquid) [kmol/m^3]
c_s	Molar concentration of lithium in electrode material (solid) [kmol/m^3]
$c_{l,\mathrm{ref}}$	Molar reference concentration of lithium in electrolyte [kmol/m^3]
$c_{s,\max}$	Molar concentration of lithium in saturated electrode material [kmol/m^3]
CAP	Cell capacity [C] [A s]
D	Effective lithium diffusivity [m^2/s]
D_0	Reference lithium diffusivity [m^2/s]
D_i	Diffusivity of species i [m^2/s]
E^0	Nernst Standard Potential at the reference molar concentration of unity [V]
E_a	Kinetic activation energy [J/kmol]
E_{SEI}	SEI activation energy [J/kmol]
F	Faraday's constant [96485339.924 C/kmol]
f	Force [N/m^3]
$f_\pm$	Mean molar activity coefficient of an electrolyte [-]
γ_i	Rate exponent of electrochemical species, i
H	Henry's constant [kmol m^{-3} Pa^{-1}]

j	Specific electric current [A/m^2]
j_0	Exchange current density [A/m^2]
j_n	SEI normal component of electric current density [A/m^2]
$j_{n,l}$	Boundary specific electric current density on the fluid [A/m^2]
$j_{n,s}$	Boundary specific electric current density on the solid [A/m^2]
$j_{n,\ s,\ i}$	Boundary specific electric current density on the solid, for species i [A/m^2]
k_f	Forward reaction rate
k_r	Reverse reaction rate
κ	Ionic conductivity [S/m]
K	Ion mobility [m^2/Vs]
K	Rate coefficient [equiv/s m^2]
K_{eq}	Equilibrium constant
K_0	Reference rate constant [equiv/s m^2]
ξ	Equilibrium concentration constant of the reaction [kg kmol^{-1}]
n	Number of electrons that are involved in electrode reaction [-]
n_+	Li+ cations in positive electrode active material [kmol]
n_-	Li+ cations in negative electrode active material [kmol]
n_{eq}	Equivalent number of electrons per mole of reactant
ν_i	Stoichiometric coefficient of electrochemical species i
N	Number flux [1/m^{2}s]
$\mathrm{N}_{n,\ s,\ i}$	Molar concentration flux of species i from the solid to the liquid [kmol/ m^2 s]
N_A	Avagadro constant 6.022136736 e^{26}[1/kmol]
n_p	Number density [1/m^3]
OCV	Open circuit voltage [V]
$OCV_{SOC=0}$	Open circuit voltage at SOC = 0
$OCV_{SOC=1}$	Open circuit voltage at SOC = 1

q	Elementary charge 1.602176487 e^{-19}[A s]
$\dot{q}_V$	Volumetric heat release [W/m^3]
$\dot{q}_A$	Per-area heat release [W/m^2]
ρ_{ion}	Ionic space charge density [A s/m^3]
r_{surf}	Surface sorption rate
R_u	Universal gas constant [8314.4621 J/kmol K]
R	Electrical resistance [Ω m^2]
R_0	Reference electrical resistance of SEI [Ω m^2]
R_{SEI}	Electrical resistance of SEI [Ω m^2]
s_+	Stoichiometric coefficient of cation in electrode reaction [-]
s_0	Stoichiometric coefficient of solvent in electrode reaction [-]
S_{c_i}	Electrochemical species source term
S_{Estat}	Electric charge density
SOC	State of charge. $SOC \in [0, 1]$ under standard operation conditions. [-]
T	Temperature [K]
t	time [s]
${t_+}^0$	Transference number of cations with respect to the velocity of the solvent species [-]
u_i	Mobility of species i [m^{2}kmol /J s]
U	Potential difference
$U_+(y)$	Relative galvanic equilibrium potential of positive active electrode material [V]
$U_-(y)$	Relative galvanic equilibrium potential of negative active electrode material [V]
U_{eq}	Relative galvanic equilibrium potential of active electrode material [V]
v	Velocity [m^2/s]
V_+	Volume of positive electrode active material [m^3]
V_-	Volume of negative electrode active material [m^3]
χ	Porosity, non-unity only in separator [-]

z_+	Charge number of cation [-]
z_i	Charge number of species i [i]
SEI	Solid electrolyte interface

Tensors or Column Vectors

$\mathbf{E}$	Electric field $-\nabla\phi$ [N/C] or [V/m]
$\mathbf{J}$	Electric current density [A/m^2]
$\mathbf{N}_+$	Lithium (cation) molar flux [kmol/m^{2}s]
$\mathbf{N}_i$	Molar flux [kmol/m^{2}s]
$\mathbf{N}_P$	Number flux [number/m^{2}s]
$\mathbf{v}$	Velocity [m/s]

Cartesian Vectors

$\mathbf{n}_l$	Normal vector pointing outwards from liquid region [-]
$\mathbf{n}_s$	Normal vector pointing outwards from solid region [-]

Electromagnetism

Many engineering applications, such as electric motors, electric switches, and transformers, involve electromagnetic phenomena. Electromagnetic phenomena can be modeled based on the classical theory of Electromagnetism, which describes the interaction between electrically charged particles in terms of electric fields, magnetic fields, and their mutual interaction.

The fundamental equations that describe electromagnetic phenomena are Maxwell's equations (Eqn. (1455)-Eqn. (1458)), which define the electromagnetic fields induced by electric charges and currents. Maxwell's equations imply the conservation of electric charge (Eqn. (1459)), which is also known as the continuity equation. Constitutive relations complete the set of equations, by specifying the relationship between the electric and magnetic fields.

Maxwell's equations cover a broad range of phenomena. Depending on the application, the equations can be reduced to a simpler form, while retaining accuracy. For example, for low-frequency electromagnetic fields in conducting media, the displacement currents can be neglected.

The general form of Maxwell's equations is presented in the section, Fundamental Equations. It is also common to express Maxwell's equations in terms of the electric potential and the magnetic potential. The potential formulation is described in this section.

Contents:

Constitutive Relations
Potential Formulation of the Governing Equations
Harmonic Time Dependence
Electrostatic Potential Model
Electrodynamic Potential Models
Magnetic Vector Potential Models
Ohmic Heating
Magnetohydrodynamics (MHD)
Electromagnetism Bibliography

Constitutive Relations

Maxwell's equations and the electric charge continuity equation do not form a closed set, as only three out of five equations are independent. Constitutive relations complete the set of equations by specifying the connection between the electric and magnetic fields.

Permittivity and Permeability

The constitutive relations between the electric field $\mathbf{E}$ and the electric flux density $\mathbf{D}$, and between the magnetic field $\mathbf{H}$ and the magnetic flux density $\mathbf{B}$ can be generalized as:

$$\mathbf{D} = \varepsilon \mathbf{E} \tag{3687}$$

$$\mathbf{B} = \mu \mathbf{H} \tag{3688}$$

where the permittivity ε describes the behavior of the material in response to an electric field, and the magnetic permeability μ describes the behavior of the material in response to a magnetic field. The inverse of the magnetic permeability is called the magnetic reluctivity, $\nu = \mu^{-1}$.

For materials other than ferroelectrics and ferromagnets, Eqn. (3687) and Eqn. (3688) are approximately linear.

In general, ε and μ are represented by second-order tensors, which define linear transformations between the vectors $\mathbf{D}$-$\mathbf{E}$ and $\mathbf{H}$-$\mathbf{B}$, respectively.

For example, for a 3D domain, Eqn. (3687) can be written in matrix form as:

$$\begin{pmatrix} D_x \\ D_y \\ D_z \end{pmatrix} = \begin{pmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz} \end{pmatrix} \begin{pmatrix} E_x \\ E_y \\ E_z \end{pmatrix} \tag{3689}$$

or, for a 2D domain on the x-y plane, as:

$$\begin{pmatrix} D_x \\ D_y \end{pmatrix} = \begin{pmatrix} \varepsilon_{xx} & \varepsilon_{xy} \\ \varepsilon_{yx} & \varepsilon_{yy} \end{pmatrix} \begin{pmatrix} E_x \\ E_y \end{pmatrix} \tag{3690}$$

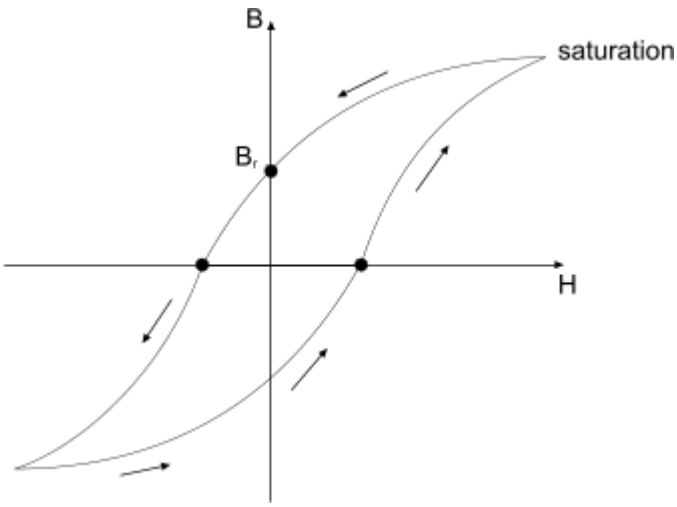
For homogeneous materials, the components of ε and μ do not depend on the position inside the material. For inhomogeneous materials, the components are functions of the position.

In the special case of isotropic materials, ε and μ reduce to diagonal matrices, with diagonal components $\varepsilon_{11} = \varepsilon_{22} = \varepsilon_{33} = \varepsilon$ and $\mu_{11} = \mu_{22} = \mu_{33} = \mu$, and can therefore be defined using scalar quantities. Specifically, ε can be defined with respect to the vacuum permittivity, $\varepsilon_0 = 8.85 \times 10^{-12}\ [\mathrm{F/m}]$, as $\varepsilon = \varepsilon_r \varepsilon_0$, where ε_r is called the relative permittivity. μ can be defined with respect to the vacuum permeability, $\mu_0 = 4\pi \times 10^{-7}\ [\mathrm{H/m}]$, as $\mu = \mu_r \mu_0$, where μ_r is called the relative permeability. For paramagnetic materials, $\mu > \mu_0$, whereas diamagnetic materials show $\mu < \mu_0$.

Ferromagnetic materials show a nonlinear response to applied electromagnetic fields. For this type of materials, Eqn. (3688) becomes a nonlinear functional equation [711]:

$$\mathbf{B} = \mu(\mathbf{B}, \mathbf{H})\mathbf{H} \tag{3691}$$

Ferromagnetic materials are often used in electric devices such as electric motors. The $\mathbf{B}$-$\mathbf{H}$ curve, or hysteresis loop, of a ferromagnetic material has the following form:



The image shows the magnetization process, where $\mathbf{B}$ increases with $\mathbf{H}$ until the material reaches saturation, and the demagnetization process, where $\mathbf{B}$ decreases with $\mathbf{H}$ until the material reaches (negative) saturation. Materials that exhibit a non-zero magnetic flux density, $\mathbf{B}_\mathrm{r}$, when the applied magnetic field is zero, $\mathbf{H} = 0$, are called permanent magnets. $\mathbf{B}_\mathrm{r}$ is called the remanent magnetic flux density.

For linear materials and soft materials which operate in the first quadrant of the B-H characteristic, the energy density stored in the magnetic field is defined as:

$$\rho_{w_m}(B) = \int_0^B \mathrm{H}'(\mathrm{B}')\mathrm{dB}' \tag{3692}$$

For linear materials, the energy density is equal to the coenergy density, which is defined as:

$$\rho_{\mathrm{w}'_m}(B) = \int_0^{H(B)} \mathrm{B}'(\mathrm{H}')\mathrm{dH}' \tag{3693}$$

Electrical Conductivity: Generalized Ohm's Law

The relationship between the electric current density $\mathbf{J}$ and the electric field $\mathbf{E}$ is described by the generalized Ohm's law:

$$\mathbf{J} = \sigma\mathbf{E} \tag{3694}$$

where the electrical conductivity σ describes the behavior of the material in response to electric currents. Like ε and μ, σ is generally defined by a second-order tensor, and reduces to a scalar in isotropic conducting materials.

For porous materials, the effective electrical conductivity can be modeled as a combination of the electrical conductivities of the fluid and solid components, using various methods:

$$\begin{aligned} \sigma_{\text{eff}} &= \chi\sigma_{\text{fluid}} + (1-\chi)\,\sigma_{\text{solid}} && \text{Volume Fraction Weighted Method} \\ \sigma_{\text{eff}} &= \sigma_{\text{solid}}(1-\chi)^{\beta} && \text{Bruggeman (solid) Method} \\ \sigma_{\text{eff}} &= \sigma_{\text{fluid}}\chi^{\beta} && \text{Bruggeman (fluid) Method} \end{aligned} \tag{3695}$$

χ is the porosity of the material, σ_{fluid} is a scalar, σ_{solid} is a second-order tensor, and β is the Bruggeman approximation exponent.

When calculating the effective electrical conductivity using the Bruggeman methods, STAR-CCM+ automatically sets the tortuosity of the porous material to:

$$\tau = \chi^{1-\beta} \tag{3696}$$

See Also:

Porous Media

Potential Formulation of the Governing Equations

To reduce the number of equations, it is convenient to reformulate the governing equations in terms of two auxiliary fields, the electric scalar potential ϕ and the magnetic vector potential $\mathbf{A}$.

Eqn. (1455) and Eqn. (1458) allow you to express the electric field $\mathbf{E}$ and the magnetic flux density $\mathbf{B}$ in terms of a scalar field ϕ and a vector field $\mathbf{A}$:

$$\mathbf{E} = -\nabla\phi - \frac{\partial \mathbf{A}}{\partial t} \tag{3697}$$

$$\mathbf{B} = \nabla \times \mathbf{A} \tag{3698}$$

where ϕ is known as the electric potential and $\mathbf{A}$ is known as the magnetic vector potential.

Writing Maxwell's equations in terms of the potentials reduces the number of independent equations, as Eqn. (1455) and Eqn. (1458) become identities. However, the potentials are not unique. To uniquely define the potential $\mathbf{A}$, it is necessary to prescribe both its curl and its divergence. The divergence of $\mathbf{A}$ is defined by an arbitrary gauge condition. For low-frequency electromagnetic fields, a common choice is the Coulomb gauge, $\nabla \cdot \mathbf{A} = 0$.

Ohm's law (Eqn. (3694)) can be written in terms of the potentials as:

$$\mathbf{J} = -\sigma\nabla\phi - \sigma\frac{\partial \mathbf{A}}{\partial t} + \mathbf{J}_{\mathrm{ex}} \tag{3699}$$

where $\mathbf{J}_{\mathrm{ex}}$ is introduced to account for any external source of electric current density. In Eqn. (3699), the direction of the electric current density is from the areas of high values to the areas of low values of the electric potential, which is known as the technical direction of the electric current density.

The electric current I flowing through a surface A is defined as:

$$I = \int_A \mathbf{J} \cdot \mathrm{d}\,\mathbf{a} \tag{3700}$$

Statics

For fields that do not vary with time, the transient terms in Maxwell's equations can be neglected, giving decoupled equations for the magnetic vector potential and the electric potential.

Electrostatics

For static fields, the transient term in Eqn. (3697) can be neglected. Therefore, the electric field only depends on the electric potential:

$$\mathbf{E} = -\nabla\phi \tag{3701}$$

The electrostatic force acting on an electric charge q is $\mathbf{F}_e = q\mathbf{E}$. The electric charge has a quantized nature and appears only in multiples of the elementary electric charge, $e = 1.602176487 \times 10^{-19}$ C.

The electric potential can be calculated from Eqn. (1457), which, together with Eqn. (3687), gives:

$$-\nabla \cdot (\varepsilon \nabla \phi) = \rho \tag{3702}$$

The STAR-CCM+ implementation of Eqn. (3702) is discussed in the section, Electrostatic Potential Model.

Magnetostatics

The magnetic vector potential can be calculated from Eqn. (1456), which, when neglecting the transient term, reduces to Ampère's law:

$$\nabla \times \frac{1}{\mu}\mathbf{B} = \mathbf{J} \tag{3703}$$

In magnetostatic applications, the transient term and the electric potential contribution in Eqn. (3699) can be neglected. By using Eqn. (3688) and Eqn. (3698), Ampère's law can then be written as:

$$\nabla \times \frac{1}{\mu}\, \nabla \times \mathbf{A} = \mathbf{J}_{\mathrm{ex}} \tag{3704}$$

The STAR-CCM+ implementation of Eqn. (3704) is discussed in the section, Magnetic Vector Potential Models.

Low-Frequency Electromagnetics in Conducting Media

For fields that vary with time, magnetic and electric fields are mutually coupled. Specifically, a time-varying magnetic field induces eddy currents which contribute to the electric field (see Eqn. (1455)). Similarly, the displacement and conduction currents associated with the electric field contribute to the magnetic field (see Eqn. (1456)). This formulation considers low-frequency electromagnetic fields in conducting materials, where the conduction currents are several orders of magnitude greater than the displacement currents, $\mathbf{J} \gg \partial \mathbf{D}/\partial t$. In many industrial applications, such as electric machines, transformers, and electric switches, the time scales of electromagnetic phenomena are relatively long, and a low-frequency assumption is appropriate.

Neglecting the displacement currents reduces Eqn. (1456) to Ampère's law (Eqn. (3703)). Also, for conducting materials, the electric current density is incompressible in nature and Eqn. (1459) simplifies to:

$$\nabla \cdot \mathbf{J} = 0 \tag{3705}$$

Inserting Eqn. (3688), Eqn. (3698), and Eqn. (3699) into Eqn. (3703) and Eqn. (3705), gives:

$$\nabla \times \frac{1}{\mu}\, \nabla \times \mathbf{A} + \sigma \frac{\partial \mathbf{A}}{\partial t} = -\sigma \nabla \phi + \mathbf{J}_{\mathrm{ex}} \tag{3706}$$

$$-\nabla \cdot \left(\sigma \nabla \phi\right) = \nabla \cdot \left(\sigma \frac{\partial \mathbf{A}}{\partial t}\right) + \nabla \cdot \mathbf{J}_{\mathrm{ex}} \tag{3707}$$

Eqn. (3706) and Eqn. (3707) describe the response of conducting media to low-frequency electromagnetic fields. These equations are a set of mutually coupled equations with two unknown fields, $\mathbf{A}$ and ϕ. In STAR-CCM+, two separate solvers compute $\mathbf{A}$ and ϕ. Based on analysis requirements, you can solve the coupled system of equations, or solve either equation singularly to compute one of the potentials in a decoupled manner. The STAR-CCM+ implementation of Eqn. (3706) and Eqn. (3707) is discussed in two separate sections, Electrodynamic Potential Models and Magnetic Vector Potential Models.

After solving for the potentials, the electromagnetic fields $\mathbf{E}$, $\mathbf{D}$, $\mathbf{H}$, and $\mathbf{B}$ can be calculated from Eqn. (3697), Eqn. (3698), and the constitutive relations.

Transverse Magnetic Modes

Eqn. (3706) accounts for 3D modes and 2D Transverse Electric (TE) modes, where the magnetic vector potential lies on the plane defined by the 2D domain. For 2D Transverse Magnetic (TM) modes, where the magnetic vector potential is normal to the 2D domain, STAR-CCM+ solves the decoupled equation:

$$-\nabla\cdot\frac{1}{\mu}\nabla A_{\mathrm{n}}+\sigma\frac{\partial A_{\mathrm{n}}}{\partial t}=J_{\mathrm{n,\ ex}} \tag{3708}$$

A_{n} and $J_{\mathrm{n,\ ex}}$ are the transverse magnetic potential and electric current density. The magnetic flux density $\mathbf{B}$ is calculated from Eqn. (3698), with the tangential components $A_{\mathrm{t}_{1,2}}$ always zero.

The implementation of Eqn. (3708) is discussed in the section, Transverse Magnetic Modes.

Harmonic Time Dependence

For fields with harmonic time dependence, it is convenient to compute the potentials using their complex phasor representation.

Consider potentials with single harmonic time dependence:

$$\begin{aligned} \mathbf{A} &= \widetilde{\mathbf{A}} \cos(\omega_0 t + \theta) \\ \phi &= \widetilde{\phi} \cos(\omega_0 t + \theta) \end{aligned} \tag{3709}$$

where $\widetilde{\mathbf{A}}$ and $\widetilde{\phi}$ are the amplitudes of the oscillations and ω_0 is the angular frequency. ω_0 is related to the frequency f_0 through:

$$\omega_0 = 2\pi f_0 \tag{3710}$$

The potentials $\mathbf{A}$ and ϕ can be written in terms of their complex phasor representations, $\widehat{\mathbf{A}}$ and $\widehat{\phi}$, as:

$$\begin{aligned} \mathbf{A} &= \mathrm{Re}\left[\widehat{\mathbf{A}}\right] = \mathrm{Re}\left[\widetilde{\mathbf{A}} e^{i\omega_0 t} e^{i\theta}\right] \\ \phi &= \mathrm{Re}\left[\widehat{\phi}\right] = \mathrm{Re}\left[\widetilde{\phi} e^{i\omega_0 t} e^{i\theta}\right] \end{aligned} \tag{3711}$$

where i is the imaginary unit, defined by $i^2 = -1$.

The time derivative of $\mathbf{A}$ can then be written as:

$$\frac{\partial \mathbf{A}}{\partial t} = \mathrm{Re}\left[i\omega \widehat{\mathbf{A}}\right] \tag{3712}$$

By replacing the potentials and the time derivatives with their complex representation, Eqn. (3706) and Eqn. (3707) become:

$$\nabla \times \frac{1}{\mu} \nabla \times \widehat{\mathbf{A}} + i\omega_0 \sigma \widehat{\mathbf{A}} = -\sigma \nabla \widehat{\phi} + \widehat{\mathbf{J}}_{\mathrm{ex}} \tag{3713}$$

$$-\nabla \cdot \left(\sigma \nabla \widehat{\phi}\right) = \nabla \cdot \left(i\omega_0 \sigma \widehat{\mathbf{A}}\right) + \widehat{S}_\phi \tag{3714}$$

where $\widehat{S}_\phi = \nabla \cdot \widehat{\mathbf{J}}_{\mathrm{ex}}$ is a user-defined source that accounts for unresolved physics (and is typically zero).

$\widehat{\mathbf{A}}$ and $\widehat{\phi}$ can be written in terms of their real and imaginary parts as:

$$\begin{aligned} \widehat{\mathbf{A}} &= \mathbf{A}' + i\mathbf{A}'' \\ \widehat{\phi} &= \phi' + i\phi'' \end{aligned} \tag{3715}$$

with $\mathbf{A}' = \mathrm{Re}\left[\widehat{\mathbf{A}}\right]$, $\mathbf{A}'' = \mathrm{Im}\left[\widehat{\mathbf{A}}\right]$, $\phi' = \mathrm{Re}\left[\widehat{\phi}\right]$, and $\phi'' = \mathrm{Im}\left[\widehat{\phi}\right]$.

Similarly, the electric current density source and the electrical conductivity can be written as:

$$\widehat{\mathbf{J}}_{\mathrm{ex}} = \mathbf{J}'_{\mathrm{ex}} + i\mathbf{J}''_{\mathrm{ex}} \tag{3716}$$

$$\sigma = \sigma' + i\sigma" \tag{3717}$$

At low-frequency, the magnetic permeability can be assumed to be a real quantity. Eqn. (3713) can then be written as:

$$\nabla\times\left[\frac{1}{\mu}\nabla\times(\mathbf{A}' + i\mathbf{A}")\right] + i\omega_0\left(\sigma' + i\sigma"\right)\left(\mathbf{A}' + i\mathbf{A}"\right) = -\left(\sigma' + i\sigma"\right)\nabla\widehat{\phi} + \left(\mathbf{J}'_{\mathrm{ex}} + i\mathbf{J}"_{\mathrm{ex}}\right) \tag{3718}$$

By using vector analysis identities and imposing the Coulomb gauge, $\nabla\cdot\widehat{\mathbf{A}} = 0$, the real and imaginary parts of Eqn. (3718) assume the simple form:

$$\begin{aligned} &-\nabla\cdot\left(\frac{1}{\mu}\nabla\mathbf{A}'\right) - \omega_0\sigma"\mathbf{A}' - \omega_0\sigma'\mathbf{A}" + \sigma'\nabla\phi' - \sigma"\nabla\phi" - \mathbf{J}'_{\mathrm{ex}} = 0 \\ &-\nabla\cdot\left(\frac{1}{\mu}\nabla\mathbf{A}"\right) + \omega_0\sigma'\mathbf{A}' - \omega_0\sigma"\mathbf{A}" + \sigma'\nabla\phi" + \sigma"\nabla\phi' - \mathbf{J}"_{\mathrm{ex}} = 0 \end{aligned} \tag{3719}$$

Similarly, the real and imaginary parts of Eqn. (3714) assume the simple form:

$$\begin{aligned} &-\nabla\cdot\left(\sigma'\nabla\phi'\right) + \nabla\cdot\left(\sigma"\nabla\phi"\right) + \omega_0\,\nabla\cdot\left(\sigma'\mathbf{A}"\right) + \omega_0\,\nabla\cdot\left(\sigma"\mathbf{A}'\right) - S'_\phi = 0 \\ &-\nabla\cdot\left(\sigma'\nabla\phi"\right) - \nabla\cdot\left(\sigma"\nabla\phi'\right) - \omega_0\,\nabla\cdot\left(\sigma'\mathbf{A}'\right) + \omega_0\,\nabla\cdot\left(\sigma"\mathbf{A}"\right) - S"_\phi = 0 \end{aligned} \tag{3720}$$

STAR-CCM+ solves the coupled system Eqn. (3719)-Eqn. (3720) for the complex potentials using the finite volume method.

It is also possible to solve for the magnetic vector potential only. In this case, STAR-CCM+ solves Eqn. (3719) neglecting the electric potential terms.

The complex magnetic flux density $\widehat{\mathbf{B}}$ and magnetic field $\widehat{\mathbf{H}}$ are calculated from:

$$\begin{aligned} \widehat{\mathbf{B}} &= \nabla\times\mathbf{A}' + i\,\nabla\times\mathbf{A}" \\ \widehat{\mathbf{H}} &= \frac{1}{\mu}(\nabla\times\mathbf{A}' + i\,\nabla\times\mathbf{A}") \end{aligned} \tag{3721}$$

The complex electric field $\widehat{\mathbf{E}}$ and the total electric current density $\widehat{\mathbf{J}}$ are:

$$\begin{aligned} \widehat{\mathbf{E}} &= -\nabla\phi' + \omega_0\mathbf{A}" - i\left(\nabla\phi" + \omega_0\mathbf{A}'\right) \\ \widehat{\mathbf{J}} &= (\sigma' + i\sigma")\widehat{\mathbf{E}} \end{aligned} \tag{3722}$$

The magnitude of the electric current density, $\left|\widehat{\mathbf{J}}\right|$, is also referred to as the peak value. The root mean square current density, $\mathbf{J}_{\mathrm{RMS}}$, which represents the time-averaged current density for sinusoidal systems, is related to $\left|\widehat{\mathbf{J}}\right|$ through $\left|\widehat{\mathbf{J}}\right| = \sqrt{2}\mathbf{J}_{\mathrm{RMS}}$.

Harmonic Transverse Magnetic Potential

Similarly, for magnetic fields lying on the plane defined by a 2D domain, Eqn. (3708) can be written as:

$$\begin{aligned} &-\nabla\cdot\left(\frac{1}{\mu}\nabla A'_{\mathrm{n}}\right) - \omega_0\sigma"A'_{\mathrm{n}} - \omega_0\sigma'A"_{\mathrm{n}} - J'_{\mathrm{n,\ ex}} = 0 \\ &-\nabla\cdot\left(\frac{1}{\mu}\nabla A"_{\mathrm{n}}\right) + \omega_0\sigma'A'_{\mathrm{n}} - \omega_0\sigma"A'_{\mathrm{n}} - J"_{\mathrm{n,\ ex}} = 0 \end{aligned} \tag{3723}$$

Electrostatic Potential Model

In electrostatic applications, STAR-CCM+ calculates the electric field induced by a distribution of electric charges from the electric potential.

Electrostatic Potential Equation

STAR-CCM+ computes the electric potential from:

$$-\oint_A \varepsilon \nabla \phi \cdot d\mathbf{a} = \int_V \rho \, dV \tag{3724}$$

which is obtained by integrating Eqn. (3702) over the cell domain. Eqn. (3724) states that the total electric flux through a closed surface A is equal to the electric charge that is contained in the volume V bounded by the surface A.

STAR-CCM+ discretizes and solves Eqn. (3724) for the electric potential using the finite volume method. The electric field and the electric flux density are calculated from the potential using Eqn. (3701) and Eqn. (3687).

Source Term

The source term for Eqn. (3724) is the electric charge density ρ. The term on the right-hand side of Eqn. (3724) represents the total electric charge contained in the volume V.

Boundary and Interface Conditions

At the domain boundaries, the solution must satisfy either Dirichlet boundary conditions, which define the electric potential ϕ, or Neumann boundary conditions, which define the component of the electric potential gradient $\nabla\phi$ normal to the boundary [711].

At a boundary Γ, Neumann boundary conditions can be defined by specifying either the:

- electric flux density $\mathbf{D}$ (see Eqn. (3687))
- specific electric flux D_n, that is, the component of $\mathbf{D}$ normal to Γ:

$$D_n = -\varepsilon \nabla \phi \cdot \mathbf{n} \tag{3725}$$

  where $\mathbf{n}$ is the boundary surface normal.
- total electric flux through Γ, that is:

$$-\int_\Gamma (\varepsilon \nabla \phi) \cdot \, \mathrm{d}\mathbf{a} \tag{3726}$$

The total and specific electric flux can also be prescribed at contact interfaces.

Electrodynamic Potential Models

In low-frequency electrodynamic applications, STAR-CCM+ calculates the electric current density induced by the non-rotational component of the electric field, from the electric potential.

Electrodynamic Potential Equations

STAR-CCM+ provides different methods for computing the electric potential. The first method is generally suitable for all electrodynamic applications, whereas the second method is specific to electric potentials with single harmonic time dependence (see Harmonic Time Dependence).

Electrodynamic Potential

STAR-CCM+ computes the electric potential from:

$$-\oint_A \sigma \nabla\phi \cdot d\mathbf{a} - \oint_A \sigma \frac{\partial \mathbf{A}}{\partial t} \cdot d\mathbf{a} = -\oint_A \sigma \boldsymbol{\rho}_\epsilon \cdot d\mathbf{a} + \int_V S_\phi \, dV \tag{3727}$$

which is obtained by integrating Eqn. (3707) over the cell domain. STAR-CCM+ discretizes and solves Eqn. (3727) using the finite volume method.

The second term on the left-hand side represents the eddy currents induced by a time-varying magnetic field. STAR-CCM+ computes this term from Eqn. (3706). When STAR-CCM+ does not solve for the magnetic vector potential, this term is not included in the equation.

Harmonic Balance FV Electrodynamic Potential

For potentials with single harmonic time dependence, STAR-CCM+ computes the complex electric potential from:

$$\begin{aligned} \oint_A (\sigma''\nabla\phi'' - \sigma'\nabla\phi') \cdot d\mathbf{a} + \omega_0 \oint_A (\sigma'\mathbf{A}'' + \sigma''\mathbf{A}') \cdot d\mathbf{a} &= \int_V S'_\phi \, dV \\ -\oint_A (\sigma'\nabla\phi'' + \sigma''\nabla\phi') \cdot d\mathbf{a} - \omega_0 \oint_A (\sigma'\mathbf{A}' - \sigma''\mathbf{A}'') \cdot d\mathbf{a} &= \int_V S''_\phi \, dV \end{aligned} \tag{3728}$$

which is obtained by integrating Eqn. (3720) over the cell domain. STAR-CCM+ discretizes and solves Eqn. (3728) using the finite volume method. Unlike Eqn. (3727), which can be solved neglecting the contribution from the electric potential, Eqn. (3728) is always solved together with the harmonic magnetic vector potential equation, Eqn. (3746).

Source Terms

The source terms on the right-hand side of Eqn. (3727) and Eqn. (3728) represent electric currents flowing through the bounding surface A.

The right-hand side of Eqn. (3727) and Eqn. (3728) is typically zero. However, to account for unresolved physics, STAR-CCM+ provides two user-defined sources, the transfer current density S_ϕ, which is a complex quantity in Eqn. (3728), and the electromotive force density $\boldsymbol{\rho}_\epsilon$, which is only available for Eqn. (3727).

In Eqn. (3707), these terms are implicitly included in the term, $\nabla \cdot \mathbf{J}_{\mathrm{ex}}$.

Additional sources of electric current density can contribute to Eqn. (3727), for example, the electric current density induced by an electrically conducting fluid in motion. These effects are discussed in specific sections.

Boundary and Interface Conditions

At the domain boundaries, the solution must satisfy either Dirichlet boundary conditions, which define the electric potential ϕ, or Neumann boundary conditions, which define the component of the electric potential gradient $\nabla\phi$ normal to the boundary. The following implementation applies to both the electric potential (Eqn. (3727)) and the complex electric potential (Eqn. (3728)). For the complex electric potential, the quantities prescribed at boundaries and interfaces are generally complex.

At a boundary face Γ, the Dirichlet b.c. is implemented as:

$$\phi_\Gamma = \phi_0 - \frac{\phi_0 - \overline{\phi}}{R^0 + R^{\mathrm{ex}}} R^0 \tag{3729}$$

where ϕ_0 is the potential value at the face neighboring cell centroid, and $\overline{\phi}$ is a user-defined potential. The electrical resistance R^0 is defined as:

$$R^0 = \frac{\mathbf{s} \cdot \mathbf{n}}{\sigma a} \tag{3730}$$

where $\mathbf{n}$ is the face area vector, $\mathbf{s}$ is the vector between the cell face and the cell centroids, and σ is the electrical conductivity of the material. R^{ex} accounts for additional sources of electrical resistance. If $R^{\mathrm{ex}} = 0$ then Eqn. (3729) reduces to a basic Dirichlet boundary condition that specifies the potential value at the boundary, $\phi_\Gamma = \overline{\phi}$.

The same condition can be prescribed at interfaces:

$$\begin{aligned} \phi_{\Gamma_0} &= \phi_0 - \frac{\phi_0 - \phi_1}{R^0 + R^{\mathrm{ex}} + R^1} R^0 = \phi_0 - J_{n_0} R^0 \\ \phi_{\Gamma_1} &= \phi_1 - \frac{\phi_1 - \phi_0}{R^0 + R^{\mathrm{ex}} + R^1} R^1 = \phi_1 - J_{n_1} R^1 \end{aligned} \tag{3731}$$

where the indices 0 and 1 identify boundary-0 and boundary-1 (that is, the two sides of the interface). The energy that is dissipated by the electric current at contact interfaces does not make any contribution to the energy equation.

For the Neumann boundary condition, the normal component of $\nabla\phi$ can be defined by specifying either the:

- electric current density $\mathbf{J}$ (see Eqn. (3694))
- total electric current through Γ, that is:

$$I_\Gamma = -\oint_\Gamma \mathbf{J} \cdot d\mathbf{a} \tag{3732}$$

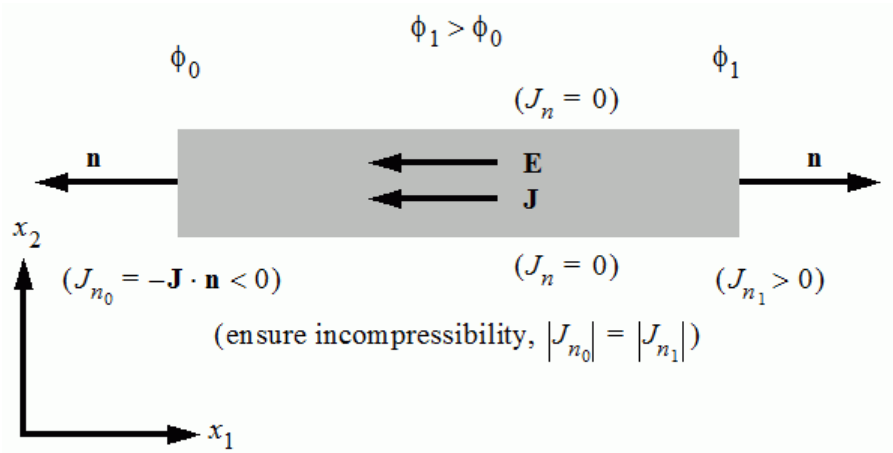
- specific electric current J_n, that is, the component of $\mathbf{J}$ normal to Γ:

$$J_n = -\mathbf{J} \cdot \mathbf{n} \tag{3733}$$

  where $\mathbf{n}$ is the boundary surface normal. A negative value of J_n means that the electric current density goes in the direction of the boundary surface normal (that is, $J_n = -\mathbf{J} \cdot \mathbf{n} < 0$).

- current-voltage characteristic, which defines the I-V curve representing the relationship between the electric current flowing through the boundary and the electric potential at the boundary. In electrochemistry applications, this relationship is often defined using the Butler-Volmer equation. For more information, see Butler-Volmer Current-Potential Characteristic.

The current-voltage characteristic, total electric current, and specific electric current can also be prescribed at interfaces. For the specific electric current, the sign convention is the same as for Neumann boundaries, except that the sign of the sources is always chosen with respect to boundary-0.



Contents:

Butler-Volmer Current-Potential Characteristic

Butler-Volmer Current-Potential Characteristic

The Butler-Volmer equation is typically used in electrochemistry applications to model the relationship between the electric current and the electric potential at an electrode, where cathodic and anodic reactions occur.

At a boundary, or interface, the over-potential η is defined as:

$$\eta = \phi_1 - \phi_0 - \phi_E - J_n R \tag{3734}$$

where R is a user-defined electrical resistance and J_n is the specific electric current (see Eqn. (3733)). For an interface, ϕ_0 and ϕ_1 denote the potential at boundary 0 and boundary 1 of the interface, respectively. For a boundary, ϕ_0 denotes the potential solution computed at the boundary, whereas ϕ_1 denotes the potential prescribed at the boundary. ϕ_E denotes the zero electric current potential.

The Butler-Volmer equation defines the relationship between the specific electric current J_n and the over-potential η, as:

$$J_n(\eta) = j_0\left(e^{\alpha_a F\eta/(R_u T)} - e^{-\alpha_c F\eta/(R_u T)}\right) \tag{3735}$$

where:

- α_a is the anodic apparent transfer coefficient
- α_c is the cathodic apparent transfer coefficient
- j_0 is the specific electric exchange current
- $F = 9.648525879 \times 10^7\ \mathrm{C\ kmol^{-1}}$ is the Faraday constant
- $R_u = 8.3144621 \times 10^3\ \mathrm{J\ kmol^{-1} K^{-1}}$ is the universal gas constant

When STAR-CCM+ solves the energy equation (Eqn. (1454)), the temperature T is automatically computed. Otherwise, STAR-CCM+ applies a default temperature of $T = 293\mathrm{K}$. Large variations of temperature between two sides of an interface should be avoided.

For an over-potential $\eta = 0$, the specific current prescribed by Eqn. (3735) is $J_n = 0$.

Magnetic Vector Potential Models

In magnetostatic or low-frequency electrodynamic applications, STAR-CCM+ calculates the magnetic field induced by electric currents from the magnetic vector potential.

Magnetic Vector Potential Equations

STAR-CCM+ solves for the magnetic vector potential using either the finite volume or the finite element method. Depending on the discretization method, STAR-CCM+ solves different integral forms of Eqn. (3706). When using a finite volume approach, STAR-CCM+ offers a choice of conservative and non-conservative formulation.

The following equations account for 3D modes and Transverse Electric modes (where the magnetic vector potential lies on the plane defined by a 2D domain). For Transverse Magnetic modes (where the magnetic vector potential is normal to the 2D domain), see Transverse Magnetic Modes.

The following equations include the electric current density induced by a spatial variation of the electric potential:

$$\mathbf{J}_{\phi} = \sigma \nabla \phi \tag{3736}$$

When STAR-CCM+ solves for the electric potential, this term is calculated from Eqn. (3727). When STAR-CCM+ does not solve for the electric potential, this term is not included in the equations.

In steady and quasi-unsteady applications, the transient term, $\partial \mathbf{A} / \partial t$, vanishes.

Finite Volume Magnetic Vector Potential: Conservative Formulation

In the conservative formulation, STAR-CCM+ does not impose a gauging condition and solves the equation:

$$\oint_A \frac{1}{\mu} \nabla \mathbf{A} \cdot d\mathbf{a} - \oint_A \frac{1}{\mu} (\nabla \mathbf{A})^{\mathrm{T}} \cdot d\mathbf{a} + \int_V \sigma \frac{\partial \mathbf{A}}{\partial t}\, dV + \int_V \sigma \nabla \phi\, dV = -\int_V \mathbf{J}_{\mathrm{ex}}\, dV \tag{3737}$$

Finite Volume Magnetic Vector Potential: Non-Conservative Formulation

In the non-conservative approach, STAR-CCM+ solves the equation:

$$\oint_A \frac{1}{\mu} \nabla \mathbf{A} \cdot d\mathbf{a} + \int_V \sigma \frac{\partial \mathbf{A}}{\partial t}\, dV + \int_V \sigma \nabla \phi\, dV = -\int_V \mathbf{J}_{\mathrm{ex}}\, dV \tag{3738}$$

STAR-CCM+ enforces the Coulomb gauge and solves an additional scalar equation for the so-called projection variable. If $\mathbf{A}$ is a solution, any potential $\mathbf{A}_0 = \mathbf{A} - \nabla \varphi$, where φ is an arbitrary scalar field called the projection variable, is also a valid solution. The projection equation is constructed by enforcing the Coulomb gauge on $\mathbf{A}_0$:

$$\nabla \cdot \nabla \varphi = \nabla \cdot \mathbf{A} \tag{3739}$$

This equation is solved in delta form.

Finite Element Magnetic Vector Potential

In steady and quasi-unsteady analyses, STAR-CCM+ solves the following regularized weak form of Eqn. (3706):

$$\int_V \frac{1}{\mu}(\nabla\times \mathbf{A})\cdot(\nabla\times \delta\mathbf{A})\,dV - \int_{\Gamma_N}\left[\mathbf{n}\times\delta\mathbf{A}\right]\cdot\left[\mathbf{n}\times\mathbf{J}_S\right]\,da + \int_V \kappa\mathbf{A}\cdot\delta\mathbf{A}\,dV = \int_V \mathbf{J}_{\mathrm{ex}}\cdot\delta\mathbf{A}\,dV \tag{3740}$$

where $\delta\mathbf{A}$ is an arbitrary test function, $\mathbf{J}_{\mathrm{ex}}$ is the source of electric current density, and $\mathbf{J}_S$ is the electric current sheet at the boundary Γ_N.

Eqn. (3740) ensures that the magnetic vector potential is uniquely defined, as it implements the gauging condition:

$$\nabla\cdot\kappa\mathbf{A} = 0 \tag{3741}$$

which states that the normal component of $\kappa\mathbf{A}$ is continuous over interfaces with different magnetic permeability.

The parameter κ is defined as:

$$\kappa = \frac{1}{\mu}\tilde{\kappa} \tag{3742}$$

where $\tilde{\kappa}$ is a user-specified regularization parameter.

In transient analyses, the regularization term:

$$\int_V \kappa\mathbf{A}\cdot\delta\mathbf{A}\,dV \tag{3743}$$

is replaced with the transient term:

$$\int_V \sigma\frac{\partial\mathbf{A}}{\partial t}\cdot\delta\mathbf{A}\,dV \tag{3744}$$

Eqn. (3740) is spatially discretized and solved using the finite element method. In each element domain, the positions are interpolated using nodal shape functions, whereas the magnetic vector potential is interpolated using lowest order H(Curl)-conforming shape functions. For more information, see H(Curl) Hierarchical Shape Functions.

For a correct representation in the lowest order finite element space, provide magnetic vector potential profiles of the form:

$$\mathbf{A} = \mathbf{a} + (\mathbf{b}\times\mathbf{x}) \tag{3745}$$

where $\mathbf{a}$ and $\mathbf{b}$ are constant vectors and $\mathbf{x}$ is the position vector.

Harmonic Balance FV Magnetic Vector Potential

For potentials with single harmonic time dependence, STAR-CCM+ computes the complex magnetic vector potential from:

$$
\begin{aligned}
-\oint_A \left(\frac{1}{\mu}\nabla \mathbf{A}'\right)\cdot d\mathbf{a} - \omega_0 \int_V (\sigma''\mathbf{A}' + \sigma'\mathbf{A}'')\, dV + \int_V (\sigma'\nabla\phi' - \sigma''\nabla\phi'')\, dV = \int_V \mathbf{J}'_{\mathrm{ex}}\, dV \\
-\oint_A \left(\frac{1}{\mu}\nabla \mathbf{A}''\right)\cdot d\mathbf{a} + \omega_0 \int_V (\sigma'\mathbf{A}' - \sigma''\mathbf{A}'')\, dV + \int_V (\sigma'\nabla\phi'' + \sigma''\nabla\phi')\, dV = \int_V \mathbf{J}''_{\mathrm{ex}}\, dV
\end{aligned}
\tag{3746}
$$

which is obtained by integrating Eqn. (3719) over the cell domain. STAR-CCM+ discretizes and solves Eqn. (3746) using the finite volume method. When STAR-CCM+ does not solve Eqn. (3728) for the electric potential, the last term on the left-hand side vanishes.

Source Terms

In general, $\mathbf{J}_{\mathrm{ex}}$ can be written as:

$$
\mathbf{J}_{\mathrm{ex}} = \mathbf{J}_{\mathrm{u}} + \mathbf{J}_{\mathrm{pm}} \tag{3747}
$$

where $\mathbf{J}_{\mathrm{u}}$ is a user-defined electric current density, and $\mathbf{J}_{\mathrm{pm}}$ is the electric current density induced by permanent magnets:

$$
\mathbf{J}_{\mathrm{pm}} = \nabla \times \frac{1}{\mu}\mathbf{B}_r \tag{3748}
$$

$\mathbf{B}_r$ is the permanent magnet remanent magnetic flux density, that is, the magnetic flux density that exists in the magnet in the absence of external magnetic fields. In Eqn. (3746), $\hat{\mathbf{J}}_{\mathrm{ex}}$ is complex (see Harmonic Time Dependence).

When modeling excitation coils or magnetohydrodynamics, STAR-CCM+ adds additional source terms to Eqn. (3747). These terms are discussed in dedicated sections.

Boundary and Interface Conditions

At the domain boundaries, the solution must satisfy either Dirichlet or Neumann boundary conditions. Dirichlet boundary conditions define the magnetic vector potential $\mathbf{A}$. Neumann boundary conditions define the electric current sheet $\mathbf{J}_S$, that is, the magnetic field tangential to the boundary:

$$
\mathbf{J}_S = \mathbf{H} \times \mathbf{n} \tag{3749}
$$

In Eqn. (3746), the magnetic vector potential and the electric current sheet are complex quantities.

The Dirichlet boundary condition implementation depends on the discretization method. In the finite volume implementation, a Dirichlet boundary condition prescribes the magnetic vector potential $\mathbf{A}$ at the boundary. In the finite element implementation, a Dirichlet boundary condition prescribes the components of $\mathbf{A}$ that are tangential to the boundary.

Typical boundary conditions are:

- **Symmetry (Perfect Magnetic Conductor)**:

  Forces the magnetic flux to cross the boundary at an angle of 90°.

  Specifically, it sets $\mathbf{J}_S$ to zero. In the finite volume implementation, the normal component of $\mathbf{A}$ is also set to zero, whereas the components tangential to the boundary are left free.

- **Anti-Symmetry (Perfect Electric Conductor)**:

  Prevents any magnetic flux from crossing the boundary.

  Specifically, it sets the tangential component of $\mathbf{A}$ to zero, while leaving the normal component free.

In the finite element framework, it is also possible to prescribe the electric current sheet $\mathbf{J}_S$ at interfaces:

$$\mathbf{J}_S = \mathbf{n} \times \left(\mathbf{H}_1 - \mathbf{H}_0\right) \tag{3750}$$

where $\mathbf{H}_1$ and $\mathbf{H}_0$ denote the magnetic field at the two sides of the interface, and $\mathbf{n}$ is the surface normal pointing from side 0 to side 1.

Eqn. (3749) follows from Eqn. (3750), where the magnetic field at the domain boundary is $\mathbf{H}_0$, and outside the simulation domain $\mathbf{H}_1 = 0$.

For electromagnetic coils, the electric sheet boundary (or interface) condition can be interpreted as:

$$\mathbf{J}_S = \frac{n_\mathrm{t} I}{L} \tag{3751}$$

where n_t is the number of turns of the coil, I is the electric current that flows through an individual coil turn, and L is the coil length.

Contents:

Transverse Magnetic Modes
Excitation Coils
Power Losses
Electromagnetic Force

Transverse Magnetic Modes

For magnetic fields that lie on a 2D domain, the magnetic vector potential has only one component perpendicular to the 2D domain. This configuration is known as Transverse Magnetic (TM).

Transverse Magnetic Potential Equations

STAR-CCM+ calculate the transverse magnetic potential using the finite volume approach. When the transverse magnetic potential has harmonic time dependence, STAR-CCM+ solves for the complex phasor representation of the transverse magnetic potential.

Transverse Magnetic Potential

For a magnetic field that lies on the x-y plane, STAR-CCM+ calculates the transverse magnetic potential A_z from:

$$-\int_A \frac{1}{\mu} \nabla A_z \cdot d\mathbf{a} + \int_V \sigma \frac{\partial A_z}{\partial t}\, dV = \int_V J_{z,\mathrm{ex}}\, dV \tag{3752}$$

which is obtained by integrating Eqn. (3708) over the cell domain. Eqn. (3752) is spatially discretized and solved using the finite volume method.

In steady and quasi-unsteady applications, the transient term, $\partial \mathbf{A} / \partial t$, vanishes.

Harmonic Balance FV Transverse Magnetic Potential

For single harmonic time dependence, STAR-CCM+ calculates the complex transverse magnetic potential $\widehat{A}_z$ from:

$$\begin{aligned} -\oint_A \left(\frac{1}{\mu}\nabla A'_z\right) \cdot d\mathbf{a} - \omega_0 \int_V \left(\sigma'' A'_z + \sigma' A''_z\right) dV &= \int_V J'_{z,\ \mathrm{ex}}\, dV \\ -\oint_A \left(\frac{1}{\mu}\nabla A''_z\right) \cdot d\mathbf{a} + \omega_0 \int_V \left(\sigma' A'_z - \sigma'' A'_z\right) dV &= \int_V J''_{z,\ \mathrm{ex}}\, dV \end{aligned} \tag{3753}$$

which is obtained by integrating Eqn. (3722) over the cell domain. Eqn. (3753) is discretized and solved using the finite volume method.

Current Conservation

In transient simulations, you can choose whether STAR-CCM+ calculates the transverse magnetic potential while imposing the conservation of electric current within the region. When conserving the electric current, STAR-CCM+ scales the total electric current density in Eqn. (3752), or Eqn. (3753), such that:

$$\int_A J_z\, da = 0 \tag{3754}$$

where the integral is over the surface that bounds the whole region. The total electric current density in Eqn. (3753) is generally complex, $\widehat{J}_z$.

Source Terms

For transverse magnetic modes, the electric current density $\mathbf{J}_{\mathrm{ex}}$ (see Eqn. (3747)) has only one component, $J_{z,\mathrm{ex}}$, which is normal to the 2D domain. For single harmonic time dependence (Eqn. (3753)), $\widehat{J}_{z,\mathrm{ex}}$ is a complex quantity.

Boundary and Interface Conditions

At the domain boundaries, the solution must satisfy either Dirichlet boundary conditions, which define the magnetic vector potential A_z normal to the boundary, or Neumann boundary conditions, which define the electric current sheet $J_{S,\ z}$ normal to the boundary. In the harmonic case (Eqn. (3753)), the prescribed quantities are complex.

In many applications, such as electrical machines, the cross-sectional field analysis can be reduced to either an odd number or an even number of poles by employing anti-periodic or periodic interface conditions, respectively. The magnetic potential has opposite sign at each side of an anti-periodic interface:

$$A_z(r,\theta) = -A_z\left(r, \theta + \frac{2(k-1)\pi}{p}\right) \quad k = 1,\ 2,\ 3\ldots \tag{3755}$$

and the same sign at each side of a periodic interface:

$$A_z(r,\theta) = A_z\left(r, \theta + \frac{2k\pi}{p}\right) \quad k = 1,\ 2,\ 3... \tag{3756}$$

where p is the even or odd number of pole pairs and r, θ are polar coordinates.

Excitation Coils

When modeling magnetic fields in solid materials, STAR-CCM+ allows you to include the effects of excitation coils in the magnetic field calculation.

In STAR-CCM+, a coil is regarded as a volumetric domain, for example, a thick wire or a closely wound bundle locally oriented according to a specified vector, $\mathbf{f}$. The cross section of the coil is calculated as:

$$a_0 = \frac{1}{n_s}\int_{a_{\mathrm{ref}}} \mathbf{f} \cdot d\mathbf{a} \tag{3757}$$

where a_{ref} is a specified reference area, which can be the actual cross section of the coil, or the area that is obtained by cutting the coil with an arbitrary section. As an arbitrary section can cut the coil region in multiple locations, STAR-CCM+ divides the reference area by the number of sections that are cut through the coil, n_s. When the reference area is equal to the actual cross section of the coil, $n_s = 1$.

$\mathbf{f}$ is a user-defined local direction field. For $\mathbf{f}$ constant on a_{ref} and $|\mathbf{f}| = 1$, a_0 is the projection of a_{ref} onto a plane section of the coil with surface normal $\mathbf{f}$. For coils with non-constant cross section (representing, for instance, wire bundles that are not bound with uniform tightness), $\mathbf{f}$ must satisfy:

$$\int_a \mathbf{f} \cdot d\mathbf{a} = \int_A \mathbf{f} \cdot d\mathbf{a} \tag{3758}$$

for any section a of the streamtube generated along $\mathbf{f}$, starting from A.

Note: STAR-CCM+ does not enforce any requirements on $\mathbf{f}$.

Electric currents flowing through coils of conducting wire induce magnetic fields. The electric current density associated with an excitation coil of cross section area a_0 is:

$$\mathbf{J}_{\mathrm{coil}} = \frac{In_t}{a_0}\mathbf{f} \tag{3759}$$

where I is the electric current that flows through an individual coil turn, n_t is the number of turns of the coil, and $\mathbf{f}$ is the specified local direction field. $\mathbf{f}/a_0$ represents the electric current density corresponding to an electric current of 1A.

To account for the magnetic field induced by the coil, STAR-CCM+ adds $\mathbf{J}_{\mathrm{coil}}$ as a source term in the magnetic vector potential equation (that is, $\mathbf{J}_{\mathrm{coil}}$ contributes to $\mathbf{J}_{\mathrm{ex}}$ in Eqn. (3706)).

The magnetic flux linkage associated with a coil is defined as the line integral of the magnetic vector potential $\mathbf{A}$ around the contour of the coil:

$$\psi = \int_A \mathbf{B} \cdot d\mathbf{a} = \int_L \mathbf{A} \cdot d\mathbf{l} \tag{3760}$$

By writing the line element $d\mathbf{l}$ as:

$$d\mathbf{l} = \mathbf{f}\frac{n_\mathrm{t}}{a_0}dV \tag{3761}$$

Eqn. (3760) can be written in terms of a_0 and n_t as:

$$\psi = \frac{n_\mathrm{t}}{a_0}\int_V \mathbf{A} \cdot \mathbf{f}dV = \int_V \rho_\psi \, dV \tag{3762}$$

where ρ_ψ is called the specific magnetic flux linkage:

$$\rho_\psi = \frac{n_\mathrm{t}}{a_0}\mathbf{A} \cdot \mathbf{f} \tag{3763}$$

The time derivative of the specific magnetic flux linkage defines the specific electromotive force induced by the coil:

$$\rho_\varepsilon = \frac{d\rho_\psi}{dt} \tag{3764}$$

The electrical resistance of the conducting coil can be calculated as:

$$R = \frac{L}{\sigma a_\mathrm{coil}} \tag{3765}$$

where σ is the electrical conductivity, $L = L_0 n_\mathrm{t}$ is the length of the coil, $a_\mathrm{coil} = a_0 p_\mathrm{c}/n_\mathrm{t}$, and p_c is the percentage of a_0 that is covered by conducting material.

Power Losses

When modeling energy converters, such as electric motors and electric generators, it is important to estimate the energy losses associated with time-varying magnetic fields.

Electric machines usually have a ferromagnetic core, where the power losses are generally due to:

- Magnetic hysteresis: in ferromagnetic materials, magnetization is a nonlinear process which dissipates energy (see B-H Curve).
- Eddy currents: time-varying magnetic fields induce electric currents which dissipate energy.

These losses can be estimated using the Steinmetz equation, which calculates the hysteresis and eddy current losses associated with a sinusoidal magnetic flux density of known frequency:

$$W_i = \underbrace{C_h f B^n{}_{pk}}_{\text{hysteresis}} + \underbrace{\frac{C_e}{2\pi^2}\overline{\left[\frac{dB}{dt}\right]^2}}_{\text{eddy currents}} \tag{3766}$$

where the first and second terms on the right-hand side account for losses by hysteresis, and losses by eddy currents, respectively, and:

- i denotes a cell, or element, of the discretized domain
- $n = a + bB_{pk}$, where a and b are called the Steinmetz coefficients and B_{pk} is the peak magnetic flux density
- C_h and C_e are the hysteresis loss coefficient and the eddy-current loss coefficient, respectively
- f is the frequency of excitation

Considering a rotating electric machine, the peak magnetic flux density is:

$$B_{pk}\Big|_i = \frac{1}{2}\left(\max(|B|_i^k) - \min(|B|_i^k)\right) \tag{3767}$$

where $k = 1, ..., N$ denotes the rotor position, with N being the total number of rotor positions.

The average time derivative of magnetic flux density over one cycle of the excitation frequency is:

$$\overline{\left[\frac{\mathrm{d}\,B}{\mathrm{d}\,t}\right]^2}\Bigg|_i = \frac{1}{N}\sum_{k=1}^{N}\left(\frac{|B|_i^k - |B|_i^{k-1}}{\delta t}\right) \tag{3768}$$

where δt denotes the time-step.

Electromagnetic Force

At a boundary or interface, STAR-CCM+ calculates the electromagnetic surface force by evaluating the Maxwell stress tensor.

For a linear material, the Maxwell stress tensor is defined as:

$$\boldsymbol{\sigma}_{\mathrm{EM}}(\mathbf{B}) = \mu^{-1}\mathbf{B} \otimes \mathbf{B} - \left(\frac{1}{2}\mu^{-1}\mathbf{B} \cdot \mathbf{B}\right)\mathbf{I} \tag{3769}$$

and for a nonlinear material:

$$\boldsymbol{\sigma}_{\mathrm{EM}}(\mathbf{B}) = \mathbf{H} \otimes \mathbf{B} - \left(\mathbf{H} \cdot \mathbf{B} - \int_0^{|\mathbf{B}|} \mu^{-1}(B')\, dB'\right)\mathbf{I} \tag{3770}$$

At a boundary, STAR-CCM+ calculates the electromagnetic surface force density as:

$$\mathbf{F}_{\mathrm{EM}} = \boldsymbol{\sigma}_{\mathrm{EM}} \cdot \mathbf{n} \tag{3771}$$

where $\boldsymbol{\sigma}_{\mathrm{EM}}$ is the Maxwell stress tensor and $\mathbf{n}$ is the surface normal.

At an interface, the electromagnetic surface force density is calculated as:

$$\mathbf{F} = \left(\boldsymbol{\sigma}_{\mathrm{EM}_1} - \boldsymbol{\sigma}_{\mathrm{EM}_0}\right) \cdot \mathbf{n}_{01} \tag{3772}$$

where $\boldsymbol{\sigma}_{\mathrm{EM}_1}$ and $\boldsymbol{\sigma}_{\mathrm{EM}_0}$ represent the Maxwell stress tensor at each side of the interface, and $\mathbf{n}_{01}$ is the surface normal pointing from side 0 to side 1.

The finite element implementation also allows for calculation of the electromagnetic force at mesh element nodes. STAR-CCM+ computes the nodal forces as:

$$\mathbf{F}_M = \int_{M \in V} \boldsymbol{\sigma}_{\mathrm{EM}} \cdot \nabla \mathcal{N}_M \, dV \tag{3773}$$

where M identifies the element nodes, V is the set of elements containing the node M, and $\mathcal{N}_M$ represents H^1 Lagrangian shape functions [714].

Ohmic Heating

Electric currents flowing in conducting materials generate heat. STAR-CCM+ accounts for this effect by adding a source term to the energy equation.

The heat source per unit volume due to an electric current density $\mathbf{J}$ is given by Joule's Law:

$$Q = \mathbf{J} \cdot \mathbf{E} \tag{3774}$$

where $\mathbf{E}$ is the electric field. For sinusoidal fields (see Harmonic Time dependence), STAR-CCM+ calculates a cycle-averaged heat source per unit volume, as:

$$Q = \frac{1}{2}\mathrm{Re}\left(\widehat{\mathbf{J}} \cdot \widehat{\mathbf{E}}^*\right) \tag{3775}$$

where $\widehat{\mathbf{E}}^*$ is the complex conjugate of $\widehat{\mathbf{E}}$ (see Eqn. (3722)).

When modeling Ohmic heating, the heat source due to the electric current density contributes as a source term to the energy equation (Eqn. (1454)).

The total power that is dissipated from a volume of conducting material is:

$$P = \int_V \mathbf{J} \cdot \mathbf{E} \ \mathrm{d}V \tag{3776}$$

On wall boundaries and contact interfaces with an electric resistance (see Eqn. (3729) and Eqn. (3731)), the heat source per unit area:

$$P = J_n^2 R^{\mathrm{User}} \tag{3777}$$

contributes to the face neighboring cell center source term of the energy equation. On contact interfaces, the contribution to the cell center source term of the energy equation is weighted by the effective thermal conductivities of the neighboring physics continua.

Magnetohydrodynamics (MHD)

STAR-CCM+ allows you to model the interaction between electrically conducting fluids, such as molten metals, electrolytes, and plasmas, and electromagnetic fields.

A conducting fluid in relative motion to a magnetic field induces an electric current density:

$$\mathbf{J}_{\mathrm{L}} = \sigma \mathbf{v} \times \mathbf{B} \tag{3778}$$

where σ is the electrical conductivity (see Eqn. (3694)), $\mathbf{v}$ is the flow velocity, and $\mathbf{B}$ is the magnetic flux density. Eqn. (3778) contributes to the total electric current density $\mathbf{J}$ (see Eqn. (3707)).

$\mathbf{J}_{\mathrm{L}}$ in turn induces a magnetic flux density which contributes to the total magnetic flux density in Eqn. (3778). STAR-CCM+ calculates $\mathbf{J}_{\mathrm{L}}$ using either a one-way coupled or a two-way coupled approach. In the one-way coupled approach, STAR-CCM+ calculates $\mathbf{J}_{\mathrm{L}}$ with respect to a prescribed magnetic flux density, $\mathbf{B}_{\mathrm{p}}$. In the two-way coupled approach, STAR-CCM+ calculates the total magnetic flux density $\mathbf{B}$, which also accounts for the magnetic flux density induced by $\mathbf{J}_{\mathrm{L}}$.

One-Way Coupled MHD

The ratio between the induced and the prescribed magnetic flux densities is defined by a non-dimensional number known as the magnetic Reynolds number:

$$Re_{\mathrm{mag}} = \mu_0 \sigma U L \tag{3779}$$

where μ_0 is the vacuum permeability, U is the characteristic flow velocity and L is the characteristic length.

The one-way coupled MHD approach is suitable for small Re_{mag}, where the induced magnetic flux density is significantly weaker than the prescribed magnetic flux density.

In this case, STAR-CCM+ solves Eqn. (3707) in a decoupled manner, neglecting the eddy currents induced by time-varying magnetic fields (see Low-Frequency Electromagnetics in Conducting Media). In absence of eddy currents, adding $\mathbf{J}_{\mathrm{L}}$ (Eqn. (3778)) to the total electric current density $\mathbf{J}$ (Eqn. (3699)) gives:

$$\mathbf{J} = -\sigma \nabla \phi + \sigma \mathbf{v} \times \mathbf{B} \tag{3780}$$

where $\mathbf{B}$ is the prescribed magnetic flux density:

$$\mathbf{B} = \mathbf{B}_{\mathrm{p}} \tag{3781}$$

Two-Way Coupled MHD

With this approach, STAR-CCM+ solves Eqn. (3706) (and, when modeling the electric potential, Eqn. (3707)). In this case, the total electric current density is:

$$\mathbf{J} = -\sigma \nabla \phi - \sigma \frac{\partial \mathbf{A}}{\partial t} + \sigma \mathbf{v} \times \mathbf{B} \tag{3782}$$

with:

$$\mathbf{B} = \nabla \times \mathbf{A} \tag{3783}$$

In quasi-unsteady analyses, the eddy current term in Eqn. (3782) is neglected. When STAR-CCM+ does not solve for the electric potential, the term $\mathbf{J}_\phi = -\sigma\nabla\phi$ in Eqn. (3782) is also neglected.

For simplicity, Eqn. (3780) and Eqn. (3782) assume that the only contribution to $\mathbf{J}_{\mathrm{ex}}$ in Eqn. (3699) is $\mathbf{J}_{\mathrm{L}}$. Additional contributions to $\mathbf{J}_{\mathrm{ex}}$ are discussed in the relevant sections.

In circuit breaker simulations (plasmas), the contribution to the electric current density that comes from the gradient of the electric potential, $\mathbf{J}_\phi = -\sigma\nabla\phi$, typically dominates over the other contributions. In electromagnetic stirring applications, the dominant contribution typically comes from the eddy currents, $\mathbf{J}_{\mathbf{A}} = -\sigma\partial\mathbf{A}/\partial t$. When the magnetic flux density and the fluid velocity are significant, $\mathbf{J}_{\mathrm{L}}$ becomes the dominant term. This is typically the case in turbulence control applications, which use the Lorentz force to modulate or suppress the flow turbulence.

Lorentz Force

The conducting fluid experiences a body force per unit volume known as the Lorentz force:

$$\mathbf{f}_{\mathrm{L}} = \mathbf{J} \times \mathbf{B} \tag{3784}$$

where $\mathbf{J}$ is the total electric current density, as defined in Eqn. (3780) and Eqn. (3782). In MHD applications, STAR-CCM+ includes the Lorentz force in the momentum equation (Eqn. (1451)).

For sinusoidal fields (see Harmonic Time dependence), STAR-CCM+ calculates a cycle-averaged Lorentz force, as:

$$\mathbf{f}_{\mathrm{L}} = \frac{1}{2}\mathrm{Re}\left(\hat{\mathbf{J}} \times \widehat{\mathbf{B}}^*\right) \tag{3785}$$

where $\hat{\mathbf{J}}$ is defined in Eqn. (3722) and $\widehat{\mathbf{B}}^*$ is the complex conjugate of $\widehat{\mathbf{B}}$ (see Eqn. (3721)). In harmonic balance applications, Eqn. (3778) does not contribute to the total electric current density.

Electromagnetism Bibliography

[710] *Ida, Nathan. 2003. Engineering Electromagnetics, 2nd ed., Springer Publishing Company, Inc.*

[711] *Jackson, John David. 1999. Classical Electrodynamics, 3rd ed., Wiley, New York, NY.*

[712] *Feynman, Richard P. and Leighton, Robert 2010. The Feynman Lectures on Physics, Volume II, Basic Books, New Millennium ed.*

[713] *Streitberger, Hans Joachim and Doessel, Karl-Friedrich, Eds., 2008. Automotive Paints and Coatings, 2nd ed., Wiley-VCH.*

[714] *Henrotte F. and Felden M., Electromagnetic force computation with the Eggshell method, Institute of Electrical Machines, 2010.*

Solid Mechanics

Solid Mechanics describes the behavior of a solid continuum in response to applied loads. Applied loads include body forces, surface loads, point forces, or thermal loads that result from changes in the solid temperature. Applied loads induce a stress field in the structure and can cause displacement of the structure —from an initial undeformed configuration to a deformed configuration.

The Solid Mechanics formulation uses the following conventions:

- The coordinates in the initial (undeformed) configuration are denoted by capital letters, whereas the coordinates in the current (deformed) configuration are denoted by lower case letters. The same convention applies to the indices identifying the components of any vector in the undeformed and deformed configurations. For example:
  - $\mathbf{X}$ is the position vector in the initial configuration, $\mathbf{x}$ is the position vector in the current configuration.
  - X_I, with $I = 1, 2, 3$, denotes the components of $\mathbf{X}$ in the initial configuration. The formulation also uses X, Y, Z for X_1, X_2, X_3.
  - σ_{ij} are the components of the stress tensor in the current configuration.
- The indices i and j (either upper or lower case) are reserved for coordinates, whereas the indices M and N identify element nodes.
- When using the index notation (see Index and Matrix Notations), this formulation uses the Einstein convention, which implies summation over a set of repeated indices.
- The terms *vertex* and *node* are used interchangeably.

Contents:

Displacement, Stress, Strain
Governing Equations
Constitutive Laws
Finite Element Discretization
Rayleigh Damping
Static and Dynamic Solutions
Bibliography

Displacement, Stress, Strain

The key quantities that describe the behavior of a structure under loading are stress, displacement, and strain.

Stress

The concept of stress is related to the force per unit area acting on a surface. In a solid structure, external forces or temperature changes induce a stress field, which can lead to motion and deformation of the solid structure. Internal forces that oppose the deformation of the structure also induce an internal elastic stress that tends to restore the structure to the original undeformed state. Some materials also present built-in stress, that exists in absence of applied forces and deformations.

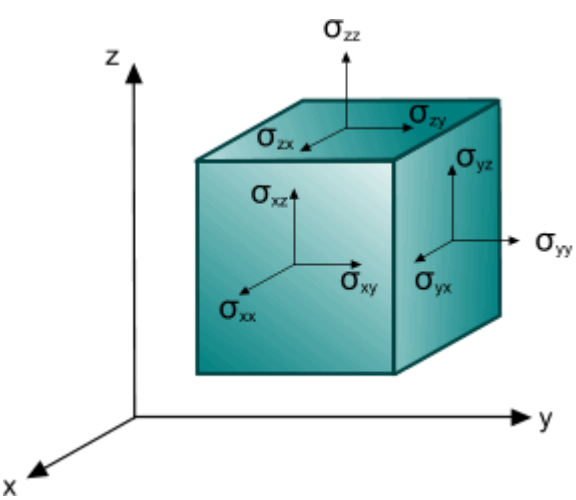
In general, the stress acting on a plane section of a body is defined by a vector $\boldsymbol{\tau}$, which is known as stress vector, or traction, as:

$$\boldsymbol{\tau} = \begin{pmatrix} \tau_x \\ \tau_y \\ \tau_z \end{pmatrix} = \begin{pmatrix} F_x/A \\ F_y/A \\ F_z/A \end{pmatrix} \tag{3786}$$

where $\mathbf{F}$ is the force acting on the plane and A is the area of the plane.

The stress at a point is the force per unit area, as the area over which the force is applied approaches zero. At a point, there is an infinite number of plane sections of the body. However, the state of stress at the point is completely defined by the stress vectors that are associated with three mutually perpendicular planes passing through the point. Therefore, the state of stress at any point, on any plane, is defined by a second-order tensor of the following form:

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{pmatrix} \tag{3787}$$



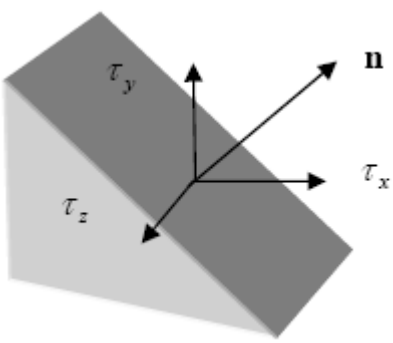
Since the sum of moments must go to zero as the body volume shrinks to zero, the stress tensor is symmetric.

The off-diagonal terms are called shear stresses, since they act tangentially to a face. The diagonal terms are referred to as normal stresses, since they act normal to a face.

The stress vector on a plane can be written as the matrix product:

$$\boldsymbol{\tau} = \boldsymbol{\sigma} \cdot \mathbf{n} = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{pmatrix} \begin{pmatrix} \mathrm{n}_x \\ \mathrm{n}_y \\ \mathrm{n}_z \end{pmatrix} \tag{3788}$$

where n is the unit vector normal to the plane.



Simple states of stress include:

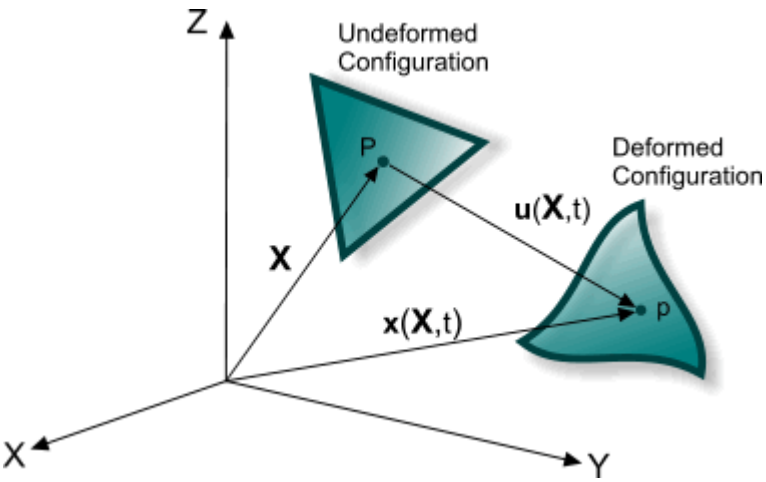
Isotropic Stress—The stress across a material is the same in all directions. If the stress is compressive, this state is also called **Hydrostatic Stress**.

Plane Stress—Thin materials with no loading on the planar surfaces are in a state of plane stress, that is, the stress in the direction normal to the planar surfaces is zero (for example, if the planar surfaces lie on the x-y plane, the stress in the z-direction is zero, $\sigma_{zz} = \sigma_{xz} = \sigma_{yz} = 0$).

Uniaxial Stress—All but one of the stress components are zero, $\sigma_{xx} \neq 0, \quad \sigma_{yy} = \sigma_{zz} = \sigma_{xy} = \sigma_{yz} = \sigma_{zx} = 0.$

Displacement

Applied loads can result in a displacement of the solid structure from an initial configuration to a deformed configuration. The total displacement is a combination of rigid body motion and the relative displacements of the points in the body, which determine the deformation of the solid structure.



If the position of a material point in the undeformed configuration is $\mathbf{X}$, and the displacement of this point to the deformed configuration is $\mathbf{u}(\mathbf{X}, t)$, the position of the material point in the deformed configuration is:

$$\mathbf{x}(\mathbf{X}, t) = \mathbf{X} + \mathbf{u}(\mathbf{X}, t) \tag{3789}$$

In component form, the displacement can be expressed as:

$$\mathbf{u} = \{u_x \ u_y \ u_z\}^{\mathrm{T}} \tag{3790}$$

The displacement field of a rigid body is completely defined by a single displacement vector. The displacement field of a deformable body is defined by the set of displacement vectors of its material points.

Deformation Gradient

The deformation gradient tensor $\mathbf{F}$ measures how the deformation changes from point to point:

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \mathbf{I} + \frac{\partial \mathbf{u}}{\partial \mathbf{X}} = \mathbf{I} + \begin{pmatrix} \frac{\partial u_x}{\partial X} & \frac{\partial u_x}{\partial Y} & \frac{\partial u_x}{\partial Z} \\ \frac{\partial u_y}{\partial X} & \frac{\partial u_y}{\partial Y} & \frac{\partial u_y}{\partial Z} \\ \frac{\partial u_z}{\partial X} & \frac{\partial u_z}{\partial Y} & \frac{\partial u_z}{\partial Z} \end{pmatrix} \tag{3791}$$

where $\mathbf{I}$ is the identity matrix and X, Y, and Z are the Cartesian components of the material point position vector $\mathbf{X}$.

Strain

Strain is a measure of the deformation of a body in terms of the relative displacement of its material points.

Consider two material points in a body that deforms from an initial configuration to some deformed configuration. In 1D, the strain $\boldsymbol{\varepsilon}$ can be defined in terms of the distance between the points in either the initial or the current configuration, as a scalar:

$$\varepsilon_{\mathrm{G}} = \frac{\mathrm{d}u}{\mathrm{d}X} \tag{3792}$$

or

$$\varepsilon_{\mathrm{A}} = \frac{\mathrm{d}u}{\mathrm{d}x} \tag{3793}$$

where ε_{G} is known as Green strain, ε_{A} is known as Almansi strain, $\mathrm{d}X$ and $\mathrm{d}x$ are the distances between the points in the initial and current configuration, respectively, and $\mathrm{d}u = \mathrm{d}x - \mathrm{d}X$ is the displacement.

In 3D, the state of strain at any point in a body is fully described by a second-order symmetric tensor:

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{xy} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{xz} & \varepsilon_{yz} & \varepsilon_{zz} \end{bmatrix} \tag{3794}$$

The diagonal terms are called normal or extensional strains and the off-diagonal terms are called shear strains.

A material is in a state of **Plane Strain** when $\varepsilon_{zz} = \varepsilon_{xz} = \varepsilon_{yz} = 0$.

Contents:

Stress and Strain Definitions

Stress and Strain Definitions

Stress and strain can be defined in various ways. For example, you can define them in either the undeformed configuration, or in the deformed configuration. In the infinitesimal strain assumption, all tensor definitions for stress and strain are equivalent.

Stress Definitions

Cauchy Stress

The stress tensor in Eqn. (3787) is defined in the deformed configuration and is more formally called Cauchy stress, or true stress. Cauchy stress is a direct measure of the traction acting on any surface of the body, in the current configuration. For output, STAR-CCM+ uses the Cauchy stress.

Second Piola-Kirchhoff Stress

The second Piola-Kirchhoff stress tensor $\mathbf{S}$ is defined as:

$$\mathbf{S} = \det\left(\mathbf{F}\right)\mathbf{F}^{-1}\boldsymbol{\sigma}\,\mathbf{F}^{-\mathrm{T}} \tag{3795}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress and $\mathbf{F}$ is the deformation gradient tensor in Eqn. (3791). In component form:

$$S_{IJ} = \det\left(F_{iI}\right)F_{iI}^{-1}\sigma_{ij}\,F_{jJ}^{-\mathrm{T}} \tag{3796}$$

Principal Stresses

At any point, there are always three planes, called principal planes, upon which the shear stress (that is, the tangential component of the stress vector) is zero. The stress vectors associated with these planes are perpendicular to the planes and are called the principal stresses $\{\sigma_1 \;\; \sigma_2 \;\; \sigma_3\}$, where σ_1 is the maximum

principal stress, σ_3 is the minimum principal stress, and σ_2 is the intermediate principal stress, with $\sigma_1 \geq \sigma_2 \geq \sigma_3$. The principal stresses correspond to the eigenvalues of the stress tensor. If the material is in a hydrostatic state of stress, all principal stresses are equal.

Solid mechanics uses the convention where stresses are positive when in tension and negative in compression, so the maximum principal is the most tensile. For this reason, it is often used as a measure of tensile failure in a material.

Maximum Shear Stress

On one plane, the shear stress is a maximum. The maximum shear stress is an important measure of ductile failure associated with plastic behavior. It is defined as:

$$\boldsymbol{\tau}_{\max} = \frac{(\boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_3)}{2} \tag{3797}$$

Mean Stress

The mean stress is defined as:

$$\sigma_{\mathrm{m}} = \frac{(\sigma_1 + \sigma_2 + \sigma_3)}{3} = \frac{(\sigma_{xx} + \sigma_{yy} + \sigma_{zz})}{3} = \frac{1}{3}\mathrm{tr}(\boldsymbol{\sigma}) \tag{3798}$$

If the material is in a hydrostatic state of stress, the mean stress is interpreted as the negative of the pressure.

Deviatoric Stress

The deviatoric stress is useful for describing plastic yielding and ductile failure. It is defined as:

$$\mathbf{s} = \boldsymbol{\sigma} - \sigma_{\mathrm{m}}\mathbf{I} \tag{3799}$$

von Mises Stress

The von Mises stress is an important measure of ductile failure. It represents the magnitude of the deviatoric stress and is independent of the mean stress:

$$\sigma_{\mathrm{vm}} = \sqrt{\frac{1}{2}\left((\sigma_1 - \sigma_2)^2 + (\sigma_1 - \sigma_3)^2 + (\sigma_2 - \sigma_3)^2\right)} \tag{3800}$$

If the material is in a state of uniaxial stress, the von Mises stress is equal to the nonzero component of the stress. Often, material yielding is studied by loading coupons which are approximately in a uniaxial state of stress.

Strain Definitions

STAR-CCM+ allows you to model linear geometry applications, where both displacements and strains are small, and nonlinear geometry applications with large rotations but small strains. In linear geometry applications, the strain can be described using the infinitesimal strain approximation, whereas nonlinear geometry applications require a finite strain (nonlinear) approximation to describe the state of strain.

The infinitesimal strain assumption is often used in structural engineering to describe the elastic behavior of materials such as steel or concrete, for which the deformations are usually small. The large-displacement, small-strain assumption is useful to describe the deformation of thin structures, that are often subject to large rotations with relatively small strains.

Green-Lagrange Finite Strain

The Green-Lagrange strain tensor defines the strain in the undeformed configuration, as:

$$\mathbf{E} = \frac{1}{2}\left(\mathbf{F}^{\mathrm{T}}\mathbf{F} - \mathbf{I}\right) \tag{3801}$$

where $\mathbf{F}$ is the deformation gradient, defined by Eqn. (3791), and $\mathbf{I}$ is the 3 x 3 identity matrix.

In Einstein notation, the components of the Green-Lagrange strain tensor are:

$$E_{IJ} = \frac{1}{2}\left(\frac{\partial u_I}{\partial X_J} + \frac{\partial u_J}{\partial X_I} + \frac{\partial u_K}{\partial X_I}\frac{\partial u_K}{\partial X_J}\right) \tag{3802}$$

Euler-Almansi Finite Strain

The Euler-Almansi finite strain tensor defines the strain in the deformed configuration, as:

$$\mathbf{e} = \frac{1}{2}\left(\mathbf{I} - \mathbf{F}^{-\mathrm{T}}\mathbf{F}^{-1}\right) \tag{3803}$$

where $\mathbf{F}$ is the deformation gradient, and $\mathbf{I}$ is the identity matrix.

In Einstein notation, the components of the Euler-Almansi strain tensor are:

$$e_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \frac{\partial u_k}{\partial x_i}\frac{\partial u_k}{\partial x_j}\right) \tag{3804}$$

Infinitesimal Strain

For small deformations, the Green-Lagrange and Euler-Almansi definitions are almost identical. The infinitesimal strain is defined as:

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left(\frac{\partial \mathbf{u}}{\partial \mathbf{X}} + \left[\frac{\partial \mathbf{u}}{\partial \mathbf{X}}\right]^{\mathrm{T}}\right) \approx \mathbf{E} \approx \mathbf{e} \tag{3805}$$

The infinitesimal strain is also called linear strain, since the strain depends linearly on the displacement.

Volumetric Strain

The volumetric strain is the change of volume relative to the undeformed volume. In the infinitesimal strain approximation, the volumetric strain is:

$$\frac{\Delta V}{V} = \varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz} \tag{3806}$$

Thermal Strain

Thermal strain is a measure of the deformation of a body due to changes in the body temperature.

If the solid is in an unconstrained state, a change in the temperature does not induce internal stresses and the material is free to expand due to a temperature increase, or to shrink due to a temperature decrease. If a solid body is constrained, a change in the temperature induces thermal stresses in the body.

STAR-CCM+ only supports linear elastic materials, for which the thermal strain is a linear function of temperature and impacts only the volume change in the material (see Constitutive Laws).

Energy Conjugate Stress-Strain Pairs

The variation of the strain energy per unit volume, due to stress, can be expressed in either the initial or current configuration, by using the correct pairing of stress and strain definitions:

$$\delta W = \delta \mathbf{e}:\boldsymbol{\sigma} = \delta \mathbf{E}:\mathbf{S} \tag{3807}$$

In the current configuration, $\delta\mathbf{e}$ is just the symmetric part of the gradient of the virtual displacement. The antisymmetric part can be dropped since the stress $\boldsymbol{\sigma}$ is symmetric:

$$\delta\mathbf{e} = \frac{1}{2}\left[\frac{\partial\delta\mathbf{u}}{\partial\mathbf{x}} + \left(\frac{\partial\delta\mathbf{u}}{\partial\mathbf{x}}\right)^{\mathrm{T}}\right] \tag{3808}$$

$\boldsymbol{\sigma}$ is the Cauchy stress, $\delta\mathbf{E}$ is the variation of the Green-Lagrange strain defined in Eqn. (3802), and $\mathbf{S}$ is the 2nd Piola-Kirchhoff stress tensor defined in Eqn. (3796). $\boldsymbol{\sigma}$ and $\delta\mathbf{e}$ are called a conjugate stress-strain pair, and so are called $\mathbf{S}$ and $\delta\mathbf{E}$.

The variation of the deformation gradient with respect to the undeformed coordinates is:

$$\delta\mathbf{F} = \frac{\partial\delta\mathbf{u}}{\partial\mathbf{X}} \tag{3809}$$

which gives the variation of the Green-Lagrange strain as:

$$\delta\mathbf{E} = \frac{1}{2}\left(\delta\mathbf{F}^{\mathrm{T}}\mathbf{F} + \mathbf{F}^{\mathrm{T}}\delta\mathbf{F}\right) = \frac{1}{2}\left[\left(\frac{\partial\delta\mathbf{u}}{\partial\mathbf{X}}\right)^{\mathrm{T}}\mathbf{F} + \mathbf{F}^{\mathrm{T}}\frac{\partial\delta\mathbf{u}}{\partial\mathbf{X}}\right] \tag{3810}$$

Governing Equations

Solid Mechanics studies the displacement of a solid continuum under prescribed loads and constraints. The fundamental laws that govern the mechanics of solids are the same laws that describe the mechanics of fluids, namely, the conservation of mass, linear momentum, angular momentum, and energy.

The conservation laws, Eqn. (1450), Eqn. (1451), and Eqn. (1454) are written in Eulerian form. In Solid Mechanics, it is more natural to express the conservation laws using a Lagrangian approach, where the observer follows the solid material as it moves through space and time.

Conservation of Mass

In the Lagrangian approach, mass is always conserved. The mass that is contained in any deformed volume is the same mass that was originally contained in the undeformed volume:

$$M = \int_V \rho(t) \ \mathrm{d}V = \int_{V_0} \rho_0 \ \mathrm{d}V = \mathrm{const} \tag{3811}$$

Since the mass within the volume is conserved, volume changes result in density changes. In fact, this leads to a slightly different interpretation of the material density specified in STAR-CCM+. The specified density is the material density in the undeformed configuration, at a reference temperature. If M_0 is the mass contained in a volume V_0, with density ρ_0, the density in the deformed configuration is:

$$\rho(V,T) = \frac{M_0}{V(T)} = \frac{\int_{V_0} \rho_0 \ \mathrm{d}V}{\int_{V(T)} \mathrm{d}V} \tag{3812}$$

where T is the current temperature.

For example, in the linear isotropic elastic assumption (see Eqn. (3821)), the infinitesimal strain is the change of volume relative to the deformed volume:

$$\frac{V - V_0}{V} = \varepsilon_v = \alpha\left(T - T_{\mathrm{ref}}\right) + \frac{\sigma_{\mathrm{m}}}{K} \tag{3813}$$

where K is the bulk modulus and the density is:

$$\rho\left(\sigma_{\mathrm{m}}, T\right) = \rho_0\left(1 - \alpha\left(T - T_{\mathrm{ref}}\right) - \frac{\sigma_{\mathrm{m}}}{K}\right) \tag{3814}$$

where T_{ref} is the reference temperature.

Conservation of Momentum: Equation of Motion

The motion of a solid body is governed by Cauchy's equilibrium equation, which expresses the conservation of linear momentum for a continuum. In the Lagrangian approach, the convective term in Eqn. (1451) vanishes and the time derivative of the velocity reduces to the partial second derivative of the displacement:

$$\rho\ddot{\mathbf{u}} - \nabla\cdot\boldsymbol{\sigma} - \mathbf{b} = 0 \tag{3815}$$

where:

- $\mathbf{u}$ is the displacement of the solid body
- $\mathbf{b}$ is the total body force per unit volume
- the Cauchy stress tensor $\boldsymbol{\sigma}$ is symmetric (see Eqn. (1453))

Eqn. (3815) is subject to Dirichlet and Neumann boundary conditions, which in Solid Mechanics are referred to as constraints and loads:

$$\begin{aligned} &\mathbf{u} = \overline{\mathbf{u}} && \text{Dirichlet b.c. (constraint)} \\ &\boldsymbol{\tau} = \boldsymbol{\sigma}\cdot\mathbf{n} = \overline{\boldsymbol{\tau}} && \text{Neumann b.c. (load)} \end{aligned} \tag{3816}$$

where $\overline{\mathbf{u}}$ is a prescribed displacement on a surface Γ_u, and $\overline{\boldsymbol{\tau}}$ is a prescribed traction vector on a surface Γ_τ, with surface normal $\mathbf{n}$. In STAR-CCM+, you can also apply constraints and forces at points.

Principle of Virtual Work

An alternative formulation of the equilibrium problem is the principle of virtual work, which is particularly suitable for discretization by the Finite Element method. To derive the principle of virtual work, Eqn. (3815) is multiplied by a test function $\delta\mathbf{u}$ and integrated over the space domain:

$$\delta\Pi = \int_V \delta\mathbf{u}\cdot(\rho\ddot{\mathbf{u}} - \nabla\cdot\boldsymbol{\sigma} - \mathbf{b})\,\mathrm{d}V = 0 \tag{3817}$$

This formulation uses as test function the virtual displacement, $\delta\mathbf{u}$, which must satisfy the Dirichlet boundary conditions. After integrating by parts and introducing Neumann boundary conditions, Eqn. (3817) reduces to:

$$\delta\Pi = \int_V \delta\mathbf{u}\cdot\rho\ddot{\mathbf{u}}\;\mathrm{d}V - \int_V \delta\mathbf{u}\cdot\mathbf{b}\;\mathrm{d}V + \int_V \delta\mathbf{e}:\boldsymbol{\sigma}\;\mathrm{d}V - \int_{\Gamma_\tau} \delta\mathbf{u}\cdot\overline{\boldsymbol{\tau}}\;\mathrm{d}\Gamma = \mathbf{0} \tag{3818}$$

where $\delta\mathbf{e}:\boldsymbol{\sigma}$ is the variation of the strain energy per unit volume induced by stress. Eqn. (3818) assumes the following boundary conditions:

- Dirichlet boundary condition (that is, prescribed displacement) $\delta\mathbf{u} = 0$ at the surface Γ_u.
- Neumann boundary condition (that is, prescribed surface traction) $\boldsymbol{\tau} = \overline{\boldsymbol{\tau}}$ at the surface Γ_τ. For simplicity, Eqn. (3818) only includes surface boundary conditions. Neumann boundary conditions at points are included in the discretized equation, Eqn. (3838).
- The remaining surfaces $\Gamma_r = \Gamma - \Gamma_u - \Gamma_\tau$ are free from loads and constraints.

The differential equation, Eqn. (3815), is called a strong form. The principle of virtual work, Eqn. (3818), is called a weak form, as the equations are satisfied in an integral, or weak, sense.

Eqn. (3818) is written in the current configuration. Most Solid Mechanics formulations use a *total* Lagrangian approach, where all quantities are specified in the initial configuration. In the infinitesimal strain approximation, there is no distinction between the initial and current configuration.

The stress-induced variation of the strain energy, $\delta\mathbf{e}:\boldsymbol{\sigma}$, can be expressed in the initial configuration as $\delta\mathbf{E}:\mathbf{S}$, where $\delta\mathbf{E}$ is the variation of the Green-Lagrange strain, and $\mathbf{S}$ is the 2nd Piola-Kirchhoff stress (see Energy Conjugate Stress-Strain Pairs).

In the initial configuration, Eqn. (3818) becomes:

$$\delta \Pi = \int_{V_0} \delta \mathbf{u} \cdot \rho_0 \ddot{\mathbf{u}} \ \mathrm{d}V - \int_{V_0} \delta \mathbf{u} \cdot \mathbf{b} \ \mathrm{d}V + \int_{V_0} \delta \mathbf{E} : \mathbf{S} \ \mathrm{d}V - \int_{\Gamma_\tau} \delta \mathbf{u} \cdot \overline{\boldsymbol{\tau}} \ \mathrm{d}\Gamma = \mathbf{0} \tag{3819}$$

where:

- ρ_0 and V_0 are the initial density and the initial undeformed volume
- the surface integral is left in the current configuration, for convenience

Eqn. (3819) satisfies conservation of linear and angular momentum. The Lagrangian approach also ensures that mass is conserved.

Constitutive Laws

The strain field is directly related to the gradient of the displacement field, whereas the stress field is related to the displacements through the strain field. Constitutive laws complete the formulation by describing the relationship between the stress tensor and the strain tensor.

Stress and strain do not have a unique definition (see Stress and Strain). In the infinitesimal strain assumption, where both displacements and strains are small, all definitions of stress and strain are equivalent. For small strains, but arbitrarily large displacements and rotations, the stress and strain tensors that appear in the following constitutive laws must be an energy-conjugate stress-strain pair (see Energy Conjugate Stress-Strain Pairs).

Linear Elastic Materials

Linear elastic materials extend proportionally to the applied load and return to the original configuration when the load is removed. This assumption, which is valid for small strains, assumes a linear stress-strain relationship of the form:

$$\boldsymbol{\sigma} = \mathbf{D}\left(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_T\right) \tag{3820}$$

where:

- $\mathbf{D}$ is called the material tangent.
- $\boldsymbol{\varepsilon}_T = \boldsymbol{\alpha}\Delta T$ is the thermal strain (see Thermal Strain), with $\Delta T = T_L - T_{\mathrm{ref}}$ being the difference between the temperature load T_L and the reference temperature T_{ref} at which the thermal strain (and stress) is assumed to be zero.
- $\boldsymbol{\alpha}$ is the vector of thermal expansion coefficients α.

Anisotropic Materials

In the most general case, materials have different material properties in different directions. Materials with this characteristic are called anisotropic.

It is convenient to express Eqn. (3820) in Voigt notation. Voigt notation is a convenient notation for symmetric tensors, as it reduces the order of the tensor by specifying only its independent components. For example, second-order symmetric tensors can be represented as 6-dimensional vectors. In Voigt notation, the tensors in Eqn. (3820) can be written as:

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{23} \\ \sigma_{13} \end{pmatrix}; \ \mathbf{D} = \begin{pmatrix} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} & c_{16} \\ & c_{22} & c_{23} & c_{24} & c_{25} & c_{26} \\ & & c_{33} & c_{34} & c_{35} & c_{36} \\ & & & c_{44} & c_{45} & c_{46} \\ & & & & c_{55} & c_{56} \\ & & & & & c_{66} \end{pmatrix}; \ \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{12} \\ 2\varepsilon_{23} \\ 2\varepsilon_{13} \end{pmatrix}; \ \boldsymbol{\alpha} = \begin{pmatrix} \alpha_{11} \\ \alpha_{22} \\ \alpha_{33} \\ 2\,\alpha_{12} \\ 2\,\alpha_{23} \\ 2\,\alpha_{13} \end{pmatrix} \tag{3821}$$

The factor 2 in the shear components of $\boldsymbol{\varepsilon}$ ensures that the variation of strain energy due to stress (Eqn. (3807)) is the same in tensor notation and Voigt notation. The strain shear components $\gamma_{xy} = 2\varepsilon_{xy}$,

$\gamma_{yz} = 2\varepsilon_{yz}$, and $\gamma_{xz} = 2\varepsilon_{xz}$ are also called engineering shear strain. The material tangent $\mathbf{D}$ is symmetric, with 21 independent parameters c_{ij}. As $\mathbf{D}$ is symmetric (that is, $c_{ij} = c_{ji}$), the bottom half of the matrix has been omitted.

Orthotropic Materials

Materials that have independent properties on three mutually-orthogonal directions are called orthotropic. Specifically, the material properties are constant along each axis, but independent from the values along the other axes. For orthotropic materials, the material tangent $\mathbf{D}$ reduces to :

$$\mathbf{D} = \begin{pmatrix} c_{11} & c_{12} & c_{13} & 0 & 0 & 0 \\ & c_{22} & c_{23} & 0 & 0 & 0 \\ & & c_{33} & 0 & 0 & 0 \\ & & & G_{12} & 0 & 0 \\ & & & & G_{23} & 0 \\ & & & & & G_{13} \end{pmatrix} \tag{3822}$$

The material tangent coefficients for orthotropic materials can therefore be completely defined in terms of nine independent constants:

$$\begin{array}{ll} E_1,\ E_2,\ E_3 & \text{Young's Moduli} \\ \nu_{12}, \nu_{23},\ \nu_{13} & \text{Poisson Ratios} \\ G_{12},\ G_{23},\ G_{13} & \text{Shear Moduli} \end{array} \tag{3823}$$

as:

$$\begin{aligned} c_{11} &= \frac{1 - \nu_{23}\,\nu_{32}}{E_2 E_3\,\Delta} \\ c_{22} &= \frac{1 - \nu_{31}\,\nu_{13}}{E_3 E_1\,\Delta} \\ c_{33} &= \frac{1 - \nu_{12}\,\nu_{21}}{E_1 E_2\,\Delta} \\ c_{12} = c_{21} &= \frac{1}{E_2 E_3\,\Delta}(\nu_{21} + \nu_{31}\,\nu_{23}) \\ c_{13} = c_{31} &= \frac{1}{E_2 E_3\,\Delta}(\nu_{31} + \nu_{21}\,\nu_{32}) \\ c_{23} = c_{32} &= \frac{1}{E_1 E_3\,\Delta}(\nu_{32} + \nu_{31}\,\nu_{12}) \end{aligned} \tag{3824}$$

where:

$$\nu_{21} = \nu_{12}\,\frac{E_2}{E_1};\ \nu_{32} = \nu_{23}\,\frac{E_3}{E_2};\ \nu_{31} = \nu_{13}\,\frac{E_3}{E_1} \tag{3825}$$

and:

$$\Delta = \frac{1 - 2\,\nu_{13}\,\nu_{21}\,\nu_{32} - \nu_{13}\,\nu_{31} - \nu_{23}\,\nu_{32} - \nu_{12}\,\nu_{21}}{E_1\,E_2\,E_3} \tag{3826}$$

For orthotropic materials, the vector of thermal expansion coefficients $\boldsymbol{\alpha}$ reduces to:

$$\boldsymbol{\alpha} = \begin{pmatrix} \alpha_{11} \\ \alpha_{22} \\ \alpha_{33} \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{3827}$$

Isotropic Materials

Materials that have identical properties in all directions are called isotropic. For isotropic materials, the material tangent $\mathbf{D}$ reduces to:

$$\mathbf{D} = \begin{pmatrix} c_{11} & c_{12} & c_{12} & 0 & 0 & 0 \\ & c_{11} & c_{12} & 0 & 0 & 0 \\ & & c_{11} & 0 & 0 & 0 \\ & & & G & 0 & 0 \\ & & & & G & 0 \\ & & & & & G \end{pmatrix} \tag{3828}$$

with:

$$\begin{aligned} c_{11} &= \frac{4}{3}G + K \\ c_{12} &= -\frac{2}{3}G + K \end{aligned} \tag{3829}$$

That is, the material tangent coefficients for isotropic materials can be completely defined in terms of two independent constants, G and K, called the shear modulus and the bulk modulus. These constants are related to the Young's modulus, E, and Poisson's ratio, ν, through:

$$\begin{aligned} G &= \frac{E}{2(1+\nu)} \\ K &= \frac{E}{3(1-2\nu)} \end{aligned} \tag{3830}$$

Materials are considered incompressible when $G/K \rightarrow 0$, that is, $\nu \rightarrow 0.5$. The current formulation is valid for materials with $\nu \leq 0.45$.

For isotropic materials, the vector of thermal expansion coefficients $\boldsymbol{\alpha}$ reduces to:

$$\boldsymbol{\alpha} = \alpha \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{3831}$$

Finite Element Discretization

STAR-CCM+ calculates the displacement of a solid based on the principle of virtual work, which is discretized using the Finite Element method.

This approach follows the total Lagrangian displacement Finite Element formulation by Zienkiewicz and Taylor [716]. For general information on the Finite Element method, see Finite Element Method.

The continuous space domain is discretized into a finite number of elements, which are interconnected at the vertices. In each element, the nodal positions and displacements are interpolated with nodal shape functions $\mathcal{N}_M$:

$$\begin{aligned} \mathbf{x} &= \mathbf{X} + \mathbf{u} \\ \mathbf{X} &= \mathcal{N}_M \mathbf{X}_M \\ \mathbf{u} &= \mathcal{N}_M \mathbf{u}_M \end{aligned} \tag{3832}$$

where $\mathbf{x}$ and $\mathbf{X}$ denotes the position vectors in the current and initial configuration, respectively. $\mathbf{X}_M$ and $\mathbf{u}_M$ are the position and displacement at the node M, and $\mathcal{N}_M$ is a node-oriented Lagrange shape function (see H1 Lagrange Shape Functions).

The discretized form of the variation of the Green-Lagrange strain is:

$$\delta\mathbf{E} = \widehat{\mathbf{B}}_M \delta\mathbf{u}_M \tag{3833}$$

where $\widehat{\mathbf{B}}_M$ is the strain-displacement matrix, which is derived by inspecting Eqn. (3810):

$$\widehat{\mathbf{B}}_M = \begin{bmatrix} F_{11}\mathcal{N}_{M,1} & F_{21}\mathcal{N}_{M,1} & F_{31}\mathcal{N}_{M,1} \\ F_{12}\mathcal{N}_{M,2} & F_{22}\mathcal{N}_{M,2} & F_{32}\mathcal{N}_{M,2} \\ F_{13}\mathcal{N}_{M,3} & F_{23}\mathcal{N}_{M,3} & F_{33}\mathcal{N}_{M,3} \\ F_{11}\mathcal{N}_{M,2} + F_{12}\mathcal{N}_{M,1} & F_{21}\mathcal{N}_{M,2} + F_{22}\mathcal{N}_{M,1} & F_{31}\mathcal{N}_{M,2} + F_{32}\mathcal{N}_{M,1} \\ F_{12}\mathcal{N}_{M,3} + F_{13}\mathcal{N}_{M,2} & F_{22}\mathcal{N}_{M,3} + F_{23}\mathcal{N}_{M,2} & F_{32}\mathcal{N}_{M,3} + F_{33}\mathcal{N}_{M,2} \\ F_{13}\mathcal{N}_{M,1} + F_{11}\mathcal{N}_{M,3} & F_{23}\mathcal{N}_{M,1} + F_{21}\mathcal{N}_{M,3} & F_{33}\mathcal{N}_{M,1} + F_{31}\mathcal{N}_{M,3} \end{bmatrix} \tag{3834}$$

The deformation gradient in discretized form is:

$$F_{iJ} = \delta_{iJ} + u_{Mi}\mathcal{N}_{M,J}(\mathbf{X}) \tag{3835}$$

where $\mathcal{N}_{M,J}$ is a short hand notation for the gradients of the shape functions with respect to the initial coordinates:

$$\mathcal{N}_{M,J} \equiv \frac{\partial \mathcal{N}_M(\mathbf{X})}{\partial X_J} \tag{3836}$$

In the infinitesimal strain approximation, the strain-displacement matrix $\widehat{\mathbf{B}}_M$ reduces to:

$$\widehat{\mathbf{B}}_M = \begin{bmatrix} \mathcal{N}_{M,1} & 0 & 0 \\ 0 & \mathcal{N}_{M,2} & 0 \\ 0 & 0 & \mathcal{N}_{M,3} \\ \mathcal{N}_{M,2} & \mathcal{N}_{M,1} & 0 \\ 0 & \mathcal{N}_{M,3} & \mathcal{N}_{M,2} \\ \mathcal{N}_{M,3} & 0 & \mathcal{N}_{M,1} \end{bmatrix} \tag{3837}$$

Substituting the discretized displacements (Eqn. (3832)) and the discretized form of $\delta\mathbf{E}$ into the weak form (Eqn. (3819)), gives:

$$\delta\Pi = \delta\mathbf{u}_M^{\mathrm{T}}\left[\int_{V_0} \widehat{\mathbf{B}}_M^{\mathrm{T}}\mathbf{S}\ \mathrm{d}V + \int_{V_0} \mathcal{N}_M\mathbf{I}\mathcal{N}_N\rho_0\,\ddot{\mathbf{u}}_N\ \mathrm{d}V - \int_{V_0} \mathcal{N}_M\mathbf{b}\ \mathrm{d}V - \int_{\Gamma} \mathcal{N}_M\overline{\boldsymbol{\tau}}\ \mathrm{d}\Gamma - \mathbf{f}_M^{\mathrm{p}}\right] = \mathbf{0} \tag{3838}$$

where $\mathbf{f}_M^{\mathrm{p}}$ was introduced to take into account point forces at node M. Since $\delta\mathbf{u}_M$ is zero at the Dirichlet boundaries, but otherwise arbitrary, the expression within brackets must be zero, leading to the discrete equilibrium equations:

$$\mathbf{f}_M^{\mathrm{int}} + \mathbf{M}_{MN}\ddot{\mathbf{u}}_N = \mathbf{f}_M^{\mathrm{ext}} \tag{3839}$$

where:

- $\mathbf{f}_M^{\mathrm{int}}$ is the internal force at node M:

$$\mathbf{f}_M^{\mathrm{int}} = \int_{V_0} \widehat{\mathbf{B}}_M^{\mathrm{T}}\,\mathbf{S}\ \mathrm{d}V \tag{3840}$$

- $\mathbf{M}_{MN}\ddot{\mathbf{u}}_N$ is the inertial term, with the mass matrix $\mathbf{M}_{MN}$ defined as:

$$\mathbf{M}_{MN} = \int_{V_0} \mathcal{N}_M\mathbf{I}\,\mathcal{N}_N\,\rho_0\ \mathrm{d}V \tag{3841}$$

- $\mathbf{f}_M^{\mathrm{ext}}$ is the external force applied at node M:

$$\mathbf{f}_M^{\mathrm{ext}} = \int_{V_0} \mathcal{N}_M\,\mathbf{b}\ \mathrm{d}V + \int_{\Gamma} \mathcal{N}_M\overline{\boldsymbol{\tau}}\ \mathrm{d}\Gamma + \mathbf{f}_M^{\mathrm{p}} \tag{3842}$$

  which includes nodal forces resulting from prescribed body forces, surface tractions, and point forces.

Stiffness Matrix

Eqn. (3840) is linear only for the special case of infinitesimal strain, and a linear constitutive law. For large displacements, the internal forces are a nonlinear function of the displacement. The sensitivity of the internal forces with respect to the displacement is defined by the stiffness matrix:

$$\mathbf{K}_{MN} = \frac{\partial\mathbf{f}_M^{\mathrm{int}}}{\partial\mathbf{u}_N} \tag{3843}$$

The stiffness matrix can be expressed as the sum of two terms, the material stiffness and the geometric stiffness:

$$\mathbf{K}_{MN} = \mathbf{K}_{MN}^{\mathrm{mat}} + \mathbf{K}_{MN}^{\mathrm{geom}} \quad \forall M, N \in \mathbb{N} \tag{3844}$$

The material stiffness is:

$$\mathbf{K}_{MN}^{\mathrm{mat}} = \int_{\Omega} \widehat{\mathbf{B}}_M^{\mathrm{T}} \mathbf{D} \widehat{\mathbf{B}}_N \, \mathrm{d}V \tag{3845}$$

where $\mathbf{D}$ is the sensitivity of the 2nd Piola-Kirchoff stress tensor to the Green-Lagrange strain tensor, which is a 6x6 matrix in Voigt notation:

$$\mathbf{D} = \frac{\partial \mathbf{S}}{\partial \mathbf{E}} \tag{3846}$$

The geometric stiffness is:

$$\mathbf{K}_{MN}^{\mathrm{geom}} = \mathbf{I} \int_{\Omega} \frac{\partial \mathcal{N}_M}{\partial \mathbf{X}}^{\mathrm{T}} \mathbf{S} \frac{\partial \mathcal{N}_M}{\partial \mathbf{X}} \, \mathrm{d}V \tag{3847}$$

For linear geometry (infinitesimal strain assumption), the geometric stiffness is neglected. If the stress-strain relationship is also linear, the internal nodal forces become linear functions of the nodal displacements:

$$\mathbf{f}_M^{\mathrm{int}} = \mathbf{K}_{MN}^{\mathrm{mat}} \, \mathbf{u}_N \tag{3848}$$

Rayleigh Damping

The formulation presented in the previous sections does not take into account the damping mechanisms that arise in time-dependent systems. Damping is the dissipation of energy in the solid structure due to a combination of different phenomena, including molecular interaction within the material.

In dynamic problems, the contribution of damping forces can be taken into account by including a velocity-dependent damping term, $c\dot{\mathbf{u}}$, in the equation of motion (Eqn. (3815)):

$$\rho\ddot{\mathbf{u}} + c\dot{\mathbf{u}} - \nabla\cdot\boldsymbol{\sigma} - \mathbf{b} = 0 \tag{3849}$$

which assumes a linear relationship between the damping force and velocity. The weak form of Eqn. (3849) can be constructed and discretized as described in the previous section, leading to the general discretized equation for a linear elastic damped system:

$$\mathbf{M}\,\ddot{\mathbf{u}}(t) + \mathbf{C}\,\dot{\mathbf{u}}(t) + \mathbf{K}\,\mathbf{u}(t) = \mathbf{f}^{\mathrm{ext}} \tag{3850}$$

The damping matrix in each element is:

$$\mathbf{C}_{MN} = \int_{V_0} \mathcal{N}_M c\,\mathbf{I}\mathcal{N}_N \;\mathrm{d}V \tag{3851}$$

As damping is a complex combination of different phenomena, the damping matrix is often approximated using Rayleigh damping, which models the damping matrix as a linear combination of the stiffness and mass matrices:

$$\mathbf{C} = \tau_K\mathbf{K} + f_M\mathbf{M} \tag{3852}$$

You can determine the scalar coefficients τ_K and f_M from a desired modal damping factor and the knowledge of the first two eigenvalues of the undamped system.

The eigenvalue problem of the undamped system is:

$$(\mathbf{K} - \omega_i^2\,\mathbf{M})\mathbf{x}_i = \mathbf{0} \tag{3853}$$

where ω_i and $\mathbf{x}_i$ are the eigenvalues and eigenvectors, respectively:

$$\mathbf{X} = \begin{bmatrix}\mathbf{x}_1 & \dots & \mathbf{x}_i & \dots & \mathbf{x}_n\end{bmatrix}, \qquad \boldsymbol{\Omega} = \begin{bmatrix}\omega_1^2 & & & & \\ & \ddots & & & \\ & & \omega_i^2 & & \\ & & & \ddots & \\ & & & & \omega_n^2\end{bmatrix} \tag{3854}$$

Since $\mathbf{K}$ and $\mathbf{M}$ are symmetric matrices, the eigenvectors are orthogonal. The eigenvectors are normalized with respect to the mass matrix $\mathbf{M}$. After solving the undamped eigenvalue problem Eqn. (3853), the equations for the damped system can be decoupled by multiplying Eqn. (3850) by $\mathbf{X}^{\mathrm{T}}$:

$$\begin{aligned}\mathbf{X}^{\mathrm{T}}\mathbf{M}\,\mathbf{X} &= \mathbf{I}\\ \mathbf{X}^{\mathrm{T}}\mathbf{K}\,\mathbf{X} &= \mathbf{\Omega}\\ \mathbf{X}^{\mathrm{T}}\mathbf{C}\,\mathbf{X} &= \mathbf{\Gamma}\end{aligned} \tag{3855}$$

It is possible to write the solution of the damped system, $\mathbf{u}$, as a linear combination of the (undamped) eigenvectors $\mathbf{x}_i$:

$$\mathbf{u} = \mathbf{X}\,\mathbf{g}(t) \tag{3856}$$

where $\mathbf{g}(t)$ are the generalized, or modal, degrees of freedom. Eqn. (3850) then reduces to the diagonal form:

$$\mathbf{I}\,\ddot{\mathbf{g}}(t) + \mathbf{\Gamma}\,\dot{\mathbf{g}}(t) + \mathbf{\Omega}\,\mathbf{g}(t) = \mathbf{q}(t) \tag{3857}$$

with the generalized loads:

$$\mathbf{q}(t) = \mathbf{X}^{\mathrm{T}}\,\mathbf{f}^{\mathrm{ext}}(t) \tag{3858}$$

Assuming frequency-proportional damping, with modal damping factors ζ_i, the diagonal matrix Γ becomes:

$$\mathbf{\Gamma} = \begin{bmatrix} 2\zeta_1\omega_1 & & & & \\ & \ddots & & & \\ & & 2\zeta_i\omega_i & & \\ & & & \ddots & \\ & & & & 2\zeta_n\omega_n \end{bmatrix} \tag{3859}$$

Each row of Eqn. (3857) is then:

$$\ddot{g}_i(t) + 2\,\zeta_i\,\omega_i\,\dot{g}_i(t) + \omega_i^2\,g_i(t) = q_i(t) \tag{3860}$$

The Rayleigh damping coefficients τ_K and f_M can now be determined by comparing the damping term in Eqn. (3860) with Eqn. (3852):

$$2\,\zeta_i\,\omega_i = f_M + \tau_K\,\omega_i^2 \tag{3861}$$

For example, you can write Eqn. (3861) for the first two fundamental eigenfrequencies ω_1 and ω_2:

$$\begin{aligned}\tau_K &= \frac{2\zeta_1\omega_1 - 2\zeta_2\omega_2}{\omega_1^2 - \omega_2^2}\\ f_M &= 2\zeta_1\omega_1 - \frac{2\zeta_1\omega_1 - 2\zeta_2\omega_2}{\omega_1^2 - \omega_2^2}\,\omega_1^2\end{aligned} \tag{3862}$$

with $\tau_K > 0$ and $f_M > 0$.

If you assume a uniform modal damping factor ζ for both frequencies, τ_K and f_M become:

$$\tau_K = \frac{2\zeta}{\omega_1 + \omega_2}$$
$$f_M = 2\zeta\omega_1\left(1 - \frac{\omega_1}{\omega_1 + \omega_2}\right) \tag{3863}$$

A typical modal damping factor is $\zeta = 0.02$.

An even simpler choice is to restrict the Rayleigh damping to stiffness proportional damping, by assuming $f_M = 0$ and $\tau_K > 0$, and tune the parameter with the fundamental eigenfrequency:

$$2\,\zeta\,\omega = \tau_K\,\omega^2 \tag{3864}$$

If you express the eigenfrequency ω as a function of the period T:

$$\omega = \frac{2\pi}{T} \tag{3865}$$

τ_K reduces to:

$$\frac{\zeta\,T}{\pi} = \tau_K \tag{3866}$$

If you then choose the time-step for the stiffness proportional damping $\tau_K = \Delta t$, the modal damping factor becomes:

$$\zeta = \frac{\Delta t}{T}\,\pi \tag{3867}$$

A time-step of $\Delta t = T/100$, with the choice of $\tau_K = \Delta t$, gives a modal damping factor of:

$$\zeta = \frac{\pi}{100} = 0.0314 \tag{3868}$$

Static and Dynamic Solutions

In dynamic problems, the displacement field is time-dependent. In static problems, the transient terms are neglected and the solution does not depend on time.

The nomenclature that is used in solid mechanics is different from the one that is used in fluid mechanics. There is not an exact one-to-one relationship between *Static* and *Steady*, and between *Dynamic* and *Transient* (*Unsteady*).

In fluid mechanics transient simulations, the inertial terms are included and the solution depends on the initial conditions and time. In solid mechanics, a load can be time-dependent but the solution can still be considered static, that is, the typical time scales to load or unload a part are very long compared to the time scales for flexural or sound waves to travel through the part. However, in truly dynamic problems, the transients due to the low-frequency flexural waves or high-frequency sound waves are important and the inertial terms must be included.

Statics

The static solution seeks the displacement field $\mathbf{u}_M$, such that the internal forces are in equilibrium with the external forces. In static problems, the inertial terms are neglected, and Eqn. (3839) reads:

$$\mathbf{f}_M^{\text{int}} = \mathbf{f}_M^{\text{ext}} \quad \forall M \in \mathbb{N} \tag{3869}$$

where $\mathbb{N}$ is the set of nodes of an element.

For large deformations, the internal forces are nonlinear in the displacements, and STAR-CCM+ solves Eqn. (3869) with Newton iterations:

$$\mathbf{K}_{MN}^i \, \Delta\mathbf{u}_N^i = \mathbf{r}_M^i \tag{3870}$$

where $\mathbf{r}_M^i = \mathbf{f}_M^{\text{ext}} - \mathbf{f}_M^{\text{int}}$ are the residual forces at node M.

STAR-CCM+ solves the equation for the displacement increments $\delta\mathbf{u}_N^i$ and updates the displacements as:

$$\mathbf{u}_N^{i+1} = \mathbf{u}_N^i + \delta\mathbf{u}_N^i \tag{3871}$$

The iteration starts with a given initial condition $\mathbf{u}_N^0$. For a linear problem, the solution is independent of the initial conditions. In addition, a direct solver computes the solution in one iteration.

Dynamics

The dynamic solution seeks the displacement field $\mathbf{u}_M$ that satisfies the equation:

$$\mathbf{M}_{MN}\ddot{\mathbf{u}}_N + \mathbf{C}_{MN}\dot{\mathbf{u}}_N = \mathbf{f}_M^{\text{ext}} - \mathbf{f}_M^{\text{int}} \quad \forall M \in \mathbb{N} \tag{3872}$$

where $\mathbf{M}_{MN}$ and $\mathbf{C}_{MN}$ are the mass and damping matrices.

STAR-CCM+ provides two different approximations of the accelerations and velocities:

1st Order Backward Euler Method

The 1st order Backward Euler method approximates the acceleration and velocity at time-step *n* as:

$$\ddot{\mathbf{u}}_N^n = \frac{\dot{\mathbf{u}}_N^n - \dot{\mathbf{u}}_N^{n-1}}{\Delta t}$$
$$\dot{\mathbf{u}}_N^n = \frac{\mathbf{u}_N^n - \mathbf{u}_N^{n-1}}{\Delta t} \tag{3873}$$

This first order approximation is not recommended for high-resolution structural dynamics, as it can introduce a large amount of numerical damping. However, the numerical damping can be used to remove unwanted initial transients, or when the goal is to reach a quasi-static solution.

2nd Order Newmark Method

The 2nd order Newmark method approximates the velocity and position at the time-step *n* as:

$$\dot{\mathbf{u}}_N^n = \dot{\mathbf{u}}_N^{n-1} + \left(\gamma\ddot{\mathbf{u}}_N^n + (1-\gamma)\ddot{\mathbf{u}}_N^{n-1}\right)\Delta t$$
$$\mathbf{u}_N^n = \mathbf{u}_N^{n-1} + \dot{\mathbf{u}}_N^{n-1}\Delta t + \left(\beta\ddot{\mathbf{u}}_N^n + \left(\tfrac{1}{2}-\beta\right)\ddot{\mathbf{u}}_N^{n-1}\right)\Delta t^2 \tag{3874}$$

In general, on the initial time-step the acceleration $\ddot{\mathbf{u}}_N^0$ is assumed to be zero.

The method is 2nd order accurate when $\gamma = \frac{1}{2}$ and $\beta = \frac{1}{4}$. The method is absolutely stable when:

$$\gamma \geq \tfrac{1}{2}, \quad \beta = \tfrac{1}{4}\left(\tfrac{1}{2}+\gamma\right)^2 \tag{3875}$$

Values of $\gamma > \frac{1}{2}$ introduce numerical damping.

The residual is:

$$\mathbf{r}_M = -\mathbf{M}_{MN}\ddot{\mathbf{u}}_N - \mathbf{C}_{MN}\dot{\mathbf{u}}_N - \mathbf{f}_M^{\text{int}} + \mathbf{f}_M^{\text{ext}} \qquad \forall M \in \mathbb{N} \tag{3876}$$

And the effective stiffness matrix for the Newmark method becomes:

$$\widetilde{\mathbf{K}}_{MN} = -\frac{\partial \mathbf{r}_M}{\partial \mathbf{u}_N} = +\mathbf{K}_{MN} + \frac{\mathbf{M}_{MN}}{\beta\Delta t^2} + \frac{\gamma\mathbf{C}_{MN}}{\beta\Delta t} \tag{3877}$$

Contents:

Residual Norms

Residual Norms

During the numerical solution, STAR-CCM+ allows you to monitor the residual forces, the displacement increments, and the strain energy increments by means of different norms.

At every iteration *i*, STAR-CCM+ computes the solution update $\Delta\mathbf{u}^i \in \mathbb{R}^N$ by solving the linear system of equations:

$$\mathbf{K}^i\Delta\mathbf{u}^i = \mathbf{r}^i \tag{3878}$$

with Newton iterations (see Static and Dynamic Solutions). $\mathbf{K}^i \in \mathbb{R}^N \times \mathbb{R}^N$ and $\mathbf{r}^i \in \mathbb{R}^N$ denote the stiffness matrix and the residual force vector at the *i*-th iteration, respectively, with N being the size of the system, that is, the total number of degrees of freedom.

After each iteration, STAR-CCM+ computes the norms of the residual forces, the displacement increments, and the strain energy increments. When specified, these norms are normalized to the norm-value calculated for the 1st iteration:

$$\overline{\mathbf{r}^i} = \frac{\left\|\mathbf{r}^i\right\|}{\left\|\mathbf{r}^1\right\|};\ \overline{\Delta\mathbf{u}^i} = \frac{\left\|\Delta\mathbf{u}^i\right\|}{\left\|\Delta\mathbf{u}^1\right\|};\ \overline{W^i} = \frac{W^i}{W^1} \tag{3879}$$

The norms of the residual force vector, $\left\|\mathbf{r}^i\right\|$, and the displacement increment vector, $\left\|\Delta\mathbf{u}^i\right\|$, can be calculated using different methods (see Residual Force and Displacement Increment Norms), whereas the strain energy increment, W^i, is calculated as the inner product between the displacement increment vector and the residual force vector (see Strain Energy Increment).

Residual Force and Displacement Increment Norms

STAR-CCM+ allows you to monitor the residual forces and the displacement increments by means of three different norms. To simplify the notation, the following definitions of norm consider a general vector $\mathbf{a}$ that represents either the displacement increment $\Delta\mathbf{u}^i$ or the residual force $\mathbf{r}^i$ at the *i*-th iteration.

The components of $\mathbf{a}$ are denoted by a_j, with $j = 1, \ldots, N$, where N is the vector dimension (that is, the total number of degrees of freedom of the system, Eqn. (3878)).

Maximum Norm

The maximum norm of a vector $\mathbf{a}$ is the maximum of the absolute values of its components, a_j:

$$\left\|\mathbf{a}\right\|_{\max} = \max_{j = 1, \ldots, N}\left(\left|a_j\right|\right) \tag{3880}$$

L^1-norm

The L^1 norm of a vector $\mathbf{a}$ is the arithmetic mean of the absolute values of its components, a_j:

$$\left\|\mathbf{a}\right\|_{L^1} = \frac{1}{N}\sum_{j = 1}^{N}\left|a_j\right| \tag{3881}$$

Euclidean Norm

The Euclidean norm, or L^2-norm, of a vector $\mathbf{a}$ is the Euclidean length of $\mathbf{a}$ normalized to the number of its components:

$$\left\|\mathbf{a}\right\|_{L^2} = \frac{1}{N}\sqrt{\sum_{j = 1}^{N} a_j^2} \tag{3882}$$

Strain Energy Increment

STAR-CCM+ computes the strain energy increment added in one iteration of the Newton scheme as:

$$W^i = \frac{1}{2} \sum_{j=1}^{N} \Delta u_j^i \, r_j^i \tag{3883}$$

where Δu_j^i and r_j^i are the components of the displacement increment vector and the residual force vector at the *i*-th iteration, and N is the vector dimension.

Bibliography

[715] *O.C. Zienkiewicz, R.L. Taylor and J.Z. Zhu, The Finite Element Method: Its Basics and Fundamentals, Elsevier Butterworth-Heinemann, Oxford, 6th edition, 2005.*

[716] *O.C. Zienkiewicz & R.L. Taylor, The Finite Element Method for Solid and Structural Mechanics, Elesevier Butterworth-Heinemann, Oxford, 6th edition, 2006.*

[717] *Chowdhury and S. P. Dasgupta, Computation of Rayleigh damping coefficients for large systems, Electronic Journal of Geotechnical Engineering, vol. 8, 2003.*

Aeroacoustics

Computational aeroacoustics (CAA) is a branch of multiphysics modeling and simulation that involves identifying noise sources that are induced by fluid flow and propagation of the sound waves then generated.

Noise sources originate from various types of flow, such as:

- Turbulent flow over solid bodies (bluff body flows)
- Turbulent boundary layer flows (for example, automobile, aircraft components)
- High-speed turbulent shear flows (for example, free jet flow)
- High-speed impinging flows (for example, jet impingement, rocket exhaust noise)
- Structural vibration that is induced by fluid flow (fluid-structure interactions)
- High-speed rotating flows (for example, rotorcrafts or turbomachinery)
- Turbulent combustion (reacting flows)
- Blast waves (explosions)

A typical CAA simulation requires the following components:

- Navier-Stokes equations for fluid flow
- High-resolution turbulence models
- Analytical or computational acoustic wave propagation models

The noise signatures at the locations of interest exhibit corresponding noise spectra—that is, the intensities of sound pressure level over a range of frequencies. The noise characteristic can be tonal noise with a distinct peak at a frequency (such as engine noise; jet impingement noise, or Noise, Vibration, and Harshness (NVH)) or broadband noise spread over a frequency range (typical of turbulence-induced noise).

The sound pressure level (*SPL* measured in decibel, dB) is:

$$SPL = 20 \log_{10} \frac{p_{rms}}{p_{ref}} \, dB \tag{3884}$$

where:

- $p_{rms} =$ root mean square pressure
- $p_{ref} =$ reference pressure (usually 20 μPa)

The mechanism of noise generation differs according to the underlying physics of the flow. When modeling noise generation using computational methods, it is essential to capture the noise sources and frequencies that are relevant to the acoustic analysis. Resolution of the noise sources relies on the fidelity of the turbulence modeling.

The computational aeroacoustics methods in STAR-CCM+ target the following applications:

Prediction of near-field noise by Large-Eddy Simulations (LES) or Detached-Eddy Simulations (DES)

Although LES and DES simulations can predict noise generation in the near-field, they are not generally suitable for predicting noise propagation to far-field locations. The flow fluctuations that generate noise are orders of magnitude lower than the hydrodynamic flow properties. Without adequate mesh resolution, sound waves quickly dissipate away from the source region. The mesh density that is required to preserve this acoustic signature at far distances restricts the feasibility of pure CFD for real world applications.

It is more practical to consider noise generation and propagation phenomena separately, adopting an appropriate hybrid CFD/CAA method. In this method, LES or DES is coupled (one-way) with a noise propagation aeroacoustic model.

Prediction of mid-field noise by acoustic wave equations, Linearized Euler Equations (LEE), or Acoustic Perturbation Equations (APE) (see Acoustic Wave Model)

High fidelity LES/DES can be combined with other accurate mesh-based CAA methods (typically requiring high-order space and time discretization) such as Linearized Euler Equations (LEE) or Acoustic Perturbation Equations (APE), but not without extreme computational burden.

Prediction of far-field noise by the Ffowcs Williams-Hawkings equation (FWH) (see Ffowcs Williams-Hawkings Model)

The Ffowcs Williams-Hawkings equation considers all fundamental components of noise sources (monopole, dipole, and quadrupole). The surface integral methods such as those of the FWH method are a practical approach for noise propagation in the far-field, being able to superimpose all sources of noise over an enclosed surface and provide acoustic signatures by accurate meshless analytical solutions of a surface integral equation.

The noise propagation models in STAR-CCM+ are based on Lighthill's equation Eqn. (3931) [743], Curle's equation, and the Ffowcs Williams-Hawkings equation (FWH). Lighthill's equation derives the acoustic field from the Navier-Stokes equations and some basic assumptions, containing the definition of quadrupole noise sources from turbulent flows. STAR-CCM+ uses LEE and Curle for noise source modeling only, not for acoustic wave modeling (LEE), or for propagation (Curle).

The FWH method is based on the assumption of sound propagation in free space, so it cannot consider any embedded structure in the propagation region. On the other hand, a Boundary Element Method (BEM) for the solution of the Helmholtz equation (a wave equation in the frequency domain) by a boundary integral equation can provide solutions of acoustic field with embedded structures. See [724], [728], [754].

Contents:

Broadband Noise Source Models
Acoustic Wave Model
Acoustic Suppression Zone Model
Ffowcs Williams-Hawkings Model
Aeroacoustics Bibliography

Broadband Noise Source Models

For near-field prediction, STAR-CCM+ offers a group of models capable of evaluating the broadband sources of noise. The noise sources consist of a volume distribution of quadrupole sources and a surface distribution of dipole sources.

In the frequency domain, a broadband noise has a continuous spectrum, where the acoustic energy is continuously distributed at all frequencies in a given range. These models calculate the location and strength of the main sources of noise that are generated aerodynamically. They use flow field quantities from Reynolds-Averaged Navier-Stokes (RANS) Turbulence equations. Because these broadband noise models use RANS simulations, the required computational time is small.

The following broadband noise source models perform the following operations on the local acoustic power:

Curle noise source model

estimates the sound per unit surface generated by dipole sources

Goldstein axisymmetric noise source model

Proudman noise source model

estimate the sound per unit volume generated by quadrupole sources

Lilley noise source model

Linearized Euler Equation (LEE) noise source model

estimate the sound generated by turbulent shear flow

Stochastic Noise Generation and Radiation (SNGR) model

computes a synthetic turbulence field

Lighthill stress tensor

estimates noise sources generated by flow

Acoustic Suppression Zone model

suppresses spurious reflections

Mesh frequency cutoff analysis

checks if the grid is fine enough to resolve turbulent flow structures of interest

The decomposition of the main sources of aerodynamically generated sound can be done onto a shear noise term, representing the interaction between the mean shear velocities and turbulent velocity components of the flow, and the self-noise term, representing turbulence-turbulence interactions.

Speed of Sound

The far-field speed of sound a_0 is the distance that a sound wave travels in a unit of time.

$$a_0 = \sqrt{\left(\gamma R T_0\right)} \tag{3885}$$

where:

- γ is the adiabatic index or the ratio of heat capacity at constant pressure to the heat capacity at constant volume for the fluid.
- R is the universal gas constant
- T_0 is the absolute temperature, measured in Kelvin.

In dry air at 293 K, the speed of sound is 343.2 m/s.

Curle Noise Source Model

The Curle Noise Source model evaluates the noise from a turbulent boundary layer flow over a solid body, assuming isotropic turbulence and low Mach numbers.

This model is typically used for air-handling subsystem (AHS) applications, including rotating parts (such as blowers), heat exchangers, mufflers, distribution ducts, and wall-enclosed spaces.

The Curle model represents dipole sources of noise, that is, noise coming from the fluctuating surface pressure that results from solid boundaries acting on the fluid. Specifically, this model calculates the local contribution to the acoustic power per unit area of body surface.

The acoustic pressure can be written in terms of the local contributions as:

$$\rho'\left(\mathbf{x}, t\right) = \frac{1}{\left(4\pi a_0^3\right)} \int_S \left[\frac{(\mathbf{x}-\mathbf{y})}{r^2} \frac{\partial p}{\partial t}\left(\mathbf{y}, t - \frac{r}{a_0}\right)\right] \cdot \mathbf{n}\ dS\left(\mathbf{y}\right) \tag{3886}$$

where:

- ρ' is the acoustic pressure.
- a_0 is the far-field sound speed.
- $t - r/a_0$ is the emission time.
- p is the surface pressure.
- $\mathbf{x}$ is the position in the far-field.
- $\mathbf{y}$ is the point in the far field where the noise is calculated, measured a face centroid on the solid boundary mesh.
- r is $|\mathbf{x} - \mathbf{y}|$.
- $\mathbf{n}$ is the wall-normal direction.

On the assumption of small perturbations and an adiabatic problem:

$$\frac{p}{\rho^\gamma} = ct \tag{3887}$$

where $\gamma = c_P / c_V$ is the ratio of specific heats at constant pressure (c_P) and constant volume (c_V).

This value can be used to relate variations in acoustic pressure with density perturbations:

$$p' = a_0^2 \rho' \tag{3888}$$

Then Eqn. (3886) becomes:

$$p'(\mathbf{x},t) = \frac{1}{(4\pi a_0)}\int_S \left[\frac{(\mathbf{x}-\mathbf{y})}{r^2}\frac{\partial p}{\partial t}\left(\mathbf{y}, t-\frac{r}{a_0}\right)\right]\cdot\mathbf{n}\, dS(\mathbf{y}) \tag{3889}$$

In the far field, the acoustic directional intensity along $\mathbf{x}-\mathbf{y}$, per unit surface of the solid body surface, is approximated with:

$$\overline{p'^2} \approx \frac{1}{16\pi^2 a_0^2}\int_S \frac{(\cos\theta)^2}{r^2}\overline{\left[\frac{\partial p}{\partial t}\left(\mathbf{y}, t-\frac{r}{a_0}\right)\right]^2} A_c(\mathbf{y})\ \mathrm{d}\ S(\mathbf{y}) \tag{3890}$$

where:

- A_c is the correlation area.
- θ is the angle between $\mathbf{r}$ and the wall-normal $\mathbf{n}$, so that $r\ \cos\theta = \mathbf{r}\cdot\ \mathbf{n}$

The measure of the local contribution to acoustic power per unit surface, Surface Acoustic Power or *SAP*, can be computed from:

$$SAP = \frac{1}{\rho_0 a_0}\left[\int_0^{(2\pi)}\int_0^{\pi}\overline{p'^2} r^2 \sin\theta\, d\theta\, d\gamma\right] = \int_S I(\mathbf{y}) dS(\mathbf{y}) \qquad = \int_S \frac{A_c(\mathbf{y})}{(12\rho_0\pi a_0^3)}\overline{\left(\frac{\partial p}{\partial t}\right)^2} dS(\mathbf{y}) \tag{3891}$$

where $I(\mathbf{y})$ is the directional acoustic intensity per unit surface.

The model can be activated for steady and unsteady cases with Reynolds-Averaged Navier-Stokes (RANS) Turbulence Models, which provide turbulence time scale, turbulence length scale, and wall shear stress necessary to compute $\overline{(\partial p/\partial t)^2}$, the mean-square time derivative of the source surface pressure [738].

The acoustic power per unit surface can be reported in dimensional units ($\mathrm{W/m^2}$) and in dB:

$$SAP(\mathrm{dB}) = 10\ \log\frac{SAP}{P_{ref}} \tag{3892}$$

where P_{ref} is the reference acoustic power.

Goldstein Axisymmetric Noise Source Model

The Goldstein model is based on Lighthill's theory for predicting aerodynamic noise from a turbulent shear flow on an assumption of locally homogeneous and anisotropic turbulence.

Goldstein [737] generalizes Ribner's model [751] for axisymmetric turbulent flow by eliminating Ribner's assumption of the space-time correlations of Reynolds stresses.

This model computes the local noise contribution per unit volume for anisotropic turbulence in axisymmetric turbulent jets.

The local contribution to the total acoustic power per unit cell volume is computed from:

$$G_{AP}(\mathbf{y}) = \int_0^{2\pi}\int_0^{\pi} I(r,\theta/\mathbf{y}) r^2 \sin(\theta) d\theta d\gamma = 2\pi r^2 \int_0^{\pi} I(r,\theta/\mathbf{y})\sin(\theta) d\theta \tag{3893}$$

where r and θ are the radial and angular coordinates of the receiver location, with the receiver assumed to be sufficiently far away from the flow. $I(r,\theta/\mathbf{y})$ is the directional acoustic intensity per unit volume of a jet:

$$I(r,\theta/\mathbf{y}) = \frac{12\rho_0\omega_f^4 L_1 L_2^2\left(\overline{u_{tl}^2}\right)^2}{5\pi c_0^5 r^2}\frac{D_{\text{self}}}{C^5} + \frac{24\rho_0\omega_f^4 L_1 L_2^4\left(\overline{u_{tl}^2}\right)}{\pi c_0^5 r^2}\left(\frac{\partial U}{\partial r}\right)^2\frac{D_{\text{shear}}}{C^5} \tag{3894}$$

C is the modified convection factor:

$$C = 1 - M_c\left(\cos\theta\right) \tag{3895}$$

The decomposition of the sound intensity into shear and self-noise (with no coupling terms) occurs under the assumption that turbulence is locally homogeneous:

$$I\left(r,\theta/\mathbf{y}\right) = I_{\text{self}}\left(r,\theta/\mathbf{y}\right) + I_{\text{shear}}\left(r,\theta/\mathbf{y}\right) \tag{3896}$$

$$I_{\text{self}}\left(r,\theta/\mathbf{y}\right) = \frac{12\rho_0\omega_f^4 L_1 L_2^2\left(\overline{u_{tl}^2}\right)^2 D_{\text{self}}}{5\pi c_0^5 r^2 C^5} \tag{3897}$$

$$I_{\text{shear}}\left(r,\theta/\mathbf{y}\right) = \frac{24\rho_0\omega_f^4 L_1 L_2^4\left(\overline{u_{tl}^2}\right) v_{\text{grad}}^2 D_{\text{shear}}}{\pi c_0^5 r^2 C^5} \tag{3898}$$

where:

- ρ_0 is the far-field density.
- c_0 is the far-field speed of sound.
- $v_{\text{grad}} = \frac{\partial U}{\partial r}$

$$D_{\text{self}} = 1 + 2\left(\frac{M}{9} - N\right)\cos^2\theta \ \ \sin^2\theta + \ \ \frac{1}{3}\left[\frac{M^2}{7} + M - 1.5N\left(3 - 3N + \frac{1.5}{\Delta^2} - \frac{\Delta^2}{2}\right)\right]\sin^4\theta \tag{3899}$$

$$D_{\text{shear}} = \cos^2\theta\left[\cos^2\theta + \frac{1}{2}\left(\frac{1}{\Delta^2} - 2N\right)\sin^2\theta\right] \tag{3900}$$

The effect of the anisotropic structure of the turbulence appears through the following parameters:

$$\begin{gathered} M = \left[\frac{3}{2}\left(\Delta - \frac{1}{\Delta}\right)\right]^2 \quad N = 1 - \frac{\left(\overline{u_{t2}^2}\right)}{\left(\overline{u_{t1}^2}\right)} \quad \Delta = \frac{L_2}{L_1} \\ L_1 = \frac{\left(\overline{u_{t1}^2}\right)^{\frac{3}{2}} T^3}{L^2} \quad L_2 = \frac{\left(\overline{u_{t2}^2}\right)^{\frac{3}{2}} T^3}{L^2} \quad \omega_f = 2\pi\left(\frac{1}{T}\right) \end{gathered} \tag{3901}$$

where L_1 and L_2 are the longitudinal and transverse correlation lengths, which are not equal. The longitudinal and transversal turbulence velocities $\overline{u_{t1}^2}$ and $\overline{u_{t2}^2}$ are not equal in all directions, so $N \neq 0$ and $\Delta \neq 1$. In the limiting case of isotropic turbulence the directivity expressions become identical to those of the Ribner model, so $N = M = 0$ and $\Delta = 1$.

The longitudinal and transverse turbulence velocities $\overline{u_{t1}^2}$ and $\overline{u_{t2}^2}$ are computed using the following relations:

$$\overline{u_{t1}^2} = \frac{8}{9}\left(\frac{L^2}{T^2}\right) \quad \overline{u_{t2}^2} = \frac{4}{9}\left(\frac{L^2}{T^2}\right) \tag{3902}$$

where:

- T is the turbulence time scale, $T = k/\varepsilon$.
- L is the turbulence length scale, $L = k^{3/2}/\varepsilon$.

k is the turbulent kinetic energy and ε is the dissipation rate.

In STAR-CCM+ the total acoustic power per unit volume can be reported in dimensional units ($\mathrm{W/m^3}$) and in dB:

$$G_{AP}(\mathrm{dB}) = 10\ \log\left(\frac{G_{AP}}{P_{ref}}\right) \tag{3903}$$

where P_{ref} is the reference acoustic power.

See also Bechara and others [720].

Proudman Noise Source Model

Like the Goldstein model, the Proudman noise source model evaluates acoustic power per unit volume, and the sound is from quadrupoles, such as the areas around blower blades. This model has been used for air handling subsystem (AHS) applications, including rotating parts (such as blowers), heat exchangers, and distribution ducts. The Proudman model assumes isotropic turbulence.

Proudman considers the generation of noise by isotropic turbulence. In Proudman's high-Reynolds model for isotropic turbulence in near incompressible flow, the *AP* (acoustic power) per unit volume is:

$$AP = \alpha\rho_0\frac{u^3}{l}\frac{u^5}{c_0^5} \tag{3904}$$

where:

- α is a constant related to the shape of the longitudinal velocity correlation.
- u is the root mean square of one of the velocity components.
- l is the longitudinal integral length scale of the velocity.
- ρ_0 is the far-field density.
- c_0 is the far-field sound speed.

Different researchers have found different values for α, ranging from 0.629 to 13. STAR-CCM+ uses a re-scaled constant α_c from Sarkar and Hussaini [753].

In Proudman's paper [750], the characteristic velocity of energy-containing eddies u can be written as:

$$u = \sqrt{\frac{2}{3}k} \tag{3905}$$

In terms of the turbulence velocity scale and of the turbulence length scale, the local acoustic power due to the unit volume of isotropic turbulence (in $\mathrm{W/m^3}$) becomes:

$$AP = \alpha_c \rho_0 \frac{U^3}{L} \frac{U^5}{a_0^5} \tag{3906}$$

with:

$$U = \frac{L}{T} \ , \quad \alpha_c = 0.629 \tag{3907}$$

where:

- ρ_0 is the far-field density.
- U is the turbulence velocity.
- L is the turbulence length scale.
- T is the turbulence time scale.
- c is the far-field sound speed.

The rescaled constant α_c is based on Direct Numerical Simulation for isotropic turbulence done by Sarkar and Hussaini [753].

The total acoustic power per unit volume can be reported in dimensional units ($\mathrm{W/m^3}$) and in dB:

$$AP(\mathrm{dB}) = 10 \ \log\left(\frac{AP}{P_{ref}}\right) \tag{3908}$$

where P_{ref} is the reference acoustic power.

Lilley Noise Source Model

The Lilley noise source models calculate the turbulent shear flow noise of quadrupole sources from the fluctuating velocity flow field.

Applications include problems where local shear effects dominate, such as flow around solid bodies and shear flows in channels. This model evaluates the location and strength of aeroacoustic sound sources. This model is available only for steady-state simulations.

The Lilley Noise Source model is compatible with the coupled and segregated flow models. Computational costs (specifically CPU time) depend on input of Number of Realizations and Number of Fourier Modes provided to the Stochastic Noise Generation and Radiation (SNGR) Model. The Lilley Noise Source model calculates three terms: Lilley Self-Noise Source, Lilley Shear-Noise Source, and Lilley Total-Noise Source.

The Lilley source terms are available for steady-state solutions computed using a Reynolds-Averaged Navier-Stokes (RANS) Turbulence model. A RANS model is the only type that can provide turbulence length scale and turbulent time scale that are dominated by local shear effects, as is the case of flow around solid bodies and shear flows in channels.

The Lilley equation can be written as:

$$\frac{D}{Dt}\left[\frac{D^2\Pi}{Dt^2} - \frac{\partial}{\partial x_j}\left(c^2\frac{\partial \Pi}{\partial x_j}\right)\right] + 2\left[\frac{\partial u_k}{\partial x_j}\frac{\partial}{\partial x_k}\left(c^2\frac{\partial \Pi}{\partial x_j}\right)\right] = -2\left[\frac{\partial u_k}{\partial x_i}\frac{\partial u_j}{\partial x_k}\frac{\partial u_i}{\partial x_j}\right] + \psi \tag{3909}$$

$$\psi = 2\left[\frac{\partial u_j}{\partial x_i}\frac{\partial}{\partial x_j}\frac{1}{\rho}\frac{\partial \tau_{ik}}{\partial x_k}\right] - \left[\frac{D}{Dt}\frac{\partial}{\partial x_i}\frac{1}{\rho}\frac{\partial \tau_{ij}}{\partial x_j}\right] + \left[\frac{D^2}{Dt^2}\frac{1}{c_p}\frac{DS}{Dt}\right] \tag{3910}$$

where c is the speed of sound, τ_{ik} is the stress tensor and ψ represents the effects of entropy fluctuations and fluid viscosity.

The entropy fluctuation is negligible for problems without heat transfer or combustion. Viscous effects are also negligible for a high Reynolds number flow. It is therefore usual to neglect these terms from the right-hand side. Then Eqn. (3909) becomes:

$$\frac{D}{Dt}\left[\frac{D^2\Pi}{Dt^2}-\frac{\partial}{\partial x_j}\left(c^2\frac{\partial\Pi}{\partial x_j}\right)\right]+2\left[\frac{\partial u_k}{\partial x_j}\frac{\partial}{\partial x_k}\left(c^2\frac{\partial\Pi}{\partial x_j}\right)\right]=-2\left[\frac{\partial u_k}{\partial x_i}\frac{\partial u_j}{\partial x_k}\frac{\partial u_i}{\partial x_j}\right] \tag{3911}$$

where:

$$\Pi=\left(\frac{c_v}{c_p}\right)\ln\frac{p}{p_0} \tag{3912}$$

and p_0 is reference pressure.

By assuming that the flow satisfies the small perturbation assumption, the flow is locally linearized as follows:

$$u_i(\mathbf{x},t)=U_i(\mathbf{x})+u_i'(\mathbf{x},t) \tag{3913}$$

where $U_i(\mathbf{x})$ is the mean velocity field and $u_i'(\mathbf{x},t)$ is the turbulent velocity field. The right-hand side of Eqn. (3911) can be expanded as follows:

$$\begin{aligned}-2\left[\frac{\partial u_k}{\partial x_i}\frac{\partial u_j}{\partial x_k}\frac{\partial u_i}{\partial x_j}\right]=-2\left[\frac{\partial U_k}{\partial x_i}\frac{\partial U_j}{\partial x_k}\frac{\partial U_i}{\partial x_j}\right]-2\left[\frac{\partial u_k'}{\partial x_i}\frac{\partial u_j'}{\partial x_k}\frac{\partial u_i'}{\partial x_j}\right]-\\ 6\left[\frac{\partial U_k}{\partial x_i}\frac{\partial U_j}{\partial x_k}\frac{\partial u_i'}{\partial x_j}\right]-6\left[\frac{\partial u_k'}{\partial x_i}\frac{\partial u_j'}{\partial x_k}\frac{\partial U_i}{\partial x_j}\right]\end{aligned} \tag{3914}$$

$$A=-2\left[\frac{\partial U_k}{\partial x_i}\frac{\partial U_j}{\partial x_k}\frac{\partial U_i}{\partial x_j}\right] \tag{3915}$$

$$B=S_{\mathrm{SelfNoiseTerm}}=-2\left[\frac{\partial u_k'}{\partial x_i}\frac{\partial u_j'}{\partial x_k}\frac{\partial u_i'}{\partial x_j}\right] \tag{3916}$$

$$C=S_{\mathrm{ShearNoiseTerm}}=-6\left[\frac{\partial U_k}{\partial x_i}\frac{\partial U_j}{\partial x_k}\frac{\partial u_i'}{\partial x_j}\right]-6\left[\frac{\partial u_k'}{\partial x_i}\frac{\partial u_j'}{\partial x_k}\frac{\partial U_i}{\partial x_j}\right] \tag{3917}$$

where A represents the mean flow shape change, the Self-Noise Term B contains effects of the turbulent (fluctuating) velocity components, and the Shear Noise Term C contains the interaction between the mean shear flow and turbulent velocity components.

The noise source terms of Lilley's equation are evaluated based on mean velocity fields that are provided by a steady-state RANS solution and turbulent velocity components. These components are generated by the synthetic turbulence method of Bailly and Juve [719], Bechara *et al.* [720], and Kraichnan [742].

Linearized Euler Equation (LEE) Noise Source Model

The Linearized Euler Equations (LEE) account for refraction and convection effects in any sheared mean flows. The equations can be used for a wider range of conditions than the Lilley equation, where the associated source term is a nonlinear function of the fluctuating velocity flow field. See [720].

Note: This model is used for noise sources only, not for solving acoustic wave propagation.

This model is available only for steady-state simulations. As Lilley equation, the LEE Noise Source model computes LEE Self-Noise Source, LEE Shear Noise Source, and LEE Total Noise Source.

The linearized form of the Euler equations [720] derives from the nonlinear Euler equations. The linearized forms can be achieved by superimposing small perturbations of turbulent and acoustic components on the mean flow as follows:

$$\begin{aligned} p_i(\mathbf{x},t) &= P_i(\mathbf{x}) + p_i'(\mathbf{x},t) + p_{ai}(\mathbf{x},t) \\ u_i(\mathbf{x},t) &= U_i(\mathbf{x}) + u_i'(\mathbf{x},t) + u_{ai}(\mathbf{x},t) \\ \rho_i(\mathbf{x},t) &= \overline{\rho_i}(\mathbf{x}) + \rho_i'(\mathbf{x},t) + \rho_{ai}(\mathbf{x},t) \end{aligned} \tag{3918}$$

where the acoustic components p_{ai}, u_{ai}, ρ_{ai} are small compared to the mean components P_i, U_i, $\overline{\rho_i}$ and the turbulent components p_i', u_i', ρ_i'.

Including the decomposition of variables in Eqn. (3918), and considering the assumption that the acoustic perturbations are isentropic:

$$p_a = c_0^2 \rho_a \tag{3919}$$

and that turbulent density fluctuations (ρ') can be neglected, then the Linearized Euler equations for velocity components become:

$$\begin{aligned} S &= \frac{\partial u_{ai}}{\partial t} + U_j \frac{\partial u_{ai}}{\partial x_j} + \frac{1}{\overline{\rho}} \frac{\partial p_a}{\partial x_i} + u_{aj} \frac{\partial U_i}{\partial x_j} - \frac{\rho_a}{\overline{\rho}^2 c_0^2} \frac{\partial P}{\partial x_i} \\ &= -U_j \frac{\partial u'_i}{\partial x_j} - u'_j \frac{\partial U_i}{\partial x_j} - u'_j \frac{\partial p'}{\partial x_i} - \frac{\partial u'_i}{\partial t} + \frac{\partial}{\partial x_j}\left(\overline{u_j' u_i'}\right) \end{aligned} \tag{3920}$$

The second set of terms includes the source terms responsible for sound generation.

The decomposition of the main sources of aerodynamically generated sound can be done onto the shear noise term $S_{\mathrm{ShearNoiseTerm}}$ (representing the interaction between the mean shear velocities and turbulent velocity components of the flow) and the self-noise term $S_{\mathrm{SelfNoiseTerm}}$ (representing turbulence-turbulence interactions):

$$S_{\mathrm{ShearNoiseTerm}} = -U_j \frac{\partial u_i'}{\partial x_j} - u_j' \frac{\partial U_i}{\partial x_j} \tag{3921}$$

$$S_{\mathrm{SelfNoiseTerm}} = -u_j' \frac{\partial u_i'}{\partial x_j} \tag{3922}$$

The turbulent velocity components are synthesized by the Stochastic Noise Generation and Radiation model.

The source terms in the Linearized Euler Equations, similar to source terms in the Lilley Equation, are available for steady-state solutions computed using a Reynolds-Averaged Navier-Stokes (RANS) Turbulence model that can provide turbulence length scale and turbulence time scale.

Stochastic Noise Generation and Radiation (SNGR) Model

The scope of the SNGR model is to compute a synthetic turbulence field.

A time-space turbulent velocity field is generated using random Fourier modes. This method, called the Stochastic Noise Generation and Radiation (SNGR) model, was proposed by Kraichnan [742] and Karweit *et al.* [740], and afterward developed by Bailly and Juve [719], Bechara *et al.* [720]. In STAR-CCM+, the SNGR method requires user input of two properties of the Lilley noise source model: *Number of Realizations* and *Number of Fourier Modes*. See Lilley and LEE Noise Source Properties.

Consider a three-dimensional Fourier decomposition of a turbulent homogeneous isotropic field at a given point at location :

$$\mathbf{u}_t(\mathbf{x}) = \int u_t(k) e^{(jk\mathbf{y})} \, \mathrm{d}\, k \tag{3923}$$

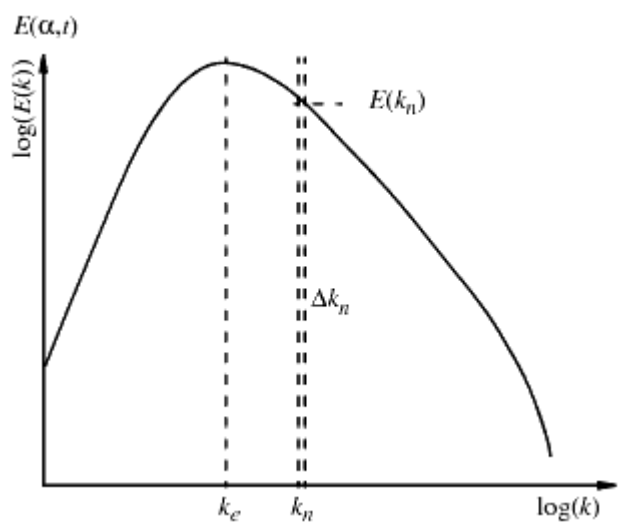
where $\mathbf{u}_t$ is the turbulent velocity field, $\mathbf{k}$ is the wave vector, and j is the imaginary unit, with $j^2 = -1$.

This Fourier integral can be transformed into a limited sum of n Fourier modes:

$$\mathbf{u}_t(\mathbf{x}) = 2\sum_{n=1}^{N} u_n \cos(\mathbf{k}_n \cdot \mathbf{x} + \psi_n)\boldsymbol{\sigma}_n \tag{3924}$$

where u_n, ψ_n and $\boldsymbol{\sigma}_n$ are the amplitude, phase, and directional unit vector respectively of the n-th Fourier mode associated with the wave vector $\mathbf{k}_n$.

The amplitude u_n of each mode is computed so that the turbulence kinetic energy $E(k_n)$ corresponds to the von Karman-Pao energy spectrum for isotropic, homogeneous turbulence:



$$u_n = \sqrt{E(k_n)(\Delta k_n)} \tag{3925}$$

where u_n is the amplitude of the mode and Δk_n is the small interval in the spectrum at k_n.

The expression for the von Karman-Pao energy spectrum can be written as:

$$E(k_n) = \frac{SNGR_\alpha \overline{u^2}}{k_e}\left(\frac{k}{k_e}\right)^4 \frac{1}{\left[1+\left(\frac{k}{k_e}\right)^2\right]^{17/6}} e^{\left(-2\left(\frac{k}{k_\eta}\right)^2\right)} \tag{3926}$$

where:

- k is the wavenumber.
- k_η is the Kolmogorov wavenumber.
- $\overline{u^2}$ is the root-mean-square value of the velocity fluctuations corresponding to the turbulent kinetic energy.
- $SNGR_\alpha$ is a numerical constant.
- k_e is the wavenumber corresponding to the most energy containing eddies at the peak in the spectrum.

Lighthill Equation

Lighthill's equation is an inhomogeneous wave equation that derives the acoustic field by analogy with the Navier-Stokes equations for a compressible fluid (Lighthill analogy [743]). In Lighthill's equation, the acoustic field is described by a second-order tensor known as the Lighthill turbulence stress tensor.

In component form, the mass and momentum conservation equations for a fluid read:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho v_i) = 0 \tag{3927}$$

$$\frac{\partial}{\partial t}(\rho v_i) + \frac{\partial}{\partial x_j}(\rho v_i v_j + p_{ij}) = 0 \tag{3928}$$

where:

- ρ is density.
- v_i and v_j are the velocity components.
- p_{ij} is the stress tensor.

$$p_{ij} = -\sigma_{ij} + (p - p_0)\delta_{ij} \tag{3929}$$

where

- p is the thermodynamic pressure of the flow field, that is, the working pressure.
- δ_{ij} is the Kronecker delta.
- σ_{ij} is the viscous stress tensor.

$$\sigma_{ij} = \mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} - \frac{2}{3}\frac{\partial v_i}{\partial x_i}\delta_{ij}\right) \tag{3930}$$

Using Einstein notation, Lighthill’s equation can be written as:

$$\frac{\partial^2 \rho}{\partial t^2} - c_0^2(\nabla^2 \rho) = \frac{\partial^2 T_{ij}}{(\partial x_i)(\partial x_j)} \tag{3931}$$

where T_{ij} is the “Lighthill turbulence stress tensor” for the acoustic field.

$$T_{ij} = \rho v_i v_j + p_{ij} - c_0^2(\rho - \rho_0)\delta_{ij} \tag{3932}$$

substitute Eqn. (3930) into Eqn. (3932) to get:

$$T_{ij} = \rho v_i v_j - \sigma_{ij} + \delta_{ij}\left[(p - p_0) - {c_0}^2(\rho - \rho_0)\right] \tag{3933}$$

where ρ_0 is the far field density.

In laminar flow, the viscous stress tensor is given by:

$$\sigma_{ij} = \sigma_{ij,\mathrm{lam}} = \mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} - \frac{2}{3}\frac{\partial v_i}{\partial x_i}\delta_{ij}\right) \tag{3934}$$

In turbulent flow, the viscous stress tensor is given by:

$$\sigma_{ij} = \sigma_{ij,\mathrm{lam}} + \sigma_{ij,\mathrm{turb}} = \mu_{eff}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} - \frac{2}{3}\frac{\partial v_i}{\partial x_i}\delta_{ij}\right) \tag{3935}$$

where the effective viscosity is $\mu_{eff} = \mu + \mu_t$, the sum of the laminar and turbulent viscosities.

In inviscid flow, the viscous stress tensor is $\sigma_{ij} = 0$.

Each of the source terms in Eqn. (3933) describes a significant role in the generation of sound as:

- $\rho v_i v_j$ describes unsteady convection of flow (or Reynold's Stress)
- σ_{ij} describes sound shear
- $\delta_{ij}\left(p - c_0^2\rho\right)$ describes nonlinear acoustic generation processes

Mesh Frequency Cutoff Formulation

The main value in performing a mesh-frequency cutoff analysis of a steady-state solution is to determine in advance of a transient calculation whether the chosen grid has sufficient resolution to capture the turbulence flow structures in the frequency range of interest.

Mendonça [749] shows that, given a cell dimension D and local turbulent kinetic energy k, the smallest length scale of a turbulent eddy structure that is captured by the mesh is $2D$; its associated isotropic fluctuation velocity is $\sqrt{(2/3)k}$. Therefore the maximum frequency f_{MC} reasonably resolved by the local grid spacing D is:

$$f_{MC} = \frac{\sqrt{\frac{2}{3}k}}{2\Delta} \tag{3936}$$

k represents the mean turbulent kinetic energy that is contained by the mean turbulent structures. Most widely used RANS turbulence models solve transport equations for k directly. The transport-equation for k is derived directly from the unsteady Navier-Stokes equations, and therefore it contains contributions from all the important mechanisms: convection, diffusion, production, and dissipation. So long as the mean flow features are captured, the k solution tends to be reasonably grid-independent.

This measure is used in preference to one derived from the turbulence time scale, that is, the ratio of kinetic energy to its dissipation rate k/ϵ. This is better representative of the dissipative scales which, in frequency and energy content, are orders of magnitude higher than the human hearing range or peak sensitivity.

Since this measure is derived from a steady-state solution, some limitations are inherent. The frequencies that are associated with time varying large-scale motions such as vortex-shedding, which convect through the mesh, are not accounted for. Instead, its usefulness is to approximate the frequencies of the turbulence scales which are modeled in RANS, and which become resolved in Large Eddy Simulation. In other words, this measure is more valid for the broadband and less so for narrowband excitations.

The mesh frequency cutoff measure has been applied to typical industrial applications at low Mach numbers flow.

Acoustic Wave Model

Ewer and Schroder presented a family of Acoustic Perturbation Equations (APE) for the simulation of flow-induced acoustic fields. These equations describe noise source terms that are determined by unsteady simulations of compressible or incompressible problems. As Lighthill derived an acoustic analogy from the basic equations of Navier-Stokes, an acoustic wave equation can be derived starting from the acoustic perturbation equations as well.

This model calculates the acoustic pressure history on the whole domain, including the effects of convection and refraction.

See [729] and [743].

The complete system of the acoustic perturbation equations in terms of the perturbation variables $(p', \mathbf{u}^a)$ is given as

$$\frac{\partial p'}{\partial t} + c^2\,\nabla\cdot\left(\overline{\rho}\mathbf{u}^a + \overline{\mathbf{v}}\frac{p'}{c^2}\right) \approx 0 \tag{3937}$$

$$\frac{\partial \mathbf{u}^a}{\partial t} + \nabla(\overline{\mathbf{v}}\cdot\mathbf{u}^a) + \nabla\left(\frac{p'}{\overline{\rho}}\right) \approx \nabla\Phi_p \tag{3938}$$

where:

- p' is perturbation pressure
- $\mathbf{u}^a$ is irrotational perturbation velocity
- $\overline{\rho}$ is time-averaged density
- $\overline{\mathbf{v}}$ is time-averaged (mean) velocity
- c is speed of sound
- Φ_p is noise source function

Using the relationship $p' = \overline{\rho}\Phi_p + p^a$ (where p^a is acoustic pressure) from Ewert and Schroder and taking the substantial derivative $\left(\frac{\partial}{\partial t} + \overline{v}\cdot\nabla\right)$ of Eqn. (3937) and the divergence of Eqn. (3938), one can obtain two equations in terms of p^a and u^a as dependent variables. Then with incompressible flow assumption, that is, $\nabla\cdot\overline{v} = 0$ and combining the two equations, a single equation is obtained in terms of acoustic pressure and noise sources from incompressible flow:

$$\frac{1}{c^2}\frac{\partial^2 p^a}{\partial t^2} + \frac{2\overline{\mathbf{v}}}{c^2}\cdot\nabla\frac{\partial p^a}{\partial t} + \frac{(\overline{\mathbf{v}}\cdot\nabla)}{c^2}(\nabla\cdot\overline{\mathbf{v}}p^a) - \nabla^2 p^a = -\left(\frac{\overline{\rho}}{c^2}\frac{\partial^2\Phi_p}{\partial t^2} + \frac{2(\overline{\mathbf{v}}\cdot\nabla)}{c^2}\overline{\rho}\frac{\partial\Phi_p}{\partial t} + \frac{(\overline{\mathbf{v}}\cdot\nabla)}{c^2}\left(\nabla\cdot\overline{\mathbf{v}}\Phi_p\right)\right) \tag{3939}$$

Convected Acoustic Wave Equation

An equation for a wave caused mainly by convection effects is derived from Eqn. (3939) by assuming that $\overline{\rho}\frac{\partial \Phi_p}{\partial t} \approx \frac{\partial P'}{\partial t}$ and $\nabla \Phi_p \approx \frac{1}{\overline{\rho}} \nabla P'$ for noise sources from incompressible flows [729]:

$$\frac{1}{c^2}\frac{\partial^2 p^a}{\partial t^2} + \frac{2\overline{v}}{c^2} \cdot \nabla \frac{\partial p^a}{\partial t} + \frac{(\overline{v} \cdot \nabla)}{c^2}(\nabla \cdot \overline{v} p^a) - \nabla^2 p^a = -\left(\frac{1}{c^2}\frac{\partial^2 P'}{\partial t^2} + \frac{2(\overline{v} \cdot \nabla)}{c^2}\frac{\partial P'}{\partial t} + \frac{(\overline{v} \cdot \nabla)}{c^2}(\nabla \cdot \overline{v} P')\right) \tag{3940}$$

Acoustic Wave Equation without Convection

In absence of mean flow, Eqn. (3940) simply reduces to

$$\frac{1}{c^2}\frac{\partial^2 p^a}{\partial t^2} - \nabla^2 p^a = -\frac{1}{c^2}\frac{\partial^2}{\partial t^2} P' \tag{3941}$$

This equation resembles Lighthill's equation except that this one is based on noise sources coming from acoustic perturbation equations for incompressible flow.

Physical Damping of Acoustic Waves

The acoustic wave equations that are presented in Eqn. (3940) and Eqn. (3941) do not include any mechanism of damping spurious waves originating from mesh coarsening transitions. From the concept of wave propagation in viscous medium, a physical damping term can be introduced into the equation as

$$\frac{1}{c^2}\frac{\partial^2 p^a}{\partial t^2} - \nabla^2\left(p^a + \tau \frac{\partial p^a}{\partial t}\right) = -\frac{1}{c^2}\frac{\partial^2 P'}{\partial t^2} \tag{3942}$$

where $\tau = \chi \frac{\Delta t}{\pi \lambda}$, with

- $\chi =$ damping coefficient (= 0 for no damping and = 1 for maximum damping)
- $\Delta t =$ time-step
- $\lambda = \frac{c \Delta t}{\Delta x} =$ local CFL number (where $\Delta x =$ grid spacing)

Acoustic waves are damped in poorly resolved zones (coarse mesh) using this damping coefficient to suppress undesired oscillations that pollute predictions inside the region of interest.

Low-Pass Time Filters

The sources of noise in the acoustic wave model come mainly from the second derivative of the flow pressure fluctuations with respect to time. Due to mesh size transitions, these sources include high-frequency spurious numerical signals which, in turn, generate high-frequency noise. A suitable low-pass filter suppresses this undesired noise.

For the following filters, the higher the order of the filter, the more signal damping it provides, while also introducing more phase error (that is, the shift in the angle of the wave from that which is expected). You can consider phase error as a measure of the distortion in the signal that is introduced by phase differences for different frequencies.

The Acoustic Wave model applies the low-pass time filter directly to the APE sources. The filered APE source is:

$$-\overline{\frac{1}{c^2}\frac{\partial^2}{\partial t^2}\mathrm{P}'} = \frac{1}{c^2}\frac{\partial^2 p^a}{\partial t^2} - \nabla\cdot\nabla\left(p^a + \tau\frac{\partial p^a}{\partial t}\right)$$

First-Order IIR Low-Pass Time Filter

The filtered form of $\frac{d^2P'}{dt^2}$ at time-step i is:

$$\overline{\left(\frac{d^2P'}{dt^2}\right)_i} = \alpha\left(\frac{d^2P'}{dt^2}\right)_i + (1-\alpha)\overline{\left(\frac{d^2P'}{dt^2}\right)_{i-1}} \tag{3943}$$

where:

$$\alpha = \frac{2\pi\Delta t f_{cut}}{2\pi\Delta t f_{cut} + 1}$$

so that the filtered value $\overline{\left(\frac{d^2P'}{dt^2}\right)}$ at a given time-step depends on the filtered value at the previous time-step.

Second-Order Butterworth Low-Pass Time Filter

The filtered form of $\frac{d^2P'}{dt^2}$ at time-step i is:

$$\overline{\left(\frac{d^2P'}{dt^2}\right)_i} = \left[\frac{n_0}{d_0}\left(\frac{d^2P'}{dt^2}\right)_i + \frac{n_1}{d_1}\left(\frac{d^2P'}{dt^2}\right)_{i-1} + \frac{n_2}{d_2}\left(\frac{d^2P'}{dt^2}\right)_{i-2}\right] + \left[-\frac{d_1}{d_0}\overline{\left(\frac{d^2P'}{dt^2}\right)_{i-1}} - \frac{d_2}{d_0}\overline{\left(\frac{d^2P'}{dt^2}\right)_{i-2}}\right] \tag{3944}$$

where:

- $n_0 = 1$, $n_1 = 2$, and
- $d_0 = c^2 + \sqrt{2}\ c + 1$
- $d_1 = -2\left(c^2 - 1\right)$
- $d_2 = c^2 - \sqrt{2}\ c + 1$
- $c = \frac{1}{\tan\left(\frac{2\pi\Delta t f_{cut}}{2}\right)}$

Third-Order Butterworth Low-Pass Time Filter

The filtered form of $\frac{d^2P'}{dt^2}$ at time-step i is:

$$\overline{\left(\frac{d^2P'}{dt^2}\right)_i} = \left[\frac{n_0}{d_0}\left(\frac{d^2P'}{dt^2}\right)_i + \frac{n_1}{d_1}\left(\frac{d^2P'}{dt^2}\right)_{i-1} + \frac{n_2}{d_2}\left(\frac{d^2P'}{dt^2}\right)_{i-2} + \frac{n_3}{d_3}\left(\frac{d^2P'}{dt^2}\right)_{i-3}\right] + \left[-\frac{d_1}{d_0}\overline{\left(\frac{d^2P'}{dt^2}\right)_{i-1}} - \frac{d_2}{d_0}\overline{\left(\frac{d^2P'}{dt^2}\right)_{i-2}} - \frac{d_3}{d_0}\overline{\left(\frac{d^2P'}{dt^2}\right)_{i-3}}\right] \tag{3945}$$

where:

- $n_0 = 1$, $n_1 = 3$, $n_2 = 3$, and
- $d_0 = c^3 + 2c^2 + 2c + 1$
- $d_1 = -3c^3 - 2c^2 + 2c + 3$
- $d_2 = 3c^3 - 2c^2 - 2c + 3$
- $d_3 = -c^3 + 2c^2 - 2c + 1$

Low-Pass Space Filter

The low-pass space filter also reduces numerical artifacts from the sources of noise by applying the filter in the spatial mode.

The narrow-band, low-pass, space-filtered value $\widetilde{P}_c$ of a pressure fluctuation P_c in cell c is computed as:

$$\widetilde{P}_c = P'_c + \frac{1}{4V_c^{1/3}} \sum_{nb \in NB[c]} \left(P'_{nb} - P'_c\right) \frac{\mathbf{A}_f \cdot \mathbf{A}_f}{\mathbf{A}_f \cdot \mathbf{ds}} \tag{3946}$$

where:

- P'_c is the unfiltered pressure fluctuation $P'_c = P - P_{mean}$.
- V_c is the volume of the cell.
- The subscript nb indicates a neighbor cell.
- $\mathbf{A}_f$ is the area vector for cell c, pointing outward toward neighbor cell nb.
- $\mathbf{ds}$ is the distance vector, $\mathbf{ds} = \mathbf{x}_{nb} - \mathbf{x}_c$, and $\mathbf{x}_c$ and $\mathbf{x}_{nb}$ are the distance vectors for the current and neighboring cells.

The wave equation solver uses the space-filtered pressure $\widetilde{P}$ to compute smoother APE sources:

$$\frac{1}{c^2}\frac{\partial^2 p^a}{\partial t^2} - \nabla \cdot \nabla\left(p^a + \tau\frac{\partial p^a}{\partial t}\right) = -\frac{1}{c^2}\frac{\partial^2}{\partial t^2}\widetilde{\mathrm{P}'}$$

Non-Reflective Boundary Condition

The non-reflective boundary condition eliminates the spurious noise at the domain boundaries, when the Acoustic Wave model is activated. It applies to incompressible simulations, unlike free-stream boundaries and the Acoustic Suppression Zone model.

The acoustic pressure is related to the normal component of the acoustic velocity:

$$p^a = \overline{\rho} c \mathbf{u}^a \cdot \mathbf{n} \tag{3947}$$

By taking the partial derivative with respect to time and using a linearized momentum equation:

$$\begin{aligned} \frac{\partial p^a}{\partial t} &= \overline{\rho} c \frac{\partial u^a}{\partial t} \cdot n \\ \overline{\rho}\frac{\partial u^a}{\partial t} &= -\nabla p^a \end{aligned} \tag{3948}$$

one can then obtain an expression of a non-reflecting boundary condition as:

$$\frac{\partial p^a}{\partial t} = -c\nabla p^a \cdot n \tag{3949}$$

Standard Newmark Method

The standard Newmark method uses the following expressions for time derivatives of acoustic pressure:

$$\begin{aligned}\left(\frac{\partial p}{\partial t}\right)^{n+1} &= \left(1-\frac{\delta}{\beta}\right)\left(\frac{\partial p}{\partial t}\right)^{n} + \left(1-\frac{\delta}{2\beta}\right)\Delta t\left(\frac{\partial^2 p}{\partial t^2}\right)^{n} + \frac{\delta}{\Delta t\beta}\left(p^{n+1}-p^{n}\right)\\ \left(\frac{\partial^2 p}{\partial t^2}\right)^{n+1} &= \left(1-\frac{\delta}{2\beta}\right)\left(\frac{\partial^2 p}{\partial t^2}\right)^{n} - \frac{\delta}{\Delta t\beta}\left(\frac{\partial p}{\partial t}\right)^{n} + \frac{\delta}{\Delta t^2\beta}\left(p^{n+1}-p\right)\end{aligned} \tag{3950}$$

where:

- $\delta \geq 0.5$
- $\beta = \frac{1}{4}\left(\delta + \frac{1}{2}\right)^2$
- n is the previous time-step.
- $n+1$ is the time-step for which the solution is being computed.

The default value of δ is 0.5, which produces no damping. Raising the value of δ slightly, to 0.502 for example, is usually sufficient to damp low frequencies.

Newmark Alpha Method

The Newmark Alpha method starts with the following expression for time derivatives of acoustic pressure:

$$M\left(\frac{\partial^2 p}{\partial t^2}\right)^{n+1} + C\left(\frac{\partial p}{\partial t}\right)^{n+1} + Kp^{n+1} = F^{n+1}$$

where M, C, and K are operators:

- $M = \frac{1}{c^2}\int_V dV$ is the operator for mass.
- $C = \int_s \tau\, \nabla\cdot\, \nabla\, ds$ is the operator for damping.
- $K = \int_s \nabla\cdot\, \nabla\, ds$ is the operator for stiffness.
- $F = -\frac{1}{c^2}\int_V \frac{\partial^2 P'}{\partial t^2} dV$ is the external force.

The method then discretizes the above equation and introduces the parameter α to obtain the following modified governing equation:

$$M\left(\frac{\partial^2 p}{\partial t^2}\right)^{n+1} + (1+\alpha)C\left(\frac{\partial p}{\partial t}\right)^{n+1} - \alpha C\left(\frac{\partial p}{\partial t}\right)^{n} + (1+\alpha)Kp^{n+1} - \alpha Kp^{n} = (1+\alpha)F^{n+1} - \alpha F^{n} \tag{3951}$$

The operators and the external force take the form of matrices, where:

- M is the mass matrix

- C is the damping matrix
- K is the stiffness matrix
- F is the external force matrix

α is a weighting parameter, $-0.33 \leq \alpha \leq 0$.

Use non-zero values of α to damp spurious high frequencies. STAR-CCM+ uses a default value of -0.05.

α is related to the Standard Newmark parameters as follows:

$\delta = \frac{1}{2} - \alpha$, $\beta = \frac{1}{4}(1-\alpha)^2$

APE Fields for FW-H Post-Processing

The Acoustic Wave model can be combined with the Ffowcs Williams-Hawkings model to calculate the following fields for far-field cases:

- APE Total Density $\Sigma\rho$:

$$\Sigma\rho = \frac{P_a}{c^2} + \rho_f \tag{3952}$$

- APE Total Velocity Σv:

$$\Sigma V = V_a + V_f \tag{3953}$$

- APE Total Pressure ΣP:

$$\Sigma P = P_a + P_f \tag{3954}$$

where:

- c is the speed of sound.
- ρ_f is the density of the incompressible flow.
- P_a is the acoustic pressure.
- P_f is the pressure of the incompressible flow.
- v_f is the velocity of the incompressible flow.
- v_a is the acoustic velocity.

v_a at the nth time-step is computed from the acoustic velocities of the previous time-step and the acoustic pressure gradient. This is obtained from second-order backward differencing of the momentum equation of the Acoustic Perturbation Equations.

$$v_a^n = \frac{1}{3}\left[4v_a^{n-1} - v_a^{n-2} + \frac{2\Delta t \nabla P_a}{\rho_f}\right] \tag{3955}$$

Acoustic Suppression Zone Model

The Acoustic Suppression Zone model (also known as Sponge Zone or Buffer Zone) is considered a practical method to suppress spurious reflective disturbances in unsteady aeroacoustic simulations at the exterior boundaries. It only applies to compressible flows, but can be used in any aeroacoustic calculation.

Running transient flow simulations produces time-dependent variations in the flow. These flow disturbances convect with the flow (for vortical disturbances) or propagate at the speed of sound (for acoustic disturbances) outward to exterior boundaries.

If the domain boundaries have reflective properties, these disturbances can be spuriously reflected back into the domain. This reflection affects the flow solution, causing problems for acoustic-type calculations.

To satisfy the requirements for good quality aeroacoustic calculations, the numerical algorithm must allow these pressure waves and vortices to travel through the boundary smoothly. Or the numerical algorithm must give the boundary quasi-non-reflective properties.

For example, a pressure outlet with non-reflecting properties allows the defined constant pressure to represent an average pressure at the outlet boundary. At the same time, it allows the flow pressure at the boundary to vary and disturbances to pass out of the domain without reflecting off the boundary.

A practical approach to constructing a non-reflective boundary treatment is to complement the respective characteristics-based boundary conditions (correct imposed flow boundary conditions values) with an Acoustics Suppression Zone model.

The Acoustic Suppression Zone model adds appropriate damping term(s) to the right-hand side of the governing equations for both coupled and segregated flow, of the form:

$$-\sigma\left(Q-Q_{ref}\right) \tag{3956}$$

where:

- σ is the relaxation coefficient.

  When the boundary distance is less than or equal to the zone thickness, the specific variation is a smooth transition from 0 to the **Maximum Relaxation Coefficient** at the boundary:

  $$\sigma = \frac{\sigma_{max}}{2}\left(1+\cos\pi\frac{d}{w}\right)$$

  where:

  - σ_{max} is the maximum relaxation coefficient. It is the inverse of the relaxation time and is measured in sec^{-1}. The value must be greater than zero, and values of several hundred are typical. The default value is zero, meaning no attenuation.
  - d is the distance from the centroid of the cell in question to the boundary using Acoustic Suppression.
  - w is the width of the Acoustic Suppression Zone.
- $Q = \{\rho, \rho v, \rho E\}$.

  where:

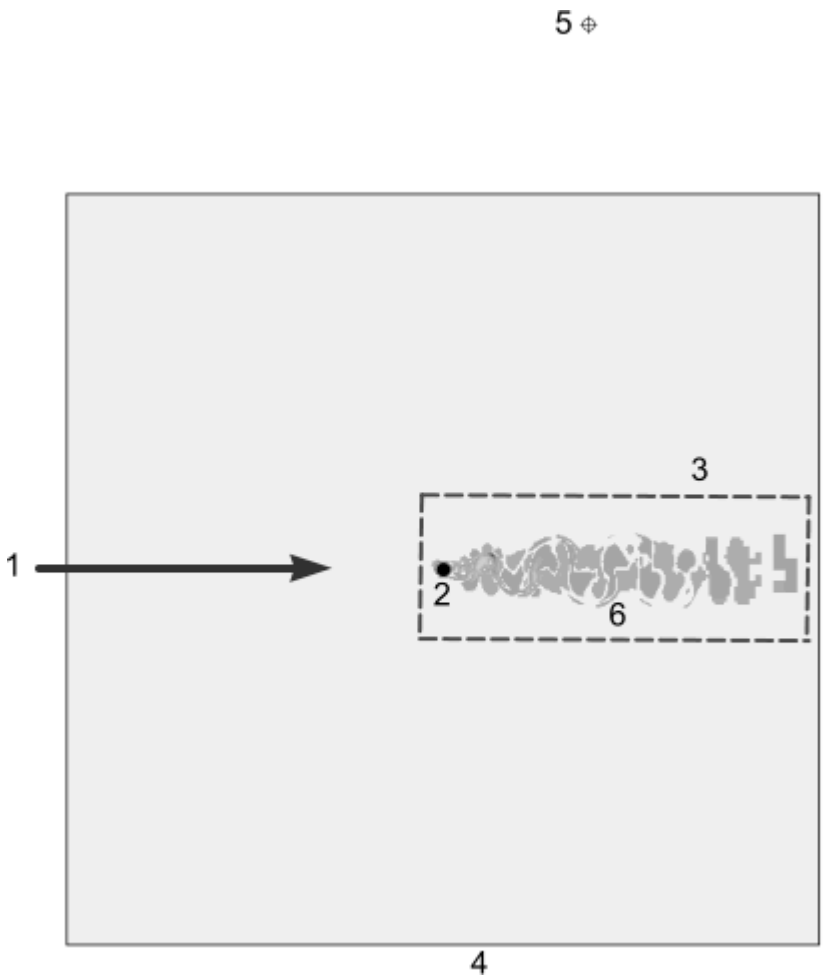
  - v is the flow velocity vector.

- E is the total energy, $E = C_v T + 0.5|v|^2$, where C_v is the heat capacity at constant volume.

These additional terms damp flow disturbances (computed for the mean or target flow conditions), before they reach the boundary. This Acoustic Suppression Zone treatment helps characteristics-based boundary conditions to produce a quasi non-reflective response, that is, to allow disturbances to travel through the boundary without producing spurious back-reflections.

Ffowcs Williams-Hawkings Model

The Ffowcs Williams-Hawkings (FW-H) acoustics integral formulation is the preferred strategy for far-field noise prediction such as overhead aircraft noise. This model calculates the far-field sound signal that is radiated from near-field flow data from a CFD solution. The goal is to predict small amplitude acoustic pressure fluctuations at the locations of each receiver.



Index

1. Flow direction
2. Impermeable surface that is located on the sources of sound body surface (e.g. cylinder surface)
3. Permeable surface enclosing the sound sources
4. Computational domain boundary / Limit of Direct Noise Simulation near-field solution
5. Receivers in the far field
6. Quadrupoles noise visualization

The FW-H acoustic model for stationary motion can simulate wind tunnel aeroacoustic measurements with stationary sources such as aircraft landing gear or an airfoil trailing edge. The FW-H acoustic model is used only to predict the propagation of sound in free space. It does not include effects such as sound reflections, refraction, or material property change.

The Ffowcs Williams-Hawkings formulations are based on Farassat's Formulation 1A ([722], [730]) and Dunn Farassat Padula 1A ([739]), both for the non-convective form of FW-H. Therefore, numerical accuracy is not an issue of great concern. With the FW-H approach, a second order discretization scheme, both spatial and temporal, together with judicious choices for mesh resolution and time-step size, can give accurate far-field noise prediction.

The Ffowcs Williams-Hawkings Model

Ffowcs Williams and Hawkings [734] extended the work of Lighthill [743] and Curle [726] to the formulation of aerodynamic sound generated by a surface in arbitrary motion.

Similar as Lighthill equation, they recast the equations of fluid motion in the form of an inhomogeneous wave equation. An important aspect in defining the FW-H equation was the casting of certain flow variables as generalized functions.

The result obtained by Ffowcs Williams and Hawkings (see [734]) was formally presented for an impermeable surface, but they understood that mathematical surface in their formulation does not have to coincide with the solid body.

Both impermeable and permeable surfaces can be used for Ffowcs Williams-Hawkings acoustic analogy model where the formulation is given below.

Ffowcs Williams-Hawkings Formulation

The Ffowcs Williams-Hawkings (FW-H) formulation is based on Farassat's Formulation 1A and the Dunn Farassat Padula Formulation 1A.

- Farassat's Formulation 1A (see [721]) is the default and preferred formulation, a non-convective form of FW-H for general subsonic source regions, including the Impermeable formulation for transient rotating motion, in both whole and periodic domains.
- Dunn Farassat Padula 1A (see [732]) is retained for backwards compatibility, a non-convective form of FW-H, for subsonic rotating sections representing the Impermeable formulation for transient rotating motion, whole and periodic domains.

The FW-H equation is an exact rearrangement of the continuity and the momentum equations into the form of an inhomogeneous wave equation. The FW-H equation gives accurate results even if the surface of integration lies in the nonlinear flow region. It is based on the free-space Green's function to compute the sound pressure at the observer location, x. The FW-H equation for pressure that is radiated into a medium at rest by a flow in a region or a set of surfaces is:

$$\mathrm{p}'(\mathbf{x},t) = \mathrm{p}'_T(\mathbf{x},t) + \mathrm{p}'_L(\mathbf{x},t) + \mathrm{p}'_Q(\mathbf{x},t) \tag{3957}$$

The monopole term is:

$$\mathrm{p}'_T(\mathbf{x},t) = \frac{1}{4\pi}\left(\left(\frac{\partial}{\partial t}\right)\int_S \left[\frac{Q}{(r(1-M_r))}\right]_{ret} \mathrm{d}\,S\right) \tag{3958}$$

The dipole term is:

$$\mathrm{p'}_L(\mathbf{x},t) = \frac{1}{4\pi}\left(\left(-\frac{\partial}{\partial x_i}\right)\int_S \left[\frac{L_i}{(r(1-M_r))}\right]_{ret} \mathrm{d}\,S\right) \tag{3959}$$

The quadrupole term is:

$$\mathrm{p'}_Q(\mathbf{x},t) = \frac{1}{4\pi}\left(\left(\frac{\partial^2}{(\partial x_i)(\partial x_j)}\right)\int_V \left[\frac{T_{ij}}{(r(1-M_r))}\right]_{ret} \mathrm{d}\,V\right) \tag{3960}$$

with:

$$Q = \rho_0 U_i n_i \tag{3961}$$

$$U_i = \left(1-\frac{\rho}{\rho_0}\right)v_i + \frac{\rho u_i}{\rho_0} \tag{3962}$$

$$L_i = P_{ij}n_i + \rho u_i(u_n - v_n) \tag{3963}$$

$$P_{ij} = (p-p_0)\delta_{ij} - \sigma_{ij} \tag{3964}$$

$$T_{ij} = \rho u_i u_j + \delta_{ij}\left[(p-p_0) - {c_0}^2(\rho-\rho_0)\right] - \sigma_{ij} \tag{3965}$$

where:

- u_i represents fluid velocity component in the i direction.
- u_n is the fluid velocity component normal to the surface.
- v_i represents surface velocity component in the i direction.
- v_n is the surface velocity component normal to the surface.
- n_i is the surface normal vector.
- σ_{ij} is the viscous stress tensor.
- ρ_0 is the far field density.
- P_{ij} is the compressive stress tensor.
- T_{ij} is the Lighthill stress tensor.

The subscript ret indicates retarded time, which is the time of emission.

The space derivatives from Eqn. (3958) and Eqn. (3959) are transformed into time derivatives. Afterwards, the time derivatives at the observer locations are moved into the integrals.

Eqn. (3967) describes the monopole source term using the advanced-time formulation. Eqn. (3968) describes the dipole source term using the advanced-time formulation.

FW-H Surface Terms

When the integration surface coincides with the surface of the solid body, the monopole term Eqn. (3958), the dipole term Eqn. (3959), and the quadrupole term Eqn. (3960) are called:

- The *Thickness Surface Term*, $\mathrm{p}'_T(\mathbf{x},t)$, resulting from the displacement of fluid as the body passes. The term is defined in Eqn. (3967) for general flows and in Eqn. (3969) for flows with rigid body motion or moving reference frames.
- The *Loading Surface Term*, $\mathrm{p}'_L(\mathbf{x},t)$, resulting from transient variation of the force distribution on the body surface. The term is defined in Eqn. (3968) for general flows and in Eqn. (3970) for flows with rigid body motion or moving reference frames.
- The *Total Surface Term*, $\mathrm{p}'_S(\mathbf{x},t)$, resulting from the sum of the *Thickness Surface Term* and the *Loading Surface Term*. The term is defined as:

$$\mathrm{p}'_S(\mathbf{x},t) = \mathrm{p}'_T(\mathbf{x},t) + \mathrm{p}'_L(\mathbf{x},t) \tag{3966}$$

Farrassat's Formulation 1A

This formulation is for general subsonic source regions, for general far-field noise prediction. See [721] and [722].

$$\mathrm{p}'_T(\mathbf{x},t) = \frac{1}{4\pi}\left(\int_{(f=0)}\left[\frac{\rho_0(\dot{U}_n + U_{\overline{n}})}{r(1-M_r)^2}\right]_{ret} \mathrm{d}\,S + \int_{(f=0)}\left[\frac{\rho_0 U_n\left[r\dot{M}_r + a_0(M_r - M^2)\right]}{r^2(1-M_r)^3}\right]_{ret} \mathrm{d}\,S\right) \tag{3967}$$

$$\begin{aligned}\mathrm{p}'_L(\mathbf{x},t) = \frac{1}{4\pi}\Bigg(&\frac{1}{a_0}\cdot\int_{(f=0)}\left[\frac{\dot{L}_r}{r(1-M_r)^2}\right]_{ret} \mathrm{d}\,S \\ &+ \int_{(f=0)}\left[\frac{(L_r - L_M)}{r^2(1-M_r)^2}\right]_{ret} \mathrm{d}\,S + \frac{1}{a_0}\cdot\int_{(f=0)}\left[\frac{L_r\left[r\dot{M}_r + a_0(M_r - M^2)\right]}{r^2(1-M_r)^3}\right]_{ret} \mathrm{d}\,S\Bigg)\end{aligned} \tag{3968}$$

where:

- $L_i = P_{ij}\cdot n_i + \rho\cdot u_i(u_n - v_n)$ where $P_{ij} = (p - p_0)\delta_{ij}$
- $M_i = v_i/a_0$
- $r = x_{\mathrm{observer}} - y_{\mathrm{face}}$

and where:

- $f = 0$ denotes a mathematical surface to embed the exterior flow problem $f > 0$ in an unbounded space.
- $f = 0$ represents the emission surface and is made coincident with a body, impermeable surface, or permeable surface.

If the data surface coincides with a solid surface, then the normal velocity of the fluid is the same as the normal velocity of the surface: $u_n = v_n$

In this case, Eqn. (3967) and Eqn. (3968) correspond to the Impermeable FW-H Surface type and some of the terms are eliminated.

Dunn-Farrassat-Padula Formulation 1A

This formulation is based on modification of Farassat's Formulation 1A, which was used in both ANOPP (Aircraft Noise Prediction Program) and DFP-ATP (Dunn-Farassat-Padula Advanced Turboprop Prediction) for subsonic rotating blades to compute the acoustic pressure, based on Hubbard [739].

For the Permeable FW-H Surface type, the FW-H Surface integrals are computed on the internal interface boundary, from an in-place interface.

$$\mathrm{p}'_T(\mathbf{x},t) = \frac{1}{4\pi}\left(\int_{(f\,=\,0)}\left[\frac{\rho_0 v_n\left[\left(r\dot{M}_r\right)+a_0\left(M_r-M^2\right)\right]}{r^2\left(1-M_r\right)^3}\right]\mathrm{d}\,S\right)_{ret} \tag{3969}$$

$$\mathrm{p}'_L(\mathbf{x},t) \tag{3970}$$

$$= \frac{1}{4\pi}\left(\frac{1}{a_0}\cdot\int_{(f\,=\,0)}\left[\frac{\dot{L}_i r_i}{r\left(1-M_r\right)^2}\right]\mathrm{d}\,S + \int_{(f\,=\,0)}\left[\frac{\left(L_r-L_i M_i\right)}{r^2\left(1-M_r\right)^2}\right]\mathrm{d}\,S + \frac{1}{a_0}\cdot\int_{(f\,=\,0)}\left[\frac{L_r\left[\left(r\dot{M}_r\right)+a_0\left(M_r-M^2\right)\right]}{r^2\left(1-M_r\right)^3}\right]\mathrm{d}\,S\right)_{ret}$$

and:

$$\dot{M}_r = \dot{M}_i r_i \tag{3971}$$

where:

- L_i is the blade load vector. $L_i = P_{ij}\cdot n_i + \rho\cdot u_i\left(u_n - v_n\right)$ where $P_{ij} = \left(p - p_0\right)\delta_{ij}$
- $\dot{L}_i$ is the derivative of the blade load vector with respect to source time.
- v_i is the local velocity of the blade surface with respect to the quiescent fluid.
- v_n is the surface normal velocity.
- r_i is the distance from a source point to the observer.
- M_r is the Mach number of the source toward the observer.

The subscript ret indicates retarded time, which is the time of emission.

Accounting for Time-Lag

The major task in evaluating the FW-H integrals is how to account for the time-lag between emission and reception times.

The advanced time algorithm looks forward in time to see when the observer perceives the currently generated sound waves:

- The procedure starts with a sequence of emission times (conveniently taken as the flow times).
- The source strengths are calculated (thickness surface noise and loading surface noise) at all source elements (faces of the integration surfaces) for a given emission time.
- The contribution of the sources is interpolated in the far-field time domain to build the sound signal.

The total sound pressure the observer perceives consists of the contribution from all source elements. The sound pressure at the receiver is obtained by accumulating the arriving signals in time slots. The overall observer acoustic signal is found from the summation of the acoustic signal from each source element of the FW-H surface during the same source time.

FW-H Volume Terms

The quadrupole noise is a volume distribution of sources, which accounts for nonlinearities in the flow.

These nonlinearities are of two types [744]. First, the local speed of sound is not constant, but varies due to particle acceleration. Second, the finite particle velocity near the body (for example blade) influences the

velocity of sound propagation. When strong shear layers exist in the flow or when the Mach number increases, the quadrupole term is not negligible.

Farassat and Brentner [721] have shown that the noise contribution from the quadrupole, $\mathrm{p}'_Q(\mathbf{x},t)$, can be expressed as a "collapsing-sphere" formulation. Using this formulation, the space derivatives from Eqn. (3960) are transformed into time derivatives:

$$\mathrm{p}'_Q(\mathbf{x},t) = \frac{1}{4\pi}\left(\left(\frac{1}{c}\right)\left(\frac{\partial^2}{\partial t^2}\right)\int_{-\infty}^{t}\left[\int_{(f>0)}\frac{T_{rr}}{r}\,\mathrm{d}\,\Omega\right]\mathrm{d}\,\tau + \left(\frac{\partial}{\partial t}\right)\int_{-\infty}^{t}\left[\int_{(f>0)}\frac{3T_{rr}-T_{ii}}{r^2}\,\mathrm{d}\,\Omega\right]\mathrm{d}\,\tau\right) + \left(c\int_{-\infty}^{t}\left[\int_{(f>0)}\frac{3T_{rr}-T_{ii}}{r^3}\,\mathrm{d}\,\Omega\right]\mathrm{d}\,\tau\right) \tag{3972}$$

where:

- T_{rr} is the double contraction of T_{ij}.
- r_i and r_i are the components of the unit vector in the direction of radiation.
- T_{ij} is the Lighthill stress tensor. $T_{ij} = \rho u_i u_j + \delta_{ij}\left[(p-p_0)-{c_0}^2(\rho-\rho_0)\right] - \sigma_{ij}$ and σ_{ij} is the viscous stress tensor.

Eqn. (3972) is transformed from a collapsing-sphere formulation to an advanced time formulation using the following four equations. In these equations, the time derivatives at the observer are moved into the integrals to prevent numerical time differentiation of the integrals. The "source-time-dominant" algorithm from [723] is used to allow the estimation of the $\mathrm{p}'_Q(\mathbf{x},t)$ volume term of the FW-H equation as follows:

$$\mathrm{p}'_Q(\mathbf{x},t) = \frac{1}{4\pi}\left(\int_{(f>0)}\left[\frac{K_1}{c^2 r} + \frac{K_2}{c r^2} + \frac{K_3}{r^3}\right]_{ret}\mathrm{d}\,V\right) \tag{3973}$$

$$K_1 = K_{11} + K_{12} + K_{13} = \left[\frac{\ddot{T}_{rr}}{(1-M_r)^3}\right] + \left[\frac{\ddot{M}_r T_{rr} + 3\dot{M}_r \dot{T}}{(1-M_r)^4}\right] + \left[\frac{3\dot{M}_r^2 T_{rr}}{(1-M_r)^5}\right] \tag{3974}$$

$$K_2 = K_{21} + K_{22} + K_{23} + K_{24} = \left[\frac{-\dot{T}_{ii}}{(1-M_r)^2}\right] + \left[\frac{4\dot{T}_{M_r} + 2T_{\dot{M}_r} + \dot{M}_r T_{ii}}{(1-M_r)^3}\right] + \left[\frac{3\left[\left(1-M^2\right)\dot{T}_{rr} - 2\dot{M}_r T_{Mr} - M_i \dot{M}_i T_{rr}\right]}{(1-M_r)^4}\right] + \left[\frac{6\dot{M}_r\left(1-M^2\right)T_{rr}}{(1-M_r)^5}\right] \tag{3975}$$

$$K_3 = K_{31} + K_{32} + K_{33} = \left[\frac{2T_{MM} - \left(1-M^2\right)T_{ii}}{(1-M_r)^3}\right] - \left[\frac{6\left(1-M^2\right)T_{Mr}}{(1-M_r)^4}\right] + \left[\frac{3\left(1-M^2\right)^2 T_{rr}}{(1-M_r)^5}\right] \tag{3976}$$

where:

$T_{rr} = T_{ij} r_i r_j$	$T_{MM} = T_{ij} M_i M_j$	$T_{Mr} = T_{ij} M_i r_j$	$T_{\dot{M}r} = T_{ij} \dot{M}_i r_j$
$\dot{T}_{Mr} = \dot{T}_{ij} M_i r_j$	$\dot{T}_{rr} = \dot{T}_{ij} r_i r_j$	$\ddot{T}_{rr} = \ddot{T}_{ij} r_i r_j$	$M_i = \frac{v_i}{c}$
$M_r = M_i r_i$	$\dot{M}_r = \dot{M} r_i$	$\ddot{M}_r = \ddot{M}_i r_i$	$\ddot{M}_{rr} = \ddot{M} r_i r_i$

$r_i = \frac{x_i - y_i}{r}$

where:

- r_i denotes the unit vector in the direction of radiation.
- A dot above a variable denotes the derivative with respect to source time of that variable.

Convective Effects

The Farassat's Formulation 1A and the Dunn Farassat Padula Formulation 1A exclude convective effects. To include these effects, the Ffowcs Williams-Hawkings model uses the effective speed of sound $c_\theta(\theta)$, the propagation speed of the pressure waves under the convective velocity effect:

$$c_\theta(\theta) = \frac{c_0\left(1 - M^2\right)}{\sqrt{1 - M^2 \sin^2(\theta)} + M\cos(\theta)} \tag{3977}$$

where θ is the angle between the convective velocity and the direction of propagation and M is the Mach number calculated from the convective velocity and the far-field speed of sound.

The FW-H formulation makes the appropriate corrections for the receiver time by accounting for the different propagation speed of the pressure waves, using the propagated distance from the face centroid of the surface to the observer, based on convective velocity at time of emission.

Aeroacoustics Bibliography

[718] *Ardavan, H. "The breakdown of the linearized theory and the role of quadrupole sources in transonic rotor acoustics," Journal of Fluid Mechanics 226, pp. 591-624, 1991.*

[719] *Bailly, C. and Juve, D. 1999. "A Stochastic Approach to Compute Subsonic Noise using Linearized Euler's Equations", AIAA Journal, AIAA-99-1872.*

[720] *Bechara, W., Lafon, P., Bailly, C., and Candel, S. 1994. "Application of a k-epsilon turbulence model to the prediction for simple and coaxial free jets", Journal Acoustics Soc. Am, 97:6, pp. 3518-3531.*

[721] *Brentner, K. S. and Farassat, F." Analytical Comparison of the Acoustic Analogy and Kirchoff Formulation for Moving Surfaces", AIAA Journal, Vol. 36, No.8, 1998, pp.1379-1386.*

[722] *Brentner, K.S. and Farassat, F. 2003. "Modeling aerodynamically generated sound of helicopter rotors", Progress in Aerospace Sciences, 39:2-3, pp. 83-120.*

[723] *Casalino, D. 2003. "An advanced time approach for acoustic analogy predictions", J Sound and Vibration, 261:4, pp. 583-612.*

[724] *Chandler-Wilde, Simon, and Langdon, Steven, Boundary element methods for acoustics" , Departament of Mathematics, Berkshire UK, July 2007 .*

[725] *Crighton, D. G., Dowling, A. P., Ffowcs Williams, J. E., Heckl, M and Leppington, F.G. "Modern Methods in Analytical Acoustics, Lecture Notes, Springer-Verlag, London, 1992, Chapter 11, Section 10*

[726] *Curle, N, "The influence of Solid Boundaries on Aerodynamic Sound", Proceedings of the Royal Society of London, A231, 1955, pp. 505-514.*

[727] *Dowling, A. P. and Ffowcs Williams, J. E. "Sound and Sources of Sound", Ellis Horword, Chichester, England, UK, 1983, Chapter 9, Section 2.*

[728] *Estorff O. V. Boundary Elements in Acoustics: Advances and Applications, WIT Press, 2000.*

[729] *Ewert, R., and Schroder, W., "Acoustic perturbation equation based on flow decomposition via source filtering," J. of Computational Physics, 188, (2003), pp. 365-398.*

[730] *Farassat, F. 2007. "Derivation of formulations 1 and 1A of Farassat", Technical Memorandum TM-2007-214853, NASA, NASA Langley Research Center (USA).*

[731] *Farassat, F. "Theory of noise generation from moving bodies with an application to helicopter rotors." NASA Technical Report R-451, 1975.*

[732] *Farassat F., Padula S.L. and Dunn M.H. 1987. "Advanced Turboprop Noise Prediction Based on Recent Theoretical Results" J. Sound and Vibration, 119:1 pp. 53-79.*

[733] *Farassat, F., Succi, G. P., "The prediction of helicopter discrete frequency noise," Vertica 7 , vol 4, pp. 309-320, 1983*

[734] *Ffowcs Williams, J. K. E. and Hawkings, D. L."Sound Generation by Turbulence and Surfaces in Arbitrary Motion," Philosophical Transactions of the Royal Society, A264, 1969, pp. 321-342.*

[735] *Frota, J., Lempereur, P. and Roger, M. 1998. "Computation of the noise of a subsonic propeller at an angle of attack", 4th AIAA/CEAS Aeroacoustic Conference, AIAA, Toulouse (France).*

[736] *Gennaro, M. and Caridi, D. 2010. "Ffowcs Williams-Hawkings Acoustic Analogy for Simulation of NASA SR2 Propeller Noise in Transonic Cruise Condition", ECCOMAS CFD 2010.*

[737] *Goldstein, M.E. and Rosenbaum, B. 1973. "Effect of Anisotropic Turbulence on Aerodynamic Noise", J. Acoustical Society of America, 54, pp. 630-645.*

[738] *Hinze, J. O., 1975. Turbulence, McGraw-Hill, pp 668-680.*

[739] *Hubbard, H.H. 1995. "Aeroacoustics of Flight Vehicles, Theory and Practice, Volume 1: Noise Sources" NASA Langley Research Center, Hampton, Virginia, pp. 13.*

[740] *Karweit, M., Blanc-Benon, P., Juve, D., and Comte-Bellot, G. 1991. "Simulation of the Propagation of an Acoustic Wave Through a Turbulent Velocity Field: A Study of Phase Variance", J. Acoustical Society of America, 89:1, pp. 52-62.*

[741] *Kim, J. W., Lau, A. S.H., Sandham, N.D. 2010. "Proposed Boundary Conditions for Gust-Airfoil Interaction Noise", AIAA Journal, 48-11.*

[742] *Kraichnan, R.H. 1970. "Diffusion by a Random Velocity Field", J. Computational Physics, 13:1, pp. 22-31.*

[743] *Lighthill, M. J. "On Sound Generated Aerodynamically. I. General Theory", Proceedings of the Royal Society of London, A211, 1952, pp.564-587.*

[744] *Lighthill, M. 1954. "On Sound Generated Aerodynamically. I. Turbulence as a Source of Sound", Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences (1934-1190), 222:1148, pp. 1-32.*

[745] *Lilley, G.M. 1993. "The radiated noise from isotropic turbulence revisited", NASA Contract Report No. 93-75, NASA Langley Research Center.*

[746] *Lin, P.H. "Computational Aero- and Hydro-acoustics by Space-Time Conservation Element and Solution Element Method", Ph. D. Thesis, Ohio State University, 2011.*

[747] *Mankbadi, R.R. "Computational aero- and hydro-acoustics" ASME United Engineering Center, 1993.*

[748] *Marinus, B. and Roger, M. 2009. "Truncated Method for Propeller Noise Prediction Up to Low Supersonic Helical Tip Mach Numbers", AIAA-2009-3330.*

[749] *Mendonça, F., Read, A., Caro, S., Debatin, K. and Caruelle, B. 2005. "Aeroacoustic Simulation of Double Diaphragm Orifices in an Aircraft Climate Cooling System", AIAA-2005-2976.*

[750] *Proudman, I. 1952. "The Generation of Noise by Isotropic Turbulence", Proc. R. Soc. Lond. 214, pp. 119-132.*

[751] *Ribner, H.S. 1964. "The Generation of Sound by Turbulent Jets", in Advances in Applied Mathematics, Academic Press, New York.*

[752] *Robinson, A.R. and Lee, D. "Oceanography and acoustics: prediction and propagation models", American Institute of Physics, 1994.*

[753] *Sarkar, S., Hussaini, M.Y. 1993. "Computation of the Sound Generated by Isotropic Turbulence", NASA Report 191543, ICASE Report No. 93-74.*

[754] *SenGupta G. "Computation of aeroacoustic scattering effects" AIAA 87-2669, October, 1987.*

[755] *Wagner, Hüttl, Sagaut (2007). "LES for Acoustics", Cambridge University Press, ISBN-13: 9780521871440.*

[756] *Witkowska, A., and Juve, D. 1993. "Numerical Simulation of Noise Generated by Homogeneous and Isotropic Turbulence,' Proc. 13th Colloquium on Aero- and Hydro-Acoustics, Lyon.*

Finite Element Method

The Finite Element method is a powerful tool for finding approximate solutions to continuous problems. The methodology is similar to other numerical techniques that approximate continuous partial differential equations with discrete algebraic equations.

STAR-CCM+ currently uses the Finite Element method in Solid Mechanics, Electromagnetism, and Viscous Flow applications.

Discretization of the Continuous Domain

Mathematical models describing physics phenomena usually consist of a set of partial differential equations, which, together with constitutive equations, form a closed set of equations. The first step for reducing the continuous set to a discrete set of algebraic equations is the discretization of the continuous space domain into a finite number of subdomains, which are called elements. Typical element shapes are triangles, quadrilaterals, tetrahedra, and hexahedra. Tetrahedra and hexahedra can be connected using transition elements like prisms and pyramids.

Weak Form of PDEs

The partial differential equations (PDE) are converted to an integral or weak form that is satisfied over each element domain. Typically, the weak form of a PDE is constructed as follows:

- The PDE is multiplied by an arbitrary test function and integrated over the element domain.
- The order of the derivatives is reduced using integration by parts.
- Appropriate Dirichlet and Neumann boundary conditions are applied.

Discretization of the Unknowns

In each element, the distribution of the dependent variables is constructed from the discrete values that the variables assume at specific locations, for example, at the element nodes or edges. Shape functions interpolate the discrete values to any other point in the element, providing the local (per-element) distribution of the dependent variables. Mapping operations between the local element domain and the global physical domain ensure that the variables are continuous across adjacent elements.

All integrals are approximated with appropriate numerical techniques, leading to the final set of discrete algebraic equations.

Solution of the Linear Algebraic Equations

The system of algebraic equations is solved using a sparse direct solver.

Contents:

Shape Functions
Integration Rules
Sparse Direct Solver
Finite Element Bibliography

Shape Functions

Polynomial shape functions approximate both the element topology and the distribution of the primary unknowns within the element. That is, the position of all points in a finite element and the primary unknowns are both interpolated using polynomial shape functions.

STAR-CCM+ provides various types of shape functions.

Contents:

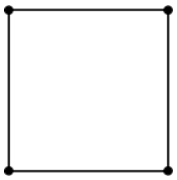
H1 Lagrange Shape Functions
H(Curl) Hierarchical Shape Functions

H^1 Lagrange Shape Functions

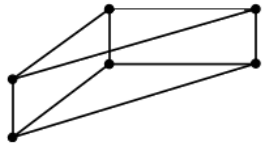
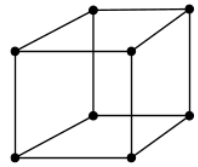
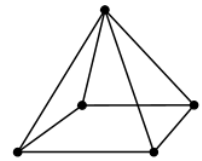
In many cases, the partial differential equations that describe the continuous problem include derivatives of the dependant variables up to the second order, so that their integral form includes first order derivatives. These problems require shape functions that are continuous over the space domain (C^0). Typical interpolation functions that satisfy this requirement are the H^1-conforming Lagrange shape functions, which are continuous functions that can be expressed in terms of Lagrange polynomials.

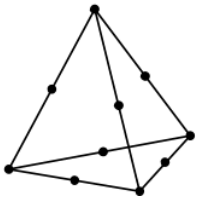
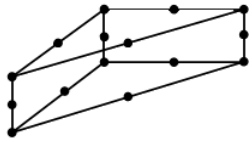
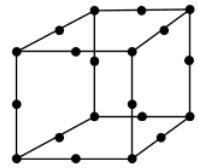
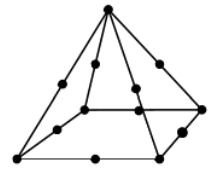
Solid Mechanics applications use isoparametric elements, where the element topology and the primary unknowns use the same order of approximation (that is, they are interpolated using shape functions of the same polynomial degree).

STAR-CCM+ offers 2D linear, 3D linear, and 3D quadratic Lagrange elements [785]. The unknowns are stored at element nodes. Quadratic elements increase accuracy by adding mid-side nodes between the corner nodes at each edge.

2D Linear Elements	
Triangle	**Quad** 

3D Linear Elements			
Tet4	**Wedge6**	**Hex8**	**Pyramid5**

3D Linear Elements			
			

3D Quadratic Elements			
Tet10	**Wedge15**	**Hex20**	**Pyramid13**
			

Consider an element with n nodes. It is convenient to define a local coordinate system, so that the position of any point within the element can be defined using local coordinates $\boldsymbol{\xi}$.

If $\boldsymbol{\xi}_P$ is the position vector of a point P in the local coordinate system, the value of $\mathbf{u}$ at point P can then be written as:

$$\mathbf{u}(\boldsymbol{\xi}_P) = \mathcal{N}_M(\boldsymbol{\xi}_P)\mathbf{u}_M \tag{3978}$$

where:

- the index $M = 1, \ldots, n$ identifies the element nodes
- $\mathbf{u}_M = \mathbf{u}(\boldsymbol{\xi}_M)$ is the value of $\mathbf{u}$ at the node M
- $\mathcal{N}_M$ is an H^1 Lagrange shape function, which determines the contribution of the nodal value $\mathbf{u}_M$ to $\mathbf{u}$
- Eqn. (3978) and all following equations use Einstein notation, which implies summation over repeated indices. For example, $\mathcal{N}_M\mathbf{u}_M \equiv \sum_{M=1}^{n}\mathcal{N}_M\mathbf{u}_M$

If M and N are two element nodes, with $M \neq N$, $\mathcal{N}_M$ is defined so that:

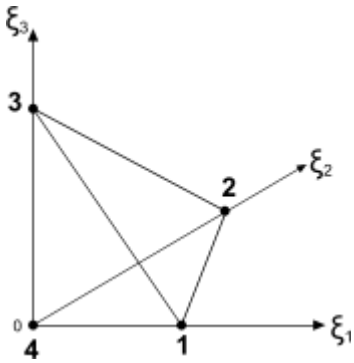
$$\begin{aligned} \mathcal{N}_M(\boldsymbol{\xi}_M) &= 1 \\ \mathcal{N}_M(\boldsymbol{\xi}_N) &= 0 \end{aligned} \tag{3979}$$

The shape functions for linear tetrahedral and hexahedral elements are described below. The local coordinate system is defined using natural coordinates:

$$\boldsymbol{\xi} = \begin{Bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{Bmatrix} \tag{3980}$$

Linear Tetrahedron (Tet4)

In tetrahedral elements, the local coordinates are $0 \leq \xi_i \leq 1$:



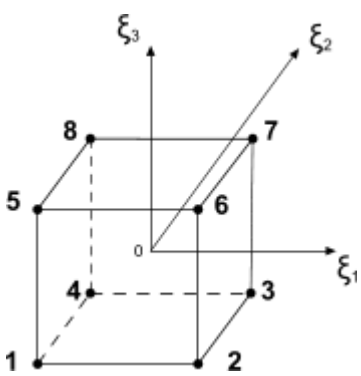
The nodal shape functions at a point of coordinates (ξ_1, ξ_2, ξ_3) are simply:

$$\begin{aligned} \mathcal{N}_1 &= \xi_1 \\ \mathcal{N}_2 &= \xi_2 \\ \mathcal{N}_3 &= \xi_3 \\ \mathcal{N}_4 &= 1 - \xi_1 - \xi_2 - \xi_3 \end{aligned} \tag{3981}$$

which correspond to the barycentric coordinates.

Linear Hexahedron (Hex8)

In hexahedral elements, the local coordinates are $-1 \leq \xi_i \leq 1$:



The nodal shape functions at a point of coordinates (ξ_1, ξ_2, ξ_3) are:

$$\mathcal{N}_M(\xi_i) = \frac{1}{8}(1 + \xi_{iM}\xi_i); \qquad M = 1, \ldots, 8 \tag{3982}$$

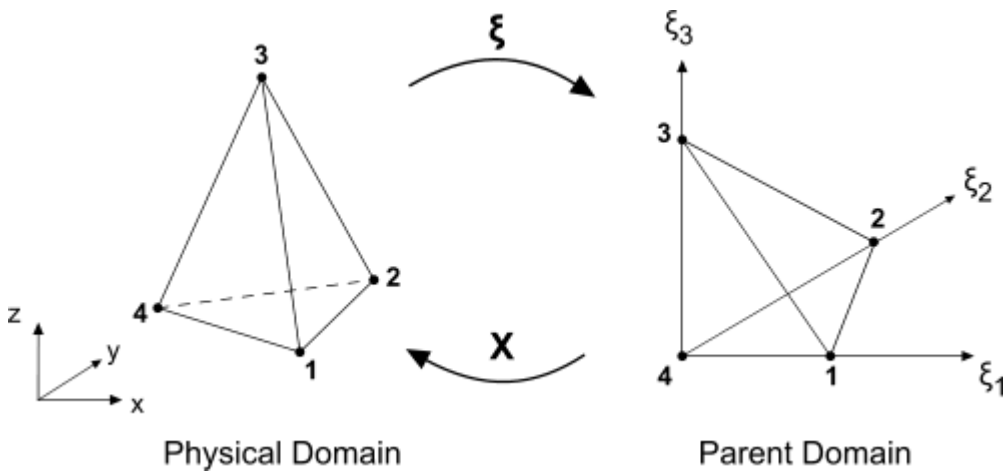
Linear hexahedra are enriched with *bubble* degrees of freedom, to overcome locking phenomena. For example, the linear shape functions Eqn. (3982) do not provide enough mode shapes to approximate shear and bending in Solid Mechanics applications, causing the elements to be overly stiff. The stiffening effect worsens for high aspect ratios and skewed element shapes. To overcome locking, STAR-CCM+ adds three additional vectors $\mathbf{u}_{oi}$ (that is, 9 internal, or bubble, degrees of freedom) to the nodal quantities $\mathbf{u}_M$:

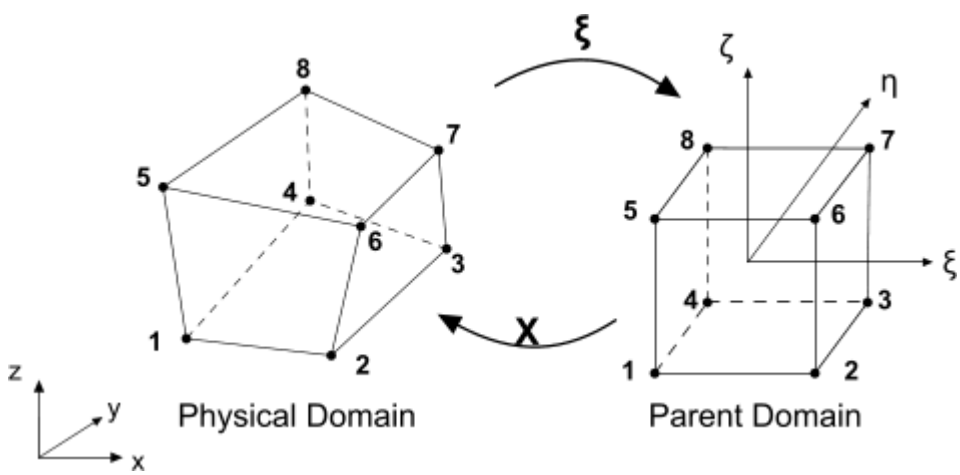
$$\mathbf{u}(\xi_i) = \mathcal{N}_M(\xi_i)\mathbf{u}_M + \mathbf{u}_{oi}(1 - \xi_i^2); \qquad i = 1, 2, 3; \quad M = 1, ..., 8; \tag{3983}$$

The accuracy of the quadratic Hex20 element is higher than the accuracy of the linear Hex8 element with bubble functions. The Hex20 is also insensitive to high aspect ratios and skewness. Tet4 elements are stiff in bending and shear and their behavior cannot be improved with the use of bubble functions.

From Local to Global: Parametric Mapping

Local-to-global mapping between the parent domain and the global physical domain ensures continuity across adjacent elements.





Shape Function Derivatives

The derivatives of the variable $\mathbf{u}$ with respect to the undeformed coordinates $\mathbf{X}$ are related to the nodal quantities $\mathbf{u}_M$ through:

$$\frac{\partial \mathbf{u}}{\partial \mathbf{X}} = \frac{\partial \mathcal{N}_M}{\partial \mathbf{X}} \mathbf{u}_M \tag{3984}$$

The derivatives of the shape functions with respect to the undeformed physical coordinates are determined from the derivatives of the shape functions with respect to the natural coordinates through:

$$\frac{\partial \mathcal{N}_M}{\partial \mathbf{X}} = \mathbf{J}^{-1} \frac{\partial \mathcal{N}_M}{\partial \boldsymbol{\xi}} \tag{3985}$$

where:

$$\frac{\partial \mathcal{N}_M}{\partial \boldsymbol{\xi}} = \begin{pmatrix} \frac{\partial \mathcal{N}_M}{\partial \xi_1} \\ \frac{\partial \mathcal{N}_M}{\partial \xi_2} \\ \frac{\partial \mathcal{N}_M}{\partial \xi_3} \end{pmatrix}; \quad \frac{\partial \mathcal{N}_M}{\partial \mathbf{X}} = \begin{pmatrix} \frac{\partial \mathcal{N}_M}{\partial X_1} \\ \frac{\partial \mathcal{N}_M}{\partial X_2} \\ \frac{\partial \mathcal{N}_M}{\partial X_3} \end{pmatrix} \tag{3986}$$

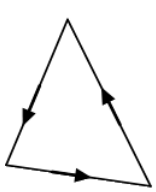
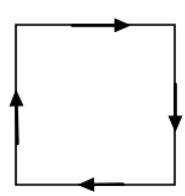
and $\mathbf{J}^{-1}$ is the inverse of the Jacobian transformation, $\mathbf{J}$:

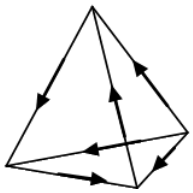
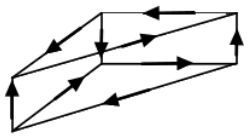
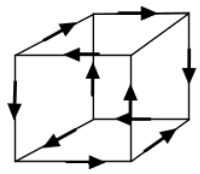
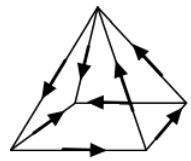
$$\mathbf{J} = \begin{bmatrix} \frac{\partial X_1}{\partial \xi_1} & \frac{\partial X_1}{\partial \xi_2} & \frac{\partial X_1}{\partial \xi_3} \\ \frac{\partial X_2}{\partial \xi_1} & \frac{\partial X_2}{\partial \xi_2} & \frac{\partial X_2}{\partial \xi_3} \\ \frac{\partial X_3}{\partial \xi_1} & \frac{\partial X_3}{\partial \xi_2} & \frac{\partial X_3}{\partial \xi_3} \end{bmatrix} \tag{3987}$$

H(Curl) Hierarchical Shape Functions

When formulating Maxwell's equations with the magnetic vector potential as dependent variable, the differential equations include second order curl-curl derivatives, so that their integral weak form includes first order curl derivatives. Furthermore, a tangential continuity condition is imposed on the vector potential and a high accuracy on the derived differential fields is desirable. These problems require shape functions whose tangential component is continuous over the space domain and their curl is well-defined. Typical interpolation functions that satisfy these requirements are the H(Curl)-conforming hierarchical shape functions.

The form of the shape functions depends on the element topology and order. STAR-CCM+ offers 2D and 3D lowest order Nédélec elements of the first kind [784].

2D Elements	
Triangle 	**Quad** 

3D Elements			
Tetrahedron	**Wedge**	**Hexahedron**	**Pyramid**
			

Consider an element with n edges. The distribution of a variable $\mathbf{u}$ can be approximated by:

$$\mathbf{u} = \boldsymbol{N}_M u_M \tag{3988}$$

where:

- the index $M = 1, \ldots, n$ identifies the element degrees of freedom
- u_M is the M-th scalar degree of freedom
- $\boldsymbol{N}_M$ is a vector H(Curl)-conforming shape function, which determines the contribution of the scalar value u_M to $\mathbf{u}$

In particular, for lowest order Nédélec elements of the first kind:

- the degrees of freedom are located at the element edges
- the number of degrees of freedom coincides with the number of element edges
- the degree of freedom can be identified as the first moment of the tangential component over each edge:

$$u_M = \int_{E_M} \mathbf{A} \cdot \boldsymbol{\tau} \, \mathrm{d}s \tag{3989}$$

  where $\boldsymbol{\tau}$ is the tangential vector along edge E_M

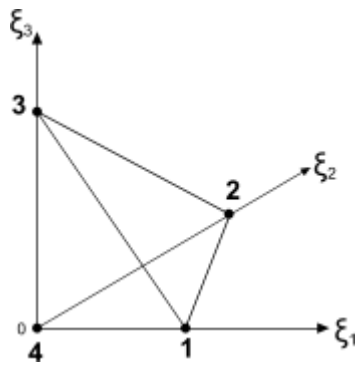
- the shape functions have constant tangential trace on the edges of the element:

$$\boldsymbol{N}_M \cdot \boldsymbol{\tau}_{E_N} = \frac{1}{|E_N|} \delta_{MN} \tag{3990}$$

  which yields the *nodality* of the basis.

  The shape functions for tetrahedral elements are described below.

Tetrahedron



In a tetrahedron, the edge shape function for the edge E_M between the nodes e_1 and e_2 is defined as:

$$\widehat{\boldsymbol{N}}_M = \nabla_\xi \mathcal{N}_{e_1} \mathcal{N}_{e_2} - \mathcal{N}_{e_1} \nabla_\xi \mathcal{N}_{e_2} \tag{3991}$$

In particular, for the edge going between the nodes 1 and 4 in the reference element, the shape function takes the explicit form:

$$\widehat{\boldsymbol{N}}_{[1;4]} = \begin{bmatrix} -1 + \xi_2 + \xi_2 \\ -\xi_1 \\ -\xi_1 \end{bmatrix} \tag{3992}$$

The shape function has a tangential component along edge $[1;4]$, but is perpendicular to all other edges.

Local-to-Global Mapping

Local-to-global mapping between the element domain and the global physical domain ensures continuity across adjacent elements:

$$\boldsymbol{N} = \mathbf{j}^{-\mathrm{T}} \cdot \widehat{\boldsymbol{N}}\big(\boldsymbol{\xi}(\mathbf{x})\big) \tag{3993}$$

where $\mathbf{j} = \partial\mathbf{x} / \partial\boldsymbol{\xi}$ and $\mathbf{x}$ is the physical deformed coordinate.

Shape Function Derivatives

The curl of the shape functions with respect to the deformed coordinates is given by:

$$\nabla_{\mathbf{x}} \times \boldsymbol{N} = j^{-1}\mathbf{j} \cdot \nabla_\xi \times \widehat{\boldsymbol{N}}\big(\boldsymbol{\xi}(\mathbf{x})\big) \tag{3994}$$

where $j = \det \mathbf{j}$

Integration Rules

All volume and surface integrals are evaluated numerically using a full Gauss integration rule appropriate to the order and topology of the element.

The Gauss rule approximates the integral of a function with a weighted sum of function values at specific points. The 3D integration is:

$$\begin{aligned}\int_{\tau} f(X_1, X_2, X_3)\mathrm{d}V &= \int_{\hat{\tau}} f(\mathbf{X}(\xi_1, \xi_2, \xi_3))\det \mathbf{J}(\xi_1, \xi_2, \xi_3)\mathrm{d}\xi_1 \mathrm{d}\xi_2\, \mathrm{d}\xi_3 = \\ &= \sum_{g=1}^{n} w_g f\left(\mathbf{X}\left(\xi_{1_g}, \xi_{2_g}, \xi_{3_g}\right)\right)\det \mathbf{J}_g\Big)\end{aligned} \tag{3995}$$

where ξ_{ig} are integration point locations, w_g are appropriate weights, and n is the number of integration points.

For example, for a hexahedral element:

n	ξ_{ig}	w_g
1 x 1 x 1 = 1	0	2
2 x 2 x 2 = 8	$\pm\frac{1}{\sqrt{3}}$	1
3 x 3 x 3 = 27	0	$\frac{8}{9}$
	$\pm\sqrt{\frac{3}{5}}$	$\frac{5}{9}$

Specifically, for the Hex8 element, STAR-CCM+ uses the standard 2x2x2 integration (2 integration points per dimension).

Sparse Direct Solver

In Finite Element simulations, STAR-CCM+ solves the system of linear equations using direct solvers.

For parallel execution, STAR-CCM+ employs MUMPS [775] a multi-frontal solver applicable to both shared-memory and distributed-memory systems. For serial execution, STAR-CCM+ employs either PARDISO [770] or MUMPS [775]. PARDISO takes advantage of multithreading, but the element-by-element calculations to compute the matrix and right-hand side are conducted in serial.

Both solvers make extensive use of graph portioning available from the Metis, BLAS, and MKL libraries for speed and accuracy. Both solvers can solve "out of core" when the available physical memory is not sufficient to store the matrix. Both also need sufficient physical memory to manage the maximum wavefront of the matrix. For an estimate of the required memory, set the solvers verbosity as appropriate.

The discretization process results in a linear system of algebraic equations of the form:

$$\mathbf{T}[\mathbf{x}] = \mathbf{r} \tag{3996}$$

where $\mathbf{T}$ is the sparse matrix of coefficients of the linear system, which is also called the tangent matrix, $\mathbf{x}$ is the unknown, and $\mathbf{r}$ the residual. The dimension of the residual vector, the unknown vector, and the tangent matrix depends on the total number of degrees of freedom. At each solver iteration, STAR-CCM+ solves Eqn. (3996) for the unknown $\mathbf{x}$.

In general, direct solvers for sparse matrices solve the system of linear equations in four steps:

- **Ordering Step**—The solver reorders the sparse matrix rows and columns, to optimize the matrix structure for solving.
- **Analysis Step**—The solver performs a pre-factorization step.
- **Factorization Step**—The solver computes the factors that express the system of equations as a matrix product.
- **Solving Step**—The solver substitutes the factors and solves the equations.

Finite Element Bibliography

[757] *Baaijens, F. P. T. 1998. "Mixed finite element methods for viscoelastic flow analysis: a review", J. Non-Newton Fluid, 79. pp. 361–385.*

[758] *Babuska, I. 1971. "Error-bounds for finite element method", Numerische Mathematik, 16, pp. 322–333.*

[759] *Babuska, I. 1972. "The finite element method with Lagrangian multipliers", Numerische Mathematik, 3, pp. 179–192*

[760] *Bird, R. B., Wiest, J. M. 1995. "Constitutive Equations for Polymeric Liquids", Annu. Rev. Fluid Mech., 27, pp. 169–193.*

[761] *Brezzi, F. 1974. "On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers", ESAIM: Mathematical Modelling and Numerical Analysis, 8, pp. 129–151.*

[762] *Brooks, A. N., Hughes, T. J. R. 1982. "Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations", Comput. Methods in Appl. Mech. Eng., 32, pp. 100–259.*

[763] *Bush, M. B., Tanner, R. I. 1983. "Numerical solution of viscous flows using integral equation methods", Int. J. Numer. Meth. Fl., 3, pp. 71–92.*

[764] *Bush, M. B., Tanner, R. I., Phan-Thien, N. 1985. "A boundary element investigation of extrudate swell", J. Non-Newton Fluid, 18, pp. 143–162.*

[765] *Crochet, M. J., Walters, K., Davies, A. R. 1984. Numerical simulation of non-Newtonian flow, Elsevier Amsterdam ; New York.*

[766] *Giraud, L., d'Humières, D., Lallemand, P 1998. "A lattice Boltzmann model for Jeffreys viscoelastic fluid", Eur. Phys. Lett. 42, p. 625.*

[767] *Guénette, R., Fortin, M. 1995. "A new mixed finite element method for computing viscoelastic flows", J. Non-Newtonian Fluid Mech., 60, pp. 27–52.*

[768] *Hughes, T. J. R., Franca, L. P., Balestra, M. 1986. "A new finite element formulation for computational fluid dynamics: V. {Circumventing} the {Babu\v{s}ka}-{Brezzi} condition: a stable Petrov-Galerkin formulation of the stokes problem accommodating equal-order interpolations", Comput. Methods in Appl. Mech. Eng., 59, pp. 85–99.*

[769] *Hughes, T. J. R. and Franca, L. P. 1987. "A new finite element formulation for computational fluid dynamics: VII. The stokes problem with various well-posed boundary conditions: Symmetric formulations that converge for all velocity/pressure spaces", Comput. Methods in Appl. Mech. Eng., 65, pp. 85–96.*

[770] *Intel Math Kernel Library, Reference Manual, Document Number: 630813-060US MKL 11.1.*

[771] *Ladyzhenskaya, O. A. 1963. The Mathematical Theory of Viscous Incompressible Flow, Gordon and Breach.*

[772] *Larson, R. G. 1998. The Structure and Rheology of Complex Fluids, Oxford University Press, USA.*

[773] *Malaspinas, O., Firètier, N., Deville, M., 2010. "Lattice Boltzmann method for the simulation of viscoelastic fluid flows", 165, pp. 1637–1653.*

[774] *McLeish, T. C. B., Larson, R. G. 1998. "Molecular constitutive equations for a class of branched polymers: The pom-pom polymer", J Rheology, 42, pp. 81–110.*

[775] *MUltifrontal Massively Parallel Solver (MUMPS 5.0.2) Users' Guide, July 15, 2016.*

[776] *Oliveira, P. J., Pinho, F. T., Pinto, G. A. 1998. "Numerical simulation of non-linear elastic flows with a general collocated finite-volume method," J. Non-Newton Fluid, 79, pp. 1–43.*

[777] *Owens, R.G., Phillips, T.N. 2002. Computational Rheology, Imperial College Press.*

[778] *Phan-Thien, N. and Tanner, R. I. 1977. "A new constitutive equation derived from network theory", J. Non-Newtonian Fluid Mech., 2, pp, 353–365/*

[779] *Giesekus, H. 1982. "A simple constitutive equation for polymer fluids based on the concept of deformation-dependent tensorial mobility", J. Non-Newtonian Fluid Mech., 11, pp. 69–109.*

[780] *Shakib, F., Hughes, T. J. R., Johan, Z. 1991. "Second World Congress on Computational Mechanics A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier-Stokes equations", Comput. Methods in Appl. Mech. Eng. 89, pp. 141–219.*

[781] *Tezduyar, T. E. 1991. "Stabilized Finite Element Formulations for Incompressible Flow Computations", Adv. Appl. Mech., 28, pp. 1–44.*

[782] *Tezduyar, T. E., Osawa, Y. 2000. "Finite element stabilization parameters computed from element matrices and vectors", Comput. Methods in Appl. Mech. Eng., 190, 411–430*

[783] *Xue, S.-C., Phan-Thien, N., Tanner, R. I. 1995. "Numerical study of secondary flows of viscoelastic fluid in straight pipes by an implicit finite volume method", J. Non-Newton Fluid, 59, pp. 191–213.*

[784] *Zaglmayr, S. 2006. High Order Finite Element Methods for Electromagnetic Field Computation. PhD thesis, Johannes Kepler University, Linz Austria.*

[785] *Zienkiewicz, O.C., Taylor, R.C., and Zhu, J.Z. 2005. The Finite Element Method: Its Basics and Fundamentals, Elsevier Butterworth-Heinemann, Oxford, 6th edition.*

~Mesh Motion~

For this release, see *User Guide > Modeling Physics > Modeling Space, Time, and Motion > Modeling Motion > General Motion Formulation*.

For specific theory, see:

- *User Guide > Modeling Physics > Modeling Space, Time, and Motion > Modeling Motion > Working with Morphing > Morphing Motion Formulation*
- *User Guide > Modeling Physics > Modeling Dynamic Fluid Body Interaction > DFBI Formulation*

For these models, theory guide content is planned for a future release.

Contents:

Morphing

Morphing

In STAR-CCM+, the initial movement that the morpher algorithm imposes on the mesh is defined via a set of control vertices. Each control vertex is associated with a displacement vector that the morpher then uses to move nearby vertices.

These specified displacements utilise interpolation theory [13] to generate an interpolation field for the region, by which the morpher motion is defined. The morpher uses the interpolation field to move mesh vertices to their new positions.

Interpolation Algorithms

STAR-CCM+ uses the Radial Basis Function (RBF) to interpolate and move the mesh.

The interpolation field is generated via a system of equations; these are created using the control vertices, and their specified displacements, where the known displacement $\mathbf{d}'_i$ of each control vertex i is expanded as

$$\mathbf{d}'_i = \sum_{j=1}^{N} f_{b,j}\left(r_{ij}\right)\boldsymbol{\lambda}_j + \boldsymbol{\alpha} \tag{3997}$$

in which $f_{b,j}\left(r_{ij}\right)$ is a radial basis function of the form

$$f_{b,j}\left(r_{ij}\right) = \sqrt{r_{ij}^{\ 2} + c_j^{\ 2}} \tag{3998}$$

and r_{ij} is the magnitude of the distance between two vertices:

$$r_{ij} = \left|\mathbf{x}_i - \mathbf{x}_j\right| \tag{3999}$$

In these equations, $\boldsymbol{\lambda}_j$ is the expansion coefficient, $\mathbf{x}_i$ is the position of vertex i, N is the number of control vertices, and c_j is the basis constant. The basis constant is set to zero in STAR-CCM+. The constant vector $\boldsymbol{\alpha}$ is used to satisfy the additional constraint that

$$\sum_{j=1}^{N} \boldsymbol{\lambda}_j = 0 \tag{4000}$$

This additional constraint bounds the expansion for large $\mathbf{x}$. The system that is described in Eqn. (3997) is non-singular if $\mathbf{x}_i \neq \mathbf{x}_j$ for all $i \neq j$.

Equations Eqn. (3997) and Eqn. (4000) are solved to give the Cartesian components of all $\boldsymbol{\lambda}_j$ and the components of constant vector $\boldsymbol{\alpha}$. This results in the desired interpolation field:

$$\mathbf{d}'(\mathbf{x}) = \sum_{j=1}^{N} f_{b,j}(r)\boldsymbol{\lambda}_j + \boldsymbol{\alpha} \tag{4001}$$

where $f_{b,j}(r)$ applies at all vertices in the mesh, that is,

$$f_{b,j}(r) = \sqrt{r^2 + c_j^{\ 2}}$$
$$r = \left|\mathbf{x}\text{-}\mathbf{x}_j\right|$$

This equation can now be used to move mesh vertices by the calculated displacement $\mathbf{d}'$.

The matrices that are involved in equations Eqn. (3997) and Eqn. (4001) are dense, and a straightforward matrix operation would require $O(N^2)$ operations. Fortunately, an $O(N\log(N))$ method has been developed to solve the linear equation Eqn. (3997) and evaluate equation Eqn. (4001) at many vertex locations. The method involves a conjugate gradient method with a special preconditioner ([14], [15]) and a fast multipole method [16] to accelerate the matrix-vector multiplication.

Boundary Plane Conditions

Two plane boundaries are provided for use with the morpher motion: boundary plane and fixed boundary plane. For the fixed boundary plane condition, the vertices which lie on the specified plane are fixed, while for the boundary plane condition they are permitted to move within the plane itself. An effect similar to the fixed boundary plane condition can be realized by using control vertices but the fixed boundary plane condition itself can be more efficient. The movement of vertices near the constraint planes is such that they join smoothly with the movement of vertices on the plane. This movement near the constraint planes can be controlled using the damping distance L_p that is described below.

These constraints are imposed before solving Eqn. (3997). In preparation, user-specified displacements $\mathbf{d}_i$ are transformed to $\mathbf{d}'_i$ by solving

$$\mathbf{d}_i = \prod_{p=1}^{N_p} \left[s_p \mathbf{n}_p \mathbf{n}_p^T + t_p \left(1 - \mathbf{n}_p \mathbf{n}_p^T \right) \right] \mathbf{d}'_i \tag{4002}$$

where $\mathbf{n}_p$ is the normal of the p-th plane and N_p is the number of planes. The displacements $\mathbf{d}'_i$ are transformed back to $\mathbf{d}_i$ together with other mesh vertices at the end of the morphing process. s_p and t_p are functions that depend on the plane type and the continuity condition you choose. These functions are given as follows:

	C_0 continuous		C_2 continuous	
	s_p	t_p	s_p	t_p
Fixed plane	$f_0(l)$	$f_0(l)$	$f_2(l)$	$f_2(l)$
In-plane	$f_0(l)$	1	$f_2(l)$	1

Here:

$$f_0(l) = \begin{cases} 1 & ;\ l \geq 1 \\ 1 - (1-l)^3; & 1 > l \geq 0 \\ 0 & ;\ l < 0 \end{cases} \tag{4003}$$

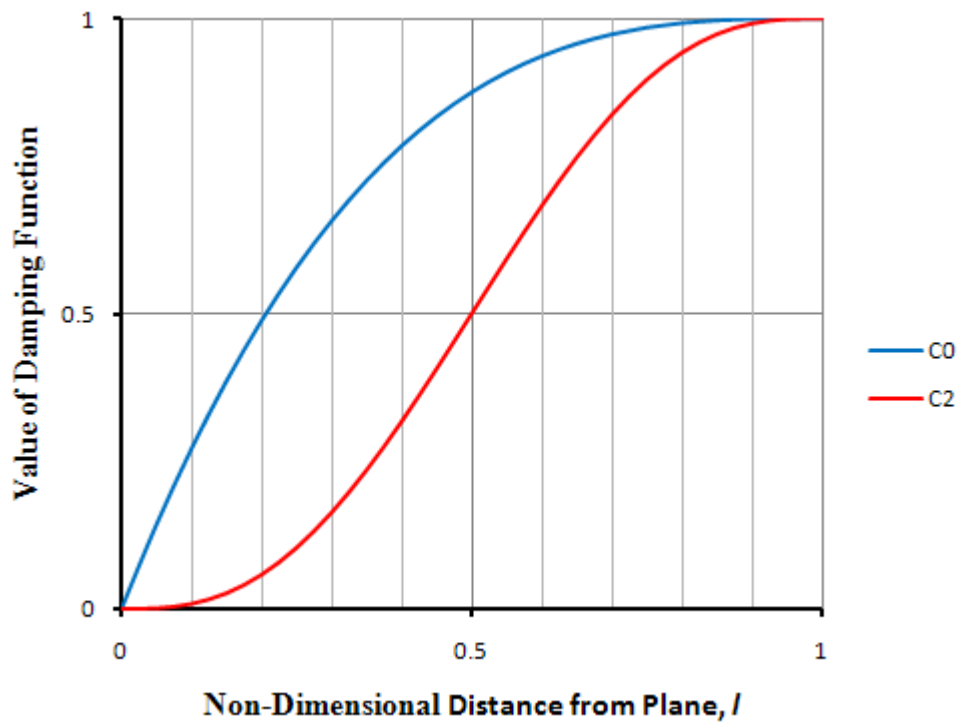
$$f_2(l) = \begin{cases} 1 & ;\ l \geq 1 \\ l^3\ \left(10 - 15l + 6l^2\right); & 1 > l \geq 0 \\ 0 & ;\ l < 0 \end{cases} \tag{4004}$$

in which

$$l = \frac{\mathbf{n}_p \left(\mathbf{x} - \mathbf{x}_{0,p} \right)}{L_p} \tag{4005}$$

is the ratio of the signed distance from the p-th plane to $\mathbf{x}_i$ and the damping distance L_p. $\mathbf{x}_{0,p}$ is a position vector on the p-th plane and L_p is either entered as a finite distance or calculated automatically as $(L_{min} + L_{max})/2$. L_{min} and L_{max} are respectively the distances from the plane to the nearest and farthest control vertices on the active side of the plane.

Functions f_0 and f_2 in the range from 0 to 1 are plotted below for comparison. A continuity level of C_2 results in a smooth transition.



Solution Steps

The steps that are involved in the morpher stage within a time step are:

- The control vertex displacements $\mathbf{d}_i$ are transformed to $\mathbf{d}'_i$ according to Eqn. (4002).
- Eqn. (3997) is solved with Eqn. (4000) to provide an interpolation field.
- The displacement vectors are transformed from $\mathbf{d}'_i$ to $\mathbf{d}_i$ using Eqn. (4002).

Rotating Flow

Rotating flow usually occurs in rotating machinery such as pumps, propellers, fans, wind turbines, and so on.

STAR-CCM+ provides many possibilities to simulate rotating flows. These possibilities include using moving reference frames or rigid body motion involving mesh motion. Both of the approaches require the resolution of the detailed blade geometry thus leading to a large number of cells, which increases analysis runtime and hardware requirements.

A further option to model rotating flows is to use simplified approaches that model the effect of rotating machinery on the surrounding flow field by adding volumetric body forces to the momentum equations or by imposing a pressure jump across a plane surface. These approaches are described in the following sections.

Contents:

Virtual Disk Model
Fan Modeling

Virtual Disk Model

The virtual disk models the effect of a rotating turbo machine on the surrounding flow field. The geometry of the turbo machine is not resolved. Instead, the body forces created by the rotating turbo machine are distributed over a cylindrical disk volume.

Contents:

Body Force Propeller Method
Blade Element Method
1D Momentum Method

Body Force Propeller Method

The body force propeller method is used to model the effects of a propeller such as thrust and torque and thereby creating propulsion without actually resolving the geometry of the propeller. The method employs a uniform volume force $\mathbf{f}_b$ distribution over the cylindrical virtual disk. The volume force varies in radial direction.

The radial distribution of the force components follows the Goldstein optimum and is given by:

$$f_{bx} = A_x r^* \sqrt{1-r^*} \tag{4006}$$

$$f_{b\theta} = A_\theta \cdot \frac{r^*\sqrt{1-r^*}}{r^*\left(1-\mathrm{r}'_h\right) + \mathrm{r}'_h} \tag{4007}$$

$$r^* = \frac{\mathrm{r}'-\mathrm{r}'_h}{1-\mathrm{r}'_h} \tag{4008}$$

$$\mathrm{r}'_h = \frac{R_H}{R_P} \text{and } \mathrm{r}' = \frac{r}{R_P}$$

where f_{bx} is the body force component in axial direction, $f_{b\theta}$ is the body force component in tangential direction, r is the radial coordinate, R_H is the hub radius and R_P is the propeller tip radius.

The constants A_x and A_θ are computed as:

$$A_x = \frac{105}{8} \cdot \frac{T}{\pi\Delta\left(3R_H + 4R_P\right)\left(R_P - R_H\right)} \tag{4009}$$

$$A_\theta = \frac{105}{8} \cdot \frac{Q}{\pi\Delta R_P\left(3R_H + 4R_P\right)\left(R_P - R_H\right)} \tag{4010}$$

where T is the thrust, Q is the torque, and Δ is the virtual disk thickness.

The computation of the body force components requires several user inputs. A propeller performance curve needs to be specified, which gives the dimensionless thrust coefficient K_T, the torque coefficient K_Q, and the propeller efficiency η_0 as functions of the advance ratio J.

$$K_T = \frac{T}{\rho n^2 d^4} \tag{4011}$$

$$K_Q = \frac{Q}{\rho n^2 d^5} \tag{4012}$$

$$J = \frac{V_A}{nD_P} \tag{4013}$$

where V_A is the speed of advance of the propeller, n the rotation rate, and D_P the propeller diameter.

Further inputs are the position of the propeller within the computational domain, the direction of the propeller rotational axis, and the direction of rotation.

The simulation is performed for a certain operating point. The operating point can be specified by either of these quantities:

- Rotation rate n
- Thrust T
- Torque Q

If the operation point is given by rotation rate n, the procedure for obtaining the body force component distribution over the virtual disk is the following:

1. The advance ratio J is calculated as:

$$J = \frac{V_{\text{Inflow Plane}}}{nD_P} \tag{4014}$$

2. The thrust coefficient K_T and the torque coefficient K_Q are interpolated from the specified propeller performance curve:

$$K_T, K_Q = f(J)$$

3. With K_T and K_Q available, thrust T and torque Q are calculated for the propeller:

$$T = \frac{K_T \rho V^2{}_{\text{Inflow Plane}} D^2{}_P}{J^2} \tag{4015}$$

$$Q = \frac{K_Q \rho V^2{}_{\text{Inflow Plane}} D^3{}_P}{J^2} \tag{4016}$$

4. With T and Q available, the axial and the tangential body force components f_{bx} and $f_{b\theta}$ are calculated according to Eqn. (4006) to Eqn. (4010).

If the operation point is given by thrust T or by torque Q, the first step of above procedure is replaced by:

1. With a given T or Q, the advance ratio is calculated by solving the following equation numerically:

$$f(J) = K_T - \mathrm{K}'_T \text{ or } f(J) = K_Q - \mathrm{K}'_Q \tag{4017}$$

   where K_T and K_Q are evaluated from the propeller performance curve.

   K'_T and K'_Q are evaluated as:

$$\mathrm{K}'_T = \frac{J^2 T_{\text{Operating Point}}}{\rho_{\text{Inflow Plane}} V^2{}_{\text{Inflow Plane}} D_P{}^2} \tag{4018}$$

$$\mathrm{K}'_Q = \frac{J^2 Q_{\text{Operating Point}}}{\rho_{\text{Inflow Plane}} V^2{}_{\text{Inflow Plane}} D_P{}^3} \tag{4019}$$

With the advance ratio J available, the calculation of the axial and tangential body force components follows the same procedure as for the rotation rate operating point continuing with the second step.

Blade Element Method

The blade element method formulation is based on the methodology that is proposed by Rajagoplan and others. It is primarily designed to model the effects of a helicopter rotor on the flow field without actually prescribing the exact geometry of rotor blades.

This approach consists of two parts. The first step is to identify the location of cells within the computational domain that belong to the virtual disk and to which the source term is applied. The second step involves the calculation of the momentum source term vector for each cell, which was identified and marked in the first step.

Step 1: Cell Marking and the Allocation to the Blade Elements/Buckets Based on the Interpolation Grid

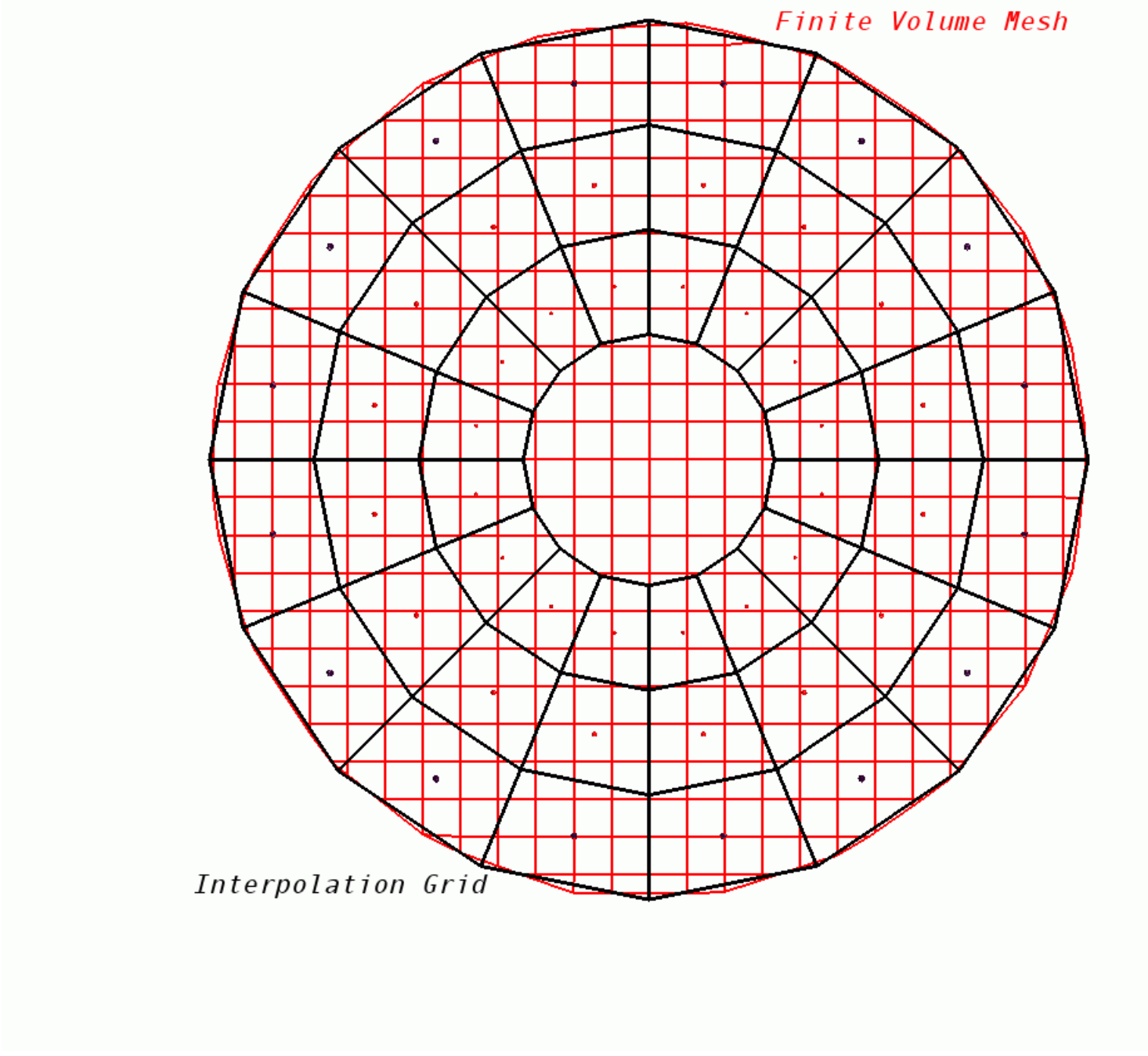
The information for this step is user-specified and provided through the virtual disk nodes:

- **Disk Geometry**

  Describes the geometrical dimensions and the orientation of the disk
- **Virtual Disk Resolution**

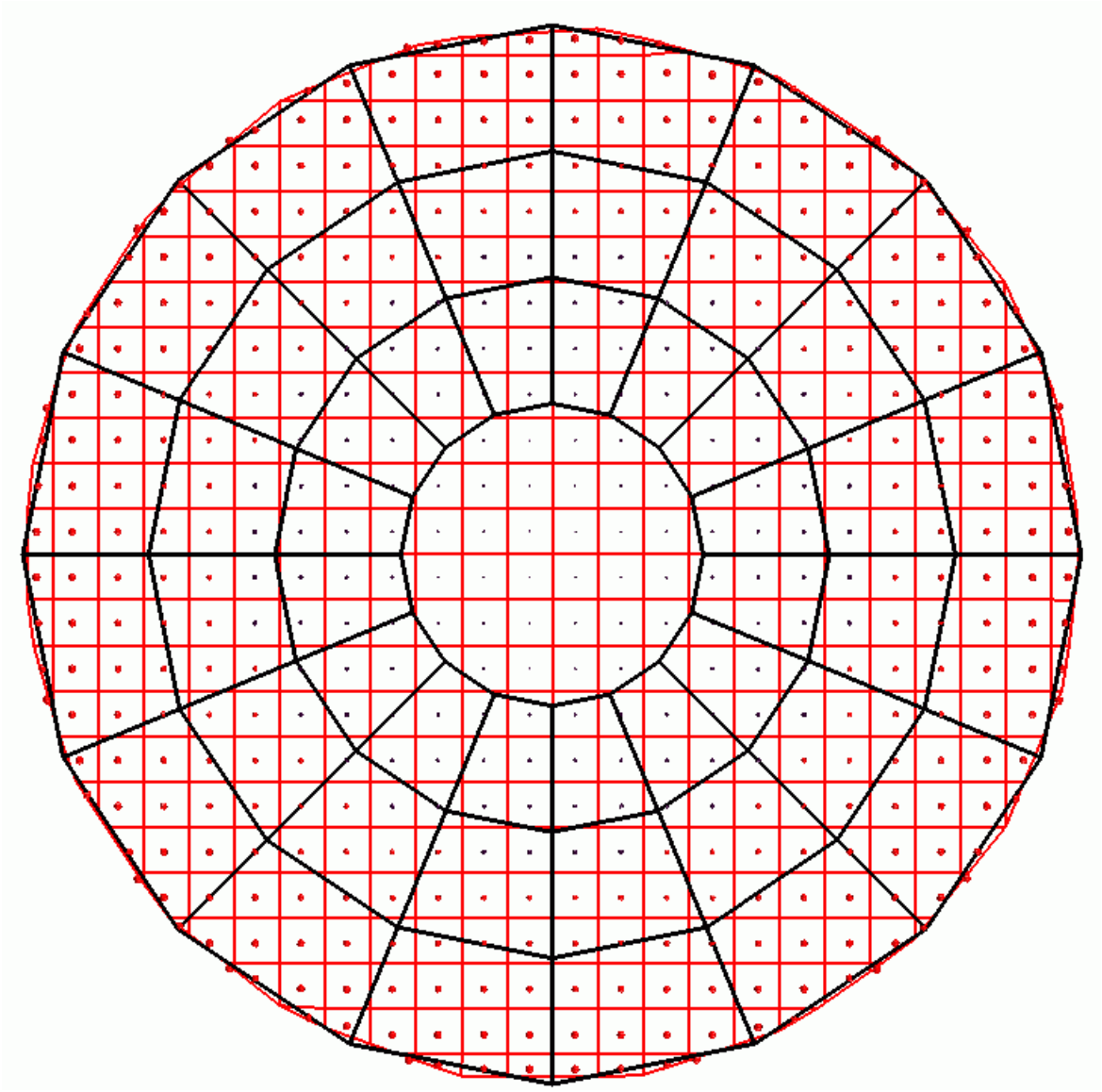
  Specifies the size of the two-dimensional interpolation grid in polar coordinates, representing the actual blade elements. The blade properties such as chord, twist, and aerodynamic characteristics prevail throughout the whole element.

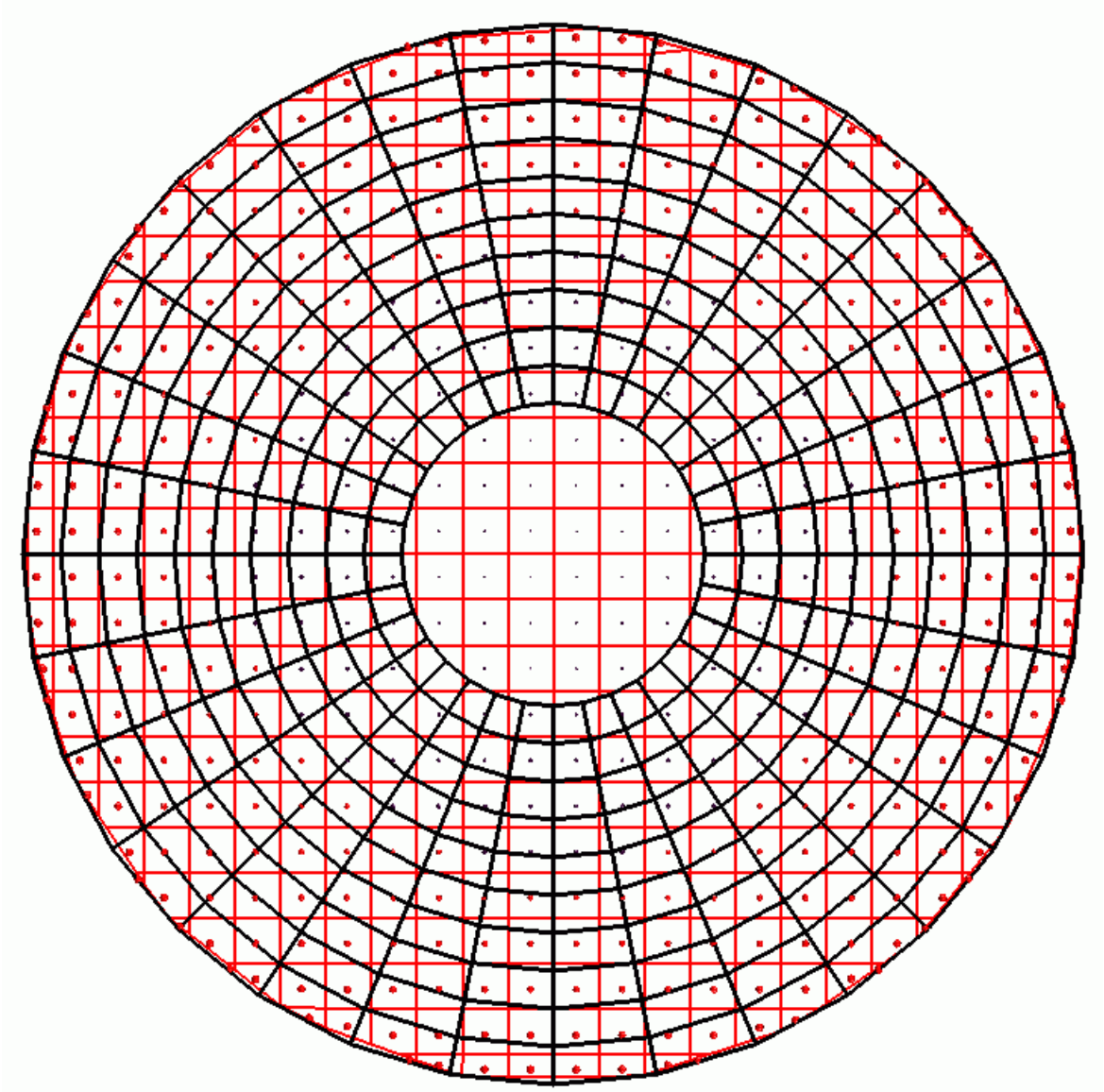
The source term is calculated per blade element on the interpolation grid and then transferred to the finite volume mesh. Each element is associated with a number of cells on the finite volume mesh. For the coupling between the local flow conditions and the modeled rotor, the information is taken from the finite volume cell which is the closest to the center of a blade element on the interpolation grid. Thus, the specified size of the interpolation grid is the actual computational effort of the influence of the rotor on the flow field. The sketch below explains the flow field coupling between the finite volume mesh (red) and the virtual disk interpolation grid (black).



The marking procedure is itself a two-step procedure. First, all the cells that fall within the circular disk area are marked to receive the source term. Second, each of the marked cells is assigned a bucket/element number corresponding to its location with respect to the interpolation grid. So each of the cells that was marked during the first step belongs to one of the elements/buckets of the interpolation grid.

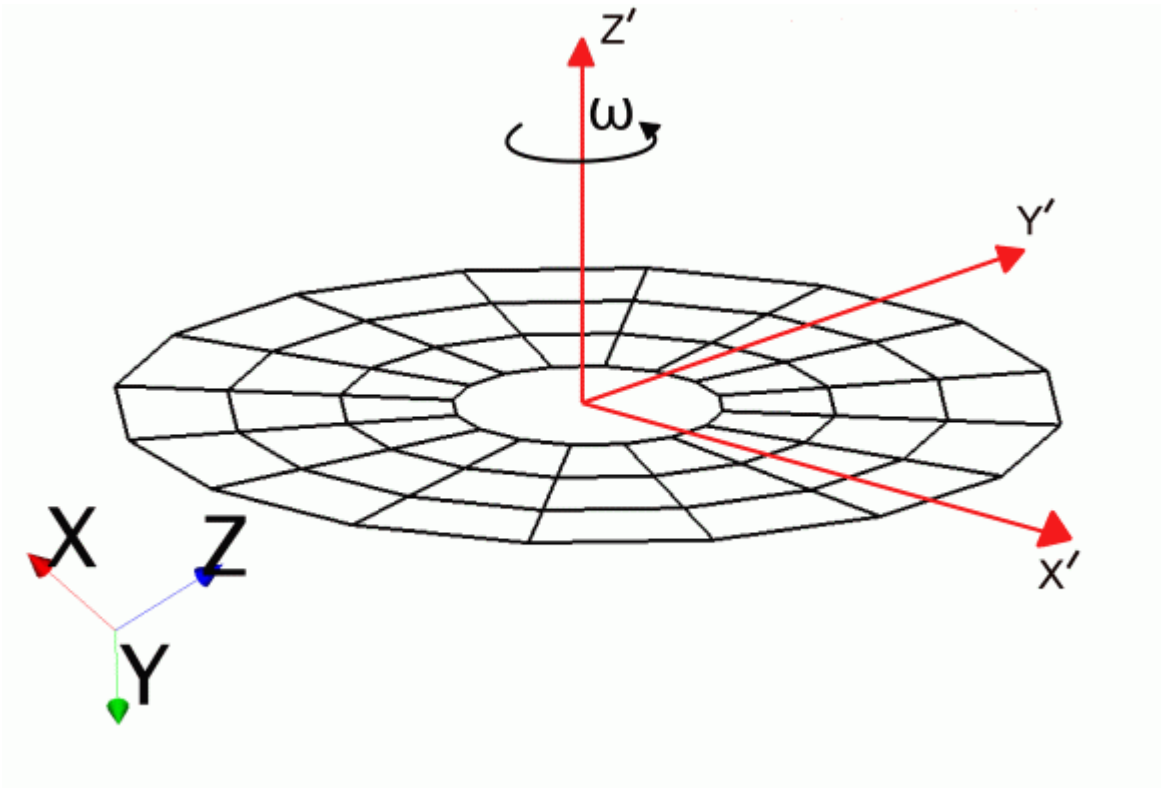
This approach requires that each element/bucket has at least one cell in its domain. If the finite volume mesh is very coarse and the interpolation grid is too fine then, you are asked to reduce the resolution. The following two figures show the interpolation grid for two different resolutions. For the fine interpolation grid in the second figure, no finite volume cell come under the domain of some for some of the elements/buckets at the innermost radius.





Step 2: Calculation of Momentum Source Term Strength

The second part of the blade element method calculates the strength of momentum source term for the elements on the interpolation grid. Based on the geometry of the rotor (that is orientation and origin), a disk local cartesian coordinate system (X', Y', Z') is created. This cartesian coordinate system has its origin at the center of the rotor and the Z' -axis in the direction corresponding to the thrusting direction of the disk. The Z' -axis is perpendicular to the plane of rotation, while the X' and Y' axes lie in the plane of the disk. The following sketch demonstrates the disk local coordinate system:



The rotational speed ω is always oriented along the Z'-direction. +ω corresponds to a counter-clockwise wise (CCW) rotation and -ω corresponds to a clock-wise (CW) rotation when viewed from the +Z' direction.

To account for blade flapping, two additional coordinate systems are defined:

- A rotor-based cylindrical system $(r, \phi, z)$with respect to the basis of the local coordinate system$(\mathrm{X}', \mathrm{Y}', \mathrm{Z}')$
- A Cartesian coordinate system with respect to the basis of the cylindrical coordinate system. It is called the flap coordinate system.

The basis of the flap system (s, ϕ, n) is defined such that $\mathbf{e}_s$ points in the spanwise direction of the blade. The basis vector $\mathbf{e}_\phi$ of the flap coordinate system points in the same direction as $\mathbf{e}_\phi$ of the rotor-based cylindrical system. The cylindrical system rotates with the blade and the flap system is tilted with respect to the cylindrical system. Given the azimuth location ψ of the blade, the cylindrical system is rotated such that $\mathbf{e}_r$ is aligned with the blade span. The flap system is tilted by $\delta(\psi)$, that is, by the flap angle of this azimuth location. All calculations are performed in the flap system. The resulting force vector is transformed back to the lab system and then added to the momentum equation as a source term.

The local flow field velocity in the laboratory coordinate system is:

$$\mathbf{v} = v_x\hat{\mathbf{i}} + v_y\hat{\mathbf{j}} + v_z\hat{\mathbf{k}} \tag{4020}$$

The local flow velocity is transferred to the flap coordinate system using a sequence of transformation matrices $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3$:

$$\begin{bmatrix} v_{x'} \\ v_{y'} \\ v_{z'} \end{bmatrix} = \mathbf{M}_3\mathbf{M}_2\mathbf{M}_1 \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \tag{4021}$$

where the transformation matrices relate the basis vectors of the various coordinate systems as:

$$\begin{bmatrix} \mathbf{e}_{x'} \\ \mathbf{e}_{y'} \\ \mathbf{e}_{z'} \end{bmatrix} = M_1 \begin{bmatrix} \mathbf{e}_x \\ \mathbf{e}_y \\ \mathbf{e}_z \end{bmatrix} \quad \begin{bmatrix} \mathbf{e}_r \\ \mathbf{e}_\phi \\ \mathbf{e}_z \end{bmatrix} = M_2 \begin{bmatrix} \mathbf{e}_{x'} \\ \mathbf{e}_{y'} \\ \mathbf{e}_{z'} \end{bmatrix} \quad \begin{bmatrix} \mathbf{e}_s \\ \mathbf{e}_\phi \\ \mathbf{e}_n \end{bmatrix} = M_3 \begin{bmatrix} \mathbf{e}_r \\ \mathbf{e}_\phi \\ \mathbf{e}_z \end{bmatrix} \tag{4022}$$

The blade has a velocity due to rotation, which can be written in the flap coordinate system as:

$$\mathbf{v}^{bl} = \mathbf{M}_3\mathbf{M}_2\mathbf{M}_1(\boldsymbol{\omega} \times \mathbf{r})$$

where $\mathbf{r}$ is the radius vector.

The velocity relative to the blade $\mathbf{v}^{rel} = v_s\mathbf{e}_s + v_\phi\mathbf{e}_\phi + v_n\mathbf{e}_n$ is thus given by:

$$\mathbf{v}^{rel} = \mathbf{v} - \mathbf{v}^{bl}$$
$$= \mathbf{M}_3\mathbf{M}_2\mathbf{M}_1\mathbf{v} - \mathbf{M}_3\mathbf{M}_2\mathbf{M}_1(\boldsymbol{\omega} \times \mathbf{r})$$

To calculate the effective angle of attack α, the following blade angles are taken into account:

- induced angle of attack (inflow angle) β
- blade flap angle $\delta(\psi)$
- geometrical pitch angle $\theta(\psi)$
- geometrical twist angle $\theta_{tw}(r)$

The blade flap angle is calculated as:

$$\delta = \delta_0 - \delta_{1c}\cos(\psi) - \delta_{1s}\sin(\psi) \tag{4023}$$

where δ_0 is the coning angle, δ_{1c} is the longitudinal flap angle, and δ_{1s} is the lateral flap angle. The time derivative is given by:

$$\dot{\delta} = \omega\frac{d\delta}{d\psi} = \omega\left(\delta_{1c}\sin(\psi) - \delta_{1s}\cos(\psi)\right) \tag{4024}$$

and the flap velocity:

$$\mathbf{v}^{flap} = \mathbf{r}\dot{\delta} \tag{4025}$$

The induced angle of attack is now calculated as:

$$\beta = \arctan\left[\frac{-\left(v_n + v_n^{flap}\right)}{\sqrt{\left(v_\phi\right)^2 + \left(v_s\right)^2}}\right] \tag{4026}$$

It can be seen from Eqn. (4026) that the blade flap angle affects the inflow angle (induced angle of attack) through the blade flap velocity, and thus the effective angle of attack α.

The pitch angle θ is defined as:

$$\theta = \theta_0 - \theta_{1c}\cos\psi - \theta_{1s}\sin\psi \tag{4027}$$

where ψ is the azimuth angle, θ_0 is the collective pitch angle, θ_{1c} is the lateral cyclic pitch angle, and θ_{1s} is the longitudinal cyclic pitch angle.

The final effective angle of attack is then calculated as:

$$\alpha = \theta + \theta_{tw} - \beta \tag{4028}$$

Knowing the angle of attack α and the blade relative velocity $\mathbf{v}^{rel}$ experienced by the airfoil, either the Mach number or the Reynolds number are calculated. The aerodynamic force coefficients are also used to calculate the Mach and the Reynolds number. You specify these coefficients in the form of tables in the simulation tree. Which number is calculated depends on how you specify the aerodynamic force coefficients—as a function of Mach number C_l, $C_d = f(\alpha, Ma)$ or or as a function of Reynolds number$f(\alpha, Re)$.

The lift and drag forces are then computed as:

$$\begin{aligned} \mathrm{L}' &= \frac{1}{2} \cdot \rho \cdot \left|\mathbf{v}^{rel}\right|^2 \cdot C_l \cdot c \cdot dr \\ \mathrm{D}' &= \frac{1}{2} \cdot \rho \cdot \left|\mathbf{v}^{rel}\right|^2 \cdot C_d \cdot c \cdot dr \end{aligned} \tag{4029}$$

where c is the blade chord.

To account for tip-loss due to trailing vortices at the tip of a rotor blade, the lift force is multiplied by a tip-loss correction factor F:

$$L'_{\mathrm{corr}} = L \cdot F \tag{4030}$$

The forces are then converted into the normal and tangential force (transformation from the wind axis to the rotor flap system):

$$\mathrm{F}'_n = \mathrm{L}'\cos\beta - \mathrm{D}'\sin\beta$$
$$\mathrm{F}'_t = \mathrm{L}'\sin\beta - \mathrm{D}'\cos\beta$$

As there are no aerodynamic forces along the blade span, $F_s = 0$. The resultant aerodynamic force vector can be assembled on the blade element as $\left(F_s, F_\phi, F_n\right)$ in the flap system.

Finally, the force vector is transformed into the laboratory coordinate system:

$$\begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix} = \mathbf{M}_1^T \mathbf{M}_2^T \mathbf{M}_3^T \begin{pmatrix} F_s \\ F_\phi \\ F_n \end{pmatrix} \tag{4031}$$

The instantaneous force acting on the fluid element is then $-\mathbf{F}$.

The time-averaged source terms $\mathbf{S} = \left(S_x, S_y, S_z\right)$ are added to the discretized momentum conservation equations. Since the blade only spends a finite fraction of its total revolution time passing through the control volume, the time-averaged source terms are scaled by the time fraction:

$$\mathbf{S} = \frac{b\Delta\varphi}{2\pi}\left(-\mathbf{F}\right) \tag{4032}$$

where b is the number of blades and $\Delta\varphi$ is the angular distance through which the blade traverses in passing through a blade element.

Numerical Trim Algorithm

In the context of rotor crafts or helicopters, the term *trim* means the correct adjustment of the aircraft controls and the overall orientation of the rotorcraft in order to obtain a desired steady flight condition. For example, for the hover condition, the objective of trimming the rotor is to achieve an equilibrium, that is, that the net sum of forces and the net sum of moments about the center of gravity are zero. STAR-CCM+ uses the isolated trim (wind tunnel trim) algorithm. The isolated trim algorithm evaluates the rotor pitch controls, which are the collective, the lateral cyclic, and the longitudinal cyclic pitch.

Main Rotor Controls

Generally, a helicopter has four independent controls: three for the main rotor and one for the tail rotor. In the STAR-CCM+ implementation, the rotor is treated in isolation and only the rotor controls are considered.

The main rotor controls are:

- Collective pitch:

  By using this control, a pilot can change the pitch angle of all the blades of the main rotor collectively, that is, by the same amount. This change effectively influences the total thrust generated by the main rotor.
- Cyclic pitch:

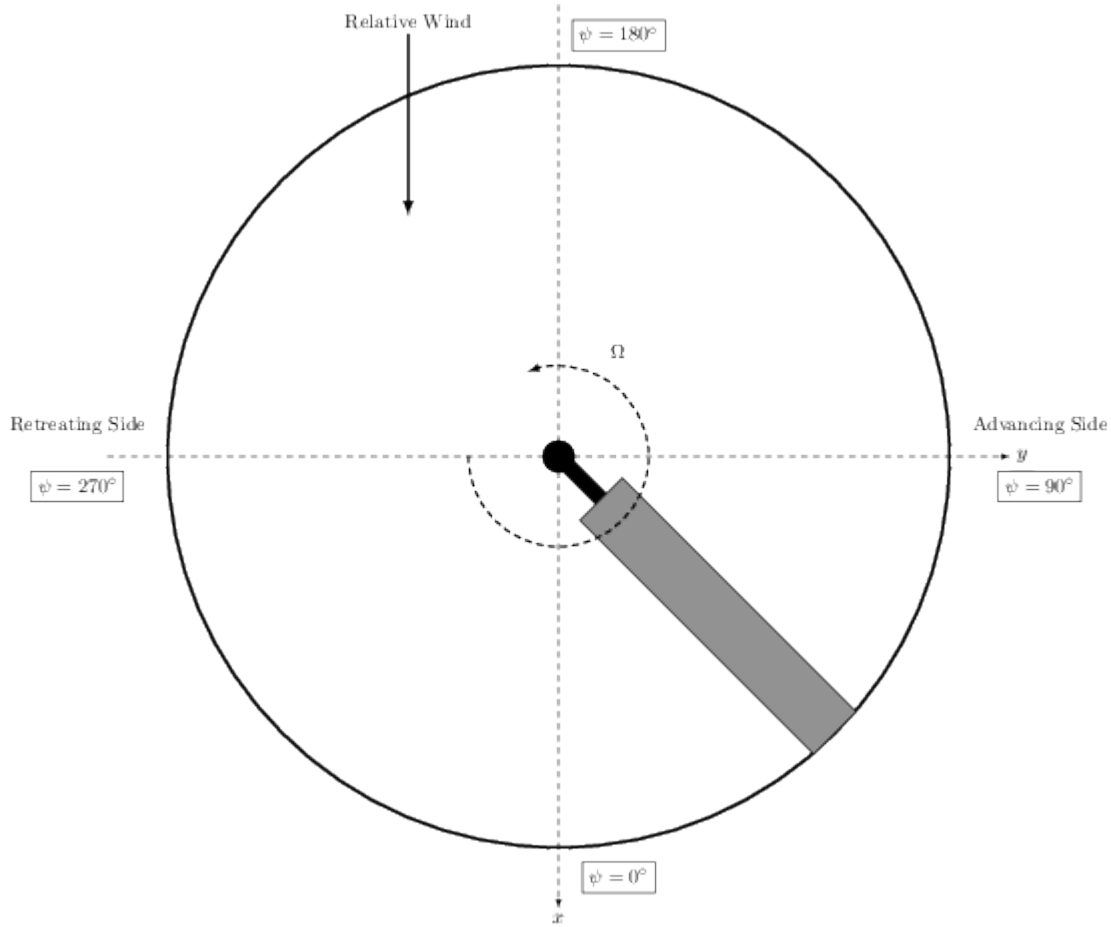
  By using this control, a pilot can change the pitch angle once per revolution. The cyclic pitch is categorized as follows:
  - Lateral cyclic pitch θ_{1c}:

    Varies as per $\cos(\psi)$. The lateral cyclic pitch is at a maximum or minimum when a blade is at 0° and 180°. The rotor response, which is phased by 90°, leads to a control effect in the lateral direction (y-axis) and effectively influences the roll moment.
  - Longitudinal cyclic pitch θ_{1s}

    Varies as per $\sin(\psi)$. The longitudinal cyclic pitch is at a maximum or minimum when a blade is at 90° and 270°. The rotor response, which is phased by 90°, leads to a control effect in the longitudinal direction (x-axis) and effectively influences the pitch moment.

The total pitch angle therefore is a function of the azimuthal angle ψ and is given by:

$$\theta = \theta_0 - \theta_{1c}\cos(\psi) - \theta_{1s}\sin(\psi) \tag{4033}$$



Algorithm

The rotor aerodynamic response is a highly coupled function of its control inputs and its aerodynamic environment. Although the thrust generated by rotor is mainly influenced by collective pitch, it is also influenced by the cyclic pitch inputs. Likewise, the roll and pitch moments are directly linked to cyclic pitch inputs but also produce a change in thrust. Defined in the local coordinate system of the virtual disk:

$$\begin{aligned} T &= T(\theta_0, \theta_{1c}, \theta_{1s}) \\ M_x &= M_x(\theta_0, \theta_{1c}, \theta_{1s}) \\ M_y &= M_y(\theta_0, \theta_{1c}, \theta_{1s}) \end{aligned} \tag{4034}$$

where T is the thrust, M_x is the roll moment, and M_y is the pitch moment.

This coupled system of equations can be interpreted as how the control input vector $\mathbf{x}$ is related to the rotor response vector $\mathbf{y}$ where:

$$\mathbf{x} = \begin{pmatrix} \theta_0 \\ \theta_{1c} \\ \theta_{1s} \end{pmatrix} \quad \text{and} \quad \mathbf{y} = \begin{pmatrix} T \\ M_x \\ M_y \end{pmatrix}$$

Each flight condition corresponds to a state of the rotor response vector $\mathbf{y}$, which requires a particular state of the control input vector $\mathbf{x}$, which in turn affects the aerodynamic characteristics of a helicopter.

To investigate the aerodynamic performance of a helicopter, you first choose a flight condition. This means that the response vector $\mathbf{y}$ is known, but not the control input vector $\mathbf{x}$. You start the iterative procedure with an initial guess of the input vector $\mathbf{x}$. $\mathbf{x}$ is then iteratively updated until the solution converges towards the target response vector $\mathbf{y}$.

For the iterative procedure to update $\mathbf{x}$, the coupled system of equations is linearized. Then the Newton-Raphson approach is applied. To linearize, the Taylor series expansion for $\mathbf{y}$ about $\mathbf{x}$ is applied:

$$\mathbf{y}\left(\mathbf{x}+\Delta\mathbf{x}\right)=\mathbf{y}\left(\mathbf{x}\right)+\mathbf{J}\Delta\mathbf{x}+O(\Delta\mathbf{x})^2 \tag{4035}$$

Considering only the first-order terms, Eqn. (4035) is rearranged to read:

$$\Delta\mathbf{x}=\mathbf{J}^{-1}\left\{\mathbf{y}\left(\mathbf{x}+\Delta\mathbf{x}\right)-\mathbf{y}\left(\mathbf{x}\right)\right\} \tag{4036}$$

where $\mathbf{J}$ is the Jacobian matrix of the dependent quantities, that is, of the response vector $\mathbf{y}$ with respect to the control inputs $\mathbf{x}$.

Eqn. (4036) is rewritten to:

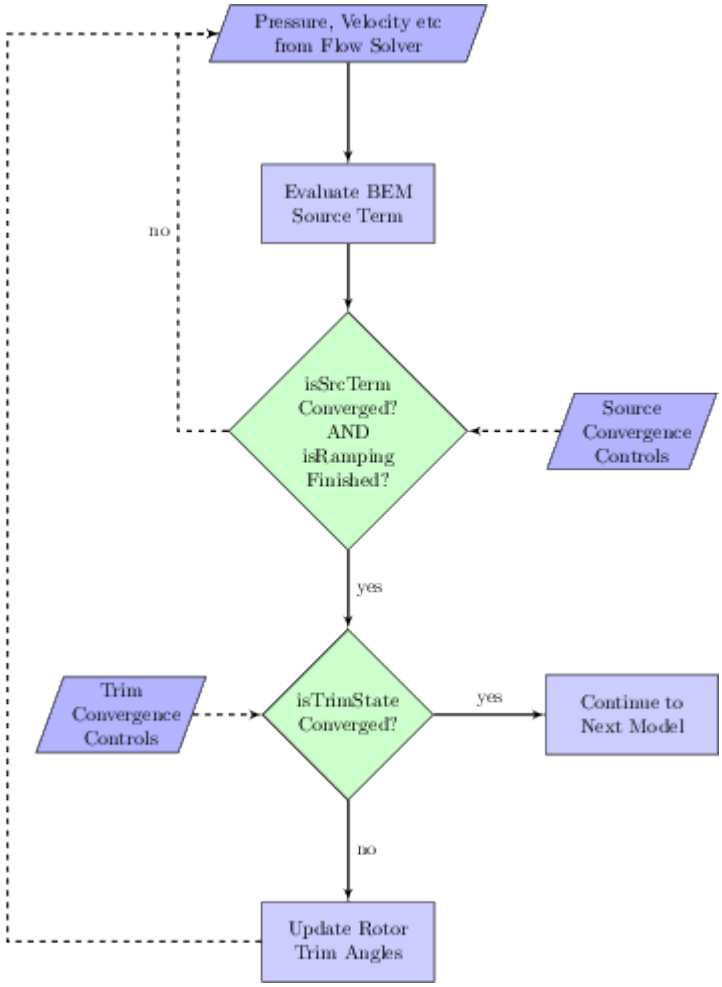
$$\begin{pmatrix}\Delta\theta_0\\ \Delta\theta_{1c}\\ \Delta\theta_{1s}\end{pmatrix}_{(i)}=\begin{pmatrix}\frac{\partial T}{\partial\theta_0} & \frac{\partial T}{\partial\theta_{1c}} & \frac{\partial T}{\partial\theta_{1s}}\\ \frac{\partial M_x}{\partial\theta_0} & \frac{\partial M_x}{\partial\theta_{1c}} & \frac{\partial M_x}{\partial\theta_{1s}}\\ \frac{\partial M_y}{\partial\theta_0} & \frac{\partial M_y}{\partial\theta_{1c}} & \frac{\partial M_y}{\partial\theta_{1s}}\end{pmatrix}_{(i)}^{-1}\times\left\{\begin{pmatrix}T\\ M_x\\ M_y\end{pmatrix}_{(d)}-\begin{pmatrix}T\\ M_x\\ M_y\end{pmatrix}_{(i)}\right\} \tag{4037}$$

In Eqn. (4037), the subscript *(i)* corresponds to the ith trim iteration and *d* corresponds to the desired target state of the rotor response vector $\mathbf{y}$.

The initial guess for $\mathbf{x}$, yields an initial $\mathbf{y}$, which is usually not equal to the target state of $\mathbf{y}_d$. To proceed with the trim iteration, the Jacobian matrix must be determined using a frozen flow field. To achieve this, each component of the control input vector is independently perturbed and the rotor response is recomputed. The partial derivatives are then computed using 2nd-order central differences. With available, the correction to the control input vector $\Delta\mathbf{x}$ is determined by solving Eqn. Eqn. (4036). The computed correction is under-relaxed to achieve numerical damping and is then used to update the control input vector. The procedure is repeated until the solution converges towards the target $\mathbf{y}$ within a specified tolerance.

Overview of the Source Term Update

The following flowchart provides an overview of the blade element source term update process. This procedure is triggered when the main flow solver is ready to accept source term contributions from optional models into the momentum equation. When the main flow related fields such as pressure and velocity are available, the blade element source term is evaluated.



Next, STAR-CCM+ checks if the blade element source term is converged up to the *Source Tolerance* (see User Guide > Modeling Physics > Virtual Disk Model > Blade Element Method > Setting the Source Convergence Control). Furthermore, STAR-CCM+ checks if the ramping of the rotation rate is finished. If either of the checks fails, the blade element source term is under-relaxed by the *Source Under Relaxation Factor* (see see User Guide > Modeling Physics > Virtual Disk Model > Blade Element Method > Setting the Source Convergence Control) and is added to the source term of the momentum equation. The solution process then continues with the next model. This step corresponds to the **no** branch of the first decision block in the flow chart above.

When the first decision block evaluates to **yes**, STAR-CCM+ checks if the main rotor response equals the target values. This check corresponds to the second decision block in the flow chart above and is interpreted as the convergence of the trim state. The trim state is converged if the user-specified *Trim Tolerance* is reached (see User Guide > Modeling Physics > Virtual Disk Model > Blade Element Method > Setting the Trim Convergence Control). If the trim state is converged, the solution process continues with the next model. If the trim state is not converged, the rotor trim angles are updated according to the numerical trim algorithm. The computed trim angles are under-relaxed by the *Trim Under Relaxation Factor* (see User Guide > Modeling Physics > Virtual Disk Model > Blade Element Method > Setting the Trim Convergence Control).

The source term computation is performed at each iteration. Therefore, any changes in the user setup resulting in a different flow field, influence the blade element method source term vector. It is important to

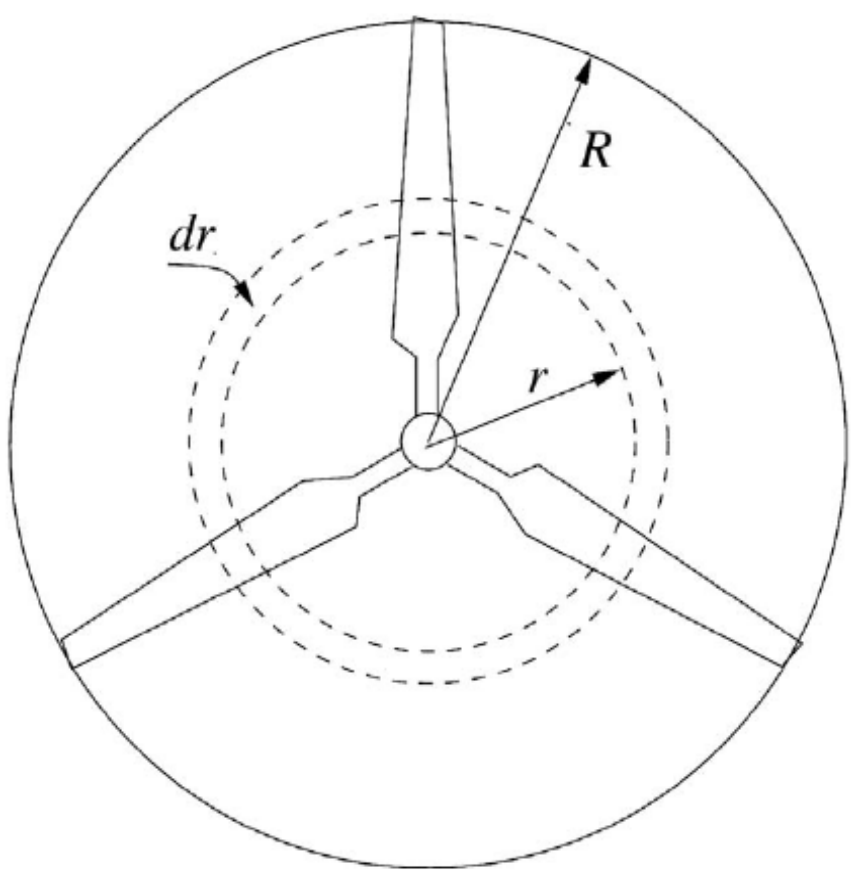
note that the trim algorithm is executed only after the blade element source term has converged to the specified level. If the external flow field is of highly unsteady nature, you are recommended to relax the convergence tolerance on the source term so that the trim angle update can spring into action. For extreme case, where the main rotor is operating in a wake and the desired flight condition is hover, you are advised to deactivate the source term convergence check by specifying a tolerance level of 100%. This setting ensures that the algorithm is executed at each iteration.

1D Momentum Method

A wind turbine extracts mechanical energy from the kinetic energy of the wind. It acts as an actuator disk, slowing down the wind speed from U_0 to U at the rotor plane, and further slowing it down to U_1 in the wake. Moving upstream to downstream, the velocity decreases and the streamlines diverge to comply with continuity.

The 1D momentum method adds a source term to the momentum equations to model the effects of a wind turbine. The source term accounts for both the axial and the tangential effects as given by the Ideal Horizontal Axis Wind Turbine (HAWT) with wake rotation. When wake rotation is included in the analysis, the induced velocity at the rotor consists of an axial component a and a tangential component a' acting in the rotor plane.

The formulation is based on the use of an annular stream tube with a radius r and a thickness dr. (Note that this is not the disk thickness.) This definition results in a cross-sectional area dA of $2\pi r dr$. The following figure shows the schematics and the notation that is used in the formulation:



Thrust Distribution

Total thrust of a wind turbine is given by:

$$T = \frac{1}{2}\rho_0 U^2{}_0 C_t \pi \left(R_0{}^2 - R_i{}^2\right) \tag{4038}$$

Thrust on an annular ring of width dr at a radius r is:

$$dT = \frac{1}{2}\rho_0 U^2{}_0 C_t 2\pi r dr \tag{4039}$$

The theory assumes an infinitesimally thin disk with zero thickness, whereas the source terms are added for a virtual disk of a finite thickness δ. Therefore, the elemental thrust ΔT is to be understood as thrust per Δr annulus per unit disk thickness δ:

$$\Delta T = \frac{1}{2}\rho_0 U^2{}_0 C_t 2\pi r \Delta r \tag{4040}$$

Using Eqn. (4038), this can be rewritten as:

$$\Delta T = T \frac{2r\Delta r}{\left(R_0{}^2 - R_i{}^2\right)} \tag{4041}$$

Thrust per unit volume within the annulus thus reads:

$$\frac{\Delta T}{\Delta V} = \frac{1}{2} \cdot \frac{\rho_0 U^2{}_0 C_t}{\delta} = \frac{T}{\delta\pi\left(R_0{}^2 - R_i{}^2\right)} \tag{4042}$$

where $\delta\pi\left(R_0{}^2 - R_i{}^2\right)$ is the geometrical volume of the disk.

Thrust per unit volume is independent of r. Thrust depends on r quadratically, but the volume of each ring also increases quadratically with r, so thrust per unit volume is constant. For each cell within a ring:

$$T_{cell} = \frac{\Delta T}{\Delta V} V_{cell} \tag{4043}$$

Eqn. (4043) needs to be scaled, because not all cells are fully within the disk. The scaling factor is:

$$\frac{V_{geom}}{\sum_{all\ cells} V_{cell}}$$

where

$$V_{geom} = \pi\left(R_0{}^2 - R_i{}^2\right)$$

The thrust per cell is given by:

$$T_{cell} = T \cdot \frac{V_{cell}}{\sum_{all\ cells} V_{cell}} \tag{4044}$$

T_{cell} corresponds to the axial force F_{axial} for a given cell, whose cell center lies within the virtual disk.

Torque Distribution

The torque distribution corresponds to the calculation of the tangential force. With a given velocity at the inflow plane U_0, the power P delivered by the wind turbine can be extracted from the user specified power curve table or polynomial. From this value, the torque is evaluated as:

$Q = \frac{P}{\Omega}$

where Ω is the rotation rate. This torque is distributed as tangential force over the disk cells.

The elemental torque dQ for an annular ring of width dr and thickness δ are obtained by applying the conservation of angular momentum. On an annular ring element, this gives:

$$dQ = d\dot{m}(\omega r)(r) = (\rho U 2\pi r dr)(\omega r)(r) \tag{4045}$$

Since $U = U_0(1-a)$ and $\mathrm{a}' = \omega/(2\Omega)$, the expression for the elemental torque reduces to:

$$dQ = 4\mathrm{a}'(1-a)\frac{1}{2}\rho U_0 \Omega r^2 2\pi r dr \tag{4046}$$

Integrating Eqn. (4046):

$$\int \mathrm{d}\,Q = 4\mathrm{a}'(1-a)\frac{1}{2}\rho U_0 \Omega 2\pi r \int_{R_i}^{R_0} r^3\,\mathrm{d}\,r \tag{4047}$$

The torque now is:

$$Q = \mathrm{a}'(1-a)\frac{1}{2}\rho U_0 \Omega \pi \left({R_0}^4 - {R_i}^4\right) \tag{4048}$$

To derive an expression for torque per unit of annular ring volume, Eqn. (4046) is rewritten by use of Eqn. (4048):

$$\Delta Q = \frac{4Qr^3\Delta r}{{R_0}^4 - {R_i}^4} \tag{4049}$$

Torque per unit volume of annular ring $\Delta V = 2\pi\delta r \Delta r$ is then:

$$\frac{\Delta Q}{\Delta V} = \frac{2Qr^2}{\pi\delta \cdot \left({R_0}^4 - {R_i}^4\right)} \tag{4050}$$

Torque for each cell within the annular ring is:

$$Q_{cell} = \frac{\Delta Q}{\Delta V} \cdot V_{cell} = \frac{2Qr_{cell}^2 V_{cell}}{\pi\delta \cdot \left({R_0}^4 - {R_i}^4\right)} \tag{4051}$$

Scaling needs to be applied, because $\sum V_{cell} \neq V_{geom}$. The scaling factor reads:

$$scf = \frac{V_{geom} \cdot \frac{1}{2}\left({R_0}^2 + {R_i}^2\right)}{\sum_{all\ cells} {r_{cell}}^2 V_{cell}} \tag{4052}$$

With scaling applied:

$$Q_{cell} = \frac{2Qr_{cell}^2 V_{cell}}{\pi\delta \cdot \left(R_0{}^4 - R_i{}^4\right)} \cdot scf \tag{4053}$$

With $V_{geom} = \pi\left(R_0{}^2 - R_i{}^2\right)\delta$, Eqn. (4053) can be further simplified:

$$Q_{cell} = \frac{Qr_{cell}^2 V_{cell}}{\sum_{all\ cells} r_{cell}{}^2 V_{cell}} \tag{4054}$$

The tangential force $F_{\text{tangential}}$ for a given cell whose cell center lies inside the virtual disk is given by:

$$F_{\text{tan}gential} = \frac{Qr_{cell}^2 V_{cell}}{\sum_{all\ cells} r_{cell}{}^2 V_{cell}} \tag{4055}$$

The axial and the tangential forces are used to assemble a source term vector with respect to the laboratory coordinate system. This source term is then added to the momentum equations.

Fan Modeling

STAR-CCM+ provides several approaches to model axial and radial fans in terms of fan performance characteristics. These approaches do not take into account the detailed geometry of a fan.

The simplified fan models are particularly useful when the exact geometry of a fan is not known or not important, but the fan performance curve is available. Using the imported fan performance curve, the lumped fan models relate the pressure rise to flow rate or velocity. Depending on the model, the fan is modeled as a plane surface or a volumetric region that induces a pressure jump as the flow passes through it. In addition, you can model the swirl velocity downstream of the fan.

You typically apply these models when you are interested in predicting the pressure rise through a fan in your system. An example for such an application is the thermal simulation of components on a motherboard where the cooling fan is just one component of many. The accurate resolution of the cooling fan on the board would require a high cell count and increase the analysis time significantly, therefore, a lumped approach is preferred.

Contents:

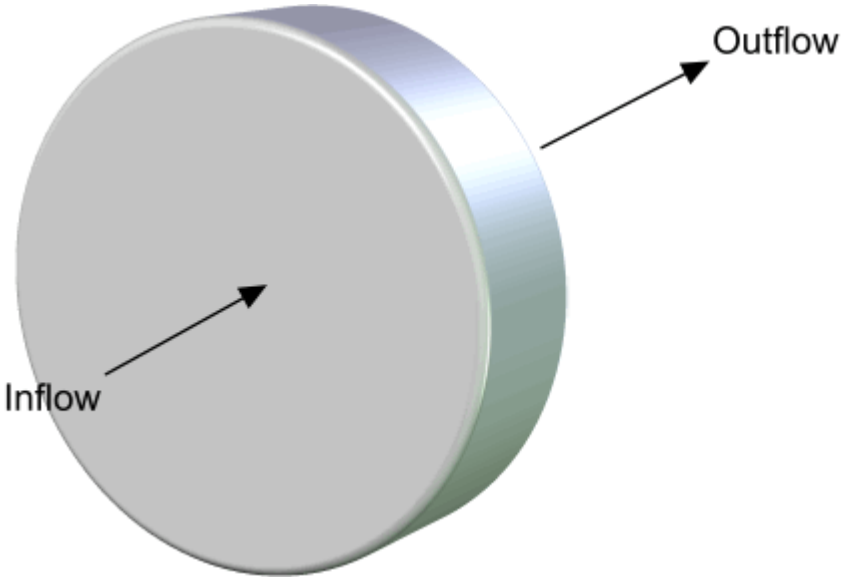
Axial Fans
Impellers and Centrifugal Fans

Axial Fans

STAR-CCM+ provides two approaches to modeling axial fans: the fan momentum source and the fan interface. The primary objective of these approaches is to model the increase in pressure across the fan.

Additionally, these approaches can model the swirl that is introduced by the fan. None of these approaches take into account the detailed geometry of fan blades. Generally, the fan interface is more robust, and it is usually recommended over the fan momentum source.

The following diagram shows a schematic of an axial fan. For an axial fan, the inflow and the outflow direction are the same, which is along the fan axis.



Fan Interface

The fan interface is a simple model that represents an axial fan by a zero-thickness interface and imposes a pressure jump across the interface. The fan interface can also add swirl to the flow downstream of the fan.

Pressure Jump Modeling

The pressure jump across the interface is obtained from a user-specified fan curve that plots pressure jump as a function of flow rate or flow velocity. The pressure jump ΔP that is read from the fan curve can be interpreted in two ways. One option is:

$$\Delta P = P_{s,down} - P_{t,up} \tag{4056}$$

where $P_{s,down}$ is the static pressure downstream of the fan and $P_{t,up}$ is the total pressure upstream of the fan. This option appears as **Standard** in STAR-CCM+.

The other option is:

$$\Delta P = P_{s,down} - P_{s,up} \tag{4057}$$

where $P_{s,up}$ is the static pressure upstream of the fan. This option appears as **Static to Static** in STAR-CCM+.

The pressure jump is usually imposed locally, which means that a separate pressure jump is applied to each pair of faces on the fan interface. STAR-CCM+ also provides the option to apply a single pressure jump to all face pairs on the fan interface.

Swirl Modeling

Assuming that the axial velocity, the radial velocity, and the density are unchanged across the fan, the total pressure jump ΔP_t across the fan is related to the static pressure jump ΔP_s, which is derived from the performance curve, as:

$$\begin{aligned} \Delta P_t &= P_{t,down} - P_{t,up} \\ &= \Delta P_s + \frac{1}{2}\rho\left[v_{t,\mathrm{down}}^2 - v_{t,\mathrm{up}}^2\right] \end{aligned} \tag{4058}$$

where $P_{t,down}$ is the total pressure downstream of the fan interface and v_t is the swirl velocity, which corresponds to the tangential component of the velocity vector.

The fan efficiency is given as:

$$\begin{aligned} \eta &= \frac{\text{Power Imparted to Air}}{\text{Power Consumed}} \\ &= \frac{\dot{V}\,\Delta\,P_t}{\dot{m}\Omega r\,\Delta\,v_t} \end{aligned} \tag{4059}$$

where $\dot{V}$ is the volumetric flow rate, $\dot{m}$ is the mass flow rate, Ω is the fan rotation rate, and r is the radial distance from the axis of the fan.

With:

$$\rho = \frac{\dot{m}}{\dot{V}}$$

The swirl velocity difference is then obtained as:

$$\Delta v_t = v_{t,down} - v_{t,up} = \frac{\Delta\,P_t}{\eta\Omega r\rho} \tag{4060}$$

Rearranging Eqn. (4060) and inserting Eqn. (4058) gives the following quadratic equation in $v_{t,down}$:

$$\frac{1}{2}v_{t,down}^2 - \eta\Omega r v_{t,down} + \left(\frac{\Delta\,P_s}{\rho} + \eta\Omega r v_{t,up} - \frac{1}{2}v_{t,up}^2\right) = 0 \tag{4061}$$

That is solved for $v_{t,down}$:

$$v_{t,down} = \eta\Omega r \pm \sqrt{\left(\eta\Omega r\right)^2 - 2\left[\frac{\Delta\,P_s}{\rho} + \eta\Omega r v_{t,up} - \frac{1}{2}v^2{}_{t,up}\right]} \tag{4062}$$

STAR-CCM+ uses only the negative square root. In the case of a complex root, STAR-CCM+ uses only the real part.

Fan Momentum Source

The fan momentum source employs an actuator disk type of methodology. Actuator disk models model the fan as a volumetric region and introduce sources in the momentum equation to model pressure rise and swirl. Various actuator disk models differ in how the sources are computed. The fan momentum source approach does not take into account the detailed geometry of the fan blades when computing source terms.

The fan momentum source term is the sum of the tangential and the axial force components $\mathbf{f}_t$ and $\mathbf{f}_{\mathrm{axial}}$:

$$\mathbf{f}_{\mathrm{fan}} = \mathbf{f}_t + \mathbf{f}_{\mathrm{axial}} \tag{4063}$$

The fan momentum source $\mathbf{f}_{\mathrm{fan}}$ enters the momentum equation Eqn. (1462) as volumetric body force.

Axial Component of Source Term for Pressure Rise Modeling

The axial component of the source term that is added to the momentum equation controls the pressure rise through the fan region.

At each time step, the net volume flow rate is calculated from the upstream boundary to the downstream boundary of the fan region, and the corresponding pressure rise is obtained from the fan curve. This pressure rise is the target pressure rise ΔP_{target} that the fan must achieve for the current mass flow rate.

The pressure rise is computed over the entire length of the fan region, therefore, the target pressure rise per unit length is given by:

$$\Delta \widehat{P}_{\text{target}} = \frac{\Delta\ \mathrm{P}_{\text{target}}}{L_{\text{FanRegion}}}$$

The axial component of the momentum equation source term is calculated as:

$$\mathbf{f}_{\text{axial}} = \frac{\Delta\ \mathrm{P}_{\text{target}}}{\Delta\ \mathrm{P}_{\text{actual}}} \cdot \widehat{P}_{\text{target}} \cdot \widehat{\mathbf{X}} \tag{4064}$$

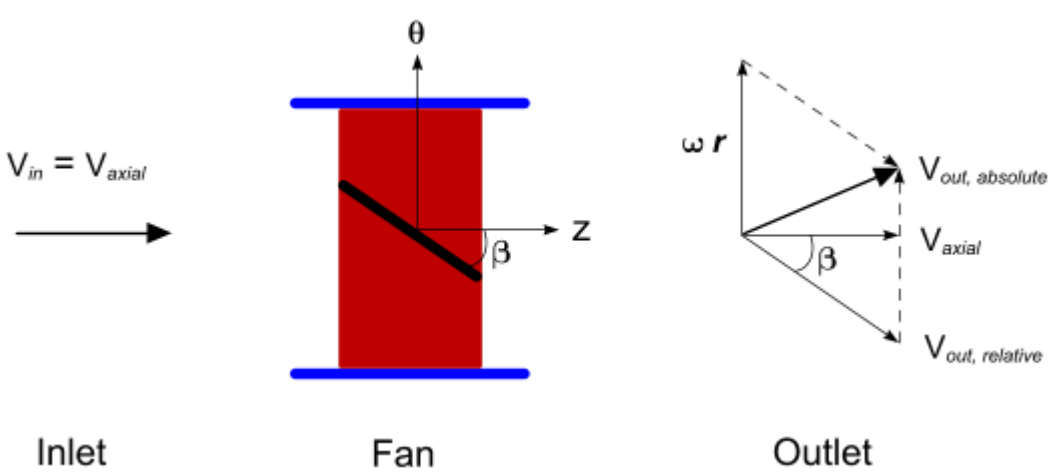
where $\widehat{\mathbf{X}}$ is a unit vector in the axial direction of the fan and:

$$\Delta\ \mathrm{P}_{\text{actual}} = \overline{P}_{\text{down}} - \overline{P}_{\text{up}} \tag{4065}$$

where $\overline{P}_{\text{down}}$ is the pressure that is averaged over the boundary downstream of the fan region and $\overline{P}_{\text{up}}$ is the pressure that is averaged over the boundary upstream of the fan region.

Tangential Component of Source Term for Swirl Modeling

To understand the forces that the fan blades impart to the fluid, consider the velocity vector diagram below. The fan axis is aligned in the z-direction. A single fan blade is shown in the fan volume.



Fluid approaches the fan with velocity v_{in}, which is assumed to be uniform and to have no tangential component $(v_{\text{in}} = v_{\text{axial}})$. The fan rotates in the tangential θ-direction with the angular velocity ω. The fan blades are at an angle β to the fluid flow and are assumed to have no curvature.

The fluid velocity leaving the fan, $v_{\text{out, absolute}}$, is the vector sum of the relative velocity $v_{\text{out, relative}}$ (assumed to be parallel to the blade angle) and the rotational velocity ωr, imparted to the fluid by the fan blades.

From the previous vector diagram, the tangential velocity component can be calculated as:

$$v_{t,\,\mathrm{target}} = \omega r - v_{\mathrm{axial}} \tan \beta \tag{4066}$$

where r is the radius measured from the center of the fan.

$\overline{v}_{t,\,\mathrm{target}}$ is calculated as the average of the target tangential velocity $v_{t,\,\mathrm{target}}$ at the downstream fan boundary. The average of the actual tangential velocity is calculated as:

$$\overline{v}_{t,\,\mathrm{actual}} = \left(\overline{\mathbf{v} \cdot \widehat{\boldsymbol{\theta}}}\right)_{\mathrm{down}} \tag{4067}$$

where $\widehat{\boldsymbol{\theta}}$ is a unit vector in the tangential direction.

A numerical factor σ is defined as:

$$\sigma_{n+1} = \sigma_n \cdot \frac{\overline{v}_{t,\,\mathrm{target}}}{\overline{v}_{t,\,\mathrm{actual}}} \tag{4068}$$

The numerical factor σ drives the code to achieve the $v_{t,\,\mathrm{target}}$, and has units of $\mathrm{Ns/m^4}$.

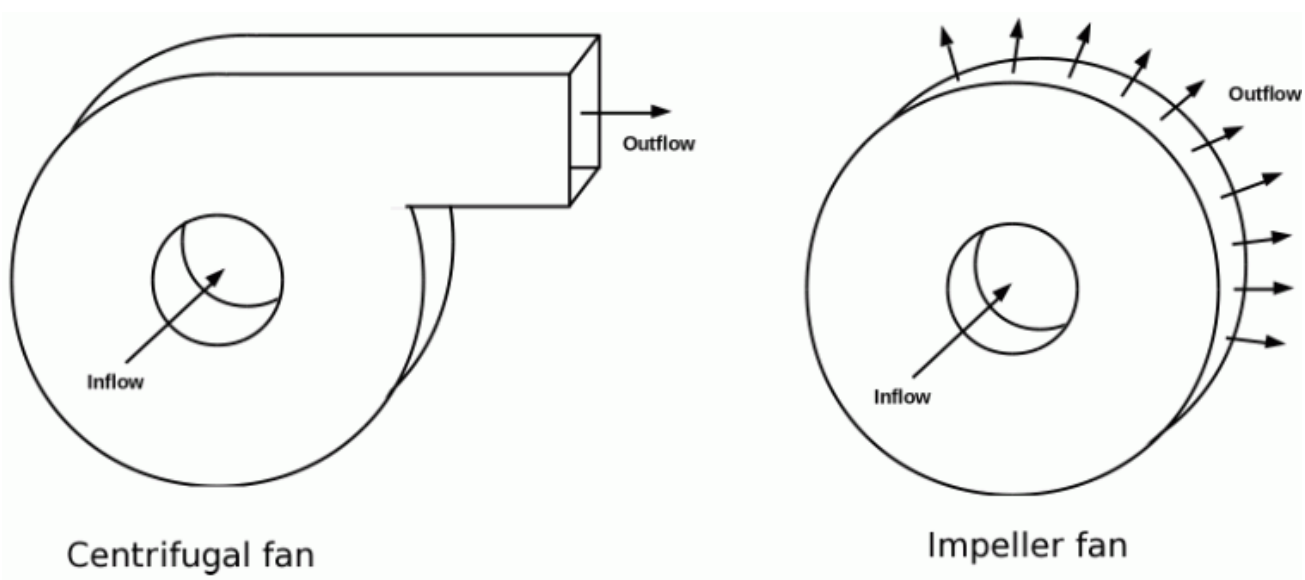
Finally, the tangential part of the source term addition to the momentum equation is:

$$\mathbf{f}_t = \nu^2 \cdot \sigma \cdot v_{t,\,\mathrm{target}} \cdot \widehat{\boldsymbol{\theta}} \tag{4069}$$

where ν is the normalized distance of the current cell from the fan upstream boundary, calculated along the fan axis. The term ν ensures that the swirl increases gradually in the fan region.

Impellers and Centrifugal Fans

For axial fans, the outflow direction is the same as the inflow direction. However, for impellers and centrifugal fans, the outflow direction is not the same as the inflow direction. These types of fans are modeled using the blower interface. The blower interface implementation in STAR-CCM+ is the same for both impellers and centrifugal fans, except that swirl at the outflow is allowed for impellers only.



Blower Interface

The blower interface is a non-conformal interface that connects the inlet and the outlet of the impeller or the centrifugal fan. Unlike an axial fan, an impeller or a centrifugal fan has the inlet and the outlet located in different places and oriented in different directions. The blower interface is a simple model that imposes a pressure jump across the inlet and the outlet of the fan. The geometry of the impeller or the centrifugal fan between the inlet and the outlet is not modeled.

Pressure Jump Modeling

The pressure jump is obtained from a user-specified fan curve that plots pressure jump as a function of volumetric flow rate. The pressure jump that is read from the fan curve is interpreted as:

$$\Delta P = P_{s,\,\mathrm{down}} - P_{t,\,\mathrm{up}} \tag{4070}$$

where $P_{s,\,\mathrm{down}}$ is the static pressure downstream of the blower interface and $P_{t,\,\mathrm{up}}$ is the total pressure upstream of the blower interface.

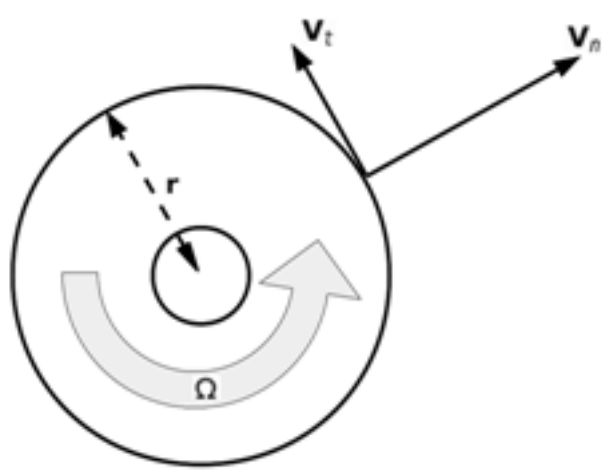
The pressure on the faces of the fan inlet are computed as:

$$P_{f,\,\mathrm{in}} = \overline{P}_{\mathrm{out}} - \Delta \mathrm{P}_{\mathrm{local}} - \frac{1}{2}\rho v_{n,\,\mathrm{local}}^2 \tag{4071}$$

where $\Delta P_{\mathrm{local}}$ is a local pressure jump that is based on the local normal velocity $v_{n,\,\mathrm{local}}$ and $\overline{P}_{\mathrm{out}}$ is the averaged pressure on the fan outlet.

Swirl Modeling

The figure below illustrates the computation of the swirl (tangential) component of the velocity at the blower outlet:



The velocity tangential to the impeller outlet is given by:

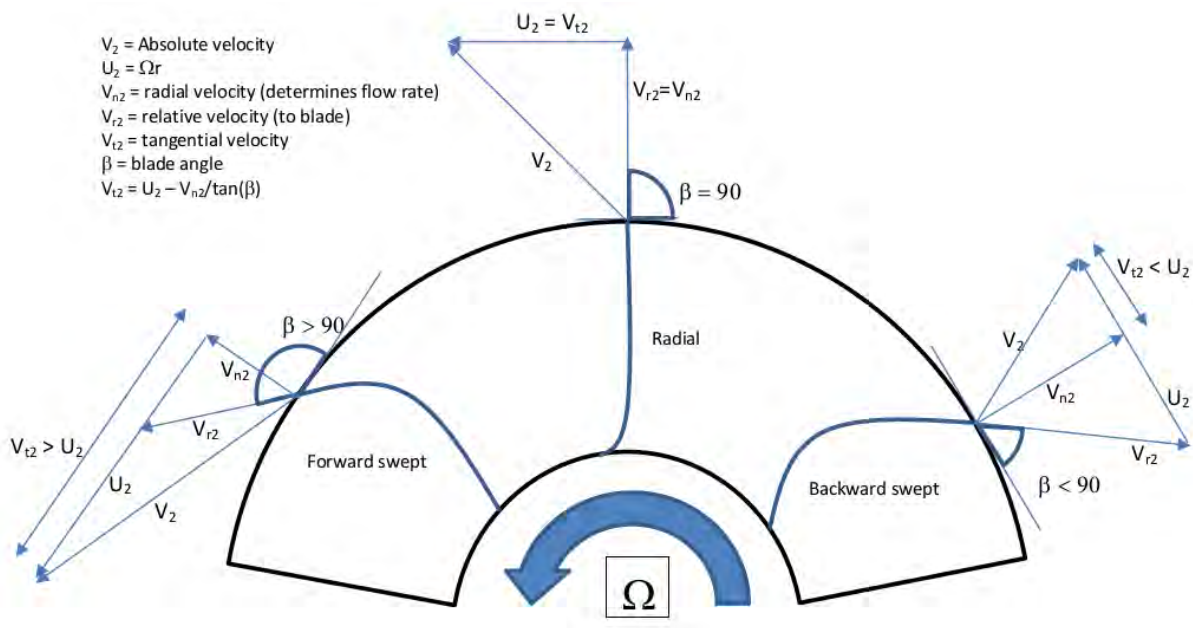
$$v_t = \Omega \mathrm{r} - \frac{v_n}{\tan\beta} \tag{4072}$$

where:

Ω	fan rotation rate

r	radial distance from the fan axis to the impeller outlet
β	blade angle
$v_n = \frac{\dot{m}}{\rho A}$	velocity normal to the impeller outlet
$\dot{m}$	mass flow rate
A	circumferential area of the impeller

The following figure illustrates the amount of tangential velocity that is added for different values of blade angle:



~Harmonic Balance~

For this release, see *User Guide* > *Modeling Physics* > *Modeling with Harmonic Balance* > *Harmonic Balance Formulation*.

Theory guide content is planned for a future release.

Adjoint

The adjoint method is an efficient means to predict the influence of many input parameters on some engineering quantities of interest in a simulation.

In other words, it provides the sensitivity of objectives (cost functions) with respect to input parameters such as design variables or values of boundary conditions. Examples of questions to which the adjoint method can help provide answers are:

- What effect does the shape of a duct have on the pressure drop (Sensitivity with Respect to Design Parameters)?
- What is the influence of inlet conditions on flow uniformity at the outlet (Sensitivity with Respect to Boundary Parameters)?
- What areas on the surface have the largest impact on the engineering quantity of interest (Surface Sensitivity)?
- What is the effect of the numerical error on the engineering quantity (Adjoint Error Estimation)?

An adjoint analysis is performed in two major steps. First, you calculate the steady-state solution for the physical phenomena of interest using the coupled flow model, or both the coupled flow and coupled energy models. The result of this simulation is referred to as the "primal solution". In the second step, you evaluate the adjoint of the simulated physics based on the previously obtained primal solution.

Simulation Inputs and Outputs

The primal simulation is performed in three steps (where an initial mesh, $\mathbf{X}^0$, is assumed to exist):

1. Design parameters prescription:

   Definition of how the shape of the initial mesh can be deformed by changing the values of design parameters, $\mathbf{D}$. This definition defines a relationship between the design variables and the resulting mesh with points $\mathbf{X}$. These mesh points can be the result of a mesh deformation or another mesh generation process.
2. Flow and energy solution:

   Iterative solution of the governing equations of the physics being simulated, denoting its solution as $\mathbf{Q}$.
3. Solution analysis:

   Creation of report objectives, $\mathbf{L}$, to evaluate physical quantities on parts, such as the drag force on a vehicle.

Following these steps, the calculation of the objectives $\mathbf{L}$ is performed in a sequence of operations $(\mathbf{X}(\mathbf{D}); \mathbf{Q}(\mathbf{X}); \mathbf{L}(\mathbf{Q}, \mathbf{X}))$ whose input is the set of prescribed design parameters $\mathbf{D}$ and output is $\mathbf{L}$.

The adjoint of the simulated physics is then calculated in the reverse order to which the primal simulation is performed.

Contents:

Sensitivity with Respect to Design Parameters
Sensitivity with Respect to Boundary Parameters
Surface Sensitivity
Adjoint Error Estimation

Sensitivity with Respect to Design Parameters

The optimization of objectives on the basis of design modifications requires the sensitivities of the objectives with respect to the design parameters, that is $\frac{d\mathbf{L}}{d\mathbf{D}}$.

The derivative $\frac{d\mathbf{L}}{d\mathbf{D}}$ is given by the chain of the Jacobians of each operation in the sequence $(\mathbf{X}(\mathbf{D}); \mathbf{Q}(\mathbf{X}); \mathbf{L}(\mathbf{Q}, \mathbf{X}))$:

$$\frac{d\mathbf{L}}{d\mathbf{D}} = \left[\frac{\partial \mathbf{L}}{\partial \mathbf{X}} + \frac{\partial \mathbf{L}}{\partial \mathbf{Q}}\frac{\partial \mathbf{Q}}{\partial \mathbf{X}}\right]\frac{d\mathbf{X}}{d\mathbf{D}} \tag{4073}$$

Consider n design variables $\mathbf{D}$ and a mesh $\mathbf{X}$ with m points. The Jacobian of the operation $\mathbf{X}(\mathbf{D})$ is the matrix of all first-order partial derivatives of this operation:

$$\frac{d\mathbf{X}}{d\mathbf{D}} = \begin{bmatrix} \frac{\partial X_1}{\partial D_1} & \cdots & \frac{\partial X_1}{\partial D_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial X_m}{\partial D_1} & \cdots & \frac{\partial X_m}{\partial D_n} \end{bmatrix} = J_{\mathbf{X}} \tag{4074}$$

where each column of this matrix is a tangent and each row is a gradient of the Jacobian.

For a large number of design variables $\mathbf{D}$, the computation of Eqn. (4073) becomes expensive with respect to memory. To alleviate this problem the derivative of the system can be composed from tangents of $J_{\mathbf{X}}$. That is, STAR-CCM+ computes:

$$\frac{d\mathbf{L}}{dD_i} = \left[\frac{\partial \mathbf{L}}{\partial \mathbf{X}} + \frac{\partial \mathbf{L}}{\partial \mathbf{Q}}\frac{\partial \mathbf{Q}}{\partial \mathbf{X}}\right]\frac{d\mathbf{X}}{dD_i} \tag{4075}$$

for all columns of $J_{\mathbf{X}}$.

In alleviating one problem, this approach introduces another, in that computational time grows linearly with the number of design variables $\mathbf{D}$. Whilst for many engineering problems the number of design variables $\mathbf{D}$ is large, which would rule out the tangent approach, there are often only a few outputs $\mathbf{L}$ which are required. If the tangents approach were to be inverted, then this would yield an efficient way of computing the derivatives of the whole system. This goal can be achieved by taking the transpose of the derivative of the system for a particular gradient of $J_{\mathbf{L}}$:

$$\frac{dL_j{}^T}{d\mathbf{D}} = \frac{d\mathbf{X}}{d\mathbf{D}}^T\left[\frac{\partial L_j{}^T}{\partial \mathbf{X}} + \frac{\partial \mathbf{Q}}{\partial \mathbf{X}}^T\frac{\partial L_j{}^T}{\partial \mathbf{Q}}\right] \tag{4076}$$

Eqn. (4076) represents the basic form of how the actual final sensitivity $\frac{d\mathbf{L}}{d\mathbf{D}}$ is built up by the adjoint of the sequence of steps listed in the introduction.

The steps in solving Eqn. (4076) are:

1. Computing the sensitivity of the report objectives with respect to the solution $\frac{\partial L_j}{\partial \mathbf{Q}}^T$. This solution analysis also computes any direct sensitivities with respect to the mesh, that is $\frac{\partial L_j}{\partial \mathbf{X}}^T$. See Adjoint of Solution Analysis.
2. Evaluating the sensitivity of the report objectives with respect to the mesh, that is the rightmost term in Eqn. (4076):

   $$\frac{dL_j}{d\mathbf{X}}^T = \frac{\partial L_j}{\partial \mathbf{X}}^T + \frac{\partial \mathbf{Q}}{\partial \mathbf{X}}^T \frac{\partial L_j}{\partial \mathbf{Q}}^T$$

   See Adjoint of Flow and Energy Solution.
3. Performing the adjoint-differentiated mesh deformation algorithm that evaluates $\frac{dL_j}{d\mathbf{D}}^T$, that is the product of $\frac{d\mathbf{X}}{d\mathbf{D}}^T$ and the result of the previous step $\frac{dL_j}{d\mathbf{X}}^T$. The evaluated term describes the sensitivity of the report objectives with respect to the mesh design parameters. Adjoint of Mesh Deformation provides the derivation of the adjoint of the Radial Basis Function morpher.

Contents:

Adjoint of Solution Analysis
Adjoint of Flow and Energy Solution
Adjoint of Mesh Deformation

Adjoint of Solution Analysis

The computation of the sensitivity of the report objective with respect to its dependent variables, $\frac{\partial L}{\partial \mathbf{Q}}^T$, provides the input to the adjoint of the governing physics solver.

Primal

The last step in the primal analysis becomes the first step in the adjoint analysis. Once a solution of the system of equations is obtained, the engineering quantities of interest—such as lift and drag forces on vehicles, pressure drop along ducts—are computed. The quantity that you want to reduce or increase is commonly termed as the cost function or objective in the context of optimization.

Suppose the objective, L, is a surface integral report of the tangential speed field, v_t, at the outlet boundary of a duct:

$$L = \int_A v_t dA \approx \sum_f v_{tf} A_f \tag{4077}$$

where

$$v_t = \|\mathbf{v} - (\mathbf{v} \cdot \mathbf{n})\mathbf{n}\| \tag{4078}$$

and $\mathbf{n}$ is the unit face normal.

The tangential speed would be supplied as a user defined field function with the definition

```
mag($$Velocity-dot($$Velocity, unit($$Area))*unit($$Area))
```

Writing the objective in the form $L(\mathbf{Q},\mathbf{X})$, where v is a component of $\mathbf{Q}$ and treating A as being synonymous with $\mathbf{X}$, its evaluation would be performed in two steps:

- Calculating the user defined field v_t
- Evaluating the surface integral of that field over all the specified parts.

Thus the objective is of the form

$$L\left(v_t(\mathbf{Q},\mathbf{X}),\mathbf{X}\right) \tag{4079}$$

Adjoint

Applying the chain rule to Eqn. (4079) gives:

$$\frac{\partial L}{\partial \mathbf{Q}} = \frac{\partial L}{\partial v_t}\frac{\partial v_t}{\partial \mathbf{Q}} \tag{4080}$$

$$\frac{\partial L}{\partial \mathbf{X}} = \frac{\partial L}{\partial v_t}\frac{\partial v_t}{\partial \mathbf{X}} + \frac{\partial L}{\partial \mathbf{X}} \tag{4081}$$

The existence of both of these sensitivities is due to the choice of report selected (in this example the surface integral) and the field function supplied to the report. Simpler report objectives may not have a direct dependency on the mesh (second term in Eqn. (4081)), such as a Sum report. Also, a simpler field function may not have a direct dependency on the solution of the transport equations, such as a geometric constraint.

Taking the transpose of Equation Eqn. (4080) and Eqn. (4081), the derivatives required by the first step of the adjoint algorithm are

$$\frac{\partial L^T}{\partial \mathbf{Q}} = \frac{\partial {v_t}^T}{\partial \mathbf{Q}}\frac{\partial L^T}{\partial v_t} \tag{4082}$$

$$\frac{\partial L^T}{\partial \mathbf{X}} = \frac{\partial {v_t}^T}{\partial \mathbf{X}}\frac{\partial L^T}{\partial v_t} + \frac{\partial L^T}{\partial \mathbf{X}} \tag{4083}$$

Both terms are computed once at the beginning of the adjoint algorithm and contribute to the right-hand side of the adjoint flow and energy linear system solvers, and to the final metrics sensitivity, respectively.

Adjoint of Flow and Energy Solution

The fundamental task in solving the adjoint of the modeled physical phenomena is to compute the cost function sensitivity with respect to the mesh, $\frac{dL^T}{d\mathbf{X}}$.

The cost function sensitivity with respect to the mesh is the product of the two rightmost terms in Eqn. (4076):

$$\frac{dL^T}{d\mathbf{X}} = \frac{\partial L^T}{\partial \mathbf{X}} + \frac{\partial \mathbf{Q}^T}{\partial \mathbf{X}}\frac{\partial L^T}{\partial \mathbf{Q}} \tag{4084}$$

where the calculation of this sensitivity takes place in two steps:

1. Iteratively solving the adjoint flow solution, providing the sensitivity of the report objective with respect to the residuals of the governing equations, $\frac{dL^T}{d\mathbf{R}}$.

2. Computing the product of the residuals with respect to the mesh, $\frac{\partial R^T}{\partial \mathbf{X}}$, with the adjoint flow solution and adding it to the explicit mesh dependency of the objective, $\frac{\partial L^T}{\partial \mathbf{X}}$. This second step is performed as a one-off operation after the adjoint flow solution is converged.

These steps require the adjoint derivation of the implicit coupled solver. The derivation of the coupled flow model is described below. The derivation of the coupled solid energy model follows a similar pattern.

Primal

The aim of the iterative scheme of the coupled solver is to drive the residuals, $\mathbf{R}(\mathbf{Q},\mathbf{X})$, of the discretized Navier-Stokes equations to zero and obtain a converged solution, $\mathbf{Q} = [p, \mathbf{v}, T]^T$. This solution can be written in the form $\mathbf{Q}\left(\mathbf{Q}^0, \mathbf{X}\right)$, where $\mathbf{Q}^0$ represents the initial and boundary conditions.

The preconditioned Navier-Stokes equations are described as:

$$\Gamma \frac{\partial}{\partial t} \int_V \mathbf{Q} dV + \oint [\mathbf{F} - \mathbf{G}] \cdot dA = \int_V \mathbf{H} dV \tag{4085}$$

where Γ is the preconditioning matrix, $\mathbf{F}$ and $\mathbf{G}$ are the inviscid and viscous fluxes, respectively, and $\mathbf{H}$ is the body forces. Its solution is iteratively updated by solving the system of equations

$$[\mathbf{P}]\delta\mathbf{Q} = -\mathbf{R} \tag{4086}$$

and updating $\mathbf{Q}$ with $\delta\mathbf{Q} + \mathbf{Q}^{old}$. The matrix $[\mathbf{P}]$ represents the first-order linearization of the residuals with respect to $\mathbf{Q}$, defined as:

$$\left[\mathbf{P}\right] = \frac{V}{\delta t}\Gamma + \sum_j^{N_f} \left(\frac{\partial \mathbf{F}_j}{\partial \mathbf{Q}} - \frac{\partial \mathbf{G}_j}{\partial \mathbf{Q}} \right) A_j \tag{4087}$$

and the residual

$$\mathbf{R} = \sum_j^{N_f} \left(\mathbf{F}_j - \mathbf{G}_j \right) A_j - \mathbf{H}V \tag{4088}$$

Adjoint

Applying the chain rule to the residual at the converged state gives:

$$\frac{d\mathbf{R}}{d\mathbf{X}} = \left[\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\right] \frac{\partial \mathbf{Q}}{\partial \mathbf{X}} + \frac{\partial \mathbf{R}}{\partial \mathbf{X}} = 0 \tag{4089}$$

where the term $\left[\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\right]$ includes 2nd order terms.

The formulation of $\frac{\partial \mathbf{Q}^T}{\partial \mathbf{X}}$ in Eqn. (4084) may begin with taking the transpose of Eqn. (4089) and rearranging the terms as:

$$\frac{\partial \mathbf{Q}}{\partial \mathbf{X}}^T = -\frac{\partial \mathbf{R}}{\partial \mathbf{X}}^T \left[\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\right]^{-T} \tag{4090}$$

As the inverse calculation for $\left[\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\right]^{-T}$ is computationally expensive, Eqn. (4090) is multiplied with the previously calculated result of $\frac{\partial L}{\partial \mathbf{Q}}^T$ (see Solution Analysis):

$$\frac{dL}{d\mathbf{X}}^T = \frac{\partial L}{\partial \mathbf{X}}^T - \frac{\partial \mathbf{R}}{\partial \mathbf{X}}^T \left[\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\right]^{-T} \frac{\partial L}{\partial \mathbf{Q}}^T \tag{4091}$$

where $\frac{\partial \mathbf{R}}{\partial \mathbf{X}}^T$ is known because $\mathbf{R}$ is known. The last two terms define the adjoint solution variable, $\Lambda_{\mathbf{Q}}$:

$$\Lambda_{\mathbf{Q}} = \left[\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\right]^{-T} \frac{\partial L}{\partial \mathbf{Q}}^T \tag{4092}$$

which can be interpreted as $\frac{dL}{d\mathbf{R}}^T$.

To avoid calculating and assembling a large, complex matrix, the adjoint is cast as a defect-correction equation:

$$[\mathbf{P}]^T \delta\Lambda_{\mathbf{Q}} = \frac{\partial L}{\partial \mathbf{Q}}^T - \left[\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\right]^T \Lambda_{\mathbf{Q}} \tag{4093}$$

In this form, only matrix vector products of the Jacobian with the adjoint variable are required, which can be built up incrementally in a manner analogous to the primal.

Adjoint of Mesh Deformation

The ultimate sensitivities of a given objective, or cost function, are those with respect to the design parameters, that is $\frac{dL}{d\mathbf{D}}$. These sensitivities are typically used to solve a shape optimization problem.

Design parameters, $\mathbf{D}$, define the mesh coordinates of a shape, $\mathbf{X}(\mathbf{D})$. A mesh deformation strategy aims at deforming the initial baseline mesh of a shape, $\mathbf{X}^0$, based on the design parameters to evolve new shapes, $\mathbf{X}(\mathbf{X}^0, \mathbf{D})$. The mesh deformation algorithm that is used for the adjoint method in STAR-CCM+ is the radial basis function (RBF) morpher algorithm. The adjoint derivative of the morpher algorithm evaluates:

$$\frac{dL}{d\mathbf{D}}^T = \frac{d\mathbf{X}}{d\mathbf{D}}^T \frac{dL}{d\mathbf{X}}^T \tag{4094}$$

where $\frac{dL}{d\mathbf{X}}^T$ is the provided final mesh sensitivity.

In a shape optimization procedure, the computed design parameter sensitivity as given by Eqn. (4094) is the basis for the calculation of the new design parameter values. The morpher deforms the mesh coordinates of the shape according to these new values.

Primal

Using radial basis functions, the interpolated position $\mathbf{x}$ about the reference position $\mathbf{x}^0$ is given by

$$\mathbf{x}\left(\mathbf{x}^0\right) = \boldsymbol{\alpha} + \sum_{j=1}^{N} \boldsymbol{\beta}_j \phi_j\left(r_j\left(\mathbf{x}^0\right)\right) \tag{4095}$$

where

$$r_j\left(\mathbf{x}^0\right) = \left\|\mathbf{x}^0 - \mathbf{x}_j^0\right\| \tag{4096}$$

There are two types of radial basis function: those with compact support and those with global support. Here, only one global support function is considered, the multiquadratic biharmonics function $\phi(r)$ given as:

$$\phi_j\left(r_j\right) = \sqrt{r_j^2 + c_j^2} \tag{4097}$$

and the constant $c_j = 0,\ \forall j$. For a discrete set of N reference positions in the domain, evaluation of all M interpolation positions in that domain can be evaluated by casting Eqn. (4095) as:

$$\begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_M \end{bmatrix} = \begin{bmatrix} 1_1 \\ \vdots \\ 1_M \end{bmatrix} \boldsymbol{\alpha} + \begin{bmatrix} \phi_{1,1} & \cdots & \phi_{1,N} \\ \vdots & \ddots & \vdots \\ \phi_{M,1} & \cdots & \phi_{M,N} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}_1 \\ \vdots \\ \boldsymbol{\beta}_N \end{bmatrix} \tag{4098}$$

where the components of the $N \times M$ matrix are computed using the radial basis function

$$\phi_{i,j} = \phi_j\left(r_j\left(\mathbf{x}_i^0\right)\right) \tag{4099}$$

The continuous interpolation function Eqn. (4095) or its discrete form of Eqn. (4098) require the coefficients α and $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_N$. These coefficients are determined by N known interpolation positions, or design parameters, $D = d_1, \ldots, d_N$, and the additional constraint that

$$\sum_{j=1}^{N} \boldsymbol{\beta}_j = 0 \tag{4100}$$

From Eqn. (4095) and Eqn. (4100), a system of equations is constructed to solve the unknown coefficients:

$$\begin{bmatrix} \phi_{1,1} & \cdots & \phi_{1,N} & 1_1 \\ \vdots & \ddots & \vdots & \vdots \\ \phi_{N,1} & \cdots & \phi_{N,N} & 1_N \\ 1_1 & \cdots & 1_N & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}_1 \\ \vdots \\ \boldsymbol{\beta}_N \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_N \\ \mathbf{0} \end{bmatrix} \tag{4101}$$

Adjoint

From a functional point of view, the independent variables of the morpher are $\mathbf{D} = \mathbf{d}_1, \ldots, \mathbf{d}_N$, and its dependent variable is $\mathbf{x}$ in the continuous form and $\mathbf{X} = \mathbf{x}_1, \ldots, \mathbf{x}_M$ in the discrete form. The positions $\mathbf{X}^0$ are constants of the system. Since only the discrete form of the morpher is of interest, the required Jacobian is then:

$$J_{\mathbf{X}} = \frac{d\mathbf{X}}{d\mathbf{D}} \tag{4102}$$

A tangent, j, of the Jacobian Eqn. (4102) is derived by applying the chain rule to Eqn. (4098):

$$\frac{\partial \mathbf{X}}{\partial \mathbf{d}_j} = \begin{bmatrix} \frac{\partial \mathbf{x}_1}{\partial \mathbf{d}_j} \\ \vdots \\ \frac{\partial \mathbf{x}_M}{\partial \mathbf{d}_j} \end{bmatrix} = \begin{bmatrix} 1_1 \\ \vdots \\ 1_M \end{bmatrix} \frac{\partial \boldsymbol{\alpha}}{\partial \mathbf{d}_j} + \begin{bmatrix} \phi_{1,1} & \cdots & \phi_{1,N} \\ \vdots & \ddots & \vdots \\ \phi_{M,1} & \cdots & \phi_{M,N} \end{bmatrix} \begin{bmatrix} \frac{\partial \boldsymbol{\beta}_1}{\partial \mathbf{d}_j} \\ \vdots \\ \frac{\partial \boldsymbol{\beta}_N}{\partial \mathbf{d}_j} \end{bmatrix} \tag{4103}$$

The derivative of the coefficients α and $\beta_1, \ldots, \beta_N$ with respect to d_j are determined by solving the following system:

$$\begin{bmatrix} \phi_{1,1} & \cdots & \phi_{1,N} & 1_1 \\ \vdots & \ddots & \vdots & \vdots \\ \phi_{N,1} & \cdots & \phi_{N,N} & 1_N \\ 1_1 & \cdots & 1_N & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial \boldsymbol{\beta}_1}{\partial \mathbf{d}_j} \\ \vdots \\ \frac{\partial \boldsymbol{\beta}_N}{\partial \mathbf{d}_j} \\ \frac{\partial \alpha}{\partial d_j} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{d}_1}{\partial \mathbf{d}_j} \\ \vdots \\ \frac{\partial \mathbf{d}_N}{\partial \mathbf{d}_j} \\ 0 \end{bmatrix} \tag{4104}$$

Therefore, the requirement for solving the tangent derivative is the capacity to evaluate Eqn. (4098) and Eqn. (4101) for specific inputs.

The aim of the adjoint evaluation of the morpher is to compute Eqn. (4094). The second term on the right hand side of this equation is already provided (see Sensitivity of Cost Function w.R.t. Flow and Energy Solution), so the adjoint of the morpher must compute the product of the two terms, that is, $\frac{dL}{d\mathbf{D}}^T$.

This product is derived by applying the chain rule to Eqn. (4098) and transposing the result:

$$\frac{\partial L}{\partial \boldsymbol{\alpha}} = \begin{bmatrix} 1_1 \\ \vdots \\ 1_M \end{bmatrix}^T \begin{bmatrix} \frac{\partial L}{\partial \mathbf{x}_1} \\ \vdots \\ \frac{\partial L}{\partial \mathbf{x}_M} \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial L}{\partial \boldsymbol{\beta}_1} \\ \vdots \\ \frac{\partial L}{\partial \boldsymbol{\beta}_N} \end{bmatrix} = \begin{bmatrix} \phi_{1,1} & \cdots & \phi_{1,N} \\ \vdots & \ddots & \vdots \\ \phi_{M,1} & \cdots & \phi_{M,N} \end{bmatrix}^T \begin{bmatrix} \frac{\partial L}{\partial \mathbf{x}_1} \\ \vdots \\ \frac{\partial L}{\partial \mathbf{x}_M} \end{bmatrix} \tag{4105}$$

On evaluating Eqn. (4105), the adjoint is obtained by solving the following system:

$$\begin{bmatrix} \phi_{1,1} & \cdots & \phi_{1,N} & 1_1 \\ \vdots & \ddots & \vdots & \vdots \\ \phi_{N,1} & \cdots & \phi_{N,N} & 1_N \\ 1_1 & \cdots & 1_N & 0 \end{bmatrix}^T \begin{bmatrix} \frac{\partial L}{\partial \mathbf{d}_1} \\ \vdots \\ \frac{\partial L}{\partial \mathbf{d}_N} \\ \mathbf{t} \end{bmatrix} = \begin{bmatrix} \frac{\partial L}{\partial \boldsymbol{\beta}_1} \\ \vdots \\ \frac{\partial L}{\partial \boldsymbol{\beta}_N} \\ \frac{\partial L}{\partial \boldsymbol{\alpha}} \end{bmatrix} \tag{4106}$$

where the result t is discarded.

Sensitivity with Respect to Boundary Parameters

The optimization of an objective based on a modification of the boundary conditions requires the sensitivity of the objective with respect to the boundary parameters, that is, $\frac{dL}{d\mathbf{Q}_b}$.

Consider a cost function L of the form:

$$L = L\left(\mathbf{Q}\left(\mathbf{Q}_b\right), \mathbf{Q}_b\right) \tag{4107}$$

where $\mathbf{Q}$ is the solution of the following set of nonlinear, discrete equations:

$$\mathbf{R}\left(\mathbf{Q}\left(\mathbf{Q}_b\right), \mathbf{Q}_b\right) = 0 \tag{4108}$$

$\mathbf{R}$ are the residuals of the flow equations and $\mathbf{Q}_b$ are the imposed boundary conditions.

Since the solution $\mathbf{Q}$ to this set of equations depends on the imposed boundary conditions $\mathbf{Q}_b$, there is an implicit dependence of L on $\mathbf{Q}_b$ in addition to the explicit dependence. The objective is to quantify this dependency.

Quantifying the dependency is achieved in a similar manner to the mesh dependency, by differentiating the residual equation and substituting the result into the objective differentiation:

$$\frac{dL}{d\,\mathbf{Q}_b} = \frac{\partial L}{\partial \mathbf{Q}_b} - \left(\frac{\partial L}{\partial \mathbf{Q}}\right)\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}^{-1}\frac{\partial \mathbf{R}}{\partial \mathbf{Q}_b} \tag{4109}$$

This equation can be simplified by using:

$$\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}^{-T}\left(\frac{\partial L}{\partial \mathbf{Q}}\right) = \mathbf{\Lambda}_\mathbf{Q} \tag{4110}$$

where the solution of the flow adjoint $\mathbf{\Lambda}_\mathbf{Q}$ is obtained from Eqn. (4093).

This simplification leads to:

$$\frac{dL}{d\mathbf{Q}_b} = \frac{\partial L}{\partial \mathbf{Q}_b} - \mathbf{\Lambda}_\mathbf{Q}{}^T\frac{\partial \mathbf{R}}{\partial \mathbf{Q}_b} \tag{4111}$$

From Eqn. (4111) it can be seen that the computation of the cost function sensitivity with respect to boundary conditions involves two steps:

- Differentiation of the explicit dependency of the cost function on the imposed boundary conditions.
- Accounting for the implicit dependence that is given by the second term in Eqn. (4111), which requires obtaining the global adjoint solution.

Surface Sensitivity

The surface sensitivity allows you to quantify the impact of changes to a part surface on an objective of interest. Faces that show large values of surface sensitivity have a large impact on the objective.

Conceptually, the surface sensitivity is similar to the Adjoint of Mesh Deformation. However, its formulation allows a much quicker calculation of sensitivity values at every face of the surface.

The surface sensitivity enables you to analyze large-scale trends in the design. You can use the surface sensitivity to guide a manual shape optimization. The surface sensitivity can also provide guidance for placing control points, which drive the displacement of mesh vertices when using the morpher.

Primal

The formulation for surface sensitivity is based on a spring-analogy mesh deformation strategy (as opposed to the strategy employed by the adjoint of mesh deformation that is based on radial basis functions). However, STAR-CCM+ does not use this strategy to deform the mesh. Instead, this method is only used to derive an approximate mesh adjoint that can provide surface sensitivities.

For this strategy, the coupling between neighboring cells is modeled using springs. Force equilibrium is used to solve for the displacement of the cell centers based on specified displacements on the boundary faces.

For cells in the interior of the domain, force equilibrium for cell i requires that the forces that are associated with the adjacent cells sum up to zero:

$$\sum_j k_{ij}\left(\delta x_j - \delta x_i\right) = 0 \tag{4112}$$

where:

- j represents an adjacent cell.
- k_{ij} represent the spring constant of the spring connecting cell i and j. The spring constant k_{ij} is inversely proportional to the distance between the centroids of cells i and j.
- δx_i is the displacement at the centroid of cell i.

For cells on the boundary of the domain, force equilibrium still requires that the forces balance. However, the displacement at the boundary face is a known quantity and can be moved to the right-hand side of the equation:

$$\sum_{j,\,\mathrm{interior}} k_{ij}\left(\delta x_j - \delta x_i\right) - k_{ib}\delta x_i = -k_{ib}\delta x_b \tag{4113}$$

where δx_b represents the displacement that is applied to the boundary face. The summation is performed only over interior faces.

These equations can be solved simultaneously throughout the entire domain, yielding a linear system of the following form:

$$\mathbf{K}\delta\mathbf{x} = B \tag{4114}$$

where B is defined as:

$$B = \begin{cases} 0 & \text{for interior cells} \\ -k_{ib}\delta x_b & \text{for cells on boundaries} \end{cases} \tag{4115}$$

The displacement for each cell centroid is computed by inverting the matrix $\mathbf{K}$:

$$\delta\mathbf{x} = \mathbf{K}^{-1}B \tag{4116}$$

Finally, the displacement for each vertex in the domain is calculated. For interior vertices, the vertex displacement is the average of the cell centroid displacements over all adjacent cells. For vertices on the boundaries, the displacement is an average of the specified displacements over all adjacent boundary faces.

Adjoint

To calculate the surface sensitivity, the above primal formulation is differentiated in reverse order. The input to this adjoint calculation is the cost function sensitivity with respect to the mesh $\frac{dL}{d\mathbf{X}}$. The output is the cost function sensitivity with respect to the surface displacements $\frac{dL}{d\left(\delta\mathbf{x_b}\right)}$. The output is associated with each face on the surface and represents the change of the objective value due to a displacement of the respective face centroid.

The first step is calculating the sensitivity with respect to the cell centroid and boundary face centroid displacements based on $\frac{dL}{d\mathbf{X}}$. The derivative with respect to the cell centroids is determined by performing a transpose average over the vertices of the cells:

$$\frac{\partial L}{\partial\left(\delta x_i\right)} = \sum_k \frac{1}{n_k^{(c)}}\left(\frac{dL}{d\mathbf{X}}\right)_k \tag{4117}$$

where $n_k^{(c)}$ is the number of cells adjacent to vertex k.

The derivative with respect to the boundary face centroids $\frac{\partial L}{\partial\left(\delta\mathbf{x_b}\right)}$ is given by a transpose average of $\frac{dL}{d\mathbf{X}}$ over the vertices of the boundary faces:

$$\frac{\partial L}{\partial\left(\delta x_{b,i}\right)} = \sum_k \frac{1}{n_k^{(b)}}\left(\frac{dL}{d\mathbf{X}}\right)_k \tag{4118}$$

where $n_k^{(b)}$ is the number of boundary faces adjacent to vertex k.

Because this mesh deformation process is linear, the mesh adjoint Λ_X is given by inverting the transpose of the matrix $\mathbf{K}$ onto the $\frac{\partial L}{\partial(\delta x)}$ vector.

$$\Lambda_X = \mathbf{K}^{-T}\frac{\partial L}{\partial(\delta x)} \tag{4119}$$

The total surface sensitivity is the sum of two components. The first component is the sensitivity of the boundary centroid displacement due to the vertex sensitivity given by Eqn. (4118). The second components is the sensitivity of the spring analogy equation with respect to the boundary displacement multiplied by the mesh adjoint. For boundary face j connected to cell i, the total surface sensitivity is given as:

$$\frac{dL}{d\left(\delta x_{b,j}\right)} = \frac{\partial L}{\partial\left(\delta x_{b,j}\right)} - k_{b,ij}\Lambda_{X,i} \tag{4120}$$

Sensitivity Smoothing

The calculated surface sensitivity $\frac{dL}{d\left(\delta x_b\right)}$ directly depends on the mesh. For this reason, the values tend to be irregularly distributed over the surface and often display large spikes that can overshadow other sensitive parts of the surface. To overcome this problem, the surface sensitivity solver provides an optional smoothing step. This smoothing step spatially averages the sensitivity on the surface and eliminates local spikes.

The smoothing step uses an implicit smoothing method and is defined as:

$$(\mathbf{I} + \epsilon\mathbf{G})\frac{\widehat{dL}}{d\left(\delta x_b\right)} = \frac{dL}{d\left(\delta x_b\right)} \tag{4121}$$

where:

- $\mathbf{I}$ is the identity matrix.
- ϵ is the Sensitivity Smoothing Factor and a Model Coefficient.
- $\mathbf{G}$ is the smoothing operator.

The smoothing operator consists of a distance-weighted average between adjacent faces on the surface. For example, the smoothing equation for a face i on the surface is given as:

$$\frac{\widehat{dL}}{d\left(\delta x_b\right)_i} + \sum_{j,\,\mathrm{surface}} \frac{\epsilon}{\Delta S_{i,j}}\left(\frac{\widehat{dL}}{d\left(\delta x_b\right)_j} - \frac{\widehat{dL}}{d\left(\delta x_b\right)_i}\right) = \frac{dL}{d\left(\delta x_b\right)_i} \tag{4122}$$

where $\Delta S_{i,j}$ is the distance between adjacent face centers. The summation is calculated only over adjacent faces on the surface.

The smoothing equations are solved for all faces simultaneously and give the smoothed sensitivity $\frac{\widehat{dL}}{d\left(\delta x_b\right)}$ over the entire surface.

Model Coefficient

Sensitivity Smoothing Factor
0

Adjoint Error Estimation

Adjoint error estimates give estimates of the impact of local discretization errors on the accuracy of the respective cost function.

The flow adjoint solution $\Lambda_{\mathbf{Q}}$ in Eqn. (4093) represents the (local) cost function sensitivity to the discrete second-order residual $\mathbf{R}[(\mathbf{Q}(\mathbf{X}), \mathbf{X})]$.

To give an estimate of the local discretization error, the third-order MUSCL correction $\Psi^{(3)}$ can be defined by using the discrete third-order MUSCL computed residual, $\mathbf{R}^{(3)}[(\mathbf{Q}(\mathbf{X}), \mathbf{X})]$:

$$\Psi^{(3)} \equiv \mathbf{R}^{(3)}[(\mathbf{Q}(\mathbf{X}), \mathbf{X})] - \mathbf{R}[(\mathbf{Q}(\mathbf{X}), \mathbf{X})] \tag{4123}$$

where $\mathbf{Q}$ is the solution of the governing equations of the physics being simulated and $\mathbf{X}$ is the mesh vertex positions.

Note: A large local discretization error is due to the numerical error not being evenly distributed over the domain. $\Psi^{(3)}$ gives a rough estimate of that error. A more consistent error estimate that is obtained through mesh refinement is computationally more expensive to compute.

Because the adjoint solution is proportional to $\frac{d\mathbf{L}}{d\mathbf{R}}$, the change in the objective due to an increase in the discretization order can be estimated by multiplying the adjoint by the change in the residual. This change provides an estimate for the discretization error in the objective at a given location. Defined more formally, the Adjoint Error Estimate is given as the (convolution) product between the (local) discretization error $\Psi^{(3)}$ and the flow adjoint solution $\Lambda_{\mathbf{Q}}$ for the given cost function L:

$$AErr^{L} \equiv \Psi^{(3)} \cdot \Lambda_{\mathbf{Q}} \tag{4124}$$

$AErr^{L}$ is a direct (but simple) estimate of the effect that the local numerical error has on the computed value of the cost function L. The sum over the entire domain:

$$\sum AErr^{L}$$

gives the total effect of the numerical error on the cost function L. The field $AErr^{L}$ provides useful information when performing mesh adaption and for improving the accuracy of the cost function.

Vector and Tensor Algebra

The physical quantities that appear in this formulation are either scalars, vectors, or 2nd-order tensors. For example, the temperature and pressure in a physics continuum can be described using scalar functions, velocity and position are represented by vector functions, whereas stress and strain are described by 2nd-order tensors.

Scalars are independent of the coordinate system. Vectors and tensors are defined with respect to a coordinate basis.

Index and Matrix Notations

Considering an n-dimensional Euclidean space, a vector $\mathbf{a}$ can be written as:

$$\mathbf{a} = a_i \mathbf{e}_i \tag{4125}$$

where $\mathbf{e}_i \;\; i = 1, \ldots, n$ are the basis vectors in a Cartesian coordinate system, and a_i are the components of $\mathbf{a}$ with respect to the vector basis. This formulation uses the Einstein summation convention, which implies summation over a set of repeated indices:

$$a_i \mathbf{e}_i \equiv \sum_{i=1}^{n} a_i \mathbf{e}_i = a_1 \mathbf{e}_1 + a_2 \mathbf{e}_2 + \ldots + a_n \mathbf{e}_n \tag{4126}$$

Similarly, a 2nd-order tensor $\mathbf{A}$ can be written as:

$$\mathbf{A} = A_{ij} \mathbf{e}_i \otimes \mathbf{e}_j \tag{4127}$$

where $\mathbf{e}_i \otimes \mathbf{e}_j$ is the dyadic product, or tensor product, between the basis vectors. See Eqn. (4137).

Eqn. (4125) and Eqn. (4127) are in index notation, which is also called tensor notation. The number of indices that are required to identify the tensor components define the order of the tensor. In fact, scalars and vectors can be considered as 0th-order and 1st-order tensors, respectively.

Tensors can be defined as covariant or contravariant based on how their components transform under a change of coordinate system. This formal description is beyond the purpose of this overview.

A convenient notation consists of representing the components of a tensor, with respect to a vector basis, as matrices. In matrix representation, the vector basis is implied.

In the 3D Euclidean space, the components of a vector $\mathbf{a}$ can be represented as a one-dimensional array of length 3:

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \tag{4128}$$

which is called the column vector representation of $\mathbf{a}$. The vector $\mathbf{a}$ can also be defined using the row vector representation:

$$\mathbf{a} = \begin{pmatrix} a_1 & a_2 & a_3 \end{pmatrix} \tag{4129}$$

The choice of row or column representation is purely conventional. Defining row and column representations allows you to write the scalar product and the dyadic product between two vectors as matrix products.

In the 3D Euclidean space, the components of a 2nd-order tensor $\mathbf{A}$ can be represented as a 3x3 matrix:

$$\mathbf{A} = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} \tag{4130}$$

The most common operations between scalars, vectors, and 2nd-order tensors (from now on, simply referred to as tensors) are presented below. All components are defined with respect to a Cartesian coordinate basis $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ in the 3D Euclidean space, in either index or matrix notation.

Vectors

Multiplication by a Scalar

The product of a vector $\mathbf{a}$ with a scalar α defines a vector:

$$\alpha\mathbf{a} = \alpha \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} \alpha a_1 \\ \alpha a_2 \\ \alpha a_3 \end{pmatrix} \tag{4131}$$

Sum of two Vectors

The sum of two vectors $\mathbf{a}$ and $\mathbf{b}$ defines a vector:

$$\mathbf{a} + \mathbf{b} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} a_1 + b_1 \\ a_2 + b_2 \\ a_3 + b_3 \end{pmatrix} \tag{4132}$$

Scalar Product of two Vectors

The scalar product between two vectors $\mathbf{a}$ and $\mathbf{b}$ is the matrix product between the row vector representation of $\mathbf{a}$ and the column vector representation of $\mathbf{b}$, which defines a scalar:

$$\mathbf{a} \cdot \mathbf{b} = \begin{pmatrix} a_1 & a_2 & a_3 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = a_1 b_1 + a_2 b_2 + a_3 b_3 \tag{4133}$$

As vectors are usually defined in column representation, the scalar product is often written as $\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^{\mathrm{T}}\mathbf{b} = \mathbf{b}^{\mathrm{T}}\mathbf{a}$, where T indicates the matrix transpose. The transpose of a column vector is a row vector, and the transpose of a row vector is a column vector.

The magnitude of a vector $\mathbf{a}$ can be written as:

$$|\mathbf{a}| = \sqrt{\mathbf{a} \cdot \mathbf{a}} \tag{4134}$$

Vector Product of two Vectors

The vector product between two vectors $\mathbf{a}$ and $\mathbf{b}$ defines a vector:

$$\mathbf{a} \times \mathbf{b} = \begin{pmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{pmatrix} \tag{4135}$$

which is obtained from the formal determinant:

$$\det \begin{pmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{pmatrix} \tag{4136}$$

Dyadic Product of two Vectors

The dyadic product, or tensor product, between two vectors $\mathbf{a}$ and $\mathbf{b}$ defines a tensor:

$$\mathbf{ab} \equiv \mathbf{a} \otimes \mathbf{b} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \begin{pmatrix} b_1 & b_2 & b_3 \end{pmatrix} = \begin{pmatrix} a_1 b_1 & a_1 b_2 & a_1 b_3 \\ a_2 b_1 & a_2 b_2 & a_2 b_3 \\ a_3 b_1 & a_3 b_2 & a_3 b_3 \end{pmatrix} \tag{4137}$$

The notation $\mathbf{ab}$ assumes that $\mathbf{a}$ is represented by a column vector and $\mathbf{b}$ is represented by a row vector.

The tensor product is often written as $\mathbf{ab}^{\mathrm{T}}$, which assumes that both $\mathbf{a}$ and $\mathbf{b}$ are represented by column vectors.

Gradient of a Scalar

The gradient of a scalar φ defines a vector:

$$\nabla \varphi = \begin{pmatrix} \dfrac{\partial \varphi}{\partial x} \\ \dfrac{\partial \varphi}{\partial y} \\ \dfrac{\partial \varphi}{\partial z} \end{pmatrix} \tag{4138}$$

Gradient of a Vector

The gradient of a vector $\mathbf{a}$ defines a tensor:

$$\nabla \mathbf{a} = \begin{pmatrix} \frac{\partial a_1}{\partial x} & \frac{\partial a_1}{\partial y} & \frac{\partial a_1}{\partial z} \\ \frac{\partial a_2}{\partial x} & \frac{\partial a_2}{\partial y} & \frac{\partial a_2}{\partial z} \\ \frac{\partial a_3}{\partial x} & \frac{\partial a_3}{\partial y} & \frac{\partial a_3}{\partial z} \end{pmatrix} \tag{4139}$$

Divergence of a Vector

The divergence of a vector $\mathbf{a}$ defines a scalar:

$$\nabla \cdot \mathbf{a} = \frac{\partial a_1}{\partial x} + \frac{\partial a_2}{\partial y} + \frac{\partial a_3}{\partial z} \tag{4140}$$

Curl of a Vector

The curl of a vector $\mathbf{a}$ defines a vector:

$$\nabla \times \mathbf{a} = \begin{pmatrix} \frac{\partial a_3}{\partial y} - \frac{\partial a_2}{\partial z} \\ \frac{\partial a_1}{\partial z} - \frac{\partial a_3}{\partial x} \\ \frac{\partial a_2}{\partial x} - \frac{\partial a_1}{\partial y} \end{pmatrix} \tag{4141}$$

which is obtained from the formal determinant:

$$\det \begin{pmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ a_1 & a_2 & a_3 \end{pmatrix} \tag{4142}$$

Tensors

Multiplication by a Scalar

The product of a tensor $\mathbf{A}$ with a scalar α defines a tensor:

$$\alpha \mathbf{A} = \alpha \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} = \begin{pmatrix} \alpha A_{11} & \alpha A_{12} & \alpha A_{13} \\ \alpha A_{21} & \alpha A_{22} & \alpha A_{23} \\ \alpha A_{31} & \alpha A_{32} & \alpha A_{33} \end{pmatrix} \tag{4143}$$

Sum of two Tensors

The sum of two tensors $\mathbf{A}$ and $\mathbf{B}$ defines a tensor:

$$\mathbf{A}+\mathbf{B} = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} + \begin{pmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{pmatrix} = \\ = \begin{pmatrix} A_{11}+B_{11} & A_{12}+B_{12} & A_{13}+B_{13} \\ A_{21}+B_{21} & A_{22}+B_{22} & A_{23}+B_{23} \\ A_{31}+B_{31} & A_{32}+B_{32} & A_{33}+B_{33} \end{pmatrix} \tag{4144}$$

Product of a Tensor with a Vector

The product of a tensor $\mathbf{A}$ with a vector $\mathbf{b}$ defines a vector:

$$\mathbf{Ab} \equiv \mathbf{A}\cdot\mathbf{b} = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} A_{11}b_1 + A_{12}b_2 + A_{13}b_3 \\ A_{21}b_1 + A_{22}b_2 + A_{23}b_3 \\ A_{31}b_1 + A_{32}b_2 + A_{33}b_3 \end{pmatrix} \tag{4145}$$

Eqn. (4145) shows that tensors define linear transformations between vectors.

Product of two Tensors

The product of two tensors $\mathbf{A}$ and $\mathbf{B}$ defines a tensor:

$$\mathbf{AB} \equiv \mathbf{A}\cdot\mathbf{B} = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{pmatrix} = \\ = \begin{pmatrix} A_{11}B_{11} + A_{12}B_{21} + A_{13}B_{31} & A_{11}B_{12} + A_{12}B_{22} + A_{13}B_{32} & A_{11}B_{13} + A_{12}B_{23} + A_{13}B_{33} \\ A_{21}B_{11} + A_{22}B_{21} + A_{23}B_{31} & A_{21}B_{12} + A_{22}B_{22} + A_{13}B_{32} & A_{21}B_{13} + A_{22}B_{23} + A_{23}B_{33} \\ A_{31}B_{11} + A_{32}B_{21} + A_{33}B_{31} & A_{31}B_{12} + A_{32}B_{22} + A_{33}B_{32} & A_{31}B_{13} + A_{32}B_{23} + A_{33}B_{33} \end{pmatrix} \tag{4146}$$

Eqn. (4146) shows that tensors also define linear transformations between tensors.

Inner Product of two Tensors

The inner product, or double dot product, of two tensors $\mathbf{A}$ and $\mathbf{B}$ defines a scalar:

$$\mathbf{A}:\mathbf{B} = A_{11}B_{11} + A_{12}B_{12} + A_{13}B_{13} + \ldots \tag{4147}$$

or, in index notation:

$$\mathbf{A}:\mathbf{B} = A_{ij}B_{ij} \tag{4148}$$

The inner product is a tensor invariant, as its value remain unchanged under a change of basis.

Divergence of a Tensor

The divergence of a tensor $\mathbf{A}$ defines the vector:

$$\nabla\cdot \mathbf{A} = \begin{pmatrix} \frac{\partial A_{11}}{\partial x} + \frac{\partial A_{21}}{\partial y} + \frac{\partial A_{31}}{\partial z} \\ \frac{\partial A_{12}}{\partial x} + \frac{\partial A_{22}}{\partial y} + \frac{\partial A_{32}}{\partial z} \\ \frac{\partial A_{13}}{\partial x} + \frac{\partial A_{23}}{\partial y} + \frac{\partial A_{33}}{\partial z} \end{pmatrix}^{\mathrm{T}} \tag{4149}$$

Eigenvalues and Eigenvectors of a Tensor

Given a tensor $\mathbf{A}$, the scalars λ and vectors $\mathbf{v}$ that satisfy the relation:

$$\mathbf{A}\cdot\mathbf{v} = \lambda\mathbf{v} \tag{4150}$$

are called the eigenvalues and eigenvectors of $\mathbf{A}$, respectively.

Inverse of a Tensor

Eqn. (4145) and Eqn. (4146) define tensors as linear transformations between vectors, or tensors. Given a tensor $\mathbf{A}$, the inverse transformation $\mathbf{A}^{-1}$ is defined as:

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{I} \tag{4151}$$

where $\mathbf{I}$ is the identity tensor:

$$\mathbf{I} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4152}$$

The inverse tensor, $\mathbf{A}^{-1}$, exists if and only if $\det(\mathbf{A}) \neq 0$, where:

$$\begin{aligned} \det(\mathbf{A}) &= \det\begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} = \\ &= A_{11}\left(A_{22}A_{33} - A_{23}A_{32}\right) + A_{12}\left(A_{23}A_{31} - A_{21}A_{33}\right) + A_{13}\left(A_{21}A_{32} - A_{31}A_{22}\right) \end{aligned} \tag{4153}$$

Symmetry

A tensor $\mathbf{A}$ is called a symmetric tensor when $\mathbf{A} = \mathbf{A}^{\mathrm{T}}$, that is:

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{21} & A_{31} \\ A_{12} & A_{22} & A_{32} \\ A_{13} & A_{23} & A_{33} \end{pmatrix} \tag{4154}$$

Tensor Invariants

Given a tensor $\mathbf{A}$, it is possible to define three independent invariants, that is, scalar quantities that remain unchanged under a change of coordinate system:

$$\mathrm{I}_0 = \mathrm{tr}(\mathbf{A}) = A_{11} + A_{22} + A_{33} = A_{ii} \tag{4155}$$

$$\mathrm{I}_1 = \left[\mathrm{tr}(\mathbf{A})^2 - \mathrm{tr}\left(\mathbf{A}^2\right)\right]/2 \tag{4156}$$

$$\mathrm{I}_2 = \det\left(\mathbf{A}\right) \tag{4157}$$

The invariant I_0 is called the **Trace** of the tensor $\mathbf{A}$.

Tutorials

Tutorials show you how to use STAR-CCM+ for various applications in a step-by-step format with recommendations for setup, initialization and steps of the solution process specific to the application. In addition to these written instructions, macro and simulation files are available for download for a large proportion of cases.

The tutorials are categorized as follows:

- Introduction to STAR-CCM+ presents the important concepts and the general workflow. If you are new to STAR-CCM+, complete this tutorial first.
- Foundational tutorials demonstrate the major features of STAR-CCM+, spanning the entire workflow.
- Geometry tutorials demonstrate how to create and modify parts using 3D-CAD.
- Mesh tutorials demonstrate how to use the surface wrapper and several of the meshing features.
- Incompressible flow tutorials demonstrate simple liquid flows. Features such as steady and unsteady flows are introduced as well as multi-component flows, porosity and solution recording.
- Compressible fluid flow tutorials demonstrate the solution of subsonic and transonic flows and turbomachinery problems.
- Heat transfer and radiation tutorials demonstrate conjugate heat transfer, radiation and thermal comfort features.
- Multiphase tutorials demonstrate VOF, Lagrangian and Eulerian multiphase and fluid film features.
- Discrete Element Method tutorials demonstrate how to model large numbers of interacting discrete particles.
- Motion tutorials demonstrate the moving reference frames, rigid body motion, dynamic fluid body interaction and mesh morphing.
- Combustion tutorials demonstrate the various combustion models and the fire and smoke wizard.
- Solid stress tutorials demonstrate how to solve for stress and strain and how to couple these interactions in a fluid structure interaction.
- Aeroacoustics tutorials demonstrate the stages in solving a complex transient acoustics problem in the near and far field.
- Electromagnetism tutorials demonstrate analyses that involve an electric field.
- Electrochemistry tutorials demonstrate chemical reactions induced by an electrical current.
- Battery tutorials demonstrate the process of setting up a battery simulation.
- Casting tutorials demonstrate the process of setting up a casting simulation.
- Automation tutorials demonstrate how to use Java macros for repetitive tasks.
- Coupling with CAE codes tutorials demonstrate how to couple STAR-CCM+ simulations.

Additionally, the Steve Portal Knowledge Base contains several video presentations that describe various features and procedures in STAR-CCM+.

Contents:

Using Tutorial Macros and Files
Introduction
Foundation Tutorials
Geometry
Mesh
Incompressible Flow
Compressible Flow
Heat Transfer and Radiation
Multiphase Flow
Discrete Element Method
Motion
Reacting Flow
Solid Stress
Aeroacoustics
Electromagnetism
Electrochemistry
Battery
Casting
Automation
Coupling with CAE Codes
Analysis Methods
Design Manager
STAR-ICE

Using Tutorial Macros and Files

Macros, input files and final simulation files for a range of tutorials are provided as an optional download package on the customer service center web-site. These macros and final simulation files are provided as an aid to the written tutorials, so that you can check your final results against the downloaded files, or against a simulation that is built and run using the macros.

Where a tutorial has several solution stages, a macro has been recorded for each solution stage. A simulation file for each completed macro is provided. You can load the solution for a preliminary stage and play the macro of the subsequent sections. The macros and sim files use naming conventions to facilitate their use.

Contents:

Downloading the Tutorial Files from the Customer Success Portal
Understanding the Directory Structure and Naming Conventions
Playing the Macro
Using the Final Simulation Files

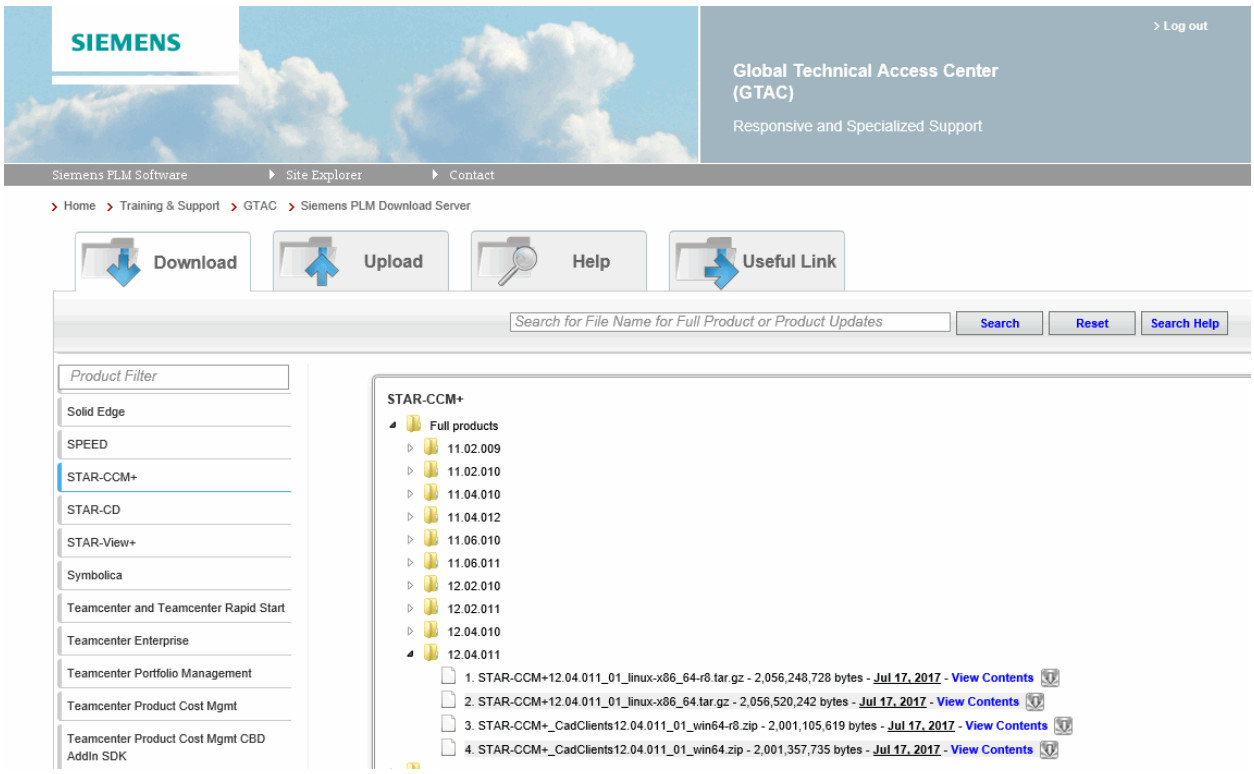
Downloading the Tutorial Files from the Customer Success Portal

In the customer success portal, Siemens PLM Software provides a bundle containing completed simulation files and macros for most of the STAR-CCM+ tutorials.

To download the compressed package of tutorial simulation files and macros:

1. Use your web browser to navigate to the customer success portal, https://thesteveportal.plm.automation.siemens.com, and log in. If you do not have your account information, contact your Siemens PLM Software representative.
2. In the customer success portal, click **Products** > **Downloads**. Your browser is redirected to the Global Technical Access Center (GTAC) of Siemens PLM Software.

   By default, STAR-CCM+ is the selected product.



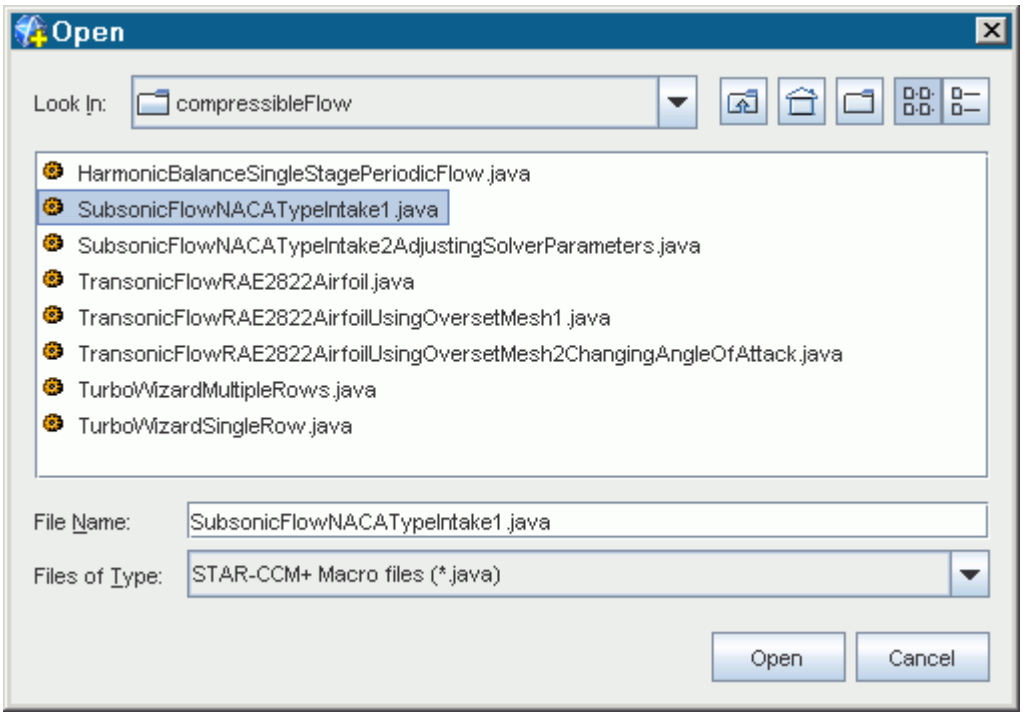
3. Expand the version folder of STAR-CCM+ for which you require tutorial files.
4. The tutorial files are presented in the **Related Files and Documentation** subfolder. Click the download link for the `STAR-CCM+TutorialFiles<Version>.7z` entry. Save the file to an appropriate directory on your disk.
5. Using a decompression tool such as 7-Zip, unzip the file. On Linux, to preserve the directory structure, use the following command line:

   ```
   7z x STAR-CCM+TutorialFiles<Version>.7z
   ```

Understanding the Directory Structure and Naming Conventions

The tutorial files are grouped in directories that are named according to the tutorial sets listed in the documentation.

Directory names do not have spaces and use a camelCase convention. For example, the files for the Subsonic Flow: NACA-Type Intake tutorial are located in the `compressibleFlow` folder.



The macro files within a directory are constructed so that each macro produces only a single solution. Macro names are derived from the tutorial name. For example, the Simple Java Macro is recreated using the macro `SimpleJavaMacroPostProcessingObjects.java`

Where more than one solver run is required within a tutorial, macros are created for each stage. Macros that refer to stages beyond the first stage include the documentation step from which they begin in their file name. For example, the introductory tutorial requires two solver runs. Its macros are:

- `Introduction1.java`
- `Introduction2AdjustingSolverParamters.java`

To run the second part of a tutorial, it is necessary to have the solution to the first part located in the same working directory. The download from User Services contains simulation files for each individual macro — that is, for intermediate stages as well as the final stage. These simulation files are named according to the macro file used to create them. Thus, `Introduction1.java` has a related `.sim` file called `Introduction1_final.sim`. This file can be loaded before running the macro for the second stage, `Introduction2AdjustingSolverParameters.java`.

Playing the Macro

STAR-CCM+ uses the tutorial macros to construct the same model that the tutorial instructions describe.

Before you play a macro, it is important to read through the appropriate section of the tutorial guide. There are three ways a tutorial macro can be used interactively:

- Playing an initial or single macro
- Playing a macro dependent on a pre-existing simulation
- Modifying the macro to play up to any section

Macros can also be run in batch mode.

Playing an Initial or Single Macro

Macros that do not have precursor solutions can be run by creating a new simulation, clicking the (**Play Macro**) toolbar button and selecting the appropriate macro file. The file executes dynamically.

Playing a Macro Dependent on an Existing Simulation

For cases that require existing solutions to be present, the recommended procedure is to load the appropriate file from the downloaded tutorial files before running the macro. For example, to execute the `SolarCollector.java` macro it is necessary to first load the `ThermalInsulator_final.sim` file.

It is possible also to use the results of an upstream macro as the start of the downstream solution. So, using a single simulation, you can run `Introduction1.java` and then`Introduction2AdjustingSolverParamters.java` to obtain the solution to the final introductory case.

Modifying the Macro to Play up to Any Page

The automation tutorials provide guidelines on how macros can be modified to suit your needs. The tutorial macros have been structured in a way that allows you to run each case up to a particular section.

To play, for example, the macro for the Domestic Fuse tutorial up to the step, “Visualizing the Volume Mesh”, do the following:

1. Open the file, `DomesticFuse.java`, in a text editor of your choice.
2. Comment out the following pages in the playlist section using block comments as shown below.

```
public class DomesticFuse extends StarMacro {

    @Override
      public void execute() {
        play(new ImportingTheSurfaceMeshAndNamingTheSimulation());
        play(new VisualizingTheImportedGeometry());
        play(new ExaminingTheImportedGeometry());
        play(new IncreasingTheTessellationDensityOfTheFuseWire());
        play(new AssigningGeometryPartsToRegions());
        play(new SettingUpTheMesh());
        play(new AddingAVolumetricControl());
        play(new GeneratingTheMesh());
        /*play(new VisualizingTheVolumeMesh());
        play(new SelectingCopperOhmicHeatingPhysicsModels());
        play(new SpecifyingCopperOhmicHeatingMaterialProperties());
        play(new SelectingCeramicPhysicsModels());
        play(new SpecifyingCeramicMaterialProperties());
        play(new SelectingSiliciumPhysicsModels());
        play(new SpecifyingSiliciumMaterialProperties());
        play(new AssigningPhysicsContinua());
        play(new SettingGlobalInitialConditions());
        play(new DefiningBoundaryConditions());
        play(new SettingUpReportsMonitorsAndPlots());
        play(new SettingUpAPlaneSection());
        play(new SettingUpScalarScene());
        play(new SettingSolverVariables());
        play(new SettingStoppingCriteria());
        if (StarTestEnv.getSingleton().getStartestHome() == null) {
              play (new RunningTheSimulation());
            }
        play(new TemperatureAndElectricPotentialPlot());
        play(new ReportPlots());*/
    }
```

3. Save the macro.
4. Start a simulation and run the modified macro.

Using the Final Simulation Files

For each of the macros that are provided, a corresponding simulation file is also provided. These can be used to check the settings in the model tree of a case you have manually created from the written tutorial instructions.

As tutorial meshes are often coarse, and the solvers tend to be run for a minimum number of time steps, it is not recommended that tutorials are used for verification or benchmarking. They have not been created with this objective in mind.

You can use the supplied simulation files in those tutorials that begin with loading an existing simulation file of a previous tutorial simulation. However, the names of these supplied files differ from the names in the tutorial exercises. The following table lists the title of each applicable tutorial, the name of the simulation file that is opened at the start of that tutorial, and the path and name of the corresponding downloadable simulation file.

Tutorial	Simulation Filename Used	Supplied Simulation File and Path
Unsteady Multi-Component Flow: Dilution Pipe	`dilPipeSteady.sim`	`incompressibleFlow/SteadyMultiComponentFlowDilutionPipe_final.sim`
Porous Resistance: Orthotropic Media	`isotropicPorousMedia.sim`	`incompressibleFlow/PorousResistanceIsotropicMedia_final.sim`
Harmonic Balance: Single Stage Periodic Flow	`singleRow.sim`	`compressibleFlow/TurboWizardSingleRow_final.sim`
Multiband Surface-to-Surface Radiation: Solar Collector	`thermalInsulator.sim`	`heatTransferAndRadiation/SurfaceToSurfaceRadiationThermalInsulator_final.sim`
Rigid Body Motion: Rotating Fan	`fan_mrf.sim`	`motion/MovingReferenceFramesRotatingFan_final.sim`
PPDF Equilibrium: Non-Premixed Propane	`ebuCombustion.sim`	`reactingFlow/ThreeStepEddyBreakUpNonPremixedPropane_final.sim`
PPDF Flamelets: Non-Premixed Hydrogen	`ebuCombustion.sim`	`reactingFlow/ThreeStepEddyBreakUpNonPremixedPropane_final.sim`

Tutorial	Simulation Filename Used	Supplied Simulation File and Path
Droplet Combustion: Non-Adiabatic Toluene	`ebuCombustion.sim`	`reactingFlow/ThreeStepEddyBreakUpNonPremixedPropane_final.sim`
DES and FW-H On-The-Fly: Noise from a Cylinder (Unsteady Analysis)	`cylinder.sim`	`aeroacoustics/BroadbandModelsNoiseFromACylinderPreparation2RefiningVolMesh_final.sim`
Adjoint Flow: Wing Shape Optimization	`adjointWing_adjoint.sim`	`incompressibleFlow/AdjointExternalFlowDualElementWing2AdjointFlowSolution_final`

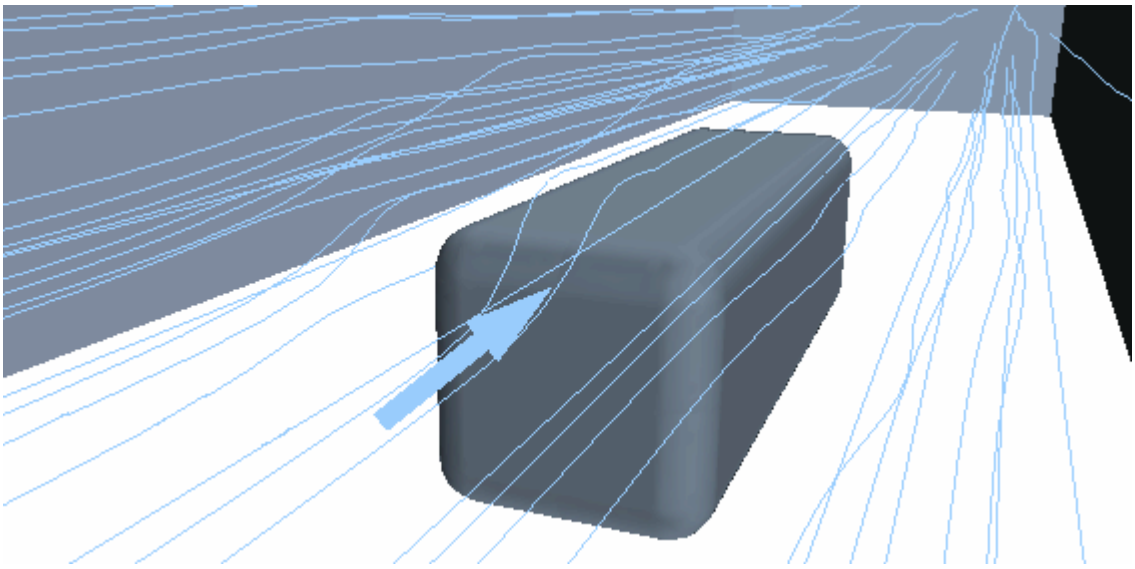
Introduction

Welcome to the STAR-CCM+ introductory tutorial. In this tutorial, you explore the important concepts and workflow. Complete this tutorial before attempting any others.

Throughout this tutorial, links to other sections of the online documentation explore important concepts. For example, for a clearer understanding of the changes to typeface in the tutorials, refer to the typographic conventions.

This tutorial is useful in addition to STAR-CCM+ training. For more help, contact your local Siemens PLM Software office. A list of contacts is found at: http://mdx2.plm.automation.siemens.com/locations.

The case is a transonic flow over an idealized symmetrical blunt body in a wind tunnel.



The tutorial workflow includes:

- Importing the geometry files.
- Generating a polyhedral mesh.
- Setting boundary names and types.
- Defining the models in the continua and applying them to the regions.
- Defining the region conditions, values, and boundary conditions.
- Running the simulation.
- Post-processing the results.

Contents:

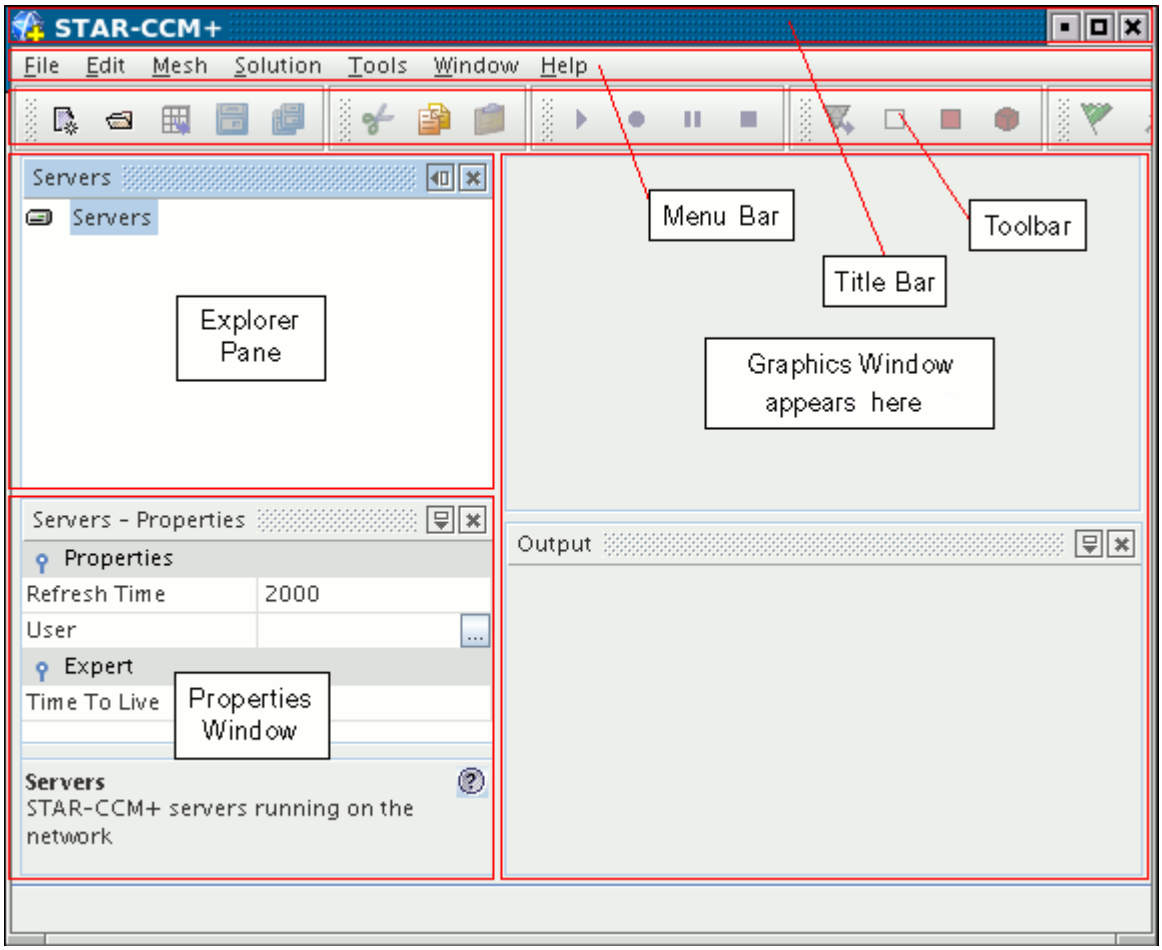
Starting a STAR-CCM+ Simulation
Saving and Naming a Simulation
Importing the Geometry
Visualizing the Imported Geometry
Defining Boundary Surfaces
Assigning Parts to Regions
Setting Boundary Types
Mesh Generation
Selecting the Physics Models
Setting Initial Conditions
Defining the Region Continuum
Setting Boundary Conditions and Values
Setting Solver Parameters and Stopping Criteria
Visualizing the Solution
Monitoring Simulation Progress
Running the Simulation
Adjusting Solver Parameters and Continuing
Visualizing the Results
Plotting Data on a Slice
Adding Streamlines
Closing and Reopening the Simulation
Summary

Starting a STAR-CCM+ Simulation

Start the simulation and familiarize yourself with the STAR-CCM+ user interface.

1. Launch STAR-CCM+ using the appropriate instructions for your operating system.

   After a brief display of the splash screen, the STAR-CCM+ client workspace opens without loading an existing simulation or creating a simulation. This interface to the STAR-CCM+ software is a self-contained graphical user interface (GUI) with panes and subwindows. Some of the GUI terminology is shown in the following screenshot.



   The menu bar provides access to application-wide actions, with some of the more important actions being duplicated in the toolbar.

2. Let the mouse hover over any of the buttons in the toolbar.

   A short description of what that button does appears in a tooltip.

Contents:

Creating a Simulation

Working with Objects

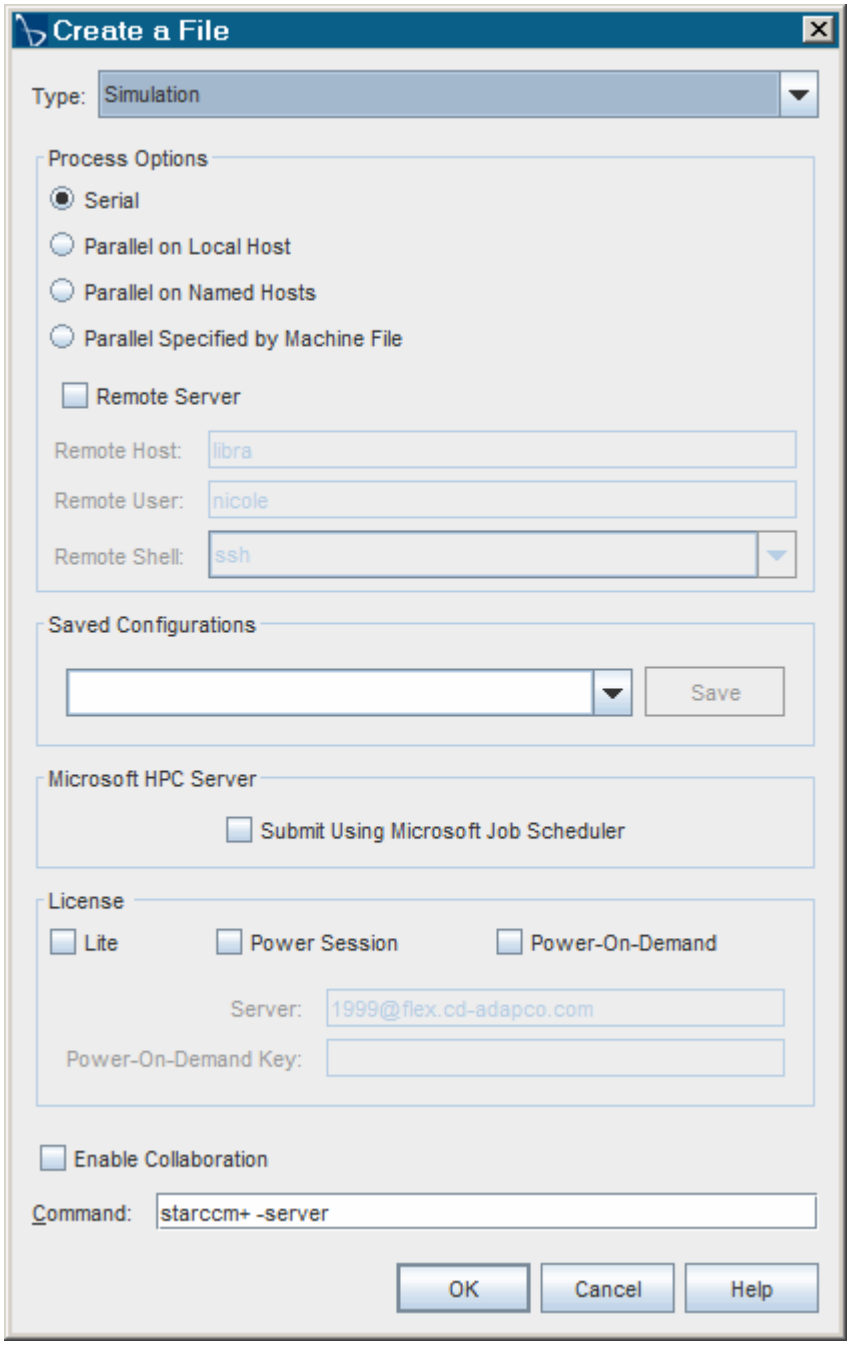
Creating a Simulation

Creating a simulation is the first step in a new STAR-CCM+ analysis.

STAR-CCM+ is a client-server application with the client (user interface or batch interpreter) running in one process, and the server (the solver) running in another process. Start a server process on the same machine as the client:

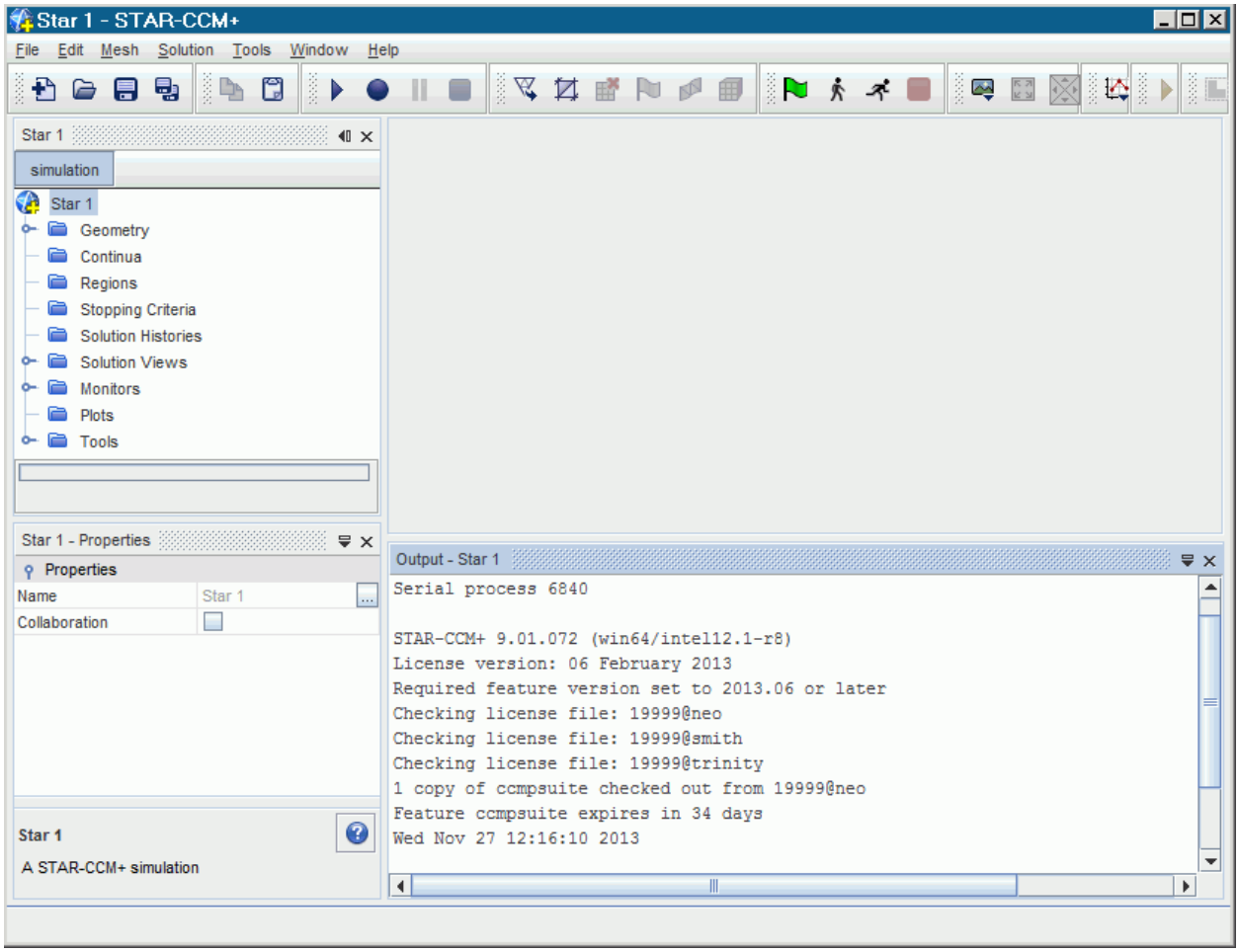
1. Start a simulation by selecting **File** > **New** from the menu bar.

   The *Create a File* dialog appears.



2. In the *Create a File* dialog, click **OK**.

   A new window containing a simulation object tree is created in the Explorer pane, with the name Star 1. The initial folder nodes for this simulation are shown in the following screenshot. Other nodes are added to the object tree as you progress further.

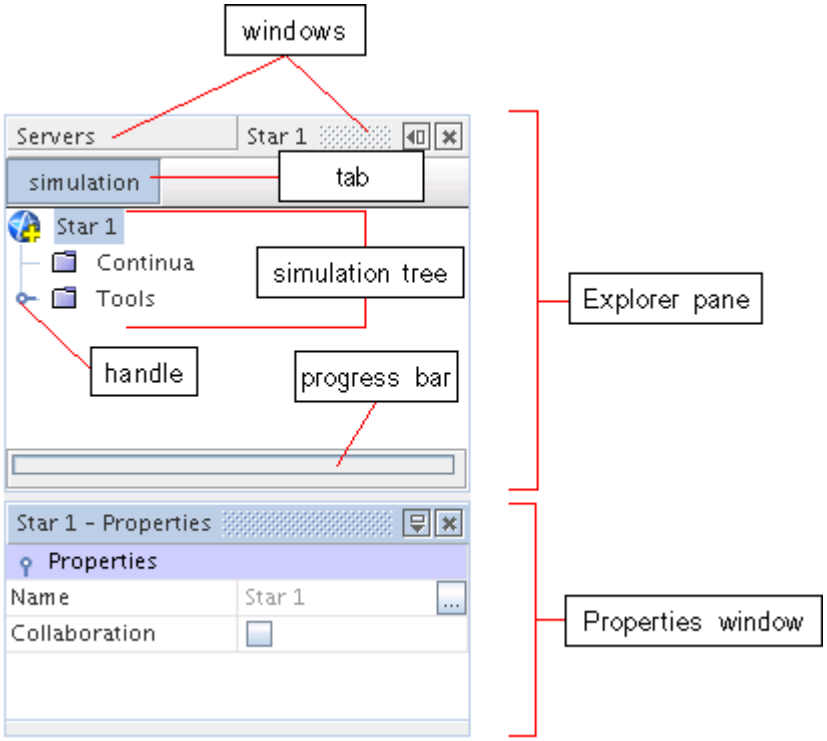


As the new simulation is created, a window that is named Output appears in the lower-right portion of the STAR-CCM+ workspace. The *Output* window describes the progress of actions in the simulation.

Working with Objects

Much of your interaction with the simulation is through the objects in the simulation tree that was added to the *Explorer* pane.

Familiarize yourself with the objects in the simulation tree:

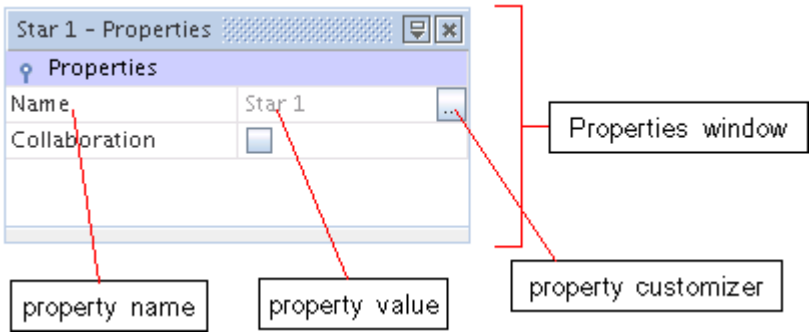


The tree represents all the objects in the simulation. Nodes are added in later sections when a geometry is imported and models are defined in the continuum. Most of your interaction with the simulation is by selecting nodes in the tree and:

- Right-clicking to expose an action menu.
- Using keys to, for example, copy and delete objects.
- Dragging objects to other tree nodes or onto visualization displays.

The handle next to a node indicates that subnodes exist below that one. To open a node and show the subnodes, click the handle. To close it, do the same for an open node.

Most objects in the tree have one or more properties that define the object. Access to the properties is through the table.



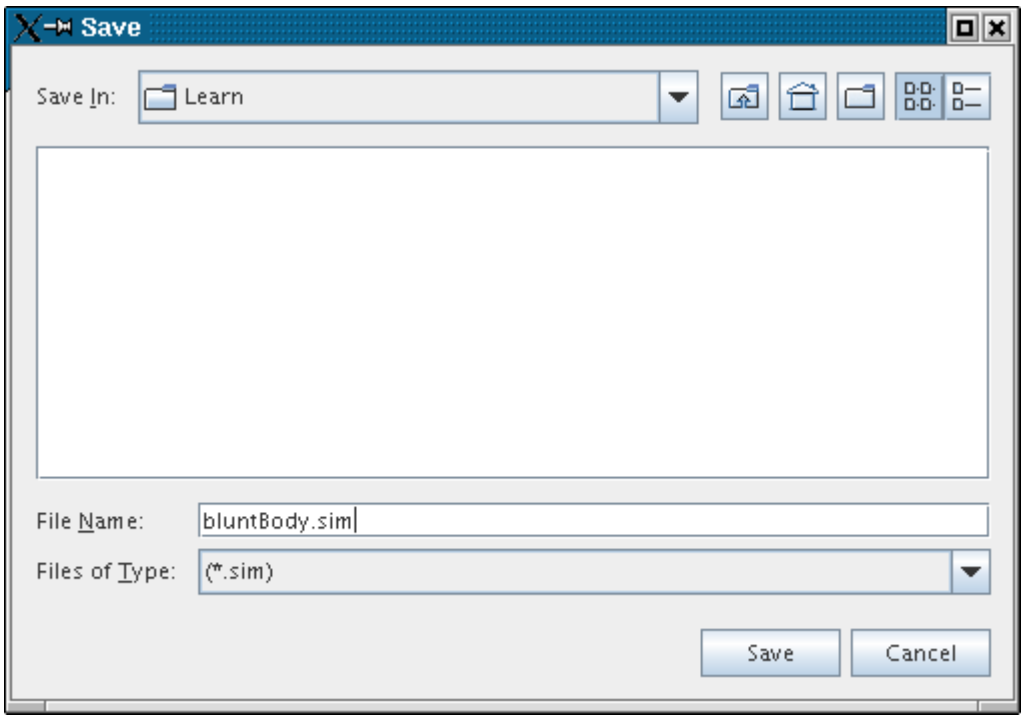
To modify the properties of an object, click its node once to select it. Edit most types of properties in the value cell. Otherwise, when setting values of complex properties, click the **Property Customizer** button to the right of the value. A property-specific dialog opens.

Saving and Naming a Simulation

The simulation file (.sim) contains STAR-CCM+ analysis setup, runtime, and results information.Here, you save the simulation to disk.

To save the simulation:

1. Select **File** > **Save As**.
2. Navigate to the directory where you want to locate the file.



3. In the *Save* dialog, type `bluntBody.sim` into the **File Name** text box and click **Save**.

   The title of the simulation window in the *Explorer* pane updates to reflect the new name.

It is useful to save work in progress periodically. This tutorial includes reminders to save the simulation at the end of each section. If you would like to return to a stage of the tutorial at a later stage, save it with a unique name.

Importing the Geometry

STAR-CCM+ provides tools to generate the geometry, but for this tutorial you start from a pre-generated Parasolid file of the design.

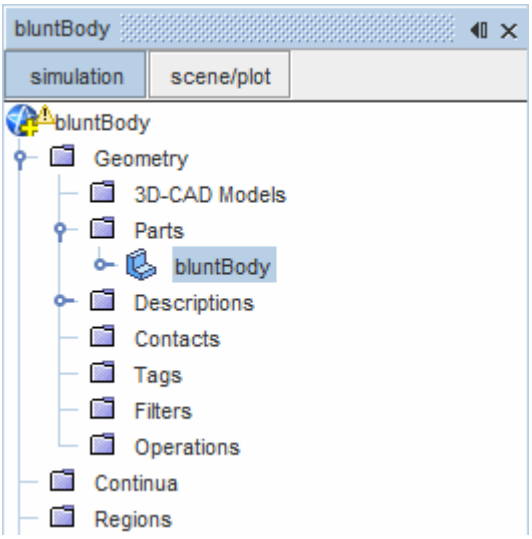
Before solving this case, you can recognize that the flow is likely to be symmetrical in two planes around the body. Hence, only a quarter of the geometry is modeled. This symmetry condition reduces computational costs and solution time without losing information or accuracy.

The first step in this tutorial is to import the geometry:

1. Select **File** > **Import** > **Import Surface Mesh** from the menu bar.
2. In the *Open* dialog, navigate to the `doc/startutorialsdata/introduction/data` subdirectory of your STAR-CCM+ installation directory and select file `bluntBody.x_t`.
3. To start the import, click **Open**.

   The *Import Surface Options* dialog appears.
4. Click **OK** to import the geometry.

   STAR-CCM+ provides feedback on the import process in the *Output* window. A new geometry scene is created in the *Graphics* window and shows the imported geometry.
5. In the simulation tree, expand the **Geometry** > **Parts** node to see the new `bluntBody` part.

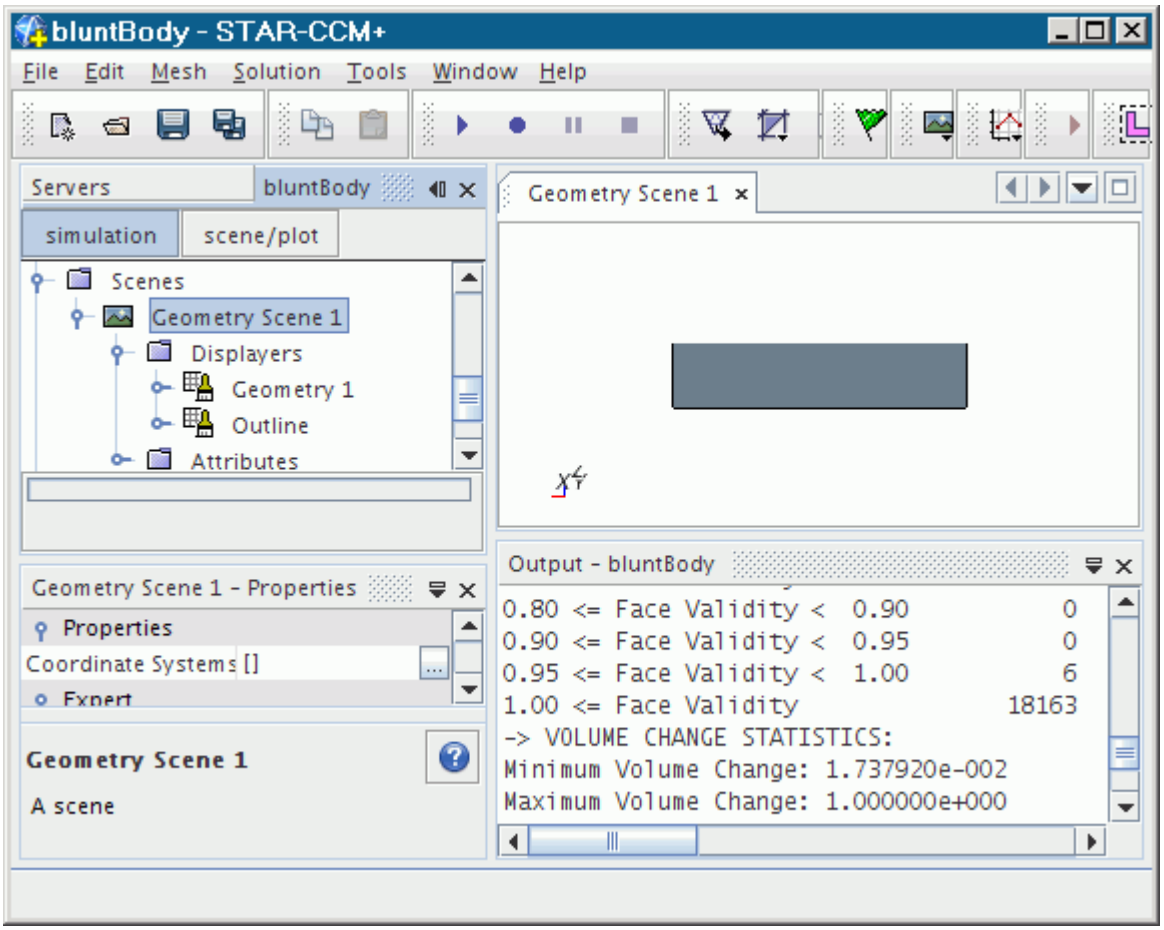


   The imported geometry represents the fluid volume around the body in the wind tunnel.
6. Save the simulation by clicking **Save**.

Visualizing the Imported Geometry

The *Geometry Scene 1* display is already present in the *Graphics* window. Initially, solid, colored surfaces show the surfaces of the geometry.

View the imported geometry in the Graphics window.



A new geometry scene contains:

- A part displayer, **Geometry 1**, which contains all faces in the geometry part and is preset to display a shaded surface.
- A second part displayer, **Outline**, which also contains all faces in the geometry part and is preset to display the mesh outline.

Contents:

Panning, Zooming, and Rotating the View

Panning, Zooming, and Rotating the View

Familiarize yourself with the options to change the view of the geometry.

In the *Graphics* window, all three mouse buttons have drag operations that change the view of the geometry:

- To rotate about the selected point, hold down the left mouse button and drag.
- To zoom in or out, hold down the middle mouse button and drag.
- To translate or pan, hold down the right mouse button and drag.
- To rotate around an axis perpendicular to the screen, press the `<Ctrl>` key and hold down the left mouse button while dragging.

There are also several “hot” keys that rotate the view:

- To align with the X-Y plane, press the `<T>` key.
- To align with the Y-Z plane, press the `<F>` key.
- To align with the Z-X plane, press the `<S>` key.
- To fit the view within the *Graphics* window, press the `<R>` key.

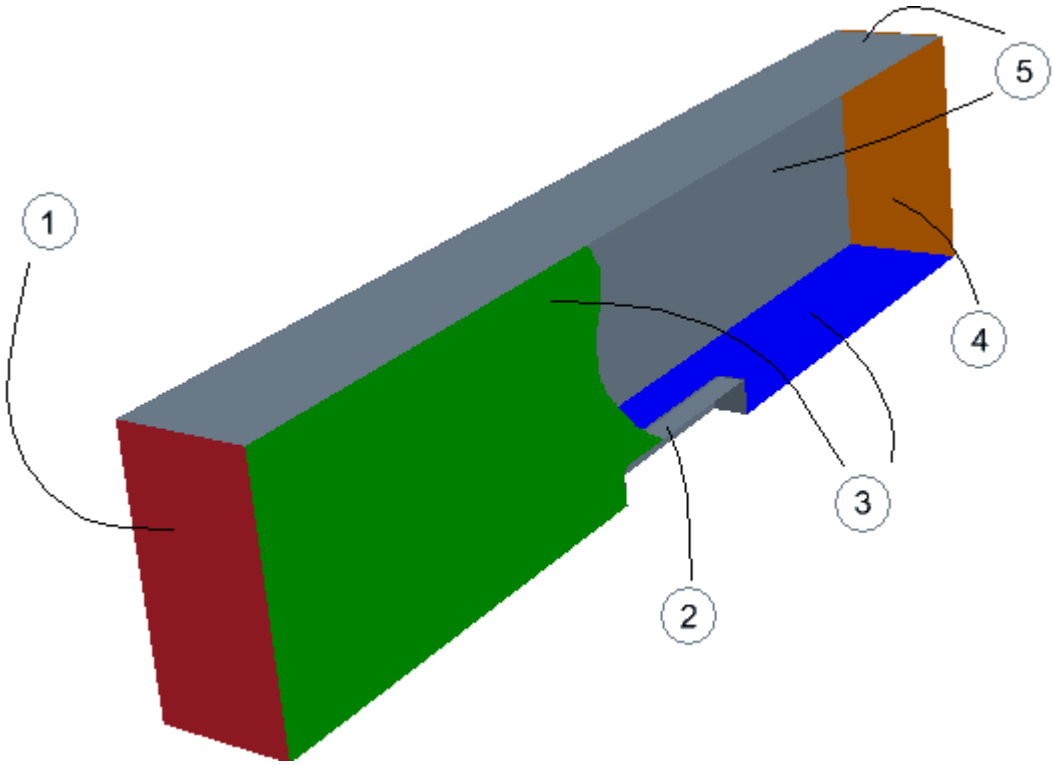
1. Adjust the view as shown in the following screenshot.

2. Save the simulation.

Defining Boundary Surfaces

Specify the boundary type for each surface.

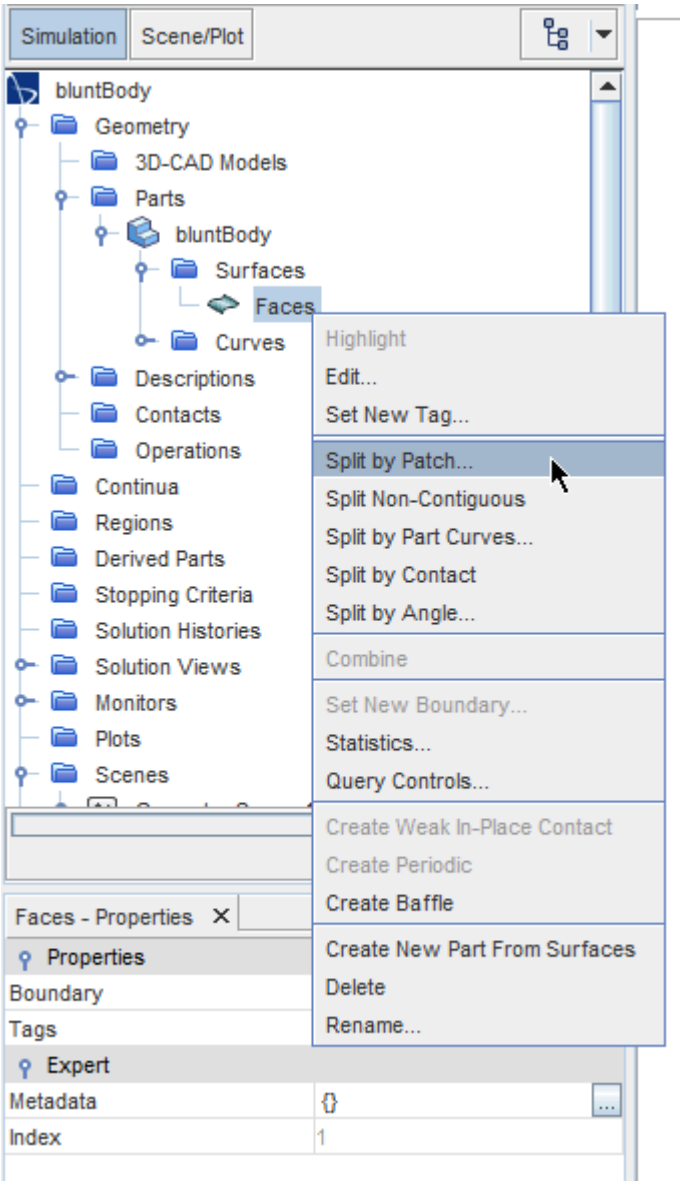
The following illustration gives an overview of how you set up this problem.



Index	Description
1	Stagnation Boundary
2	Wall
3	Symmetry Planes
4	Pressure Boundary
5	Slip Walls

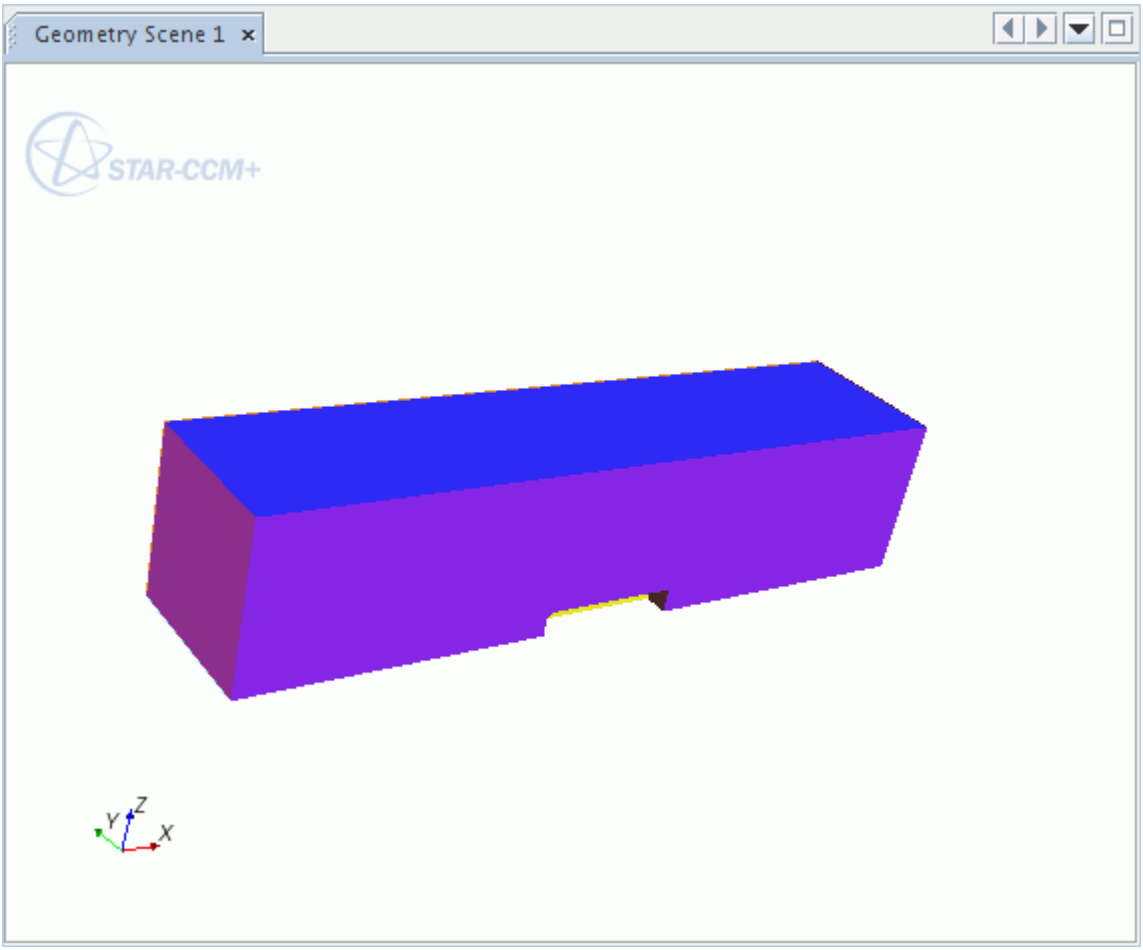
Split the surfaces that form specific boundaries:

1. Open the **Geometry** > **Parts** > **bluntBody** > **Surfaces** > **Faces** node.
2. Right-click the **Faces** node and from the pop-up menu, select **Split by Patch**.

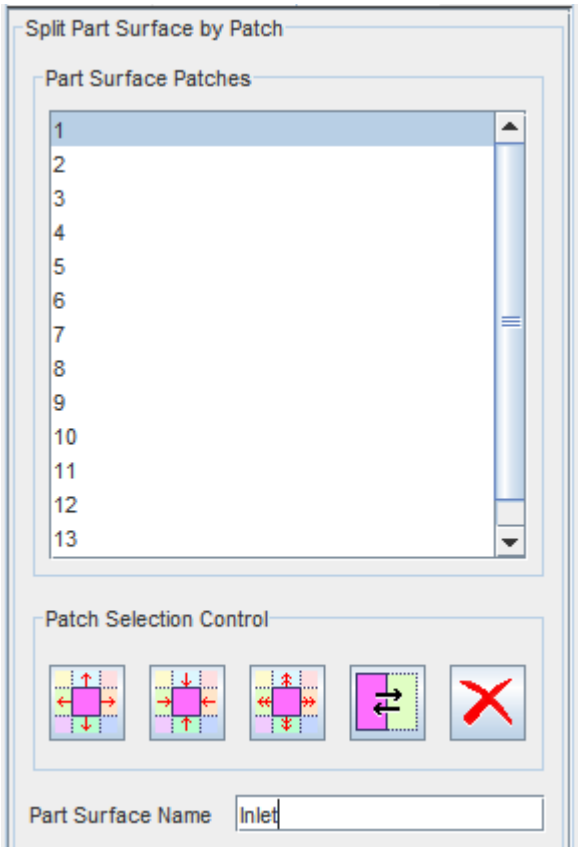


The *Split Part Surface by Patch* dialog appears.

3. In *Geometry Scene 1*, select the face in the low X direction, rotating the geometry as necessary.



4. Type `Inlet` in the **Part Surface Name** field.



5. Click **Create**.

   The face is removed from the scene and the new surface appears within the surfaces manager node belonging to the parent part.

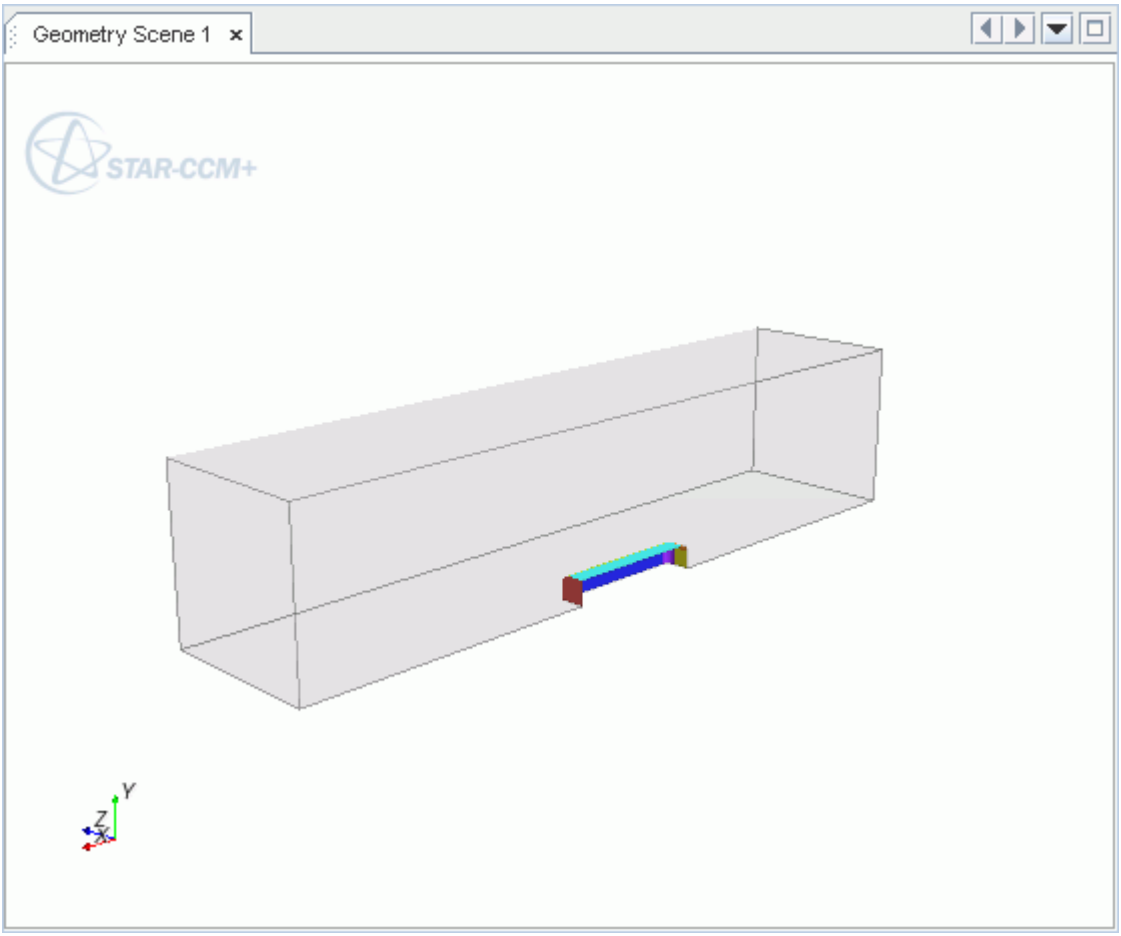
6. Repeat the previous steps for the following surfaces:

Location	Corresponding part surface patch number	Name
High X face	3	`Pressure`
Low Z face	4	`Symmetry_plane1`
Low Y face	6	`Symmetry_plane2`

The outside walls of the region can be defined as a single boundary by combining two faces into one part surface using the multi-select functionality.

7. Hold down the `<Ctrl>` key and click both the High Y and High Z faces.
8. Type `Slip_wall` in the **Part Surface Name**.
9. Click **Create**.

   The remaining patches remain as part of the default surface.



10. Click **Close**.
11. Save the simulation.

Contents:

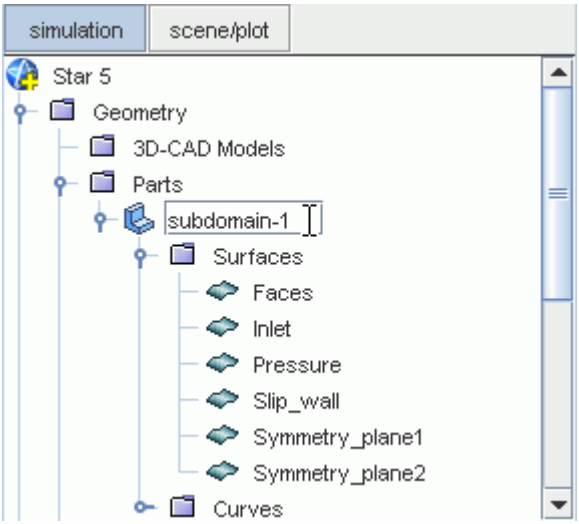
Renaming the Surface and Part

Renaming the Surface and Part

Renaming nodes is a common operation in STAR-CCM+. There are two ways a node can be renamed.

Rename the **Faces** node and the **bluntBody** part:

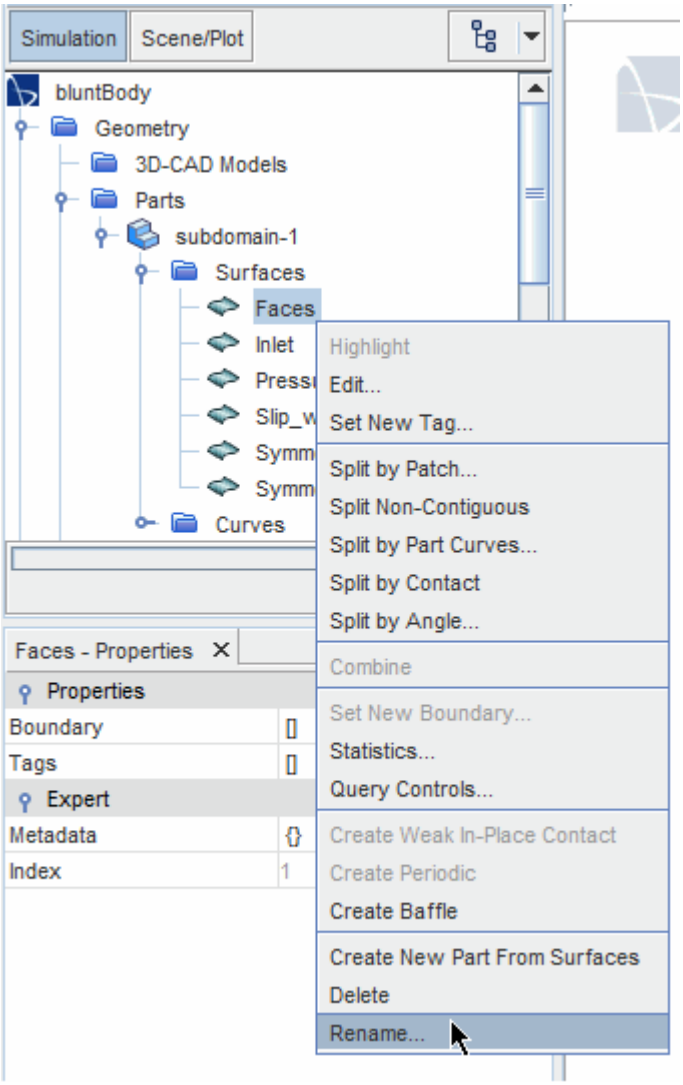
1. Select the **Geometry** > **Parts** > **bluntBody** node.
2. Press <`F2`>.
3. Rename the part `subdomain-1`.



4. Press <`Enter`> or click another node to accept.

To rename the Faces part surface, the previous method can be used. Use the alternative right-click menu option:

5. Right-click the **Geometry** > **Parts** > **subdomain-1** > **Surfaces** > **Faces** node and select **Rename...**.



6. Rename the part surface `Inner_wall`.

   The parts are ready to be converted to a region and boundaries.

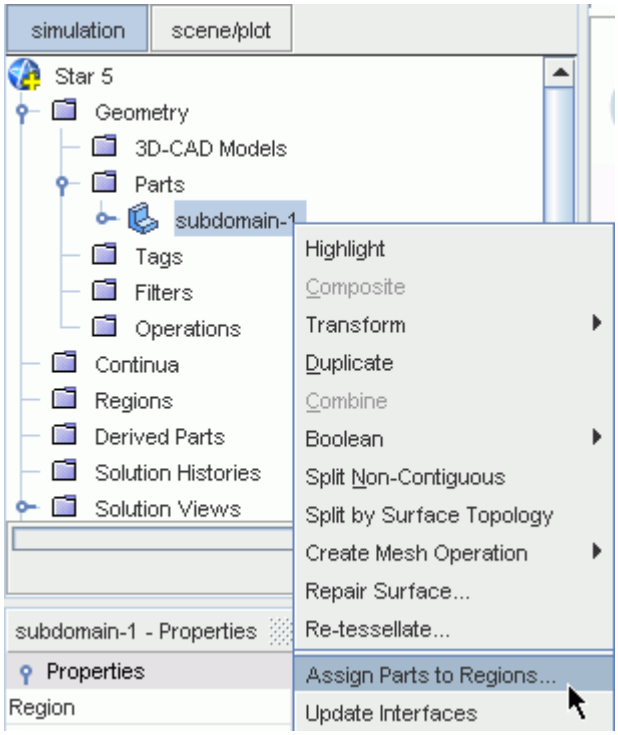
7. Save the simulation.

Assigning Parts to Regions

Geometry parts are used to prepare the spatial representation of the model. The computational model to which physics can be applied is defined in terms of regions, boundaries, and interfaces.

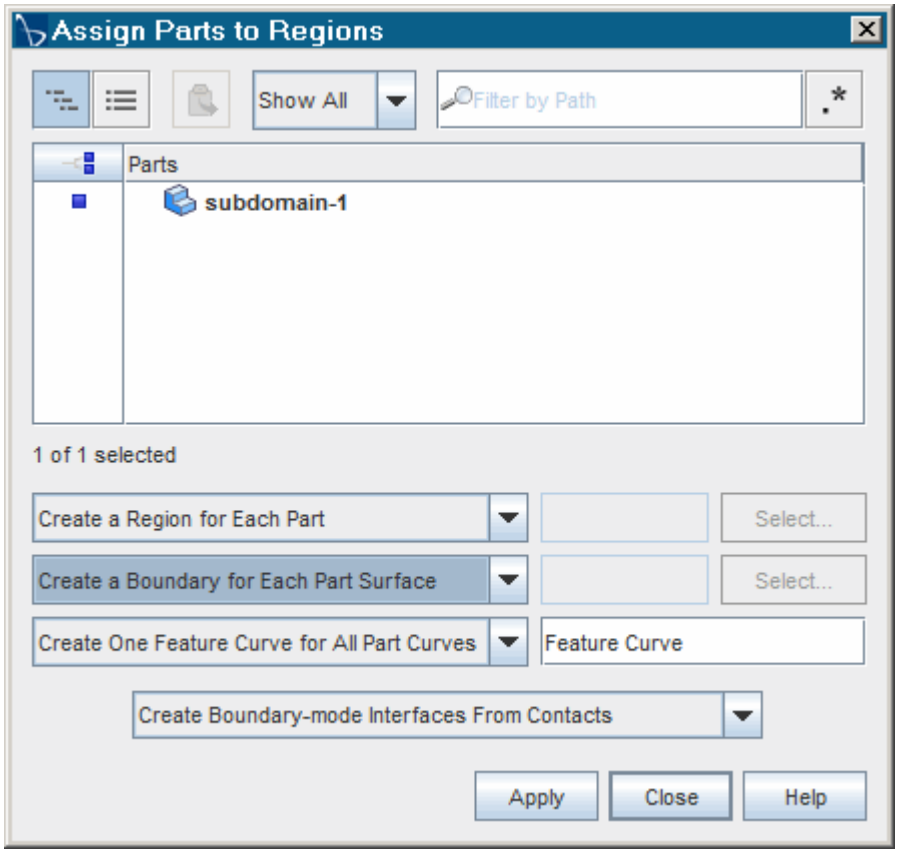
Create a region and associated boundaries from the geometry part and its surfaces:

1. Right-click **Geometry** > **Parts** > **subdomain-1** and select **Assign Parts to Regions...**.



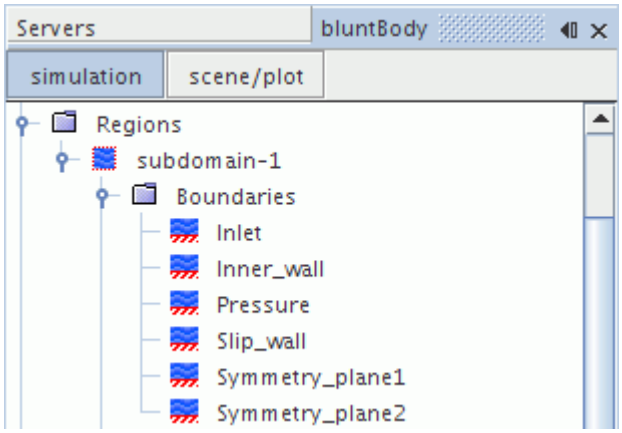
   The *Assign Parts to Regions* dialog appears.

2. Select **Create a Region for Each Part**.
3. Select **Create a Boundary for Each Part Surface**.



4. Click **Apply**.
5. Close the dialog.

   The portion of the object tree below the **Regions** node appears as shown below. All of the surfaces appear as individual boundaries within the region.



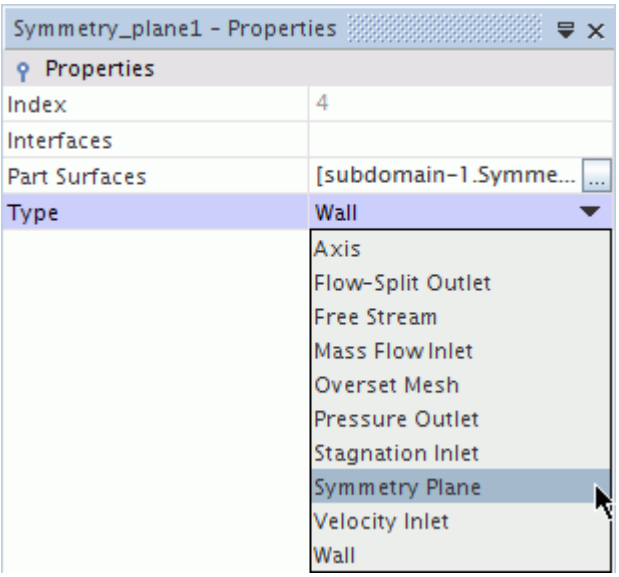
6. Save the simulation.

Setting Boundary Types

Having previously given the boundaries sensible names, you can set the boundary types.

Specify the appropriate boundary types:

1. Select the **Symmetry_plane1** node and set **Type** to **Symmetry Plane**.



   The boundary node icon changes to reflect the new type.

2. Using the same technique, change **Symmetry_plane2** to **Symmetry Plane**.

For compressible flows, the most appropriate inflow and outflow types are stagnation inlet and pressure outlet.

3. Change the type of the **Inlet** boundary to **Stagnation Inlet** and the type of the **Pressure** boundary to **Pressure Outlet**.

   The **Slip_wall** and **Inner_wall** boundaries retain the default **Wall** type. Slip walls are boundary conditions and are set up later.

4. Save the simulation.

Contents:

Selecting Parts
Adding and Removing Parts from a Scene

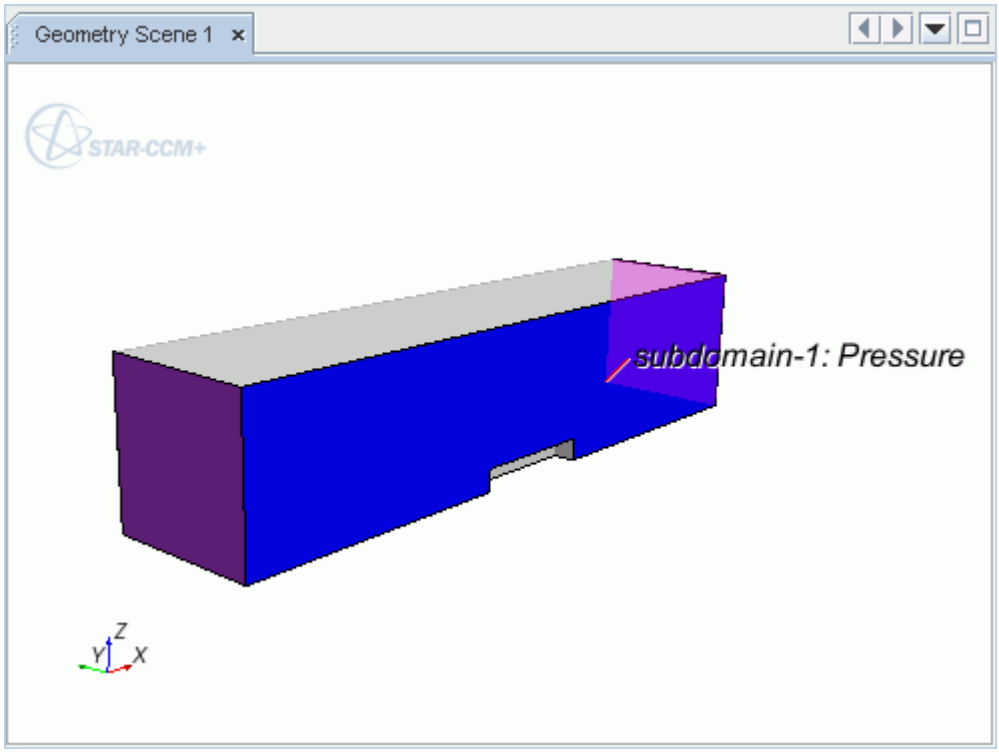
Selecting Parts

STAR-CCM+ highlights selected parts in the visualization window.

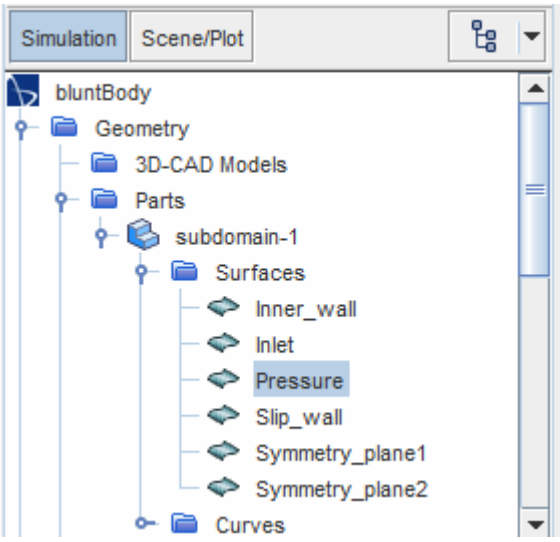
To highlight parts:

1. Click a part in the *Geometry Scene 1* display, for example the `High X` face.

   The object becomes highlighted and a label appears with the name of the object selected.



   In the **bluntBody** object tree, the node that corresponds to this object is also highlighted.



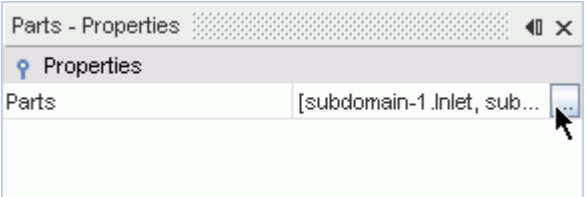
   Conversely, if a boundary node is selected in the **bluntBody** tree, the surface corresponding to that boundary is highlighted in the *Graphics* window.

Adding and Removing Parts from a Scene

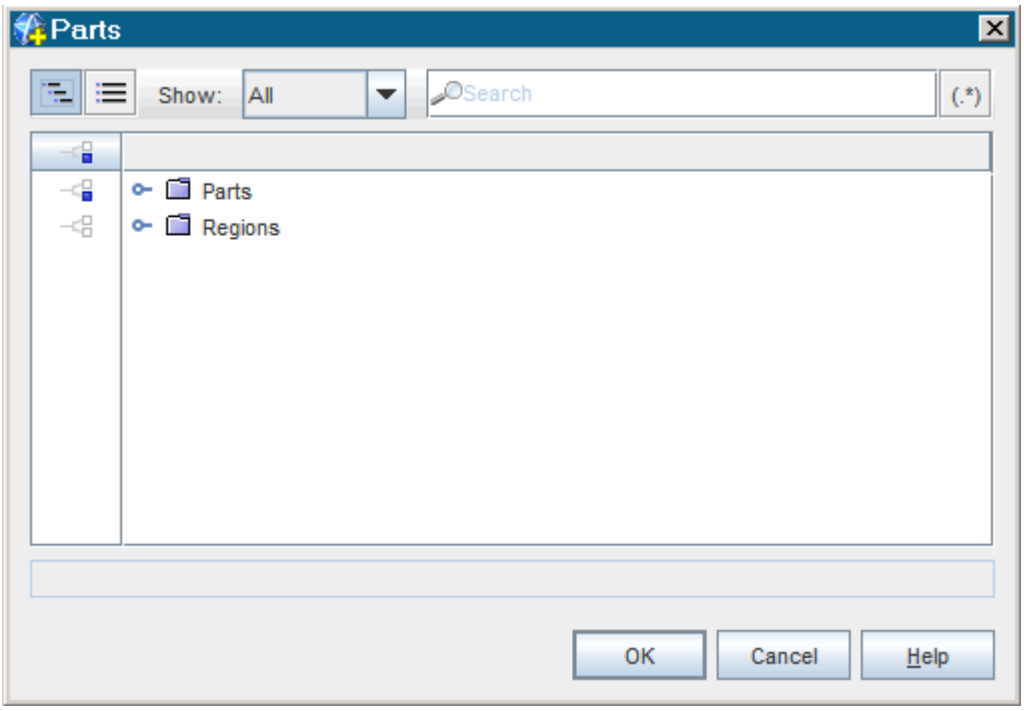
Use the part selector dialog to add and remove parts from a scene.

To add and remove some parts:

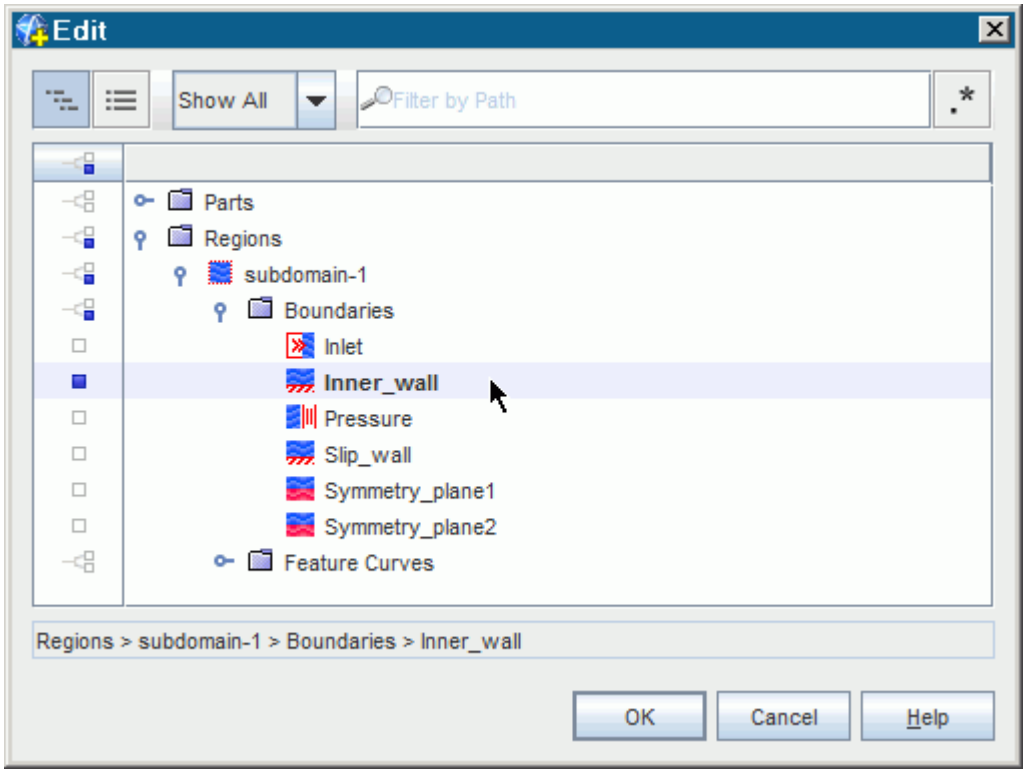
1. Open the **Scenes** > **Geometry Scene 1** > **Displayers** > **Geometry 1** > **Parts** node and click the ellipsis (**Custom Editor**) for the **Parts** value.



The *Parts* dialog appears.

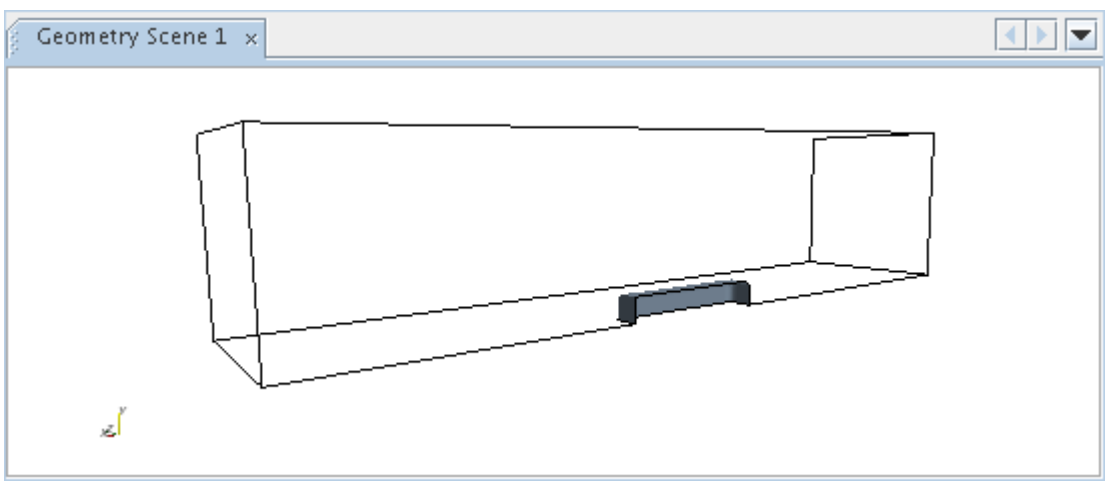


2. Click (**Column Selection**) twice to clear the current selections.
3. Expand **Regions** > **subdomain-1** > **Boundaries**.
4. Select **Inner_wall**.



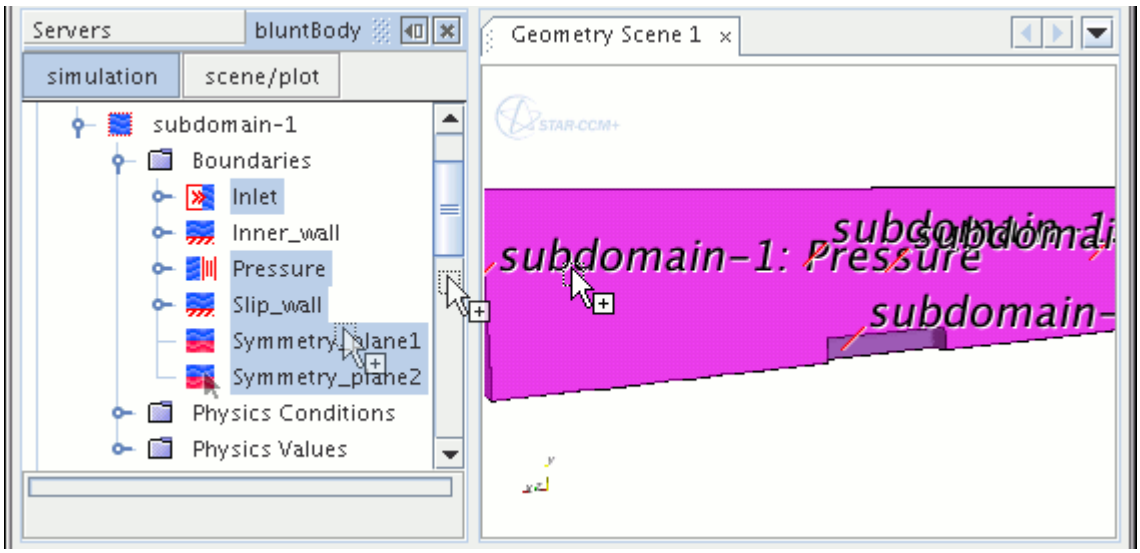
5. Click **OK**.

   This selection removes the solid-colored boundaries representing the wind tunnel from the scene, leaving the outlines and the blunt body.



Add those parts back to the scene using a drag-and-drop method:

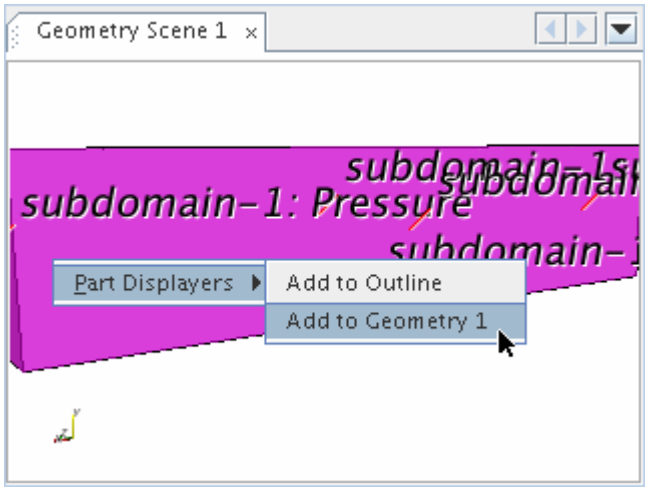
6. Open the **Regions** > **subdomain-1** > **Boundaries** node.
7. To extend your selection, hold down the <Ctrl> key and select all the boundary nodes except **Inner_wall**.
8. Release the <Ctrl> key but continue to keep the left mouse button pressed.
9. Drag the nodes onto the display and release the left mouse button.



A pop-up menu appears.

Use this menu to choose which of the part displayers in the scene receive the parts that you have dragged across.

10. Select **Add to Geometry 1**.



The parts are restored to the scene.

11. Save the simulation.

Mesh Generation

Several steps are involved in defining and generating a volume mesh.

To generate the mesh:

1. Create an Automated Mesh operation.
2. Choose appropriate meshers.
3. Specify initial global settings.
4. Modify the boundary-specific settings.

Contents:

Selecting the Meshers
Specifying the Mesh Settings
Generating the Volume Mesh
Displaying the Volume Mesh
Customizing the Mesh

See Also:

Selecting the Meshers
Specifying the Mesh Settings
Generating the Volume Mesh
Customizing the Mesh

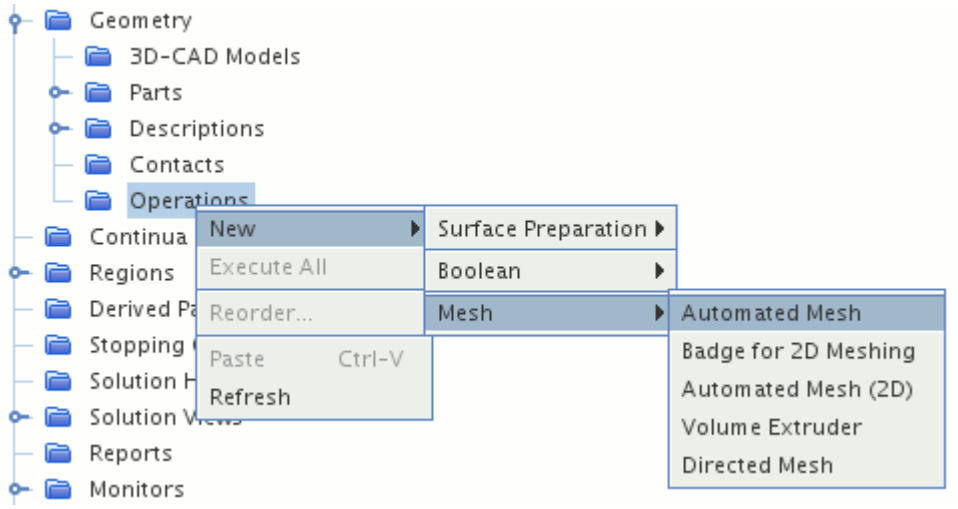
Selecting the Meshers

You select different meshers for the generation of the surface mesh, the volume mesh, and the prism layers that are created at the wall boundaries of the computational domain. To select the required meshers, you create an Automated Mesh operation.

The selected meshers within this Automated Mesh operation apply to one or more selected geometry parts. You can create several Automated Mesh operations containing different meshers and mesh settings. Here, you create only one Automated Mesh operation that applies to one geometry part.

To select the meshers:

1. Right-click **Geometry** > **Operations** and select **New** > **Mesh** > **Automated Mesh**.

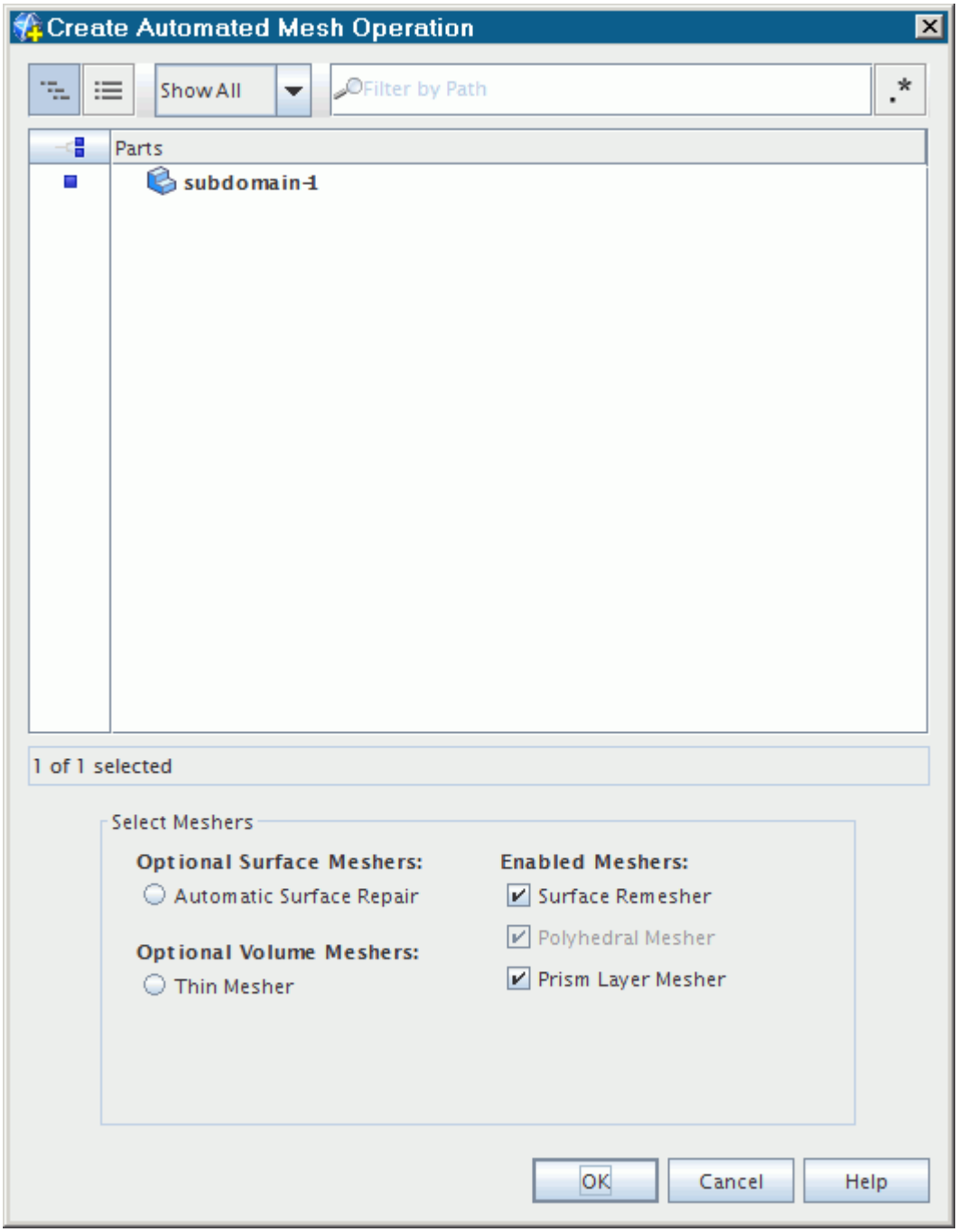


2. In the *Create Automated Mesh Operation* dialog:
   a) Select **subdomain-1** from the *Parts* list.

      The subsequently selected meshers apply to this selected part.

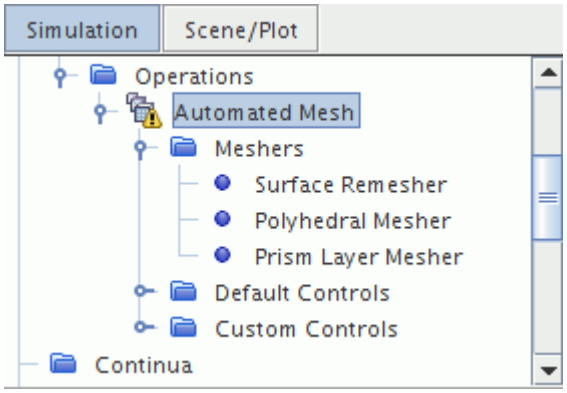
   b) Select the following meshers, in order:

Group	Mesher
Surface Meshers	**Surface Remesher**
Core Volume Meshers	**Polyhedral Mesher**
Optional Boundary Layer Meshers	**Prism Layer Mesher**



c) Click **OK**.

3. Expand the **Operations** > **Automated Mesh** > **Meshers** node.



Specifying the Mesh Settings

Generating a mesh often requires several iterations to achieve the desired density and distribution of cells. In this tutorial, you generate a volume mesh, add some simple refinement, and regenerate.

To specify the mesh settings:

1. Right-click the **Geometry** > **Operations** > **Automated Mesh** > **Default Controls** node and select **Edit...**.
2. In the *Default Controls* dialog, set the following properties:

Node	Property	Setting
Base Size	**Value**	`0.01 m`
Numbers of Prism Layers	**Numbers of Prism Layers**	`5`

With this base size, approximately 8 cells are created across the width of the region.

Generating the Volume Mesh

Now that you have specified the appropriate settings, you can generate the volume mesh.

1. Click (**Generate Volume Mesh**) in the toolbar or select **Generate Volume Mesh** in the **Mesh** menu.

   The run and progress of the meshers are displayed in the *Output* window.

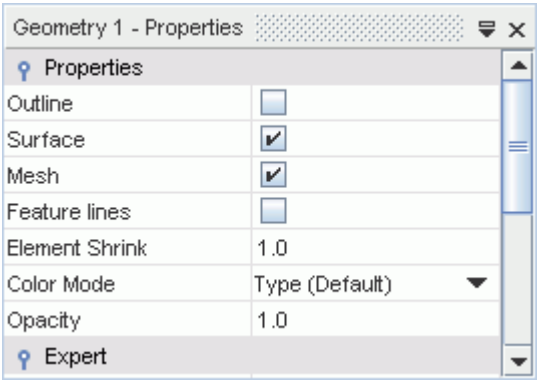
```
Output - bluntBody

The following regions were moved into the "Parts Meshes" continuum because
they are associated with geometry parts that are assigned to a volume meshing operation:
  subdomain-1

Converting mesh into finite volume representation in Region subdomain-1
   Converted mesh for subdomain-1 CPU Time: 0.55, Wall Time: 0.55, Memory: 104.95 MB
Mesh conversion completed, CPU Time: 0.86, Wall Time: 0.86, Memory: 92.23 MB
Loading module: MetricsModel
Loading module: MotionModel
Volume Meshing Pipeline Completed: CPU Time: 8.28, Wall Time: 8.28, Memory: 94.61 MB
        Cells: 17788        Faces: 87918        Vertices: 64648
----------------------------------------
Automated Mesh Operation: Automated Mesh complete. CPU Time: 8.36, Wall Time: 8.36, Memory: 94.61 MB
----------------------------------------
Volume Meshing Pipeline Completed: CPU Time: 11.17, Wall Time: 11.17, Memory: 94.61 MB
        Cells: 17788        Faces: 87918        Vertices: 64648
```

Displaying the Volume Mesh

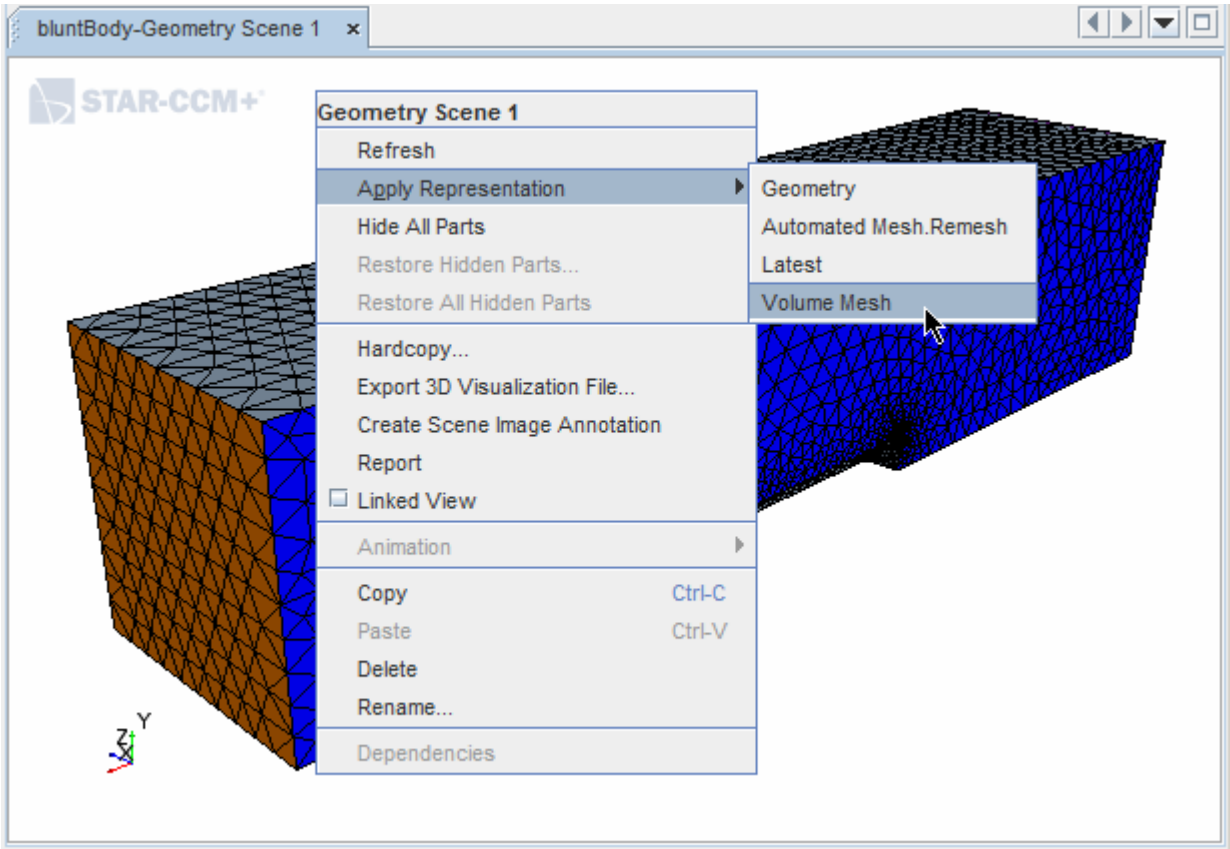
Display the mesh to permit its inspection in the existing displayer.

To display the volume mesh:

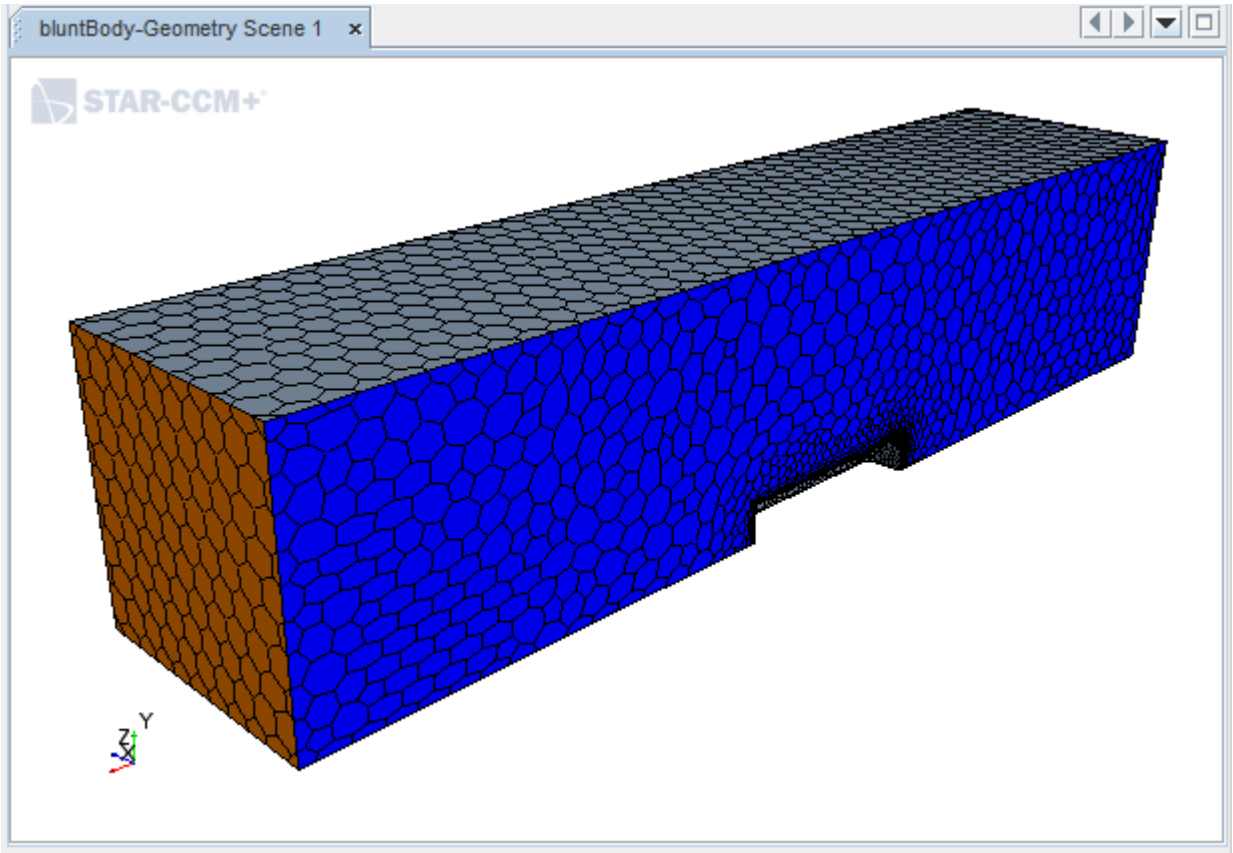
1. Select the **Scenes** > **Geometry Scene 1** > **Displayers** > **Geometry 1** node.
2. To display the surface of the mesh, activate **Mesh**.



3. Right-click an empty part of the display and select **Apply Representation** > **Volume Mesh**.



This action reveals the mesh on the boundaries.



4. Save the simulation.

Customizing the Mesh

You can further customize the mesh to modify the mesh settings only on specific surfaces.

By default, the prism layers are generated on the inner and the slip walls by default. However, a fluid boundary layer does not form on slip walls and prism layers are not required. You can customize the prism mesh settings on that boundary to disable the prism layer generation.

To better resolve certain areas of the geometry, you can locally reduce the cell size on a surface.

To disable the prism layer generation on the slip wall:

1. Right-click the **Geometry** > **Operations** > **Automated Mesh** > **Custom Controls** node and select **New** > **Surface Control**.
2. Right-click the **Custom Controls** > **Surface Control** node and select **Edit**.
3. Set the following properties:

Node	Property	Setting
Surface Control	*Part Surfaces*	**Slip_wall**
└ **Controls** > **Prism Layers**	*Prism Layers*	**Disable**

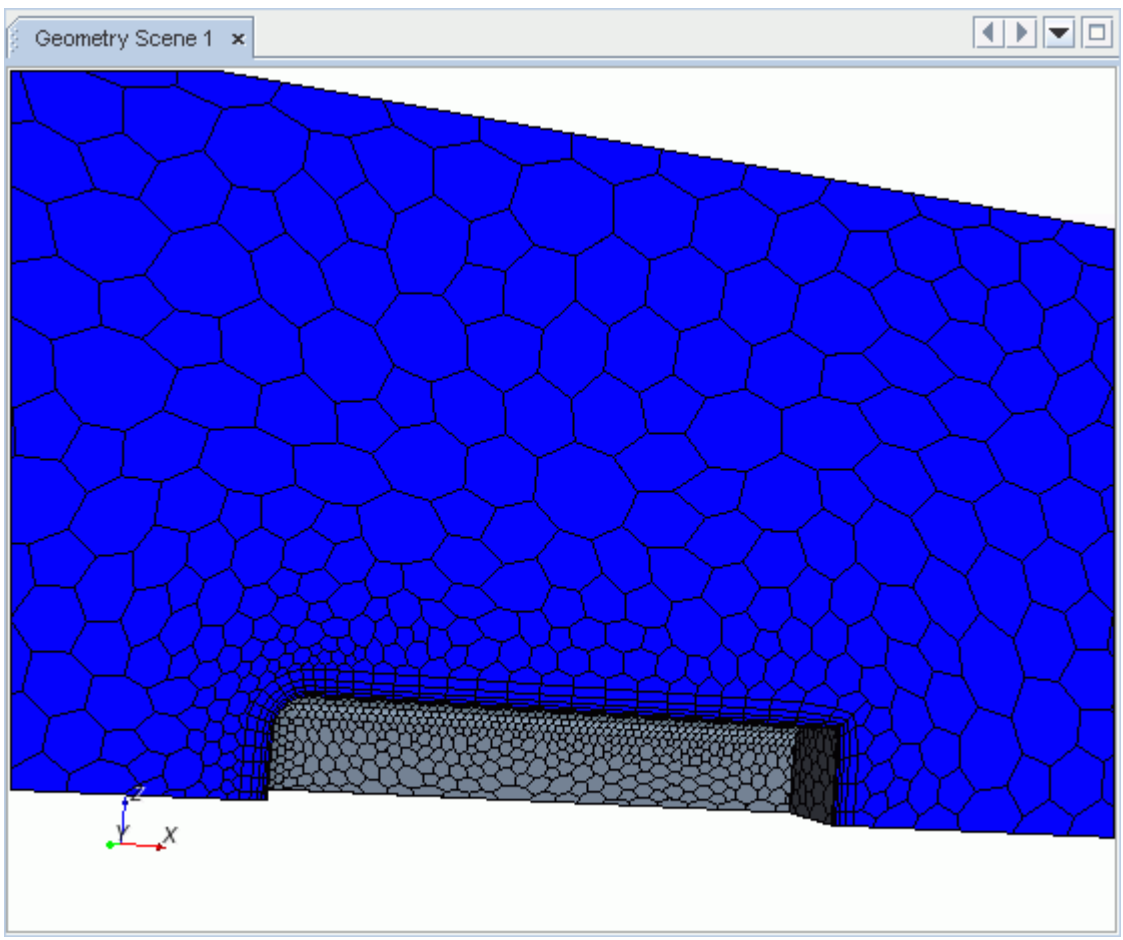
To provide better definition of the blunt body, modify the surface mesh at the inner wall:

4. Right-click the **Automated Mesh** > **Custom Controls** node and select **New** > **Surface Control**.
5. Right-click the **Custom Controls** > **Surface Control 2** node and select **Edit**.
6. Set the following properties:

Node	Property	Setting
Surface Control 2	*Part Surfaces*	**Inner_wall**
└ **Controls** > **Minimum Surface Size**	*Minimum Surface Size*	**Custom**
└ **Values** > **Minimum Surface Size**	*Absolute*	**0.001 m**

To regenerate the mesh:

7. Click (**Generate Volume Mesh**).
8. Zoom in to the area around the slip walls and inner walls to see the improved mesh.



This mesh is satisfactory for an initial solution.

9. Save the simulation.

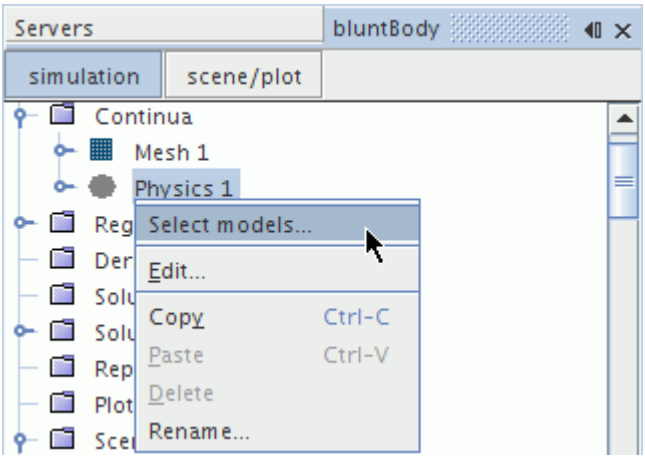
Selecting the Physics Models

Physics models define the primary variables of the simulation, including pressure, temperature and velocity, and the mathematical formulation.

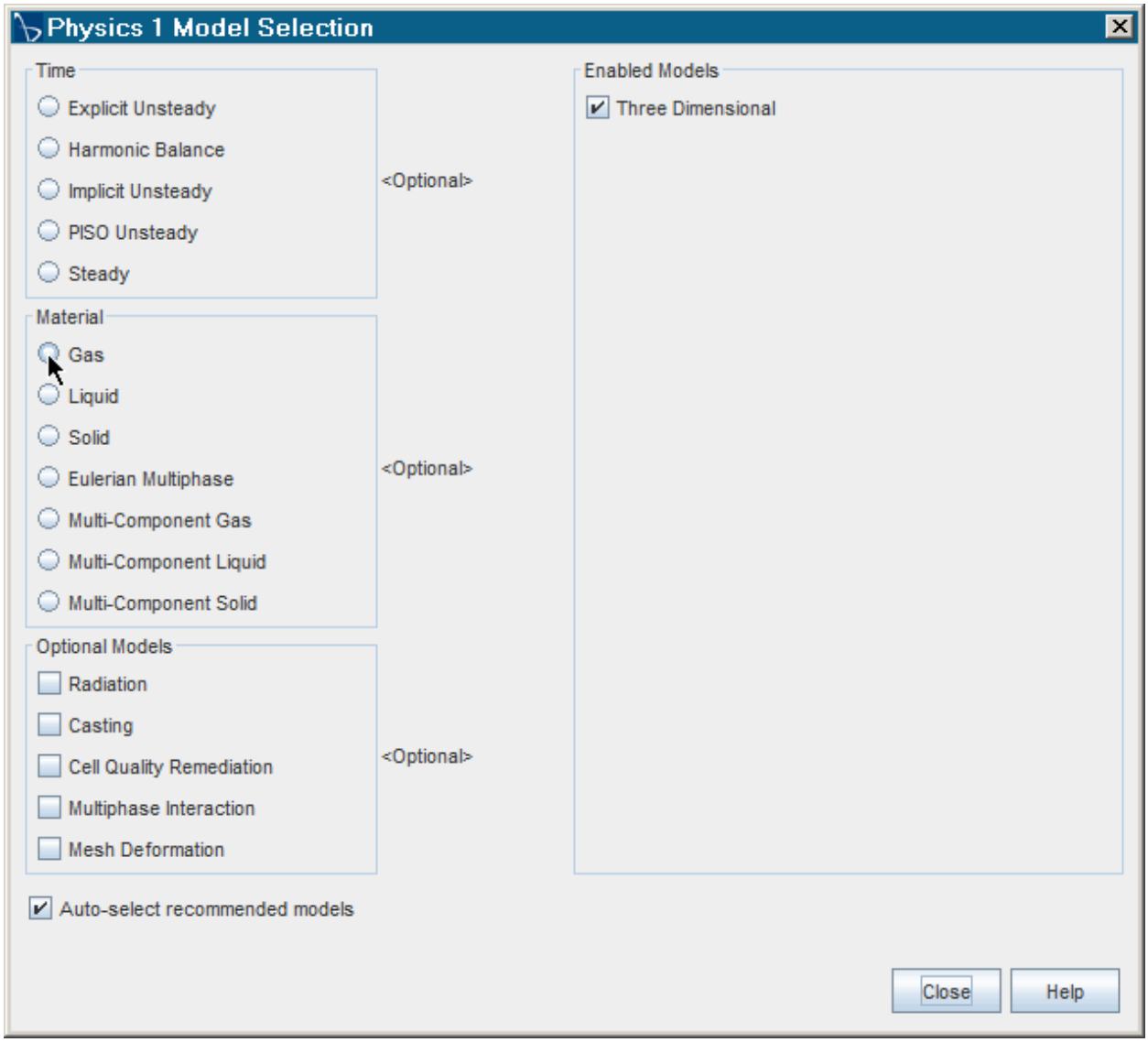
In STAR-CCM+, the physics models are defined on a physics continuum. In this example, the flow is turbulent and compressible. You use the Coupled Flow model together with the default K-Epsilon Turbulence model.

To select the physics models:

1. Right-click the **Continua** > **Physics 1** node and choose **Select models...**.



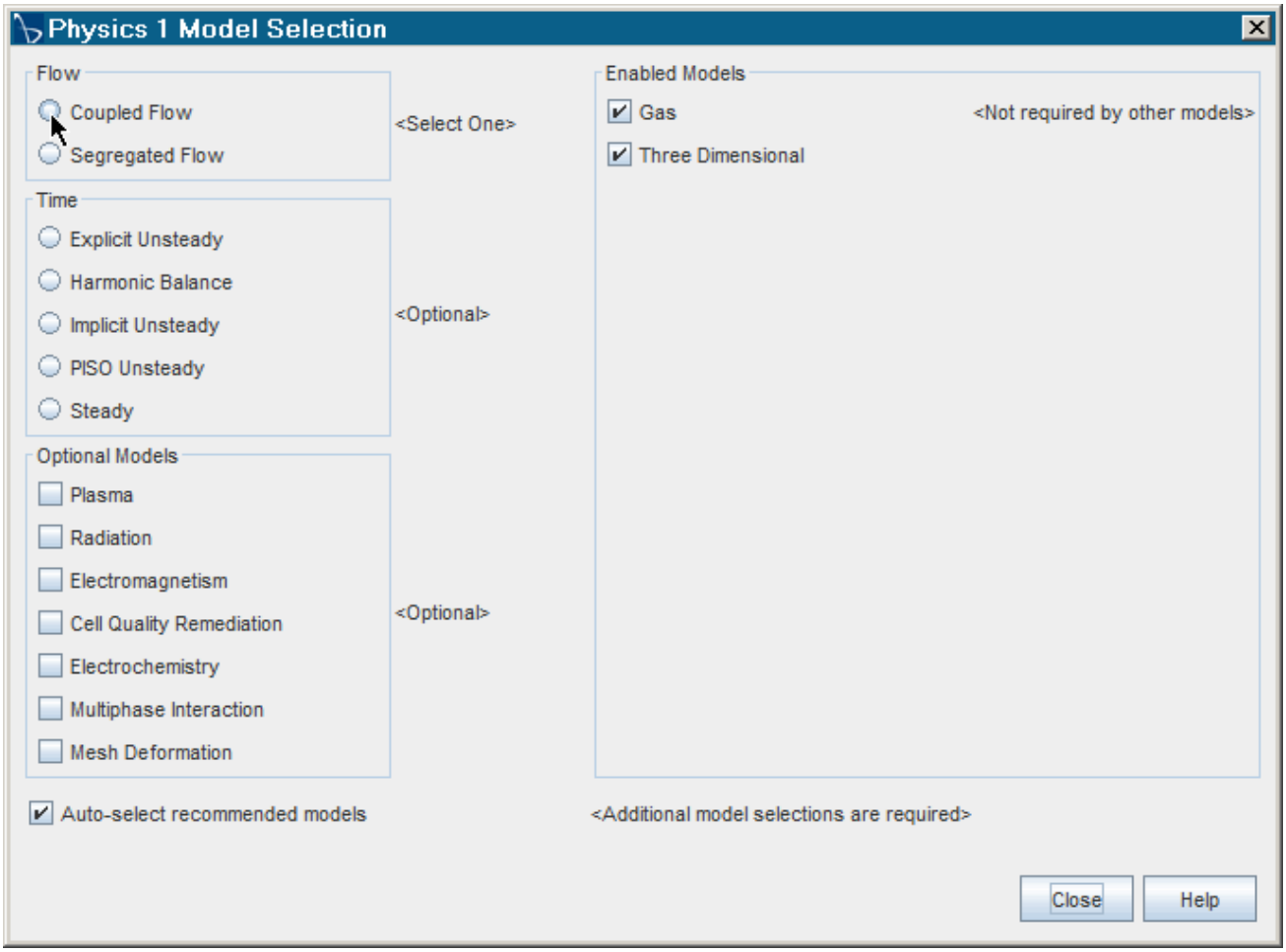
The *Physics Model Selection* dialog appears as shown in the following screenshot.



2. Select the **Gas** radio button from the *Material* group box, since this exercise involves an idealized gas (air).

Since the **Auto-select recommended models** checkbox is activated, the *Physics Model Selection* dialog guides you through the model selection process by selecting certain default models automatically as you make some choices.

Certain models, when activated in a continuum, require other models also to be activated in that continuum. For instance, once a continuum contains a liquid or a gas, it also needs a flow model. Once it has a flow model, it needs a viscous model (inviscid, laminar, or turbulent). Once turbulence is activated within a fluid continuum, select a turbulence model. The prompt **Additional model selections are required** alerts you to the fact that you have not completed the model selection.



3. The following selections are required for this simulation:
   a) Select **Coupled Flow** from the *Flow* group box.
   b) Select **Ideal Gas** from the *Equation of State* group box.

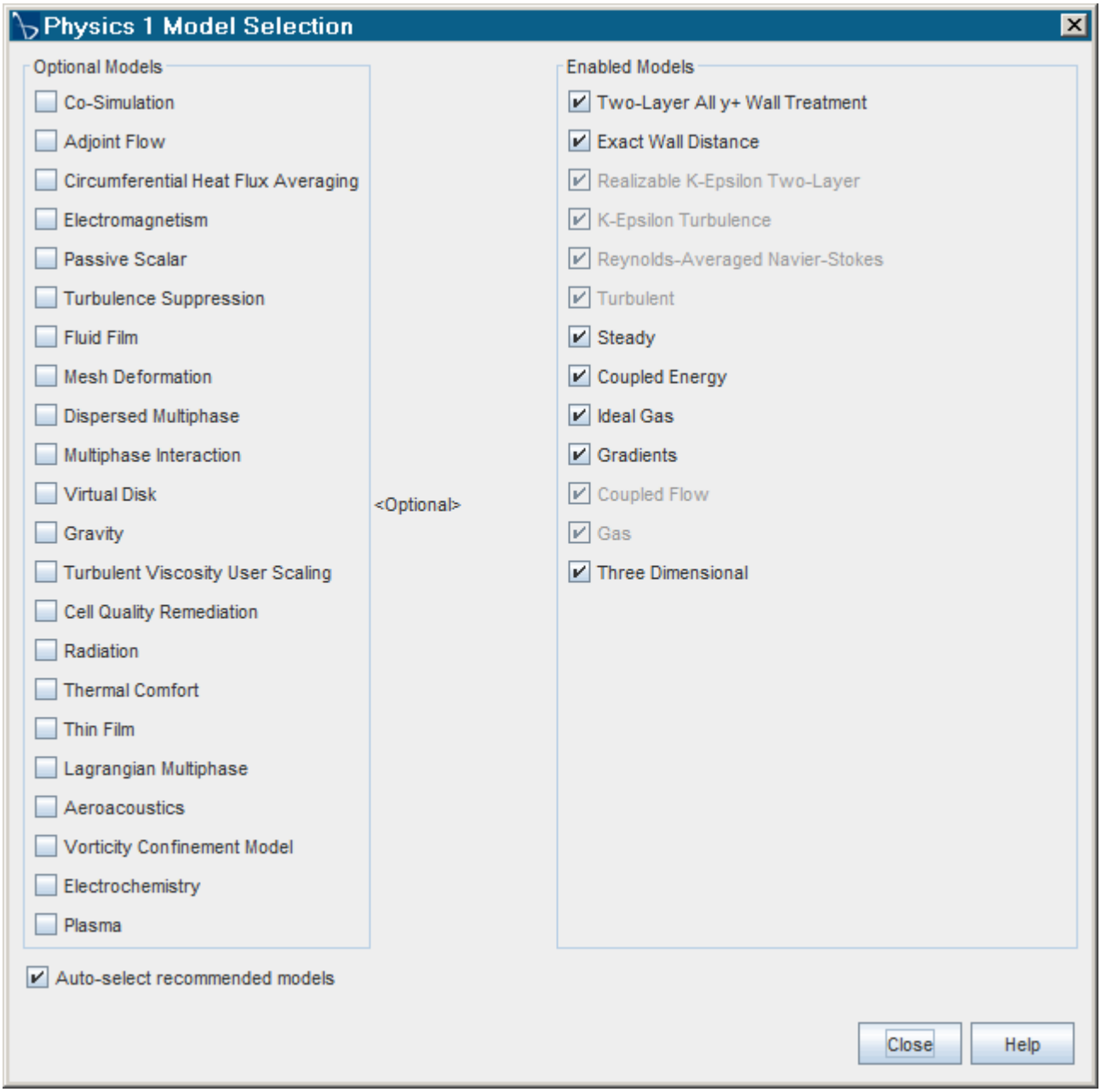
      The **Coupled Energy** model is selected automatically.
   c) Select **Steady** from the *Time* group box.
   d) Select **Turbulent** from the *Viscous Regime* group box.

      The **Reynolds-Averaged Navier-Stokes** model is selected automatically.
   e) Select **K-Epsilon Turbulence** from the *Reynolds-Averaged Turbulence* group box.

      The **Realizable K-Epsilon Two-Layer**, **Exact Wall Distance**, and **Two-Layer All y+ Wall Treatment** models are selected automatically.

To reverse part or all of the model selection process, simply clear the checkboxes of the models you wish to deactivate. Other active models require the selections that are grayed out. Therefore, deactivate the models that are not grayed out to begin with.

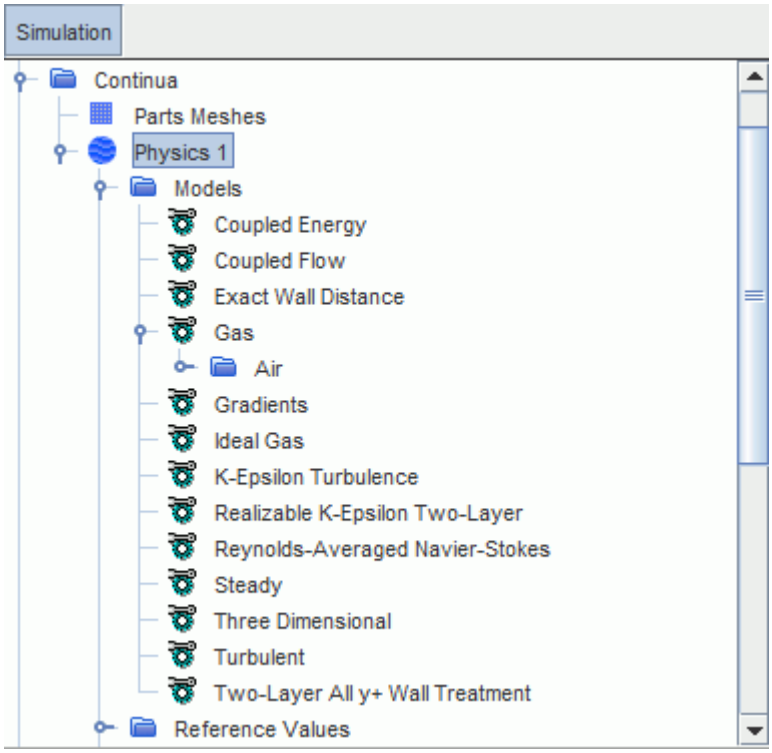
When complete, the *Physics Model Selection* dialog appears as shown in the following screenshot:



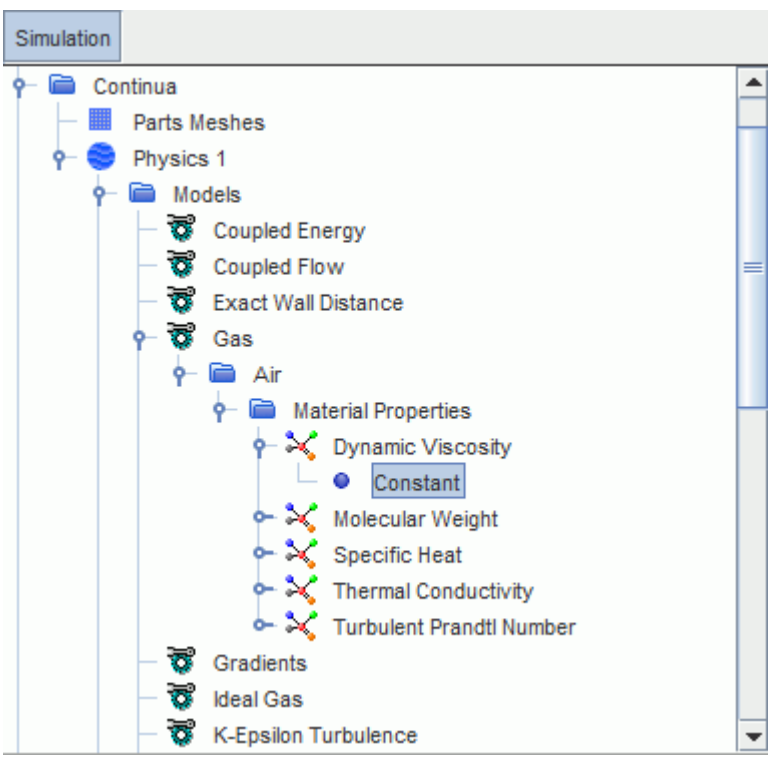
No optional models are required for this simulation.

4. Click **Close**.
5. To see the selected models in the simulation tree, open the **Continua** node in the *bluntBody* window of the *Explorer* pane.

   The color of the **Physics 1** node has turned from gray to blue to indicate that models have been selected.
6. To display the selected models, open the **Physics 1** node and then the **Models** node.



You can view the model properties in the object tree.



When you select the **Gas** model, the properties of air such as dynamic viscosity are used by default. Since this problem uses air, the properties are acceptable.

7. Save the simulation.

Setting Initial Conditions

Initial conditions in a continuum specify the initial field data for the simulation.

Examples of initial conditions are:

- Pressure
- Temperature
- Velocity components
- Turbulence quantities

Each model requires sufficient information so that the primary variables of the model can be set. For some models, such as turbulence models, the option of specifying the information in a more convenient form is presented. For example, turbulence intensity and turbulent viscosity ratio instead of turbulent kinetic energy and turbulent dissipation rate.

In steady-state simulations, the solution ought to converge independently of the initial field. However, the initial field still affects the path to convergence, and with it the cost in computing power. Therefore specify initial conditions and values judiciously, particularly when the physics is complex.

Contents:

Specifying the Initial Conditions for the Simulation

Specifying the Initial Conditions for the Simulation

The stagnation inlet boundary has conditions that correspond to a Mach number of 0.75. The equivalent freestream velocity is roughly 300 m/s, the value you use to initialize the velocity field.

To specify the initial conditions:

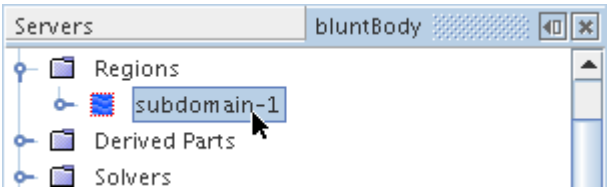
1. Open the **Continua** > **Physics 1** > **Initial Conditions** node.
2. Select the **Velocity** node and set **Value** to `[300.0, 0.0, 0.0]`.
3. Select the **Turbulent Viscosity Ratio** node and set **Value** to `50`, which is the same as the turbulent viscosity ratio that is set on the stagnation boundary condition.
4. Save the simulation.

Defining the Region Continuum

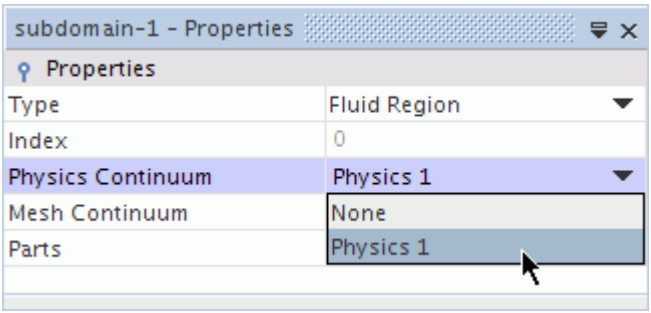
You can associate a continuum with zero, one, or more regions. Multiple continua can be defined and each region assigned a different one. In this tutorial, only one geometry part is present, which requires a single region to represent it, which was named `subdomain-1`. There is a simple way to determine which set of continua models are allocated to a region.

To define the region continuum:

1. Select the **subdomain-1** node in the **Regions** node.



2. Set the **Physics Continuum** property to **Physics 1**, the name of the continuum where you defined all the models earlier.



During this tutorial, there is one continuum in the above drop-down list, but as you add more (to the **Continua** node in the simulation tree), they are added to the drop-down list. This process determines which continua is used in the analysis, though continua can be defined and left unused.

Setting Boundary Conditions and Values

Although the concept of a boundary type is fairly unambiguous (wall, stagnation inlet, and so on.), models need more information to deal with the type. Conditions provide this information.

For example, with a boundary of type wall, the conditions specify whether this wall is to be a no-slip wall, a slip wall, or a moving wall. The conditions also tell you whether you want to apply a specified temperature (Dirichlet) thermal boundary condition or a specified heat flux (Neumann) thermal boundary condition.

The types and conditions inform models how to deal with a boundary (or region or interface) but they do not specify actual numerical input. Values provide this input. Value nodes are added in response to choices made on conditions nodes.

Contents:

Setting Inlet Conditions and Values
Setting the Slip Wall Condition

Setting Inlet Conditions and Values

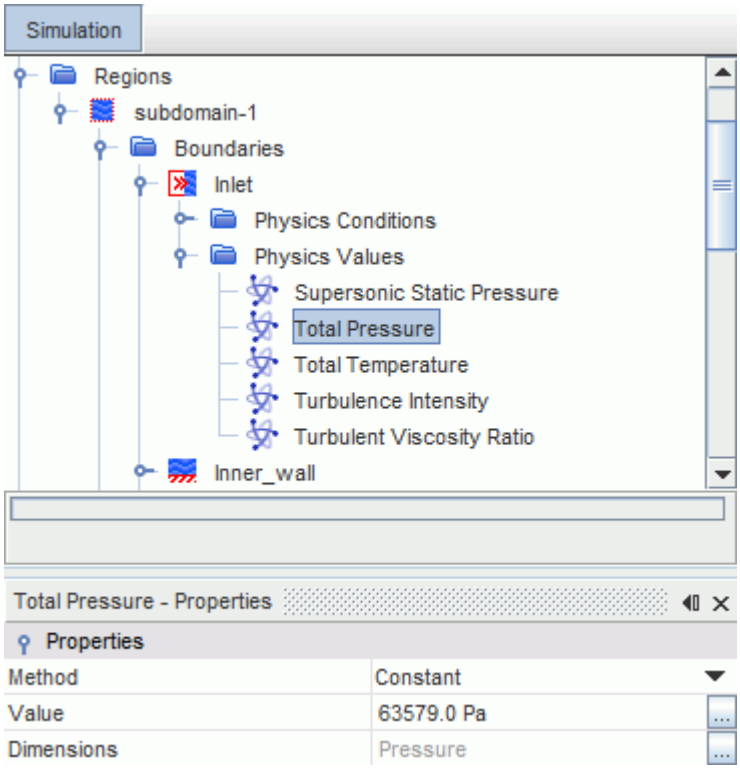
Set the appropriate conditions and values for the Inlet boundary.

To achieve a Mach number of approximately 0.75 at the inlet, isentropic relations were used to determine the inlet total pressure and outlet static pressure for a given total temperature. For an outlet static pressure equal to one atmosphere (absolute) and a static temperature of 300 K, the inlet total pressure is 164,904 Pa (absolute). The inlet total temperature is 344.8 K.

The boundary values are specified as gauge pressures. Therefore, using the default reference pressure of 101,325 Pa (one atmosphere), set the inlet total pressure to 63,579 Pa (gauge). The outlet static pressure is 0 Pa (gauge), the default value.

To set the Inlet conditions and values:

1. Open the **Regions** > **subdomain-1** > **Boundaries** node.
2. Open the **Inlet** > **Physics Values** node, and select the **Total Pressure** node.

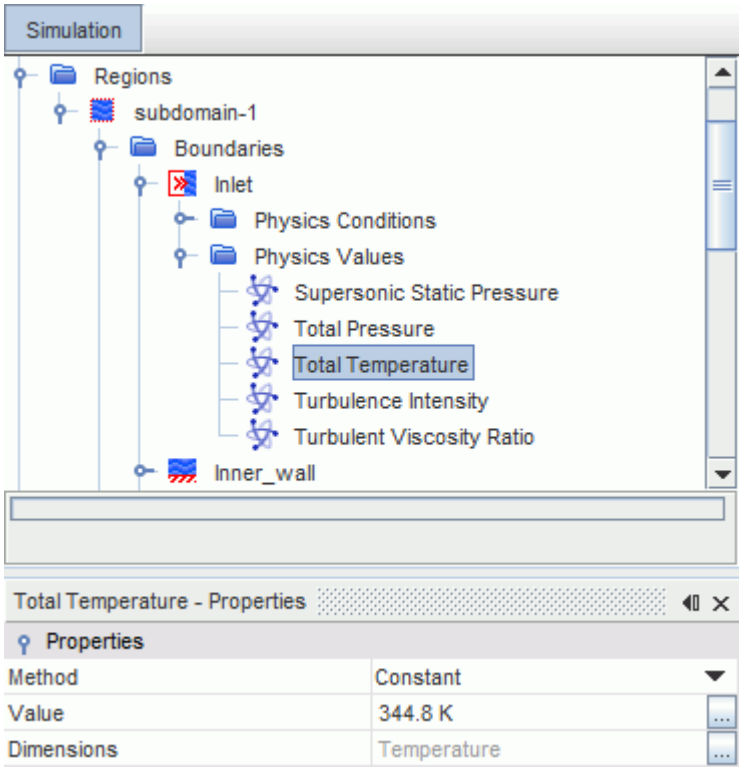


3. Set **Value** to `63579`.

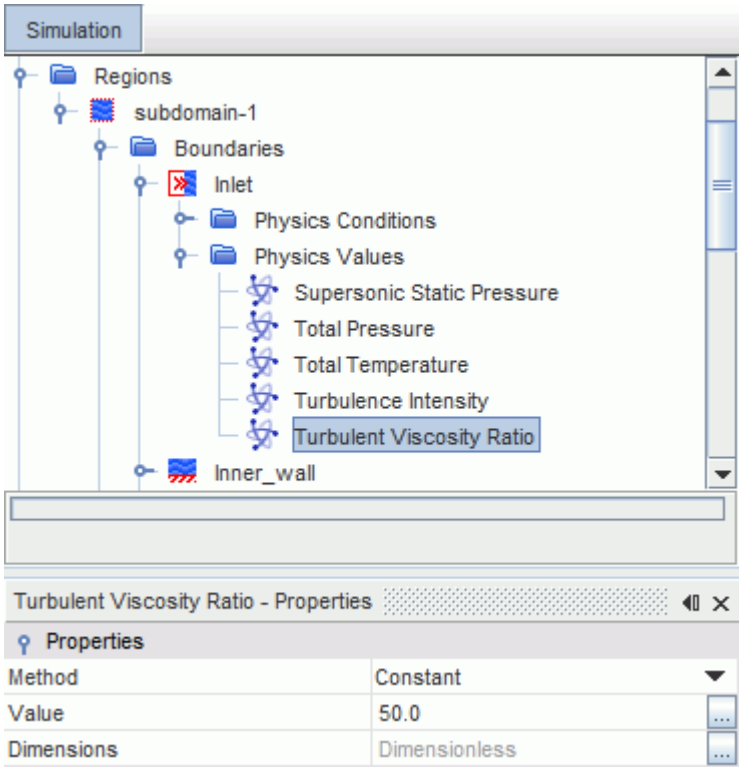
The total pressure setting is relative to the operating pressure value of 101,325 Pa. A supersonic static pressure is also required as part of a stagnation inlet condition but is only used if the inlet velocity becomes supersonic at some instance during solution iteration. The default value of 0.0 Pa relative pressure is sufficient as long as supersonic flow does not occur.

The next value to set is the total temperature.

4. In the same **Physics Values** node, select the **Total Temperature** node.



5. Set the **Value** to `344.8`.
6. In the same **Physics Values** node, select the **Turbulent Viscosity Ratio** node.



7. Set **Value** to `50`.

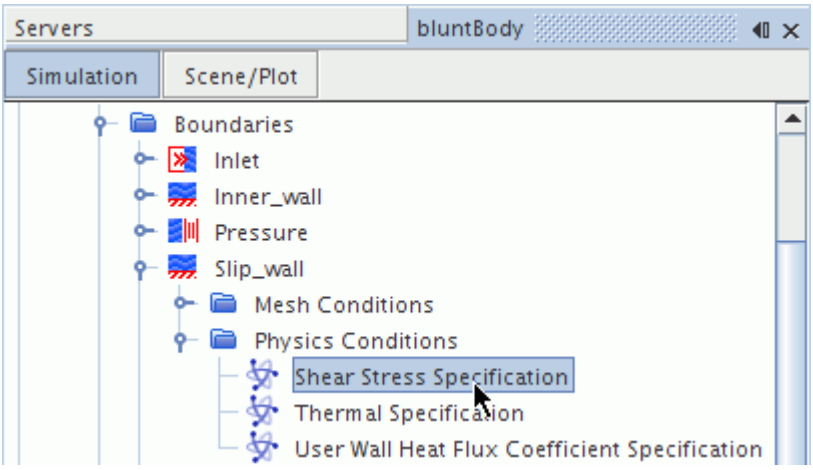
   This value has been determined to give a reasonable decay of turbulence in the channel core (for the default turbulence intensity).

Setting the Slip Wall Condition

To simulate the body in a wind tunnel, the upper boundary is set to a slip wall, thus avoiding the need to resolve the boundary layer on this wall.

To set the Slip Wall condition:

1. Select the **Slip_wall** > **Physics Conditions** > **Shear Stress Specification** node.



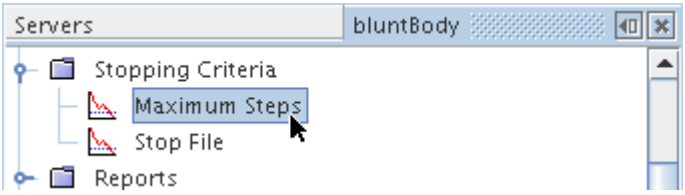
2. Set **Method** to **Slip**.

Setting Solver Parameters and Stopping Criteria

Set the appropriate stopping criteria for the simulation.

To set the stopping criterion:

1. Select the **Stopping Criteria** > **Maximum Steps** node.



2. Set the **Maximum Steps** to `300`.

The solution does not converge in this number of iterations. The solution does not run for more than 300 iterations, unless this stopping criterion is changed or deactivated.

3. Save the simulation.

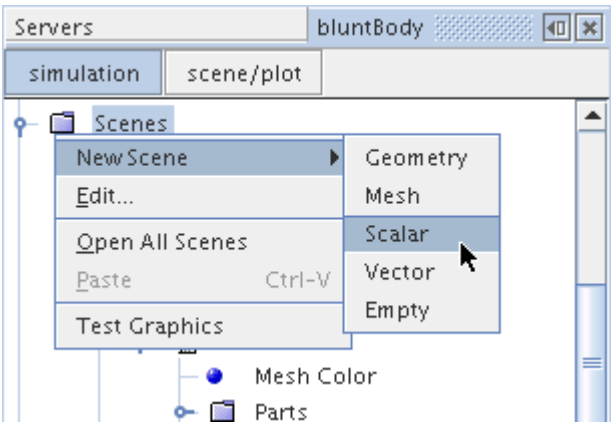
Visualizing the Solution

Create a scalar scene to display the results of the simulation.

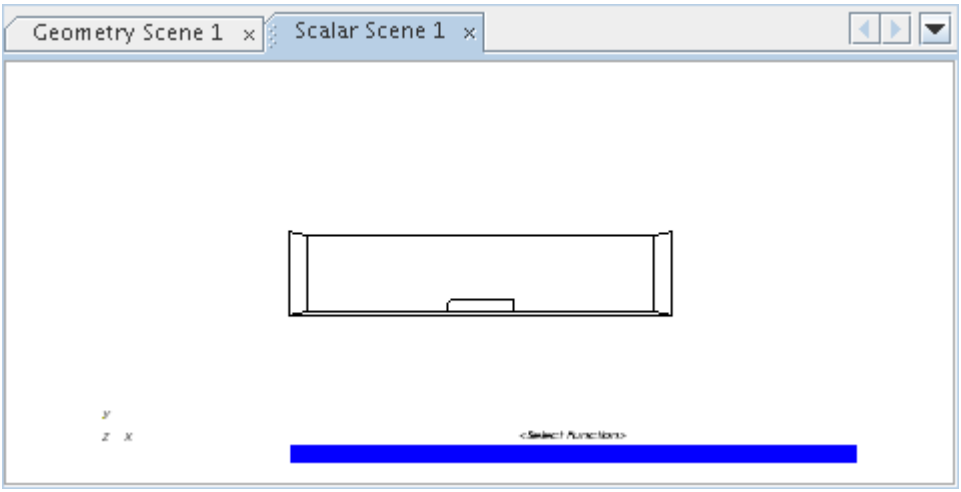
To watch the Mach number on the blunt body and vertical symmetry plane solution developing, set up a scalar scene.

To visualize the solution:

1. Right-click the **Scenes** node and select **New Scene** > **Scalar**.

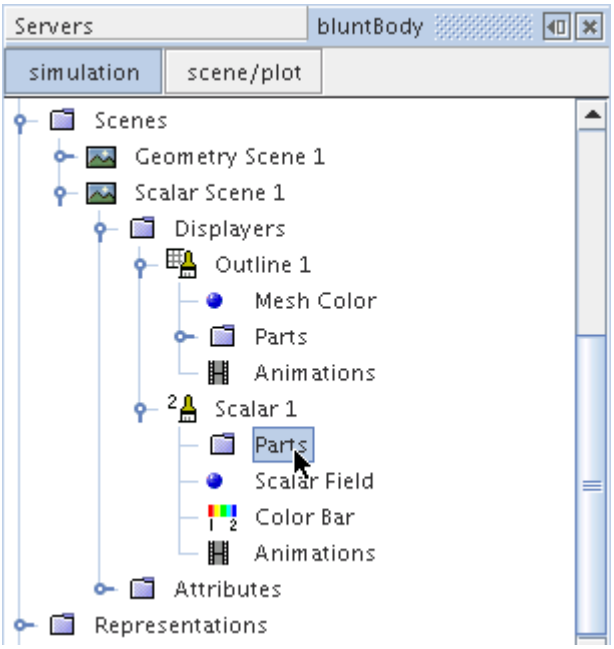


A new *Scalar Scene 1* display appears.

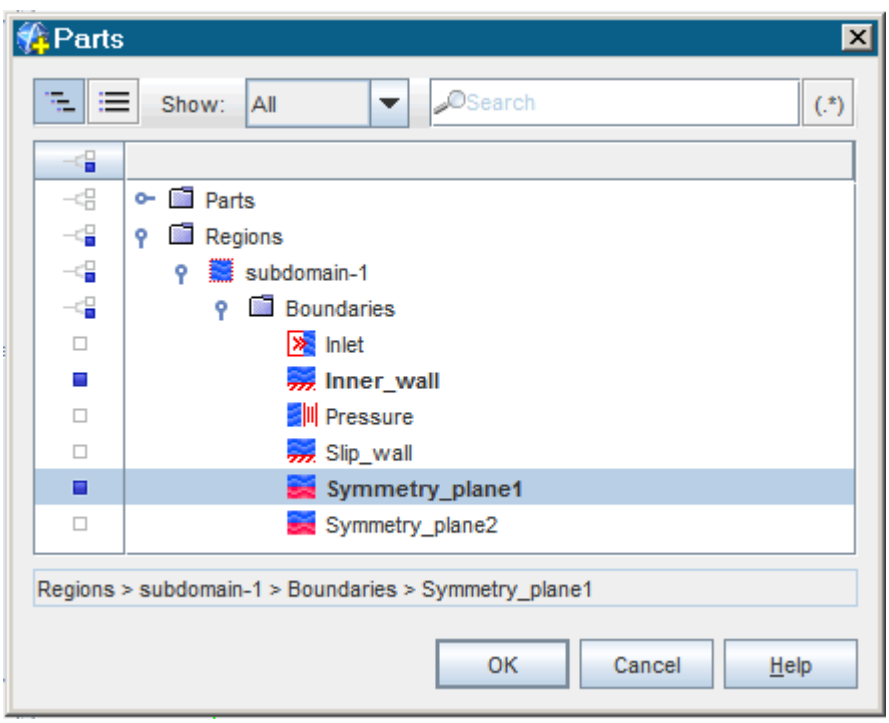


Hide all parts except the blunt body itself and the vertical symmetry plane.

2. Select the new **Scenes** > **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Parts** node.

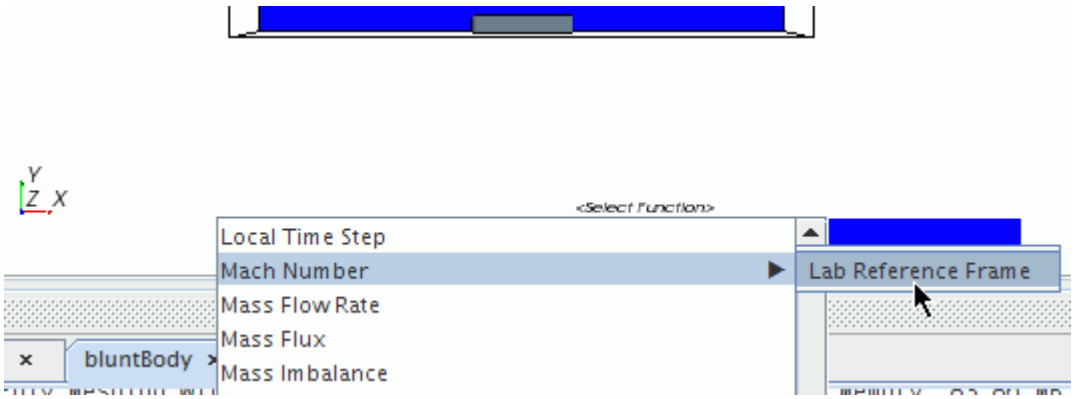


3. Click the (**Custom Editor**) in the right half of the **Parts** property.
4. In the *Parts* dialog, expand **Regions** > **subdomain-1** > **Boundaries**, select **Inner_wall** and **Symmetry_plane1**, and click **OK**.



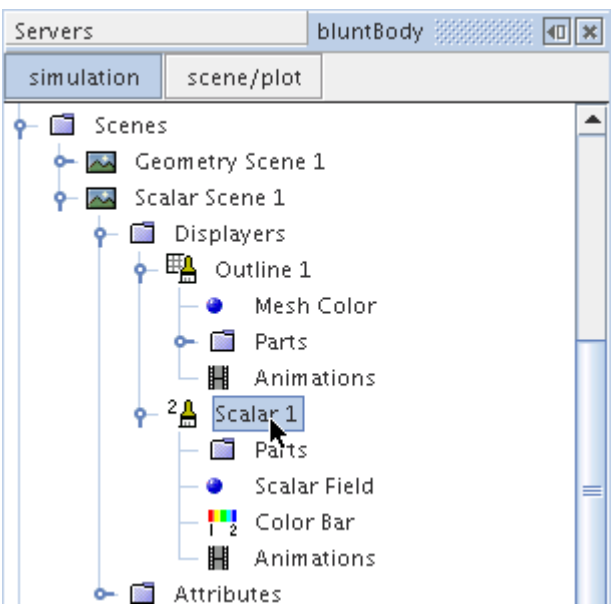
Define Mach number as the scalar to display:

5. Right-click the scalar bar (near the bottom of the *Graphics* window).
6. Select **Mach Number** > **Lab Reference Frame** from the pop-up menu.

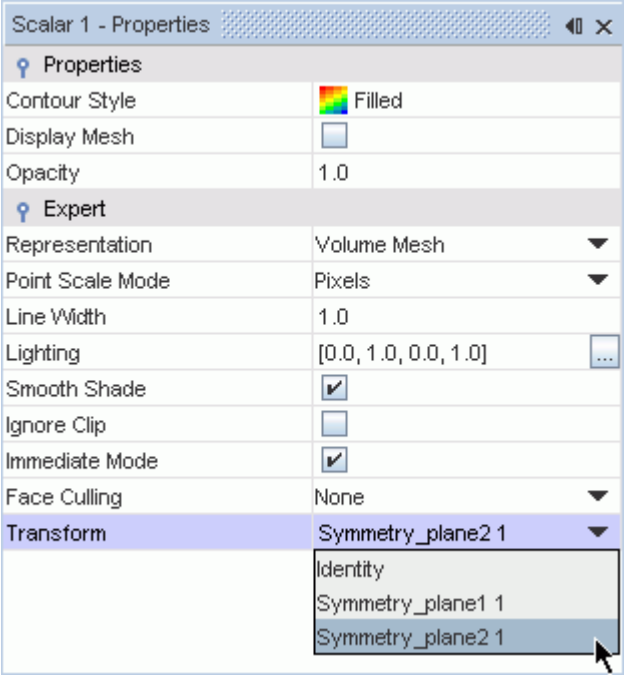


Since the geometry of this example is symmetric, the mesh was cut into quarters with two symmetry planes to reduce computing costs. However, the symmetric repeat transform lets you create the visual effect of the complete geometry by setting up the mirror image of the model in the Graphics window. In this case, you only do one repeat so that half the model is shown.

7. Select the **Scalar 1** node.

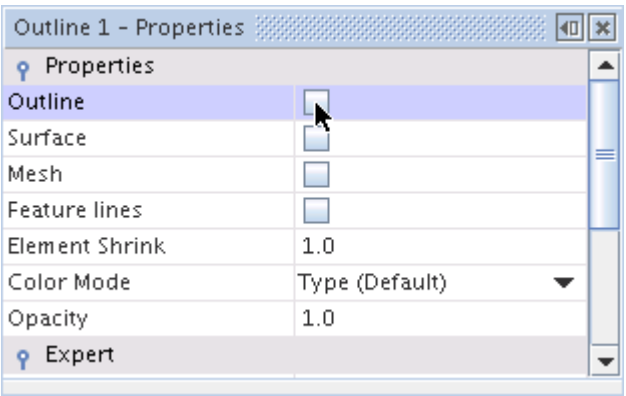


8. Set the **Transform** expert property to **Symmetry_plane2 1**.



To remove the outline:

9. Select the **Displayers** > **Outline 1** node.
10. Clear the checkbox of the **Outline** property.



The scalar scene appears as below.

11. Save the simulation.

Monitoring Simulation Progress

STAR-CCM+ can dynamically monitor a quantity of interest while the solution develops.

The process involves:

- Setting up a report that defines the quantity of interest and the region parts that are monitored

  See Setting Up a Report.
- Defining a monitor using that report, which controls the update frequency and normalization characteristics
- Setting up an X-Y plot using that monitor

  See Setting Up a Monitor and a Plot.

For this case, force is monitored on the body in the x-direction of the flow, which effectively is the total drag force. This process starts with the report definition.

Contents:

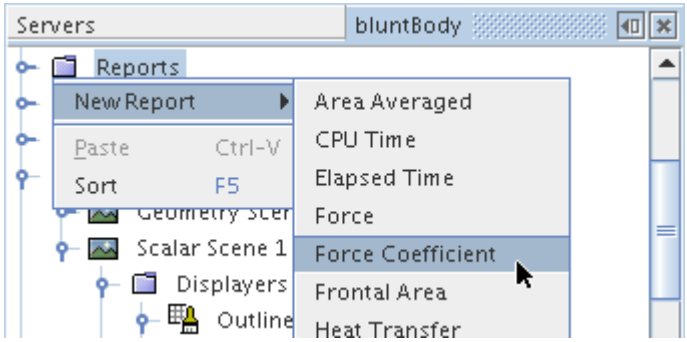
Setting Up a Report
Setting Up a Monitor and a Plot

Setting Up a Report

Create a report to collect the simulation results.

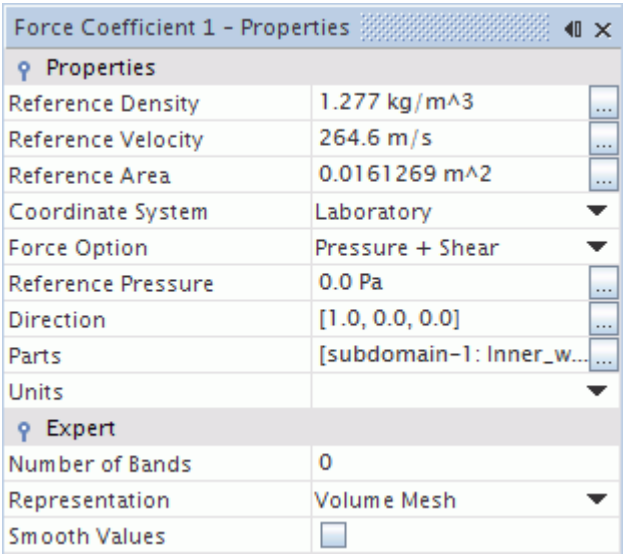
To set up a report:

1. Right-click the **Reports** node and select **New Report** > **Force Coefficient**.



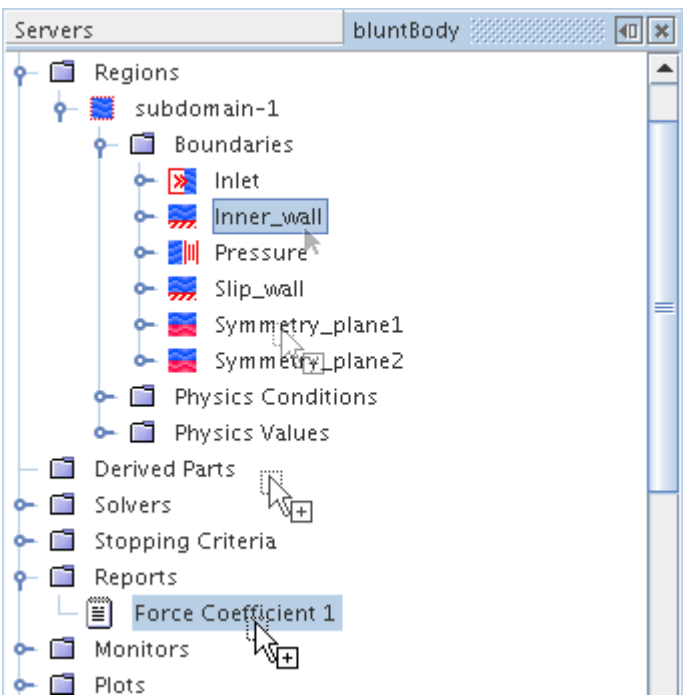
   This action creates a **Force Coefficient 1** node under the **Reports** node.
2. Select the **Force Coefficient 1** node and enter the settings for the report in the *Properties* window.
   a) Enter `1.277` for **Reference Density**, the density of the freestream air.
   b) Enter `264.6` for **Reference Velocity**, the velocity at the inlet.
   c) Enter a value of `0.0161269` for **Reference Area**, the projected area of the quarter of the blunt body that is used in the simulation.

d) Make sure that **Direction** is `[1.0,0.0,0.0]` (for drag).



Use the drag-and-drop method to select the monitored part, namely the **Inner_wall** boundary:

3. Open the **Regions** > **subdomain-1** > **Boundaries** node and then drag the **Inner_wall** node onto the **Force Coefficient 1** node.



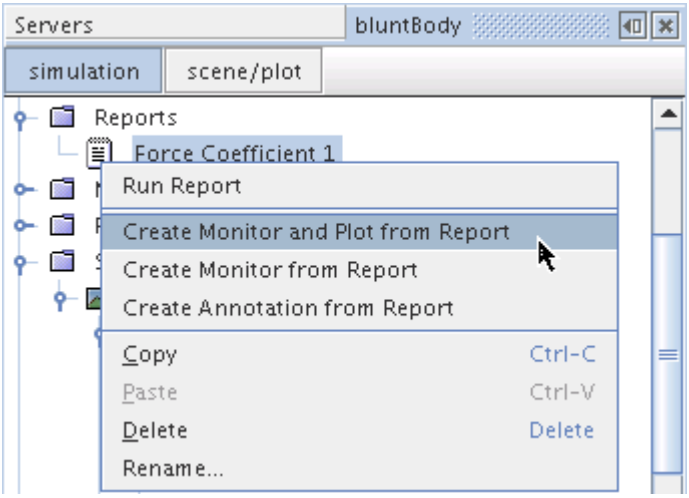
The **Inner_wall** entry is now listed in the **Parts** property of the **Force Coefficient 1** node.

The setup for the report is now complete, and a monitor and a plot can be made from that report.

Setting Up a Monitor and a Plot

Create a monitor and a plot to display the report data.

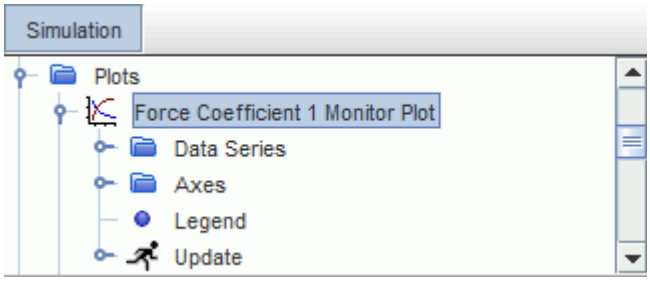
1. Right-click the **Force Coefficient 1** node and select **Create Monitor and Plot from Report**.



This action creates a **Force Coefficient 1 Monitor** node under the **Monitors** node.

The existing monitors in this branch of the simulation tree are the residual monitors from the solvers that the models use. With the new monitor node selected, the default settings for the **Force Coefficient 1** monitor appear.

In addition to the new monitor node, a **Force Coefficient 1 Monitor Plot** node is created in the **Plots** node.



2. Select the new **Force Coefficient 1 Monitor Plot** node.

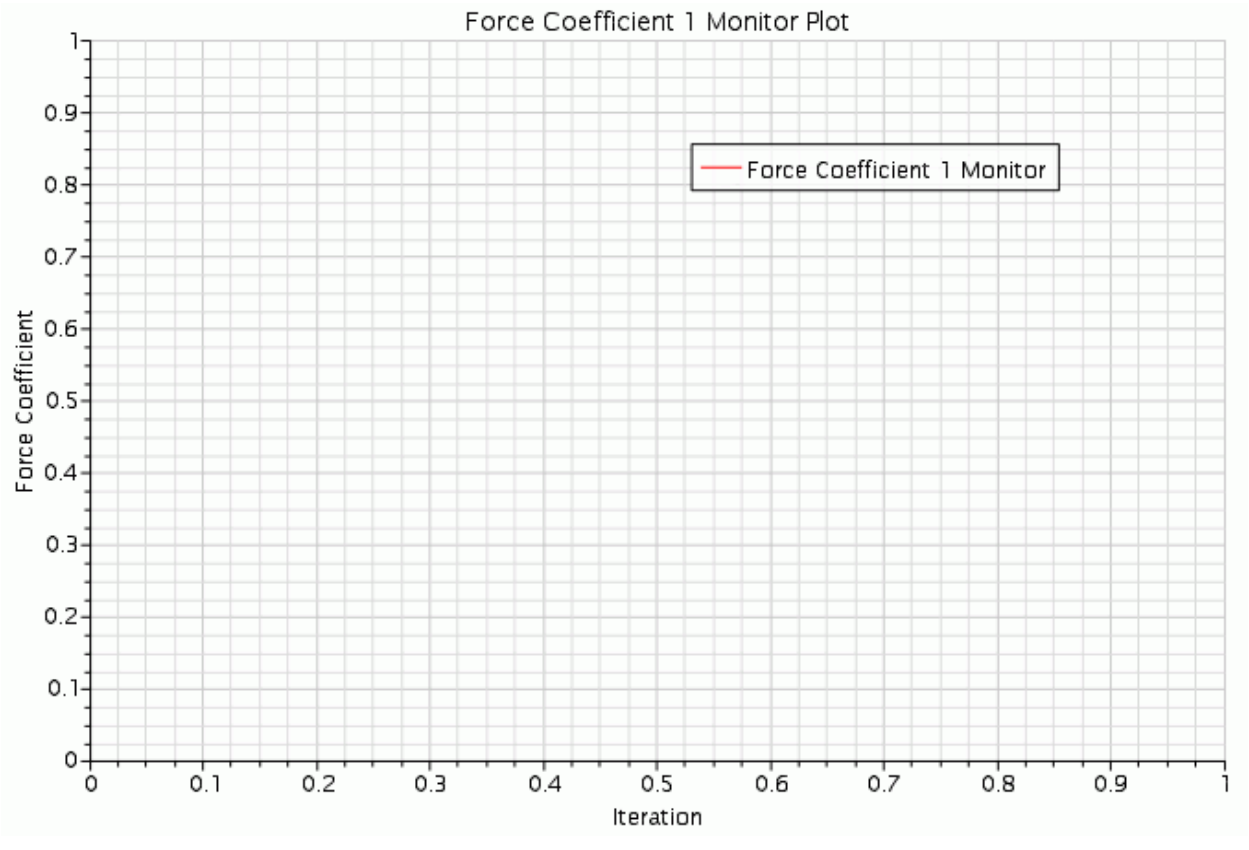
   The property settings for the plot appear.

Additional properties of the plot can be adjusted using the subnodes of the **Force Coefficient 1 Monitor Plot** node.

3. To view the plot display (if it is not already open), do one of the following:
   - Double-click the **Force Coefficient 1 Monitor Plot** node.

     or
   - Right-click the **Force Coefficient 1 Monitor Plot** node and select **Open** from the pop-up menu.

A *Force Coefficient 1 Monitor Plot* appears in the *Graphics* window.



This plot display is automatically updated with the drag force coefficient when the solver starts to run.

4. Save the simulation.

The pre-processing setup is completed. The simulation can now be run to convergence.

Running the Simulation

Preparation of the simulation is now complete, and the simulation can be run.

To run the simulation:

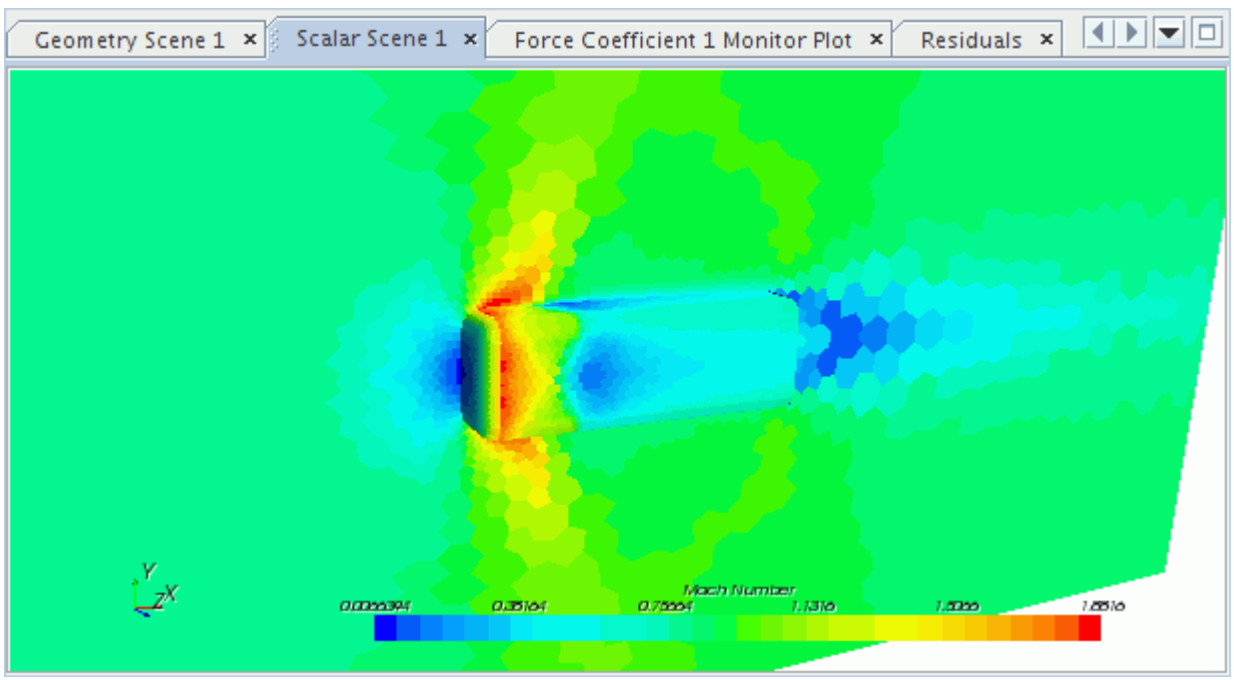
1. Do one of the following:

   - Click (**Run**) in the *Solution* toolbar.
   - Use the **Solution** > **Run** menu item.

   The *Residuals* display is created automatically and shows the progress of the solver. If necessary, click the *Residuals* tab to bring the *Residuals* plot into view. An example of a residual plot is shown in a separate part of the User Guide. This example looks different from your residuals, since the plot depends on the models selected.

The tabs at the top of the *Graphics* window allow you to select any one of the active displays for viewing.

2. To see the results, click the tab of the *Scalar Scene 1* display.
3. Rotate and zoom if desired for a better view.



4. During the run, it is possible to stop the process by clicking (**Stop**) in the toolbar.

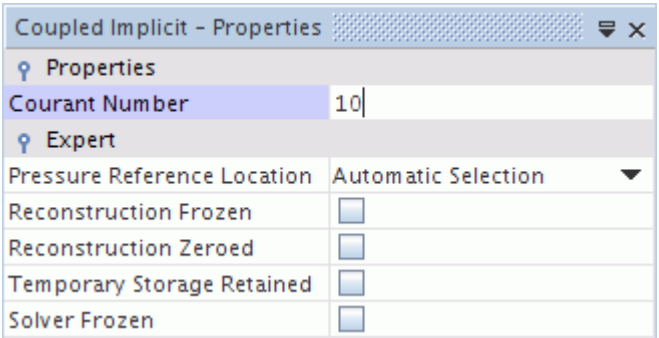
   If you do halt the simulation, you can resume it by clicking (**Run**). If left alone, the simulation continues until the stopping criterion of 300 iterations is satisfied.
5. After the run is finished, save the simulation.

Adjusting Solver Parameters and Continuing

The default Courant number of 5 is conservative for this type of problem on a mesh of reasonable quality, so it can be increased to accelerate convergence. When the solution has reached the stopping criterion of 300 iterations, adjust the solver parameters.

To adjust the solver parameters:

1. Select the **Solvers** > **Coupled Implicit** node and set **Courant Number** to `10.0`.



2. Select the **Stopping Criteria** > **Maximum Steps** node and set the **Maximum Steps** to `1500`.

   This setting instructs STAR-CCM+ to iterate 1200 steps in addition to the 300 already done.

The maximum number of 1500 steps is a reasonable setting, in this case, to get to convergence. Normally, you do not know ahead of time what a suitable iteration count is. You can either set a large iteration and watch the residuals or a monitor plot, or add a stopping criterion from a monitor.

3. To continue running the simulation, click (**Run**).

Contents:

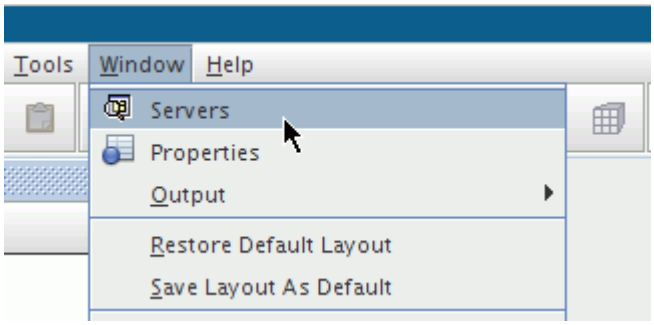
Disconnecting and Reconnecting
Completing the Run

Disconnecting and Reconnecting

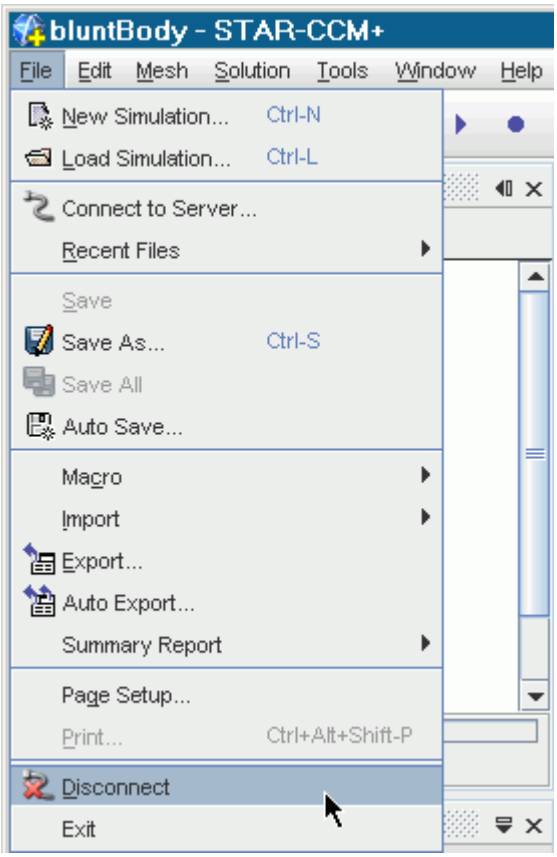
The STAR-CCM+ client-server design permits you to run a simulation, disconnect the client from the running server, and then reconnect later to check on results. You can also reconnect from a different machine, even a machine of another architecture.

To disconnect the client from the server:

1. Open the *Servers* window (if it is not already open) by selecting **Window** > **Servers**.



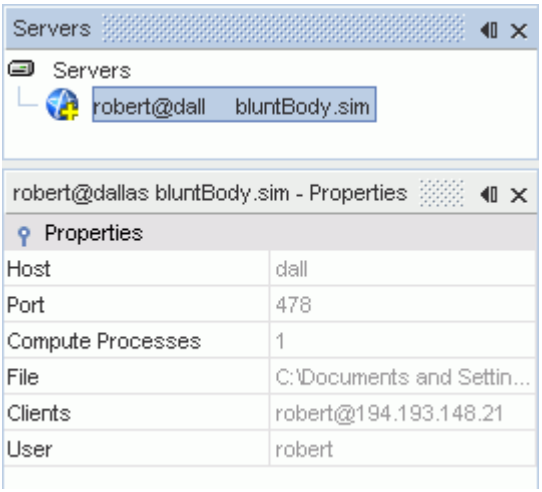
2. To see your simulation, look in the *Servers* window of the *Explorer* pane.
3. Reselect the simulation window in the *Explorer* pane, and select the **File** > **Disconnect** menu item.



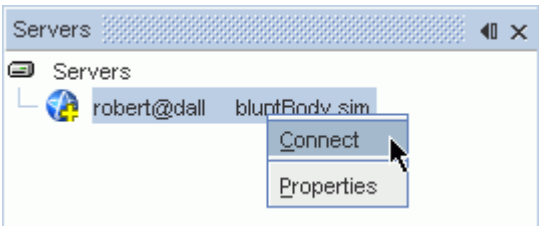
4. In the *OK to disconnect* dialog, click **OK**.

   The client disconnects from the server, but both are still running, although the simulation window no longer appears in the client. If desired, the client can be shut down using **File** > **Exit**.

   The *Servers* window of the *Explorer* pane lists all server processes (simulations) that are running locally.
5. Selecting the node in the tree displays the server process properties.



6. Continue by right-clicking the server process node in the **Servers** tree and selecting **Connect** from the pop-up menu.



When you reconnect, none of the previously created scenes and plots are displayed in the *Graphics* window.

7. To display a scene of interest, do one of the following:
   - Double-click its node in the simulation object tree.

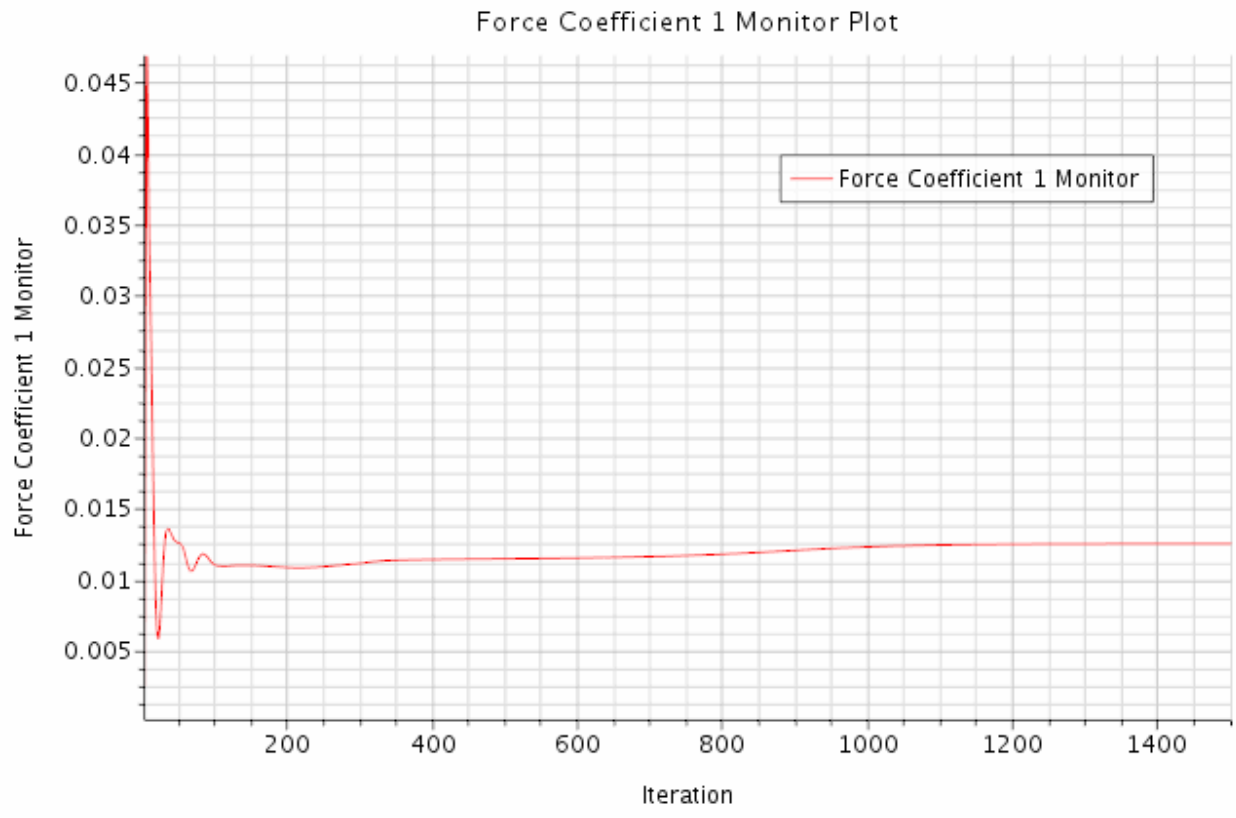
     or
   - Right-click the node and select **Open**.

Completing the Run

At the end of the calculation, the *Residuals* display show most of the residuals flattening out, which is a good indicator of convergence.

To confirm that the solution has converged:

1. Select the *Force Coefficient 1 Monitor Plot* display to verify that the drag coefficient has also flattened out.



2. Once the run is finished, save the simulation.

Visualizing the Results

View the results of the simulation using the scalar and vector scenes that you prepared earlier.

Once the solution is finished, you can examine the results in scalar scenes and vector scenes.

Contents:

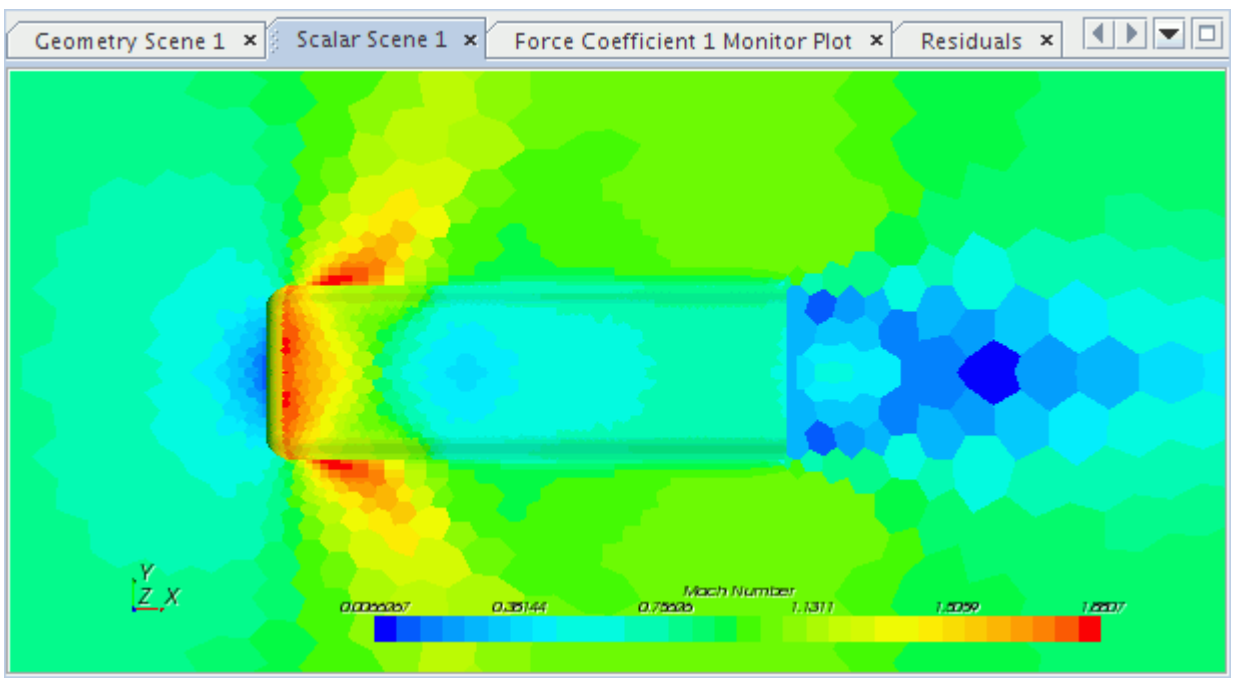
Examining Scalars
Examining Vectors

Examining Scalars

View the results of the simulation in a scalar scene.

To see the Mach number results for the finished solution:

1. Make the *Scalar Scene 1* display active.

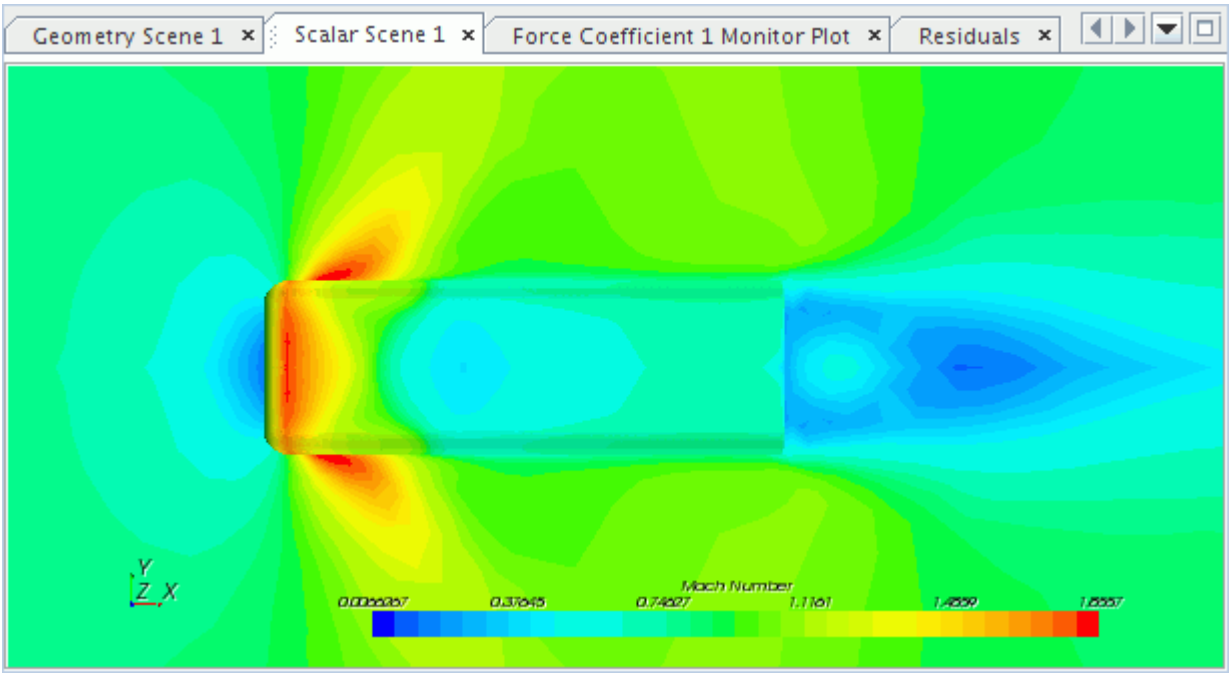


By default, the **Automatic** contour style is shown.

To display smooth contours:

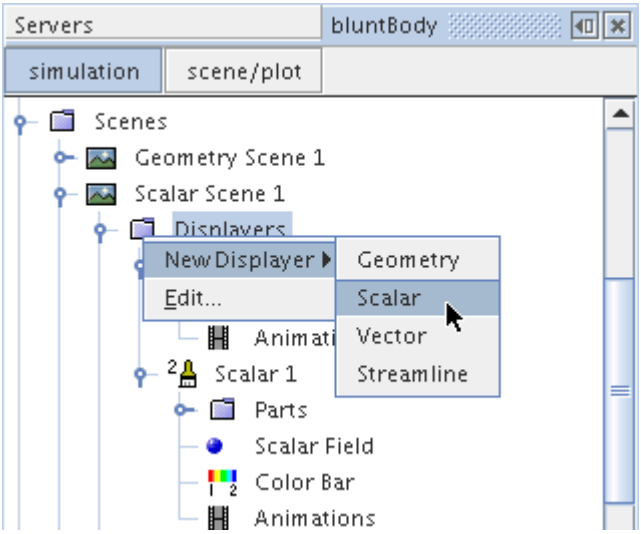
2. Select the **Scalar 1** node and set the **Contour Style** to (**Smooth Filled**).

   The contours of the scalar display now appear smooth.



Display two scalar values at the same time: one on the symmetry plane, the other on the inner wall:

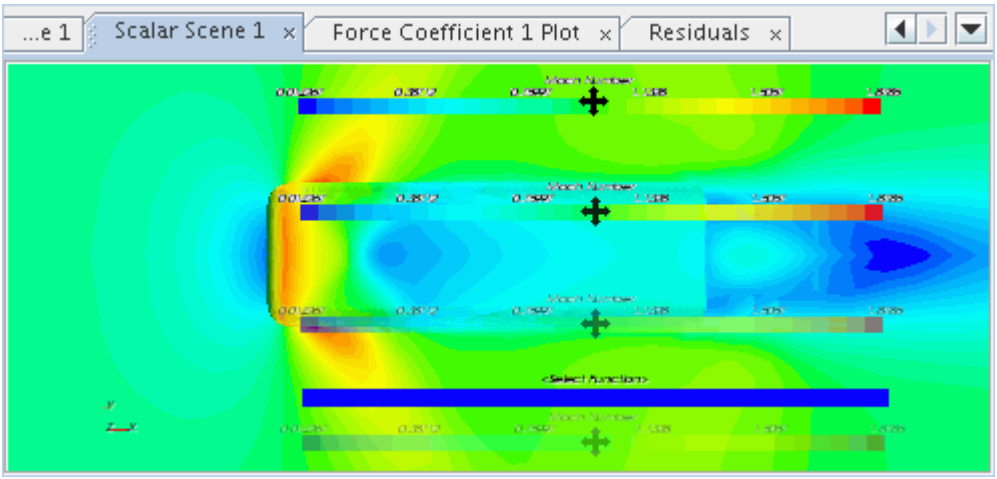
3. Add another scalar displayer by right-clicking the **Displayers** node and selecting **New Displayer** > **Scalar**.



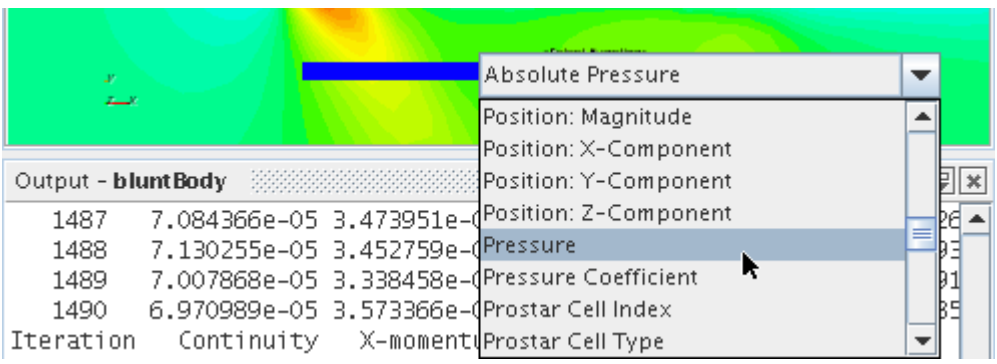
The creation of the second displayer has added a second scalar bar in the display.

To view both scalar bars:

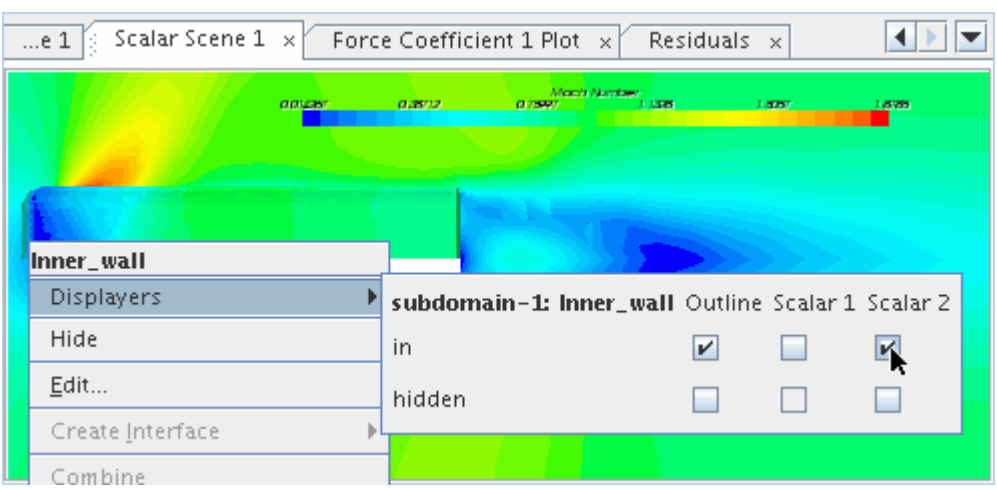
4. Point the mouse in the interior of a scalar bar to drag either one of them to a different location.



5. In the display, right-click the scalar bar (blue) of the new displayer and select **Pressure**.

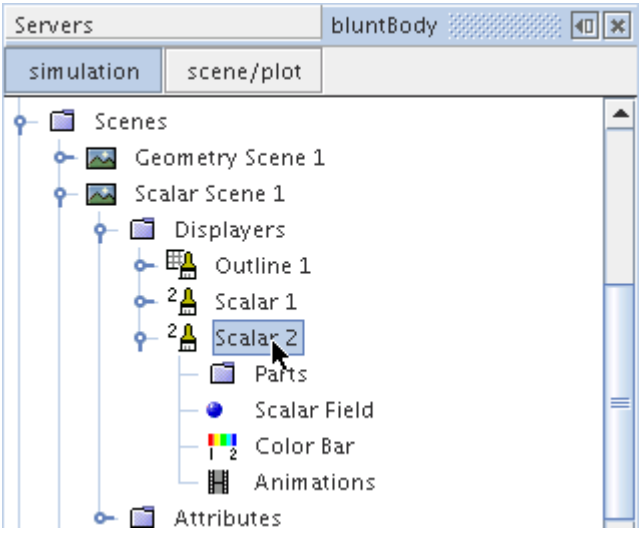


6. Right-click the inner wall in the display, select **Displayers**, and open its submenu.
7. Among the checkboxes to the right of **in**, clear the checkbox under **Scalar 1** and tick the one under **Scalar 2**.



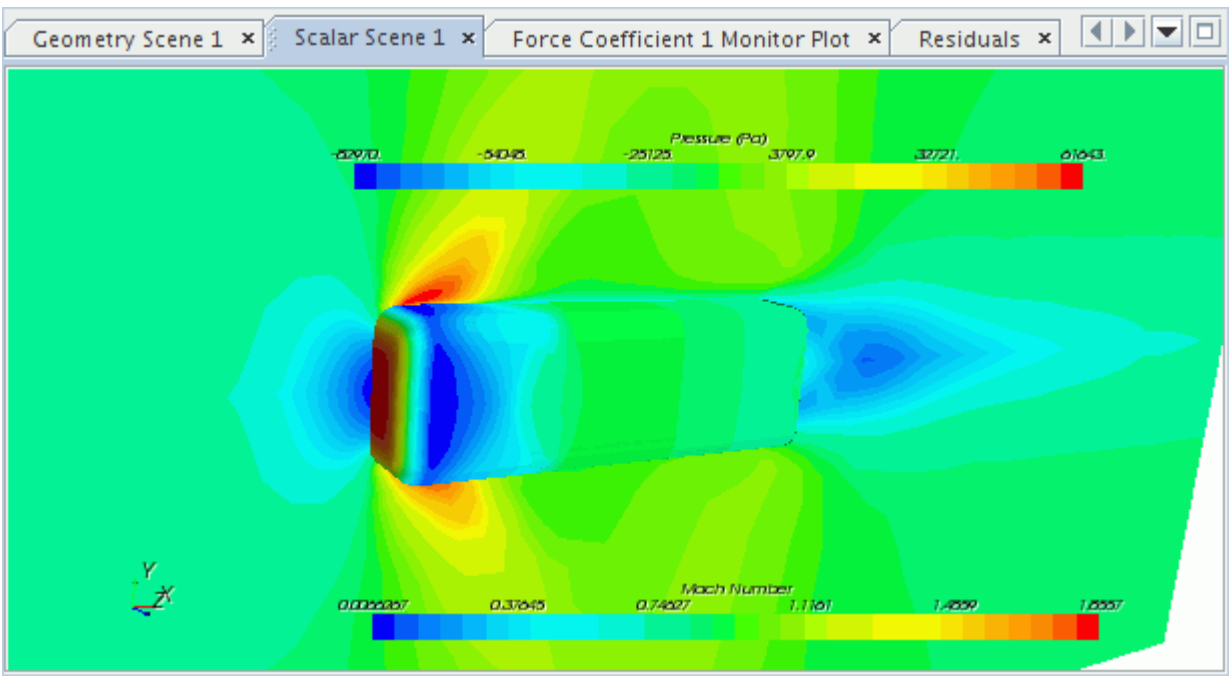
This single step transfers the part from one displayer to another.

8. Select the **Scenes** > **Scalar Scene 1** > **Displayers** > **Scalar 2** node.



9. Set the **Contour Style** to (**Smooth Filled**).
10. Set the **Transform** expert property to `Symmetry_plane2 1`.

    The blunt body shows pressure values while the symmetry plane shows Mach number.

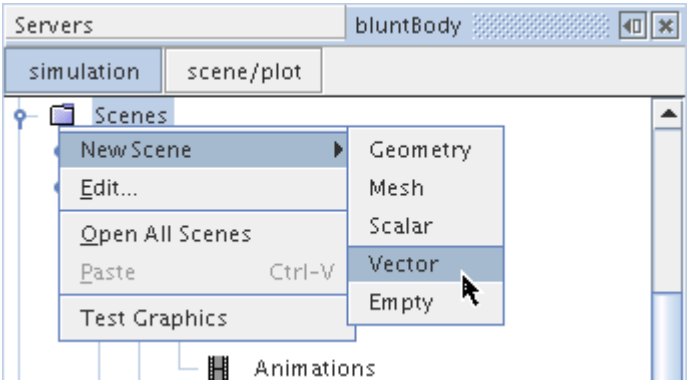


11. Save the simulation.

Examining Vectors

This section demonstrates how to examine velocity vectors in the model. In this example, you set up vectors in a reflected symmetry plane around a solid-colored blunt body.

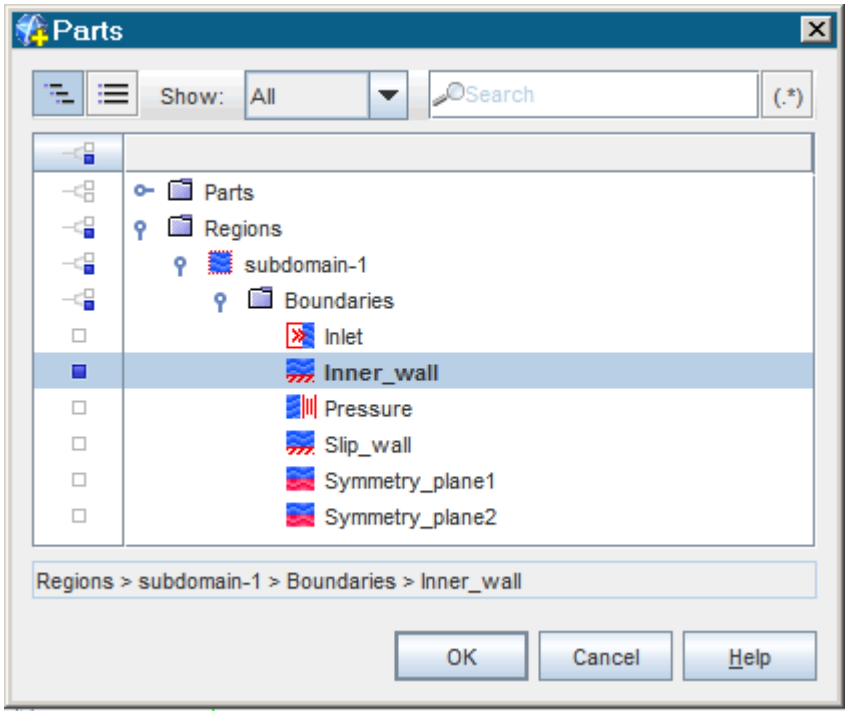
1. First, create a scene by right-clicking the **Scenes** node.
2. Select **New Scene** > **Vector**.



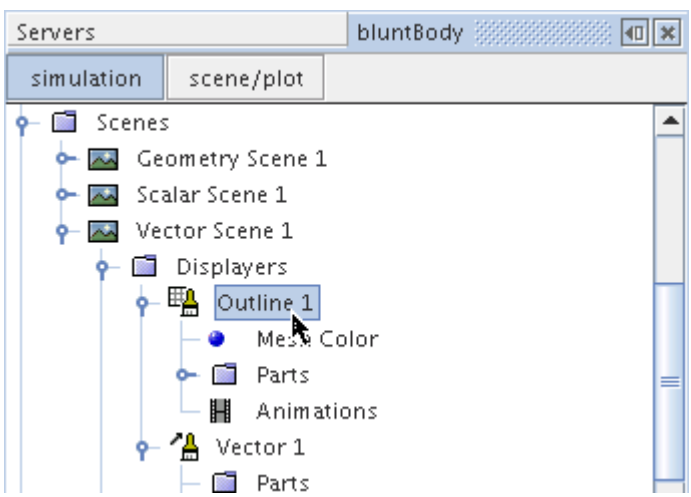
A new *Vector Scene 1* display appears.

You do not need the outlines that the **Outline** part displayer provides by default, so reuse that displayer to show the blunt body as a solid object.

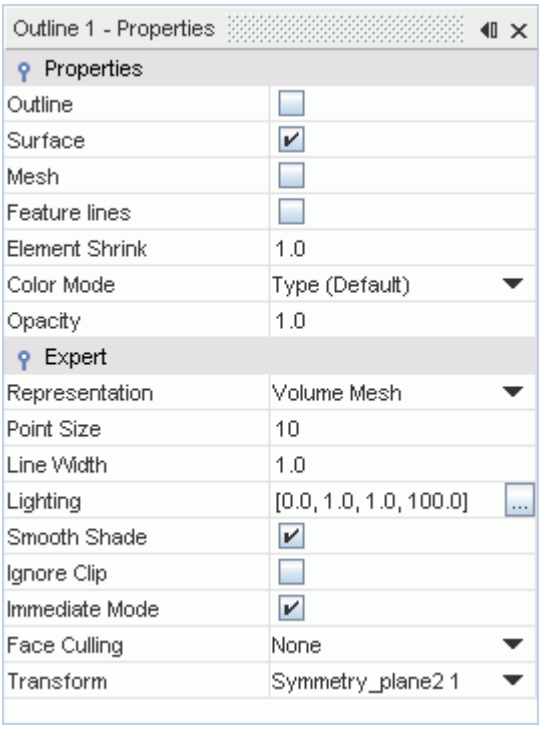
3. Select the **Vector Scene 1** > **Displayers** > **Outline 1** > **Parts** node and click ... (**Custom Editor**).
4. In the *Parts* dialog, click the column selection button twice.
5. Expand the **Regions** > **subdomain-1** > **Boundaries** node.
6. Select the **Inner_wall** node.
7. Click **OK**.



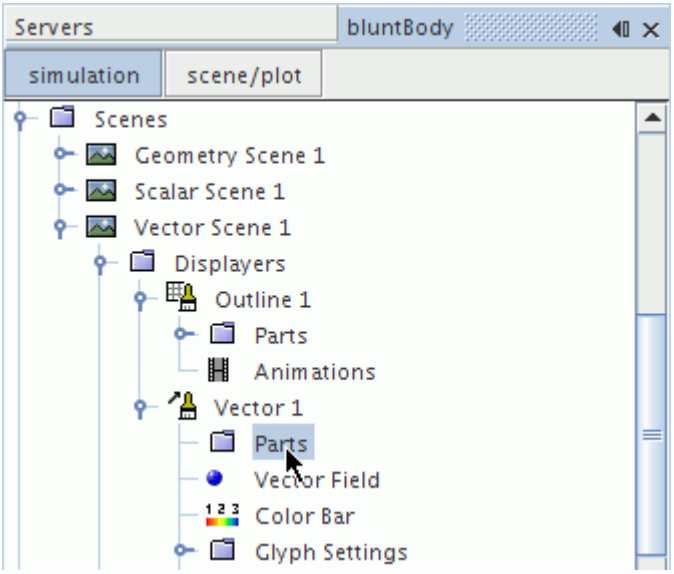
8. Select the **Outline 1** node in the **Displayers** node.



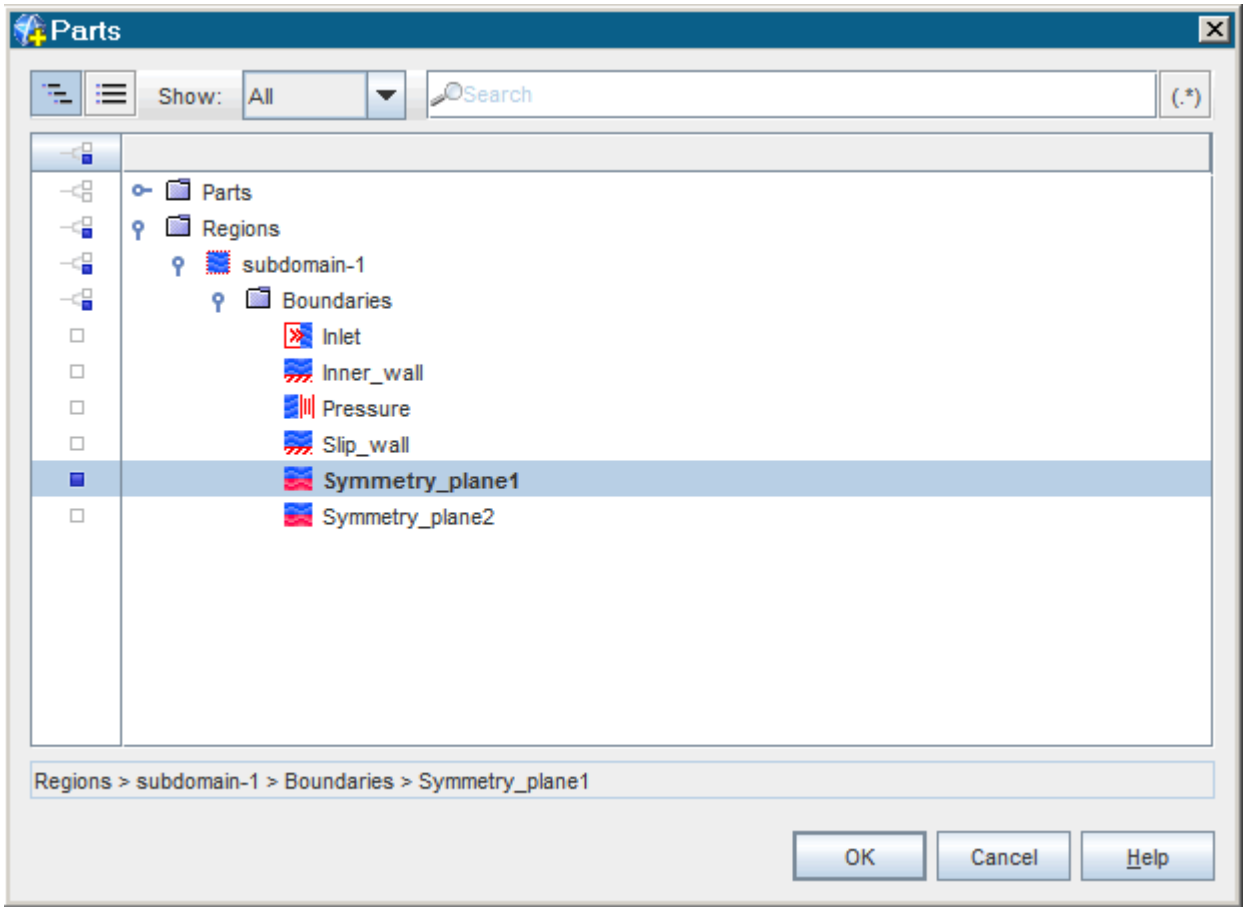
9. Make the following changes to the **Outline** displayer properties:
   a) Deactivate the **Outline** property.
   b) Activate the **Surface** property.
   c) Set the **Transform** expert property to `Symmetry_plane2 1`.



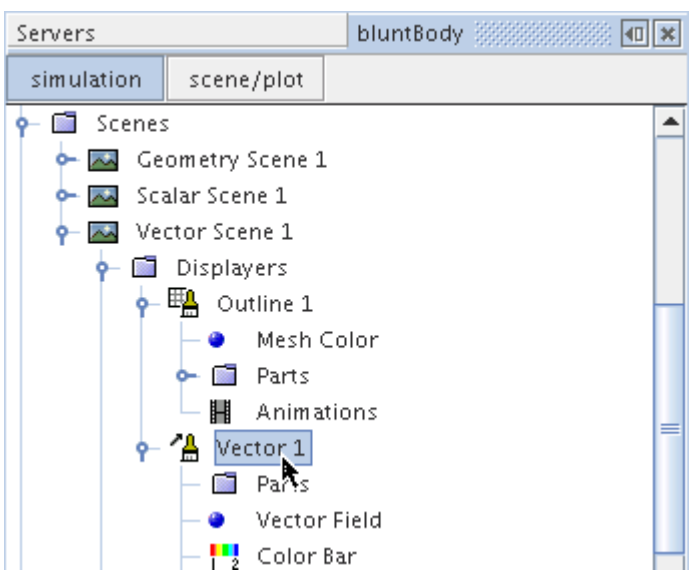
10. Select the **Vector 1** > **Parts** node and click ... (**Custom Editor**) for the **Parts** value.



11. In the *Parts* dialog, expand the **Regions** and **subdomain-1** nodes. Select the **Symmetry_plane1** node and click **OK**.

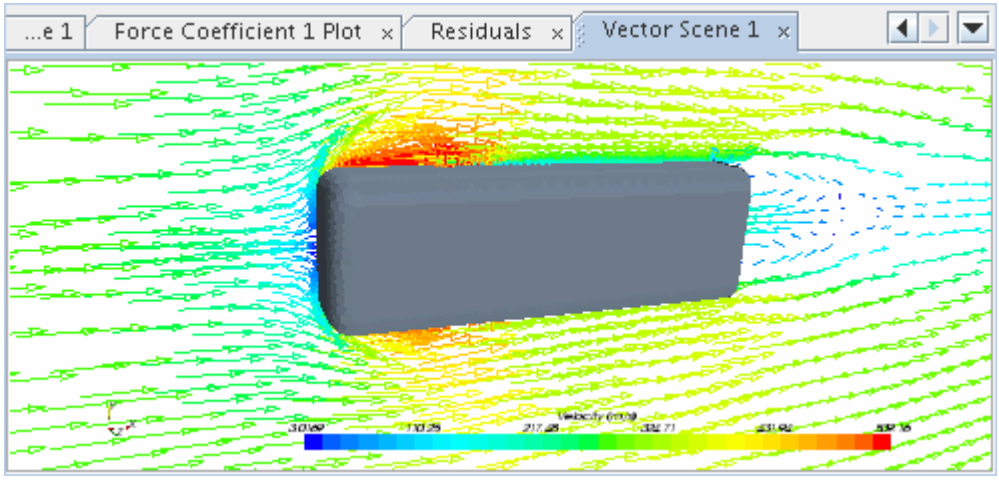


12. Select the **Vector 1** node.



13. Set the **Transform** expert property to `Symmetry_plane2 1`.

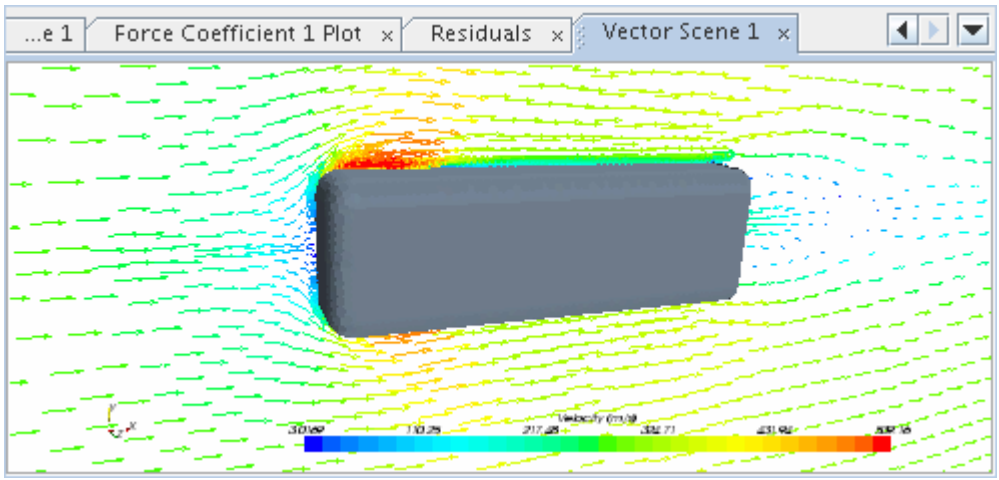
    These settings show a solid blunt body with velocity vectors surrounding it.



It is possible to make the arrows shorter to see the vectors more clearly.

14. Select the **Vector 1** > **Glyph** > **Relative Length** node and change **Glyph Length (%)** to `2.0`.

    The arrows now appear shorter.



15. Save the simulation.

Plotting Data on a Slice

Create an x-y plot to visualize the pressure coefficient (on the y-axis) along a slice of the blunt body.

Creating the x-y plot requires the following steps:

1. Create a derived part for the slice.
2. Create a plot using the new derived part.

Contents:

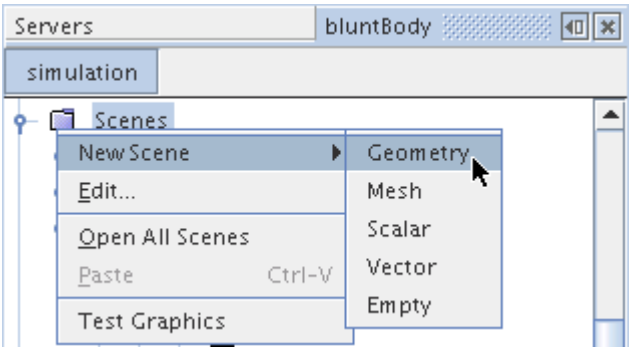
Creating a Derived Part
Plotting Simulation Data

Creating a Derived Part

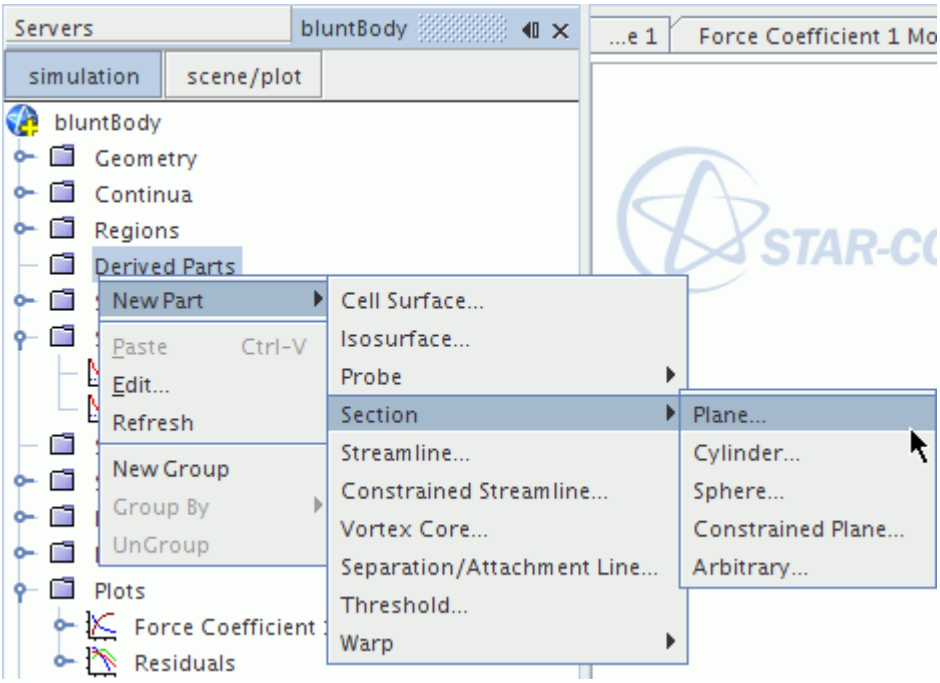
Create a plane section that cuts the blunt body. In STAR-CCM+ terminology, this plane section is called a derived part.

To create the derived part:

1. Right-click the **Scenes** node and select **New Scene** > **Geometry**.



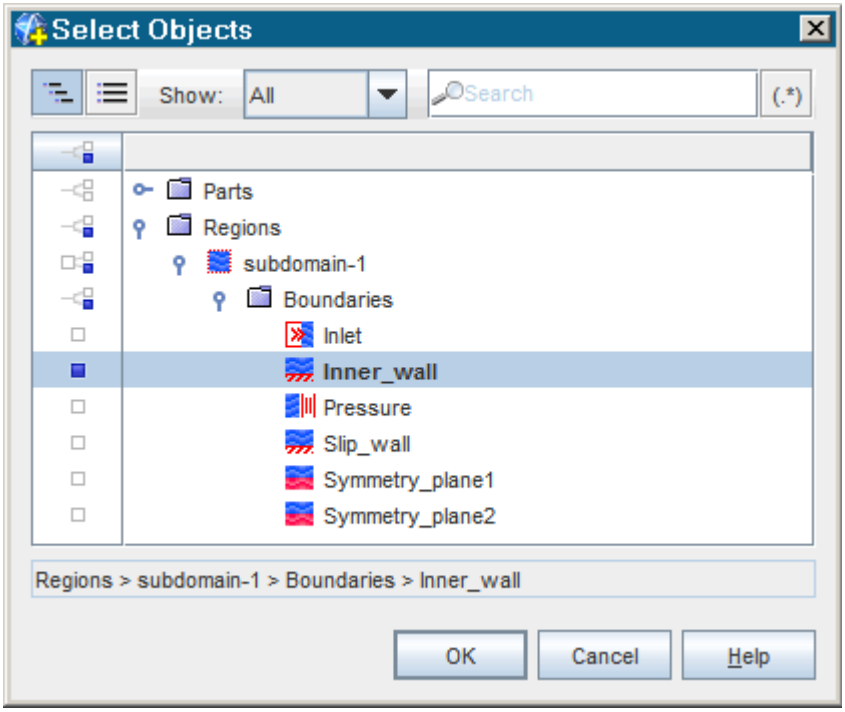
2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane...**.



A new interactive in-place dialog is activated to allow definition of the derived part.

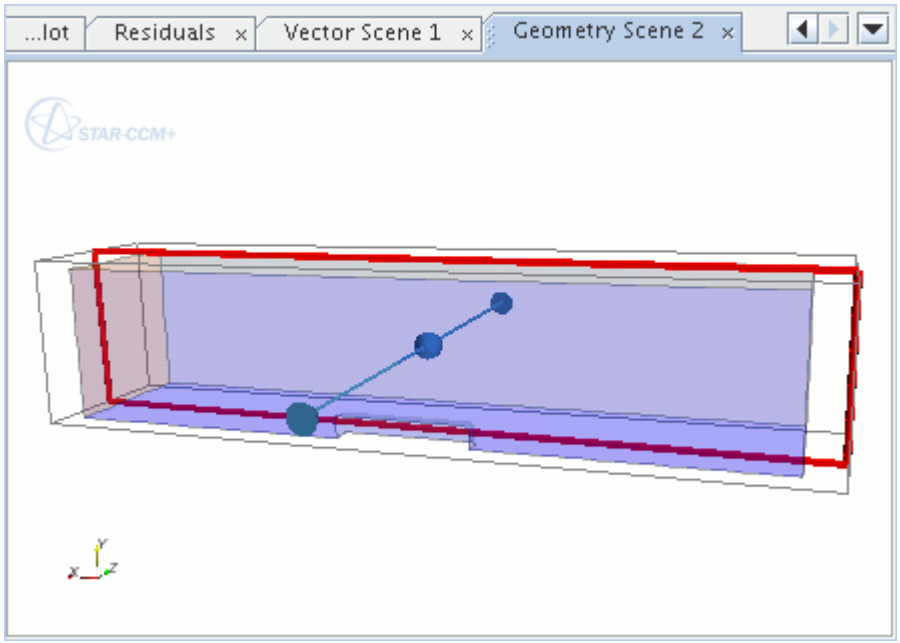
To define the source part for this derived part:

3. In the *Input Parts* group box, click **[subdomain-1]**.
4. In the dialog that appears, click the column selection button twice.
5. Expand the **Regions** > **subdomain-1** > **Boundaries** node.
6. Select the **Inner_wall** boundary and click **OK**.



The plane tool in the scene display allows interactive definition of the plane that cuts the blunt body.

7. In the *Plane Parameters* group box, set the **Normal** to `0.0, 0.0, 1.0`.
8. Rotate and pan the image as needed so that the inner wall is in front of the plane tool rectangle as shown.

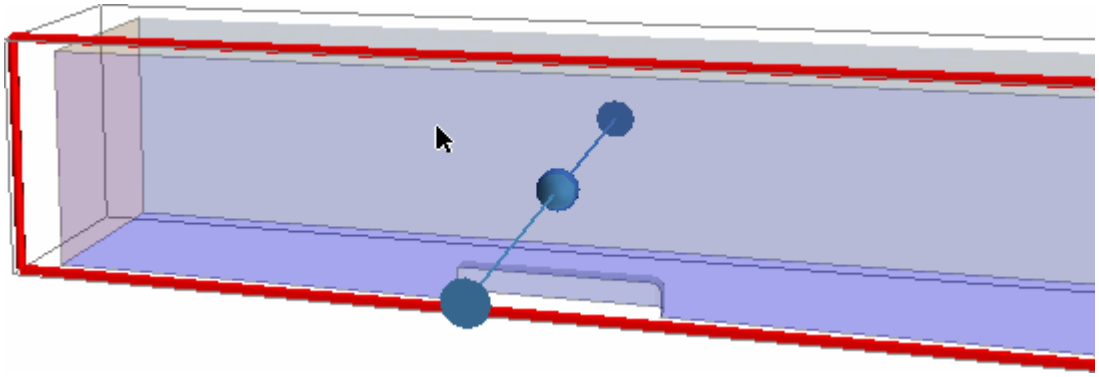


9. In the dialog, leave the *Display* box at its default setting.

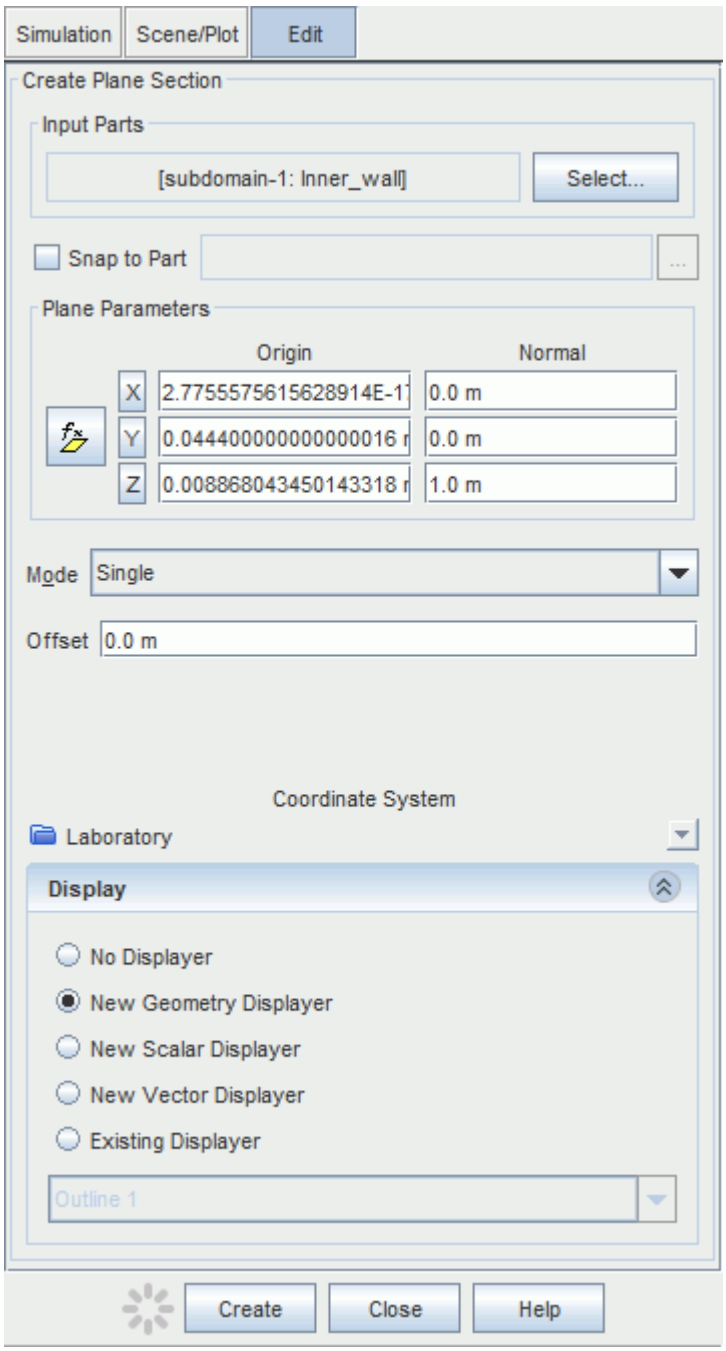
   The “barbell” represents the normal of the cutting plane, which can be changed by dragging one of the balls to another location. The plane can also be dragged to change the origin of the cutting plane.

A bounding box slightly larger than the regions in the scene is displayed to limit the movement of the plane.

10. Move the mouse over the plane tool, then click-and-drag the rectangle so that it cuts the blunt body close to the symmetry plane.

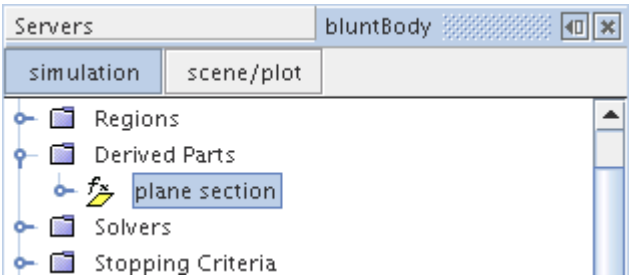


The properties of the *edit* dialog are similar to those in the following screenshot:



11. Click **Create** and then **Close**.

    In the *bluntBody* window, a node, **Plane Section**, has been added within the **Derived Parts**.



It is now possible to plot data on the new surface.

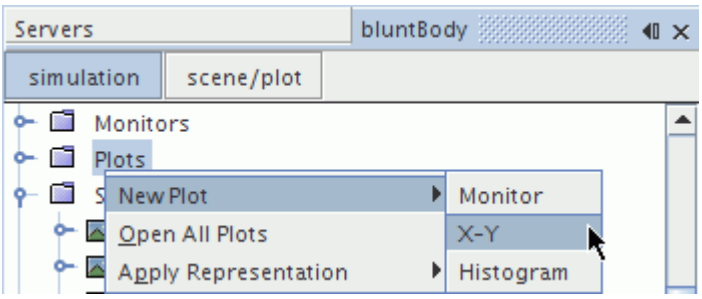
12. Save the simulation.

Plotting Simulation Data

Set up a plot of pressure coefficients for the new derived part.

To create the plot:

1. In the *bluntBody* window, right-click the **Plots** node and select **New Plot** > **XY Plot**.



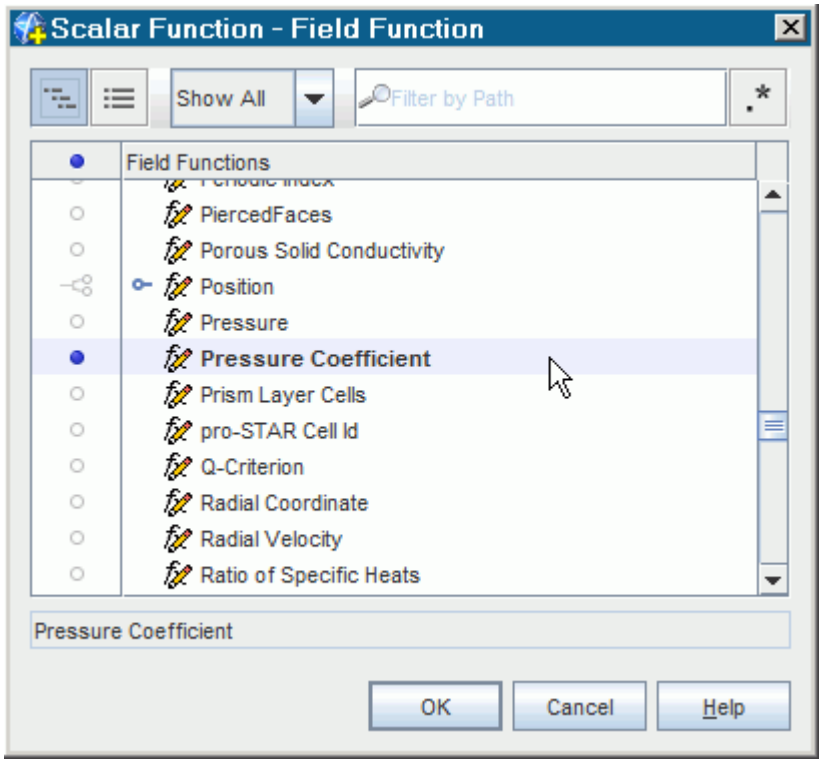
An empty plot node is added to the tree and opens in the *Graphics* window.

To select the part on which to plot:

2. Select the **Plots** > **XY Plot 1** node and click ... (**Custom Editor**) for the **Parts** property.
3. In the *XY Plot 1 - Parts* dialog, make sure that **Derived Parts** > **Plane Section** is ticked.
4. Click **OK**.

To set the variable to plot on the y-axis:

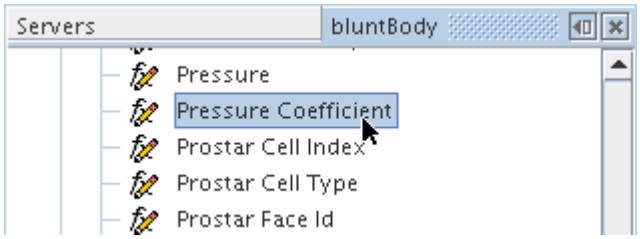
5. Select the **XY Plot 1** > **Y Types** > **Y Type 1** node and make sure that the **Type** property is set to **Scalar**.
6. Select the **Y Type 1** > **Scalar Function** node and click ... (**Custom Editor**) for the **Field Function** property.
7. In the *Scalar Function - Field Function* dialog, select **Pressure Coefficient**.



8. Click **OK**.
9. Select the **XY Plot 1** > **Axes** > **Bottom Axis** node and ensure that the **Lock Minimum** and **Lock Maximum** properties are deactivated.

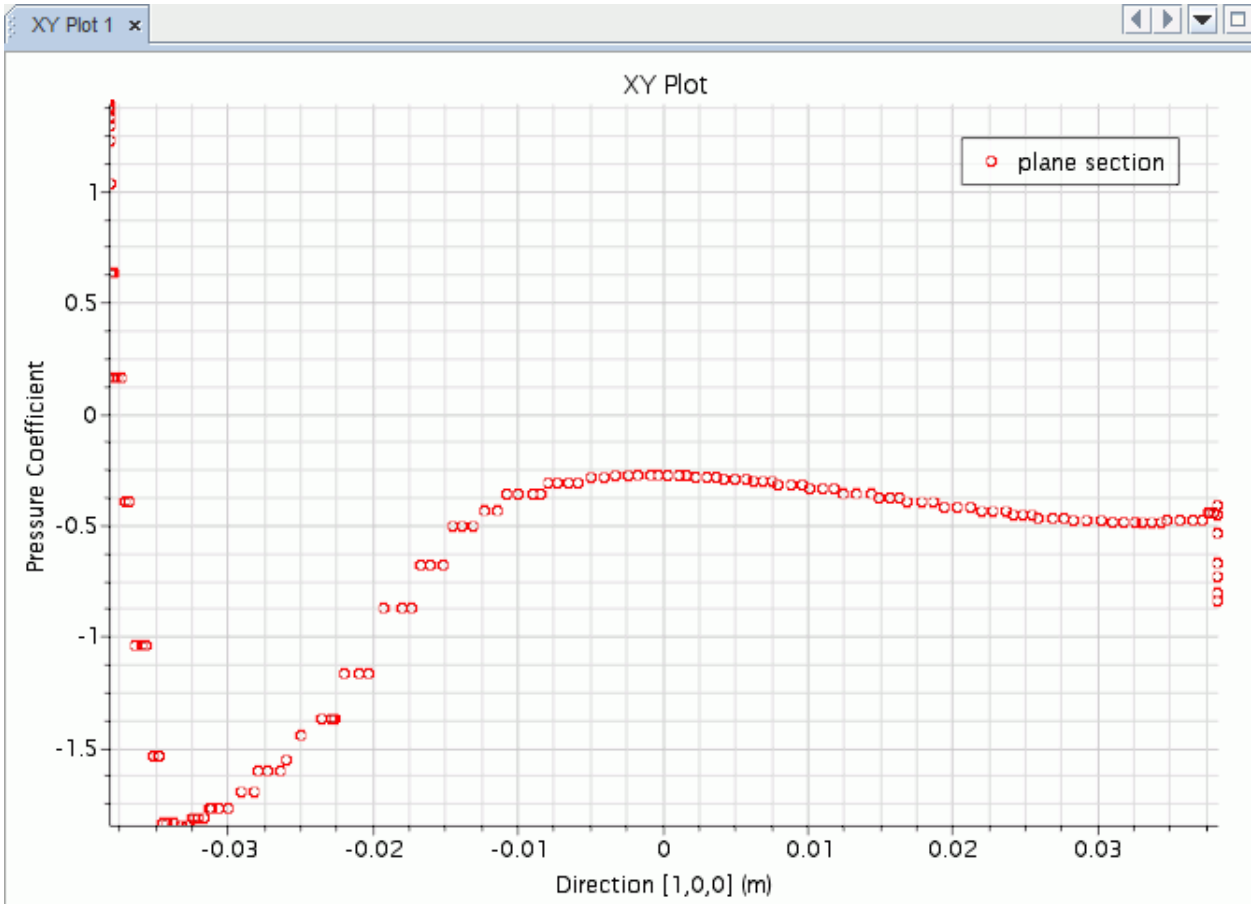
Match the reference density, reference pressure, and reference velocity of the **Pressure Coefficient** field function in the **Force Coefficient 1** report. Copy and paste these properties from the report to the field function:

10. Select the **Reports** > **Force Coefficient 1** node.
11. In the *Properties* window, select the value of the **Reference Density** property, and press **<Ctrl+C>**.
12. Select the **Tools** > **Field Functions** > **Pressure Coefficient** node.



13. In the *Properties* window, select the value of the **Reference Density** property, press `<Ctrl+V>`, and then `Enter`.
14. Repeat the above steps for the **Reference Velocity**.
15. Leave the **Reference Pressure** property at `0.0 Pa`.

    The plot appears as shown below.



16. Save the simulation.

Adding Streamlines

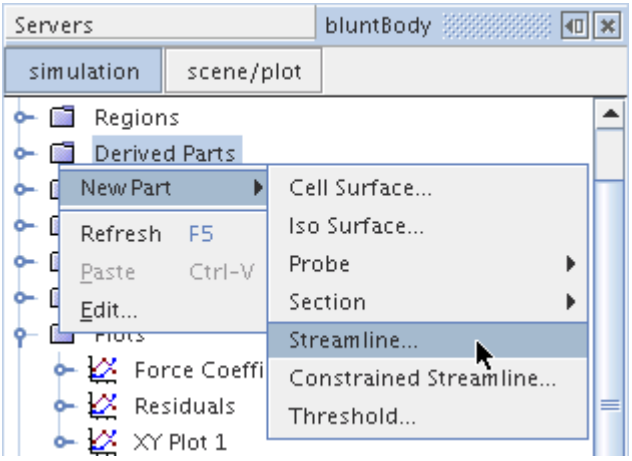
Set up a streamline derived part so that it shows the recirculating flow behind the blunt body. Modify the part to show streamlines over the blunt body.

To visualize streamlines in STAR-CCM+, do the following:

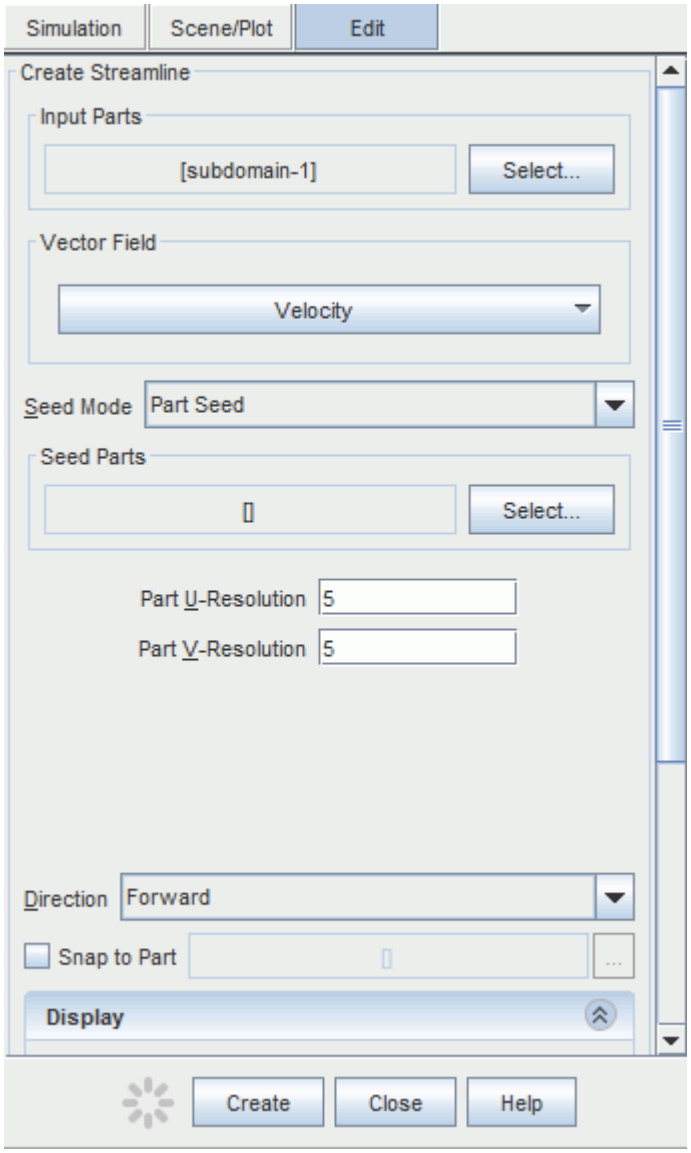
- Create a scene.
- Create a streamline derived part.
- Add a streamline displayer, with suitable properties, to the new scene based on the streamline derived part.

To create a scene and add streamlines:

1. Right-click the **Scenes** node and select **New Scene** > **Geometry**.
2. Right-click the **Derived Parts** node.
3. Select **New Part** > **Streamline**.

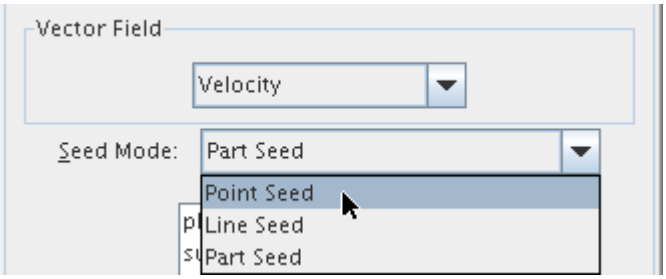


As with the plane section earlier, an interactive in-place dialog appears.



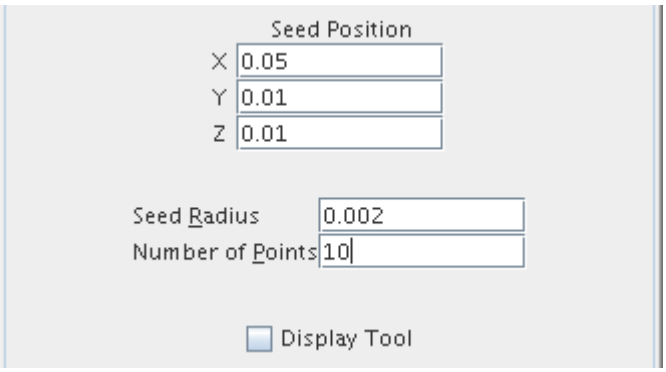
The entire fluid region is used as the input part for the streamline part.

4. In the *Input Parts* group box, click the **Select** button.
5. In the *Select Objects* dialog, make sure that `subdomain-1` is the only region selected.
6. Click **OK**.
7. In the *Vector Field* group box, make sure that **Velocity** is selected.
8. For the seed mode, select **Point Seed**.



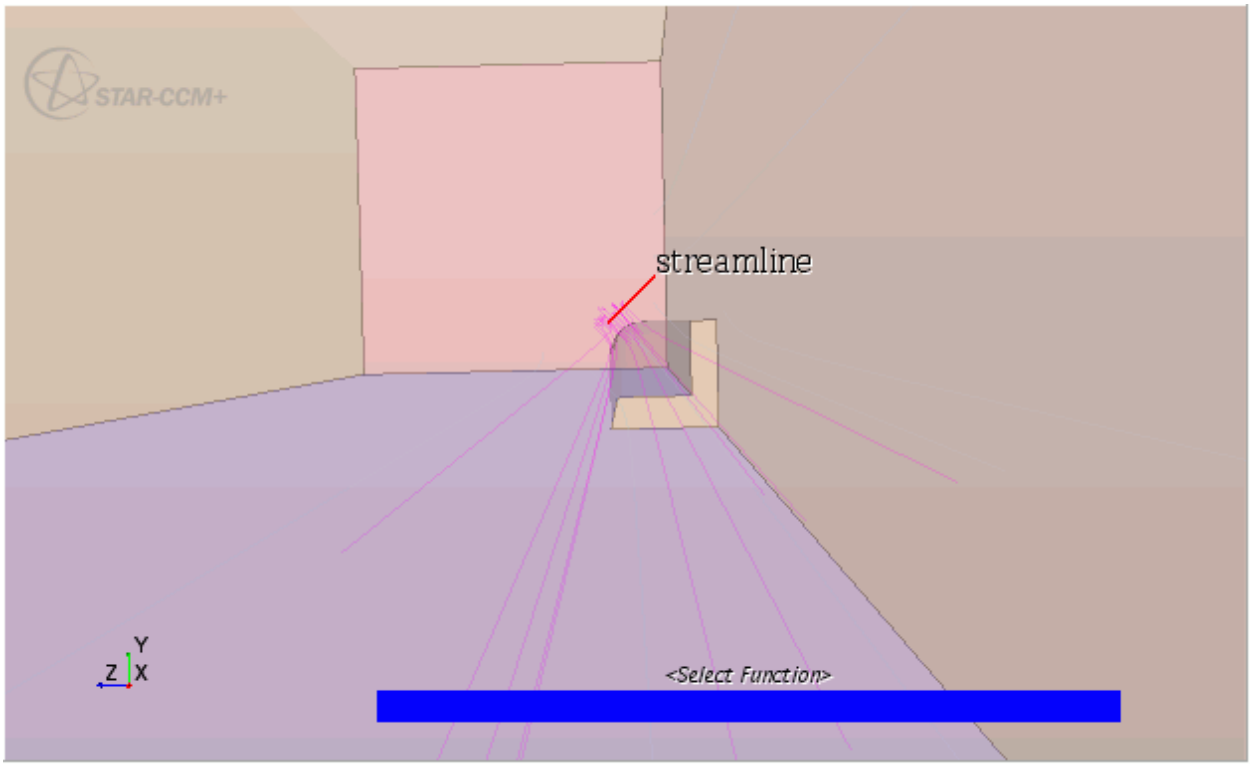
To position the seed point just after the back of the blunt body (in the direction of flow):

9. Enter the following coordinates under **Seed Position**:
   `0.05` for X, `0.01` for Y, and `0.01` for Z
10. Enter `0.002` for **Seed Radius** and `10` for **Number of Points**.



11. Click **Create**.

    After a few seconds, the streamlines are shown in the display.

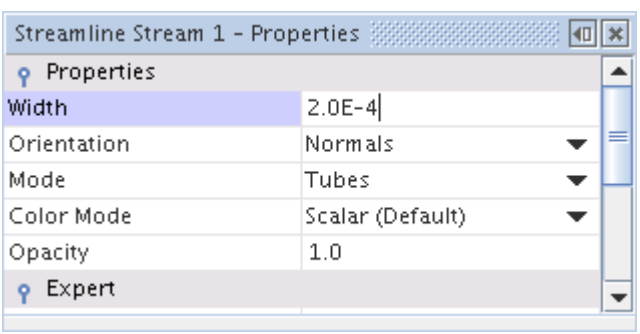


In the *bluntBody* window, a **Streamline** node appears in the **Derived Parts** node.

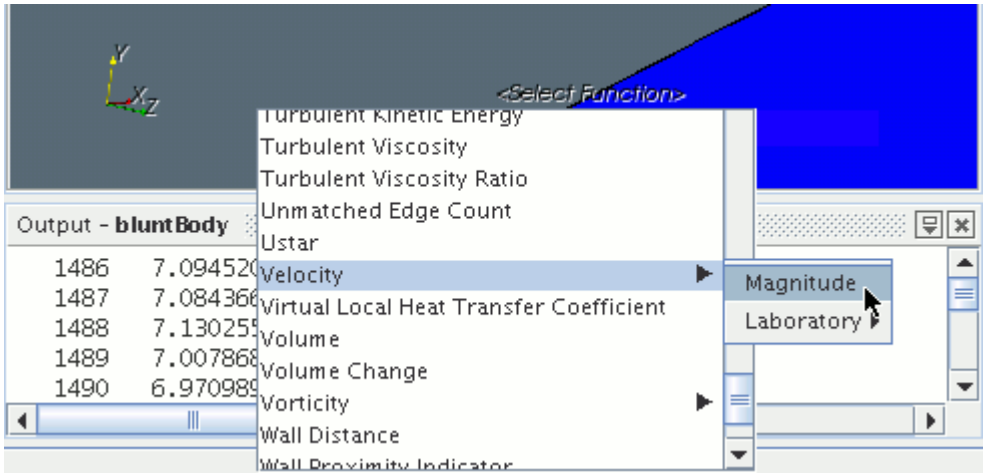
12. Click **Close** in the interactive in-place dialog.

    A new streamline displayer has been added to the scene.

13. Select the **Displayers** > **Streamline Stream 1** node to customize how the streamlines appear:
    a) Set the **Mode** to **Tubes** so that the streamlines are displayed as tubes instead of lines.
    b) Make sure that **Orientation** is set to **Normals**.
    c) Specify **Width** at `2.0E-4`, the width of the tube in meters.

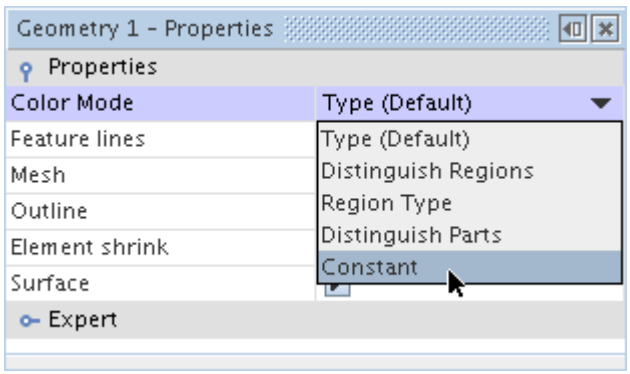


14. In the display, right-click the color bar and select **Velocity: Magnitude**.



Improve the contrast between boundaries and streamlines:

15. Select the **Geometry 1** node and set the **Color Mode** to **Constant**.



You now fine-tune the streamline derived part.

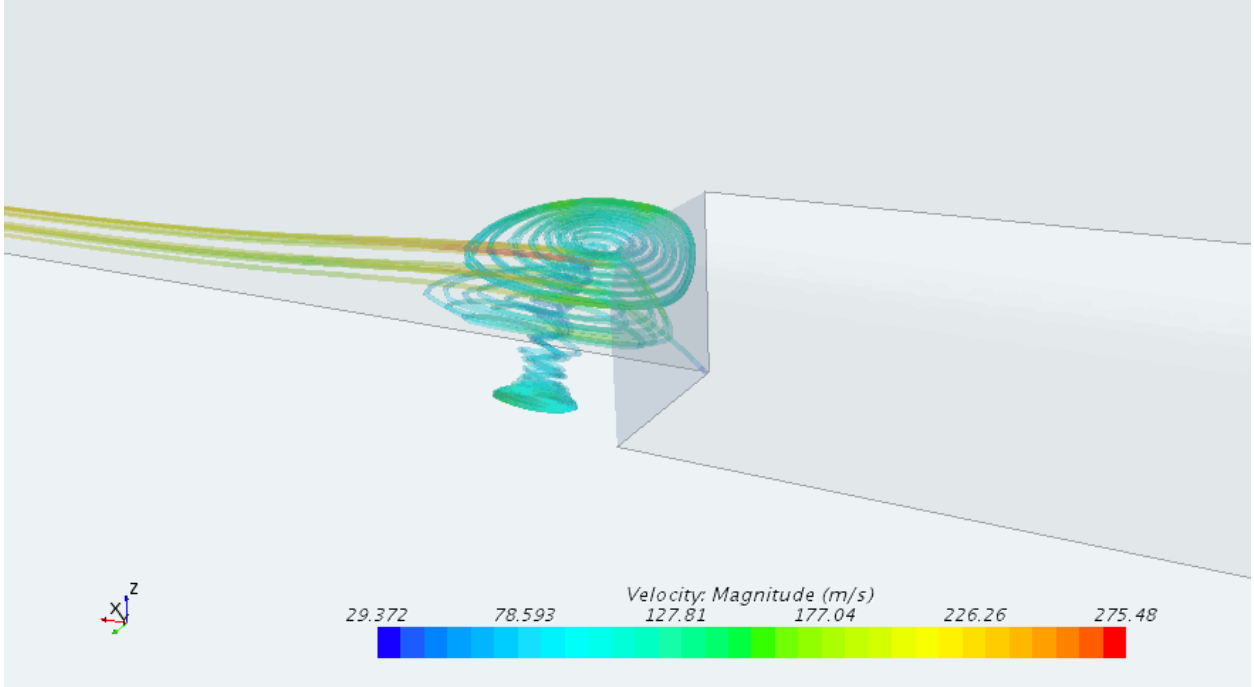
16. Select the **Derived Parts** > **Streamline** node and make sure that the **Rotation Scale** expert property is `1.0`.
17. Select the **Streamline** > **2nd Order Integrator** node and set the **Integration Direction** to **Both** so that the streamlines are generated both upstream and downstream.

In the Expert properties:

18. Set **Initial Integration Step** to `0.1` to provide more resolution to the streamlines.
19. Set **Maximum Propagation** to `5.0`.
20. Set **Max Steps** to `1000`.

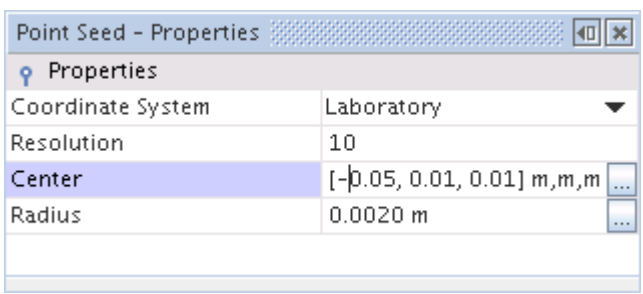
    This option, together with the maximum propagation, provide a stopping criterion to make sure that the streamlines are not calculated endlessly.

    The modified streamlines appear as below.

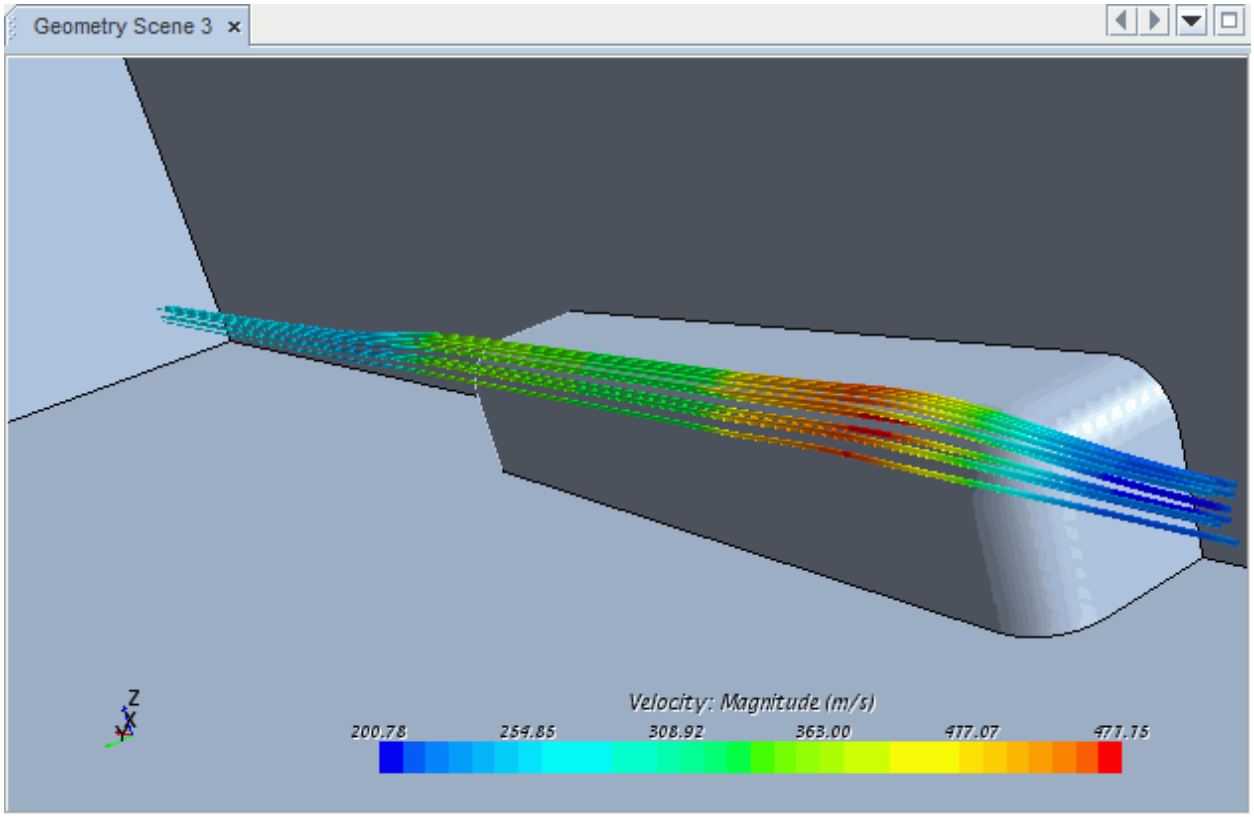


The next step involves moving the seed point of the streamline derived part to the front of the blunt body.

21. Select the **2nd Order Integrator** node and change **Integration Direction** to **Forward**.
22. Select the **Streamline** > **Point Seed** node and set the x-value of the **Center** to `-0.05`, so that the value of the property is `-0.05, 0.01, 0.01`.



The modified streamlines appear in the display.



23. Save the simulation.

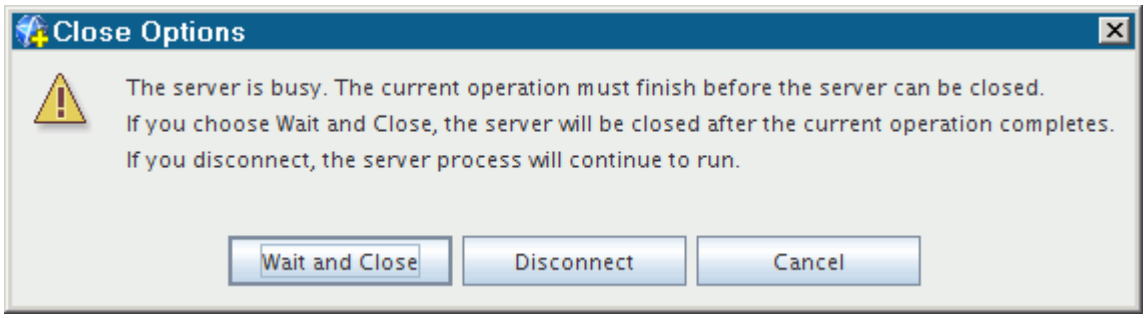
Closing and Reopening the Simulation

Close the simulation and reopen it again.

To close a simulation while it is running:

1. Close the *bluntBody* window.

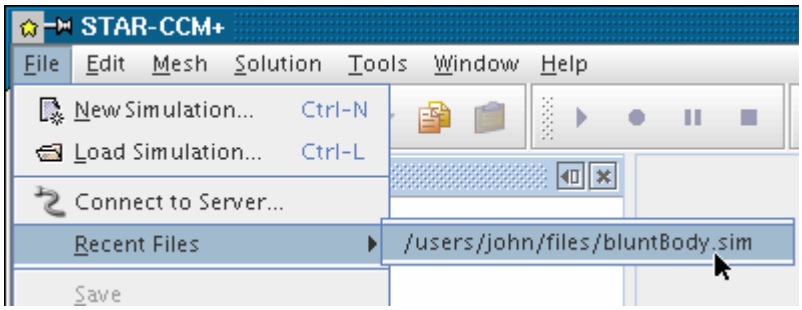
   The following dialog prompts you to choose whether to wait for the operation to finish before closing the window, or close the window and leave the operation to complete in the background.



2. Click **Disconnect**.

It is easy to reopen the file and restore the displays.

3. Select **File** > **Recent Files** > **/bluntBody.sim**.



Summary

This tutorial has introduced the following STAR-CCM+ features:

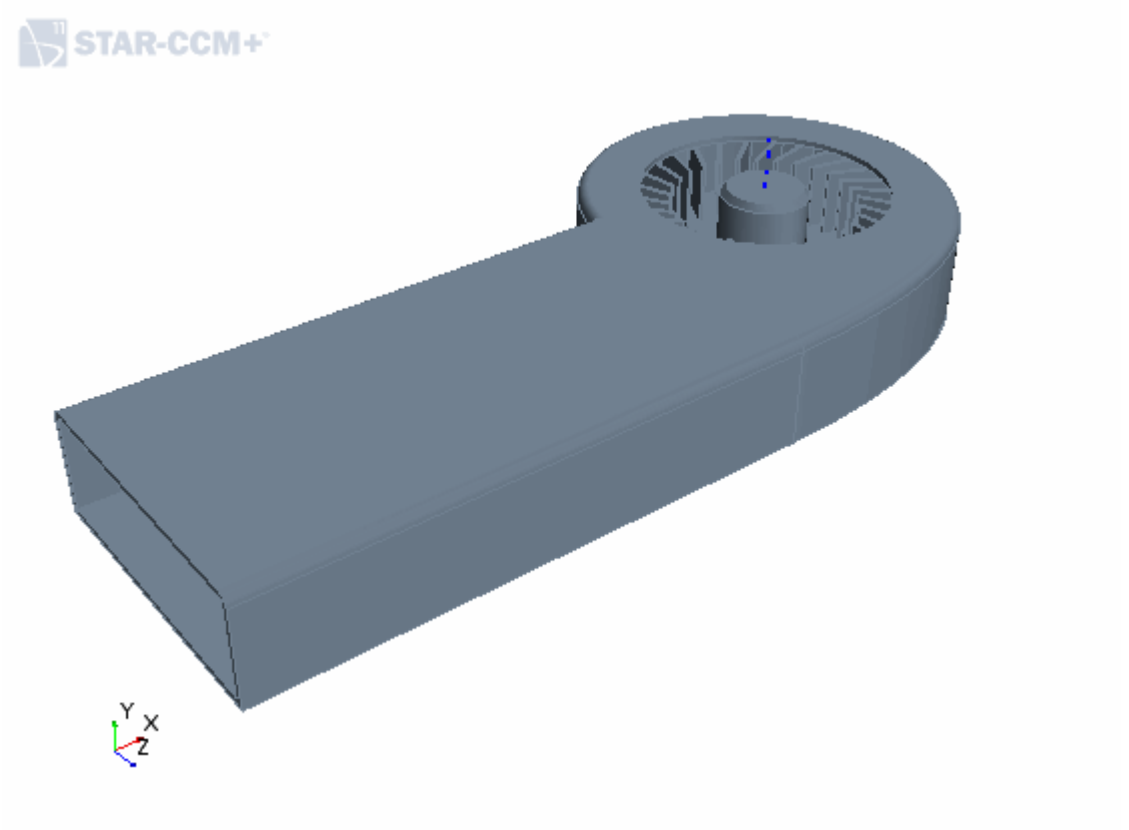
- Starting STAR-CCM+ and creating a simulation.
- Saving and naming the simulation.
- Importing a geometry file.
- Visualizing the imported geometry.
- Splitting and naming the geometry surfaces.
- Renaming techniques.
- Changing types of boundaries.
- Designing a three-dimensional mesh
- Visualizing the mesh.
- Selecting the physics models.
- Defining the initial conditions.
- Defining the boundary conditions and values.
- Setting the solver parameters and stopping criteria.
- Setting up a monitoring report and plot.
- Running the solver until the residuals are satisfactory.
- Disconnecting from a server and reconnecting during a run.
- Analyzing results using the visualization, monitor-plot, and XY-plot features.
- Creating streamlines.
- Reopening a simulation.

Foundation Tutorials

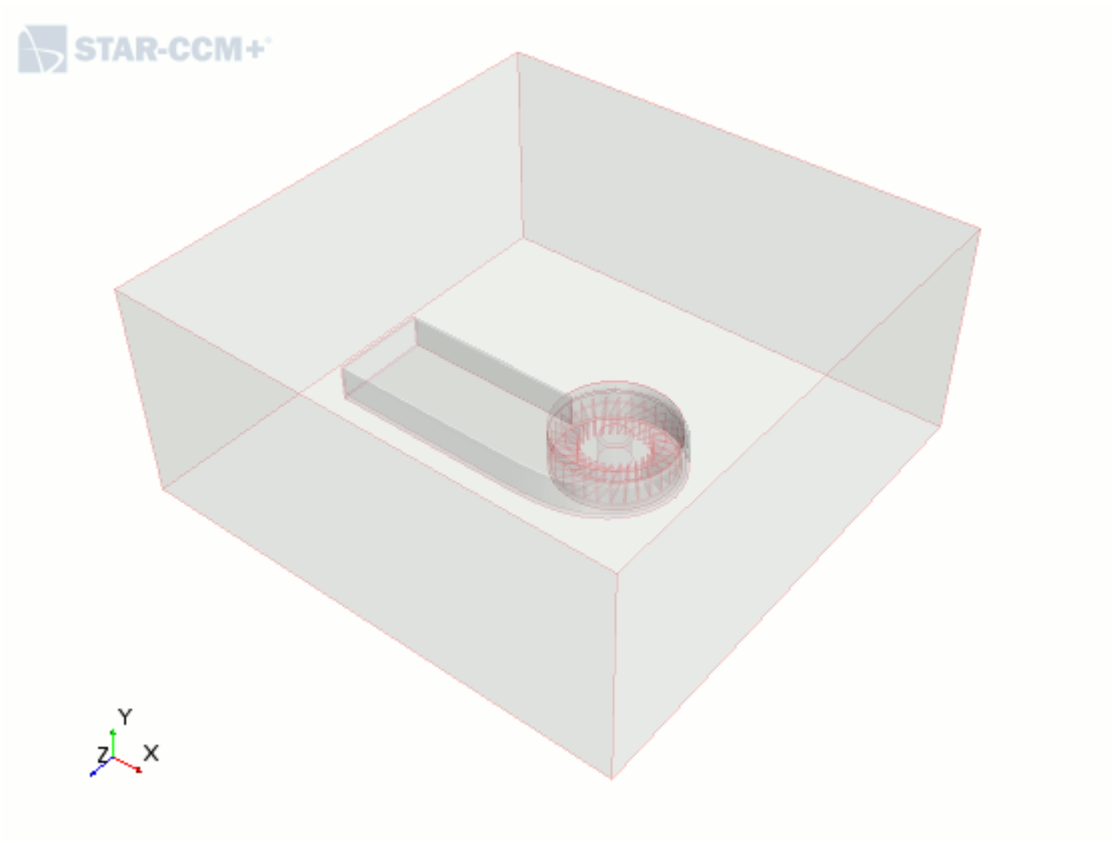
The foundation tutorials showcase the major features of STAR-CCM+ in a series of short tutorials.

The foundation tutorials span the entire workflow in STAR-CCM+, from creating geometry in 3D-CAD, to setting up the physics and post-processing the results.

The main theme for the foundation tutorials is a graphics card with a blower fan. This geometry is well-suited for this purpose as utilises a large number of features available in STAR-CCM+ without being overly complex.



The graphics card is then positioned inside an empty box that emulates a test bench environment.



For this release, the foundation tutorials cover the design and creation of the geometry in 3D-CAD. In future releases, more tutorials will be added to demonstrate how to set up the mesh, the physics, run the solver, and how to post-process the results.

Contents:

3D-CAD
Geometry Parts
Mesh
Physics - Flow
Usability

3D-CAD

Contents:

Basic 3D-CAD: Creating a Sketch with Constraints and Dimensions
Basic 3D-CAD: Creating a Solid Body by Revolving a Sketch
Basic 3D-CAD: Adding Blades as a Pattern of Bodies
Basic 3D-CAD: Sketching a Profile with Curves and Splines
Basic 3D-CAD: Creating the Enclosure Solid Shell
Basic 3D-CAD: Define Fluid Volume for Rotating Fan and Subtract the Blades

Basic 3D-CAD: Creating a Sketch with Constraints and Dimensions

When you wish to create CAD geometry directly using 3D-CAD, you generally start by creating a sketch on a default sketch plane. A closed sketch becomes a solid body when you extrude it or revolve it in subsequent operations.

In this tutorial, you create a profile sketch of a graphics card fan hub.

To start sketching in 3D-CAD:

1. Create a new STAR-CCM+ simulation.
2. Right-click the **Geometry** > **3D-CAD Models** node and select **New**.

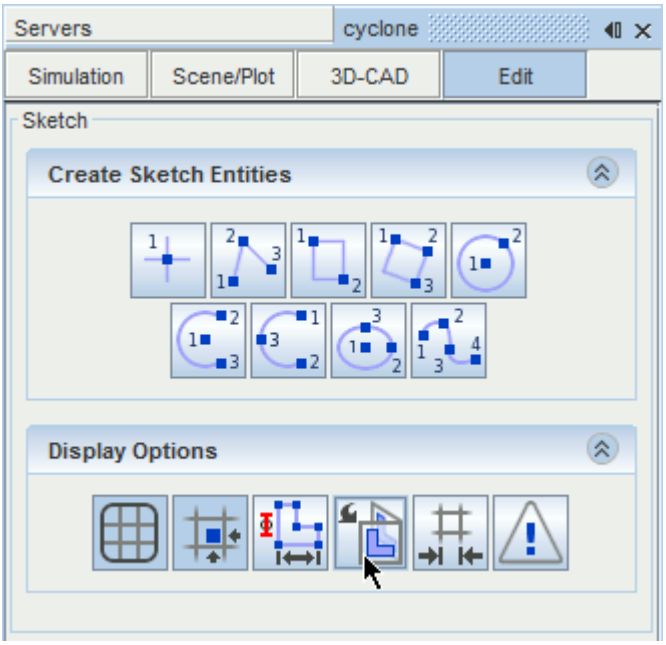
   The 3D-CAD environment launches and a 3D-CAD model is created. Within 3D-CAD, an object tree appears that is specific to the 3D-CAD model under construction. This tree contains **Bodies**, **Features**, and **Design Parameters** nodes.

All geometry in CAD starts from a sketch. Sketches are drawn on sketch planes. 3D-CAD has three default sketch planes (XY, YZ, ZX) that pass through the origin (point [0, 0, 0]).

Create a sketch for the fan hub:

3. In the 3D-CAD feature tree, right-click the **XY** sketch plane and select **Create Sketch**.

   Creating a sketch activates sketch mode, in which a grid is displayed on the sketch plane in the *3D-CAD View* scene, and the *Sketch* panel opens.



4. In the *Sketch* panel, click (**View Normal to Sketch Plane**) to align the sketch plane with the plane of the screen.
5. Specify a more suitable grid spacing for the primary and secondary grids:
   a) Click (**Set Sketch Grid Spacing**).
   b) In the *Grid Spacing* dialog, set *Grid Spacing* to `0.005 m` and *Number of Fine Grid Divisions* to `5.0`.
   c) Click **OK**.
6. Using the wheel action of the middle mouse button, zoom in on the scene.
7. Click (**Create Line**) and move the mouse over the grid in the *3D-CAD View* scene.

   The mouse pointer has a purple sphere that is attached to it, which indicates the position of the point on the grid. The coordinates of this point are displayed in the *Line Properties* box of the *Sketch* panel, and change as you move the mouse. The (**Snap to Grid**) option is active by default, so the point jumps to an intersection between two grid lines as it approaches them.
8. Move the mouse over the origin (position [0, 0]) and click to start drawing.

The **Line** tool creates continuous lines, so the end point of the first line is the start point of the next. The coordinates that are shown in the *Line Properties* box are automatically updated as each line is defined.

9. Move the mouse to the right and draw a horizontal line.

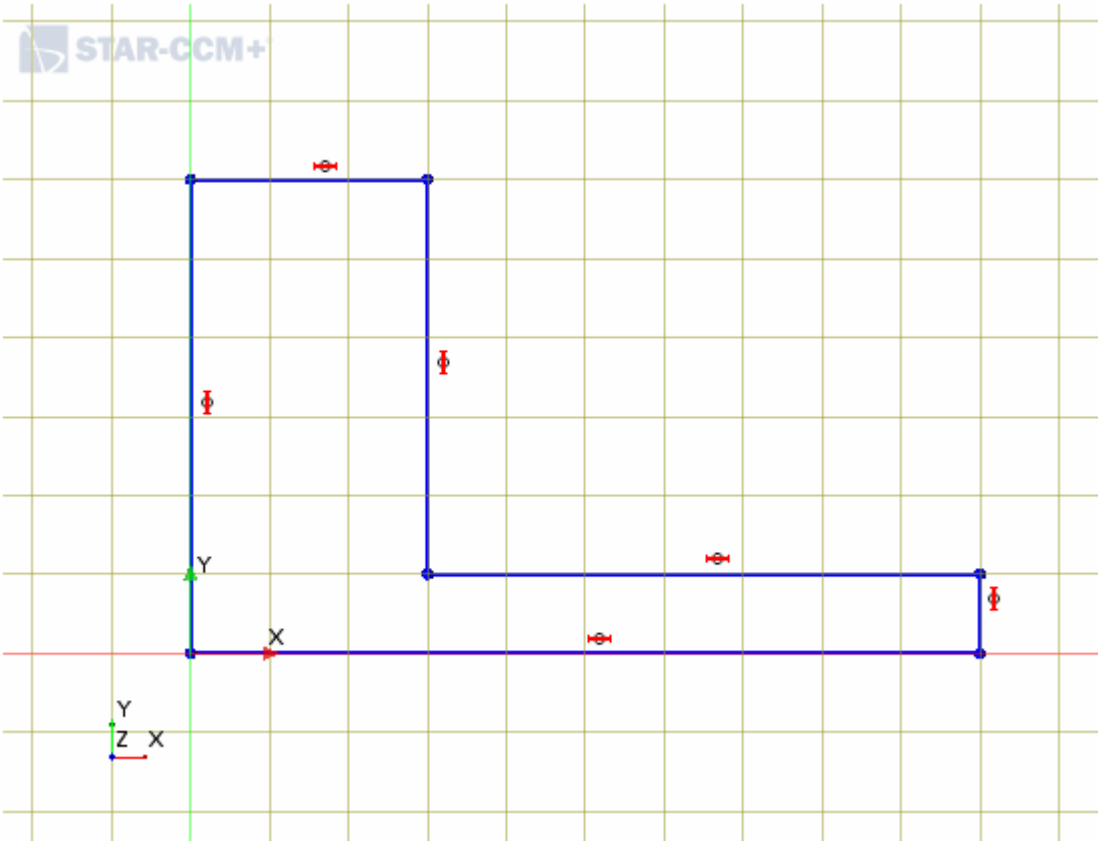
As the line approaches the horizontal position, it snaps and changes to a yellow colour. An icon, depicting the horizontal constraint, appears above the mouse pointer. These visual cues indicate that when you click the mouse button, the horizontal constrain will be applied to the line.

10. Click the left mouse button to place the second point.

    The horizontal line is constrained automatically.

11. Press <`Esc`> to exit the line tool.

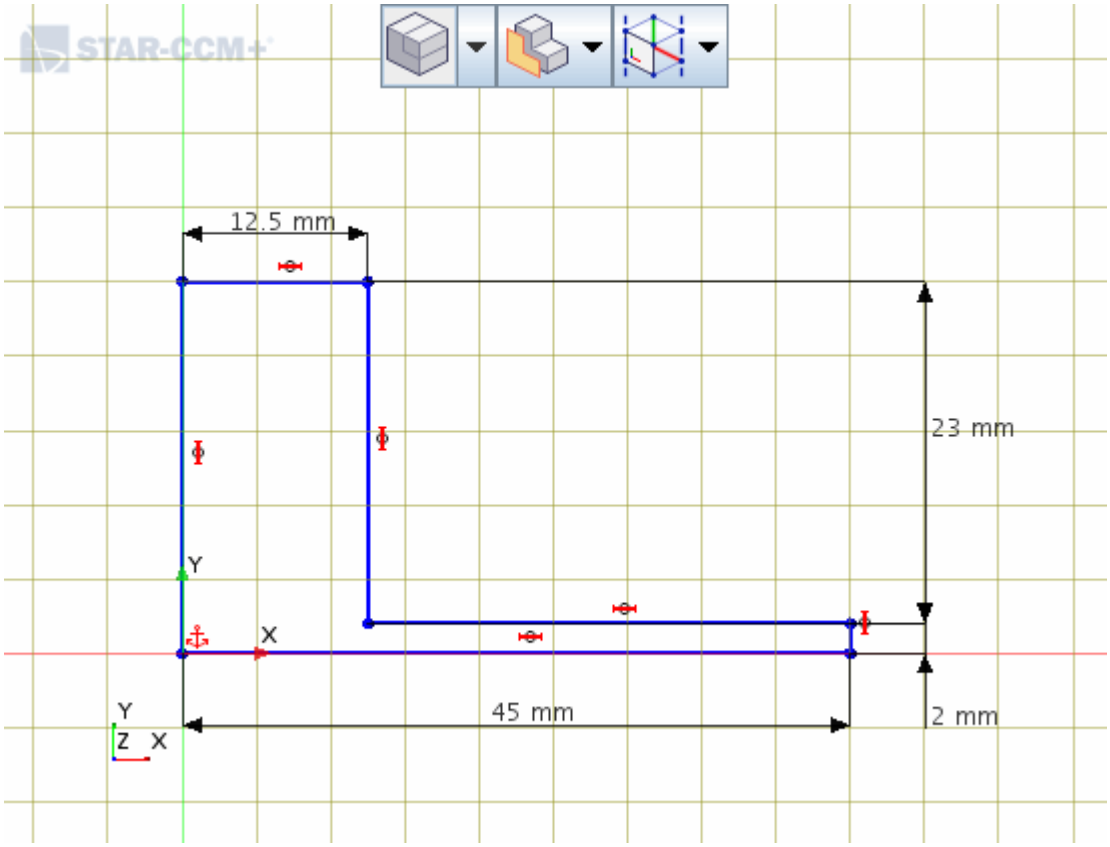
12. Starting from the previous end point, use the line tool to complete the sketch as shown below. You need not press <Esc> after each line—continue clicking until you finally click on the start point of the initial line. When sketching horizontal lines, these are displayed in yellow. Vertical lines are displayed in red. After confirming the sketch position, all lines are displayed in the default blue color.



The initial sketch does not have to be exact. Aim to draw the correct shape before you finalize the sketch.

Finalize the sketch by adding a fixation constraint and dimensions. Constraints maintain the shape of the overall sketch by constraining the movement or orientation of individual entities or groups of entities. Dimensions specify the size of individual entities. Typically, you apply constraints before you add dimensions.

13. Right-click the point on the origin and select **Apply Fixation Constraint**.
14. Add dimensions to the sketch:
    a) Right-click the bottom horizontal line and select **Apply Length Dimension**.
    b) In the *Dimension* dialog, set the length to **45 mm**.
    c) Click **OK**.
15. Continue to dimension the sketch using length dimensions. Use the following diagram as a guide:



Check the status of the sketch:

16. Click (**Show Status of Sketch Primitive**).

    The status of sketch primitives indicates the validity of the sketch. This feature is useful for finding issues such as small gaps in sketches or overlapping primitives. Leaving these issues undetected can lead to errors later in the design process. Black edges and vertices indicate that the sketch is fully constrained.

17. Click (**Show Status of Sketch Primitive**) again to hide the status of the sketch.
18. In the *Sketch* panel, click **OK** to exit sketch mode.

    A sketch feature, **Sketch 1**, is added to the **Features** node.
19. Rename the sketch feature to something more descriptive:
    a) Right-click **Features** > **Sketch 1** and select **Rename**.
    b) In the inline editor, enter `Fan Hub Profile`.
20. Save the simulation as `foundationTutorial_1.sim`.

Basic 3D-CAD: Creating a Solid Body by Revolving a Sketch

In this 3D-CAD tutorial, you create a solid body by revolving a closed sketch. The axis about which you revolve the sketch can be part of the sketch itself in the form of a construction line, or can be defined in the revolve feature.

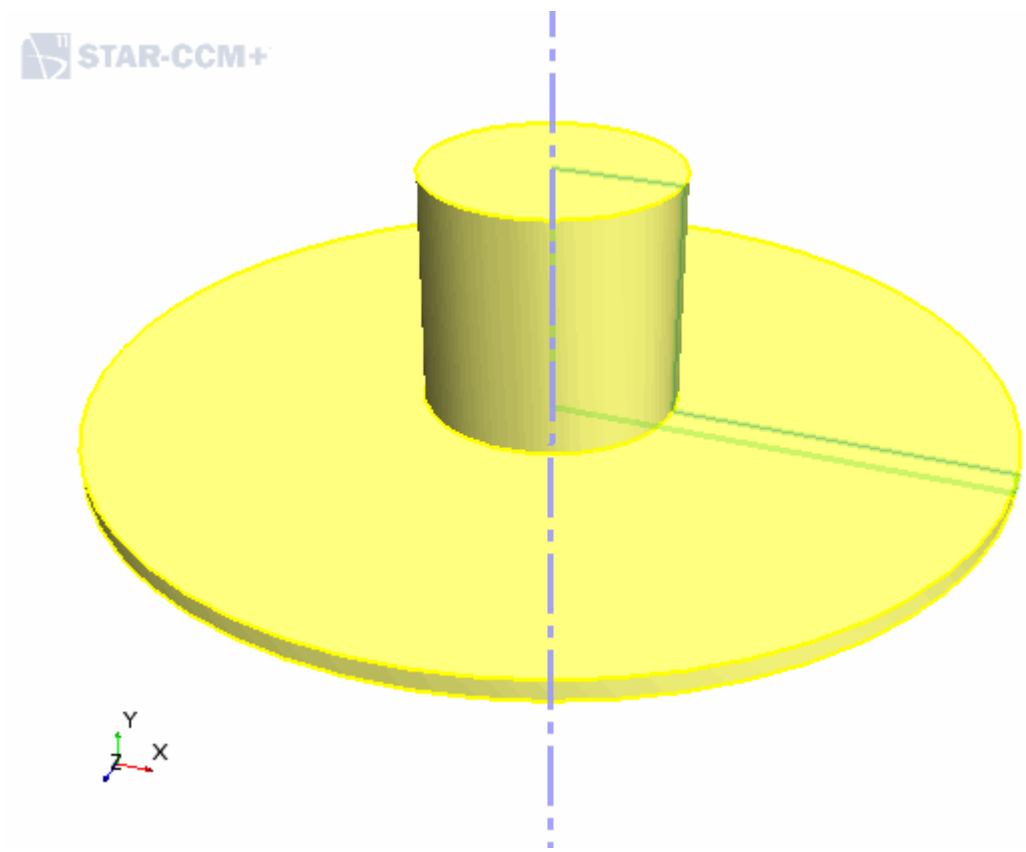
In this tutorial, you define the axis of rotation in the revolve feature rather than using a construction line. If a sketch contains a construction line, and you choose to revolve the sketch, 3D-CAD automatically uses the construction line as the axis of rotation.

To revolve the sketch:

1. Launch STAR-CCM+ and load **foundationTutorial_1.sim**.

   You can either use the sim file that you saved from the previous tutorial, or load the sim file provided in `[INSTALL_DIR]/doc/startutorialsdata/foundationTutorials/data`.
2. Save the simulation as `foundationTutorial_2.sim`.
3. To resume the existing 3D-CAD model, right-click **Geometry** > **3D-CAD Models** > **3D-CAD Model 1** and select **Edit...**.
4. Revolve the sketch to form a solid body:
   a) Right-click **Fan Hub Profile** and select **Revolve**.
   b) In the *Revolve* panel, set *Angle* to `360`.
   c) Set *Axis Direction* to `[0.0, 1.0, 0.0] m`.

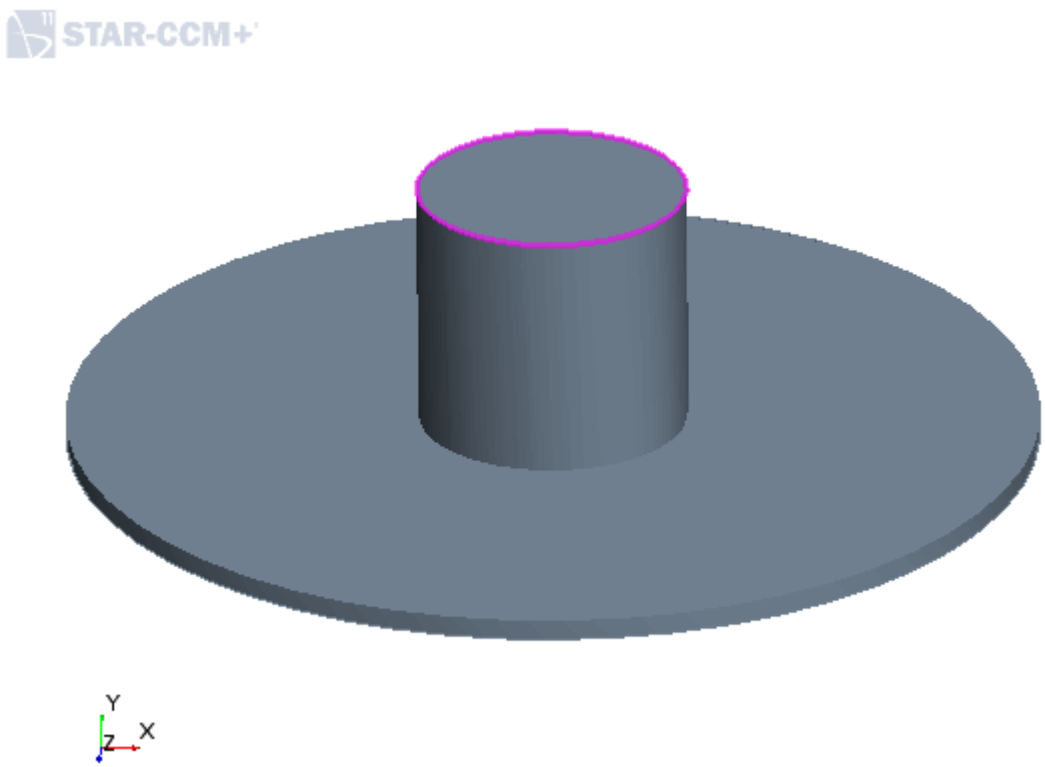
      This value sets the Y axis as the axis rotation. As the sketch is already located on the origin, you can accept the default *Axis Position*.



   d) Click **OK**.

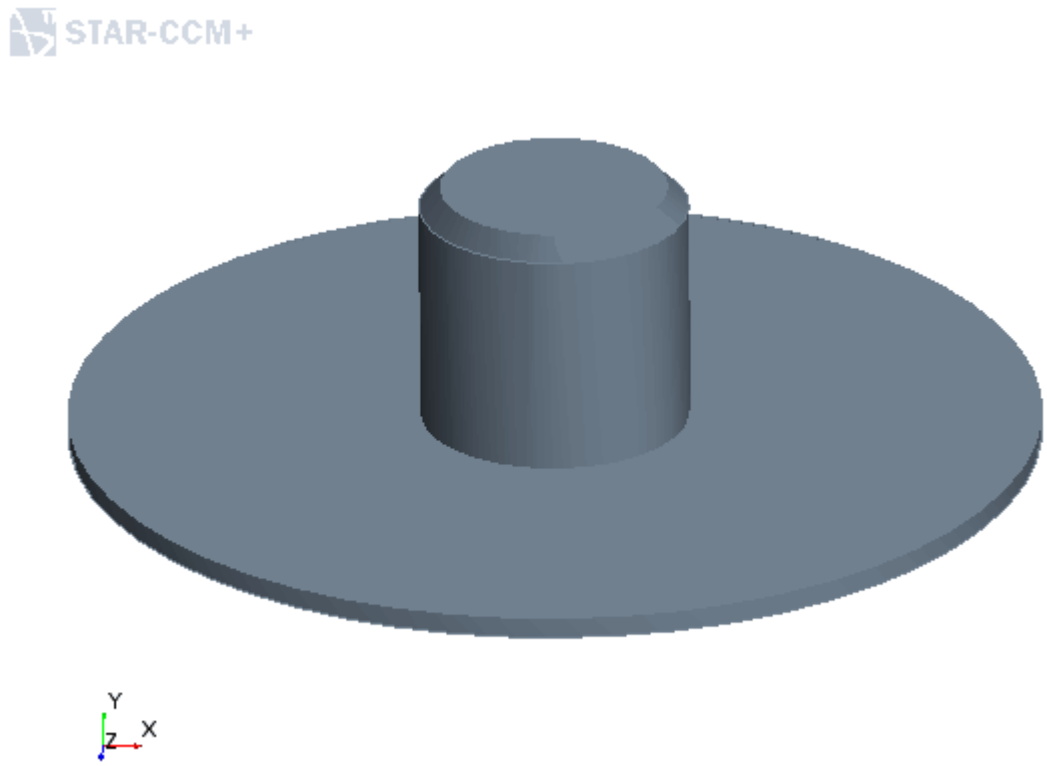
      A revolve feature, **Revolve 1**, is added the **Features** node.
   e) Rename **Features** > **Revolve 1** to `Fan Hub`.

5. Add a chamfer to the top edge of the fan hub:
   a) In the *3D-CAD View* scene, right-click the top edge on the cylinder and select **Create Chamfer**.



   b) In the **Chamfer** panel, set *Distance 1* and *Distance 2* to `2 mm` each.
   c) Click **OK**.
      A chamfer feature, **Chamfer 1**, is added the **Features** node.
   d) Rename **Features** > **Chamfer 1** to `Hub Chamfer`.
6. Save the simulation.

The fan hub is complete.

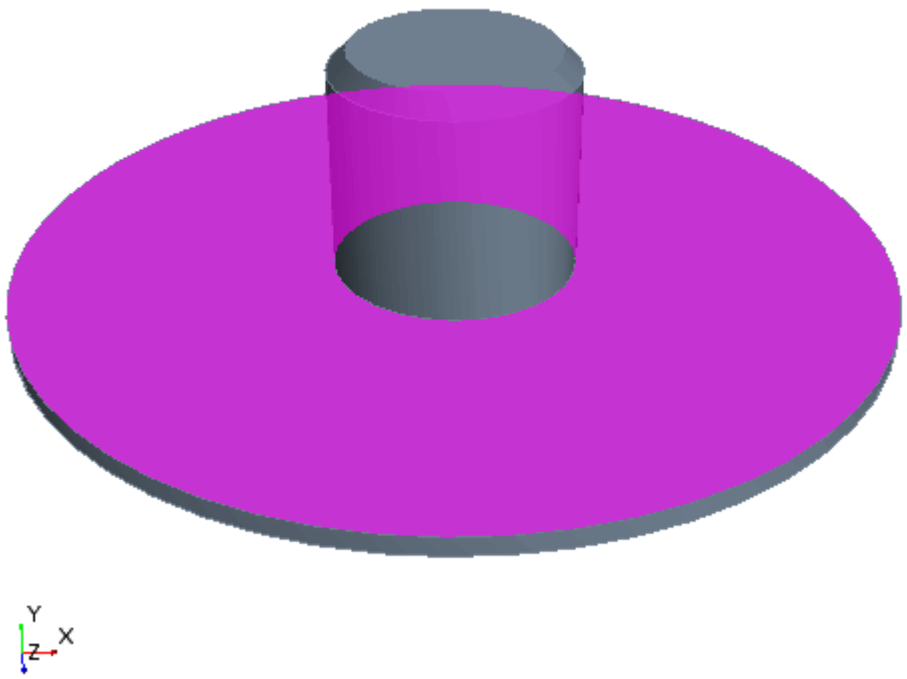


Basic 3D-CAD: Adding Blades as a Pattern of Bodies

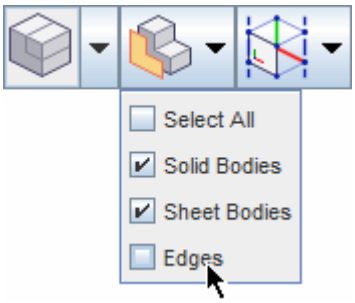
3D-CAD allows you to create a set of identical bodies in an ordered pattern. Here, you create a single fan blade and apply it in a pattern around the central axis.

To create the fan blade, first place a sketch on the large flat surface of the fan hub:

1. Launch STAR-CCM+ and load **foundationTutorial_2.sim**.
2. Save the simulation as `foundationTutorial_3.sim`.
3. Resume the 3D-CAD model.
4. Sketch the outline of a single fan blade:
   a) Right-click the face shown below and select **Create Sketch** > **On Face**.



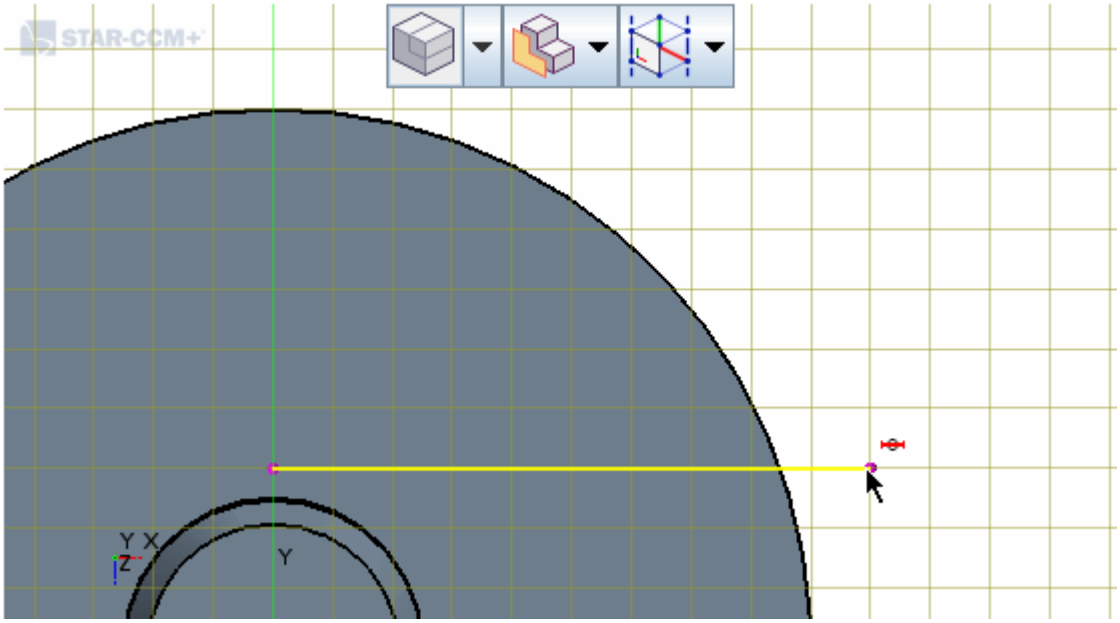
b) In the toolbar at the top of the *3D-CAD View* scene, click **Model Visibility** and activate the **Edges** option.



The edges on the geometry are displayed.

c) In the *Sketch* panel, click (**View Normal to Sketch Plane**).

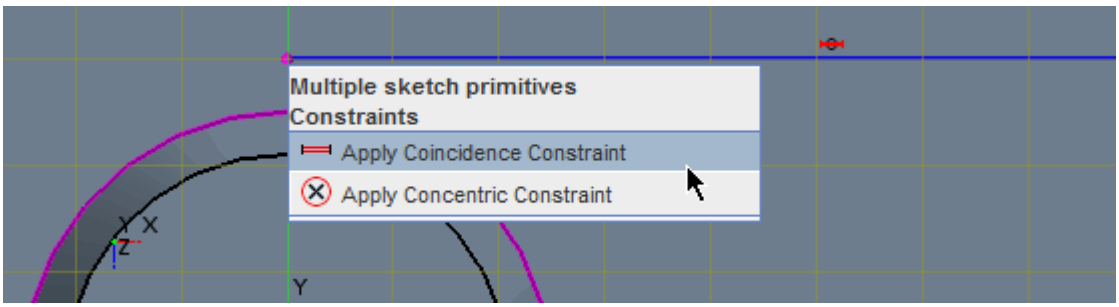
d) Click (**Create Line**) and draw a horizontal line from the outer edge of the hub center that extends beyond the flat surface of the hub, and then click <`Esc`>.



e) Multi-select the starting point of the sketch and the outer circle of the hub center.

Press and hold `<CTRL>` after the initial click to select additional entities.

f) Right-click the selected corner and choose **Apply Coincidence Constraint**.



When applying constraints, 3D-CAD allows you to reference entities from other bodies, such as the edges of the hub. 3D-CAD automatically projects these entities to the active sketch and applies the constraints.

g) Multi-select the same outer circle on the hub and the sketch line.

h) Right-click one of the edges and choose **Apply Tangent Constraint**.

i) Right-click the line entity and apply a length dimension of `44 mm`.

j) Click **OK**.

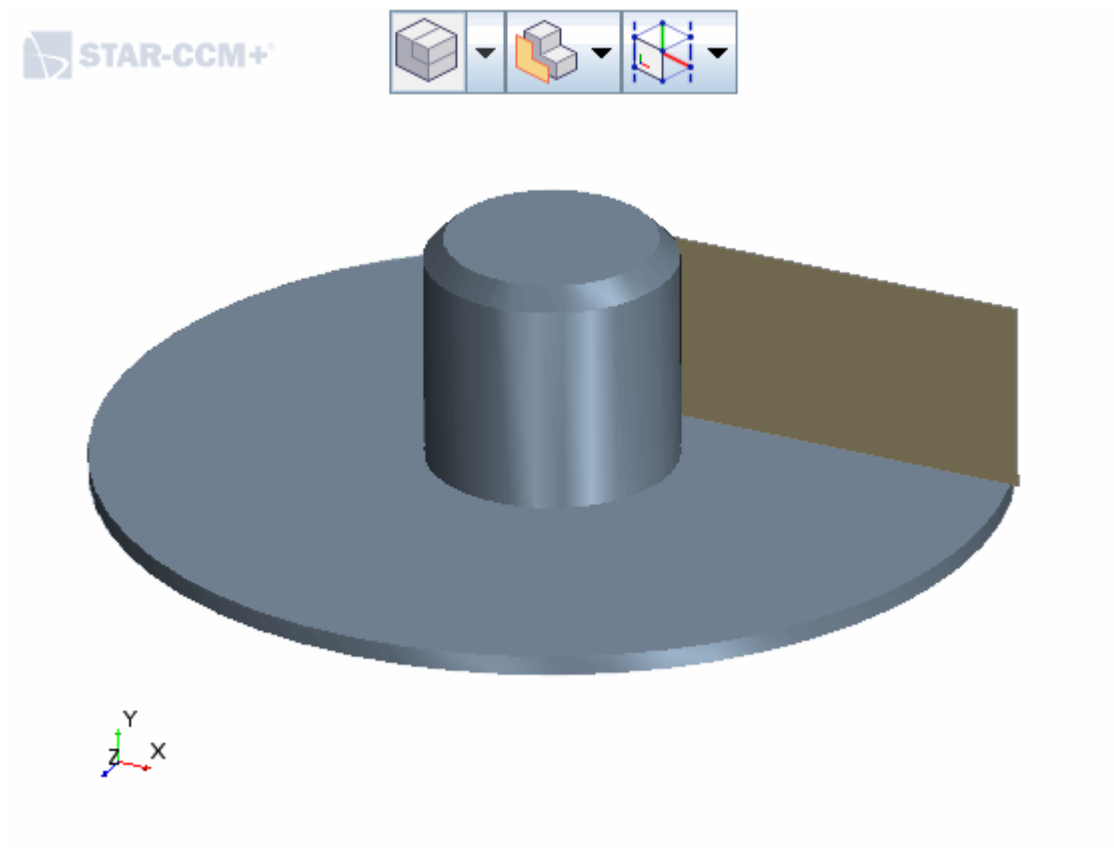
k) In the toolbar at the top of the *3D-CAD View* scene, click **Model Visibility** and deactivate the **Edges** option.

l) Rename **Features** > **Sketch 1** to `Fan Blade Outline`.

5. Extrude the sketch to form the solid fan blade:

a) Right-click **Fan Blade Outline** and select **Extrude**.

b) In the *Extrude* panel, set *Distance* to `19 mm` and click **OK**.



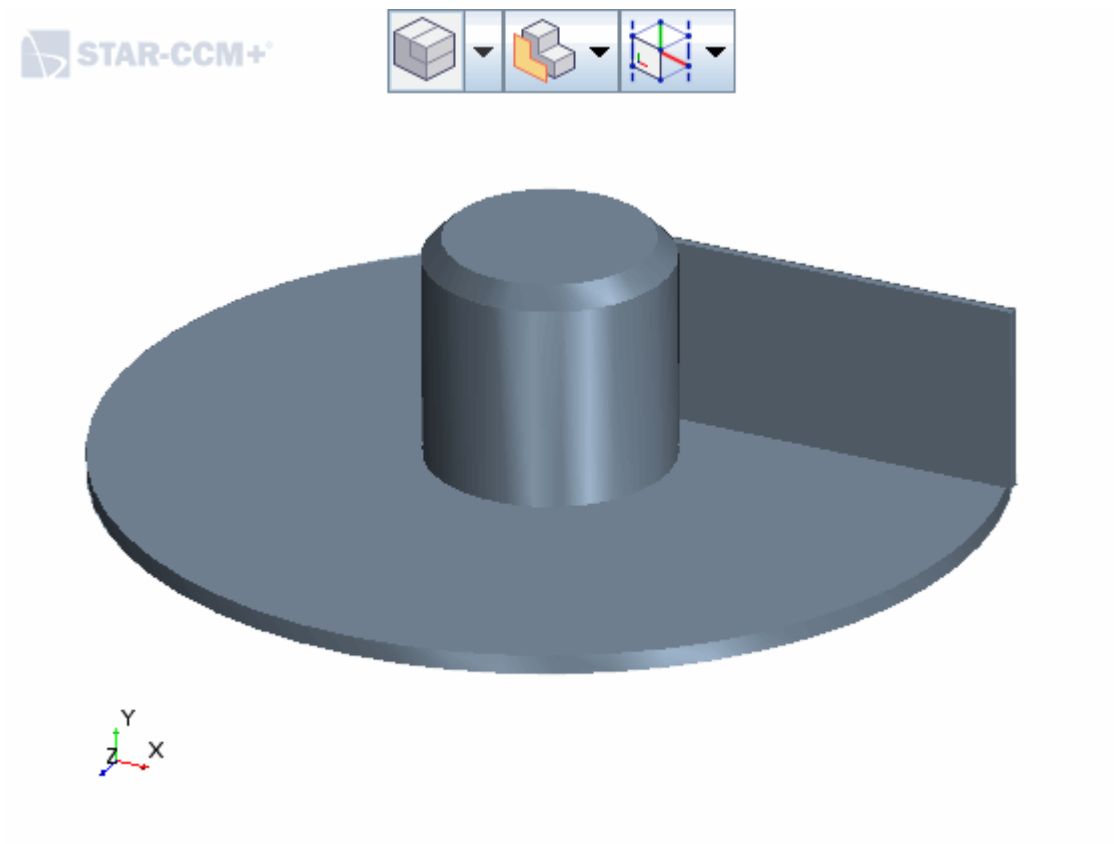
The brown color of the blade body means it is a sheet body. Sheet bodies have zero thickness.

c) Rename **Features** > **Extrude 1** to `Fan Blade`.

6. Thicken the blade body:

   a) In the *3D-CAD View* scene, right-click the blade body and select **Body** > **Thicken**.

   b) In the *Thicken* panel, set *Thickness* to `1 mm` and click **OK**.



The sheet body is converted to a solid body.

c) Rename **Features** > **ThickenSheetBodies 1** to `Fan Blade Thicken`.

7. Slice the blade to remove the part of the fan blade that extends beyond the fan hub:

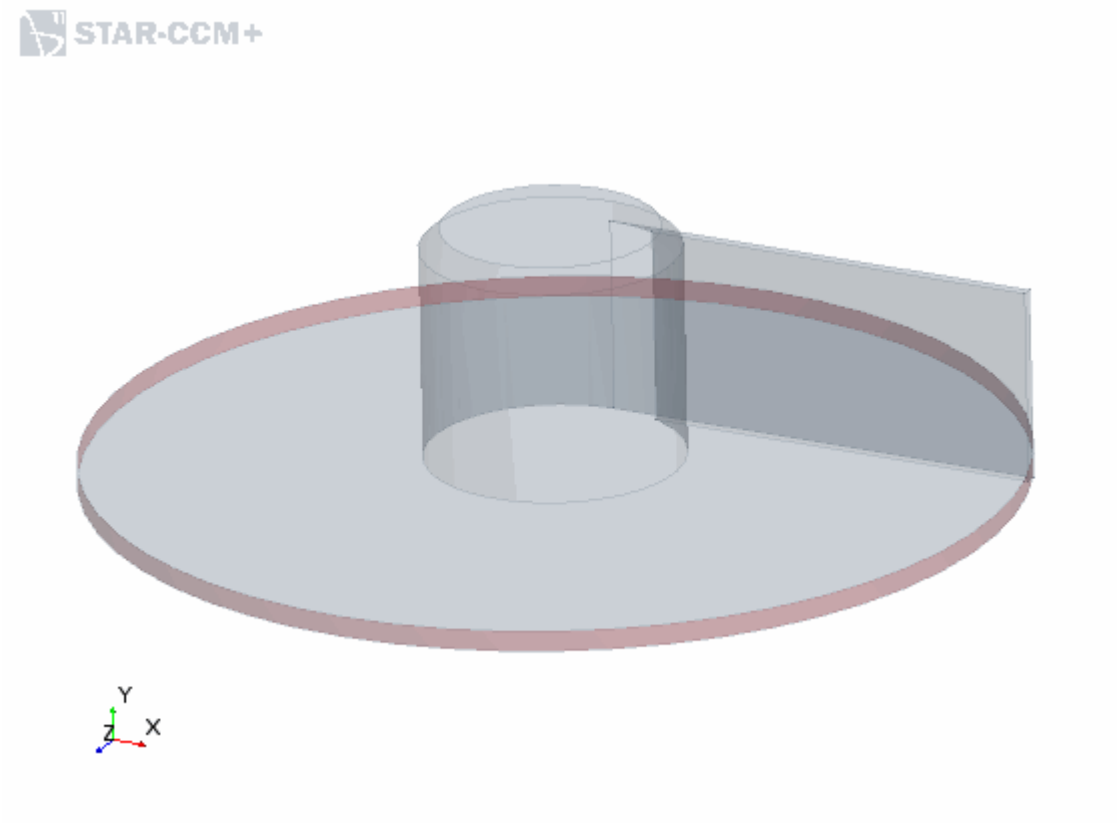
a) Expand the **Bodies** node, right-click **Body 2** and select **Boolean** > **Slice**.

b) In the *Slice* panel, set the following parameters:

Parameter	Setting
Keep Side	**Back Side**
Slice By	**Face**

c) Click inside the *Tool Faces* group box to activate the selection mode for this property.

d) In the *3D-CAD View* scene, select the face on the side of the fan hub as shown below.



e) Activate the **Extend Face** option to extend the selected face through the protruding fan blade.

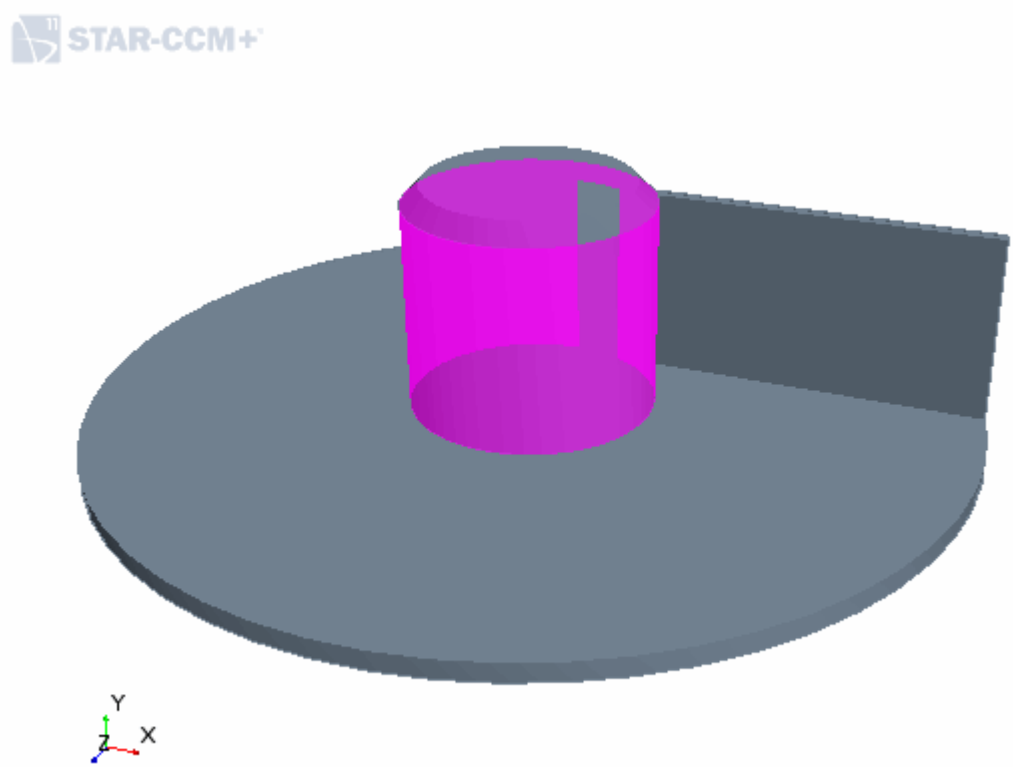
f) Click **OK** to slice the protrusion.

g) Rename the **Features** > **SliceBodies 1** to `Fan Blade Slice`.

Create the remaining fan blades around the fan hub using a pattern. You can pattern the existing fan blade about the axis of revolution.

8. Create a reference axis:

   a) Right-click the neck of the fan hub and select **Face** > **Reference Geometry** > **Axis From Revolved Entity**.



b) In the *Axis From Revolved Entity* panel, click **OK**.

An axis feature, **Axis 1**, is added to the **Features** node.

c) Rename **Features** > **Axis 1** to `Axis of Revolution`.

9. Create a pattern about the axis of revolution:

a) Right-click the **Bodies** > **Body 2** node and select **Pattern** > **Circular Pattern**.

b) In the *Circular Pattern* panel, set *Axis Type* to **Reference Axis**.

c) Click inside the **Reference Axis** field box to activate the selection for this feature, and then select the reference axis from the *3D-CAD View* scene.

d) Set *Number of Instances* to `30`.

This option controls the total number of fan blades in the pattern.

e) Click (**Expose Design Parameter**) next to *Number of Instances*.

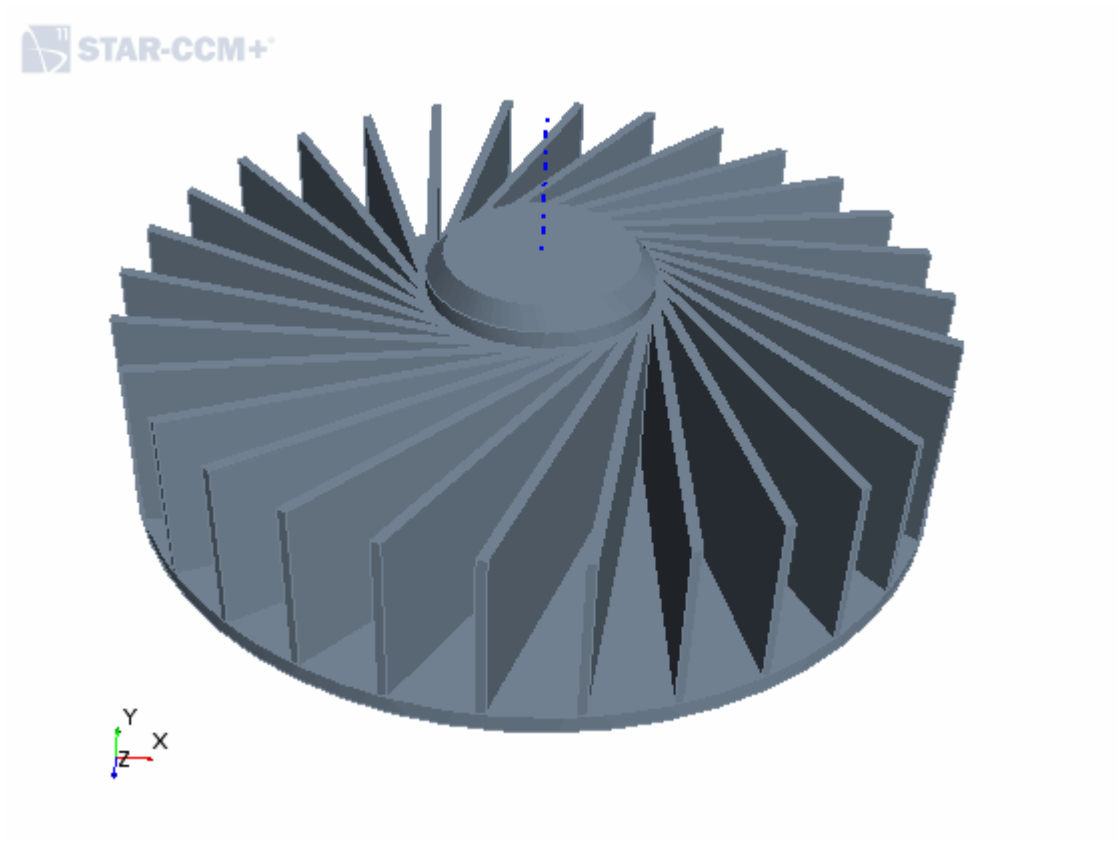
f) In the *Design Parameter* dialog:

1. Activate the **Expose Parameter** option.
2. Set *Name* to `Number of Fan Blades`.
3. Click **OK**.

This option exposes the parameter that controls the number of fan blades in the pattern so that you can access it outside of 3D-CAD at the parts level.

g) Click **OK** to create the pattern.

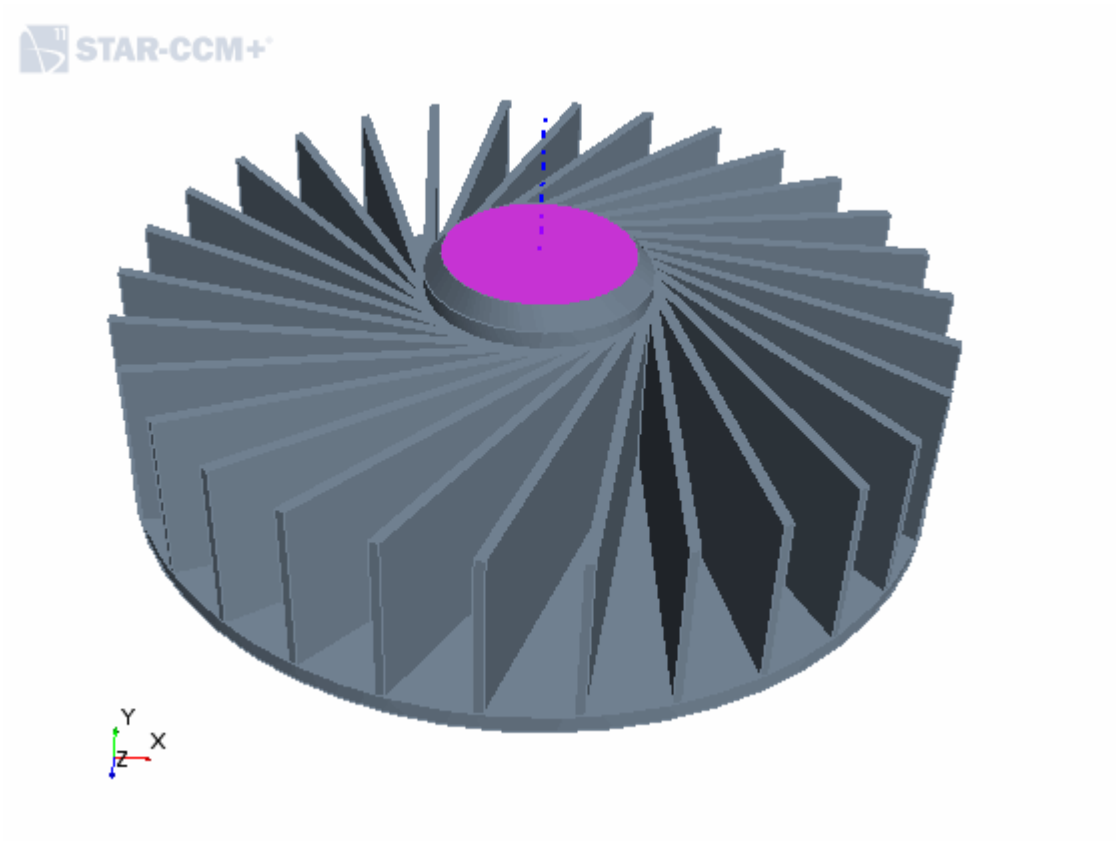
A pattern feature, **Circular Pattern 1**, is added to the feature tree.



h) Rename **Features** > **Circular Pattern 1** to `Fan Blade Pattern`.

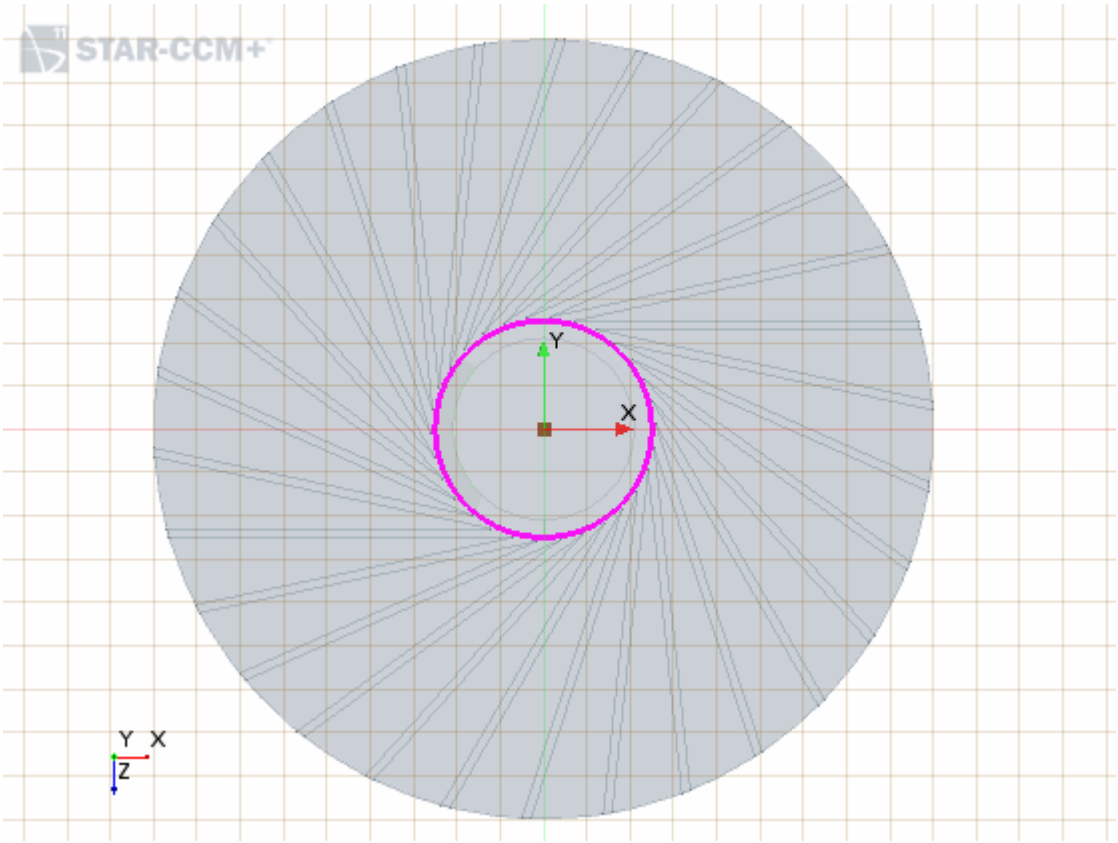
10. Cut away a portion of the fan blades near the center of the hub to allow more air to flow into the fan as it spins:

    a) Right-click on the top face of the fan hub and select **Create Sketch** > **On Face**.



b) In the sketch, right-click the outer edge on the hub center and select **Project to Sketch**.

Use the (**Make Scene Transparent**) tool to make it easier to visualize the sketch. Remember to deactivate this tool when done.



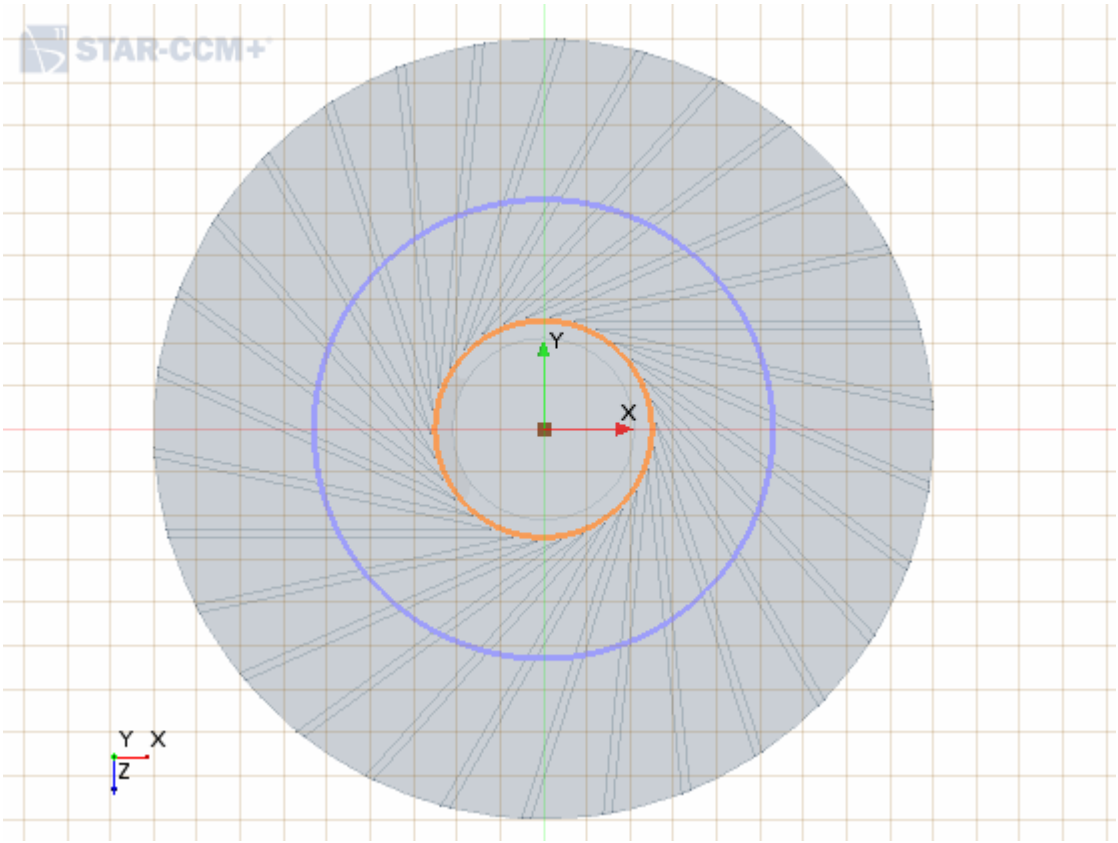
The edge from the fan hub body is projected to the active sketch. Projected edges are displayed in orange in Sketch mode. Projected entities are linked to the underlying edge on the parent body, so that if the parent body changes, the projected edge changes as well.

c) Right-click the projected edge and select **Offset Sketch Primitive**.

d) In the *Offset Sketch* panel, set *Offset Distance* to `14 mm` and click **OK**.

Make sure *Offset Options* is set to **Outwards**.

A copy of the selected edge is offset by the specified amount.



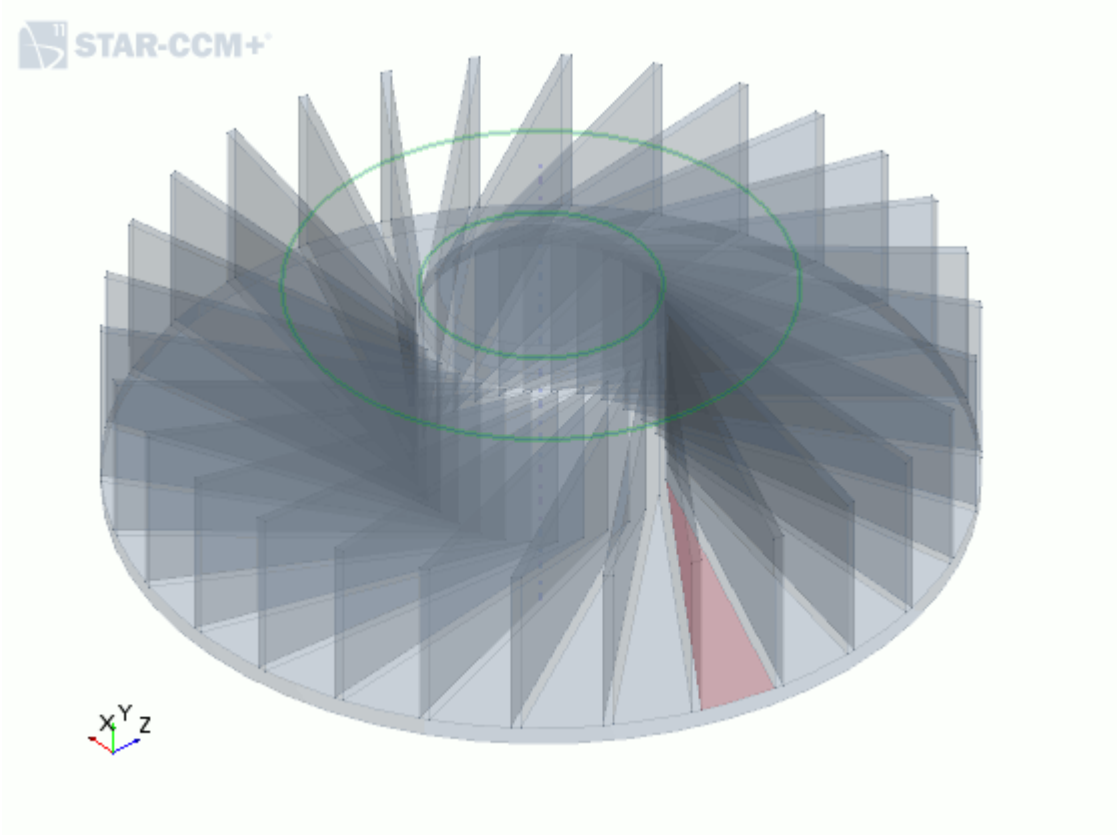
e) Click **OK** to exit sketch mode.

f) Rename **Features** > **Sketch 1** to `Fan Blade Cut Outline`.

g) Right-click **Fan Blade Cut Outline** and select **Extrude Cut**.

h) In the *Extrude Cut* panel, set *Method* to **Up to Face**.

i) In the *3D-CAD View* scene, select the face shown below.



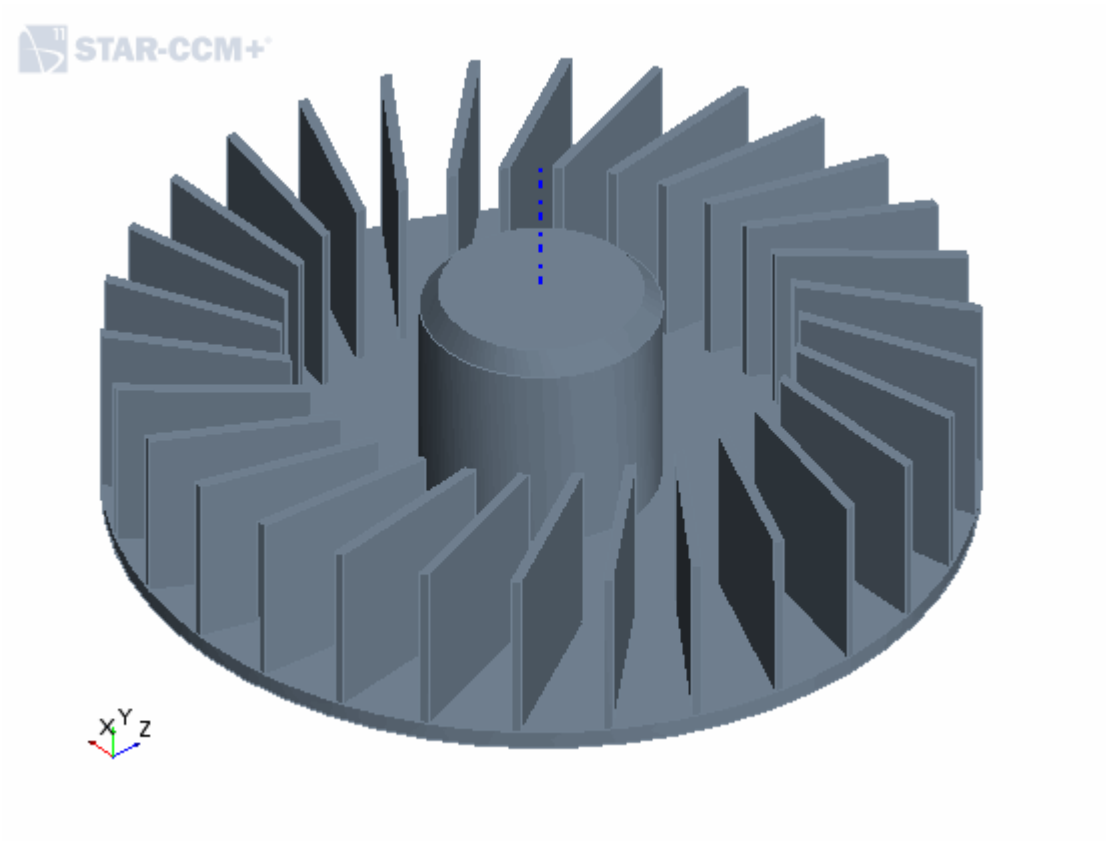
j) Click **OK**.

The portion of the fan blades near the hub center are cut.

k) Rename **Features** > **Extrude Cut 1** to `Fan Blade Cut`.

11. Expand the **Bodies** node and rename **Body 2 (2)** to `Fan`.
12. Save the simulation.

The finished fan is displayed below:



Basic 3D-CAD: Sketching a Profile with Curves and Splines

Sketch profiles can contain a combination of simple curves, such as arcs and splines. Split and trim tools allow you to remove unwanted curves within a shape. Here, you create a sketch for the fan enclosure.

The fan is positioned 2 mm above the bottom of the fan enclosure. The fan enclosure is 2 mm thick.

To sketch the fan enclosure:

1. Launch STAR-CCM+ and load **foundationTutorial_3.sim**.

   You can either use the sim file that you saved from the previous tutorial, or load the sim file provided in `[INSTALL_DIR]/doc/startutorialsdata/foundationTutorials/data`.
2. Save the simulation as `foundationTutorial_4.sim`.
3. Resume the 3D-CAD model.
4. Create a transform sketch plane to hold the sketch of the enclosure:
   a) Rotate the fan so that the underside is visible.
   b) Right-click the underside face and select **Face** > **Reference Geometry** > **Plane by Transformation**.
   c) In the *Plane by Transformation* panel, set the *Z* component of the *Translation Vector* to `4 mm`.

      The fan enclosure is 2 mm thick. The fan is positioned 2 mm above the bottom surface of the enclosure. The total distance to offset the transform sketch plane is 4 mm.
   d) Click **OK**.

      A sketch plane feature, **Plane 1**, is added to the **Features** node.

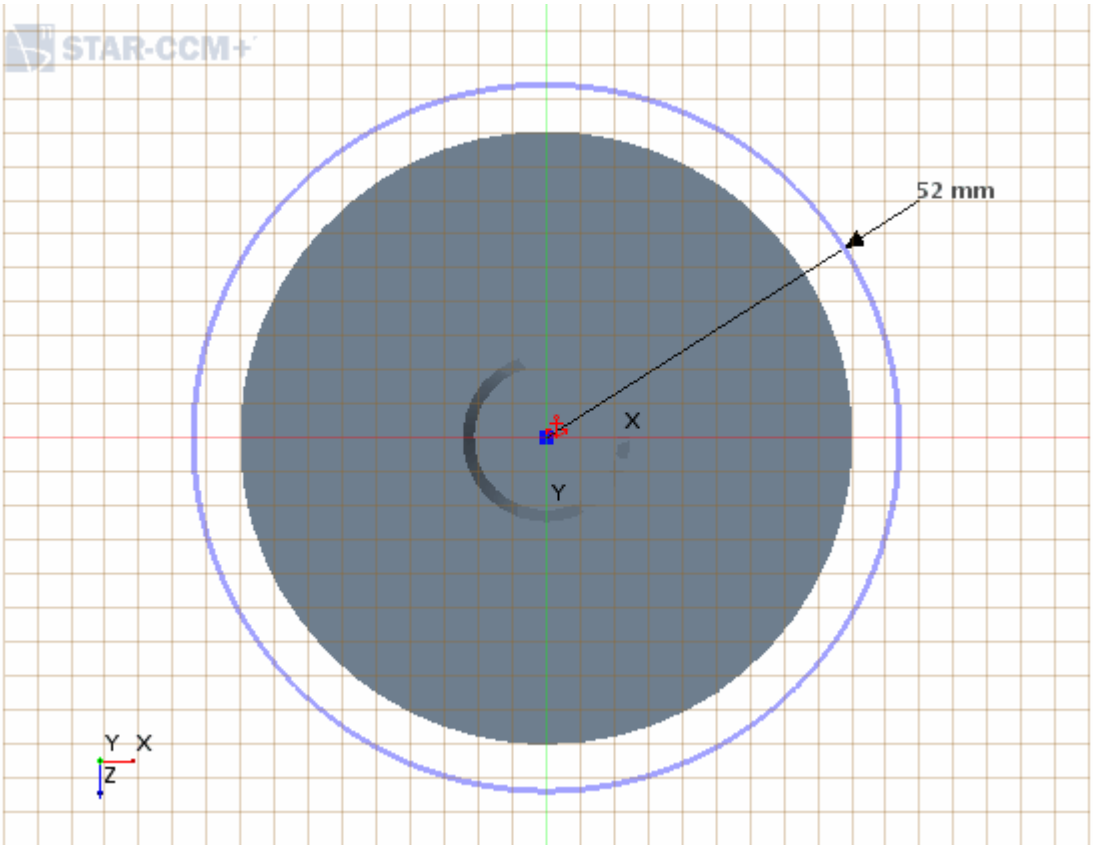
e) Rename **Features** > **Plane 1** to `Enclosure Plane`.

Create the sketch for the card enclosure:

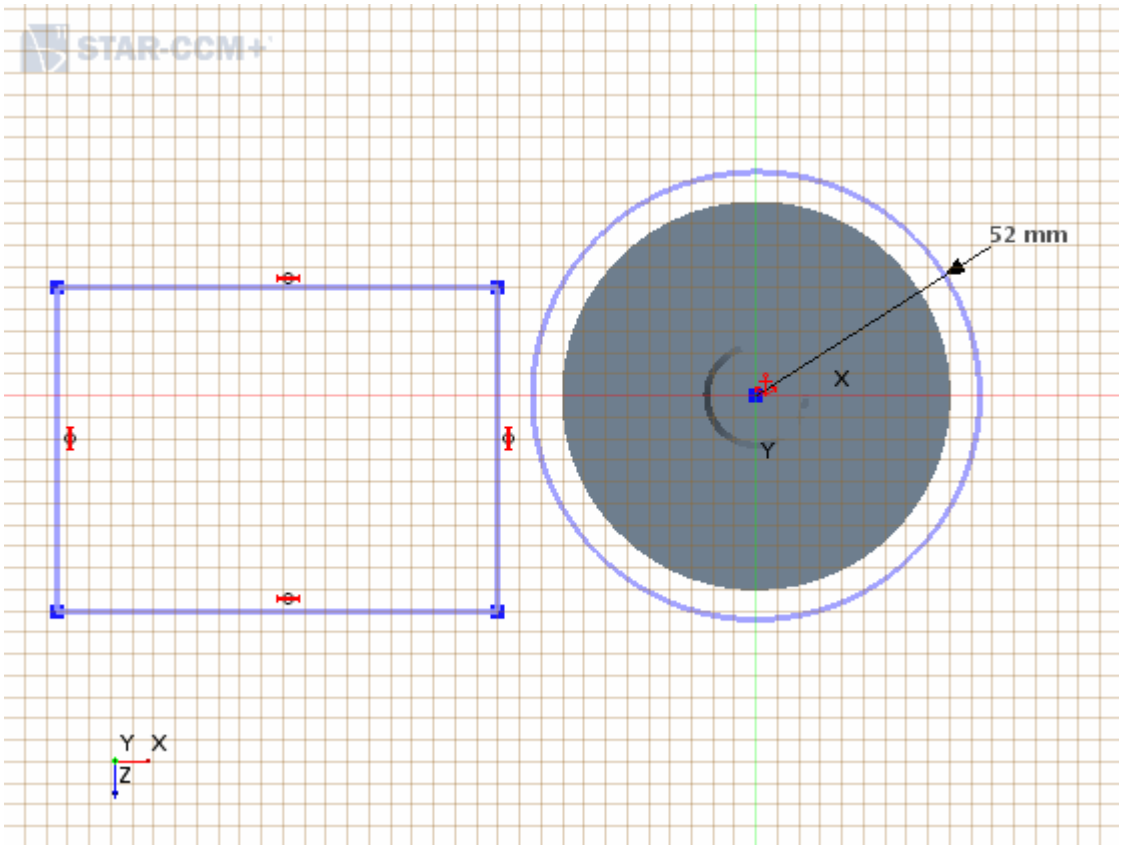
5. In the *3D-CAD View* scene, right-click the visible sketch plane and select **Create Sketch**.

6. To view the sketch from the top of the card, from the *Vis* toolbar, click (**Save-Restore-Select Views**) and select **Views** > **+Y** > **Up -Z**.

7. Click (**Create Circle**) and draw a circle with center at the origin.
8. Set a fixation constraint on the center of the circle and set the radius of the circle to `52 mm`.

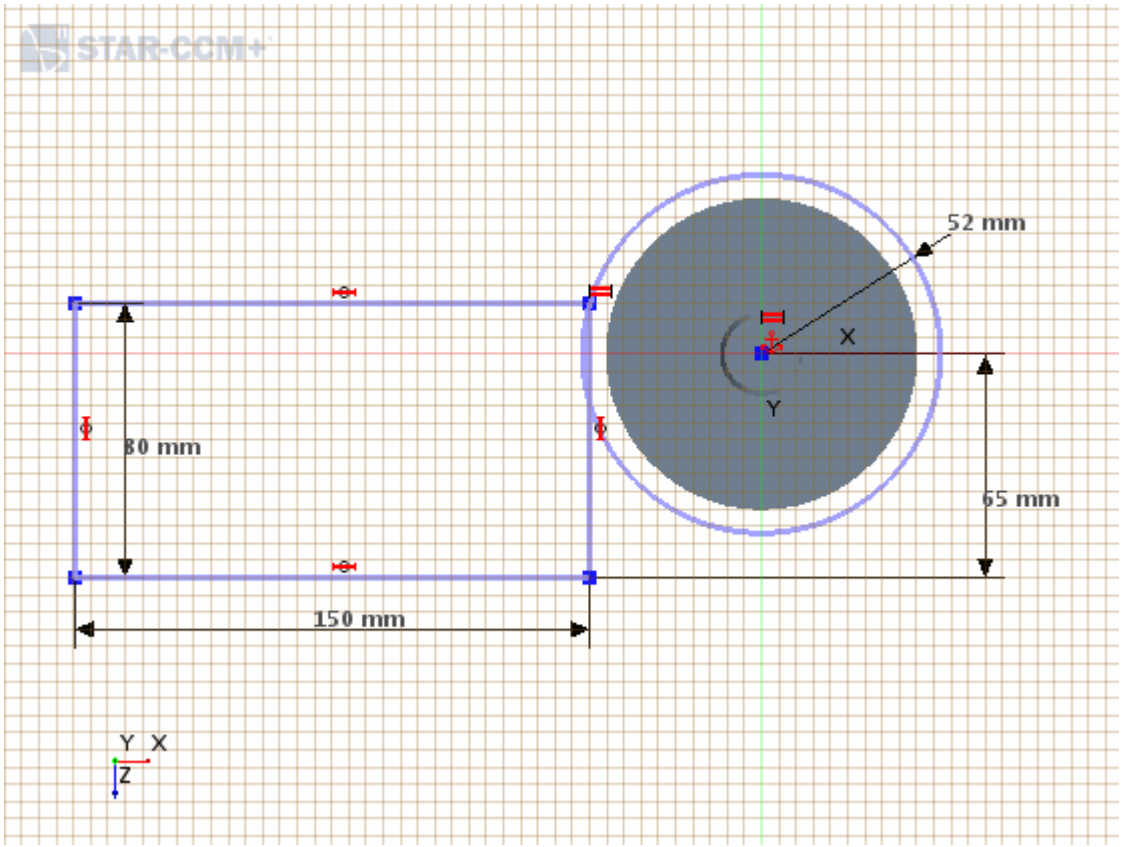


9. Click (**Create Rectangle**) and draw a rectangle to the left of the circle.



10. Dimension the rectangle so that the vertical edge is `80 mm` tall and the horizontal edge is `150 mm` long.

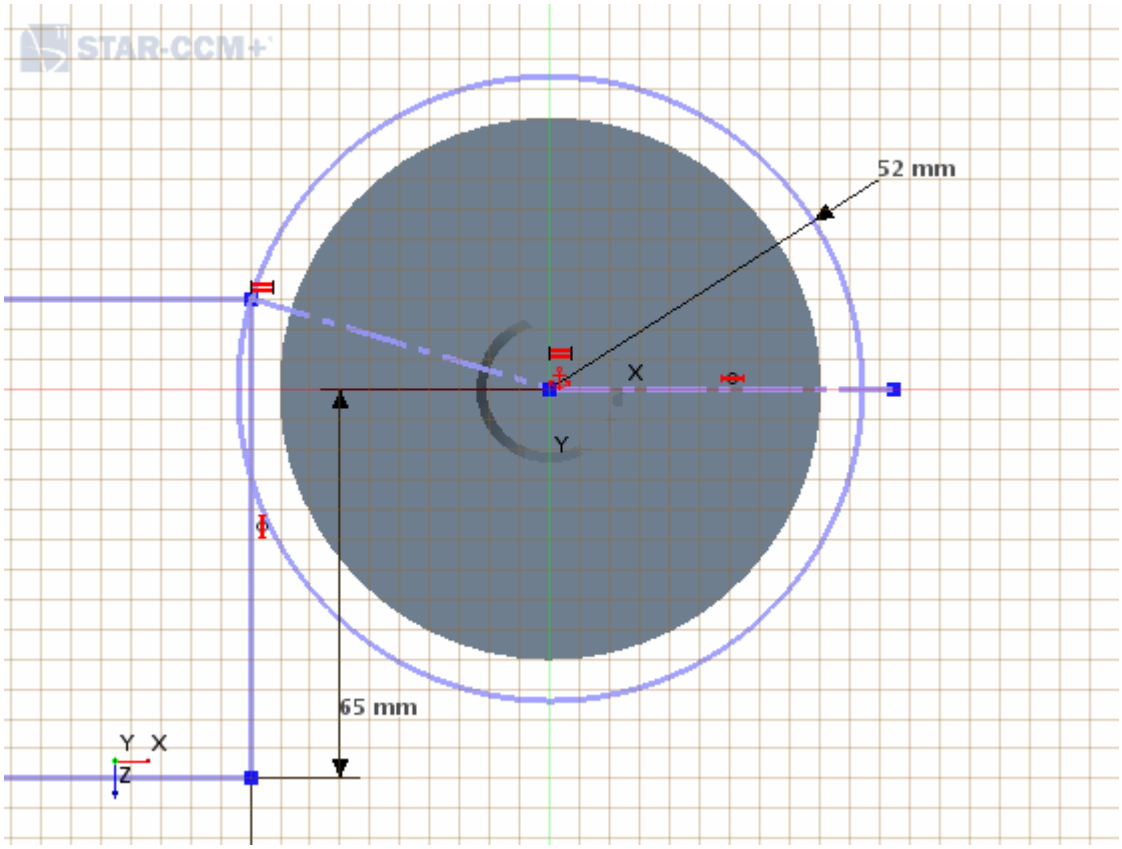
    If the rectangle moves away from the circle, you can click and drag it closer to the circle.
11. Make the top-right point of the rectangle coincident with the circle:
    a) Multi-select the point at the top-right corner of the rectangle and the edge of the newly created circle.
    b) Right-click a selection and choose **Apply Coincidence Constraint**.
12. Multi-select the bottom-right corner (point) of the rectangle and the center of the circle and set a vertical distance of 65 mm between these points.



13. Using the line tool, create a line between the top-right corner of the rectangle and the center of the circle.
14. Create another straight line that starts from the center of the circle and extends beyond the circle to the right.

    Apply a horizontal constraint to this line.
15. Select both straight lines and set them as construction lines.

    You will use these lines for trimming the circle.

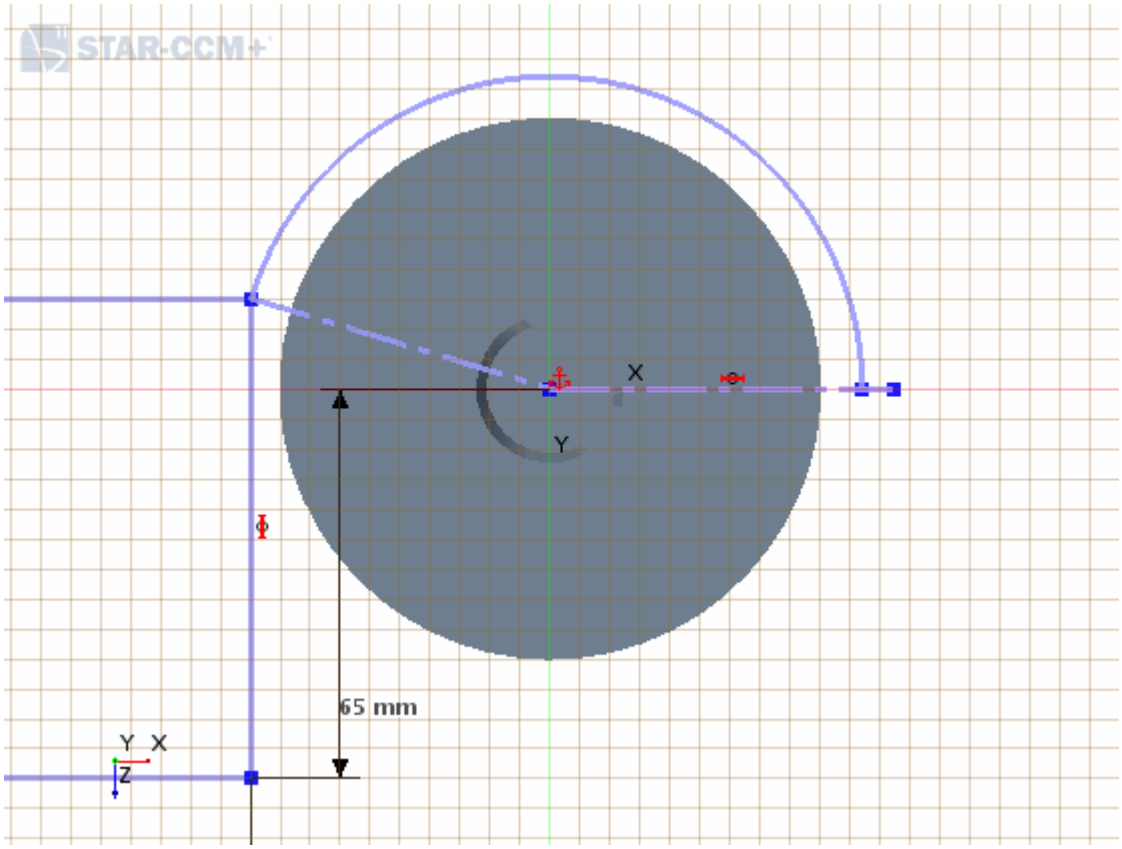


16. Using the <`Ctrl`> key, multi-select the circle and the two construction lines, in that order. Right-click one of the lines and select **Split for Trim**.

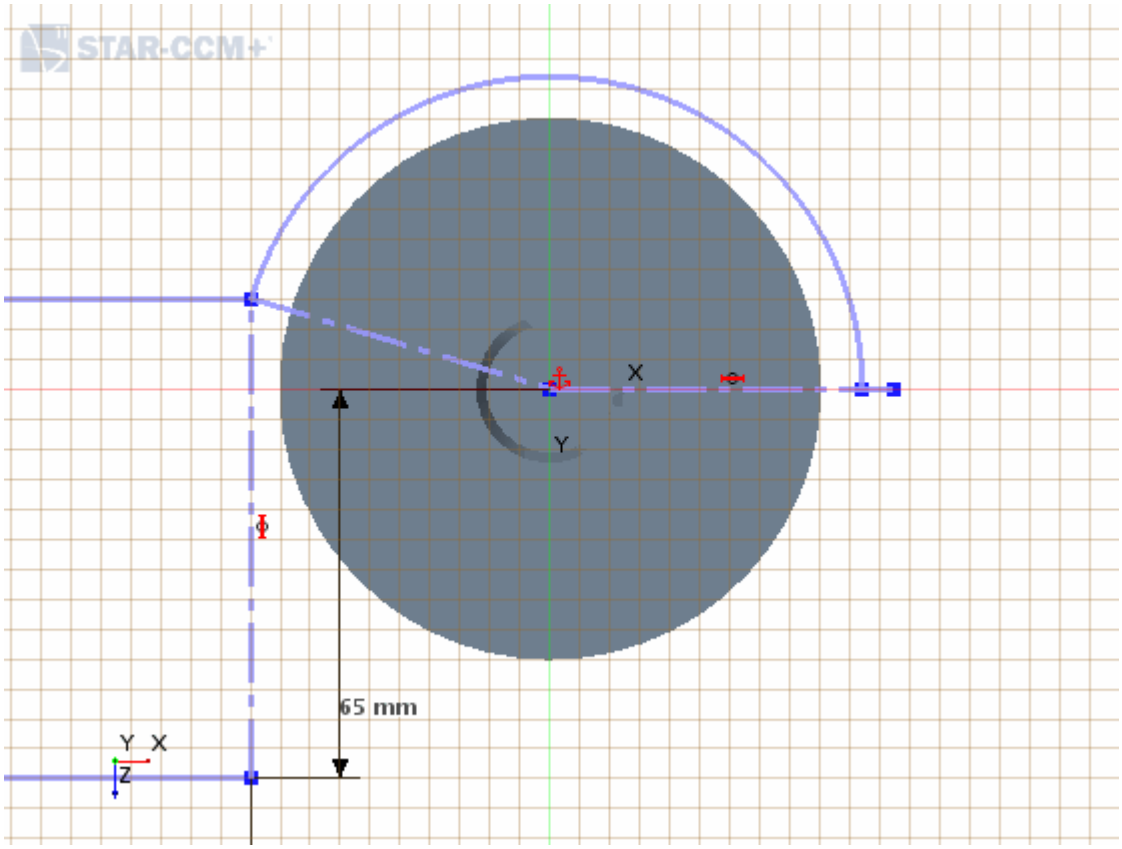
    The order in which you select the sketch entities is important and defines what entity is cut. The entity that you select first is the target. All subsequent selections are the tools that cut the target.

    The radius dimension for the circle is consumed by this action.

17. Select the bottom part of the circle and delete it using the <`Delete`> key.

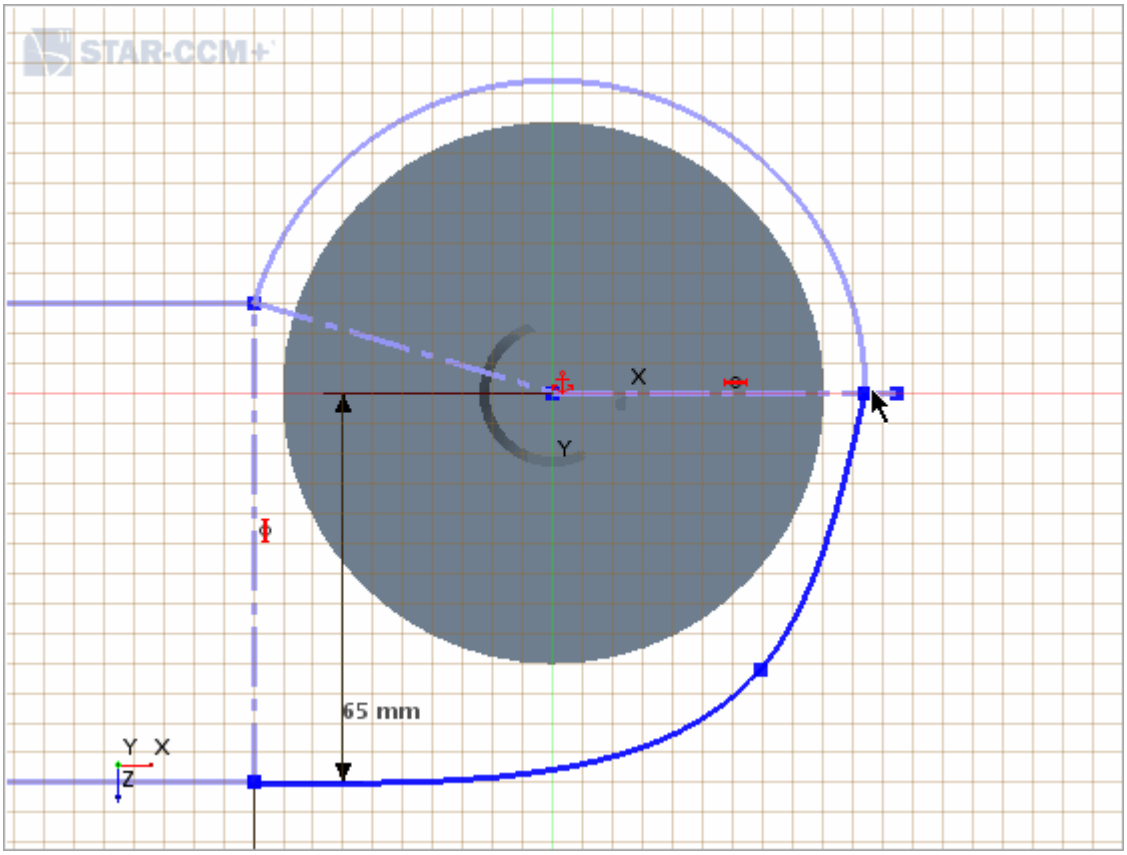


18. Select the right, vertical, edge of the rectangle and set this as a construction line.

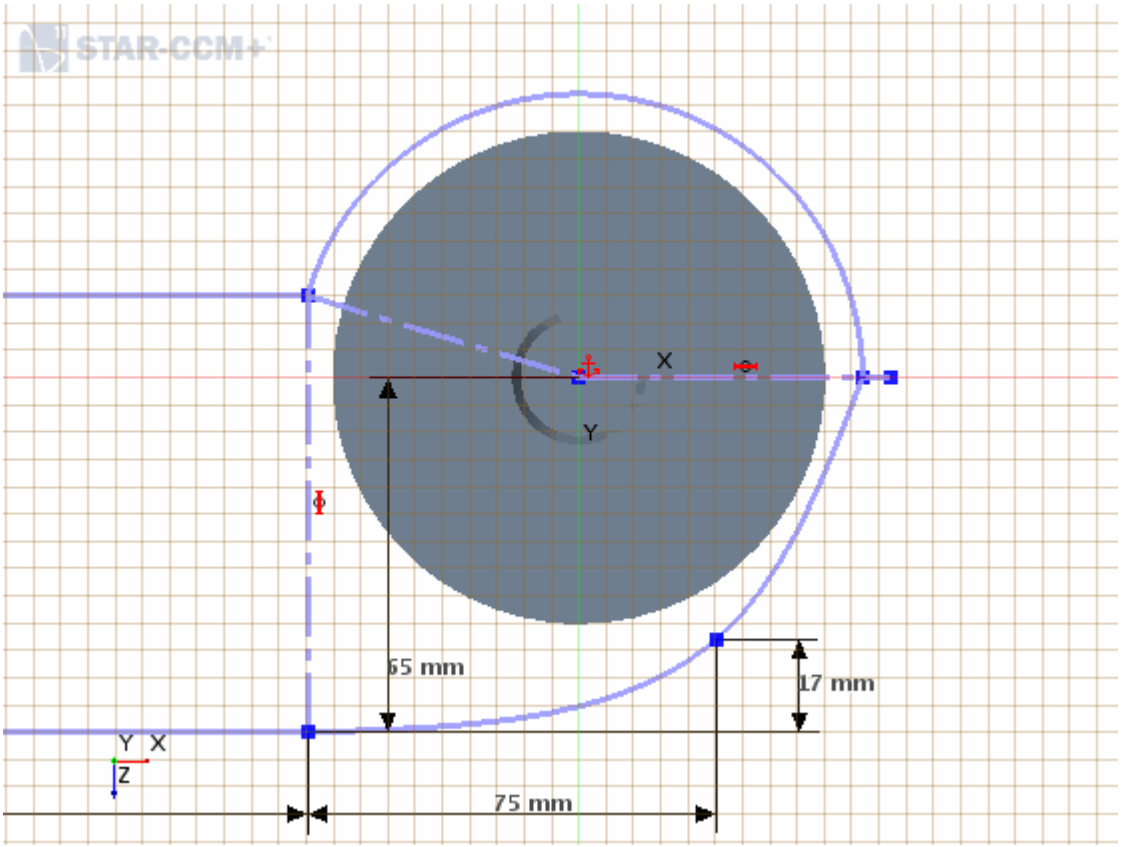


19. Draw a spline that links up the bottom edge of the rectangle to the right edge of the circle:

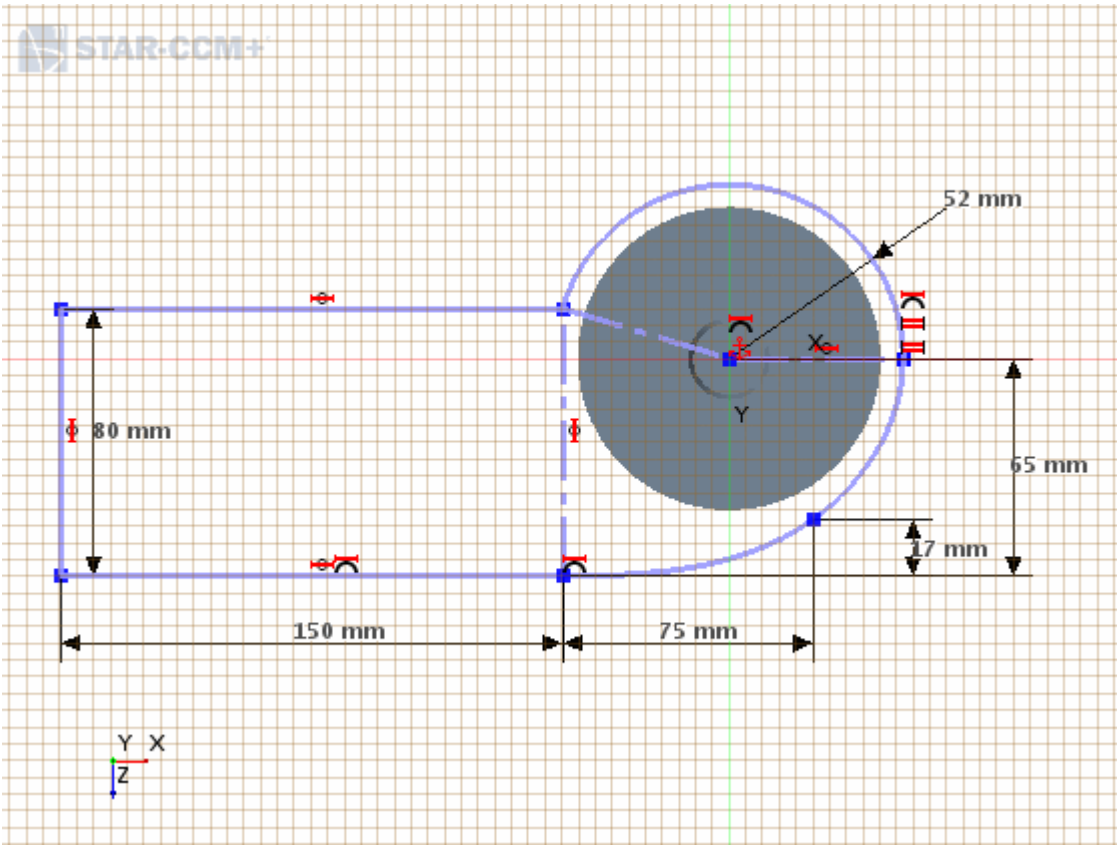
    a) Click (**Create Spline**) and draw a spline having three points from the bottom-right corner of the rectangle to the right edge of the circle. Place the second point to the lower-right of the circle as shown.



b) Define the spline by setting a vertical distance of `17 mm` and a horizontal distance of `75 mm` between the bottom-right edge of the rectangle and the middle spline point.



20. Make the spline tangent to the circle and to the bottom edge of the rectangle:
    a) Select the spline and the circle.
    b) Right-click a selection and choose **Apply Tangent Constraint**.
    c) Do the same between the bottom edge of the rectangle and the spline.
21. As a result of some actions, the sketch may move or change. While setting the tangent constraint, the split circle detached from the horizontal construction line. Make some final adjustments to the sketch:
    a) Select the right point on the split circle and the horizontal construction line and add a **Coincidence Constraint**.
    b) Add a radius dimension of `52 mm` to the split circle.



22. Click **OK** to complete the sketch.
23. Rename **Features** > **Sketch 1** to `Enclosure Sketch`.
24. In the *3D-CAD View* scene, right-click the sketch plane and select **Hide**.
25. Save the simulation.

Basic 3D-CAD: Creating the Enclosure Solid Shell

In cases where you create the wetted volume first, STAR-CCM+ allows you to surround the volume with a solid shell of specified thickness. This additional solid is then available for simulations that model phenomena in both solid and fluid materials.

Create the solid body for the graphics card enclosure and thereafter expose the fan within the enclosure.

1. Launch STAR-CCM+ and load **foundationTutorial_4.sim**.

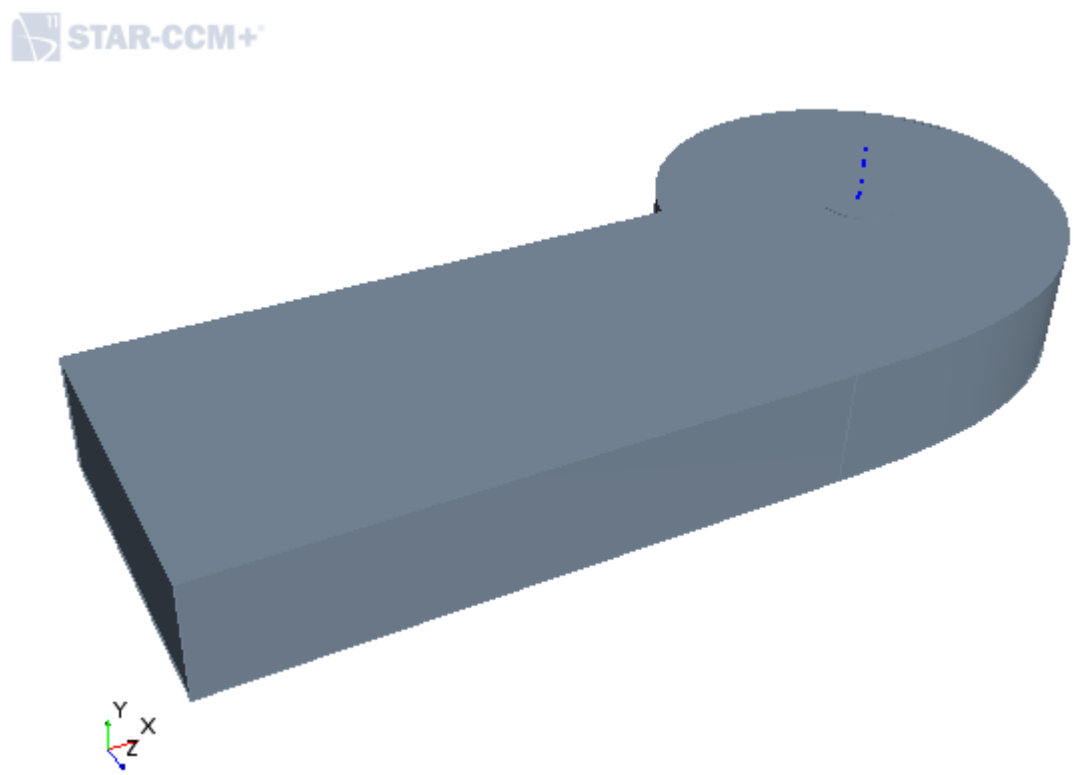
   You can either use the sim file that you saved from the previous tutorial, or load the sim file provided in `[INSTALL_DIR]/doc/startutorialsdata/foundationTutorials/data`.
2. Save the simulation as `foundationTutorial_5.sim`.
3. Resume the 3D-CAD model.
4. Extrude the enclosure sketch to form a solid body:
   a) Right-click **Enclosure Sketch** and select **Extrude**.
   b) In the *Extrude* panel, set *Direction Type* to **Reverse**.

      This option reverses the direction of the extrusion so that the solid encloses the fan.

c) Set *Method* to **Up to Face** and select the top face on the graphics card fan hub.
The extrusion extends up to the selected face.

d) To keep the fan separate from the surrounding body, set *Body Interaction* to **None**.

e) Click **OK**.



f) Rename **Features** > **Extrude 1** to `Enclosure Extrusion`.

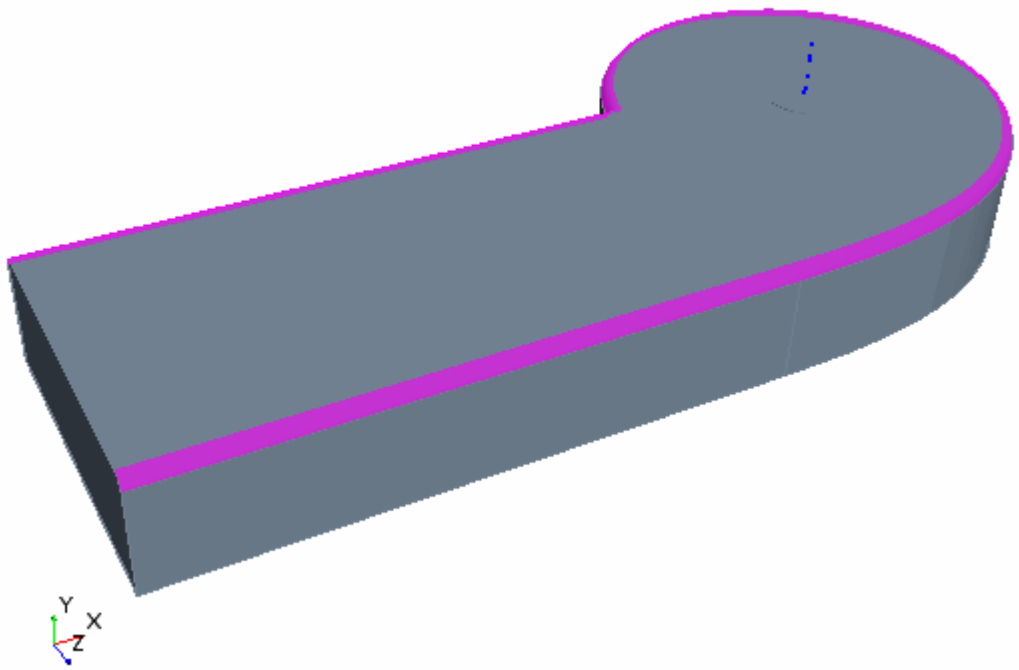
5. Add a fillet to the top edge of the enclosure:

a) In the *3D-CAD View*, multi-select the side and circular edges around the top of the enclosure.

b) Right-click the selection and choose **4 Edges** > **Create Fillet**.

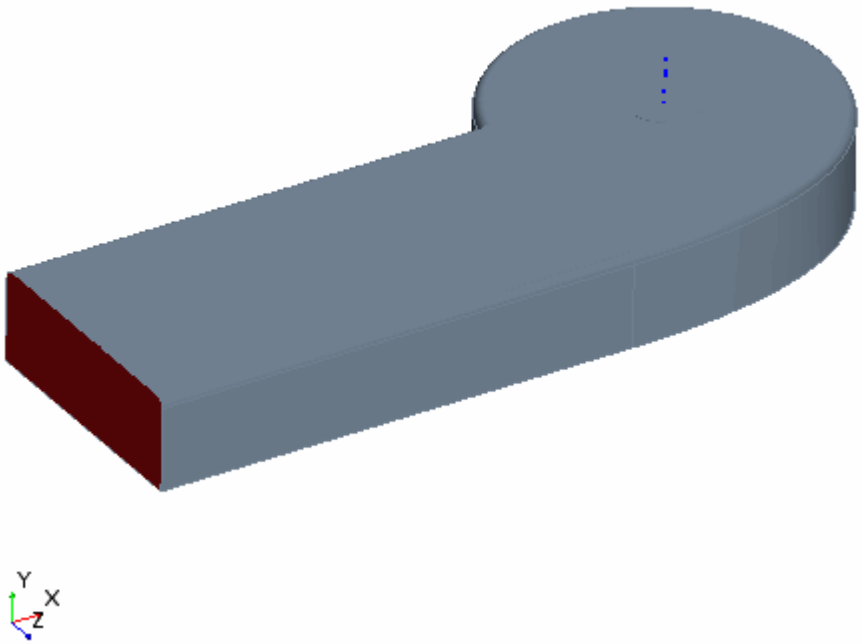
c) In the *Fillet* panel, set *Radius* to **3 mm**.

d) Click **OK**.



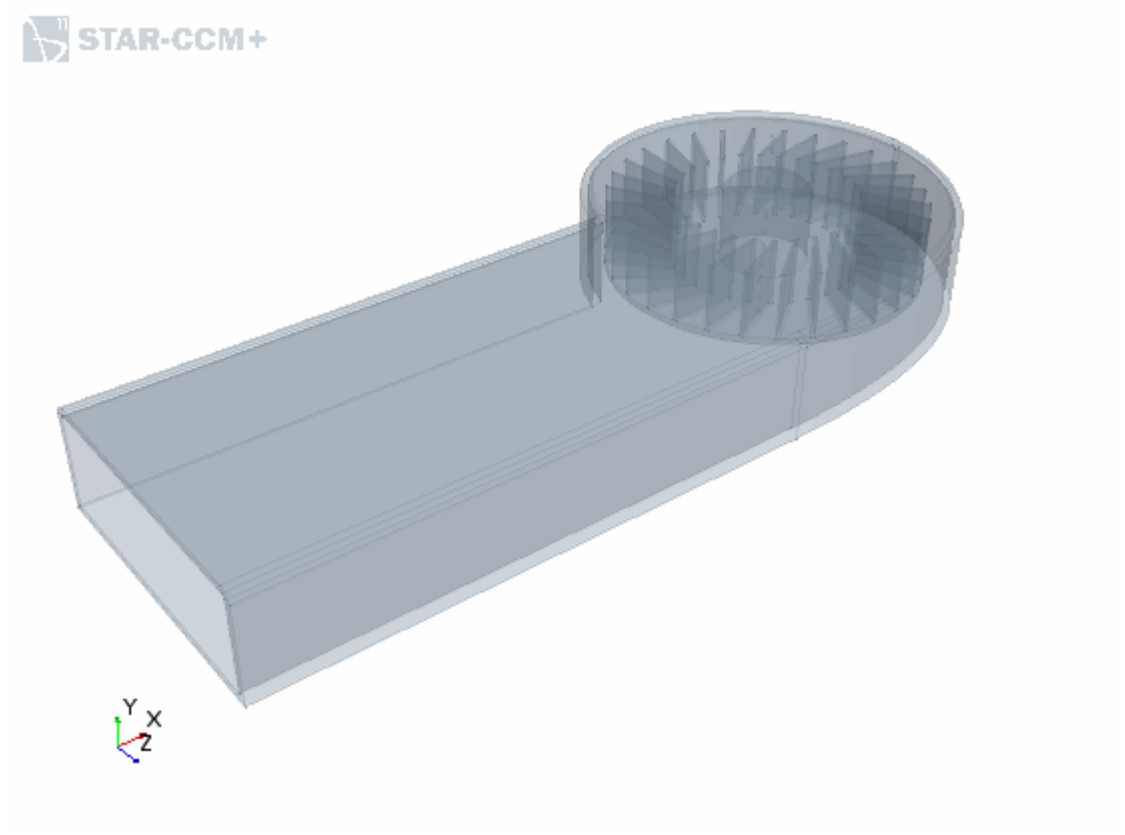
e) Rename **Features** > **Fillet** to `Enclosure Fillet`.

6. Shell the enclosure:
   a) In the *3D-CAD View*, right-click the enclosure body and select **Body** > **Shell**.
   b) In the *Shell* panel, set *Thickness* to **2 mm**.
   c) Click inside the *Faces to Remove* group box to activate face selection.
   d) In the *3D-CAD View* scene, select the face that corresponds to the exhaust of the enclosure.



This face is removed from the shelling operation so that the enclosure has an opening.

e) Click **OK**.



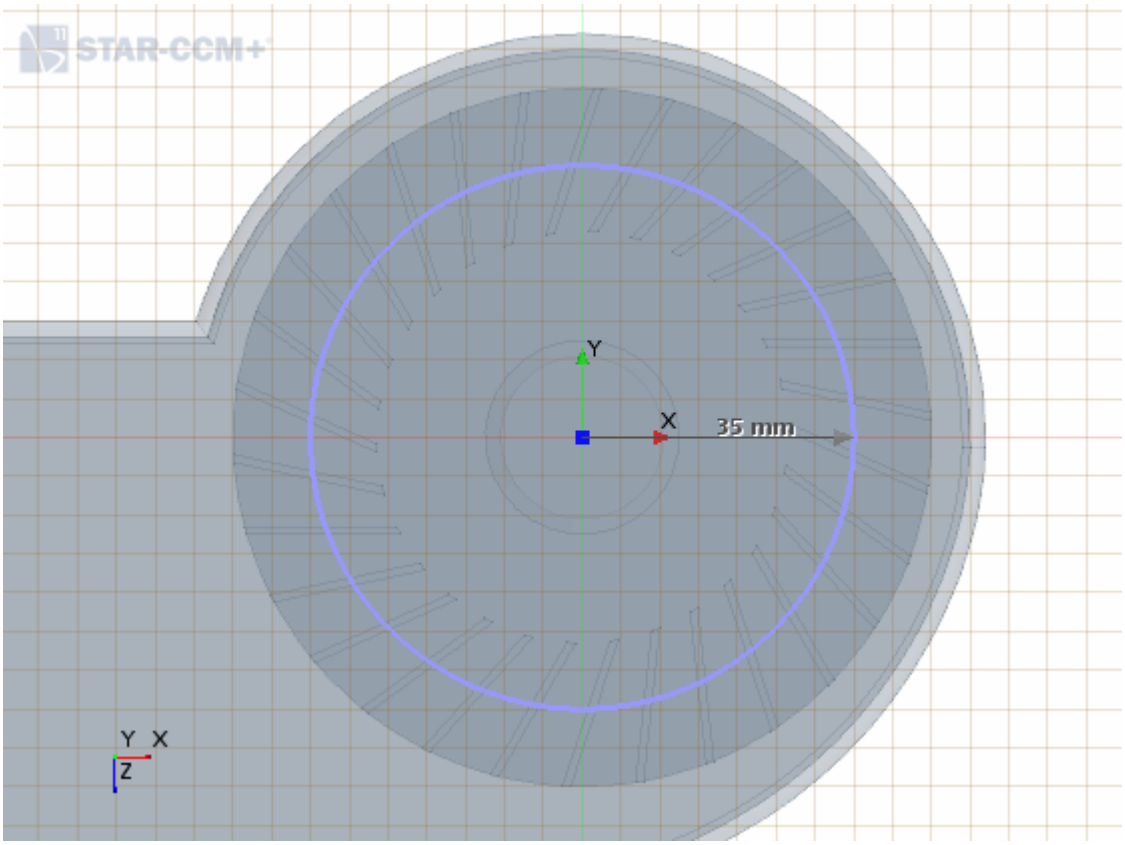
f) Rename **Features** > **Shell 1** to `Enclosure Shell`.

7. Draw a sketch for the fan opening:

   a) Right-click the top surface of the enclosure and select **Face** > **Create Sketch** > **On Face**.

   b) Draw a circle about the fan axis that has a radius of `35 mm`.

   Activate scene transparency so that you can see the fan.



c) Click **OK**.

d) Rename **Features** > **Sketch 1** to `Fan Opening Outline`.

8. Cut out the opening for the fan:

a) Right-click **Fan Opening Outline** and select **Extrude Cut**.

b) In the *Extrude Cut* panel, set *Method* to **Through Body First Segment**.

As the enclosure is a shell, this option cuts through the top layer of the shell, but not through the bottom layer.

c) Click the *Through Body* field and select the enclosure body in the *3D-CAD View* scene.

d) Set *Bodies to Cut* to **Selected**.

By default the operation cuts all bodies in its path. Here, you only want to cut the enclosure, not the fan.

e) Click the *Cut Bodies* group box, and select the enclosure body in the *3D-CAD View* scene.

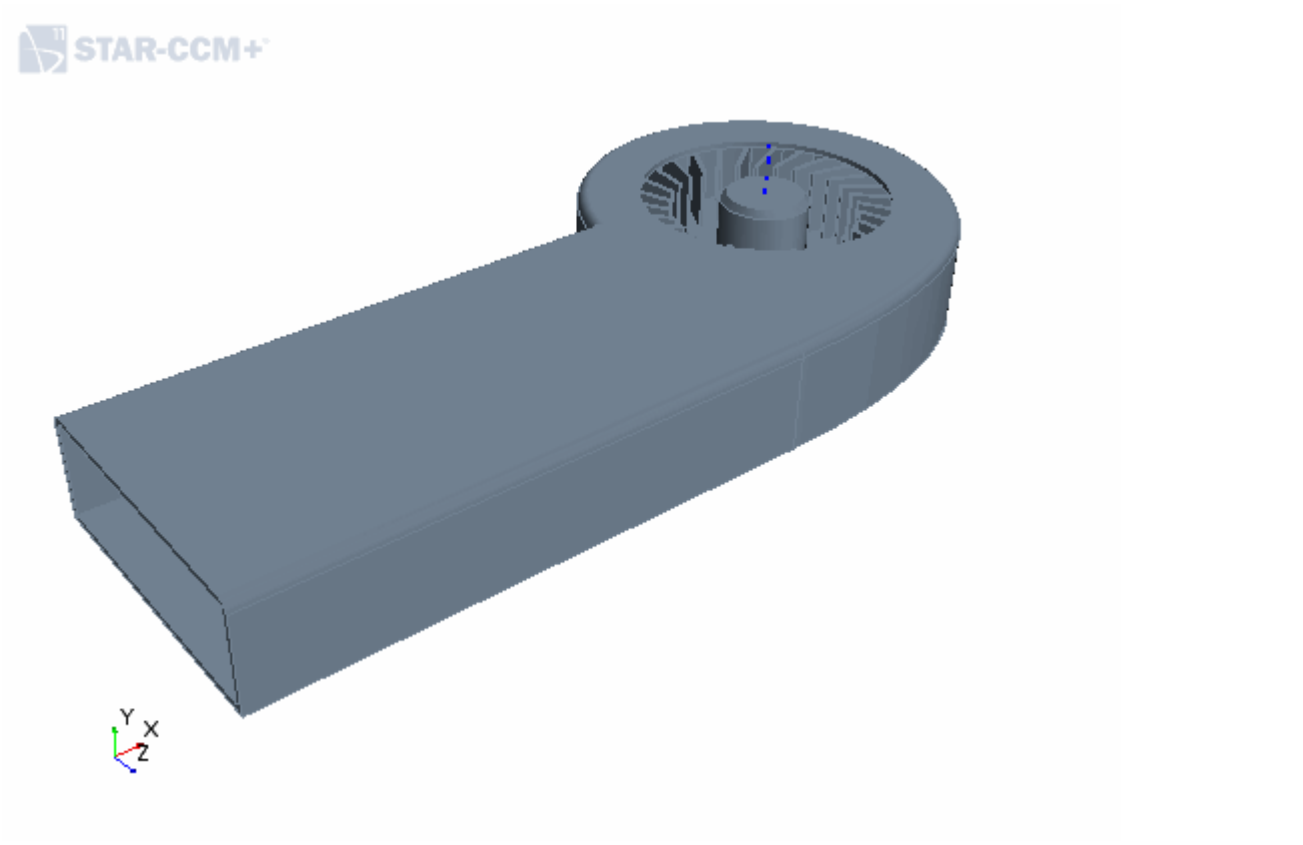
f) Click **OK**.

g) Rename **Features** > **Extrude Cut** to `Fan Opening`.

9. Expand the **Bodies** node and rename the new body to `Enclosure`.

10. Save the simulation.

The completed enclosure is displayed below.



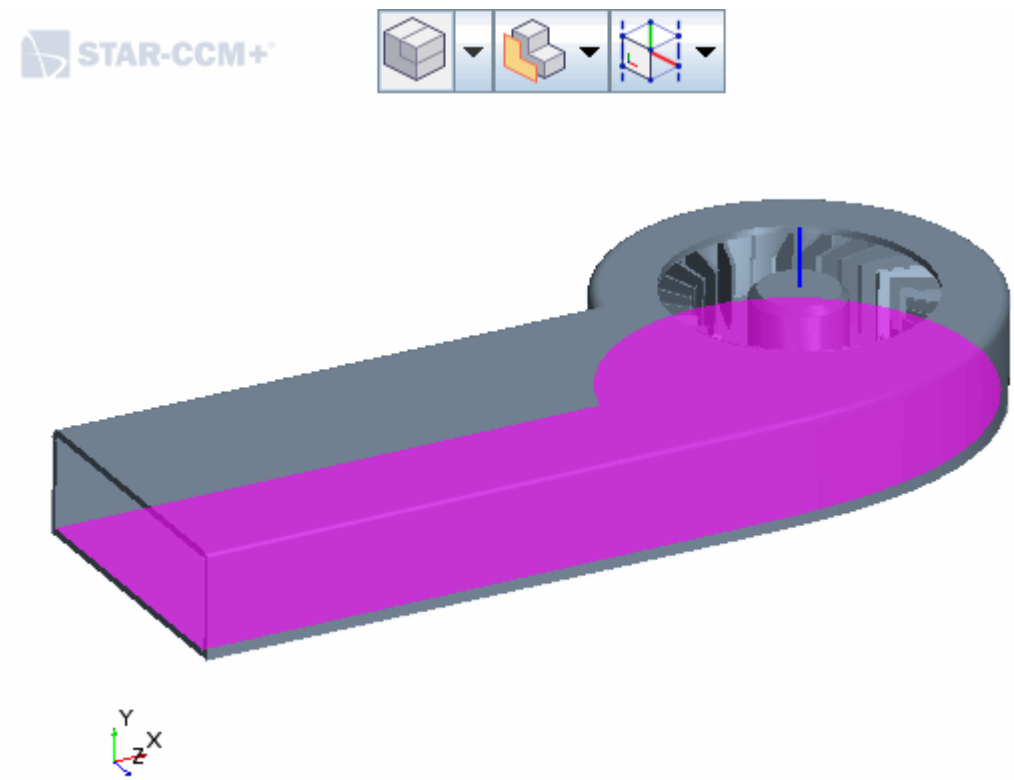
Basic 3D-CAD: Define Fluid Volume for Rotating Fan and Subtract the Blades

Modeling anything with a moving part, such as the graphics card fan, requires at least two simulation regions. One region rotates to model the motion of the fan, while the other region remains stationary to represent the remaining air. Here, you create the rotating region in 3D-CAD.

To create the rotating region:

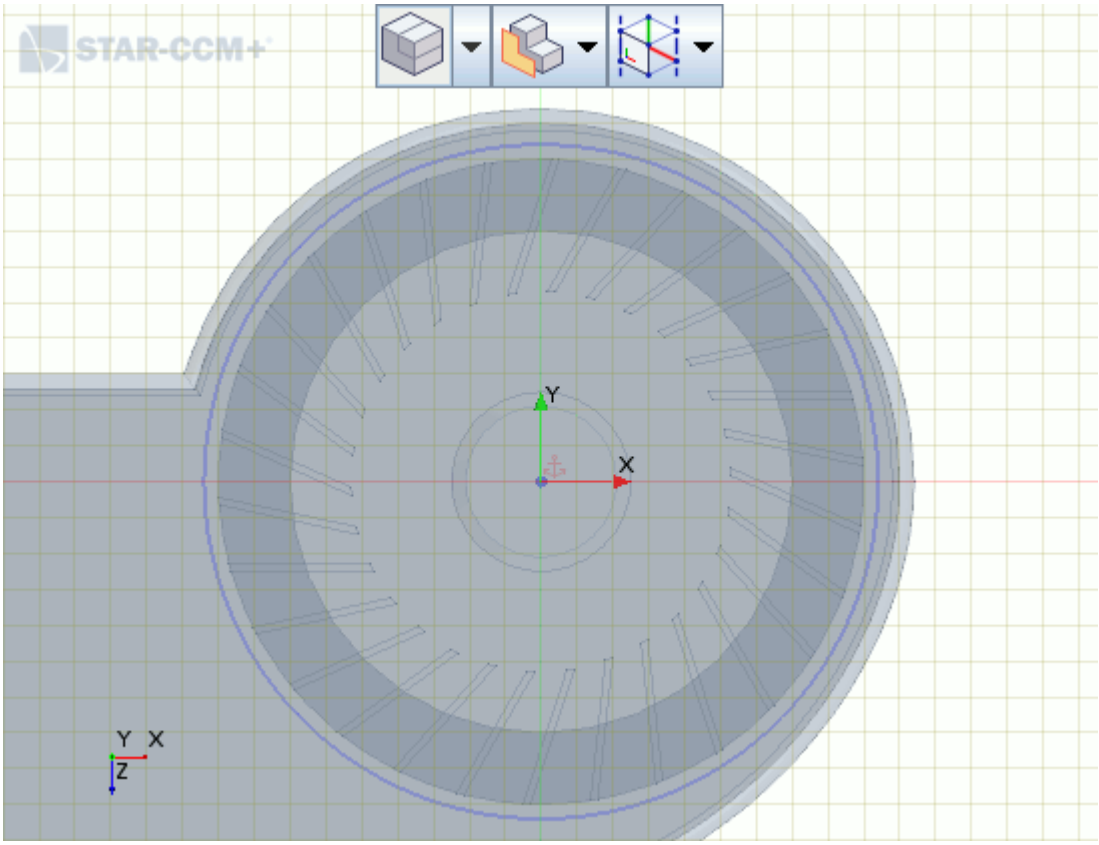
1. Launch STAR-CCM+ and load **foundationTutorial_5.sim**.

   You can either use the sim file that you saved from the previous tutorial, or load the sim file provided in `[INSTALL_DIR]/doc/startutorialsdata/foundationTutorials/data`.
2. Save the simulation as `foundationTutorial_6.sim`.
3. Resume the 3D-CAD model.
4. Draw a sketch around the fan:
   a) Select the bottom face in the cavity of the enclosure, as shown below, and select **Face** > **Create Sketch** > **On Face**.



b) Make the view normal to the sketch plane and make the geometry transparent.

c) Draw a circle with a radius of 47 mm and centered around the origin.



d) Click **OK** to complete the sketch.

e) Rename **Features** > **Sketch 1** to `Rotating Region Outline`.

5. Create the body for the rotating region:

a) Right-click **Rotating Region Outline** and select **Extrude**.

b) Set *Distance* to `35 mm`.

c) Set *Body Interaction* to **None** to create a separate body from this operation.

d) Click **OK**.

e) Rename **Features** > **Extrude 1** to `Rotating Region`.

f) Expand the **Bodies** node and rename the new body to `Rotating Region`.

6. Create a copy of the rotating region to subtract from the static region in the next section:

a) Right-click **Bodies** > **Rotating Region** and select **Duplicate**.

b) In the *Duplicate Bodies* panel, click **OK**.

c) Rename **Bodies** > **Rotating Region 2** to `Disc`.

d) Rename **Features** > **DuplicateBody 1** to `Duplicate Rotating Region`.

7. Subtract the fan and enclosure from the rotating region:

a) Within **Bodies**, multi-select the **Enclosure**, **Fan** and **Rotating Region** nodes.

b) Right-click on a selection and select **Boolean** > **Subract**.

c) In the *Subtract* panel, make sure that *Target Bodies* is set to **Rotating Region** and *Tool Bodies* is set to **Enclosure** and **Fan**.

To change the selections, click inside the relevant group box and select the desired body from the *3D-CAD View* scene or use the 3D-CAD feature tree. Target bodies are displayed in green while tool bodies are displayed in red.

d) Activate **Keep Tool Bodies** to keep the tool bodies after performing the operation.

By default, all tool bodies are absorbed by the operation. If you want to keep the bodies for visualization or simulation purposes, activate this option.

e) Click **OK** to perform the subtraction.

f) Rename **Features** > **SubtractBodies 1** to `Subtract Fan and Enclosure`.

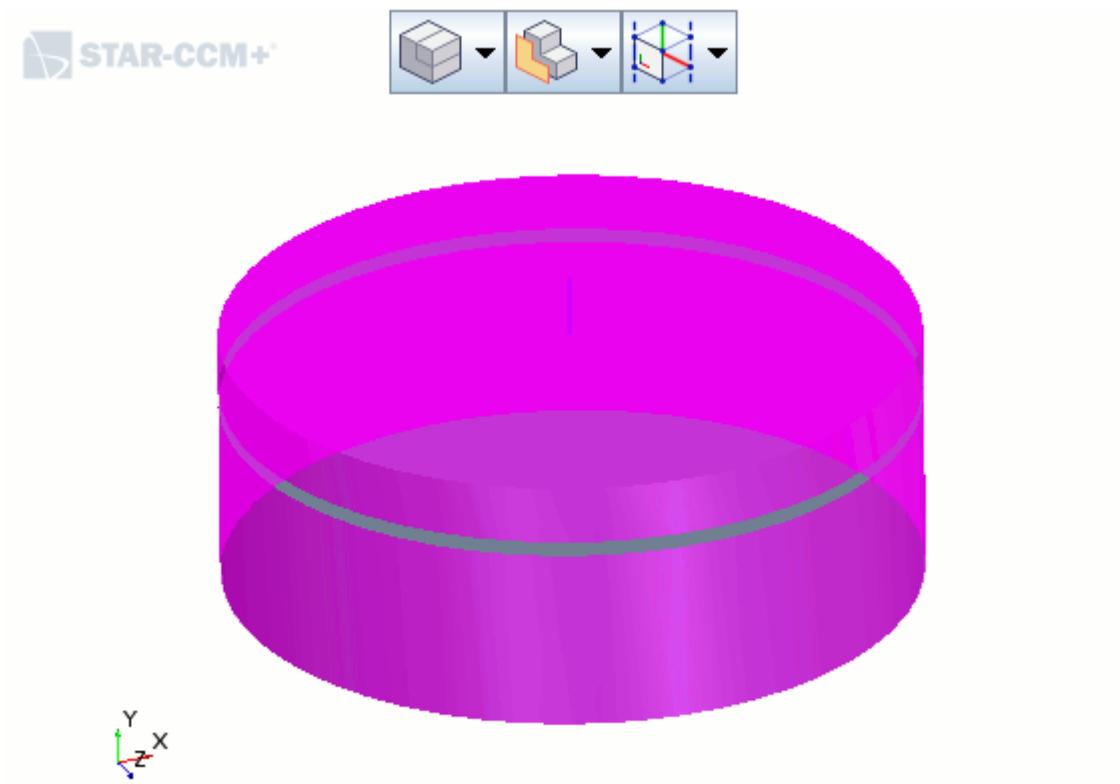
8. Rename the outer faces of the rotating region body:

a) In the 3D-CAD feature tree, expand the **Bodies** node.

b) Right-click the **Rotating Region** body and select **Show Only**.

All bodies other than the selection are hidden in the *3D-CAD View* scene.

c) In the *3D-CAD View* scene, select the large top and side faces in the scene. Do not select the faces in the narrow gap or the bottom face.



d) Right-click a selection and choose **3 Faces** > **Rename**.

e) In the *Rename* dialog, enter `Interface` and click **OK**.

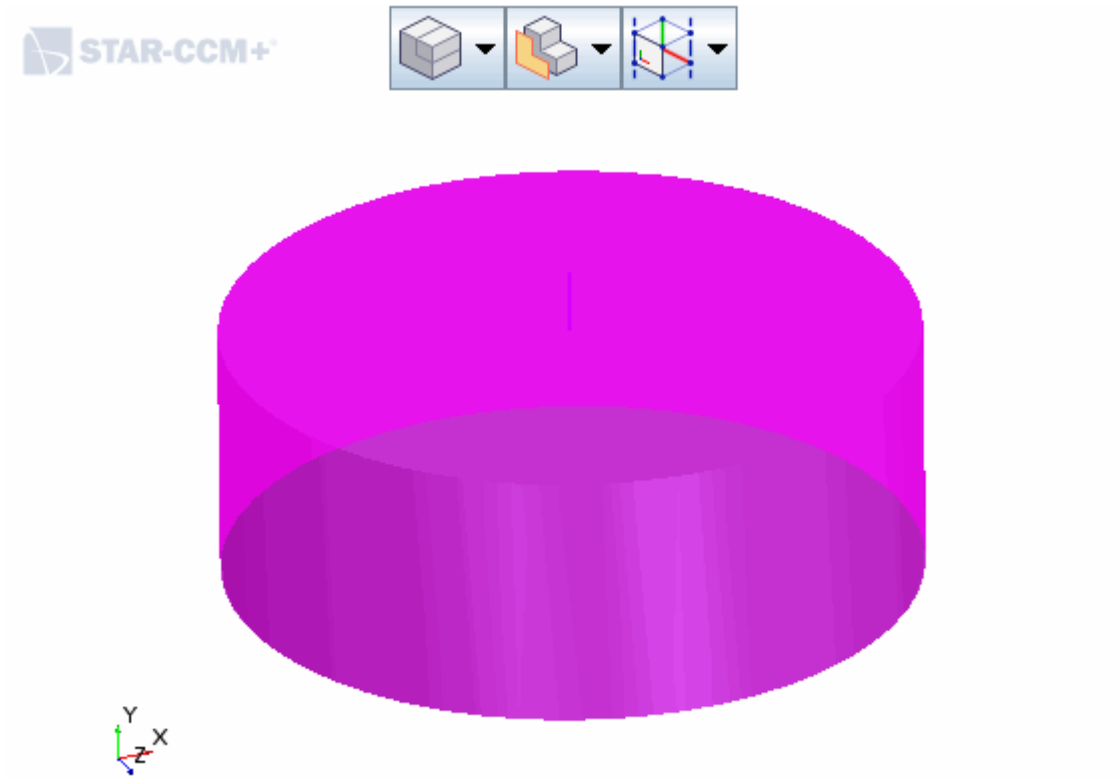
The selected faces are renamed. These faces are listed under the **Bodies** > **Rotating Region** > **Named Faces** node. From here, you can rename or delete the custom name if you desire.

f) In the **Bodies** > **Rotating Region** > **Named Faces** node, rename the **Default** node to `Fan`.

9. Rename the outer faces of the disc body:

a) In the 3D-CAD feature tree, right-click the **Disc** body and select **Show Only**.

b) In the *3D-CAD View* scene, select the top and side faces in the scene. Do not select the bottom face.



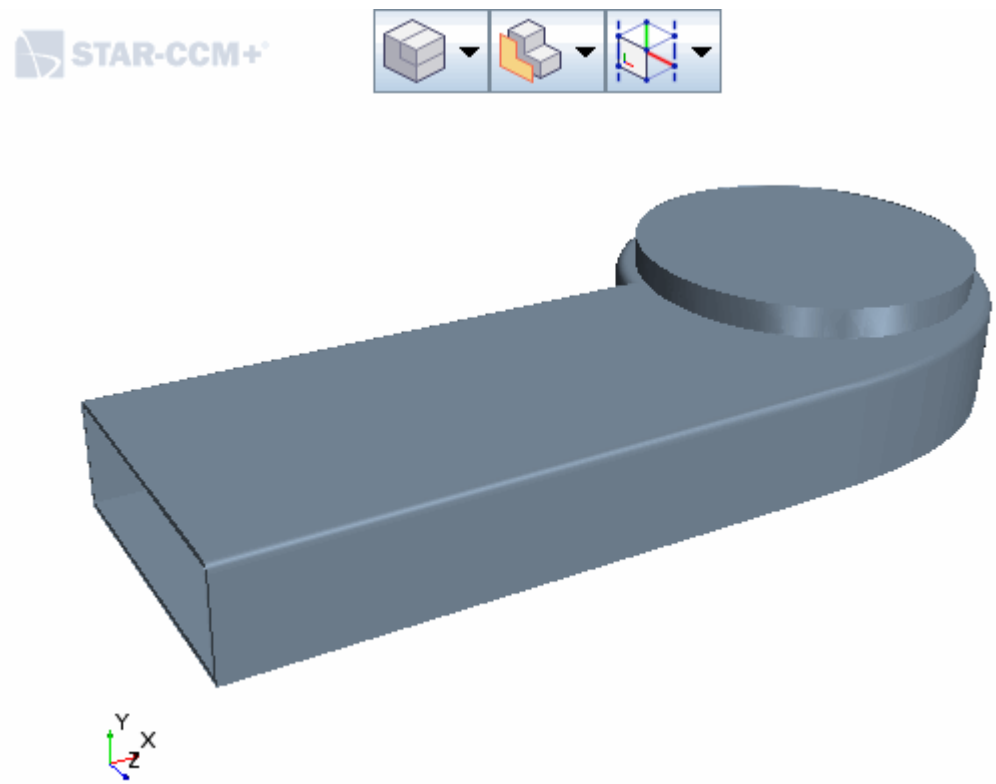
c) Right-click a selection and choose **2 Faces** > **Rename**.

d) In the *Rename* dialog, enter `Interface` and click **OK**.

The selected faces are renamed. These faces are listed under the **Bodies** > **Disc** > **Named Faces** node.

e) In the **Bodies** > **Rotating Region** > **Named Faces** node, rename the **Default** node to `Base`.

10. Right-click a blank space in the *3D-CAD View* scene and select **Restore All Hidden...** > **Bodies**.



11. Save the simulation.

Geometry Parts

Contents:

Geometry Parts: Creating the Main Fluid Volume and Subtracting the Enclosure and Fan

Geometry Parts: Creating the Main Fluid Volume and Subtracting the Enclosure and Fan

Although you can create the main fluid volume using 3D-CAD, STAR-CCM+ provides additional tools at the geometry parts level with which you can achieve the same goal. Using the part-level tools helps you differentiate between detailed and bulk geometry creation.

Here, you first create parts from the 3D-CAD model before creating a part shape that defines the outer fluid volume:

1. Launch STAR-CCM+ and load **foundationTutorial_6.sim**.

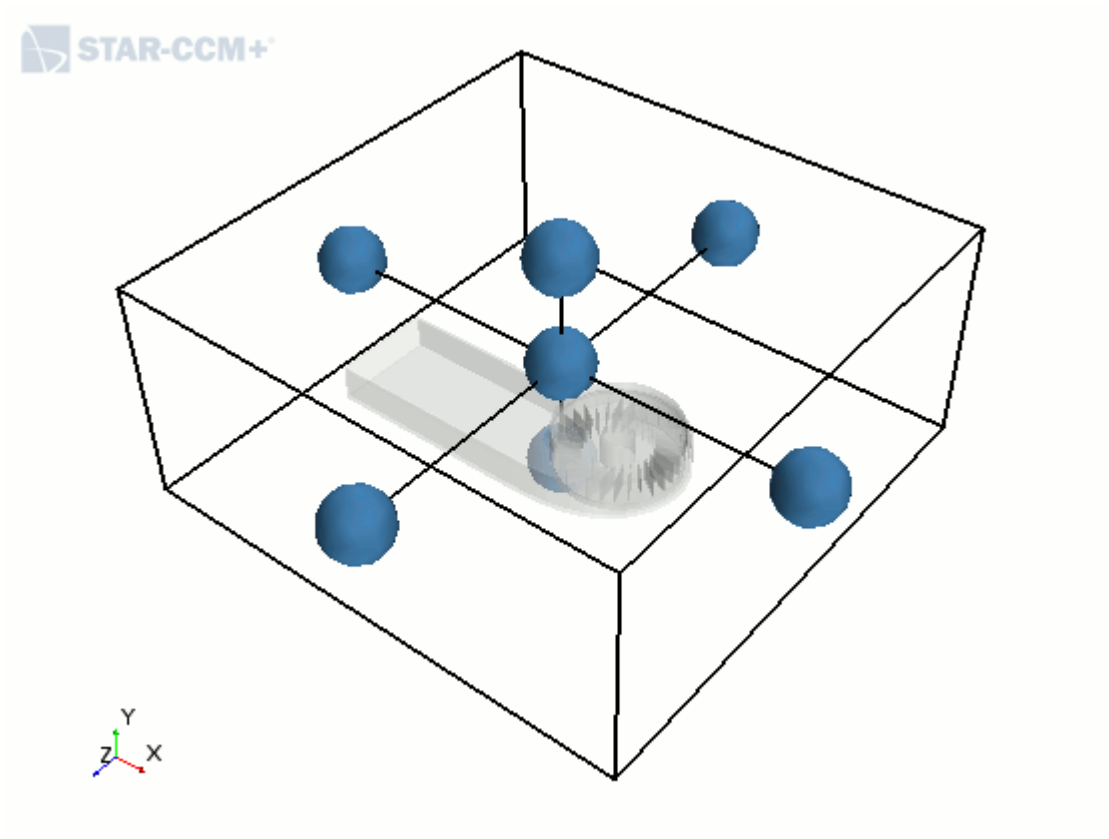
   You can either use the sim file that you saved from the previous tutorial, or load the sim file provided in `[INSTALL_DIR]/doc/startutorialsdata/foundationTutorials/data`.
2. Save the simulation as `foundationTutorial_7.sim`.
3. Create geometry parts from the 3D-CAD model:
   a) Expand the **Geometry** > **3D-CAD Models** node.

      The 3D-CAD model of your geometry is stored here.
   b) Right-click **3D-CAD Model 1** and select **New Geometry Part**.
   c) In the *Part Creation Options* dialog, click **OK** to accept the default settings.

   The solid bodies from the 3D-CAD model are added as geometry parts.
4. Right-click the **Scenes** node and select **New Scene** > **Geometry**.

   A new geometry scene, **Geometry Scene 1**, is added to the **Scenes** node. This scene automatically displays all parts that are listed under the **Geometry** > **Parts** node.
5. Create a part shape for the surrounding fluid volume:
   a) Right-click the **Geometry** > **Parts** node and select **New Shape Part** > **Block**.
   b) In the *Create Block Part* panel, set the following parameters:

Parameter	Setting
Corner 1	`[-0.250, -0.025, -0.200] m`
Corner 2	`[0.150, 0.150, 0.200] m`



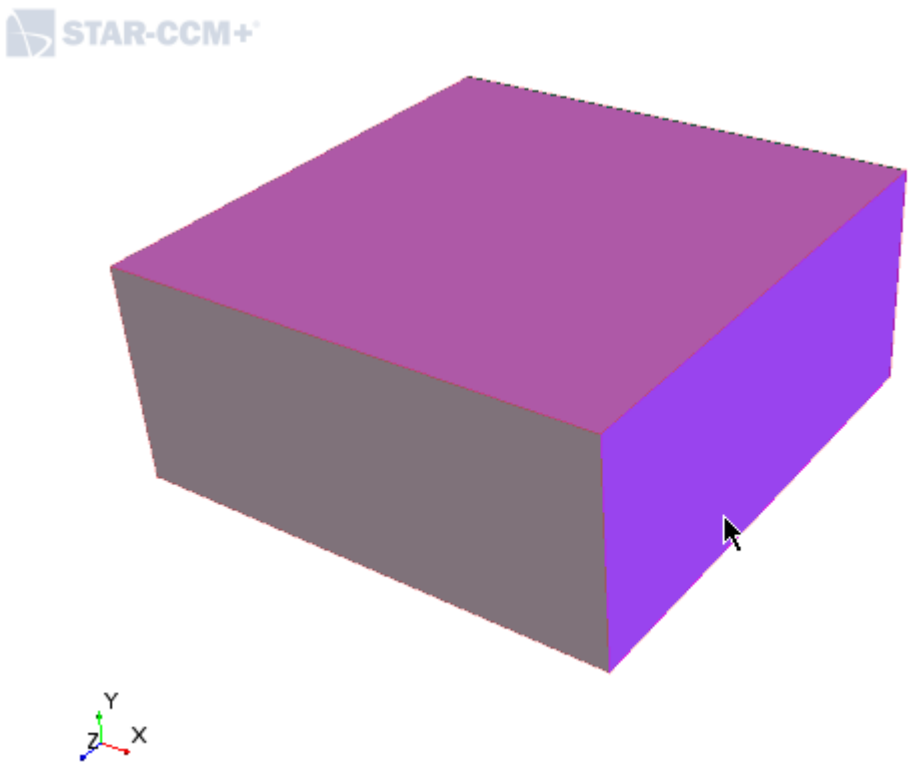
c) Click **Create** followed by **Close**.

A shape part, **Block**, is created and added to the **Parts** node.

6. Split the surfaces of the block in preparation for the extruder mesher that you will set up in a later tutorial:

a) Right-click the **Block** > **Surfaces** > **Block Surface** node and select **Split by Patch**.

b) In *Geometry Scene 1*, select the face that is upstream of the flow.

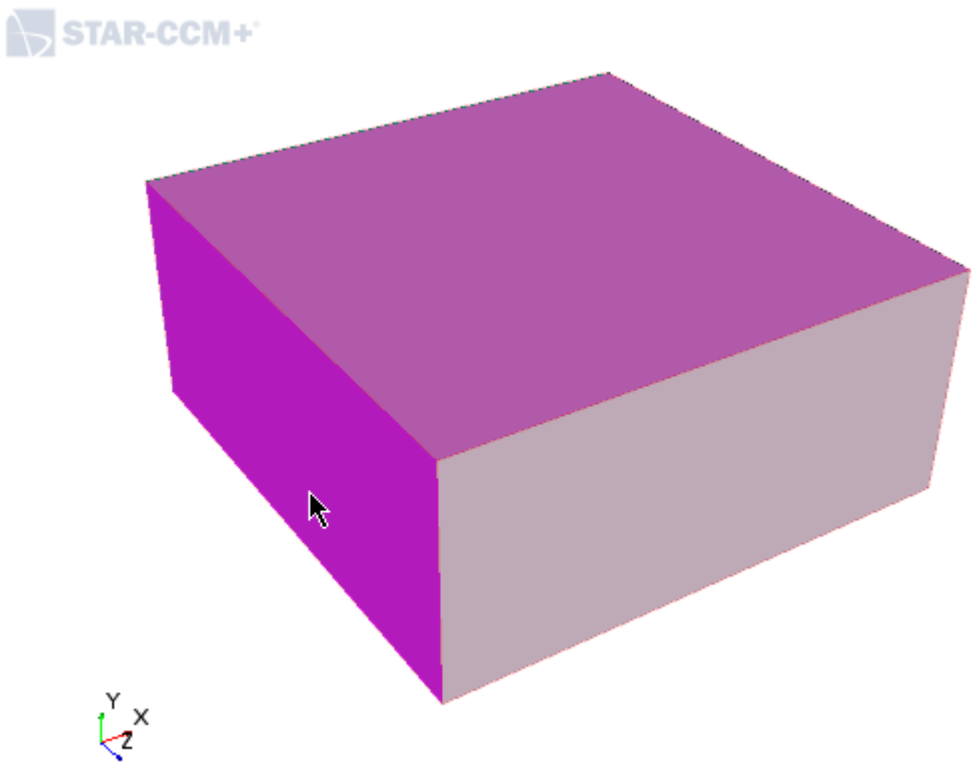


To help you identify the correct face to select, pay attention to the orientation of the direction axes in the scene. Select the face that lies on the plane with the largest X value.

c) In the *Part Surface Name* field, enter `Upstream` and click **Create**.

The selected patch is removed from the scene.

d) In the same scene, select the surface that is downstream of the flow.



To help you identify the correct face to select, pay attention to the orientation of the direction axes in the scene. Select the face that lies on the plane with the smallest X value.

e) In the *Part Surface Name* field, enter `Downstream` and click **Create**.

f) Click **Close**.

The selected patches are split and added to the **Geometry** > **Parts** > **Block** > **Surfaces** node.

7. Subtract the graphics card and disc from the static region:

a) Right-click the **Geometry** > **Operations** node and select **New** > **Boolean** > **Subtract**.

b) In the *Create Subtract Operation* dialog, set the following parameters:

Parameter	Setting
Input Parts	**Block**
	Disc
	Enclosure
Target Part	**Block**
Perform CAD Boolean	**Activated**

c) Click **OK**.

A subtract operation is added to the **Operations** node. A subtract placeholder for the operation is also added to the **Parts** node.

d) Rename the **Parts** > **Subtract** node to `Static Region`.

e) Rename the **Operations** > **Subtract** node to `Subtract Block`.

f) Right-click the **Operations** > **Subract Block** node and select **Execute**.

8. Modify the geometry scene to display only the parts that correspond to the simulation domain:

a) Click **Scene/Plot** above the simulation object tree.

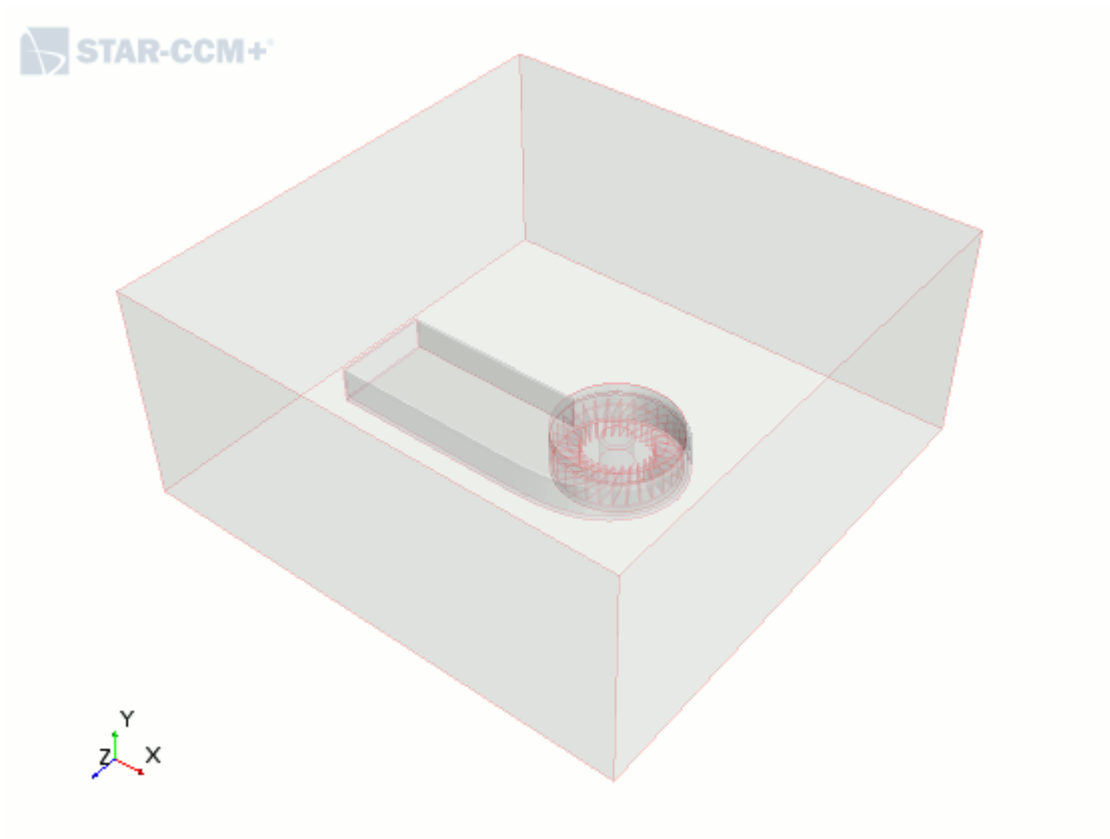
The object tree displays only the properties of the active scene, *Geometry Scene 1*.

b) Double-click the **Displayers** > **Geometry 1** > **Parts** node.

c) In the *Parts* dialog, right-click on a blank area of the dialog and select **Deselect All** to clear the selection.

d) In the same dialog, select the **Rotating Region** and **Static Region** and click **OK**.

e) Click (**Make Scene Transparent**) to get a better view of the static region.



9. To create a part contact between the static region and the rotating region, imprint these two parts together:

Part contacts define the connection between two part surfaces. The meshers aim to generate a conformal mesh at the contact, which then becomes an interface in the physics region.

a) Click **Simulation** above the simulation object tree.

b) Right-click **Geometry** > **Operations** and select **New** > **Boolean** > **Imprint**

c) In the *Create Imprint Operation* dialog, select **Rotating Region** and **Static Region**.

d) Activate the **Perform CAD Imprint** option.

e) Click **OK**.

An imprint operation is added to the **Operations** node.

f) Right-click the **Operations** > **Imprint** node and select **Execute**.

The two parts are imprinted together and a part contact is created between the two parts.

10. Save the simulation.

Mesh

Contents:

Mesh: Setting Up the Simulation Domain and Selecting the Meshers
Mesh: Checking the Mesh Quality
Mesh: Extending the Simulation Domain with the Mesh Extruder
Mesh: Refining the Mesh
Mesh: Rotating Region

Mesh: Setting Up the Simulation Domain and Selecting the Meshers

Prepare the regions for the simulation domain and select the meshers.

Begin by creating the regions:

1. Launch STAR-CCM+ and load **foundationTutorial_7.sim**.
2. Save the simulation as `foundationTutorial_8.sim`.
3. Create the regions from the geometry parts:
   a) If the geometry scene is not open, double-click the **Scenes** > **Geometry Scene 1** node.
   b) Expand the **Geometry** > **Parts** node.
   c) Select the **Rotating Region** and **Static Region** nodes.
   d) Right-click on a selection and choose **Assign Parts to Regions...**.
   e) In the *Assign Parts to Regions* dialog, select:
      1. **Create a Region for Each Part**
      2. **Create a Boundary for Each Part Surface**
      3. **Create Contact-mode Interfaces From Contacts**
      4. Click **Apply** then **Close**.

      The geometry in the scene changes to a dark gray color to show that parts displayed in the scene are assigned to regions. Parts that have no region association are displayed in a light gray color.
4. Create an automated mesh operation and select the mesh models:
   a) Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
   b) In the *Create Automated Mesh Operation* dialog:
      1. Select **Rotating Region** and **Static Regon** from the *Parts* list.
      2. Select the following meshers, in order:

Group	Mesher
Surface Meshers	**Surface Remesher**
Core Volume Meshers	**Polyhedral Mesher**
Optional Boundary Layer Meshers	**Prism Layer Mesher**

The Surface Remesher retriangulates the surface of the geometry and improves the overall quality of the initial surface. The Polyhedral Mesher generates polyhedral cells that are suitable for turbulent and swirling flow simulations. The Prism Layer Mesher generates a prism layer (orthogonal prismatic cells) next to wall surfaces or boundaries. Prism layer cells are important for resolving the boundary layer and the velocity gradient of the flow near surface.

c) Click **OK**.

An automated mesh operation, **Automated Mesh**, is added to the **Operations** node. The mesh models that you select are located inside this operation under the **Meshers** node.

5. Set up the mesh parameters:

a) Expand the **Operations** > **Automated Mesh** > **Default Controls** node.

The global mesh parameters are located under this node.

b) Select **Base Size** and set *Value* to `2 mm`.

The base size is the reference length for all reference size controls. Using the value that you specify in the base size, you can scale the mesh using relative size controls as a percentage of the base size, or set these controls independently using absolute values. Typically you set the base size to the scale of the object of interest, in this tutorial this is the fan blades in the graphics card.

c) Select the **Target Surface Size** node, select **Relative to Base** and set *Percentage of Base* to `1000` in the Properties Window.

The target surface size is set as a percentage of the base size and specifies the cell size that the meshers should aim for when generating the mesh. The target size can be smaller or larger than the base size, and you can use it to grow or shrink the cells in the mesh.

d) Select the **Minimum Surface Size** node, select **Relative to Base** and set *Percentage of Base* to `50` in the Properties Window.

The minimum surface size specifies the smallest element size that the mesher can generate. The thickness of the fan blade is 1 mm, while the base size is 2 mm. Setting the minimum surface size to 50% of the base size returns 1 mm and allows the mesh to capture the details of the fan blades.

e) Select the **Prism Layer Total Thickness** node, set *Size Type* to **Absolute** and set *Absolute Size* to `2 mm` in the Properties Window.

By default, all size controls are set as a percentage of the base size. By setting the size type to absolute, you make this property independent of the base size. Absolute values are useful when you want to have full control over the size of a particular setting.

f) Select the **Prism Layer Total Thickness** > **Absolute Size** node and set *Value* to `2 mm`.

The total thickness of the prism layer specifies the thickness of the prism layer, including all prism cells.

Boundaries in the simulation regions affect how the mesh is generated. More specifically, prism layers are generated next to wall boundaries and interfaces (the contacting surface between the regions). The far field walls of the simulation domain (the walls that are far away from the object of interest) are not important for this simulation; however, for the purpose of this tutorial, the size of the simulation domain is smaller than what is recommended for industrial cases. Ideally the far field surfaces in the simulation domain should be further away, as having these surfaces close to the geometry of interest will affect the flow inside the simulation domain. One way to reduce the effect of the walls on the overall flow inside the simulation domain is to set the outer surfaces to symmetry boundaries. Prism layers are not generated on symmetry boundaries.

6. Set the walls of the simulation domain to symmetry boundaries:

a) Expand the **Regions** > **Static Region** > **Boundaries** node.

b) Select the **Block.Block Surface**, **Block.Downstream**, and **Block.Upstream** nodes and set *Type* to **Symmetry Plane**.

The selected boundary surfaces change to a blue color in the geometry scene to indicate that they are symmetry boundaries.

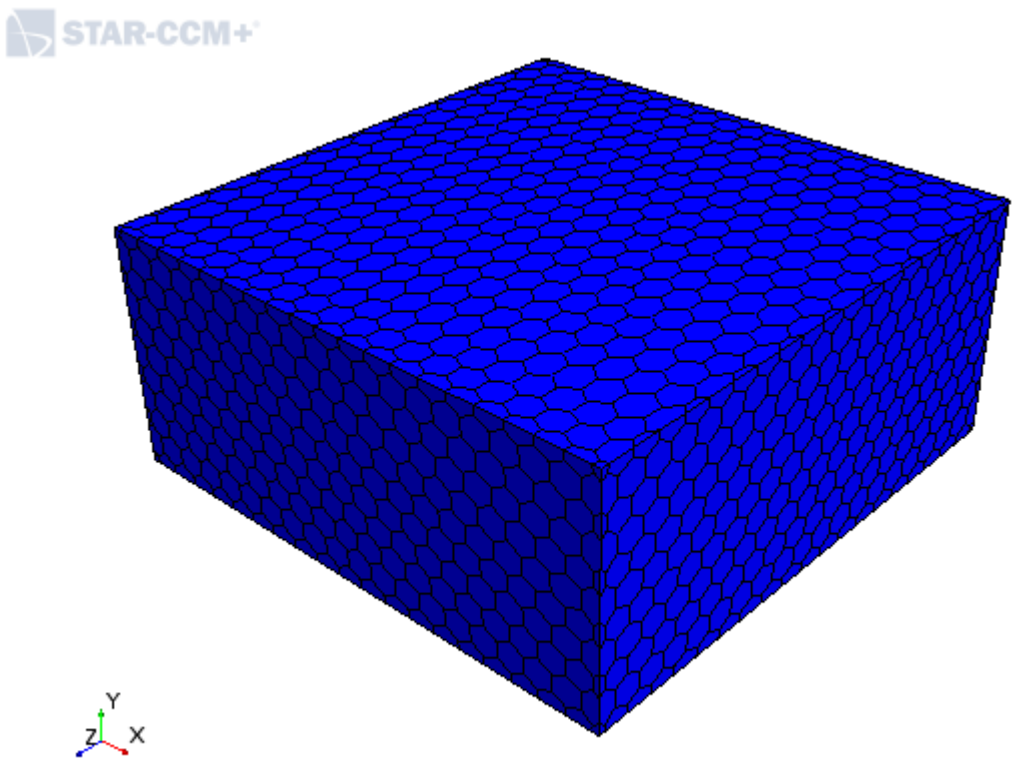
7. Save the simulation.

8. To generate the volume mesh, right-click the **Automated Mesh** node and select **Execute**.

The surface remesher retriangulates the initial geometry surface tessellation into more evenly sized triangles. The space for the prism layer is calculated and reserved, this is called the prism layer subsurface. The polyhedral mesher creates three-dimensional cells, called tetrahedra, from the retriangulated surface and converts them to polyhedra. The prism layer is then generated and fills the reserved space with prism layer cells.

9. Create a mesh scene to visualize the mesh:

a) Right-click **Scenes** and select **New Scene** > **Mesh**.



b) Rename the **Scenes** > **Mesh Scene 1** node to `Simulation Domain`.

c) Create another mesh scene and rename it to `Fan and Enclosure`.

d) Click **Scene/Plot** above the simulation object tree.

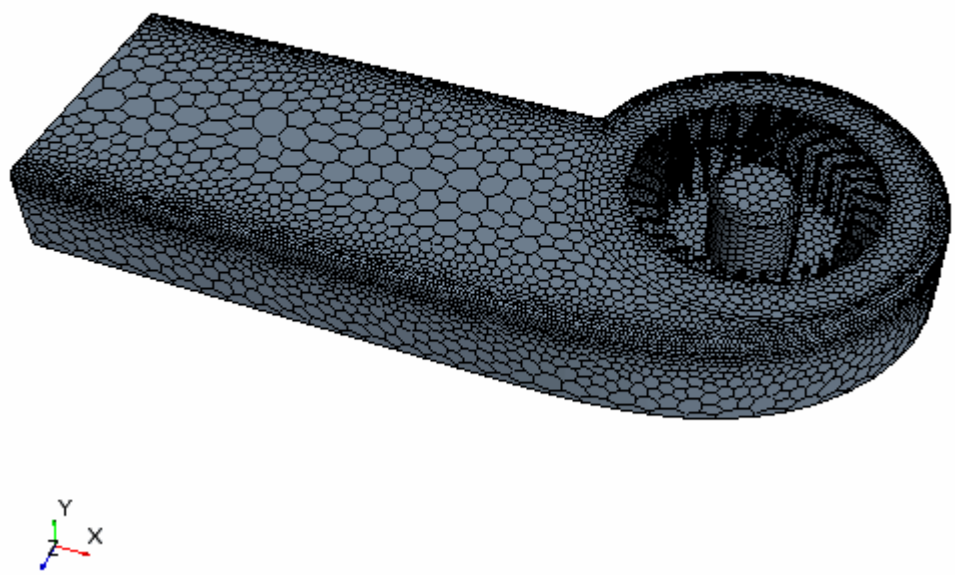
e) Double-click the **Displayers** > **Mesh 1** > **Parts** node.

f) In the *Parts* dialog, right-click on a blank area of the dialog and select **Deselect All** to clear the selection.

g) In the same dialog, expand the **Regions** node.

h) Select the **Rotating Region** > **Boundaries** > **Fan** and **Static Region** > **Boundaries** > **Enclosure.Default**.

i) Click **OK**.



The cell size automatically adjusts, within the specified parameters, to capture the shape of geometry.

10. Save the simulation.

Mesh: Checking the Mesh Quality

Use the field function and tools available in STAR-CCM+ to check the quality of the generated mesh.

The volume mesh generated in the previous tutorial appears fine at first glance, however, to check the mesh quality properly, use the tools available in STAR-CCM+.

To check the quality of the mesh:

1. Launch STAR-CCM+ and load **foundationTutorial_8.sim**.

   You can either use the sim file that you saved from the previous tutorial, or load the sim file provided in `[INSTALL_DIR]/doc/startutorialsdata/foundationTutorials/data`.

2. Save the simulation as `foundationTutorial_9.sim`.

3. Run the included Java macro to check the quality of the mesh:

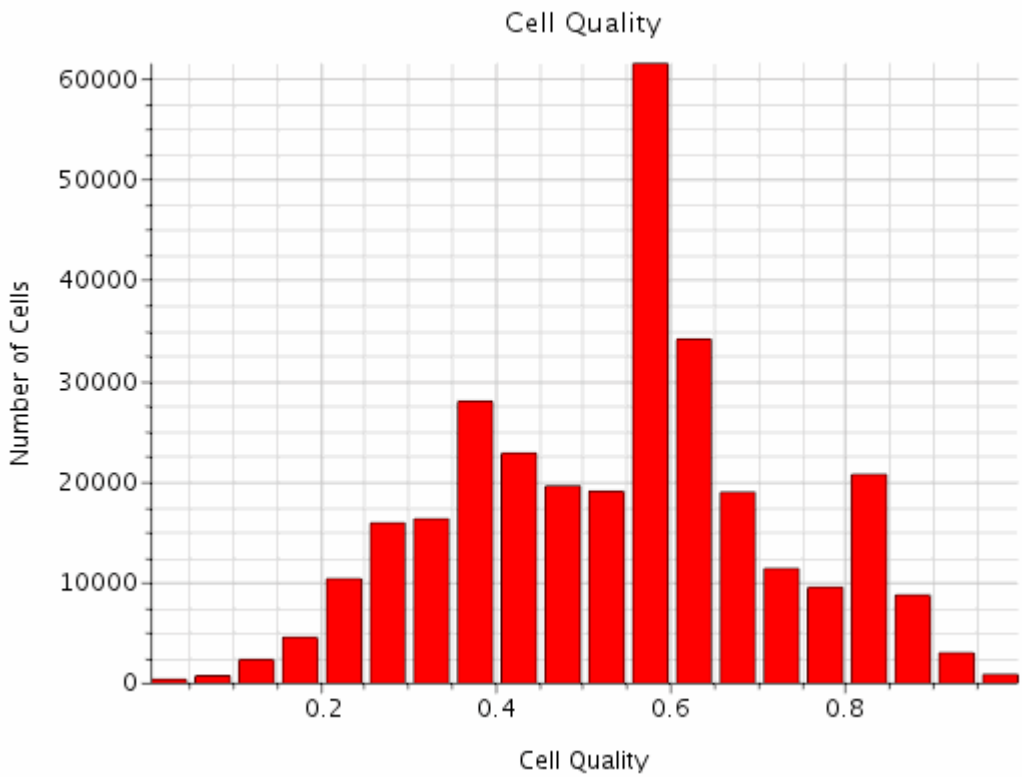
a) In the *Macro* toolbar, click (**Play Macro**).

b) In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/foundationTutorials/data`, select `checkMeshQuality.java` and click **Open**.

The macro sets up mesh quality plots. The cell quality, cell skewness angle, and volume change are displayed in histogram plots. The plots are opened automatically after running the macro. The macro also sets up derived parts and scenes to help you visualize the mesh and the mesh quality inside the scenes.

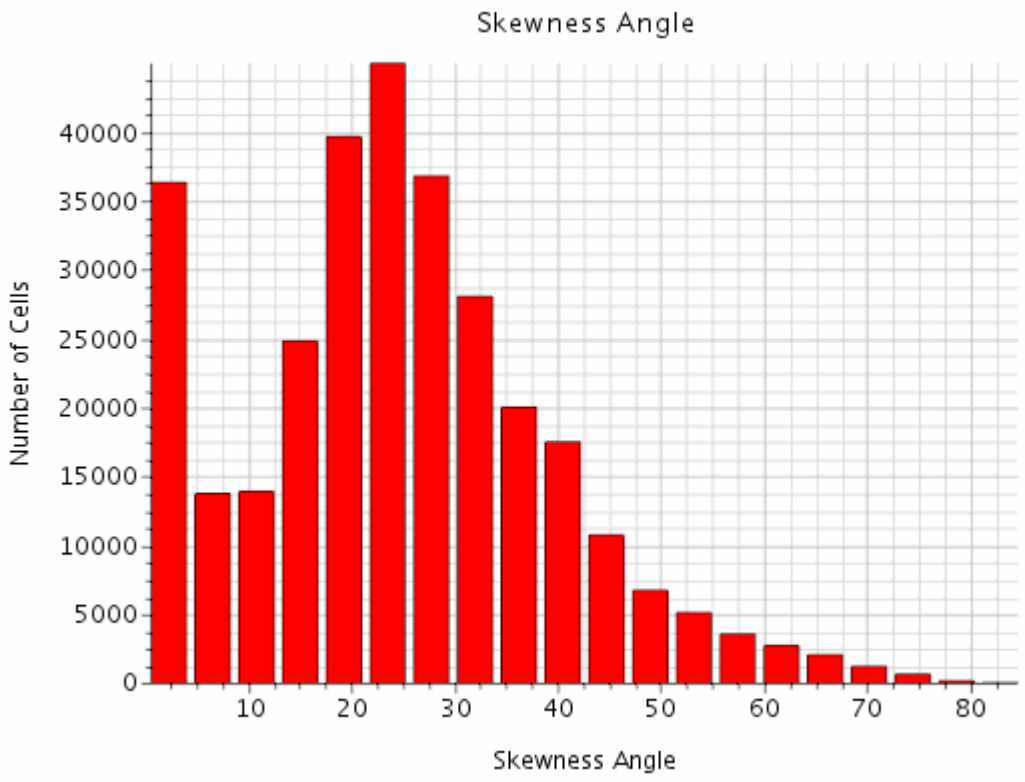
4. View the histogram plots:

a) Select the *Cell Quality Histogram* tab in the window.



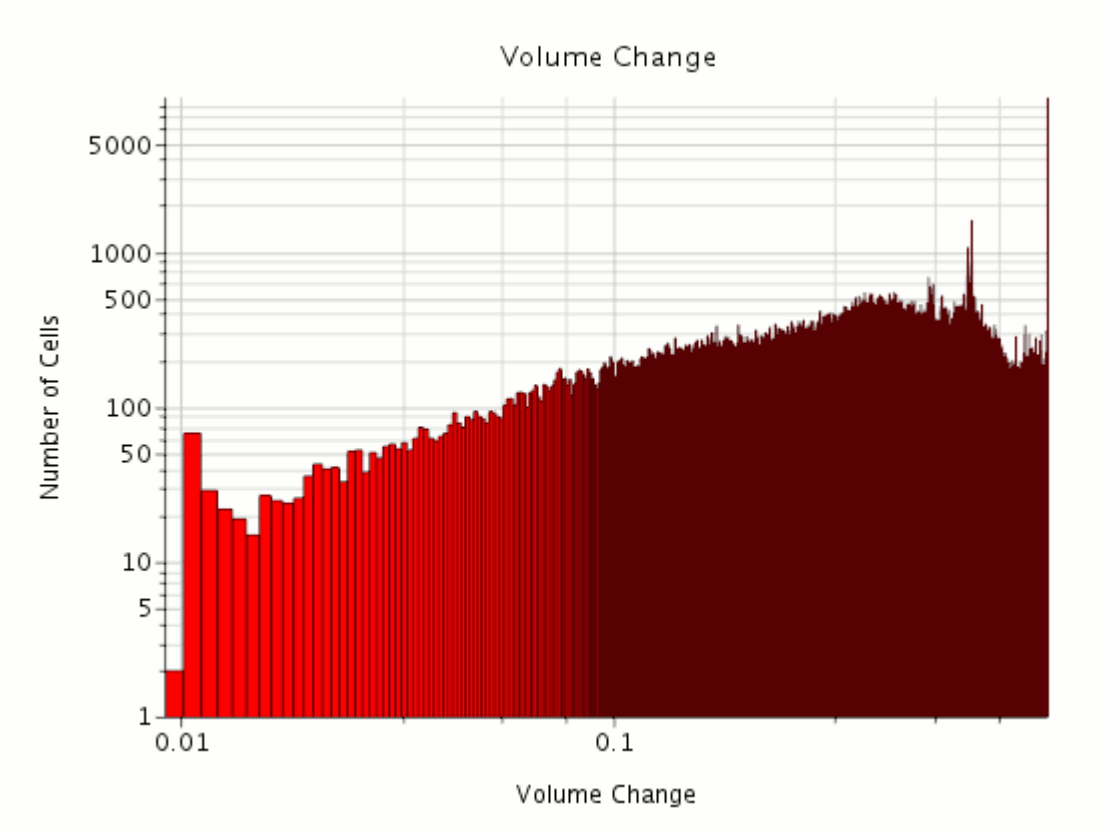
The shape of the cell affects the cell quality. Typically, uniform cells that have an even distance between the cell edge and the cell center are considered good quality cells. Squashed and uneven cells are considered poor quality cells. A perfect cell has a cell quality of 1.0. Cells with a quality less than 1.0e-5 are considered bad. You should aim for cells with a quality of 0.5 or greater.

b) Select the *Skewness Histogram* tab in the window.



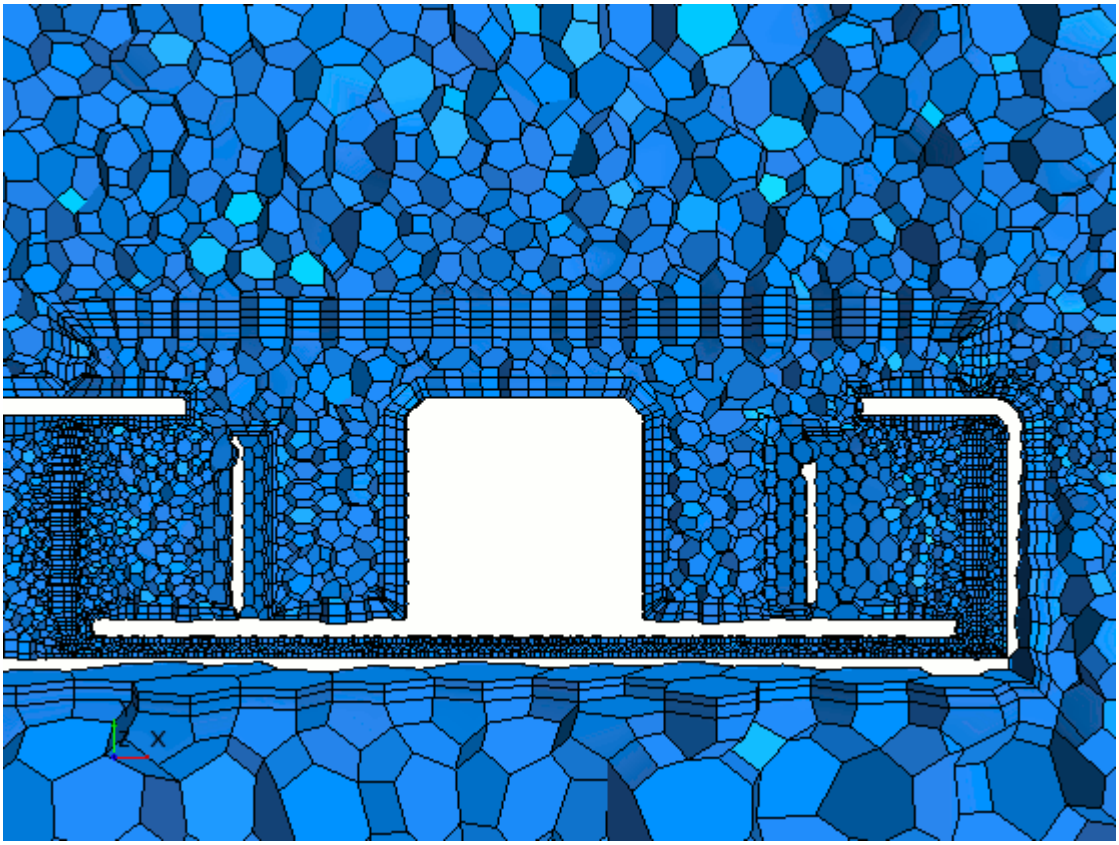
The skewness angle measures the angle between a face normal and the vector that connects two neighboring cell centroids. This measure is designed to show whether the cells on either side of a face are formed in such a way as to permit diffusion of quantities without these quantities becoming unbounded. A skewness angle of 0° indicates a perfectly orthogonal mesh. Cells with a skewness angle greater than 85° are considered bad. Skewness angles of 90 or greater typically result in convergence issues.

c) Select the *Volume Change Histogram* tab in the window.



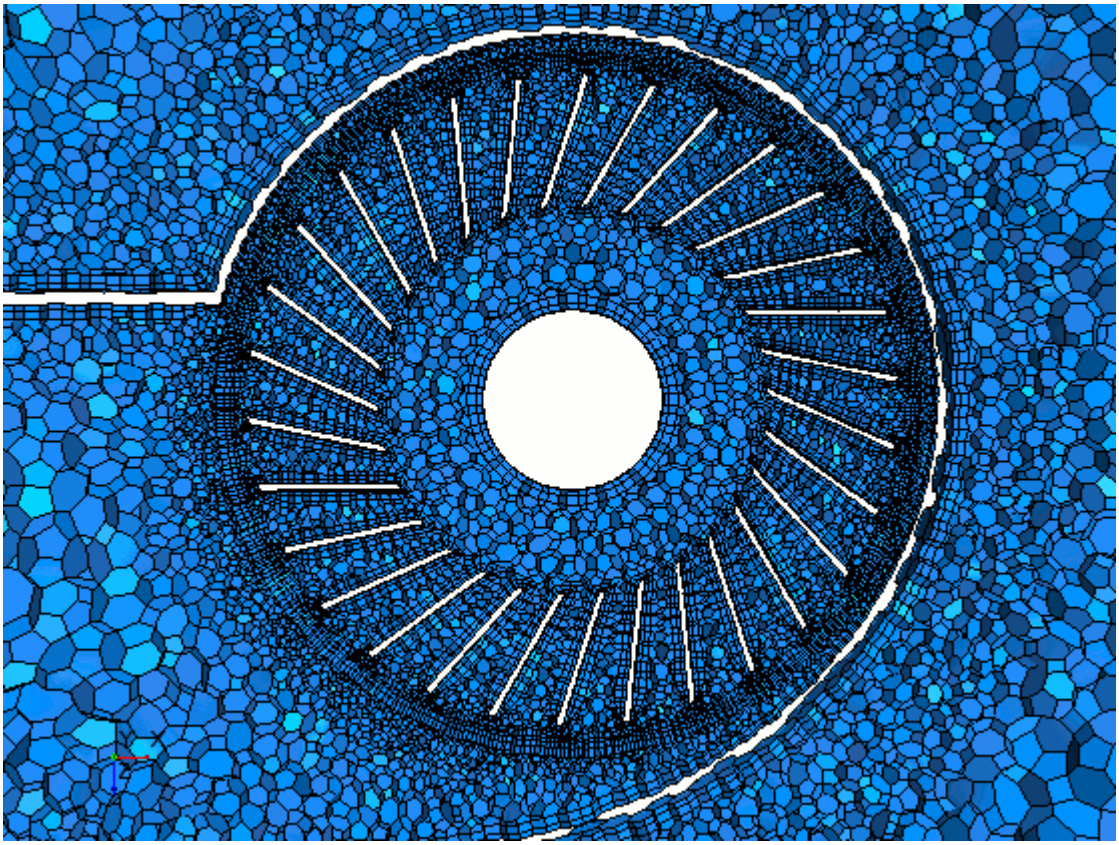
The volume change metric describes the ratio of the volume of a cell to that of its largest neighbor. A value of 1.0 indicates that the cell has a volume equal to or higher than its neighbors. Cells with a volume change of 0.01 or lower are considered bad cells. A large jump in volume from one cell to another can cause inaccuracies and instability in the solvers. From the histogram, you can see that there are some bad cells in the mesh.

5. View the mesh quality scenes:
   a) Select the *Mesh View: XY* tab in the window.



This scene shows a slice through the center of the simulation domain and displays the volume mesh in and around the rotating region. Upon closer inspection, you notice that the prism layer is generated on the interface between the two regions in the simulation domain. Prism layers are not required at this location. Another issue you can notice is that the prism layer is collapsed in the small gap between the fan and the card enclosure.

b) Select the *Mesh View: XZ* tab in the window.



This scene shows a different view of the volume mesh inside the card. In this scene, you can see that the prism layer around the fan blades are too thick to capture the velocity gradient effectively, and the prism layer cells start to collapse near the tip of the blades.

In general, the quality of the mesh inside the rotating region needs refinement to help achieve a more accurate solution. Prism layers are generated on the interface between the static region and rotating region, when they are not needed. The prism layer is not generated on the bottom surface underneath the fan. It is important to have prism layers in this area. The prism layers around the fan blades are too large to resolve the velocity gradient accurately.

6. Save the simulation.

Mesh: Extending the Simulation Domain with the Mesh Extruder

Cell count is important due to computational costs. Larger meshes increase computational time for the solver. One way of keeping the cell count down, while at the same time extending the simulation domain, especially on the inflow and outflow boundaries, is to use the extruder mesher.

The extruder mesher is made up of two operations: the surface extruder, and the volume extruder. The surface extruder extends the edges of the input surface to form the bounding surfaces of the extruded region. You specify the extrusion distance in the surface extruder operation. The volume extruder fills in the extruded region with extruded volume cells from the input surface. You control the amount of divisions in the extruded region in the volume extruder operation.

1. Launch STAR-CCM+ and load **foundationTutorial_9.sim**.

2. Save the simulation as `foundationTutorial_10.sim`.
3. If the scene showing the simulation domain is not open, double-click the **Scenes** > **Simulation Domain** node.
4. Create the Surface Extruder operation for the upstream region:
   a) Right-click the **Operations** node and select **New** > **Surface Preparation** > **Surface Extruder**.
   b) In the *Create Surface Extruder Operation* dialog, click **Select** next to the *Input Parts* field, select **Static Region** and click **OK**.
   c) Click **Select** next to the *Input Part Surfaces* field, select **Static Region** > **Block** > **Upstream** and click **OK**.
   d) Set *Extrusion Distance* to `0.25 m`.
   e) Click **OK** to close the *Create Surface Extruder Operation* dialog.

      A surface extruder operation is added to the **Operations** node, and a placeholder for the surface extruder part is added to the **Parts** node.
   f) Rename the **Parts** > **Surface Extruder** node to `Upstream Region`.
   g) Rename the **Operations** > **Surface Extruder** node to `Upstream Surface Extruder`.
5. Create the Surface Extruder for the downstream region:
   a) Click and drag the **Operations** > **Upstream Surface Extruder** node onto the **Operations** node.

      A copy of the surface extrude operation is created and added to the **Operation** node and a copy of the surface extruder part is added to the **Parts** node.
   b) Rename the **Parts** > **Copy of Upstream Surface Extruder** node to `Downstream Region`.
   c) Rename the **Operations** > **Copy of Upstream Surface Extruder** node to `Downstream Surface Extruder`.
   d) Select the **Operations** > **Downstream Surface Extruder** node and switch the *Input Part Surfaces* to **Static Region** > **Block** > **Downstream**.
   e) Select the **Downtream Surface Extruder** > **Controls** > **Direction Mode** > **Extrusion Distance** node and set *Extrusion Distance* to `0.5 m`.
6. Create the Volume Extruder operation:
   a) Right-click the **Operations** node and select **New** > **Mesh** > **Volume Extruder**.
   b) In the *Create Volume Extruder Operation* dialog, select **Downstream Region** and **Upstream Region** and click **OK**.

      A volume extruder operation is added to the **Operations** node.
   c) Select the **Volume Extruder** > **Controls** > **Number of Layers** node and set *Number of Layers* to `10`.
   d) Select the **Volume Extruder** > **Controls** > **Stretching Function** node and set *Stretching Method* to **One Sided Hyperbolic**.
7. Assign the extruded parts to regions:
   a) In the **Geometry** > **Parts** node, select the **Upstream Region** and **Downstream Region** nodes.
   b) Right-click on a selection and choose **Assign Parts to Regions...**.
   c) In the *Assign Parts to Regions* dialog, select:
      1. **Create a Region for Each Part**
      2. **Create a Boundary for Each Part Surface**

3. **Create Contact-mode Interfaces From Contacts**
4. Click **Apply** then **Close**.

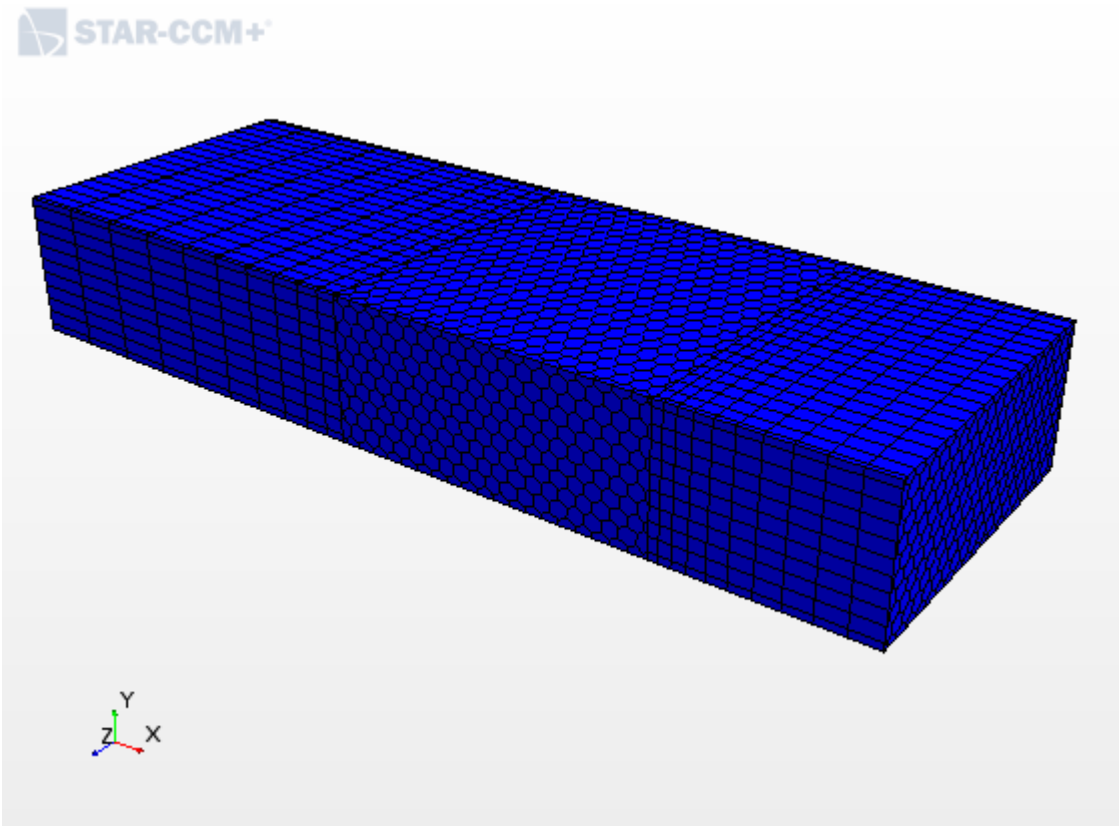
d) Expand the **Regions** > **Downstream Region** > **Boundaries** node, select the **Block.Downstream** and **Static Region.Block.Block Surface** boundaries, and set *Type* to **Symmetry Plane**.

e) Expand the **Regions** > **Upstream Region** > **Boundaries** node, select the **Block.Upstream** and **Static Region.Block.Block Surface** boundaries, and set *Type* to **Symmetry Plane**.

8. Right-click the **Operations** node and select **Execute All**.
9. Add the new regions to the mesh scene showing the simulation domain:

a) In the **Regions** node, multi-select the **Downstream Region** and **Upstream Region** regions.

b) Click and drag these regions into the *Simulation Domain* scene and click **Add to Mesh 1**.



10. Save the simulation.

Mesh: Refining the Mesh

Customize and refine specific areas of the simulation domain using custom controls.

Custom controls override the default controls for the areas that you specify. For example, you can customize the number of prism layers that are generated on a surface that is specified within a custom control. The number of prism layers that are generated outside the custom control match the settings in the default controls.

Local meshing speeds up the re-execution of the surface remesher after local changes are made to the input geometry or mesh settings. With this feature, STAR-CCM+ only remeshes surfaces within a defined location and leaves other surfaces unchanged. With local meshing, you set limits on where you want remeshing to occur by defining local extents. In this tutorial, the area surround the fan contains all the changes and refinements to the mesh, so it would be ideal to use a volume extent that encloses the fan.

Create custom controls that focus on specific areas of the simulation domain:

1. Launch STAR-CCM+ and load **foundationTutorial_10.sim**.
2. Save the simulation as `foundationTutorial_11.sim`.
3. If the scene showing the fan and enclosure is not open, double-click the **Scenes** > **Fan and Enclosure** node.
4. Adjust the prism layer settings to improve the quality of the prism layer cells around the fan blades and in the small gap between the base of the fan and the enclosure:
   a) Expand the **Operations** > **Automated Mesh** > **Meshers** node.
   b) Select the **Prism Layer Mesher** node.
   c) Set *Gap Fill Percentage* to `45`.

      The gap fill percentage specifies the percentage of space between a narrow gap that the prism layer mesher fills with prism cells.
   d) Set *Boundary March Angle* to `25`.

      The boundary march angle controls the way the prism layers are projected on a boundary that does not have a prism layer.
   e) Select the **Automated Mesh** > **Default Controls** > **Number of Prism Layers** node and set *Number of Prism Layers* to `3`.
   f) Right-click the **Automated Mesh** node and select **Execute**.
5. Set up local meshing:

   The changes made in the previous step improve the prism layers, however further refinement is still required in the rotating region. Local meshing limits the changes to the specified extent, reducing the total time it takes to generate the surface mesh. Areas outside the extent remain unchanged.
   a) Select the **Operations** > **Automated Mesh** node and activate the **Perform Local Surface Meshing** option.

      The **Local Extents** node is added to the **Automated Mesh** operation node and a pin is added to the icon of the Automated Mesh operation.
   b) Right-click the **Automated Mesh** > **Local Extents** node and select **New** > **Volume Extent**.

      A **Volume Extent** node is added to the **Local Extents** node.
   c) Select the **Local Extents** > **Volume Extent** node.
   d) Set *Parts* to **Disc**.
   e) Set *Inflation Distance* to `2 mm`.

      The inflation distance specifies a distance beyond the extent. Any surfaces falling within this distance are included in remeshing, regardless of whether they are connected to the original surface parts covered by the extent.
   f) Set *Inflation Layers* to `3`.

      The inflation layer specifies a number of adjacent bands of triangles by which the extent can grow.

6. Create a custom control that disables the generation of prism layers on the interface between the static region and the rotating region:

   Prism layers are not required at this location so you can disable them to reduce the cell count in the mesh.

   a) Right-click **Automated Mesh** > **Custom Controls** and select **New** > **Surface Control**.

      A surface control node, **Surface Control**, is added to the **Custom Controls** node.

   b) Rename **Surface Control** to `Disable Prism Layer`.

   c) Select the **Disable Prism Layer** node and click ... (**Custom Editor**) next to *Part Surfaces*.

   d) In the *Disable Prism Layer - Part Surfaces* dialog, select the **Rotating Region** > **Surfaces** > **Interface** and **Static Region** > **Disc** > **Interface** nodes and click **OK**.

   e) Expand the **Disable Prism Layer** > **Controls** node.

   f) Select the **Prism Layers** node and set *Prism Layers* to **Disable**.

7. Create a custom control that refines the cell size in the rotating region and customizes the prism layer around the fan blades:

   The rotating graphics card fan contains narrow gaps and would benefit from a finer mesh to capture the flow in and around the fan more accurately. The prism layer is especially important around the fan blades, where the swirling flow is generated. The current prism layer is too thick to resolve the boundary layer accurately, and the prism layer cells are a too large. To improve the mesh, decrease the prism layer total thickness.

   a) Right-click **Automated Mesh** > **Custom Controls** and select **New** > **Volumetric Control**.

      A volumetric control node, **Volumetric Control**, is added to the **Custom Controls** node.

   b) Rename **Volumetric Control** to `Fan Cell Refinement`.

   c) Select the **Fan Cell Refinement** node and click ... (**Custom Editor**) next to *Parts*.

   d) In the *Fan Cell Refinement - Parts* dialog, select the **Disc** node and click **OK**.

   e) Edit the **Fan Cell Refinement** > **Controls** node and set the following parameters:

Node	Property	Setting
Surface Remesher	**Customize Size**	**Activated**
Prism Layer Mesher	**Customize Total Thickness**	**Activated**
Polyhedral Mesher	**Customize Polyhedral Mesher**	**Activated**

   f) Edit the **Fan Cell Refinement** > **Values** node and set the following parameters:

Node	Property	Setting
Custom Size	**Size Type**	**Absolute**
	Absolute Size	`1.5 mm`
Custom Prism Values > **Prism Layer Total Thickness**	**Size Type**	**Absolute**
	Absolute Size	`0.7 mm`

8. Save the simulation.

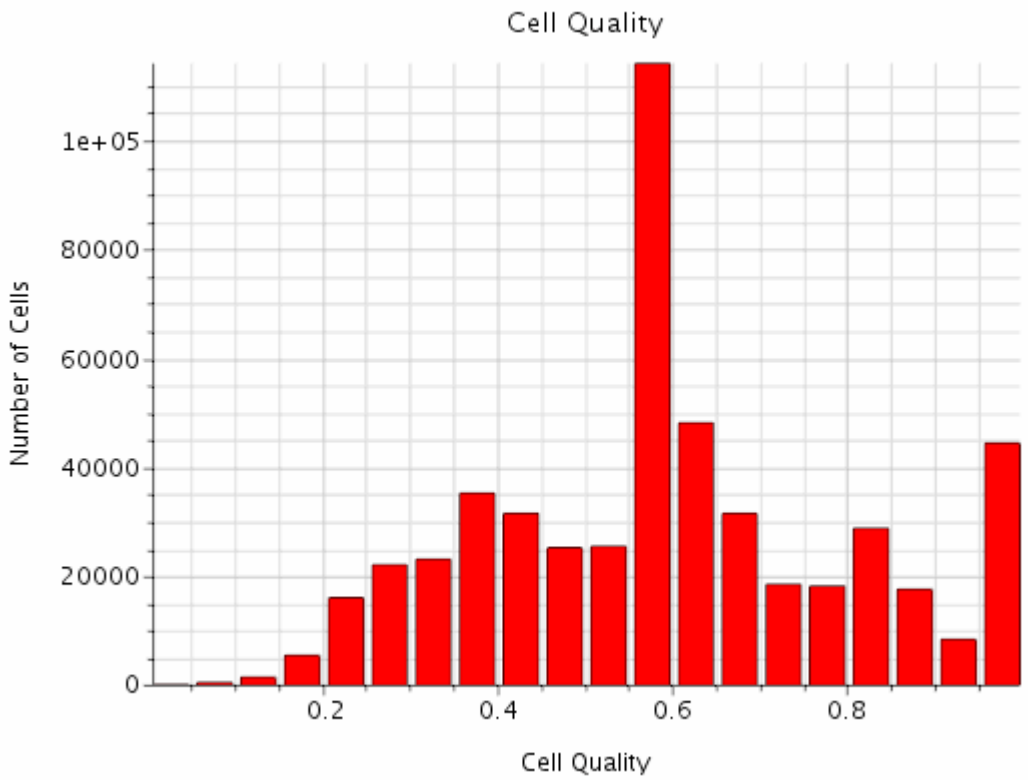
9. To update the volume mesh with the changes, right-click the **Automated Mesh** node and select **Execute**.

The mesh generation takes less time as only the surface mesh inside the extent are regenerated. Surfaces outside the extent remain unchanged and do not mesh. If you want to modify an area outside the extent, either add another extent to cover this area or if more general changes are required, disable local meshing.

10. To update the surface and volume extruder, right-click the **Operations** node and select **Execute All**.

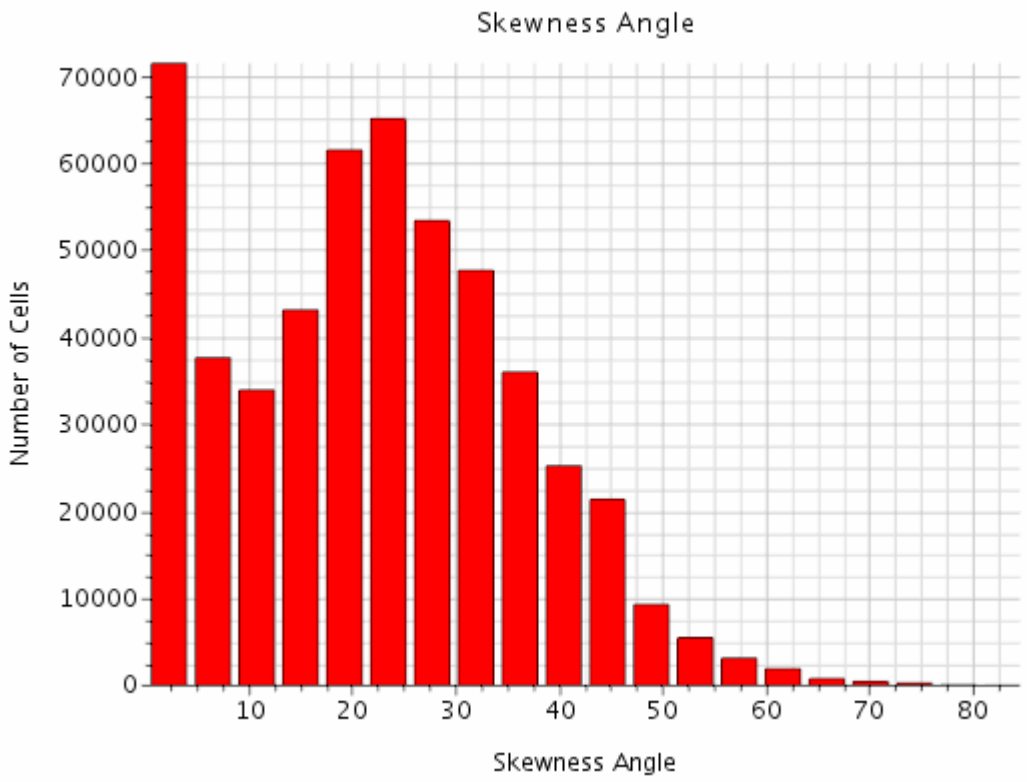
Use the mesh metric plots and scene that you set up in Foundation Tutorial 9 to check how the changes have affected the mesh quality.

11. Look at the histogram plots:
    a) If the histogram plots are not displayed in the graphics window, right-click the **Plots** node and select **Open All**.
    b) Select the *Cell Quality Histogram* tab in the window.



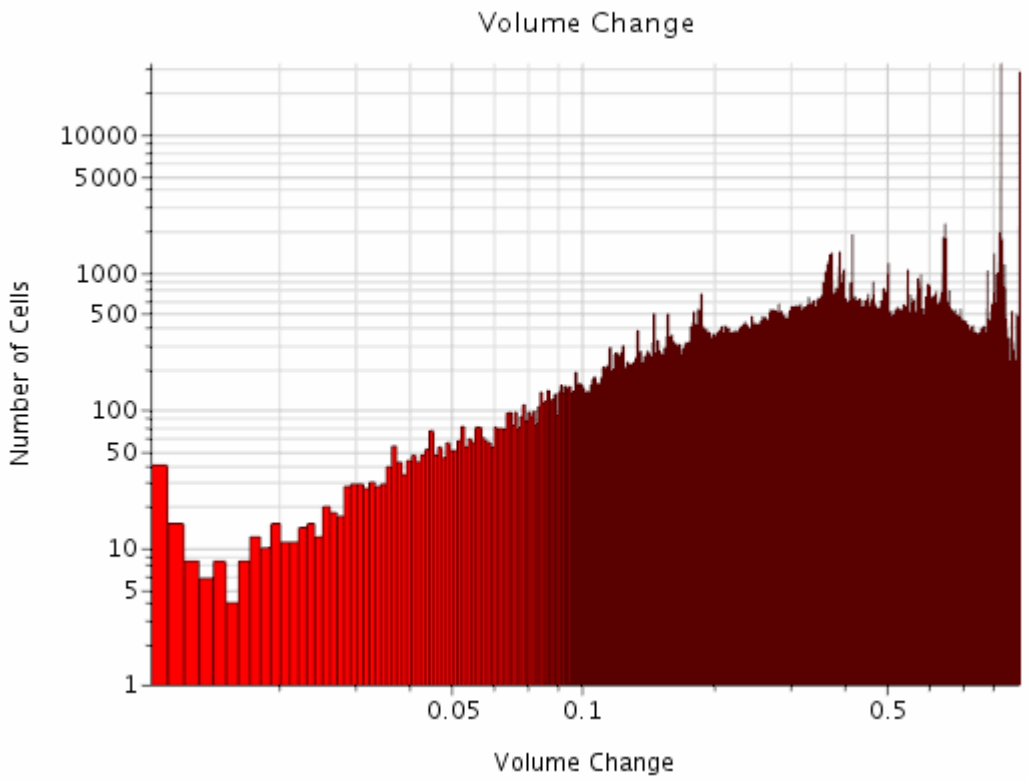
In general the cell quality has improved. The most notable change here is that there are now a large number of perfect cells (with cell quality of 1.0). These perfect cells are attributed to the even cells generated by the volume control.

    c) Select the *Skewness Histogram* tab in the window.



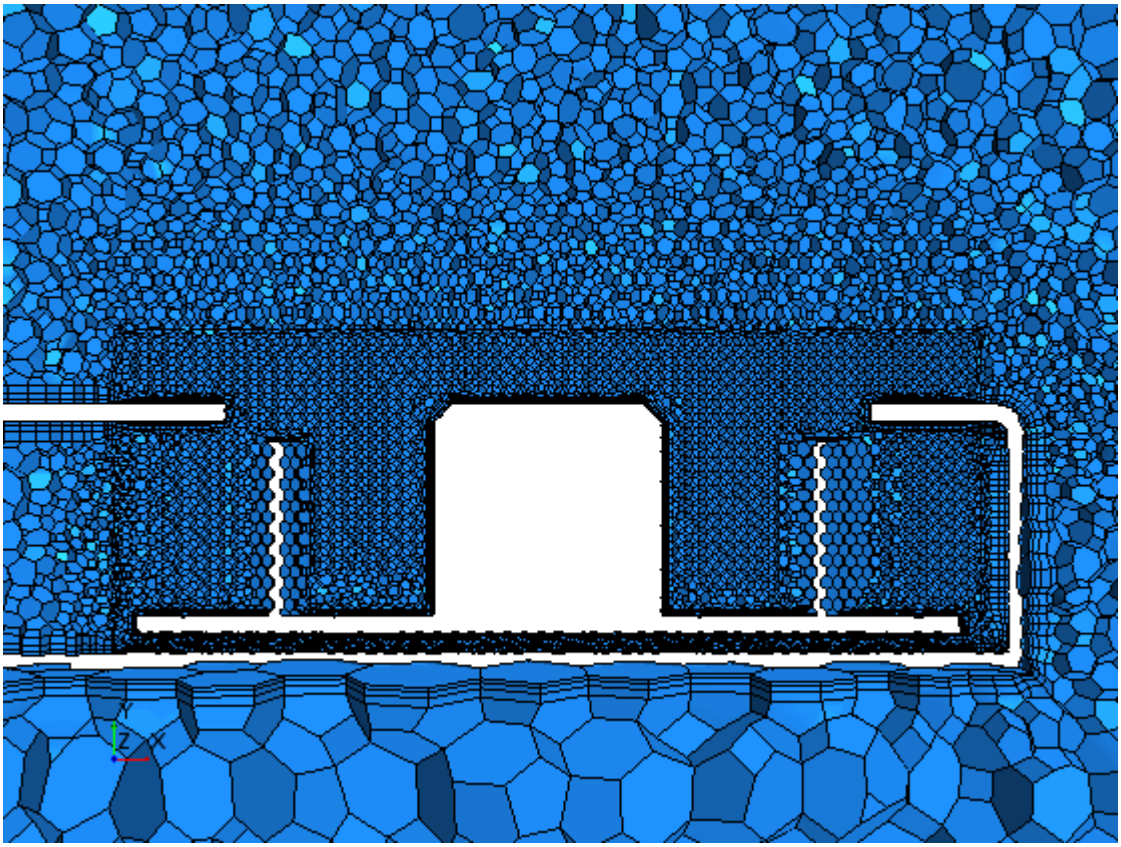
In general, the skewness angle has improved. This improvement can be attributed to the even cells generated by the volume control.

d) Select the *Volume Change Histogram* tab in the window.



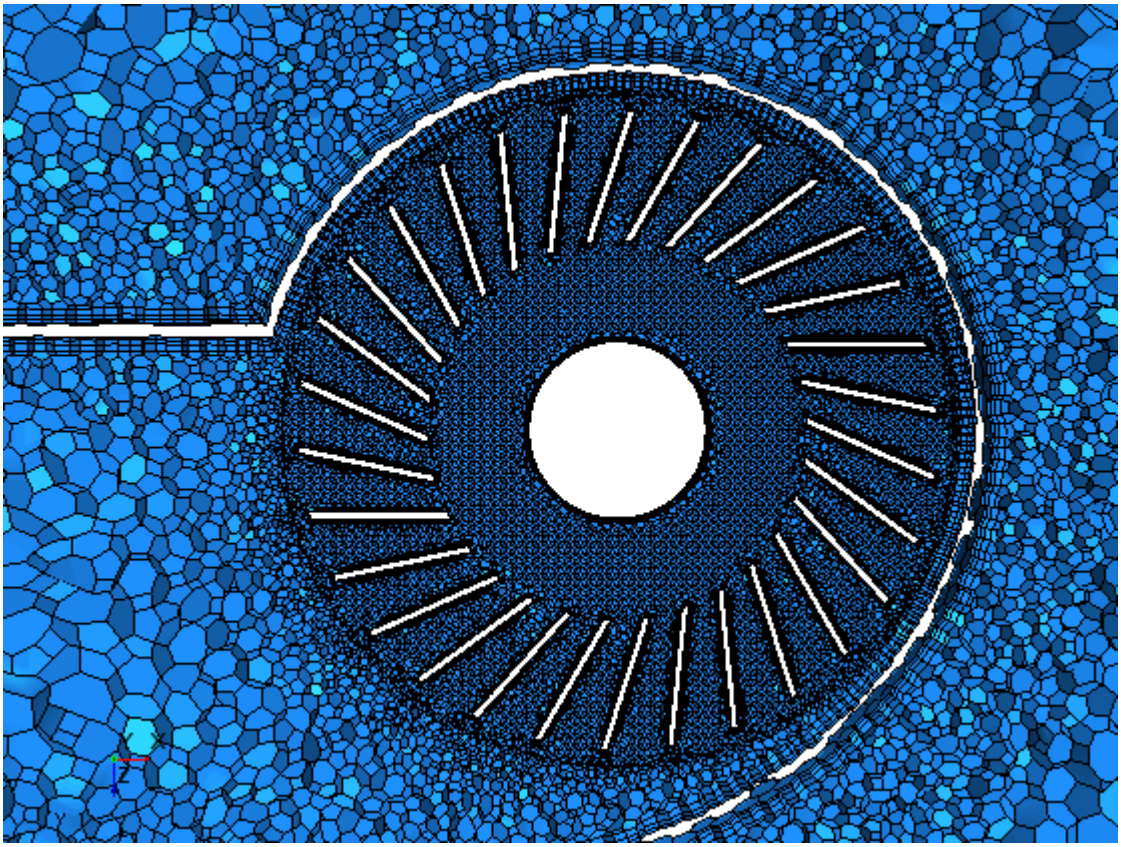
There are no longer any bad cells (with volume change of 0.01 or lower).

12. View the mesh quality scenes:
    a) If the mesh quality scenes are not displayed in the graphics view, expand the **Scenes** > **Mesh View Scenes** node and double-click each scene node.
    b) Select the *Mesh View: XY* tab in the window.



The refined cells inside the rotating region are clearly visible. The prism layer thickness inside the rotating region shrinks to fit inside the tight spaces. The prism layer is no longer collapsed in the small gap below the fan.

c) Select the *Mesh View: XZ* tab in the window.



The refined cells inside the rotating region are clearly visible. The prism layer is no longer generated between the two regions. The prism layer is no longer collapsed at the tips of the fan blades.

Mesh: Rotating Region

Add motion to the fan in the rotating region of the rotating region of the blower fan simulation.

1. Launch STAR-CCM+ and load **foundationTutorial_11.sim**.
2. Save the simulation as `foundationTutorial_12.sim`.

Enter an example that illustrates the current task (optional).

Enter the tasks the user should do after finishing this task (optional).

Physics - Flow

Contents:

Multi-Component Fluid: Steady Flow in a Dilution Pipe
Multi-Component Fluid: Time-Varying Boundary Conditions in a Dilution Pipe

Multi-Component Fluid: Steady Flow in a Dilution Pipe

Multi-component flows are composed of multiple chemical species that are mixed at the molecular level and share velocity and temperature. STAR-CCM+ allows you to model multi-component flows in a single physics continuum.

In this tutorial, constant steady streams of air and methane gas flow into a simplified dilution pipe at 5 m/s and 10 m/s, respectively. The flow is turbulent and compressible. For this case, the assumption of steady flow is a simplification—the real flow behavior is not captured.

1. Launch STAR-CCM+ and load `foundationTutorial_dilutionPipe_Start.sim` from `[INSTALL_DIR]/doc/startutorialsdata/foundationTutorials/data`.
2. Save the simulation as `foundationTutorial_dilutionPipe_Steady.sim`
3. Right-click the **Continua** > **Physics 1** node and choose **Select models**

The *Physics Model Selection* dialog guides you through the model selection process by showing only options that are appropriate to the choices already made.

4. Select the following models in order:

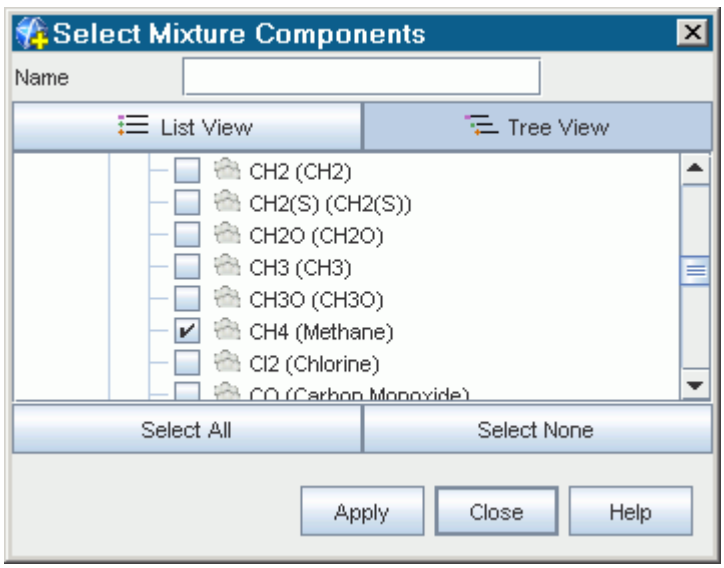
Group Box	Model
Material	**Multi-Component Gas**
Reaction Regime	**Non-Reacting**
Flow	**Segregated Flow**
	Gradients (Selected automatically)
	Segregated Species (Selected automatically)
Equation of State	**Ideal Gas**
Energy	**Segregated Fluid Temperature**
Time	**Steady**
Viscous Regime	**Turbulent**
	Reynolds-Averaged Navier-Stokes (Selected automatically)
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**
	Exact Wall Distance (Selected automatically)

Group Box	Model
	Two-Layer All y+ Wall Treatment (Selected automatically)
	Realizable K-Epsilon Two-Layer (Selected automatically)

5. Click **Close**.

Now assign material properties to the two fluid components. For air, the default properties provided by STAR-CCM+ are used. For methane, specify a molecular viscosity of 1.175E-5 Pa-s, a molecular weight of 16 kg/kmol, a specific heat of 2230 J/kg-K, and a thermal conductivity of 0.038 W/m-K. The gas mixture density is calculated using the Ideal Gas Law.

6. Within **Continua** > **Physics 1**, right-click the **Models** > **Multi-Component Gas** > **Gas Components** node and choose **Select Mixture Components**.
7. In the *Select Mixture Components* dialog, expand the **Material Databases**, **Standard**, and **Gases** nodes to reveal the available gaseous components. Select the components **Air** and **CH4 (Methane)**, then click **Apply** and **Close**.



8. Select the **Multi-Component Gas** > **Material Properties** > **Thermal Conductivity** node and set *Method* to **Mass-Weighted Mixture**.
9. Open the **Multi-Component Gas** > **Gas Components** > **CH4** > **Material Properties** nodes and set the following values:

Node	Property	Value
Dynamic Viscosity > **Constant**	*Value*	`1.175E-5 Pa-s`
Molecular Weight > **Constant**	*Value*	`16 kg/kmol`
Specific Heat > **Constant**	*Value*	`2230 J/kg-K`
Thermal Conductivity > **Constant**	*Value*	`0.038 W/m-K`

The material properties of the mixture as a whole, under **Models** > **Multi-Component Gas** > **Material Properties**, are calculated from the properties and proportions of the individual components, using the following methods:

Property	Method
Dynamic Viscosity	**Mass-Weighted Mixture**
Molecular Diffusivity	**Schmidt Number**
Molecular Weight	**Mixture**
Specific Heat	**Mass-Weighted Mixture**
Thermal Conductivity	**Mass-Weighted Mixture**
Turbulent Prandtl Number	**Constant**
Turbulent Schmidt Number	**Constant**

Set the initial conditions within the dilution pipe.

10. Select the **Physics 1** > **Initial Conditions** > **Species Mass Fraction** node and click ... (**Custom Editor**) for the *Value* property.
11. In the *Species Mass Fraction - Value* dialog, enter `1.0` for the *Air* mass fraction, then click **OK**.
12. Select the **Static Temperature** node and change the *Value* to `293.0 K`.

Specify boundary conditions at the two inlets as well as the pressure boundary. Methane enters the pipe through the inner inlet, while air enters through the outer ring.

13. Open the **Regions** > **Dilution Pipe** > **Boundaries** > **Inlet Inner** > **Physics Values** node and set the following boundary conditions:

Node	Property	Value
Species Mass Fraction	*Value*	`Air = 0.0, CH4 = 1.0`
Static Temperature	*Value*	`323.0 K`
Turbulence Intensity	*Value*	`0.1`
Turbulent Viscosity Ratio	*Value*	`100`
Velocity Magnitude	*Value*	`10` m/s

Define conditions at the air inlet.

14. Open the **Inlet Outer** > **Physics Values** node and set the following boundary conditions:

Node	Property	Value
Species Mass Fraction	*Value*	`Air = 1.0, CH4 = 0.0`
Static Temperature	*Value*	`293.0 K`
Turbulence Intensity	*Value*	`0.1`
Turbulent Viscosity Ratio	*Value*	`40`

Node	Property	Value
Velocity Magnitude	*Value*	5 m/s

Similar values must be assigned to the pressure boundary. These values are only applied on cell faces where backflow occurs.

15. Edit the **Outlet** > **Physics Values** node and set the following boundary conditions:

Node	Property	Value
Species Mass Fraction	*Value*	`Air = 1.0, CH4 = 0.0`
Static Temperature	*Value*	`308.0 K`
Turbulence Intensity	*Value*	`0.1`
Turbulent Viscosity Ratio	*Value*	`40`

The default **Pressure** value is suitable for this tutorial.

Run the calculation for 300 iterations, which is sufficient to achieve an approximate steady-state solution. In reality, the flow contains an impinging jet passing over an orifice and so is unlikely to exhibit true steady behavior.

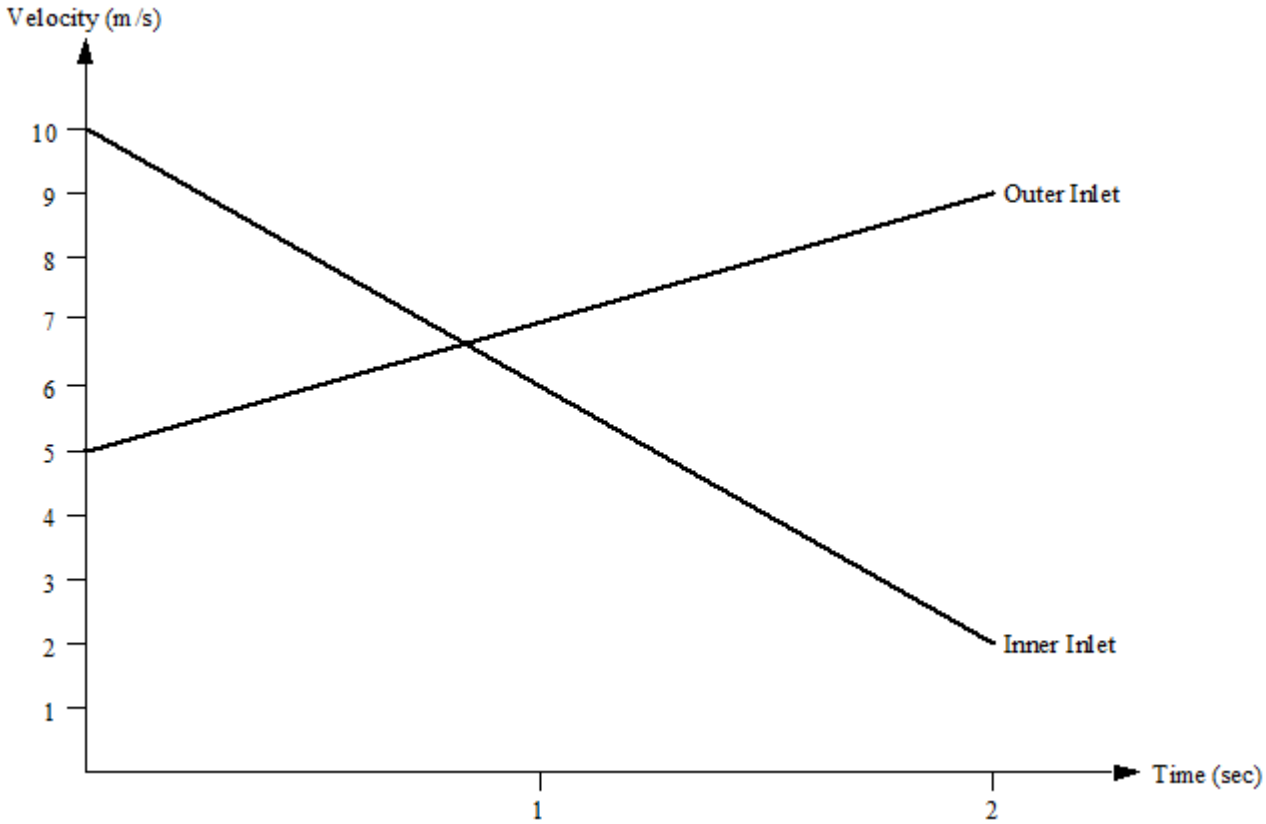
16. Select the **Stopping Criteria** > **Maximum Steps** node and set the *Maximum Steps* to `500`.
17. Save the simulation.
18. Click (**Run**) in the top toolbar.
    The Residuals display is created automatically and shows the progress of the solver.
19. Save the simulation at the end of the run.

Multi-Component Fluid: Time-Varying Boundary Conditions in a Dilution Pipe

Some analyses require that you set time-varying conditions on boundaries. User-defined field functions provide a method whereby you can do so.

This tutorial resumes from the results that were obtained in the previous tutorial, "Multi-Component Fluid: Steady Flow in a Dilution Pipe." It solves the same problem except that, in this case, the velocities at the two inlet boundaries vary with time over a period of two seconds.

The velocity-time profiles at the inlets are shown below.



1. Launch STAR-CCM+ and load `foundationTutorial_dilutionPipe_Steady.sim`.

   You can either use the sim file that you saved from the previous tutorial, or load the sim file provided in `[INSTALL_DIR]/doc/startutorialsdata/foundationTutorials/data`.
2. Save the simulation as `foundationTutorial_dilutionPipe_Unsteady.sim`

Most of the models have already been selected in the tutorial Multi-Component Fluid: Steady Flow in a Dilution Pipe. However, you must change the analysis type to unsteady:

3. Right-click the **Continua** > **Physics 1** node and choose **Select models...**
4. In the *Physics Model Selection* dialog:
   a) From the *Enabled Models* group box, deselect **Steady**.
   b) From the *Time* group box, select **Implicit Unsteady**.
   c) Click **Close**.

To define the inlet velocity profiles as linear functions of time, you create user-defined field functions that can be assigned to the velocity inlet boundary conditions. The required functions are:

- Inner Inlet: $v = 10 - 4t$
- Outer Inlet: $v = 5 + 2t$

5. Right-click the **Tools** > **Field Functions** node and select **New** > **Scalar**.

   The *Properties* window for the new function opens.
6. Rename the field function node **User Field Function 1** to **Inner Inlet Velocity**.
7. Select the **Inner Inlet Velocity** node and set:

   a) *Definition* to `10-4*$Time`
   b) *Function Name* to `Inner Inlet Velocity`
8. Create another scalar field function and rename its node to **Outer Inlet Velocity**.
9. Select the **Outer Inlet Velocity** node and set:
   a) *Definition* to `5+2*$Time`
   b) *Function Name* to `Outer Inlet Velocity`

To specify the inlet velocity boundary conditions using these field functions:

10. Select the **Regions** > **Dilution Pipe** > **Boundaries** > **Inlet Inner** > **Physics Values** > **Velocity Magnitude** node and set *Method* to **Field Function**.
11. Click the right side of the *Scalar Function* property.
12. In the dialog that appears, select the **Inner Inlet Velocity** field function node, then click **OK**.
13. Open the **Boundaries** > **Inlet Outer** node. Following the same procedure as described above, define the velocity magnitude at the outer inlet by the **Outer Inlet Velocity** field function.
14. Save the simulation.

Set the time-step length and the run time. The simulation is to restart from the steady-state result that was obtained previously and then run for 2 seconds of physical time. As a time-step of 0.002 seconds is used, the analysis requires 1000 steps.

15. Select the **Solvers** > **Implicit Unsteady** node and set *Time-Step* to `0.002 s`.
16. Select the **Stopping Criteria** > **Maximum Physical Time** node and set *Maximum Physical Time* to `2.0 s`.
17. Select **Plots** > **Residuals** and set the *X-Axis Monitor* property to **Physical Time**.
18. Save the simulation.
19. To run the simulation, click (**Run**) in the top toolbar.

    If necessary, click the *Residuals* tab to bring the Residuals plot into view. The tabs at the top of the *Graphics* window make it possible to select any of the active displays for viewing.

    During the run, it is possible to stop the process by clicking (**Stop**) in the toolbar. If you do halt the simulation, it can be continued later by clicking (**Run**). If left alone, the simulation runs for 2.0 s of physical time.
20. Save the simulation.

Usability

Contents:

Usability: Creating a Template Simulation File
Usability: Applying the Template Simulation File

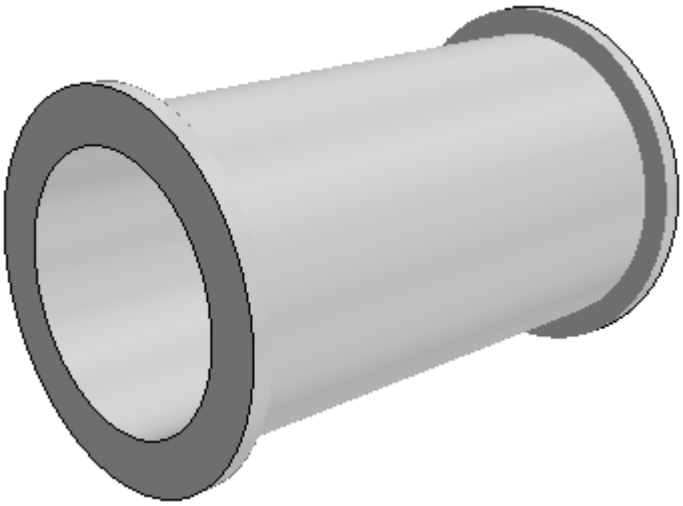
Usability: Creating a Template Simulation File

Template simulation files allow you to define a set of physics models and conditions once, and then apply those settings to multiple geometries.

By using dynamic queries in the meshing pipeline, and for region and boundary assignment, you can create a template file that requires little effort to apply in subsequent simulations.

In this tutorial you take a prototype simulation file and convert it to a template simulation file. The prototype simulation is for a cold-flow analysis of pipework. You import the solid geometries of the pipework—STAR-CCM+ extracts the wetted volume. Although the sample geometry contains only a single pipe section, the resulting template file is suitable for pipework with multiple sections.

A key factor in designing the template is to use mesh operations that can function based on dynamic queries.



This tutorial and the one that follows it only consider the internal volume. The template simulation file is for an internal cold flow simulation only. Therefore there are no interfaces required between the fluid and solid. Also, since a single internal volume is extracted, there are no fluid-fluid interfaces between pipes at the regions level.

The starting file already has the following:

- An initial sample part
- The meshing pipeline defined. The steps in the pipeline are:
  1. Fill Holes

2. Imprint
3. Extract Volume
4. Automated Mesh

The Fill Holes operation is necessary for closing the ends of the open pipes before applying the Extract Volume operation. The Imprint operation ensures that part contacts are defined for all touching surfaces.

- A region and its boundaries defined
- Physics models selected. The models are typical for an internal cold-flow analysis, and include the Realizable K-Epsilon Two-Layer turbulence model.
- Several scenes

1. Launch STAR-CCM+ and load `QueryTutorial_StartingFile.sim` from `[INSTALL_DIR]/doc/startutorialsdata/foundationTutorials/data`.
2. Save the simulation as `QueryTutorial_TemplateFile.sim`.

Convert the meshing pipeline to use dynamic queries for selecting the input parts to which each operation applies:

3. Create two tags for identifying the upstream and downstream faces of the pipework.

   Tags allow you to identify almost any object within a simulation. Here, you create tags that are then applied to faces within geometry parts.

   a) Right-click the **Tools** > **Tags** node and select **New Tag**.
   b) Rename the **Tags** > **Tag** node, `Inlet End`.
   c) Create a second tag and rename it, `Outlet End`.

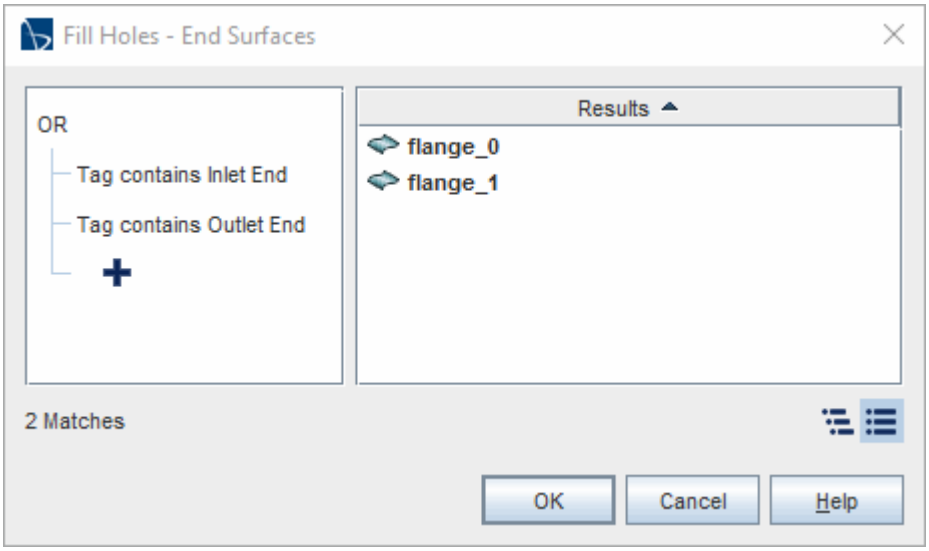
4. Apply the tags to the corresponding upstream and downstream faces of the sample part:

   a) Expand the **Geometry** > **Parts** > **pipe_section** > **Surfaces** node.
   b) Right-click the **flange_0** node and select **Set Tags** > **Inlet End**.
   c) Select the **flange_1** node and apply a tag using another method. In the *Tags* property, select **Outlet End** from the drop-down list.

5. To identify the end surfaces that guide the **Fill Holes** operation, use these same tags in a dynamic query:

   a) Select the **Geometry** > **Operations** > **Fill Holes** node.
   b) In the right half of the *End Surfaces* property, click the query editor button ( ).
   c) Activate the drop-down list and select **OR**.
   d) To add the first predicate, click (**Add a new item**). Use the controls to set `Tag contains Inlet End` and click (**Commit changes**).
   e) Add a second predicate and set it to `Tag contains Outlet End`.

STAR-CCM+ provides a preview of the objects that are selected by these predicates. As you already applied the **Inlet End** and **Outlet End** tags, you can see that the **flange_0** and **flange_1** parts are selected.



f) Click **OK**.

6. Also for the Fill Holes operation, set the *Input Parts* property to select any input part that contains "pipe" in its name:

   a) In the right half of the *Input Parts* property, click the query editor button ( ).

   b) Add a predicate, `Name contains pipe`.

   c) Click **OK**.

For the Imprint and Extract Volume operations, you create a filter that identifies the surfaces that surround the wetted volume. This filter assumes that all internal surfaces of any imported pipework have the text "pipe" within their names.

7. To create the filter for internal wetted surfaces:

   a) Right-click the **Tools** > **Filters** node and select **New**.

   b) In the *Filter* window, from the drop-down list, select **Name**.

   c) In the middle drop-down list, leave the setting at its default, **contains**.

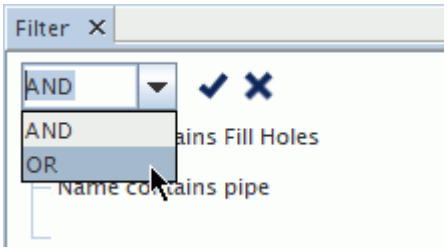
   d) In the text box to the right of the second drop-down list, enter `Fill Holes`.

   e) To finish the first entry, or *predicate*, click (**Commit changes**).

   f) To begin the second predicate, click (**Add a new item**).

   g) Set this predicate to `Name contains pipe`.

   h) Click the AND operator in the upper left corner of the view. In the drop-down list that appears, change it to **OR**.



The first filter is now complete.

i) Close the *Filter* window.

j) Rename the **Filters** > **Filter** node `Wetted Surfaces Filter`.

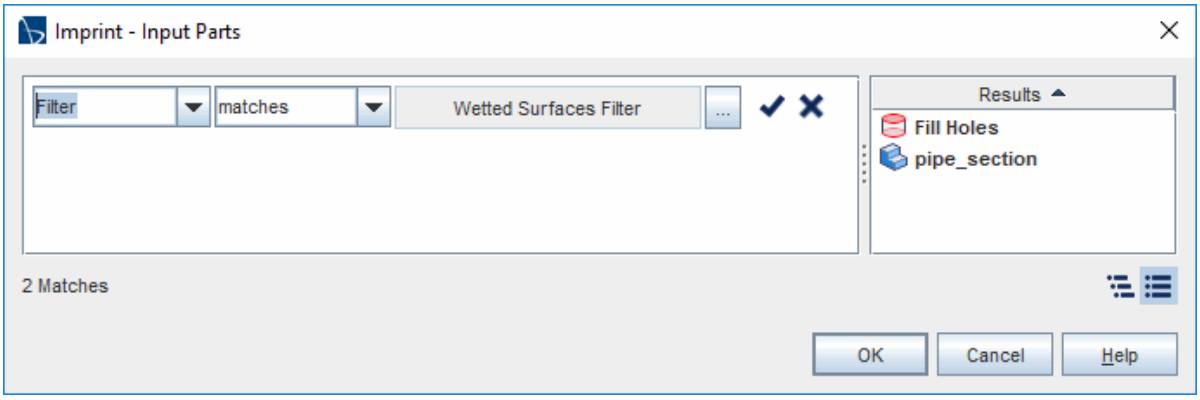
8. Now move on to the **Imprint** operation. Here, you want to choose all parts that participate in the meshing pipeline, including the result of the **Fill Holes** operation.

   In this example, the *Merge/Imprint Method* is set to **Discrete Imprint Curves**, which is the most suitable method for this case.

   a) Select the **Operations** > **Imprint** node.

   b) For the *Input Parts* property, click the query editor button ( ).

   c) Set the predicate to `Filter matches Wetted Surfaces Filter`. Click (**Commit changes**).



   d) Click **OK**.

9. Select the **Operations** > **Extract Volume** node, and repeat the previous steps.

The **Automated Mesh** operation keeps its original settings since it references the **Extract Volume** operation. You are ready to build the mesh.

10. Right-click the **Operations** node and select **Execute All**.

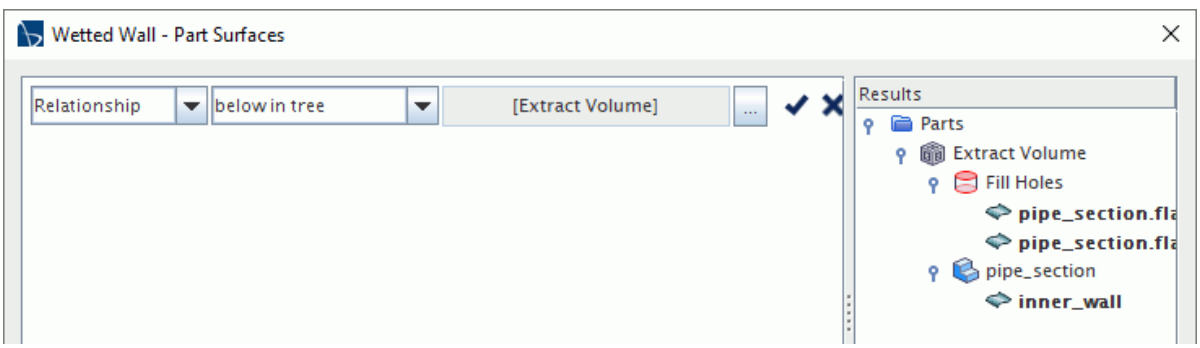
You now convert the input parts properties for boundaries so that they use dynamic queries for assigning part surfaces.

In order to identify correctly the inlet and outlet surfaces for their respective boundary conditions, you must manually apply tags to the correct surfaces within the **Extract Volume** part. This part is the result of the Extract Volume operation. When you apply your template to a new geometry, you also repeat this manual step after running the mesh pipeline.

11. Identify the inlet and outlet faces at the geometry level:
    a) Expand the **Geometry** > **Parts** > **Extract Volume** > **Inherited Parts** > **Fill Holes** node.
    b) Right-click the **pipe_section.flange_0** node and select **Set Tags** > **Inlet End**.
    c) Right-click the **pipe_section.flange_1** node and select **Set Tags** > **Outlet End**.
12. Set the fluid inlet and outlet boundaries to use dynamic queries based on the same tags:
    a) Select the **Regions** > **Fluid** > **Boundaries** > **Inlet** node.
    b) In the right half of the *Part Surfaces* property, click the query editor button ( ).
    c) Set the predicate to `Tag contains Inlet End`.
    d) Select the **Outlet** node and, using the same technique, set the predicate of its dynamic query to `Tag contains Outlet End`.

For the wetted wall boundary, you select all surfaces that are not already applied to other boundaries. To achieve this goal, you make use of the selection priority property for boundaries.

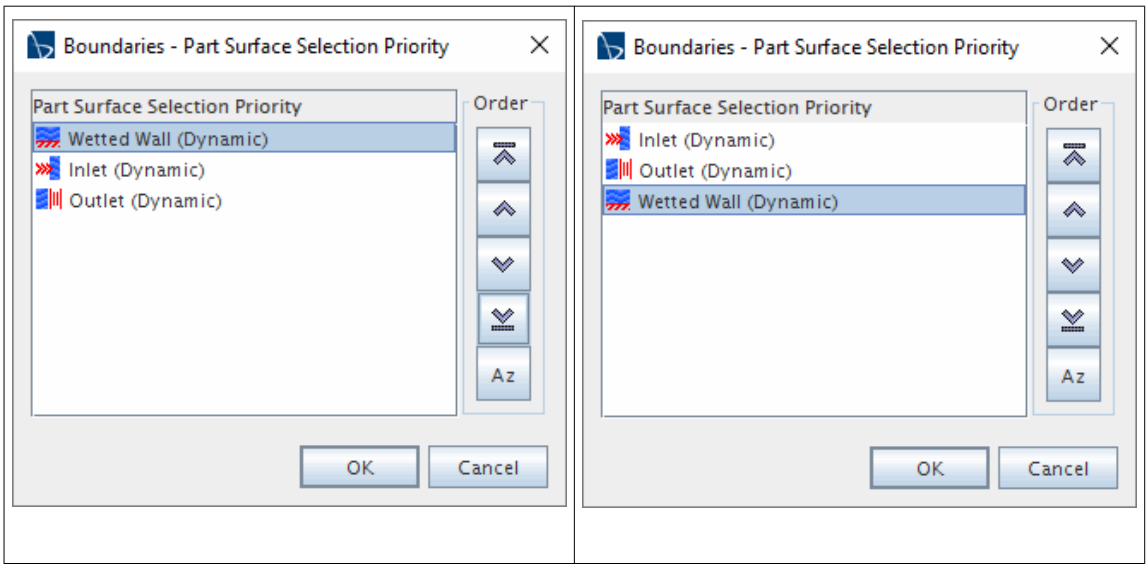
13. To select the remaining surfaces and apply selection priority:
    a) Select the **Wetted Wall** node.
    b) In the right half of the *Part Surfaces* property, click the query editor button ( ).
    c) Set the predicate to `Relationship below in tree [Parts >] Extract Volume`.



    d) Click **OK**.

       At this point, you can observe that all part surfaces are assigned to Wetted Wall. That is, Inlet and Outlet no longer have any surfaces assigned. The following steps rectify this assignment.
    e) Select the **Boundaries** node.
    f) In the *Part Surface Selection Priority* property, click ... (**Custom Editor**).
    g) In the *Boundaries - Part Surface Selection Priority* dialog, change the ordering so that STAR-CCM+ applies part surfaces to the **Wetted Wall** boundary last:
       1. Select **Wetted Wall (Dynamic)** in the list.
       2. Click (**Move to Bottom**) in the right side of the dialog to move that boundary to the bottom of the list.

Before:	After:



3. Click **OK**.

14. Finally, remove the sample part and its mesh, and save the final template file:
    a) Right-click the **Geometry** > **Parts** > **pipe_section** node and select **Delete**. Accept the prompt and continue.
    b) Select **Mesh** > **Clear Generated Meshes**.
    c) Save the simulation.

Usability: Applying the Template Simulation File

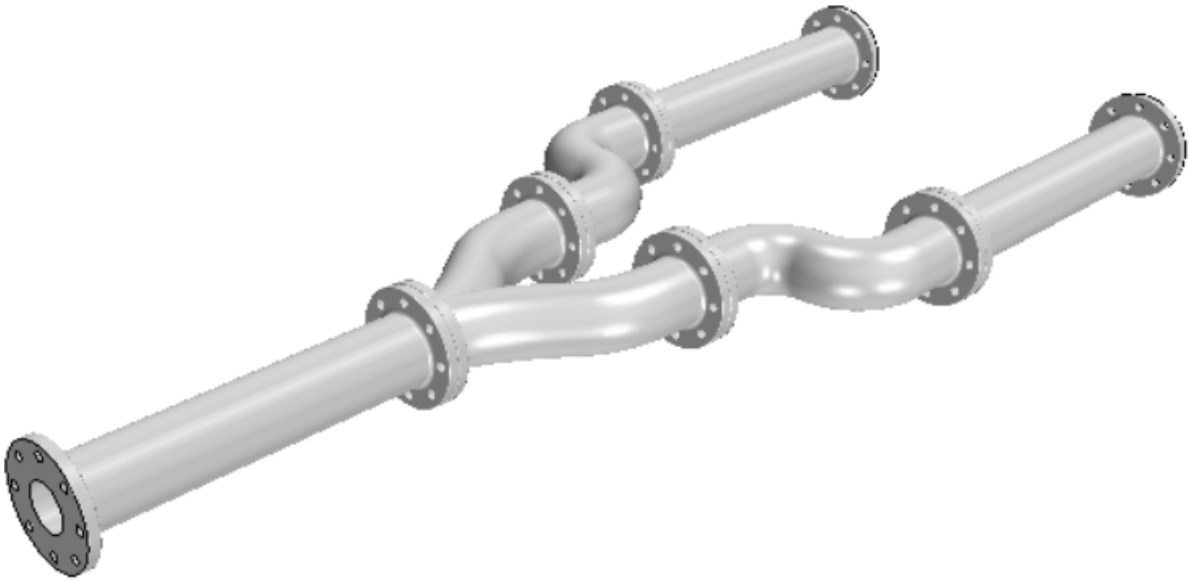
This tutorial demonstrates applying the template simulation file to another geometry.

1. Open the template simulation file created in the last tutorial, `QueryTutorial_TemplateFile.sim`.
2. Save the simulation as `QueryTutorial_MultiplePipes.sim`.

Import the next geometry and visualize it. You are importing a set of connected pipes.

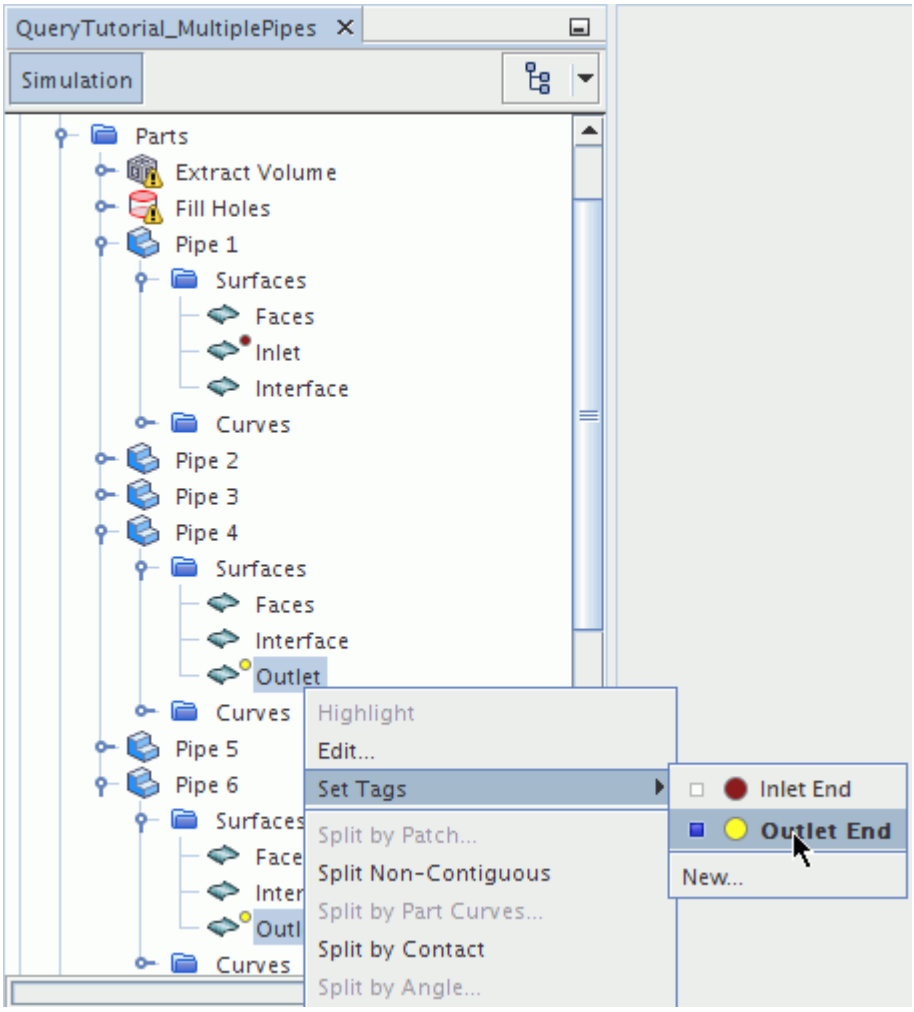
3. Import the connected pipes on which to run the simulation:
    a) Select **File** > **Import** > **Import Surface Mesh**.
    b) Navigate to the surface file `pipeAssy.x_t` in `[INSTALL_DIR]/doc/startutorialsdata/foundationTutorials/data`.
    c) In the *Import Surface Options* dialog, accept the default settings and click **OK**.

    The complete geometry appears in the scene.



Before running the mesh pipeline, you must identify the upstream and downstream faces of the imported assembly. The **Fill Holes** operation requires that these faces be correctly assigned.

4. Apply the inlet and outlet tags on the imported geometry parts:
   a) Expand the **Geometry** > **Parts** > **Pipe 1** > **Surfaces** node.
   b) Right-click the **Inlet** node and select **Set Tags** > **Inlet End**.
   c) Expand the **Pipe 4** > **Surfaces** and **Pipe 6** > **Surfaces** nodes.
   d) Multi-select the **Outlet** nodes in these **Surfaces** nodes, right-click one of them, and select **Set Tags** > **Outlet End**.



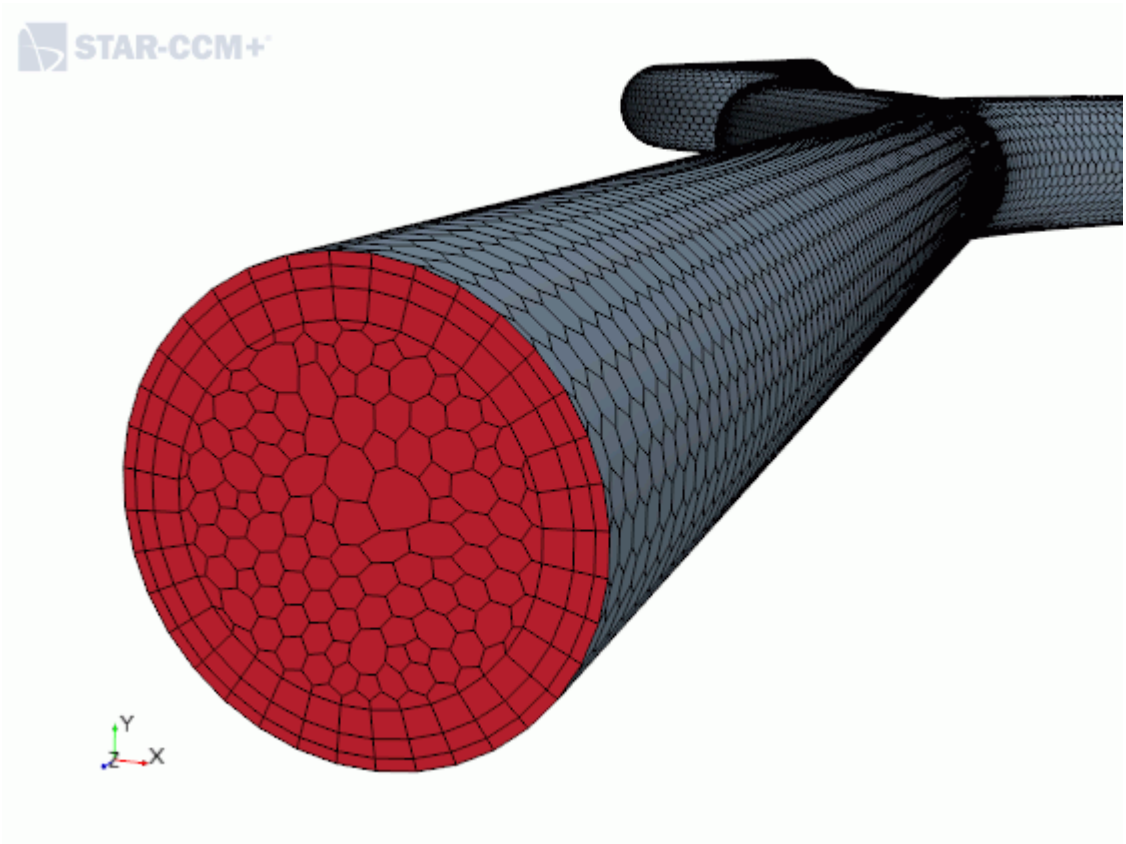
5. To run the mesh pipeline as far as Extract Volume, expand the **Geometry** > **Operations** node, right-click the **Fill Holes**, **Imprint**, and **Extract Volume** nodes in turn and select **Execute** for each.

You must assign the boundary conditions before running the Automated Mesh operation, as otherwise prism layers are generated across the inlet and outlet faces. By assigning boundaries first, the prism layer mesher only generates layers on the pipe walls.

6. To assign the boundary conditions, apply the **Inlet End** and **Outlet End** tags to the respective faces within the **Extract Volume** part:
   a) Expand the **Parts** > **Extract Volume** > **Inherited Parts** > **Fill Holes** node.
   b) To the **Pipe 1.Inlet** node, apply the tag, **Inlet End**.
   c) To the **Pipe 4.Outlet** and **Pipe 6.Outlet** nodes, apply the tag, **Outlet End**.
7. To generate the volume mesh, right click the **Geometry** > **Operations** > **Automated Mesh** node and select **Execute**.
8. To review the boundary assignments of the parts, expand the **Regions** > **Fluid** > **Boundaries** node:
   - The **Inlet** boundary has one part surface assigned to it.
   - The **Outlet** boundary has two part surfaces.
   - The **Wetted Wall** boundary contains the rest of the parts.

These assignments of part surfaces are correct because you prioritized the selection of them in the template simulation.

9. To visualize the mesh, create a mesh scene.



Since the physics models were prepared in the template, you are ready to run the simulation.

10. Click (**Run**) in the top toolbar.

    The Residuals display is created automatically and shows the progress of the solver.

11. You can evaluate the design with scalar or vector scenes, or plots.
12. Save the simulation.

Geometry

The tutorials in this set illustrate various STAR-CCM+ features for creating and working with parts and 3D-CAD.

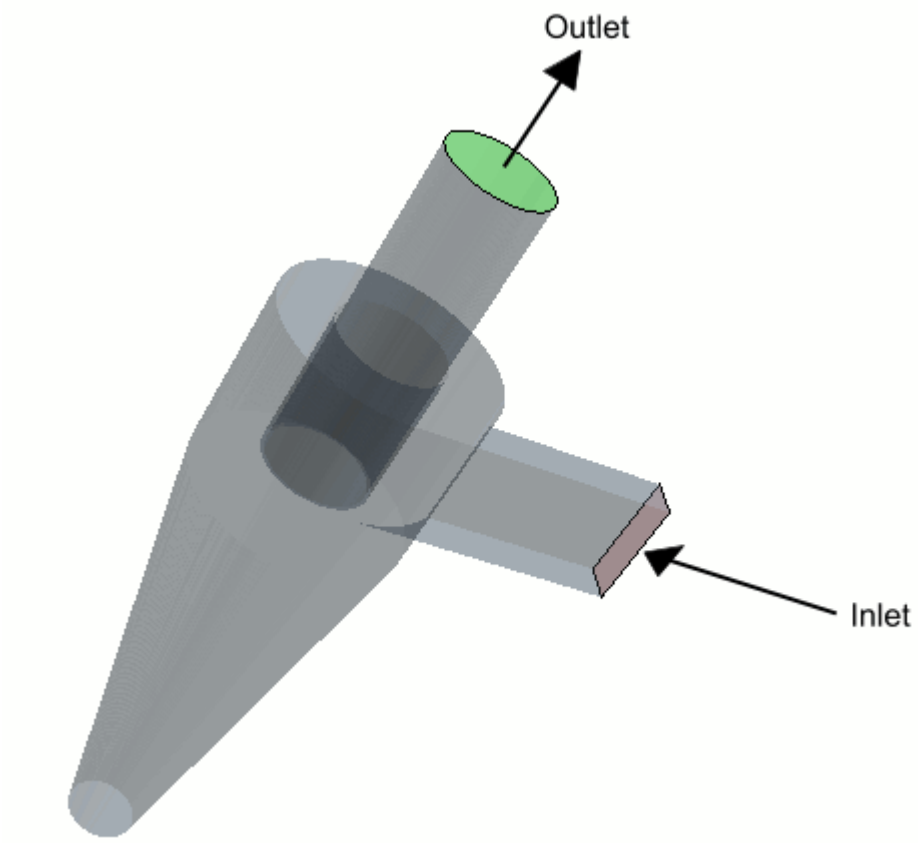
Contents:

3D-CAD: Cyclone Separator
Geometry Parts: CAD Assembly

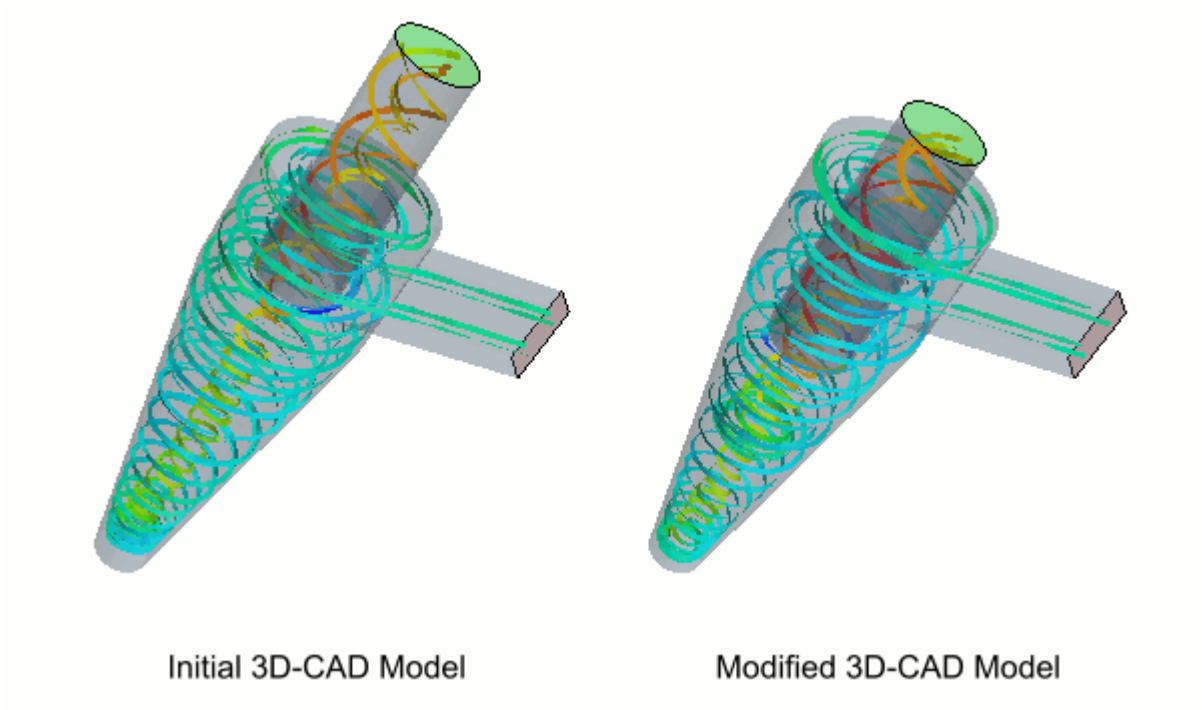
3D-CAD: Cyclone Separator

This tutorial demonstrates how to build and run a simulation entirely within STAR-CCM+.

You create the geometry for this simulation in 3D-CAD, the modeler available in STAR-CCM+. You then convert the 3D-CAD model to a part and perform a full simulation. The geometry in this simulation is a simplified cyclone separator.



While creating the geometry in 3D-CAD, you set up a design parameter to allow you to modify the geometry from the main simulation tree. The purpose of this simulation is to analyze the effect on the flow pattern when placing the outlet pipe at two different levels within the cyclone chamber.



The following basic concepts of 3D-CAD are introduced in this tutorial:

- Creating a 3D-CAD model
- Creating sketches and applying constraints and dimensions
- Revolving and extruding sketches
- Naming faces
- Using a 3D-CAD model in a simulation
- Modifying the model geometry and rerunning the case

Contents:

Prerequisites
Creating the 3D-CAD Model
Creating a Geometry Part
Assigning a Part to a Region
Selecting Physics Models
Generating the Volume Mesh
Preparing a Scalar Scene
Running the Simulation
Creating a Streamlines Scene
Modifying the Geometry and Rerunning
Summary

Prerequisites

The instructions in the 3D-CAD tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Techniques	Associated Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

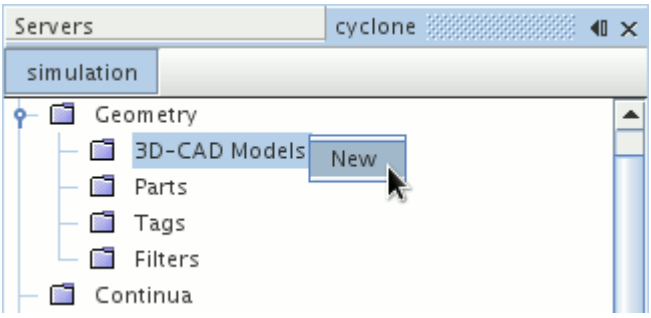
Creating the 3D-CAD Model

Create a 3D-CAD model.

1. Launch STAR-CCM+.
2. Save the simulation under the name `cyclone.sim`.

Launch 3D-CAD and create the geometry:

3. Right-click **Geometry** > **3D-CAD Models** and select **New**.



The 3D-CAD environment launches and a 3D-CAD model is created. Within 3D-CAD, an object tree that is specific to the 3D-CAD model is displayed. This tree contains **Bodies**, **Features**, and **Design Parameters** nodes. A *3D-CAD View* scene is also created automatically. Initially, this scene displays three starting sketch planes which are used as a basis for adding features to the model.

4. Rename the **3D-CAD Model 1** node to `Cyclone`.

Contents:

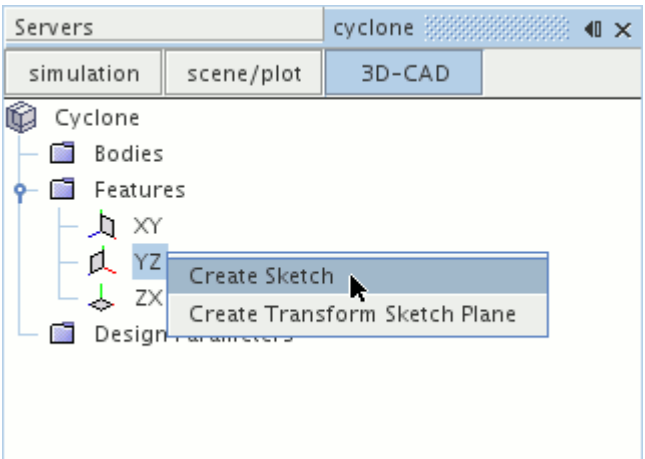
Sketching the Cyclone Profile
Revolving the Cyclone Profile
Creating the Outlet Pipe
Creating the Inlet Duct
Creating a Design Parameter
Specifying Inlet and Outlet Faces

Sketching the Cyclone Profile

Create a sketch that contains the profile of the cyclone chamber.

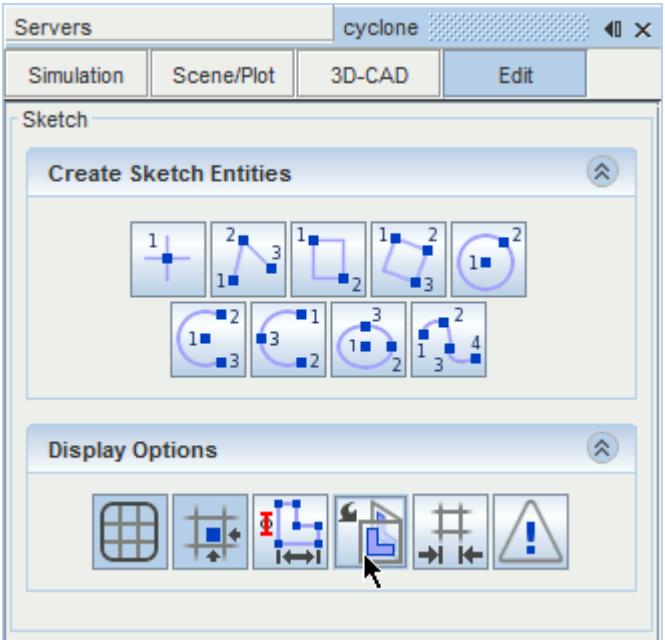
Every 3D-CAD model contains three default sketch planes: **XY**, **YZ**, and **ZX**. Create a sketch on the **YZ** sketch plane:

1. Right-click the **Features** > **YZ** node and select **Create Sketch**.

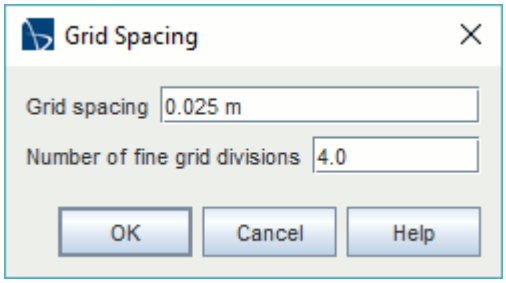


Creating a sketch activates sketch mode, in which a grid is displayed on the sketch plane in the *3D-CAD View* scene, and the *Sketch* panel opens. This panel contains the tools for creating sketch entities, such as lines, circles, and arcs. The panel also allows you to control the settings for the grid that is displayed in the scene. Each sketch plane has local X and Y axes relative to the position and orientation of the plane in relation to the global coordinate system.

2. Click (**View normal to sketch plane**) to align the sketch plane with the plane of the screen.



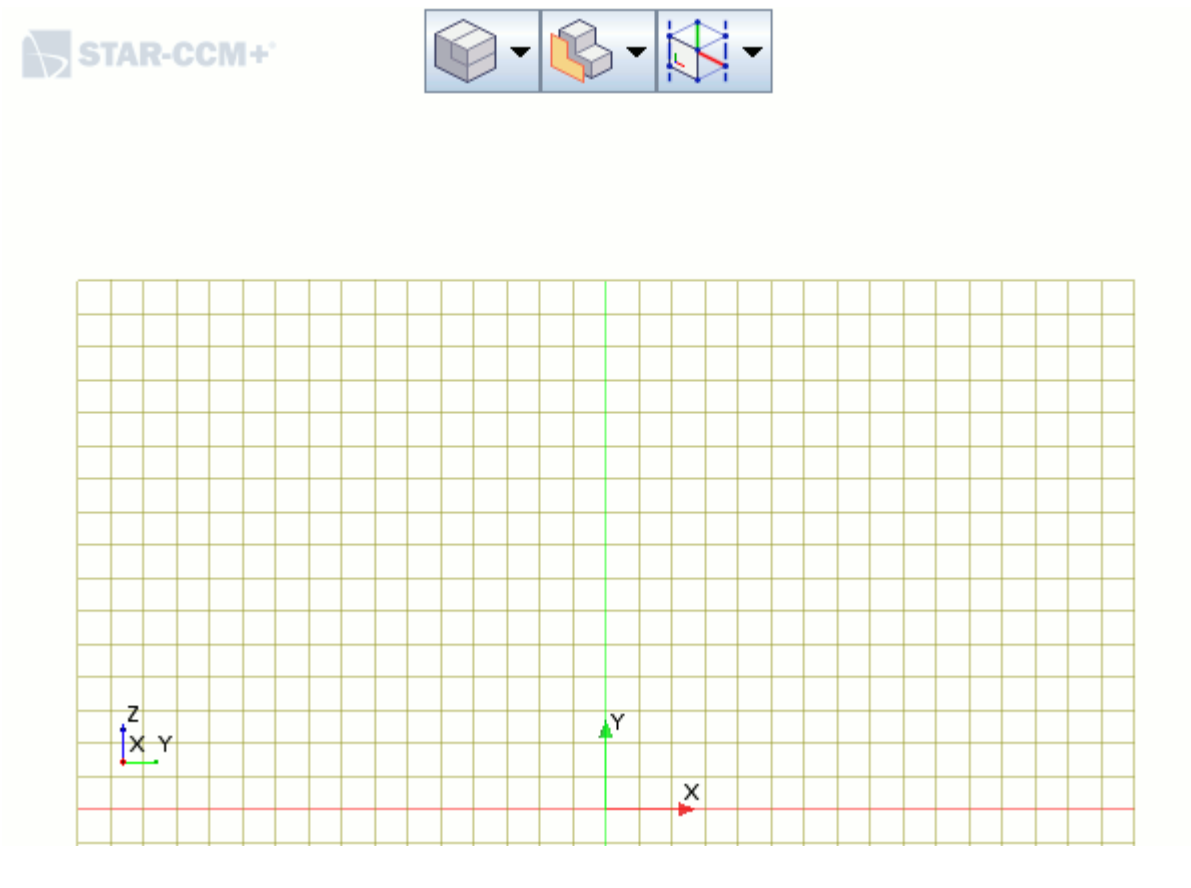
3. Click (**Set sketch grid spacing**) to specify the distance between the grid lines on the sketch plane so that the spacing is suitable for the dimensions of the model geometry.
4. In the *Grid Spacing* dialog, set *Grid Spacing* to `0.025 m`.



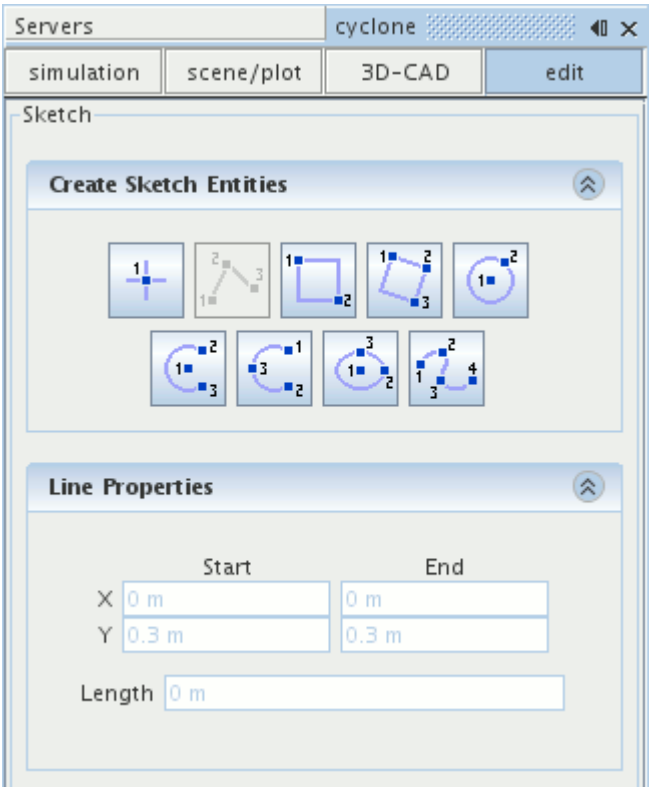
5. Click **OK**.

Use the mouse to adjust the position, orientation, and zoom level of the sketch plane.

6. In the *3D-CAD View* scene:
   - Left-click and drag the mouse in any direction to rotate the view.
   - While holding down the <`Ctrl`> key, click and drag to roll the view.
   - Right-click and drag to pan the view.
   - Middle-click and drag to zoom in or out. Alternatively, if your mouse has a scroll wheel, turn the mouse wheel to adjust the zoom level.
7. Click (**View normal to sketch plane**) to realign the sketch plane with the screen.
8. Adjust the view in the *3D-CAD View* scene so that the top half of the grid is shown. Leave some white space above the grid as the sketch will cover some of this space.



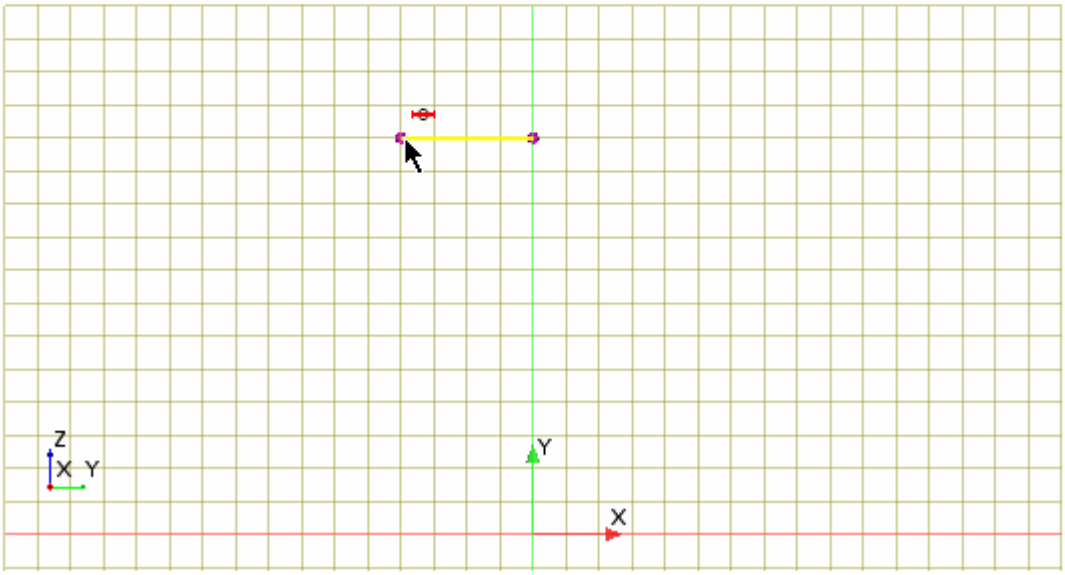
9. Click (**Create line**) and move the mouse over the grid in the *3D-CAD View* scene.
   The mouse pointer has a purple dot that is attached to it, which indicates the position of the point on the grid. The coordinates of this point are displayed in the *Line Properties* box of the *Sketch* panel, and change as you move the mouse. The (**Snap to Grid**) option is enabled by default, so the point jumps to an intersection between two grid lines as it is approaches them.
10. Position the point at `[0.0, 0.3] m`.



11. Click to fix the point in position.

    Create the first set of lines with approximate lengths. To exit the tool while creating a sketch entity, press the <Esc> key. If you want to reposition any sketch entities while you are sketching, exit the tool first, then click-and-drag the relevant sketch entities to the desired position. To resume sketching, simply select the appropriate tool from the *Create Sketch Entities* box and continue.

12. Move the mouse pointer to the left. A preview is shown to indicate the position of the line. The yellow color of the line indicates that the line is horizontal. Position the mouse so that the line extends horizontally left of the starting point and click the left mouse button.

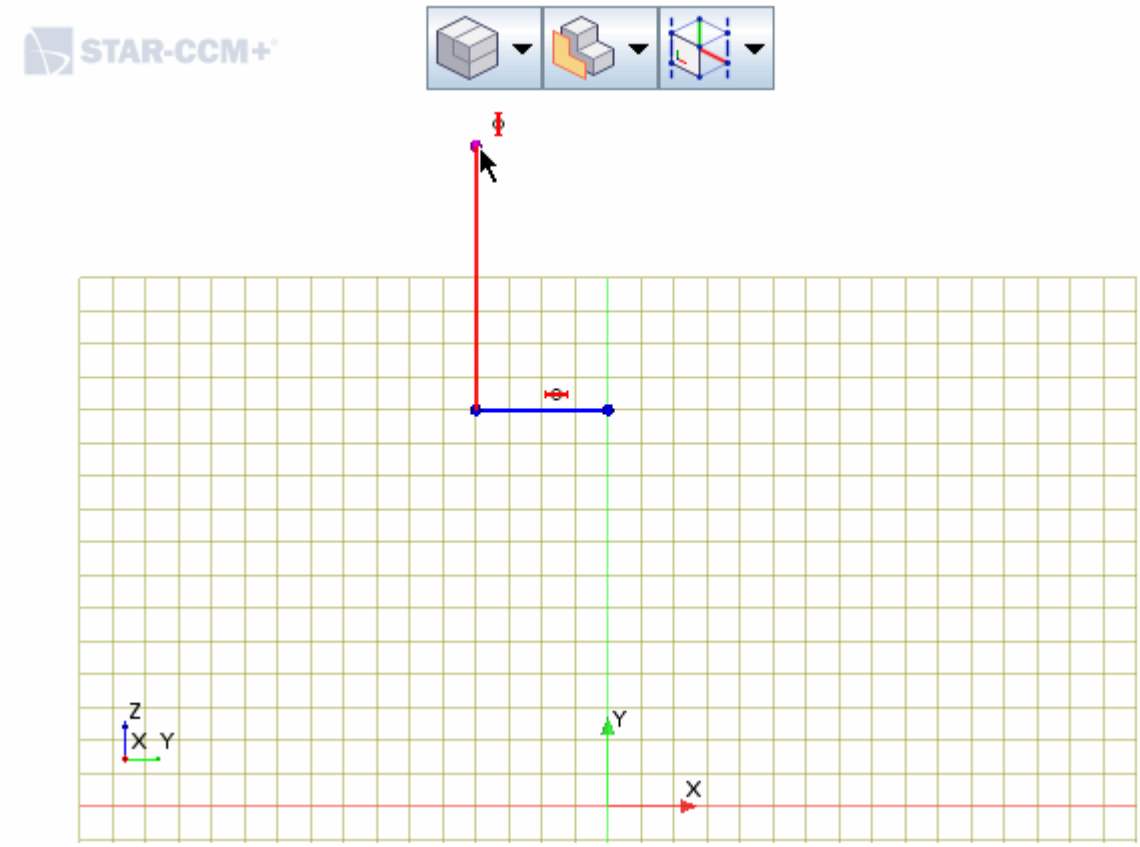


The **Line** tool creates continuous lines, so the end point of the first line is the start point of the next. The coordinates that are shown in the *Line Properties* box are automatically updated as each line is defined.

A horizontal constraint is added to the line, indicated by the glyph in the scene. Constraints control the position of a sketch entity.

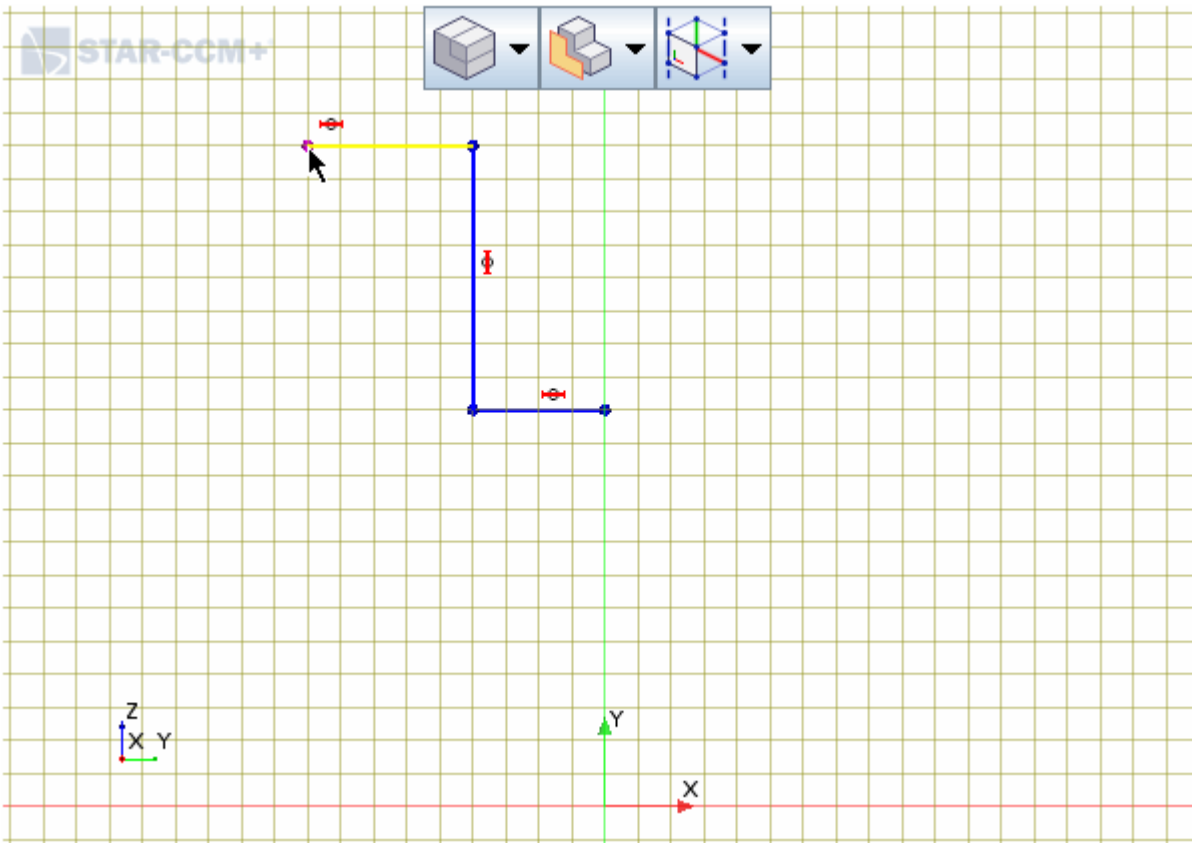
13. Move the mouse pointer into the white space above the grid and notice that it still snaps to grid intersections even though they are not visible.
14. Position the mouse so that the second line is vertical and ends in the white space above the grid.

    The line changes to a red color when it is vertical. You can adjust the zoom level while you are in the process of creating sketch entities.
15. Left-click to fix the end point of the second line.

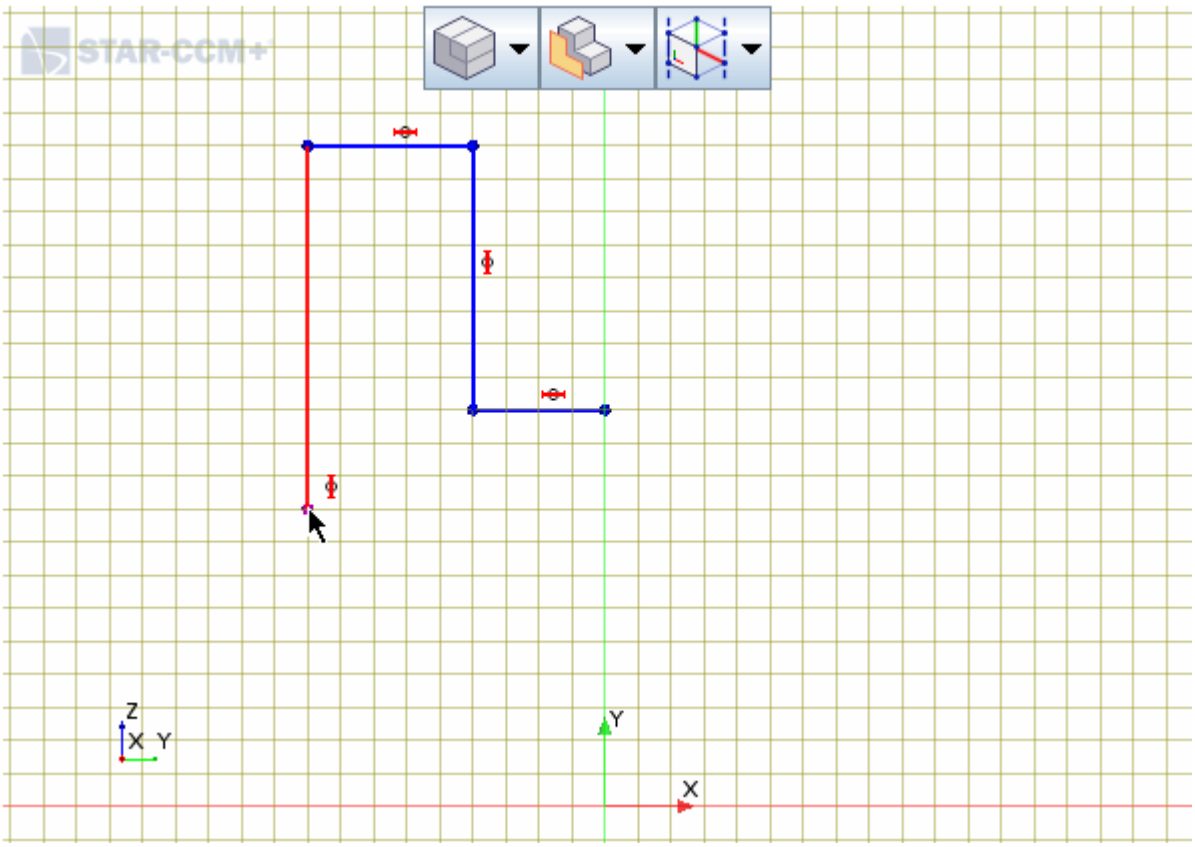


When you draw sketch entities outside of the grid that is shown in the *3D-CAD View* scene, the grid automatically expands.

16. Create a short horizontal line that extends to the left of this point and left-click to fix the end point in position.



17. Move the mouse vertically downwards to create a longer vertical line and click to fix the point.



18. Press the <Esc> key to exit the line creation tool.

Contents:

Adding Constraints and Dimensions
Adding a Construction Line

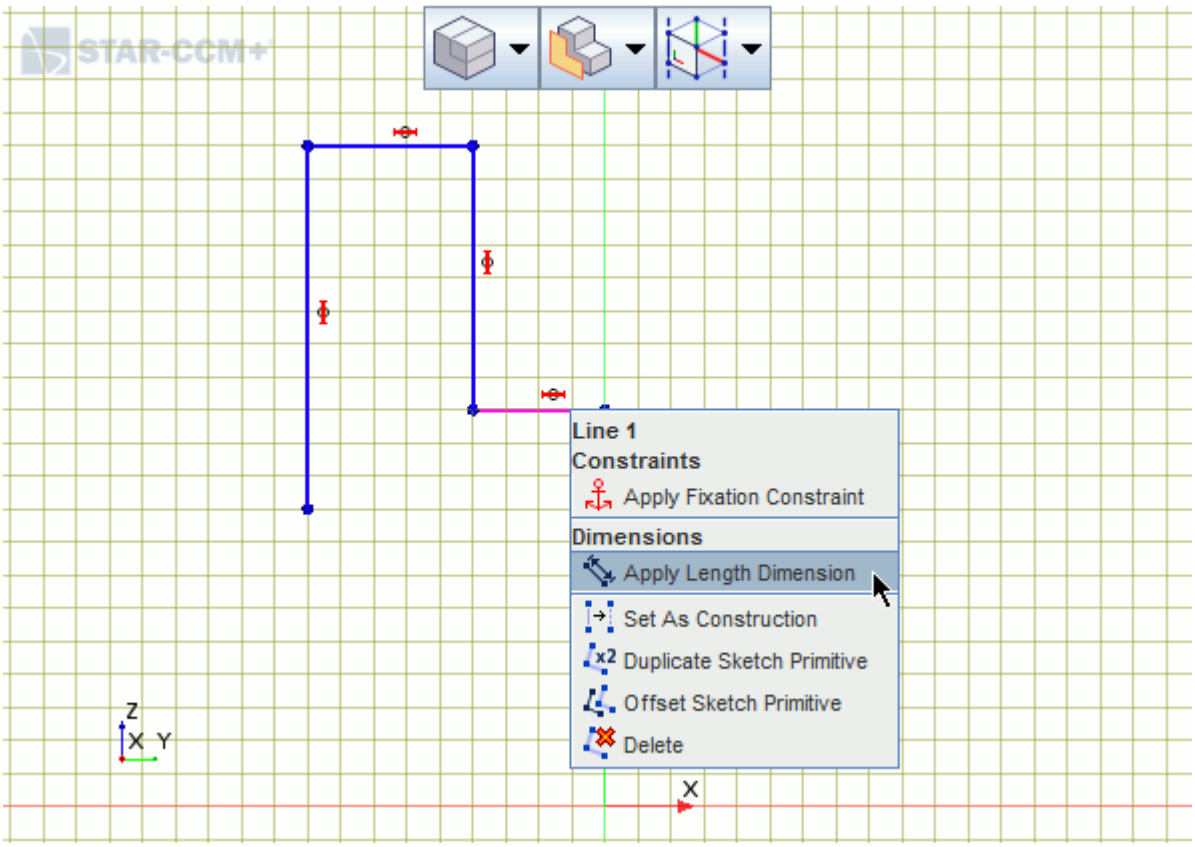
Adding Constraints and Dimensions

Constraints and dimensions are applied to the sketch to control the position and size of the sketch entities on the sketch plane.

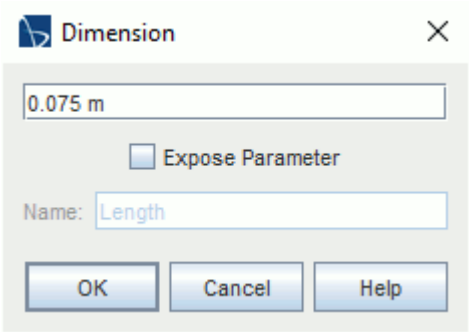
When sketching, typically you apply constraints first to maintain the current orientation of the lines. Following this, you apply dimensions to define the length of each line. When drawing lines in the horizontal or vertical direction, constraints are automatically added for you.

Set the length of each line by applying dimensions:

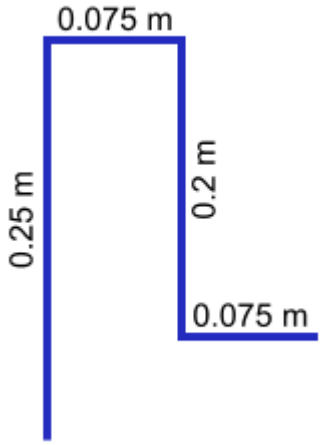
1. Right-click the first horizontal line and select **Apply Length Dimension**.



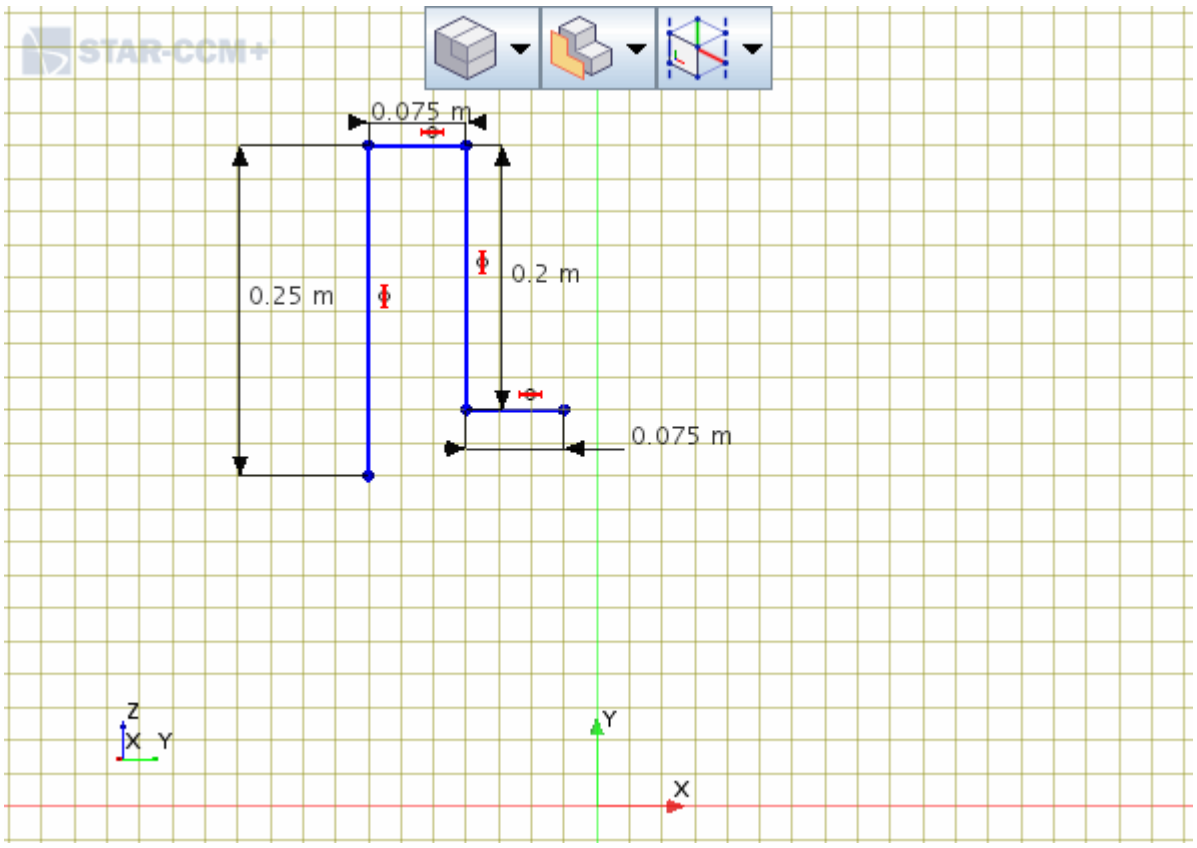
2. In the *Dimension* dialog, enter a value of `0.075 m`.



3. Click **OK**.
4. Repeat the previous two steps for the remaining lines. Use the values that are displayed in the following sketch:



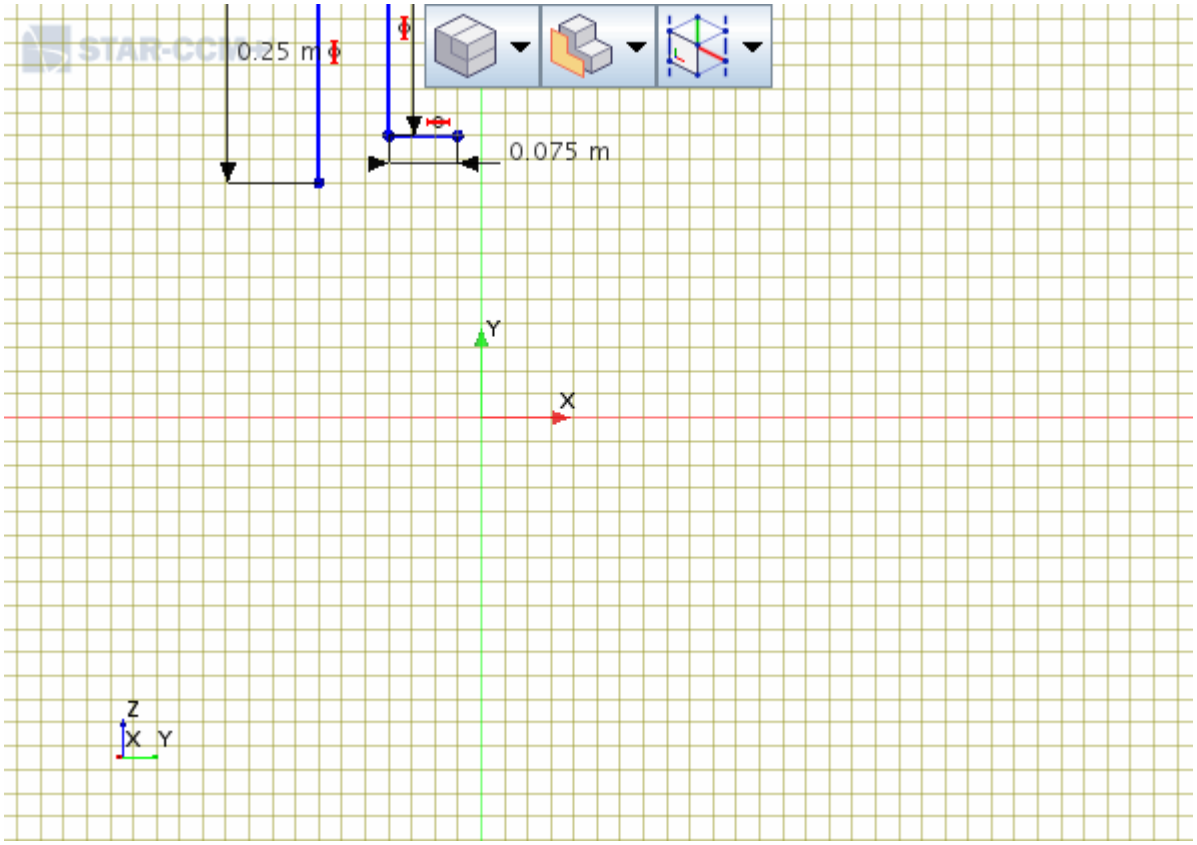
The sketch appears as follows:



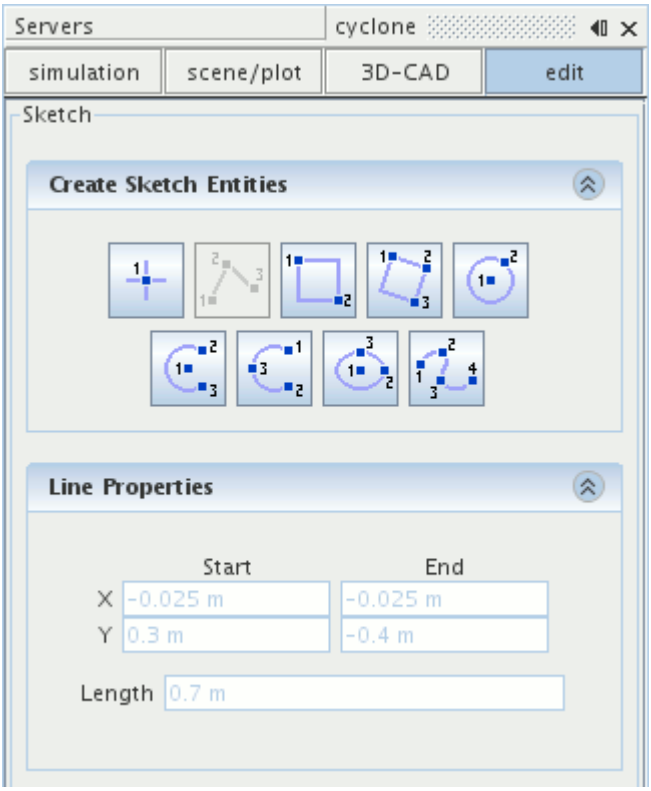
Notice that when you select each line, a list of the constraints and dimensions that are applied to that line is displayed in the *Constraints/Dimensions* box in the *Line Properties*.

Draw the remainder of the cyclone profile, and use some additional constraints to control its position on the sketch plane:

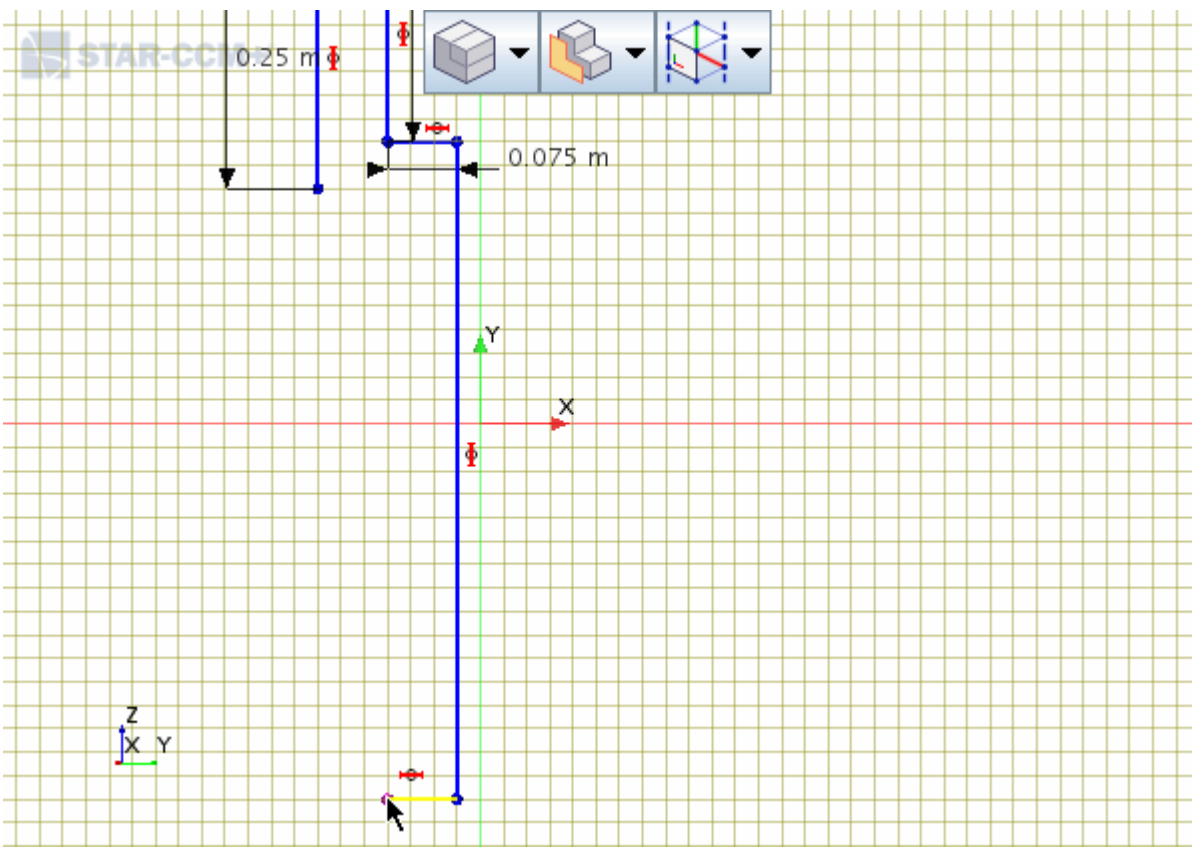
5. Pan the view down and adjust the zoom level so that the 3D-CAD View scene appears as follows:



6. Click (**Create line**) and move the mouse to the starting point for the previous set of lines.
   The mouse automatically snaps to existing sketch entities within the sketch, so when the blue square is close to the point, it jumps onto that point.
7. Click the top of the existing point.
8. Move the mouse vertically downward until a line *Length* of `0.7 m` is shown in the *Line Properties* box.
   Left-click to fix the point at the grid intersection.

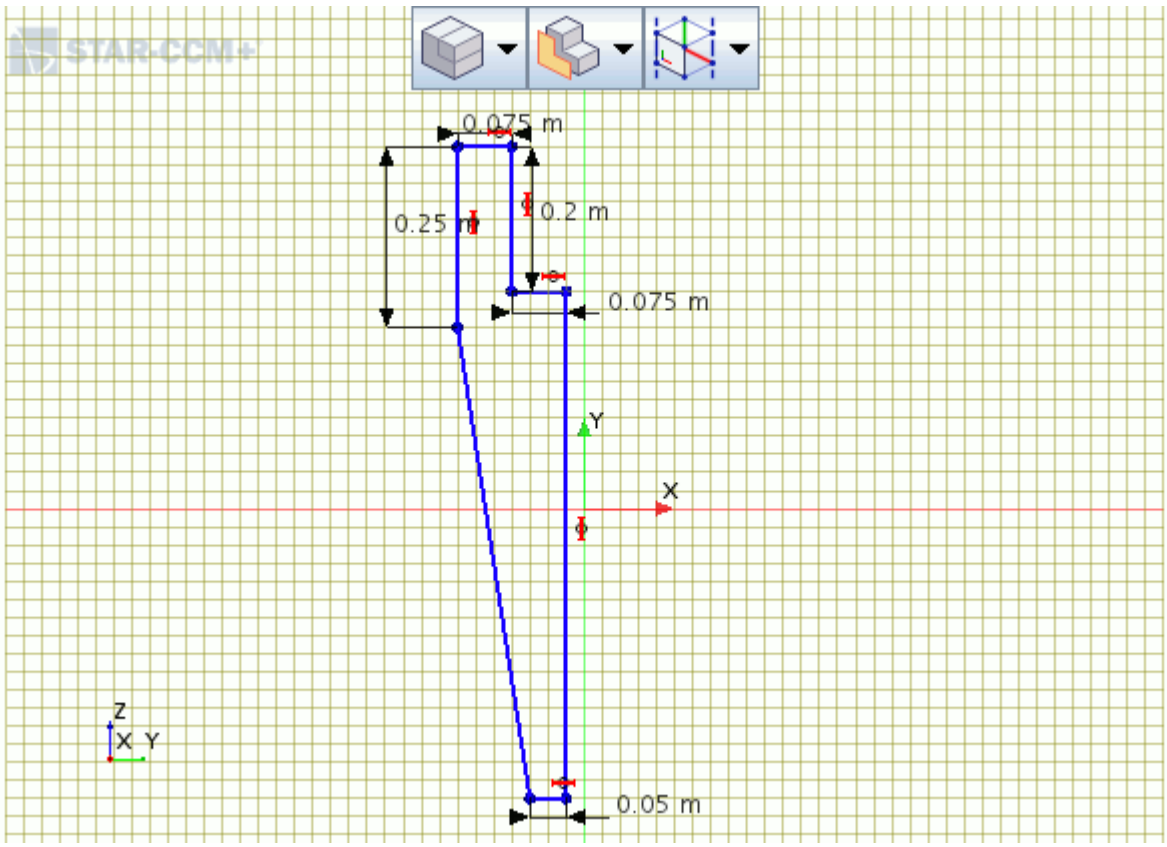


9. Move the mouse to the left and create a short line that extends horizontally from this point.



Complete the profile by creating a diagonal line to define the cone shape of the cyclone:

10. Move the mouse pointer to the free end of the first set of lines and click to close the sketch profile.
11. Right-click the horizontal line at the bottom of the sketch and select **Apply Length Dimension**.
12. In the *Dimension* dialog, enter a value of `0.05 m` and click **OK**.



Move the axis of the cyclone so that it is aligned with the Y-axis of the sketch plane. To do this you can drag-and-drop the sketch.

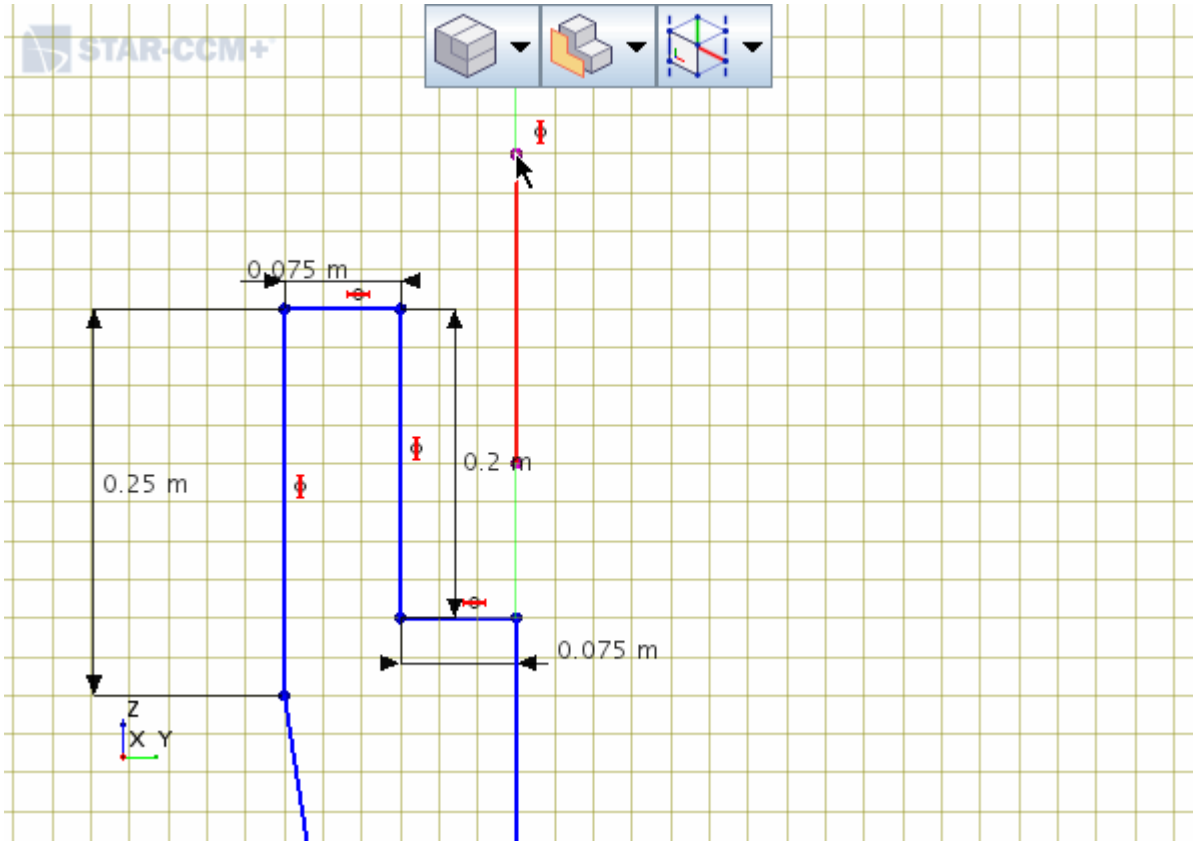
13. If necessary, adjust the zoom level. Left-click and hold at any point on the longest vertical line, drag the mouse pointer onto the origin and release the mouse button.

Adding a Construction Line

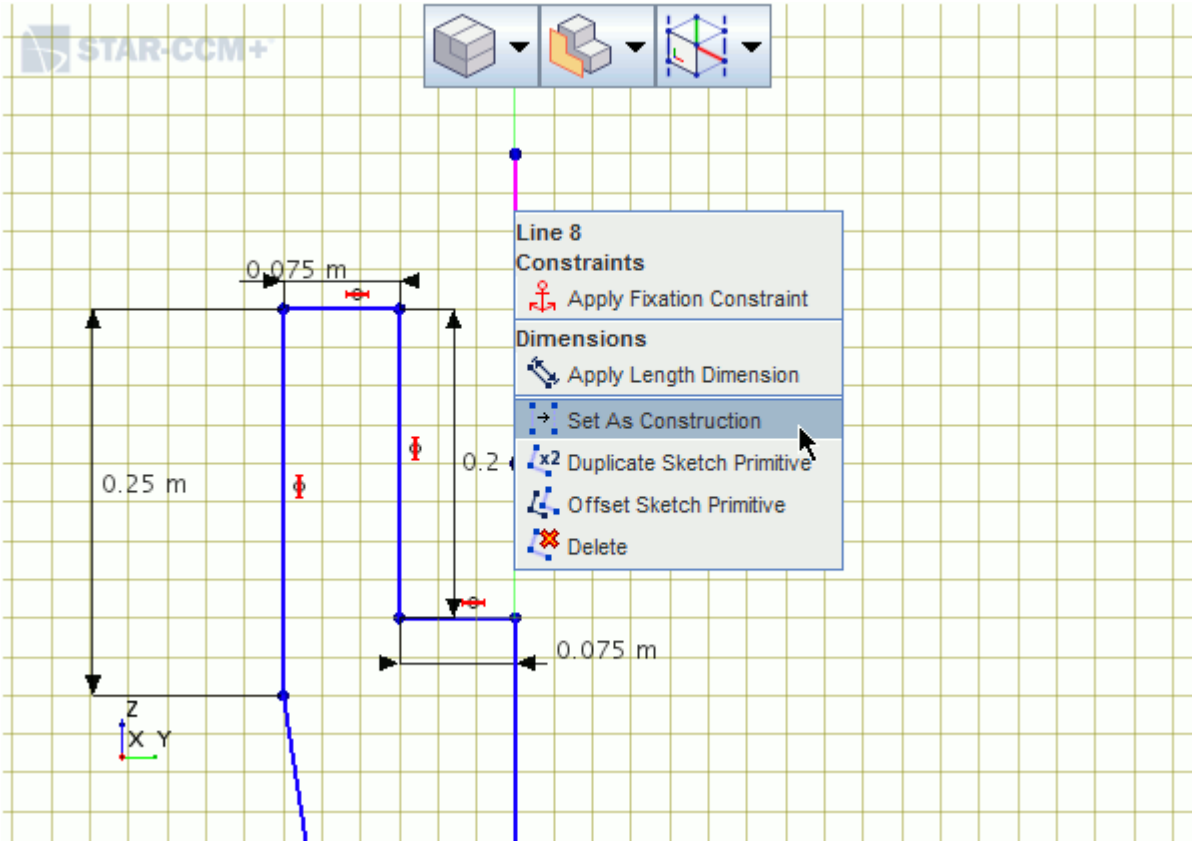
A construction line defines the axis around which the cyclone profile revolves.

Add a construction line to the profile sketch:

1. Click (**Create line**) and create a line that is aligned with the Y-axis as shown below.



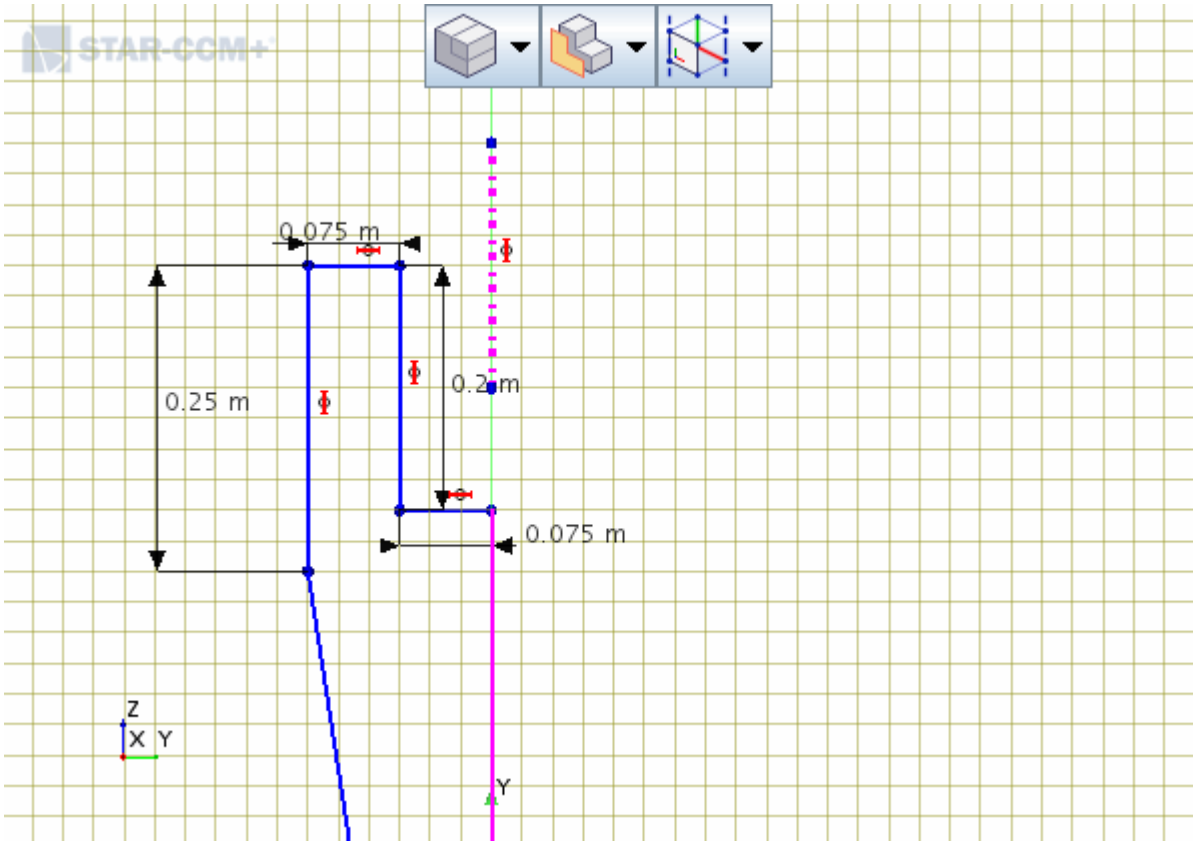
2. Press <`Esc`> to exit the line creation tool.
3. Right-click the line that you just added and select **Set As Construction**.



The appearance of the line changes from solid to dashed to indicate that it is now a construction line. Upon exiting the sketch, this line is no longer visible, as construction lines are only shown when you are creating or editing a sketch.

Apply a constraint between the construction line and the vertical line at the center of the cyclone to make sure that they are collinear:

4. Select the construction line, hold down the <`Ctrl`> key and select the long vertical line on the Y-axis.
5. Right-click either of the lines and select **Apply Collinear Constraint**.



The cyclone profile is now complete and is ready to revolve.

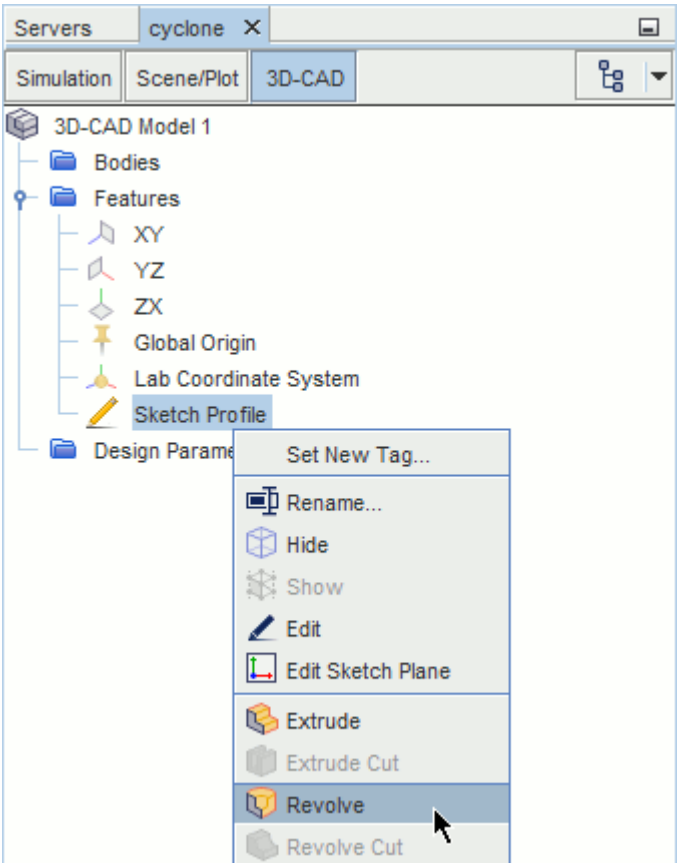
6. Click **OK** at the bottom of the *Sketch* panel to exit sketch mode.

Revolving the Cyclone Profile

Right-clicking a sketch in the feature tree lets you perform various CAD operations, such as revolving a profile around an axis.

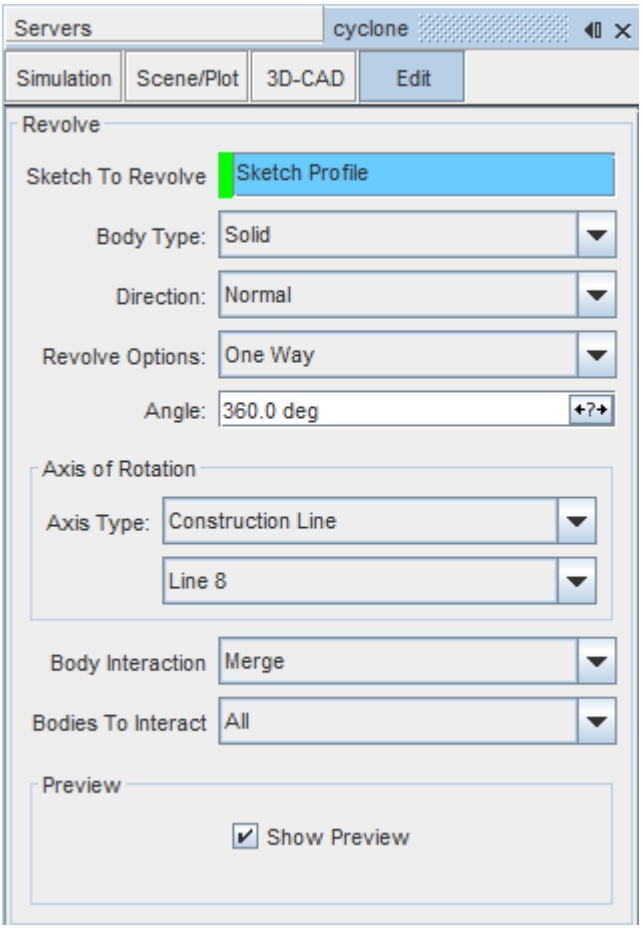
Revolve the sketch of the cyclone chamber profile:

1. Rename the **Sketch 1** node to `Sketch Profile`.
2. Right-click **Sketch Profile** and select **Revolve**.

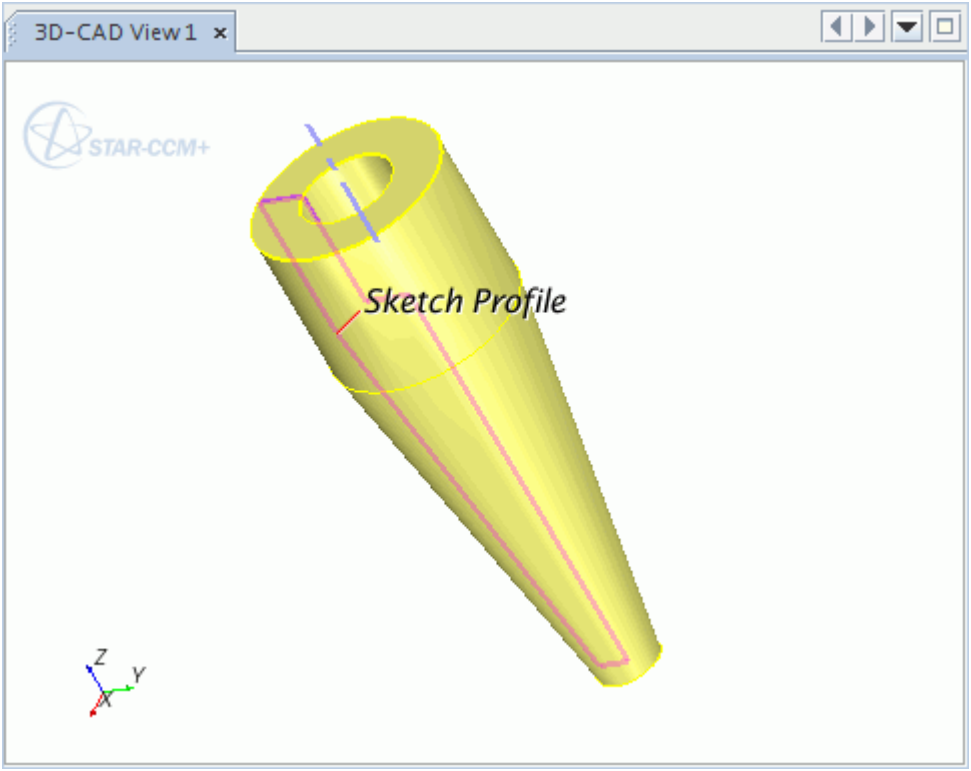


The *Revolve* panel lets you specify the parameters for the revolved feature. The *Direction* option defines the direction in which the sketch is revolved. The *Body Interaction* option specifies how the body that is created from the revolution interacts with any other bodies in the model. The *Interacting Bodies* option specifies the bodies which interact with the revolved body. In this case, the sketch is revolved through 360 degrees and no other bodies are present. Therefore the *Direction* and *Body Interaction* settings do not affect the outcome and can be left as the default. The *Axis* option lets you choose which axis to use as the axis of revolution. In this case, the sketch only contains one construction line, **Line 8**, and is selected by default. A blue preview of the selected axis is shown in the *3D-CAD View* scene.

3. In the *Revolve* panel, set *Angle* to `360.0 deg` and press <`Enter`>.

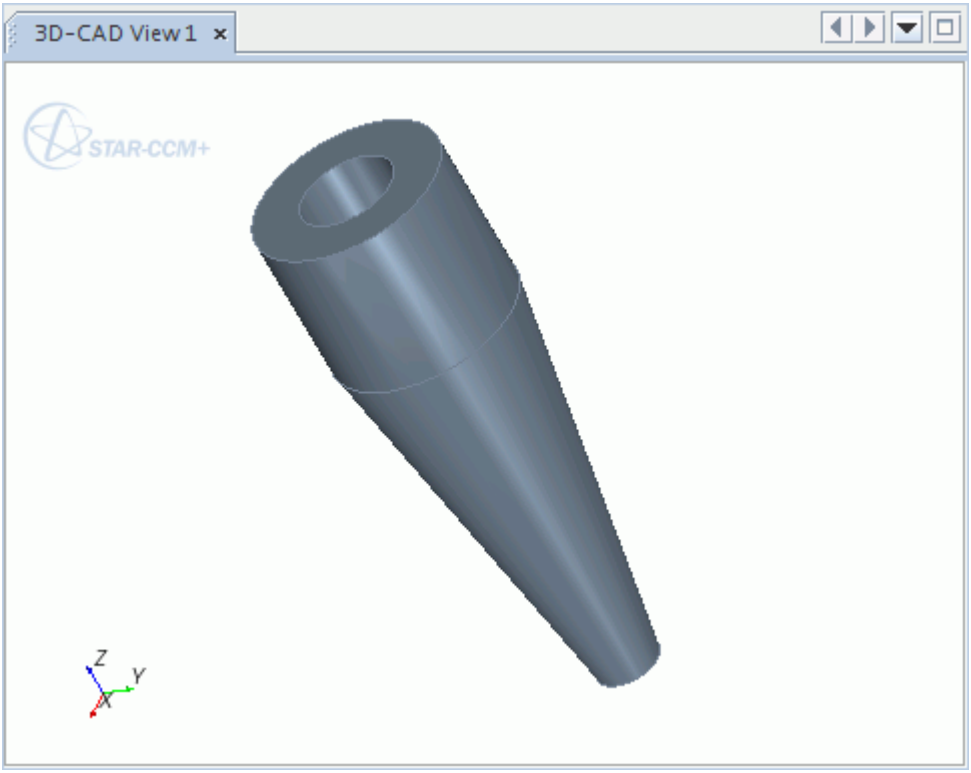


A preview of the feature is shown in the *3D-CAD View* scene.



4. Click **OK** to confirm the settings and create the revolved feature.

   A **Revolve 1** feature node is added to the feature tree, and the solid body is displayed in the *Graphics* window. Bodies that form a 3D-CAD model are shown under the **Bodies** node.



5. Expand the **Bodies** node and select **Body 1**.

   The corresponding body is highlighted in the *3D-CAD View 1* scene.

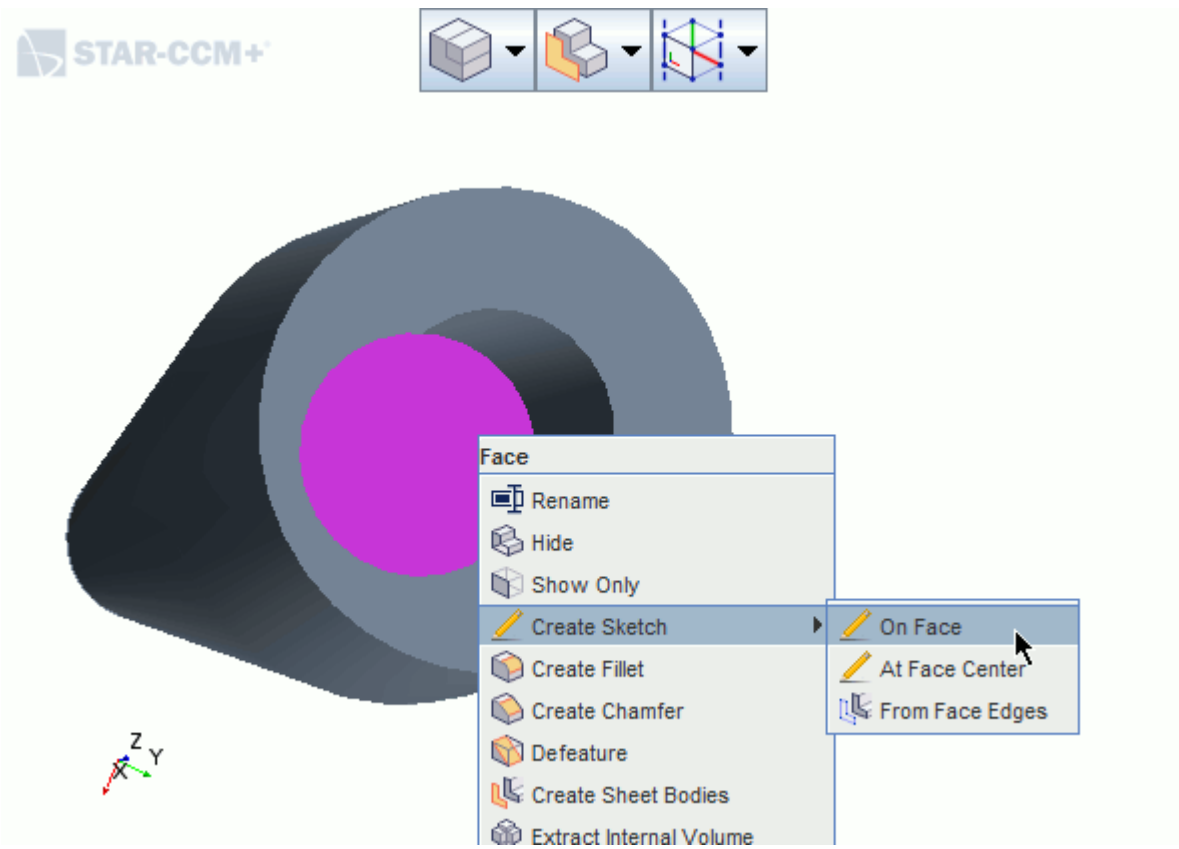
6. Rename the **Body 1** node to `Fluid`.

Creating the Outlet Pipe

The outlet pipe has a circular cross-section and is created by sketching a circle and extruding it. The circle that is sketched is slightly smaller in diameter than the hole in the top of the cyclone, and the difference between the two is the thickness of the outlet pipe.

Create the sketch for the outlet pipe on an existing face:

1. Rotate the model in the *3D-CAD View 1* scene so that you can see the sunken face in the top of the model.
2. Right-click the sunken face and select **Create Sketch** > **On Face**.

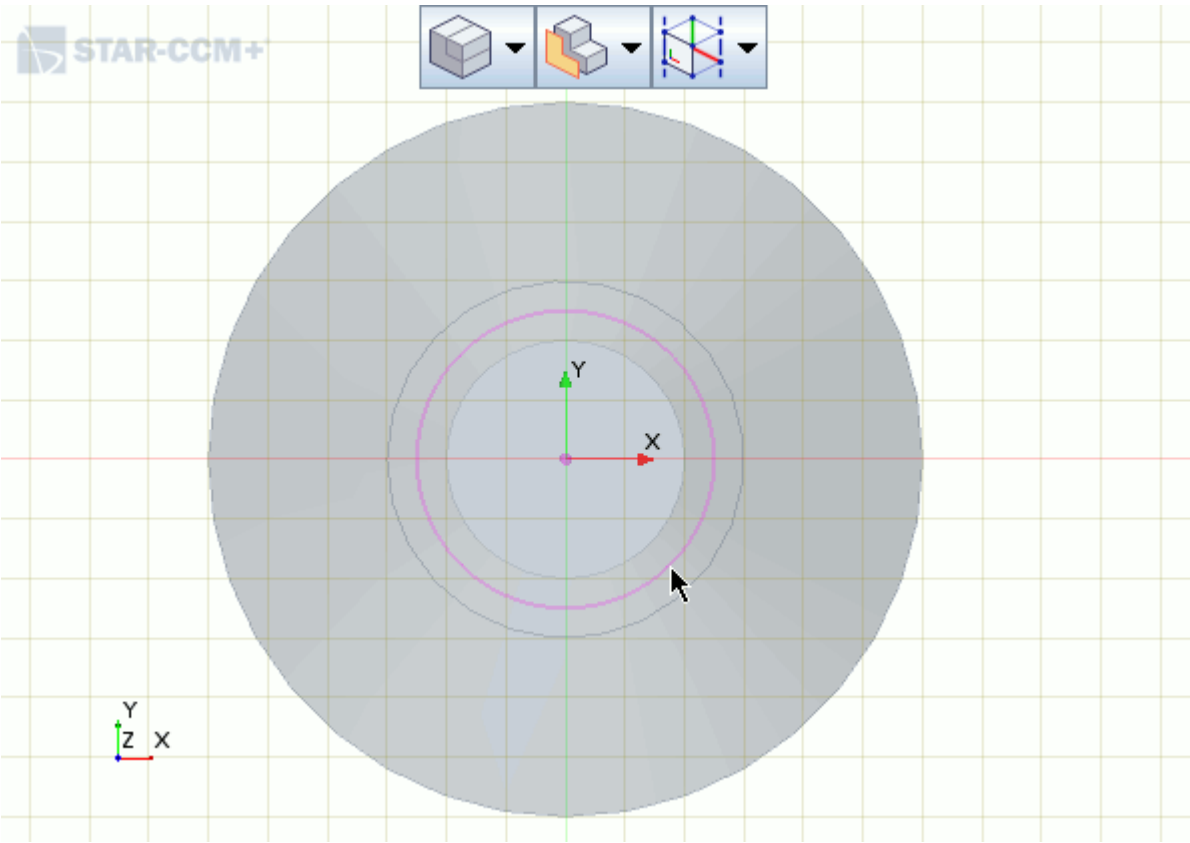


   A sketch plane that is defined by the selected face is displayed on the model in the *3D-CAD View* scene.

3. Click (**View normal to sketch plane**).
4. To see the hole, in the *Vis* toolbar, click (**Make scene transparent**).
5. Click (**Create circle**) in the *Create Sketch Entities* box and click at `[0.0, 0.0] m`.

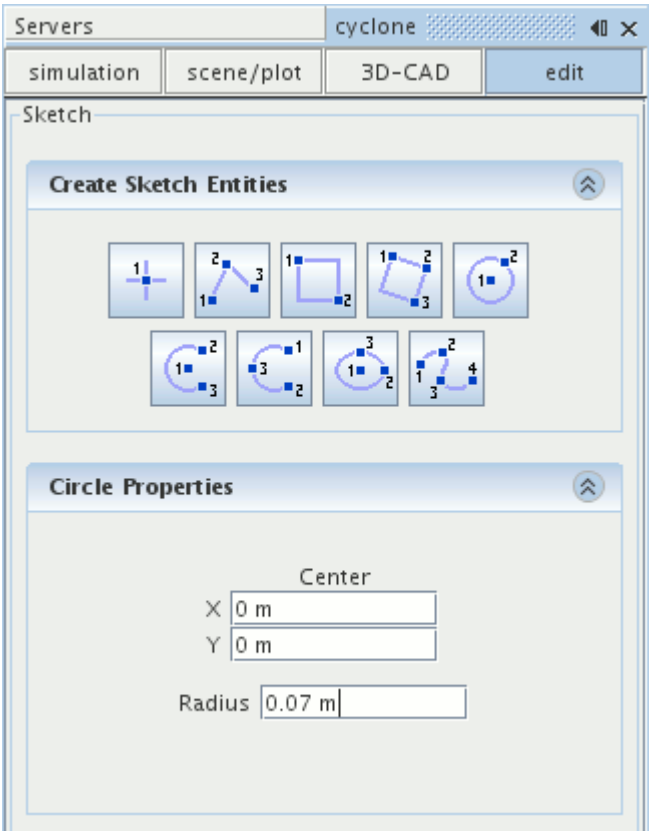
The first click positions the circle at the origin of the sketch. The second mouse click lets you specify the radius of the circle.

6. Move the mouse so that the radius is approximately `0.06 m`, using the *Circle Properties* box for reference, and click.



You can define the circle radius more accurately by entering an exact value in the *Circle Properties* box.

7. In the *3D-CAD View 1* scene, click on the circle.
8. Set *Radius* to `0.07 m`



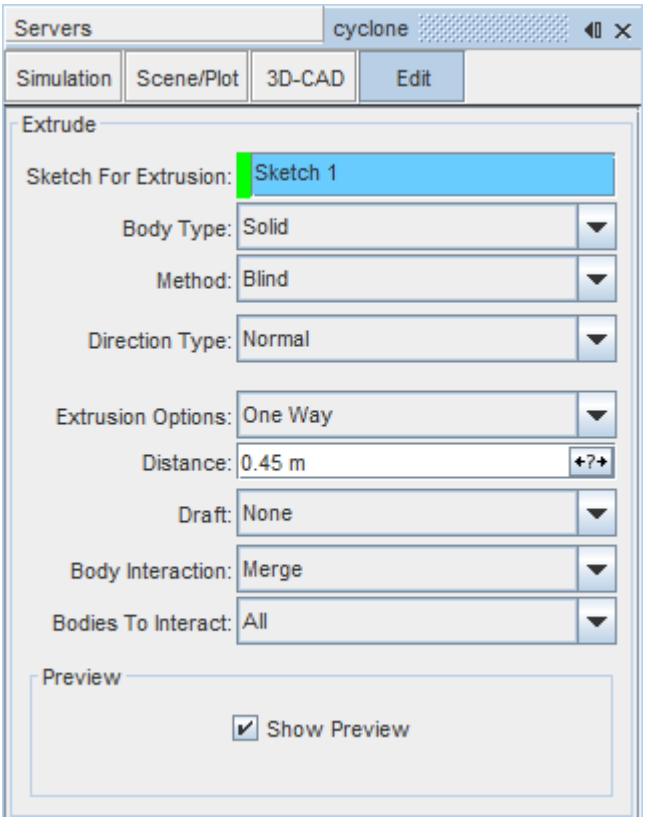
9. Click **OK** to exit the sketch.

10. In the *Vis* toolbar, click (**Make Scene Transparent**).

Extrude the sketch to form the outlet section of the cyclone geometry:

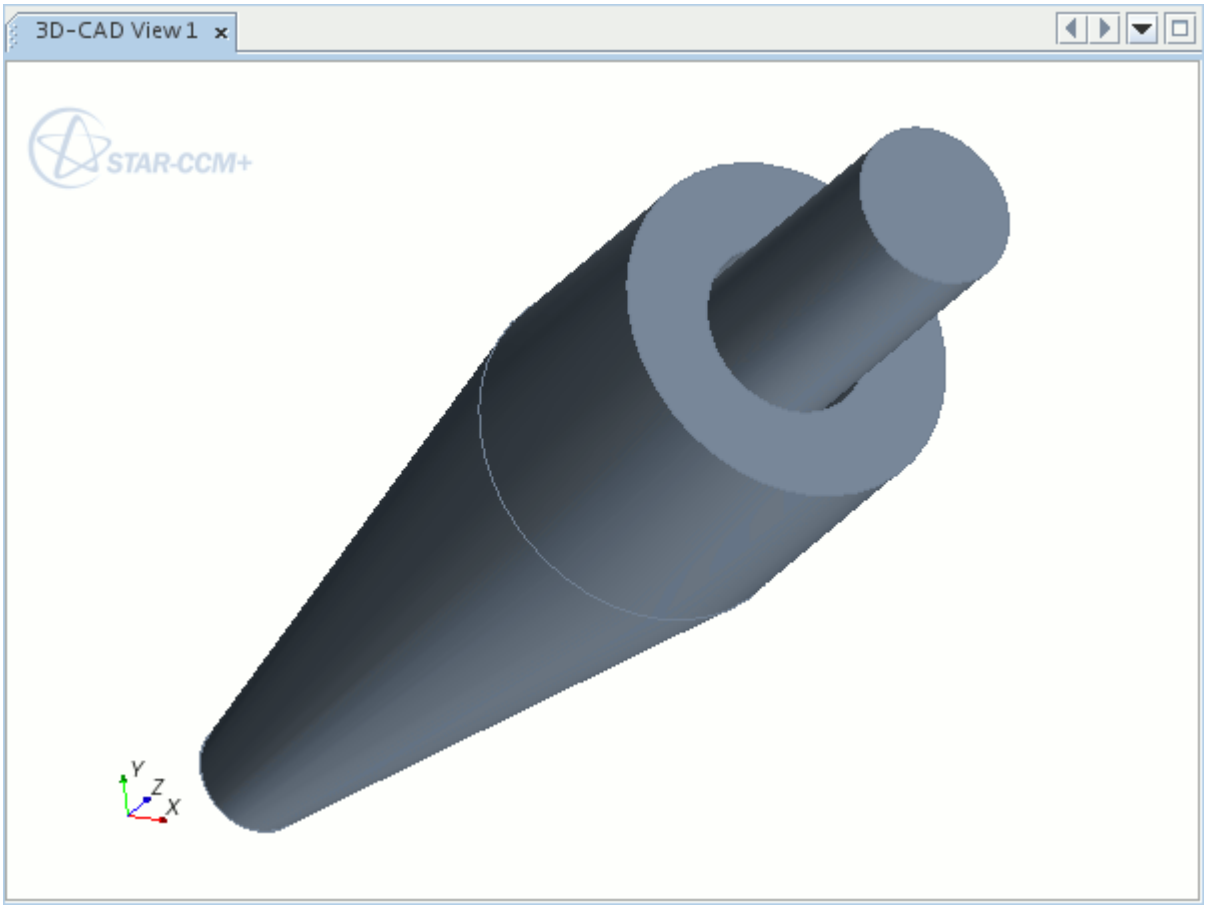
11. Right-click the **Sketch 1** node in the feature tree and select **Extrude**.
12. In the *Extrude* panel, make sure that *Body Interaction* is set to the default option of **Merge**.

    This option causes it to merge with the **Fluid** body, resulting in a single body.
13. Set *Distance* to `0.45 m`.



14. Retain the default settings for the other options.
15. Use the mouse to rotate the model and check the preview of the extrusion.
16. Click **OK**.

The outlet pipe is added to the cyclone chamber:

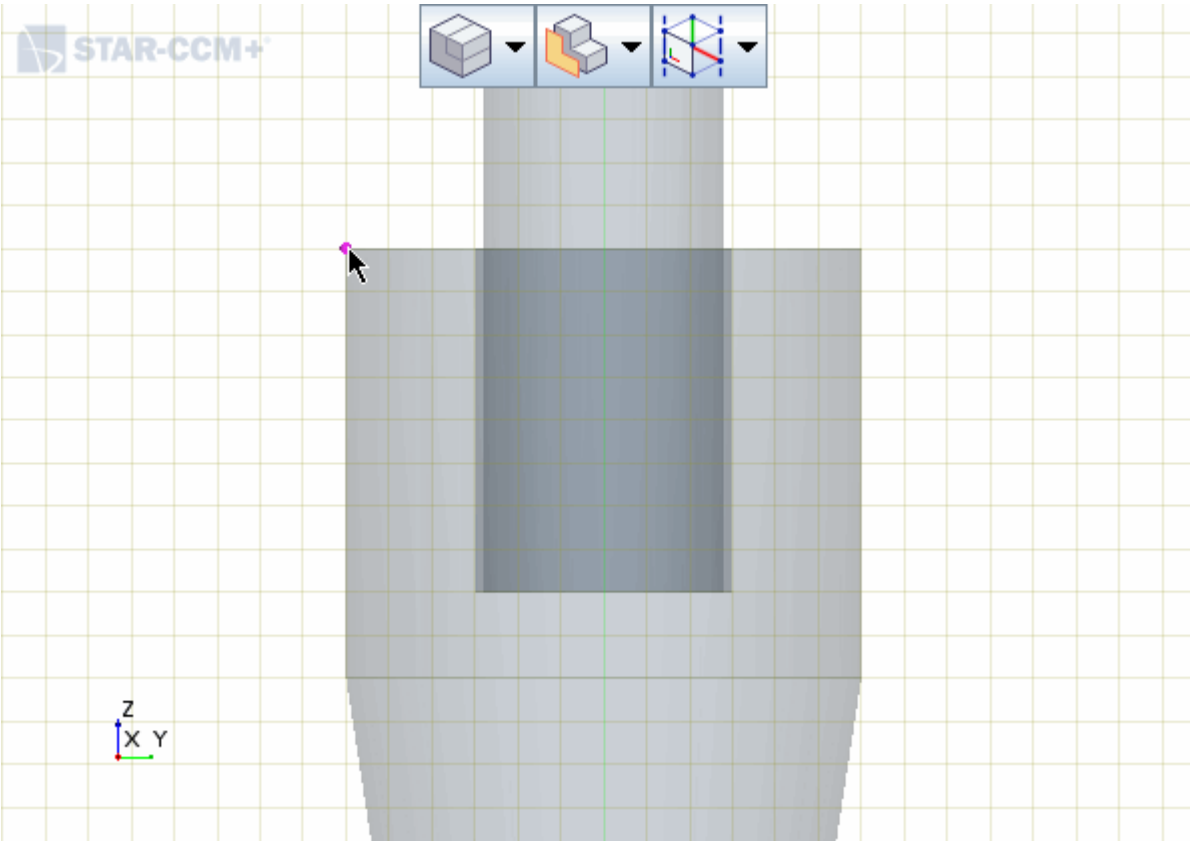


Creating the Inlet Duct

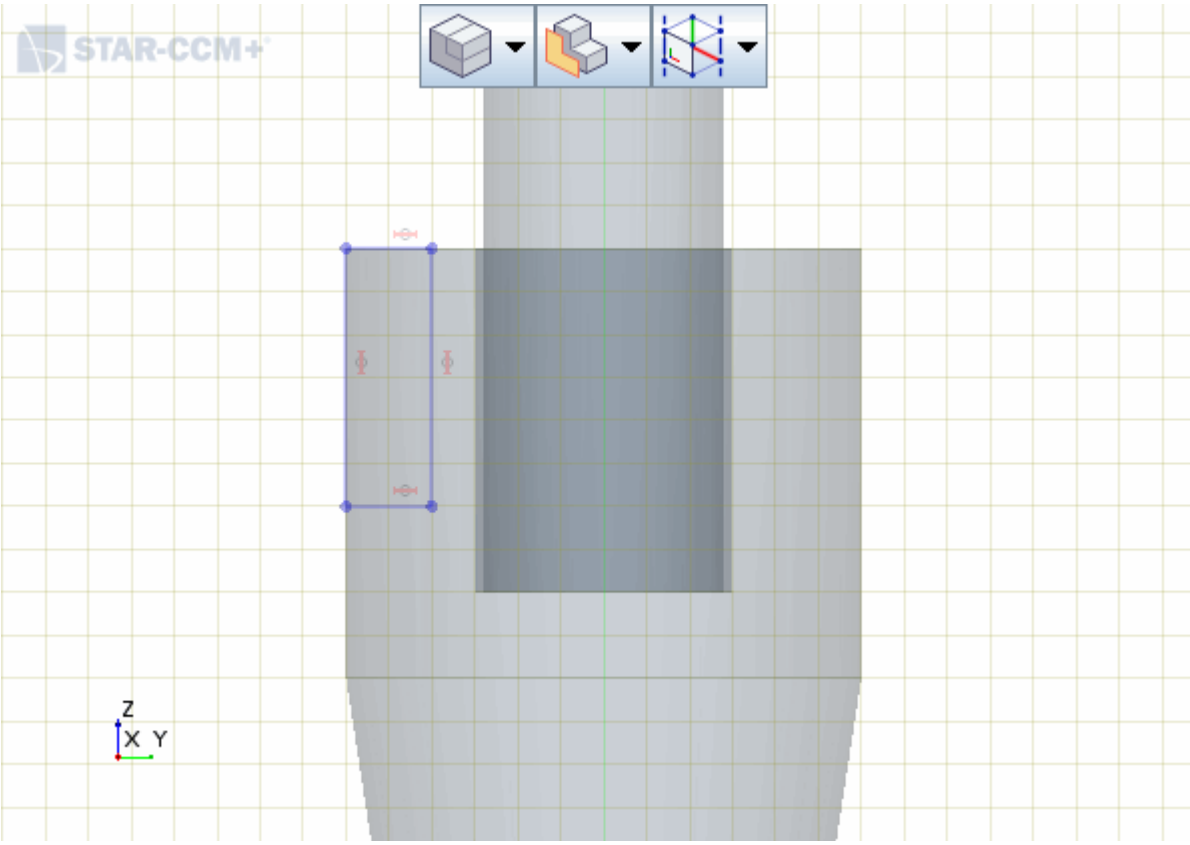
Fluid enters the top of the chamber horizontally through a rectangular inlet duct.

Create the inlet duct:

1. Right-click the **Features** > **YZ** node and select **Create Sketch**.
2. Click (**View normal to sketch plane**).
3. Click (**Make scene transparent**).
4. Zoom in on the top left-hand corner of the geometry.
5. Click (**Create rectangle**) and click the top left corner of the cyclone.



6. Move the mouse pointer down and to the right until the rectangle *Length* that is shown in the properties box is `0.05 m` and the *Width* is `0.15 m`.



7. Left-click to fix the point in position.
8. Click **OK**.

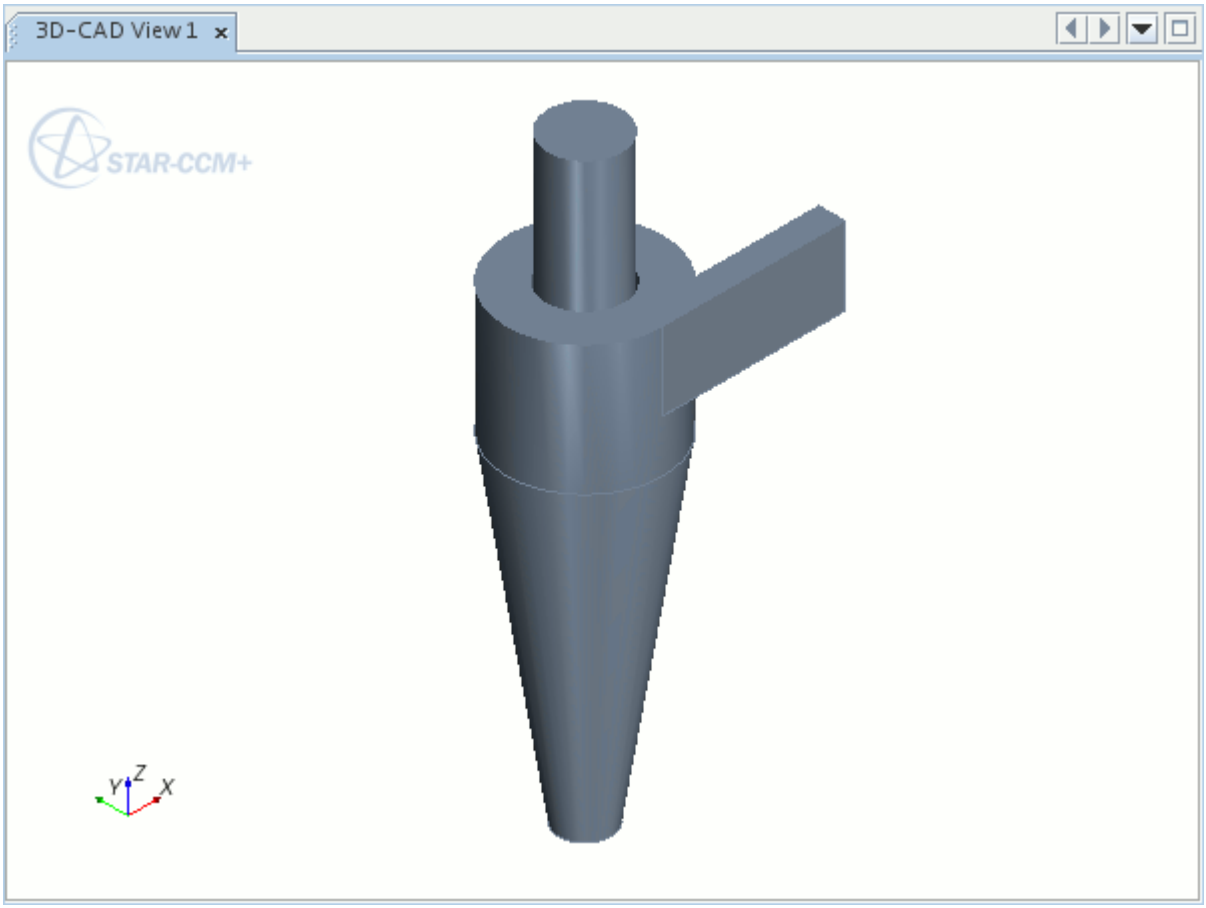
   Constraints are applied automatically to a rectangle to make sure that the sides are vertical and horizontal.

9. Click (**Make Scene Transparent**).

Extrude the sketch:

10. Right-click the **Sketch 2** node and select **Extrude**.
11. In the *Extrude* panel, set the extrusion *Distance* to `0.35 m` and click **OK**.

The inlet duct is added to the cyclone separator:

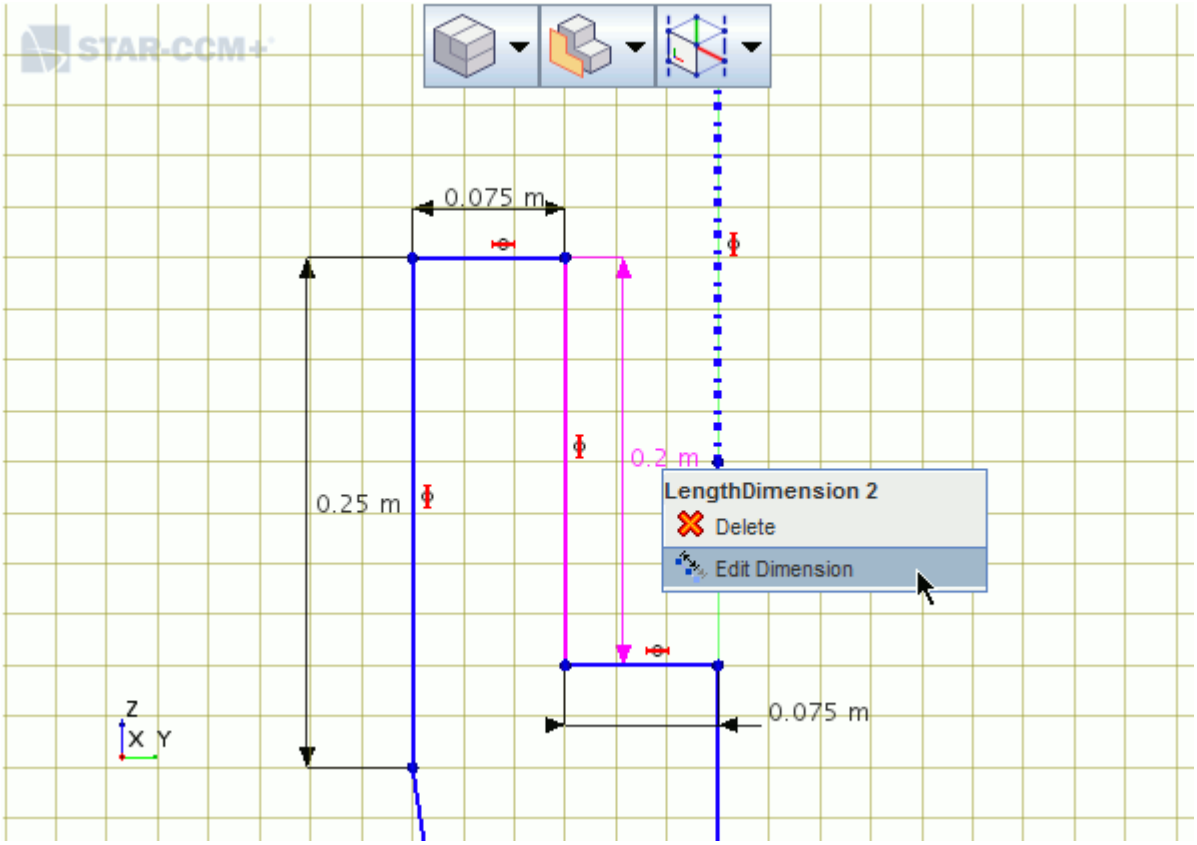


Creating a Design Parameter

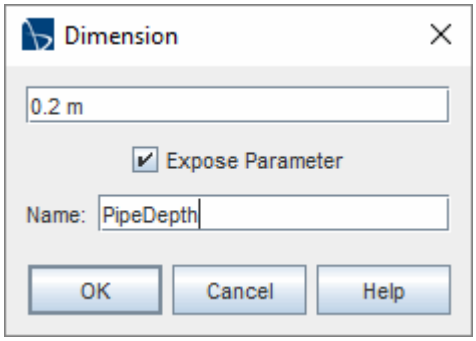
Design parameters allow you to modify an aspect of the geometry from outside 3D-CAD.

Create a design parameter to control the position of the outlet pipe:

1. Right-click the **Sketch Profile** node and select **Edit**.
2. Click (**View normal to sketch plane**).
3. Zoom in on the top of the sketch.
4. Select the vertical line that defines the hole at the top of the cyclone chamber. Right-click the dimension arrow for this line and select **Edit Dimension**.

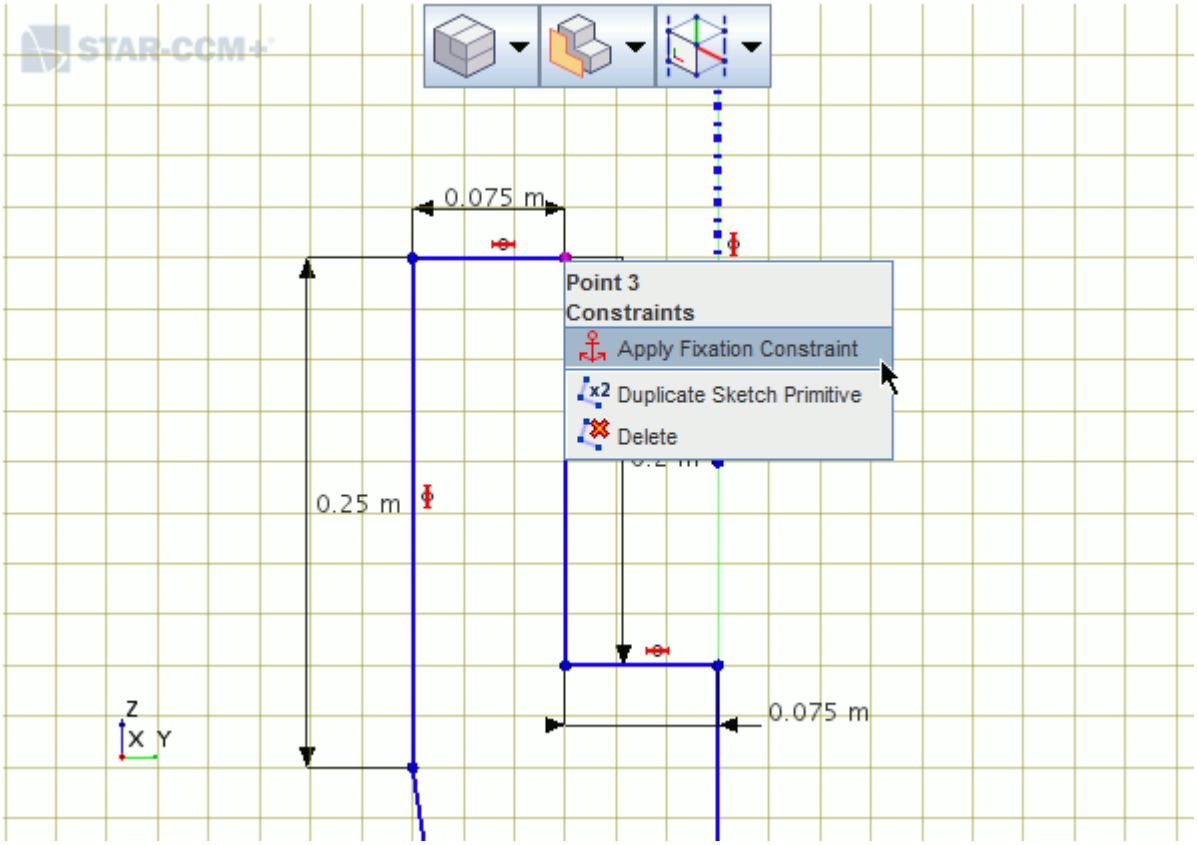


5. In the *Dimension* dialog, activate the *Expose Parameter* option.
6. In the *Name* text box enter `PipeDepth` and click **OK**.

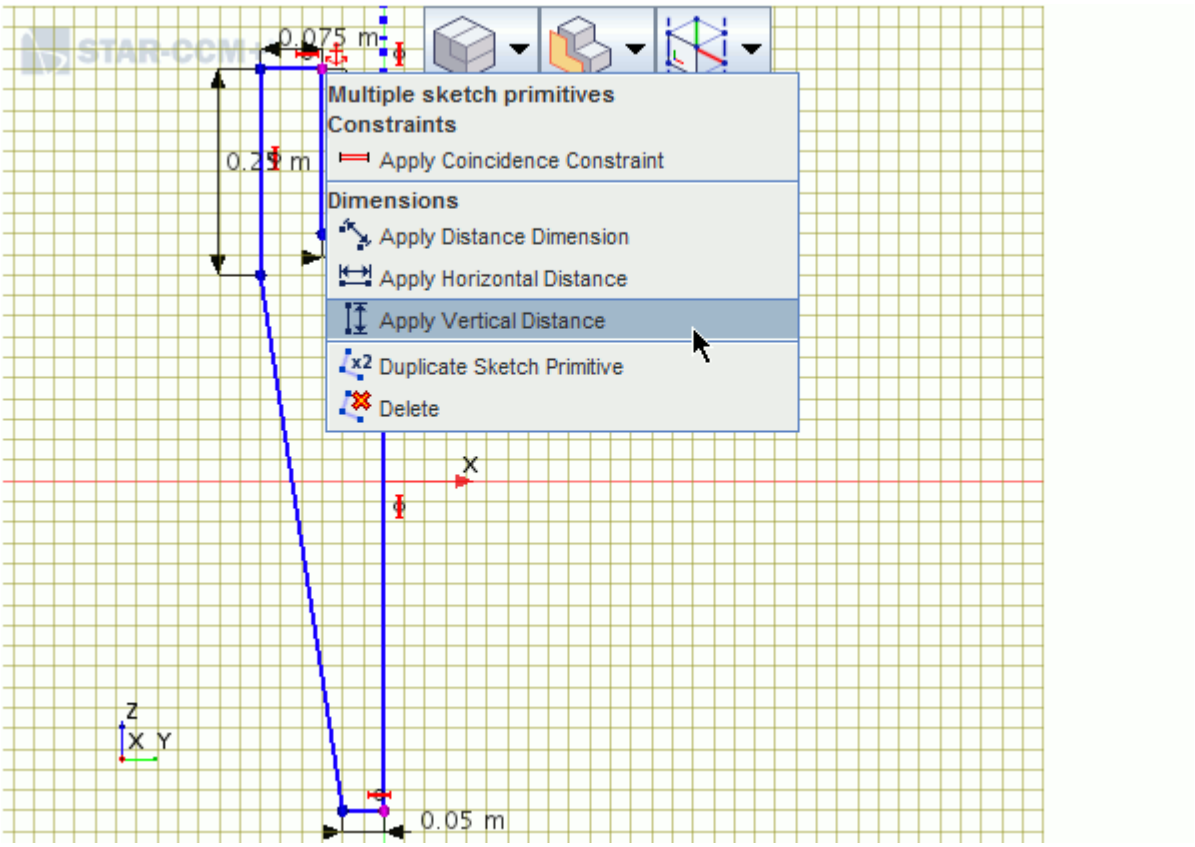


To make sure that the changes that are made to the length of this line only affect the depth of the hole into the cyclone chamber, fix the total height of the cyclone using a vertical dimension. Also fix the position of a point at the top of the cyclone.

7. Right-click the point at the top of the **PipeDepth** line and select **Apply Fixation Constraint**.

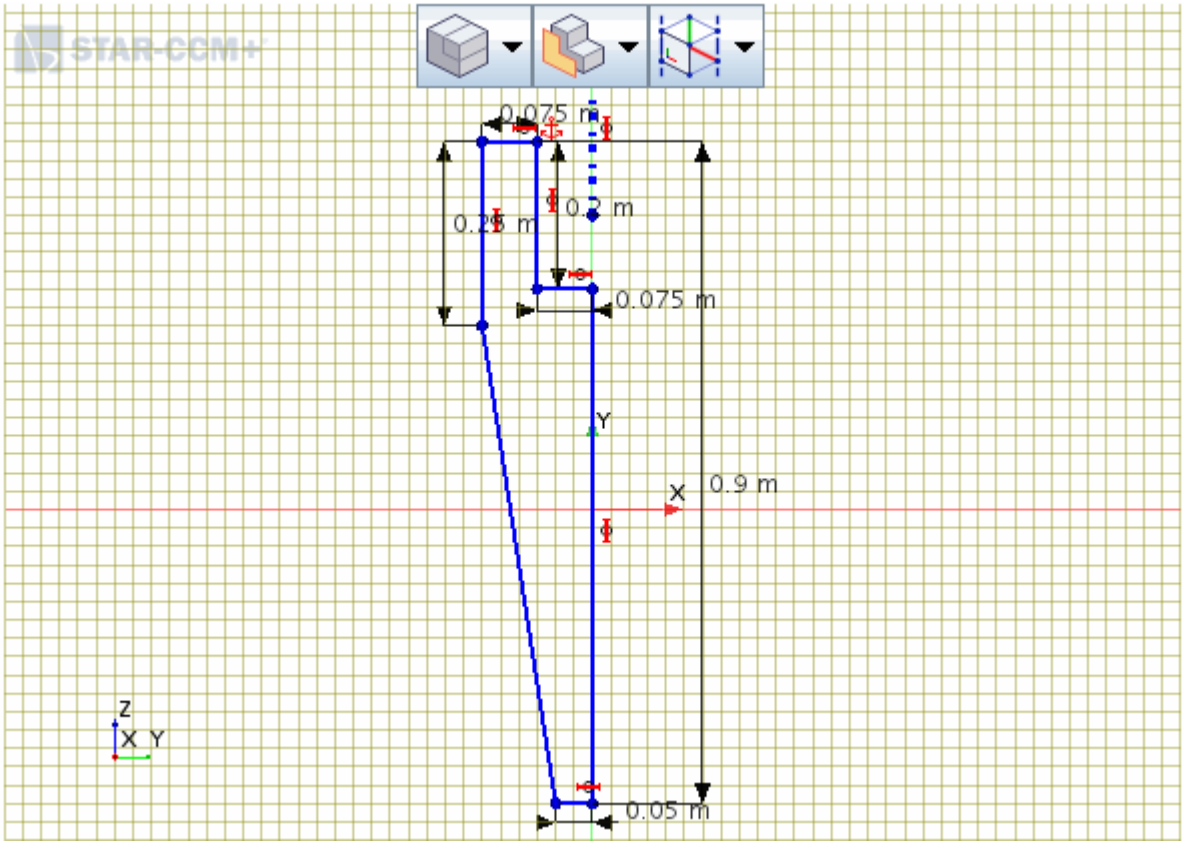


8. Zoom out so that the whole sketch is visible.
9. Select the point at the top of the **PipeDepth** line, hold down the <`Ctrl`> key and select the point at the bottom-right corner of the sketch.
10. Right-click one of the selected points and select **Apply Vertical Distance**.



11. In the *Dimension* dialog, make sure that the value is `0.9 m` and click **OK**.

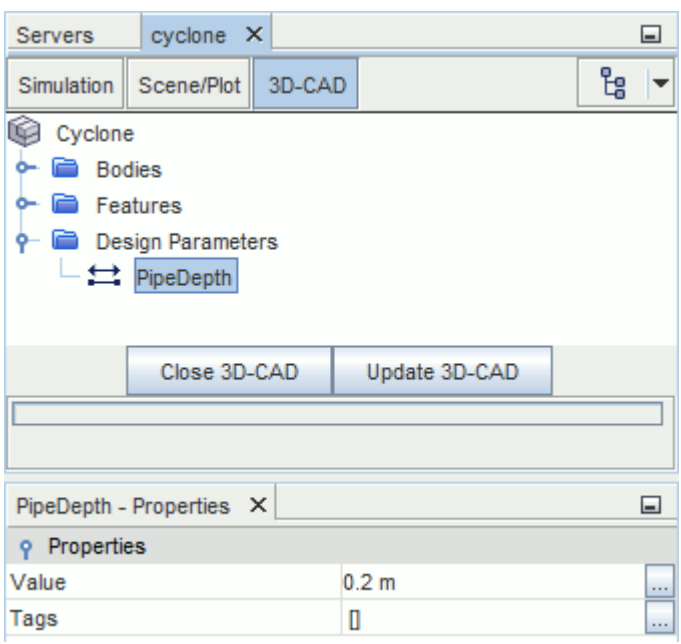
    The final sketch appears as follows:



12. Click **OK** to exit the sketch.

The design parameter is displayed under the **Design Parameters** node in the object tree:

13. Expand the **Design Parameters** node and select the **PipeDepth** node.



The value of this design parameter is shown. Editing this quantity has a direct effect on the dimensions of the model.

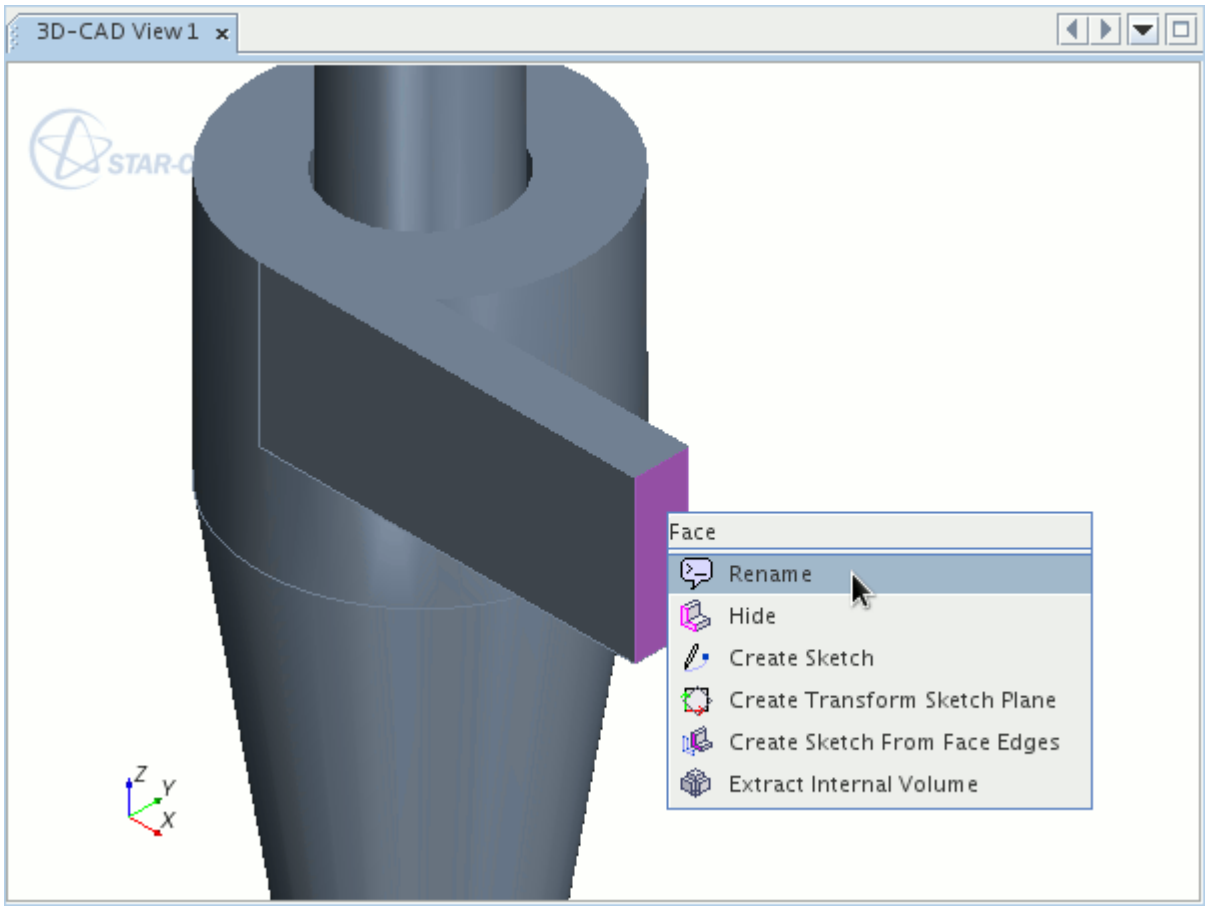
Specifying Inlet and Outlet Faces

The final step in preparing the model geometry is to specify the inlet and outlet faces of the model by setting face names.

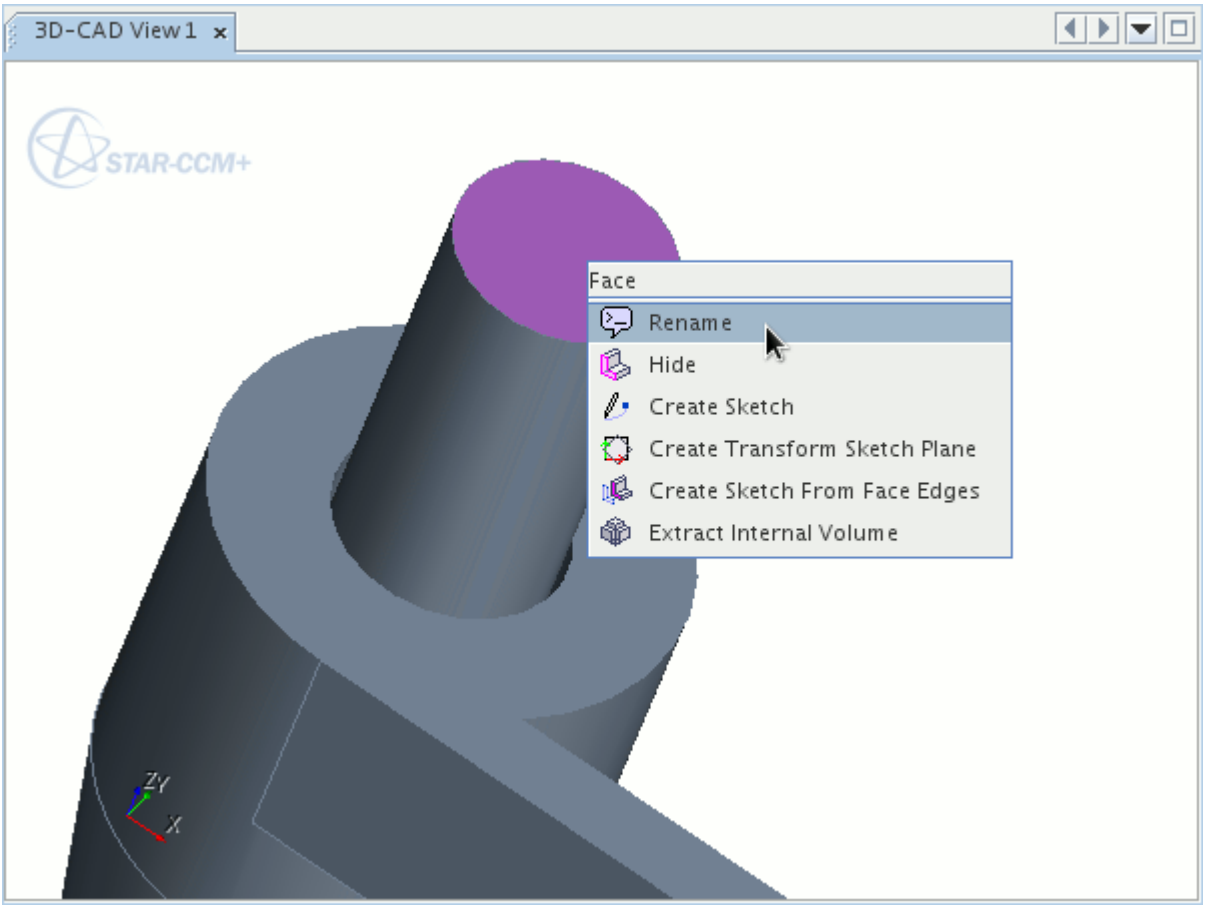
When the 3D-CAD model is imported into the simulation via geometry parts, faces that have been named are defined as separate part surfaces. Therefore, when the geometry is assigned to a region, these surfaces can easily become separate boundaries.

Name the faces for the inlet and outlet:

1. Rotate the model so that the inlet duct is visible.
2. Right-click the rectangular inlet face and select **Rename**.



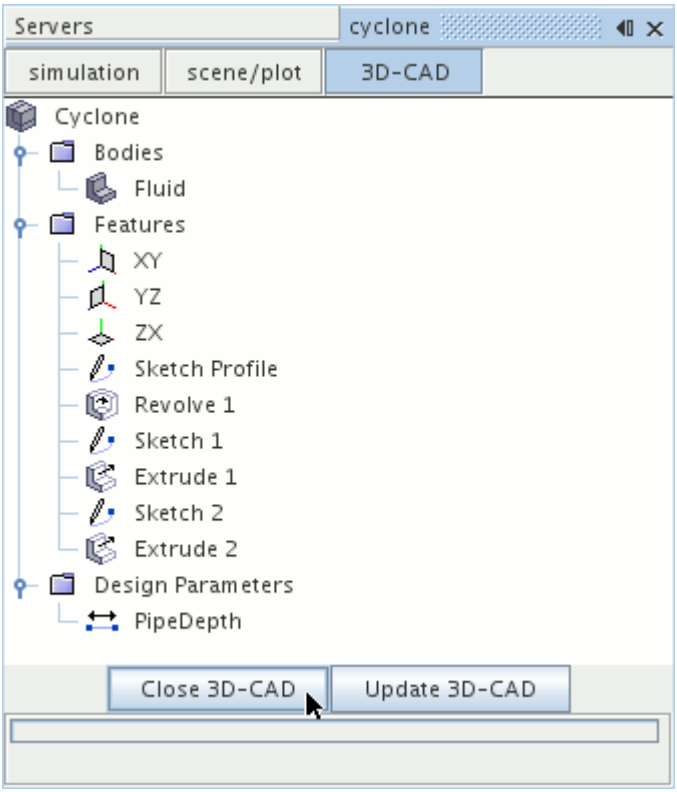
3. In the *Rename* dialog, enter `Inlet` and click **OK**.
4. Right-click the circular face at the top of the outlet pipe and select **Rename**.



5. In the *Rename* dialog, enter `Outlet` and click **OK**.

   The geometry is now complete and you can exit 3D-CAD.

6. At the bottom of the object tree, click **Close 3D-CAD**.



When you exit 3D-CAD, the *3D-CAD View 1* scene closes automatically.

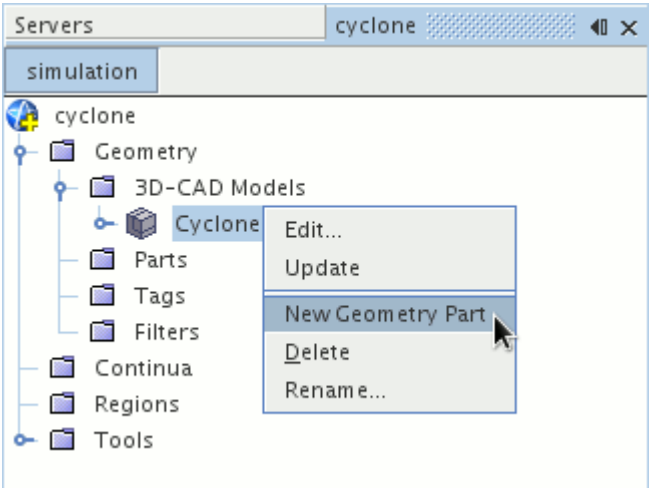
7. Save the simulation.

Creating a Geometry Part

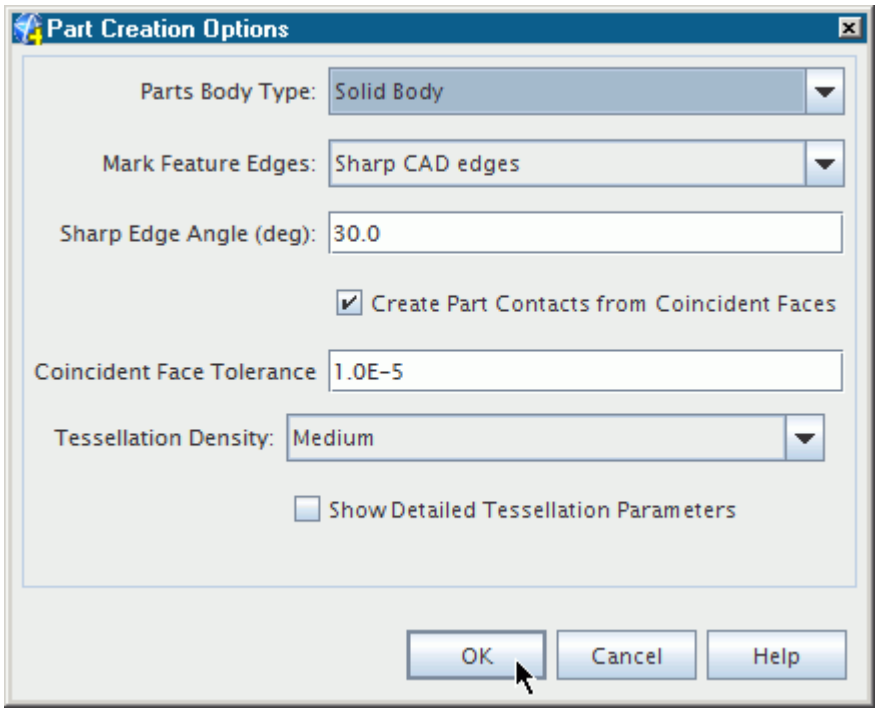
Create a geometry part from the 3D-CAD model.

To use the 3D-CAD model in a simulation, create a geometry part:

1. Right-click the **Geometry** > **3D-CAD Models** > **Cyclone** node and select **New Geometry Part**.



2. In the *Parts Creation Options* dialog, click **OK** to close the dialog. The default settings are acceptable.



3. Expand the **Parts** > **Fluid** > **Surfaces** node.

   The inlet and outlet faces that you specified in the previous section are defined as separate surfaces.

Assigning a Part to a Region

To define the computational domain, assign the cyclone separator part to a region.

1. Right-click the **Geometry** > **Parts** > **Fluid** node and select **Assign Parts to Regions**.
2. In the *Assign Parts to Regions* dialog, select:
   a) **Create a Region for Each Part**
   b) **Create a Boundary for Each Part Surface**
3. Click **Apply** then **Close**.

   A new region is created with boundaries that define the inlet, outlet, and main fluid volume.

Check that the region and boundaries are defined correctly:

4. Create a Geometry Scene.
5. Expand the **Regions** > **Fluid** > **Boundaries** node and select each of the boundary nodes to make sure that they are specified correctly.

Selecting Physics Models

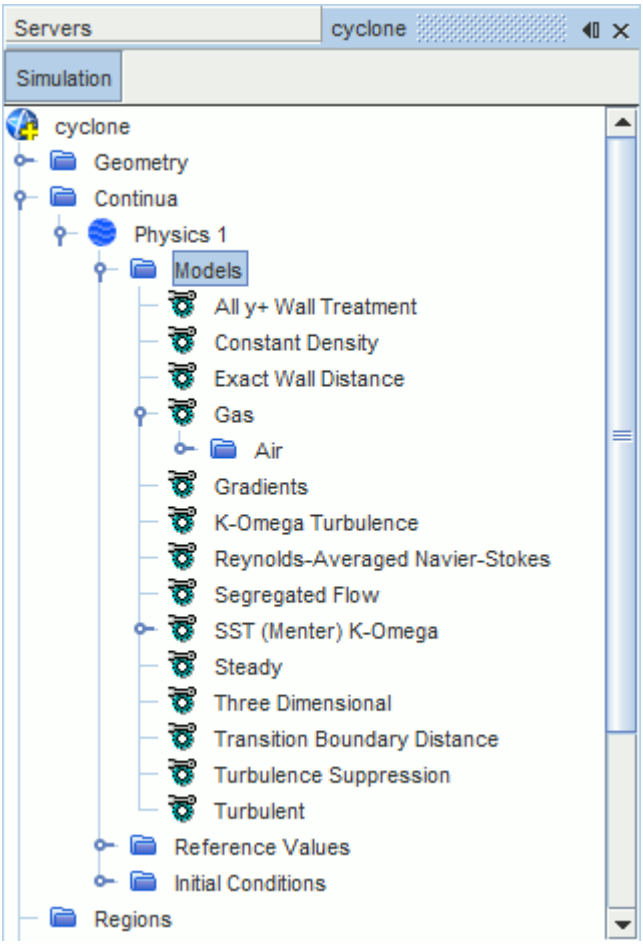
Physics models define the physical variables and phenomena in the simulation. In this tutorial, the K-Omega turbulence model is used to limit the simulation run time.

To select the physics models:

1. Create a physics continuum.
2. For the physics continuum, **Continua** > **Physics 1**, select the following models in order:

Group Box	Model
Space	**Three-Dimensional**
Time	**Steady**
Material	**Gas**
Flow	**Segregated Flow**
	Gradients (selected automatically)
Equation of State	**Constant Density**
Viscous Regime	**Turbulent**
	Reynol-Averaged Navier-Stokes (selected automatically)
Reynolds-Averaged Turbulence	**K-Omega Turbulence**
	SST (Menter) K-Omega (selected automatically)
	Exact Wall Distance (selected automatically)
	All y+ Wall Treatment (selected automatically)
Transition	**Turbulence Suppression**
	Transition Boundary Distance (selected automatically)

3. Click **Close**.
4. To review the models, open the **Physics 1** > **Models** node.



5. Save the simulation.

Contents:

Specifying Boundary Conditions

Specifying Boundary Conditions

Specify conditions at the inlet and outlet boundaries:

1. Edit the **Regions** > **Fluid** > **Boundaries** node.
2. Set the following properties:

Node	Property	Setting
Inlet	*Type*	**Velocity Inlet**
└ **Physics Conditions** > **Turbulence Specification**	*Method*	**Intensity + Length Scale**
└ **Physics Values** > **Velocity Magnitude**	*Value*	`5 m/s`
Outlet	*Type*	**Outlet**

Generating the Volume Mesh

Use a polyhedral mesh to analyze the flow patterns in the cyclone separator.

1. Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
2. In the *Create Automated Mesh Operation* dialog:
   a) Select **Fluid** from the *Parts* list.
   b) Select the following meshers, in order:

Group	Mesher
Surface Meshers	**Surface Remesher**
Core Volume Meshers	**Polyhedral Mesher**
Optional Boundary Layer Meshers	**Prism Layer Mesher**

   c) Click **OK**.

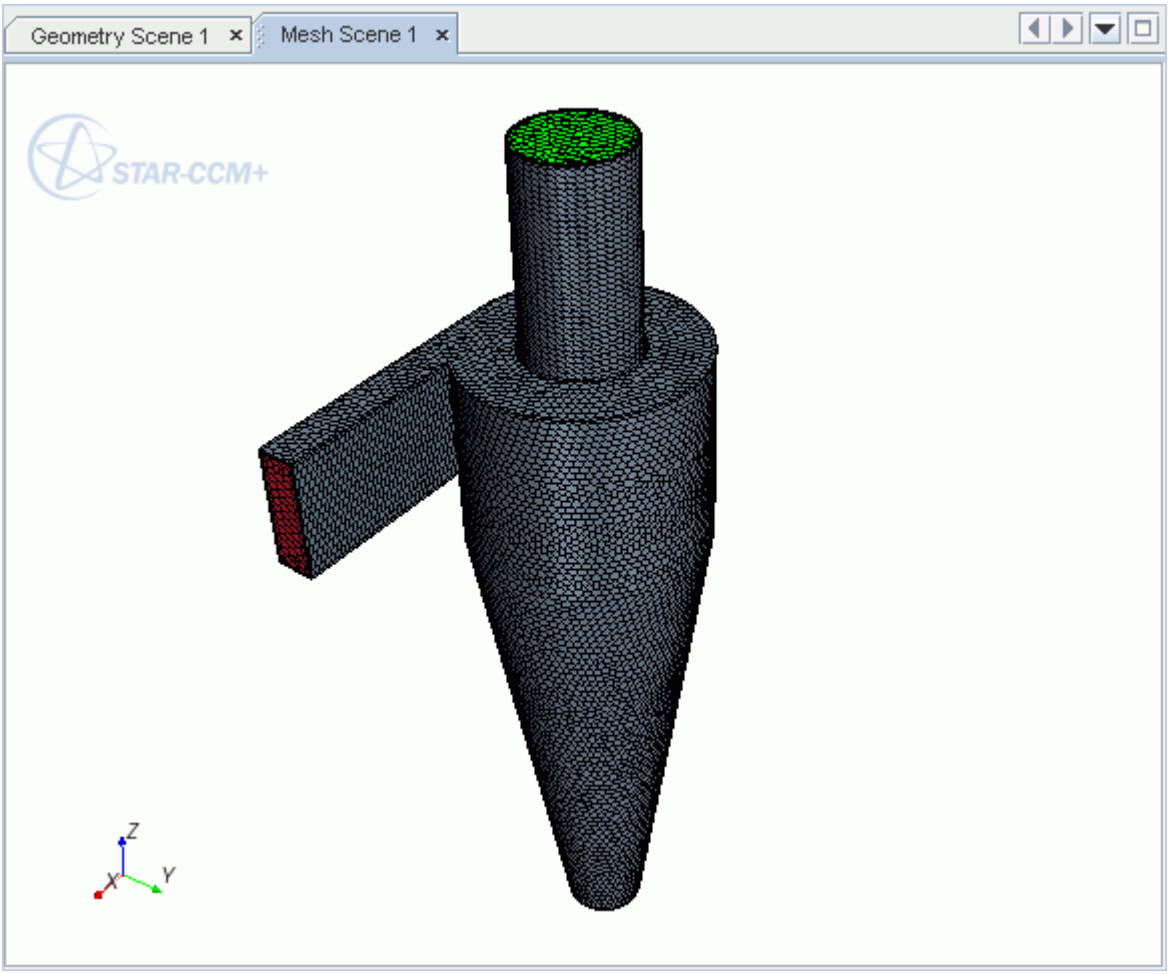
   An automated mesh operation is added to the **Operations** node.
3. Select the **Automated Mesh** > **Default Controls** > **Base Size** node and set *Value* to `0.01 m`.

Generate the volume mesh:

4. Click **(Generate Volume Mesh)**.

   When the volume mesh is generated, the message `Volume Meshing Pipeline Completed` is displayed in the *Output* window. The number of Cells, Faces, and Vertices are also reported.
5. Create a mesh scene.

   The volume mesh is displayed below.



6. Save the simulation.

Preparing a Scalar Scene

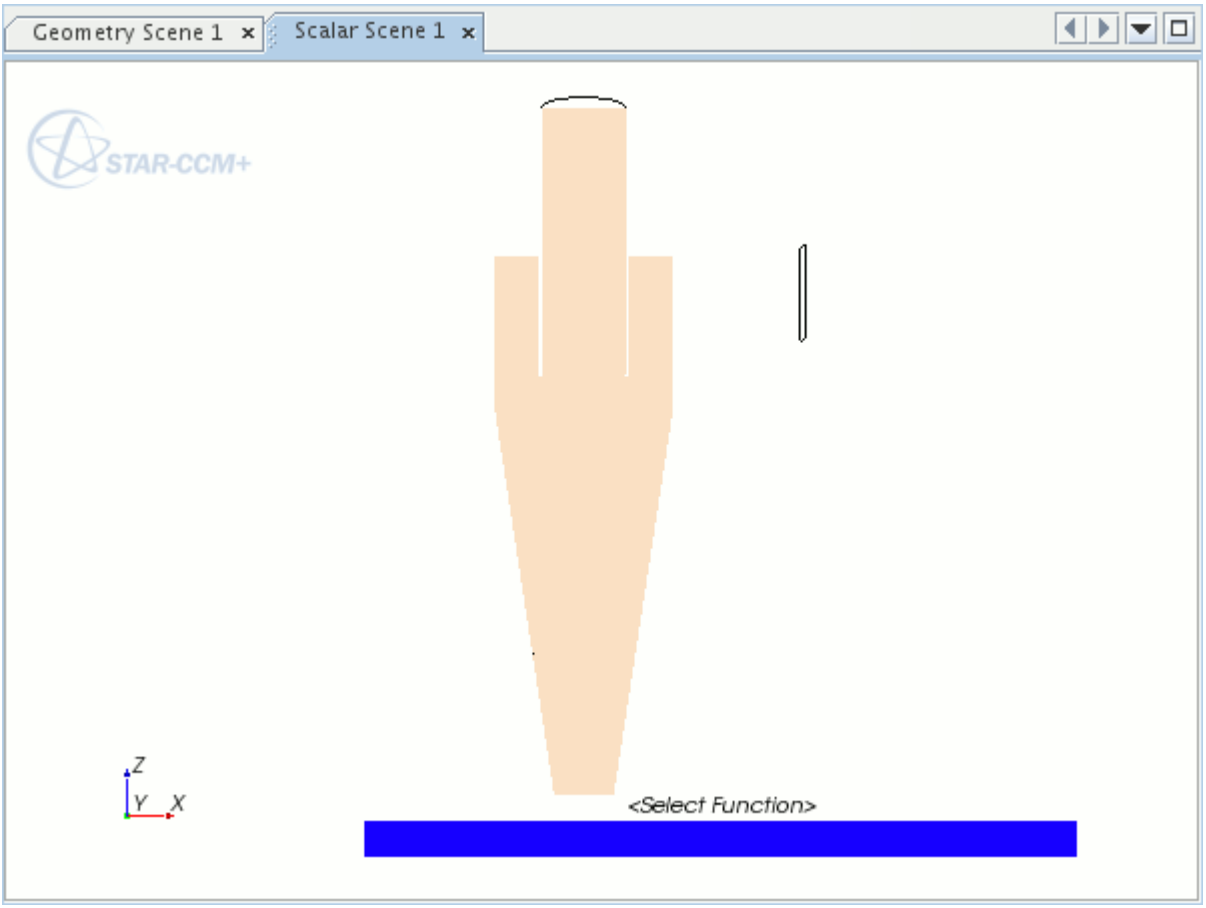
Use a scalar scene to visualize the vertical velocity on a section through the cyclone chamber.

You can use this scene to visualize the solution while the simulation is running.

1. Create a scalar scene.
2. Create a section plane with the following properties:

Property	Setting
Input Parts	**Regions** > **Fluid**
origin	`[0, 0, 0] m`
normal	`[0, 1, 0] m`
Display	**Existing Displayer** > **Scalar 1**

3. Rotate the view in *Scalar Scene 1* so that the plane section is visible.



Define the scene settings:

4. Click **scene/plot**.
5. Edit the **Displayers** node and set the following properties:

Node	Property	Setting
└ **Scalar 1**	*Contour Style*	**Smooth Blended**
└ **Scalar Field**	*Function*	**Total Pressure** > **Lab Reference Frame**

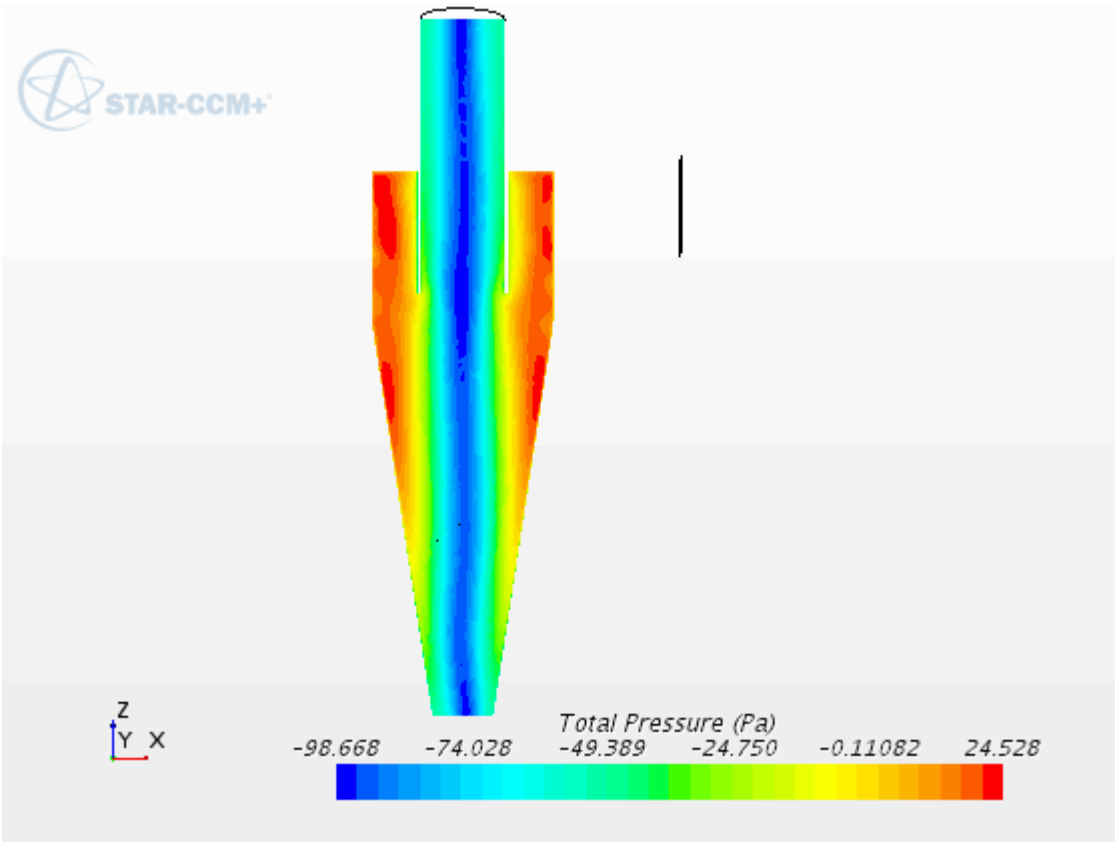
6. Save the simulation.

Running the Simulation

Preparation of the simulation is now complete, and the simulation can be run.

1. Click (**Run**) in the *Solution* toolbar.
2. While the simulation is running, select the *Scalar Scene 1* tab at the top of the *Graphics* window to visualize the solution.
3. When the simulation has finished running, save it.

The scalar scene appears as follows:



Creating a Streamlines Scene

Create a scene to display the streamlines inside the cyclone separator.

1. Create a Geometry Scene.
2. Rename the **Geometry Scene 2** node to `Streamlines`.

Use the *Create Streamline* panel to specify the properties of the streamlines. In this case, the streamlines are grown from the cyclone inlet.

3. Create a Streamline derived part with the following properties:

Property	Setting
Input Parts	**Regions** > **Fluid**
Seed Parts	**Regions** > **Fluid** > **Boundaries** > **Inlet**
Part U-Resolution	2
Part V-Resolution	8

Define the scene settings:

4. Click **scene/plot**.

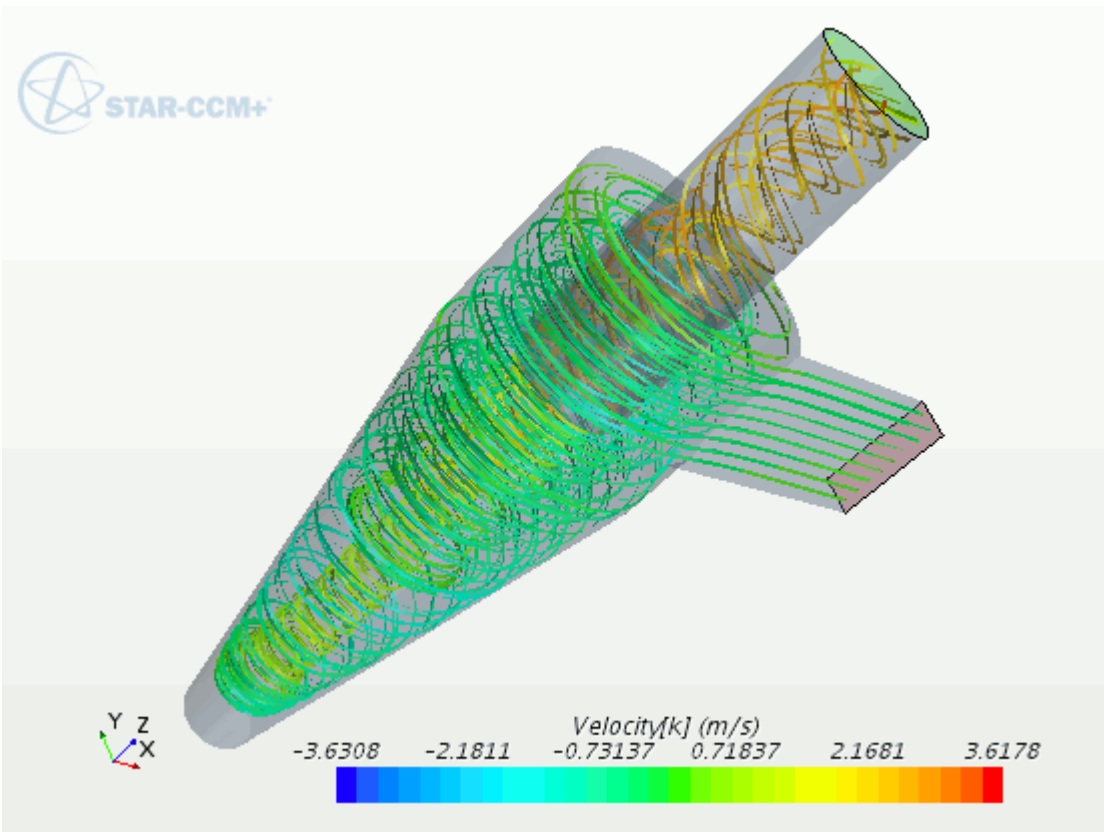
5. Edit the **Streamlines** > **Displayers** node and set the following properties:

Node	Property	Setting
Displayers		
└ **Geometry 1**	*Opacity*	`0.3`
└ **Streamline Stream 1**	*Mode*	**Ribbons**
└ **Scalar Field**	*Function*	**Velocity** > **Lab Reference Frame** > **Laboratory** > **k**

To extend the streamlines through the fluid domain up to the outlet, increase the maximum propagation property. This defines how far the streamlines are propagated through the fluid domain from the starting points. A higher value causes the streamlines to extend further.

6. Click **Simulation**.
7. Select the **Derived Parts** > **streamline** > **2nd Order Integrator** node and set *Maximum Propagation* to `15`.

The streamlines appear as follows:



Contents:

Animating the Streamlines

Animating the Streamlines

Animate the streamlines inside the cyclone chamber.

To animate streamlines:

1. Click **scene/plot**.
2. Edit the **Displayers** node and set he following properties:

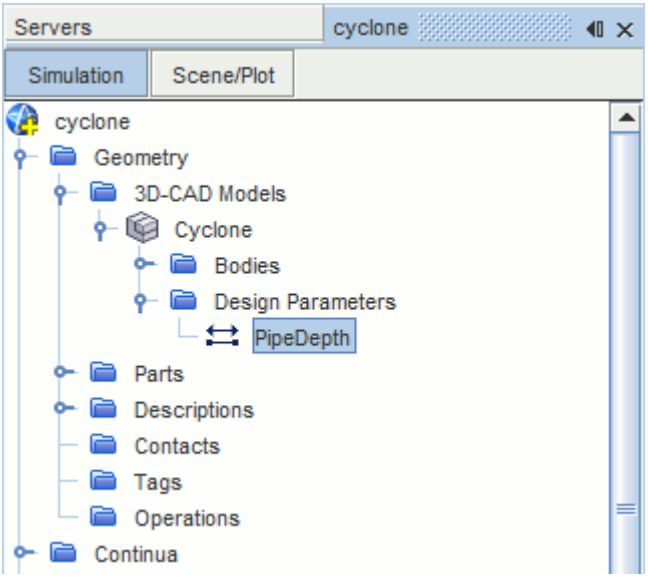
Node	Property	Setting
Displayers		
└ **Streamline Stream 1** > **Animations**	*Animation Mode*	**Tracers**
└ **Streamline Settings**	*Delay between tracers (sec)*	`6`
	Head size	`0.01`
	Tail Length (sec)	`1`

3. In the *Animation* toolbar:
   a) Click (**Play**)
   b) To stop the animation, click (**Stop**)

Modifying the Geometry and Rerunning

Modify the position of the outlet pipe using the design parameter, **PipeDepth**, that you created earlier.

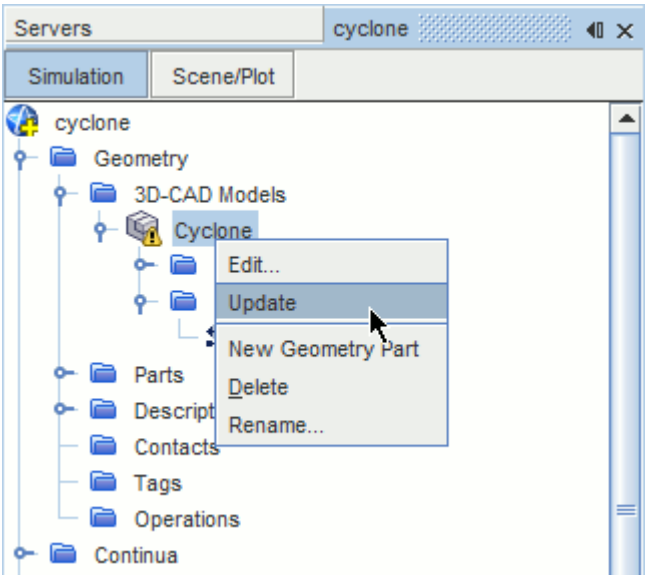
1. Expand the **Geometry** > **3D-CAD Model** > **Cyclone** node.



2. Select the **Design Parameters** > **Pipe Depth** node and set **Value** to `0.35 m`.

The **Cyclone** node shows an **Out-of-Date** feature state icon to indicate that the design parameter has changed and that the model requires updating.

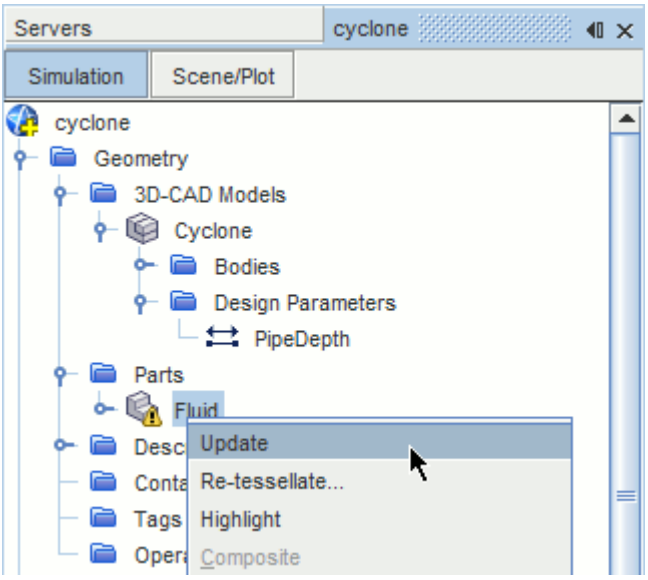
3. Right-click the **Cyclone** node and select **Update**.



The outlet pipe moves further down inside the chamber, and less of the pipe is visible from the top.

To display the changes to the geometry, update the geometry part:

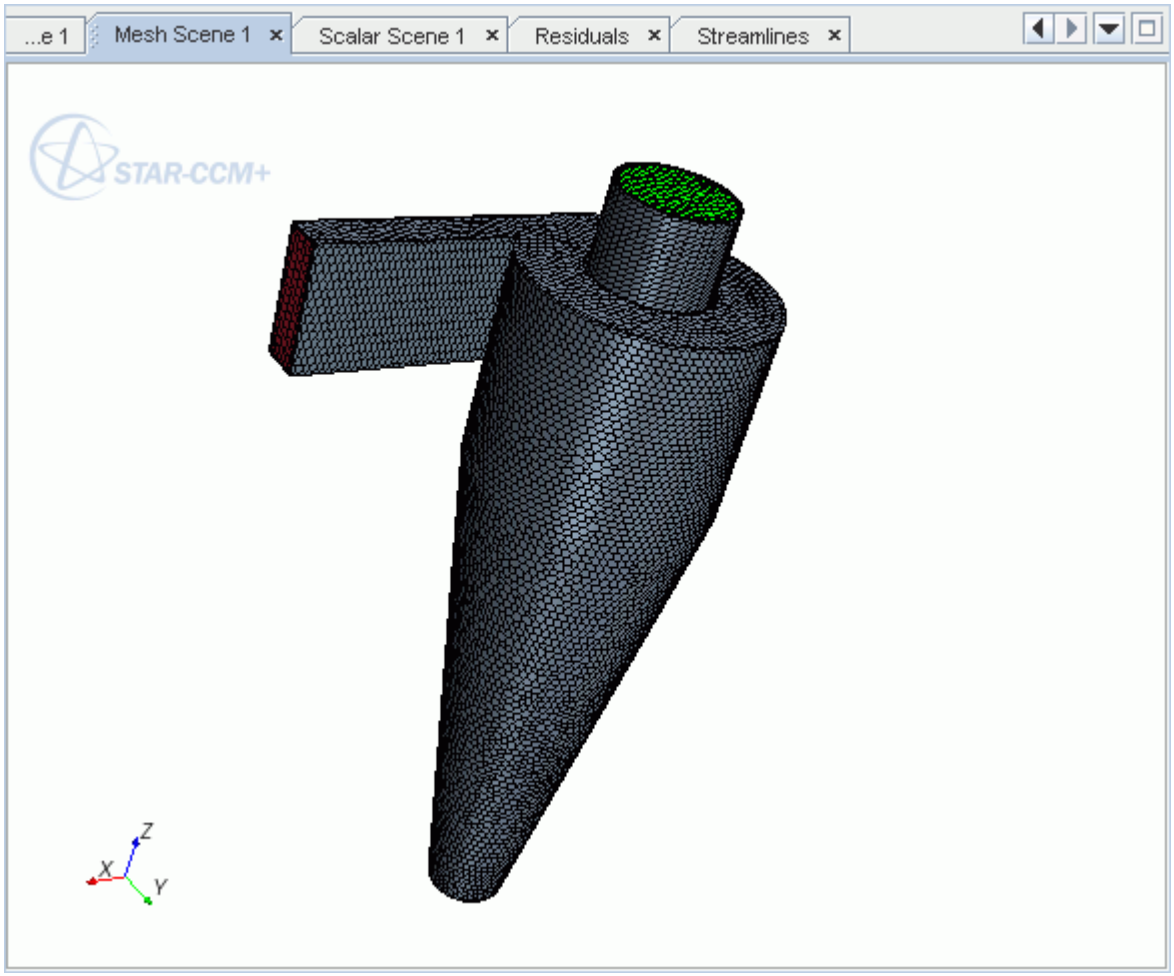
4. Display **Geometry Scene 1** in the *Graphics* window.
5. Right-click the **Geometry** > **Parts** > **Fluid** node and select **Update**.



The changes are applied to the geometry in the scene. These changes are not yet applied to the volume mesh.

Update the volume mesh:

6. Select the **Mesh Scene 1** tab in the *Graphics* window.
7. Regenerate the volume mesh.



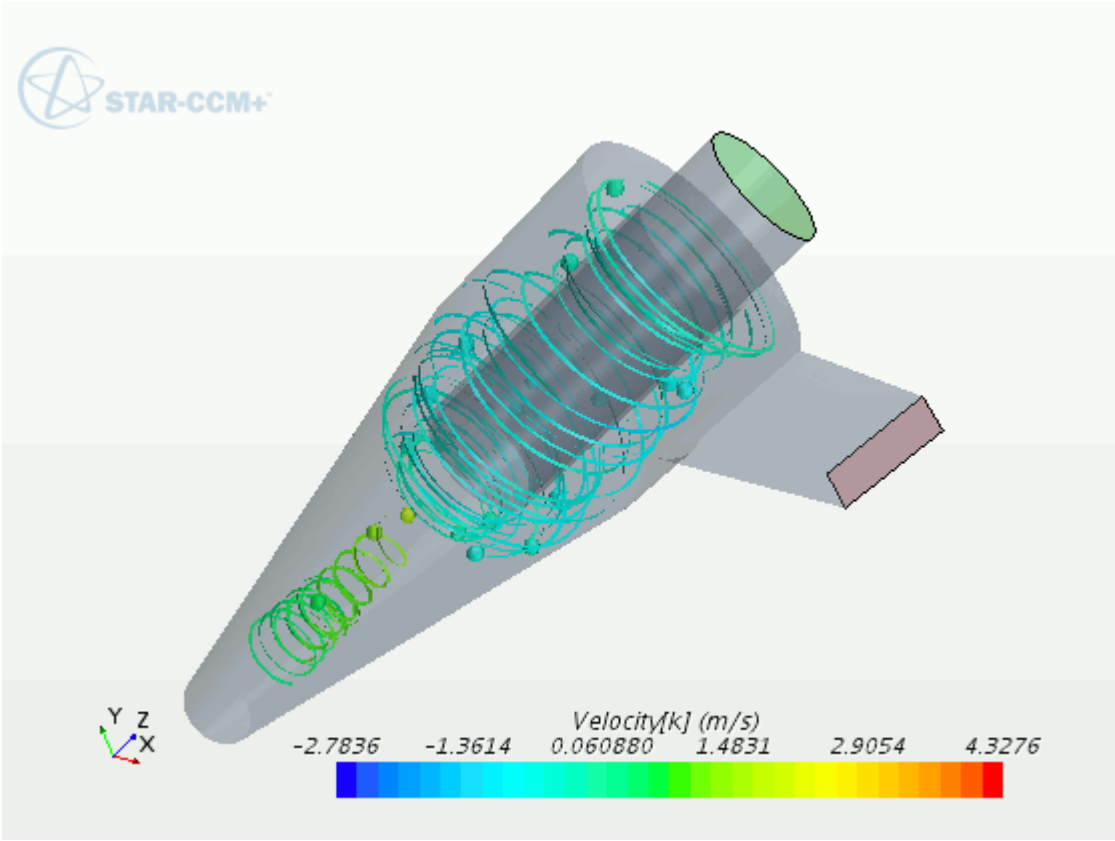
The new volume mesh shows the outlet pipe positioned lower in the cyclone chamber.

The solution from the original geometry is automatically mapped onto the new mesh, decreasing the simulation run time for the second run. As fewer iterations are required to solve the case, alter the stopping criteria:

8. Select the **Stopping Criteria** > **Maximum Steps** node and set **Maximum Steps** to `1400`.
9. Close the *Streamlines Scene*.

   Leaving the *Streamlines Scene* open increases the run time because streamlines are calculated after each iteration.
10. Save the simulation as `cycloneModified.sim`.
11. Run the Simulation.
12. When the simulation has finished running, save it.
13. Open the Streamlines Scene and animate the scene as described previously.

The streamlines for the modified geometry appear as follows:



Summary

This tutorial has introduced the following STAR-CCM+ features:

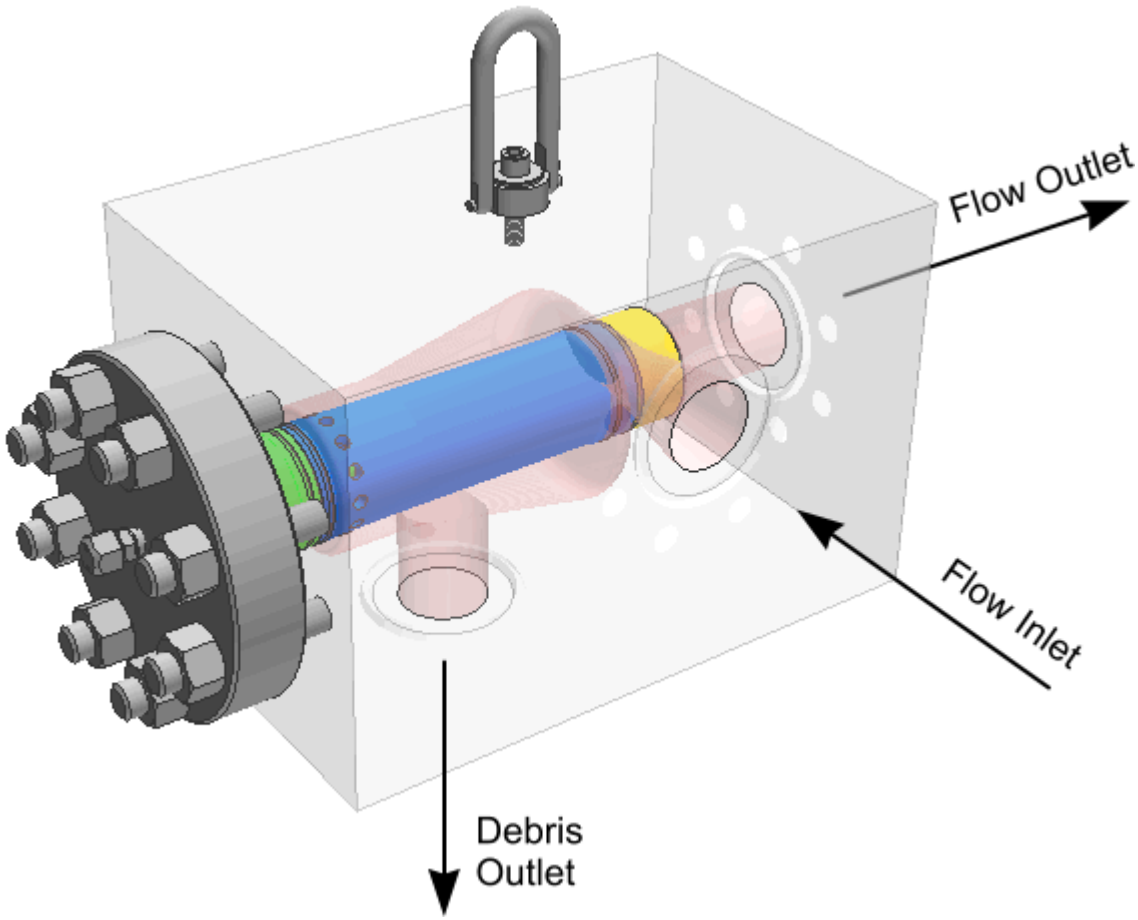
- Using basic modelling techniques in 3D-CAD: Sketching, Revolving, and Extruding
- Exposing a design parameter
- Specifying inlet and outlet faces within 3D-CAD
- Creating geometry parts from the 3D-CAD model
- Assigning the 3D-CAD part to a region
- Modifying the 3D-CAD model using a design parameter and rerunning the case

Geometry Parts: CAD Assembly

This tutorial demonstrates how to use the geometry parts feature in STAR-CCM+ with an imported CAD assembly.

STAR-CCM+ also offers a number of tools that make it easier to prepare a simulation that is based on geometry which contains multiple parts. Several of these tools are demonstrated in this tutorial, such as: reorganizing the assembly structure, splitting and combining parts, and repairing part surfaces. The process of assigning geometry parts to regions is also covered, as well as the use of tags, filters, and meta data.

The geometry that is used in this tutorial is based on a device that is used to extract debris from high-pressure gas flows. The device operates by spinning the oncoming fluid around a central chamber and using centrifugal force to pull sand or other debris from the fluid. The clean flow media exits through a central pipe while the debris is collected from an outlet at the base. The assembly and its constituent parts were prepared using a standard CAD package, and are provided for this tutorial in Parasolid Transmit format.



Note: The geometry is based on an original design from Wood Group Pressure Control Ltd (WGPC), and is used with their permission.

A computational mesh is generated for the component in three stages. To begin, a closed internal surface mesh is generated using the surface wrapper. This wrapped surface is then retriangulated using the surface remesher. Finally a volume mesh is generated using the trimmer and prism layer meshers, and the extruder is used to extend the inlet and outlet of the device. To capture the complex and intricate features of the geometry the mesh generation process utilizes contact prevention conditions, volumetric controls and the wrapper scale factor.

Contents:

Prerequisites
Importing the Surface Mesh
Reorganizing the Assembly
Re-Tessellating Parts
Repairing Part Surfaces
Splitting and Combining Part Surfaces
Creating a Part from Part Surfaces
Creating a Tag
Applying a Tag to a Part
Setting Part Metadata
Setting Up Filters
Assigning Parts to Regions
Assigning Part Surfaces to Boundaries
Creating the Automated Mesh Operation
Setting Global Mesh Properties
Creating Volume Shapes
Creating a Volumetric Control
Setting Wrapper Volume of Interest
Specifying a Contact Prevention Condition
Setting Up Extrusions
Deleting Existing Feature Curves
Initializing the Mesh
Generating the Surface Mesh
Generating the Volume Mesh
Acknowledgements
Summary

Prerequisites

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Importing the Surface Mesh

To set up the STAR-CCM+ simulation, launch a simulation and import the supplied surface mesh.

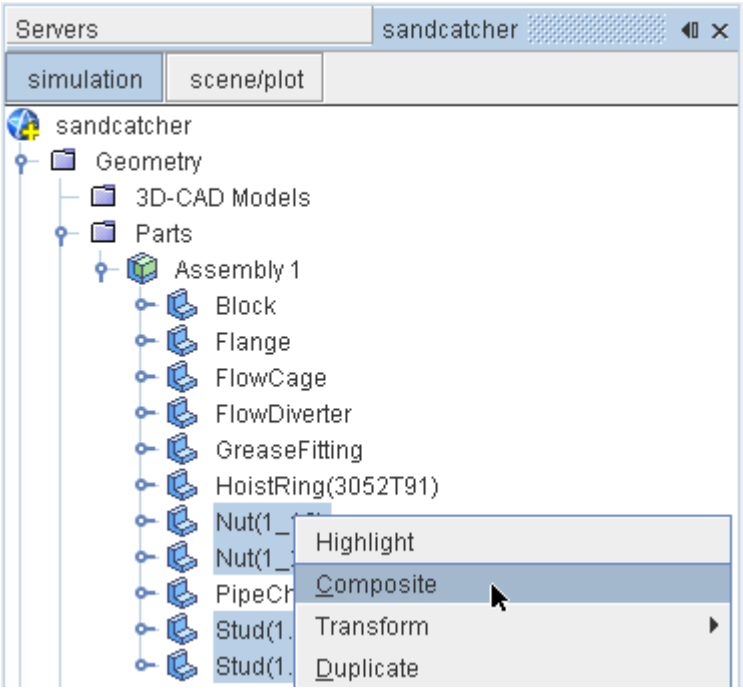
1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Surface Mesh**.
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/geometry/data`.
5. Select `sandcatcher.x_b` then click **Open**.
6. In the *Import Surface Options* dialog, click **OK** to accept the default options.
7. Save the simulation as `sandcatcher.sim`.

Reorganizing the Assembly

Reorganize the components in the assembly.

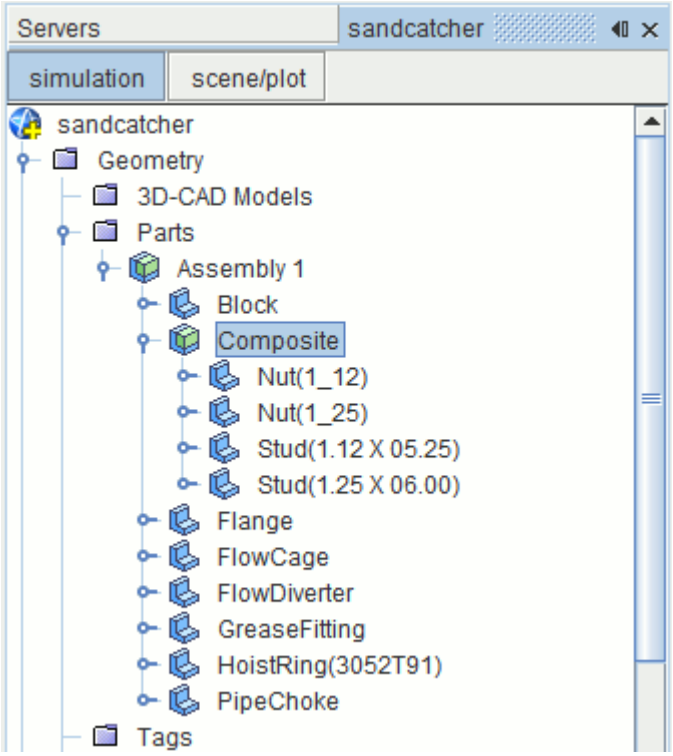
One of the benefits of using geometry parts is that you can reorganize the assembly nodes to suit the needs of the analysis. In this case, the assembly has a number of nuts and studs which are superficial and do not affect the analysis of the flow through the part. Group these nuts and studs into a composite part to simplify the object tree.

1. Expand the **Geometry** > **Parts** > **Assembly 1** node.
2. Select the **Nut(1_12)**, **Nut(1_25)**, **Stud(1.12 X 05.25)**, and **Stud(1.25 X 06.00)** nodes.
3. Right-click any selected node and choose **Composite**.



A composite part that contains the nuts and studs is created.

4. Expand the **Geometry** > **Parts** > **Assembly 1** > **Composite** node.



Note: To reverse this process, you can explode the composite part.

Give the assembly and composite parts more suitable names:

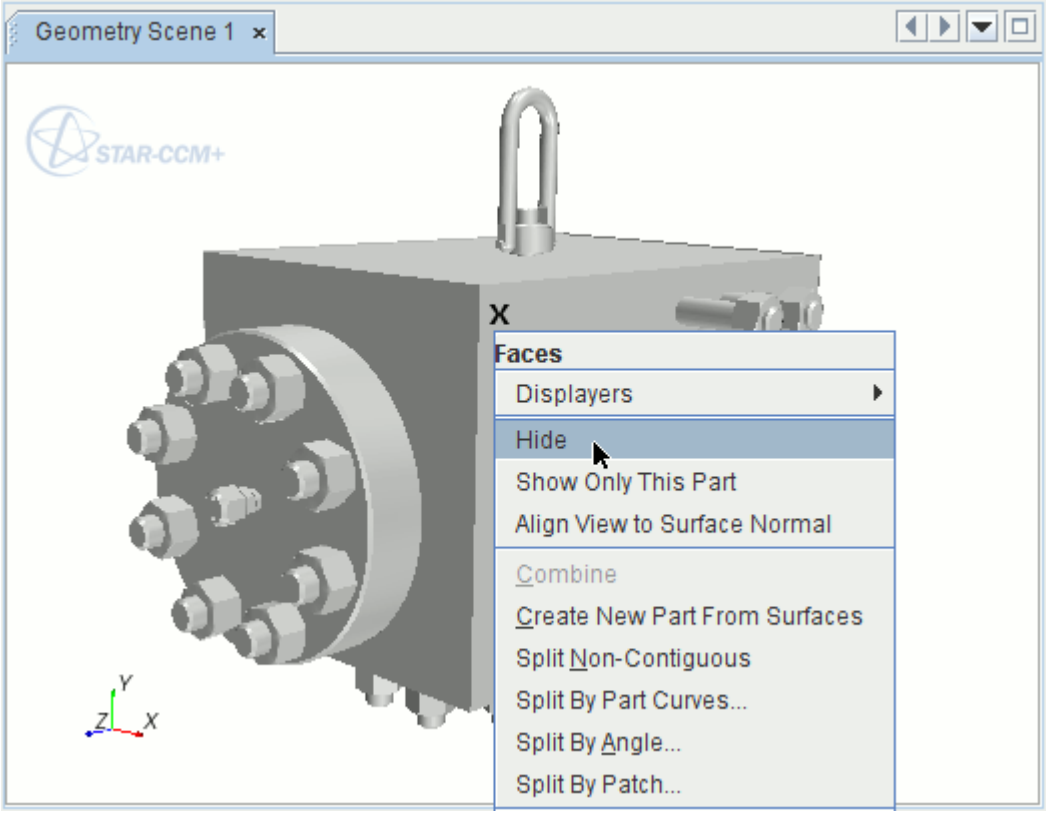
5. Rename the **Assembly 1** node to `Sandcatcher`.
6. Rename the **Composite** node to `Nuts/Studs`.

Re-Tessellating Parts

It is possible to change the surface triangulation and edge definition of a part. Use this capability to generate a fine tessellated surface on the **FlowCage** part.

First, display only the **FlowCage** in *Geometry Scene 1*:

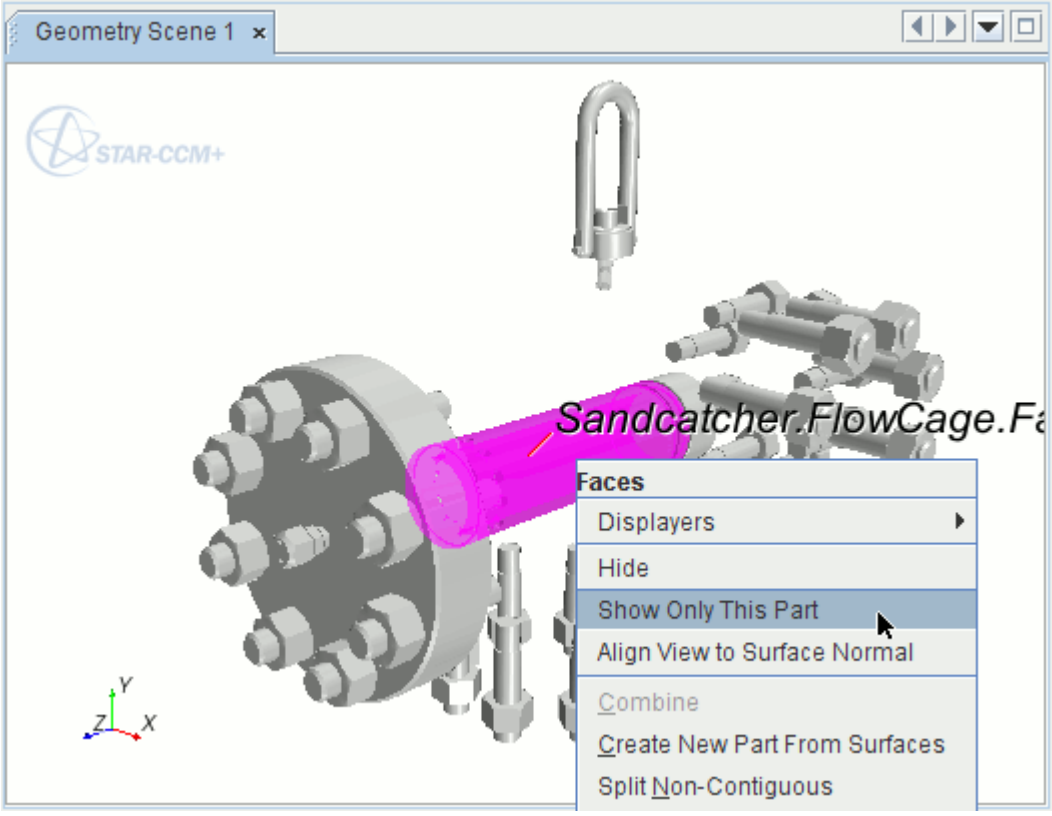
1. Right-click the **Block** part, which is marked using an X in the screenshot below, and select **Hide**.



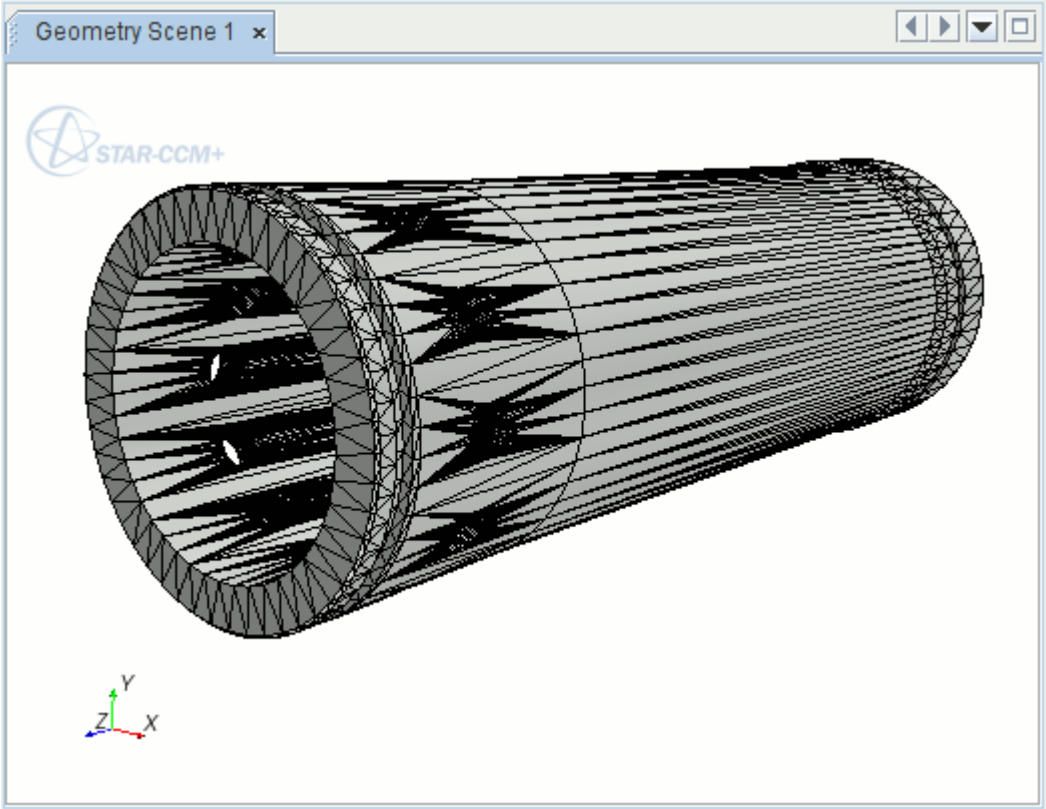
The selected part is hidden from view. To restore hidden parts at any time, right-click on the white area of the scene and select **Restore Hidden Parts...**.

Rotate the model so that the internal parts are clearly visible:

2. In the *Graphics* window, right-click the **FlowCage** and select **Show Only This Part**. The **FlowCage** is highlighted in the following screenshot.



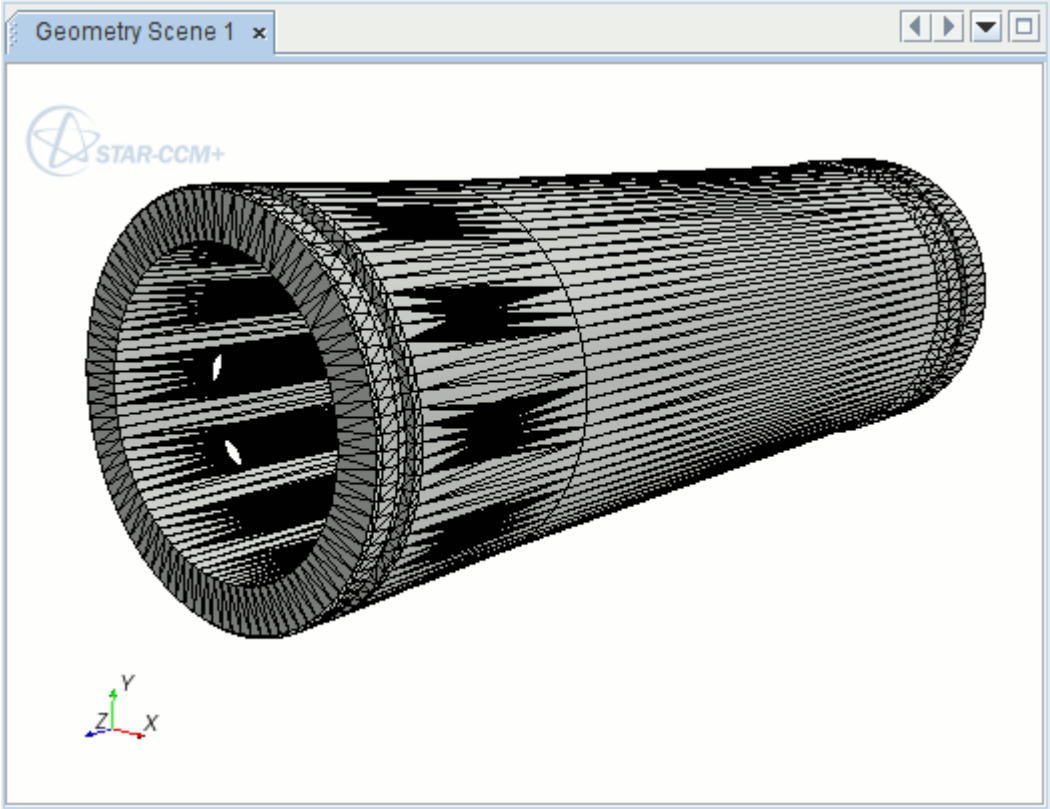
3. In the *Vis* toolbar, click the (**Show All Meshes**) button.
4. Position the geometry as follows:



5. Right-click the **Geometry** > **Parts** > **Sandcatcher** > **FlowCage** node and select **Re-tessellate**.
6. In the *Tessellation Options* dialog, set the *Tessellation Density* to **Fine** and click **OK**.

   After a moment, the part is regenerated with a fine tessellated surface.
7. Click in the white region of the *Graphics* window to deselect the part.

   The scene appears as follows:

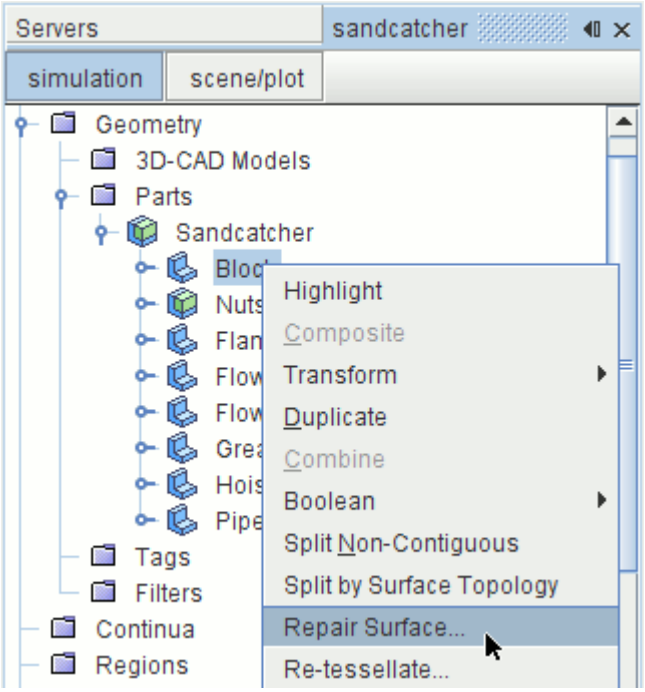


Repairing Part Surfaces

Fill the holes in the geometry using the Surface Repair mode of STAR-CCM+.

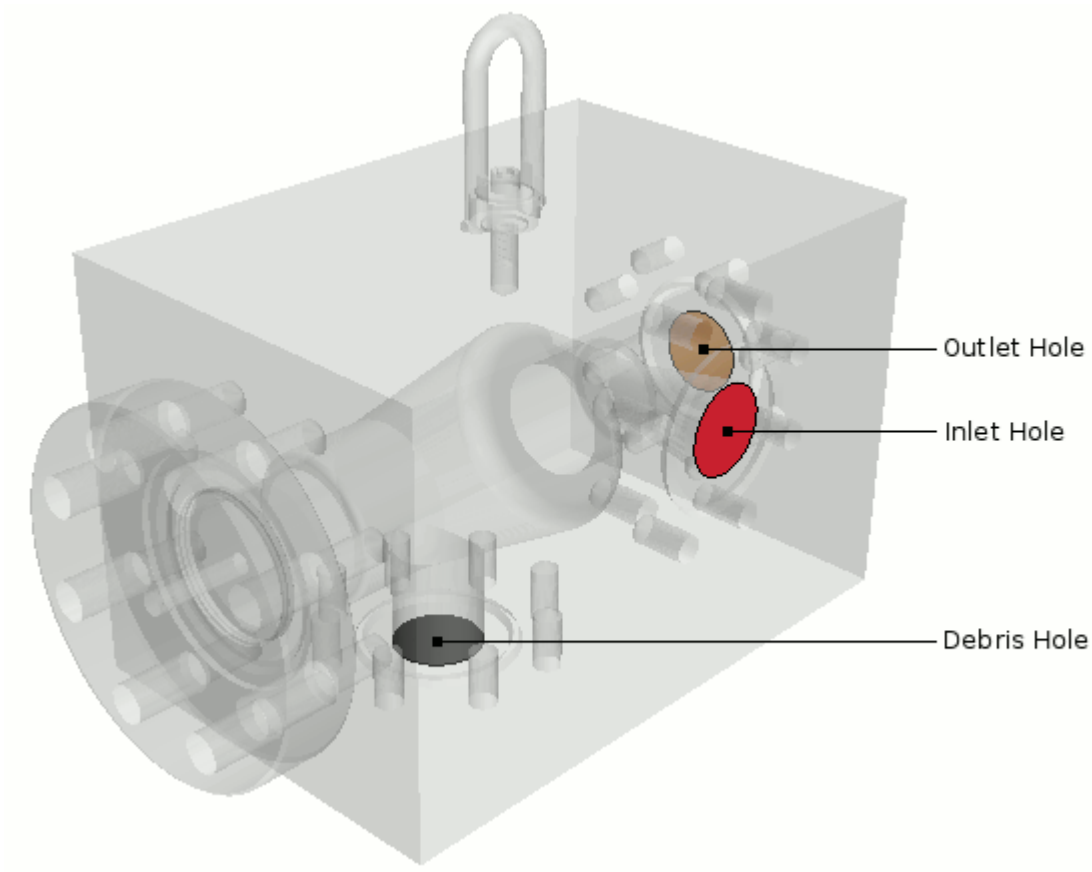
There are three holes in the main block that must be filled: the inlet, outlet, and a hole for debris. Fill the holes by repairing the surface of the block:

1. Right-click the **Geometry** > **Parts** > **Block** node and select **Repair Surface**.



2. In the *Surface Repair Options* dialog:
   a) Make sure that *Geometry Parts* is set to `[Sandcatcher.Block]`.
   b) Set *Faces to Show on Start Up* to **All Faces**.
   c) Click **OK**.

   The following diagram shows the location of the inlet, outlet, and debris holes in the block part. Fill each of the holes in turn.



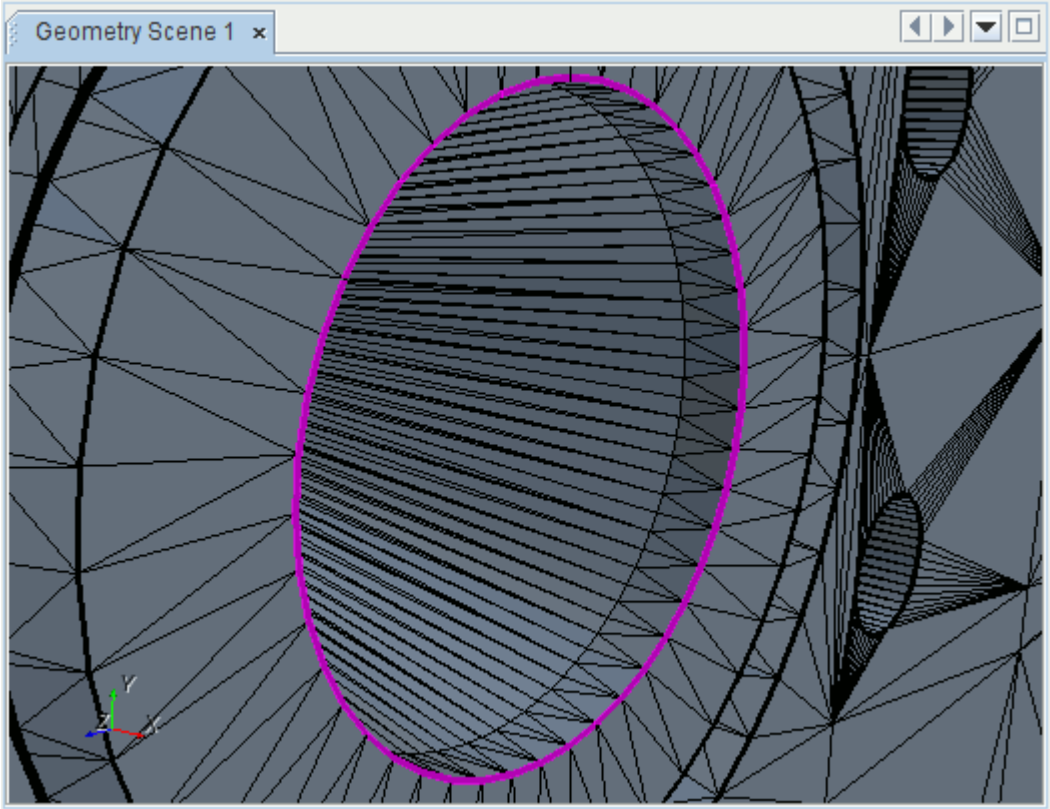
3. Zoom in on the inlet hole so that the edges are clearly visible.

Fill the holes by selecting the continuous edge around each hole and creating a patch. First, alter the Selection Controls to select edges:

4. At the bottom of the *Graphics* window, in the selection controls, deactivate the **Faces** option and activate the **Edges** option.

Click edges in the *Graphics* window to select them. To select continuous edges, double-click a single edge in the desired range. Use this technique to select the continuous edge around the hole:

5. Double-click one of the edges around the rim of the hole.

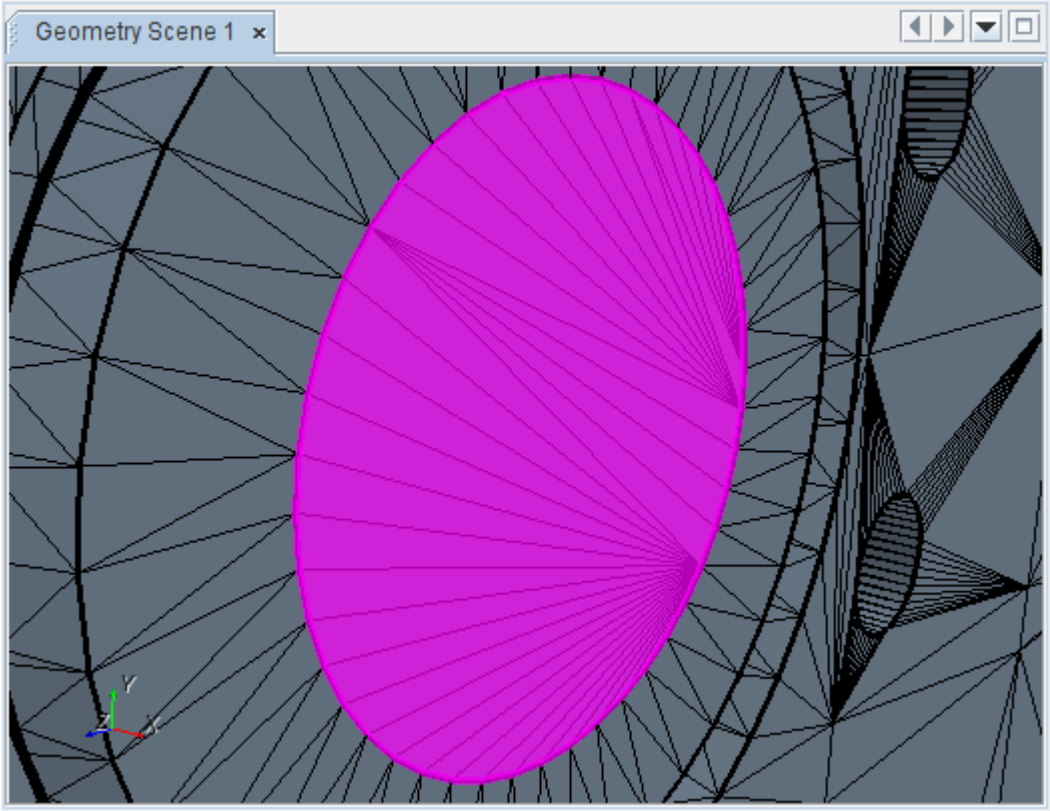


To clear any select items at any time, click (**Select None**).

The selected edges are marked with a pink line.

6. In the *Surface Repair* box of the *Repair* tab, click (**Fill holes using selected edges**).

   The hole is filled with a tessellated surface.



While using the Surface Repair tool, you can undo previous actions by pressing <`Ctrl+Z`> or by clicking the (**Undo**) button at the bottom of the *Edit* panel.

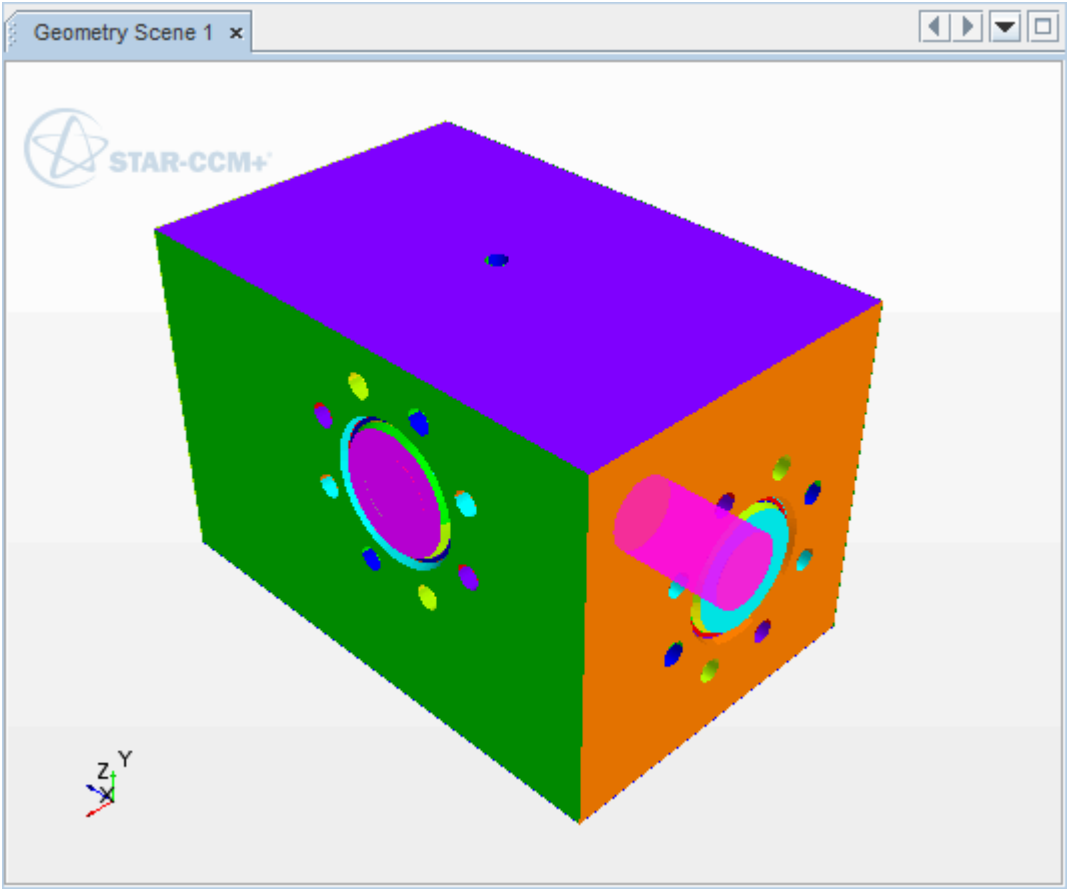
7. Use the same technique to fill the outlet and debris holes.
8. Click **Close** in the *Edit* panel.

Splitting and Combining Part Surfaces

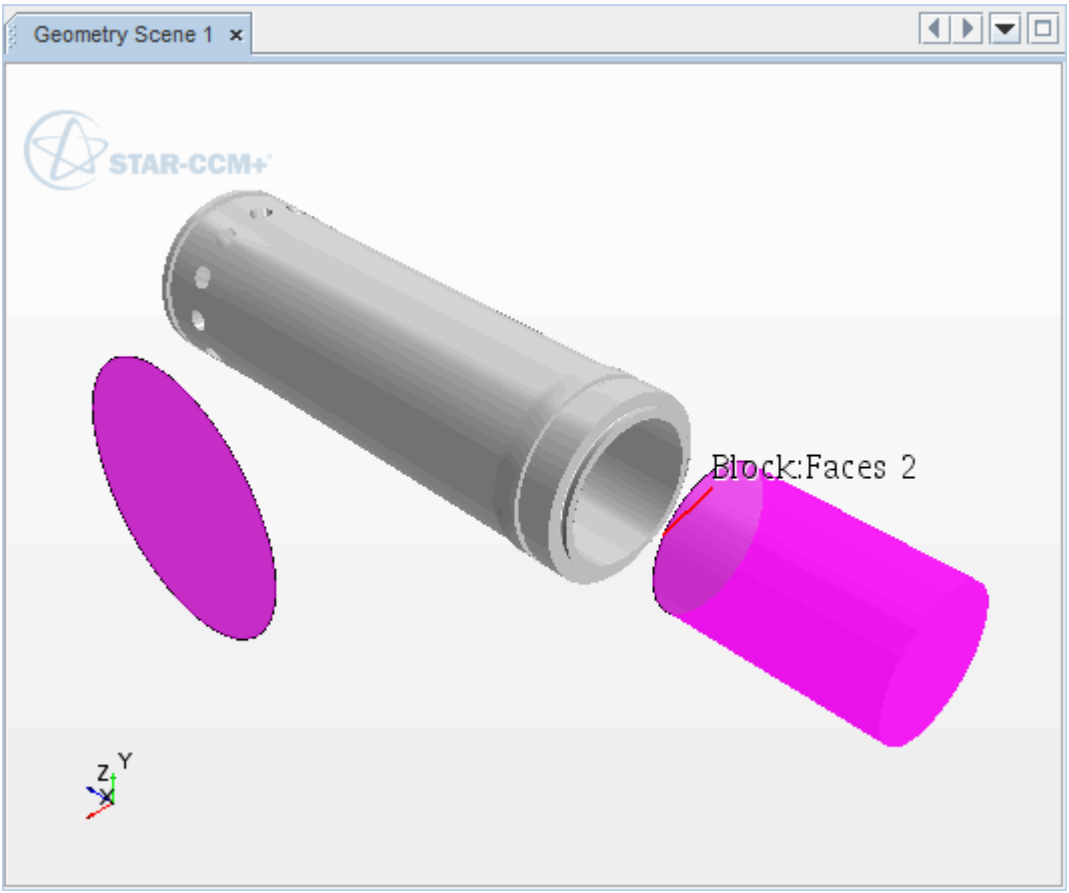
You can manipulate part surfaces in a similar way to manipulating boundaries. Split and combine various part surfaces to define the inlet and outlet faces in the block part:

1. In the *Vis* toolbar, click (**Hide All Meshes**).
2. In the *Vis* toolbar, click (**Save-Restore-Select views**) and select **Views** > **+X +Y -Z** > **Up +Y**.
3. Select the **Sandcatcher** > **Block** > **Surfaces** > **Faces** node and click (**Reset View**) in the *Vis* toolbar.
4. Right-click the **Faces** node and select **Split By Patch**.
5. In the *Graphics* window, select the faces that correspond to the inlet and outlet as shown highlighted below.

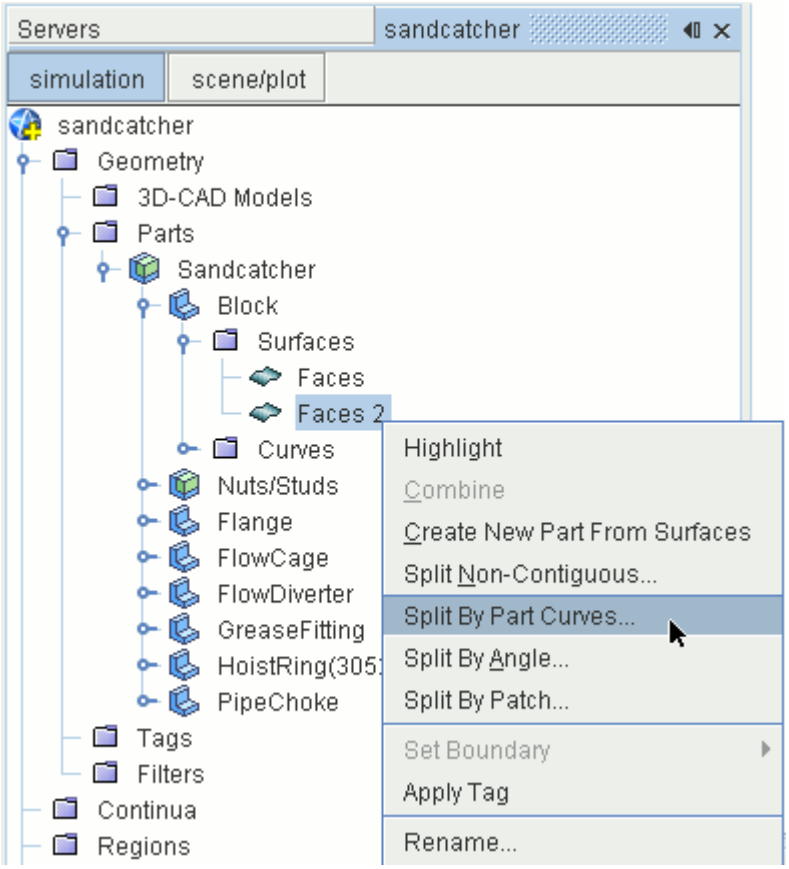
   Some of the faces next to the holes are also selected. When the holes are filled, each face is automatically joined onto an existing local patch. Further manipulation of the faces is required to split them apart and make sure that the inlet and outlet are defined using separate part surfaces.



6. Click **Create** and then **Close**.
   A surface, **Faces 2**, is created from the patches that you selected. To display the surface in the *Graphics* window:
7. Select **Surfaces** > **Faces 2**.
8. In the *Vis* toolbar, click (**Reset View**) and use the mouse buttons to position the part as shown in the following screenshot.



9. Right-click the **Faces 2** node and select **Split By Part Curves**.



10. In the *Split Part Surfaces by Part Curves* dialog:
    a) Set *Part Surfaces* to **Faces 2**.
    b) Set *Part Curves* to **Edges**.
    c) Click **OK**.

    Manipulate the created surfaces to achieve a single surface for the main body of the block, and separate surfaces for the inlet and outlet.
11. Holding down the <**Ctrl**> key, select the **Faces**, **Faces 2 2**, and **Faces 2 3** nodes. Right-click the **Faces** node and select **Combine**.

    Note: The surfaces are combined under the name of the first node that you select, which in this case, is **Faces** .
12. Rename the **Faces 2 2** node to `Outlet`.
13. Rename the **Faces 2 4** node to `Inlet`.

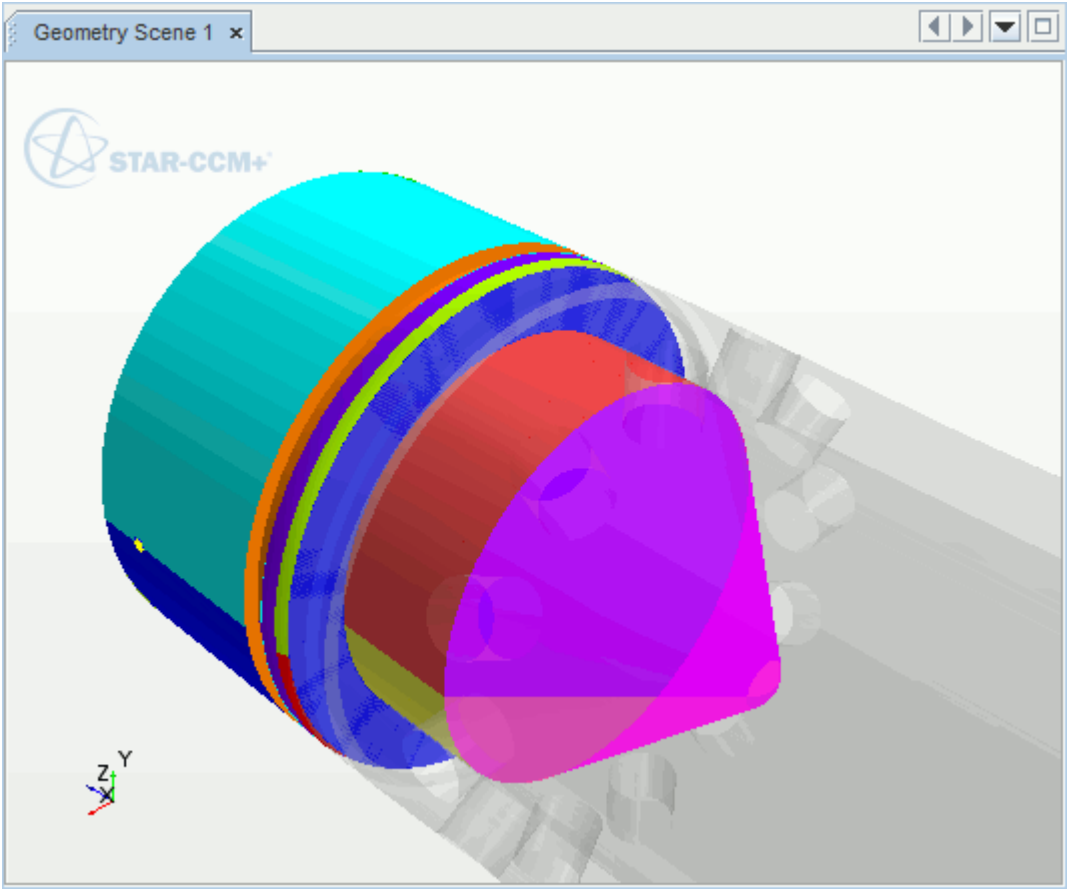
Creating a Part from Part Surfaces

Create a part from the part surfaces.

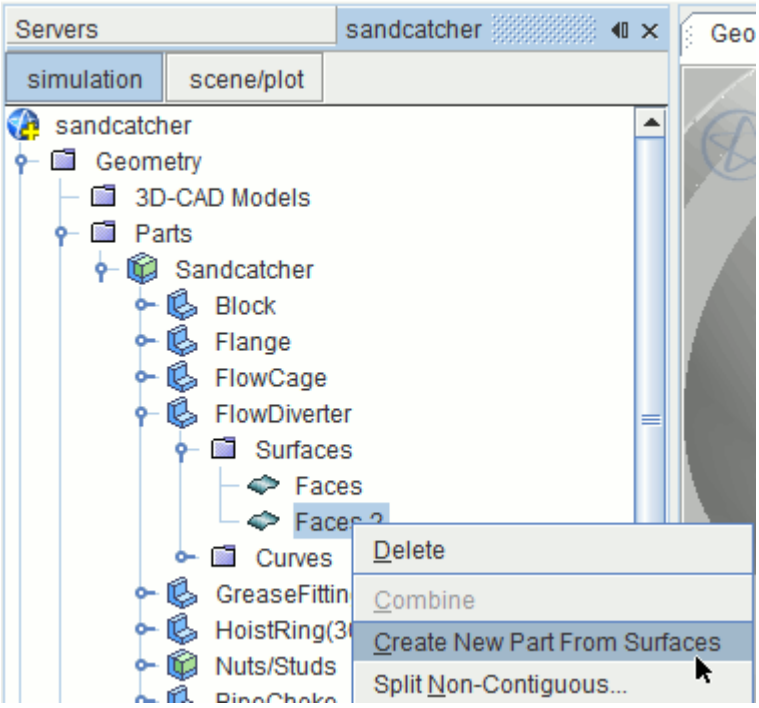
Before you create the parts, you must first split the existing part it into separate parts. You then create parts from the separated surfaces. Use the following technique to split the **FlowDiverter** into two parts.

1. Right-click the **FlowDiverter** > **Surfaces** > **Faces** node and select **Split By Patch**.

2. In the *edit* panel, hold down the <**Ctrl**> key and select patches **[5132, 5138]** and **[5133, 5139]** from the *Part Surface Patches* list. These patches correspond to the funnel, including the tip.



3. Click **Create** then **Close**.
4. Right-click the **Faces 2** node and select **Create New Part From Surfaces**.



5. Rename the **Sandcatcher** > **Part** node to `FlowDiverterFunnel`.
6. Rename the **FlowDiverterFunnel** > **Surfaces** > **Faces 2** node to `Faces`.
7. Save the simulation.

Creating a Tag

Create a tag that is applied to all parts that are in contact with the fluid.

Tags allow you to have greater control over the selection of parts when setting up a simulation. They allow you to filter "tagged" parts in selection dialogs, a feature that is useful when working with models that have many parts. Any number of tags can be created in a simulation.

To create a tag:

1. Right-click the **Tools** > **Tags** node and select **New Tag**.
2. Rename the **Tags** > **Tag** node to **FluidContact**.

Applying a Tag to a Part

Tag the relevant parts with the **FluidContact** tag using three different techniques.

1. Expand the **Geometry** > **Parts** > **Sandcatcher** node.
2. Multi-select the **FlowCage**, **FlowDiverter**, and **FlowDiverterFunnel** nodes.
3. Right-click a selected node and choose **Set Tags** > **FluidContact**.
   A small tag icon has appeared next to each of the tagged parts to indicate that these parts are in contact with the fluid. The **FlowDiverter** is not in contact with the fluid, so remove the tag.

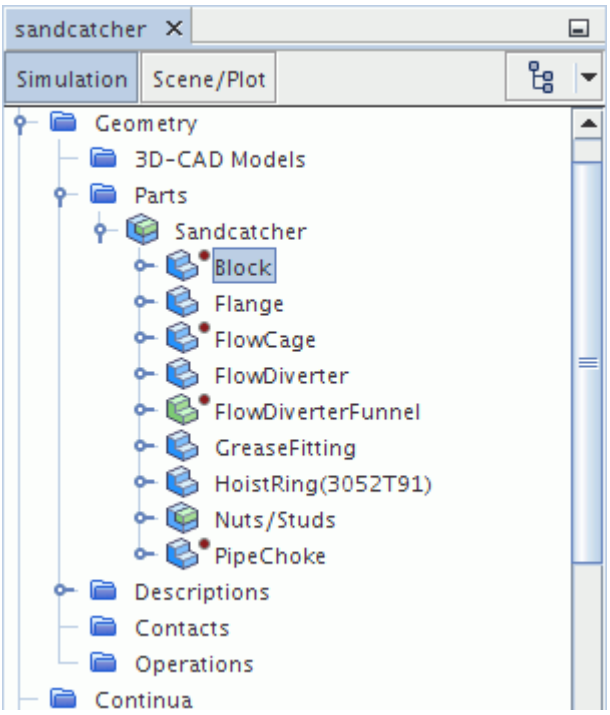
4. Select the **FlowDiverter** node. In the *Tags* property, deselect **FluidContact** from the drop-down list.

Use another method to apply the tag to the pipe choke part.

5. Expand the **Sandcatcher** and **Tags** nodes. Drag-and-drop the **FluidContact** node onto the **PipeChoke** node.

Apply the tag to the block part:

6. Select the **Block** node and set *Tags* to **FluidContact**.
   The simulation tree appears as follows:



7. Save the simulation.

Setting Part Metadata

Specify metadata for the location of some of the parts in the model by using the name 'position' and the values 'internal' and 'top'.

Metadata is useful for specifying the properties of parts, and can be used in a similar way to tags. To apply metadata you provide a suitable name, for example 'material' and a suitable value corresponding to this name for the relevant parts, for example 'steel' or 'Al-alloy'.

To apply metadata:

1. Multi-select the **FlowCage**, **FlowDiverter**, **FlowDiverterFunnel**, and **PipeChoke** nodes.
   The common properties of the selected objects are displayed.
2. Click **{}** to the right of the *Metadata* property and type `{position:internal}`.
3. Press <`Enter`> to confirm the value.

4. Select the **HoistRing(3052T91)** node and set *Metadata* to `{position:top}`.

Setting Up Filters

Create two filters.

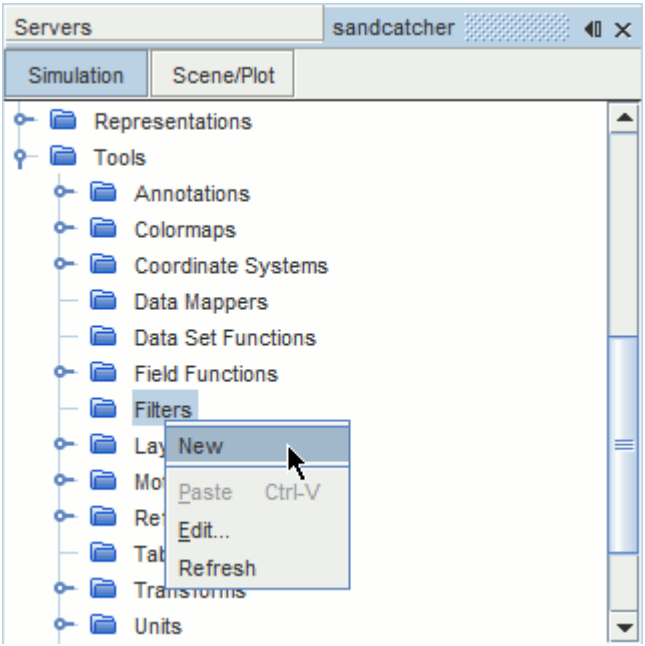
Filters are based on tags and metadata and they allow more rapid selection of related parts when setting up a simulation.

Create two filters to allow you to select:

- Any parts that are in contact with the fluid
- Only internal parts that are in contact with the fluid

To create the filters:

1. Right-click the **Tools** > **Filters** node and select **New**.



2. In the *New Filter* window, select **Tag** from the drop-down menu.
3. From the new drop-down menu that appears, select **contains**.
4. Click ... (**Custom Editor**) .
5. In the *Select Object* dialog, select the **FluidContact** tag.
6. Click **OK** to close the *Select Object* dialog.

   The filter is added to the **Filters** node.
7. Rename the new **Filter** node to `FluidContact`.

   The first filter is now complete.
8. Close the *FluidContact* window.

To create the second filter, you can make a copy of the **FluidContact** tag filter, which shows only those parts that are in contact with the fluid, and add a metadata filter to show only the internal parts:

9. Make a copy of the **FluidContact** filter and rename it to `InternalContact`.
10. Right-click the **InternalContact** filter and select **Edit...**.
11. In the *InternalContact* window, click (**Add a new item**).
    A new entry is added to the filter.
12. From the new drop-down menu, select **Metadata**.
13. From the second drop-down menu, select **Contains**.
14. In the first text field, enter `position`.
15. In the second text field, enter `internal`.
16. Press `<Enter>`.
17. Close the *InternalContact* window.
18. Save the simulation.

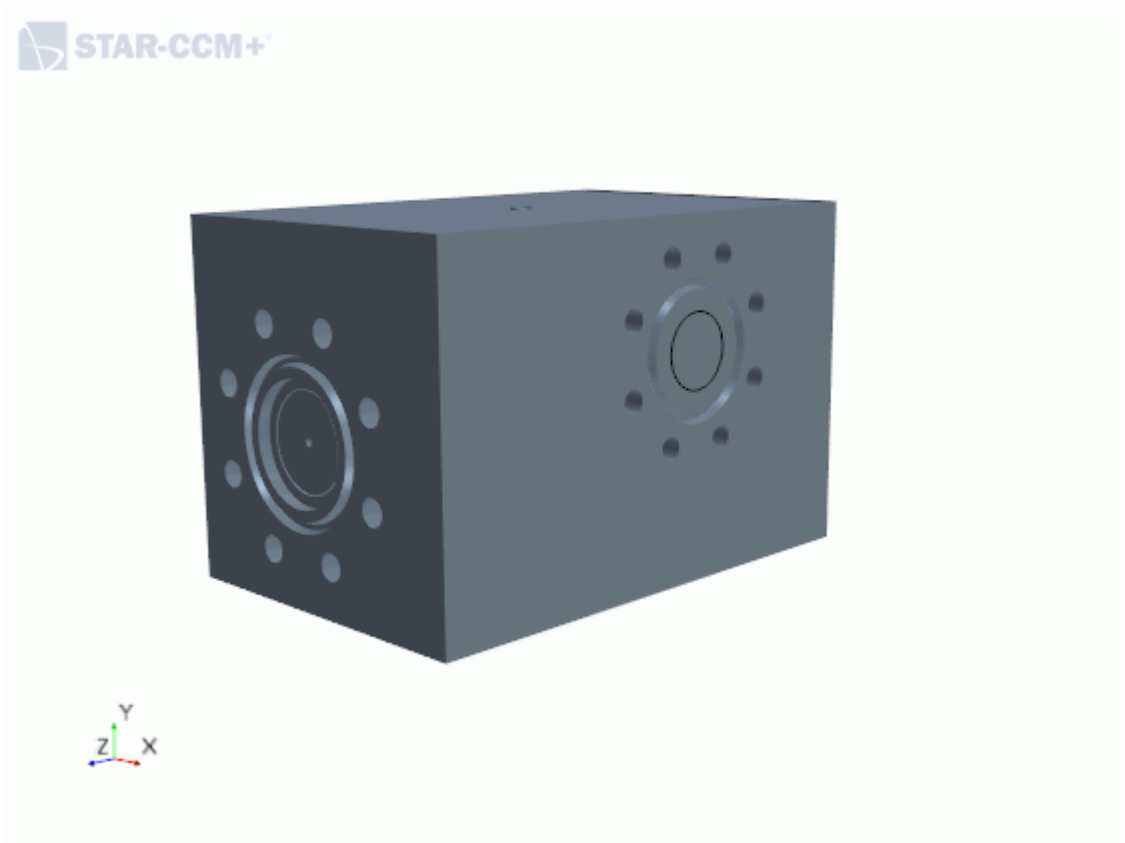
Assigning Parts to Regions

Assign parts to a region using two methods.

In the first method, create a region and assign a number of parts to that region directly through the **Geometry** > **Parts** node.

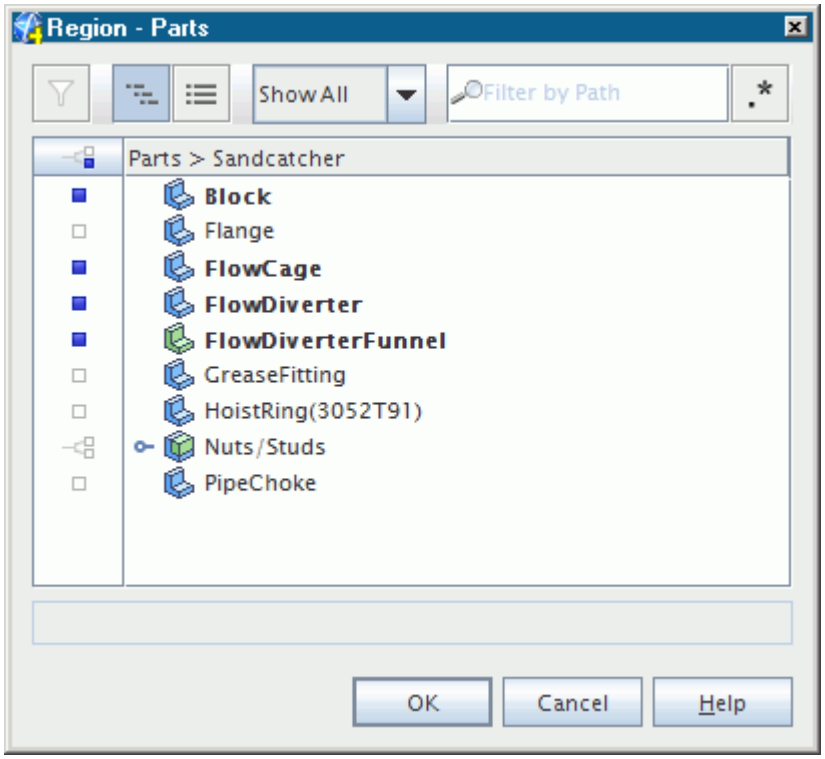
1. Expand the **Geometry** > **Parts** > **Sandcatcher** node.
2. Multi-select the **Block**, **FlowCage**, **FlowDiverter**, and **FlowDiverterFunnel** nodes.
3. Right-click a selection and choose **Assign Parts To Regions**.
4. In the *Assign Parts to Regions* dialog:
   a) Select **Create a Boundary for Each Part Surface**.
   b) Leave all other options unchanged.
   c) Click **Apply** then **Close**.
5. In the *Vis* toolbar, click (**Reset View**).
6. Deactivate the selected parts by clicking anywhere in the white region of the Graphics window.
7. Expand the **Regions** > **Region** > **Boundaries** node.
   A region was created and given the default name, **Region**, and the surfaces that are associated with the parts that were selected have been assigned to boundaries. Again, the part names have been preserved.
8. Click and drag the **Region** node onto the blank area in *Geometry Scene 1*, and select **Part Displayers** > **Add to Geometry 1**.

   Note: When a part is assigned to a region, the color changes from light gray to dark gray.
9. Click a blank area in *Geometry Scene 1* to deactivate all selections in the scene.
10. In the *Vis* toolbar, click (**Save-Restore-Select views**) and select **Views** > **+X+Y+Z** > **Up +Y**.
11. Use the mouse buttons to position the part as shown in the following screenshot:



Now use another method to assign an additional part to the region using one of the filters that was created previously.

12. Select the **Region** node and click the ... (**Custom Editor**) button to the right of the **Parts** property.
    The *Region - Parts* dialog appears.



13. At the top left of the *Region - Parts* dialog, click the **Show/Hide Filters** button.

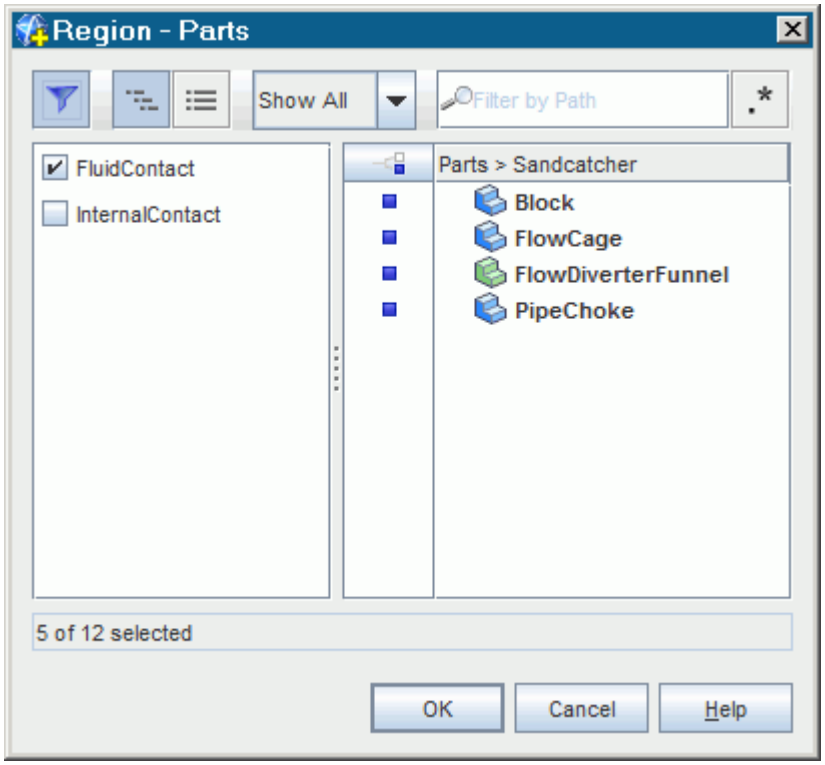
    The filters are shown in the *Filters* box to the left of the dialog.

14. Select the **FluidContact** filter.

    The contents of the *Tree View* window shows only those parts that are in contact with the fluid.

15. Select **PipeChoke**.

    All the parts that are in contact with the fluid are now selected.



16. Click **OK**.

    The **PipeChoke** part is now assigned to **Region**.

Assigning Part Surfaces to Boundaries

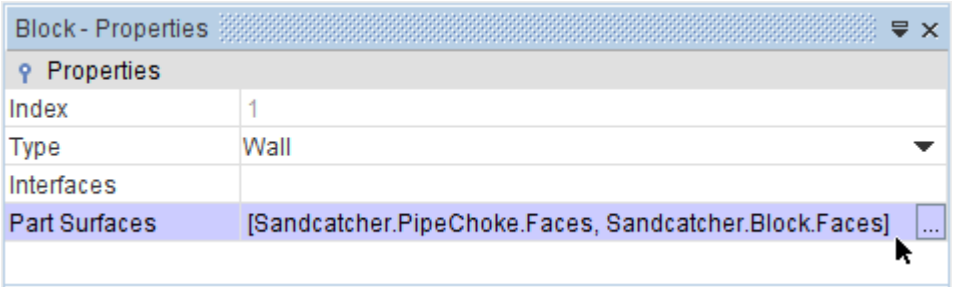
Assign the part surfaces to boundaries.

When the **PipeChoke** part is assigned to **Region**, the associated surfaces are assigned to the default boundary, which in this case is **Block**.

Create a boundary for the **PipeChoke** part surfaces.

1. Select the **Region** > **Boundaries** > **Block** node and note the value of the *Part Surfaces* property. Resize the window if necessary.

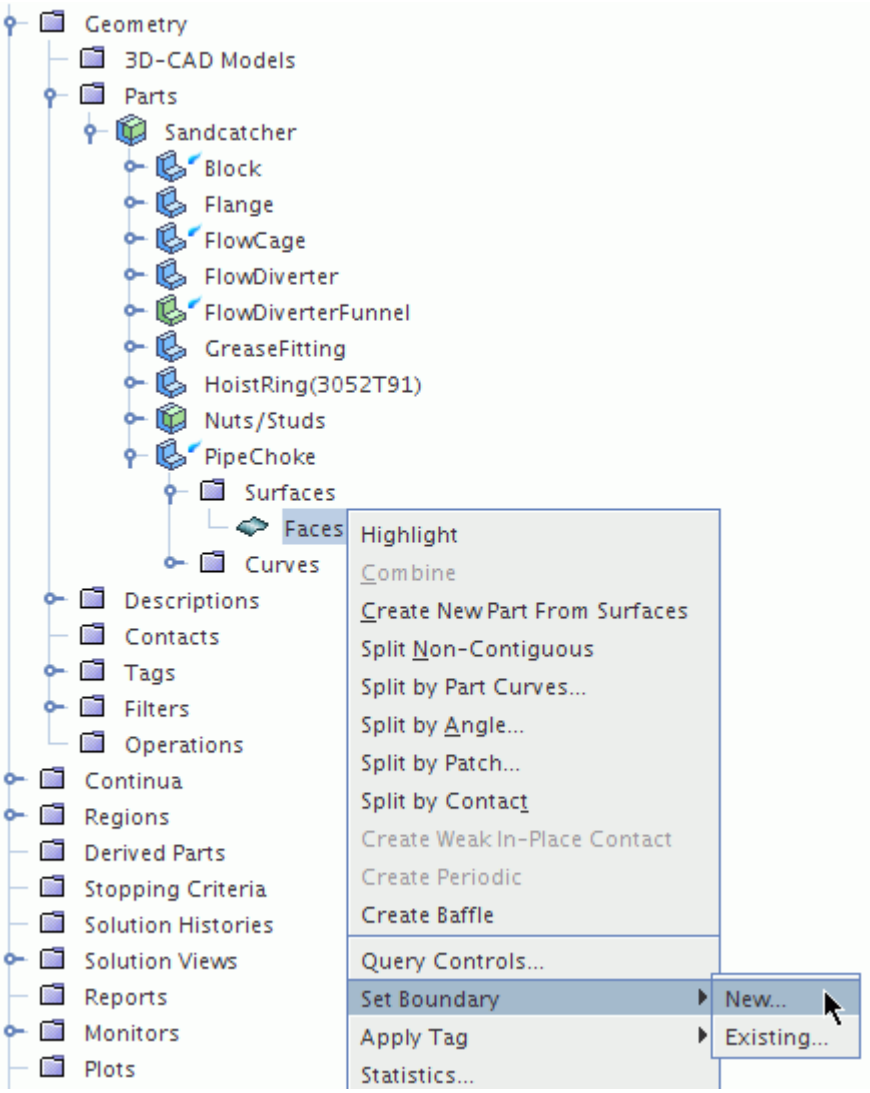
   The **PipeChoke** surface is currently associated with the **Block** boundary.



   To create a separate boundary for the **PipeChoke** part surface:

Return to the **Geometry** > **Parts** simulation tree and use the **Set Boundary** option on the relevant part surface:

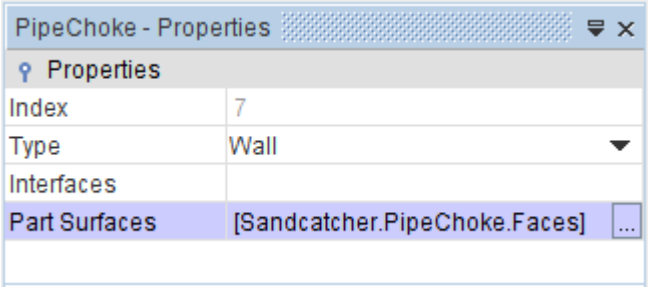
2. Expand the **Geometry** > **Parts** > **Sandcatcher** > **PipeChoke** > **Surfaces** node.
3. Right-click the **Faces** node and select **Set Boundary** > **New**.



4. In the *New Boundary* dialog, enter `PipeChoke` and click **OK**.

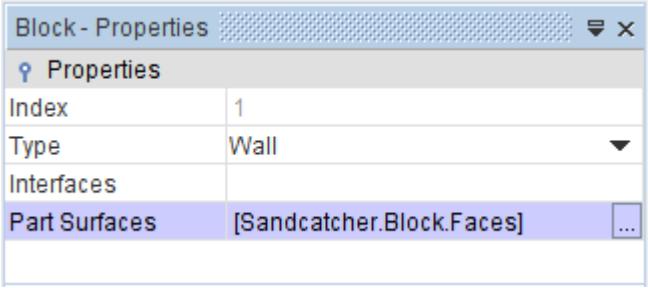
   A new node, **PipeChoke** is added to the **Regions** > **Region** node.
5. Select the **Boundaries** > **PipeChoke** node.

   Notice that the value of the *Part Surfaces* property is **Sandcatcher.PipeChoke.Faces**.



6. Select the **Region** > **Boundaries** > **Block** node.

   The **PipeChoke** part surface is automatically removed from the **Block** boundary. The only part surface that is associated with the **Block** is **Sandcatcher.Block.Faces**.



7. Save the Simulation.

Creating the Automated Mesh Operation

Select the meshing models for the simulation.

To extract the internal surface of the component, use the surface wrapper model.

1. Create a mesh continuum.
2. For the mesh continuum, **Continua** > **Mesh 1**, select the following models in order:

Group Box	Model
Surface Mesh	**Surface Wrapper** **Surface Remesher**
Volume Mesh	**Trimmer**
Optional Models	**Prism Layer Mesher** **Extruder**

3. Click **Close**.

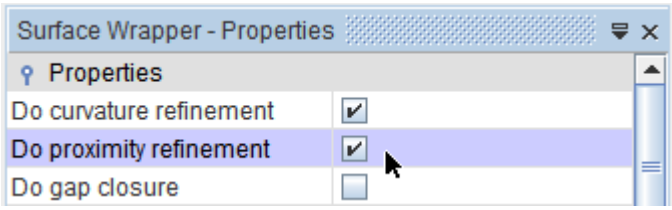
Setting Global Mesh Properties

Set the global mesh properties for the simulation.

Use the surface wrapper to generate a closed surface mesh for the internal fluid region. Use the surface remesher to retriangulate the surface and optimize it for generation of the volume mesh. Use a surface wrap that is fine enough to resolve the complex features of the geometry, use a coarser remeshed surface to reduce the final mesh cell count. To reconcile these requirements, use the wrapper scale factor. The wrapper scale factor allows you to define global mesh properties for the remeshed surface, and to specify how much finer the wrapped surface is, by setting a percentage factor. The percentage factor determines how much smaller the base size for the wrapped surface is. In this case, the length of the wrapped surface is 20% of the global size that defines the remeshed surface.

Use the proximity refinement option to refine the mesh around faces that are close together. This setting is turned on by default for the Surface Remesher and Trimmer meshing models.

1. Select the **Mesh 1** > **Models** > **Surface Wrapper** node and activate **Do proximity refinement**.



2. Edit the **Mesh 1** > **Reference Values** node.
3. Set the following properties:

Node	Property	Setting
Base Size	**Value**	`15 mm`
Prism Layer Thickness	**Percentage of Base**	`20`
Surface Proximity	**# Points in Gap**	`20`
Wrapper Scale Factor	**Percentage**	`20`

Creating Volume Shapes

Refine the mesh in two areas of the geometry that contain narrow tubes. Specify these areas by creating volume shapes, and define the mesh size with a volumetric control.

1. Right-click the **Tools** > **Volume Shapes** node and select **New Shape** > **Cylinder**.
2. In the *Create Cylinder Volume Shape* dialog, set the following properties:

Property	Setting
Start Coordinate	`[0, 0, 0.1] m`
End Coordinate	`[0, 0, 0.14] m`
Cylinder Radius	`0.1 m`

3. Click **Create**.

   Create a second cylinder with different properties in the same *Create Cylinder Volume Shape* panel:

4. In the *Create Cylinder Volume Shape* dialog, set the following properties:

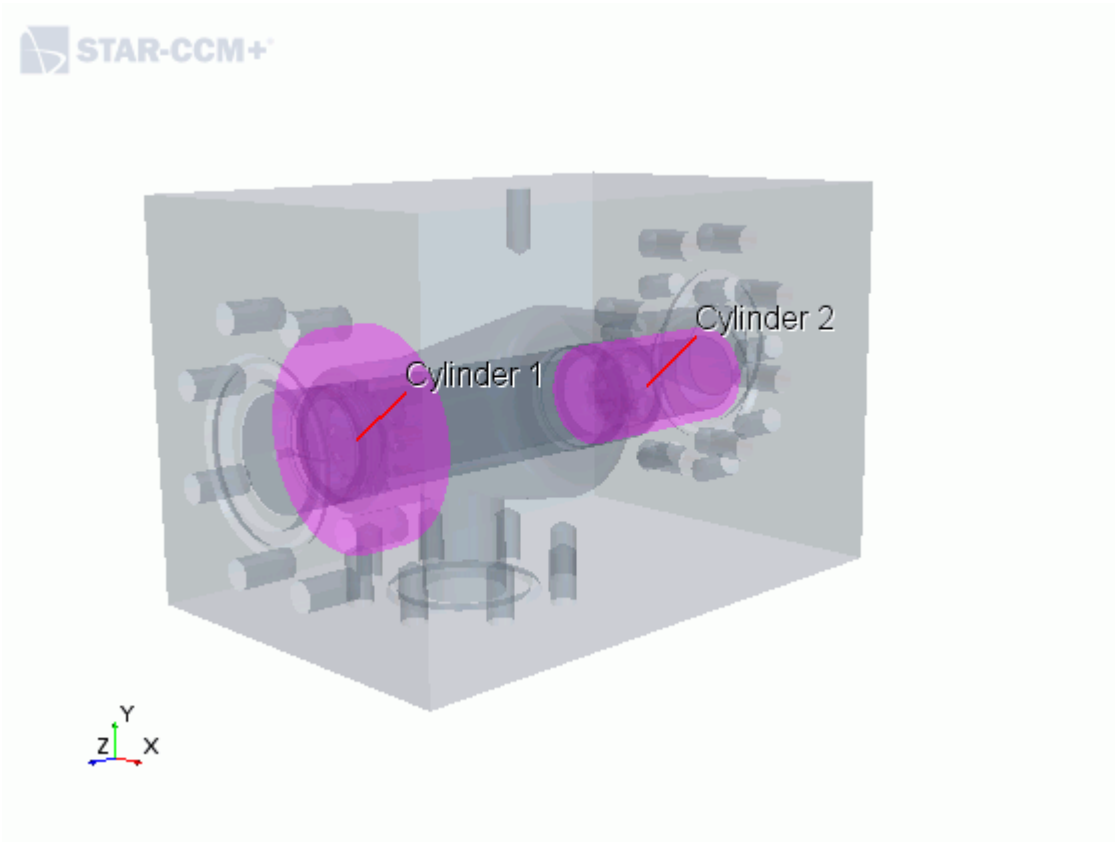
Property	Setting
Start Coordinate	`[0, 0, -0.17] m`
End Coordinate	`[0, 0, -0.35] m`
Cylinder Radius	`0.05 m`

5. Click **Create** then **Close**.

   To show the position of the two cylinders in the *Graphics* window:

6. Expand the **Volume Shapes** node and multi-select the **Cylinder 1** and **Cylinder 2** nodes.

7. In the *Vis* toolbar, click (**Make Scene Transparent**).



8. In the *Vis* toolbar, click (**Restore Scene Transparency**) button.

Creating a Volumetric Control

Create a volumetric control. The mesh refinement within each cylindrical volume is the same, so only one volumetric control is needed.

1. Expand the **Continua** > **Mesh 1** node.

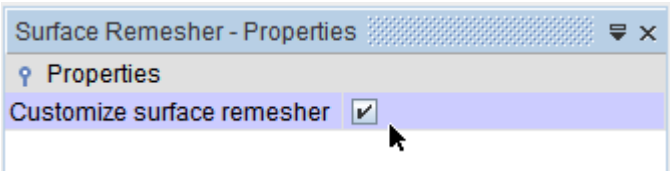
2. Right-click the **Volumetric Controls** node and select **New**.
3. Select the **Volumetric Control 1** node and click the (**Custom Editor**) button to the right of the **Shapes** property.
4. In the *Volumetric Control 1* dialog, click the **Column Selection** button so that both volume shapes are selected.
5. Click **OK**.

   By assigning the volume shapes to the volumetric control, the **Volumetric Control 1** node is activated.

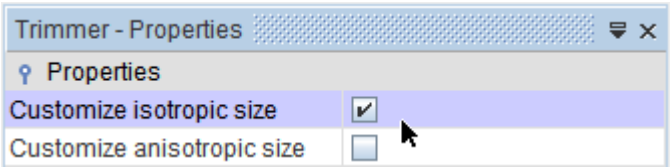
   Note: The volumetric control is not applied to the Surface Wrapper as the wrapped surface mesh has a sufficient level of refinement due to the wrapper scale factor.

Specify the mesh properties within the cylinders:

6. Select the **Volumetric Control 1** > **Mesh Conditions** > **Surface Remesher** node and activate **Customize surface remesher**.



7. Select the **Trimmer** node and activate **Customize isotropic size**.



   Define the relative size for both the remeshed surface and trimmed volume.

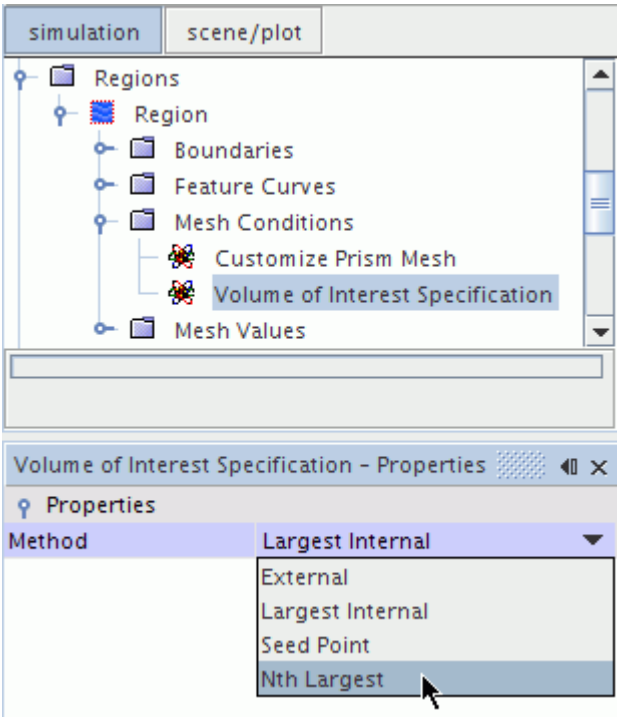
8. Select the **Mesh Values** > **Custom Size** > **Relative Size** node and set **Percentage of Base** to `15`.
9. Save the simulation.

Setting Wrapper Volume of Interest

The wrapper volume of interest is specified as a Mesh Condition for the Region that is wrapped, which in this case is **Region**.

As the fluid region is the third largest volume in the sandcatcher component, use the Nth largest option to specify the volume of interest.

1. Select the **Regions** > **Region** > **Mesh Conditions** > **Volume of Interest Specification** node and set **Method** to **Nth Largest**.



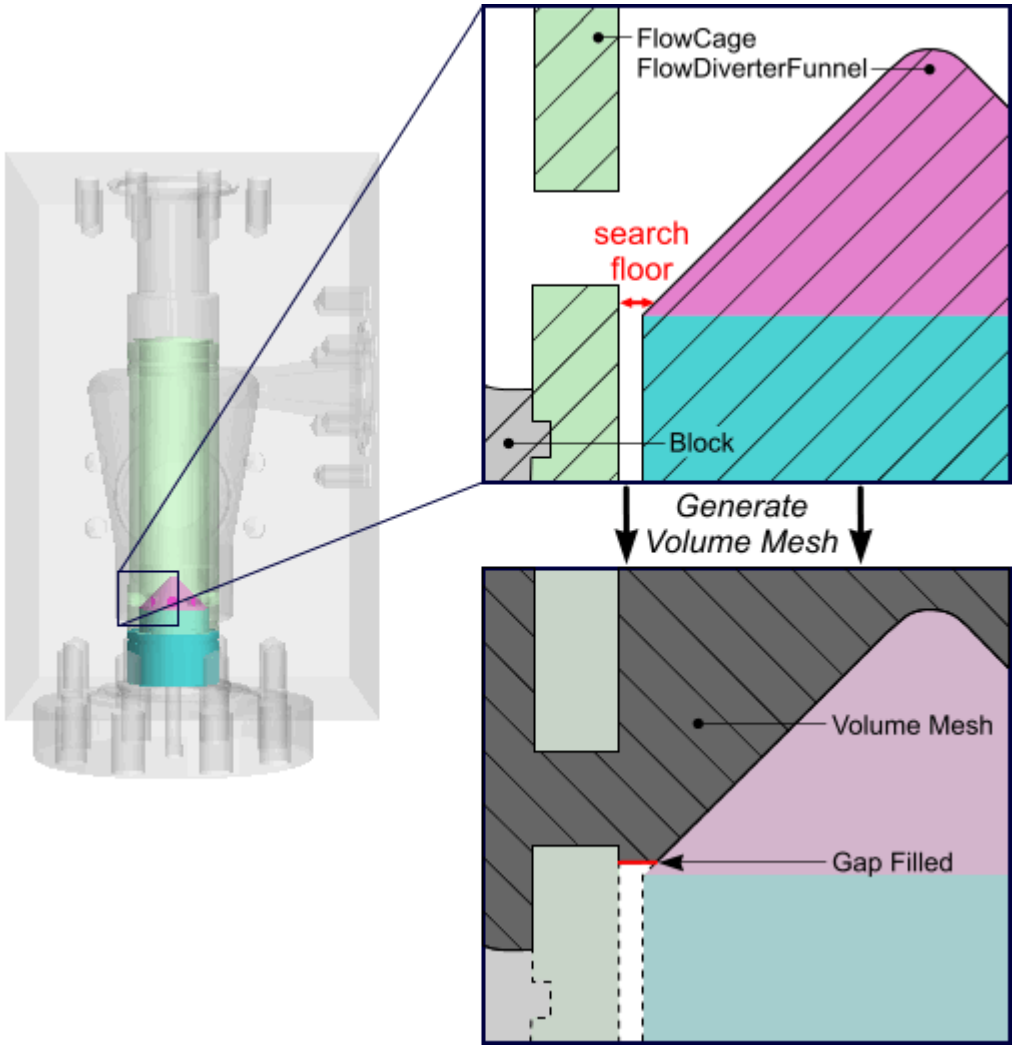
2. Select the **Mesh Values** > **Volume Number** node and set **Volume Number** to 3.

Specifying a Contact Prevention Condition

Specify a contact prevention condition to capture the geometry between the **FlowCage** and the **FlowDiverterFunnel**.

A contact prevention condition can be used to make sure that two or more surfaces that are close to one another are not joined when the surface is wrapped. This feature can be useful when there is an intricate feature in the geometry that would otherwise be lost, depending on the surface size settings provided.

Use a contact prevention condition to capture the geometry between the **FlowCage** and the **FlowDiverterFunnel** as shown in the diagram below.



When defining the contact prevention, provide a value for the minimum distance between the two surfaces at which contact prevention is guaranteed. This distance is known as the search floor. A small gap exists between the **FlowCage** and **FlowDiverterFunnel**/**FlowDiverter**. Select a value for the search floor so that the gap is filled, while capturing an adequate level of detail in the geometry.

1. Under the **Region** > **Mesh Values** node, right click **Contact Prevention** and select **New** > **One Group Contact Prevention Set**.

2. Select the **Contact Prevention 1** node and set **Search Floor** to `0.55 mm`.
3. Click the (**Custom Editor**) button to the right of the **Boundaries** property.
4. In the *Contact Prevention 1 - Boundaries* dialog, select **FlowCage** and **FlowDiverterFunnel**.
5. Click **OK**.

Setting Up Extrusions

Generate an extruded mesh at the inlet and outlet to aid solution convergence.

The Extruder meshing model was selected previously, so now specify the mesh conditions and values at the relevant boundaries.

1. Expand the **Regions** > **Region** > **Boundaries** node.
2. Set the **Block.Inlet** and **Block.Outlet** boundary types to the following:

Boundary	Type
Block.Inlet	**Velocity Inlet**
Block.Outlet	**Pressure Outlet**

3. Edit the **Block.Inlet** node and set the following properties:

Node	Property	Setting
Mesh Conditions		
└ **Extrusion Type**	*Extrusion Type*	**Constant rate normal**
Mesh Values		
└ **Normal Extrusion Parameters**	*Number of Layers*	`20`
	Stretching	`1.5`
	Magnitude	`0.16 m`
	Specify a New Region	**none**

4. Edit the **Block.Outlet** node and set the following properties:

Node	Property	Setting
Mesh Conditions		
└ **Extrusion Type**	*Extrusion Type*	**Constant rate normal**
Mesh Values		
└ **Normal Extrusion Parameters**	Number of Layers	`80`
	Stretching	`4.0`
	Magnitude	`0.9 m`
	Specify a New Region	**none**

Deleting Existing Feature Curves

It is good practice to delete any existing feature curves before using the surface wrapper to avoid them being imprinted on the wrapped surface. The surface wrapper generates a set of feature-curves that are based on the sharp edge angle that was defined in the mesh continuum.

Delete the feature curves:

1. Expand the **Region** > **Feature Curves** node, right-click **Feature Curve** and select **Delete**.
2. In the *Confirm Object Deletion* dialog, click **Yes**.

3. Save the simulation.

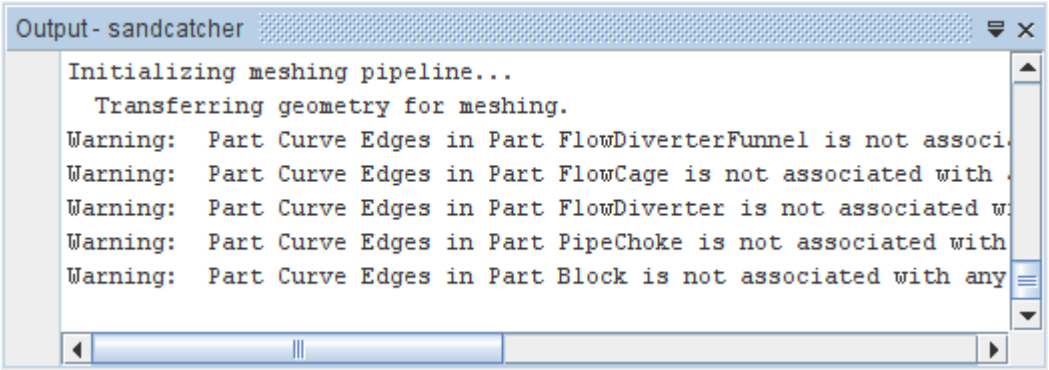
Initializing the Mesh

When working with geometry parts the mesh it is necessary to initialize the mesh before surface or volume meshing to provide the starting surface for the meshing models using the current surface and curve assignments.

In this case, a number of parts have been assigned to **Region**, which is associated with a mesh continuum, so the initial surface is comprised of only these parts.

1. In the *Mesh Generation* toolbar, click the (**Initialize Meshing**).

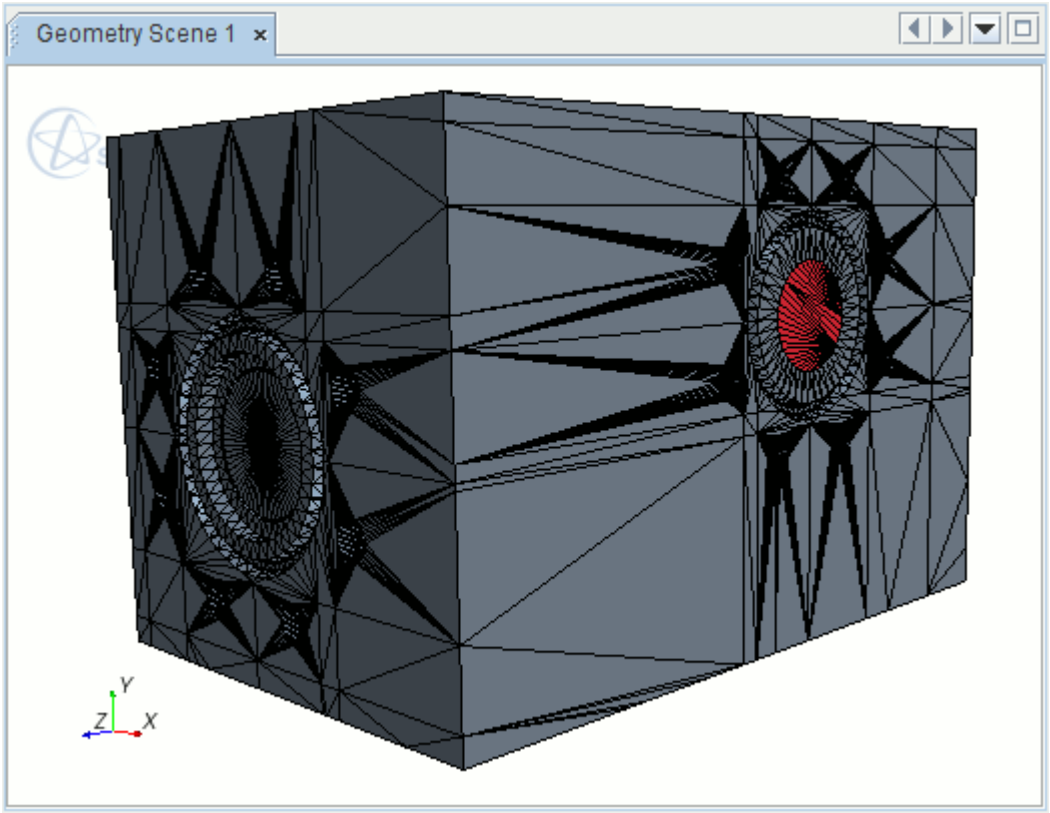
   Note: There are some warning messages in the *Output* window.



   The status message **Initializing Meshing: Finished** is displayed in the status bar when mesh initialization is complete. View the initial surface representation.

2. In the *Vis* toolbar, click (**Show All Meshes**).
3. Expand the **Scenes** node. Right-click the **Geometry Scene 1** node and select **Apply Representation** > **Initial Surface**.

The initial surface appears as follows:



Generating the Surface Mesh

The mesh parameters are fully defined and you can proceed to generate the surface mesh:

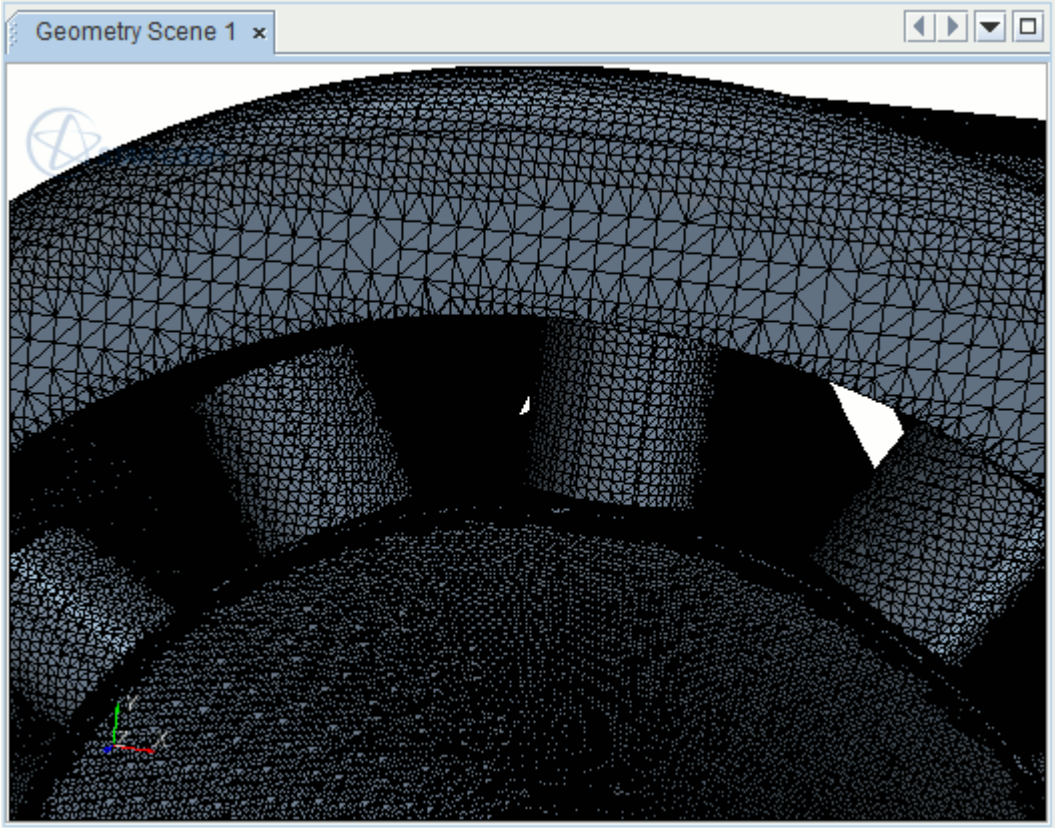
1. Click (**Generate Surface Mesh**).

   The output from the mesher is shown in the *Output* window and the ongoing status is shown in the status bar. The meshing process takes some time and is finished when the message beginning with `Surface Meshing Pipeline Completed` appears in the *Output* window.

View each of the wrapped surface mesh and remeshed surfaces in the active scene and compare the level of mesh refinement in each:

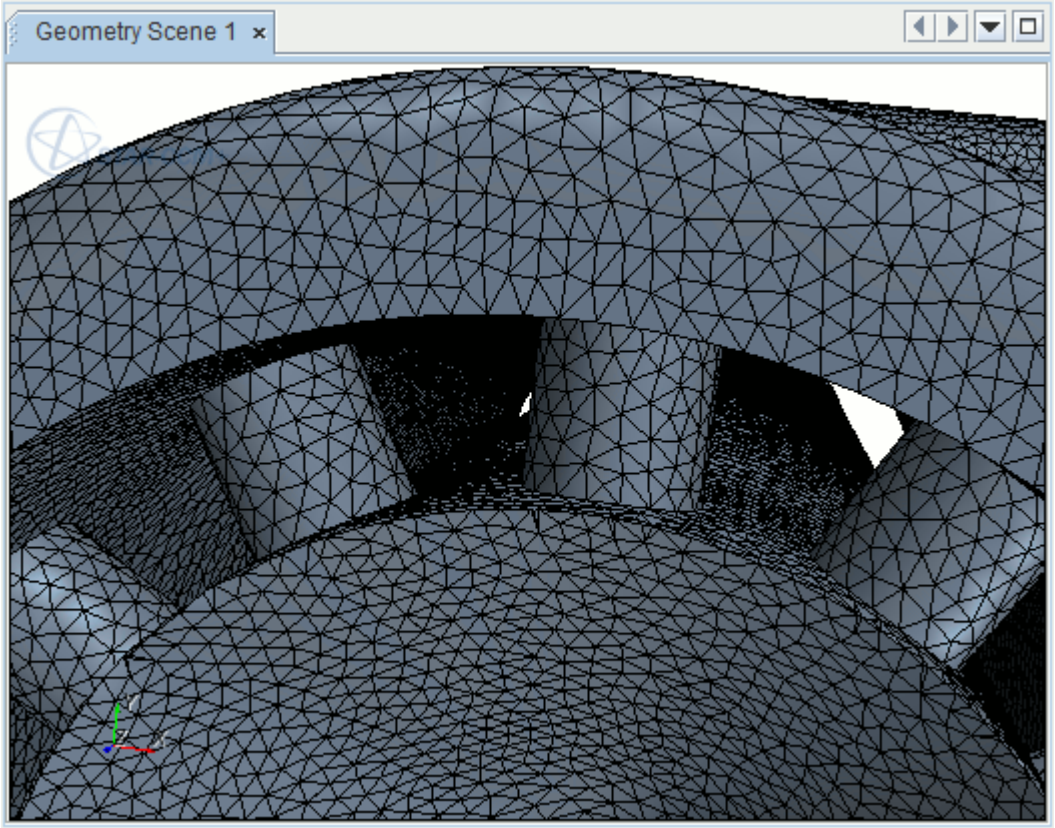
2. Right-click in the white region of the active scene and select **Apply Representation** > **Wrapped Surface**.

   The wrapped surface is displayed in the *Graphics* window.
3. Use the mouse buttons to zoom, pan and rotate the view to examine the mesh.



To view the remeshed surface:

4. Right-click in the white region of the active scene and select **Apply Representation** > **Remeshed Surface**.



5. Save the simulation.

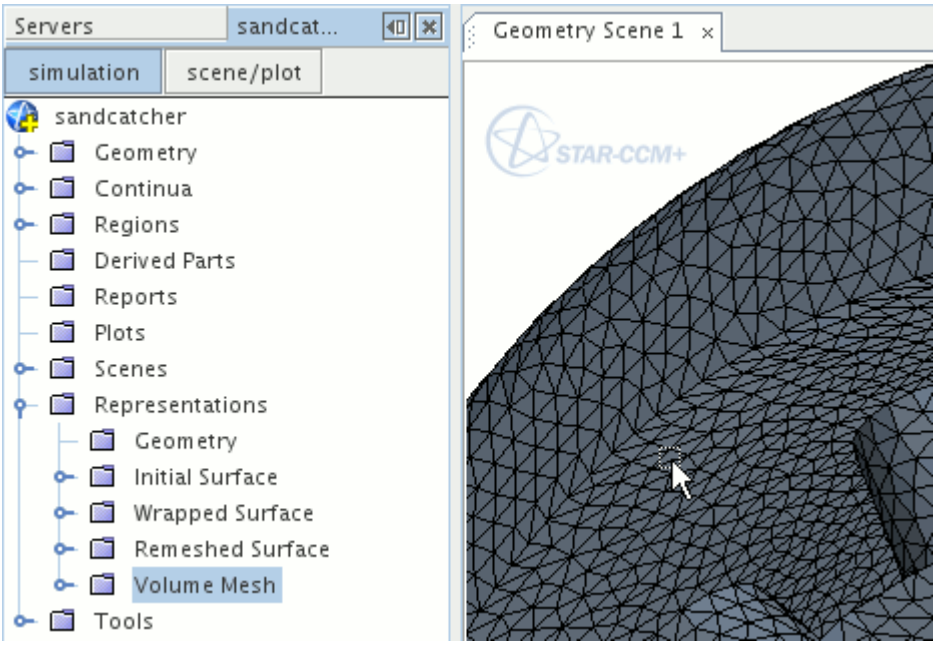
Generating the Volume Mesh

Generate the trimmed volume mesh, including extrusions at the inlet and outlet, and prism layers at wall boundaries.

1. Click (**Generate Volume Mesh**).
   The output from the meshers is shown in the *Output* window and the ongoing status is shown in the status bar. The meshing process takes some time and is finished when the message beginning with `Volume Meshing Pipeline Completed` appears in the *Output* window.

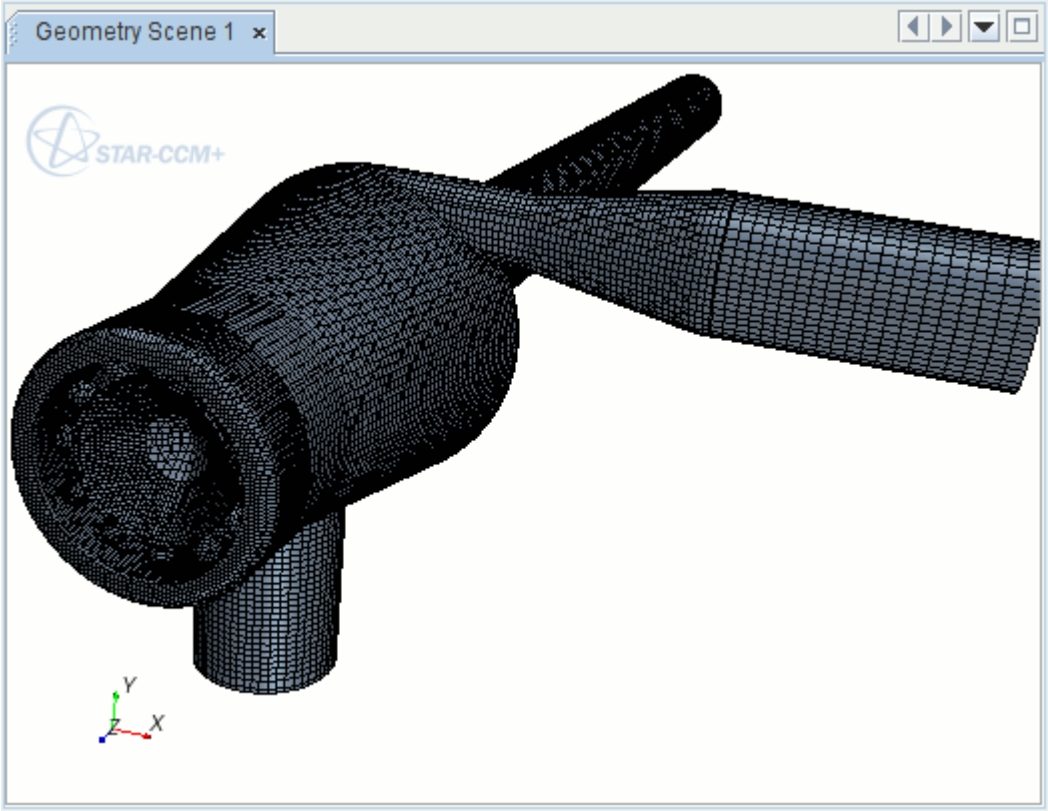
Apply the volume mesh representation to the active scene. Use an alternative method to what you used for the surface mesh representation:

2. Expand the **Representations** node.
3. Drag-and-drop the **Volume Mesh** node from the object tree into the active scene.



Update the selected parts in the geometry displayer for the current scene:

4. Expand the **Scenes** node.
5. Edit the **Displayers** > **Geometry 1** > **Parts** node.
6. In the *Edit* dialog, select all the boundaries in the region.
7. Click **OK**.
8. In the *Vis* toolbar, click the (**Reset View**).
9. Use the mouse buttons to pan, zoom, and rotate the view to examine the volume mesh.



10. Save the simulation.

Acknowledgements

The assistance of Mr. Stephen McCulloch and Mr. Jim Brinkley of Wood Group Pressure Control Ltd. is gratefully acknowledged, as is the permission of WGPC Ltd to present a modified form of the geometry.

Summary

This tutorial has introduced the following STAR-CCM+ features:

- Importing the mesh and naming the simulation.
- Reorganizing the assembly.
- Retessellating parts.
- Repairing part surfaces.
- Splitting and combining part surfaces.
- Creating a part from part surfaces.
- Creating a tag and applying it to a part.
- Setting part meta data.
- Setting up filters.
- Assigning parts to regions.
- Assigning part surfaces to boundaries.

- Creating a mesh continuum and setting up the mesh.
- Creating volume shapes.
- Creating a volumetric control.
- Setting wrapper volume of interest.
- Specifying a contact prevention condition.
- Setting up extrusions.
- Deleting existing feature curves.
- Initializing the mesh.
- Generating and visualizing surface and volume meshes.

Mesh

The tutorials in this set illustrate various STAR-CCM+ features for building CFD meshes.

Contents:

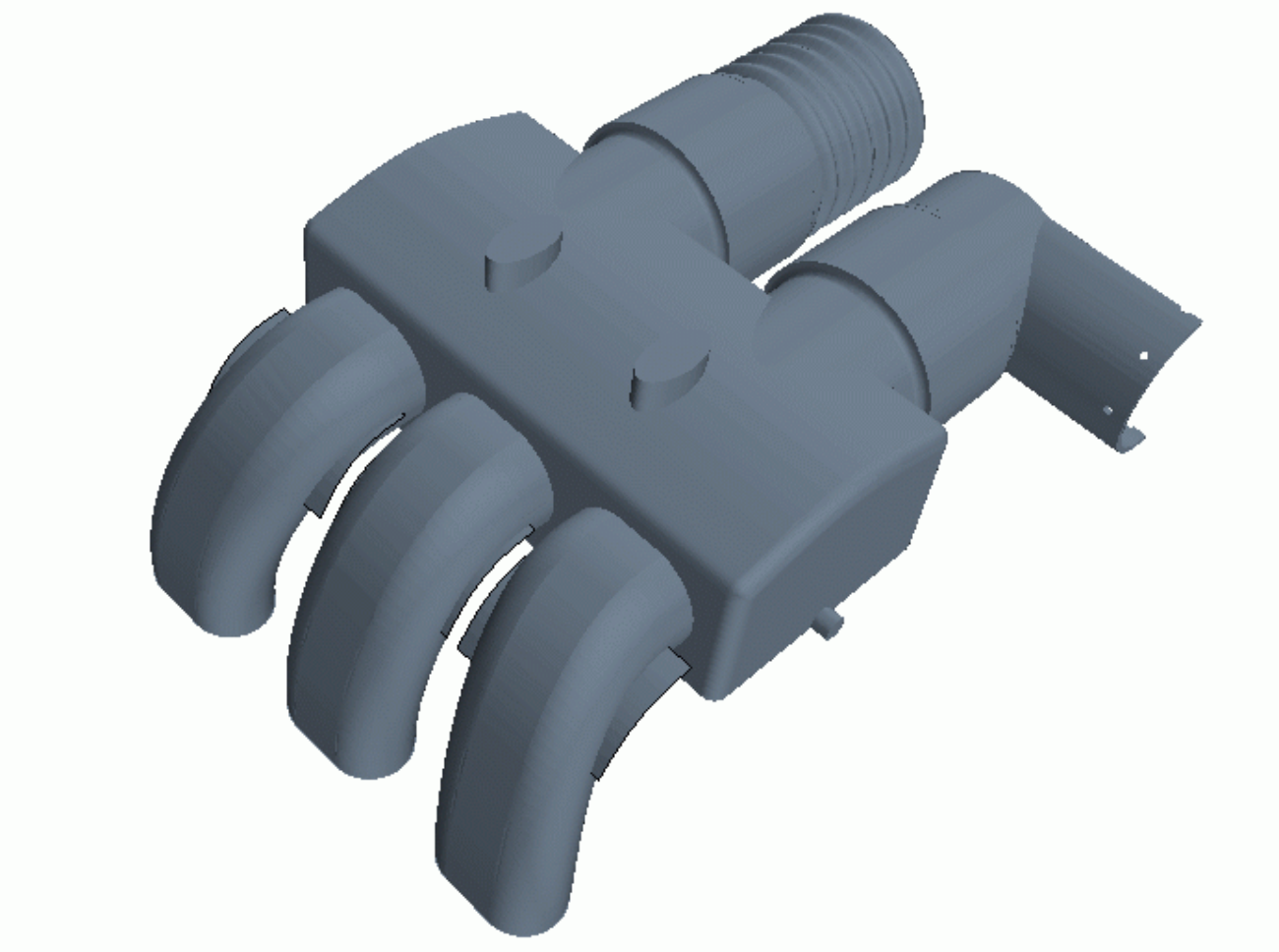
Surface Wrapper: Intake Manifold
Surface Repair Tools Brake Assembly
Parts-Based Meshing External Aerodynamics
Mesh Operations: Imported Control Valve
Meshing: Multi-Part Heat Exchanger
Directed Meshing: Electrical Machine

Surface Wrapper: Intake Manifold

This tutorial demonstrates how to use the surface wrapper to extract the internal fluid surface from a collection of parts that represent an intake manifold assembly.

The surface wrapper is run several times to present the general iterative nature of the tool when dealing with new or unfamiliar geometries. With each iteration, additional features and options are included to improve the overall fidelity of the resultant wrapped surface. With experience, you can utilize the surface wrapper and get the desired result in one or two iterations.

The geometry is shown below:



The manifold assembly is composed of several parts, including the main body, mounting struts, air-flow sensor, control valve, and directional flow baffles. Most parts are in their original CAD form and have not been prepared for use in a CFD analysis.

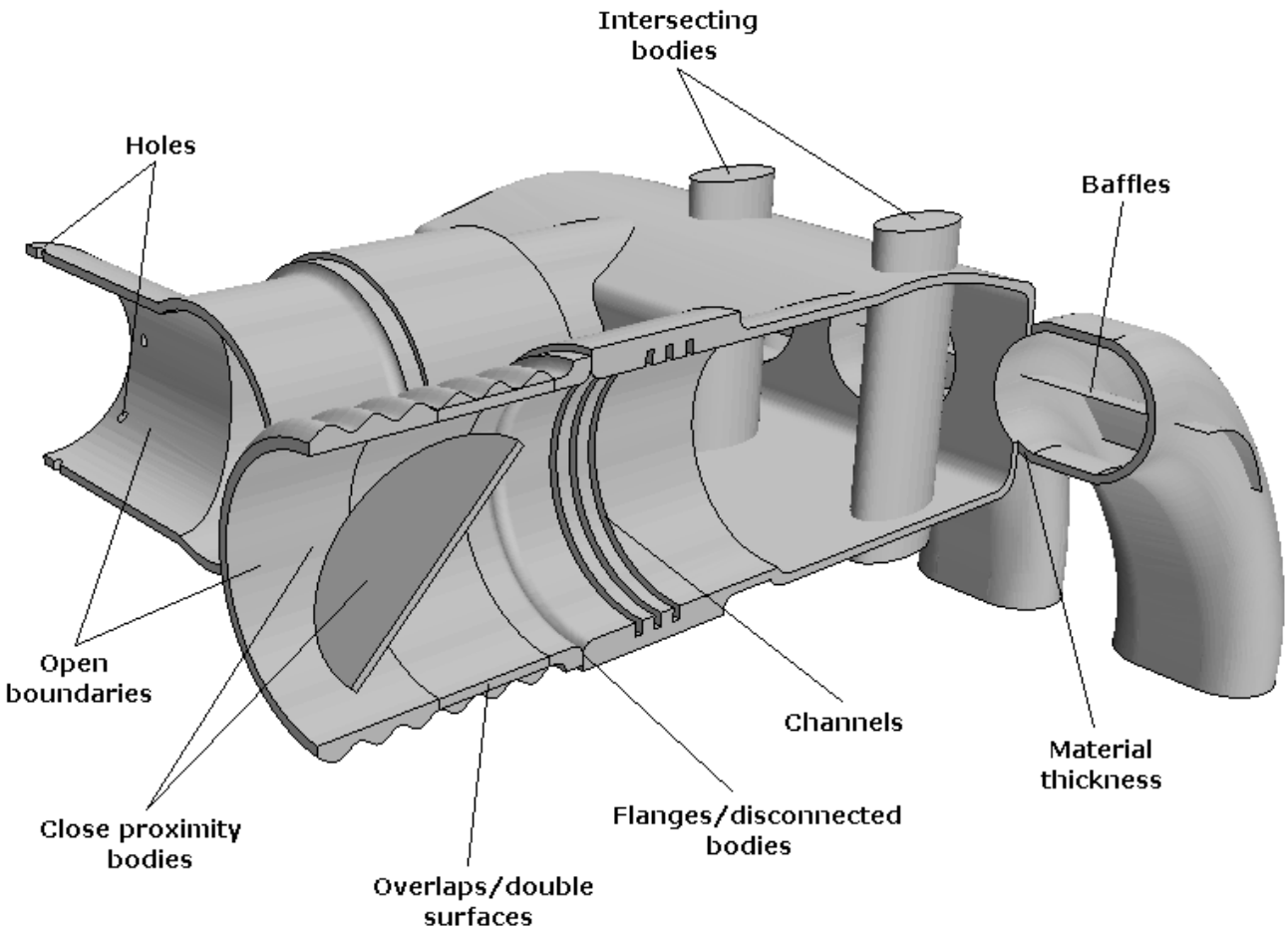
For a simple case with only a few parts, manual repair to remove unwanted surfaces and stitch together remaining ones to form a closed volume geometry, is a realistic solution. As the number of parts and overall complexity increases, however, manual repair becomes inefficient. Instead, use the surface wrapper to extract

a high-fidelity representation of the geometry. Use the representation in the surface remesher to obtain a high-quality triangulation from which the volume meshing tools can proceed.

The issues that the surface wrapper deals with in this example are:

- Material thickness
- Open flow boundaries
- Disconnected geometry/flanges
- Overlaps/double surfaces
- Intersecting bodies (struts)
- Surface holes
- Channels
- Internal thin wall baffles
- Close proximity bodies (valve, sensor)

These issues are illustrated in the cut-away diagram below:



The surface wrapper can be a fast and efficient process of extracting both internal and external surfaces from CAD type geometries. Successful use of the wrapper requires practice, due to the numerous paths available.

For example, tweaking some of the basic input values to deal with a particular feature can sometime have a significant effect on the overall result. Repeated use of the wrapper however builds up a wealth of experience.

Contents:

Prerequisites
Importing the Surface Mesh
Identifying Important Boundaries
Activating the Wrapper Model
Setting the Initial Reference Values
Capping the Geometry
Running the Leak Detector
Producing a Basic Wrap
Improving the Manifiold Body Fidelity and Curvature
Improving the Sensor Part Definition
Improving the Channels Definition
Improving the Flange Definition
Improving the Valve Definition
Improving the Baffle Definition
What to Do Next
Summary

Prerequisites

The instructions in the Surface Wrapper: Intake Manifold tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+
Parts-Based Meshing	Parts-Based Meshing: External Aerodynamics

Importing the Surface Mesh

To set up the STAR-CCM+ simulation, launch a simulation and import the supplied surface mesh.

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Surface Mesh**.
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/mesh/data`.

5. Select `baffles.x_t`, `manifold_body.x_t`, `sensor.x_t`, `struts.x_t`, and `valve.x_t` then click **Open**.
6. In the *Import Surface Options* dialog, set the following properties:

Property	Setting
Import Mode	**Create New Part**
Create Part Contacts from Coincident Entities	**Deactivated**
Tessellation Density	**Fine**

7. To accept the remaining default *Import Surface Options*, click **OK**.
8. Save the simulation as `Star1.sim`.

Identifying Important Boundaries

The import process created 11 new parts which you can combine and rename appropriately.

Merging parts allows you to rename several parts with one name. You can then rename the boundaries on the parts as necessary.

Combine and rename the parts:

1. Expand the **Geometry** > **Parts** node.
2. Multi-select **baffles1**, **baffles2**, and **baffles3**.
3. Right-click one of the selected parts and select **Combine**.
4. Rename the combined part to `Baffles`.
5. Combine and rename the remaining parts as shown in the table below:

Parts	New Part Name
manifold_body1, **valve2**	`Cross pipe`
manifold_body2	`Manifold body`
manifold_body3	`Inlet pipe`
sensor	`Sensor`
struts1, **struts2**	`Struts`
valve1	`Valve`

Activating the Wrapper Model

Close or “cap off” any open boundaries or large holes in the surface so that the wrapper can determine what is inside the geometry and what is outside.

To assist with the hole detection process, activate the surface wrapper model and specify initial surface sizes so that the leak detector can be invoked to check for large holes. The leak detection tool also contains a

subset of the surface repair tools, which you can use initially to close the gaps in the model. The wrapping process can close holes that are smaller than the specified target mesh size for each surface.

Activate the Surface Wrapper:

1. Right-click the **Geometry** > **Operations** node and select **New** > **Surface Preparation** > **Surface Wrapper**.
2. In the *Create Surface Wrapper Auto Mesh Operation* dialog, select all parts then click **OK**.

   Two **Surface Wrapper** nodes appear in their unexecuted state in the simulation tree, at the following locations:

   - **Geometry** > **Parts**

     This node represents the part that is produced by the Surface Wrapper operation.
   - **Geometry** > **Operations**

     This node represents the action of the Surface Wrapper and allows you to execute the Surface Wrapper operation.

Setting the Initial Reference Values

Values for the surface wrapper must be set with consideration of the parameters of the geometry.

To use the leak detector, specify some initial first guess reference values for the surface minimum and target sizes. For this tutorial, use relative size specification with a base size of 0.09 m, which is approximately the width of the main manifold body.

1. Navigate to **Geometry** > **Operations** > **Surface Wrapper** > **Default Controls**.
2. Set the following properties:

Node	Property	Setting
Base Size	*Value*	`0.09 m`
Target Surface Size	*Percentage of Base*	`10.0`
Minimum Surface Size	*Percentage of Base*	`5.0`

You can influence the surface wrapper with the following default controls:

CAD Projection	Allows you to turn on or off the *Project to CAD* option. Activating this option allows vertices to be projected back to the imported CAD surface at the end of the wrapping process, resulting in a surface definition closer to the original. By default, the *Project to CAD* property is activated. Retain this setting for this tutorial.
Surface Curvature	Allows the specification of the number of triangles that are produced on curved surfaces. Increasing the # *Pts/circle* value, increases the resultant triangulation in curved areas of the geometry. For this tutorial, maintain the default value of `36`.

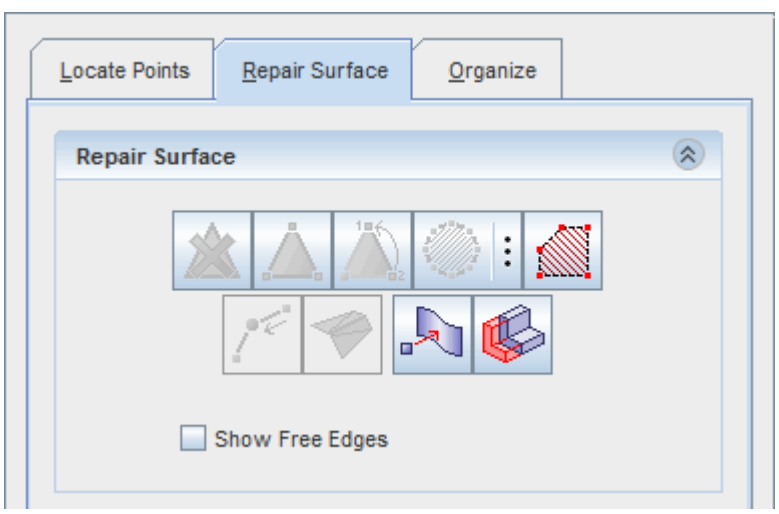
Feature Angle	Provides an angle setting for which feature curves are generated during the wrapping process. In general, pre-existing feature curves in the imported surface are ignored in preference to feature curves that are generated during the wrapping process. The only exception is when the feature curves are on a flat surface and define a surface detail of some kind (such as an imprint). Decreasing the *Angle* value increases the number of feature edges that the wrapping process generates. For this tutorial, maintain the default value of `30` degrees.

Capping the Geometry

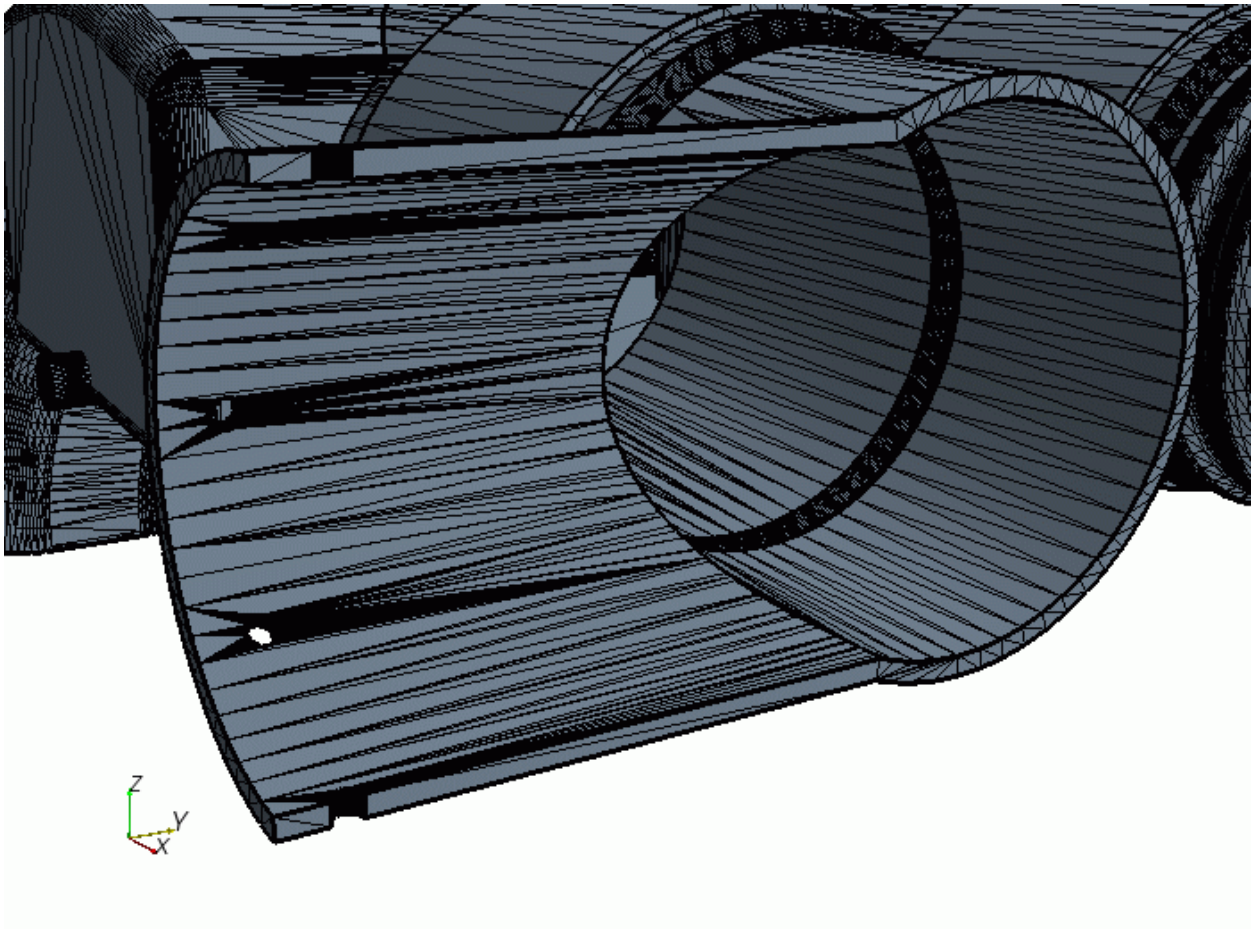
As you are extracting the internal volume of the manifold, then a mostly watertight import surface must be supplied as a starting point. Achieve this surface by capping-off the open inlet and outlet boundary areas in the manifold.

Two different tools are available to help close off holes, namely the polygonal patch fill and the hole filler. The polygonal patch filler is a quick and easy way of closing arbitrary shaped holes which do not have a closed loop definition and/or are not planar. The process literally “patches” up the surface by creating faces that cover the hole area completely. The hole filler is a more exact method of filling well-defined holes by using a closed loop feature edge definition around the hole. Both methods are used in this example.

1. To open the *Wrapper Leak Detection* dialog and associated surface repair tools:
   a) Navigate to **Geometry** > **Operations**.
   b) Right-click the **Surface Wrapper** node and select **Run Leak Detection**.
2. In the *Edit* dialog, select the tab for **Repair Surface**.

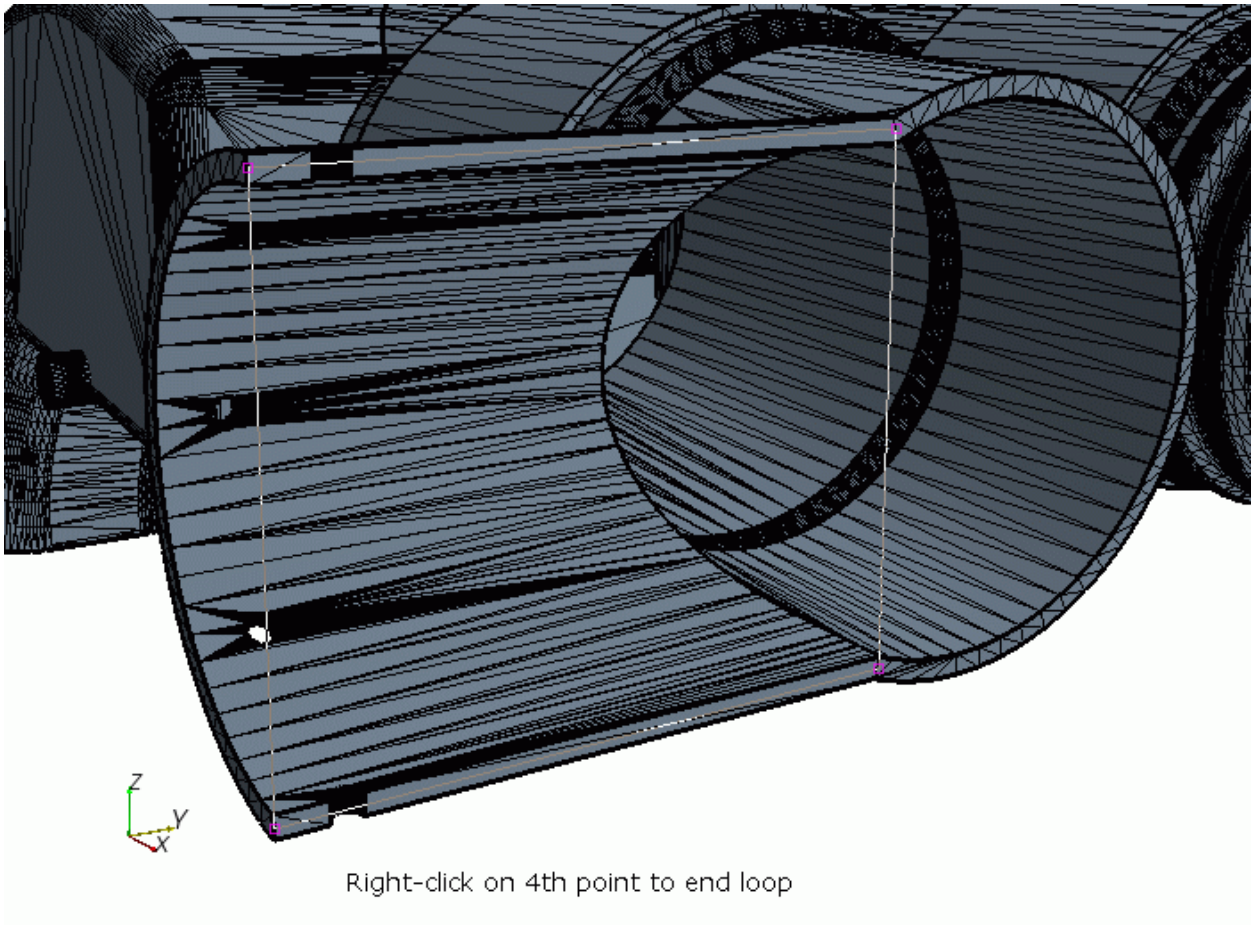


3. Change the view so that the manifold inlet pipe part is as shown below:

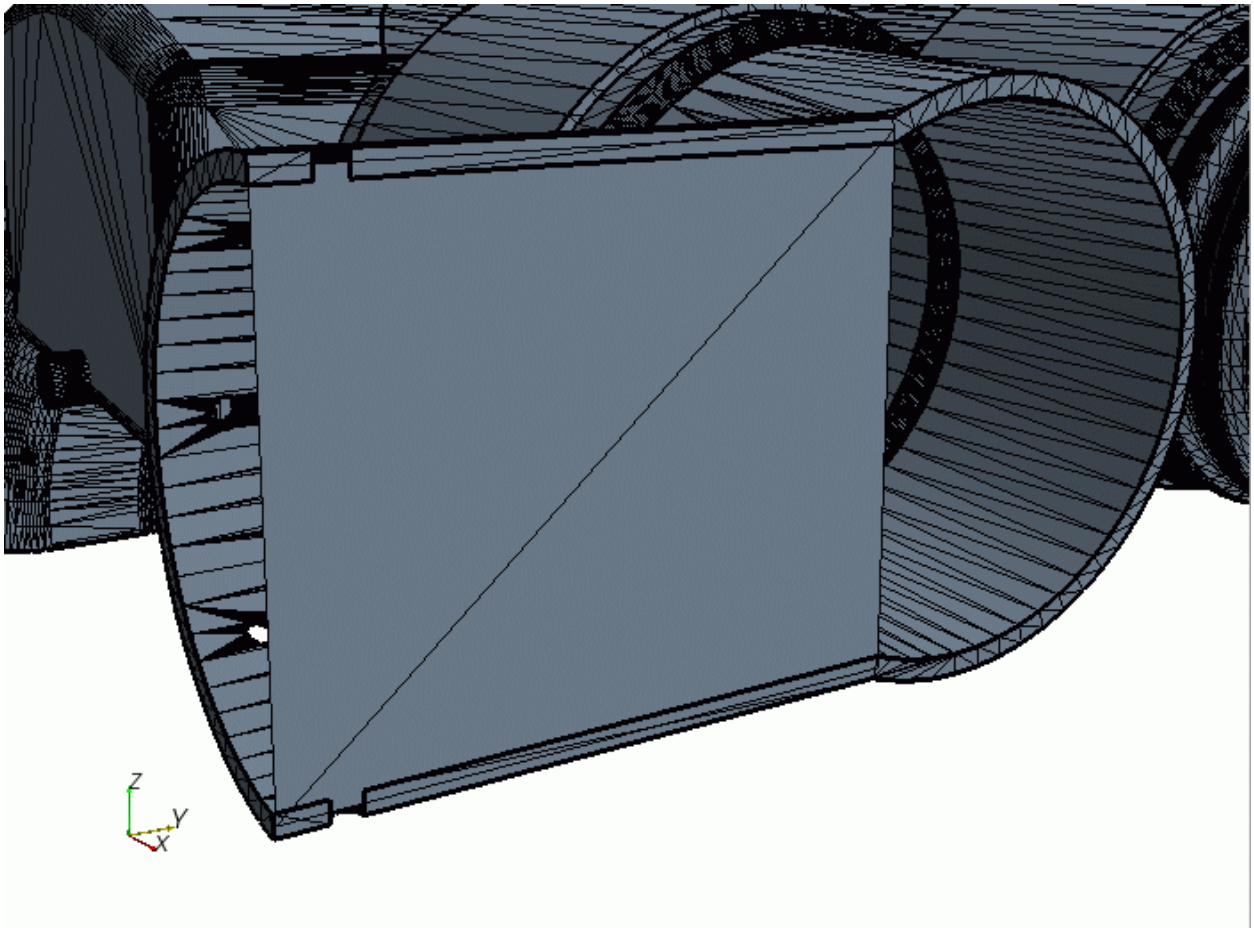


The inlet boundary is capped off in three stages. Start with the rectangular section on the mid-plane:

4. In the *Repair Surface* panel, click (**Fill Polygonal Patch**).
5. Click once on each of the four opposite corners of the mid-plane of the inlet pipe material thickness surface, as shown below. To complete the patch fill operation, right-click when selecting the fourth corner point.

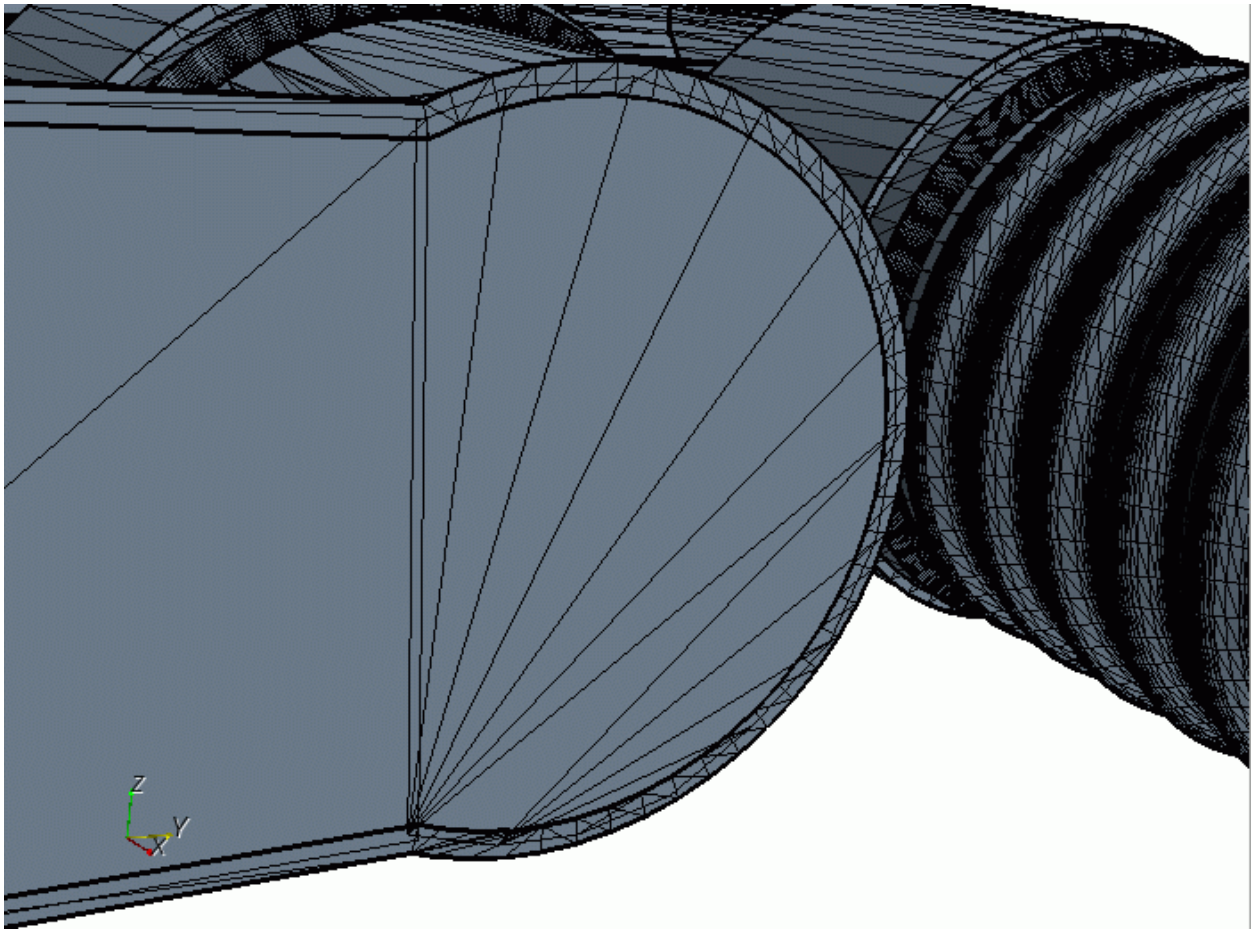


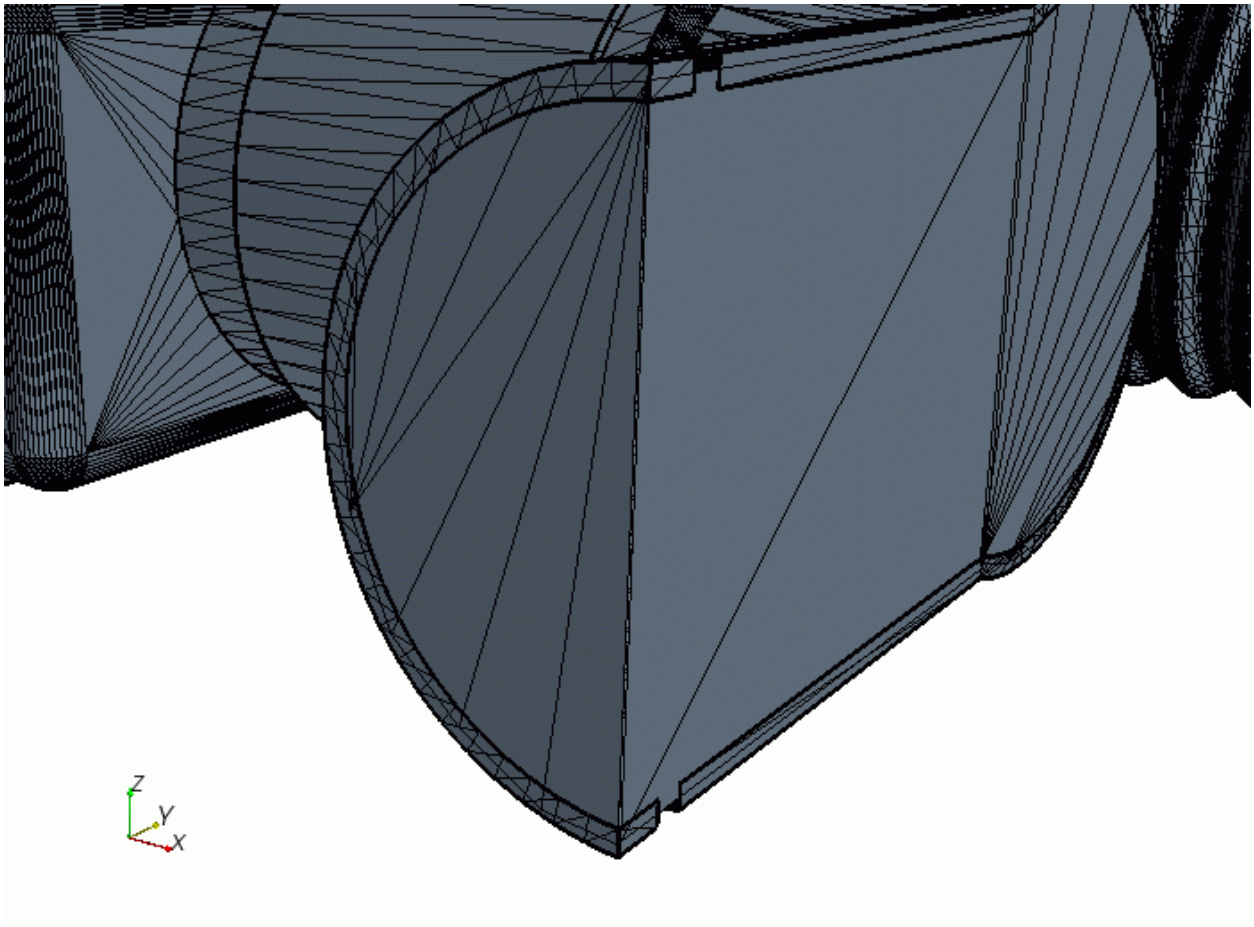
The surface patch is filled in as shown:



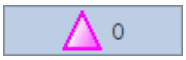
Note: If you make a mistake, use (**Undo**). Having an overlap in the patched surface is perfectly acceptable as the wrapper deals with this overlap automatically.

6. Repeat the process above for the cylindrical portion of the mid-plane and on the inlet plane half-cylinder. The patched surfaces are shown below:

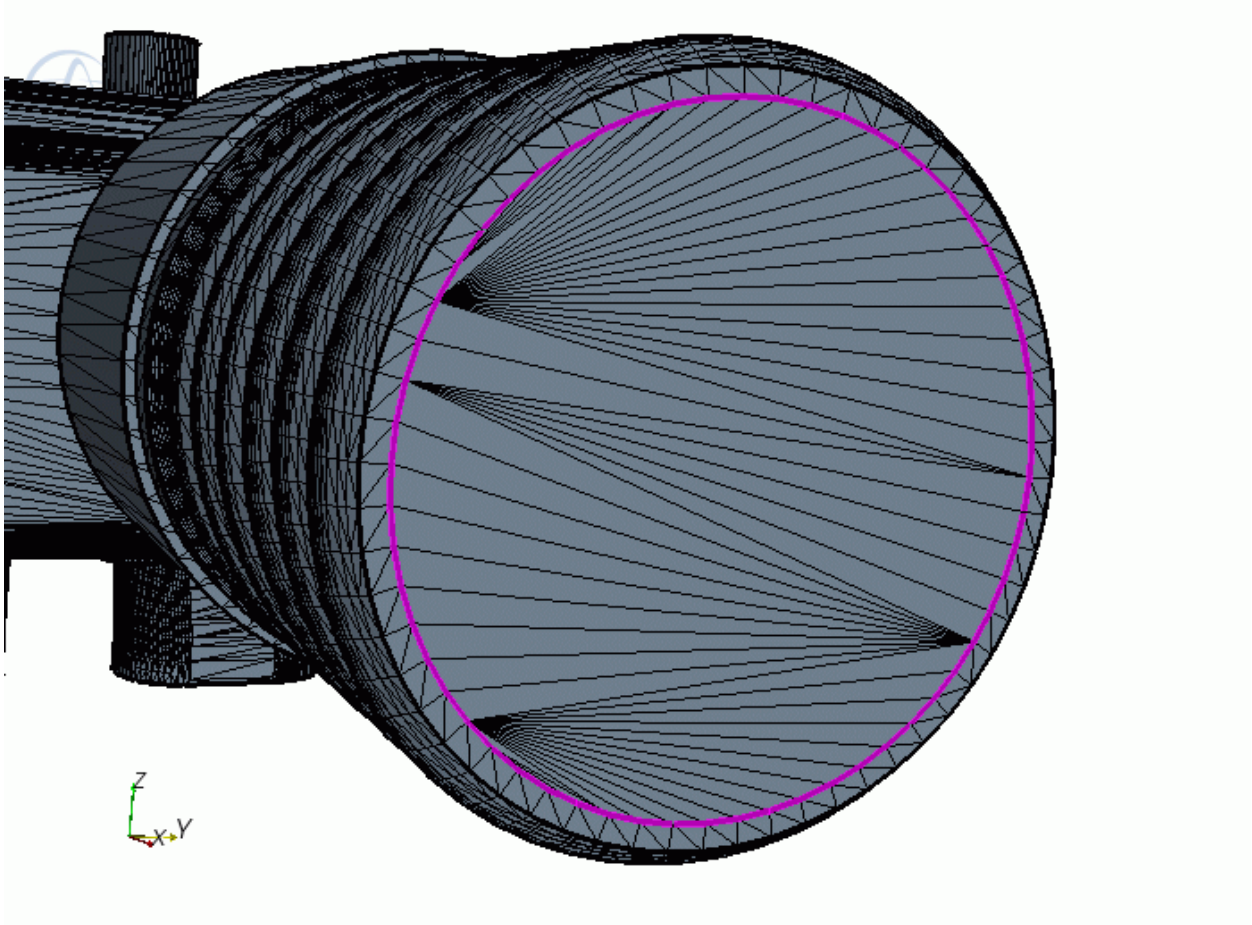

Z
Y
X



For the remaining open boundaries, use the hole filler instead of the polygonal patch filler. The default settings of the wrapper leak detection tool allow you to select only faces. To select edges or vertices, activate the corresponding options:

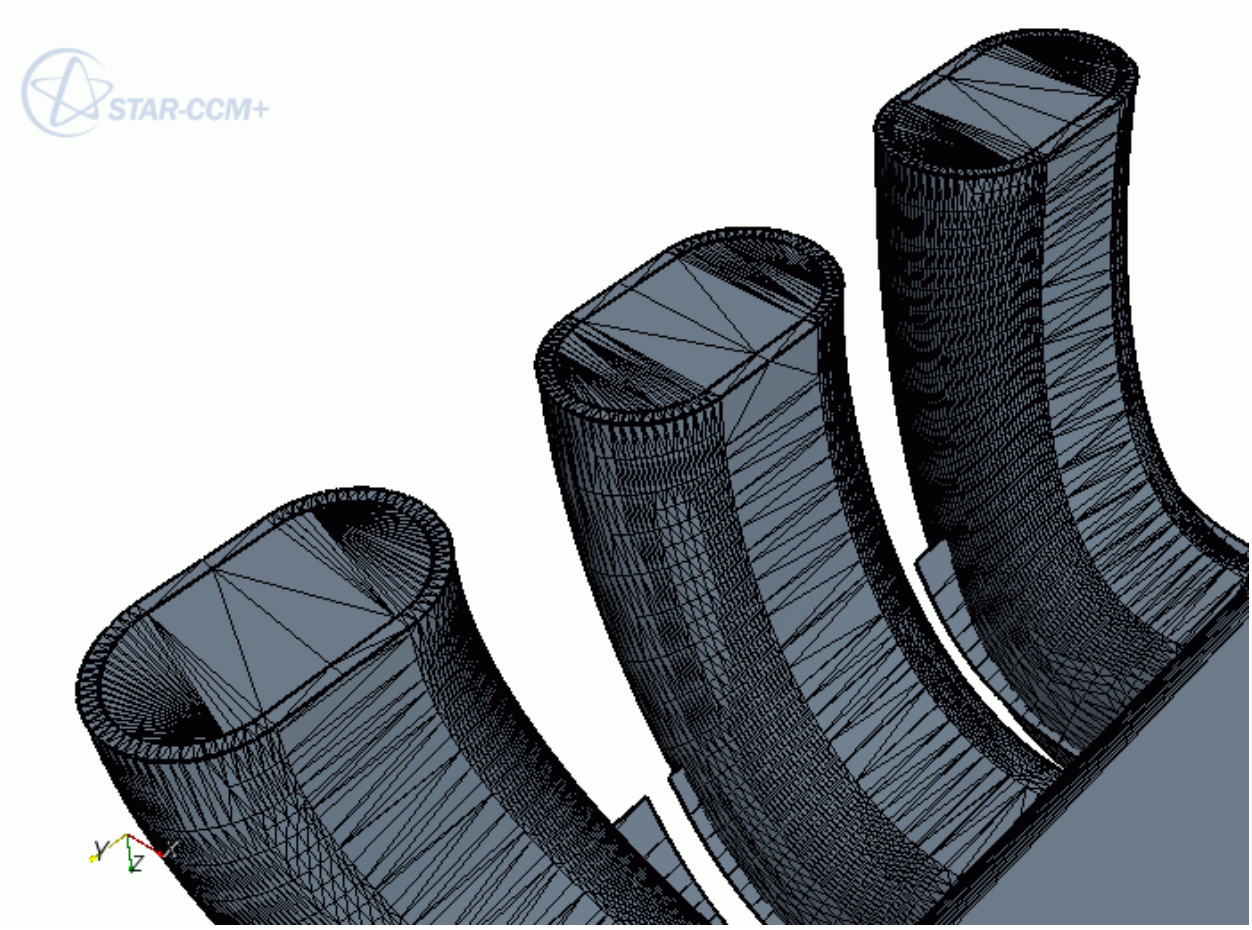
7. At the bottom of the *Graphics* window, deactivate  (**Allow faces to be selected**), and activate (**Allow edges to be selected**).
8. Rearrange the view to look at the open boundary of the cross pipe.
9. Double-click the feature edges around the hole to select them all in one go.
10. To fill the hole, in the *Repair Surface* panel, click (**Fill Holes using selected edges**).

    The result of the hole fill is shown below:



11. Repeat the above process and fill the three holes at the end of the outlet pipes of the main manifold body. To select all three feature curve loops, hold the `<Ctrl>` key during your selection.

    The result of the hole fill on the outlet pipes is shown below:



All the major holes in the geometry are now closed.

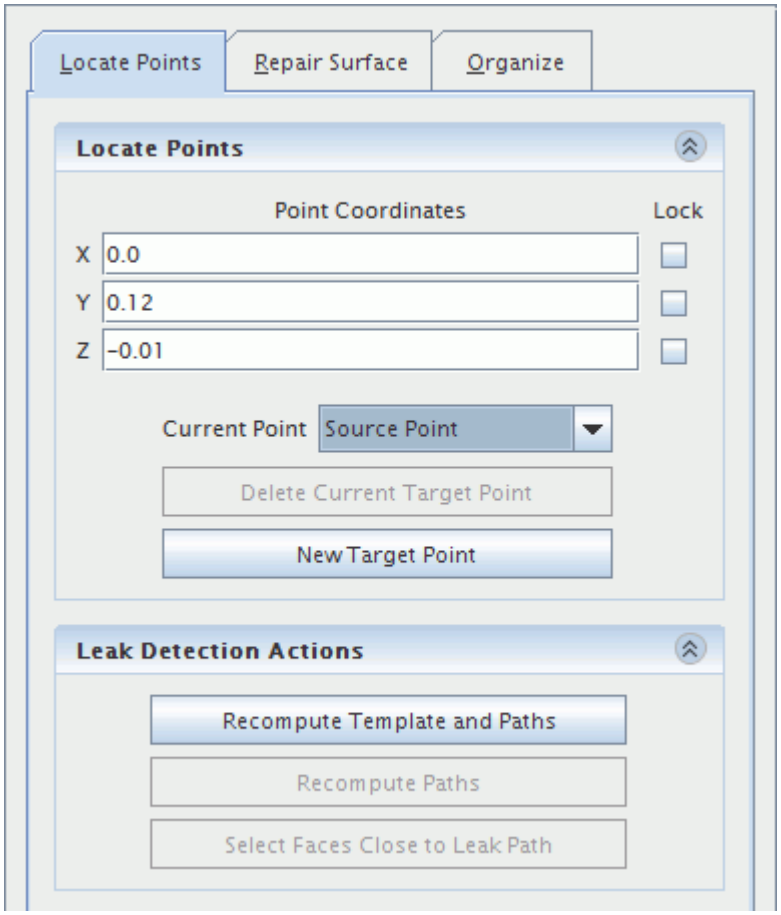
Running the Leak Detector

To highlight any remaining holes before the first wrap, run leak detection.

Leaks are not desired because they cause an inversion in the wrapping process, which results in the incorrect volumes being wrapped.

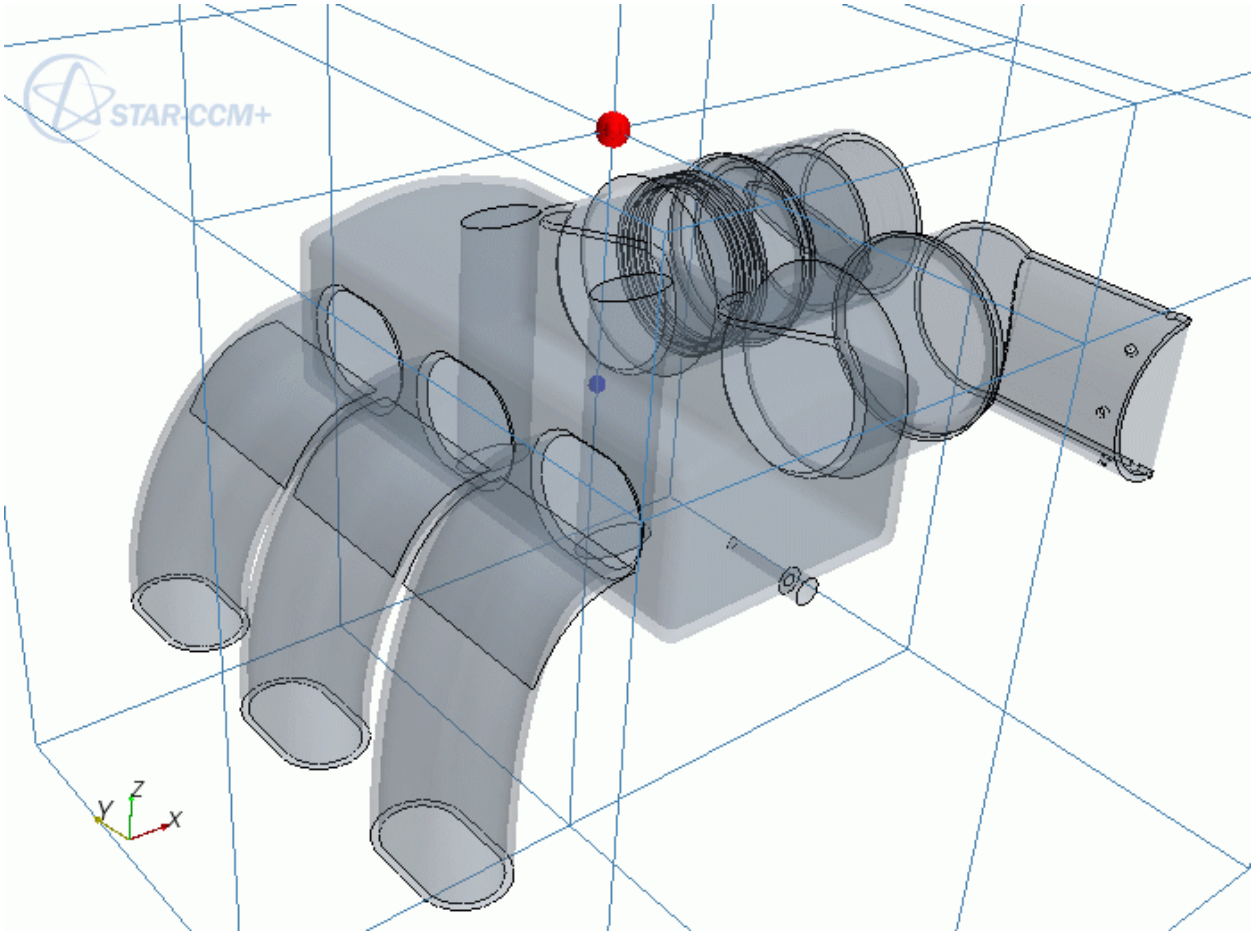
To perform a leak detection:

1. In the *Edit* dialog, select the tab, **Locate Points**.
2. In the **Locate Points** panel, set **Current Point** to **Source Point** and use the point tool in the *Graphics* window to position the source point (small blue ball) in the middle of the manifold body (approximate coordinates are X=0, Y=0.12, and Z=-0.01).



3. Set **Current Point** to **Target Point 1**.
4. Use the point tool to position the target point (small red ball) to anywhere outside the manifold geometry (for example, approximately X=0, Y=0.12, Z=0.1).

   The source and target points are positioned correctly:



5. To run the leak detector, within the *Leak Detection Actions* group box, click **Recompute Template and Paths**. In the *Recompute Template* confirmation dialog, click **Yes**.

   STAR-CCM+ takes several seconds to compute the octree mesh and detect the presence of leaks. In this case, the message `No path found between source and target points. These points will be on different sides of the final wrapped surface` is shown in the *output* window, indicating that no leak is present.

   If a hole is detected, use the polygonal patch filler to patch the leak area. To rerun the leak detector, click **Recompute Paths**.
6. At the bottom of the edit dialog, click **Close**.

Producing a Basic Wrap

Before refining the surface wrapper, review the outcome from running the surface wrapper with the initial default settings.

Run the surface wrapper based on the initial settings:

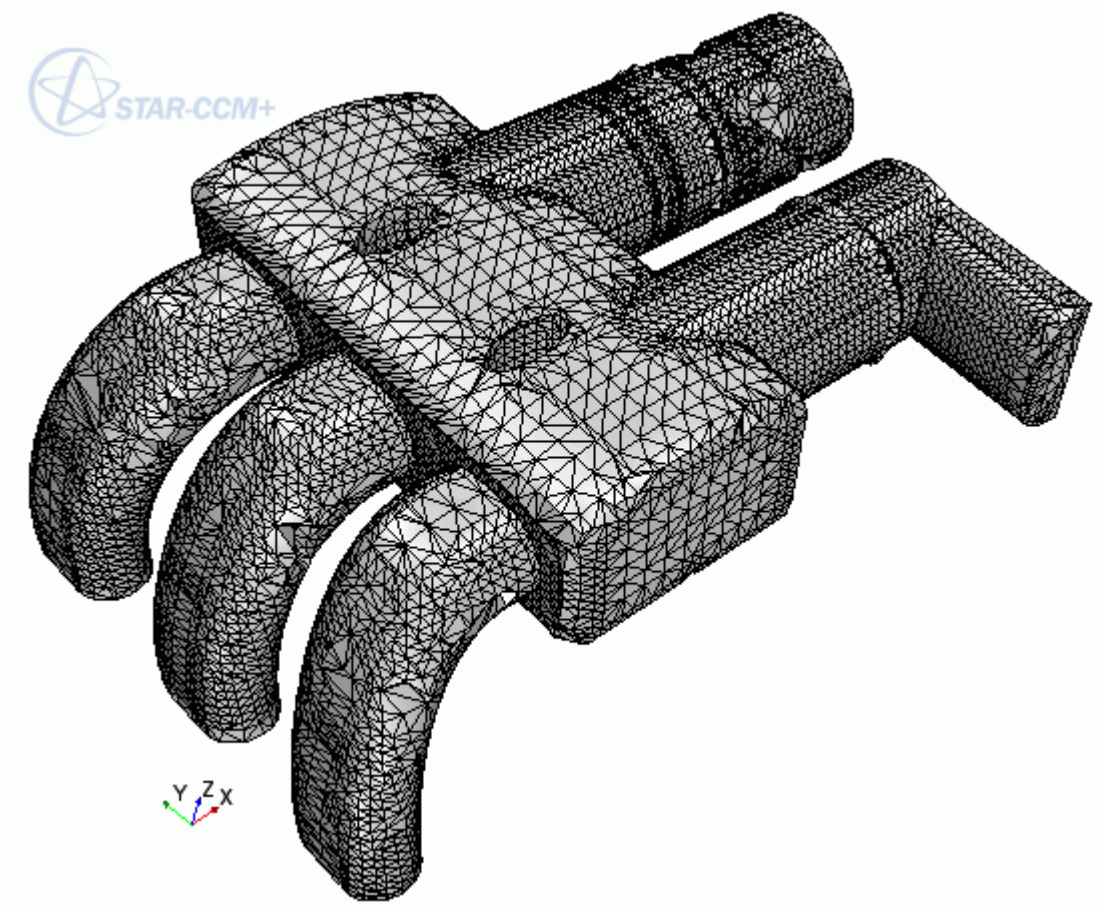
1. Right-click **Operations** > **Surface Wrapper** and select **Execute**.

   The Surface Wrapper nodes now appear in their executed state, without the exclamation mark.

The output of the surface wrapper is shown in the *Output* window. When the wrapper is complete, the message `Surface Wrapper Completed` appears.

View the result:

2. Navigate to **Geometry Scene 1** > **Displayers** > **Geometry 1** > **Parts**.
3. Set **Parts** to **Surface Wrapper**.
4. Navigate to **Outline 1** > **Parts**.
5. Set **Parts** to **Surface Wrapper**.
6. Click (**Show All Meshes**). If active, deactivate (**Make Scene Transparent**).



This initial wrapped surface is coarse and defeatured in many key areas, such as the valve and baffles.

However, the surface wrapper did achieve the following:

- Extracted the largest internal volume and eliminated the material thickness.
- Closed the small holes in the inlet pipe surface automatically.
- Defeatured the channels and flanges.
- Eliminated the overlap/double surface in the cross-pipe area.
- Intersected the struts with the main manifold body.
- Extracted the feature curves for the most important features.

Using the DeviationDistance field function, you can visualize the amount by which the surface wrap deviates from the original geometry. First, identify the original part surfaces:

7. Expand the **Tools** > **Field Functions** node.
8. Select the **DeviationDistance.Root** node and set *Parts* to **Baffles**, **Cross pipe**, **Inlet pipe**, **Manifold body**, **Sensor**, **Struts**, and **Valve**.

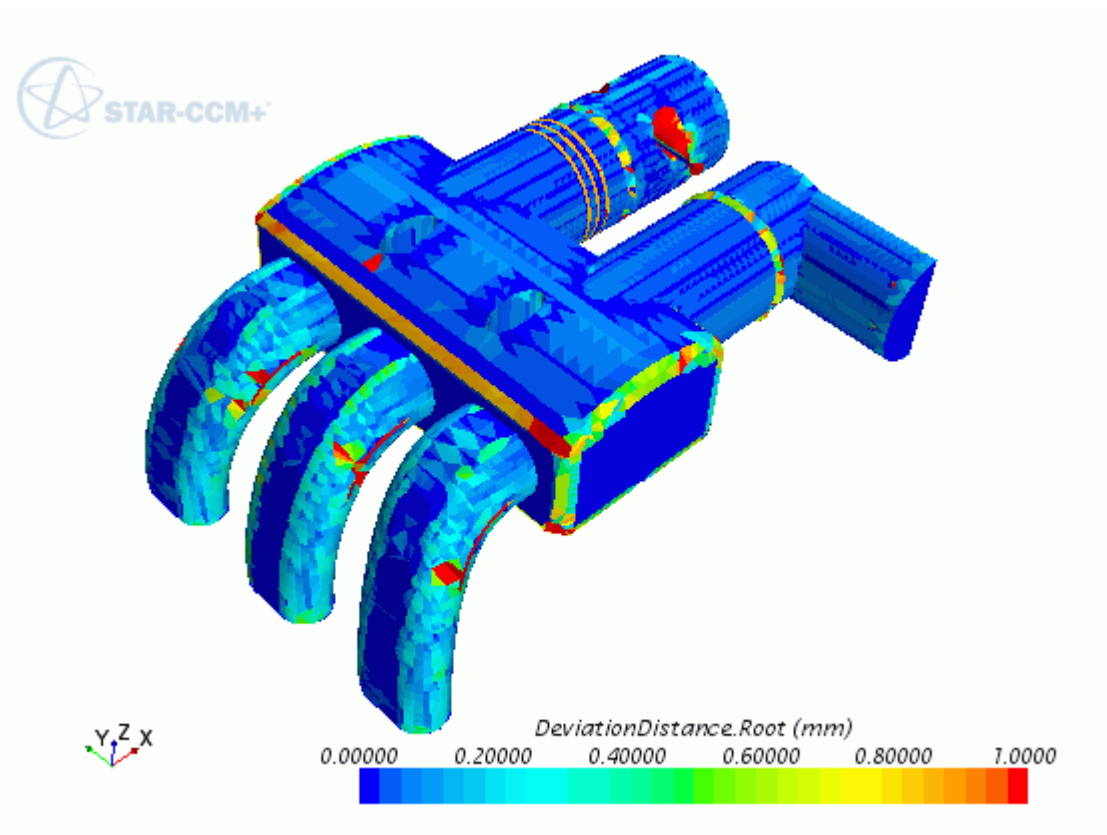
   Do not select the **Surface Wrapper** part at this point. You want to compare the surface wrap against the original surfaces.

Next create the scene in which you visualize the deviation:

9. Create a scalar scene.
10. Select the **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Parts** node and set *Parts* to **Surface Wrapper**.
11. Select the **Scalar Field** node and set the following properties:

Property	Setting
Function	**DeviationDistance.Root**
Clip	**Off**
Units	**mm**
Max	`1.0`

The scene appears as follows:



You can improve on the first wrap result by adjusting the reference settings and including additional options to deal with feature specific issues. The remainder of this tutorial concentrates on improving each geometry feature in turn, identifying the issue and the solution to rectify the problem.

Improving the Manifiold Body Fidelity and Curvature

Reducing the Manifold Body Fidelity and Curvature values has a significant and immediate impact on the general model fidelity.

1. Navigate to **Operations** > **Surface Wrapper** > **Default Controls**.
2. Set the following properties:

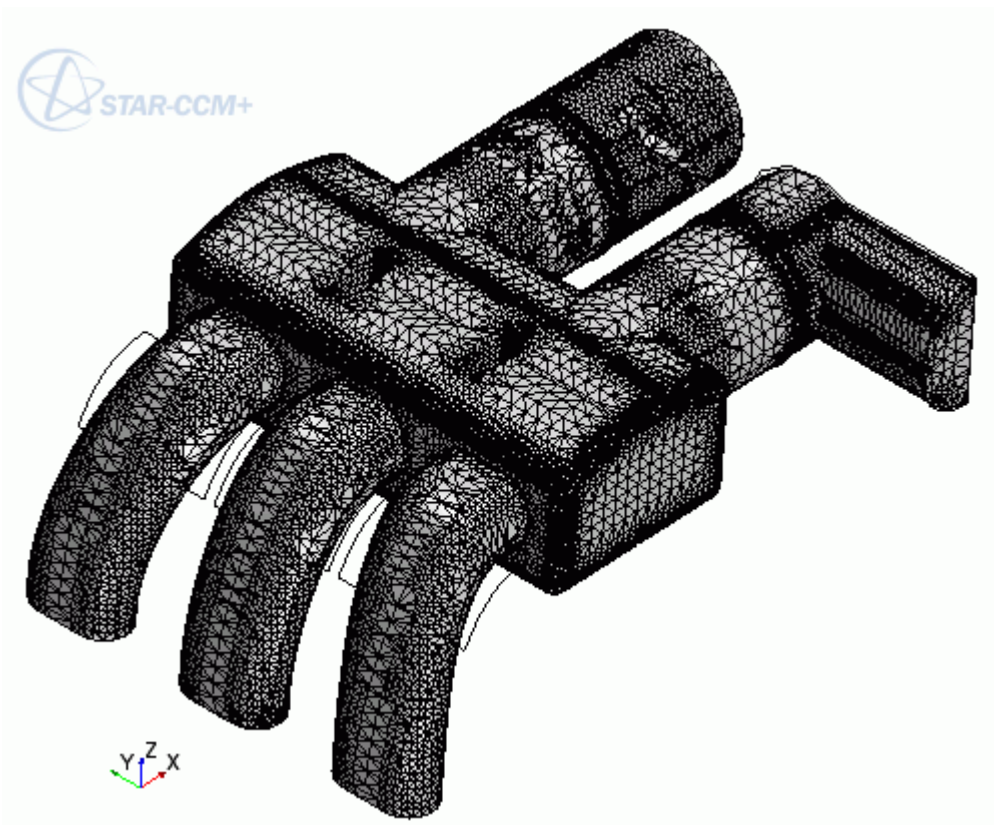
Node	Property	Setting
Target Surface Size	**Percentage of Base**	8.0
Minimum Surface Size	**Percentage of Base**	2.5

Note: When changing the target size, it is recommended that you rerun the leak detector.

To re-run the Surface Wrapper:

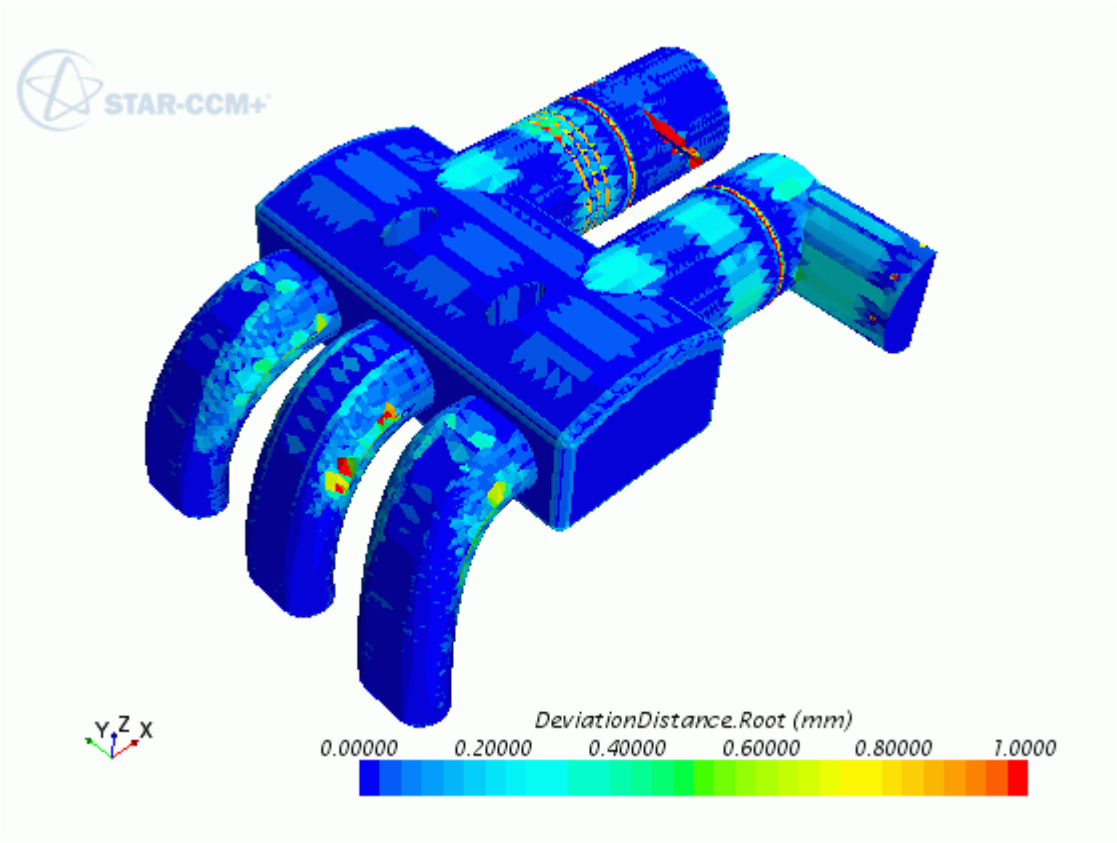
3. Switch to the Geometry Scene.
4. Right-click **Operations** > **Surface Wrapper** and select **Execute**.

   Due to the smaller surface sizes, the wrapper takes slightly longer to run this time. When the surface wrapper is complete, the display is updated with the new result:



View the deviation distance for this new surface wrap:

5. Switch to the Scalar Scene.



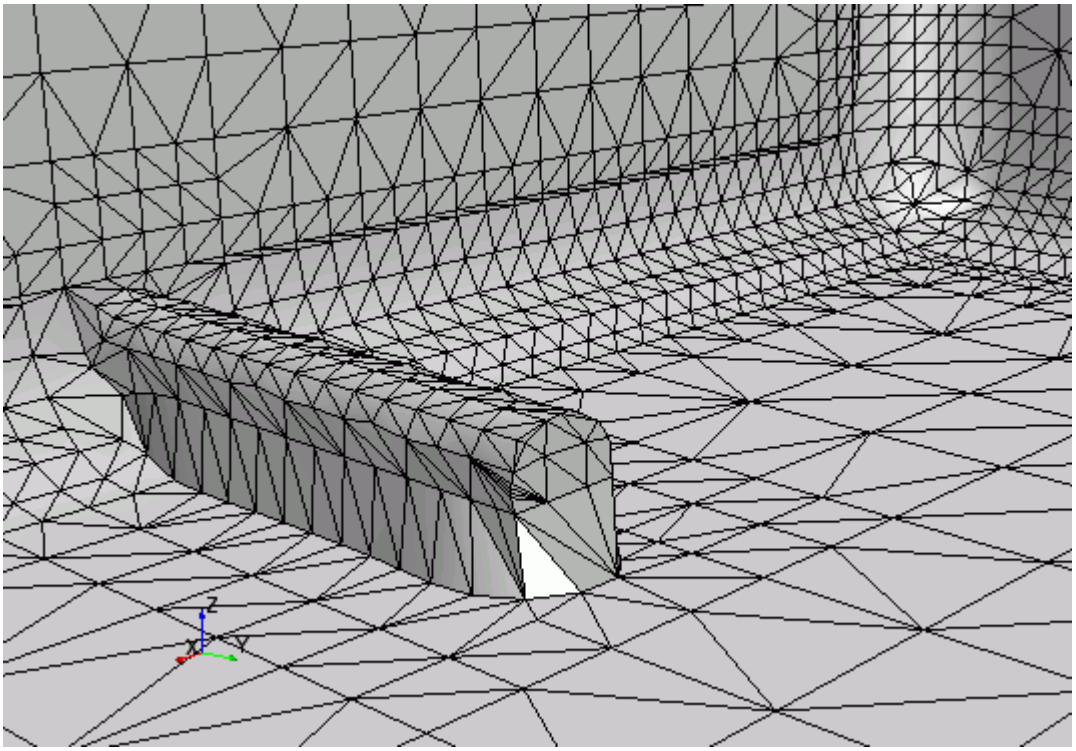
On inspecting this new wrap, you can see the following improvements in the surface definition:

- Better curvature definition for the main manifold fillets.
- Better curvature for the struts.
- Better overall representation of the three curved outlet pipes.

Improving the Sensor Part Definition

The sensor part is joined to the bottom surface of the manifold body, which is not desirable.

The wrapped surface for the sensor area currently appears as shown:



The clearance between the sensor and the base is approximately 2 mm, which is less than the current target surface size of 8 mm.

To prevent the wrapper from joining the two boundaries, use a contact prevention.

1. Navigate to **Geometry** > **Operations** > **Surface Wrapper**.
2. Right-click **Contact Prevention** and select **New** > **One Group Contact Prevention Set**.

   A **Contact Prevention** > **Contact Prevention** node is created.
3. Select the **Contact Prevention** > **Contact Prevention** node and set the following properties:

Property	Setting
Surfaces	**Manifold Body** and **Sensor**
Minimum Size	`0.5 mm` (note the units)

   The **Minimum Size** value indicates the triangle size in the space between the two parts, which determines the octree refinement in the area. The value is smaller than the gap size (typically one half or one quarter of the distance). In this example, four octree triangles are generated in the 2 mm gap between the sensor and the manifold base.

   You can still join the surfaces even though a **Minimum Size** value is set. The **Minimum Size** value is simply a stopping size for the surface wrapper refinement, which results in the two surfaces being separated. The value also has to be greater than zero.
4. Rename **Contact Prevention** to `Sensor CP`.

   Since the sensor is also fairly small in comparison to the manifold, increase the triangulation in this area by specifying a localized target and minimum surface size for the sensor boundary.
5. Right-click **Surface Wrapper** > **Custom Controls** and select **New** > **Surface Control**.

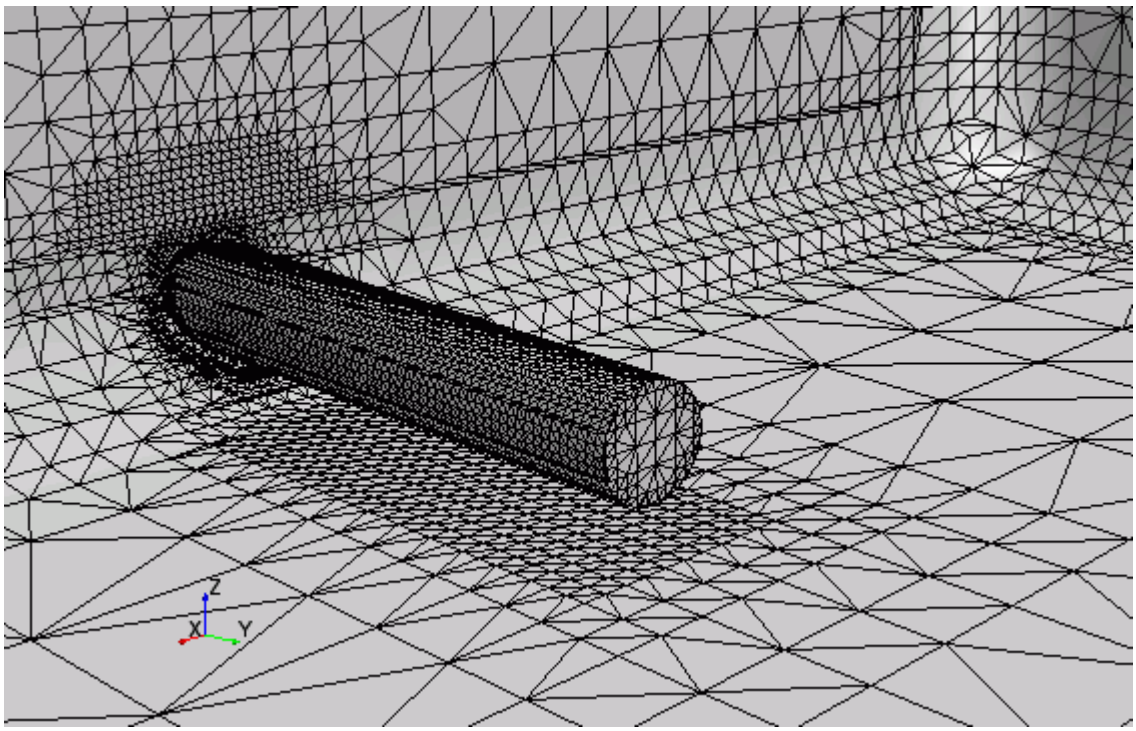
6. Edit the **Custom Controls** > **Surface Control** node and set the following properties:

Node	Property	Setting
Surface Control	**Part Surfaces**	**Sensor**
⌊ **Controls** > **Minimum Surface Size**	**Minimum Surface Size**	**Custom**
⌊ **Controls** > **Target Surface Size**	**Target Surface Size**	**Custom**
⌊ **Values** > **Minimum Surface Size**	**Percentage of Base**	`0.5`
⌊ **Values** > **Target Surface Size**	**Percentage of Base**	`1.0`

Rerun the Surface Wrapper:

7. Right-click **Operations** > **Surface Wrapper** and select **Execute**.

   When the wrapping process is complete, STAR-CCM+ updates the display with the new result:

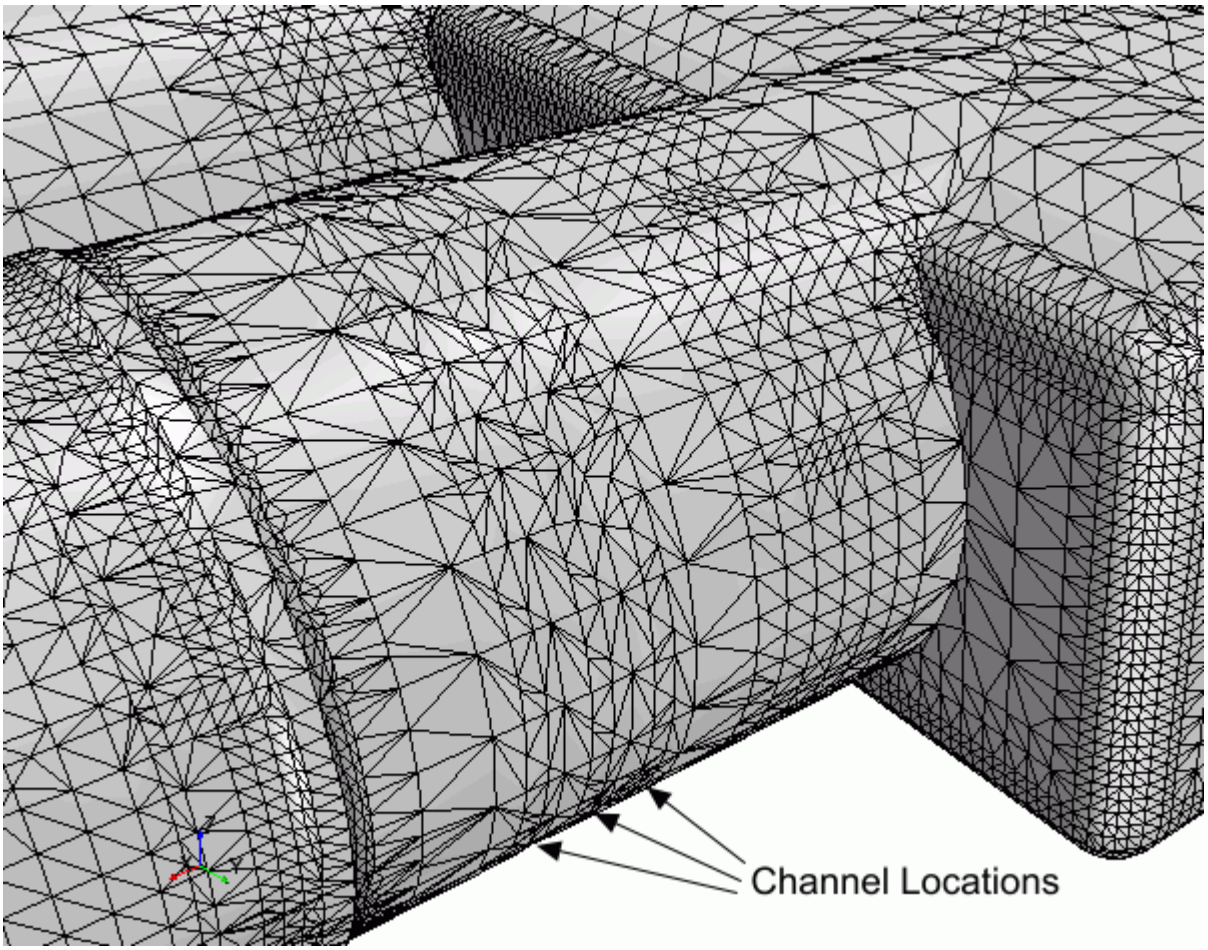


The sensor part is no longer joined to the manifold body. The surface resolution has also improved such that the sensor is well-defined and intersects the manifold side wall crisply.

Improving the Channels Definition

In addition to refining the surfaces of parts in the wrap, you can perform partial wrapping, where you exclude selected parts or part surfaces from the wrap operation. When you exclude parts from the wrap, the original tessellation of the excluded parts is passed through to the final wrap surface.

Based on the current reference target size of 8 mm, the three channels that exist in the manifold cross pipe have been defeatured. The location of the channels in the current wrap is shown below:



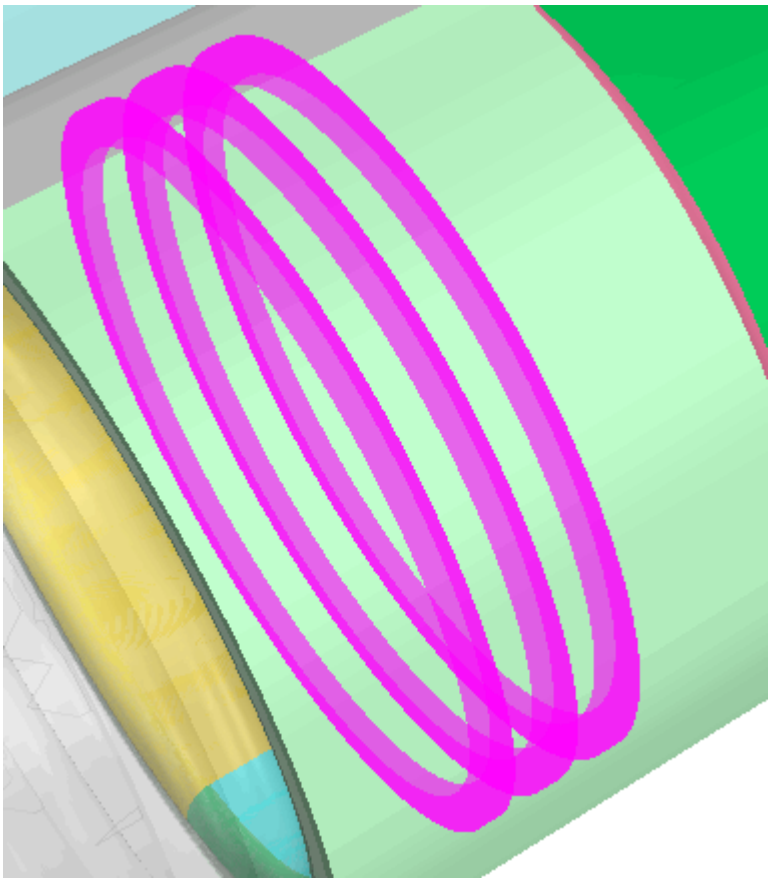
Depending on the requirements of the flow analysis, defeaturing the channels can be a desirable result, as it simplifies the geometry without adversely affecting the flow solution. The defeaturing occurs as the channels themselves are approximately only 2.5 mm wide. The octree triangles in this area are based on the target size, which means the surface wrapper cannot capture the detail.

To maintain this feature, extract the patches that make up the feature and choose to keep the input surfaces.

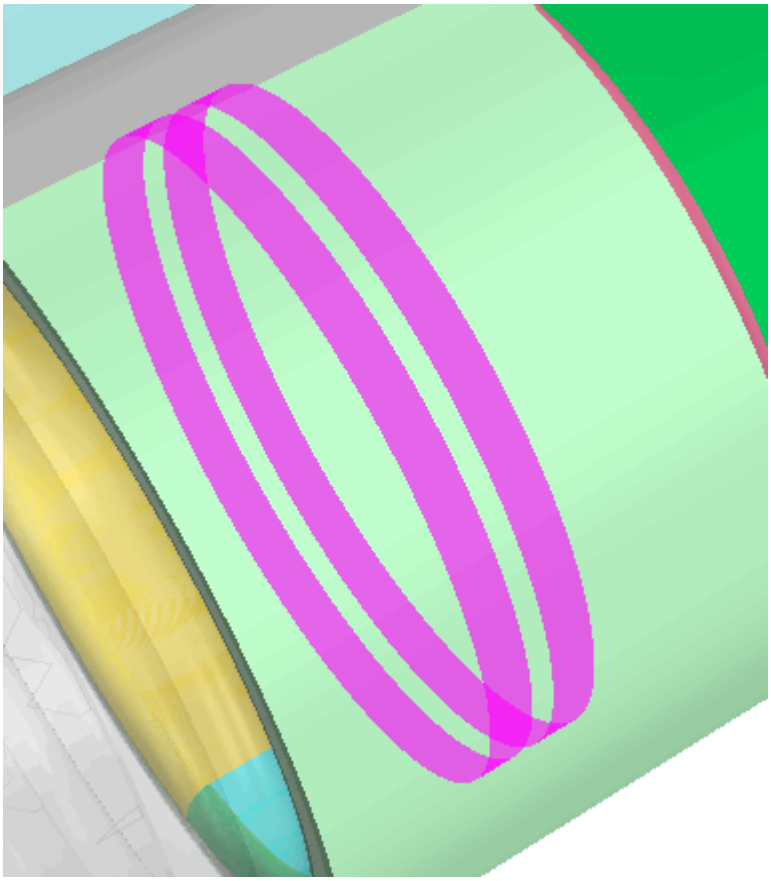
1. Split the surface of the manifold body that contains the channels:
   a) Expand the **Geometry** > **Parts** > **Manifold Body** node.
   b) Right-click **Surfaces** > **Faces** and select **Split by Patch**.
   c) In the *Split Part Surface by Patch* in-place dialog, select the patch entry **[92, 227]**.

d) In the *Patch Selection Control* group box, click **Multi-grow selected patches**. In the *Patch multiple grow options* dialog, set *Number of Steps* to **Finite** with a value of **5**. Click **OK**.

The patches that correspond to the channels and the patches that connect the channels are selected. To make it easier to visualize, the channel patches are shown below.



The patches that connect the channels are shown below.



e) Set **Part Surface Name** to `Channels`.

f) Click **Create** then **Close**.

The new surface appears in the simulation tree.

2. Preserve the input surfaces for the **Channels** part surface:

   a) Select the **Operations** > **Surface Wrapper** node and activate **Perform Partial Wrapping**.

   b) Expand the **Operations** > **Surface Wrapper** > **Default Controls** node.

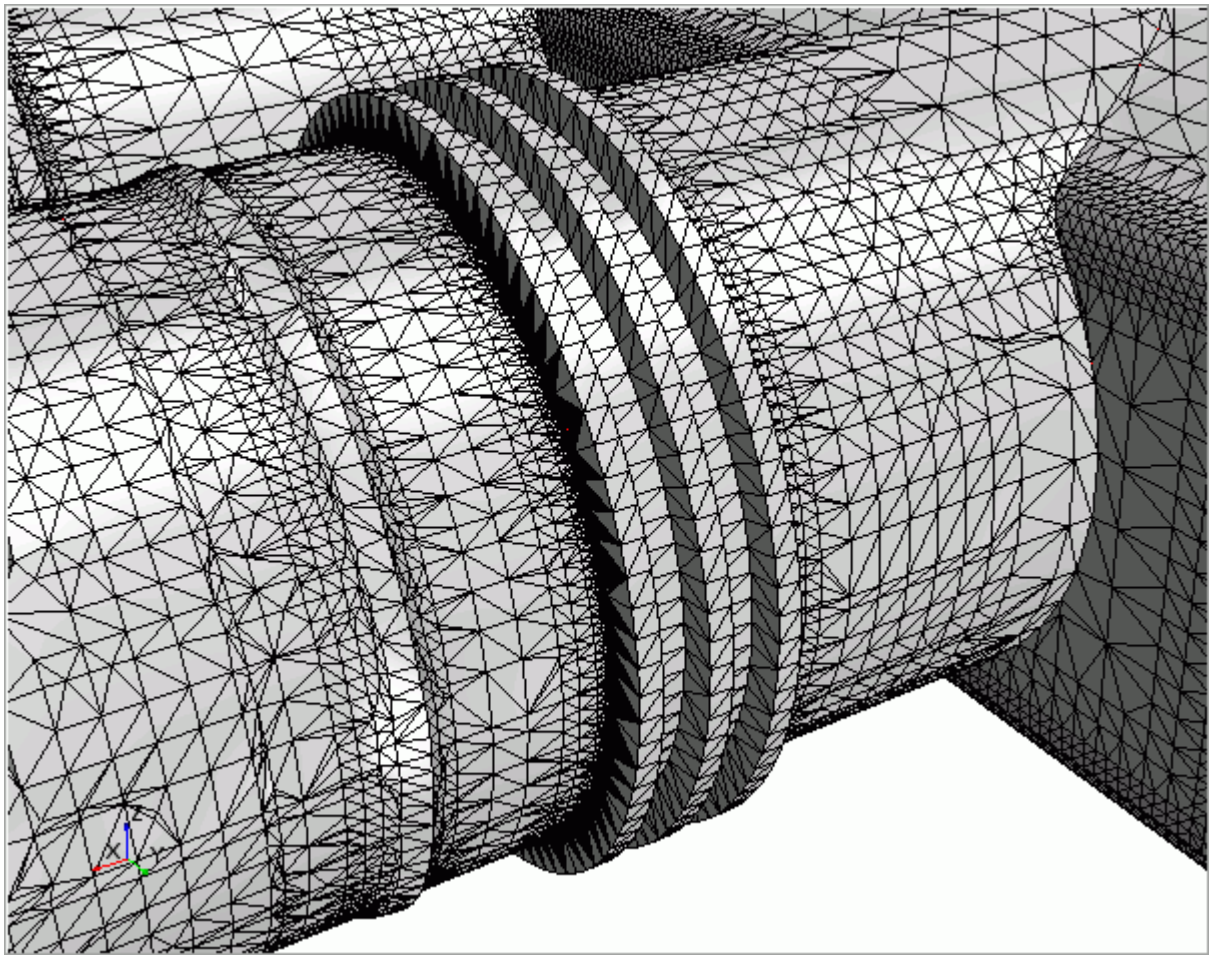
   c) Select the **Preserved Input Surfaces** node and set **Excluded Surfaces** to **Manifold Body** > **Surfaces** > **Channels**.

Rerun the Surface Wrapper:

3. Right-click **Operations** > **Surface Wrapper** and select **Execute**.

When the process is complete, display the channel part surfaces:

4. Expand the **Displayers** > **Geometry 1** node.
5. Select the **Parts** node, maintain the current selection of parts, and also select the **Surface Wrapper** > **Manifold Body** > **Channels** part surface.



The channels are now included using the original tessellation.

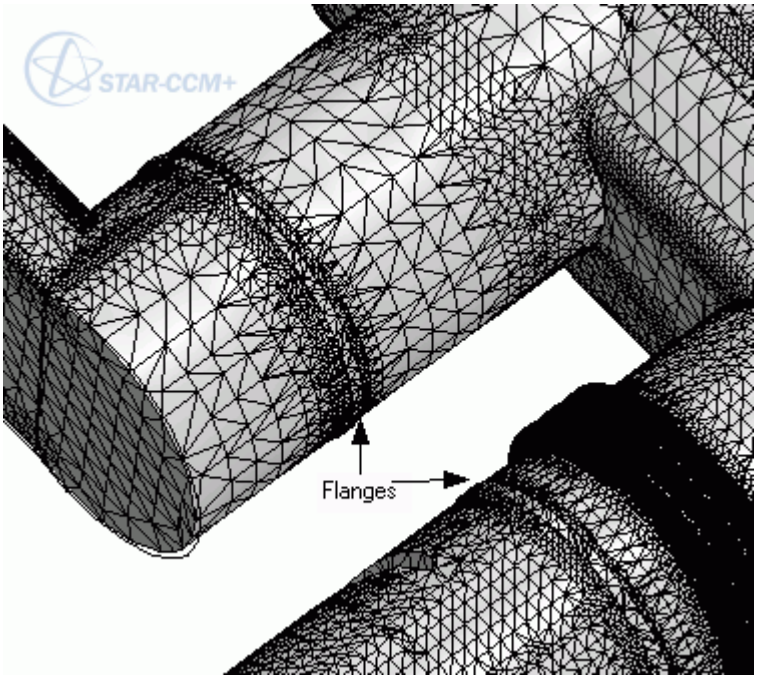
Improving the Flange Definition

Since the surface wrapper has defeatured the original detail of the flange parts, a contact prevention is used to improve the output of the surface wrapper around the flanges.

There are two sets of flanges in the manifold assembly:

- between the inlet pipe and manifold body
- between the cross-pipe and the manifold body

These flanges represent high curvature areas that are close to one another but are not joined. Each wrapping iteration results in defeaturing of the flange areas, as shown below:



The flange features must be eliminated from the wrapper result by isolating the original boundaries that correspond to the flanges. Local surface target and minimum sizes -- which are bigger than the existing global reference sizes, must be set for the flanges . This effectively flattens out the area between the manifold and corresponding pipes.

However, to include the flanges in the volume mesh, a contact prevention option must be used. Since there are three different boundaries in this case (the manifold body, inlet pipe, and cross-pipe) then the two group contact prevention set option can be used to minimize the setup effort involved.

1. Right-click **Operations** > **Surface Wrapper** > **Contact Prevention** and select **New** > **Two Group Contact Prevention Set**.

   A new node, **Contact Prevention** appears under the **Contact Prevention** node.

2. Select the **Contact Prevention** > **Contact Prevention** node and set the following properties:

Property	Setting
Surface set 1	**Manifold Body**
Surface set 2	**Cross pipe** and **Inlet pipe**
Minimum Size	`2.0E-4 m (0.2 mm)`

   In this case, a contact prevention set is formed between the parts in the Surface set 1 group and the parts in the Surface set 2 group. Contact prevention is not enforced between the inlet pipe and cross pipe parts.

   The *Minimum Size* property determines the cut-off thickness where the two curved surfaces in the flange come together. A reasonably small value is used to capture the high curvature and produce a uniform join in the circumferential direction.

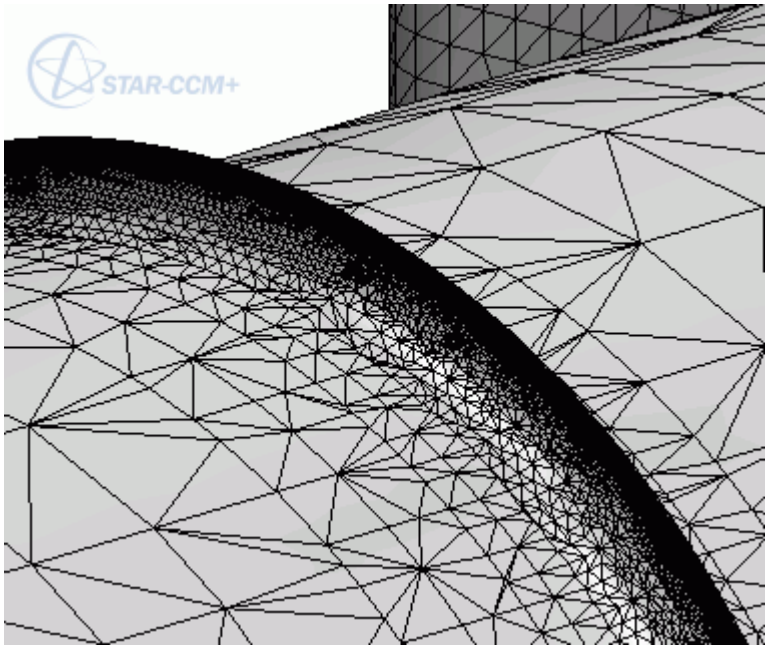
   This means that the smallest triangle size used at the interface where the two flange surfaces come together is 0.2 mm. A larger value could be used but it would result in a lower definition.

3. Rename the **Contact Prevention** node to `Flange CP`.

Rerun the Surface Wrapper with the new settings:

4. Right-click **Operations** > **Surface Wrapper** and select **Execute**.

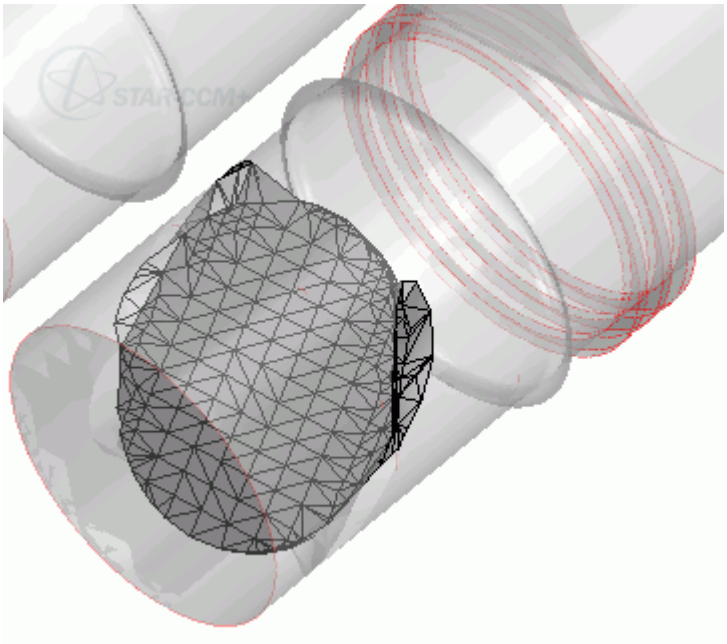
   When the process is complete, STAR-CCM+ updates the display with the new result.



The flanges are now well-defined and have a sharp edge in the circumferential direction.

Improving the Valve Definition

Since the surface wrapper has defeatured the gap between the valve and the cross pipe surface, another contact prevention is used to improve the output of the surface wrapper.



Since the valve and the cross pipe surface are in close proximity, the surface wrapper defeatures the gap between the two parts. A contact prevention is used to improve the resulting poor definition in those areas. The minimum size is calculated based on the gap between the two parts, which is approximately 1 mm. No additional refinement of the valve part is required in this case since it is planar.

1. Right-click **Operations** > **Surface Wrapper** > **Contact Prevention** and select **New** > **Two Group Contact Prevention Set**.
2. Select the **Contact Prevention** node and set the following properties:

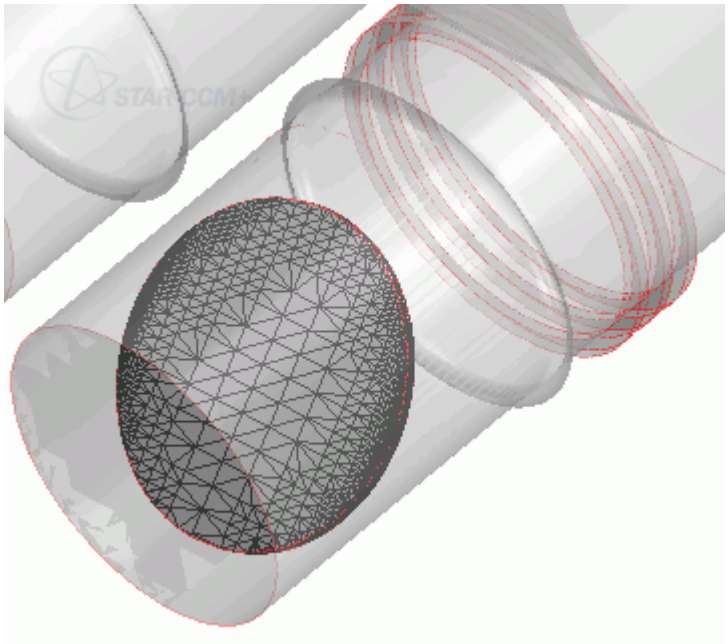
Property	Setting
Minimum Size	`0.5 mm`
Surface set 1	**Cross Pipe**
Surface set 2	**Valve**

   The *Minimum Size* value determines the number of triangles that are placed in the gap. A value of half the gap size produces a good result and does not produce too much surface refinement.
3. Rename the **Contact Prevention** node to `Valve CP`.

Rerun the Surface Wrapper with the new settings:

4. Right-click **Operations** > **Surface Wrapper** and select **Execute**.

   When the process is complete, STAR-CCM+ updates the display with the new result:

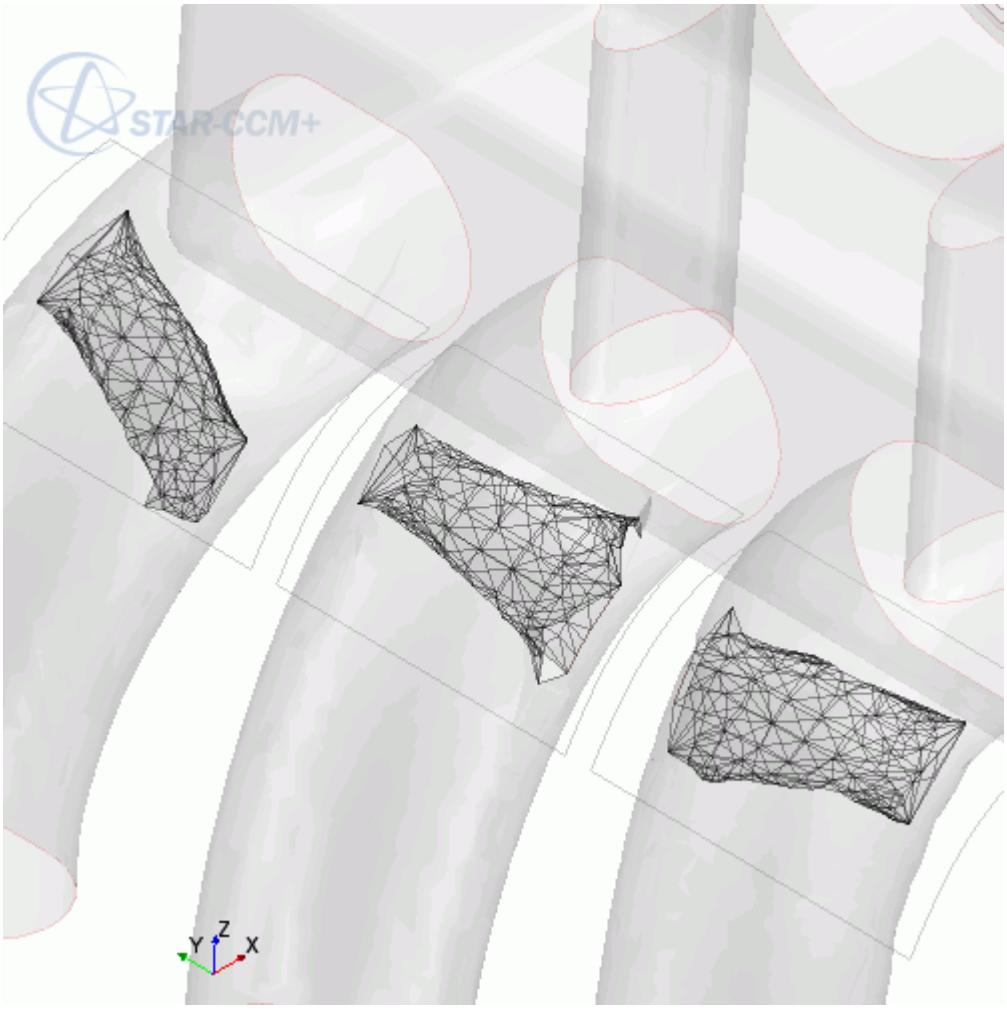


The valve is now well-defined and does not intersect the side wall of the cross-pipe.

Improving the Baffle Definition

The baffles within the geometry are initially represented as intersecting surfaces which pass through the outlet pipes that emerge from the manifold body.

The surface wrapper can detect these surfaces and automatically “inflate” them, so that a small thickness is introduced into the shape of the baffle. So far, the wrapper has not done a good job at capturing either the surface, or the intersection of the baffles with the pipes.



Three main options exist for improving the quality of the resultant baffle definition:

- Creating a true baffle of zero thickness
- Using contact prevention and local boundary sizes
- Using the offset faces/edges tool to inflate the baffle

By default, the surface wrapper is able to inflate surfaces so that they have a thickness. However, the inflation thickness is dependent on the triangulation of the surface. This method is useful if you simply want to provide an internal wall to impede flow. Alternatively, inflate the baffle using the surface preparation offset faces/edges tool and specify the exact inflation thickness and direction before the wrapper is used. The surface wrapper then treats the inflated surfaces the same as any other closed body and incorporates them into resultant surface.

If you want to generate a mesh with zero-thickness or “true” baffles, you can use the surface wrapper to stitch the baffle surfaces onto the outlet walls. To achieve this, some surface preparation for the baffles is required to connect the baffles to the outlet walls, and they must be defined as interfaces within the model. When the surface wrapper is run, the mesher identifies the baffles and they are stitched into the final mesh with zero thickness.

Note: To try each of the above methods for comparison, first save the simulation file to be able to revert to the model before the baffles were altered.

Contents:

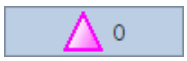
Creating a True Baffle
Using the Built-in Wrapper Baffle Inflation with Contact Prevention
Using the Offset Faces/Edges Tool to Inflate the Baffles Manually

Creating a True Baffle

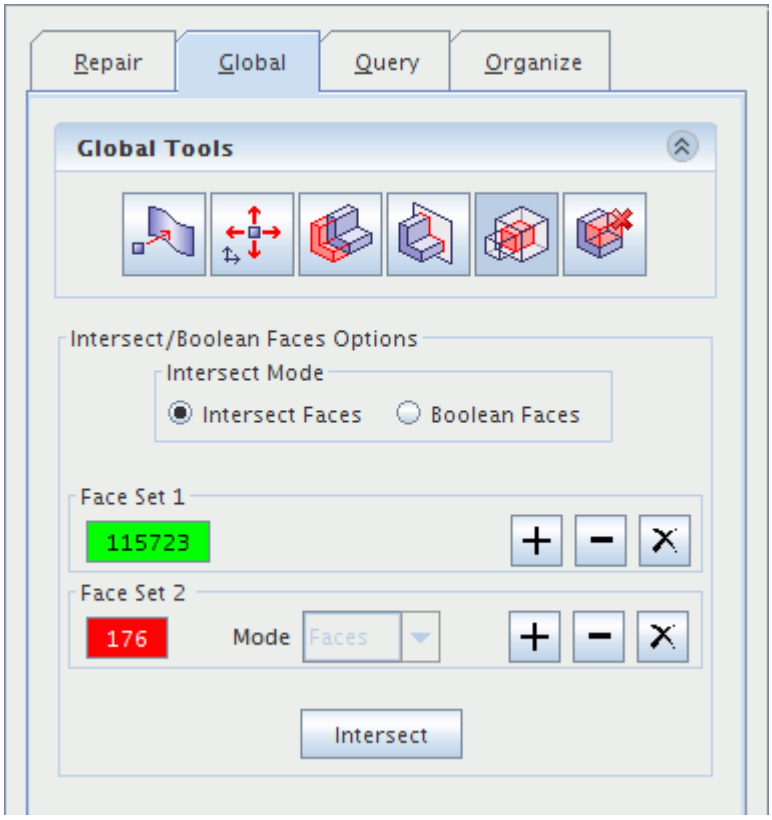
For cases where a zero-thickness baffle is required, and the original imported surface has the necessary faces, the surface wrapper can stitch in the baffle faces after the main wrap.

In this process, you connect the baffle to the repaired input surface. When the wrapped surface is fed into the surface remesher, the result is a high-quality, zero-thickness, interior wall created for the baffles.

1. For *Geometry Scene 1*, select **Displayers** > **Geometry 1** > **Parts** and set *Parts* to **Baffles**, **Cross pipe**, **Inlet pipe**, **Manifold body**, **Sensor**, **Struts**, and **Valve**.
2. Enter surface repair mode with the baffle and input surface parts:
   a) Within the main simulation tree, expand **Geometry** > **Parts**, and multi-select **Baffles** and **Manifold Body**.
   b) Right-click one of the highlighted nodes and select **Repair Surface**.
   c) In the *Surface Repair Options* dialog that appears, click **OK**.
   d) Within the *Edit* dialog, select the tab, *Global*.
   e) Within *Global Tools*, click (**Intersect or Boolean Selected Faces**).

   The **Intersect/Boolean faces** tool requires you to select two sets of faces, *Face Set 1* and *Face Set 2*.
3. Select the faces for *Face Set 1*:
   a) At the bottom of the *Graphics* window, deactivate (**Allow edges to be selected**) and activate  (**Allow faces to be selected**).
   b) Click (**Select faces/edges on part surfaces/part curves**) to select faces on parts.
   c) In the *Select Part Surfaces/Curves* dialog, select the **Manifold Body** part.
   d) Click **Apply** then **Close**.

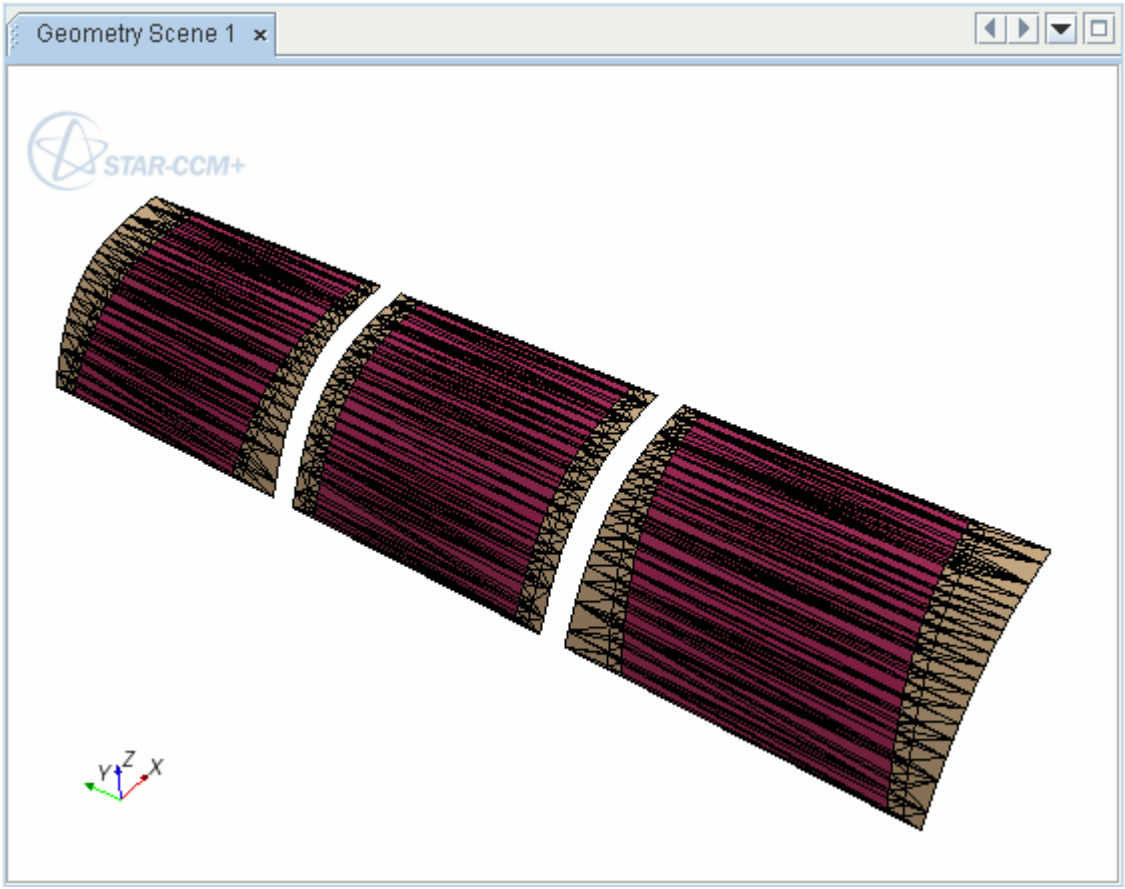
   The Manifold Body part surfaces appear selected.
4. Add the selected faces to *Face Set 1*:
   a) In the *Face Set 1* box within the **Intersect/Boolean Faces Options** group box, click (**Add selected faces to the set**).
   b) In the *Graphics* window, click (**Clear selected faces**) to clear the existing selection.
5. Repeat the above procedure to add all Baffles part surfaces to Face Set 2.
6. In the **Intersect/Boolean Faces Options** box, click **Intersect**.



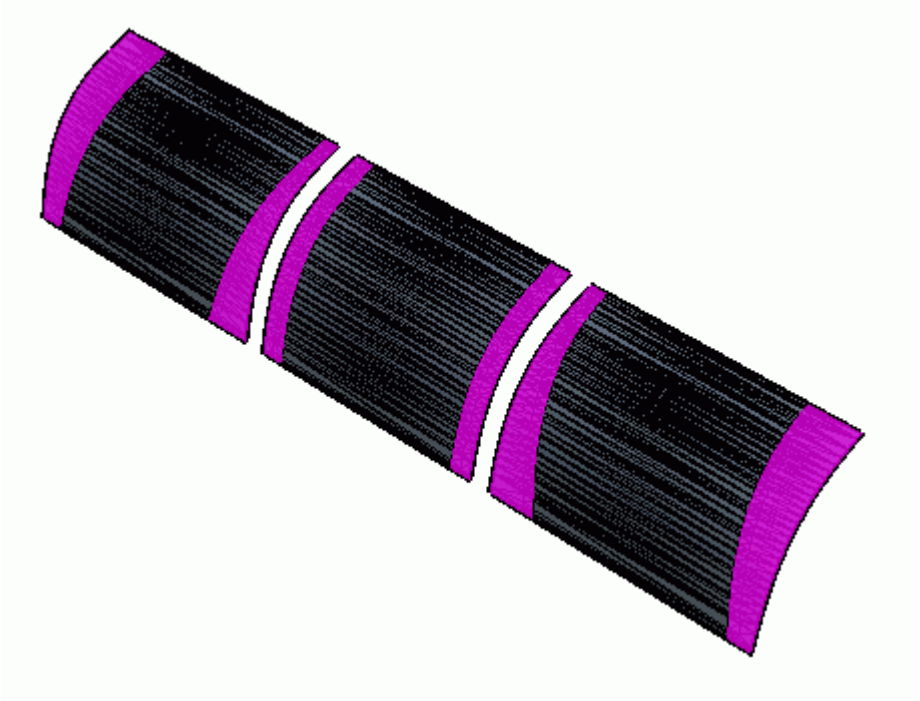
The display in the *Graphics* window is updated to show the lines of intersection between the two sets of faces. A clearer view can be obtained by showing only the Baffle surfaces in the scene.

7. In the *Graphics* window, right-click on the surface of the Manifold Body and select **Manifold body** > **Hide**

If there are baffle surfaces protruding beyond the wrapped surface, the volume mesher does not operate properly. To avoid this, the protruding portions of the baffle surfaces must be removed. These portions are shown below in beige:



8. Delete the protruding portions of the baffle surfaces:
   a) In the *Graphics* window, double-click one of the faces within a protruding area.
      This action selects the whole portion within any bounding edges.
   b) Hold down `<Ctrl>` and continue to double-click faces within the other portions shown previously, until all the portions are selected.



c) In the *Edit* dialog, select tab *Repair*.

d) In the *Surface Repair* group box, click (**Delete selected faces**).

e) At the bottom of the *Edit* dialog, click **Close**.

Create contacts between the baffles and the manifold body:

9. Expand the **Geometry** > **Parts** > **Baffles** > **Surfaces** node.
10. Multi-select **Faces**, **Faces 2**, and **Faces 3**, then right-click one of the selected nodes and select **Create Baffle**.

Rerun the Surface Wrapper:

11. Right-click **Operations** > **Surface Wrapper** and select **Execute**.

    When the process is complete, STAR-CCM+ updates the display with the new result.

These baffles are ready for use in the surface remesher. The final surface mesh is then suitable for volume meshing as long as parts are assigned to regions.

Using the Built-in Wrapper Baffle Inflation with Contact Prevention

To improve the surface definition using the default built-in wrapper baffle inflation option, include a contact prevention for the baffle surfaces and increase the surface triangulation on the baffles.

A single group contact prevention can be used in this instance, since the baffles all belong to one part and the manifold body to another. Use a small Minimum Size value due to curvature occurring in two different directions at the intersection of the baffles and manifold pipes. The Minimum Size value also has some influence over the thickness for the baffle inflation.

1. If you completed the steps for creating a true baffle, close the simulation without saving, and open the `.sim` file that you saved earlier.
2. Right-click **Geometry** > **Operations** > **Surface Wrapper** > **Contact Prevention** and select **New** > **One Group Contact Prevention Set**.
3. Select the **Contact Prevention** node and set the following properties:

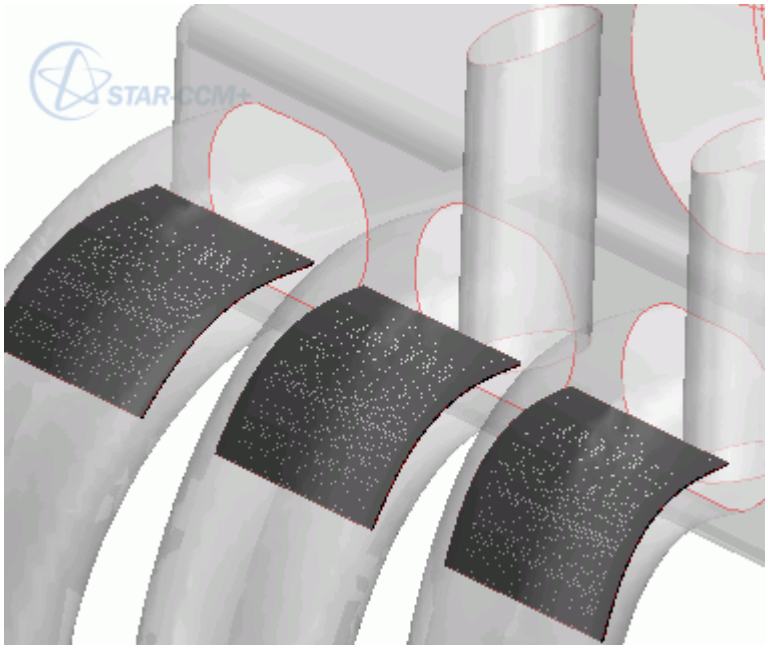
Property	Setting
Surfaces	**Baffles** and **Manifold Body**
Minimum Size	`1.0 mm`

4. Rename the **Contact Prevention** node to `Baffles CP`.
5. To obtain a more refined surface, change the baffle surface sizes using another Surface Control:
   a) Right-click **Operations** > **Surface Wrapper** > **Custom Controls** and select **New** > **Surface Control**.
   b) Select the **Surface Control 2** node and set *Part Surfaces* to **Baffles**, then set the following properties:

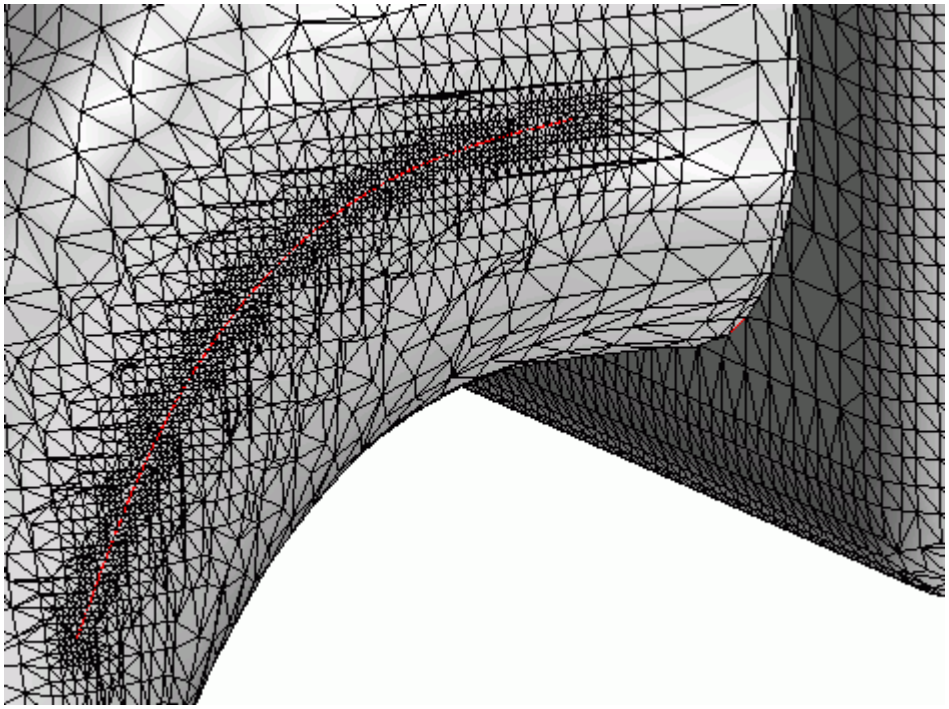
Node	Property	Setting
Controls		
└ **Target Surface Size**	*Target Surface Size*	**Specify Custom Value**
└ **Minimum Surface Size**	*Minimum Surface Size*	**Specify Custom Value**
Values		
└ **Target Surface Size**	*Percentage of Base*	`2.0`
└ **Minimum Surface Size**	*Percentage of Base*	`1.0`

6. To rerun the Surface Wrapper, Right-click **Operations** > **Surface Wrapper** and select **Execute**.

   When the process is complete, STAR-CCM+ updates the display with the new result:



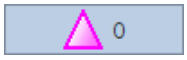
A close up of the baffle intersection area is shown below:



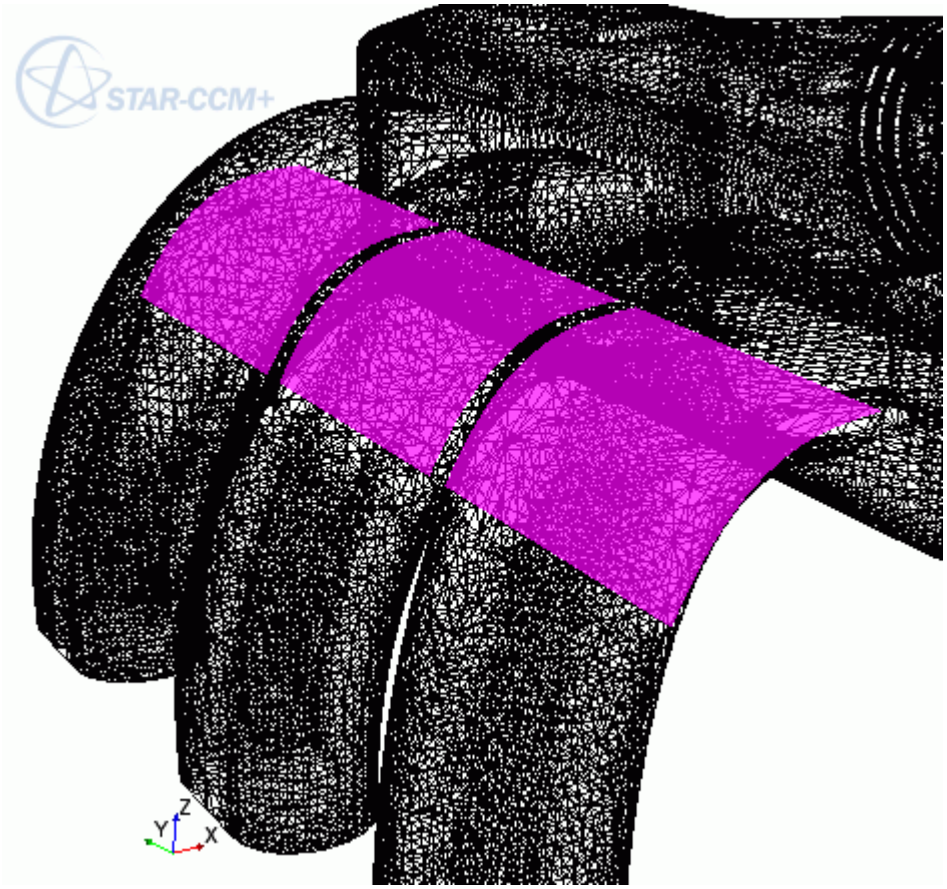
Using the Offset Faces/Edges Tool to Inflate the Baffles Manually

Use the surface preparation offset faces/edges tool to inflate surfaces so that they have a thickness.

The inflation can take place in any specified direction and can be done on more than one boundary at a time. By inflating the surfaces, the wrapper can then deal with the resultant closed body shape more easily. The feature is also incorporated as part of the final surface and a good definition is maintained.

1. If you completed the steps for creating a true baffle or using the built-in wrapper baffle inflation with contact prevention, close the simulation without saving, and open the `.sim` file that you saved earlier.
2. For *Geometry Scene 1*, select **Displayers** > **Geometry 1** > **Parts** and set *Parts* to **Baffles**, **Cross pipe**, **Inlet pipe**, **Manifold body**, **Sensor**, **Struts**, and **Valve**.
3. Multi-select the **Geometry** > **Parts** > **Baffles** and **Manifold body** parts.
4. Right-click one of the highlighted Parts and select **Repair Surface**.
5. In the *Surface Preparation* in-place dialog, click **OK**.
   No checks are required as you are not directly repairing the surface and do not need to know the location or number of problems that exist.
6. In the *Edit* dialog, select tab *Global*.
7. In *Global Tools*, click (**Inflate or translate selected faces or edges**).
8. Select the surfaces to offset:
   a) At the bottom of the *Graphics* window, deactivate (**Allow edges to be selected**) and activate  (**Allow faces to be selected**).
   b) Click (**Select faces/edges on part surfaces/part curves**).
   c) Select **Baffles**.
   d) Click **Apply** then **Close**.

   The baffle surfaces are highlighted:



Specify the offset parameters:

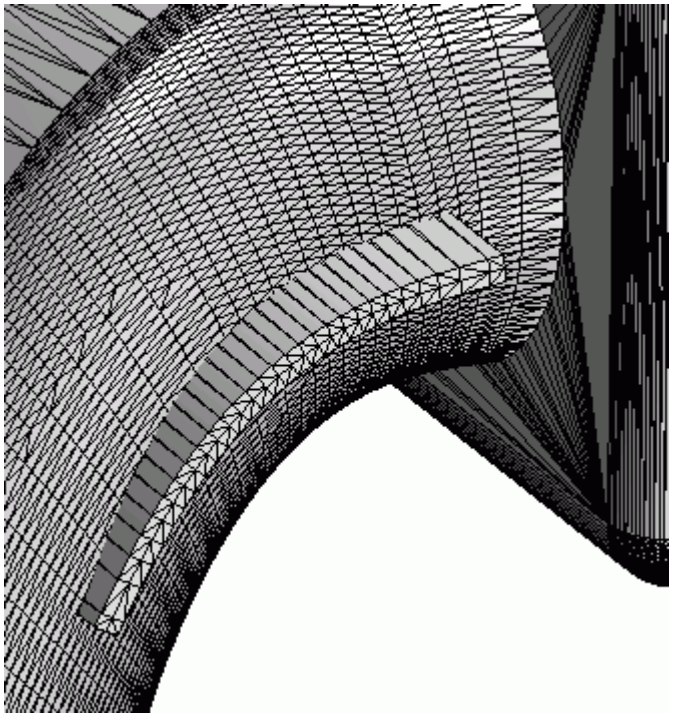
9. In the *Offset Faces/Edges Options* group box, set the following properties:

Property	Setting
Offset Mode	**Faces**
Offset Action	**Bi-directional Inflate**
Thickness	`0.003 m`
Direction	**+ Normal Direction**

Perform the inflation process:

10. Click **Offset** then **Close**.

    The scene is updated with the new geometry definition for the baffles, as shown below:



In this instance, the original baffle surface has been expanded in both directions and the sides were filled to form a closed body. The original boundary has been removed and no feature curves are included in the new body as there were no feature curves in the original surface (which is desirable in this case). No additional surface preparation is required.

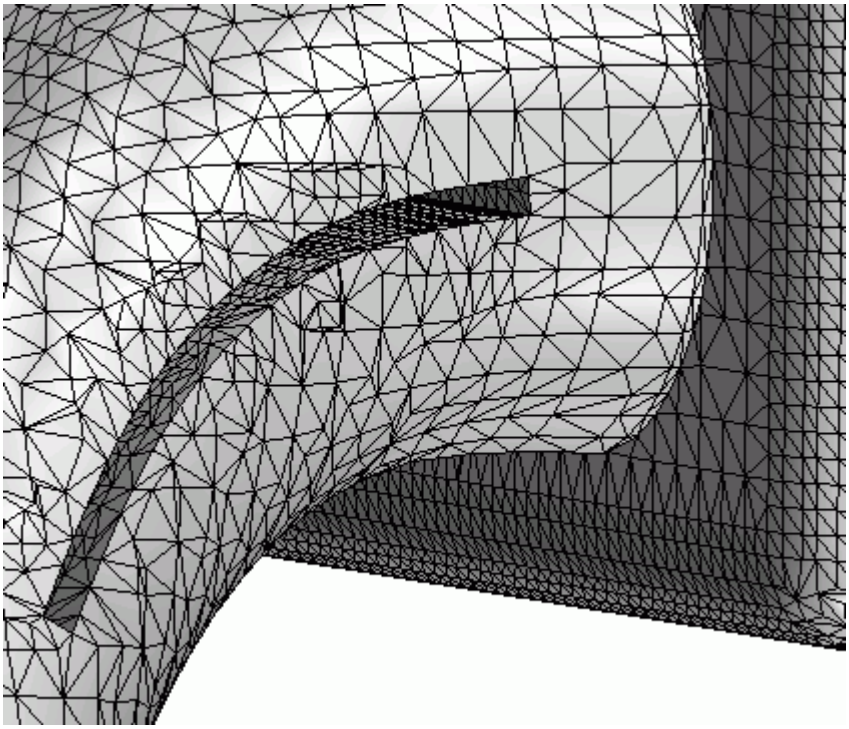
11. To obtain a more refined surface, change the baffle surface sizes using another Surface Control:
    a) Right-click **Operations** > **Surface Wrapper** > **Custom Controls** and select **New** > **Surface Control**.
    b) Select the **Surface Control 2** node and set *Part Surfaces* to **Baffles**, then set the following properties:

Node	Property	Setting
Controls		
└ **Target Surface Size**	*Target Surface Size*	**Specify Custom Value**
└ **Minimum Surface Size**	*Minimum Surface Size*	**Specify Custom Value**
Values		
└ **Target Surface Size**	*Percentage of Base*	`2.0`
└ **Minimum Surface Size**	*Percentage of Base*	`1.0`

Rerun the Surface Wrapper:

12. Right-click **Operations** > **Surface Wrapper** and select **Execute**.

In this instance, the wrapper has incorporated the inflated body into the main geometry and left a well-defined void to represent the baffle.



What to Do Next

The surface wrapper always provides a closed surface definition. The next step is to improve the overall quality of the triangulation so that you can generate the volume mesh.

Use the **Operations** > **Automated Mesh** > **Surface Remesher**. As a starting point, the same surface sizes as used for the wrapper can be used for the remesher, but alter the boundaries sizes to optimize the triangulation for the volume meshing process.

The image below shows the result of the surface remesher using the same exact sizes as for the wrapper:

Once you have obtained a valid, high-quality surface mesh, you can assign parts to regions, then generate a volume mesh using the same process as described in the Parts-Based Meshing: External Aerodynamics tutorial in this manual.

Summary

This tutorial introduced the following:

- Importing multiple CAD files and combining regions.
- Using the fill polygonal patch option to cap open boundaries.
- Setting the wrapper properties and reference values.
- Using the leak detection tool.
- Implementing contact prevention.
- Using surface refinement for a boundary.

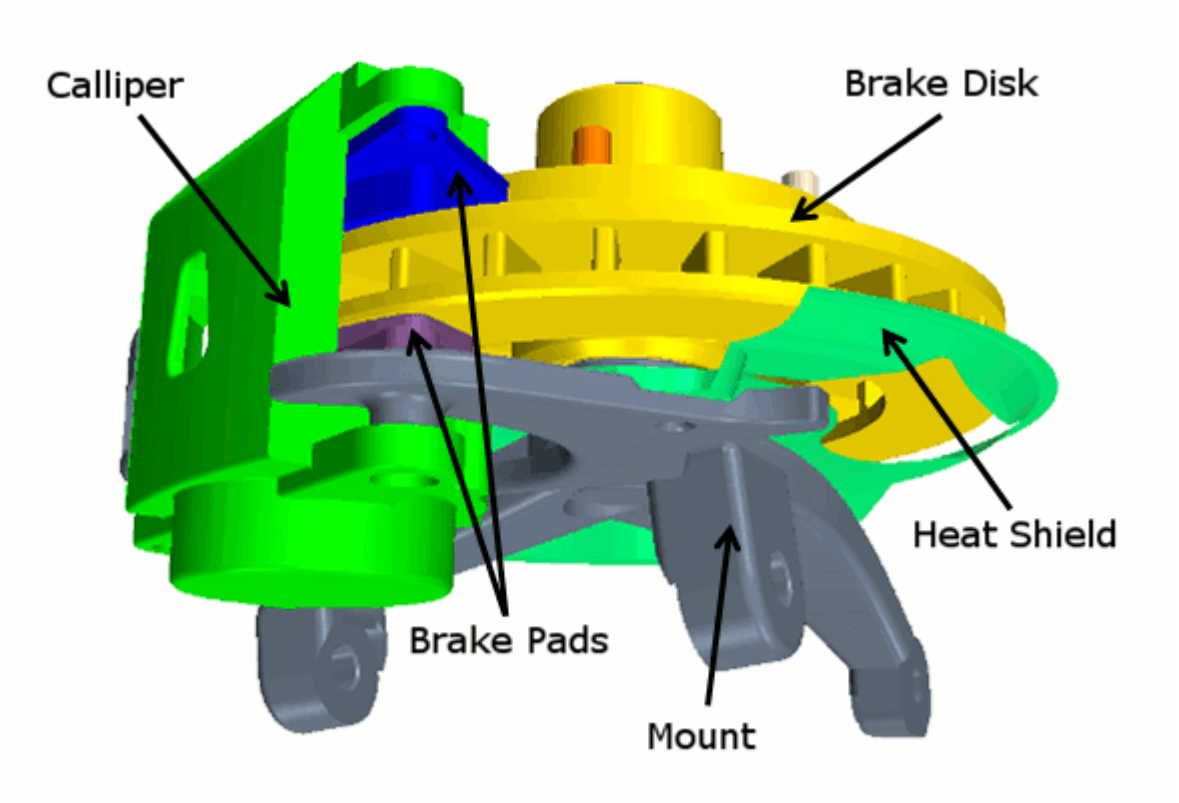
Surface Repair Tools Brake Assembly

STAR-CCM+ provides a suite of tools to help you prepare surface geometry so that it is ready for meshing. This tutorial provides an introduction to some of these tools, using the example of a brake assembly.

Volume meshers in STAR-CCM+ take a closed surface mesh as their input. Errors in the surface mesh have a direct impact on the success of the volume meshers, and on the quality of the resulting mesh.

Geometry that is imported from CAD is often not prepared for CFD analysis. The imported geometry can have unconnected parts, zero-thickness surfaces, and other defects such as gaps and holes.

Note: For complex geometries that contain too many problems to repair manually, STAR-CCM+ provides the surface wrapper. See the Surface Wrapper tutorial.



Contents:

Prerequisites
Loading the Simulation File
Storing the View
Using the Surface Repair Tools
Fixing Face Proximity: Removing Duplicate Faces
Removing Free Edges: Inflating Thin Surfaces

Removing Free Edges: Filling the Hole
Repairing Features
Closing Gaps
Boolean Operation
Merging and Imprinting Adjacent Parts
Remeshing the Surface
Checking Surface Mesh Quality in Manual Repair Mode
Summary

Prerequisites

The instructions in the Surface Repair Tools Brake Assembly tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Loading the Simulation File

For this tutorial, you are provided with a simulation file that contains pre-defined objects.

Load the initial simulation:

1. Launch STAR-CCM+.
2. Select **File** > **Load Simulation...**
3. In the *Load Simulation* dialog, click **Browse**
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/mesh/data`.
5. Select `surfaceRepairCAD.sim` then click **Open**.
6. In the *Load Simulation* dialog, click **OK**.
7. Save the simulation as `surfaceRepairTutorial.sim`.

Storing the View

Create a geometry scene, position the geometry in a specific orientation, and save the view.

Saving a custom view allows you to reposition the geometry quickly in the same orientation in the future.

1. Create a Geometry Scene.

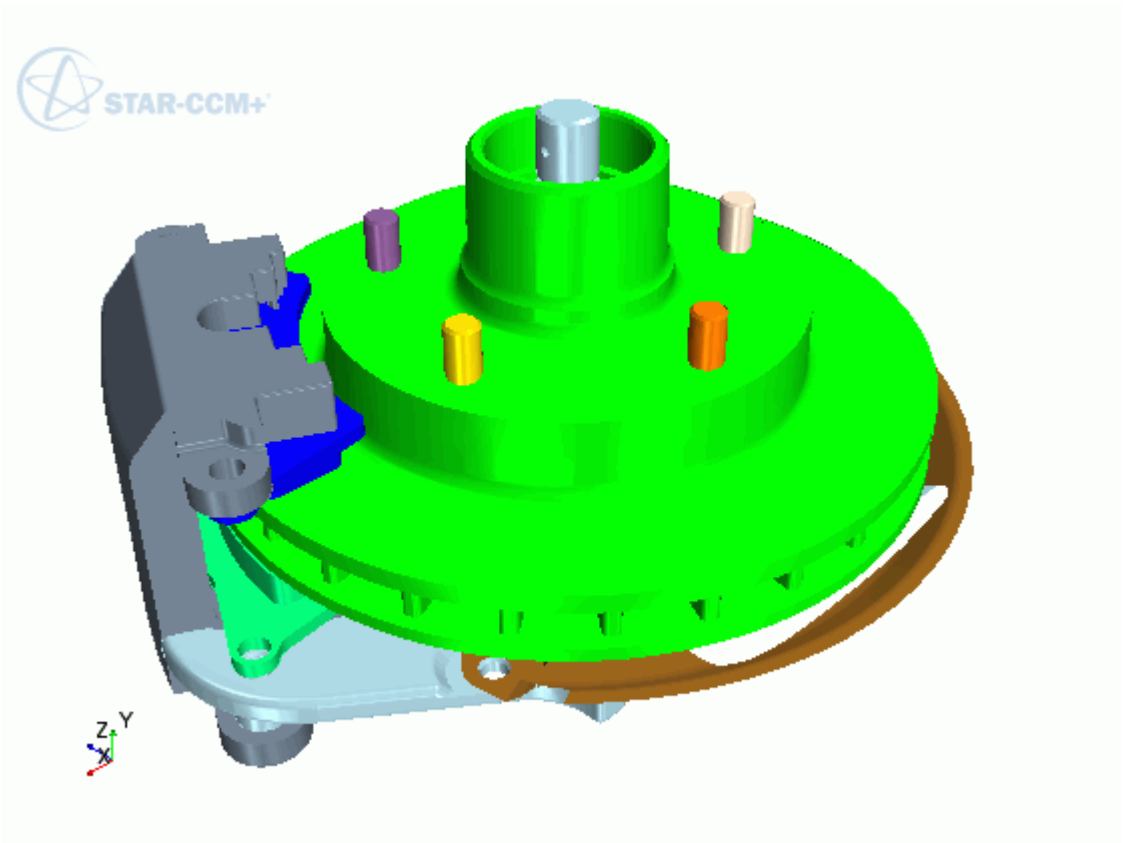
Change the color mode and remove the outline:

2. Expand the **Geometry Scene 1** > **Displayers** node and set the following properties:

Node	Property	Setting
Geometry 1	*Color Mode*	**Distinguish Inputs**
Outline 1	*Outline*	deactivated

Adjust the view:

3. In the *Vis* toolbar, click (**Save-Restore-Select Views**).
4. Select **Views** > **+X +Y -Z** > **Up +Y**.



Store the current view:

5. Click (**Save-Restore-Select Views**) and select **Store Current View**.

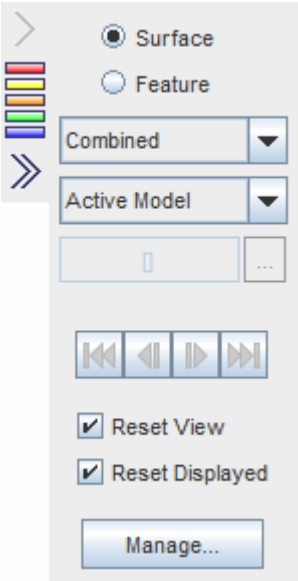
Using the Surface Repair Tools

The surface repair tools let you find and fix problems with the imported geometry.

Launch the surface repair tools:

1. In the *Mesh Generation* toolbar, click (**Launch Surface Repair**).
2. In the *Surface Repair Options* dialog, next to **Geometry Parts**, click **Select**.

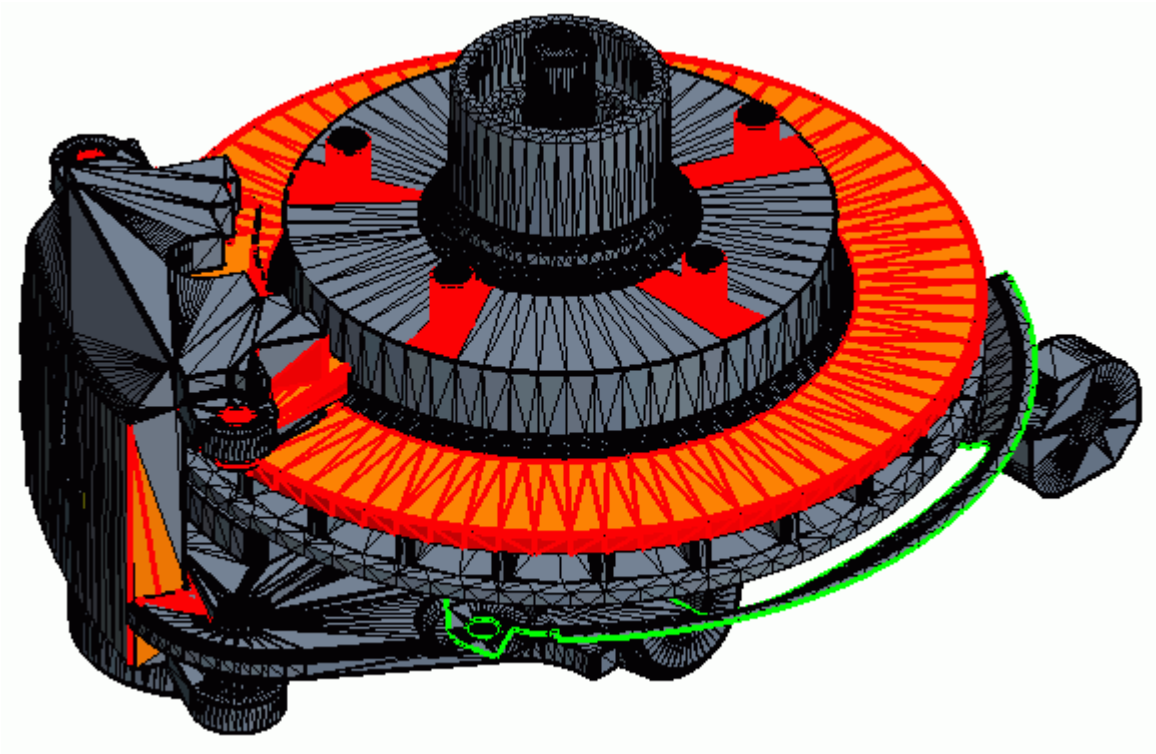
3. In the *Select Objects* dialog, make sure that all parts are selected and click **OK**.
4. In the *Surface Repair Options* dialog, click **OK**.
5. In the surface diagnostics dialog at the right side of the scene, click **Manage**.



6. Activate all Thresholds then click **OK**.

   The dialog reports the number of tessellated faces that fail each of the checks.
7. Click each of the threshold **Count** numbers in turn.

   The *Geometry Scene 1* window updates to show all faces with problems that each of the Threshold problems causes:



For example:

8. Click the red **Count** number for the pierced faces (default) threshold.

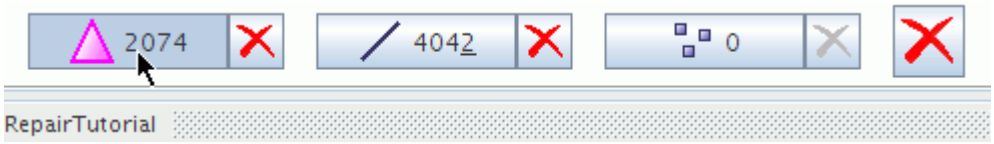
   The *Geometry Scene 1* window highlights all faces that have problems that the pierced faces cause.

Fixing Face Proximity: Removing Duplicate Faces

In this case, many of the faces that fail the Face Proximity check are due to duplicate faces on the upper surface of the brake disk.

Check for duplicate faces and merge them:

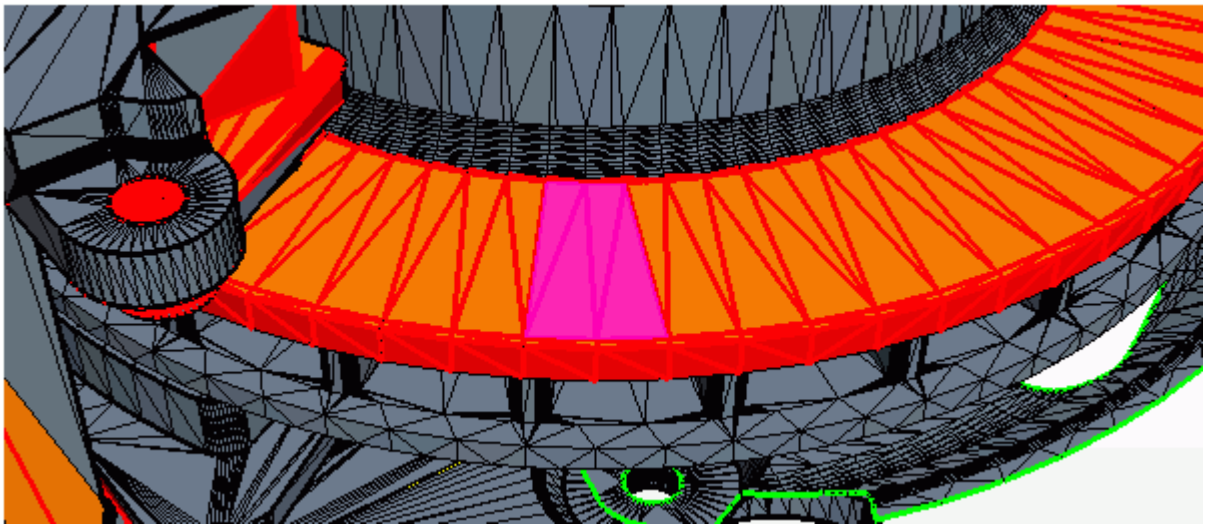
1. Near the bottom of the scene, activate only **Faces**.



2. Click (**Select None**).

   Any selections are reset.

3. Multi-select a number of the faces on the upper surface of the brake disk.



4. In the *Edit* window, make sure that the **Repair** tab is active. In the *Surface Repair* group box, click (**Delete selected faces**).

   This action deletes the chosen faces, and reveals duplicate faces beneath.

5. In the *Surface Repair* group box, click the split button in the right portion of the **Merge adjacent vertices/duplicate faces** button.

   In the *Merge Coincident Options* group box, *Operate On* is set to **Active Model**, which means that all duplicate faces in the geometry are merged. The duplicate surfaces are removed. The surface diagnostics dialog shows the reduction in the face proximity threshold count.

6. Click (**Merge adjacent vertices/duplicate faces**) to accept the default settings.

Reset the selection and deselect all faces and vertices:

7. Click (**Select none**), or press <SHIFT> and <N>.

Removing Free Edges: Inflating Thin Surfaces

Use the offset tool to convert zero-thickness surfaces into a thin volume.

Surfaces that do not enclose a particular volume are a source of free edges. You can inflate these surfaces to avoid errors that free edges cause.

1. At the top of the *Edit* window, make sure that the **Repair** tab is active.
2. In the surface diagnostics dialog, click the green **Count** number for the free edges (default) diagnostic.

   All free edges are highlighted in the *Geometry Scene 1* window.

The free edges that surround the heat shield indicate that it is a thin surface. Here, you inflate the thin surface to form a volume and so remove the free edges.

Take a closer look at the heat shield:

3. In the *Vis* toolbar:

   a) Click (**Save-Restore-Select Views**).

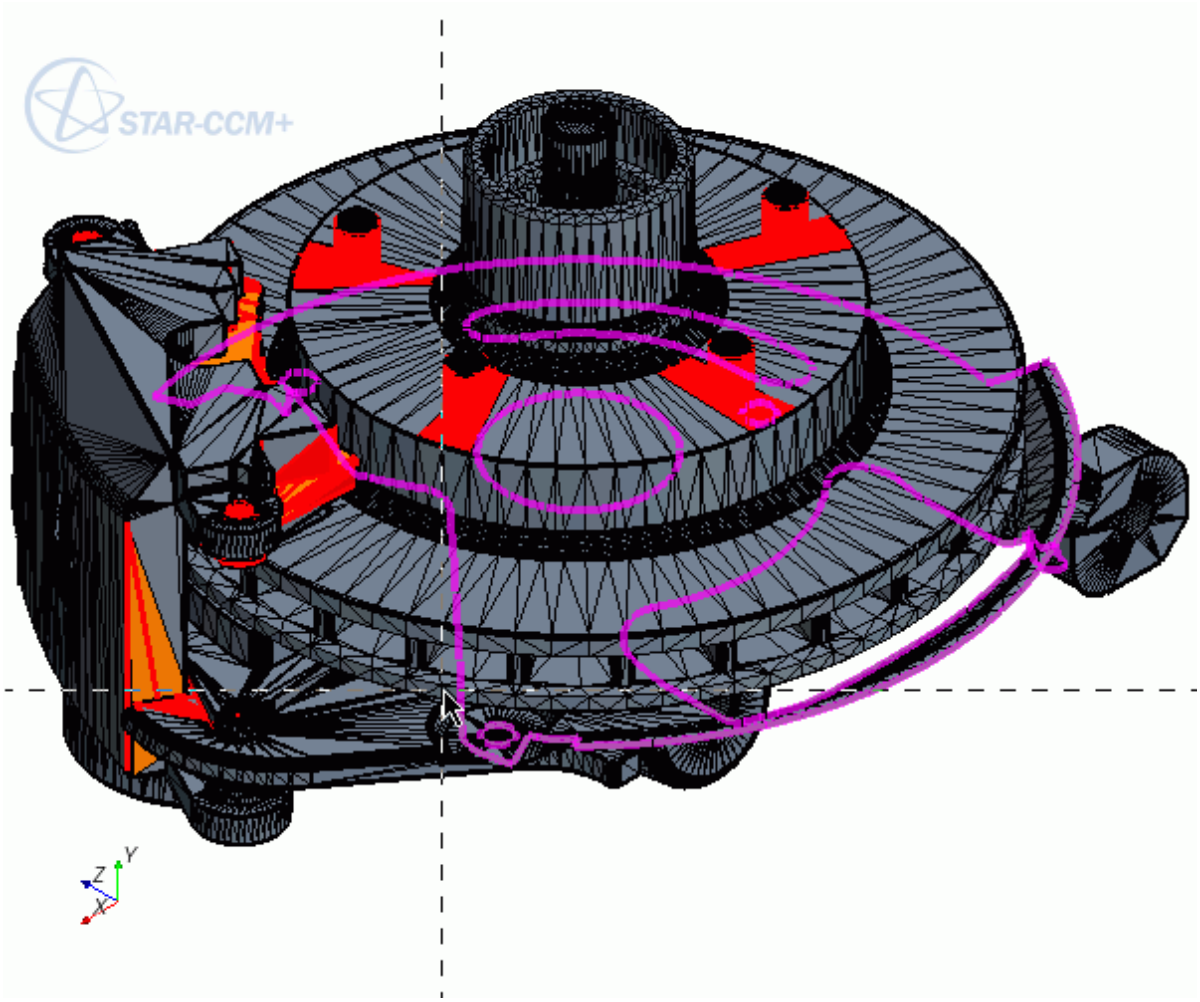
   b) Select **Restore View** > **View 1**.

   c) Click (**Rubberband Zoom**).

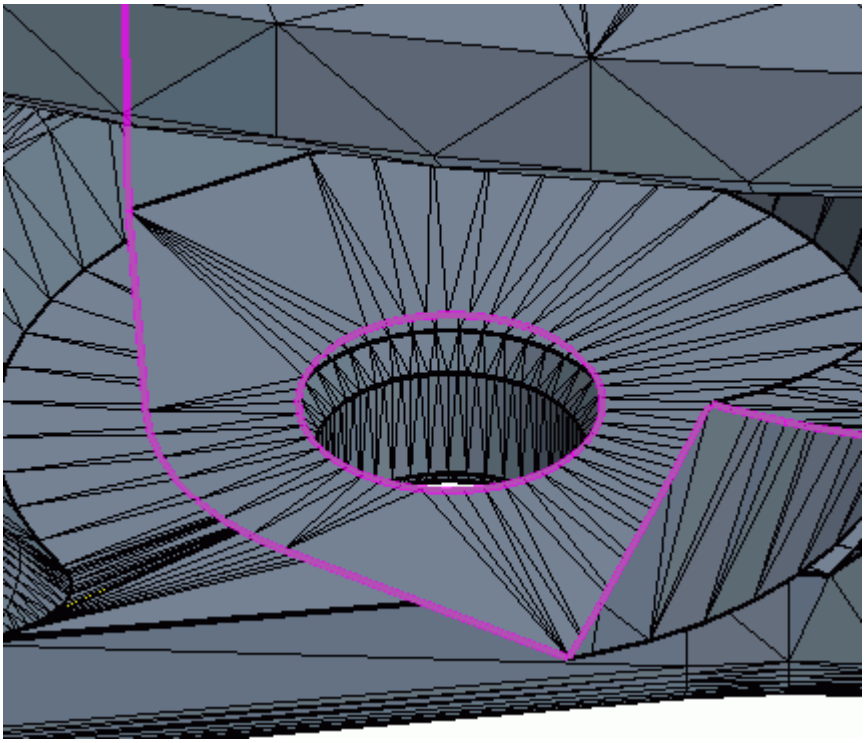
      The rubberband zoom selection tool appears as two lines, horizontal and vertical, that intersect each other in the *Geometry Scene 1* window.

4. Start the selection:

   a) Click once in the area that is shown below:



   b) Move the mouse diagonally down and right until you have selected the following area:



c) Click once to complete your selection.

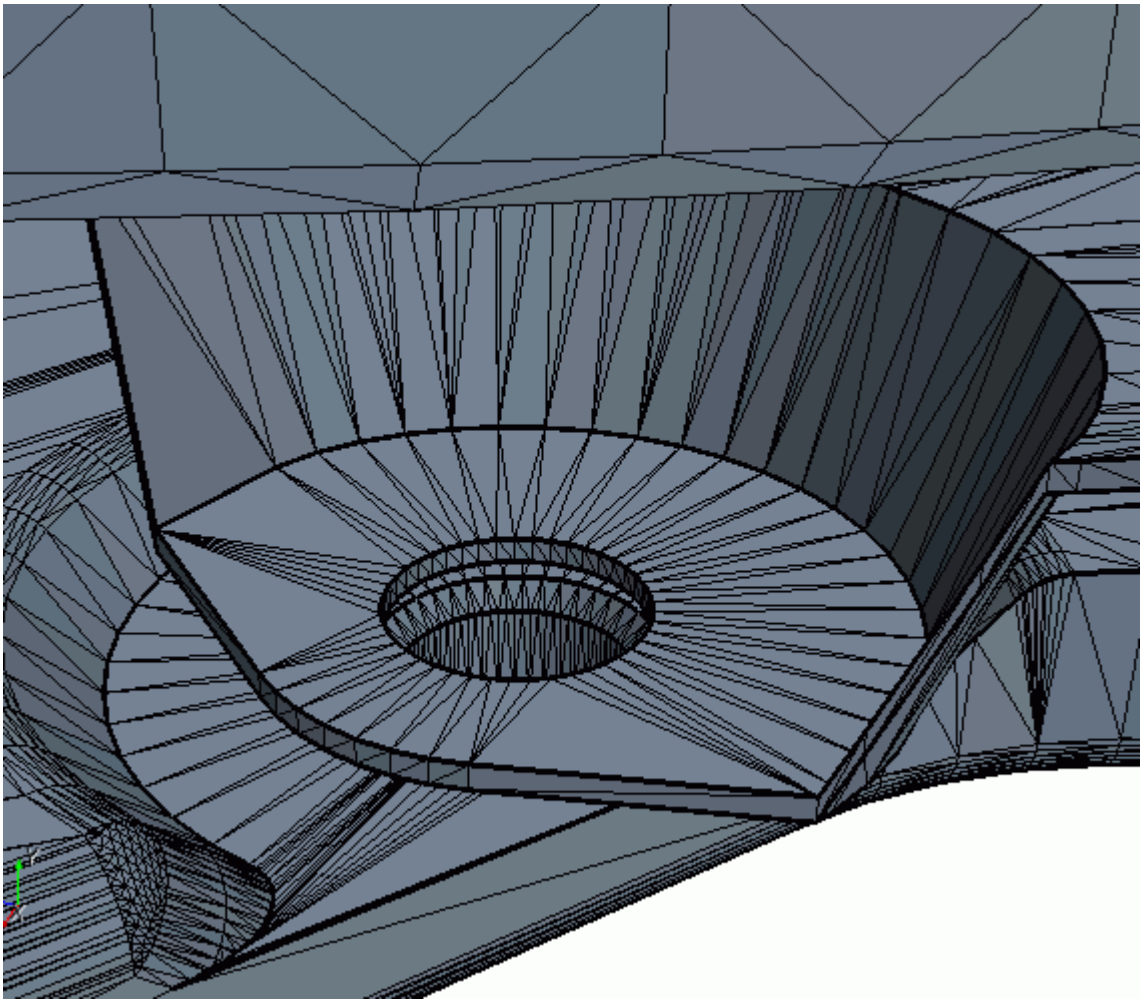
5. At the top of the *Edit* window, make the **Global** tab active and click (**Inflate or translate selected faces or edges**).
6. Reset the selection:
   a) Near the bottom of the scene, deactivate the selections of all **Faces**, **Edges**, and **Vertices**.
   b) Click (**Select None**), or press <SHIFT> and <N>.
7. Activate the **Faces** selection.
8. In the *Geometry Scene 1* window, right-click one face of the heat shield and select: **Part (Heat Shield)** > **Select**.
9. In the *Offset Faces/Edges Options* group box within the *Edit* window, set the following Properties:

Property	Setting
Offset Action	**Inflate**
Thickness	0.001 m
Direction	**+ Normal Direction**

10. Click **Offset**.

Note: Take care not to click **Offset** more than once. Doing so creates duplicate faces which can cause problems when performing boolean operations.

Since the heat shield is a volume instead of a surface, it has no free edges.



Removing Free Edges: Filling the Hole

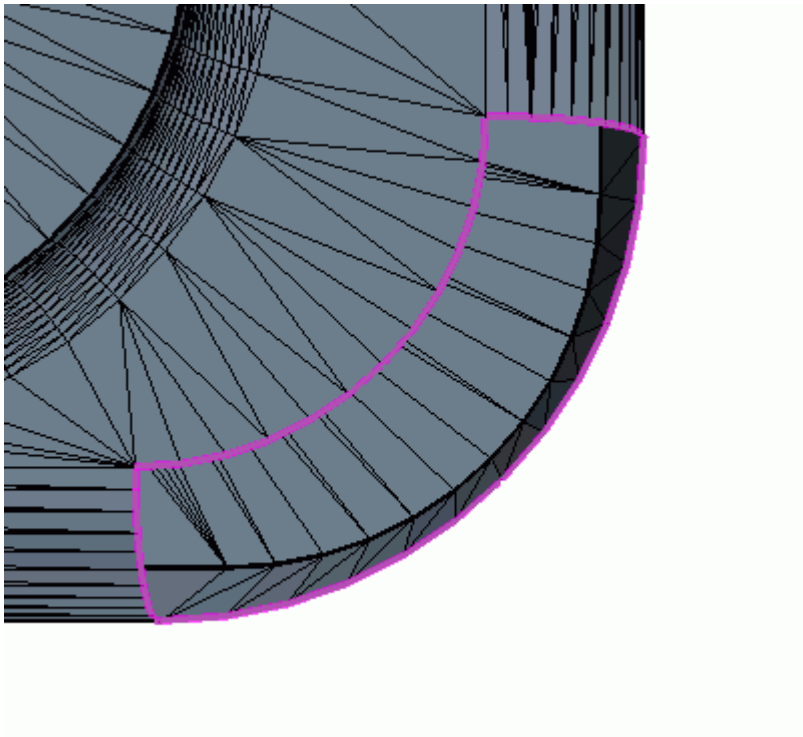
Free edges that result from holes or missing faces on the surface must also be repaired.

For surfaces that contain holes or have missing faces, you can fill the holes using the surface repair tools: Fill Polygonal Patch, or, Fill holes using selected edges. If the hole is intentional, then use the Flag edges as a feature tool and maintain the hole as a feature of the geometry.

1. At the top of the *Edit* window, make the **Repair** tab active.
2. In the surface diagnostics dialog, click the green **Count** number for the free edges (default) diagnostic.

   The remaining free edges are highlighted on the rim of a hole in *Geometry Scene 1*.

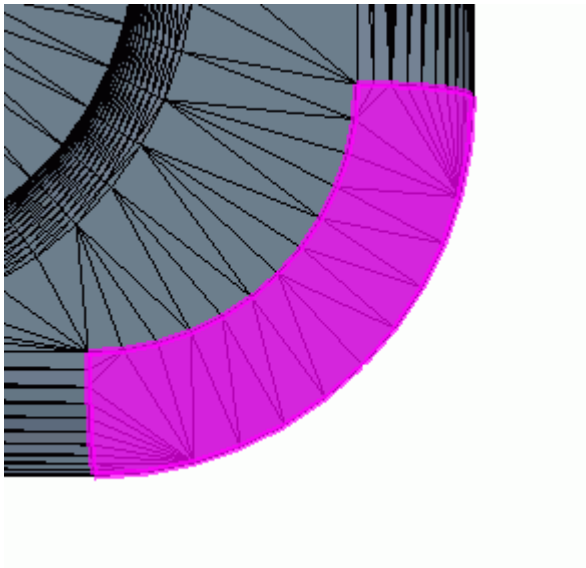
3. In the *vis* toolbar, click (**Save-Restore-Select Views**) and select **Standard Views** > **(+ Z) top**.
4. Use the (**Rubberband Zoom**) selection tool to view the following area:



The free edges around this hole are flagged as feature edges. A feature edge is an edge that is important to the geometry, so leaving these free edges flagged as features edges would interfere with the fill holes operation.

5. In the *Feature Repair* group box, click (**Unflag feature edges**).
   The free edges are no longer marked as feature edges.

6. In the *Surface Repair* group box, click (**Fill holes using selected edges**).
   The hole is filled, but the local curvature is not maintained. As a result, the patch is not consistent with the surrounding geometry:

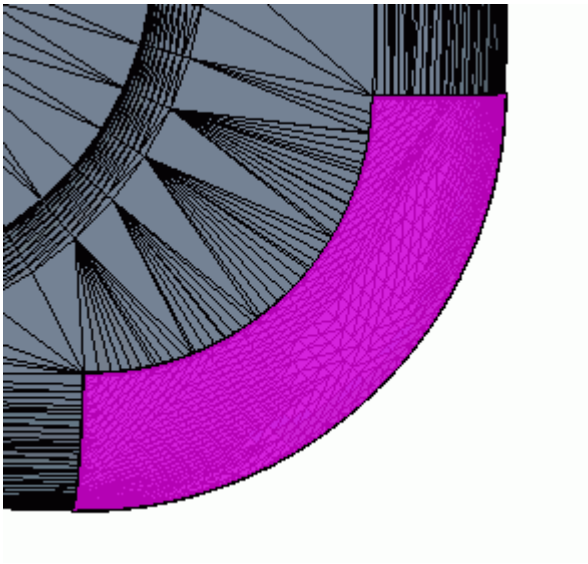


7. At the bottom of the *Edit* window, click (**Undo**).

Note: If you use the **Undo** button to remove any unwanted surface repairs on existing faces, the faces with those surface repairs disappear from the *Geometry Scene 1* window. The faces are not permanently deleted. To redisplay the missing faces, in the *Display Control* toolbar near the top of the scene, click (**Show all faces**).

8. In the *Surface Repair* group box, click the split button in the right portion of the **Fill holes using selected edges** button.
9. In the *Fill Hole Options* group box, activate **Maintain Local Curvature**.
10. Click (**Fill holes using selected edges**).

The hole is filled with a surface that follows the local curvature.



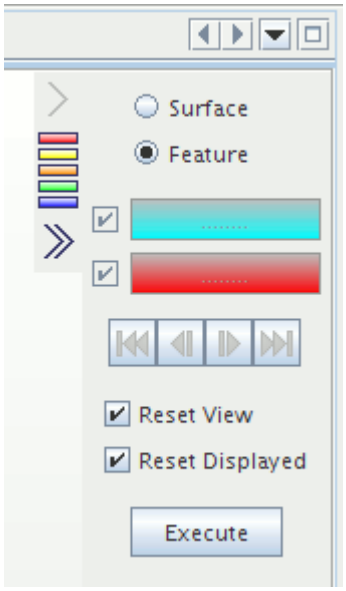
In the surface diagnostics dialog, there are no free edges, that is, the green **Count** number is zero.

Repairing Features

Ensure that all feature curves that are required to represent the geometry are correctly identified in the surface mesh.

Feature curves are edges that you wish to retain in the surface mesh.

1. Set the mode of the surface diagnostics dialog to **Feature**.



2. In the feature diagnostics dialog, click **Execute**.

   This process identifies feature errors - that is, edges that are features but are not represented as such, and edges that are incorrectly represented as features.

3. In the **Repair** tab, in the *Feature Repair* group box, click (**Fix all feature curve errors**).
   All feature errors are now corrected.

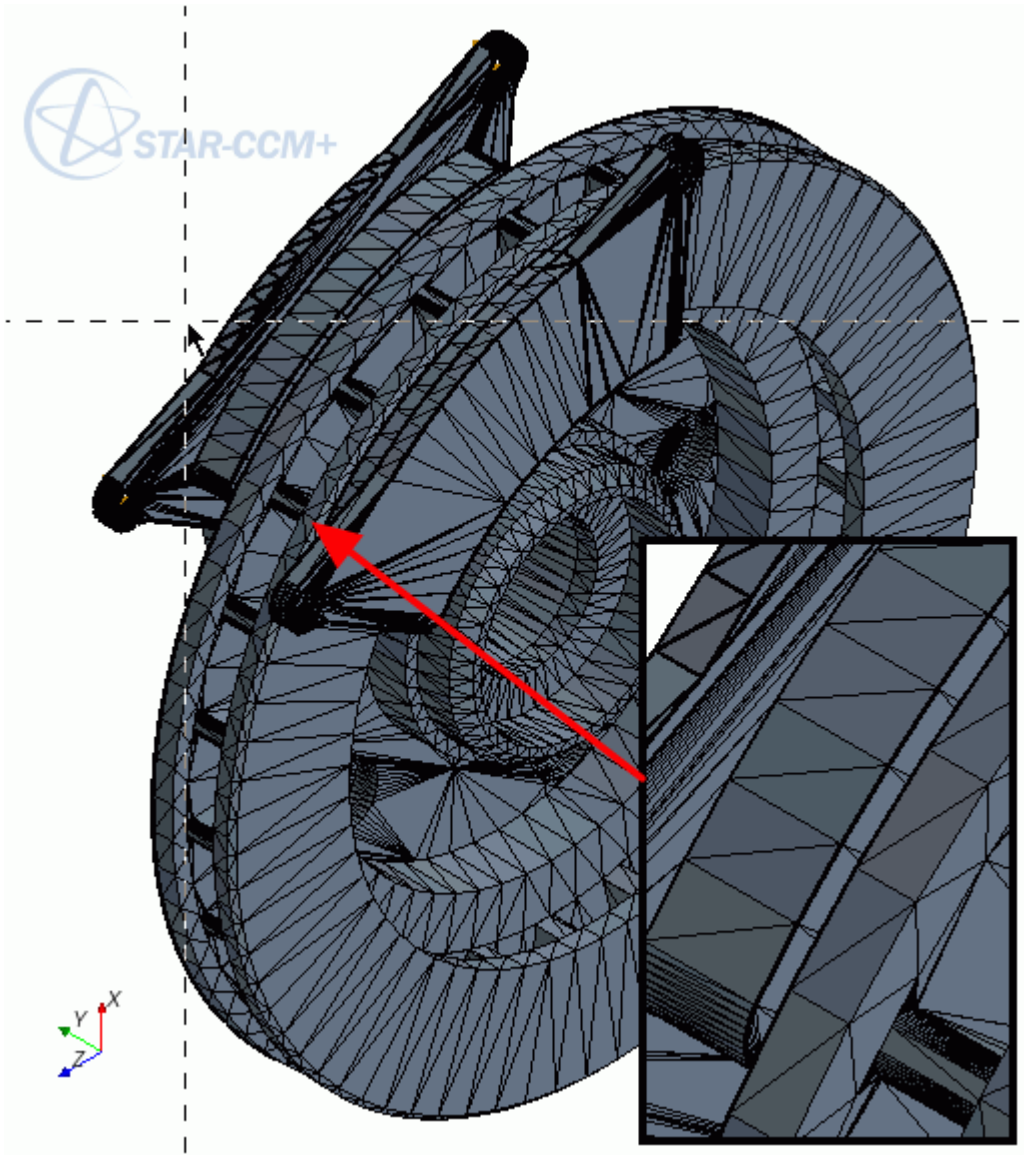
Closing Gaps

Project vertices onto a set of faces to close a gap.

An inspection of the brake assembly geometry reveals that there are gaps between the brake pads and the disks. For this tutorial, you close the gaps using the (**Project selected vertices**) tool.

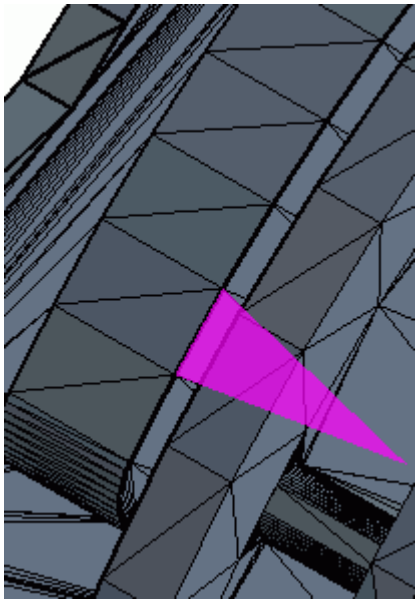
Select the required vertex and faces, and project the vertices:

1. At the top of the *Edit* window, make sure that the **Repair** tab is active.
2. In the *Vis* toolbar, click (**Save-Restore-Select Views**) and select **Restore View** > **View 1**.
3. In the *Display Control* toolbar near the top of the scene:
   a) Click (**Hide all faces**).
   b) Click (**Show faces on Part Surfaces**).
   c) In the *Show Part Surfaces* dialog, select **Brake Disk**, **Lower Pad**, and **Upper Pad**.
   d) Click **Apply**, then click **Close**.
4. Click (**Save-Restore-Select views**) and select **Views** > **+X -Y +Z** > **Up +X**.
5. Use the (**Rubberband Zoom**) tool to view the following area:

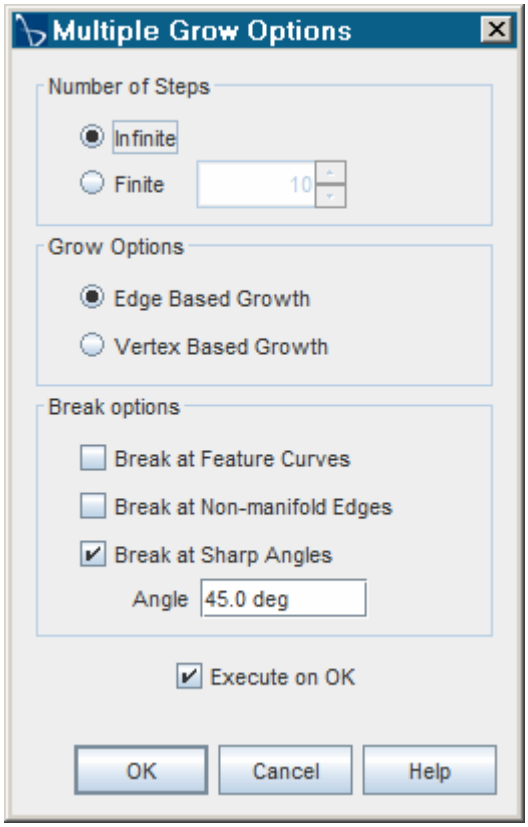


Select all the faces on the surface of the brake pad closest to the brake disk:

6. Select a face on the surface of the brake pad closest to the brake disk:



7. In the *Selection Control* toolbar at the left of the scene, click the split button of the (**Multiple grow selected**) button.
8. In the *Multiple grow options* dialog, activate **Break at Sharp Angles**.



9. Click **OK** to accept the remaining default selections.

   The surface of the brake pad is selected.

Select all of the vertices on the surface of the brake pad closest to the brake disk:

10. In the *Selection Control* toolbar:

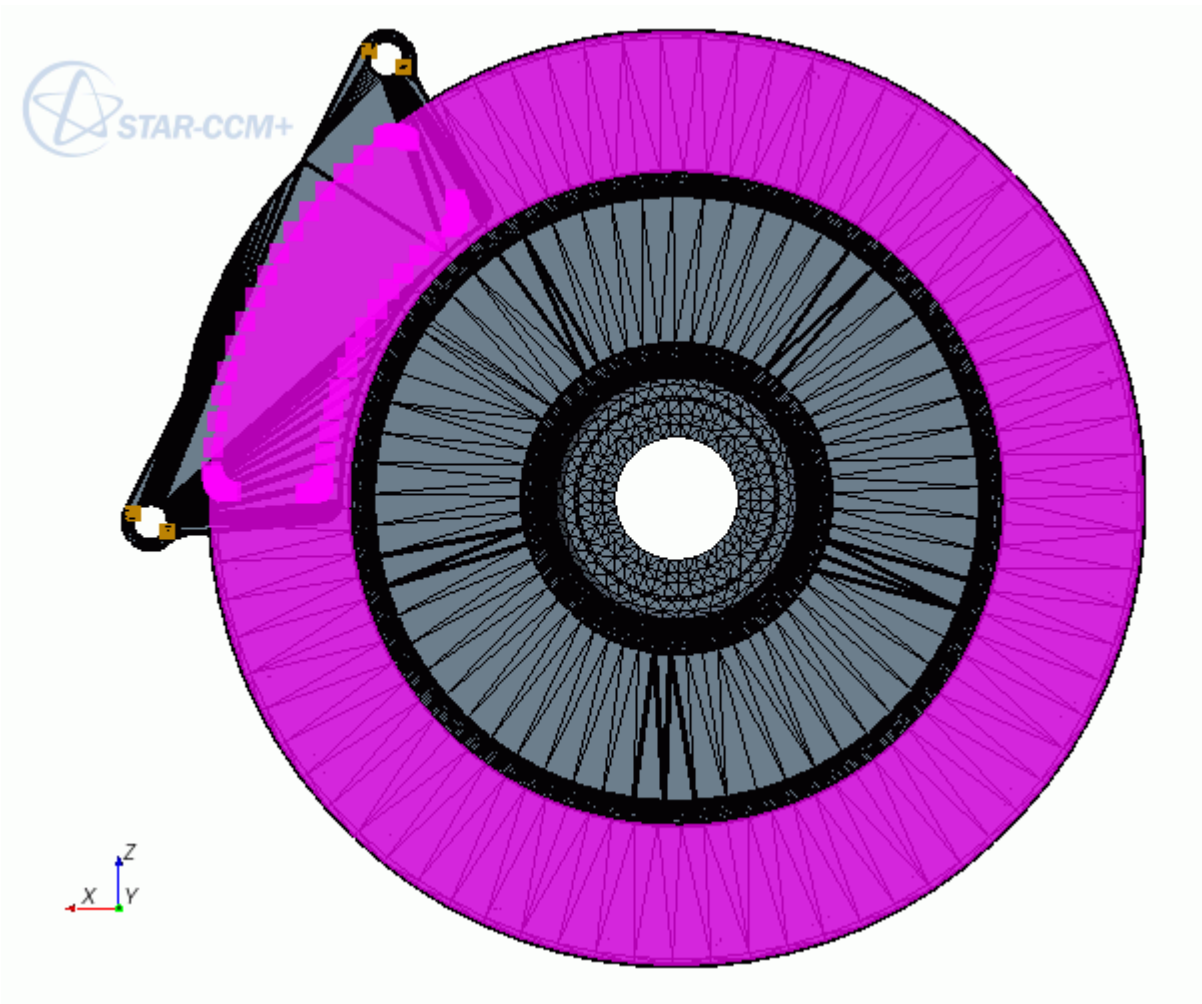
    a) Click (**Select attached vertices**).

    b) Next to the **Faces** button near the bottom of the scene, click (**Clear selected faces**).
    The **Faces** option must remain active.

11. In the *Vis* toolbar, click (**Save-Restore-Select Views**) and select **Standard Views** > **(+Y) right**.

Select the faces on the surface of the brake disk:

12. Press and hold `<Ctrl>` then double-click a face on the brake disk surface.
    All vertices of the brake pad and faces of the brake disk on adjacent surfaces are selected:



Project the vertices onto the faces:

13. Under the **Global** tab, in the *Global Tools* group box, click (**Project selected vertices**).
14. In the *Projection Options* in-place dialog, make sure that the settings are as follows:

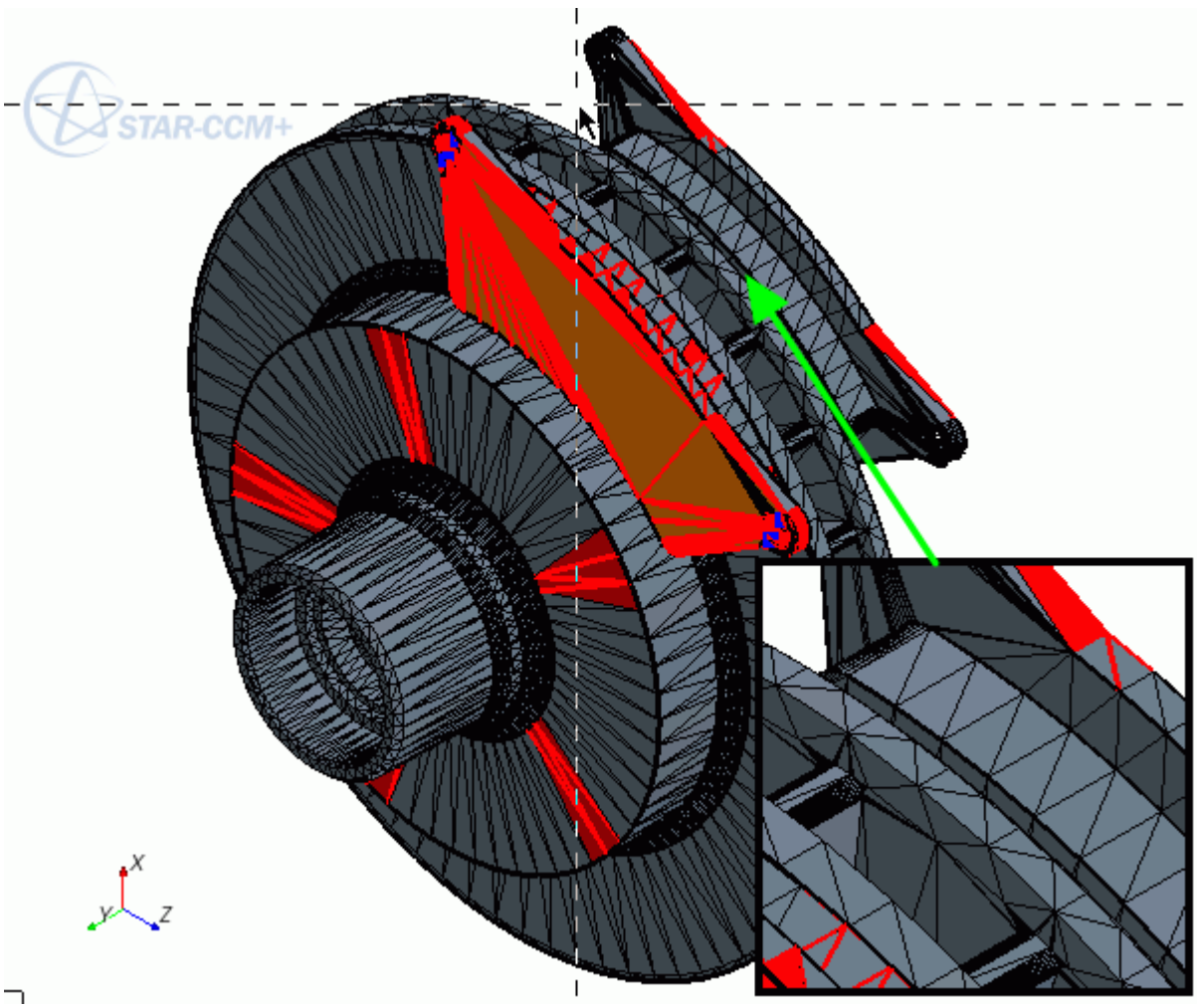
Property	Setting
Projection Mode	**Project to a Surface**
Projection Method	**Shortest Distance**

15. Click **Project Vertices**.

    The gap between one brake pad and the brake disk is closed.

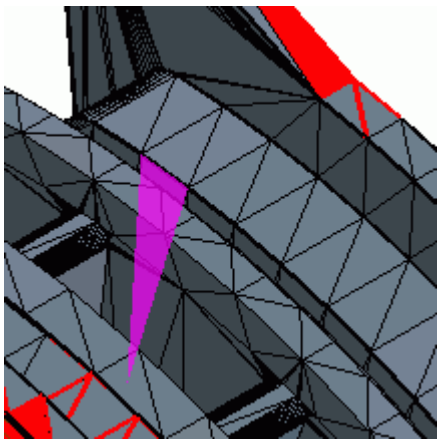
Proceed to close the gap between the other brake pad and the brake disk:

16. Near the bottom of the scene, click (**Select none**).
17. In the Vis toolbar, click (**Save-Restore-Select Views**) and select **Views** > **+X +Y +Z** > **Up +X**.
18. Use the (**Rubberband Zoom**) tool to view the following area:



Select all the faces on the surface of the second brake pad closest to the brake disk:

19. Press and hold `<Ctrl>` then double-click a face on the surface of the brake pad closest to the brake disk.



The surface of the brake pad is selected.

Select all of the vertices on the surface of the brake pad closest to the brake disk:

20. In the *Selection Control* group box:

    a) Click (**Select attached vertices**).

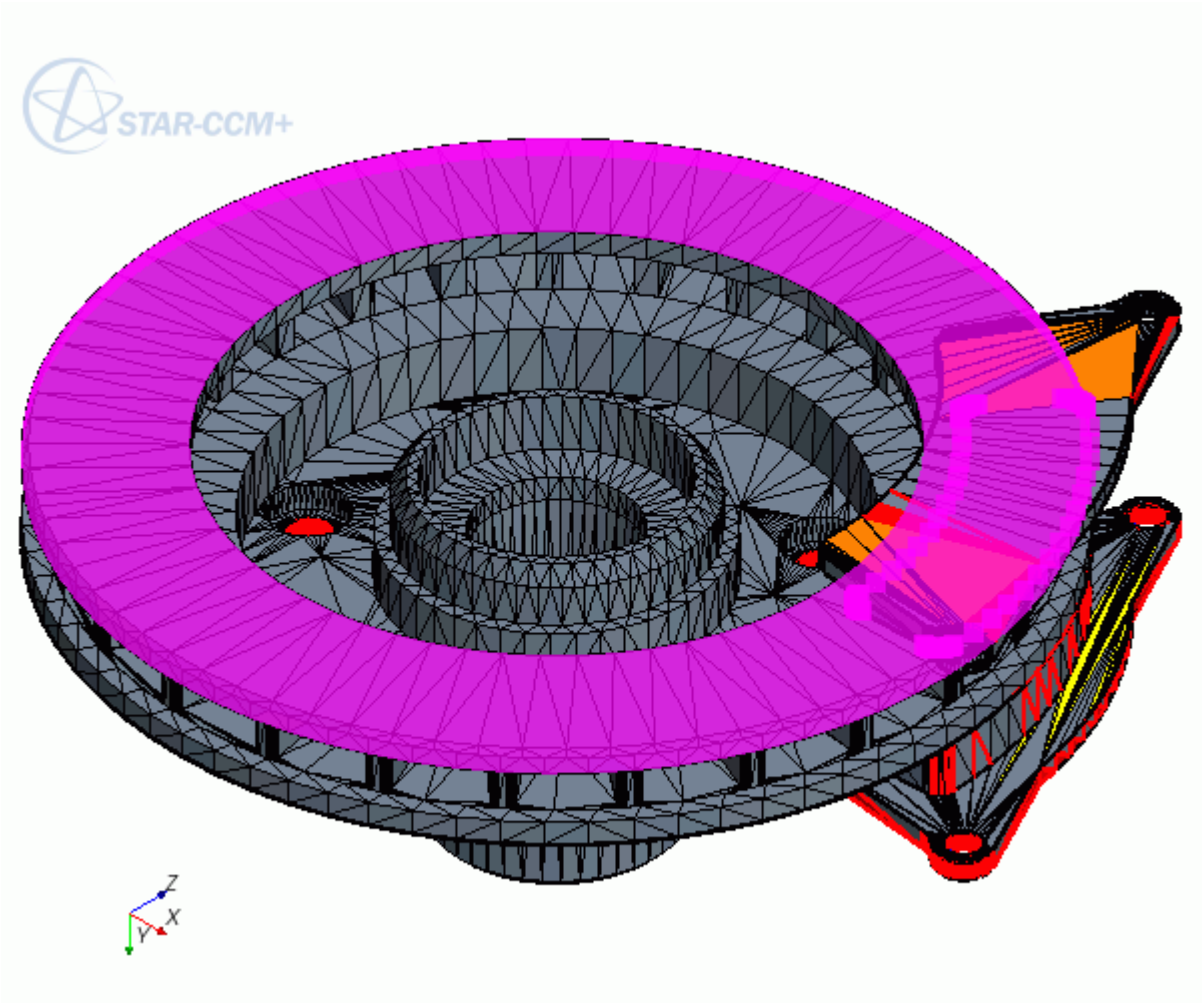
    b) Click (**clear selected faces**).
    The **Faces** option must remain active.

21. In the *Vis* toolbar, click (**Save-Restore-Select Views**) and select **Views** > **+X -Y -Z** > **Up -Y**.

Select the faces on the surface of the brake disk:

22. Press and hold `<Ctrl>` then double-click a face on the brake disk surface.

    All vertices of the brake pad and faces of the brake disk on adjacent surfaces are selected:



Project the vertices onto the faces:

23. In the *Global Tools* group box, click (**Project selected vertices**).
24. In the *Projection Options* in-place dialog, make sure that the settings are as follows:

Property	Setting
Projection Mode	**Project to a Surface**
Projection Method	**Shortest Distance**

25. Click **Project Vertices**.

    The gap between the other brake pad and the brake disk is closed.

    The Pierced faces and Face Proximity thresholds show deterioration as extra faces are now in close contact without intersecting. This problem is resolved in the step for imprinting surfaces.
26. To exit surface repair mode, click **Close**.

Boolean Operation

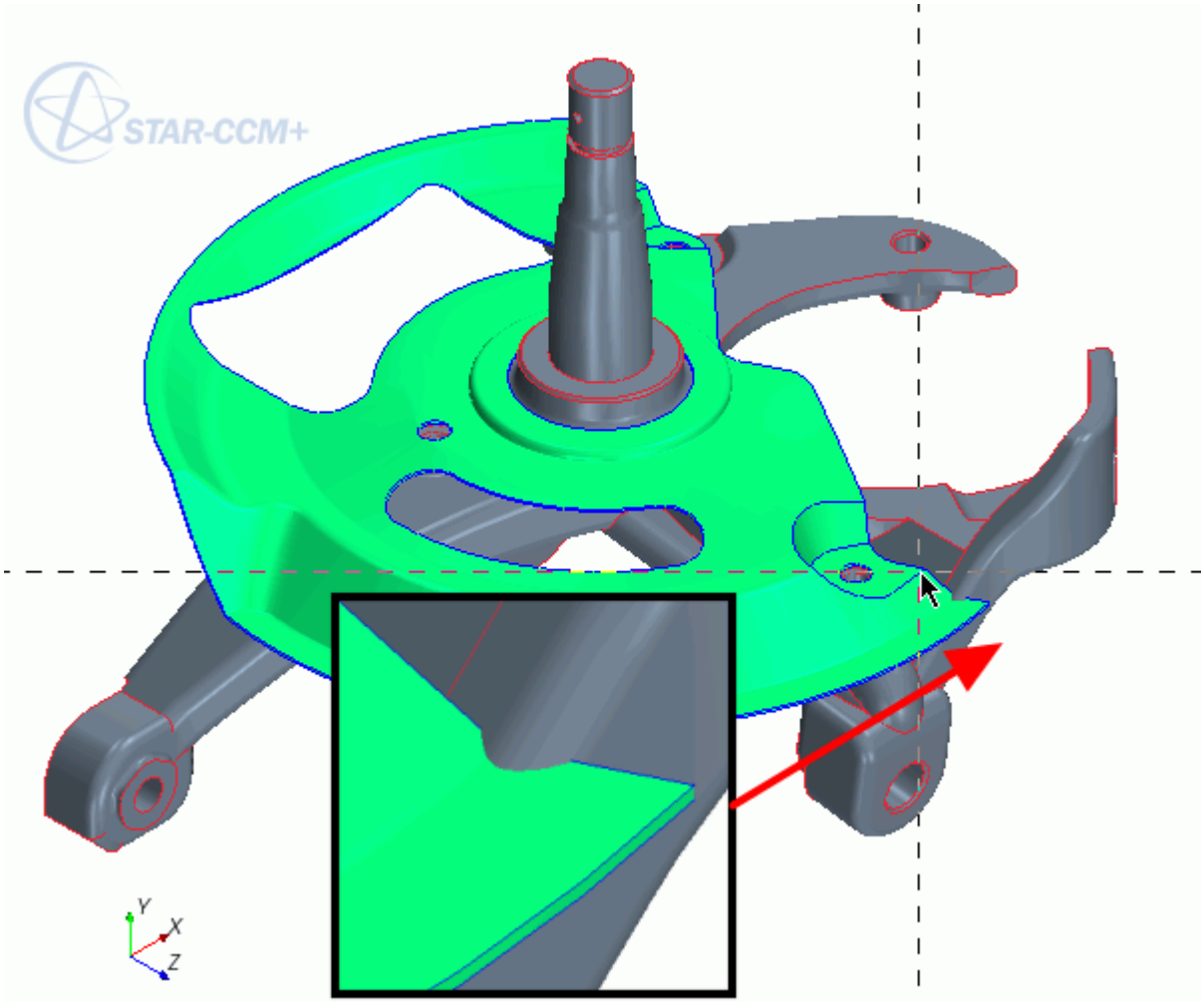
There are several components of the brake assembly that overlap each other or do not intersect cleanly. These components require further attention using the Boolean operation tools.

Note: You can click (**Show All Mesh**) or (**Hide All Mesh**) in the *Vis* toolbar depending if you want to view the mesh or not.

Fix the overlapping heat shield and mount:

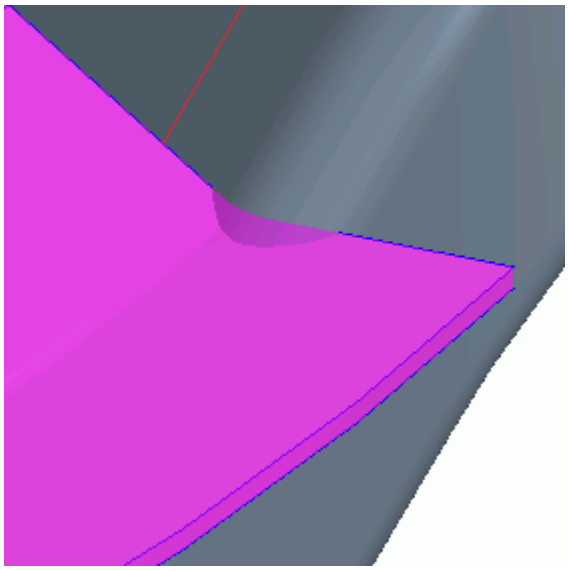
1. Make sure that *Geometry Scene 1* is active.
2. In the *Vis* toolbar, click (**Save-Restore-Select Views**) and select **Restore View** > **View 1**.
3. View only the heat shield and mount:
   a) Right-click **Geometry Scene 1** > **Displayers** > **Geometry 1** > **Parts** and select **Edit**.
   b) In the *Parts* dialog, deselect all parts.
   c) Select the **Heat Shield** and **Mount** parts then click **OK**.

   Only the selected parts are displayed in the *Geometry Scene 1* window.
4. In the Vis toolbar, click (**Save-Restore-Select Views**) and select **Views** > **-X +Y +Z** > **Up +Y**.
5. Use the (**Rubberband Zoom**) tool to view the area where the parts overlap:



6. Select the **Geometry** > **Parts** > **Heat Shield** node.

   The heat shield appears highlighted in the *Geometry Scene 1* window and shows where the Heat Shield intersects the Mount.



7. Create a new Subtract part that does not intersect the Mount:
   a) Multi-Select the **Geometry** > **Parts** > **Heat Shield** and **Mount** nodes.
   b) Right-click one of the selected nodes and select **Boolean** > **Subtract Parts**.
   c) In the *Subtract Parts* dialog, select **Heat Shield** as the **Target Part**.
   d) Click **OK** to accept **Discrete** as the default **Boolean Mode**.

   Since the geometry has been modified, you are operating on a tessellated definition.

   The portions of the Heat Shield part that overlap the Mount part are removed, and a new Subtract part is generated which cleanly intersects with the Mount part.
8. Delete the Heat Shield part.
9. Rename the **Subtract** node to `Heat Shield`.
10. Display the new Heat Shield part in the Geometry Scene 1 window:
    a) Click and hold the **Heat Shield** node.
    b) Drag the part to a space in the *Geometry Scene 1* window.
    c) Select **Part Displayers** > **Add to Geometry 1**.

Fix the Intersecting Brake Disk and Rotor Studs, create a new Subtract part for the five Rotor Stud parts and the Brake Disk part.

11. In the *Vis* toolbar, click (**Save-Restore-Select Views**) and select **Views** > **+Y** > **Up +X**.
12. View only the five Rotor Stud parts and the Brake Disk part:
    a) At the top of the simulation tree, click *Scene/Plot*.
    b) Edit the **Geometry Scene 1** > **Displayers** > **Geometry 1** > **Parts** node and make the five **Rotor Stud** parts and the **Brake Disk** part the active parts.
    c) Click **OK**.

    Only the selected parts are displayed in the Geometry Scene window.
13. Select the rotor stud and brake disk parts that intersect:
    a) Expand the **Geometry** > **Parts** node and multi-select the five rotor stud nodes.
    b) While the rotor stud nodes are still selected, hold down `<Ctrl>` and select the **Brake Disk** node.
14. Create the Subtract part:
    a) Right-click one of the selected nodes and select **Boolean** > **Subtract Parts**.
    b) In the *Subtract Parts* dialog, set **Target Part** to **Brake Disk**.
    c) Click **OK** to accept **Discrete** as the default **Boolean Mode**.

    A new **Subtract** part appears under **Geometry** > **Parts**.
15. Delete the **Geometry** > **Parts** > **Brake Disk** part.
16. Rename the **Subtract** node to `Brake Disk`.
17. Display the new Brake Disk part in the *Geometry Scene 1* window:
    a) Click and hold the **Brake Disk** node.
    b) Drag the part to a space in the *Geometry Scene 1* window.
    c) Select **Part Displayers** > **Add to Geometry 1**.

Merging and Imprinting Adjacent Parts

Generate a surface mesh suitable for CFD, imprint adjacent parts, and merge the faces and nodes.

View all **Parts** in **Geometry Scene 1**:

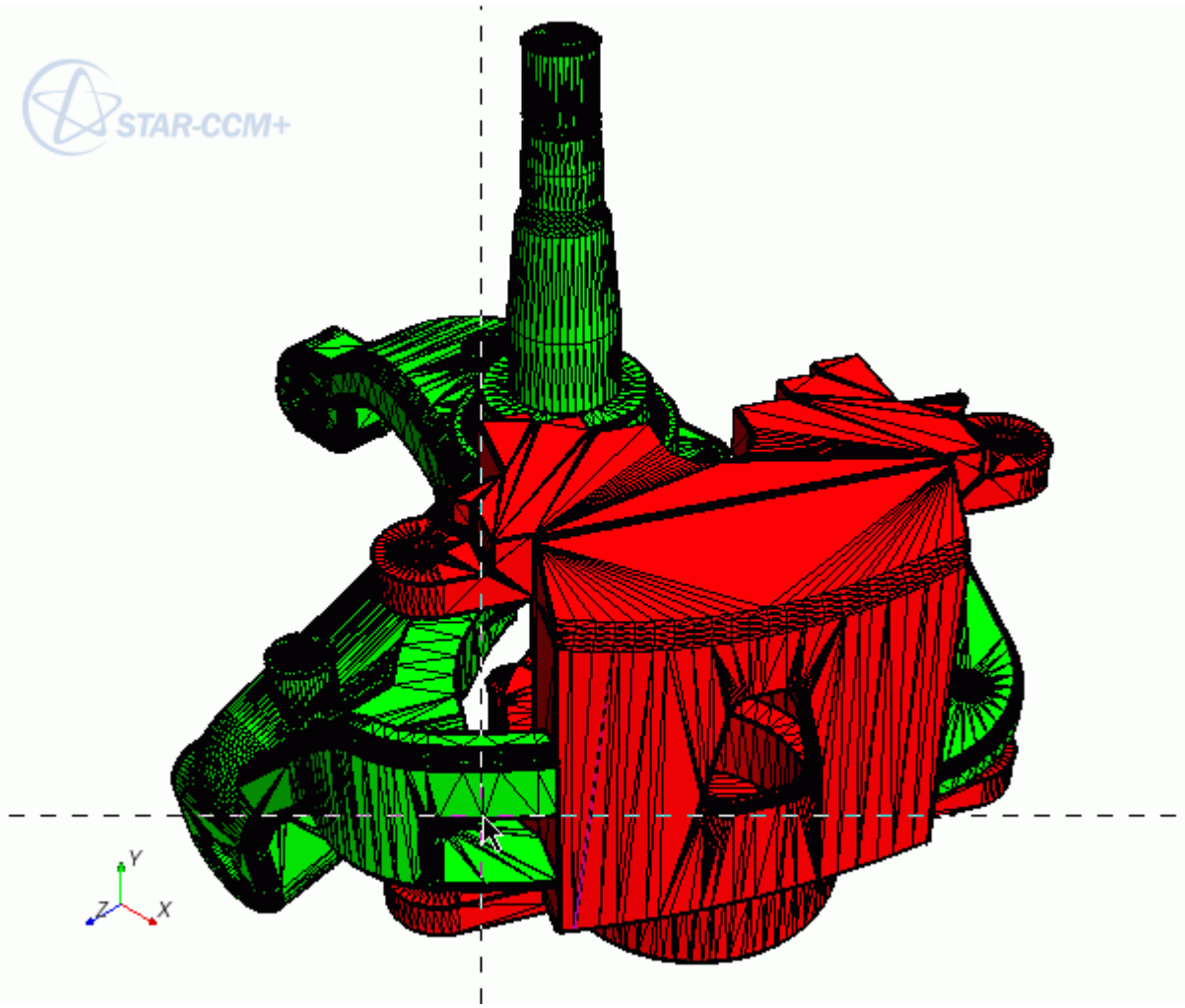
1. In the *Mesh Generation* toolbar, click (**Launch Surface Repair**).
2. Set **Geometry Parts** to all parts and click **OK**.
3. At the top of the *Edit* window, make the **Global** tab active and click (**Merge/Imprint single or multiple parts**).
4. In the *Merge/Imprint Options* group box, set the **Imprint Mode** to **Multi-Part Imprint**.
5. In the *Part Browser* box, click **Find Pairs**.

   The Part Browser identifies adjacent pairs of parts. Most of the pairs of parts can be imprinted immediately by using Imprint All. However, in the case of the Calliper and Mount parts, surfaces that are next to the contact area meet with a small angle. These parts need imprinting with different settings.

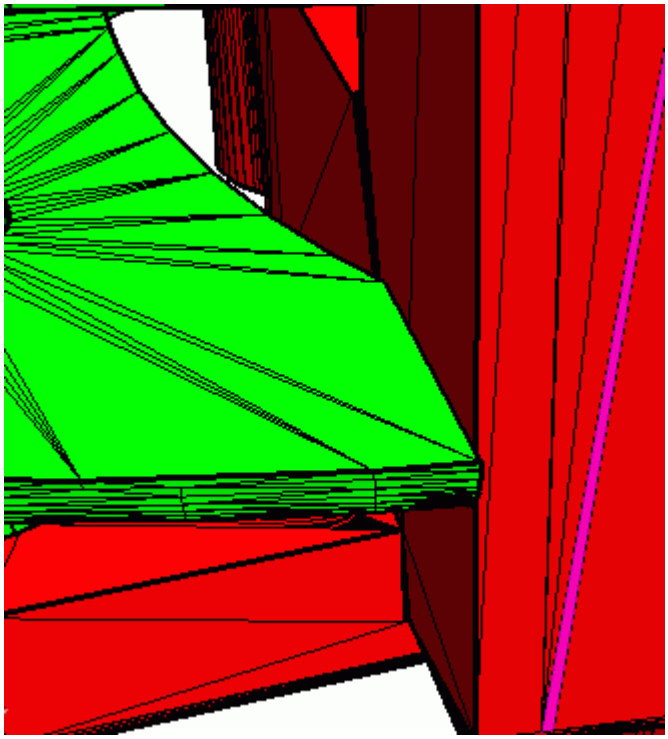
View the surfaces on the **Calliper** and **Mount** parts that meet with a small angle:

6. In the *Part Browser* box within the *Merge/Imprint Options* group box, select the **Calliper** and **Mount** parts from the drop-down menus.

   The **Calliper** and **Mount** parts are displayed in the *Geometry Scene 1* window.
7. In the *Vis* toolbar, click (**Save-Restore-Select Views**) and select **Views** > **+X +Y +Z** > **Up +Y**.
8. Use the **Rubberband Zoom** tool to view the following area:



The surfaces on the calliper and mount parts that meet with a small angle are shown.



9. In the *Merge/Imprint Options* group box, activate **Show Advanced Parameters** and set the following properties:

Property	Setting
Merge Angle	`20.0 deg`
Tolerance	`0.1 mm`

Setting the **Merge Angle** lower for this pairing prevents nodes and faces on non-contacting surfaces from being incorrectly projected.

10. Click **Imprint Pair**.

11. In the Vis toolbar, click (**Reset View**).

The merged surfaces between the calliper and mount appear highlighted in the *Geometry Scene 1* window.

12. In the *Merge/Imprint Tool* group box, set the following properties:

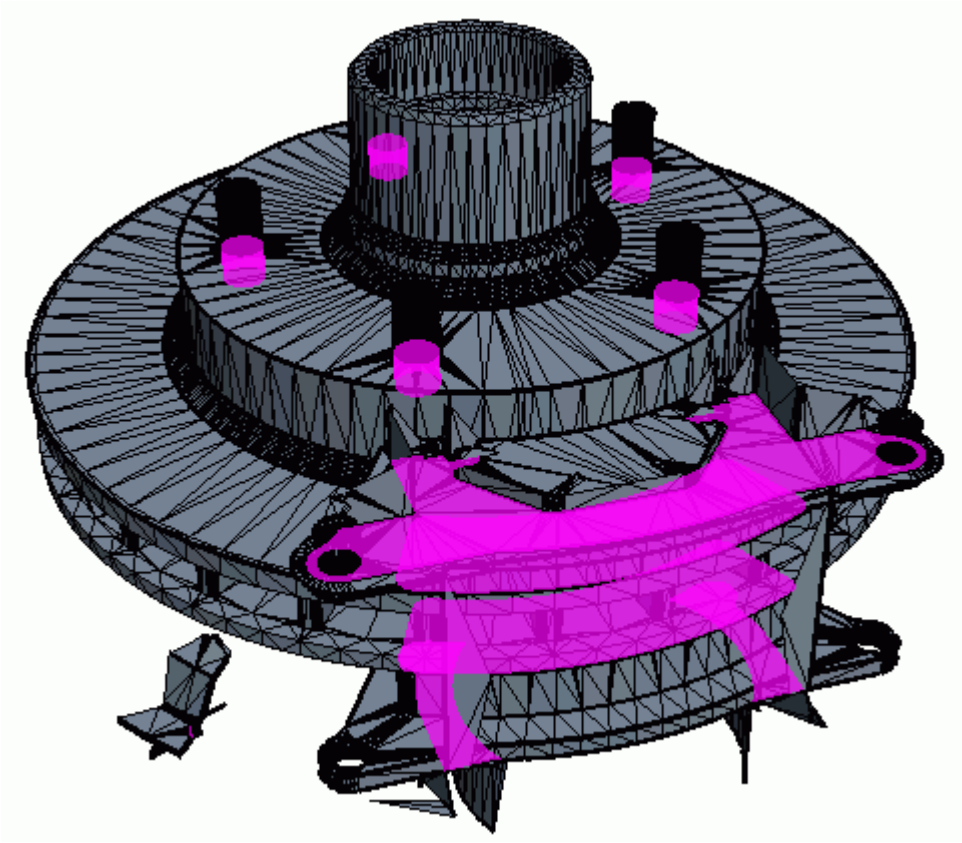
Property	Setting
Merge Angle	`45 deg`
Tolerance	`0.1 mm`

13. In the *Part Browser* box, click **Stop** then click **Find Pairs**.

Pairs with this specified **Merge Angle** are found.

14. Click **Imprint All**.

The surfaces of the remaining contacting pairs are imprinted and highlighted. All parts now connect conformally, and faces at the contact surfaces between parts are merged.



Only parts that were merged and imprinted are displayed in the *Geometry Scene 1* window.

Display all parts:

15. In the *Display Control* group box, click (**Show all faces**).

Remeshing the Surface

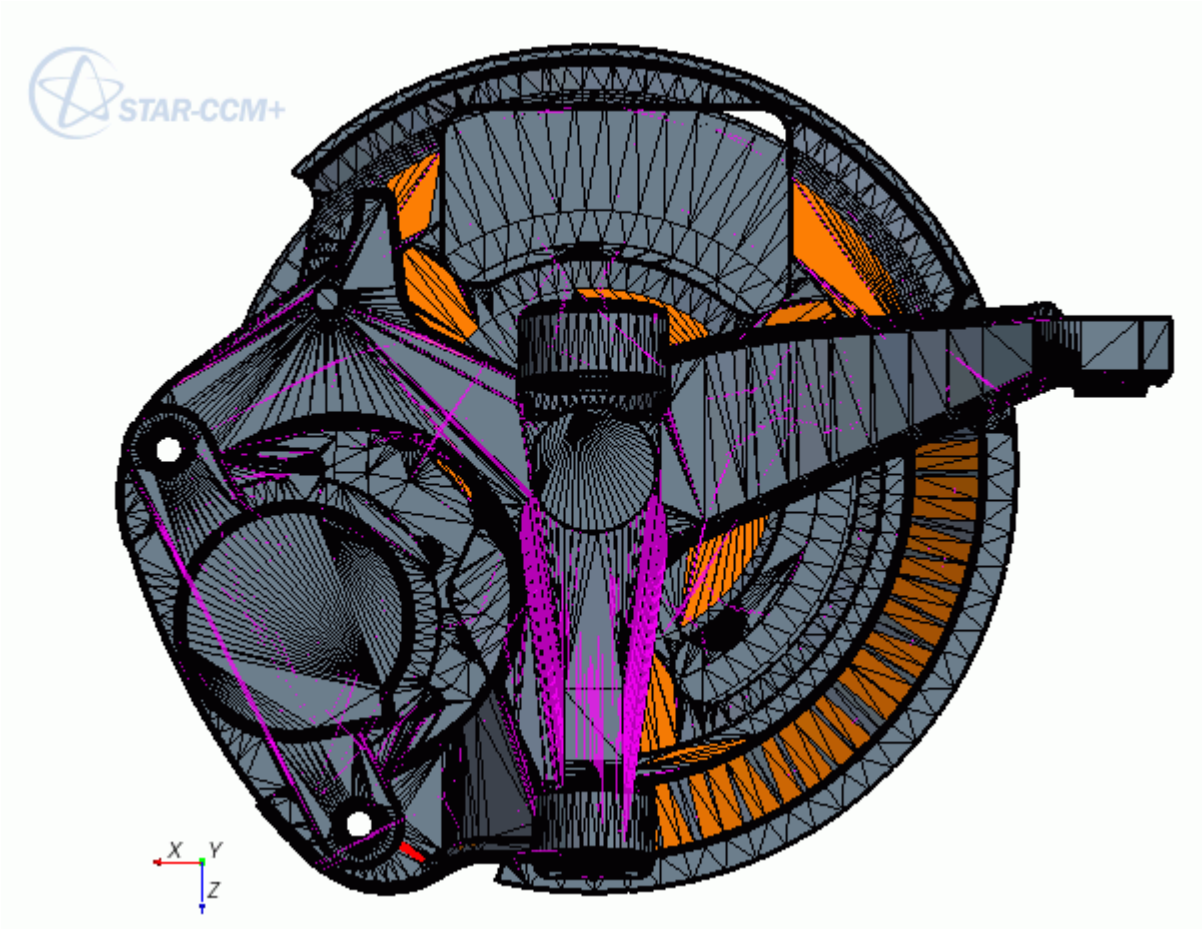
Produce a good quality surface mesh that is suitable for volume meshing.

It is important to check for any further issues that require surface repair and produce a good quality surface mesh so that any surface errors are not projected through the volume mesh.

Check for any surfaces that possibly still require surface repair:

1. Make the **Repair** tab active.
2. In the surface diagnostics dialog, click **Manage**.
3. Make sure that all thresholds are activated.
4. Click **OK**.

   There are still some issues that are displayed in the surface diagnostics dialog.
5. For face quality, click the yellow **Count** button.
6. In the *Vis* toolbar, click (**Save-Restore-Select Views**) and select **Views** > **-Y** > **Up -Z**.



In the Graphics window, poor quality faces are displayed, especially on the underside of the Brake Assembly. Most of the selected faces appear thin, with sharp edges.

Since the geometry still has a tessellated surface, this potential issue is likely to resolve itself when the surface is remeshed. So there is no need for further manual repair to fix this issue.

7. For face proximity, click the orange **Count** button.

   In the Graphics window, faces that are in close proximity to others appear selected. Most of the selected faces are on the heat shield part of the Brake Assembly. Faces that are in close proximity are a potential issue in some scenarios. However, in this case the heat shield part is intentionally close to the rest of the geometry. This potential issue is not a problem as long as a fine enough mesh is created.

8. Click **Close**.
9. Click the simulation tab.
10. Create a Surface Remesher Automated Mesh Operation:
    a) Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
    b) Select all parts and activate **Surface Remesher**.
    c) Click **OK**.
11. Edit the **Automated Mesh** > **Default Controls** and set the **Base Size** > **Value** to `0.01 m`.
12. Right-click the **Automated Mesh** node and select **Execute**.
13. View the remeshed surface:

a) Right-click a blank space in the *Graphics* window and select **Apply Representation** > **Automated Mesh.Remesh**.

b) In the *Vis* toolbar, click (**Show All Meshes**).

The underside of the Brake Assembly no longer has any thin or sharp faces as the tessellated surface has been replaced with a fine surface mesh.

Checking Surface Mesh Quality in Manual Repair Mode

Check the surface mesh quality in more detail.

Visual inspection of the remeshed surface shows that it is a good quality surface mesh. Since the quality of the surface mesh has a direct impact on the quality of the volume mesh, it is important to ensure that the surface mesh is free from errors.

Review the remeshed surface in the manual surface repair mode.

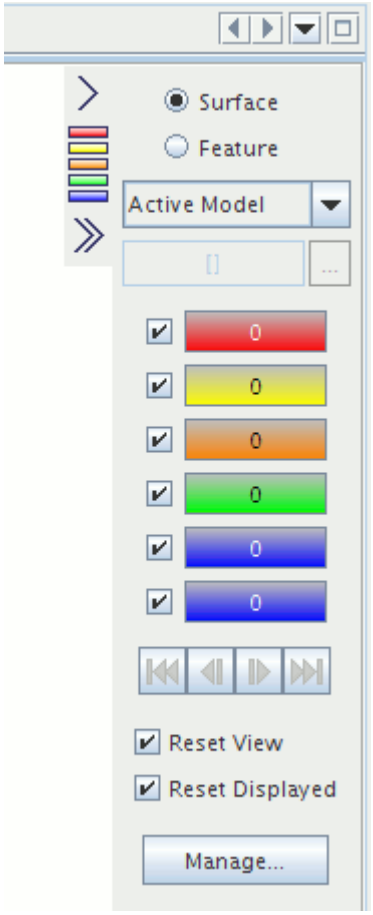
1. In the *Mesh Generation* toolbar, click (**Launch Surface Repair**).
2. In the *Surface Repair Options* dialog, set the following properties:

Property	Value
Geometry Description	**Automated Mesh.Remesh**
Geometry Parts	All Parts

3. In the *Surface Repair Options* dialog, click **OK**.

   The *Edit* window appears.
4. At the top of the *Edit* window, make sure that the **Repair** tab is active.
5. In the surface diagnostics dialog, click **Manage**.
6. Activate all Thresholds then click **OK**.

   The surface diagnostics dialog reports that none of the remeshed faces fail the diagnostic checks.



You now have a geometry with a good quality surface mesh which is ready to use with the STAR-CCM+ volume meshers.

7. Save the simulation.

Summary

This tutorial used a brake assembly to demonstrate some of the typical steps that are involved in geometry preparation and repair using STAR-CCM+.

The steps covered were:

- Using surface diagnostics
- Removing duplicate parts
- Repairing features
- Inflating thin surfaces
- Filling holes
- Carrying out boolean operations on geometry
- Merging and imprinting parts
- Remeshing the surface
- Checking surface mesh quality

- Closing gaps

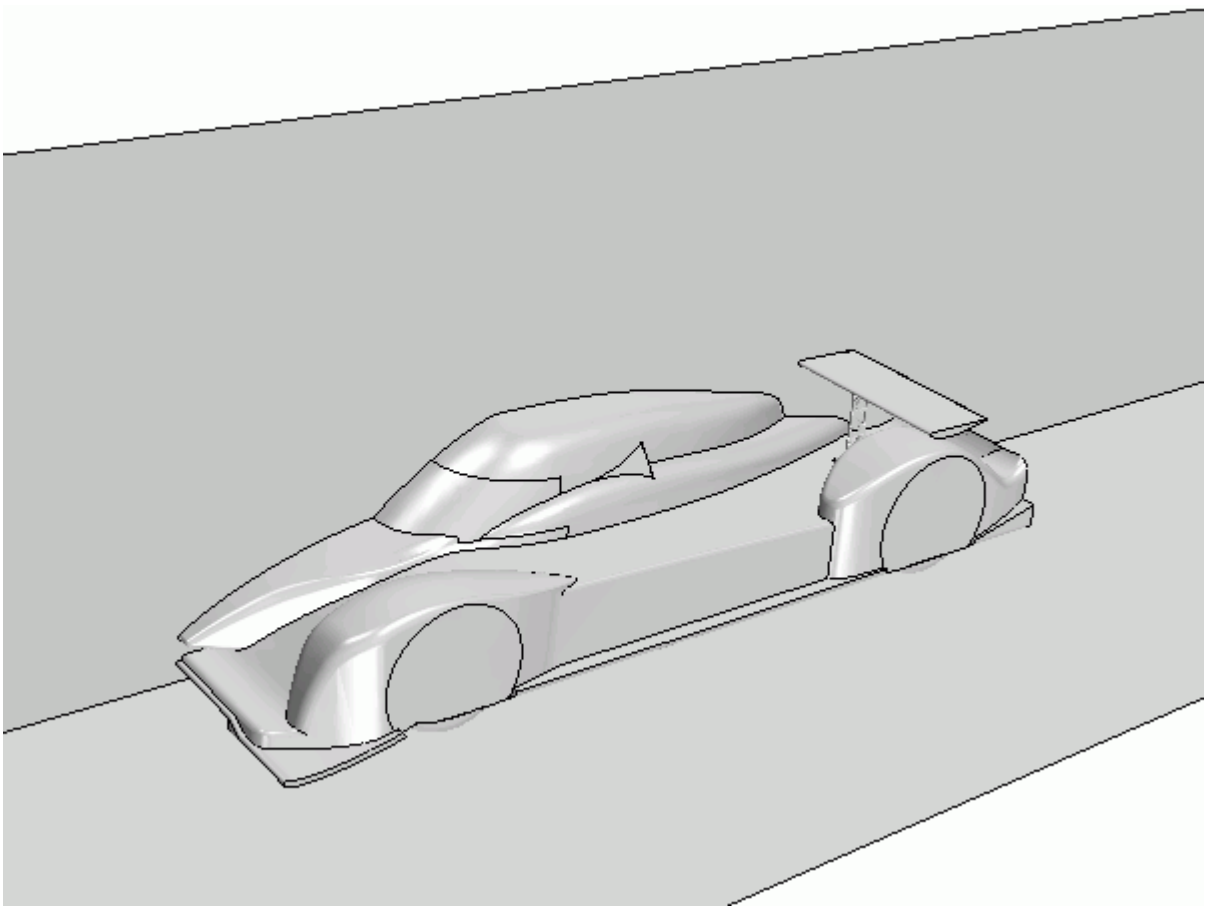
Parts-Based Meshing External Aerodynamics

This tutorial demonstrates how to set up a simulation using the parts-based meshing approach.

Parts-based meshing is a method for generating surface and volume meshes on geometry parts. It is different to region-based meshing in that it uses a series of mesh operations to define the process from the initial geometry to the final volume mesh. Having a series of operations allows you to modify the initial geometry and repeat the entire process with no additional inputs. This makes parts-based meshing particularly suitable for parametric design studies.

In this tutorial, you generate a volume mesh that is suitable for an external aerodynamics analysis of a car using the parts-based approach. Each step in the tutorial highlights some of the differences between parts-based meshing and region-based meshing. As part of the tutorial, you replace the rear wing on the prototype class racing car. Local surface remeshing only remeshes surfaces in the predefined area of the fluid domain, reducing the total amount of time needed to remesh the geometry.

The following image shows the geometry:



Contents:

Prerequisites

Overview
Importing the Surface Mesh
Storing the View
Preparing the Initial Surface
Filling a Hole on the Car Surface
Creating a Composite Assembly
Generating a Wrapped Surface Mesh
Creating the Fluid Volume
Assigning the Fluid Volume to a Region
Setting Boundary Types
Generating the Volume Mesh
Replacing the Wing and Executing the Pipeline
Summary

Prerequisites

The instructions in the Parts-Based Meshing External Aerodynamics tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Overview

There are several stages within the Parts-Based Meshing External Aerodynamics tutorial.

The tutorial stages are:

1. Importing and visualizing the CAD parts.

   Begin by launching STAR-CCM+ and importing CAD files that contain surface meshes for the car, wing, and bounding box. Then, set up the view and hide some surfaces so that you can examine the vehicle.
2. Preparing the initial surface.

   Remove the original wing from the car and import a new wing that is in a different position. This action creates several problems on the geometry surface, which you fix at later stages of the tutorial.
3. Filling a hole on the car surface.

   Use the fill hole mesh operation to fill the hole that you introduced when you removed the original wing. This step demonstrates the process for creating a mesh operation.
4. Creating a composite assembly.

   One of the advantages of parts-based meshing is that you can perform mesh operations on an assembly of several parts. Create an assembly to contain the parts that you are using in this tutorial.

5. Generating a wrapped surface mesh.

   At this stage you generate a surface wrap of the Vehicle Assembly, which fixes the intersections in the surface mesh. You also set up curve controls and contact prevention to maintain important features on the surface mesh.
6. Creating the fluid volume.

   The fluid simulation requires a volume mesh between the surface of the car and the walls of the bounding box. To create the part that defines the outer surface of the volume mesh, you subtract the output of the surface wrapper operation from the bounding box.
7. Assigning the fluid volume to a region.

   Before generating a volume mesh, you must assign the fluid volume to a region. This action allows the mesher to convert the volume mesh to a finite volume representation that the solvers can use.
8. Setting boundary types.

   Certain boundary types affect the volume mesh. In this tutorial, you have a velocity inlet, pressure outlet, and symmetry boundaries that do not require prism layers.
9. Generating the volume mesh.

   Use an automated mesh operation to generate the volume mesh.
10. Replacing the wing and executing the pipeline.

    An advantage of parts-based meshing is that when you have defined the series of mesh operations, you can repeat them on different initial geometries. Here, you replace the wing component and run the meshing pipeline to generate a new volume mesh.

Importing the Surface Mesh

Launch a simulation and import the supplied surface mesh.

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Surface Mesh**.
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/mesh/data`.
5. Select `pbm_car.x_b` and `pbm_boundingBox.x_b` then click **Open**.
6. Set *Tessellation Density* to **Very Fine**.
7. To accept the remaining default *Import Surface Options*, click **OK**.
8. Save the simulation as `pbm_externalAero.sim`.

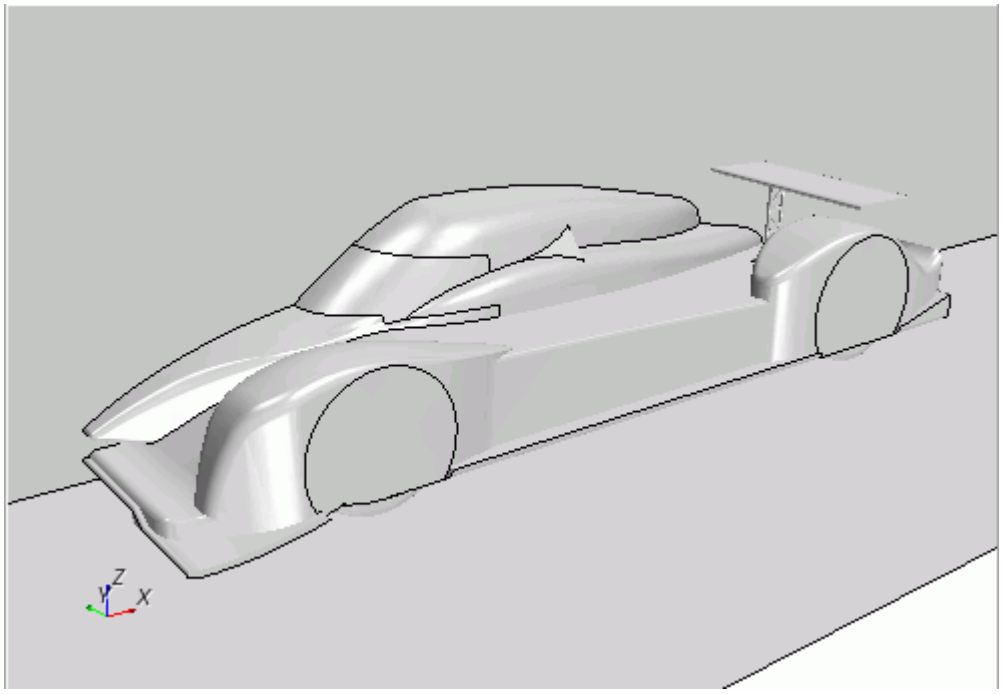
Storing the View

Set up and store a convenient view of the geometry for use throughout the tutorial.

To visualize the geometry:

1. Click the *Scene/Plot* tab.
2. Expand the **Displayers** > **Geometry 1** > **Parts** node.

3. Select the **Bounding Box.Far Field** node and deactivate the **Visible** option.
4. Set the view as shown below:



5. To save this view for later use, in the *Vis* toolbar, click (**Save-Restore-Select views**) and select **Store Current View**.
6. Click the *Simulation* tab.
7. Save the simulation.

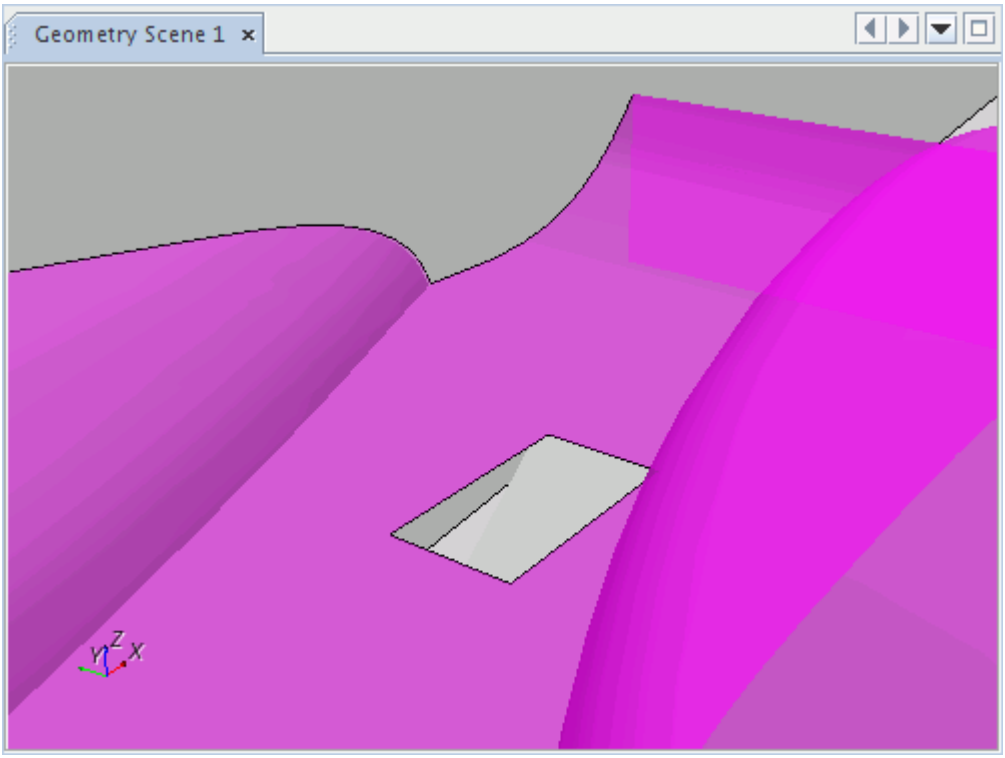
Preparing the Initial Surface

In this stage of the tutorial, you remove the original wing from the car and import a new wing that is in a different position. These steps create several problems on the geometry surface, which you fix at later stages of the tutorial.

To delete the original wing:

1. Expand the **Geometry** > **Parts** > **Car** > **Surfaces** node.
2. Delete the **Strut** and **Wing** part surfaces.

   Deleting the original wing leaves a hole in the surface mesh.



You fix this hole using the fill holes operation in a later stage of this tutorial.

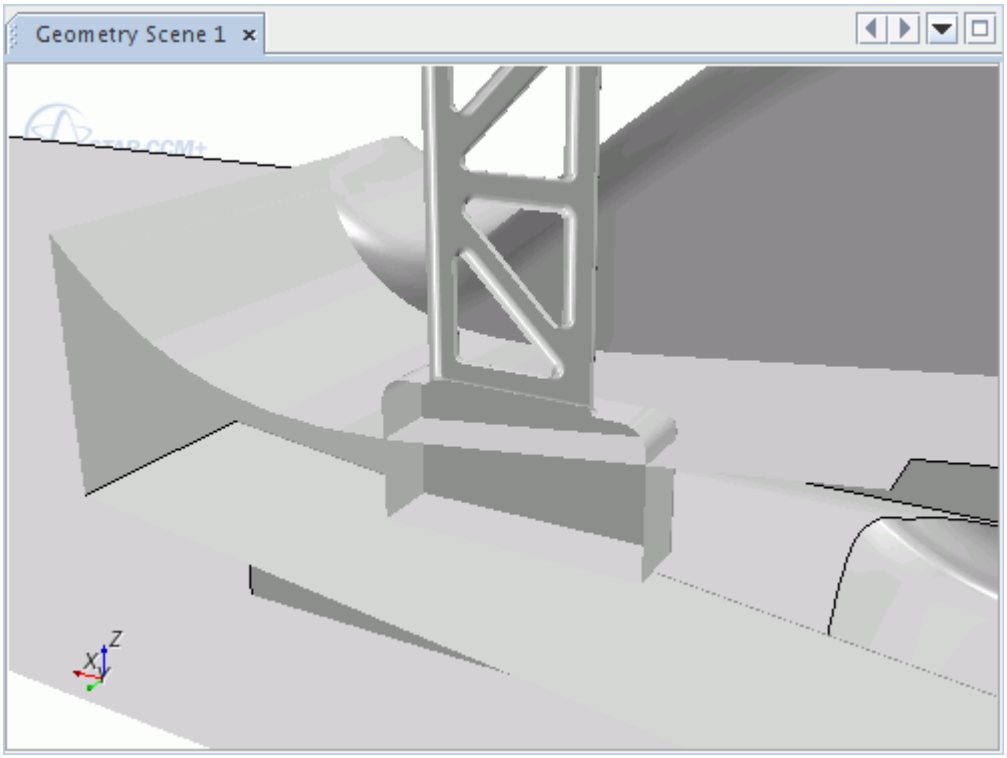
To import the new wing:

3. From the menu bar, select **File** > **Import** > **Import Surface Mesh**.
4. In the *Open* dialog, select `pbm_wing1.x_b` from the `[INSTALL_DIR]/doc/startutorialsdata/mesh/data` directory and click **Open**.
5. In the *Import Surface Options* dialog, deactivate the **Open Geometry Scene After Import** option and click **OK**.

To add the imported wing to the *Graphics* window:

6. Drag and drop the **Geometry** > **Parts** > **Wing** node onto *Geometry Scene 1*.
7. In the **Part Displayers** menu, select **Add to Geometry 1**.
8. Repeat this process and add the **Wing** > **Curves** > **Edges** node to *Geometry Scene 1*.
9. Save the simulation.

The new wing intersects the car body surface, resulting in pierced faces on the surface. You fix the pierced faces by wrapping the external surface of the entire vehicle at a later stage of this tutorial.



Filling a Hole on the Car Surface

Use the fill hole mesh operation to fill the hole in the surface mesh that appeared when you removed the initial wing.

To fill the hole on the car surface:

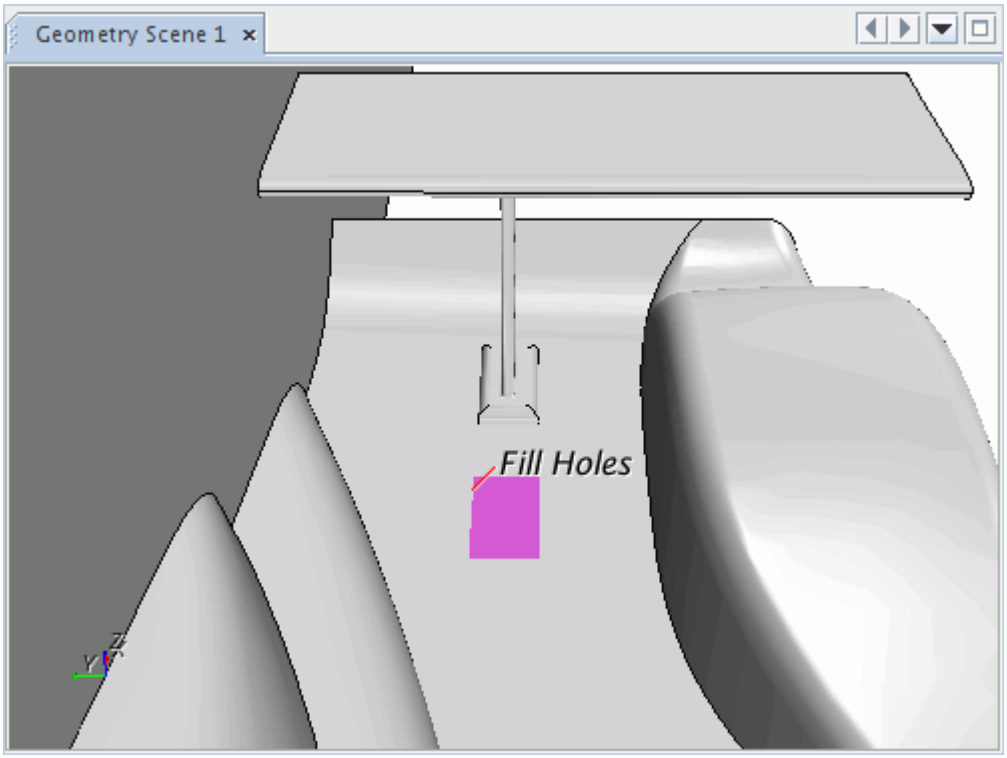
1. Right-click the **Geometry** > **Operations** node and select **New** > **Surface Preparation** > **Fill Holes**.
2. In the *Create Fill Holes Operation* dialog, set *Input Parts* to **Car** and click **OK**.

   Two **Fill Holes** nodes appear in their pre-executed state in the simulation tree:

   - **Geometry** > **Parts** > **Fill Holes**.
   - **Geometry** > **Operations** > **Fill Holes**.
3. Select the **Operations** > **Fill Holes** node and set *End Surfaces* to **Main Body**.
4. Right-click the **Operation** > **Fill Holes** node and select **Execute**.

   STAR-CCM+ performs the fill holes operation. This operation causes both **Fill Holes** nodes to be displayed in their executed state.
5. Drag and drop the **Parts** > **Fill Holes** node onto *Geometry Scene 1* and select **Part Displayers** > **Add to Geometry 1**.

The fill holes operation has repaired the car surface.



Creating a Composite Assembly

One of the advantages of parts-based meshing is that you can perform mesh operations on an assembly of several parts. Here, you create an assembly to contain the parts that you are using in this tutorial.

To create a composite assembly:

1. Expand the **Geometry** > **Parts** node and multi-select the **Car**, **Fill Holes**, and **Wing** nodes.
2. Right-click one of the selected nodes and select **Composite**.

   STAR-CCM+ creates a composite node and places the original part nodes within it.
3. Rename the **Composite** node to `Vehicle Assembly`.
4. Save the simulation.

Generating a Wrapped Surface Mesh

At this stage you generate a surface wrap of the Vehicle Assembly to fix the pierced faces. You also set up curve controls and contact prevention to maintain important features on the surface mesh.

The steps to generate a wrapped surface in this tutorial are:

1. Create a Surface Wrapper operation and define its properties. See Creating and Setting Up the Surface Wrapper.
2. Apply mesh refinement to part curves. See Refining the Mesh on Part Curves.

3. Apply contact prevention controls to maintain separation between surfaces on the nose and on the main body. See Preventing Contact between Close Surfaces.
4. Execute the Surface Wrapper operation. See Executing the Surface Wrapper Operation.

Contents:

Creating and Setting Up the Surface Wrapper
Refining the Mesh on Part Curves
Preventing Contact between Close Surfaces
Executing the Surface Wrapper Operation

Creating and Setting Up the Surface Wrapper

Create a surface wrapper operation and set the properties that apply to all geometry parts on which it operates.

To set up the surface wrapper mesh operation:

1. Right-click the **Geometry** > **Operations** node and select **New** > **Surface Preparation** > **Surface Wrapper**.
2. In the *Create Surface Wrapper Auto Mesh Operation* dialog, Select **Vehicle Assembly**.
3. Click OK.

   Two **Surface Wrapper** nodes appear in their pre-executed state in the simulation tree:
   - **Geometry** > **Parts** > **Surface Wrapper**.
   - **Geometry** > **Operations** > **Surface Wrapper**.
4. Expand the **Operations** > **Surface Wrapper** > **Default Controls** node and set the following properties:

Node	Property	Setting
Base Size	*Value*	`0.02 m`
Minimum Surface Size	*Percentage of Base*	`5.0`
Volume of Interest	*Method*	**External**

   Setting the Volume of Interest method to External instructs the surface wrapper to ignore any internal surfaces within the Vehicle Assembly part. This fixes the pierced faces at the intersection between the wing and the vehicle surface.

Refining the Mesh on Part Curves

To make sure that you capture the detailed features in the surface mesh, apply mesh refinement to all the part curves.

To apply refinement to all part curves:

1. Right-click the **Surface Wrapper** > **Custom Controls** node and select **New** > **Curve Control**.

   STAR-CCM+ adds a curve control node within the **Custom Controls** node.
2. Edit the **Custom Controls** > **Curve Control** node and do the following:

Node	Property	Setting
Curve Control	*Part Curves*	**Vehicle Assembly** > **Car** > **Curves** > **Car Edges**
		Vehicle Assembly > **Car** > **Curves** > **Edges**
		Vehicle Assembly > **Car** > **Curves** > **Wing Edges**
		Vehicle Assembly > **Wing** > **Curves** > **Edges**
└ **Controls** > **Target Surface Size**	*Target Surface Size*	**Custom**
└ **Values** > **Target Surface Size**	*Percentage of Base*	`5.0`
└ **Controls** > **Minimum Surface Size**	*Minimum Surface*	**Custom**
└ **Values** > **Minimum Surface Size**	*Percentage of Base*	`2.5`

Preventing Contact between Close Surfaces

To make sure that these two surfaces do not touch each other in the wrapped surface, apply a contact prevention between the nose and main body.

To apply contact prevention:

1. Right-click the **Operations** > **Surface Wrapper** > **Contact Prevention** node and select **New** > **One Group Contact Prevention Set**.
2. Rename the **Contact Prevention** node to `Nose - Main Body`.
3. Select the **Contact Prevention** > **Nose - Main Body** node and set the following properties:

Property	Setting
Surfaces	**Vehicle Assembly** > **Car** > **Surfaces** .
Minimum Size	`1 mm`

Executing the Surface Wrapper Operation

Execute the operation to generate a wrapped surface.

To execute and visualize the surface wrapper operation:

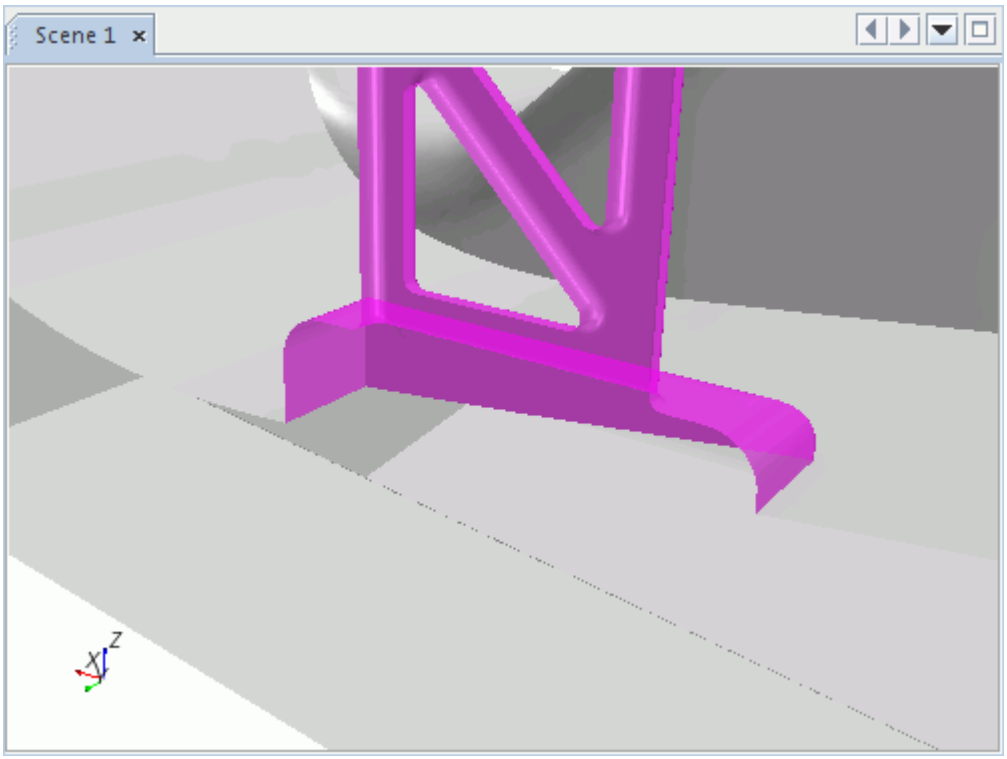
1. Right-click the **Geometry** > **Operations** > **Surface Wrapper** node and select **Execute**.

   STAR-CCM+ performs the surface wrap operation. The result of the operation is that both **Surface Wrapper** nodes beneath the **Geometry** > **Parts** node and the **Geometry** > **Operations** node, are displayed in their executed state.

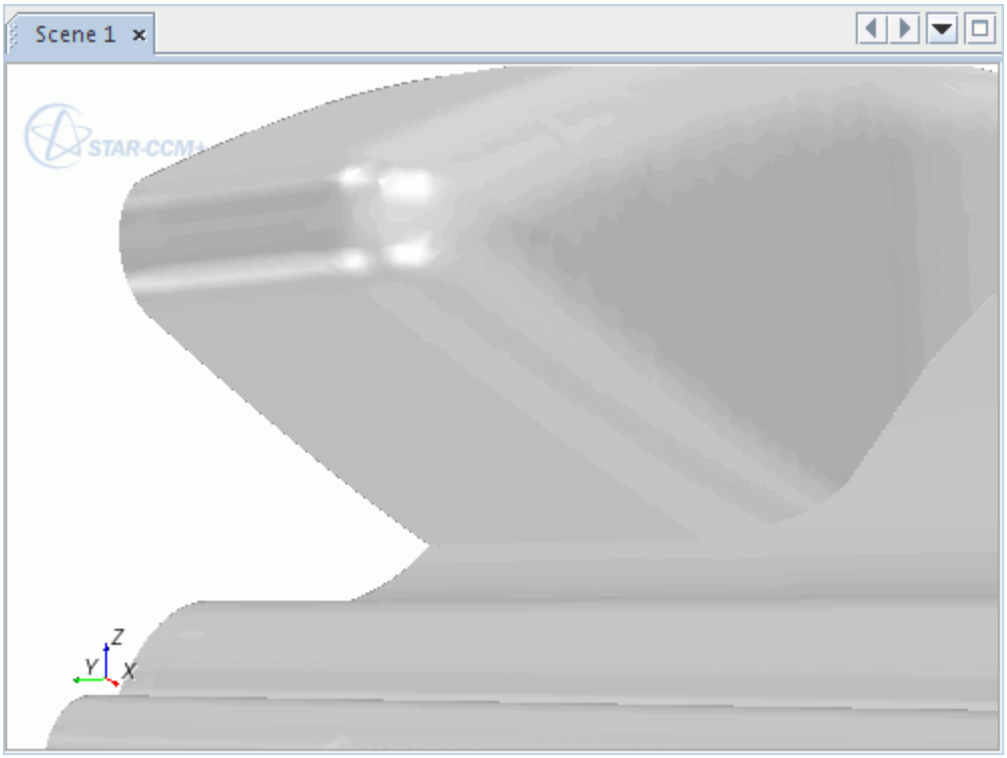
2. When the operation is complete, create an empty scene.
3. Drag and drop the **Geometry** > **Parts** > **Surface Wrapper** node onto *Scene 1*.
4. In the *Vis* toolbar, click (**Save-Restore-Select views**) and select **Restore View** > **View 1**.
5. Save the simulation.

Visually inspecting the surface shows that the wrapper has fixed the surface intersection issues and prevented contact between close surfaces:

Surface Intersections:



Nose and Main Body Contact Prevention:



Creating the Fluid Volume

The fluid simulation requires a volume mesh between the surface of the car and the walls of the bounding box. To create the part that defines the outer surface of the volume mesh, you subtract the output of the surface wrapper operation from the bounding box.

This tutorial uses the subtract operation instead of the extract volume operation. The reason for using the subtract operation is that the car tires (contained within the Surface Wrapper part) intersect the surface of the Bounding Box part. The extract volume operation cannot intersect surfaces; the subtract operation does.

To create the fluid volume:

1. Right-click the **Geometry** > **Operations** node and select **New** > **Boolean** > **Subtract**.
2. In the *Create Subtract Operation* dialog, set the following properties:

Property	Setting
Input Parts	**Bounding Box**
	Surface Wrapper
Target Part	**Bounding Box**

3. Click **OK**.

   A **Subtract** node appears beneath the **Operations** node, and a **Subtract** part appears beneath the **Parts** node.
4. Rename the **Parts** > **Subtract** node to `Fluid Volume`.
5. Right-click the **Operations** > **Subtract** node and select **Execute**.

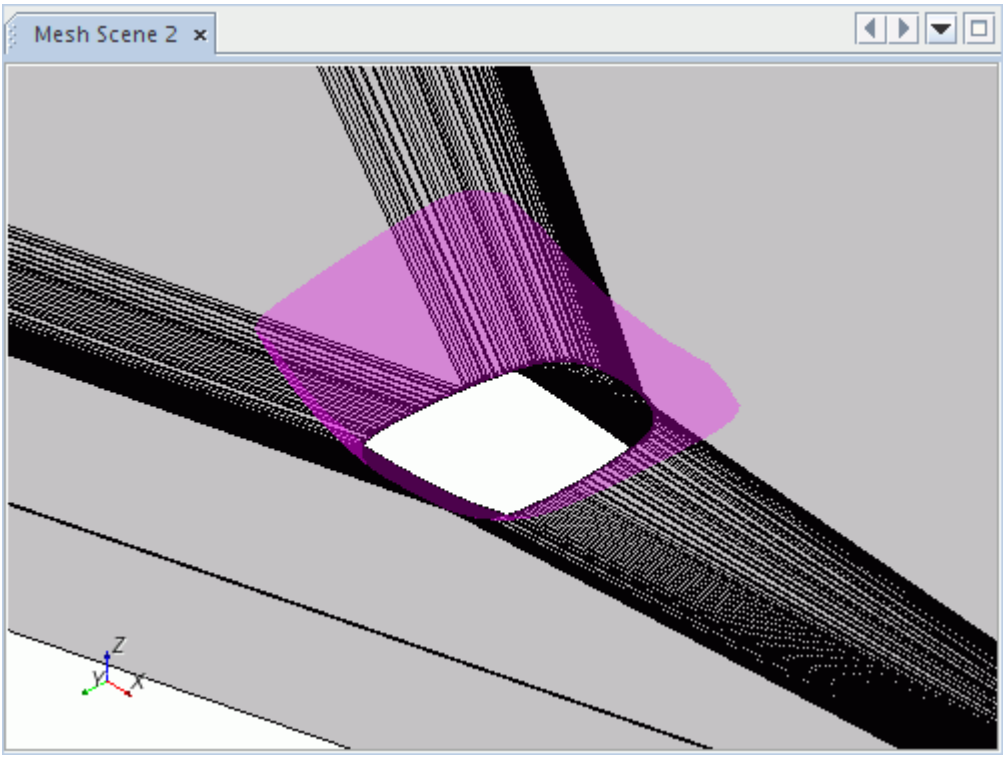
To view the fluid volume:

6. In the *Vis* toolbar, click **Create/Open Scenes** and select **Mesh**.
7. Edit the *Mesh* scene and set the following properties:

Node	Property	Setting
Displayers > **Mesh 1**	**Outline**	activated
Parts	**Parts**	**Fluid Volume**

8. In the *Vis* toolbar, click (**Save-Restore-Select views**) and select **Restore View** > **View 1**.
9. Select the **Displayers** > **Mesh 1** > **Parts** > **Fluid Volume.Bounding Box.Far Field** node and deactivate the **Visible** option.
10. In the *Vis* toolbar, click **Hide All Meshes**.
11. Save the simulation.

    On the under-side of the Fluid Volume part, the subtract operation has intersected and removed the surface that represents the tire contact patch. The extract volume mesh operation is not capable of making these geometric changes to a surface.



Assigning the Fluid Volume to a Region

Before generating a volume mesh, you must assign the fluid volume to a region. This action allows the mesher to convert the volume mesh to a finite volume representation that the solvers can use.

To assign the fluid volume to a region:

1. Right-click the **Parts** > **Fluid Volume** node and select **Assign Parts to Regions**.
2. In the *Assign Parts to Regions* dialog, select the following options:
   - **Create a Region for Each Part**
   - **Create a Boundary for Each Part Surface**
3. Click **Apply** and then **Close**.

Setting Boundary Types

Certain boundary types affect the volume mesh. In this tutorial, you have a velocity inlet, pressure outlet, and symmetry boundaries that do not require prism layers. STAR-CCM+ does not grow prism layers on these boundaries by default .

To set the boundary types:

1. Expand the **Regions** > **Fluid Volume** > **Boundaries** node.
2. Set the following properties:

Node	Property	Setting
Bounding box.Far Field	*Type*	**Symmetry Plane**
Bounding Box.Symmetry Plane	*Type*	**Symmetry Plane**
Bounding Box.Inlet	*Type*	**Velocity Inlet**
Bounding Box.Outlet	*Type*	**Pressure Outlet**

Generating the Volume Mesh

Use an automated mesh operation to generate the volume mesh.

The steps required to generate a volume mesh are:

1. Create an automated mesh operation. See Creating the Automated Mesh Operation.
2. Refine the mesh for the vehicle assembly surfaces. See Refining the Vehicle Assembly Surfaces.
3. Generate and view the volume mesh. See Generating and Viewing the Volume Mesh.

Contents:

Creating the Automated Mesh Operation
Refining the Vehicle Assembly Surfaces
Generating and Viewing the Volume Mesh

Creating the Automated Mesh Operation

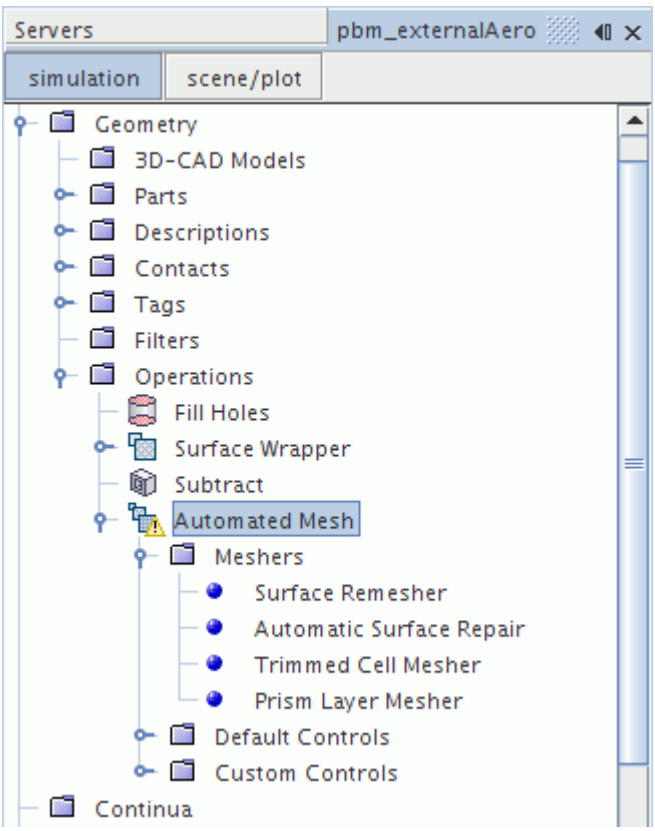
An automated mesh operation determines the meshing models and controls for generating the surface and volume meshes.

To create the automated mesh operation:

1. Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
2. In the *Create Automated Mesh Operation* dialog, select the **Parts** > **Fluid Volume part**.
3. Activate the following meshing models:

Group box	Model
Surface Meshers	**Surface Remesher**
Optional Surface Meshers	**Automatic Surface Repair**
Core Volume Meshers	**Trimmed Cell Mesher**
Optional Boundary Layer Meshers	**Prism Layer Mesher**

4. Click **OK**.
5. Expand the **Automated Mesh** > **Meshers** node to see the models.



To set the sizes and controls for the operation:

6. Expand the **Operations** > **Automated Mesh** > **Default Controls** node.
7. Set the following properties:

Node	Property	Setting
Base Size	*Value*	`0.02 m`
Number of Prism Layers	*Number of Prism Layers*	`5`

Node	Property	Setting
Volume Growth Rate	*Surface Growth Rate*	**Fast**

Refining the Vehicle Assembly Surfaces

To capture the vehicle surface accurately, refine the surface mesh. This refinement also affects the volume mesh as the trimmed mesher aims to produce cells that match the surface size.

To refine the mesh that surrounds the Wrapped Vehicle surfaces:

1. Right-click the **Operation** > **Automated Mesh** > **Custom Controls** node and select **New** > **Surface Control**.
2. Rename the **Custom Controls** > **Surface Control** node to `Bounding Box`.
3. Select the **Bounding Box** node and set **Part Surfaces** to **Fluid Volume** > **Bounding Box**.
4. Expand the **Bounding Box** node and set the following properties:

Node	Property	Setting
Controls		
└ **Minimum Surface Size**	*Minimum Surface size*	**Specify Custom Value**
└ **Target Surface Size**	*Target Surface size*	**Specify Custom Value**
Values		
└ **Minimum Surface Size**	*Percentage of Base*	`100`
└ **Target Surface Size**	*Percentage of Base*	`500`

Increasing the minimum and target surface sizes for the bounding box that surrounds the vehicle assembly, avoids unnecessary refinement on the outer boundaries. The mesh in the bounding box remains refined near the vehicle assembly and grows coarse nearer the external boundaries. It is computationally less expensive to produce a coarse mesh in areas which do not require refinement.

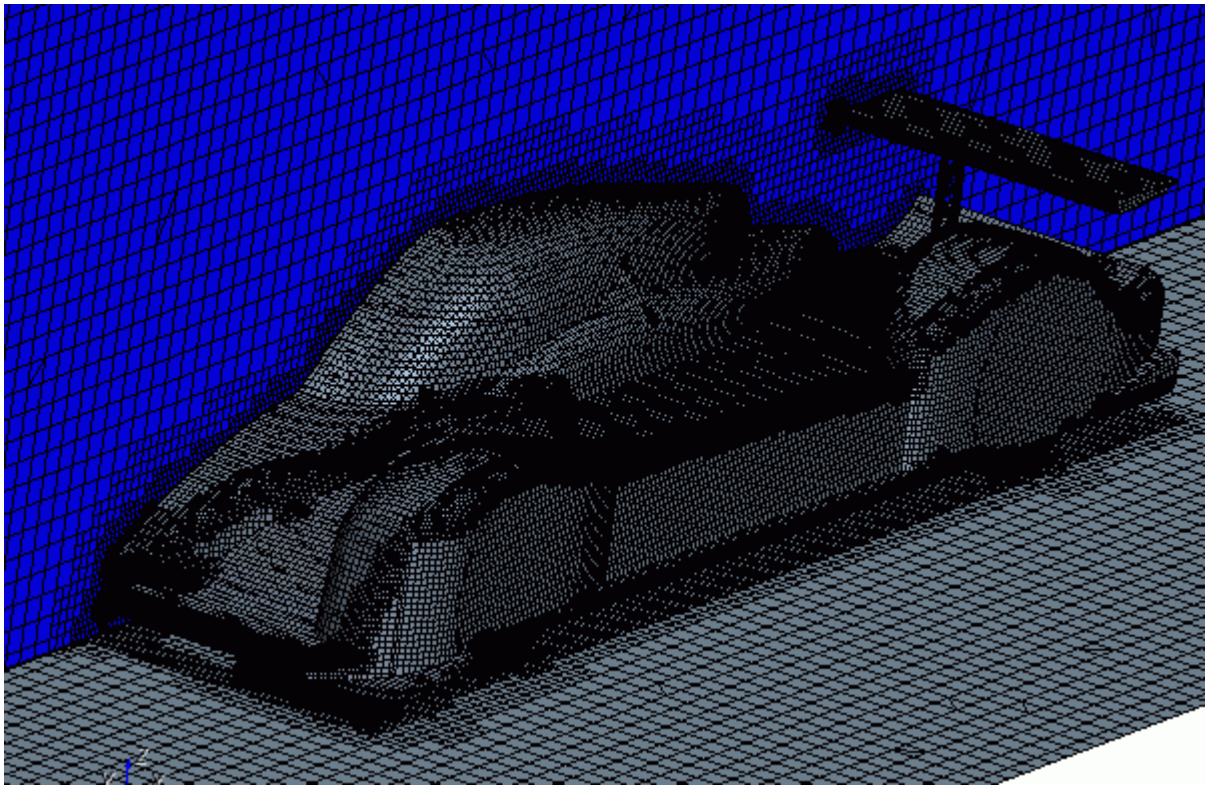
Generating and Viewing the Volume Mesh

Executing the automated mesh operation runs the surface and volume meshers. When the meshers complete, you can view the volume mesh in a mesh scene.

To generate and view the volume mesh:

1. Right-click the **Operations** > **Automated Mesh** node and select **Execute**.

   STAR-CCM+ generates the volume mesh. This operation takes several minutes to complete.
2. When the automated mesh operation is complete, display *Mesh Scene 1* in the *Graphics* window.
3. In the *Vis* toolbar, click (**Show All Meshes**).
4. Drag and drop the **Representations** > **Volume Mesh** node into *Mesh Scene 1*.



5. Save the simulation.

Replacing the Wing and Executing the Pipeline

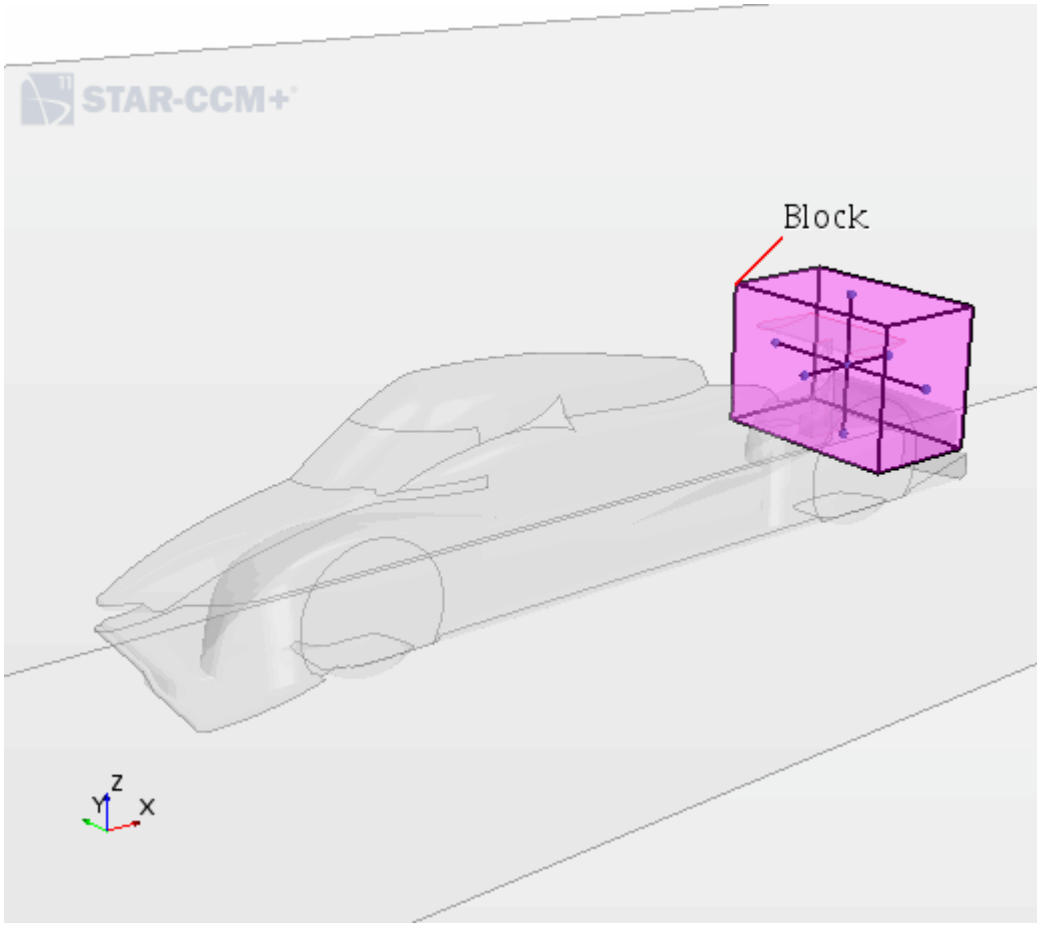
An advantage of parts-based meshing is that when you have defined the series of mesh operations, you can repeat them on different initial geometries.

In this tutorial, you replace the rear wing component and run the meshing pipeline to generate a new volume mesh. With local surface meshing active, STAR-CCM+ only remeshes surfaces in predefined areas of the fluid domain and uses the existing mesh elsewhere. Local surface remeshing can significantly reduce the amount of time required for remeshing geometries.

Create a part shape that encloses the rear wing component:

1. Right-click the **Geometry** > **Parts** node and select **New Shape Part** > **Block**.
2. In the *Create/Edit Block Part* dialog, enter the following properties:

Property	Setting
Corner 1	`[3.71, -1.05, 0.30] m`
Corner 2	`[4.33, 0.03, 1.06] m`



3. Click **OK**.

   A new block part, **Block**, is added to the **Parts** node. Every surface that is contained within this volume is remeshed locally. All surfaces outside this volume remain unchanged.
4. Rename the **Block** part to `Local Meshing`.

Activate local meshing and select the volume extent to define the volume that is meshed locally:

5. Select the **Geometry** > **Operations** > **Automated Mesh** node and set *Perform Local Surface Meshing* to **True**.

   The **Local Extents** node is added to the **Automated Mesh** node.
6. Right-click the **Local Extents** node and select **New** > **Volume Extent**.
7. Select the **Local Extents** > **Volume Extent** node and set the following properties:

Property	Setting
Parts	**Local Meshing**
Inflation Distance	`0.01 m`

   The inflation distance provides a transition zone that allows the mesh to change outside the specified part.

Swap the rear wing for a different design and remesh the geometry:

8. Expand the **Geometry** > **Parts** > **Vehicle Assembly** node.
9. Right-click the **Wing** node and select **Replace Part**.
10. In the *Open* dialog, navigate to the `[INSTALL_DIR]/doc/startutorialsdata/mesh/data` directory, select `pbm_wing2.x_b`, and click **Open**.

    Operations that require an update are marked with a yellow warning triangle.
11. Right-click the **Geometry** > **Operations** node and select **Execute All**.

    When complete, the *Graphics* window displays the updated volume mesh.
12. Save the simulation.

Summary

This tutorial demonstrated how to set up a simulation using Parts-Based Meshing.

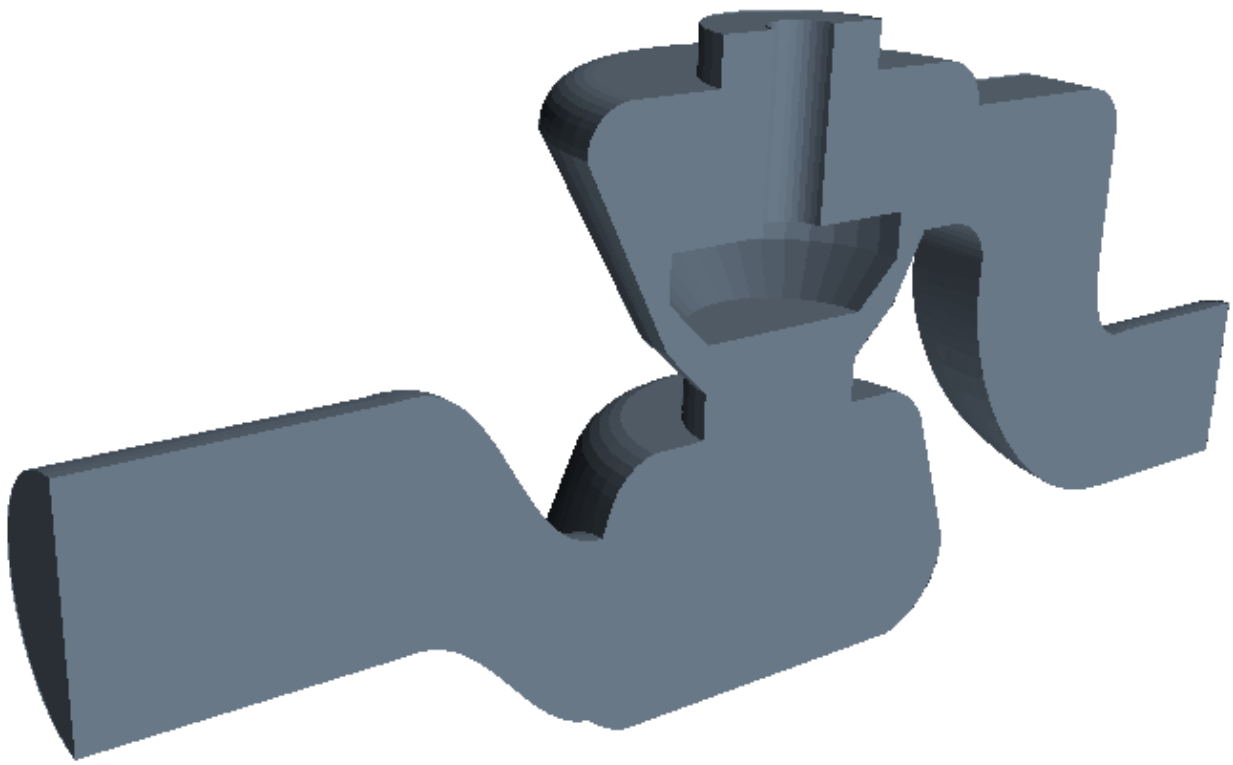
The steps covered were:

- The parts-based meshing procedure
- Using mesh operations
- Using the surface wrapper
- The relationship between parts and regions
- The effect of boundary types on prism layer settings
- Automated mesh operations
- Mesh controls
- The meshing pipeline

Mesh Operations: Imported Control Valve

This tutorial demonstrates how to use the surface and volume meshing operations within STAR-CCM+ to generate a polyhedral volume mesh with prism layers for a single region. The inputs required to build a trimmed cell mesh with prism layers are also covered.

Start from a geometry from an imported Parasolid file. The geometry is that of a control valve typically used for water flow management.



The control valve is in an almost closed position, with flow entering at the near end and exiting at the far end. You use Boolean and transform operations as part of the meshing process.

This tutorial is split into the following sections:

- Polyhedral meshing example
- Trimmed cell meshing example

Contents:

Prerequisites

Polyhedral Meshing Example
Trimmed Cell Meshing Example
Summary

Prerequisites

The instructions in the Mesh Operations Imported Control Valve tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Polyhedral Meshing Example

To set up the STAR-CCM+ simulation and generate a polyhedral volume mesh, launch a simulation and import the supplied surface mesh.

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Surface Mesh...**.
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/mesh/data`.
5. Select `Control_Valve.x_t` then click **Open**.
6. In the *Import Surface Options* dialog, make the following settings:
   - In the *Import Mode* group box, select **Create New Part**.
   - In the *Parasolid Transmit Import Options* group box:
     - To prevent automatic creation of contacts, deactivate **Create Part Contacts from Coincident Entities**.
     - Activate **Check Parts' Validity**.
     - Activate **Repair Invalid Bodies**.
7. Click **OK** to accept the remaining default *Import Surface Options*.
8. Save the simulation as `control_valve1.sim`.

Contents:

Repositioning the Valve and Extracting the Fluid Volume
Defining the Simulation Topology
Generating the Surface Mesh
Creating a Mesh Refinement
Generating the Volume Mesh
Examining Mesh Statistics with Histogram Plots

Repositioning the Valve and Extracting the Fluid Volume

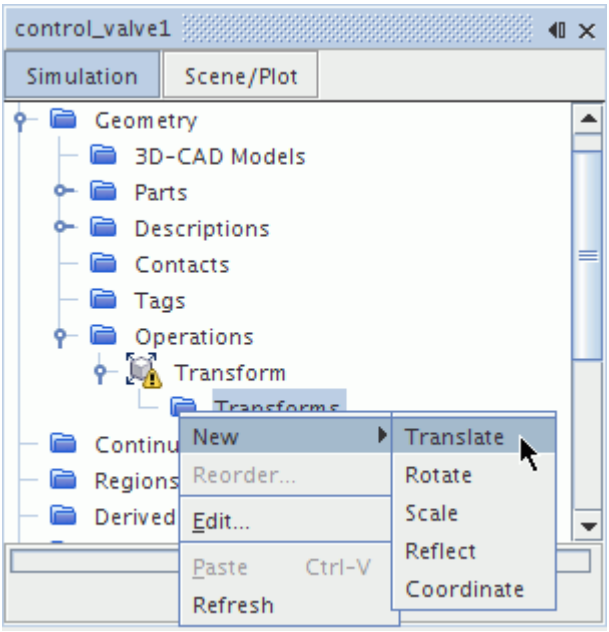
Use a transform operation to move the valve into its final location, and a subtract operation to define the fluid volume.

Set up a transform operation:

1. Right-click **Geometry** > **Parts** > **Valve** and select **Create Mesh Operation** > **Surface Preparation** > **Transform**.
2. In the object selector, click **OK** to accept the default selection.

To create a valve lift in the y-direction, use the mesh translation feature:

3. To create a **Translate** node, right-click the **Geometry** > **Operations** > **Transform** > **Transforms** node and select **New** > **Translate**.



4. Select the **Translate** node and set the *Translation Vector* property to `0, -5, 0 mm`.
5. Right-click the **Operations** > **Transform** node and select **Execute**.

   The position of the valve changes as you execute the change in this value.

Create a Boolean subtract, so that the housing is left over without the valve.

6. Right-click both part nodes under the **Parts** node and select **Create Mesh Operation** > **Boolean** > **Subtract**.
7. In the *Create Subtract Operation* dialog:
   a) Select the **Housing** and **Valve** parts as the *Input Parts*.
   b) Select the **Housing** part as the *Target Part*.
   c) Activate the **Execute Operation Upon Creation** option.
   d) Click **OK**.

   The **Subtract** node appears in the **Parts** node and the operation is executed automatically.

8. Save the simulation.

Defining the Simulation Topology

Create the region and assign the appropriate boundary types.

To create the region from the parts:

1. Right-click the **Parts** > **Subtract** node and select **Assign Parts to Regions**.
2. In the *Assign Parts to Regions* dialog, select:
   a) **Create a Region for Each Part**
   b) **Create a Boundary for Each Part Surface**

   Leave all other settings at their default value.
3. Click **Apply** then **Close**.
   The region and boundaries are generated.
4. Rename the **Regions** > **Subtract** node to `control_valve`.

Setting the boundary types at this stage is important as it affects which boundaries have prism layers next to them. Only wall boundaries have prism layers generated next to them by default. To set the boundary types:

5. Expand the **Regions** > **control_valve** > **Boundaries** node and set the following properties:

Node	Property	Setting
Housing.Faces	*Type*	**Wall**
Housing.Inlet	*Type*	**Mass Flow Inlet**
Housing.Outlet	*Type*	**Pressure Outlet**
Valve.Faces	*Type*	**Wall**

Generating the Surface Mesh

Use mesh operations to generate the surface mesh.

For this example, since you are selecting the surface remesher and volume meshers in tandem, you could proceed and generate both meshes. In this tutorial, however, first generate the surface mesh and view it before proceeding with the volume mesh.

In the automated mesh operation, you choose the surface remesher, polyhedral volume mesher, and the prism layer mesher.

1. Right-click the **Geometry** > **Parts** > **Subtract** node and select **Create Mesh Operation** > **Mesh** > **Automated Mesh**.
2. In the *Create Automated Mesh Operation* dialog, select the following meshers, in order:

Group	Mesher
Surface Meshers	**Surface Remesher**
Core Volume Meshers	**Polyhedral Mesher**

Group	Mesher
Optional Boundary Layer Meshers	**Prism Layer Mesher**

3. Click **OK**.
4. Select the **Geometry** > **Operations** > **Automated Mesh** > **Meshers** > **Prism Layer Mesher** node and set the *Near Core Layer Aspect Ratio* expert property to `0.5`.
5. Edit the **Automated Mesh** > **Default Controls** node.
6. Set the following properties:

Node	Property	Setting
Base Size	*Value*	`5.0 mm`
Number of Prism Layers	*Number of Prism Layers*	`4`
Prism Layer Total Thickness	*Percentage of Base*	`50.0`

7. Right-click the **Automated Mesh** node and select **Execute Surface Meshers**.
8. While the surface mesh is being generated, create a mesh scene.

   The scene appears when the mesh generation process is finished.
9. Save the simulation.

Creating a Mesh Refinement

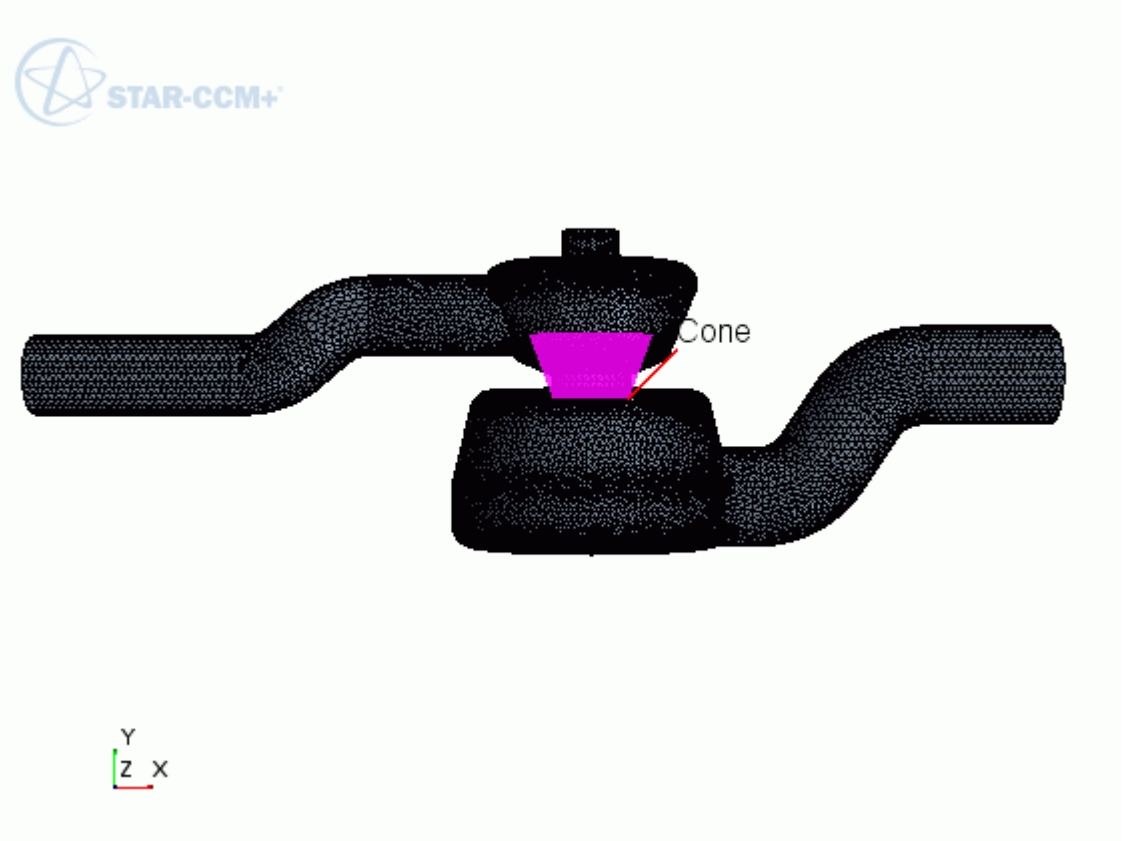
Create a refinement zone for the valve-base portion of the mesh.

To define the refinement zone, create a cone part.

1. Right-click the **Geometry** > **Parts** node and select **New Shape Part** > **Cone**.
2. In the *Create/Edit Cone Part* dialog, set the following properties:

	Start Circular Disc	End Circular Disc	Start Radius	End Radius
X	`0.0 mm`	`0.0 mm`	`25.0 mm`	`40.0 mm`
Y	`-14.268 mm`	`25.887 mm`		
Z	`0.0 mm`	`0.0 mm`		

3. Click **Create** then **Close**.
4. To view the location of the new cone part, make the mesh scene active and select the **Geometry** > **Parts** > **Cone** node.



Set up the volumetric control.

5. Right-click the **Geometry** > **Operations** > **Automated Mesh** > **Custom Controls** node and select **New** > **Volumetric Control**.
6. Edit the **Custom Controls** > **Volumetric Control** node and set the following properties:

Node	Property	Setting
Volumetric Control	*Parts*	**Cone**
└ **Controls**		
└ **Surface Remesher**	*Customize Size*	**Activated**
└ **Polyhedral Mesher**	*Customize Polyhedral Mesher*	**Activated**
└ **Values**		
└ **Custom Size**	*Percentage of Base*	40.0

7. Save the simulation.

Generating the Volume Mesh

Generate and view the polyhedral mesh along a plane.

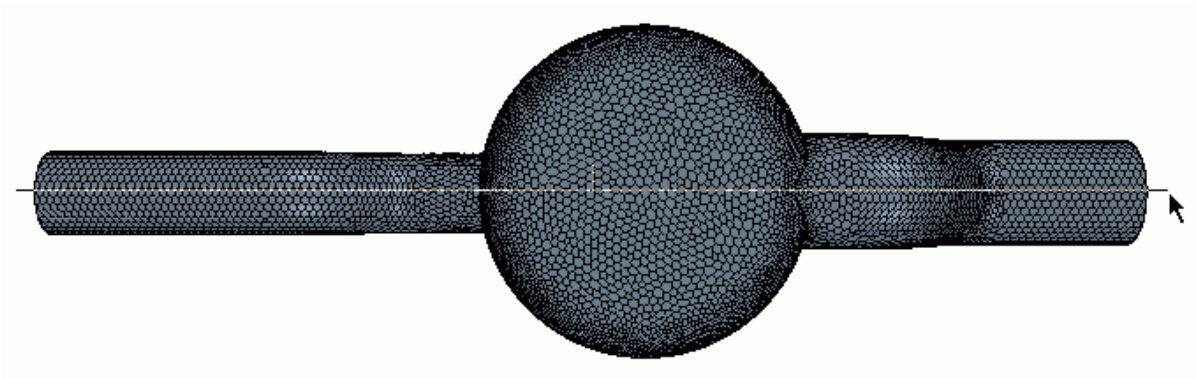
1. Generate the volume mesh by clicking (**Generate Volume Mesh**) in the toolbar.

   The *Output* window displays a summary of the mesh quality criteria (namely, the cell quality, face validity, skewness angle, and volume ratio) for the core and prism meshes.
2. To view the volume mesh when it is finished, right-click in the mesh scene and select **Apply Representation** > **Volume Mesh**.
3. In the *Vis* tool bar, click (**Save-Restore-Select Views**) and select **Views** > **-Y** > **Up +Z**.
4. Create a plane through the middle of the housing:

   a) In the *Vis* tool bar, click (**Create Plane Section**).

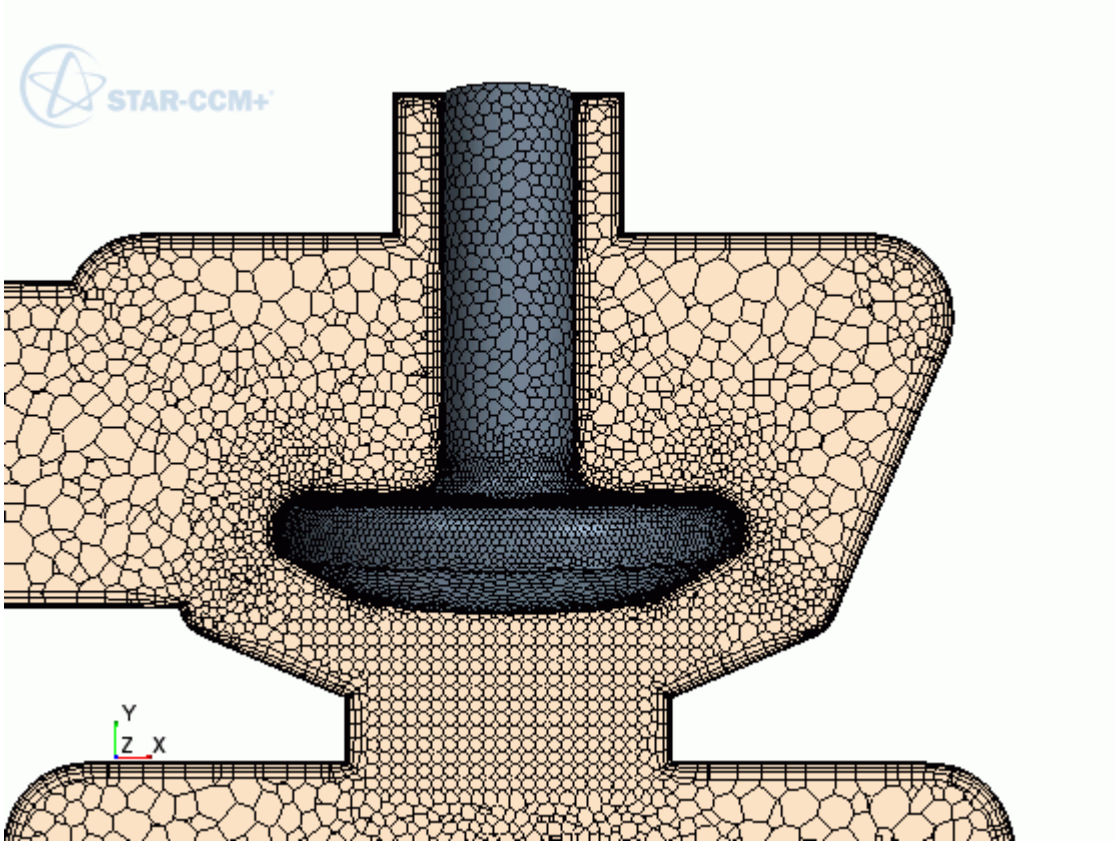
   b) Click just outside the housing at the inlet, at the middle.

   c) Complete the line by clicking just outside the outlet, also at the middle.



5. In the *Displayer* dialog that appears, select **Existing Displayer**, and specify **Mesh 1**.
6. Click **OK**.
7. To view the plane only, right-click the outside of the housing and select **Hide**.

   The polyhedral mesh is visible along the plane section.



Note: The totals for cells, interior faces, and vertices can differ slightly depending on your machine type and architecture.

8. Save the simulation.

Examining Mesh Statistics with Histogram Plots

Use histogram plots to examine characteristics of the polyhedral mesh.

Histogram plots are two-dimensional bar charts that can be used to display data distribution. They can be used to evaluate mesh quality.

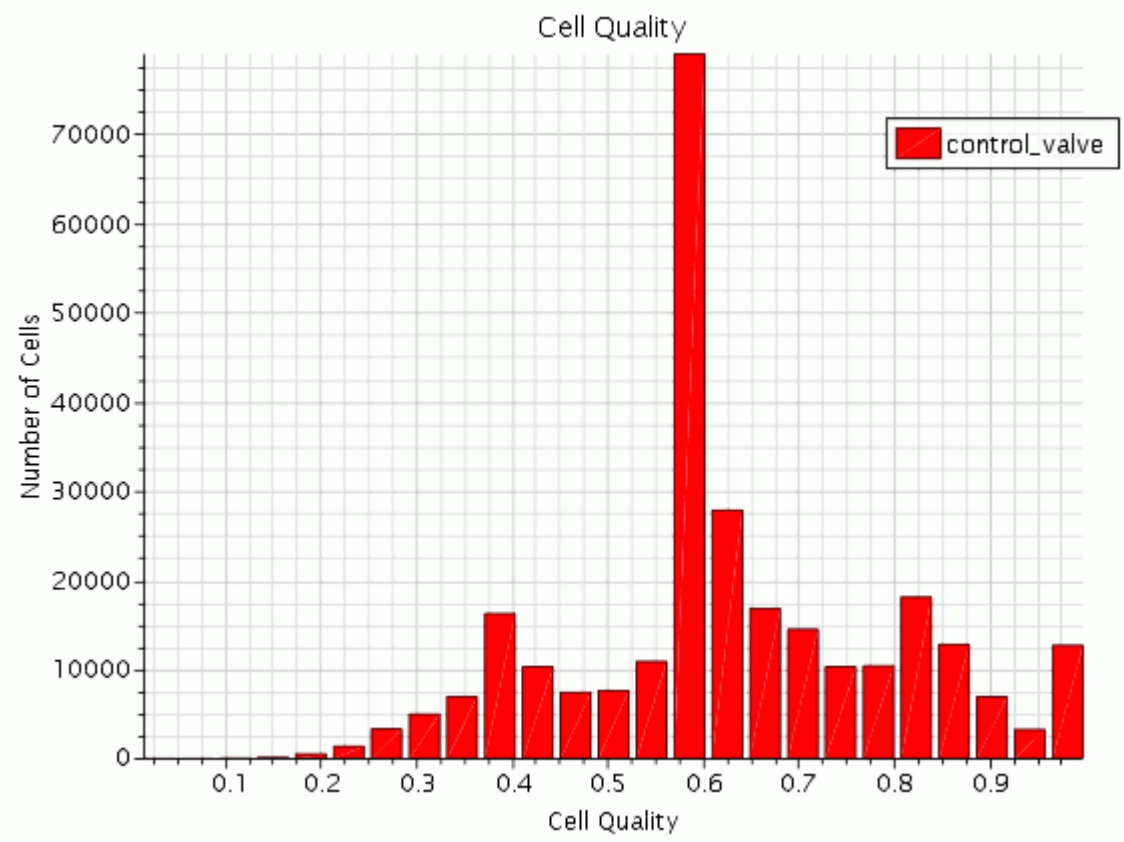
Set up a histogram plot for cell quality.

1. Right-click the **Plots** node and select **New Plot** > **Histogram Plot**.
2. Rename this plot to `Cell Quality`.
3. Edit the **Cell Quality** node and set the following properties:

Node	Property	Setting
Cell Quality	*Title*	`Cell Quality`
	Parts	**Regions** > **control_valve**
└ **Axis Type**		

Node	Property	Setting
└ **Bin**	*Field Function*	**Cell Quality**
└ **Axes**		
└ **Left Axis** > **Title**	*Title*	`Number of Cells`

This histogram plot displays the distribution of cells for each cell quality value.

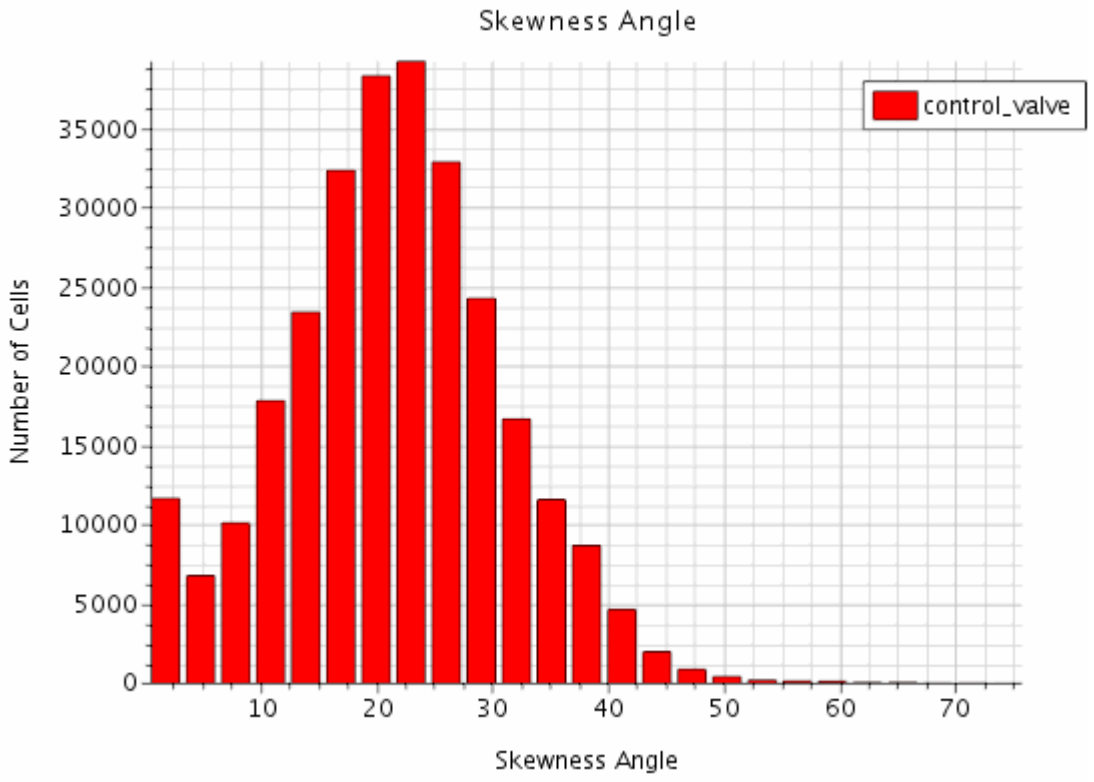


Set up another histogram plot for skewness angle.

4. Make a copy of the **Cell Quality** node.
5. Rename this copy to `Skewness Angle`.
6. Edit the **Skewness Angle** node and set the following properties:

Node	Property	Setting
Skewness Angle	*Title*	`Skewness Angle`
└ **Axis Type**		
└ **Bin**	*Field Function*	**Skewness Angle**

This histogram plot displays the distribution of cells for each skewness angle value.

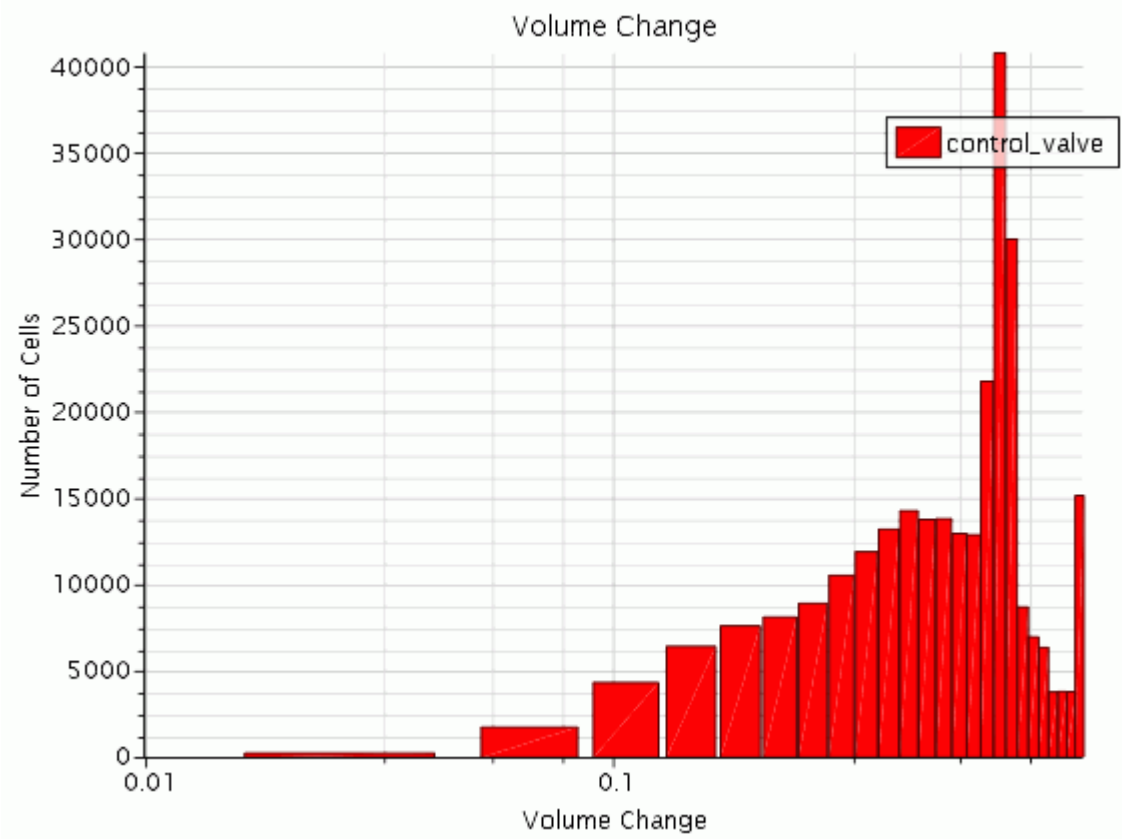


Set up a third histogram plot for volume change. Minimum acceptable volume change is small, so use a logarithmic scale for this plot.

7. Make a copy of the **Skewness Angle** node.
8. Rename this copy to `Volume Change`.
9. Edit the **Volume Change** and set the following properties:

Node	Property	Setting
Volume Change	*Title*	`Volume Change`
└ **Axis Type**		
└ **Bin**	*Field Function*	**Volume Change**
└ **Axes**		
└ **Bottom Axis**	*Logarithmic*	**Activated**

This histogram plot displays the distribution of cells for each value of volume change.



10. Save the simulation.

Trimmed Cell Meshing Example

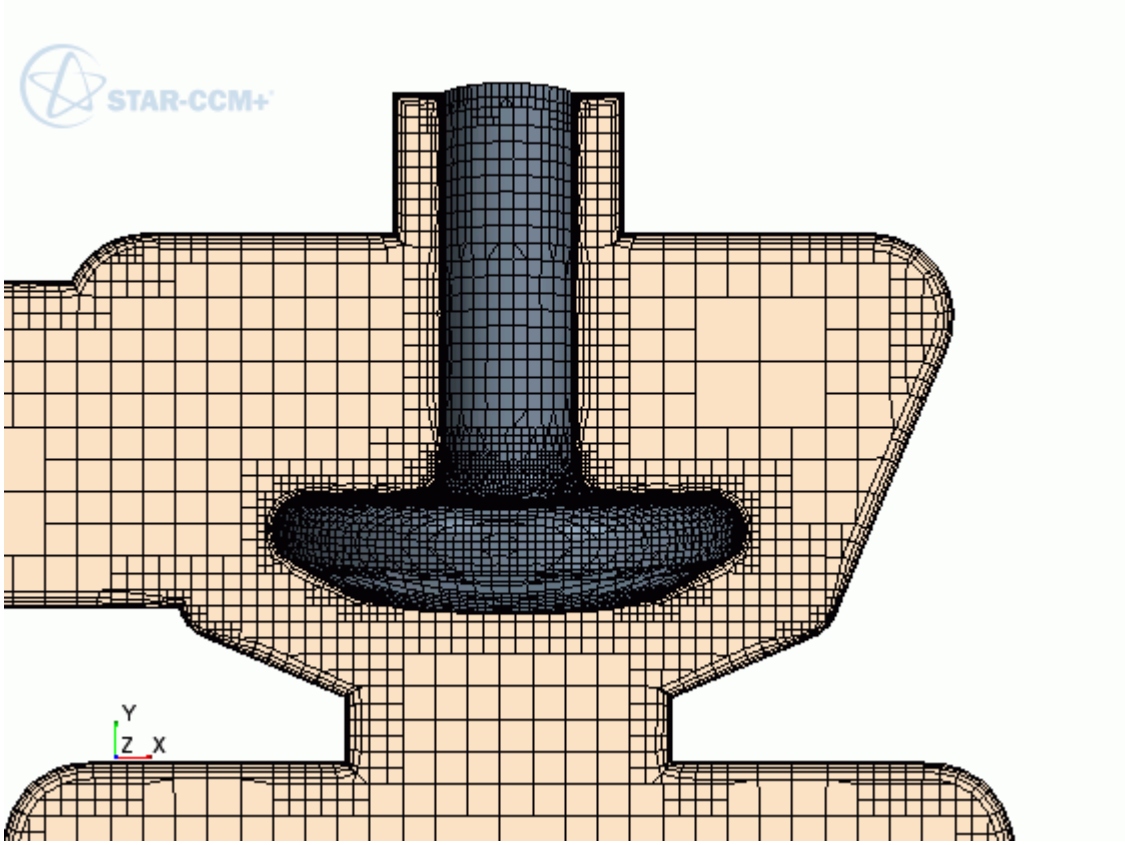
To generate a high-quality hexahedral based mesh for the original control valve geometry, use the trimmed cell mesher instead of the polyhedral mesher. The trimmed cell mesher also improves the distribution of cells which means fewer cells are required.

1. Right-click the **Geometry** > **Operations** > **Automated Mesh** > **Meshers** node and choose **Select Meshers**.
2. In the *Select Meshers* dialog, deactivate the meshers and activate the following ones, in order:

Group	Mesher
Surface Meshers	**Surface Remesher**
Core Volume Meshers	**Trimmed Cell Mesher**
Optional Boundary Layer Meshers	**Prism Layer Mesher**

3. Right-click the **Automated Mesh** node and select **Execute**.

   Look at the trimmer mesh.



Contents:

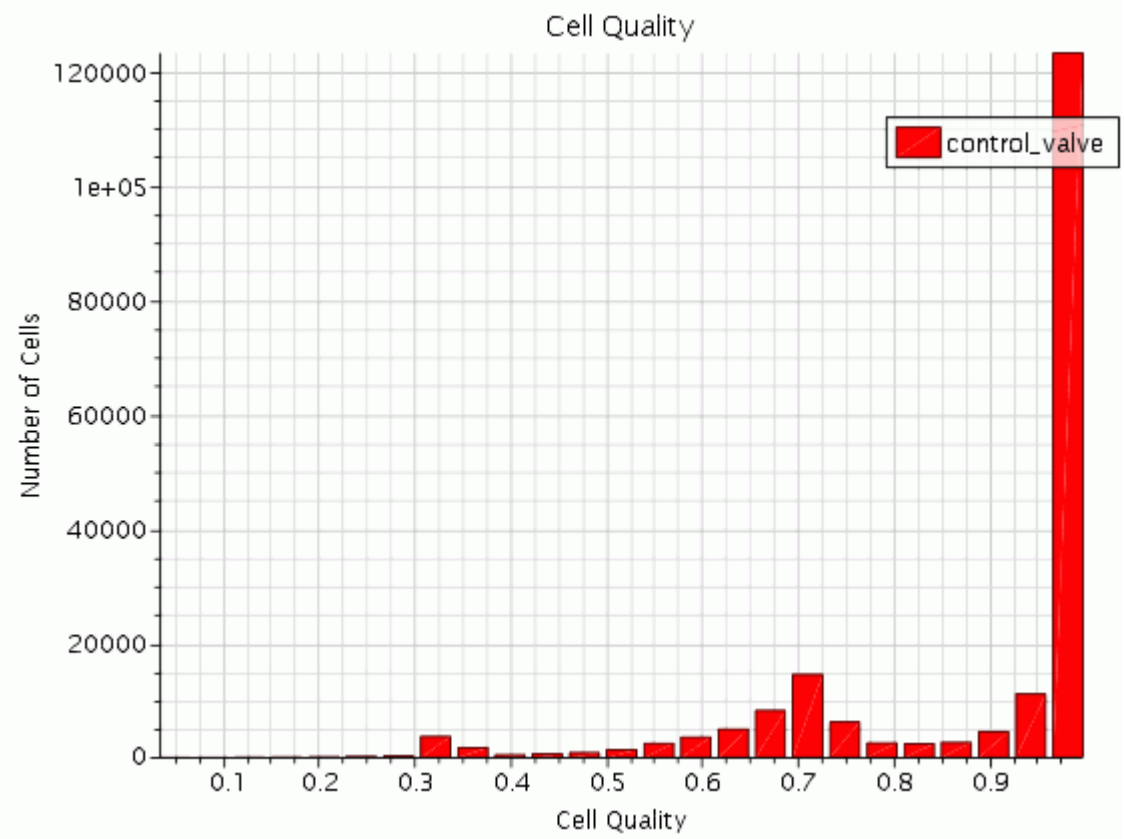
Examining Statistics for the Trimmer Mesh

Examining Statistics for the Trimmer Mesh

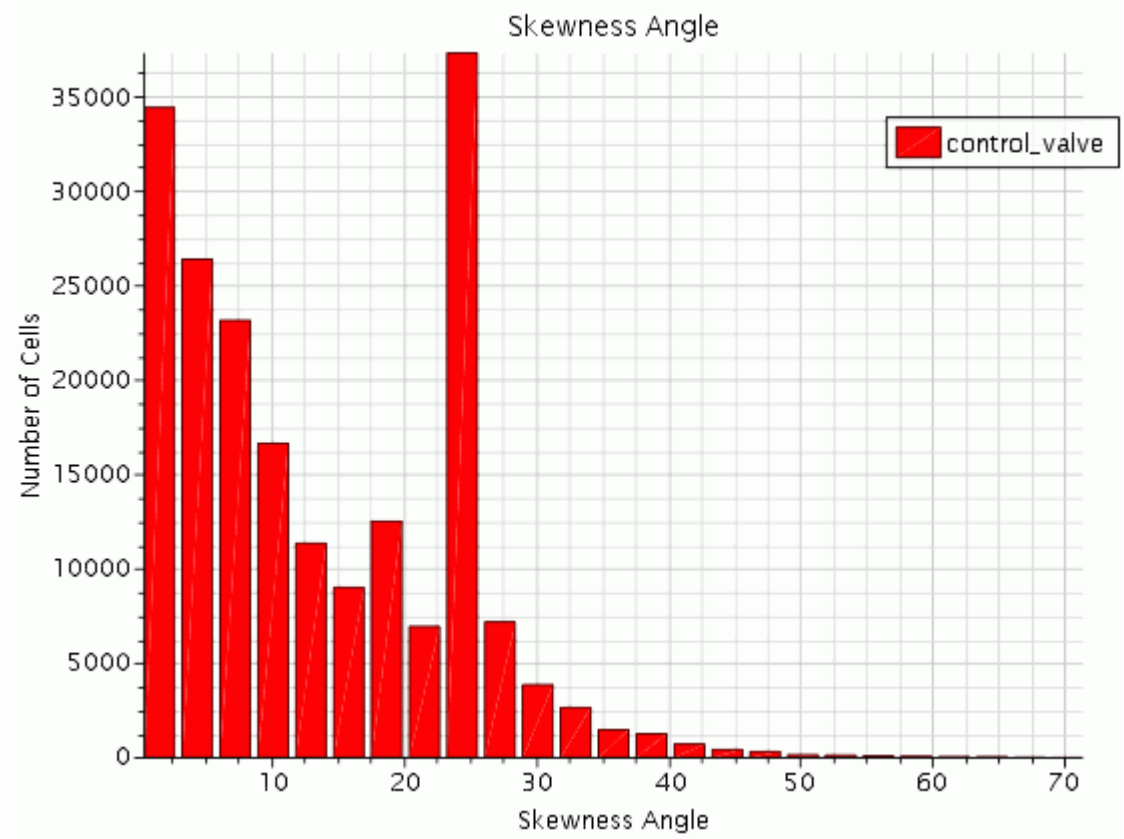
View the trimmer mesh quality in the histogram plots.

Take a look at the plots:

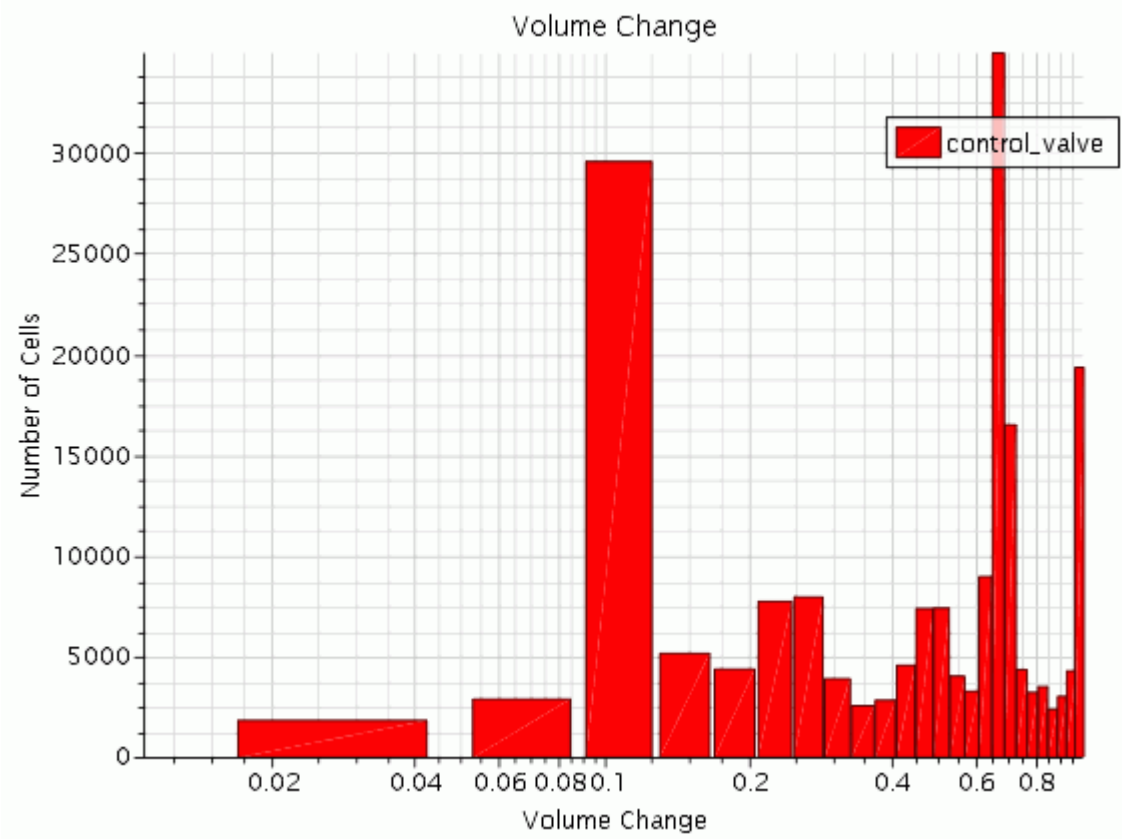
- Cell Quality



- Skewness Angle



- Volume Change



Summary

This tutorial demonstrated how to use the surface and volume meshing tools within STAR-CCM+ to build a single region polyhedral volume mesh with prism layers. It also demonstrated use of the trimmed mesher.

This tutorial introduced the following features:

- Importing an `.x_t` type surface data file
- Using transform mesh operations
- Using Boolean mesh operations
- Creating the region and assigning boundary types
- Setting mesh controls and generating a surface mesh
- Refining part of the mesh using a simple part and a volumetric control
- Running the surface remesher, polyhedral mesher, and prism layer mesher
- Running the trimmer mesher
- Examining the mesh quality with histogram plots

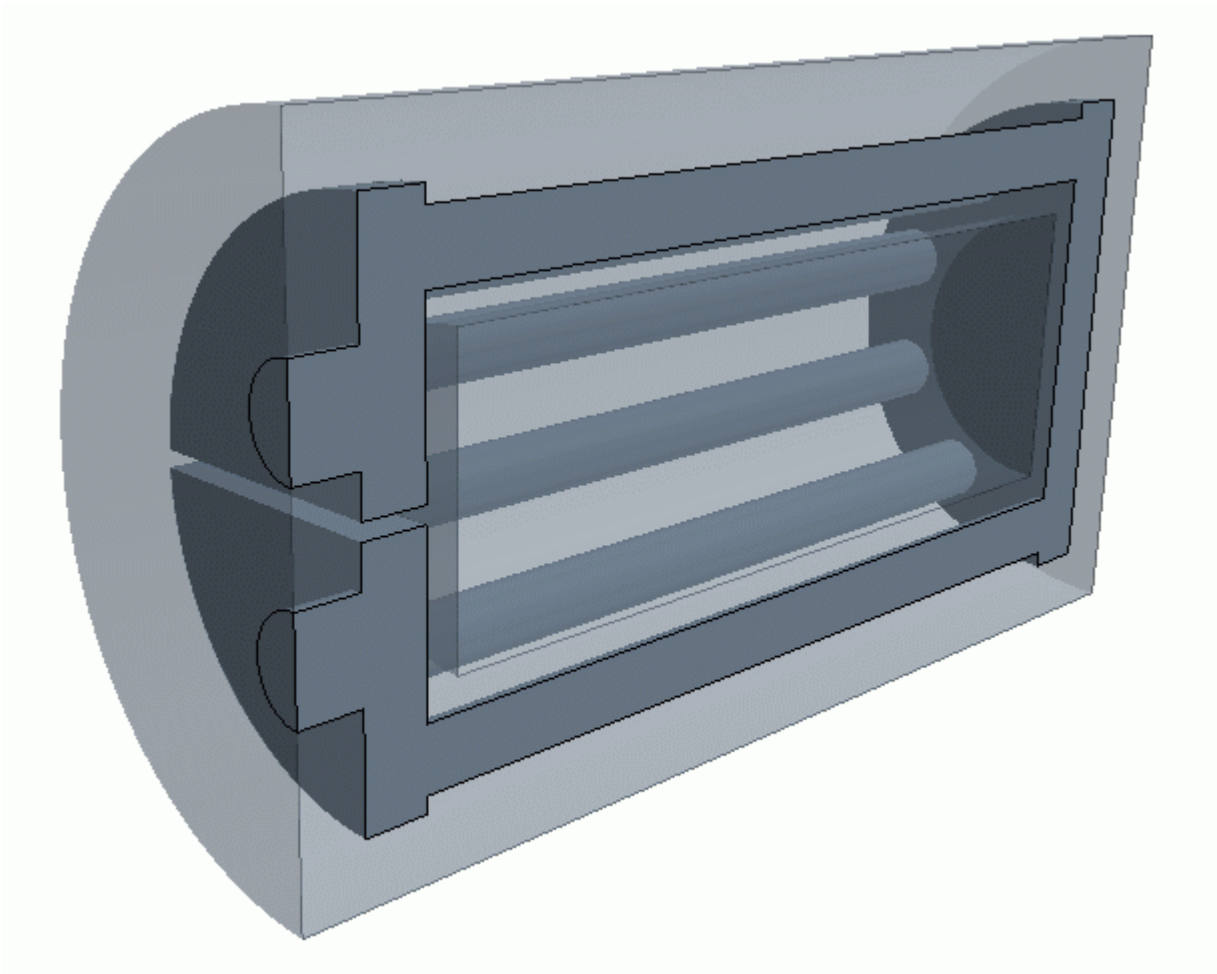
Meshing: Multi-Part Heat Exchanger

This tutorial demonstrates how to use the surface and volume meshing tools while working with multiple parts within STAR-CCM+.

Starting from a Parasolid Transmit geometry file, build a multi-part polyhedral volume mesh with prism layers. Define the appropriate surfaces, create a mesh for each part, and a conformal mesh at interfaces between parts.

The heat exchanger unit to be meshed contains multiple fluid flow passages connected by two end chambers, one of which is split to force the flow around the system. The flow domain is contained inside a solid cylinder, which is hollow on the inside. Two separate bodies are contained in the single Parasolid Transmit file. These are the fluid and solid parts of the geometry.

The heat exchanger geometry is shown below. The case is symmetrical so import and mesh half the geometry, only.



Contents:

Prerequisites
Importing the Surface Mesh
Defining the Part Surfaces
Assigning Parts to Regions
Defining the Region and Boundary Types
Selecting the Meshers
Setting the Mesh Default Controls
Disabling the Prism Layer Generation in the Solid
Remeshing the Surface Mesh
Generating the Volume Mesh
Summary

Prerequisites

The instructions in the Meshing Multi-Part Heat Exchanger tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Importing the Surface Mesh

To set up the STAR-CCM+ simulation, launch a simulation and import the supplied surface mesh.

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Surface Mesh**.
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/mesh/data`.
5. Select `hx_unit_asm.x_t` then click **Open**.
6. Click **OK** to accept the default settings.
7. Save the simulation as `heatExchanger.sim`.

Defining the Part Surfaces

Identify the flow, symmetry, and interface surfaces so that appropriate meshes for each can be generated.

During the import process, the Parasolid Transmit file geometry is translated into a triangulated surface representation with a total of two parts and one surface per part. Feature curves are included for each part based on the supplied angle.

CAD information distinguishing individual surfaces is stored as patches. Each part surface can be split into constituent surfaces using patch information.

Contents:

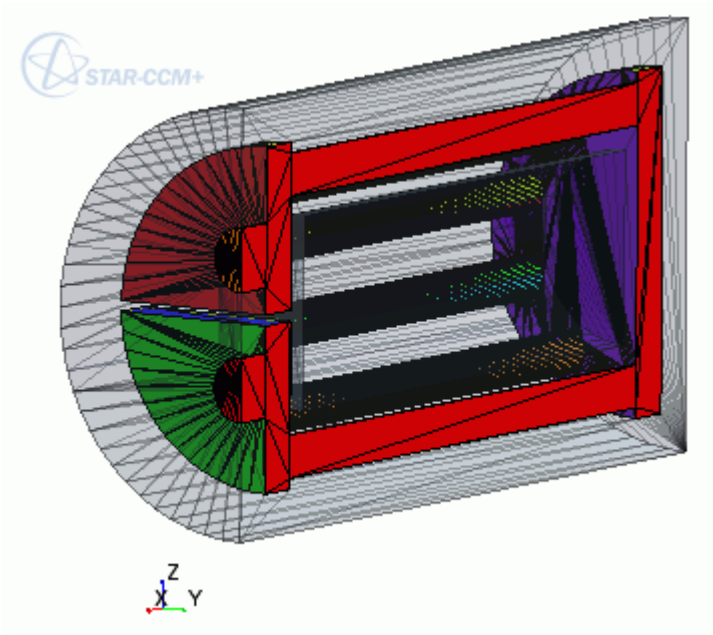
Extracting the Fluid Surfaces
Extracting the Solid Surfaces

Extracting the Fluid Surfaces

Split the imported fluid part surface into the constituent surfaces.

1. Expand the **Geometry** > **Parts** node.
2. Rename **hx_unit_asm1** to `Fluid Body`.
3. Right click the **Fluid Body** > **Surfaces** > **Faces** node and select **Split by Patch...**

   The *Split Part Surface by Patch* dialog appears. Surfaces that belong to the fluid part are color-coded according to their patch definition. All other surfaces are translucent, as you can see below – where **Show All Meshes** is activated:



You can use the *Split Part Surface by Patch* dialog to select a patch or patches and create a surface from the original. Select the patch directly on the model by clicking the appropriate face; the corresponding patch number is highlighted in the dialog.

Create the following surfaces for the fluid region:

4. In the *Split Part Surface by Patch* dialog:
   a) Select patch number 283.

      The associated surface is highlighted in the model.
   b) Set *Part Surface Name* to `Inlet` and click **Create**.

The newly extracted surface patch is removed from the list and from the model display.

c) Repeat the above process to create the following surfaces:

Patch Number	Surface Name
284	`Outlet`
285	`Symmetry`

d) Click **Close**.

Inlet, **Outlet**, and **Symmetry** nodes appear under the **Fluid Body** > **Surfaces** node.

The fluid part original surface, **Faces**, is now composed of surfaces that belong solely to the interface plane between the fluid and solid parts.

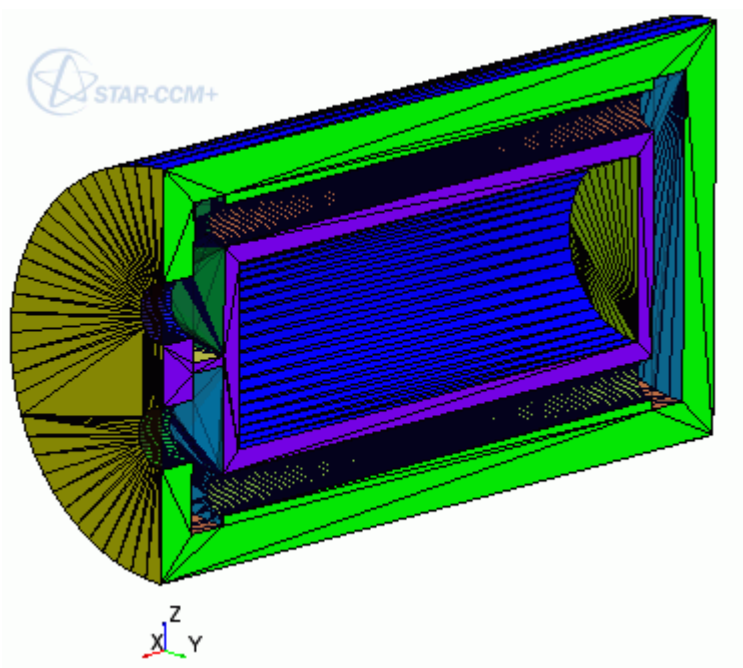
5. Rename **Faces** to `Interface`.

Extracting the Solid Surfaces

Extract the solid surfaces using the *Split by Patch* technique.

1. Expand the **Geometry** > **Parts** node.
2. Rename **hx_unit_asm2** to `Solid Body`.
3. Right click the **Solid Body** > **Surfaces** > **Faces** node and select **Split by Patch...**

   The *Split Part Surface by Patch* dialog appears. Surfaces that belong to the solid part are color-coded according to their patch definition. All other surfaces are translucent, as shown below with **Show All Meshes** activated:



4. In the *Split Part Surface by Patch* dialog:

   a) Create the following surfaces for the solid part:

Patch Number	Surface Name
288, **289**, **290**	`Outer cylinder wall`
291, **292**	`Symmetry`
293, **294**, **295**	`Inner cylinder wall`

b) Click **Close**.

The surface patches are removed from the list and from the model display. **Outer cylinder wall**, **Symmetry**, and **Inner cylinder wall** nodes appear under the **Solid Body** > **Surfaces** node.

The solid part original surface, **Faces**, is now composed of surfaces that belong solely to the interface plane between the fluid and solid parts.

5. Rename **Faces** to `Interface`.

Assigning Parts to Regions

Before setting up the mesh for this tutorial, assign the imported parts to regions.

1. Expand the **Geometry** > **Parts** node and multi-select the **Fluid Body** and **Solid Body** nodes.
2. Right-click one of the selected nodes and select **Assign Parts to Regions...**
3. In the *Assign Parts to Regions* dialog, select:
   a) **Create a Region for Each Part**
   b) **Create a Boundary for Each Part Surface**
   c) **Create Boundary-mode Interfaces From Contacts**
4. Click **Apply** then **Close**.

   STAR-CCM+ adds two regions, **Fluid Body** and **Solid Body**, within the **Regions** node and assigns all part surfaces to the corresponding region boundaries. An interface between the regions, **Fluid Body/ Solid Body** is also created.

Defining the Region and Boundary Types

Set appropriate types for the regions and their boundaries.

1. Select the **Regions** > **Solid Body** node and set *Type* to **Solid Region**.

   The **Fluid Body** region *Type* is already set to **Fluid Region**.

To set the boundary types:

2. Edit the **Regions** > **Fluid Body** > **Boundaries** node and set the following properties:

Node	Property	Setting
Inlet	*Type*	**Velocity Inlet**
Outlet	*Type*	**Pressure Outlet**
Symmetry	*Type*	**Symmetry Plane**

3. Select the **Solid Body** > **Boundaries** > **Symmetry** node and set *Type* to **Symmetry Plane**.

Selecting the Meshers

Create a single Automated Mesh operation for both the fluid and the solid.

Use the **Surface Remesher** to generate the surface mesh. Use the **Polyhedral Mesher** and the **Prism Layer Mesher** to generate the volume mesh.

1. Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
2. In the *Create Mesh Operation* dialog:
   a) From the *Parts* list, select all parts.
   b) From the *Select Meshers* group box, select the following meshers:

Group Box	Mesher
Surface Meshers	**Surface Remesher**
Core Volume Meshers	**Polyhedral Mesher**
Optional Boundary Layer Meshers	**Prism Layer Mesher**

   c) Click **OK**.

   The default properties for these models are appropriate to this simulation.

Setting the Mesh Default Controls

Set the default controls to generate the mesh.

For this example, absolute size specification is used for the prism layer total thickness and surface size values. Although the model dimensions are in meters (automatically set during the import process of the Parasolid Transmit file), you can use any units to specify the inputs for the meshers.

Increase the # Points in a gap value for Surface Proximity so that an increased surface triangulation density is obtained in the vicinity of the pipes and inner cylinder surface, which in turn results in a higher concentration of volume cells.

1. Edit the **Geometry** > **Operations** > **Automated Mesh** > **Default Controls** node and set the following properties:

Node	Property	Setting
Number of Prism Layers	*Number of Prism Layers*	`1`
Prism Layer Total Thickness	**Absolute**	`2.0 mm`
Surface Proximity	*# Points in gap*	`4.0`
	Search Floor	`0.0 m` (default)
Minimum Surface Size	**Absolute**	`4.0 mm`
Target Surface Size	**Absolute**	`8.0 mm`

Disabling the Prism Layer Generation in the Solid

Define a custom Surface Control to deactivate prism layer generation in the solid part.

To deactivate the prism layers for the solid:

1. Right-click the **Automated Mesh** > **Custom Controls** node and select **New** > **Surface Control**.

Apply the surface control to the solid body surfaces:

2. Select the **Custom Controls** > **Surface Control** node and click (**Custom Editor**) for the *Part Surfaces* property.
3. In the *Surface Control - Part Surfaces* dialog, expand the **Solid Body** > **Surfaces** node, select all surfaces, and click **OK**.
4. Select the **Surface Control** > **Controls** > **Prism Layers** node and set *Prism Layers* to **Disable**.

   The generation of prism layers for the surfaces in the solid part is now deactivated.

Remeshing the Surface Mesh

Since the surface remesher and volume mesher have been selected in tandem, you could proceed and generate both meshes in one go. However, for training purposes, first generate the surface remesher mesh and view it before proceeding with the volume meshing.

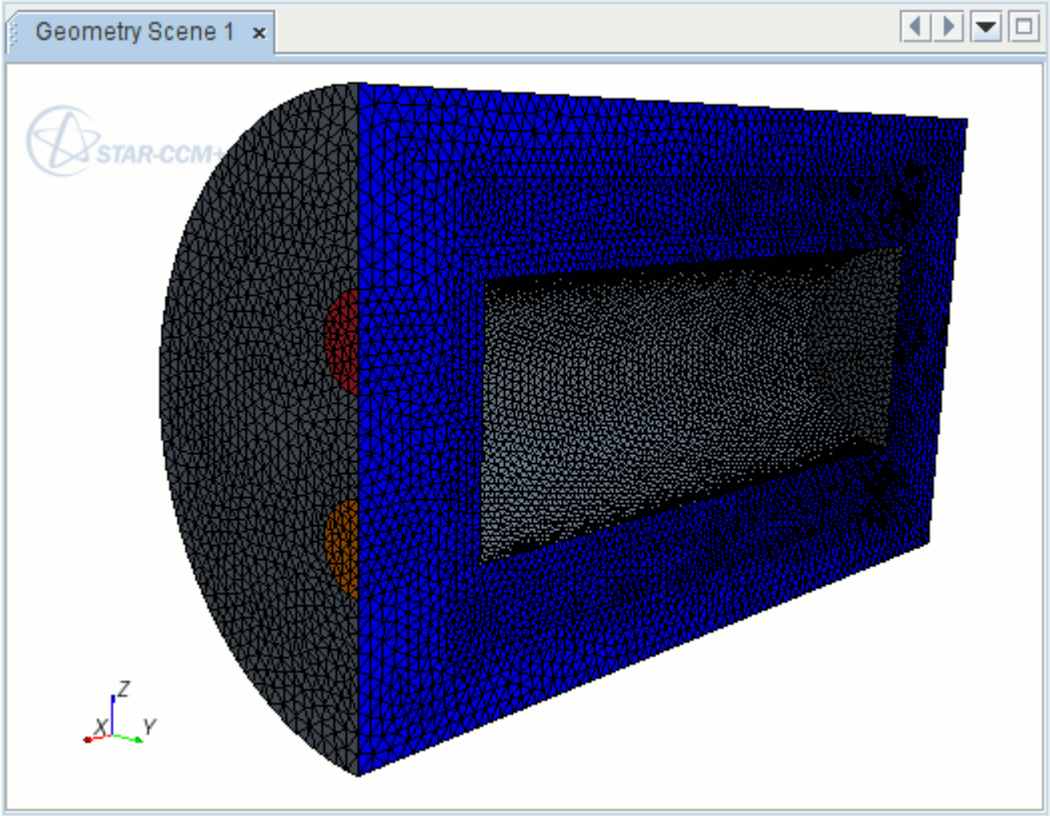
1. To generate the surface remesher mesh, either:
   - In the Mesh Generation toolbar, click (**Generate Surface Mesh**), or
   - Select **Mesh** > **Generate Surface Mesh**.

   The output of the surface remesher is shown in the Output window and status shown in the status bar. When the process is complete, the message `Surface Meshing Pipeline Completed` is shown in the *Output* window.

To view the remeshed surface representation:

2. Right-click the **Scenes** > **Geometry Scene 1** node and select **Apply Representation** > **Automated Mesh.Remesh**. Make sure that **Show All Meshes** is activated.

   The scene updates to display the triangulated surface for the heat exchanger unit geometry.



Surface curvature and proximity-based refinement is visible on the curved parts (that is, the pipes) of the geometry. The properties of the remeshed surface can be seen by clicking the appropriate node in the Representations node.

Generating the Volume Mesh

Run the polyhedral and prism layer volume meshers.

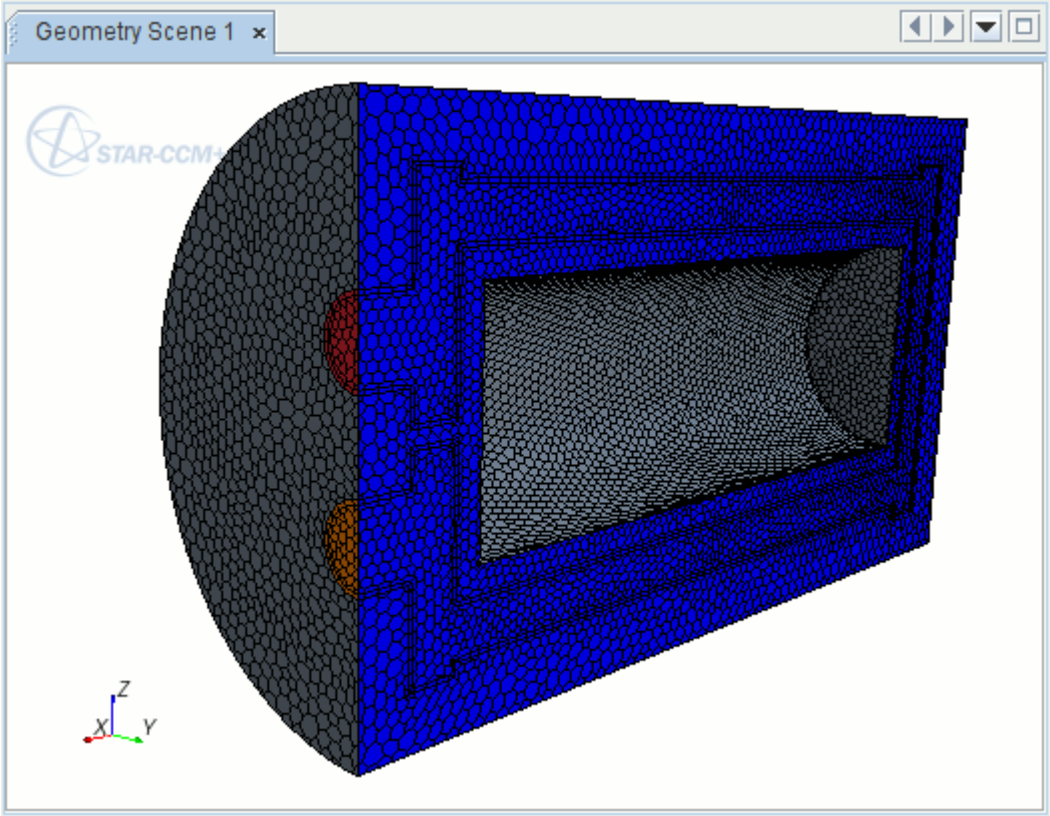
1. To run the polyhedral and prism layer meshers, either:
   - In the **Mesh Generation** toolbar, click (**Generate Volume Mesh**), or
   - Select **Mesh** > **Generate Volume Mesh**.

   As before, the output from the meshers is shown in the *Output* window and status updated in the status bar. When the process is complete, the message `Volume Meshing Pipeline Completed` is shown in the *Output* window. A summary of the mesh generation process for the core and prism meshes can be viewed by scrolling up the *Output* window.

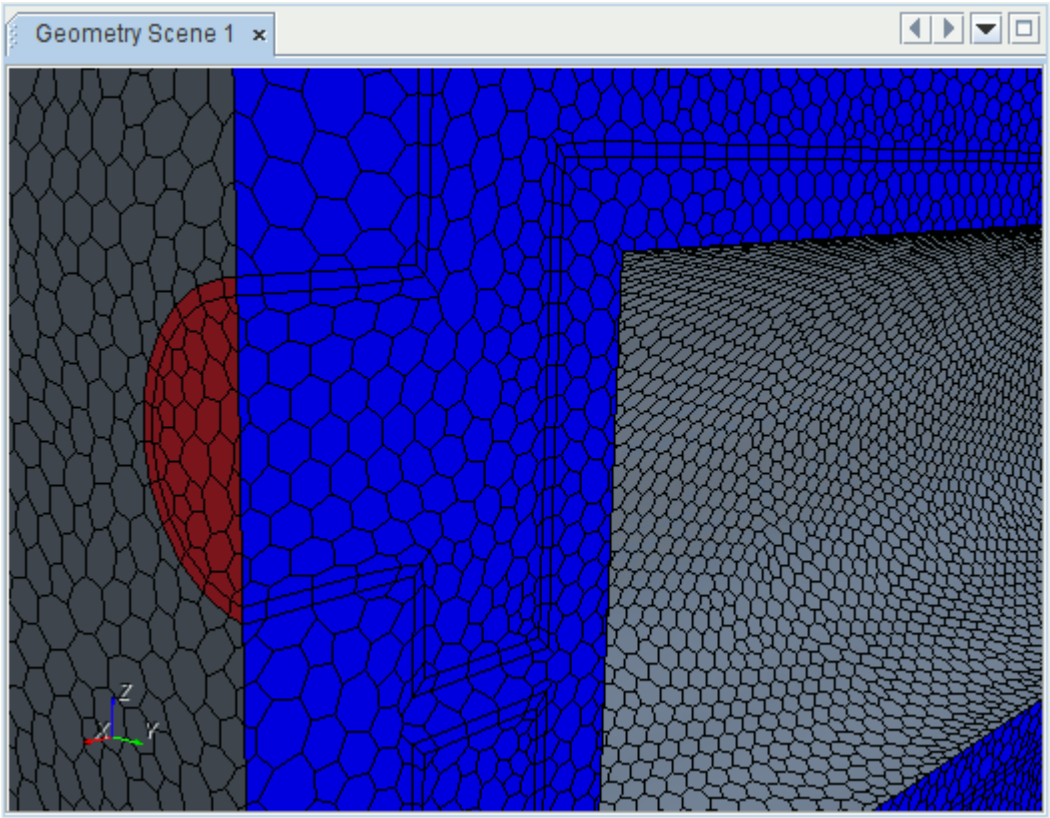
To view the volume mesh representation:

2. Right-click the **Scenes** > **Geometry Scene 1** node and select **Apply Representation** > **Volume Mesh**

   The scene updates to show the polyhedral volume mesh with a prismatic near wall layer for the fluid region:

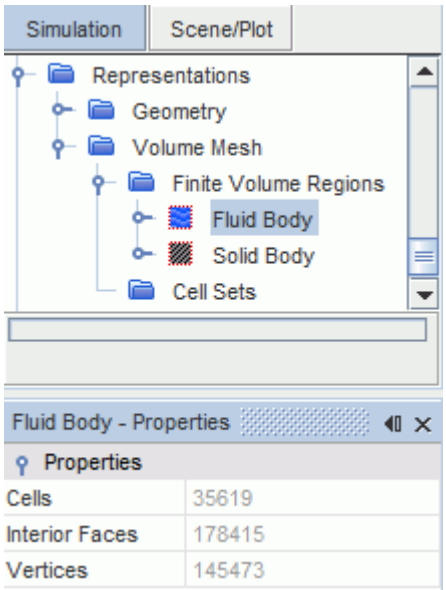


A close up of the mesh in the vicinity of the inlet plenum is shown below:



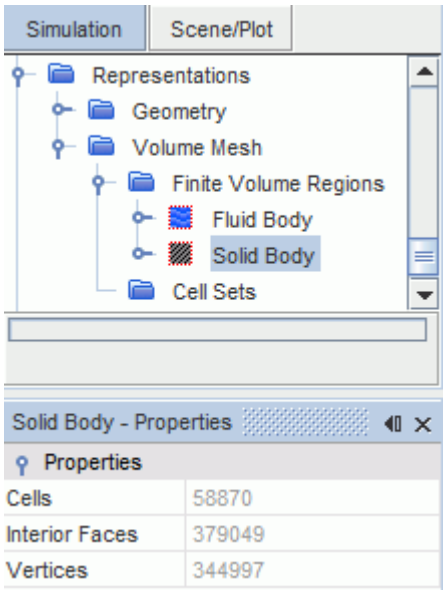
To view the properties of the fluid volume mesh, including the number of cells in the fluid region:

3. Expand the **Representations** > **Volume Mesh** > **Finite Volume Regions** node and select the **Fluid Body** node.



Similarly, to view the properties of the solid body volume mesh:

4. Select the **Solid Body** node.



The exact values for cells, interior faces, and vertices differ to the above depending on the machine precision.

Summary

This tutorial demonstrated how to use the surface and volume meshing tools while working with multiple parts within STAR-CCM+.

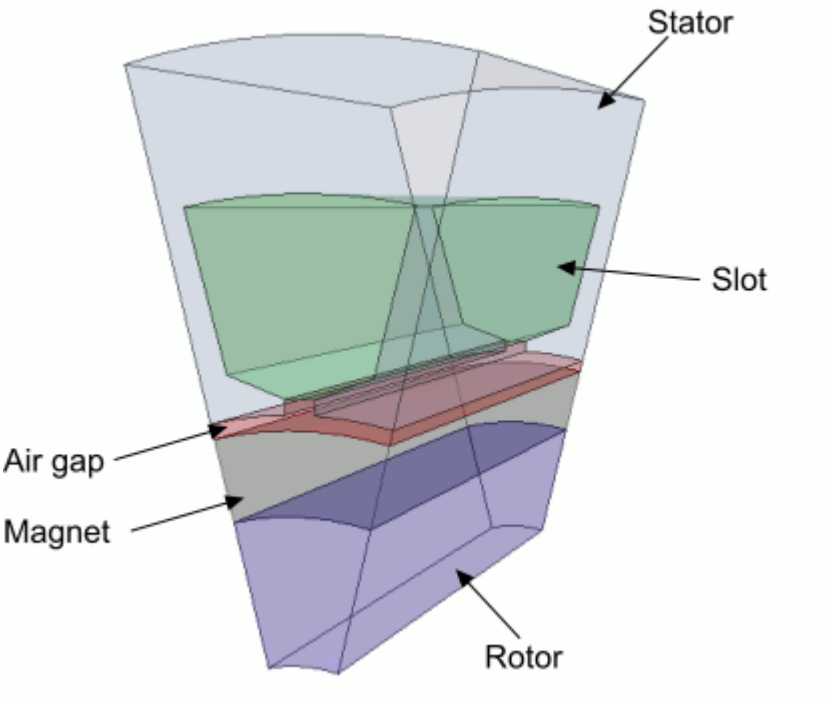
The steps covered were:

- Importing a Parasolid Transmit type geometry data file.
- Creating an Automated Mesh operation.
- Selecting the meshers and define default controls.
- Disabling the prism layers on selected surfaces.
- Running the Surface Remesher, Polyhedral, and Prism Layer meshers.
- Displaying representations.

Directed Meshing: Electrical Machine

This tutorial demonstrates how to use the directed meshing tool in STAR-CCM+. Directed meshing allows you to generate a high-quality structured mesh by creating quadrilateral elements on a surface of a body and then sweeping them through its volume.

In this tutorial, you are given a section of an electrical machine comprising five parts: rotor, magnet, air gap, slot, and stator. The parts are all connected and have regular section profiles, making this geometry a good case to use with directed meshing.



Contents:

Prerequisites
Loading the Simulation
Checking Suitability for Directed Meshing
Assigning Parts to Regions
Using Patch Meshing
Using Existing Meshes
Summary

Prerequisites

The instructions in the Directed Meshing tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Loading the Simulation

For this tutorial, you are provided with a simulation file that contains pre-defined simulation objects.

Load the initial simulation:

1. Launch STAR-CCM+.
2. Select **File** > **Load Simulation...**
3. In the *Load Simulation* dialog, click **Browse...**
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/mesh/data`.
5. Select `directedMeshCAD.sim` then click **Open**.
6. In the *Load Simulation* dialog, click **OK**.
7. Save the simulation as `directedMeshTutorial.sim`.

Checking Suitability for Directed Meshing

The directed meshing tool takes a surface mesh that is generated on a source surface and sweeps it through the volume to the target surface. The tool uses the remaining surfaces to guide the mesh. Therefore, each part that you choose to use in directed meshing must have at least three distinct surfaces. You can make sure that the parts satisfy this requirement by looking through the defined surfaces of each geometry part.

1. Create a geometry scene.

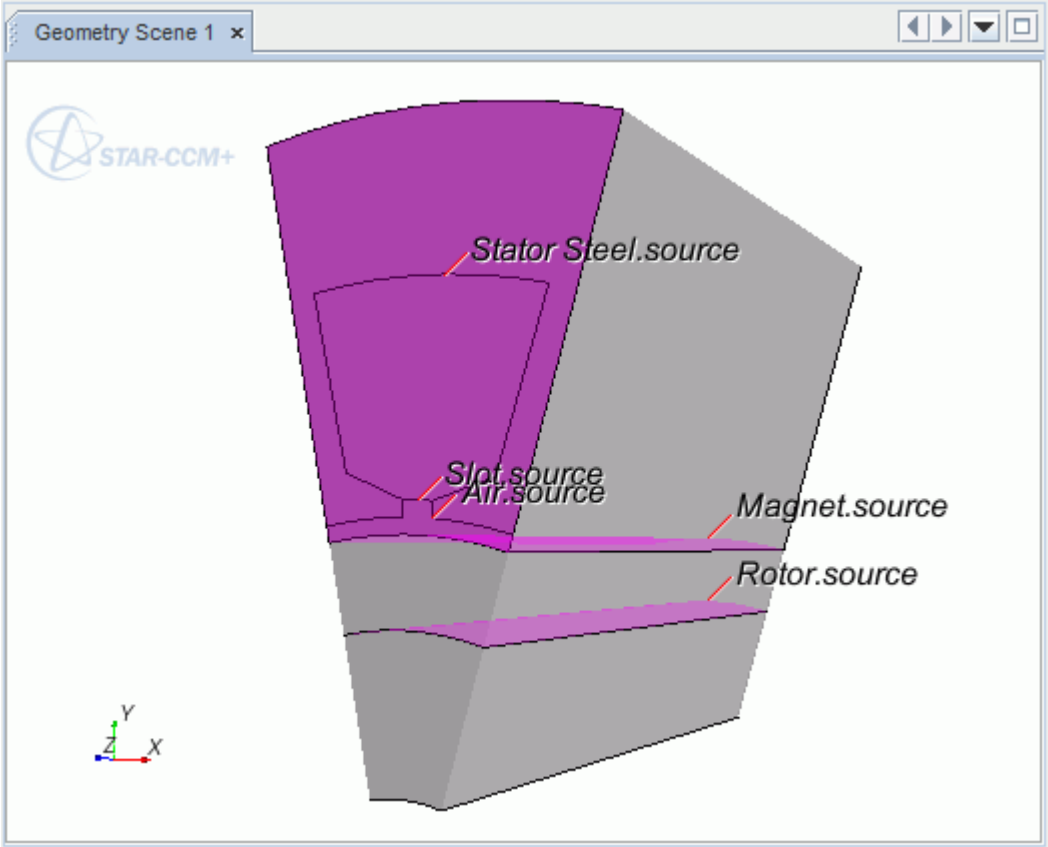
View the CAD bodies and parts:

2. Expand the **Geometry** > **3D-CAD Models** > **3D-CAD Model 1** > **Bodies** node.
3. To view the parts that are shown in the geometry scene, expand the **Geometry** > **Parts** node.

Each part is associated with a CAD body. To use directed meshing, associate each part with exactly one body.

4. Expand all **Geometry** > **Parts** nodes.
5. Multi-select all **[Part]** > **Surfaces** > **[source]** nodes.

   The selected surfaces are highlighted in the *Geometry Scene*.



The subdivision of part surfaces has been chosen to demonstrate the two methods available in the Directed Mesh tool for creating quadrilateral meshes on the source surfaces:

- Using manually created patches
- Using an extracted surface of an existing volume mesh

The **Stator Steel.source**, **Slot.source**, and **Air.source** faces are meshed using the patch meshing option. They can be meshed together since the source surfaces are connected and lie on the same plane. The patch mesh is then swept through the volume of the Stator Steel, Slot, and Air parts to produce a volume mesh for these parts.

The **Magnet.source** and **Rotor.source** faces are meshed using an existing mesh of a part surface they are in contact with. The **Air.bottom** surface mesh is mapped to the **Magnet.source** surface, allowing you to generate a volume mesh for the Magnet part. The **Magnet.target** surface mesh is then mapped to the **Rotor.source** surface. Note the **Air.bottom** surface has been separated out from the other Air surfaces to represent the surface that is in contact with the **Magnet.source** surface. This breakdown of surfaces is a requirement if you want to map an existing mesh to a source surface mesh–both parts in contact must have a separate part surface for the interface.

Assigning Parts to Regions

Since a volume mesh cannot be generated for parts that are not associated with a region, assign the parts to regions.

1. Multi-select all **Geometry** > **Parts**.

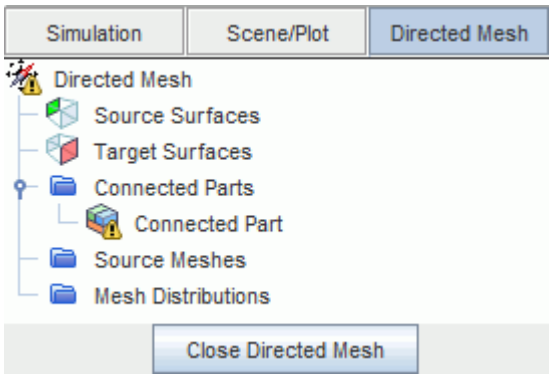
2. Right-click one of the selected parts and select **Assign Parts to Regions**.
3. In the Assign Parts to Regions dialog, select the following options:
   - **Create a Region for Each Part**
   - **Create One Boundary for All Part Surfaces**
4. Click **Apply** then **Close**.
5. Save the simulation.

Using Patch Meshing

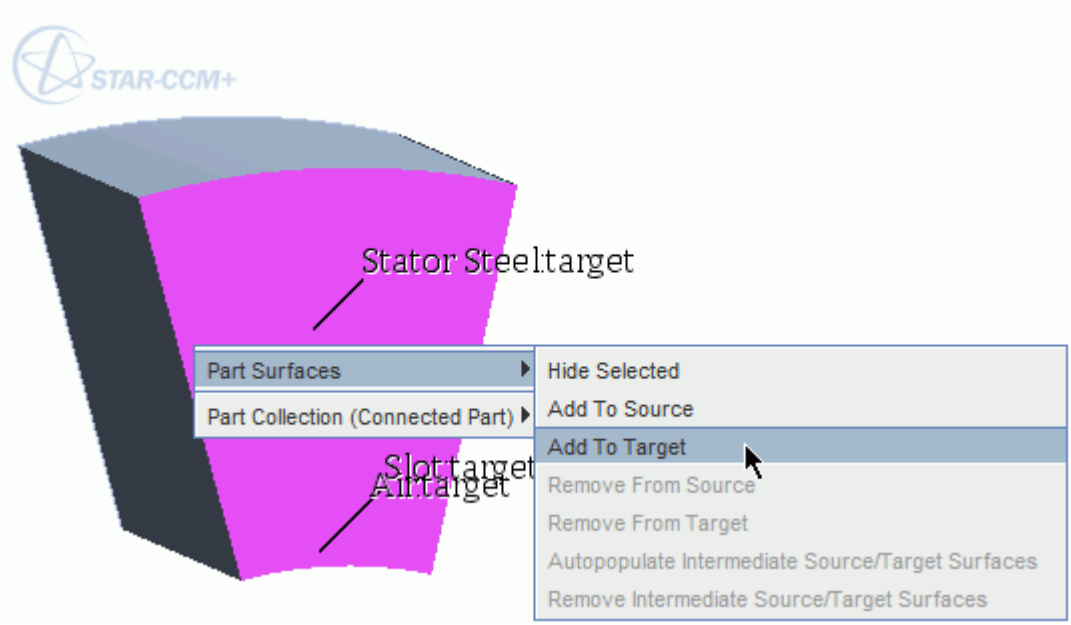
You can mesh the Air, Slot, and Stator Steel parts at the same time using patch meshing. Patch meshing requires you to create patches on the source surface manually. STAR-CCM+ uses these patches to generate a quadrilateral surface mesh. The three parts have contacts that are defined between them, allowing them to be used together in one directed mesh operation.

1. Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Directed Mesh**.
2. In the *Create Directed Mesh Operation* dialog, select the **Air**, **Slot**, and **Stator Steel** parts.
3. Click **OK**.
4. Right-click the **Geometry** > **Operations** > **Directed Mesh** node and select **Edit**.

   The *Directed Mesh* panel that is used for all directed mesh operations appears. Only the selected parts are shown in the *Graphics* window.

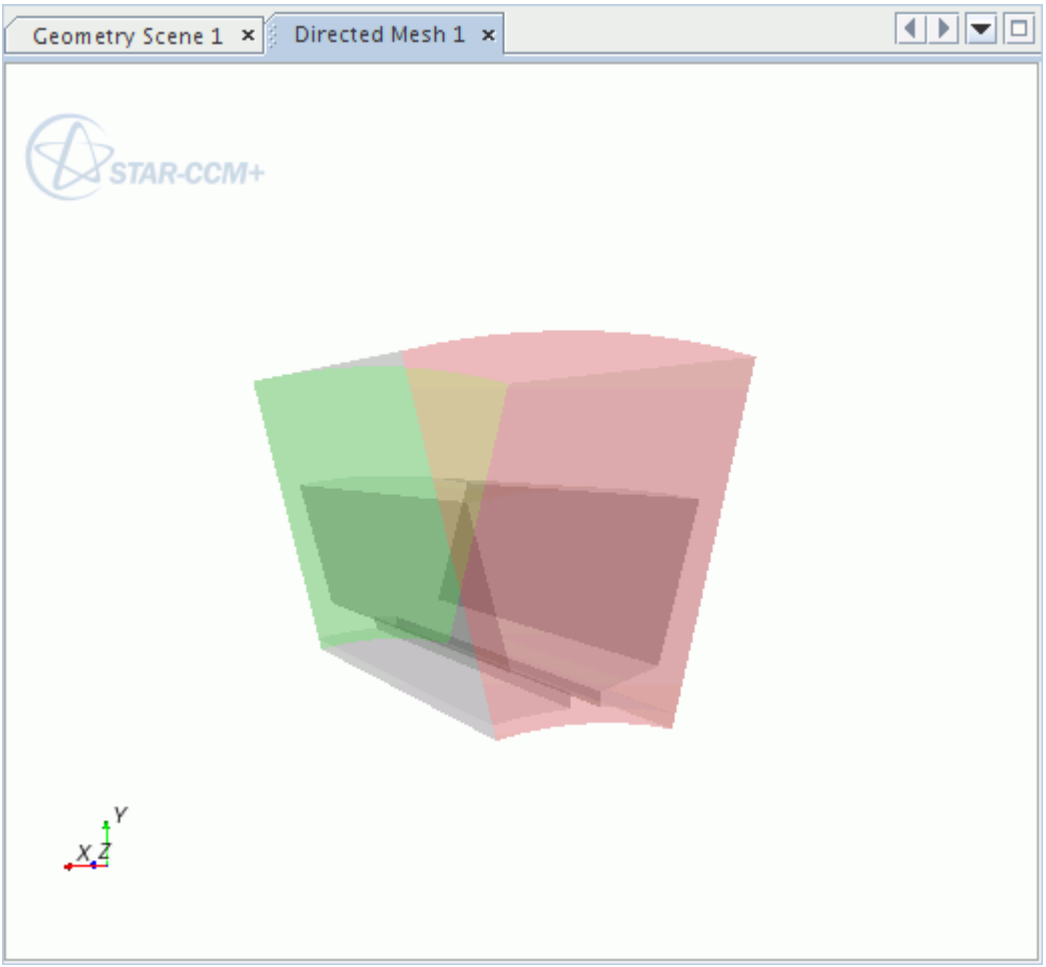


5. In the *Directed Mesh 1* window, select the three target surfaces:
   - **Stator Steel.target**
   - **Slot.target**
   - **Air.target**
6. Right-click on one of the selections and choose **Part Surfaces** > **Add To Target** .



7. In the *Directed Mesh 1* window, select the three source surfaces:
   - **Stator Steel.source**
   - **Slot.source**
   - **Air.source**
8. Right-click on one of the selections and choose **Part Surfaces** > **Add To Source**.

   The source surfaces are grouped together in the Properties of the **Directed Mesh** > **Source Surfaces** node and are colored green in the *Graphics* window. The target surfaces are grouped in the **Target Surfaces** node and are colored red. The surfaces are shown in the following image with the scene transparency turned on.



Contents:

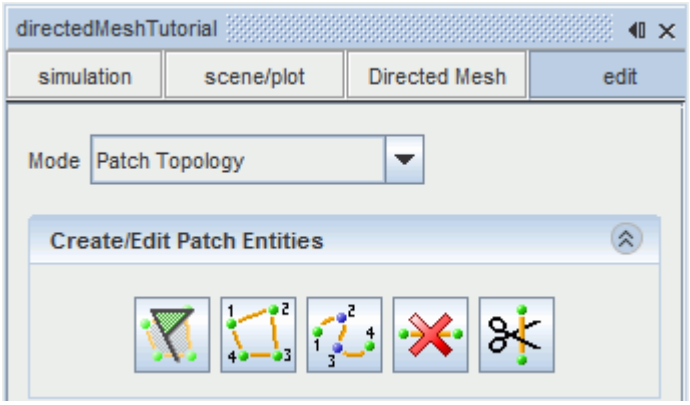
Creating a Patch Topology
Creating a Patch Mesh
Generating a Volume Mesh

Creating a Patch Topology

To use patch meshing, you create patches on the source surfaces. A closed loop of four curves define the patches.

1. Right-click the **Source Meshes** node and select **New Source Mesh** > **Patch Mesh**.
2. In the *Part Collection for this source mesh* dialog, select **Connected Part** and click **OK**.

   The *Edit* panel that is used to create patches and a patch mesh appears.

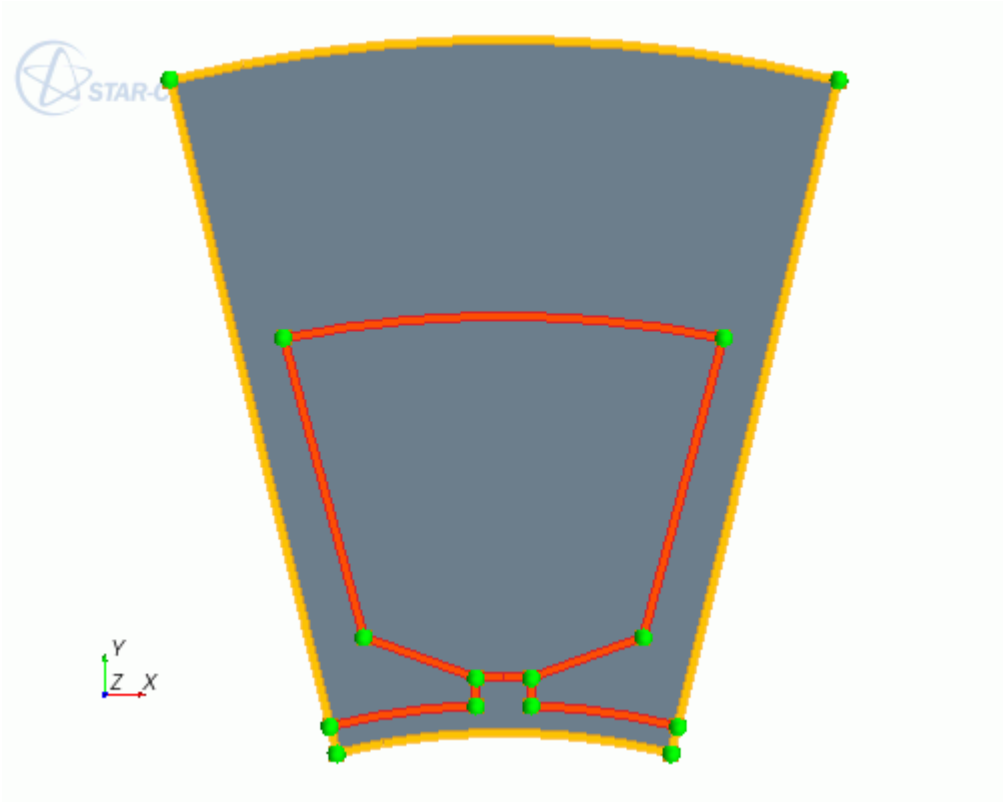


The *Graphics* window displays the source surface in gray with feature curves outlined in yellow.

3. Click (**Save-Restore-Select Views**) and select **Views** > **+Z** > **Up +Y**.

4. In the *Create/Edit Patch Entities* box, click (**Auto-populate feature edges with patch curves**).

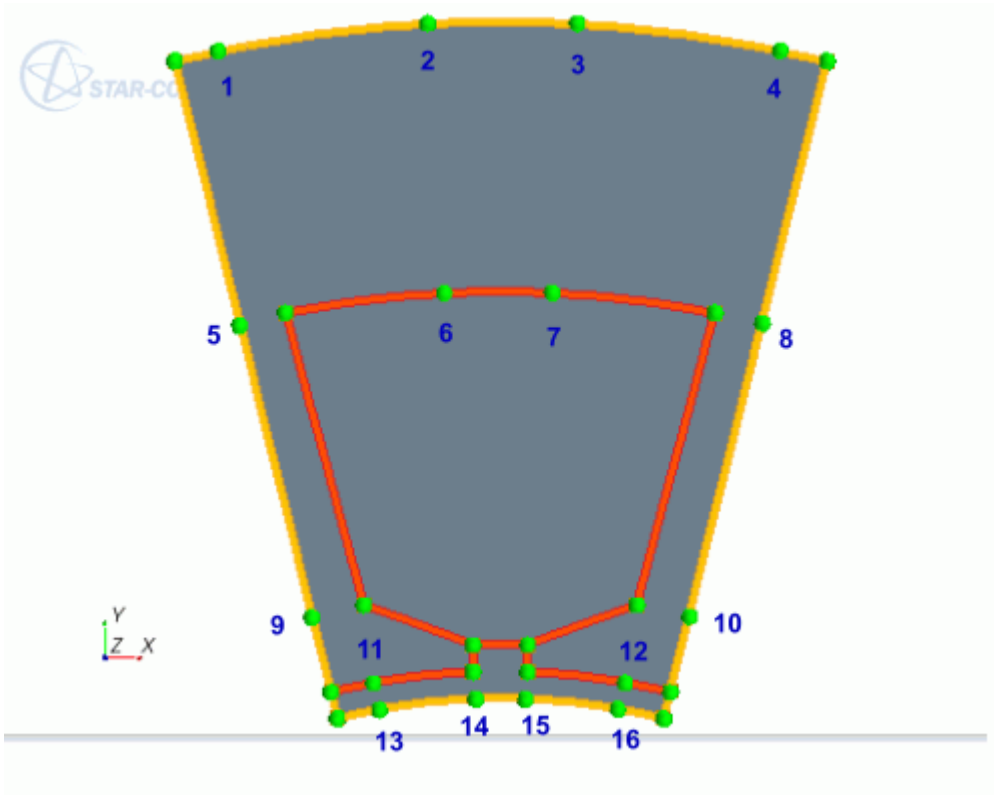
   This step creates patch curves on feature edges.



This patch topology is not in a valid state for patch meshing. Patches are composed of four curves and appear in green when the topology is valid. To generate proper patches, split some of the current patch curves:

5. In the *Create/Edit Patch Entities* box, click (**Split the selected patch curve**) and maintain the split patch curve option, **At Picked Location**.

6. In the *Graphics* window, following the number points shown below, click the existing feature curves.

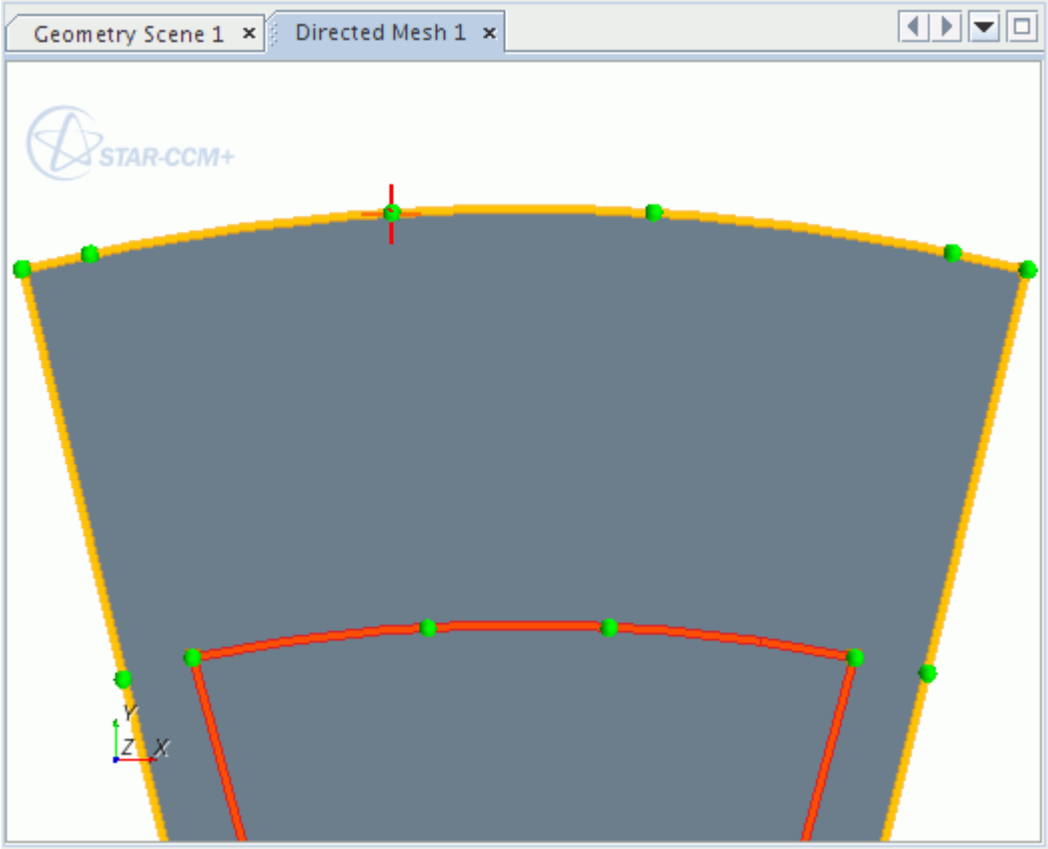


Note: you can move vertices at any point in the patch meshing process.

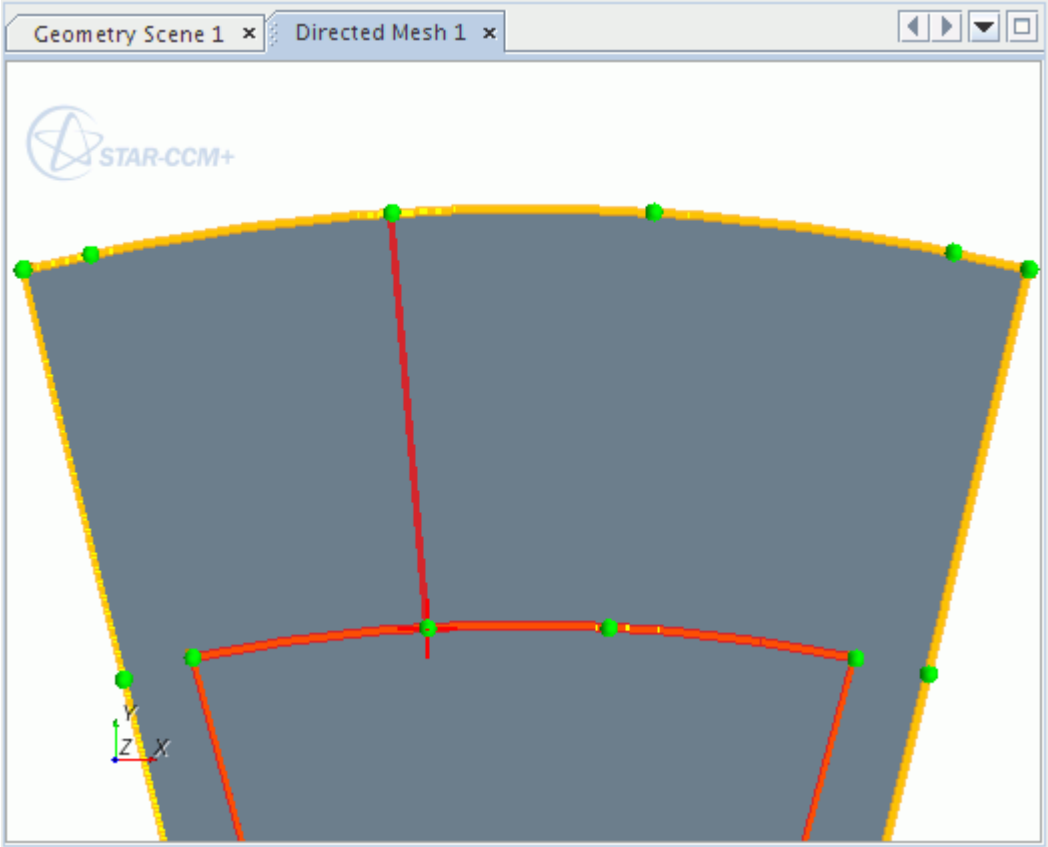
Connect these additional vertices to form complete patches.

7. In the *Create/Edit Patch Entities* box, click (**Create patch curve**).
8. Click the vertex point marked “2” in the previous screenshot.

   The red lines passing through the point indicate that the point has the “focus”, meaning that the next curve starts from this point.



9. Click the vertex point “6”. A patch curve is created to connect the two points.



10. Continue to create additional patch curves between vertices as shown in the following screenshot.

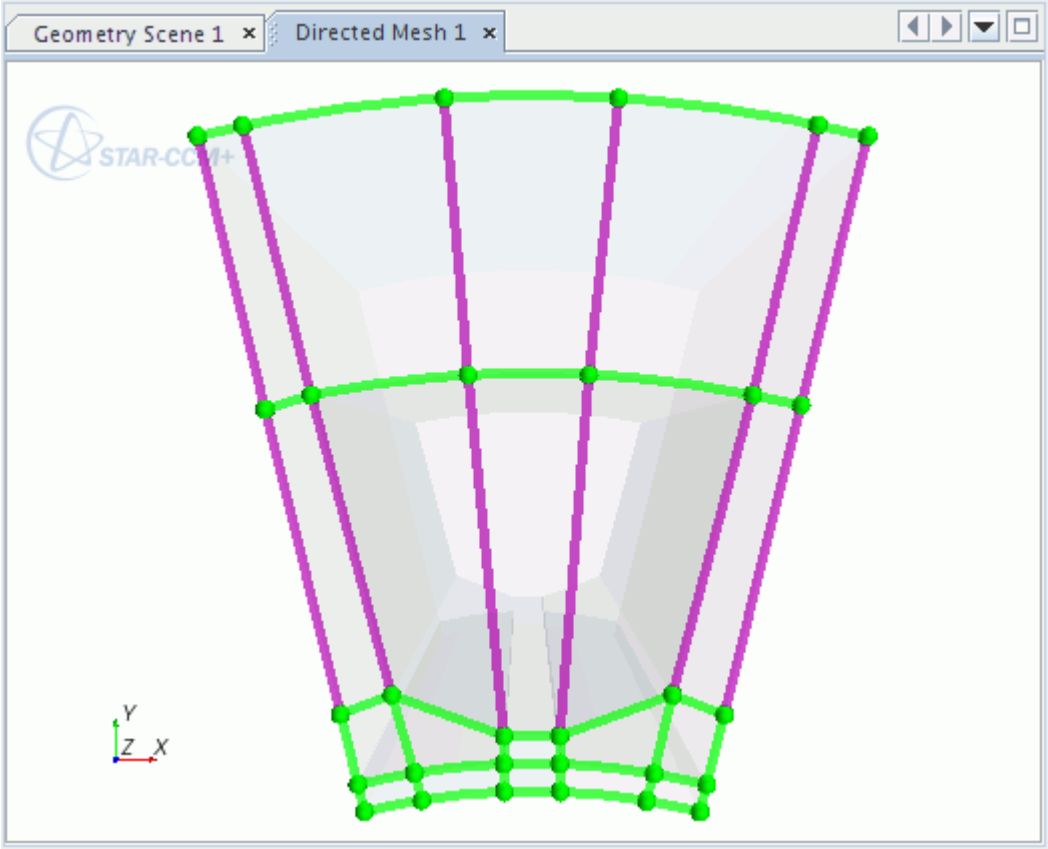
    To change the focus point as you create the curves, click a blank area in the *Graphics* window to deselect an old focus point before selecting a new focus point.

    Note: If you make a mistake in creating a curve, you can delete a patch curve.

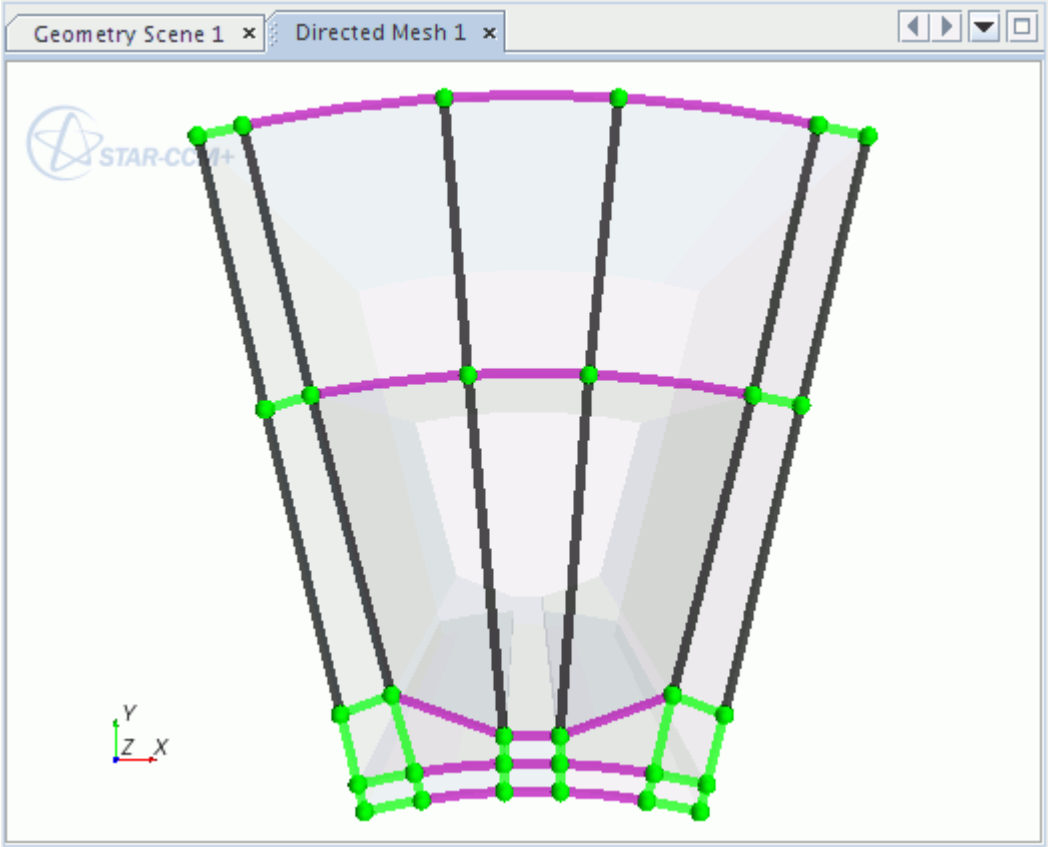
Creating a Patch Mesh

Once a patch topology is fully defined, STAR-CCM+ enables the Patch Mesh mode. You define the mesh distribution properties of the patch mesh within this mode.

1. In the *Edit* panel, set **Mode** to **Patch Mesh**.
2. In the *Graphics* window, multi-select the set of curves that are shown below.



3. In the *Mesh Properties* box, set **Number of Divisions** to `14` and click **Apply**.
4. Select a second set of patch curves as shown below. The patch curves for which mesh properties have been defined appear in black.



5. In the *Mesh Properties* box, set **Number of Divisions** to `10` and click **Apply**.
6. Select the remaining short patch curves. Set **Number of Divisions** to `4` and click **Apply**.

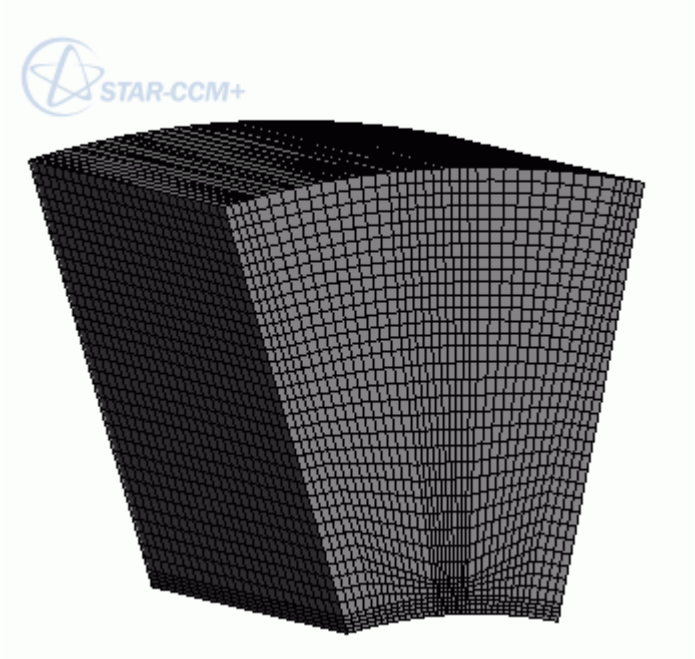
   The source mesh is shown in the *Graphics* window.
7. **Close** the *Edit* panel.

Generating a Volume Mesh

The next step is to generate a volume mesh using the patch mesh. This step is done by defining a volume mesh distribution for the connected part.

1. Right-click the **Mesh Distributions** node and select **New Volume Distribution**.
2. In the *Parts for creating a mesh distribution* dialog, select **Connected Part** and click **OK**.
3. Select the **Mesh Distributions** > **Volume Distribution** > **Default Controls** > **Number of Layers** node and set **Number of Layers** to `24`.
4. Right-click the **Directed Mesh** node and select **Execute**.

   The meshed parts are shown in the *Graphics* window.



5. To return to the simulation tree, click **Close Directed Mesh** .

Using Existing Meshes

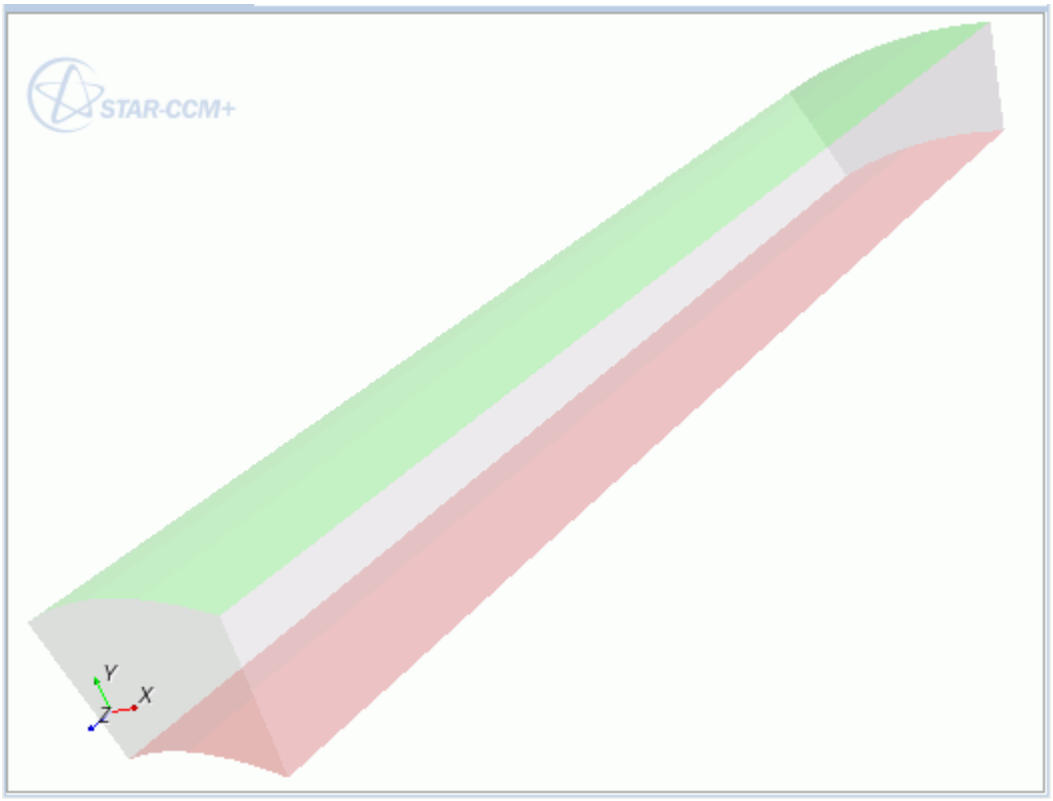
Use the volume mesh that is generated from the patch topology to generate a source mesh for the remaining Magnet and Rotor parts.

Note that the Air, Magnet and Rotor parts have contacts defined between them.

1. As with patch meshing, right-click the **Geometry** > **Operations** > node and select **New** > **Mesh** > **Directed Mesh**.
2. In the *Create Directed Mesh Operation* dialog, select the **Magnet** part and click **OK**.

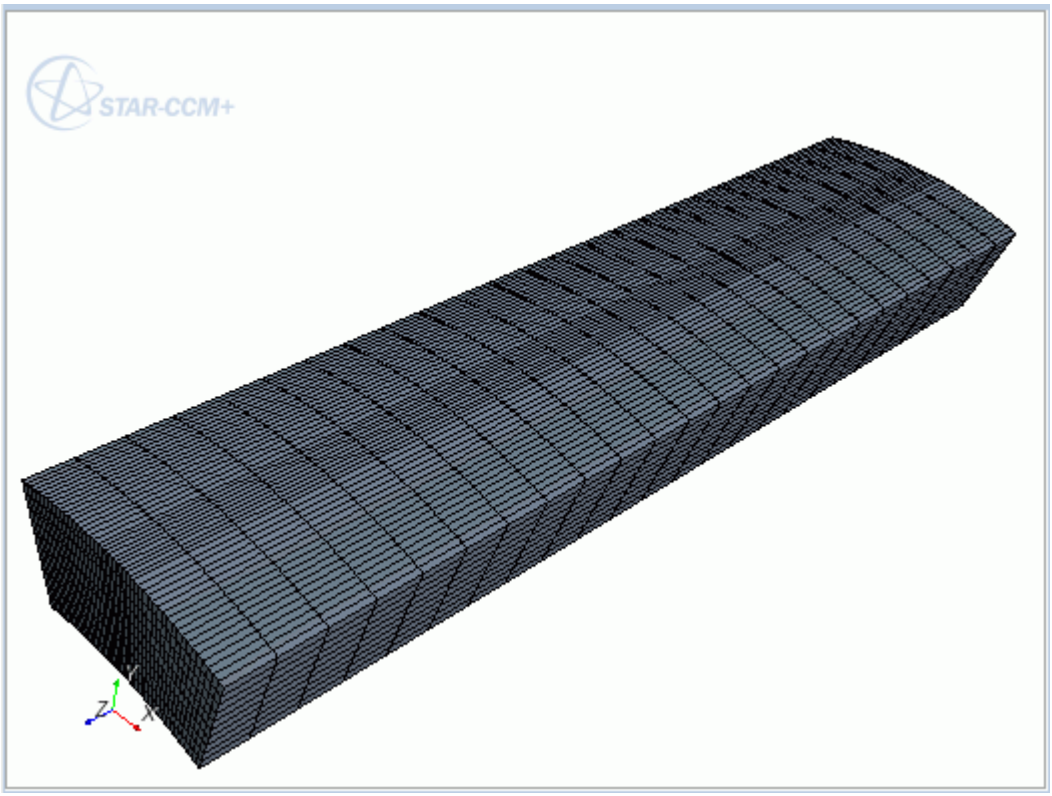
   Only the Magnet and Rotor parts appear in the dialog as the other three parts are already associated with a directed mesh.
3. Right-click the **Operations** > **Directed Mesh 2** node and select **Edit**.
4. In the *Directed Mesh 1* window:
   a) Select the top surface of the Magnet part (Magnet:source), right-click the selection, and choose **Part Surface (source)** > **Add to Source**.
   b) Select the bottom surface of the Magnet part (Magnet:target), right-click the selection, and choose **Part Surface (target)** > **Add to Target**.

   The source surface is displayed in green and the target surface in red.



5. Right-click the **Source Meshes** node and select **New Source Mesh** > **Use Existing Mesh**.
6. In the *Part Collection For This Source Mesh* dialog, select **Magnet**.

   STAR-CCM+ maps the base of the volume mesh for the Air part to the source surface of the Magnet part. It adds an Existing Mesh node, **Magnet**, to Mesh Settings.
7. Right-click the **Mesh Distributions** node and select **New Volume Distribution**.
8. In the *Parts for creating a mesh distribution* dialog, select **Magnet** and click **OK**.
9. Select the **Mesh Distributions** > **Volume Distribution** > **Default Controls** > **Number of Layers** node and set **Number of Layers** to `12`.
10. Right-click the **Directed Mesh 2** node and select **Execute**.

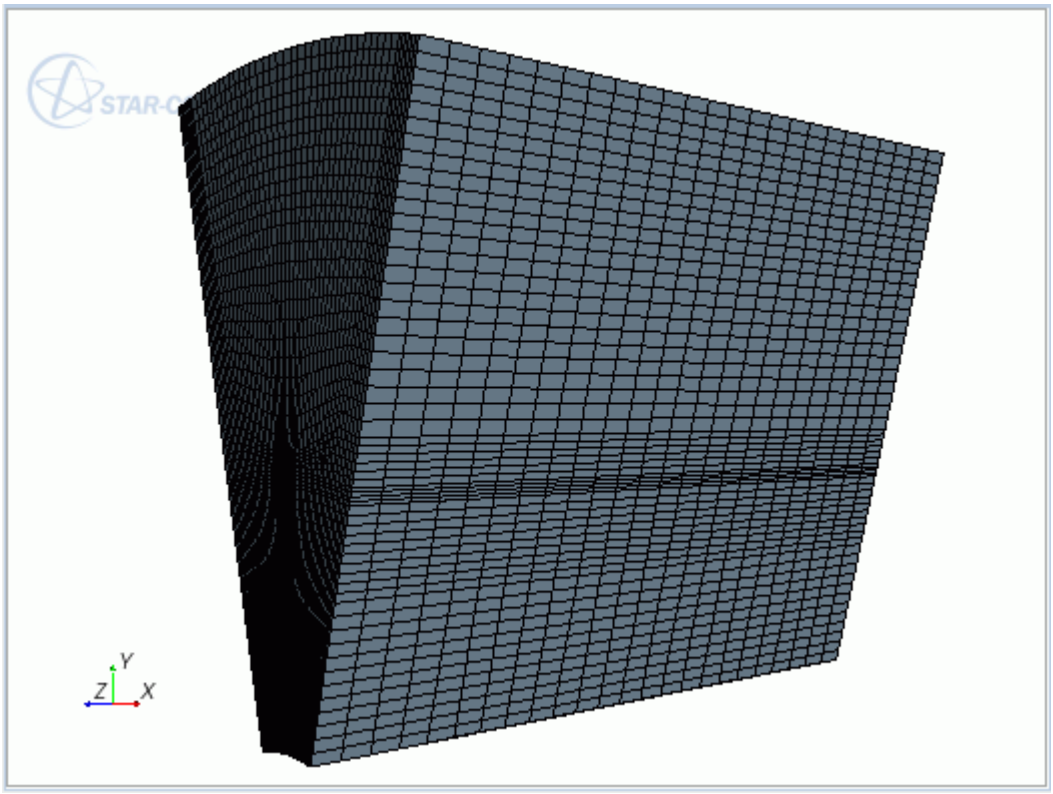


11. Click **Close Directed Mesh** to return to the simulation tree.

Repeat the process for the Rotor part:

12. Create a Directed Mesh operation.
13. Edit the **Directed Mesh 3** node and select the source and target surfaces.
14. Create a source mesh using an existing mesh. Select the **Rotor** part as the part collection for the source mesh.
15. Create a volume distribution and set the **Number of Layers** to `12`.
16. Execute the directed mesh operation.
17. When the volume mesh for the Rotor part is generated, click **Close Directed Mesh**.
18. Create a Mesh Scene.

The final mesh is displayed below:



Summary

This tutorial demonstrated how to use the directed meshing tool in STAR-CCM+.

The steps covered were:

- Checking the suitability of CAD geometry for the directing mesh operation.
- Launching a directed mesh operation.
- Identifying source and target surfaces.
- Creating a patch topology for the source surface mesh.
- Using a previously defined volume mesh for the source surface mesh.
- Defining a volume mesh distribution and generating a volume mesh.

Incompressible Flow

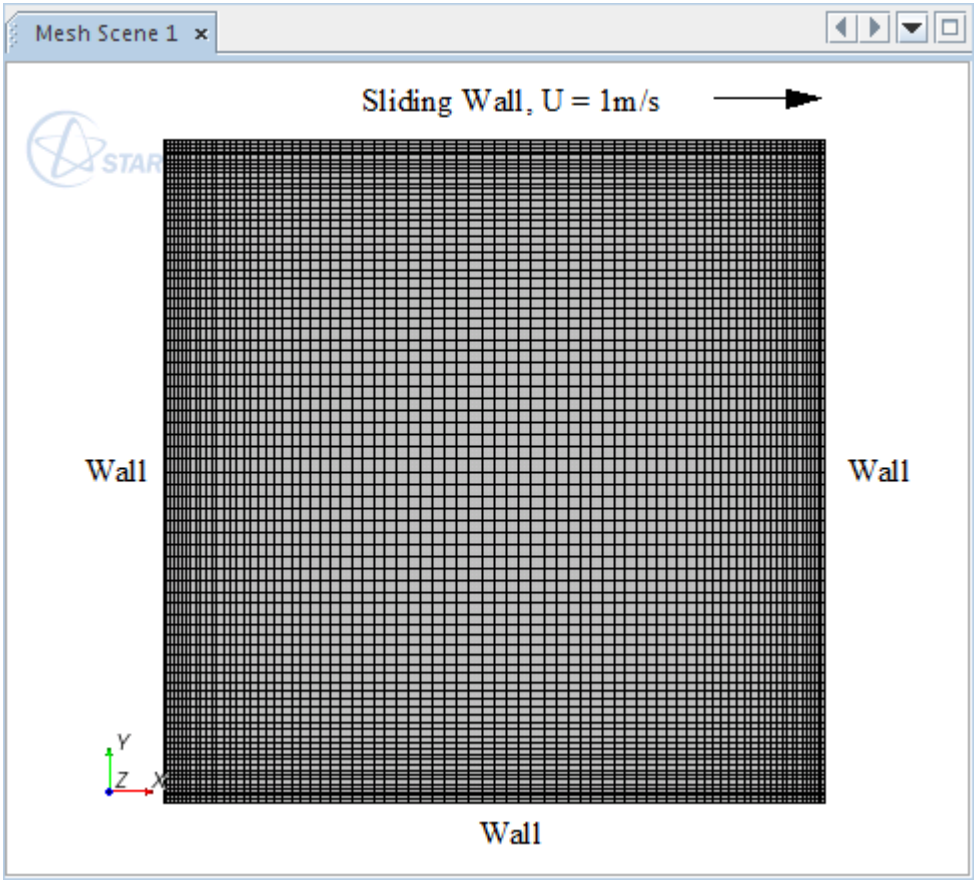
The tutorials in this set illustrate various STAR-CCM+ features for incompressible fluid flows as well as porosity and solution recording

Contents:

Steady Flow: Lid-Driven Cavity Flow
Steady Flow: Channel Flow with Multiple Meshes
Steady Flow: Laminar and Turbulent in an S-Bend
Steady Flow: Backward Facing Step
Adjoint Flow Solver: External Flow over a Dual Element Wing
Adjoint Flow: Wing Shape Optimization
Porous Resistance: Isotropic Media
Porous Resistance: Orthotropic Media
Solution Recording and Playback: Vortex Shedding
Generalized Newtonian Fluid: Flow in a Static Mixer
Viscoelastic Flow: Basic Extrusion

Steady Flow: Lid-Driven Cavity Flow

This tutorial demonstrates the performance of STAR-CCM+ in solving a traditional square lid-driven cavity flow. The problem geometry consists of a two-dimensional 1 m^2 cavity. The cavity is covered by an impermeable wall that moves in the x-direction with constant velocity of 1 m/s. A stretched quadrilateral mesh with 6400 cells is used.



Material properties:

Density (kg/m^3)	1.0
Dynamic viscosity (Pa.s)	0.0002

Boundary conditions:

Top wall	U = 1.0 (m/s), V=0.0
Bottom wall	U = V = 0.0
Left wall	U = V = 0.0
Right wall	U = V = 0.0

Initial conditions:

U, V velocity components (m/s)	0.0
Operating pressure (Pa)	0.0

The Reynolds number based on the width of the cavity, top wall velocity, fluid density and fluid dynamic viscosity is 5000. The calculation is performed using the Segregated Flow model. Both the U- and V-velocity profiles along lines passing through the center of the cavity are compared with data from the literature [Ghia].

Contents:

Importing the Mesh and Naming the Simulation
Visualizing the Imported Mesh
Setting Up the Physics Models
Setting Fluid Properties
Setting Boundary Conditions and Values
Visualizing the Solution
Setting Up Lines for Plotting
Plotting Simulation Data
Plotting Reference Data
Preparing Solver Parameters and Stopping Criteria
Initializing and Running the Simulation
Analyzing the Results
Summary
Bibliography

Importing the Mesh and Naming the Simulation

Create a new simulation and set up the mesh.

A quadrilateral-cell mesh has been prepared for this analysis and is stored in a `.ccm` format file. Import it into the simulation:

1. Start up STAR-CCM+ in a manner that is appropriate to your working environment and select the New Simulation option from the menu bar. .

   STAR-CCM+ creates a window for the new simulation in the *Explorer* pane and give it the default name Star 1
2. Select **File** > **Import** > **Import Volume Mesh** from the menu bar.
3. In the *Open* dialog, navigate to the `doc/startutorialsdata/incompressibleFlow/data` subdirectory of your STAR-CCM+ installation directory and select the file `cavityQuad.ccm`.
4. Click the **Open** button to start the import.

   STAR-CCM+ provides feedback on the import process in the *Output* window. A mesh region that is named **cell-2** is created under the **Regions** node to represent the solution domain.
5. Save the new simulation as `cavityQuad.sim`.

Visualizing the Imported Mesh

View the imported mesh, and access various parts that are contained in the region.

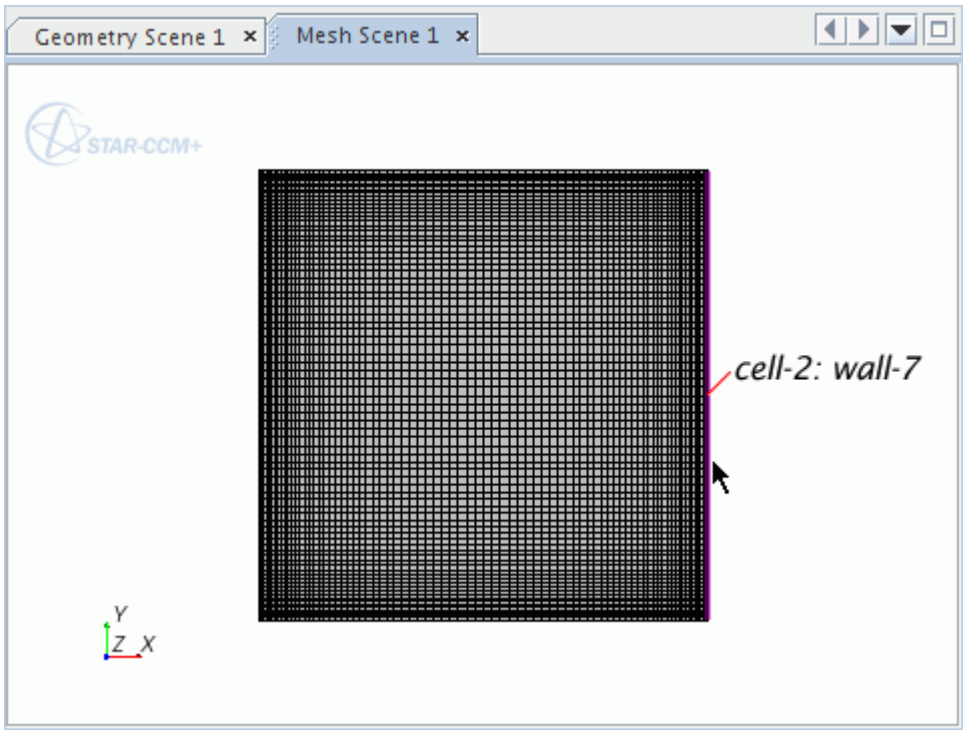
Use the following steps:

1. Right-click the **Scenes** node and then select **New Scene** > **Mesh**.

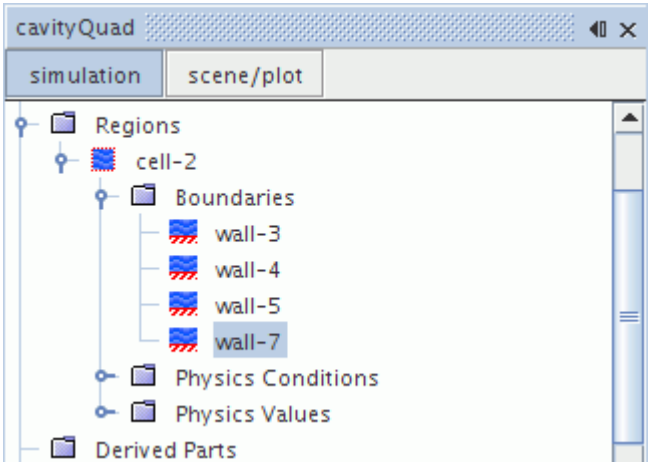
   The mesh appears in a scene in the *Graphics* window.

2. The edges of the square are boundaries. Click one of them.

   The selected boundary is highlighted and a label with its name appears on the display.



In the *cavityQuad* window, the node that corresponds to this boundary is also highlighted.



Similarly, if any other boundary node is selected in the *cavityQuad* window, the part corresponding to that boundary is highlighted in the *Graphics* window.

Setting Up the Physics Models

The default continuum (**Physics 1**) is edited so that the appropriate physical models for the simulation are selected.

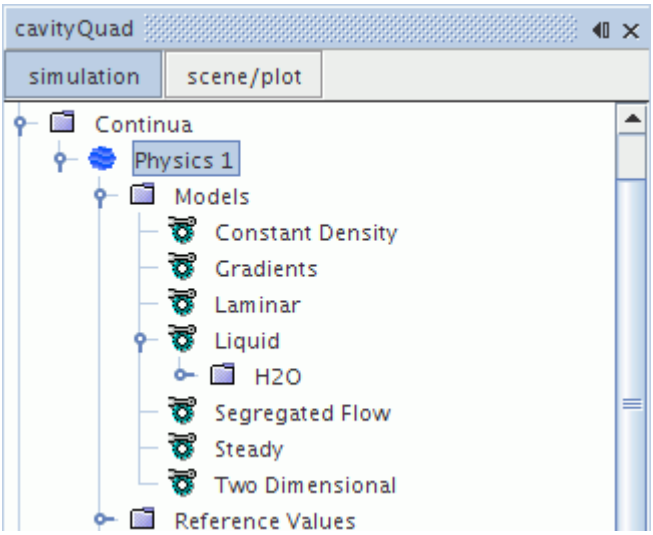
To select physical models for this continuum:

1. For the physics continuum **Continua** > **Physics 1**, select the following models, in order:

Group Box	Model
Enabled Models	**Two Dimensional** (Pre-selected)
Material	**Liquid**
Flow	**Segregated Flow**
	Gradients (Selected automatically)
Equation of State	**Constant Density**
Time	**Steady**
Viscous Regime	**Laminar**

2. Click `Close`.

   The color of the **Physics 1** node has turned from gray to blue to indicate that models have been activated.



3. Save the simulation.

Setting Fluid Properties

Set the density and dynamic viscosity fluid properties.

Use the following steps:

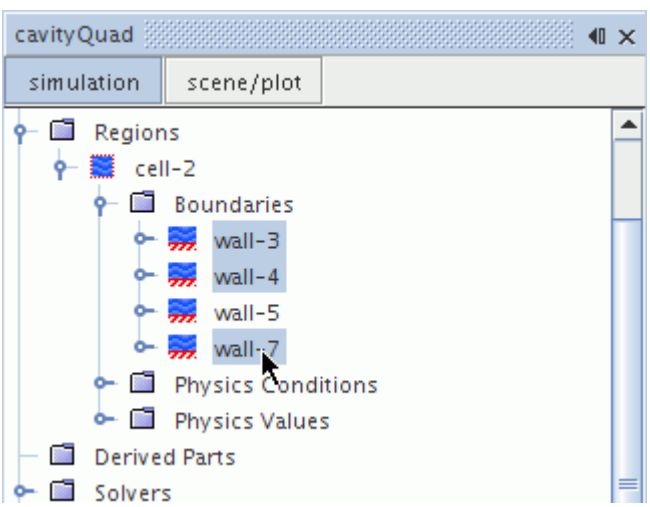
1. Expand the **Continua** > **Physics 1** > **Models** > **Liquid** > **H2O** > **Material Properties** > **Density** node.
2. Select the **Constant** node and set the *Value* to `1.0 kg/m^3`.
3. Select the **H2O** > **Material Properties** > **Dynamic Viscosity** > **Constant** node and set the *Value* to `2.0E-4 Pa-s`.

Setting Boundary Conditions and Values

This problem requires only two boundary conditions: moving and stationary. In STAR-CCM+, it is possible to combine two or more boundaries into one.

Use the following steps:

1. Open the **Regions** > **cell-2** > **Boundaries** node.
2. Multi-select the boundary nodes **wall-3**, **wall-4**, and **wall-7**.



3. Right-click and select **Combine**.

   This creates a single boundary, wall-3, that consists of all sides of the square except the top.
4. Rename the **wall-3** node to `Stationary Wall`.
5. Rename the **wall-5** node to `Moving Wall`.

For the moving wall boundary, set the velocity of 1 m/s in the x-direction:

6. Select the **Moving Wall** > **Physics Conditions** > **Tangential Velocity Specification** node and set the *Method* to be `Vector`.
7. Select the **Physics Values** > **Relative Velocity** node and enter `1,0,0` in the *Value* property.
8. Save the simulation.

Visualizing the Solution

Post-processing scenes can be updated every iteration (or time-step) in STAR-CCM+. This allows for dynamic monitoring of various quantities, such as the velocity that you want to visualize in this tutorial. A vector scene is therefore created before running.

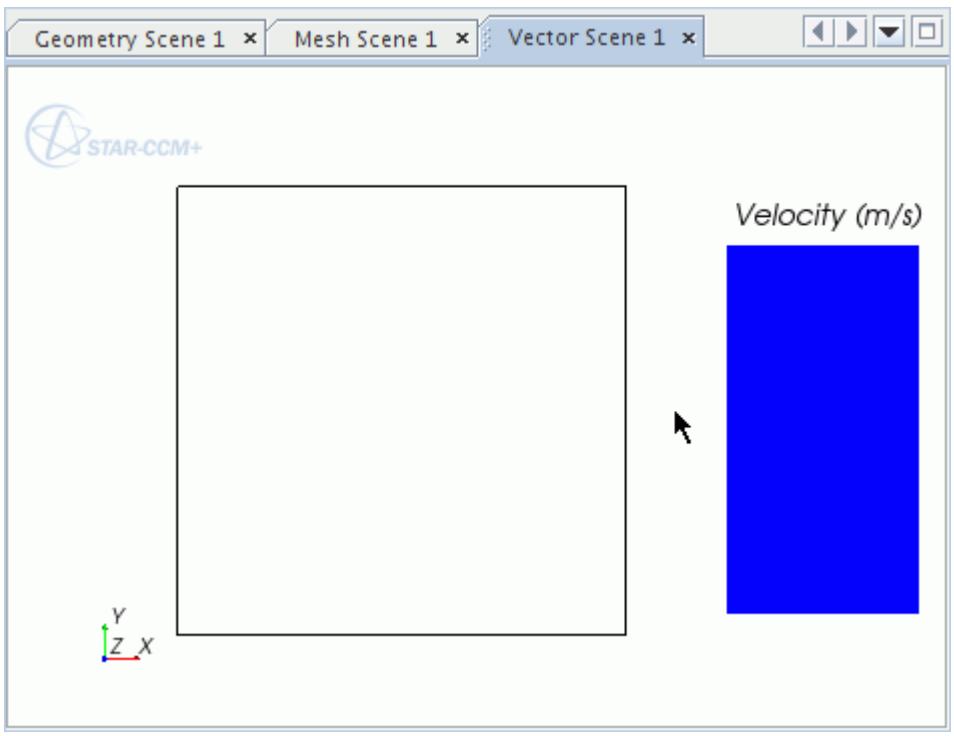
Use the following steps:

1. Right-click the **Scenes** node and then select **New Scene** > **Vector**.

   A new *Vector Scene 1* display appears.

The vector bar shows the default setting of `Velocity`.

2. Click and drag the vector bar toward the right side of the display until it switches to a vertical orientation.



3. The edges of the bar can be dragged to make the label larger. Move the square to make more room for the vector bar.
4. To verify that the scene has the entire region selected for the data, select the **Scenes** > **Vector Scene 1** > **Displayers** > **Vector 1** > **Parts** node.

The *Parts* property shows `cell-2`, the name of the region.

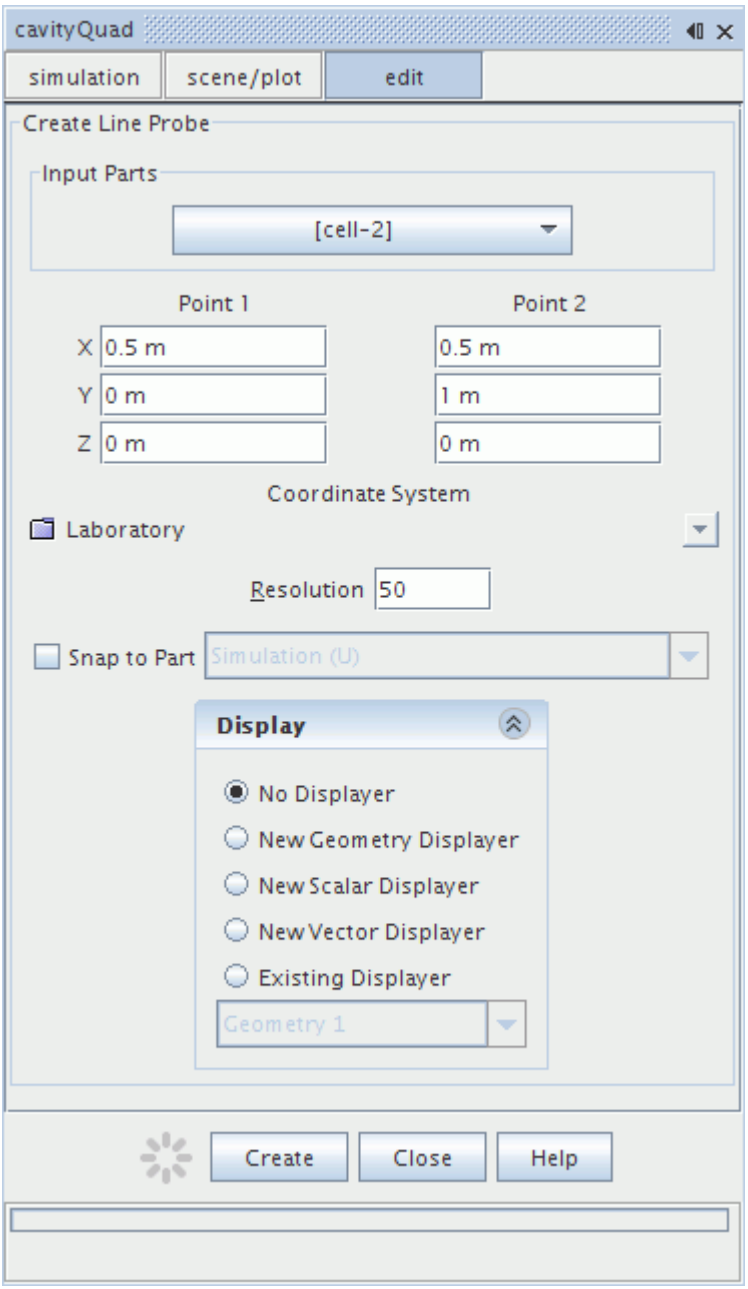
Setting Up Lines for Plotting

Data are plotted for both the U- and V-velocity profiles along line probes passing through the center of the cavity.

To create these lines:

1. Make sure that at least one scene display is active in the *Graphics* window.
2. Right-click the **Derived Parts** node and select **New Part** > **Probe** > **Line**

   The *Create Line Probe* dialog appears in the *Explorer* pane to specify the desired line.



3. Set the following properties for the U line:

Property	Value
Input Parts	`cell-2`
Point 1	`[0.5, 0.0, 0.0]`

Point 2	`[0.5, 1.0, 0.0]`
Resolution	`50`
Display	**No Displayer**

4. Click **Create**, then click **Close**.

   A new derived part named **line-probe** is created in the **Derived Parts** manager node.
5. Rename the node to `Simulation (U)`

Next, create the V line probe. This can be done simply by making a copy of the existing derived part and modifying it.

6. Right-click the **Simulation (U)** node and select **Copy**.
7. Right-click the **Derived Parts** node and select **Paste**.
8. Rename the new node to `Simulation (V)`
9. Enter `0`, `0.5`, `0` for the *Point 1* property.
10. Enter `1`, `0.5`, `0` for the *Point 2* property.
11. Save the simulation.

Plotting Simulation Data

This part of the tutorial produces the plots of the U- and V-velocity profiles. In a subsequent step, experimental data are imported from files and plotted alongside the numerical data. The development of the solution can be observed throughout the run by viewing these plots.

To create the U-velocity plot:

1. Right-click the **Plots** node and select **New Plot** > **X-Y Plot**.
2. Rename the **XY Plot 1** node `U-Velocity Profile`
3. Set the following properties:

Node	Property	Value
U-Velocity Profile	*Parts*	**Derived Parts** > **Simulation (U)**
└ **X Type**	*Type*	**Direction**
└ **Vector Quantity**	*Value*	`[0,1,0]`
└ **Y Types** > **Y Type 1**	*Type*	**Scalar**
└ **Scalar Function**	*Field Function*	**Velocity** > **Lab Reference Frame** > **Laboratory** > **i**
└ **Simulation (U)** > **Line Style**	*Style*	———— (**Solid**)

Node	Property	Value
Symbol Style	*Shape*	`None`
Axes > **Left Axis**	*Minimum*	`-0.6`
	Maximum	`1.0`
Major Labels	*Spacing*	`0.2`

To create the V-velocity plot:

4. Right-click the **Plots** node and select **New Plot** > **X-Y Plot**.
5. Rename this **XY Plot 1** node `V-Velocity Profile`
6. Select the `V-Velocity Profile` node and click the right-hand side of the *Parts* property.

Node	Property	Value
V-Velocity Profile	*Parts*	**Derived Parts** > **Simulation (V)**
Y Types > **Y Type 1**	*Type*	**Scalar**
Scalar Function	*Field Function*	**Velocity** > **Lab Reference Frame** > **Laboratory** > **j**
Simulation (V) > **Line Style**	*Style*	———— (**Solid**)
Symbol Style	*Shape*	`None`
Axes > **Left Axis**	*Minimum*	`-0.7`
	Maximum	`0.5`
Major Labels	*Spacing*	`0.2`

7. Save the simulation.

Plotting Reference Data

To compare the experimental velocity data with the velocity plots, it is necessary to import the data in supplied files into STAR-CCM+ tables.

Use the following steps:

1. Open the **Tools** node.
2. Right-click the **Tables** node and select **New Table** > **File Table**.

3. In the *Open* dialog navigate to the `doc/startutorialsdata/incompressibleFlow/data` subdirectory of your STAR-CCM+ installation directory, set the File Type drop-down list to `(*.xy)`, and select the file `UVelocity.xy`.
4. Click **Open**.

   The data in that file is now stored in a table named **UVelocity**.

To display the data on the U-velocity plot:

5. Return to the **U-Velocity Profile** node.
6. Right-click the **U-Velocity Profile** > **Data Series** node and select **Add Data**.
7. In the *Add Data Providers to Plot* dialog, select **UVelocity** and click **OK**.
8. Select the **U-Velocity Profile** > **Data Series** > **UVelocity** node and set *Legend Name* to `Reference Data (U)`.
9. Make sure that `column 0` is set for the *X column* and `column 1` is set for the *Y column*.
10. Select the **UVelocity** > **Line Style** node and make sure that the *Style* property is set to `None`.
11. Select the **Symbol Style** node and set the following properties:

Property	Value
Size	`6`
Shape	`Filled Triangle`

12. Click (**Custom Editor**) for the *Color* property. In the dialog that appears:
    a) Select the **RGB** tab.
    b) Set *Red* to `0`, *Green* to `0`, and *Blue* to `255`.
13. Click **OK**.

    The supplied experimental data appears in the U-velocity plot. Numerical data is shown once the simulation run has started.

Use the same techniques for the V-velocity plot:

14. Right-click the **Tables** node and select **New Table** > **File Table**.
15. In the *Open* dialog, select the file `VVelocity.xy` and click **Open**.
16. In the **V-Velocity Profile** > **Data Series** node, add data and select **VVelocity** in the *Add Data Providers to Plot* dialog.
17. Select the **V-Velocity Profile** > **Data Series** > **VVelocity** node and set *Legend Name* to `Reference Data (V)`.
18. Make sure that `column 0` is set for the *X column* and `column 1` is set for the *Y column*.
19. Select the **VVelocity** > **Line Style** node and make sure that the *Style* property is set to `None`.
20. Select the **Symbol Style** node and set the following properties:

Property	Value
Size	`6`
Shape	`Filled Triangle`

21. Click ... (**Custom Editor**) for the *Color* property. In the dialog that appears:
    a) Select the **RGB** tab.
    b) Set *Red* to `0`, *Green* to `0`, and *Blue* to `255`.
22. Click **OK**.

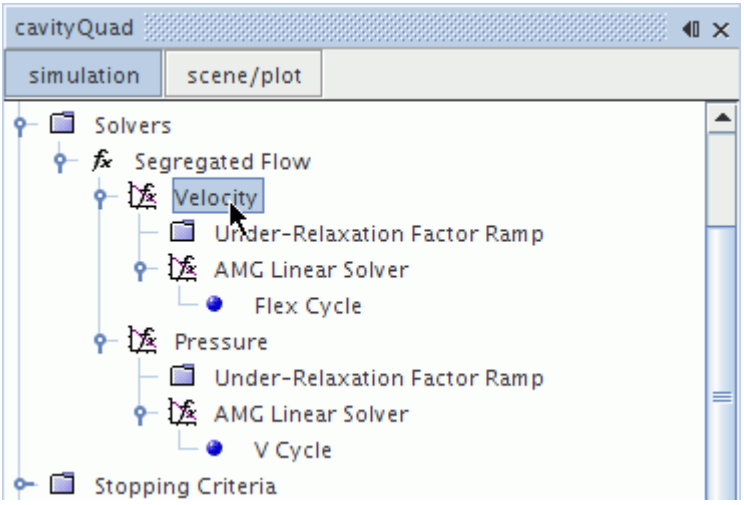
    Both plots show experimental data alongside the numerical data.
23. Save the simulation.

Preparing Solver Parameters and Stopping Criteria

Before running the analysis, look at the velocity solver, as well as the maximum number of iterations to be performed.

Use the following steps:

1. Select the **Solvers** > **Segregated Flow** > **Velocity** node and make sure that the *Under-Relaxation Factor* is `0.7`.



2. Select the **Stopping Criteria** > **Maximum Steps** node and make sure that the *Maximum Steps* value is `1000`.

Initializing and Running the Simulation

Initialize the solution and run the analysis.

Use the following steps:

1. Click (**Initialize Solution**) in the Solution toolbar or use the **Solution** > **Initialize Solution** menu item. The initial zero velocity condition is shown in the plot displays.
2. Click (**Run**) in the Solution toolbar to run the analysis.

The Residuals display is created automatically and shows the progress of the solver. If necessary, click the *Residuals* tab to bring the Residuals plot into view. An example of a residual plot is shown in a separate part of the User Guide.

During the run, it is possible to stop the process by clicking (**Stop**) in the toolbar. The simulation can be continued later by clicking (**Run**). If left alone, the simulation continues until the stopping criteria are satisfied.

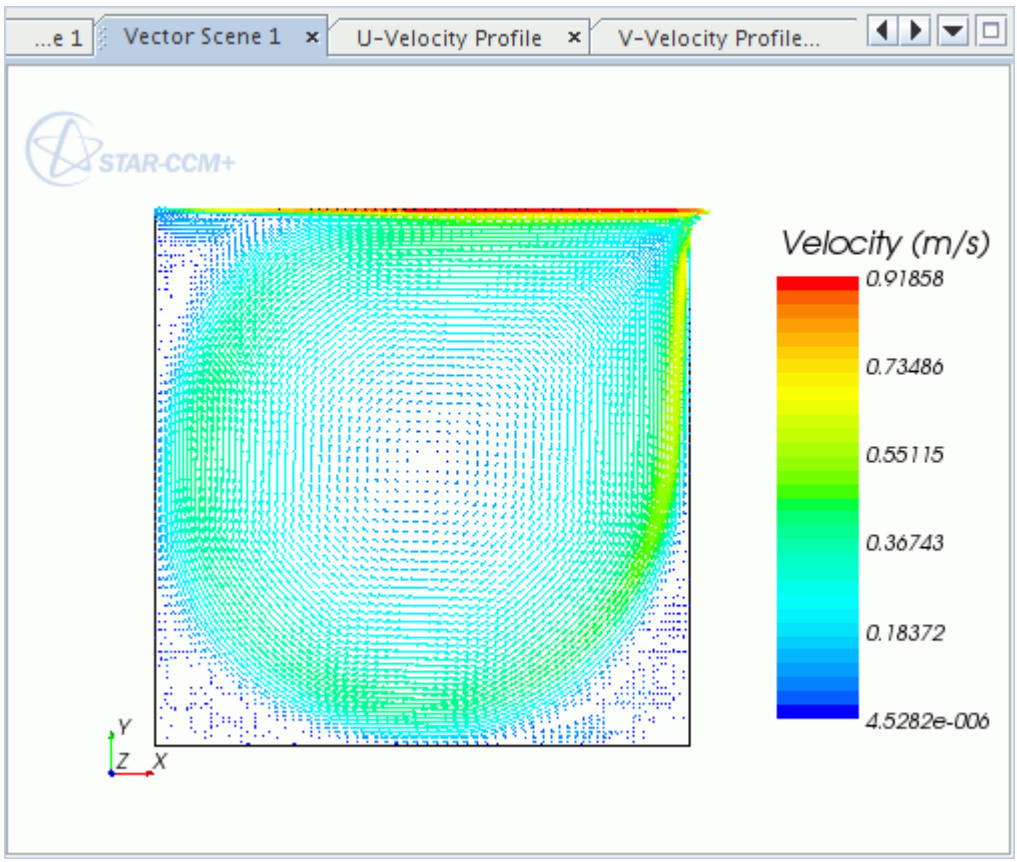
3. After the run is finished, save the simulation.

Analyzing the Results

Once the calculations are finished, the displays show the results based on the pre-processing settings.

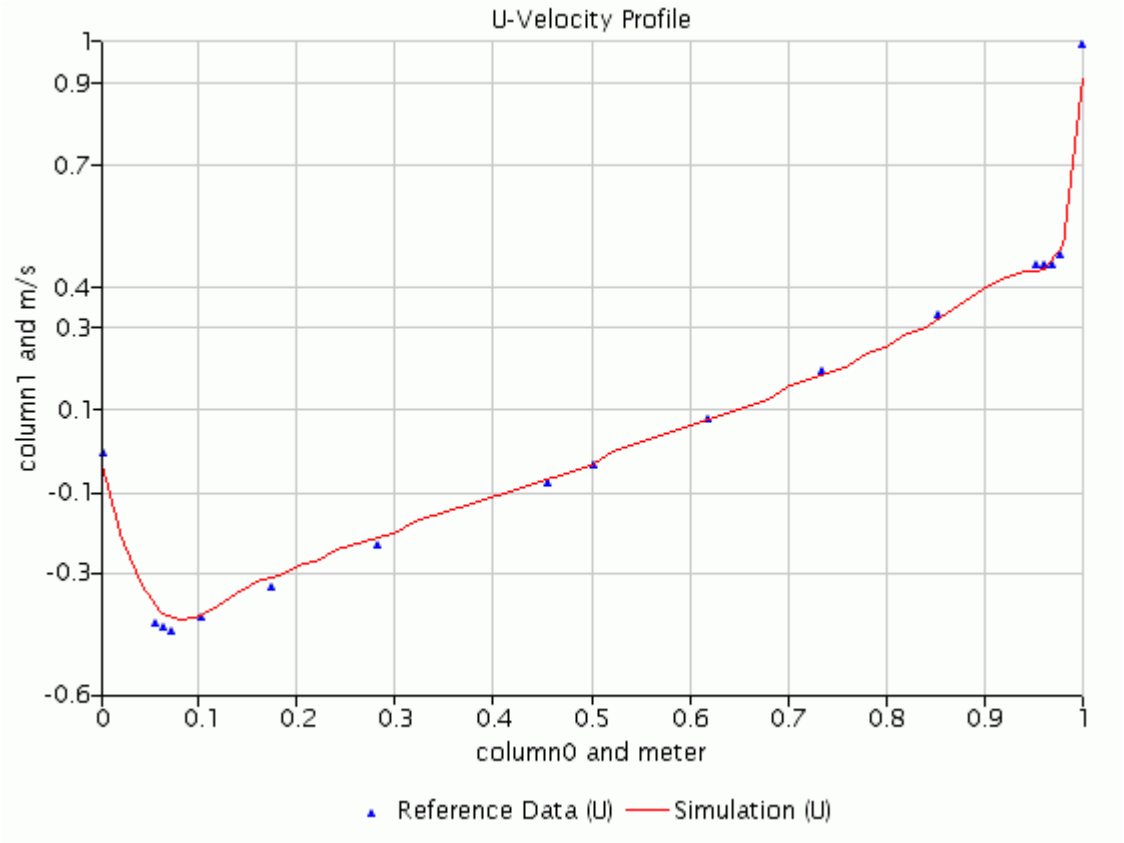
Use the following steps:

1. Click the *Vector Scene 1* display tab to view the vectors.

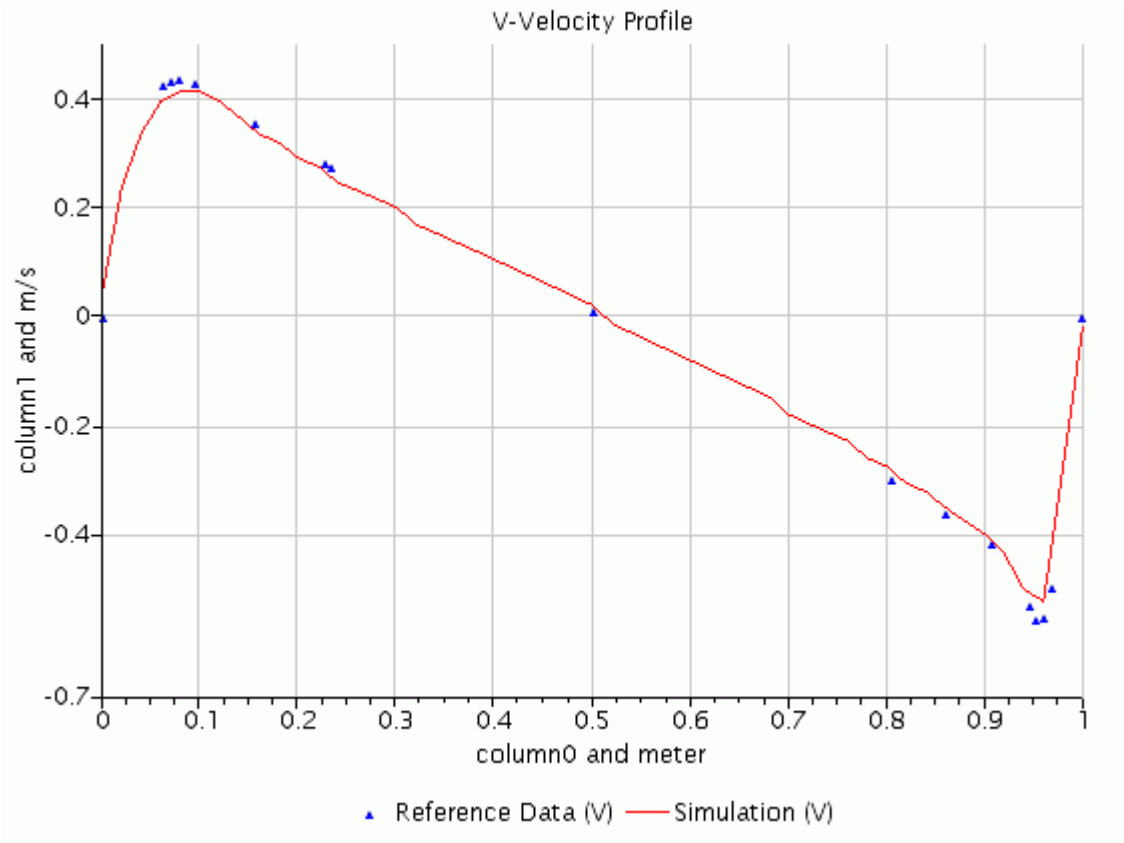


The plots also contain numerical results that can be compared to the experimental data:

2. Click the *U-Velocity Profile* display tab:



3. Click the *V-Velocity Profile* display tab:



Summary

This tutorial solved a traditional square lid-driven cavity flow.

The tutorial demonstrated the following STAR-CCM+ features:

- Defining physical models.
- Defining material properties that are required for the selected models.
- Defining boundary conditions.
- Reviewing solver parameters and stopping criteria.
- Monitoring the solution progress.
- Running the solver until the residuals are satisfactory.
- Analyzing results using STAR-CCM+'s visualization and XY-plot facilities.

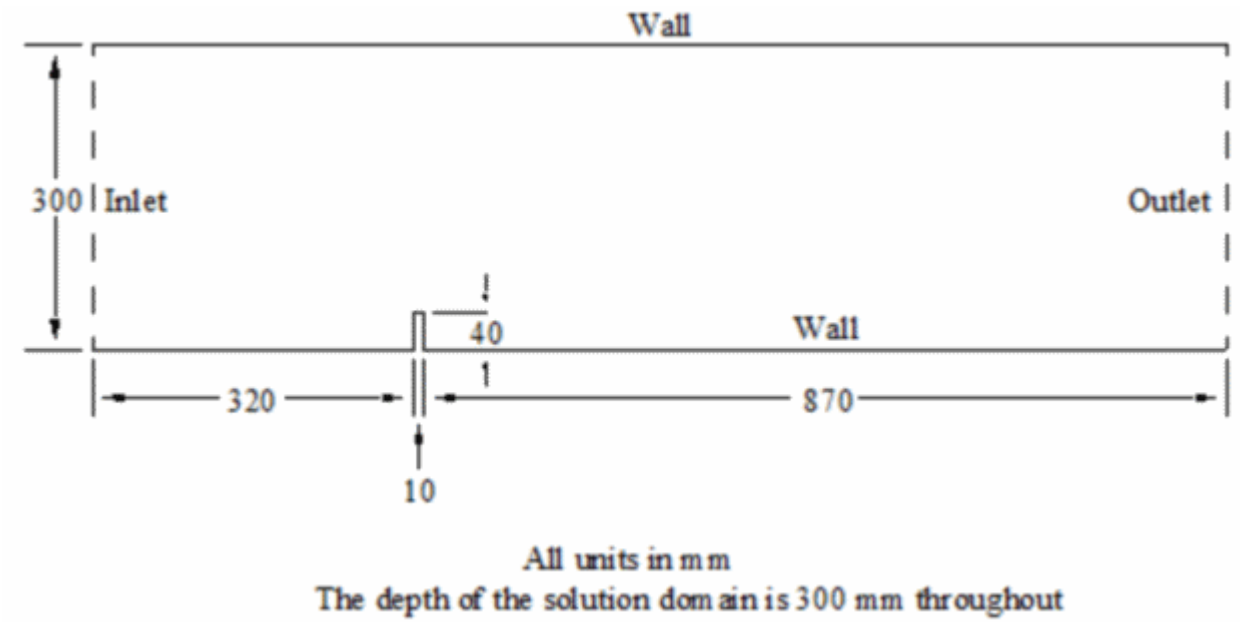
Bibliography

Ghia, U., Ghia, K.N., and Shin, C.T. 1982. "High-Re solution for incompressible flow using the Navier-Stokes equations and a multigrid method", *J. Comp. Phys.*, **48**, p. 387.

Steady Flow: Channel Flow with Multiple Meshes

This tutorial describes a steady airflow simulation over an obstacle that is mounted at the bottom wall of a three-dimensional channel.

The problem geometry is shown below.



The case is based on an investigation from the Danish Maritime Institute, whose experimental data is compared with the STAR-CCM+ predictions. In the experimental set-up, the flow velocity and turbulence characteristics were non-uniform across the inlet section. However, to simplify the modeling procedure, the case that is presented here is based on a parabolic inlet velocity profile with fixed turbulence characteristics.

The boundaries consist of an inlet, an outlet, a symmetry plane on one of the two ‘side’ faces and default no-slip walls. The fluid is air and its physical properties are assumed constant and equal to the STAR-CCM+ default values.

Air enters the solution domain at standard pressure and temperature (1 bar and 293 K) with a free-stream velocity of 1.17 m/s. Based on the height of the obstacle, the Reynolds number is 3,115. The turbulence kinetic energy and dissipation rate at the inlet are set to 0.024 m^2/s^2 and 0.07 m^2/s^3, respectively, these values having been derived from experimental data. All fluid mass entering the solution domain exits through the outlet. The flow is isothermal, turbulent, and incompressible; turbulence is simulated using the standard linear k-e model that is combined with the Wolfstein two-layer model.

The same problem is modeled three times, each variant using a different mesh, as described in:

- Hexahedral Mesh
- Tetrahedral Mesh
- Polyhedral Mesh

Comparisons are then drawn between the accuracy and computational speed of each type of mesh. All three meshes are based on the same initial shell surface defining the problem geometry but each was generated using a different technique.

Contents:

Hexahedral Mesh
Tetrahedral Mesh
Polyhedral Mesh
Summary

Hexahedral Mesh

The first mesh for this tutorial is hexahedral. Create a new simulation and import the mesh.

Use the following steps:

1. Create a directory for the tutorial called `meshComparison`.
2. Start up STAR-CCM+ in a manner that is appropriate to your working environment and select the **New Simulation** option from the menu bar.

Continue by importing the hexahedral mesh and naming the simulation. The grid has been saved in the `.ccm` file format, which contains all necessary cell, vertex, and boundary information for the problem geometry.

3. Select **File** > **Import** > **Import Volume Mesh** from the menu bar.
4. In the *Open* dialog, navigate to the `[INSTALL_DIR]/doc/startutorialsdata/incompressibleFlow/data` subdirectory of your STAR-CCM+ installation directory and select file `hex.ccm`.
5. Click **Open** to start the import.

STAR-CCM+ provides feedback on the import process, which takes a few seconds, in the *Output* window.

6. Open a geometry scene in the *Graphics* window.
7. Finally, save the new simulation under file name `hexMesh.sim`.

Contents:

Creating a Macro
Setting Up the Physics Models
Setting Initial Conditions
Setting Boundary Conditions and Values
Setting Stopping Criteria
Visualizing the Solution
Setting Up Line Surfaces for Plotting
Plotting Simulation Data
Initializing and Running the Simulation
Visualizing the Results
Examining Plots

Exporting Simulation Data

Creating a Macro

Record a macro file to avoid having to repeat the steps. Use this macro later to set up the cases for the other two meshes.

The Tetrahedral Mesh and Polyhedral Mesh parts of this tutorial that follow require the same problem set-up.

1. Begin recording the macro by clicking (**Start Recording**) in the Macro toolbar.
2. In the *Save* dialog, enter `setup.java` as the **File Name** and click **Save**.

All actions you perform from now until the recording is stopped are recorded in the macro.

Note: Do not save the simulation while the macro is being recorded to prevent later simulations being saved under incorrect names.

Setting Up the Physics Models

Select the physics models for the gas.

The boundaries read in from file `hex.ccm` already have suitable names. Rename the region:

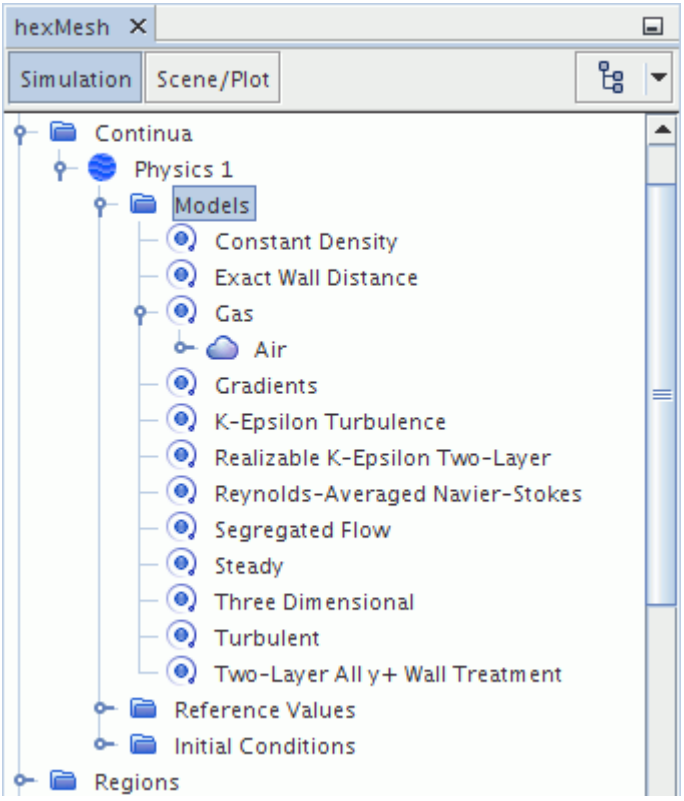
1. Rename the **Regions** > **Default_Fluid** node to `Fluid`.
2. Open the **Fluid** > **Boundaries** node.
3. Select each boundary node in turn.
   The corresponding boundary location is shown in the geometry scene.

Suitable models must be chosen for the simulation, as follows:

4. Open the **Continua** node.
5. Right-click the **Physics 1** node and choose **Select models**.
6. In the *Physics 1 Model Selection* dialog, select the following models, in order:

Group Box	Model
Material	**Gas**
Flow	**Segregated Flow**
Equation of State	**Constant Density**
Time	**Steady**
Viscous Regime	**Turbulent**
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**

   In the `hexMesh` window, the color of the **Physics 1** node has turned from gray to blue to indicate that models have been activated.
7. Click **Close**.
8. Open the **Physics 1** > **Models** node to see the selected models.



The default material properties for air and the reference values are suitable for this simulation.

Setting Initial Conditions

Set some initial conditions for this simulation.

1. Open the **Continua** > **Physics 1** > **Initial Conditions** node.
2. Select the **Turbulence Specification** node and set the *Method* to **K + Epsilon**.
3. Select the **Turbulent Dissipation Rate** node and set the *Value* to `0.02 m^2/s^3`
4. Similarly, change the **Turbulent Kinetic Energy** to `0.01` J/kg.

Setting Boundary Conditions and Values

The default boundary conditions for the wall, outlet, and symmetry boundaries are all suitable so the only changes required are to the inlet boundary.

Use the following steps:

1. Select the **Regions** > **Fluid** > **Boundaries** > **Inlet** > **Physics Conditions** > **Turbulence Specification** node and set the *Method* to `K + Epsilon`.
2. Select the **Physics Values** > **Turbulent Dissipation Rate** node and set the *Value* to `0.07 m^2/s^3`.
3. Similarly, change the turbulent kinetic energy at the inlet to `0.024` J/kg.

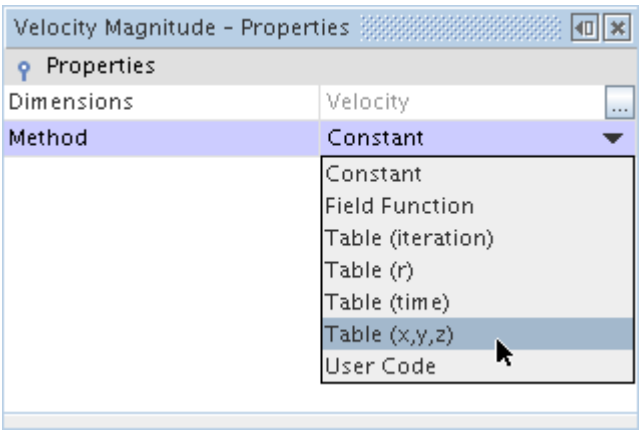
Give the velocity at the inlet a parabolic profile, using a text file called `inlet.xy`. Before such data can be used as a boundary condition, it must be imported as a table.

4. Open the **Tools** node.
5. Right-click the **Tables** node and select **New Table** > **File Table**.
6. In the *Open* dialog, navigate to the `doc/startutorialsdata/incompressibleFlow/data` directory and select file `inlet.xy`.
7. Click **Open**.

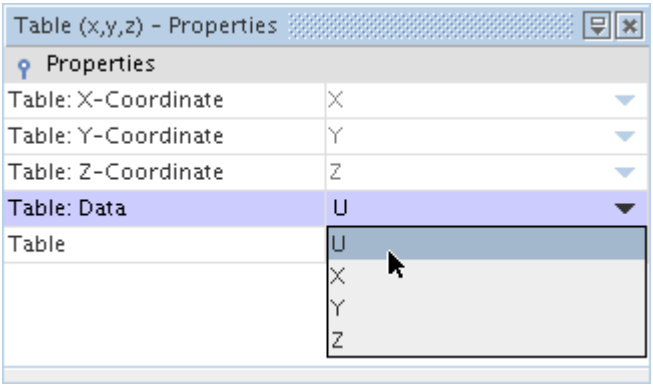
   The data is now stored in a table called `inlet`.

To assign the data as an inlet velocity magnitude boundary condition:

8. Select the **Regions** > **Fluid** > **Boundaries** > **Inlet** > **Physics Values** > **Velocity Magnitude** node and set the *Method* to **Table (x,y,z)**.



9. Select the **Velocity Magnitude** > **Table (x,y,z)** node and set the *Table* to **inlet** and the *Table: Data* to **U**.



Setting Stopping Criteria

The default stopping criterion is set to end the calculation after 1000 iterations. Such a value is excessive, so reduce the maximum number of iterations.

- Select the **Stopping Criteria** > **Maximum Steps** node and set the *Maximum Steps* to `500` .

Visualizing the Solution

Display velocity field vectors on the symmetry plane and contours of y+ values on the walls.

Start by creating a vector scene.

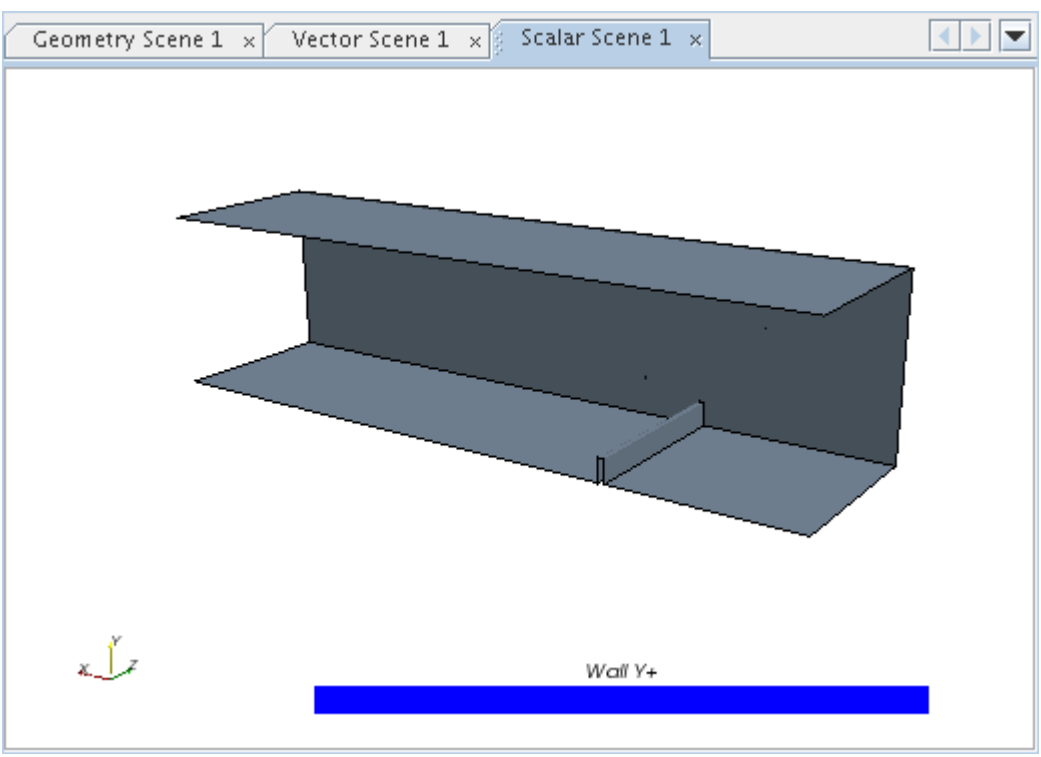
1. Right-click the **Scenes** node and then select **New Scene** > **Vector**.

As you only want to see vectors on the symmetry plane, make sure that this is the only part selected for both the vector and outline displayers.

2. Select the **Scenes** > **Vector Scene 1** > **Displayers** > **Vector 1** > **Parts** node and click the right-hand side of the *Parts* property.
3. Select **Regions** > **Fluid** > **Symplane** in the *Parts* dialog. Click **OK**.
4. Select the **Parts** node within the **Outline 1** node.
5. Edit the *Parts* property so that only **Fluid: Symplane** is selected.
6. Zoom in on the outline in the display window.

Create a plot showing y+ contours on the walls of the geometry:

7. Right-click the **Scenes** node and select **New Scene** > **Scalar**.
8. Select the **Scenes** > **Scalar Scene 1** > **Displayers** > **Outline 1** > **Parts** node and edit the *Parts* property so that the parts **Fluid: BottomWall**, **Fluid: Obstruction**, **Fluid: SideWall** and **Fluid: TopWall** are selected.
9. Select the same parts for the **Scalar 1** displayer.
10. Right-click the scalar bar at the bottom of the *Scalar Scene 1* display and select **Wall Y+**.
11. Use the mouse buttons to change the viewpoint to one similar to that shown below.



Setting Up Line Surfaces for Plotting

Create two parts by intersecting a plane with the bottom wall and symmetry plane boundary surfaces.

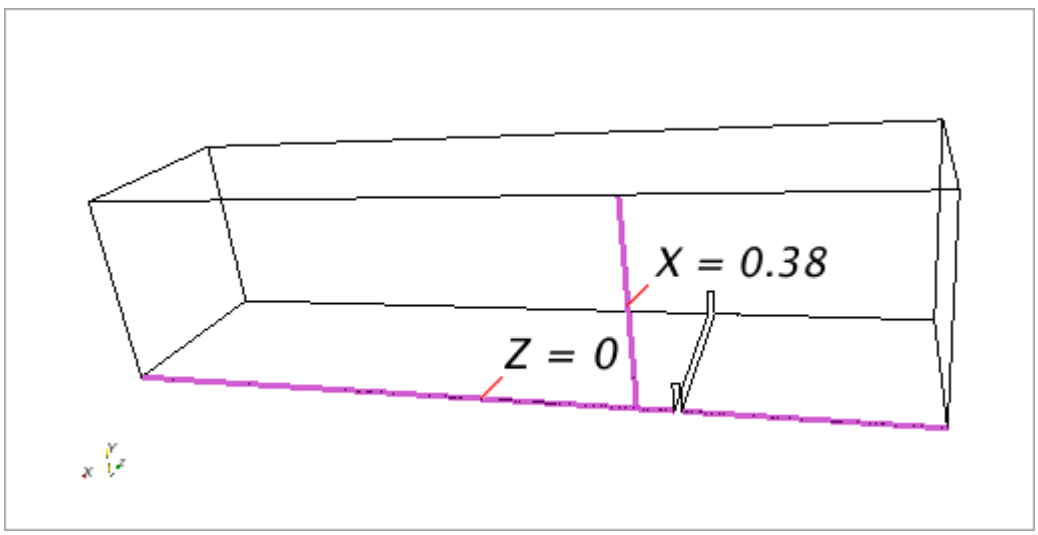
This results in the two line segments that are shown below.

To create a section:

1. Click the *Geometry Scene 1* tab in the *Graphics* window to activate it.
2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane**

   A new interactive in-place dialog (*Create Plane Section*) appears.
3. Set the following values:

Box	Property	Value
Input Parts	**Fluid: BottomWall**	Selected
Plane Parameters	**Origin**	`[0.0, 0.0, 1e-8]` m
	Normal	`[0, 0, 1]` m
Display	**No Displayer**	Selected



4. Click **Create**, then click **Close**.

   A new derived part named **Plane Section** is created.
5. Rename this node to `Z = 0`

The section is offset slightly from the symmetry plane to make sure that realistic wall shear stress values are displayed.

Monitor the velocity along a line segment on the symmetry plane at x = 0.38 m. To create a section for this segment:

6. For a second time, right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane**

The in-place dialog appears.

7. Set the following values:

Box	Property	Value
Input Parts	**Fluid: Symplane**	Selected
Plane Parameters	**Origin**	[`0.38, 0.0, 0.0`] m
	Normal	[`1, 0, 0`] m
Display	**No Displayer**	Selected

8. Click **Create**, then click **Close**.
9. Rename the new **Plane Section** node to `X = 0.38`

Plotting Simulation Data

Create plots of velocity and wall shear stress.

For the velocity plot, import experimental data from a file and plot it alongside the numerical data. The development of the solution can be observed throughout the run by viewing these plots.

To create the wall shear stress plot:

1. Right-click the **Plots** node and select **New Plot** > **XY Plot**.
2. Rename the **XY Plot 1** node to `WSS Plot`.
3. Select the **WSS Plot** node and set *Parts* to **Derived Parts** > **Z = 0**.

Recirculation zones occur upstream and downstream of the obstacle and are visible on the graph as regions where the wall shear stress is negative. To determine the length of the downstream recirculation zone, change the graph formatting.

4. Expand the **WSS Plot** > **Y Types** > **Y Type 1** node and set the following properties:

Node	Property	Value
Scalar Function	*Field Function*	**Wall Shear Stress** > **Laboratory** > **i**
Z = 0	*Sort Plot Data*	Activated
└ **Line Style**	*Style*	**Solid**
└ **Symbol Style**	*Shape*	**None**

5. Expand the **WSS Plot** > **Axes** node and set the following properties:

Node	Property	Value
Bottom Axis	*Maximum*	`0.9`
	Minimum	`0.3`
└ **Major Labels**	*Spacing*	`0.05`
Left Axis	*Maximum*	`0.005`

Node	Property	Value
	Minimum	`-0.0075`
└ **Major Labels**	*Spacing*	`0.0025`

To create the velocity data plot:

6. Right-click the **Plots** node and select **New Plot** > **XY Plot**.
7. Rename the **XY Plot 1** node to `Velocity Plot at X = 0.38`.
8. Expand the **Velocity Plot at X = 0.38** node and set the following properties:

Node	Property	Value
Velocity Plot at X = 0.38	*Parts*	**Derived Parts** > **X = 0.38**
└ **Y Types** > **Y Type 1**	*Data Type*	**Scalar**
└ **Y Types** > **Y Type 1** > **Scalar Function**	*Field Function*	**Velocity** > **Lab Reference Frame** > **Laboratory** > **i**
└ **X Type**	*Data Type*	**Direction**
└ **X Type** > **Vector Quantity**	*Value*	`[0.0, 1.0, 0.0] m`

To compare the experimental velocity data with the velocity plot, import the data in the supplied file into a table:

9. Right-click the **Tools** > **Tables** node and select **New Table** > **File Table**.
10. In the *Open* dialog, navigate to the `doc/startutorialsdata/incompressibleFlow/data` directory and select file `exp380.xy`.
11. Click **Open**.

The data in that file are now stored in a table named ‘exp380’.

To display it on the velocity plot:

12. Return to the **Velocity Plot at X = 0.38** node.
13. Right-click the **Velocity Plot at X = 0.38** > **Data Series** node and select **Add Data**.
14. Select **exp380** and click **OK**.

    The supplied experimental data appear in the plot. Numerical data is shown once the simulation has started.
15. Right-click the **Velocity Plot at X = 0.38** > **Axes** node and select **Orient Axes**.
16. In the *Orient Axes* dialog, click the (horizontal) and (clockwise) rotation buttons, then click **OK**.

To rename the axes:

17. Expand the **Axes** node and set the following properties:

Node	Property	Setting
Bottom Axis > **Title**	*Title*	`Velocity (m/s)`
Left Axis > **Title**	*Title*	`Y (m)`

18. To view the solution while it is being generated, look at the scalar and vector scenes.

The pre-processing set-up for the simulation is complete.

19. Stop the macro recording by clicking (**Stop Recording**) in the Macro toolbar.
20. Save the simulation.

Initializing and Running the Simulation

Initialize the solution and run the analysis.

1. Click the (**Initialize Solution**) button in the *Solution* toolbar or use the **Solution** > **Initialize Solution** menu item. The initial zero velocity condition is shown in the *Velocity Plot at X = 0.38* display.
2. Click the (**Run**) button in the *Solution* toolbar to run the analysis.

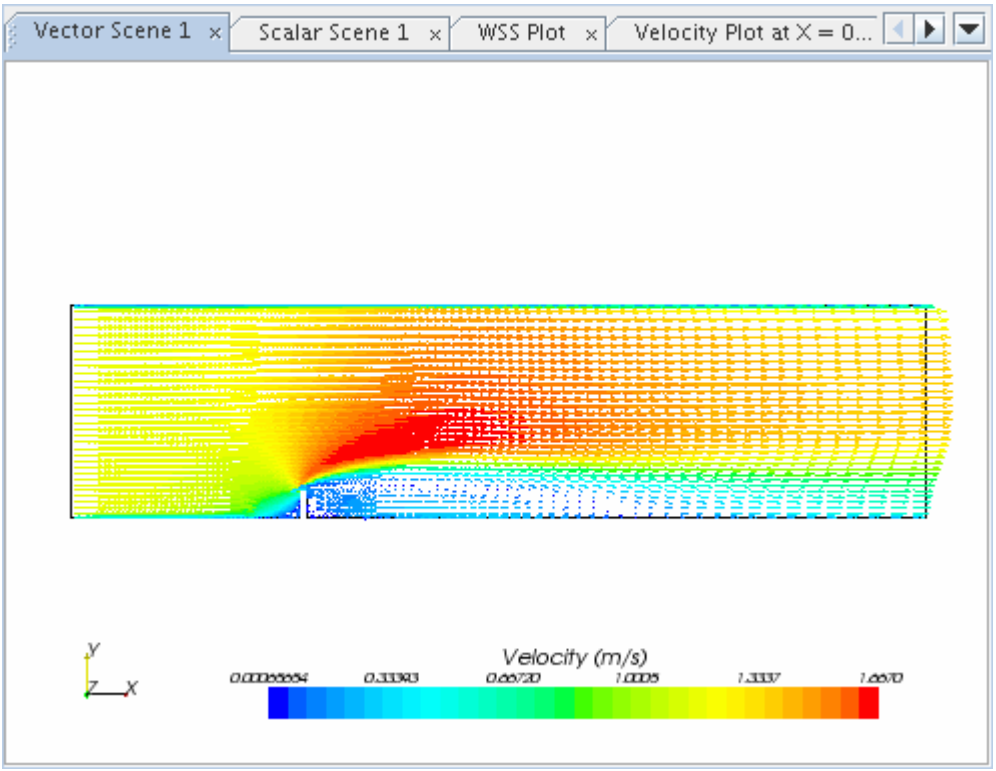
   The *Residuals* display is automatically created and shows the progress of the solver. The tabs at the top of the *Graphics* window make it possible to select any active display for viewing. Select the *Velocity Plot at X = 0.38* display if you wish to observe the numerical solution development.

   During the run, it is possible to stop the process by clicking the (**Stop**) button in the toolbar. If you do halt the simulation, it can be continued later by clicking the (**Run**) button. If left alone, the simulation continues until the stopping criteria are satisfied.
3. After the run is finished, save the simulation.

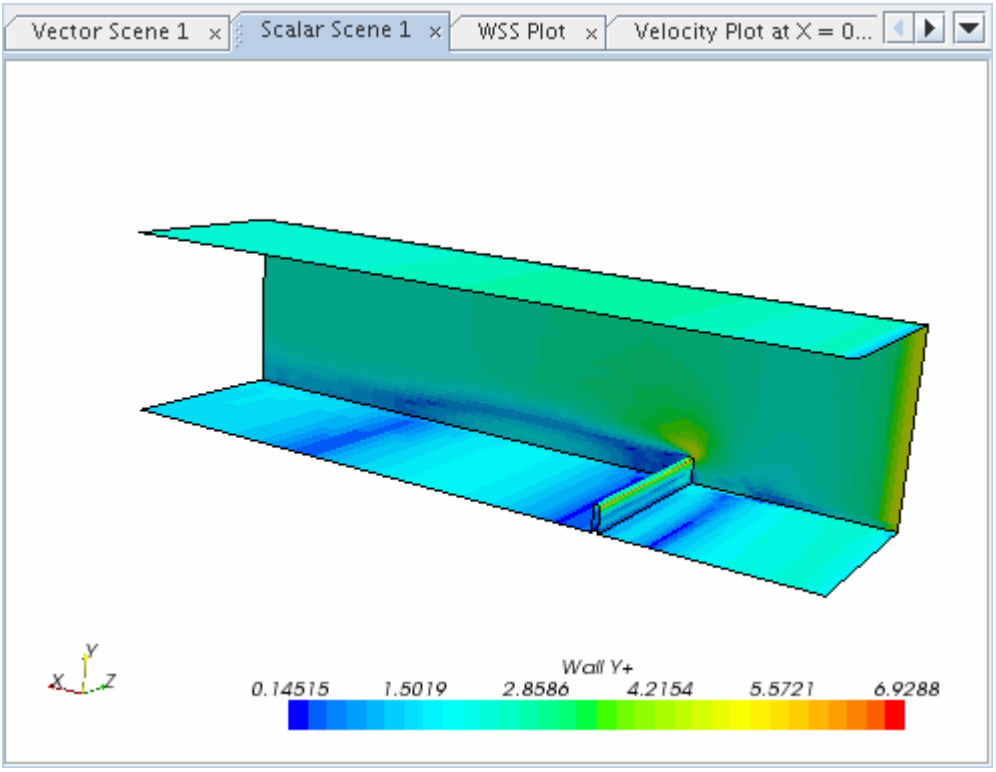
Visualizing the Results

View the vector and scalar scenes.

1. Activate the *Vector Scene 1* display to view the velocity vector field:



2. Click the tab for the *Scalar Scene 1* display to see the y+ distribution along the walls.



The maximum y+ value is 6.9288, at the top of the obstruction. But for the vast majority of the flow regime the y+ values are below 3.0, the maximum value considered acceptable for a rigorous examination of wall data.

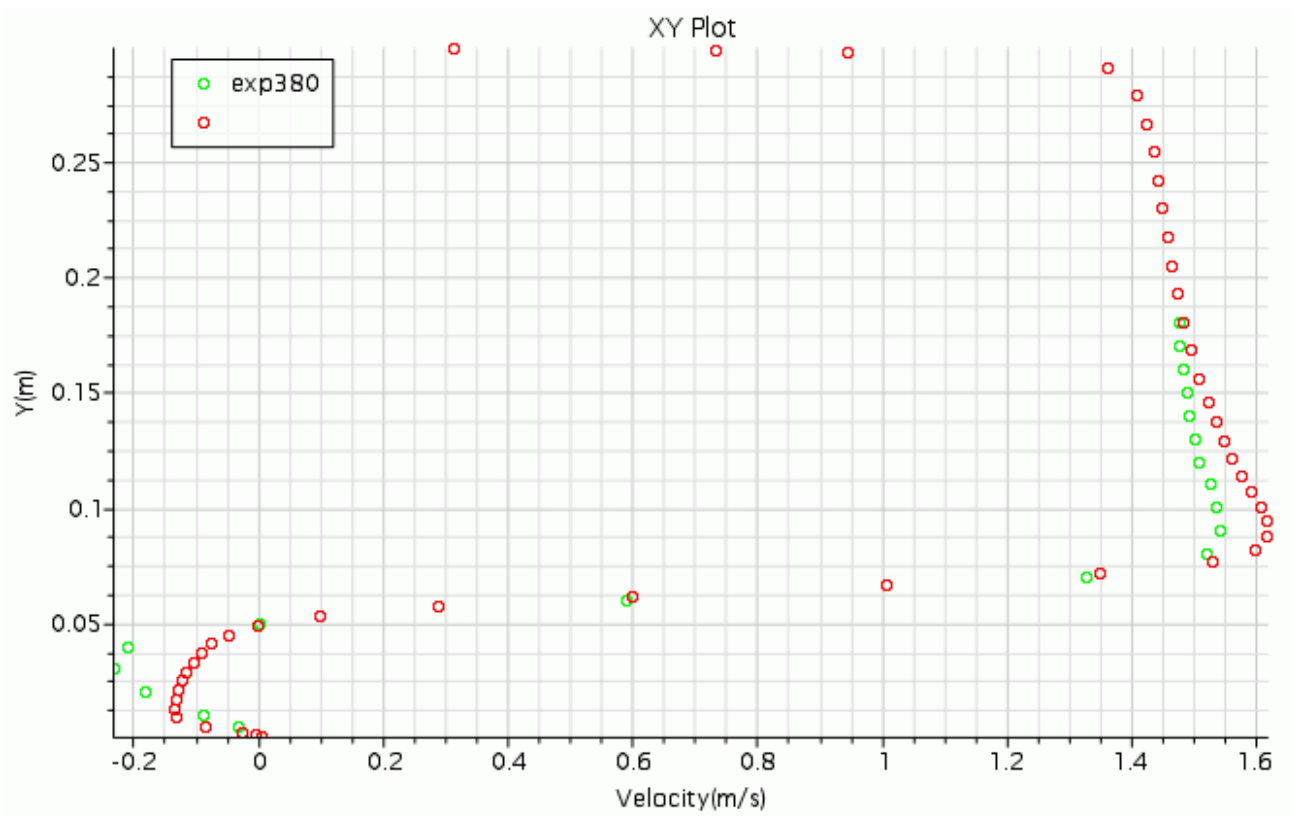
Examining Plots

View the velocity and shear stress plots.

Use the following steps:

1. Activate the *Velocity Plot at X = 0.38* display.

The plot now appears as shown below.

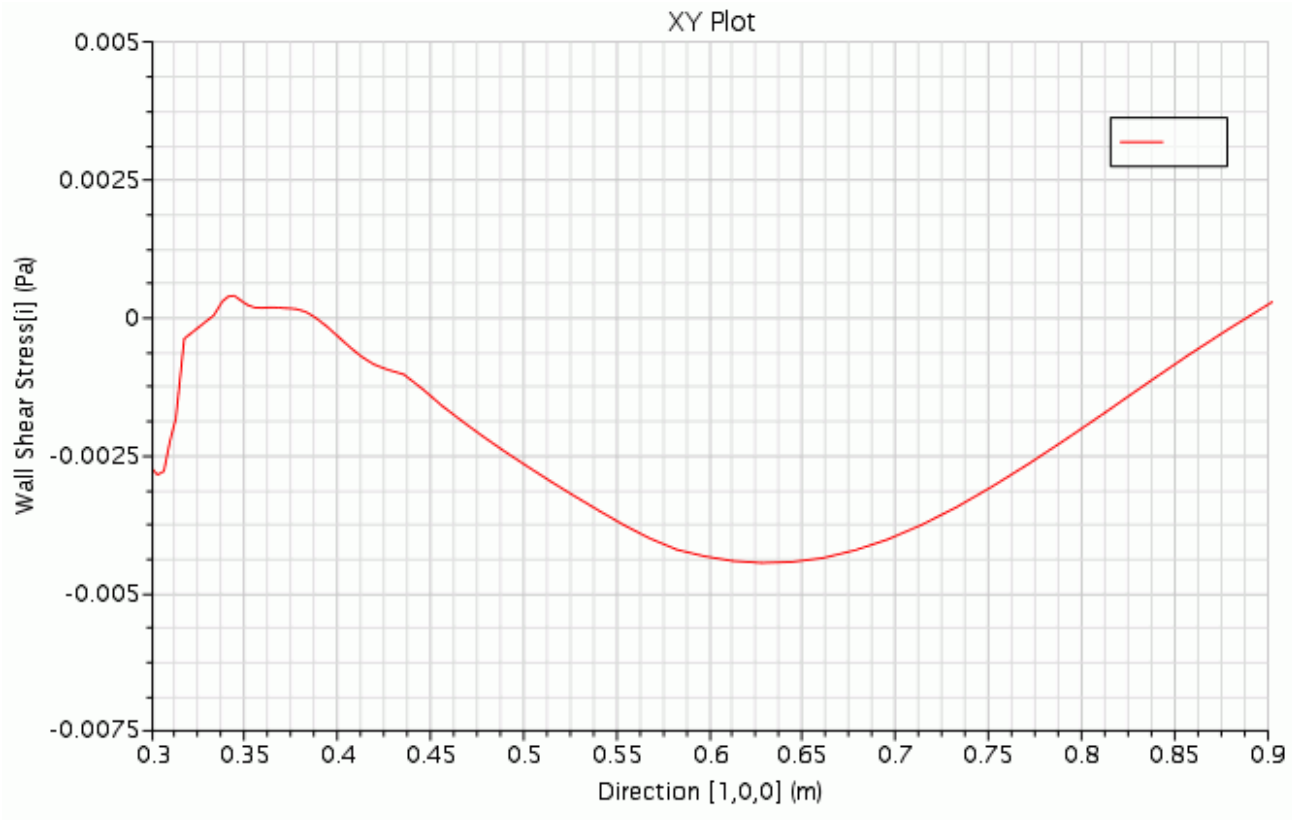


The experimental results compare reasonably well with the numerical data. You can see that the greatest discrepancy is found in the recirculation zone downstream of the obstacle.

Given that the y+ contours indicate that the wall data is also reasonably accurate, you can estimate the length of the recirculation zone from the wall shear stress plot.

2. Select the *WSS Plot* tab at the top of the *Display* window.

The plot now appears as shown below:

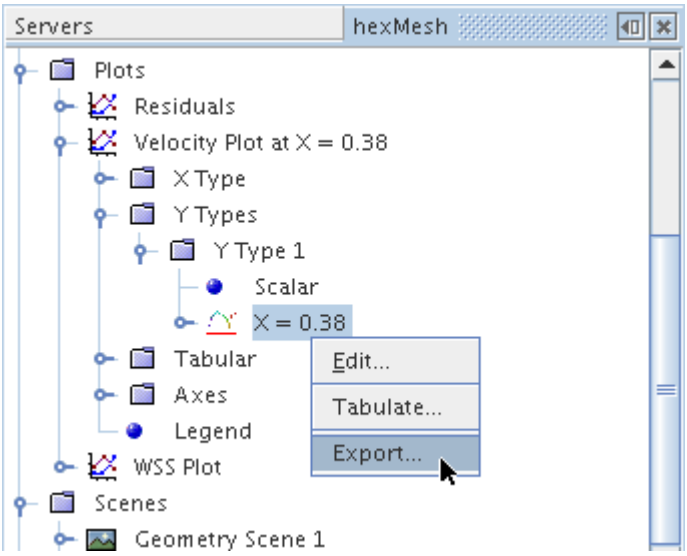


The above result shows that the downstream recirculation zone extends from approximately X = 0.38 m to X = 0.9 m and hence is roughly 0.5 m in length.

Exporting Simulation Data

Export velocity plot data to a file for comparison with similar data produced using the other two meshes.

1. Open the **Plots** > **Velocity Plot at X = 0.38** > **Y Types** > **Y Type 1** node.
2. Right-click the **X = 0.38** node within and select **Export**.



3. In the *Save* dialog, enter `hex380` as the File Name and then click **Save**.
4. Close the simulation.

The section dealing with the hexahedral mesh is now complete.

Tetrahedral Mesh

This part of the tutorial models the same problem as the Hexahedral Mesh part, shown in the introductory Steady Flow: Channel Flow with Multiple Meshes section. However, the mesh used in this case is tetrahedral. All problem conditions and properties are defined using the java macro file created when setting up the Hexahedral Mesh case.

Contents:

Importing the Mesh and Naming the Simulation
Running the Macro
Opening the Plots
Initializing and Running the Simulation
Visualizing the Results
Examining Plots
Exporting Simulation Data

Importing the Mesh and Naming the Simulation

Create a new simulation and import the tetrahedral mesh.

1. Select **New Simulation** from the menu bar.

Continue by importing the tetrahedral mesh and naming the simulation. The grid has been saved in the `.ccm` file format, which contains all necessary cell, vertex and boundary information for the problem geometry.

2. Select **File** > **Import** > **Import Volume Mesh** from the menu bar.
3. In the *Open* dialog, simply navigate to the `[INSTALL_DIR]/doc/startutorialsdata/incompressibleFlow/data` subdirectory of your STAR-CCM+ installation directory and select file `tet.ccm`
4. Click **Open** to start the import. A geometry scene is created in the *Graphics* window.
5. Finally, navigate back to the `meshComparison` directory and save the new simulation to disk under file name `tetMesh.sim`.

Running the Macro

Replay the macro you created to set up the entire problem and generate the necessary scenes and plots.

1. Click (**Play Macro**) in the Macro toolbar.
2. In the *Open* dialog find and select file `setup.java` created for the Hexahedral Mesh, then click **Open**.

The macro takes a few minutes to run. When it is complete, the vector scene 1 and scalar scene 1 is created in the *Graphics* window.

Opening the Plots

Open the required plots.

1. Open **Plots** node and right-click the **Velocity Plot at X = 0.38** and **WSS Plot** nodes in turn and select **Open**.
2. Save the simulation.

Initializing and Running the Simulation

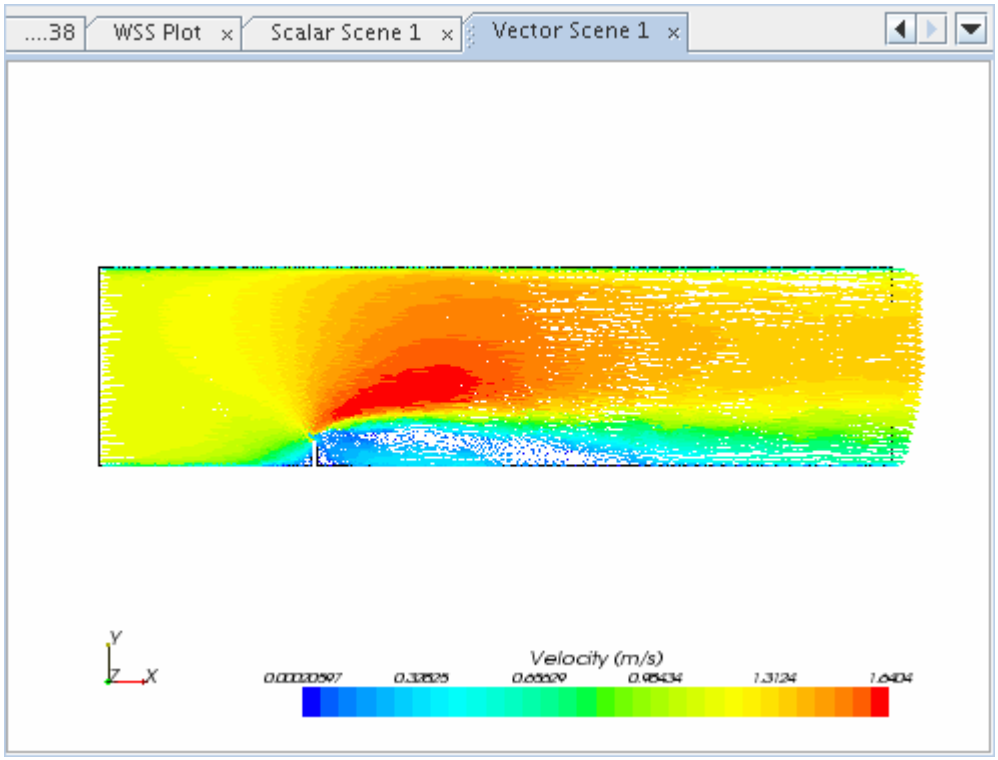
Initialize and run the simulation.

1. Click (**Initialize Solution**) in the toolbar.
2. Click (**Run**) in the toolbar to run the analysis.
3. After the run is finished, save the simulation.

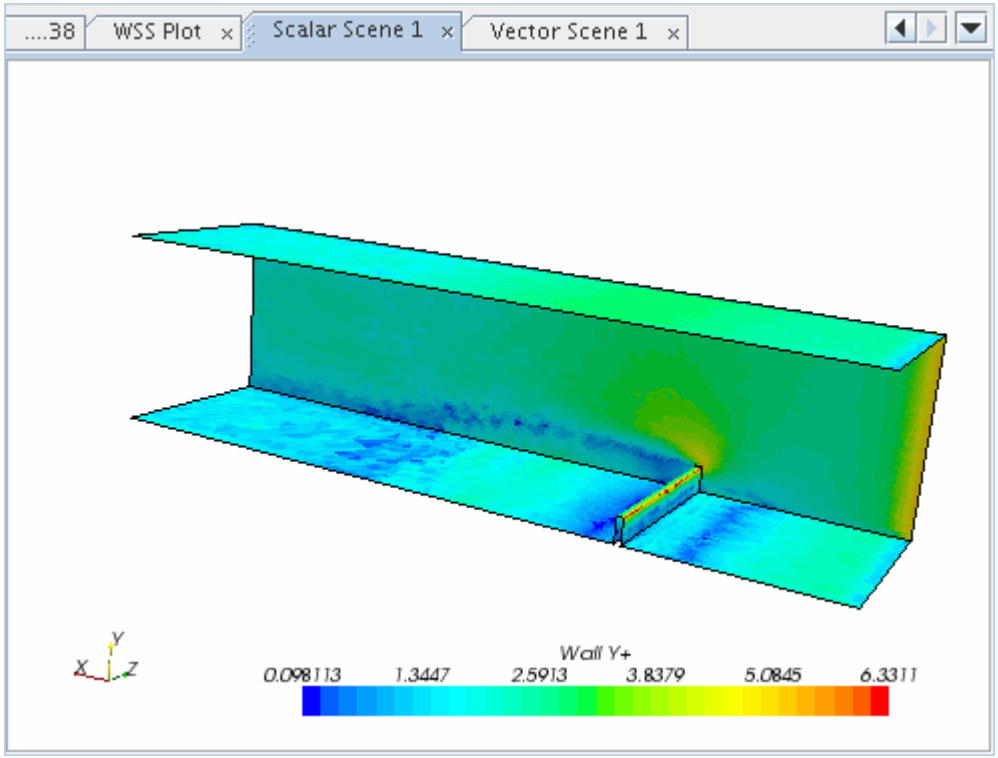
Visualizing the Results

View the vector and scalar scenes.

1. Activate the *Vector Scene 1* display to view the velocity vector field.



2. Click the tab for the *Scalar Scene 1* display to see the y+ distribution on the walls.



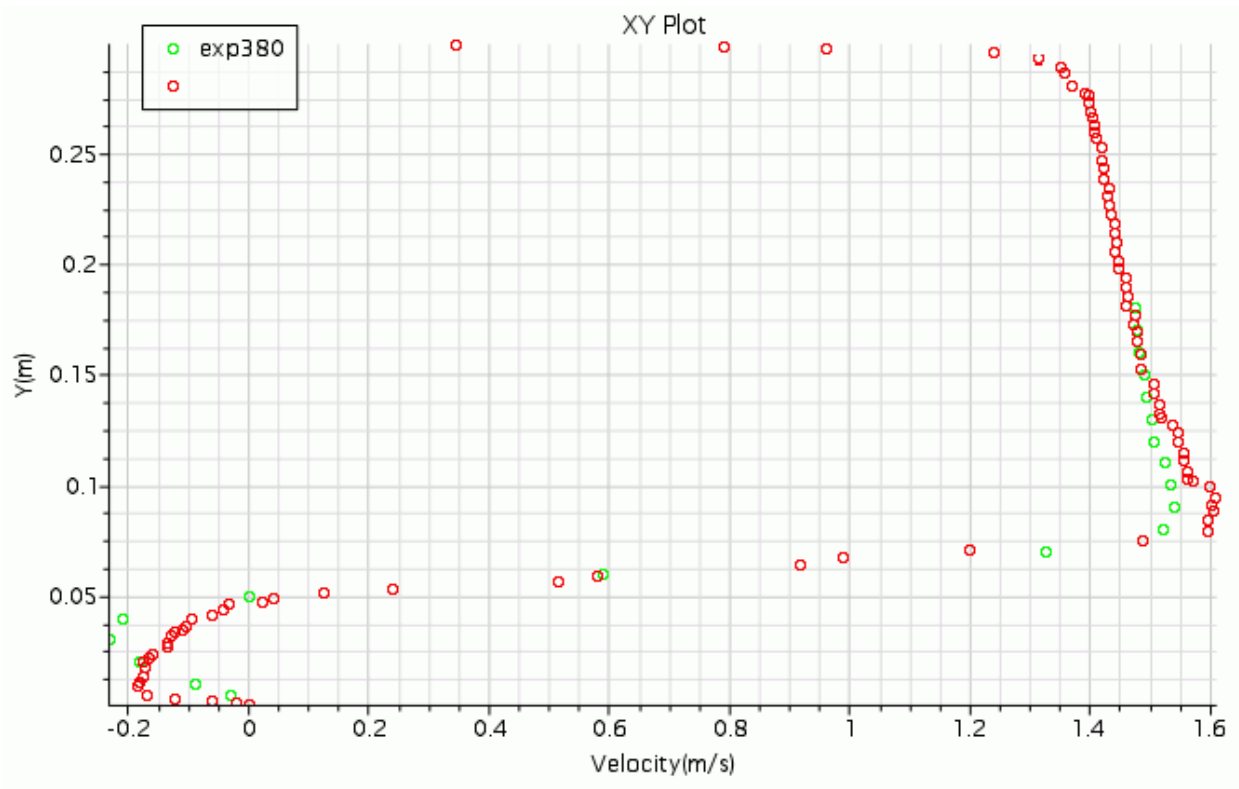
The maximum y+ value is 6.3311, at the top of the obstruction. however for the vast majority of the flow regime the y+ values are below 3.0, the maximum value considered acceptable for a rigorous examination of wall data.

Examining Plots

View the velocity and shear stress plots.

1. Activate the *Velocity Plot at X = 0.38* display.

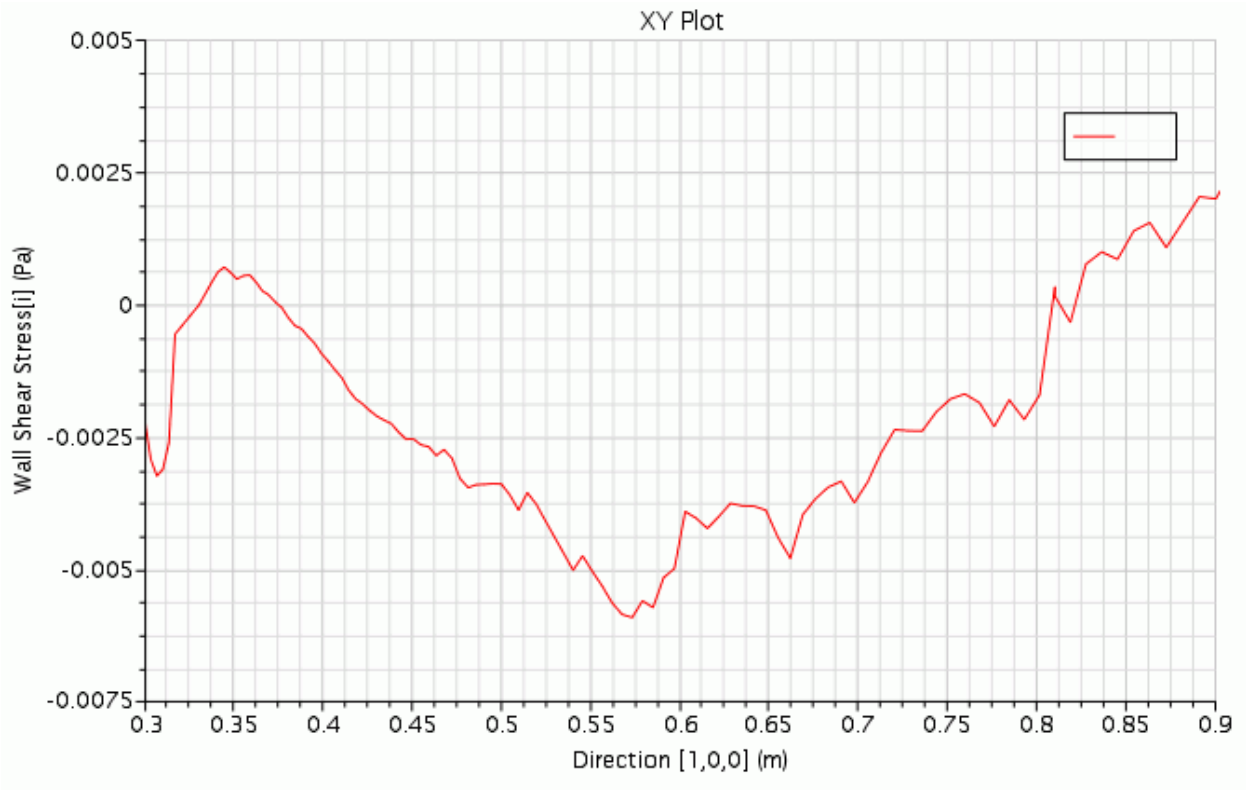
   The plot now appears as shown below.



   The experimental results generally compare well with the numerical data. Again the greatest discrepancy is found in the recirculation zone downstream of the obstacle.

2. Select the *WSS Plot* tab at the top of the *Display* window.

   The plot now appears similar to that shown below.

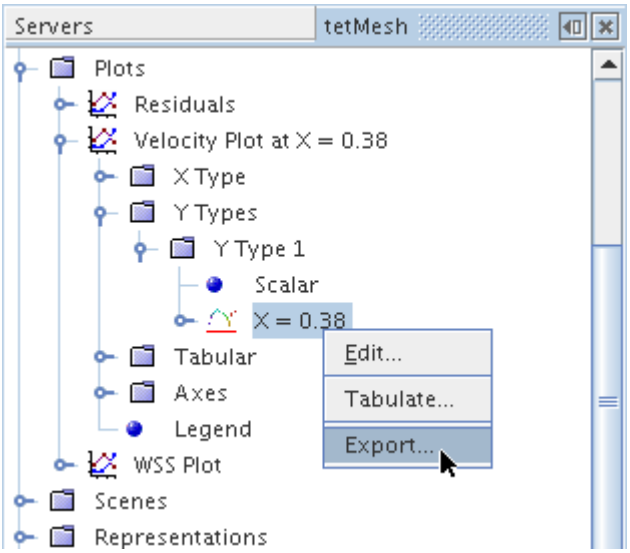


The above plot shows that the downstream recirculation zone extends from approximately X = 0.37 m to X = 0.82 m and hence is roughly 0.45 m in length.

Exporting Simulation Data

Export data from the velocity plot to a file for comparison with similar data produced in the other parts of this tutorial:

1. Select the **Plots** > **Velocity Plot at X = 0.38** > **Y Types** > **Y Type 1** > **X = 0.38** node and enable the *Sort Plot Data* option.
2. Right-click the **X = 0.38** node within and select **Export**.



3. In the *Save* dialog, enter `tet380` as the File Name and then click **Save**.

The tetrahedral mesh part of the tutorial is now complete.

4. Close the simulation.

Polyhedral Mesh

This part of the tutorial models the same problem as the previous two parts. However, the mesh used in this case is polyhedral. All problem conditions and properties are defined using the java macro file created when setting up the Hexahedral Mesh case.

Contents:

Importing the Mesh and Naming the Simulation
Running the Macro
Opening the Plots
Initializing and Running the Simulation
Visualizing the Results
Examining Plots
Comparing Mesh Types

Importing the Mesh and Naming the Simulation

Create a new simulation and import the polyhedral mesh.

Use the following steps:

1. Select **New Simulation** from the menu bar.

Continue by importing the polyhedral mesh and naming the simulation. The grid has been saved in the `.ccm` file format, which contains all necessary cell, vertex and boundary information for the problem geometry.

2. Select **File** > **Import** > **Import Volume Mesh** from the menu bar.
3. In the *Open* dialog, simply navigate to the `[INSTALL_DIR]/doc/startutorialsdata/incompressibleFlow/data` subdirectory of your STAR-CCM+ installation directory and select file `poly.ccm`.
4. Click **Open** to start the import. A geometry scene is created in the *Graphics* window.
5. Finally, navigate back to the `meshComparison` directory and save the new simulation to disk under file name `polyMesh.sim`.

Running the Macro

Run the setup macro to create the vector and scalar scenes.

Use the following steps:

1. Click (**Play Macro**) in the **Macro** toolbar.
2. In the *Open* dialog find and select file `setup.java` created in the Hexahedral Mesh, and click **Open**.

   The macro takes a few minutes to run. When it is complete, the *Vector Scene 1* and *Scalar Scene 1* is created in the *Graphics* window.

Opening the Plots

Open the required plots.

- Open the **Plots** node and right-click the **Velocity Plot at X = 0.38** and **WSS Plot** nodes in turn and select **Open**.

  The pre-processing setup for the case is complete.

Initializing and Running the Simulation

Initialize and run the simulation.

1. Click (**Initialize Solution**) in the toolbar.
2. Click (**Run**) in the toolbar to run the analysis.

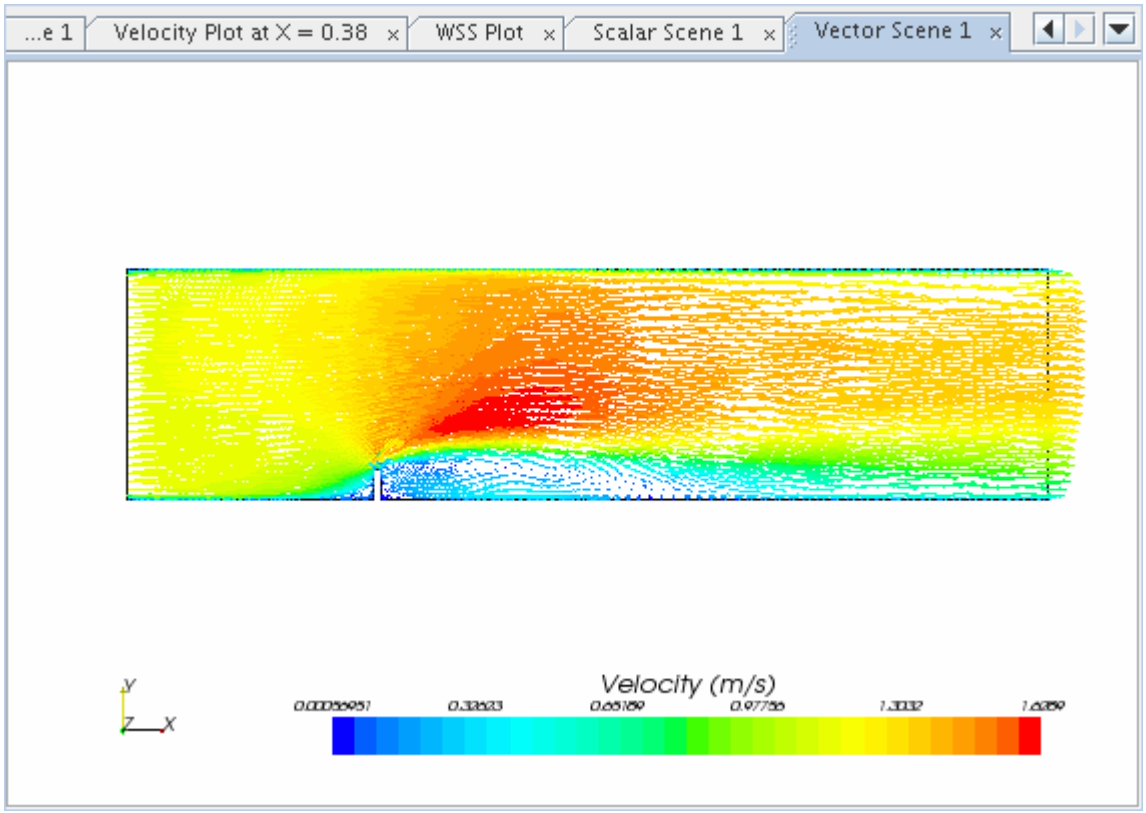
   The *Residuals* display is automatically created and shows the progress of the solver. The tabs at the top of the *Graphics* window make it possible to select any active display for viewing. Select the *Velocity Plot at X = 0.38* display if you wish to observe the numerical solution development.

   During the run, it is possible to stop the process by clicking (**Stop**) in the toolbar. If you do halt the simulation, it can be continued later by clicking (**Run**). If left alone, the simulation continues until the stopping criteria are satisfied.
3. After the run is finished, save the simulation.

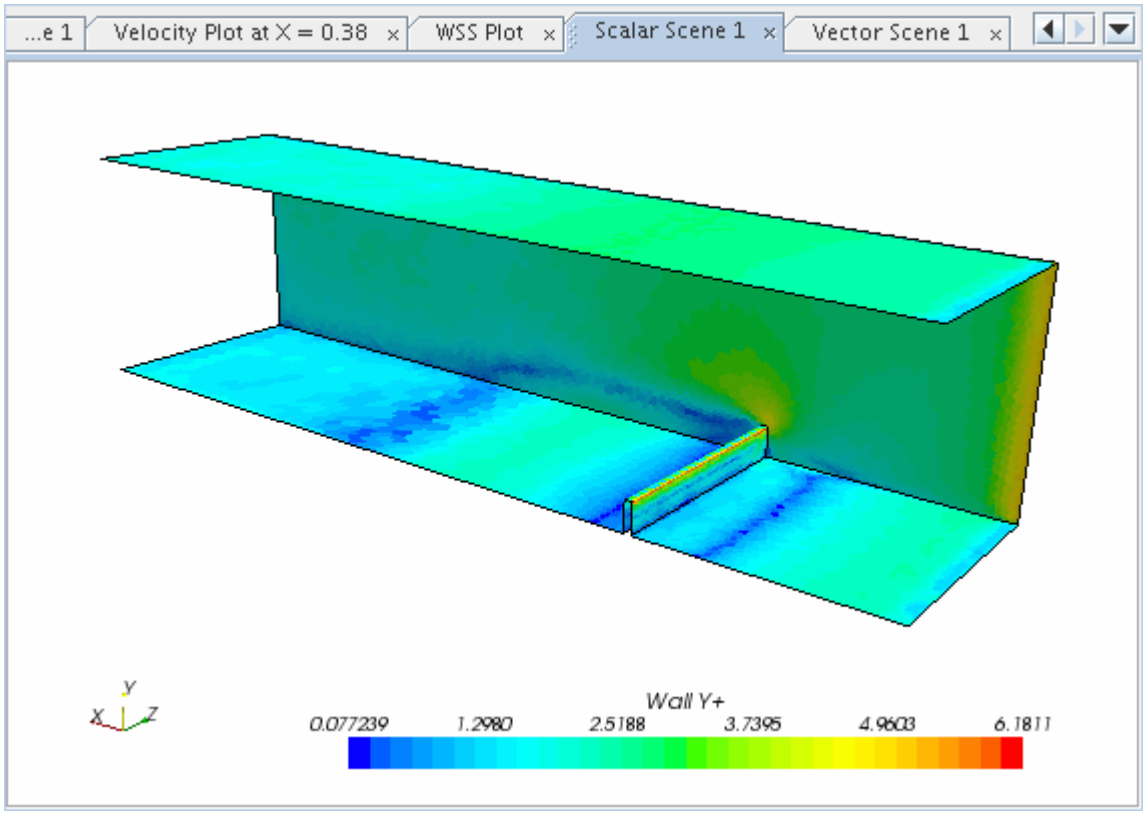
Visualizing the Results

View the vector and scalar scenes.

1. Activate the *Vector Scene 1* display to view the velocity vector field.



2. Click the tab for the *Scalar Scene 1* display to see the y+ distribution on the walls.



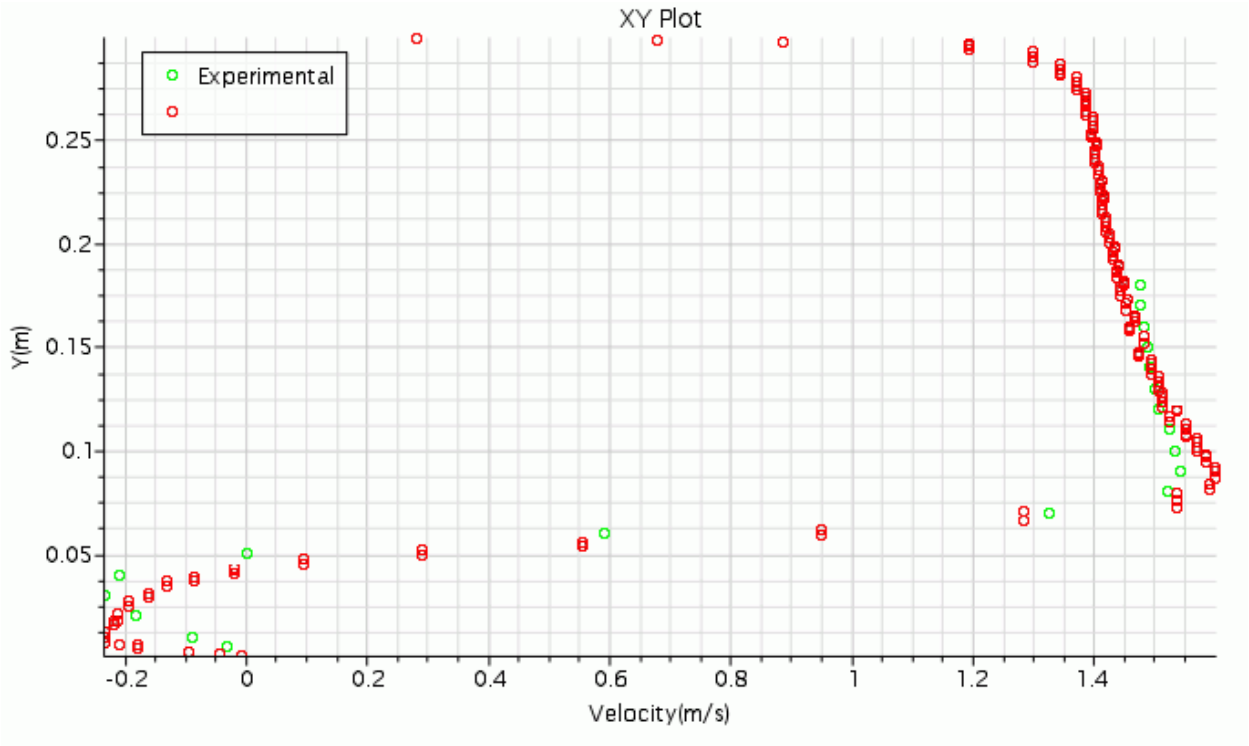
The maximum y+ value is 6.1811, at the top of the obstruction. However, for the vast majority of the flow regime the y+ values are below 3.0, the maximum value considered acceptable for a rigorous examination of wall data.

Examining Plots

View the velocity and shear stress plots.

1. Activate the *Velocity Plot at X = 0.38* display.

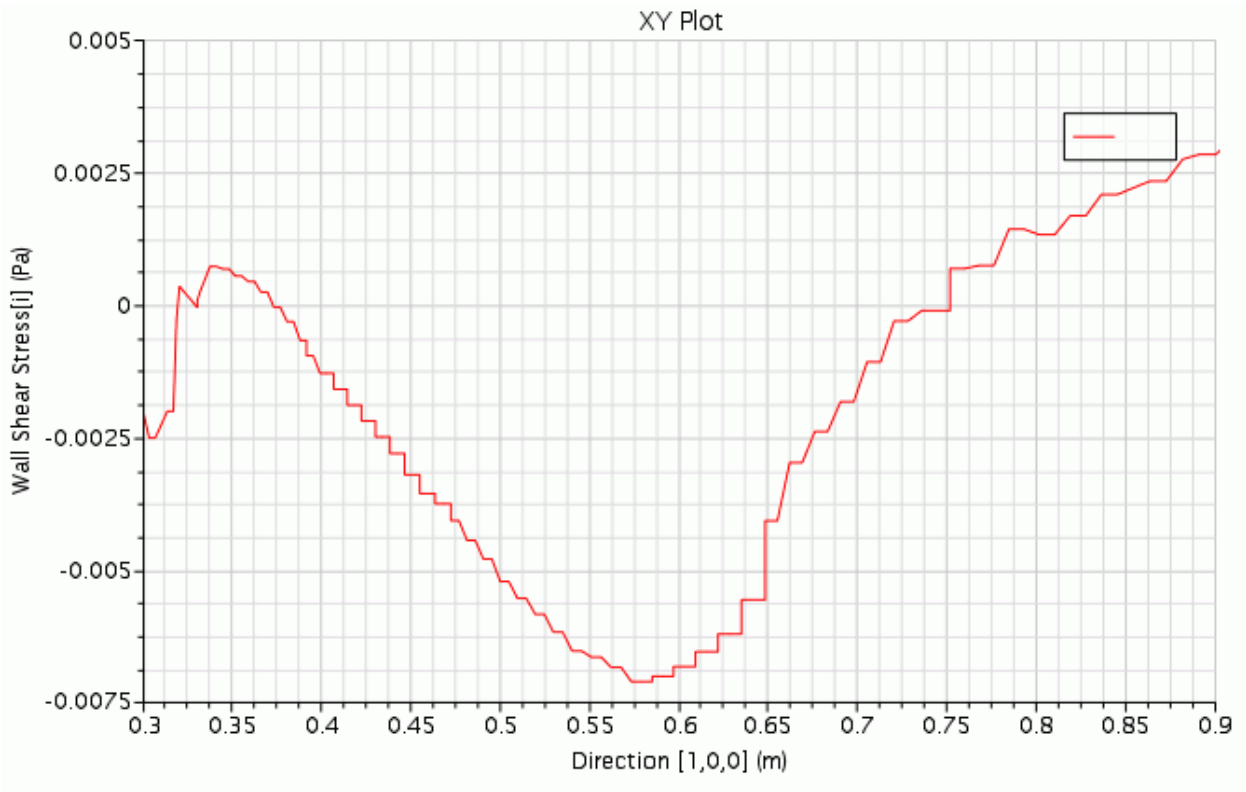
   The plot appears as shown below:



The experimental results generally compare well with the numerical data. Again the greatest discrepancy is found in the recirculation zone downstream of the obstacle.

2. Select the *WSS Plot* tab at the top of the *Display* window.

   The plot now appears as shown below.



The above plot shows that the downstream recirculation zone extends from approximately X = 0.37 m to X = 0.77 m and hence is roughly 0.4 m in length. This value is the same as the value the tetrahedral mesh analysis predicts but shorter than the value the hexahedral mesh analysis predicts.

Comparing Mesh Types

You have already examined the velocity plot and seen that the numerical solution compares well with the experimental data. Compare the velocity data that is produced on this mesh with that produced on the Hexahedral Mesh and the Tetrahedral Mesh.

First, import the data from the previous two simulations:

1. Open the **Tools** node.
2. Right-click the **Tables** node and select **New Table** > **File Table**.
3. Select file `hex380.csv` and click **Open**.
4. Do the same to import the data saved in file `tet380.csv`.

To plot the imported data:

5. Open the **Plots** > **Velocity Plot at X = 0.38** node.
6. Right-click the **Data Series** node and select **Add Data**.
7. In the *Add Data Providers to Plot* dialog, select **hex380** and click **OK**.
8. Expand the **Data Series** > **hex380** node and set the following properties:

Node	Property	Value
hex380	*Legend Name*	**HexMesh**
	Sort Plot Data	Activated
⌊ **Line Style**	*Style*	Solid
⌊ **Symbol Style**	*Shape*	**None**

9. Once again, right-click the **Data Series** node and select **Add Data**.
10. In the *Add Data Providers to Plot* dialog, select **tet380** and click **OK**.
11. Select the **Data Series** > **tet380** node and set the following properties:

Node	Property	Value
tet380	*Legend Name*	**TetMesh**
	Sort Plot Data	Activated
⌊ **Line Style**	*Style*	Solid
⌊ **Symbol Style**	*Shape*	**None**

12. Expand the **Data Series** > **exp380** node and set the following properties:

Node	Property	Value
exp380	*Legend Name*	`Experimental`
⌊ **Symbol Style**	*Color*	**Black**

13. Expand the **Plots** > **Velocity Plot at X = 0.38** > **Y Types** > **Y Type 1** node and set the following properties:

Node	Property	Value
X=0.38	*Legend Name*	**PolyMesh**
	Sort Plot Data	Activated
⌊ **Line Style**	**Style**	**Solid**
⌊ **Symbol Style**	**Shape**	**None**

The plot appears as shown below.

There is little difference in the quality of the results from the three meshes, although the result from the polyhedral mesh is slightly better than the other two in the recirculation zone. The polyhedral mesh has only 60,838 cells. The hexahedral mesh has 67,699 cells and the tetrahedral mesh has 140,141 cells. The analysis on the polyhedral mesh only required 121 iterations to reach convergence which compares to 221 for the hexahedral mesh and 181 for the tetrahedral mesh. The results therefore demonstrate the ability of polyhedral meshes to produce good quality results with lower cell counts and shorter run times than other mesh types.

14. Save the simulation.

Summary

This tutorial described a steady airflow simulation, using three different meshing methods.

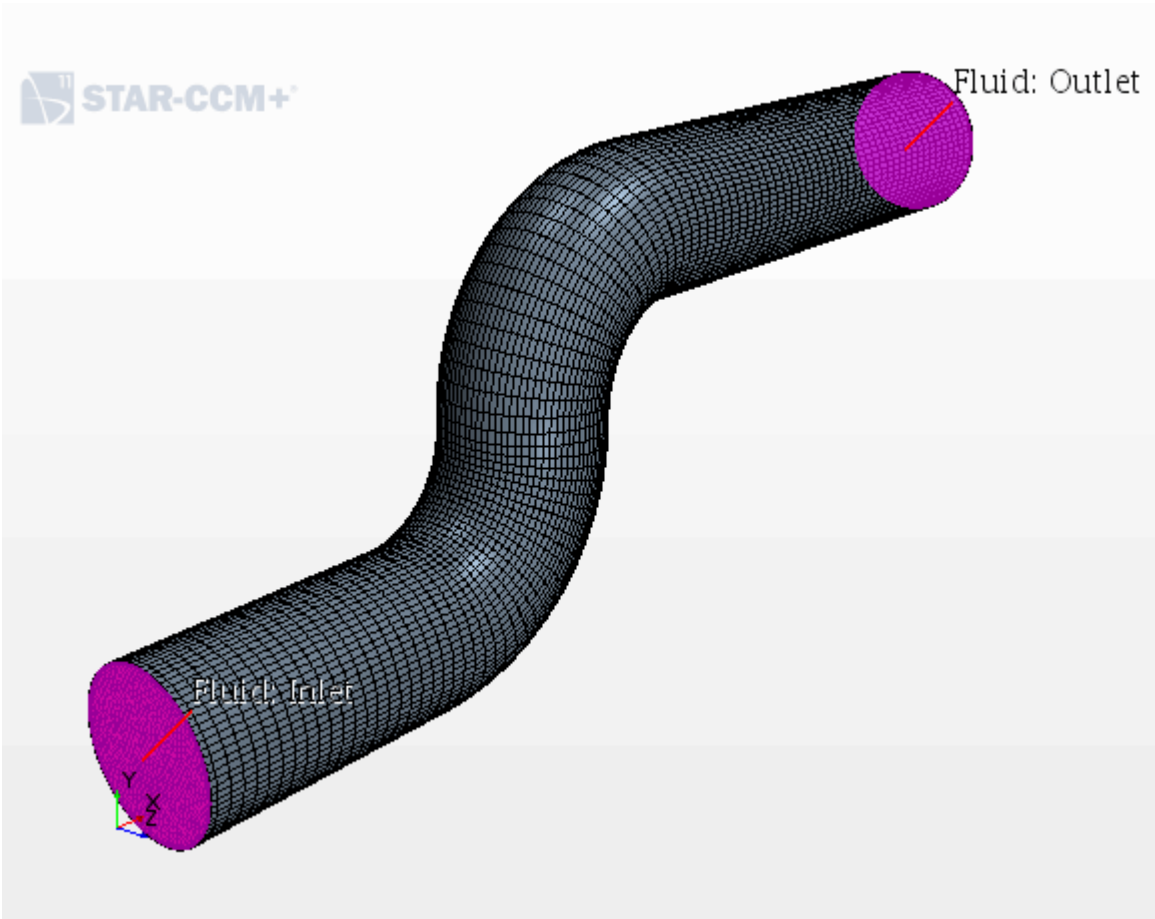
The tutorial introduced the following STAR-CCM+ features:

- Creating and replaying a java macro
- Renaming a region
- Defining models for single-phase, single-component incompressible fluid flow
- Defining boundary conditions
- Setting a stopping criterion from residuals
- Setting up a line segment on which solution results are plotted
- Plotting wall and flow field data
- Specifying plot options
- Initializing and running the solver until the residual magnitudes are sufficiently small
- Analyzing results using STAR-CCM+'s visualization facilities
- Examining XY-plots
- Exporting simulation data
- Comparing results from different simulations using XY-plots

Steady Flow: Laminar and Turbulent in an S-Bend

This tutorial demonstrates the flow of an incompressible gas through an s-bend of constant diameter (2 cm), for both laminar and turbulent flow. The first part of the tutorial covers the laminar flow, with a Reynolds number (Re) of 500, and the second part covers the turbulent flow, Re = 50,000. The same pipe geometry is used in both cases.

The geometry is created using 3D-CAD, which is the parametric solid modeler available within STAR-CCM+. The necessary region is created and a polyhedral mesh is obtained using the generalized cylinder meshing model. This model extrudes polyhedral cells in the direction of the flow, which results in a mesh that is suited for pipe flows.



Contents:

Prerequisites
Creating the S-Bend Geometry
Creating a Geometry Part
Assigning a Part to a New Region
Setting Up the Case

Generating a Surface Mesh
Generating a Volume Mesh
Selecting Physics Models
Modifying Material Properties
Setting Initial Conditions and Boundary Settings
Preparing a Scalar Scene
Setting Up Stopping Criteria
Running the Simulation
Visualizing the Results
Changing to a Turbulent Flow
Suggestions for Further Practice
Summary

Prerequisites

The instructions in the Steady Flow: Laminar and Turbulent in an S-Bend tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Associated Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Creating geometry in 3D-CAD	Cyclone Separator
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Creating the S-Bend Geometry

Create a new simulation and use 3D-CAD to set up the geometry.

Use the following steps:

1. Start up STAR-CCM+ in a manner that is appropriate to your working environment and create a New Simulation.
2. Save the new simulation to disk with the file name `sBend.sim`.

The geometry is created using 3D-CAD, which is the parametric solid modeler available within STAR-CCM+.

3. To activate 3D-CAD, right-click the **Geometry** > **3D-CAD Models** node and select **New**.

Start by renaming the 3D-CAD model.

4. Rename the **3D-CAD Model 1** node to `S-Bend`.

Contents:

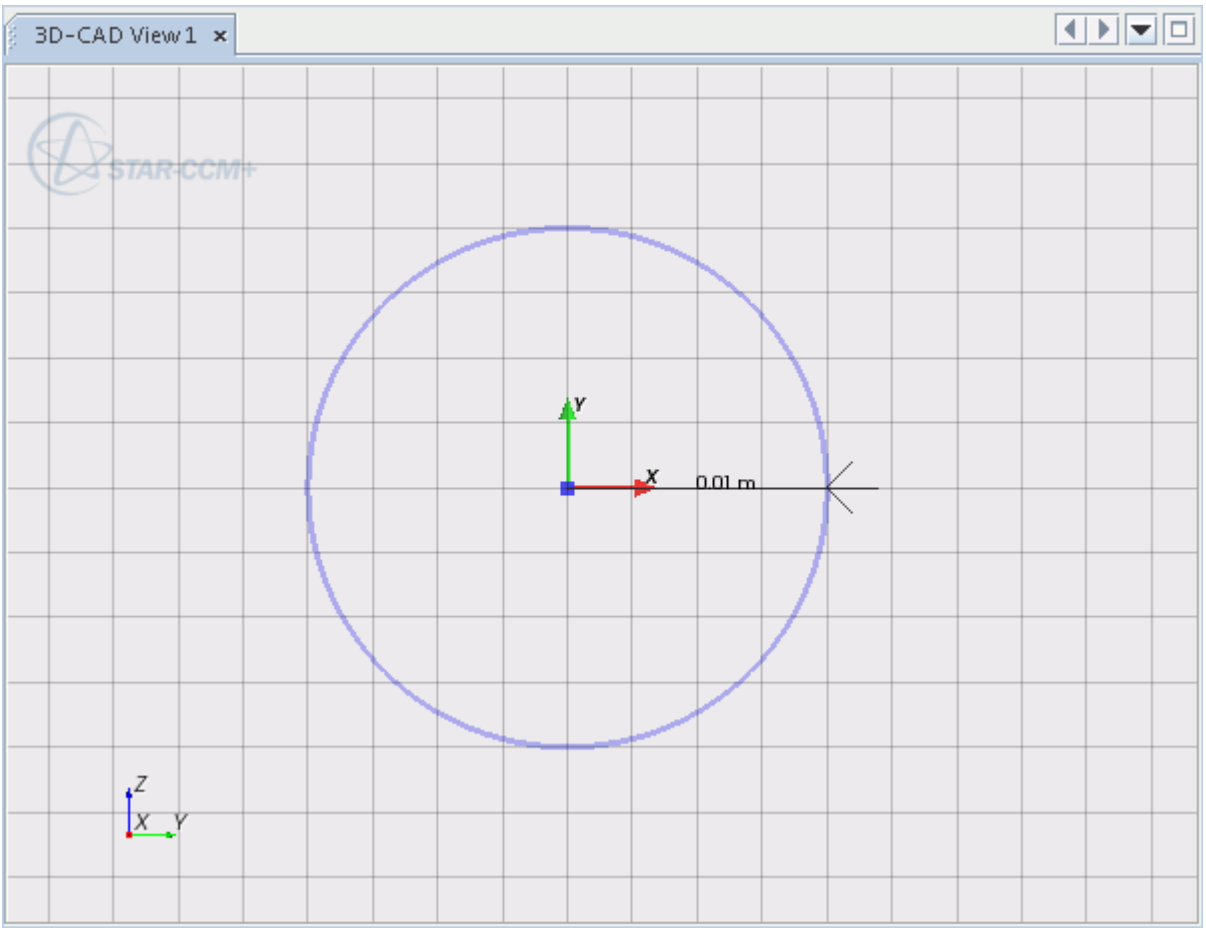
Creating the Geometry
Specifying Inlet and Outlet Faces

Creating the Geometry

Create the geometry of the s-bend pipe, including the circular inlet face.

Use the following steps:

1. Create a sketch on the YZ plane by right-clicking the **Features** > **YZ** node and selecting **Create Sketch**.
2. Click (**Set sketch grid spacing**) and change the *Grid Spacing* to `0.0025 m`.
3. Click (**View normal to sketch plane**) to bring the sketch plane into view.
4. Use the (**Create circle**) tool to draw a circle with a *radius* of `0.01 m` and whose center is on the origin, position [0, 0].

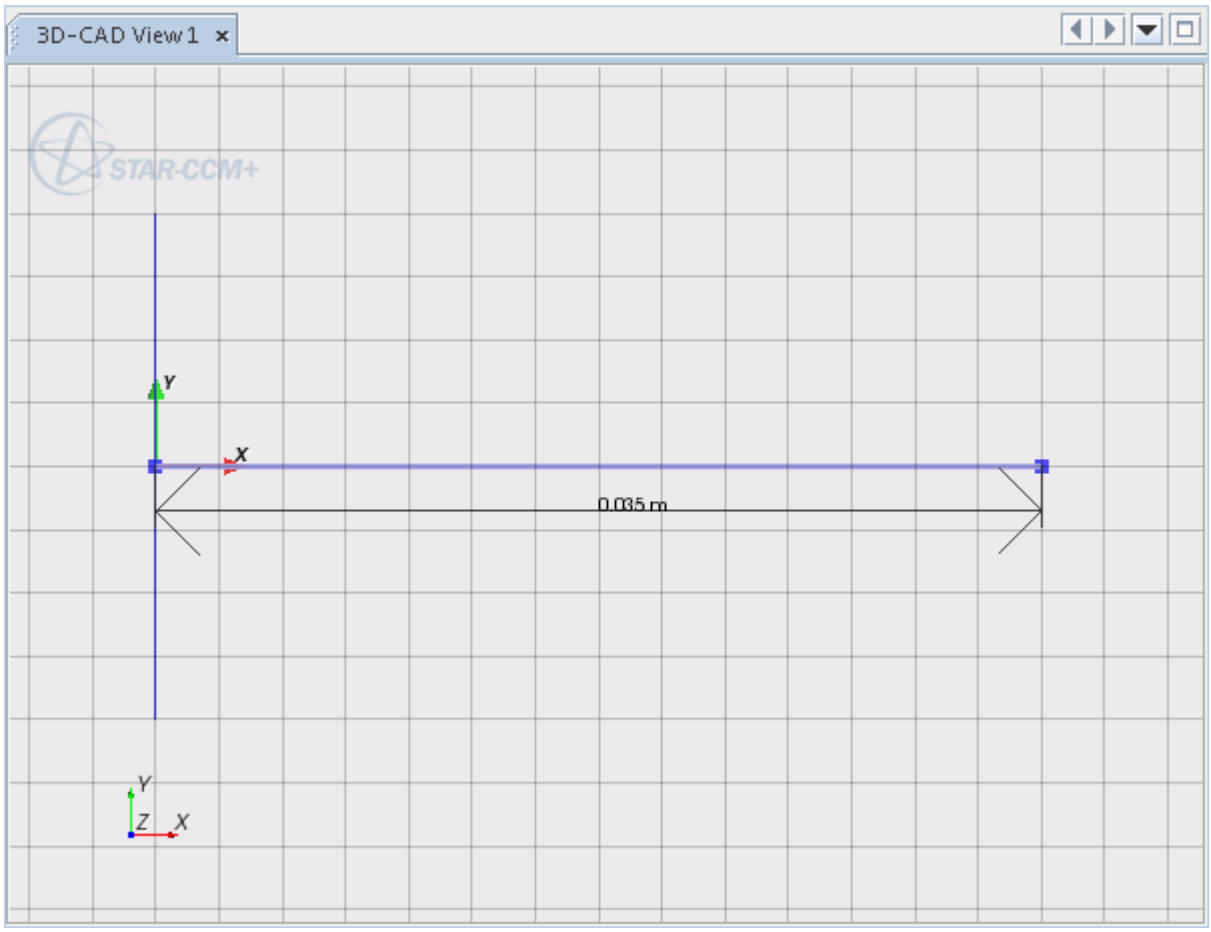


5. Click **OK** to exit the sketch.

To create the profile of the pipe along its length:

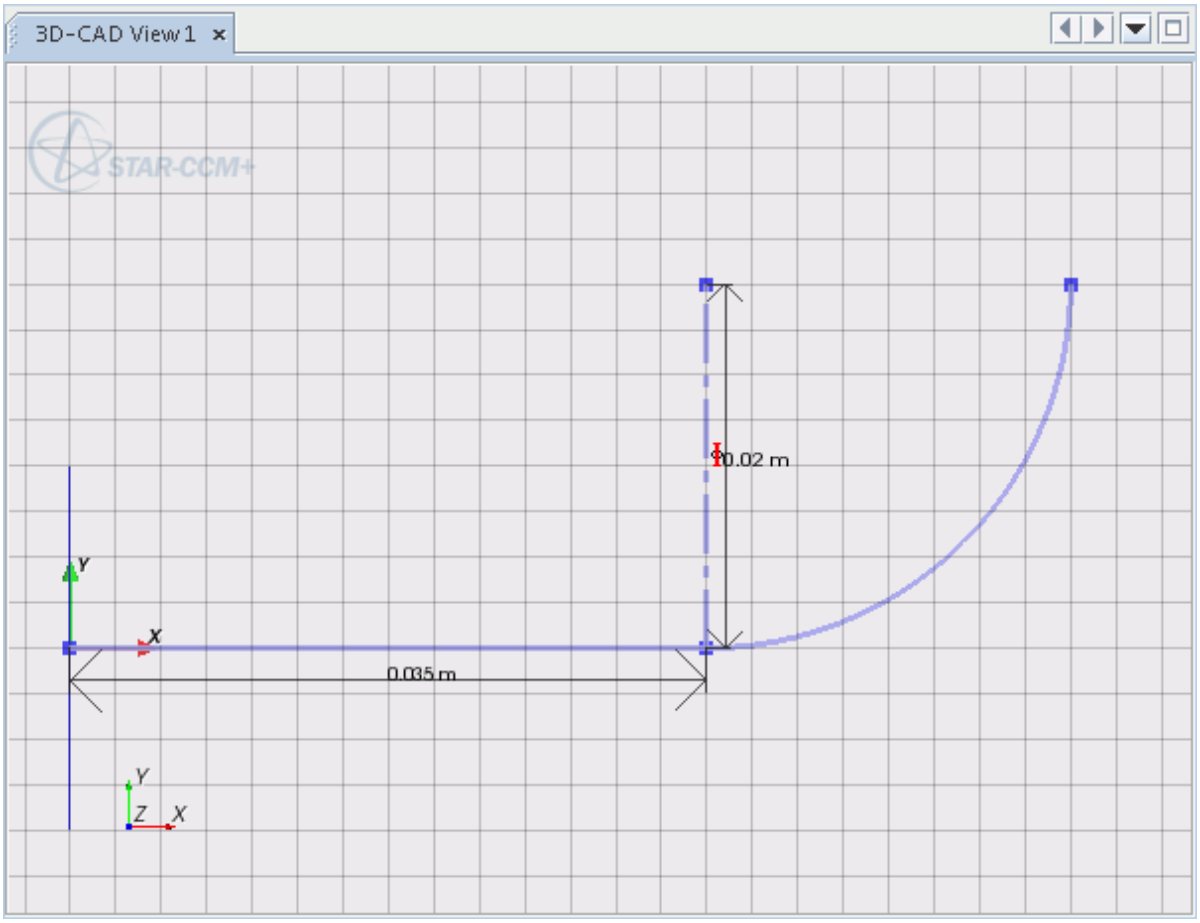
6. Create a sketch on the XY plane.
7. Click (**View normal to sketch plane**).

8. Use the (**Create line**) tool to draw a line of length `0.035 m` starting at the origin and extending in the positive x direction.



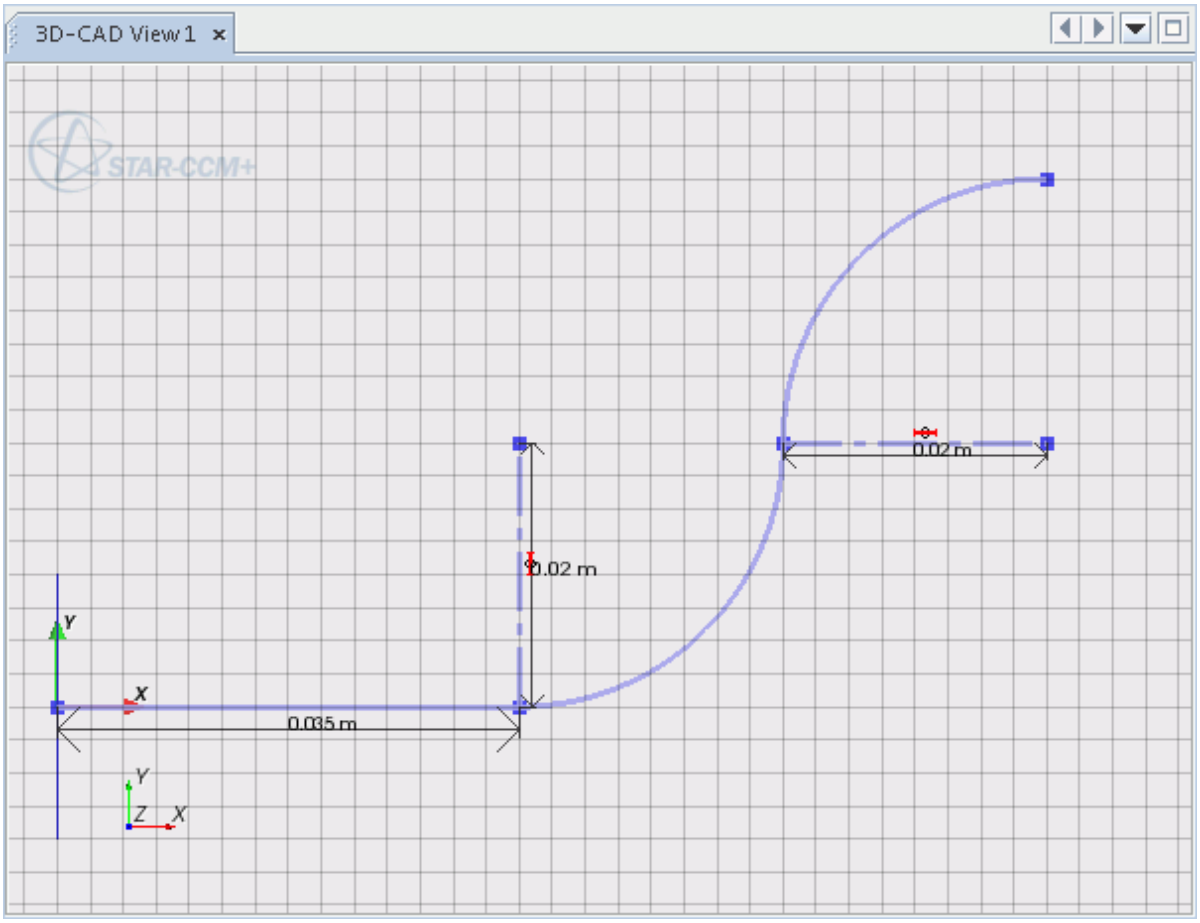
9. Press `Esc` to exit the line tool.

10. Use the (**Center point circular arc**) tool to draw an arc with a radius of 0.02 m. Three mouse clicks are required:
    a) Click position `[0.035 m, 0.02 m]` to define the center point.
    b) Click position `[0.035 m, 0.0 m]` to define the start point.
    c) Click position `[0.055 m, 0.02 m]` to define the end point.



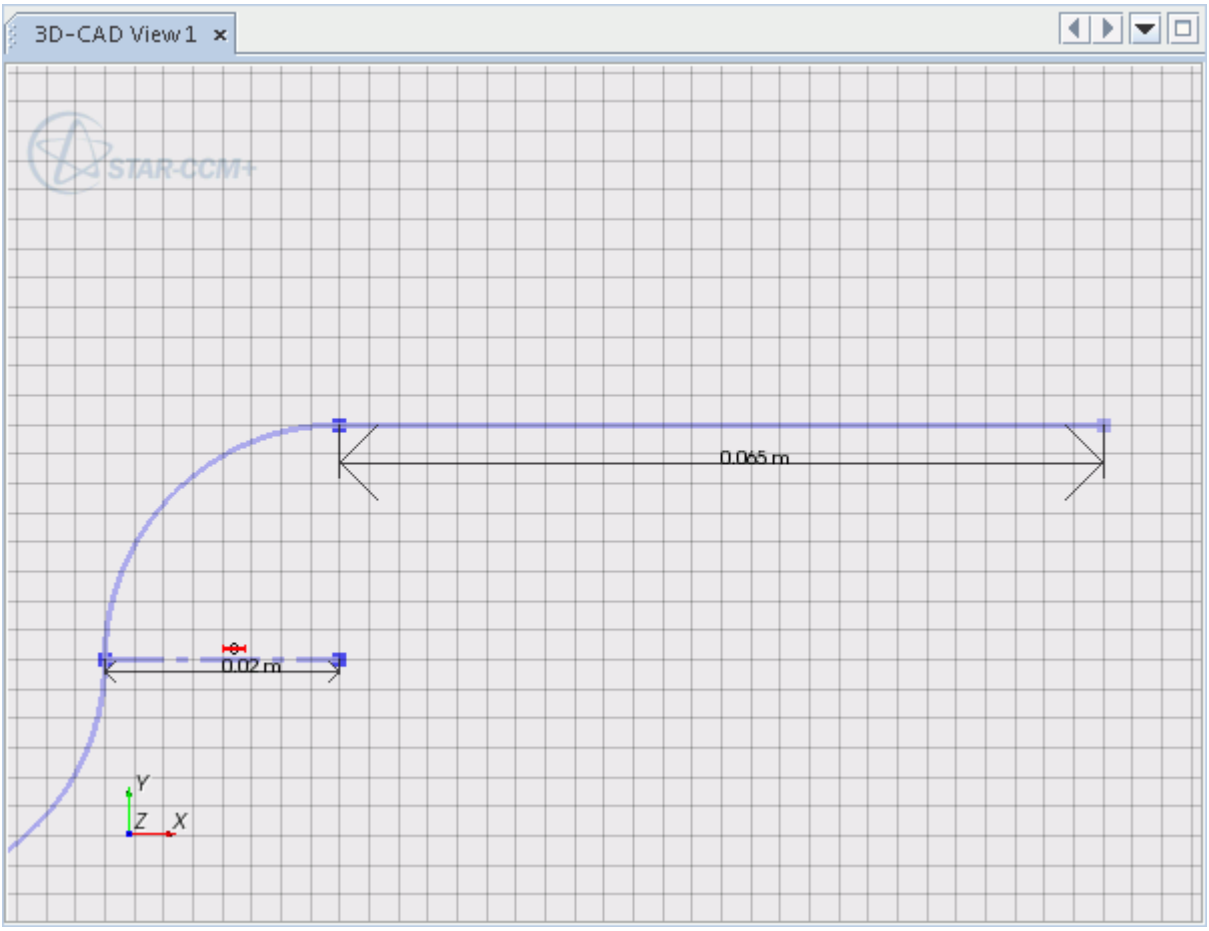
To create a second arc to complete the “S” shape:

11. Use the (**Center point circular arc**) tool to create the second arc. The coordinates for the three clicks are listed below:
    a) Center: `[0.075 m, 0.02 m]`.
    b) Start point: `[0.075 m, 0.04 m]`.
    c) End point: `[0.055 m, 0.02 m]`.

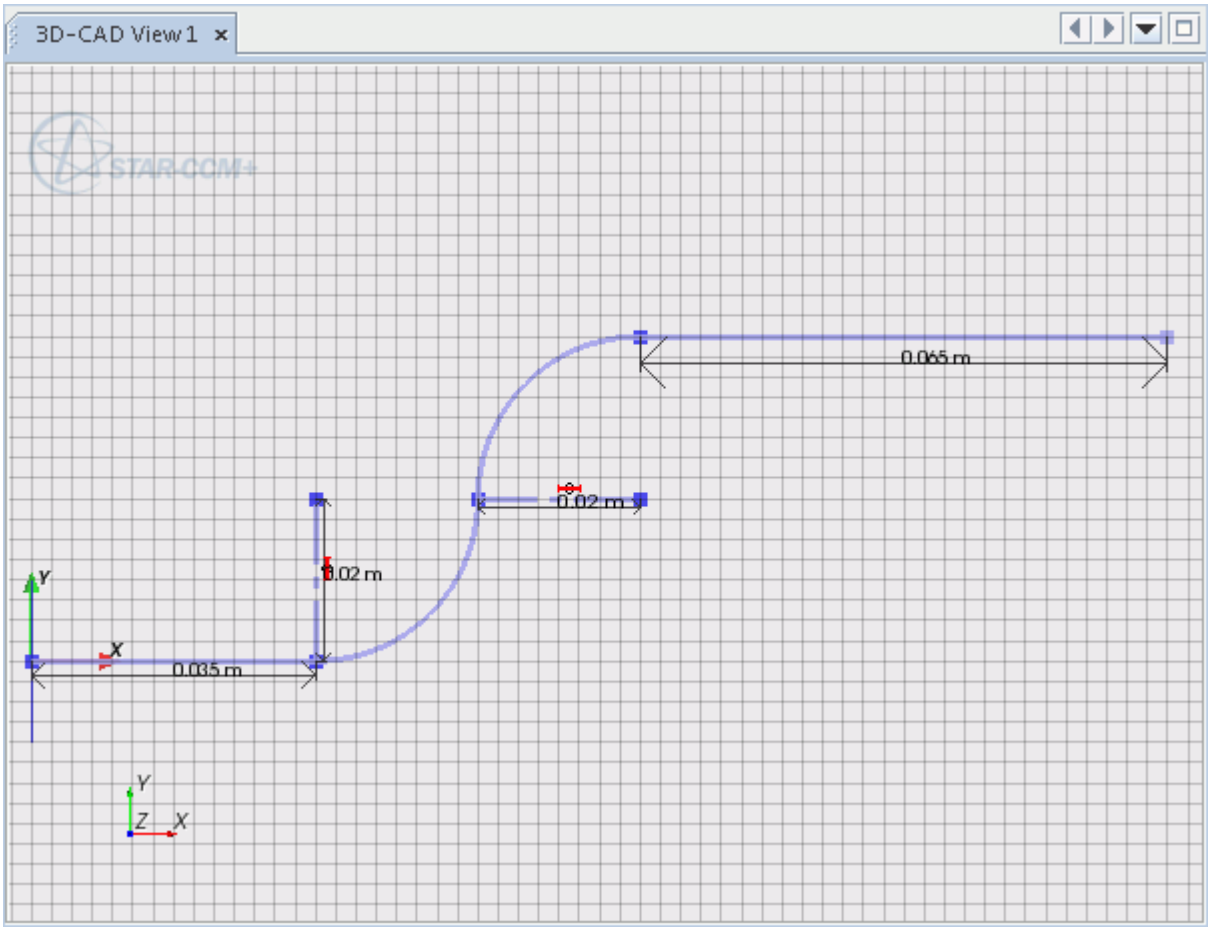


Draw the final section of the pipe.

12. Draw a line with start point `[0.075 m, 0.04 m]` that extends `0.065 m` in the positive x direction.



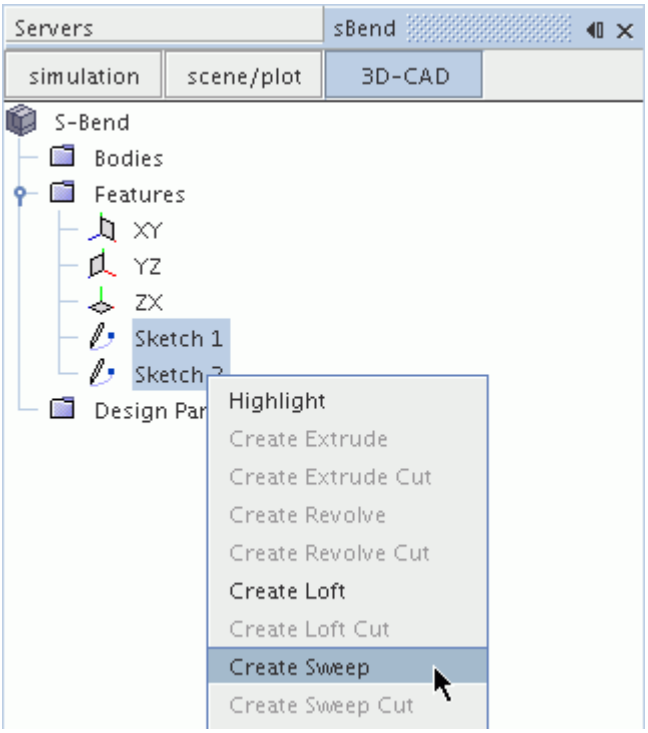
The completed sketch is shown below.



13. Click **OK** to exit the sketch.

Create the solid body of the pipe using the sweep feature. A sweep requires two sketches: one to act as the profile being swept; the other to act as the path for the sweep.

14. Select both sketches from the 3D-CAD object tree, holding down the Shift key to select multiple items.
15. Right-click one of the highlighted nodes and select **Create Sweep**.

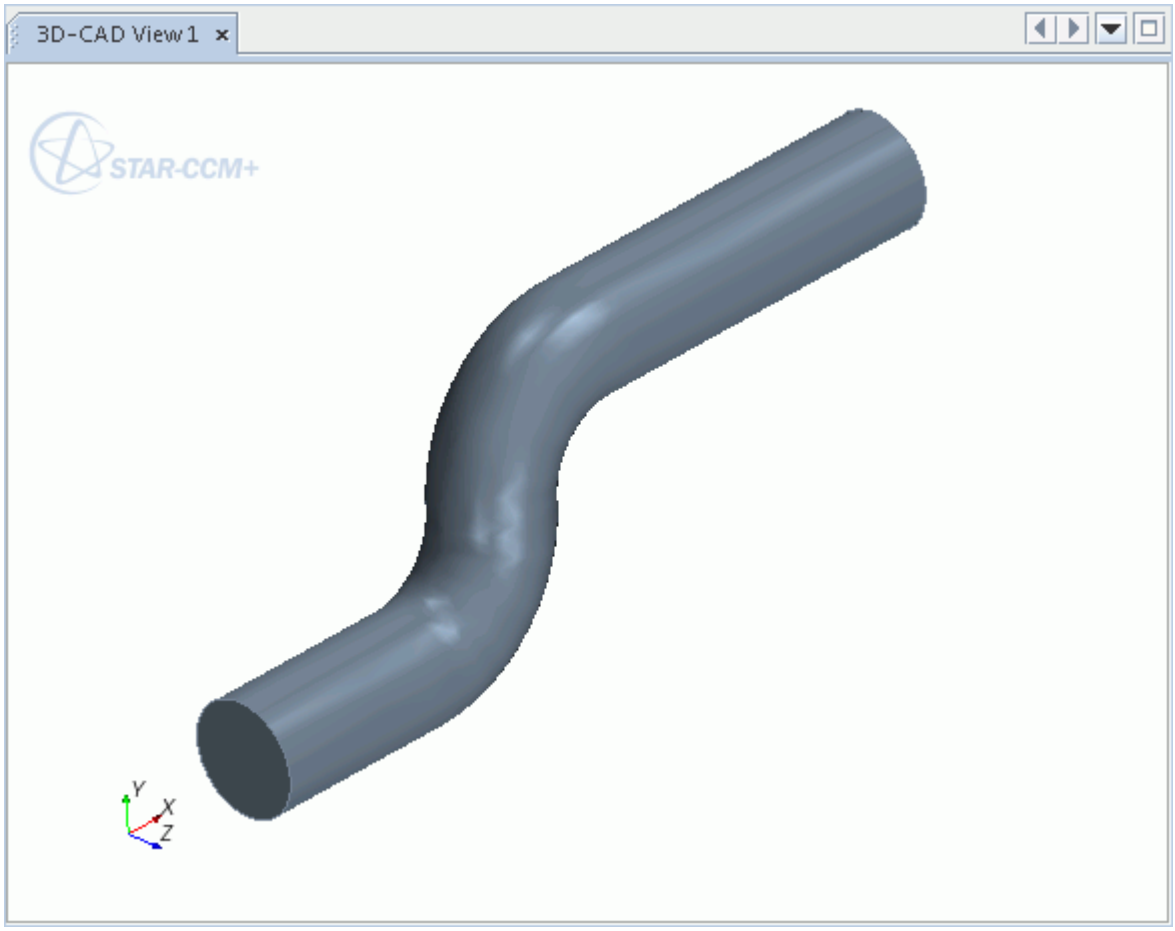


The *Sweep* dialog appears.

16. Accept the default settings and click **OK**.

A new body node, **Body 1**, is added below the **Bodies** node in the object tree.

17. Rename the **Bodies** > **Body 1** node to `Fluid`.
18. Position the geometry as shown below.



Specifying Inlet and Outlet Faces

The final step in preparing the model geometry is to specify the inlet and outlet faces of the model by setting face names.

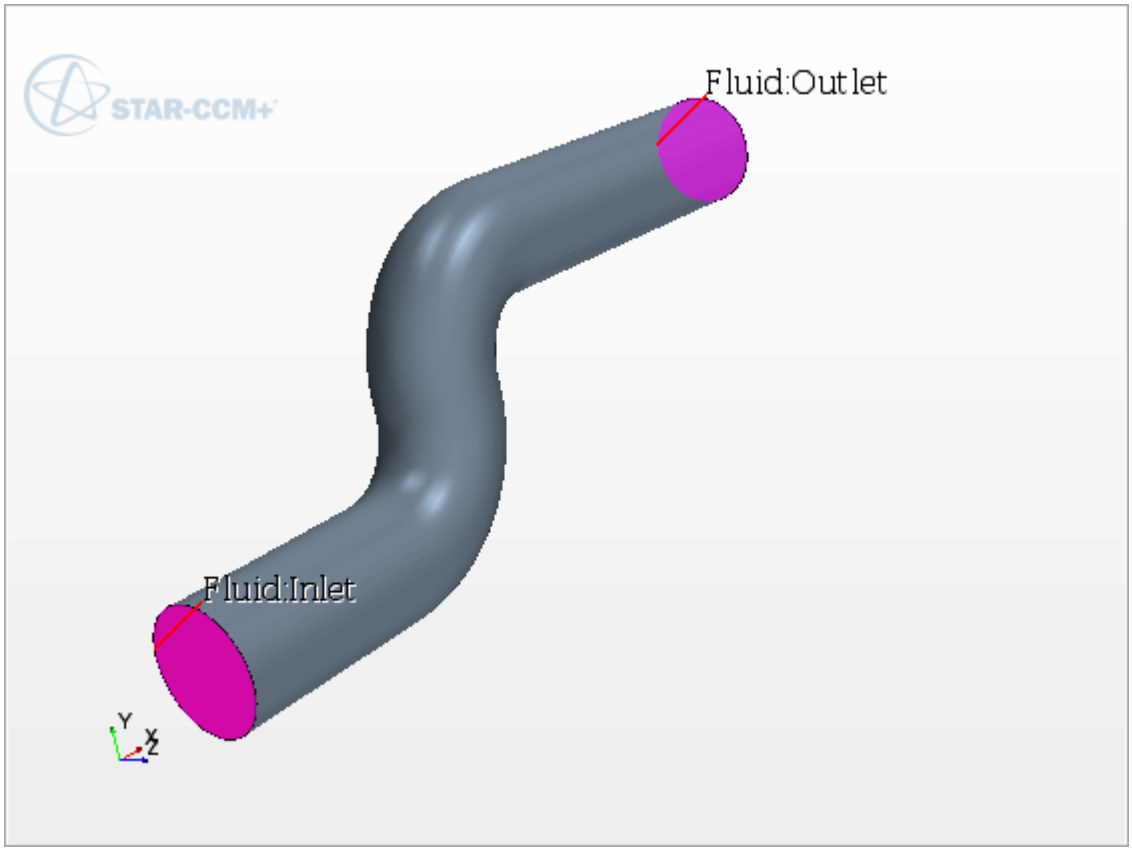
Use the following steps:

1. Zoom in to the pipe end, right-click the circular face and select **Rename**
2. Enter `Inlet` in the *Rename* dialog and click **OK**.

The benefit of renaming the face in the 3D-CAD model is that it retains its unique identity when the 3D-CAD model is converted to a geometry part.

To rename the outlet:

3. Position the geometry so that the outlet face on the other end of the pipe is visible.
4. Right-click the circular face at the other end of the pipe, select **Rename**, and enter Outlet in the *Rename* dialog. Click **OK** to complete the action.



The geometry is now complete and you can exit 3D-CAD.

5. Click the **Close 3D-CAD** button at the bottom of the object tree.

   Note: The *3D-CAD View* scene is closed automatically upon exiting 3D-CAD.

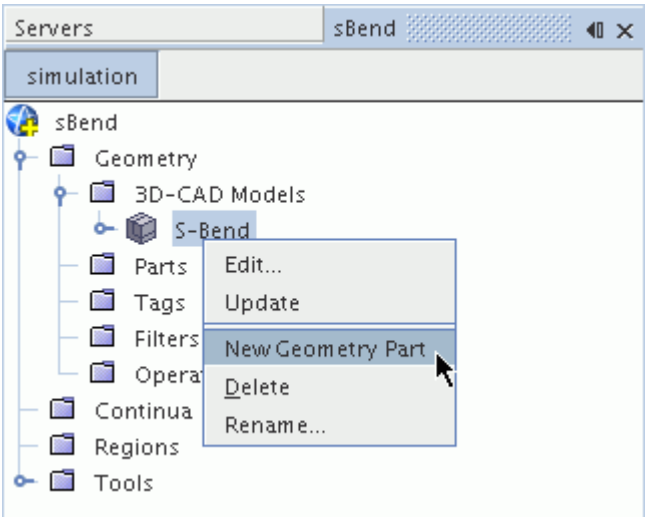
6. Save the simulation.

Creating a Geometry Part

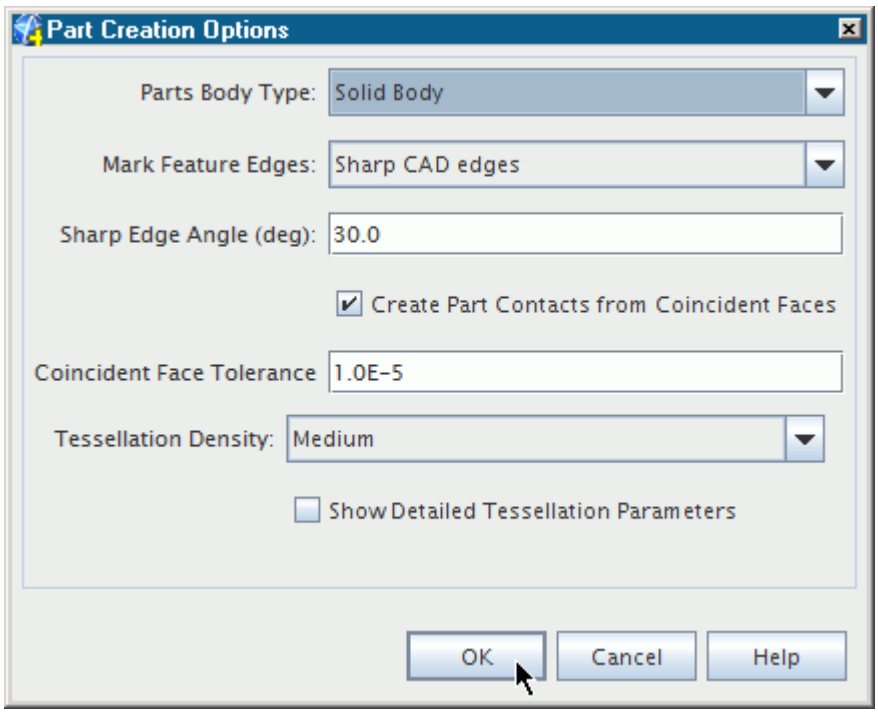
A new geometry part is created using the 3D-CAD model.

Use the following stesp:

1. Right-click the **Geometry** > **3D-CAD Models** > **S-Bend** node and select **New Geometry Part**.

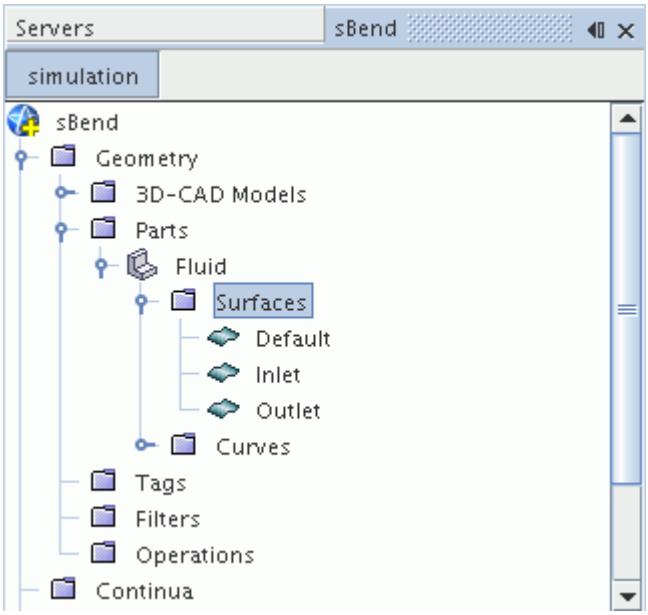


The *Part Creation Options* dialog appears.



2. Click **OK** to accept the default settings.
3. To see the surfaces, expand the **Parts** > **Fluid** > **Surfaces** node.

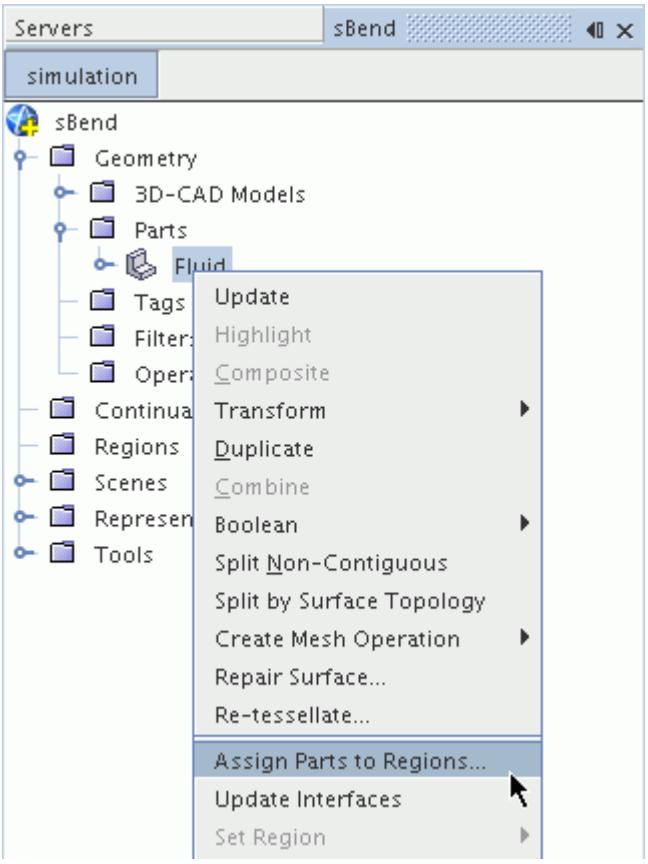
The inlet and outlet faces in Specifying Inlet and Outlet Faces are defined as separate surfaces.



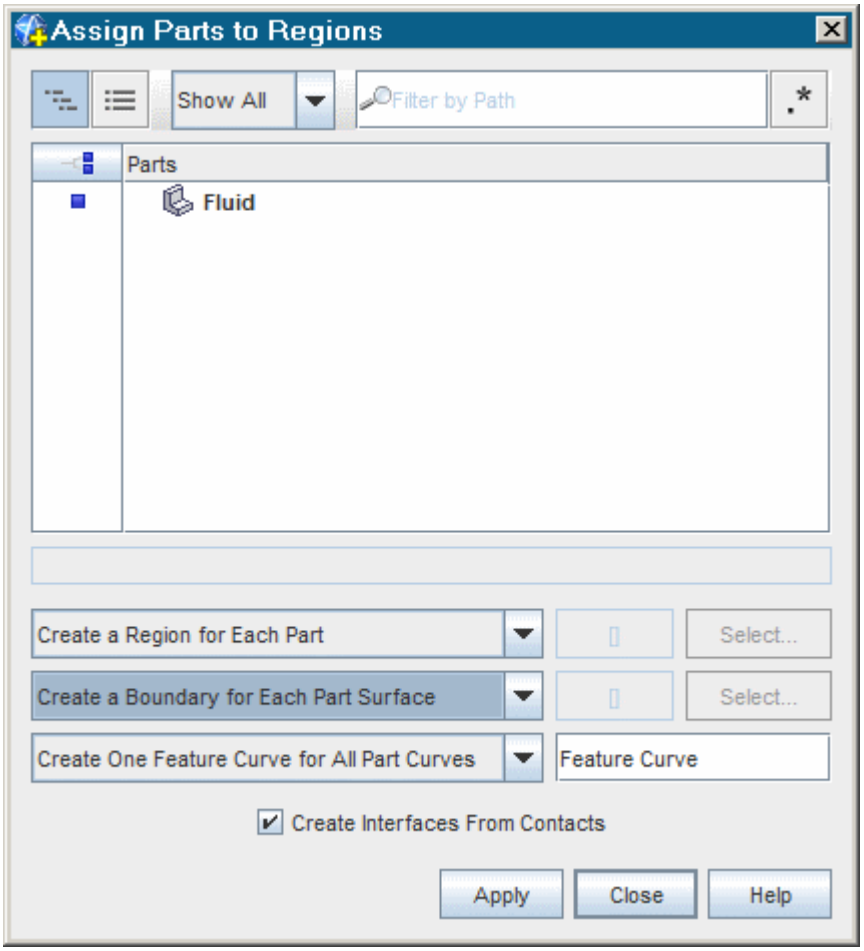
Assigning a Part to a New Region

Assign the geometry part to a region. The inlet and outlet surfaces have been renamed in 3D-CAD, and are automatically assigned to separate boundaries.

1. Right-click the **Parts** > **Fluid** node and select **Assign Parts to Regions**.



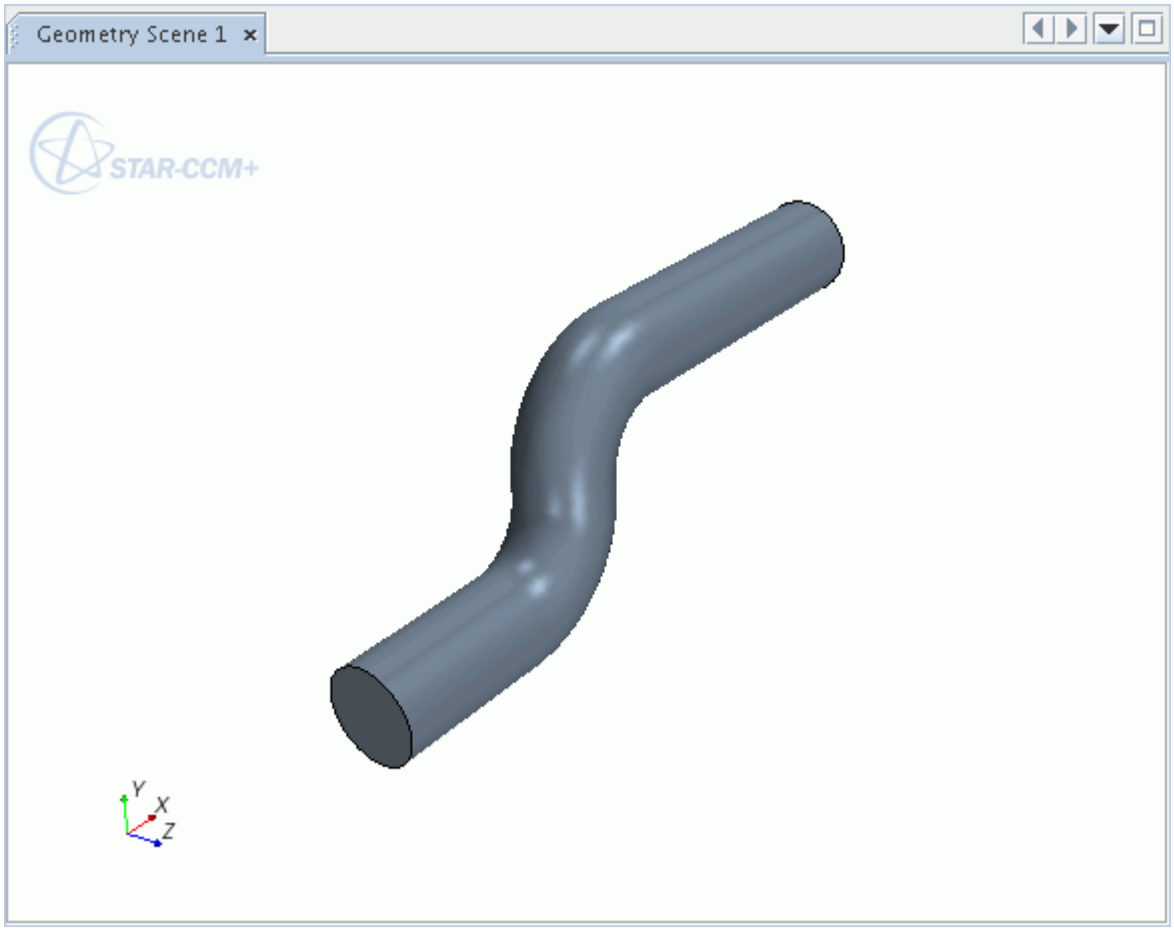
2. In the *Assign parts to region* dialog set:
   - **Region Mode** (top drop-down) to **Create a Region for Each Part**.
   - **Boundary Mode** (middle drop-down) to **Create a Boundary for Each Part Surface**.



3. Click **Apply** and then **Close**.

A new region has been created with boundaries defining the inlet, outlet, and main fluid volume.

4. Create a geometry scene. Use the mouse to rotate the model in the *Graphics* window to set up the view shown below.



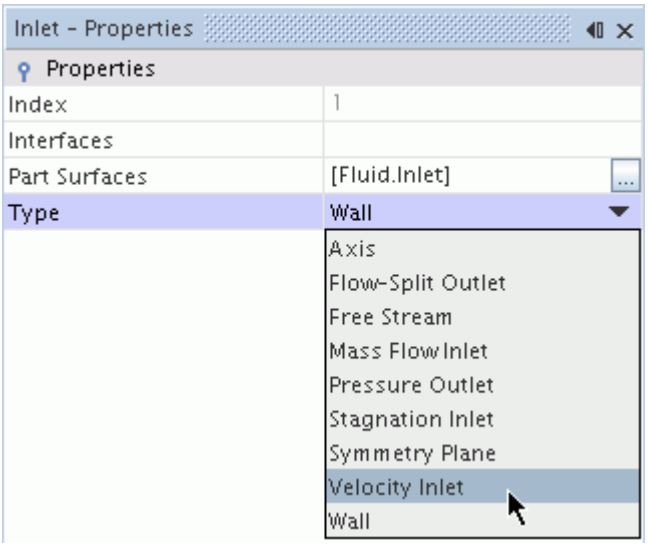
5. Expand the **Regions** > **Fluid** > **Boundaries** node and select each of the boundary nodes to make sure that they have been specified correctly.

Setting Up the Case

In this first part of the tutorial, run the simulation using a steady state laminar flow (Re = 500).

By default, all boundaries are set to be walls. Modify the boundary type for the **Inlet** and **Outlet**:

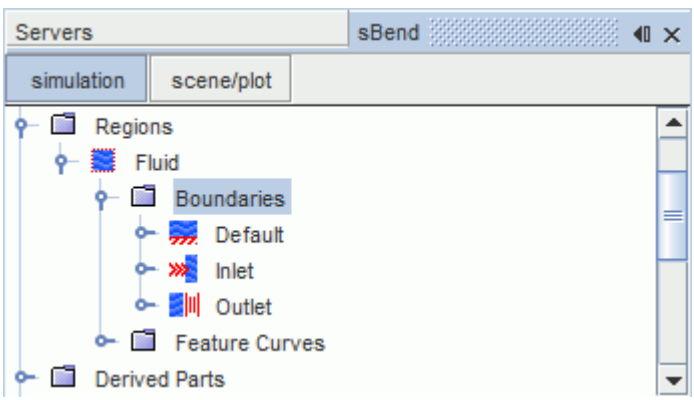
1. Select the **Regions** > **Fluid** > **Boundaries** > **Inlet** node and set the *Type* to **Velocity Inlet**.



The icon next to the **Inlet** boundary changes, indicating that it is now a velocity inlet boundary.

2. Select the **Outlet** boundary node and set the *Type to* **Pressure Outlet**.

The modified **Boundaries** node is shown below.



Generating a Surface Mesh

Generate a polyhedral mesh using the generalized cylindrical mesher.

This meshing model is suited to pipe flow. Define a prism layer mesh along the wall boundary, to assure that near wall effects are adequately resolved.

1. Right-click the **Continua** node and select **New** > **Mesh Continuum**.
2. Right-click the **Mesh 1** > **Models** node and select the following models, in order:

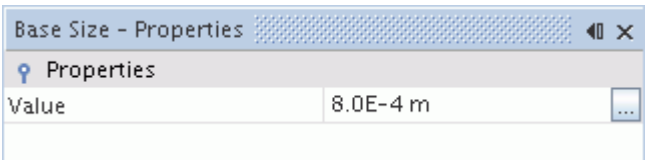
Group Box	Model
Surface Mesh	**Surface Remesher**
Volume Mesh	**Polyhedral Mesher**

Group Box	Model
Optional Models	**Generalized Cylinder**
	Prism Layer Mesher

3. Click **Close**.

To define the mesh settings:

4. Select the **Mesh 1** > **Reference Values** > **Base Size** node.
5. Set the *Value* to `0.0008` m.

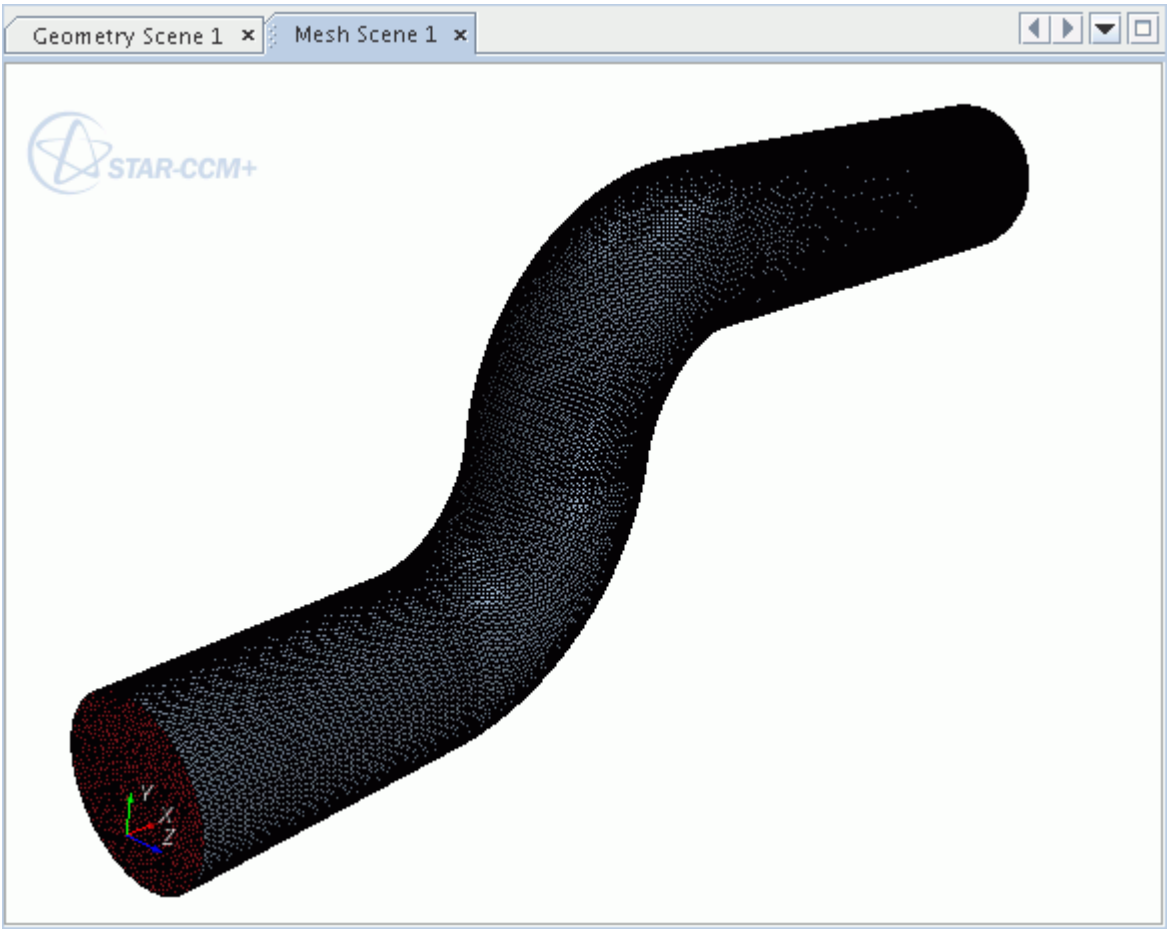


6. Continuing with the reference values section, set the following properties:

Node	Property	Value
Number of Prism Layers	*Number of Prism Layers*	`5`
Prism Layer Stretching	*Prism Layer Stretching*	`1.8`
Prism Layer Thickness	*Percentage of Base*	`45`

Generate and check the surface mesh before running the generalized cylinder mesher. A good quality surface mesh is needed before the volume mesh can be generated.

7. Click (**Generate Surface Mesh**).
8. Create a mesh scene and examine the surface mesh.

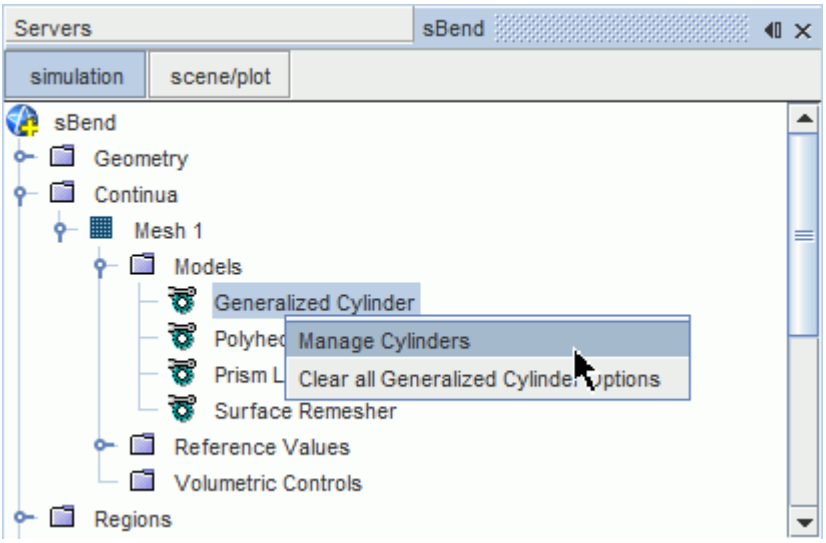


Generating a Volume Mesh

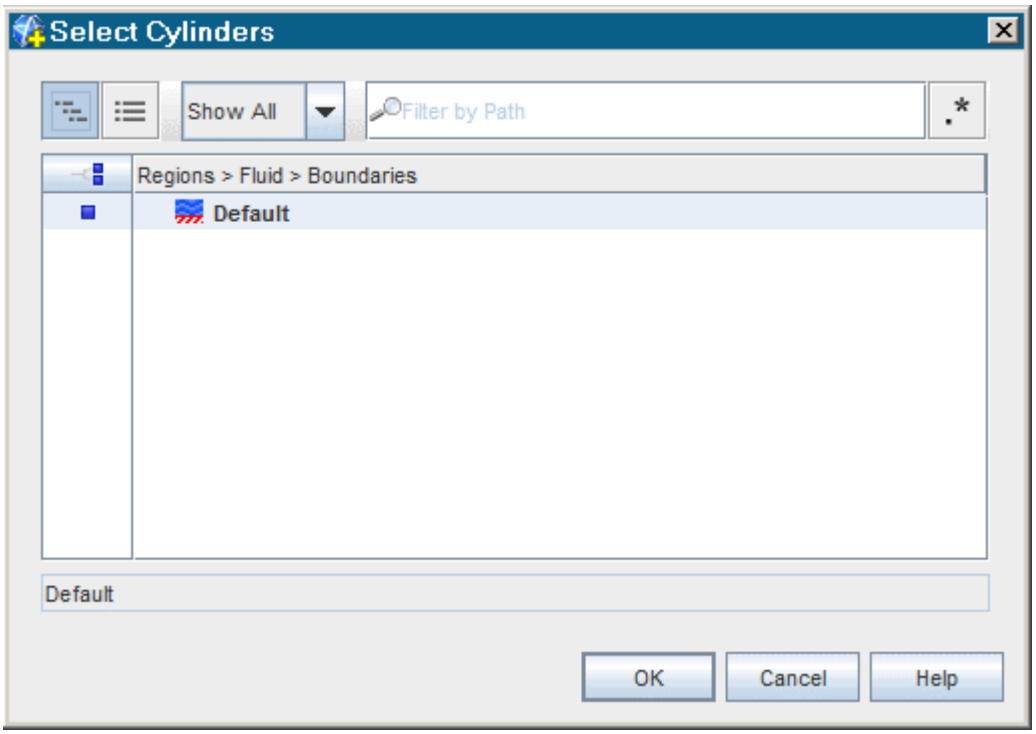
Before generating the volume mesh, define the boundaries that form the cylindrical geometry.

Use the following steps:

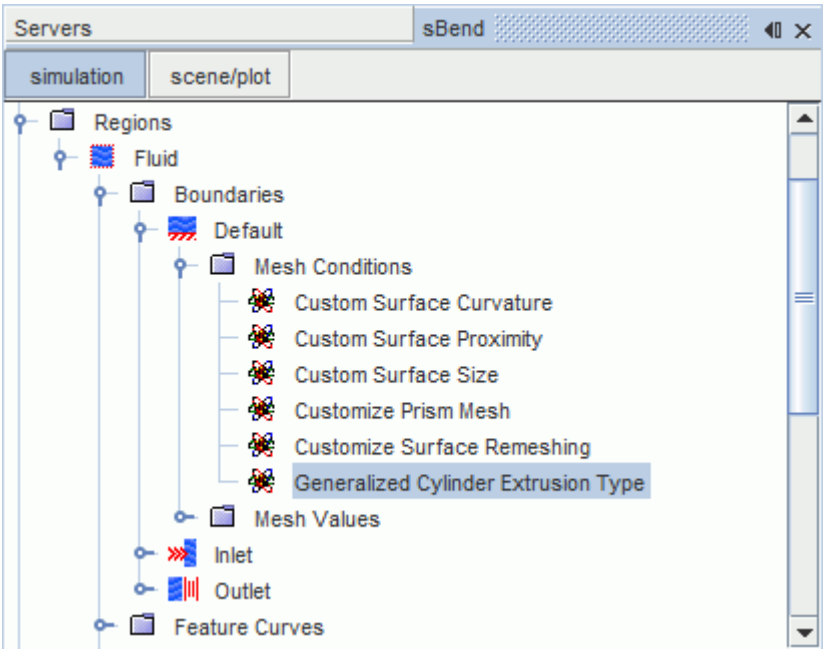
1. Right-click the **Mesh 1** > **Models** > **Generalized Cylinder** and select **Manage Cylinders**.



The *Select Cylinders* dialog appears.



2. Select the **Default** cylinder, then click **OK**.
3. Select the **Regions** > **Fluid** > **Boundaries** > **Default** > **Mesh Conditions** > **Generalized Cylinder Extrusion Type** node.



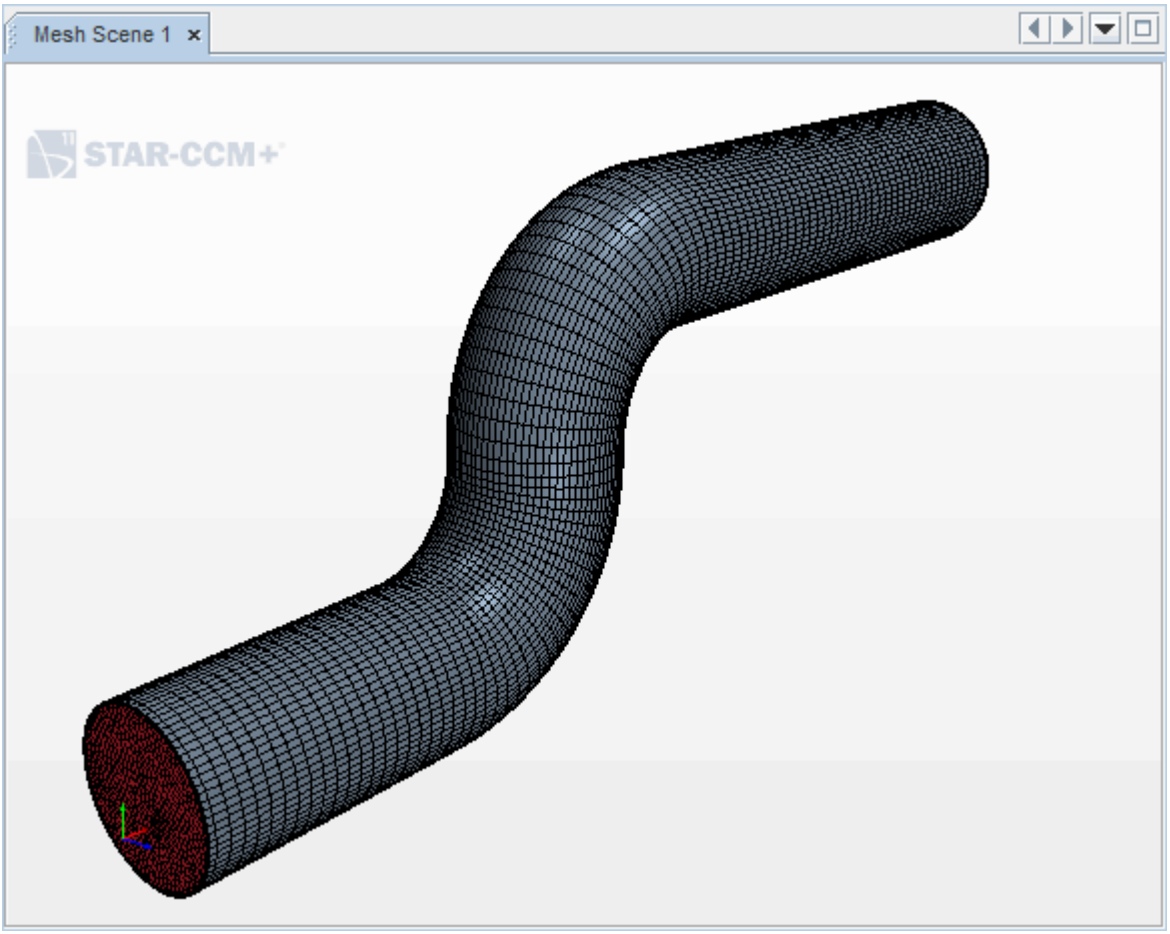
Verify that the *Extrusion law* property was set to **Constant.**

4. Within the same **Default** boundary, select the **Default** > **Mesh Values** > **Generalized Cylinder Parameters** node.

Notice that the *Number of Layers* property was set to `103`. STAR-CCM+ calculates this value from the geometry model.

The volume mesh can now be generated.

5. Click the (**Generate Volume Mesh**) button.
6. Right-click a blank space in *Mesh Scene 1* and select **Apply Representation** > **Volume Mesh**.
7. Examine the final volume mesh.



8. Save the simulation.

Selecting Physics Models

Select the models for the incompressible fluid.

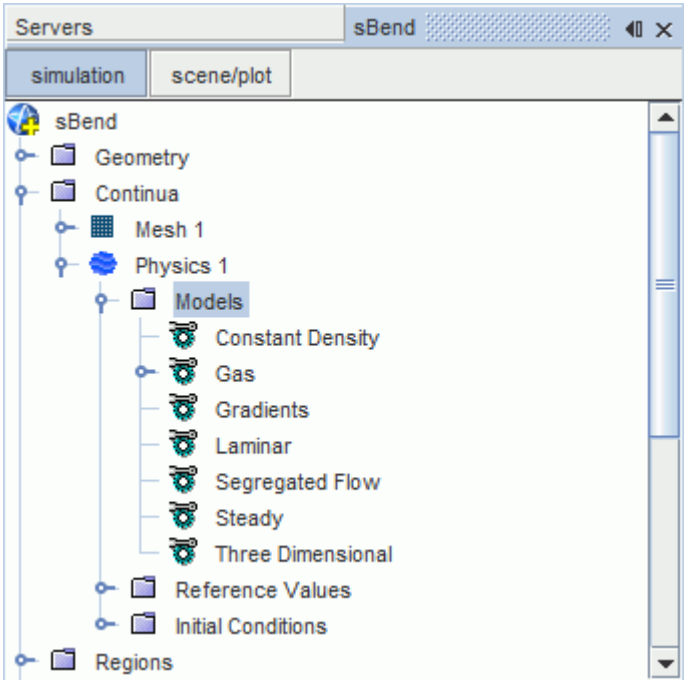
A physics continuum was automatically created during the mesh generation process. To select the physics models:

1. Right-click the **Continua** > **Physics 1** > **Models** node and select the following models, in order:

Group Box	Model
Time	**Steady**
Material	**Gas**
Flow	**Segregated Flow**
Equation of State	**Constant Density**
Viscous Regime	**Laminar**

2. Click **Close**.

The selected models now appear.



Modifying Material Properties

Set the values for the dynamic viscosity and density of air.

Use the following steps:

1. Select the **Models** > **Gas** > **Air** > **Material Properties** > **Density** > **Constant** node and set the *Value* to `1 Kg/m^3`.
2. Select the **Dynamic Viscosity** > **Constant** node and set the *Value* to `1.716E-5 Pa-s`.

Setting Initial Conditions and Boundary Settings

To achieve a Reynolds number of 500 for this gas and pipe diameter, you require a mean velocity of 0.429 m/s.

To assist solution convergence, define the initial condition for velocity:

1. Select the **Physics 1** > **Initial Conditions** > **Velocity** node and set the *Value* to `[0.429, 0.0, 0.0] m/s`.

Set the same velocity on the **Inlet** boundary.

2. Select the **Regions** > **Fluid** > **Boundaries** > **Inlet** > **Physics Values** > **Velocity Magnitude** node and set the *Value* to `0.429 m/s`.

Preparing a Scalar Scene

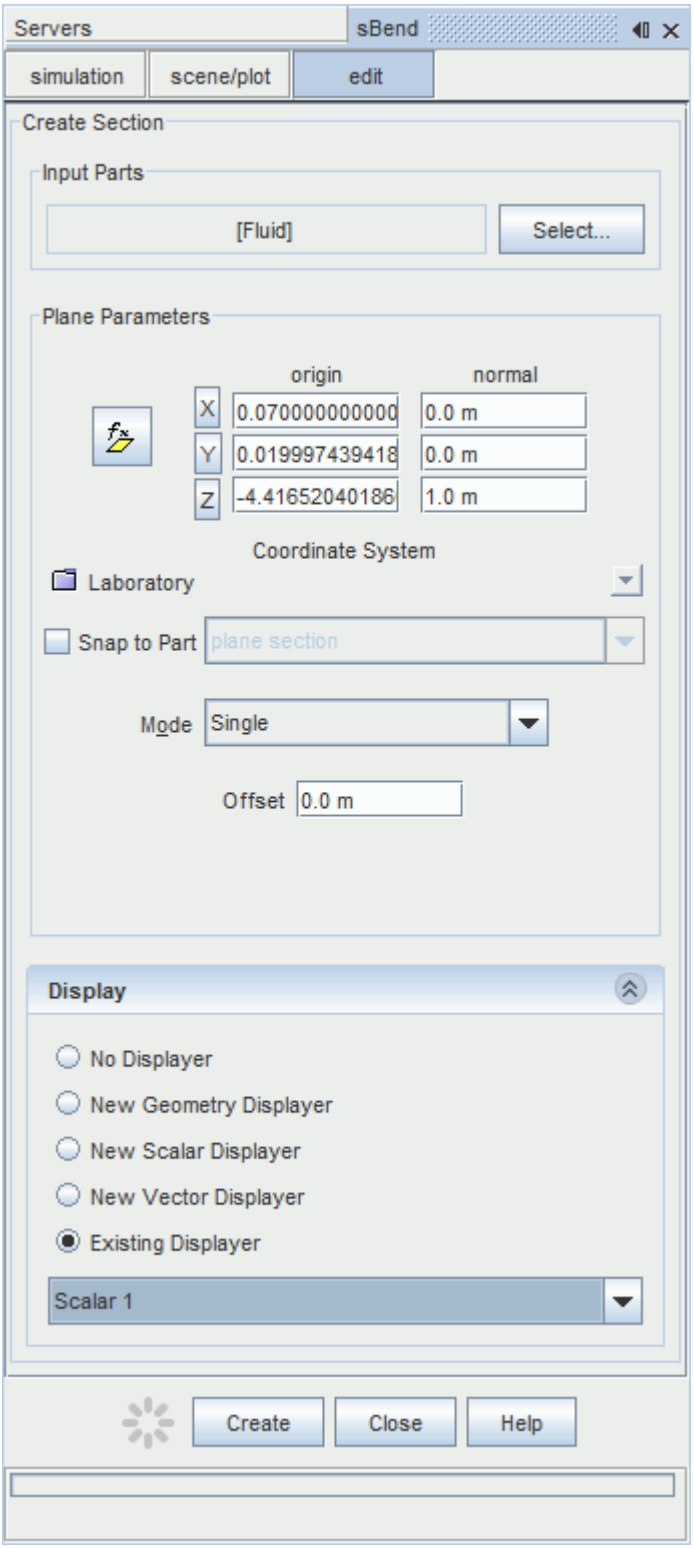
A scalar scene is created to show the velocity magnitude on a section plane through the center of the s-bend. Use this scene to visualize the solution while the simulation is running.

Use the following steps:

1. Create a scalar scene.
2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane**
3. In the *Create Section* panel, set the following values:

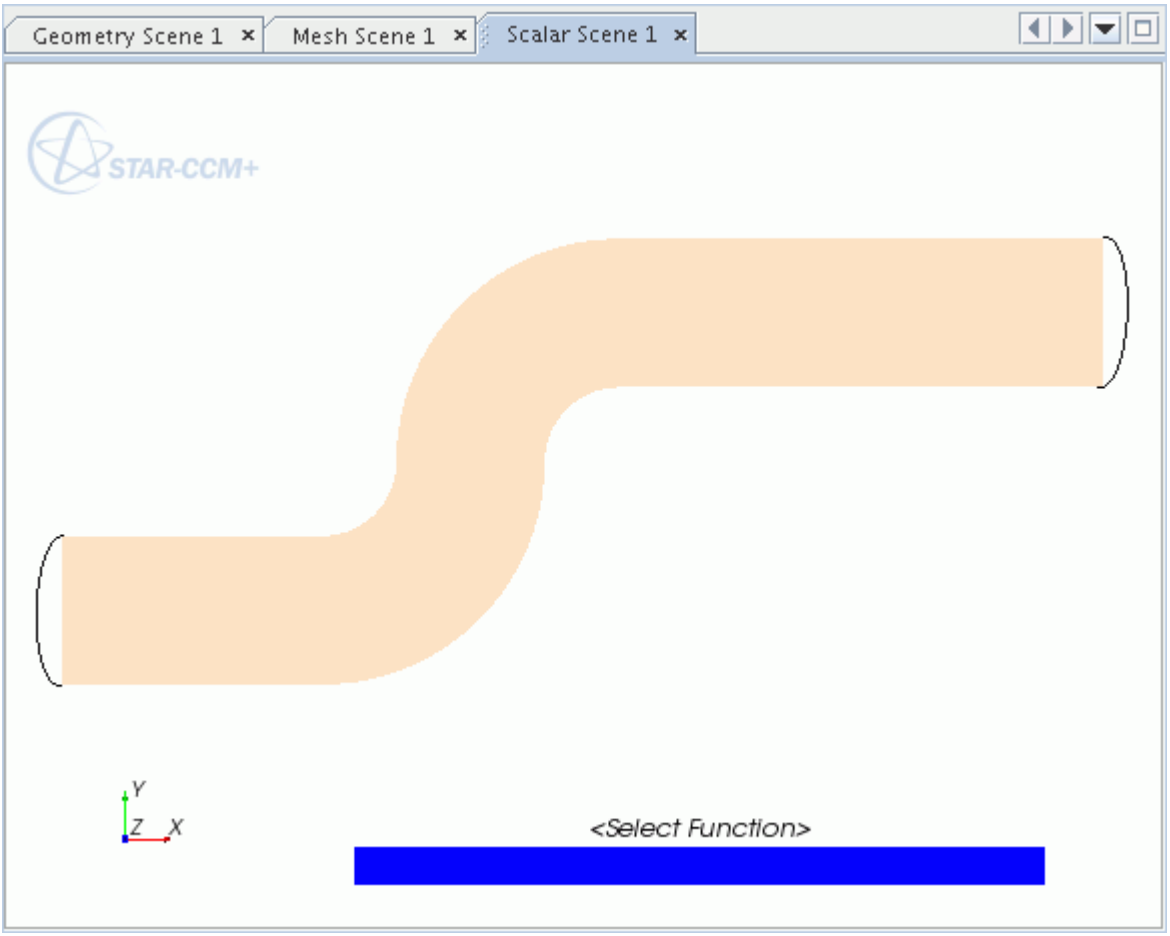
Property	Value
Input Parts	**Fluid**
normal	`[0, 0, 1]` m
Display	**Existing Displayer**
Dropdown menu	**Scalar 1**

The completed dialog is shown below.



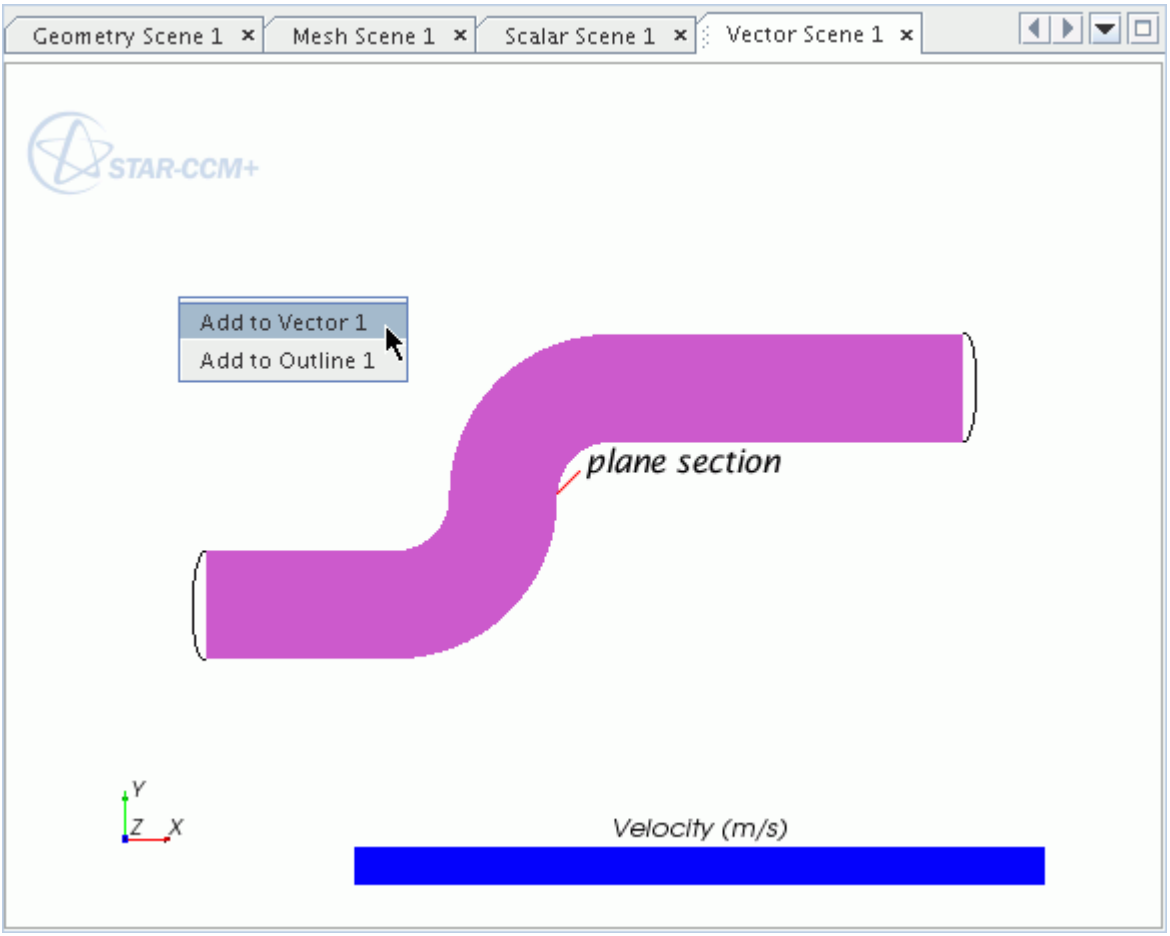
4. Click **Create** and then **Close**.

   Scalar Scene 1 is shown below.



5. Click the **scene/plot** button.
6. Select the **Displayers** > **Scalar 1** > **Scalar Field** node and set the *Function* to **Velocity** > **Magnitude**.
7. Select the **Displayers** > **Scalar 1** node and set the *Contour Style* to **Smooth Filled**.
8. Click the **Simulation** button to return to the STAR-CCM+ simulation object tree.

   A vector plot displaying the velocity magnitude is created.
9. Create a vector scene.
10. Using the drag-and-drop method, add the section plane from the **Derived Parts** node to the scene.



Setting Up Stopping Criteria

Limit the number of iterations for the solver to 500.

500 iterations are sufficient for the solution to converge. Use the following steps:

1. Select the **Stopping Criteria** > **Maximum Steps** node and set the *Maximum Steps* to `500`.
2. Save the simulation.

Running the Simulation

The simulation is now ready to be run.

Use the following steps:

1. Click (**Run**).

   While the simulation is running, you can click the tabs at the top of the *Graphics* window to view each of the scalar scenes.

During the run, it is also possible to stop the process by clicking (**Stop**) on the toolbar. If you do halt the simulation, it can be continued later by clicking (**Run**). If left alone, the simulation continues until 500 iterations are complete.

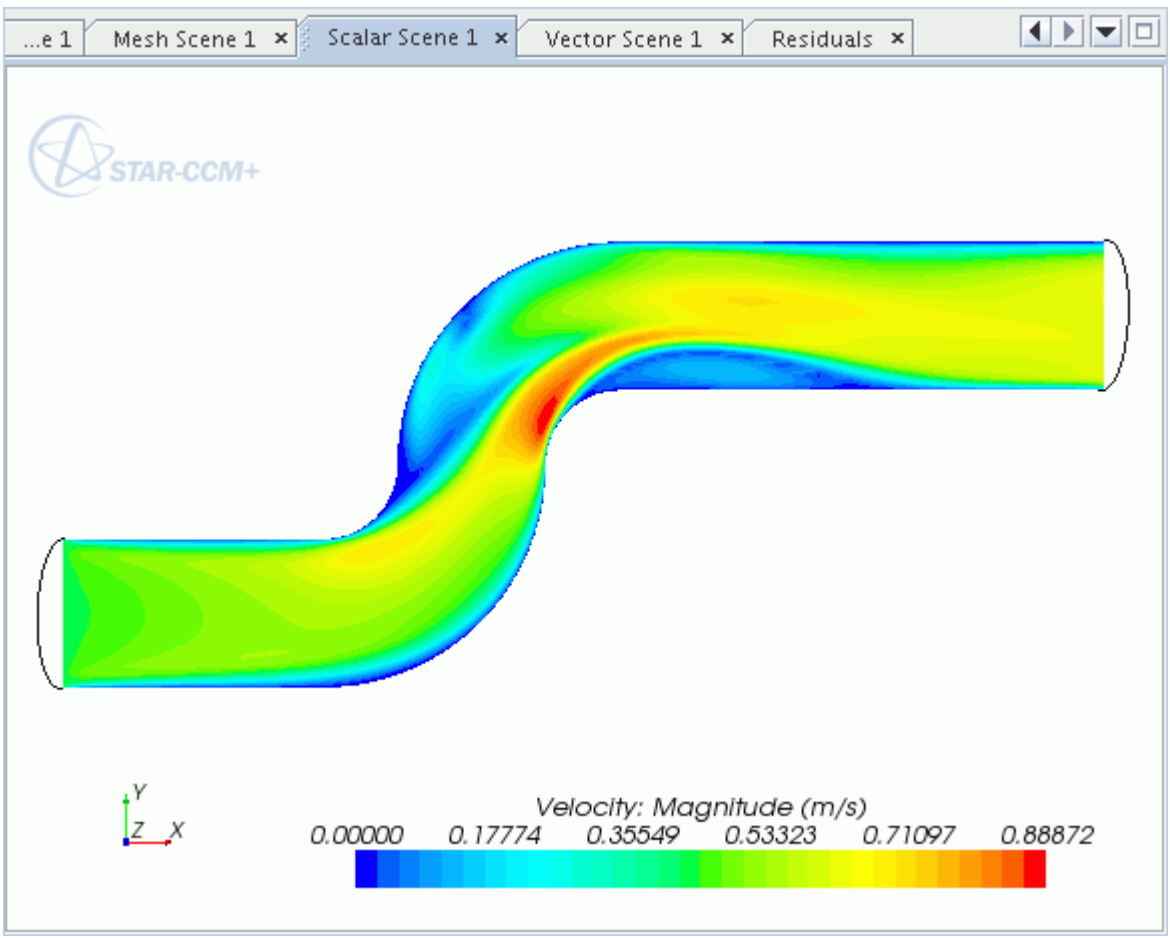
The Residuals display is created automatically and shows the progress of the solvers.

2. While the simulation is running, select the *Scalar Scene 1* tab at the top of the *Graphics* window to visualize the solution.
3. When the simulation has finished running, save it.

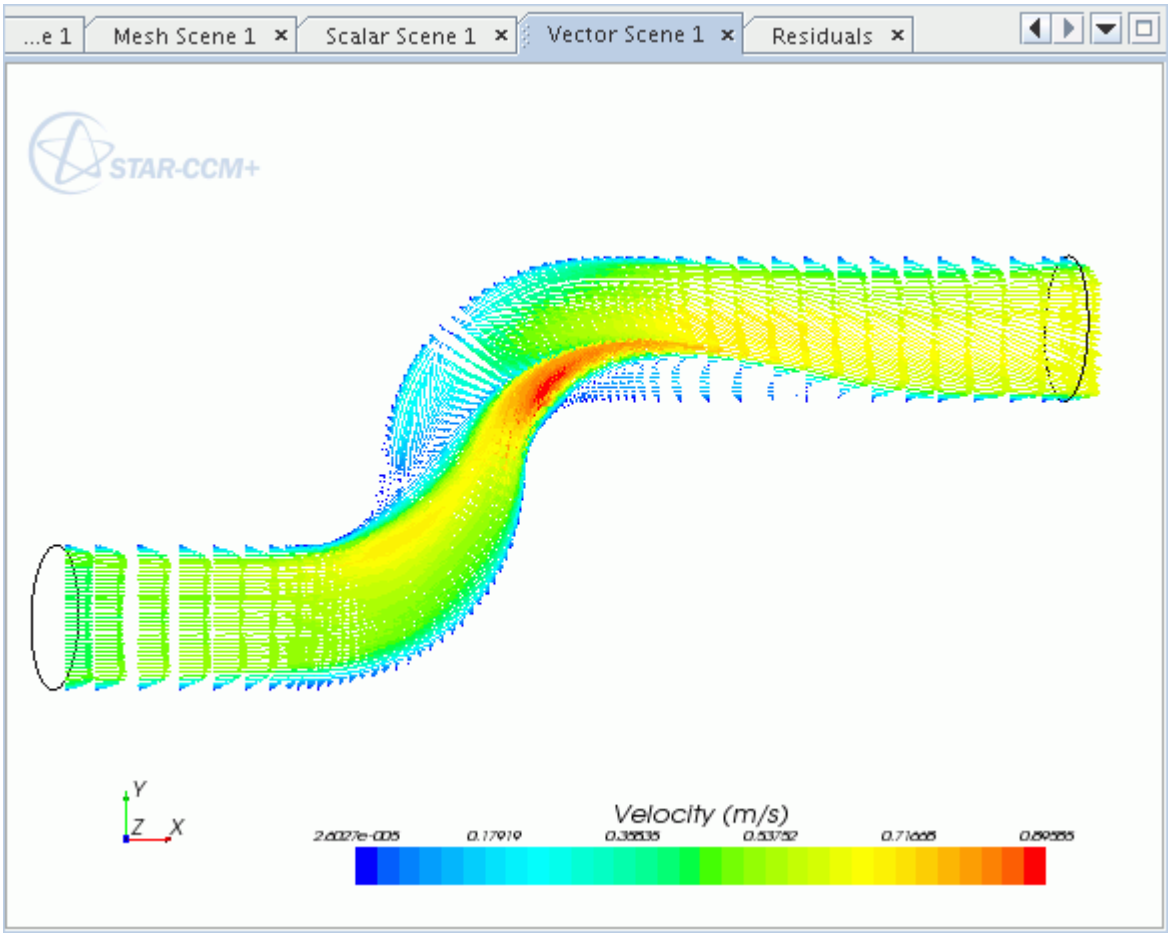
Visualizing the Results

Examine the scalar and vector scenes.

The scalar scene is shown below.



The vector scene is shown below.



Changing to a Turbulent Flow

Edit the models, initial conditions, and boundary conditions to simulate a turbulent flow (Reynolds number = 50,000).

First, save the state of the simulation as developed so far:

- Save the simulation as `sBendTurbulent.sim`.

Contents:

Modifying the Physics Continuum
Modifying Initial Conditions and Boundary Settings
Extending the Stopping Criteria
Clearing the Solution and Running the Simulation
Visualizing the Results for the Turbulent Flow

Modifying the Physics Continuum

Edit the physics models to represent turbulent flow.

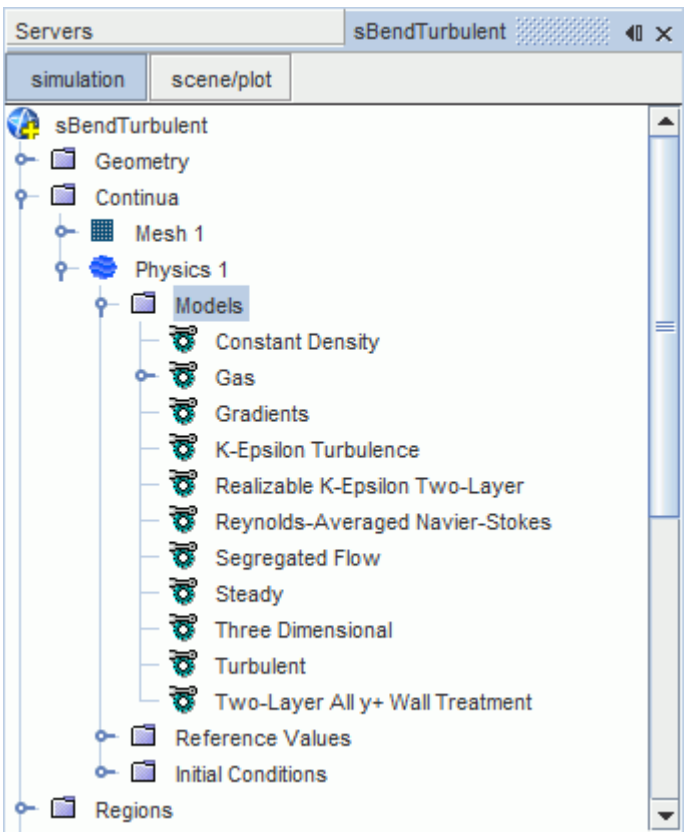
Disable the laminar model and enable the turbulent and K-epsilon models.

1. For the physics continuum **Continua** > **Physics 1**, deselect **Laminar** in the *Enabled Models* box.
2. Select the following models, in order:

Group Box	Model
Viscous Regime	**Turbulent**
	Reynold-Averaged Navier-Stokes (Selected automatically)
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**
	Realizable K-Epsilon Two-Layer (Selected automatically)
	Two-Layer All y+ Wall Treatment (Selected automatically)

3. Click **Close**.

The selected models now appear.



Modifying Initial Conditions and Boundary Settings

To achieve a turbulent flow (Re = 50,000) in the existing pipe, increase the velocity of the air to 42.9 m/s at the **Initial Conditions** node and at the **Inlet** boundary condition. Change the **Turbulence Specification** so that the turbulence intensity and length scale can be defined.

Use the following steps:

1. Select the **Continuum** > **Physics 1** > **Initial Conditions** > **Velocity** node and set the *Value* to `[42.9, 0.0, 0.0] m/s`.
2. Select the **Turbulence Specification** node and set its *Method* to **Intensity + Length Scale**.

   The **Turbulent Length Scale** node is added to the object tree.
3. Set the following values:

Node	Property	Value
Turbulent Length Scale	*Value*	`0.0014 m` The turbulent length scale that is used in this case is approximately 7% of the hydraulic diameter, which for circular pipes is the same as the pipe diameter.
Turbulent Velocity Scale	*Value*	`4.29 m/s` The turbulent velocity scale that is used in this case is 10% of the free-stream velocity.
Turbulence Intensity	*Value*	`0.12` The turbulence intensity is approximately 12% and was calculated using the following equation: $$I = \sqrt{\frac{\frac{3}{2}v_t^2}{U^2}}$$ (4158) where I is the turbulence intensity, v is the turbulent velocity scale, and U is the free-stream velocity.

Use the same values at the **Inlet** boundary. Start by setting the **Turbulence Specification** to use **Intensity + Length Scale**.

4. Select the **Regions** > **Fluid** > **Boundaries** > **Inlet** > **Physics Conditions** > **Turbulence Specification** node and set the *Method* to **Intensity + Length Scale**.
5. Set the following values:

Node	Property	Value
Physics Values > **Turbulent Length Scale**	*Value*	`0.0014 m`
Regions > **Fluid** > **Boundaries** > **Inlet** > **Physics Values**		
└ **Velocity Magnitude**	*Value*	`42.9 m/s`
└ **Turbulence Intensity**	*Value*	`0.12`

Extending the Stopping Criteria

The turbulent flow requires a longer time to converge. Increasing the number of iterations to 1000 gives the solver sufficient time to converge.

1. Select the **Stopping Criteria** > **Maximum Steps** node and set the *Maximum Steps* to `1000`.
2. Save the simulation.

Clearing the Solution and Running the Simulation

Clear the solution from the laminar flow before running the solver.

Use the following steps:

1. Select **Solution** > **Clear Solution** from the menu.

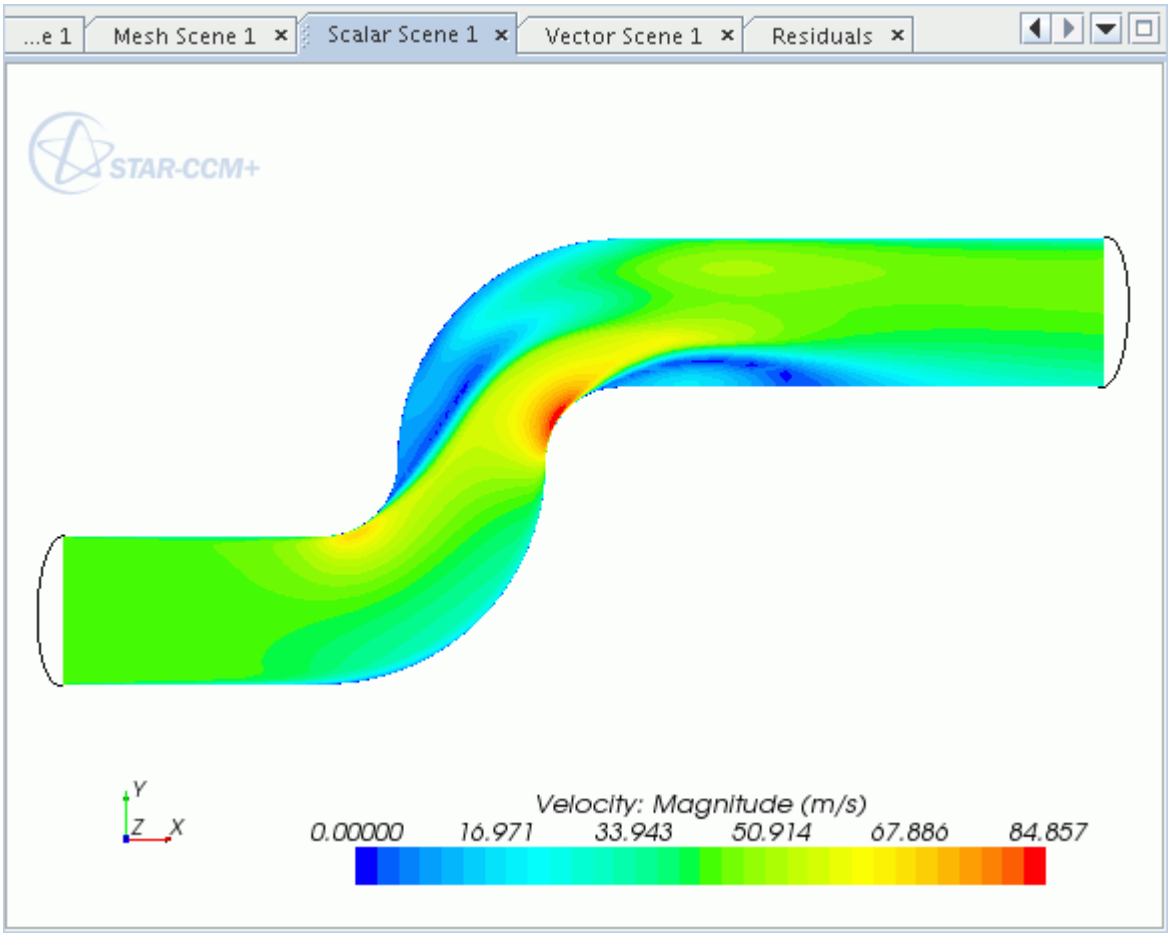
   The *Clear Solution* dialog appears.
2. Accept the default settings and click **OK**.

   The simulation is now ready to be run.
3. Click (**Run**).
4. While the simulation is running, select the *Scalar Scene 1* tab at the top of the *Graphics* window to visualize the solution.
5. When the simulation has finished running, save it.

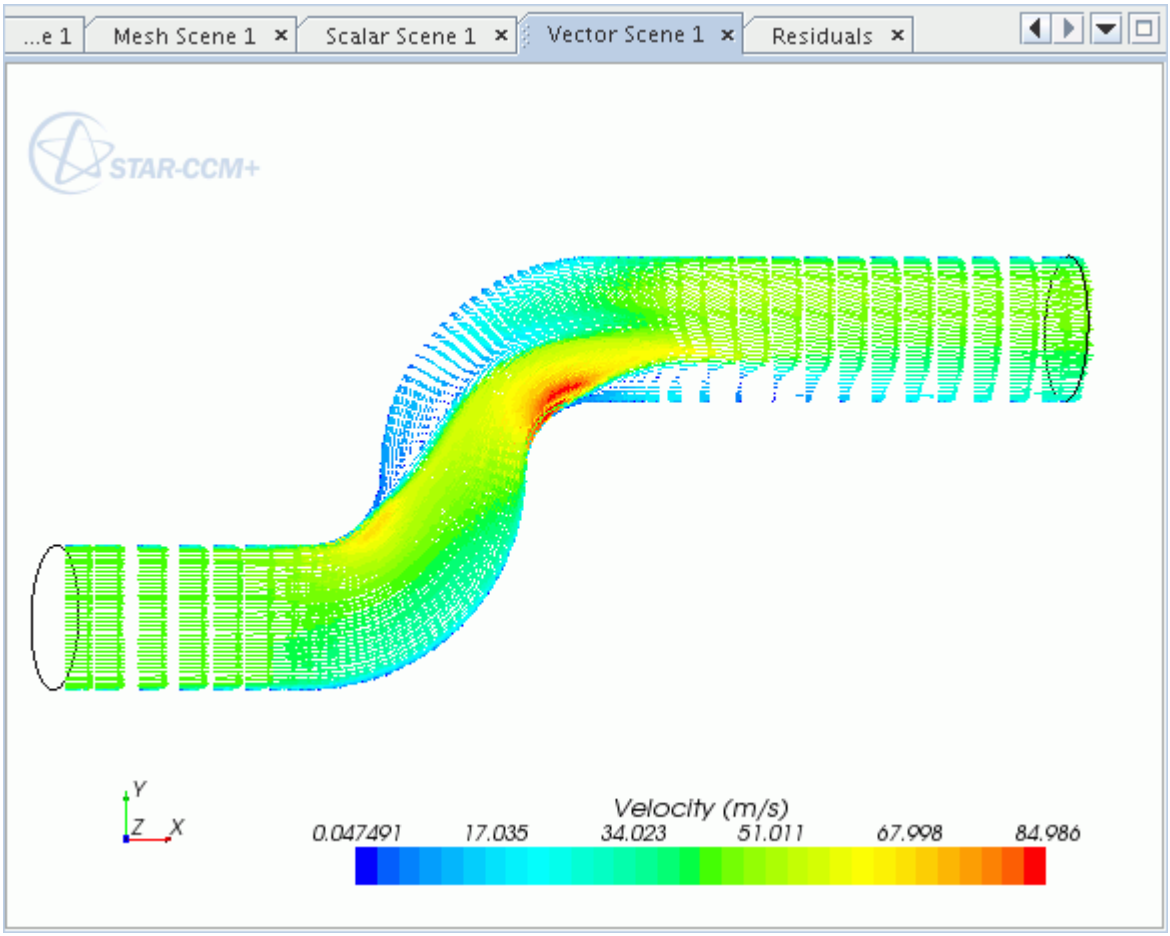
Visualizing the Results for the Turbulent Flow

Examine the scalar and vector scenes.

The scalar scene is shown below.



The vector scene is shown below.



Suggestions for Further Practice

Try running the simulation with different initial and boundary conditions.

To see how the laminar and turbulent models differ, you could try running the Re = 50,000 case with the laminar physics model.

1. Open the `sBend.sim` file.
2. Increase the flow velocity to `42.9 m/s` in the **Initial Conditions** and at the **Inlet** boundary.
3. Clear the solution and run the simulation.

The residuals do not converge, but instead start to oscillate which indicates that the model setup is incorrect.

Summary

This tutorial examined the flow of an incompressible fluid through an s-bend of constant diameter, at different Reynolds numbers.

This tutorial covered the following topics:

- Creating a geometry model in 3D-CAD

- Using the generalized cylinder mesher
- Solving a laminar flow through the pipe
- Solving a turbulent flow through the pipe
- Plotting results of each case

Steady Flow: Backward Facing Step

This tutorial demonstrates the flow of an incompressible gas over a backward facing step (with height, H = 1 m).

The turbulent flow has a Reynolds number of 28,000 (based on the step height, H, and the free stream velocity). The K-Epsilon turbulence model with High y+ wall treatment is used as it allows you to use a relatively coarse mesh without prism layers. The bottom wall is also heated with a heat flux of 500 W/m^2. The geometry of the backward facing step is shown below.

Contents:

Prerequisites
Setting Up the Geometry
Setting Up the Backward Facing Step Simulation
Summary
Backward Facing Step Bibliography

Prerequisites

The instructions in the Backward Facing Step tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Techniques	Associated Tutorial
The STAR-CCM+ workflow	Introduction
Creating geometry in 3D-CAD	3D-CAD: Cyclone Separator
Using visualization tools, scenes and plots	Introduction

Setting Up the Geometry

Create the model for the backward-facing step.

1. Start up STAR-CCM+ in a manner that is appropriate to your working environment and create a New Simulation.
2. Save the new simulation with the file name `backwardFacingStep.sim`

The geometry is created using 3D-CAD.

3. Right-click the **Geometry** > **3D-CAD Models** node and select **New**.

   This creates a model and activates 3D-CAD.

Start by renaming the 3D-CAD model.

4. Select the **3D-CAD Model 1** node at the top of the object tree. Rename the model to `Fluid Domain`.

Contents:

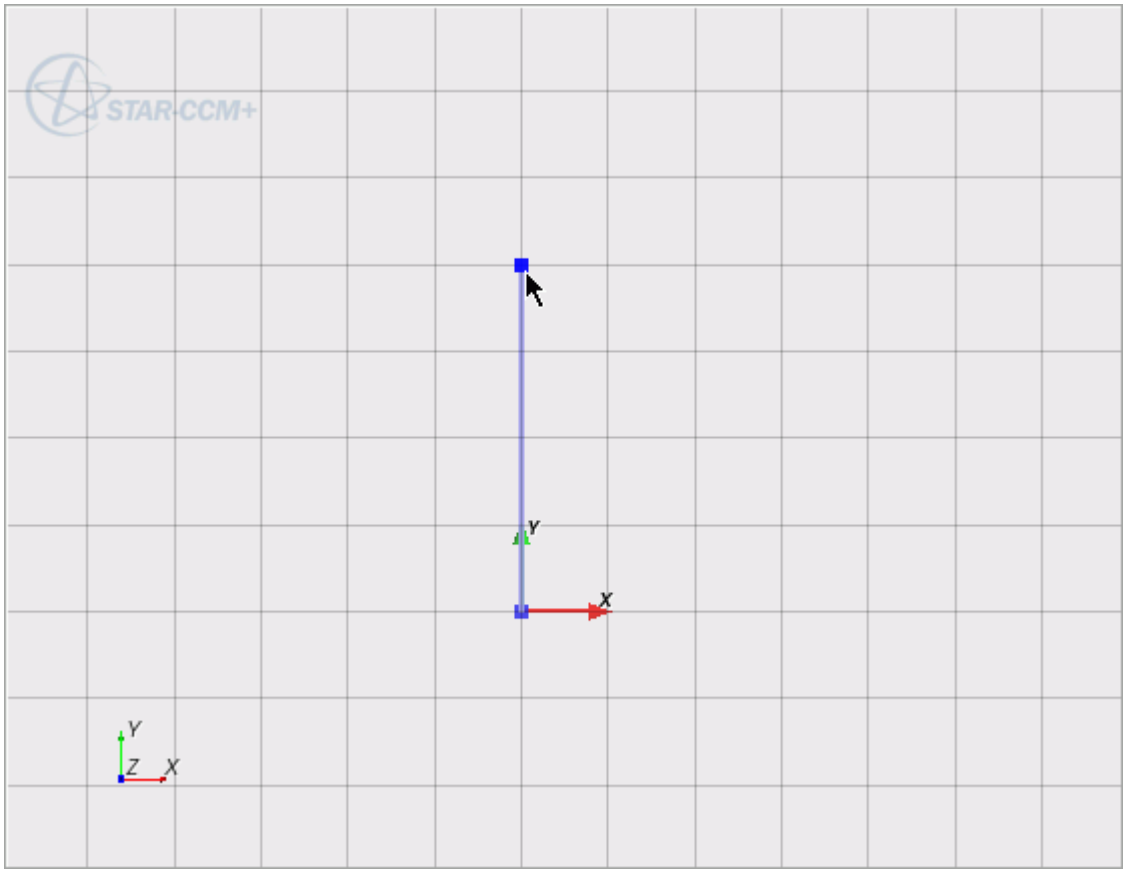
Creating the Geometry
Specifying Boundary Faces on the Backward Facing Step Geometry
Creating the Shape for Volumetric Mesh Refinement

Creating the Geometry

Create the geometry of the backward facing step with the sketch plane and **Create Line** tool.

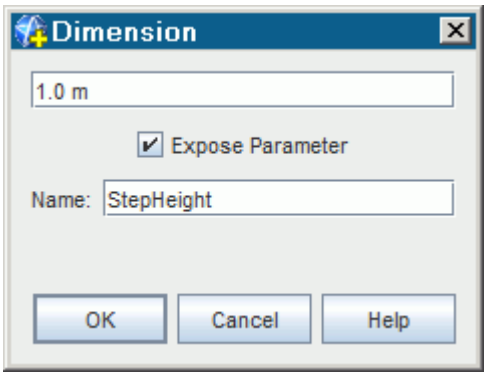
Use the following steps:

1. Create a sketch on the XY plane by right-clicking the **Features** > **XY** node and selecting **Create Sketch**.
2. Click (**Set Sketch Grid Spacing**) and change the *Grid Spacing* to `0.25 m`.
3. Click (**View Normal to Sketch Plane**) to bring the sketch plane into view.
4. Use (**Create Line**) to draw a line of length `1 m` starting at the origin (position `[0, 0]`) and extending in the positive y direction. After placing the end point, click `<ESC>`.



5. Right-click the vertex that lies on the origin and select **Apply Fixation Constraint**. This prevents the selected vertex on the line from moving.
6. Right-click the line and select **Apply Length Dimension**.
7. In the *Dimension* dialog:
   a) Enter `1 m` for the dimension.
   b) Activate the **Expose Parameter** option.

      This exposes the dimension as a design parameter which you can modify outside of 3D-CAD at a later stage.
   c) Enter `StepHeight` as the name of the parameter.



d) Click **OK**.

Create the other lines in relation to the step height:

8. Draw a line starting from the top of the step height and extending approximately `3.8 m` in the negative x direction.

   You do not have to enter exact dimensions as you draw the sketch--these can be added later. You are advised, however, to add vertical and horizontal constraints as you create each line. This practice makes sure that the overall shape of the sketch does not change.

9. Right-click the horizontal line and select **Apply Length Dimension**.
10. In the *Dimension* dialog, enter `3.8 * $StepHeight` and click **OK**.

    The length of the line is associated with the height of the step. The **StepHeight** design parameter is referenced with the `$` sign. As the step height is set to 1 m, the length of the line is set to 3.8 m.
11. Using the same techniques, continue by creating the other lines as shown below.

    Create each line with relation to the step height, H.

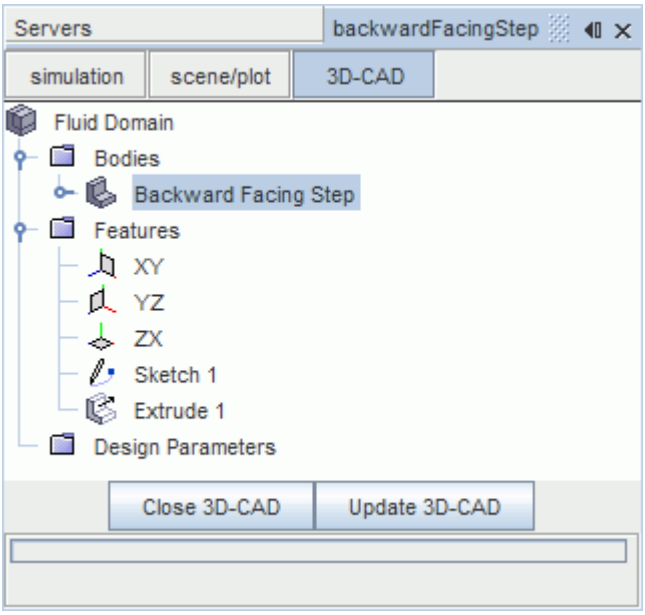
12. Click **OK** to exit the sketch.

Extrude the sketch profile to obtain a solid body:

13. Right-click **Sketch 1** and select **Extrude**.

    The *Extrude* dialog appears.
14. Set *Distance* to `0.5 m`
15. Click **OK** to perform the extrusion.

    A new body, **Body 1**, is added below the **Bodies** node in the object tree.
16. Select the **Bodies** > **Body 1** node and rename it to `Backward Facing Step`.

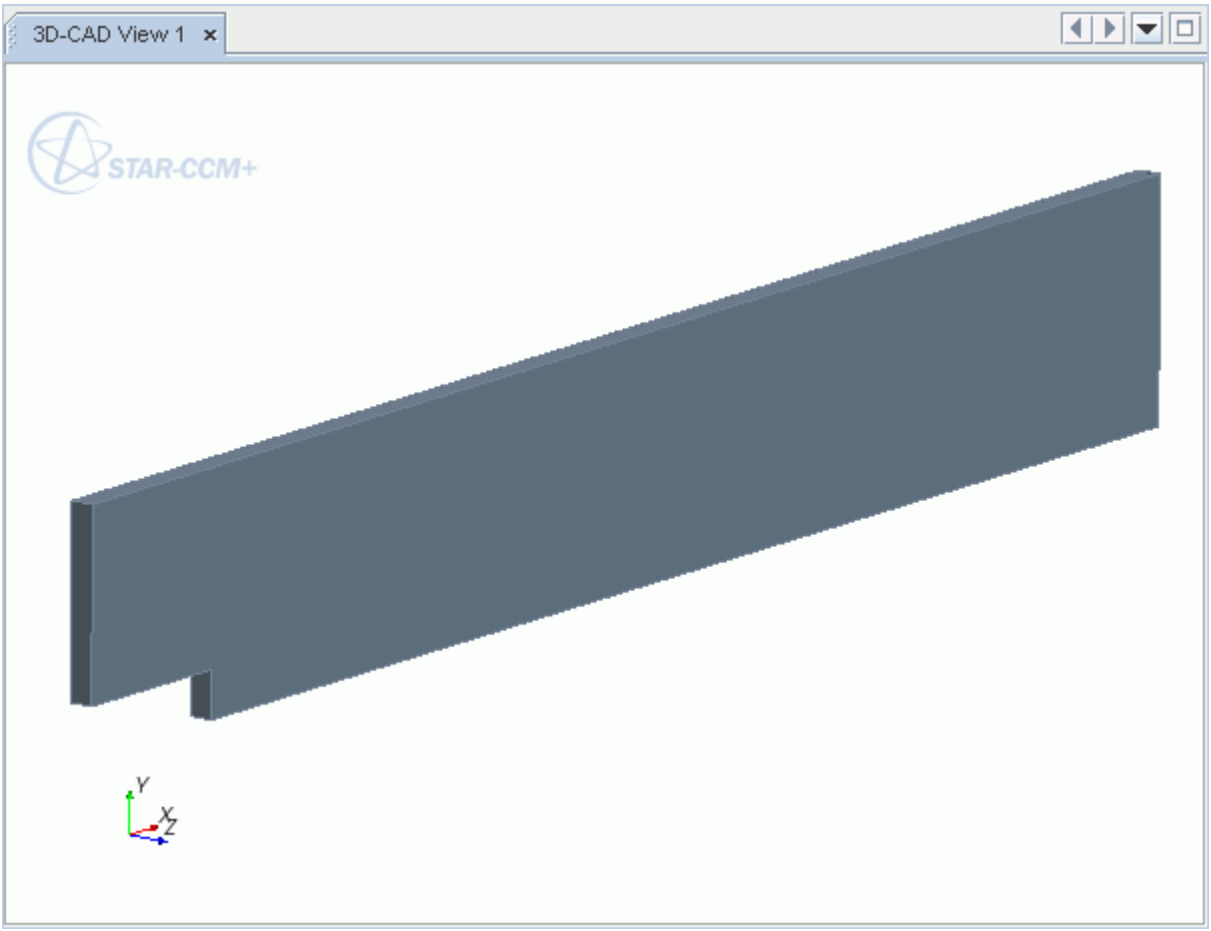


Specifying Boundary Faces on the Backward Facing Step Geometry

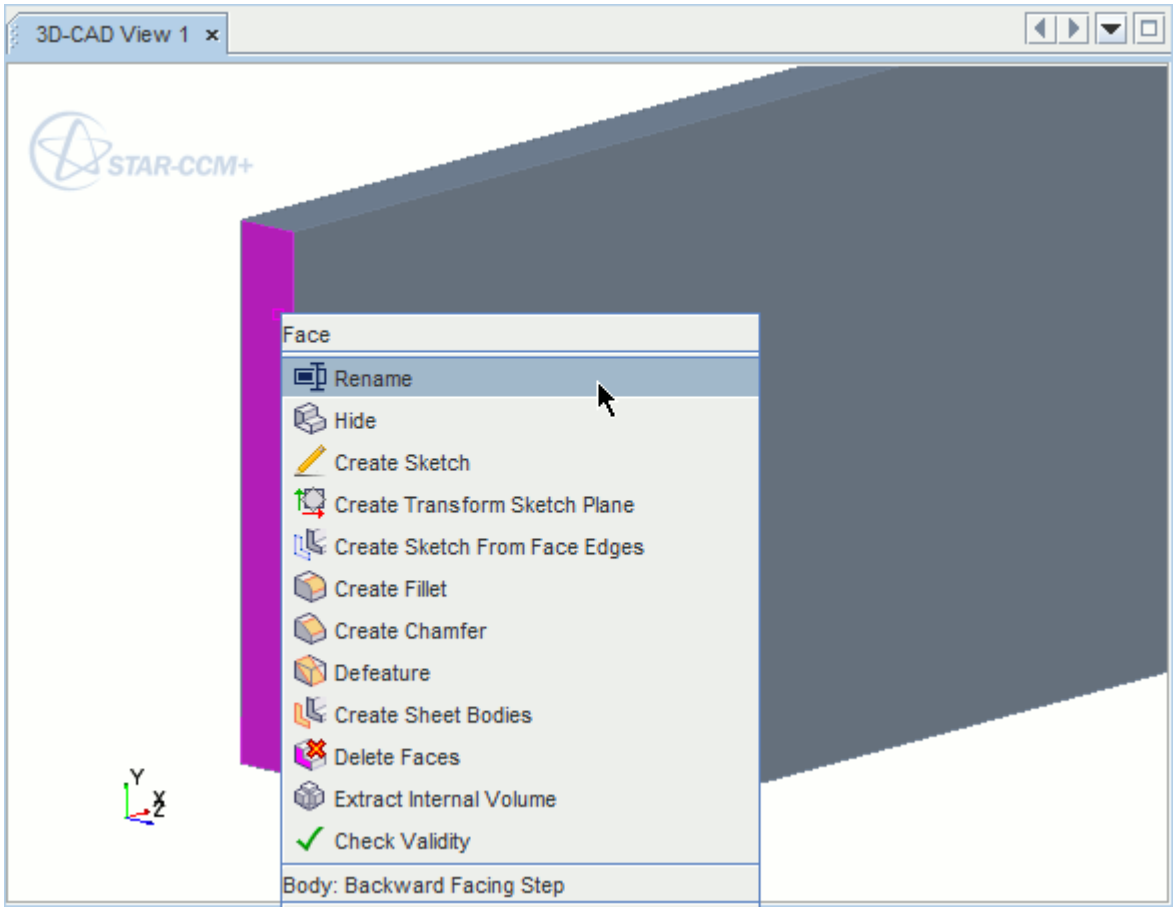
The next step in preparing the model geometry is to identify the boundary faces of the model by setting face names. The benefit of renaming the face in the 3D-CAD model is that it retains its unique identity when the 3D-CAD model is converted to a geometry part in the next stage of the tutorial.

Use the following steps:

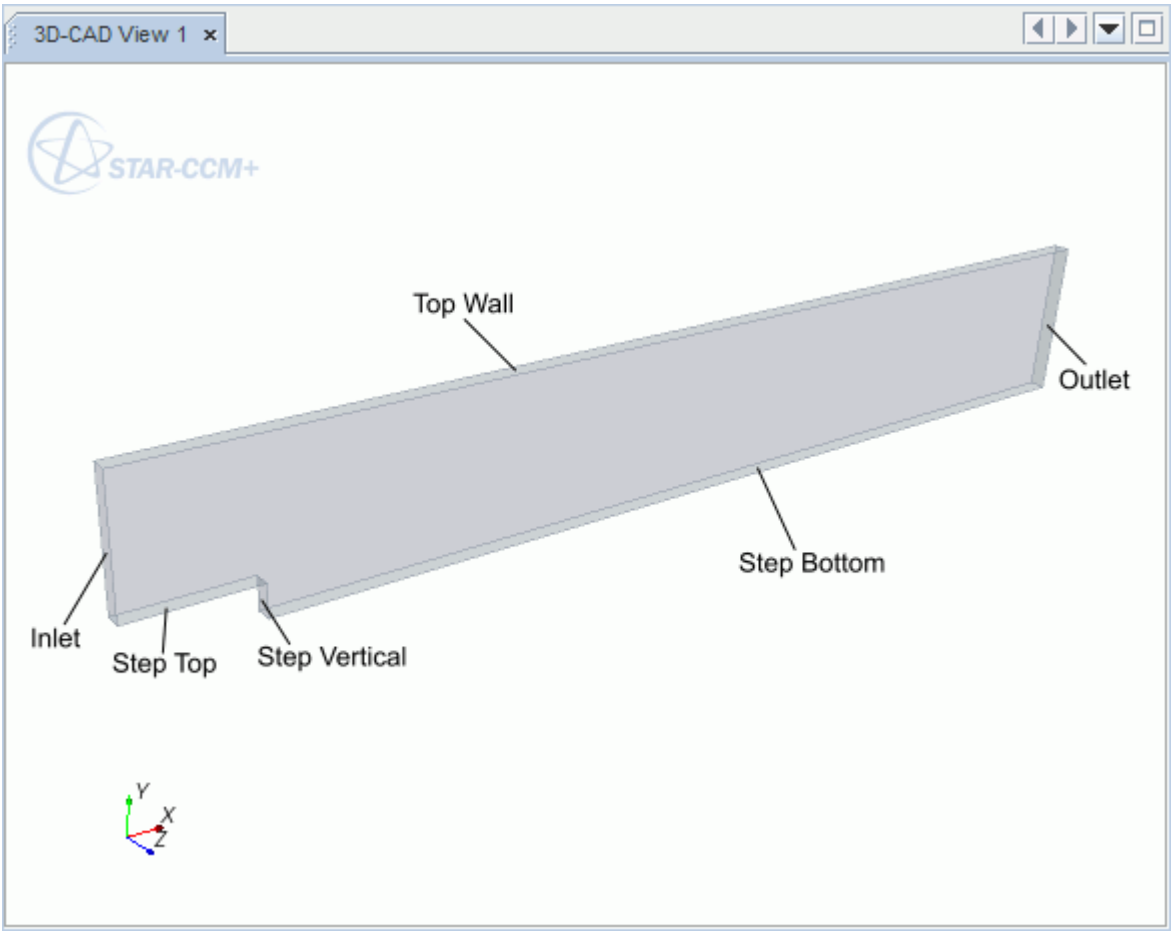
1. Position the geometry as shown below.



2. Right-click the face shown below and select **Rename**.

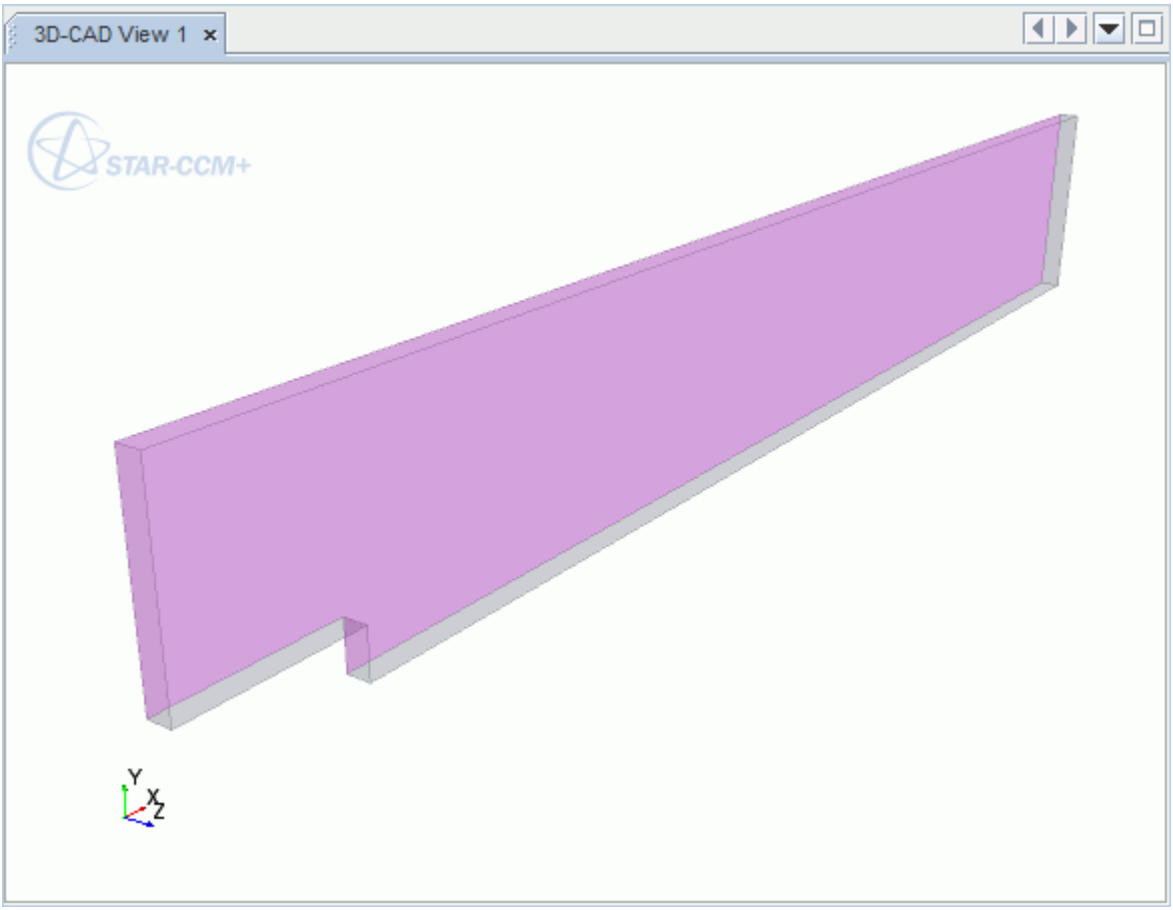


3. Enter `Inlet` in the *Rename* dialog and click **OK**.
4. Rename the other faces on the geometry to match those shown below.

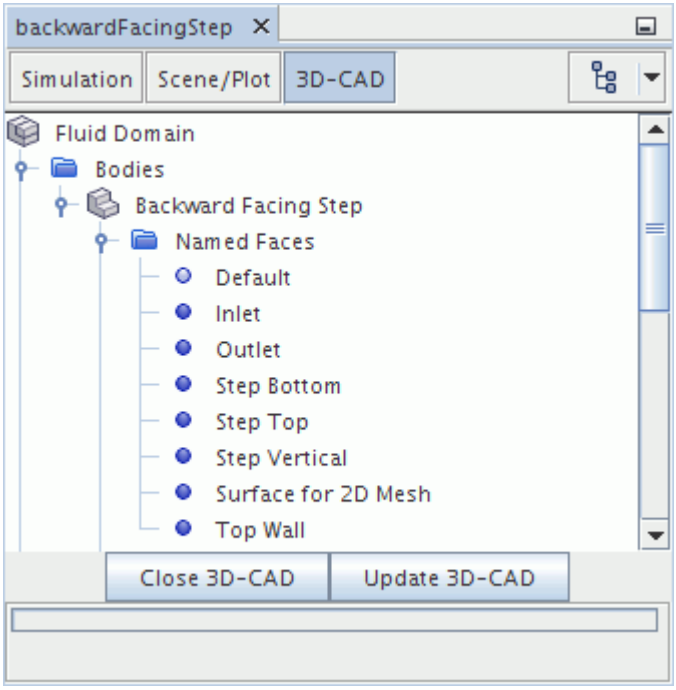


Rename the surface that lies on the Z = 0 plane. This surface is used at a later stage to generate a 2D mesh.

5. Right-click the following surface and rename it to `Surface for 2D Mesh`.



The renamed faces are listed in the **Bodies** > **Backward Facing Step** > **Named Faces** node, as shown below.



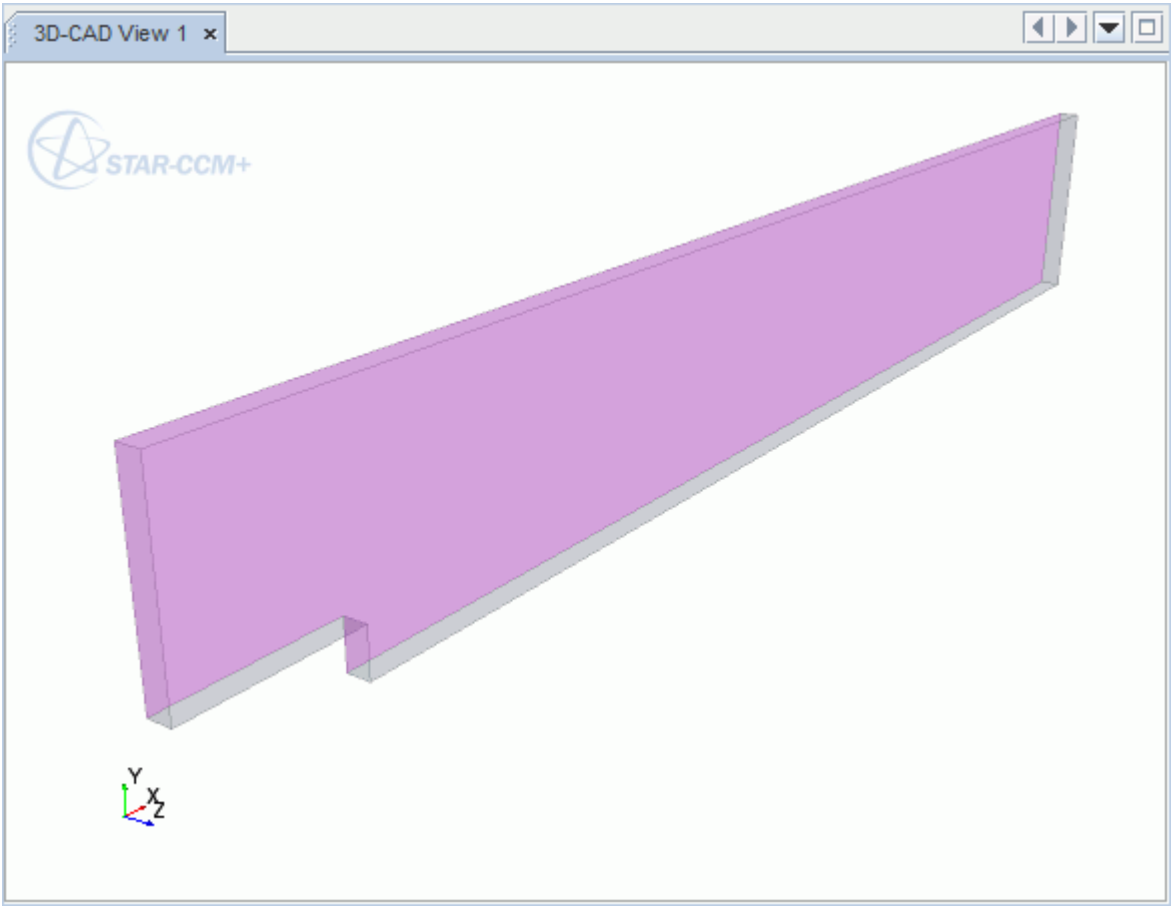
Clicking a node in the object tree highlights the relevant face in the *3D-CAD view* scene.

The backward facing step geometry is now complete.

Creating the Shape for Volumetric Mesh Refinement

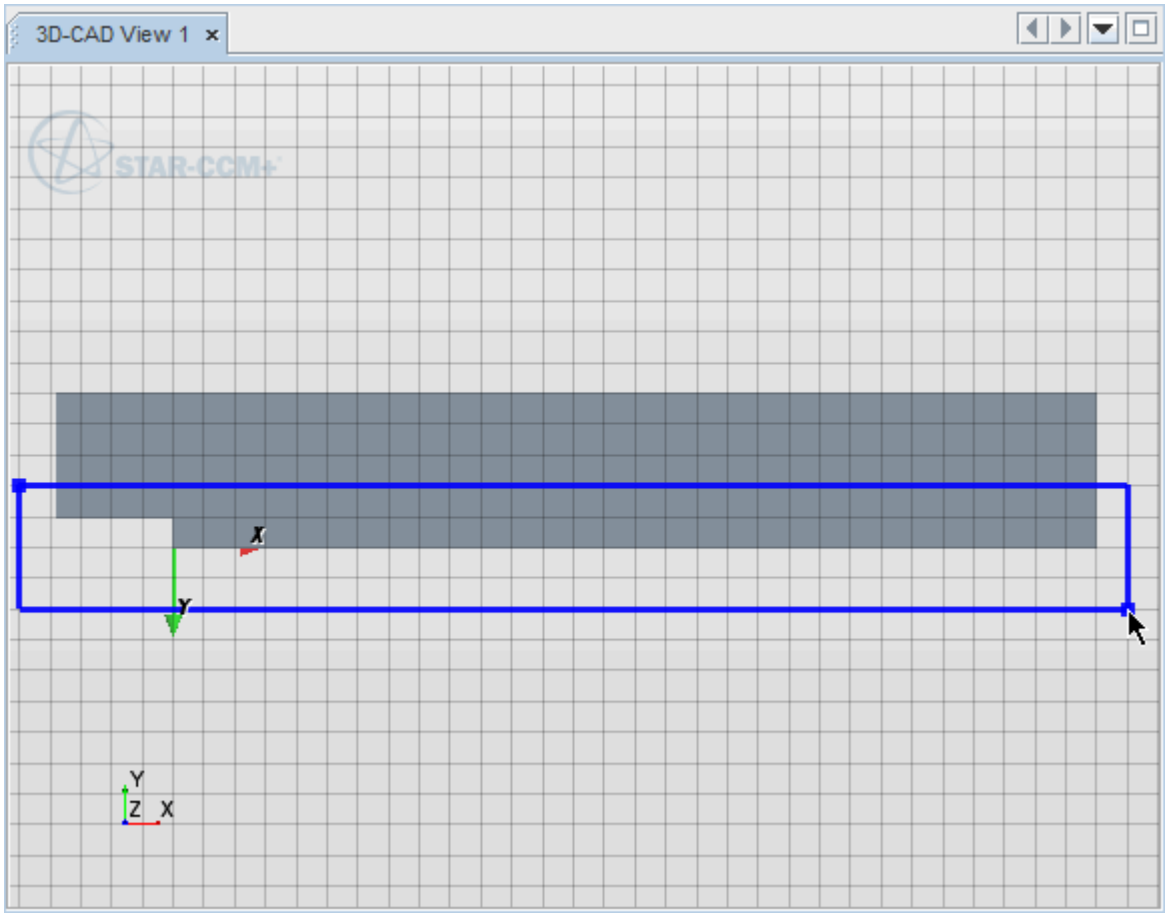
Create the shape to use as volumetric refinement area.

1. Right-click the face shown below and select **Create Sketch** > **On Face**.



2. In *Sketch Mode*, click the <T> key to change to the side view.

3. Click (**Create Rectangle**) and draw a rough box around the bottom half of the geometry so that it covers the step.

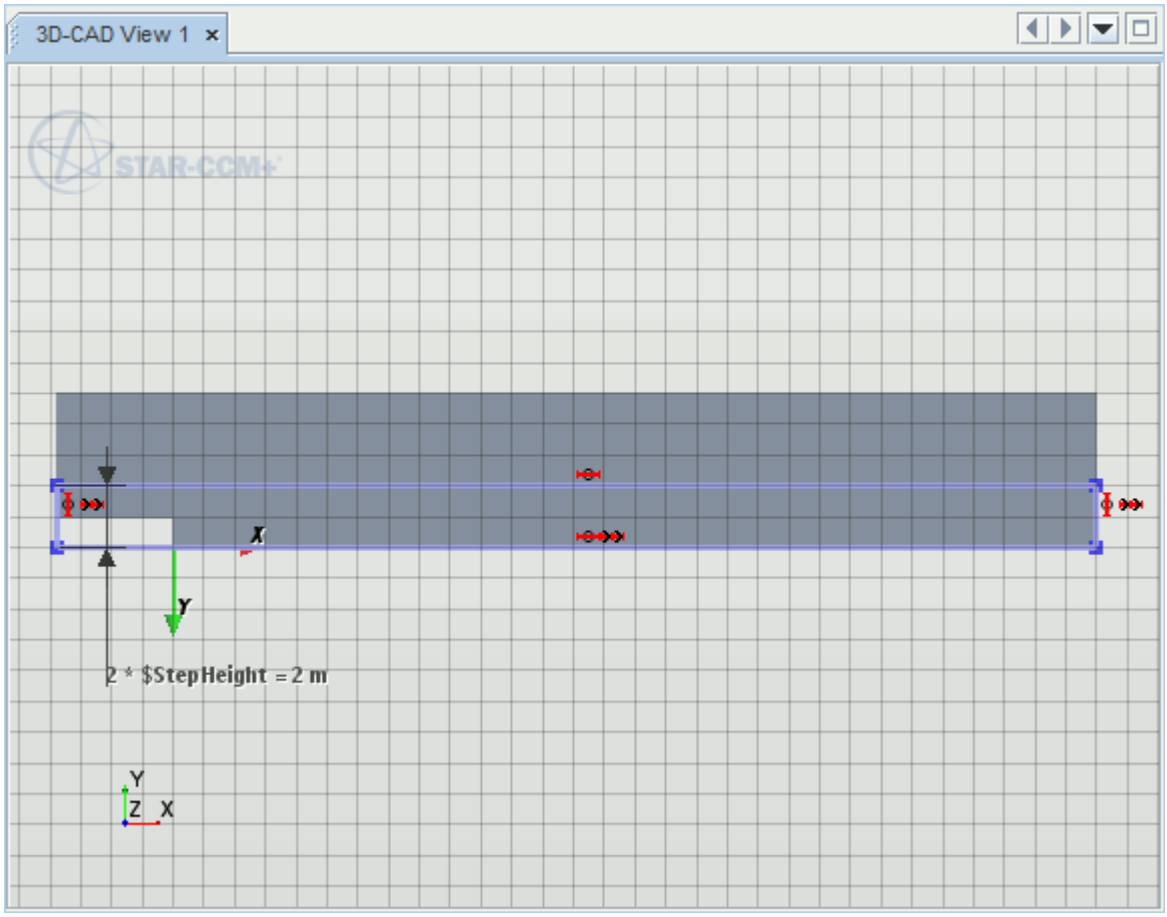


4. Constrain the sketch so that the left, right, and bottom edges of the sketch are collinear with the corresponding edges on the backward facing step geometry.

   Constrain each edge of the rectangle separately. For example, select the bottom edge of the rectangle and the bottom edge of the solid part, then right-click on a selection and select **Apply Collinear Constraint**.

Add a dimension for the height of the box, so that it is twice height of the step height.

5. Right-click the left edge of the sketch and select **Apply Length Dimension**.
6. In the *Dimension* dialog, enter `2 * $StepHeight` and click **OK**.



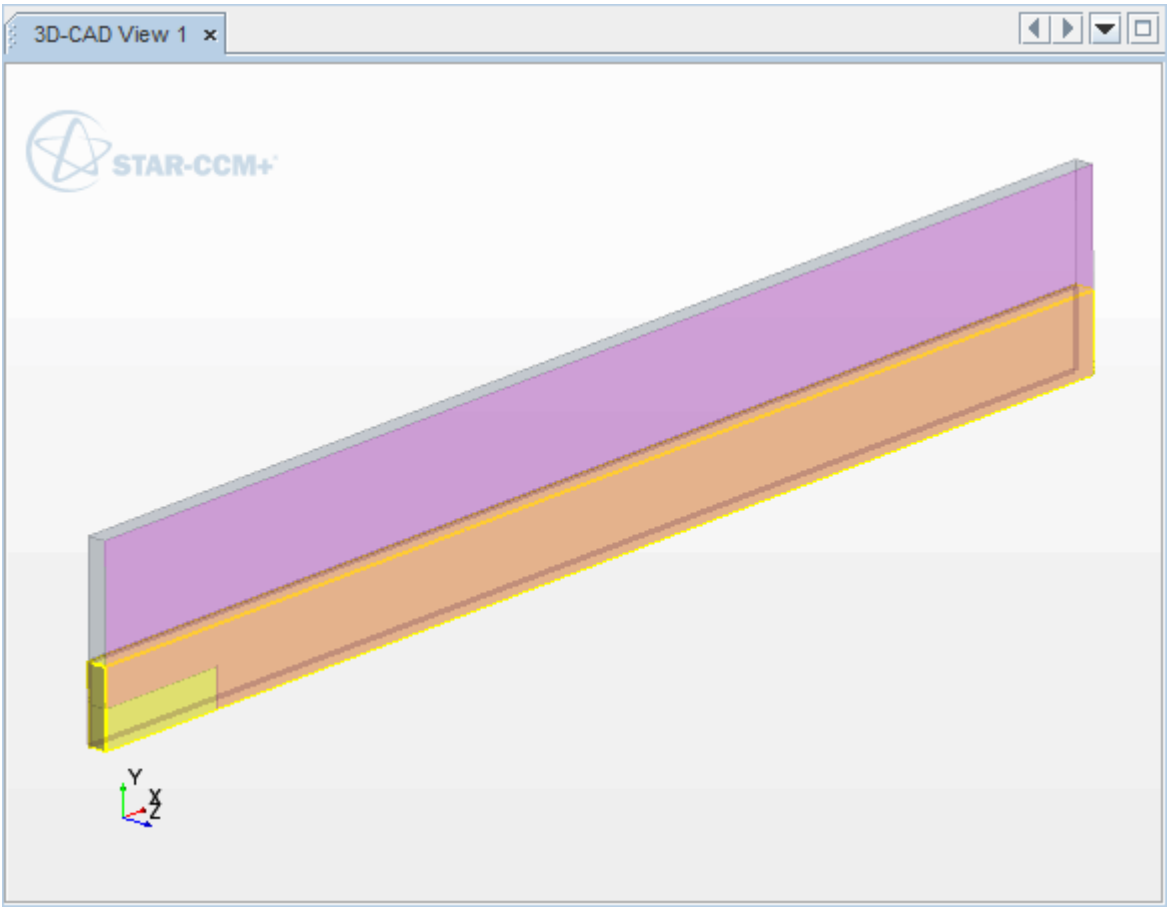
7. Click **OK** to exit *Sketch Mode*

Create a solid body from the sketch:

8. Right-click **Sketch 2** and select **Extrude**.
9. In the *Extrude* panel, set the following properties:

Method	**Up to Face**
Direction	**Reverse**
Body Interaction	**None**

10. Select the surface opposite the sketch plane.



A preview of the box appears.

11. Click **OK**.

    A new body is created. This body is associated with the shape of the backward facing step geometry. If the backward facing step domain changes, the box changes with it.

12. Rename the **Bodies** > **Body 2** node to `Refinement Box`.
13. Click **Close 3D-CAD**.
14. Save the simulation.

Setting Up the Backward Facing Step Simulation

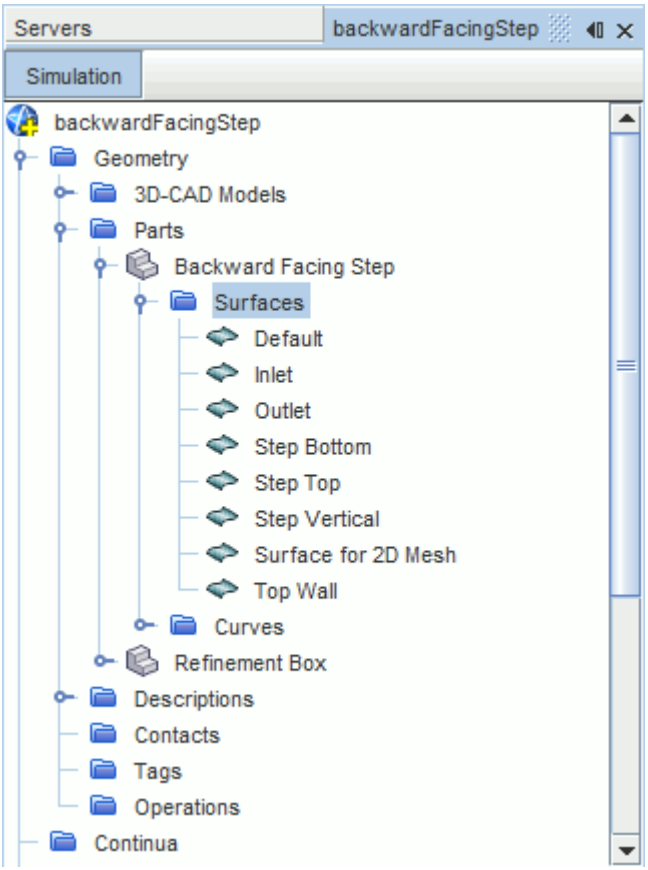
Create a new geometry part using the 3D-CAD model.

Use the following steps:

1. Right-click the **Geometry** > **3D-CAD Models** > **Fluid Domain** node and select **New Geometry Part**.

   The *Parts Creation Options* dialog appears.

2. The default settings are acceptable, so you can click **OK** to close the dialog.
3. Expand the **Parts** > **Backward Facing Step** > **Surfaces** node and note that the boundary faces specified in 3D-CAD are defined as separate surfaces.



Contents:

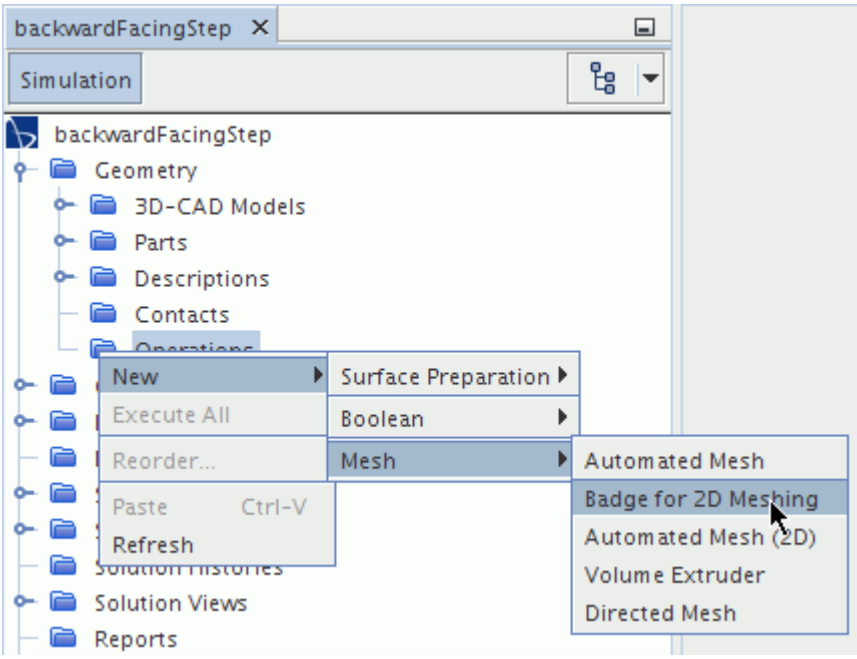
Badging Parts for 2D Meshing
Assigning a Part to a New Region
Setting the Boundary Types
Setting Up the Meshers
Defining the Volumetric Control
Generating the 2D Mesh
Selecting Physics Models
Modifying Material Properties and Setting Initial Conditions
Setting Up Boundary Conditions
Preparing Scenes for Visualization
Preparing the Plot
Setting Up Stopping Criteria
Modifying Solver Settings
Running the Simulation
Visualizing the Results
Validating the Results
Modifying the Step Height

Badging Parts for 2D Meshing

Before you can generate a 2D mesh as part of an automated mesh pipeline, you must badge the part. Badging a part notifies STAR-CCM+ that this part will be used to generate a 2D mesh. The requirement for this feature is that one separate surface must lie on the Z = 0 plane. In a previous section, you renamed the surface that lies on the Z = 0 plane to `Surface for 2D Mesh`, so in this case the requirement is met.

To badge a part for 2D Meshing:

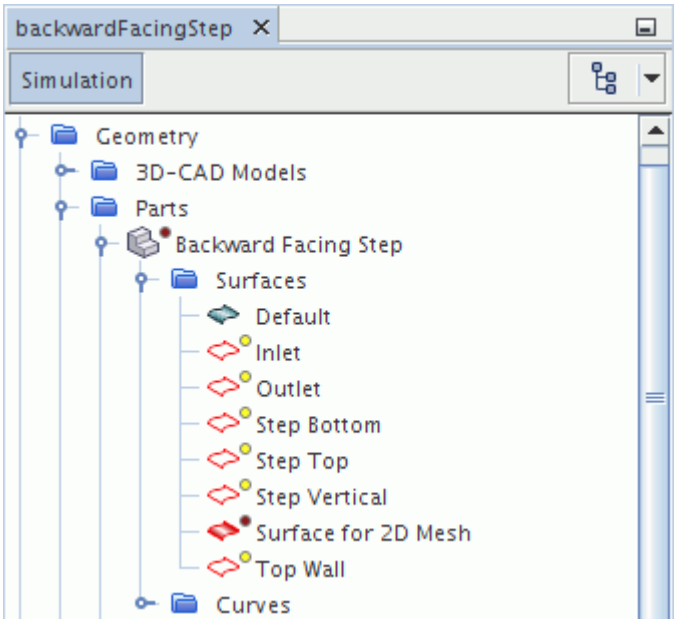
1. Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Badge for 2D Meshing**.



2. In the *Create Badge for 2D Meshing Operation* dialog, select **Backward Facing Step** from the *Parts* list, and click **OK**.

   A new operation, **Badge for 2D Meshing** is added below the **Operations** node.
3. Right-click the **Badge for 2D Meshing** node and select **Execute**.
4. Expand the **Geometry** > **Parts** > **Backward Facing Step** > **Surfaces** node.

A badge is added next to the surfaces that are identified for 2D meshing.

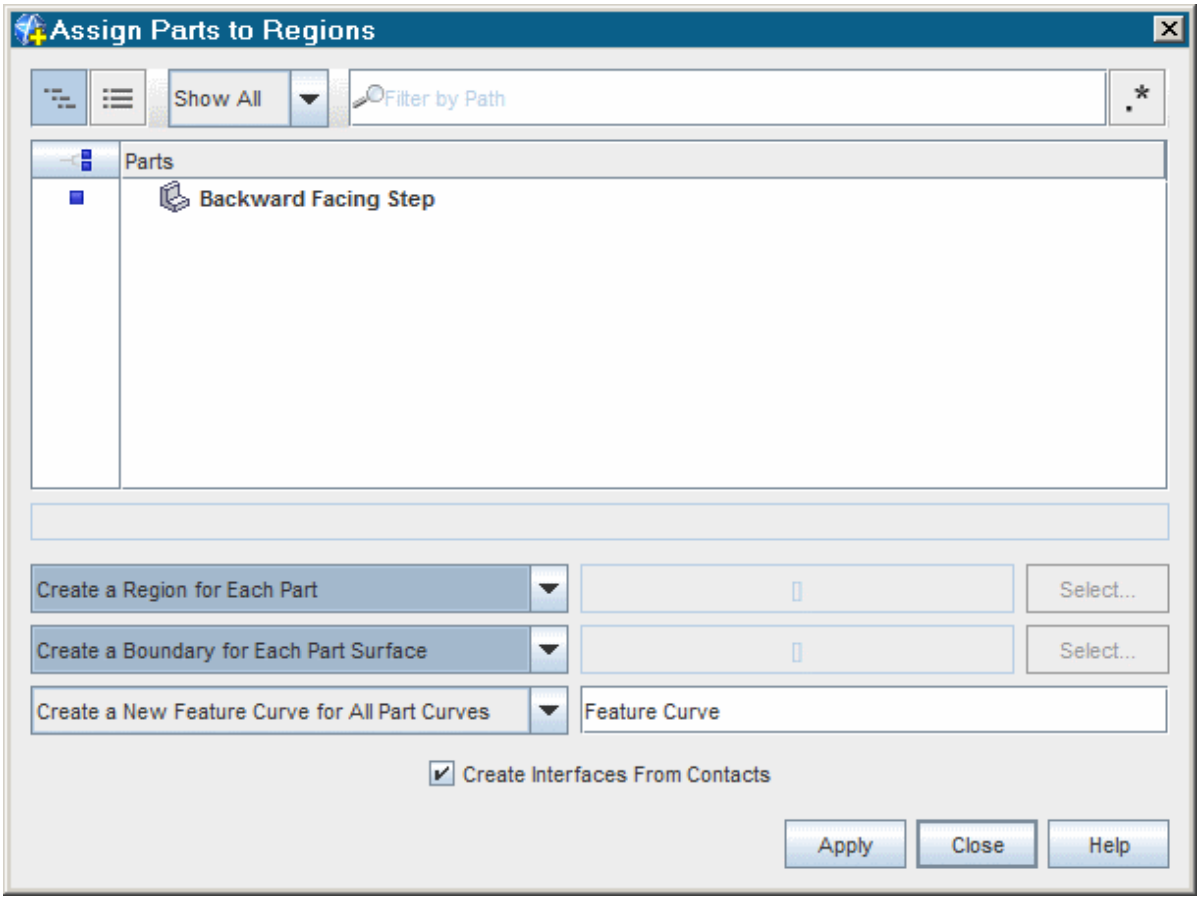


When you generate a mesh with this part, you will get a 2D mesh. The surface with the dark red tag (on the node icon) becomes the 2D region, while the surfaces with the yellow tags become 2D boundaries.

Assigning a Part to a New Region

Assign the geometry part to a region and create boundaries.

1. Right-click the **Parts** > **Backward Facing Step** node and select **Assign Parts to Regions**.
2. In the *Assign Parts to Regions* dialog, select:
   a) **Create a Region for Each Part**.
   b) **Create a Boundary for Each Part Surface**.



3. Click **Apply** then **Close**.

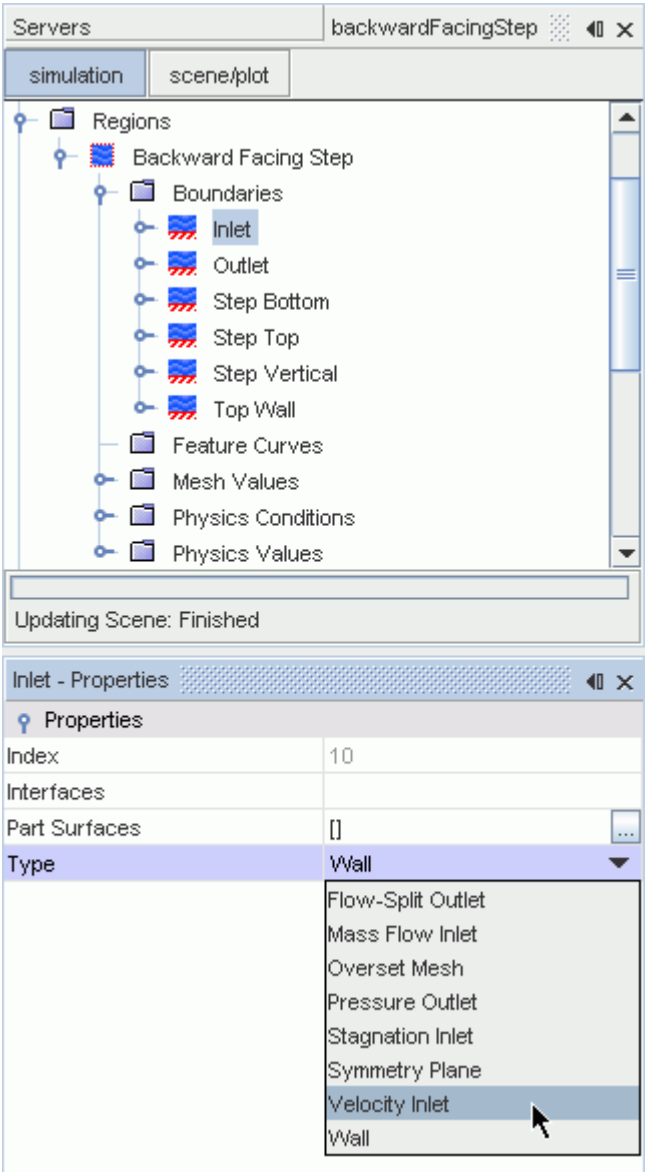
A new region is created. Each labeled surface is now a separate boundary in the region.

4. Create a geometry scene.

Setting the Boundary Types

Set the boundary types for the inlet and outlet boundaries.

1. Select the **Regions** > **Backward Facing Step** > **Boundaries** > **Inlet** node and set the *Type* to **Velocity Inlet**.



2. Select the **Outlet** node and set the *Type* to **Pressure Outlet**.

Setting Up the Meshers

Generate a volume mesh using trimmed cells.

To make sure that the y+ value calculated by the turbulence model is around 50, do not refine the mesh near the walls.

1. Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
2. In the *Create Automated Mesh Operation* dialog:
   a) Select **Backward Facing Step** from the *Parts* list.
   b) Select the following meshers, in order:

Group	Mesher
Surface Meshers	**Surface Remesher**
Core Volume Meshers	**Trimmed Cell Mesher**

3. Click **OK**.

Modify the mesher parameters:

4. Select the **Automated Mesh** > **Meshers** > **Trimmed Cell Mesher** node and activate the *Perform Mesh Alignment* option.

   This option allows you to align the mesh at a specific point, for example at the top corner of the step.
5. Edit the **Automated Mesh** > **Default Controls** node and set the following properties:

Node	Property	Value
Base Size	*Value*	`0.2 m`
Minimum Surface Size	*Percentage of Base*	`100`
Mesh Alignment Location	*Location*	`[0.0, 1.0, 0.0] m`

6. Save the simulation.

Defining the Volumetric Control

Create a volumetric control to refine the mesh near the bottom wall.

To create the volumetric control:

1. Right-click **Operations** > **Automated Mesh** > **Custom Controls** and select **New** > **Volumetric Control**.

   A new node, **Volumetric Control**, is added below the **Custom Controls** node.
2. Select the **Volumetric Control** node and edit the *Parts* property.
3. Select **Refinement Box**.
4. Click **OK**.
5. Select the **Volumetric Control** > **Controls** > **Trimmer** node and activate the *Customize isotropic size* option.

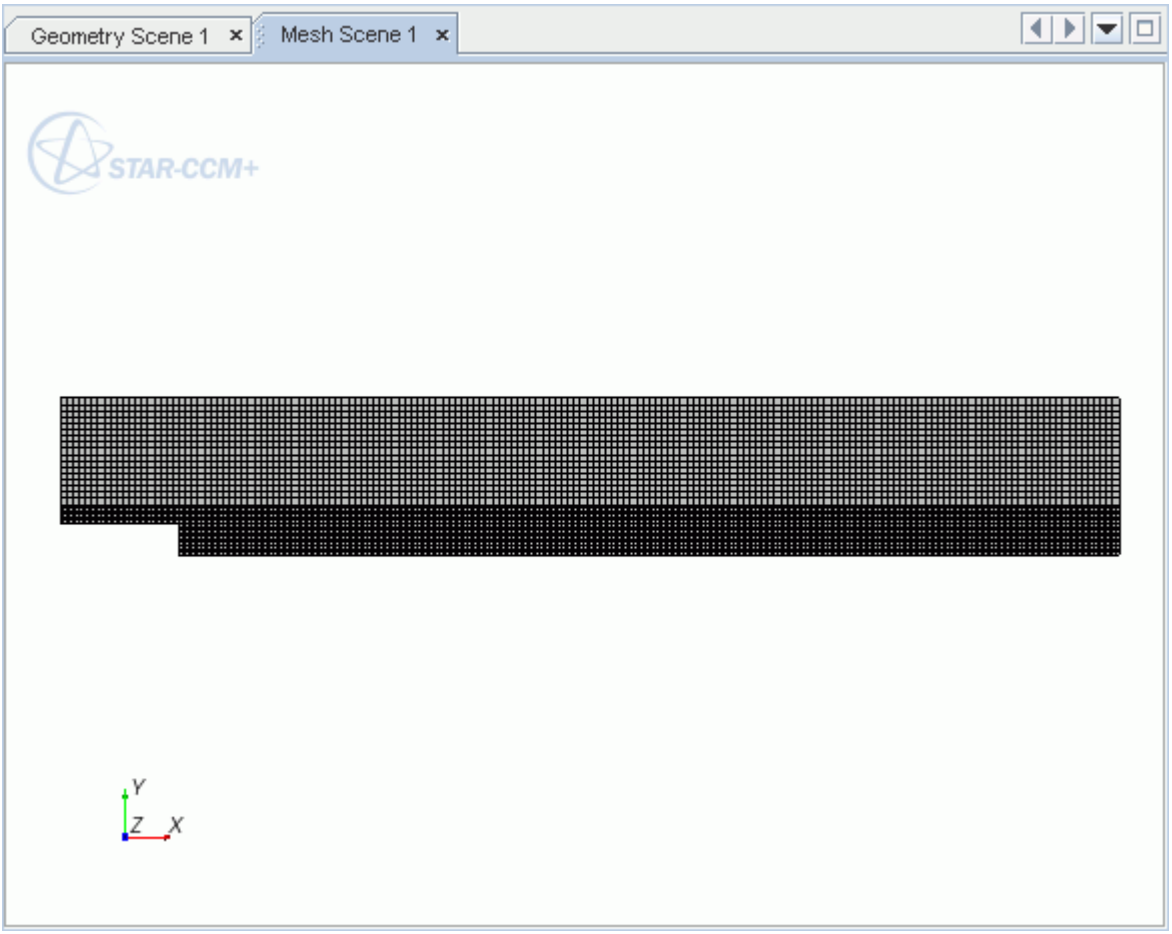
   A new node, **Custom Size**, is added to the **Volumetric Control** > **Values** node.
6. Select the **Values** > **Custom Size** node and set *Percentage to Base* to `50`.

Generating the 2D Mesh

The 2D mesh is now ready to be generated.

To generate the 2D mesh:

1. Click (**Generate Volume Mesh**).
2. Create a mesh scene and examine the mesh. The 2D mesh is shown below.



3. Save the simulation.

Selecting Physics Models

A physics continuum was automatically created during the mesh generation process.

Select the physics models for the first part of the tutorial.

1. Expand the **Continua** node and rename **Physics 1** to `Fluid`.
2. Select the following models for **Fluid**, in order:

Group Box	Model
Space	**Two-Dimensional** (Pre-selected)
Time	**Steady**
Material	**Gas**
Flow	**Coupled Flow**
Gradient Metrics	**Gradients** (Selected automatically)
Equation of State	**Constant Density**
Viscous Regime	**Turbulent**

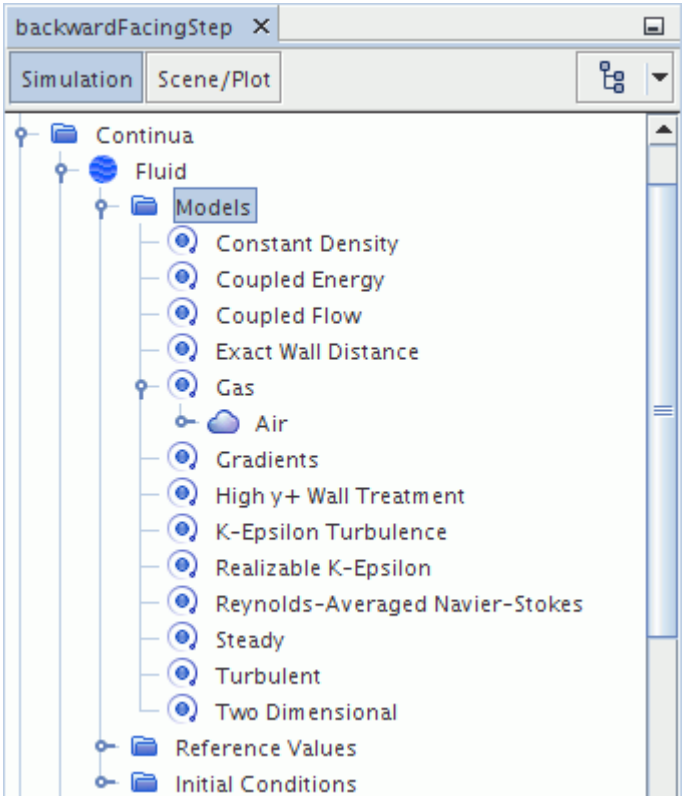
Group Box	Model
Turbulence	**Reynolds-Averaged Navier-Stokes** (Selected automatically)

3. Deactivate **Auto-select recommended models**. The appropriate models are selected manually.
4. Select the following models, in order:

Group Box	Model
Reynolds Averaged Turbulence	**K-Epsilon Turbulence**
K-Epsilon Turbulence Models	**Realizable K-Epsilon**
Wall Distance	**Exact Wall Distance** (Selected automatically)
K-Epsilon High y+ Wall Treatment	**High y+ Wall Treatment**
Optional Models	**Coupled Energy**

5. Click **Close**.

The selected models now appear.



Modifying Material Properties and Setting Initial Conditions

Modify the material properties of air.

1. Edit the **Fluid** > **Models** > **Gas** > **Air** > **Material Properties** node and set the following values:

Node	Property	Value
Density > **Constant**	*Value*	`2 kg/m^3`
Dynamic Viscosity > **Constant**	*Value*	`7.142E-5 Pa-s`
Specific Heat > **Constant**	*Value*	`1000.0 J/kg-K`
Thermal Conductivity > **Constant**	*Value*	`0.1 W/m-K`

Define the initial conditions.

2. Edit the **Fluid** > **Initial Conditions** node and select the following values:

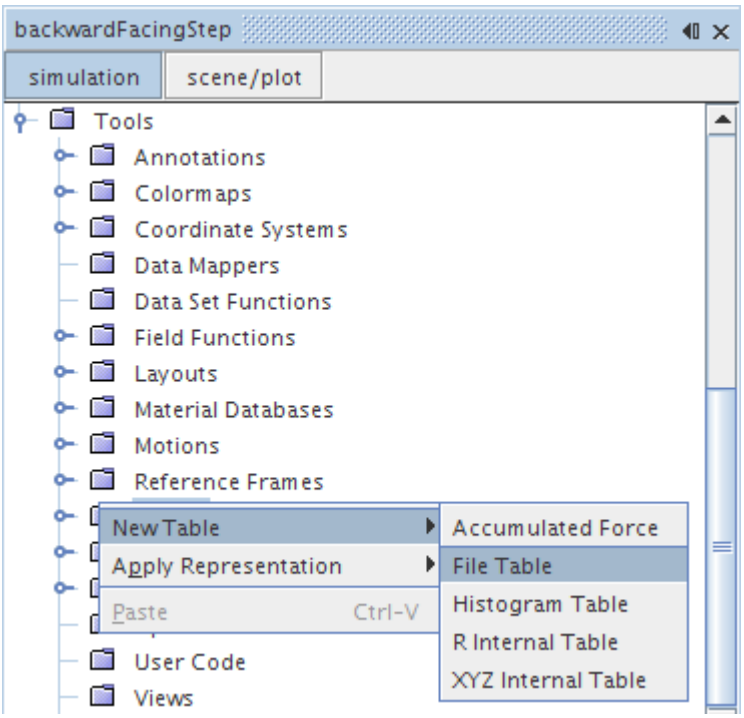
Node	Property	Value
Turbulence Specification	*Method*	**K + Epsilon**
Turbulent Dissipation Rate	*Value*	`3.0E-6 m^2/s^3`
Turbulent Kinetic Energy	*Value*	`1.5E-4 J/kg`
Velocity	*Value*	`[1.0, 0.0] m/s`

Setting Up Boundary Conditions

Define the inlet boundary using inlet data from a table.

Use the following steps:

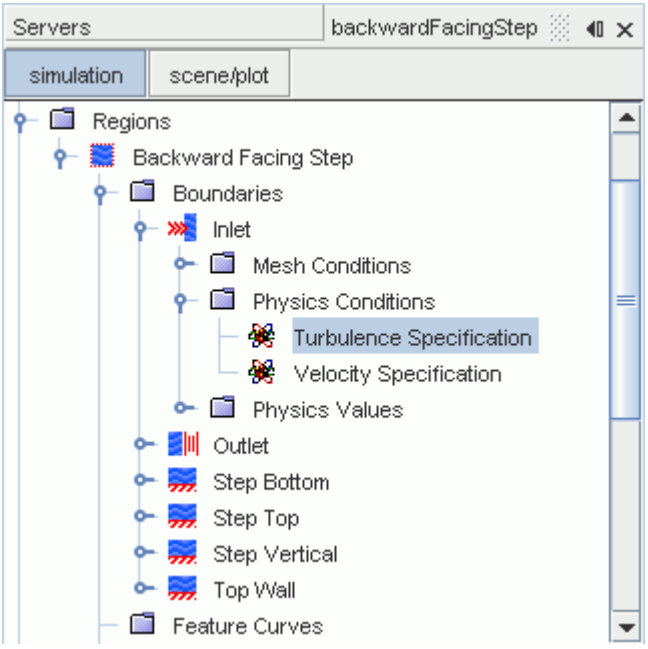
1. Right-click **Tools** > **Tables** and select **New Table** > **File Table**.



2. In the *Open* dialog, navigate to the `doc/startutorialsdata/incompressibleFlow/data` subdirectory of your STAR-CCM+ installation directory and select file `backStepInletData.csv`.

   The file is added to the **Tables** node.

3. Select the **Regions** > **Backward Facing Step** > **Boundaries** > **Inlet** > **Physics Conditions** > **Turbulence Specification** node and set the *Method* to **K + Epsilon**.



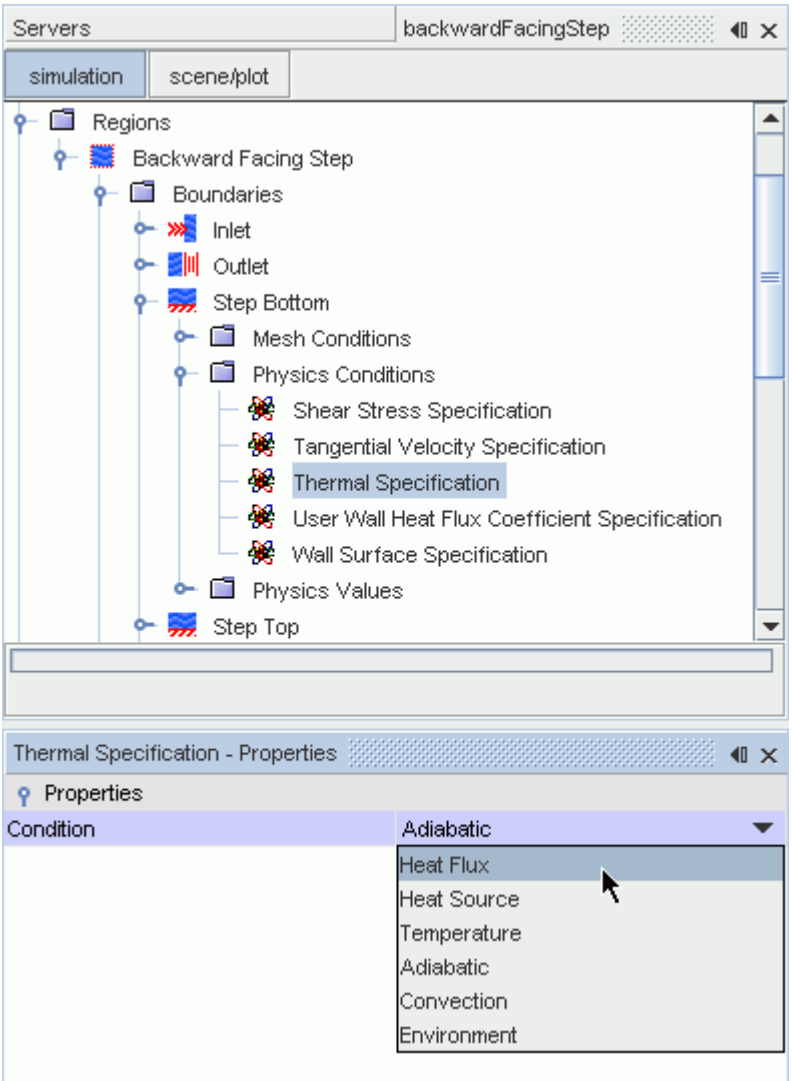
4. In the same **Physics Conditions** node select the **Velocity Specification** node and set the *Method* to **Components**.

5. Edit the **Physics Values** node and set the following properties:

Node	Property	Value
Turbulent Dissipation Rate	*Method*	**Table (x,y,z)**
└ **Table (x,y,z)**	*Table*	**backStepInletData**
	Table: Data	**Turbulent Dissipation Rate**
Turbulent Kinetic Energy	*Method*	**Table (x,y,z)**
└ **Table (x,y,z)**	*Table*	**backStepInletData**
	Table: Data	**Turbulent Kinetic Energy**
Velocity	*Method*	**Composite**
└ **Composite** > **X Component**	*Method*	**Table (x,y,z)**
└ **Table (x,y,z)**	*Table*	**backStepInletData**
	Table: Data	**Velocity[i]**

Set the heat flux on the bottom wall:

6. Select the **Boundaries** > **Step Bottom** > **Physics Conditions** > **Thermal Specification** node and set the *Condition* to **Heat Flux**.



7. Select the **Physics Values** > **Heat Flux** node and set the *Value* to `500 W/m^2`
8. Save the simulation.

Preparing Scenes for Visualization

A scalar scene is created to show the velocity magnitude on the 2D mesh. Use this to visualize the solution while the simulation is running.

Use the following steps:

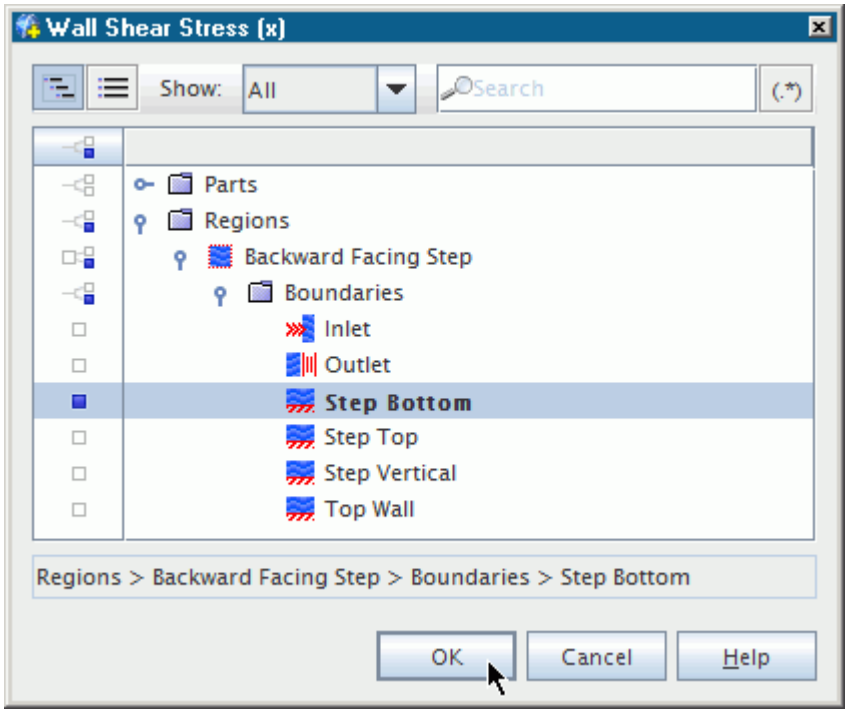
1. Create a scalar scene.
2. Click the **scene/plot** button above the object tree and select the **Displayers** > **Scalar 1** > **Scalar Field** node and set the *Function* to **Velocity** > **Magnitude**.
3. Click the **Simulation** button to return to the STAR-CCM+ simulation object tree.

Preparing the Plot

Monitor the wall shear stress in the x direction.

Use the following steps:

1. Create an X-Y plot.
2. Rename the new plot to `Wall Shear Stress (x)`.
3. Select the **Plots** > **Wall Shear Stress (x)** node and click to the right of the *Parts* property.
4. In the *Wall Shear Stress* dialog:
   a) Expand the **Regions** > **Backward Facing Step** > **Boundaries** node and select **Step Bottom**.
   b) Click **OK**.



5. Select the **Plots** > **Wall Shear Stress (x)** > **Y Types** > **Y Type 1** > **Scalar Function** node and set *Field Function* to **Wall Shear Stress** > **Laboratory** > **i**.

Copy and paste the Wall Shear Stress (x) plot and modify its parameters so that it displays the Nusselt Number. The Nusselt number is the ratio of the heat transferred between two parallel plates (at different temperatures) by a moving fluid to the heat transfer that would occur by conduction alone.

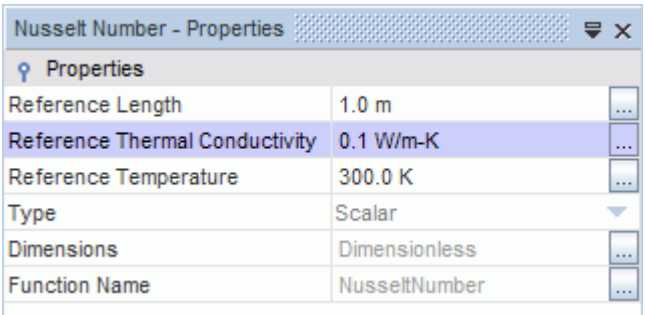
6. Right-click the **Plots** > **Wall Shear Stress (x)** plot and select **Copy**.
7. Right-click the **Plots** node and select **Paste**.

A new plot, **Copy of Wall Shear Stress (x)**, is added to the **Plots** node.

8. Rename the new plot to `Nusselt Number`.
9. Select the **Plots** > **Nusselt Number** > **Y Types** > **Y Type 1** > **Scalar Function** node and set *Field Function* to **Nusselt Number**.

In the **Modifying Material Properties and Setting Initial Conditions** section you modified the default thermal conductivity value. As a result, the **Nusselt Number** field function, which is affected by this change, needs to be updated.

10. Select the **Tools** > **Field Function** > **Nusselt Number** node and set the *Reference Thermal Conductivity* to `0.1 W/m-K`.



Setting Up Stopping Criteria

Limit the number of iterations for the solver to 500. This is sufficient for the solution to converge.

1. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to `500`.
2. Save the simulation.

Modifying Solver Settings

Set the Courant number to 100 and activate the grid sequencing and Expert Driver models.

These tools assist the coupled solver in coming to a converged solution.

- Select the **Solvers** > **Coupled Implicit** node and set the following properties:

Node	Property	Value
Coupled Implicit	*Courant Number*	`100`
└ **Expert Initialization**	*Method*	**Grid Sequencing**
└ **Solution Driver**	*Driver*	**Expert Driver**

Running the Simulation

The simulation is now ready to be run.

1. Click (**Run**).

   While the simulation is running you can click the tabs at the top of the *Graphics* window to view each of the scalar scenes.

During the run, it is also possible to stop the process by clicking (**Stop**) on the toolbar. If you do halt the simulation, it can be continued later by clicking (**Run**). If left alone, the simulation continues until 500 iterations are complete.

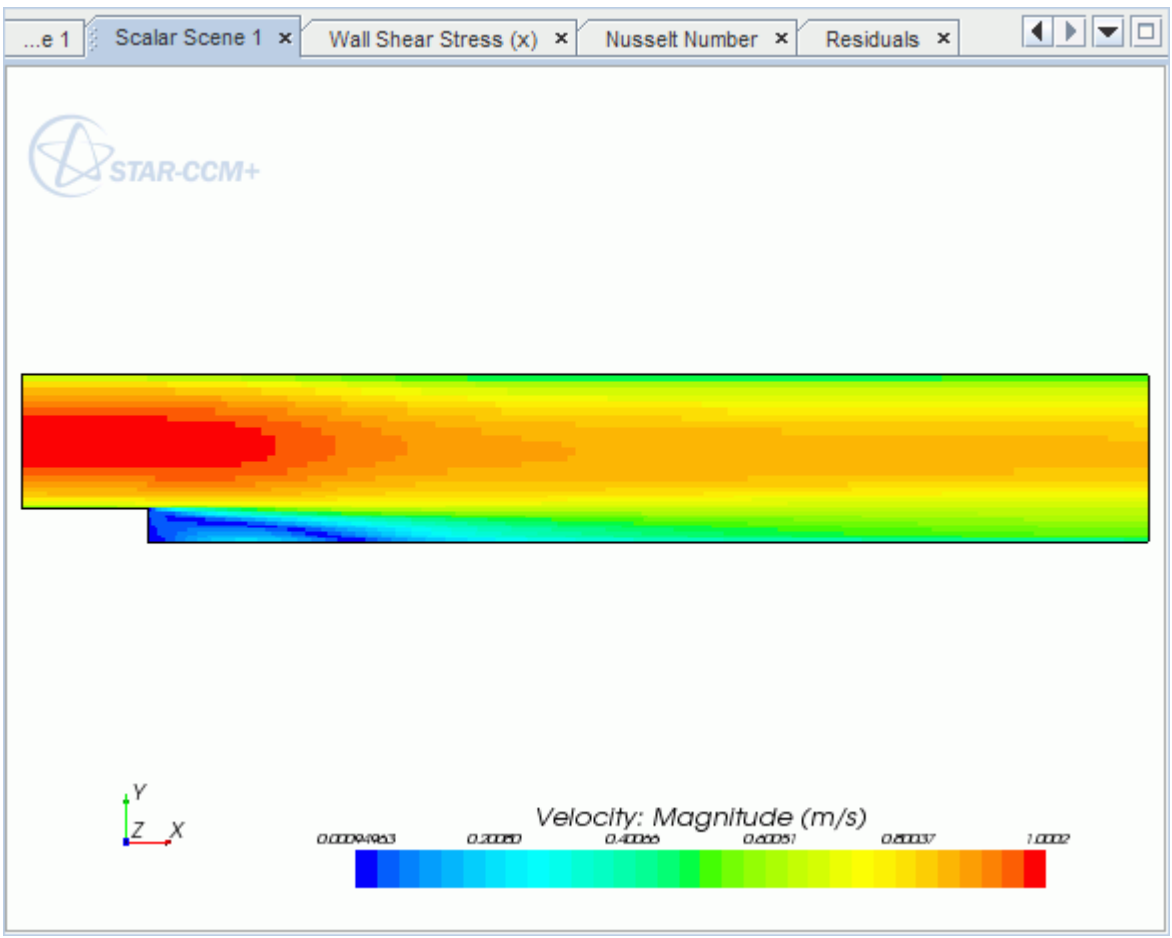
The *Residuals* display is created automatically and shows the progress of the solvers.

2. While the simulation is running, select the *Scalar Scene 1* tab at the top of the *Graphics* window to visualize the solution.
3. When the simulation has finished running, save the simulation.

Visualizing the Results

Examine *Scalar Scene 1*.

The scalar scene is shown below.



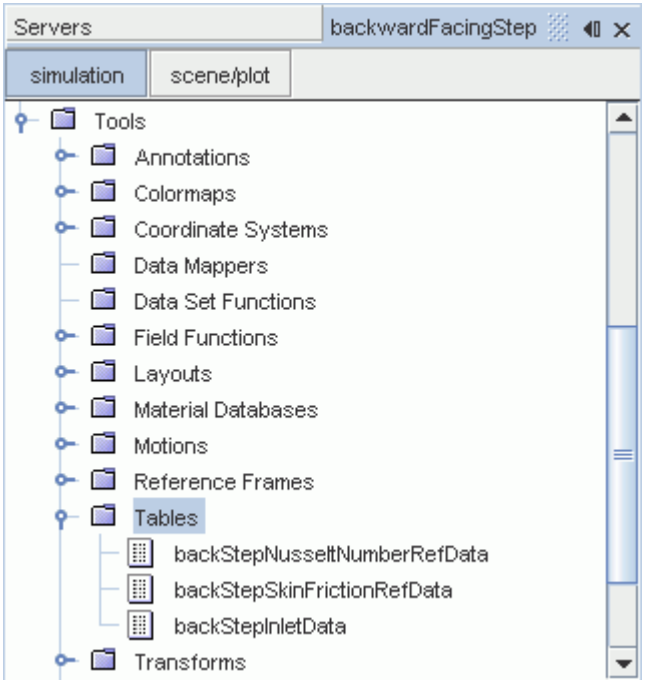
Validating the Results

Import the experimental data into STAR-CCM+ and add it to the Wall Shear Stress (x) and Nusselt Number plots.

In this simulation, the wall shear stress and skin friction coefficient are the same.

1. Right-click the **Tools** > **Tables** node and select **New Table** > **File Table**
2. In the *Open* dialog, navigate to the `doc/startutorialsdata/incompressibleFlow/data` subdirectory of your STAR-CCM+ installation directory and multi-select the files `backStepSkinFrictionRefData.csv` and `backStepNusseltNumberRefData.csv`.
3. Click **Open**.

The files are added to the **Tables** node, as shown below.



Add the tabular data to **Plots**, starting with the **Wall Shear Stress (x)** plot.

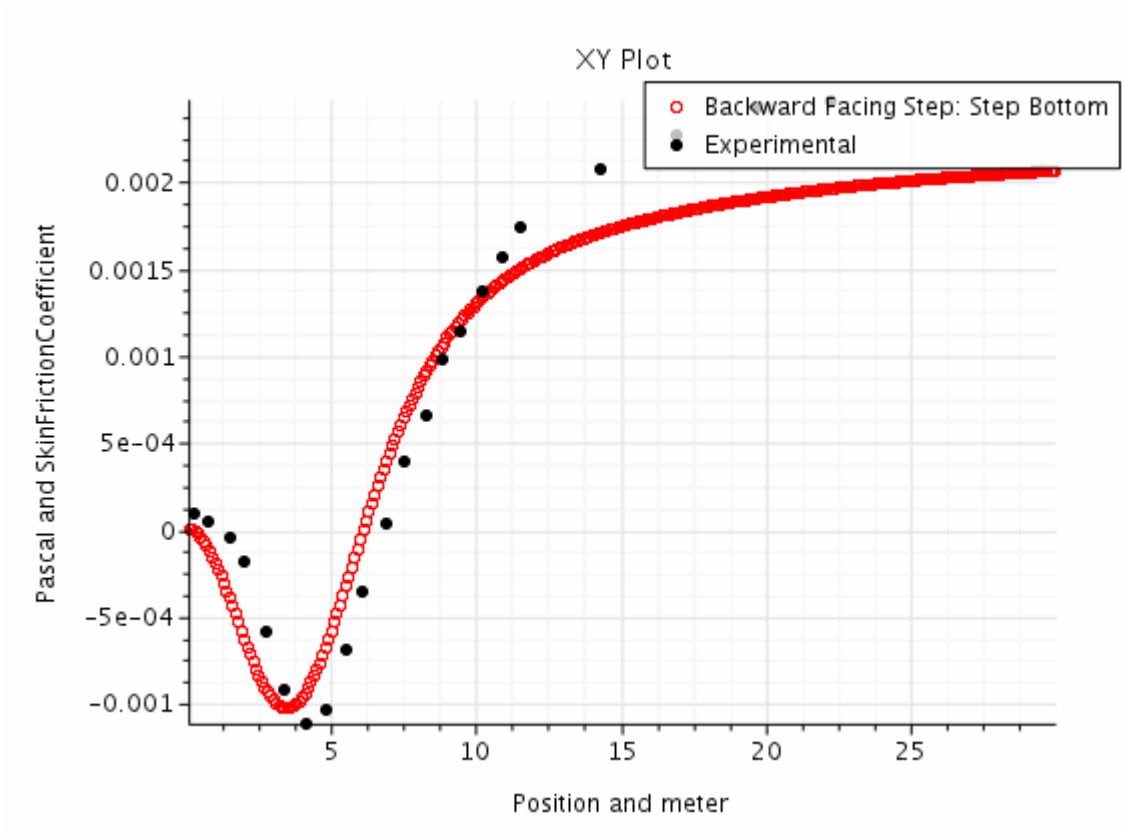
4. Right-click the **Wall Shear Stress (x)** > **Data Series** node and select **Add Data**.
5. Choose backStepSkinFrictionRefData Table.
6. Rename backStepSkinFrictionRefData Legend Name to `Experimental`.
7. Select the **Data Series** > **backStepSkinFrictionRefData** node and set the following values:

Property	Value
Table	**backStepSkinFrictionRefData**
X Column	**Position**
Y Column	**SkinFrictionCoefficient**

8. Select the **backStepSkinFrictionRefData** > **Symbol Style** node and set the following values:

Property	Value
Color	**black**
Shape	**Filled Circle**

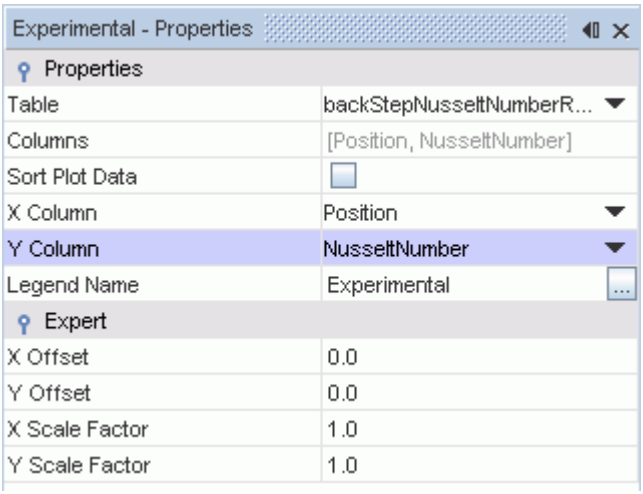
The Wall Shear Stress (x) plot is shown below.



Add the tabular data to the Nusselt Number plot.

9. Right-click the **Data Series** node and select **Add Data**.
10. Choose backStepNusseltNumberRefData Table.
11. Rename backStepNusseltNumberRefData Legend Name to `Experimental`.
12. Select the **Data Series** > **backStepNusseltNumberRefData** node and set the following values:

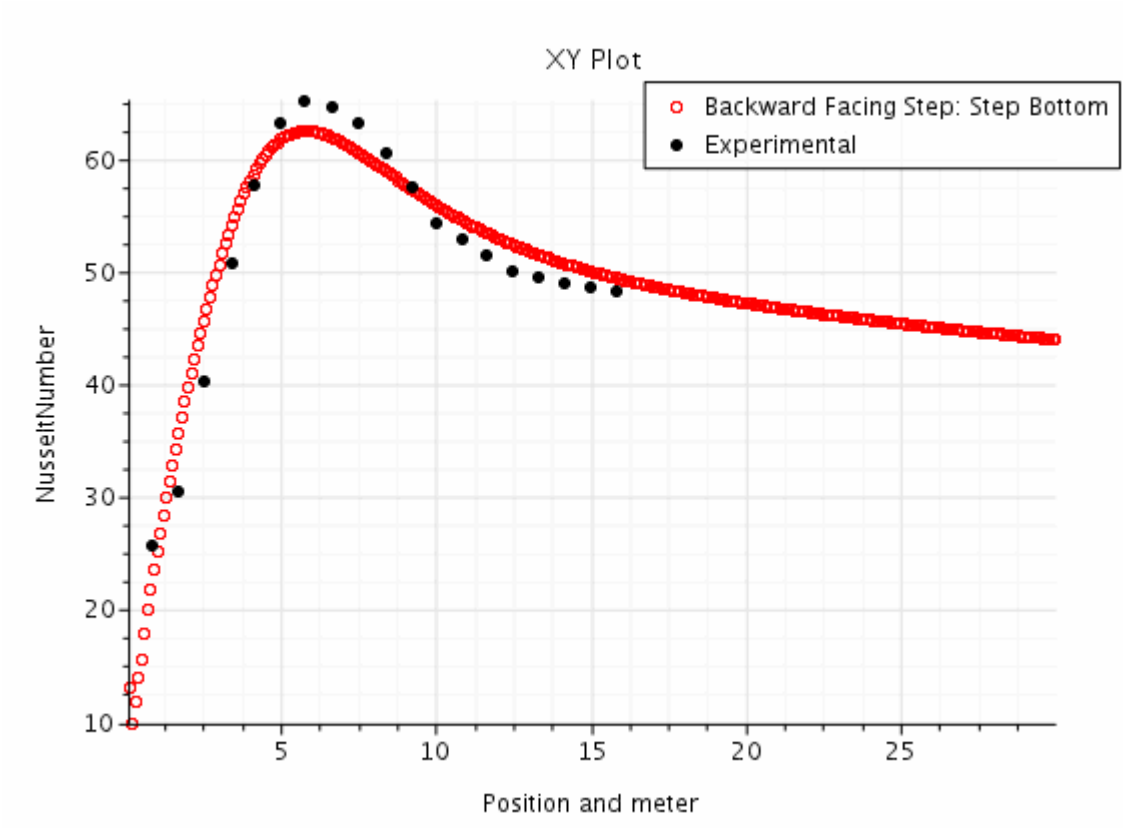
Property	Value
Table	**backStepNusseltNumberRefData**
X Column	**Position**
Y Column	**NusseltNumber**



13. Select the **Experimental** > **Symbol Style** node and set the following values:

Property	Value
Color	**black**
Shape	**Filled Circle**

The Nusselt Number plot is shown below.



Modifying the Step Height

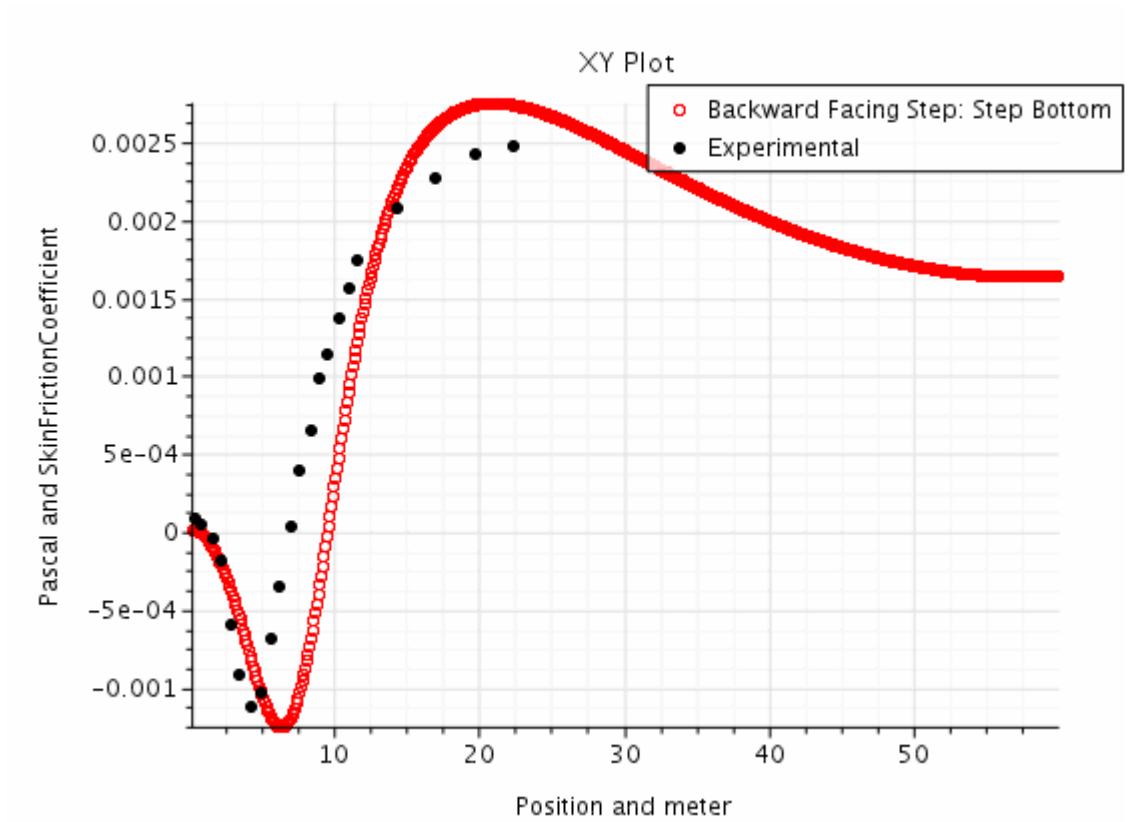
Modify the step height and see how this change affects the flow.

By changing the size of the fluid domain, while keeping the fluid properties unchanged, you can modify the Reynolds number.

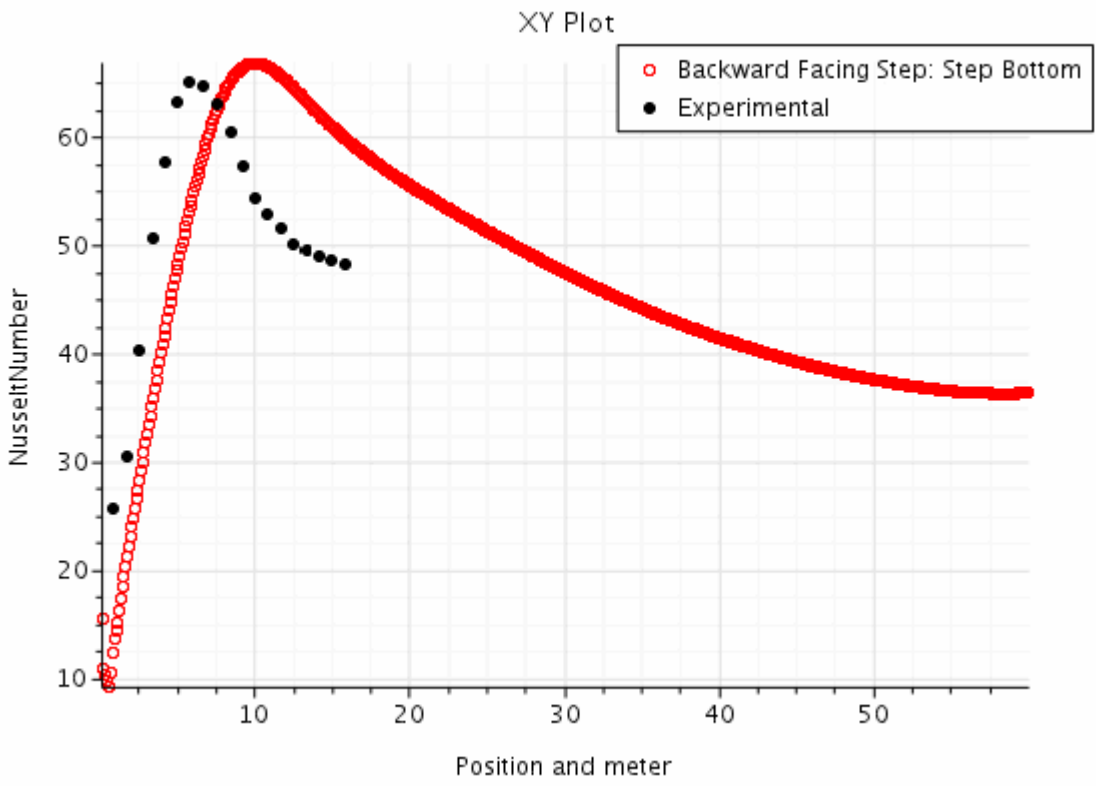
Change the size of the step height:

1. Expand the **Geometry** > **3D-CAD Models** > **Fluid Domain** > **Design Parameters** node.
2. Select the **Design Parameters** > **StepHeight** node and set *Value* to `2.0 m`
3. Click (**Generate Volume Mesh**).

   The geometry part is automatically updated with the new values and the 2D mesh is updated. The volumetric refinement region scales with the step height and retains its size relative to the fluid domain.
4. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to `1500`.
5. Click (**Run**) and observe how the different Reynolds number affects the flow:
   - Wall Shear Stress (x):



   - Nusselt Number:



Summary

This tutorial demonstrated the flow of an incompressible fluid over a backward facing step.

The tutorial covered the following:

- Creating a geometry model in 3D-CAD
- Applying fully developed flow data to the inlet boundary
- Plotting the wall shear stress and Nusselt number, and comparing these with experimental results

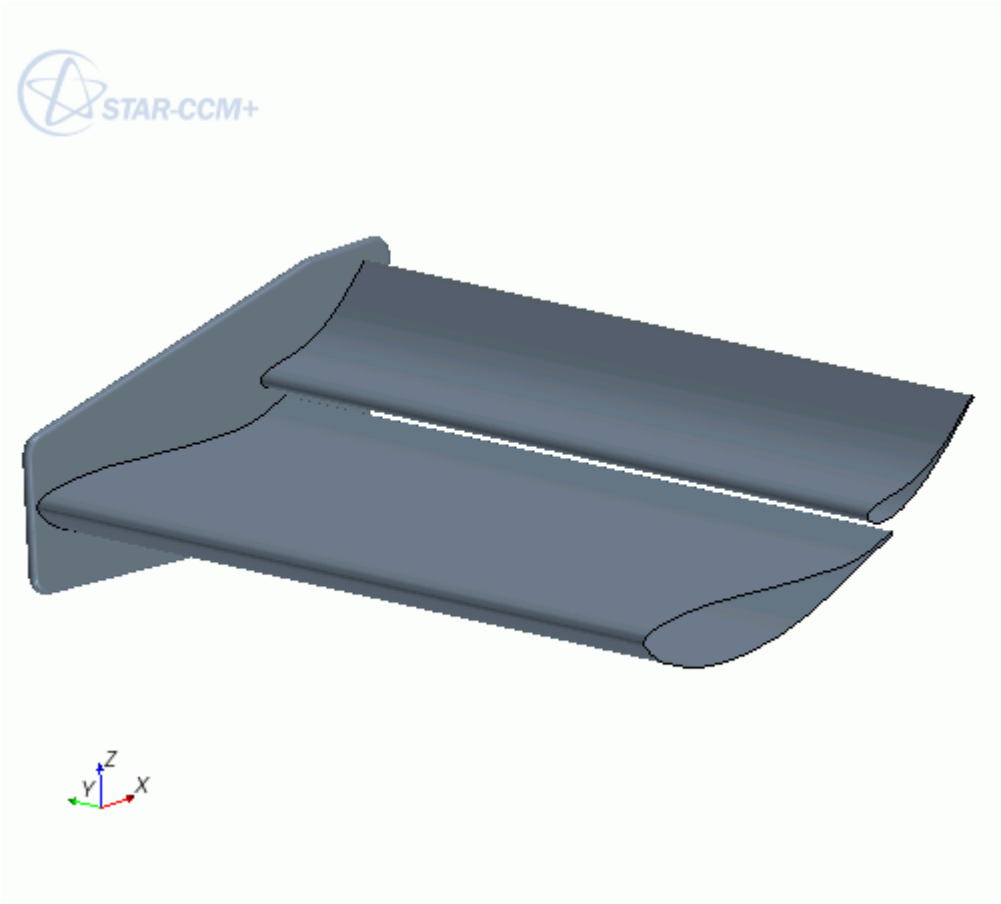
Backward Facing Step Bibliography

Vogel, J. C and Eaton, J. K., 1985. “Combined Heat Transfer and Fluid Dynamic Measurements Downstream of a Backward-Facing Step.” *AIAA J.* 23(2) pp. 163-171.

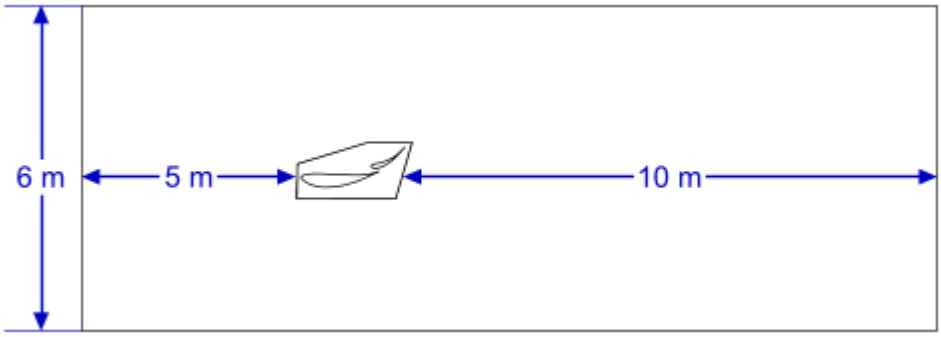
Adjoint Flow Solver: External Flow over a Dual Element Wing

This tutorial demonstrates how to run an adjoint flow analysis on a dual element wing for a race car.

The velocity of the flow is 100 kph. The objective of the analysis is to identify the factors that influence the downforce generated by the wing.



The angle of attack of the wing is 16.2 degrees. The fluid domain consists of a velocity inlet, pressure outlet, and symmetry boundaries. The wing is positioned in the domain as follows:



An adjoint flow analysis is split into two stages. In the first stage, you run the primary (or primal) flow solvers until the solution converges to machine precision. In the second stage, you run the adjoint solver that takes the primal flow solution as an input.

There are two aspects of achieving a well-converged solution:

- generating a high-quality volume mesh
- applying appropriate solution controls

In this tutorial, you address both aspects to achieve convergence at the level of machine convergence.

For this tutorial, you are provided with a simulation file that contains the following objects already defined:

- An automated mesh operation with mesh settings. To keep the file size to a minimum, the volume mesh is not provided—the mesh is ready for you to generate.
- A physics continuum with physics models required for the initial flow solution. The Adjoint Flow model is activated in a later step.
- Initial conditions.
- Regions and boundaries (including the boundary types).
- A mesh scene.

You require a double-precision version of STAR-CCM+ to complete this tutorial.

Contents:

Prerequisites
Loading the Simulation File and Visualizing the Mesh
Modifying Solver Parameters
Creating the Downforce Report, Monitor, and Plot
Setting Stopping Criteria
Running the Primal Solver
Activating the Adjoint Model
Creating the Adjoint Cost Function
Running the Adjoint Flow Solver
Visualizing Sensitivity of Downforce with Respect to the Flow
Summary

Prerequisites

The instructions in this tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

You must be familiar with the following techniques before attempting this tutorial:

Technique	Associated Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Loading the Simulation File and Visualizing the Mesh

Load `adjointWing_start.sim` and display the volume mesh.

It is best to run this tutorial in parallel, using at least 4 cores. This tutorial case has a large number of cells and can take a long time to run to machine convergence if using only one core.

Load the provided simulation file:

1. Start up STAR-CCM+ in a manner that is appropriate to your working environment. Remember to specify multiple cores.
2. Load `adjointWing_start.sim` from the `[INSTALL_DIR]/doc/startutorialsdata/adjoint/data` directory.

   STAR-CCM+ provides feedback on the loading process in the *Output* window.

The automated mesh operation with all the mesh settings have been created for you already.

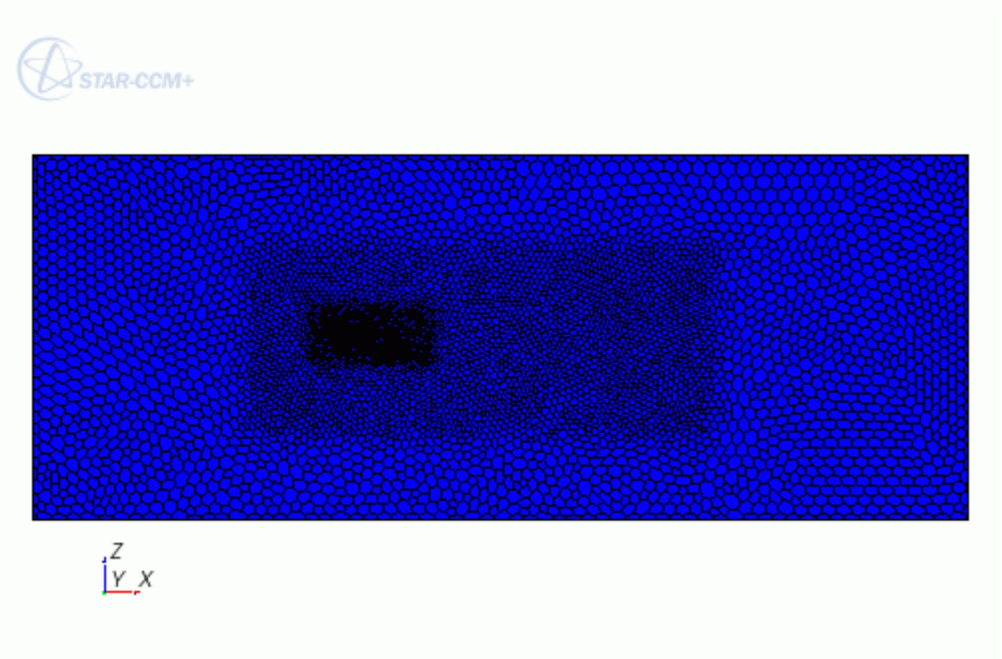
To generate the volume mesh:

3. Click (**Generate Volume Mesh**).

To visualize the mesh:

4. Right-click the **Scenes** > **Mesh Scenes 1** node and choose **Open**.
5. Right-click in a blank area of the scene and select **Apply Representation** > **Volume Mesh**.
6. Press the `<S>` key to display the side view of the geometry.

   The volume mesh appears as follows.



7. Save the simulation as `adjointWing_primal.sim`.

Modifying Solver Parameters

When performing an adjoint flow analysis, aim for convergence levels that approach machine precision.

This precision level means that residuals drop to a level around 1.0E-12. For most industrial cases, convergence to machine accuracy is difficult to achieve. In this tutorial, you apply some common steps for improving the convergence level.

Contents:

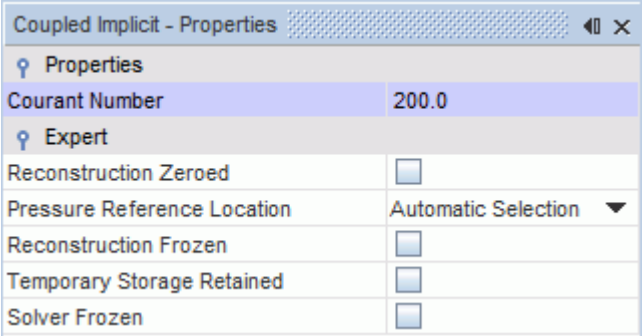
Increasing the Courant Number
Activating Grid Sequencing
Activating the Expert Driver

Increasing the Courant Number

For steady-state simulations, the Courant number controls the local pseudo-time step size. Increasing this value increases the rate of convergence. For this case, a Courant number of 200 accelerates convergence without causing the solver to become unstable. See Coupled Flow Controls and General Setup Recommendations.

To modify the Courant number:

- Select the **Solvers** > **Coupled Implicit** node and set *Courant Number* to `200`.



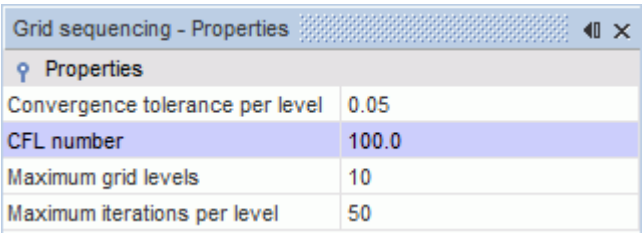
Activating Grid Sequencing

With the Coupled solver, you can activate grid sequencing to perform initialization followed by the computation of an approximate inviscid solution for the flow. See Grid-Sequencing and Expert Initialization.

To activate grid sequencing:

1. Select the **Solvers** > **Coupled Implicit** > **Expert Initialization** node and set *Method* to **Grid Sequencing**.

   The **Grid sequencing** sub-node appears.
2. Select the **Grid sequencing** node and set *CFL Number* to `100`.



Activating the Expert Driver

The expert driver is a tool that helps control and accelerate convergence of steady-state simulations that use the Coupled Implicit solver.

For this case, the Courant number is ramped from 0.25 to 200 over 100 iterations. This helps the solution converge during the first 100 iterations. See Using the Expert Driver.

To activate the expert driver:

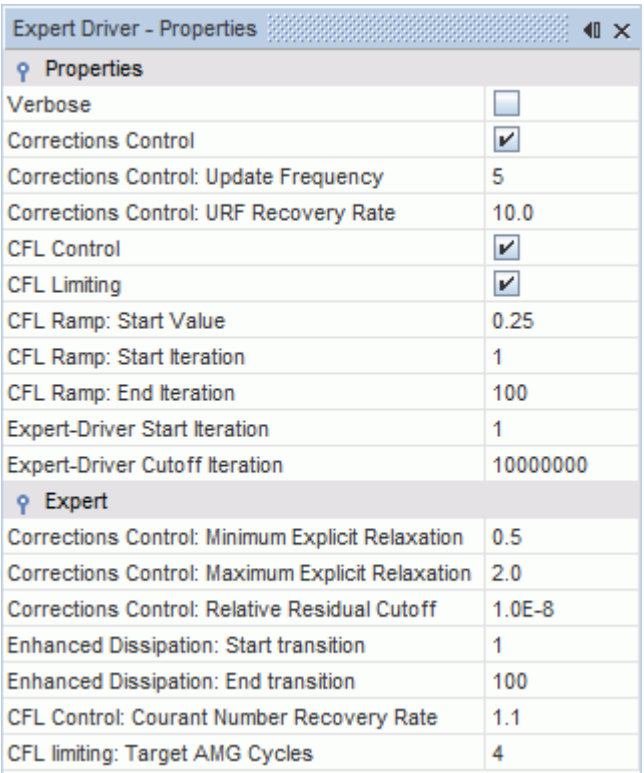
1. Select the **Solvers** > **Coupled Implicit** > **Solution Driver** node and set *Driver* to **Expert Driver**.

   The **Expert Driver** sub-node appears.
2. Select the **Expert Driver** sub-node and set the following values:

Property	Value
CFL Ramp: End Iteration	`100`

Property	Value
Corrections Control: Minimum Explicit Relaxation	`0.5`
Corrections Control: URF Recovery Rate	`10`
Corrections Control: Relative Residual Cutoff	`1.0E-8`

The final expert driver properties are displayed below.



Expert Driver - Properties

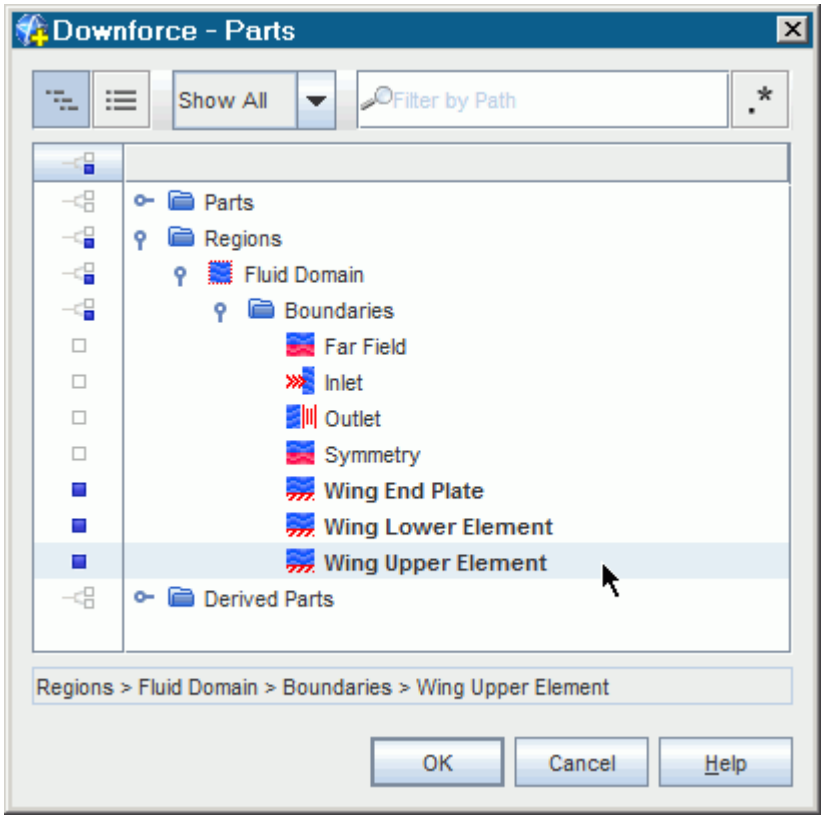
Properties	
Verbose	☐
Corrections Control	☑
Corrections Control: Update Frequency	5
Corrections Control: URF Recovery Rate	10.0
CFL Control	☑
CFL Limiting	☑
CFL Ramp: Start Value	0.25
CFL Ramp: Start Iteration	1
CFL Ramp: End Iteration	100
Expert-Driver Start Iteration	1
Expert-Driver Cutoff Iteration	10000000
Expert	
Corrections Control: Minimum Explicit Relaxation	0.5
Corrections Control: Maximum Explicit Relaxation	2.0
Corrections Control: Relative Residual Cutoff	1.0E-8
Enhanced Dissipation: Start transition	1
Enhanced Dissipation: End transition	100
CFL Control: Courant Number Recovery Rate	1.1
CFL limiting: Target AMG Cycles	4

Creating the Downforce Report, Monitor, and Plot

The aim of this adjoint flow analysis is to investigate the design with a view to increase the amount of downforce that is generated by the wing.

To monitor the downforce, create a force report:

1. Right-click the **Reports** node and select **New Report** > **Force**.
2. Rename the report to `Downforce`.
3. Select the **Downforce** node and set *Direction* to `[0.0, 0.0, -1.0]`.
4. Edit the *Parts* property, expand the **Regions** > **Fluid Domain** > **Boundaries** node, and select **Wing End Plate**, **Wing Lower Element**, and **Wing Upper Element**.



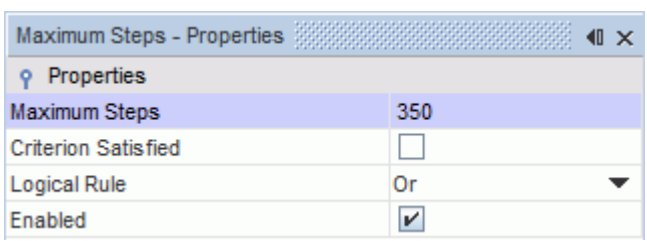
5. Click **OK**.
6. Right-click **Reports** > **Downforce** and select **Create Monitor and Plot from Report**.
7. Right-click **Plots** > **Downforce Monitor Plot** and choose **Open**.

Setting Stopping Criteria

For this simulation, the primal solver is left to run for 350 iterations.

To set the stopping criteria:

- Select **Stopping Criteria** > **Maximum Steps** and set *Maximum Steps* to `350`.



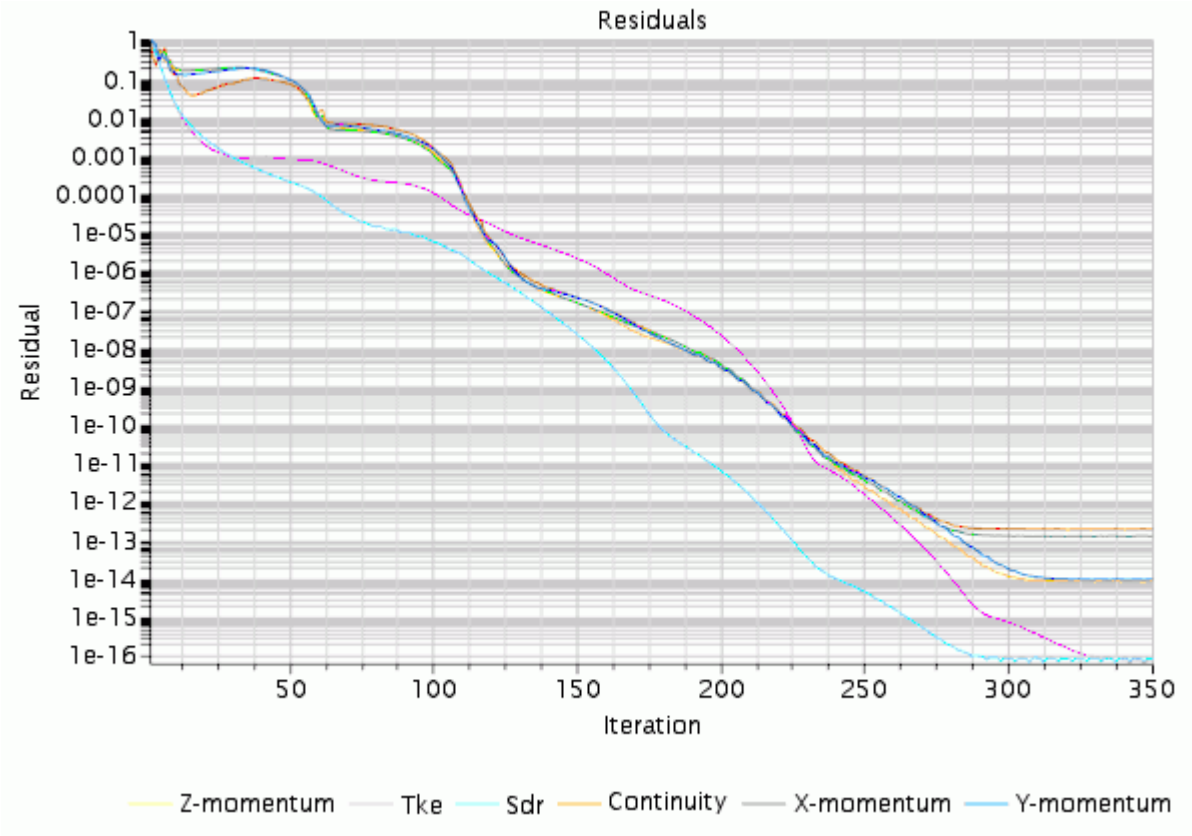
Running the Primal Solver

Run the simulation.

To run the simulation:

1. In the **Solution** toolbar, click (**Run**).
2. When the solver is finished, save the simulation.

   The *Residuals* plot is displayed below:



Note: Your residuals may differ slightly, depending on the accuracy of your machine.

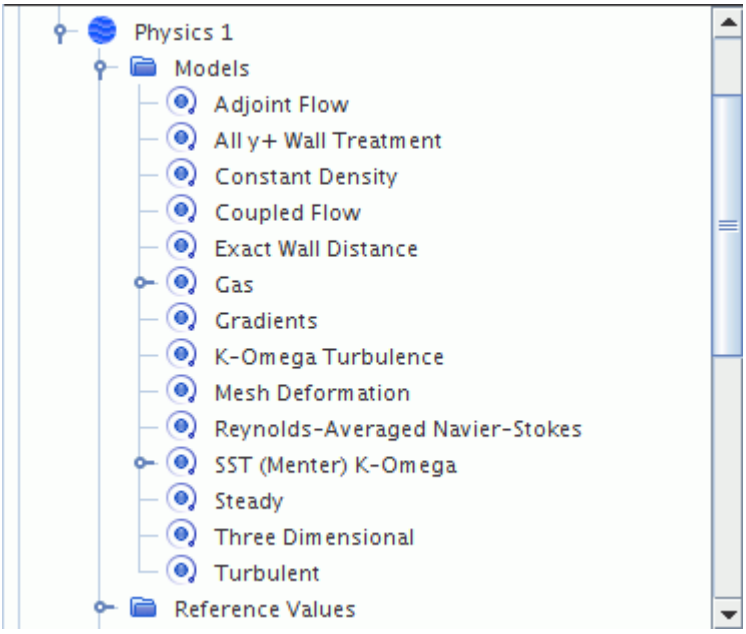
Activating the Adjoint Model

Activate the Adjoint Flow physics model.

To activate the model:

1. Right-click the **Continua** > **Physics 1** node and choose **Select Models...**.

   The *Physics 1 Model Selection* dialog appears.
2. Select **Adjoint Flow** from the *Optional Models* group box.
3. Click **Close**.
4. To view the active physics models, open the **Physics 1** > **Models** node.



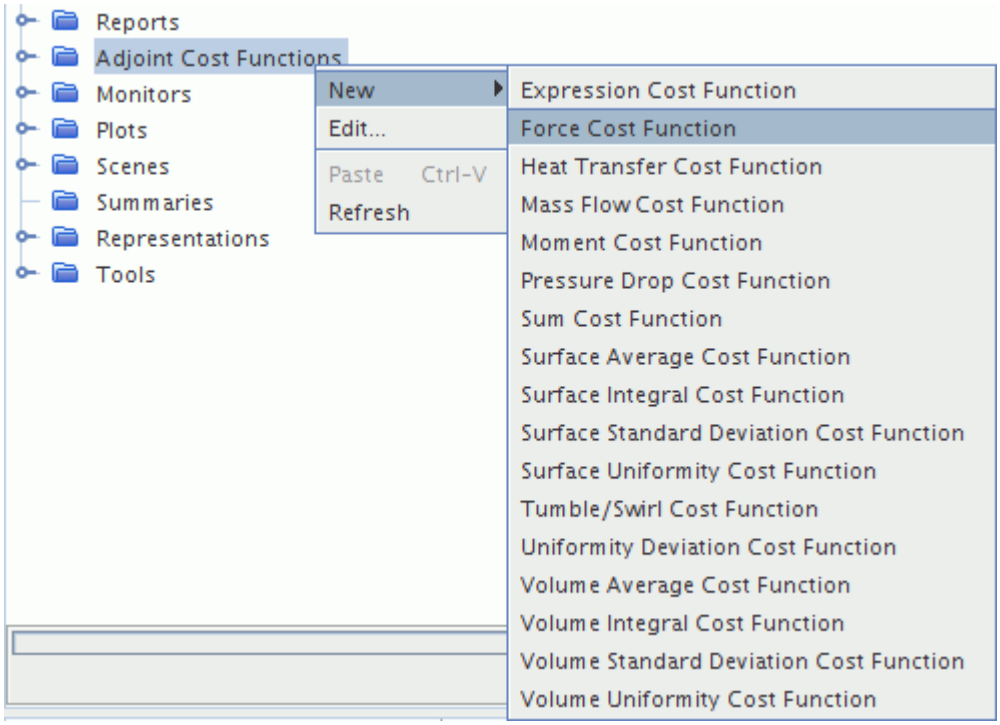
Creating the Adjoint Cost Function

The adjoint cost function specifies which engineering quantity is the subject of the sensitivity analysis.

In this tutorial you want to know how to increase the amount of downforce that is generated by the wing. A force cost function provides this information.

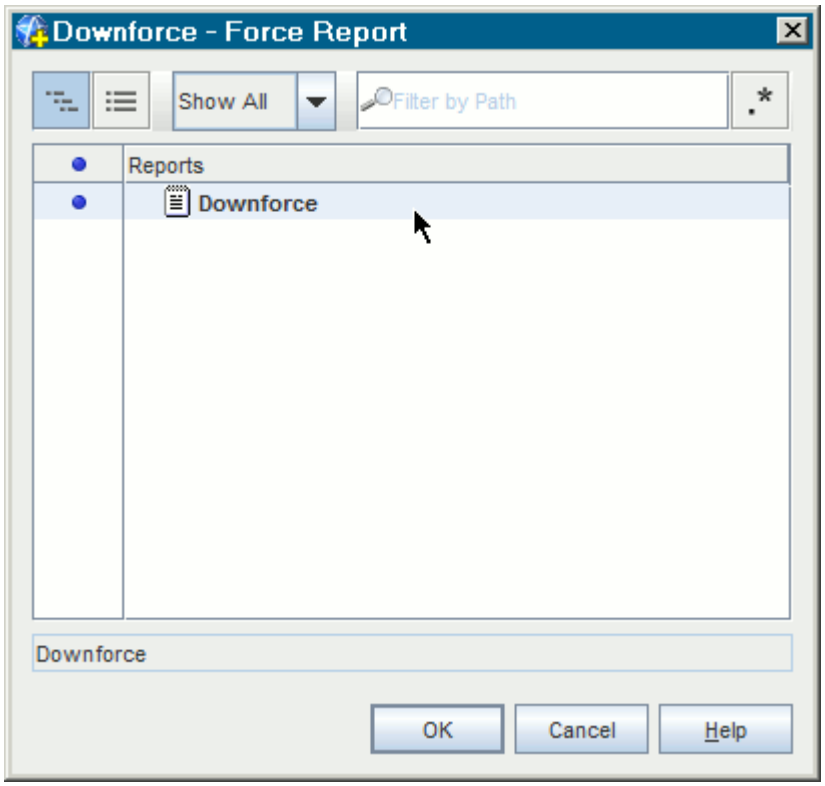
To create a force cost function:

1. Right-click the **Adjoint Cost Functions** node and select **New** > **Force Cost Function**.



A **Force** sub-node is created.

2. Rename the force cost function to `Downforce`.
3. Select the **Downforce** node and set *Force Report* to **Downforce**.



The adjoint force cost function adds the following monitors to the **Monitors** node: **Downforce::P**, **Downforce::U**, **Downforce::V**, and **Downforce::W**.

Running the Adjoint Flow Solver

The adjoint flow solver is run from the Solvers node, as opposed to the primal flow solver which is run using the Run button in the Solution toolbar.

The primal flow solution is frozen while the adjoint flow solver runs. Avoid stepping or running the primal flow solver after running the adjoint flow solver.

To run the adjoint flow solver:

1. Select **Stopping Criteria** > **Maximum Steps** and set *Maximum Steps* to `600`.
2. Right-click the **Solvers** > **Adjoint** node and select **Run**.

   The adjoint solver runs.
3. When the solver is finished, save the simulation as `adjointWing_adjoint.sim`.

Visualizing Sensitivity of Downforce with Respect to the Flow

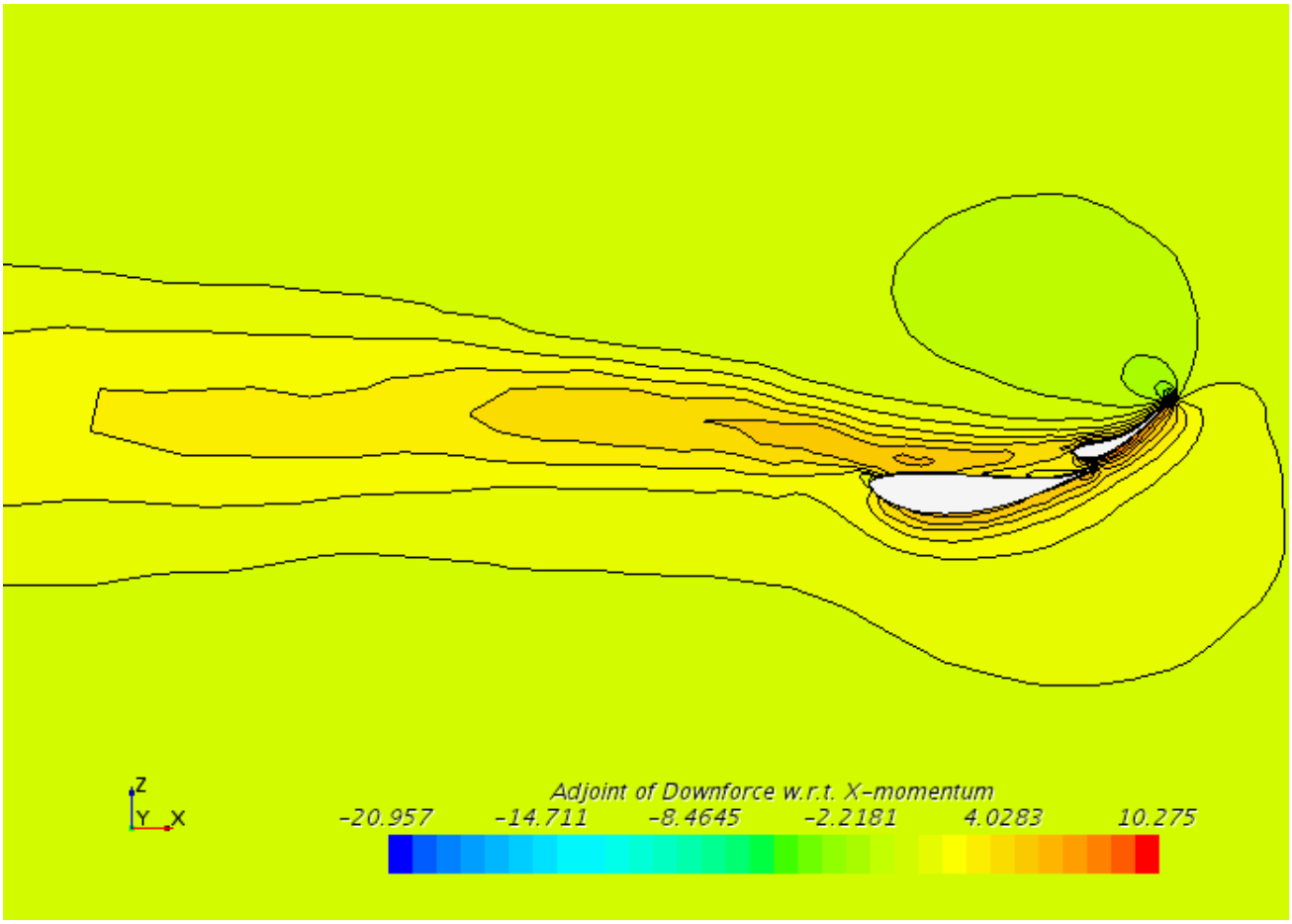
Create a scalar scene to display the sensitivity of downforce with respect to the flow field.

To create a scalar scene:

1. In the **Vis** toolbar, click (**Create/Open Scenes**) and select **Scalar**.
2. Press the `<S>` key to display the side view.
3. Click **scene/plot**.
4. Right-click **Displayers** > **Scalar 1** > **Parts** and select **Edit...**.

   The *Edit* dialog appears.
5. Select **Regions** > **Fluid Domain** > **Boundaries** > **Symmetry**.
6. Click **OK**.
7. Select the **Scalar 1** > **Scalar Field** node and set *Function* to **Adjoint** > **Downforce** > **Adjoint of Downforce w.r.t. X-momentum**.
8. Select the **Scalar 1** node and set *Contour Style* to **Smooth Filled + Lines**.
9. In the **Vis** toolbar, click (**Save-Restore-Select Views**) and select **Projection Mode** > **Parallel**.

The scalar scene appears as follows:



This scene displays the sensitivity of the desired adjoint cost function with respect to the flow field. In this case, downforce is the desired adjoint cost function. Directing X-momentum towards areas with positive values will increase the downforce. Similarly, directing X-momentum away from areas with negative values will also increase downforce.

Summary

This tutorial demonstrated how to run an adjoint flow analysis on a dual element wing for a race car.

This tutorial has introduced the following STAR-CCM+ features:

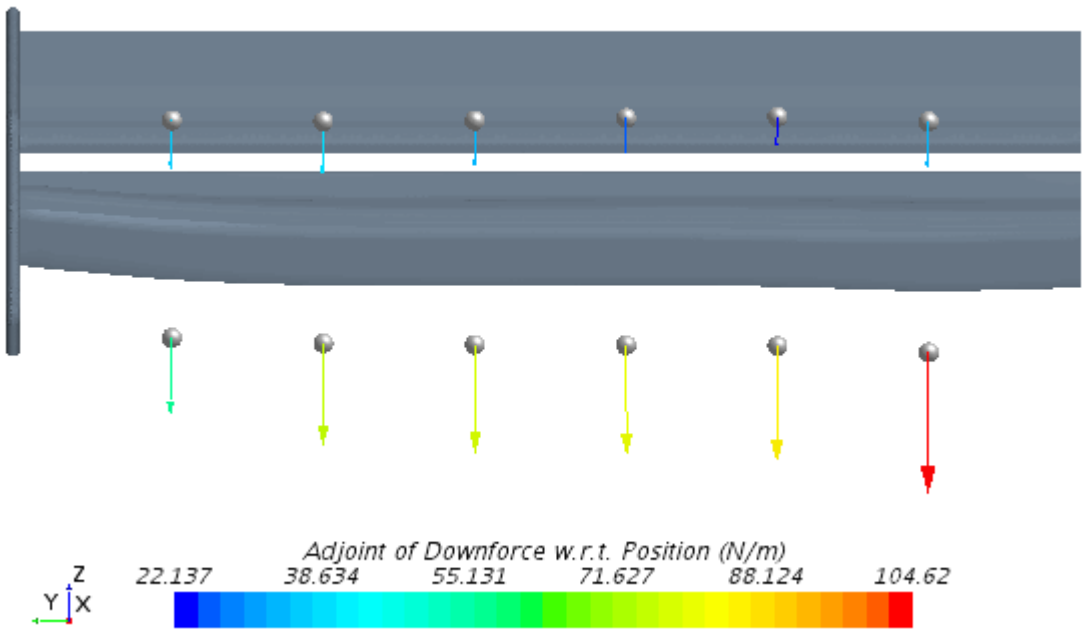
- Loading a simulation file and visualizing a volume mesh.
- Running the primal solver to machine precision.
- Setting the stopping criteria.
- Reporting, monitoring and plotting.
- Setting up Grid Sequencing and the Expert Driver.
- Selecting the physics model for adjoint flow analysis.
- Setting up the Adjoint cost function for force.
- Running the Adjoint Flow solver.
- Calculating the Adjoint Mesh.

- Visualizing the adjoint solution.

Adjoint Flow: Wing Shape Optimization

This tutorial shows how to exploit an adjoint flow analysis to set up a shape optimization with respect to a specified cost function.

The tutorial continues from the final solution obtained in the **Adjoint Flow Solver: External Flow over a Dual Element Wing** tutorial. In particular, the tutorial shows how the shape of the lower element can be modified to increase the downforce generated by the wing.



The shape optimization process uses the morphing procedure to deform the geometry of interest. Specified control points are displaced by amounts calculated using mesh sensitivities that the Adjoint Mesh solver provides. The displacement of the control points controls the deformation calculated by the morpher. The displaced positions of the specified control points are set to maximize/minimize the cost function of interest.

For more information about the shape optimization workflow using the Adjoint solvers see the User Guide - Setting up a Shape Optimization.

Contents:

Prerequisites
Loading the Simulation
Setting Mesh Deformation Boundary Conditions
Creating the Initial Control Points
Computing the Mesh Sensitivity
Performing the First Shape Optimization

Running The Simulation
Visualizing the Results
Performing Multiple Optimization Cycles
Summary

Prerequisites

The instructions in the Adjoint Flow: Wing Shape Optimization tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Associated Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+
Running an adjoint flow analysis	Adjoint Flow Solver: External Flow over a Dual Element Wing

Note: You require a double-precision version of STAR-CCM+ to complete this tutorial.

Loading the Simulation

For this tutorial, you are provided with a simulation file that contains pre-defined objects.

Due to the large number of cells, this tutorial simulation should be run in parallel mode on at least 4 cores.

To load the simulation file:

1. Start a double precision version of STAR-CCM+ in a manner that is appropriate to your working environment and select **Load Simulation** from the menu bar.
2. Load the `adjointWing_adjoint.sim` file that you created during the Adjoint Flow Solver: External Flow over a Dual Element Wing tutorial. Make sure that you specify multiple cores.

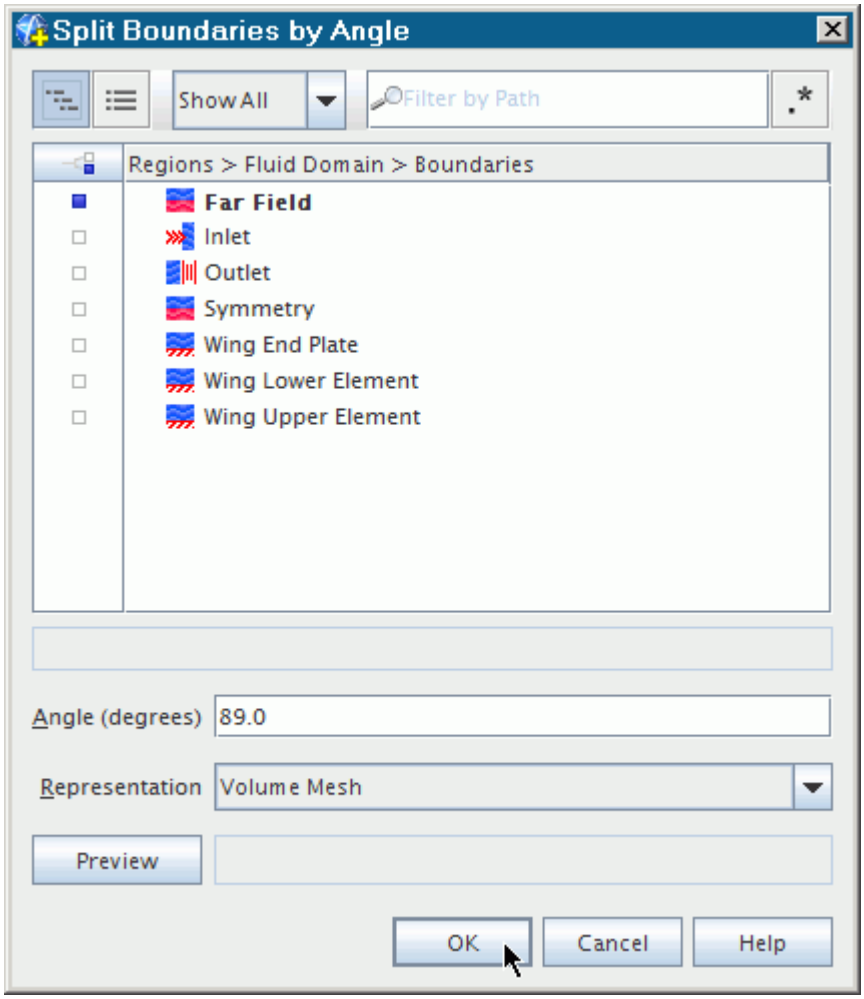
Setting Mesh Deformation Boundary Conditions

The morpher redistributes mesh vertices according to the design points and their associated displacements. The associated displacements are generated during the optimization cycle on the basis of mesh sensitivity.

The morpher algorithm and its adjoint require each symmetry boundary to have no more than one planar surface:

1. Right-click the **Regions** > **Fluid Domain** > **Boundaries** > **Far Field** node and select **Split by Angle**.

   The *Split Boundaries by Angle* dialog appears.
2. Accept the default settings by clicking **OK**.



This tutorial focuses on the optimization of the shape of the wing lower element, whereas the shape of other wing components is fixed.

To specify the morphing conditions set the appropriate *Method* for the **Morpher** node within the **Physics Conditions** node of each boundary.

3. Select the **Regions** > **Fluid Domain** > **Boundaries** > **Wing Lower Element** > **Physics Conditions** > **Morpher Specification** node and set *Specification* to **Floating**.
4. Make sure that for the **Inlet, Outlet**, **Wing End Plate** and **Wing Upper Element** boundaries the *Specification* for the **Morpher Specification** node is set to **Fixed**.
5. Save the simulation as `adjointWing_shapeOptimization.sim`.

Creating the Initial Control Points

You create initial control points at the beginning of the shape optimization which the adjoint solver uses to compute the mesh sensitivity.

The displacements of these points to new optimized positions are then used to calculate the corresponding mesh deformation, that is, the optimized shape.

You start by creating a new **Geometry Scene** and editing it in the following way:

1. Right-click the **Scenes** node and select **New Scene** > **Geometry**.
2. Right-click the **Geometry Scene 1** > **Displayers** > **Geometry 1** > **Parts** node and select **Edit...**.

   The *Edit* dialog appears.
3. In the *Edit* dialog right click on a blank area and select **Deselect All**.
4. Expand the **Regions** > **Fluid Domain** > **Boundaries** node and select **Wing End Plate**, **Wing Lower Element**, and **Wing Upper Element**.
5. Click **OK**.
6. Right-click the **Geometry Scene 1** > **Outline 1** node and select **Delete**.
7. In the Vis toolbar click (**Save-Restore-Select Views**) and select **Projection Mode** > **Parallel**.

To create the initial control points:

8. Right-click the **Point Sets** node and select **New Point Set** > **Line...**.
9. In the *Create Point Set along Line* dialog, set the following properties:

Property	Value
Point 1	`[0.1 m, 0.1 m, 0.01 m]`
Point 2	`[0.1 m, 0.6 m, 0.01 m]`
Number	`6`
Display	**Existing Displayer** > **Geometry 1**

10. Click **Create**.
11. Create another set of control points along a line and set the following properties:

Property	Value
Point 1	`[0.1 m, 0.1 m, 0.15 m]`
Point 2	`[0.1 m, 0.6 m, 0.15 m]`
Number	`6`
Display	**Existing Displayer** > **Geometry 1**

12. Click **Create**, then **Close**.

    You can see the locations of the created initial control points in *Geometry Scene 1*.
13. Rename the **Line Point Set** node to `Lower` and rename the **Line Point Set 2** node to `Upper`.

Computing the Mesh Sensitivity

When the mesh sensitivity computation is performed, the Adjoint Mesh solver calculates the variation of the cost function of interest, in each spatial direction, at each control point position.

To compute the mesh sensitivity:

1. Select the **Solvers** > **Mesh Deformation** node and make sure that *Enable Automatic Thinning* is activated.

   Although you activate **Automatic Thinning** under the **Mesh Deformation** node, this property also affects the computation of the mesh sensitivity. You thin out the number of points, thus speeding up the mesh sensitivity computation.

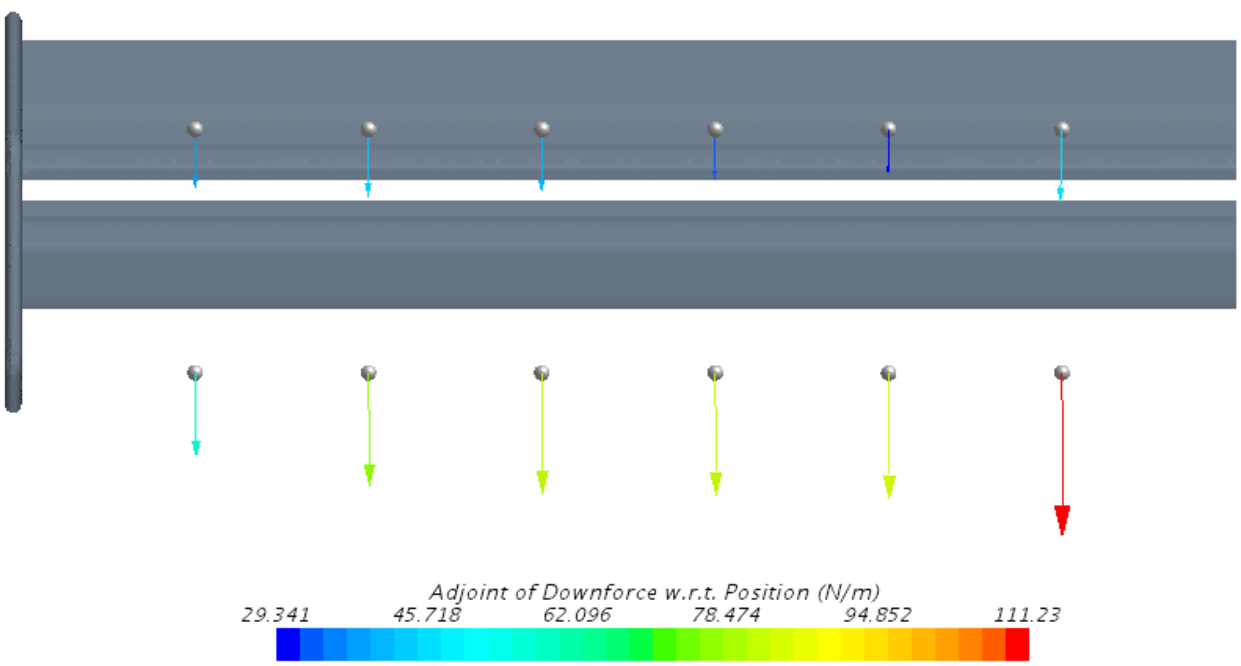
2. Right-click the **Solvers** > **Adjoint Mesh** node and select **Compute Mesh Sensitivity**.

   The mesh sensitivity computation starts and takes several seconds to complete.

To view the mesh sensitivity, visualize the **Adjoint of Downforce w.r.t. Position** field:

3. Right-click the **Geometry Scene 1** > **Displayers** node and select **New Displayer** > **Vector**.
4. Edit the **Diplayers** node and set the following properties:

Node	Property	Setting
Vector 1		
└ **Parts**	*Parts*	**(Point Sets) [Lower, Upper]**
└ **Vector Field**	*Function*	**Adjoint of Downforce w.r.t. Position**
└ **Glyph**	*Vector Scale*	**Absolute**
	Vector Style	**2D Filled**



You can now use the computed mesh sensitivities for each control point to deform the mesh.

Performing the First Shape Optimization

After computing the mesh sensitivity with respect to the point sets, you continue by deforming the mesh accordingly. You now perform the first step of the shape optimization.

Generally, you can perform a shape optimization by employing an external optimization tool that uses the mesh sensitivities computed by the Adjoint Mesh solver. This optimization tool calculates the new positions of the control points based on an optimization algorithm of your choice.

For this tutorial, you use a first-order Steepest Descent algorithm with a step size of 0.0001 to update the point set positions. To update the control point positions for the shape optimization cycles, you take into account the total displacement of a control point from its original position. You perform this optimization step completely within STAR-CCM+. You achieve this by associating a user-defined field function that defines the Steepest Descent deformation with the morpher. This user-defined field function uses a built-in field function for the cumulative morpher displacement.

To perform the shape optimization:

1. Right-click the **Tools** > **Field Functions** node and select **New** > **Vector**.
2. Rename this field function to `Displacement` and set the following properties:

Property	Setting
Function Name	`Displacement`
Definition	`$$CumulativeMorpherDisplacement + 1.e-04*$$ {Adjoint1::CoordAdjoint}`

   The field function is defined such that it multiplies the step size of 0.0001 with the **Adjoint of Downforce w.r.t. Position** field function. You obtain this field by computing the mesh sensitivity of the **Downforce** cost function on the point sets (see Computing the Mesh Sensitivity).

3. Multi-edit the **Point Sets** > **Lower** > **Physics Values** and the **Point Sets** > **Upper** > **Physics Values** nodes and set the following properties:

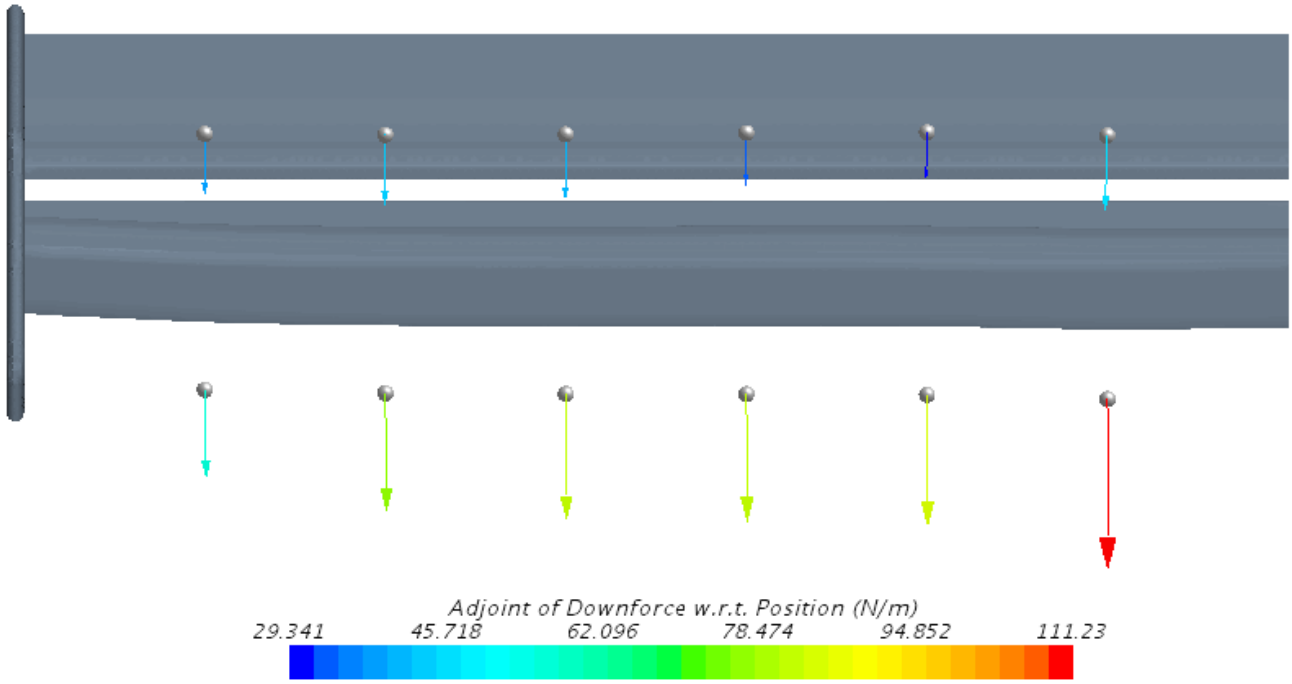
Node	Property	Setting
└ **Morpher: Displacement**	*Method*	**Field Function**
	Vector Function	**Displacement**

4. Right-click the **Solvers** > **Mesh Deformation** node and select **Deform Mesh**.

To display the improved wing lower element shape, update the **Geometry Scene 1**:

5. Right-click a blank area in the **Geometry Scene 1** and select **Refresh**.

The shape obtained after the first optimization cycle is displayed in the following image.



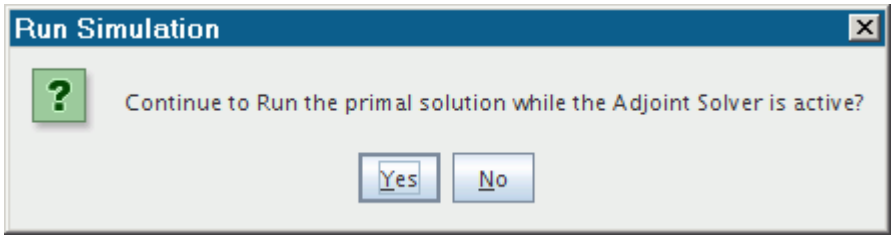
Running The Simulation

Run the simulation to observe how the downforce generated by the wing increases due to the optimization of the wing lower element shape.

1. Right-click the **Plots** > **Downforce Monitor Plot** node and select **Open**.
2. Select the **Stopping Criteria** > **Maximum Steps** node and set the *Maximum Steps* to `1000`.

To run the primal solver:

3. Click (**Run**) to run the simulation.
   The *Run Simulation* dialog appears.
4. Click **Yes** to confirm.



To run the adjoint solver:

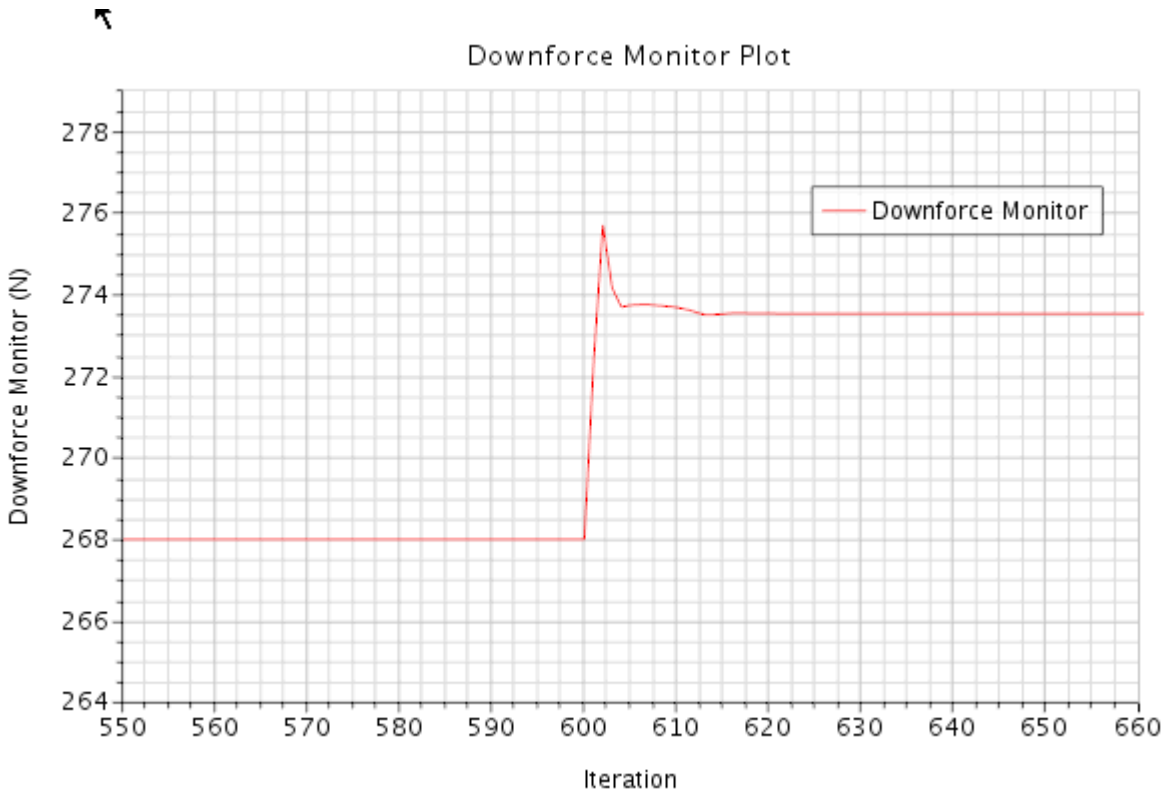
5. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to `1300`.
6. Right-click the **Solvers** > **Adjoint** node and select **Run**.

7. When both solvers are complete, save the simulation.

Visualizing the Results

Visualize how the first shape optimization cycle improves the downforce.

1. Right-click the **Plots** > **Downforce Monitor Plot** node and select **Open**.
2. In the toolbar, click the **Zoom Selected Plot** button.
3. In the Downforce Monitor Plot, draw a box around the area where there is an increase in downforce (at Iteration 600).



Performing Multiple Optimization Cycles

You can perform as many optimization cycles as you think appropriate, repeating the process at every cycle.

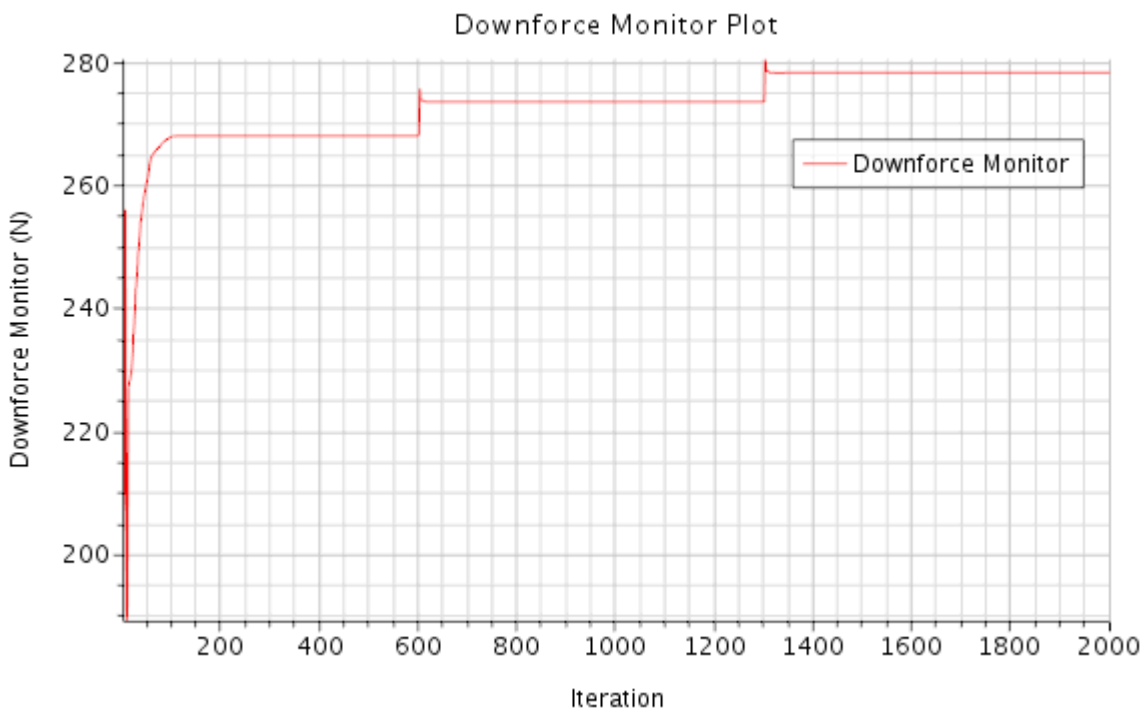
The following steps outline the process of each optimization cycle:

- Compute the Mesh sensitivity
- Deform the Mesh
- Run the Primal Solver
- Run the Adjoint Solver

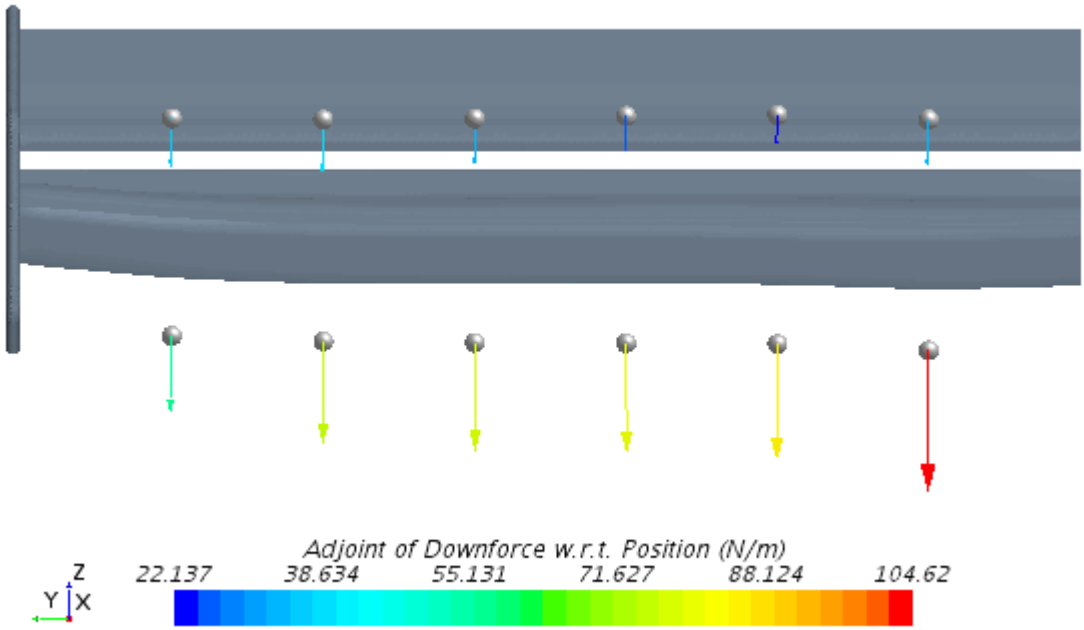
Perform a second shape optimization:

1. Compute the mesh sensitivity:
   a) Right-click the **Solvers** > **Adjoint Mesh** node and select **Compute Mesh Sensitivity**.
2. Deform the mesh:
   a) Right-click the **Mesh Deformation** node and select **Deform Mesh**.
3. Run the Primal solver:
   a) Set the **Stopping Criteria** > **Maximum Steps** to `1700`.
   b) Click (**Run**).
4. Run the Adjoint Solver:
   a) Set the **Stopping Criteria** > **Maximum Steps** to `2000`.
   b) Right-click the **Solvers** > **Adjoint** node and select **Run**.

The Downforce Plot history is shown below.



The following image shows the geometry after the second optimization cycle.



Summary

This tutorial presented the general workflow to set up a shape optimization within the adjoint framework.

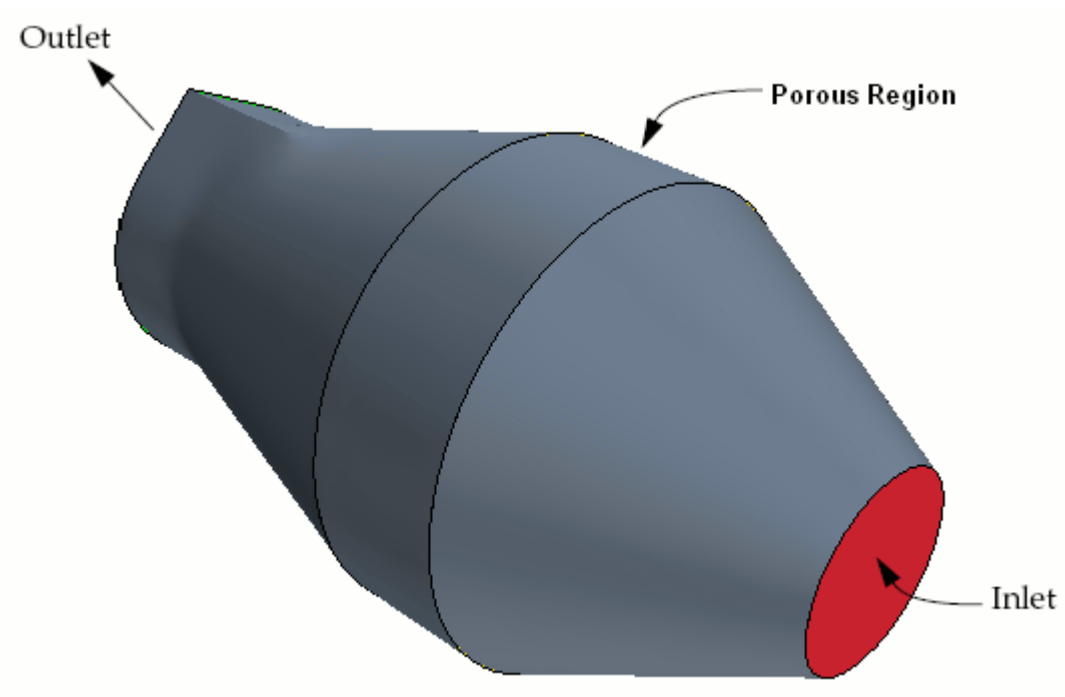
The following concepts have been introduced:

- Creating control points.
- Computing the mesh sensitivity.
- Deforming the mesh using the morphing procedure.
- Running the solver to observe the cost function improvement.
- Performing additional optimization cycles.

Porous Resistance: Isotropic Media

This tutorial models flow through a catalyst geometry.

The geometry is shown below. The catalyst material at the center of this geometry is modeled as a cylindrical porous region of length 0.03 m and diameter 0.1 m.



In the porous region, the theoretical pressure drop per unit length can be determined using the equation:

$$\frac{\Delta p}{L} = -\left(P_i|v| + P_v\right)v$$

Where v is the superficial velocity through the medium and P_i, P_v are coefficients defining the porous resistance, known as the inertial resistance and viscous resistance, respectively.

Values for the resistance coefficients can be measured experimentally or derived using various empirical relationships, depending on the exact nature of the problem. In this case, P_i = 25 kg/m^4 and P_v = 1500 kg/m^{3}s. These values are roughly what you would expect from an isotropic porous catalyst.

Contents:

Prerequisites
Importing the Mesh and Naming the Simulation
Visualizing the Imported Mesh
Scaling the Mesh
Setting Up the Models
Setting Boundary Conditions and Values

Specifying Porosity Coefficients
Monitoring the Run Progress
Setting Stopping Criteria
Visualizing the Solution
Running the Simulation
Visualizing Results
Validating the Results
Summary

Prerequisites

The instructions in the Porous Resistance: Isotropic Media tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Importing the Mesh and Naming the Simulation

Import a polyhedral mesh from `doc/startutorialsdata/incompressibleFlow/data`, and save and name the new simulation.

1. Start up STAR-CCM+ in a manner that is appropriate to your working environment.
2. Select the New Simulation option from the menu bar.

Continue by importing the mesh and naming the simulation. A polyhedral cell mesh has been prepared for this analysis and is stored in a `ccm` format file.

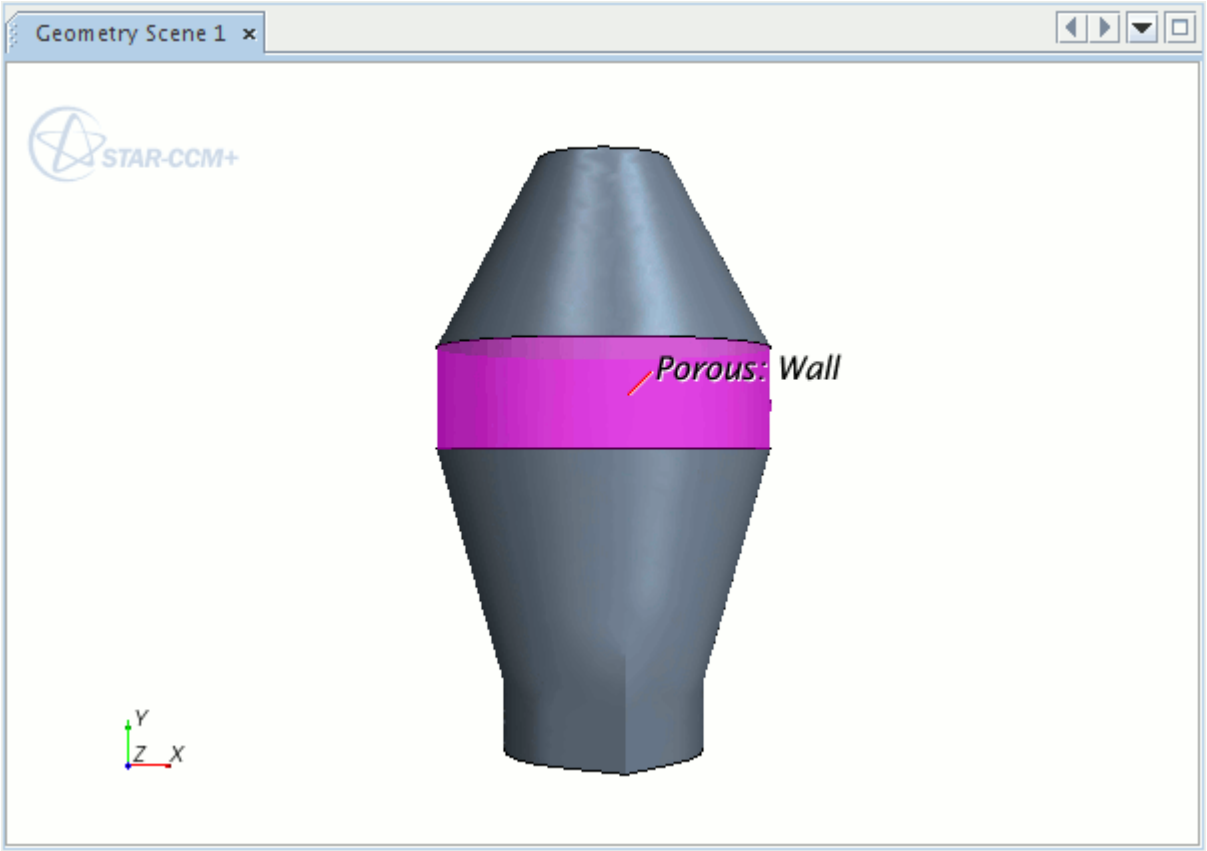
3. Select **File** > **Import** > **Import Volume Mesh** from the menu bar.
4. In the *Open* dialog, navigate to the `doc/startutorialsdata/incompressibleFlow/data` subdirectory of your STAR-CCM+ installation directory and select the file `catalyst.ccm`.
5. Click **Open** to start the import.

   STAR-CCM+ provides feedback on the import process in the *Output* window. Two mesh regions named **Fluid** and **Porous** are created under the **Regions** node to represent the two parts of the solution domain.
6. Save the new simulation as `isotropicPorousMedia.sim`.

Visualizing the Imported Mesh

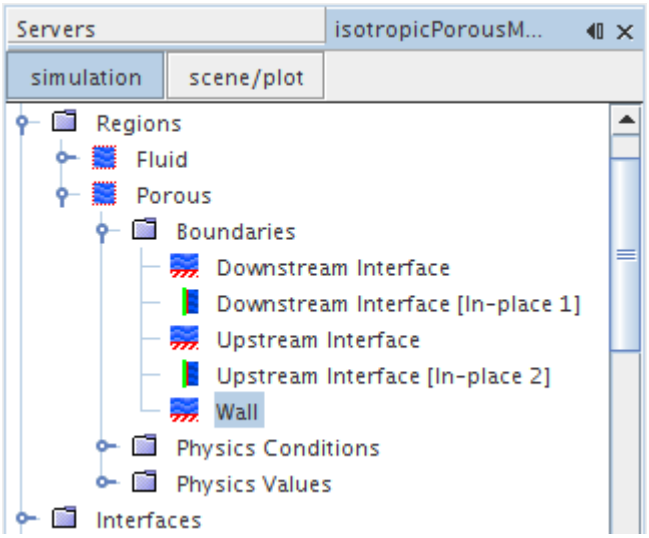
A geometry scene was automatically created after the import process. Examine the *Geometry Scene 1* display in the *Graphics* window. Initially all parts of the mesh are shown as solid, colored surfaces.

1. Click one of the parts.



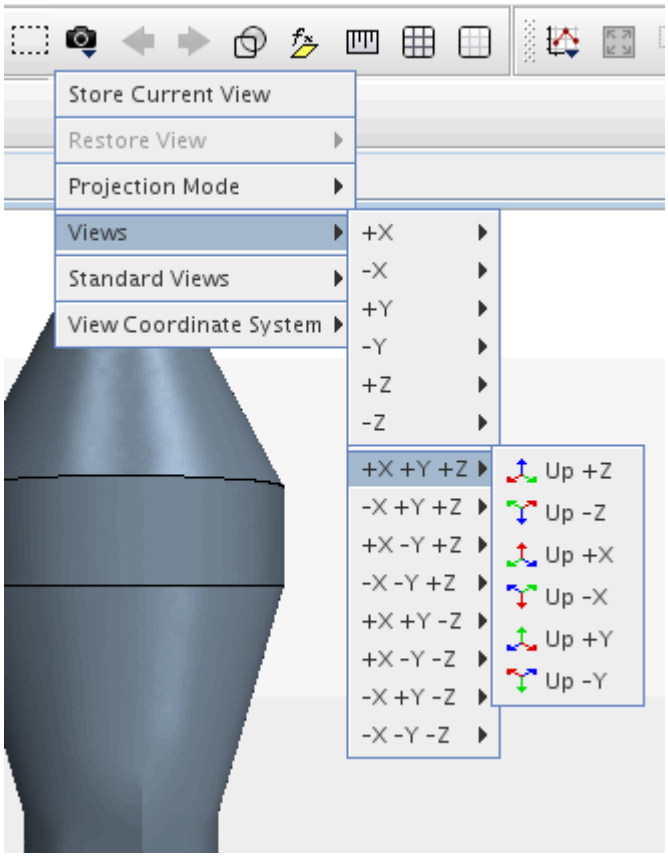
The boundary selected is highlighted and a label with its name appears on the display.

In the *Simulation* window, the node that corresponds to this boundary is also highlighted.

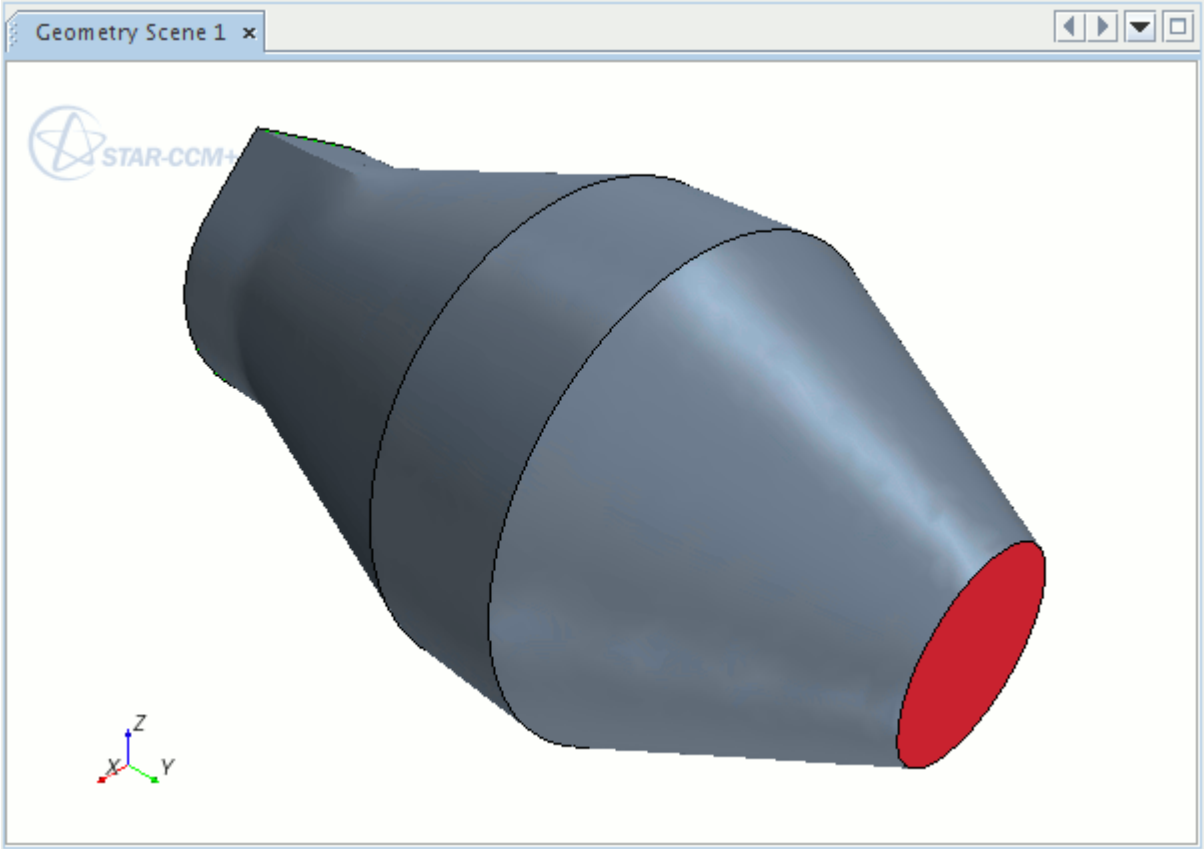


Similarly, if any other boundary node is selected in the *Simulation* window, the surface corresponding to that boundary are highlighted in the *Graphics* window.

2. Left-click anywhere in the *Graphics* window and drag the mouse in any direction. The view rotates about the location where the left mouse button is clicked (marked by a small square).
3. Similarly, right-click and drag the mouse in any direction to pan or shift the display.
4. To zoom in and out, middle-click in the *Graphics* window and move the mouse up and down.
5. Rather than use the mouse facilities to adjust the view approximately, click the button at the top of the *Geometry Scene 1* window and select **Views** > **+X +Y +Z** > **Up +Z**, as shown below.



6. Use the mouse buttons to zoom in and position the part so that it is similar to that shown below.

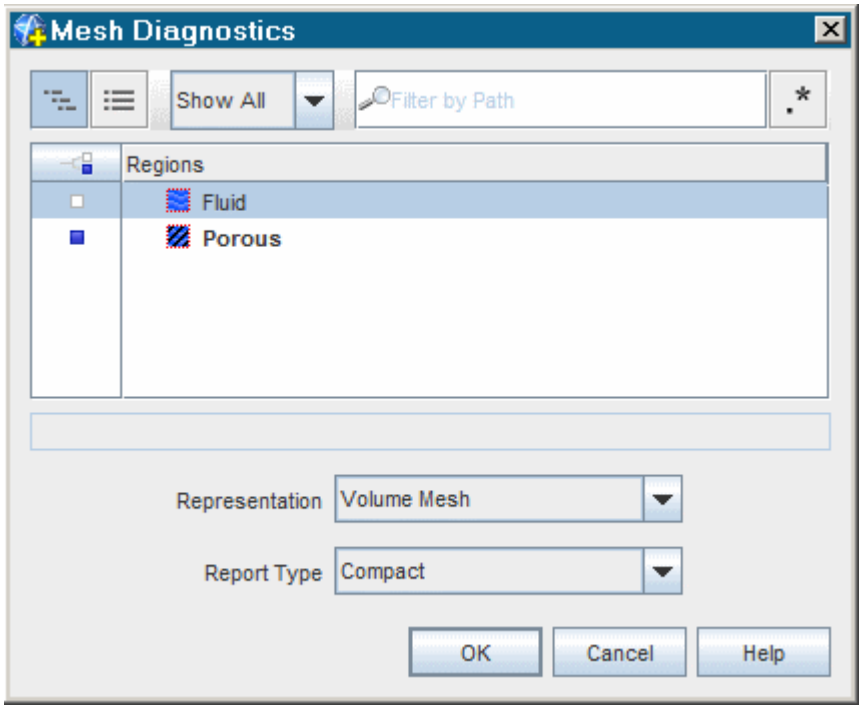


Scaling the Mesh

The original mesh was not built to scale and therefore requires scaling so that the dimensions of the porous region are 0.1 x 0.03 x 0.1 meters.

To determine the current dimensions of the porous region:

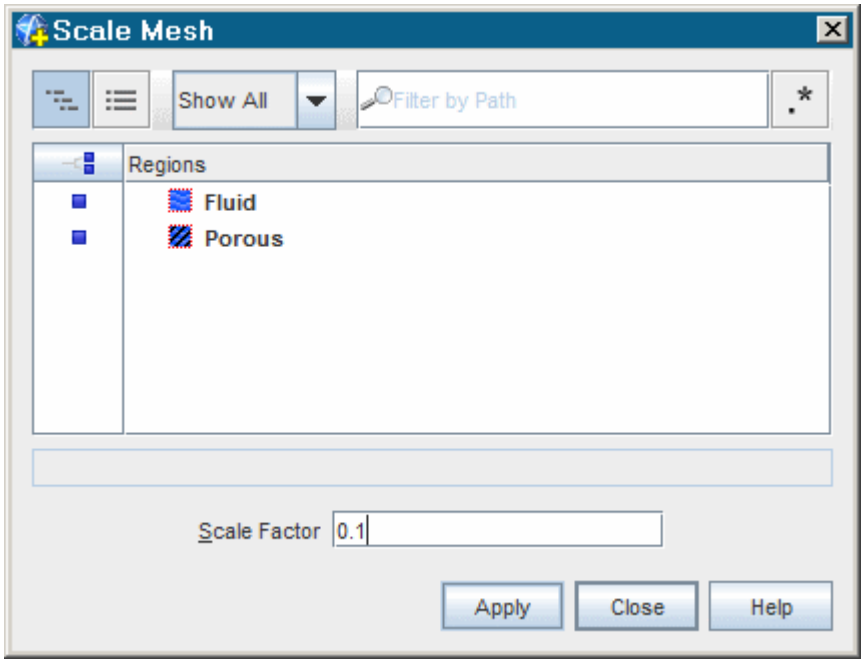
1. Select **Mesh** > **Diagnostics**.
2. In the *Mesh Diagnostics* dialog, deselect **Fluid** and then click **OK**.



The dimensions of the porous region are displayed in the *Output* window as approximately -0.5 to 0.5 in the x-direction, 0.435 to 0.735 in the y-direction and approximately -0.5 to 0.5 in the z-direction.

Without any further action, these values are interpreted by STAR-CCM+ as specified in meter units. The region therefore needs to be scaled by a factor of 0.1 so that the dimensions are correctly interpreted.

3. Select **Mesh** > **Scale Mesh** from the menu bar.
4. In the *Scale Mesh* dialog, select the **Fluid** and **Porous** regions.
5. Enter a value of `0.1` for the scale factor.



6. To scale the domain, click **Apply**.

   The mesh region reduces in size.
7. Click **Close**.
8. Click (**Reset View**) in the visualization toolbar to restore the previous viewing distance for the scaled domain in the display.

To verify that scaling has been applied, you may perform the mesh diagnostics check and review the output values.

Setting Up the Models

Models define the primary variables of the simulation, including pressure, temperature and velocity, and what mathematical formulation is used to generate the solution. In this example, the flow is turbulent and incompressible. Use the **Segregated Flow** model together with the standard K-Epsilon turbulence model.

The default continuum is automatically named **Physics 1** when the mesh is imported.

1. To give the continuum a more appropriate name, rename the **Continua** > **Physics 1** node to `Air`.
2. For the physics continuum **Continua** > **Air**, select the following models, in order:

Group Box	Model
Space	**Three-Dimensional** (Selected automatically)
Material	**Gas**
Flow	**Segregated Flow**
Equation of State	**Constant Density**
Time	**Steady**
Viscous Regime	**Turbulent**
	Reynolds-Averaged Navier-Stokes (Selected automatically)

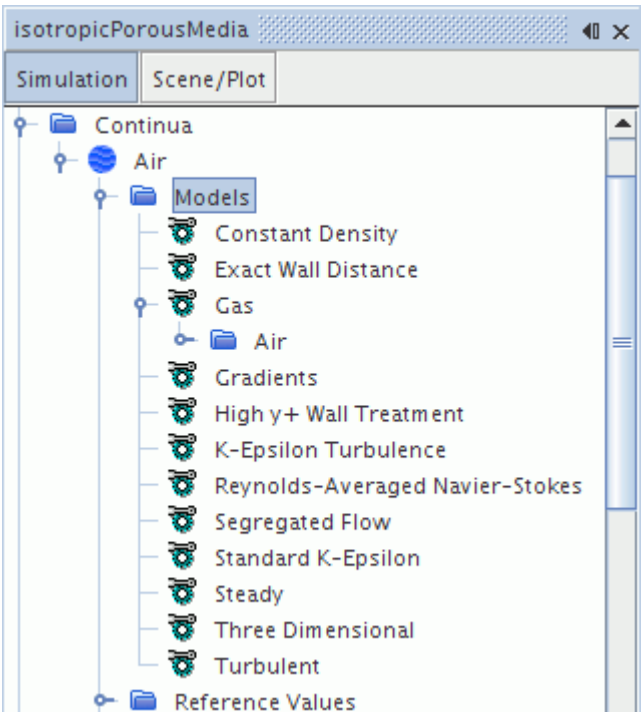
3. Deactivate **Auto-select recommended physics models**.

   This is because the standard K-Epsilon model is used in this case instead of the realizable two-layer K-Epsilon model, which is the STAR-CCM+ default.
4. Select the following models, in order:

Group Box	Model
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**
K-Epsilon Turbulence Models	**Standard K-Epsilon**
	Exact Wall Distance (Selected automatically)
K-Epsilon High y+ Wall Treatment	**High y+ Wall Treatment**

   The color of the **Air** node has turned from gray to blue to indicate that models have been activated.

5. To see the selected models, expand the **Air** > **Models** node.

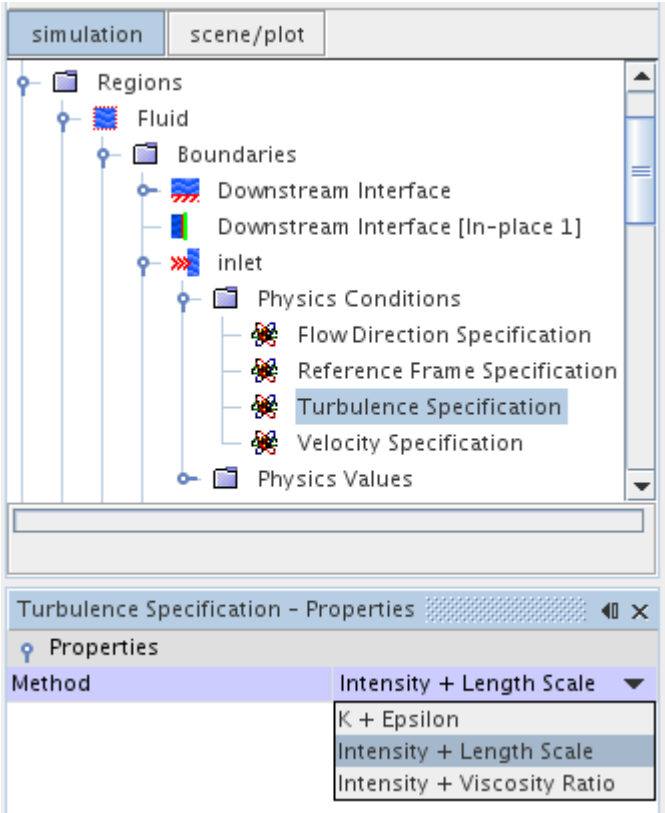


6. Save the simulation.

Setting Boundary Conditions and Values

The default boundary conditions on the wall and outlet boundaries are suitable for this analysis so the only user specifications required are on the inlet boundary. A fixed velocity of 20 m/s and appropriate turbulence parameters need to be specified there.

1. Select the **Regions** > **Fluid** > **Boundaries** > **inlet** > **Physics Conditions** > **Turbulence Specification** node and set the *Method* to **Intensity + Length Scale**.



The boundary values for turbulence can now be set.

2. Edit the **Inlet** > **Physics Values** node and set the following boundary conditions:

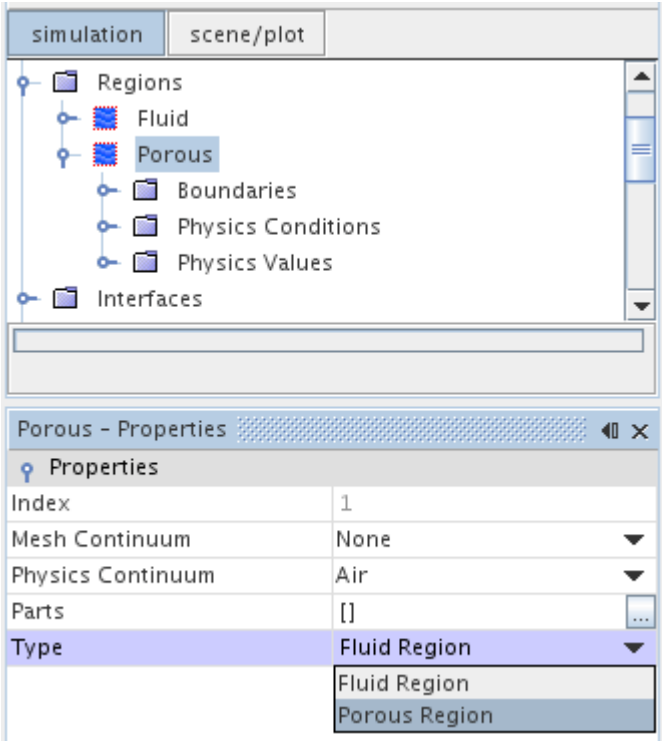
Node	Property	Value
Turbulence Intensity	*Value*	0.05
Turbulence Length Scale	*Value*	0.005 m
Velocity Magnitude	*Value*	20 m/s

Specifying Porosity Coefficients

As discussed in the introduction to this tutorial, the porous region is defined by its inertial and viscous resistance coefficients, which are 25 kg/m^4 and 1500 kg/m^{3}s, respectively. As the porous region is isotropic, these values hold in all directions. The coefficients, along with the turbulence parameters in the porous region, must now be specified.

To start with, the **Porous** region must be defined as a porous medium.

1. Select the **Regions** > **Porous** node and set the *Type* to **Porous Region**.



2. Select the **Porous** > **Physics Conditions** > **Turbulence Specification** node.
3. Change the *Method* to **Intensity + Length Scale.**
4. Select the **Physics Values** > **Porous Inertial Resistance** node and make sure that the *Method* property is set to **Principal Tensor**.

   This is suitable because the constant resistance coefficients correspond to the principal (diagonal) components of the porous resistance tensor.
5. Expand the **Porous Inertial Resistance** > **Principal Tensor** node and multi-select the **XX Component**, **YY Component** and **ZZ Component** nodes.
6. In the *Multiple Objects Properties* window, change the *Value* to `25` kg/m^4.

   The **XX Axis** and **YY Axis** do not need to be changed because the porous resistance components are specified in the directions of the default axes.
7. Open the **Porous Viscous Resistance** > **Principal Tensor** node.
8. Using the same method as for the inertial resistance, specify the three components of the viscous resistance to be 1500 kg/m^3-s.
9. Select the **Turbulence Intensity** node and change the *Value* to `0.1`

   The default turbulent length scale of 0.01 m is suitable so it does not need changing.
10. Save the simulation.

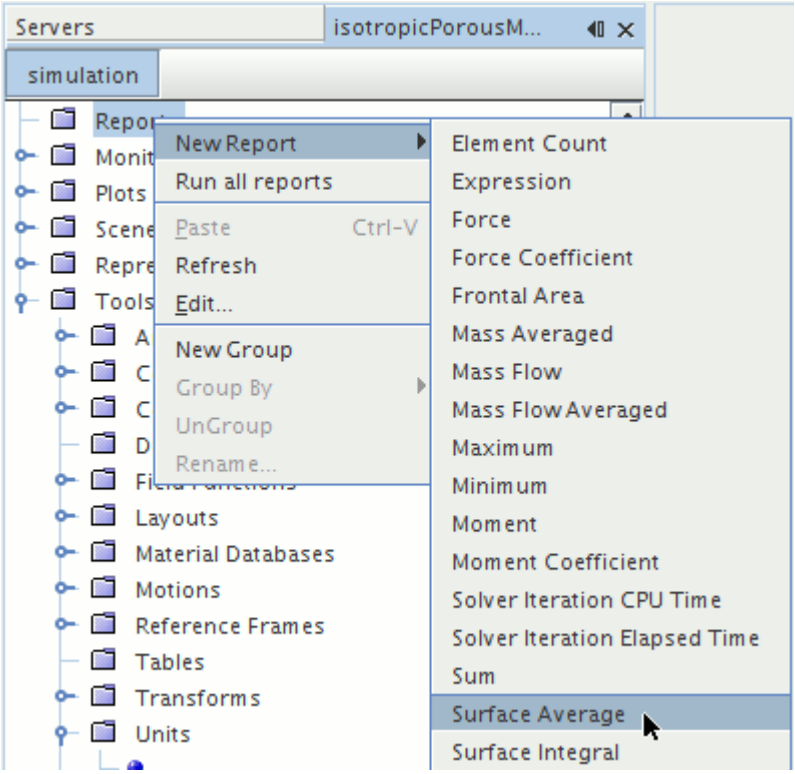
Monitoring the Run Progress

As part of the post-processing operations, make sure that the calculated pressure drop across the porous region is close to the theoretical drop given by Eqn.(2538).

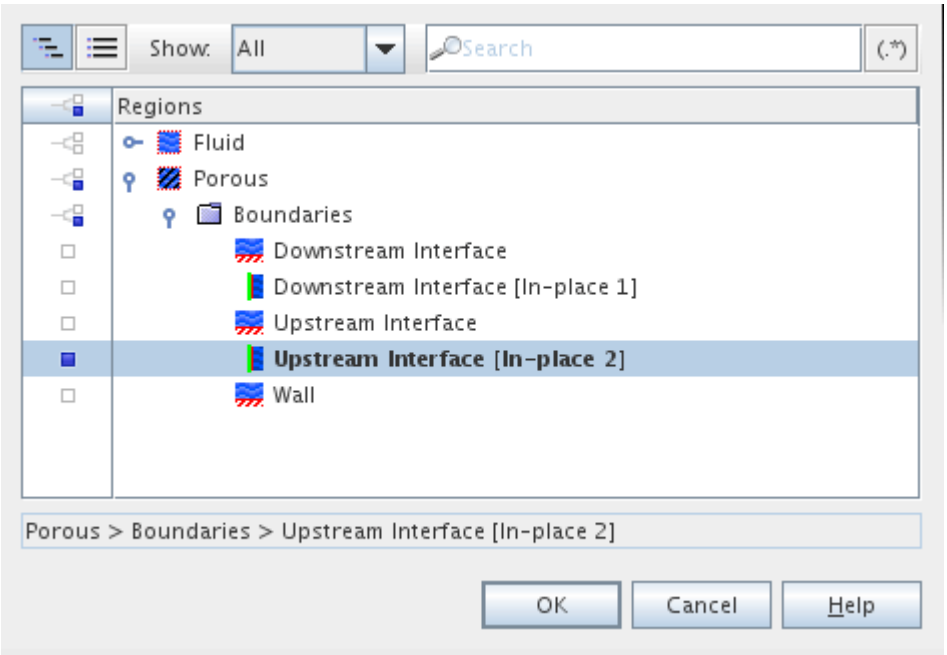
To help you do this, monitor the average pressure on both fluid-porous interfaces, the pressure drop across the porous region and the mass flow rate through the porous region throughout the run. Use these monitors to decide when the solution has converged.

To create an average pressure report on the upstream interface:

1. Right-click the **Reports** node and select **New Report** > **Surface Average**.

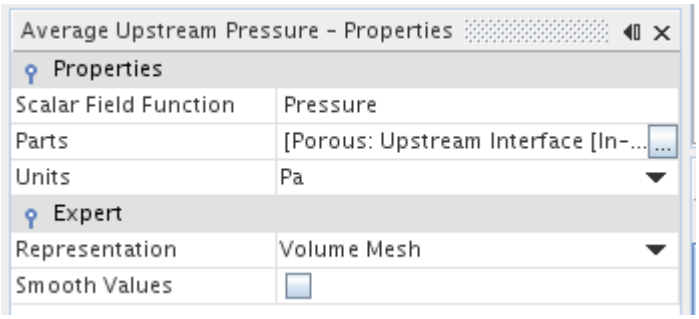


2. Rename the **Surface Average 1** node to `Average Upstream Pressure`.
3. Select the **Average Upstream Pressure** node and set the *Scalar Field Function* to **Pressure**.
4. Edit the *Parts* property and select **Porous** > **Upstream Interface [In-place 2]**.



5. Click **OK**.

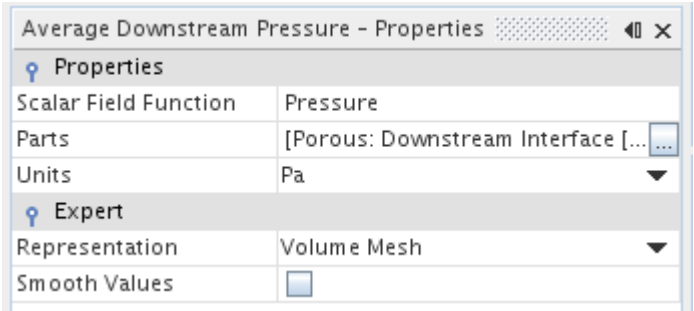
   The *Properties* window for the new report should have the settings that are shown below.



Create a similar report for the downstream pressure:

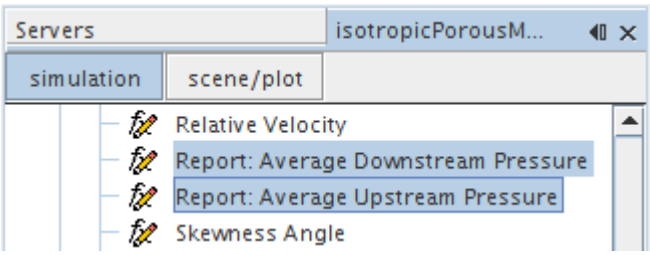
6. Right-click the **Reports** node and select **New Report** > **Surface Average**.
7. Rename the **Surface Average 1** node to `Average Downstream Pressure`.
8. Select the **Average Downstream Pressure** node and set the *Scalar Field Function* to **Pressure** and the *Parts* to **Porous** > **Downstream Interface [In-place 1]**.

The completed *Properties* window should have the settings that are shown below.



Two new field functions corresponding to these reports have been automatically added to the object tree.

9. Open the **Tools** > **Field Functions** node and scroll down to see the new nodes:

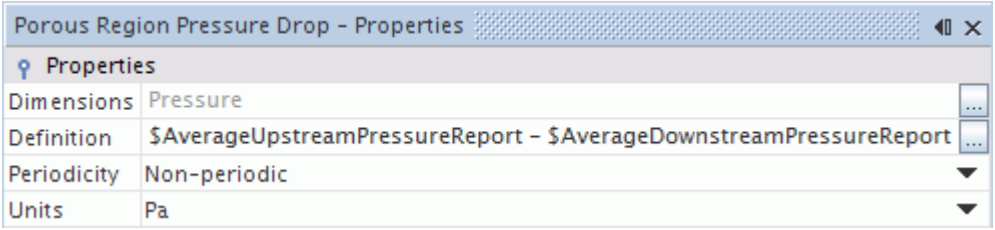


Use these functions to create an expression report for the pressure drop:

10. Right-click the **Reports** node and select **New Report** > **Expression**.
11. Rename the **Expression 1** node to `Porous Region Pressure Drop`.
12. Select the **Porous Region Pressure Drop** node and click ... (**Custom Editor**) to enter the *Definition* property as:

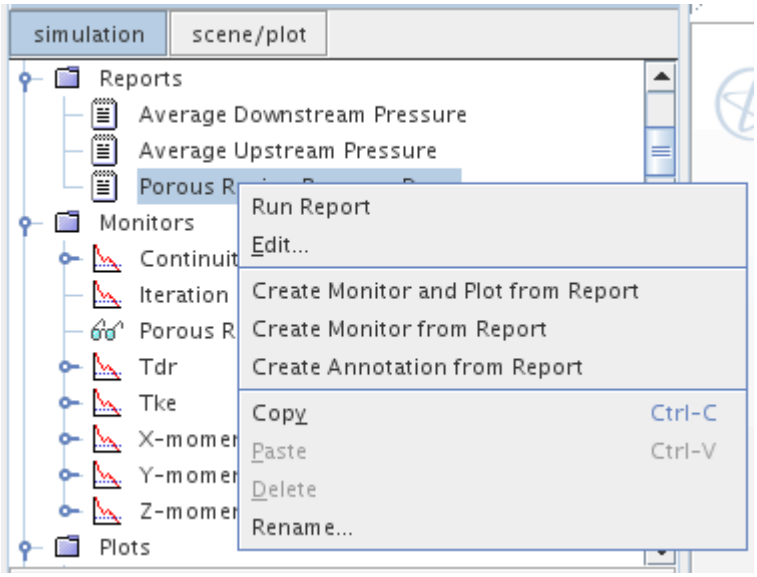
    `$AverageUpstreamPressureReport - $AverageDownstreamPressureReport`
13. Click **OK**.
14. Click ... (**Custom Editor**) for the *Dimensions* property.
15. In the *Porous Region Pressure Drop - Dimensions* dialog, set the **Pressure** dimension to `1`.
16. Click **OK**.

The *Properties* window for the **Porous Region Pressure Drop** report should have the settings that are shown below.

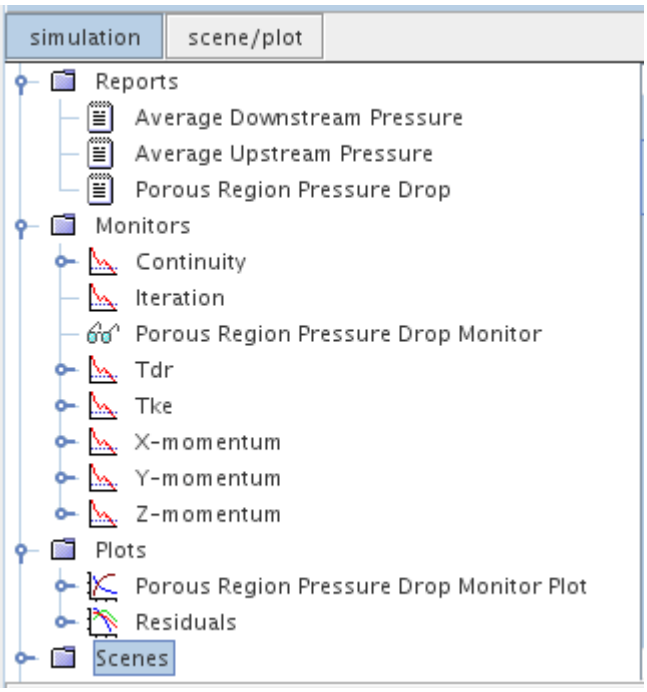


Create a monitor and plot from the **Porous Region Pressure Drop** report.

17. Right-click the **Reports** > **Porous Region Pressure Drop** node and select **Create Monitor and Plot from Report**.



The monitor and plot are added to the object tree:



The empty plot is also displayed.

Create a final report for the mass flow rate through the porous region:

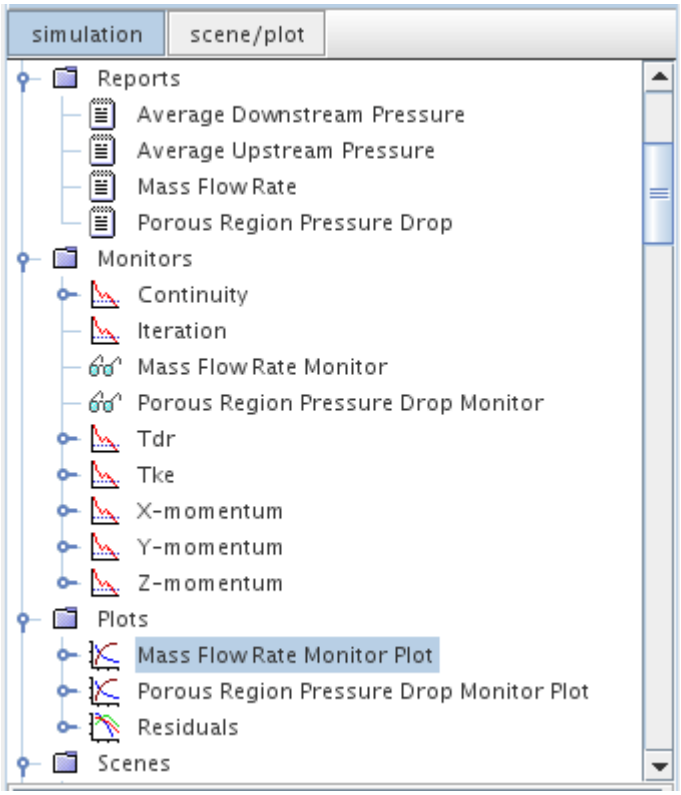
18. Right-click the **Reports** node and select **New Report** > **Mass Flow**
19. Rename the **Mass Flow 1** node to `Mass Flow Rate`.

20. Select the `Mass Flow Rate` node and select **Porous** > **Upstream Interface [In-place 2]** in the object selector.

Create a monitor and plot from the **Mass Flow Rate** report.

21. Right-click the **Reports** > **Mass Flow Rate** node and select **Create Monitor and Plot from Report**.

    The monitor and plot are added to the object tree:

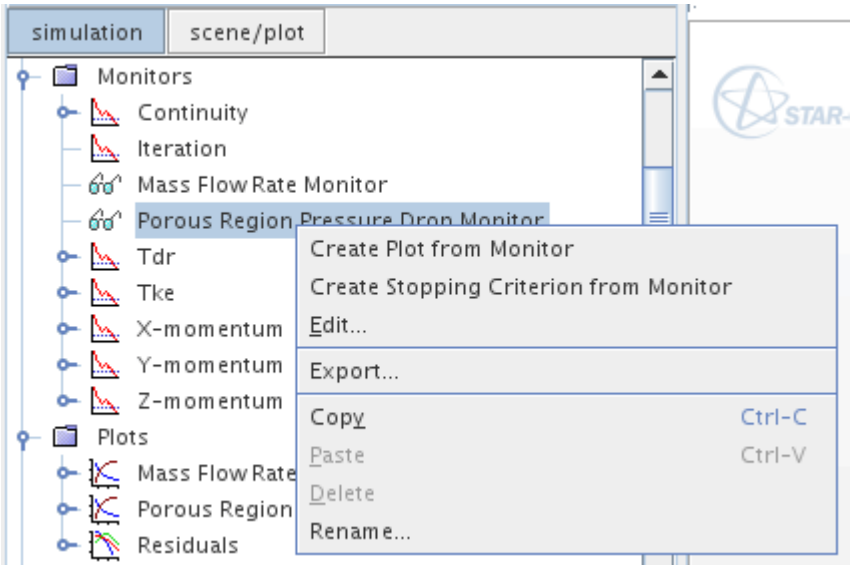


    The empty plot is also displayed.

Setting Stopping Criteria

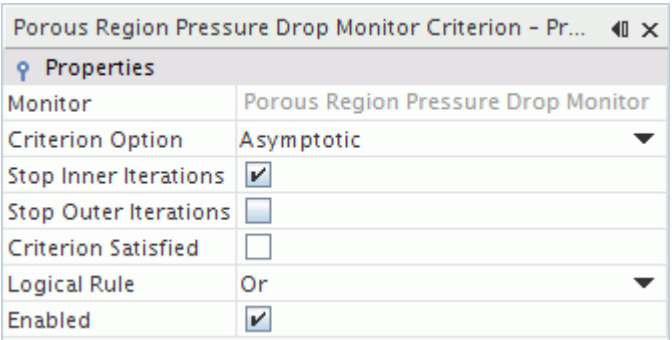
Rather than arranging for the analysis to run for a set number of iterations, it is often useful to specify stopping criteria based on residual values and monitored quantities. For this simulation, define a stopping criterion that checks whether the porous region pressure drop monitor has reached a steady value.

1. Right-click the **Monitors** > **Porous Region Pressure Drop Monitor** node and select **Create Stopping Criterion from Monitor**.

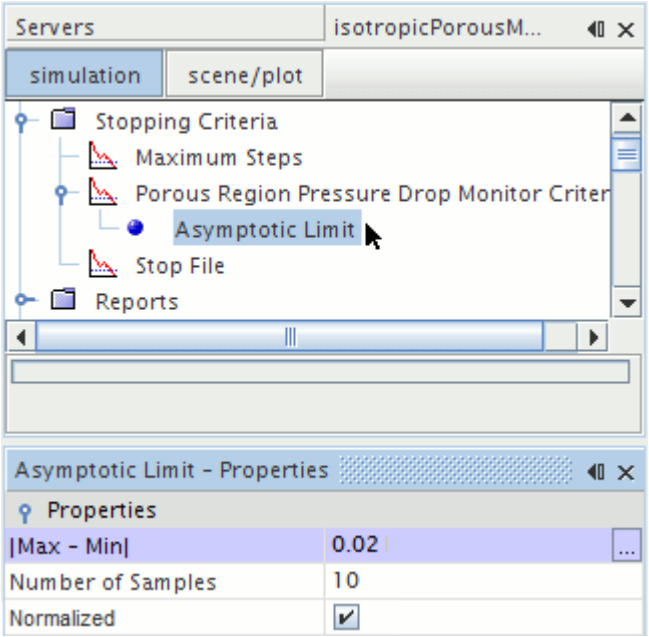


A new node named **Porous Region Pressure Drop Monitor Criterion** appears under the **Stopping Criteria** node.

2. Select the **Porous Region Pressure Drop Monitor Criterion** and change the *Criterion Option* to **Asymptotic**.



3. Select the **Asymptotic Limit** node. Make sure the *Normalized* checkbox is activated and change |*Max - Min*| to `0.02`



The analysis runs until the variation in the **Porous Region Pressure Drop Monitor** has a range of less than 0.02 Pa for 10 successive iterations. Since the pressure on the upstream interface is expected to be of the order of 100 Pa, this should only occur when the solution has practically reached convergence.

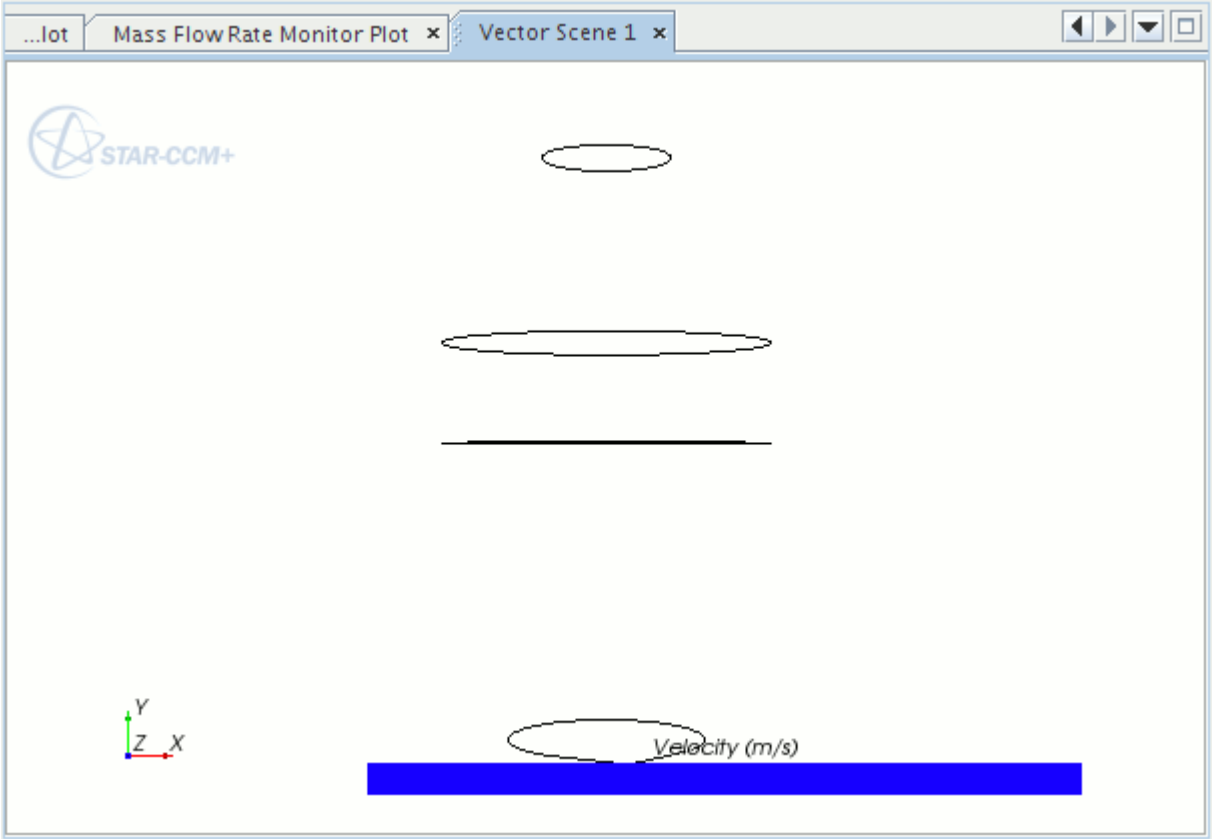
Visualizing the Solution

View the velocity vector field as the solution develops on a plane bisecting the fluid and porous regions.

Start by creating a vector scene:

1. Right-click the **Scenes** node and then select **New Scene** > **Vector**.

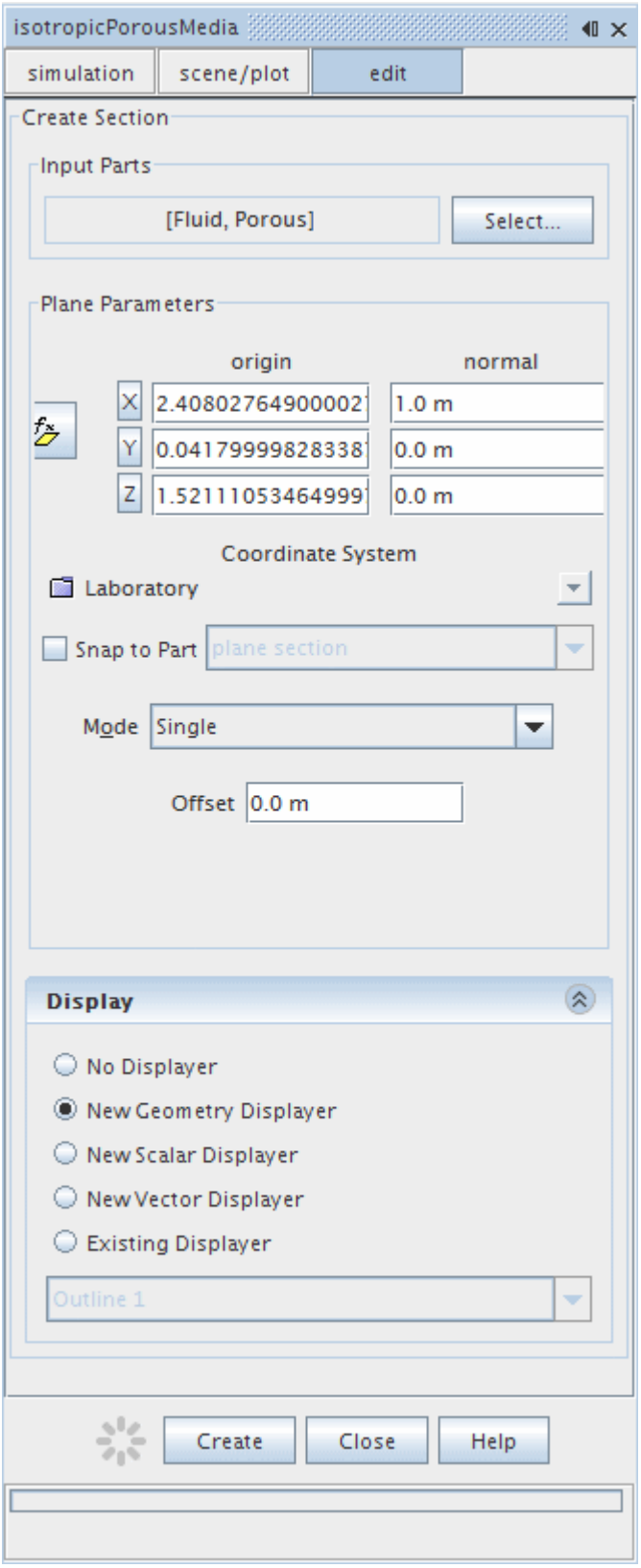
   The *Vector Scene 1* display appears.



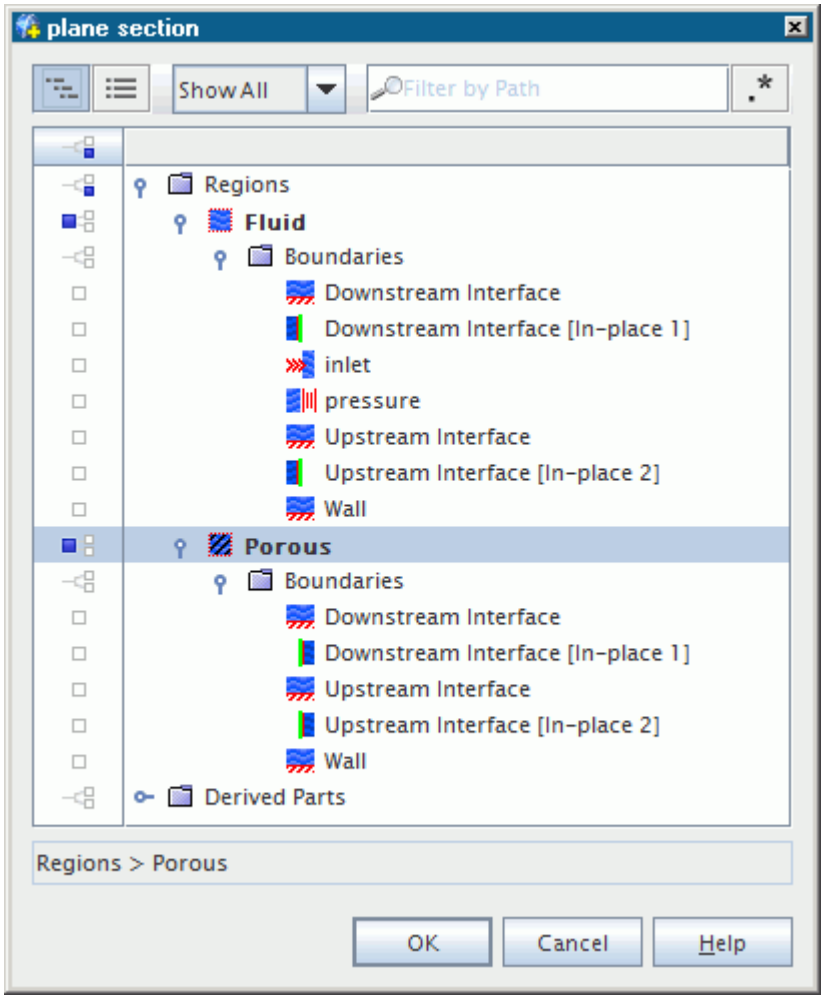
There are currently no parts in *Vector Scene 1*. You therefore need to create a plane section to display the vectors on.

2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane**.

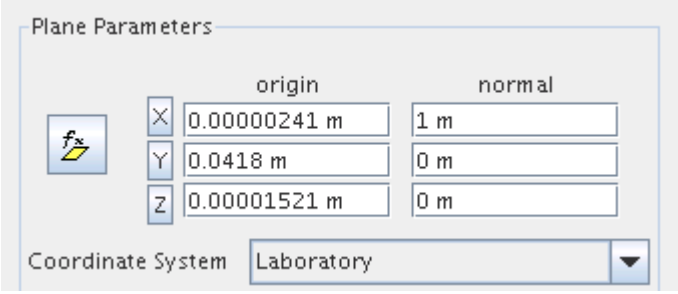
   The *Create Section* dialog appears in the *edit* tab of the simulation window.



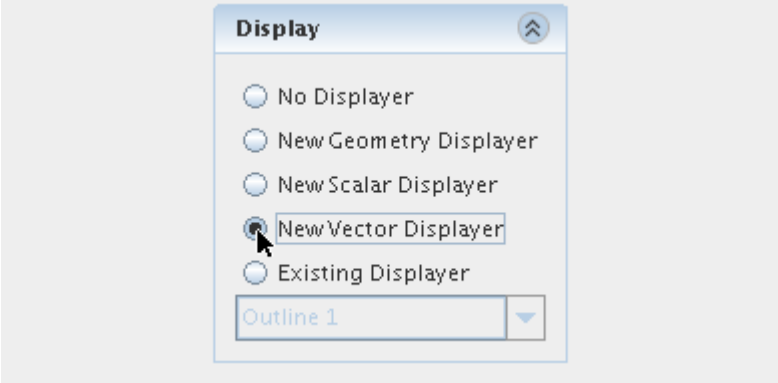
3. Click **Select...** to open the *plane section* dialog. Make sure that the **Fluid** and **Porous** regions (without their boundaries) are selected.



4. Click **OK**.
5. In the *Plane Parameters* box, make sure that the normal vector is set to [`1`, `0`, `0`].



6. In the *Display* box, select **New Vector Displayer**.

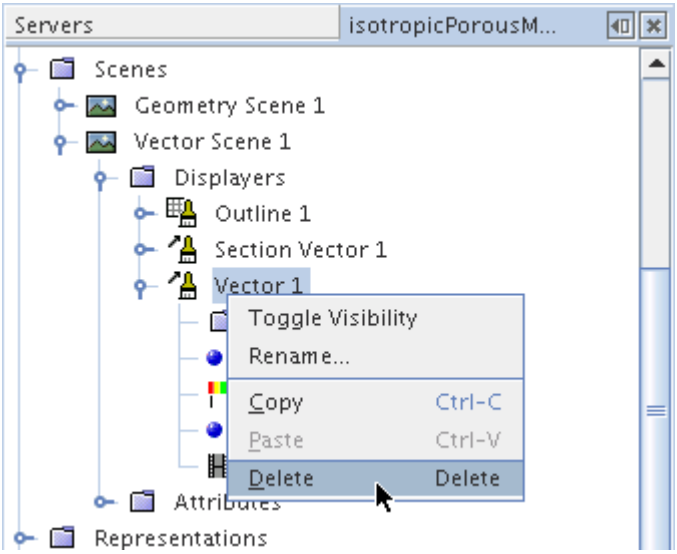


7. Click **Create** and then **Close**.

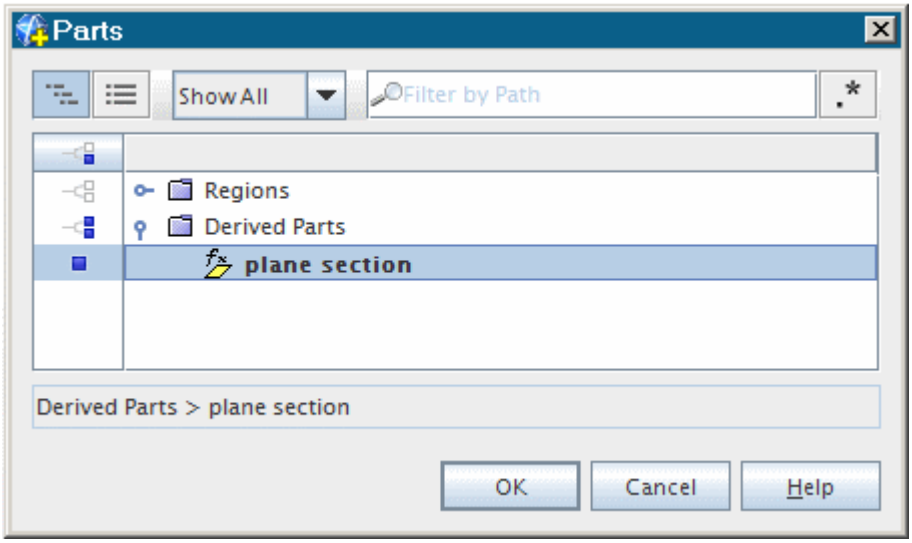
8. Click the button at the top of the *Vector Scene 1* window and select **Views** > **-X** > **Up +Z**.
   The section is now clearly visible as part of the display.

There are currently two vector displayers included in the plot. To remove the unnecessary one:

9. Open the **Vector Scene 1** and **Displayers** nodes, right-click the **Vector 1** node and select **Delete**.



10. Click **Yes** in the *Confirm Object Deletion* dialog.
11. Select the **Outline 1** > **Parts** node and click the right half of the *Parts* property.
12. In the dialog that appears, click the column selection button twice to clear all selections.
13. Select the **Derived Parts** > **plane section** node.



14. Click **OK**.

    The vector scene is now ready.

15. Save the simulation.

Running the Simulation

Run the simulation and observe the Residuals plot, the monitor plots, and the vector scene.

To run the simulation:

1. Click (**Run**) in the top toolbar.

   The Residuals display is created automatically and shows the progress being made by the solver. If necessary, click the *Residuals* tab to bring the Residuals plot into view. An example of a residual plot is shown in a separate part of the User Guide. This example looks different from your residuals, since the plot depends on the models selected.

   The two monitor plots created in the Monitoring the Run Progress section and the vector scene should also be displayed. The tabs at the top of the *Graphics* window make it possible to select any one of the active displays for viewing. They can also be rearranged and multiple scenes can even be shown simultaneously, by dragging the tabs around the edges of the *Graphics* window.

   The monitor plots show large fluctuations in the early iterations which are reduced as the simulation converges to a steady state solution.

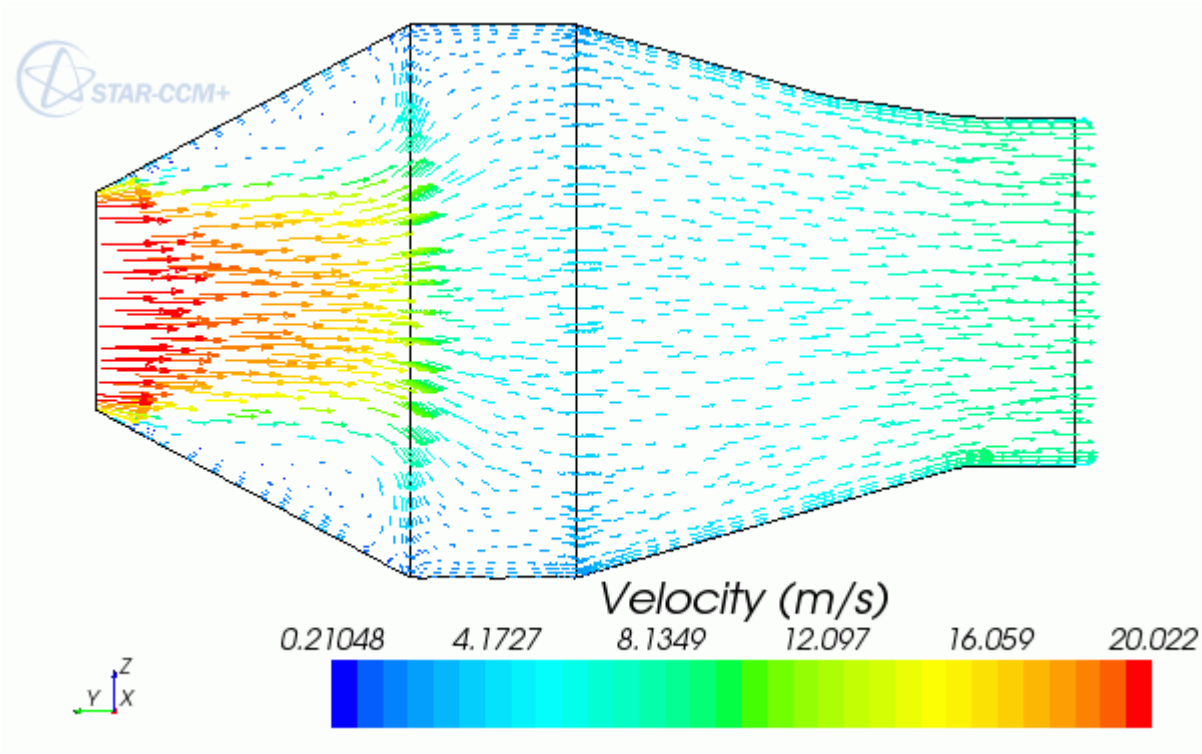
   During the run, it is possible to stop the process by clicking the (**Stop**) button in the toolbar. If you do halt the simulation, it can be continued later by clicking the (**Run**) button. If left alone, the simulation continues until the stopping criterion is satisfied.

2. Save the simulation.

Visualizing Results

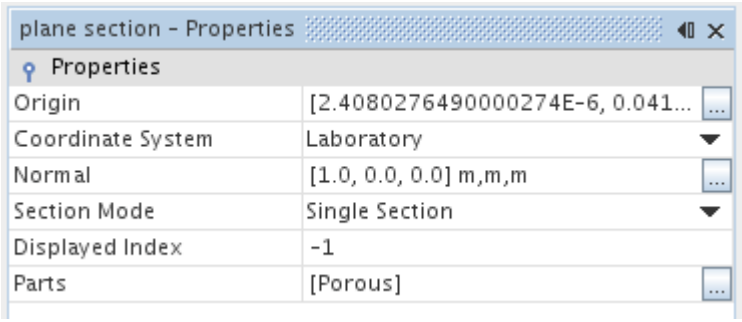
Examine the vector scene in detail.

1. Click the *Vector Scene 1* display tab.
2. Using the click and drag technique, expand the width and the height of the color bar. Drag the color bar to a suitable location.
3. Use the mouse buttons to zoom in and position the part and the color bar so that it is similar to that shown below.



To examine the velocity vectors in just the porous region, remove the fluid region from the plane section.

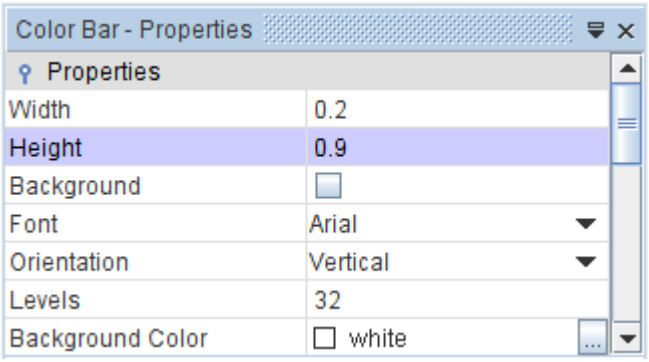
4. Select the **Derived Parts** > **plane section** node then click the *Parts* property and clear the fluid region.



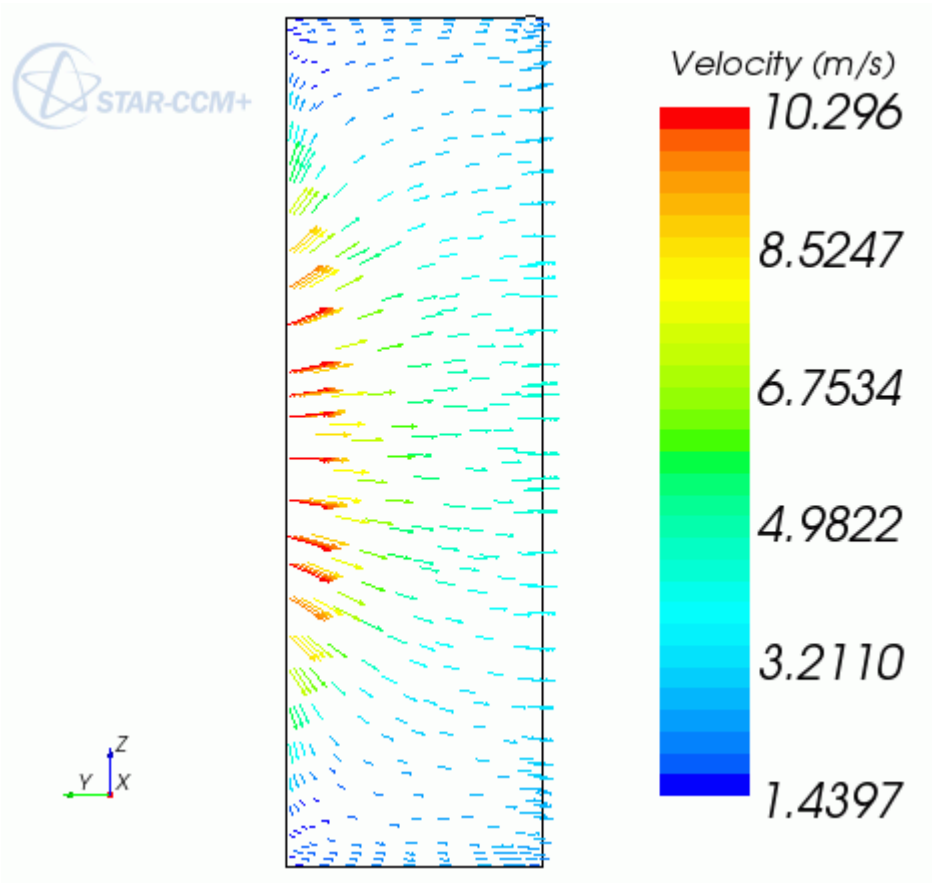
5. Click away from the **plane section** node and the velocity vectors is displayed within the porous region only.

To alter the orientation and location of the legend:

6. Select the **Scenes** > **Vector Scene 1** > **Displayers** > **Section Vector 1** > **Color Bar** node and change the *Orientation* to **Vertical**.
7. Set the *Width* and *Height* properties to `0.2` and `0.9`, respectively.



8. In the *Graphics* window, click the legend and drag it to a suitable location, such as that shown below.



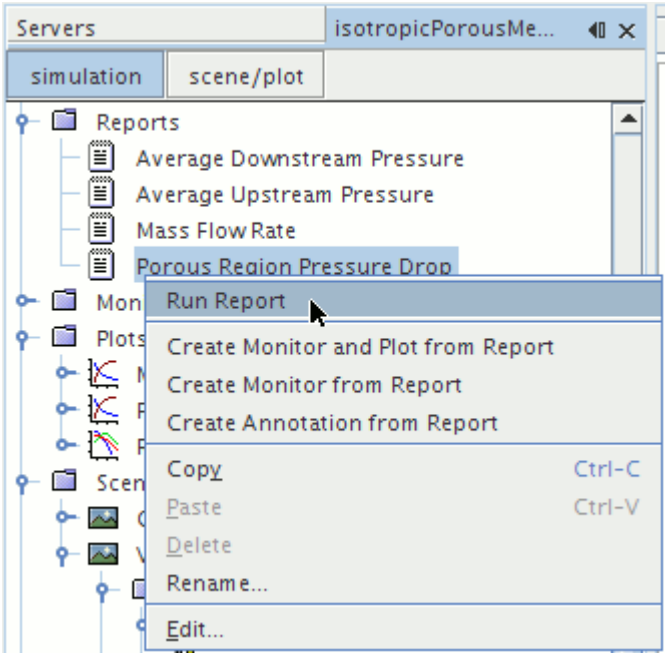
As a result of the isotropy of the porous region, the vectors are not aligned in any particular direction.

9. Save the simulation.

Validating the Results

As discussed at the start of the tutorial, the calculated pressure drop across the porous region can be compared to the theoretical pressure drop derived from the mass flow rate.

1. Right-click the **Reports** > **Porous Region Pressure Drop** node and select **Run Report** from the menu.



The value produced by this report is displayed in the *Output* window as 138.59 Pa.

To determine the theoretical pressure drop:

2. Right-click the **Mass Flow Rate** report node and select **Run Report**.

The mass flow rate through the porous region is 2.91E-02 kg/s. Since the fluid density is 1.18415 kg/m^3 and the cross-sectional area of the porous region 0.00786 m^2, this corresponds to a superficial velocity through the region of 3.12 m/s. Putting this superficial velocity into Eqn.(2538), with L = 0.03 m, gives a pressure drop of 147.75 Pa across the porous region, which means that the calculated pressure drop is within 10% of the theoretical drop. This is acceptable because the theoretical pressure drop formula assumes a constant superficial velocity in the porous medium. In this comparison, you have used an average velocity value based upon the mass flow rate, implying that the theoretical value is only an approximation to the actual pressure drop.

Summary

This tutorial models flow through the catalyst geometry.

This tutorial illustrates the following STAR-CCM+ features:

- Defining physical models.
- Defining material properties required for the selected models.

- Creating region interfaces.
- Defining boundary conditions.
- Specifying porosity coefficients.
- Monitoring the solution progress.
- Setting up stopping criteria.
- Initializing and running the solver until monitored values approach an asymptotic solution.
- Evaluating the solution results using the visualization and reporting facilities of STAR-CCM+.

Porous Resistance: Orthotropic Media

This tutorial solves the same problem as the Porous Resistance: Isotropic Media, except that the fluid in the porous region cannot flow in any direction other than the bulk flow (y-) direction.

This type of orthotropic porosity is specified by assigning large values to the inertial and viscous resistance components in the cross-flow directions (x- and z- directions in this case). The values of the components in the y-direction are the same as in the Porous Resistance: Isotropic Media, that is:

$$P_i = \begin{bmatrix} 10000 & 0 & 0 \\ 0 & 25 & 0 \\ 0 & 0 & 10000 \end{bmatrix} \text{kg/m}^4$$

and

$$P_v = \begin{bmatrix} 100000 & 0 & 0 \\ 0 & 1500 & 0 \\ 0 & 0 & 100000 \end{bmatrix} \text{kg/m}^3\text{s}$$

Contents:

Prerequisites
Loading an Existing Simulation
Renaming the Simulation
Setting Initial Conditions
Changing the Porous Resistance Coefficients
Setting Stopping Criteria
Initializing the Simulation
Running the Simulation
Visualizing the Results
Comparing Results
Summary

Prerequisites

The instructions in the Porous Resistance: Orthotropic Media tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Loading an Existing Simulation

To start this tutorial, you can open one of two versions of the simulation file: the one that you created locally (if applicable), or a previously solved simulation file that is available online.

The following instructions apply to a locally created simulation file.

1. Select **File** > **Load Simulation** from the menu bar.

   The *Load Simulation* dialog appears.
2. Click **Browse** and locate file `isotropicPorousMedia.sim`, created in the Porous Resistance: Isotropic Media tutorial.
3. Select this file and click **Open** in the file browser.
4. Click **OK** in the *Load Simulation* dialog.

Renaming the Simulation

Rename and save the simulation.

1. Select **File** > **Save As** to create a file for the new simulation.
2. In the *Save* dialog, navigate to the directory where you want to place the file.
3. Type `orthotropicPorousMedia` in the **File Name** text box.
4. Click **Save**

The name of the simulation displayed in the title bars of the *Explorer*, *Properties*, and *Output* windows changes accordingly.

Setting Initial Conditions

Set the initial velocity as 3 m/s in the bulk flow (y-) direction.

1. Select the **Continua** > **Air** > **Initial Conditions** > **Velocity** node.
2. Enter `0, -3.0, 0` as the *Value* property.

Changing the Porous Resistance Coefficients

The only model parameters that need to be changed for this problem are the XX and ZZ components of the inertial and viscous porous resistance tensors.

1. Select the **Regions** > **Porous** > **Physics Values** > **Porous Inertial Resistance** > **Principal Tensor** > **XX Component** node.
2. Change the *Value* to `10000` kg/m^4.
3. Do the same for the **ZZ Component** of the inertial resistance tensor.

   Leave the **YY Component** at the value of `25` kg/m^4.
4. Using the same method, change both the **XX Component** and the **ZZ Component** of the viscous resistance tensor to `100000` kg/m^{3}s.

Keep the **YY Component** value of `1500` kg/m^{3}s.

5. Save the simulation.

Setting Stopping Criteria

Disable the stopping criterion based on the pressure drop monitor and run the analysis for 250 iterations instead.

1. Select the **Stopping Criteria** > **Porous Region Pressure Drop Monitor Criterion** node and deactivate *Enabled*.
2. Select the **Maximum Steps** node and change the *Maximum Steps* to `250`.
3. Save the simulation.

Initializing the Simulation

Despite the similarity between the two problems, it is preferable in this case to start the analysis from scratch rather than restart from the Porous Resistance: Isotropic Media solution.

1. To clear the existing solution, select **Clear Solution** from the **Solution** menu and click **OK** in the *Clear Solution* dialog.
2. To re-initialize the solution, click (**Initialize Solution**) in the toolbar.
3. Save the simulation.

Running the Simulation

Run the simulation to generate the solution.

1. To run the simulation, click (**Run**) in the top toolbar.

   If you do not see this button, use the **Solution** > **Run** menu item. You could also show the **Solution** toolbar by selecting **Tools** > **Toolbars** > **Solution** and then clicking the toolbar button.

   The tabs at the top of the *Graphics* window make it possible to select any one of the active displays for viewing. You may want to view the Pressure Drop Monitor Plot or the Mass Flow Rate Monitor Plot to observe the solution convergence rate.

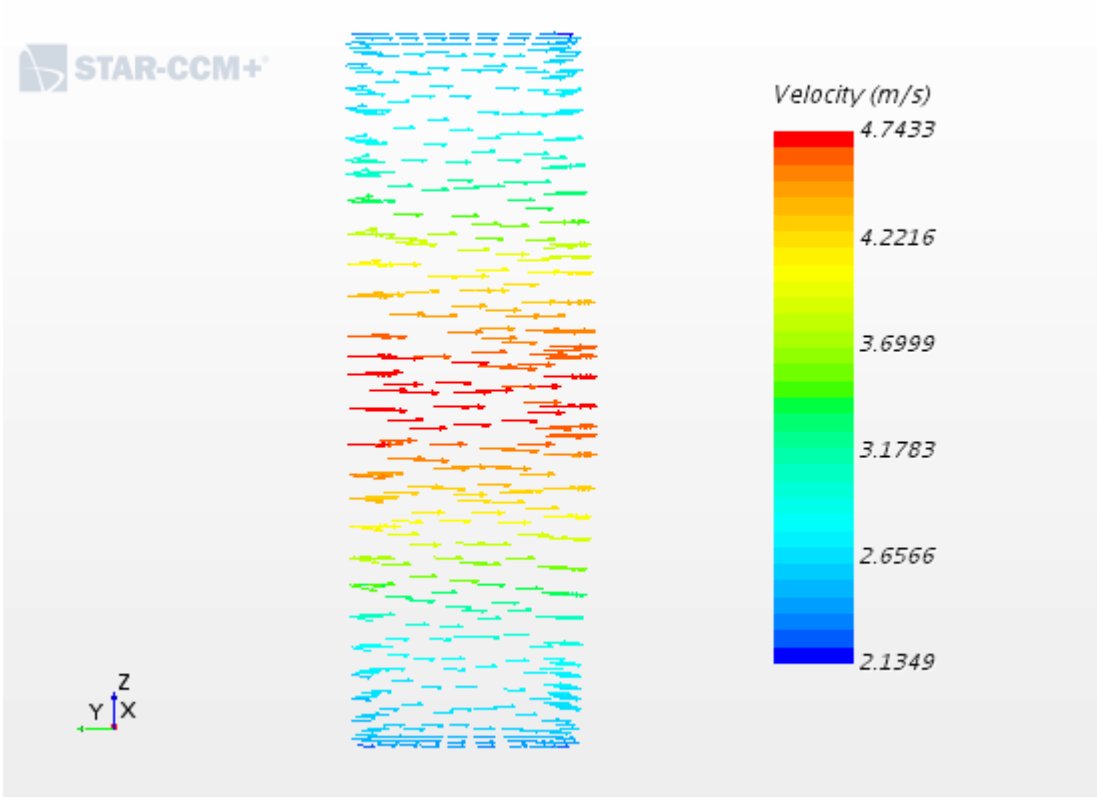
   During the run it is possible to stop the process by clicking the (**Stop**) button in the toolbar. If you do halt the simulation, it can be continued later by clicking the (**Run**) button.

   The Residuals display is created automatically and shows the progress being made by the solver. If necessary, click the *Residuals* tab to bring the Residuals plot into view. An example of a residual plot is shown in a separate part of the User Guide. This example looks different from your residuals, since the plot depends on the models selected.
2. Save the simulation after the run has finished.

Visualizing the Results

Examine the velocity vectors and magnitude map.

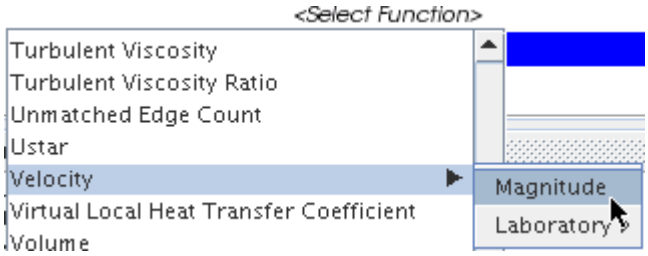
1. At the end of the calculations, open the **Scenes** node and double-click the **Vector Scene 1** node to display the velocity vectors in the porous region.



The flow in this porous medium is essentially unidirectional.

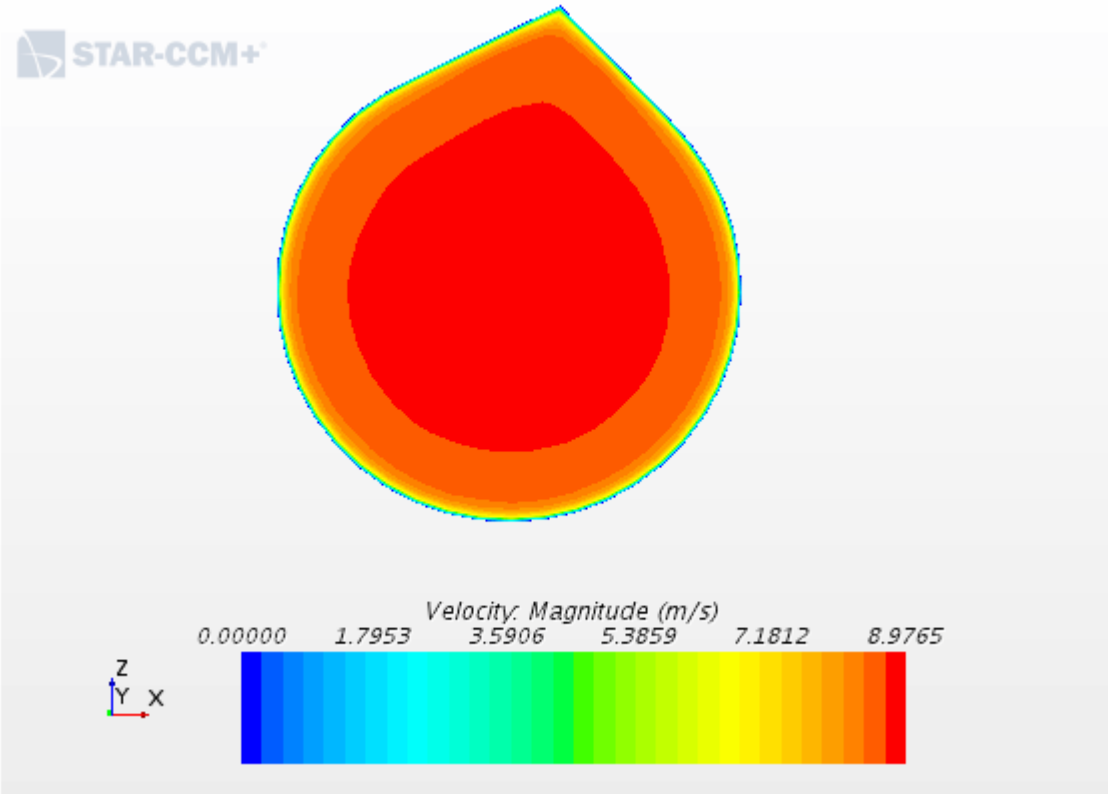
Examine how the porous properties affect the flow downstream of the porous region by comparing velocity magnitude contours at the outlet from the two simulations side-by-side.

2. Right-click the **Scenes** node and select **New Scene** > **Scalar**.
3. Select the **Outline 1** displayer node and deactivate the *Outline* property.
4. Select the **Scalar 1** displayer node and set *Contour Style* to **Smooth Filled**
5. Select the **Scalar 1** > **Parts** node and add **Fluid: pressure** to the Parts list.
6. In the *Graphics* window, right-click the legend color scale and select **Velocity** > **Magnitude** from the list.



7. Click in the toolbar and select **Views** > **-Y** > **Up +Z**.
8. Using the click and drag technique, expand the width and the height of the color bar. Drag the color bar to a suitable location.

   The velocity magnitude contours on the outlet boundary plane are displayed.



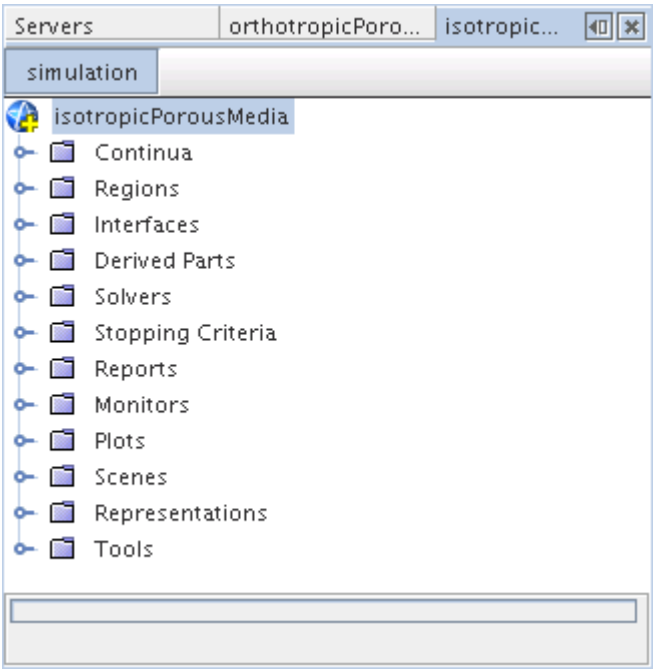
9. Save the simulation.

Comparing Results

Compare the results of this tutorial with those of the Porous Resistance: Isotropic Media tutorial.

Open the simulation file from the Porous Resistance: Isotropic Media and create an identical scalar plot for it.

1. Select **File** > **Load Simulation** and open the isotropicPorousMedia.sim file. Its simulation tree is displayed in the *Explorer* pane.

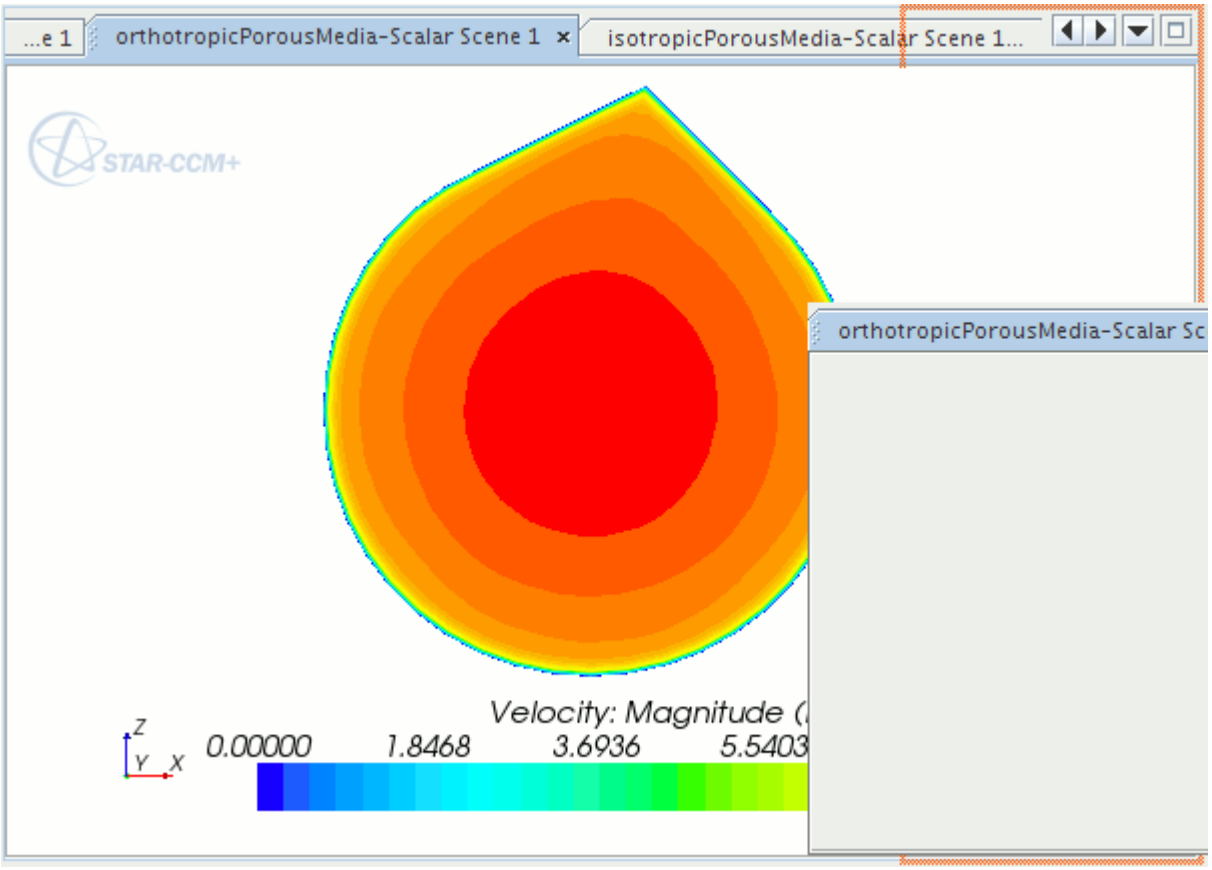


2. Return to the *orthotropicPorousMedia* window, right-click the **Scalar Scene 1** node, and select **Copy**.
3. Open the *isotropicPorousMedia* window, right-click the **Scenes** node and select **Paste** from the menu.

   A scene named *isotropicPorousMedia Scalar Scene 1* is shown in the *Graphics* window.

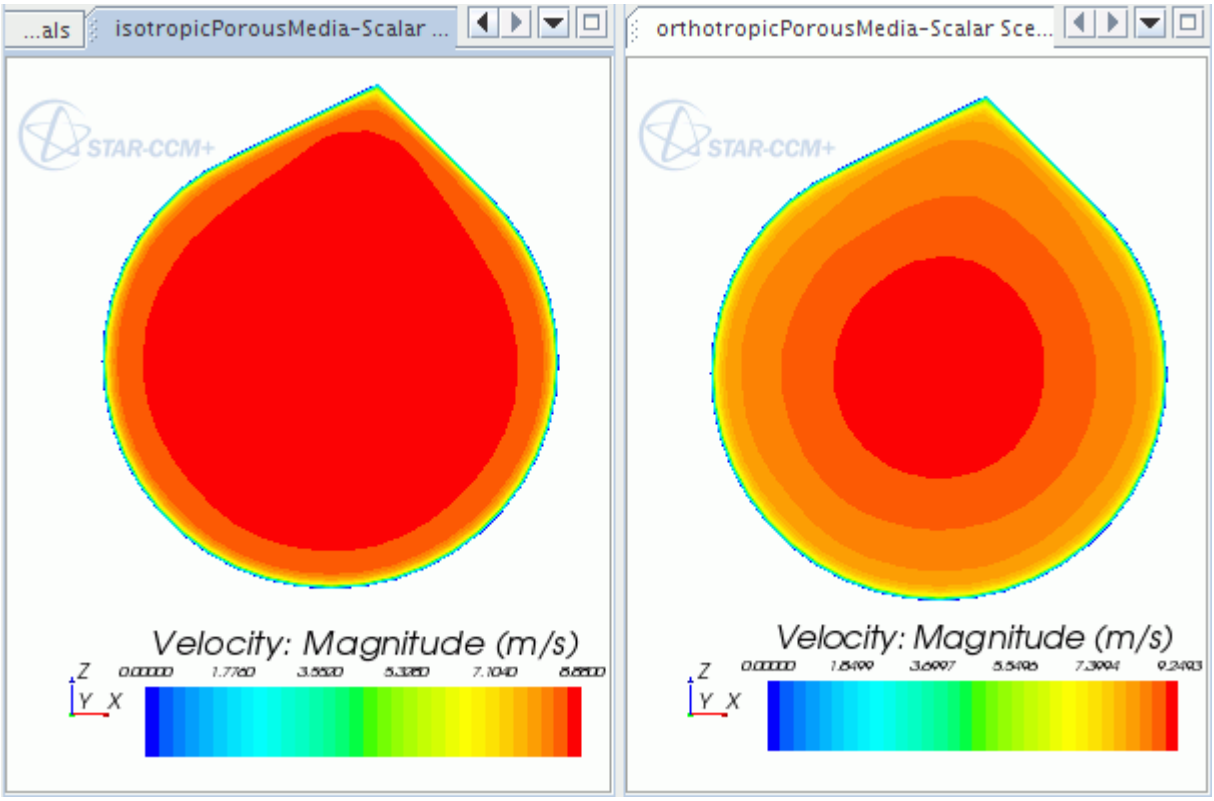
To display the two scenes side-by-side for comparison:

4. Click the *orthotropicPorousMedia Scalar Scene 1* tab at the top of the *Graphics* window and drag it to the right-hand side. Release the mouse button when you see a red box marking out a strip at the edge of the window, as shown below.



Display the two scalar scenes side-by-side.

5. Click the bar between them and drag it to the left so that they have roughly the same width.



6. Save the simulation.

Summary

This tutorial solved the same problem as the Porous Resistance: Isotropic Media tutorial, but for a fluid that can only flow through the porous region in the direction of the bulk flow.

This tutorial illustrates the following STAR-CCM+ features:

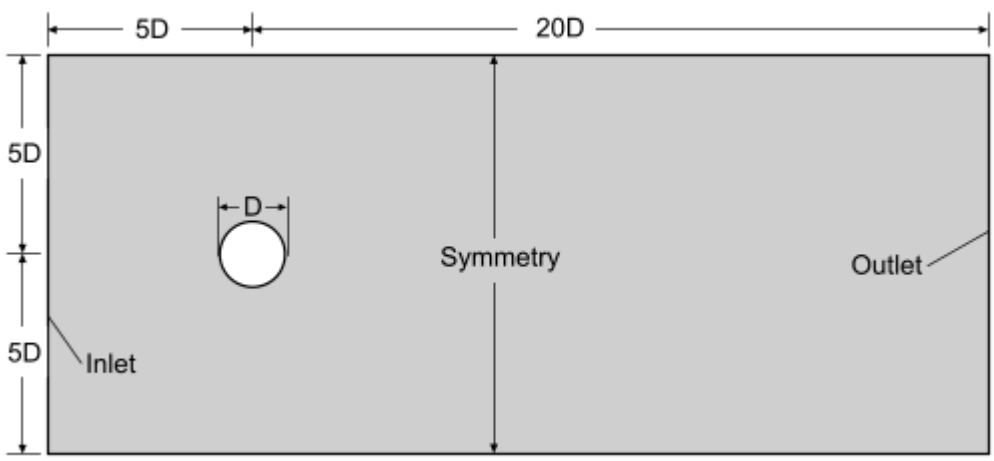
- Loading an existing simulation.
- Renaming and saving the simulation.
- Specifying porous resistance coefficients for an isotropic porous medium.
- Setting the solver parameters and stopping criteria.
- Initializing the simulation.
- Running the solver for a set number of iterations.
- Analyzing results using the built-in visualization facilities.
- Comparing results from different simulations.

Solution Recording and Playback: Vortex Shedding

This tutorial demonstrates how to use the solution recording and playback module for capturing the results of transient phenomena.

The particular scenario being modeled is that of incompressible air flowing over a cylinder with diameter D = 0.01 m. Under the correct conditions, vortices are formed and shed from the cylinder in a regular pattern. The free-stream velocity is 0.15 m/s and the flow is laminar with a Reynolds number (Re) of 75. The predicted Strouhal number and shedding frequency can be determined from a graph of measured lift forces and compared to results obtained by Daily et al.[413].

A 2D volume mesh of a simple cylinder in a fluid domain is provided for this tutorial, the dimensions of which are shown below.



Contents:

Prerequisites
Importing the Volume Mesh
Selecting Physics Models
Modifying Material Properties and Setting Initial Conditions
Setting Boundary Conditions
Creating a Scalar Scene
Preparing the Lift Plot
Modifying Solver Settings
Setting Up Stopping Criteria
Setting Up the Solution History File
Running the Simulation
Visualizing the Results
Validating the Results

Creating a Recorded Solution View
Creating an Animation from the Solution View
Summary
Bibliography

Prerequisites

The instructions in the Solution Recording and Playback: Vortex Shedding tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

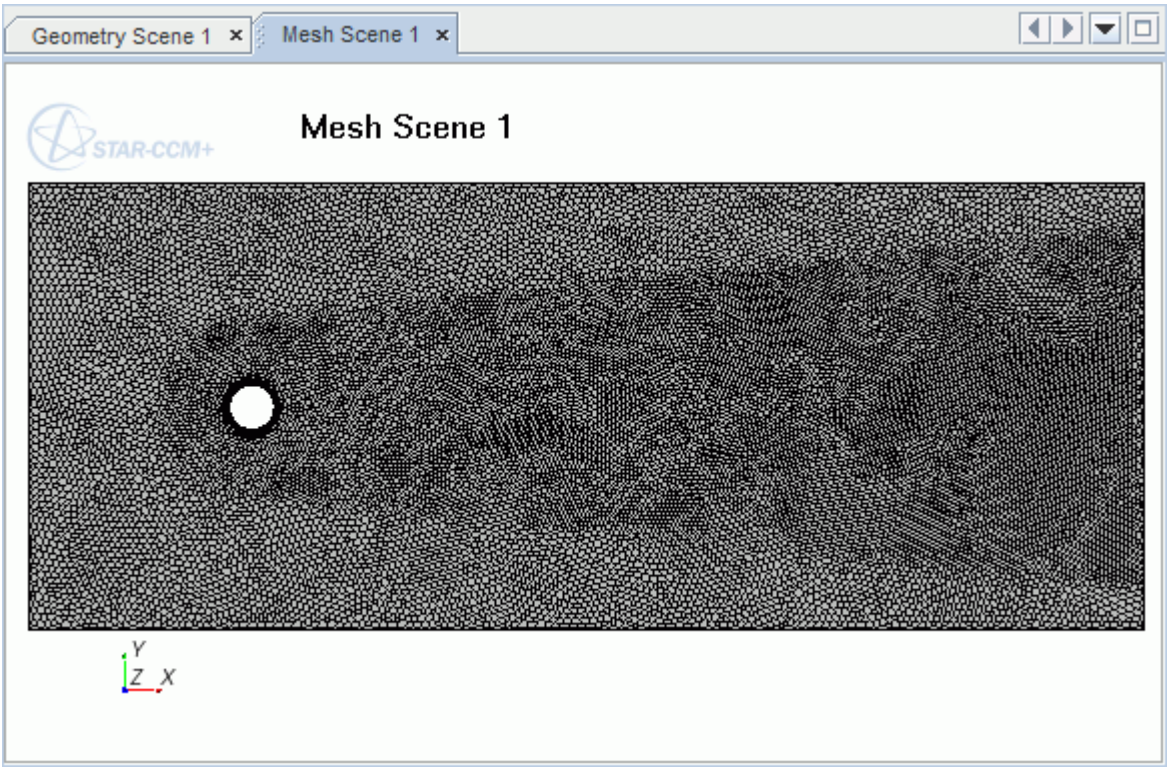
Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Importing the Volume Mesh

To set up the STAR-CCM+ simulation, launch a simulation and import the supplied volume mesh.

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh**.
4. In the *Open* dialog, navigate to `/doc/startutorialsdata/incompressibleFlow/data`.
5. Select `vortexSheddingDomainMesh.ccm` then click **Open**.

   STAR-CCM+ provides feedback on the import process in the *Output* window and a mesh region is created under the **Regions** node.
6. Create a mesh scene and examine the 2D mesh.



7. Save the simulation as `vortexShedding.sim`.

Selecting Physics Models

A physics continuum was added to the object tree when the volume mesh was imported. Select the physics models required to run this case.

The fluid used in this tutorial is air, and the flow is incompressible and laminar. Vortex shedding is a periodic phenomenon and requires the use of a transient solver.

To select the physics models:

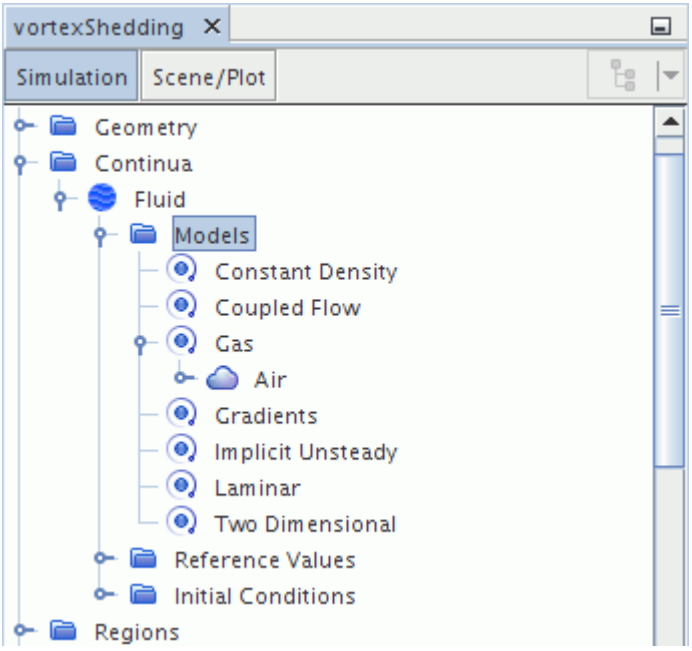
1. Rename the **Continua** > **Physics 1** continuum to `Fluid`.
2. For the physics continuum, **Continua** > **Fluid**, select the following models in order:

Note: If you deactivated the default option *Run Mesh Diagnostics After Importing Mesh* (see STAR-CCM+ Default Options in the User Guide), you must first activate the **Two Dimensional** model in the *Space* group box.

Group Box	Model
Time	**Implicit Unsteady**
Material	**Gas**
Flow	**Coupled Flow**
Equation of State	**Constant Density**

Group Box	Model
Viscous Regime	**Laminar**

3. Click **Close**.
4. To review the models, open the **Fluid** > **Models** node.



Modifying Material Properties and Setting Initial Conditions

Modify the material properties of air, so that the correct Reynolds number is obtained.

1. Within the **Fluid** continuum, select the **Models** > **Gas** > **Air** > **Material Properties** > **Density** > **Constant** node and set Value to `1 kg/m^3`.
2. Select the **Dynamic Viscosity** > **Constant** node and set *Value* to `2.0E-5 Pa-s`.

Set the initial conditions so that the simulation begins with the fluid in a state of motion.

3. Select the **Fluid** > **Initial Conditions** > **Velocity** node and set *Value* to `[0.15, 0.0] m/s`.

Setting Boundary Conditions

Set the required velocity at the inlet boundary.

To set the boundary conditions:

1. Edit the **Regions** > **Fluid_Domain** > **Boundaries** > **Inlet** node.
2. Set the following properties:

Node	Property	Setting
Physics Conditions > **Velocity Specification**	*Method*	**Components**
Physics Values > **Velocity**	*Value*	`[0.15, 0.0] m/s`

Creating a Scalar Scene

Create a scalar scene displaying vorticity to visualize the solution while the simulation is running.

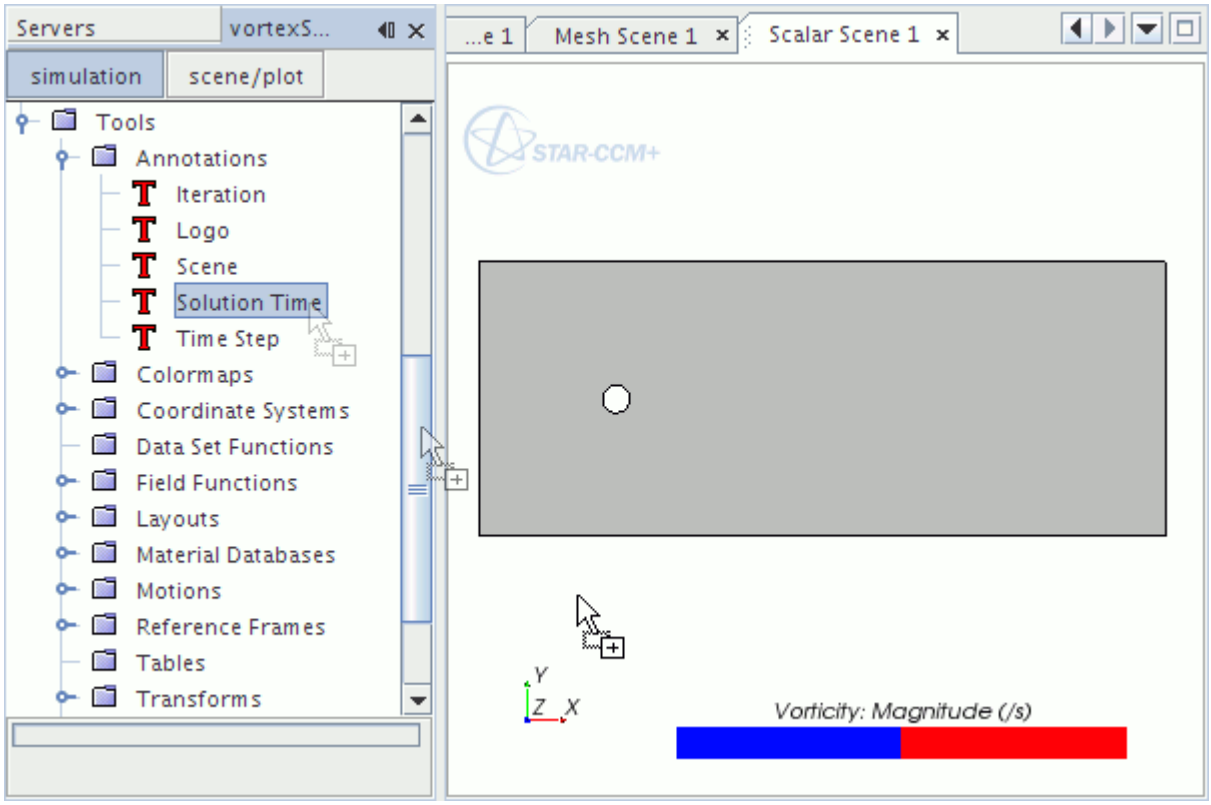
1. Create a scalar scene.
2. Click the **Scene/Plot** button located above the object tree.
3. Select the **Displayers** > **Scalar 1** > **Scalar Field** node and set Function to **Vorticity** > **Magnitude**.
4. In the same window, set the following properties:

Property	Value
Min	`0`
Max	`28`
Clip	`Off`
Auto Range	`Off`

5. Select the **Displayers** > **Scalar 1** node and set *Contour Style* to **Smooth Filled**.
6. Click the **Simulation** button to return to the STAR-CCM+ simulation object tree.

Add an annotation displaying solution time to the scene:

7. Expand the **Tools** > **Annotations** node and drag the **Solution Time** node into the scene.



The Solution Time annotation is added to the bottom left of the scene.

Preparing the Lift Plot

Monitor the lift that the cylinder wall is experiencing to determine the period of oscillation for the vortex shedding.

First create a report:

1. Right-click the **Reports** node and select **New Report** > **Force Coefficient**.
2. Rename the new report to `Coefficient of Lift`.
3. Select the **Coefficient of Lift** node and set the following:

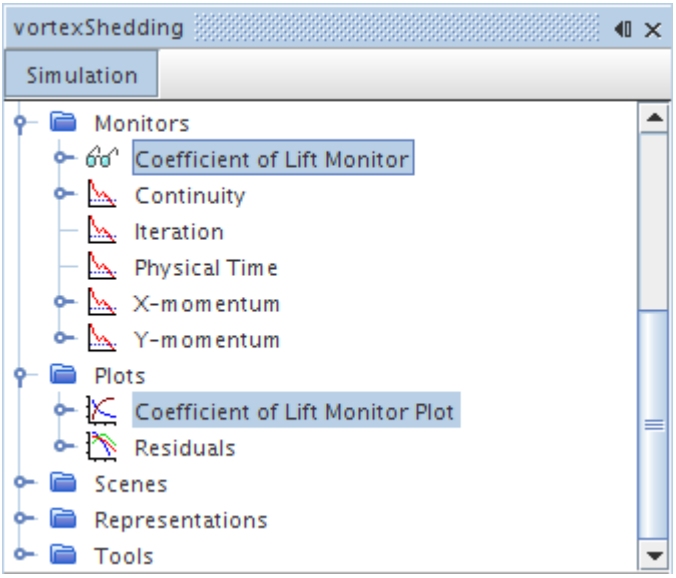
Property	Setting
Reference Velocity	`0.15 m/s`
Reference Area	`0.01 m^2`
Force Option	**Pressure**
Direction	`[0.0, 1.0, 0.0]`

4. Set *Parts* to **Regions** > **Fluid_Domain** > **Cylinder**.

Create a monitor and plot from this report.

5. Right-click the **Coefficient of Lift** node and select **Create Monitor and Plot from Report**.

A new monitor and plot is added to the **Monitors** and **Plots** nodes respectively.



Modifying Solver Settings

Modify the solver settings to more appropriate values.

1. Select the **Solvers** > **Implicit Unsteady** node and set the following properties:

Property	Setting
Time-Step	`0.02 s`
Temporal Discretization	**2nd-order**

2. Select the **Solvers** > **Coupled Implicit** node and set Courant Number to `100`.

Setting Up Stopping Criteria

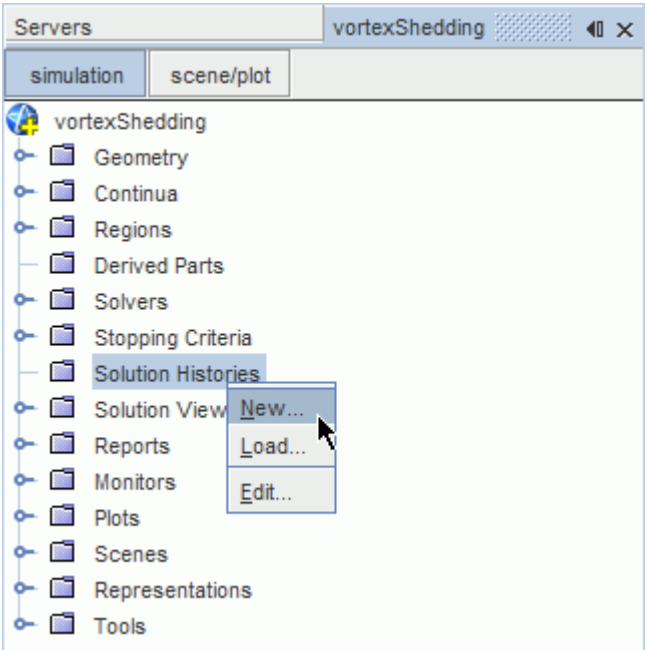
Adjust the number of inner iterations for each time step and extend the maximum time for the solver to run.

1. Select the **Stopping Criteria** > **Maximum Inner Iterations** node and set *Maximum Inner Iterations* to `15`.
2. Select the **Stopping Criteria** > **Maximum Physical Time** node and set *Maximum Physical Time* to `8 s`.
3. Save the simulation.

Setting Up the Solution History File

Create a solution history, `simh`, file and use it to store selected solution data at specified time intervals.

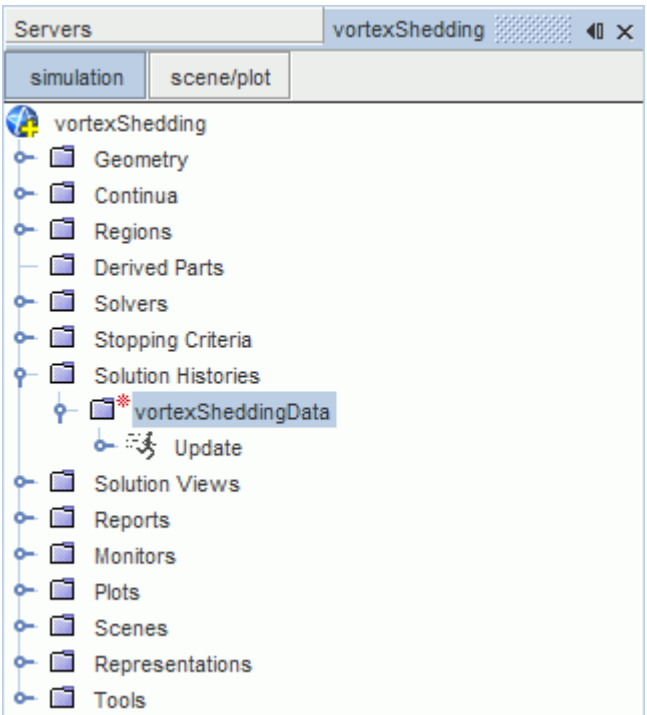
1. Right-click the **Solution Histories** node and select **New...**.



The *Save* dialog appears.

2. Enter `vortexSheddingData.simh` as the name of the solution history file and click **Save**.

   A new node containing the name of the solution history file is added to the object tree below the **Solution Histories** node.



The red asterisk next to this node means that data is actively written to the file when the simulation runs.

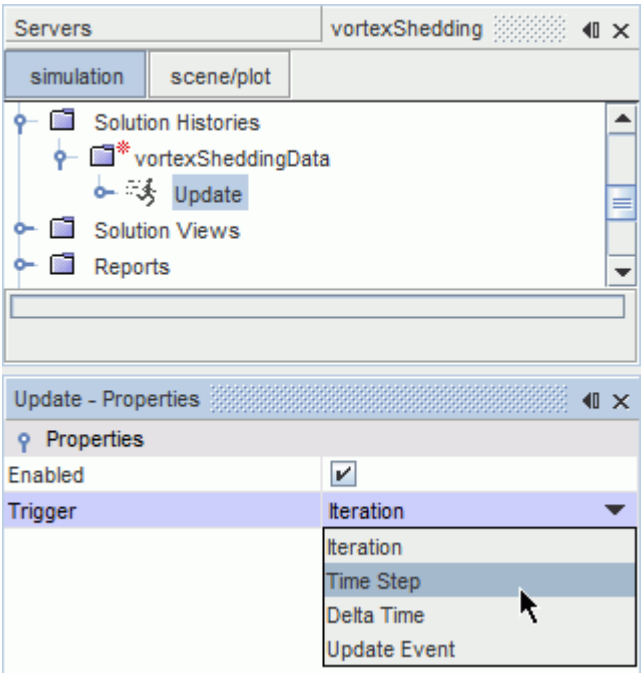
Choose what data to save to the solution history file. As this is a vortex shedding case it is appropriate to store results for pressure, velocity and vorticity.

3. Select the **Solution Histories** > **vortexSheddingData** node and set the following properties:

Property	Setting
Functions	**Pressure**
	Velocity > **Velocity**
	Vorticity > **Vorticity**

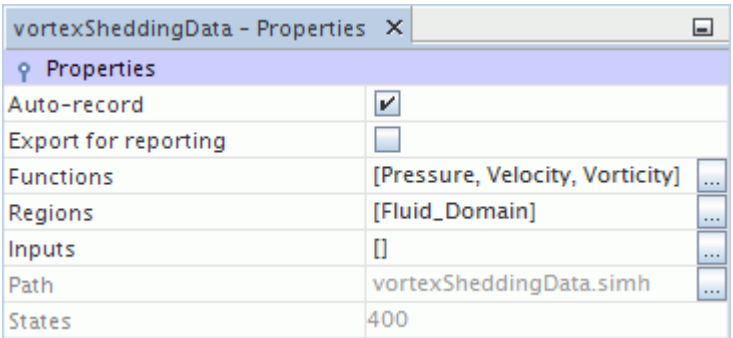
Set the frequency with which the selected data is written to the solution history file:

4. Select the **Solution Histories** > **vortexSheddingData** > **Update** node and set *Trigger* to **Time Step**.



5. Select the **Update** > **Time-Step Frequency** node and make sure that *Frequency* is set to `1`.
6. Select the **Solution Histories** > **vortexSheddingData** node.

   A summary of the properties for the solution history file is shown.



The *Auto-record* property records the data to the solution history file at the required intervals. If you do not want to record the data when the solver is running, deactivate this setting. The *Regions* property is automatically populated with all regions in the simulation. In cases with multiple regions, you may remove regions by clicking to the right of the *Regions* property and clearing the checkbox next to the region you want to remove. The *Path* property displays the relative path to the simulation history file. The *States* property displays the number of saved states stored in the selected solution history file; currently this is displaying 0 as the solver has not yet run.

7. Save the simulation.

Running the Simulation

The simulation is ready to run.

To start the analysis:

1. In the **Solution** toolbar, click (**Run**).

   While the simulation is running you can click the tabs at the top of the **Graphics** window to switch between the plot and the scene. Also, the **Solution Views** > **Current Solution** node displays the current *Iteration*, *Time Step*, and *Solution Time*. The selected data is saved to the solution history file at every time step.

   The Residuals display is created automatically and shows the progress of the solvers. The simulation continues until the physical time of the simulation reaches 8 seconds.

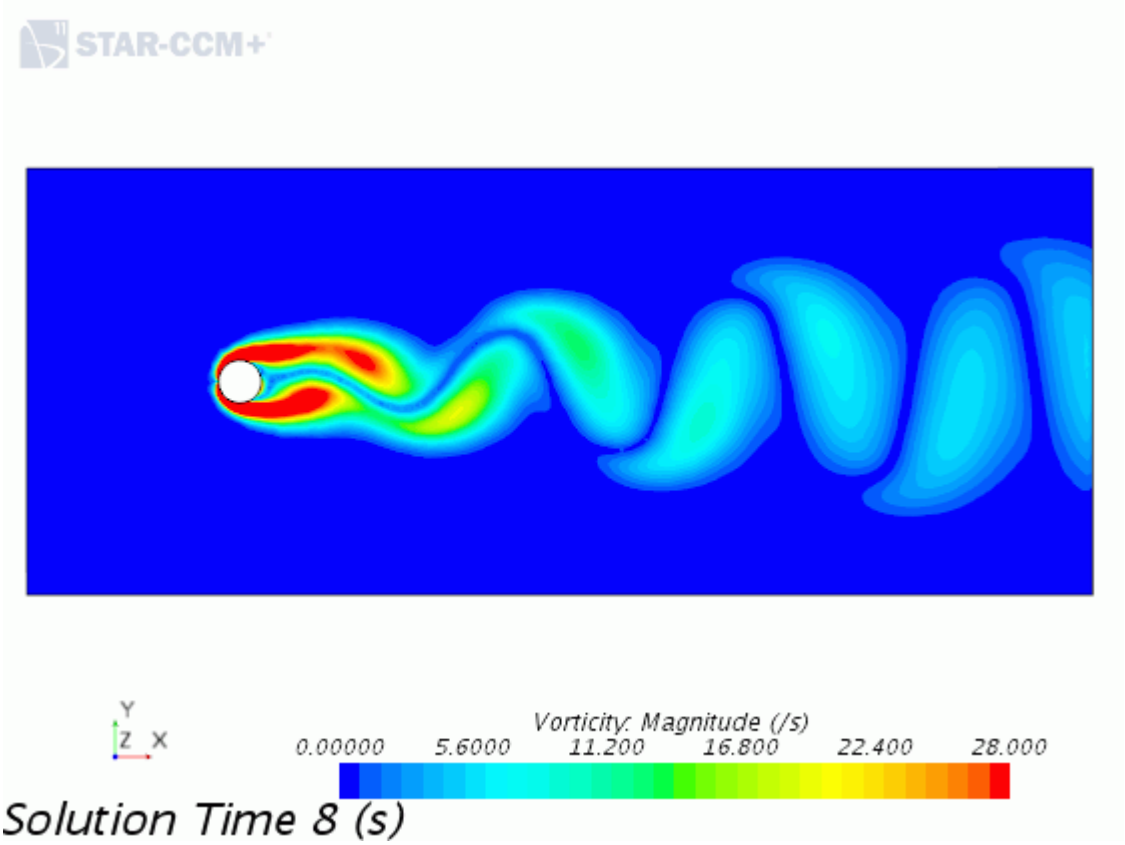
While the simulation is running,

2. While the simulation is running, select the **Scalar Scene 1** tab at the top of the **Graphics** window to visualize the solution.
3. When the simulation has finished, save the simulation.

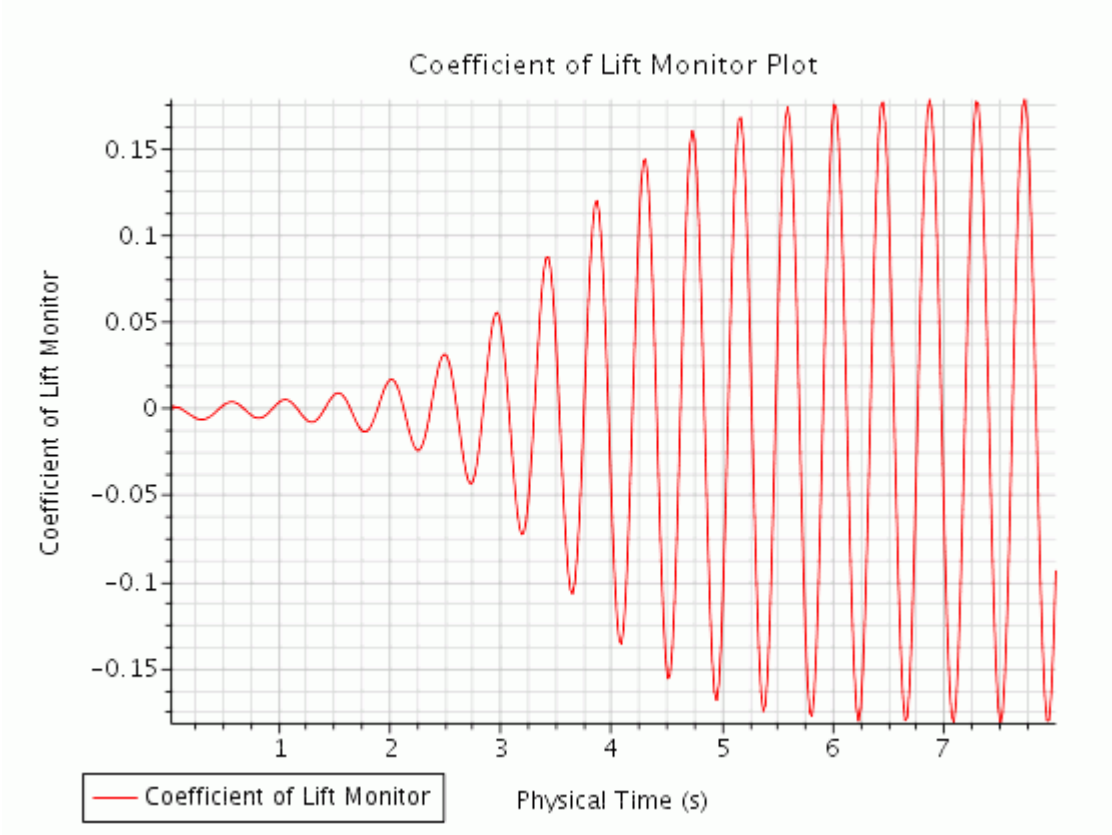
Visualizing the Results

Examine the Vorticity scene and the Coefficient of Lift plot.

The scalar scene after 8 s is shown below.



The Coefficient of Lift monitor plot is shown below.



Validating the Results

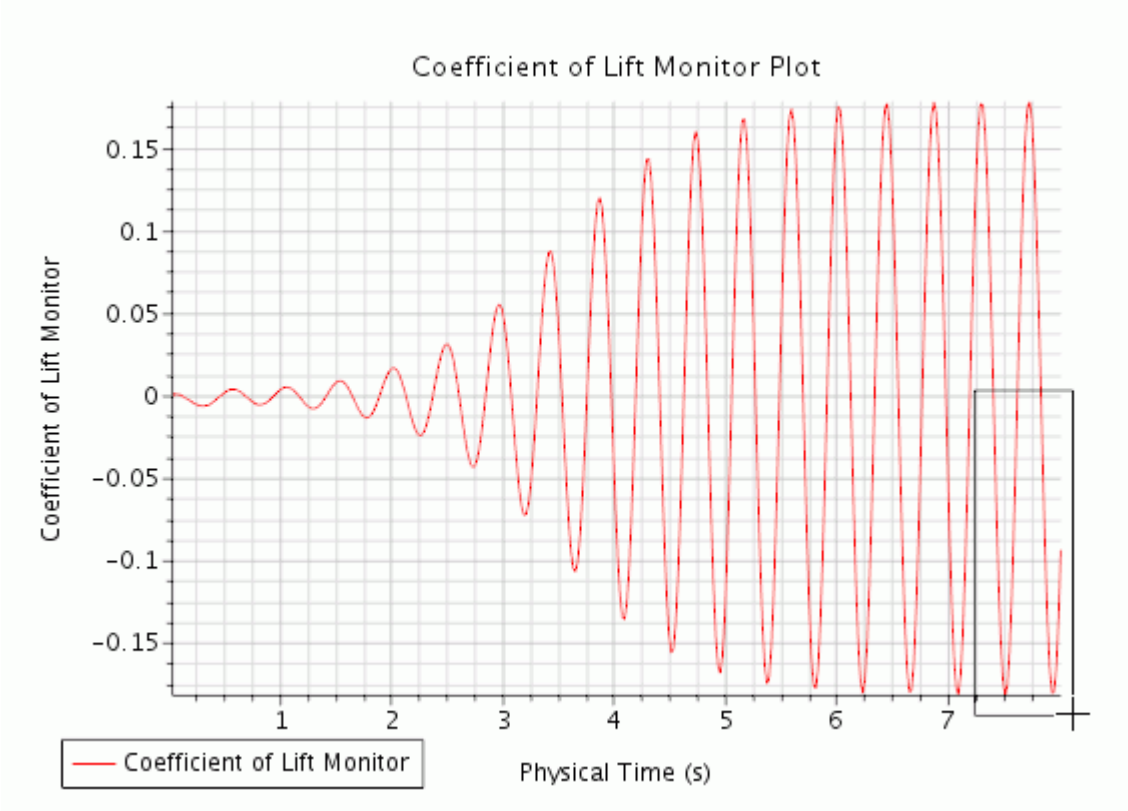
From the scalar scene and the monitor plot, it is clear that vortex shedding is occurring. The Strouhal number (St) is commonly used when describing oscillating flows and is defined as:

$$St = \frac{fD}{U} \tag{4159}$$

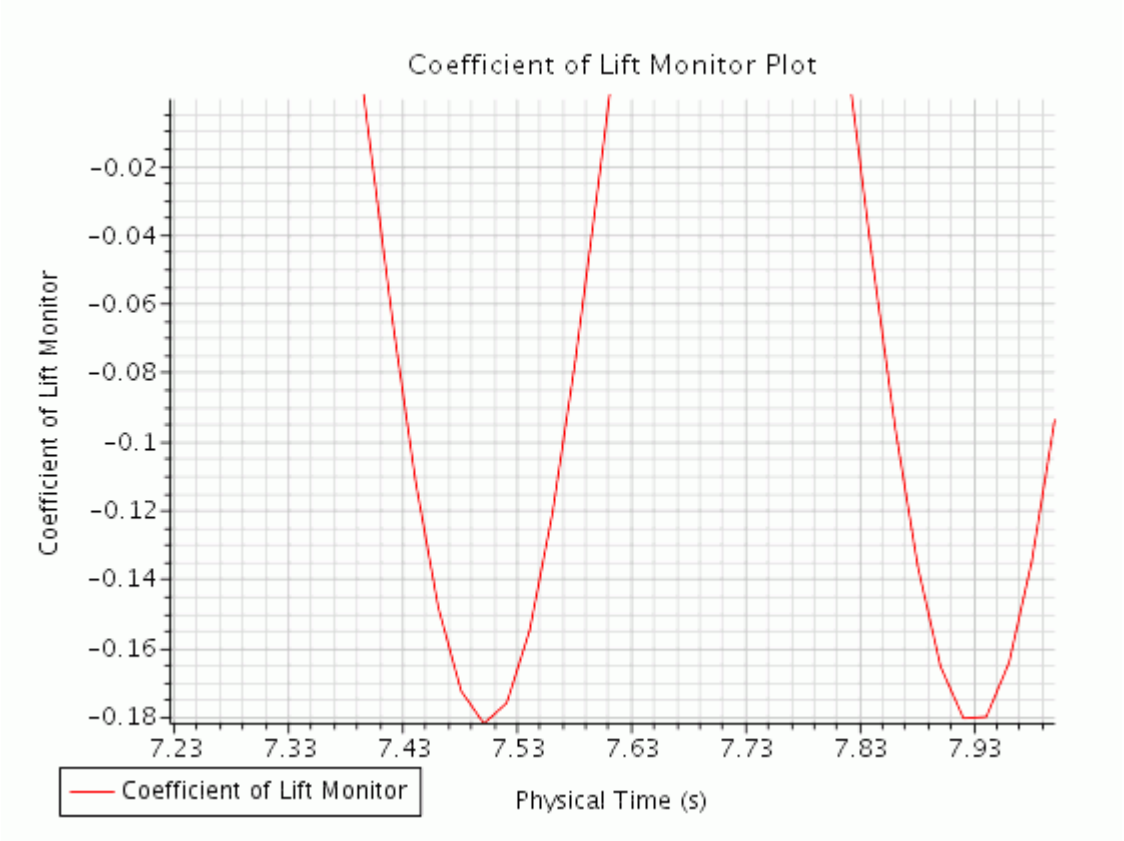
Where f is the frequency of vortex shedding, D is the cylinder diameter, and U is the free-stream velocity. In this case, the Strouhal number is given as 0.15 by Daily et al. [413]. The theoretical frequency of vortex shedding is therefore calculated as 2.25 Hz, which gives a period of 0.444 seconds.

The predicted period of shedding can be obtained by zooming into the last two troughs of the monitor plot.

1. Click (**Zoom Selected Plot**) in the **Plot** toolbar.
2. Drag a box around the last two troughs on the plot, as shown below.

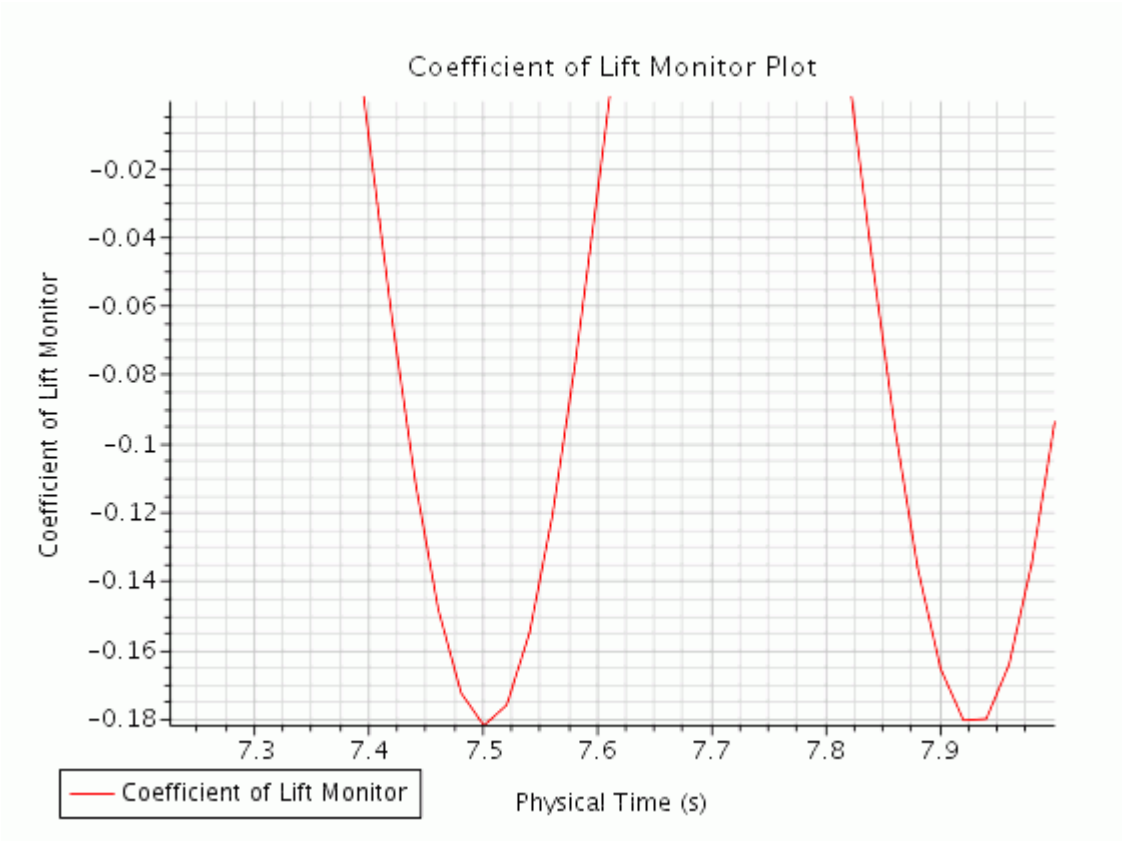


The resulting plot is shown below.



3. Click the **Scene/Plot** button.
4. Select the **Coefficient of Lift Monitor Plot** > **Axes** > **Bottom Axis** > **Major Labels** node, deactivate *Auto Spacing* and *Auto Precision*, and set *Spacing* to `0.1`.
5. Select **Minor Ticks** and set *Count* to `4`.
6. Click the **simulation** button.

   The enlarged scale on the X-axis makes it possible to measure the period.



The predicted period is shown to be approximately 0.44 seconds.

Note: A relatively large time step was used for this tutorial resulting in a limited number of data points in the plot. If you want more accurate results and a smoother plot, reduce the time step.

There is a difference of less than 1% between the predicted period and the reference period, which is good agreement. The corresponding predicted frequency of 2.27 Hz is also in good agreement with the theoretical frequency of vortex shedding of 2.25 Hz.

Creating a Recorded Solution View

Solution views are used to interrogate this data and make it available for post-processing.

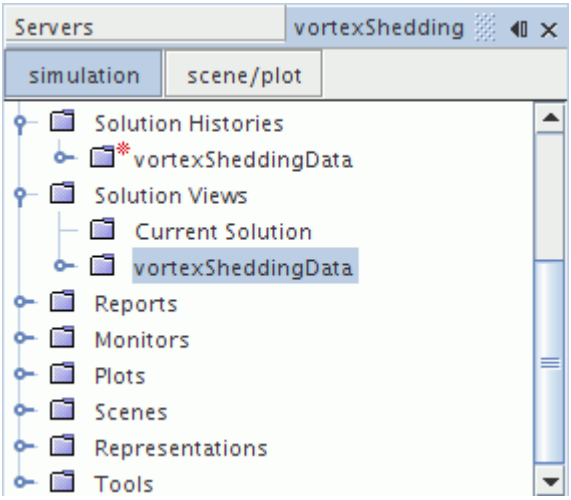
Properties of the solution view set the point in the solution history at which data is read. Data is read into a separate representation linked to the solution view. The solution history file contains all the data that was specified in the previous part of the tutorial.

1. Select the **Solution Histories** > **vortexSheddingData** node.

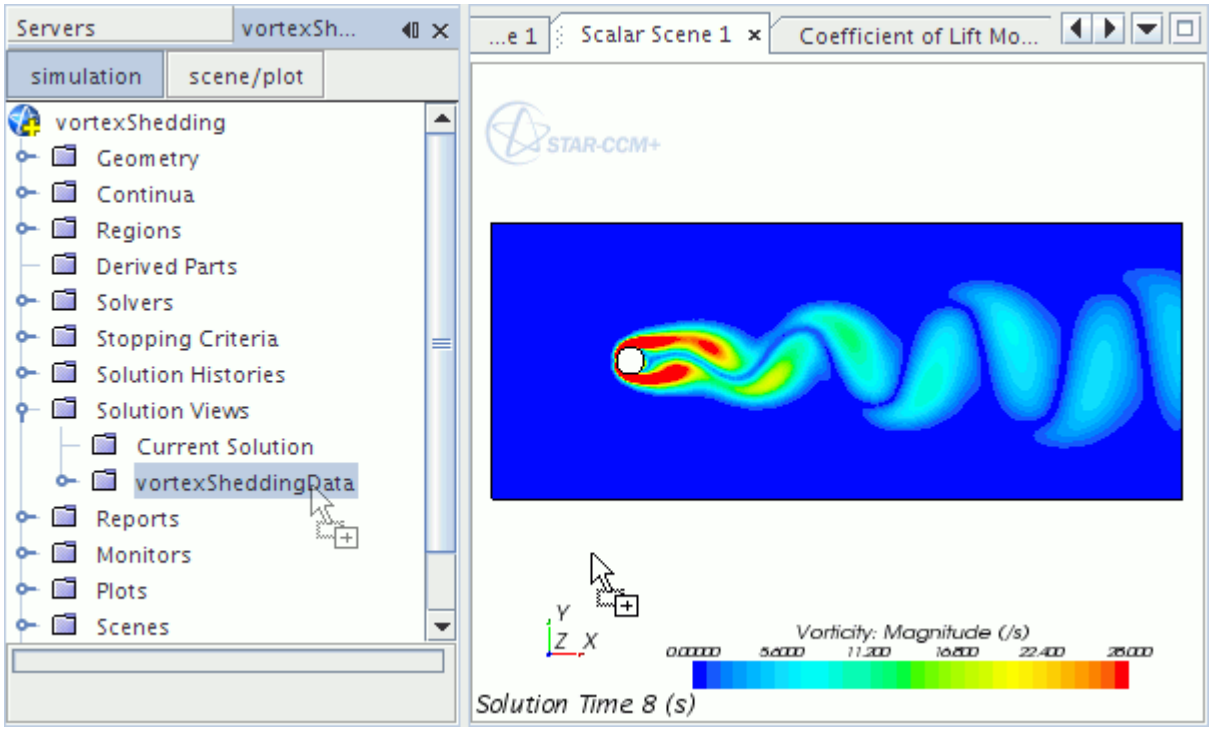
   The number of states stored in the solution history file is displayed next to the *States* property.

   There are 400 states stored in the solution history file.
2. Create a solution view to access the states.
3. Right-click the **Solution Histories** > **vortexSheddingData** node and select **Create Recorded Solution View**.
4. Click **Yes** in the confirmation dialog.

   A new node, **vortexSheddingData**, is added to the **Solution Views** node. The properties of the solution view node control the data shown by the representation associated with it.

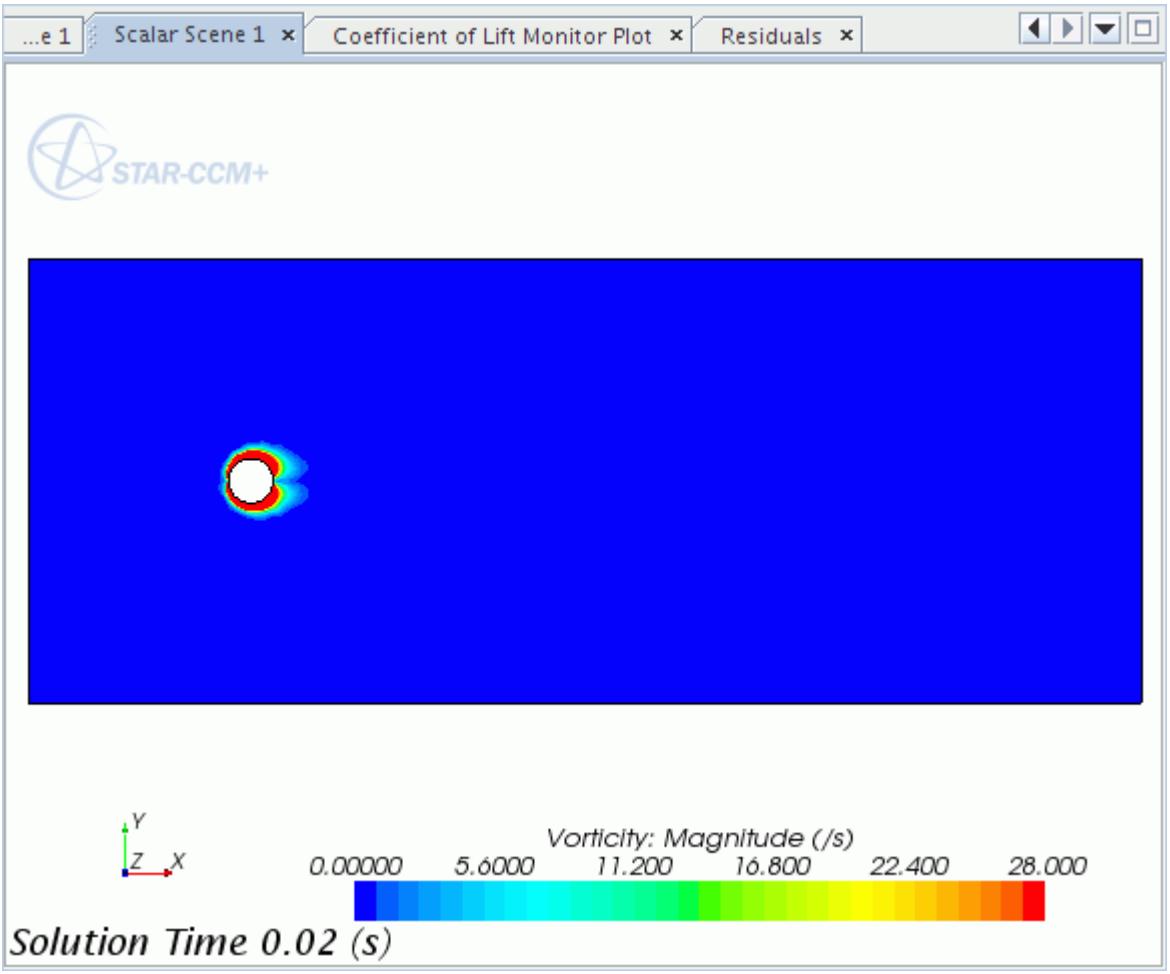


   An additional representation linked to this view was added to the object tree under the **Representations** node. This representation stores solution data from the solution history file.
5. Click the **Scalar Scene 1** tab in the **Graphics** window.
6. Drag and drop the **Solution Views** > **vortexSheddingData** node onto a blank area in the scene window**.**



This action sets the *Representation* property for all displayers (**Outline 1** and *Scalar 1*) and solution history annotations (**Solution Time**) in the scene to **vortexSheddingData**.

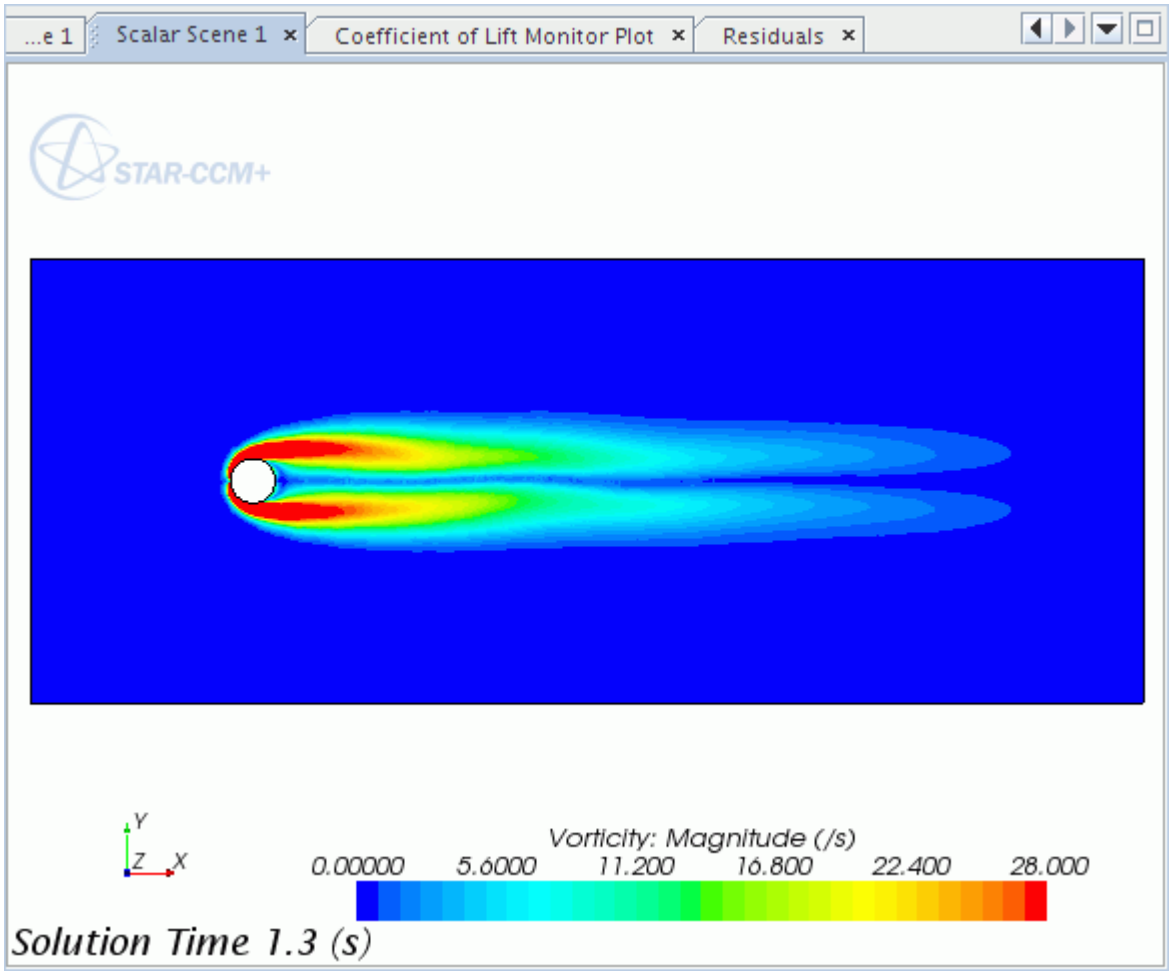
The scene is shown below.



This scene corresponds to the data stored in the solution history file for the first time-step. Adjust the solution time to display the solution data at 1.3 seconds.

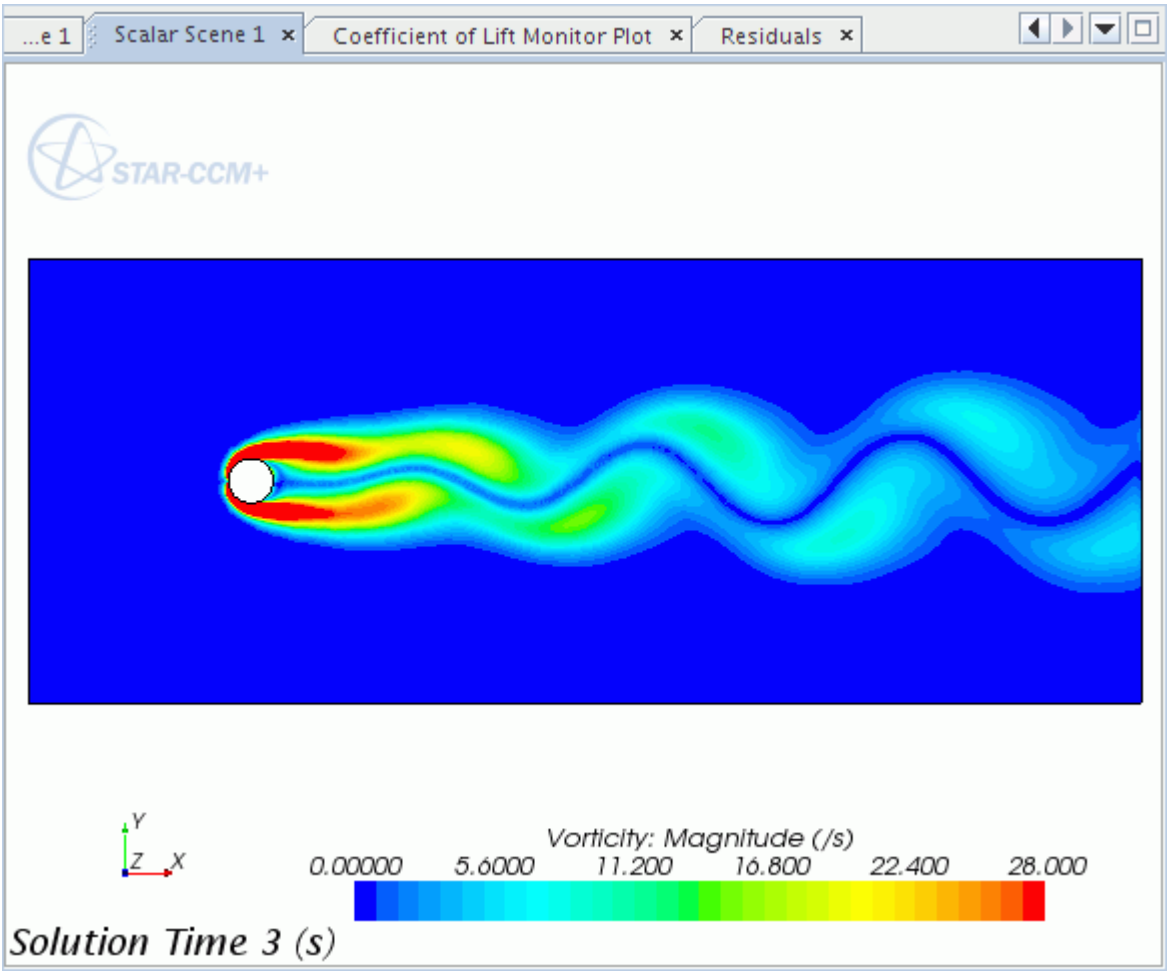
7. Select the **Solution Views** > **vortexSheddingData** node and set *Solution Time* to `1.3 s`.

   The scene updates as shown below.



8. Set *Solution Time* to `3 s`.

   The updated scene is shown below.



Restore the scene to use the most recent solution from the simulation file (stored on the volume mesh representation).

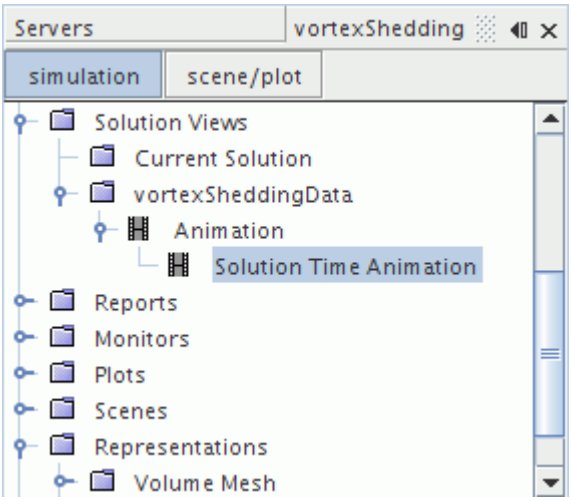
9. Drag the **Solution Views** > **Current Solution** node into the scene as previously described.
10. Save the simulation.

Creating an Animation from the Solution View

Create an animation that shows the development of vortex shedding from the start to a regular periodic state.

1. Drag the recorded solution view, **Solution Views** > **vortexSheddingData**, back into the scene.
2. Select the **Solution Views** > **vortexSheddingData** > **Animation** node and set the *Animation Mode* to **Solution Time**.

   A new node, **Solution Time Animation**, appears below the **Animation** node.



When creating an animation, it is important to set the correct framerate. If the framerate is set too high, the video may reuse identical frames. If the framerate is set too low, the video may appear to play too quickly.

One way to determine the correct framerate is to divide the total number of states (frames) by the total physical time of the simulation. In this case you have 400 states and a physical time of 8 seconds. Knowing this, you can calculate that 50 frames make up one second of simulation time.

Adjust the framerate for the animation.

3. Click the **scene/plot** button.
4. Select the **Scalar Scene 1** > **Attributes** > **Scene animation** node and set the *Target frame rate (fps)* to `50`.

Record the animation.

5. Click (**Write Movie**) in the **Animation** toolbar.

   The *Write animation* dialog appears.
6. Set the following properties:

Property	Value
Animation Length	`8`
Size	A resolution of your choice
File Name	`vorticityAnimation.avi`

7. Click **Save** to write the animation to disk.
8. Play back the animation using a player of your choice.

Summary

This tutorial demonstrated how to use the solution recording and playback module for capturing the results of transient phenomena.

The tutorial covered the following:

- Creating a Solution History file.
- Selecting appropriate scalars to save to the Solution History file at suitable intervals.
- Plotting the lift coefficient and validating the period of shedding against reference data.
- Creating a Recorded Solution View.
- Viewing recorded solution data in a scene.
- Recording an animation from the Solution History file.

Bibliography

[786] *Daily, J.W., and Harleman, D.R.F. Fluid Dynamics, Addison-Wesley, MA, 1966*

Generalized Newtonian Fluid: Flow in a Static Mixer

This tutorial demonstrates the process of setting up and analyzing the flow of a viscous fluid through a static mixer.

Static mixers are used to mix fluids, immiscible liquids, and liquid/gas combinations. Commercial applications include food production, the petrochemical industry, and plastics manufacture, among other industries.

The approach here is to start with a simple model of the mixer and then add the more complex model for viscous energy. You can then visualize the residence time of material moving through the mixer and the temperature levels that the flow generates. Residence time gives a measure of the speed of the mixing process. The operating temperature allows you to determine if the mixing generates heat in an acceptable range; for instance, you generally do not want food stuffs to cook while in the mixer.

Contents:

Prerequisites
Creating the Simulation and Importing the Surface Mesh
Defining Simulation Topology
Generating the Volume Mesh
Setting Up Viscous Flow Physics
Setting Up Boundary Conditions
Setting Up Solver Properties
Preparing to Visualize Viscous Flow
Running the Simulation
Solve with Viscous Energy

Prerequisites

The Generalized Newtonian Fluid: Flow in a Static Mixer tutorial assumes that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Creating the Simulation and Importing the Surface Mesh

To set up the STAR-CCM+ simulation, launch a simulation and import the supplied surface mesh.

1. Launch STAR-CCM+.
2. Start a simulation.

3. Select **File** > **Import** > **Import Surface Mesh**.
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/incompressibleFlow/data`.
5. Select `static_mixer.x_b` then click **Open**.
6. To accept the default *Import Surface Options*, click **OK**.
7. Click **Create/Open Scenes** and select **Geometry**.
   A geometry scene opens.
8. Move the mixer in the scene and click **Make Scene Transparent** .

9. Save the simulation as `StaticMixer`.

Defining Simulation Topology

Create regions and boundaries, and designate the inlet and outlet.

1. Right-click **Geometry** > **Parts** > **tube** and select **Assign Parts to Regions...**
2. In the dialog, select the following:
   - **Create a Region for Each Part**
   - **Create a Boundary for Each Part Surface**
3. Click **Apply**, then **Close**.

4. Select **Regions** > **tube** > **Boundaries** > **Inlet** and set *Type* to **Mass Flow Inlet**.
5. Continuing within **Boundaries**, select **Outlet** and set *Type* to **Pressure outlet**.

Generating the Volume Mesh

The Viscous Flow solver relies on a finite element method. Provide it with a mesh that is composed of nodes and elements rather than cells and faces. In STAR-CCM+, a mesh that is composed of tetrahedral cells can support finite element solvers.

To create a tetrahedral mesh in the mixer:

1. Right-click **Geometry** > **Operations** and select **New** > **Mesh** > **Automated Mesh**.
2. Select the following:

Pane	Selection
Parts window	tube
Select Meshers group box	Surface Remesher Tetrahedral Mesher

3. Click **OK**.

Elements at the core of the mesh can be larger than elements on the surface. To achieve this size, you set a base size of 0.05 m, and then apply smaller sizing on the wetted surfaces.

4. Select **Automated Mesh** > **Default Controls** > **Base Size** and set *Value* to `0.05 m`.
5. Right-click **Automated Mesh** > **Custom Controls** and select **New** > **Surface Control**.
6. Select **Custom Controls** > **Surface Control** and edit *Part Surfaces*. In the dialog, select the following:
   - **Blade_A**
   - **Blade_B**
   - **Tube_wall**

   Click **OK**.
7. Expand **Surface Control** > **Controls** and set the following properties:

Node	Property	Value
Target Surface Size	*Target Surface Size*	**Custom**
Minimum Surface Size	*Minimum Surface Size*	**Custom**

8. Select **Surface Control** > **Values** and set the following properties

Node	Property	Value
Target Surface Size	*Percentage of Base*	`10.0`
Minimum Surface Size	*Percentage of Base*	`5.0`

9. Click **Generate Volume Mesh** .
10. Click **Show All Meshes** to show the mesh in the *Geometry* scene.

Save the simulation.

Setting Up Viscous Flow Physics

The Viscous Flow model appears alongside the Segregated and Coupled Flow models for a physics continuum.

To select the physics models:

1. For the physics continuum, **Continua** > **Physics 1**, select the following models, in order:

Group Box	Model
Enabled Models	**Three Dimensional** (Selected automatically)
Time	**Steady**
Material	**Liquid**
Flow	**Viscous Flow** **Constant Density** (Selected automatically) **Laminar** (Selected automatically)
Optional Models	**Residence Time**

Group Box	Model
Rheology	**Generalized Newtonian**

An important step in preparing a viscous flow simulation is choosing the rheology model that best represents the fluid that you are working with. In this tutorial, the viscous behavior of liquid polystyrene fits the characteristics of a Non-Newtonian Generalized Carreau-Yasuda Fluid. Here, the model parameters were previously tuned to match the fluid behavior.

2. Select **Physics 1** > **Models** > **Liquid** > **H2O** and rename it to `Polystyrene`.
3. Select **Polystyrene** > **Material Properties** and set the following properties:

Node	Property	Value
Density > **Constant**	*Value*	`1500 kg/m^3`
Dynamic Viscosity	*Method*	**Non-Newtonian Generalized Carreau-Yasuda Fluid**
└ Non-Newtonian Generalized Carreau-Yasuda Fluid	*Power Constant*	`0.25`
	a Parameter	`2`
	Zero Shear Viscosity	`1100000 Pa-s`
	Infinite Shear Viscosity	`3000.0 Pa-s`
	Relaxation Time	`8.0 s`

Setting Up Boundary Conditions

Set the boundary conditions on the inlet.

1. Open **Region** > **tube** > **Boundaries**.
2. Select **Inlet** > **Physics Values** > **Mass Flow Rate** and set *Value* to `0.05 kg/s`.

Save the simulation.

Setting Up Solver Properties

Set the Viscous Flow solver to use barycentric shape functions, so that variables are evaluated only at element vertices.

- Select **Solvers** > **Viscous Flow Solver** and deselect the *Parametric Elements* property.

Preparing to Visualize Viscous Flow

Create cross sections and vector and scalar scenes for fluid velocity and residence time.

1. Create the plane sections:
   a) Right-click **Derived Parts** and select **New Part** > **Section** > **Plane...**

b) In the dialog, enter the following properties:

Property	Value
Origin	`[0.0, 0.0, 0.0]`
Normal	`[1.0, 0.0, 0.0]`
Display	**No Displayer**

c) Click **Create**.

STAR-CCM+ adds the first cross-section.

d) In the dialog, enter the following properties:

Property	Value
Origin	`[0.0, 0.0, 0.0]`
Normal	`[0.0, 1.0, 0.0]`
Display	**No Displayer**

e) Click **Create** and then **Close**.

2. Create the velocity vector scene:
   a) Create a vector scene.
   b) Open **Scenes** and rename **Vector Scene 1** to `Velocity`.
   c) In the *Scene/Plot* panel, select **Displayers** > **Vector 1** > **Parts** and set *Parts* to **Derived Parts** > **Plane Section** and **Plane Section 2**.
   d) Select **Vector 1** > **Color Bar** and set *Color Map* to **purple-red**.
3. Create the residence time scalar scene:
   a) Create a scalar scene.
   b) Rename **Scalar Scene 1** to `Residence Time`.
   c) In the *Scene/Plot* panel, select **Displayers** > **Scalar 1** > **Scalar Field**, and set *Function* to **Residence Time**.
   d) Select **Scalar 1** > **Parts** and set *Parts* to **Derived Parts** > **Plane Section** and **Plane Section 2**.
   e) Select **Scalar 1** > **Color Bar** and set *Color Map* to **gray-darkred**.
4. Select the **Scalar 1** node and set *Contour Style* to **Smooth Filled**.

Save the simulation.

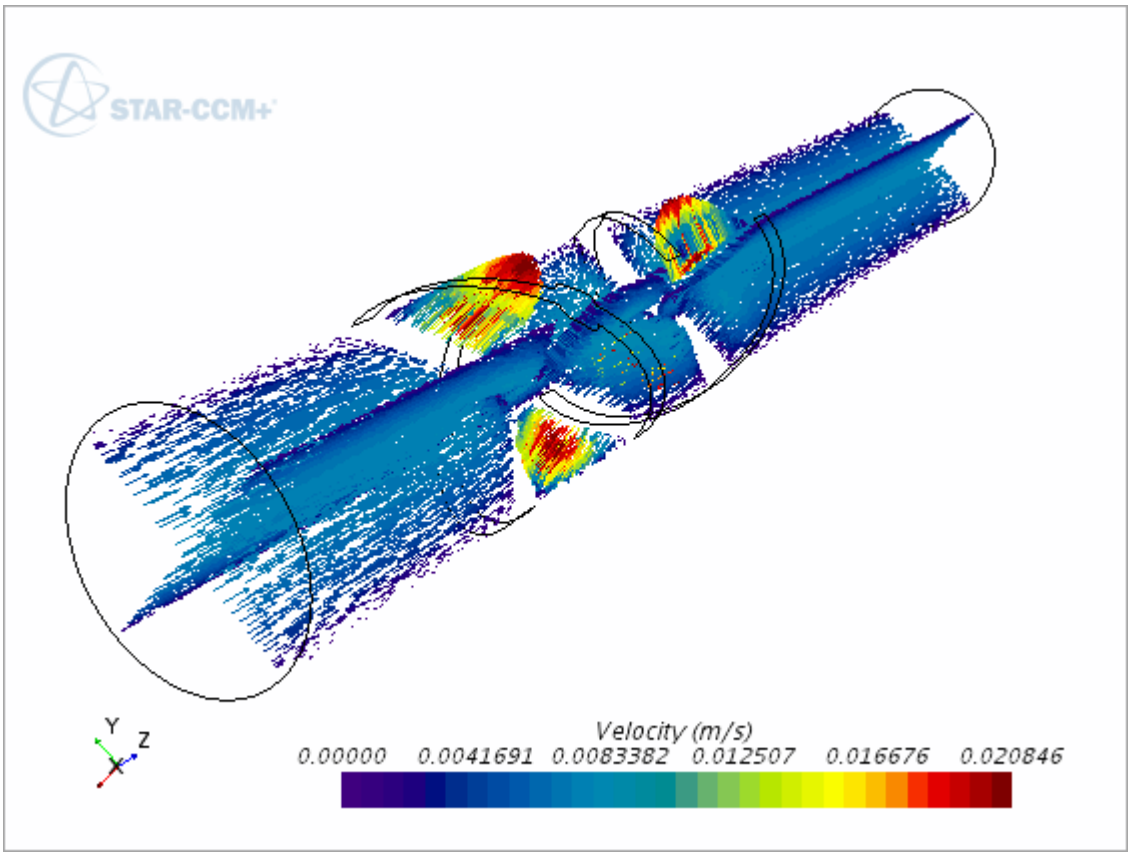
Running the Simulation

Run the simulation and view the results.

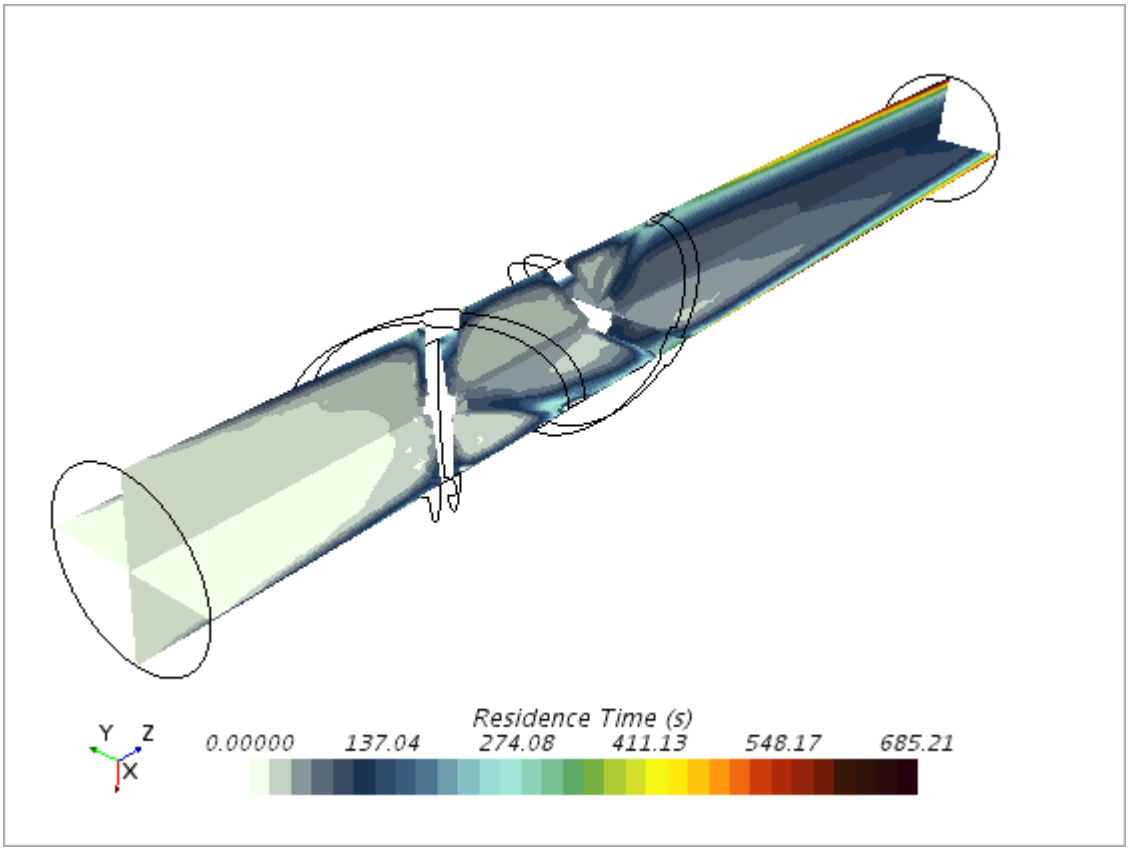
1. Select **Stopping Criteria** > **Maximum Steps**.
2. Set the *Maximum Steps* property to `15`.
3. Select **Stopping Criteria** > **Material Relaxation** and deactivate **Enabled**.
4. Click Run .

5. Examine the *Velocity* and *Residence Time* scenes. Move the images to get views like the ones in the following screenshots.

The *Velocity* scene:



The *Residence Time* scene:



Solve with Viscous Energy

Add the Viscous Energy model to the simulation and create a scene to track temperature.

1. Add the Viscous Energy model:
   a) Right-click **Physics 1** > **Models** and select **Viscous Energy**.
   b) Select **Models** > **Liquid** > **Polystyrene** > **Material Properties** and set the following properties:

Node	Property	Value
Specific Heat	*Method*	**Constant**
└ **Constant**	*Value*	`2000 J/kg-K`
Temperature Shift Factor	*Method*	**Arrhenius**
└ **Arrhenius**	*Activation Energy*	`52000 J/kmol`
	Reference Temperature	`413.0 K`

   c) Go to **Regions** > **tube** > **Boundaries** > **Inlet**, select **Physics Values** > **Static Temperature**, and set the *Value* property to `413.0 K`.
2. Create a Temperature Scalar scene:
   a) Right-click **Scenes** and select **New Scene** > **Scalar**.

b) Rename **Scalar Scene 1** to `Temperature`.

c) Click the *Scene/Plot* tab, select **Displayers** > **Scalar 1** > **Scalar Field**, and set the property *Function* to **Temperature**.

d) Select **Scalar 1** > **Parts**.

e) Edit the *Parts* and select **Derived Parts** > **Plane Section** and **Plane Section 2**.

f) Select **Displayers** > **Scalar 1** > **Color Bar** and set the property *Color Map* to **blue-red-balanced**.

g) Select the **Scalar 1** node and set the property *Contour Style* to **Smooth Filled**.

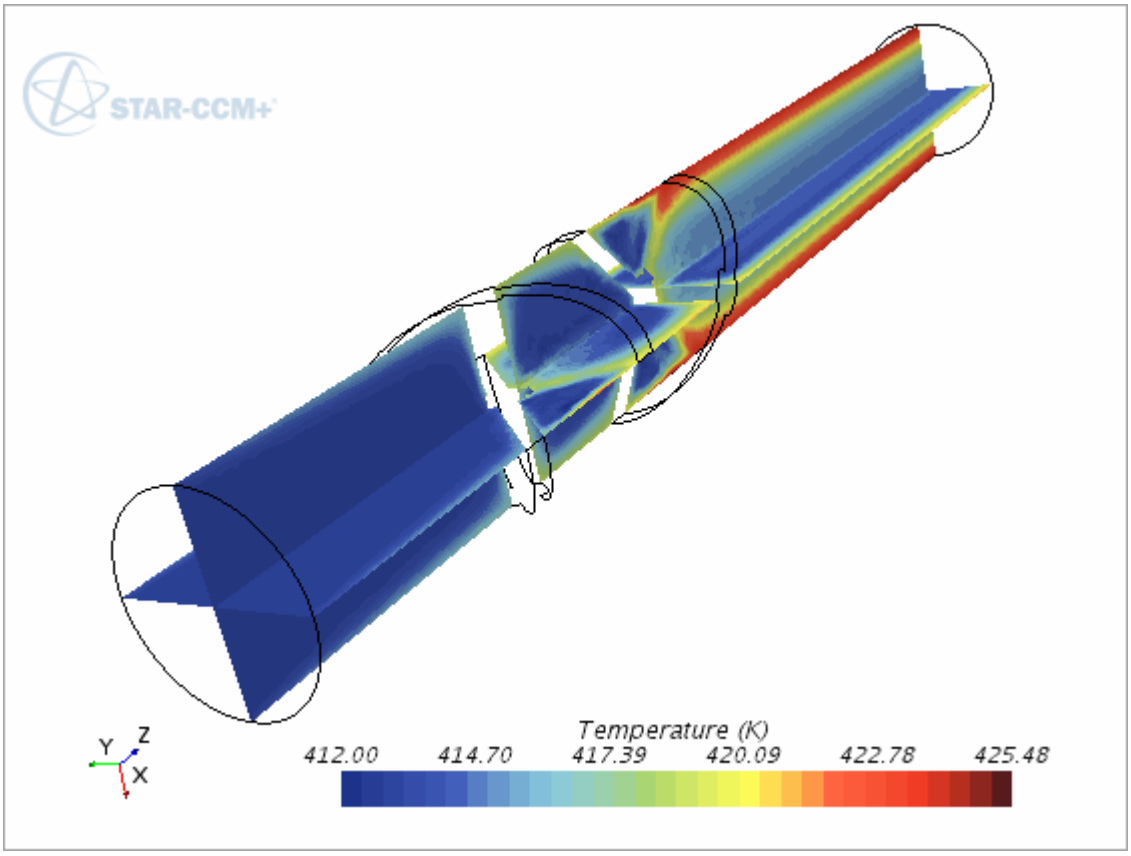
Now run the simulation again:

3. Select **Solution** > **Clear Solution** and click **OK** in the dialog.

4. Click **Run** .

5. Examine the *Temperature* scene. Move the image to get a view like the one in the following screenshot.

The *Temperature* scene shows that the temperature of the liquid rises close to the walls of the mixer:



Viscoelastic Flow: Basic Extrusion

This tutorial illustrates the extrusion of molten rubber through a die.

Many industrial processes involve the extrusion of viscous materials. In the automotive industry, rubber seals are formed by extruding molten rubber through a die. Simulations such as this one allow you to anticipate the swelling of the extrusion as it leaves the die

In the course of the tutorial, you compare the results of an extrusion simulated as a simple viscous fluid and then as a viscoelastic fluid:

1. Set up the die and the extrusion of a viscous fluid (liquid rubber) through it, modeled as a Generalized Newtonian fluid.
2. Visualize the flow velocity through the simulation and the swelling of the extrusion.
3. Change the fluid model from Generalized Newtonian to Viscoelastic.
4. Visualize the flow velocity and extrusion swelling again with the change of model.

The tutorial uses the simpler Generalized Newtonian model first to establish that the simulation gives reasonable results. It then moves to the more complex and computationally expensive Viscoelastic model for greater accuracy.

Contents:

Prerequisites
Creating the Simulation and Importing the Surface Mesh
Defining Simulation Topology
Generating the Volume Mesh
Setting Up the Physics
Setting Up Boundary Conditions
Setting Up Solver Properties
Preparing to Visualize the Velocity Field and Shear Rate
Running the Simulation
Introducing Viscoelasticity
Running the Viscoelastic Simulation

Prerequisites

The Viscoelastic Flow: Basic Extrusion tutorial assumes that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Creating the Simulation and Importing the Surface Mesh

Create a simulation, name it `ExtrusionDie`, and import the supplied geometry.

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Surface Mesh** and navigate to `[INSTALL-DIR]/doc/startutorialsdata/incompressibleFlow/data`.
4. Select `RubberSealGeometry.x_b` then click **Open**.
5. To accept the default *Import Surface Options*, click **OK**.

   STAR-CCM+ imports the file.
6. Move the image into position, as shown in the following screenshot, and click **Make Scene Transparent**

   .

   The scene shows molten rubber passing through the die. The short section of the geometry represents the die. The long section represents the extrusion. The free surface of the extrusion is typically set to ten times the greatest dimension of the die cross section, to promote rapid convergence. If the free surface is too short, the solution still converges, but the extrusion profile does not allow a fully developed flow to form.
7. Save the simulation as `ExtrusionDie`.

Defining Simulation Topology

Create regions and boundaries, and designate the inlet and outlet.

1. Expand **Geometry** > **Parts** and select **die** and **extrusion**.
2. Right-click and select **Assign Parts to Regions...**
3. In the dialog, select the following:
   - **Create a Region for Each Part**
   - **Create a Boundary for Each Part Surface**
4. Click **Apply**, then **Close**.
5. Expand **Regions** and set the following properties:

Node	Property	Value
die > **Boundaries**		
└ **die symmetry plane**	*Type*	**Symmetry Plane**
└ **inlet**	*Type*	**Mass Flow Inlet**
extrusion > **Boundaries**		

Node	Property	Value
└ **extrusion symmetry plane**	*Type*	**Symmetry Plane**
└ **free surface**	*Type*	**Free Stream**
└ **outflow**	*Type*	**Free Stream**

The symmetry plane on the bottom surfaces of the die and the extrusion allow you to simulate an arrow-shaped cross section using the simpler shape of half an arrow.

Setting the **free surface** and **outflow** boundaries to type **Free Stream** allows morphing, to simulate extrusion swelling.

Save the simulation.

Generating the Volume Mesh

Generate mesh for die and extrusion. The following steps create a volume mesh suitable for use with the finite element method.

This procedure meshes the die and the extrusion separately. Both get directed meshes, but the extrusion mesh uses a stretch function to increase the length of the cells further from the die. Extrusions must have directed meshes, directed in the direction of the extrusion. The die region does not require a directed mesh; for more complex geometries, use simpler methods such as tetrahedral meshing.

Specify the mesh for the die.

1. Right-click **Geometry** > **Operations** and select **New** > **Mesh** > **Directed Mesh**.
2. In the *Create Directed Mesh Operation* dialog, select **die**, then click **OK**.
3. Right-click **Operations** > **Directed Mesh** and select **Edit...**

   The *Directed Mesh* tab opens.
4. In the *Directed Mesh* tab, select **Source Surfaces** and set *Source Surfaces* to **inlet**.
5. Select **Target Surfaces** and set *Target Surfaces* to **die interface**.

The directed mesher creates a mesh by sweeping an initial mesh from the source surface to the target surface.

6. Set the base size for the mesh:
   a) Right-click **Source Meshes** and select **New Source Mesh** > **Automated Source Mesh**.
   b) In the *Part Collection For This Source Mesh* dialog, select **die** in the upper pane. From within the *Select Meshers* group box, select **Triangular Mesher**. Click **OK**.
   c) Select **Source Meshes** > **Auto Mesh** > **Default Controls** > **Base Size** and set *Value* to `4.0 mm`.
   d) Right-click **Auto Mesh** and select **Update...**
7. Set the number of layers:
   a) Right-click **Mesh Distributions** and select **New Volume Distribution**.

b) In the *Parts for creating a new mesh distribution* dialog, select **die** and click **OK**.

c) Select **Mesh Distributions** > **Volume Distribution** > **Default Controls** > **Number of Layers** and set *Number of Layers* to `10`.

8. At the top of the object tree, right-click **Directed Mesh** and select **Execute**.

   STAR-CCM+ sweeps the source mesh along the part.

9. Click **Close Directed Mesh** at the bottom of the *Directed Mesh* tab.

Specify the mesh for the extrusion:

10. Right-click **Geometry** > **Operations** and select **New** > **Mesh** > **Directed Mesh**.
11. In the *Create Directed Mesh Operation* dialog, select **extrusion**, then click **OK**.
12. Right-click **Operations** > **Directed Mesh 2** and select **Edit...**

    The *Directed Mesh* tab opens.

13. In the *Directed Mesh* tab, select **Source Surfaces** and set *Source Surfaces* to **extrusion interface**.
14. Select **Target Surfaces** and set *Target Surfaces* to **outflow**.
15. Extend the die mesh into this part:

    a) Right-click **Source Meshes** and select **New Source Mesh** > **Use Existing Mesh**.

    b) In the *Part Collection For This Source Mesh* dialog, select **extrusion**, then click **OK**.

16. Set the number of layers and their distribution:

    a) Right-click **Mesh Distributions** and select **New Volume Distribution**.

    b) In the *Parts for creating a new mesh distribution* dialog, select **extrusion** and click **OK**.

    c) Select **Mesh Distributions** > **Volume Distribution** > **Default Controls** > **Number of Layers** and set *Number of Layers* to `25`.

    d) Select **Volume Distribution** > **Default Controls** > **Stretching Function** and set **Stretching Method** to **One Sided Hyperbolic**.

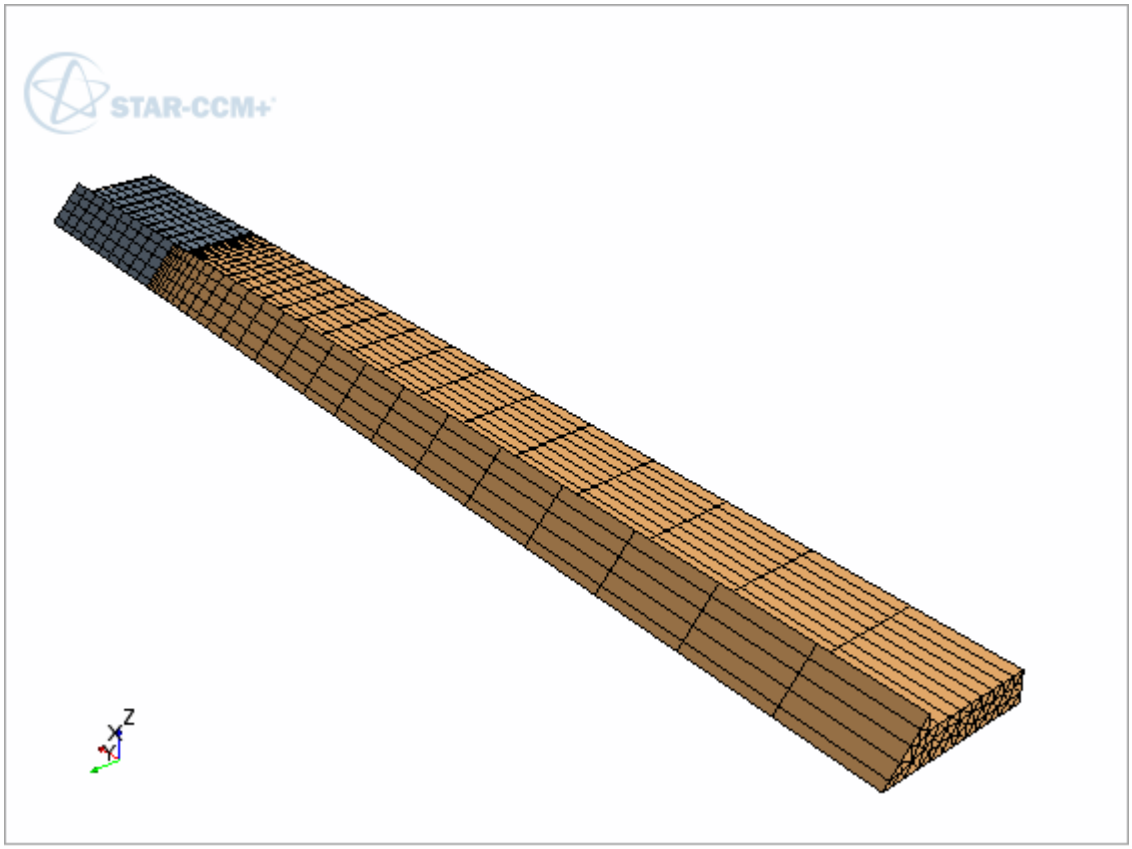
    e) Select **Stretching Function** > **One Sided Options** > **Stretch Value** and set **Stretch Value** to `2.2`.

       This sets the stretching factor. See the Hyperbolic Tangent stretching function.

17. At the top of the tree, right-click **Directed Mesh 2** and select **Execute**.
18. Click **Close Directed Mesh** at the bottom of the *Directed Mesh* tab.

Visualize the final mesh:

19. Click **Create/Open Scenes** and select **Mesh** to show the mesh scene.



20. Save the simulation.

Setting Up the Physics

Physics models define the physical variables and phenomena in the simulation. The main phenomena to solve for in this simulation represent the material properties of liquid rubber and a flow of viscous fluid through the die, out into an extrusion.

To select the physics models:

1. For the physics continuum, **Continua** > **Physics 1**, select the following models, in order:

Group Box	Model
Enabled Models	**Three Dimensional** (Selected automatically)
Time	**Steady**
Material	**Liquid**
Flow	**Viscous Flow** **Constant Density** (Selected automatically) **Laminar** (Selected automatically)
Rheology	**Generalized Newtonian**
Optional Models	**Free Surface**

Selecting the Viscous Flow model adds mesh morphing to the simulation. This addition produces the swelling of the extrusion as it leaves the die.

2. Select **Physics 1** > **Models** > **Liquid** > **H2O** and rename it to `Liquid Rubber`.
3. Select **Liquid Rubber** > **Material Properties** and set the following properties:

Node	Property	Value
Density	*Method*	**Constant**
└ **Constant**	*Value*	`1500 kg/m^3`
Dynamic Viscosity	*Method*	**Constant**
└ **Constant**	*Value*	`2500 Pa-s`

4. Select **Models** > **Free Surface** and set the following properties:

Property	Value
Extrusion Direction	**X**
Kinematic Condition	**Line**
Restrain Y	Activated

The **Kinematic Condition** setting of **Line** is the default and requires the velocity vector on the boundary to be aligned with cell edges and with the extrusion direction. The less restrictive **Surface** setting only requires the velocity vector to be parallel to cell faces. This requirement leaves a rotation degree of freedom to the free surface that is uniquely determined with a minimum energy principle.

Restrain Y must be activated because no plane of symmetry orthogonal to Y exists. This setting requires that the integral of the Y component of the velocity is 0.

Setting Up Boundary Conditions

Set the boundary conditions on the die and extrusion.

1. Edit the **Regions** > **die** > **Boundaries** node.
2. Select **inlet** > **Physics Values** > **Mass Flow Rate** and set *Value* to `0.05 kg/s`.
3. Edit the **Regions** > **extrusion** > **Boundaries** > **free surface** node and set the following properties:

Node	Property	Value
Physics Conditions		
└ **Morpher Specification**	*Specification*	**Displacement**
└ **Morpher Displacement Specification**	*Specification*	**Internal**

4. Edit the **Regions** > **extrusion** > **Boundaries** > **outflow** node and set the following properties:

Node	Property	Value
Physics Conditions		
└ **Morpher Specification**	*Specification*	**Constraint**
└ **Morpher Constraint Specification**	*Specification*	**Boundary Plane**
└ **Traction**	*Method*	**Aligned Exit**

Setting the **Morpher** method to **Internal** at the **free surface** boundary triggers a free surface calculation on this boundary.

Setting the **Morpher** method to **Constraint** at the **outflow** boundary leaves the boundary free to move without activating free surface tracking.

Setting the **Traction** method to **Aligned Exit** sets the velocity tangential to the normal direction to 0, but leaves the aligned component free.

Traction Velocity represents a pull exerted on the extrusion.

The methods available for the **Traction** condition are:

- **No Traction** does not apply any force at the Free Surface exit.
- **Traction Velocity** applies a given pulling velocity at the end of the free surface.
- **Aligned Exit** (used here) applies zero pulling force and zero velocity in the plane normal to the extrusion direction, but leaves the aligned component free.
- **Normal Velocity** applies a velocity normal to the free surface.

Setting Up Solver Properties

Set the Viscous Flow solver to use barycentric shape functions, so that variables are evaluated only at element vertices.

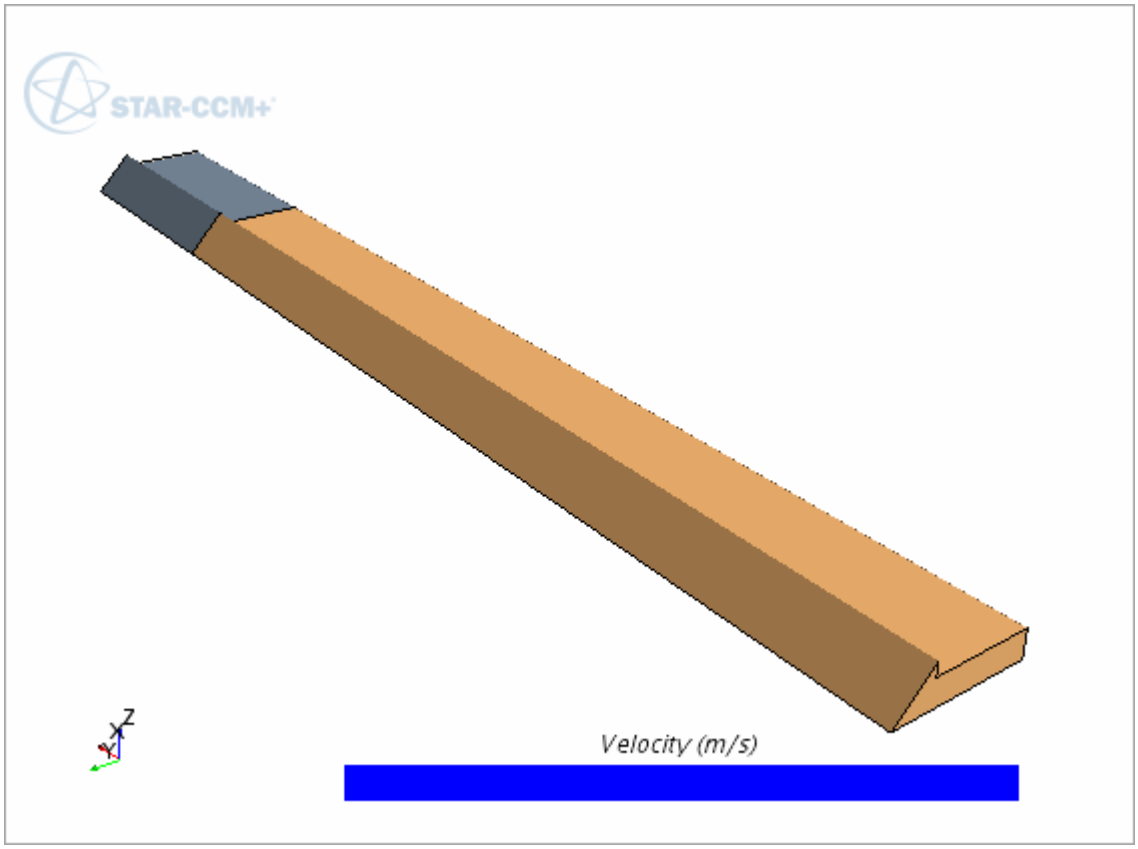
- Select **Solvers** > **Viscous Flow Solver** and deselect the *Parametric Elements* property.

Preparing to Visualize the Velocity Field and Shear Rate

Create a vector scene for the fluid velocity and a scalar scene for the shear rate in the fluid.

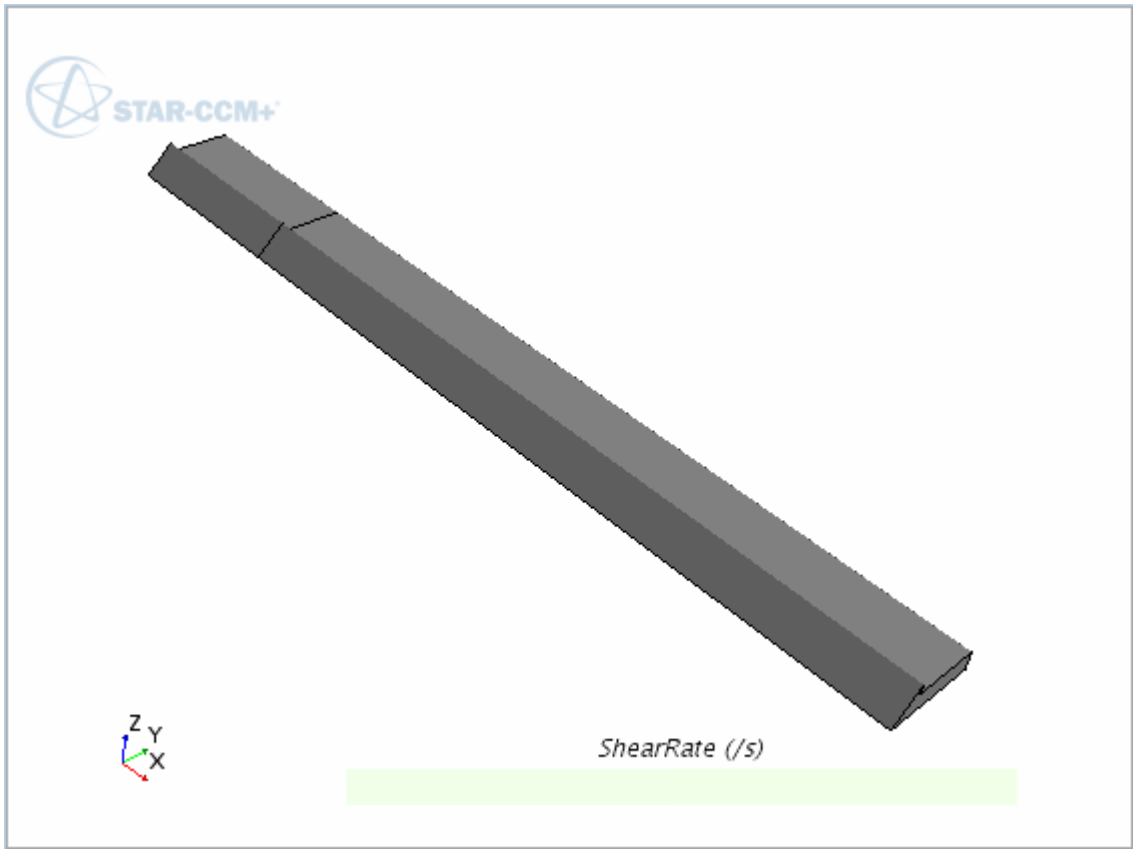
Create the vector scene for the fluid velocity:

1. Right-click **Scenes** and select **New Scene** > **Vector**.
2. Open **Scenes** and rename **Vector Scene 1** to `Velocity`.
3. Select **Velocity** > **Displayers** > **Vector 1** > **Parts** and set *Parts* to **Regions** > **die** and **Regions** > **extrusion**.
4. Select **Displayers** > **Outline 1** and activate **Surface**.
5. Move the image into position, as shown in the following screenshot:



Create the scalar scene for the shear rate:

6. Right-click **Scenes** and select **New Scene** > **Scalar**
7. Open **Scenes** and rename **Scalar Scene 1** to `Shear Rate`.
8. Select **Shear Rate** > **Displayers** > **Scalar 1** > **Parts** and set *Parts* to **Regions** > **die** and **Regions** > **extrusion**.
9. Select **Scalar Field** and set *Function* to **Shear Rate**.
10. Select **Color Bar** and set *Color Map* to **gray-darkred**.
11. Select **Displayers** > **Outline 1** and activate **Surface**.
12. Move the image into position, as shown in the following screenshot.

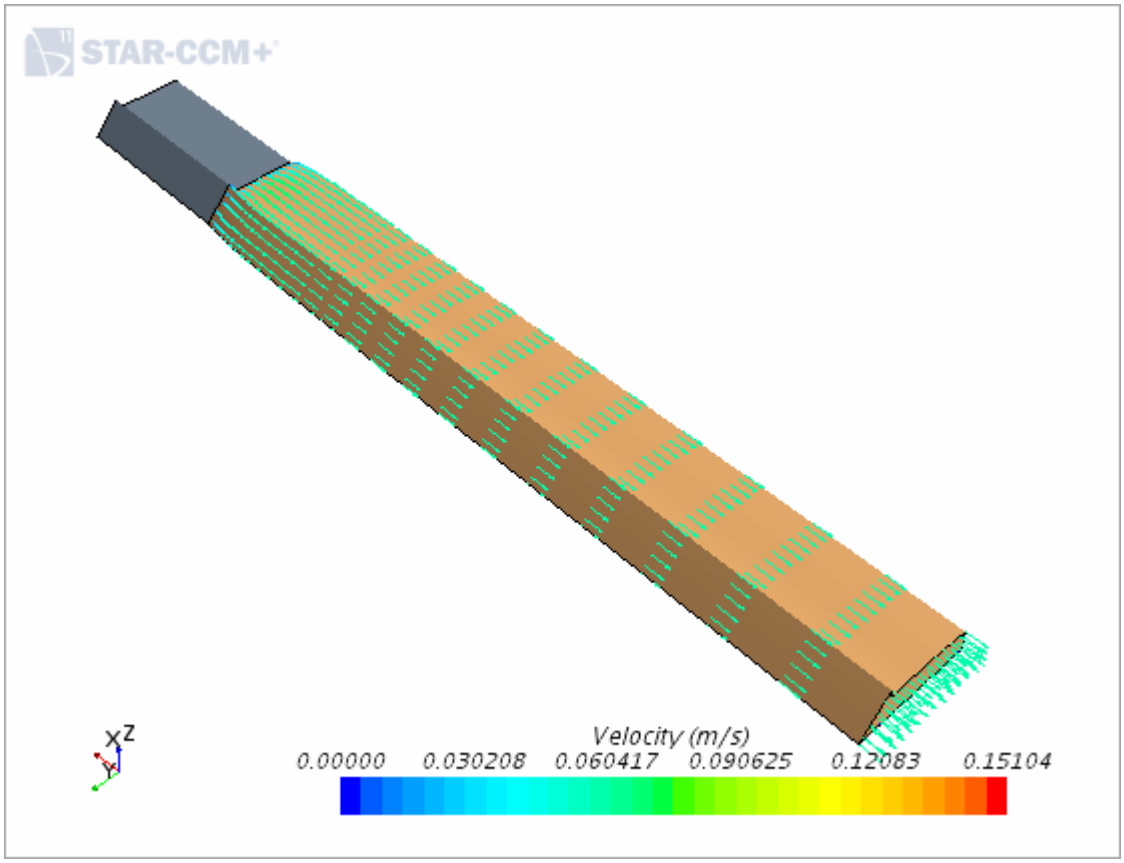


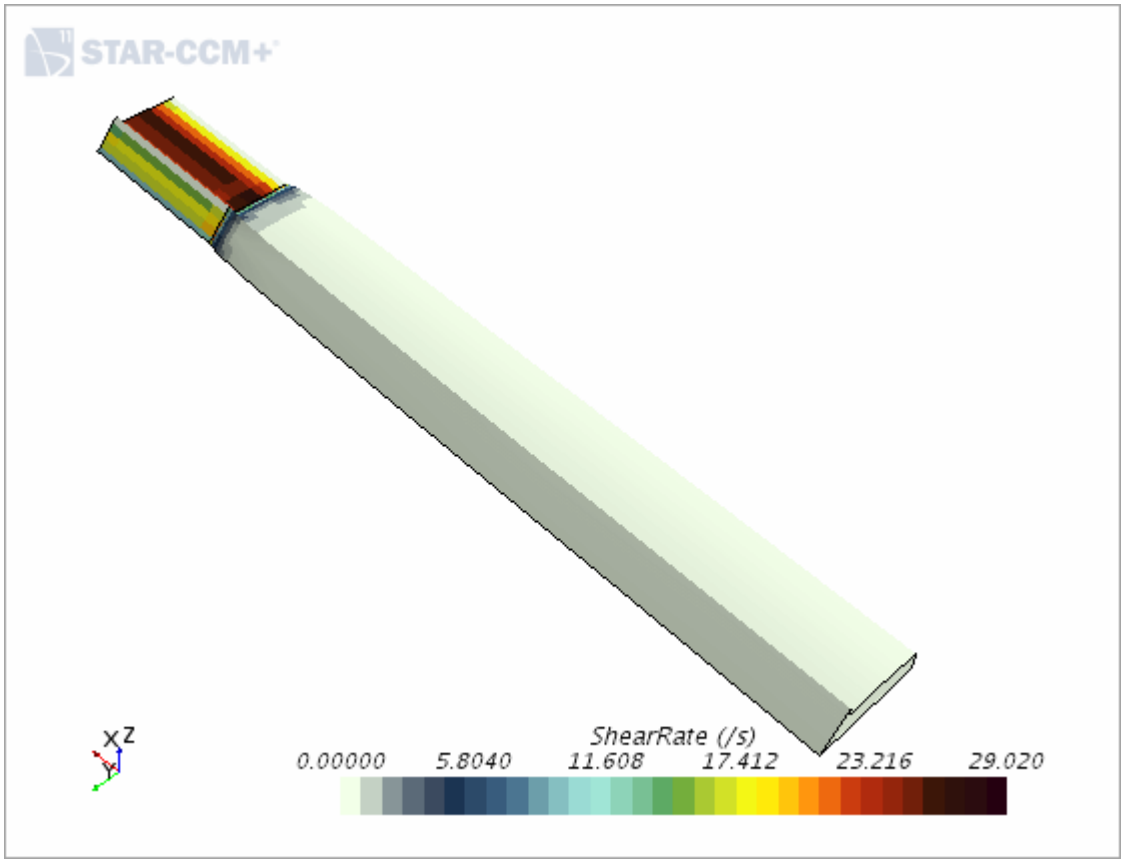
Save the simulation.

Running the Simulation

Run the simulation and view the results.

1. Select **Stopping Criteria** > **Maximum Steps** and set the *Maximum Steps* property to `15`.
2. Select **Stopping Criteria** > **Material Relaxation** and deactivate **Enabled**.
3. Click Run .
4. Examine the *Velocity* scene and the *Shear Rate* scene. The extrusion swells slightly as it leaves the die. The velocity of the extrusion slows and becomes more uniform over the cross-section after leaving the die. The shear rate drops and becomes more uniform over the cross-section after leaving the die.


STAR-CCM+
Velocity (m/s)
0.00000
0.030208
0.060417
0.090625
0.12083
0.15104



Save the simulation.

Introducing Viscoelasticity

Add the Viscoelasticity model to the simulation.

1. In **Physics 1** > **Models**, de-select **Generalized Newtonian**.
2. Select **Viscoelastic** from the *Rheology* box and close.
3. Select **Liquid** > **Liquid Rubber** > **Material Properties** > **Viscoelastic mode 1** and set the *Method* property to **GiesekusLeonov**.
4. Select **Viscoelastic mode 1** > **GiesekusLeonov** and set the following properties:

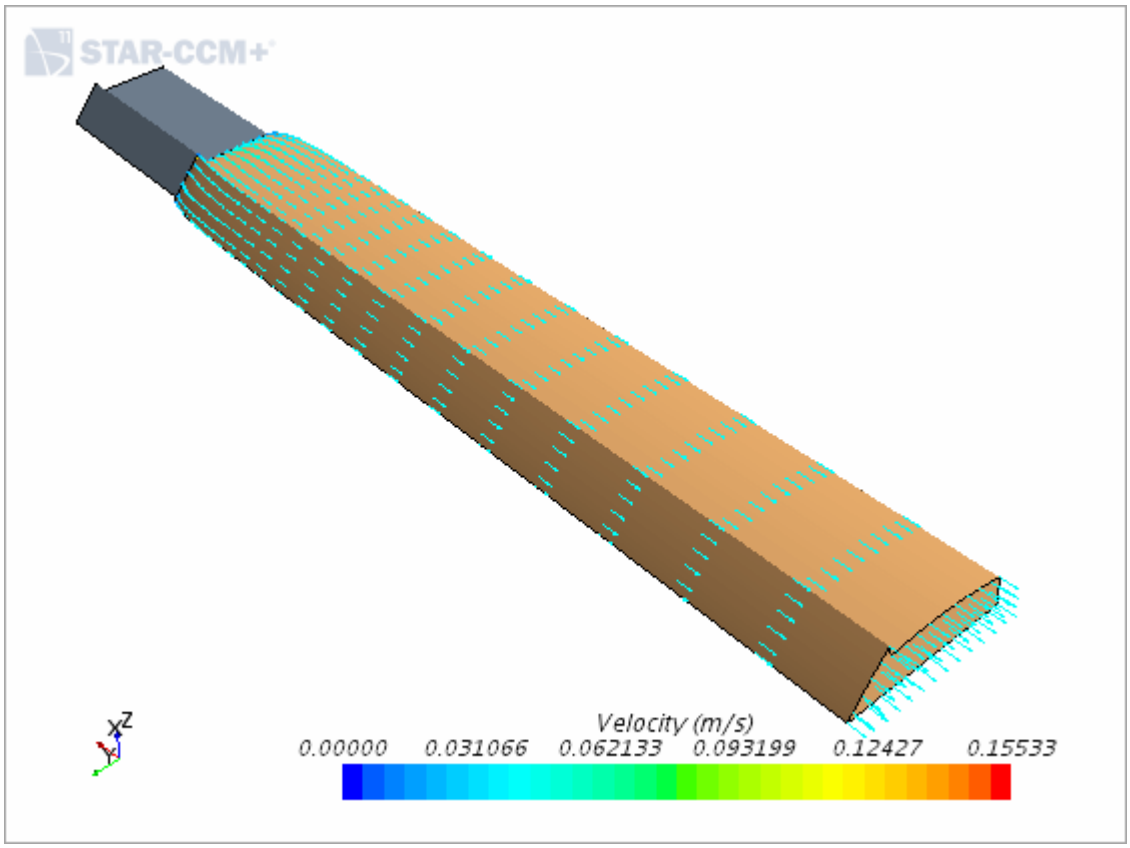
Property	Value
Viscosity	`2500.0 Pa-s`
Lambda	`0.1 s`
Alpha	`0.2`

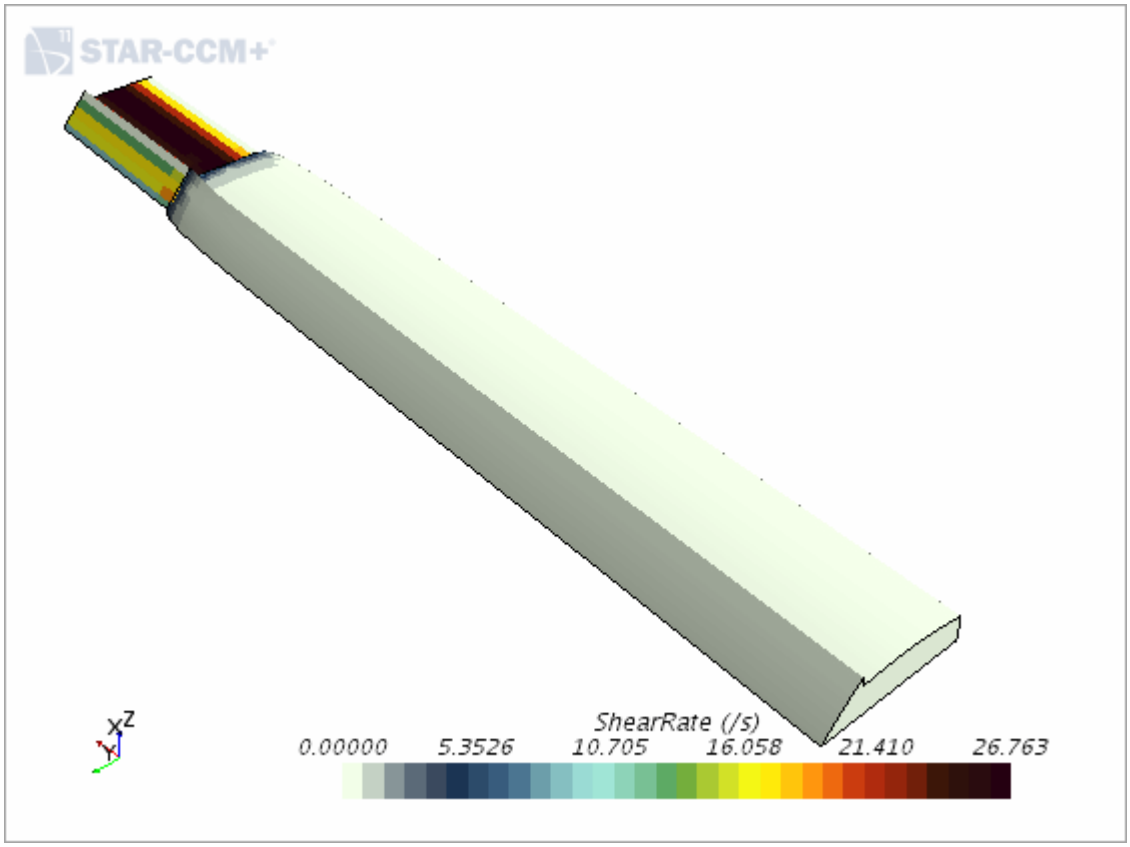
5. Save the simulation as `ExtrusionDie-2`.

Running the Viscoelastic Simulation

Switching to the Viscoelastic model improves the realism of the simulation. Run the simulation again and view the change to the vector scene.

1. Select **Solution** > **Clear Solution** and click **OK** in the dialog.
2. Click **Run** .
3. Re-examine the Velocity scene and the Shear Rate scene. Notice the change in velocity in the extrusion and the increase in swelling at the die exit.




STAR-CCM+
ShearRate (/s)
0.00000
5.3526
10.705
16.058
21.410
26.763

Compressible Flow

The tutorials in this set illustrate various STAR-CCM+ features for compressible fluid flows as well as turbo wizard and harmonic balance

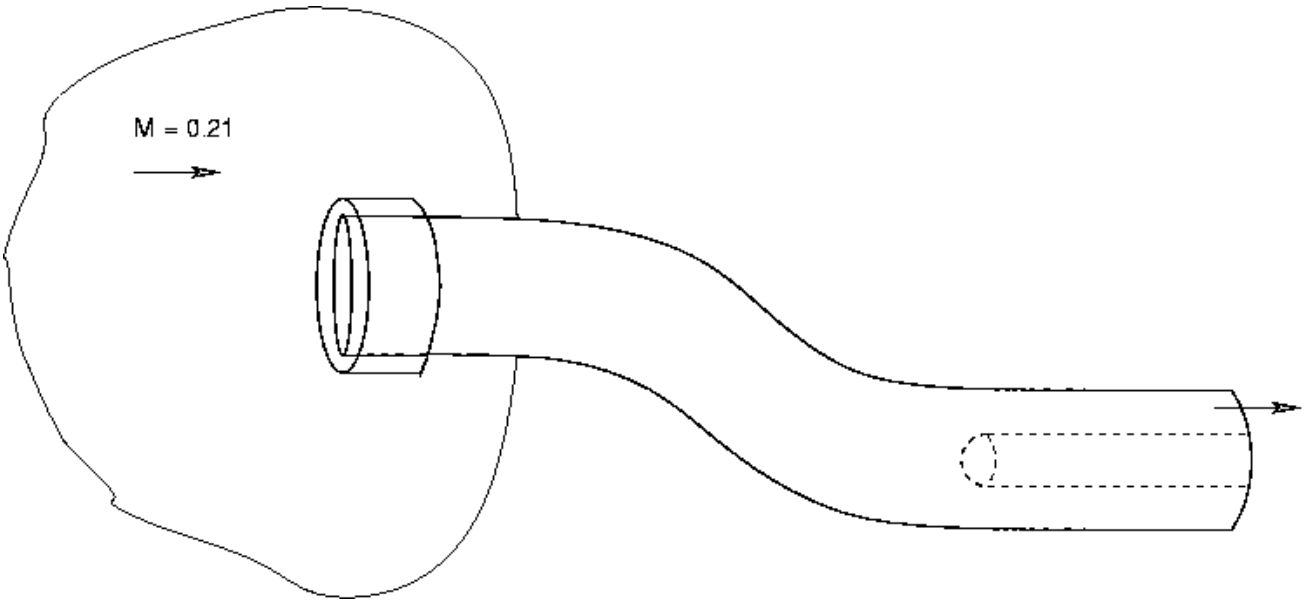
Contents:

Subsonic Flow: NACA-Type Intake
Transonic Flow: RAE2822 Airfoil
Transonic Flow: RAE2822 Airfoil Using Overset Mesh
Adjoint Flow Solver: Flow through an S-Bend
Turbo Wizard: Single Row
Turbo Wizard: Multiple Rows
Harmonic Balance: Single Stage Periodic Flow

Subsonic Flow: NACA-Type Intake

This tutorial illustrates how to solve a three-dimensional compressible flow problem using STAR-CCM+.

The geometry in question is a NACA-type axisymmetric inlet in a freestream air flow of Mach 0.21. The flow that is captured by the inlet is forced into an S-bend before reaching the engine face.



Caution: To keep the cell count low, certain compromises have been made. Even though the case is run turbulent for this exercise, the mesh is more suited for inviscid flow. Because no attempt is made to resolve the boundary layers, the results are thus essentially what one would obtain from an inviscid simulation. This practice is not recommended for most simulations.

To illustrate how STAR-CCM+ “fuses” separate mesh sections into a contiguous mesh, the mesh has been created in three separate non-conformal sections.

In spite of the fact that the mesh is too coarse to produce a mesh-independent solution, the simulation results are shown to compare favorably with experimental data for pressure on the S-duct walls.

The mesh that is used for this intake example is based on an S-duct inlet mesh from NASA.

Contents:

Importing the Mesh and Naming the Simulation
Visualizing the Mesh
Manipulating Regions and Boundaries
Setting Up the Physics Models
Setting Fluid Properties
Setting Boundary Conditions
Setting Initial Conditions

Adding Field Functions
Setting Stopping Criteria
Setting Solver Parameters
Running the Simulation
Adjusting Solver Parameters and Continuing
Visualizing the Mach Number
Visualizing the Velocity Vectors
Setting Up a Line Surface for Plotting
Plotting Simulation Data
Plotting Reference Data
Closing and Restarting
Summary

Importing the Mesh and Naming the Simulation

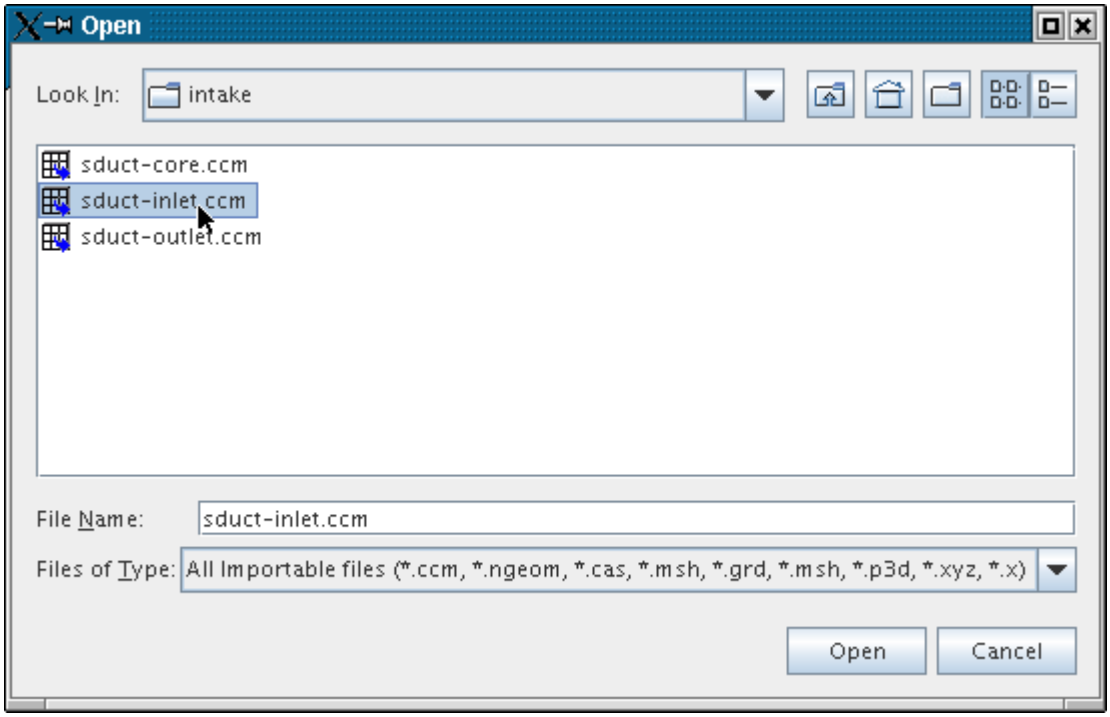
Create a new simulation, import the supplied mesh, and name the simulation `intake`.

Use the following steps:

1. Start up STAR-CCM+ and select the **New Simulation** option from the menu bar.

   Continue by importing the mesh and naming the simulation. The mesh has been created in three separate blocks, and stored in separate `.ccm` files.

2. Select **File** > **Import** > **Import Volume Mesh...** .
3. In the *Open* dialog, navigate to the `doc/startutorialsdata/compressibleFlow/data` subdirectory of your STAR-CCM+ installation directory.



4. Multi-select files `sduct-inlet.ccm`, `sduct-core.ccm`, and `sduct-outlet.ccm`.
5. To start the import, click **Open**.
6. To create a file for the new simulation, select **File** > **Save**. Navigate to the directory where you want the file located.
7. In the *Save* dialog, type `intake` into the **File Name** text box.
8. Click **Save**.

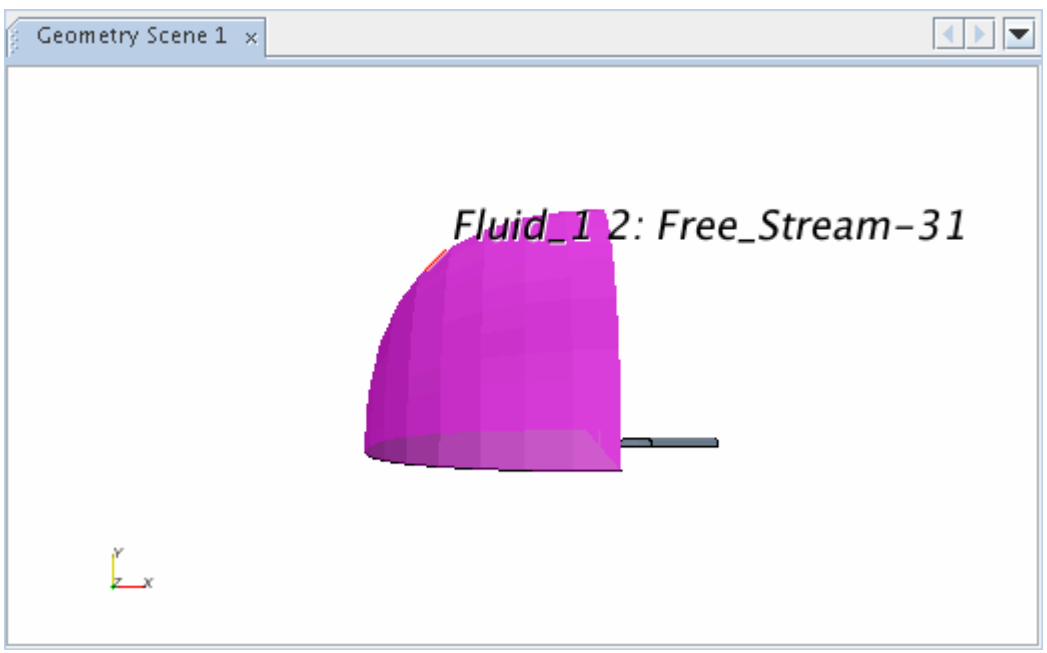
Visualizing the Mesh

The *Geometry Scene 1* display is present in the **Graphics** window. Set the display to show the mesh.

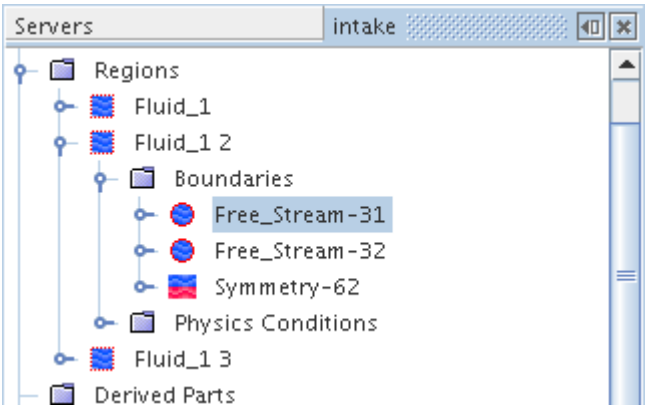
Initially all parts of the mesh are shown as solid, colored surfaces.

1. Click one of the parts.

   The object (a boundary in the illustration below) becomes highlighted and a label appears with the name of the object selected.

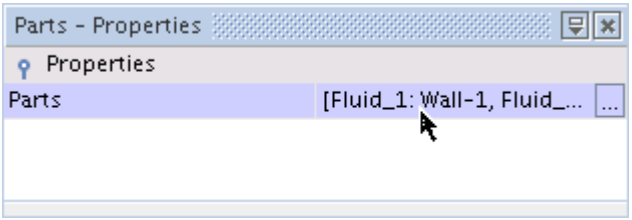


   In the **intake** window, the node that corresponds to this object is highlighted.

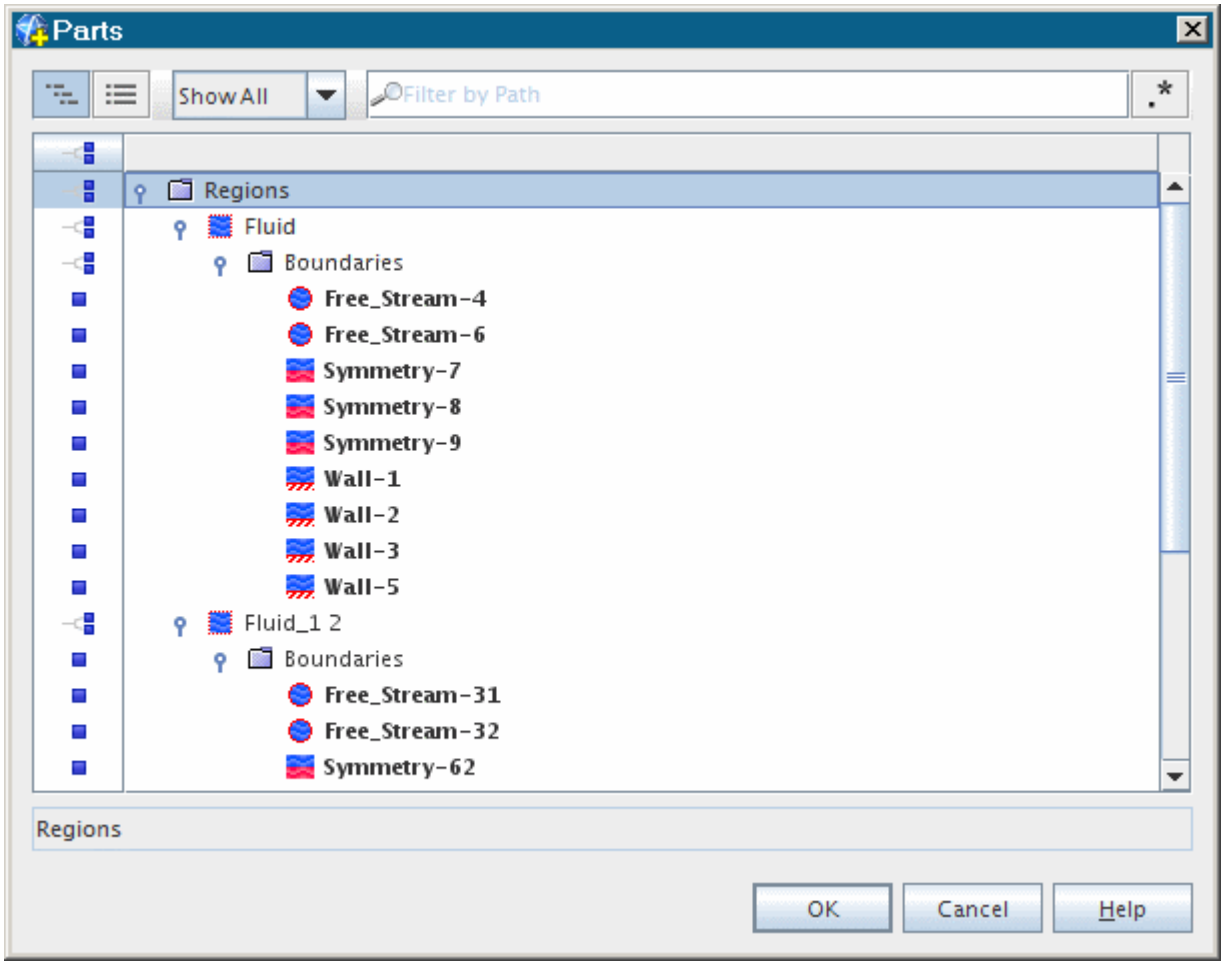


Similarly, if a different boundary node is selected, the surface corresponding to that boundary is highlighted in the **Graphics** window.

2. Select the **Scenes** > **Geometry Scene 1** > **Displayers** > **Geometry 1** > **Parts** node and edit the *Parts* property.

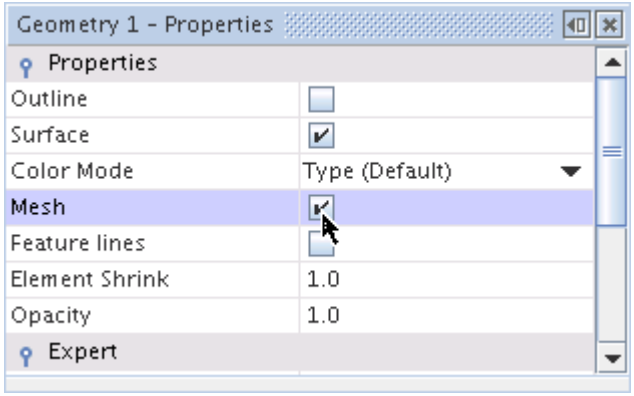


3. Expand all the nodes under the **Regions** node.

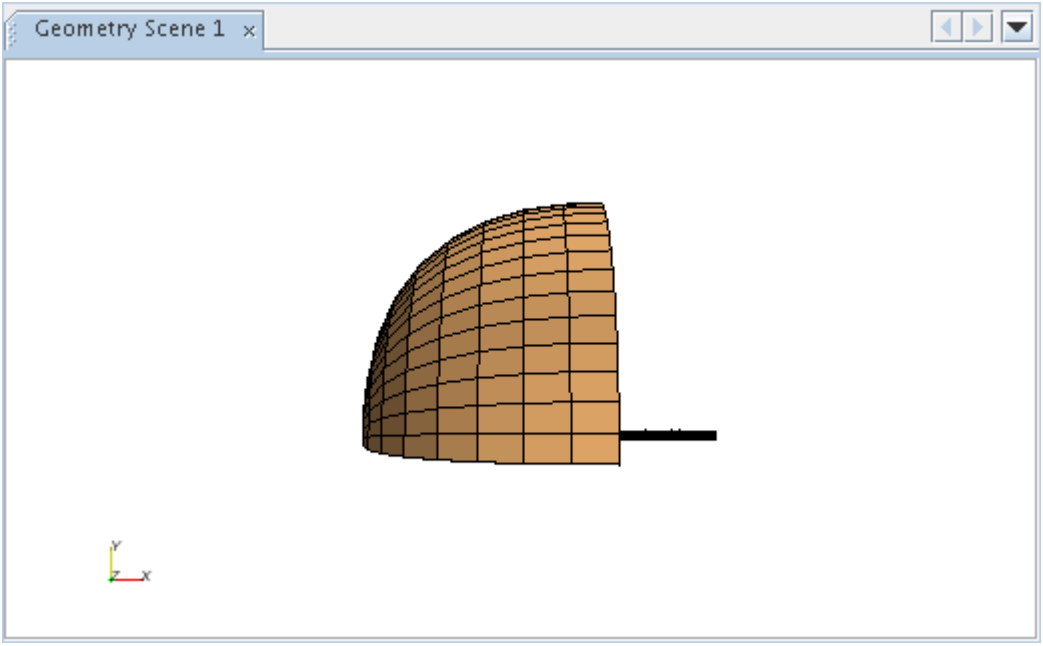


This dialog contains a list of the parts to be displayed. By default, all boundary surfaces are displayed.

4. Click **Cancel**.
5. To display the surface mesh on the selected parts, select the **Geometry 1** node and activate *Mesh*:



The surface mesh appears as shown below:



Manipulating Regions and Boundaries

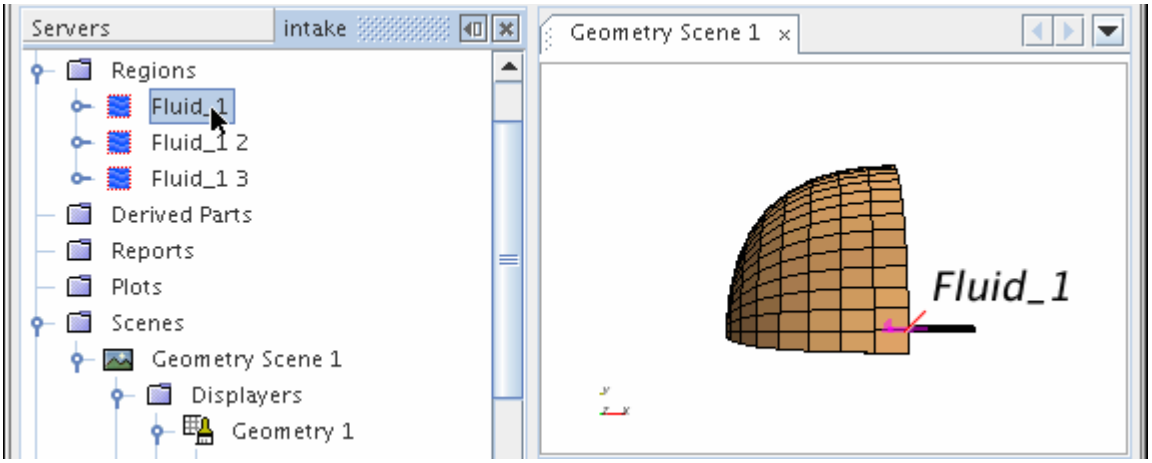
Combine the separate mesh blocks into a single contiguous mesh, then rename the mesh entities before continuing with the simulation.

Use the following steps:

1. In the **intake** window, open the **Regions** node.

   See three separate regions, corresponding to the three mesh blocks that were imported.
2. Click to select each of the region nodes (**Fluid_1**, **Fluid_1 2**, and **Fluid_1 3**) in turn.

   The selected region is highlighted and labeled in the *Geometry Scene 1* display.



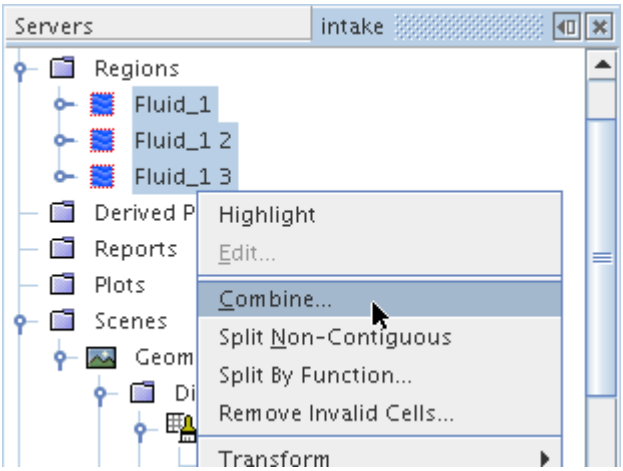
There are two alternative approaches for combining these regions so that information can propagate between them:

- Create internal interfaces.
- Fuse the regions and boundaries.

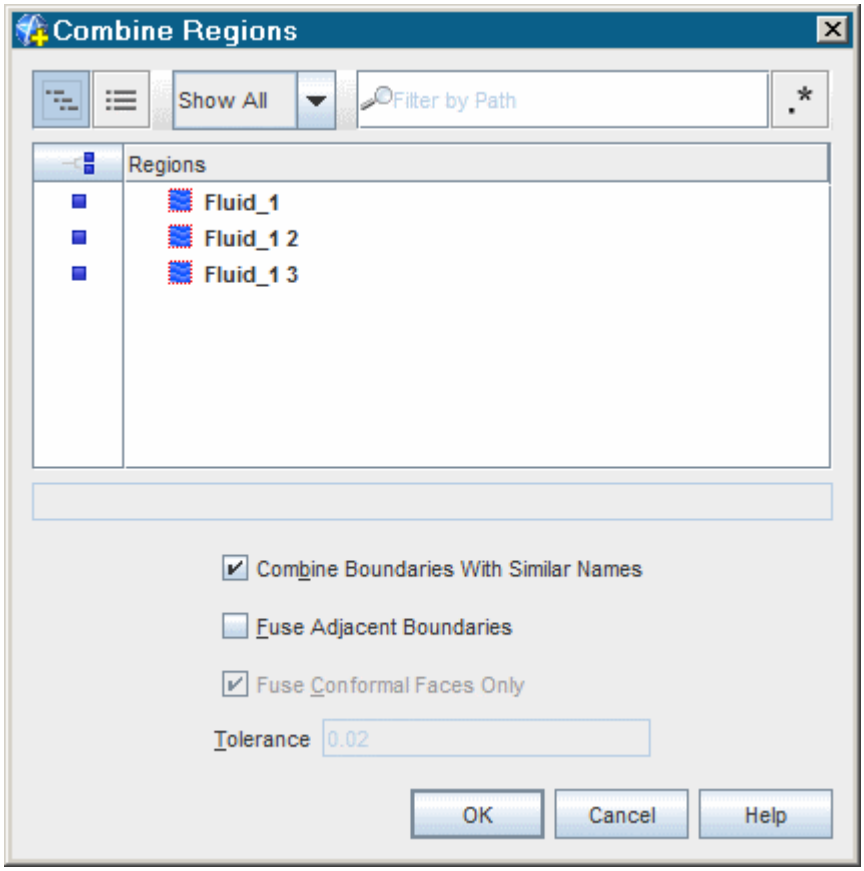
Using internal interfaces is advantageous if the mesh is to be manipulated further at a later stage, since interfaces can easily be deleted and created. However, there is likely to be some (most likely small) computational overhead that is associated with the interface. In this case, since the mesh does not need to be changed, combine the regions and fuse the adjacent boundaries, creating a single contiguous mesh. The process of “fusing” creates a (possibly non-conformal) interface, and then merges that interface into the mesh.

To combine the separate regions into one, use **Shift Click** to select all three region nodes.

3. Right-click and select **Combine...** .



4. In the *Combine Regions* dialog, make sure that all regions under it are selected.

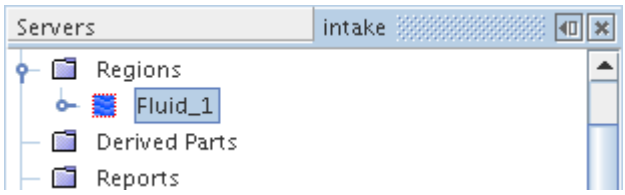


5. Activate **Combine boundaries with similar names**.

In many situations, the **Fuse adjacent boundaries** option can also be used. However, in this case, the specified tolerance is not suitable for all the boundaries to be fused, so complete the fusing operation later.

6. Click **OK** to combine the regions without fusing the boundaries.

   As a result of this operation, a single region node remains.



Since there is only one region now, you can simplify the region name.

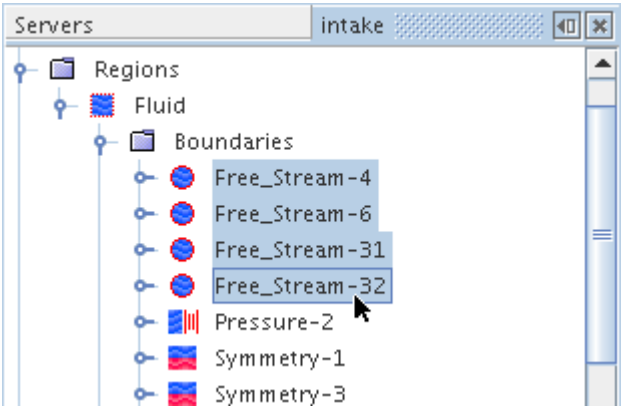
7. Rename the **Fluid_1** node to `Fluid`.

Now that the regions have been combined, fuse the adjacent boundaries that are currently separating the regions. The mesh is not conformal at these boundaries, so a non-conformal intersection needs to be computed as part of the fusing process. Essentially, an interface is created and then merged into the mesh.

The intersection process requires the specification of a tolerance. Use a value of 0.02 (default 0.05) and then perform checks to make sure that this is suitable.

Before creating interfaces, the freestream boundary types must be redefined as walls. This allows the solution to be initialized without activating a flow model, normally a requirement of freestream boundaries.

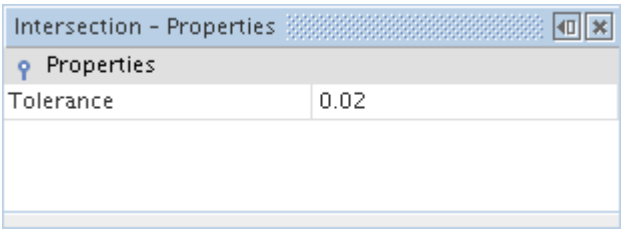
8. Open the **Fluid** > **Boundaries** node.
9. Select all the freestream boundary nodes: **Free_Stream-4**, **Free_Stream-6**, **Free_Stream-31** and **Free_Stream-32** and set the *Type* to **Wall**.



This bypasses the need to activate any models before initializing, the latter being a necessary step for testing the intersecting interface tolerance.

10. Select the **Free_Stream-4** node, then use the `Ctrl Click` approach to select the **Free_Stream-32** node.

    (It is important to select the boundaries in this order.)
11. Right-click and select **Create Interface**.
12. Open the **Interfaces** > **Interface 1** > **Physics Values** node, and select the **Intersection** node and set the *Tolerance* to `0.02`.

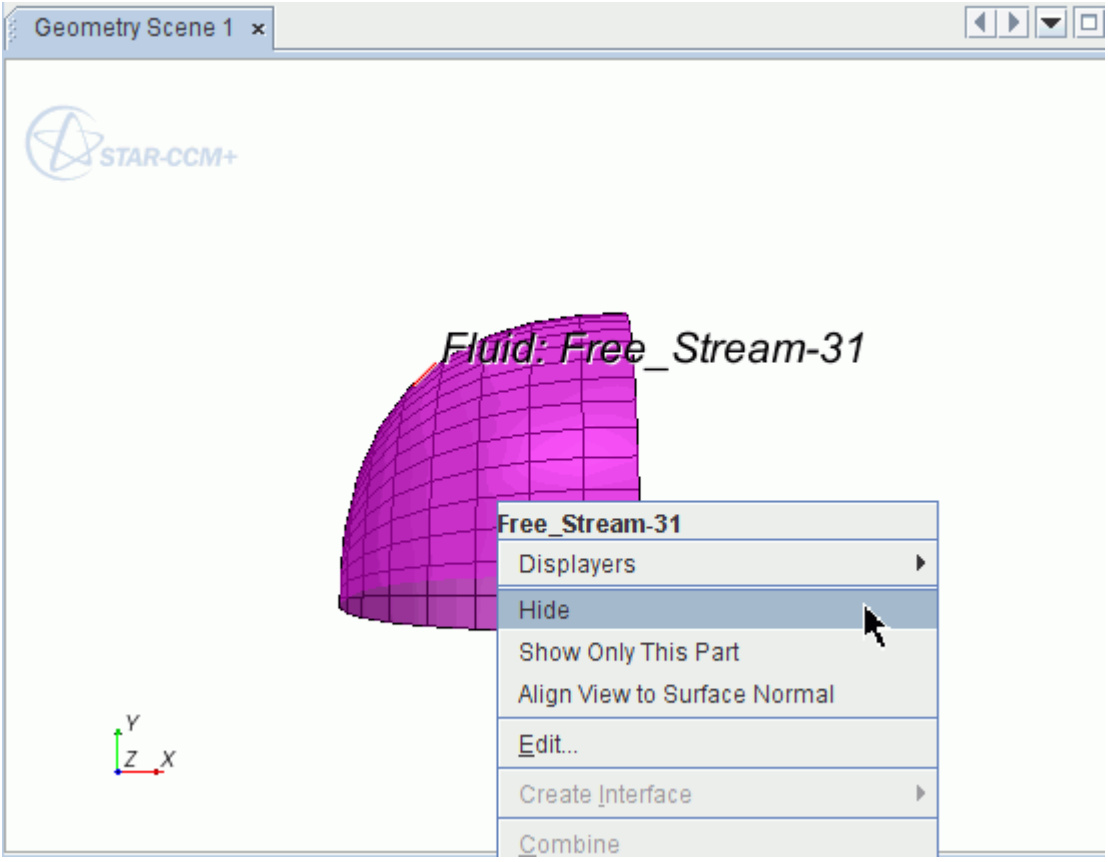


You are now ready to initialize the solution manually.

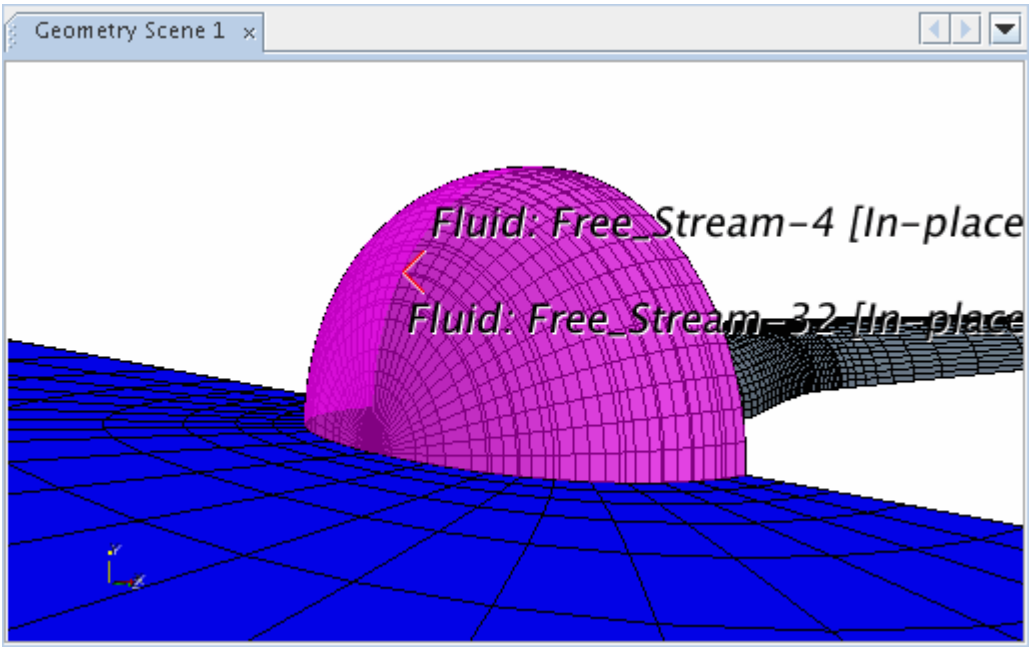
13. Click (**Initialize Solution**) on the toolbar or select **Solution** > **Initialize Solution**.

To determine whether the test was successful, inspect boundaries **Free_Stream-4 [Interface 1]** and **Free_Stream-32 [Interface 1]** in the *Geometry* scene.

14. To gain access to them, first locate the **Free_Stream-31** part by clicking its node.
15. Right-click that part in the scene and select **Hide**.

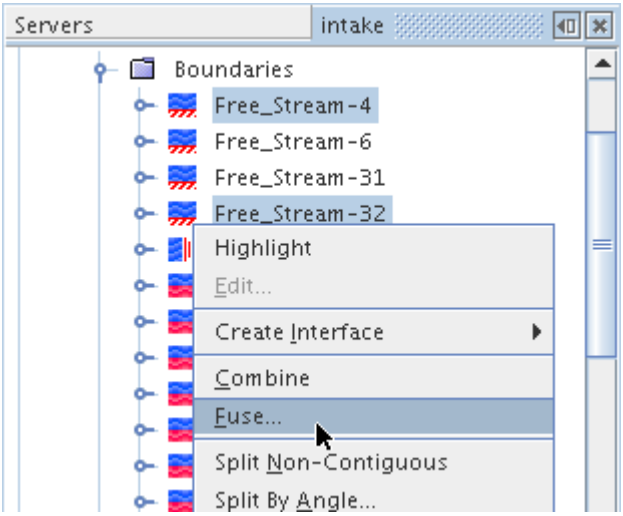


16. Next, select the nodes of the two internal interface boundaries, then click the (**Reset View**) button on the toolbar to bring them into focus. (Zoom and rotate if necessary.)



The parts appear intact, also showing that the interface was intersected correctly. This shows that the tolerance value of `0.02` is the right one. Delete the interface so that the boundaries can be fused.

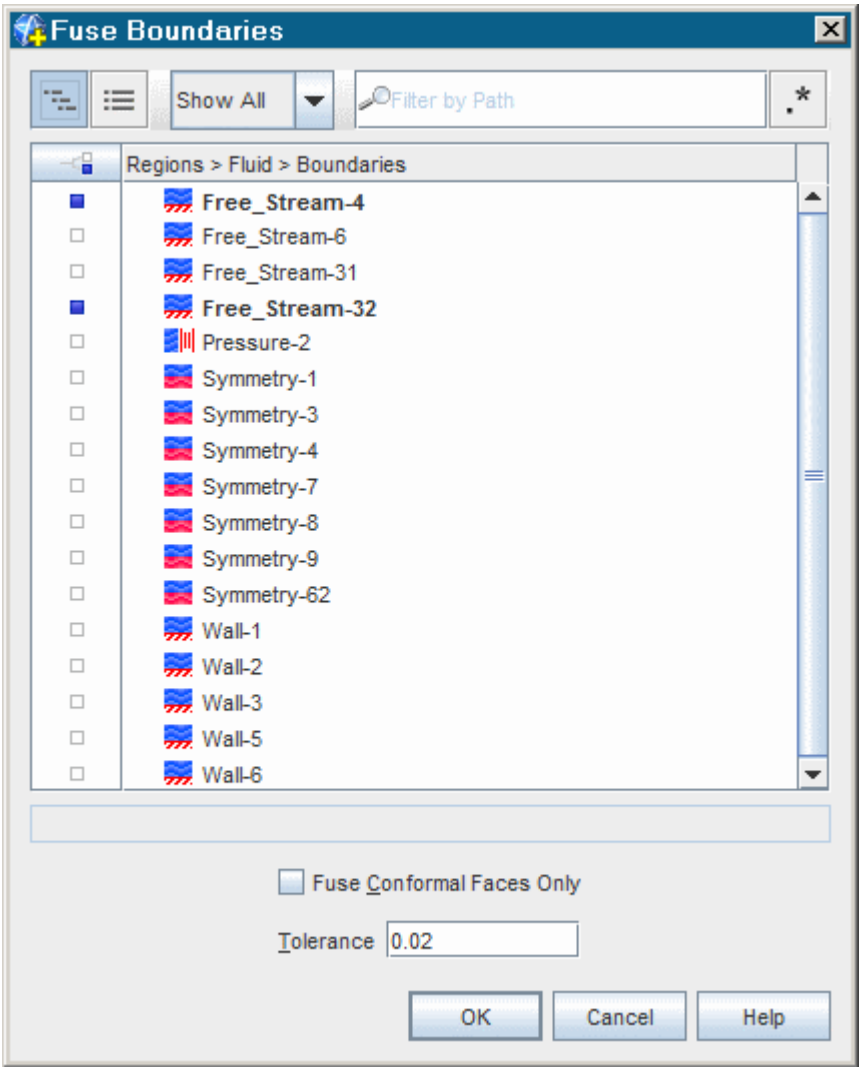
17. Right-click the **Interfaces** > **Interface 1** node and select **Delete**.
18. Click **Yes** in the confirmation dialog.
19. Select the nodes of the **Free_Stream-4** and **Free_Stream-32** boundaries (in that order), right-click and select **Fuse**.



The sequence of boundary selection can affect the results of the fuse operation. When you fuse boundaries, STAR-CCM+ attempts to take the second boundary selected by you and project it onto the first.

Hence, when deciding on the sequence of boundaries to select for fusing, always select the one with the finer mesh grid first. This instructs STAR-CCM+ to project the coarser boundary onto the finer one, leading to better results in the mesh grid. If you were to select the boundaries in the opposite order (trying to project the finer boundary onto the coarser one), this may essentially fail, as you may see only parts of the fused boundaries shown in the scene display.

20. In the *Fuse Boundaries* dialog, make sure that these boundaries are selected:



21. Click **OK**.

    Both these boundaries disappear from the tree, since fusing boundaries incorporates them into the interior mesh grid.

22. Repeat the fusing process for the **Symmetry-1** and **Symmetry-7** boundaries. Again, use the default tolerance, `0.02`.

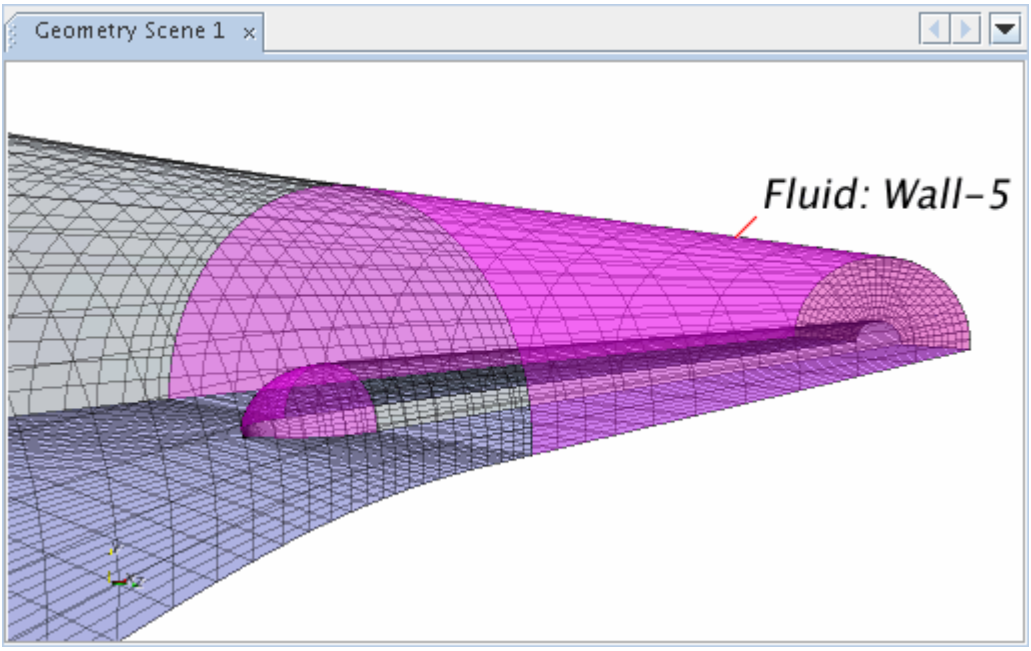
23. Combine (do not fuse) the **Free_Stream-31** and **Free_Stream-6** boundaries by selecting their nodes, right-clicking, and selecting **Combine**.

    This creates a single freestream boundary entity.

24. Select all the remaining symmetry plane nodes, right-click, and select **Combine**.

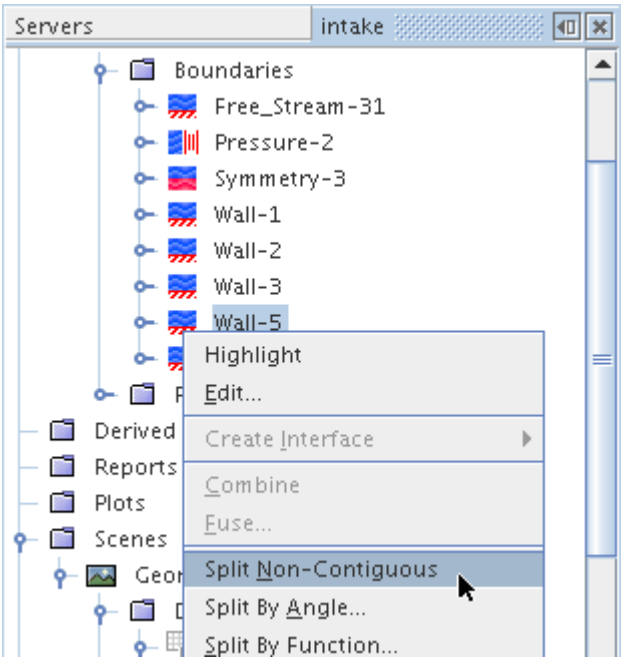
25. Select the **Wall-5** node and look in the *Geometry* scene.

26. To see the part more clearly, click (**Make Scene Transparent**).

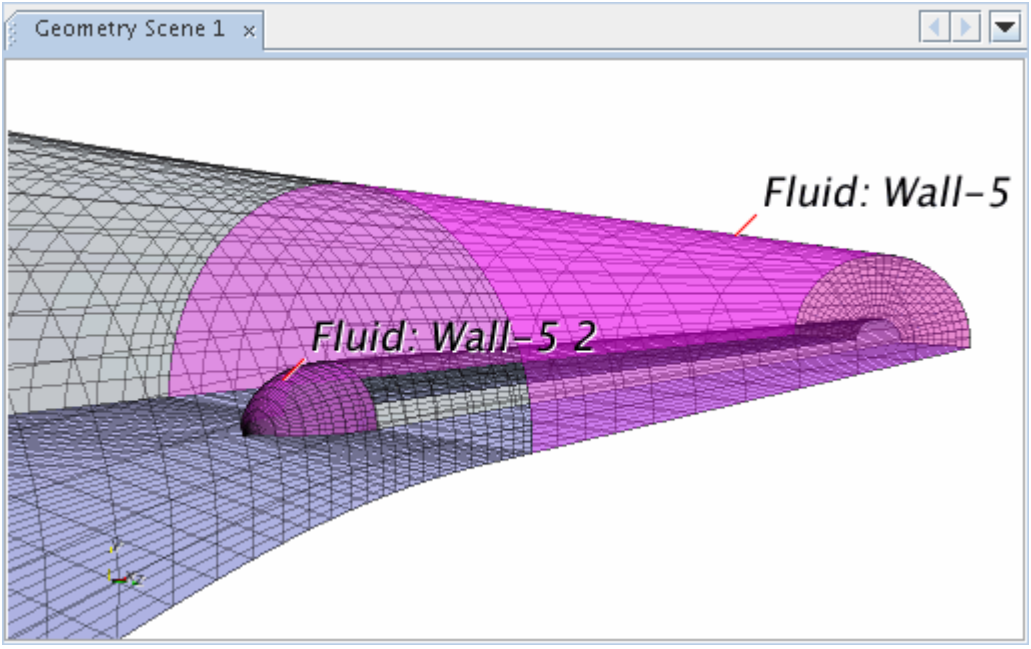


This boundary includes two non-contiguous surfaces.

27. To correct this, right-click the **Wall-5** node and select **Split Non-Contiguous**.



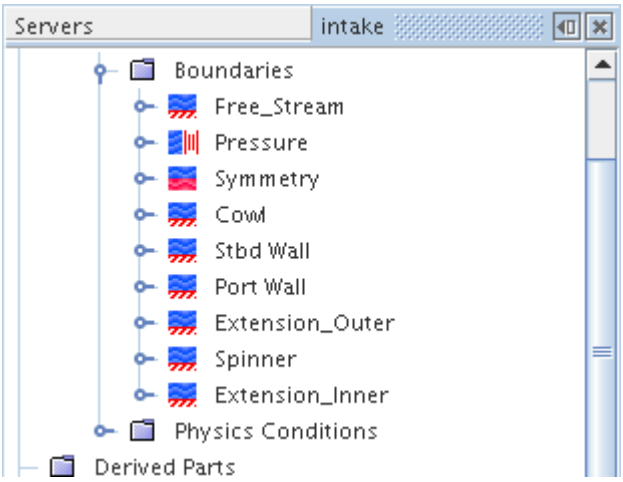
28. Select the **Wall-5** node and then the new **Wall-5 2** node separately. The split action has made them into two separate boundaries.



29. Rename the following nodes:

Node	New Name
Free_Stream-31	`Free_Stream`
Pressure-2	`Pressure`
Symmetry-3	`Symmetry`
Wall-1	`Cowl`
Wall-2	`Stbd Wall`
Wall-3	`Port Wall`
Wall-5	`Extension_Outer`
Wall-5 2	`Spinner`
Wall-6	`Extension_Inner`

This portion of the simulation tree appears as shown below:



30. Save the simulation.

Setting Up the Physics Models

Models define the primary variables of the simulation, including pressure, temperature and velocity, and what mathematical formulation is used to generate the solution.

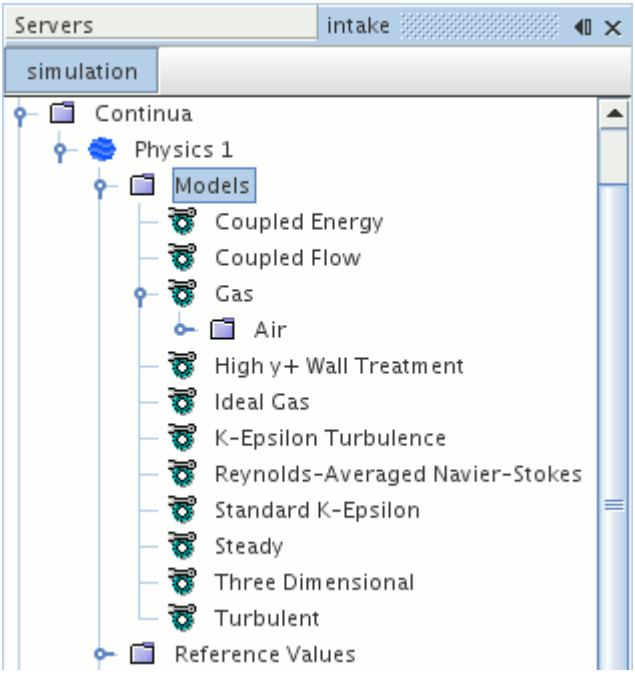
In this example, the flow is turbulent and compressible. The Coupled Flow Model is used together with the default K-Epsilon Turbulence model.

1. For the physics continuum **Continua** > **Physics 1**, deactivate **Auto-select recommended models** and select the following models, in order:

Group Box	Model
Material	**Gas**
Flow	**Coupled Flow**
Equation of State	**Ideal Gas**
Energy	**Coupled Energy**
Time	**Steady**
Viscous Regime	**Turbulent**
Turbulence	**Reynolds-Averaged Navier-Stokes**
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**
K-Epsilon Turbulence Models	**Standard K-Epsilon**
K-Epsilon High y+ Wall Treatment	**High y+ Wall Treatment**

2. Click **Close**.
3. In the **intake** window, open the **Continua** > **Physics 1** > **Models** node.

   The selected models now appear within that node.

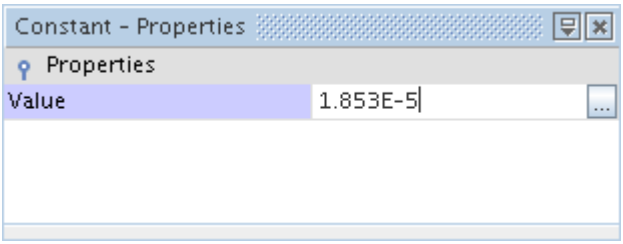


4. Save the simulation.

Setting Fluid Properties

Since the Ideal Gas model has been chosen, the viscosity, specific heat, and thermal conductivity fluid properties must be set.

1. Select the **Continua** > **Physics 1** > **Models** > **Gas** node, then select the **Air** > **Material Properties** > **Dynamic Viscosity** > **Constant** node, and set *Value* to `1.853E-5 Pa-s`.



2. Within the same **Air** node, select the **Specific Heat** > **Constant** node and set *Value* to `1005 J/kg-K`.

The thermal conductivity can be left at its default value of `0.0260305 W/m-K`.

Setting Boundary Conditions

In this simulation, the freestream Mach number is 0.21, the freestream pressure is 98,258.18 Pa, and the freestream temperature is 300 K. Use the default reference pressure of 101,325 Pa, which means that the freestream gauge pressure is -3066.82 Pa.

1. Select the **Regions** > **Fluid** > **Boundaries** > **Free_Stream** node.
2. Set the *Type* property back to **Free Stream**.
3. In the **Boundaries** > **Free_Stream** > **Physics Values** node, set the following properties:

Node	Property	Setting
Mach Number	*Value*	`0.21`
Pressure	*Value*	`-3066.82 Pa`

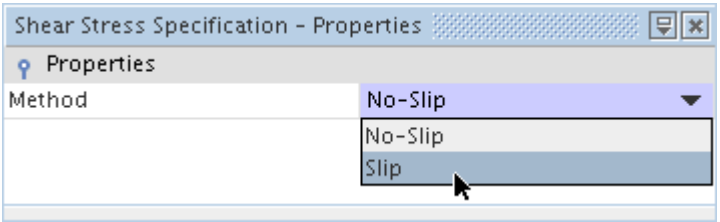
The gauge pressure at the duct outlet is specified as -7903.35 Pa.

4. Under the **Boundaries** > **Pressure** > **Physics Values** node, select **Pressure** and set *Value* to `-7903.35 Pa`.

The mesh in the section downstream of the engine face is coarse. For this reason, no attempt is made to resolve the boundary layer. Instead of the usual no-slip condition, a slip wall is specified on the inner and outer walls.

5. In the **Boundaries** node, set the following properties:

Node	Property	Setting
Extension_Inner		
└ **Physics Conditions**		
└ **Shear Stress Specification**	*Method*	**Slip**
Extension_Outer		
└ **Physics Conditions**		
└ **Shear Stress Specification**	*Method*	**Slip**



Setting Initial Conditions

In compressible fluid problems, it is advisable to initialize the flow to a non-zero initial velocity and a meaningful pressure. Here, you use the freestream velocity and the duct outlet pressure.

1. Open the **Physics 1** > **Initial Conditions** node and set the following properties:

Node	Property	Setting
Pressure	*Value*	`-7903.35`
Velocity	*Value*	`72,0,0`

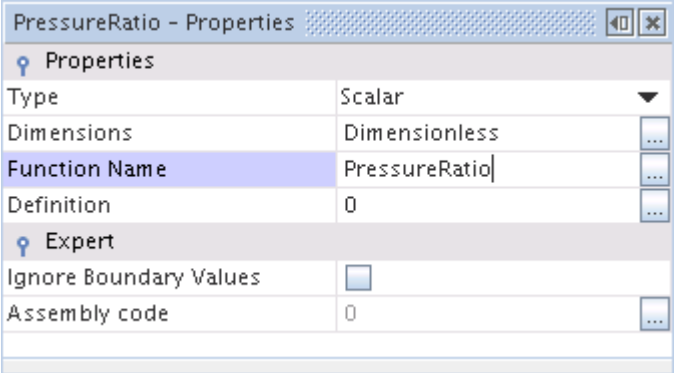
2. Save the simulation.

Adding Field Functions

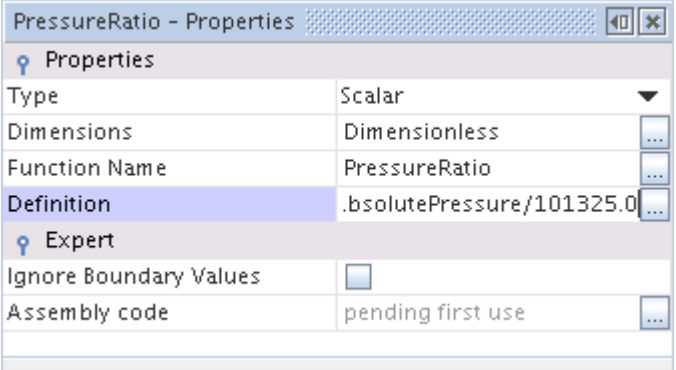
Field functions provide a mechanism by which fields (raw data from the simulation stored in cells and/or on boundaries) may be viewed and defined.

User field functions are field functions that you create to access field data. This section illustrates how to create a field function for post-processing purposes that represents the ratio of absolute pressure to atmospheric pressure.

1. Open the **Tools** node, right-click the **Field Functions** node and select **New** > **Scalar**.
2. Rename the new field function node to `PressureRatio`.
3. Select the **PressureRatio** node and set *Function Name* to `PressureRatio`.



4. Set *Definition* to `$AbsolutePressure/101325.0`



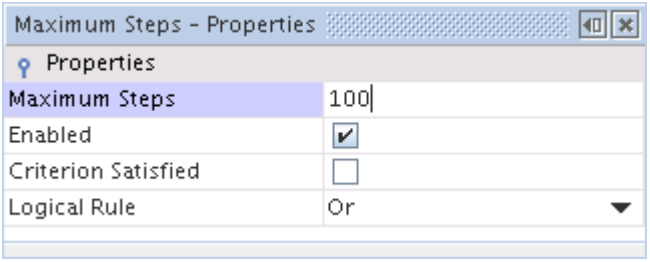
This entry instructs your new field function to reference the Absolute Pressure field function, and states that the atmospheric pressure value is 101,325.0 Pa.

Setting Stopping Criteria

Stopping criteria allow you to specify the length of the simulation, in this case the number of iterations to be performed.

To set a stopping criterion:

1. Open the **Stopping Criteria** node and select the **Maximum Steps** node.
2. Set the *Maximum Steps* to `100`.

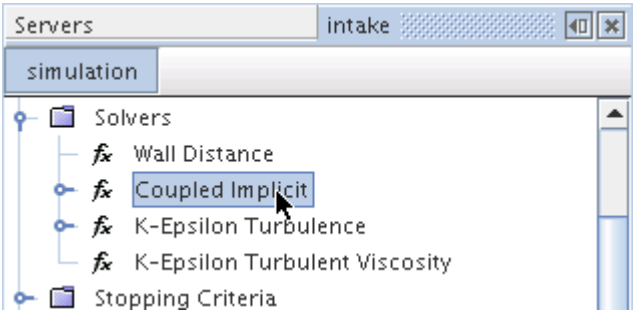


Setting Solver Parameters

In this simulation, the Coupled Flow model is controlled by the Coupled Implicit solver.

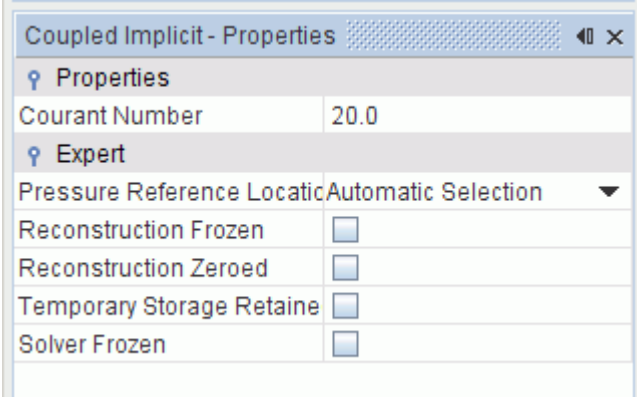
This solver enables the local iteration step to be adjusted in terms of the Courant number. The default value of 5 is conservative for a low Mach number problem on a mesh of reasonable quality, so it can be increased to speed convergence.

1. Select the **Solvers** > **Coupled Implicit** node.



2. Set the *Courant Number* to `20`.

This is sufficiently low for the flow field to become established. After 100 iterations, increase the Courant number further to allow a converged solution to be reached more rapidly.



3. Save the simulation.

All necessary material properties, boundary conditions, field functions, and solver parameters for this tutorial have been applied, so the simulation can now be run.

Running the Simulation

Run the simuation and examine the Residuals display.

1. To run the simulation, click the (**Run**) button in the toolbar.

   If you do not see this button, use the **Solution** > **Run** menu item. You could also show the **Solution** toolbar by selecting **Window** > **Toolbars** > **Solution**, and then clicking the toolbar button.

The Residuals display is created automatically and shows the progress of the solver. If necessary, click the **Residuals** tab to bring the Residuals plot into view.

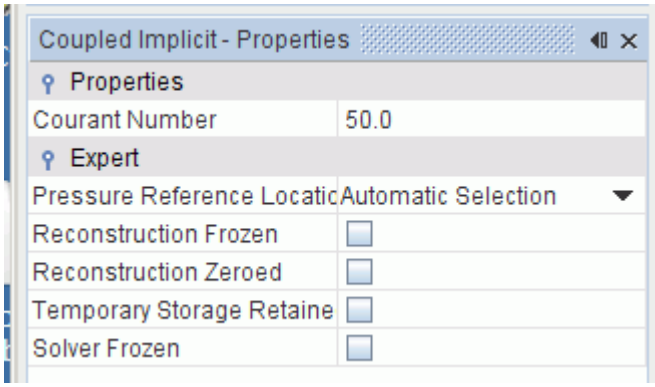
During the run, it is possible to stop the process by clicking the (**Stop**) button on the toolbar. If you do halt the simulation, it can be continued later by clicking the (**Run**) button. If left alone, the simulation continues until the stopping criterion of 100 iterations is satisfied.

2. Save the simulation after the run has finished.

Adjusting Solver Parameters and Continuing

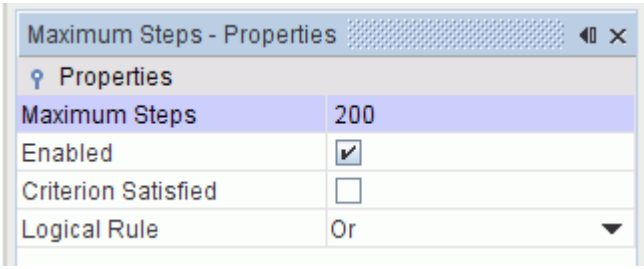
Now that the flow field has become established and the solution is converging well, the Courant number can be further increased.

1. Select the **Solvers** > **Coupled Implicit** node.
2. Set the *Courant Number* to `50`.



3. Select the **Stopping Criteria** > **Maximum Steps** node.
4. Set the *Maximum Steps* to `200`

   This instructs STAR-CCM+ to iterate for a further 100 steps in addition to those already performed.



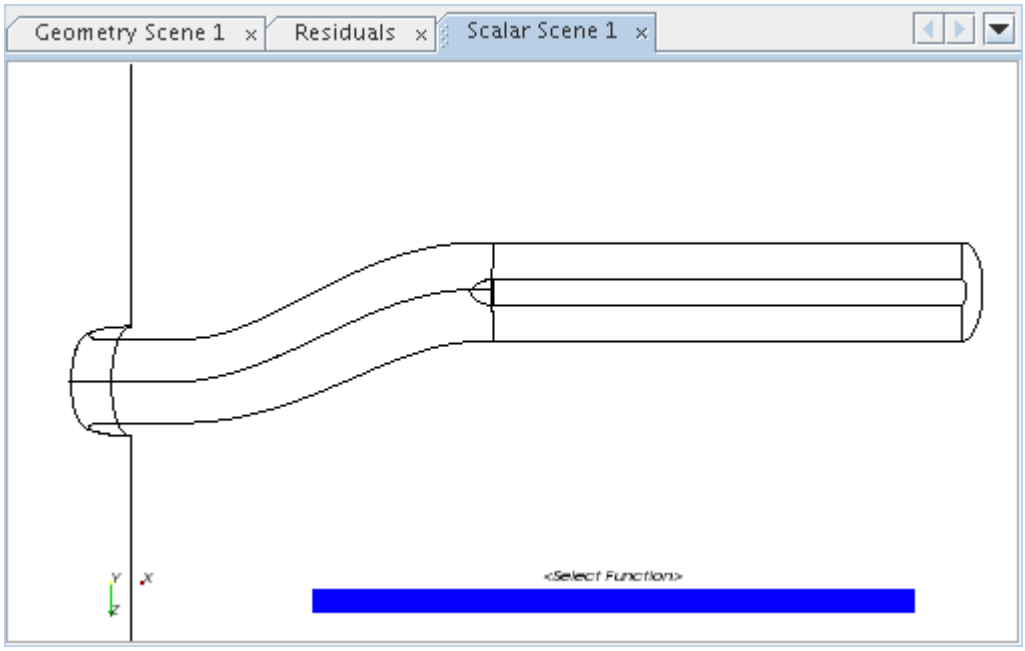
5. Click (**Run**) to continue running the simulation.
6. Once the run is finished, save the simulation.

Visualizing the Mach Number

As an introduction to data visualization, this section explains how to examine the Mach number field.

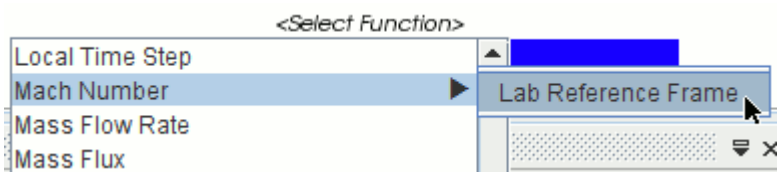
1. Right-click the **Scenes** node and select **New Scene** > **Scalar**.
2. For a clearer view of the model, click (**Save-Restore-Select Views**) on the toolbar. Select **Views** > **+Y** > **Up -Z**.

   This aligns the model along the chosen axes. You can now zoom and pan to adjust the view further.

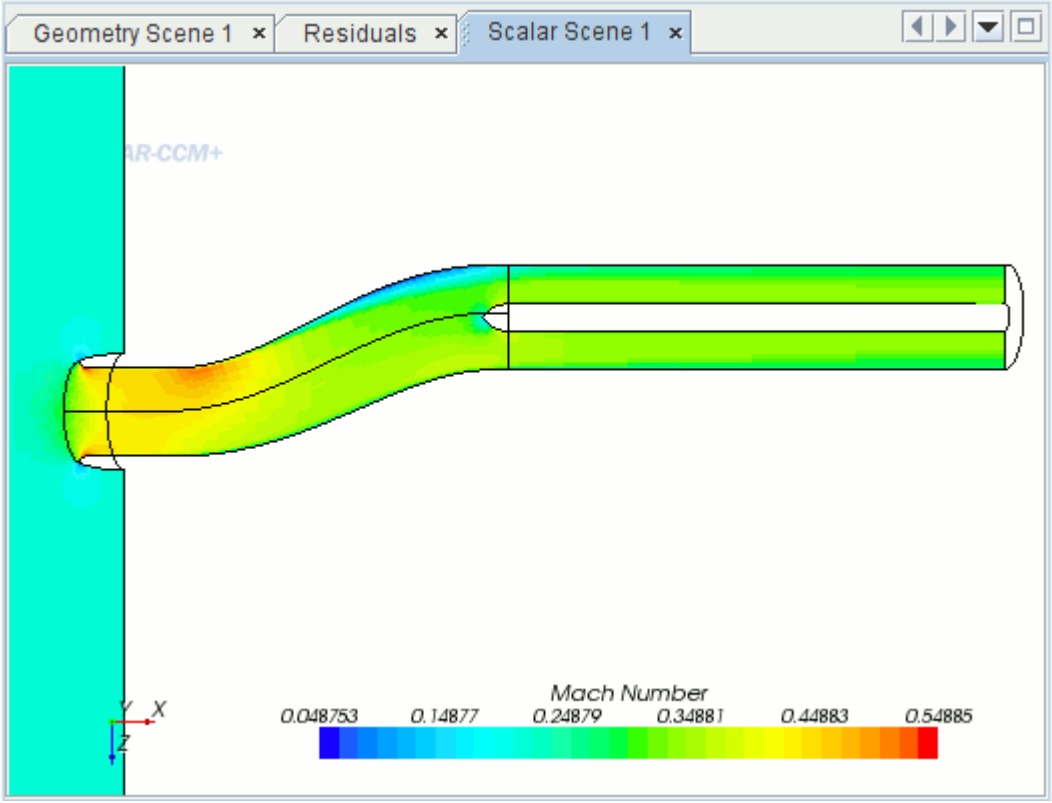


3. In the simulation tree, select the **Scenes** > **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Parts** node and edit the *Parts* property.

   The *Parts* dialog appears.
4. Select the **Symmetry** object.
5. Click **OK**.
6. Right-click the scalar bar in the display. In the pop-up menu that appears, select **Mach Number** > **Lab Reference Frame**.



The result appears in the *Scalar Scene 1* display.



By default, **Filled** cell values are shown.

7. Select the **Scalar 1** node and set the *Contour Style* to (**Smooth Filled**).

The contours of the scalar display now appear smooth.

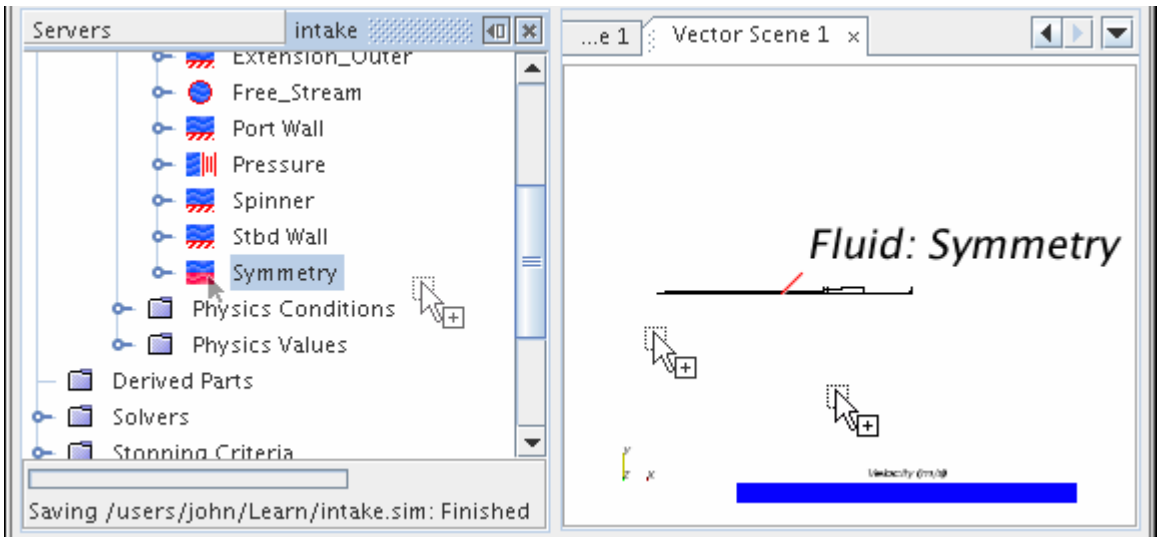
8. Save the simulation.

Visualizing the Velocity Vectors

This section describes how to examine velocity vectors.

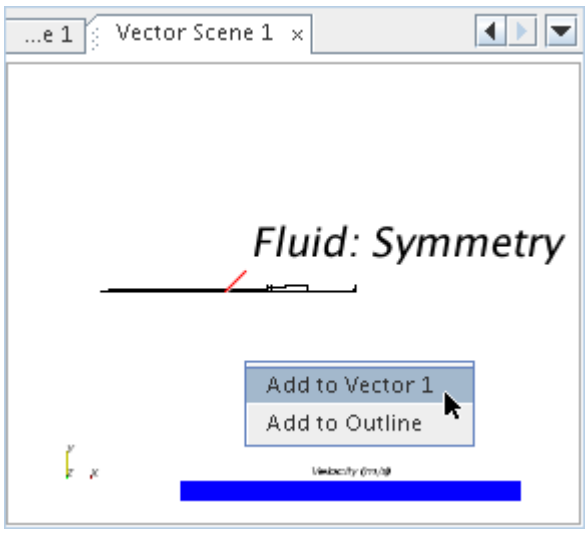
Use the following steps to produce a vector scene:

1. Right-click the **Scenes** node. Selecting **New Scene** > **Vector** causes a *Vector Scene 1* display to appear.
2. Open the **Regions** > **Fluid** > **Boundaries** node, and then drag the **Symmetry** node into the display.



When your mouse pointer arrives on the display, a pop-up menu appears letting you decide the part displayer to receive the new part.

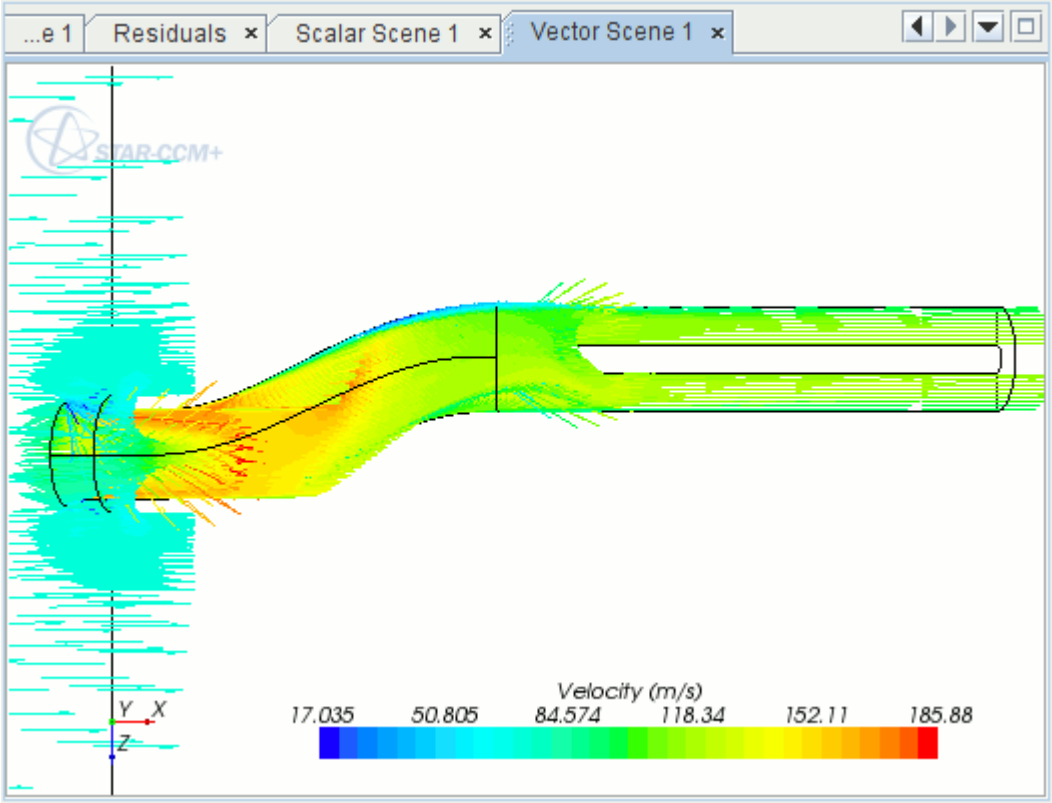
3. Select **Add to Vector 1**.



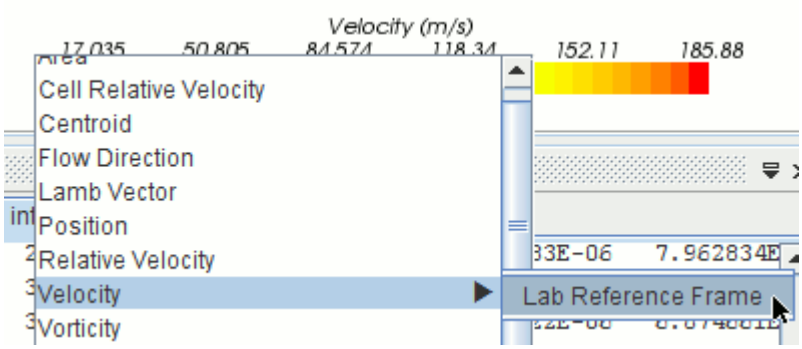
At this point, the part is in the scene, though it is difficult to see from the default view.

4. For a clearer view of the model, click (**Save-Restore-Select Views**) on the toolbar to activate the **Save-Restore-Select Views** menu.
5. In that menu, select **View** > **+Y** > **Up -Z**. You can also zoom and pan the view as needed.

   The result appears in the *Vector Scene 1* display.

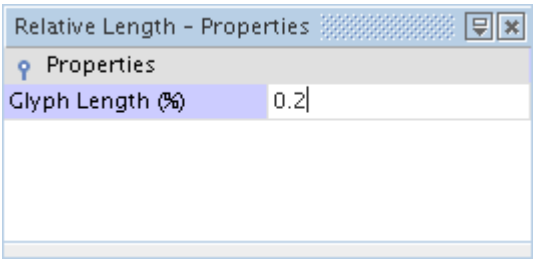


Velocity is the default selection for vector displays. However, if you do not see any vectors after adding a part to the scene, right-click the vector bar in the display. In the pop-up menu that appears, select **Velocity** > **Lab Reference Frame**.

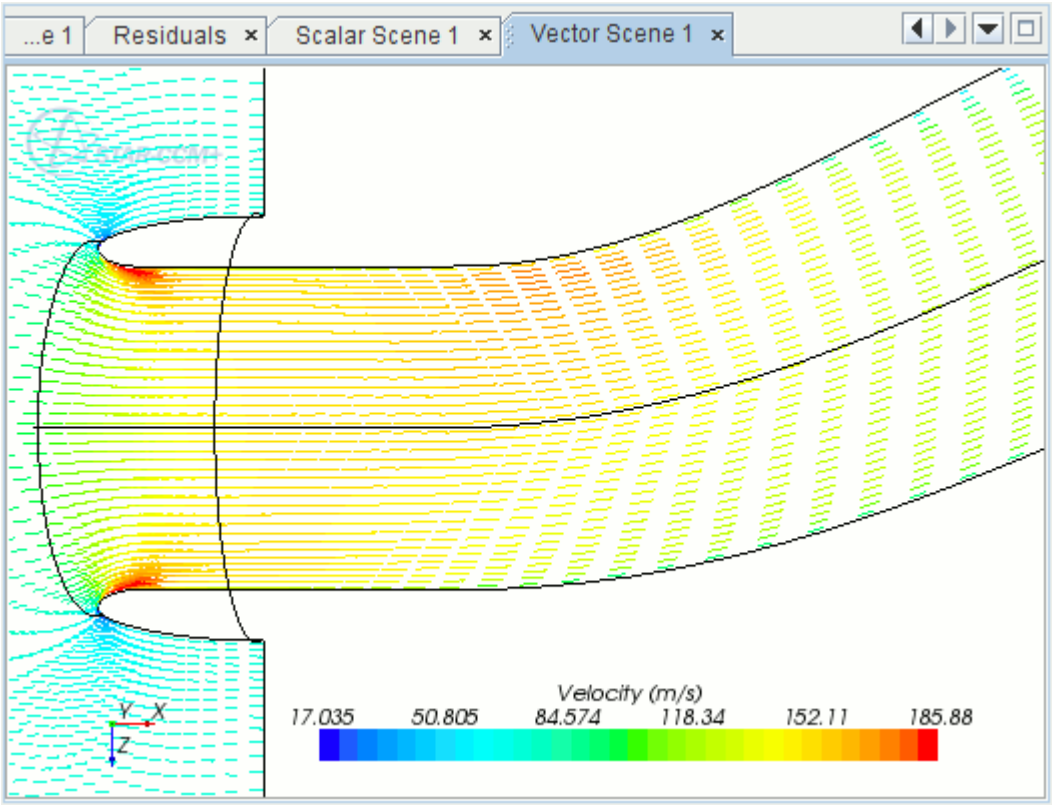


The vector arrows in this scene are too long. To make them shorter:

6. Select the **Scenes** > **Vector Scene 1** > **Displayers** > **Vector 1** > **Glyph** > **Relative Length** node and set the *Glyph Length (%)* to `0.2`.



The arrows now appear shorter.

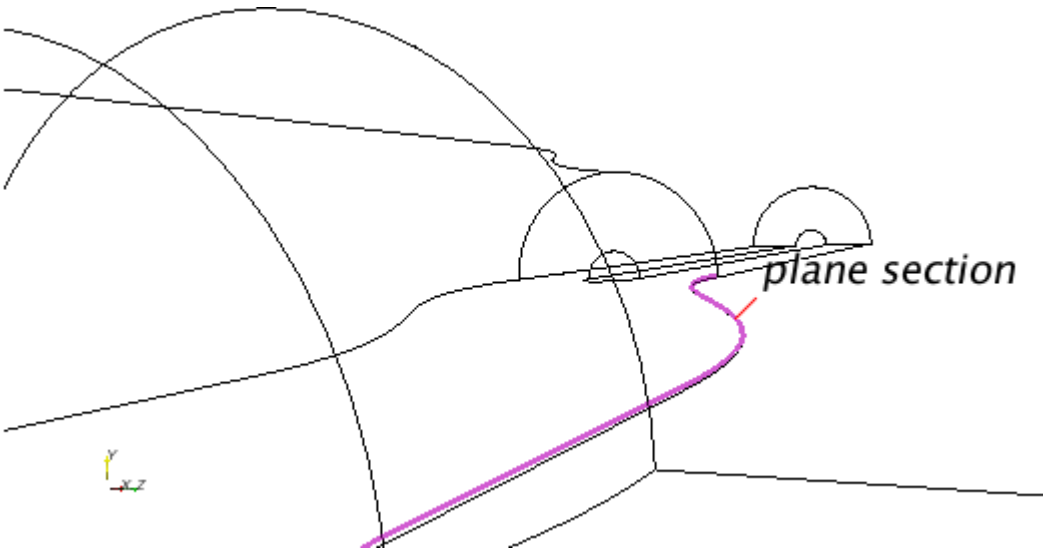


7. Save the simulation.

Setting Up a Line Surface for Plotting

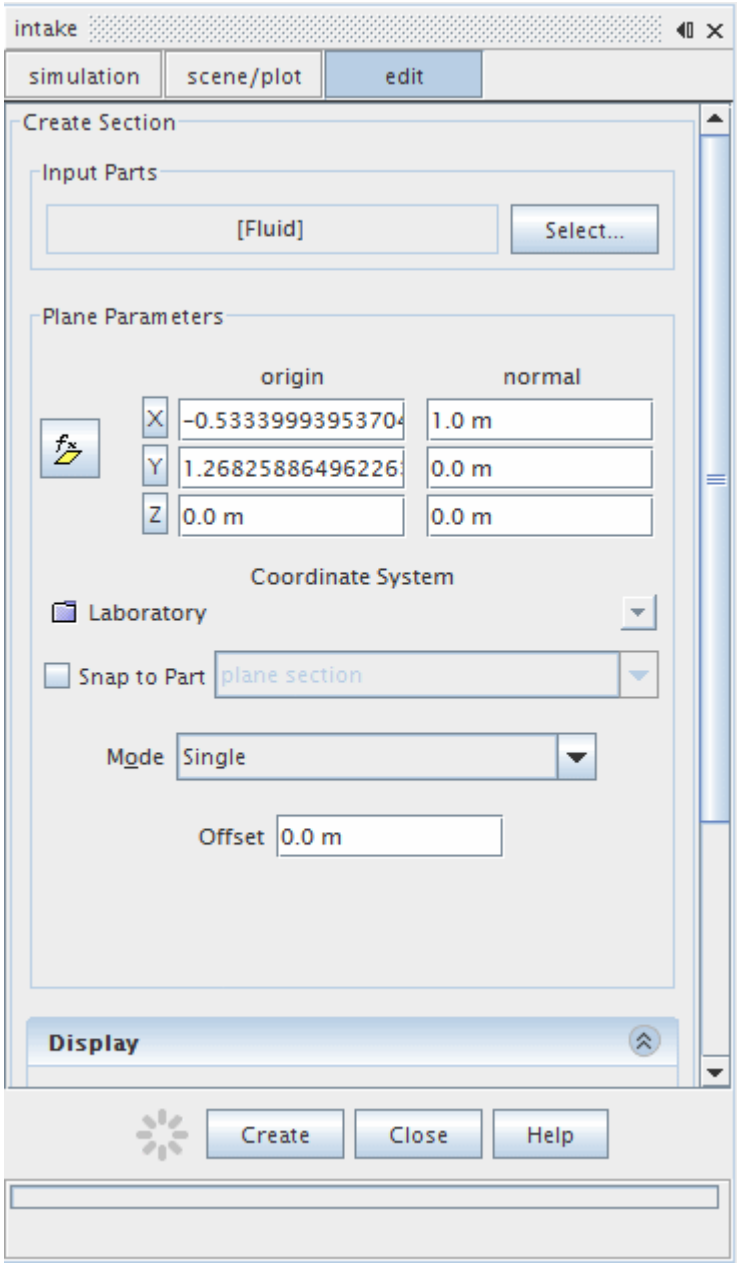
For comparison with the experimental data, a part needs to be created on which to plot the duct wall pressure ratio.

Create the part by intersecting a plane with the surface corresponding to the duct wall, resulting in a line surface.



1. Right-click the **Scenes** node and select **New Scene** > **Geometry**.
2. In the **intake** window, right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane**.

   An interactive in-place dialog appears with default settings to allow definition of the derived part.

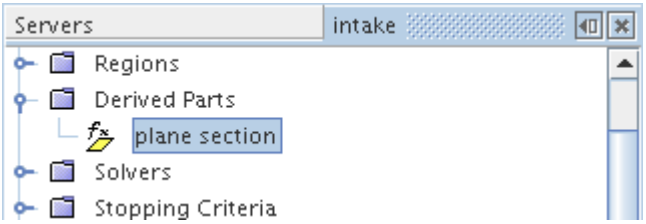


3. Click **Input Parts** to activate the *Select Objects* dialog.
4. Expand the **Regions** > **Fluid** nodes and select **Port Wall** only.

   Note: The **Fluid** region must be deselected.

5. Click **OK**.
6. In the **Plane Parameters** box, enter values for the y-coordinate: `0.001` for the origin and `1` for the normal. Set all other values to zero.
7. In the **Display** box, select **No Displayer**.
8. Click **Create**.

   In the **intake** window, a node representing the implicit surface created is added under the **Derived Parts** node: **plane section**.



9. Click **Close** in the dialog.

   It is now possible to plot data on the new implicit surface.

10. Save the simulation.

Plotting Simulation Data

In this section, the simulation data is plotted using the user-defined field function for pressure ratio and the line surface created from the S-duct wall.

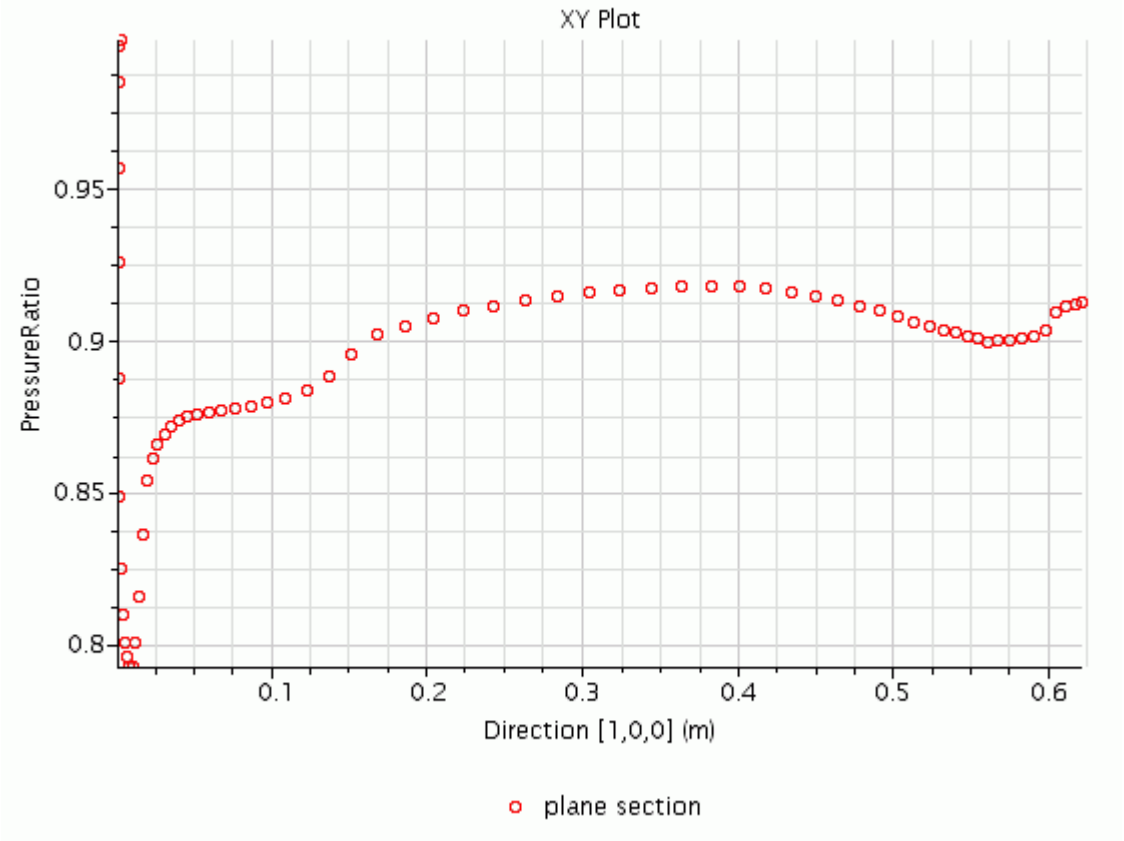
Use the following steps:

1. In the **intake** window, right-click the **Plots** node and select **New Plot** > **XY**.

   This creates an empty plot in the tree and displays it in the **Graphics** window.

2. Select the **XY Plot 1** node and click the right half of the *Parts* property.
3. In the *Select Objects* dialog, select the **Derived Parts** > **plane section**, then click **OK**.
4. Select the **XY Plot 1** > **Y Types** > **Y Type 1** node and make sure that the *Type* property is set to **Scalar**.
5. Select the **Y Type 1** > **Scalar Function** node and set the *Field Function* to **PressureRatio**.

   A plot like the one shown below appears:



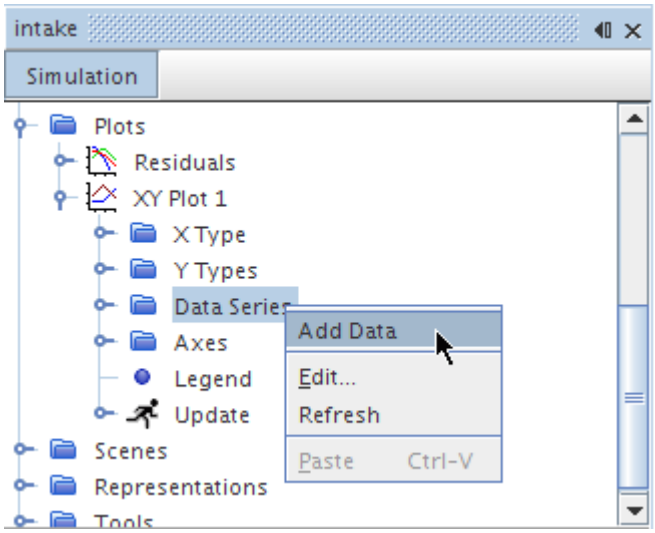
6. Save the simulation.

Plotting Reference Data

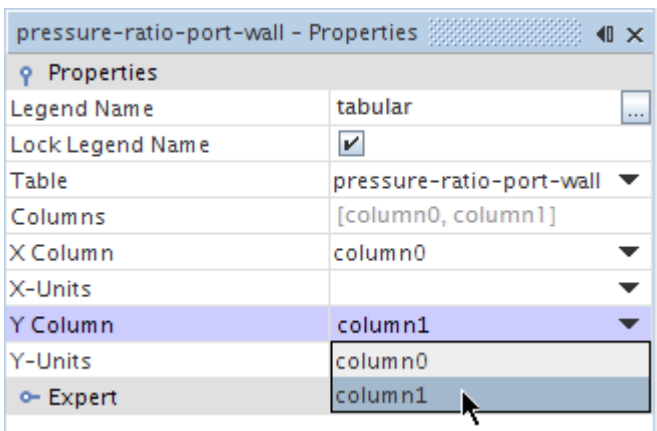
Display an external data set consisting of experimental data together with the simulation data.

First, however, the external data must be loaded into a table.

1. Right-click the **Tools** > **Tables** node and select **New Table** > **File Table**.
2. In the *Open* dialog, select `.xy` from the Files of Type drop-down menu.
3. Navigate to the `pressure-ratio-port-wall.xy` file (this is located in the `doc/startutorialsdata/compressibleFlow/data` subdirectory of your STAR-CCM+ installation directory) and open it.
4. Right-click the **Plots** > **XY Plot 1** > **Data Series** node and select **Add Data**.



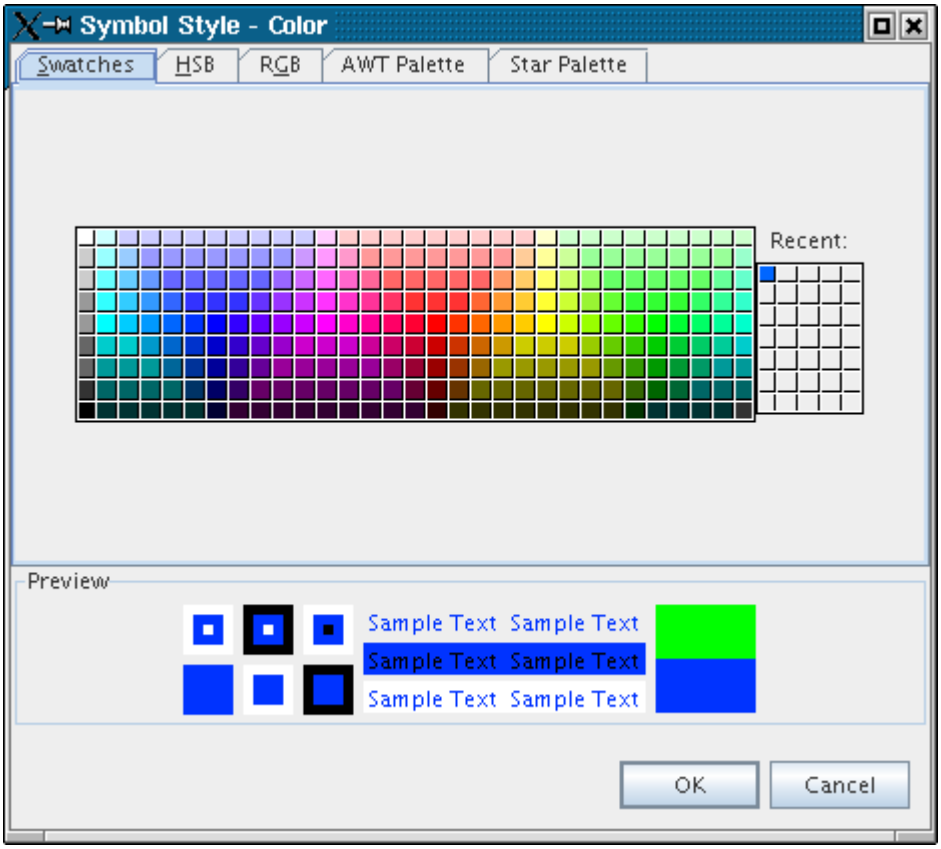
5. Select **pressure-ratio-port-wall** from the *Add Data Providers to Plot* dialog.
6. Select the **Data Series** > **pressure-ratio-port-wall** node.
7. Set *X Column* to **column0**.
8. Set *Y Column* to **column1**.



The tabular data is displayed on the plot. To view this more clearly:

9. Select the **pressure-ratio-port-wall** > **Symbol Style** node:
   a) Set *Shape* to **Filled Triangle**.
   b) Set *Size* to `12`.
   c) Edit the *Color* property.

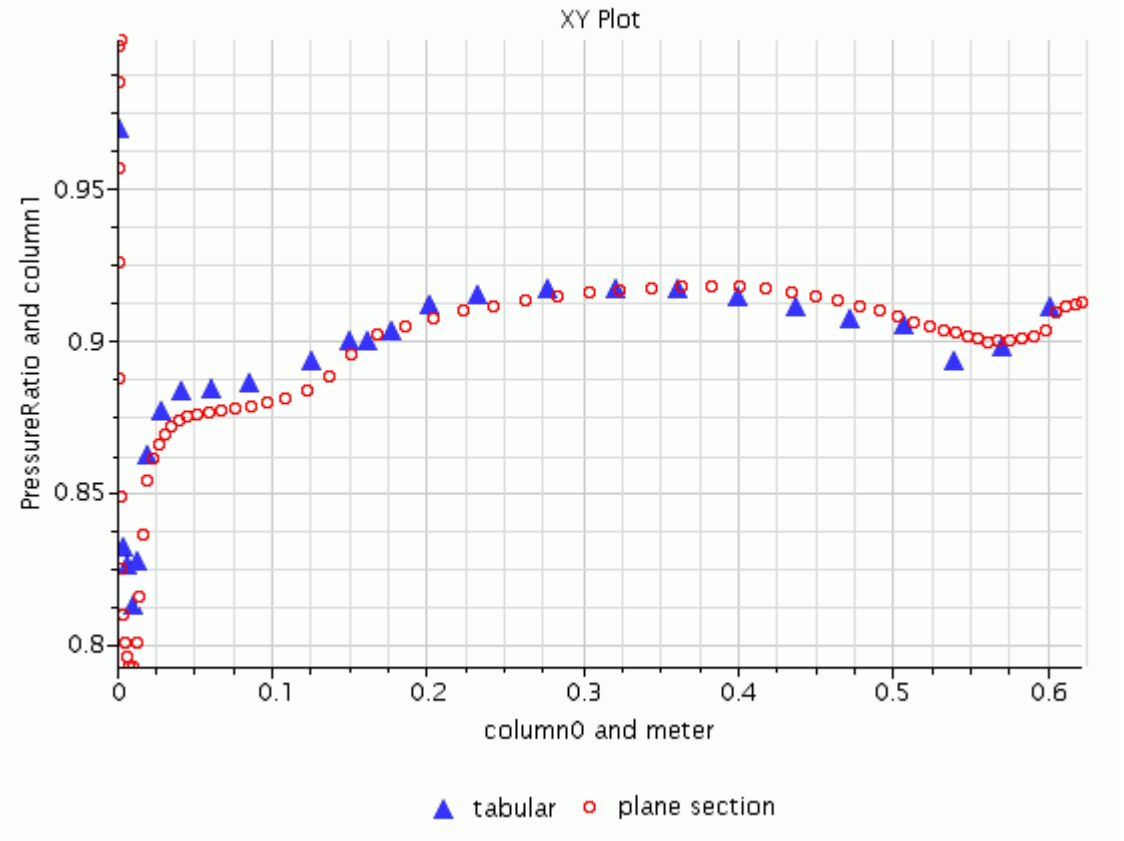
      The *Symbol Style - Color* dialog appears.



Make sure that the **Swatches** tab is active, and then select a bright shade of blue to contrast with the existing plot.

d) Click `OK`.

The external data appears more clearly.



The external data in the preceding illustration is from the following source: Fluid Dynamics Panel Working Group 13. 1991. “Air Intakes for High Speed Vehicles”, *Agard Advisory Report 270*, pp. 139-162.

10. Save the simulation.

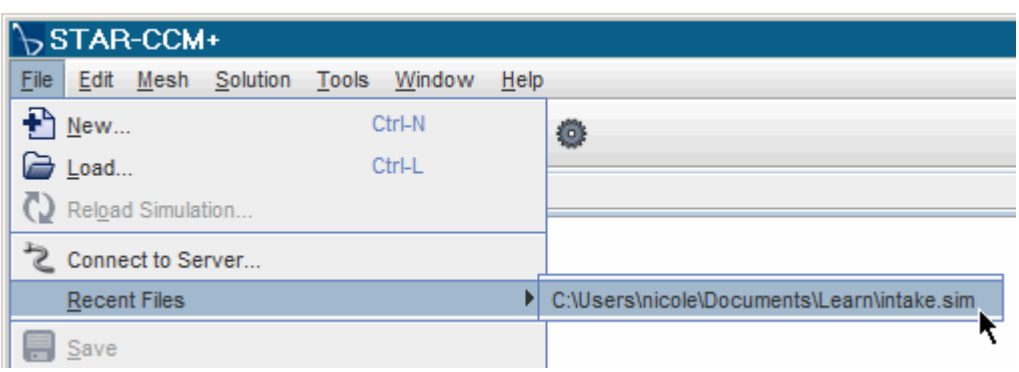
Closing and Restarting

Close and restart the simulation.

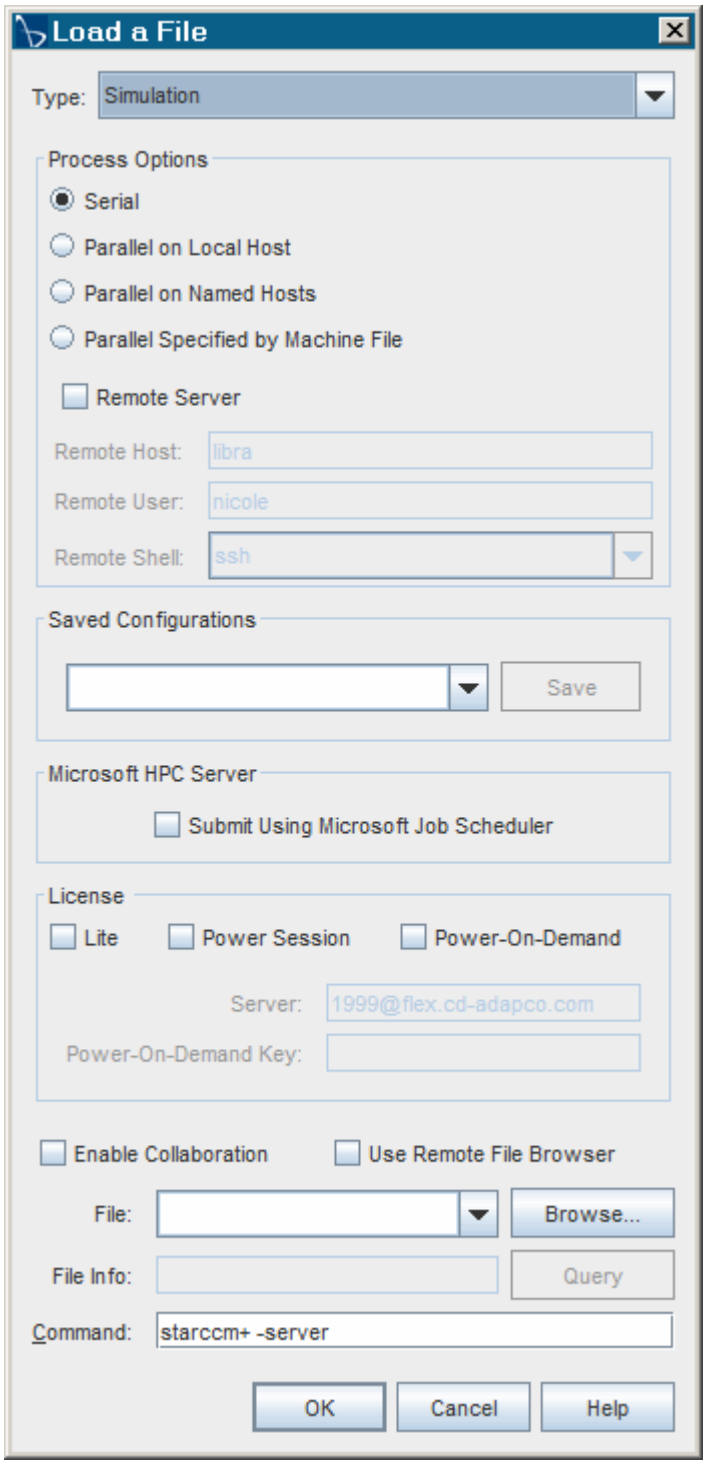
1. Close the file by closing the **intake** window.

It is easy to reopen the file and restore the displays.

2. Select **File** > **Recent Files** > **intake.sim** from the top menus.



Alternatively, you can select the **File** > **Load** menu item. This activates the *Load a File* dialog.



3. Click the **Browse** button to activate the *Open* dialog. Use it to locate the simulation file and then click **Open**.

   Back in the *Load a File* dialog, the name and path of the simulation file appears in the **File** text box.

4. Click **OK**.

   A new window appears for the simulation.

The next step is to restore the displays and plots

5. Expand the **Scenes** node and double-click each scene display's node one at a time.
6. Expand the **Plots** node and double-click the **Residuals** and **XY Plot 1** nodes.

Summary

The Subsonic Flow tutorial illustrated how to solve three-dimensional compressible flow problems.

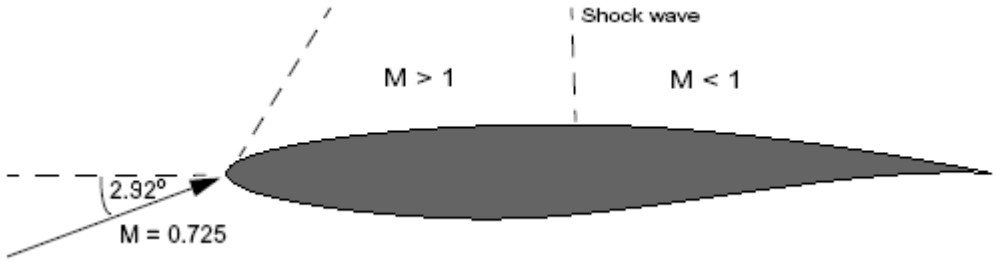
The tutorial demonstrated the following features of STAR-CCM+:

- Combining and fusing regions and boundaries.
- Selecting the models to be used.
- Defining the material properties that are required for the selected models.
- Defining the region and boundary conditions.
- Setting the solver parameters.
- Running the solver until the residuals are satisfactory.
- Analyzing results using the visualization or XY-plot facilities.
- Reopening the simulation.

Transonic Flow: RAE2822 Airfoil

The tutorial simulates two-dimensional, turbulent, compressible, transonic air flow over an idealized airfoil.

The freestream Mach number is 0.725, the Reynolds number based on the chord length is 6.5E6, and the angle of attack is 2.92° corresponding to RAE2822 case 6 in [[787]]



The freestream flow is subsonic, becoming supersonic on the suction side of the airfoil, and subsonic through a shock wave. The lift and drag coefficients are monitored to help determine whether convergence is reached. The final distribution of the pressure coefficient on the airfoil is then compared to experimental data.

Contents:

Importing the Mesh and Naming the Simulation
Converting to a Two-Dimensional Mesh
Selecting the Physics Models
Setting Initial Conditions
Setting Material Properties
Setting Boundary Conditions and Values
Setting Solver Parameters and Stopping Criteria
Visualizing and Initializing the Solution
Plotting Graphs
Running the Simulation
Visualizing the Results
Validating the Results
Summary
Airfoil Tutorial Bibliography

Importing the Mesh and Naming the Simulation

Create a new simulation, import the supplied mesh, and name the simulation `aerofoil.sim`.

Use the following steps:

1. Start up STAR-CCM+ and select the **New Simulation** option from the menu bar.

Continue by importing the mesh and naming the simulation. A one-cell-thick, three-dimensional, hexahedral mesh has been prepared for this analysis. The mesh corresponds to an angle of attack of 0° in the default Laboratory coordinate system.

2. Select **File** > **Import** > **Import Volume Mesh** from the menus.
3. In the *Open* dialog, simply navigate to the `doc/startutorialsdata/compressibleFlow/data` subdirectory of your STAR-CCM+ installation directory and select file `aerofoil.ccm`, which contains the mesh and boundary definitions.
4. Click the **Open** button to start the import.

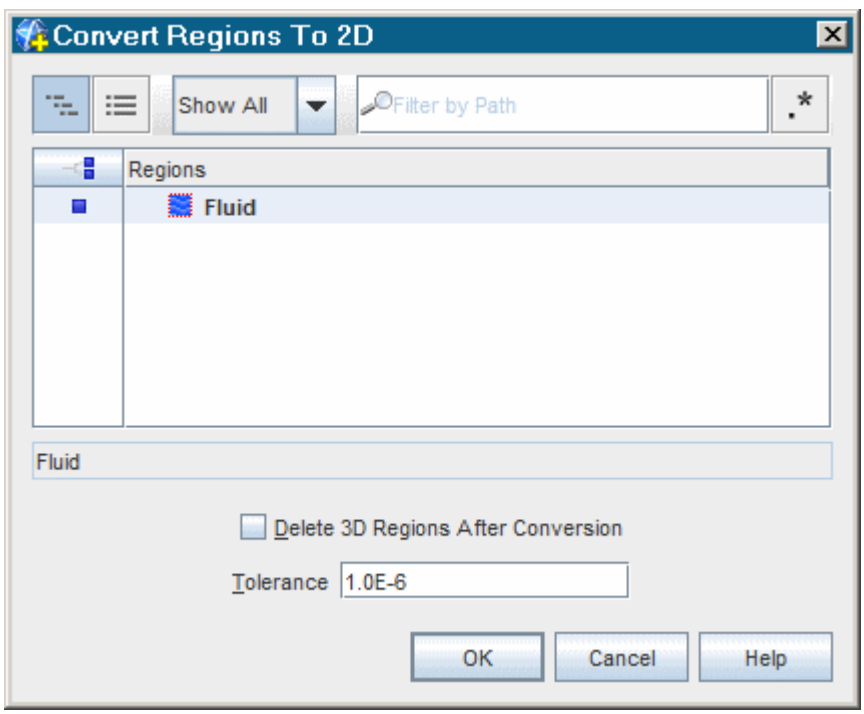
   STAR-CCM+ provides feedback on the import process, which takes a few seconds, in the *Output* window. A geometry scene is created in the *Graphics* window.
5. Finally, save the new simulation to disk under file name `aerofoil.sim`.

Converting to a Two-Dimensional Mesh

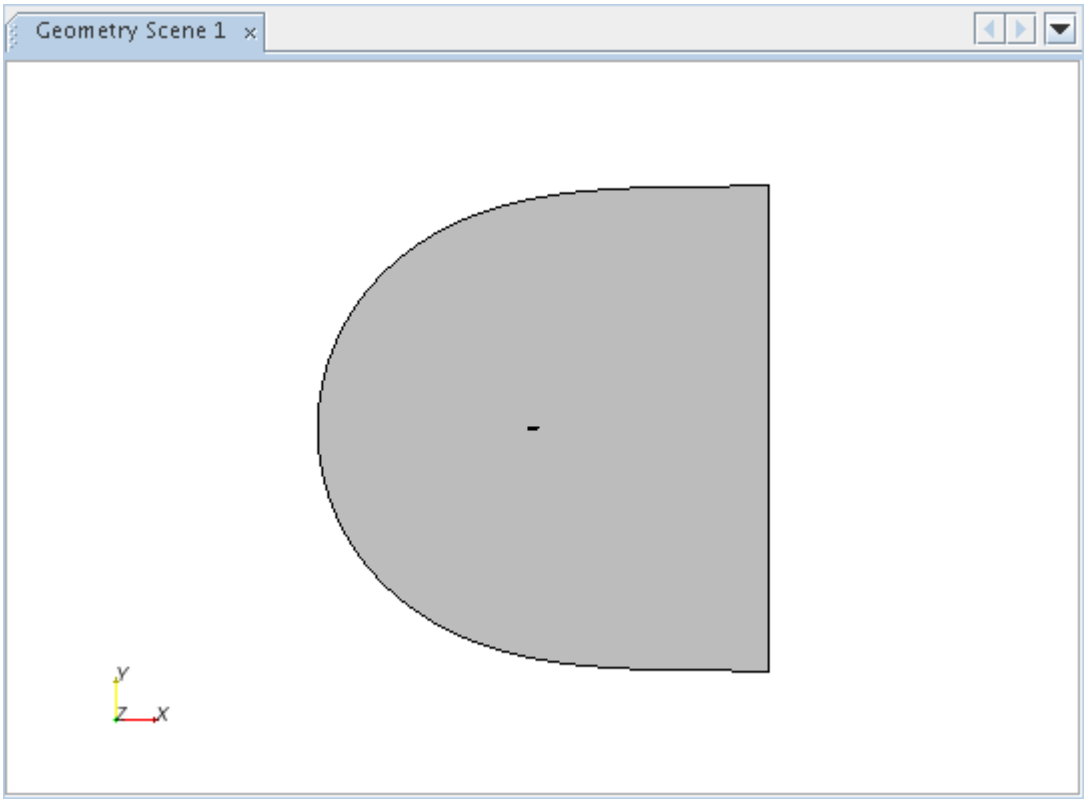
Convert the mesh region to a two-dimensions.

The existing mesh aligns with the X-Y plane and has a boundary plane at Z = 0 and thus satisfies the requirements to convert three-dimensional meshes to two dimensions:

1. Select **Mesh** > **Convert to 2D**
2. In the *Convert Regions to 2D* dialog that appears, activate **Delete 3D regions after conversion** and click **OK**.



Once you have clicked **OK**, the mesh conversion takes place and the *Geometry Scene 1* display shows the two-dimensional geometry in the *Graphics* window. (If the image does not appear immediately, click (**Reset View**) on the toolbar.)



All the geometry parts are shown, viewed from the positive Z-direction. The mouse rotation option is suppressed for two-dimensional scenes.

3. Right-click the **Physics 1** continuum node and select **Delete**.
4. Click **Yes** in the confirmation dialog.

Selecting the Physics Models

Models define the spatial and temporal solution methods and the physical properties of the flow. In this example, the flow is steady, turbulent, and compressible.

Use the default Spalart-Allmaras turbulence model and the ideal gas model. Use the coupled solver, which is recommended for all supersonic and transonic compressible flows.

By default, a **Physics 1 2D** continuum is created when the mesh is converted to two-dimensional.

1. Rename the **Physics 1 2D** node to `Aerofoil`.

Edit the continuum definition to select appropriate physical models for the fluid.

2. For the physics continuum **Continua** > **Aerofoil**, select the following models, in order:

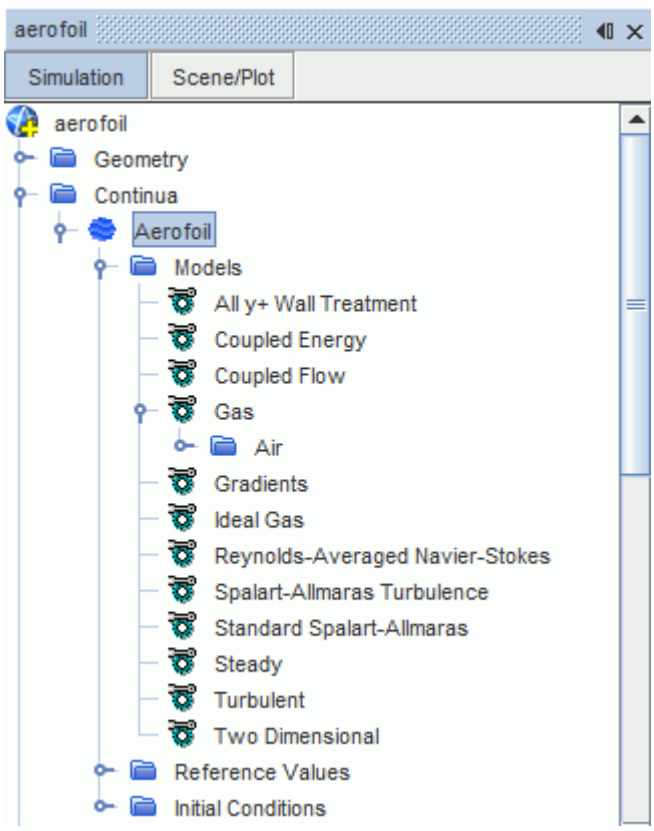
   Make sure that the **Two Dimensional** model is activated and the **Auto-select** option is selected.

Group Box	Model
Material	**Gas**
Flow	**Coupled Flow**
Equation of State	**Ideal Gas**
Time	**Steady**
Viscous Regime	**Turbulent**
Reynolds-Averaged Turbulence	**Spalart-Allmaras Turbulence**

3. Click **Close**.

Inside the **Continua** node, the color of the **Aerofoil** node has turned from gray to blue to indicate that models have been activated.

4. To review the models, open the **Aerofoil** > **Models** node.



5. Save the simulation.

Setting Initial Conditions

Specify the coordinate system and the velocity and temperature.

The initial velocity field applies freestream conditions across the entire domain. The initial velocity of 252 m/s is calculated as:

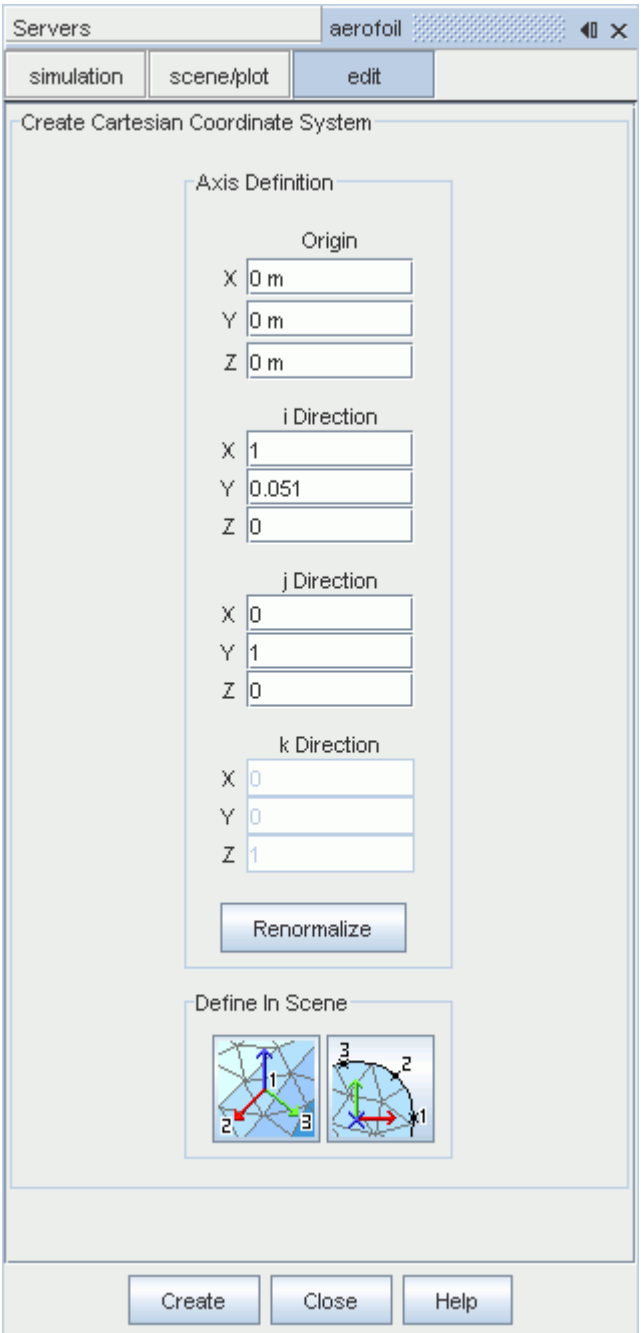
$$u = M\sqrt{\gamma RT}$$

where the Mach number $M = 0.725$, the ratio of specific heats $\gamma = 1.4$, the specific gas constant for air R = 287 J/kg-K and the temperature T = 300 K.

To specify an angle of attack of 2.92° for the initial velocity, create a coordinate system.

1. Open the **Tools** node at the bottom of the simulation tree and right-click the **Coordinate Systems** > **Laboratory** > **Local Coordinate Systems** node.
2. Select **New** > **Cartesian**.
3. In the in-place dialog, in the *Axis Definition* box, change the i Direction to the following:

```
[1, 0.051, 0]
```



4. Click **Renormalize**.

   The j Direction changes automatically to make sure that the axes are perpendicular. The i Direction readjusts itself slightly.

5. Click **Create**, then **Close**.

A **Cartesian 1** node is created within the **Coordinate Systems** node. The resulting *Properties* window is shown below.

This node now defines a coordinate system that, when viewed down the positive Z-axis, has its x- and y-axes rotated anti-clockwise through an angle of 2.92° compared to the laboratory system.

6. Edit the **Initial Conditions** node and set the following properties:

Node	Property	Setting
Velocity	*Coordinate System*	**Laboratory - > Cartesian 1**
	Value	`[252, 0.0, 0.0]`
Static Temperature	*Value*	`300`

The default values for the remaining initial conditions are suitable for this problem.

7. Save the simulation.

Setting Material Properties

Set the material properties for gas and air.

The dynamic viscosity for air is calculated as:

$$\mu = \frac{\rho u L}{Re}$$

where the velocity u = 252 m/s, the chord length L = 1 m, and the Reynolds number Re = 6.5E6. The density ρ has a value of 1.177 kg/m^2 and is calculated according to the ideal gas law:

$$\rho = \frac{P}{RT}$$

where the pressure P = 101325 Pa, the specific gas constant for air R = 287 J/kg-K, and the temperature T = 300 K.

The thermal conductivity for air is calculated as:

$$\lambda = \frac{c_p \mu}{Pr}$$

where the specific heat is c_p = 1003.62 J/kg-K and the Prandtl number for air is Pr = 0.72.

To specify the resulting values of 4.58E-5 Pa-s for the dynamic viscosity and 0.0638414 W/m-K for the thermal conductivity:

1. Open the **Gas** and **Air** nodes.

The material properties for air are contained within.

2. Edit the **Material Properties** node and set the following properties:

Node	Property	Setting
Dynamic Viscosity	*Method*	**Constant**
└ **Constant**	*Value*	`4.58E-5 Pa-s`
Thermal Conductivity	*Method*	**Constant**

Node	Property	Setting
└ **Constant**	*Value*	`0.0638414 W/m-K`

Setting Boundary Conditions and Values

Set the physical conditions at the domain boundaries.

The geometry for this tutorial has only two boundaries:

- A wall boundary representing the surface of the airfoil.
- A free-stream boundary at the external edge of the solution domain.

1. Rename the **Regions** > **Default_Fluid 2D** node to `Fluid`.
2. Select the **Fluid** > **Boundaries** > **freestream** > **Physics Conditions** > **Flow Direction Specification** node and set the *Method* property to **Components**.
3. Edit the **Physics Values** node and set the following properties:

Node	Property	Value
Flow Direction	*Coordinate System*	**Laboratory -** > **Cartesian 1**
Mach Number	*Value*	`0.725`
Static Temperature	*Value*	`300` K

4. Save the simulation.

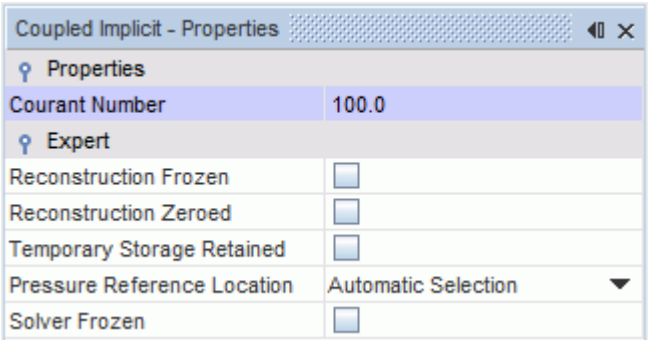
Setting Solver Parameters and Stopping Criteria

Set the Courant number and the maximum number of steps.

The simplicity of this problem allows a rapidly converging solution to be attained using a large Courant number.

In problems involving more complex geometries or physics, attempting to shorten the run time in this way can cause the solution to diverge. To increase the Courant number:

1. Select the **Solvers** > **Coupled Implicit** node and change the *Courant Number* to `100.0`.



2. Select the **Stopping Criteria** > **Maximum Steps** node and set the *Maximum Steps* to `300`.
3. Save the simulation.

Visualizing and Initializing the Solution

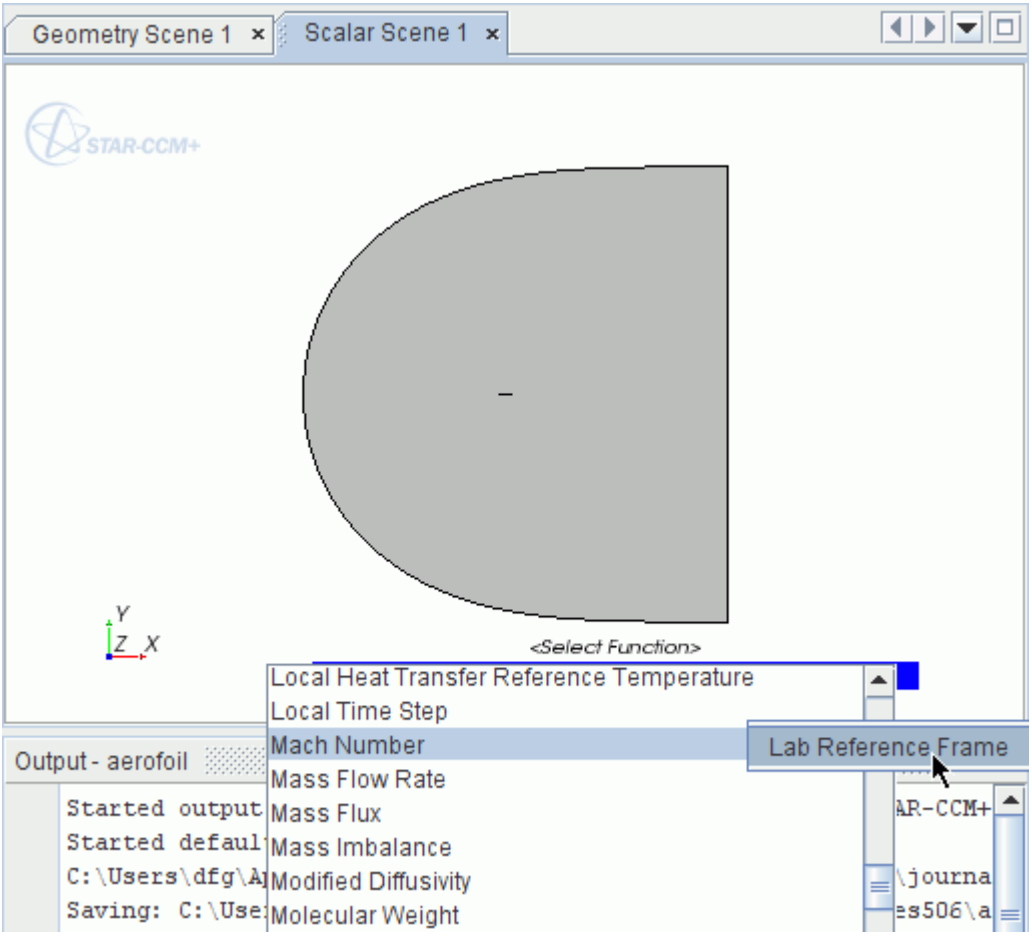
View the Mach number profile during the run to monitor the supersonic flow region above the airfoil.

Start by creating a scalar scene:

1. Right-click the **Scenes** node, and select **New Scene** > **Scalar**.

The *Scalar Scene 1* display appears.

2. Right-click the scalar bar at the bottom of the display and select **Mach Number** > **Lab Reference Frame** from the pop-up menu.



3. Initialize the run by clicking (**Initialize Solution**) in the toolbar, then use the middle mouse button to zoom in on the airfoil in the center of the scalar scene.

To change the style of the Mach number contours:

4. Select the **Scalar Scene 1** > **Displayers** > **Scalar 1** node and set the *Contour Style* to **Smooth Filled**.
5. Save the simulation.

Plotting Graphs

Plot the lift and drag coefficients to determine when the analysis has converged.

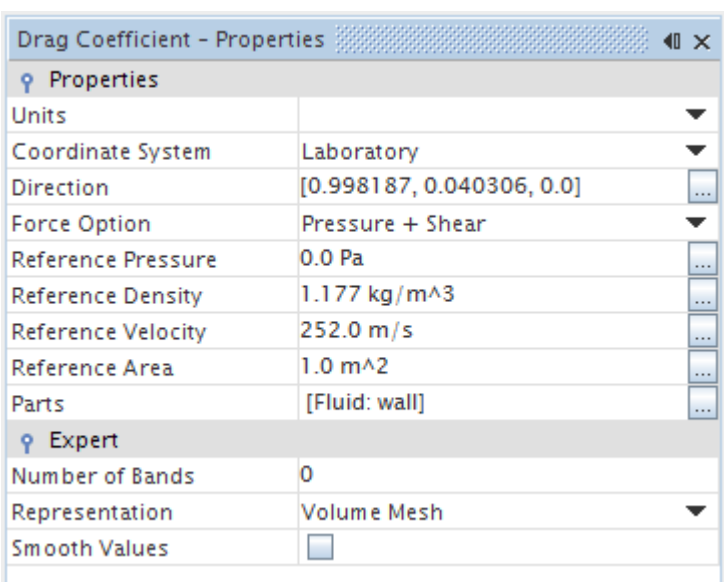
Use the following steps:

1. Right-click the **Reports** node and select **New Report** > **Force Coefficient**.

   A new report node named **Force Coefficient 1** is created.

2. Rename this node `Drag Coefficient` then enter the following information:

Property	Value
Units	Empty
Coordinate System	**Laboratory**
Direction	`[0.998187, 0.040306, 0.0]`
Force Option	**Pressure + Shear**
Reference Pressure	`0.0 Pa`
Reference Density	`1.177 kg/m^3`
Reference Velocity	`252.0 m/s`
Reference Area	`1.0 m^2`
Parts	**[Fluid: wall]**
Number of Bands	`0`
Representation	**Volume Mesh**
Smooth Values	Deactivated



3. Right-click the **Drag Coefficient** node and select **Create Monitor and Plot from Report**.

A new plot node appears named **Drag Coefficient Monitor Plot**.

4. Select the **Drag Coefficient Monitor Plot** node and set the *Title* to `Drag Coefficient Monitor Plot`.

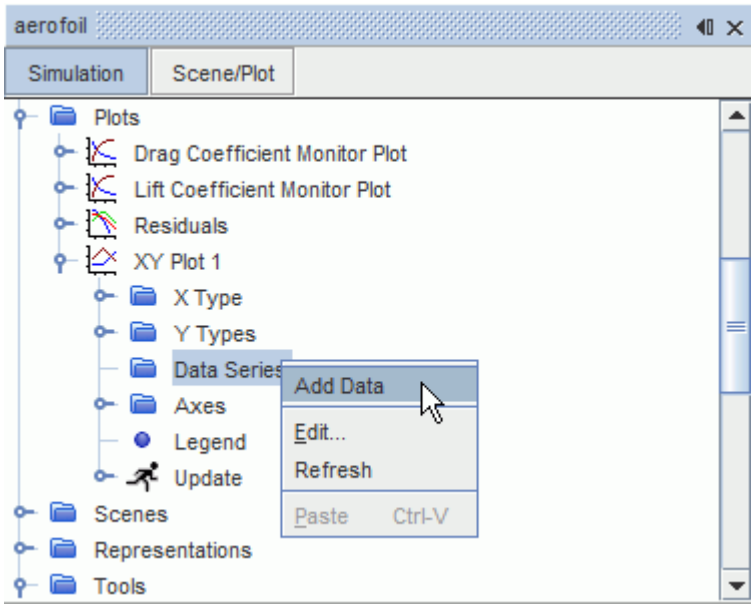
To create a plot for the lift coefficient:

5. Repeat the steps described above to create and display a plot for the lift coefficient. All settings are the same as for the drag coefficient.
6. Rename the report node **Lift Coefficient** and set its *Direction* property to `[-0.0403, 0.9981, 0.0]`.
7. Select the **Lift Coefficient Monitor Plot** node and set the *Title* to `Lift Coefficient Monitor Plot`.

Experimental data for the pressure coefficient on the airfoil are provided in file `aero_exp.xy` in the `doc/startutorialsdata/compressibleFlow/data` directory. Plot the data on a graph alongside the results of the analysis.

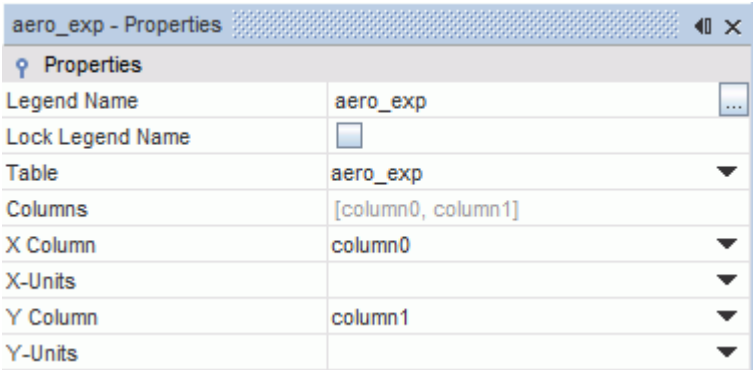
To plot the experimental data:

8. Right-click the **Tools** > **Tables** node and select **New Table** > **File Table**.
9. Locate and open file `aero_exp.xy`.
10. Right-click the **Plots** node and select **New Plot** > **XY Plot**.
11. Right-click the **XY Plot 1** > **Data Series** node and select **Add Data**.



12. In the *Add Data Providers to Plot* window, select **aero_exp**.
13. Select the **Data Series** > **aero_exp** node and set **aero_exp** for the *Table* property.

Make sure that the *X Column* and *Y Column* properties are as shown below:



A graph of the experimental data appears in XY Plot 1.

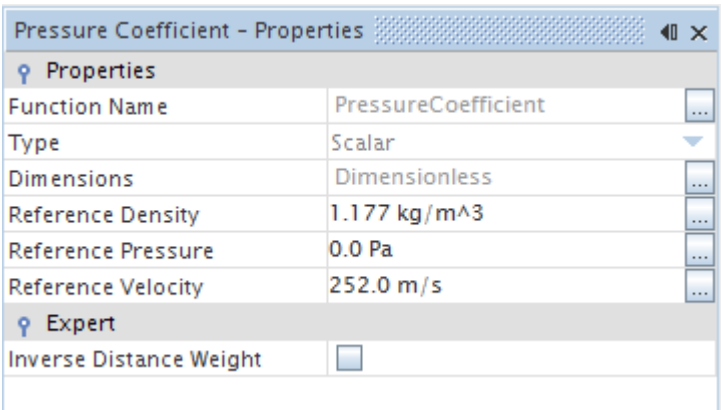
To add the numerical data to the same graph:

14. Select the **XY Plot 1** node and click the *Parts* property.
15. Select **Fluid: wall** in the *XY Plot 1 - Parts* dialog.
16. Select the **Y Types** > **Y Type 1** > **Scalar Function** node and set **Pressure Coefficient** for the *Field Function* property.

The initial pressure coefficient is shown in the XY Plot 1 as being zero everywhere.

The pressure coefficient requires specification of a reference pressure and a reference velocity.

17. Select the **Tools** > **Field Functions** > **Pressure Coefficient** node and enter a *Reference Density* of `1.177 kg/m^3` and a *Reference Velocity* of `252 m/s`, have the settings shown below:



The usual convention in aerodynamics problems is to reverse the y-axis orientation in pressure coefficient plots.

18. Select the **XY Plot 1** > **Axes** node and set the following properties:

Property	Setting
Preferred X-Axis	**Left Axis**
Preferred Y-Axis	**Bottom Axis**

19. Select the **Axes** > **Left Axis** and activate **Reverse**.

20. Save the simulation.

Running the Simulation

The simulation is ready to run.

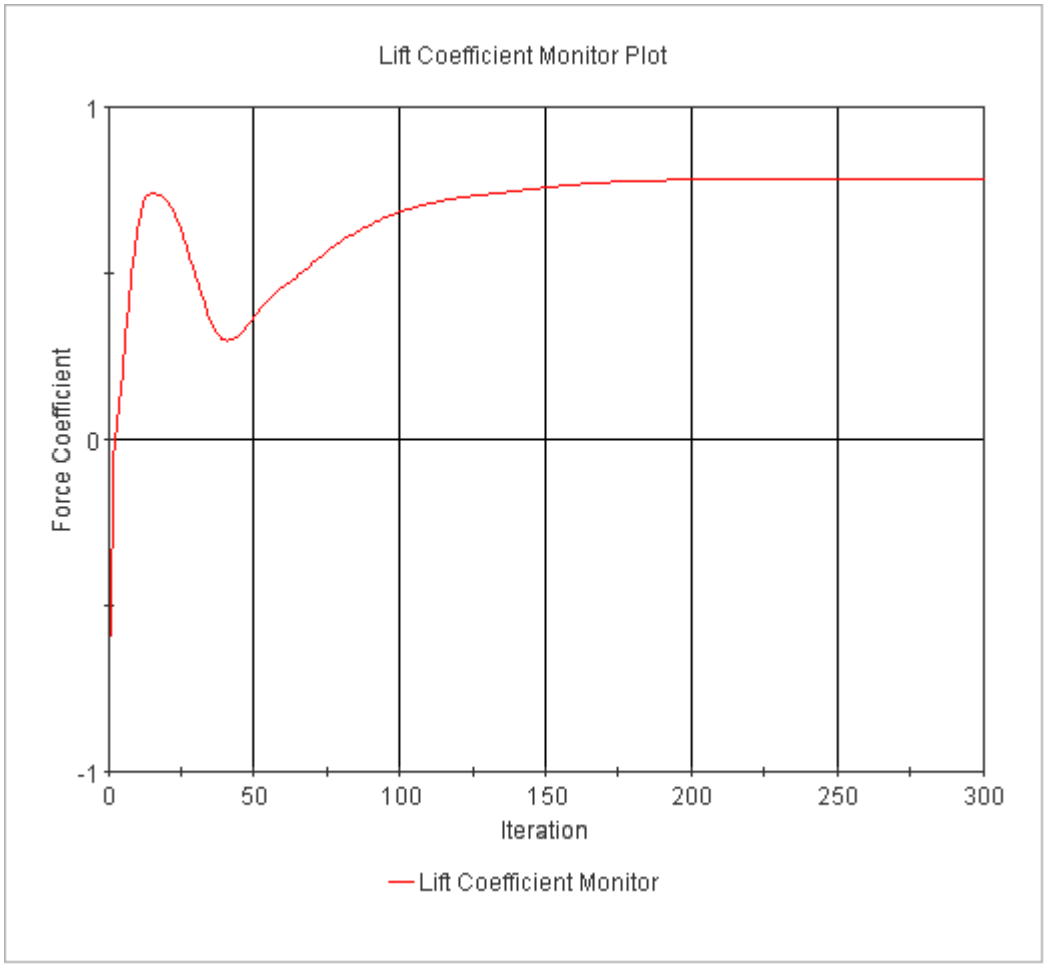
1. To run the simulation, click (**Run**) in the top toolbar, or use the **Solution** > **Run** menu item.

The Residuals display is created automatically and shows the progress of the solver. Observe the run progress by selecting one of the tabs at the top of the *Graphics* window.

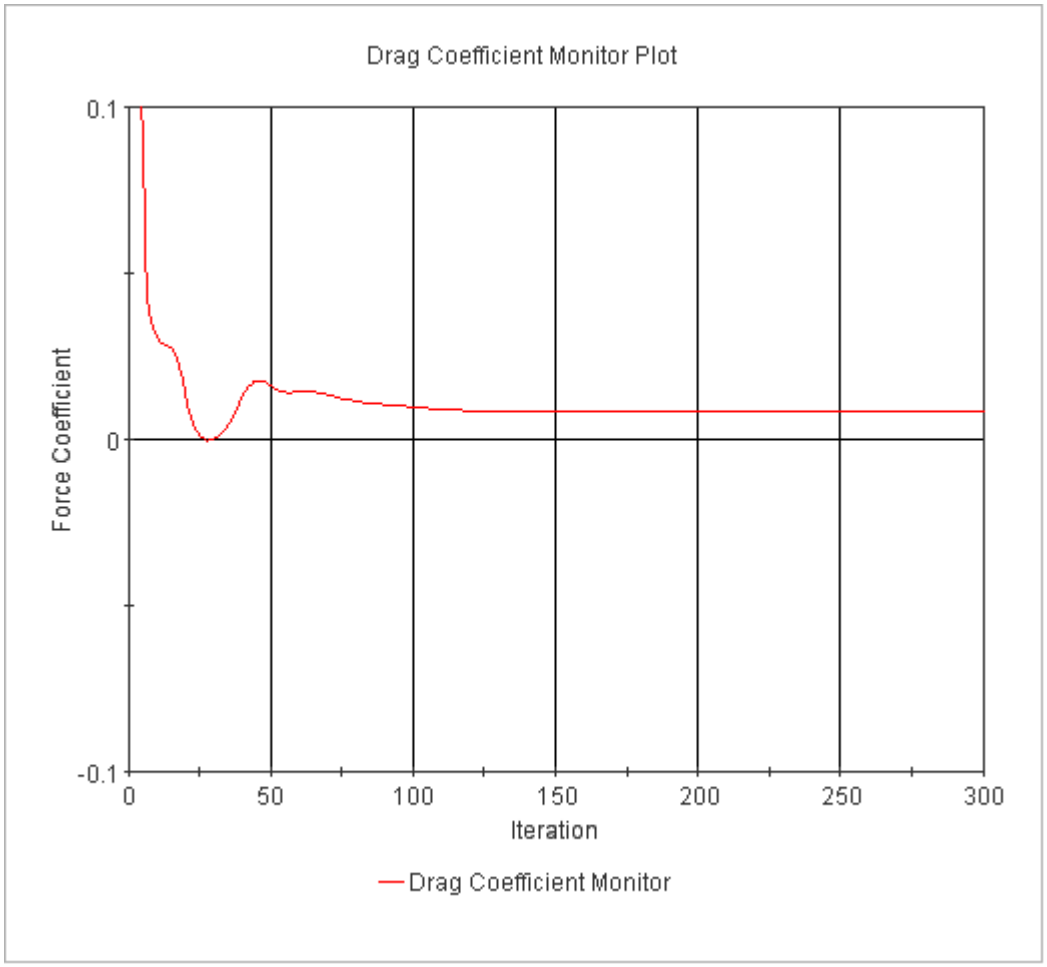
During the run, it is possible to stop the analysis by clicking (**Stop**) in the toolbar. Click (**Run**) to continue the solution. If left alone, the simulation continues until 300 iterations have been completed.

Once this stage is reached, make sure that the solution has converged by examining the lift and drag coefficient plots.

2. Double-click the **Lift Coefficient Monitor Plot** node to display the results in the *Graphics* window.



3. Similarly, display the drag coefficient plot and adjust its y-axis scale.

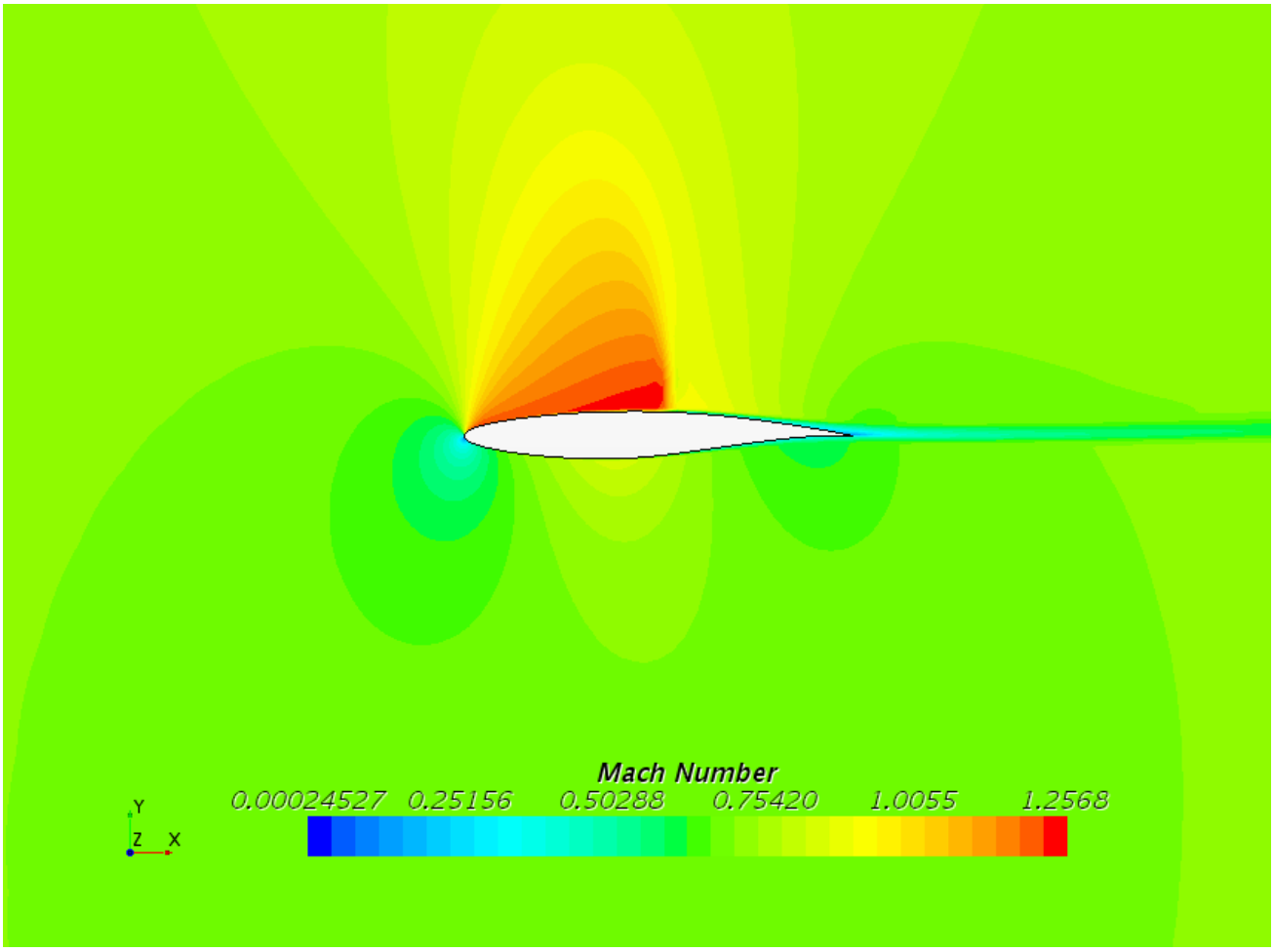


Both monitors have reached constant values so it is reasonable to conclude that the solution has converged.

4. Save the simulation.

Visualizing the Results

The *Scalar Scene 1* display shows the Mach number profile at the end of the run. The profile shows the transonic flow around the airfoil, including the shock wave above it.



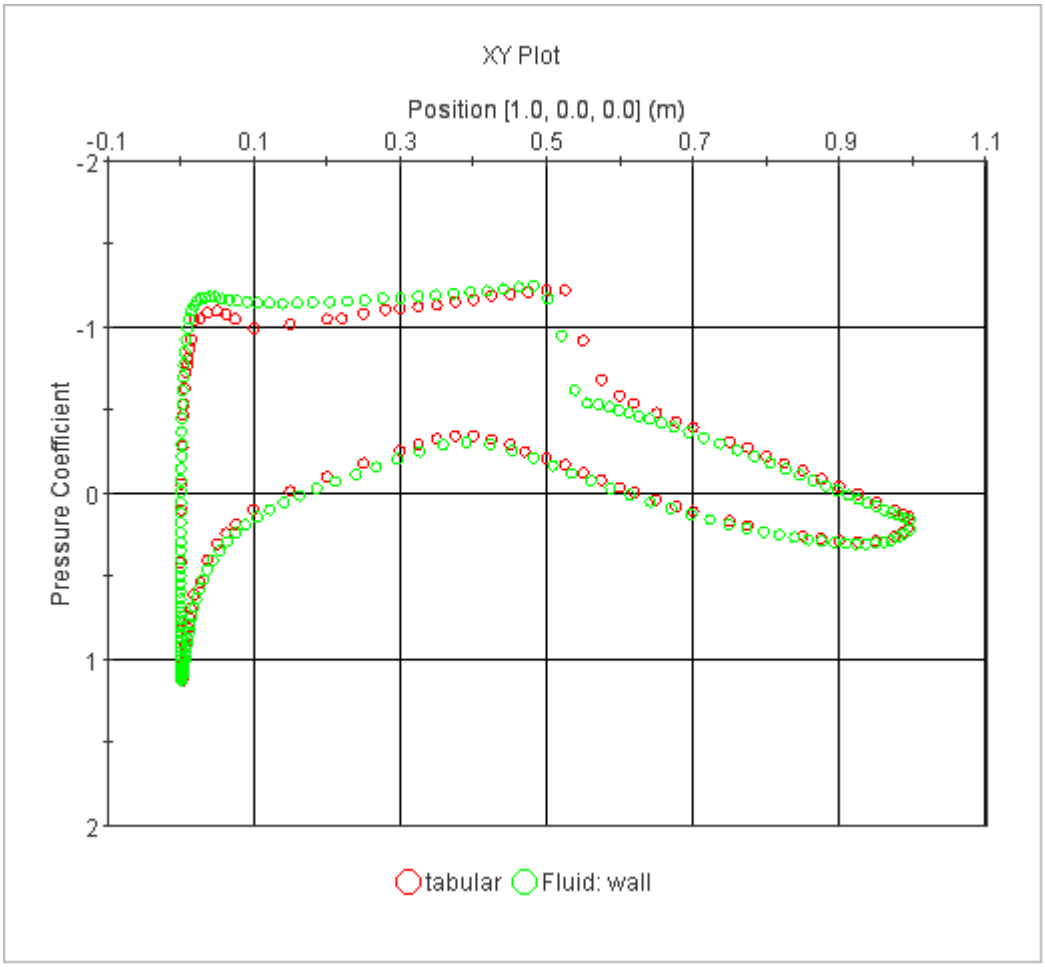
Validating the Results

Examine the XY plot and life coefficient report.

To view the numerical and experimental data:

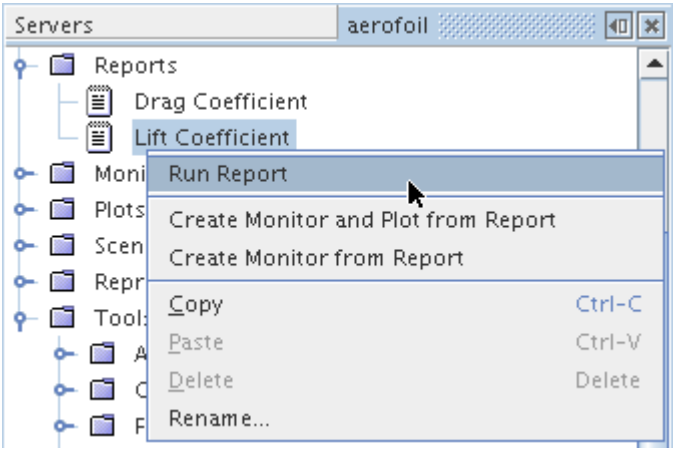
1. Select the *XY Plot 1* tab.

This plot appears as shown below with the X-axis scale adjusted.



Other than the shock position, which is in error as a result of the mesh coarseness and choice of turbulence model, the numerical pressure coefficients compare well with the experimental data. To see the calculated lift coefficient:

2. Right-click the **Reports** > **Lift Coefficient** node and then select **Run Report**.



In the *Output* window, the *Lift Coefficient Report* returns a value of 0.79. Similarly, the drag coefficient report gives a value of 0.0081.

Summary

The Transonic Flow tutorial simulated two-dimensional, turbulent, compressible, transonic flow.

This tutorial introduced the following features of STAR-CCM+:

- Defining models for compressible flow problems.
- Defining the material properties for the selected models.
- Setting solver parameters for a steady-state run.
- Plotting graphs comparing results with experimental data.
- Initializing and running the solver to a specified stopping criterion.
- Analyzing the results using the built-in visualization facilities.

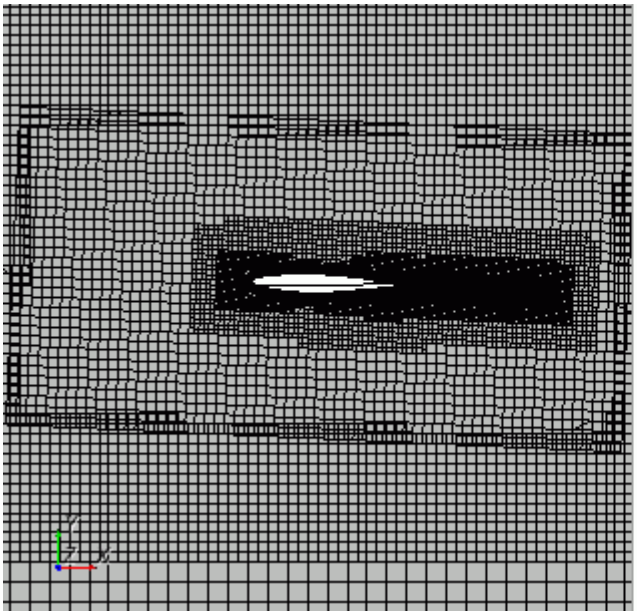
Airfoil Tutorial Bibliography

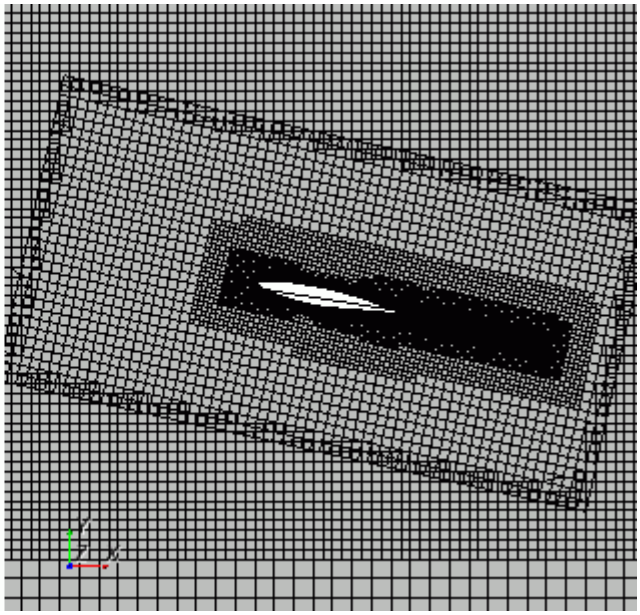
[787] *Cook, P.H., M.A. McDonald, M.C.P. Firmin "Aerofoil RAE 2822 - Pressure Distributions, and Boundary Layer and Wake Measurements Experimental Data Base for Computer Program Assessment", AGARD Report AR 138, 1979*

Transonic Flow: RAE2822 Airfoil Using Overset Mesh

This tutorial demonstrates how to use an overset mesh to simulate transonic flow over the RAE2822 airfoil.

The physics models, initial, and boundary conditions are the same as in the Transonic Flow tutorial. By using an overset mesh for the region surrounding the airfoil, this tutorial shows how the angle of attack is changed without having to generate a mesh. This approach greatly simplifies the case setup for parametric studies.





Contents:

Importing the Geometry
Working with Regions
Preparing the Meshes
Specifying Angle of Attack
Coupling the Overset Region
Setting up Physics Models
Setting Material Properties
Setting Initial Conditions
Setting Boundary Conditions
Setting Solver Parameters
Preparing a Scalar Scene
Monitoring Convergence
Running the Simulation
Visualizing the Results
Changing Angle of Attack

Clearing and Rerunning the Simulation
Visualizing the Results
Summary

Importing the Geometry

Load a simulation prepared with the geometry.

The simulation file contains geometry parts for:

- Background region of the mesh
- Overset region that contains the airfoil
- Several parts for use in volumetric mesh controls

To load the file:

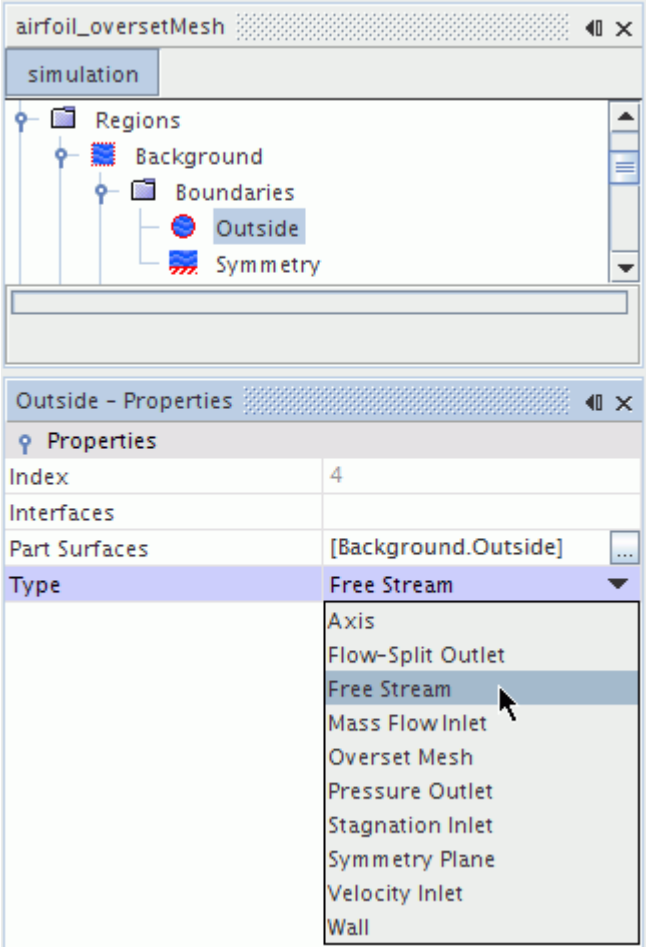
1. Start up STAR-CCM+ in a manner that is appropriate to your working environment.
2. Select **File** > **Load Simulation**.
3. In the *Load Simulation* dialog, navigate to the `doc/startutorialsdata/compressibleFlow/data` subdirectory of your STAR-CCM+ installation directory and select `airfoil_oversetParts.sim`.
4. Click **Open** to load the simulation.
5. Select **File** > **Save As** and save the simulation under the name `airfoil_oversetMesh.sim`.

Working with Regions

Create and define regions and boundaries for the domain.

First, assign the Background and Overset parts to regions:

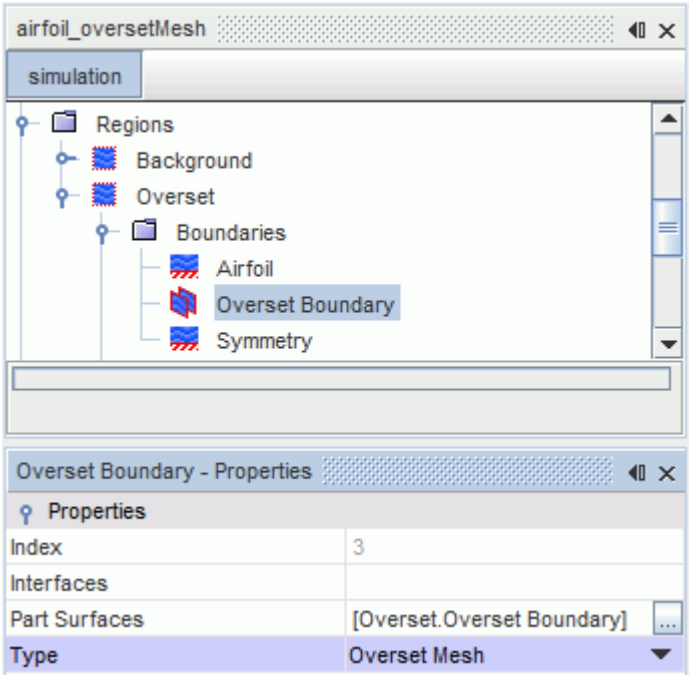
1. Within the **Parts** node, multi-select the **Background** and **Overset** parts.
2. Right-click the selected parts and choose **Assign Parts to Regions...**.
3. In the *Assign Parts to Regions* dialog, select **Create a Region for Each Part** for the **Region Mode** and **Create a Boundary for Each Part Surface** for the **Boundary Mode**.
4. Click **Apply**, and then **Close**.
5. Select the **Regions** > **Background** > **Boundaries** > **Outside** node and set the **Type** to **Free Stream**.



6. Repeat this procedure for the **Symmetry** boundary, but set its type to **Symmetry Plane**.
7. For the **Overset** > **Boundaries** node, set the following properties:

Property	Setting
Airfoil	**Wall**
Overset Boundary	**Overset Mesh**
Symmetry	**Symmetry Plane**

The boundaries for the Overset region are shown below. The Overset Mesh boundary is highlighted.



The Overset Mesh boundary has no effect on the flow solution. It is used with an Overset Mesh interface to couple the overset and background regions.

Preparing the Meshes

You create two Automated Mesh Operations to represent the Background and Overset regions.

The Automated Mesh Operations contain the same meshers, but use different mesh settings. It is important that, in the overlapping zone, which is defined by the Overset Boundary, mesh cells are of a comparable size in both meshes.

Contents:

Setting up the Background Mesh
Setting up the Overset Mesh
Generating the Mesh

Setting up the Background Mesh

First, you specify the mesh settings and volumetric refinements for the background mesh.

To set up the background mesh:

1. Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
2. In the *Create Automated Mesh Operation* dialog:
   a) Select **Background** from the *Parts* list.
   b) Select the following meshers, in order:

Group	Model
Surface Meshers	**Surface Remesher**
Core Volume Meshers	**Trimmed Cell Mesher**
Optional Boundary Layer Meshers	**Prism Layer Mesher**

3. Rename the **Operations** > **Automated Mesh** node to `Background Mesh`.
4. Edit the **Operations** > **Background Mesh** > **Default Controls** node and set the following properties:

Node	Property	Setting
Base Size	**Value**	`0.5 m`
Minimum Surface Size	**Percentage of Base**	`25.0`
Volume Growth Rate	**Default Growth Rate**	**Slow**
	Surface Growth Rate	**Slow**

5. Right-click the **Operations** > **Background Mesh** > **Custom Controls** node and select **New** > **Volumetric Control**.
6. Edit the **Volumetric Control** node and set the following properties:

Node	Property	Setting
Volumetric Control	**Parts**	**Overlap**
Controls		
└ **Surface Remesher**	**Customize Size**	Activated
└ **Trimmer**	**Customize Isotropic Size**	Activated
Values		
└ **Custom Size**	**Size Type**	**Relative to base**
	Percentage of Base	`12.5`

Setting up the Overset Mesh

Now, you specify the mesh settings for the overset mesh.

To set up the overset mesh:

1. Copy and paste the **Background Mesh** node into the **Operations** node and rename the **Copy of Background Mesh** node to `Overset Mesh`.
2. Select the **Operations** > **Overset Mesh** node and set **Input Parts** to **Overset**.
3. Select the **Operations** > **Overset Mesh** > **Meshers** > **Prism Layer Mesher** node and set **Distribution Mode** to **Wall Thickness**.
4. Edit the **Operations** > **Overset Mesh** > **Default Controls** node and set the following properties:

Node	Property	Setting
Base Size	**Value**	`0.5 m`
Minimum Surface Size	**Percentage of Base**	`25.0`
Number of Prism Layers	**Number of Prism Layers**	`15`
Prism Layer Total Thickness	**Percentage of Base**	`15.0`
Volume Growth Rate	**Default Growth Rate**	**Slow**
	Surface Growth Rate	**Slow**
Prism Layer Near Wall Thickness	**Value**	`1.0E-4 m`

5. Rename the **Operations** > **Overset Mesh** > **Custome Controls** node to `Overlap`.

   You can reuse the volumetric control that you created for the background mesh.

6. Following the same procedure as for the background mesh, create three additional new volumetric controls and give them the following names:
   - `Airfoil`.
   - `Leading Edge`.
   - `Trailing Edge`.
7. In the **Controls** node of all four volumetric controls, set the following properties:

Node	Property	Setting
Surface Remesher	**Customize Size**	Activated
Trimmer	**Customize isotropic size**	Activated

8. For each volumetric control, select the following parts and the following values for **Values** > **Custom Size**:

Volumetric Control	Part	Percentage of Base
Overlap	**Overlap**	`12.5`
Airfoil	**Airfoil**	`5.0`
Leading Edge	**Leading Edge**	`1.0`
Trailing Edge	**Trailing Edge**	`2.0`

Generating the Mesh

With both Automated Mesh Operations for the background mesh and the overset mesh set up, you generate the volume mesh. You then convert the mesh from 3D to 2D to simplify the solution.

To generate the mesh:

1. To build the volume mesh, click the **Generate Volume Mesh** button in the toolbar. This process can take a few minutes.
2. To view the mesh, create a mesh scene.
3. Zoom in on the Overset boundary to compare the cell size of the overset mesh to the background mesh at this boundary.

You can convert the mesh from 3D to 2D to simplify the solution.

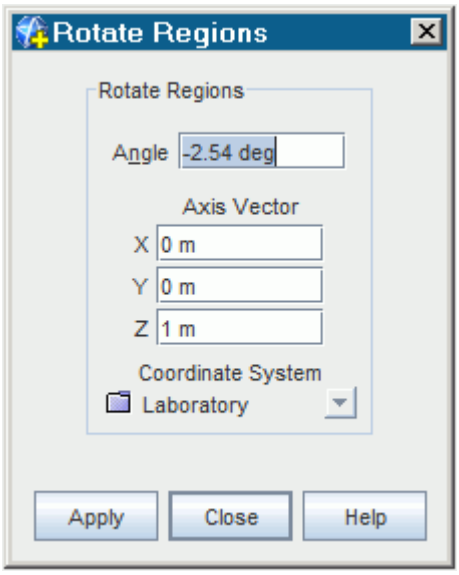
4. In the menu bar, select **Mesh** > **Convert to 2D**
5. In the *Convert Regions to 2D* dialog, tick the **Delete 3D regions after conversion** option and click **OK**.
6. Save the simulation.

Specifying Angle of Attack

Rotate the Overset region to correspond to a 2.54° angle of attack of the airfoil.

The steps presented here demonstrate how using overset meshes allows you to change the relative position of bodies without the need to generate a mesh.

1. Right-click the **Regions** > **Overset 2D** node and select **Transform** > **Rotate**
2. In the **Rotate Regions** dialog, set the **Angle** to `-2.54 deg`.



3. Click **Apply**, and then click **Close**.

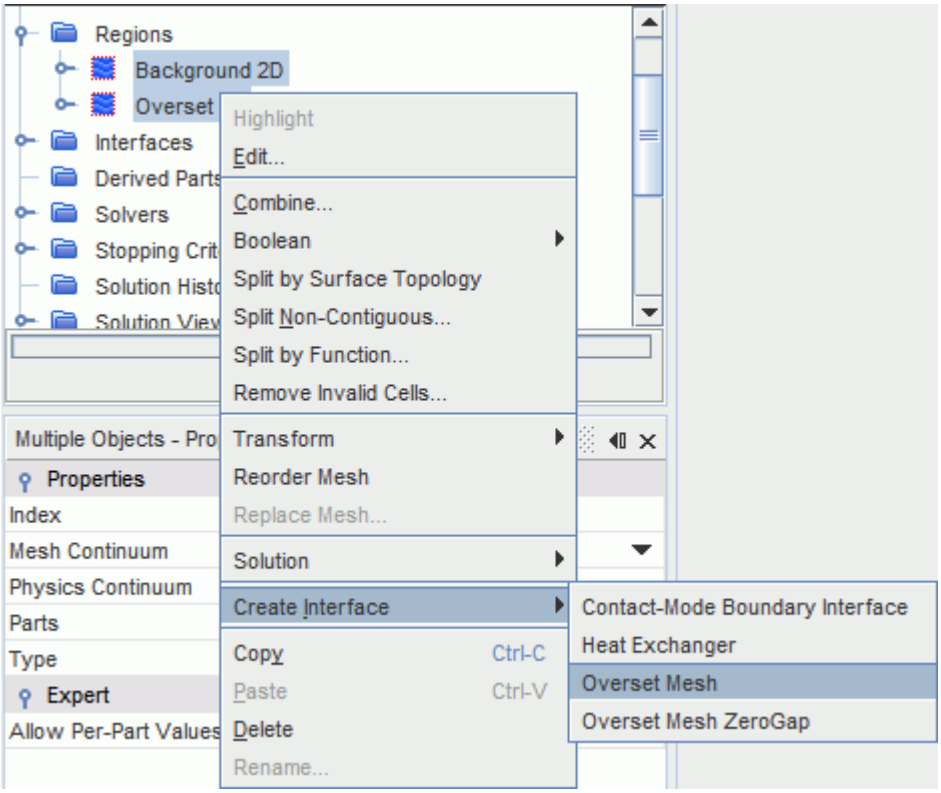
Coupling the Overset Region

Overset regions are coupled to the background region using the Overset Mesh interface.

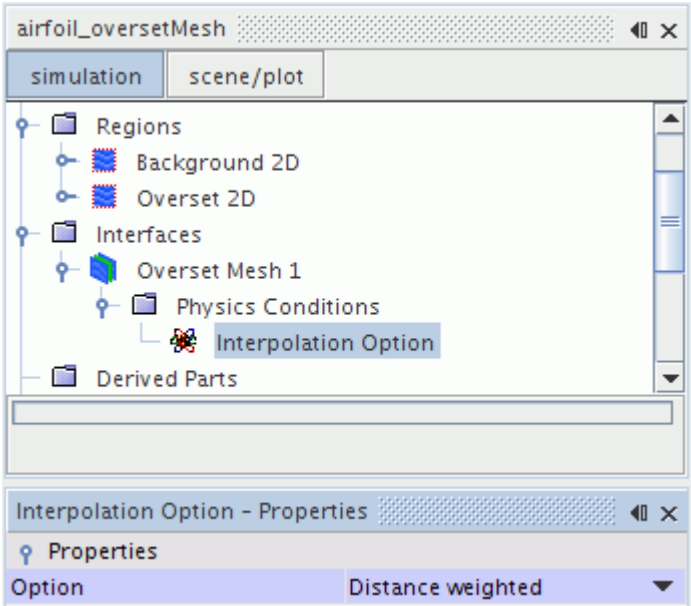
Normally, this interface can be created before generating the background and overset meshes. To activate this interface feature, you must create a physics continuum. Current implementation requires that the background and overset regions use the same physics continuum. In this tutorial, you are creating the Overset Mesh interface after converting the mesh to 2D as this step deletes all previously created interfaces.

To create the interface:

1. Multi-select the **Background 2D** and **Overset 2D** regions.
2. Right-click the selected regions and choose **Create Interface** > **Overset Mesh**.



An **Overset Mesh 1** node appears under the **Interfaces**. Use the default **Distance Weighted Interpolation Option** for the mesh coupling.



When you initialize the Overset Mesh interface, STAR-CCM+ chooses acceptor cells on the overset region. Cells in the background region that are overlapped by the overset region are deactivated.

Contents:

Analyzing Overset Mesh

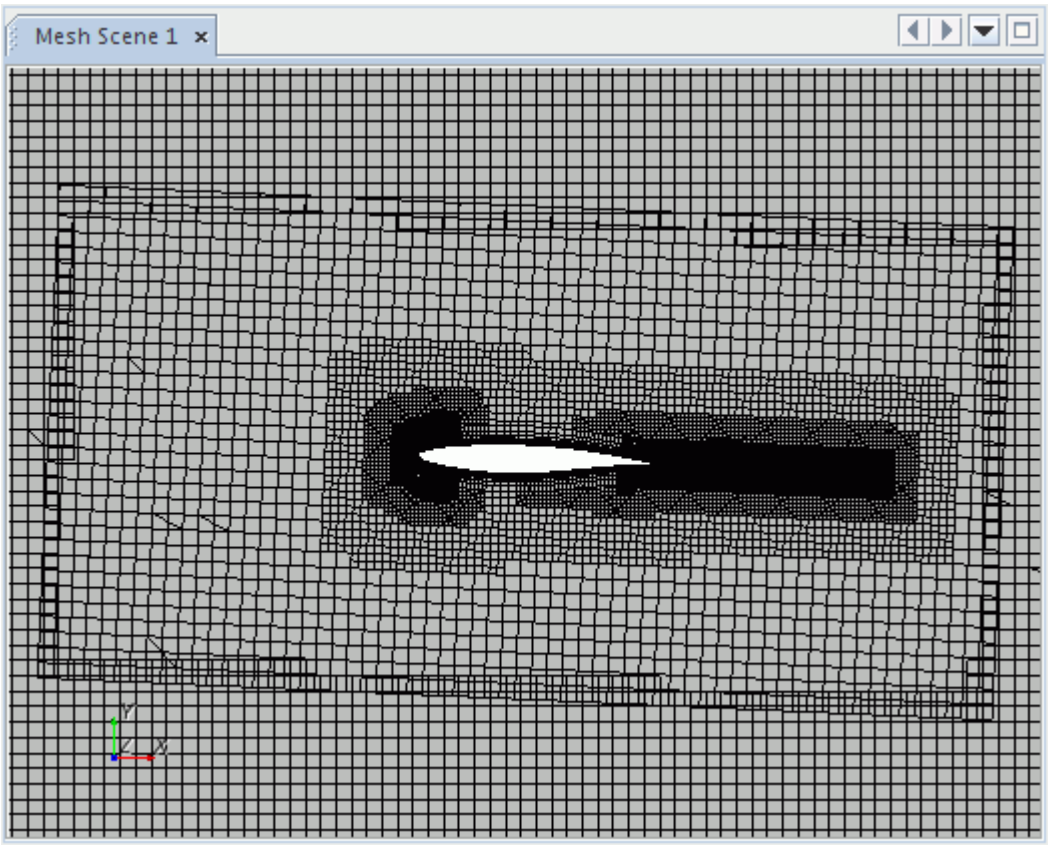
Analyzing Overset Mesh

Display cell status in a scalar scene.

To analyze the mesh before running the simulation, initialize the Overset Mesh interface:

1. Right-click the **Interfaces** node and select **Initialize**.

   Notice that the background mesh within the airfoil part has been removed in *Mesh Scene 1.*

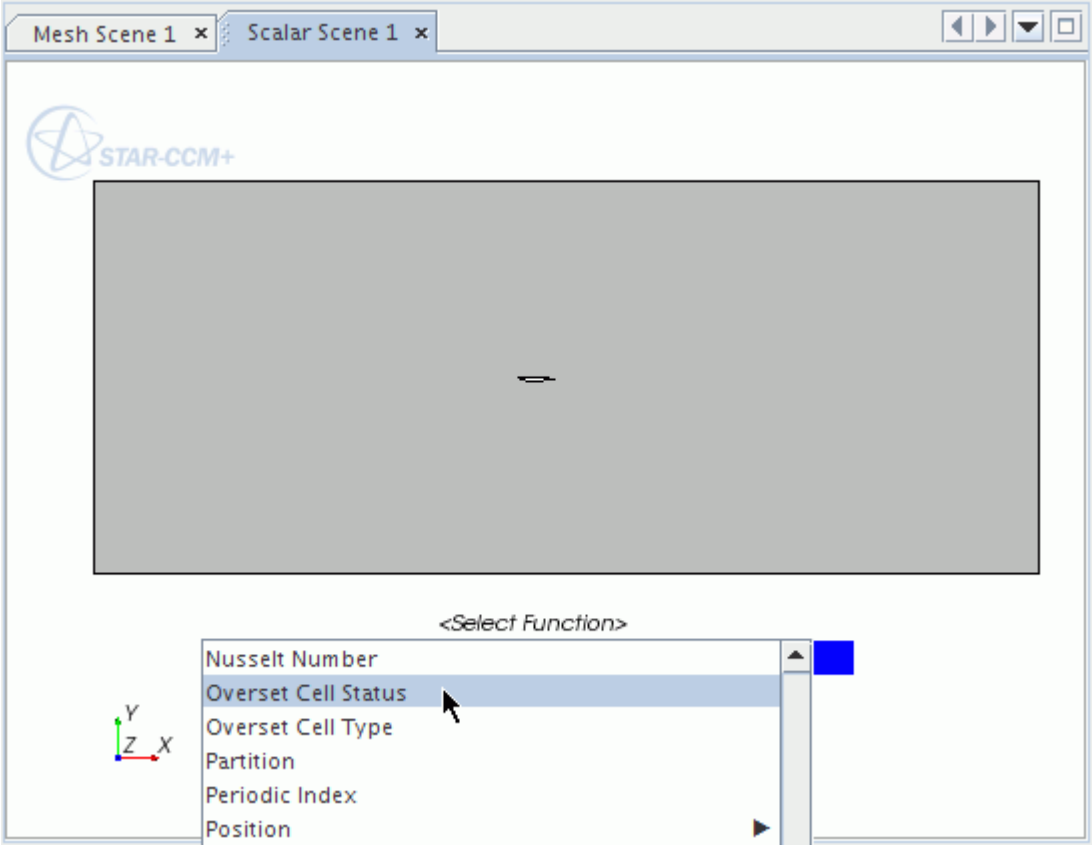


Analyze the overset mesh using the Overset Cell Status field function.

2. Create a scalar scene.

   The *Scalar Scene 1* display appears.

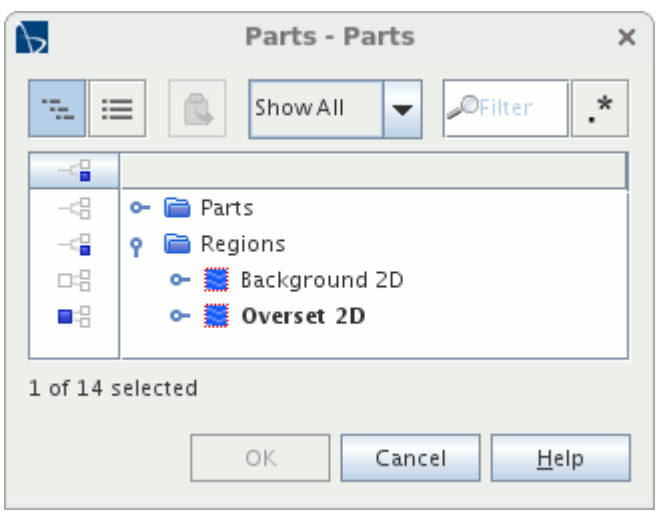
3. Right-click the scalar bar at the bottom of the display and select **Overset Cell Status** from the pop-up menu.



4. Select the **Scalar Scene 1** > **Displayers** > **Scalar 1** node and set *Representation* to **Overset Mesh**.

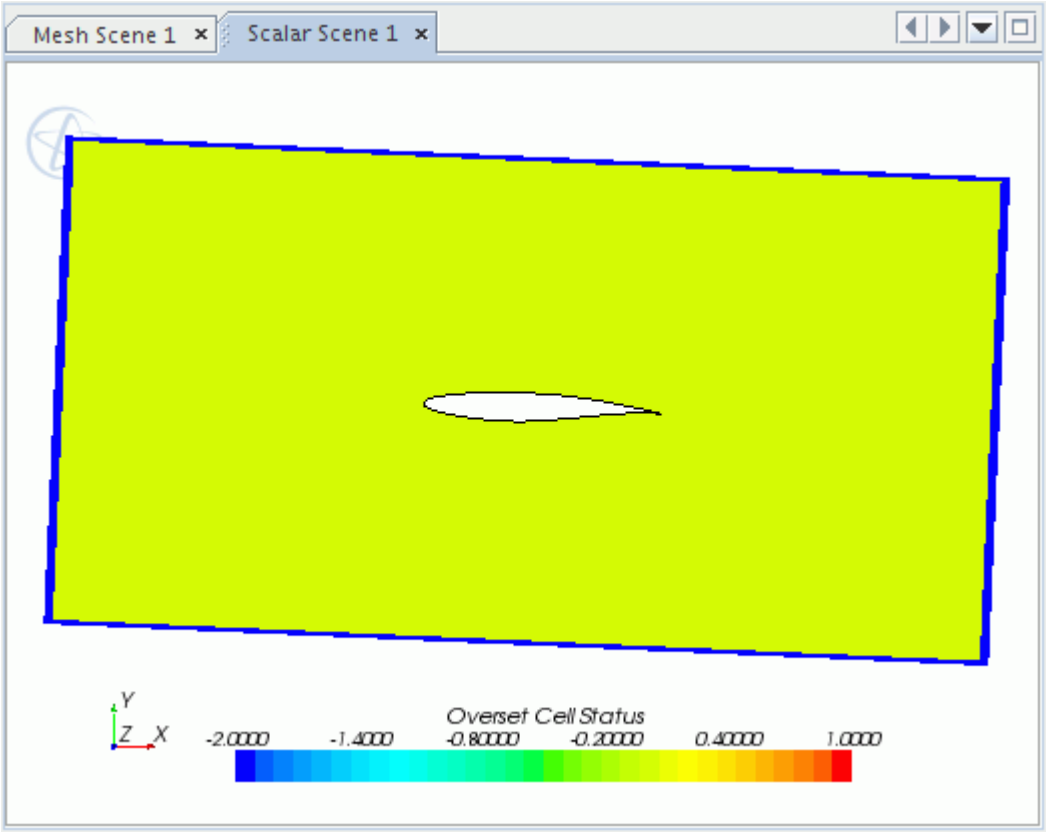
First look at the cell status in the overset region.

5. Select the **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Parts** node and tick the **Overset 2D** region in the **Parts** property.



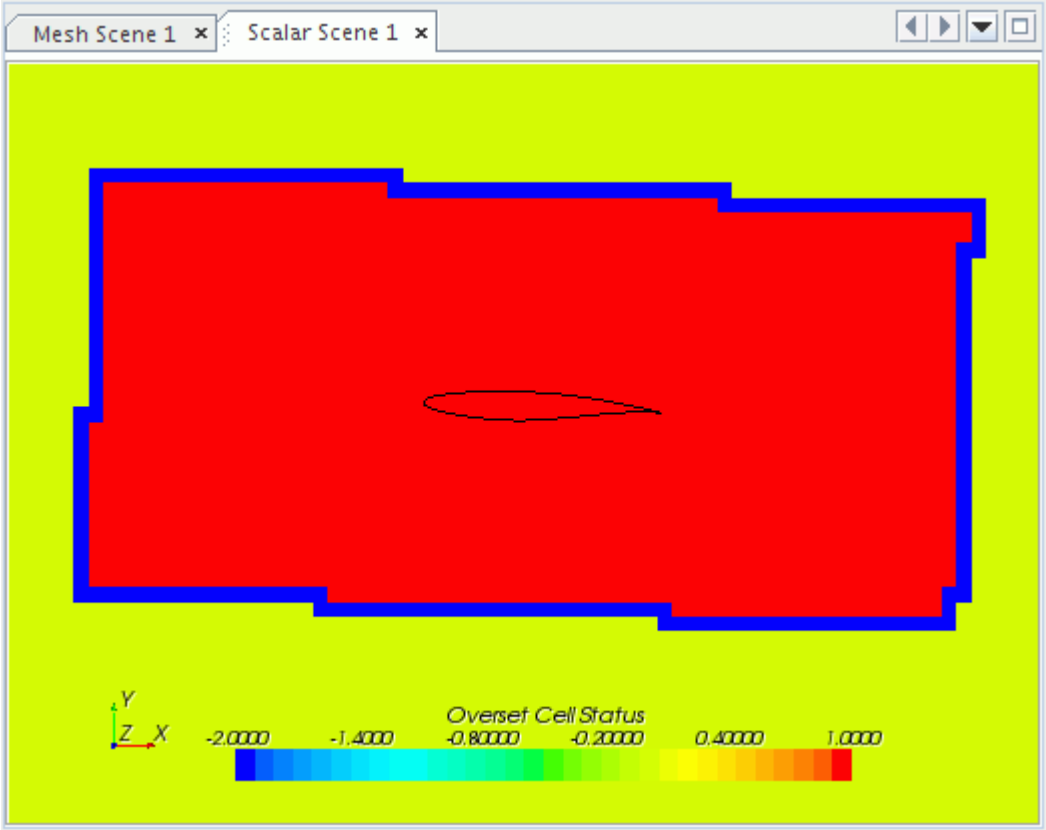
6. Select the **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Scalar Field** node and set the **Min** to `-2.0` and the **Max** to `1.0`.

Acceptor cells outline the overset region in blue.



7. To see the background region cell status, change the **Parts** to **Background 2D** within the **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Parts** node.

   The red cells in the background mesh are inactive.



Setting up Physics Models

Models define the spatial and temporal solution methods and the physical properties of the flow.

In this example, the flow is steady, turbulent, and compressible. Use the default Spalart-Allmaras Turbulence model and the ideal gas model. The analysis also uses the coupled solver, which is recommended for all supersonic and transonic compressible flows.

By default, a continuum called **Physics 1** is created when the three-dimensional mesh is generated and a **Physics 1 2D** continuum is created when the mesh is converted to two-dimensional. You can delete the **Physics 1** continuum, as only one is used in this simulation.

1. Delete the **Continua** > **Physics 1** node.

To use a more appropriate name for the **Physics 1 2D** continuum:

2. Rename the **Continua** > **Physics 1 2D** node to `Airfoil`

You can now choose the physics models for the continuum.

3. For the **Airfoil** continuum, select the following models, in order:

Group Box	Model
Space	**Two-Dimensional** (Selected automatically)

Time	**Steady**
Material	**Gas**
Flow	**Coupled Flow**
Gradient Metrics	**Gradients** (Selected automatically)
Equation of State	**Ideal Gas**
Viscous Regime	**Turbulent** **Reynolds-Averaged Navier-Stokes** (Selected automatically)
Reynolds-Averaged Turbulence	**Spalart-Allmaras Turbulence** **Standard Spalart-Allmaras** (Selected automatically) **Exact Wall Distance** (Selected automatically) **All y+ Wall Treatment** (Selected automatically)

4. Click **Close**.

Setting Material Properties

Set the Reynolds number to 6.5 x 10^6, changing the dynamic viscosity

Use the following steps:

1. Open the **Continua** > **Airfoil** > **Models** > **Gas** > **Air** node.
2. Edit the **Material Properties** node and set the following properties:

Node	Property	Setting
Dynamic Viscosity	*Method*	**Constant**
└ **Constant**	*Value*	`4.58E-5 Pa-s`
Thermal Conductivity	*Method*	**Constant**
└ **Constant**	*Value*	`0.0638414 W/m-K`

Setting Initial Conditions

As in the Transonic Flow tutorial, initialize the velocity field to 250.0 m/s.

1. Edit the **Continua** > **Airfoil** > **Initial Conditions** node and set the following properties:

Node	Property	Setting
Velocity	*Value*	`[252.0,0.0]`
Static Temperature	*Value*	`300 K`

The default values for the remaining initial conditions are suitable for this tutorial.

Setting Boundary Conditions

Set the conditions at the boundaries of the airfoil and the domain.

The geometry used for this tutorial has only two boundaries for the physics:

- A wall boundary representing the surface of the airfoil and defined in the Overset region.
- A freestream boundary at the external edge of the entire solution domain defined in the Background region.

Set the freestream conditions on the outer perimeter of the Background region.

1. Edit the **Regions** > **Background 2D** > **Boundaries** > **Outside** > **Physics Values** node and set the following properties:

Node	Property	Setting
Mach Number	*Value*	`0.725`
Static Temperature	*Value*	`300 K`

All other conditions for the freestream boundary and the default wall are suitable for this problem.

Setting Solver Parameters

The simplicity of this problem allows a rapidly converging solution to be obtained using a larger Courant number.

In problems involving more complex geometries or physics, attempting to shorten the run time in this way may cause the solution to diverge.

To increase the Courant number:

1. Select the **Solvers** > **Coupled Implicit** node and set *Courant Number* to `20.0`.
2. Save the simulation.

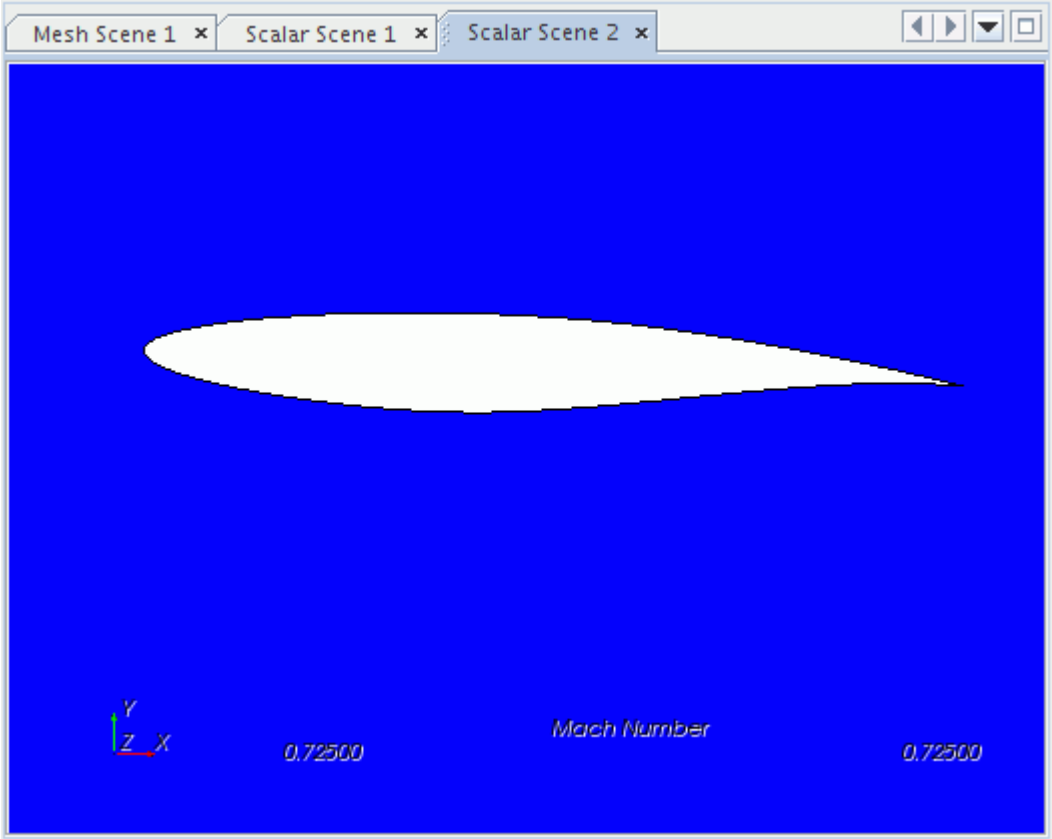
Preparing a Scalar Scene

View the Mach number profile during the run to monitor the supersonic flow region above the airfoil.

Use the following steps:

1. Create a scalar scene.

   A new *Scalar Scene* display appears.
2. Right-click the scalar bar at the bottom of the display and select **Mach Number** > **Lab Reference Frame** from the pop-up menu.
3. Initialize the run by clicking **Initialize Solution** in the toolbar, then use the middle mouse button to zoom in on the airfoil in the center of the scalar scene.



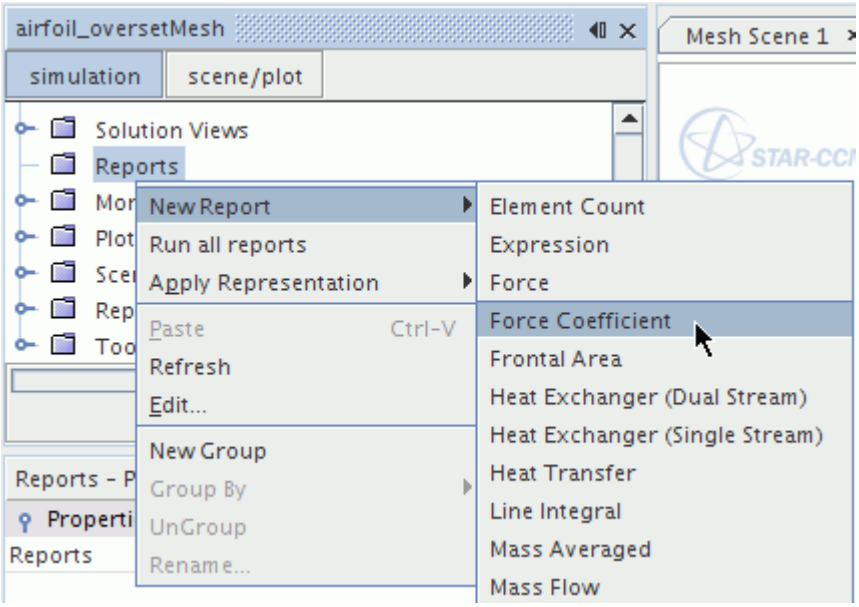
4. To change the style of the Mach number contours, select the **Scalar Scene 2** > **Displayers** > **Scalar 1** node and set *Contour Style* to **Smooth Filled**.

Monitoring Convergence

To determine when the problem has converged, plot the lift and drag coefficients.

Use the following steps:

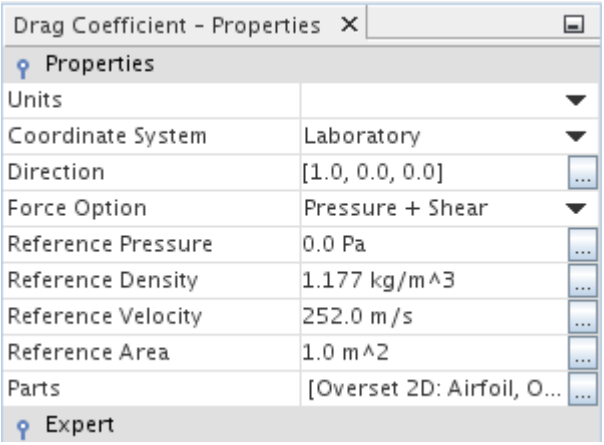
1. Right-click the **Reports** node and select **New Report** > **Force Coefficient**.



A new report node named **Force Coefficient 1** is created.

2. Rename this node `Drag Coefficient`
3. Select the **Drag Coefficient** node and set the following properties:

Property	Setting
Reference Density	`1.177 kg/m^3`
Reference Velocity	`252 m/s`
Parts	**Regions** > **Overset 2D** > **Airfoil**



4. Right-click the **Drag Coefficient** node and select **Create Monitor and Plot from Report**.

   A new plot node appears named **Drag Coefficient Monitor Plot**.

5. Repeat the steps described above to create and display a plot for the lift coefficient. Use the same settings as for the drag coefficient except rename the report node **Lift Coefficient** and set its *Direction* property to `[0.0, 1.0, 0.0]`
6. Save the simulation.

Running the Simulation

You are now ready to run the simulation.

Use the following steps:

1. Click **Run** in the toolbar.
   The Residuals display is created automatically and shows the progress of the solver.

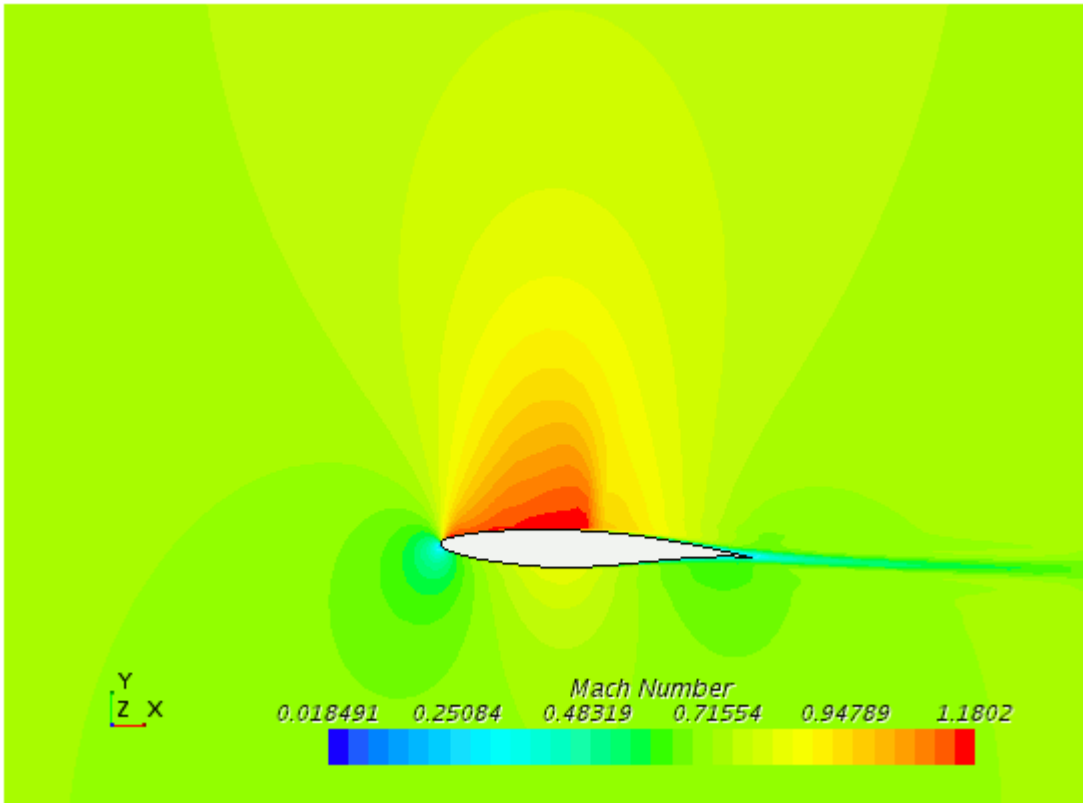
During the run, it is possible to stop the analysis by clicking (**Stop**) in the toolbar. If you do halt the simulation, it can be continued later by clicking (**Run**). If left alone, the simulation continues until 1000 iterations have been completed.

2. Save the simulation after the solver has finished.

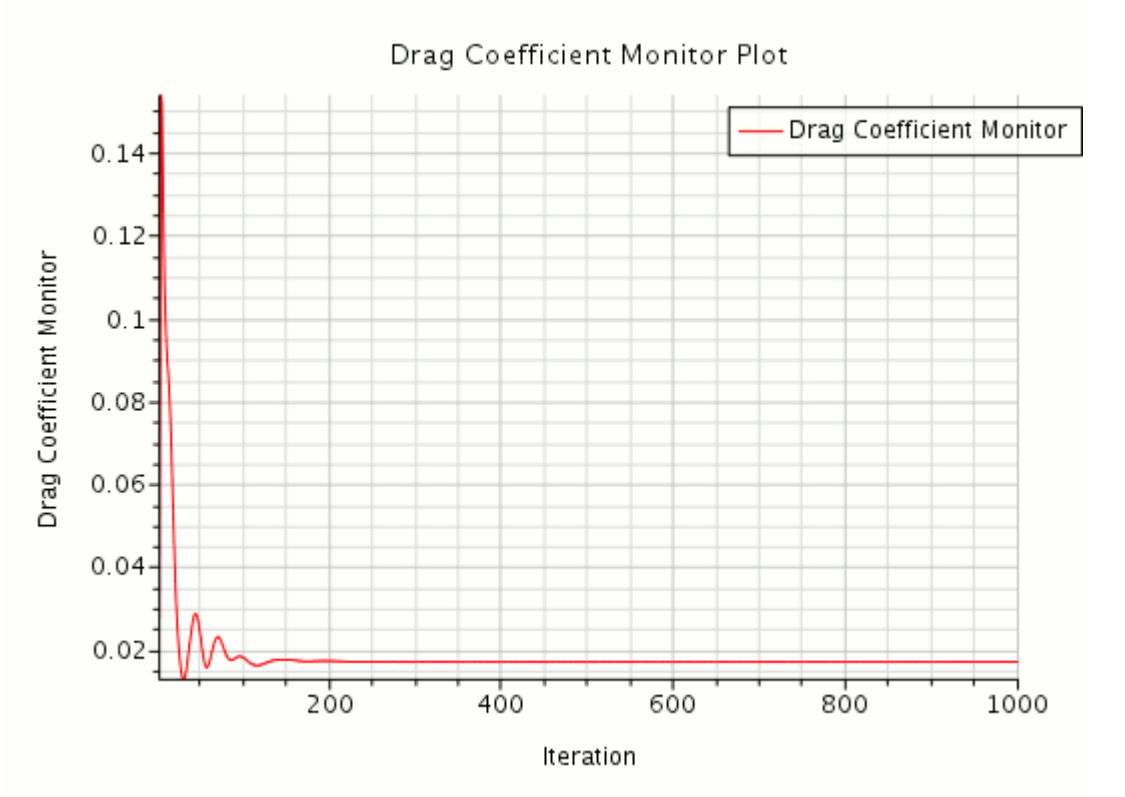
Visualizing the Results

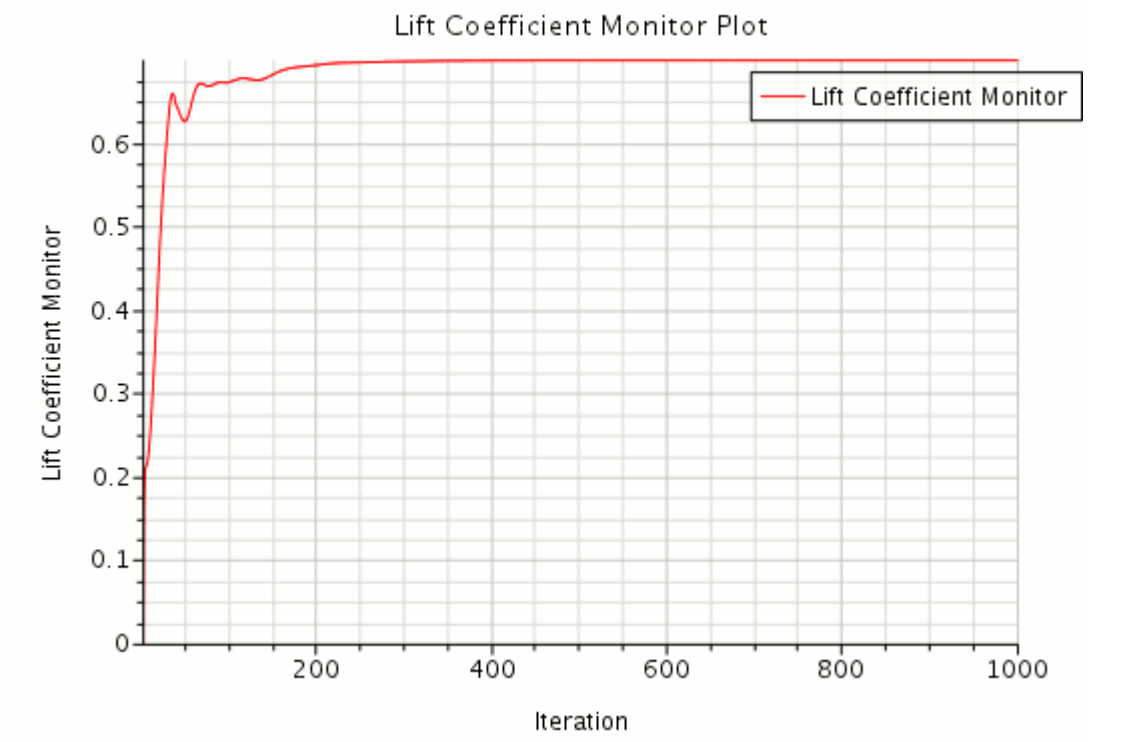
The *Scalar Scene* display shows the Mach number profile at the end of the run.

The profile shows the transonic flow around the airfoil, including the shock wave produced behind the supersonic region on the upper surface.



The lift and drag coefficient plots provide a check for convergence. As there is little variation in both monitors, it is reasonable to conclude that the solution has converged.


Drag Coefficient Monitor Plot
Drag Coefficient Monitor
Drag Coefficient Monitor
0.14
0.12
0.1
0.08
0.06
0.04
0.02
200
400
600
800
1000
Iteration

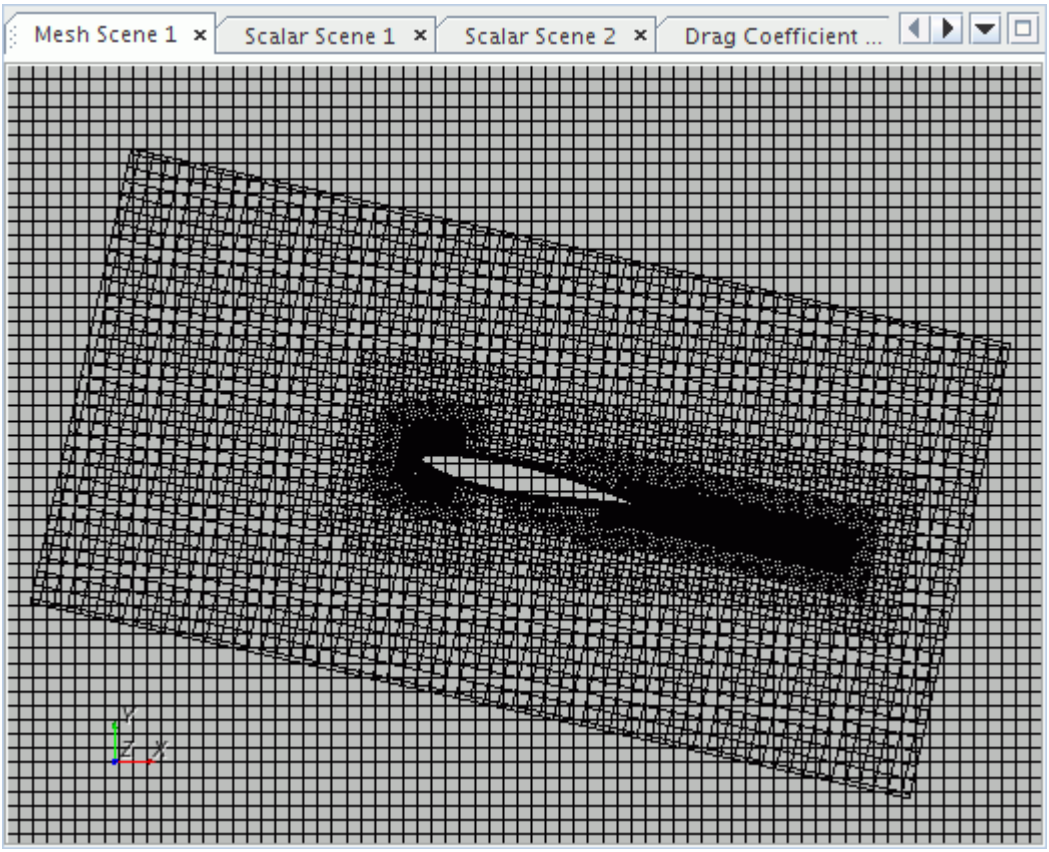

Lift Coefficient Monitor Plot
Lift Coefficient Monitor
Lift Coefficient Monitor
0.6
0.5
0.4
0.3
0.2
0.1
0
200
400
600
800
1000
Iteration

Changing Angle of Attack

Rotate the overset mesh.

One advantage of using an overset mesh is being able to transform regions without having to remesh. If, for instance, you want to study the airfoil at a different angle of attack, you can rotate the Overset region by a different angle and rerun the simulation.

1. Right-click the **Regions** > **Overset 2D** node and select **Transform** > **Rotate...** .
2. In the *Rotate Regions* dialog, set **Angle** to `-10 deg`.
3. Click **Apply**.



Note: The transformed overset region must be completely within that refined part of the background region where the background mesh is of the same size as the overset mesh.

Clearing and Rerunning the Simulation

Clear the solution and rerun the simulation using the same physics setup.

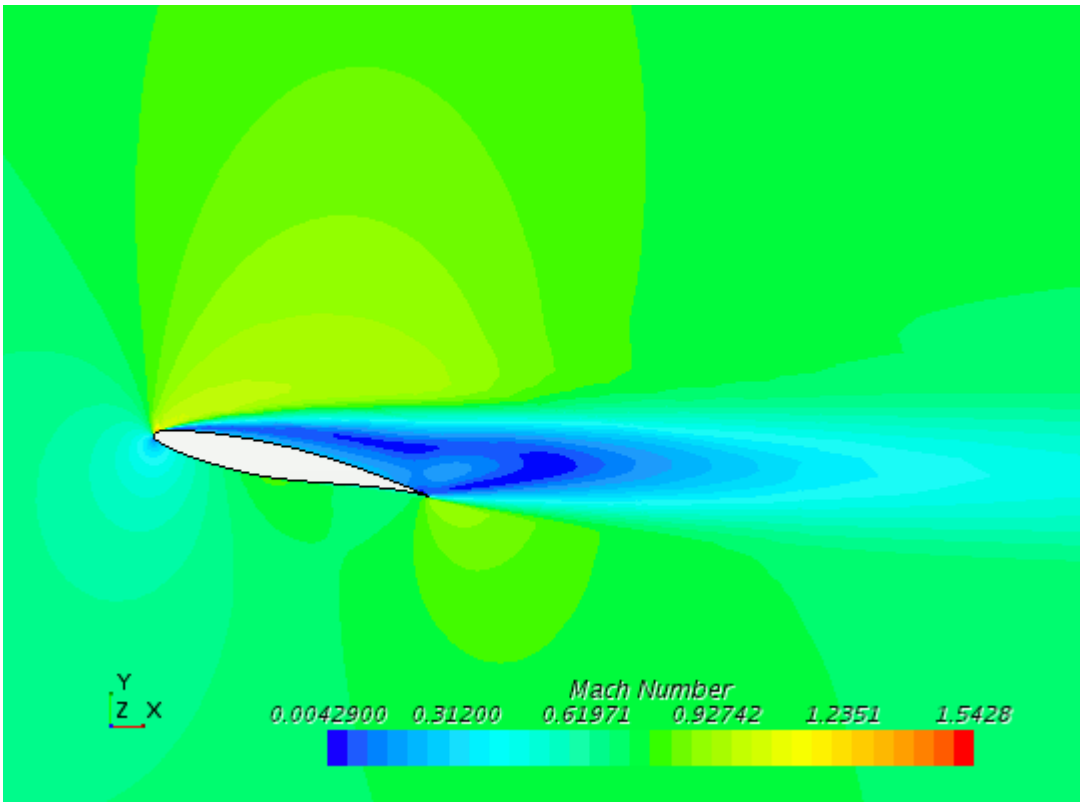
Use the following steps:

1. In the menu bar, select **Solution** > **Clear Solution**.
2. Click **Run** in the toolbar.

Visualizing the Results

Re-examine the scalar scene.

The scalar scene solution is shown below. You can see that if the angle of attack is increased further, the steep adverse pressure gradient on the upper surface of the airfoil causes the flow to separate close to the leading edge, resulting in a large wake.



Summary

This Transonic Flow tutorial demonstrated how to use overset mesh to simulate transonic flow over an airfoil

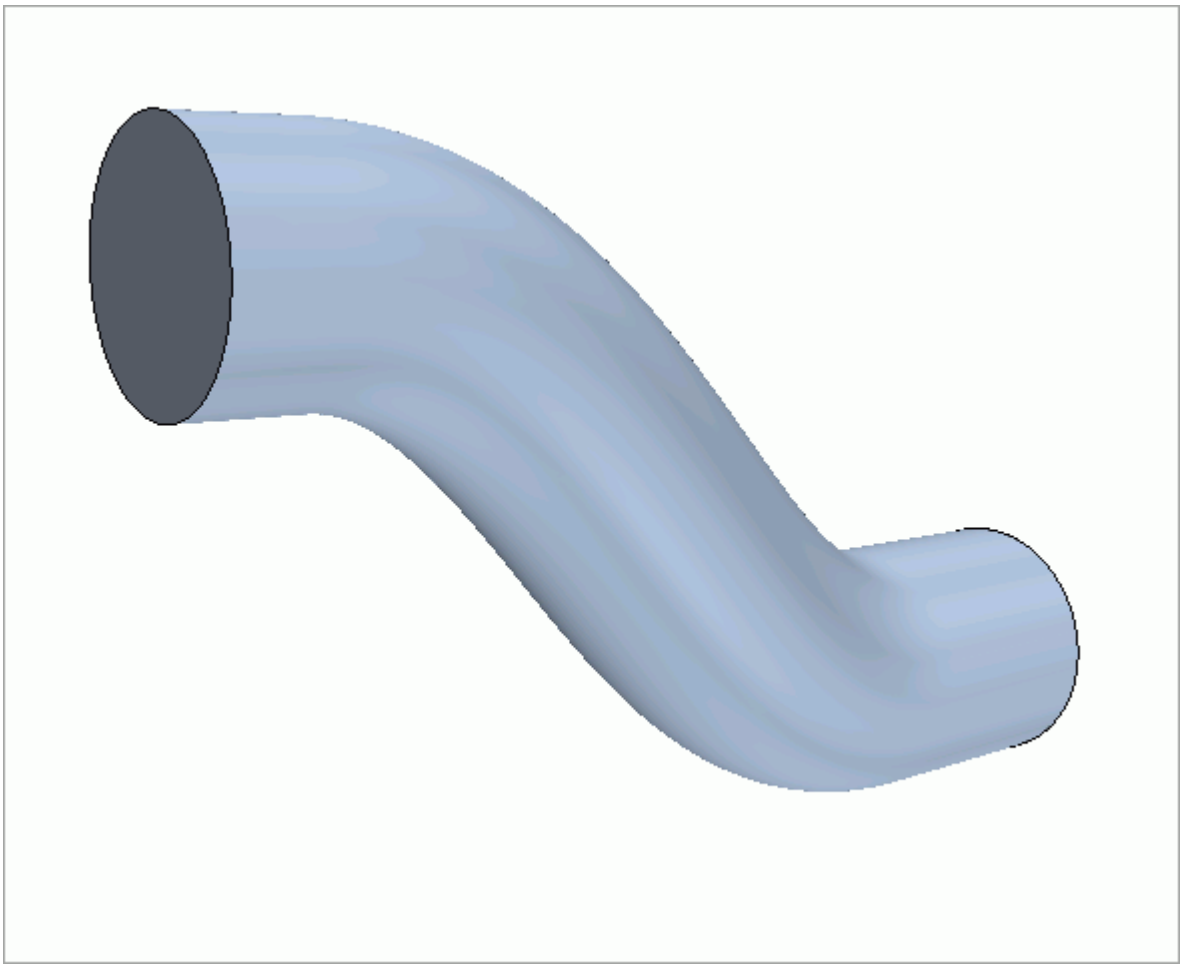
This tutorial introduced the following features of STAR-CCM+:

- Creating and analyzing an overset mesh for an airfoil problem.
- Defining models for a compressible flow problem.
- Defining the material properties required for the selected models.
- Setting solver parameters for a steady-state run.
- Analyzing results and convergence.
- Modifying angle of attack.

Adjoint Flow Solver: Flow through an S-Bend

This tutorial demonstrates the basic workflow for setting up an adjoint flow analysis.

The geometry is an s-bend duct with compressible air flowing through it. The flow enters the domain through a stagnation inlet boundary and leaves it through a pressure outlet boundary. A wall boundary condition is imposed on the duct surface. The case is run at a Mach number of 0.2.



An adjoint flow analysis is set up and run in two steps:

1. The setup of the primal flow solution corresponds to the setup of any flow solution simulation using the coupled solver. Ideally, the primal flow solution should be converged down to machine precision. This is important for the adjoint flow solver to subsequently run robustly and produce optimal results.

   Note: For industrial cases, this high level of convergence might not be achievable. The adjoint flow solver can still obtain good results as this depends on the individual case.

2. The adjoint solver setup is performed as a separate step after the primal flow solution has been run and converged. Separate solver parameters are available for the adjoint flow solver. The engineering

objective to be analyzed by the adjoint method is defined by use of cost functions. Additional field functions are created that can be used to visualize the adjoint flow solution.

Contents:

Importing the Volume Mesh
Selecting the Physics Models
Setting Boundary Conditions
Setting the Flow Solver Parameters and Stopping Criteria
Visualizing the Flow Solution
Running the Simulation
Visualization of Flow Results
Selecting the Adjoint Flow Model
Setting up the Adjoint Cost Function
Setting the Adjoint Solver Parameters and Stopping Criteria
Running the Adjoint Solver
Visualizing the Adjoint Flow Results
Summary

Importing the Volume Mesh

A cylindrical mesh has been prepared for this tutorial. You require a double-precision version of STAR-CCM+ to complete this tutorial.

To import the mesh and save the simulation:

1. Create a directory for the tutorial, called `ductAdjoint`.
2. Copy the mesh file `sBend.ccm` to this directory from the `doc/startutorialsdata/adjoint/data` subdirectory of your STAR-CCM+ installation directory.
3. Start up a double precision version of STAR-CCM+ and create a new simulation by clicking **File** > **New Simulation...**
4. Select **File** > **Import** > **Import Volume Mesh**.
5. In the *Open* dialog, browse to the `ductAdjoint` directory, select `sBend.ccm`, and click **Open**.

   The imported mesh appears in the *Geometry Scene 1* window.
6. Select **File** > **Save As** and save the simulation under the name `ductAdjointTutorial.sim`.

Selecting the Physics Models

Activate the models for the flow solution.

To select the physics models:

1. For the continuum **Continua** > **Physics 1** select the following models, in order:

Group Box	Model
Enabled Models	**Three Dimensional** (selected automatically)
Time	**Steady**
Material	**Gas**
Flow	**Coupled Flow**
	Gradients (selected automatically)
Equation of State	**Ideal Gas** **Coupled Energy** (selected automatically)
Viscous Regime	**Turbulent** **Reynolds-Averaged Navier-Stokes** (selected automatically)
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**
	Realizable K-Epsilon Two-Layer (selected automatically)
	Exact Wall Distance (selected automatically)
	Two-Layer All y+ Wall Treatment (selected automatically)

2. Click **Close**.

Setting Boundary Conditions

Set the inflow and outflow boundary conditions:.

Use the following steps:

Set the **Stagnation Inlet** boundary.

1. Edit the **Regions** > **Duct** > **Boundaries** node and set the following properties:

Node	Property	Setting
Stagnation_in	*Type*	**Stagnation Inlet**
⌊ **Physics Values**		
⌊ **Total Pressure**	*Value*	`2865.6 Pa`
⌊ **Total Temperature**	*Value*	`275.4 K`

Set the **Pressure Outlet** boundary:

2. Select **Regions** > **Duct** > **Boundaries** and set the following properties:

Node	Property	Setting
Pressure_out	*Type*	**Pressure Outlet**
⌊ **Physics Values** > **Pressure**	*Value*	`0.0 Pa.`

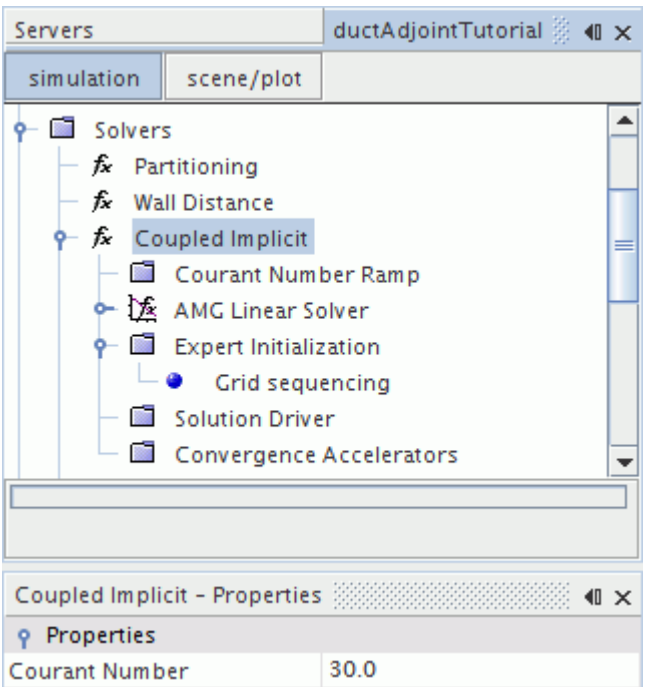
Setting the Flow Solver Parameters and Stopping Criteria

Set the Courant number, grid sequencing, and maximum time-steps.

Use the following steps:

To set the numerical parameters for the flow solution:

1. Select the **Solvers** > **Coupled Implicit** node and set *Courant Number* to `30`.
2. For a good initialization of the flow variables to enhance convergence, select the **Solvers** > **Coupled Implicit** > **Expert Initialization** and set *Method* to **Grid Sequencing**.



To set the stopping criteria:

3. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to `800`.

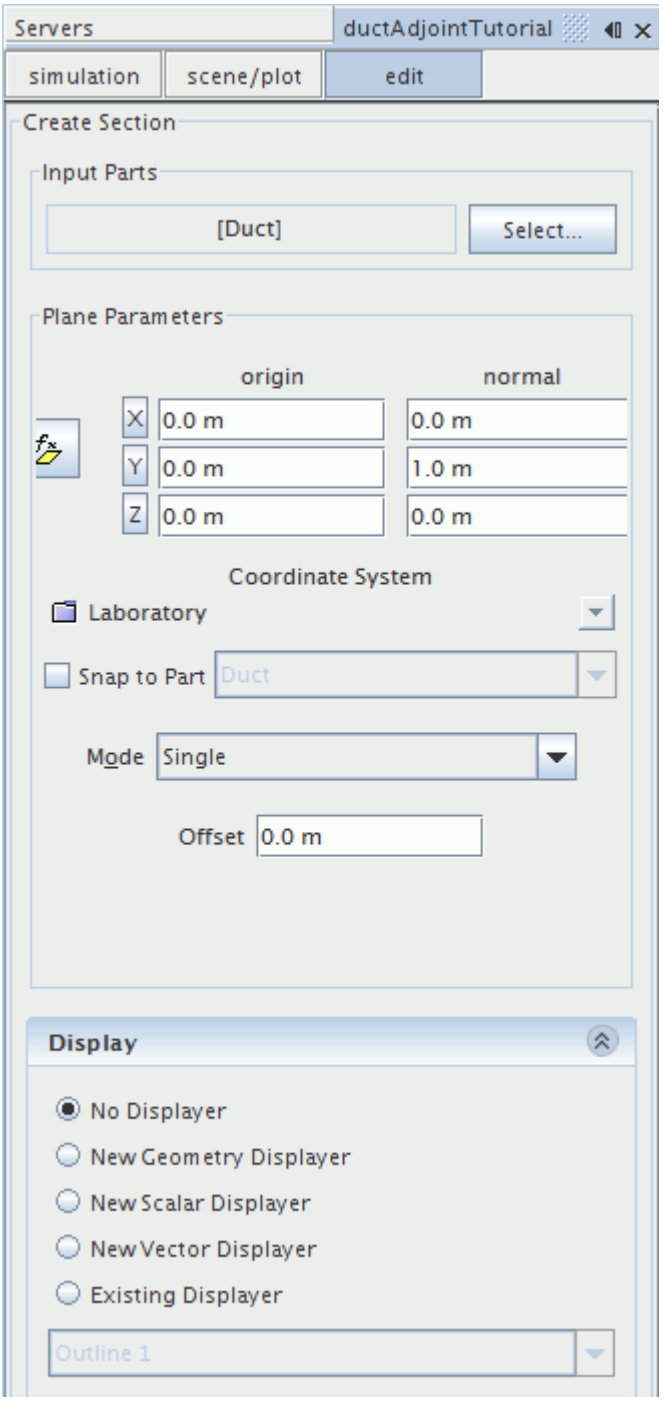
Visualizing the Flow Solution

Create scenes for the velocity vectors and pressure contours on a plane bisecting the s-bend duct.

Use the following steps:

Create a section through the s-bend:

1. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane...**
2. In the edit dialog under the **Plane Parameters** box, set the origin to [0.0,0.0,0.0] and the normal vector to [0.0,1.0,0.0].
3. Select **No Displayer** in the **Display** section.



4. Click **Create**, then click **Close**.

To visualize the velocity vectors, create a vector scene:

5. Right-click the **Scenes** node and select **New Scene** > **Vector**.
6. Click **Derived Parts** > **Plane Section** and drag and drop the **plane section** node into the open vector scene in the **Graphics** window.
7. In the **Displayer Selection** menu, select **Add to Vector 1**.
8. To obtain a side view of the section, press the `S` key.

To visualize the pressure contours, create a scalar scene:

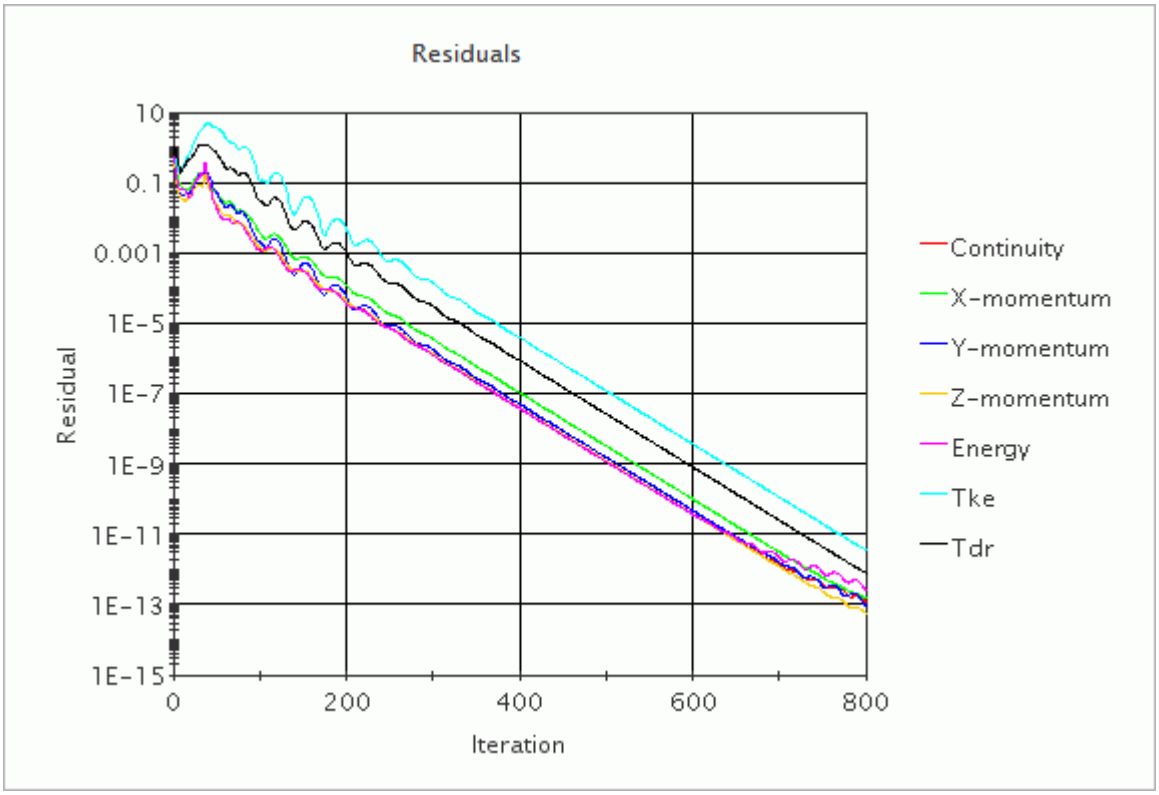
9. Right-click the **Scenes** node and select **New Scene** > **Scalar**.
10. Click **Derived Parts** > **Plane Section** and drag and drop the **plane section** node into the open scalar scene in the **Graphics** window.
11. In the **Displayer Selection** menu, select **Add to Scalar 1**.
12. Click **Scene/Plot**.
13. Select the **Displayers** > **Scalar 1** > **Scalar Field** node and set the *Function* to Pressure.
14. Select the **Displayers** > **Scalar 1** node and set *Contour Style* to **Smooth Filled**.
15. To obtain a side view of the section, press the `S` key.
16. Save the simulation.

Running the Simulation

The simulation is now ready to run.

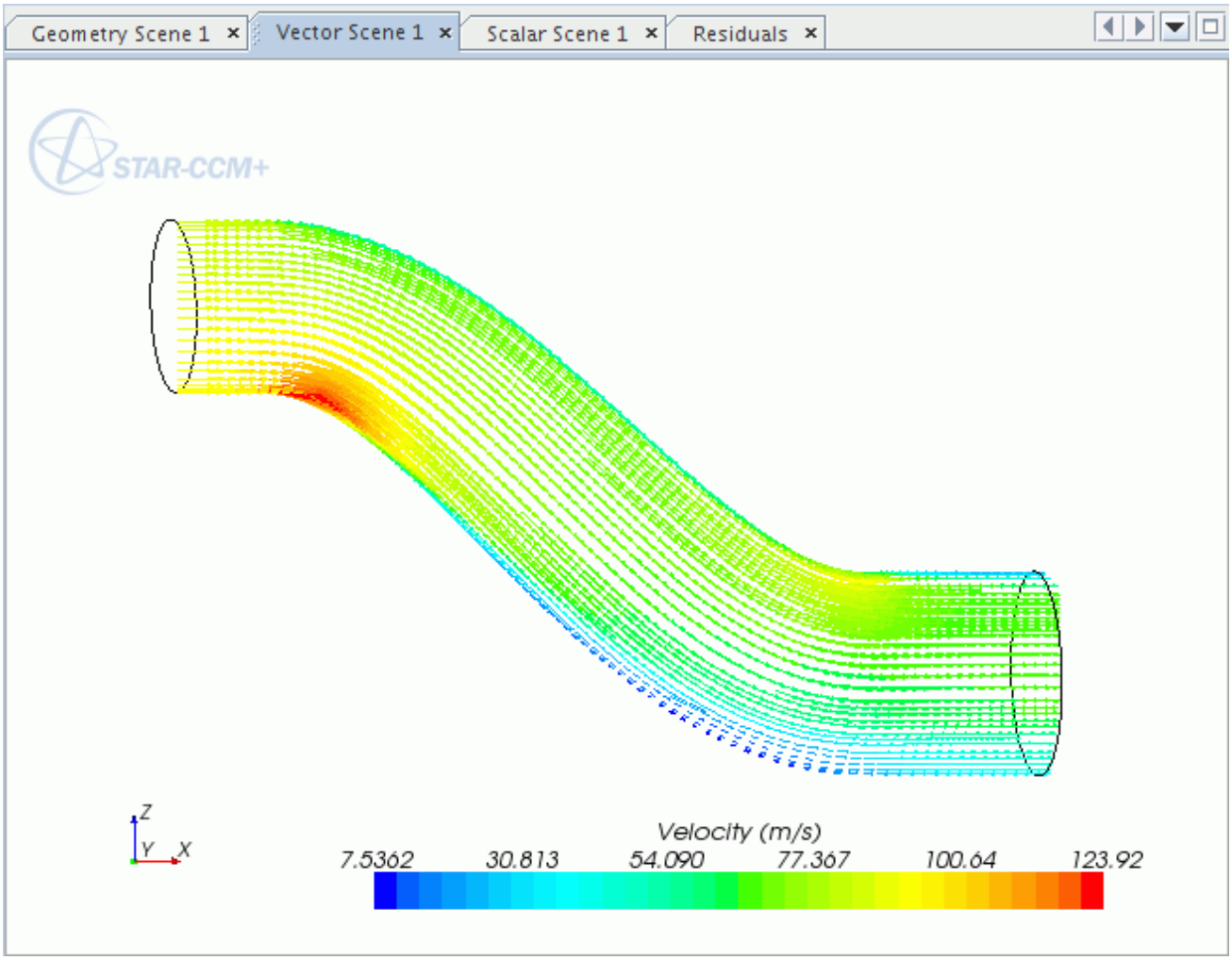
Click (**Run**).

While the simulation is running, you can click the tabs at the top of the **Graphics** window to view each of the scenes. The Residuals display is created automatically and shows the progress of the solvers.

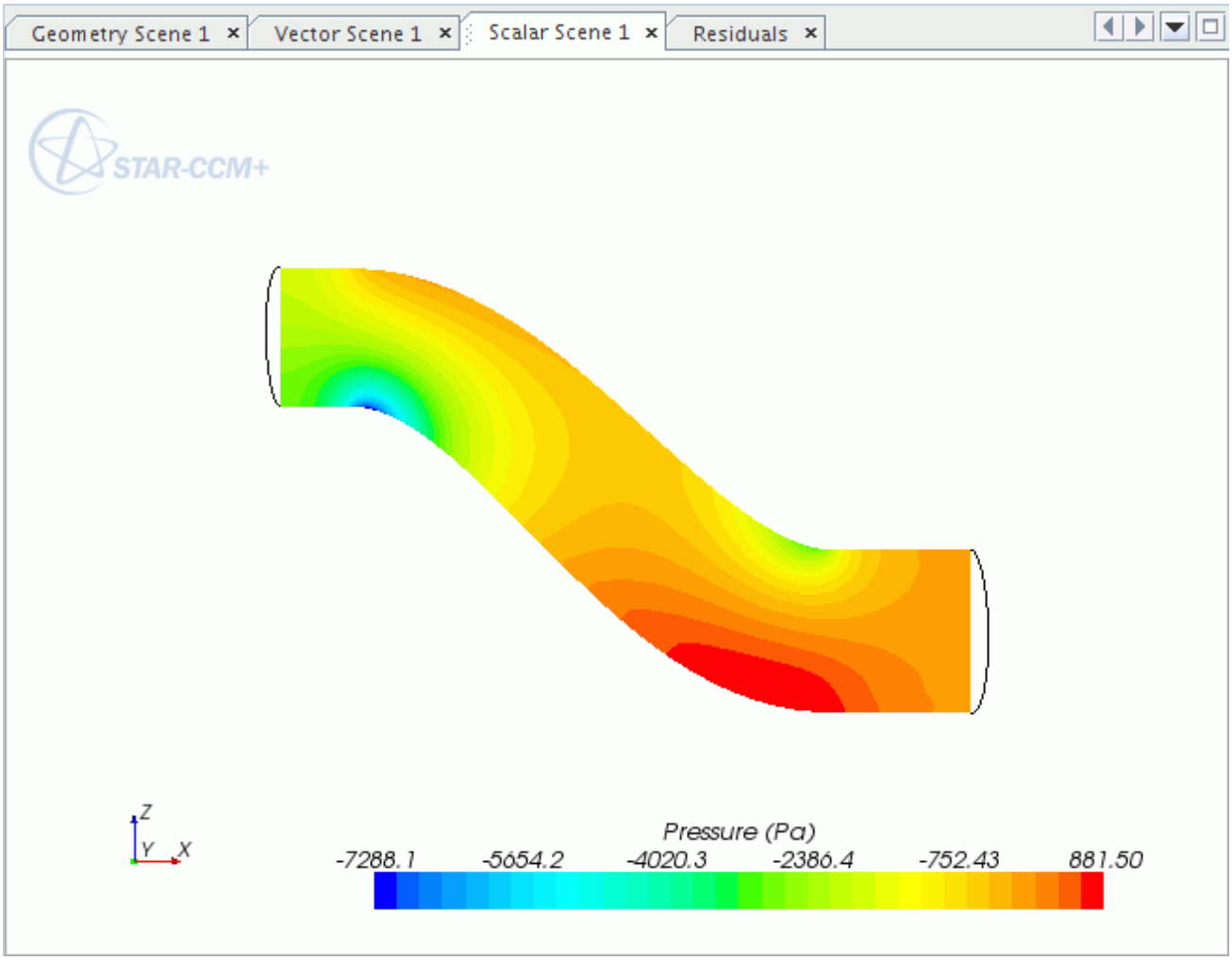


Visualization of Flow Results

Velocity vectors are shown below.



Pressure contours are shown below.

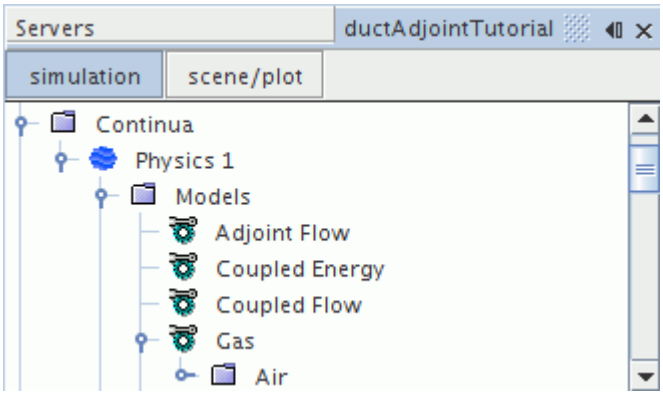


Selecting the Adjoint Flow Model

Activate the adjoint flow model in addition to the physical flow models already selected:

1. Right-click the **Continua** > **Physics 1** > **Models** node and choose **Select models...**
2. In the *Model Selection* dialog:
   a) Select **Adjoint Flow** from the **Optional Models** group box.
   b) Click **Close**.

An additional **Adjoint Flow** node has been created under **Continua** > **Physics 1** > **Models**.



Setting up the Adjoint Cost Function

A cost function represents the objective of the adjoint simulation with respect to a result.

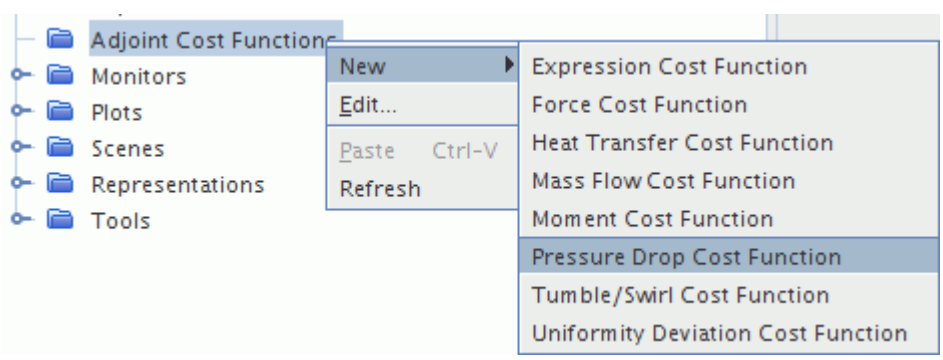
For this tutorial case, the objective is to reduce the pressure drop between inflow and outflow boundaries along the s-bend duct. Therefore, create a pressure drop cost function.

To set up the pressure drop cost function:

1. Right-click the **Reports** node and select **New Report** > **Pressure Drop**.
2. Select the **Reports** > **Pressure Drop 1** node and set the following properties:

Property	Setting
High Pressure	**[Duct:Stagnation_in]**
Low Pressure	**[Duct:Pressure_out]**

3. Right-click the **Adjoint Cost Functions** node and select **New** > **Pressure Drop Cost Function**.



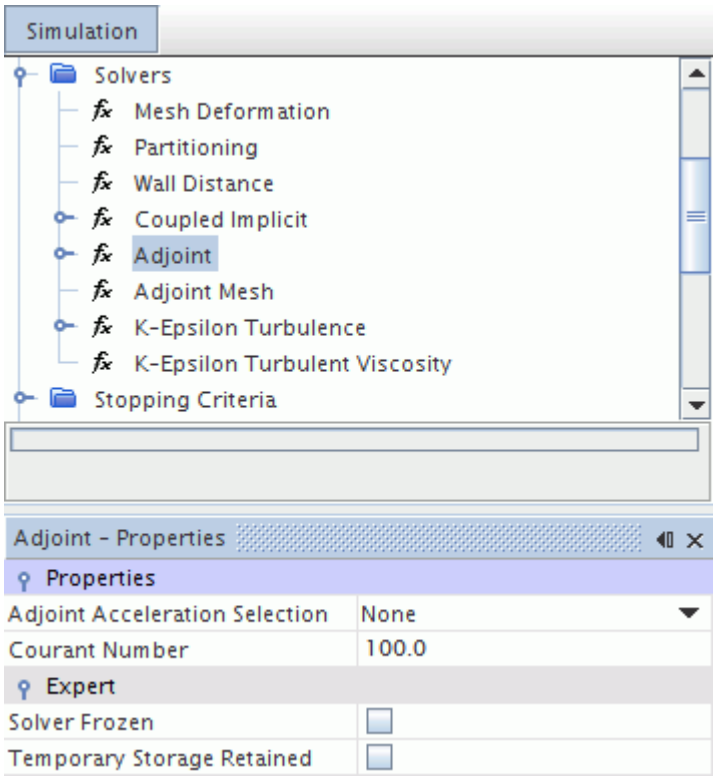
4. Select the **Adjoint Cost Functions** > **PressureDrop** node and set *Pressure Drop Report* to **Pressure Drop 1**.

Setting the Adjoint Solver Parameters and Stopping Criteria

You can find the numerical parameters specific to the adjoint solver under the **Solvers** > **Adjoint** node.

To view and confirm the current default settings:

1. Select the **Solvers** > **Adjoint** node.
2. Make sure that the *Courant Number* is set to `100.0`.



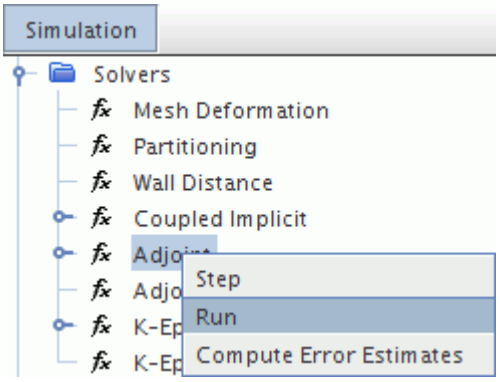
Set the stopping criteria for the adjoint solver run. Increase the number of maximum iterations:

3. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to `1200`.

Running the Adjoint Solver

The adjoint solver is now ready to run.

The actions to step or run the adjoint flow solver are directly attached to the **Adjoint** node. They are separated from the (**Run**) button that is used to start the primal flow solution. To run the adjoint solver, right-click the **Solvers** > **Adjoint** node and select **Run**.



Do not run the primal flow solver after the adjoint solution has begun.

The adjoint solver starts to iterate. To follow the progress of the residuals, click the Residuals tab in the **Graphics** window. The flow variables of the primal flow solution are frozen during the adjoint solver run.

Visualizing the Adjoint Flow Results

In order to analyze the adjoint flow results, additional adjoint field functions are created.

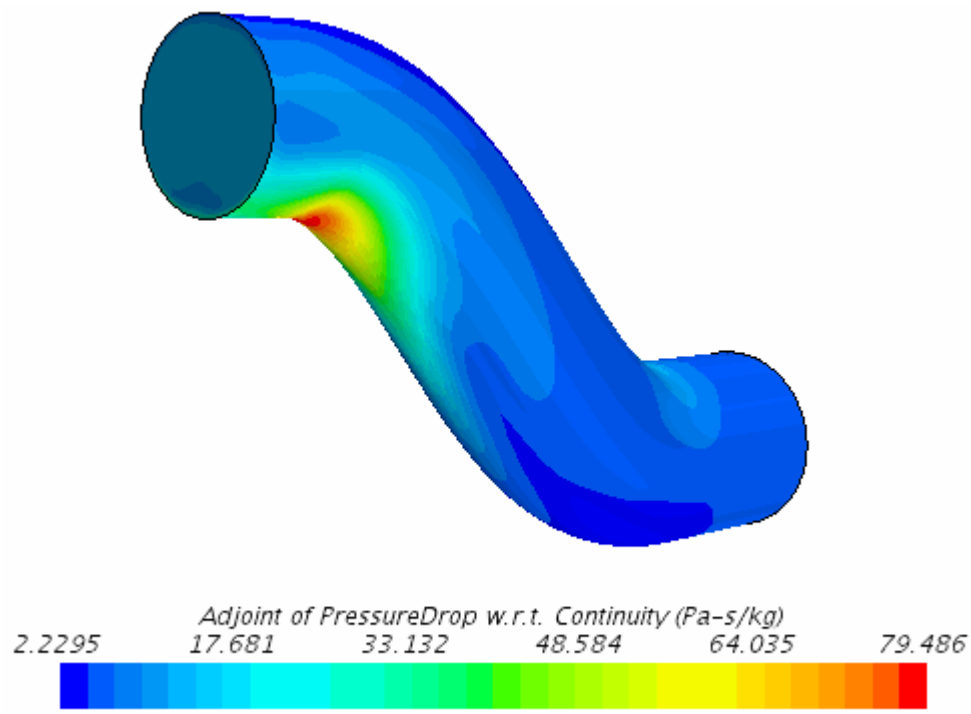
You can use these field functions in the scenes to visualize the results.

To visualize the adjoint of pressure drop with regard to continuity:

1. Create a new scalar scene by right-clicking the **Scenes** node and selecting **New Scene** > **Scalar**.
2. Click the **scene/plot** button.
3. Select the **Displayers** > **Scalar 1** node and set the following properties:

Node	Property	Setting
	Contour Style	**Smooth Filled**
└ **Parts**	**Parts**	**Regions** > **Duct** > **Boundaries** > **Pressure_out** **Regions** > **Duct** > **Boundaries** > **Stagnatio_in** **Regions** > **Duct** > **Boundaries** > **Wall**
└ **Scalar Field**	*Function*	**Adjoint** > **Pressure Drop** > **Adjoint of PressureDrop w.r.t. Continuity**

A contour plot of the adjoint of pressure drop with regard to continuity is shown below.

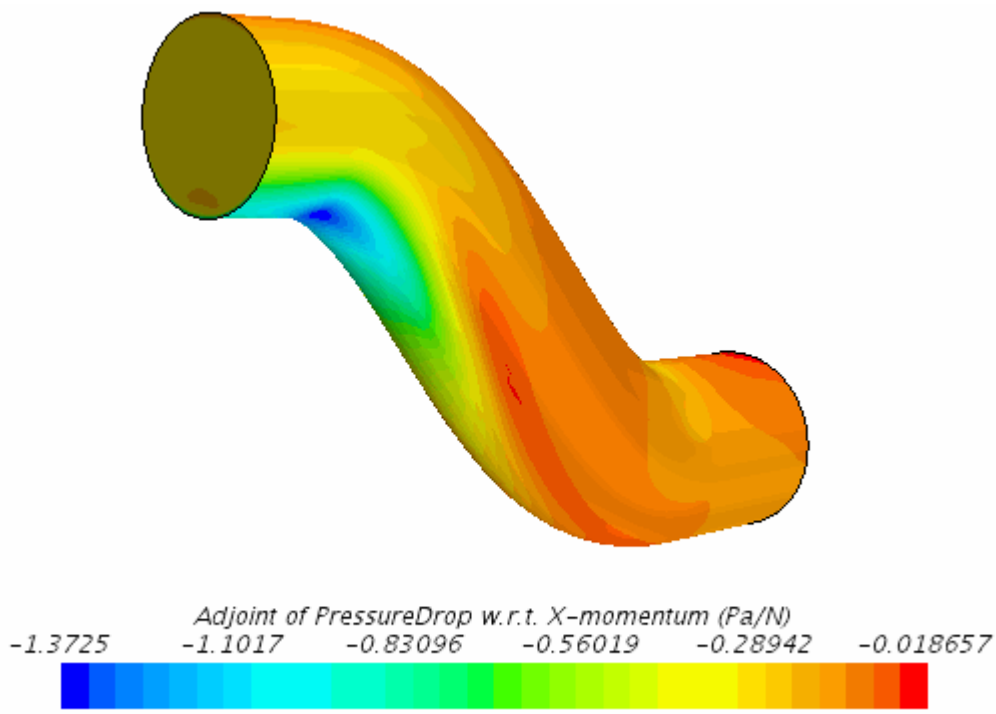


The objective is to reduce the pressure drop through the geometry. Directing mass away from areas with positive values decreases the pressure drop. Red areas indicate that directing mass away from here has the most effect on lessening the pressure drop.

To visualize the adjoint of pressure drop with regard to the momentum in x direction:

4. Select the **Displayers** > **Scalar 1** > **Scalar Field** node and set *Function* to **Adjoint** > **Pressure Drop** > **Adjoint of PressureDrop w.r.t. X-momentum**.

   A contour plot of the adjoint of pressure drop with regard to the momentum in x direction is shown below.



Directing x-momentum towards areas with negative values decreases the pressure drop. Blue areas indicate that directing x-momentum here has the most effect on lessening the pressure drop.

Summary

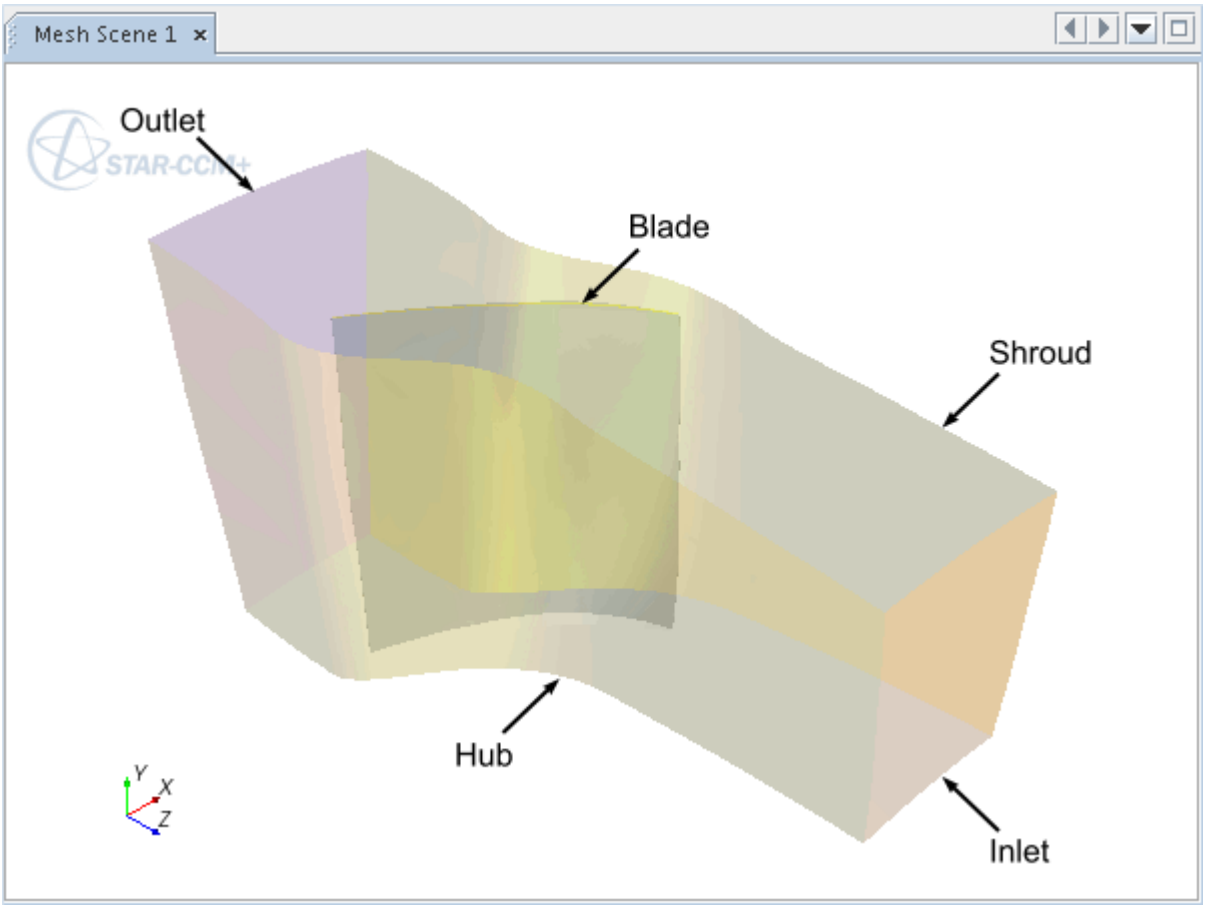
This tutorial has introduced the following STAR-CCM+ features:

- Importing and visualizing a volume mesh.
- Using the coupled solver Courant number and the Grid Sequencing Initialization to enhance convergence.
- Running the primal solver to machine precision.
- Selecting the physics model for adjoint flow analysis.
- Setting up the adjoint cost function for pressure drop.
- Running the adjoint flow solver.
- Visualizing the adjoint flow solution.

Turbo Wizard: Single Row

This tutorial illustrates the simulation of an axial compressor blade in STAR-CCM+ and goes through the process of creating a mesh for a single blade row.

The blade geometry that is used in this tutorial is Rotor37 [415]. The cross-sectional cuts are stored in the `Rotor37.estg` file and imported into STAR-CCM+. The problem geometry is shown below:



The tutorial starts with Importing the Turbo Blades and Generating the Mesh.

Contents:

Importing the Turbo Blades and Generating the Mesh
Checking the Models
Visualizing the Mesh
Setting Up Solver and Stopping Criteria
Visualizing the Solution
Running the Simulation
Analyzing Results
Summary

Single Row Tutorial Bibliography

Importing the Turbo Blades and Generating the Mesh

Create a subdirectory for the tutorial called `axialSingleRow`.

Copy the `Rotor37.estg`, `inlet.bndy`, `exit.bndy`, `hub.bndy` and `tip.bndy` files from the `doc/startutorialsdata/compressibleFlow/data/SingleRow` subdirectory of your STAR-CCM+ installation into this directory.

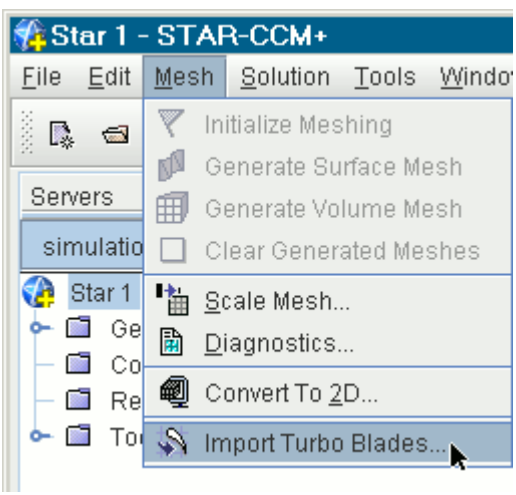
`Rotor37.estg` contains geometry data for 13 aerofoil sections between the Hub and Tip in a standard formatted file:

- `inlet.bndy` contains a spline definition for the inlet boundary.
- `exit.bndy` contains a spline definition for the exit boundary.
- `hub.bndy` contains a spline definition for the hub boundary.
- `tip.bndy` contains a spline definition for the tip (shroud) boundary.

Start up STAR-CCM+ in a manner that is appropriate to your working environment and select the **New Simulation** option from the menu bar.

STAR-CCM+ creates an additional tabbed window for the new simulation in the *Explorer* pane and give it the default name `Star 1`. Import the supplied data files into the simulation:

1. Select **Mesh** > **Import Turbo Blades** from the menu bar:



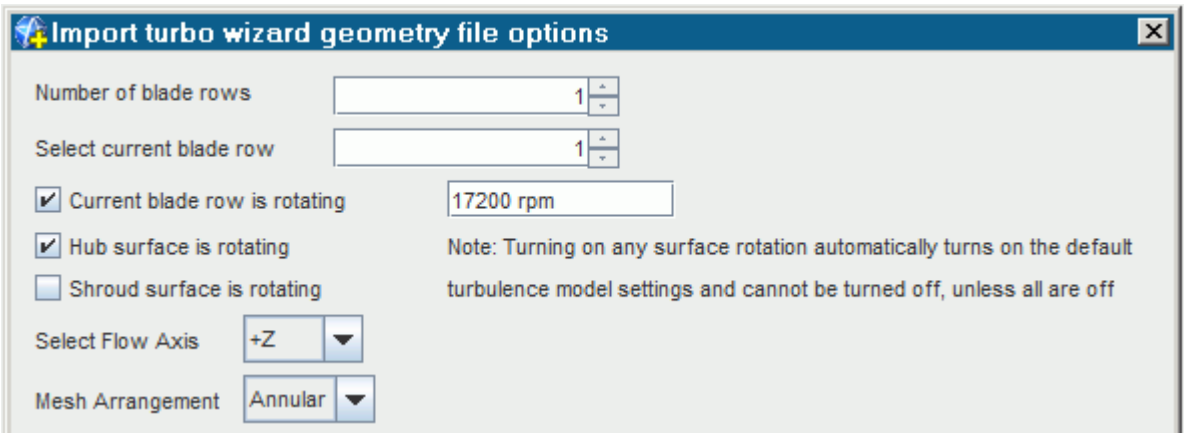
   The *Import turbo wizard geometry file options* dialog appears.

2. Make sure that **Number of blade rows** is set to `1`.
3. Make sure that **Select current blade row** is set to `1`.
4. Activate **Current blade row is rotating**.

   Additional options appear in the dialog.

5. Change the rotational speed to `17,200 rpm`.

6. Activate **Hub surface is rotating**.
7. Set *Select Flow Axis* to **+Z**.

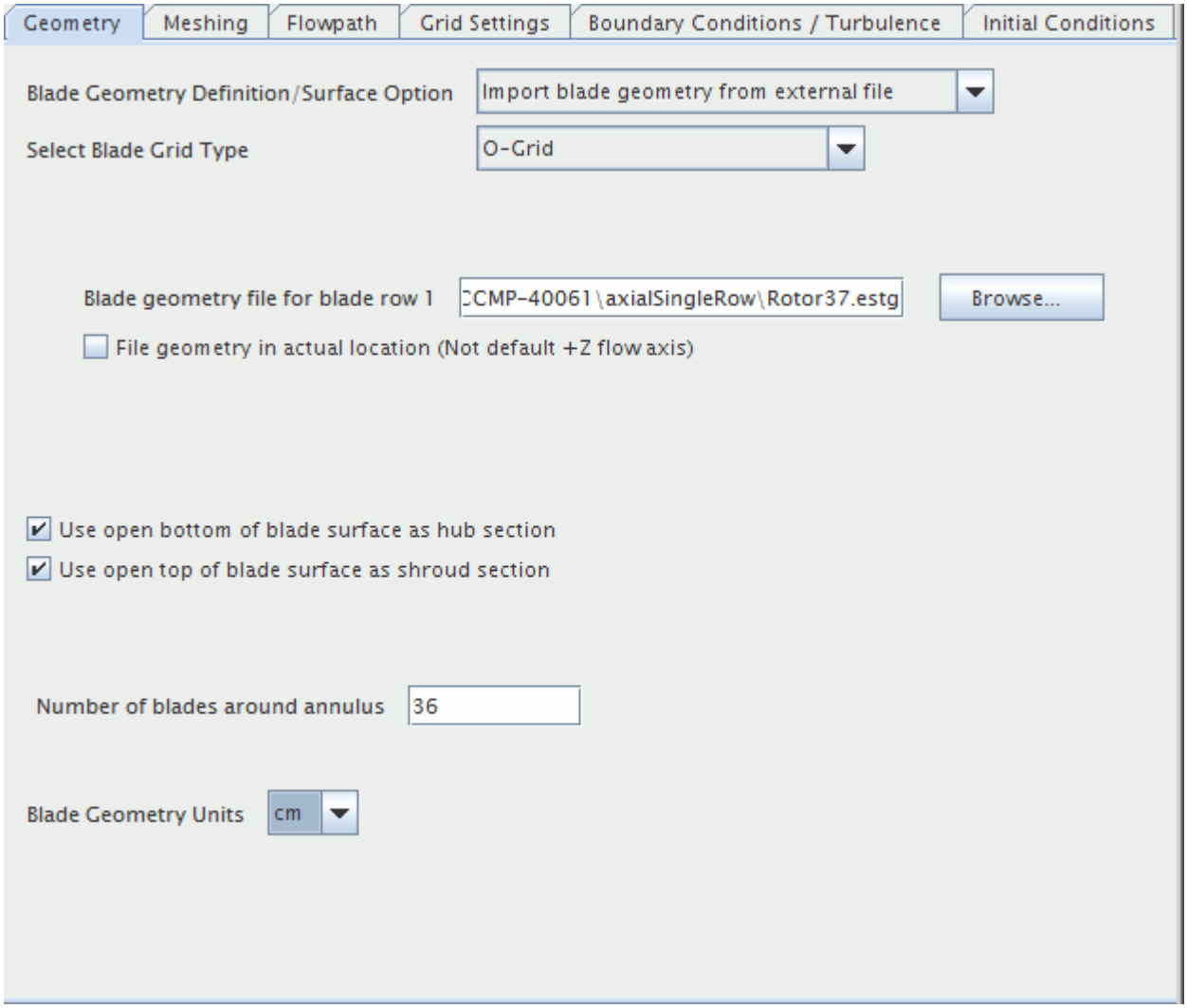


There are six tabs on the *Import turbo wizard geometry file options* dialog. The *Geometry*, *Meshing*, *Flowpath*, *Grid Settings*, *Boundary Conditions / Turbulence* and *Initial Conditions* tabs are used to specify the required files and parameters for the imported blade geometry.

Geometry Tab

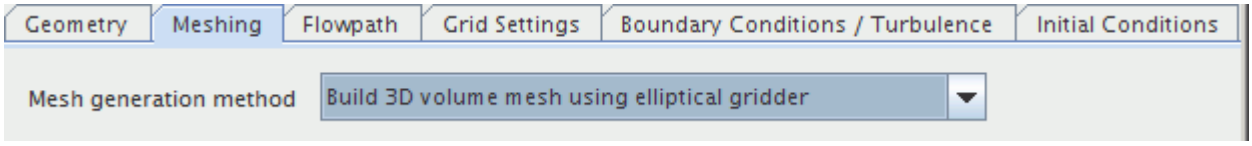
8. Set the following properties:

Property	Setting
Blade Geometry Definition/Surface Option	**Import blade geometry from external file**
Select Blade Grid Type	**O-Grid**
Blade geometry file for blade row 1	`axialSingleRow/Rotor37.estg`
Use open bottom of blade surface as hub section	Activated
Use open top of blade surface as shroud section	Activated
Number of blades around annulus	`36`
Blade Geometry Units	`cm` (centimeters)



Meshing Tab

9. Make sure that **Mesh generation method** is set to **Build 3D volume mesh using elliptical gridder**.

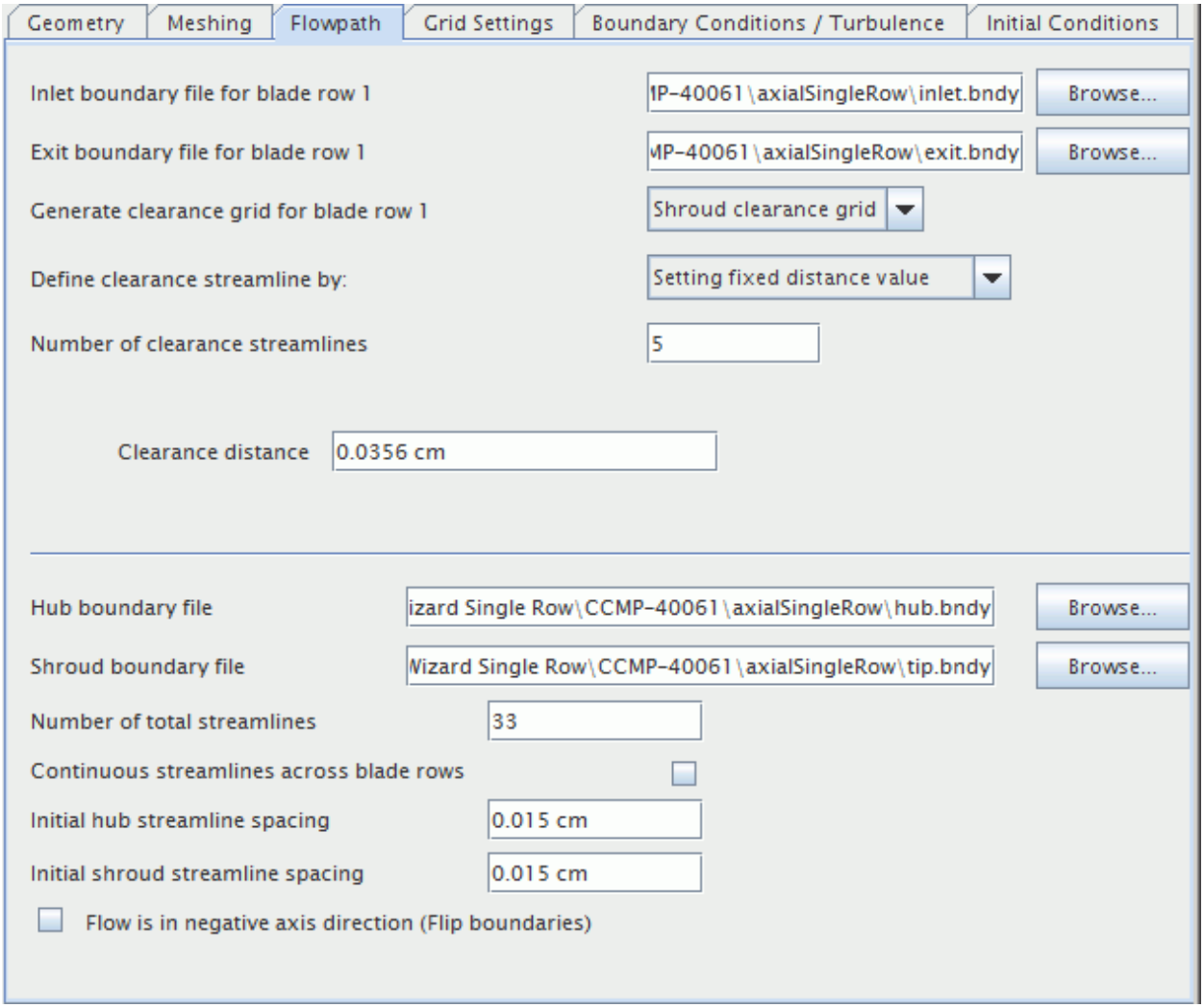


Flowpath Tab

10. Set the following properties:

Property	Setting	
Inlet boundary file for blade row 1	`inlet.bndy`	

Exit boundary file for blade row 1	`exit.bndy`	
Generate clearance grid for blade row 1	**Shroud Clearance grid**	This reduces y+ values near the blade tip, which improves convergence of the turbulence model.
Define clearance streamline by:	**Setting fixed distance value**	
Number of clearance streamlines	`5`	
Clearance distance	`0.0356 cm`	
Hub boundary file	`hub.bndy`	
Shroud boundary file	`tip.bndy`	
Number of total streamlines	`33`	This controls the number of streamlines from the inlet to outlet boundaries, on which the mesh layers between hub and shroud are constructed.
Initial hub streamline spacing	`0.015 cm`	This defines the thickness of the mesh layers next to the hub surfaces.
Initial shroud streamline spacing	`0.015 cm`	This defines the thickness of the mesh layers next to the shroud surfaces.

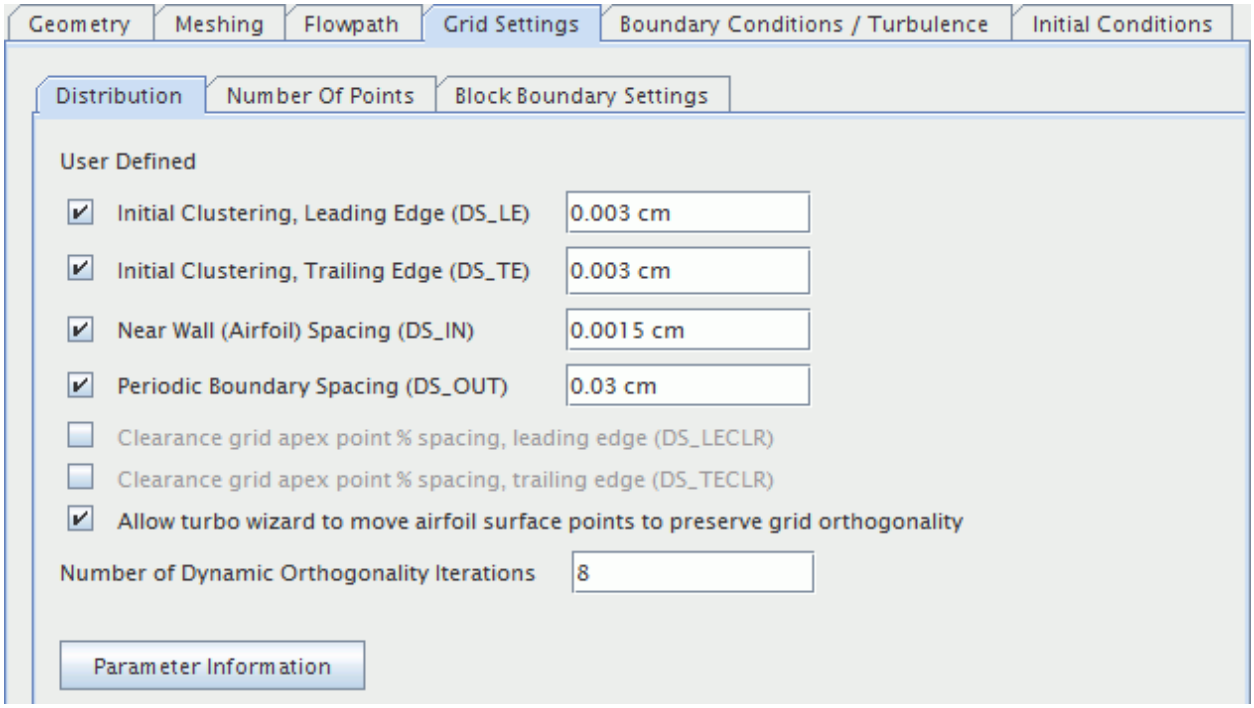


Grid Settings Tab

11. In the *Distribution* subtab, set the following properties:

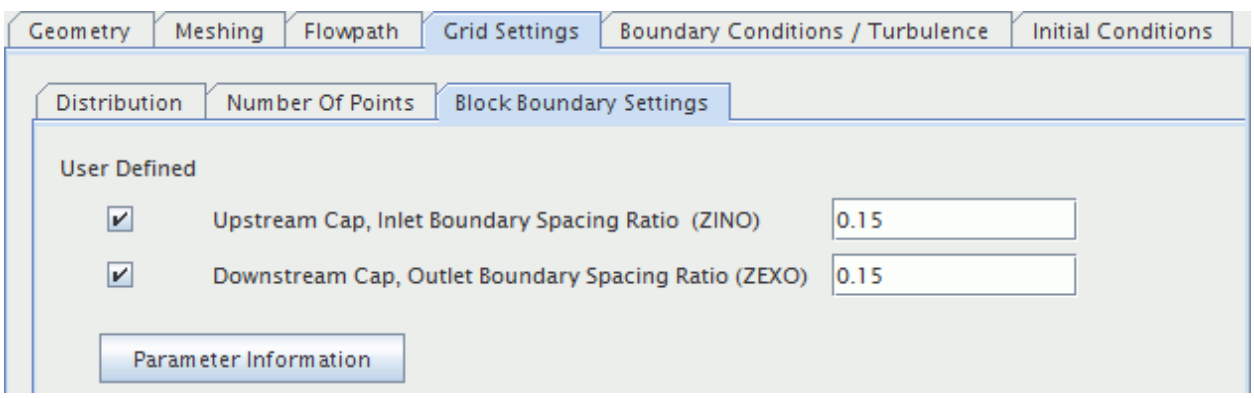
Property	Setting
Initial Clustering, Leading Edge (DS_LE)	Activated, `0.003 cm`
Initial Clustering, Trailing Edge (DS_TE)	Activated, `0.003 cm`
Near Wall (Airfoil) Spacing (DS_IN)	Activated, `0.0015 cm`
Periodic Boundary Spacing (DS_OUT)	Activated, `0.03 cm`
Allow turbo wizard to move airfoil surface points to preserve grid orthogonality	Activated

The number of iterations for this option can be left at the default value.



12. Accept the default settings on the *Number Of Points* subtab.
13. On the *Block Boundary Settings* subtab, set the following properties:

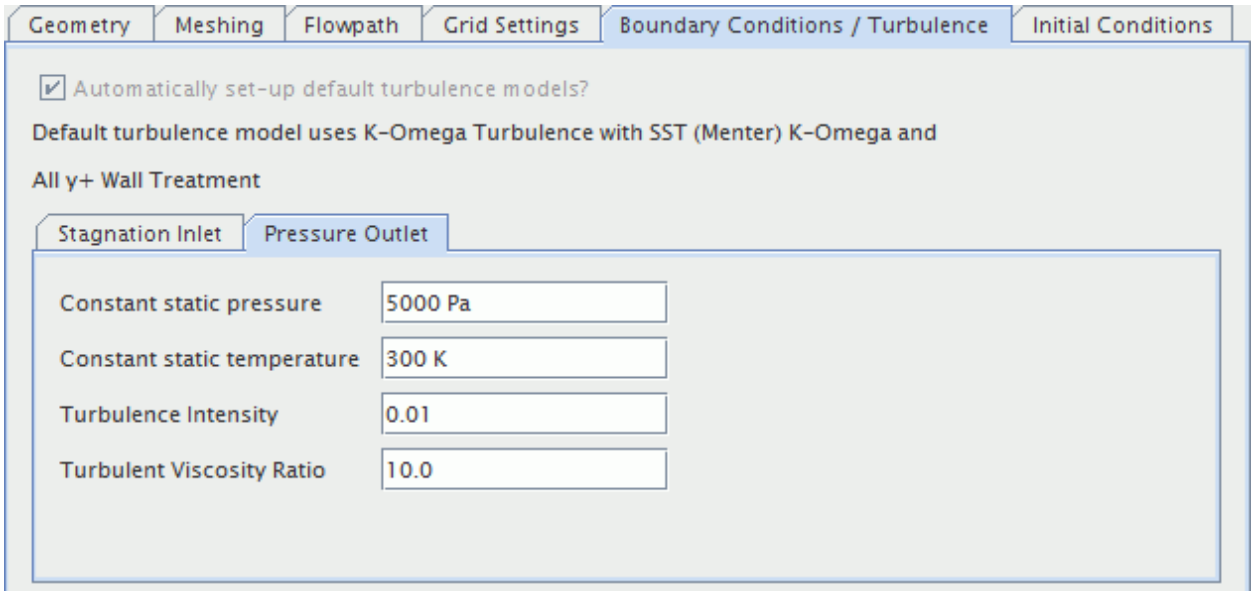
Property	Setting
Upstream Cap, Inlet Boundary Spacing Ratio (ZINO)	Activated, `0.15`
Downstream Cap, Outlet Boundary Spacing Ratio (ZEXO)	Activated, `0.15`



Boundary Conditions / Turbulence Tab

14. **Automatically set-up default turbulence models?** is activated and grayed out, since this setting is enabled automatically with the rotation of the hub and/or the shroud.
15. Accept the default settings on the *Stagnation Inlet* subtab.

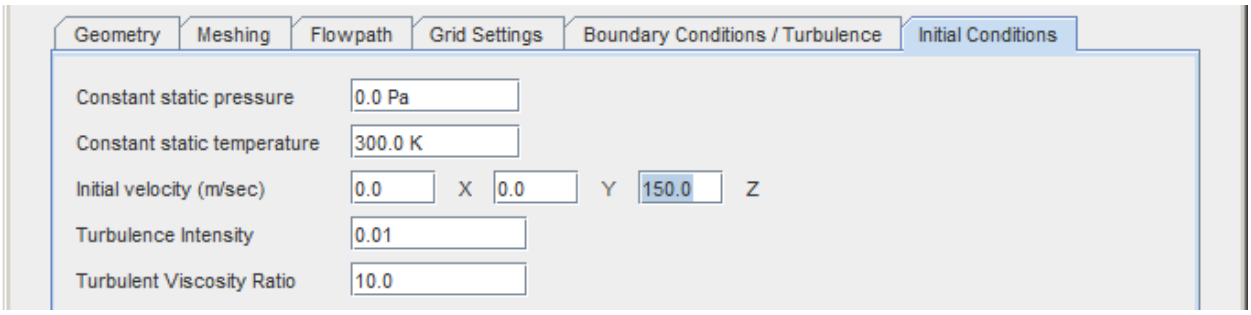
16. In the *Pressure Outlet* subtab, change the **Constant static pressure** value to `5000 Pa`, leaving the other parameters at their defaults.



Initial Conditions Tab

17. Change the **Initial Velocity Z** value to `150 m/sec`, leaving the other parameters at their defaults.

    This sets a non-zero velocity in the axial direction.



18. Click **Save Turbo Wizard Settings** and, in the *Save Turbo Wizard Data* dialog, navigate to the `axialSingleRow` directory created previously.
19. **Save the data as** `turboWizardData`.

The saved turbo wizard data can then be loaded and modified for the same case or used to set up other cases.

20. To generate the volume mesh, click **OK**.

STAR-CCM+ starts creating the grid plane for all streamlines. Once the mesh is generated, STAR-CCM+ converts it into finite volume representation. See the *Output* window for more details. At the end of the import process, STAR-CCM+ also creates an initial (default) simulation tree in the *Star 1* window.

To create a file for the simulation, and give it a proper name:

21. Select **File** > **Save** from the menu bar.
22. Type `singleRow` in the **File Name** text box then click **Save**.

    The title of the simulation window also changes to *singleRow*.

Checking the Models

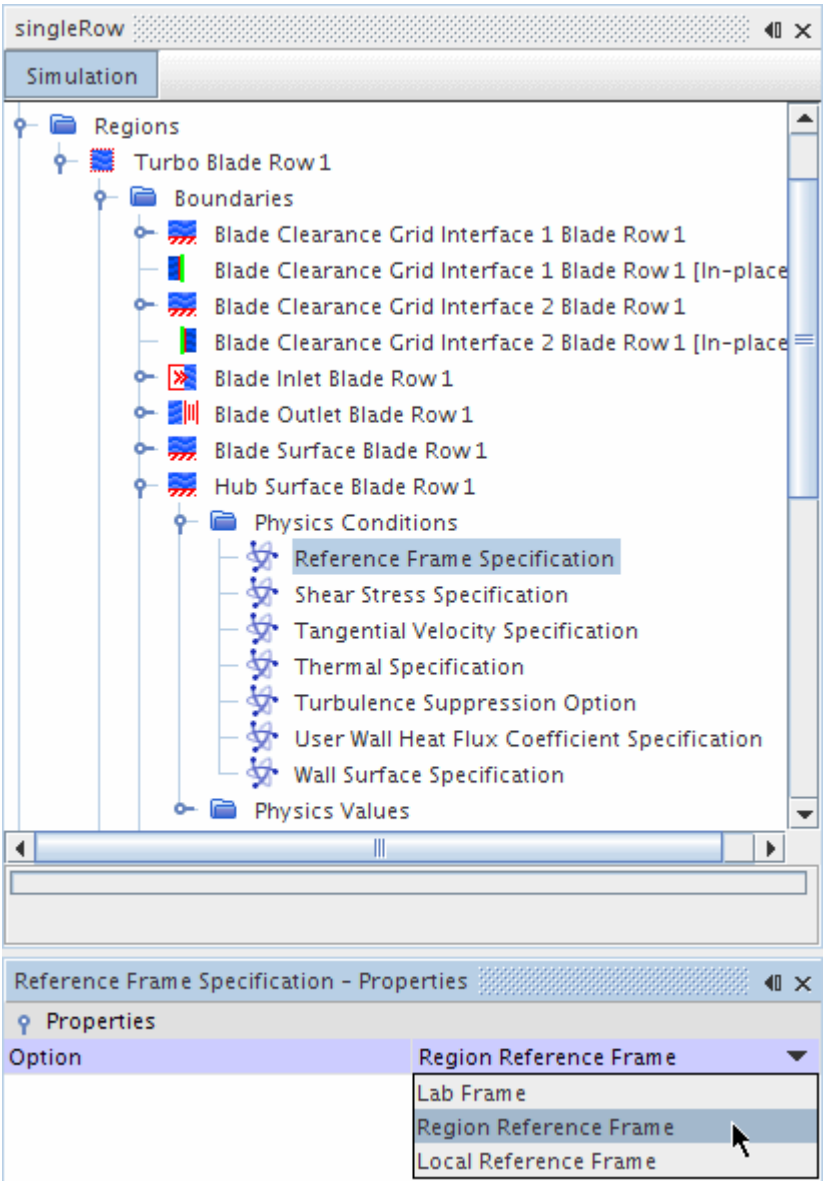
By default the Turbo Wizard sets up a steady-state simulation in which:

- A three-dimensional grid is employed.
- The fluid is a compressible gas (air) that obeys the ideal gas equation of state.
- The flow is turbulent and non-isothermal.
- The K-Omega model is used for representing turbulence effects.
- The problem is solved using the coupled flow solver.

These settings can be checked by expanding the Models node of the Physics 1 continuum.

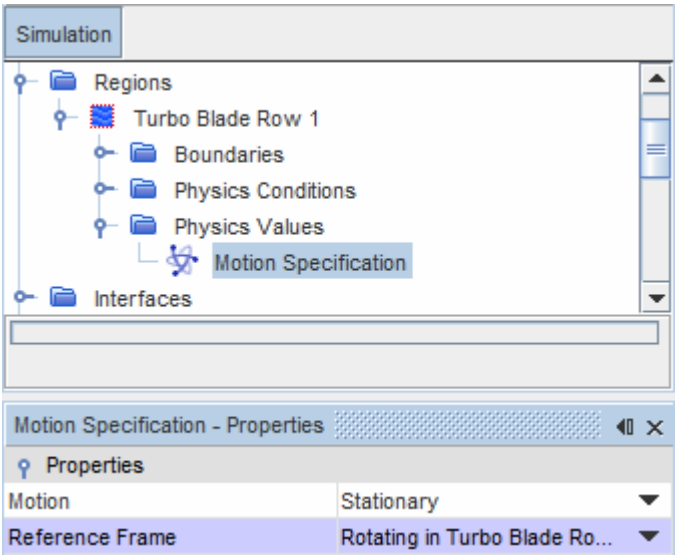
1. Expand the **Regions** > **Turbo Blade Row 1** > **Boundaries** > **Hub Surface Blade Row 1** > **Physics Conditions** node and select **Reference Frame Specification**.

   The *Option* property is set to **Region Reference Frame**.

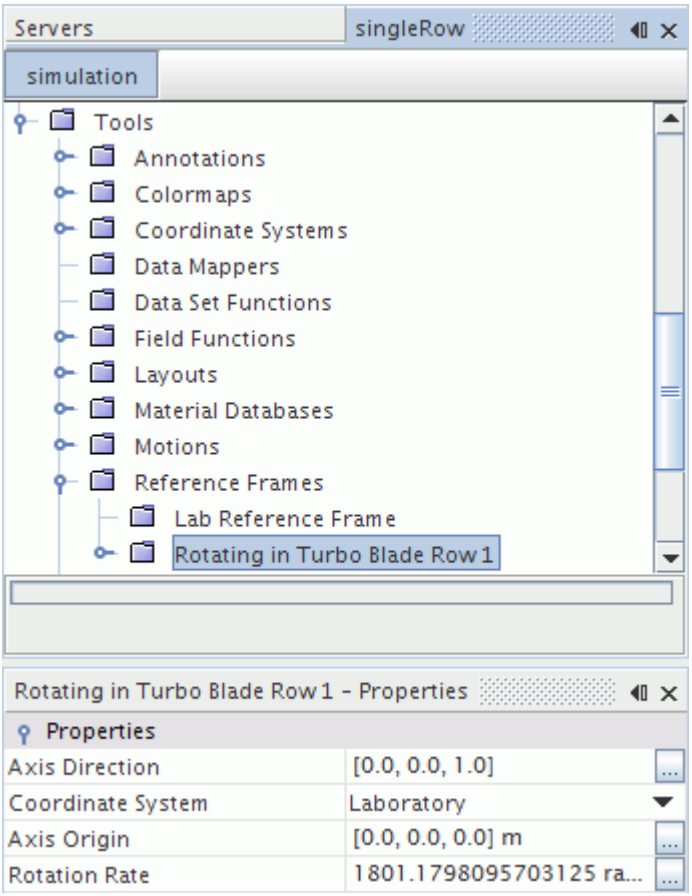


2. Expand the **Regions** > **Turbo Blade Row 1** > **Physics Values** node and select **Motion Specification**.

   The *Reference Frame* property is set to **Rotating in Turbo Blade Row 1**.



3. The properties of this reference frame can be viewed by expanding the **Tools** > **Reference Frames** node and selecting **Rotating in Turbo Blade Row 1**.



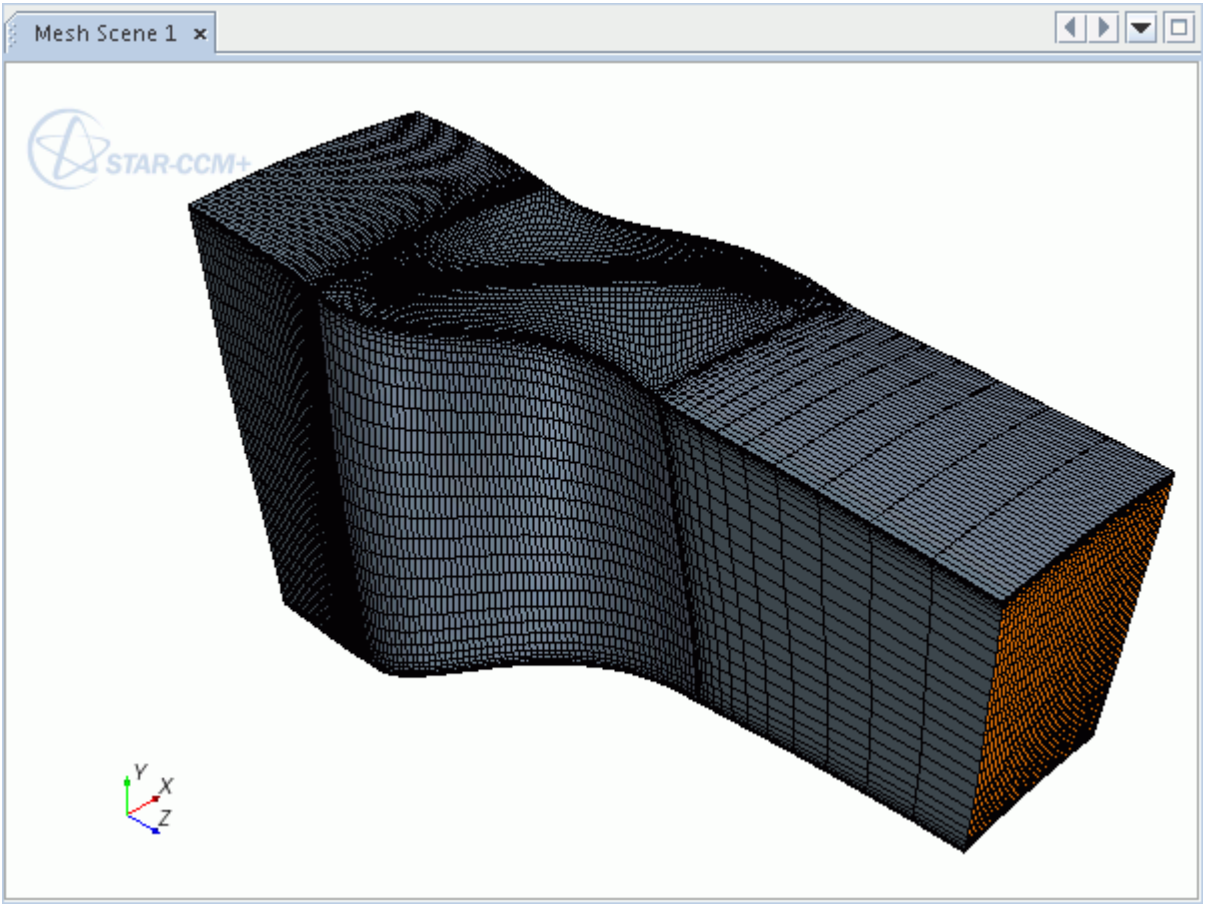
Visualizing the Mesh

To inspect the mesh:

1. Right-click the **Scenes** node and select **New Scene** > **Mesh**.

The *Graphics* window also shows the mesh. Examine the mesh more closely:

2. Produce a *Graphics* window display such as the one shown in the following screenshot.



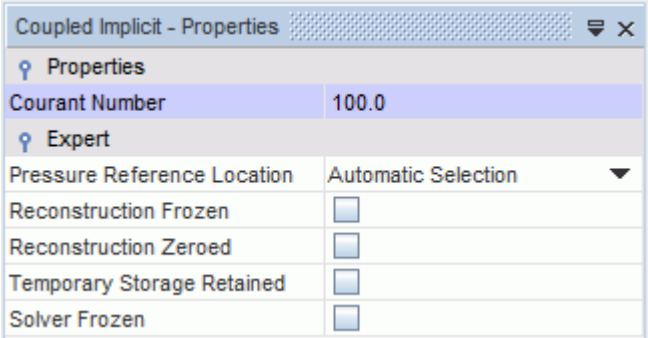
3. Verify that no malformed or irregular cells are present on the mesh surface.

Setting Up Solver and Stopping Criteria

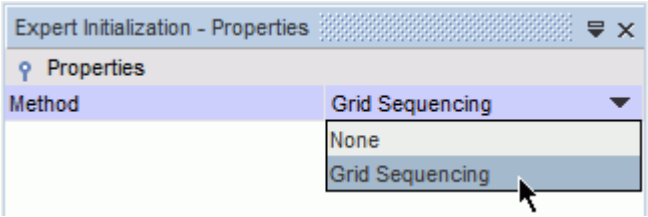
By default, the solver parameters are set to the Coupled Implicit with a Courant number of 5.

Increase this number to reduce the number of iterations the solver needs to converge. Also uses Grid Sequencing and Expert Driver to make the solver more robust and control the stability of the solution.

1. Select the **Solvers** > **Coupled Implicit** node and set **Courant Number** to `100.0`.

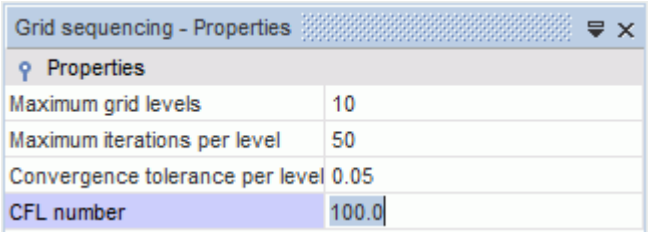


2. Open the **Solvers** > **Coupled Implicit** > **Expert Initialization** node and set **Method** to **Grid Sequencing**.



3. Open the **Expert Initialization** > **Grid sequencing** node and set **CFL number** to `100.0`.

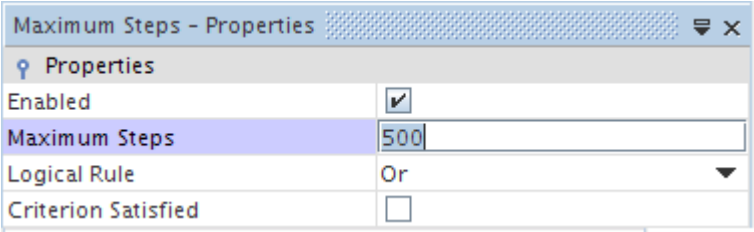
   This CFL number is specific to the grid sequencing setup and does not need to be equal to the Courant number defined earlier.



4. Open the **Solvers** > **Coupled Implicit** > **Solution Driver** node and set **Driver** to **Expert Driver**. Use the Expert Driver's default settings.

Before running the analysis, reduce the maximum number of steps to account for increasing the Courant number and using Grid Sequencing and Expert Driver to control the solver's convergence.

5. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to `500`.

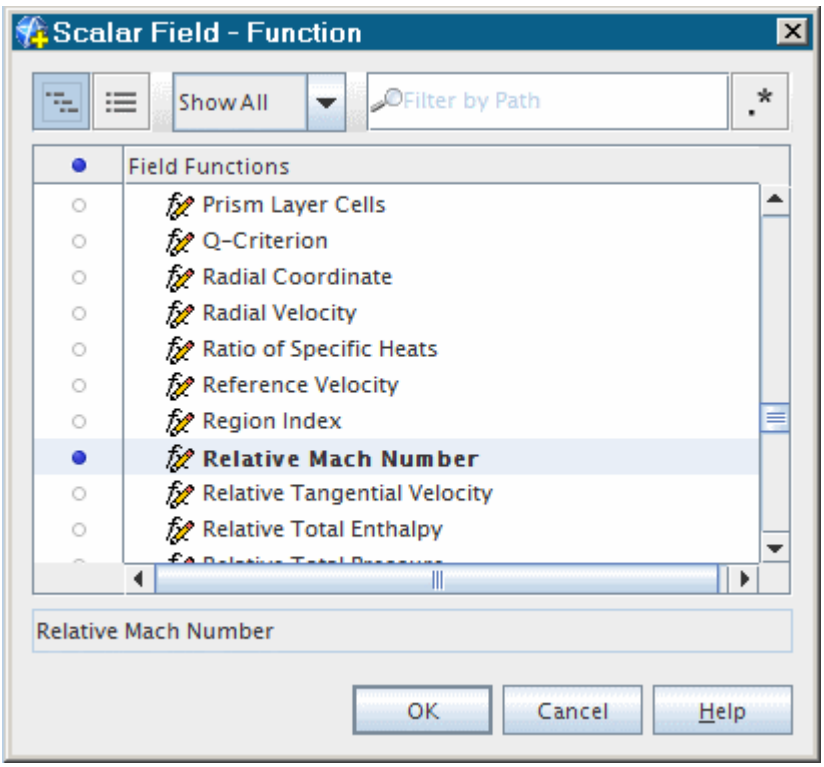


Visualizing the Solution

In this section you define three isosurfaces including blade-to-blade and meridional views, to create a Relative Mach number scene.

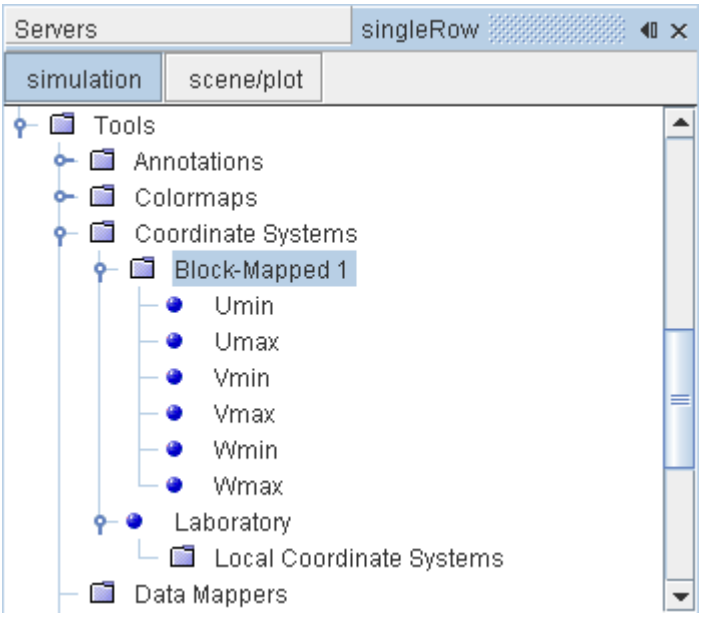
1. Right-click the **Scenes** node and select **New Scene** > **Scalar**.

   A new node that is called **Scalar Scene 1** is created in the simulation tree.
2. Select the **Scenes** > **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Scalar Field** node and set *Function* to **Relative Mach Number**.



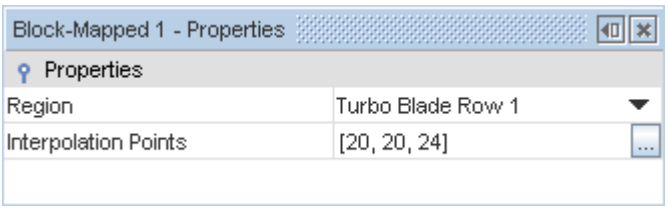
3. Open the **Tools** node.
4. Right-click the **Coordinate Systems** node and select **New Block-Mapped**.

   A new node, **Block-Mapped 1**, appears in the simulation tree.



5. Select the **Block-Mapped 1** node and set the *Interpolation Points* to `[20, 20, 24]`

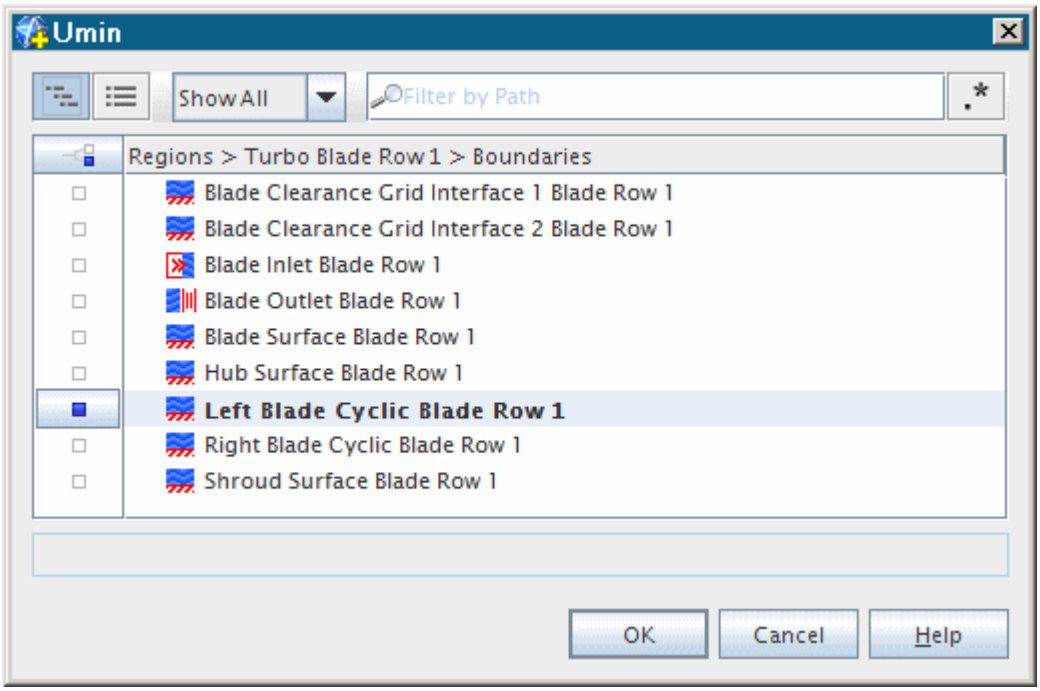
   This setting gives greater resolution in the meridional direction.



6. Select the **Umin** node and click in the right half of the *Boundaries* property.

   The *Select Objects* dialog opens.

7. Activate **Regions** > **Turbo Blade Row 1** > **Left Blade Cyclic Blade Row 1** and click **OK**.



8. Repeat the above steps for the following nodes:

Node	Property	Setting
Umax	*Boundaries*	**Right Blade Cyclic Blade Row 1**
Vmin	*Boundaries*	**Hub Surface Blade Row 1**
Vmax	*Boundaries*	**Shroud Surface Blade Row 1**
Wmin	*Boundaries*	**Blade Inlet Blade Row 1**
Wmax	*Boundaries*	**Blade Outlet Blade Row 1**

Create a derived part:

9. Right-click the **Derived Parts** node and select option **New Part** > **Isosurface...**

   An interactive in-place dialog (*Create Isosurface*) appears on the left of the *Graphics* window to help you define the surface.

   In this dialog:

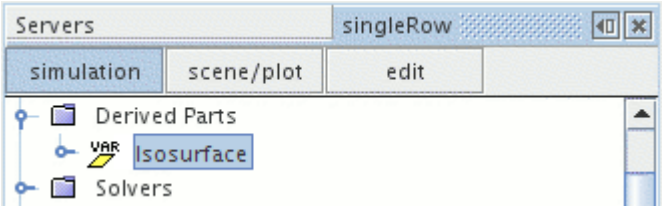
   a) In the **Scalar** box, select **Block-Mapped 1: Wmin to Wmax**.

   b) Change the isovalue to `0.01`.

   c) From the *Display* box, select **Existing Displayer** and select `Scalar 1`.

   d) Click **Create**, but not **Close**, as you wish to create further isosurfaces.

10. Click the *simulation* tab at the top of the simulation explorer. A new node, **isosurface**, appears under the **Derived Parts** node in the simulation tree.



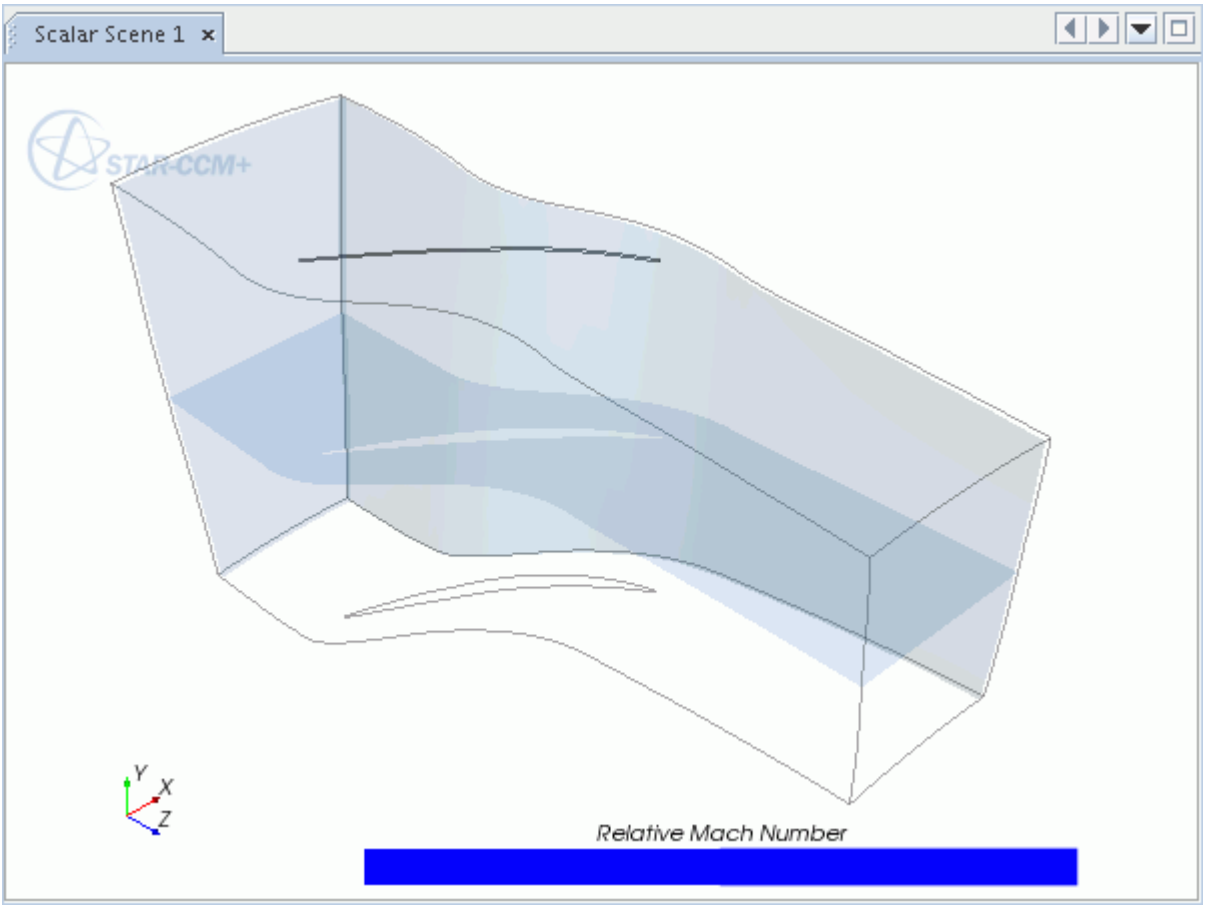
11. Rename the **isosurface** node to `AxialView`.
12. Click the *edit* tab to switch panes, and in the *Create Isosurface* dialog:
    a) Select **Block-Mapped 1: Vmin to Vmax** in the **Scalar** box.
    b) Change the isovalue to `0.5`.
    c) Click **Create** and switch back to the simulation tree by clicking the *simulation* tab at the top.

In the simulation tree:

13. Rename the **isosurface** node to `Blade2BladeView`.
14. Click the *edit* tab to switch panes, and in the *Create Isosurface* dialog:
    a) Select **Block-Mapped 1: Umin to Umax** in the **Scalar** box.
    b) Change the isovalue to `0.025`.
    c) Click **Create and then Close**.

In the simulation tree:

15. Rename the **isosurface** node to `MeridionalView`.
16. Click **(Make Scene Transparent)** to display the above-defined isosurfaces as shown in the following screenshot.



17. Click **(Restore Scene Transparency)** to set the transparency of the scene to its original setting.
18. Select the **Scenes** > **Scalar Scene 1** > **Displayers** > **Scalar 1**node and set *Contour Style* to `Smooth Filled`.
19. Save the simulation.

The pre-processing task is now complete. The next step is to run the analysis, as described in Running the Simulation.

Running the Simulation

Run the simulation.

1. Click **(Run)** to start the solver.

The analysis starts and the results, in the form of a graph of residuals vs. iteration number, is displayed automatically in the *Graphics* window.

The solution stops after 500 steps.

2. Save the simulation.

Analyzing Results

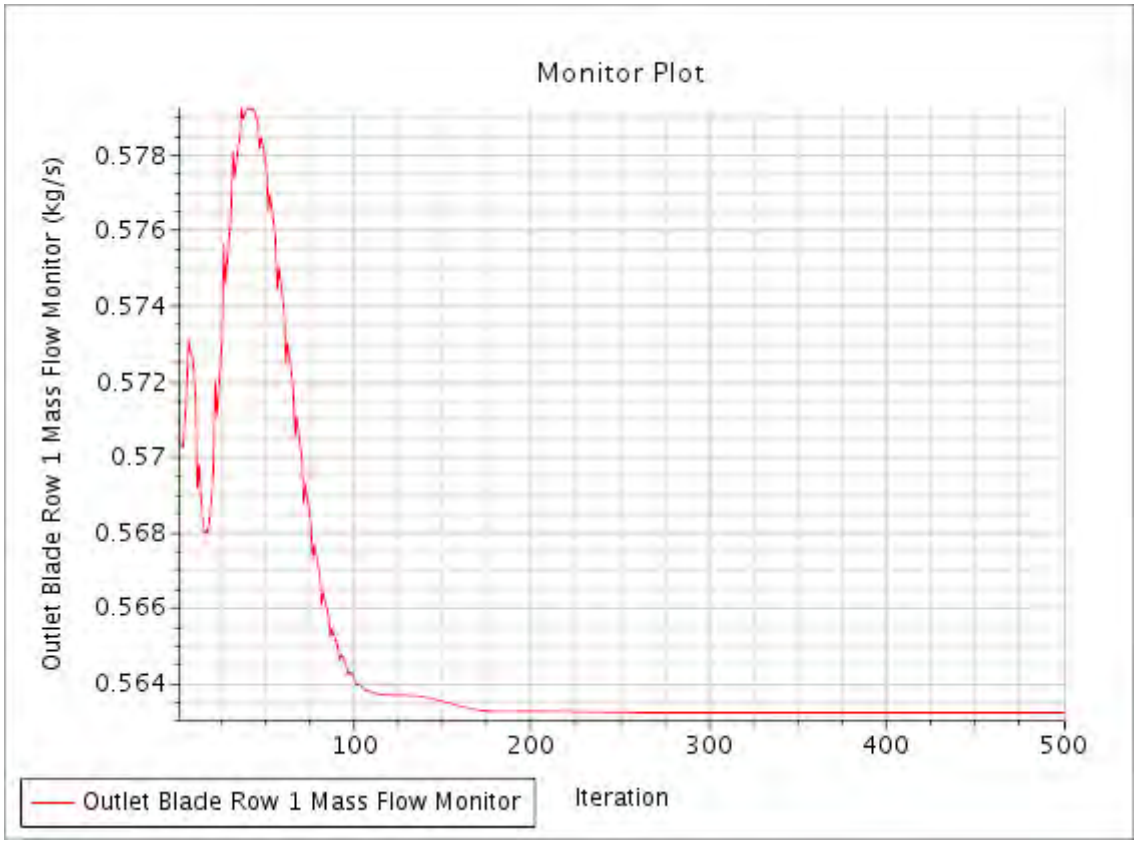
This section of the tutorial covers post-processing.

To assess the degree of convergence, it is helpful to look at the inlet and outlet mass flow plots to make sure that the inlet mass flow is equal to the outlet mass flow. For an inlet area, the normal vector introduces a negative sign.

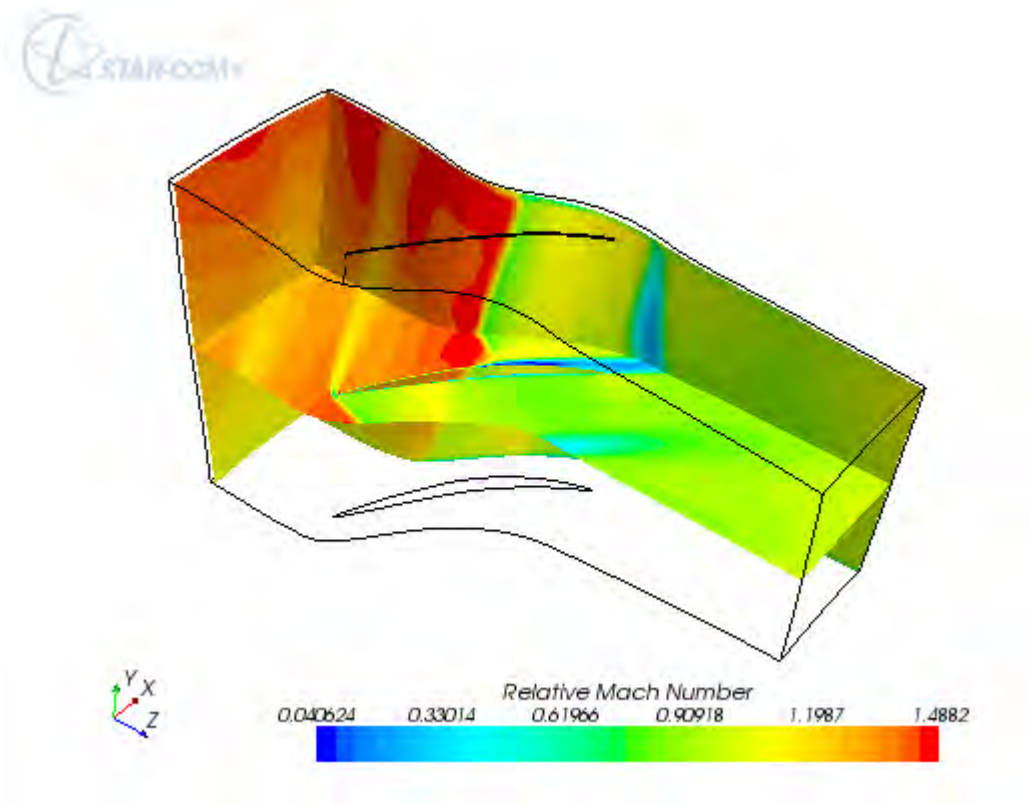
1. Open the **Plots** node.
2. Right-click **Plots** > **Inlet Blade Row 1 Mass Flow Plot** node and select **Open**.

   This command opens the *Inlet Blade Row 1 Mass Flow Plot* scene in the *Graphics* window.

3. Similarly, you can view the *Outlet Blade Row 1 Mass Flow Plot*.



The scalar scene is shown below.



4. Save the simulation.

Summary

This STAR-CCM+ tutorial introduced the following features:

- Starting the code and creating a simulation
- Importing the geometry and flowpath boundary data for an axial compressor blade
- Generating a volume mesh
- Defining the simulation models
- Visualizing the mesh
- Setting up solver and stopping criteria
- Setting up a scalar scene with data from derived parts
- Running the solver for a set number of iterations
- Analyzing the results using the built-in plotting and visualization facilities

Single Row Tutorial Bibliography

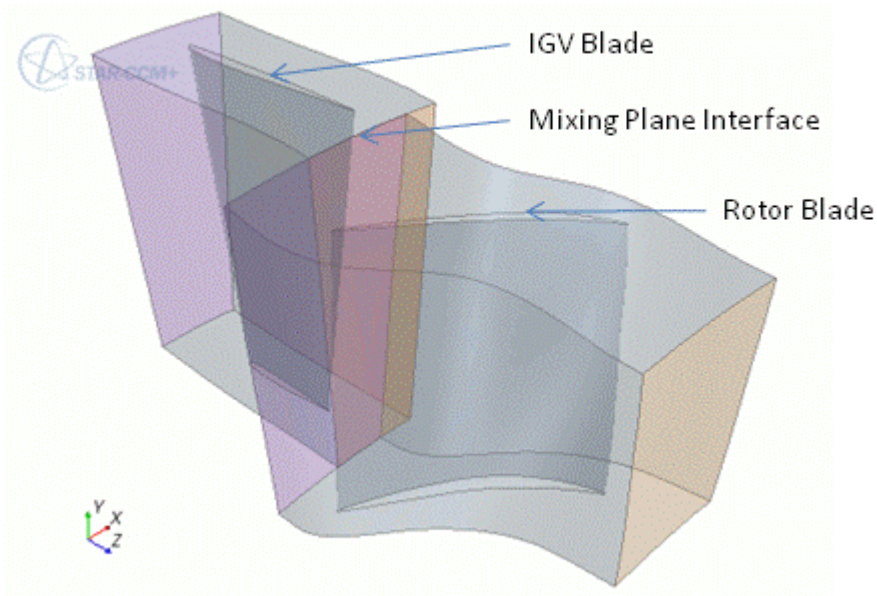
[788] *Reid, L and Moore, R. D., 1978. "Performance of Single-Stage Axial-Flow Transonic Compressor With Rotor and Stator Aspect Ratios of 1.19 and 1.26, Respectively, and With Design Pressure Ratio of 1.82", NASA TP-1338.*

Turbo Wizard: Multiple Rows

Having successfully created a mesh for a single row axial blade and run it in the previous section, add one additional row to the problem geometry in this tutorial.

The two blades that are used are the Rotor37 axial compressor blade, with an additional tip clearance, and an Inlet Guide Vane (IGV) positioned upstream of it. You define blade row 1 to be the IGV and blade row 2 to be the compressor rotor blade. The cross-sectional cuts for Rotor37 and IGV are stored in `Rotor37.estg` and `IGV.estg` files, respectively. These files are imported into STAR-CCM+.

The problem geometry is shown below:



Contents:

Importing the Turbo Blades
Checking the Models
Visualizing the Mesh
Setting Up Solver and Stopping Criteria
Visualizing the Solution
Running the Simulation
Analyzing Mass Flow Reports
Summary

Importing the Turbo Blades

Create a subdirectory for the tutorial called `axialMultipleRow`.

1. Copy the following files from the `doc/startutorialsdata/compressibleFlow/data/MultipleRow` subdirectory of your STAR-CCM+ installation into this directory:

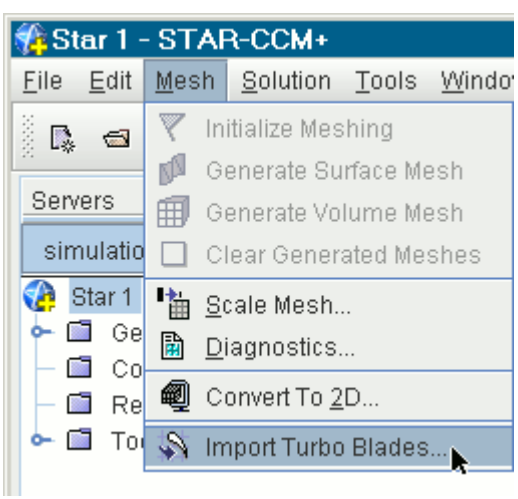
File	Description
`Rotor37.estg`	Contains the geometry data of the Rotor for 13 airfoil sections between the Hub and Tip in a standard formatted file.
`IGV.estg`	Contains the geometry data of the IGV for two airfoil sections between the Hub and Tip in a standard formatted file.
`inlet1.bndy`	Contains a spline definition for the inlet boundary for the IGV.
`inlet2.bndy`	Contains a spline definition for the inlet boundary for the Rotor.
`exit1.bndy`	Contains a spline definition for the exit boundary for the IGV.
`exit2.bndy`	Contains a spline definition for the exit boundary for the Rotor.
`hub.bndy`	Contains a spline definition for the hub boundary.
`tip.bndy`	Contains a spline definition for the shroud (tip) boundary.

2. Start up STAR-CCM+ in a manner that is appropriate to your working environment and select the **New Simulation** option from the menu bar.

   STAR-CCM+ creates an additional tabbed window for the new simulation in the *Explorer* pane and gives it the default name Star 1.

3. Import the supplied data files into the simulation:

   a) Select **Mesh** > **Import Turbo Blades** from the menu bar:



   This selection opens the *Import turbo wizard geometry file options* dialog.

   b) Change the **Number of blade rows** to `2`.

The following tabs are used here to specify the files and parameters for creating a volume mesh for the imported blade geometries:

- *Geometry*
- *Meshing*
- *Flowpath*
- *Grid Settings*
- *Boundary Conditions / Turbulence*
- *Initial Conditions*

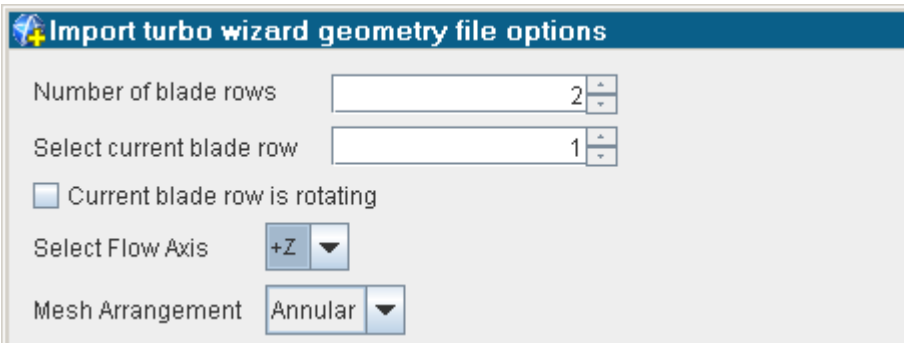
Contents:

Blade Row 1 (IGV)
Blade Row 2 (Rotor)

Blade Row 1 (IGV)

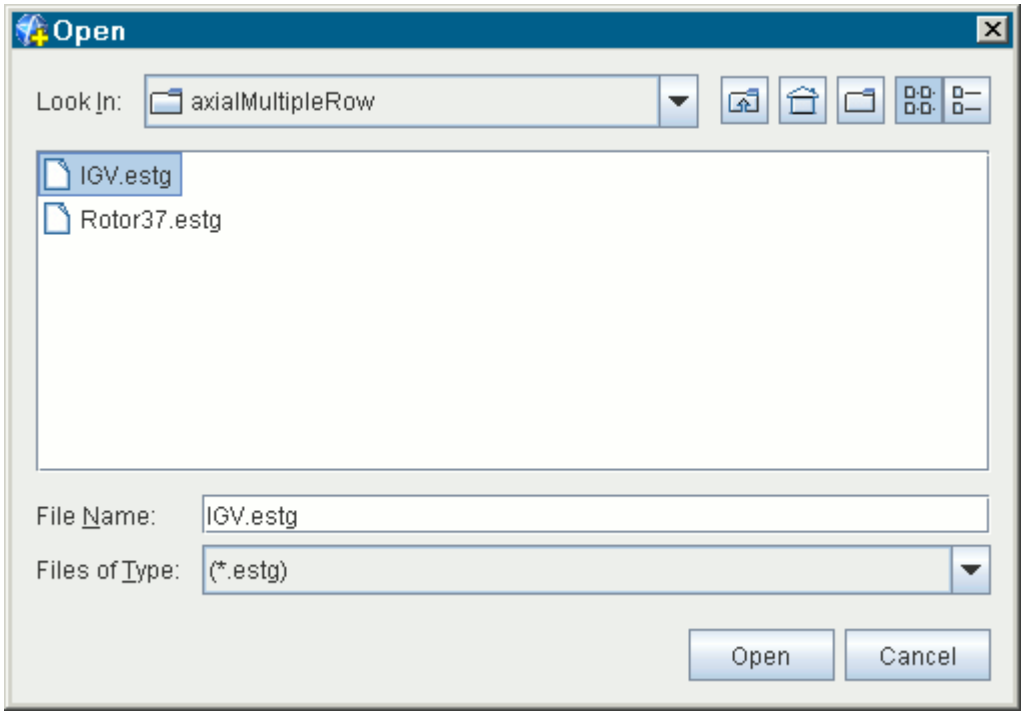
Set specifications for the first blade row.

1. Set the **Select current blade row** to `1`.
2. Make sure **Current blade row is rotating** is de-selected, as the blade row is static.
3. Choose **+Z** from the *Select Flow Axis* drop-down menu.



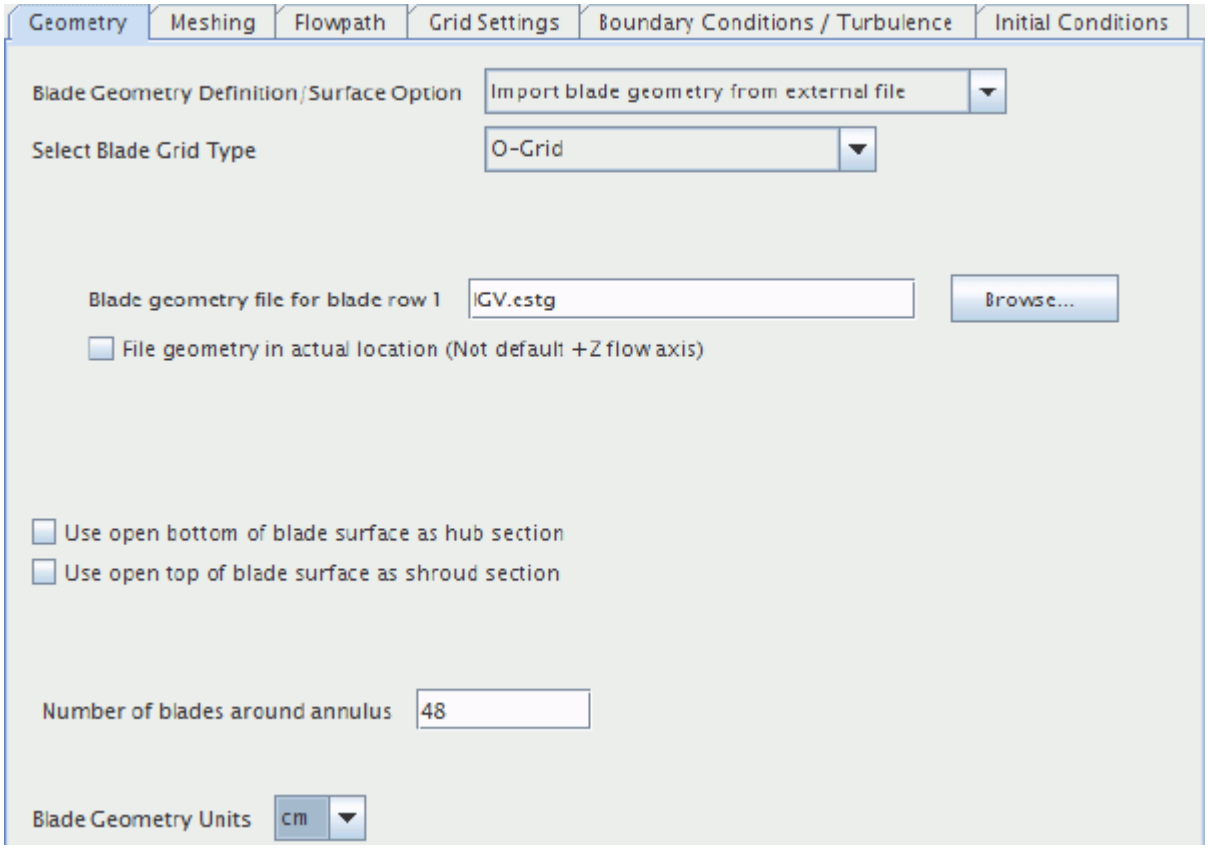
Geometry Tab

4. Click **Browse** and navigate to the `axialMultipleRow` directory created previously.



5. Select `IGV.estg` and set the following properties:

Property	Setting
Select Blade Grid Type	**O-Grid**
Blade geometry file for blade row 1	`IGV.estg`
Use open bottom of blade surface as hub section	Deactivated
Use open top of the blade surface as shroud section	Deactivated
Number of blades around annulus	`48`
Blade Geometry Units	**cm**



Meshing Tab

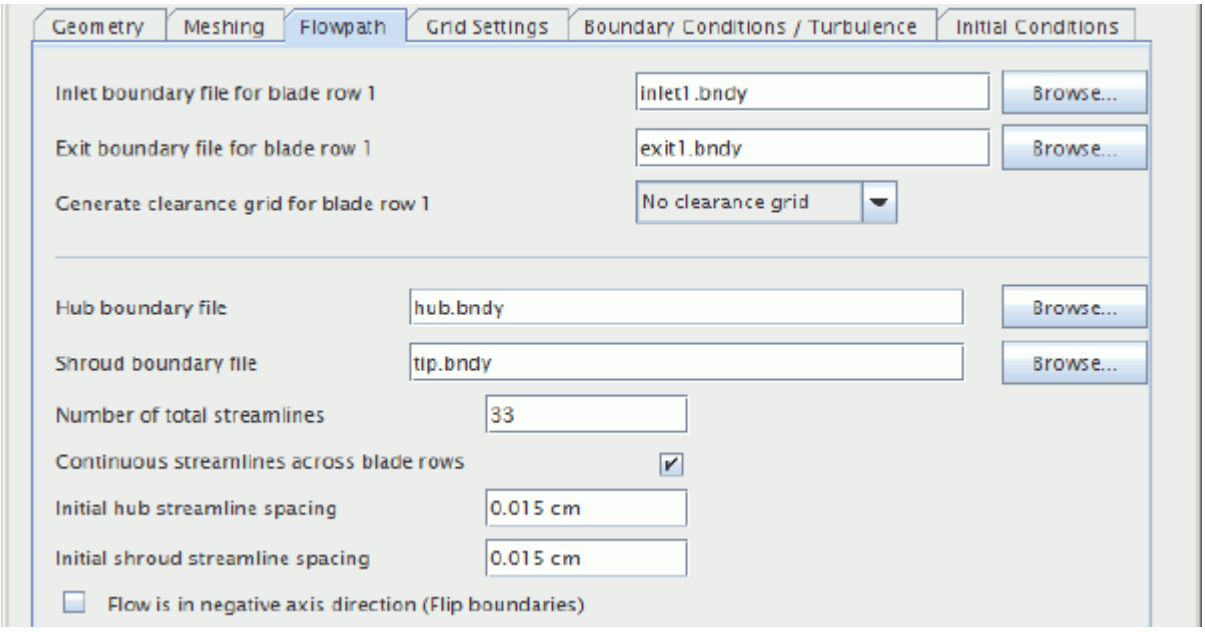
6. Set **Mesh generation method** to **Build 3d volume mesh using elliptical gridder**.

Flowpath Tab

7. Select the following files and set the following properties:

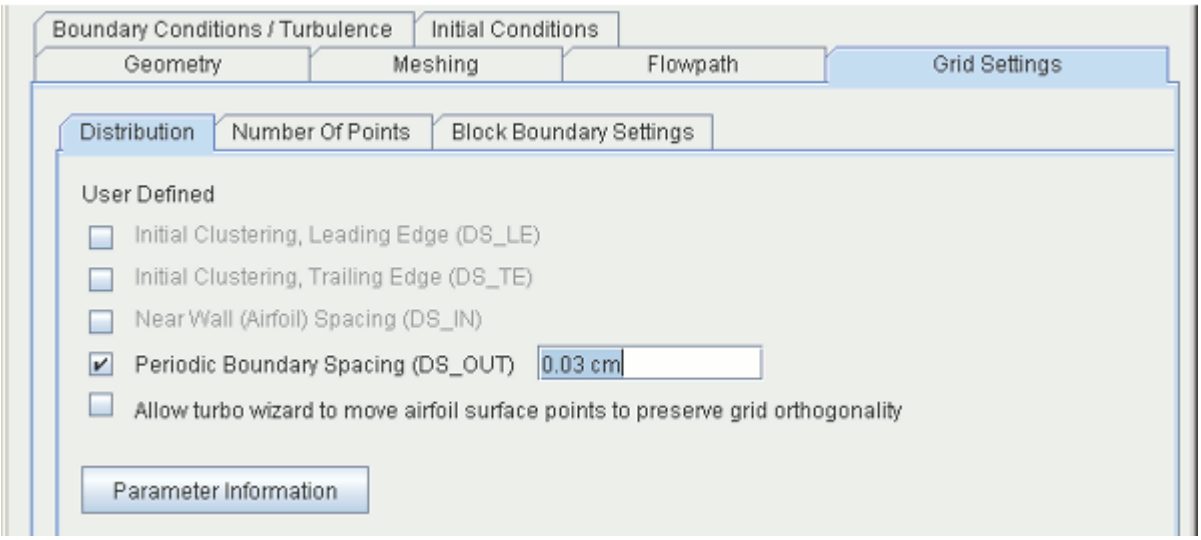
Property	Setting	
Inlet boundary file for blade row 1	`inlet1.bndy`	
Exit boundary file for blade row 1	`exit1.bndy`	
Generate clearance grid for blade row 1	**No clearance grid**	The IGV is static, so no clearance is required.
Hub boundary file	`hub.bndy`	
Shroud boundary file	`tip.bndy`	
Number of total streamlines	`33`	This controls the number of streamlines from the inlet to the outlet boundaries of the Rotor, on which the mesh layers between hub and shroud are constructed.

Continuous streamlines across blade rows	Activated	
Initial hub streamline spacing	`0.015 cm`	
Initial shroud streamline spacing	`0.015 cm`	



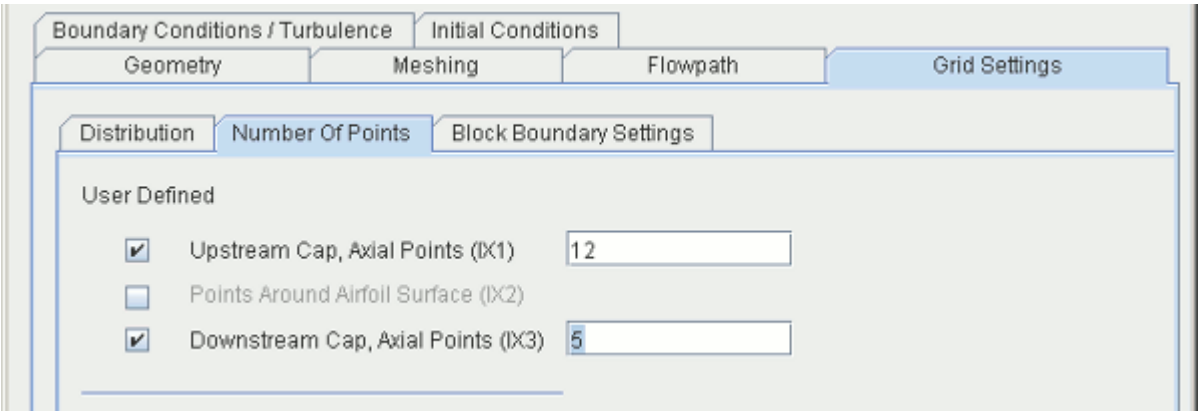
Grid Settings Tab

8. Select the *Distribution* subtab.
9. Activate **Periodic Boundary Spacing** and enter `0.03 cm` in the text box.



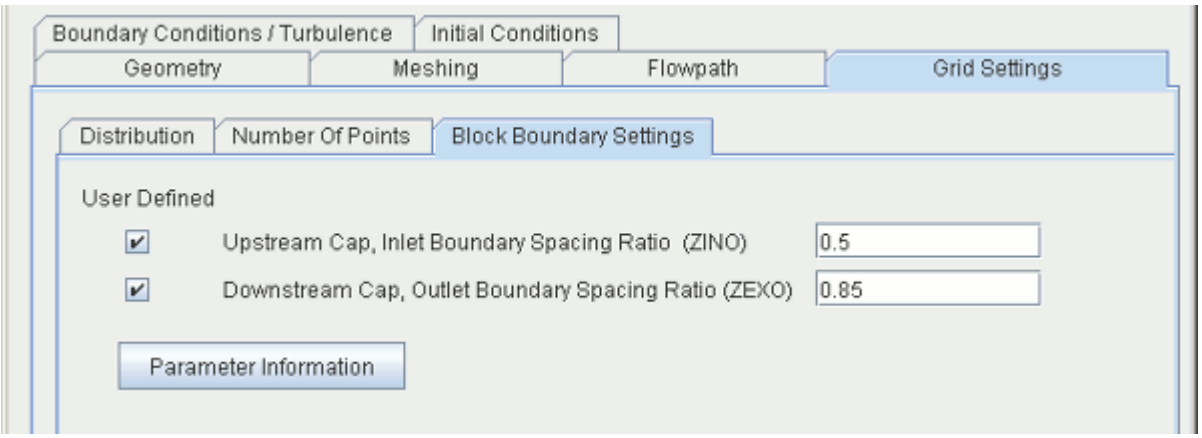
10. Select the *Number Of Points* subtab and enter the following properties:

Property	Setting
Upstream Cap, Axial Points	Activated, 12
Downstream Cap, Axial Points	Activated, 5



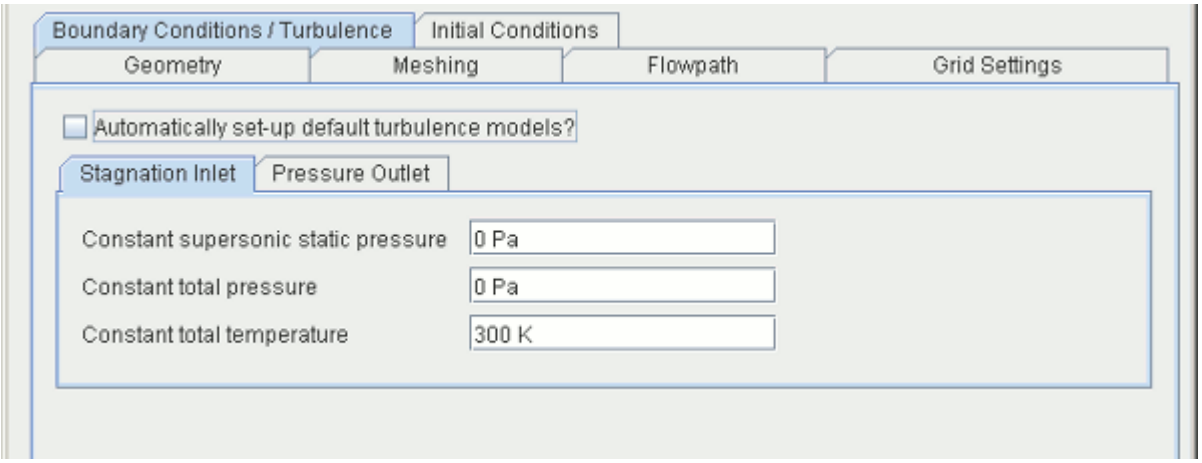
11. Select the *Block Boundary Settings* subtab and enter the following properties:

Property	Setting
Upstream Cap, Inlet Boundary Spacing Ratio	Activated, 0.5
Downstream Cap, Outlet Boundary Spacing Ratio	Activated, 0.85



Boundary Conditions / Turbulence Tab

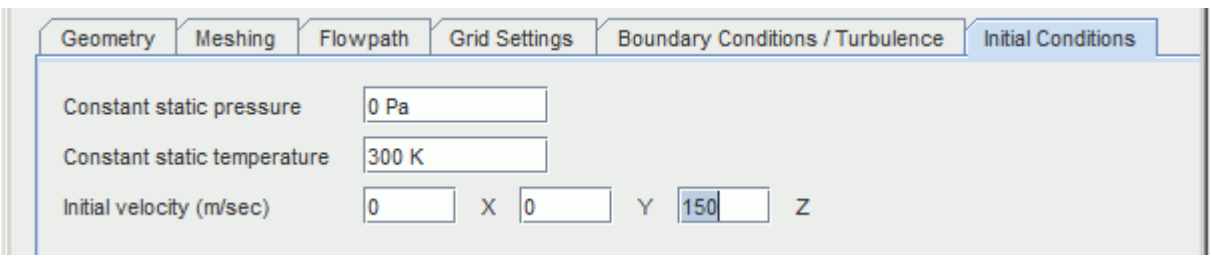
12. Accept the default settings.



Initial Conditions Tab

13. Change the **Initial Velocity Z** value to `150 m/sec`, leaving the other parameters at their defaults.

    This sets a non-zero velocity in the axial direction.

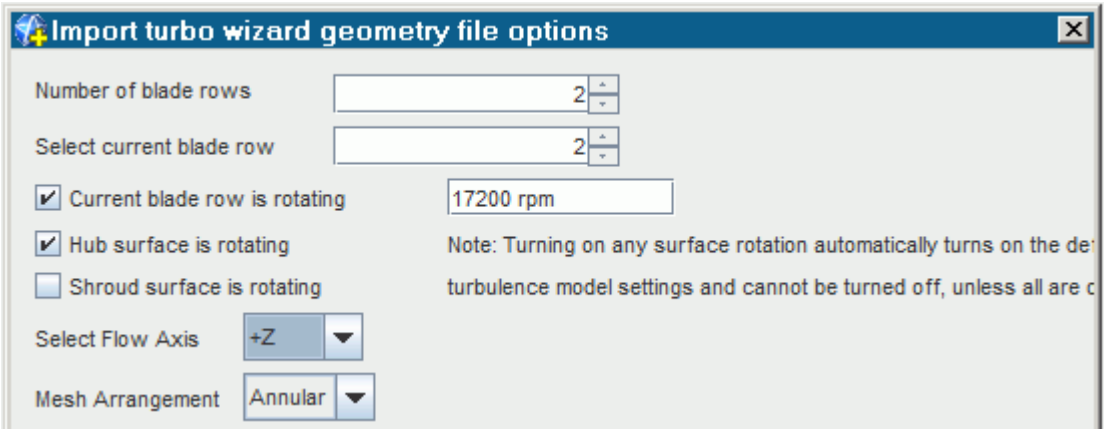


Blade Row 2 (Rotor)

Set specifications for the second blade row.

1. Set the following properties:

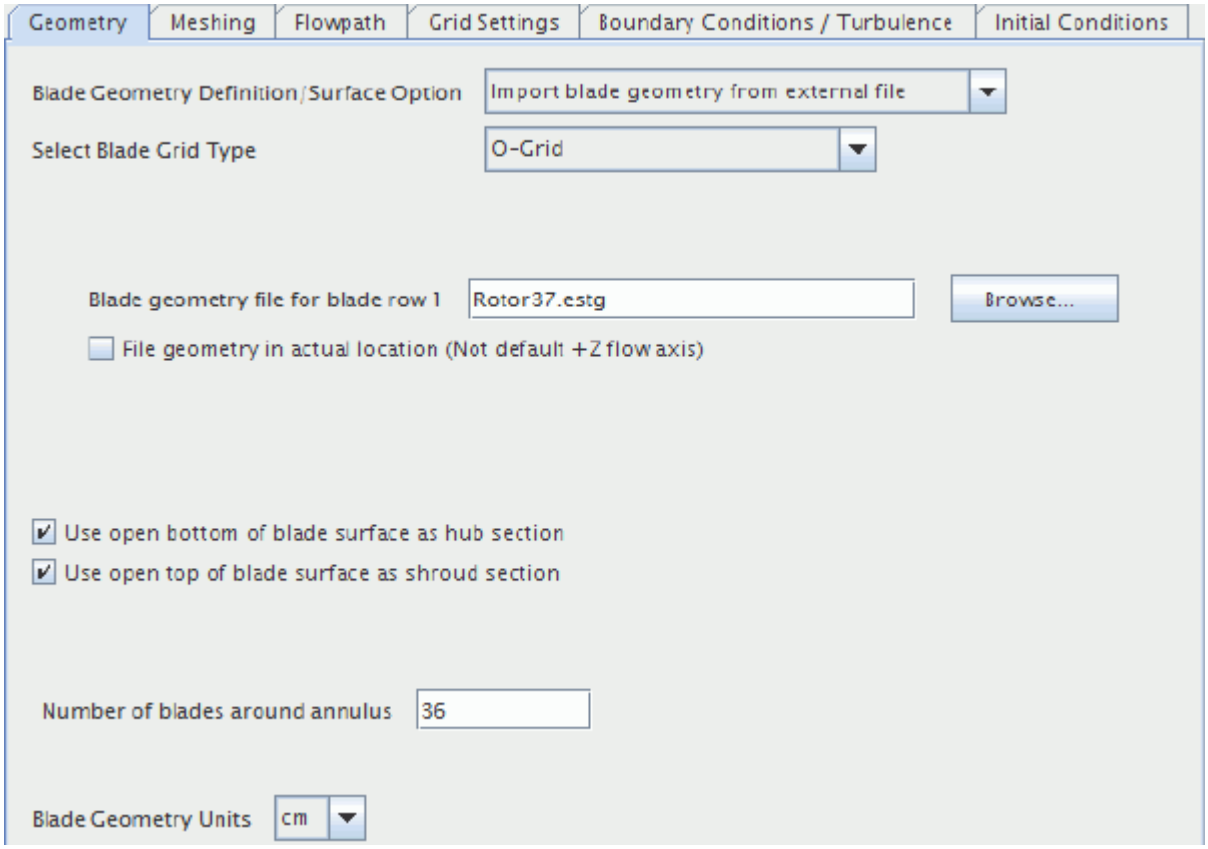
Property	Setting
Select current blade row	2
Current blade row is rotating	Activated, `17200 rpm`
Hub surface is rotating	Activated



Geometry Tab

2. Click **Browse**, navigate to `Rotor37.estg`, then click **Open**.
3. Set the following properties:

Property	Setting
Select Blade Grid Type	**O-Grid**
Blade geometry file for blade row 2	`Rotor37.estg`
Use open bottom of blade surface as hub section	Activated
Use open top of the blade surface as shroud section	Activated
Number of blades around annulus	`36`
Blade Geometry Units	**cm**



Meshing Tab

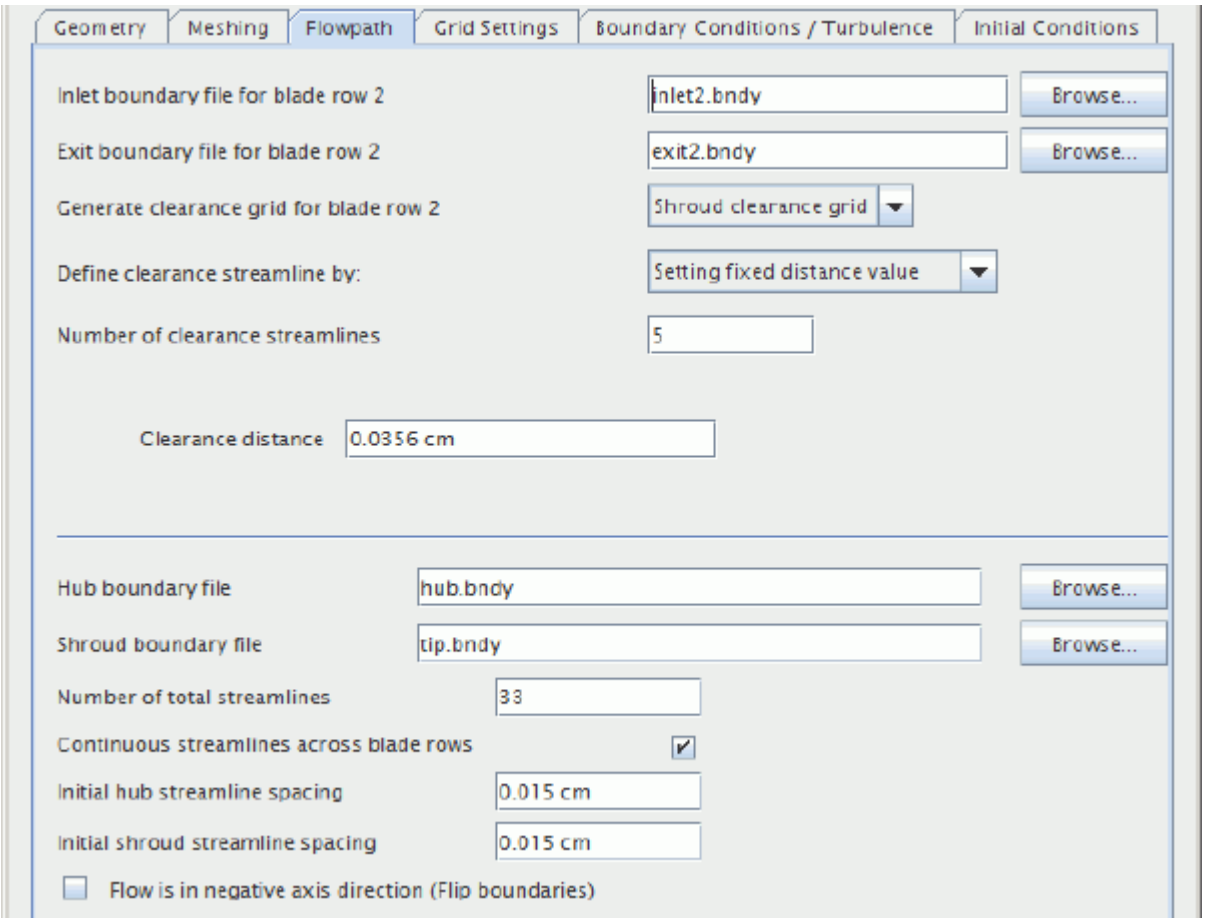
4. Set **Mesh generation method** to **Build 3d volume mesh using elliptical gridder**.

Flowpath Tab

5. Select the following files and set the following properties:

Property	Setting
Inlet boundary file for blade row 2	`inlet2.bndy`
Exit boundary file for blade row 2	`exit2.bndy`
Generate clearance grid for blade row 2	**Shroud clearance grid**
Define clearance streamline by	**Setting fixed distance value**
Number of clearance streamlines	`5`
Clearance distance	`0.0356 cm`
Hub boundary file	`hub.bndy`
Shroud boundary file	`tip.bndy`
Number of total streamlines	`33`

Continuous streamlines across blade rows	Activated
Initial hub streamline spacing	`0.015 cm`
Initial shroud streamline spacing	`0.015 cm`

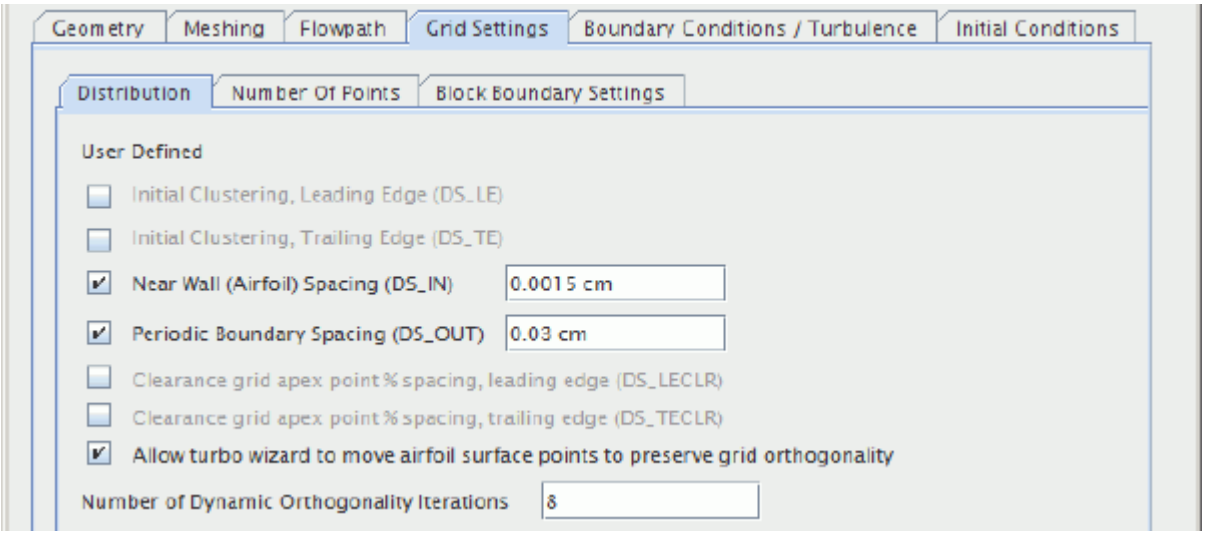


Grid Settings Tab

6. Select the *Distribution* subtab and set the following properties:

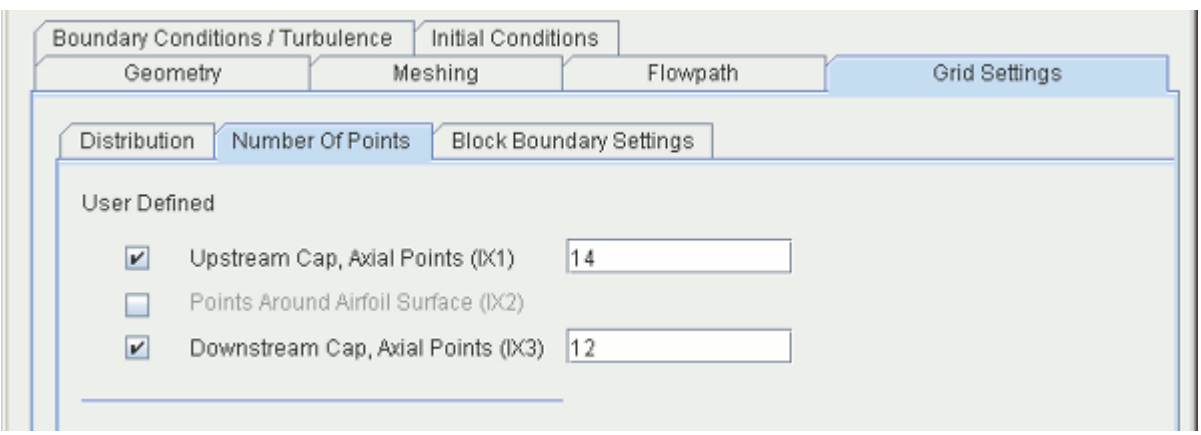
Property	Setting
Near Wall (Airfoil) Spacing	Activated, `0.0015 cm`
Periodic Boundary Spacing	Activated, `0.03 cm`
Allow turbo wizard to move airfoil surface point to preserve grid orthogonality	Activated

Leave the number of iterations at the default setting.



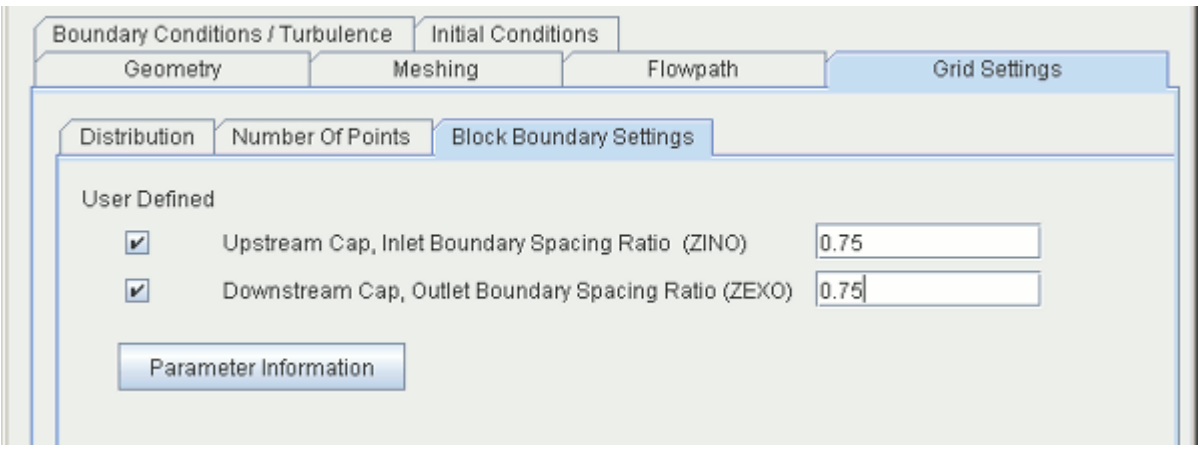
7. Select the *Number Of Points* subtab and set the following properties:

Property	Setting
Upstream Cap, Axial Points	Activated, `14`
Downstream Cap, Axial Points	Activated, `12`



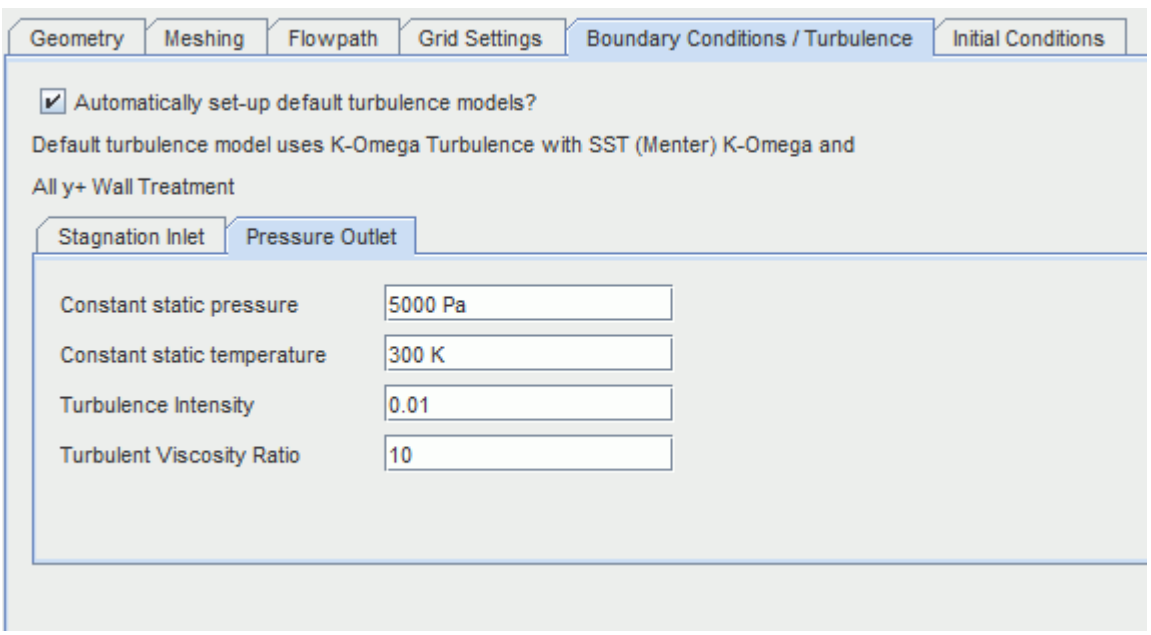
8. Select the *Block Boundary Settings* subtab and set the following properties:

Property	Setting
Upstream Cap, Inlet Boundary Spacing Ratio	Activated, `0.75`
Downstream Cap, Outlet Boundary Spacing Ratio	Activated, `0.75`



Boundary Conditions / Turbulence Tab

9. Select the *Pressure Outlet* subtab, change the **Constant static pressure** value to `5000 Pa`, and leave the other parameters at their defaults



10. Click **Save Turbo Wizard Settings** and, in the *Save Turbo Wizard Data* dialog, navigate to the `axialMultipleRow` directory created previously.
11. Enter `turboWizardData` for the **File Name** then click **Save**.
12. Click **OK** to generate the volume mesh.

STAR-CCM+ starts creating the grid plane for all the streamlines for row (stage) 1 and then 2. Once it completes building the mesh, it converts both meshes into finite volume representation. See the *Output* window for more details. At the end of the import process, it also creates an initial (default) simulation tree in the *Star 1* window.

To create a file for the simulation, and give it a proper name:

13. Select **File** > **Save** from the menu bar.
14. Type `multipleRow` in the **File Name** text box then click **Save**.

Note: the title of the simulation window also changes to *multipleRow*.

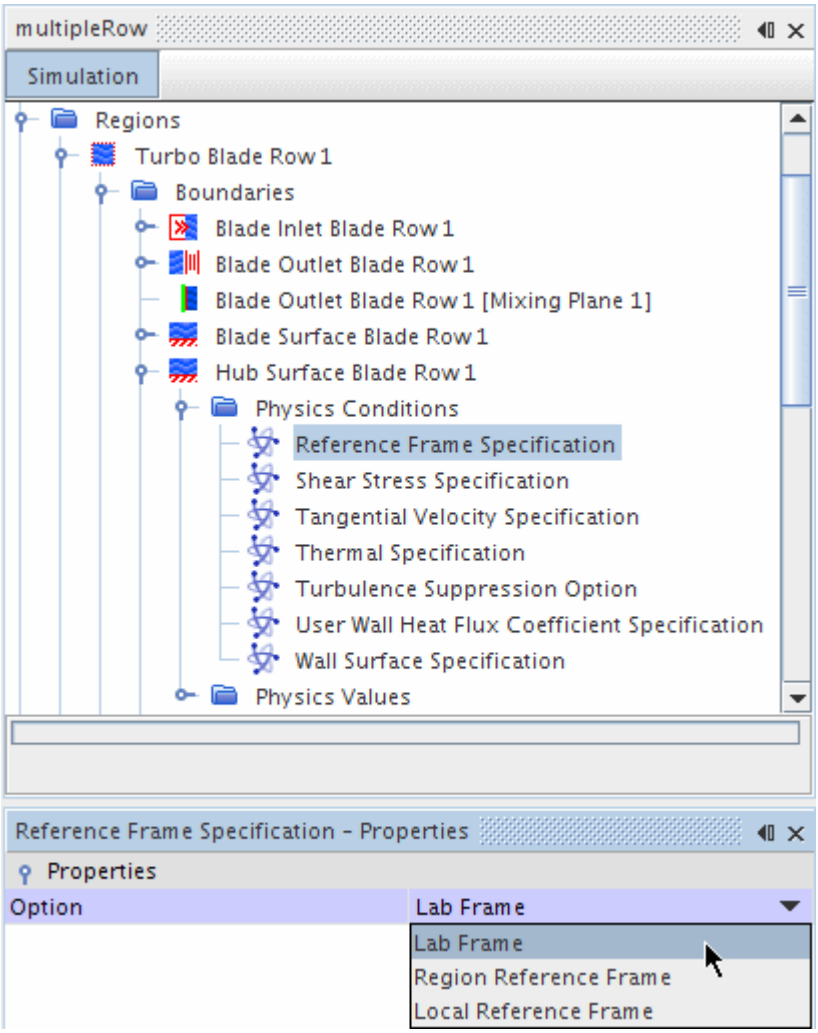
Checking the Models

The Turbo Wizard has generated a steady-state flow model in which:

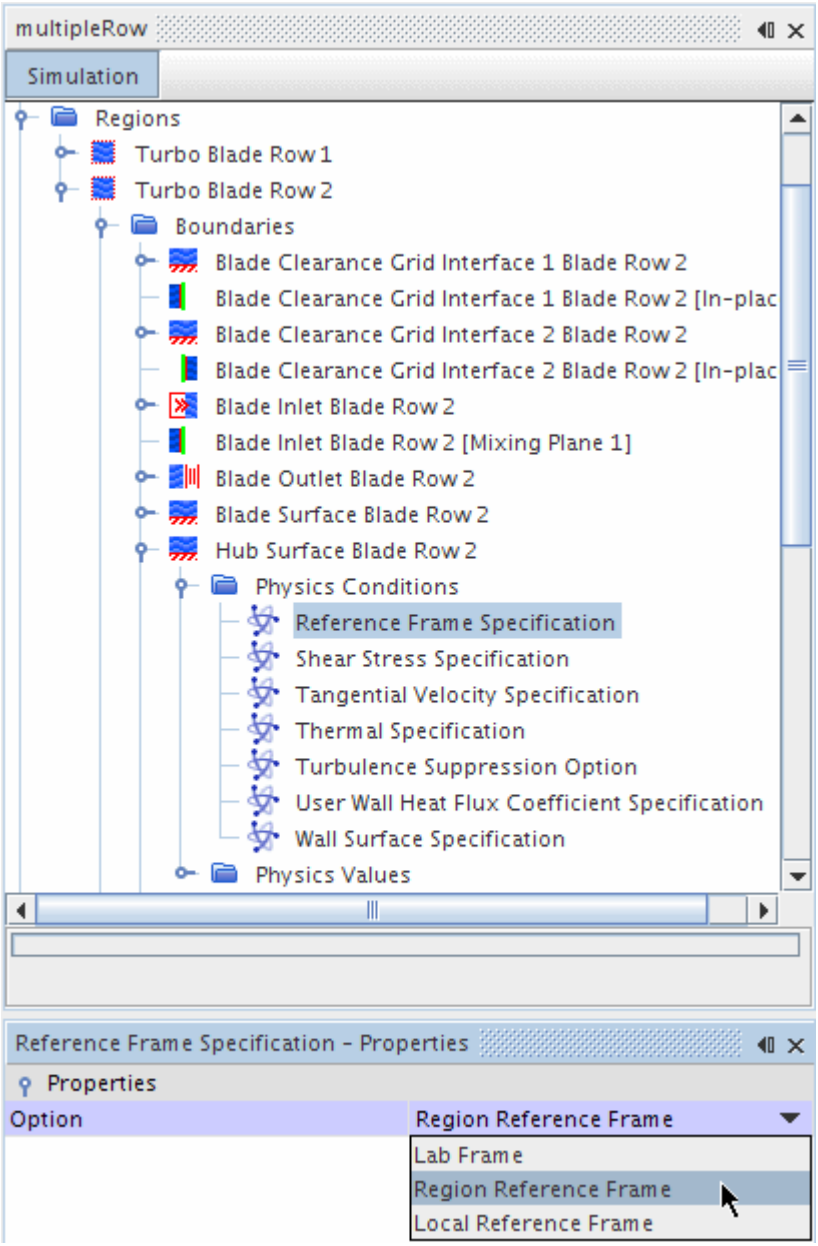
- A three-dimensional grid is employed
- The fluid is a compressible gas (air) that obeys the ideal gas equation of state
- The flow is turbulent and non-isothermal
- The K-Omega model is used for representing turbulence effects
- The problem is solved using the coupled flow solver

These settings are all made automatically when using the Turbo Wizard, and so no further selections are required. The models can be checked as described in the corresponding section of the Axial Single Row tutorial.

1. Expand the **Regions** > **Turbo Blade Row 1** > **Boundaries** node, go to the **Hub Surface Blade Row 1** > **Physics Conditions** node, and select **Reference Frame Specification**. Set the *Option* property to **Lab Frame**.

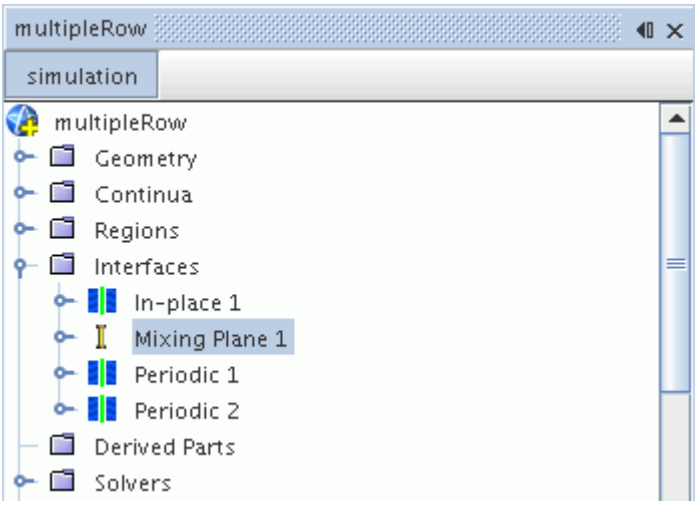


2. Expand the **Regions** > **Turbo Blade Row 2** > **Boundaries** node, go to the **Hub Surface Blade Row 2** > **Physics Conditions** node, and select **Reference Frame Specification**. Set the *Option* property to **Region Reference Frame**.

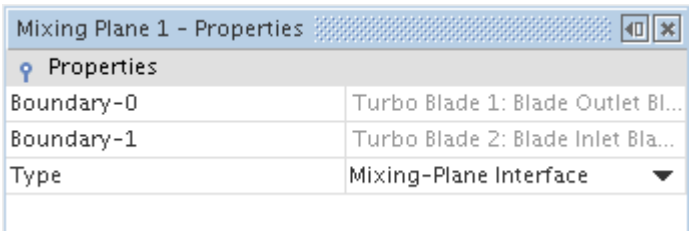


Note: the interface between the IGV and rotor is automatically set up as an implicit mixing-plane interface. The presence of this interface can be checked by expanding the Interfaces node.

3. Expand the **Interfaces** node and select node **Mixing Plane 1**



The *Properties* window shows the boundaries that are used in the composition of the interface.



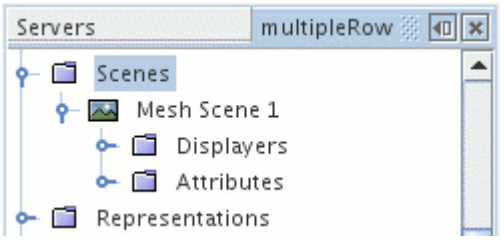
An in-place interface has been created for the tip clearance. Repeat the process above to inspect its property settings.

Visualizing the Mesh

To examine the mesh:

Right-click the **Scenes** node and select **New Scene** > **Mesh** .

This creates a node, **Mesh Scene 1**, in the simulation tree.

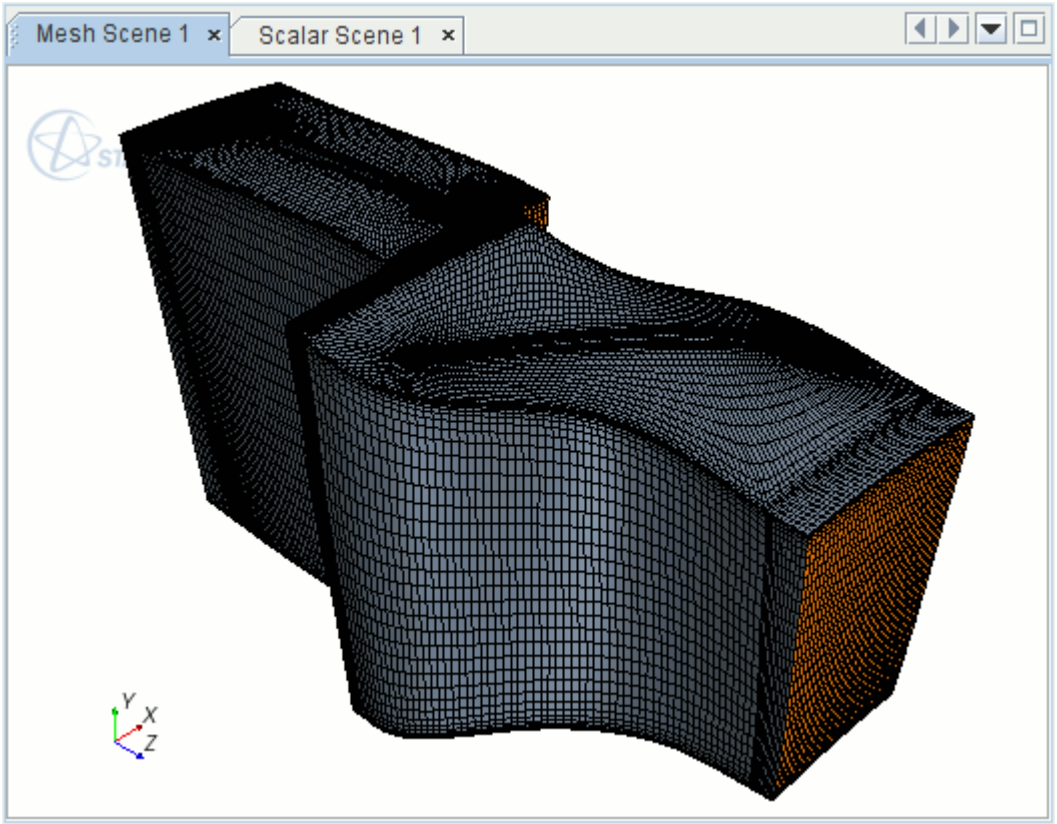


In addition to the **Mesh Scene 1** node, the *Graphics* window also shows the mesh. Examine the mesh more closely by holding and dragging:

- the left mouse button to rotate it

- the right button to pan it
- the middle button to zoom in and out

and produce a *Graphics* window display such as the one shown below.



Verify that no malformed or irregular cells are present on the mesh surface.

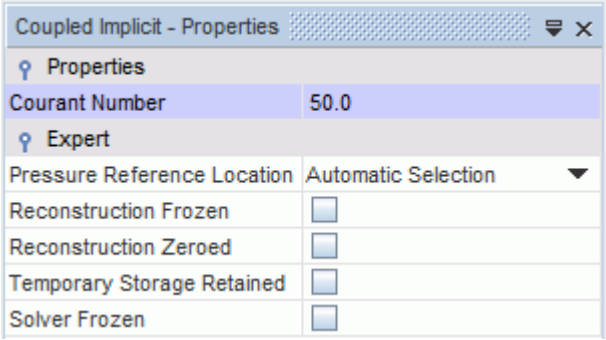
View each of the interfaces to check that they have been created appropriately.

Setting Up Solver and Stopping Criteria

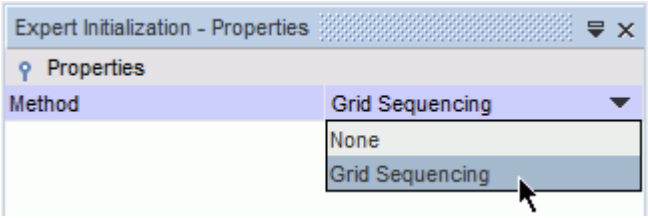
By default, the solver parameters are set to the **Coupled Implicit** with a *Courant Number* of `5.0`.

Increase this number to reduce the number of iterations the solver needs to converge. Also utilize Grid Sequencing and Expert Driver to make the solver more robust and control the stability of the solution.

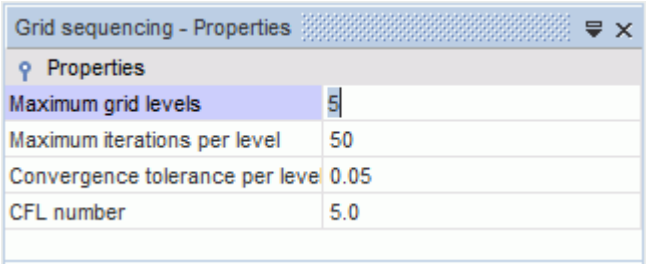
1. Open the **Solvers** node and then select the **Coupled Implicit** node and set the Courant Number to `50.0`.



2. Open the **Solvers** > **Coupled Implicit** > **Expert Initialization** node and set the **Method** to **Grid Sequencing**.



3. Open the **Grid sequencing** node and set the **Maximum grid levels** to 5.

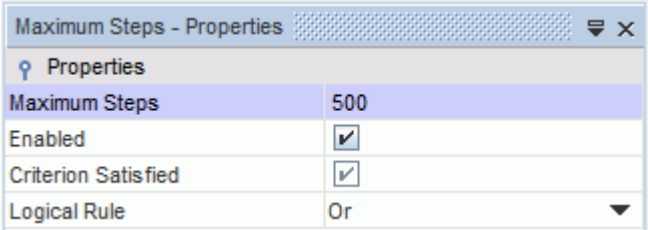


Grid Sequencing creates coarse grid levels that are based on the specified maximum grid levels setting. It is necessary to reduce the maximum number of grid levels in this case because you are using few cells in your original mesh. If there are too many grid levels relative to the number of cells in the mesh, the coarsest of the levels might end up with cell numbers that are too small for the solver to converge.

4. Open the **Solvers** > **Coupled Implicit** > **Solution Driver** node and set **Driver** to **Expert driver**. Use the default settings of the Expert Driver.

Before running the analysis, reduce the maximum number of steps.

5. Open the **Stopping Criteria** node and then select the **Maximum Steps** node and set the *Maximum Steps* value to 500.

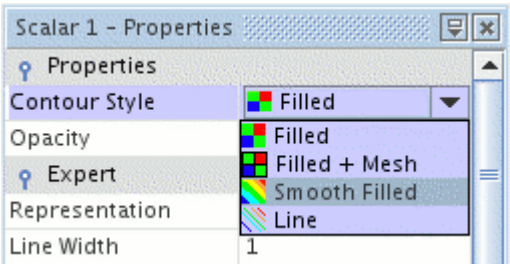


Visualizing the Solution

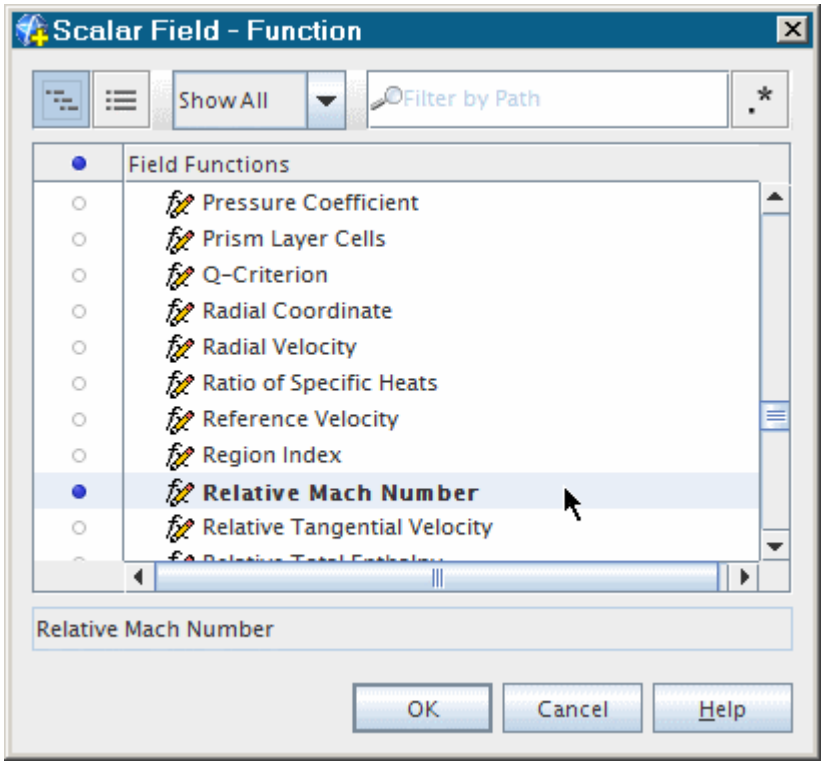
In this section you define three isosurfaces including blade-to-blade and meridional views, to create a Relative Mach number scene.

1. Right-click the **Scenes** node and select **New Scene** > **Scalar**.

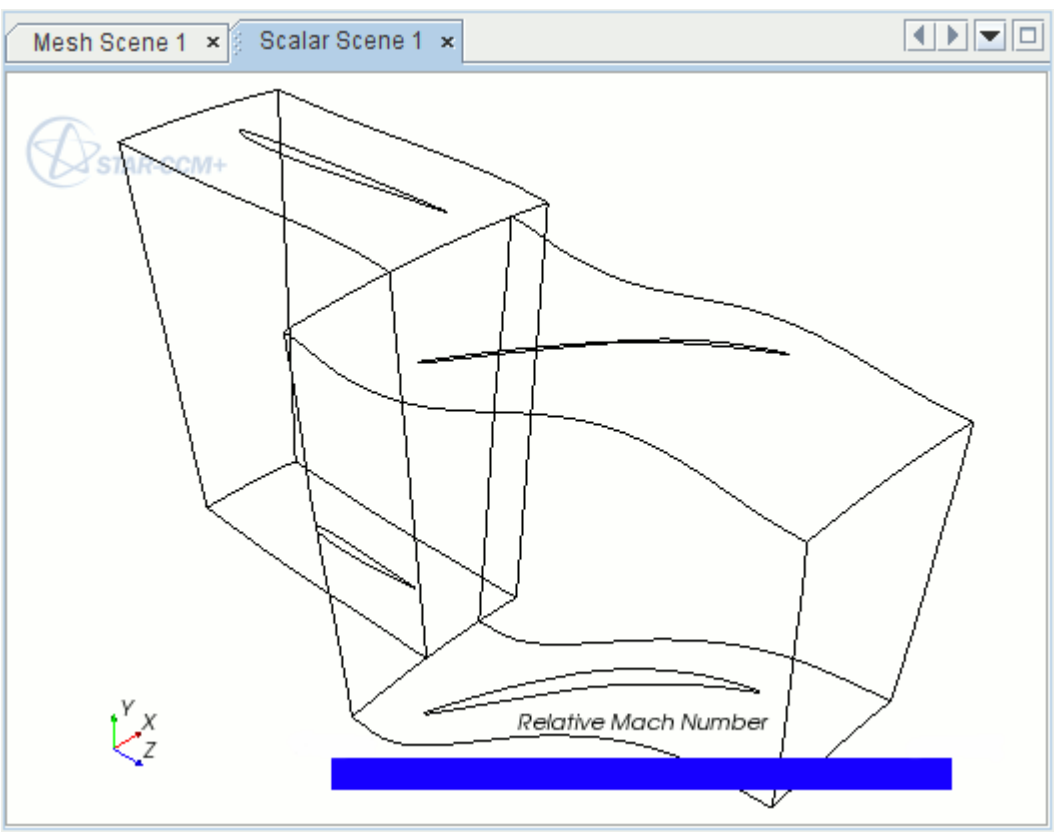
   A new node that is called **Scalar Scene 1** is created in the simulation tree.
2. Open the **Scalar Scene 1** > **Displayers** > **Scalar 1** node and click the right half of the *Contour Style* property, then select **Smooth Filled** from the pop-up menu.



3. Select the **Scalar Field** node and edit the *Function* property.
4. Select **Relative Mach Number** from the list displayed.



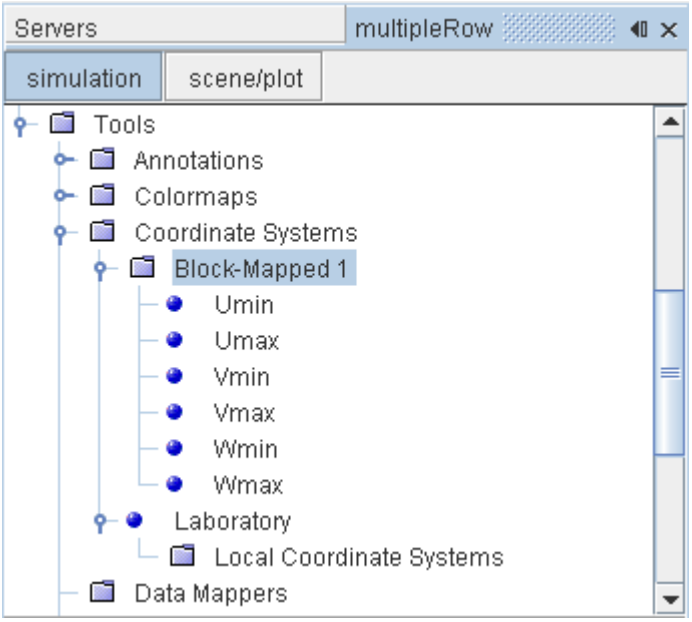
5. Use the mouse buttons to change the direction of the *Scalar Scene 1* to correspond roughly to that of the *Mesh Scene 1* above.



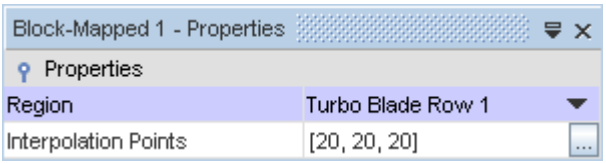
Define a block mapped coordinate system for the IGV.

6. Open the **Tools** node.
7. Right-click the **Coordinate Systems** node and select **New Block-Mapped**.

   A node, **Block-Mapped 1**, appears in the simulation tree as soon as this operation is complete.

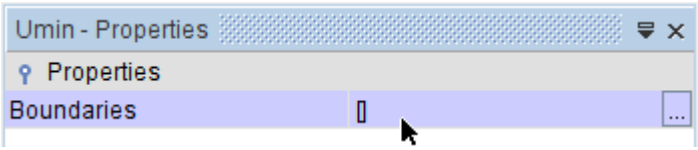


8. Select the **Block-Mapped 1** and set *Region* to **Turbo Blade Row 1**.

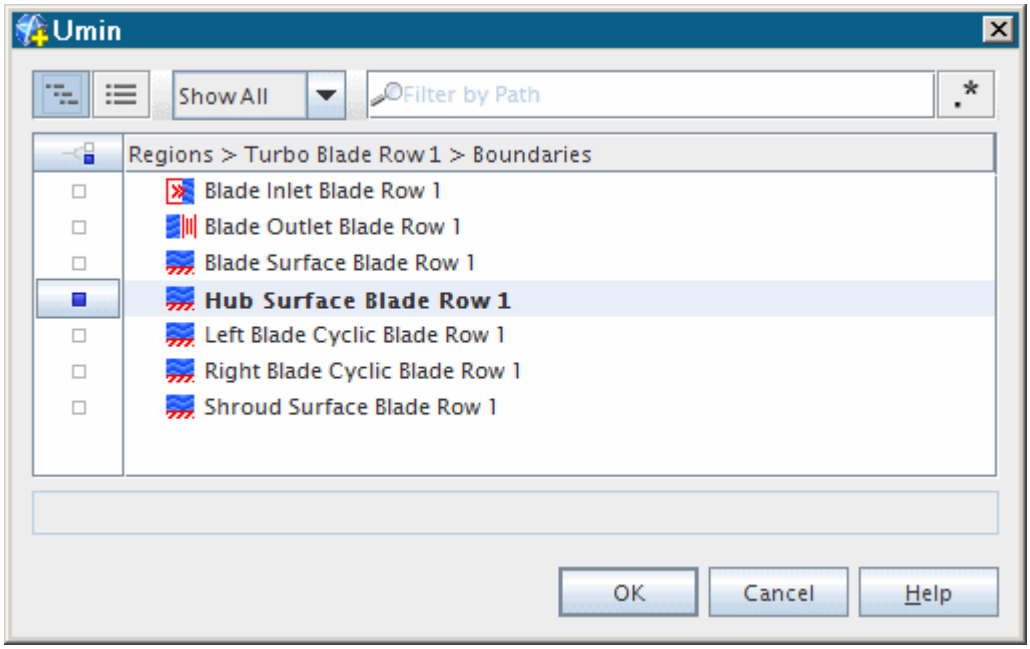


9. Click **Umin** and click to the right of the *Boundaries* property.

   This opens the *Select Objects* dialog.



10. Select **Regions** > **Turbo Blade Row 1** > **Hub Surface Blade Row 1** and click **OK**.

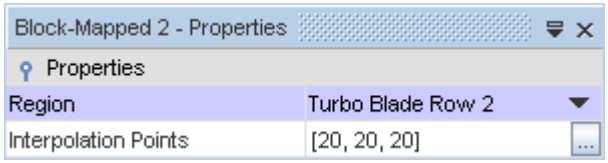


11. Repeat the above steps for the following nodes:

Node	Property	Setting
Umax	*Boundaries*	**Shroud Surface Blade Row 1**
Vmin	*Boundaries*	**Left Blade Cyclic Blade Row 1**
Vmax	*Boundaries*	**Right Blade Cyclic Blade Row 1**
Wmin	*Boundaries*	**Blade Inlet Blade Row 1**
Wmax	*Boundaries*	**Blade Outlet Blade Row 1**

Now follow the same steps to define a block mapped coordinate system for the Rotor37 blade:

12. Right-click the **Coordinate Systems** node and select **New Block-Mapped**.
13. Select the **Block-Mapped 2** node and set *Region* to **Turbo Blade Row 2**.



14. Set the following properties:

Node	Property	Setting
Umin	*Boundaries*	**Hub Surface Blade Row 2**
Umax	*Boundaries*	**Shroud Surface Blade Row 2**

Vmin	*Boundaries*	**Left Blade Cyclic Blade Row 2**
Vmax	*Boundaries*	**Right Blade Cyclic Blade Row 2**
Wmin	*Boundaries*	**Blade Inlet Blade Row 2**
Wmax	*Boundaries*	**Blade Outlet Blade Row 2**

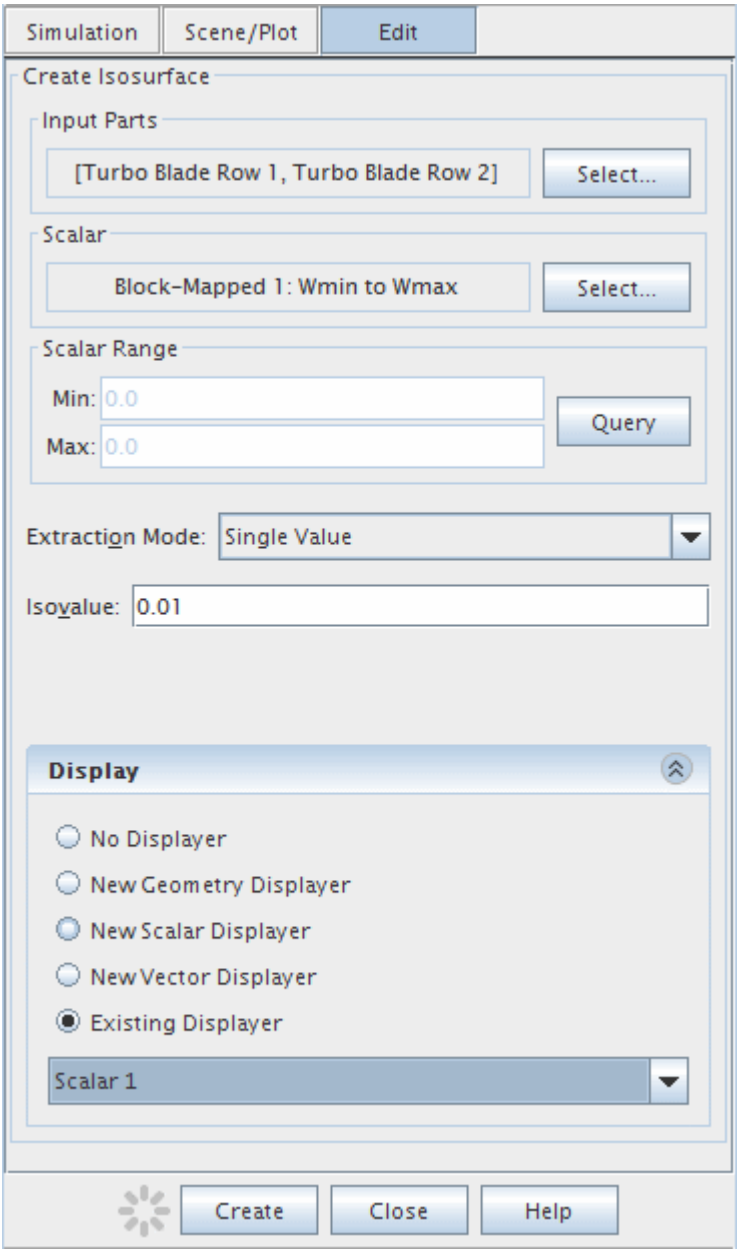
In this section, Isosurfaces for the IGV are created.

15. Right-click the **Derived Parts** node and select **New Part** > **Isosurface**.

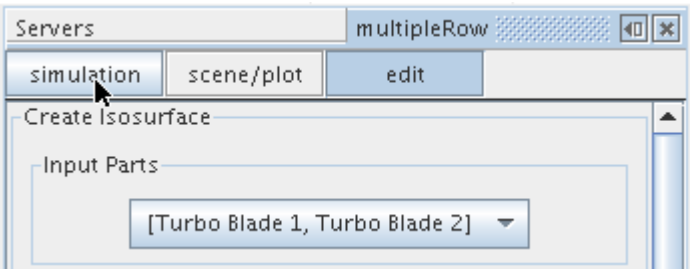
    A new interactive in-place dialog (*Create Isosurface*) appears on the left of the *Graphics* window to help you define the desired surface.

16. Set the following properties:

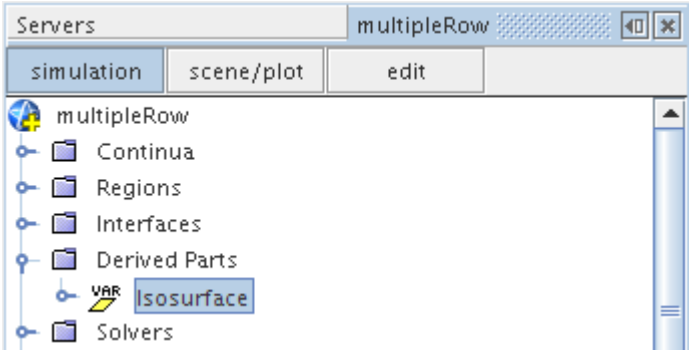
Property	Setting
Scalar	**Block-Mapped 1: Wmin to Wmax**
Iso value	`0.01`
Display	**Existing Displayer**, **Scalar 1**



17. Click **Create**.
18. Switch back to the object tree using the **simulation** button above the *Create Isosurface* dialog.



A node, **Isosurface**, appears under the **Derived Parts** node in the simulation tree.



19. Switch back to the *Create Isosurface* dialog using the **edit** button.
20. Set the following properties:

Property	Setting
Scalar	**Block-Mapped 1: Umin to Umax**
Iso value	`0.5`

21. Click **Create**.

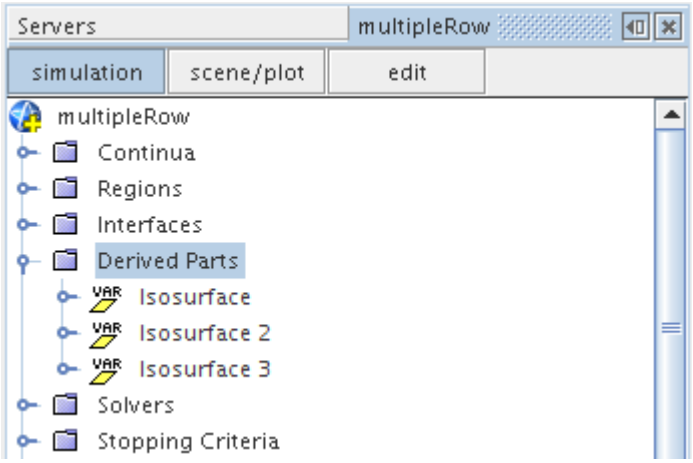
A new node, **isosurface 2**, appears under the **Derived Parts** node in the simulation tree.

22. Set the following properties:

Property	Setting
Scalar	**Block-Mapped 1: Vmin to Vmax**
Iso value	`0.01`

23. Click **Create**.

    A new node, **isosurface 3**, appears under the **Derived Parts** node in the simulation tree. All three nodes are shown below.



24. Rename the nodes:

Old Name	New Name
isosurface	`AxialViewIGV`
isosurface 2	`Blade2BladeViewIGV`
isosurface 3	`MeridionalViewIGV`

Follow the above outlined procedure to create the isosurfaces for the Rotor blade.

25. Create the following isosurfaces:

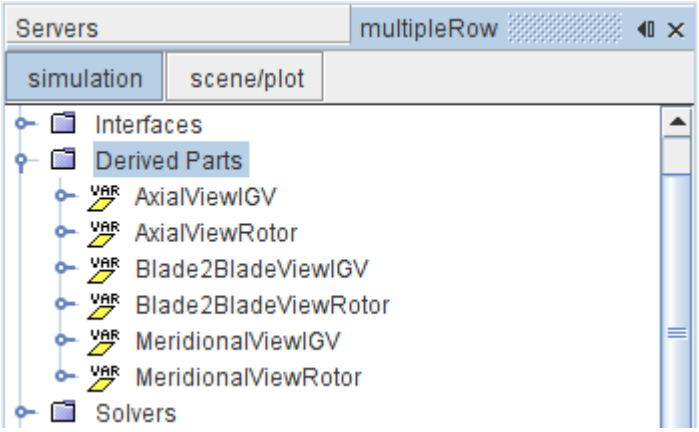
Isosurface Scalar	Iso Value
Block-Mapped 2: Wmin to Wmax	`0.01`
Block-Mapped 2: Umin to Umax	`0.5`
Block-Mapped 2: Vmin to Vmax	`0.01`

26. Click **Close** on the *Create Isosurface* dialog when these operations are complete.

This results in three new nodes with the same default names as before (**isosurface**, **isosurface 2**, and **isosurface 3**).

27. Rename the nodes:

Old Name	New Name
isosurface	`AxialViewRotor`
isosurface 2	`Blade2BladeViewRotor`
isosurface 3	`MeridionalViewRotor`

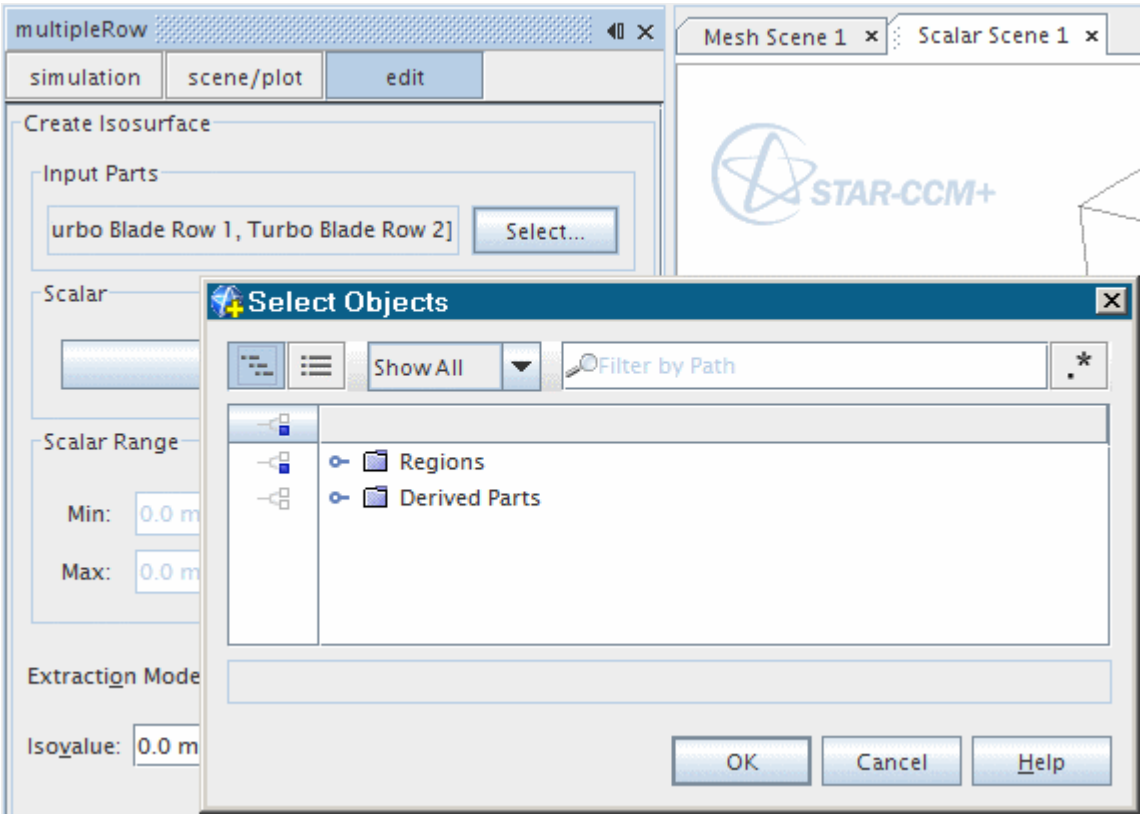


Note: the reordering that is shown above is achieved by right-clicking the **Derived Parts** node and selecting **Refresh** from the pop-up menu.

To create a plot of blade profile pressure variation, you now create an isosurface interaction at the blade surface.

28. Right-click the **Derived Parts** node and select option **New Part** > **Isosurface** as before.
29. In the *Create Isosurface* dialog, click in the **Input Parts** box.

    This opens the *Select Objects* dialog as shown in the following screenshot.



30. Use the column selection button to clear the selections.
31. Select **Regions** > **Turbo Blade Row 2** > **Blade Surface Blade Row 2** and click **OK**
32. Set the following properties:

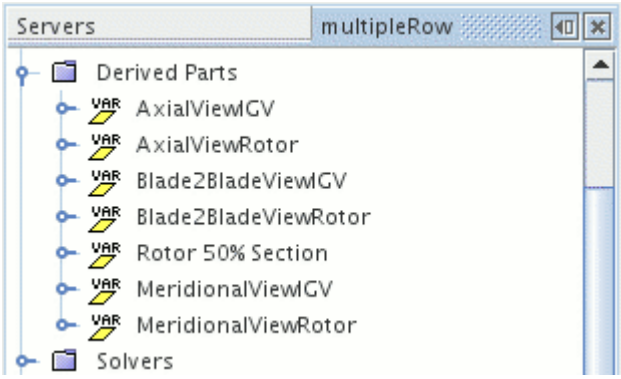
Property	Setting
Scalar	**Block-Mapped 2: Umin to Umax**
Iso value	`0.5`
Display	**Existing Displayer**, **Scalar 1**

33. Click **Create** and then **Close**.

A new node, **isosurface**, appears under the **Derived Parts** node in the simulation tree.

34. Rename the **isosurface** node to `Rotor 50% Section`

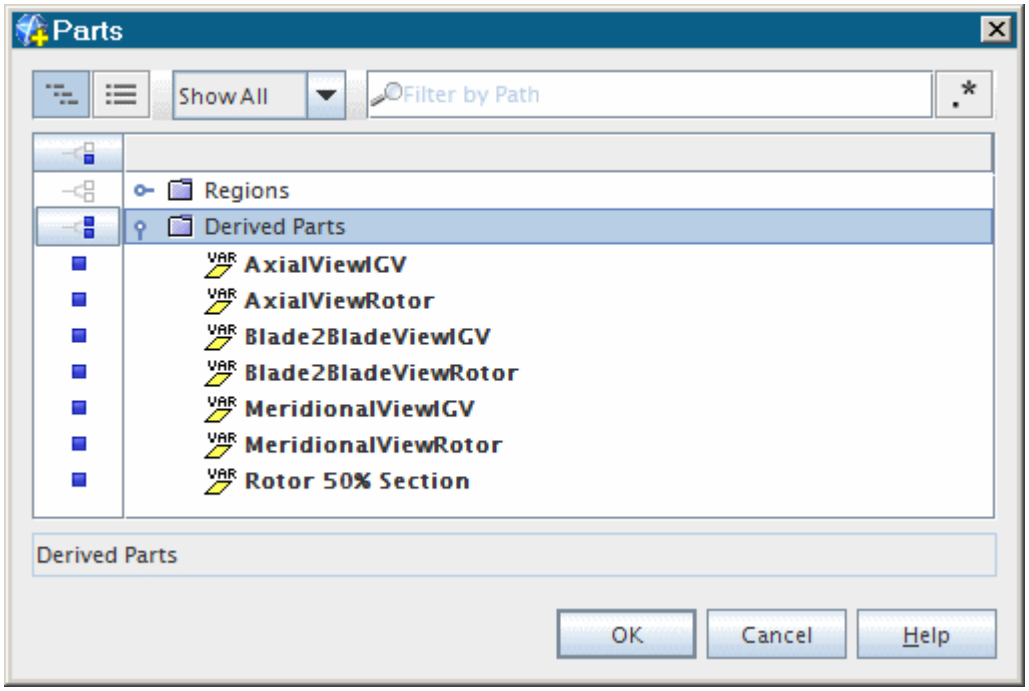
At the end of a successful completion of creating isosurfaces, the simulation tree contains the following additional nodes.



35. Select the **Scenes** > **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Parts** node and click the right half of the *Parts* property.

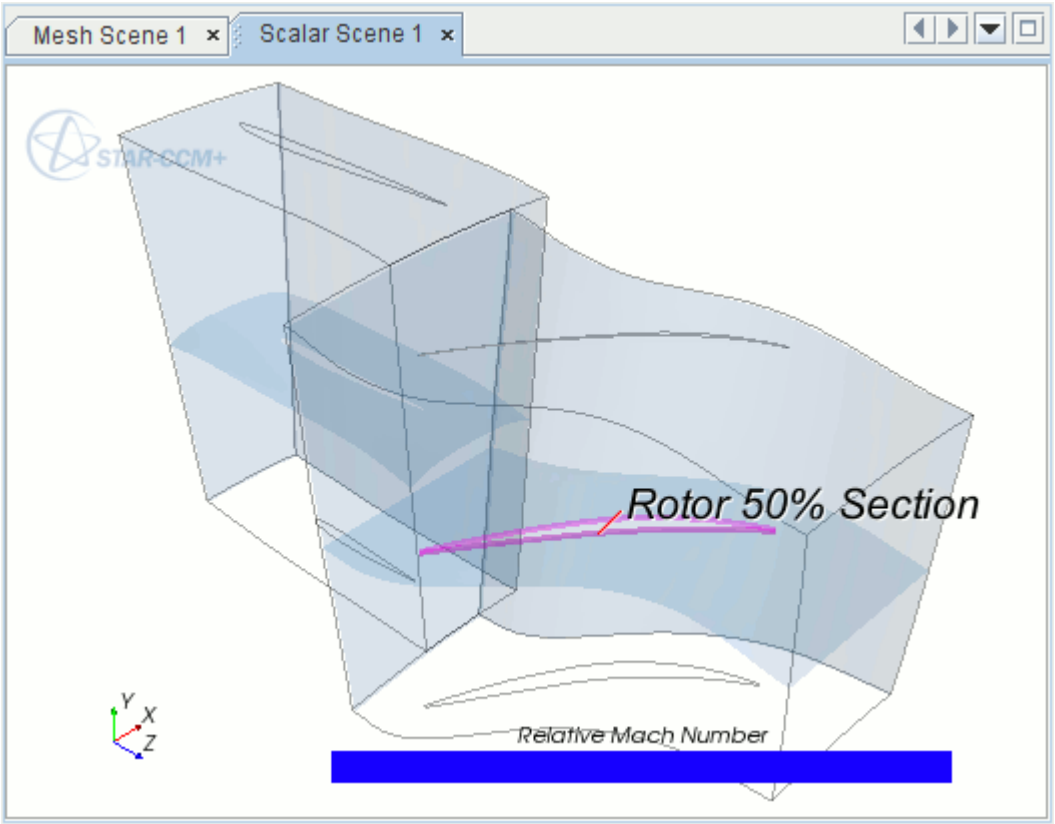
The *Select Objects* dialog appears.

36. Make sure that only those objects belonging to the above defined isosurfaces are selected, as shown below



37. Click **OK**.

38. Click **(Make Scene Transparent)** to display the above defined isosurfaces as shown below.



In the above picture, the Rotor 50% Section isosurface is superimposed to show its location more clearly by clicking its node under the **Derived Parts** node.

39. Click **(Make Scene Transparent)** to set the transparency of the scene to its original setting.
40. Save the simulation.

The pre-processing task is now complete. The next step is to run the analysis, as described in Running the Simulation.

Running the Simulation

Run the simulation.

1. Click **(Run)** to start the solver.

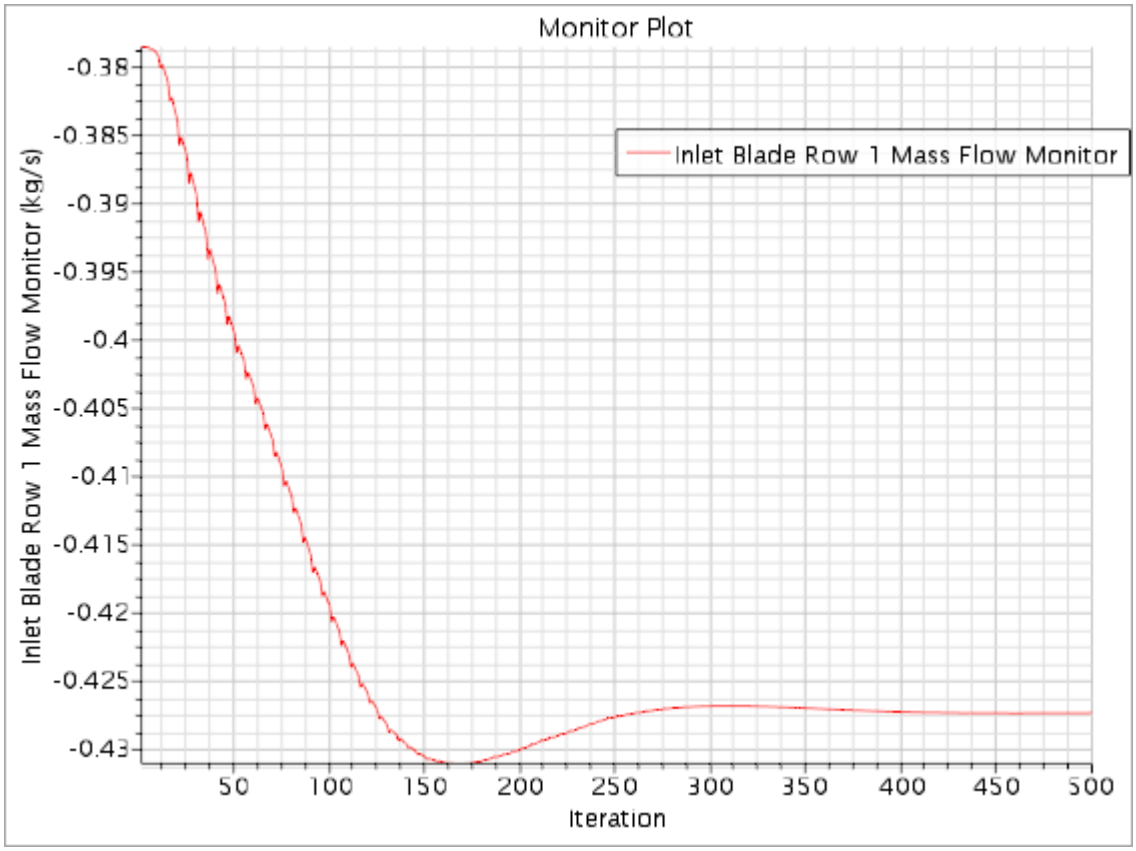
   The analysis now starts and the results, in the form of a graph of residuals vs. iteration number, is displayed automatically in the *Graphics* window.

   The solution stops after 500 steps.
2. When the run is complete, save the simulation.

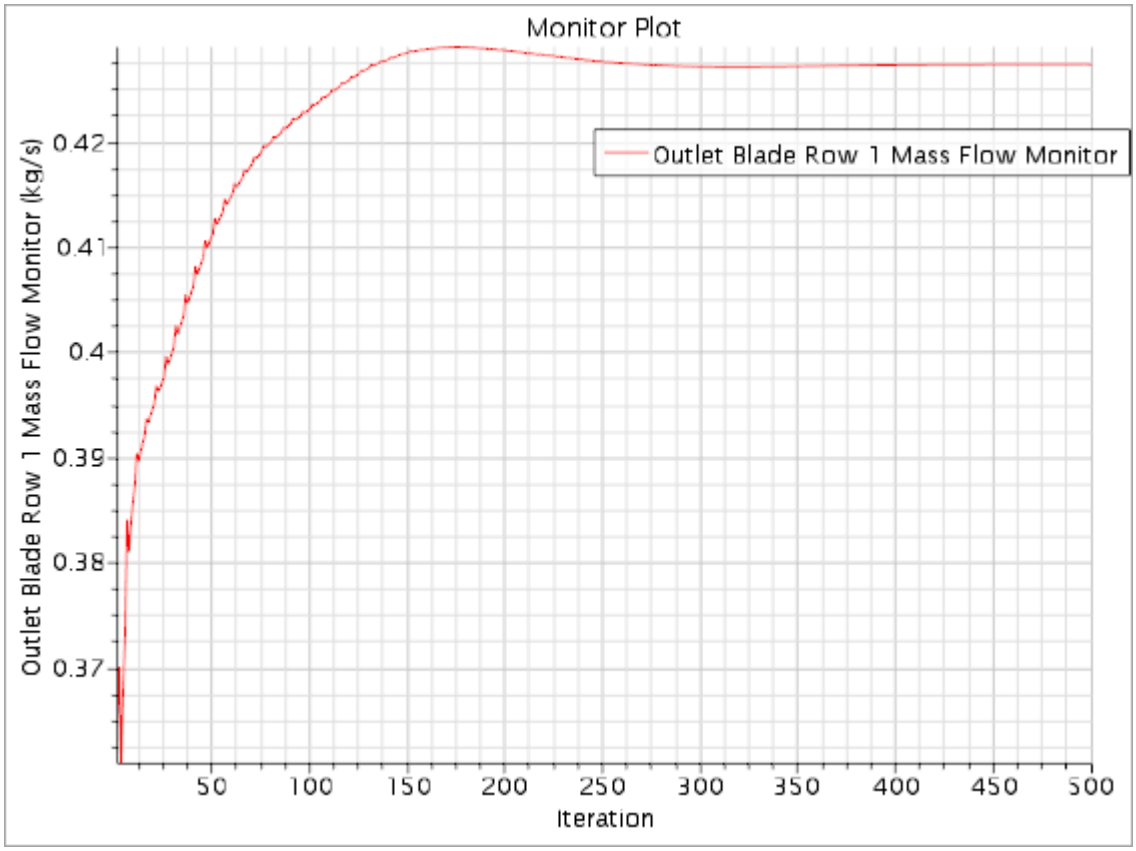
Analyzing Mass Flow Reports

To assess the degree of convergence, it is helpful to look at the inlet and outlet mass flow plots for the IGV and Rotor.

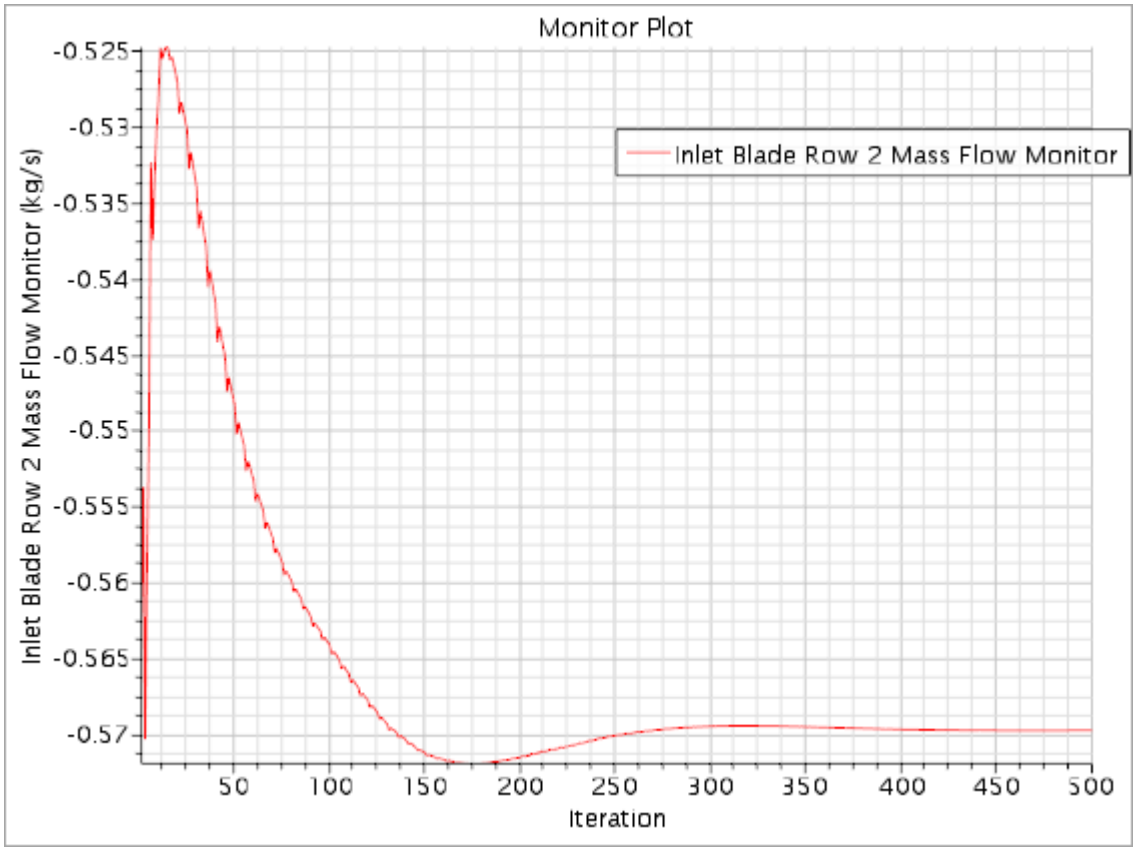
1. Open the **Plots** node.
2. Double-click **Inlet Blade Row 1 Mass Flow Plot**.



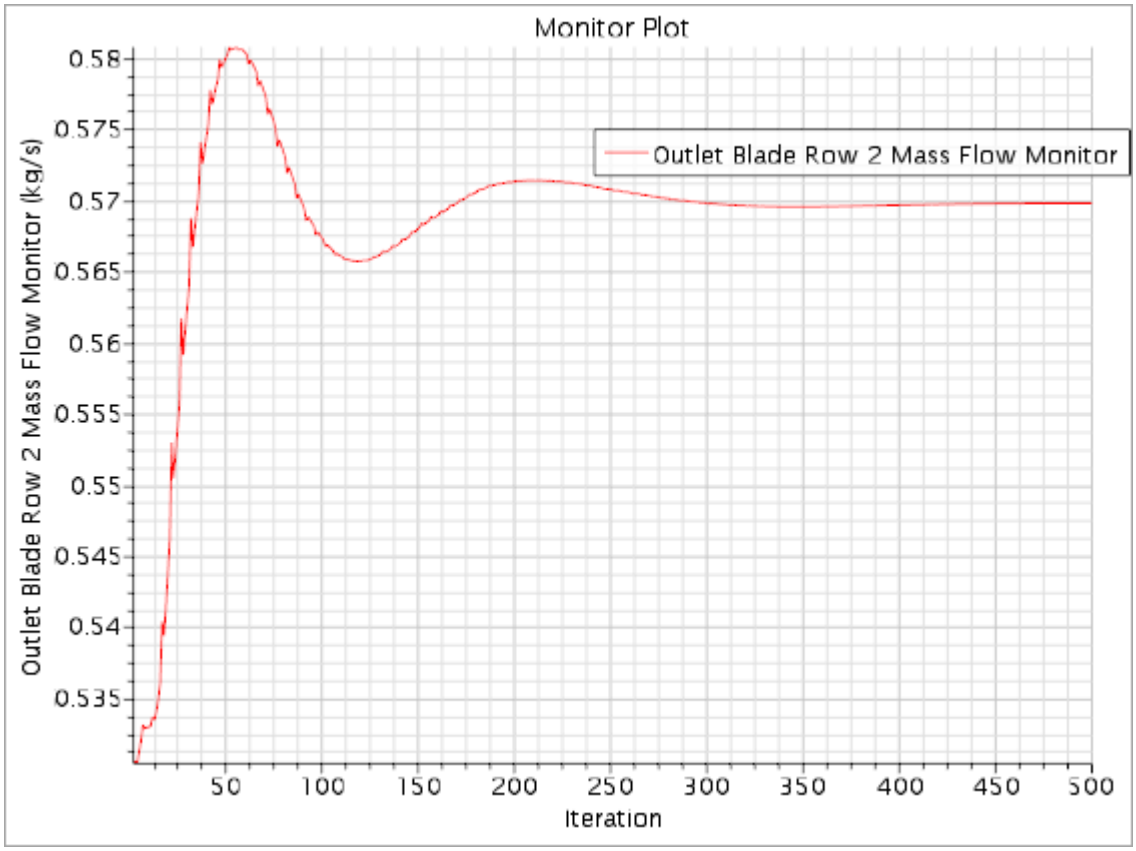
3. Double-click **Outlet Blade Row 1 Mass Flow Plot**.



4. Double-click **Inlet Blade Row 2 Mass Flow Plot**.

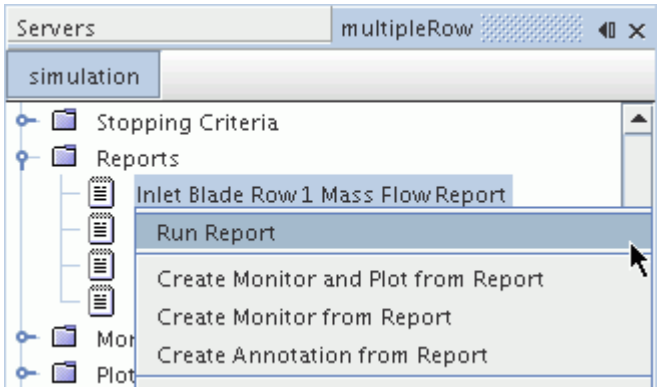


5. Double-click **Outlet Blade Row 2 Mass Flow Plot**.



The mass flows reported for Blade Row 1 are different from those reported for Blade Row 2. This difference is expected as Blade Row 1 has 48 blades and Blade Row 2 has 36 blades. Running the reports manually and multiplying by the appropriate number of blades gives the total mass flow through the system.

6. Right-click **Reports** > **Inlet Blade Row 1 Mass Flow Report**. Select item **Run Report** from the pop-up menu.



The result of this operation is shown in the output window.

```
Output - multipleRow

        Mass Flow

Part                                  Value (kg/s)
------------------------------        -------------
Turbo Blade Row 1: Blade Inlet Blade Row 1        -4.273305e-01
                                      -------------
                         Total:       -4.273305e-01
```

7. Repeat the operation for the **Inlet Blade Row 2 Mass Flow Report** and note the result.

The results from these two reports are multiplied by the respective number of blades to give the overall mass balance.

Mass Flow Report	Reported Mass Flow (kg/s)	Number of Blades	Total Mass Flow (kg/s)
Inlet Blade Row 1	0.42733	48	20.5119
Inlet Blade Row 2	0.56977	36	20.5119

This comparison can be used to assess the overall convergence of the solution between the blade rows for the current computational mesh.

Contents:

Analyzing Mach Number Scalar Scene
Analyzing Blade Pressure Profile Plot

Analyzing Mach Number Scalar Scene

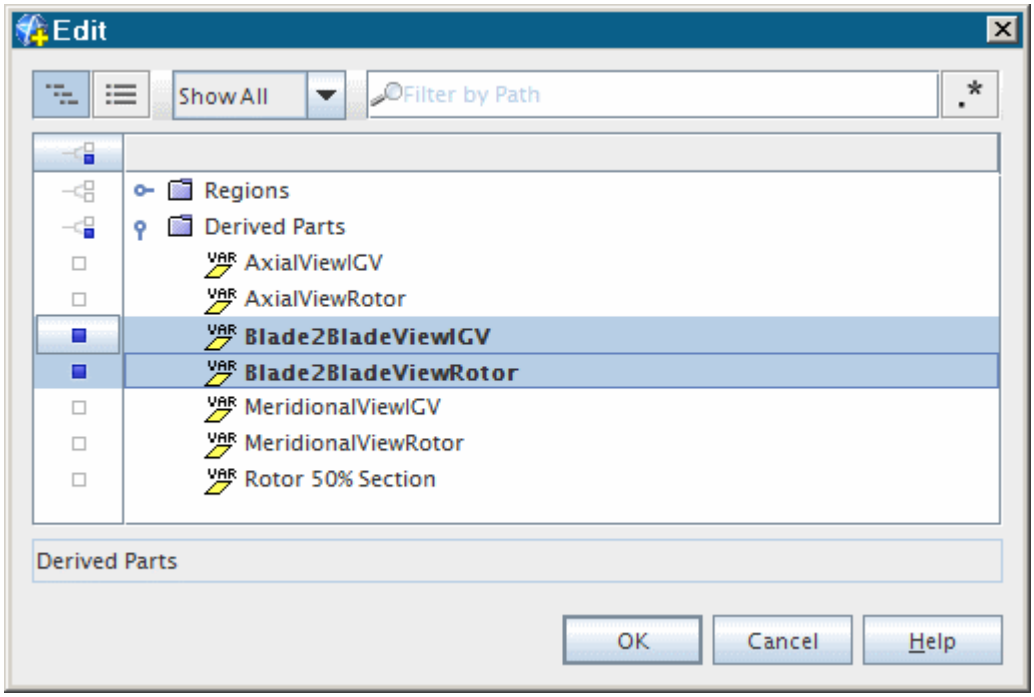
Open the scalar scene showing the Relative Mach Number to view the solution.

To help visualization, only plot the scalar on the Blade2BladeViewIGV and Blade2BladeViewRotor isosurfaces.

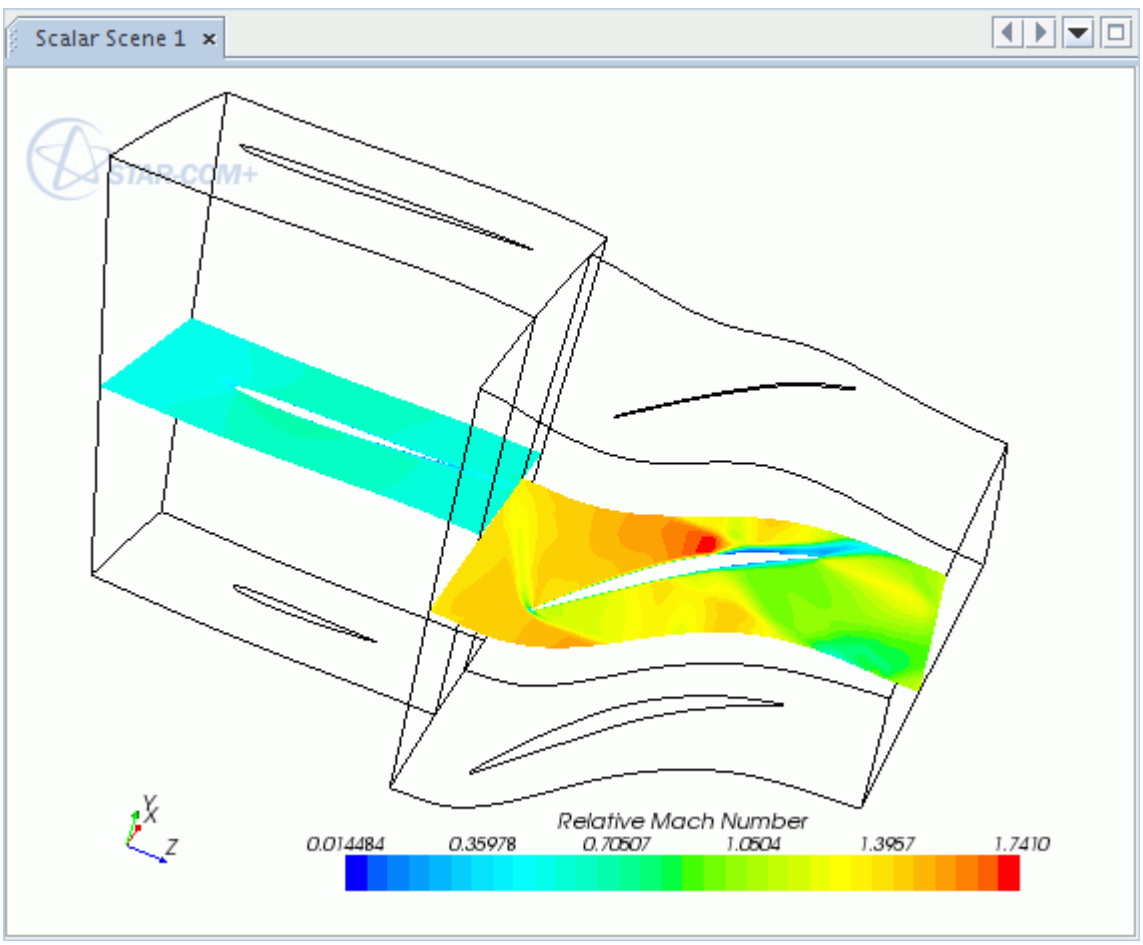
1. Select the **Scenes** > **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Parts** node and click the right half of the *Parts* property.

   The *Select Objects* dialog appears.

2. In the Selected list, de-select all except the **Blade2BladeViewIGV** and **Blade2BladeViewRotor** objects as shown below.



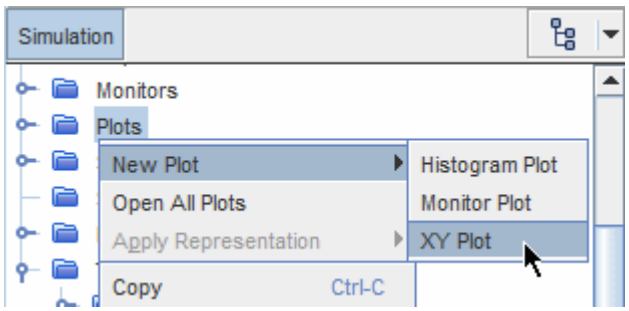
The scene resembles the one shown below.



Analyzing Blade Pressure Profile Plot

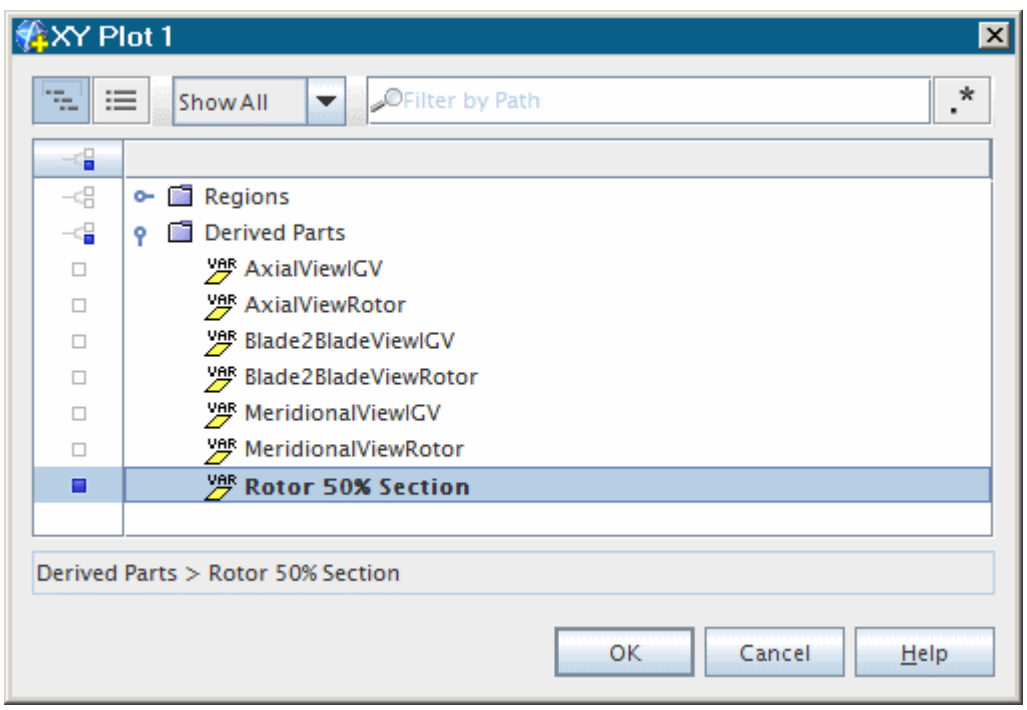
You now create an XY plot of blade profile pressure variation on the Rotor 50% Section isosurface.

1. Right-click the **Plots** node and select **New Plot** > **XY Plot**.



A new node, **XY Plot 1**, appears in the simulation tree and XY Plot 1 graph is displayed automatically in the *Graphics* window. The new **XY Plot 1** node is selected automatically.

2. Click the right half of the *Parts* property and select Derived Parts > Rotor 50% Section.

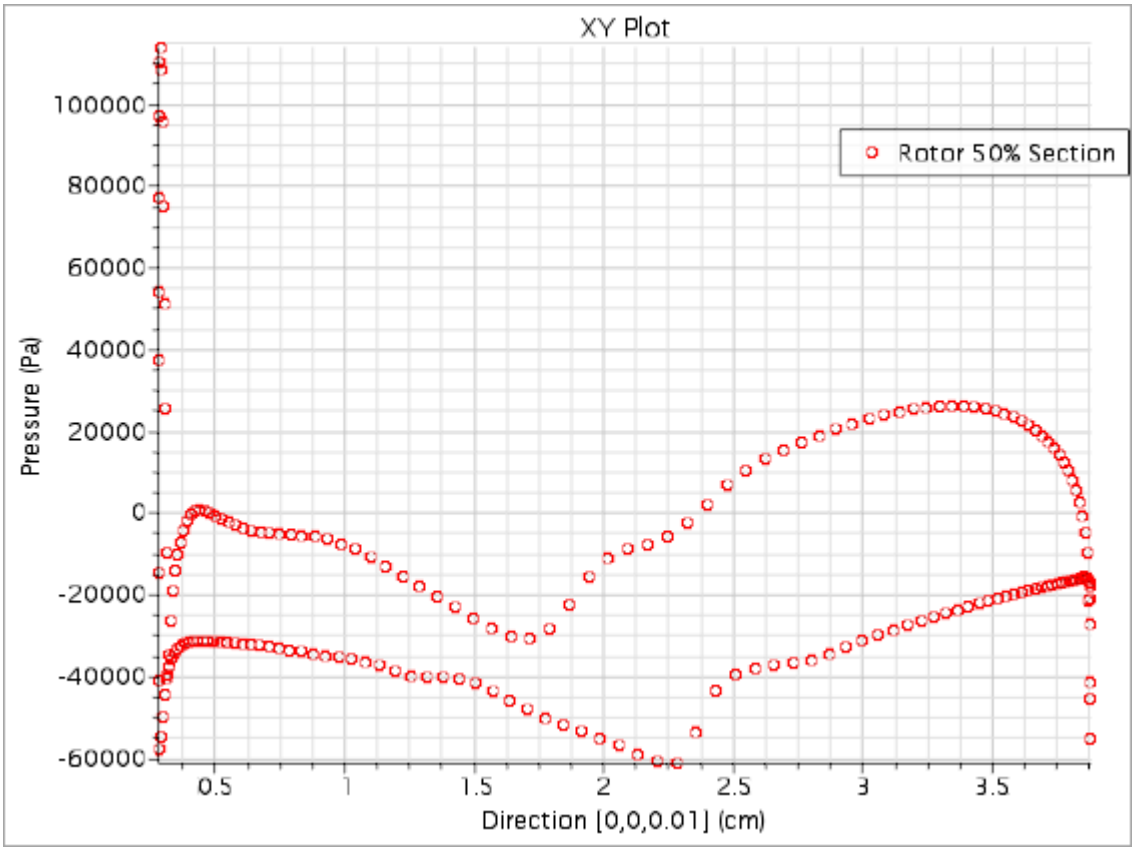


3. Set the following properties:

Node	Property	Setting
XY Plot 1 > **X Type** > **Vector Quantity**	*Value*	`[0, 0, 1] cm`

Y Types > **Y Type 1** > **Scalar Function**	*Field Function*	**Pressure**

The *XY Plot* is displayed as shown in the following screenshot.



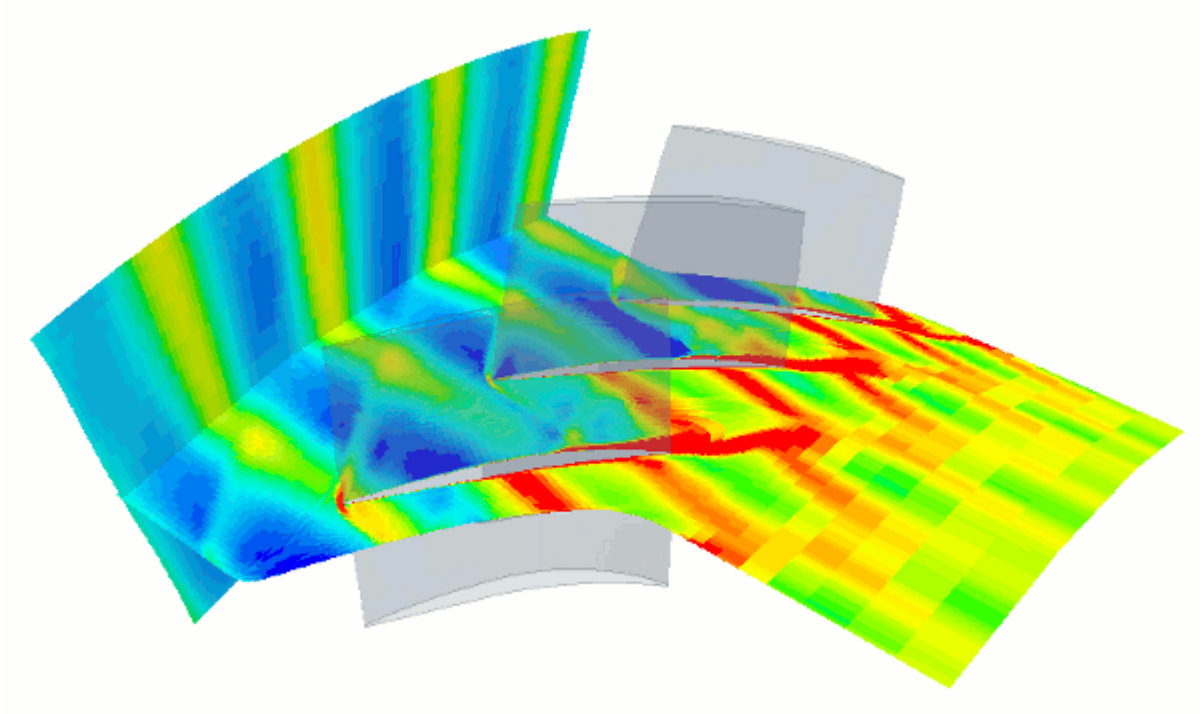
Summary

This STAR-CCM+ tutorial introduced the following features:

- Starting the code and creating a simulation
- Importing the geometry and flowpath boundary data for an axial compressor and Inlet Guide Van blade geometries
- Generating a volume mesh
- Defining the simulation models
- Visualizing the mesh
- Setting up solver and stopping criteria
- Setting up a scalar scene with data from derived parts
- Running the solver for a set number of iterations
- Analyzing the results using the built-in plotting and visualization facilities, including the XY Plot of a scalar variation

Harmonic Balance: Single Stage Periodic Flow

The tutorial illustrates the steps necessary to modify the single axial compressor case so that the effect of a periodic wake can be simulated using the Harmonic Balance method.



As this case uses the model and simulation resulting from the Single Row Tutorial, it is necessary to complete that tutorial before continuing.

Contents:

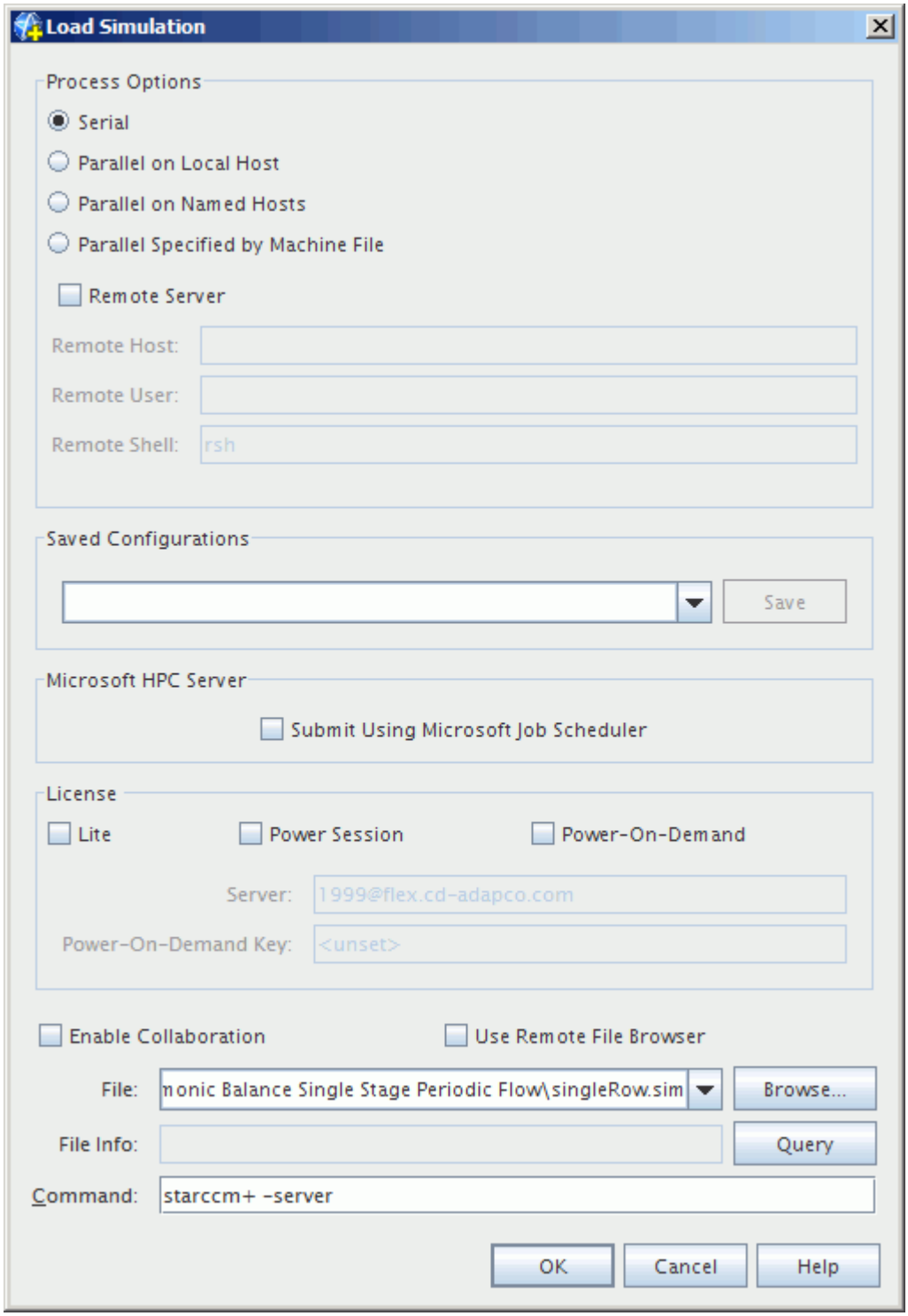
Loading the Single Row Tutorial Simulation
Selecting the Physical Models for Harmonic Balance
Defining Harmonic Balance Parameters
Defining Boundary Conditions
Setting Solver and Stopping Criteria Parameters
Visualizing the Solution
Running the Simulation
Analyzing Results
Summary

Loading the Single Row Tutorial Simulation

Begin with loading the previously created single row simulation.

Note: To start this tutorial, you can open one of two versions of the simulation file: the one that you created locally (if applicable), or a previously solved simulation file that is available online. The following instructions apply to a locally created simulation file.

1. Create a subdirectory for the tutorial called `harmonicBalanceTut`.
2. In this directory, start up STAR-CCM+ in a manner that is appropriate to your working environment, and select **File** > **Load Simulation** from the menu bar.
3. In the *Load Simulation* dialog, use the **Browse** button to navigate to the location of the previously saved `singleRow.sim` file, select it, and click **Open**.
4. The *Load Simulation* dialog shows the path to the selected file in the *File* textbox.
5. Click **OK** to launch a server process and load the selected simulation.



Save the simulation with a different file name before changing anything:

6. Select **File** > **Save As...** from the menu bar.
7. In the *Save* dialog, navigate to the `harmonicBalanceTut` directory and enter `harmonicBalance` in the *File Name* textbox.
8. Click **Save**.

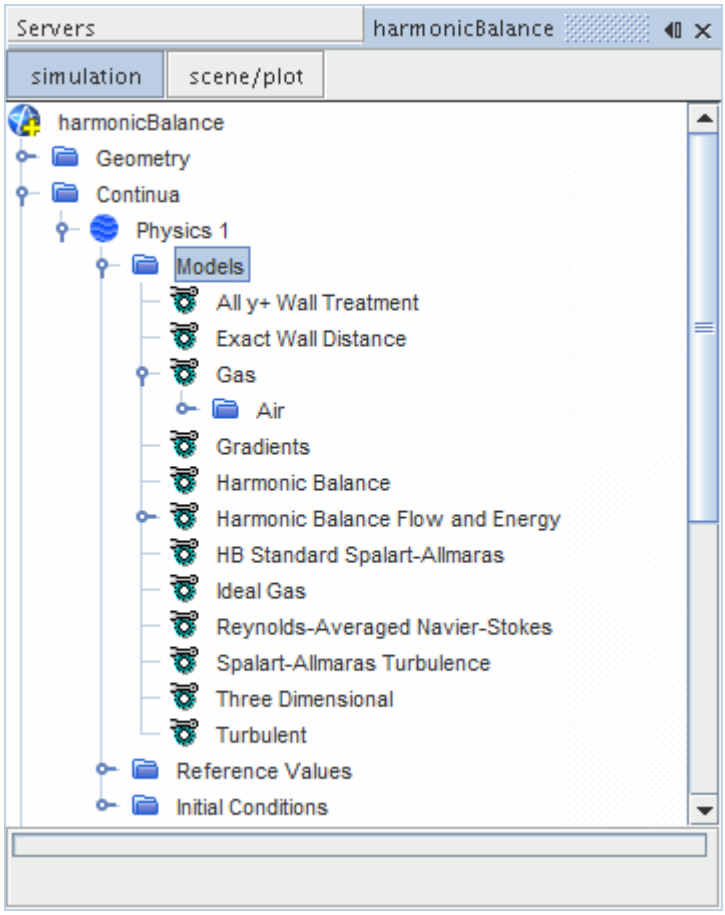
Selecting the Physical Models for Harmonic Balance

As physical models were previously selected for the single row tutorial, it is necessary to deactivate a number of the models so that the required options are made available.

1. For the physics continuum **Continua** > **Physics 1**, deactivate the following models from the **Enabled Models** list:
   - **Transition Boundary Distance**
   - **Turbulence Suppression**
   - **All y+ Wall Treatment**
   - **SST (Menter) K-Omega**
   - **K-Omega Turbulence**
   - **Reynolds-Averaged Navier-Stokes**
   - **Turbulent**
   - **Coupled Energy**
   - **Ideal Gas**
   - **Coupled Flow**
   - **Steady**
2. Activate **Auto-select recommended models**.
3. Select the following models, in order:

Group Box	Model
Time	**Harmonic Balance**
Viscous Regime	**Turbulent**
Reynolds-Averaged Turbulence	**Spalart-Allmaras Turbulence** **HB Standard Spalart-Allmaras** (Selected automatically) **All y+ Wall Treatment** (Selected automatically) **Exact Wall Distance** (Selected automatically)

4. Click **Close**.
5. To see the selected models, expand the **Continua** > **Physics 1** > **Models** node.



Defining Harmonic Balance Parameters

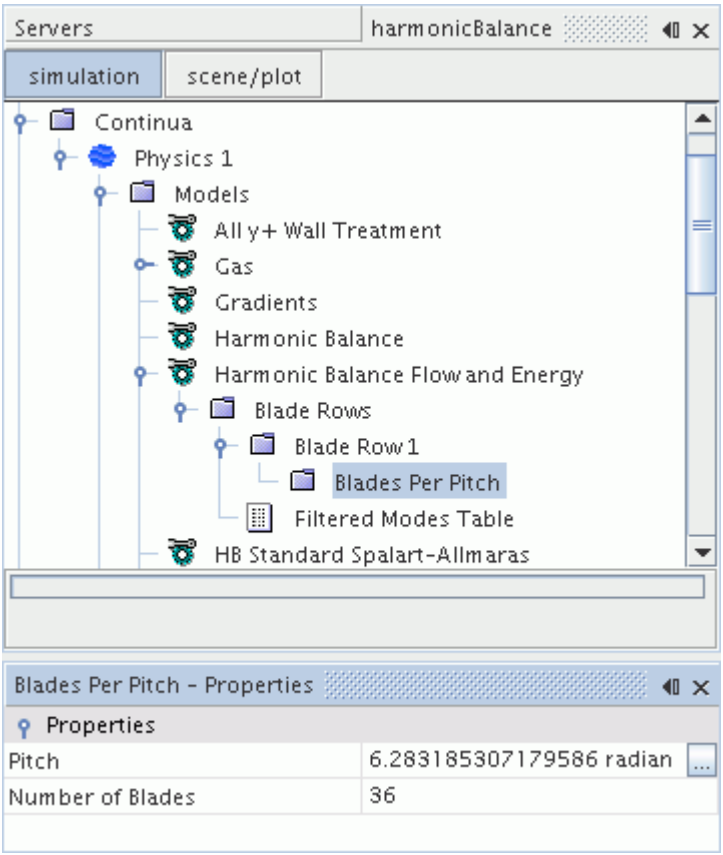
The Harmonic Balance Flow and Energy model requires definition of blade row parameters.

In this example, a single blade row that consists of 36 blades is required. The number of modes is set to 3, which provides a sufficiently accurate solution and keeps the demand on system resources relatively low.

To specify the required parameters:

1. Expand the **Continua** > **Physics 1** > **Models** > **Harmonic Balance Flow and Energy** manager. right-click the **Blade Rows** node, and select **New**.

   A new node named **Blade Row 1** appears under the **Blade Rows** node and is selected automatically.
2. Select the **Blade Row 1** node and set *Modes* to `3`.
3. Select the **Blade Row 1** > **Blades Per Pitch node** and set *Number of Blades* to `36` .
4. Leave the *Pitch* property at its default setting of radians.

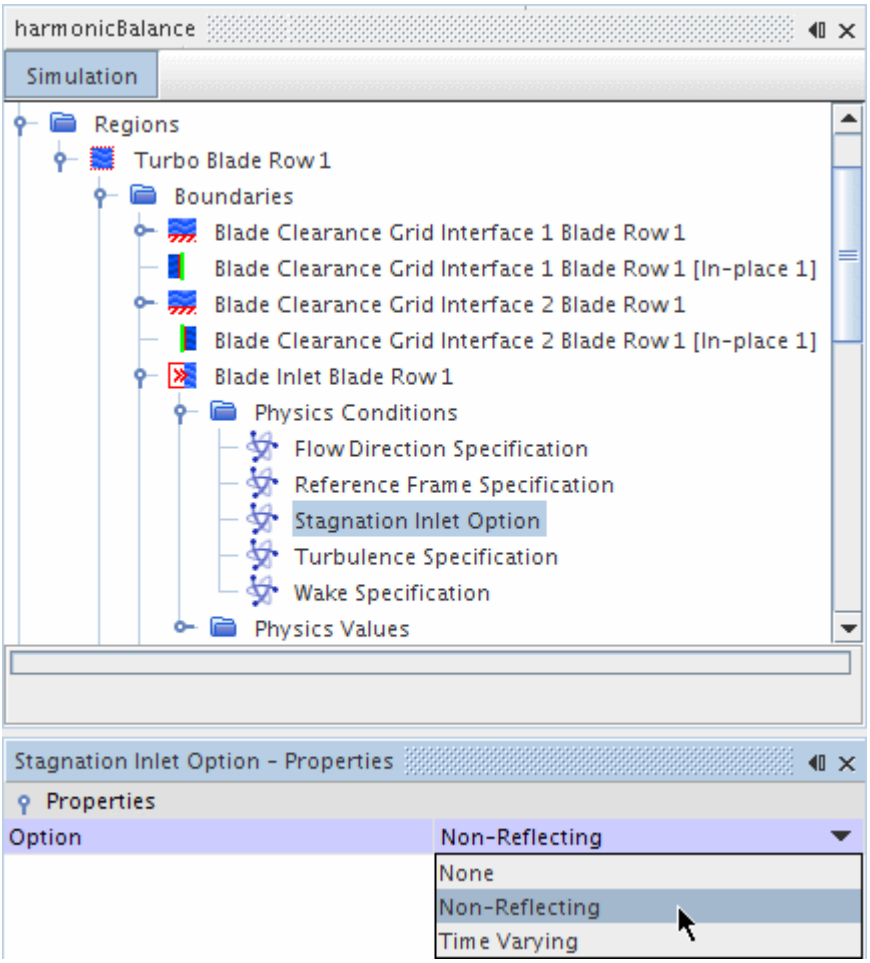


Defining Boundary Conditions

The Non-Reflecting option is activated for both the inlet and outlet boundaries.

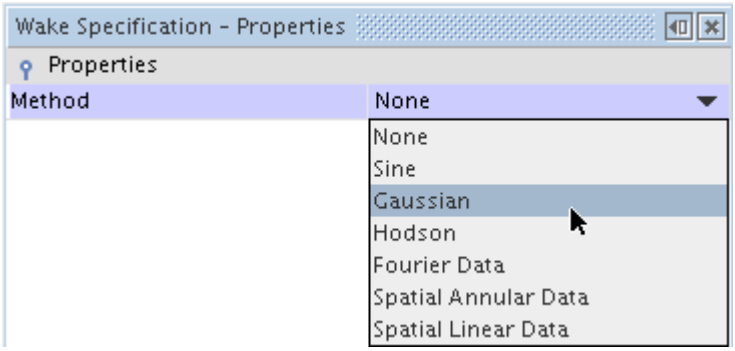
The Wake specification at the inlet is also set to define the periodic perturbation. This consists of defining the characteristics of the wake profile and the upstream blade row. In this example, a Gaussian wake profile is used, with a width of 0.15 and a defect of 2. The upstream blade row has 48 blades and is stationary (0.0 rads/s), as it represents a stator.

1. Expand the **Regions** > **Turbo Blade Row 1** > **Boundaries** > **Blade Inlet Blade Row 1** > **Physics Conditions** > **Stagnation Inlet Option** node and set *Option* to **Non-Reflecting**.



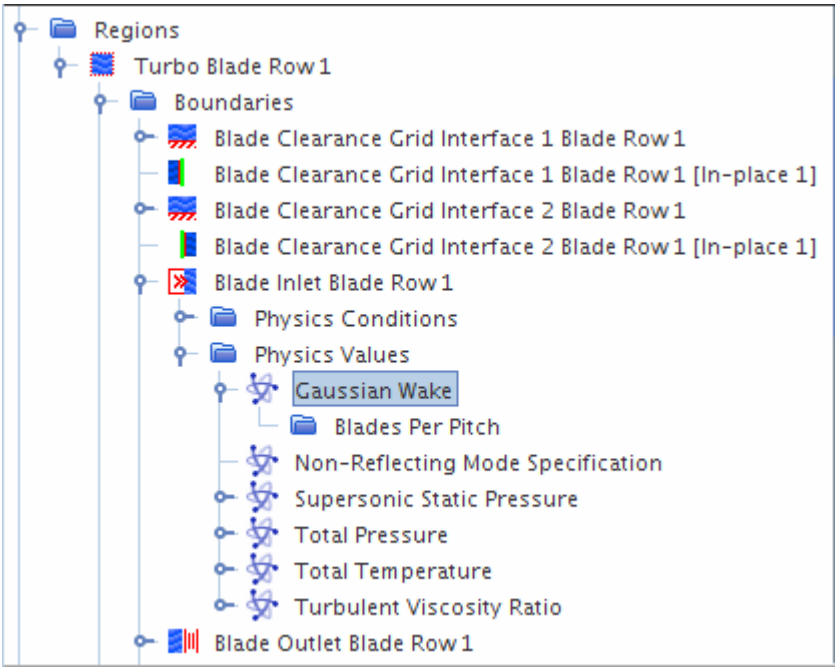
An additional node, **Wake Specification**, is added to the **Physics Conditions** manager for this boundary when the Non-Reflecting option is activated.

2. Select the **Wake Specification** node and set *Method* to **Gaussian**.

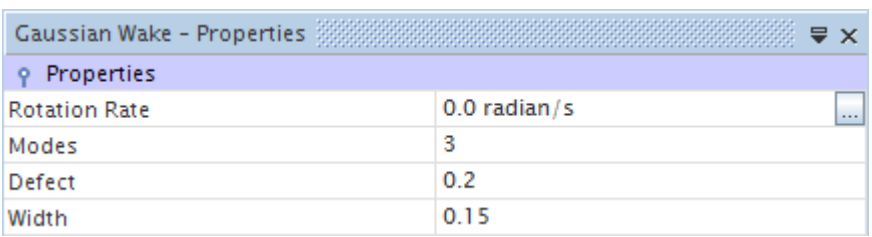


In response to these changes, there are now two extra nodes in the **Physics Values** folder that must be set.

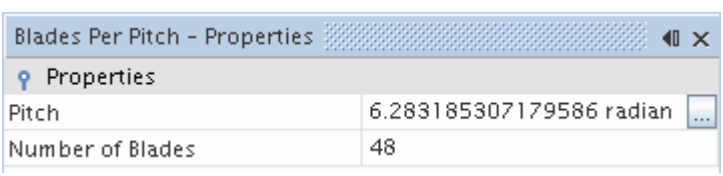
3. Select the **Physics Values** > **Gaussian Wake** node,



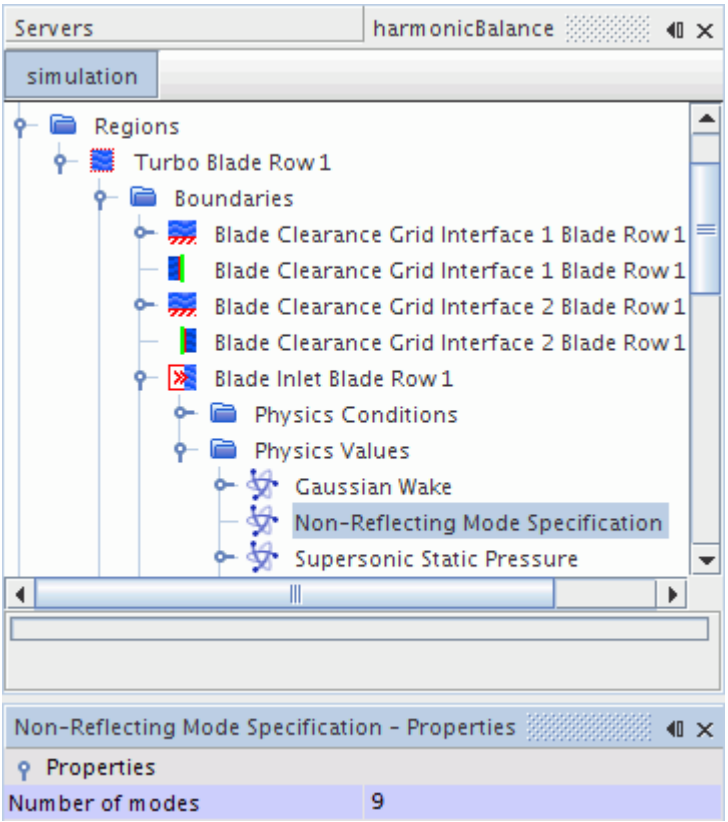
set *Rotation Rate* to `0.0 radian/s`, and set *Modes* to `3`.



4. Select the **Physics Values** > **Gaussian Wake** > **Blades Per Pitch** node and set *Number of Blades* to `48`.

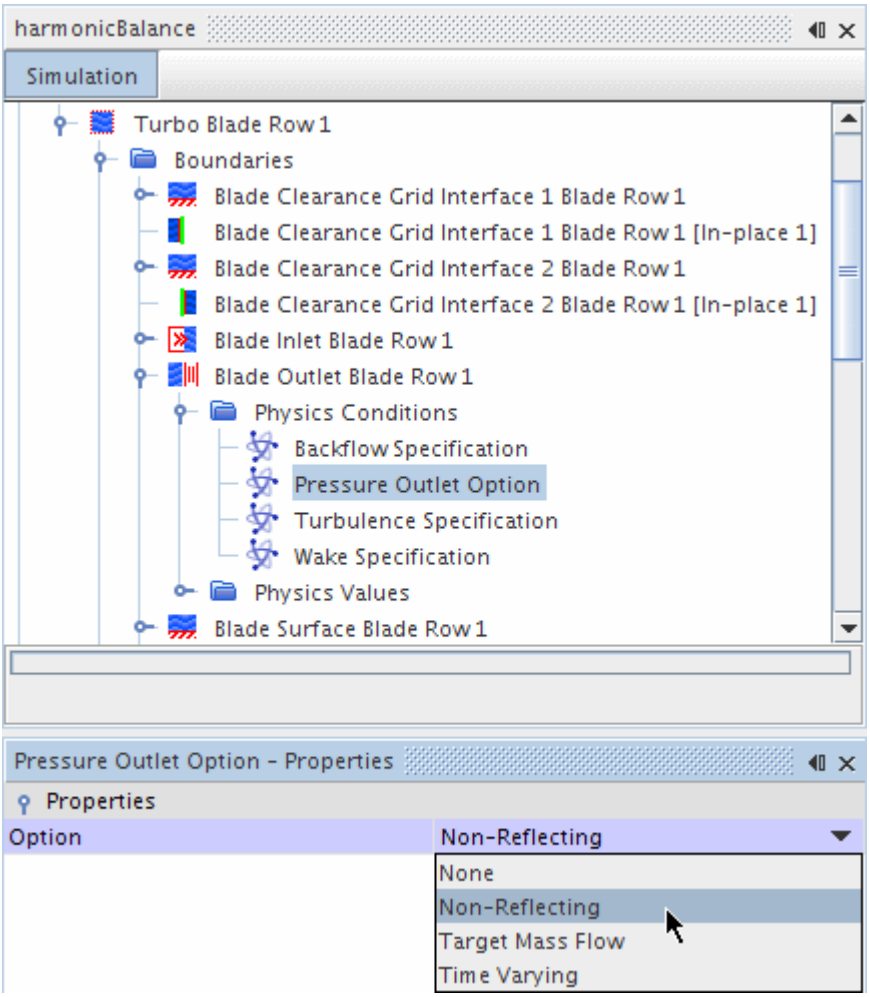


5. Select the **Physics Values** > **Non-Reflecting Mode Specification** node and set *Number of modes* to `9`.

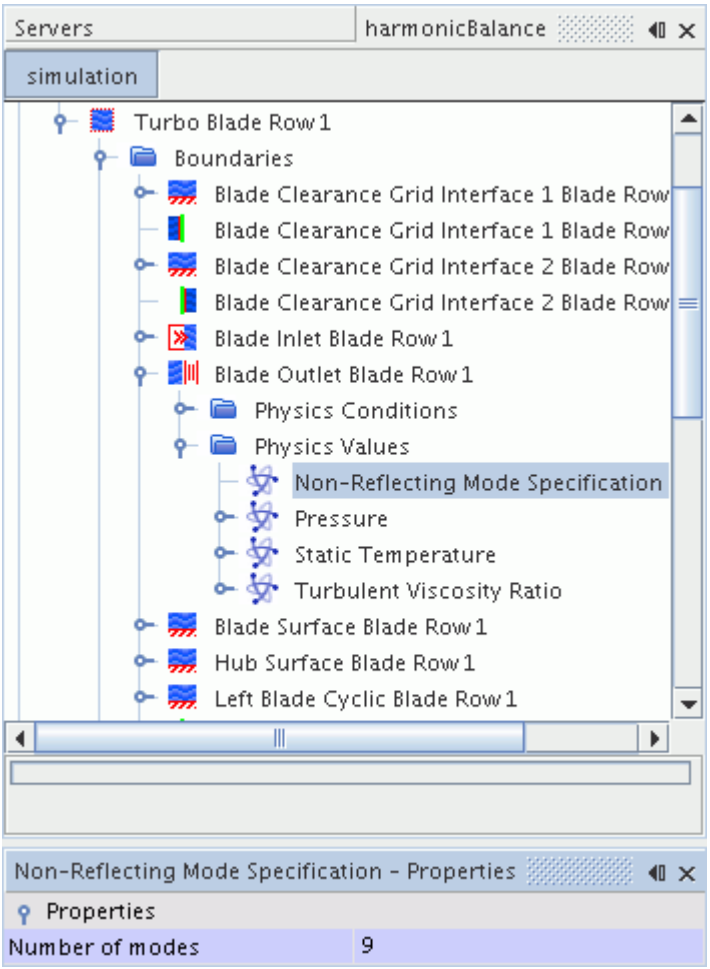


Similarly, modify the outlet boundary without the wake specification.

6. Select **Regions** > **Turbo Blade Row 1** > **Boundaries** > **Blade Outlet Blade Row1** > **Physics Conditions** > **Pressure Outlet Option** and set *Option* to **Non-Reflecting**.

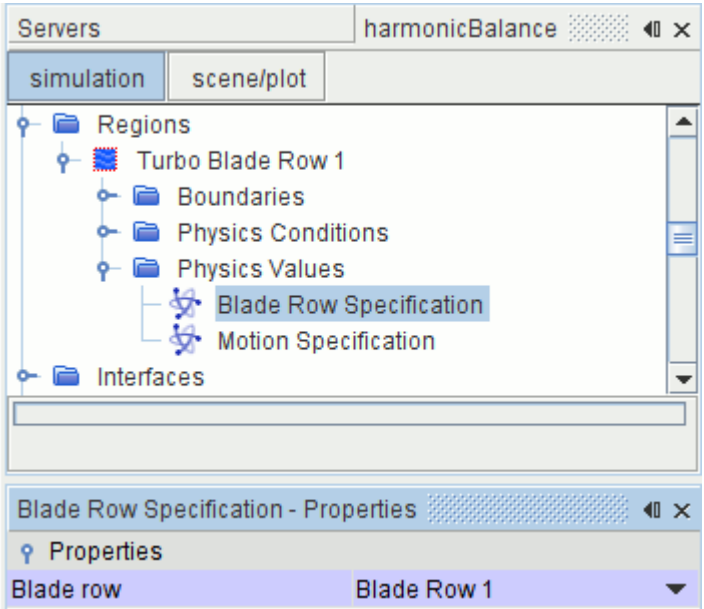


7. For the same boundary, select the **Physics Values** > **Non-Reflecting Mode Specification** node and set *Number of modes* to 9



Define the blade row specification:

8. Select the **Regions** > **Turbo Blade Row 1** > **Physics Values** > **Blade Row Specification** node and set *Blade Row* to **Blade Row 1**.

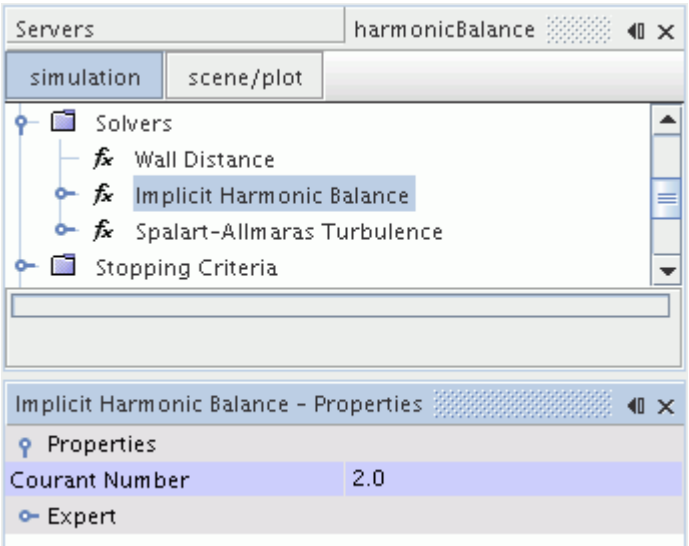


9. Save the simulation.

Setting Solver and Stopping Criteria Parameters

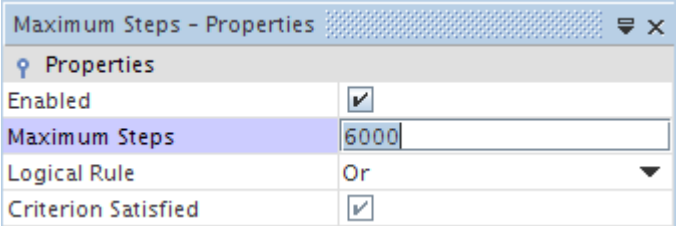
Set solver parameters:

1. Expand the **Solvers** > **Implicit Harmonic Balance** manager and set *Courant Number* to `2.0`.



To make sure that sufficient steps have been allowed, particularly as the simulation continues from a previously converged solution, modify the stopping criteria:

2. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to `6000`.



3. Save the simulation.

Visualizing the Solution

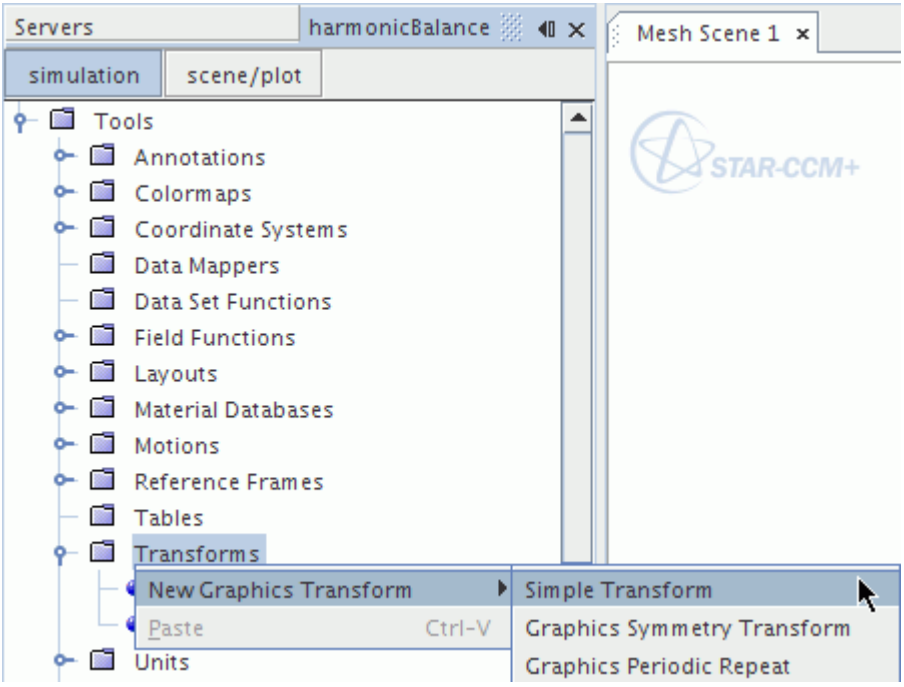
To show the temperature profile on a plane cutting through the center of the mesh, create a scalar scene.

This scene includes multiple transformed sections of the mesh to show the effect of phase lag on the solution.

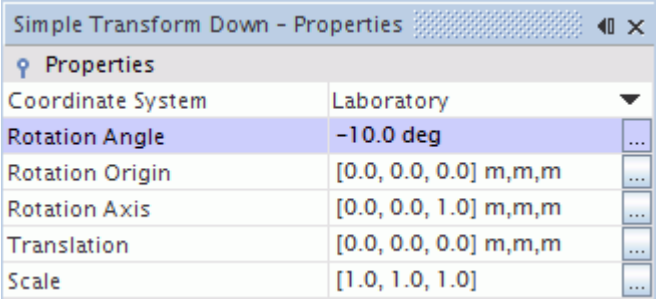
1. Right-click the **Scenes** manager and select **New Scene** > **Scalar**.

Before proceeding further, define two new transforms that are used to create multiple mesh sections in the scene.

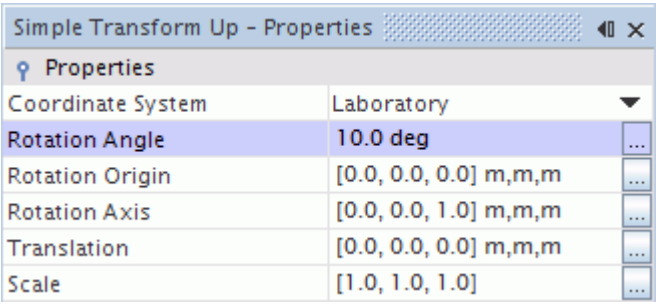
2. Right-click the **Tools** > **Transforms** manager and select **New Graphics Transform** > **Simple Transform**.



3. Rename the **Simple Transform 1** node to `Simple Transform Down`.
4. Select the **Simple Transform Down** node and set *Rotation Angle* to `-10.0 deg`.

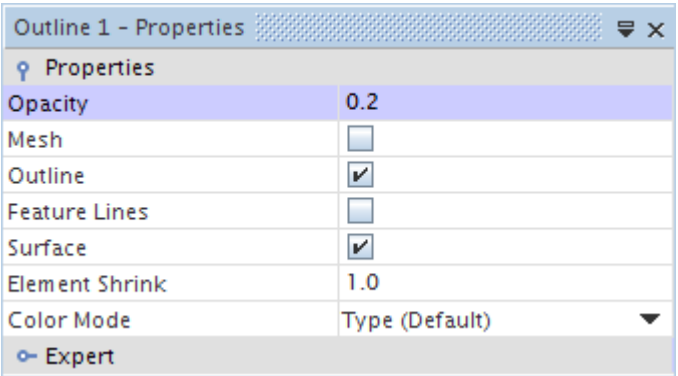


5. Right-click the **Transforms** node and select **New Graphics Transform** > **Simple Transform** as before.
6. Rename this node to `Simple Transform Up`.
7. Select the **Simple Transform Up** node and change the *Rotation Angle* to `10.0 deg`.

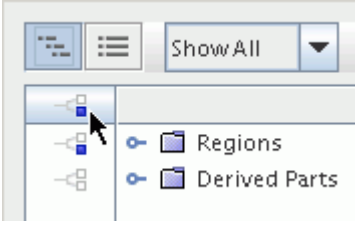


Switch back to creating the Scalar Scene.

8. Select the **Scenes** > **Scalar Scene 2** node and click the **scene/plot** button above the object tree.
9. Select the **Scalar Scene 2** > **Displayers** > **Outline 1** node, activate the *Surface* property, and set the *Opacity* to `0.2`.

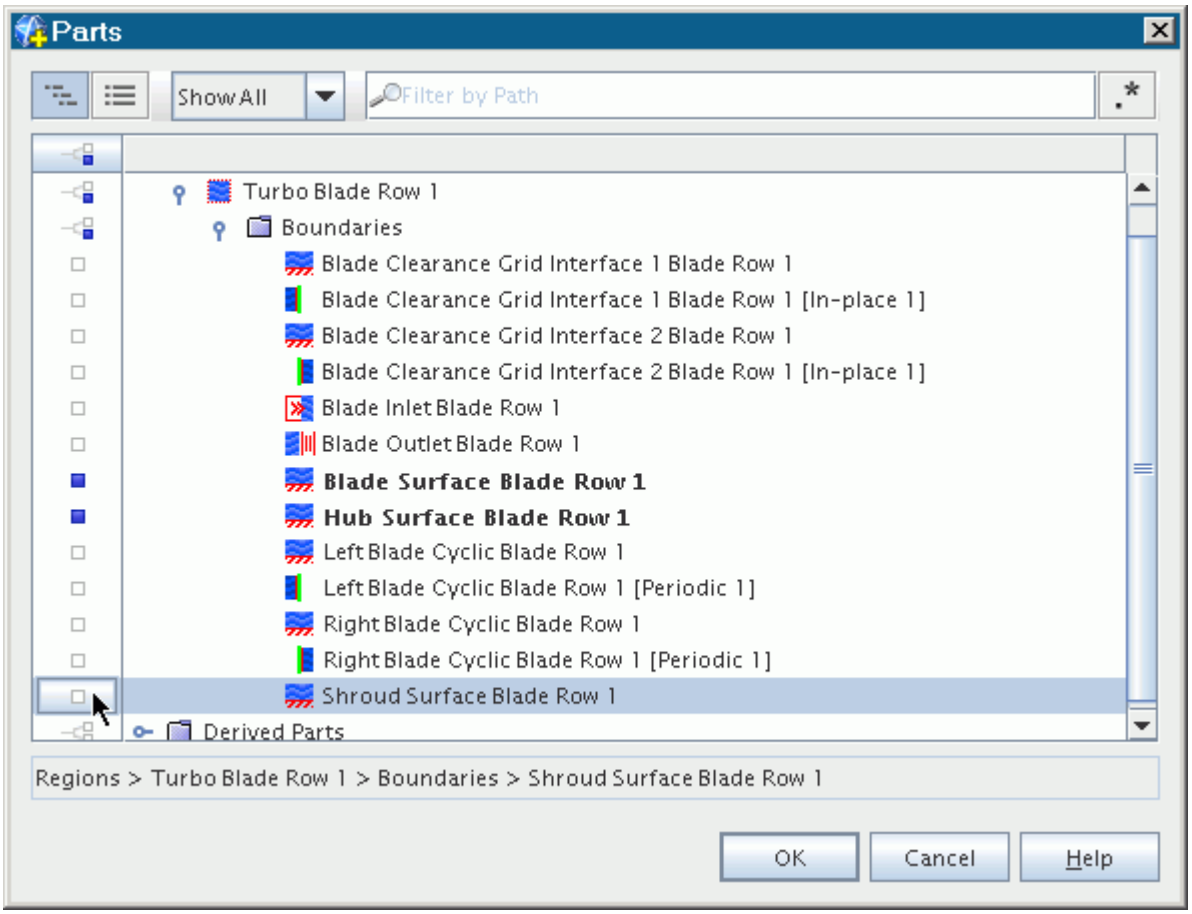


10. Select the **Parts** node and edit the *Parts* property. In the *Parts* dialog:
    a) To clear all selections, double-click the **Column Selection** button in the upper left portion of the dialog.



b) Expand the **Regions** > **Turbo Blade Row 1** manager.

c) To select **Blade Surface Blade Row 1**, **Hub Surface Blade Row 1**, and **Shroud Surface Blade Row 1**, click once on the selection button to the left of each of them, as shown in the following screenshot.



d) Click **OK**.

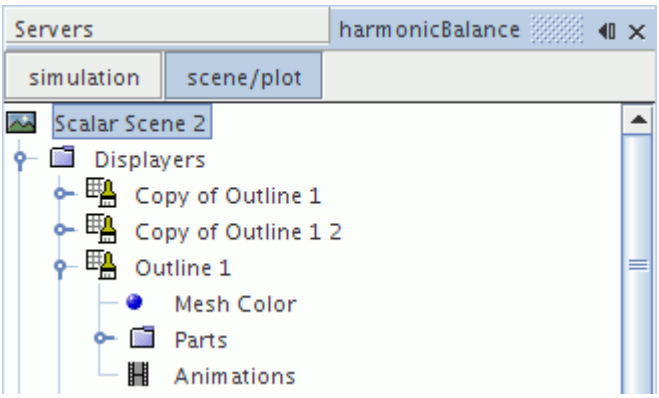
Copy the **Outline 1** node twice: once to provide a transformed outline rotated 10 deg up from the modeled blade, and a second time to provide a transformed outline rotated 10 deg down.

11. Right-click the **Outline 1** node and select **Copy**.
12. Right-click the **Scalar Scene 2** node at the top of the panel and select **Paste**.

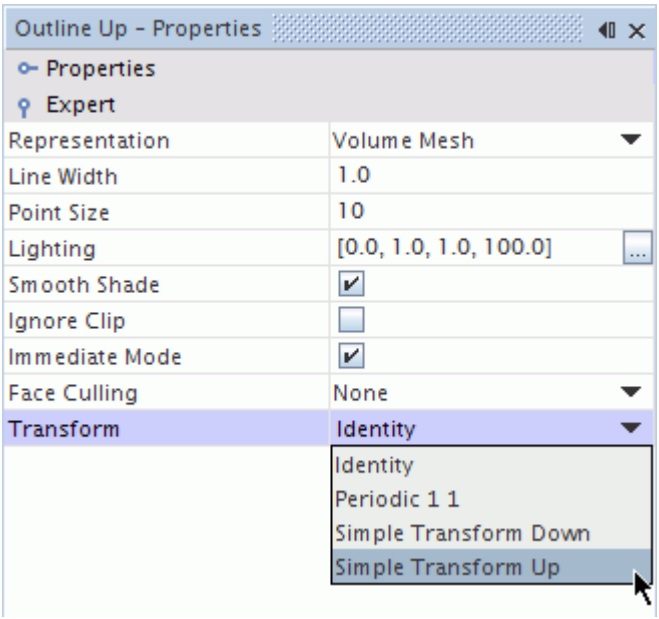
This operation pastes the copied node within the **Displayers** folder with the name, **Copy of Outline 1**.

13. To add a second copy of **Outline 1** to the **Displayers** folder, repeat this copy-and-paste operation.

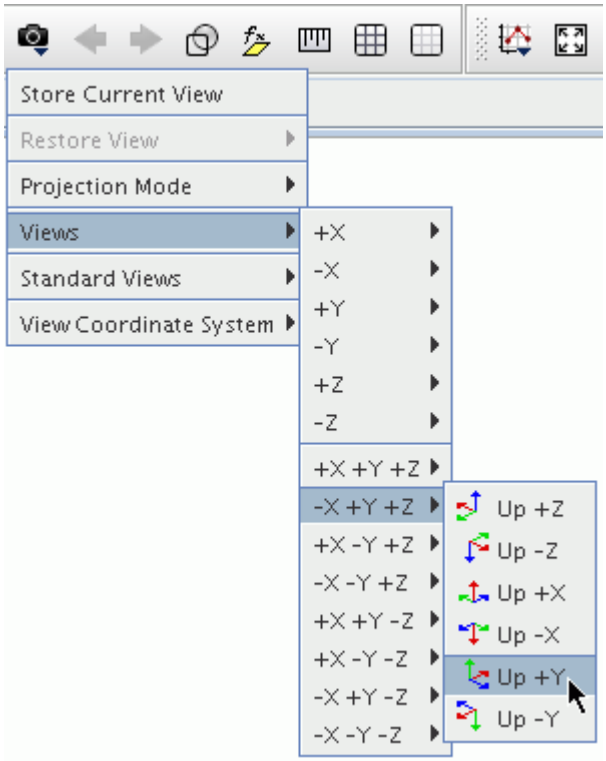
The second copy is named **Copy of Outline 1 2**.



14. Rename the **Copy of Outline 1** node to `Outline Up`.
15. Select the **Outline Up** node and set *Transform* to **Simple Transform Up**.

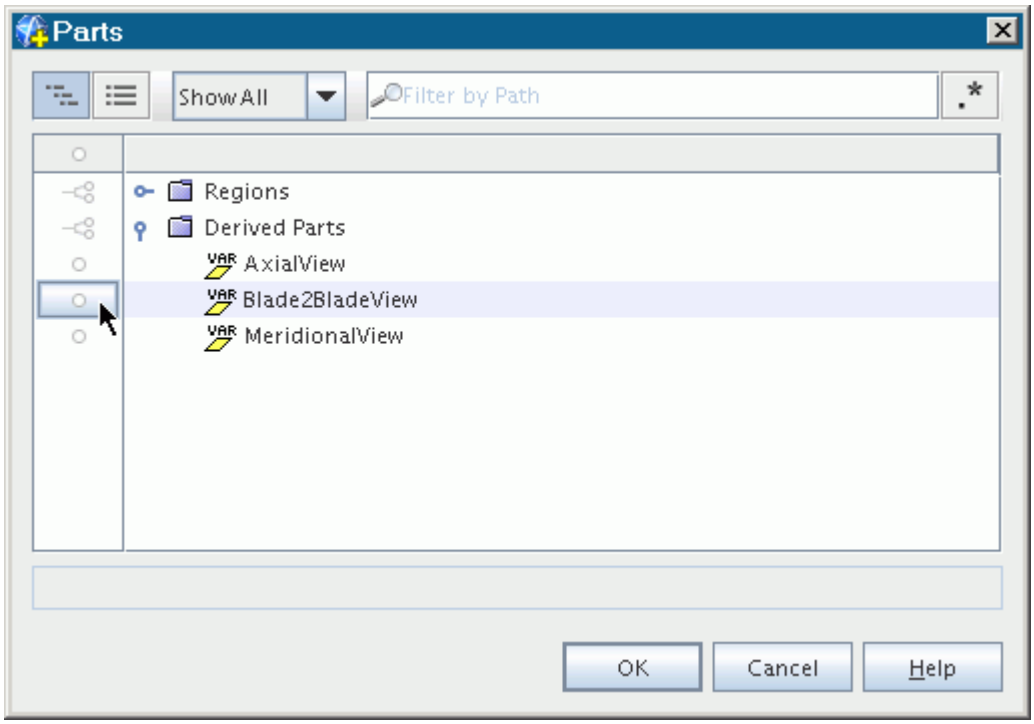


16. Rename the **Copy of Outline 1 2** node to `Outline Down`.
17. Select the **Outline Down** node and set the *Transform* to **Simple Transform Down**.
18. Click the (**Save-Restore-Select views**) button in the toolbar and select **Views** > **-X+Y+Z** > **Up +Y**.



The scalar displayers within the scene are set up in a similar manner to the outline displayers.

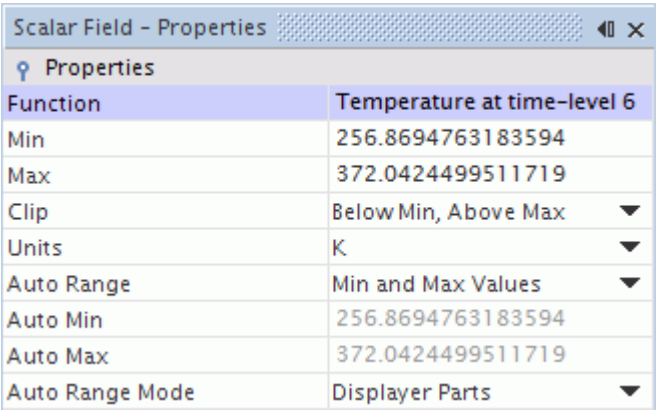
19. Select the **Scalar 1** > **Parts** manager and edit the *Parts* property.
20. In the *Parts* dialog, expand the **Derived Parts** manager and select **Blade2BladeView**.



21. Click **OK**.

22. To make sure that the Harmonic Balance Functions are activated, click the (**Initialize Solution**) button.

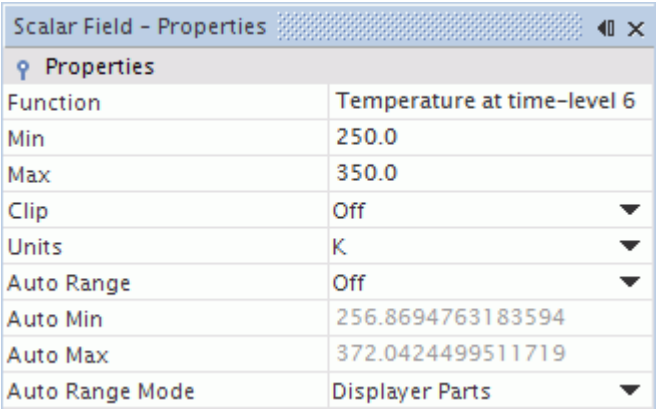
23. Select the **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Scalar Field** node and set *Function* to **Harmonic Balance** > **Temperature** > **Time Level** > **006** > **Temperature at time-level 6**.



24. Also for the **Scalar 1** > **Scalar Field** node, set the following properties:

Property	Setting
Min	`250.0`
Max	`350.0`
Clip	**Off**
Auto Range	**Off**

The final set of properties is as shown below.



Copy the **Scalar 1** node twice for either side of the central position:

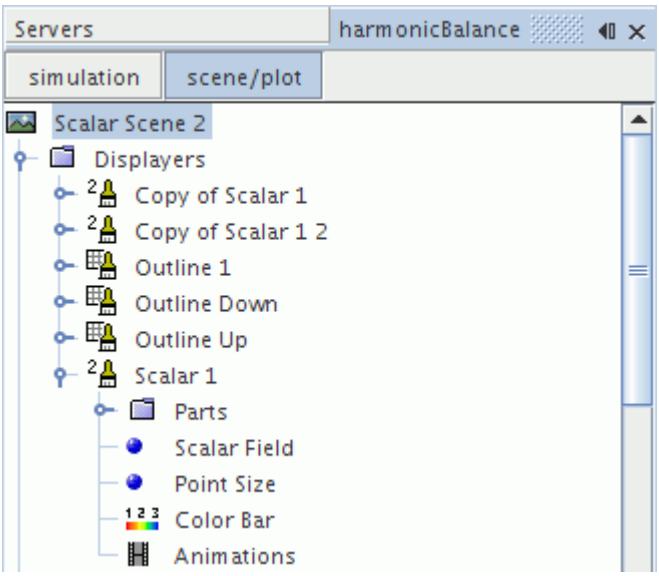
25. Right-click the **Scalar 1** node and choose **Copy**.

26. Right-click the **Scalar Scene 2** node (at the top of the subtree) and choose **Paste**.

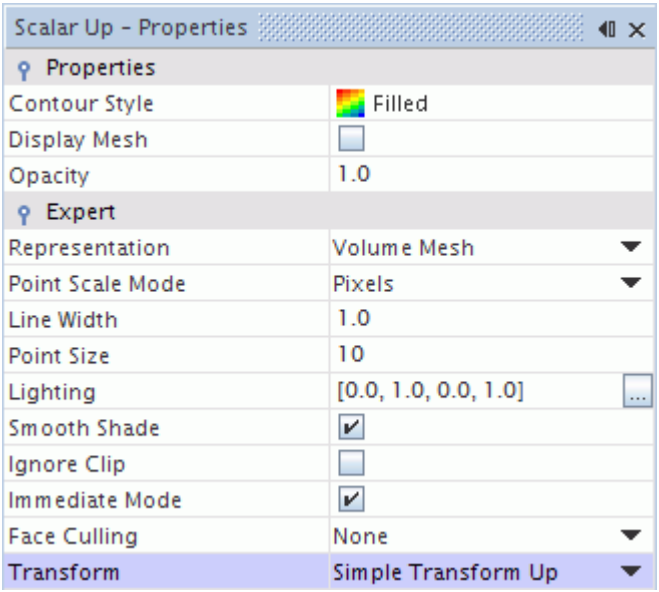
A copy of the **Scalar 1** node is created with the name **Copy of Scalar 1**.

27. To add a second copy of **Scalar 1** to the **Displayers** folder, repeat this copy-and-paste operation.

    The second copy is named **Copy of Scalar 1 2**.



28. Rename the **Copy of Scalar 1** node to `Scalar Up`.
29. Select the **Scalar Up** node and set *Transform* to **Simple Transform Up**.



30. Expand the **Scalar Up** node, select the **Scalar Field** node, and set *Function* to **Harmonic Balance** > **Temperature** > **Time-Level** > **006** > **Temperature Level 6 Phase Lagged Up**.

31. Also set the same values for the *Min* and *Max* properties that were used previously, that is `250.0` and `350.0` respectively. Make sure that both the *Auto Range* and *Clip* properties are set to **Off**.
32. Select the **Scalar Up** > **Color Bar** node and deactivate *Visible*.

    The color bar is removed from the scene for this particular displayer.

A similar process is required for the final scalar displayer.

33. Rename the **Copy of Scalar 1 2** node to `Scalar Down`.
34. Select the **Scalar Down** node and set *Transform* to **Simple Transform Down**.
35. Expand the **Scalar Down** node, select the **Scalar Field** node, and set *Function* to **Harmonic Balance** > **Temperature** > **Time Level** > **006** > **Temperature Level 6 Phase Lagged Down**.
36. Also set the *Min* and *Max* properties to `250.0` and `350.0` respectively. Make sure that both the *Auto Range* and *Clip* properties are set to **Off**.
37. Select the **Scalar Down** > **Color Bar** node and deactivate *Visible*.

The steps for **Scalar Scene 2** are complete.

38. To switch back to the main object tree, click the **simulation** button.
39. Save the simulation.

Running the Simulation

Click **(Run)** to start the solver.

The analysis now starts from the previous solution that was obtained in the Single Row tutorial. The residual values in the Residuals plot immediately increase due to the change in the model properties. However, these residuals decrease at a faster rate in the continued solution than they did in the original Single Row tutorial.

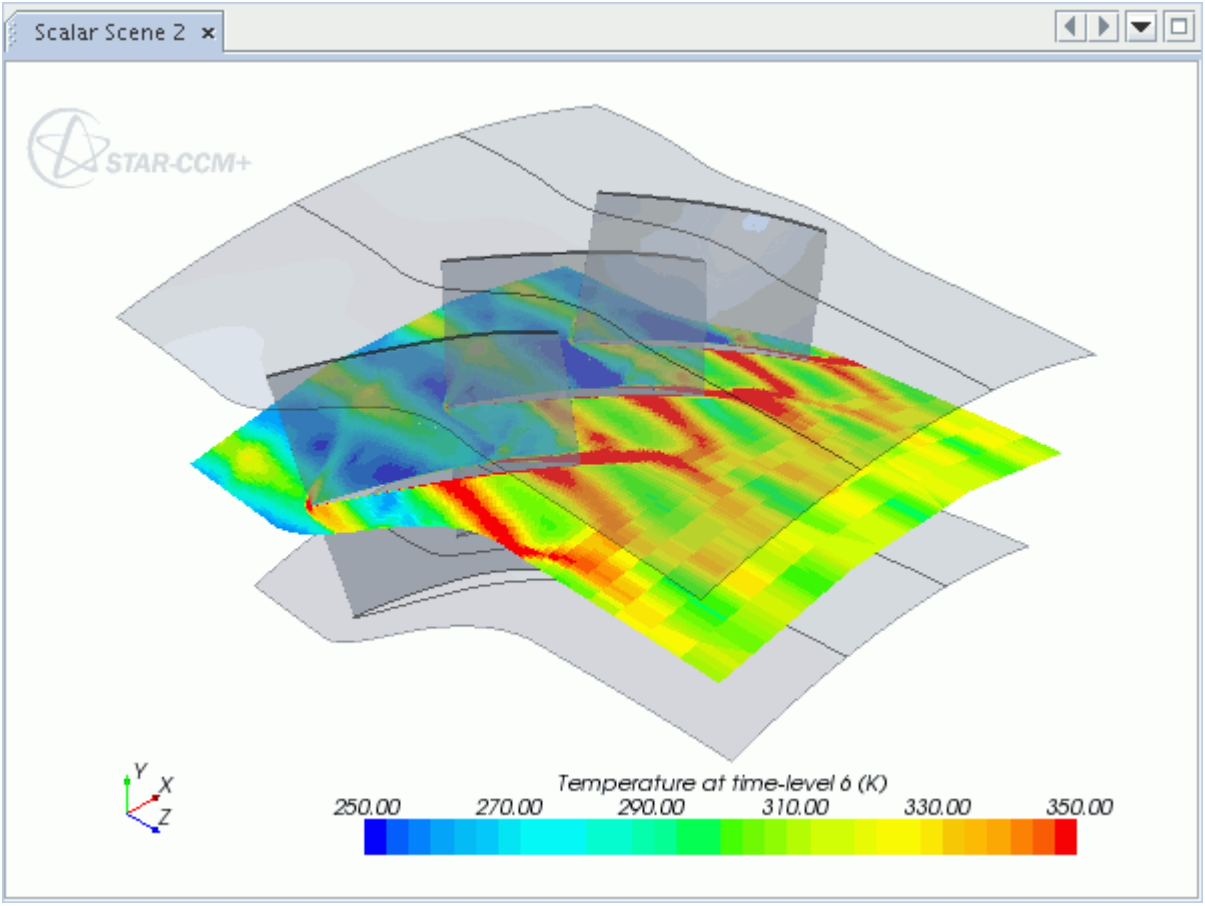
Note: The run time for the Harmonic Balance method is longer than it was for the original tutorial. This is because the Harmonic Balance method obtains solutions for several time levels simultaneously.

Analyzing Results

Examine temperature and pressure on the blade surface.

Temperature

1. The temperature at time level 6, shown in **Scalar Scene 2**, is as follows:



The effect of the wake specification at the inlet boundary can be clearly observed on the temperature plot, along with the effect of the phase angle.

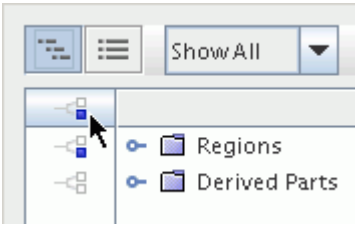
Obtaining XY Plot of Pressure on the Blade Surface

To plot the pressure profile at different time levels on the surface of the blade, it is first necessary to create a derived part that defines the location of the profile.

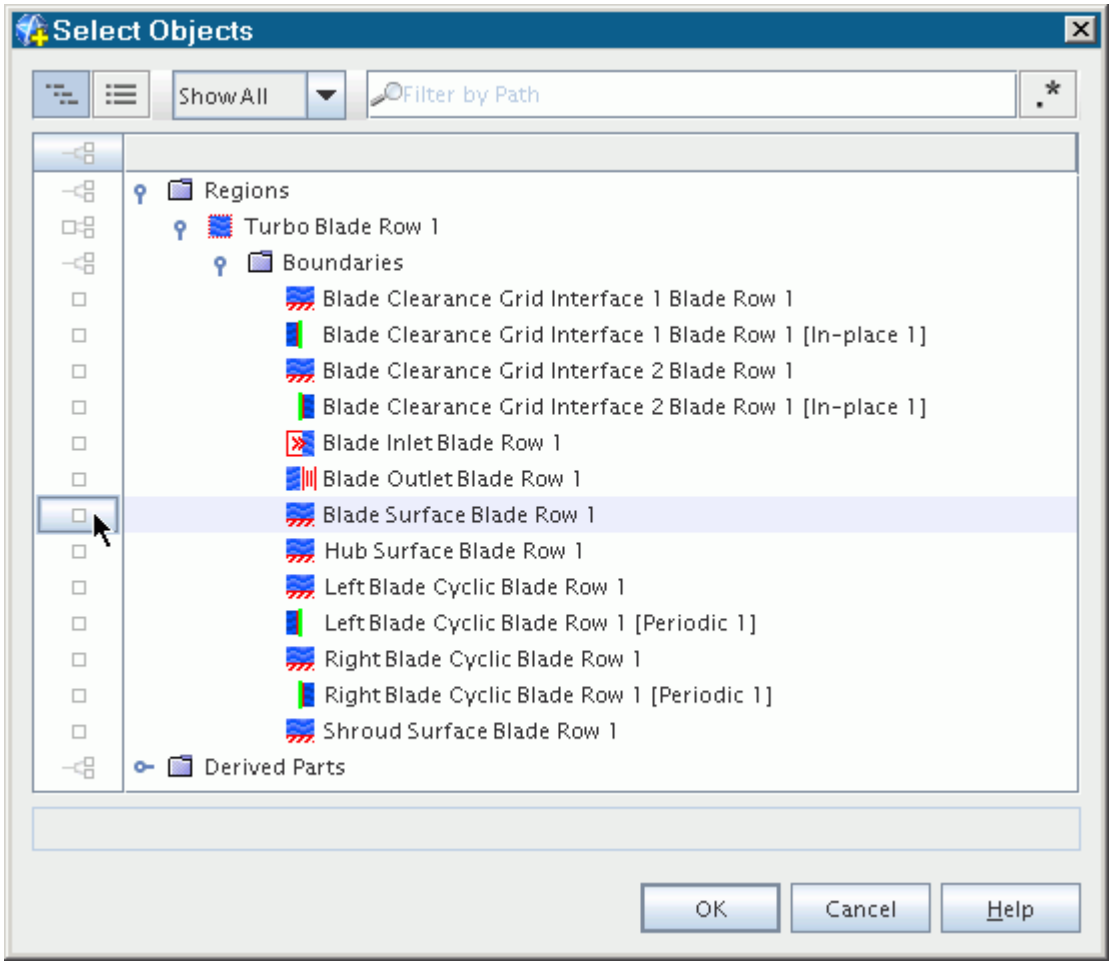
2. Right-click **Scenes** > **Mesh Scene 1** and select **Open**.
3. Right-click the **Derived Parts** manager and select **New Part** > **Isosurface**.

   The *Create Isosurface* dialog opens.
4. Click the menu in **Input Parts** box.

   The *Select Objects* dialog appears.
5. To clear all selections, double-click the **Column Selection** button in the upper left portion of the dialog.

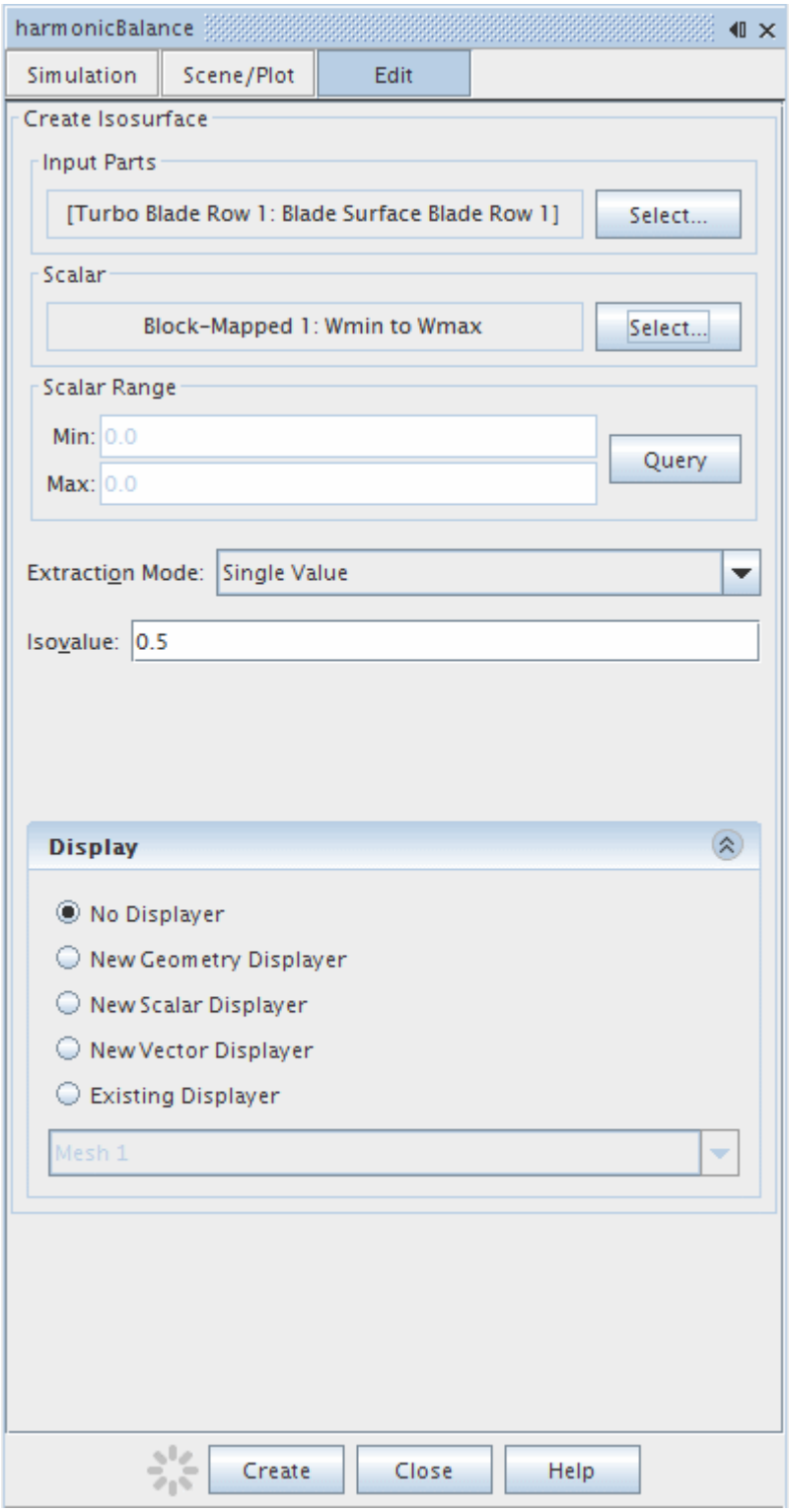


6. Expand the **Regions** > **Turbo Blade Row 1** manager.
7. Select **Blade Surface Blade Row 1**.



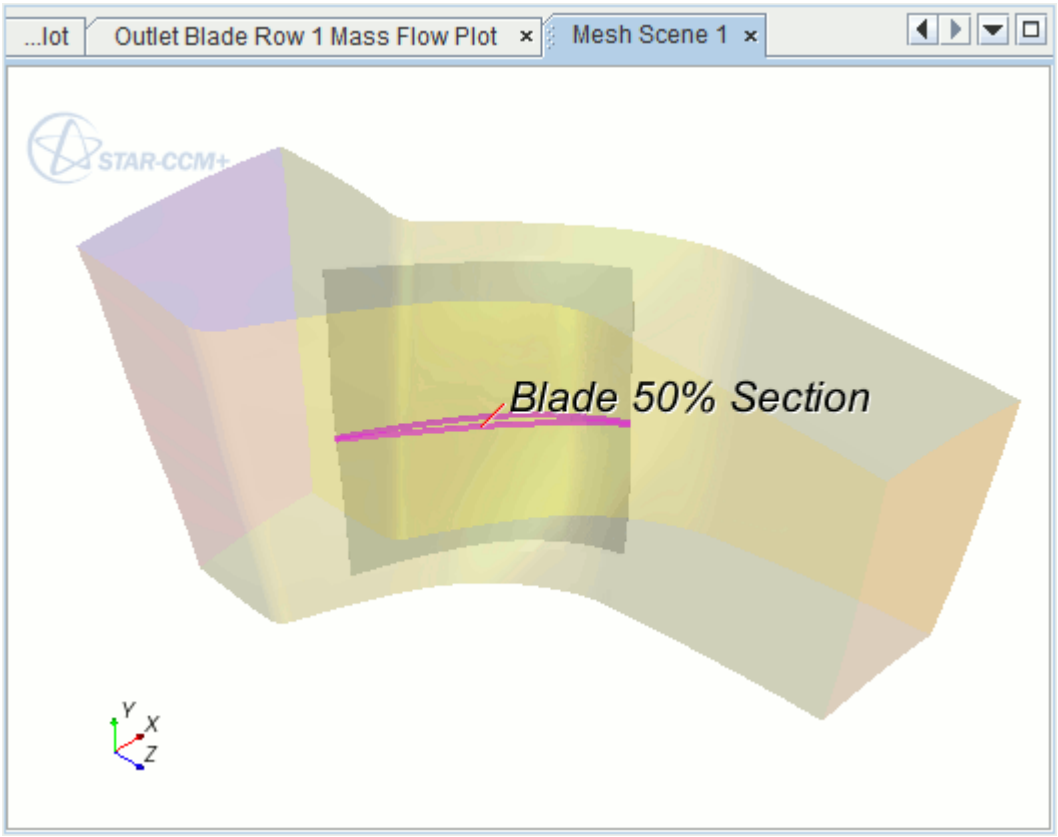
8. Click **OK**.
9. Click in the **Scalar** box and select **Block-Mapped 1: Vmin to Vmax**.
10. Enter `0.5` in the *iso value* text box.
11. Select **No Displayer** in the **Display** section.

    The final panel is as shown below.



12. Click **Create** and then **Close**.
13. Rename the **Derived Parts** > **isosurface** node to `Blade 50% Section`.
14. Click (**Hide All Meshes**).
15. Click **(Make Scene Transparent)**.

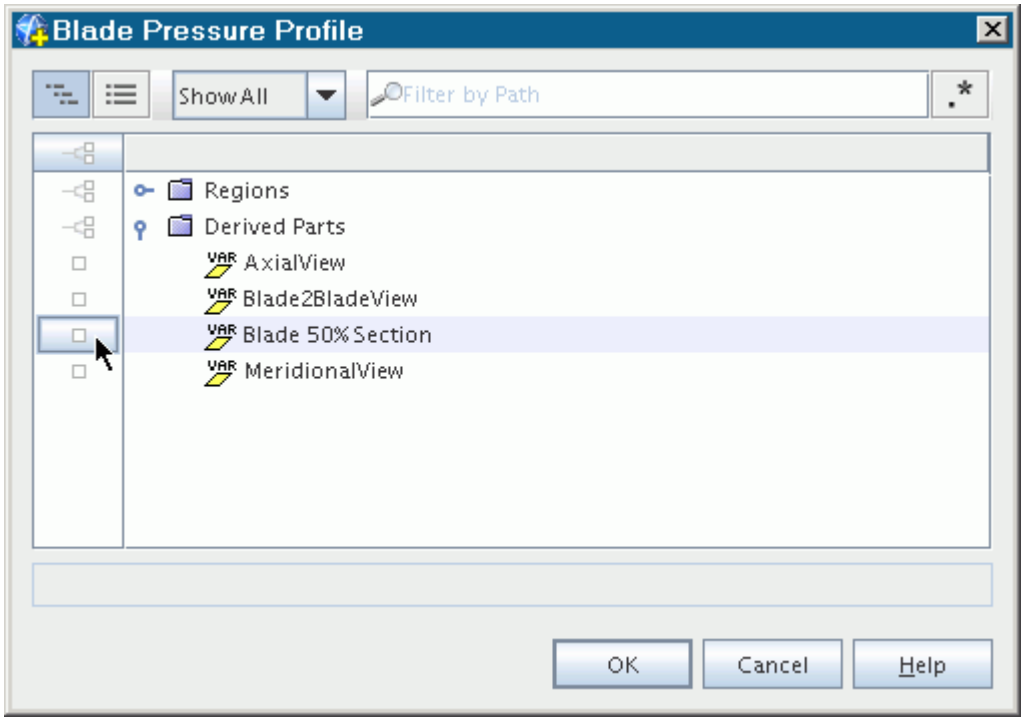
The derived part is shown in the following image for illustration.



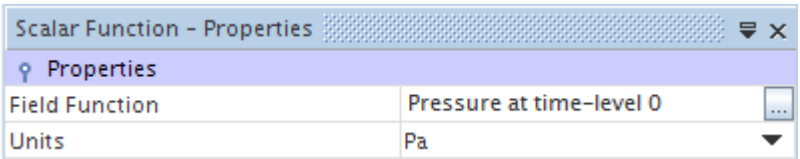
This derived part is now used in an XY plot.

16. Right-click the **Plots** manager and select **New Plot** > **X-Y Plot**.

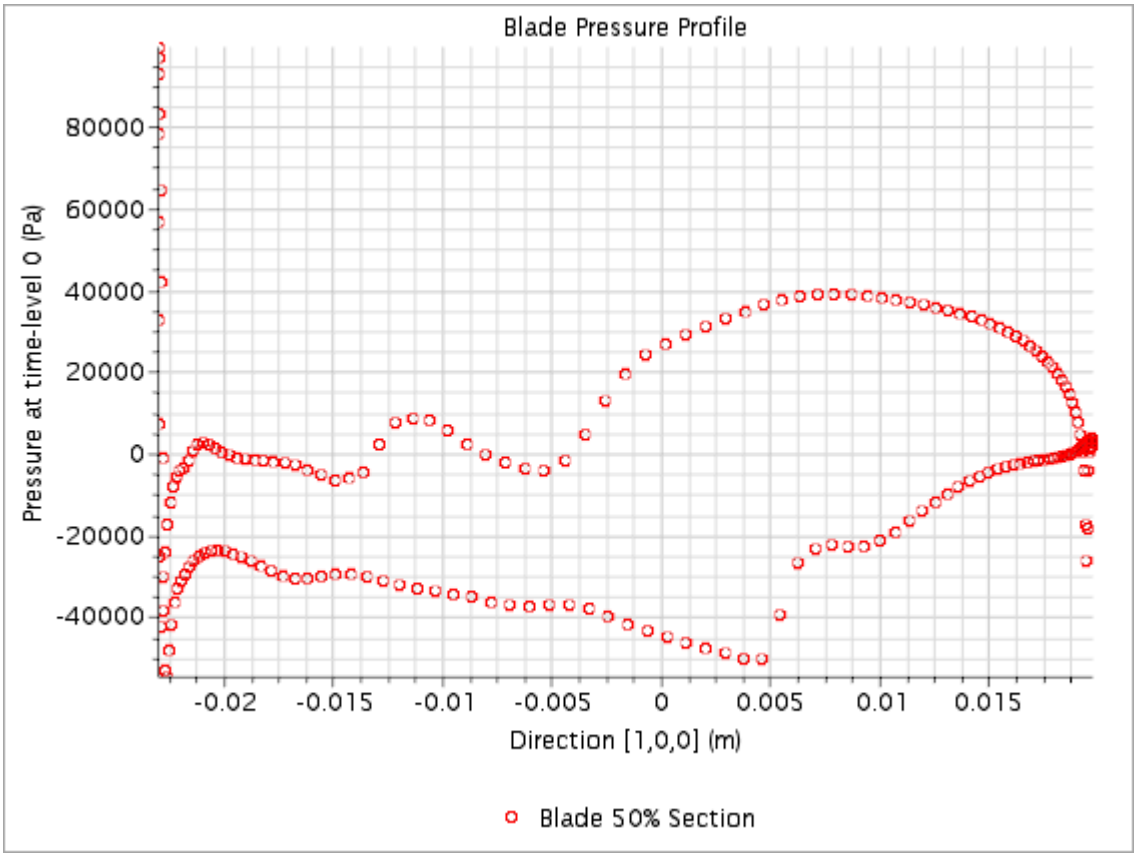
    A new node, **XY Plot 1**, appears in the tree below.
17. Rename the **XY Plot 1** node to **Blade Pressure Profile**.
18. Select the **Blade Pressure Profile** node and edit the *Parts* property.
19. In the *Blade Pressure Profile* dialog, expand the **Derived Parts** manager and select **Blade 50% Section**.



20. Click **OK**.
21. Select the **Plots** > **Blade Pressure Profile** > **Y Types** > **Y Type 1** > **Scalar Function** node and set *Field Function* to **Harmonic Balance** > **Pressure** > **Time-level** > **000** > **Pressure at time-level 0**.



The profile plot appears as shown in the **Graphics** window.



Summary

This tutorial demonstrated the following features:

- Resuming a previously converged simulation and converting it to run with the Harmonic Balance method
- Setting the necessary boundary conditions including selection of an unsteady wake
- Preparing a scalar scene with transformed parts and phase-shifted scalars
- Running a Harmonic Balance simulation
- Obtaining an XY plot of pressure on a specific section of the blade

Heat Transfer and Radiation

The tutorials in this set illustrate various STAR-CCM+ features for heat transfer, radiation, and thermal comfort.

Contents:

Conjugate Heat Transfer: Heated Fin Introduction
Multi-Part Solid: Graphics Card Cooling
Natural Convection: Concentric Cylinders Introduction
Electronics Cooling Toolset: Natural Convection JEDEC
Electronics Cooling Toolset: Graphics Card Cooling
Dual Stream Heat Exchanger: Car Radiator
Surface-to-Surface Radiation: Thermal Insulator
Multiband Surface-to-Surface Radiation: Solar Collector
Thermal Comfort Wizard: Single Occupant in a Cabin

Conjugate Heat Transfer: Heated Fin Introduction

This tutorial incorporates both multi-region conjugate heat transfer and natural convection in a STAR-CCM+ simulation.

The starting point for this tutorial is a one-cell-thick hexahedral grid representing a solid fin within a rectangular enclosure containing air, as shown in the Figure below. Built with pro-STAR, the grid includes a non-conformal grid interface between the fin and surrounding air. The file containing grid data also includes information about the boundary types present in the problem.

The three-dimensional grid is converted to a two-dimensional one and also scaled to the appropriate size before being used for the analysis. All conditions and properties are defined using the STAR-CCM+ GUI and solved using the coupled solver. The results are displayed using the post-processing tools available in STAR-CCM+.

The analysis includes the following heat transfer modes:

- Conduction in the solid fin.
- Diffusion and natural convection in the surrounding fluid.

Radiation effects are not included as part of this analysis.

Contents:

Prerequisites
Importing the Mesh and Naming the Simulation
Converting to a Two-Dimensional Mesh
Renaming Regions and Boundaries
Scaling the Mesh
Visualizing the Interior Two-Dimensional Mesh
Setting Up the Models
Setting Material Properties
Setting Initial Conditions and Reference Values
Creating Interfaces
Setting Boundary Conditions and Values
Visualizing the Solution
Reporting and Monitoring with Plots
Initializing and Running the Simulation
Visualizing the Results
Adding Streamlines
Summary

Prerequisites

The instructions in the Conjugate Heat Transfer tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	STAR-CCM+ Introductory Tutorial
Using visualization tools, scenes, and plots	STAR-CCM+ Introductory Tutorial

Importing the Mesh and Naming the Simulation

To set up the STAR-CCM+ simulation, launch a simulation and import the supplied volume mesh.

Use the following steps:

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh**.
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/heatTransferAndRadiation/data`.
5. Select `fin.ccm` then click **Open**.

   STAR-CCM+ provides feedback on the import process in the *Output* window. Two mesh regions named **AIR** and **AL** are created in the **Regions** node representing the grid domain. A geometry scene is created in the *Graphics* window.
6. Save the simulation as `heatedFin.sim`.

Converting to a Two-Dimensional Mesh

For greater efficiency, convert the mesh region from a one-cell-thick three-dimensional grid to a two-dimensional one.

There are special requirements in the STAR-CCM+ product for three-dimensional meshes that are converted to two-dimensional ones. These requirements are:

- The grid must be aligned with the X-Y plane.
- The grid must have a boundary plane at the Z = 0 location.

The mesh imported for this tutorial was built with these requirements in mind. Were the grid not to conform to the above conditions, it would have been necessary to realign the region using the transformation and rotation facilities in STAR-CCM+.

1. Select **Mesh** > **Convert To 2D**.
2. In the *Convert Regions To 2D* dialog:
   - Make sure that all regions are selected.
   - Activate the **Delete 3D Regions After Conversion** option.

3. Click **OK**.

   Once you click **OK**, the mesh conversion takes place. The new two-dimensional mesh is shown, viewed from the z-direction, in the *Geometry Scene 1* display which was created after you imported the mesh. The mouse rotation option is suppressed for two-dimensional scenes.
4. Right-click the **Continua** > **Physics 1** continuum node and select **Delete**.
5. Click **Yes** in the confirmation dialog.
6. Repeat these steps for the **Physics 2** continuum node.

Renaming Regions and Boundaries

Change the default region and boundary names to more suitable names.

Use the following steps:

1. Rename **AIR 2D** to `Fluid`.
2. Rename the boundaries within the **Fluid** node:

Original Name	New Name
wall	`Air-Interface`
wall 2	`Vertical-Walls`
wall 3	`Horizontal-Walls`

3. Rename **AL 2D** to `Solid`.
4. Rename the boundaries within the **Solid** node:

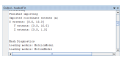
Original Name	New Name
Default_Boundary_Region	`Fin-Interface`
wall	`Fin-Bottom`

   This portion of the object tree appears as shown below.

Scaling the Mesh

The imported grid uses units that differ from those in STAR-CCM+. To compensate, scale the mesh.

The original grid was built using centimeters as a length unit, but no scaling factor was applied when the grid was saved to the `ccm` file. The current dimensions of the mesh regions are already available from the mesh import process and are displayed in the *Output* window.



The output shows that the mesh regions extend from 0 to 10.0 in the x-direction, 0 to 10.0 in the y-direction and 0.0 to 1.0 in the (now defunct) z-direction. Without any further action, STAR-CCM+ interprets these values as specified in meters. Therefore, scale the region by a factor of 0.01 so that the dimensions are correctly specified.

1. To scale the region, select **Mesh** > **Scale Mesh** from the menu bar.
2. In the *Scale Mesh* dialog, set the **Scale Factor** to `0.01` for the fluid and solid regions.
3. Click **Apply**.

   The mesh region reduces in size.
4. Click **Close**.
5. To restore the previous viewing distance for the scaled domain in the display, click (Reset View) in the toolbar.
6. To verify that scaling has in fact been applied:
   a) Select **Mesh** > **Diagnostics** from the menu bar.
   b) In the *Mesh Diagnostics* dialog, click **OK** for all regions.
7. Check and review the output values.

Visualizing the Interior Two-Dimensional Mesh

This part of the tutorial involves examining the interior of the two-dimensional mesh of the heated fin.

Use the following steps:

1. Select the **Scenes** > **Geometry Scene 1** > **Displayers** > **Geometry 1** node.
2. Activate the checkbox of the **Mesh** property.

   This shows the mesh in the solution domain interior.
3. To get a clearer view of individual mesh regions, highlight them. For example, select the **Regions** > **Fluid** node.

Setting Up the Models

Models define the primary variables of the simulation, including pressure, temperature, velocity, and what mathematical formulation is used to generate the solution. In this example, the flow is turbulent and compressible.

The Coupled Flow model is used together with the default K-Epsilon turbulence model. As there are two materials (air and aluminum) present, two regions are required for the analysis. Define two sets of continuum models, each appropriate to a different region. By default, **Physics 1 2D** and **Physics 2 2D** are automatically created when the mesh is converted to two-dimensional. Use these continua for the fluid and the solid.

1. Rename the continua:

Original Name	New Name
Physics 1 2D	`Air`

Original Name	New Name
Physics 2 2D	`Aluminum Alloy`

2. For the physics continuum, **Continua** > **Air**, select the following models in order:

Group Box	Model
Enabled Models	**Two Dimensional** (Selected automatically)
Time	**Steady**
Material	**Gas**
Flow	**Coupled Flow** **Gradients** (Selected automatically)
Equation of State	**Ideal Gas**
Viscous Regime	**Turbulent** **Reynolds-Averaged Navier-Stokes** (Selected automatically)
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence** **Realizable K-Epsilon Two-Layer** (Selected automatically) **Two-Layer All y+ Wall Treatment** (Selected automatically)
Optional Models	**Gravity**

3. Click **Close**.

   The color of the **Air** node turns from gray to blue to indicate that models have been activated.

4. To review the models, open the **Air** > **Models** node.

5. For the continuum of the solid region, **Continua** > **Aluminum Alloy**, select the following models in order:

Group Box	Model
Enabled Models	**Two Dimensional** (Selected automatically)
Material	**Solid**
Optional Models	**Coupled Solid Energy** **Gradients** (Selected automatically)
Equation of State	**Constant Density**
Time	**Steady**

6. Click **Close**.

   The color of the **Aluminum Alloy** node turns from gray to shaded gray to indicate that models have been activated.

7. To review the models, open the **Aluminum Alloy** > **Models** node.

8. Save the simulation.

Setting Material Properties

This part of the tutorial involves changing the solid material properties for the aluminum alloy fin.

The fin has a density of 2800 kg/m^3, a specific heat of 880 J/kg-K and a thermal conductivity of 180 W/m-K. The surrounding air is considered compressible based on the Ideal Gas Law. By default it has a variable density, a molecular weight of 28.9664 kg/kg.mol, a specific heat of 1003.62 J/kg-K, a thermal conductivity of 0.0260305 W/m-K, and a dynamic (molecular) viscosity of 1.85508E-5 Pa-s.

1. Navigate to the **Continua** > **Aluminum Alloy** > **Models** > **Solid** > **Al** node.
2. Edit the **Material Properties** node and set the following properties:

Node	Property	Setting
Density > **Constant**	**Value**	`2800 kg/m^3`
Specific Heat > **Constant**	**Value**	`880 J/kg-K`
Thermal Conductivity > **Constant**	**Value**	`180 W/m-K`

3. Save the simulation.

Setting Initial Conditions and Reference Values

This part of the tutorial involves setting the direction and magnitude of the gravity vector and the reference density for the simulation.

Start with the acceleration force in the negative y-direction.

1. Edit the **Continua** > **Air** > **Reference Values** node and set the following properties:

Node	Property	Setting
Gravity	**Value**	`[0.0, -9.81] m/s^2`
Reference Density	**Value**	`1.27588 kg/m^3`

2. Edit the **Continua** > **Air** > **Initial Conditions** > **Static Temperature** node and set **Value** to `293 K`.
3. Edit the **Continua** > **Aluminum Alloy** > **Initial Conditions** > **Static Temperature** node and set **Value** to `343 K`.

   This temperature is close to the final expected temperature of the solid region.
4. Save the simulation.

Creating Interfaces

STAR-CCM+ requires the existence of an interface between all regions to transfer the appropriate mass and energy quantities during the calculation.

For this example, create a contact-type interface between the fluid and solid regions. This procedure involves creating a so-called “in-place interface” between the wall boundaries enclosing each of the two regions, and then defining this interface as being of an appropriate “contact” type for fluid-solid heat transfer.

1. Using the multi-select method, select the following nodes:
   - **Regions** > **Fluid** > **Boundaries** > **Air-Interface**
   - **Regions** > **Solid** > **Boundaries** > **Fin-Interface**
2. Right-click any of the selected nodes and select **Create Interface** to create an in-place interface.

   This step creates the following new nodes:

   - **Fluid** > **Boundaries** > **Air-Interface [Interface 1]**
   - **Solid** > **Boundaries** > **Fin-Interface [Interface 1]**

   When you initialize the flow, these boundaries form the connection between the fluid and solid regions, superseding the original wall boundaries.

   In addition to the new in-place boundaries, a folder node named **Interfaces** appears in the object tree. It contains an interface node called **Interface 1**.
3. Edit the **Interfaces** > **Interface 1** node and set the following properties:

Property	Setting
Type	`Contact Interface`
Topology	`In-place`

4. Save the simulation.

Setting Boundary Conditions and Values

The heated fin consists of an inverted “T” shape placed in a closed box. The only boundary specifications required for this analysis are wall thermal conditions.

Apply a fixed temperature of 343 K to the base of the solid fin to provide the heat source. Set the vertical sides of the fluid-containing box to a fixed temperature of 293 K. All other wall boundary conditions are considered to be adiabatic. Use a contact-type conducting boundary at the solid-fluid interface.

Start with the wall boundary definitions for the fluid region.

1. Navigate to the **Regions** > **Fluid** > **Boundaries** > **Vertical-Walls** > **Physics Conditions** node.
2. Select the **Thermal Specification** node and set **Condition** to `Temperature`.

   The temperature value for this boundary can now be set.
3. Select the **Vertical-Walls** > **Physics Values** > **Static Temperature** node and set Value to 293 K.

   The conditions for the Horizontal-Walls boundary (upper and lower walls of the fluid region) can remain at the default adiabatic setting.

You can now set the solid wall boundary conditions and values in a similar way to those of the fluid.

4. Navigate to the **Regions** > **Solid** > **Boundaries** > **Fin-Bottom** > **Physics Conditions** node.
5. Select the **Thermal Specification** node and set **Condition** to `Temperature`.

   The temperature value for this boundary can now be set.
6. Select the **Fin-Bottom** > **Physics Values** > **Static Temperature** node and set Value to 343 K.
7. Save the simulation.

Visualizing the Solution

This part of the heated fin tutorial involves setting up a scene for displaying scalars and vectors of the solution.

To view the temperature of the fluid and solid regions as the solution develops:

1. Create a scalar scene.

   This scene includes both the fluid and solid regions.
2. Select **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Scalar Field**.
3. Set the *Function* property to `Temperature`.

   The velocity vectors for the fluid can also be included in the scene by adding a vector displayer to the scalar scene.
4. Create a vector displayer: right-click on **Scalar Scene 1** > **Displayers** and select **New Displayers** > **Vector**.
5. Edit the **Scenes** > **Scalar Scene 1** > **Displayers** > **Vector 1** > **Parts** node and select the fluid region.
6. Save the simulation.

Reporting and Monitoring with Plots

STAR-CCM+ can dynamically monitor virtually any quantity while the solution develops.

To monitor a quantity, set up a report which defines the quantity of interest and the region parts to monitored. Define a monitor based on the report, which controls the update frequency and normalization characteristics. Finally, create an X-Y graph plot can from the monitor.

Use a report monitor to track the heat flux at the interface of the heated fin and surrounding air:

1. Create a heat transfer report: right-click **Reports** and select **New Report** > **Heat Transfer**.

   This creates a node called **Heat Transfer 1** within the **Reports** node.
2. Edit the **Reports** > **Heat Transfer 1** > **Parts** property and select the **Solid** > **Boundaries** > **Fin-Interface [Interface 1]** node.

   The setup for the report is now complete, and a monitor can be made from that report.
3. Create a monitor from the report: right-click the **Heat Transfer 1** node and select **Create Monitor from Report**.
4. Create a plot from the monitor: right-click the **Heat Transfer 1 Monitor** node and select **Create Plot from Monitor**.
5. Save the simulation.

Initializing and Running the Simulation

To verify that the initial conditions for the regions are correct before you run the simulation, STAR-CCM+ allows you simply to perform solver initialization and then view the results using an appropriate scene.

Use the following steps:

1. Initialize the simulation.
2. To see the initialization result, look at the scalar scene display.

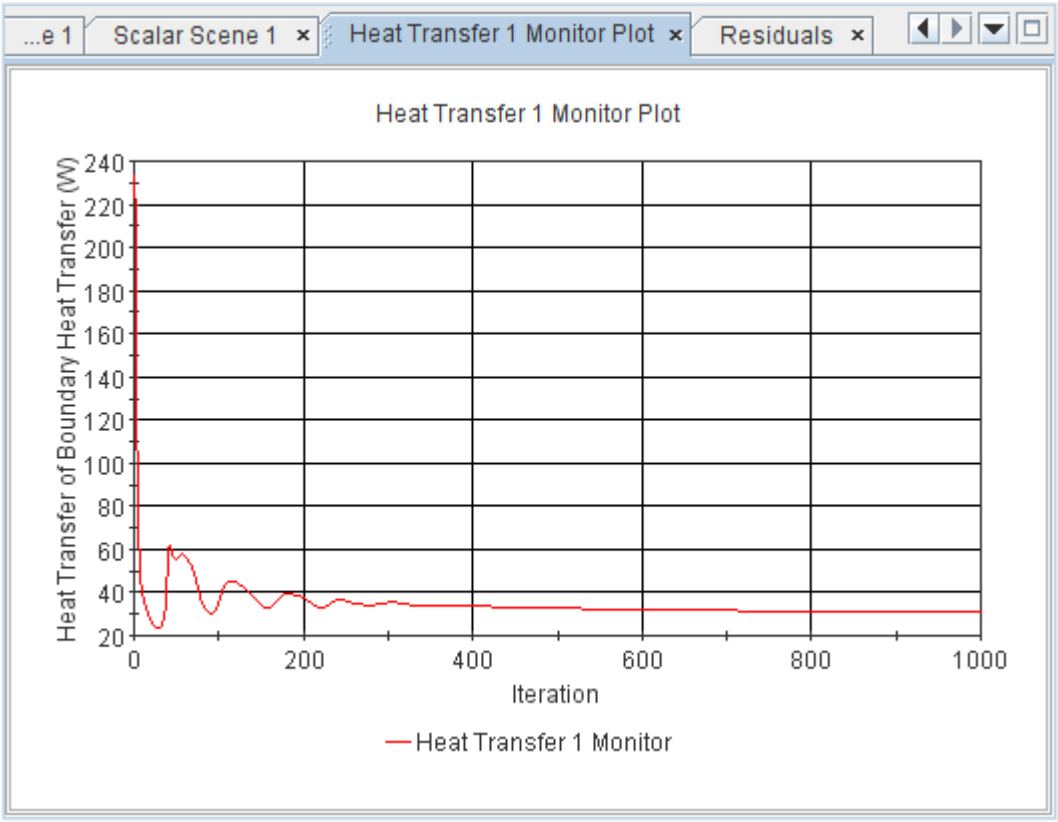
   As the colors show, the solid fin is at 343 K while the surrounding air is at 293 K. The velocity value for each cell is 0. The simulation can now be run.
3. Run the simulation.
4. To watch the solution being updated while the simulation is running, look at the scalar scene display.
5. When the simulation has finished running, save it.

Visualizing the Results

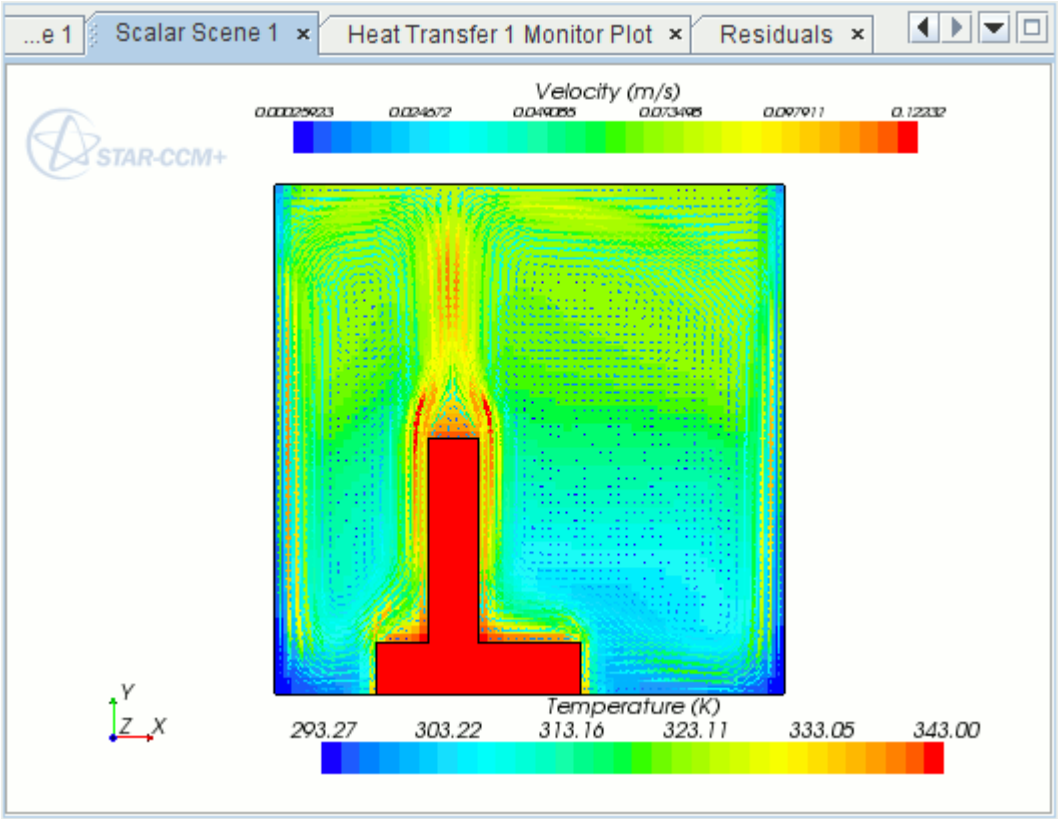
After the solution finishes, you can examine results in plots and scenes.

Click the tab of each display that you set up before the run.

1. To view the heat transfer variation at the solid-fluid interface, activate the *Heat Transfer 1 Monitor Plot* display.



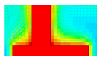
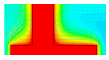
2. To see the temperature and velocity vector results, click the tab for the *Scalar Scene 1* display.



By default, filled cell values are shown. To make the display smoother:

3. Select the **Scenes** > **Scalar Scene 1** > **Displayers** > **Scalar 1** node and set Contour Style to `Smooth Filled`.

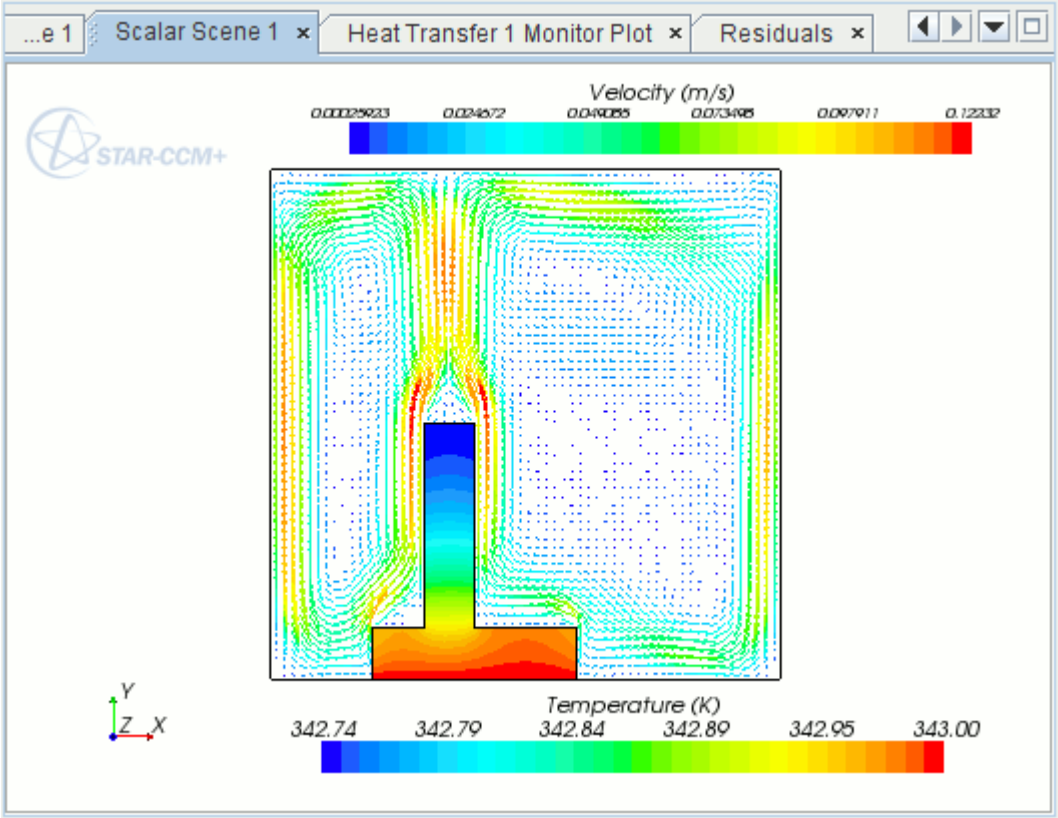
   The contours of the scalar display now appear smooth as seen in the following zoomed-in screenshot.

Filled:

Smooth Filled:


Additionally, the temperature solution of each region can be viewed by selecting the appropriate part.

4. Select the **Displayers** > **Scalar 1** > **Parts** node and deselect the Fluid region.

   The display appears as shown below.



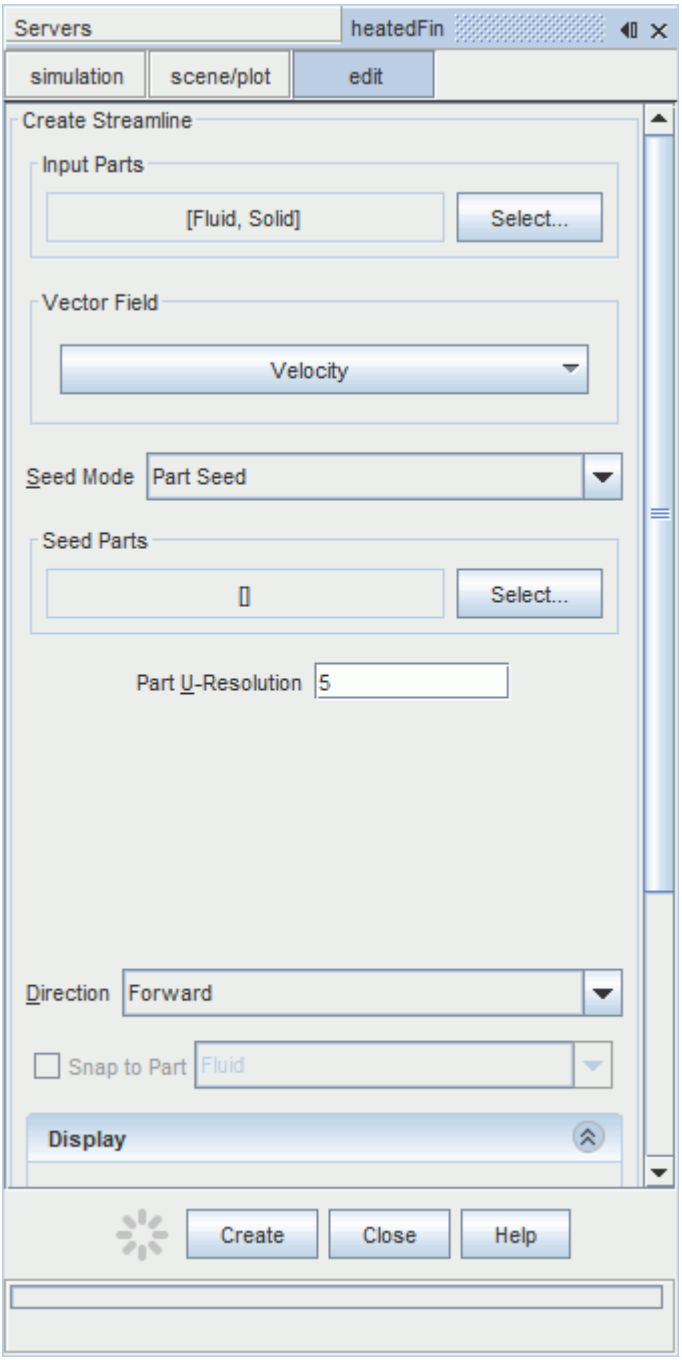
Adding Streamlines

In this part of the heated fin tutorial, use streamlines to depict flow paths.

Streamlines can be added into any of the scalar or geometry scene plots with the creation of a “streamline” derived part.

1. Make the *Geometry Scene 1* display active.
2. Create a new streamline derived part.

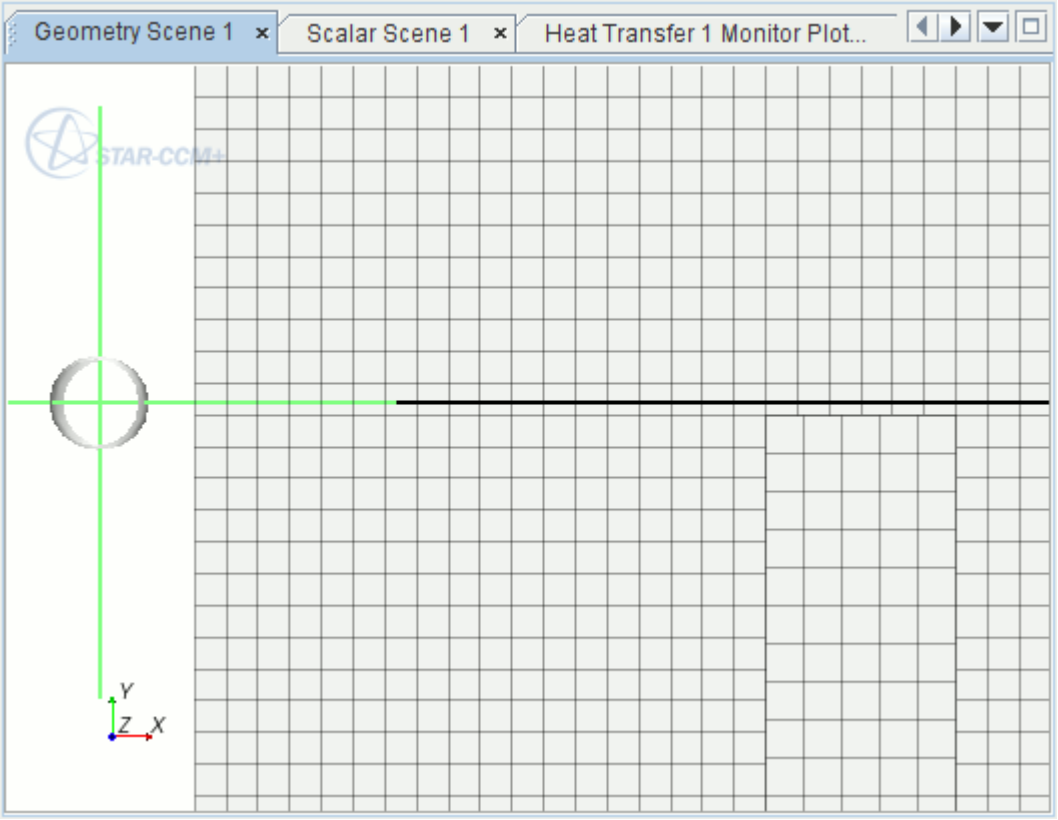
   The *Create Streamline* inset panel appears to allow definition of the derived part.



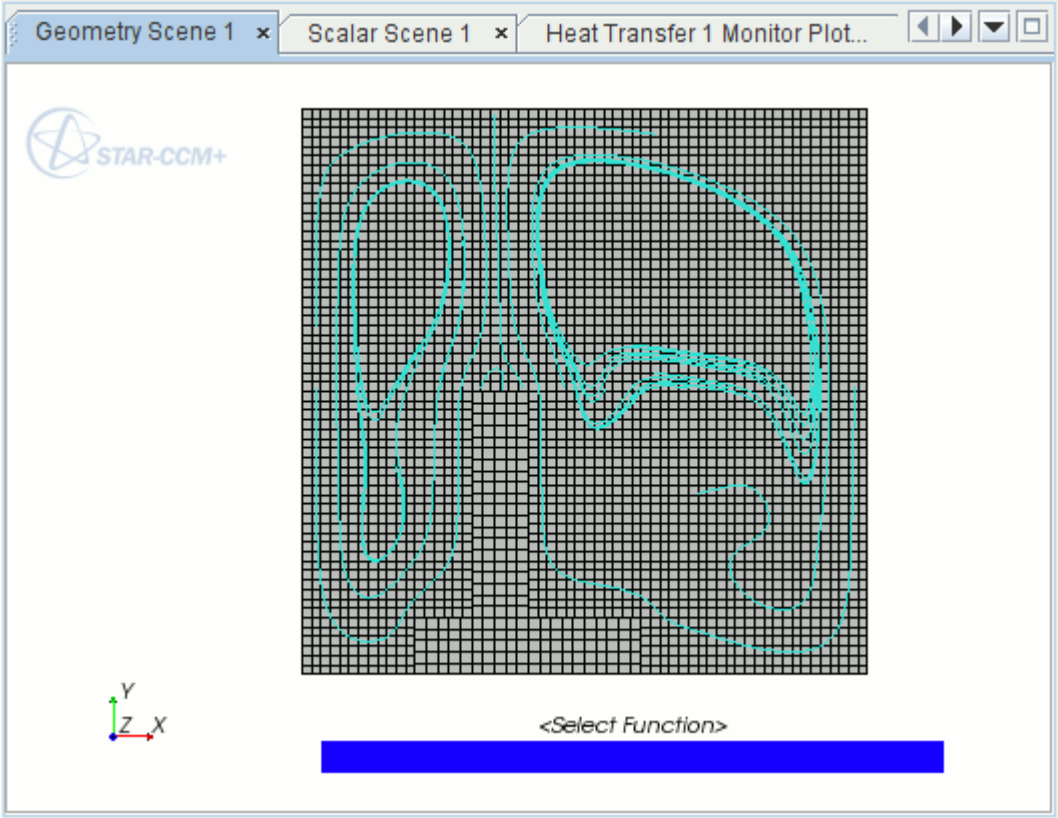
3. To select input parts, click **Select**.
4. In the object selector, select only the Fluid region.
5. In the *Create Streamline* panel, make the following settings:
   a) For the seed mode, select `Line Seed`.
   b) Activate the display tool option, and set the resolution to `15`.

      The streamline line seed tool appears in the display.
6. Work with the tool in the scene as follows:
   a) Using the mouse buttons, zoom-in and position the part as shown below.

b) Adjust the line so that it is horizontal, approximately half a cell above the top of the solid fin. To adjust the line, drag the end spheres with the mouse.



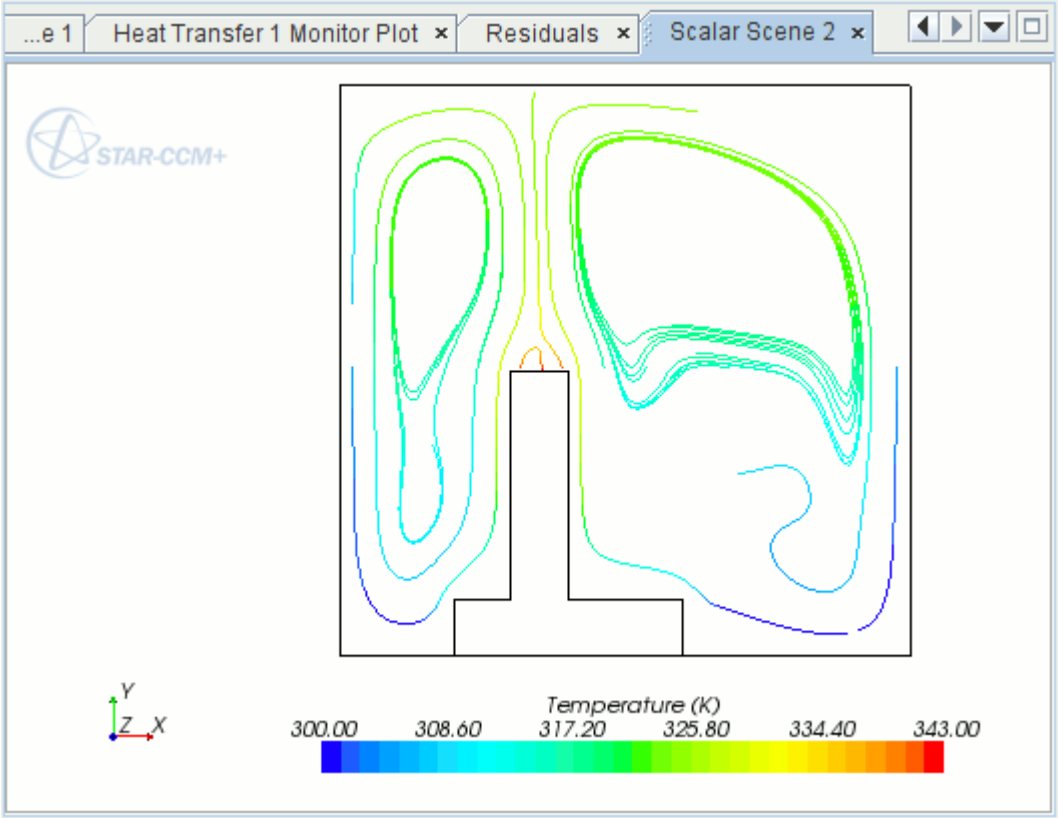
7. Click **Create** and then **Close**.
8. Click (**Reset View**) in the toolbar.
   The streamlines are generated and shown in the display.
9. Position the part as shown below.



The streamline part can now be added to any suitable scene or be displayed in its own scene if desired.

10. Create a scalar scene, *Scalar Scene 2*: right-click **Scenes** and select **New Scene** > **Scalar**.
11. Set the scalar value to `Temperature`: select **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Scalar Field** and set the *Function* property to **Temperature**.
12. By default, the regions Fluid and Solid are included in the scene. To display the streamlines only:
    a) Edit the **Scenes** > **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Parts** node.
    b) Deselect the regions and select the derived part.
13. Edit the **Scenes** > **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Scalar Field** node:
    a) Set **Min** to `300`.
    b) Set **Max** to `343`.

    Streamlines colored according to the local temperature appear in the *Scalar Scene 2* display.



14. Save the simulation.

Summary

The Conjugate Heat Transfer tutorial used multi-region conjugate heat transfer and natural convection to introduce a variety of STAR-CCM+ features.

The features that were introduced include:

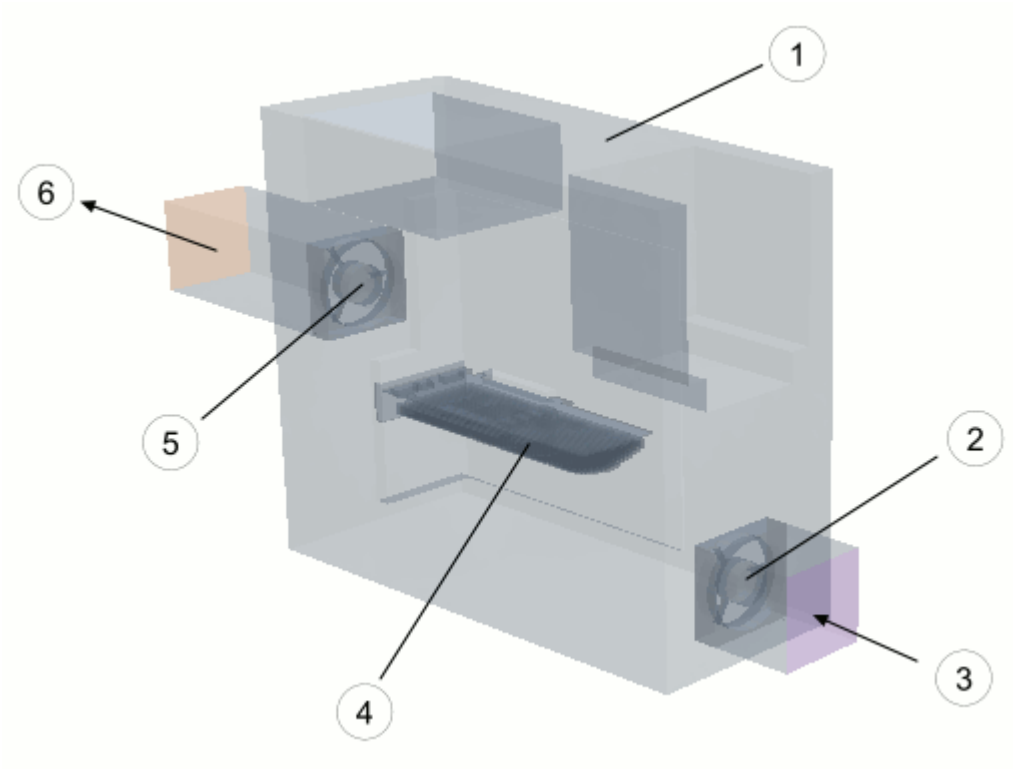
- Scaling the mesh.
- Plotting geometry and scalar scenes.
- Defining models for multiple continua.
- Defining material properties required for the selected models.
- Initializing and running the solver until the residual magnitudes are sufficiently small.
- Analyzing results using the visualization and monitor plot facilities.
- Creating streamlines.

Multi-Part Solid: Graphics Card Cooling

This tutorial demonstrates how you can model multiple solid parts in a single continuum using the Multi-Part Solid physics model.

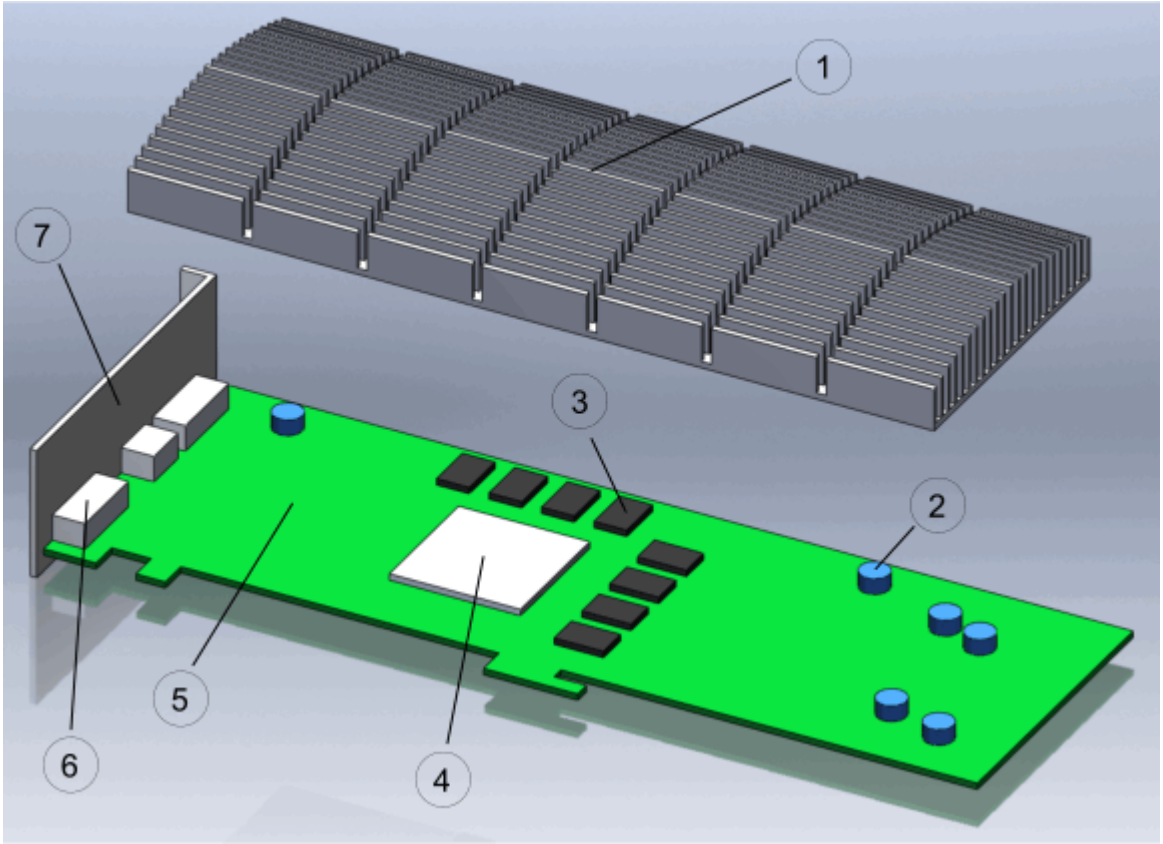
This tutorial also demonstrates how to apply different heat sources on parts in a single region using Parts-Based Profiles.

The geometry for this simulation consists of a simplified graphics card inside a PC tower case. Two fans, represented by fan interfaces, drive the flow inside the case.



Index	Description
1	PC Tower Case
2	Inlet Fan
3	Inlet
4	Graphics Card
5	Outlet Fan
6	Outlet

The graphics card and its part materials are shown in the following diagram.



Index	Description	Material
1	Heat Sink	Aluminum
2	Capacitor	Alumina
3	Memory Chip	Silicon
4	GPU Chip Casing	Aluminum
5	Printed Circuit Board	FR-4 + Copper
6	Connector Port Block	Aluminum
7	Back Plate	Aluminum

Using multi-part solids and per-part values requires that you import the geometry as parts. From these imported parts, you generate a mesh that has polyhedral cells and prims layers. Thin parts of the geometry are meshed using the thin mesher.

Contents:

Prerequisites
Importing the Surface Mesh
Defining Regions Layout
Selecting Physics Models
Assigning Continua to Regions

Setting the Boundary Types
Assigning Materials to the Multi-Part Solid Parts
Setting up the Heat Sources
Defining the Anisotropic Thermal Conductivity of the PCB
Setting up the Fan Interfaces
Setting up the Meshers
Preparing Scenes for Visualization
Reporting Temperature with a Monitor and a Plot
Modifying Solver Settings and Stopping Criteria
Running the Simulation
Visualizing the Results
Summary

Prerequisites

The instructions in the Multi-Part Solid tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Importing the Surface Mesh

The meshing process starts from a CAD geometry in Parasolid format.

To import the geometry:

1. Launch STAR-CCM+.
2. Start a simulation.

   You are advised to run this tutorial in parallel using at least 8 cores. This tutorial case has many cells and can take a long time to run to convergence on one core.

To use the Multi-Part Solid model with Parts-Based Profiles, import all CAD geometry as parts:

3. Select **File** > **Import** > **Import Surface Mesh**.
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/heatTransferAndRadiation/data`.
5. Select `graphicsCardAndEnclosure.x_t` then click **Open**.
6. To accept the default *Import Surface Options*, click **OK**.

   A total of 26 parts are imported. Part contacts between coincident parts are automatically generated during the import process.

7. Save the simulation as `graphicsCardCooling.sim`.

Defining Regions Layout

Two regions are required for this simulation: one for the air and one for the solid components.

The Printed Circuit Board (PCB) exhibits anisotropic thermal conductivity. The multi-part solid model in STAR-CCM+ allows you to model anisotropic thermal conductivity in a single region. This region contains all the solid components of the graphics card and the fan casings.

Parts-based interfaces allow you to group related surfaces in a single interface. Grouping similar interfaces together simplifies the simulation setup and can provide performance improvements, especially in cases with large amounts of interfaces. You can automatically create parts-based interfaces when assigning parts to regions.

To create the air region:

1. Right-click the **Geometry** > **Parts** > **Air Case** node and select **Assign Parts to Regions**.
2. In the *Assign Parts to Regions* dialog:
   a) From the *Parts* list, select **Air Case**, **Air Inlet**, and **Air Outlet**.
   b) Select **Create One Region for All Parts** and enter `Air` in the neighboring text box.
   c) Select **Create a Boundary for Each Part Surface**.
   d) Select **Create Contact-mode Interfaces From Contacts**.
   e) Click **Apply**.

   STAR-CCM+ creates one region for the selected parts, and separate boundaries for each named surface.

Create a region for all the solid components:

3. In the same *Assign Parts to Regions* dialog, select all the parts in the dialog.
4. Deselect the following parts:
   - **Air Case**
   - **Air Inlet**
   - **Air Outlet**
5. Select **Create One Region for All Parts** and enter `Solid Components` in the neighboring text box.
6. Click **Apply** then **Close**.

   STAR-CCM+ creates the region, **Solid Components**.
7. Expand the **Interfaces** node.
8. Rename the interfaces as follows:

Old Name	New Name
Air Case/Air Inlet	`Case Fans`
Air Case/Capacitor 1	`Air/Solid Components`
Capacitor 1/PCB	`Solid Components/Solid Components`

9. Save the simulation.

Selecting Physics Models

This tutorial requires two physics continua: one for the air and one for solid components.

Contents:

Creating the Physics Continuum for the Solid Components
Creating the Physics Continuum for the Air

Creating the Physics Continuum for the Solid Components

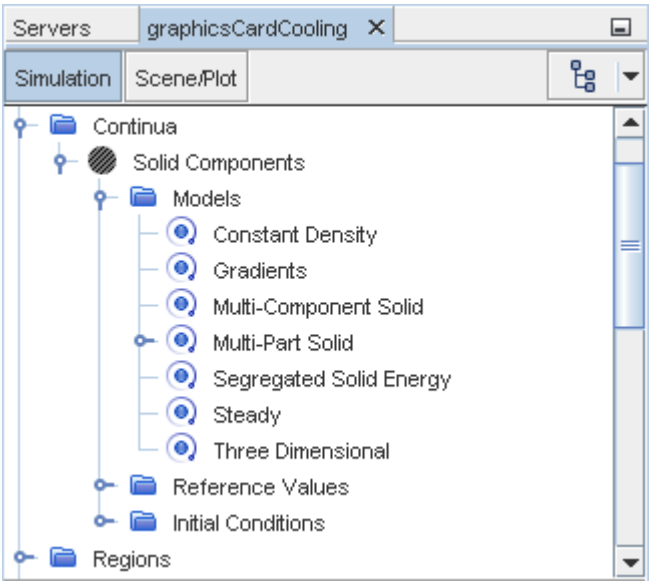
Set up the continuum for the solid components.

The first physics continuum that you create is used to model all the solid components of the graphics card and the fan casings:

1. Create a physics continuum.
2. Rename the continuum to `Solid Components`.
3. For the physics continuum, **Continua** > **Solid Components**, select the following models in order:

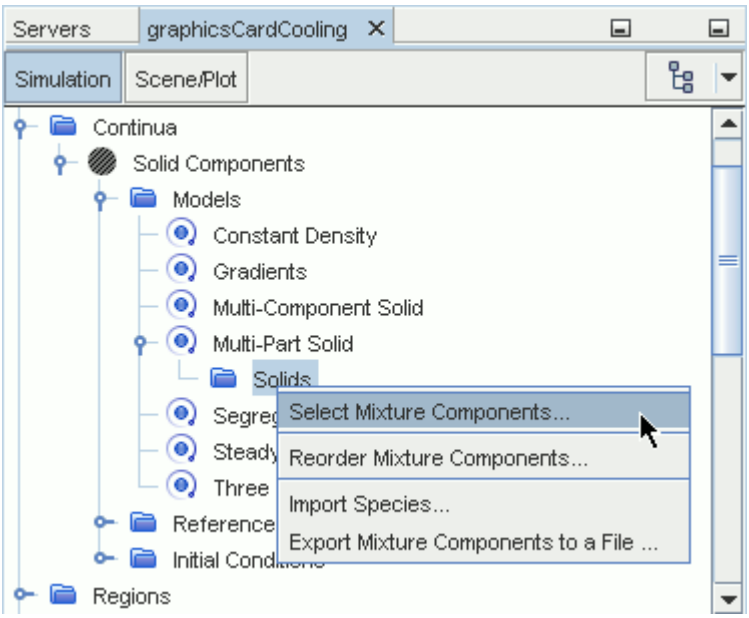
Group Box	Model
Space	**Three Dimensional**
Time	**Steady**
Material	**Multi-Component Solid**
Multi-Component Solid	**Multi-Part Solid**
Optional Models	**Segregated Solid Energy**
	Gradients (Selected automatically)
Equation of State	**Constant Density**

4. Click **Close**.
5. To review the models, open the **Solid Components** > **Models** node.



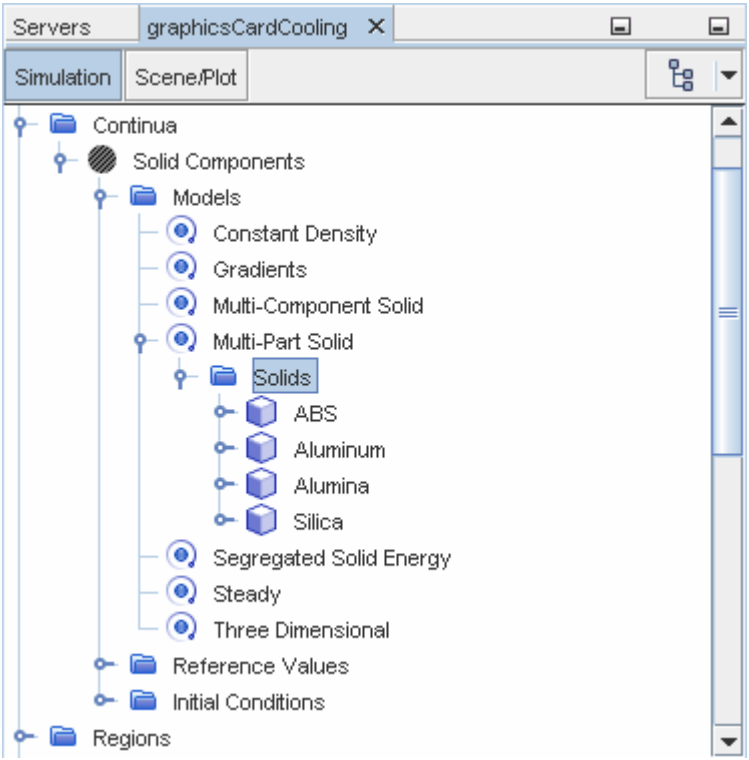
Multiple solid parts are modeled in the Solid Components continuum. Some parts are made of different materials. To define these materials in the multi-part solid model:

6. Expand the **Continua** > **Solid Components** > **Models** > **Multi-Part Solid** node.
7. Right-click the **Solids** node and choose **Select Mixture Components...**.



8. In the *Select Mixture Components* dialog, expand the **Material Databases** > **Standard** > **Solids** node.
9. Select any four materials from the list.

   The materials that you choose here are not important — you will change the material properties for these selections in the subsequent steps.
10. Click **Apply** and then **Close**.
11. Rename the materials to `ABS`, `Aluminum`, `Alumina`, and `Silicon`.



12. Edit the **Multi-Part Solid** node and set the material properties as follows:

Node	Property	Setting
ABS		
└ **Material Properties**		
└ **Density** > **Constant**	*Value*	`1050 kg/m^3`
└ **Specific Heat** > **Constant**	*Value*	`2050 J/kg-k`
└ **Thermal Conductivity** > **Constant**	*Value*	`2.5 W/m-K`

Node	Property	Setting
Aluminum		
└ **Material Properties**		
└ **Density** > **Constant**	*Value*	`2700 kg/m^3`
└ **Specific Heat** > **Constant**	*Value*	`896 J/kg-k`
└ **Thermal Conductivity** > **Constant**	*Value*	`167 W/m-K`

Node	Property	Setting
Alumina		
└ **Material Properties**		
└ **Density** > **Constant**	*Value*	`3960 kg/m^3`
└ **Specific Heat** > **Constant**	*Value*	`850 J/kg-k`
└ **Thermal Conductivity** > **Constant**	*Value*	`30 W/m-K`

Node	Property	Setting
Silicon		
└ **Material Properties**		
└ **Density** > **Constant**	*Value*	`2330 kg/m^3`
└ **Specific Heat** > **Constant**	*Value*	`700 J/kg-k`
└ **Thermal Conductivity** > **Constant**	*Value*	`124 W/m-K`

13. Click **Close**.

The PCB is constructed from layers of FR-4 glass-reinforced laminate and copper foil. As a result, the PCB has the same thermal conductivity within the same layer but a different thermal conductivity between layers.

Add a material for the PCB to the mixture of solids:

14. Right-click the **Solids** node and choose **Select Mixture Components...**.
15. In the *Select Mixture Components* dialog, expand the **Material Databases** > **Standard** > **Solids** node, and select any material from the list.
16. Click **Apply** and then **Close**.
17. Rename the new material to `FR-4 + Copper`.
18. Expand the **FR-4 + Copper** > **Material Properties** node and set the material properties as follows:

Node	Property	Setting
FR-4 + Copper		
└ **Material Properties**		
└ **Density** > **Constant**	*Value*	`33 kg/m^3`
└ **Specific Heat** > **Constant**	*Value*	`37 J/kg-k`
└ **Thermal Conductivity**	*Method*	**Anisotropic**

The anisotropic properties of the PCB are defined in a later section.

Creating the Physics Continuum for the Air

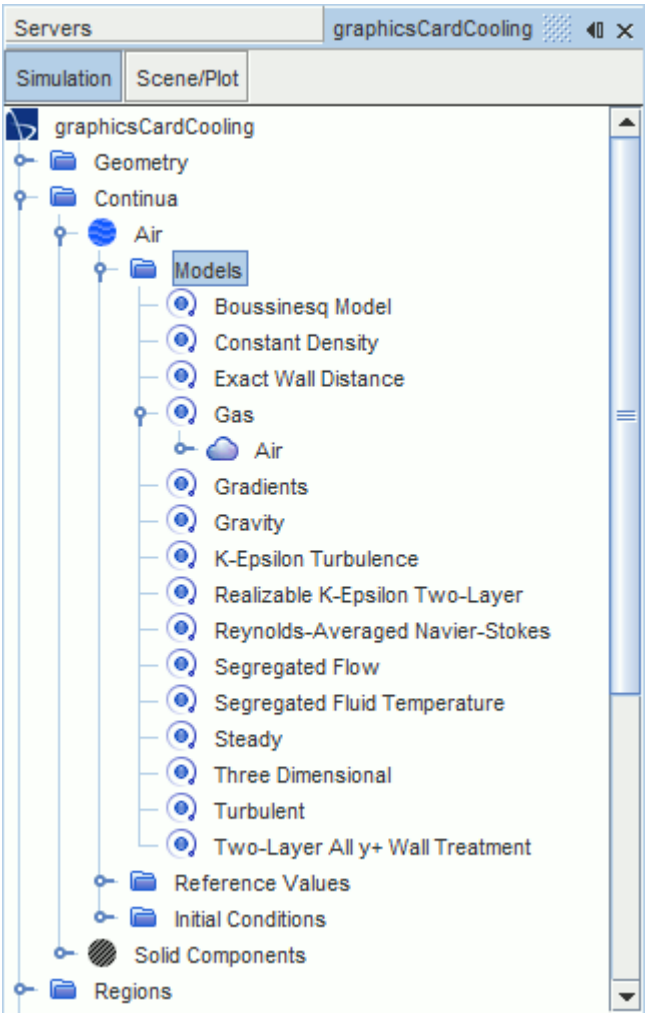
Create the physics continuum that is used to model the air in the fluid domain.

Constant density is used with the Boussinesq model. The Realizable K-Epsilon Two-Layer turbulence model is used to complement the mesh which contains two prism layers.

1. Create a physics continuum.
2. Rename the continuum to `Air`.
3. For the physics continuum, **Continua** > **Air**, select the following models in order:

Group Box	Model
Space	**Three Dimensional**
Time	**Steady**
Material	**Gas**
Flow	**Segregated Flow**
	Gradients (Selected automatically)
Equation of State	**Constant Density**
Viscous Regime	**Turbulent**
	Reynolds-Averaged Navier-Stokes (Selected automatically)
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**
	Realizable K-Epsilon Two-Layer (Selected automatically)
	Exact Wall Distance (Selected automatically)
	Two-Layer All y+ Wall Treatment (Selected automatically)
Optional Models	**Gravity**
	Segregated Fluid Temperature
	Boussinesq Model

4. Click **Close**.
5. To review the models, open the **Air** > **Models** node.



The Boussinesq model requires you to set the thermal coefficient of expansion for the fluid region. To define this property:

6. Expand the **Continua** > **Air** > **Models** > **Gas** > **Air** > **Material Properties** > **Thermal Expansion Coefficient** node.
7. Select the **Constant** node and set **Value** to `3.33E-3 /K`.

Adjust the reference gravity vector to act in the correct direction:

8. Select the **Continua** > **Air** > **Reference Values** > **Gravity** node and set *Value* to `[0.0, -9.81, 0.0] m/s^2`.

Assigning Continua to Regions

Since this simulation has multiple regions and continua, you must explicitly associate each region with the correct continuum.

1. Select the **Regions** > **Air** node and set *Physics Continuum* to **Air**
2. Select the **Regions** > **Solid Components** node and set *Physics Continuum* to **Solid Components**

Setting the Boundary Types

Define the boundary types for the inlet and outlet boundaries.

To set the types:

1. Expand the **Regions** > **Air** > **Boundaries** node.
2. Select the **Air Inlet.Inlet** node and set *Type* to **Stagnation Inlet**.
3. Select the **Air Outlet.Outlet** node and set *Type* to **Pressure Outlet**.

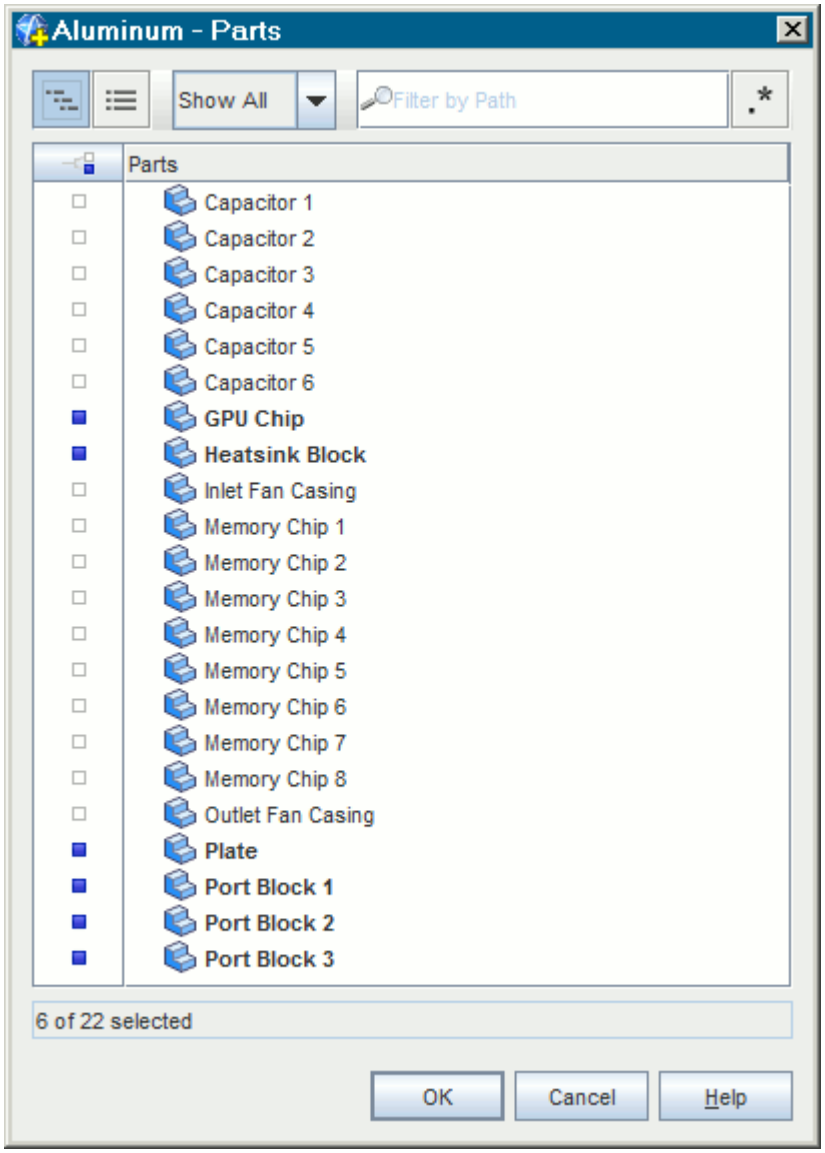
Assigning Materials to the Multi-Part Solid Parts

The **Components** region contains multiple solid parts that you must match with the materials defined in the **Multi-Part Solid** physics continuum.

Assign the parts to materials as follows:

1. Expand the **Regions** > **Solid Components** > **Physics Values** > **Material Part Groups** node.
2. Select the **Aluminum** node and click ... (**Custom Editor**) next to the *Parts* property.
3. In the *Aluminum* dialog, select the parts that are made of aluminum:
   - **GPU Chip**
   - **Heatsink Block**
   - **Plate**
   - **Port Block 1**
   - **Port Block 2**
   - **Port Block 3**

   The following screenshot shows the completed dialog.



4. Click **OK**.
5. Assign the other solid components to the remaining materials:
   - All the capacitors, **Capacitor 1** to **Capacitor 6**, to the **Alumina** node.
   - All the memory chips, **Memory Chip 1** to **Memory Chip 8**, to the **Silicon** node.
   - The fan casings, **Inlet Fan Casing**, and **Outlet Fan Casing**, to the **ABS** node.
   - The PCB, **PCB**, to the **FR-4 + Copper** node.

   Parts can only be assigned to one particular material. Parts that are already assigned to a material are removed from the list of parts of that material when you assign them to a new material.
6. Save the simulation.

Setting up the Heat Sources

Heat generated by the electronic components is represented in STAR-CCM+ using total heat sources. Heat sources are set on regions.

The GPU chip emits 25 W of heat. Each memory chip emits 4 W of heat. You can allocate different heat sources to different parts in the same region using Per-Part Values.

1. Select the **Regions** > **Solid Components** node and set *Allow Per-Part Values* to **True**.

   The **Part Subgroupings** node is added to the **Components** node.
2. Expand the **Solid Components** > **Part Subgroupings** > **Subgrouping 1** node.

   The **Subgroup 1** node contains all the parts that make up the **Solid Components** region. These parts have no heat assigned to them.
3. Rename **Subgroup 1** to `Default`.
4. Right-click the **Subgrouping 1** node and select **New**.

   A new subgroup, **Subgroup 1**, is added to **Subgrouping 1**.
5. Rename **Subgroup 1** to `GPU Chip`.
6. Select **GPU Chip** and set *Objects* to **GPU Chip**.
7. Create another subgroup with the following properties:

Property	Setting
Name	`Memory Chips`
Objects	**Memory Chip 1**
	Memory Chip 2
	Memory Chip 3
	Memory Chip 4
	Memory Chip 5
	Memory Chip 6
	Memory Chip 7
	Memory Chip 8

Modify the physics conditions to accept total heat source as the energy source:

8. Expand the **Solid Components** > **Physics Conditions** node.
9. Select the **Energy Source Option** node and set *Energy Source Option* to **Total Heat Source**.

Set the heat sources for the solid components.

10. Expand the **Solid Components** > **Physics Values** node.
11. Select the **Heat Source** node and set *Method* to **By Part Subgroup**.
12. Expand the **Heat Source** > **By Part Subgroup** node.

    The subgroups that you created appear here.
13. Edit the **By Part Subgroup** node and set the following properties:

Node	Property	Setting
Default	*Value*	*0.0 W*
GPU Chip	*Value*	*25.0 W*
Memory Chips	*Value*	*32.0 W*

Each memory chip emits 4 W of heat. There are 8 memory chips in this region, hence you multiply 4 W by the number of memory chips to get 32 W.

14. Save the simulation.

Defining the Anisotropic Thermal Conductivity of the PCB

The PCB is made up of multiple layers. While the thermal conductivity is the same within the same layer, it differs between the layers.

Define the anisotropic thermal conductivity of the PCB:

1. Expand the **Regions** > **Solid Components** > **Physics Values** > **Anisotropic Thermal Conductivity** > **FR-4 + Copper** > **Anisotropic Thermal Conductivity** node.
2. Edit the **Principal Tensor** node and set the properties as follows:

Node	Property	Setting
XX Component	*Value*	`10 W/m-k`
YY Component	*Value*	`0.5 W/m-k`
ZZ Component	*Value*	`10 W/m-k`

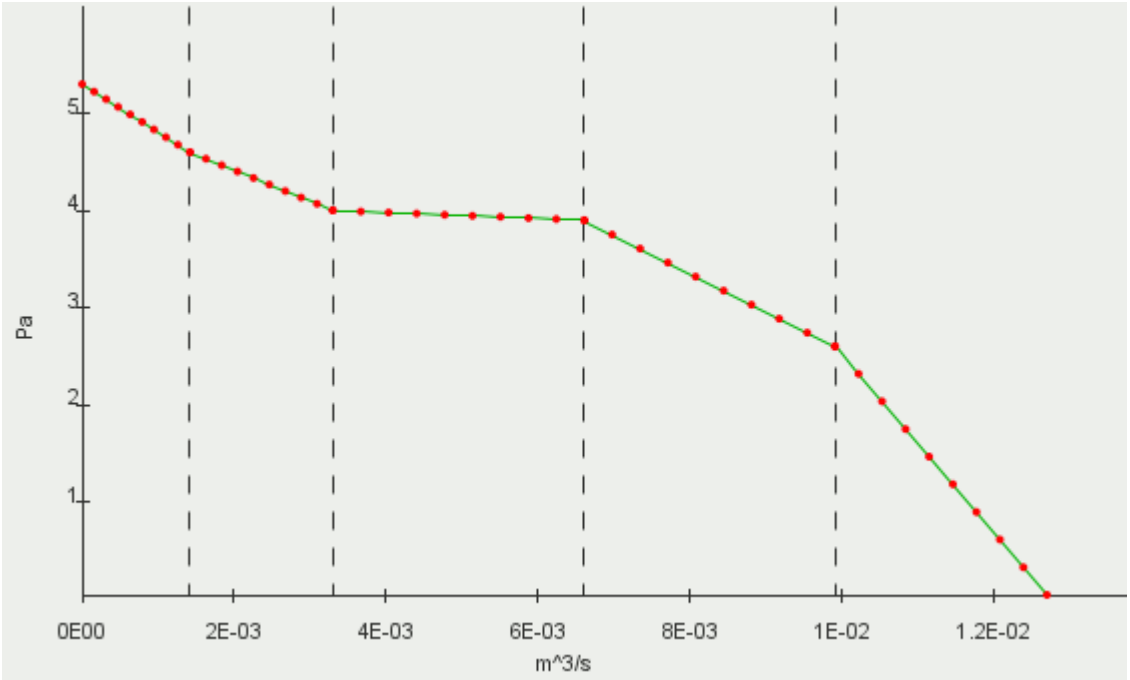
Setting up the Fan Interfaces

Two fans drive the flow inside the PC case.

In this tutorial, you model the case fans using fan interfaces. A fan interface models the pressure rise through an axial fan as a function of local flow rates or velocity. A fan interface can only be used between regions that belong to the same fluid continuum. In this tutorial, the fan interfaces are located between **Air Case** and **Air Inlet** and between **Air Case** and **Air Outlet**.

Using parts-based interfaces and per-contact values, you can model multiple fans using a single interface. In this tutorial, you model two identical fans, each defined by a part contact, using a single interface.

When you set up a fan interface, you supply the fan performance curve that corresponds to the physical fan that you are modeling. The fan performance curve specifies the pressure drop across the fan interface as a function of volumetric flow rate. The performance curve for the fans that are used in this tutorial is shown below. This data corresponds to the fan running at 800 rpm. In this tutorial, you run the fan at 1200 rpm—STAR-CCM+ scales the curve using fan laws.



The polynomials that define the fan performance curve are:

$$y = 5.3 - 494.0x$$

$$y = 5.05 - 318.0x$$

$$y = 4.1 - 30.3x$$

$$y = 6.5 - 394.0x$$

$$y = 11.7 - 918.0x$$

Set up the fan interface:

1. Expand the **Interfaces** node.
2. Select the **Case Fans** node and set *Type* to **Fan Interface**.
3. Set *Allow Per-Contact Values* to **True**.

   The **Contact Subgroupings** node is added to the **Case Fans** node.
4. Expand the **Case Fans** > **Contact Subgroupings** > **Subgrouping 1** node.

   The **Subgroup 1** node contains the contacts for the inlet fan and outlet fan.
5. Rename **Subgroup 1** to `Inlet Fan`.
6. Right-click the **Subgrouping 1** node and select **New**.

   A new subgroup, **Subgroup 1**, is added to **Subgrouping 1**.
7. Rename **Subgroup 1** to `Outlet Fan`.
8. Select **Outlet Fan** and set *Objects* to **Air Outlet/Air Case** > **Outlet Downstream/Outlet Upstream**.

Make sure that the orientation of each fan is correct. The flow moves from the upstream side of the interface to the downstream side of the interface:

9. Select the **Case Fans** > **Orientation** node and set *Mode* to **Manual**.

10. Right-click **Orientation** and select, **Reverse Contacts**.
11. In the *Reverse Contacts* dialog, expand the contacts folders.

    Each contact has an icon with a direction arrow, , to indicate the orientation of the interface. For **Inlet Upstream/Inlet Downstream**, the arrow indicates that the flow is moving from the upstream side of the interface to the downstream side of the interface, which is correct. For the **Outlet Downstream/Outlet Upstream** contact, arrow indicates that the flow is moving from the downstream side of the interface to the upstream side of the interface, which is incorrect.
12. To correct the orientation of the outlet fan interface, select the **Air Outlet/Air Case** > **Outlet Downstream/Oultet Upstream** node and click **OK**.
13. Select the **Case Fans** > **Physics Conditions** > **Fan Curve Type** node and set *Curve Type* to **Polynomial**.
14. Expand the **Case Fans** > **Physics Values** node.
15. Select the **Fan Curve Polynomial** node and set *Specify by Part Subgroup* to **True**.

    The **By Contact Subgroup** node appears within the **Fan Curve Polynomial** node.

Define the fan curve for the inlet fan:

16. Expand the **Fan Curve Polynomial** > **By Contact Subgroup** > **Inlet Fan** node.
17. Select the **Fan Curve Polynomial** node and click ... next to the *Polynomial* property.
18. In the *Fan Curve Polynomial* dialog:
    a) Set *Number of Intervals* to `5`.

       This property specifies the number of polynomials that define the fan curve.
    b) Set *Interval Ranges* to `[0.0, 0.00142, 0.0033, 0.00661, 0.00991, 0.0127]`.

       This property defines the upper and lower value for the volumetric flow rate for each interval. For example, the first interval has a lower value of `0.0 m^3/s` and an upper value of `0.00142 m^3/s`.
    c) Set *Number of Coefficients* to `[2, 2, 2, 2, 2]`.

       This property specifies how many coefficients each polynomial contains. In this tutorial, each polynomial consists of 2 coefficients. The number of coefficients in each range sets the number of exponents.
    d) Set *Coefficients* to `[5.3, -494.0, 5.05, -318.0, 4.1, -30.3, 6.5, -394.0, 11.7, -918.0]`.

       This property specifies the values of the coefficients for each of the polynomials.
    e) Click **OK** to close the *Fan Curve Polynomial* dialog.

The *Data Rotation Rate* property specifies the speed at which the fan data was collected, in this case 800 rpm. The *Operating Rotation Rate* property specifies the speed at which you want the fan to run — in this case 1200 rpm.

19. Set the *Operating Rotation Rate* to `1200.0 rpm`.
20. Set the *Data Rotation Rate* to `800.0 rpm`.

The inlet and outlet fan are the same. Copy the fan data from the inlet fan to the outlet fan:

21. Right-click the **Inlet Fan** > **Fan Curve Polynomial** node and select **Copy**.

22. Right-click the **Outlet Fan** > **Fan Curve Polynomial** node and select **Paste**.

To have the correct thermal conductivity on either side of the interfaces, make sure that all remaining interfaces in the simulation are set to contact interfaces:

23. Multi-select all the interfaces, except for the fan interface, and make sure that *Type* is set to **Contact Interface**.
24. Save the simulation.

Setting up the Meshers

For this simulation, you generate a volume mesh using the polyhedral mesher for the bulk mesh, and the thin mesher for the graphics card.

To define the mesh operation:

1. Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
2. In the *Create Automated Mesh Operation* dialog:
   a) Select all parts from the *Parts* list.
   b) Select the following meshers, in order:

Group	Mesher
Surface Meshers	**Surface Remesher**
Core Volume Meshers	**Polyhedral Mesher**
Optional Boundary Layer Meshers	**Prism Layer Mesher**
Optional Volume Meshers	**Thin Mesher**

   c) Click **OK**.

Modify the mesher settings:

3. Expand the **Automated Mesh** > **Meshers** node.
4. Select the **Surface Remesher** node and activate the *Perform Compatibility Refinement* option.
   When activated, this option increases the number of faces based on the mesh size on nearby surfaces.
5. Edit the **Polyhedral Mesher** node and set the following parameters:

Property	Setting
Enable Growth Rate	**Activated**
Optimization Cycles	3
Quality Threshold	0.7

6. Edit the **Thin Mesher** node and set the following parameters:

Property	Setting
Automatic Thin/Bulk Shape Detection	**Activated**
Optimization Cycles	3

Property	Setting
Quality Threshold	`0.7`

Modify the default controls for the mesh:

7. Edit the **Operations** > **Automated Mesh** > **Default Controls** node and set the following parameters:

Node	Property	Setting
Base Size	*Value*	`0.02 m`
Prism Layer Total Thickness	*Percentage of Base*	`30.0`
Volume Growth Rate	*Volume Growth Rate*	`1.1`
Maximum Cell Size	*Percentage of Base*	`140.0`

8. Save the simulation.

Contents:

Creating the Custom Controls
Generating the Volume Mesh

Creating the Custom Controls

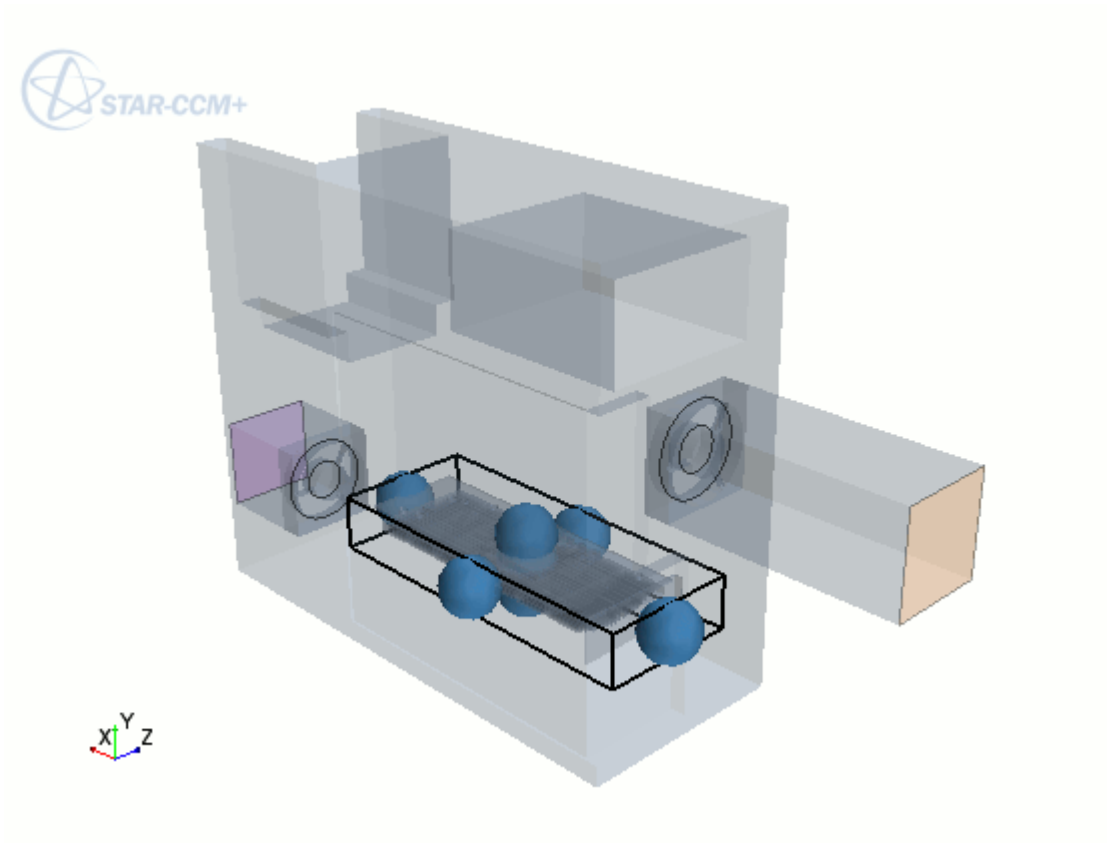
Use custom controls to modify the mesh on the graphics card and to disable the prism layers in the solid regions.

Create the volume shape to refine the volume mesh near the graphics card:

1. Right-click the **Geometry** > **Parts** node and select **New Shape Part** > **Block**.
2. In the *Create/Edit Block Part* in-place dialog, make the following settings in the *Maximum and Minimum Coordinates* box:

Property	Setting
Corner 1	`[-0.4, -0.15, -0.06] m`
Corner 2	`[-0.07, -0.1, 0.07] m`

A preview of the block shape is visible in the scene.



3. Click **Create** and then **Close**.

   The new part, **Block**, is added to the **Parts** node.

Continue by creating the volumetric control:

4. Right-click **Operations** > **Automated Mesh** > **Custom Controls** and select **New** > **Volumetric Control**.

   A new node, **Volumetric Control**, is added to the **Custom Controls** node.

5. Edit the **Volumetric Control** node and set the following properties:

Node	Property	Setting
Volumetric Control	*Parts*	**Block**
└ **Controls**		
└ **Prism Layer Mesher**	*Customize Total Thickness*	**Activated**
└ **Values**		
└ **Custom Prism Values** > **Prism Layer Total Thickness**	*Percentage of Base*	`10.0`

Create the custom control for the thin parts:

6. Right-click **Custom Controls** and select **New** > **Part Control**.

   A new node, **Part Control**, is added to the **Custom Controls** node.

7. Edit the **Part Control** node and set the following properties:

Node	Property	Setting
Part Control	*Parts*	**Capacitor 1** to **Capacitor 6**
		GPU Chip
		Heatsink Block
		Memory Chip 1 to **Memory Chip 8**
		PCB
		Plate
└ **Controls**		
└ **Thin Mesher**	*Thin Mesher*	**Custom**
└ **Customize**	*Customize Thickness Threshold*	**Activated**
└ **Values**		
└ **Custom Thin Values** > **Thin Mesher Maximum Thickness**	*Size Type*	**Absolute**
└ **Absolute Size**	*Value*	`4.0 mm`

Activating the customize thickness threshold option allows you to set a custom threshold value for the thin mesher. Any part that is thinner than this threshold is meshed using the thin mesher.

Create the custom control to disable the prism layers on the parts in the solid regions and customize the surface size:

8. Right-click **Custom Controls** and select **New** > **Surface Control**.

   A new node, **Surface Control**, is added to the **Custom Controls** node.

9. Edit the **Surface Control** node and set the following properties:

Node	Property	Setting
Surface Control	*Part Surfaces*	**Capacitor 1** to **Capacitor 6**
		GPU Chip
		Heatsink Block
		Inlet Fan Casing
		Memory Chip 1 to **Memory Chip 8**
		Outlet Fan Casing
		PCB
		Plate
		Port Block 1 to **Port Block 3**

Node	Property	Setting
└ **Controls**		
└ **Target Surface Size**	*Target Surface Size*	**Custom**
└ **Prims Layers**	*Prism Layers*	**Disable**
└ **Values**		
└ **Target Surface Size**	*Percentage of Base*	`25.0`

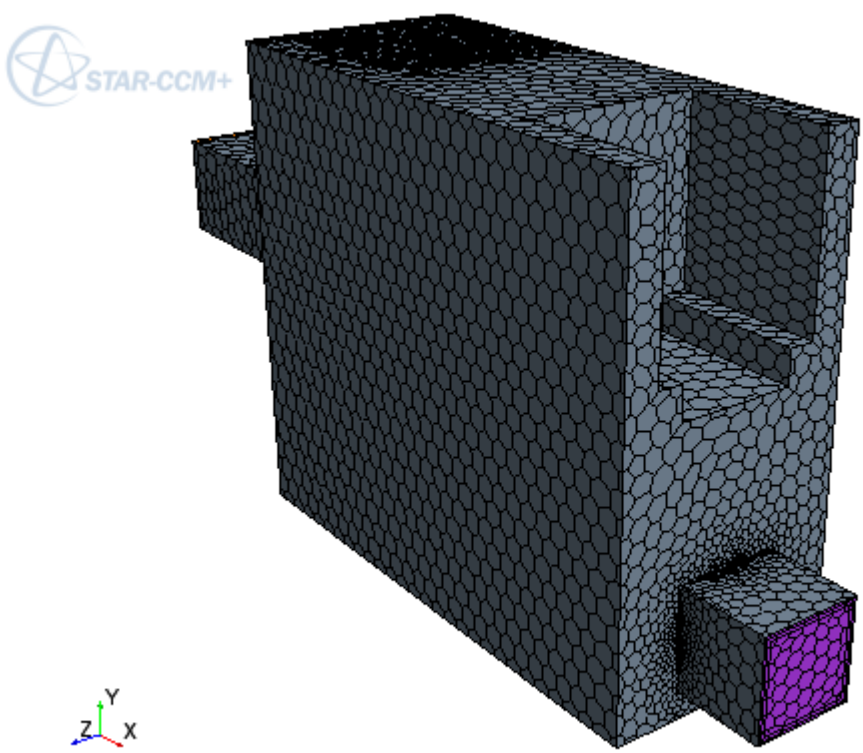
10. Save the simulation.

Generating the Volume Mesh

Generate and visualize a conformal mesh.

The volume mesh is ready to be generated.

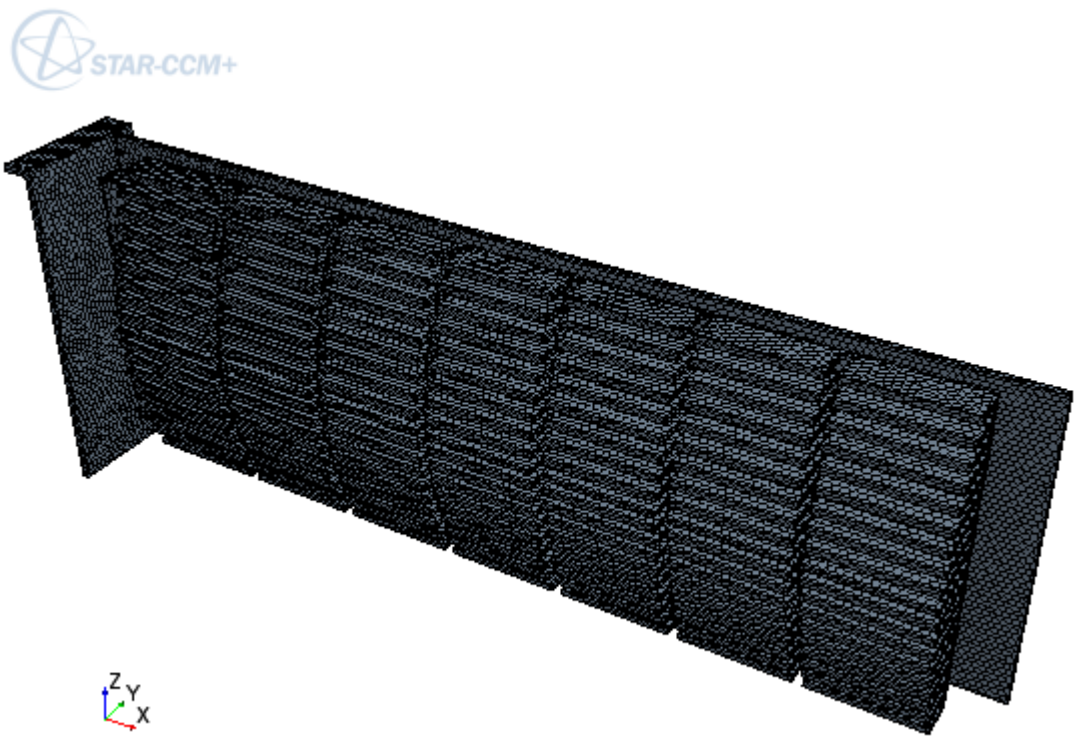
1. Click (**Generate Volume Mesh**) in the **Mesh** toolbar.
   A conformal mesh is generated between all regions.
2. To visualize the mesh, create a mesh scene and inspect the fluid domain mesh.



3. Edit the **Mesh Scene 1** > **Displayers** > **Mesh 1** > **Parts** node.

4. In the *Parts* dialog, deselect all regions and select all parts.
5. Expand the **Parts** node and deselect the following items:
   - **Air Case**
   - **Air Inlet**
   - **Air Outlet**
   - **Block**
   - **Inlet Fan Casing**
   - **Outlet Fan Casing**

   The mesh on the graphics card is shown in the following screenshot.



6. Save the simulation.

Preparing Scenes for Visualization

Create two scalar scenes: one to display the velocity magnitude on a plane cutting through the tower case, and the other to display the temperature on the graphics card.

Begin by creating the section plane:

1. Create a scalar scene.
2. Create a section plane with the following properties:

Property	Setting
Input Parts	**Parts** > **Air Case**
origin	`[0.0, 0.0, 0.035] m`
normal	`[0.0, 0.0, 1.0] m`
Display	**Existing Displayer** > **Scalar 1**

3. Select the **Scenes** > **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Scalar Field** node and set *Function* to **Velocity** > **Lab Reference Frame** > **Magnitude**.
4. Select the **Scalar 1** node and set *Contour Style* to `Smooth Filled`.

Set up the second scalar scene for displaying the temperature on the graphics card:

5. Create a scalar scene.
6. Edit the **Scenes** > **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Parts** node.
7. In the *Parts* dialog, expand the **Parts** node and select all items contained within.
8. In the same dialog, deselect the following items:
   - **Air Case**
   - **Air Inlet**
   - **Air Outlet**
   - **Block**
   - **Inlet Fan Casing**
   - **Outlet Fan Casing**
9. Select the **Scalar 1** > **Scalar Field** node and set the following properties:

Property	Setting
Function	`Temperature`
Units	`C`

Reporting Temperature with a Monitor and a Plot

Create a monitor and plot of maximum temperature to help you judge the rate at which the solution is converging.

The maximum temperature in the simulation is plotted. This plot is useful for assessing solution convergence.

To monitor the temperature:

1. Create a **Maximum** report.
2. Rename the new report to `Maximum Temperature`.
3. Select the **Reports** > **Maximum Temperature** node and set the following properties:

Property	Setting
Parts	**Regions** > **Solid Components**
Field Function	**Temperature**

Property	Setting
Units	**C**

4. Right-click the **Maximum Temperature** node and select **Create Monitor and Plot from Report**.

   New monitor and plot nodes are added to the **Monitors** and **Plots** nodes respectively.

Modifying Solver Settings and Stopping Criteria

Modify the under-relaxation factors for the Segregated Energy solvers to help accelerate convergence.

Use the following steps:

1. Edit the **Solvers** node and set the following properties:

Node	Property	Setting
Segregated Energy	*Fluid Under-Relaxation Factor*	`0.99`
	Solid Under-Relaxation Factor	`0.9999`

Modify the maximum number of iterations for the solver:

2. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to `500`.
3. Save the simulation.

Running the Simulation

The simulation is now ready to run.

To run the simulation:

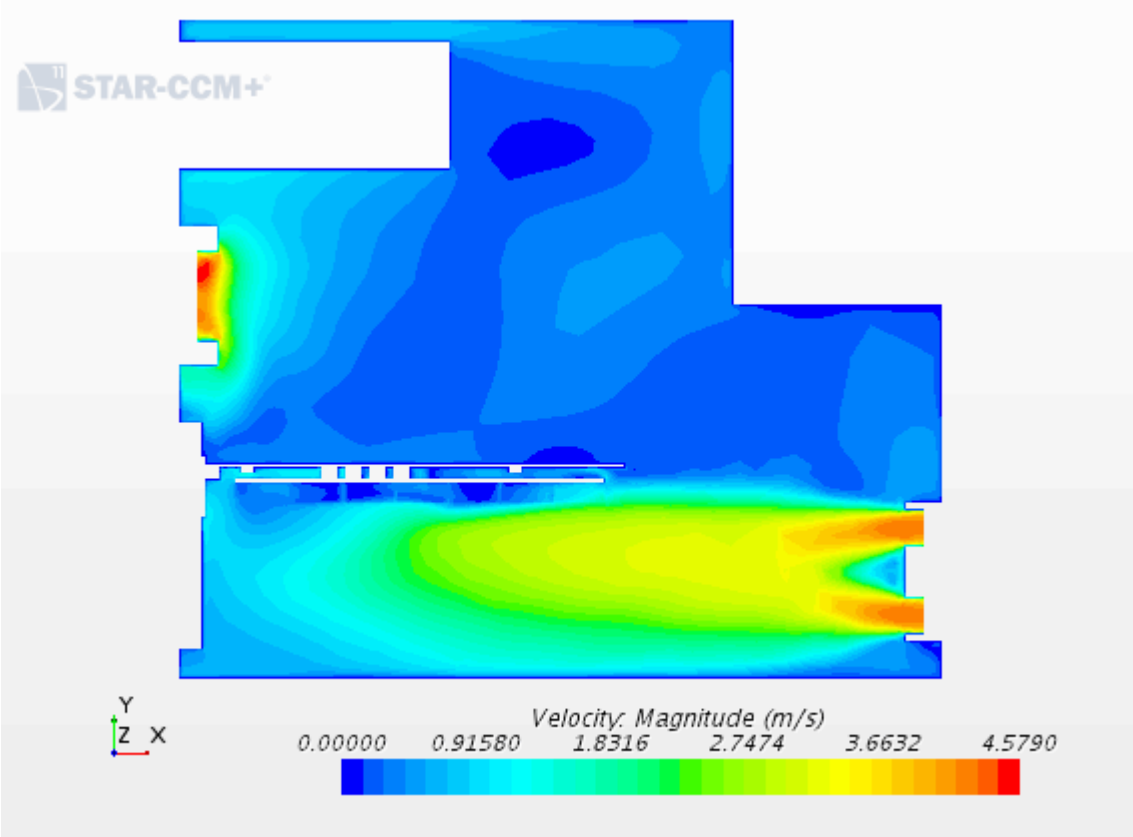
1. Click (**Run**) in the *Solution* toolbar.
2. While the simulation is running, select one of the scalar scenes to visualize the solution.
3. When the simulation has finished running, save it.

Visualizing the Results

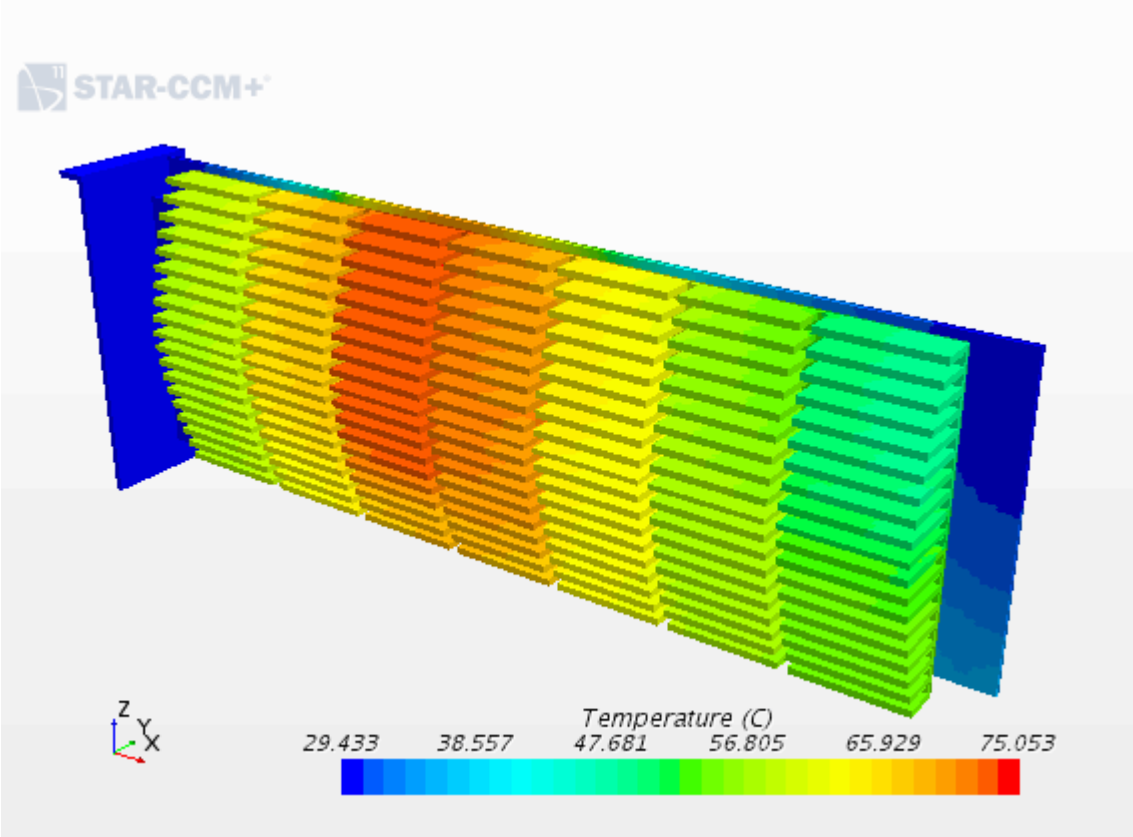
View the results of the solution.

1. After the run is finished, look at the scalar and plot displays.

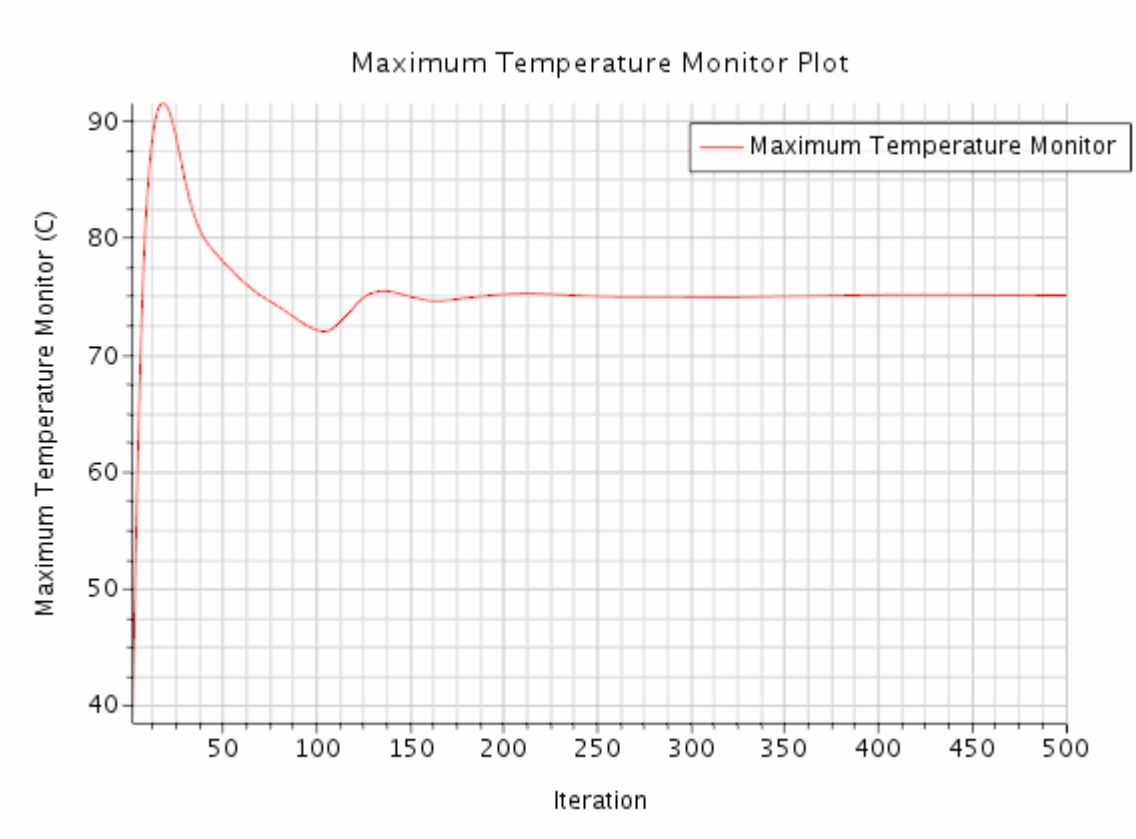
   The following screenshot illustrates air flow inside the case.



The temperature on the graphics card is shown in the following screenshot.



The temperature monitor plot is displayed in the following screenshot. From this plot, you can see that the solution has converged.



Summary

The Multi-Part Solid tutorial used simplified geometry of a PC tower and graphics card to introduce various STAR-CCM+ features.

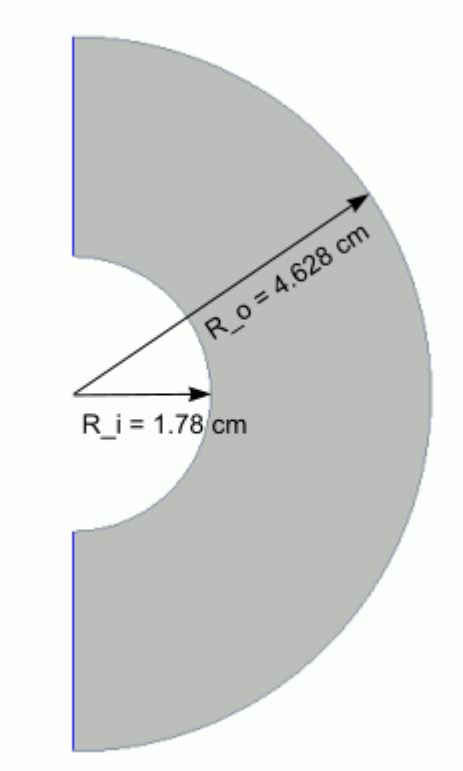
The features that were introduced include:

- Importing a CAD geometry as Parts.
- Using the Multi-Part Solid material model to model multiple solids with different materials.
- Using Parts-Based Profiles to apply different heat sources to different parts in the same region.
- Setting up fan interfaces to model the inlet and outlet fans in the PC tower case.

Natural Convection: Concentric Cylinders Introduction

This tutorial demonstrates how to solve a natural convection problem in STAR-CCM+.

An infinitely long cylinder of radius 1.78 cm, heated at a constant temperature of 306.3 K, is placed inside another larger cylinder of radius 4.628 cm at a lower fixed temperature of 293.7 K. These temperatures are chosen to give an even temperature distribution about a bulk temperature (T_b) of 300 K in the annulus. As the flow is expected to be symmetrical about the center-line, it is only necessary to work with half the geometry, as shown below.



Set up the model for an ideal gas with the following material properties, evaluated at bulk temperature of 300 K and pressure of 100,000 Pa.

Property	Setting
Density (kg/m^2)	1.1614
Dynamic viscosity (Pa-s)	1.846E-5
Kinematic viscosity (m^2/s)	1.589E-5
Specific Heat (J/kg-K)	1007.0
Thermal Coefficient of Volumetric Expansion (1/K)	1/T_b (Ideal Gas) = 0.00333
Thermal Conductivity (W/m-K)	0.0263

Property	Setting
Thermal Diffusivity (m^2/s)	2.249E-5

This case repeats the experiments of Kuehn and Goldstein [416]. Set up the physics to correspond to a Rayleigh number of 2.66 X 10^4.

Contents:

Prerequisites
Importing the Mesh and Naming the Simulation
Setting Up the Physics Models
Setting Boundary Physics Conditions
Setting Solver Criteria
Preparing Scalar and Vector Scenes
Preparing Reports to Monitor Convergence
Running the Simulation
Visualizing the Results
Comparing Results to Experimental Data
Summary
References

Prerequisites

The instructions in the Natural Convection tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Importing the Mesh and Naming the Simulation

To set up the STAR-CCM+ simulation, launch a simulation and import the supplied volume mesh.

Use the following steps:

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh**.
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/heatTransferAndRadiation/data`.
5. Select `concylMesh.ccm` then click **Open**.

STAR-CCM+ provides feedback on the import process in the *Output* window and a geometry scene is displayed in the *Graphics* window.

6. Save the simulation as `conCyl.sim`.

Setting Up the Physics Models

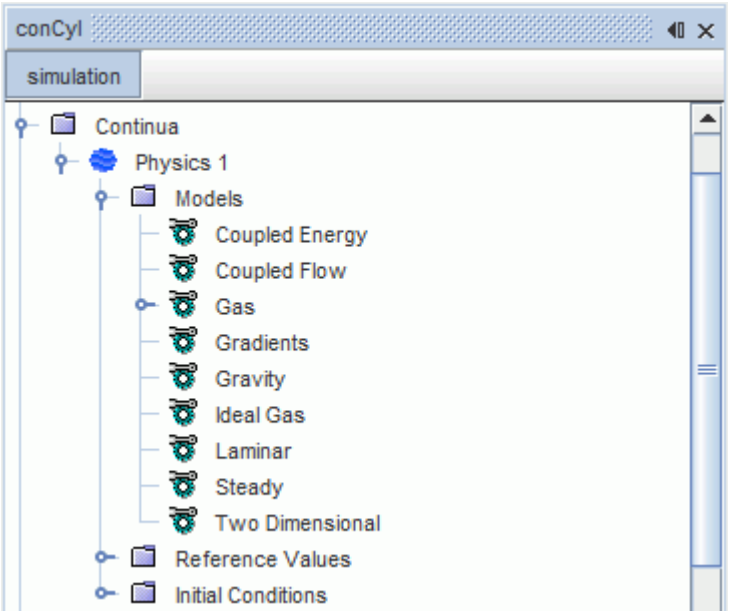
Set up the physics using an implicit coupled flow solver due to its increased robustness over the segregated solver.

Use the following steps:

1. For the physics continuum, **Continua** > **Physics 1**, select the following models in order:

Group Box	Model
Enabled Models	**Two Dimensional** (Selected automatically)
Time	**Steady**
Material	**Gas**
Flow	**Coupled Flow**
Gradient Metrics	**Gradients** (Selected automatically)
Equation of State	**Ideal Gas**
Energy	**Coupled Energy** (Selected automatically)
Viscous Regime	**Laminar**
Optional Models	**Gravity**

2. Click **Close**.
3. To review the models, open the **Physics 1** > **Models** node.



4. Expand the **Continua** > **Physics 1** > **Models** > **Gas** > **Air** node.
5. Edit the **Material Properties** node and set the following properties:

Node	Property	Setting
Dynamic Viscosity (Pa-s)		
Constant	*Value*	`1.846E-5`
Specific Heat (J/kg-k)		
Constant	*Value*	`1007.0`
Thermal Conductivity (W/m-K)		
Constant	*Value*	`0.0263`

6. Click **Close**.
7. Edit the **Continua** > **Physics 1** > **Reference Values** node and set the following properties:

Node	Property	Setting
Reference Density (kg/m^3)	*Value*	`1.1614`
Reference Pressure (Pa)	*Value*	`100000.0`

Note: The initial static temperature is by default 300 K, which is the average of the fixed inner and outer wall temperatures. The initial static temperature is located in the **Continua** > **Physics 1** > **Initial Conditions** > **Static Temperature** node. The material properties were evaluated at this bulk temperature.

Setting Boundary Physics Conditions

A fixed temperature difference across the annulus is achieved by setting the static temperatures of the inner and outer cylinder walls.

Use the following steps:

1. Expand the **Regions** > **ConvectionCylinders** node.
2. Edit the **Boundaries** node and set the following properties:

Node	Property	Setting
Inner Wall		
Physics Conditions		
Thermal Specification	*Condition*	`Temperature`

Node	Property	Setting
Physics Values		
Static Temperature	*Value*	`306.3 K`
Outer Wall		
Physics Conditions		
Thermal Specification	*Condition*	`Temperature`
Physics Values		
Static Temperature	*Value*	`293.7 K`

Setting Solver Criteria

Increase the default value of the Courant number of the coupled implicit solver to achieve faster convergence.

You generally want a Courant number that balances the total number of iterations against the time per iteration.

1. Select the **Solvers** > **Coupled Implicit** node and set *Courant Number* to `100.0`.

Because you are using a relatively large Courant number on a simple convection problem, you can expect fast convergence. Therefore decrease the number of iterations for which to run the solver.

2. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to `300`.

Preparing Scalar and Vector Scenes

A scalar and a vector scene are created to display the temperature and velocity vectors in the region bounded by the concentric cylinders during the simulation.

Use the following steps:

1. Create a scalar scene.
2. Expand the **Scalar Scene 1** > **Displayers** > **Scalar 1** node.
3. Select the **Scalar Field** node and set *Function* to `Temperature`.
4. Create a vector scene.

   By default, *Vector Field* is set to `Velocity`.

Preparing Reports to Monitor Convergence

To judge convergence for this natural convection problem, monitor the heat balance in your model, which sums to zero for a converged case.

Prepare a report for the inner wall.

1. Create a **Heat Transfer** report.
2. Rename the new report to `Heat Transfer Inner`.
3. Edit the *Parts* property.
4. In the *Heat Transfer Inner - Parts* dialog, select **Regions** > **ConvectionCylinders** > **Inner Wall**.
5. Click **OK**.

Prepare a report for the outer wall.

6. Create another **Heat Transfer** report.
7. Rename this report to `Heat Transfer Outer`.
8. Edit the *Parts* property.
9. In the *Heat Transfer Outer - Parts* dialog, select **Regions** > **ConvectionCylinders** > **Outer Wall**.
10. Click **OK**.

Prepare an expression report that calculates the heat balance using the results of the reports. A corresponding field function is generated for each report, allowing you to use them in an expression.

11. Create an **Expression** report.
12. Rename this report to `Heat Balance`.
13. Click ... (**Custom Editor**) in the *Definition* property.
14. In the *Heat Balance - Definition* dialog, enter `${HeatTransferOuterReport} + ${HeatTransferInnerReport}`.
15. Click **OK**.
16. Right-click the **Heat Balance** node and select **Create Monitor and Plot from Report**.

    New monitor and plot nodes are added to the **Monitors** and **Plots** nodes respectively.
17. Open the **Heat Balance Monitor Plot**.

Running the Simulation

The simulation is now ready to run.

Use the following steps:

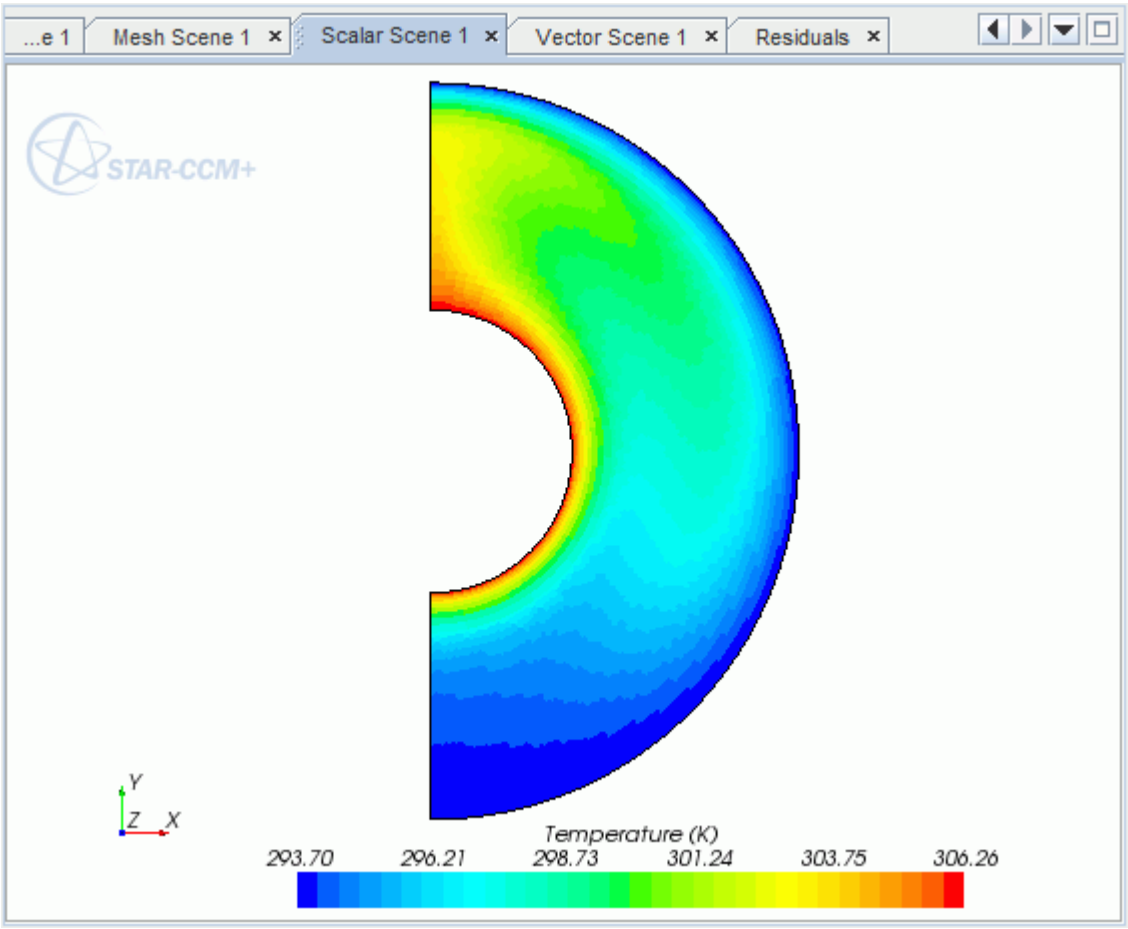
1. Run the simulation.
2. To watch the solution being updated while the simulation is running, look at the scalar scene and vector scene displays.
3. When the simulation has finished running, save it.

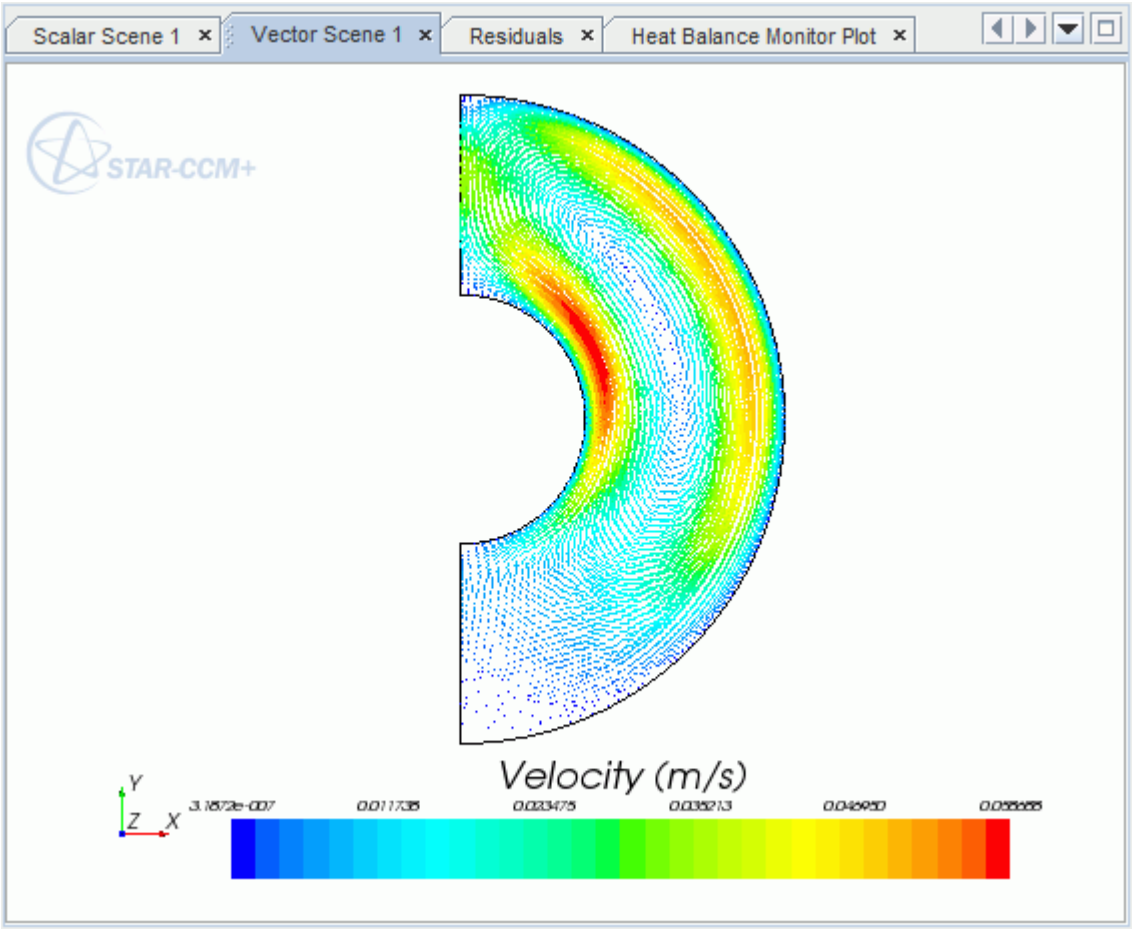
Visualizing the Results

The scalar and vector scenes contain the solution results.

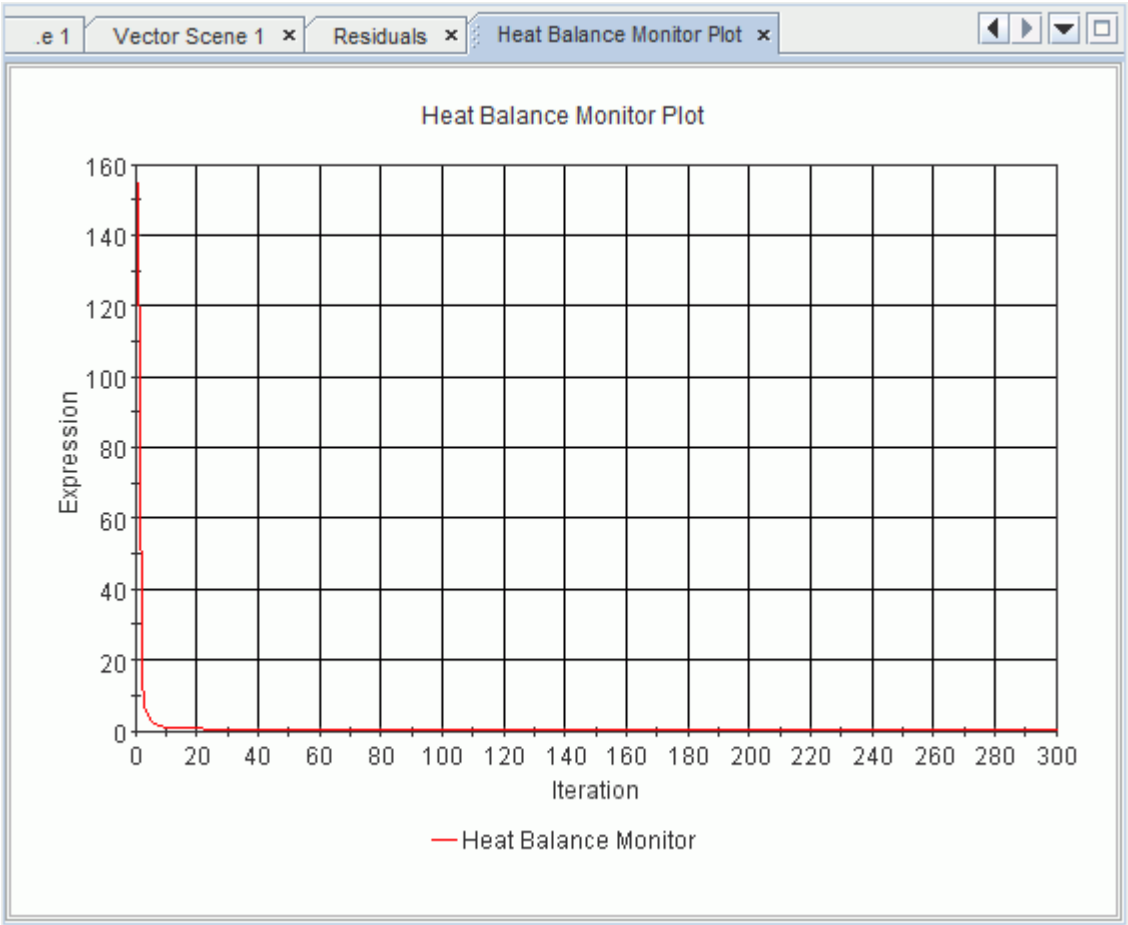
As expected, the temperature decreases between the wall of the inner cylinder and that of the outer cylinder. This decrease corresponds to the difference in the fixed wall temperatures of the two cylinders. The warmer air rises as it moves from the wall of the inner cylinder towards the outer cylinder, and sinks to the bottom of the annulus as it cools. The air recirculates within the annulus due to natural convection, with the largest velocities seen near the walls of the cylinders, towards the top.

1. Look at these results in the displays.





The heat balance quickly drops to about zero, showing that the problem converged correctly.



Comparing Results to Experimental Data

This part of the tutorial compares results and data using equivalent conductivity.

Kuehn and Goldstein reported their experimental and numerical results in the form of an equivalent conductivity. This equivalent conductivity is defined as the ratio of the actual heat transfer measured (or predicted) across the annulus to the heat transfer that would be obtained by pure conduction alone. That is:

$$k_{eq} = \frac{q_{act}}{q_{cond}} \tag{4160}$$

For concentric cylinders, q_{cond} is defined as:

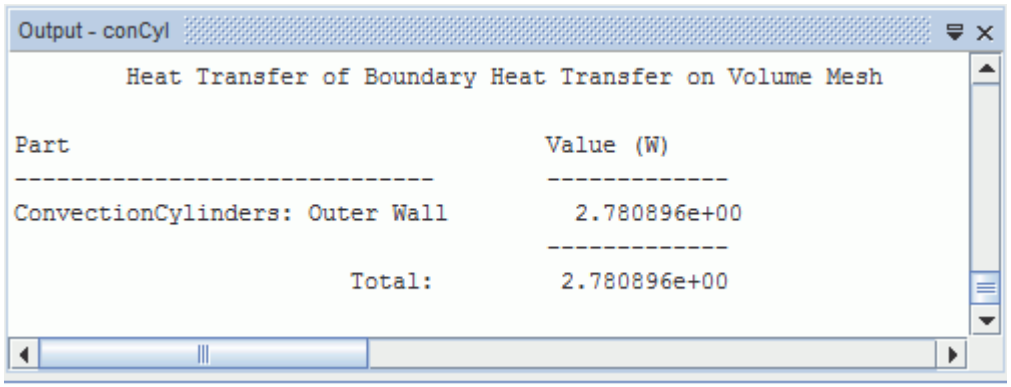
$$q_{cond} = \frac{2\pi k \Delta T}{\ln(R_o/R_i)} \tag{4161}$$

For the case being considered in this tutorial, Kuehn and Goldstein obtained an equivalent conductivity of 2.58.

1. To begin evaluating conductivity, calculate the conductive heat transfer based on Eqn. (4161) using values provided earlier in your setup. This comes out to $q_{cond} = 2.177 \ \ W/m$.

2. To retrieve the value of q_{act} for one half of the cylinders, double-click the **Heat Transfer Inner** or **Heat Transfer Outer** node.

   The heat transfer appears in the *Output* window.



The total q_{act} for the complete annulus is therefore 5.56 W.

Inserting this value into Eqn. (4160) above calculates an equivalent conductivity of:

$$k_{eq} = \frac{5.56}{2.177} = 2.55 \tag{4162}$$

There is only about a 1% difference between this result and the value given by Kuehn and Goldstein, showing that the numerical model has predicted the heat transfer consistent to what has been reported by others.

Note: Because you are working with a 2D representation of the cylinders, the prediction of heat transfer is per meter length, that is, W/m.

Summary

The Natural Convection tutorial used two concentric cylinders to introduce a variety of STAR-CCM+ features.

The features that were introduced include:

- Importing a volume mesh.
- Defining physics models for laminar flow with natural convection.
- Defining the material properties for the selected models.
- Defining fixed physics conditions at the regions level.
- Setting a convergence condition for the coupled implicit solver.
- Preparing scalar and vector scenes.
- Preparing a heat balance report.
- Running the simulation.
- Examining the temperature, velocity field and heat balance solutions.
- Comparing solution to experimental results.

References

This tutorial repeats the experiments of Kuehn and Goldstein (1978).

[789] *Kuehn, T. H and Goldstein, R. J. An experimental study of natural convection heat transfer in concentric and eccentric horizontal cylindrical annuli. J. Heat Trans. (100) pp. 635-640, 1978.*

Electronics Cooling Toolset: Natural Convection JEDEC

Although many electronic devices benefit from cooling fans in removing excess heat, some devices rely on natural convection only. The Electronics Cooling Toolset includes a method for estimating cooling due to natural convection.

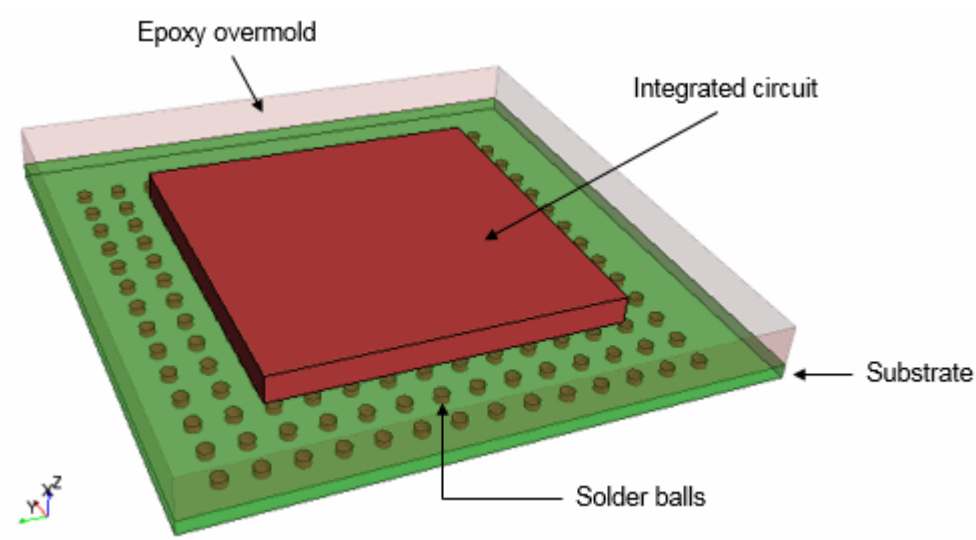
The simulation models the JEDEC standard test method for integrated circuits in natural convection [790]. The purpose of this test method is to measure the junction-to-ambient (θ_{JA}) thermal resistance of a chip package in a standardized still air environment. θ_{JA} allows a thermal performance comparison between chip packages and is calculated as:

$$\theta_{JA} = \frac{T_J - T_A}{P_H}$$

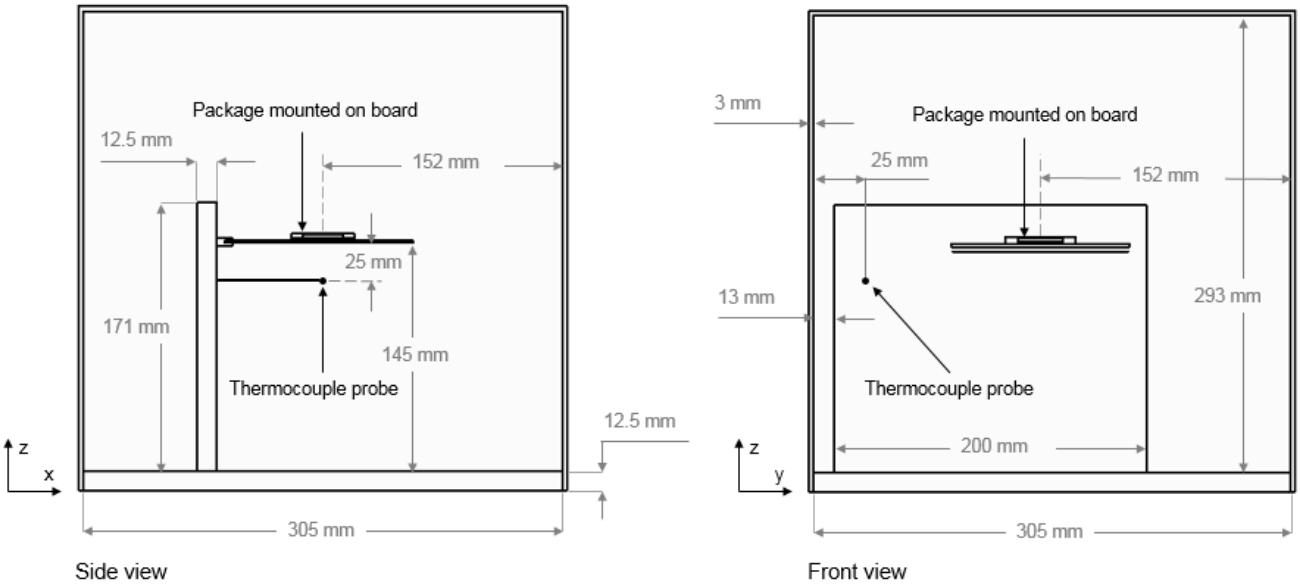
where:

- T_J is the junction temperature of the chip package.
- T_A is the ambient temperature.
- P_H is the power dissipation of the integrated circuit.

In this tutorial, the test package is a generic BGA (ball grid array) package, depicted as follows:



The test standard arranges the chip package on a test board in the center of a closed chamber:



In this tutorial, you build the complete geometry of the test rig, the chip package, and the chamber using templated QuickParts. The analysis includes the following heat transfer modes:

- Conduction in the solid components of the test rig, the chip package, and the chamber.
- Diffusion, natural convection, and radiation in the enclosed air.

To conclude the analysis, you display the simulation results using the post-processing capabilities that are available in the Electronics Cooling Toolset.

Contents:

Prerequisites
Launching the Electronics Cooling Toolset and Naming the Simulation
Defining the Setup
Creating and Positioning the Components
Extracting the Air Volume
Generating the Mesh
Preparing Scenes for Visualization
Preparing Temperature Report
Running the Simulation
Analyzing the Results
Summary
Bibliography

Prerequisites

This tutorial guides you through the setup of the natural convection JEDEC simulation using the dedicated Electronics Cooling Toolset in STAR-CCM+.

Before attempting this tutorial, make sure that the (**Launch Electronics Cooling App**) button is visible in the STAR-CCM+ toolbar. For information on how to include the button in the toolbar, see Accessing the Electronics Cooling Toolset.

Launching the Electronics Cooling Toolset and Naming the Simulation

Launch the Electronics Cooling Toolset from within STAR-CCM+ and save the simulation.

To launch the Electronics Cooling Toolset:

1. Start up STAR-CCM+ and create a new simulation by clicking **File** > **New Simulation**.
2. From the STAR-CCM+ toolbar, click (**Launch Electronics Cooling App**).

   The *Electronics Cooling* tab opens within STAR-CCM+.

To save the simulation:

3. Select **File** > **Save As...** and save the simulation under the name `eCoolToolset_JedecTutorial.sim`

Defining the Setup

Set up the solution physics and materials for the solid components and the flow domain. For the length dimension, change the default unit.

To define the solution physics:

1. Right-click the **Setup** > **Physics** node and select **Edit**.

   An additional *Edit* tab opens that displays the default ambient conditions, solution physics, and materials.
2. Within the *Solution Physics* group box, make sure that **Solve** is set to **Natural Convection**.

When solving for a natural convection problem, the Electronics Cooling Module by default solves for flow and energy in a turbulent flow regime including gravity and radiation effects. For the given geometry and temperature differences in this tutorial, you can expect a laminar flow:

3. Set **Flow Regime** to **Laminar**.

Within the *Materials* group box, define the materials of the chip package and the test device:

- Silicon for the integrated circuit, an alloy of tin (Sn) and lead (Pb) for the solder balls, and an epoxy molding compound for the case of the BGA package
- A material of low thermal conductivity for the table and support fixture, here polypropylene.

To specify the various solid materials:

4. Next to the **Solid Sim Materials** box, click **Select**.
5. In the *Select Materials* dialog, click **Add Material**.
6. From the database of standard solid materials, select **Si (Silicon)** and click **OK**.
7. In the list of **Solid Simulation Materials**, select **Si** and click **Duplicate Material**.
8. Select the additional material and set the following properties:

Property	Setting
Material Name	`SnPb`
Thermal Conductivity	`51 W/m-K`
Specific Heat	`180 J/kg-K`
Density	`9250 kg/m^3`

9. Repeat steps 7 and 8 to create the remaining solid materials using the following properties:

Property	Setting
Material Name	`Epoxy`
Thermal Conductivity	`0.4 W/m-K`
Specific Heat	`1100 J/kg-K`
Density	`1250 kg/m^3`

Property	Setting
Material Name	`Polypropylene`
Thermal Conductivity	`0.23 W/m-K`
Specific Heat	`1920 J/kg-K`
Density	`950 kg/m^3`

10. Click **Apply** and **Close**.

To set polypropylene as the default material for a new solid component in the simulation:

11. Click the **Default Solid** box and select **Polypropylene**.
12. Finish the physics setup by clicking **Close** at the bottom of the tab.

To simplify geometry creation, set the default length unit to mm:

13. Right-click **Setup** > **Units** and select **Edit**.
14. For the **Length** dimension, set **Preferred Units** to **mm**.
15. Click **Close**.

    The physics and default units for the natural convection JEDEC simulation are now complete.

Save the simulation:

16. In the *System* toolbar (see Toolbar Reference), click (**Save**).

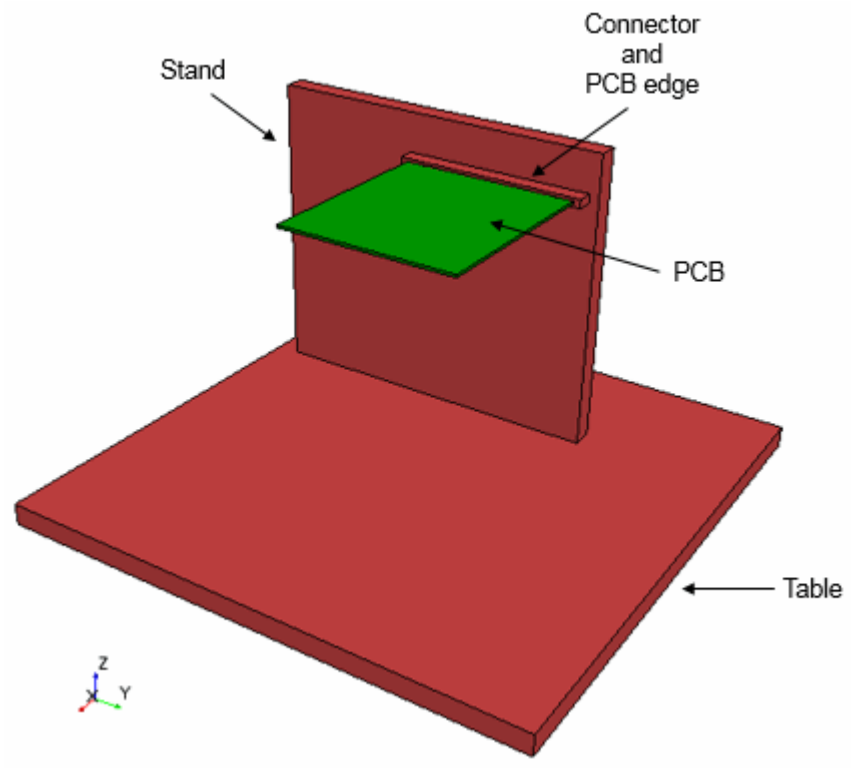
Creating and Positioning the Components

Create and place the components of the test rig, the chip package, and the chamber using templated QuickParts.

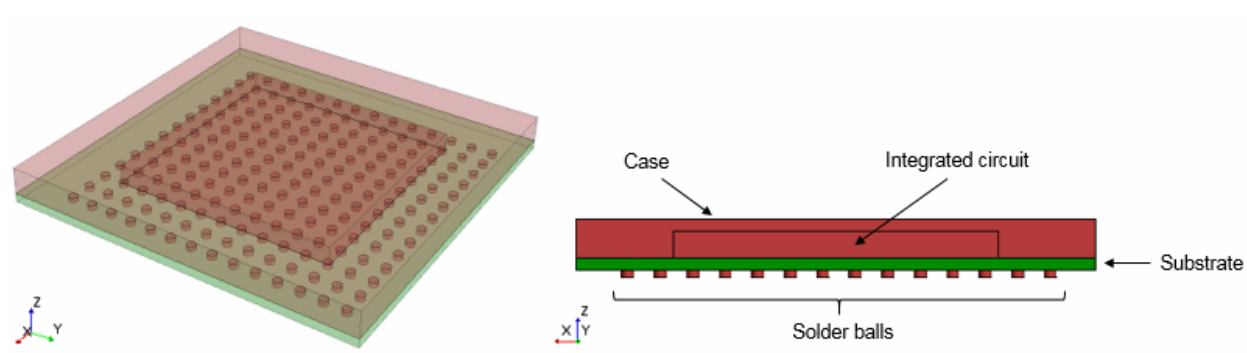
Within the Electronics Cooling Toolset, a QuickPart allows you to define the geometry, placement, and physics of a component in one step. For more information, see QuickParts.

The next steps describe the setup of the following components:

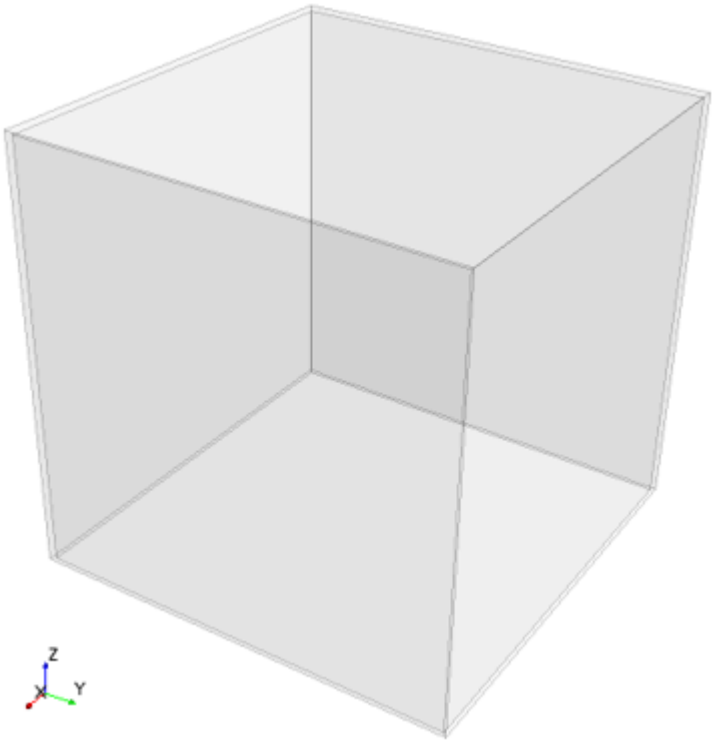
- Test rig



- Chip package



- Chamber



Contents:

Test Rig
Chip Package
Chamber

Test Rig

Define the geometry, placement, and physics of the JEDEC test rig using Generic and PCB QuickParts.

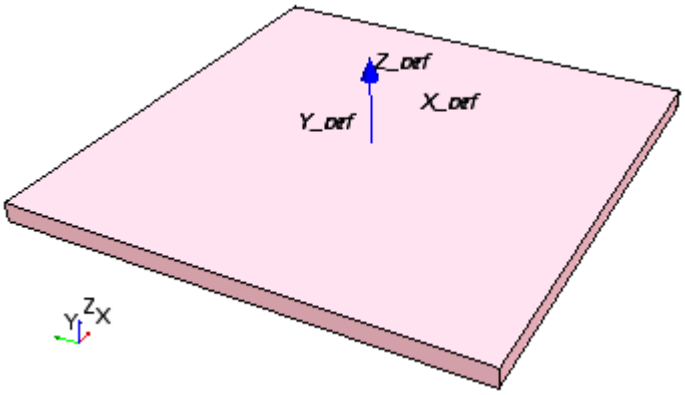
To define the table:

1. Right-click the **QuickParts** node and select **New QuickPart** > **Generic** > **Block**.
2. In the *Edit* dialog, set **Name** to `Table`.
3. Set the following *Geometry* properties:

Property	Setting
X	`305.0 mm`
Y	`305.0 mm`
Z	`12.5 mm`

4. To preview the block geometry in the *Graphics* window, click **Update** at the bottom of the dialog. Change the view to obtain a similar table position to the one shown below.

   For information on how to change the view in the *Graphics* window, see Mouse Configurations and Changing the View.



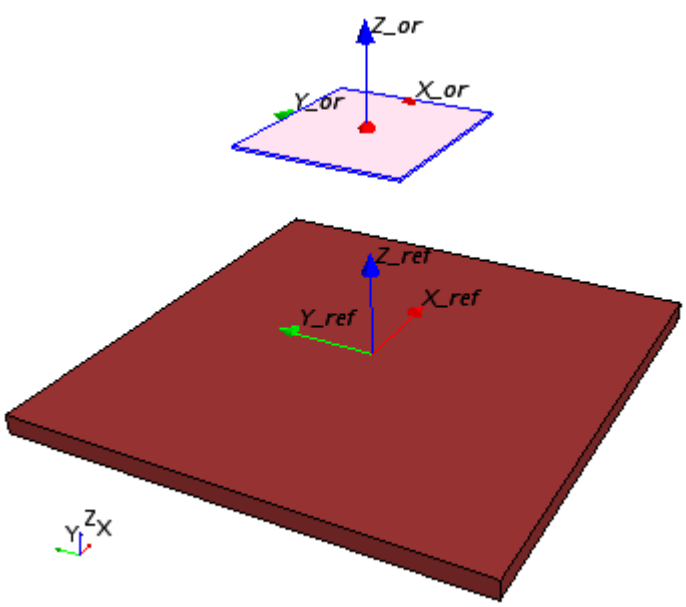
5. Click **OK**.

To define the PCB test board:

6. In the *Graphics* window, right-click the top surface of the table and select **New QuickPart** > **PCB** > **Rectangular**.
7. In the *Edit* dialog, set the following *Geometry* properties:

Property	Setting
X	`114.5 mm`
Y	`101.5 mm`

8. To define a centered *Placement* of the PCB above the table, set the **Location** vector to `[0.0, 0.0, 145.0] mm`.
9. Click **Update**.



To account for the effect of conductive tracks and non-conductive substrate on the heat transfer within the PCB, set the following *Physics* properties:

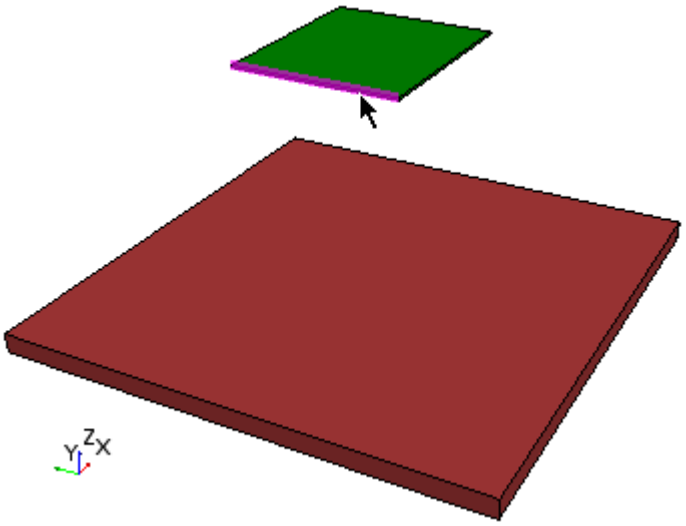
10. Set the **Thermal Specification** to **Specify By Layer**.
11. Set the metal fractions of the layers to the following values:

Name	Metal Fraction
Layer 1	`0.9`
Layer 2	`0.01`
Layer 3	`0.9`

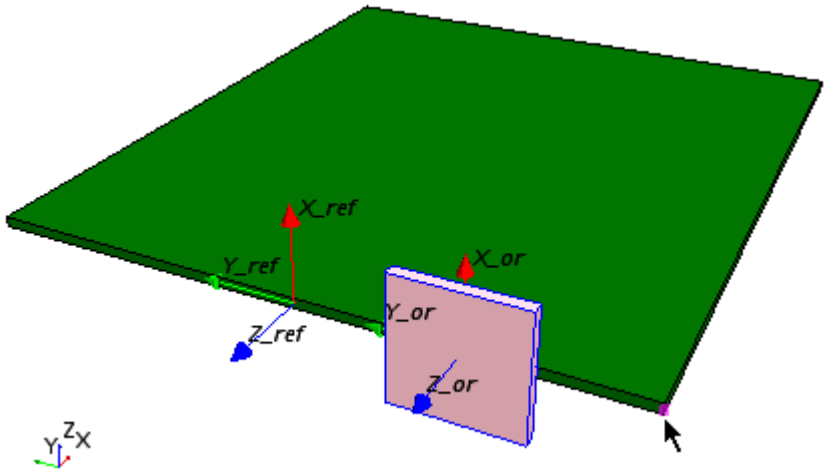
12. Click **OK**.

To define the PCB-edge:

13. In the *Graphics* window, right-click the PCB face that is highlighted below (note the coordinate system) and select **New QuickPart** > **Generic** > **Block**.



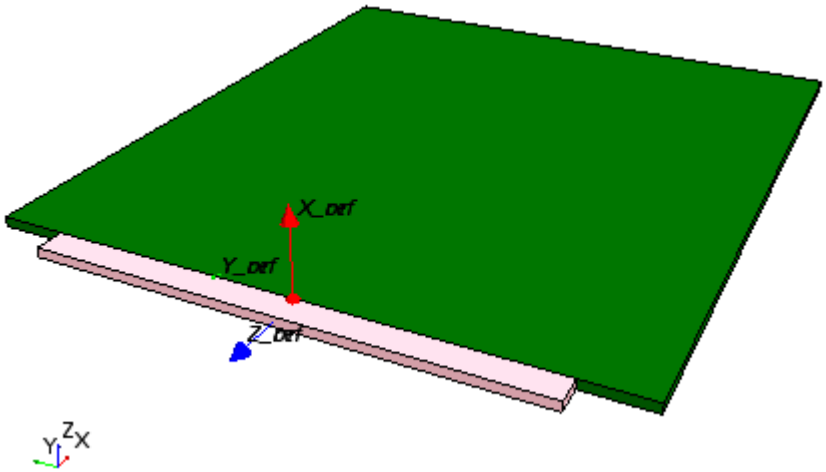
14. In the *Edit* dialog, set **Name** to `PCB-edge`.
15. To set the height of the block based on the thickness of the PCB, right-click one of the short edges of the PCB as highlighted below and select **Set Dimension...** > **X**.



16. Set the following *Geometry* properties:

Property	Setting
Y	`80.0 mm`
Z	`5.0 mm`

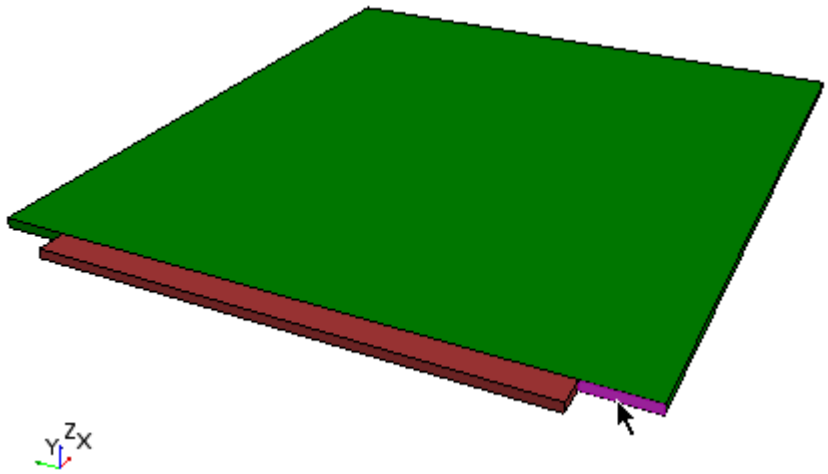
17. Set the **Location** vector to `[0.0, 0.0, 0.0] mm`.
18. Click **Update**.



19. Within the *Physics* section, click the **Solid** material box and select **Al**.
20. Click **OK**.

To define the connector:

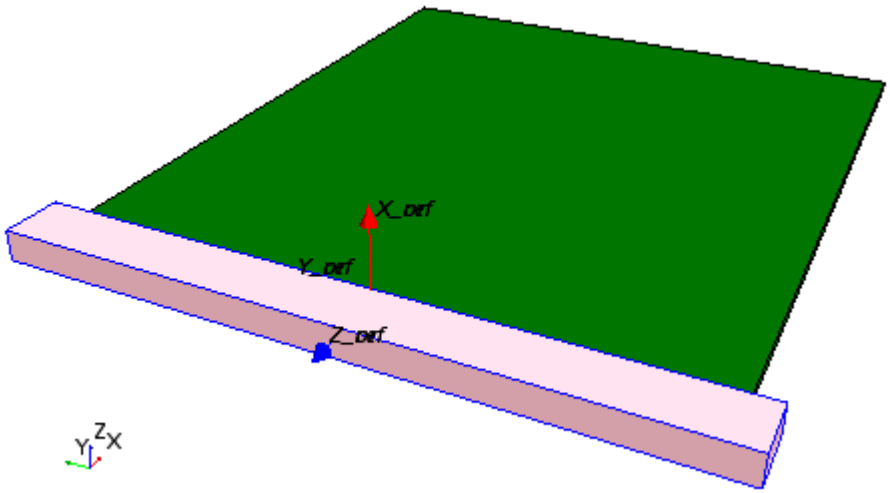
21. On the PCB, right-click the face that is highlighted below and select **New QuickPart** > **Generic** > **Block**.



22. In the *Edit* dialog, set **Name** to `Connector`.
23. Set the following *Geometry* properties:

Property	Setting
X	`5.0 mm`
Y	`110.0 mm`
Z	`10.0 mm`

24. Set the **Location** vector to `[0.0, 0.0, 0.0] mm`.
25. Click **Update**.



26. Click **OK**.

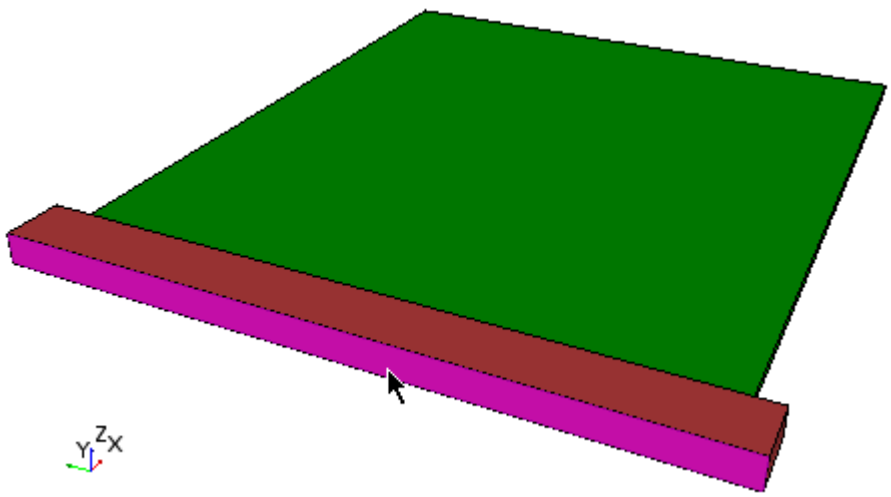
Subtract the PCB-edge from the Connector:

27. Right-click the **QuickParts** > **Connector** node and select **Edit Interactions**.
28. In the *Subtract* group box, click the **Solid Quick Objects** box and select **PCB-edge**.

29. Click **Apply** and **Close**.

To define the stand:

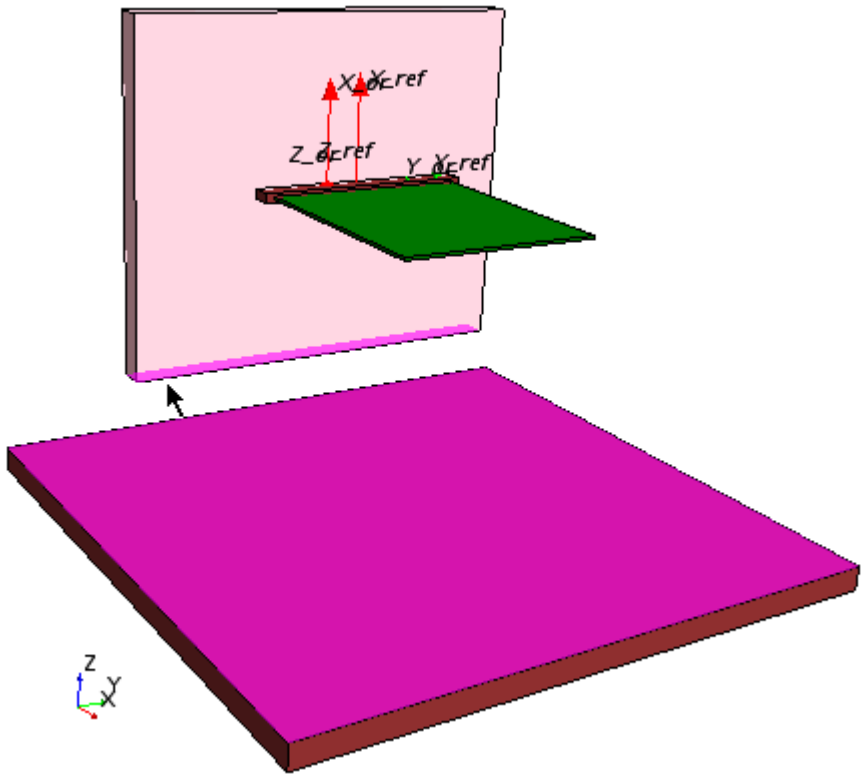
30. On the connector, right-click the face that is highlighted below and select **New QuickPart** > **Generic** > **Block**.



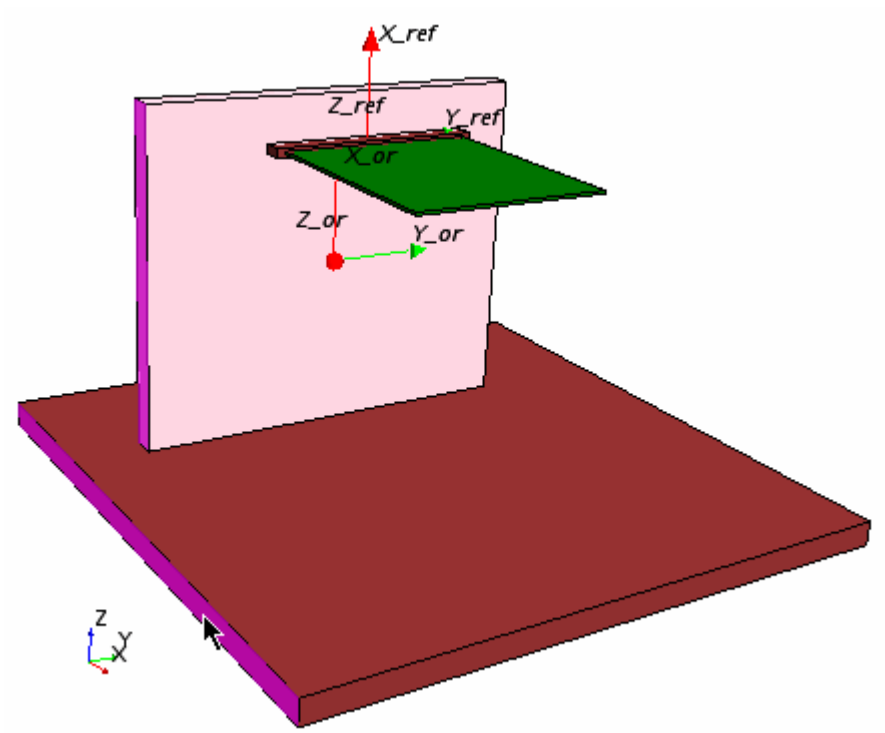
31. In the *Edit* dialog, set **Name** to `Stand`.
32. Set the following *Geometry* properties:

Property	Setting
X	`171.0 mm`
Y	`200.0 mm`
Z	`12.5 mm`

33. Click **Update**.
34. Using `<Ctrl>`, multi-select the top surface of the table and the bottom surface of the stand. Right-click and select **Position...** > **Snap**.

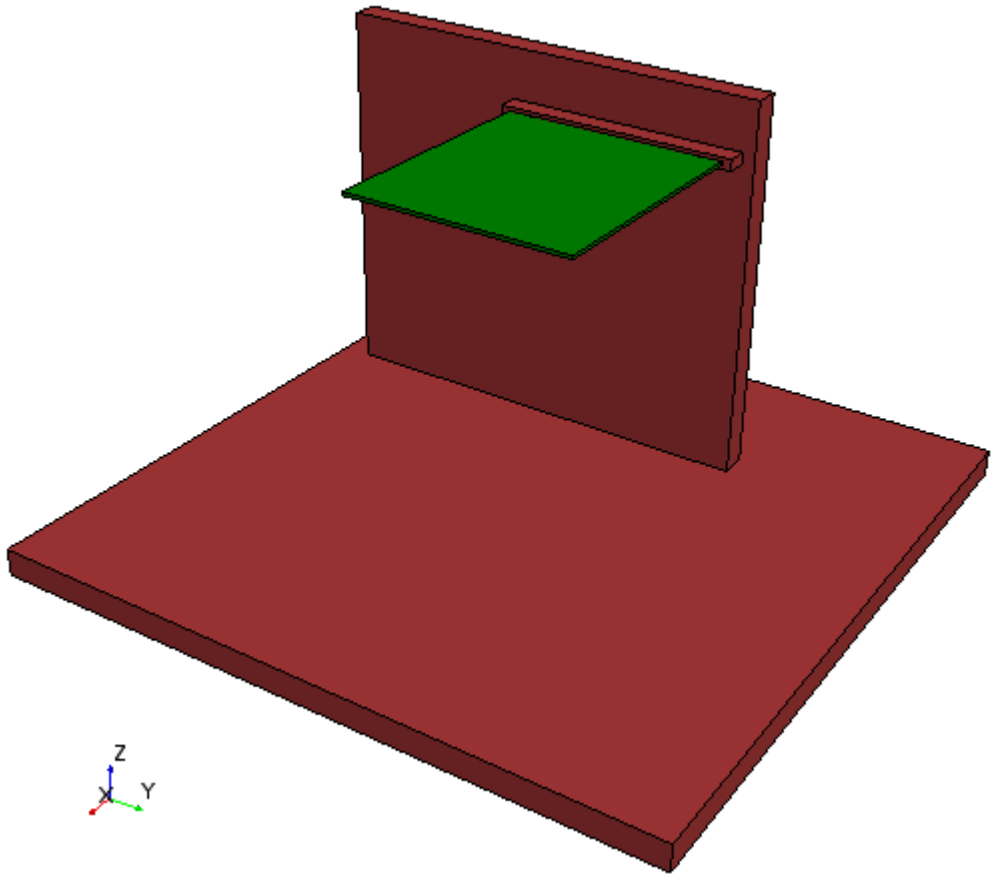


35. Multi-select the two faces that are highlighted below, right-click and select **Position...** > **Set Distance**.



36. In the *Set Distance* dialog, set **Distance** to `13.0 mm` and click **OK**.
37. Click **OK**.

The setup of the test rig is now complete.



38. Save the simulation .

Chip Package

Define the geometry, placement, and physics of the chip package using Generic and PCB QuickParts.

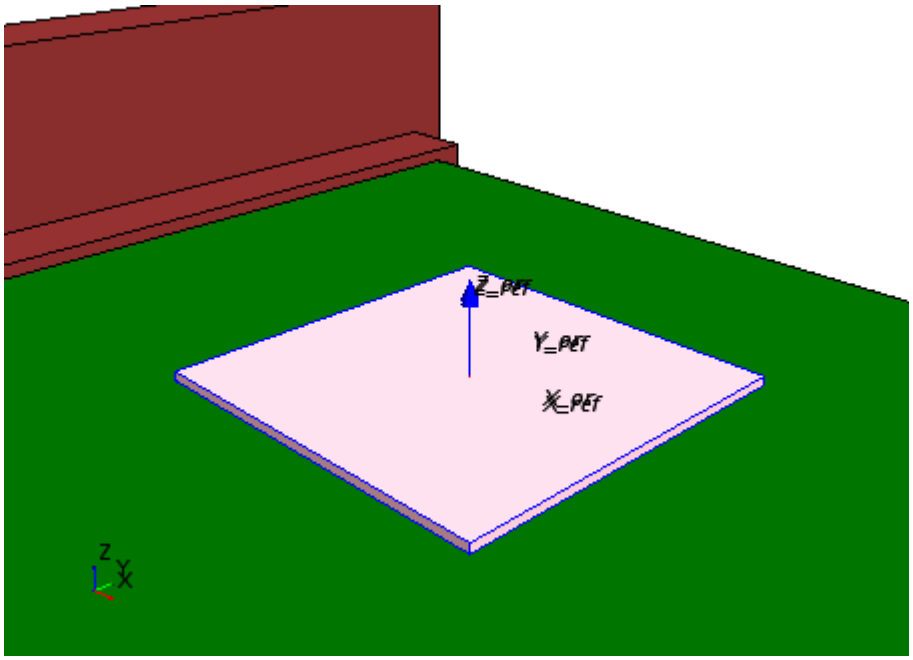
To define the substrate:

1. In the *Graphics* window, right-click the top surface of the PCB and select **New QuickPart** > **PCB** > **Rectangular**.
2. In the *Edit* dialog, set **Name** to `Substrate`.
3. Set the following *Geometry* properties:

Property	Setting
X	`40.0 mm`
Y	`40.0 mm`
Thickness	`1.0 mm`

4. Set the **Location** vector to `[0.0, 0.0, 0.5] mm`.

5. Click **Update**.



6. Within the *Physics* section, set **Thermal Specification** to **Specify By Layer**.
7. Set the thickness and metal fractions of the layers to the following values:

Name	Thickness	Metal Fraction
Layer 1	`0.3 mm`	`0.9`
Layer 2	`0.4 mm`	`0.01`
Layer 3	`0.3 mm`	`0.9`

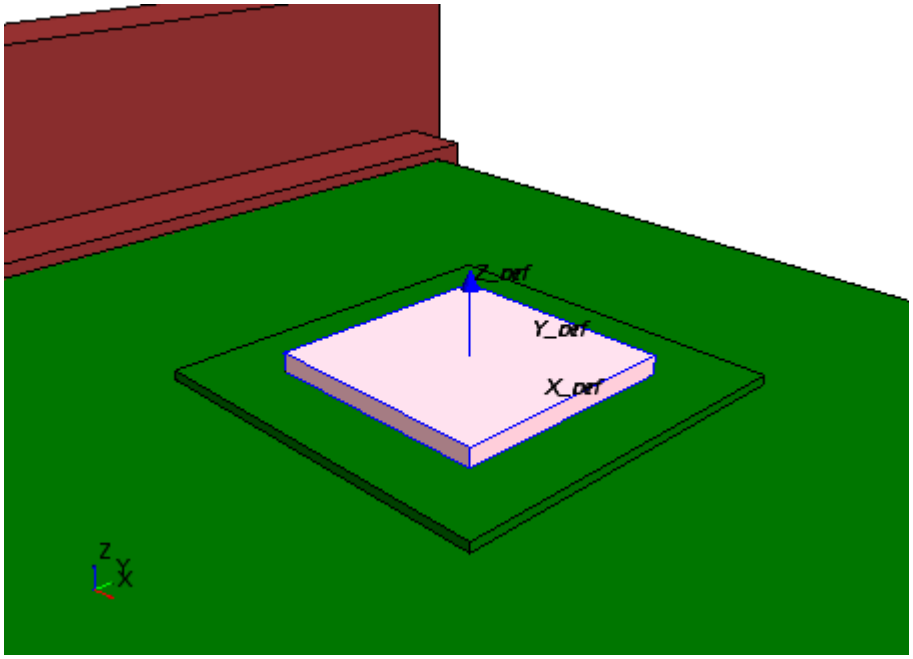
8. Click **OK**.

To define the integrated circuit:

9. Right-click the top surface of the substrate and select **New QuickPart** > **Generic** > **Block**.
10. In the *Edit* dialog, set **Name** to `IntegratedCircuit`.
11. Set the following *Geometry* properties:

Property	Setting
X	`25.0 mm`
Y	`25.0 mm`

12. Set the **Location** vector to `[0.0, 0.0, 0.0] mm`.
13. Click **Update**.

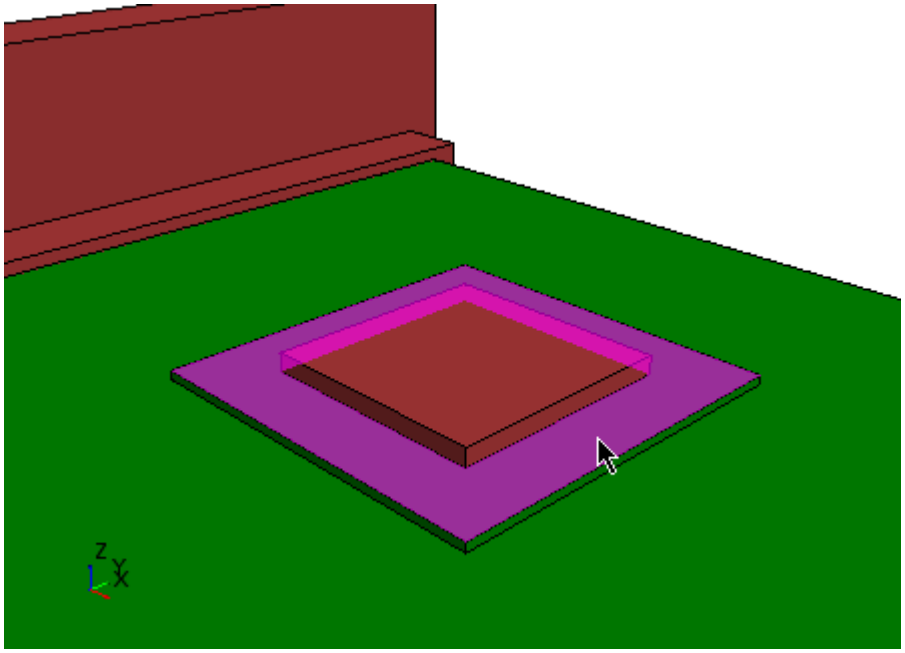


Account for the heat generation and material of the integrated circuit:

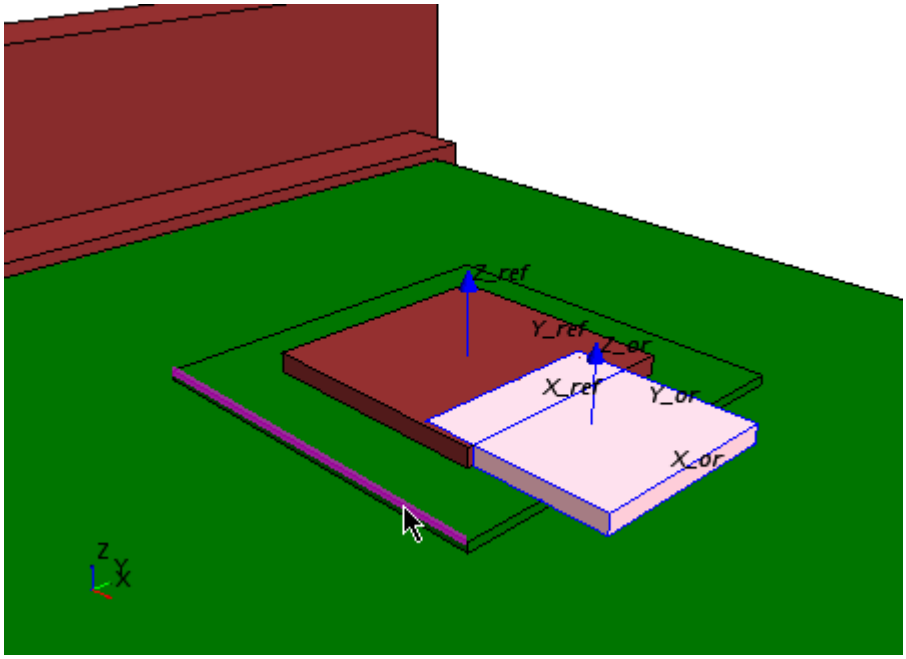
14. In the *Physics* section, set **Heat Source** to `1.0 W`.
15. Click the **Solid** material box and select **Si**.
16. Click **OK**.

To define the case:

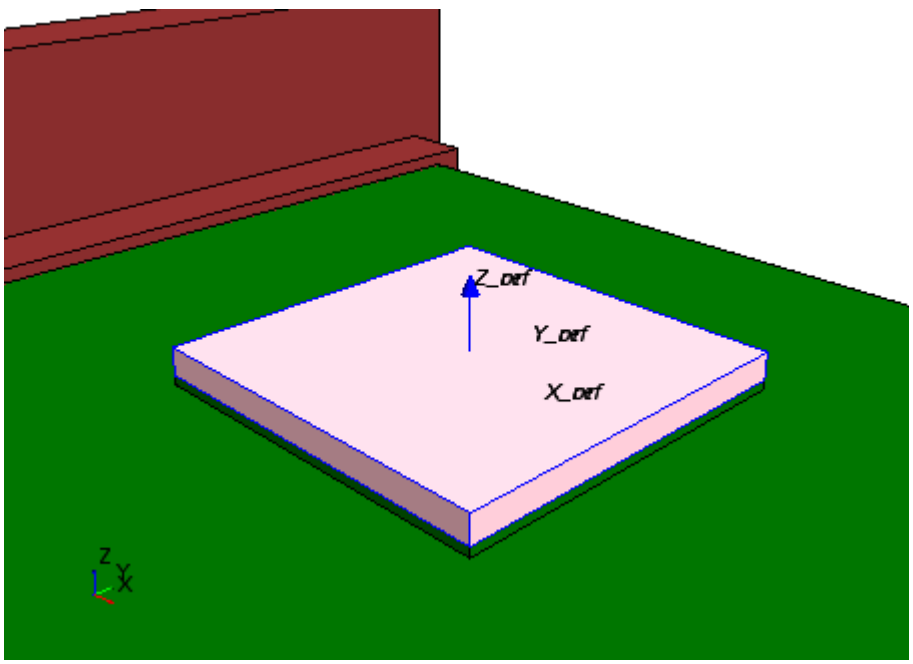
17. On the substrate, right-click the top face as highlighted below and select **New QuickPart** > **Generic** > **Block**.



18. In the *Edit* dialog, set **Name** to `Case`.
19. Right-click one of the long edges of the substrate as highlighted below and select **Set Dimension...** > **X**.



20. Repeat step 19 and select **Set Dimension...** > **Y**.
21. Within the *Geometry* section, set **Z** to `3.0 mm`.
22. Set the **Location** vector to `[0.0, 0.0, 0.0] mm`.
23. Click **Update**.



24. In the *Physics* section, click the **Solid** material box and select **Epoxy**.
25. Click **OK**.

To subtract the Integrated Circuit from the Case:

26. Right-click the **QuickParts** > **Case** node and select **Edit Interactions**.
27. In the *Subtract* group box, click the **Solid Quick Objects** box and select **IntegratedCircuit**.
28. Click **Apply** and **Close**.

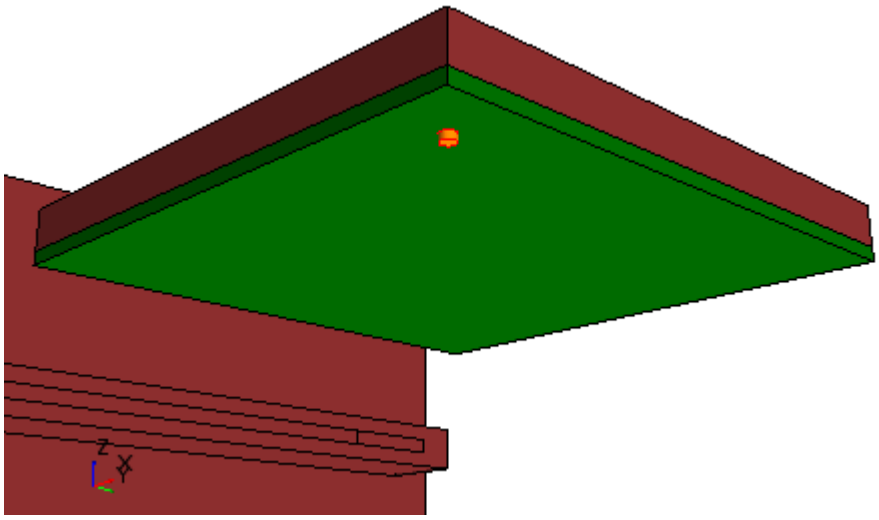
The next steps describe how to set up the solder balls. Spherical solder balls require extensive mesh refinement in the gaps between the spheres and the attached components. To achieve a good quality mesh with reasonable cell count for this tutorial, the solder balls are modeled as cylinders. To set up the cylindrical shape of a solder ball:

29. Right-click the **QuickParts** > **PCB** node and select **Hide**.

    The PCB geometry disappears from the scene.
30. In the *Graphics* window, right-click the bottom surface of the substrate and select **New Construction Geometry** > **Cylinder**.
31. In the *Edit* dialog, set the following *Geometry* properties:

Property	Setting
Height	`0.5 mm`
Base Radius	`0.5 mm`

32. Set the **Location** vector to `[-16.0, -16.0, 0.0] mm`.
33. Click **Update** and **OK**.



The solder balls follow a regular pattern along straight lines. To set up the pattern:

34. Right-click the **Construction Geometry** > **Cylinder** node and select **Pattern Construction Geometry**.
35. In the *Edit* dialog, within the *Direction (n)* group boxes, set the following properties:

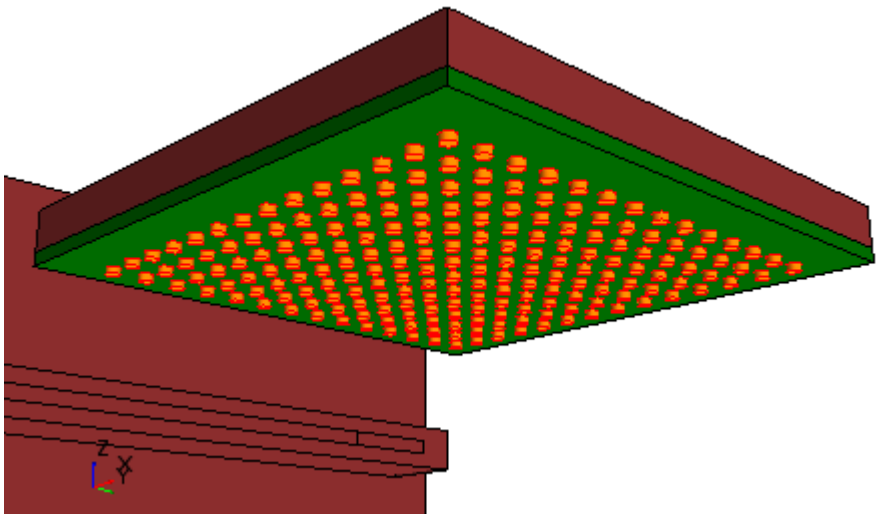
    a) *Direction (1)*

Property	Setting
Distance	`-2.5 mm`
Number of Instances	`14.0`

b) *Direction (2)*

Property	Setting
Distance	`2.5 mm`
Number of Instances	`14.0`

36. Click **Update** and **OK**.



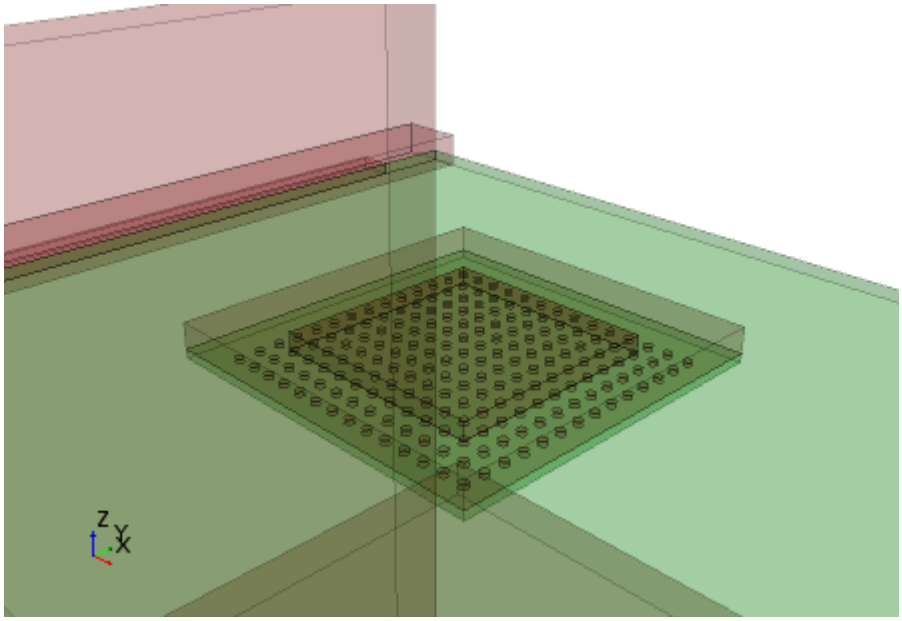
To assign physics to the solder balls:

37. Right-click the **QuickParts** node and select **New QuickPart** > **Generic** > **Custom**.
38. In the *Select Construction Geometry* dialog, select **Construction Pattern** and click **OK**.
39. Right-click the **QuickParts** > **Generic** node and select **Edit**.
40. In the in-place dialog set **Name** to `SolderBalls`.
41. Within the *Physics* section, click the **Solid** material box and select **SnPb**.
42. Click **Apply** and **Close**.

    All components of the chip package are now complete.
43. Right-click the **QuickParts** > **PCB** node and select **Show**.

To visualize the overlay of the different chip package components, make the Default scene transparent:

44. Right-click the **Scenes** > **Default** node and select **Edit Scene**.
45. In the in-place dialog, set **Opacity** to `0.1`.
46. Click **Apply** and **Close**.



Organize the components of the chip package in an assembly:

47. Under the **QuickParts** node, multi-select the **Case**, **SolderBalls**, **IntegratedCircuit**, and **Substrate** nodes.
48. Right-click the selected QuickParts and select **Composite**.
49. Right-click the **QuickParts** > **Composite** node, select **Rename...**, and enter `Package`.
50. Save the simulation .

Chamber

Define the geometry, placement, and physics of the chamber using an Enclosure QuickPart.

To define the chamber:

1. Right-click the **QuickParts** node and select **New QuickPart** > **Enclosure** > **Rectangular**.
2. In the *Edit* dialog, set **Name** to `Chamber`.
3. Set the following *Geometry* properties:

Property	Setting
X	`305.0 mm`
Y	`305.0 mm`
Z	`305.0 mm`
Thickness	`3.0 mm`

4. Click **Update** and **OK**.

To remove the bottom wall of the enclosure:

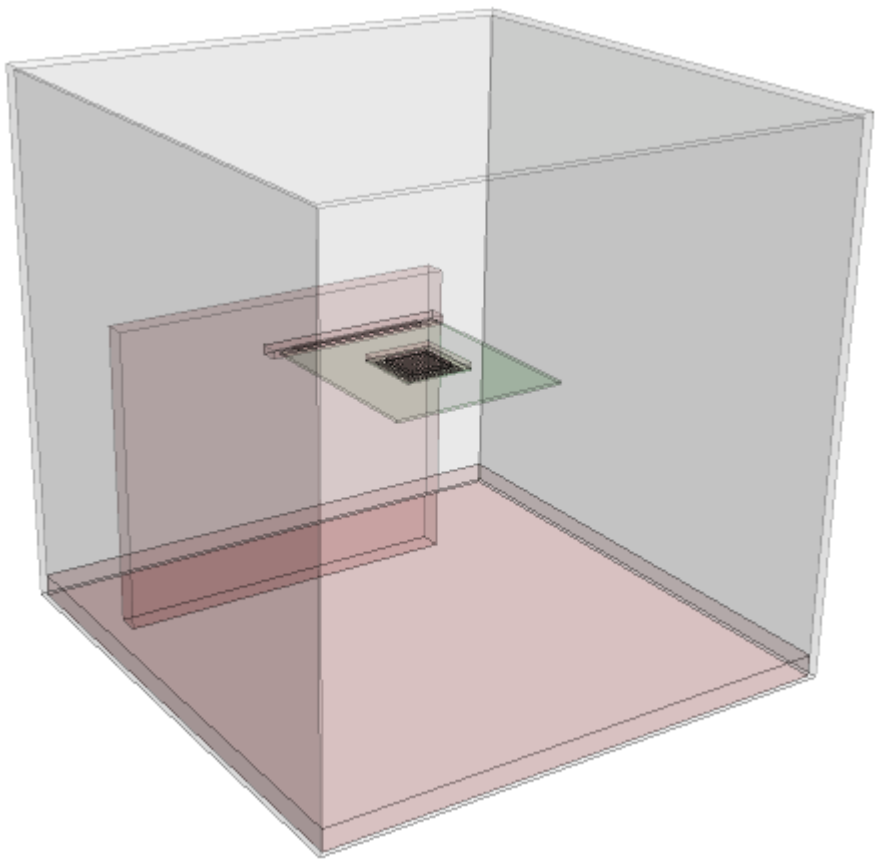
5. In the *Graphics* window, right-click the bottom face of the enclosure and select **Create Open Enclosure Face**.

Apply environmental conditions outside of the chamber:

6. In the *Graphics* window, multi-select the four outer side surfaces and the top surface of the chamber. Right-click and select **Create QuickPart Surface**.
7. In the *Rename* dialog, set **Name** to `Environment` and click **OK**.

   The selected surfaces are split from the **Default : Solids** QuickPart Surface of the chamber.
8. Right-click the **QuickParts** > **Chamber** > **Environment** node and select **Edit**.
9. In the *Edit* dialog, set **Thermal Specification** to **Environment**.
10. Click **Apply** and **Close**.

    All solid components of the JEDEC test case are now complete.



11. Save the simulation .

Extracting the Air Volume

Define the air domain that fills the void between the chamber, the test rig, and the chip package.

To extract the air volume:

1. Right-click the **QuickParts** node and select **Create Extract Volume**.
2. In the *Edit* dialog, set *Extract Volume Domain Defined By:* to **QuickParts + Const. Geometry**.
3. Next to the **QuickParts + Const. Geometry** text box, click **Select...**.
4. In the *Select Objects* dialog, select the **Chamber** and **Table** QuickParts and click **OK**.
5. Click **OK**.

   An **Extract Volume** QuickPart appears in the tree.
6. Save the simulation .

Generating the Mesh

Create the volume mesh for the solid components and the fluid domain. For this tutorial, leave the overall mesh settings as default, but refine the mesh on the solder balls.

The mesh is refined suitably for this tutorial. However, to save computational expense, this mesh is coarser than the mesh that would be required for an industrial analysis.

To define the mesh refinement on the solder balls:

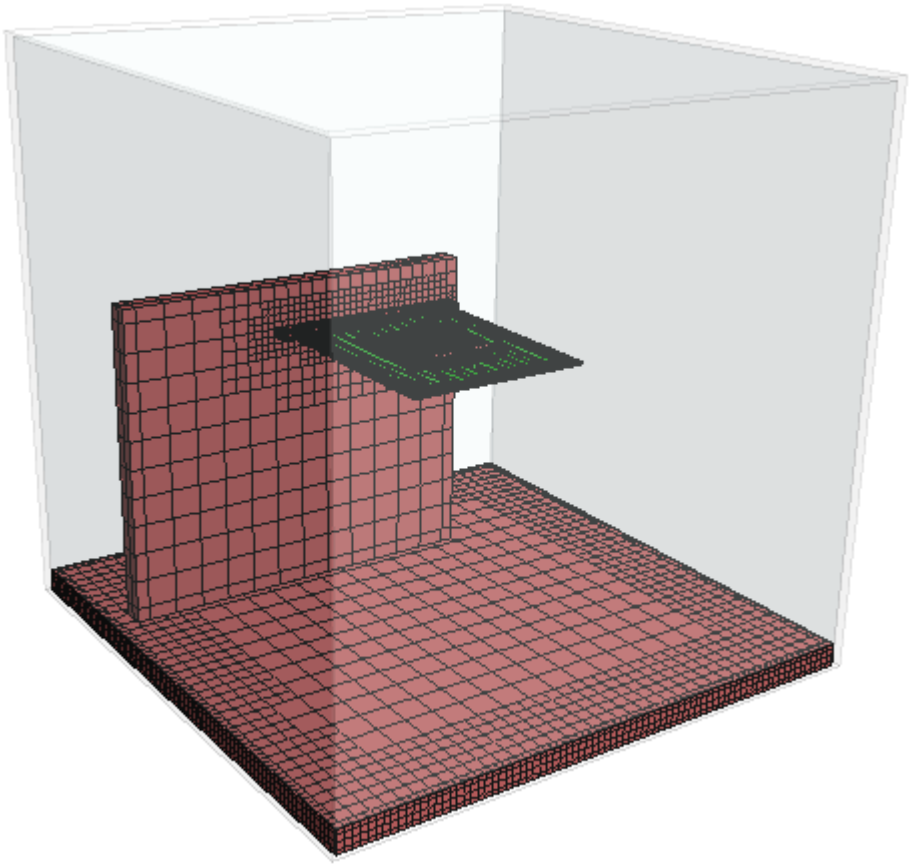
1. Right-click the **Mesh** > **Custom Controls** node and select **New** > **Surface Control**.
2. In the *Edit* dialog, set **Name** to `SolderBalls_SurfaceControl`.
3. Click the **Surfaces** text box, select the **Package** > **Solderballs** QuickPart, and click **OK**.
4. For the **Target Size**, select **Specify Custom Value** from the drop-down menu.
5. Activate **Absolute Size** and set the value to `2.0 mm`
6. Click **Apply** and **Close**.

To generate the mesh:

7. Click (**Generate Volume Mesh**) in the *Mesh Generation* toolbar.

Upon completion of the mesh generation process, make the mesh visible on the test rig and the chip package:

8. Right-click the **Scenes** node and select **New**.
9. In the in-place dialog, set **Name** to `Mesh`.
10. Click the **Drawable Items** text box and select all QuickParts except Chamber and Extract Volume.
11. From the **Display Mesh** drop-down menu, select the **Volume Mesh** option.
12. Click **OK**.
13. Position the parts to obtain a similar view to the one shown below.



14. Save the simulation .

Preparing Scenes for Visualization

Create two scenes: one to display the velocity vectors on a plane cutting through the center of the simulation domain and one to display the temperature on the chip package.

Begin with hiding the Mesh scene:

1. Select the **Scenes** > **Mesh** node and select **Toggle Visibility**.

To create the plane section:

2. Right-click the **Scenes** > **Drawables** node and select **New** > **Section**.
3. Within the in-place dialog, in the *Plane Parameters* group box, set the **Normal** vector to `[0.0, 1.0, 0.0] mm`.
4. Click **OK**.

Set up the scene that displays the velocity vectors on the plane section:

5. Right-click the **Scenes** node and select **New**.
6. In the in-place dialog for the scene, set the following properties:

Property	Setting
Name	`Air Velocity`
Drawable Items	**Section**
Displayer Mode	**Vector**
Field Function	**Velocity**

7. Click **OK**.

Set up the scene that displays the temperature on the chip package:

8. Create a new scene and set the following properties:

Property	Setting
Name	`Package Temperature`
Drawable Items	**Package**
Displayer Mode	**Scalar**
Field Function	**Temperature**
Color Map	**cool-warm**
Contour Style	**Smooth Filled**

9. Save the simulation .

Preparing Temperature Report

Define a report for the junction temperature and the temperature at the thermocouple probe.

Start with creating the point probes:

1. Right-click the **Results** > **Derived Data** node and select **New** > **Quick Point Probe**.
2. In the *Edit* dialog, set the following properties:

Property	Setting
Name	`Junction`
QuickParts	**Package** > **IntegratedCircuit**
Point [X, Y, Z]	`[0.0, 0.0, 161.5] mm`

3. Click **OK**.
4. Create a second point probe with the following properties:

Property	Setting
Name	`Thermocouple`
QuickParts	**Extract Volume**
Point [X, Y, Z]	`[0.0, -127.5, 132.5] mm`

To extract the temperature values at the point probes, create a report:

5. Right-click the **Results** > **Reports** node and select **Create Post Report**.
6. In the *Edit* dialog, set the following properties:

Property	Setting
Name	`Probe Temperatures`
Report Field	**Temperature**
Report Items	**Results** > **Derived Data** > **Junction** **Results** > **Derived Data** > **Thermocouple**

7. Click **OK**.
8. Save the simulation .

Running the Simulation

The simulation is now set up and ready to run.

To run the simulation:

1. Click (**Run**) in the *Solution* toolbar.

   The *Residuals* display is created automatically and shows the progress of the solver. While the simulation is running, you can click the *Electronics Cooling 1* tab at top of the Graphics window to view the solution updating.

2. After the run is finished, save the simulation .

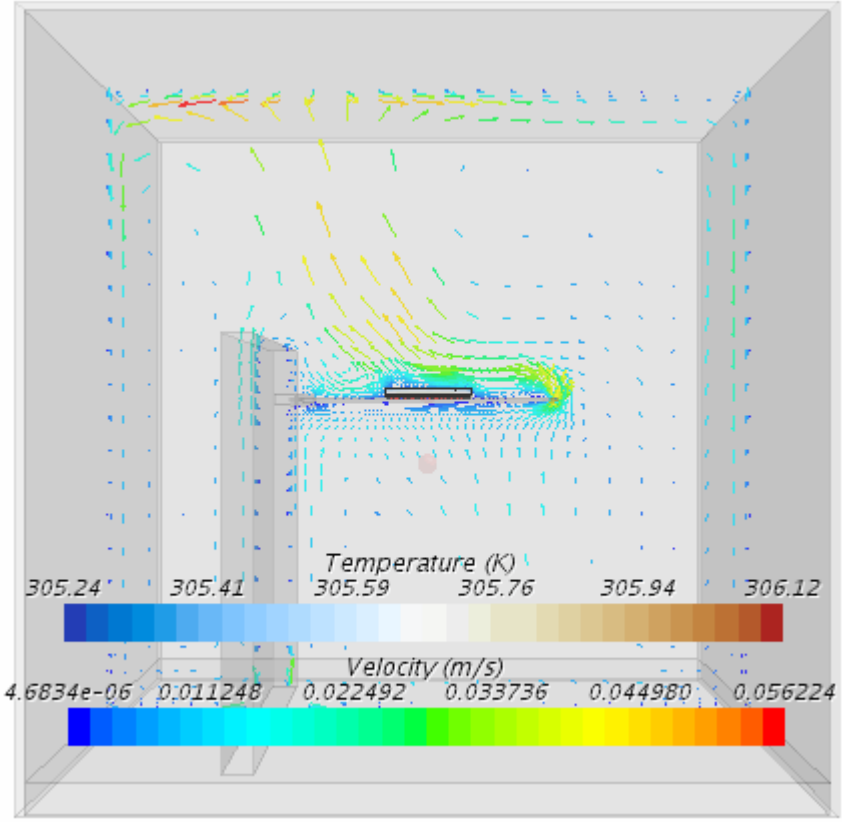
Analyzing the Results

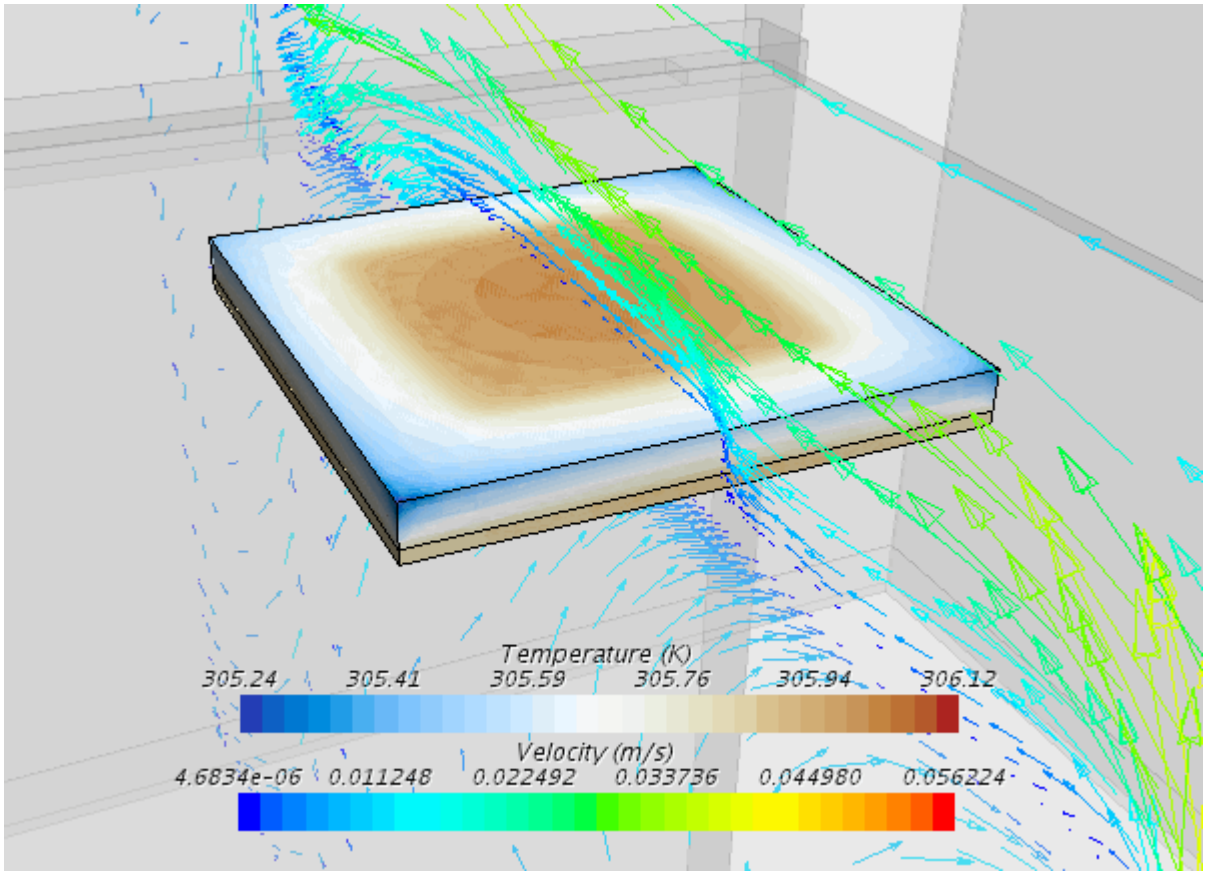
View the results of the converged simulation and calculate the junction-to-ambient thermal resistance.

After the run is finished, examine the velocity and temperature displays:

1. Click the *Electronics Cooling 1* tab at top of the Graphics window.

   The following screenshots display the air flow inside the chamber and the temperature of the package.


Temperature (K)
305.24
305.41
305.59
305.76
305.94
306.12
Velocity (m/s)
4.6834e-06
0.011248
0.022492
0.033736
0.044980
0.056224



Determine the junction-to-ambient thermal resistance θ_{JA}:

2. Right-click the **Results** > **Reports** > **Probe Temperatures** node and select **Tabulate**.

   Within the *Tabular Data* dialog, the following data are displayed:

QP/QPS	Field	Minimum/Maximum/ Average
Junction	Temperature	306.112 K
Thermocouple	Temperature	300.536 K

3. Calculate θ_{JA} as:

$$\theta_{JA} = \frac{T_J - T_A}{P_H} = \frac{306.112\ \mathrm{K} - 300.536\ \mathrm{K}}{1.0\ \mathrm{W}} = 5.576\ \frac{\mathrm{K}}{\mathrm{W}}$$

Summary

The Natural Convection JEDEC tutorial uses a chip package in a standardized still air environment to introduce various features of the Electronics Cooling Toolset.

The features that were introduced include:

- Invoking the Electronics Cooling Toolset.

- Defining the setup for a natural convection driven flow.
- Setting up geometry and physics using templated QuickParts.
- Extracting the air volume.
- Meshing the solution domain including a customized refinement.
- Preparing scenes and reports for post-processing.
- Running the simulation.
- Analyzing the flow and temperature field and reporting temperature data.

Bibliography

This tutorial references the following paper:

[790] *JEDEC Solid State Technology Association, JEDEC Standard JESD51-2A "Integrated Circuits Thermal Test Method Environmental Conditions—Natural Convection (Still Air)", January 2008*

Electronics Cooling Toolset: Graphics Card Cooling

When you are positioning fans that cool electronic components, the Electronics Cooling Toolset allows you to assess the impact that they have on removing heat. By comparing multiple designs, you can decide which arrangement gives you the best performance.

This tutorial models the same problem as introduced in Multi-Part Solid: Graphics Card Cooling. The geometry consists of a simplified graphics card inside a PC tower case. Two fans drive the flow inside the case to cool the card. In this tutorial, the geometry of the fans is not imported but constructed directly using Axial Fan QuickParts.

The analysis covers the following heat transfer modes:

- Conduction in the solid components of the graphics card and the fans.
- Diffusion and forced convection in the enclosed air.

Simulation results are displayed using the post-processing capabilities that are available in the Electronics Cooling Toolset.

Contents:

Prerequisites
Launching the Electronics Cooling Toolset and Naming the Simulation
Defining the Setup
Importing the CAD Model
Setting up the Components
Extracting the Air Volume
Generating the Mesh
Preparing Scenes for Visualization
Preparing the Temperature Report
Running the Simulation
Analyzing the Results
Summary

Prerequisites

This tutorial guides you through the setup of the graphics card cooling simulation using the dedicated Electronics Cooling Toolset in STAR-CCM+.

Before attempting this tutorial, make sure that the (**Launch Electronics Cooling App**) button is visible in the STAR-CCM+ toolbar. For information on how to include the button in the toolbar, see Accessing the Electronics Cooling Toolset.

Launching the Electronics Cooling Toolset and Naming the Simulation

Create a new simulation, launch the Electronics Cooling Toolset, and name the simulation `eCoolToolset_GraphicsCardTutorial.sim`.

To launch the Electronics Cooling Toolset:

1. Start up STAR-CCM+ and create a new simulation by clicking **File** > **New Simulation**.
2. From the STAR-CCM+ toolbar, click (**Launch Electronics Cooling App**).

   An *Electronics Cooling* tab opens within STAR-CCM+.

To save the simulation:

3. Select **File** > **Save As...** and save the simulation under the name `eCoolToolset_GraphicsCardTutorial.sim`

Defining the Setup

Set up the solution physics and materials for the solid components and the flow domain. For the solver, reduce the maximum number of iterations.

To define the solution physics:

1. Right-click the **Setup** > **Physics** node and select **Edit**.

   An additional *Edit* tab opens that displays the default ambient conditions, solution physics, and materials.
2. Within the *Solution Physics* group box, set **Solve** to **Forced Convection**.

   When solving for a forced convection problem, the Electronics Cooling Module by default solves for flow and energy in a turbulent flow regime.

Within the *Materials* group box, define the materials for the graphics card components:

3. Next to the **Solid Sim Materials** box, click **Select**.
4. Within the *Select Materials* dialog, in the **Solid Simulation Materials** box, select **Al**.
5. In the table of material properties, set **Thermal Conductivity** to `167 W/m-K`.
6. Click **Add Material**.
7. From the database of standard solid materials, select **Si (Silicon)** and click **OK**.
8. Set the **Si** material properties to the following values:

Property	Setting
Thermal Conductivity	`124 W/m-K`
Specific Heat	`700 J/kg-K`
Density	`2330 kg/m^3`

9. In the list of materials, select **Si** and click **Duplicate Material**.
10. Select the added material and set the following properties:

Property	Setting
Material Name	`ABS`
Thermal Conductivity	`2.5 W/m-K`
Specific Heat	`2050 J/kg-K`
Density	`1050 kg/m^3`

11. Repeat steps 8 and 9 to create another solid material using the following properties:

Property	Setting
Material Name	`Alumina`
Thermal Conductivity	`30 W/m-K`
Specific Heat	`850 J/kg-K`
Density	`3960 kg/m^3`

12. Click **Apply** and **Close**.
13. Finish the setup by clicking **Close** at the bottom of the tab.

In this tutorial, the simulation converges within 500 iterations. To set the maximum number of iterations:

14. Right-click the **Setup** > **Solver** node and select **Edit**.
15. Within the *Stopping Criteria* group box, set **Maximum Steps** to `500`.
16. Click **Close**.

Save the simulation:

17. In the *System* toolbar (see Toolbar Reference), click (**Save**).

Importing the CAD Model

The setup of the graphics card components and the air volume starts from a CAD geometry in Parasolid format.

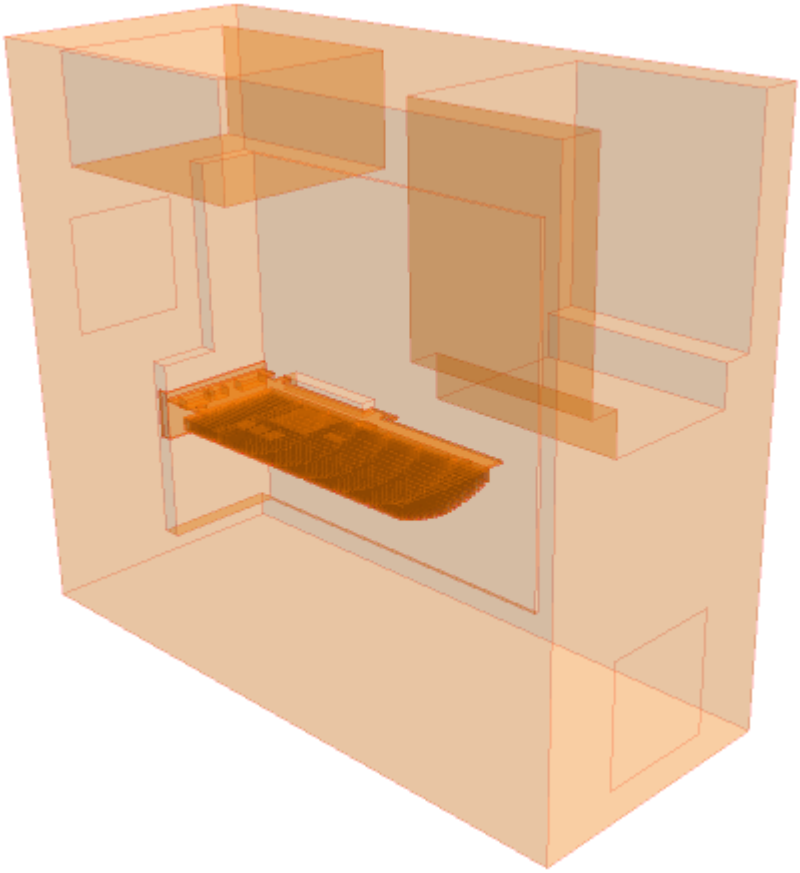
To import the geometry:

1. Right-click the **Construction Geometry** node and select **Import CAD Model**.
2. In the Open dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/heatTransferAndRadiation/data`.
3. Select `eCoolToolset_GraphicsCardTutorial_Geometry.x_b` and click **Open**.

   A total of 22 parts are imported.

4. Click (**Make Scene Transparent**) in the *Visualization* toolbar.
5. Position the parts to obtain a similar view to the one shown below.

   For information on how to change the view in the *Graphics* window, see Mouse Configurations and Changing the View.



The components of the imported graphics card and PC tower case are described in Multi-Part Solid: Graphics Card Cooling.

6. Save the simulation .

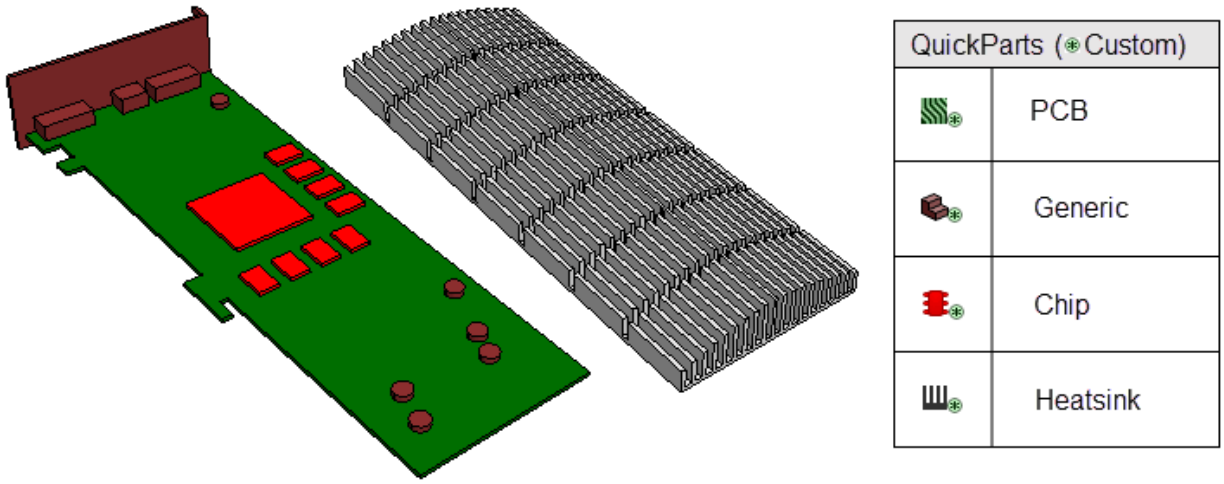
Setting up the Components

Create Custom QuickParts for the imported graphics card components. At the inlet and outlet patches of the PC tower case, place templated Axial Fan QuickParts.

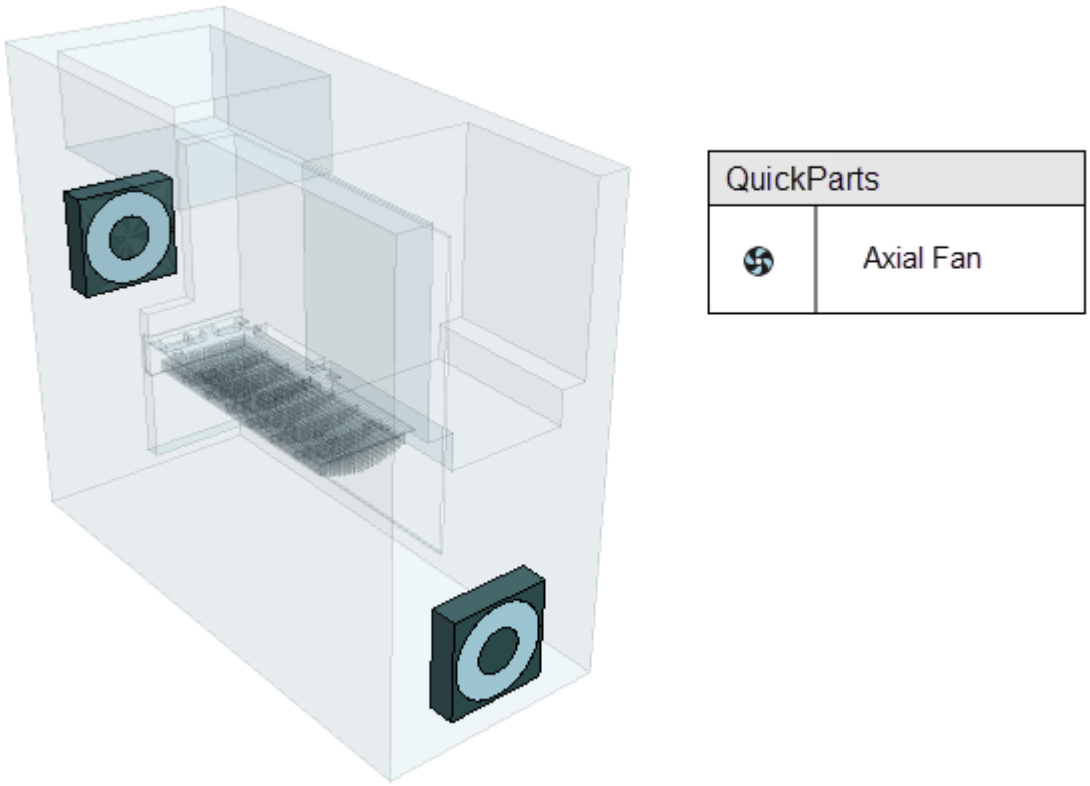
Within the Electronics Cooling Toolset, a QuickPart allows you to define the geometry, placement, and physics of a component in one step. For more information, see QuickParts.

The next steps describe the setup of the following components:

- Graphics card



- Inlet and outlet fans



Contents:

Graphics Card
Inlet and Outlet Fans

Graphics Card

Define the geometry, placement, and physics of the graphics card using Custom PCB, Chip, Heatsink, and Generic QuickParts.

To define the printed circuit board:

1. Right-click the **QuickParts** node and select **New QuickPart** > **PCB** > **Custom**.
2. In the *Select Construction Geometry* dialog, select **PCB** and click **OK**.
3. Right-click the **QuickParts** > **PCB** node and select **Edit**.
4. In the *Edit* dialog, set the following *Physics* properties:

Property	Setting
Thermal Specification	`Specify By Layer`
PCB Normal	`[0.0, 1.0, 0.0]`

5. Set the metal fractions of the layers to the following values:

Name	Thickness	Metal Fraction
Layer 1	`0.25 mm`	`0.3`
Layer 2	`1.5 mm`	`0.001`
Layer 3	`0.25 mm`	`0.3`

6. Click **Apply** and **Close**.

To define the memory chips and the GPU chip casing:

7. Right-click the **QuickParts** node and select **New QuickPart** > **Chip** > **Custom**.
8. In the *Select Construction Geometry* dialog, select **GPU Chip** and all memory chips, that is **Memory Chip 1** to **Memory Chip 8**.
9. Click **OK**.

   A total of nine Chip QuickParts are created.
10. Right-click the **QuickParts** > **Chip** node and select **Edit**.
11. In the *Edit* dialog, set **Name** to `GPU Chip Casing`.
12. Set the following *Physics* properties:

Property	Setting
Material	`Si`
Heat Source	`25.0 W`

13. Click **Apply** and **Close**.
14. Multi-select the **QuickParts** > **Chip 2** to **Chip 9** nodes, right-click and select **Edit**.

    To select multiple nodes hold down the `<Ctrl>` key.
15. Set the following *Physics* properties:

Property	Setting
Material	`Si`
Heat Source	`4.0 W`

Organize the memory chips in an assembly:

16. Multi-select the **QuickParts** > **Chip 2** to **Chip 9** nodes, right-click and select **Composite**.
17. Right-click the **QuickParts** > **Composite** node, select **Rename...**, and enter the name `Memory Chips`.

To set up the capacitors:

18. Right-click the **QuickParts** node and select **New QuickPart** > **Generic** > **Custom**.
19. In the *Select Construction Geometry* dialog, select all capacitors, that is **Capacitor 1** to **Capacitor 6**, and click **OK**.
20. Multi-select the **QuickParts** > **Generic** to **Generic 6** nodes, right-click and select **Edit**.
21. In the *Edit* dialog, click the **Material** > **Solid** box and select **Alumina**.
22. Click **Apply** and **Close**.
23. Multi-select the **QuickParts** > **Generic** to **Generic 6** nodes, right-click and select **Composite**.
24. Rename the **Quick Parts** > **Composite** node, `Capacitors`.

To set up the connector port blocks and the back plate:

25. Right-click the **QuickParts** node and select **New QuickPart** > **Generic** > **Custom**.
26. In the *Select Construction Geometry* dialog, select **Port Block1**, **Port Block 2**, **Port Block 3**, and **Plate** and click **OK**.
27. Rename the **QuickParts** > **Generic** node, `Back Plate`.
28. Multi-select the **QuickParts** > **Generic 2**, **Generic 3**, and **Generic 4** nodes, right-click and select **Composite**.
29. Rename the **QuickParts** > **Composite** node, `Connector Port Blocks`.

To set up the heat sink:

30. Right-click the **QuickParts** node and select **New QuickPart** > **Heatsink** > **Custom.**
31. In the *Edit* dialog, within the *Geometry* section, click the **Base** box and select **Heatsink Block**.
32. Click **OK**.

The components of the graphics card are now set up. Finally, organize the components in an assembly:

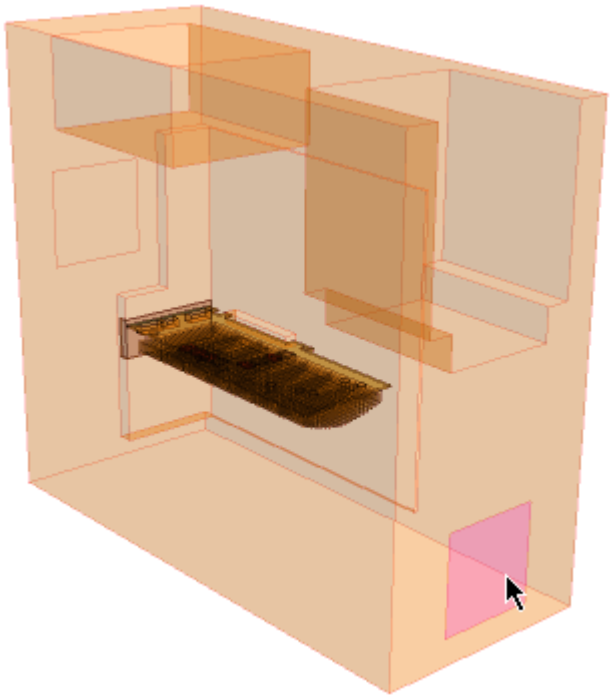
33. Multi-select all QuickParts in the tree, right-click and select **Composite**.
34. Rename the **QuickParts** > **Composite** node, `Graphics Card`.
35. Save the simulation .

Inlet and Outlet Fans

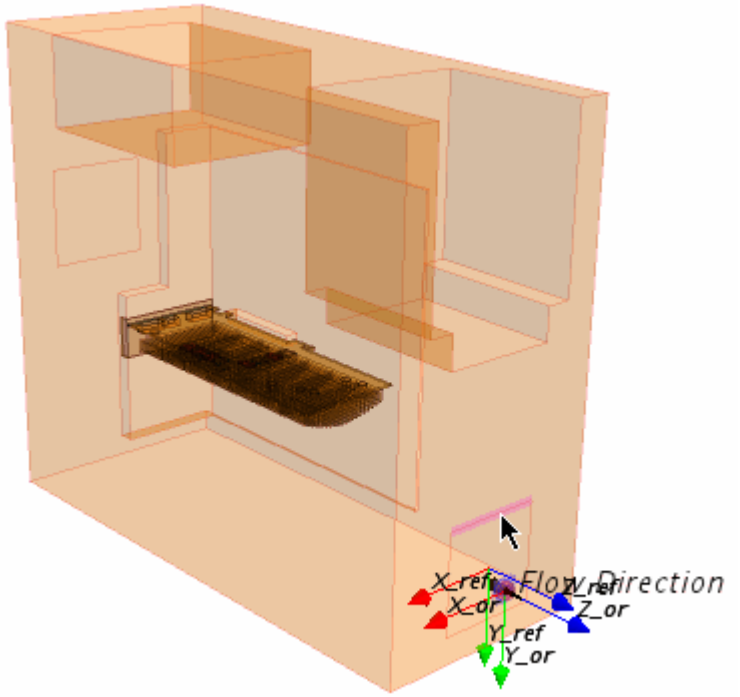
Define the inlet fan and outlet fan using template-based Axial Fan QuickParts.

To define the inlet fan:

1. In the *Graphics* window, select the inlet patch that is highlighted below.



2. Right-click and select **New QuickPart** > **Axial Fan** > **Templated**.
3. In the *Edit* dialog, set **Name** to `Inlet Fan`.
4. In the *Graphics* window, select one of the horizontal edges of the patch as displayed below, right-click and select **Set Dimension** > **Housing Width**.



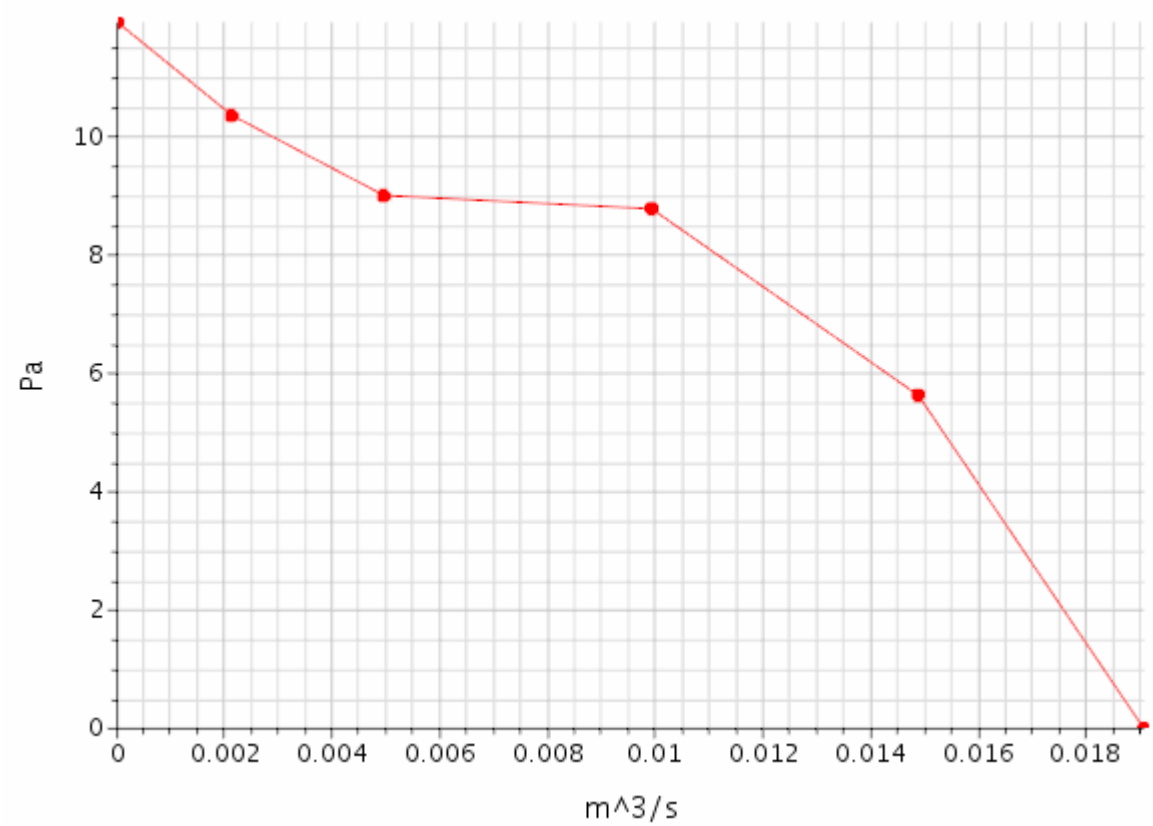
5. In the *Edit* dialog, set the following *Geometry* properties:

Property	Setting
Housing Depth	`0.025 m`
Fan Radius	`0.043 m`
Hub Radius	`0.02 m`

6. Set the **Location** vector to `[0.0, 0.0, 0.0] m`.
7. Click **Update**.
8. Within the *Physics* section, set the following properties:

Property	Setting
Housing Material	**ABS**
Hub Material	**ABS**
Fan Surface	**Back Face**

To model the pressure rise across the axial fan, the Electronics Cooling Toolset requires the specification of a fan performance curve. A fan performance curve describes the pressure rise through the fan as a function of volumetric flow rates. The performance curve for the fans that are used in this tutorial is shown below. The curve corresponds to a fan operating rotation rate of 1200 rpm.



To import the fan curve

9. Set the **Fan Curve Type** to **Table**.
10. Next to the **Fan Curve File** box, click **Import** and navigate to `[INSTALL_DIR]/doc/startutorialsdata/heatTransferAndRadiation/data`.
11. Select `eCoolToolset_GraphicsCardTutorial_FanCurve.csv` and click **Open**.

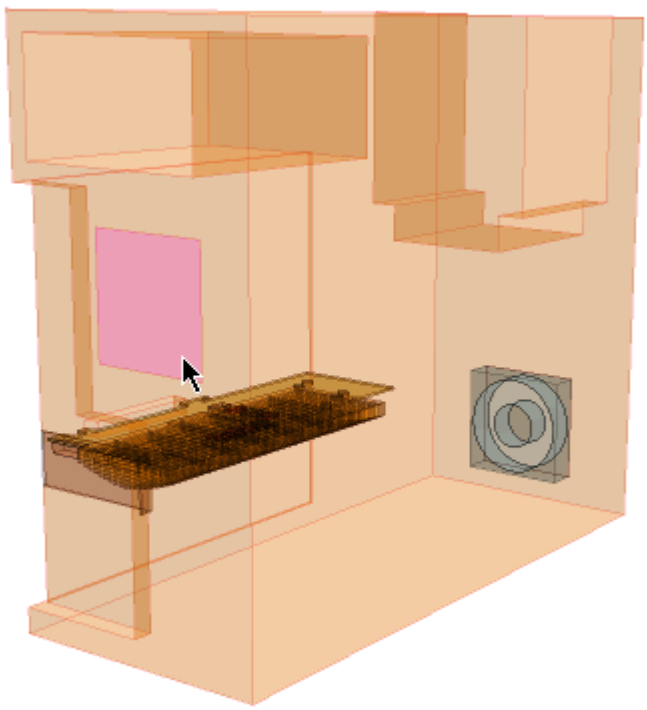
    The fan curve is imported.
12. For the imported data, set the following units:

Property	Setting
Volume Flow Units	**m^3/s**
Pressure Units	**Pa**

13. Click **OK**.

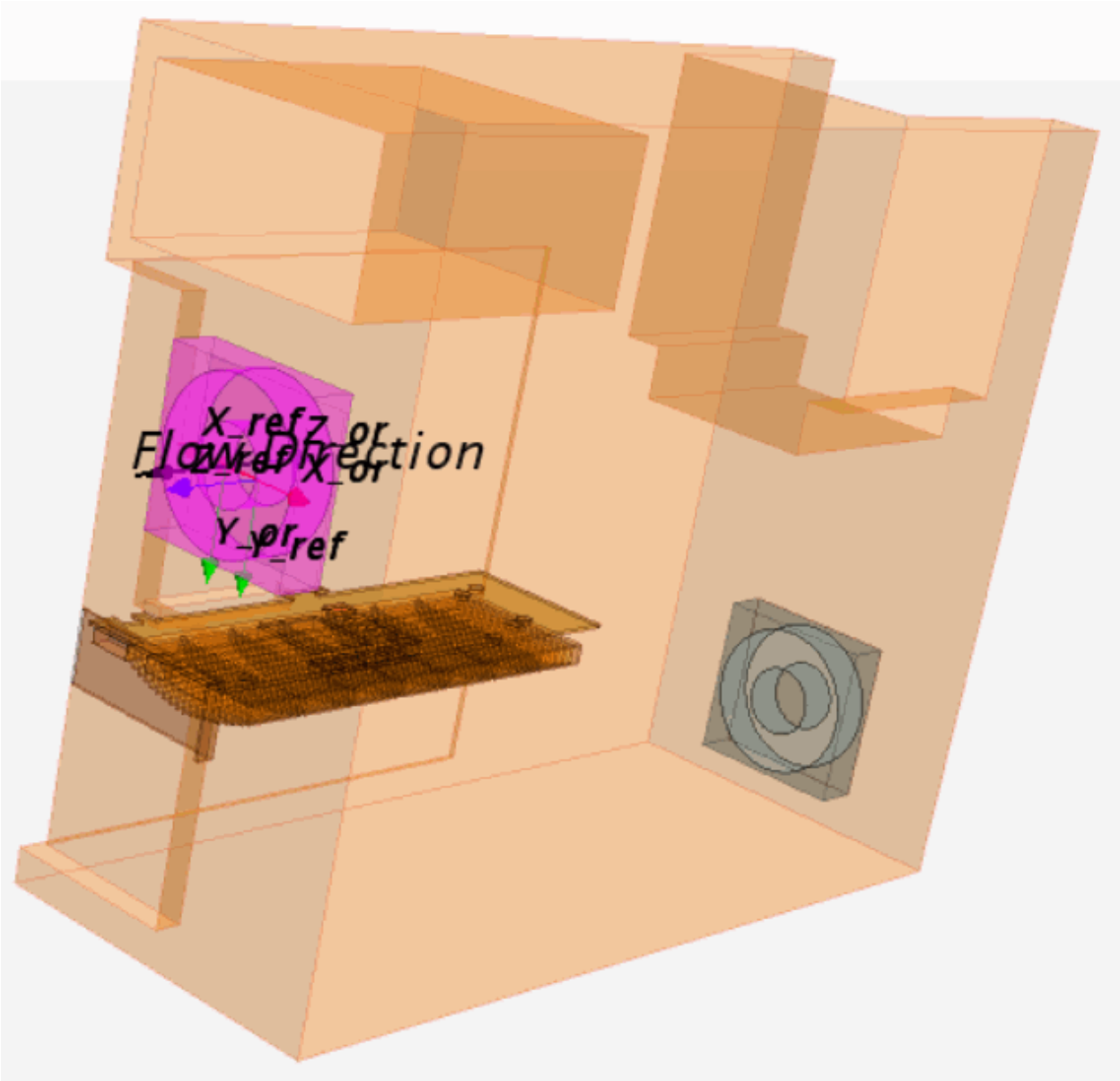
To define the outlet fan, copy the inlet fan, move it to the outlet position, and reverse the flow direction:

14. In the *Electronics Cooling* panel, right-click the **QuickParts** > **Inlet Fan** node and select **Copy QuickParts**.
15. In the *Graphics* window, select the outlet patch as displayed below, right-click, and select **Paste QuickParts**.



16. In the *Electronics Cooling* panel, right-click the **QuickParts** > **Inlet Fan Copy** node and select **Edit**.
17. In the *Edit* dialog, set **Name** to `Outlet Fan`.
18. In the *Geometry* section, set **Local Origin** to **C**.

    The local origin is moved to the back face of the fan.



19. Set the following *Physics* properties:

Property	Setting
Fan Surface	**FrontFace**
Reverse Direction	✓

20. Click **Apply** and **Close**.
21. Save the simulation .

Extracting the Air Volume

Define the air domain that fills the void between the graphics card, the axial fans, and the PC tower case.

To extract the air volume:

1. Right-click the **QuickParts** node and select **Create Extract Volume**.
2. In the *Edit* dialog, set *Extract Volume Domain Defined By:* to **Construction Geometry**.
3. Next to the **Construction Geometry** text box, click **Select...**.
4. In the *Select Object* dialog, select the **Air Volume** Construction Geometry and click **OK**.
5. Click **OK**.

   An **Extract Volume** QuickPart appears in the tree.
6. Save the simulation .

Generating the Mesh

Create the volume mesh for the graphics card components, the fans, and the air volume.

For this tutorial, leave the mesh settings as default. The Electronics Cooling Toolset then calculates the base and minimum cell size dependent on the overall model size.

To generate the mesh:

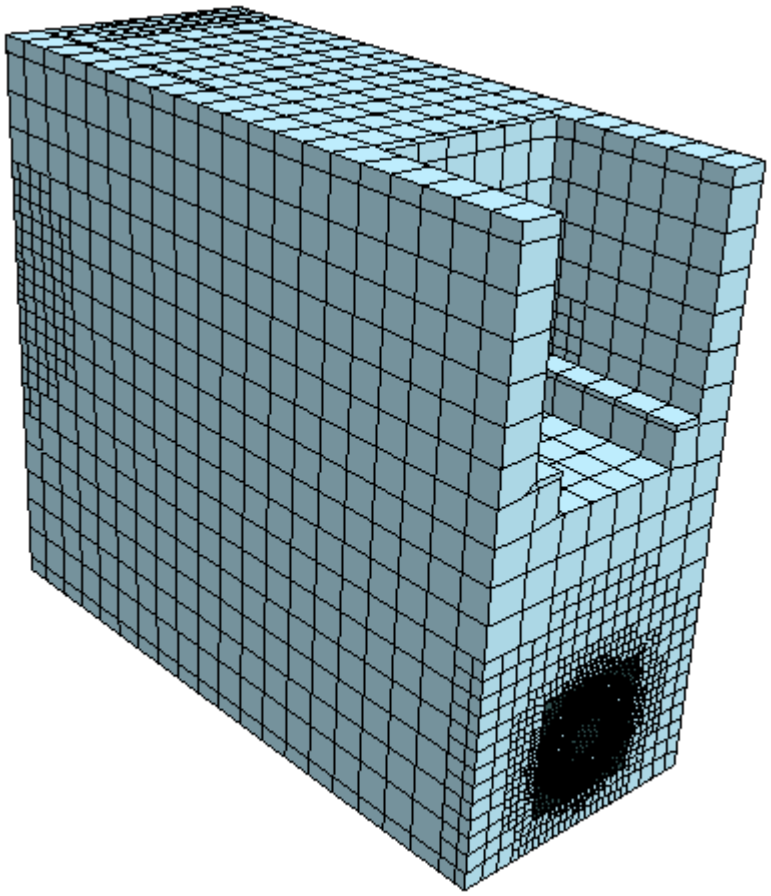
1. Click (**Generate Volume Mesh**) in the *Mesh Generation* toolbar.

   The mesh generation process takes several seconds, depending on the specifications of your computer.

To visualize the mesh, hide the default scene and create a mesh scene:

2. Right-click the **Scenes** > **Default** node and select **Toggle Visibility**.
3. Right-click the **Scenes** node and select **New**.
4. In the *Edit* dialog, set **Name** to `Mesh`.
5. Click the **Drawable Items** text box and select all QuickParts.
6. From the **Display Mesh** drop-down menu, select the **Volume Mesh** option.
7. Click **OK**.
8. Click (**Make Scene Transparent**) in the *Visualization* toolbar.

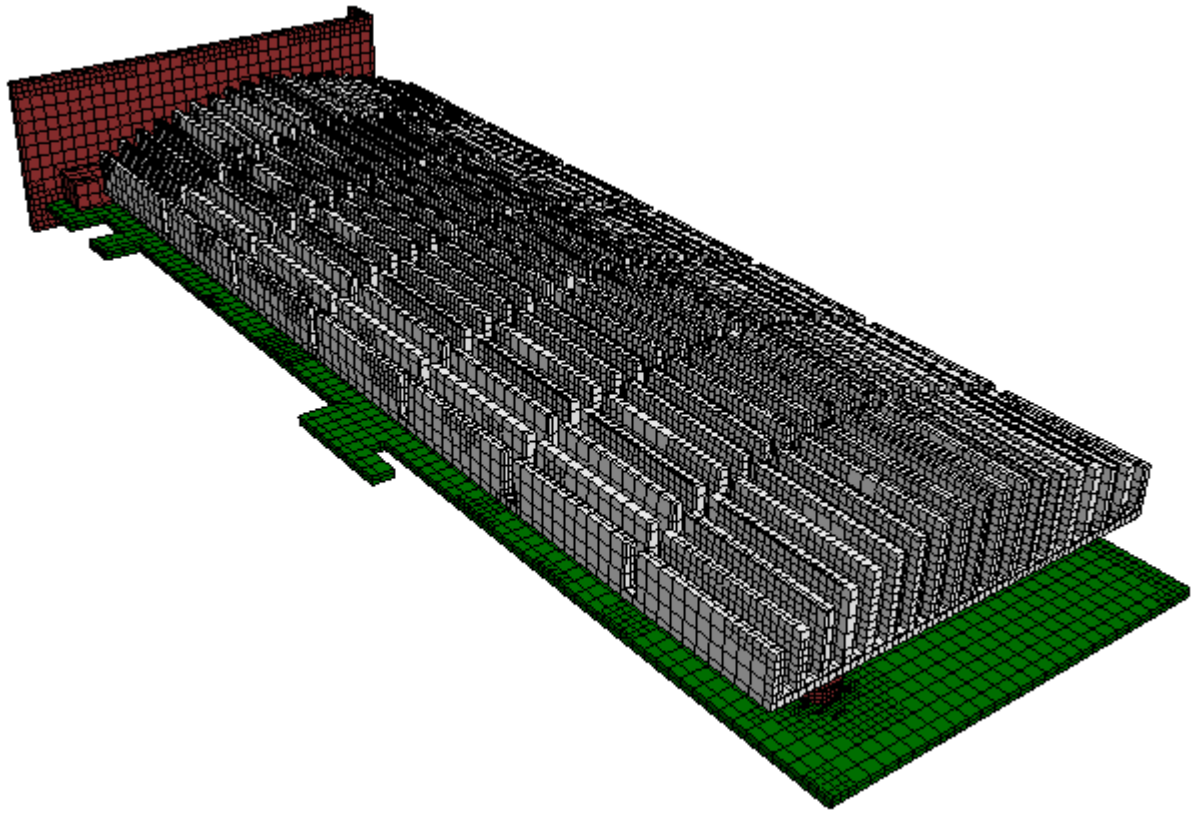
   The mesh of the fluid domain is shown below.



To inspect the mesh on the graphics card:

9. Edit the **Scenes** > **Mesh** node.
10. Click the **Drawable Items** text box and deselect **Extract Volume**, **Inlet Fan**, and **Outlet Fan**.
11. Click **Apply** and **Close**.

    The following screenshot shows the mesh on the graphics card:



The mesh is relatively coarse. It is suitable for this tutorial to save computational expense. For an industrial analysis, a finer mesh would be required.

12. Save the simulation .

Preparing Scenes for Visualization

Create two scenes: one to display the velocity magnitude on a plane cutting through the tower case, and another to display the temperature on the graphics card.

Begin with creating the plane section:

1. Right-click the **Scenes** > **Drawables** node and select **New** > **Section**.
2. Within the *Edit* dialog, in the *Plane Parameters* group box, set the **Origin** vector to `[0.0, 0.0, 0.035] m`.
3. Click **OK**.

Create a scene to display the velocity magnitude on the plane section:

4. Right-click the **Scenes** node and select **New**.
5. In the in-place dialog for the scene, set the following properties:

Property	Setting
Name	`Air Velocity`
Drawable Items	**Section**

Property	Setting
Displayer Mode	**Scalar**
Field Function	**Velocity** > **Lab Reference Frame** > **Magnitude**
Contour Style	**Smooth Filled**

6. Click **OK**.

Create a second scene for displaying the temperature on the graphics card:

7. Create a new scene and set the following properties:

Property	Setting
Name	`Graphics Card Temperature`
Drawable Items	**Graphics Card**
Displayer Mode	**Scalar**
Field Function	**Temperature**
Color Map	**cool-warm**
Contour Style	**Smooth Filled**

To have a better view of the graphics card temperature, hide the mesh scene:

8. Right-click the **Scenes** > **Mesh** node and select **Toggle Visibility**.
9. Save the simulation .

Preparing the Temperature Report

Graphics cards operate effectively within a certain operation temperature range. To check the maximum, minimum, and average temperature values of the graphics card, create a report.

To define a report for the graphics card temperature:

1. Right-click the **Results** > **Reports** node and select **Create Post Report**.
2. Within the *Edit* dialog, set the following properties:

Property	Setting
Name	`Graphics Card Temperature`
Report Field	**Temperature**
Report Items	**Graphics Card**

3. Click **OK**.
4. Save the simulation .

Running the Simulation

The simulation is now set up and ready to run.

To run the simulation:

1. Click (**Run**) in the *Solution* toolbar.

   The *Residuals* display is created automatically and shows the progress of the solver. While the simulation is running, you can click the *Electronics Cooling 1* tab at the top of the Graphics window to visualize the solution.

2. After the run is finished, save the simulation .

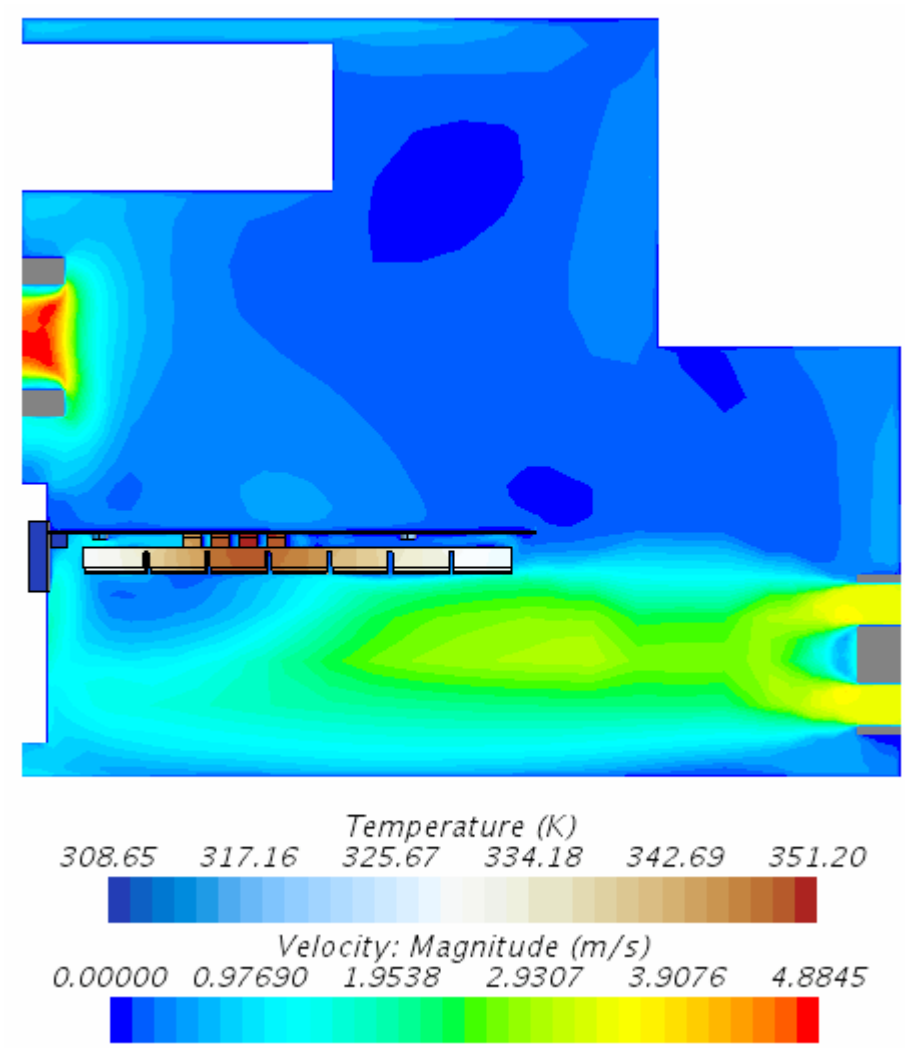
Analyzing the Results

View the results of the converged simulation and determine the maximum, minimum, and average temperatures from the graphics card.

After the run is finished, examine the velocity and temperature displays:

1. Click the *Electronics Cooling 1* tab at top of the Graphics window.

   The following screenshot displays the air flow inside the PC tower case and the temperature on the graphics card.



To extract the maximum, minimum, and average temperature values from the graphics card:

2. Right-click the **Results** > **Reports** > **Graphics Card Temperature** node and select **Tabulate**.

   The *Tabular Data* dialog, displays the following temperature data:

QP/QPS	Field	Minimum	Maximum	Average
GraphicsCard	Temperature	308.687 K	351.207 K	336.089 K

Summary

The Electronics Cooling Toolset: Graphics Card Cooling tutorial uses a graphics card in a PC tower case to introduce various features of the Electronics Cooling Toolset.

The features that were introduced include:

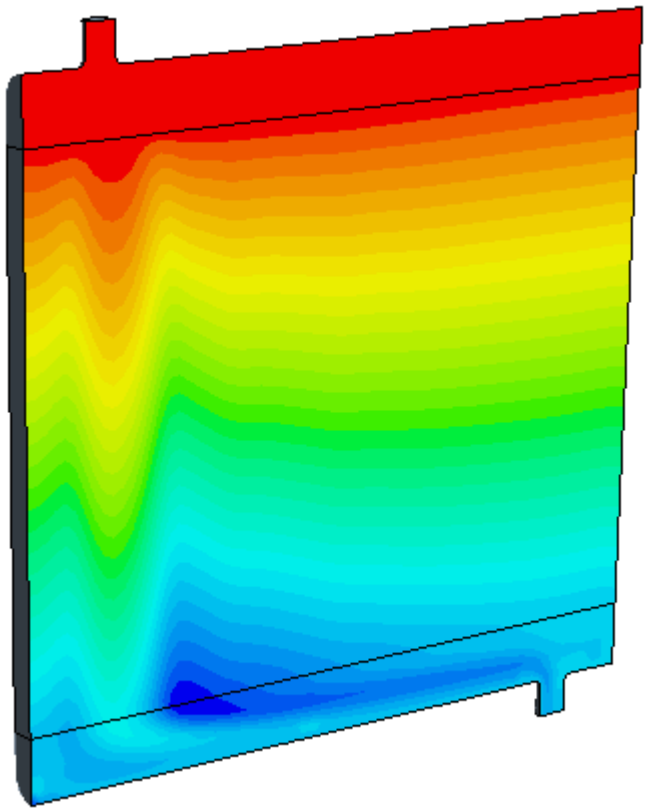
- Invoking the Electronics Cooling Toolset.
- Defining the setup for a forced convection driven flow.
- Importing a CAD model.

- Setting up geometry and physics for imported objects using Custom QuickParts.
- Creating axial fans from scratch using templated Axial Fan QuickParts.
- Importing a fan performance curve.
- Extracting the air volume.
- Meshing the solution domain.
- Preparing scenes and reports for post-processing.
- Running the simulation.
- Analyzing the flow and temperature field and reporting temperature data.

Dual Stream Heat Exchanger: Car Radiator

This tutorial simulates a three-dimensional fluid flow through an idealized water-cooled car radiator.

The purpose of this tutorial is to demonstrate the heat exchanger capabilities of STAR-CCM+, where the radiator transfers heat from the liquid coolant to the outside air.



Two key operations are demonstrated in this tutorial, which uses the trimmed cell method for the volume mesh:

- The mesh alignment option is activated so that a conformal match is obtained across mesh interfaces. STAR-CCM+ does not automatically generate a conformal mesh across interfaces for the trimmed cell mesh type.
- The core region of the radiator is duplicated, which is an essential requirement for the heat exchanger model in STAR-CCM+. Some modification of the interfaces is required to make sure that the correct regions are connected after the duplication.

Contents:

Prerequisites

Importing the Surface Mesh and Naming the Simulation

Assigning Parts to Regions
Generating the Volume Mesh
Preparing Boundaries and Interfaces
Selecting Physics Models
Assigning Physics Continua
Setting the Boundary Conditions
Specifying Porosity Coefficients
Creating the Actual Heat Exchanger Interface
Reporting and Monitoring with Plots
Setting the Stopping Criteria
Visualization Scenes
Running the Simulation
Visualizing Results
Summary

Prerequisites

The instructions in the Dual Stream Heat Exchanger tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Importing the Surface Mesh and Naming the Simulation

To set up the STAR-CCM+ simulation, launch a simulation and import the supplied surface mesh.

A surface mesh for a car radiator has been prepared for this analysis.

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Surface Mesh**.
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/heatTransferAndRadiation/data`.
5. Select `radiator_parts.x_b` then click **Open**.
6. To accept the default *Import Surface Options*, click **OK**.

   STAR-CCM+ provides feedback on the import process in the *Output* window. A geometry scene is also created in the *Graphics* window.
7. Save the simulation as `radiator_hx.sim`.
8. Copy the `uag_table.csv` file to the current working directory.

Assigning Parts to Regions

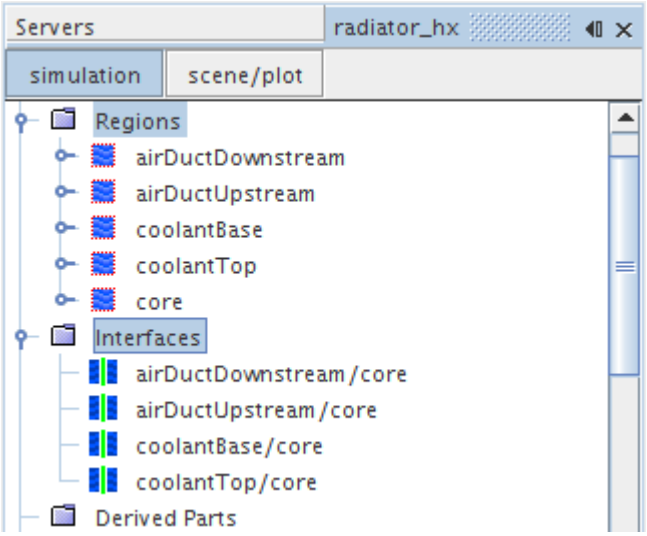
Five fluid regions are required for this simulation: Two for the air ducts, two for the coolant, and one for the core.

Assign all the parts to new regions:

1. Multi-select all the parts within the **Geometry** > **Parts** node.
2. Right-click one of the selected parts and select **Assign Parts to Regions**.
3. In the *Assign Parts to Regions* dialog, set the modes as follows:

Mode	Setting
Region mode	**Create a Region for Each Part**
Boundary mode	**Create a Boundary for Each Part Surface**

4. Click **Apply** and then click **Close**.
5. Expand the **Regions** and **Interfaces** nodes.



The required regions, boundaries, and interfaces corresponding to the imported parts are all created. You can see that there are four interfaces corresponding to the four sets of part/core contact data.

Generating the Volume Mesh

Create a volume mesh for the car radiator.

Activate the surface remesher and trimmed cell mesher to generate the volume mesh.

1. Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
2. In the *Create Automated Mesh Operation* dialog:
   a) From the *Parts* list, select all parts.
   b) From the *Select Meshers* group box, select the following meshers in order:

Group	Mesher
Surface Meshers	**Surface Remesher**
Core Volume Meshers	**Trimmed Cell Mesher**

In this tutorial, the meshing is done per part.

3. Select the **Operations** > **Automated Mesh** node and activate *Per-Part Meshing*.
4. Select the **Automated Mesh** > **Meshers** > **Trimmed Cell Mesher** node and activate *Perform Mesh Alignment*.
5. Edit the **Automated Mesh** > **Default Controls** node and set the following properties:

Node	Property	Setting
Base Size	*Value*	`2.56 cm`
Mesh Alignment Location	*Location*	`[-0.166693, 0.293814, 0.126133] m`
Minimum Surface Size	*Percentage of Base*	`1`

Create a block shape part around the core. This shape is for the volumetric controls that define mesh refinement.

6. Right-click the **Geometry** > **Parts** node and select **New Shape Part** > **Block**.
7. In the *Create/Edit Block Part* dialog, enter the following settings in the *Maximum and Minimum Coordinates* box:

Property	Setting
Corner 1	`[-0.18477, -0.15894, 0.100453]` m
Corner 2	`[0.29612, 0.38635, 0.179]` m

8. Click **Create** and then **Close**.

Create a volumetric control and select the shape that defines the source:

9. Right-click the **Operations** > **Automated Mesh** > **Custom Controls** node and select **New** > **Volumetric Control**.

   A new node, **Volumetric Control**, is added below the **Custom Controls** node.
10. Edit the **Custom Controls** > **Volumetric Control** node and set the following properties:

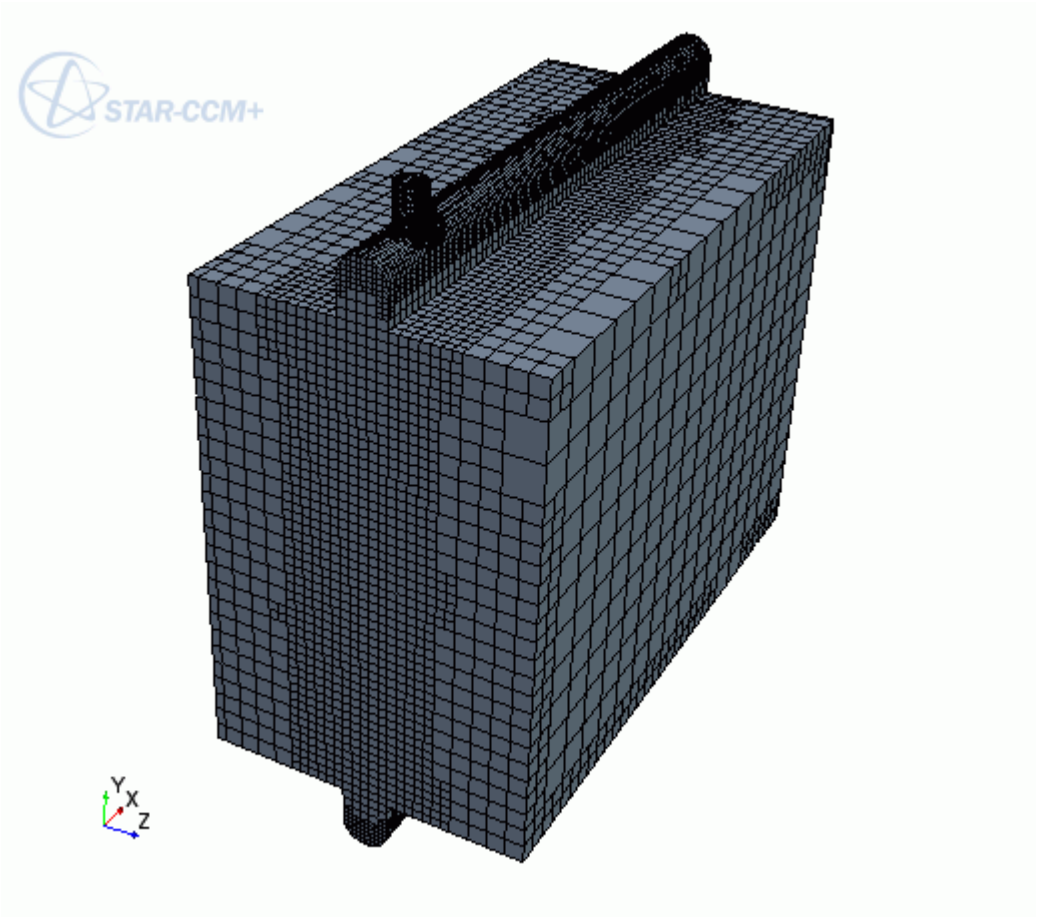
Node	Property	Setting
Volumetric Control	*Parts*	**Block**
└ **Controls**		
└ **Surface Remesher**	*Customize Size*	**Activated**
└ **Trimmer**	*Customize Isotropic Size*	**Activated**
└ **Values**		

Node	Property	Setting
└ **Custom Size**	*Size Type*	**Absolute**
	Value	`0.64 cm`

11. To start the mesh generation process, click (**Generate Volume Mesh**) in the **Mesh** toolbar.

    The output from the meshers is shown in the *Output* window. When the meshing process is complete, the message `Volume Meshing Pipeline Completed` is displayed in the *Output* window.

12. To view the resulting mesh, create a mesh scene.



13. Save the simulation.

Preparing Boundaries and Interfaces

STAR-CCM+ requires the existence of an interface between all regions to transfer the appropriate mass and energy quantities during the calculation.

Rename the radiator core and create a duplicate.

1. Rename the **Regions** > **core** node to `air_core`.

2. Create a copy of the **air_core** node and rename it `coolant_core`.
3. Expand the **Interfaces** node and delete the **coolantBase/core** and **coolantTop/core** nodes.

Create the required interfaces.

4. Expand the **Regions** node and create two interfaces:
   a) Create the first interface between the **coolant core** > **Boundaries** > **base_face** node and the **coolantBase** > **Boundaries** > **Coolant_base_interface** node.
   b) Create the second interface between the **coolant core** > **Boundaries** > **top_face** node and the **coolantTop** > **Boundaries** > **Coolant_top_interface** node.

   By default, the **Interface 1** and **Interface 2** interfaces have the *Type* property set to `Internal Interface` and the *Topology* property set to `In-place`.

Selecting Physics Models

Physics models are required to define the behavior of the air and coolant continua.

Use the following steps:

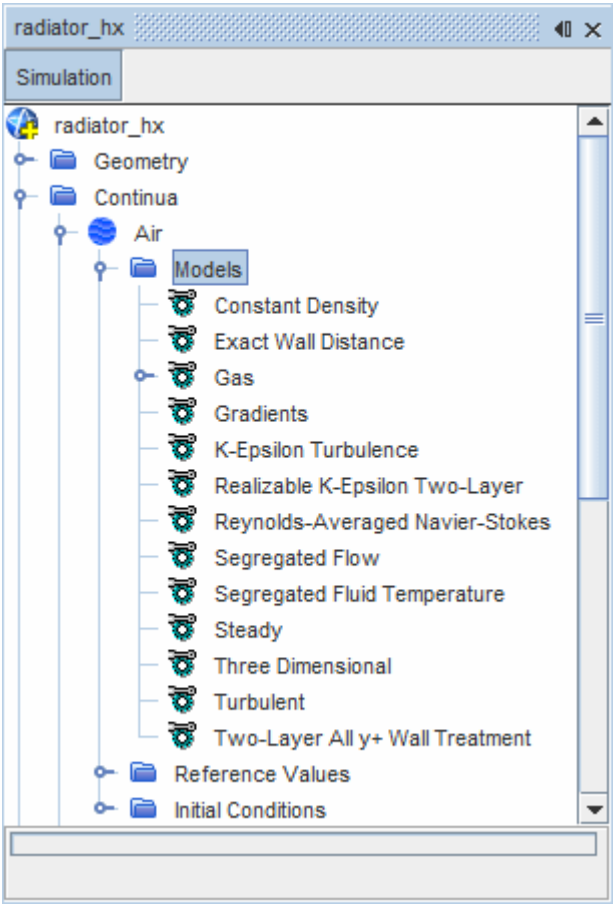
1. Rename the **Continua** > **Physics 1** node to `Air`.
2. For the physics continuum, **Continua** > **Air**, select the following models in order:

Group Box	Model
Enabled Models	**Three Dimensional** (Selected automatically)
Time	**Steady**
Material	**Gas**
Flow	**Segregated Flow**
	Gradients (Selected automatically)
Equation of State	**Constant Density**
Viscous Regime	**Turbulent**
	Reynolds-Averaged Navier-Stokes (Selected automatically)
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**
	Realizable K-Epsilon Two-Layer (Selected automatically)
	Two-Layer All y+ Wall Treatment (Selected automatically)
Wall Distance	**Exact Wall Distance** (Selected automatically)
Optional Models	**Segregated Fluid Temperature**

3. Click **Close**.

   The color of the **Air** node turns from gray to blue to indicate that air models have been activated and the continuum is valid.

4. To review the models, open the **Air** > **Models** node.



5. To specify the physics models for the coolant, create a physics continuum.
6. Rename the **Continua** > **Physics 1** node to `Coolant`.
7. For the physics continuum, **Continua** > **Coolant**, select the following models in order:

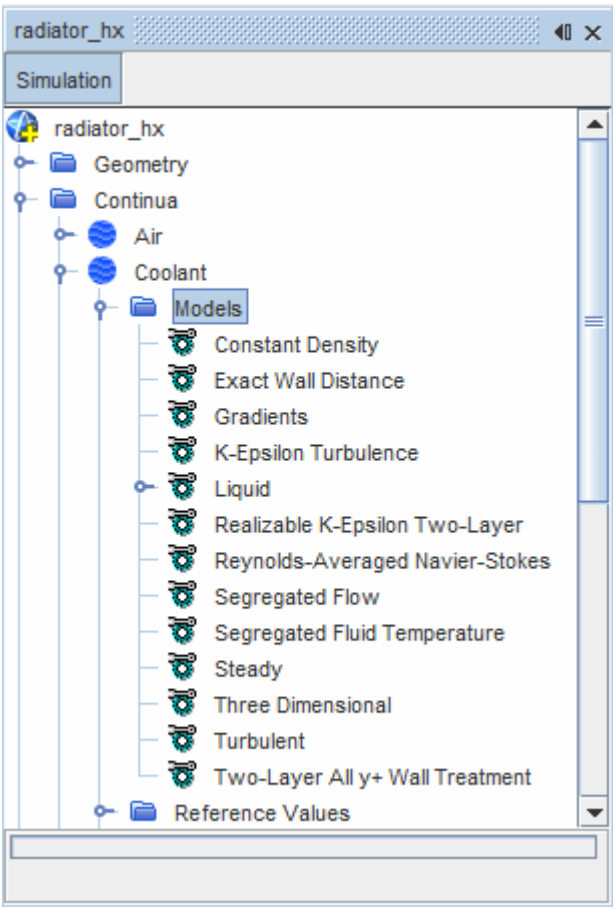
Group Box	Model
Space	**Three Dimensional**
Time	**Steady**
Material	**Liquid**
Flow	**Segregated Flow**
	Gradients (Selected automatically)
Equation of State	**Constant Density**
Viscous Regime	**Turbulent**
	Reynolds-Averaged Navier-Stokes (Selected automatically)
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**

Group Box	Model
	Realizable K-Epsilon Two-Layer (Selected automatically)
	Two-Layer All y+ Wall Treatment (Selected automatically)
Wall Distance	**Exact Wall Distance** (Selected automatically)
Optional Models	**Segregated Fluid Temperature**

8. Click **Close**.

   The color of the **Coolant** node turns from gray to blue to indicate that coolant models have been activated and the continuum is valid.

9. To review the models, open the **Coolant** > **Models** node.



Assigning Physics Continua

After physics continua have been defined, they can be assigned to their corresponding regions.

Use the following steps:

1. Edit the **Regions** node and set the following properties:

Node	Property	Setting
coolant_core	*Physics Continuum*	`Coolant`
coolantBase	*Physics Continuum*	`Coolant`
coolantTop	*Physics Continuum*	`Coolant`

2. Save the simulation.

Setting the Boundary Conditions

By default the boundaries representing the radiator are all set to no-slip walls. Some of them need to be changed to other boundary types.

Use the following steps:

1. Select the **Regions** > **airDuctUpstream** > **Boundaries** > **Air_inlet** node and set *Type* to `Velocity Inlet`.
2. Edit the **Air_inlet** > **Physics Values** node and set the properties as follows:

Node	Property	Setting
Static Temperature	*Value*	`308 K`
Velocity Magnitude	*Value*	`20 m/s`

3. Click **Close**.
4. Select the **Regions** > **airDuctDownstream** > **Boundaries** > **Air_pressure_out** node and set *Type* to `Pressure Outlet`.
5. Select the **Regions** > **coolantTop** > **Boundaries** > **Coolant_inlet** node and set *Type* to `Mass Flow Inlet`.
6. Edit the **Coolant_inlet** > **Physics Values** node and set the properties as follows:

Node	Property	Setting
Mass Flow Rate	*Value*	`1.8 kg/s`
Total Temperature	*Value*	`380 K`

7. Click **Close**.
8. Select the **Regions** > **coolantBase** > **Boundaries** > **Coolant_pressure_out** node and set *Type* to `Pressure Outlet`.

Specifying Porosity Coefficients

Define the air_core and coolant_core regions as porous and set their respective porosity coefficients.

Use the following steps:

1. Edit the **Regions** node and set the properties as follows:

Node	Property	Setting
air_core	*Type*	**Porous Region**
coolant_core	*Type*	**Porous Region**

2. Select the **air_core** > **Physics Values** > **Porous Inertial Resistance** node.

   The *Method* property is set to `Principal Tensor` by default. This setting is suitable because the constant resistance coefficients correspond to the principal (diagonal) components of the porous resistance tensor.

3. Edit the **Porous Inertial Resistance** > **Principal Tensor** node and set the properties as follows:

Node	Property	Setting
XX Component	*Value*	1e4 kg/m^4
YY Component	*Value*	1e4 kg/m^4
ZZ Component	*Value*	90 kg/m^4

Note: Do not change the **XX Axis** and **YY Axis** coordinates since the porous resistance components are specified in the directions of the default axes.

4. Edit the **Porous Viscous Resistance** > **Principal Tensor** node and set the properties as follows:

Node	Property	Setting
XX Component	*Value*	1e5 kg/m^3-s
YY Component	*Value*	1e5 kg/m^3-s
ZZ Component	*Value*	450 kg/m^3-s

5. Edit the **coolant_core** > **Physics Values** > **Porous Inertial Resistance** > **Principal Tensor** node and set the properties as follows:

Node	Property	Setting
XX Component	*Value*	1e8 kg/m^4
YY Component	*Value*	2e6 kg/m^4
ZZ Component	*Value*	1e8 kg/m^4

6. Edit the **coolant_core** > **Physics Values** > **Porous Viscous Resistance** > **Principal Tensor** node and set the properties as follows:

Node	Property	Setting
XX Component	*Value*	1e6 kg/m^3-s
YY Component	*Value*	5.5e4 kg/m^3-s
ZZ Component	*Value*	1e6 kg/m^3-s

Creating the Actual Heat Exchanger Interface

Specify the actual heat exchanger interface and use the actual flow dual stream option for the heat exchanger method.

The heat exchanger method uses the flow and energy values that are calculated for the simulation itself, to compute the mean temperature difference and hence the local heat transfer coefficient. You also set the heat exchanger data specification option to UAG Table. This option assumes that the overall heat transfer coefficient (times the heat exchanger area) across the entire heat exchanger is known as a table of the cold stream mass flow rate (say, from an experimental source).

To define the actual heat exchanger interface:

1. Multi-select the following nodes:
   - **Regions** > **air_core**
   - **Regions** > **coolant_core**
2. Right-click one of the nodes and select **Create Interface** > **Heat Exchanger**.
3. Select the **Interfaces** > **Heat Exchanger 1** node and review the region assignments. The cold region is **air_core** and the hot region is **coolant_core**.
4. Select the **Interfaces** > **Heat Exchanger 1** > **Physics Conditions** > **Heat Exchanger Method** node and set *Option* to `Actual Flow Dual Stream`.
5. Select the **Heat Exchanger Data Specification** node and set *Method* to `UAG Table`.
6. Load the file containing the data:
   a) From the simulation tree, right-click the Tools > Table node and select New Table > File Table.
   b) In the *Open* dialog, navigate to your working directory and select the `uag_table.csv` file.
   c) Click **Open**.
7. Select the **Interfaces** > **Heat Exchanger 1** > **Physics Values** > **UAG Table** node and set the following properties:

Property	Setting
UAG Table	`uag_table`
UAG Table: Mass Flow Rate	`column0`
UAG Table: UAG	`column1`
Cold Upstream Boundaries	`upstream_face [airDuctUpstream/core]`
Hot Upstream Boundaries	`top_face [Interface 2]`

8. Select the **Physics Values** > **Heat Exchanger First Iteration** node and set *First Heat Exchanger Iteration* to `90`.
9. Save the simulation.

Reporting and Monitoring with Plots

By default STAR-CCM+ produces a residuals plot which can be used to monitor the convergence of the solution. You can also create more reports, monitors, and plots to help in assessing convergence.

This part of the tutorial involves creating three reports:

- A surface area report for the outlet air temperature.
- A surface area report for the outlet coolant temperature.
- A heat exchanger report.

Then you create monitors and plots for the above reports and use the plots to monitor the solution during the run.

1. Right-click **Reports** and select **New Report** > **Surface Average**.
2. Rename the new report to `Air Temperature Out`.
3. Select the **Air Temperature Out** node and set the following properties:

Property	Setting
Scalar Field Function	`Temperature`
Parts	`Regions > airDuctDownstream > Air_pressure_out`

4. Create another **Surface Average** report.
5. Rename this report to `Coolant Temperature Out`.
6. Select the **Coolant Temperature Out** node and set the following properties:

Property	Setting
Scalar Field Function	`Temperature`
Parts	`Regions > coolantBase > Coolant_pressure_out`

7. Create a **Heat Exchanger (Dual Stream)** report.

   The **Heat Exchanger (Dual Stream) 1** node is selected automatically.
8. Set the following properties:

Property	Setting
Heat Exchanger	`Heat Exchanger 1`
Cold Downstream Boundaries	`downstream_face [airDuctDownstream/core]`
Hot Downstream Boundaries	`base_face [Interface 1]`

9. Create monitors and plots for the reports:
   a) Select the **Air Temperature Out** and **Coolant Temperature Out** nodes.
   b) Right-click and select **Create Monitor and Plot from Report**.
   c) In the *Create Plot From Reports* dialog, select **Single Plot**.

d) Create a monitor and plot for the **Heat Exchanger (Dual Stream) 1** report.

e) Rename the **Reports Plot** node to `Temperature Plot`.

10. Open the **Heat Exchanger (Dual Stream) 1 Monitor Plot** and **Temperature Plot** plots.

Setting the Stopping Criteria

This is the point at which you would adjust the maximum number of iterations.

To set the stopping criteria:

1. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to `1000`.
2. Save the simulation.

Visualization Scenes

Create a scalar scene to visualize temperature at a plane section through the air and coolant parts, and a vector scene to display the velocity vectors at a plane section through the coolant part.

Use the following steps:

1. Create a scalar scene.

Create the two plane sections.

2. Create a plane section derived part.
3. In the *Create Plane Section* inset panel, make the following settings:

Property	Setting
Input Parts	`[air_core, airDuctDownstream, airDuctUpstream]`
Plane Parameters	*origin:* `[0 m, 0 m, 0 m]` *normal:* `[1 m, 0 m, 0 m]`
Display	`No Displayer`

4. Click **Create** and then **Close**.
5. Rename the **Derived Parts** > **plane section** node to `Section Air`.
6. Create a second plane section with the following properties:

Property	Setting
Input Parts	`[coolant_core, coolantBase, coolantTop]`
Plane Parameters	*origin:* Default values *normal:* `[0 m, 0 m, 1 m]`
Display	`No Displayer`

7. Click **Create** and then **Close**.

8. Rename the **Derived Parts** > **plane section** node to `Section Coolant`.
9. Edit the **Scenes** > **Scalar Scene 1** > **Displayers** > **Outline 1** > **Parts** node.
10. In the *Edit* dialog, deselect all regions, then open the **Derived Parts** node and select **Section Air** and **Section Coolant**.
11. Click **OK**.
12. Copy the **Outline 1** > **Parts** node to **Scalar 1** > **Parts** node.
13. Select the **Scalar 1** > **Scalar Field** node and set *Function* to `Temperature`.
14. Select the **Scalar 1** node and set *Contour Style* to `Smooth Filled`.

Create a vector scene. For this scene, you leave the outline 1 parts all included.

15. Edit the **Scenes** > **Vector Scene 1** > **Displayers** > **Vector 1** > **Parts** node.
16. In the *Edit* dialog, select the **Derived Parts** > **Section Coolant** node.
17. Click **OK**.
18. Save the simulation.

Running the Simulation

The simulation is now ready to run.

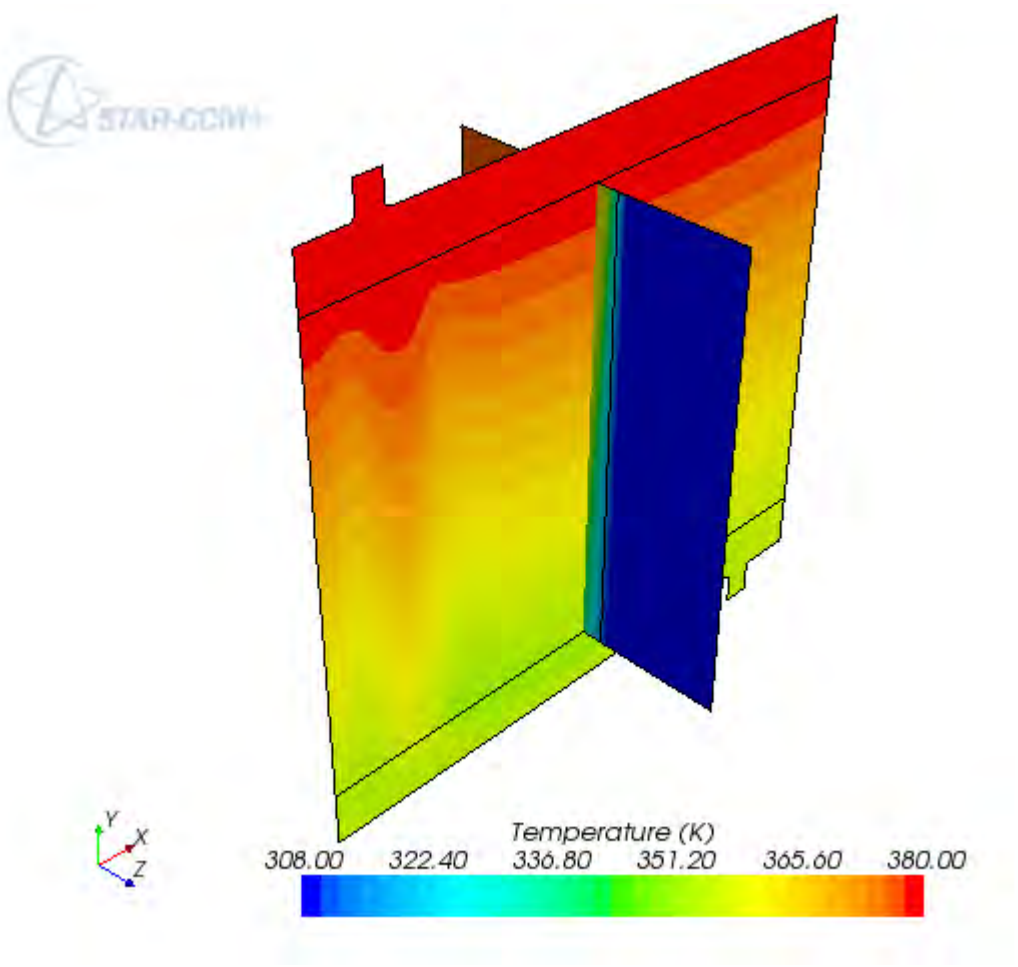
Use the following steps:

1. Run the simulation.
2. To watch the solution being updated while the simulation is running, look at the scalar scene, vector scene, and plot displays.
3. When the simulation has finished running, save it.

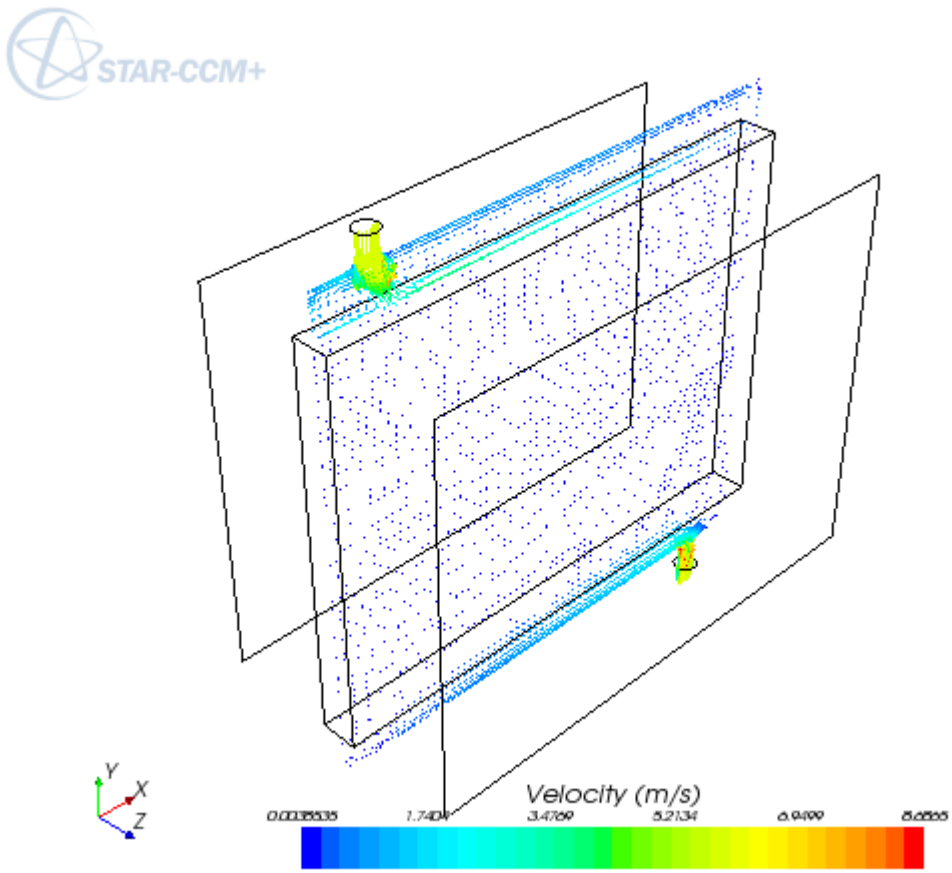
Visualizing Results

To examine temperatures, velocity, and other results from the converged solution, look at the scenes and the plots.

1. To see the temperature distribution, look at the *Scalar Scene 1* display.



2. Look at the *Vector Scene 1* display.



3. Open the *Heat Exchanger (Dual Stream) 1 Monitor Plot* display.
4. Edit the **Plots** > **Heat Exchanger (Dual Stream) 1 Monitor Plot** > **Axes** node and set the following properties:

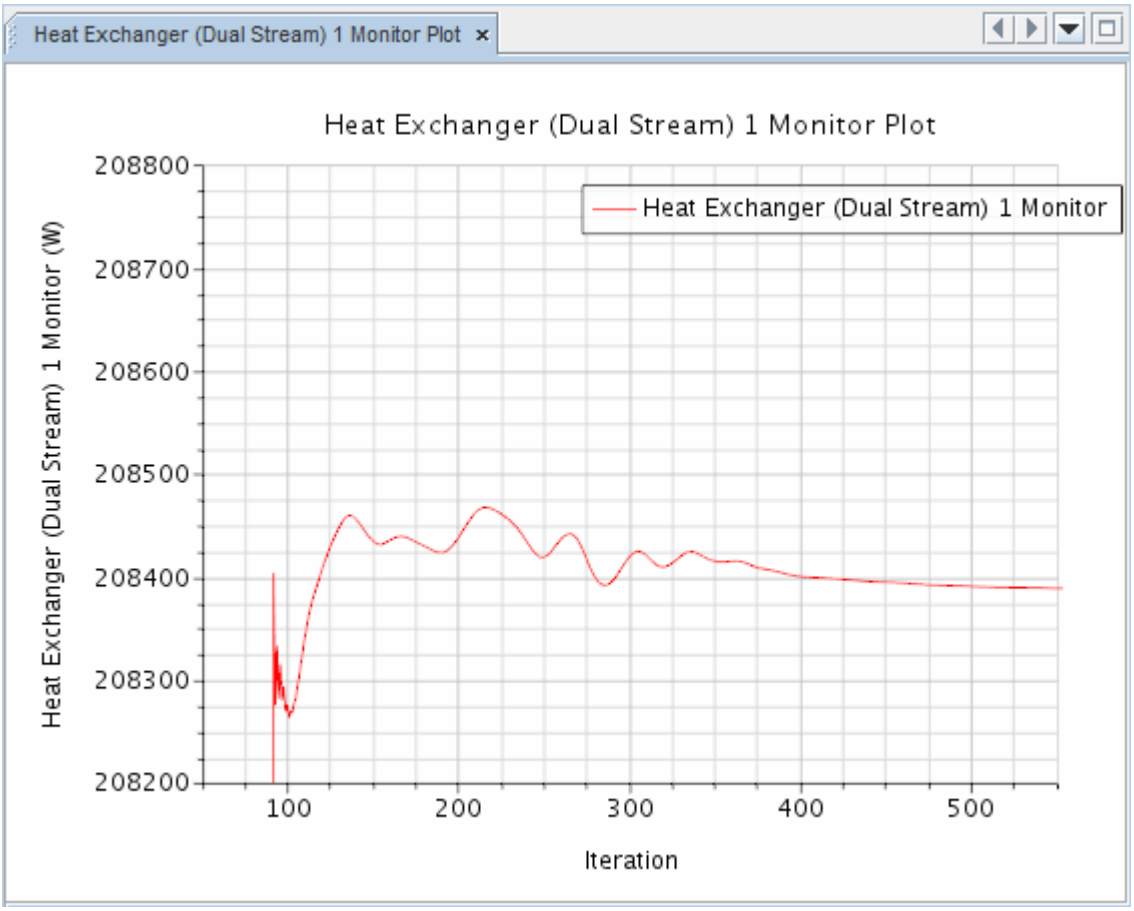
Bottom Axis

Property	Setting
Minimum	50
Maximum	550

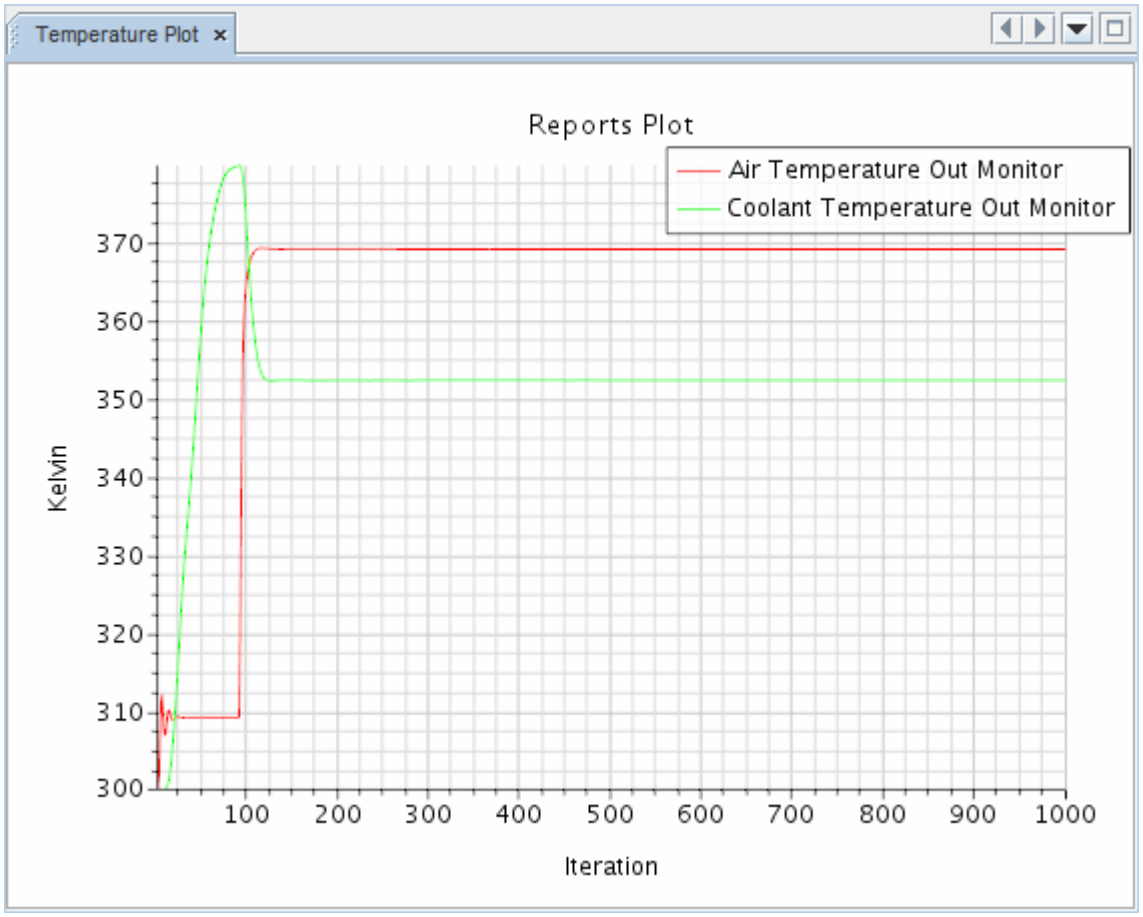
Left Axis

Property	Setting
Minimum	208200
Maximum	208800

The graph appears similar to the example that is shown below.



5. To see the air and coolant temperatures at the outlet for the converged solution, open the *Temperature Plot* display.



Summary

The Dual Stream Heat Exchanger tutorial used an idealized water-cooled car radiator to introduce a variety of STAR-CCM+ features.

The features that were introduced include:

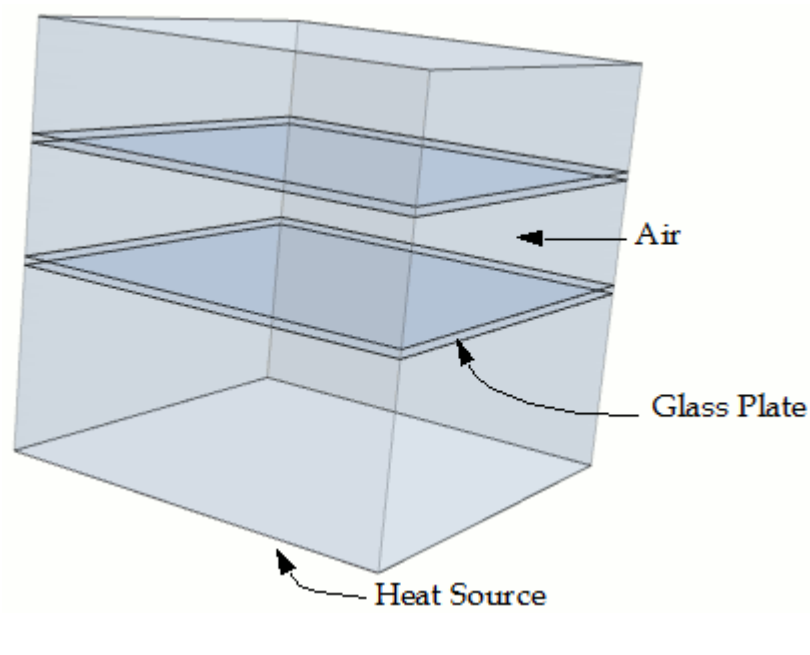
- Selecting fluid physics models.
- Creating and assigning a physics continuum.
- Specifying porosity coefficients.
- Creating an actual heat exchanger interface.
- Specifying the heat exchanger data specification using a UAG table.
- Creating reports, monitors, and plots.
- Running the simulation.
- Visualizing the results.

Surface-to-Surface Radiation: Thermal Insulator

This tutorial incorporates gray thermal radiation in a STAR-CCM+ simulation.

All conditions and properties are defined using the STAR-CCM+ GUI and solved using the segregated solver. The results are displayed using the post-processing tools available in STAR-CCM+.

The starting point for the tutorial is a three-dimensional hexahedral grid representing two glass plates within a rectangular enclosure containing air, as shown in the Figure below. Built with pro-STAR, the grid includes interfaces between the glass plates and surrounding air. The imported file contains information about the geometry, mesh, boundaries, and materials present in the problem.



The analysis covers the following heat transfer modes:

- Conduction
- Gray thermal radiation

Contents:

Prerequisites
Importing the Mesh and Naming the Simulation
Splitting the Boundary
Renaming Regions and Boundaries
Creating Interfaces
Setting Up the Models
Setting Material Properties
Setting Boundary Conditions and Values
Setting Solver Parameters and Stopping Criteria

Visualizing the Solution
Reporting and Monitoring with Plots
Initializing and Running the Simulation
Visualizing the Results
Summary

Prerequisites

The instructions in the Surface-to-Surface Radiation tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Importing the Mesh and Naming the Simulation

To set up the STAR-CCM+ simulation, launch a simulation and import the supplied volume mesh.

Use the following steps:

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh**.
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/heatTransferAndRadiation/data`.
5. Select `insulator.ccm` then click **Open**.

   STAR-CCM+ provides feedback on the import process in the *Output* window and two mesh regions are created under the **Regions** node.
6. Save the simulation as `thermalInsulator.sim`.

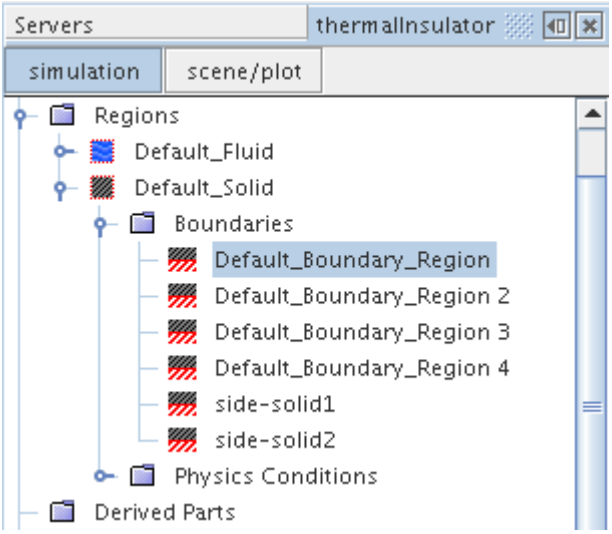
Splitting the Boundary

One of the solid boundaries must be split into non-contiguous boundaries.

Use the following steps:

1. Expand the **Regions** > **Default_Solid** > **Boundaries** node.
2. Right-click the **Default_Boundary_Region** node and select **Split Non-Contiguous** from the pop-up menu.

   The boundary is split into four and three new boundary nodes are created as shown.



3. Save the simulation.

Renaming Regions and Boundaries

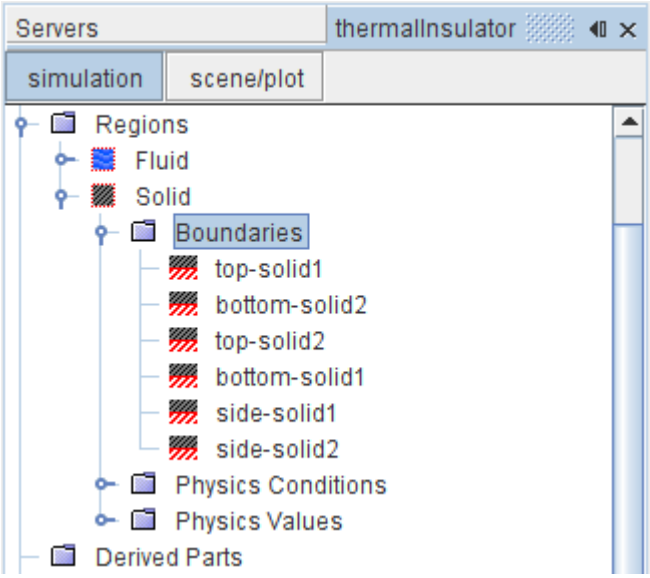
The simulation currently has the default region and boundary names. Choose more meaningful names.

Use the following steps:

1. Expand the **Regions** node.
2. Rename **Default_Fluid** to `Fluid`.
3. Rename **Default_Solid** to `Solid`.
4. Rename the boundaries within the **Solid** node:

Original Name	New Name
Default_Boundary_Region	`top-solid1`
Default_Boundary_Region 2	`bottom-solid2`
Default_Boundary_Region 3	`top-solid2`
Default_Boundary_Region 4	`bottom-solid1`

This portion of the object tree looks as shown below.



5. Save the simulation.

Creating Interfaces

STAR-CCM+ requires the existence of interfaces between different regions to transfer the appropriate physical quantities during the calculation.

For this example, you create a contact-type interface between the fluid and solid regions. This involves creating a so-called “in-place interface” between the wall boundaries enclosing each of the two regions, and then defining this interface as being of an appropriate “contact” type for fluid-solid heat transfer.

1. Using the multi-select method, select the following nodes:
   - **Regions** > **Fluid** > **Boundaries** > **top-solid1**
   - **Regions** > **Solid** > **Boundaries** > **top-solid1**
2. Right-click and select **Create Interface**.

   The previous step creates the following new nodes:

   - **Fluid** > **Boundaries** > **top-solid1 [Interface 1]**
   - **Solid** > **Boundaries** > **top-solid1 [Interface 1]**

   These boundaries form the connection between the fluid and solid regions once the flow initialization step has been performed and supersede the original wall boundaries forming the connection.
3. Select the following nodes:
   - **Regions** > **Fluid** > **Boundaries** > **bottom-solid1**
   - **Regions** > **Solid** > **Boundaries** > **bottom-solid1**
4. Right-click and select **Create Interface**.
5. Select the following nodes:
   - **Regions** > **Fluid** > **Boundaries** > **top-solid2**
   - **Regions** > **Solid** > **Boundaries** > **top-solid2**
6. Right-click and select **Create Interface**.

7. Select the following nodes:
   - **Regions** > **Fluid** > **Boundaries** > **bottom-solid2**
   - **Regions** > **Solid** > **Boundaries** > **bottom-solid2**
8. Right-click and select **Create Interface**.

   In addition to the new in-place boundaries, a folder node named **Interfaces** appears in the object tree. Within it, four interface nodes that are called **Interface 1** to **Interface 4** are created.
9. Save the simulation.

Setting Up the Models

Models define the primary variables of the simulation, including pressure, temperature, velocity, and what mathematical formulation is used to generate the solution. In this example, the flow is laminar and incompressible.

The Segregated Energy model is used together with the Surface-to-Surface Radiation model to determine the temperature field.

As there are two materials (air and glass) present, two regions are required for the analysis. You define two sets of Physics continuum models, each appropriate to a different region. By default, **Physics 1** and **Physics 2** are automatically created when the mesh is imported. You use these continua for the fluid and the solid.

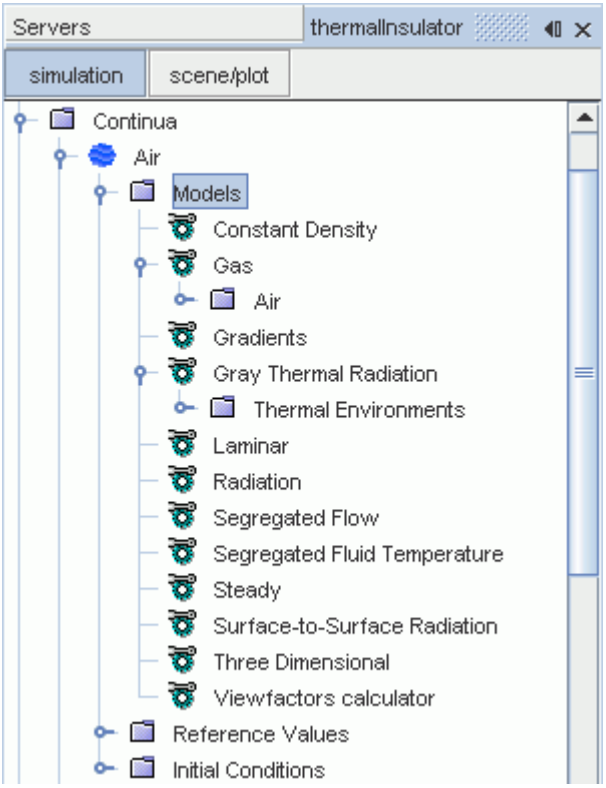
1. Rename the **Continua** > **Physics 1** node to `Air`.
2. Rename the **Continua** > **Physics 2** node to `Glass`.
3. For the physics continuum, **Continua** > **Air**, select the following models in order:

Group Box	Model
Enabled Models	**Three Dimensional** (Selected automatically)
Time	**Steady**
Material	**Gas**
Flow	**Segregated Flow** **Gradients** (Selected automatically)
Equation of State	**Constant Density**
Viscous Regime	**Laminar**
Optional Models	**Segregated Fluid Temperature** **Radiation**
Radiation	**Surface-to-Surface Radiation** **View Factors Calulator** (Selected automatically)
Radiation Spectrum	**Gray Thermal Radiation**

4. Click **Close**.

   The color of the **Air** node turns from gray to blue to indicate that air models have been activated and the continuum is valid.

5. To review the models, open the **Air** > **Models** node.



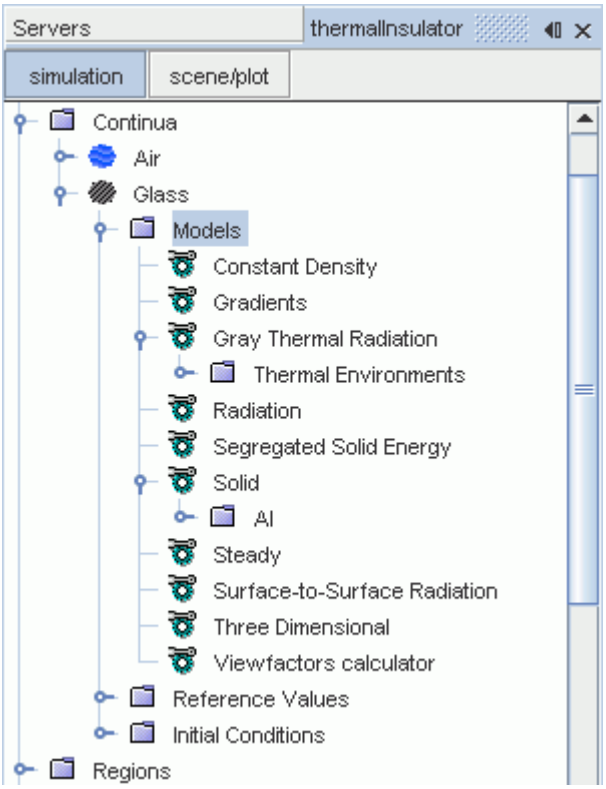
6. For the physics continuum, **Continua** > **Glass**, select the following models in order:

Group Box	Model
Enabled Models	**Three Dimensional** (Selected automatically)
Time	**Steady**
Material	**Solid**
Optional Models	**Segregated Solid Energy** **Gradients** (Selected automatically) **Radiation**
Equation of State	**Constant Density**
Radiation	**Surface-to-Surface Radiation** **View Factors Calulator** (Selected automatically)
Radiation Spectrum	**Gray Thermal Radiation**

7. Click **Close**.

   The color of the **Glass** node has turned from gray to shaded gray to indicate that models have been activated for this continuum.

8. To review the models, open the **Glass** > **Models** node.



9. Save the simulation.

Setting Material Properties

Change the default material for the solid continuum to glass.

By default, glass has a density of 2500 kg/m^3, a specific heat of 840 J/kg-K and a thermal conductivity of 1.4 W/m-K. The surrounding air is considered incompressible: by default, it has a density of 1.18415 kg/m^3, a specific heat of 1003.62 J/kg-K, a thermal conductivity of 0.0260305 W/m-K, and a dynamic viscosity of 1.85508E-5 Pa-s.

1. Right-click the **Continua** > **Glass** > **Models** > **Solid** > **Al** node and select **Replace with...**.
2. In the *Replace Material* dialog, expand the **Material Databases** > **Standard** > **Solids** node. Select **Glass (Glass Plate)** and click **OK**.

Setting Boundary Conditions and Values

The thermal insulator consists of a closed heated box with two internal solid glass panels that are surrounded by air. The boundary specifications that are required for this analysis are wall thermal conditions and radiation properties at the boundaries.

A fixed heat flux of 1000 W/m^2 is applied to the base of the box to provide the heat source. The top of the box is given a fixed temperature of 300 K which corresponds to the temperature of the environment. The vertical sides of the solid panels which are in contact with the environment is set to a fixed heat flux of -62.5 W/m^2. All

other wall boundary conditions are considered to be adiabatic, which is the default condition. Contact-type conducting boundaries are also used at solid-fluid interfaces.

Optical radiation properties emissivity, reflectivity, transmissivity, and radiation temperature must also be set at all boundaries. The solver assumes Kirchhoff's Law. Make sure that, for each boundary or interface, emissivity + reflectivity + transmissivity = 1.

Transmissivity of the solid-fluid interfaces is defined at interfaces.

Start with the wall boundary definitions for the fluid region.

1. Edit the **Regions** > **Fluid** > **Boundaries** > **collector** node.
2. Set the following properties:

   STAR-CCM+ adds the decimal point and units automatically, so by simply entering `1000` you get an entry of `1000.0 W/m^2`. A positive value for heat flux indicates that heat is entering the domain while a negative value would indicate that heat is leaving the domain.

Node	Property	Setting
Physics Conditions		
⌊ **Thermal Specification**	*Condition*	**Heat Flux**
Physics Values		
⌊ **Heat Flux**	*Method*	**Constant**
	Value	`1000` W/m^2
⌊ **Surface Emissivity**	*Method*	**Constant**
	Value	`1`
⌊ **Surface Reflectivity**	*Method*	**Auto Calculate**
⌊ **Surface Transmissivity**	*Method*	**Constant**
	Value	`0`

The conditions for the **side-1**, **side-2**, and **side-3** wall boundaries (side walls of the fluid region) are left at the default adiabatic setting but the optical properties at these boundaries must be set. First, set the **side-1** wall boundary.

3. Edit the **Regions** > **Fluid** > **Boundaries** > **side-1** > **Physics Values** node.
4. Set the following properties:

Node	Property	Setting
Surface Emissivity	*Method*	**Constant**
	Value	`0.95`
Surface Reflectivity	*Method*	**Auto Calculate**
Surface Transmissivity	*Method*	**Constant**
	Value	`0`

For this boundary, the application of Kirchhoff's law gives a surface reflectivity of `0.05`.

The conditions for the **side-2** and **side-3** wall boundaries are identical to the conditions for the **side-1** boundary. They can be set up in the same way manually or copied and pasted from the **side-1** boundary.

5. Copy from the **Regions** > **Fluid** > **Boundaries** > **side-1** node and paste onto the **side-2** and **side-3** nodes.

   If you prefer, examine the properties of the **side-2** and **side-3** nodes to verify that the radiative boundary conditions are correct.

Set the conditions for the **top** wall boundary.

6. Edit the **Regions** > **Fluid** > **Boundaries** > **top** node.
7. Set the following properties:

Node	Property	Setting
Physics Conditions		
∟ **Thermal Specification**	*Condition*	**Temperature**
Physics Values		
∟ **Static Temperature**	*Method*	**Constant**
	Value	`300 K`
∟ **Surface Emissivity**	*Method*	**Constant**
	Value	`0.8`
∟ **Surface Reflectivity**	*Method*	**Auto Calculate**
∟ **Surface Transmissivity**	*Method*	**Constant**
	Value	`0.15`

The boundary conditions must be set for the solid-fluid interfaces. The emissivity and reflectivity of the solid-fluid interfaces must be specified at each **Interface [Interface #]** boundary, not the **Interface** boundary, as the latter became defunct when the interface was created (see the previous section Creating Interfaces).

8. Edit the **Regions** > **Fluid** > **Boundaries** node.
9. Set the following properties:

Node	Property	Setting
bottom-solid1 [Interface 2]		
∟ **Physics Values**		
∟ **Surface Emissivity**	*Method*	**Constant**
	Value	`0.775`
∟ **Surface Reflectivity**	*Method*	**Auto Calculate**

Node	Property	Setting
		The **Auto Calculate** option for the **Surface Reflectivity** results in a reflectivity of `0.075` during the simulation.

Node	Property	Setting
bottom-solid2 [Interface 4]		
└ **Physics Values**		
└ **Surface Emissivity**	*Method*	**Constant**
	Value	`0.725`
└ **Surface Reflectivity**	*Method*	**Auto Calculate**

Node	Property	Setting
top-solid1 [Interface 1]		
└ **Physics Values**		
└ **Surface Emissivity**	*Method*	**Constant**
	Value	`0.8`
└ **Surface Reflectivity**	*Method*	**Auto Calculate**

Node	Property	Setting
top-solid2 [Interface 3]		
└ **Physics Values**		
└ **Surface Emissivity**	*Method*	**Constant**
	Value	`0.75`
└ **Surface Reflectivity**	*Method*	**Auto Calculate**

It is also necessary to set the emissivity, reflectivity, and transmissivity for interface boundaries in the solid region. For this tutorial, it is acceptable to set the emissivity and reflectivity to the same values on the solid side of the interface as on the fluid side. The copy and paste operation is therefore sufficient.

10. For each of the boundary names listed below, copy from the node in **Regions** > **Fluid** > **Boundaries**, and paste onto the boundary node with the matching name in **Regions** > **Solid** > **Boundaries**:
    - **bottom-solid1 [Interface 2]**
    - **bottom-solid2 [Interface 4]**
    - **top-solid1 [Interface 1]**
    - **top-solid2 [Interface 3]**

Set the solid wall boundary conditions and values in a similar way to the fluid ones.

11. Edit the **Regions** > **Solid** > **Boundaries** > **side-solid1** node.
12. Set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Thermal Specification**	*Condition*	**Heat Flux**
Physics Values		
└ **Heat Flux**	*Method*	**Constant**
	Value	`-62.5` W/m^2 The negative value indicates that heat is leaving the domain.
└ **Surface Emissivity**	*Method*	**Constant**
	Value	`0`
└ **Surface Reflectivity**	*Method*	**Constant**
	Value	`1`
└ **Surface Transmissivity**	*Method*	**Constant**
	Value	`0`

The conditions for wall boundary **side-solid2** are the same as for **side-solid1** and can be copied and pasted from the **side-solid1** boundary.

13. Copy from the **side-solid1** node and paste onto the **side-solid2** node.

Set the surface transmissivity at the interfaces.

14. Expand the **Interfaces** node.
15. In each interface node, select the **Physics Values** > **Surface Transmissivity** node and set *Value* as follows for each interface:

Interface Node	Surface Transmissivity Value
Interface 1	`0.15`
Interface 2	`0.15`
Interface 3	`0.175`
Interface 4	`0.175`

16. Save the simulation.

Setting Solver Parameters and Stopping Criteria

Adjust solver controls such as the under-relaxation factors and the maximum number of iterations.

Since you are only solving for temperature in this tutorial, the flow is frozen. In most situations, the Freeze Flow feature is primarily a “debugging” tool and not supported for normal simulation work.

1. Select the **Solvers** > **Segregated Flow** node and activate *Freeze Flow*.
2. To assist solution convergence, select the **Solvers** > **Segregated Energy** node and set the under-relaxation factors for fluid and solid segregated energy as follows:

Property	Setting
Fluid Under-Relaxation Factor	`0.55`
Solid Under-Relaxation Factor	`0.99`

The default stopping criterion is set to end the calculation after 1000 iterations. Such a value is excessive, so reduce the maximum number of iterations.

3. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to `500`.

Create a stopping criterion that is based on the energy equation residual. Set the minimum tolerance to 1E-6.

4. Right-click the **Stopping Criteria** node and select **Create New Criterion** > **From Monitor**, then select **Energy** in the *Select Monitor* dialog.
5. Select **Stopping Criteria** > **Energy Criterion** > **Minimum Limit** node and set *Minimum Value* to `1.0E-6`
6. Save the simulation.

Visualizing the Solution

View the temperature of the fluid and solid regions as the solution develops.

Start by creating a plane section part for display.

1. Create a plane section derived part.
2. In the *Create Section* inset panel, make the following settings:

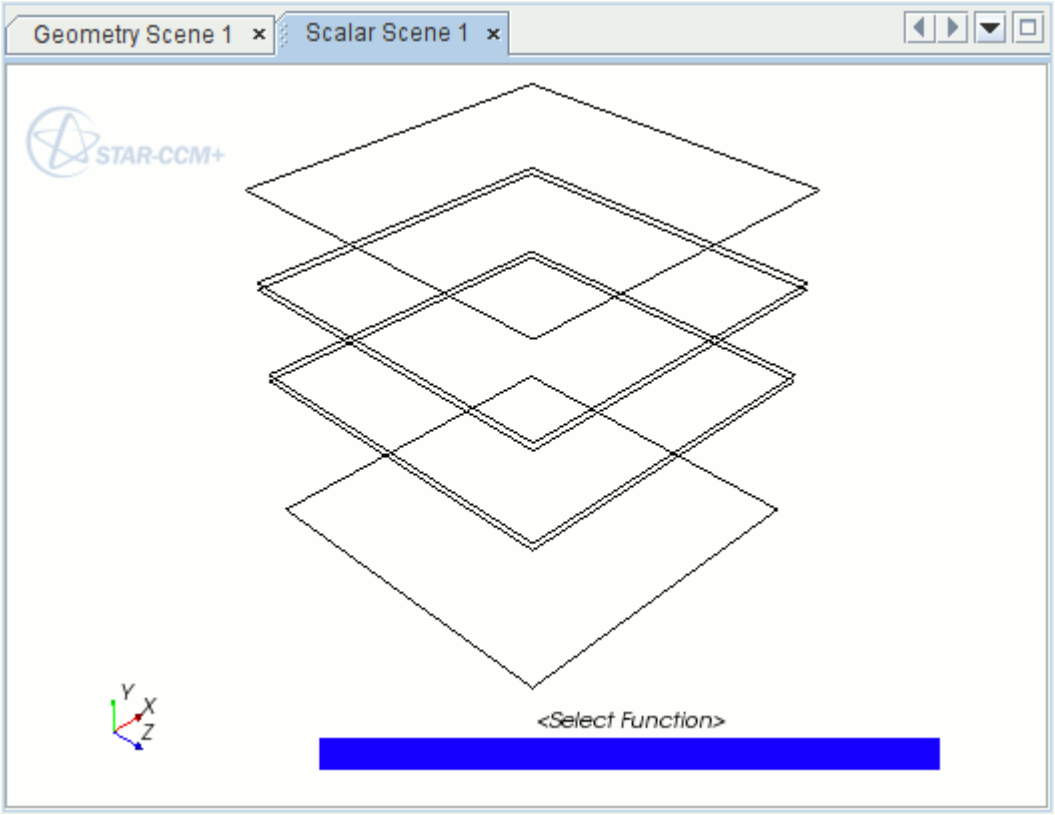
Property	Setting
Input Parts	`[Fluid, Solid]`
Plane Parameters	*origin:* `[0.5 m, 0.5 m, 0.5 m]` *normal:* `[1 m, 0 m, 0 m]`
Display	`Existing Displayer`

3. Click **Create** and then **Close**.

   A node appears within the **Derived Parts** node: **Plane section**.

The plane section part can now be added to any suitable scene or be displayed in its own scene if desired. Create a scalar scene and display the plane section and other parts colored by the temperature scalar value.

4. Create a scalar scene.
5. To zoom in and position the part as shown below, use the mouse buttons and click (**Save-Restore-Select views**).



6. Edit the **Scenes** > **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Parts** node.
7. In the *Edit* dialog, do the following:
   a) Expand the **Regions** > **Fluid** > **Boundaries** node and the **Derived Parts** node.
   b) Select the following parts:
      - **top**
      - **bottom-solid1 [Interface2]**
      - **top-solid2 [Interface3]**
      - **collector**
      - **plane section**
8. Click **OK**.
9. Right-click the scalar bar in the *Graphics* window display. In the pop-up menu that appears, select **Temperature**.
10. Look at the *Scalar Scene 1* display to verify that it contains the thermal insulator geometry along with a color bar for temperature.
11. Save the simulation.

Reporting and Monitoring with Plots

In this case, a report and plot are used for monitoring the irradiation at one of the interfaces between the glass panels and surrounding air.

STAR-CCM+ can dynamically monitor virtually any quantity while the solution develops. The process involves setting up a report which defines the quantity of interest and the region parts that are monitored. A monitor is then defined based on the report, which controls the update frequency and normalization characteristics. Finally, a graph plot can then be created from the monitor.

This process starts with the report definition.

1. Create a **Surface Average** report.
2. Rename the report to `Irradiation1`.
3. Select the **Irradiation1** node and set the following properties:

Property	Setting
Scalar Field Function	`Boundary Irradiation`
Parts	`Fluid:bottom-solid1 [Interface 2]`

The reason for selecting the **bottom-solid1 [Interface 2]** boundary instead of the **bottom-solid1** boundary is that after initialization, the in-place interface conditions supersede the conditions of the original wall boundary. Therefore, any values you wish to extract from boundaries on region interfaces must always be from the in-place boundary interfaces.

The setup for the report is now complete, and a monitor and plot can be made from that report.

4. Create a monitor and a plot for the report.
5. Open the plot display.

   A new plot display that is called Irradiation1 Monitor Plot appears in the *Graphics* window.

   This display is automatically updated with boundary irradiation data when the run starts.

Create three more reports in the same way as described above.

6. Create a **Surface Average** report.
7. Rename the report to `HeatFlux1`.
8. Select the **HeatFlux1** node and set the following properties:

Property	Setting
Scalar Field Function	`Boundary Heat Flux`
Parts	`Fluid:top`

9. Create a **Maximum** report.
10. Rename the report to `MaxTempFluid1`.
11. Select the **MaxTempFluid1** node and set the following properties:

Property	Setting
Scalar Field Function	`Temperature`
Parts	`Fluid`

12. Create another **Maximum** report.
13. Rename the report to `MaxTempSolid1`.

14. Select the **MaxTempSolid1** node and set the following properties:

Property	Setting
Scalar Field Function	`Temperature`
Parts	`Solid`

15. Create monitors and plots for the reports:
    a) Select the **MaxTempSolid1** and **MaxTempFluid1** nodes.
    b) Create monitors and plots for them.
    c) In the *Create Plot From Reports* dialog, select **Single Plot**.
16. Open the **Reports Plot** plot.

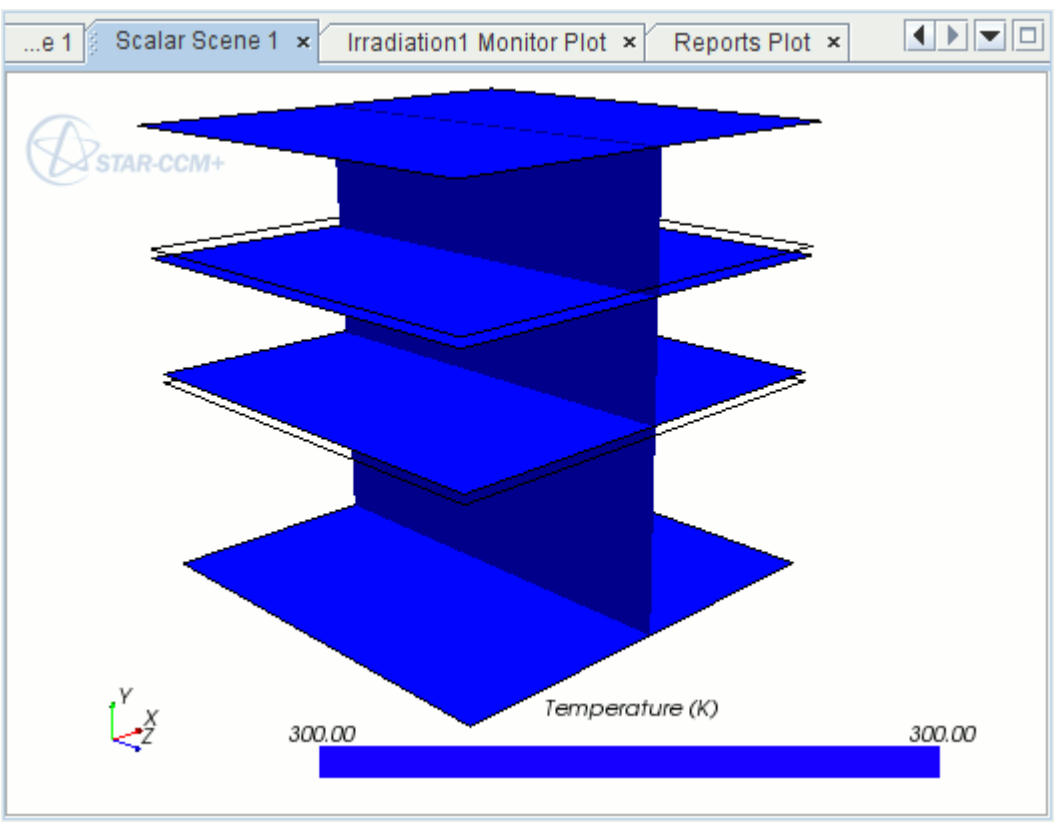
    This completes the pre-processing setup for the case.
17. Save the simulation.

Initializing and Running the Simulation

To verify that the initial conditions for the regions are correct before you run the simulation, STAR-CCM+ allows you simply to perform solver initialization and then view the results using an appropriate scene.

Use the following steps:

1. Initialize the simulation.
2. To see the initialization result, look at the scalar scene display.



Based on your radiation model selections, STAR-CCM+ generates radiation patches and calculates view factors automatically. Patches are generated when the solution is initialized, and view factors are calculated during the first iteration of the simulation run. Hence, this process is reported in the *Output* window before the iteration information appears.

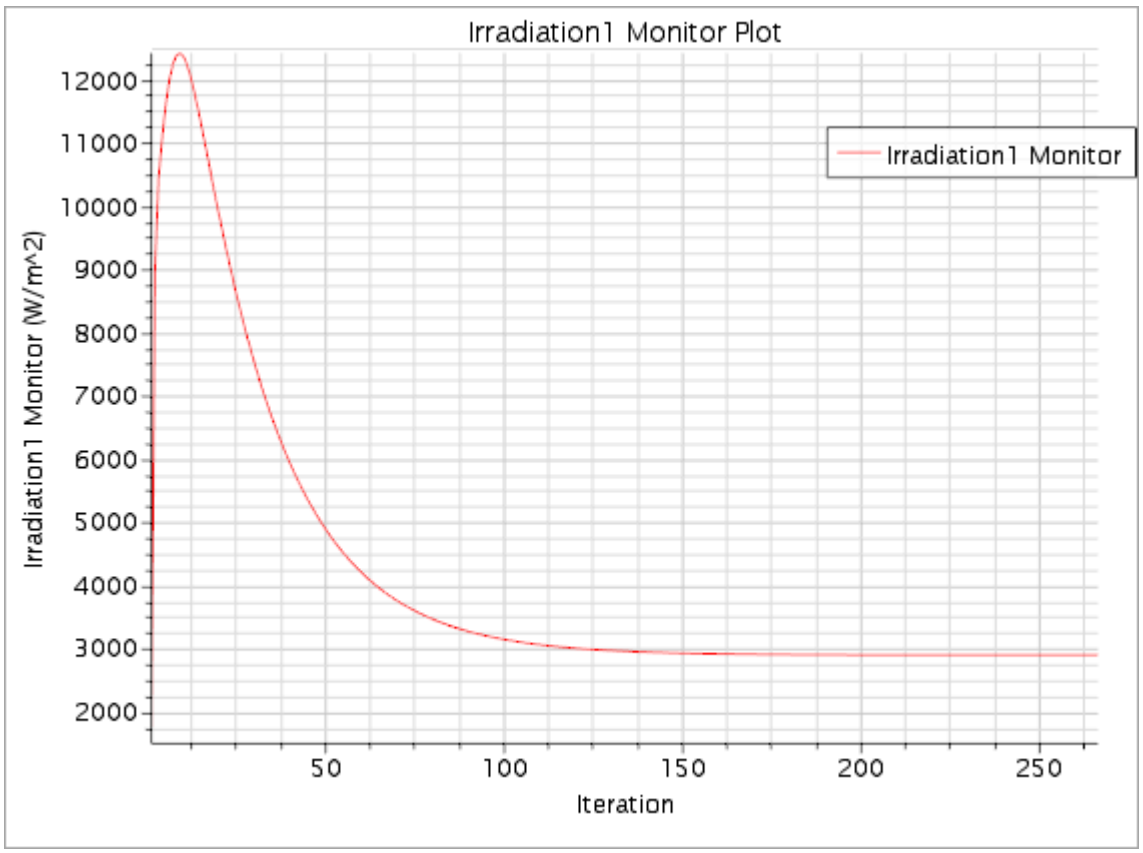
The simulation can now be run.

3. Run the simulation.
4. To watch the solution being updated while the simulation is running, look at the scalar scene display.
5. When the simulation has finished running, save it.

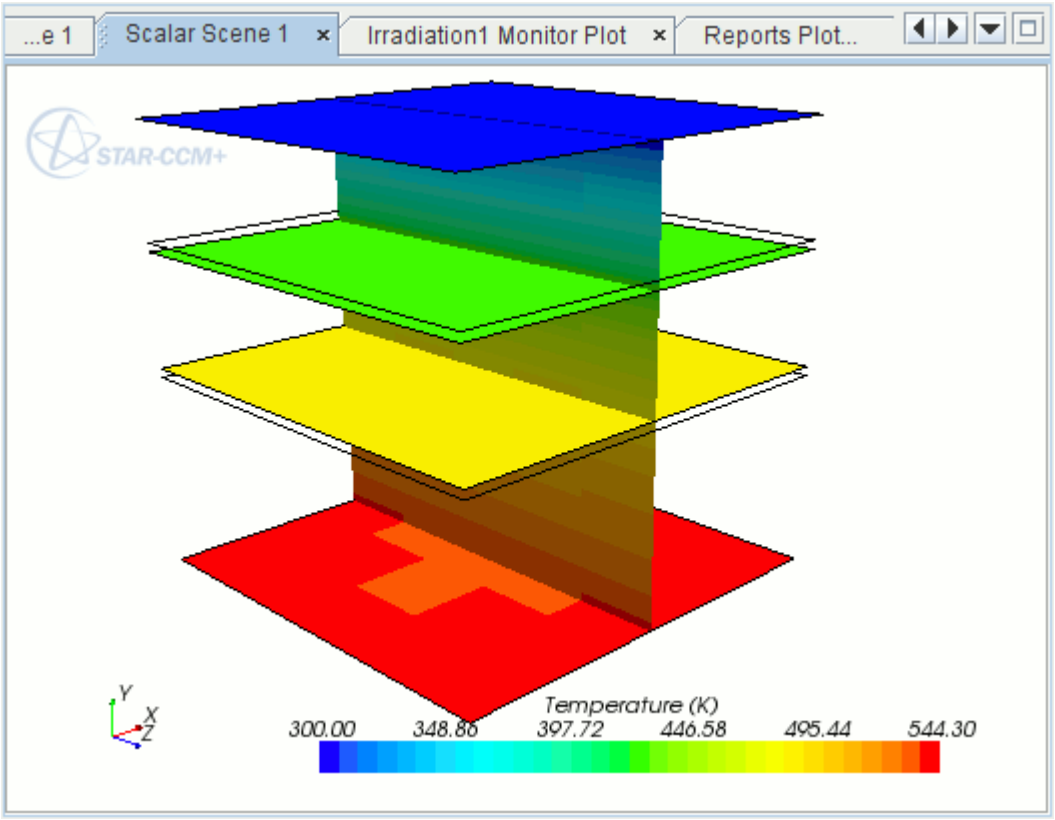
Visualizing the Results

To examine irradiation, temperature, and other results from the converged solution, look at the plots and the scene.

1. At the end of the calculation, to view the irradiation variation at the solid-fluid interface, activate the *Irradiation1 Monitor Plot* display.

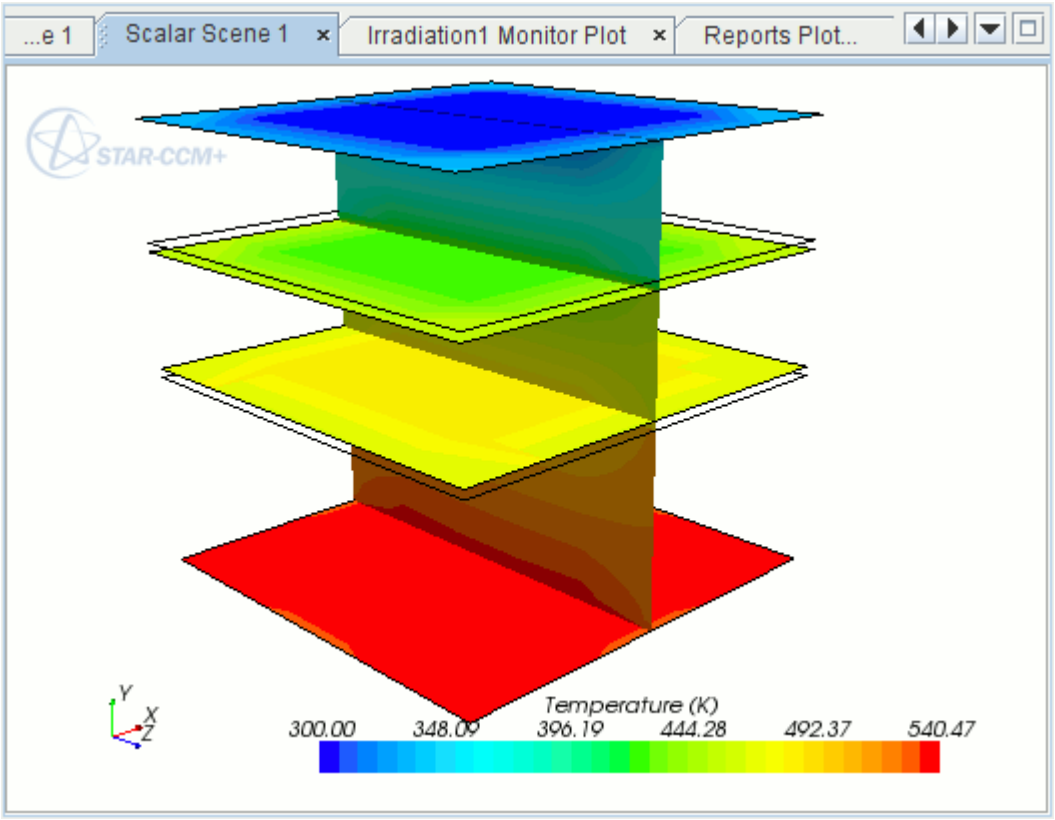


2. To see the temperature result for the converged solution, look at the *Scalar Scene 1* display.

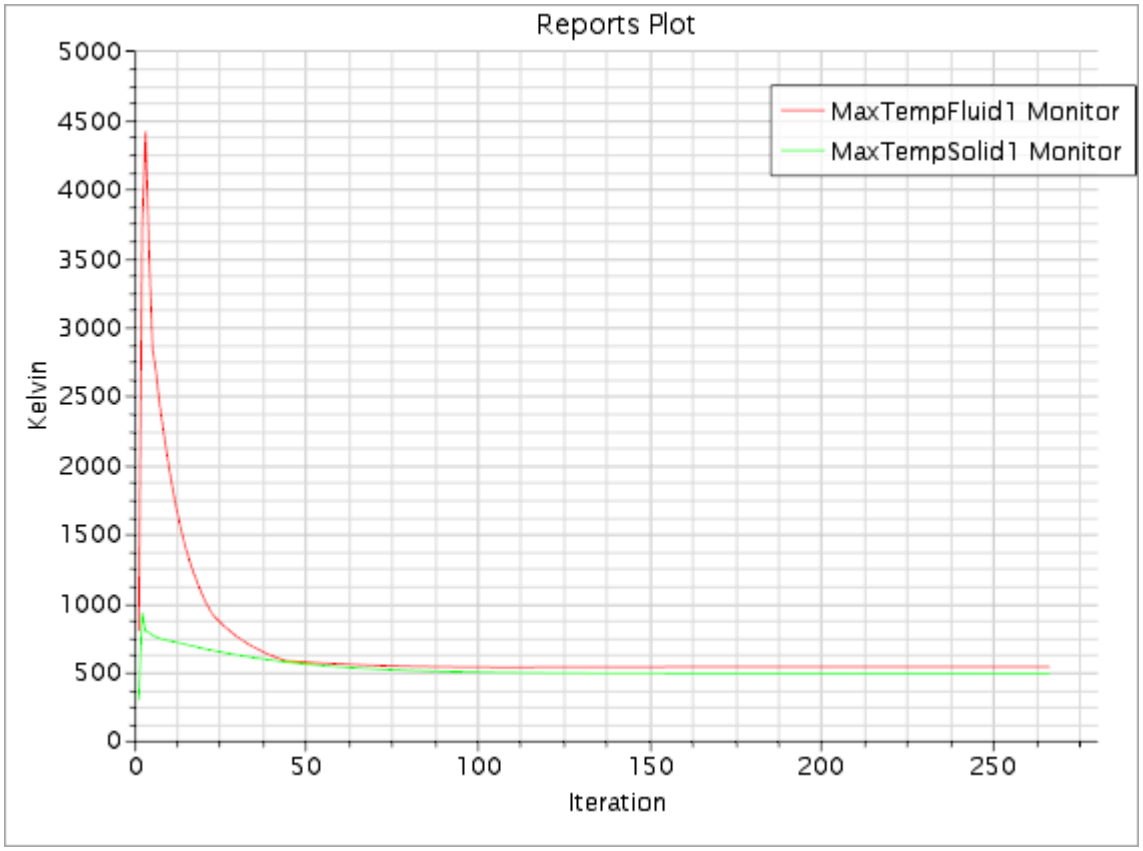


3. To make the display smoother, set *Contour Style* to `Smooth Filled`.

   The contours of the scalar display now appear smooth.



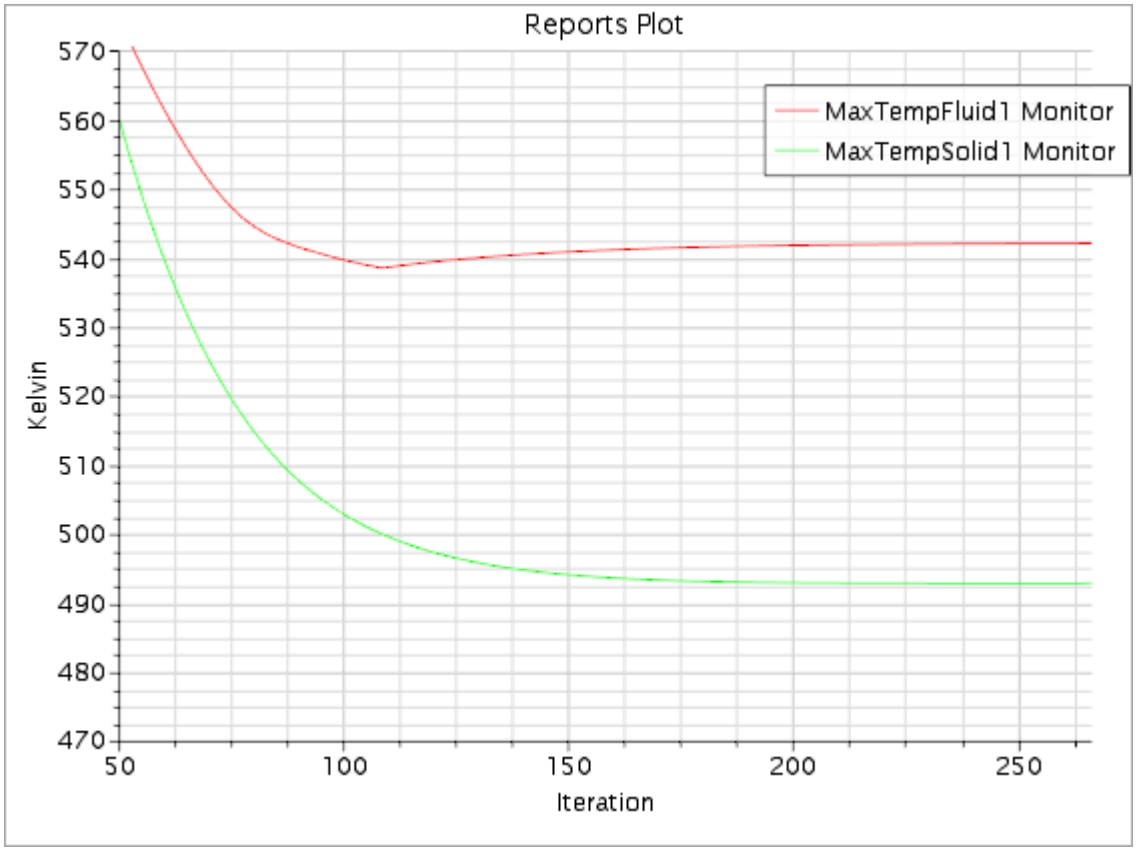
4. Look at the *Reports Plot* display to see the graph of the maximum temperature monitors for the fluid and the solid.



5. To zoom in on the relevant part of the graph, edit the **Plots** > **Reports Plot** > **Axes** node and set the following properties:

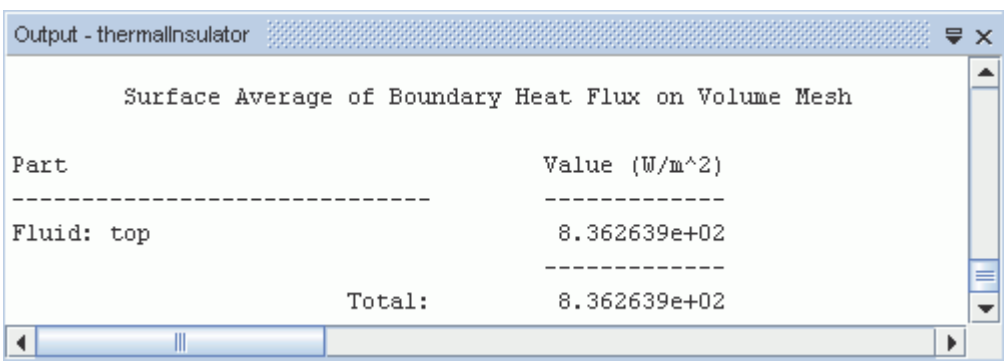
Node	Property	Setting
Bottom Axis	*Minimum*	50
Left Axis	*Minimum*	470
	Maximum	570
└ **Major Labels**	*Spacing*	10

The graph now appears as shown below.



6. To determine the heat flux that is leaving the system at the top boundary, run the **HeatFlux1** report.

The *Output* window shows that heat flux.



7. Save the simulation.

Summary

The Surface-to-Surface Radiation tutorial used two glass plates and air within a rectangular enclosure to introduce a variety of STAR-CCM+ features.

The features that were introduced include:

- Splitting boundaries.
- Defining models for multiple continua.
- Defining material properties that are required for the selected models.
- Defining boundary conditions.
- Setting solver parameters.
- Setting up a monitoring report and plot.
- Initializing and running the solver until the residual magnitudes are sufficiently small.
- Analyzing results using the visualization, reports, and monitor plot facilities.

Multiband Surface-to-Surface Radiation: Solar Collector

This tutorial uses the simulation created for the Surface-to-Surface Radiation: Thermal Insulator.

This tutorial uses the geometry and initial problem setup from the Surface-to-Surface Radiation: Thermal Insulator. It solves the same problem, except that solar radiation effects are also included using a solar load and the multiband radiation model. The multiband radiation model is used to simulate the behavior of thin glass, which transmits most solar radiation while absorbing most infrared radiation. For the purposes of this tutorial, the glass is given a high transmissivity for wavelengths in the range 0.1 to 3.0 mm , and lower values in the range 3.0 mm to 1.0 m.

While the multiband radiation model is used here to represent radiative properties that are dependent on wavelength, modeling solar radiation does not require the multiband model. If property variations with wavelength are not being considered, solar loads may also be applied with the gray radiation spectrum model.

The influence of this additional heat transfer mechanism on the overall heat transfer to the collector is considered by comparing the two sets of results.

The analysis covers the following heat transfer modes:

- Conduction
- Multiband thermal radiation
- Solar load

Note: Radiative properties given in this tutorial are for demonstration only.

Contents:

Prerequisites
Loading an Existing Simulation
Modifying the Model Selection
Setting Solar Loads Model Properties
Setting Multiband Thermal Radiation Properties
Setting Boundary Conditions and Values
Setting Solver Parameters
Reporting and Monitoring with Plots
Running the Simulation
Visualizing the Results
Summary

Prerequisites

The instructions in the Multiband Surface-to-Surface Radiation tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Loading an Existing Simulation

For this tutorial, you are provided with a simulation file that contains pre-defined objects.

Load the initial simulation:

1. Launch STAR-CCM+.

   Note: To start this tutorial, you can open one of two versions of the simulation file: the one that you created locally (if applicable), or a previously solved simulation file that is available online. The following instructions apply to a locally created simulation file.

2. Select **File** > **Load Simulation...**
3. In the *Load Simulation* dialog, click **Browse...**
4. In the *Open* dialog, locate the `thermalInsulator.sim` file that was created in the Thermal Insulator Tutorial.
5. Select `thermalInsulator.sim` then click **Open**.
6. In the *Load Simulation* dialog, click **OK**.
7. Save the simulation as `solarCollector.sim`.

Modifying the Model Selection

Set up the models for the solar collector case to include a solar load and multiband radiation.

Most of the models that are required in this case have already been selected (see Setting Up the Models in the Thermal Insulator Tutorial) and so do not need to be specified. It is, however, necessary to apply a solar load and set the properties for multiband radiation.

For the physics continuum, **Continua** > **Air**, change the choice of radiation models:

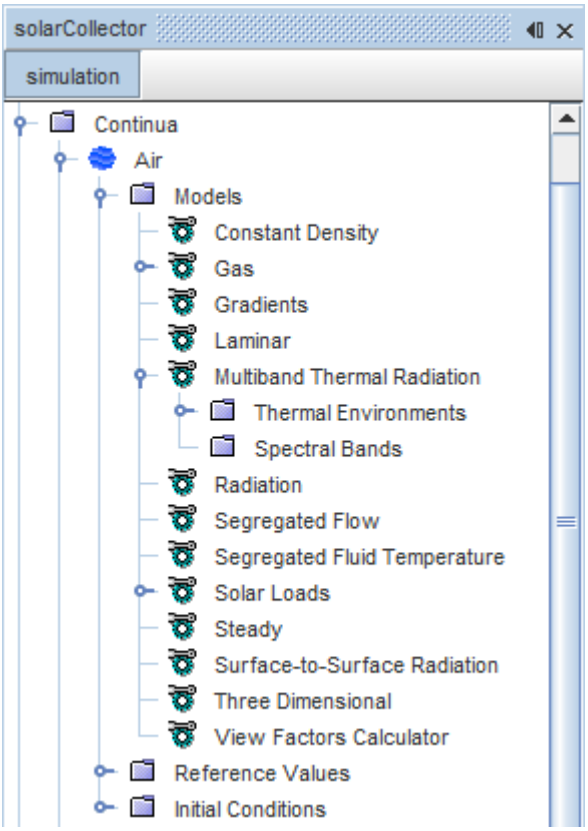
1. In the *Enabled Models* group box of the model selection dialog, deselect the **Gray Thermal Radiation** model.

   The *Radiation Spectrum* box appears on the left.

2. Select the following models in order:

Group Box	Model
Radiation Spectrum	**Multiband Thermal Radiation**
Solar Radiation	**Solar Loads**

3. Click **Close**
4. To review the models, open the **Air** > **Models** node.



Since solar radiation heat transfer is to be modeled in the glass solid as well as in the air, the Multiband Thermal Radiation and Solar Loads models must be activated in the solid continuum.

For the physics continuum, **Continua** > **Glass**, change the choice of radiation models:

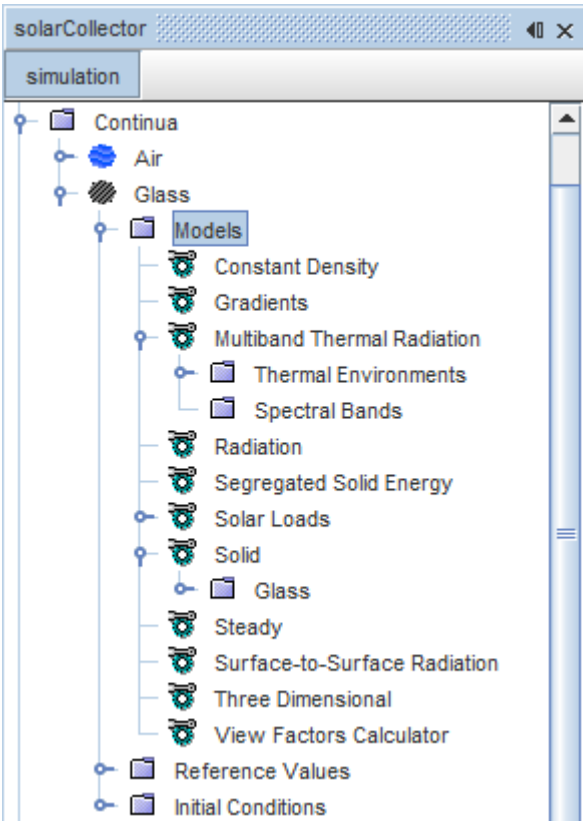
5. In the *Enabled Models* group box of the model selection dialog, deselect the **Gray Thermal Radiation** model.

   The *Radiation Spectrum* box appears on the left.

6. Select the following models in order:

Group Box	Model
Radiation Spectrum	**Multiband Thermal Radiation**
Solar Radiation	**Solar Loads**

7. Click **Close**
8. To review the models, open the **Glass** > **Models** node.



9. Save the simulation.

Setting Solar Loads Model Properties

The Solar Loads Model requires definition of the sun direction vector in a specified coordinate system, the direct solar flux, and the diffuse solar flux.

The Sun Direction Vector is defined as having a horizontal component of 90 degrees and a vertical component of 0 degrees.

1. Select the **Continua** > **Air** > **Models** > **Solar Loads** > **Manual Specification** node and set *Azimuth* to `-90` with units of degrees.

The solar load model properties are the same for the glass continuum and may be copied and pasted from the air continuum node.

2. Copy from the **Continua** > **Air** > **Models** > **Solar Loads** node and paste onto the **Continua** > **Glass** > **Models** > **Solar Loads** node.

   If you prefer, examine the properties of the **Solar Loads** node to verify that the properties have copied correctly.
3. Save the simulation.

Setting Multiband Thermal Radiation Properties

Two spectral bands are created for this tutorial.

The first spectral band covers the range 0.1 to 3.0 μm, and the second covers the range 3.0 μm to 1.0 m.

1. Expand the **Continua** > **Air** > **Models** > **Multiband Thermal Radiation** node.
2. Right-click the **Spectral Bands** node and create a spectral band.

   A node appears within the **Spectral Bands** node: **Thermal 1**.
3. Create a second spectral band.
4. Edit the **Continua** > **Air** > **Models** > **Multiband Thermal Radiation** > **Spectral Bands** node and set the following properties:

Node	Property	Setting
Thermal 1	*Upper Wavelength*	`3.0E-6 m`
	Lower Wavelength	`1.0E-7 m`
Thermal 2	*Upper Wavelength*	`1.0 m`
	Lower Wavelength	`3.0E-6 m`

Corresponding spectral bands must also be created for the **Glass** continuum. While it is possible to do this by repeating the same procedure, it can be more easily accomplished by copying and pasting.

5. Copy from the **Continua** > **Air** > **Models** > **Multiband Thermal Radiation** > **Spectral Bands** node and paste onto the **Continua** > **Glass** > **Models** > **Multiband Thermal Radiation** > **Spectral Bands** node.

   This operation adds two nodes, **Thermal 1** and **Thermal 2**, to the **Spectral Bands** node within the **Glass** continuum.
6. Save the simulation.

Setting Boundary Conditions and Values

Applying the Multiband Thermal Radiation model means that the optical properties at boundaries and interfaces must be revised.

For each boundary, it is now necessary to set the emissivity, reflectivity and transmissivity for each of the spectral bands that were created for the Multiband Thermal Radiation model. Property values that were set for the Thermal Insulator Tutorial are used for the **Thermal 2** spectral band, and new property values are assigned for the **Thermal 1** spectral band.

To make sure that almost all of the energy in the **Thermal 1** spectral band passes into the box, the top of the collector is assigned a reflectivity of 0 and a transmissivity of 0.98. The base of the collector is assigned a reflectivity of 0 and a transmissivity of 0 for both spectral bands.

Note: The non-radiative thermal conditions for the boundaries remain as before.

1. Select the **Regions** > **Fluid** > **Boundaries** > **collector** > **Physics Values** > **Surface Emissivity** > **Constant** node.

As there are multiple spectral bands present, the **Constant** node takes the form of an array profile value. The first value in the array corresponds to the **Thermal 1** spectral band and the second value corresponds to the **Thermal 2** spectral band. The array is shown in short notation, and can be fully visualized by opening the custom editor.

2. Edit the *Value* property of the **Constant** node.
3. In the *Constant - Value* dialog, set both *Thermal 1* and *Thermal 2* (the constant emissivity value) to `1.0`.
4. Click **OK**.
5. Edit the **collector** > **Physics Values** node.
6. Set the following properties:

Node	Property	Setting
Surface Reflectivity	*Method*	`Auto Calculate`
Surface Transmissivity	*Method*	**Constant**
	Value	`[0.0, 0.0]`

Set the optical properties for the **side-1**, **side-2**, and **side-3** wall boundaries:

7. Edit the **Regions** > **Fluid** > **Boundaries** > **side-1** > **Physics Values** node.
8. Set the following properties:

Node	Property	Setting
Surface Emissivity	*Method*	**Constant**
	Value	`[0.95, 0.95]`
Surface Transmissivity	*Method*	**Constant**
	Value	`[0.0, 0.0]`

The properties of the **side-1** node can be copied and pasted to the **side-2** and **side-3** nodes.

9. Copy from the **side-1** node and paste onto the **side-2** and **side-3** nodes.

Set the optical properties for the **top** boundary node.

10. Edit the **Regions** > **Fluid** > **Boundaries** > **top** > **Physics Values** node.
11. Set the following properties:

Node	Property	Setting
Surface Emissivity	*Method*	
	Value	`[0.02, 0.8]`
Surface Transmissivity	*Method*	
	Value	`[0.98, 0.15]`

As in the Thermal Insulator tutorial, it is also necessary to set the optical properties on either side of the solid-fluid interfaces. Set the fluid side properties first.

12. Edit the **Regions** > **Fluid** > **Boundaries** node.
13. Set the following properties:

Node	Property	Setting
bottom-solid1 [In-place 2] > **Physics Values**		

Node	Property	Setting
└ **Surface Emissivity**	*Method*	**Constant**
	Value	`[0.1, 0.775]`
bottom-solid2 [In-place 4] > **Physics Values**		
└ **Surface Emissivity**	*Method*	**Constant**
	Value	`[0.2, 0.725]`
top-solid1 [In-place 1] > **Physics Values**		
└ **Surface Emissivity**	*Method*	**Constant**
	Value	`[0.15, 0.8]`
top-solid2 [In-place 3] > **Physics Values**		
└ **Surface Emissivity**	*Method*	**Constant**
	Value	`[0.175, 0.75]`

Set the same optical properties for the corresponding solid-side of each interface. An efficient way of doing this is to copy and paste the relevant nodes.

14. For each of the boundary names listed below, copy from the node in **Regions** > **Fluid** > **Boundaries**, and paste onto the boundary node with the matching name in **Regions** > **Solid** > **Boundaries**:
    - **bottom-solid1 [In-place 2]**
    - **bottom-solid2 [In-place 4]**
    - **top-solid1 [In-place 1]**
    - **top-solid2 [In-place 3]**

The final boundaries to set are the side wall boundaries of the **Solid** region.

15. Edit the **Regions** > **Solid** > **Boundaries** > **side-solid1** > **Physics Values** node.
16. Set the following properties:

Node	Property	Setting
Surface Emissivity	*Method*	**Constant**
	Value	`[0.0, 0.0]`
Surface Transmissivity	*Method*	**Constant**
	Value	`[0.0, 0.0]`

    This setting results in a **Surface Reflectivity** of `1.0` for both spectral bands at the edges of the solid. These same settings are copied to the other solid boundary.
17. Copy from the **side-solid1** node and paste onto the **side-solid2** node.

Set the surface transmissivity at the interfaces.

18. Expand the **Interfaces** node.

19. In each interface node, select the **Physics Values** > **Surface Transmissivity** node and set *Value* as follows for each interface:

Interface Node	Surface Transmissivity Value
Interface 1	`[0.8, 0.15]`
Interface 2	`[0.8, 0.15]`
Interface 3	`[0.75, 0.175]`
Interface 4	`[0.75, 0.175]`

20. Save the simulation.

Setting Solver Parameters

To ensure solution convergence, reduce the under-relaxation factor for fluid segregated energy.

Use the following steps:

1. Select the **Solvers** > **Segregated Energy** node and change the *Fluid Under-Relaxation Factor* to `0.5`.
2. Save the simulation.

Reporting and Monitoring with Plots

The monitor plots created in the Thermal Insulator Tutorial are used to monitor the simulation progress.

STAR-CCM+ can dynamically monitor virtually any quantity while the solution develops. The process involves setting up a report which defines the quantity of interest and the region parts that are monitored. A monitor is then defined based on the report, which controls the update frequency and normalization characteristics. A graph plot can also be created from the monitor.

Several reports and monitors have already been set up for the solution in the Reporting, Monitoring, and Plotting section of the Thermal Insulator Tutorial.

1. Open the **Irradiation1 Monitor Plot** and **Reports Plot** plots.

To monitor the irradiation of the collector for the first spectral band, create a report.

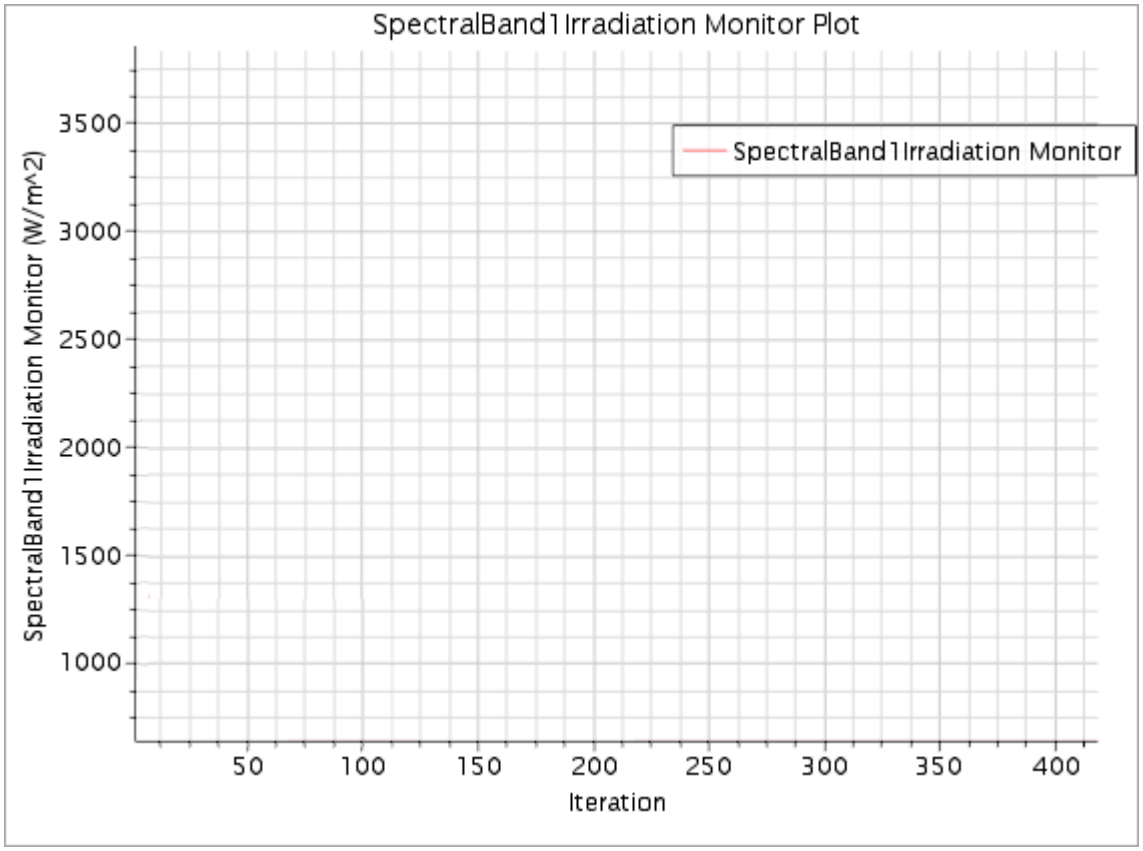
2. Create a **Surface Average** report.
3. Rename the report to `SpectralBand1Irradiation`.
4. Select the **SpectralBand1Irradiation** node and set the following properties:

Property	Setting
Scalar Field Function	`Boundary Irradiation of Thermal 1`
Parts	`collector`

The setup for the report is now complete and a monitor and plot can be made from that report.

5. Create a monitor and a plot for the report.
6. Open the plot display.

A new plot display that is called SpectralBand1Irradiation Monitor Plot appears in the *Graphics* window.



This display is automatically updated with boundary solar irradiation data when the run starts.

This completes the pre-processing setup for the case.

7. Save the simulation.

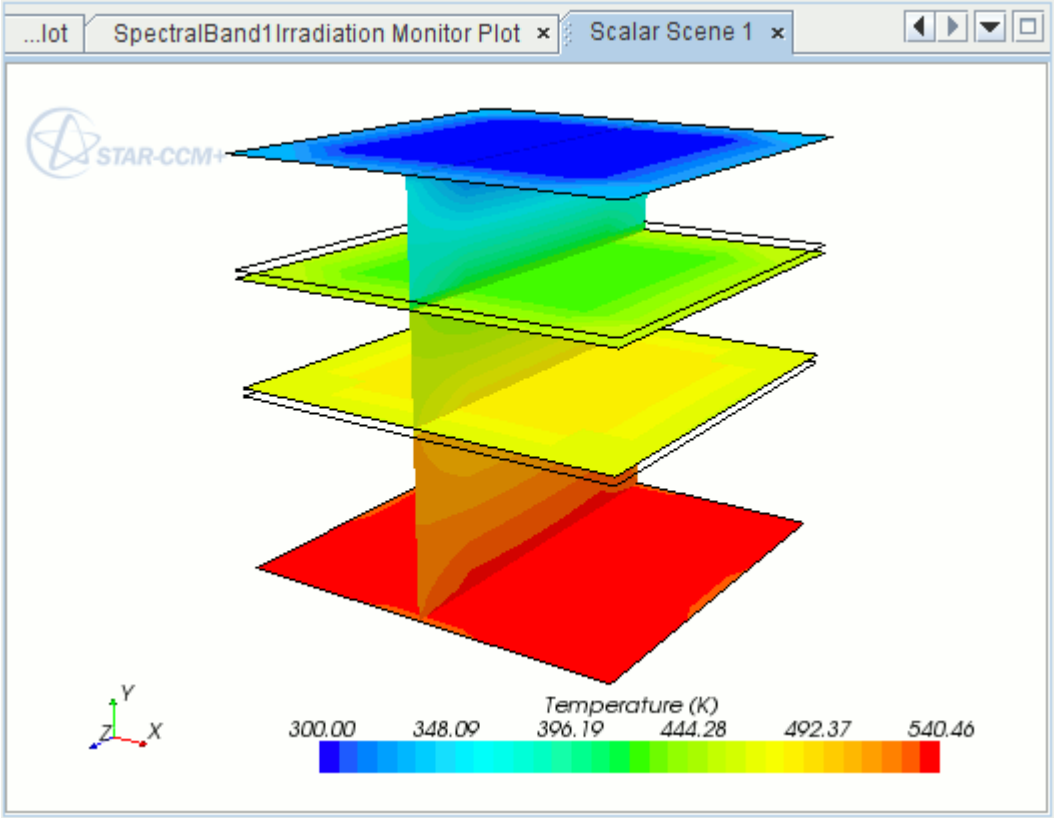
Running the Simulation

The solution from the Thermal Insulator tutorial can be inspected by displaying the initial temperature profile.

Review the previous solution results.

1. Open *Scalar Scene 1*.

   The temperature profile from the previous solution is displayed in the *Graphics* window.



Before running this simulation, clear the previous solution:

2. Select **Solution** > **Clear Solution**.
3. In the *Clear Solution* dialog, click **OK**.

   The simulation can now be run.

Generate a new solution.

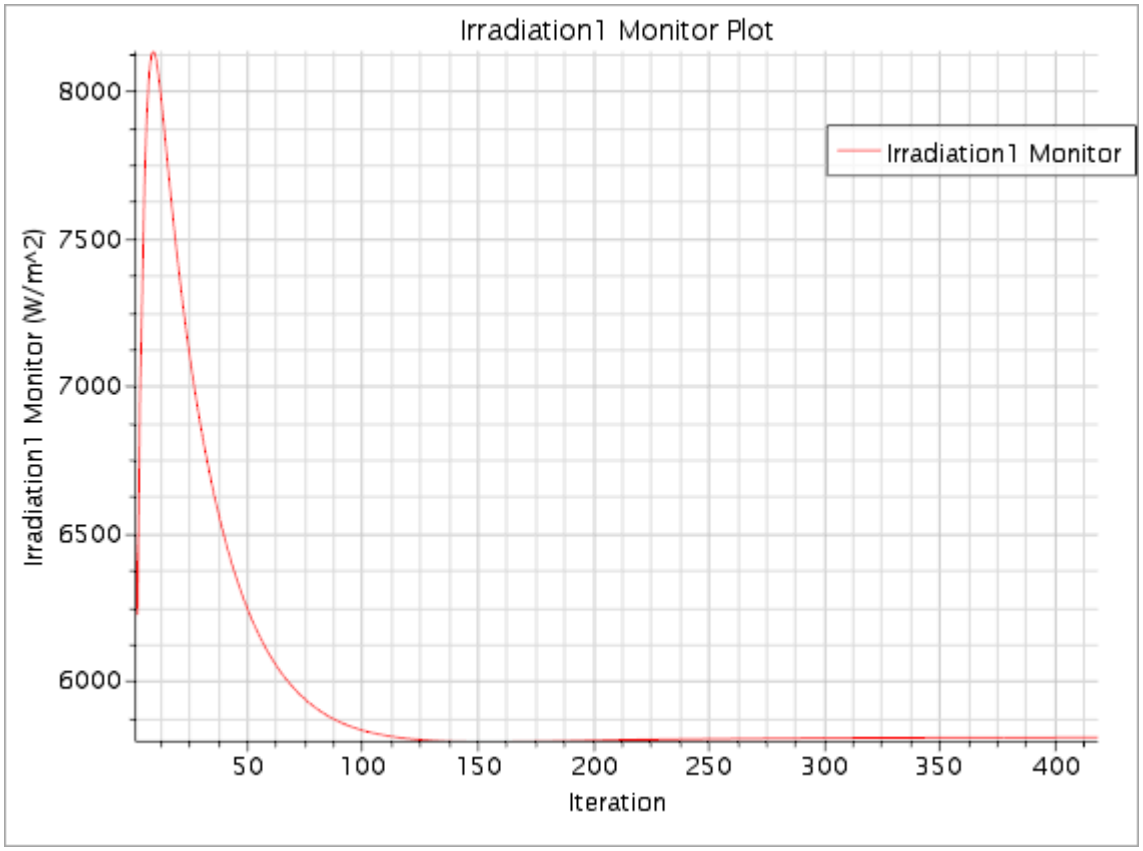
4. Run the simulation.

   View factors are recalculated during the first iteration of the simulation run. Hence, see this process reported in the Output window before the iteration information appears.
5. To watch the solution being updated while the simulation is running, look at the scalar scene, vector scene, and plot displays.
6. When the simulation has finished running, save it.

Visualizing the Results

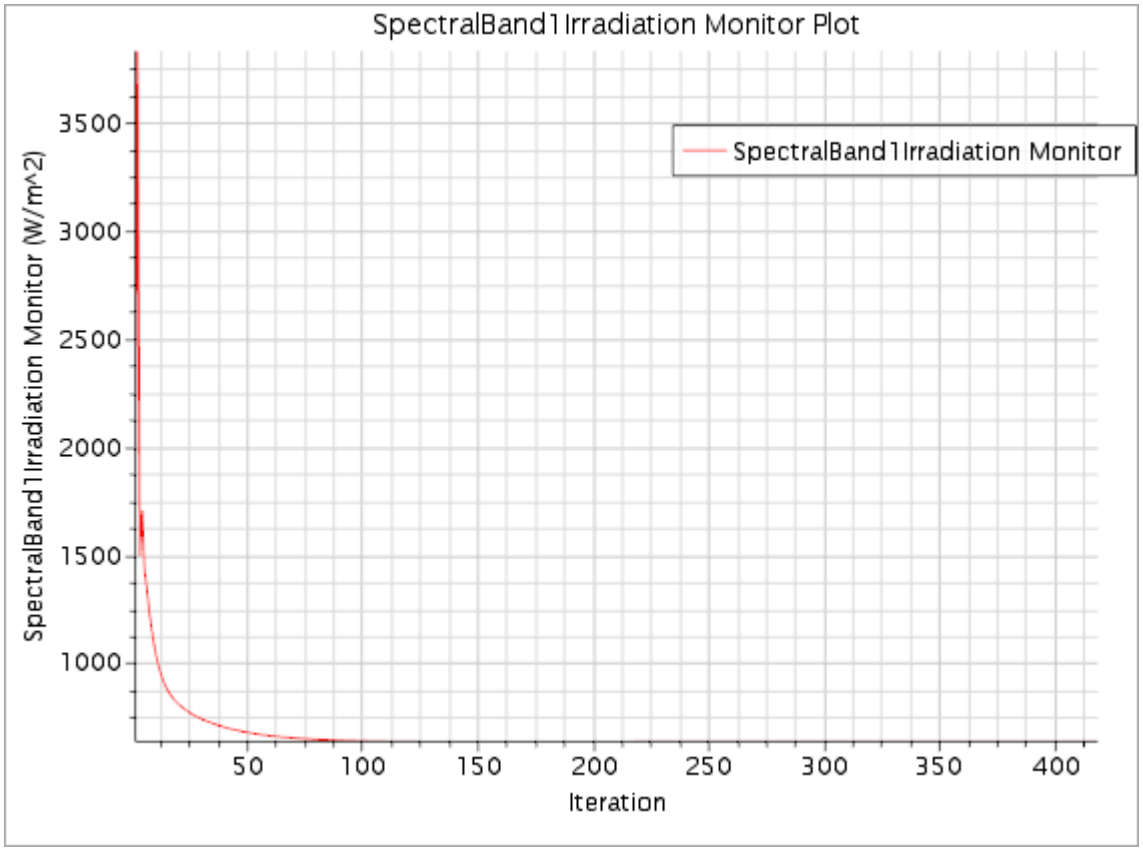
Examine the plots and scenes.

1. At the end of the calculation, to view the irradiation variation at the solid-fluid interface, activate the *Irradiation1 Monitor Plot* display.

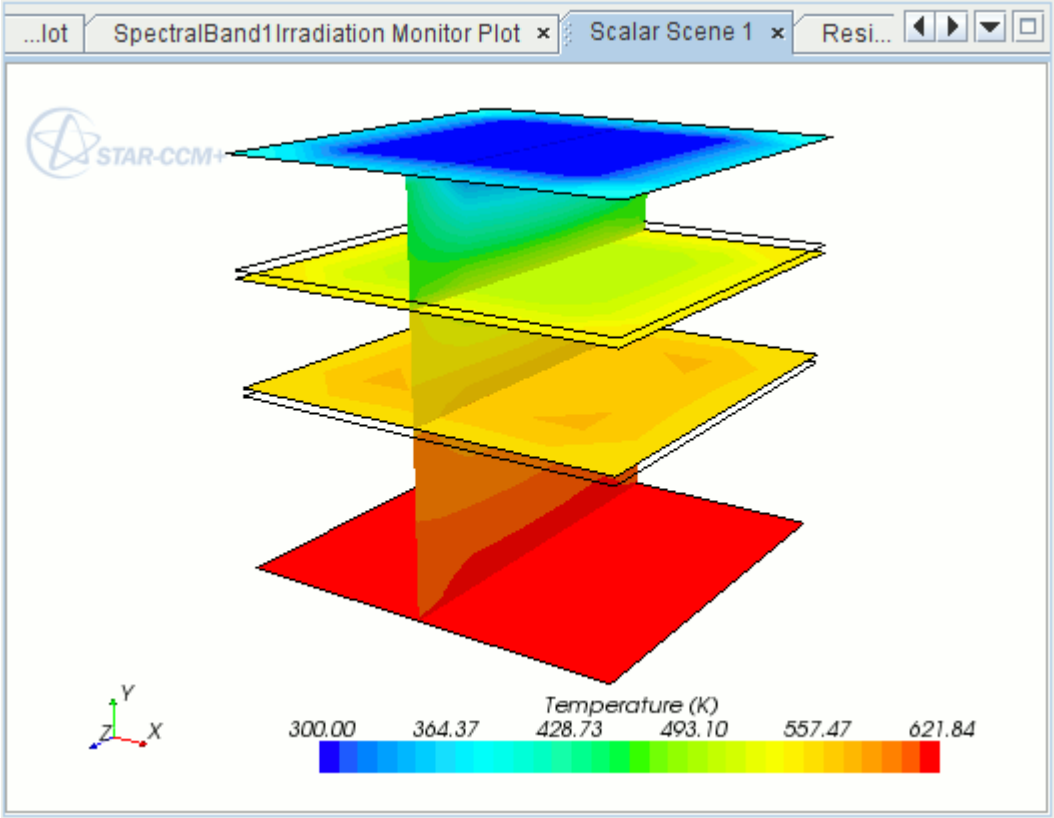


The boundary irradiation is higher than its Thermal Insulator Tutorial counterpart due to the additional solar irradiation.

2. To view the solar irradiation at the collector, look at the SpectralBand1Irradiation Monitor Plot display.



3. To see the temperature result for the converged solution, look at the *Scalar Scene 1* display.

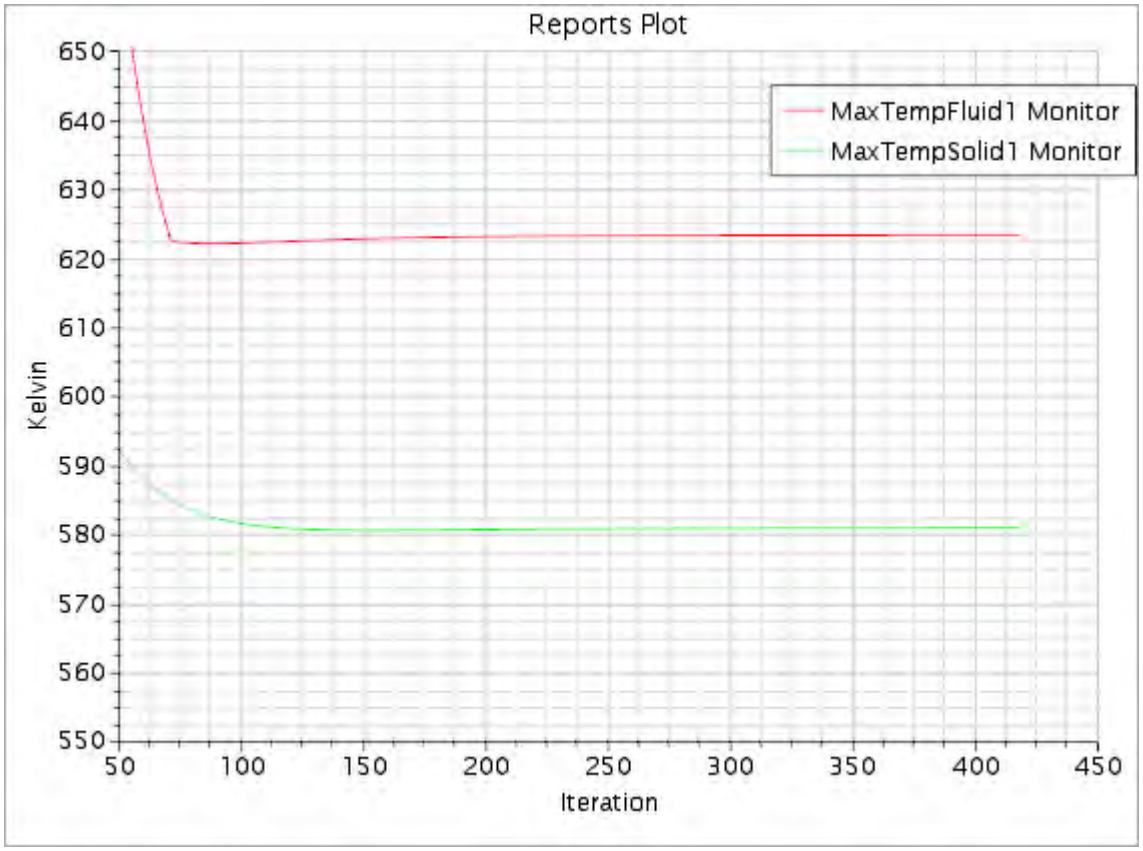


The maximum temperature has increased by approximately 80 K compared with the previous solution.

4. Look at the *Reports Plot* display to see the graph of the maximum temperature monitors for the fluid and the solid.
5. To zoom in on the relevant part of the graph, edit the **Plots** > **Reports Plot** > **Axes** node and set the following properties:

Node	Property	Setting
Bottom Axis	*Minimum*	50
	Maximum	450
└ **Major Labels**	*Spacing*	50
Left Axis	*Minimum*	550
	Maximum	650
└ **Major Labels**	*Spacing*	10

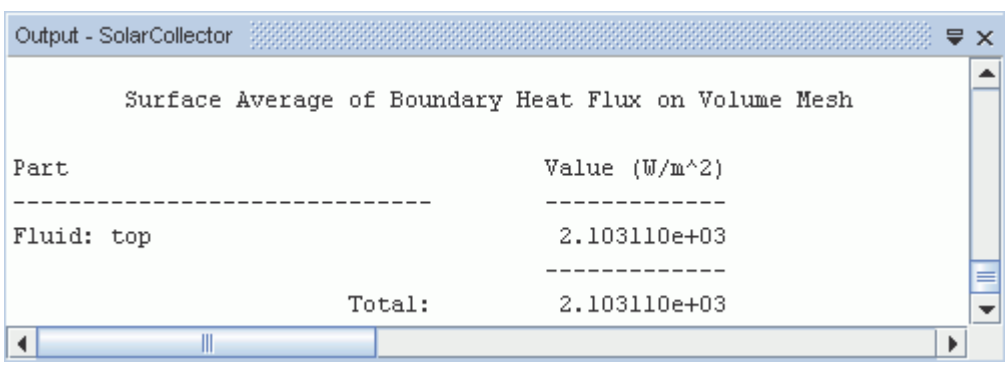
The graph now appears as shown below.



Compare with the Reports Plot from the previous solution. The maximum temperature of the fluid has increased from ~540 to ~620 as a result of the solar radiation effect. The maximum temperature of the solid has increased from ~490 to ~580.

The **Heat Flux 1** report may be run .

6. To determine the heat flux leaving the system at the top boundary, run the **HeatFlux1** report.
7. The *Output* window shows the heat flux that is leaving the system.



8. Save the simulation.

Summary

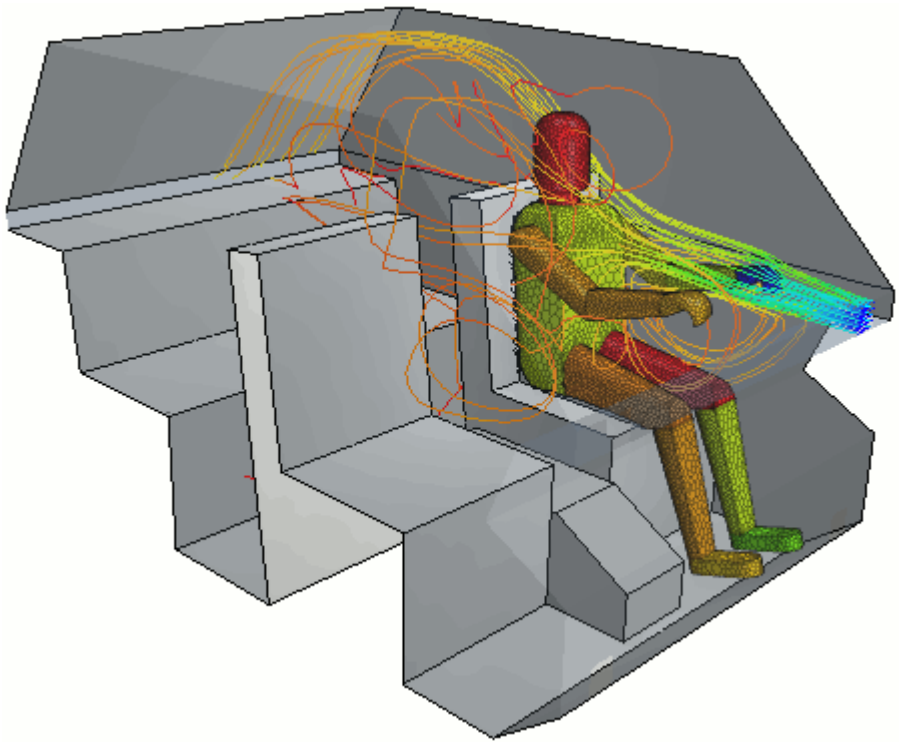
The Multiband Surface-to-Surface Radiation tutorial used the geometry and initial problem setup from the Thermal Insulator Tutorial to introduce a variety of STAR-CCM+ features.

The features that were introduced include:

- Starting the code and loading an existing simulation.
- Introducing solar loads and multiband thermal radiation into the problem conditions.
- Defining boundary conditions for multiband thermal radiation.
- Setting up a monitoring report and plot.
- Running the solver until the residual magnitude is sufficiently small.
- Analyzing results using the visualization, reports and monitor plot facilities.

Thermal Comfort Wizard: Single Occupant in a Cabin

Use the Thermal Comfort Wizard to set up and run a thermal comfort analysis case for a single occupant in the cabin of a car.



The Thermal Comfort Management (TCM) model accounts for the properties of the human body. It allows you to observe the effects of flow scenarios on skin surface temperatures. The TCM model runs the TIM program (a German acronym for “Thermophysiological Occupant Model”) which calculates the thermal state of the occupant, based on the surrounding conditions.

Contents:

Prerequisites
Importing the Surface Mesh and Naming the Simulation
Visualizing the Imported Geometry and Surface Mesh
Assigning Parts to Regions
Generating the Volume Mesh
Setting Up the Physics Continuum
Preparing Boundaries for the Thermal Comfort Wizard
Activating the Thermal Comfort Wizard
Using the Thermal Comfort Wizard
Setting Stopping Criteria
Creating a Scalar Scene

Running the Solver
Visualizing the Results with Streamlines
Summary

Prerequisites

The instructions in the Thermal Comfort Wizard tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction
Using visualization tools, scenes, and plots	Introduction

Importing the Surface Mesh and Naming the Simulation

To set up the STAR-CCM+ simulation, launch a simulation and import the supplied surface mesh.

Note: It is recommended that STAR-CCM+ is launched from an empty working directory as a number of files are generated during the TCM run.

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Surface Mesh**.
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/heatTransferAndRadiation/data`.
5. Select `singleOccTCM.dbs` then click **Open**.
6. In the *Import Surface Options* dialog, select **Create New Part** from the *Import Mode* group box. To accept the default Import Surface Options, click **OK**.

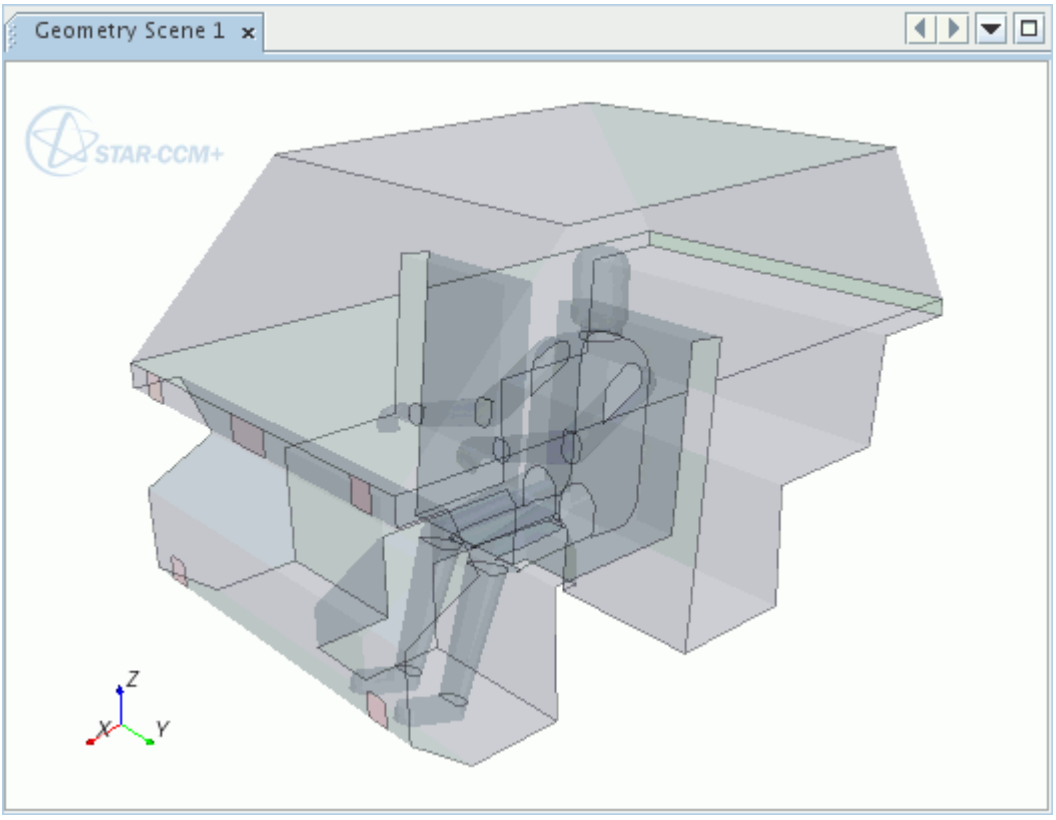
   STAR-CCM+ provides feedback on the import process in the *Output* window.
7. Save the simulation as `personCabin.sim`.

Visualizing the Imported Geometry and Surface Mesh

Use the visualization features to view selected parts of the cabin and examine the mesh.

The tutorial geometry consists of a single occupant in the cabin of a vehicle. There are five inlet vents at the front of the cabin, and an outlet vent at the rear.

1. Look at the *Geometry Scene 1* display.
2. Click (**Make Scene Transparent**).
3. Use the mouse buttons to adjust the view so that it appears as follows:

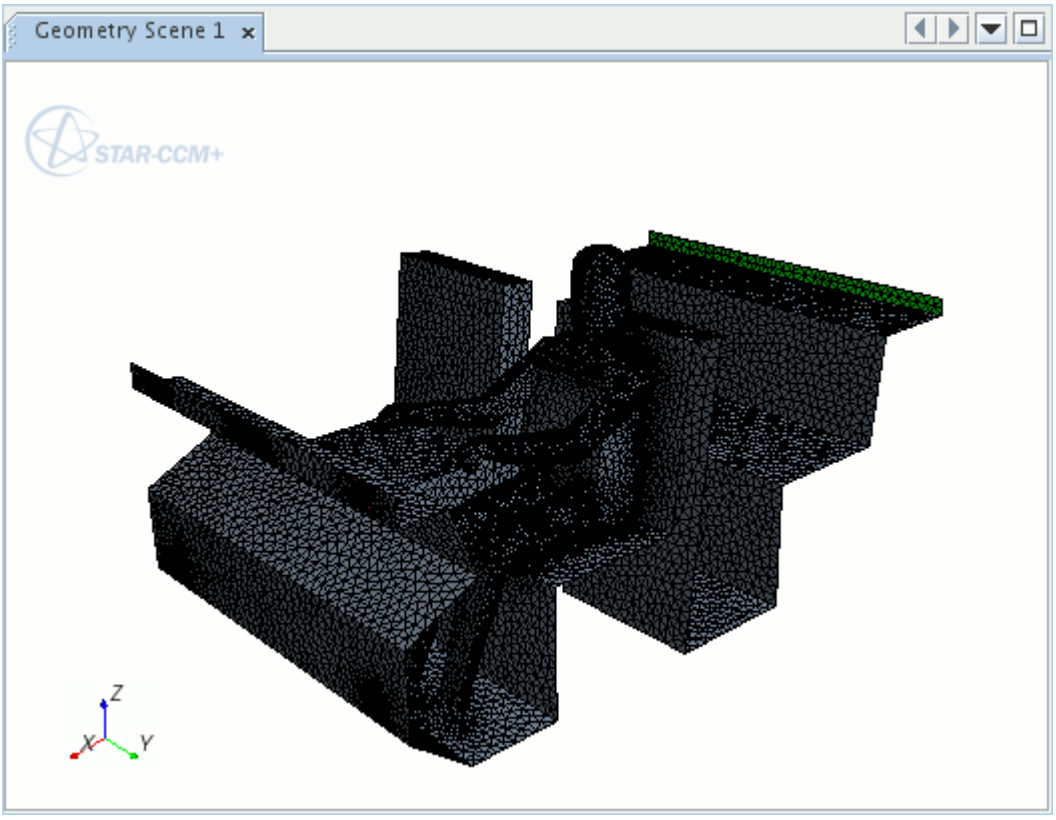


4. Click (**Save-Restore-Select views**) and select **Store Current View**.

The part surfaces defining each body segment have been named according to the requirements of the Thermal Comfort Wizard.

Examine the imported surface mesh.

5. Click (**Make Scene Transparent**).
6. Edit the **Scenes** > **Geometry Scene 1** > **Displayers** > **Geometry 1** > **Parts** node.
7. In the *Parts* dialog, expand the **Surfaces** node and deselect the **Doors**, **Roof**, and **Windows** part surfaces.
8. Click **OK**.
9. Right-click the **Outline 1** node and select **Toggle Visibility**.
10. Click (**Show All Meshes**).



Assigning Parts to Regions

Before creating the volume mesh, you must transfer the geometry part to the regions level.

Create a region and associated boundaries from the geometry part and its surfaces:

1. Right-click **Geometry** > **Parts** > **singleOccTCM** and select **Assign Parts to Regions...**.
2. In the *Assign Parts to Regions* dialog:
   a) Select **Create a Region for Each Part**.
   b) Select **Create a Boundary for Each Part Surface**.
   c) Click **Apply**, then **Close**.

   In the object tree, all of the surfaces appear as individual boundaries within the region.
3. Save the simulation.

Generating the Volume Mesh

Create a volume mesh for the car cabin.

The starting surface can be used to generate a volume mesh without the need of a surface mesher.

To select the meshers and specify some of the global mesh parameters:

Create an automated mesh operation:

1. Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
2. In the *Create Automated Mesh Operation* dialog:
   a) Select **singleOccTCM** from the Parts list.
   b) Select the following meshers, in order:

Group	Mesher
Core Volume Meshers	**Polyhedral Mesher**
Optional Boundary Layer Meshers	**Prism Layer Mesher**

   The **Automated Mesh** node appears in the **Operations** node.

Reference sizes must now be set at the global level to make sure cell sizes are appropriate for the simulation.

3. Edit the **Geometry** > **Operations** > **Automated Mesh** > **Default Controls** node and set the following properties:

Node	Property	Setting
Base Size	*Value*	`40 mm`
Prism Layer Total Thickness	*Percentage of Base*	`10`

4. Click **Close**.

The mesh has now been defined.

5. To start the mesh generation process, right-click the **Geometry** > **Operations** node and select **Execute All**.

   The output from the meshers is shown in the *Output* window. When the meshing process is complete, the message `Volume Meshing Pipeline Completed` is displayed in the *Output* window.
6. To view the mesh in the scene, right-click the **Scenes** > **Geometry Scene 1** node and select **Apply Representation** > **Volume Mesh**.
7. Save the simulation.

Setting Up the Physics Continuum

Models define the primary variables of the simulation, including pressure, temperature, velocity, and what mathematical formulation is used to generate the solution. In this example, the flow is turbulent with constant density.

To model the air flow through the cabin:

1. For the physics continuum, **Continua** > **Physics 1**, select the following models in order:

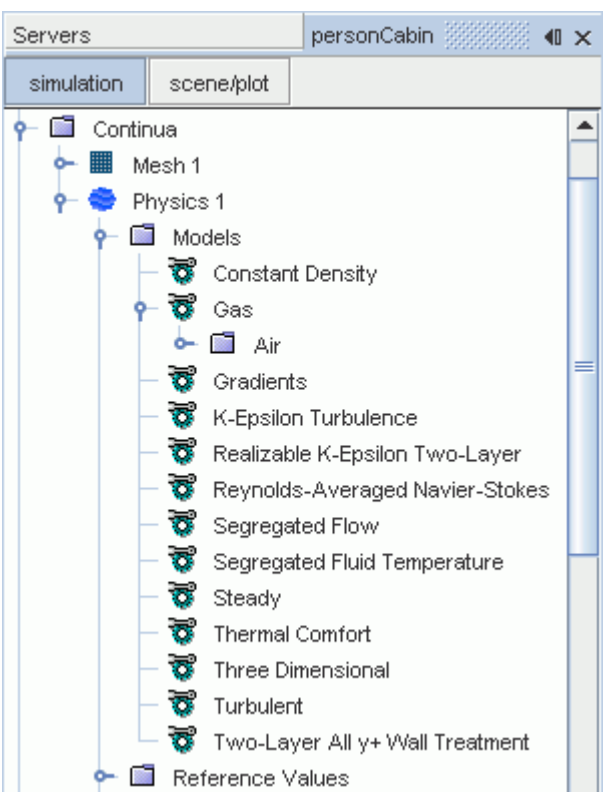
Group Box	Model
Enabled Models	**Three Dimensional** (Selected automatically)
Time	**Steady**
Material	**Gas**
Flow	**Segregated Flow**

Group Box	Model
	Gradients (Selected automatically)
Equation of State	**Constant Density**
Viscous Regime	**Turbulent** **Reynolds-Averaged Navier-Stokes** (Selected automatically)
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence** **Exact Wall Distance** (Selected automatically) **Realizable K-Epsilon Two-Layer** (Selected automatically) **Two-Layer All y+ Wall Treatment** (Selected automatically)
Optional Models	**Segregated Fluid Temperature** **Thermal Comfort**

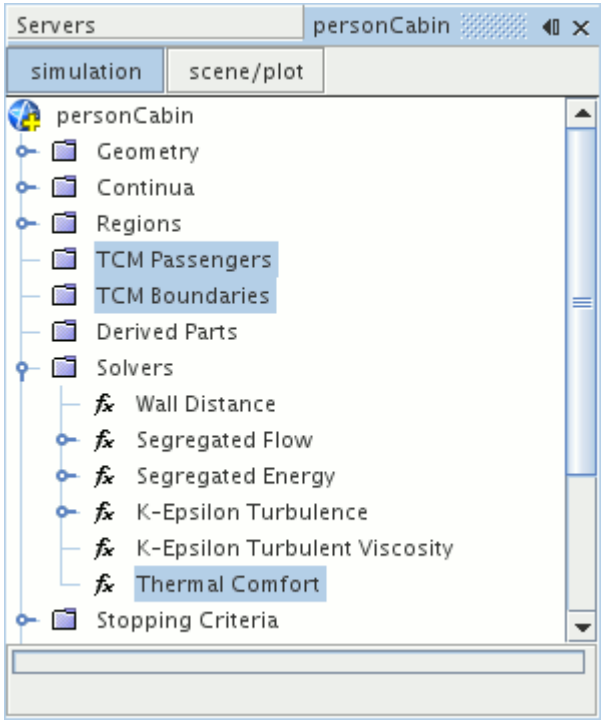
2. Click **Close**.

   The color of the **Physics 1** node turns from gray to blue to indicate that air models have been activated and the continuum is valid.

3. To review the models, open the **Physics 1** > **Models** node.



Activating the thermal comfort model adds two new nodes to the object tree, **TCM Passengers** and **TCM Boundaries**, as well as the **Solvers** > **Thermal Comfort** node. The new nodes are placeholders for information from the Thermal Comfort Wizard.



Preparing Boundaries for the Thermal Comfort Wizard

To specify thermal properties of each boundary within the Thermal Comfort Wizard, first choose a thermal specification at the boundary level for each boundary in the model.

The thermal values for each boundary are set using the wizard.

Note: The wizard can run the simulation for a number of iterations before calling the TIM program. During the initial iterations, all physics values are read from the active simulation, and not from the wizard. Enter alternative physics values for the initial iterations here if they are required. Initial values are overwritten by the TIM program when the TCM model calls it.

To choose the adiabatic, convection and temperature thermal specifications for the boundaries in the model:

1. Rename the **Regions** > **singleOccTCM** node to `Fluid`.
2. Expand the **Regions** > **Fluid** > **Boundaries** node.
3. Using the multi-select method, select all boundary nodes from **1 - Head** to **14 - RightFoot**, inclusive.
4. Edit the group of selected nodes.
5. Set the following properties:

Node	Property	Setting
Physics Conditions		

Node	Property	Setting
└ **Thermal Specification**	*Condition*	**Temperature**

6. Click **Close**.

Use a similar method to set the thermal specification of a number of the cabin boundaries to `Convection`.

7. Using the multi-select method, edit the following boundary nodes:
   - **Cabin**
   - **Console**
   - **Doors**
   - **Roof**
   - **Windows**
8. Set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Thermal Specification**	*Condition*	**Convection**

9. Click **Close**.

Contents:

Setting Up the Inlet and Outlet Boundaries

Setting Up the Inlet and Outlet Boundaries

The inlet flow values are also set using the Thermal Comfort Wizard. For them to be read correctly from the wizard into the simulation, select the correct boundary types before the TCM model is run.

Specify the inlet boundaries as mass flow inlets. Change the turbulence specification at the inlets to turbulence intensity with length scale. To specify the properties of the inlets simultaneously, use the multi-edit dialog.

1. Expand the **Regions** > **Fluid** > **Boundaries** node.
2. Using the multi-select method, edit all boundary nodes from **InletDashCentre** to **InletFeetRight**, inclusive.
3. In the *Multi-edit* dialog:
   a) Set *Type* to **Mass Flow Inlet**.
   b) Set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Turbulence Specification**	*Method*	**Intensity + Length Scale**

   c) Click **Close**.

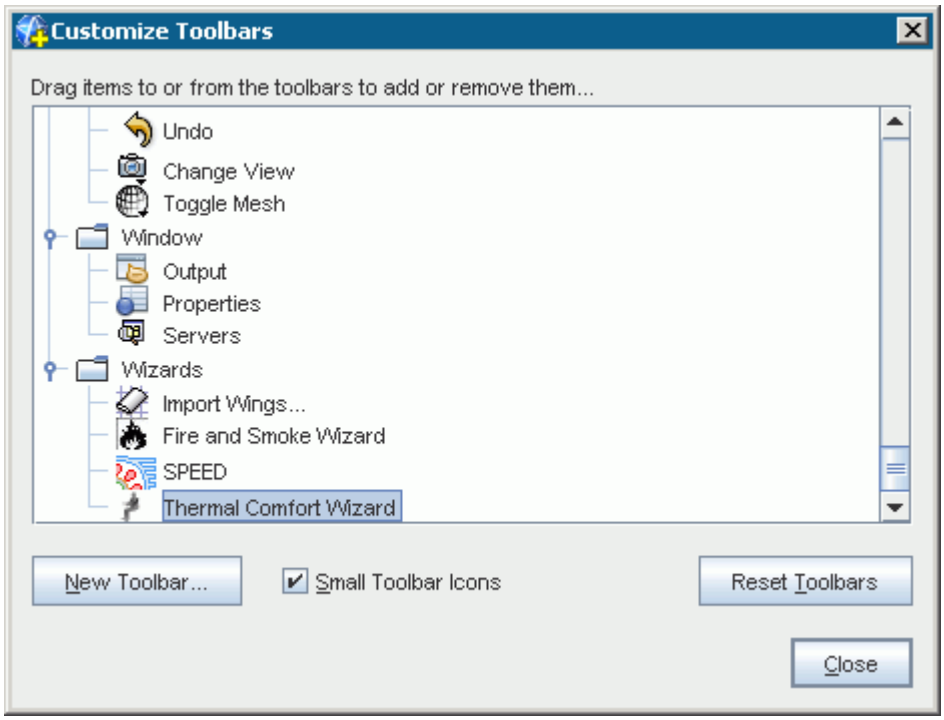
4. Select the **Outlet** node.
5. Set *Type* to **Outlet**.
6. Save the simulation.

Activating the Thermal Comfort Wizard

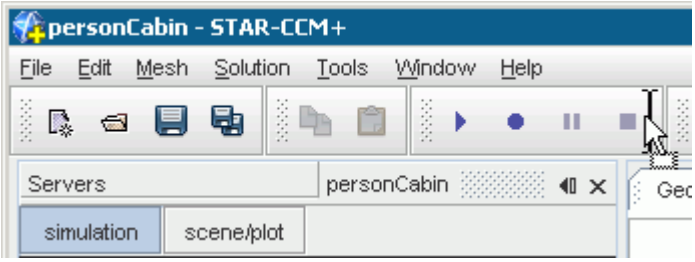
To activate the wizard, add it to a toolbar in the STAR-CCM+ workspace.

Use the following steps:

1. In the menu bar, select **Window** > **Toolbars** > **Customize...**
2. In the *Customize Toolbars* dialog, scroll down to the bottom of the list and select **Thermal Comfort Wizard**.



3. Drag and drop the **Thermal Comfort Wizard** icon from the dialog onto the end of one of the toolbars at the top of the workspace.



4. Click **Close**.

Using the Thermal Comfort Wizard

Use of the Thermal Comfort wizard begins with launching this feature.

To launch the Thermal Comfort wizard:

1. Click (**Thermal Comfort Wizard**).

Contents:

Entering Passenger Properties
Specifying Advanced Settings
Setting Thermal Properties
Saving TCM Settings

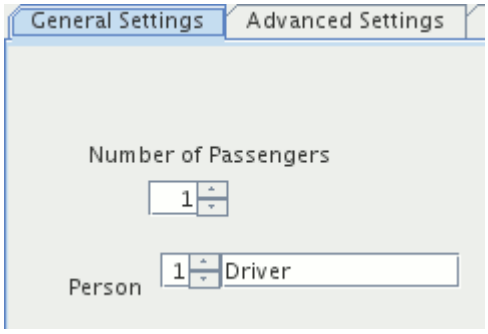
Entering Passenger Properties

To begin setting up the thermal comfort simulation, specify key physical characteristics of the occupants of the car.

In the *Thermal Comfort Wizard*, each occupant is referred to as a “passenger”. The properties of each passenger are entered in the *General Settings* tab of the wizard, such as their height, metabolic rate, and clothing resistance. In this tab, you also specify which boundary corresponds to each of the TCM body segments.

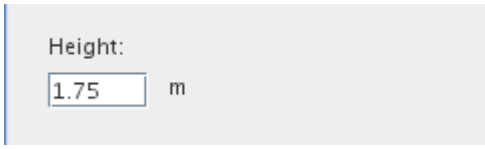
Number of Passengers and Person Name/Number:

1. As you have modeled only one passenger, you can retain the default settings for *Number of Passengers* and *Person*. Had the cabin contained several occupants, you would define each person with a different name and number. Notice that the first passenger is given the default name of `Driver`.



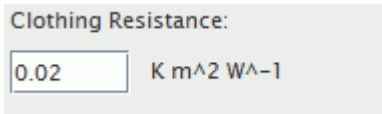
Height:

2. The TIM program uses the passenger height to calculate the body mass distributions for the vascular system and the surface area values. Set the *Height* to `1.75` m.



Clothing Resistance:

3. The clothing resistance defines the thermal conductivity. Set the *Clothing Resistance* to `0.02` K m^2/W.

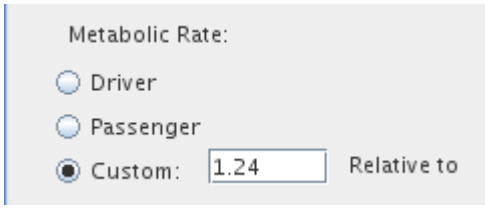


Metabolic Rate:

The metabolic rate is the rate at which the person releases heat energy. It depends on the level of activity of each person. It is set by selecting one of the **Driver** or **Passenger** radio buttons, or by entering a **Custom** value. When the **Passenger** option is selected the metabolic rate is set to `58 W/m^2`, which is equivalent to a person who is seated and relaxed. When the **Driver** option is chosen the metabolic rate is set to `75.4 W/m^2` (`1.3` times the passenger value), indicating that the driver is more active. The **Custom** option allows you to specify an alternative value by entering a scaling factor relative to 58 W/m^2. It is usually in the range of `0.8` to `2` (where `0.8` is equivalent to reclining, and `2` corresponds to medium activity.)

Set a custom metabolic rate for the single occupant:

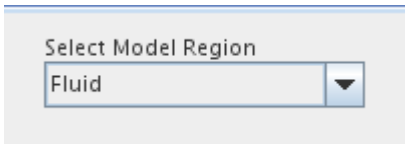
4. Under the *Metabolic Rate* property, select the **Custom** radio button and specify a value of `1.24`.



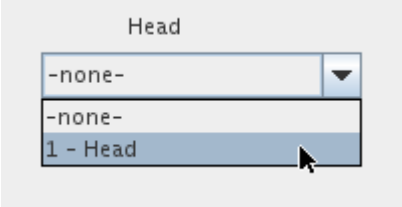
Assigning Body Segments:

Now specify which boundary corresponds to each passenger body segment. The wizard recognizes the body segment boundaries according to their names, so it is important to observe the Thermal Comfort Wizard naming conventions when preparing a model.

5. Make sure that **Fluid** is selected under the *Select Model Region* property.

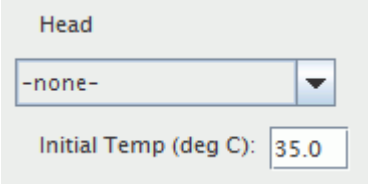


6. Under the *Head* property, use the drop-down menu to select the **1 - Head** boundary.



Note: Only those boundaries with names corresponding to relevant body segment are listed in the drop-down menu.

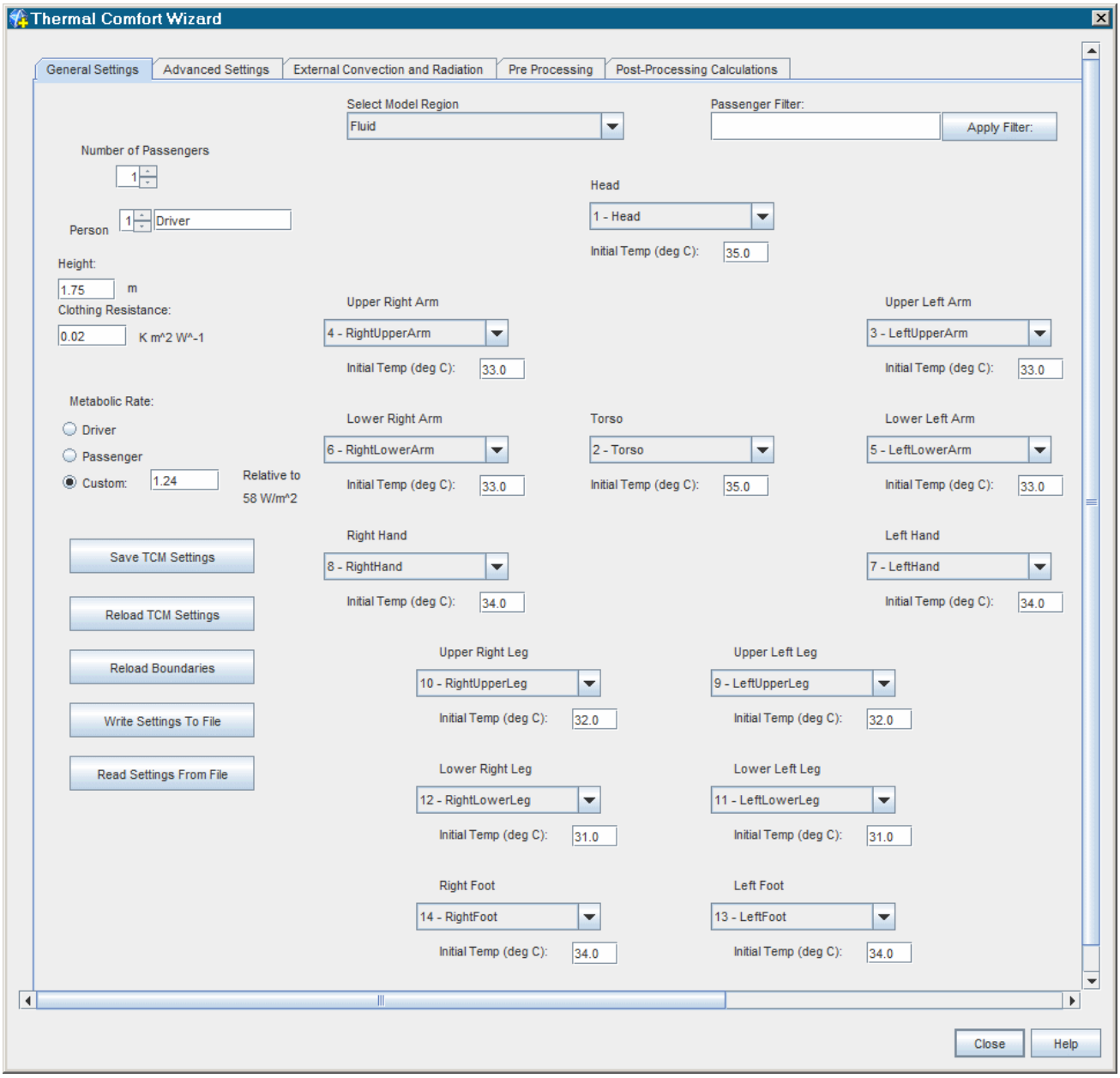
7. The initial temperature for the **Head** boundary is `35 C`. make sure that the *Initial Temp* property is set to `35`. All the temperatures that you specify in the *Thermal Comfort Wizard* are in degrees Celsius.



8. The corresponding boundary and initial temperature for each body segment is given in the table below. Use the data in this table to select the appropriate boundary and set the initial temperature for the remaining body segments.

Boundary Segment	Boundary	Initial Temperature (C)
Head	1 - Head	35
Torso	2 - Torso	35
Upper Left Arm	3 - LeftUpperArm	33
Upper Right Arm	4 - RightUpperArm	33
Lower Left Arm	5 - LeftLowerArm	33
Lower Right Arm	6 - RightLowerArm	33
Left Hand	7 - LeftHand	34
Right Hand	8 - RightHand	34
Upper Left Leg	9 - LeftUpperLeg	32
Upper Right Leg	10 - RightUpperLeg	32
Lower Left Leg	11 - LeftLowerLeg	31
Lower Right Leg	12 - RightLowerLeg	31
Left Foot	13 - LeftFoot	34
Right Foot	14 - RightFoot	34

The panel appears as follows:



The data that you entered in the *Thermal Comfort Wizard* dialog is retained automatically, allowing you to move between each of the tabs and even close and reopen the wizard without losing any information. The data in the wizard is only cleared if the STAR-CCM+ client is shut down.

Specifying Advanced Settings

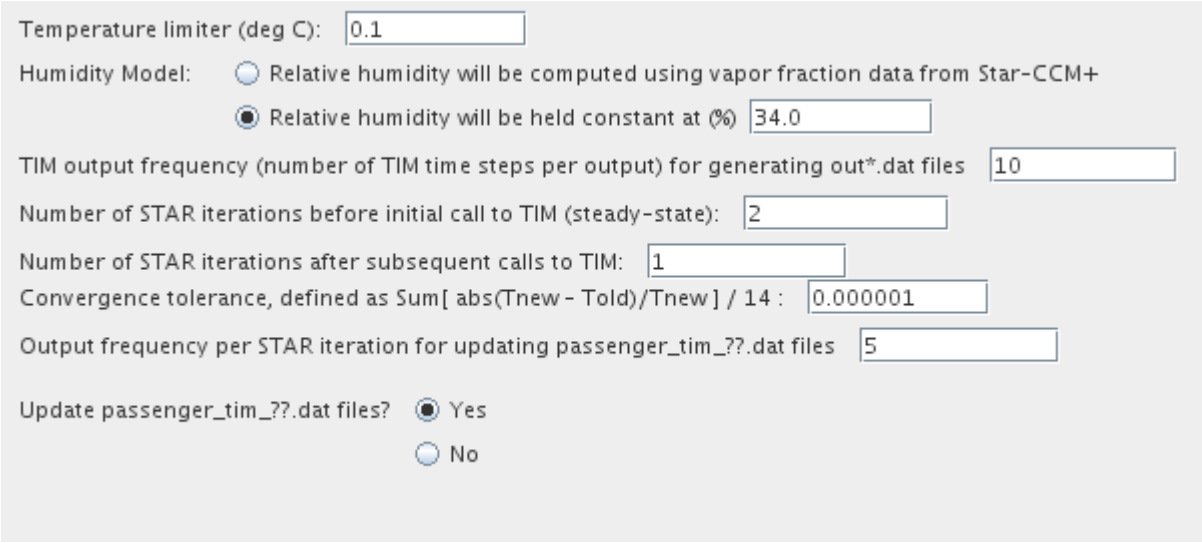
Refine the TCM settings such as the convergence tolerance and the frequency for generating output files.

Use the following steps:

1. Click the *Advanced Settings* tab at the top of the *Thermal Comfort Wizard* dialog.
2. Make the following settings:

Property	Setting
Temperature limiter (deg C)	`0.1`
Humidity Model	`Relative humidity will be held constant at (%)` `34`
Number of STAR iterations before initial call to TIM (steady-state)	`2`
Number of STAR iterations after subsequent calls to TIM	`1`
Convergence tolerance, defined as ...	`0.000001`
Update passenger_tim_??.dat file?	`Yes`

The panel appears as follows:



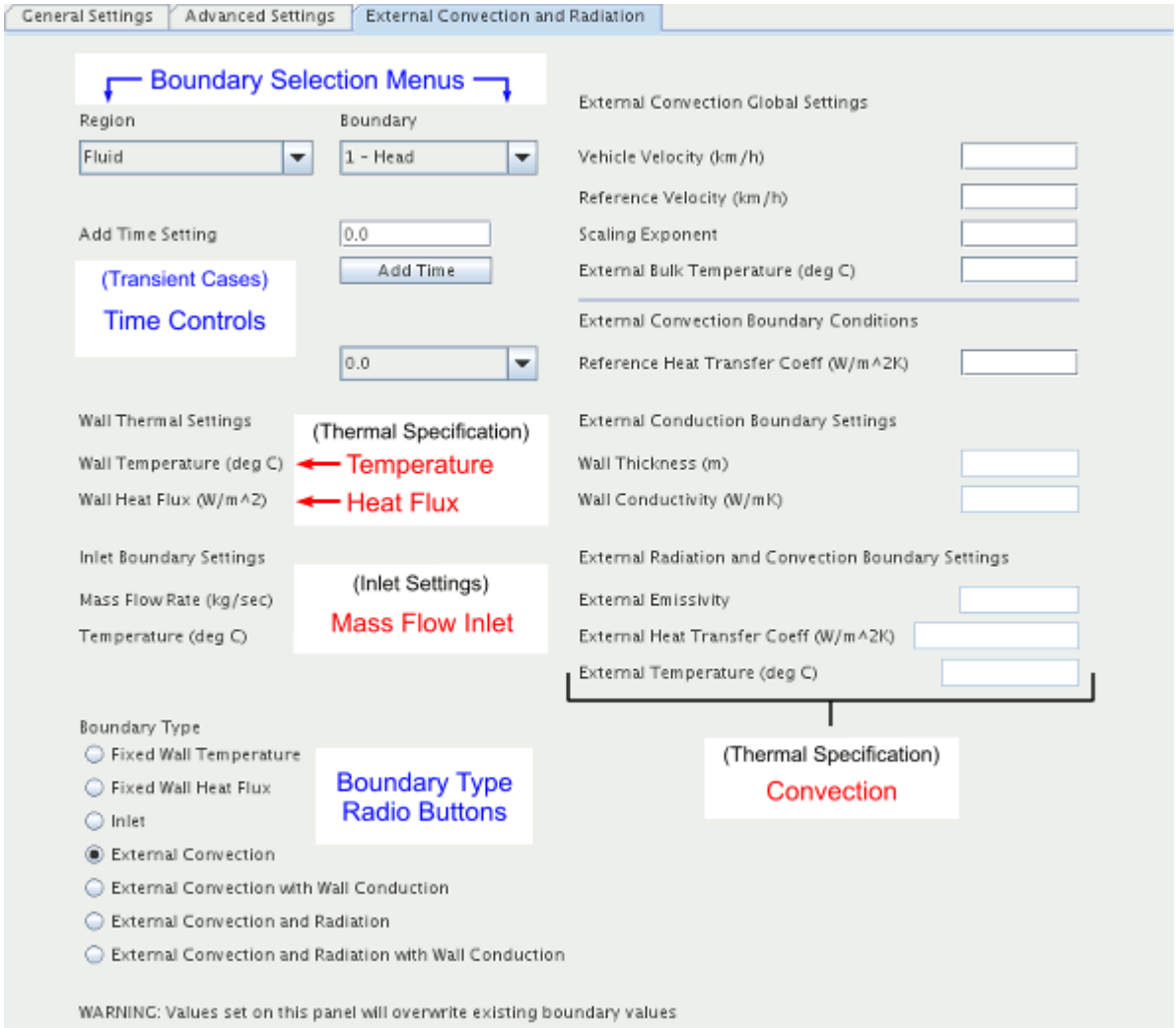
Setting Thermal Properties

Define the properties of any thermal boundaries in the model, and specify the mass flow rate and temperature at the inlets.

Use the following steps:

1. Click the *External Convection and Radiation* tab at the top of the *Thermal Comfort Wizard* dialog.

   The panel is divided into separate sections that correspond to the different types of thermal boundary, as shown in the diagram.



The thermal specification that was defined earlier for each boundary corresponds to a particular section of this panel.

External Convection Settings:

To specify the thermal properties for the Convection boundaries, values are entered under the External Convection section.

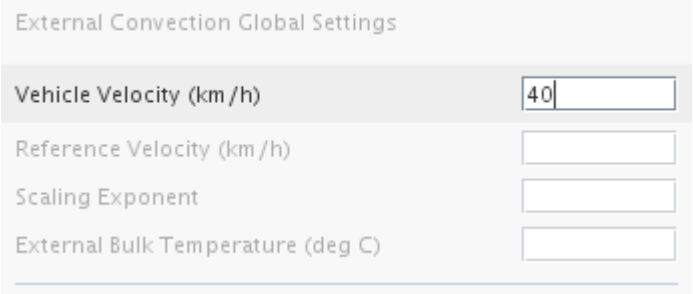
A number of settings, common to all convection boundaries are specified in the External Convection Global Settings section. The settings include: vehicle velocity, reference velocity, and scaling exponent. The settings allow heat transfer coefficients for a reference velocity to be applied to a model operating at the vehicle velocity.

The external bulk temperature is also specified globally. The remaining external convection settings are boundary-specific.

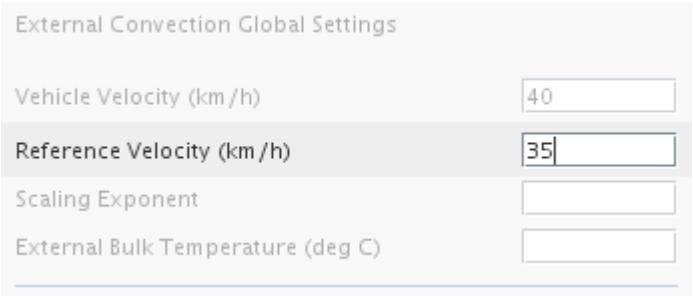
For the tutorial, the boundaries are defined as having convection only.

To specify the global external convection settings:

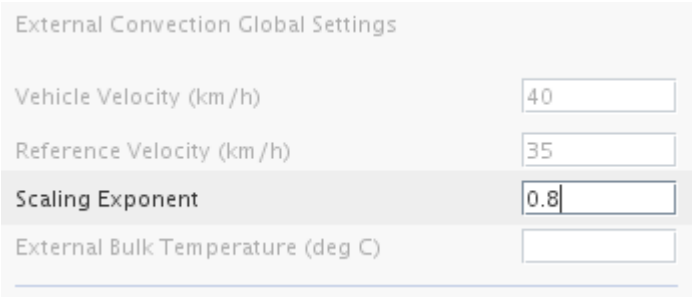
2. In the *External Convection Global Settings* section of the dialog, set the *Vehicle Velocity* to `40`.



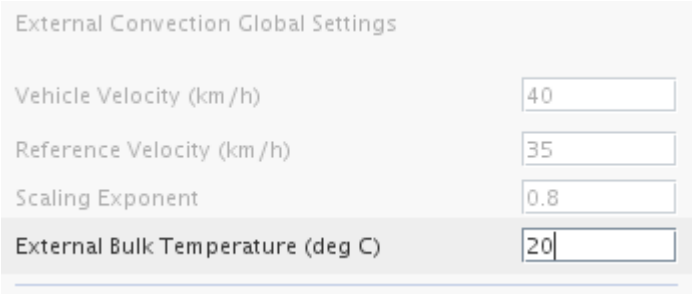
3. Set the *Reference Velocity* to `35`.



4. Set the *Scaling Exponent* to `0.8`.

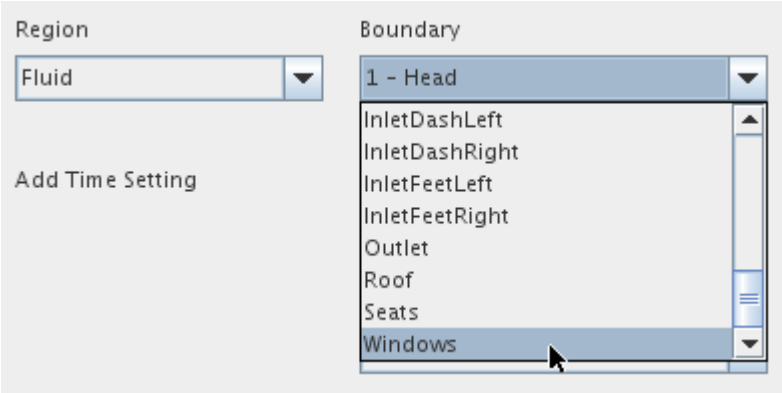


5. Set the *External Bulk Temperature* to `20`.



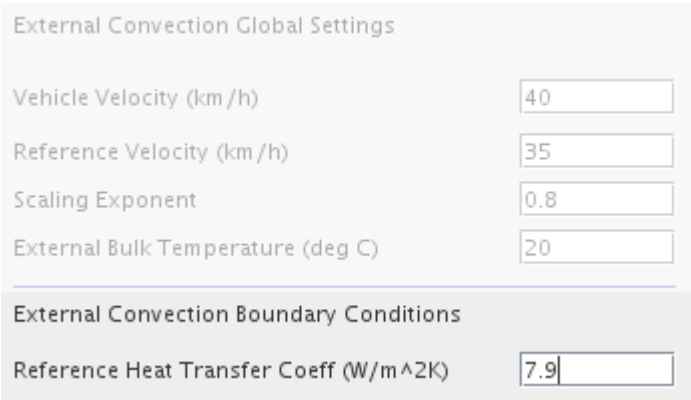
To specify the reference heat transfer coefficient for each of the convection boundaries:

6. Make sure that the **Region** is set to `Fluid`. Use the drop-down menu to set the **Boundary** to `Windows`.



By default the Boundary Type is set to `External Convection`.

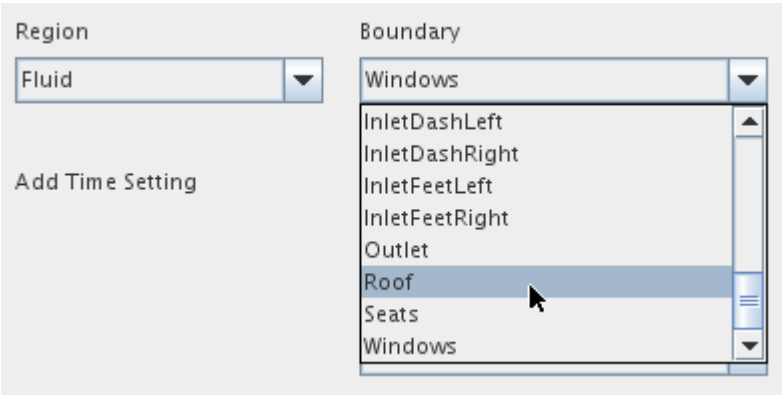
7. In the *External Convection Boundary Conditions* section on the right-hand-side of the panel, set the *Reference Heat Transfer Coefficient* to `7.9`.



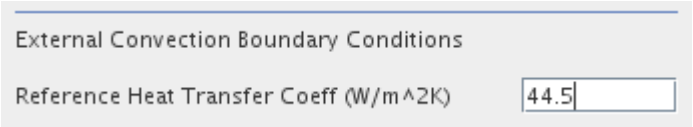
Settings are saved automatically for each boundary.

Select the next convection boundary and set the heat transfer coefficient.

8. Using the drop-down menu, set the **Boundary** to `Roof`.



9. In the *External Convection Settings* section, set the *Reference Heat Transfer Coefficient* to `44.5`.



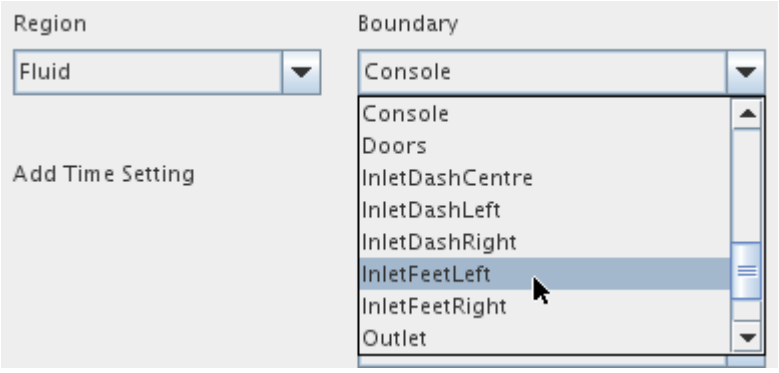
10. The properties of all the convection boundaries are shown in the table below. Use the technique described above and data from the table to set the properties of the remaining convection boundaries. You can check the properties that have been set for each boundary at any time by selecting the appropriate boundary from the drop-down list.

Boundary	Reference Heat Transfer Coefficient (W/m^2 K)
Windows	7.9
Roof	44.5
Cabin	44.5
Doors	44.5
Console	22.3

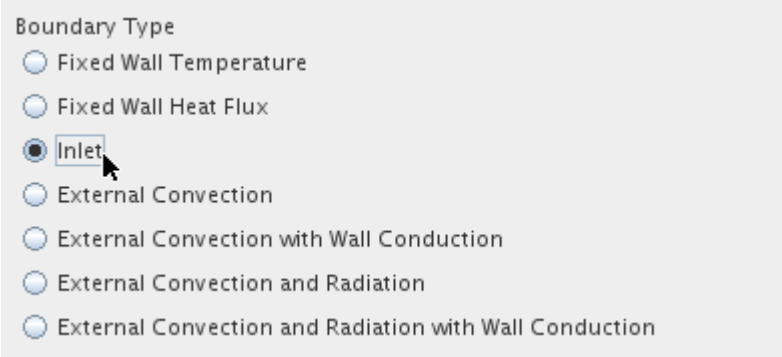
Inlet Boundary Settings

Set the inlet mass flow rate and temperature of the fluid conditions in the Inlet Boundary Settings section.

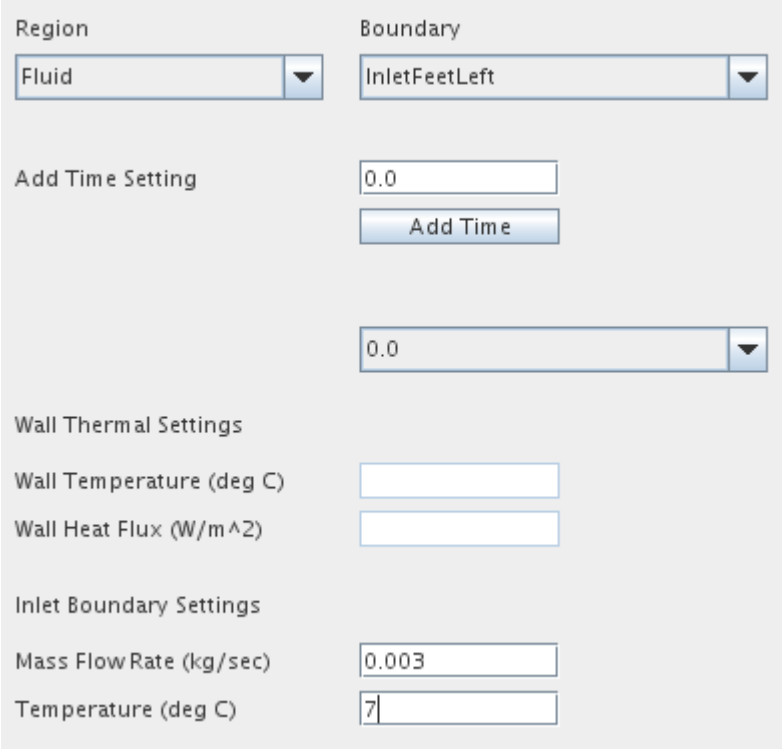
11. Set the **Boundary** to `InletFeetLeft`.



12. Select the **Inlet** radio button under *Boundary Type* to activate the input boxes for the Inlet Boundary Settings.



13. Enter a *Mass Flow Rate* of `0.003` in the `Inlet Boundary Settings` section.
14. Set the *Temperature* to `7`



15. Using this technique and values from the following table, set the properties of the remaining inlet boundaries.

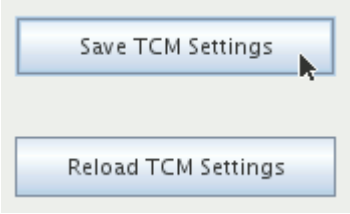
Boundary	Mass Flow Rate (kg/s)	Temperature (C)
InletFeetLeft	0.003	7
InletDashCentre	0.006	14
InletDashLeft	0.007	13
InletDashRight	0.007	13
InletFeetRight	0.003	7

Saving TCM Settings

When you are finished working with the Thermal Comfort wizard, save the settings in a step that is distinct from the step of saving the simulation file.

To save TCM settings:

1. Click **Save TCM Settings**, and click **Close**.



The information from the wizard is added to the managers in the object tree.

Setting Stopping Criteria

Specify the maximum number of iterations.

1. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to `300`.

Creating a Scalar Scene

To show the surface temperature of the different body segments of the person, set up a scalar scene.

Use the following steps:

1. Create a scalar scene.
2. Edit the **Scenes** > **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Parts** node.
3. In the *Edit* dialog, expand the **Regions** > **Fluid** node.
4. Select the body boundaries **1 - Head** boundary to **14 - RightFoot**, inclusive.
5. Click **OK**.
6. Select the **Scalar 1** > **Scalar Field** node.
7. Set *Function* to `Temperature`.
8. Set *Units* to `C`.
9. Delete the **Displayers** > **Outline 1** node.
10. Create a new geometry displayer.
11. Edit the **Displayers** > **Geometry 1** > **Parts** node.
12. In the *Edit* dialog, expand the **Regions** > **Fluid** > **Boundaries** node.
13. Select the following boundaries:
    - **Cabin**
    - **Console**

- **Doors**
- **Roof**
- **Seats**
- **Windows**

14. Click **OK**.
15. Select the **Geometry 1** node and set the following properties:
    - Deactivate *Outline*.
    - Activate *Surface*.
    - Set *Opacity* to `0.3`.

Contents:

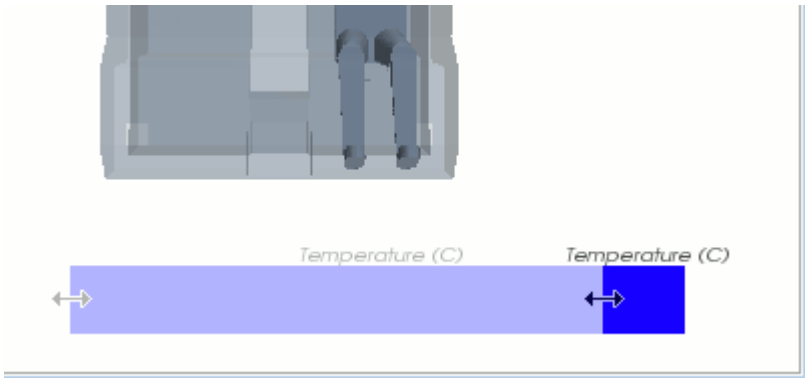
Adjusting the Color Bar

Adjusting the Color Bar

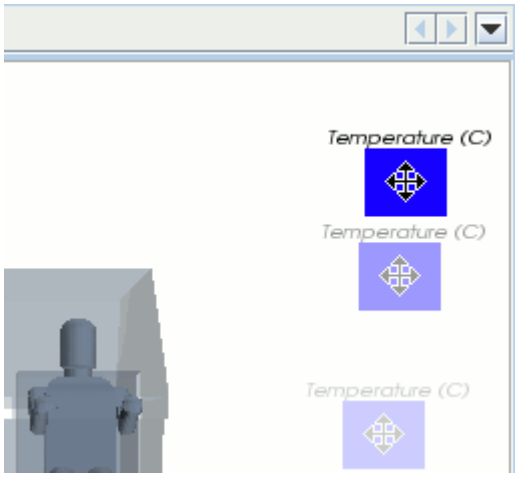
To optimize your view of the scalar data, adjust the color bar in the scene.

To adjust the size and position of the color bar in *Scalar Scene 1*:

1. In the *Graphics* window, click the left-hand edge of the color bar and drag it to decrease the width.

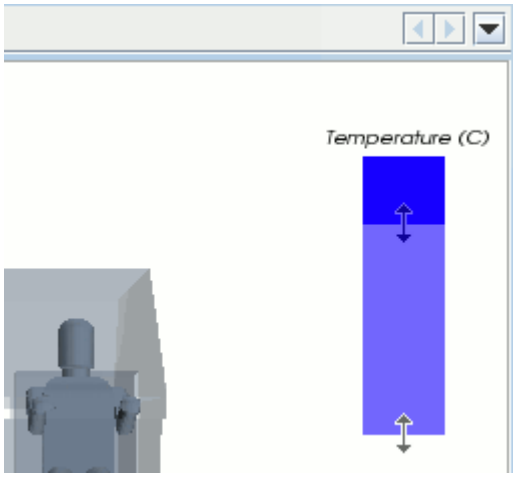


2. Drag the color bar into the top right corner of the scene.



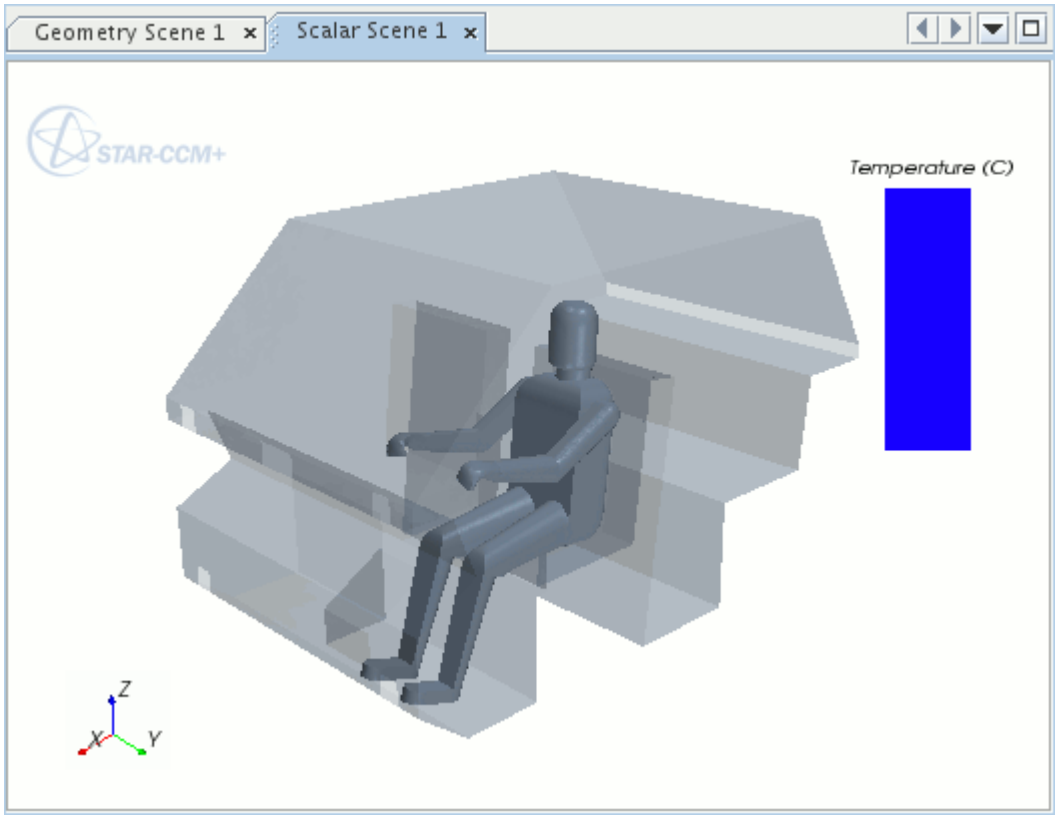
The color bar changes orientation automatically.

3. Drag the bottom edge of the color bar to increase the length.



4. Click (**Save-Restore-Select views**) and select **Restore View** > **View 1**.

   The scene appears as follows:



5. Save the simulation.

Running the Solver

The simulation is now ready to run.

To obtain a solution:

1. Run the simulation.
2. To watch the solution being updated while the simulation is running, look at the scalar scene.

   Note: The simulation is run for 2 iterations before the TIM program starts and updates the body surface temperatures.

3. When the simulation has finished running, save it.

Visualizing the Results with Streamlines

The solution for the surface temperature of each body segment is shown in the scalar scene. Also show the flow path of the air particles that enter the cabin through the inlet vents.

To show the air flow path, set up a streamlines displayer:

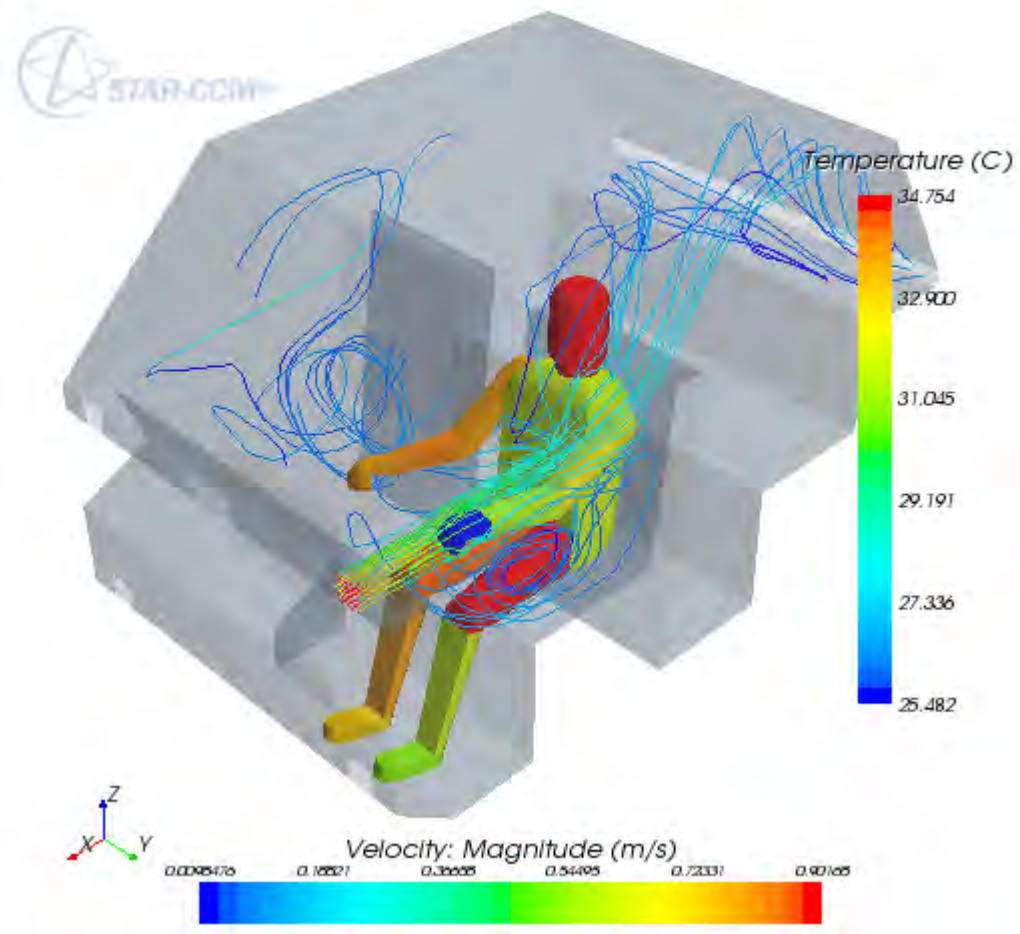
1. Make the *Scalar Scene 1* display active.
2. Create a streamline derived part.

3. In the *Create Streamline* inset panel, make the following settings:

Property	Setting
Input Parts	`[Fluid]`
Vector Field	`Velocity`
Seed Parts:	`InletDashLeft`

4. Click **OK**.
5. Click **Create** and **Close**.
6. Select the **Scenes** > **Scalar Scene 1** > **Displayers** > **Streamline Stream 1** > **Scalar Field** node.
7. Set *Function* to `Velocity > Magnitude`.
8. In the *Graphics* window, click in the center of the color bar and drag it to a more suitable position.

The scene appears as shown below.



9. Save the simulation.

Summary

The Thermal Comfort Wizard tutorial used an analysis of a single occupant in the cabin of a car to introduce a variety of STAR-CCM+ features.

The features that were introduced include:

- Generating the volume mesh.
- Setting up the physics continuum.
- Preparing boundaries for the *Thermal Comfort Wizard*.
- Activating the *Thermal Comfort Model*.
- Activating the *Thermal Comfort Wizard*.
- Using the *Thermal Comfort Wizard* to:
  - Enter passenger properties;
  - Specify advanced settings;
  - Set the thermal properties of other boundaries in the model;
  - Save TCM settings.

- Creating a scalar scene and adjusting the color bar.
- Running the TCM model.
- Visualizing streamline results.

Multiphase Flow

The tutorials in this set illustrate various STAR-CCM+ features for simulating multiphase fluid flow problems

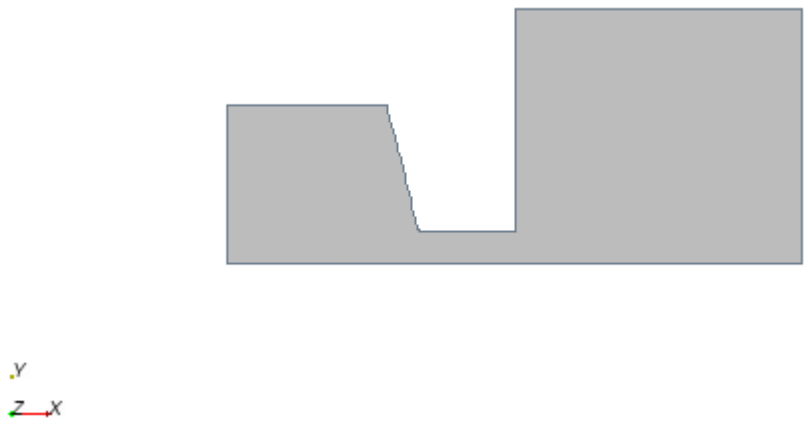
Contents:

VOF: Gravity-Driven Flow
VOF: Capillary Effects
VOF: Cavitation
VOF: Boiling
VOF: Melting-Solidification
Lagrangian: Particle-Laden Flow
Lagrangian: Solid Particle Erosion
Eulerian: Hibiki's Bubble Column
Eulerian: Mixture Settling
Eulerian: Bubble Formation in a Fluidized Bed
Eulerian: Degassing Boundary
Eulerian: Wall Boiling
Eulerian: Conjugate Heat Transfer Wall Boiling
Eulerian Multiple Flow Regimes: Pressurized Water Reactor
Eulerian: Pipe Flow with Multi-Speed AMUSIG
Fluid Film: Liquid Film Flow
Fluid Film: Binary Liquid Film Flow with Evaporation and Edge Stripping
Fluid Film - VOF: Rivulets on an Inclined Plane
Dispersed Multiphase: Airfoil Icing

VOF: Gravity-Driven Flow

This tutorial demonstrates how to set up a gravity-driven flow problem in STAR-CCM+. It simulates two-dimensional gravity-driven compressible flow through a channel connecting two chambers.

The geometry is shown below. Initially, the chamber on the left is filled with water; the one on the right and the connecting channel with air. All boundaries are solid walls except for the horizontal top left surface where a constant (atmospheric) static pressure is applied.



Under the action of gravity, water flows into the right chamber under assumed turbulent conditions. At the same time, water also flows in through the top left boundary to maintain a constant fluid level. The pressure in the right chamber increases due to the air compression, resulting in a reduction of the flow rate thorough the pressure boundary. After some time, all fluid elements are at rest and in a hydro-static equilibrium.

Contents:

Prerequisites
Importing the Mesh and Naming the Simulation
Converting to a Two-Dimensional Mesh
Visualizing the Mesh Interior
Selecting the Physics Models
Setting the Material Properties
Setting Initial Conditions and Reference Values
Setting Boundary Conditions
Setting Solver Parameters and Stopping Criteria
Visualizing and Initializing the Solution
Running the Simulation
Visualizing Results
Animating the Results
Summary

Prerequisites

The instructions in the VOF: Gravity-Driven Flow tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Importing a volume mesh	Simple Pipe Flow
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Importing the Mesh and Naming the Simulation

Launch STAR-CCM+, import the supplied geometry, and save the simulation.

A one-cell-thick, three-dimensional, polyhedral cell mesh has been prepared for this analysis.

To import the mesh and name the simulation:

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh**.
4. In the *Open* dialog, choose `doc/startutorialsdata/multiphaseFlow/data/gridfs.ccm` then click **Open**.

   STAR-CCM+ provides feedback on the import process, which takes a few seconds, in the *Output* window. A geometry scene is created in the *Graphics* window.
5. Save the simulation as `freeSurface.sim`.

Converting to a Two-Dimensional Mesh

You can now convert the three-dimensional mesh to two dimensions.

There are special requirements in STAR-CCM+ for converting three-dimensional meshes into two-dimensional meshes. These requirements are:

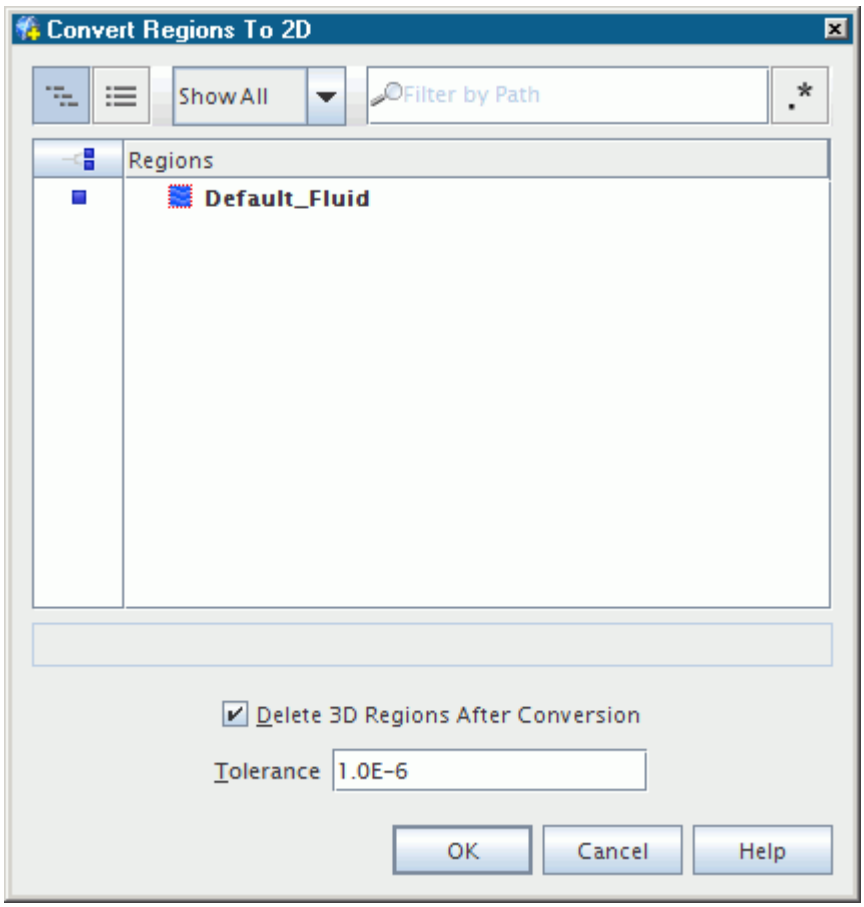
- The grid must be aligned with the X-Y plane.
- The grid must have a boundary plane at the Z = 0 location.

The mesh that is imported for this tutorial is built with these requirements in mind. Were the grid not to conform to the above conditions, it would have been necessary to realign the region using the transformation and rotation facilities in STAR-CCM+.

To convert the mesh to two dimensions:

1. Select **Mesh** > **Convert to 2D**.

   The *Convert Regions To 2D* dialog appears.
2. Make sure that the **Delete 3D Regions After Conversion** option is activated.



3. Click **OK**.

   Once you click **OK**, the mesh conversion takes place and the new two-dimensional mesh is shown, viewed from the z-direction, in the *Geometry Scene 1* display. The mouse rotation option is suppressed for two-dimensional scenes.
4. Right-click the **Continua** > **Physics 1** node and select **Delete**.
5. In the confirmation dialog, click **Yes**.

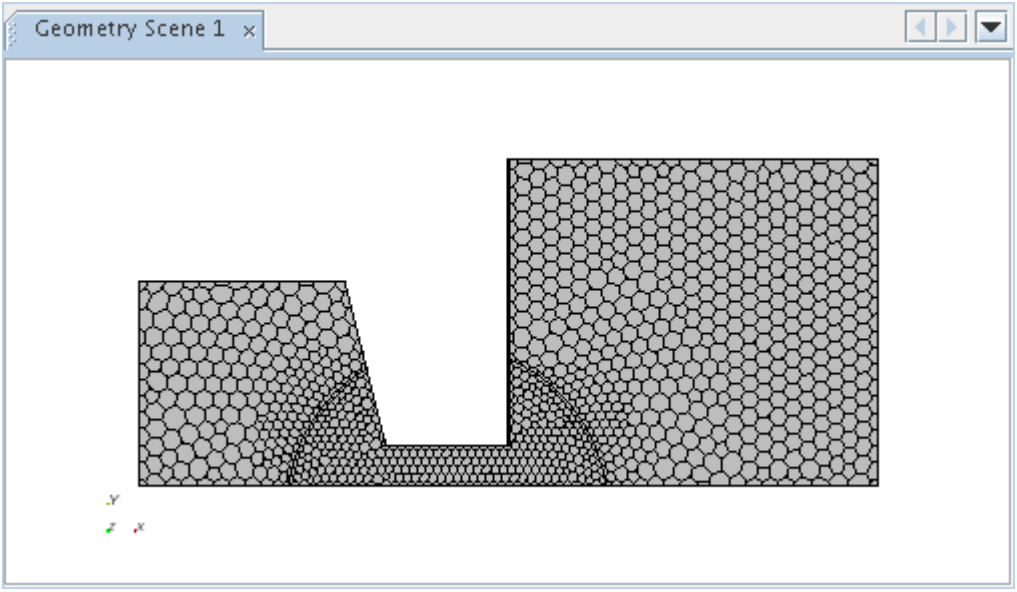
Visualizing the Mesh Interior

After you have converted the volume mesh to a two-dimensional mesh, you can view the mesh structure.

To visualize the mesh:

1. In the simulation tree, select the **Scenes** > **Geometry Scene 1** > **Displayers** > **Geometry 1** node.
2. To display the surface mesh, activate the *Mesh* property.

   The interior mesh structure is shown below.



Selecting the Physics Models

Physics models define the primary variables of the simulation, including pressure, temperature, velocity, and what mathematical formulation is used to generate the solution.

In this example, the flow is turbulent. Use the default K-Epsilon Turbulence model, and apply a gravitational force in the -y direction. As the problem also involves multiphase flow, two fluids (air and water) are required for the analysis. However, since these fluids occupy the same domain, only one continuum and one mesh region are required to set up the simulation. By default, a continuum that is called **Physics 1 2D** is created when the mesh is converted to two-dimensional.

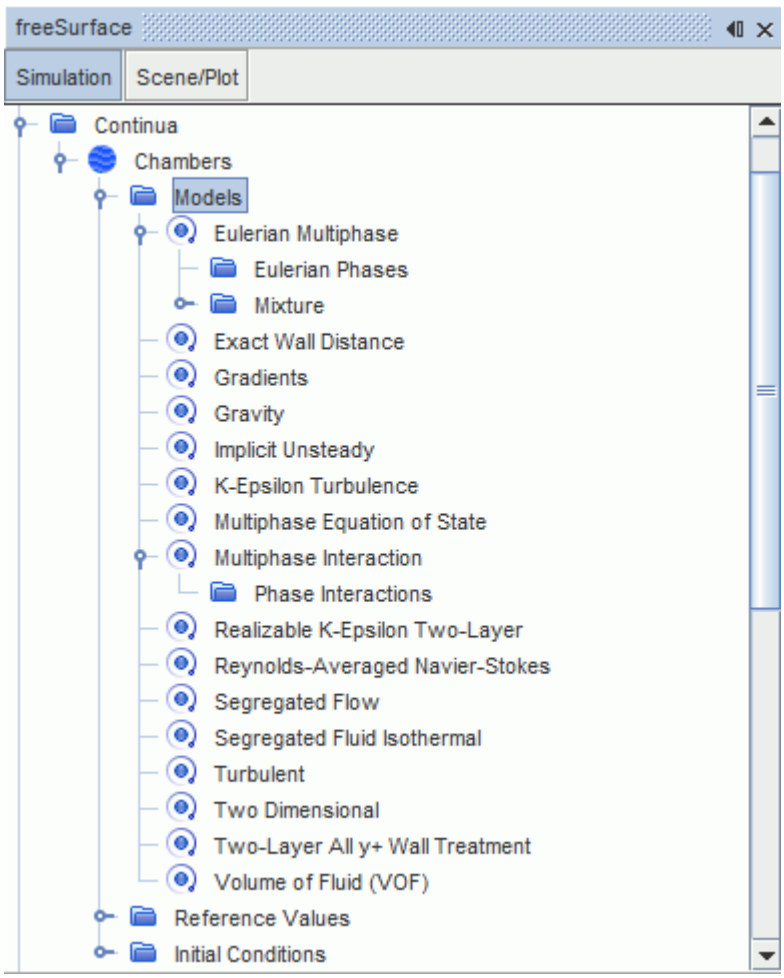
To select the physics models:

1. Rename the **Physics 1 2D** node to **Chambers**.
2. For the physics continuum, **Continua** > **Chambers**, select the following models in order:

Group Box	Model
Enabled Models	**Two Dimensional** (Pre-selected)
Time	**Implicit Unsteady**
Material	**Eulerian Multiphase** **Multiphase Interaction** (selected automatically)
Eulerian Multiphase Model	**Volume of Fluid (VOF)** **Gradients** (selected automatically) **Multiphase Equation of State** (selected automatically) **Segregated Flow** (selected automatically)
Viscous Regime	**Turbulent**

Group Box	Model
	Reynolds-Averaged Navier-Stokes (selected automatically)
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence** **Exact Wall Distance** (selected automatically) **Realizable K-Epsilon Two-Layer** (selected automatically) **Two-Layer All y+ Wall Treatment** (selected automatically)
Optional Models	**Gravity** **Segregated Fluid Isothermal**

3. Click **Close**.
4. To review the models, open the **Chambers** > **Models** node.



5. Save the simulation.

Setting the Material Properties

Define the materials corresponding to each of the mixture components under the **Eulerian Phases** node.

To define the phases and set the material properties:

1. In the **Chambers** continuum, select the **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and create a new phase.
2. Rename the **Phase 1** node to **H2O**.
3. For the **H2O** phase, select the following models:

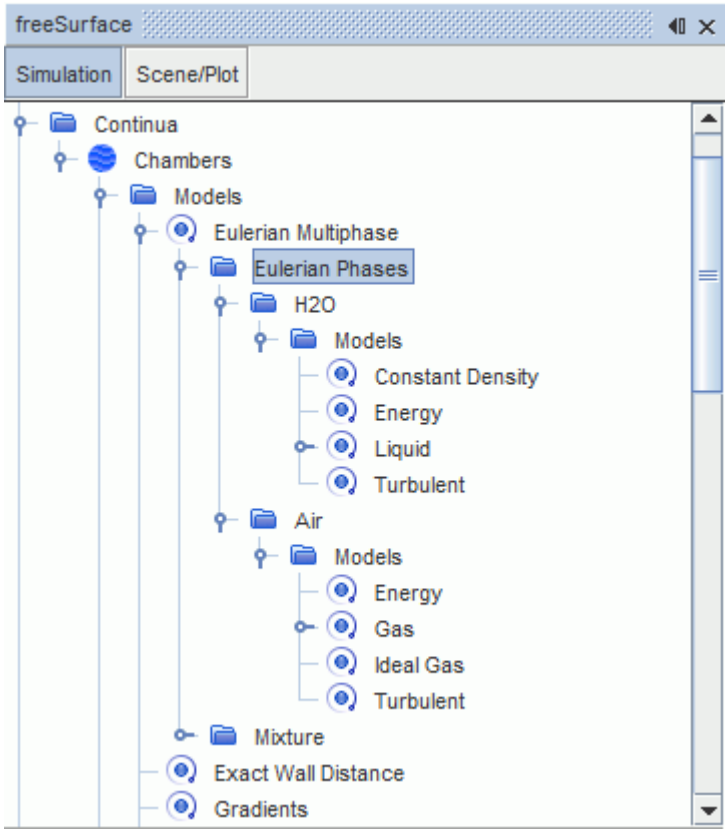
Group Box	Model
Material	**Liquid**
Equation of State	**Constant Density**

4. Click **Close**.
5. Create a second phase and rename it to **Air**.
6. For the **Air** phase, select the following models:

Group Box	Model
Material	**Gas**
Equation of State	**Ideal Gas**

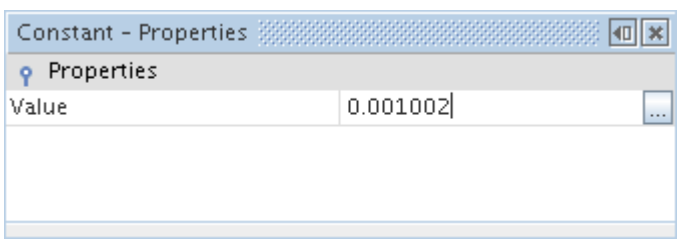
7. Click **Close**.

   The **Eulerian Multiphase** node contains the phase model objects as shown below:



The default selections for materials—water and air—are appropriate for this exercise, so there is no need to replace them with other materials from the STAR-CCM+ material database. The existing material properties for air are suitable but the dynamic viscosity of water must be changed.

8. Select the **H2O** > **Models** > **Liquid** > **H2O** > **Material Properties** > **Dynamic Viscosity** > **Constant** node and change the dynamic viscosity **Value** to `0.001002 Pa-s`.



9. Save the simulation.

Setting Initial Conditions and Reference Values

Set the initial conditions and reference values for the physics continuum. You define a field function to use for specifying the initial condition for the spatial distribution of the two fluids in the continuum.

The usual practice for initially stationary fluids is to specify low, non-zero values for the turbulence parameters. If these quantities are set to zero, numerical problems can arise.

The initial condition for the spatial distribution of the two fluids in the continuum is that you only have water in the left-hand chamber and only air in the right-hand chamber and channel. Both fluids are also stationary. A convenient way of specifying such a distribution is by creating and using field functions.

The direction and magnitude of the gravity vector are set using the **Reference Values** node. In this case, a gravity force is applied in the negative y-direction.

To set the initial conditions and reference values:

Define the field function to use for specifying the initial condition for the spatial distribution of the two fluids in the continuum:

1. Right-click the **Tools** > **Field Functions** node and select **New** > **Scalar** from the pop-up menu.
2. Rename the **User Field Function 1** to `Initial Distribution`.
3. Select the **Initial Distribution** node and set the following properties:

Property	Setting
Function Name	`Initial Distribution`
Definition	`($$Position[0]<=-1)?1:0`

Set the initial conditions and reference values for the **Chambers** continuum.

4. Edit the **Initial Conditions** node and set the following properties:

Node	Property	Setting
Turbulence Intensity	**Value**	`0.01`
Turbulent Viscosity Ratio	**Value**	`100.0`
Volume Fraction	**Method**	**Composite**
└ **Composite** > **H2O**	**Method**	**Field Function**
	Scalar Function	**Initial Distribution**

It is not necessary to define a similar distribution for the Air node because any user-specified initial condition for the volume fraction of the last-defined phase in multiphase flows is always ignored. Instead, this condition is obtained by subtracting the sum of the volume fraction distributions of the other phases from 1.0.

5. Edit the **Reference Values** node and set the following properties:

Node	Property	Setting
Gravity	**Value**	`[0.0,-9.81] m/s^2`

6. Save the simulation.

Setting Boundary Conditions

Define the boundary types and specify the appropriate property values. The geometry that is used for this tutorial has six boundaries. Assign no-slip wall conditions to five of them, and assign the remaining boundary a pressure boundary condition.

To set the boundary conditions:

1. Rename the **Default_Fluid 2D** node to **Fluid**.
2. Edit the **Fluid** > **Boundaries** node and set the boundary types as follows:

Boundary	Type
Bottom	**Wall**
Left	**Wall**
Middle	**Wall**
Right	**Wall**
TopRight	**Wall**
TopLeft	**Pressure Outlet**

The default conditions are suitable for the wall boundaries, but the conditions for the pressure boundary must be reset.

3. Edit the **TopLeft** > **Physics Values** node and set the following properties:

Node	Property	Setting
Turbulence Intensity	**Value**	`0.01`
Turbulent Viscosity Ratio	**Value**	`100.0`
Volume Fraction	**Value**	`[1.0, 0.0]`

   The **Volume Fraction** setting enforces the condition that only liquid water enters the solution domain through that boundary.
4. Save the simulation.

Setting Solver Parameters and Stopping Criteria

As you are solving an unsteady problem, it is necessary to specify the time-step size and the elapsed simulation time. Run this calculation for 5.0 s with a time-step size of 0.005 s.

To set the solver parameters and stopping criteria:

1. Edit the **Solvers** node and set the following property:

Node	Property	Setting
Implicit Unsteady	**Time-Step**	`0.005 s`

2. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Physical Time	**Enabled**	Activated
	Maximum Physical Time	5.0 s
Maximum Steps	**Enabled**	Deactivate
Maximum Inner Iterations	**Maximum Inner Iterations**	5

3. Save the simulation.

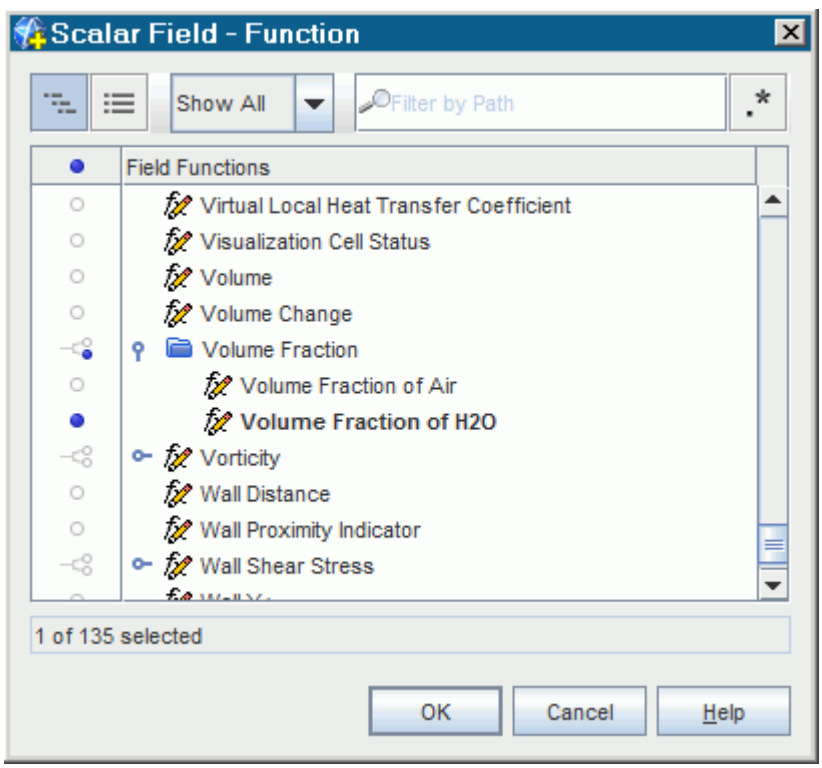
Visualizing and Initializing the Solution

You create a scalar scene to view the results of the simulation. You can view the air and water distributions throughout the run.

To visualize the results of the simulation:

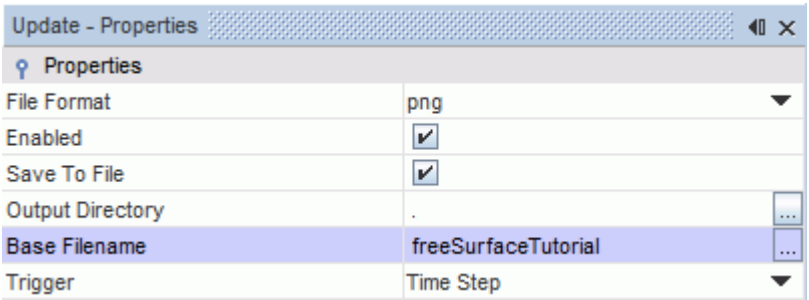
1. Right-click the **Scenes** node and select **New Scene** > **Scalar**.

   The *Scalar Scene 1* display appears.
2. Select the new **Scenes** > **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Scalar Field** node and set *Function* to **Volume Fraction** > **Volume Fraction of H2O**.

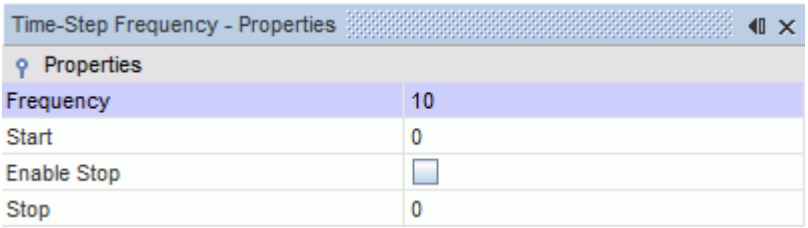


3. Select the **Scalar 1** node and set *Contour Style* to **Smooth Filled**.
4. Select the **Attributes** > **Update** node and set *Trigger* to **Time Step**.
5. Activate **Save To File**.

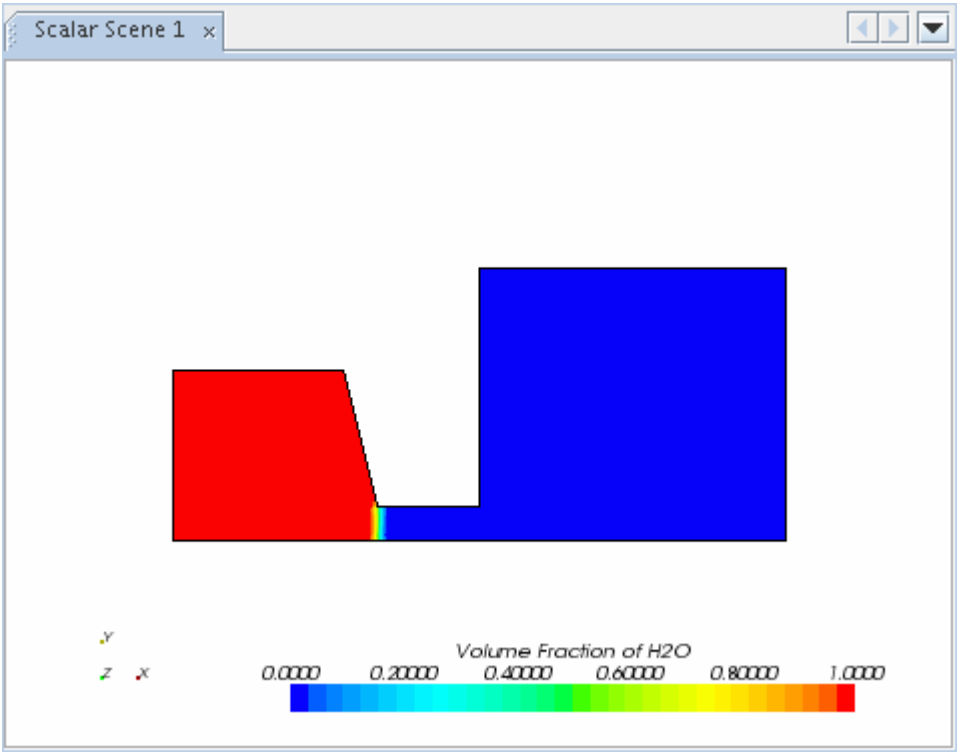
6. Specify the *Base Filename* as desired.



7. Select the **Attributes** > **Update** > **Time-Step Frequency** node and set *Frequency* to `10`.



8. Click (**Initialize Solution**) in the top toolbar or use the **Solution** > **Initialize Solution** menu item.
9. Make the *Scalar Scene 1* display active to look at the initialization result.



At the start of the run, the left chamber is entirely filled with water whereas the right chamber and the connecting channel are entirely filled with air. A small region in which both fluids are apparently present is visible at the interface between the two fluids, but this effect is due to the coarseness of the mesh.

10. Save the simulation.

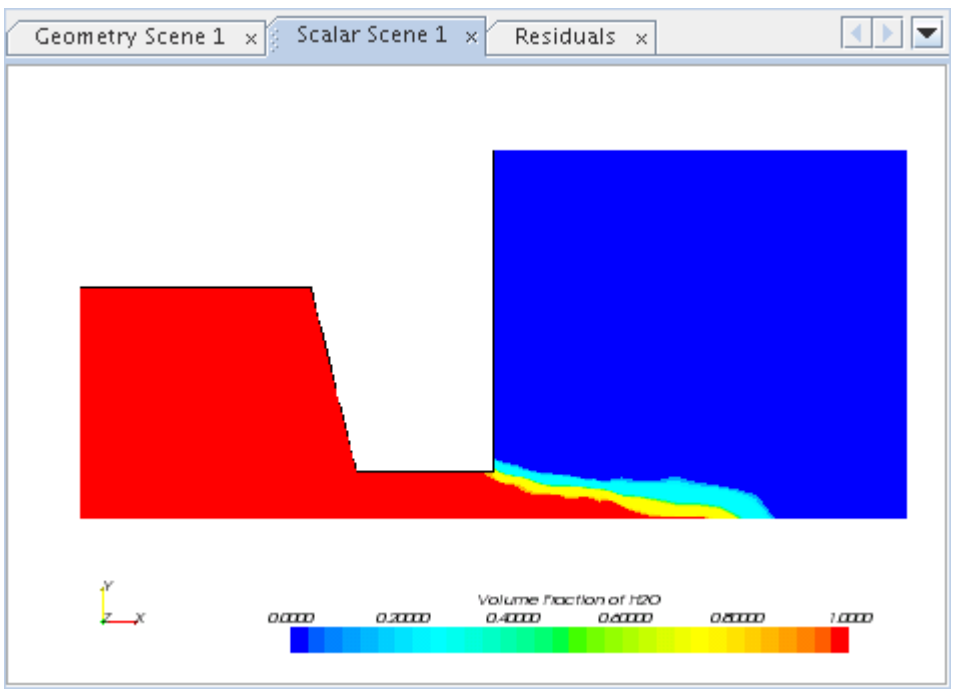
Running the Simulation

Preparation of the simulation is now complete, and the simulation can be run.

To run the simulation:

1. Click (**Run**) in the *Solution* toolbar.

   The solution progress is displayed in the *Output* window. The *Residuals* display is created automatically in the *Graphics* window and shows the progress of the solvers. You can also click the *Scalar Scene 1* tab at the top of the *Graphics* window to view the simulation progress.



   During the run, you can stop the process by clicking (**Stop**) on the toolbar. If you do stop the simulation, you can click (**Run**) to resume it.

   If left alone, the simulation continues until 1000 time-steps are complete. The numbers that are displayed in the *Output* window represent the inner iterations, not the time-steps that the solver performs. As there are 5 inner iterations per time-step, you expect the run to perform up to 5,000 inner iterations.

   When the last iteration is complete, the following message is displayed in the *Output* window:

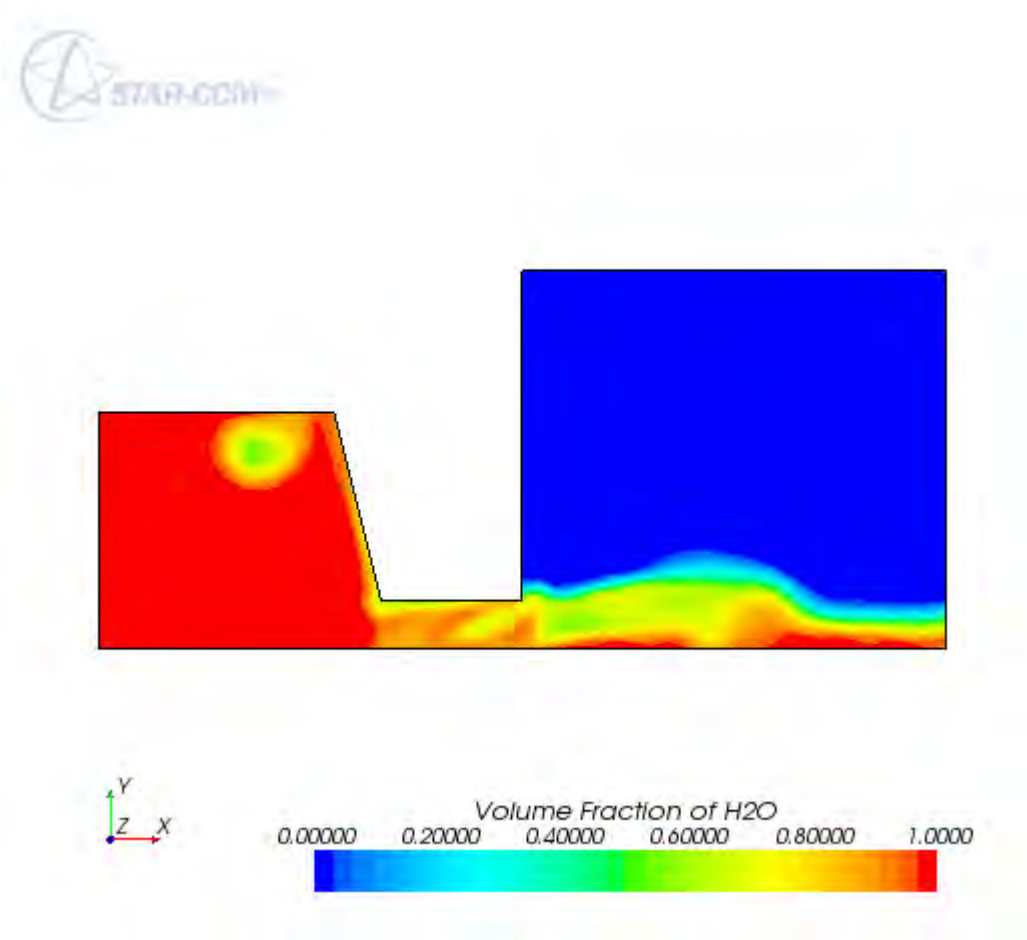
   ```
   Stopping criterion Maximum Physical Time satisfied.
   ```

2. When the simulation has finished running, save it.

Visualizing Results

View the results of the simulation.

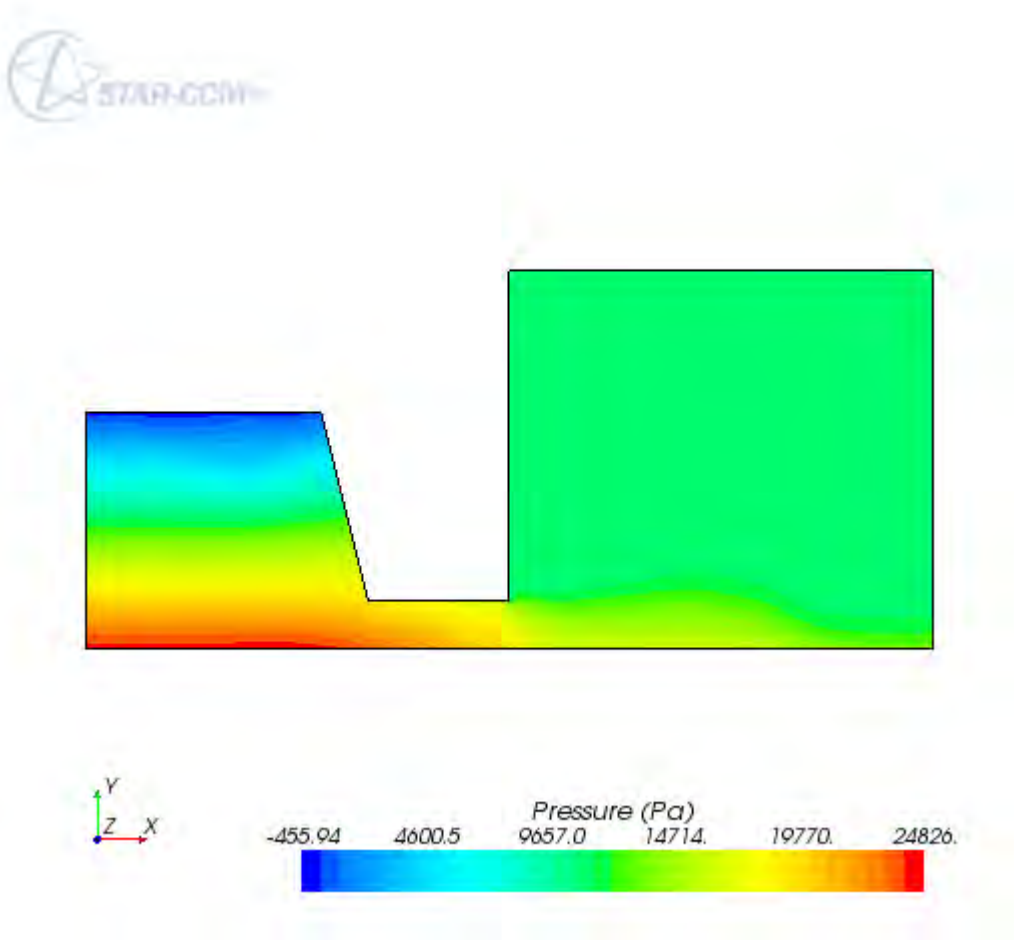
The *Scalar Scene 1* display shows the water volume fraction profile at the end of the 5.0 s run.



To display the pressure profile:

1. Right-click the scalar bar in the display.
2. Select **Pressure** from the pop-up menu.

   The following screenshot shows the compression of the air in the right chamber.

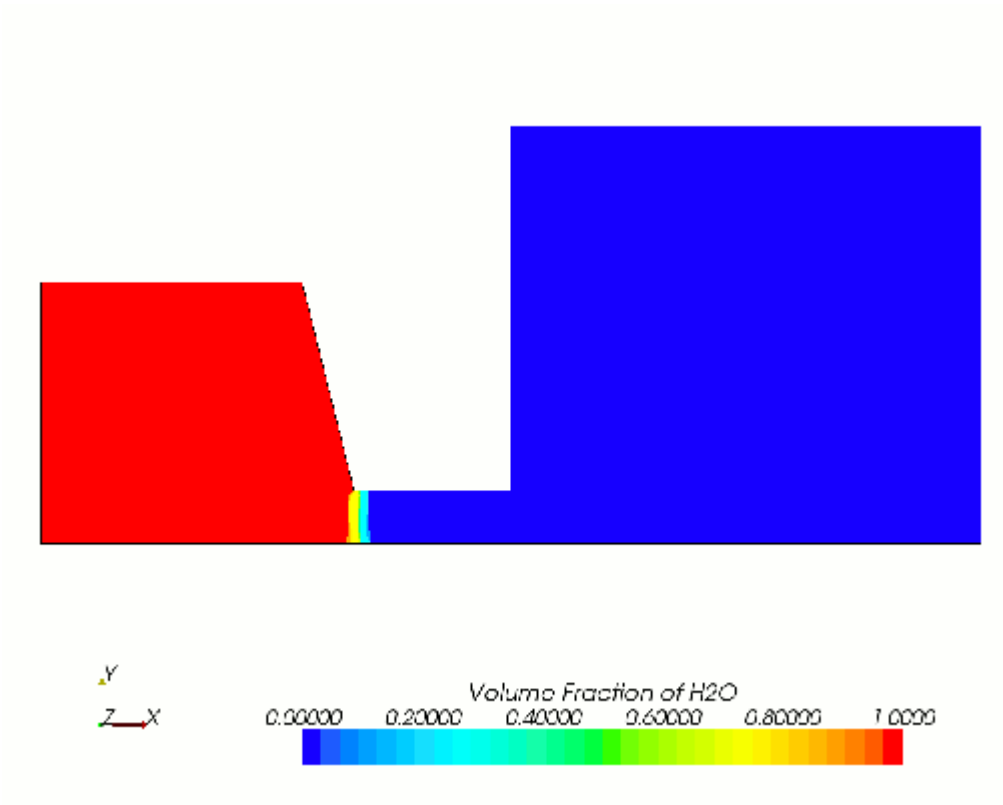


3. Save the simulation.

Animating the Results

You can use the animation feature in STAR-CCM+ to animate the results.

In the Visualizing and Initializing the Solution section, you instructed STAR-CCM+ to save the water volume fraction profile every 10 time-steps. The resulting series of `png` files can be used to produce an animation using an external package, such as the `animate` tool included with most Linux and Unix systems. An example of such an animation is shown below.



You can also modify the simulation and use the animation feature in STAR-CCM+. See the following sections in the tutorial Solution Recording and Playback: Vortex Shedding:

- Setting Up the Solution History
- Creating a Recorded Solution View
- Creating an Animation from the Solution View

Summary

This tutorial demonstrated how to set up a gravity-driven flow problem in STAR-CCM+.

This tutorial introduced the following STAR-CCM+ features:

- Defining models for multiphase flow problems.
- Defining the material properties that are required for the selected models.
- Setting initial conditions and reference values.
- Creating and applying field functions.
- Displaying the results as an animation.

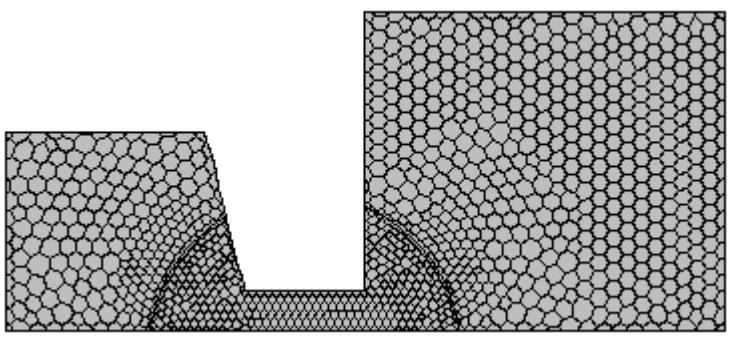
VOF: Capillary Effects

This tutorial demonstrates how to set up a capillary effects problem in STAR-CCM+. It simulates two-dimensional forced flow of liquid glycerine through a nozzle and into an air-filled chamber at atmospheric pressure.

The mesh that is used (shown below) is the same as in VOF: Gravity-Driven Flow except that, in this instance, the physical dimensions of the problem are scaled down by a factor of 1,000. Therefore, the nozzle width is about 1 mm. The boundary on the left of the problem geometry is an inlet with fluid velocity of 1 mm/s and the boundary on the right is at atmospheric pressure. The boundary at the bottom is a symmetry plane and all other boundaries are solid walls.

Initially, the left chamber is filled with liquid and the remainder of the solution domain is filled with air. For the given geometry and inlet velocity, the flow can be assumed to be laminar. Gravity acts in the positive x-direction and so helps to drive the flow through the nozzle. The shape of the free-surface that develops in the chamber downstream of the nozzle depends on the contact angle between liquid and wall. This contact angle is specified as 45°.

The Interface Momentum Dissipation phase interaction model is included in the simulation. This model stabilizes the solution significantly by reducing irregular velocities near the liquid-gas interface.



Contents:

Prerequisites
Importing the Mesh and Naming the Simulation
Converting to a Two-Dimensional Mesh
Scaling the Mesh
Selecting the Physics Models
Setting the Material Properties
Defining the Phase Interactions
Setting Initial Conditions and Reference Values
Setting Boundary Conditions
Setting Solver Parameters and Stopping Criteria
Visualizing and Initializing the Solution

Running the Simulation
Visualizing Results
Effects of Changing the Contact Angle
Summary

Prerequisites

The instructions in the VOF: Capillary Effects tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Importing the Mesh and Naming the Simulation

Launch STAR-CCM+, import the supplied geometry, and save the simulation.

A one-cell-thick three-dimensional polyhedral cell mesh has been predefined for this analysis.

To import the mesh and name the simulation:

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh**.
4. In the *Open* dialog, choose `doc/startutorialsdata/multiphaseFlow/data/gridfs.ccm` then click **Open**.

   STAR-CCM+ provides feedback on the import process, which takes a few seconds, in the *Output* window.
5. Save the simulation as `capillaryEffects.sim`.

Converting to a Two-Dimensional Mesh

The mesh can now be converted from three dimensions to two dimensions.

There are special requirements in STAR-CCM+ for three-dimensional meshes that are to be converted to two-dimensional:

- The grid must be aligned with the X-Y plane.
- The grid must have a boundary plane at the Z = 0 location.

The mesh that is imported for this tutorial was built with these requirements in mind. Were the grid not to conform to the above conditions, it would have been necessary to realign the region using the transformation and rotation facilities in STAR-CCM+.

To convert the mesh to two dimensions:

1. Select **Mesh** > **Convert to 2D...**.
2. In the *Convert Regions to 2D* dialog, activate **Delete 3D regions after conversion**, and click **OK**.

The mesh conversion takes place and the two-dimensional mesh is created.

Note: The mouse rotation option is suppressed for two-dimensional scenes.

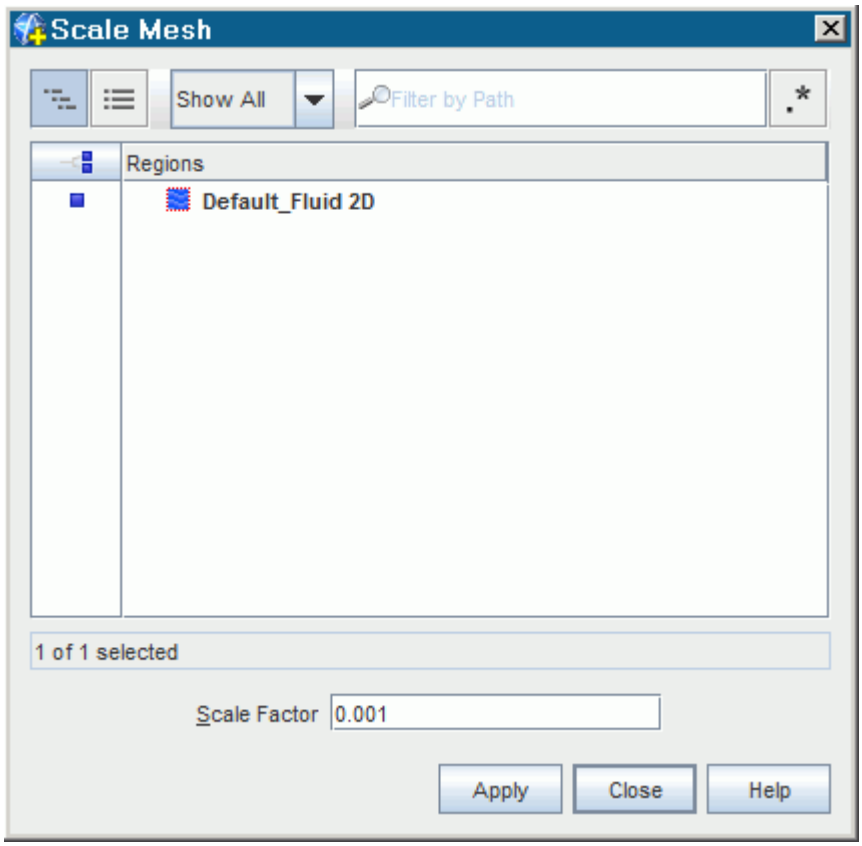
3. Right-click the **Physics 1** continuum node and select **Delete**.
4. Click **Yes** in the confirmation dialog.

Scaling the Mesh

The original mesh was not built to the correct scale and therefore requires scaling down by a factor of 1000.

To scale the mesh:

1. Select **Mesh** > **Scale Mesh** from the menu bar.
2. In the *Scale Mesh* dialog, make sure that the **Default_Fluid 2D** region is selected.



3. Enter a value of `0.001` for the scale factor.
4. Click **Apply** to scale the domain, and then click **Close**.

   After clicking **Apply**, the mesh region reduces in size.

To restore the previous viewing distance for the scaled domain in the display:

5. Click the (**Reset View**) button in the visualization toolbar.

To verify that scaling has been applied, you can perform the mesh diagnostics check and review the output values.

Selecting the Physics Models

Physics models define the primary variables of the simulation, including pressure, temperature, velocity, and what mathematical formulation is used to generate the solution.

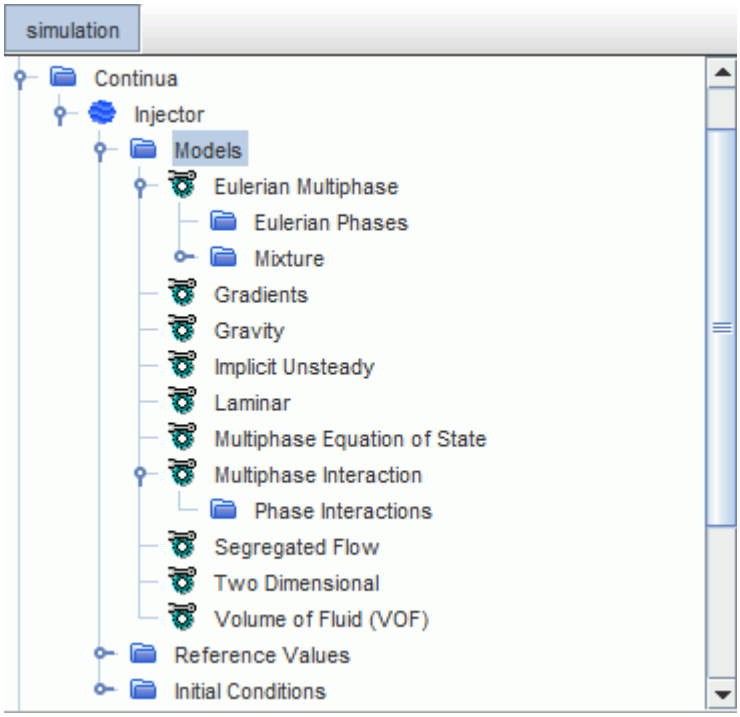
In this simulation, the flow is laminar, and a gravity force is applied in the +x direction. As the problem also involves multiphase flow, two fluids (air and glycerine) are required for the analysis. However, since these fluids occupy the same domain, only one continuum and one region are required to set up the simulation. By default, a continuum that is called **Physics 1 2D** is created when the mesh is converted to two-dimensional.

To select the physics models:

1. Rename the **Physics 1 2D** node to **Injector**.
2. For the physics continuum, **Continua** > **Injector**, select the following models in order:

Group Box	Model
Enabled Models	**Two Dimensional** (Pre-selected)
Time	**Implicit Unsteady**
Material	**Eulerian Multiphase** **Multiphase Interaction** (selected automatically)
Eulerian Multiphase Model	**Volume of Fluid (VOF)** **Multiphase Equation of State** (selected automatically) **Segregated Flow** (selected automatically) **Gradients** (selected automatically)
Viscous Regime	**Laminar**
Optional Models	**Gravity**

3. Click **Close**.
4. To review the models, open the **Injector** > **Models** node.



5. Save the simulation.

Setting the Material Properties

Define the phases and set the appropriate material properties.

Define the materials corresponding to each of the mixture components under the **Eulerian Phases** node.

Note: In multiphase flow problems that include surface tension effects, define the densest fluid first because contact angle values at the wall are always specified relative to the first-defined fluid.

To define the phases and set the material properties:

1. In the **Injector** continuum, right-click the **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and create a new phase.
2. Rename the **Phase 1** node to **C3H8O3**.
3. For the **C3H8O3** phase, select the following models:

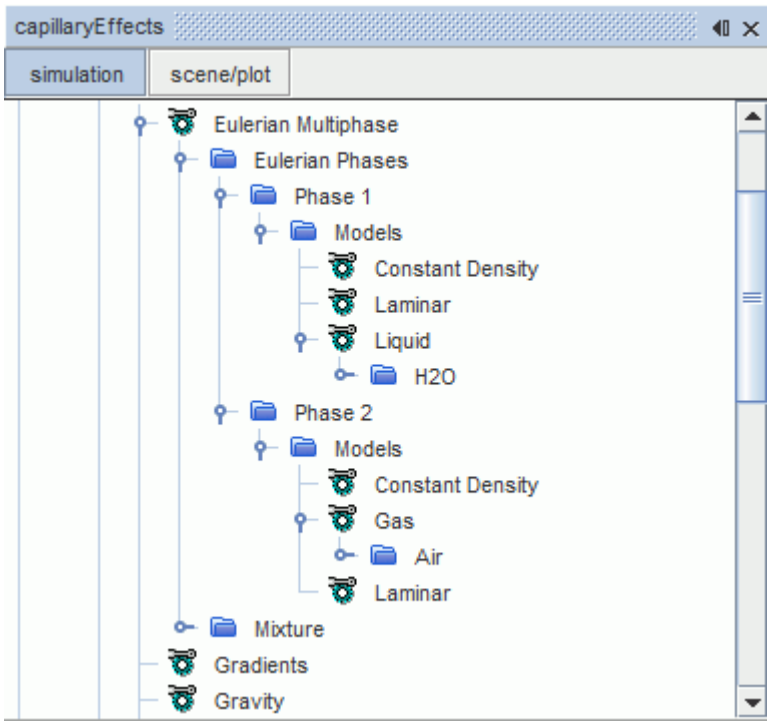
Group Box	Model
Material	**Liquid**
Equation of State	**Constant Density**

4. Click **Close**.
5. Create a second phase and rename it to **Air**.
6. For the **Air** phase, select the following models:

Group Box	Model
Material	**Gas**
Equation of State	**Constant Density**

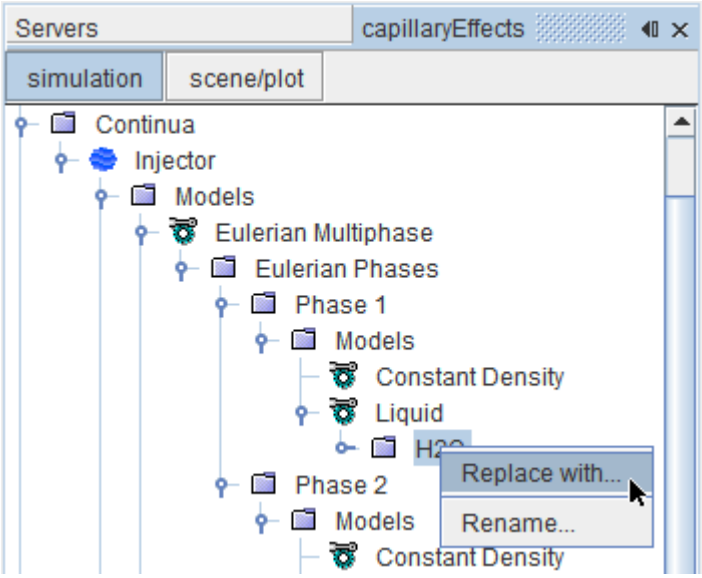
7. Click **Close**.

   The **Eulerian Multiphase** node contains the phase model objects as shown below:

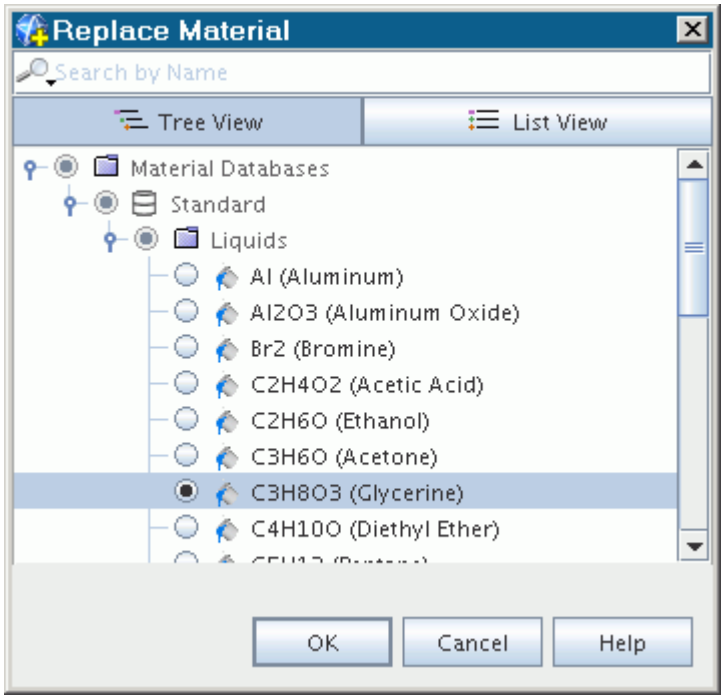


The default selection of air is appropriate for this exercise, but water must be replaced with glycerine from the STAR-CCM+ material database.

8. Right-click the **C3H8O3** > **Models** > **Liquid** > **H2O** node and select **Replace with**.

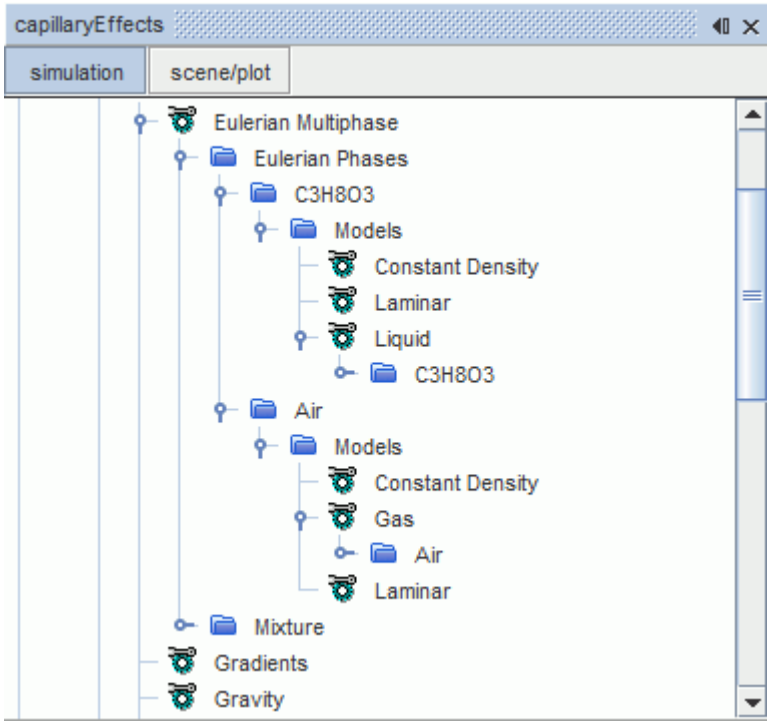


9. In the *Replace Material* dialog, expand the **Material Databases** > **Standard** > **Liquids** node.
10. Activate **C3H8O3 (Glycerine)**.



11. Click **OK** to replace the material.

    The phase model objects appear as shown below.



12. Save the simulation.

Defining the Phase Interactions

Using the **Multiphase Interaction** model, you can define the interaction between the air and liquid phases.

In this tutorial, you are interested in the surface tension of glycerine. Therefore, use the Surface Tension Force and VOF multiphase interaction models:

To define the phase interactions:

1. In the **Injector** continuum, right-click the **Models** > **Multiphase Interaction** > **Phase Interactions** node and create a new phase interaction.
2. For the **Phase Interaction 1** phase interaction, select the following models in order:

Group Box	Model
Phase Interaction Topology	**VOF-VOF Phase Interaction**
Optional Models	**Interface Momentum Dissipation** **Surface Tension Force** **Multiphase Material** (selected automatically)

3. Click **Close**.

The value of the surface tension coefficient is changed for the phase interaction. Set the primary and secondary phases for the phase interaction.

4. Edit the **Phase Interaction 1** > **Models** node and set the following properties:

Node	Property	Setting
Multiphase Material > **Multiphase Material** > **Material Properties** > **Surface Tension** > **Constant**	**Value**	`0.059688 N/m`
VOF-VOF Phase Interaction	**Primary Phase**	**C3H8O3**
	Secondary Phase	**Air**

5. Save the simulation.

Setting Initial Conditions and Reference Values

Set the initial conditions and reference values for the physics continuum.

The initial condition for the spatial distribution of the two fluids in the continuum is that you only have glycerine in the left-hand chamber and only air in the right-hand chamber and channel. A convenient way of specifying such a distribution is by creating and using a field function.

The direction and magnitude of the gravity vector is set using the Reference Values node. In this case, a gravity force is applied in the positive x-direction.

To set the initial conditions and reference values:

Define the field function to use for specifying the initial condition for the spatial distribution of the two fluids in the continuum:

1. Right-click the **Tools** > **Field Functions** node and select **New** > **Scalar** from the pop-up menu.
2. Rename the **User Field Function 1** to `Initial Distribution`.
3. Select the **Initial Distribution** node and set the following properties:

Property	Setting
Function Name	`Initial Distribution`
Definition	`($$Position[0]>=-0.001)?0:1`

Set the initial conditions and reference values for the **Injector** continuum.

4. Edit the **Initial Conditions** node and set the following properties:

Node	Property	Setting
Volume Fraction	**Method**	**Composite**
└ **Composite** > **C3H8O3**	**Method**	**Field Function**
	Scalar Function	**Initial Distribution**

It is not necessary to specify a similar distribution for the Air node because any user-specified initial condition for the volume fraction of the last-defined phase in multiphase flows is always ignored. Instead,

this condition is obtained by subtracting the sum of the volume fraction distributions of the other phases from 1.0.

5. Edit the **Reference Values** node and set the following properties:

Node	Property	Setting
Gravity	**Value**	`[9.81,0.0] m/s^2`

6. Save the simulation.

Setting Boundary Conditions

Define the boundary types and specify the appropriate property values.

To set the boundary conditions:

1. Rename the **Default_Fluid 2D** node to **Fluid**.

Set the boundary types.

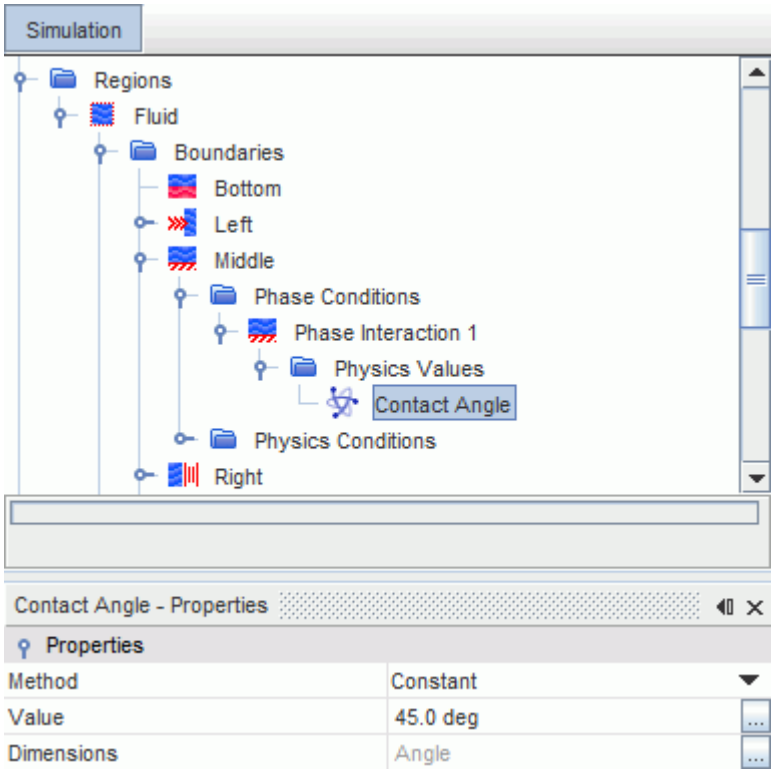
2. Edit the **Fluid** > **Boundaries** node and set the boundary types as follows:

Boundary	Type
Bottom	**Symmetry Plane**
Left	**Velocity Inlet**
Middle	**Wall**
Right	**Pressure Outlet**
TopRight	**Wall**
TopLeft	**Wall**

The only condition that is specified for the wall boundaries is the contact angle of the glycerine-air interface.

3. Edit the **Middle** > **Phase Conditions** > **Phase Interaction 1** > **Physics Values** node and set the following property:

Node	Property	Setting
Contact Angle	**Value**	`45 deg`



4. Repeat the previous step for the other two wall boundaries: **TopLeft** and **TopRight**.

Set the boundary conditions for the inlet and pressure boundaries.

5. Edit the **Left** > **Physics Values** node and set the following properties:

Node	Property	Setting
Velocity Magnitude	**Value**	`0.001 m/s`
Volume Fraction	**Value**	`[1.0, 0.0]`

The **Volume Fraction** setting enforces the condition that only glycerine enters the solution domain through that boundary.

6. Edit the **Right** > **Physics Values** node and set the following property:

Node	Property	Setting
Volume Fraction	**Value**	`[0.0, 1.0]`

This setting enforces the condition that only air enters the solution domain through that boundary.

7. Save the simulation.

Setting Solver Parameters and Stopping Criteria

As you are solving an unsteady problem, it is necessary to specify the time-step size and the elapsed simulation time.

Run this calculation for 2.0 s with a time-step size of 0.001 s, so 2,000 time-steps are required.

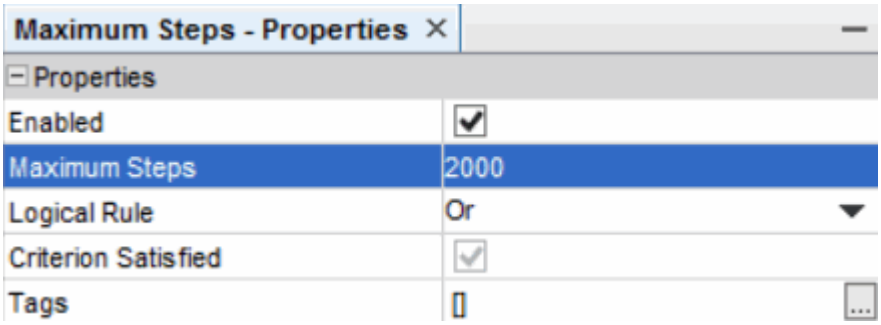
To set the solver parameters and the stopping criteria:

1. Edit the **Solvers** node and set the following property:

Node	Property	Setting
Implicit Unsteady	**Time-Step**	`0.001 s`

2. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Steps	**Enabled**	Activated
	Maximum Steps	`2000`
Maximum Physical Time	**Enabled**	Deactivate



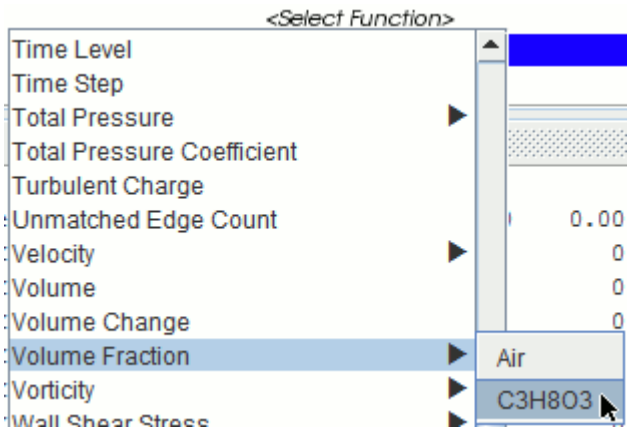
3. Save the simulation.

Visualizing and Initializing the Solution

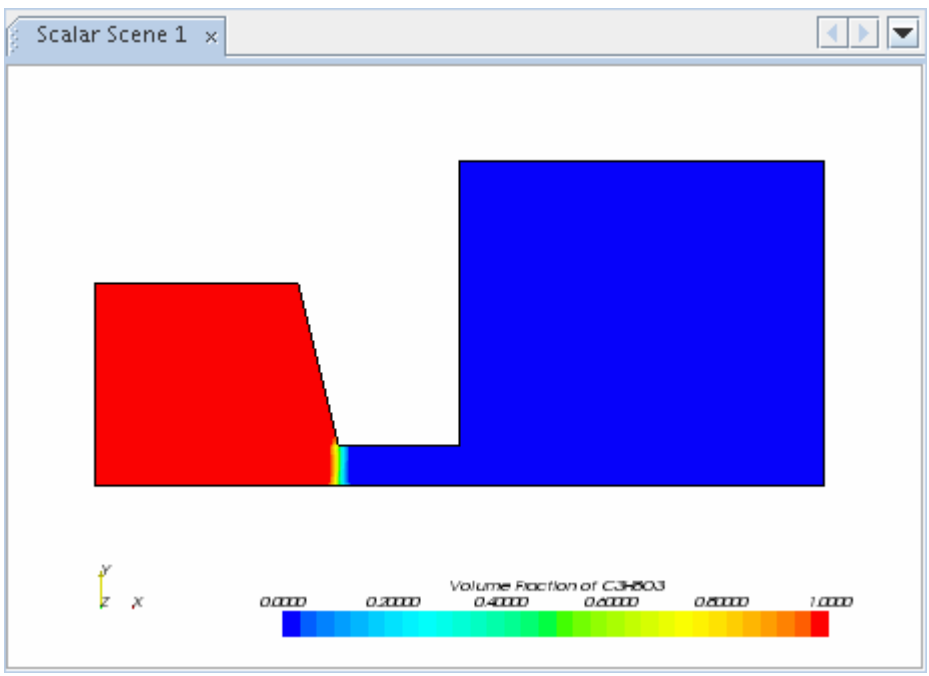
Use a scalar scene to view the air and glycerine distributions throughout the run.

To set up the scalar scene:

1. Right-click the **Scenes** node and then select **New Scene** > **Scalar**.
2. In the *Graphics* window, right-click the scalar bar and select **Volume Fraction** > **C3H8O3**.



3. Select the **Scenes** > **Scalar Scene 1** > **Displayers** > **Scalar 1** node.
4. Change the *Contour Style* property setting to **Smooth Filled**.
5. Click (**Initialize Solution**) in the top toolbar or use the **Solution** > **Initialize Solution** menu option.
6. Activate the *Scalar Scene 1* display to look at the initialization results.



As expected, at the start of the run the left chamber is entirely filled with glycerine whereas the right chamber and the connecting channel are entirely filled with air. A small region in which both fluids are apparently present is visible at the interface between the two, but this effect is due to the coarseness of the mesh.

7. Save the simulation.

Running the Simulation

Preparation of the simulation is now complete, and the simulation can be run.

To run the simulation:

1. Click (**Run**) in the *Solution* toolbar.

   The solution progress is displayed in the *Output* window. The *Residuals* display is created automatically in the *Graphics* window and shows the progress of the solvers.

   You can click the *Scalar Scene 1* tab at the top of the *Graphics* window to view the simulation progress.

   During the run, you can stop the process by clicking (**Stop**) on the toolbar. If you do stop the simulation, you can click (**Run**) to resume it.

   If left alone, the simulation continues until 2000 time-steps are complete. The numbers that are displayed in the *Output* window represent the inner iterations of the solver, not time-steps. As there are 5 inner iterations per time-step, you expect the run to perform 10,000 inner iterations.

   When the last iteration is complete, the following message is displayed in the *Output* window:

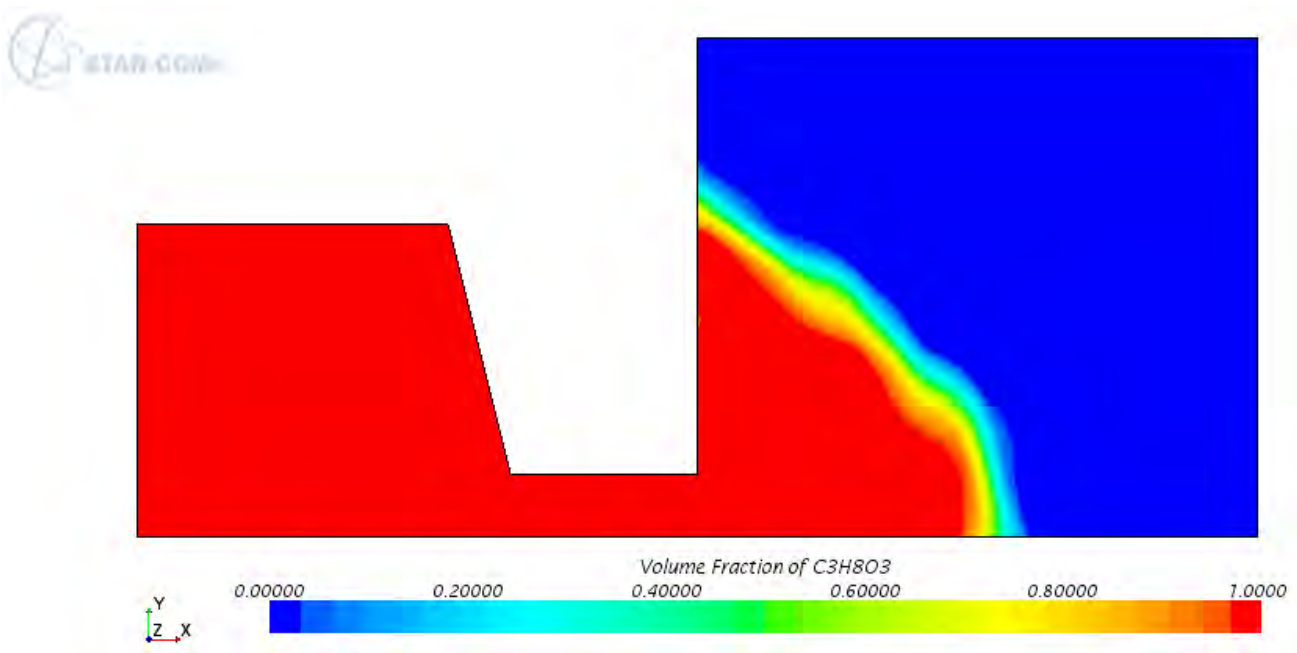
   ```
   Stopping criterion Maximum Steps satisfied.
   ```

2. When the simulation has finished running, save it.

Visualizing Results

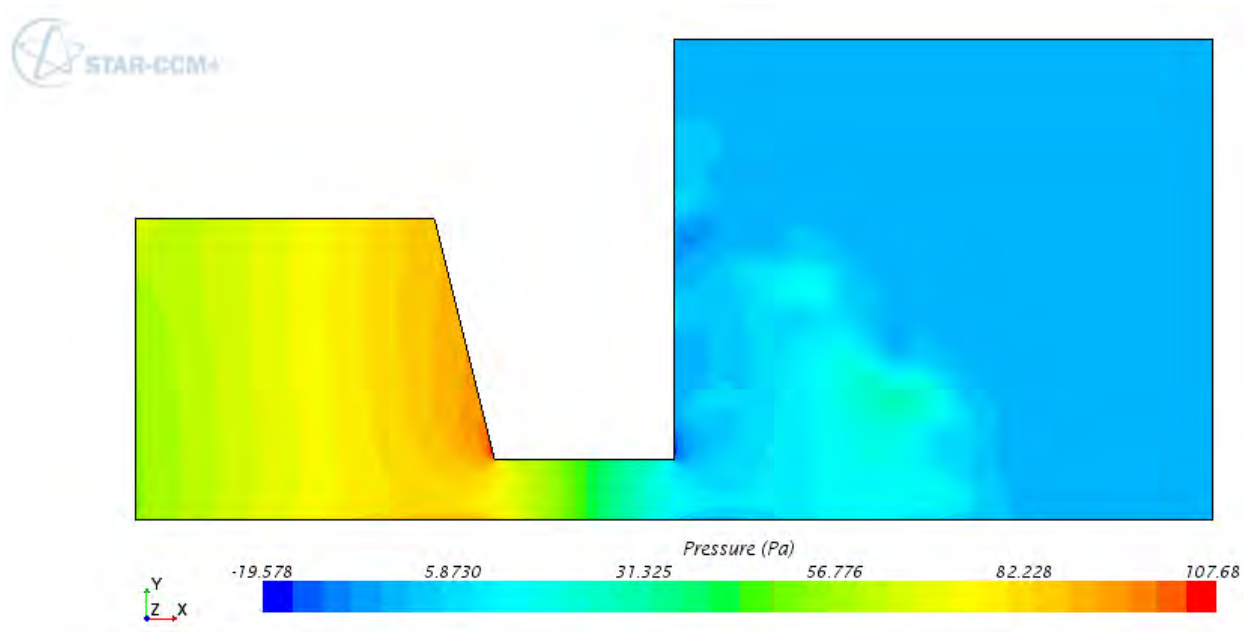
View the results of the simulation.

The *Scalar Scene 1* display shows the glycerine volume fraction profile at the end of the 2.0 s run. The observable contact angle is indeed close to 45º.



To view the results of the simulation:

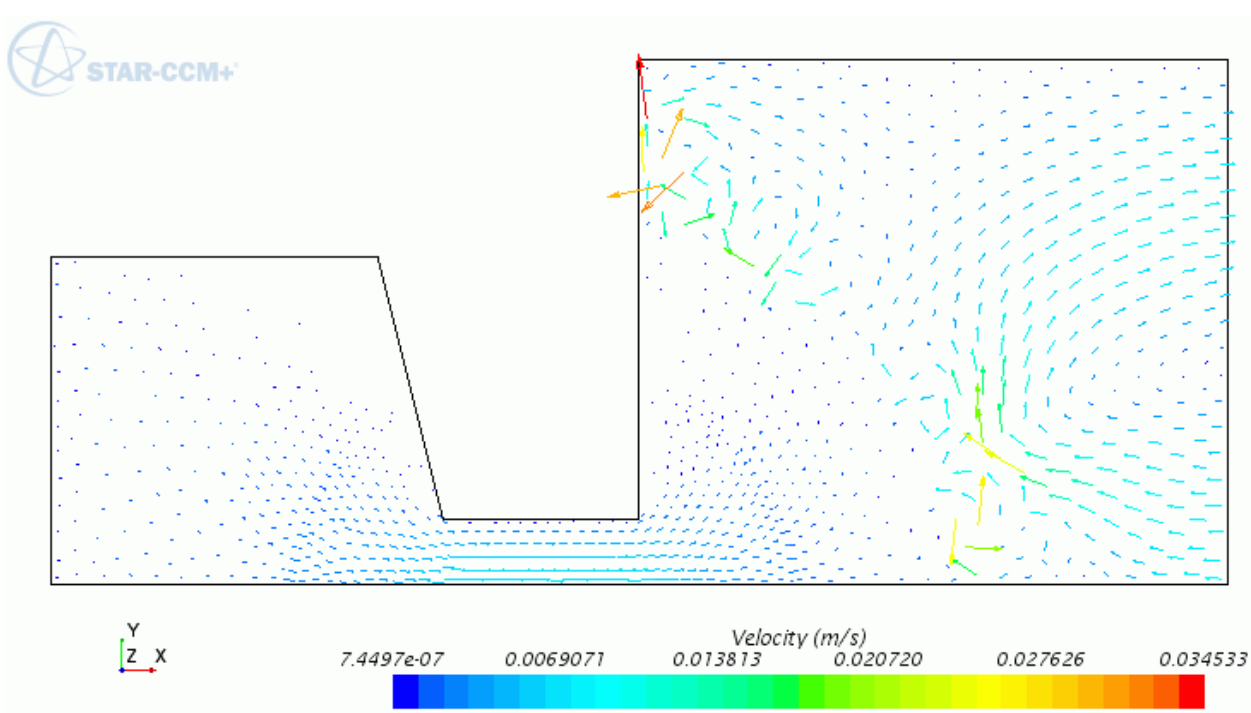
1. Right-click the scalar bar in the display.
2. Select **Pressure** from the pop-up menu to display the pressure profile.



To display velocity vectors:

3. Right-click the **Scenes** node and select **New Scene** > **Vector**.

   The vector scene appears in the *Graphics* window.



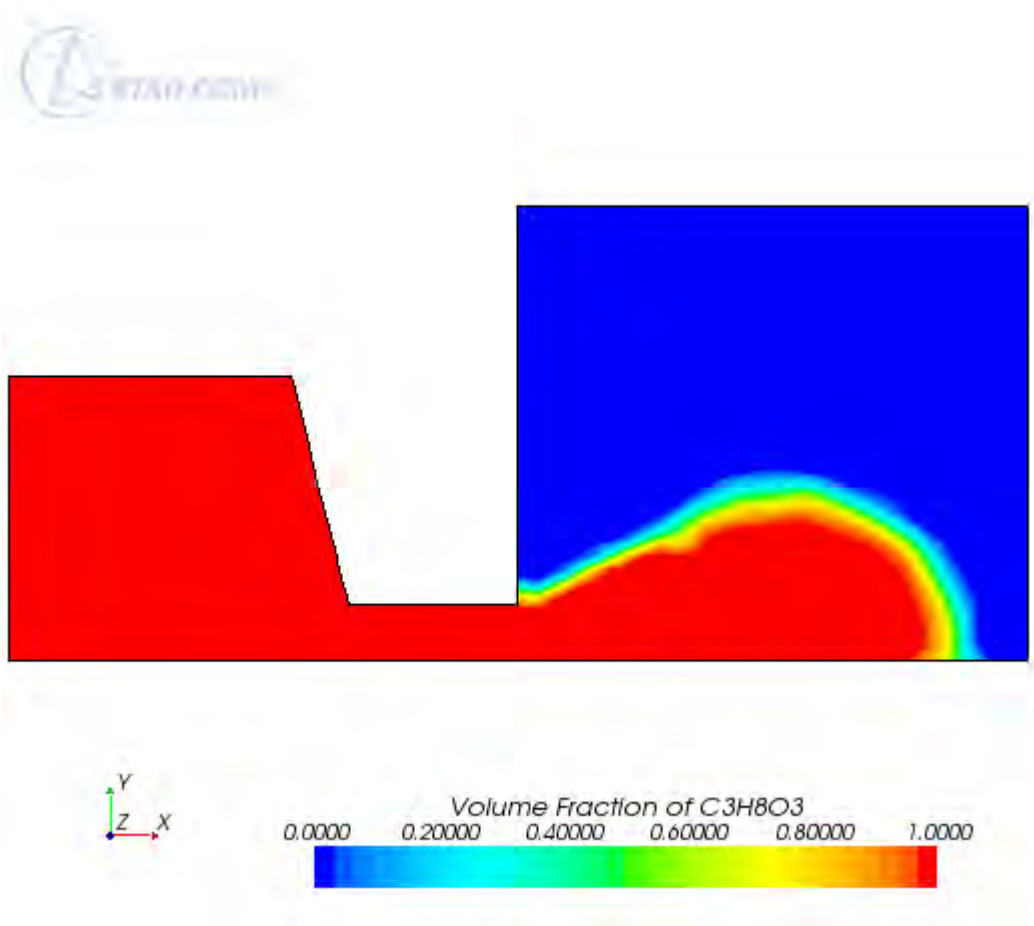
The velocity vector plot shows high air velocities close to the free surface. These numerical inaccuracies are known as parasitic currents. These currents arise because the surface tension and pressure forces are much larger than all other terms in the momentum equations, and their balance on an irregular grid is difficult to achieve numerically due to the discontinuous variation of pressure across the free surface and a large discretization error that is associated with it. Parasitic currents become appreciable when the problem size is small and fluid velocity and viscosity are low. For flows where diffusion and convection forces are of a similar magnitude to surface tension forces, these problems are not so pronounced. Since artificial velocities are generated only within the air, their effect on the liquid flow (which is usually what you are trying to predict) is small.

4. Save the simulation.

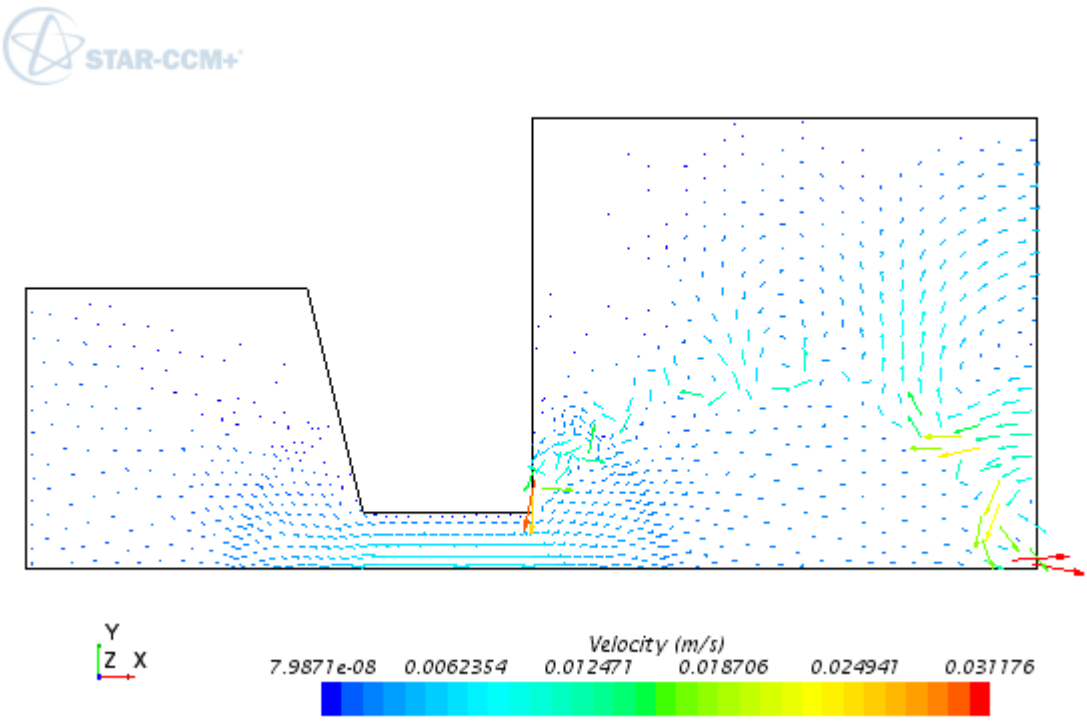
Effects of Changing the Contact Angle

Consider the effects of changing the contact angle.

The contours of glycerine volume fraction shown below demonstrate the effect of changing the contact angle. In this case, the contact angle at all wall boundaries was changed from 45° to 135°. All other modeling options and material properties were kept the same and the analysis is run for a physical time of 2.0 s, as before.



A vector scene similar to the one shown below is also displayed.



Summary

This tutorial demonstrated how to set up a capillary effects problem in STAR-CCM+.

This tutorial has introduced the following STAR-CCM+ features:

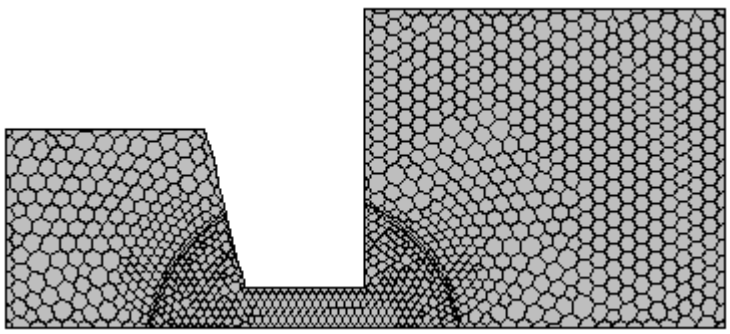
- Defining models for multiphase flow problems.
- Defining the material properties that are required for the selected models.
- Setting initial conditions and reference values.
- Creating and applying field functions.
- Defining boundary conditions.
- Setting solver parameters for an unsteady run.
- Initializing and running the solver for a given number of time-steps.
- Analyzing results using the built-in visualization facilities.
- Examining the effect of changing the contact angle value.

VOF: Cavitation

This tutorial demonstrates how to set up a cavitation problem in STAR-CCM+. It simulates a two-dimensional forced flow of water through a nozzle and into an air-filled chamber at atmospheric pressure.

The mesh that is used (shown below) is the same as in VOF: Gravity-Driven Flow except that, in this instance, the physical dimensions of the problem are scaled down by a factor of 1,000. Therefore, the nozzle width is about 1 mm. The boundary on the left of the problem geometry has a prescribed pressure of 5×10^7 Pa and the boundary on the right is at atmospheric pressure. The boundary at the bottom is a symmetry plane and all other boundaries are solid walls.

Initially, the left chamber is filled with water and the remainder of the solution domain is filled with air. Turbulence is modeled using the default K-Epsilon model. The sharp corner at the entrance to the narrow nozzle combined with the flow acceleration produces a low-pressure region in which cavitation takes place.



Contents:

Prerequisites
Importing the Mesh and Naming the Simulation
Converting to a Two-Dimensional Mesh
Scaling the Mesh
Selecting the Physics Models
Setting the Material Properties
Defining the Phase Interactions
Setting the Cavitation Model Parameters
Setting Initial Conditions
Setting Boundary Conditions
Setting Solver Parameters and Stopping Criteria
Visualizing and Initializing the Solution
Running the Simulation
Visualizing Results
Summary

Prerequisites

The instructions in the VOF: Cavitation tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Importing the Mesh and Naming the Simulation

Launch STAR-CCM+, import the supplied geometry, and save the simulation.

A one-cell-thick three-dimensional polyhedral cell mesh has been predefined for this analysis.

To import the mesh and name the simulation:

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh**.
4. In the *Open* dialog, choose `doc/startutorialsdata/multiphaseFlow/data/gridfs.ccm` then click **Open**.

   STAR-CCM+ provides feedback on the import process, which takes a few seconds, in the *Output* window. A geometry scene is created in the *Graphics* window.
5. Save the simulation as `cavitation.sim`.

Converting to a Two-Dimensional Mesh

Convert the three-dimensional mesh to two dimensions.

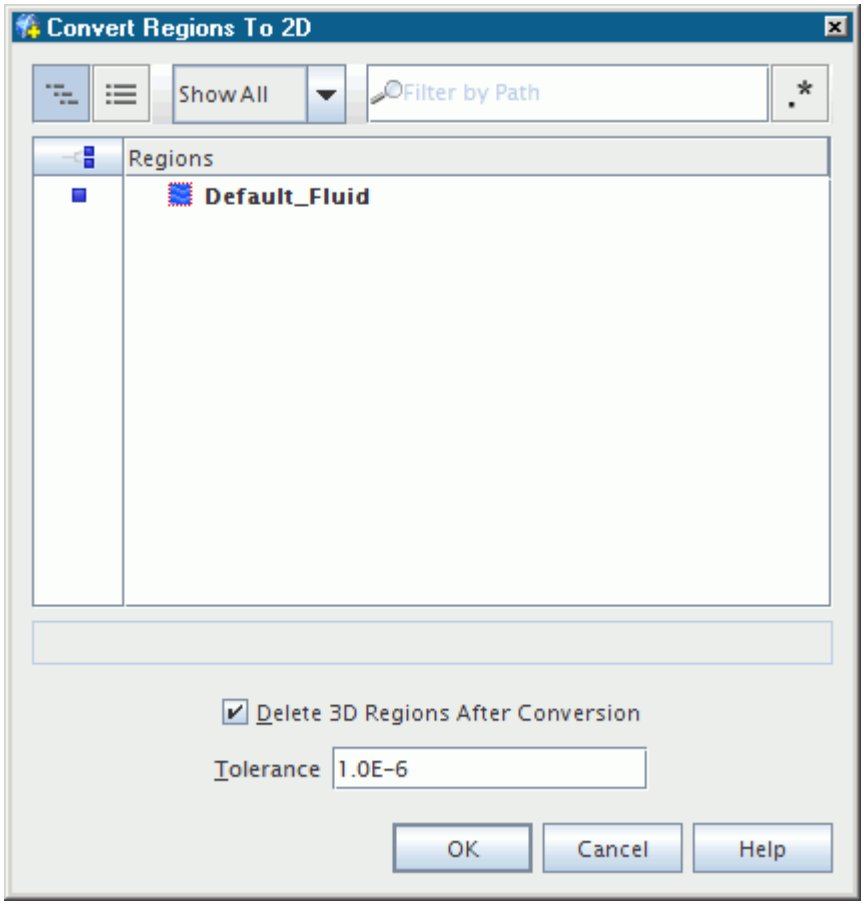
There are special requirements in STAR-CCM+ for converting three-dimensional meshes to two-dimensional meshes:

- The grid must be aligned with the X-Y plane.
- The grid must have a boundary plane at the Z = 0 location.

The mesh that is imported for this tutorial is built with these requirements in mind. Had the grid not met them, it would have been necessary to realign the region using the mesh transformation and rotation capabilities in STAR-CCM+.

To convert the mesh region to two dimensions:

1. Select **Mesh** > **Convert to 2D...**.
2. In the *Convert Regions to 2D* dialog that appears, make sure that **Delete 3D regions after conversion** is activated, and click **OK**.



Once you click **OK**, the mesh conversion takes place and the new two-dimensional mesh is shown, viewed from the z-direction, in the *Geometry Scene 1* display.

Note: The mouse rotation option is suppressed for two-dimensional scenes.

Delete the three-dimensional physics continuum:

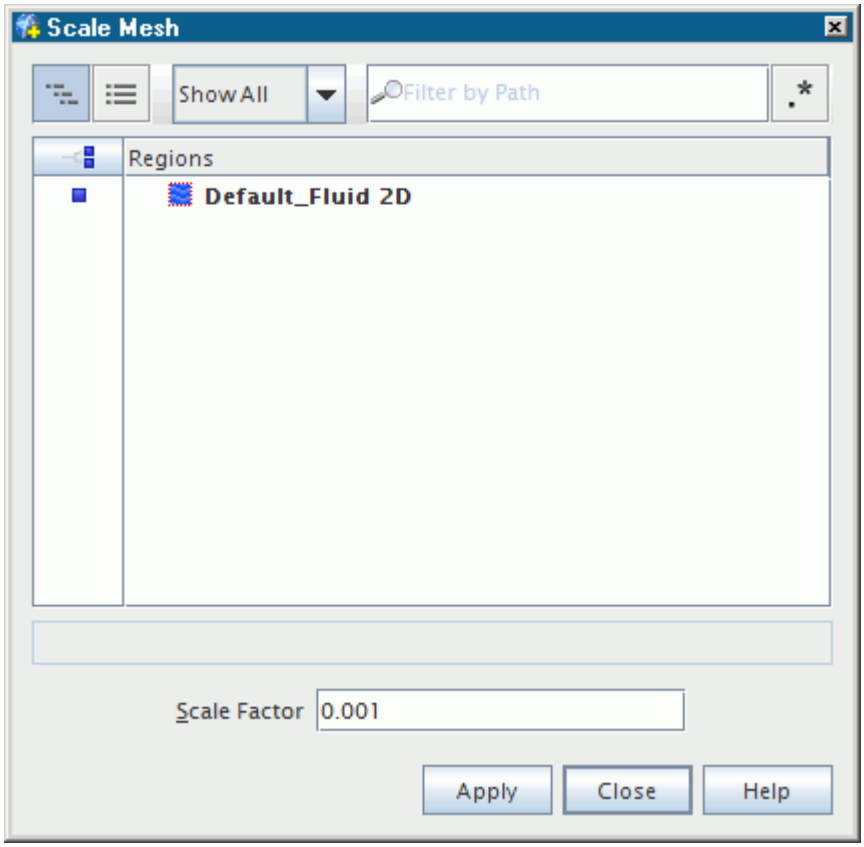
3. Right-click the **Physics 1** continuum node and select **Delete**.
4. Click **Yes** in the confirmation dialog.

Scaling the Mesh

The original mesh is not built to the correct scale and therefore requires scaling down by a factor of 1000.

To scale the mesh:

1. Select **Mesh** > **Scale Mesh** from the menu bar.
2. In the *Scale Mesh* dialog, make sure that the **Default_Fluid 2D** region is selected.
3. Enter a value of `0.001` for the scale factor.



4. Click **Apply** to scale the domain, and then click **Close**.

   The mesh region reduces in size.

To restore the previous viewing distance for the scaled domain in the display:

5. Click (**Reset View**) in the visualization toolbar.

To verify that scaling has in fact been applied, you can perform the mesh diagnostics check and review the output values.

Selecting the Physics Models

Physics models define the primary variables of the simulation, including pressure, temperature, velocity, and what mathematical formulation is used to generate the solution.

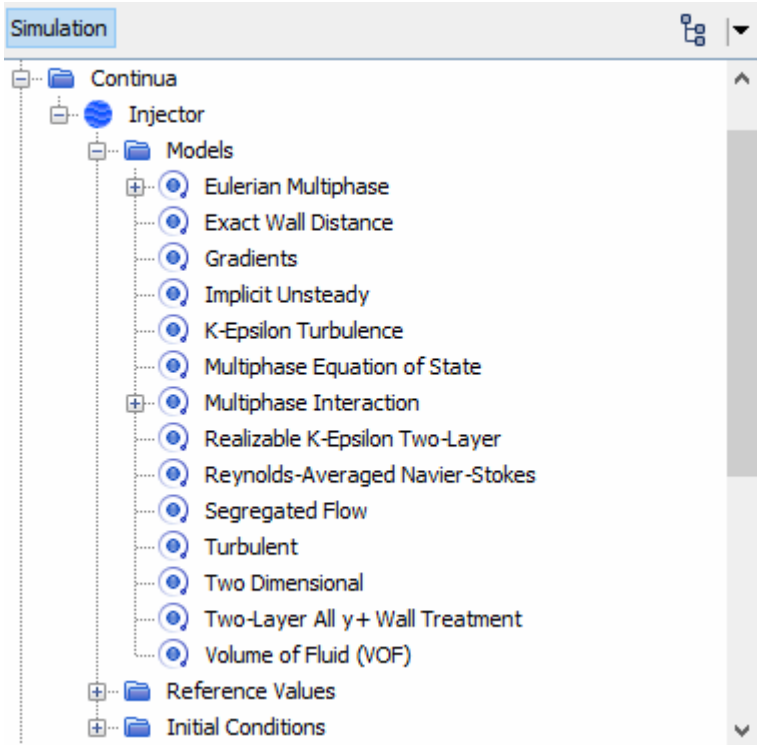
In this simulation, the flow is turbulent and the problem involves multiphase flow and cavitation. Three fluids (air, water, and water vapor) are required for the analysis. However, since these fluids occupy the same domain, only one continuum and one region are required to set up the simulation. By default, a continuum that is called **Physics 1 2D** is created when the mesh is converted to two-dimensional.

To select the physics models:

1. Rename the **Physics 1 2D** continuum node to **Injector**.
2. For the physics continuum, **Continua** > **Injector**, select the following models in order:

Group Box	Model
Enabled Models	**Two Dimensional** (Pre-selected)
Time	**Implicit Unsteady**
Material	**Eulerian Multiphase** **Multiphase Interaction** (selected automatically)
Eulerian Multiphase Model	**Volume of Fluid (VOF)** **Gradients** (selected automatically) **Multiphase Equation of State** (selected automatically) **Segregated Flow** (selected automatically)
Viscous Regime	**Turbulent** **Reynolds-Averaged Navier-Stokes** (selected automatically)
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence** **Realizable K-Epsilon Two-Layer** (selected automatically) **Exact Wall Distance** (selected automatically) **Two-Layer All y+ Wall Treatment** (selected automatically)

3. Click **Close**.
4. To review the models, open the **Injector** > **Models** node.



5. Save the simulation.

Setting the Material Properties

You define the phases and set the appropriate material properties.

Define the materials corresponding to each of the mixture components under the **Eulerian Phases** node.

To define the phases and set the material properties:

1. Right-click the **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and create a new phase.
2. Rename the **Phase 1** node to **H2O**.
3. For the **H2O** phase, select the following models:

Group Box	Model
Material	**Liquid**
Equation of State	**Constant Density**

4. Click **Close**.
5. Create a second phase and rename it to **Air**.
6. For the **Air** phase, select the following models:

Group Box	Model
Material	**Gas**
Equation of State	**Constant Density**

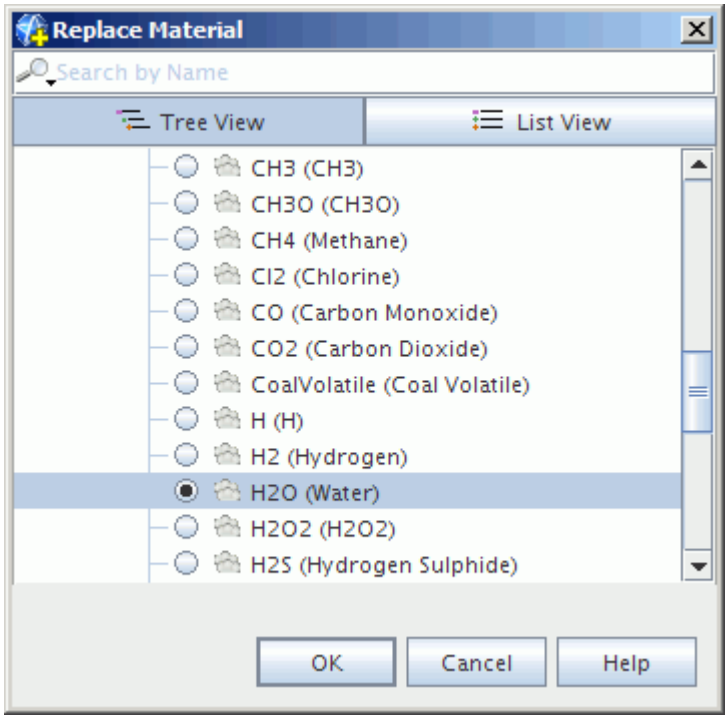
7. Click **Close**.
8. Create a third phase and rename it to **H2O (G)**.
9. For the **H2O (G)** phase, select the following models:

Group Box	Model
Material	**Gas**
Equation of State	**Constant Density**

10. Click **Close**.

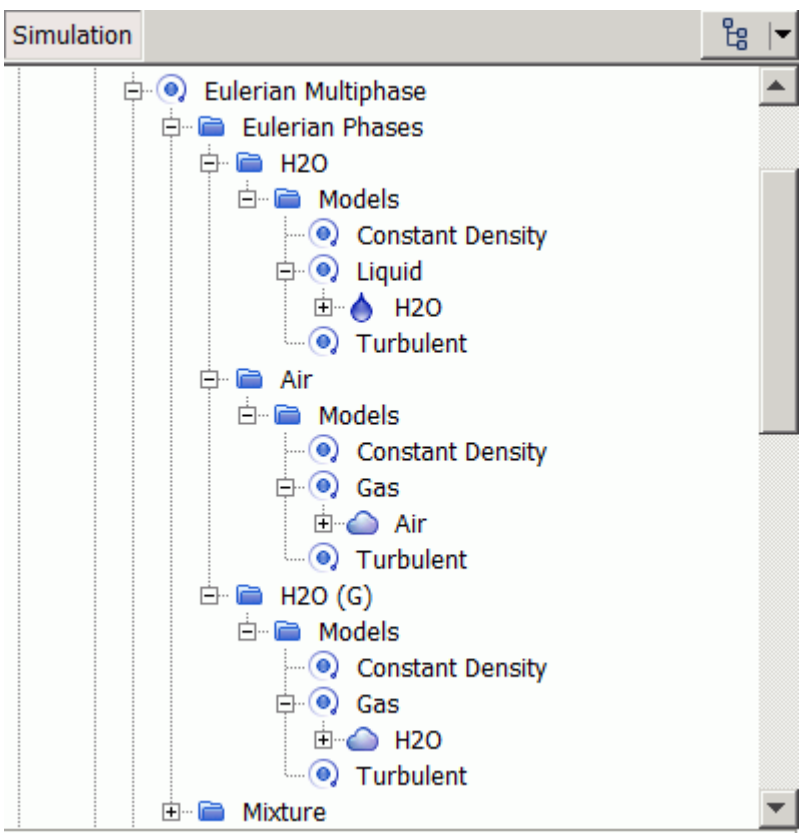
If you were setting up a standard free-surface problem, only two phases would be required. However, because this problem involves cavitation, it is necessary to define a third phase for the product of cavitation, that is, water vapor. Hence, the air in the third phase is replaced with water vapor from the STAR-CCM+ material database.

11. Right-click the **H2O (G)** > **Models** > **Gas** > **Air** node and select **Replace with**.
12. In the *Replace Material* dialog, expand the **Material Databases** > **Standard** > **Gases** node.
13. Activate **H2O (Water)**.



14. Click **OK** to replace the material.

    The phase model objects appear as shown below.



The default material properties for the three phases are suitable for this problem.

15. Save the simulation.

Defining the Phase Interactions

Using the **Multiphase Interaction** model, you can define the interaction between the gas and liquid phases.

To model cavitation, use the **VOF Multiphase Interaction** model and the **Schnerr-Sauer** cavitation model.

To define the phase interactions:

1. In the **Injector** continuum, right-click the **Models** > **Multiphase Interaction** > **Phase Interactions** node and create a new phase interaction.
2. For the **Phase Interaction 1** phase interaction, select the following models in order:

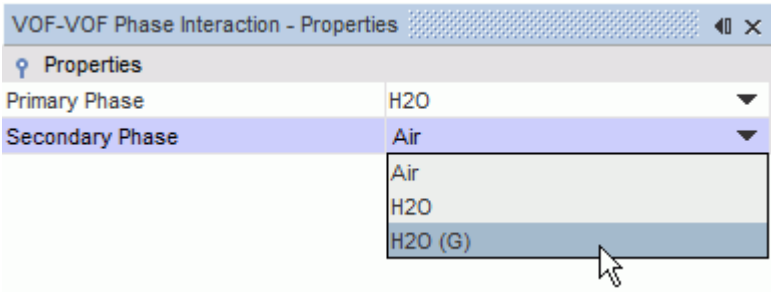
Group Box	Model
Phase Interaction Topology	**VOF-VOF Phase Interaction**
Optional Models	**Multiphase Material**

3. Click **Close**.

Before you can select the cavitation model, you set the primary and secondary phases for the phase interaction.

4. Edit the **Phase Interaction 1** > **Models** node and set the following properties:

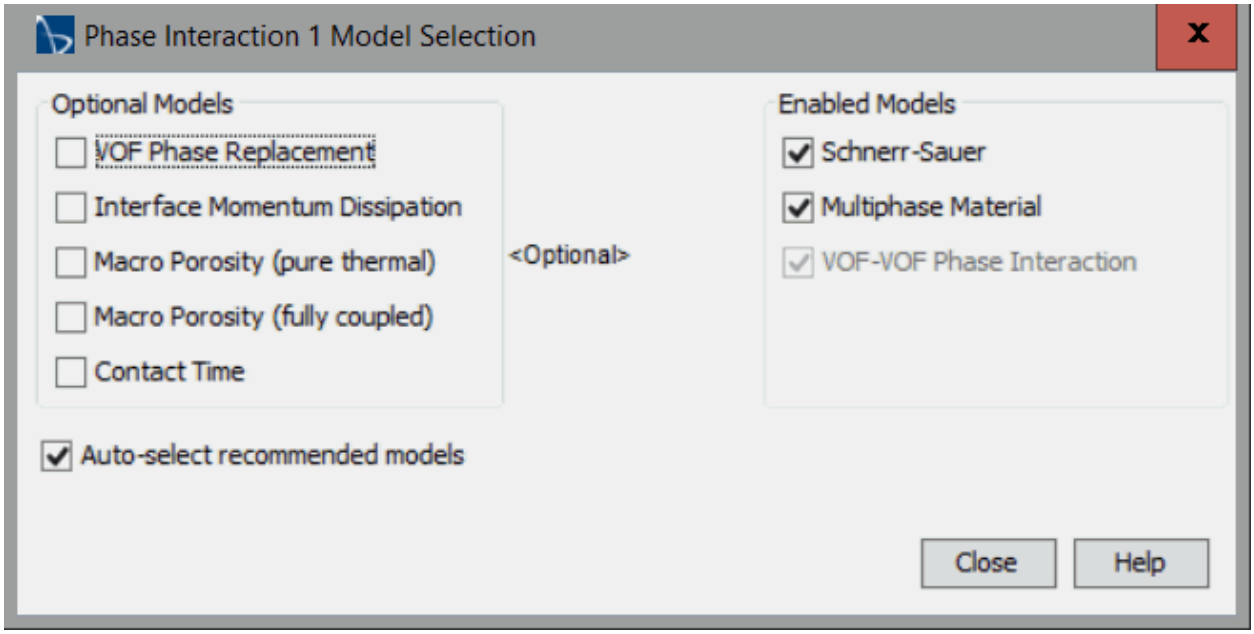
Node	Property	Setting
VOF-VOF Phase Interaction	**Primary Phase**	**H2O**
	Secondary Phase	**H2O(G)**



You can now select the cavitation model.

5. For the **Phase Interaction 1** phase interaction, select the following model:

Group Box	Model
Cavitation Models	**Schnerr-Sauer**



6. Click **Close**.
7. Save the simulation.

Setting the Cavitation Model Parameters

Define the appropriate parameters for the Schnerr-Sauer cavitation model. The saturation pressure is set on the relevant material property node for the liquid phase.

To set the Schnerr-Sauer cavitation model parameters:

1. For the physics continuum, **Continua** > **Injector**, select the **Models** > **Eulerian Multiphase** > **Eulerian Phases** > **H2O** > **Models** > **Liquid** > **H2O** > **Material Properties** > **Saturation Pressure** > **Constant** node and set *Value* to `2338.0` Pa.
2. Save the simulation.

Setting Initial Conditions

You set the initial conditions for the physics continuum. The initial condition for the spatial distribution of the two primary fluids in the continuum is as follows: you have only water in the left-hand chamber and only air in the right-hand chamber and channel. A convenient way of specifying such a distribution is by creating and using field functions.

To set the initial conditions:

Define the field function to use for specifying the initial condition for the spatial distribution of the two fluids in the continuum:

1. Right-click the **Tools** > **Field Functions** node and select **New** > **Scalar** from the pop-up menu.
2. Rename the **User Field Function 1** to `Initial Distribution (Air)`.

3. Select the **Initial Distribution (Air)** node and set the following properties:

Property	Setting
Function Name	`Initial Distribution (Air)`
Definition	`($${Position}[0] >= -0.001) ? 1 : 0`

Set the initial conditions for the **Injector** continuum.

4. Edit the **Initial Conditions** node and set the following properties:

Node	Property	Setting
Volume Fraction	**Method**	**Composite**
└ **Composite** > **Air**	**Method**	**Field Function**
	Scalar Function	**Initial Distribution (Air)**

5. Repeat the steps that are outlined above to specify the initial distribution of the H2O phase using a second field function named `Initial Distribution (H2O)` with the following definition:

   ```
   ($${Position}[0] >= -0.001) ? 0 : 1
   ```

   Since these two field functions are the inverse of each other, the water vapor concentration is zero everywhere. This value is appropriate in this case as there is no cavitation in the stationary fluids.
6. Save the simulation.

Setting Boundary Conditions

Define the boundary types and specify the appropriate property values. The geometry that is used for this tutorial has six boundaries, three of which have no-slip wall conditions that are assigned to them. Of the remaining three boundaries, define two as pressure outlet boundaries and define the last one as a symmetry plane.

To set the boundary conditions:

1. Rename the **Default_Fluid 2D** node to **Fluid**.
2. Edit the **Fluid** > **Boundaries** node and set the boundary types as follows:

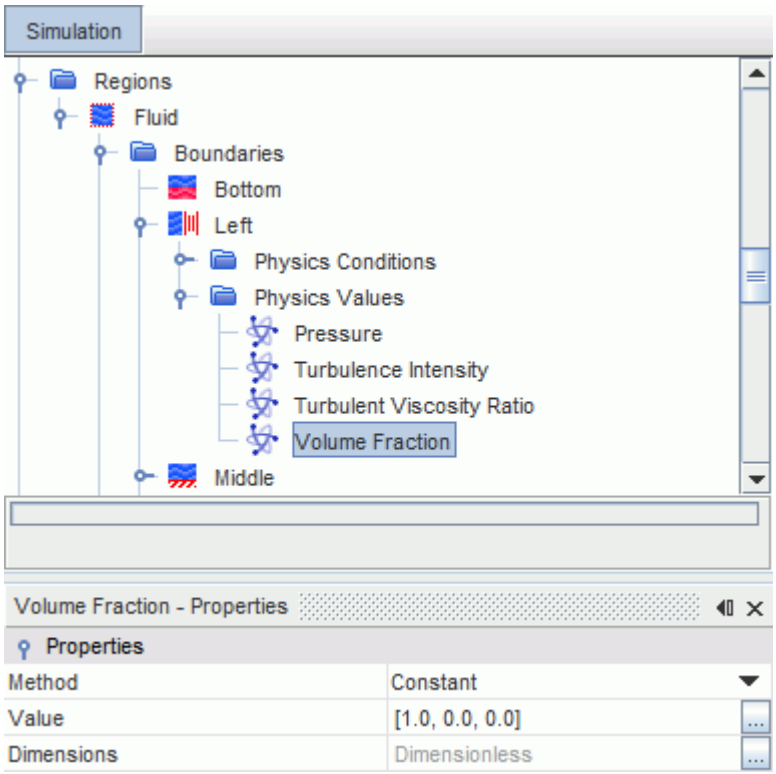
Boundary	Type
Bottom	**Symmetry Plane**
Left	**Pressure Outlet**
Middle	**Wall**
Right	**Pressure Outlet**
TopRight	**Wall**

Boundary	Type
TopLeft	**Wall**

The only boundary conditions that are specified are at the two pressure boundaries.

3. Edit the **Left** > **Physics Values** node and set the following properties:

Node	Property	Setting
Pressure	**Value**	`5.0E7 Pa`
Volume Fraction	**Value**	`[1.0, 0.0, 0.0]`



The **Volume Fraction** setting enforces the condition that only liquid water enters the solution domain through that boundary.

4. Edit the **Right** > **Physics Values** node and set the following property:

Node	Property	Setting
Volume Fraction	**Value**	`[0.0, 1.0, 0.0]`

This setting enforces the condition that only air enters the solution domain through that boundary.

5. Save the simulation.

Setting Solver Parameters and Stopping Criteria

Set the appropriate solver parameters and stopping criteria for the simulation. As you are solving an unsteady problem, it is necessary to specify the time-step size and the elapsed simulation time.

This calculation runs for 5 x 10^{-5} s with a time-step size of 10^{-7} s, so requires 500 time-steps.

To set the solver parameters and stopping criteria:

1. Edit the **Solvers** node and set the following property:

Node	Property	Setting
Implicit Unsteady	**Time-Step**	`1e-7 s`

2. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Physical Time	**Enabled**	Activated
	Maximum Physical Time	`5e-5 s`
Maximum Inner Iterations	**Maximum Inner Iterations**	`5`

3. Save the simulation.

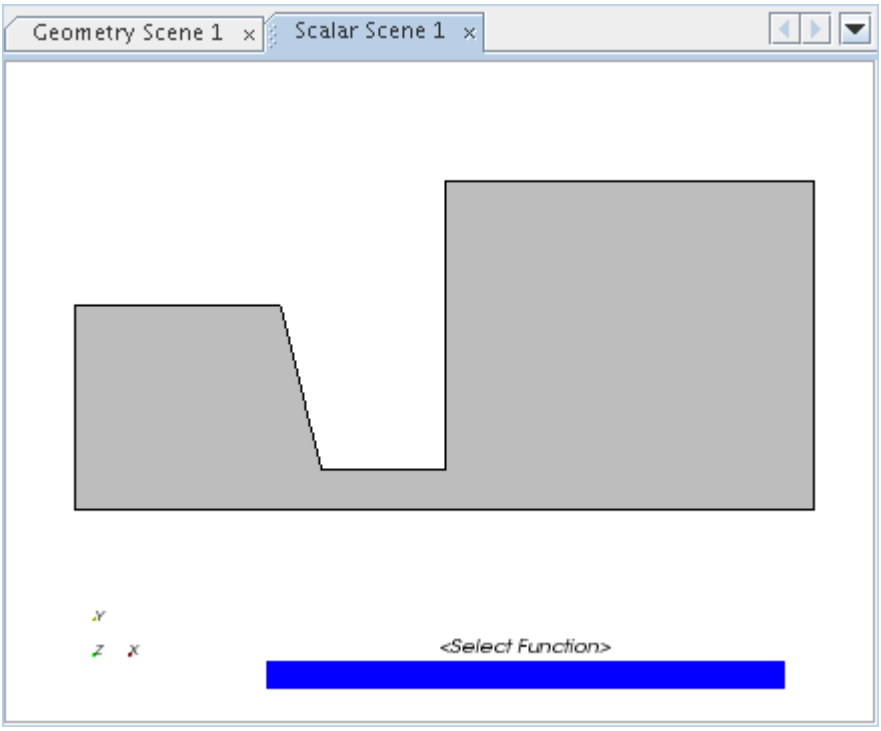
Visualizing and Initializing the Solution

View the distributions of the three-phases throughout the run by creating a scalar scene.

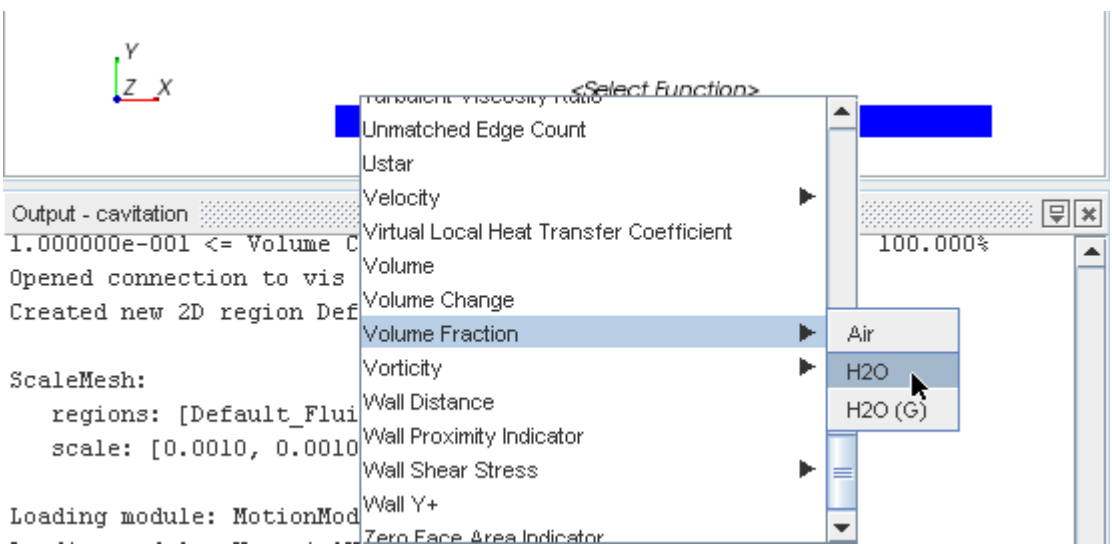
To visualize and initialize the solution:

1. Right-click the **Scenes** node and then select **New Scene** > **Scalar**.

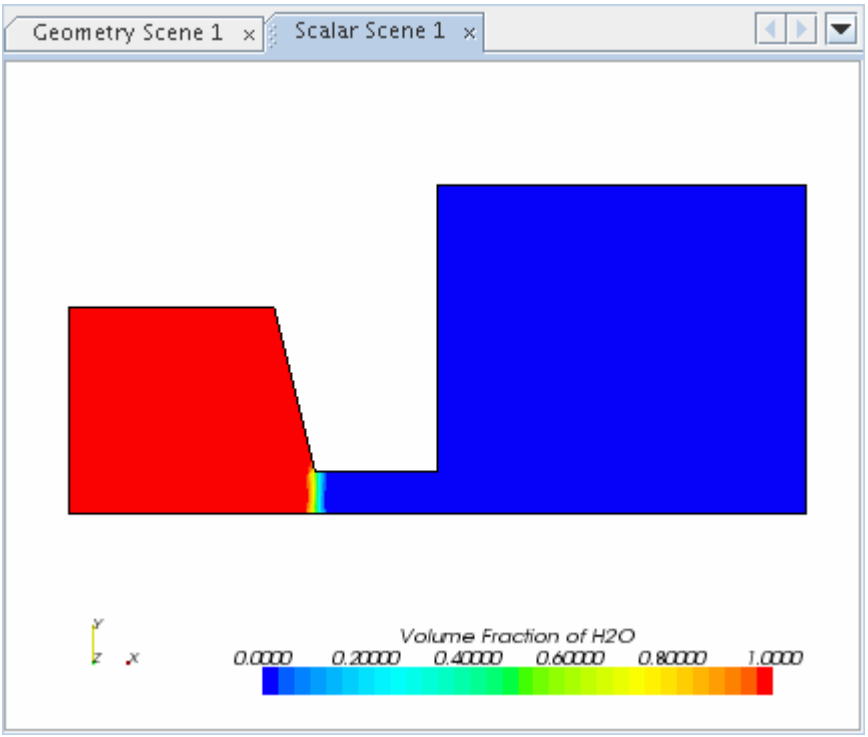
   The *Scalar Scene 1* display appears.



2. In the *Graphics* window, right-click the scalar bar and select **Volume Fraction** > **H2O**.



3. Select the new **Scenes** > **Scalar Scene 1** > **Displayers** > **Scalar 1** node.
4. Set *Contour Style* to **Smooth Filled**.
5. Click (**Initialize Solution**) in the top toolbar or use the **Solution** > **Initialize Solution** menu item.
6. Make the *Scalar Scene 1* display active to check the initialization results.



As expected, at the start of the run the left chamber is entirely filled with water. Changing the scalar function to **Volume Fraction of Air** shows that the right chamber and the connecting channel are entirely filled with air. A small region in which both fluids are apparently present is visible at the interface between the two fluids, but this effect is due to the coarseness of the mesh.

7. Save the simulation.

Running the Simulation

Preparation of the simulation is now complete, and the simulation can be run.

To run the simulation:

1. Click (**Run**) in the *Solution* toolbar.

   The solution progress is displayed in the *Output* window. The *Residuals* display is created automatically in the *Graphics* window and shows the progress of the solvers.

   You can click the *Scalar Scene 1* tab at the top of the *Graphics* window to view the simulation progress. Examine the distributions of each of the three-phases by changing the scalar function regularly.

   During the run, you can stop the process by clicking (**Stop**) on the toolbar. If you do stop the simulation, you can click (**Run**) to resume it.

   If left alone, the simulation continues until 500 time-steps are complete. The numbers that are displayed in the *Output* window represent the inner iterations of the solver, not time-steps. As there are 5 inner iterations per time-step, you expect the run to perform 2,500 inner iterations.

   When the last iteration is complete, the following message is displayed in the *Output* window:

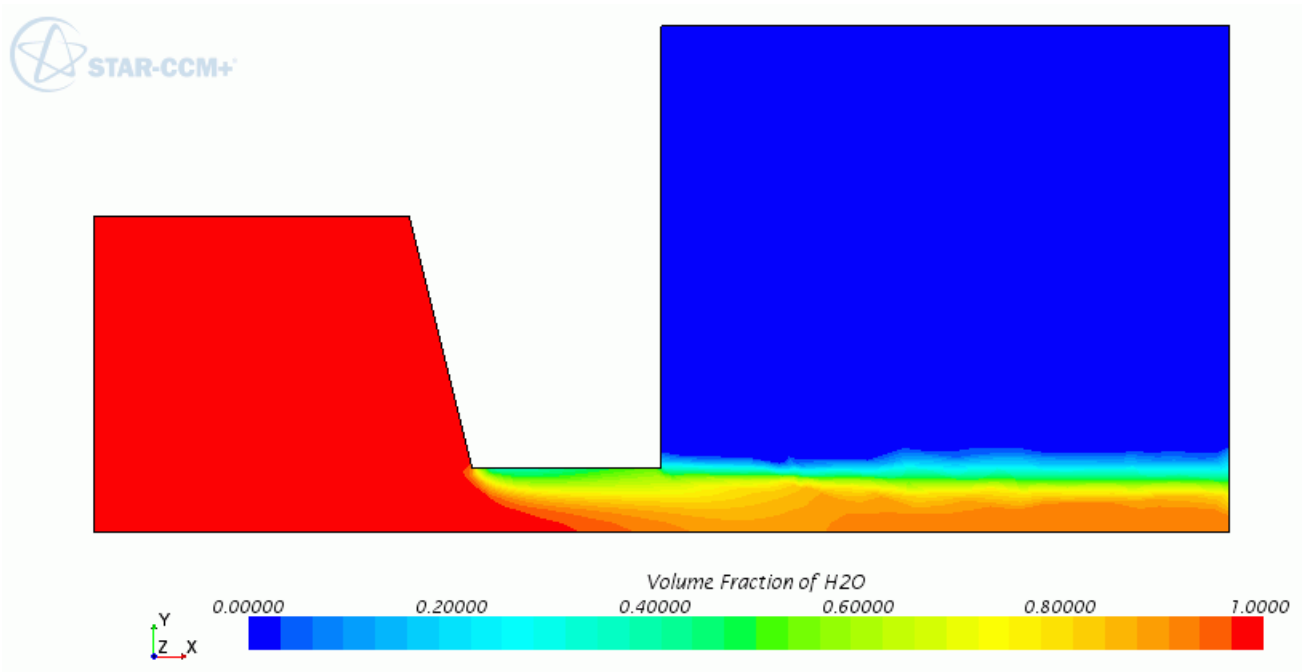
```
Stopping criterion Maximum Steps satisfied.
```

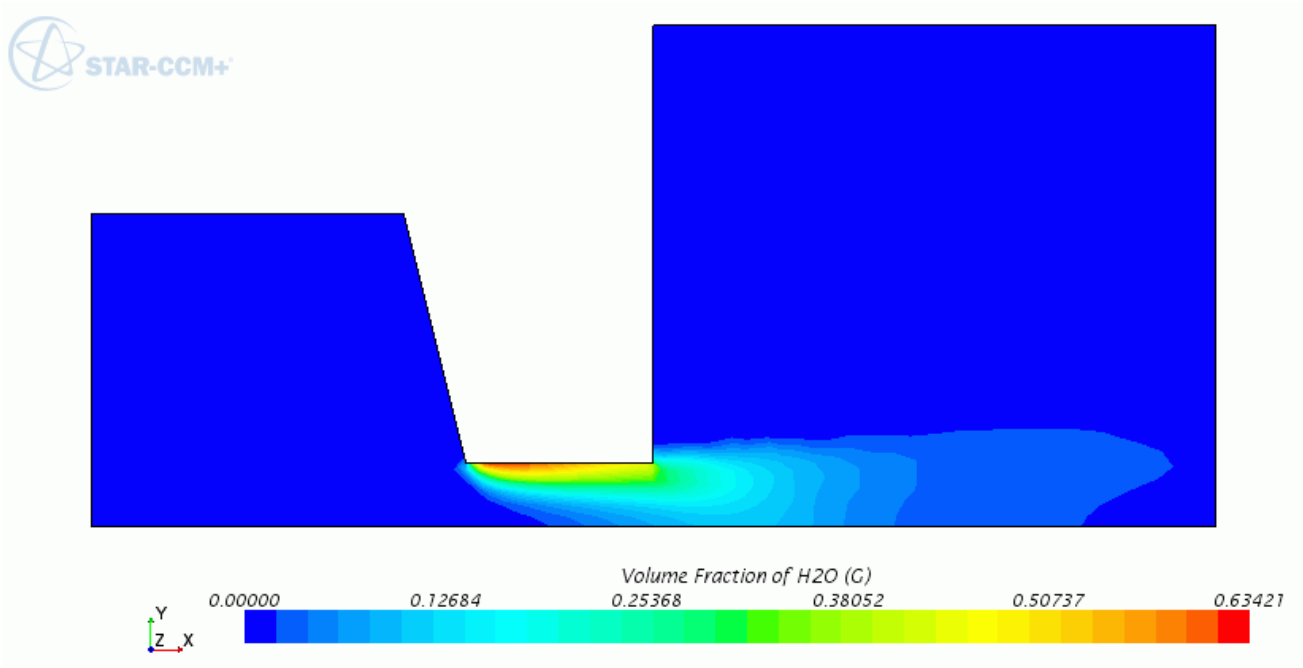
2. When the simulation has finished running, save it.

Visualizing Results

You can view the results of the simulation.

Examine the final distributions of water and water vapor in the *Scalar Scene 1*, as shown below.





As expected, cavitation occurs at the nozzle entrance as a result of the low-pressure region that the sharp corner and the accelerating flow produce.

You can create a contour plot that approximately shows both the water and water vapor distributions by creating and plotting a field function.

To visualize the results:

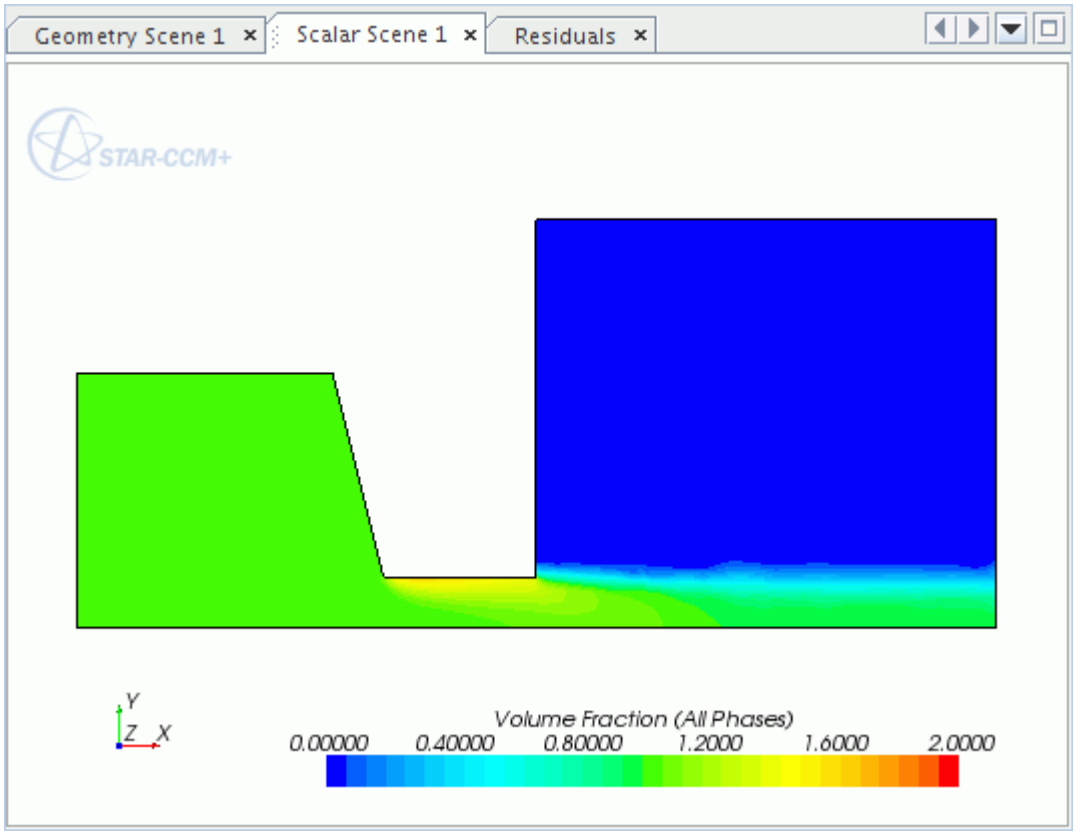
Create the field function to plot.

1. Open the **Tools** node and then right-click the **Field Functions** node and select **New** > **Scalar**.
2. Rename the **User Field Function 1** node to `Volume Fraction (All Phases)`.
3. Select the **Volume Fraction (All Phases)** node and do the following:
   a) Change the *Function Name* to `Volume Fraction (All Phases)`.
   b) To open the custom editor, click the *Definition* box.
   c) In the editor window, enter the following syntax:
   `$VolumeFractionH2O+2*${VolumeFractionH2O (G)}`
   d) To exit from the editor, click **OK**.
4. In the *Scalar Scene 1* display, right-click the color bar and select **Volume Fraction (All Phases)**.

For this plotting technique to work, fix the legend scale to from 0 through 2.

5. Select the **Scenes** > **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Scalar Field** node and set *Max* to `2`

   The contour plot appears as shown below.



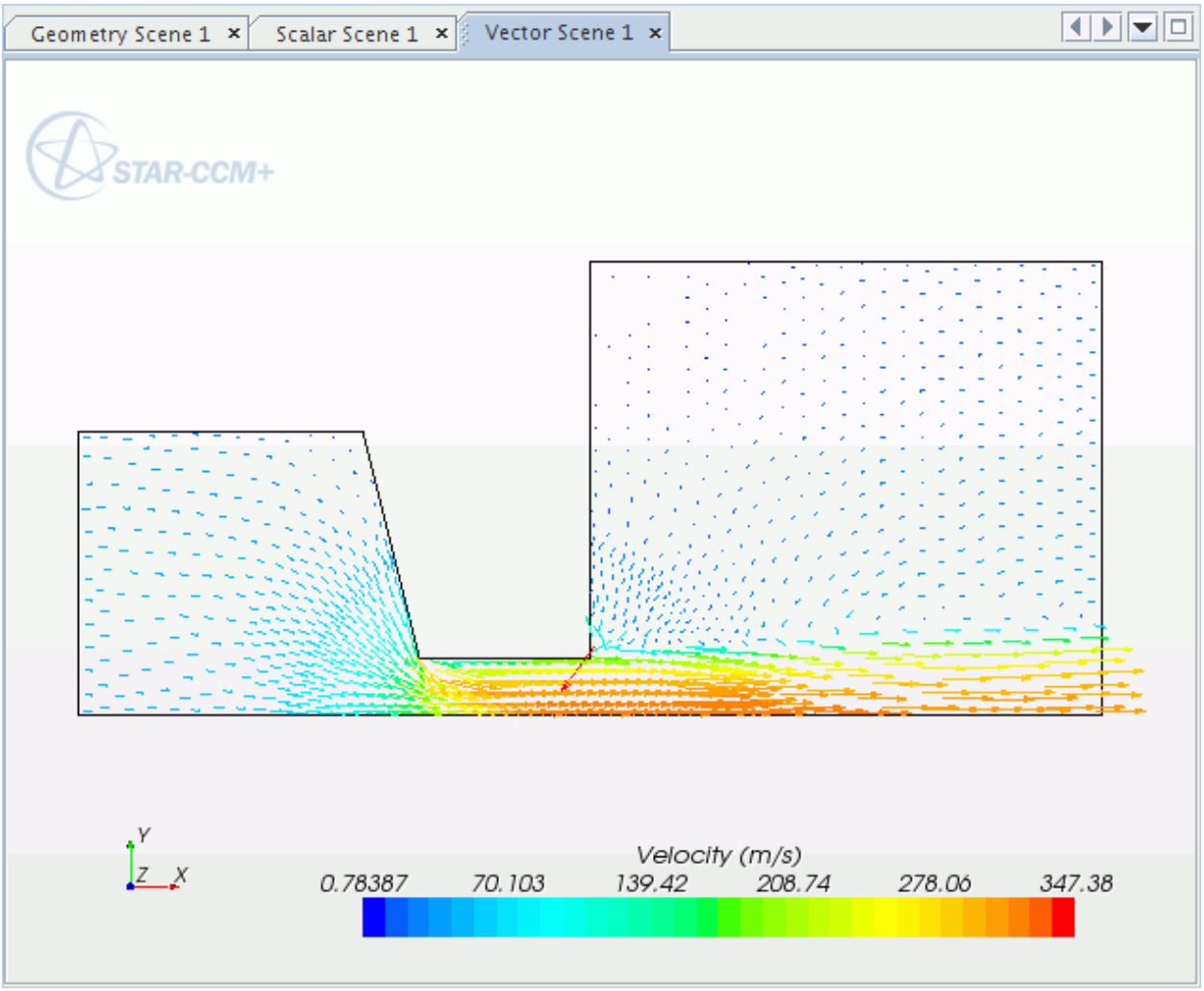
To a reasonable approximation, the blue and turquoise areas are occupied predominantly by air, the green areas predominantly by liquid water and the yellow, orange, and red areas predominantly by water vapor.

Manipulating the results in this way provides a qualitative visual estimate of the spatial extent of each phase. However, the resulting plots are prone to inaccuracies in those places where air and water vapor occur together. Inaccuracies also occur where the liquid water volume fraction is only slightly greater than the water vapor volume fraction. In a case such as this, the latter leads to an over prediction of the size of the region of high water vapor concentration.

To display velocity vectors:

6. Right-click the **Scenes** node and select **New Scene** > **Vector**.

   The vector scene appears in the *Graphics* window.



7. Save the simulation.

Summary

This tutorial demonstrated how to set up a cavitation problem in STAR-CCM+.

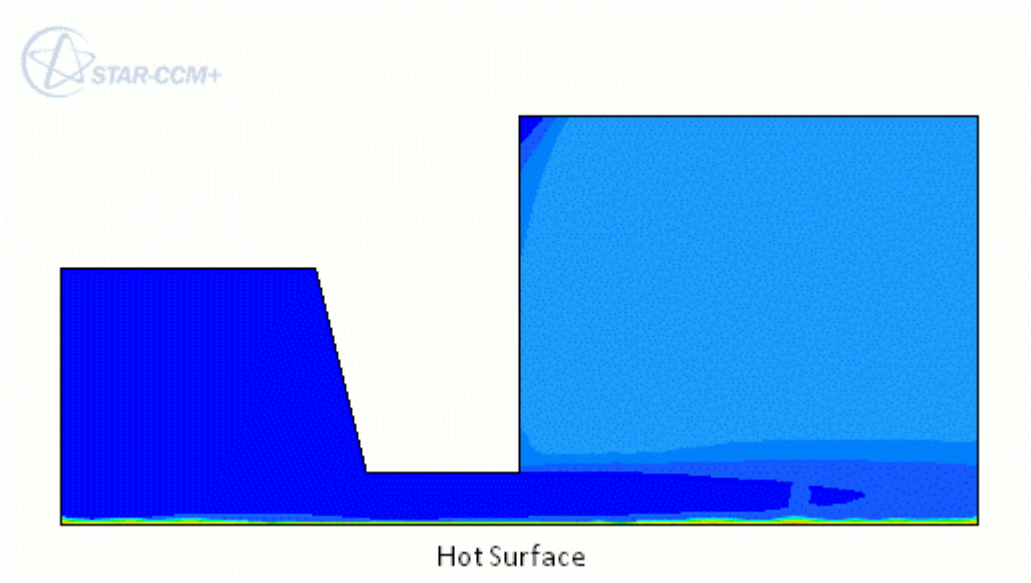
This tutorial has introduced the following STAR-CCM+ features:

- Defining a model for cavitating flow problems.
- Defining material properties that the model requires.
- Defining cavitation model parameters.
- Setting initial conditions.
- Creating and applying field functions.
- Defining boundary conditions.
- Setting solver parameters for an unsteady run.
- Initializing and running the solver for a given number of time-steps.
- Analyzing results.

VOF: Boiling

This tutorial demonstrates how to set up a boiling problem in STAR-CCM+. It simulates water boiling as it flows over a heated surface.

The purpose of this tutorial is to show the typical workflow in setting up and running a boiling simulation with phase change. One of the objectives is to show the impact of the boiling model parameters on the amount of heat transfer obtained. The mesh that is used is the same as in VOF: Gravity-Driven Flow except that, in this instance, you scale down the physical dimensions of the problem so that the nozzle width is about 50 mm.



Water enters the computational domain from the boundary on the left (with a prescribed velocity of 1 m/s and a temperature of 350 K). Water leaves from the boundary on the right (with a prescribed temperature of 370 K at atmospheric pressure). The boundary at the bottom is assumed to be a wall with a prescribed fixed temperature of 540 K. All other boundaries are assumed to be adiabatic solid walls.

Contents:

Prerequisites
Importing the Mesh and Naming the Simulation
Converting to a Two-Dimensional Mesh
Scaling the Mesh
Selecting the Physics Models
Setting the Material Properties
Defining the Phase Interactions
Defining the Boiling Model Parameters
Setting Initial Conditions
Setting Boundary Conditions

Setting Solver Parameters and Stopping Criteria
Reporting, Monitoring, and Plotting
Visualizing the Solution
Running the Simulation
Visualizing Results
Modifying the Boiling Model Parameters
Running the Simulation
Visualizing Results
Summary
Bibliography

Prerequisites

The instructions in the VOF: Boiling tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Importing the Mesh and Naming the Simulation

Launch STAR-CCM+, import the supplied geometry, and save the simulation.

A one-cell-thick three-dimensional polyhedral cell mesh has been predefined for this analysis.

To import the mesh and name the simulation:

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh**.
4. In the *Open* dialog, choose `doc/startutorialsdata/multiphaseFlow/data/gridfs.ccm`, then click **Open**.

   STAR-CCM+ provides feedback on the import process, which takes a few seconds, in the *Output* window.
5. Save the simulation as `vofBoiling.sim`.

Converting to a Two-Dimensional Mesh

You convert the mesh from a three-dimensional mesh to two-dimensions.

There are special requirements in STAR-CCM+ to convert a three-dimensional mesh to two-dimensions. These requirements are:

- The grid must be aligned with the X-Y plane.
- The grid must have a boundary plane at the Z = 0 location.

The mesh that is imported for this tutorial is built with these requirements in mind. Had the grid not met them, it would have been necessary to realign the region using the mesh transformation and rotation capabilities in STAR-CCM+.

To convert the mesh to two-dimensions:

1. Right-click the **Scenes** node and then select **New Scene** > **Mesh**.
2. Select **Mesh** > **Convert to 2D**.
3. In the *Convert Regions To 2D* dialog that appears, do the following:
   a) Ensure that the **Default_Fluid** region is selected.
   b) Activate **Delete 3D Regions After Conversion**.
   c) Click **OK**.

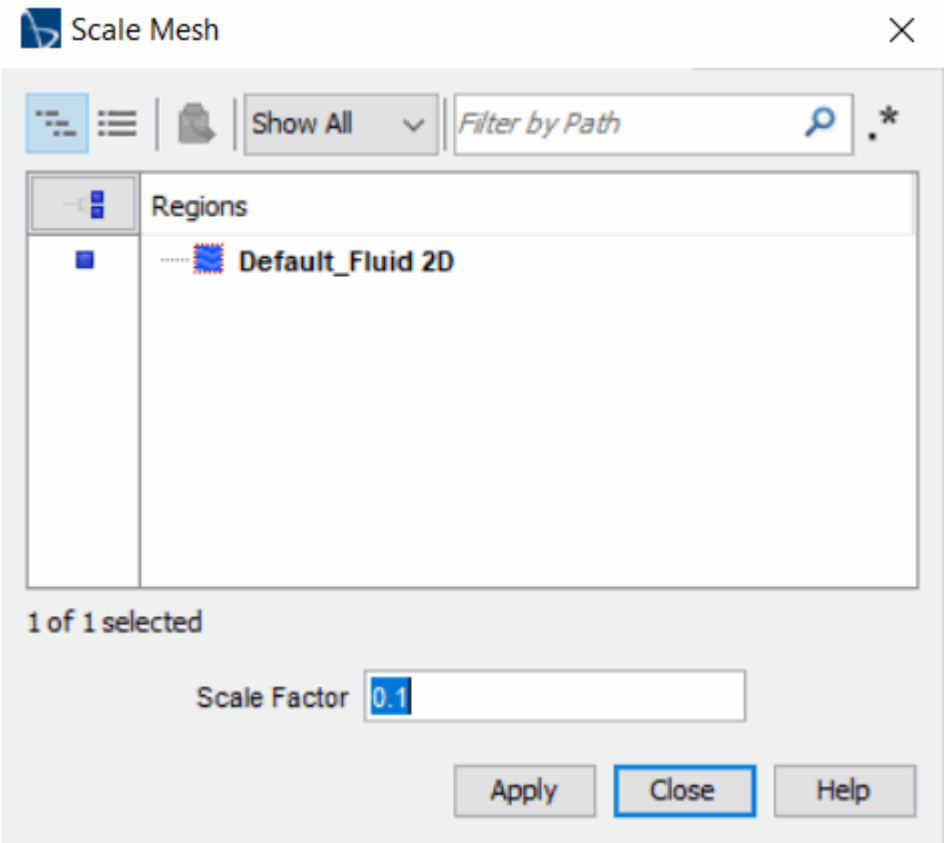
The mesh conversion takes place and the new two-dimensional mesh is shown, viewed from the z-direction, in the *Mesh Scene 1* display. The mouse rotation option is suppressed for two-dimensional scenes.

Scaling the Mesh

The original mesh is not built to the correct scale and therefore requires scaling down by a factor of ten.

To scale the mesh:

1. Select **Mesh** > **Scale Mesh** from the menu bar.
2. In the *Scale Mesh* dialog:
   a) Ensure that the **Default_Fluid 2D** region is selected.
   b) Set the **Scale Factor** to `0.1`.
   c) Click **Apply** to scale the domain and then click **Close**.



When the **Apply** button is clicked, the mesh region reduces in size.

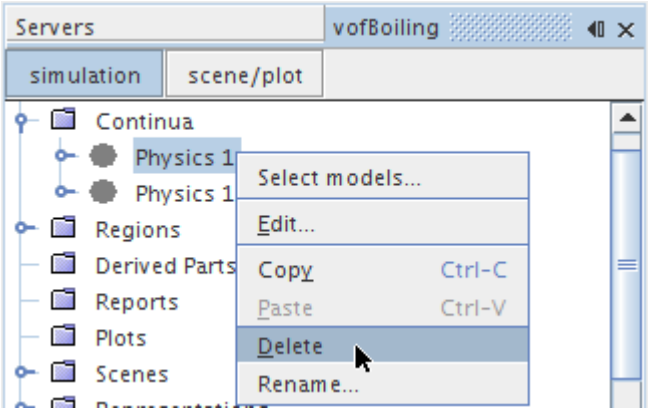
3. To restore the previous viewing distance, click (**Reset View**) in the visualization toolbar.
4. Select **Mesh** > **Diagnostics** from the menu bar.
5. In the *Mesh Diagnostics* dialog, click **OK**.

The extent of the geometry is as given below:

```
x: [-4.0000e-01, 5.0000e-01] m
y: [-5.0586e-18, 4.0000e-01] m
z: [0.00000e+00, 0.0000e+00] m
```

The physics continuum that is associated with the deleted three-dimensional region is no longer required. To delete it:

6. Expand the **Continua** node. Right-click the **Physics 1** continuum node and select **Delete**.



7. Click **Yes** in the *Confirm Object Deletion* dialog.

Selecting the Physics Models

Physics models define the primary variables of the simulation, including pressure, temperature, velocity, and what mathematical formulation is used to generate the solution.

In this example, the flow is turbulent and the problem involves multiphase flow and boiling. Two fluids (water and water vapor) are required for the analysis. However, since these fluids occupy the same domain, only one continuum and one region are required to set up the simulation. By default, a continuum that is called **Physics 1 2D** is created when the mesh is converted to two-dimensional.

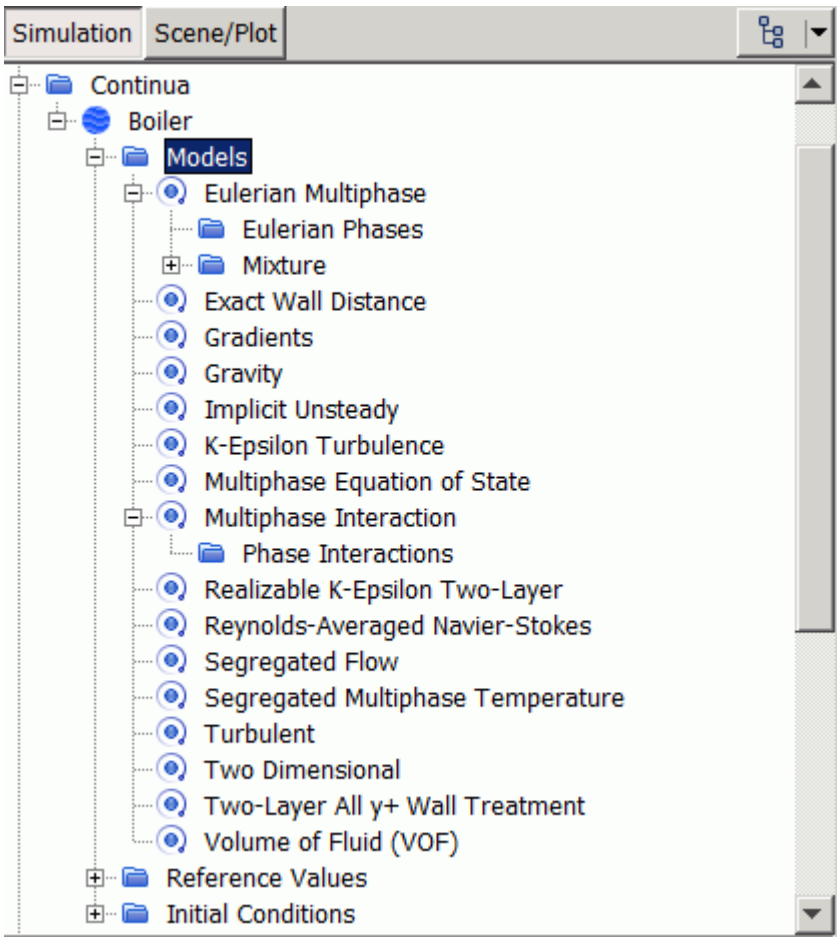
To select the physics models:

1. Rename the **Physics 1 2D** node to **Boiler**.
2. For the physics continuum, **Continua** > **Boiler**, select the following models in order:

Group Box	Model
Enabled Models	**Two Dimensional** (Pre-selected)
Time	**Implicit Unsteady**
Material	**Eulerian Multiphase** **Multiphase Interaction** (selected automatically)
Eulerian Multiphase Model	**Volume of Fluid (VOF)** **Multiphase Equation of State** (selected automatically) **Segregated Flow** (selected automatically) **Gradients** (selected automatically)
Viscous Regime	**Turbulent** **Reynolds-Averaged Navier-Stokes** (selected automatically)
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**

Group Box	Model
	Realizable K-Epsilon Two-Layer (selected automatically) **Exact Wall Distance** (selected automatically) **Two-Layer All y+ Wall Treatment** (selected automatically)
Optional Models	**Segregated Multiphase Temperature** **Gravity**

3. Click **Close**.
4. To review the models, open the **Boiler** > **Models** node.



5. Save the simulation.

Setting the Material Properties

Define the phases and set the appropriate material properties. Define the materials corresponding to each of the mixture components under the **Eulerian Phases** node.

To define the phases and set the material properties:

1. In the **Boiler** continuum, select the **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and create a new phase.
2. Rename the **Phase 1** node to **H2O**.
3. For the **H2O** phase, select the following models:

Group Box	Model
Material	**Liquid**
Equation of State	**Constant Density**

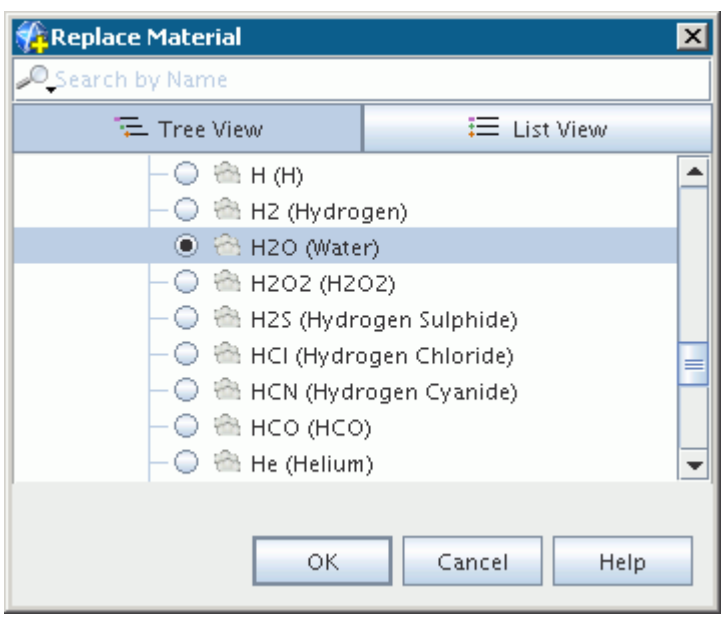
4. Click **Close**.
5. Create a second phase and rename it to **H2O (G)**.
6. For the **H2O (G)** phase, select the following models:

Group Box	Model
Material	**Gas**
Equation of State	**Constant Density**

7. Click **Close**.

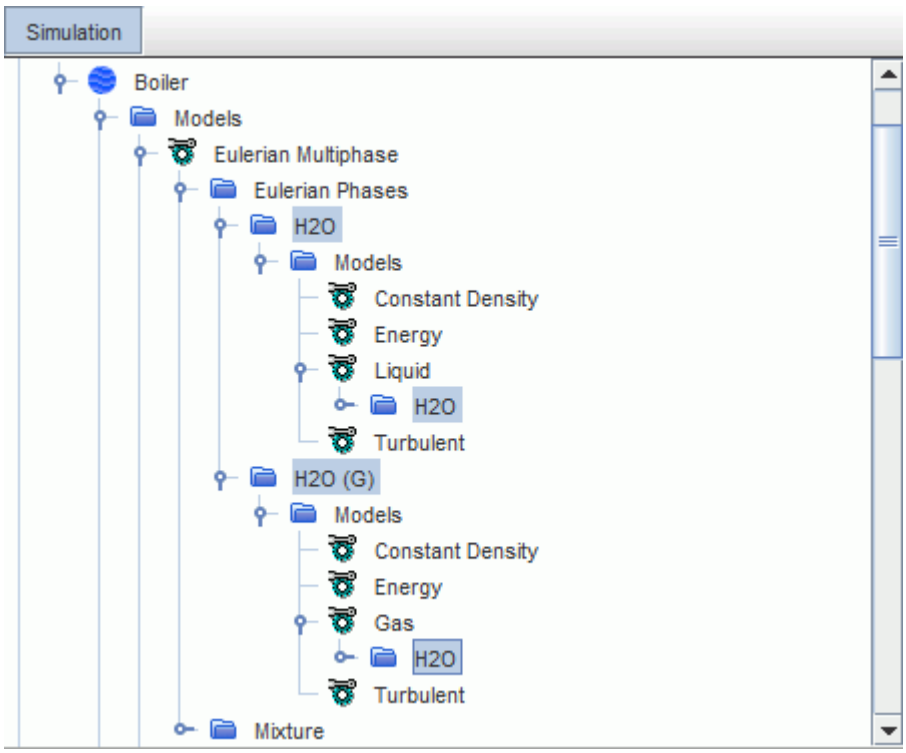
The default gas must be replaced with water from the STAR-CCM+ material database.

8. Right-click the **H2O (G)** > **Models** > **Gas** > **Air** node and select **Replace with**.
9. In the *Replace Material* dialog, expand the **Material Databases** > **Standard** > **Gases** node.
10. Activate **H2O (Water)**.



11. Click **OK** to replace the material.

    The phase model objects appear as shown below.



The default material properties for the two phases are suitable for this problem.

12. Save the simulation.

Defining the Phase Interactions

Define the interactions between the liquid and vapor phases. You specify the H2O phase as Primary Phase and the H2O (G) phase as Secondary Phase.

Using the **Multiphase Interaction** model, you can define the interaction between the liquid and vapor phases.

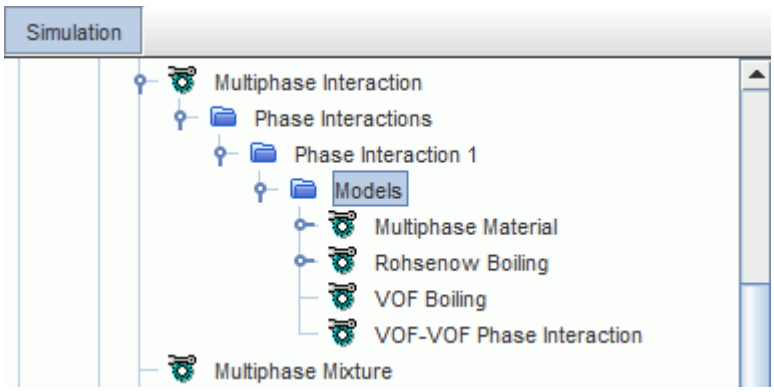
To define the phase interactions:

1. In the **Boiler** continuum, right-click the **Models** > **Multiphase Interaction** > **Phase Interactions** node and create a new phase interaction.
2. For the **Phase Interaction 1** phase interaction, select the following models in order:

Group Box	Model
Phase Interaction Topology	**VOF-VOF Phase Interaction**
Optional Models	**VOF Boiling**
Boiling Models	**Rohsenow Boiling** **Multiphase Material** (selected automatically)

3. Click **Close**.

   The selected models appear within the **Phase Interaction 1** > **Models** node, as shown below.



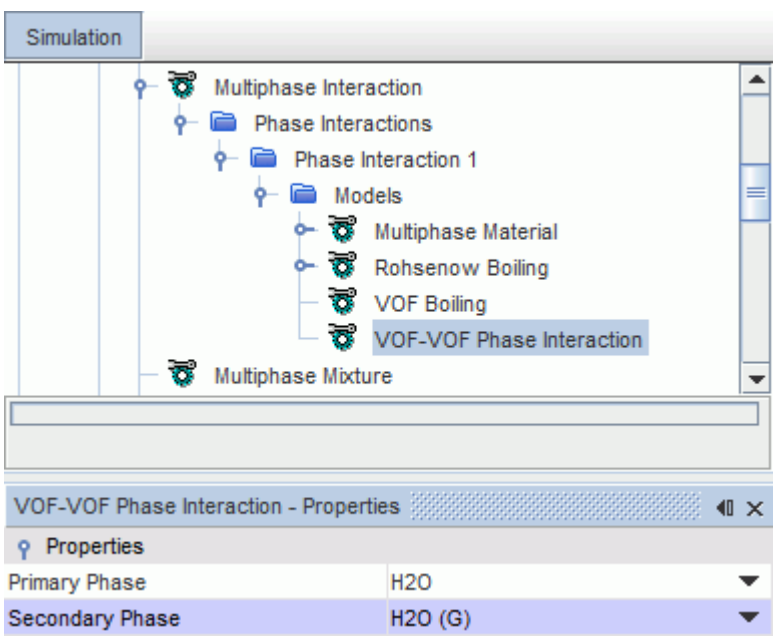
4. Save the simulation.

Defining the Boiling Model Parameters

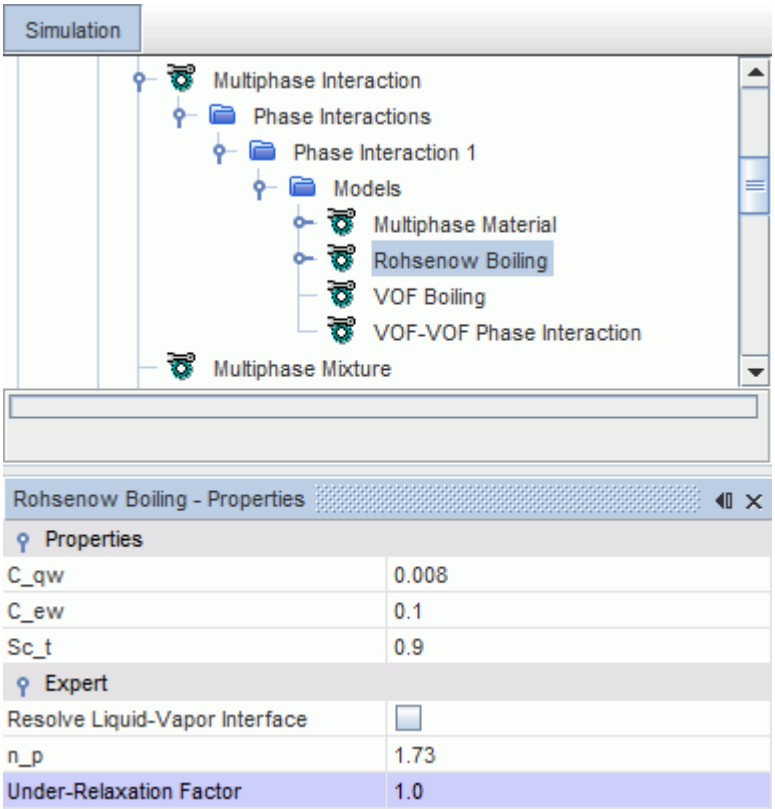
Define the appropriate parameters for the Boiling model. You must specify which liquid phase is boiling and which gas phase is the product of boiling.

To define the Boiling model parameters:

1. Select the **Phase Interaction 1** > **Models** > **VOF-VOF Phase Interaction** node and set **H2O** for the *Primary Phase* property and **H2O (G)** for the *Secondary Phase* property.



Wall heat flux due to boiling is a highly non-linear function at wall boundaries where the wall temperature is part of the numerical solution. For example, for given heat flux wall boundaries or solid-fluid interfaces. In these cases, to improve the convergence, you can reduce the under-relaxation factor for heat flux due to boiling (*Under-Relaxation Factor* property under the **Rohsenow Boiling** node).



2. Save the simulation.

Setting Initial Conditions

Set the initial conditions for the physics continuum. You specify the initial volume fraction and temperature within the continuum. There is no initial water vapor in the continuum.

To set the initial conditions for the **Boiler** continuum:

1. Edit the **Initial Conditions** node and set the following properties:

Node	Property	Setting
Volume Fraction	**Value**	`[1.0, 0.0]`
Static Temperature	**Value**	`350 K`

2. Save the simulation.

Setting Boundary Conditions

You define the boundary types and specify the appropriate property values.

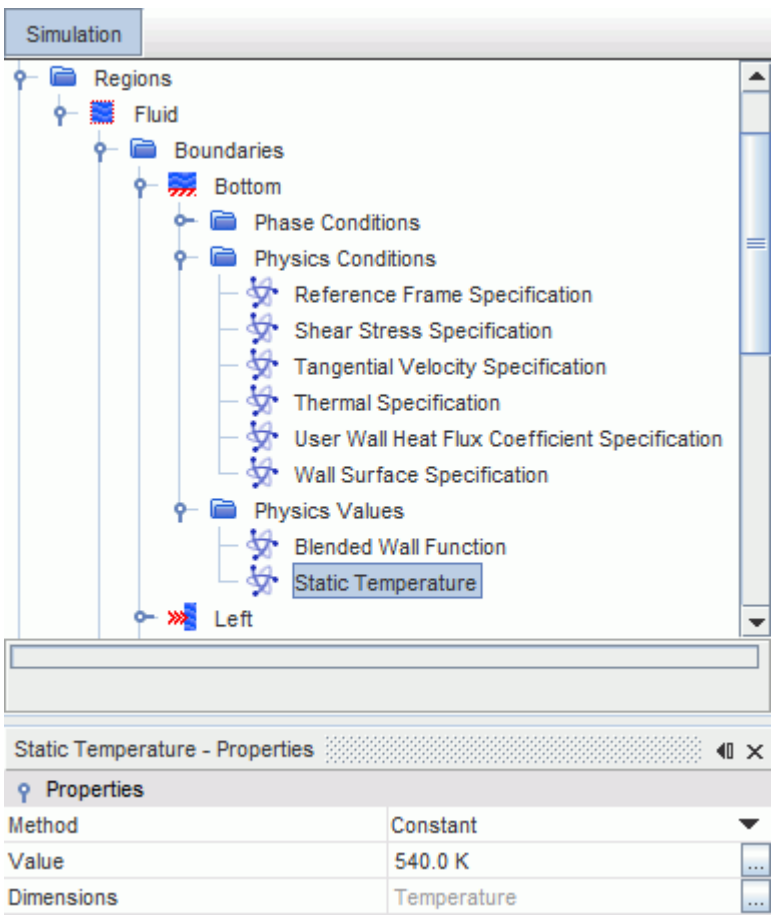
To set the boundary conditions:

1. Rename the **Regions** > **Default_Fluid 2D** node to **Fluid**.
2. Edit the **Fluid** > **Boundaries** node and set the boundary types as follows:

Boundary	Type
Bottom	**Wall**
Left	**Velocity Inlet**
Middle	**Wall**
Right	**Pressure Outlet**
TopRight	**Wall**
TopLeft	**Wall**

3. Edit the **Bottom** boundary and set the following properties:

Node	Property	Setting
Physics Conditions > **Thermal Specification**	**Condition**	**Temperature**
Physics Values > **Static Temperature**	**Value**	`540 K`

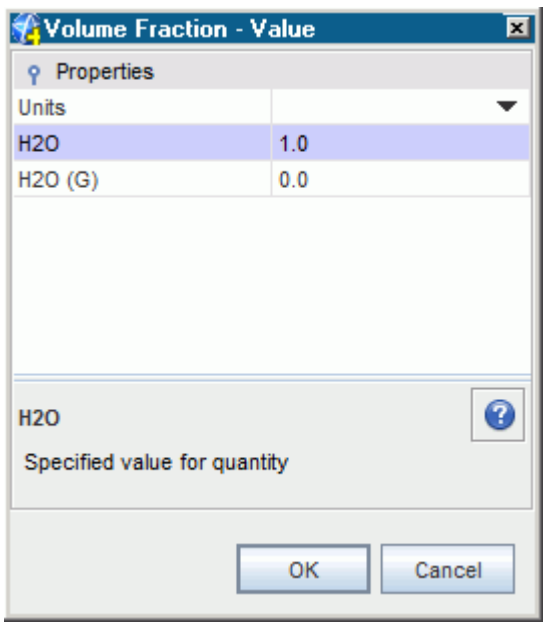


4. Edit the **Left** > **Physics Values** node and set the following properties:

Node	Property	Setting
Static Temperature	**Value**	`350 K`
Velocity Magnitude	**Value**	`1 m/s`
Volume Fraction	**Value**	`[1.0, 0.0]`

5. Edit the **Right** > **Physics Values** node and set the following properties:

Node	Property	Setting
Static Temperature	**Value**	`370 K`
Volume Fraction	**Value**	`[1.0, 0.0]`



6. Save the simulation.

Setting Solver Parameters and Stopping Criteria

Set the appropriate solver parameters and stopping criteria for the simulation. As you are solving an unsteady problem, it is necessary to specify the time-step size and the elapsed simulation time. This calculation runs for a maximum physical time of 3 s with a time-step size of 0.01 s.

Boiling flows are typically unsteady, with strong buoyancy effects, and large-scale flow instabilities. It is therefore recommended that unsteady solvers are used for this type of flows. You also have cases where the mean flow is dominant (as is the case in this tutorial) and where, eventually, a steady-state solution is possible. In these situations, try to simulate these flows using the steady-state model and an appropriate set of under-relaxation parameters. However, it is recommended that even for these cases, you use a time-marching solver with one iteration per time-step and a time-step size that secures a stable run.

To set the solver parameters and stopping criteria:

1. Edit the **Solvers** node and set the following properties:

Node	Property	Setting
Implicit Unsteady	**Time-Step**	`0.01 s`
Segregated Flow > **Velocity**	**Under-Relaxation Factor**	`0.8`
Segregated VOF > **Single Step**	**Under-Relaxation Factor**	`0.1`

2. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Physical Time	**Enabled**	Activated
	Maximum Physical Time	`3 s`
Maximum Inner Iterations	**Maximum Inner Iterations**	`1`

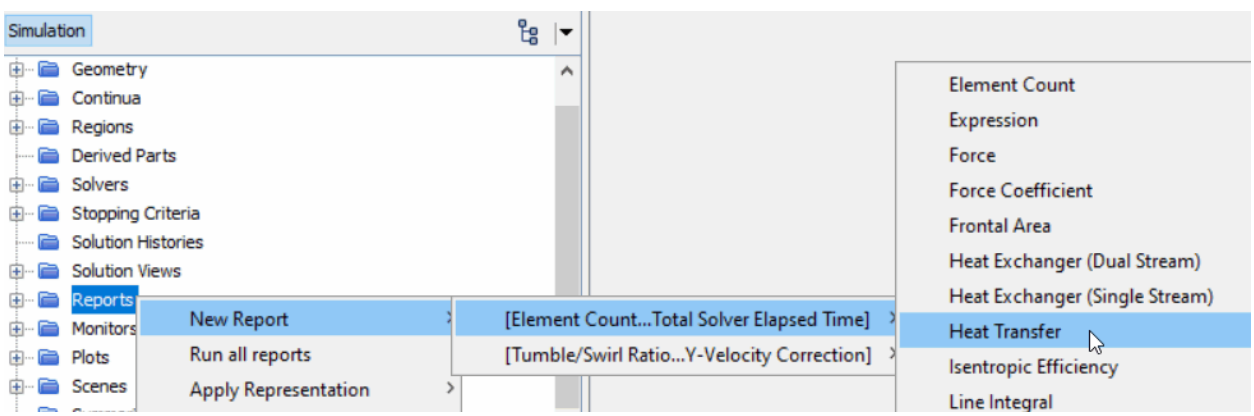
3. Save the simulation.

Reporting, Monitoring, and Plotting

You create a heat flux report, monitor, and plot for the bottom wall to assess convergence to a steady-state solution.

To create the report, monitor, and plot:

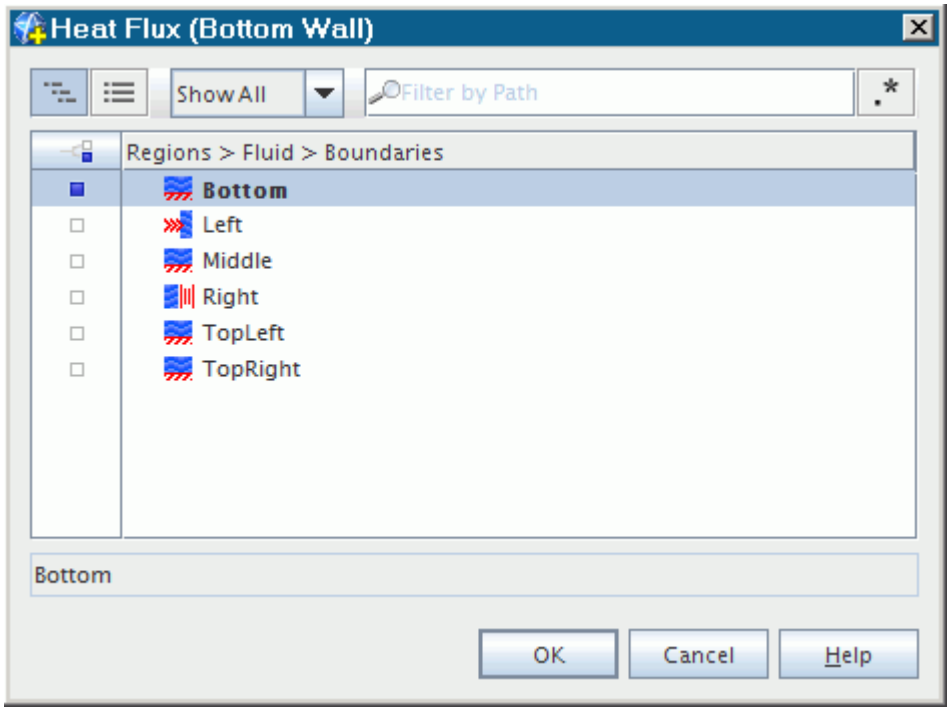
1. Right-click the **Reports** node and select **New Report** > **Heat Transfer.**



A new report is created named **Heat Transfer 1**.

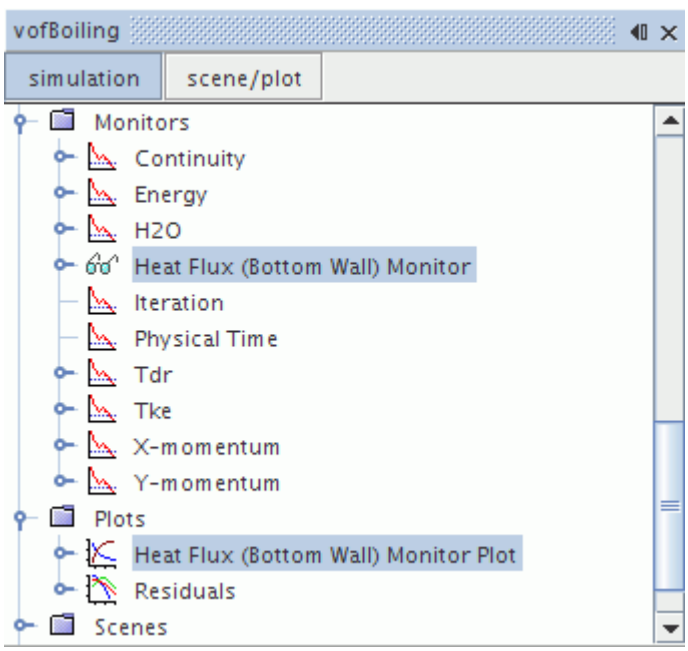
To monitor heat flux on the bottom wall:

2. Rename the **Heat Transfer 1** node to `Heat Flux (Bottom Wall)`.
3. Make sure that the **Reports** > **Heat Flux (Bottom Wall)** node is selected and click to the right of *Parts* property.
4. Select **Bottom** and click **OK**.



5. Right-click the **Heat Flux (Bottom Wall)** node and select **Create Monitor and Plot from Report**.

   A new monitor and plot is created.



6. Select **Plots** > **Heat Flux (Bottom Wall) Monitor Plot** and set *x-Axis Monitor* to **Iteration**.

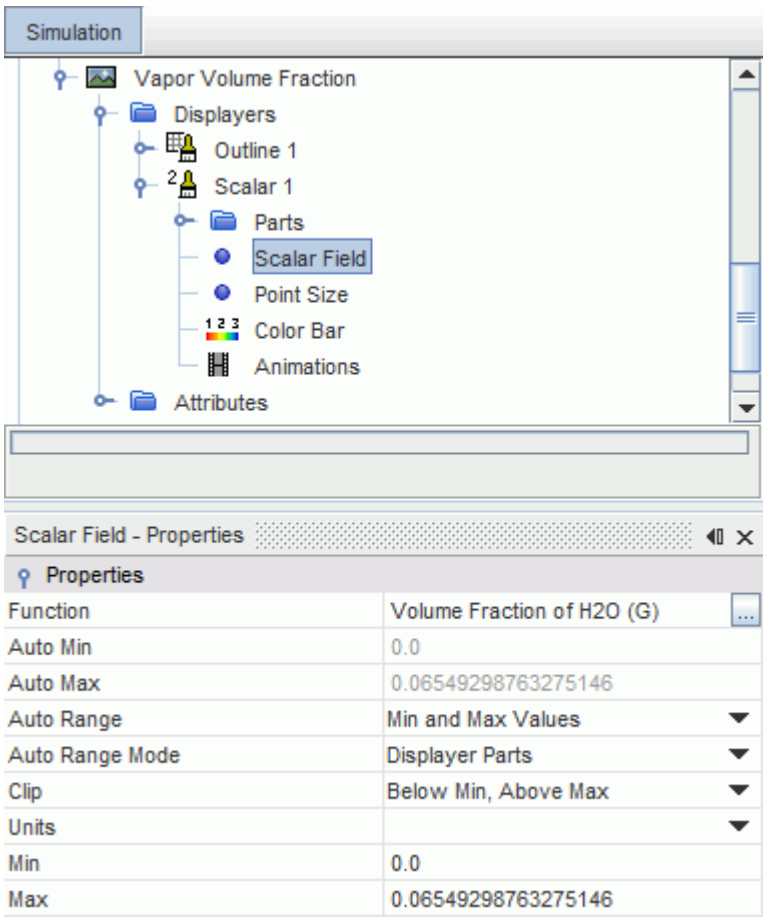
Visualizing the Solution

Create a scene to view the results of the simulation. You can view the distributions of the vapor volume fraction and fluid temperature throughout the run.

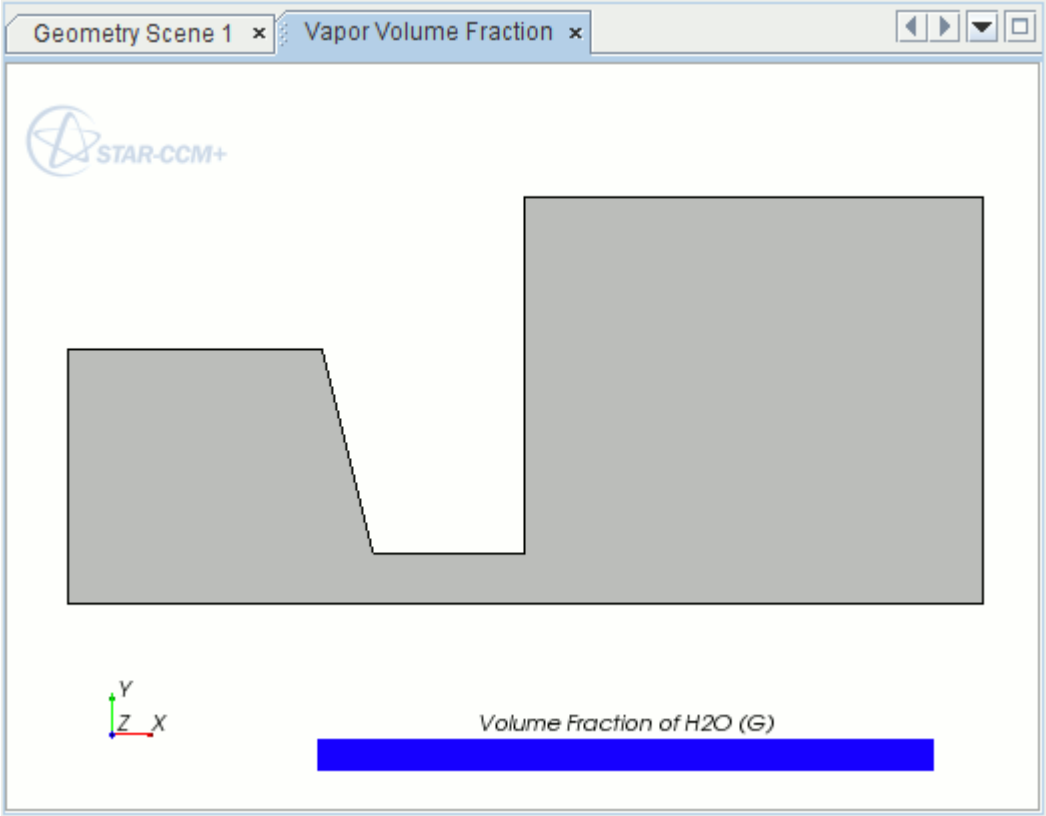
To view the results of the simulation:

1. Right-click the **Scenes** node and select **New Scene** > **Scalar**.

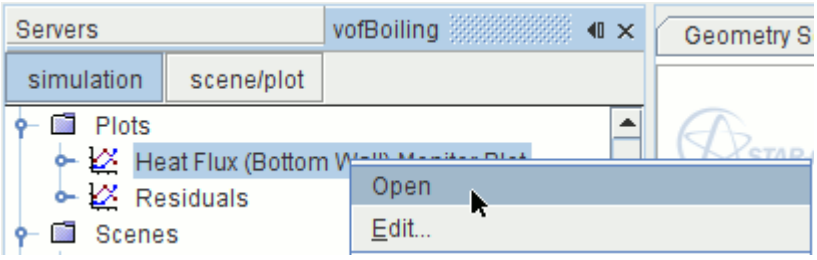
   The *Scalar Scene 1* display appears in the *Graphics* window.
2. Open the *scene/plot* window.
3. Rename the **Scalar Scene 1** node to `Vapor Volume Fraction`.
4. Select the **Displayers** > **Scalar 1** > **Scalar Field** node and set *Function* to **Volume Fraction** > **Volume Fraction of H2O (G)**.



5. Select the **Displayers** > **Scalar 1** node and set *Contour Style* to **Smooth Filled**.
6. Use the mouse buttons to zoom in and position the part to obtain a view similar to the view shown below.



7. Follow the above steps and create another scalar scene.
8. Rename the scene to `Fluid Temperature`.
9. Set *Function* to `Temperature`.
10. Right-click the **Plots** > **Heat Flux (Bottom Wall) Monitor Plot** and select **Open**.



11. Save the simulation.

Running the Simulation

Preparation of the simulation is now complete, and the simulation can be run.

To run the simulation:

1. Click (**Run**) in the *Solution* toolbar.

The solution progress is displayed in the *Output* window. The *Residuals* display is created automatically in the *Graphics* window and shows the progress of the solvers.

While the simulation is running, you can click the tabs at the top of the *Graphics* window to view scenes and plots. You can monitor the convergence by viewing the *Heat Flux (Bottom Wall) Monitor Plot*.

During the run, you can stop the process by clicking (**Stop**) on the toolbar. If you do stop the simulation, you can click (**Run**) to resume it.

If left alone, the simulation continues until 300 time-steps are complete. When the last time-step is complete, the following message is displayed in the *Output* window:

```
Stopping criterion Maximum Steps satisfied.
```

2. When the simulation has finished running, save it.

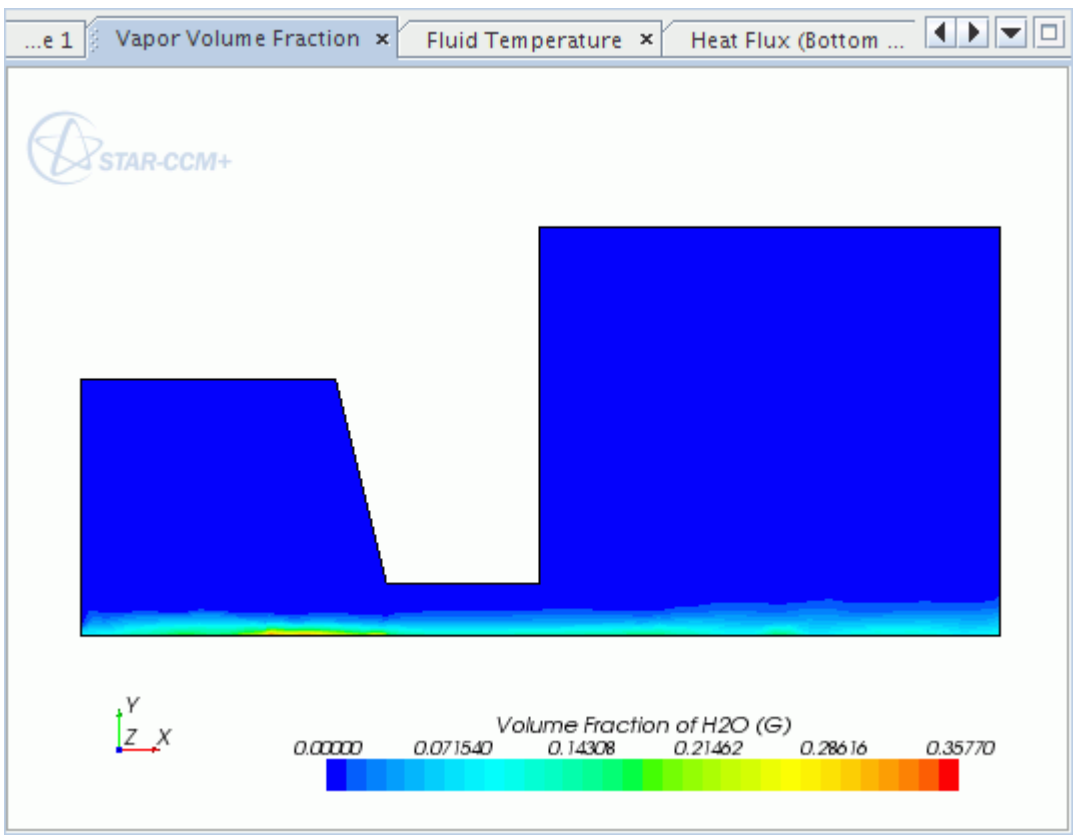
Visualizing Results

The scene tabs display the results of the simulation.

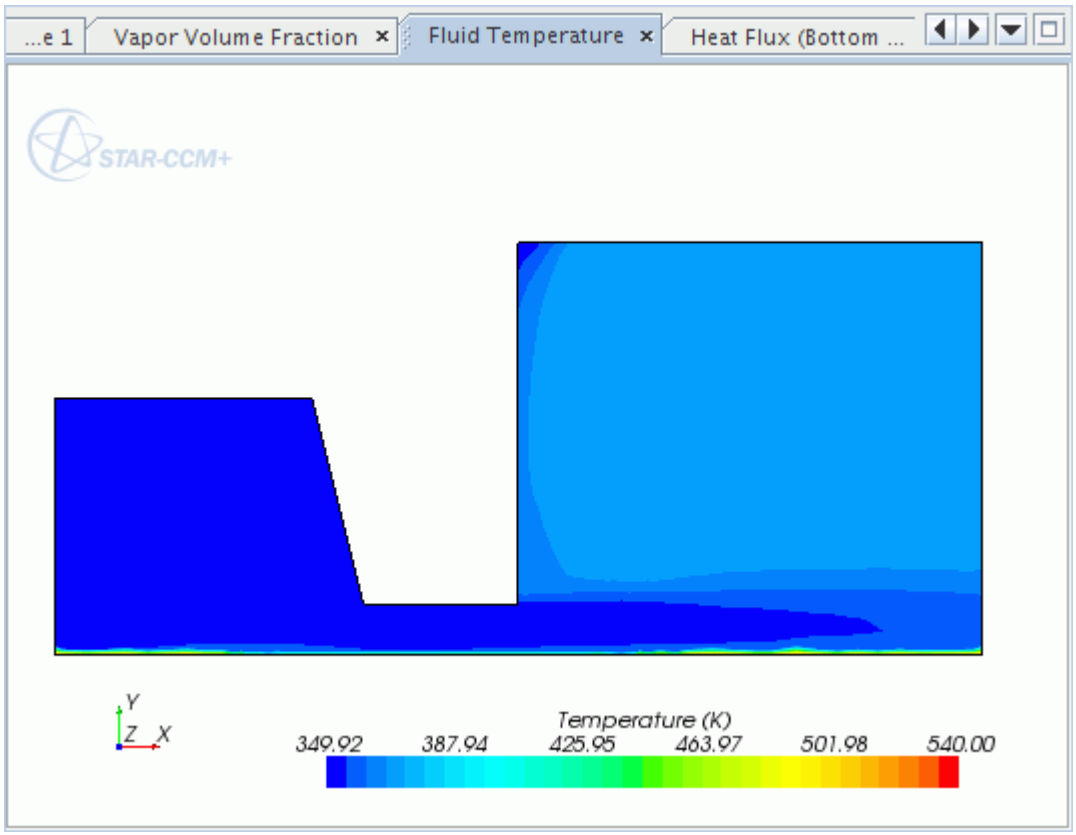
To view the results of the simulation:

1. Select the scene tabs in turn.

   A vapor volume fraction scene similar to the one shown below is displayed.



A fluid temperature scene similar to the one shown below is displayed.



A heat flux profile similar to the one shown below is displayed.

2. Save the simulation.

Modifying the Boiling Model Parameters

The Rohsenow empirical correlation is used to calculate the surface heat flux due to boiling. You specify the two parameters of this correlation, namely *C_qw* and *n_p*.

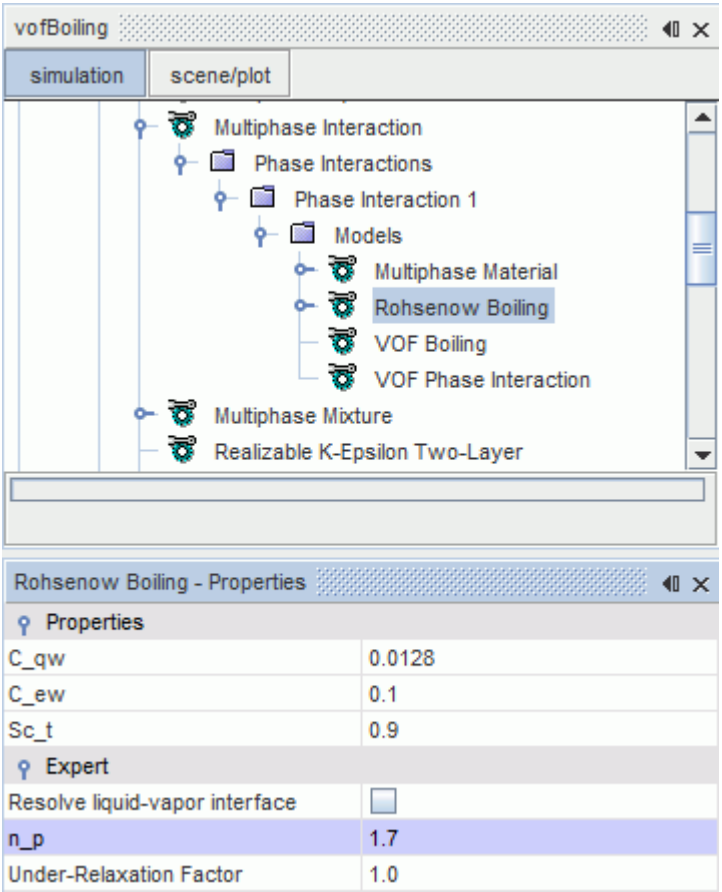
In STAR-CCM+, the [791] empirical correlation is used to calculate the surface heat flux due to boiling. The two parameters of this correlation, namely *C_qw* and *n_p*, are specified. These empirical coefficients vary with the liquid-surface combination and the surface finish. [792] provides some typical parameters for various conditions.

Assume that the bottom wall is a polished copper and specify appropriate values for *C_qw* and *n_p* parameters.

To modify the Boiling model parameters:

1. Select the **Continua** > **Boiler** > **Models** > **Multiphase Interaction** > **Phase Interaction 1** > **Models** > **Rohsenow Boiling** node and set *C_qw* to `0.0128` and *n_p* to `1.7`.

Leave the remaining parameters at their default values.



2. Save the simulation.

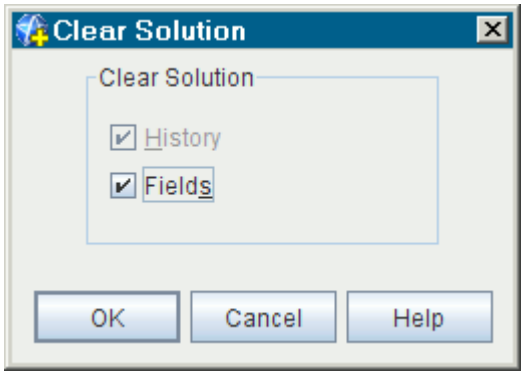
Running the Simulation

Clear the results of the previous simulation and run the simulation again.

To clear the results and run the simulation again:

Clear the previous solution.

1. In the menu, select **Solution** > **Clear Solution**.
2. In the *Clear Solution* dialog, click **OK**.



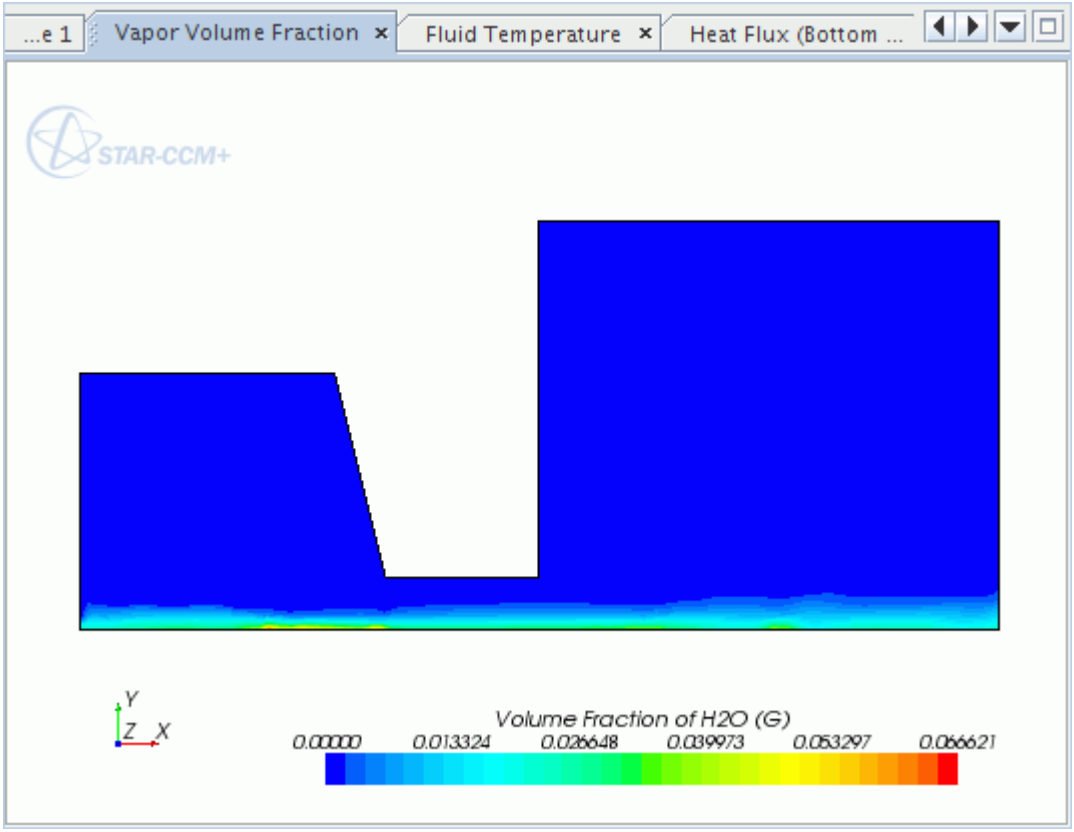
The simulation is now ready to be run.

3. Click (**Run**) in the *Solution* toolbar.
4. When the simulation has finished running, save it.

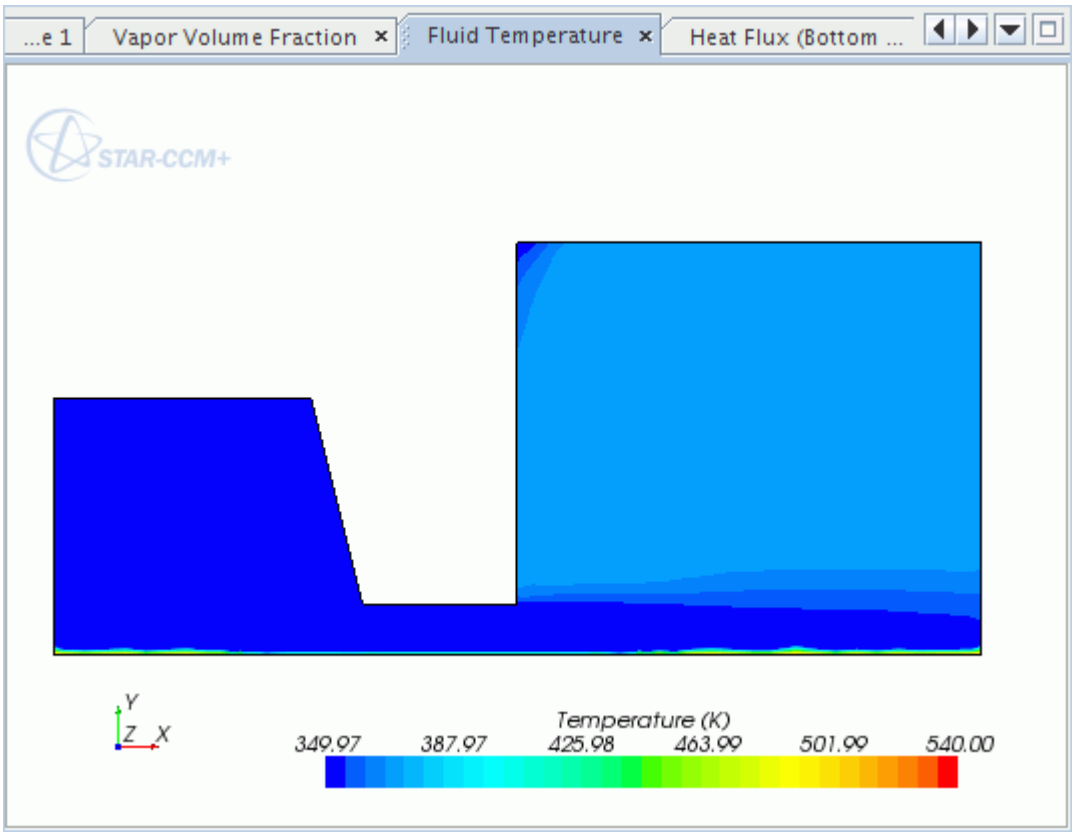
Visualizing Results

The scene tabs display the results of the simulation.

A vapor volume fraction scene similar to the one shown below is displayed.



A fluid temperature scene similar to the one shown below is displayed.



A heat flux profile similar to the one shown below is displayed.

From the above scenes, it is clear that the *C_qw* and *n_p* parameters have significant effects on the predicted levels of wall heat flux and the vapor volume fraction. In this example, you halved the heat flux on the bottom wall which resulted in the production of five times less water vapor. It is important therefore to make sure that correct values are used, to reflect the liquid-surface combination and the surface finish.

1. Save the simulation.

Summary

This tutorial demonstrated how to set up a boiling problem in STAR-CCM+.

This tutorial has introduced the following STAR-CCM+ features:

- Defining a model for boiling flow problems.
- Defining and modifying boiling model parameters.
- Setting initial conditions.
- Setting solver parameters for an unsteady run.
- Analyzing wall boiling results.

Bibliography

The following papers were referenced in this tutorial.

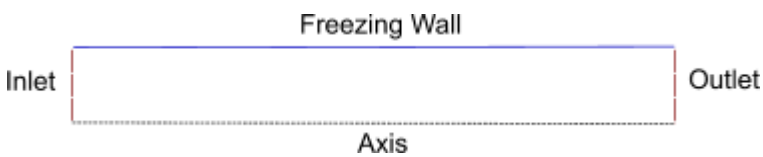
[791] *Rohsenow, W.M. 1952. "A Method of Correlation Heat Transfer Data for Surface Boiling of Liquid", Trans. ASME, 74, 969.*

[792] *Saiz Jabardo, J. M., Fockink da Silva, E., Ribatski, G. and de Barros, S. F. 2004. "Evaluation of the Rohsenow Correlation Through Experimental Pool Boiling of Halocarbon Refrigerants on Cylindrical Surfaces", J. of the Braz. Soc. of Mech. Sci. and Eng, April-June 2004, Vol. XXVI, No. 2.*

VOF: Melting-Solidification

This tutorial demonstrates how to set up a melting and solidification analysis in STAR-CCM+. It simulates the freezing of water in a pipe.

The geometry is shown in the following figure:



Water flows along a pipe with freezing walls. This scenario can be modeled as an axisymmetric two-dimensional pipe. The wall temperature is a function of position, with the middle of the wall being at the lowest temperature according to:

$$T_{wall} = \min\left(273.1,\ 2010.0(x - 0.15)^2 + 253.0\right) \tag{4163}$$

Water and Ice Properties

Property	Value
Solidus temperature	273.0 K
Liquidus temperature	273.0 K
Density of water and ice	997.561 kg/m^3 (No expansion)
Viscosity of H2O	8.8871e-4 Pa s
Latent heat of fusion	100000.0 J/kg
Specific heat	4181.72 J /kg K
Thermal conductivity	Step function in T

The simulation is run using the melting-solidification model. The post-processing tools in STAR-CCM+ are used to visualize the results.

Contents:

Prerequisites
Importing the Mesh and Naming the Simulation
Selecting the Physics Models
Defining the Liquid Phase for the Melting-Solidification Model
Setting the Initial Conditions

Setting Boundary Conditions
Setting Solver Parameters and Stopping Criteria
Creating a Mass Flow Monitor Plot
Setting Up Scenes to Visualize Results
Running the Simulation
Analyzing the Results
Summary

Prerequisites

The instructions in the VOF: Melting-Solidification tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Importing the Mesh and Naming the Simulation

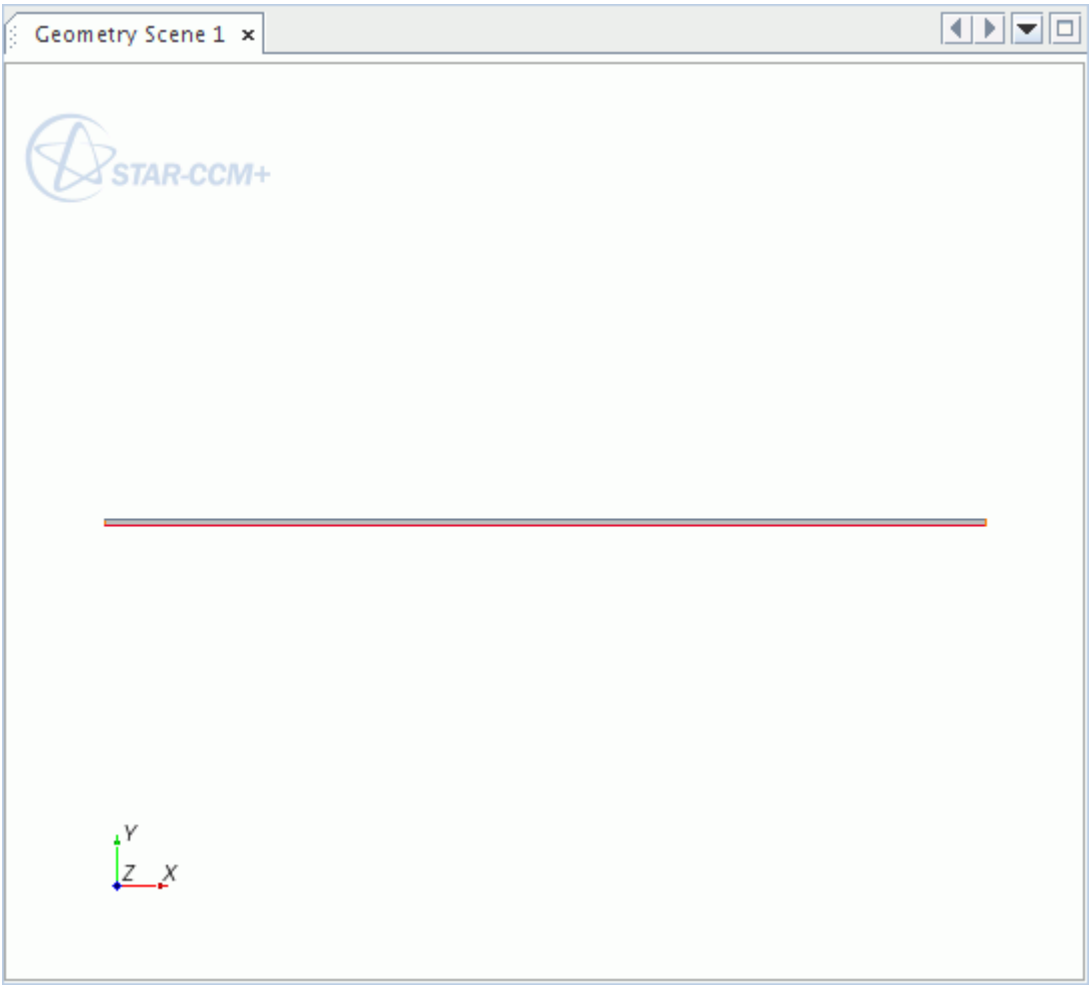
Launch STAR-CCM+, import the supplied geometry, and save the simulation.

A mesh file has been prepared for this tutorial.

To import the mesh and name the simulation:

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh...**.
4. In the *Open* dialog, choose `doc/startutorialsdata/multiphaseFlow/data/freezingWaterDuct.ccm`, then click **Open**.

   STAR-CCM+ provides feedback on the import process, which takes a few seconds, in the *Output* window. A geometry scene is created in the *Graphics* window.



5. Save the simulation as `freezingWaterDuct.sim`.

Selecting the Physics Models

Physics models define the primary variables of the simulation, including pressure, temperature, velocity, and what mathematical formulation is used to generate the solution.

In this case, the transient Reynolds-Averaged Navier Stokes equations are solved for using the K-Epsilon turbulence model and the Segregated Flow solver. The Volume of Fluid (VOF) model is activated before the melting and solidification option becomes available for Eulerian phases.

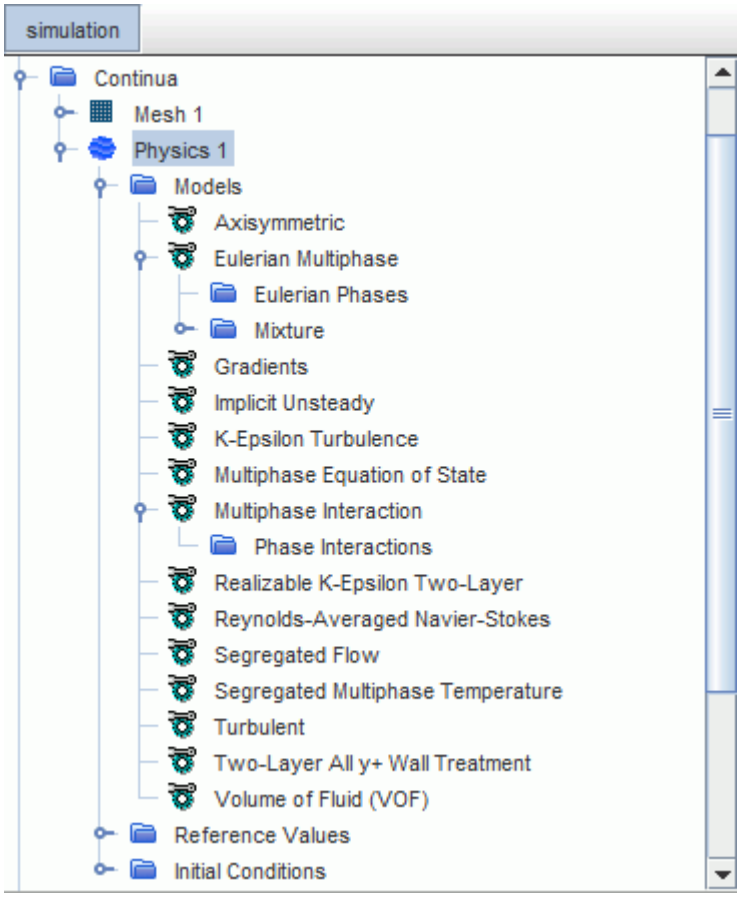
To select the physics models:

1. Right-click **Continua** > **Physics 1** and click **Select models...**.
2. In the *Enabled Models* group box, deactivate **Two-Dimensional**.
3. Select the following models in order:

Group Box	Model
Space	**Axisymmetric**
Time	**Implicit Unsteady**

Group Box	Model
Material	**Eulerian Multiphase** **Multiphase Interaction** (Selected automatically)
Eulerian Multiphase Model	**Volume of Fluid (VOF)** **Gradients** (Selected automatically) **Segregated Flow** (Selected automatically) **Multiphase Equation of State** (Selected automatically)
Viscous Regime	**Turbulent** **Reynolds-Averaged Navier-Stokes** (Selected automatically)
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence** **Realizable K-Epsilon Two-Layer** (Selected automatically) **Two-Layer All y+ Wall Treatment** (Selected automatically)
Optional Models	**Segregated Multiphase Temperature**

4. Click **Close**.
5. To review the models, open the **Physics 1** > **Models** node.



6. Save the simulation.

Defining the Liquid Phase for the Melting-Solidification Model

You define the liquid phase and apply the Melting-Solidification model.

You model the freezing water as a Eulerian phase in STAR-CCM+. The properties of water come from the material database, with some modifications to suit the problem. For this tutorial, the value of latent heat is reduced so that freezing takes place sooner than it would with the default value. For a real case, however, leave the value at its original setting.

To define the liquid phase for the Melting-Solidification model:

Create the H2O phase.

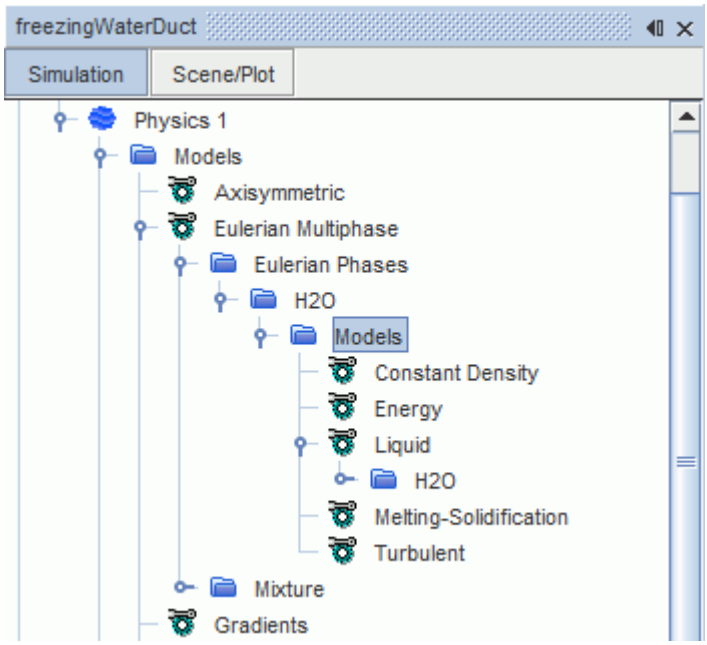
1. In the **Physics 1** continuum, right-click the **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and create a new phase.
2. Rename the **Phase 1** node to **H2O**.
3. For the **H2O** phase, select the following models in order:

Group Box	Model
Enabled Models	**Energy** (selected automatically) **Turbulent** (selected automatically)

Group Box	Model
Material	**Liquid**
Equation of State	**Constant Density**
Optional Models	**Melting-Solidification**

4. Click **Close**.

   The **Models** tree under **Eulerian Phases** appears as follows:



Set the solid volume fraction threshold above which flow is stopped in the fluid phase.

5. Edit the **Eulerian Phases** > **H2O** > **Models** node and set the following properties:

Node	Property	Setting
Melting-Solidification	**Flow Stop Relative Solid Fraction**	`0.999`
	Under-Relaxation Factor	`0.1`

Set the flow stop value in the **Melting-Solidification Flow Stop** node.

6. Edit the **Physics 1** > **Models** node and set the following properties:

Node	Property	Setting
Melting-Solidification Flow Stop	**Flow Stop Solid Fraction**	`0.999`

A step function defines this property: H2O in liquid state flows without any resistance in cells where the solid volume fraction is 0.999 or less. STAR-CCM+ sets the fluid velocity to zero for cells with solid volume fraction above the flow stop value.

The thermal conductivity of H2O varies with phase and temperature. For this simulation, assume that the thermal conductivity is constant for each phase.

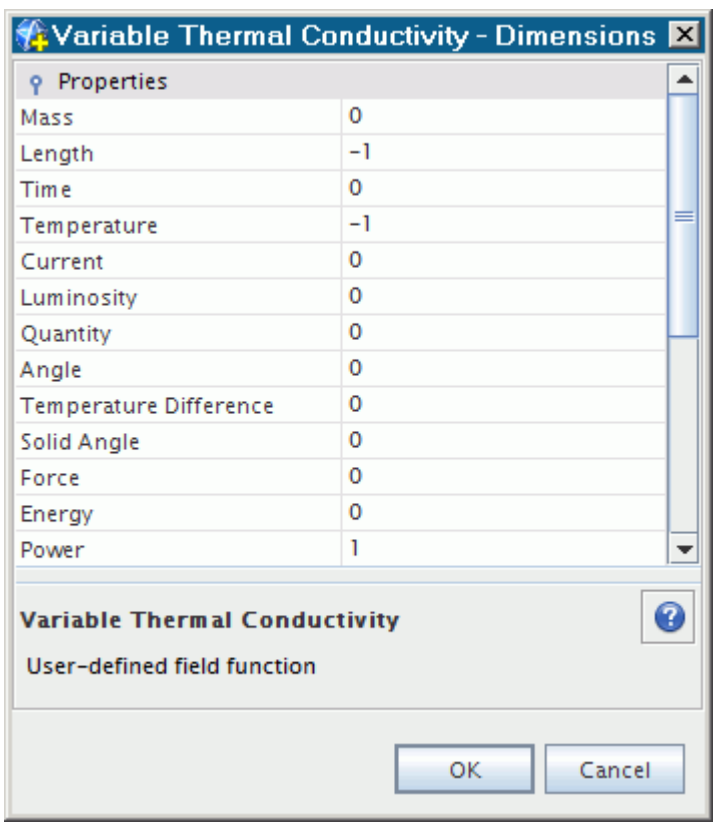
The thermal conductivity of ice is 2.33 W / m K while the conductivity of H2O is 0.620271 W/m K. Use a conditional expression to set the conductivity of the H2O phase. Set the conductivity to 2.33 W/m K when the temperature is less than or equal to 273 K, and 0.620271 W / m K for higher temperatures.

Create a step function to set the conductivity of H2O:

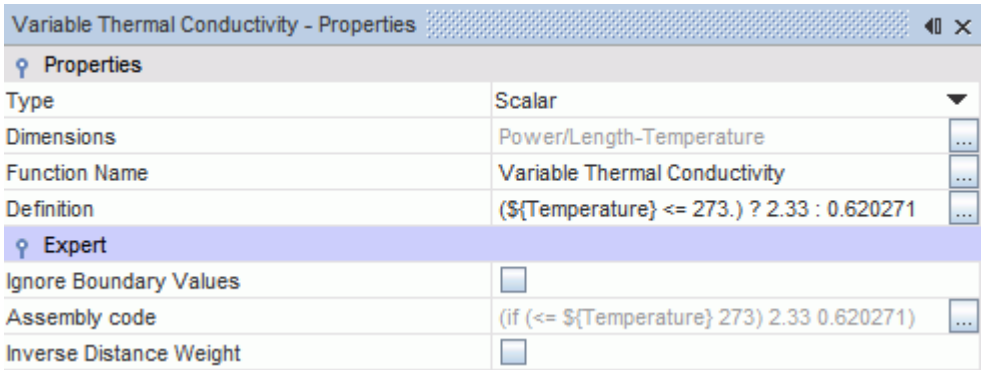
7. Right-click the **Tools** > **Field Functions** node and select **New** > **Scalar** from the pop-up menu.
8. Rename the **User Field Function 1** to `Variable Thermal Conductivity`.
9. Select the **Variable Thermal Conductivity** node and set the following properties:

Property	Setting
Function Name	`Variable Thermal Conductivity`
Definition	`(${Temperature} <= 273.) ? 2.33 : 0.620271`

10. Set **Dimensions** to **Power/Length-Temperature** by filling the *Dimensions* dialog with the following values:



The *Properties* window appears as shown.



Use this variable thermal conductivity for the H2O phase.

11. Edit the **Eulerian Phases** > **H2O** > **Models** > **Liquid** > **H2O** > **Material Properties** node and set the following properties:

Node	Property	Setting
Thermal Conductivity	**Method**	**Field Function**
Thermal Conductivity > **Field Function**	**Scalar Function**	**Variable Thermal Conductivity**
Latent Heat of Fusion > **Constant**	**Value**	`100000.0 J/kg`
Liquidus Temperature > **Constant**	**Value**	`273.0 K`
Solidus Temperature > **Constant**	**Value**	`273.0 K`

12. Save the simulation.

Setting the Initial Conditions

You set the initial conditions for the physics continuum.

A field function sets up the initial pressure field as a linear profile: this profile is close to the expected solution. Setting the initial pressure in this way reduces the time that is required to obtain a solution. The pressure profiles at the inlet and outlet are specified to define the linear pressure field correctly.

In this case, the inlet is not modeled as a velocity inlet, since the simulation diverges when the whole-cross section is frozen, and the flow stops. A combination of pressure outlets at both inlet and outlet allows the flow to be stopped completely in the duct without violating any boundary condition

When choosing initial conditions, it is important to select a field that is bounded by the boundary conditions to avoid divergence. To satisfy this criterion, specify a linear pressure field with maximum and minimum values that are set as the boundary conditions.

The initial temperature is set to the inlet temperature. The initial velocity field is set to be close to the expected average velocity during the initial stages of the flow, that is, before the freezing commences, for faster convergence.

To set the initial conditions:

1. Create the following field functions using the same procedure as in the previous section.

   The desired gauge inlet pressure is taken as 1.0 Pa.

Node Title	**Target Inlet Pressure**
Dimensions	`Pressure`
Function Name	`TargetInletPressure`
Definition	`1.0`

   As previously mentioned, the inlet is modeled as a pressure outlet: the dynamic pressure and turbulent stresses are added to make sure that the static inlet pressure is always 1.0 Pa.

Node Title	**Inlet Pressure**
Dimensions	`Pressure`
Function Name	`InletPressure`
Definition	`${TargetInletPressure} + 0.5*${Density}*pow(mag($${Velocity}), 2)`

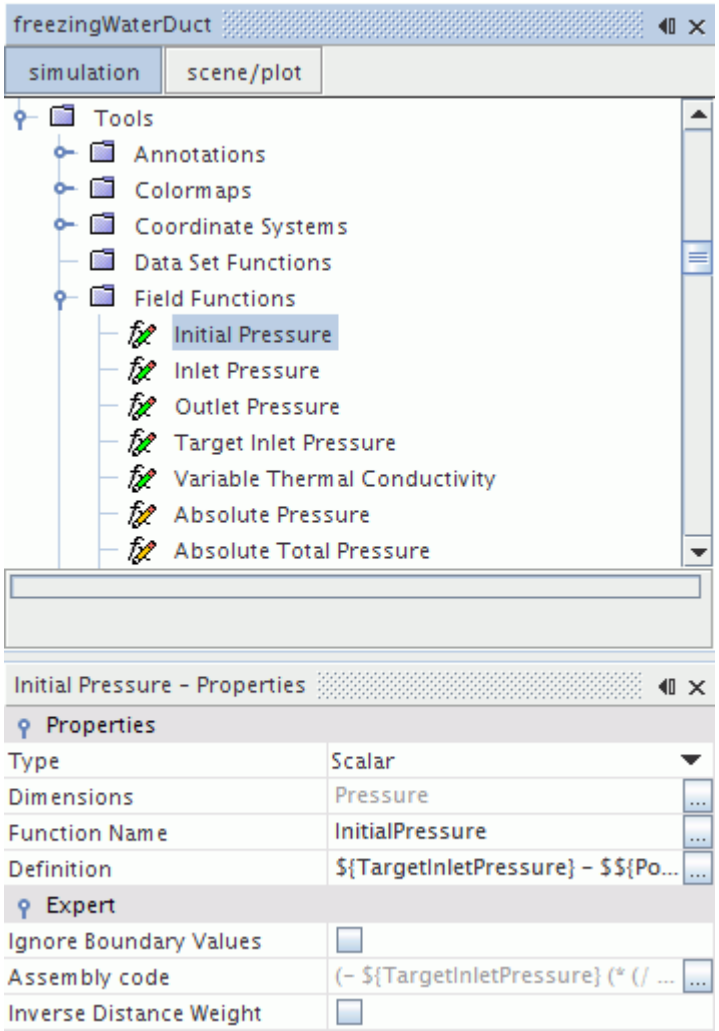
   The outlet pressure is set to 0.0 Pa.

Node Title	**Outlet Pressure**
Dimensions	`Pressure`
Function Name	`OutletPressure`
Definition	`0.0`

   The initial pressure field is then taken as a linear profile from 1.0 Pa at X = 0 m to 0.0 Pa at X = 0.3 m at the lower end of the pipe.

Node Title	**Initial Pressure**
Dimensions	`Pressure`
Function Name	`InitialPressure`
Definition	`${TargetInletPressure} - $${Position}[0]/0.3*(${TargetInletPressure} - ${OutletPressure})`

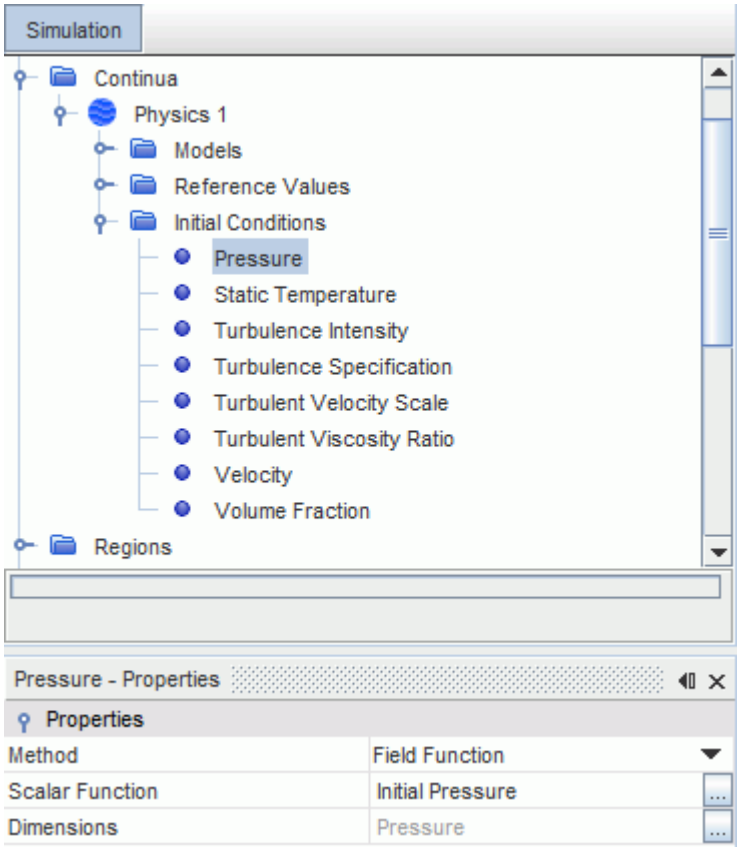
2. Right-click **Tools** > **Field Functions** and select **Refresh** to order the created field functions alphabetically.



Set the initial conditions for the solved variables in the **Physics 1** continuum.

3. Edit the **Physics 1** > **Initial Conditions** node and set the following properties:

Node	Property	Setting
Pressure	**Method**	**Field Function**
	Scalar Function	**Initial Pressure**
Static Temperature	**Value**	`273.1 K`
Velocity	**Value**	`[0.012, 0.0] m/s`
Volume Fraction	**Value**	`[1.0]`



4. Save the simulation.

Setting Boundary Conditions

You define the boundary types and specify the appropriate property values. Set the boundary conditions for the inlet, outlet, and pipe wall.

To set the boundary conditions:

1. Expand the **Regions** > **Tube_2D** > **Boundaries** node to reveal the different boundaries.

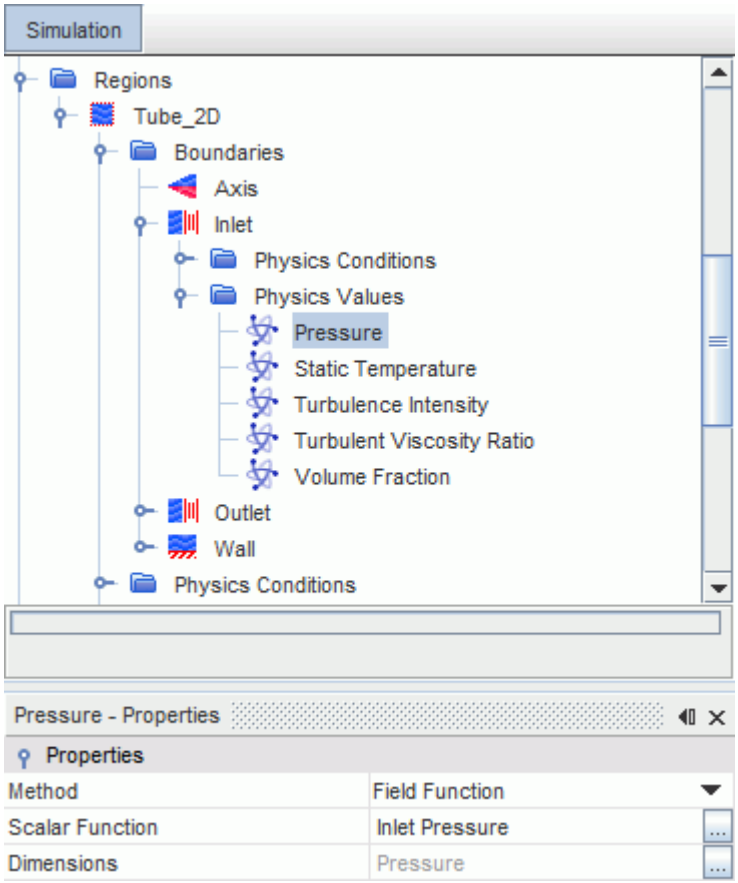
Start with the Inlet boundary conditions. The inlet temperature is 273.1 K and the volume fraction is 1.0.

The dynamic pressure and contribution to stresses due to turbulence are removed from the specified pressure at a pressure outlet in case of reverse flow. The inlet pressure profile is therefore specified as: $1.0 + \frac{1}{2}\rho u^2$ as specified in the Inlet Pressure field function. This setting is to make sure that the static inlet pressure is always effectively 1.0 Pa when water flows into the pipe.

2. Edit the **Inlet** > **Physics Values** node and set the following properties:

Node	Property	Setting
Pressure	**Method**	**Field Function**
	Scalar Function	**Inlet Pressure**
Static Temperature	**Value**	`273.1 K`

Node	Property	Setting
Volume Fraction	**Value**	[1.0]



The outlet pressure is set to atmospheric pressure, that is, 0 Pa gauge pressure. However, a field function is already defined for outlet pressure in the previous section and is reused here. The advantage of this method is: if you must change the pressure profile across the whole domain, you alter the outlet pressure value in only one place, rather than two.

3. Edit the **Outlet** > **Physics Values** node and set the following properties:

Node	Property	Setting
Pressure	**Method**	**Field Function**
	Scalar Function	**Outlet Pressure**
Static Temperature	**Value**	273.1 K
Volume Fraction	**Value**	[1.0]

Specify the freezing wall boundary conditions.

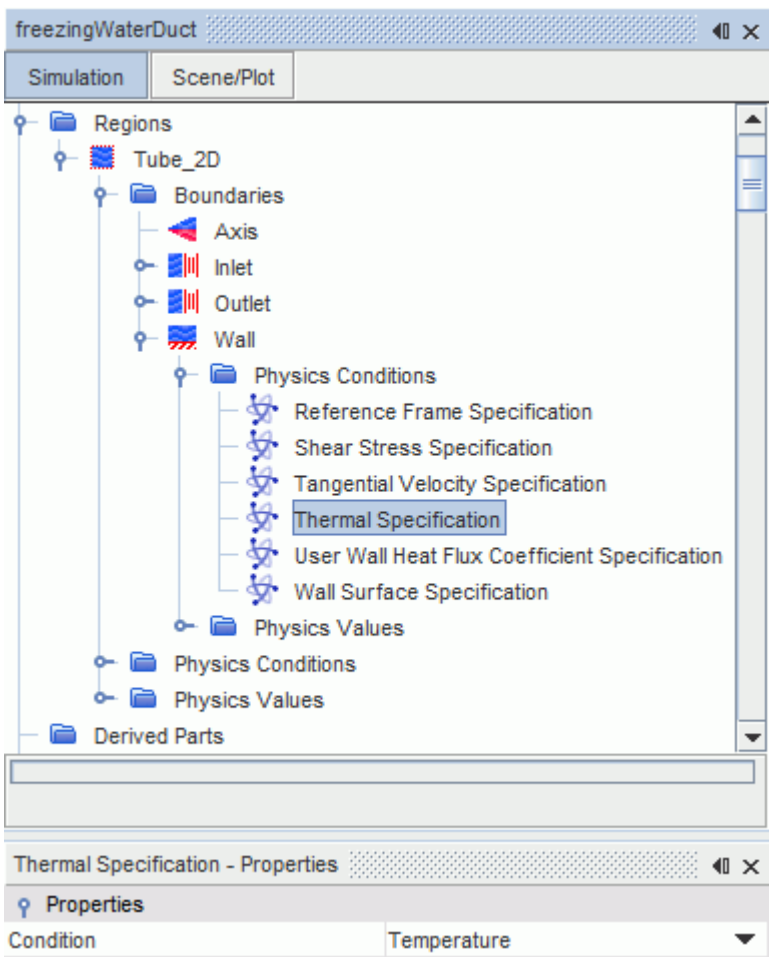
4. Create the following field function using the same procedure as in the previous sections.

Node Title	Wall Temperature

Dimensions	`Temperature`
Function Name	`WallTemperature`
Definition	`min(273.1, 2010.0*pow($${Position}[0] - 0.15, 2) + 253.0)`

5. Edit the **Wall** boundary and set the following properties:

Node	Property	Setting
Physics Conditions > **Thermal Specification**	**Condition**	**Temperature**
Physics Values > **Static Temperature**	**Method**	**Field Function**
	Scalar Function	**Wall Temperature**



STAR-CCM+ applies a no-slip boundary condition to a wall by default, which is what is required in this simulation.

6. Save the simulation.

Setting Solver Parameters and Stopping Criteria

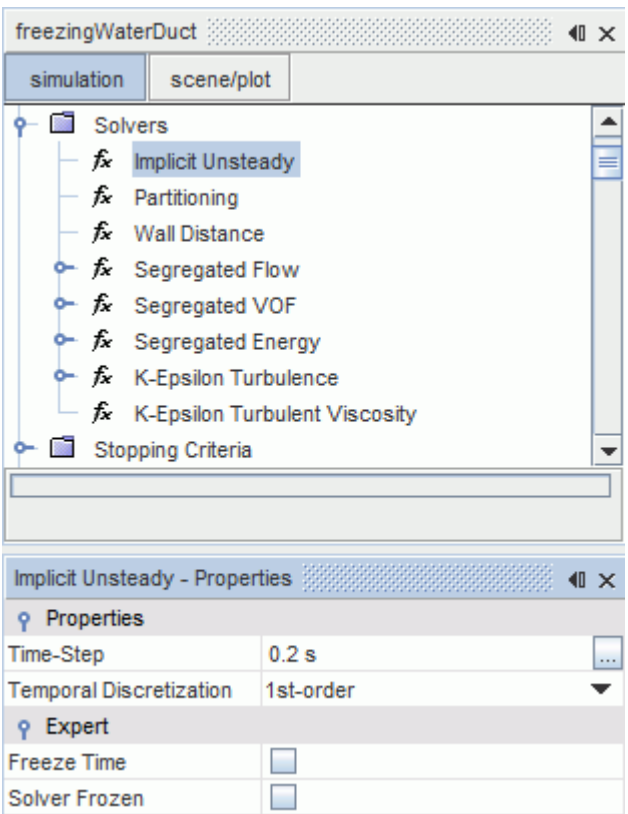
Set the appropriate solver parameters and stopping criteria for the simulation. You specify the time-step and the maximum number of iterations per time step.

The flow stop relative solid fraction property for melting and solidification is implemented as a step function. Therefore, cells containing H2O in liquid state require a few inner iterations to realize that velocity is zero in the newly frozen cells. This process is dynamic and many cells can be ‘frozen’ at subsequent iterations during one time-step. Therefore, many inner iterations are required to make sure that results are accurate.

To set the solver parameters and stopping criteria:

1. Edit the **Solvers** node and set the following properties:

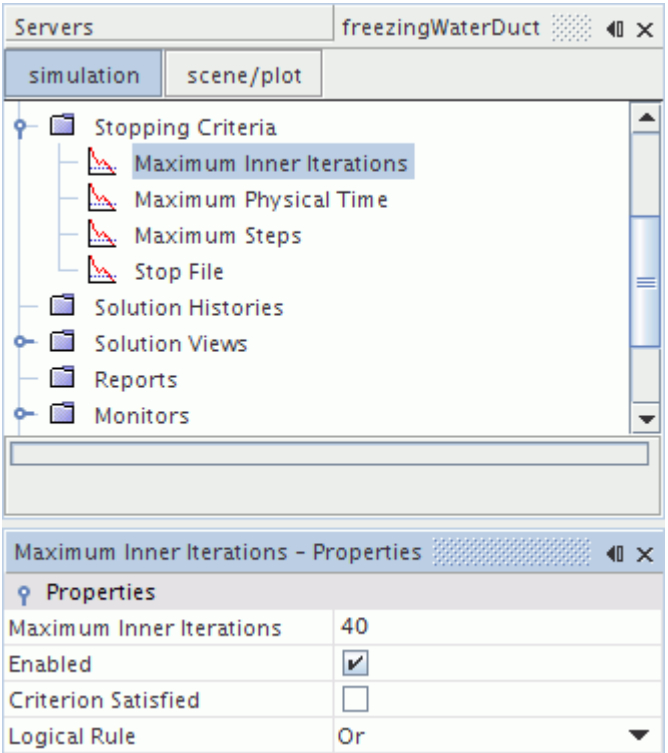
Node	Property	Setting
Implicit Unsteady	**Time-Step**	`0.2 s`
Segregated Energy	**Fluid Under-Relaxation Factor**	`0.5`



2. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Inner Iterations	**Maximum Inner Iterations**	`40`
Maximum Steps	**Enabled**	Deactivate

Node	Property	Setting
Maximum Physical Time	**Enabled**	Activated
	Maximum Physical Time	20.0 s



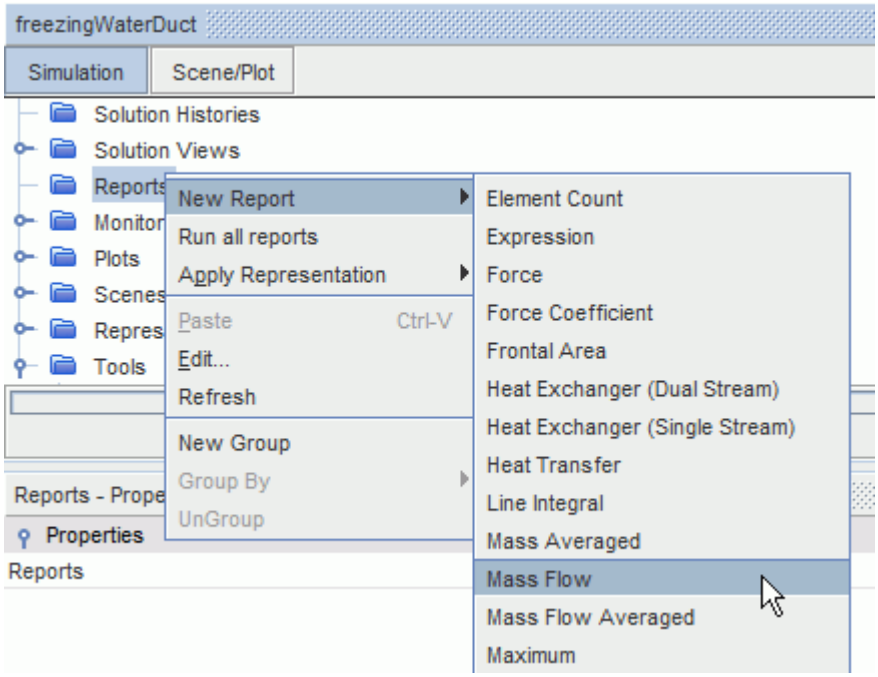
3. Save the simulation.

Creating a Mass Flow Monitor Plot

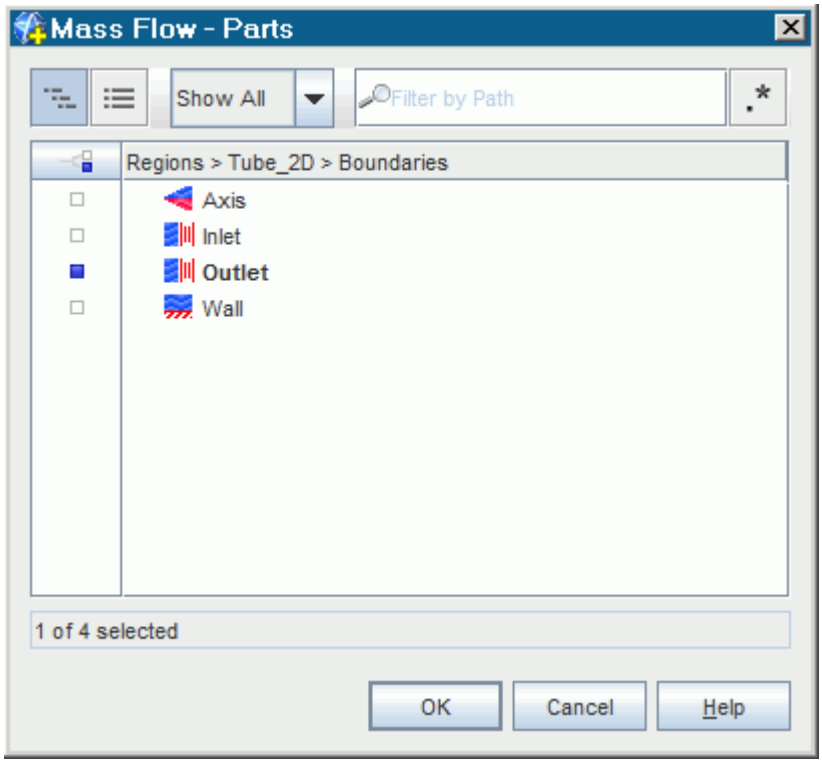
You create a monitor plot for normalized mass flow through the outlet.

To create a monitor plot for normalized mass flow through the outlet:

1. Right-click **Reports** and select **New Report** > **Mass Flow**.

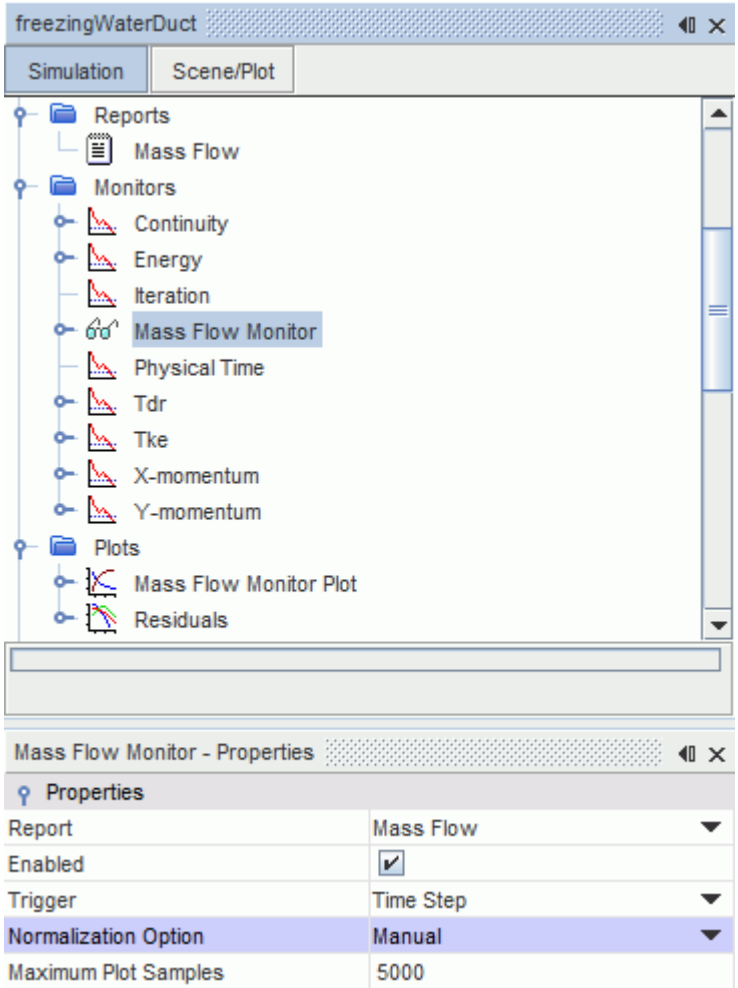


2. Rename the **Mass Flow 1** report to **Mass Flow**.
3. Select the **Mass Flow** node and set the *Parts* to **Tube_2D** > **Outlet**.
4. Select **Outlet** from the dialog which appears when editing the *Parts* property value.



5. Right-click **Reports** > **Mass Flow** and select **Create Monitor and Plot from Report**.
6. Select **Monitors** > **Mass Flow Monitor**.

7. Set *Normalization Option* to **Manual**.
8. Make sure that the *Trigger* property is set to **Time Step**.



9. Expand the **Mass Flow Monitor** node and select **Manual Normalization**.

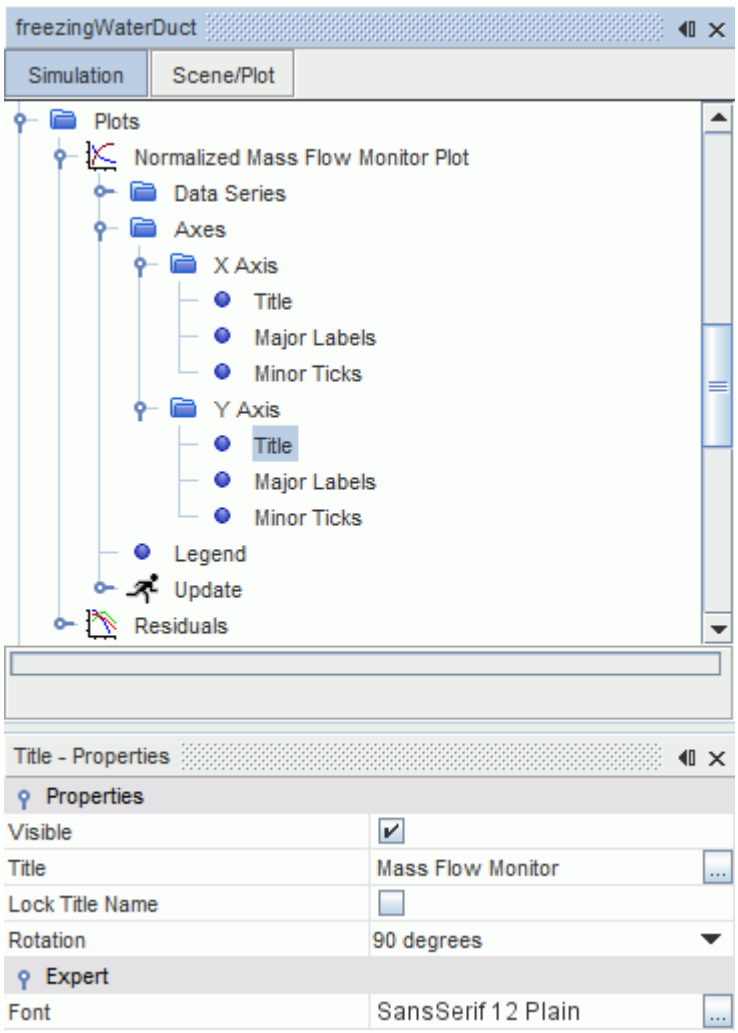
   The normalization factor is chosen as the mass outflow for the first iteration.
10. Set *Normalization Factor* to `3.74E-5`.

Edit the plot graphics settings.

11. Rename **Plots** > **Mass Flow Monitor Plot** to **Normalized Mass Flow Monitor Plot**.
12. Set the following properties:

Node	Property	Setting
Normalized Mass Flow Monitor Plot	*Title*	**Normalized Mass Flow Monitor Plot**
└ **Axes** > **Bottom Axis**	*Minimum*	`0.0`
	Lock Minimum	Deactivated
	Maximum	`20.0`

Node	Property	Setting
	Lock Maximum	Activated
└ **Title**	*Title*	`Physical Time (s)`
└ **Axes** > **Left Axis**	*Minimum*	`0.0`
	Lock Minimum	Deactivated
	Maximum	`1.0`
	Lock Maximum	Activated
└ **Title**	*Title*	`Mass Flow Monitor`



13. Save the simulation.

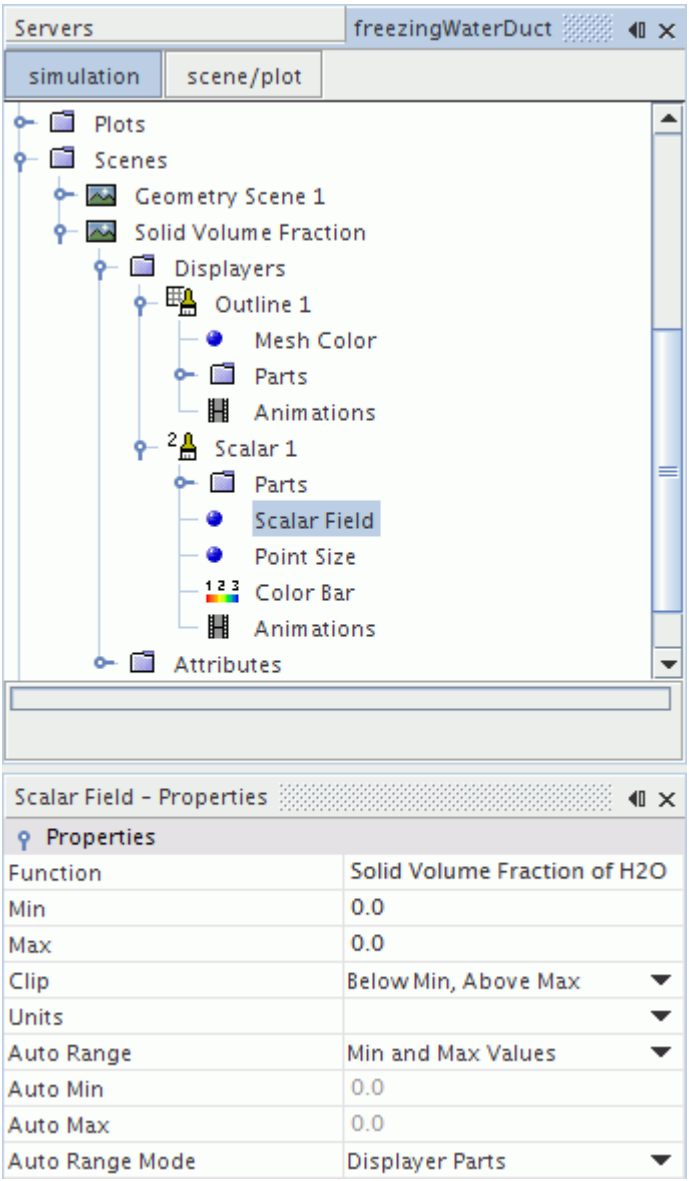
Setting Up Scenes to Visualize Results

Create scalar scenes to visualize the results as the simulation progresses.

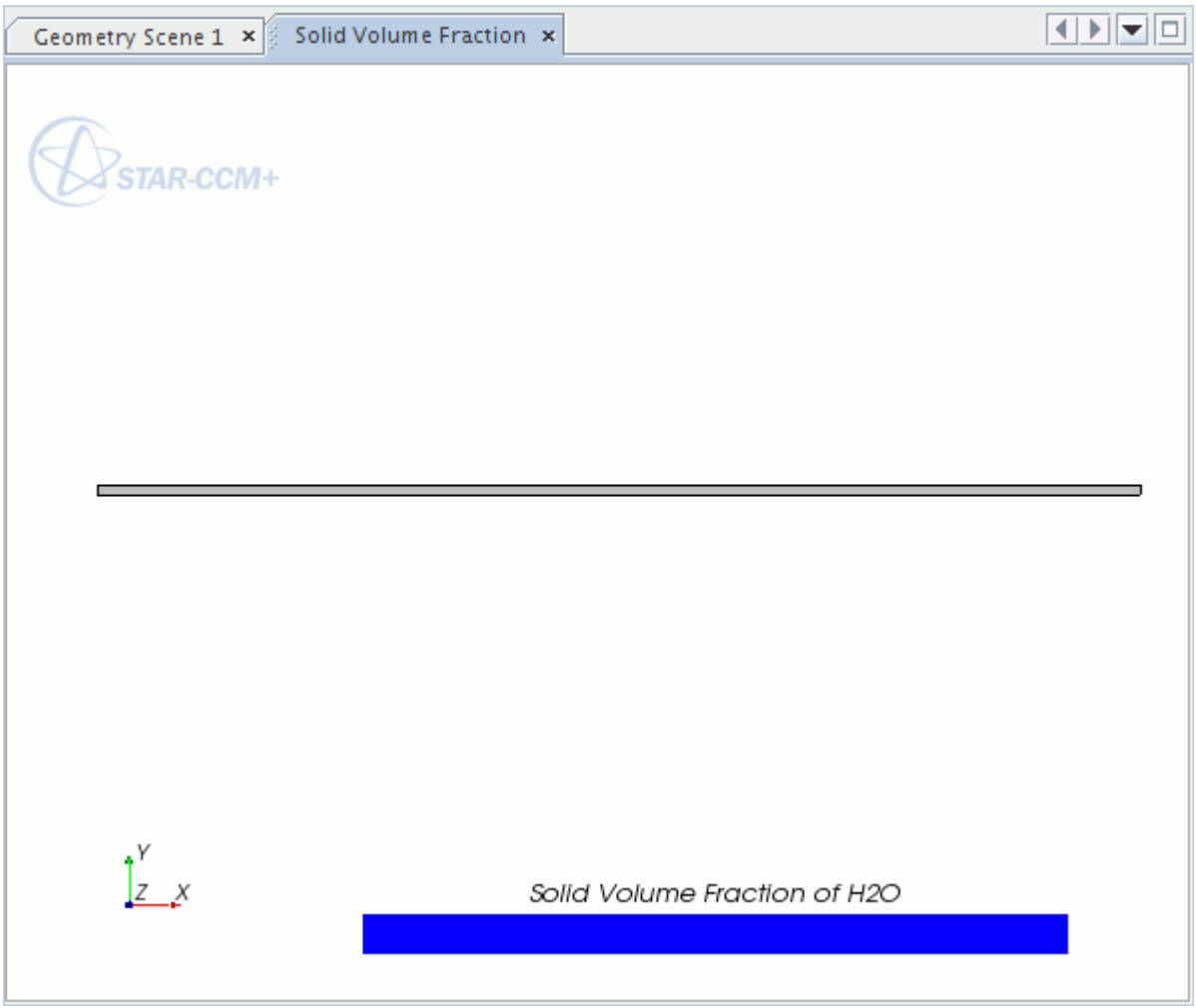
To create scalar scenes to visualize the results:

1. Right-click **Scenes** and select **New Scene** > **Scalar**.
2. Rename **Scalar Scene 1** to **Solid Volume Fraction**.

   The solid volume fraction function represents the state of the H2O phase in each cell. When solid volume fraction is 1.0, ice is present in the cell, and when solid volume fraction is 0.0, (liquid) water is present. Do not confuse solid volume fraction with volume fraction which represents the phase in the cell. Volume fraction is always 1.0 in this case, since there is only one phase in this simulation: H2O.
3. Select **Scenes** > **Solid Volume Fraction** > **Displayers** > **Scalar 1** > **Scalar Field** and set *Function* to **Solid Volume Fraction** > **Solid Volume Fraction of H2O**.

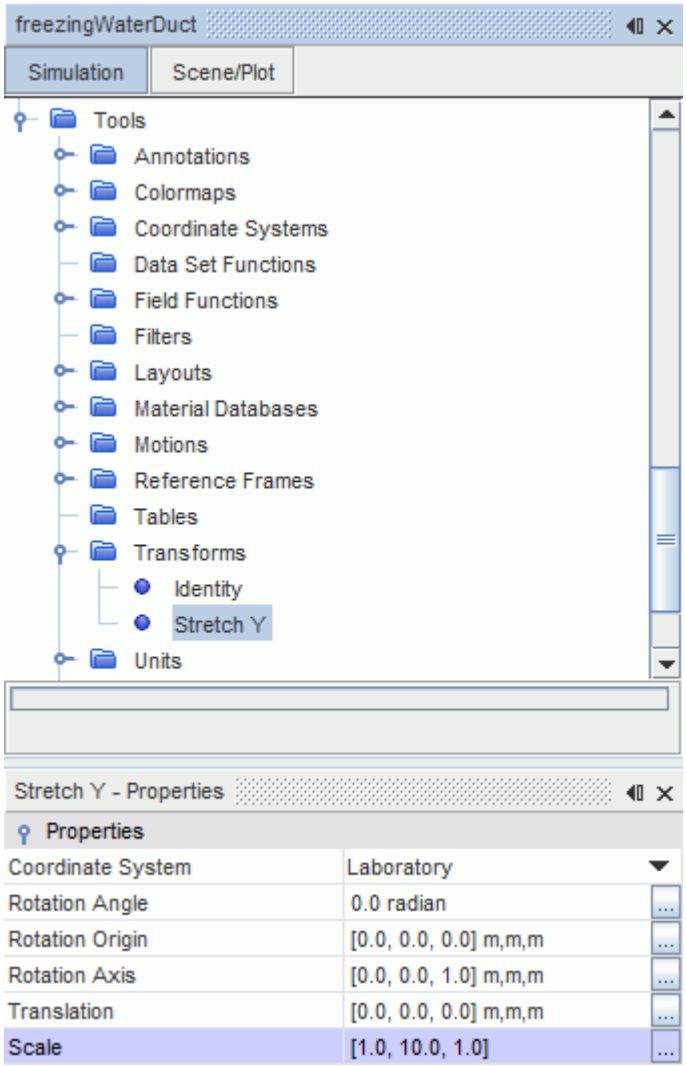


The domain is long in X and narrow in Y: this aspect ratio makes it hard to visualize any scalar field.



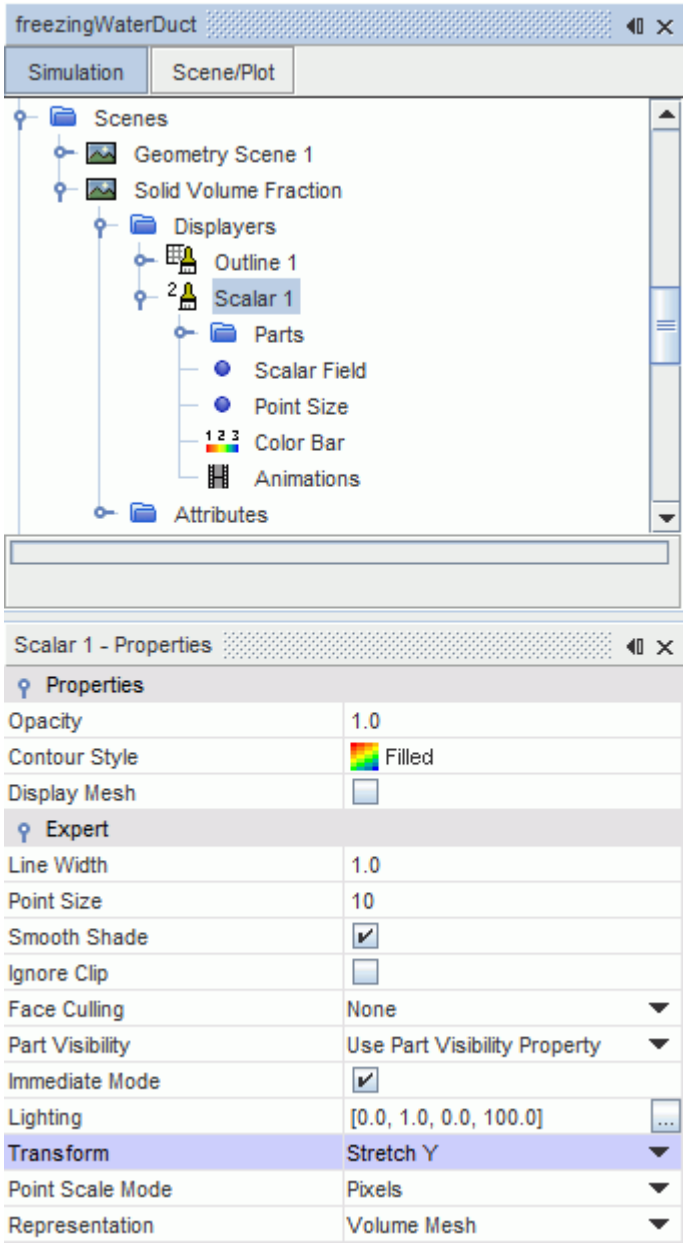
Apply a simple transform to scale the Y axis for a better solution representation.

4. Right-click **Tools** > **Transforms** and select **New Graphics Transform** > **Simple Transform**.
5. Rename the **Simple Transform 1** node to **Stretch Y**.
6. Set *Scale* to `[1.0, 10.0, 1.0]`.



Apply the **Stretch Y** transform to the **Displayers** in the **Solid Volume Fraction** scene.

7. Select **Scenes** > **Solid Volume Fraction** > **Displayers** > **Outline 1** and set *Transform* to `Stretch Y`.
8. Select **Scenes** > **Solid Volume Fraction** > **Displayers** > **Scalar 1** and set *Transform* to `Stretch Y`.



Create more scenes to display pressure and velocity.

9. Click **Scenes** > **Solid Volume Fraction** and while holding the left mouse button, drag the **Solid Volume Fraction** node onto the **Scenes** node and rename the new node as **Pressure**.
10. Select **Scenes** > **Pressure** > **Displayers** > **Scalar 1** > **Scalar Field** and set *Function* to **Pressure**.
11. Save the simulation.

Running the Simulation

Preparation of the simulation is now complete, and the simulation can be run.

To run the simulation:

1. Click (**Run**) in the *Solution* toolbar.

   The solution progress is displayed in the *Output* window. The *Residuals* display is created automatically in the *Graphics* window and shows the progress of the solvers.

   During the run, you can stop the process by clicking (**Stop**) on the toolbar. If you do stop the simulation, you can click (**Run**) to resume it. If left alone, the simulation continues until 10 seconds of physical time is complete.

   While the simulation is running, you can click the tabs at the top of the *Graphics* window to view scenes and plots.

You can check the *Normalized Mass Flow Monitor* while the solution is running.

2. Double-click **Plots** > **Normalized Mass Flow Monitor Plot** to display the results in the *Graphics* window.

   When the last time-step is complete, the following message is displayed in the *Output* window:

   ```
   Stopping criterion Maximum Physical Time satisfied.
   ```

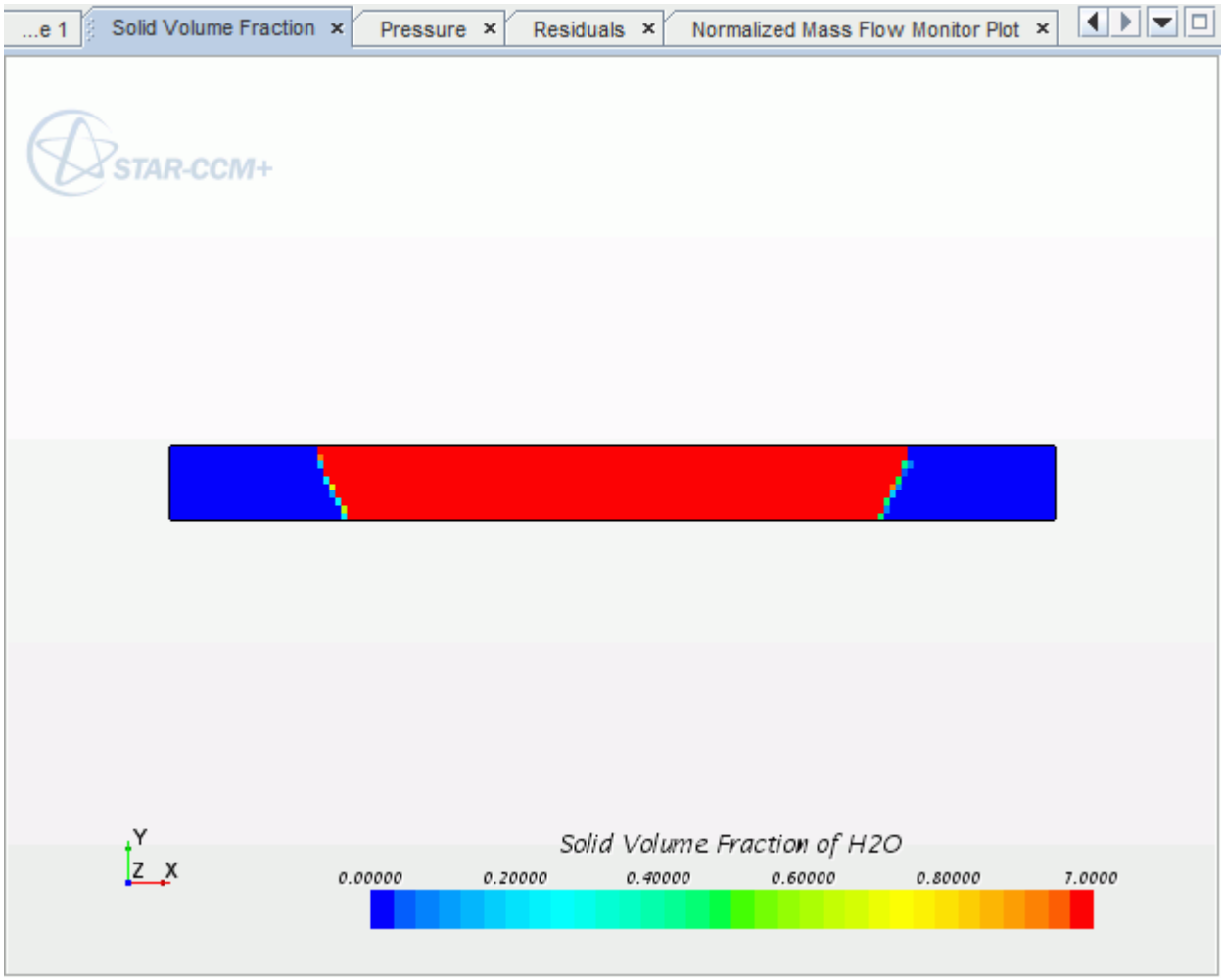
3. When the simulation has finished running, save it.

Analyzing the Results

View and examine the results of the simulation.

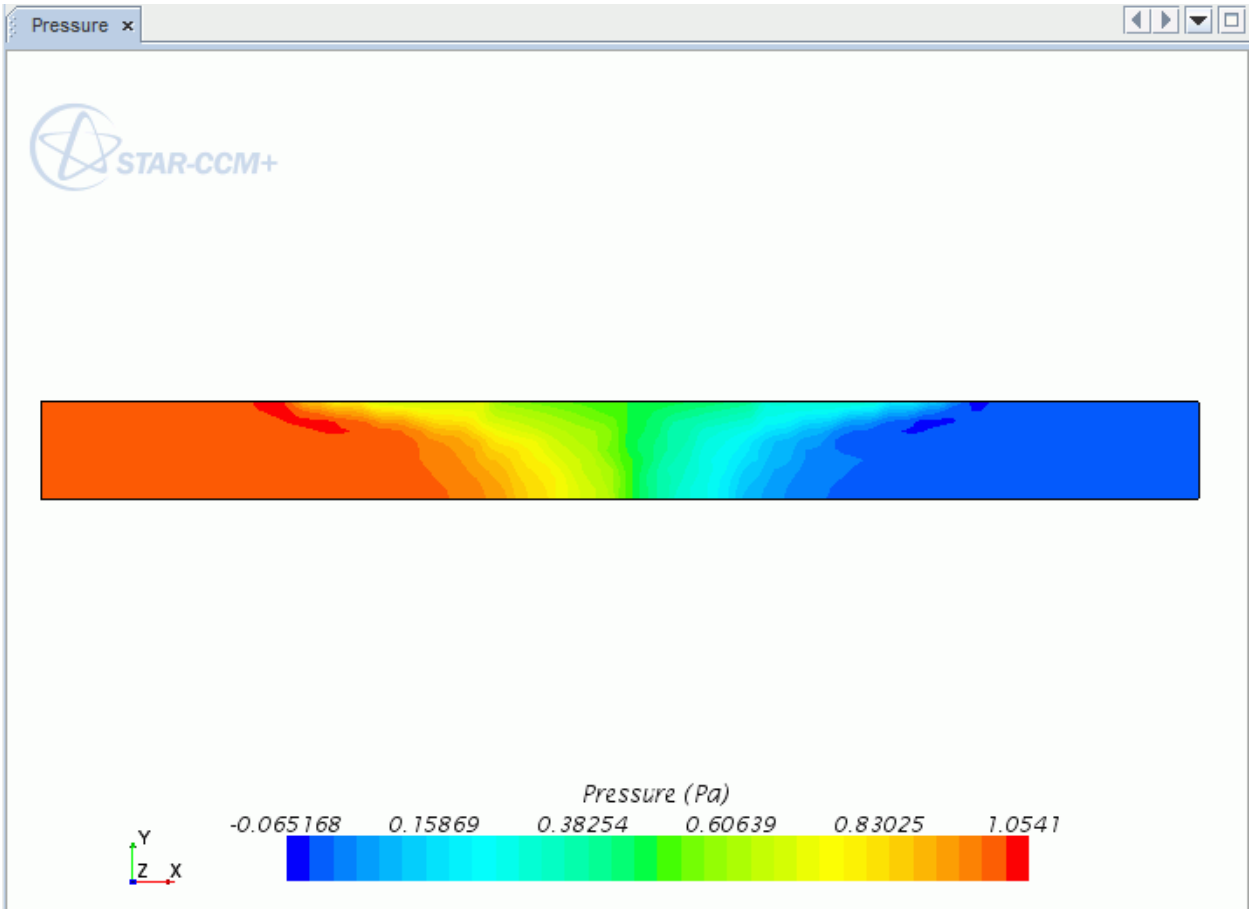
The freezing starts in the middle of the top wall. The water in the whole pipe eventually solidifies as time progresses.

1. Click the *Solid Volume Fraction* tab in the *Graphics* window.



The velocity in the frozen region must be zero since to the *Flow Stop Relative Solid Fraction* property is set to a value less than one.

2. Click the *Pressure* tab in the *Graphics* window.



3. Save the simulation.

Summary

This tutorial demonstrated how to set up a melting-solidification problem in STAR-CCM+.

This tutorial has introduced the following STAR-CCM+ features:

- Defining models for melting-solidification problems.
- Defining material properties that are required for the selected models.
- Setting solver parameters for a transient run.
- Plotting a normalized mass flow monitor.
- Running the solver to a specified maximum physical time.
- Displaying the results using the built-in visualization facilities.

Lagrangian: Particle-Laden Flow

This tutorial demonstrates how to set up a simple Lagrangian multiphase analysis in STAR-CCM+. It simulates a particle-laden flow of air through a partially blocked elbow duct.

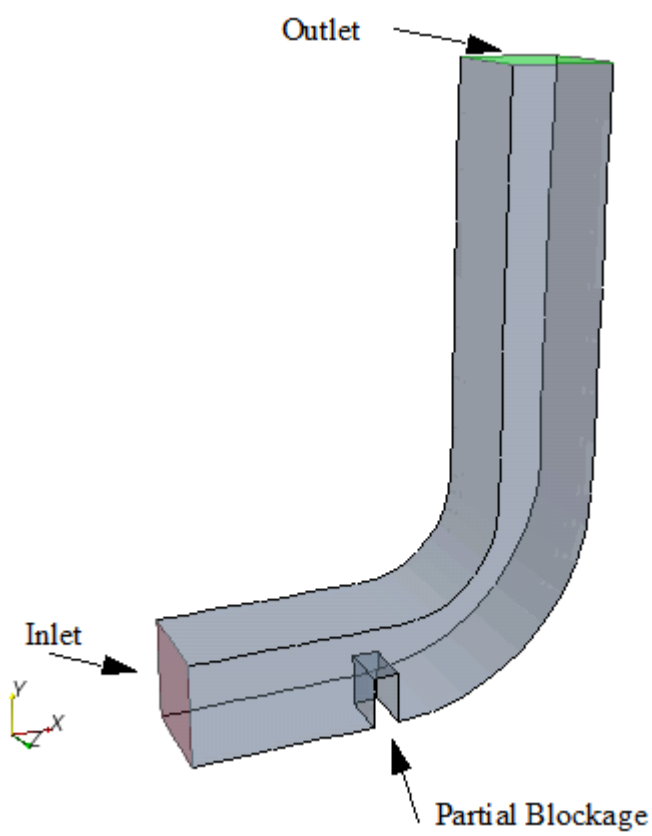
Air at standard pressure (1 atm) enters the channel at a velocity of 10 m/s. The fluid, having passed a partial blockage and a 90 degree elbow, exits vertically through the outlet. All fluid properties are assumed to be constant.

The air stream is seeded with solid particles, uniformly distributed at the duct inlet. The volumetric loading of particles in the incoming air is 0.01%, corresponding to a volumetric particle flow rate of 6.4516 x 10^{-7} m^3 /s. The particle properties are as follows:

- Density 1.2 x 10^3 kg/m^3
- Diameter 4 x 10^{-5} m
- Initial velocity (10, 0, 0) m/s

Adiabatic and no-slip conditions are assumed at the duct walls. Turbulence is modeled using the default K-Epsilon model. All particles rebound perfectly from the walls and gravity effects are neglected.

The geometry that is used in the tutorial is shown below.



Contents:

Prerequisites
Importing the Mesh and Naming the Simulation
Visualizing the Imported Geometry
Selecting the Physics Models
Setting Continuous Phase Material Properties
Selecting the Lagrangian Phase Models
Setting the Lagrangian Phase Model Properties
Setting the Lagrangian Phase Boundary Conditions
Setting the Continuous Phase Boundary Conditions
Setting Up the Injector
Setting Solver Parameters and Stopping Criteria
Visualizing the Solution
Initializing the Solution and Running the Simulation
Visualizing the Results
Displaying Particle Tracks
Using Tracks in Field Functions and Derived Parts
Summary

Prerequisites

The instructions in the Lagrangian: Particle-Laden Flow tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Importing the Mesh and Naming the Simulation

Launch STAR-CCM+, import the supplied geometry, and save the simulation.

A three-dimensional hexahedral cell mesh has been prepared for this analysis. The grid has been saved in the .ccmg file format which contains all the required cell, vertex, and boundary information for the problem geometry.

To import the mesh and name the simulation:

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh**.

4. In the *Open* dialog, choose `doc/startutorialsdata/multiphaseFlow/data/elbow.ccmg`, then click **Open**.

   STAR-CCM+ provides feedback on the import process, which takes a few seconds, in the *Output* window. One mesh region named **Default_Fluid** is created in the **Regions** node representing the grid domain. A geometry scene is created in the *Graphics* window.

5. Save the simulation as `obstructedElbow.sim`.

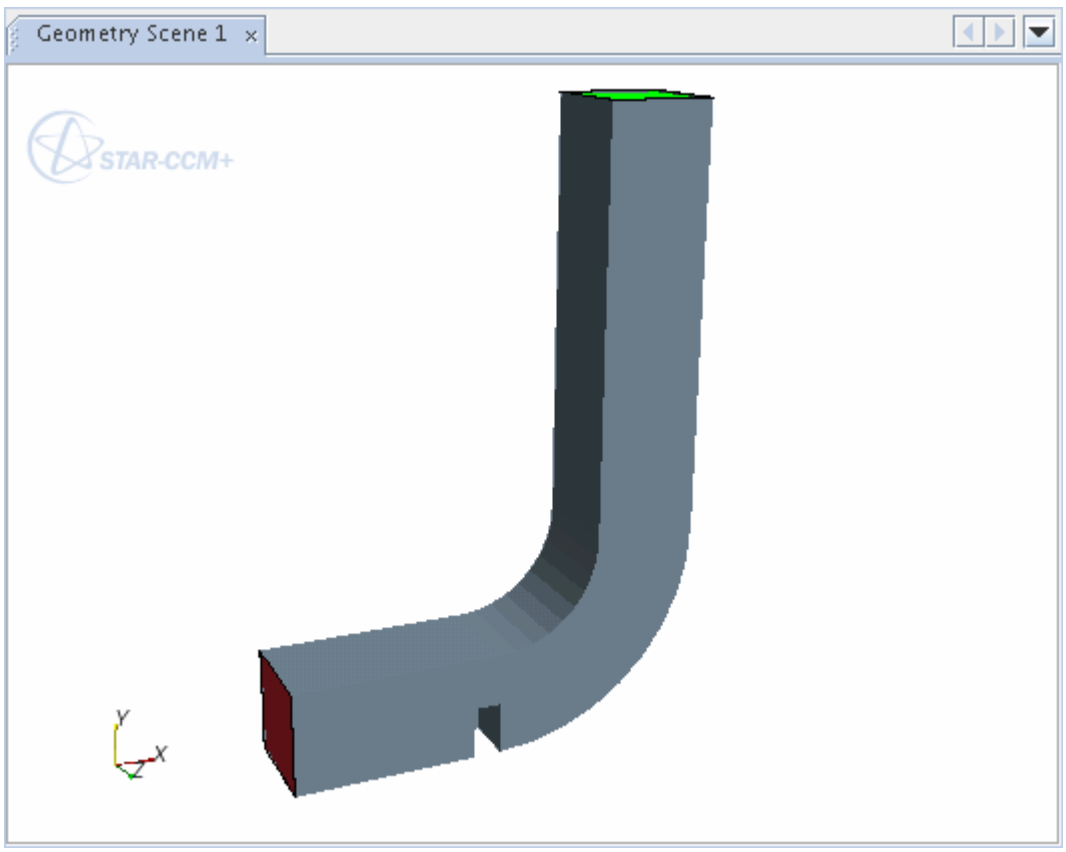
Visualizing the Imported Geometry

Examine the imported geometry.

Examine the *Geometry Scene 1* display in the *Graphics* window.

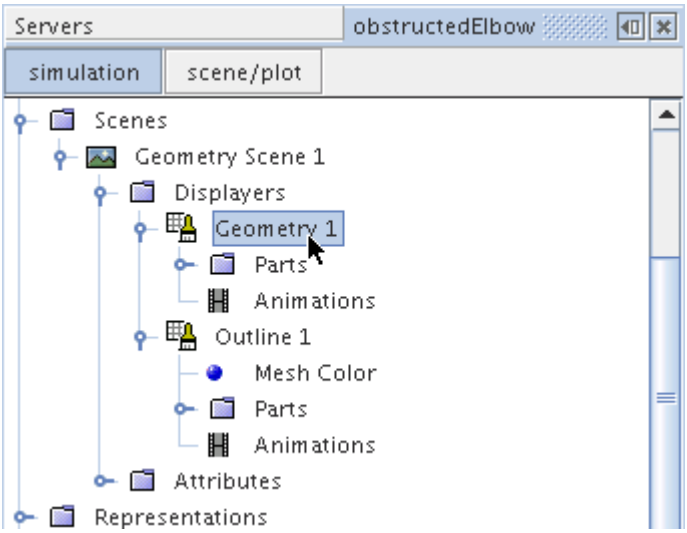
To visualize the imported geometry:

1. Use the mouse controls to adjust the view as shown below.

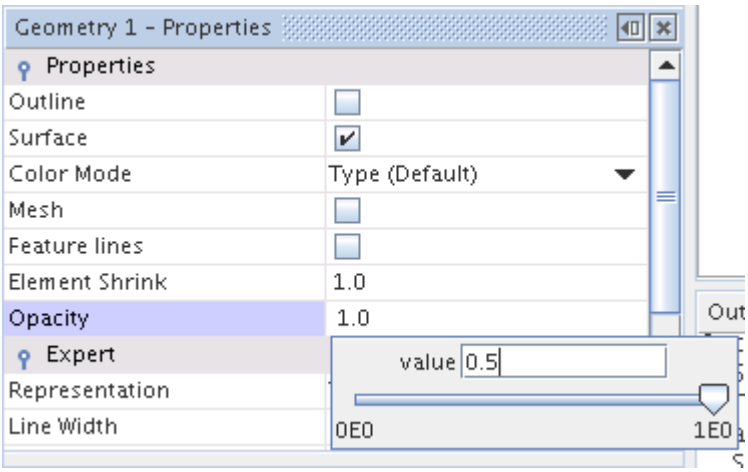


To make the interior features of the geometry visible, change the opacity of its surfaces.

2. Open the **Scenes** > **Geometry Scene 1** > **Displayers** node, then select the **Geometry 1** node.



3. Set the *Opacity* to `0.5`.



The obstruction in the elbow geometry is now clearly visible.

4. Save the simulation.

Selecting the Physics Models

Physics models define the primary variables of the simulation, including pressure, temperature, velocity, and what mathematical formulation is used to generate the solution.

In this simulation, the flow is turbulent and incompressible. The Segregated Flow model is used together with the default K-Epsilon Turbulence model. The Lagrangian Multiphase model is used to model the dispersed phase.

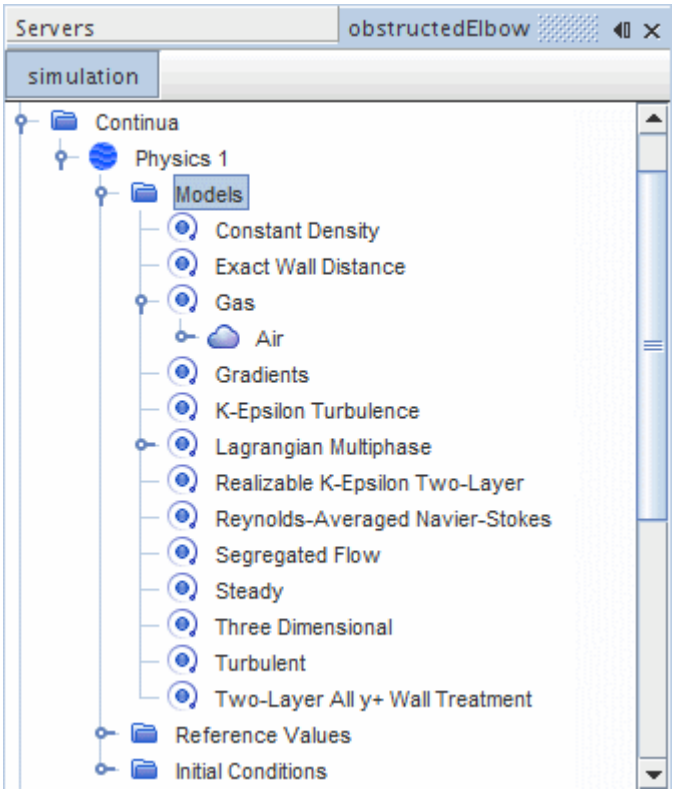
Only one region for the continuous phase, air, is required for the analysis. By default, the continuum **Physics 1** is automatically created when the mesh is imported. Edit the continuum so that appropriate physical models are selected for the air fluid region.

To select the physics models:

1. For the physics continuum, **Continua** > **Physics 1**, select the following models in order:

Group Box	Model
Enabled Models	**Three Dimensional** (Pre-selected)
Time	**Steady**
Material	**Gas**
Flow	**Segregated Flow**
Gradient Metrics	**Gradients** (Selected automatically)
Equation of State	**Constant Density**
Viscous Regime	**Turbulent** **Reynolds-Averaged Navier-Stokes** (Selected automatically)
Reynolds Averaged Turbulence	**K-Epsilon Turbulence** **Realizable K-Epsilon Two-Layer** (Selected automatically) **Two-Layer All y+ Wall Treatment** (Selected automatically) **Exact Wall Distance** (Selected automatically)
Optional Models	**Lagrangian Multiphase**

2. Click **Close**.
3. To review the models, open the **Physics 1** > **Models** node.



4. Save the simulation.

Setting Continuous Phase Material Properties

You set the material properties of the continuous phase. The default material for the continuous phase is air and, for this tutorial, is considered incompressible. You update the default material properties for air so that it has a density of 1.205 kg/m^3 and a dynamic viscosity of 1.82E-5 Pa-s.

To set the continuous phase material properties:

1. Edit the **Physics 1** > **Models** > **Gas** > **Air** > **Material Properties** node and set the following properties:

Node	Property	Setting
Density > **Constant**	**Value**	`1.205 kg/m^3`
Dynamic Viscosity > **Constant**	**Value**	`1.82E-5 Pa-s`

2. Save the simulation.

Selecting the Lagrangian Phase Models

You create the Lagrangian phase and select the appropriate phase models. These models represent the characteristics of the Lagrangian phase.

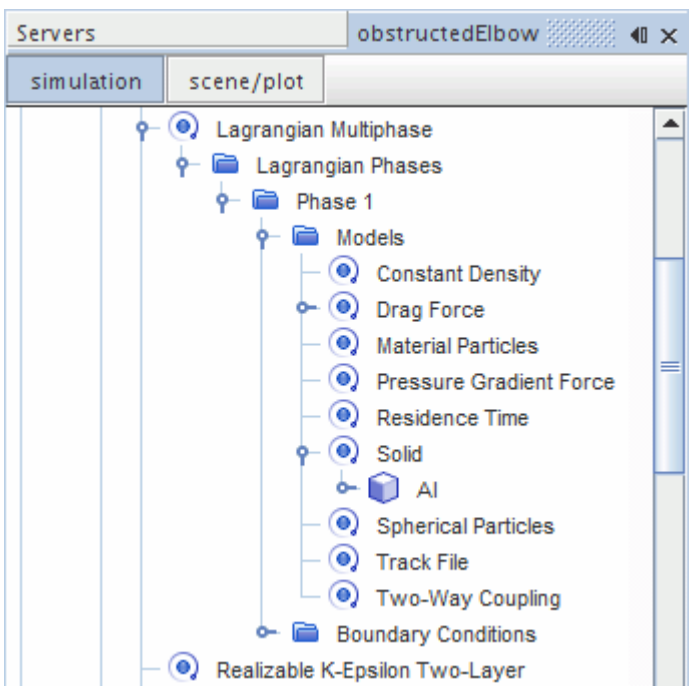
To create the Lagrangian phase and select the phase models:

1. In the **Physics 1** continuum, right-click the **Models** > **Lagrangian Multiphase** > **Lagrangian Phases** node and create a new phase.
2. For the **Phase 1** phase, select the following models:

Group Box	Model
Enabled Models	**Residence Time** (Pre-selected)
Particle Type	**Material Particles** **Pressure Gradient Force** (Selected automatically) **Spherical Particles** (Selected automatically)
Material	**Solid**
Equation of State	**Constant Density**
Optional Particle Forces	**Drag Force**
Track Sampling	**Track File**
Optional Models	**Two-Way Coupling**

3. Click **Close**.

   The **Phase 1** phase node appears as shown below:



4. Save the simulation.

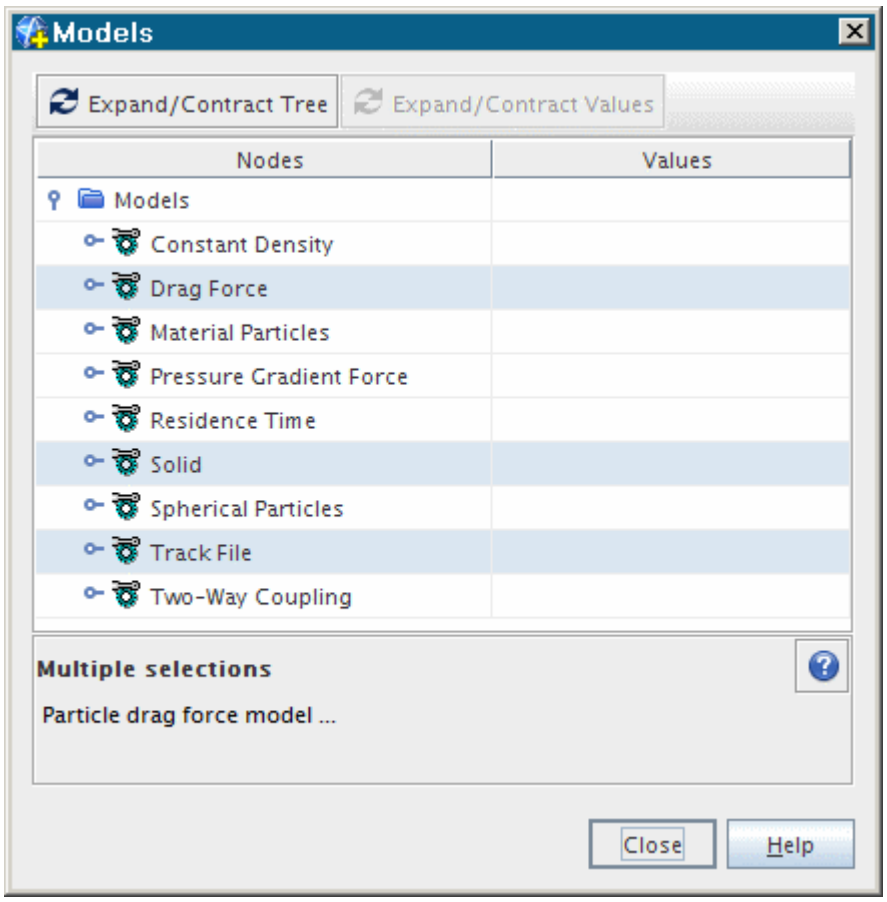
Setting the Lagrangian Phase Model Properties

You specify the appropriate properties of the Lagrangian phase models.

To set the properties for the Lagrangian Phase models:

1. Edit the **Lagrangian Phases** > **Phase 1** > **Models** node and set the following properties:

Node	Property	Setting
Solid > **Al** > **Material Properties** > **Density** > **Constant**	**Value**	`1200 kg/m^3`
Drag Force > **Drag Coefficient**	**Method**	**Schiller-Naumann**
Track File	**Vectors**	**Parcel.Centroid** **Particle.Velocity**



2. Save the simulation.

Setting the Lagrangian Phase Boundary Conditions

You set the boundary interaction mode for Lagrangian Phase Boundary Conditions.

To set the Lagrangian phase boundary conditions:

1. Edit the **Lagrangian Multiphase** > **Lagrangian Phases** > **Phase 1** > **Boundary Conditions** > **Wall** node and set the following properties:

Node	Property	Setting
Physics Conditions > **Mode**	**Active Mode**	**Rebound**
Physics Values > **Normal Restitution Coefficient**	**Value**	`1.0`
└ **Tangential Restitution Coefficient**	**Value**	`1.0`

   Setting the **Value** to `1` represents a perfect elastic rebound.

2. Save the simulation.

Setting the Continuous Phase Boundary Conditions

Define the continuous phase boundary conditions. The default boundary conditions for the outlet and wall boundaries are suitable for this analysis. You specify the boundary conditions at the inlet, a fixed velocity of 10 m/s and appropriate turbulence parameters.

To set the continuous phase boundary conditions:

1. Edit the **Regions** > **Default_Fluid** > **Boundaries** > **inlet** node and set the following properties:

Node	Property	Setting
Physics Conditions > **Turbulence Specification**	**Method**	**Intensity + Length Scale**
Physics Values > **Turbulent Length Scale**	**Value**	`0.001 m`
└ **Turbulence Intensity**	**Value**	`0.005`
└ **Velocity Magnitude**	**Value**	`10 m/s`

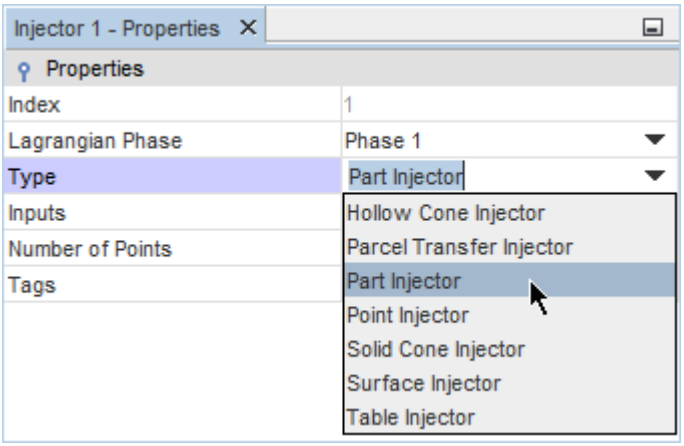
2. Save the simulation.

Setting Up the Injector

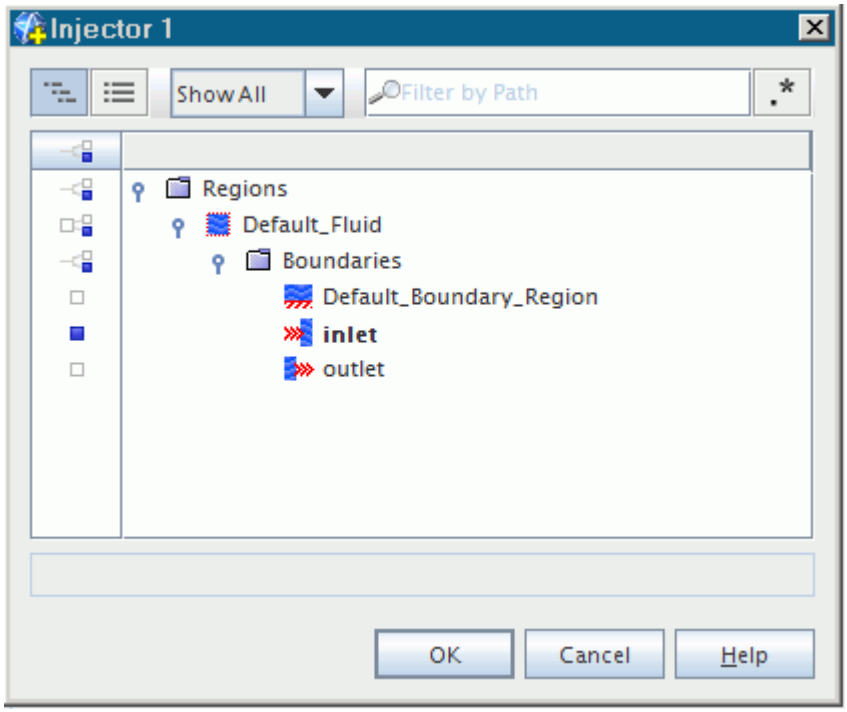
You set up an injector to introduce the particles of the dispersed Lagrangian phase into the solution domain. The injector defines the initial state of the particles.

To set up the injector:

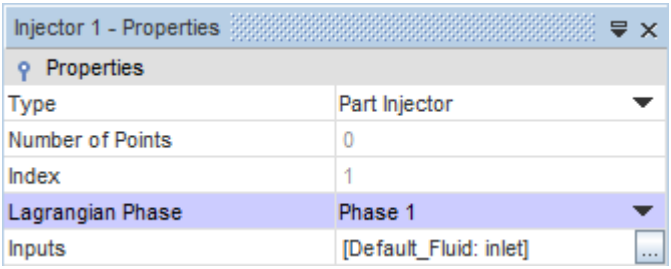
1. Right-click the **Injectors** node and select **New**.
2. Select the **Injectors** > **Injector 1** node and set *Type* to **Part Injector**.



3. Set *Inputs* to **Regions** > **Default Fluid** > **Boundaries** > **inlet**.

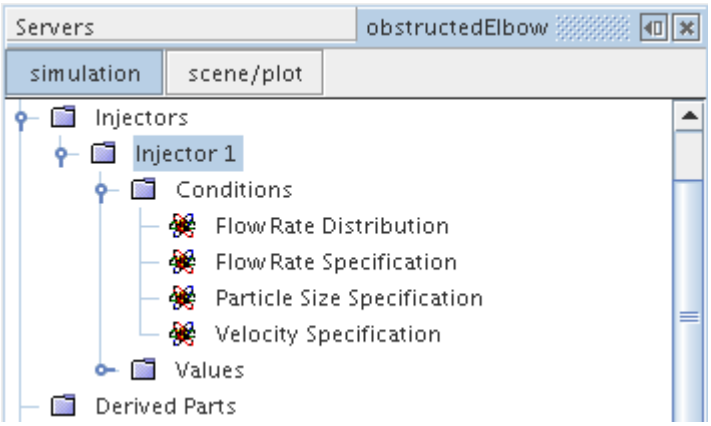


4. Click **OK**.
5. Set *Lagrangian Phase* to **Phase 1**.



The *Number of Points* property is 0 until the injection points are calculated during solution initialization. Specify the conditions and values of the injector to define the particles.

6. Open the **Injectors** > **Injector 1** node.



7. Edit the following nodes and set the appropriate properties:

Node	Property	Setting
Conditions > **Flow Rate Specification**	*Method*	**Volume Flow Rate**
└ **Flow Rate Distribution**	*Method*	**Per Injector** This signifies that the volume flow rate applies to the whole injector, rather than per injection point.
└ **Velocity Specification**	*Method*	**Magnitude + Direction**
Values > **Volume Flow Rate**	*Value*	`6.4516E-7` m^3/s
└ **Particle Diameter**	*Value*	`4.0E-5` m
└ **Velocity Magnitude**	*Value*	`10` m/s

8. Save the simulation.

Setting Solver Parameters and Stopping Criteria

Set the appropriate solver parameters and stopping criteria for the simulation. You create a new stopping criterion based on the continuity equation residual and set it to a minimum tolerance of 1E-4.

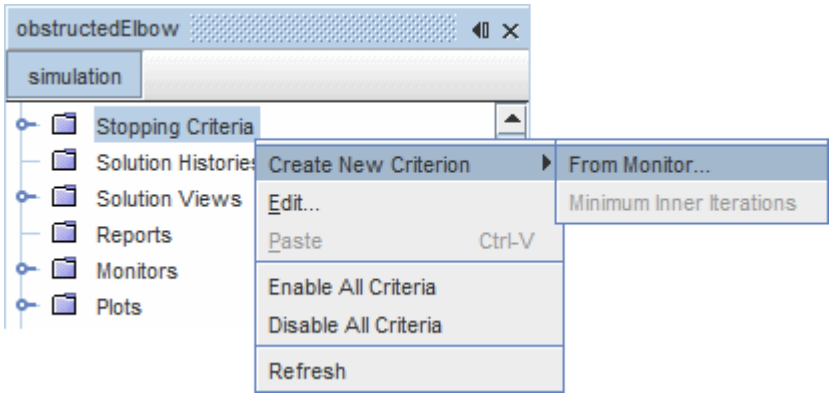
To set the solver parameters and stopping criteria:

1. Edit the **Solvers** node and set the following property:

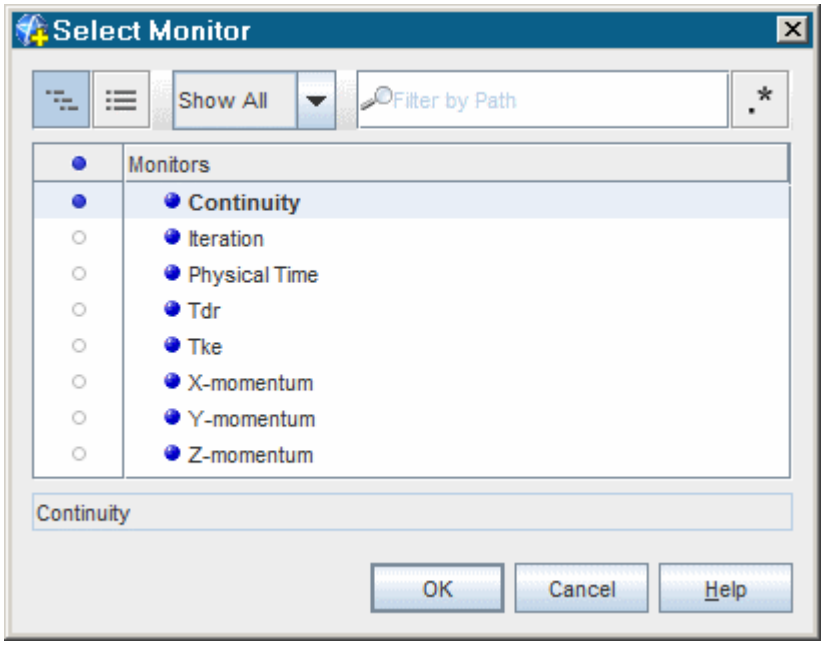
Node	Property	Setting
Lagrangian Multiphase > **Steady**	**Maximum Residence Time**	`0.1`

A new stopping criterion is created based on the continuity equation residual and with a minimum tolerance of 1E-4.

2. Right-click the **Stopping Criteria** node and select **Create New Criterion** > **From Monitor**.



3. Select **Continuity** from the Select Monitor dialog and click **OK**.



4. Edit the **Stopping Criteria** node and set the following property:

Node	Property	Setting
Continuity Criterion > **Minimum Limit**	**Minimum Value**	`1.0E-4`

5. Save the simulation.

Visualizing the Solution

View the results of the simulation. You can view the velocity vectors on a cross-section through the fluid region as the solution develops.

To visualize the solution:

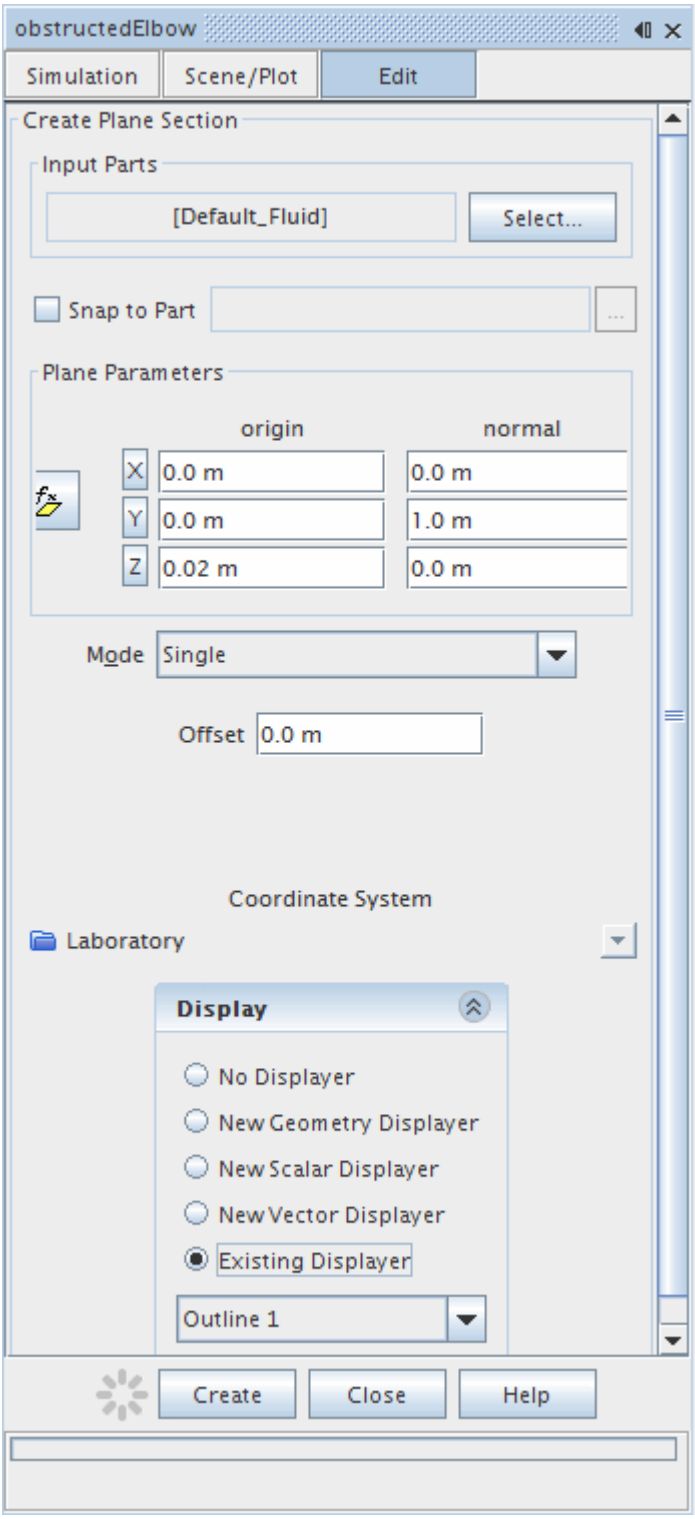
1. Right-click the **Scenes** node, and select **New Scene** > **Vector**.
   The *Vector Scene 1* display appears.

Create a plane section part for display.

2. Right-click the **Derived Parts** node, and select **New Part** > **Section** > **Plane**.

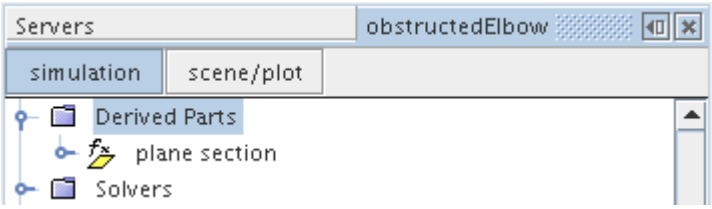
   A new interactive in-place dialog, *Create Plane Section*, appears.
3. In the *Plane Parameters* box, set the normal to `(0,0,1)` and the origin to `(0,0,0.02)`.
4. In the *Display* box, select the radio button **Existing Displayer** to show the plane section in the current scene.

   The completed dialog is shown below.



5. Click **Create** to create the plane section.
6. Click **Close** in the *Create Plane Section* dialog.

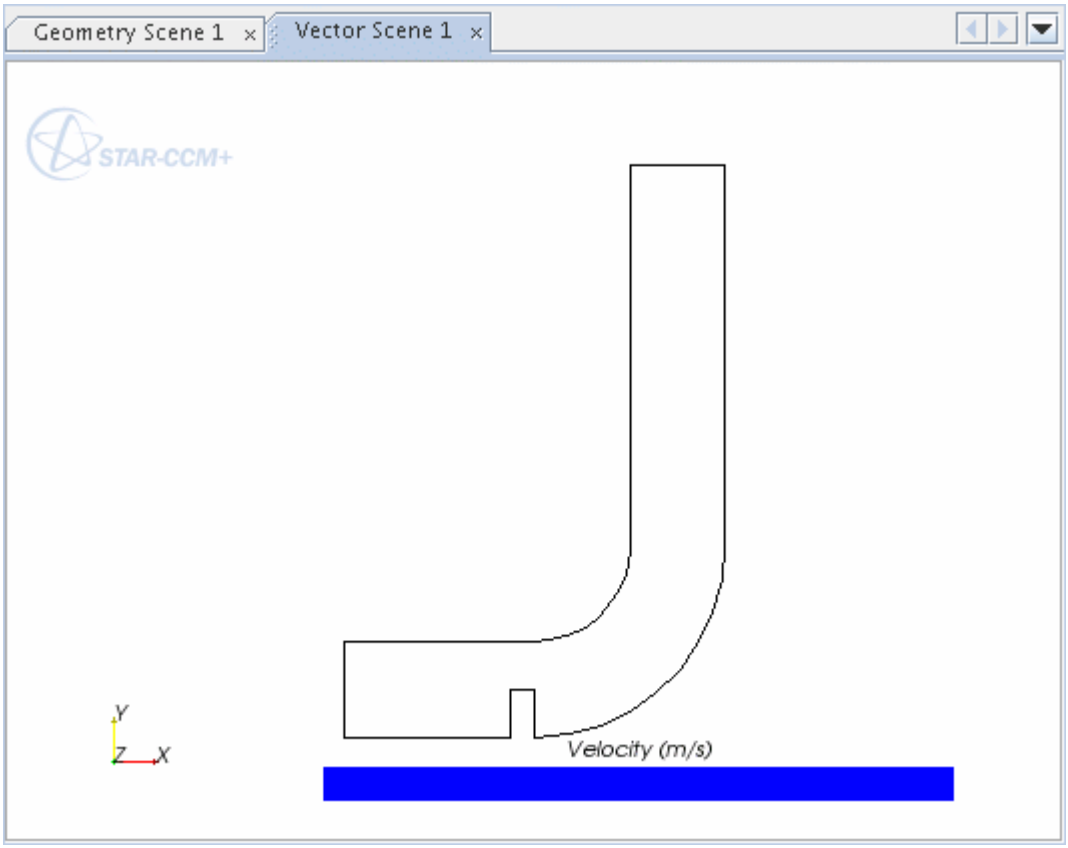
   A node appears within the **Derived Parts** node, **plane section**.



The plane section part is displayed in the vector scene geometry. To display velocity vectors on the plane section, the part must also be added to the Vector 1 displayer.

7. Select the **Scenes** > **Vector Scene 1** > **Displayers** > **Vector 1** > **Parts** node and set *Parts* to **Derived Parts** > **plane section**.
8. Select the **Scenes** > **Vector Scene 1** > **Displayers** > **Outline 1** > **Parts** node and set *Parts* to **Derived Parts** > **plane section**.

The vector scene is now ready. No vectors appears in the scene until the run is started.



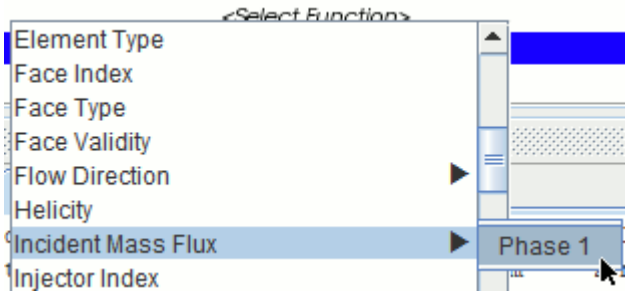
Create a scalar scene.

9. Right-click the **Scenes** node, and select **New Scene** > **Scalar**.

   The *Scalar Scene 1* display appears.

To define the scalar scene:

10. Select the new **Scalar Scene 1** > **Displayers** > **Scalar 1** node and change the *Contour Style* to **Smooth Filled**.
11. Select the **Scalar 1** > **Parts** node and set *Parts* to **Regions** > **Default Fluid** > **Default_Boundary_Region**.
12. Right-click the scalar bar in the display. In the pop-up menu that appears, select **Incident Mass Flux** > **Phase1**.



13. Save the simulation.

Initializing the Solution and Running the Simulation

Preparation of the simulation is now complete, and the simulation can be run.

To initialize the solution and run the simulation:

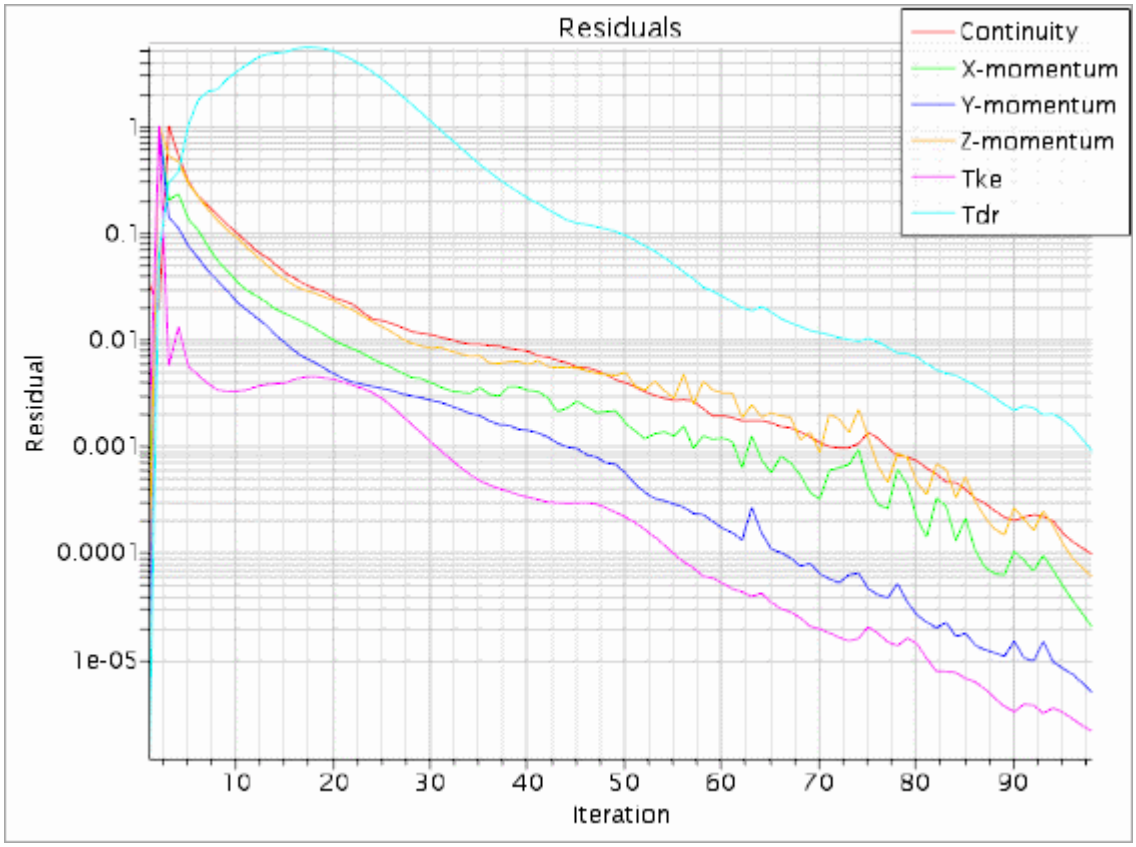
1. Click (**Initialize Solution**) in the *Solution* toolbar or select **Solution** > **Initialize Solution** from the menu.
2. Click (**Run**) in the *Solution* toolbar.

   The solution progress is displayed in the *Output* window. You can ignore the warning message about smoothing in the *Output* window. The *Residuals* display is created automatically in the *Graphics* window and shows the progress of the solvers.

   While the simulation is running, you can click the tabs at the top of the *Graphics* window to view scenes and plots.
3. Select the *Vector Scene 1* tab to view the velocity vectors cross-section as the solution develops.

   During the run, you can stop the process by clicking (**Stop**) on the toolbar. If you do stop the simulation, you can click (**Run**) to resume it.

   If left alone, the simulation continues until 1000 iterations are complete or the continuity residual criterion, 1.0E-4, is satisfied.



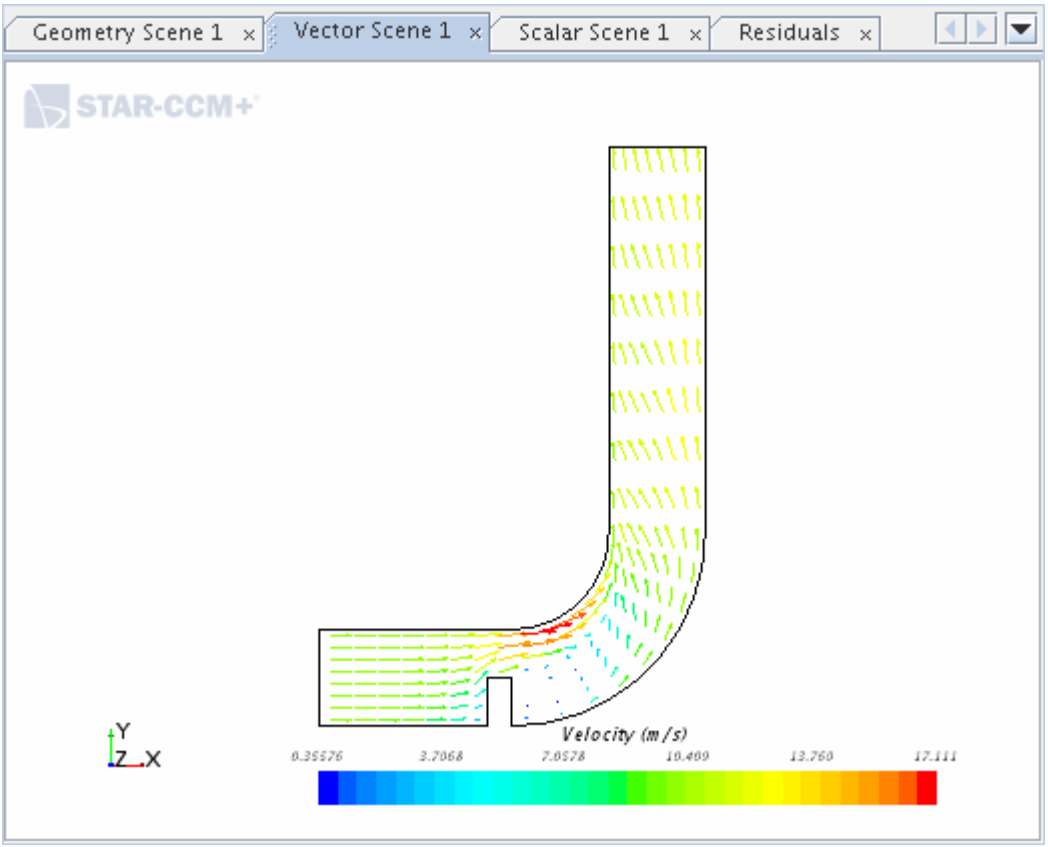
4. When the simulation has finished running, save it.

Visualizing the Results

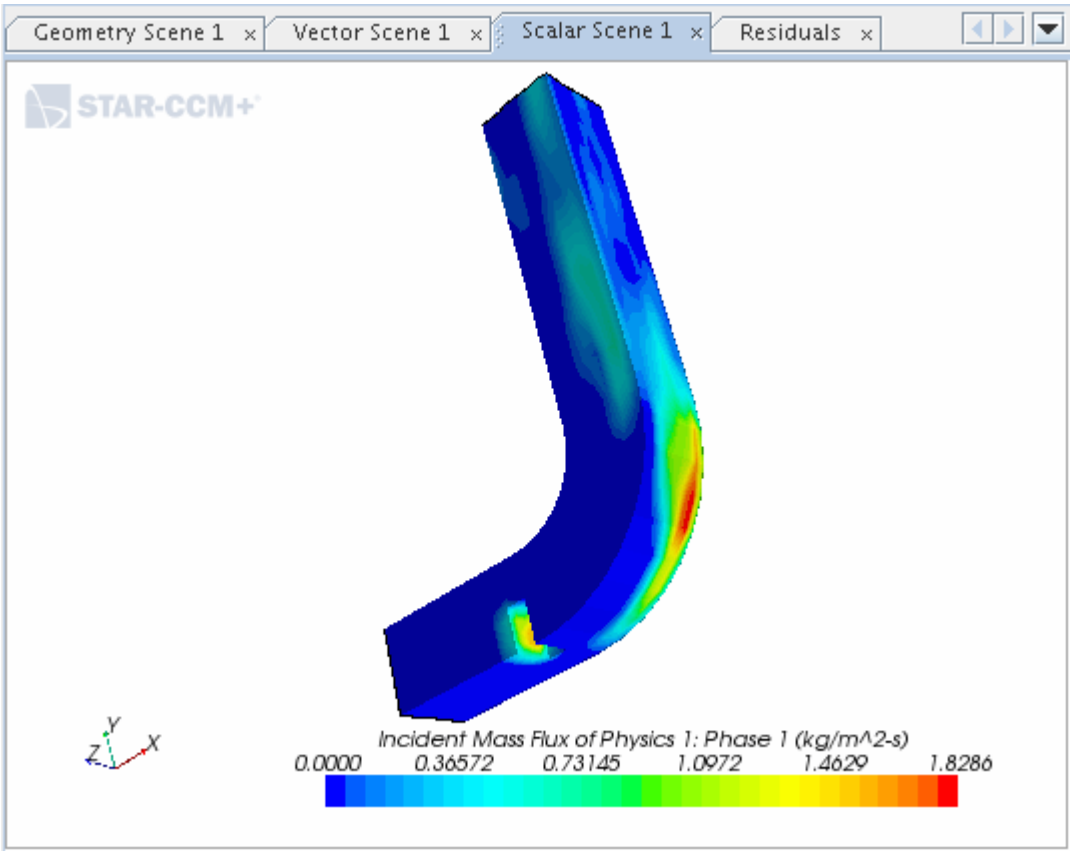
You can view the results of the simulation. The *Vector Scene 1* displays the velocity vector cross-section for the converged solution. The *Scalar Scene 1* displays the incident mass flux profile for the converged solution.

To visualize the results:

1. Select the *Vector Scene 1* display to view the velocity vector cross-section for the converged solution.



2. Select the *Scalar Scene 1* display to view the incident mass flux profile for the converged solution.
3. Use the mouse controls to adjust the view as shown below.



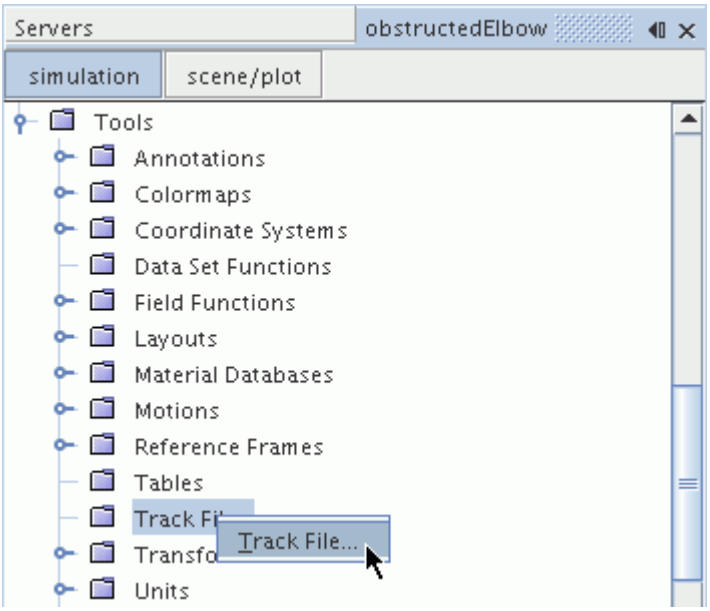
The particle impacts are concentrated on the outer radius of the bend. In a realistic modeling exercise, a considerably larger number of particles tracks would be needed to increase confidence in the results.

Displaying Particle Tracks

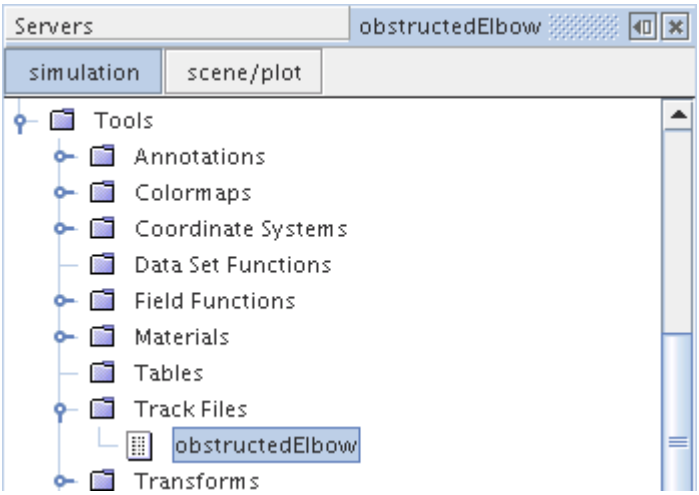
You can display the particle tracks recorded by the Track File model.

To display the particle tracks:

1. Right-click the **Tools** > **Track Files** node and select **Track File**.



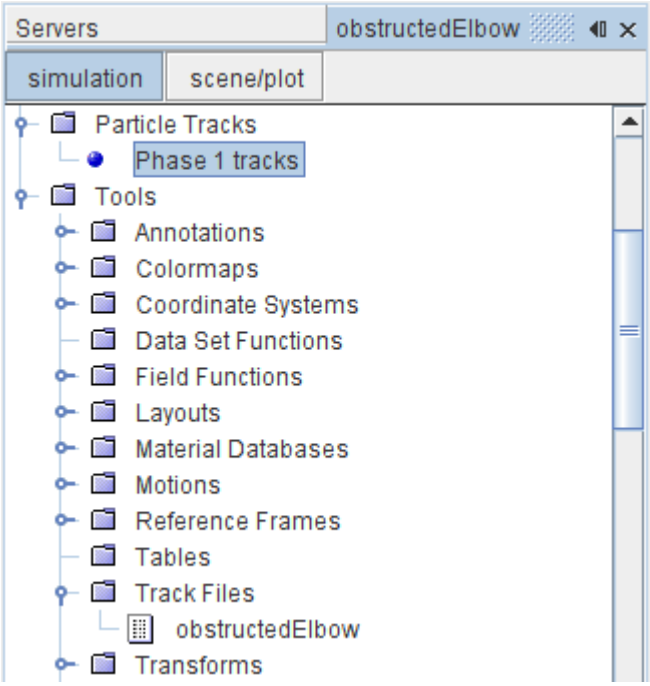
2. In the *Open* dialog, select the `obstructedElbow.trk` file and click **Open**.



Note: The simulation must be saved before the track file appears in the dialog.

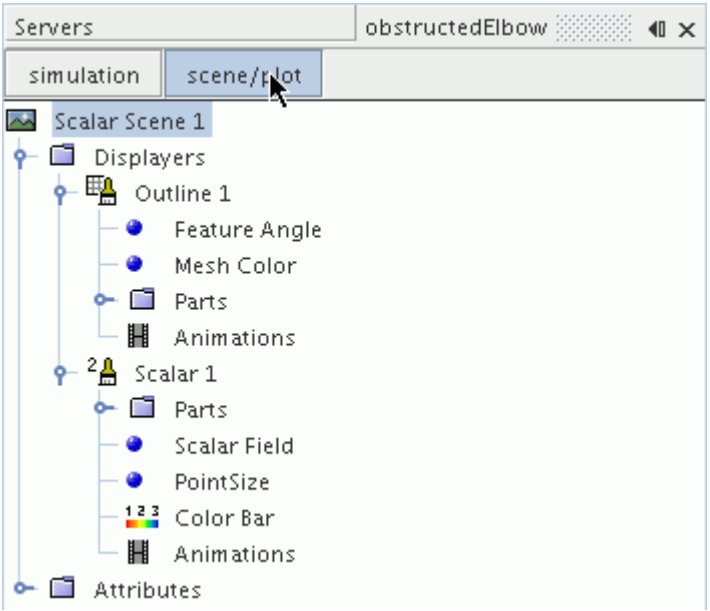
A new node **obstructedElbow** appears under the **Track Files** node. Another new node, **Particle Tracks**, also appears in the simulation tree.

3. Open the **Particle Tracks** node to see the **Phase 1 tracks** node that was populated from the `obstructedElbow.trk` file.



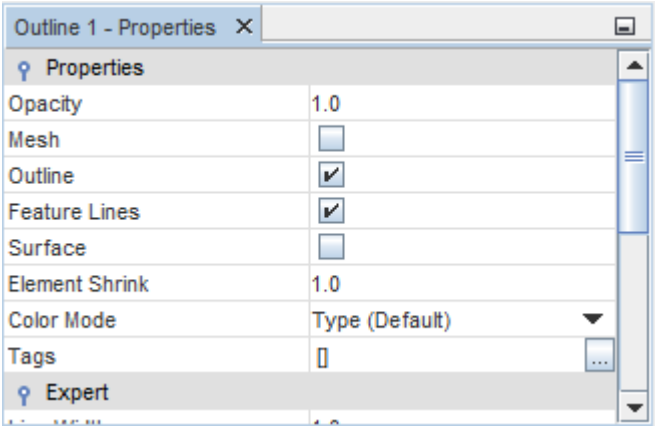
Add particle tracks to the scalar scene.

4. Select the *Scalar Scene 1* display in the *Graphics* window.
5. Jump directly to the object tree for the **Scalar Scene 1** node by clicking the **scene/plot** button above the simulation tree.



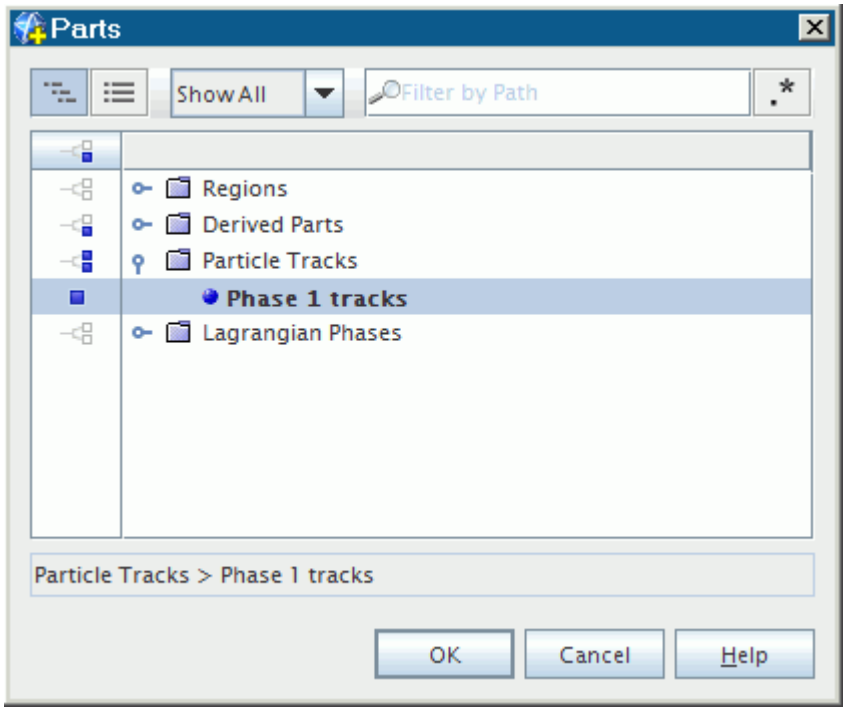
Add feature lines first, and then add particle tracks.

6. Select **Scalar Scene 1** > **Displayers** > **Outline 1** and tick the *Feature Lines* property.



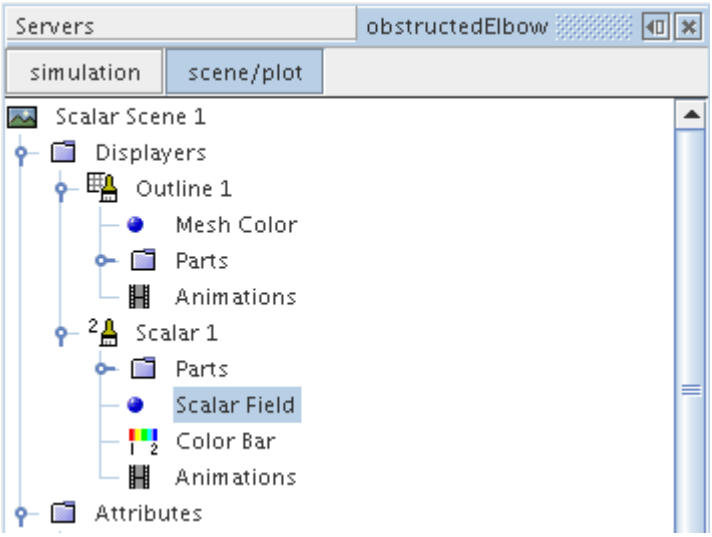
7. Select the **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Parts** node and click the right half of the *Parts* property.

8. In the dialog that appears, use (**Column Selection**) to **Deselect All**, expand the **Particle Tracks** node and select **Phase 1 tracks** as shown below.

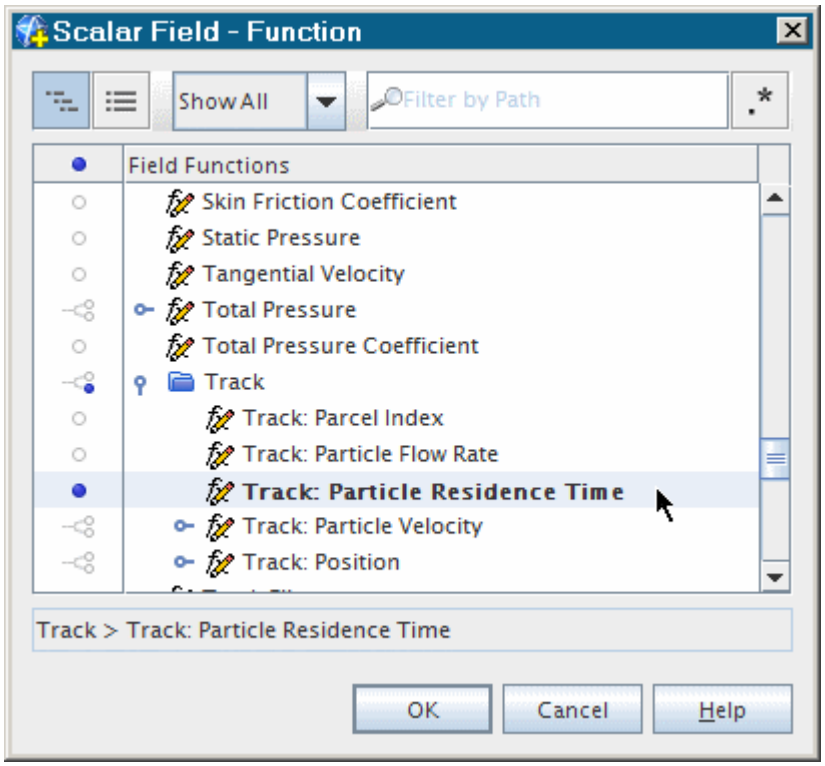


9. Click **OK**.

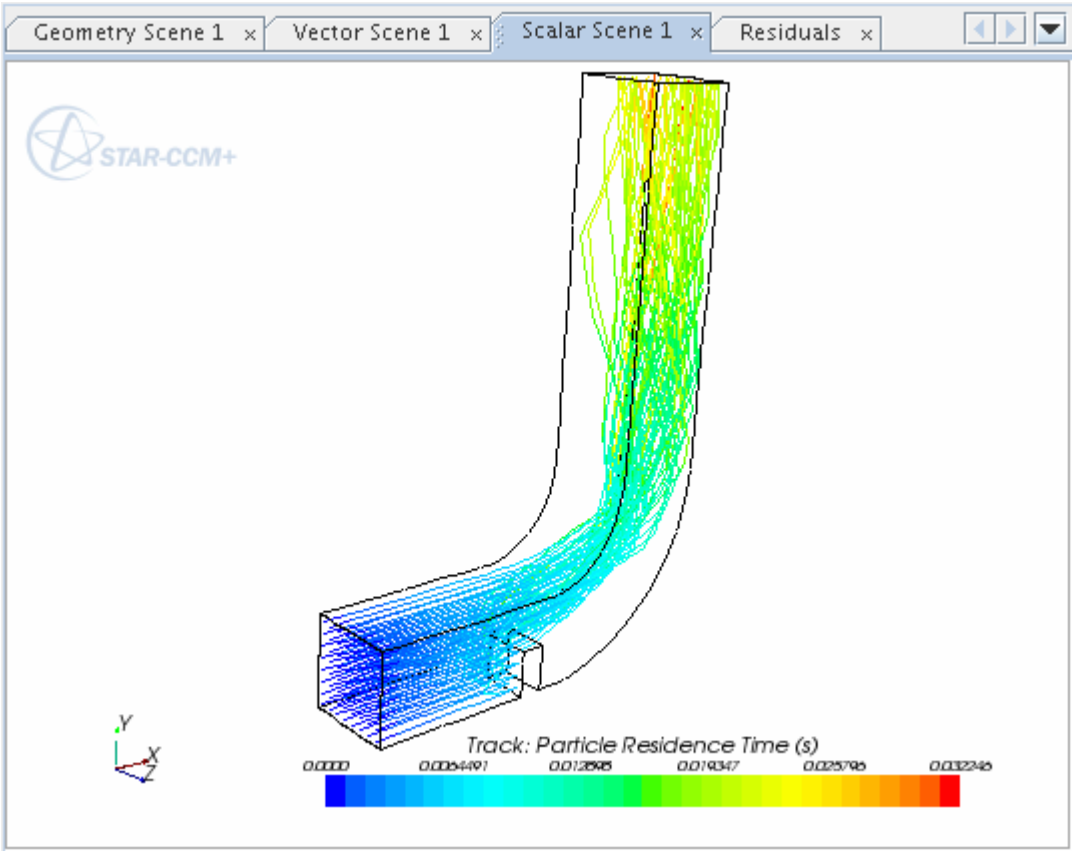
10. Continuing within the **Scalar 1** node, select the **Scalar Field** node.



11. Set the *Function* to **Track** > **Track: Particle Residence Time** using the drop-down menu.



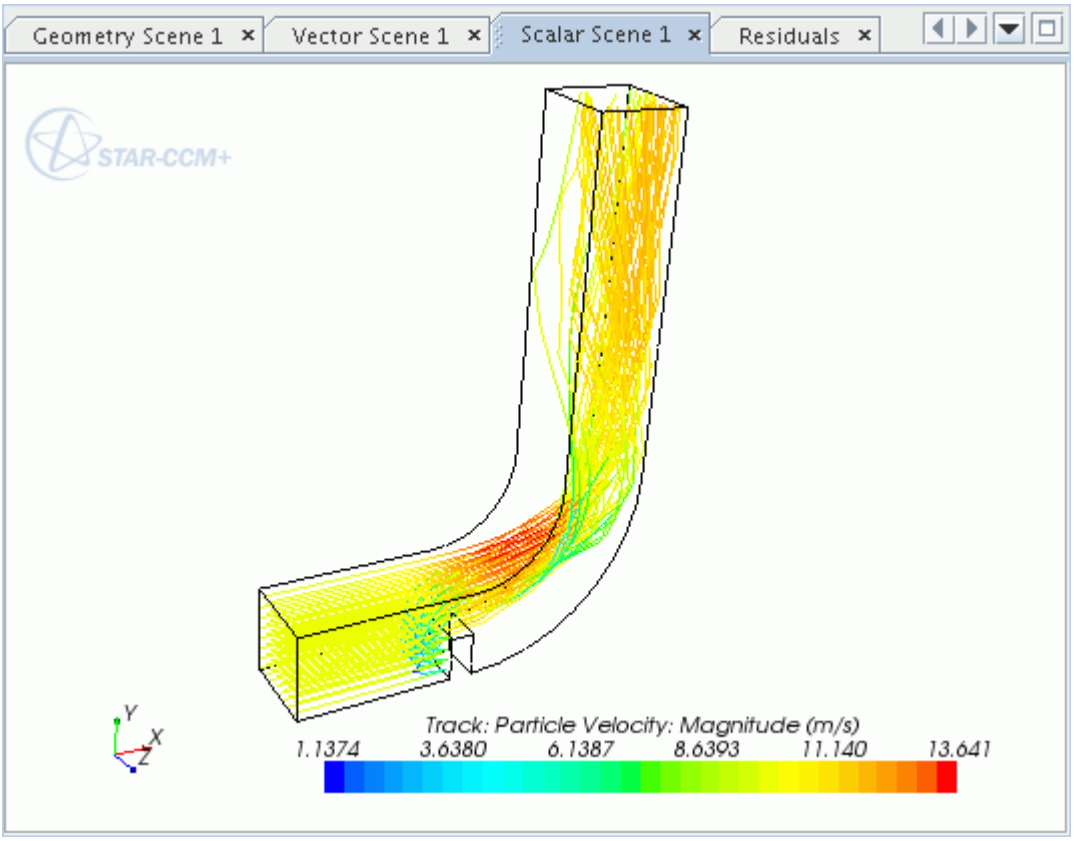
12. In the *Scalar Scene 1* display, use the mouse controls to adjust the view as shown below.



The maximum time taken for all particle tracks to leave the domain is approximately 0.032 seconds. So, the solver parameter value that is specified for *Maximum Residence Time*, 0.1s, is more than sufficient.

13. With the **Scalar 1** > **Scalar Field** node still selected in the partial tree for **Scalar Scene 1**, set *Function* to **Track:** > **Track: Particle Velocity** > **Magnitude**.

    The particle tracks are now colored according to the velocity of the particles.



14. Click **Simulation** to return to the STAR-CCM+ object tree.
15. Save the simulation.

Using Tracks in Field Functions and Derived Parts

In simulations where many particle tracks are present, it is likely that only a small number are required for plotting in a graphics scene. You can create a field function to filter the available tracks and apply the filter in a derived part.

A field function is created that returns the Track Parcel Index number for every eighth track. The function name for the Track Parcel Index is **TrackParcelId** (which can be found from the *Function Name* property of the **Track: Parcel Index** node in **Tools** > **Field Functions**).

To use tracks in field functions and derived parts:

1. Right-click the **Tools** > **Field Functions** node and select **New** > **Scalar** from the pop-up menu.
2. Rename the **User Field Function 1** node to `TrackFilter`.
3. Select the **Track Filter** node and set the following properties:

Property	Setting
Function Name	`Track Filter`
Dimensions	*Dimensionless*

Property	Setting
Definition	`(fmod($TrackParcelId, 8) <= 0.0001) ? $TrackParcelId : -1`

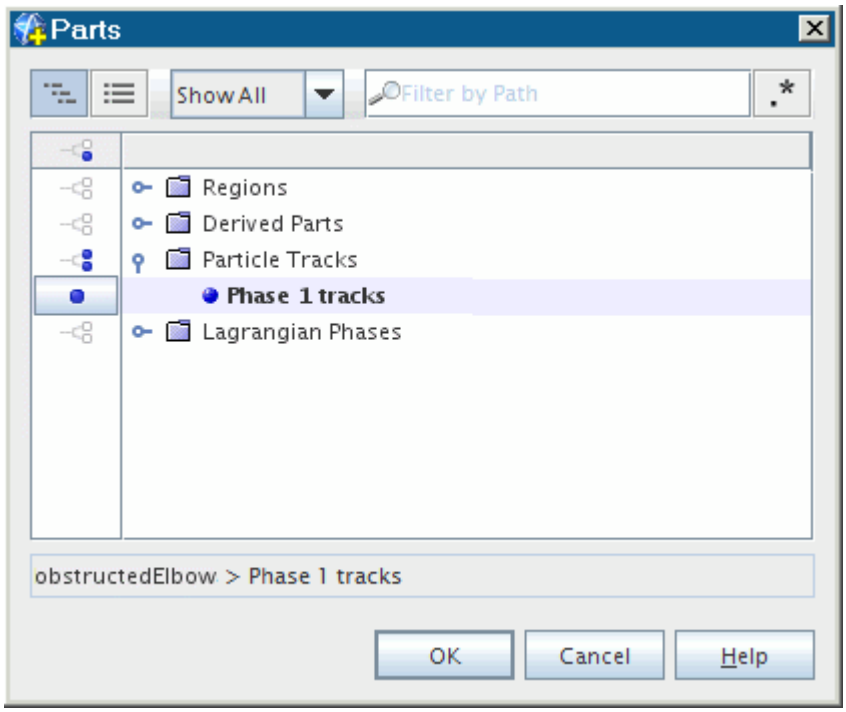
The function entered here uses the fmod function and the conditional expression that is defined by the ternary operator “?:”. The function essentially says that if an integer results from the division of **TrackParcelId** by 8, then the **TrackParcelId** is returned; otherwise the value -1 is returned.

Create a derived part based on a threshold to use this behavior.

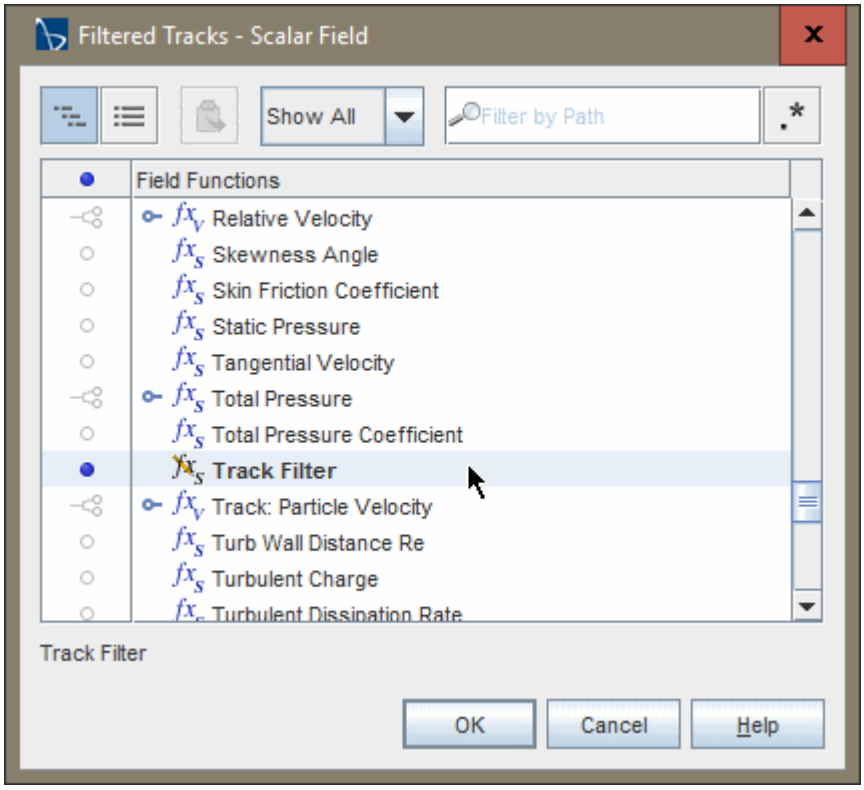
4. Right-click **Derived Parts** and select **New Part** > **Threshold** from the pop-up menu.

   The *Create Threshold* dialog appears above the object tree.

5. In the *Input Parts* box, click the drop-down menu. In the dialog that appears, use (**Column Selection**) to **Deselect All**, expand the **Particle Tracks** node and select **Phase 1 tracks** item.

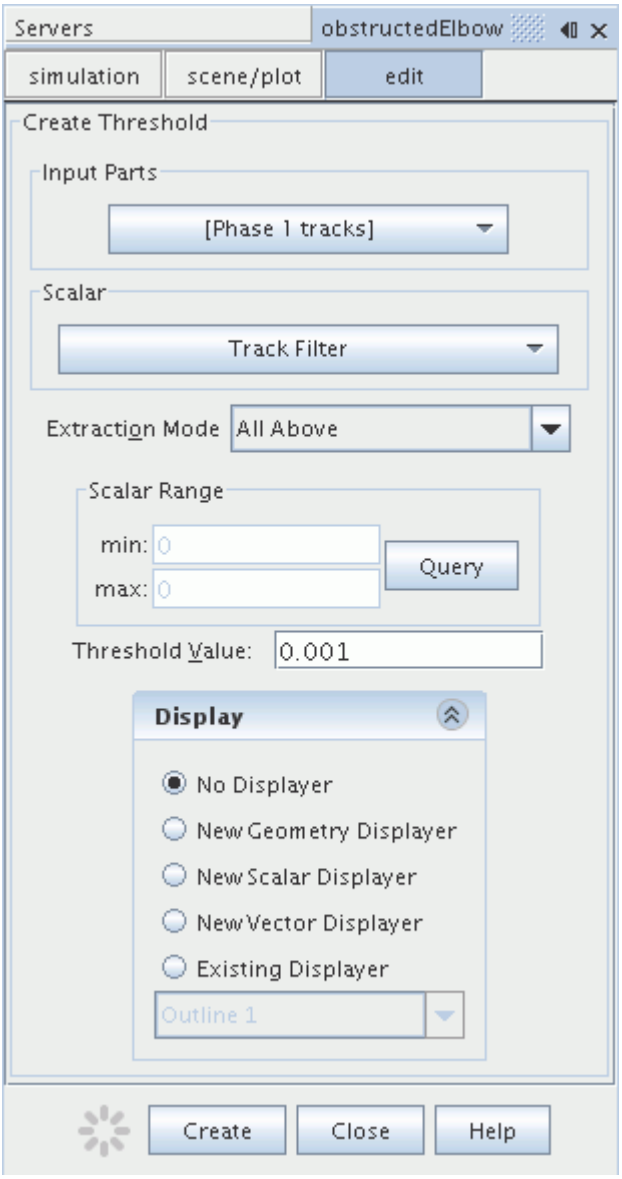


6. Click **OK**.
7. Click the drop-down menu in the *Scalar* box and choose item **Track Filter** as shown.



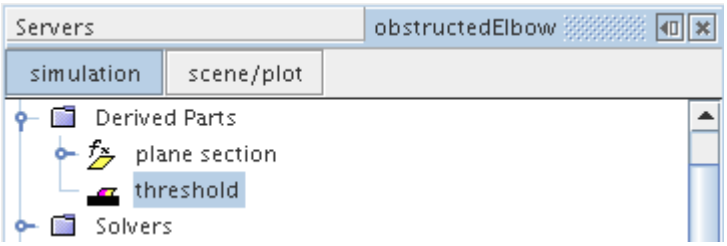
8. Set the **Extraction Mode** to **All Above** and the **Threshold Value** to `0.001`.
9. Select **No Displayer** in the *Display* box.

   The completed panel is shown below.



10. Click **Create** and then **Close**.

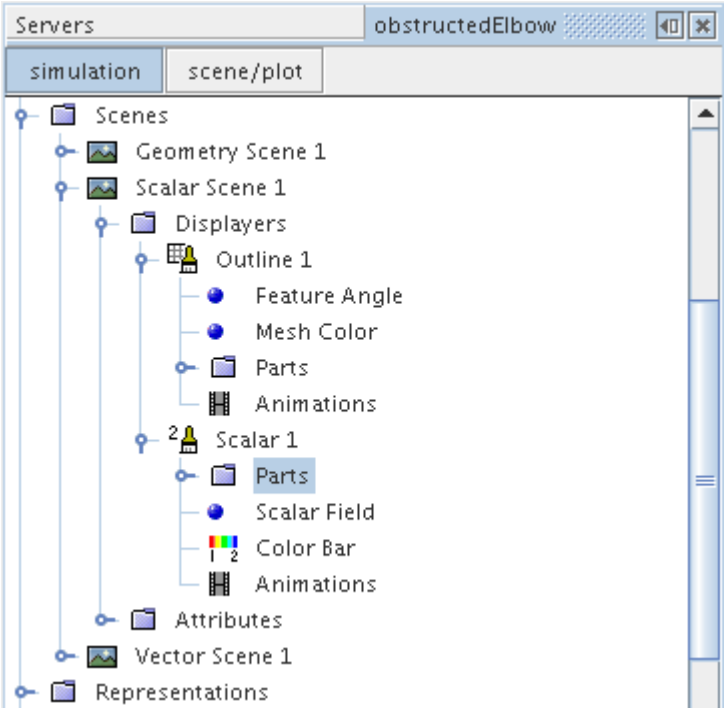
    A new derived part named **threshold** appears in the object tree.



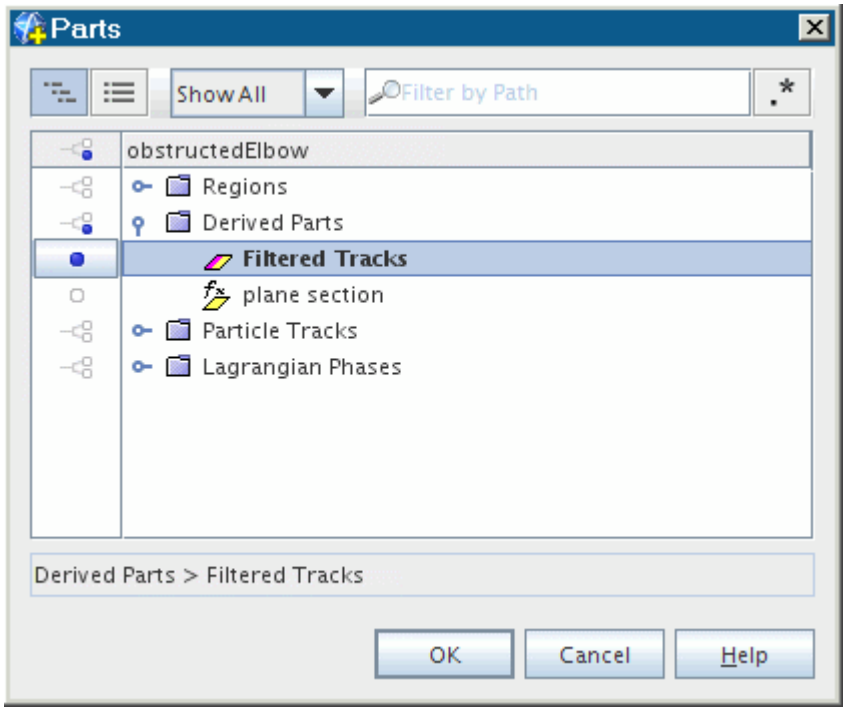
11. Rename the **threshold** node to `Filtered Tracks`.

Modify *Scalar Scene 1* to display the filtered tracks:

12. Select the **Scenes** > **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Parts** node.

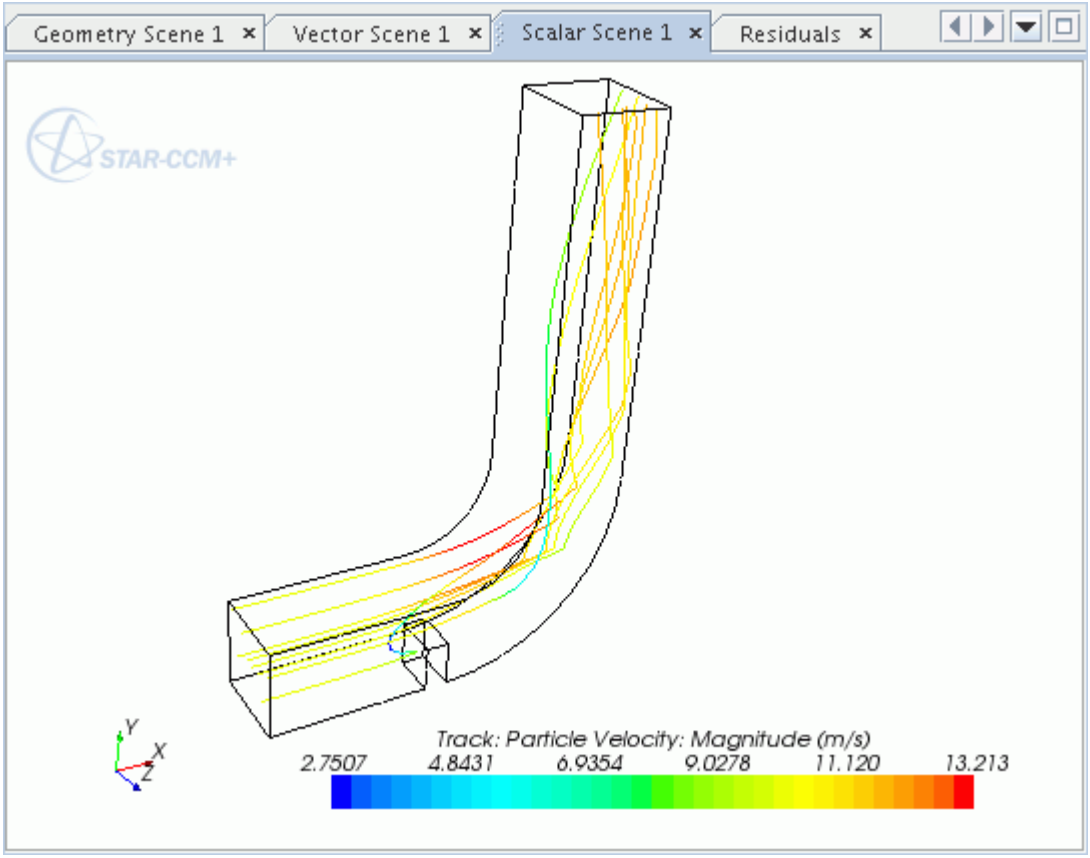


13. Click the right half of the *Parts* property.
14. In the dialog that appears, click the drop-down menu.
15. In the dialog that appears, use (**Column Selection**) to **Deselect All**, expand the **Derived Parts** node and select **Filtered tracks** as shown below.



16. Click **OK**.

Scalar Scene 1 now shows the filtered set of particle tracks.



Summary

This tutorial has demonstrated how to set up a simple Langrangian analysis in STAR-CCM+.

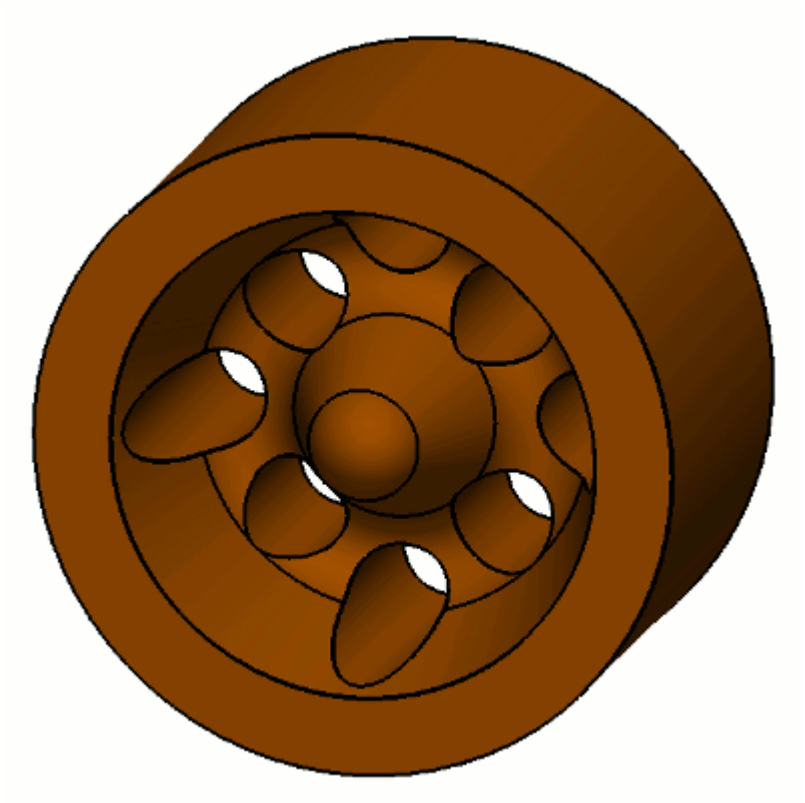
This tutorial has introduced the following STAR-CCM+ features:

- Selecting models for a Lagrangian phase.
- Setting up an injector for Lagrangian Multiphase modeling.
- Defining material properties that are required for the selected models.
- Defining boundary conditions.
- Setting solver parameters.
- Initializing and running the solver until the residual magnitudes are sufficiently small.
- Analyzing results using the visualization facilities.
- Displaying particle tracks.
- Creating a field function and derived part based on particle tracks.

Lagrangian: Solid Particle Erosion

This tutorial demonstrates how to set up an erosion modeling analysis in STAR-CCM+. It simulates the erosion that is caused by solid particles in a liquid flow.

The geometry that is used in this tutorial was originally designed as a pressure reduction device for choke valves under erosive service, and is the subject of a US Patent ([793]). The device is shown below.



This device was tested under erosive service using a sand-liquid erosion test facility (as reported by Wallace, [794]), and the conditions that are used in this tutorial are based on the actual tests. In the erosion tests, the flow restrictor was mounted between lengths of 53.1 mm diameter piping. Static pressure measurements were made at two positions: 106 mm upstream of the restrictor and 318 mm downstream. Overall mass loss was obtained by weighing the flow restrictor periodically during the tests.

In this tutorial, water at room temperature is used for the liquid phase. Inlet conditions are as follows (corresponding to a flow rate of 28.57 liters/s in a 53.1 mm diameter pipe):

Liquid Phase

Property	Value	Units
Fluid Density	997.561	kg/m^3
Fluid Viscosity	8.8871e-4	Pa-s

Property	Value	Units
Inlet Velocity	12.901	m/s
Turbulence Intensity	0.029	
Turbulent Length Scale	0.0037	m

The particle loading by mass is 0.39%. The particle properties are as follows:

Particle Phase

Property	Value	Units
Density	2650.0	kg/m^3
Diameter	2.75e-4	m
Initial axial velocity	12.901	m/s

Contents:

Prerequisites
Importing the Mesh and Naming the Simulation
Visualizing the Imported Geometry
Selecting the Physics Models
Selecting the Lagrangian Phase Models
Setting the Material Properties
Setting the Lagrangian Phase Model Properties
Defining Field Functions for the Restitution Coefficients
Setting the Lagrangian Phase Boundary Conditions
Setting the Erosion Model
Setting Continuous Phase Boundary Conditions
Setting Up the Injector
Setting Solver Parameters and Stopping Criteria
Visualizing the Solution
Reporting and Plotting the Pressure Drop
Initializing and Running the Simulation
Visualizing the Results
Estimating the Overall Erosion Rate
Comparison with Experimental Data
Implementing Alternative Functions for Erosion Ratio
Summary
Bibliography

Prerequisites

The instructions in the Lagrangian: Solid Particle Erosion tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Importing the Mesh and Naming the Simulation

Launch STAR-CCM+, import the supplied geometry, and save the simulation.

A three-dimensional polyhedral cell mesh has been prepared for this analysis. The grid has been saved in the .ccm file format which contains all the required cell, vertex, and boundary information for the problem geometry.

To import the mesh and name the simulation:

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh...**
4. In the *Open* dialog, choose `doc/startutorialsdata/multiphaseFlow/data/flow_restrictor.ccm` then click **Open**.

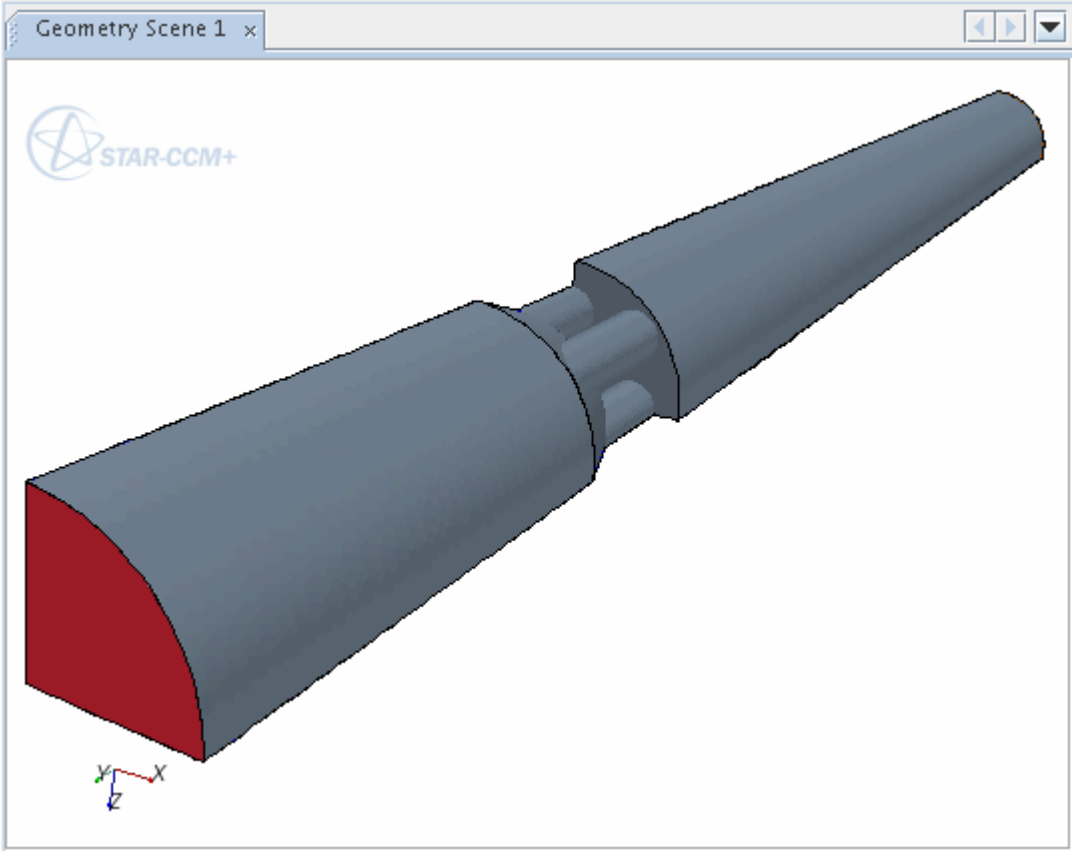
   STAR-CCM+ provides feedback on the import process, which takes a few seconds, in the *Output* window. One mesh region named **Fluid** is created in the **Regions** node representing the grid domain. A geometry scene is created in the *Graphics* window.
5. Save the simulation as `erosion.sim`.

Visualizing the Imported Geometry

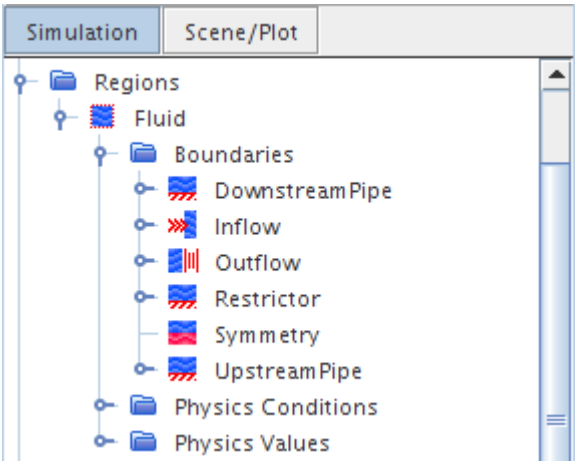
Examine the *Geometry Scene 1* display in the *Graphics* window.

To visualize the imported geometry:

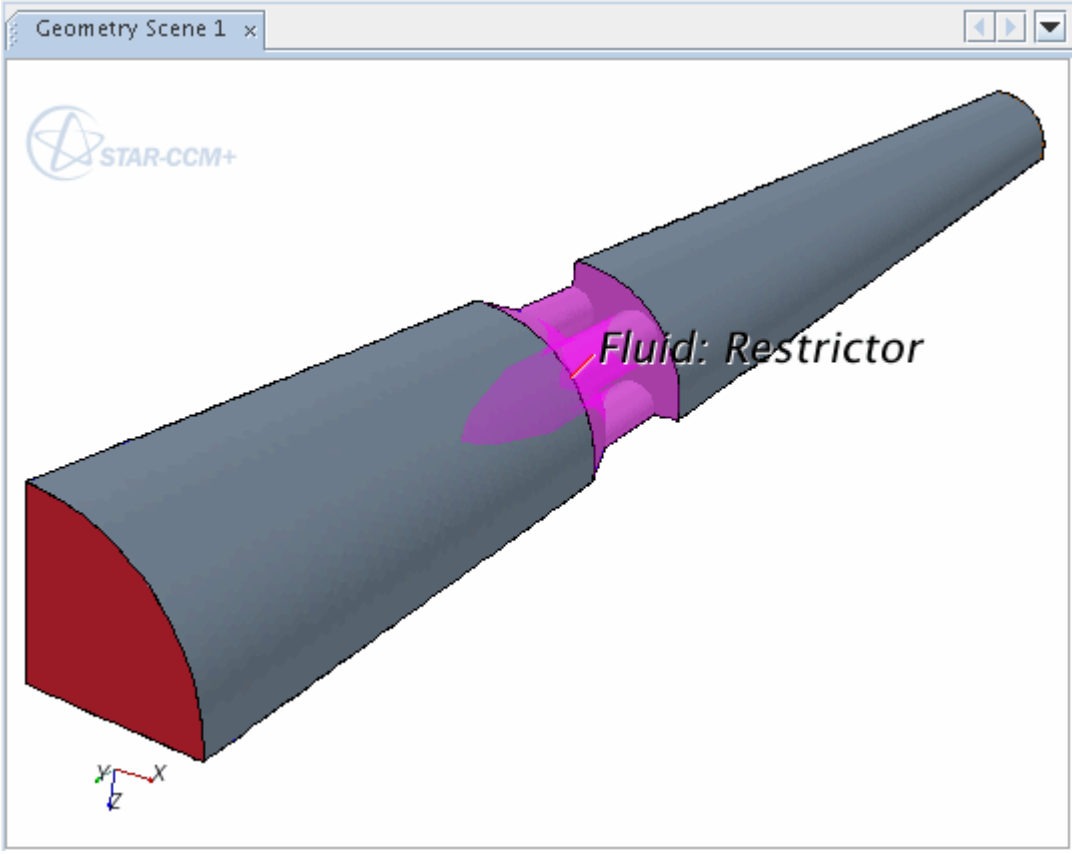
1. Use the mouse controls to adjust the view as shown below.



2. To see the boundary regions that were imported with the mesh, open the **Regions** > **Fluid** > **Boundaries** node.



3. To observe the surfaces of the flow restrictor in the *Geometry Scene 1* display, select the **Restrictor** boundary node.



Selecting the Physics Models

Physics models define the primary variables of the simulation, including pressure, temperature, velocity, and what mathematical formulation is used to generate the solution.

Several models are required in this tutorial as there is not only a turbulent continuous phase to consider, but also the particle motion within the continuous phase. Two different strategies are employed within STAR-CCM+ to simulate these phases. The continuous fluid phase is modeled using an Eulerian formulation whereby fluid properties are obtained at fixed points throughout the fluid domain. The particulate phase is modeled using a Lagrangian method where the trajectories of representative particles are tracked throughout the continuous phase.

STAR-CCM+ permits two-way coupling between the continuous fluid phase and dispersed particulate phases. For this tutorial, however, the coupling is not considered to reduce the amount of simulation time required.

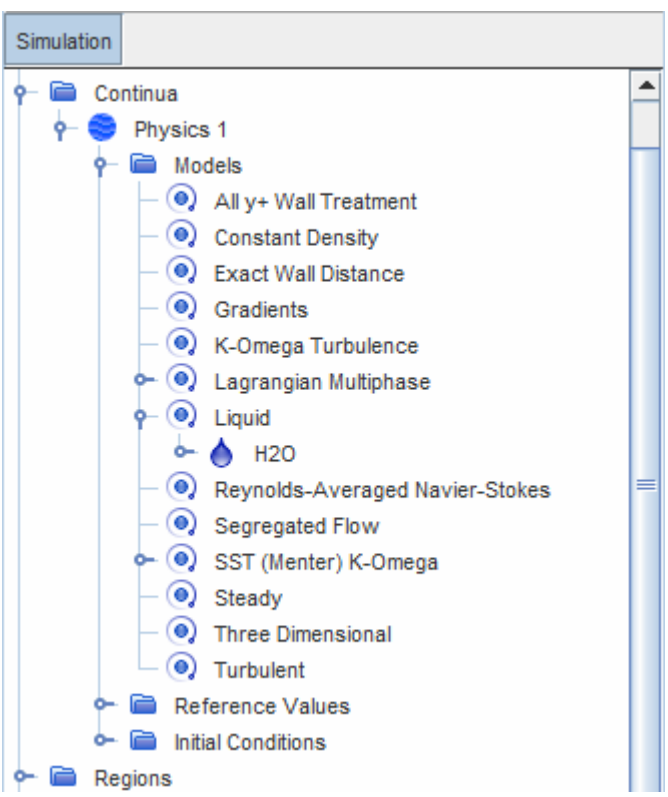
Only one region for the continuous phase is required for this analysis. By default, the continuum **Physics 1** is automatically created when the mesh is imported. The continuum is edited so that appropriate physical models are selected for the water fluid region.

To select the physics models:

1. For the physics continuum, **Continua** > **Physics 1**, select the following models in order:

Group Box	Model
Enabled Models	**Three Dimensional** (Pre-selected)
Time	**Steady**
Material	**Liquid**
Flow	**Segregated Flow** **Gradients** (selected automatically)
Equation of State	**Constant Density**
Viscous Regime	**Turbulent** **Reynolds-Averaged Navier-Stokes** (selected automatically)
Reynolds-Averaged Turbulence	**K-Omega Turbulence** **SST (Menter) K-Omega** (selected automatically) **All y+ Wall Treatment** (selected automatically) **Exact Wall Distance** (selected automatically)
Optional Models	**Lagrangian Multiphase**

2. Click **Close**.
3. To review the models, open the **Physics 1** > **Models** node.



4. Save the simulation.

Selecting the Lagrangian Phase Models

Create the Lagrangian phase and select the appropriate phase models.

The **Lagrangian Multiphase** model requires the selection of further models representing the characteristics of the Lagrangian phase.

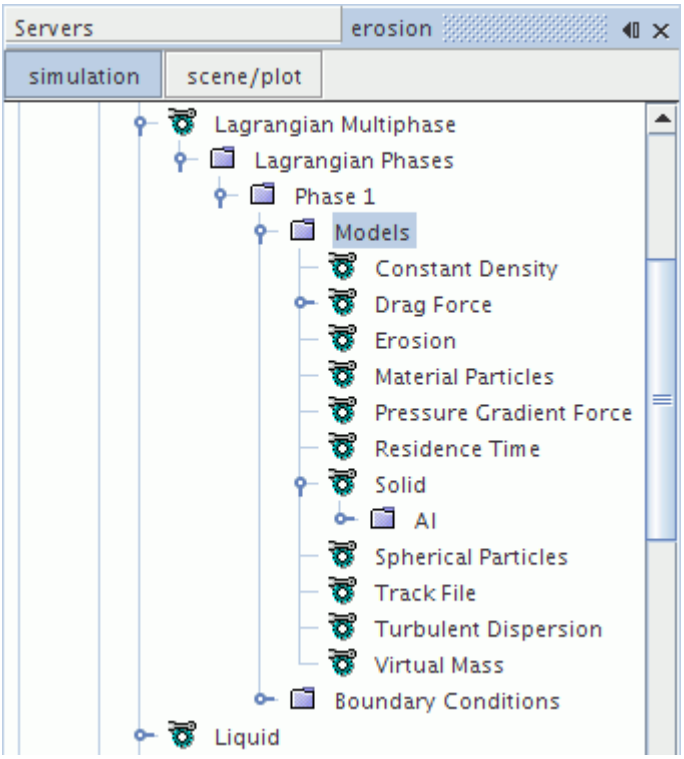
To create the Lagrangian phase and select the phase models:

1. In the **Physics 1** continuum, right-click the **Models** > **Lagrangian Multiphase** > **Lagrangian Phases** node and create a new phase.
2. For the **Phase 1** phase, select the following models in order:

Group Box	Model
Enabled Models	**Residence Time** (Pre-selected)
Particle Type	**Material Particles** **Pressure Gradient Force** (Selected automatically) **Spherical Particles** (Selected automatically)
Material	**Solid**
Equation of State	**Constant Density**
Optional Particle Forces	**Drag Force** **Virtual Mass**
Optional Models	**Turbulent Dispersion** **Erosion**
Track Sampling	**Track File**

3. Click **Close**.

   The **Phase 1** phase node appears as shown below:



4. Save the simulation.

Setting the Material Properties

Define the material properties of the continuous and Lagrangian phases.

This part of the tutorial involves working with material properties for two phases: continuous and Lagrangian. The default properties for water are acceptable for the continuous phase in this simulation. Sand particles are entrained in the liquid flow in this simulation and so the density of the Lagrangian phase must be set to the density of sand.

To set the material properties:

Specify the material properties for the Lagrangian phase.

1. Edit the **Physics 1** > **Models** > **Lagrangian Multiphase** > **Lagrangian Phases** > **Phase 1** > **Models** > **Solid** > **Al** > **Material Properties** node and set the following property:

Node	Property	Setting
Density > **Constant**	**Value**	`2650.0 kg/m^3`

2. Rename the **Al** node to **Sand**.
3. Save the simulation.

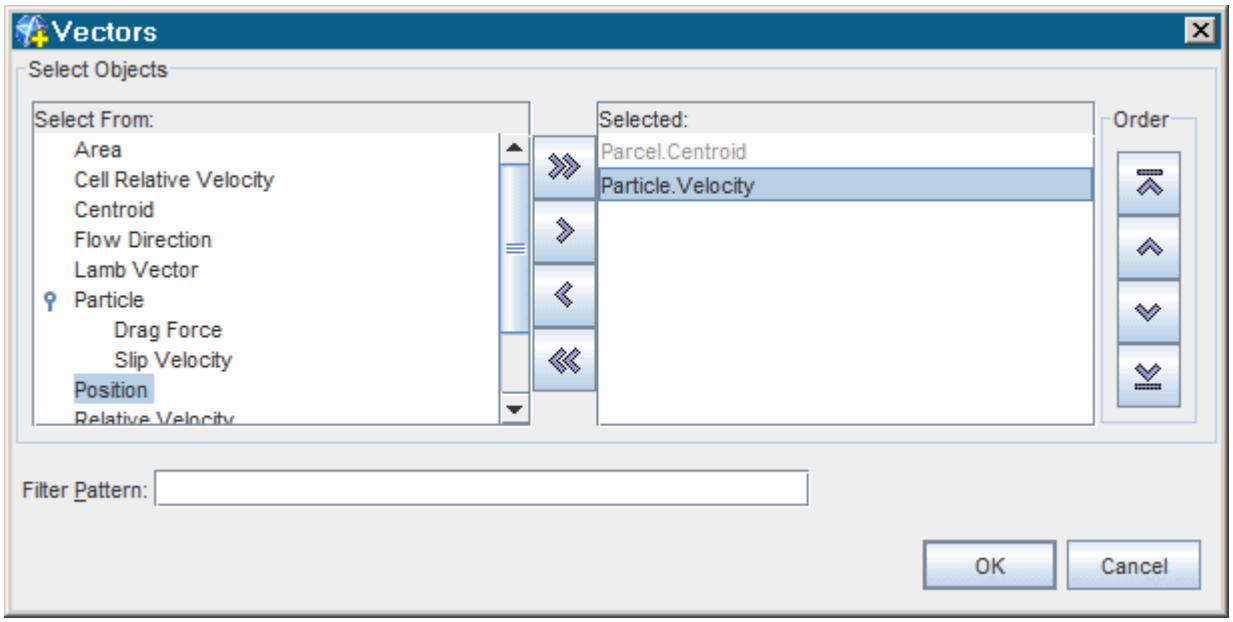
Setting the Lagrangian Phase Model Properties

Define the Lagrangian phase model properties. The Drag Force model uses the default method for the Drag Coefficient, which is the Schiller-Naumann correlation.

To set the properties for the Lagrangian phase models:

1. Edit the **Lagrangian Phases** > **Phase 1** > **Models** node and set the following properties:

Node	Property	Setting
Drag Force > **Drag Coefficient**	**Method**	**Schiller-Naumann**
Track File	**Vectors**	**Parcel.Centroid** **Particle.Velocity**



2. Save the simulation.

Defining Field Functions for the Restitution Coefficients

Create field functions to represent the restitution coefficients.

Restitution coefficients are used to predict the angle with which a particle rebounds off a solid wall. In this tutorial, the relations that were developed by Forder et al. ([795]) are used:

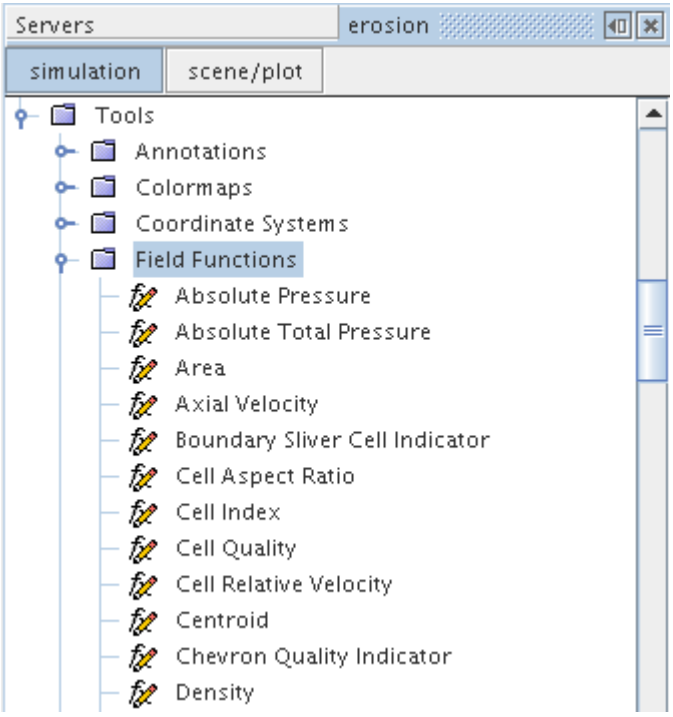
$$\epsilon_T = 1 - 0.78\alpha_p + 0.84\alpha_p^2 - 0.21\alpha_p^3 + 0.028\alpha_p^4 - 0.022\alpha_p^5$$

$$\epsilon_N = 0.988 - 0.78\alpha_p + 0.19\alpha_p^2 - 0.024\alpha_p^3 + 0.027\alpha_p^4$$

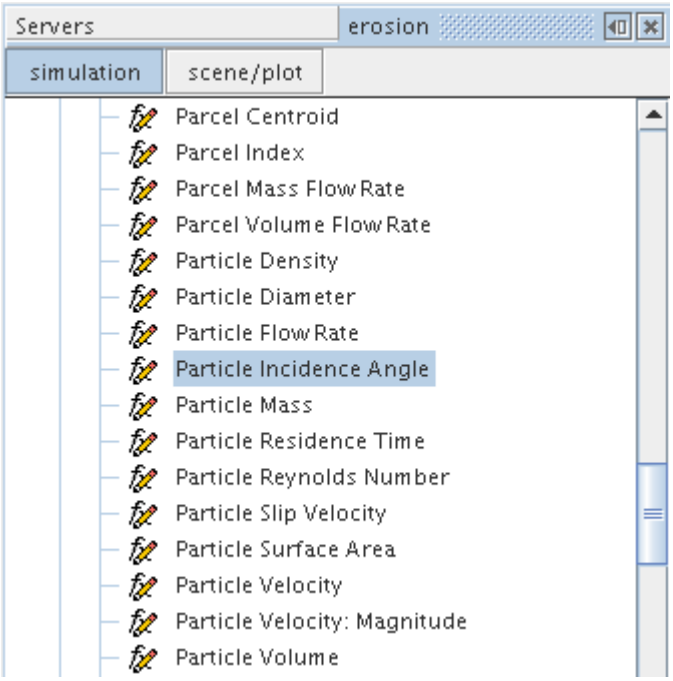
Represent these equations in STAR-CCM+ using user field functions. The quantities referred to in user field functions must come from the predefined system field functions or other user field functions. In this simulation you require a predefined system field function for the particle incidence angle.

To define field functions for the restitution coefficients:

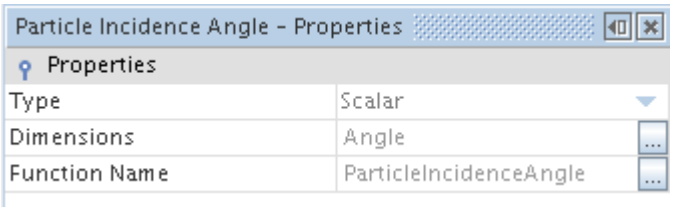
1. Open the **Tools** > **Field Functions** node.



2. Scroll down the list until the **Particle Incidence Angle** function is visible.



3. Select **Particle Incidence Angle** and note the value of the *Function Name* property.



Having noted the required name of **ParticleIncidenceAngle**, it is now possible to create the function for the Normal Restitution Coefficient.

4. Right-click the **Tools** > **Field Functions** node and select **New** > **Scalar** from the pop-up menu.

   A new **User Field Function 1** node appears in the object tree.
5. Rename **User Field Function 1** to `ENormal`.
6. Select **ENormal** and set the following properties:

Property	Setting
Function Name	`NormalRestitutionCoefficient`
Dimensions	*Dimensionless*
Definition	`0.988 - 0.78* $ParticleIncidenceAngle + 0.19*(pow($ParticleIncidenceAngle, 2)) - 0.024*(pow($ParticleIncidenceAngle, 3)) + 0.027*(pow($ParticleIncidenceAngle, 4))`

The same steps are now repeated for the tangential restitution coefficient.

7. Right-click the **Tools** > **Field Functions** node and select **New** > **Scalar** from the pop-up menu.

   A new **User Field Function 1** node appears in the object tree.
8. Rename **User Field Function 1** to `ETan`.
9. Select **ETan** and set the following properties:

Property	Setting
Function Name	`TangentialRestitutionCoefficient`
Dimensions	*Dimensionless*
Definition	`1 - 0.78*$ParticleIncidenceAngle + 0.84*(pow($ParticleIncidenceAngle, 2)) - 0.21*(pow($ParticleIncidenceAngle,`

Property	Setting
	`3))` `+ 0.028*(pow($ParticleIncidenceAngle, 4)) - 0.022*(pow($ParticleIncidenceAngle, 5))`

Setting the Lagrangian Phase Boundary Conditions

Define the Lagrangian phase boundary conditions.

The Lagrangian Phase Boundary Conditions are used to define the behavior of particles and droplets when they hit a solid boundary. Impact behavior can be defined for the following boundary types:

- Baffle
- Contact (interface)
- Wall

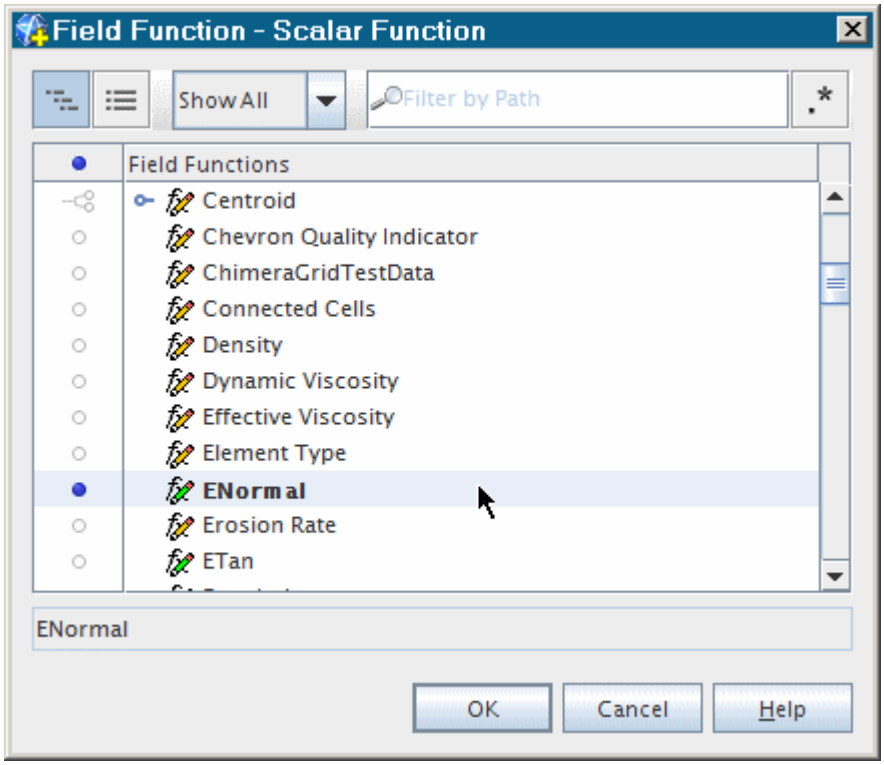
In this simulation, it is only necessary to define the boundary interaction models for the Wall boundary types, as there are no baffles or contact interfaces present. Parameters for the Erosion model are made in this section.

The field functions that you created in the previous section are now applied to the Lagrangian Phase Boundary Conditions.

To set the Lagrangian phase boundary conditions:

1. Edit the **Physics 1** > **Models** > **Lagrangian Multiphase** > **Lagrangian Phases** > **Phase 1** > **Boundary Conditions** > **Wall** node and set the following properties:

Node	Property	Setting
Physics Conditions > **Mode**	**Active Mode**	**Rebound**
Physics Values > **Normal Restitution Coefficient**	**Method**	**Field Function**
	Scalar Function	**ENormal**
└ **Tangential Restitution Coefficient**	**Method**	**Field Function**
	Scalar Function	**ETan**



2. Save the simulation.

Setting the Erosion Model

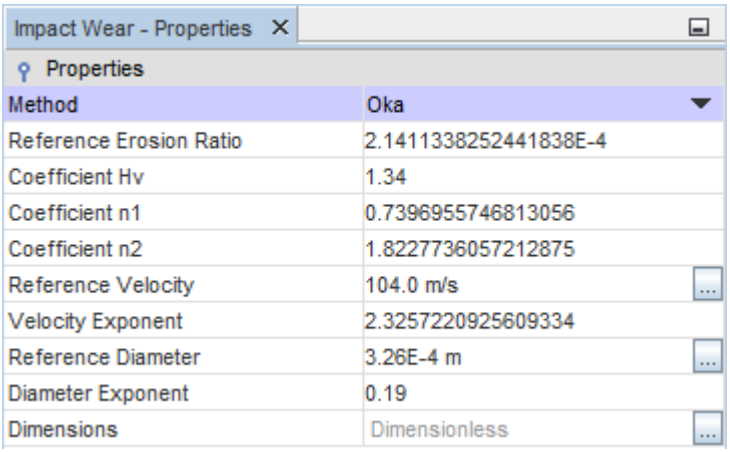
Select the appropriate erosion model.

When modeling erosion using CFD methods, it is necessary to select an erosion model that is based on a knowledge of the material undergoing erosion and the conditions under which the erosion is taking place. It is also important to appreciate that erosion modeling does not necessarily predict material loss with a consistent degree of accuracy and it is not recommended to estimate component lifetimes using CFD modeling alone. If estimating component lifetime is important, then additional experimental validation should be obtained to assess the effectiveness of the erosion model under appropriate conditions.

In this tutorial, use the Oka model initially, and use an additional user-defined model afterwards for comparison. The default coefficients that are given for the Oka model in STAR-CCM+ are for the sand erosion of carbon steel.

To set the erosion model:

1. In the **Continua** > **Physics 1** > **Models** > **Lagrangian Multiphase** > **Lagrangian Phases** > **Phase 1** > **Boundary Conditions** > **Wall** > **Physics Values** node, select the **Impact Wear** node and set *Method* to **Oka**.



2. Review the model properties.
3. Save the simulation.

Setting Continuous Phase Boundary Conditions

Define the continuous phase boundary conditions.

The default continuous phase boundary conditions for the wall boundaries are sufficient for this analysis, and there are no user-definable parameters for the symmetry planes. However, the inlet and outlet boundaries must be modified.

To set the continuous phase boundary conditions:

1. Edit the **Regions** > **Fluid** > **Boundaries** > **Inflow** node and set the following properties:

Node	Property	Setting
Physics Conditions > **Turbulence Specification**	**Method**	**Intensity + Length Scale**
└ **Velocity Specification**	**Method**	**Components**
Physics Values > **Turbulence Intensity**	**Value**	`0.029`
└ **Turbulent Length Scale**	**Value**	`0.0037 m`
└ **Velocity**	**Value**	`[0.0, -12.901, 0.0] m/s`

2. Edit the **Regions** > **Fluid** > **Boundaries** > **Outflow** node and set the following properties:

Node	Property	Setting
Physics Conditions > **Turbulence Specification**	**Method**	**Intensity + Length Scale**
Physics Values > **Turbulence Intensity**	**Value**	`0.029`

Node	Property	Setting
└ **Turbulent Length Scale**	**Value**	`0.0037 m`

3. Save the simulation.

Setting Up the Injector

Define the injector that is used to introduce the particles into the solution domain.

The particles of the dispersed Lagrangian phase are introduced into the solution domain using an injector. The injector defines the initial state of the particles and their spatial distribution. In this tutorial, introduce particles near the inlet with a velocity matching that of the continuous phase.

A new derived part based on a presentation grid is created to define the injection points, as this improves the distribution of particles across the pipe section.

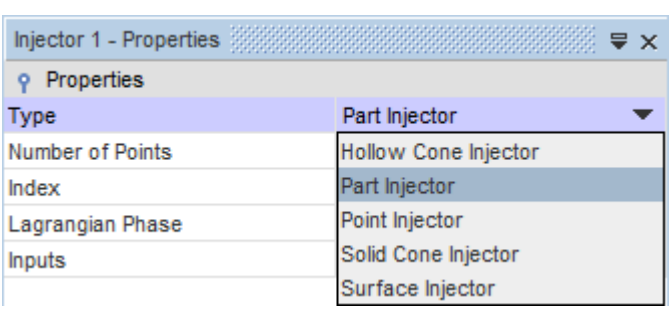
To set up the injector:

1. Right-click the **Derived Parts** node and select **New Part** > **Probe** > **Presentation Grid...**
2. In the *Create Presentation Grid* dialog, add the region Fluid to the **Input Parts**, click **Create**, and then **Close**.
3. Rename the **Derived Parts** > **Presentation Grid** node to `Injection Grid`.
4. Select the **Injection Grid** node and set the following values for the grid:

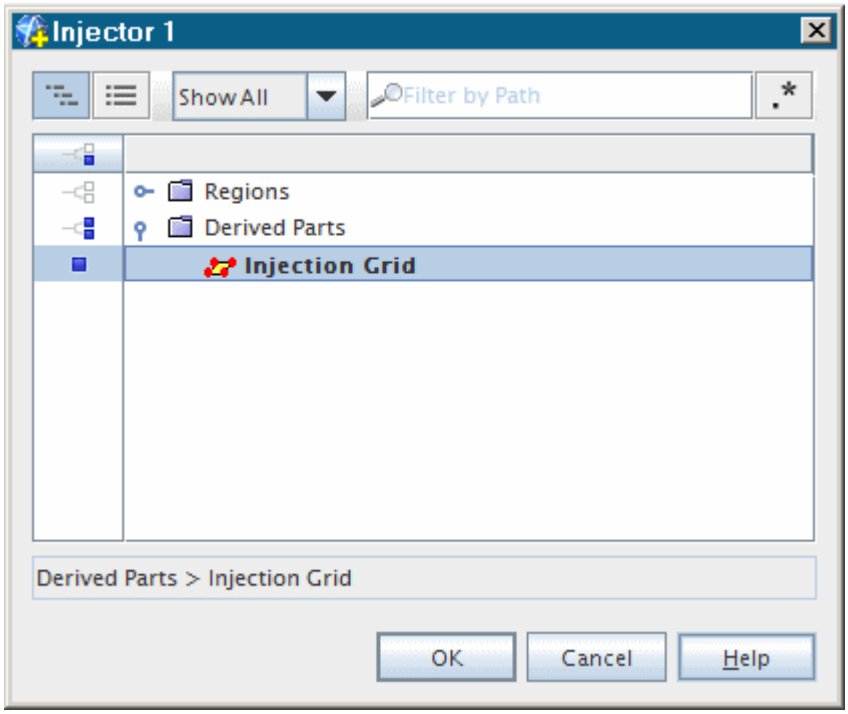
X Resolution	`100`
Y Resolution	`100`
Coordinate System	**Laboratory**
Point 1	`[0.11065, 0.22, -0.0265] m,m,m`
Point 2	`[0.137, 0.22, 0.0] m,m,m`
Normal	`[0.0, -1.0, 0.0] m,m,m`
Origin	`[0.0, 0.22, 0.0] m,m,m`

Create the injector:

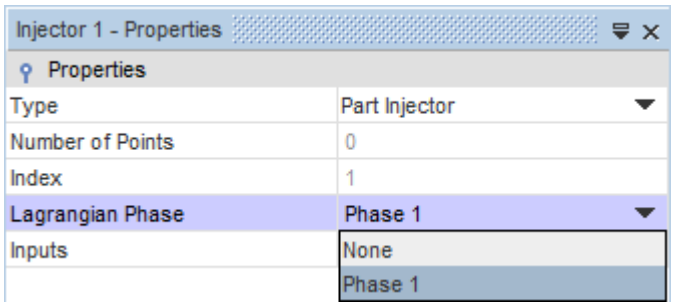
5. Right-click the **Injectors** node and select **New**.
6. Select the **Injector 1** node and set *Type* to **Part Injector**.



7. Set *Inputs* to **Derived Parts** > **Injection Grid**.



8. Click **OK**.
9. Finally set *Lagrangian Phase* to **Phase 1** as shown.



The *Number of Points* property is 0 until the injection points are calculated during solution initialization. The conditions and values of the injector must now be set up to define the particles.

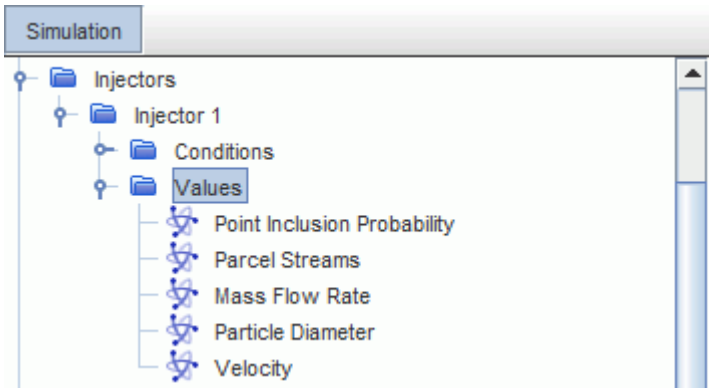
10. Edit the **Injectors** > **Injector 1** > **Conditions** node and set the following properties:

Node	Property	Setting
Flow Rate Specification	*Method*	**Mass Flow Rate**
Flow Rate Distribution	*Method*	**Per Injector**
Particle Size Specification	*Method*	**Particle Size**
Velocity Specification	*Method*	**Components**

The particular values for these quantities can now be set.

11. Edit the **Injectors** > **Injector 1** > **Values** node and set the following properties:

Node	Property	Setting
Velocity	*Value*	`[0.0, -12.901, 0.0] m/s`
Mass Flow Rate	*Value*	`0.027787 kg/s`
Parcel Streams	*Value*	`3`
Particle Diameter	*Value*	`2.75E-4 m`



12. Save the simulation.

Setting Solver Parameters and Stopping Criteria

Set the appropriate solver parameters and stopping criteria for the simulation.

In a Lagrangian multiphase simulation, it is important to set the maximum time for which a particle is tracked, as it is possible for particles to become trapped within a fluid region (such as a stagnant zone) and remain there indefinitely.

To set the solver parameters and stopping criteria:

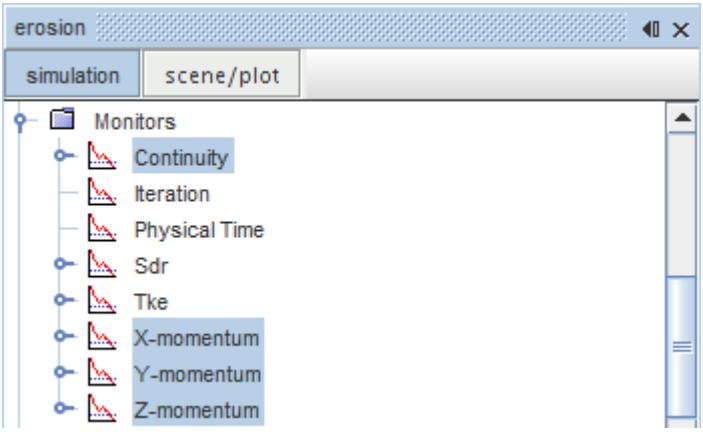
1. Edit the **Solvers** node and set the following property:

Node	Property	Setting
Lagrangian Multiphase > **Steady**	**Maximum Residence Time**	`0.1 s`
	Verbosity	**High**

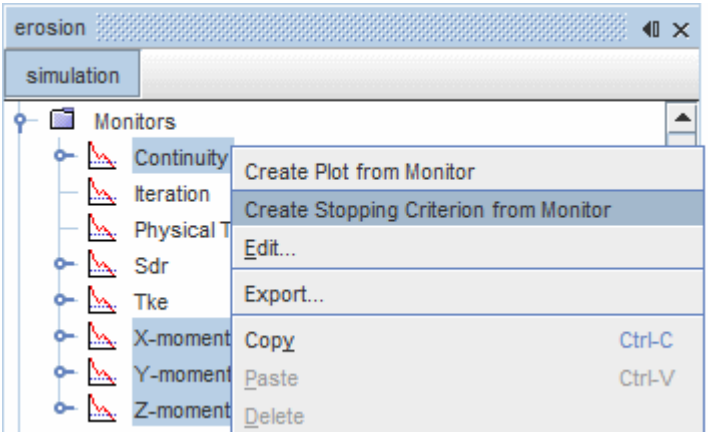
New stopping criteria are created based on the monitors that appeared when the segregated flow solver was selected. The goal is to achieve a residual of 1.0E-3 for the continuity and momentum quantities before stopping the solution.

2. Open the **Monitors** node.
3. Multi-select the following nodes:

- **Continuity**
- **X-momentum**
- **Y-momentum**
- **Z-momentum**

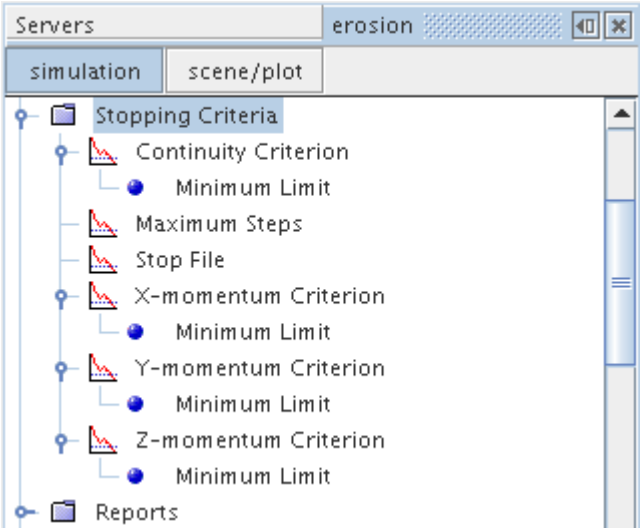


4. Right-click one of the selected nodes and select **Create Stopping Criterion from Monitor**.



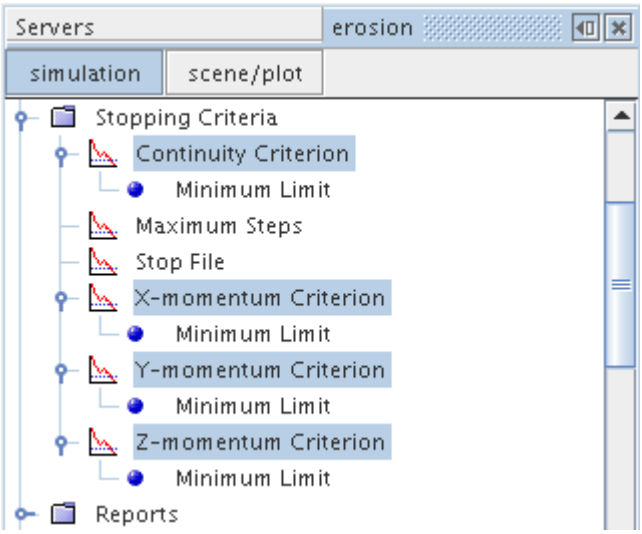
5. Open the **Stopping Criteria** > **Continuity Criterion** node.

   A stopping criterion has been created for each of the monitors selected previously.

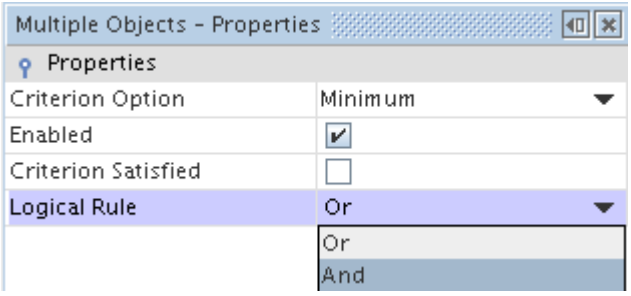


It is possible to change the properties for all the nodes in one operation, rather than modifying each node individually.

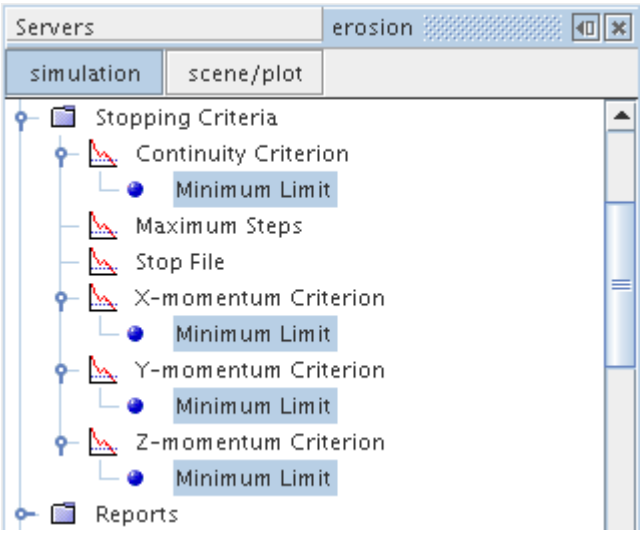
6. Multi-select the following nodes:
   - **Continuity Criterion**
   - **X-momentum Criterion**
   - **Y-momentum Criterion**
   - **Z-momentum Criterion**



7. In the *Properties* window, set **Logical Rule** to **And**.



8. Multi-select the following nodes:
   - **Continuity Criterion** > **Minimum Limit**
   - **X-momentum Criterion** > **Minimum Limit**
   - **Y-momentum Criterion** > **Minimum Limit**
   - **Z-momentum Criterion** > **Minimum Limit**



9. In the *Properties* window, set **Minimum Value** to `0.001`.
10. Save the simulation.

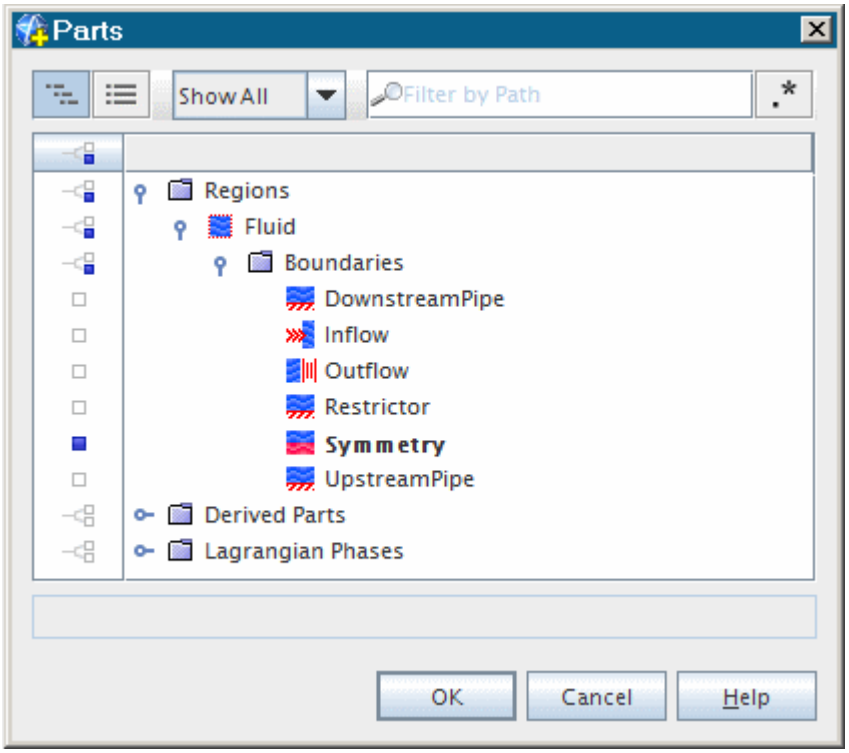
Visualizing the Solution

Create a scalar scene to view the results of the simulation.

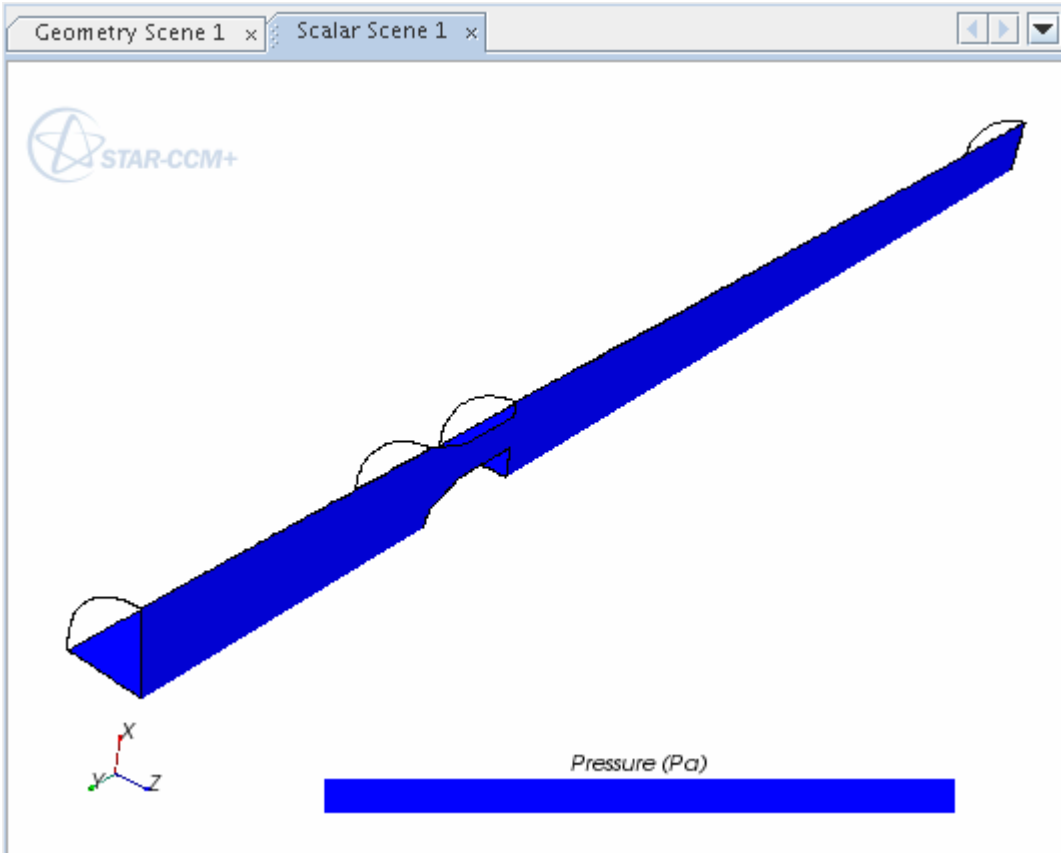
Observe the pressure distribution on the symmetry planes as the solution develops.

To visualize the solution:

1. Right-click the **Scenes** node and select **New Scene** > **Scalar**.
2. Select the **Scenes** > **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Parts** node and set *Parts* to **Regions** > **Fluid** > **Boubndaries** > **Symmetry**.



3. Click **OK**.
4. Select the **Scalar 1** > **Scalar Field** node and set *Function* to **Pressure**.
5. Use the mouse buttons to orient *Scalar Scene 1* as shown below.



6. Select the **Color Bar** node and set the following:

Setting	Value
Orientation	**Vertical** (Set this before setting *Width* and *Height*.)
Width	`0.1`
Height	`0.7`
Position	`[0.77, 0.11]`

Reporting and Plotting the Pressure Drop

Create reports of average pressure at the inlet and outlet boundaries, and use the reports in a field function that gives the pressure drop.

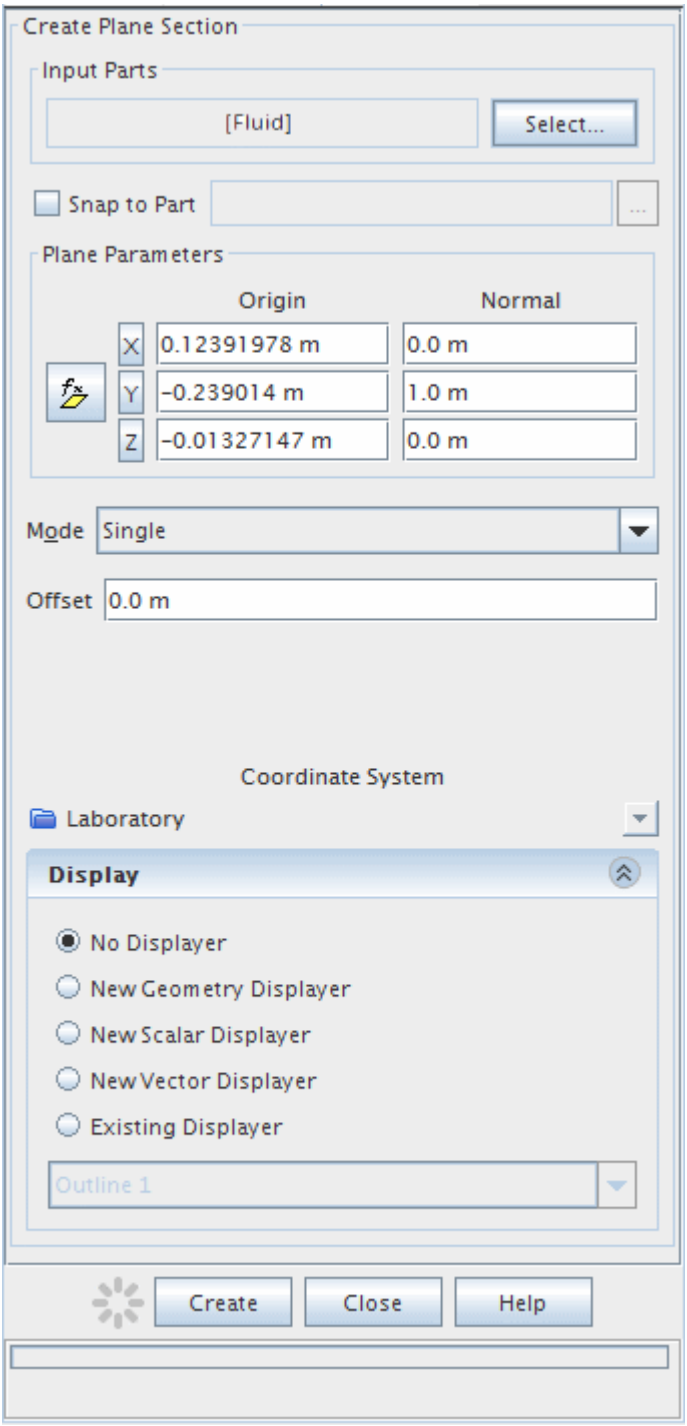
Experimental measurements are available for the pressure drop across the flow restrictor when the flowrate is 28.09 liters/s. These can be compared to the prediction of pressure drop for this simulation to give some estimate of the accuracy of results. To do this, it is necessary to create reports of average pressure at the inlet and outlet boundaries, and to use these reports in a user-defined field function that gives the pressure drop. This field function can be plotted while running the simulation to estimate convergence.

The upstream static pressure was measured at a distance of two pipe diameters before the restrictor, and the downstream static pressure was measured at a distance of six pipe diameters after the restrictor. The inflow

boundary corresponds to the upstream measurement, but a section plane must be created at the point of the downstream measurement. The section plane is created first:

1. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane...**

   The *Create Plane Section* dialog appears in place above the object tree.
2. In the *Input Parts* box, select Fluid.
3. In the *Plane Parameters* box, set *Y origin* to `-0.239014 m`, the *X normal* to `0 m`, and the *Y normal* to `1 m`. The other parameters can be left as they are.
4. In the *Display* box, select **No Displayer** as shown below.

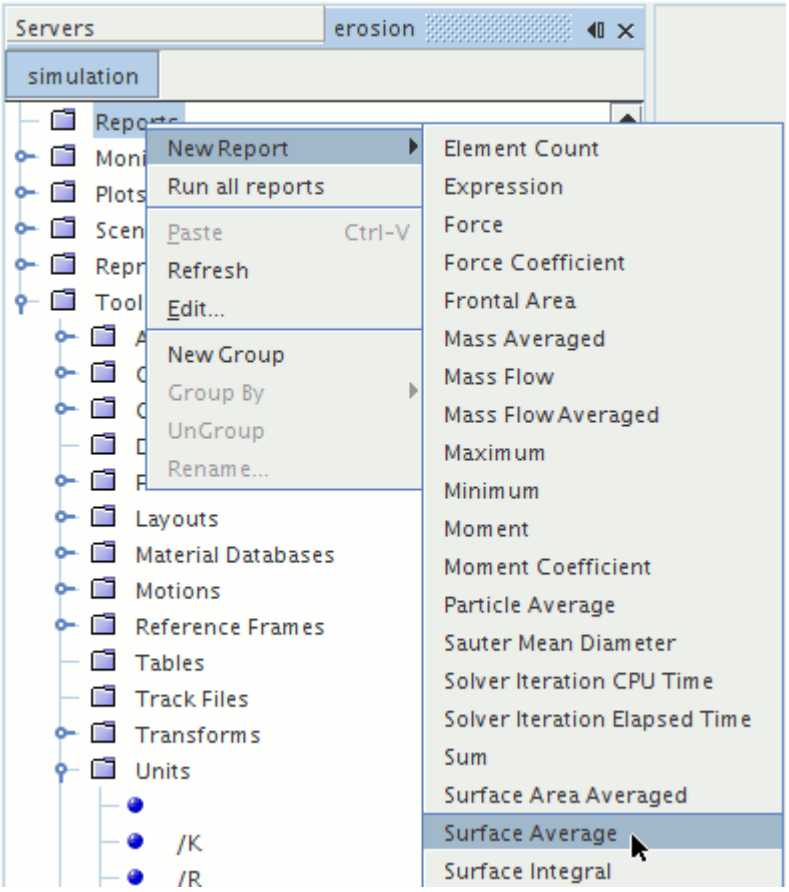


5. Click **Create**, and then **Close**.

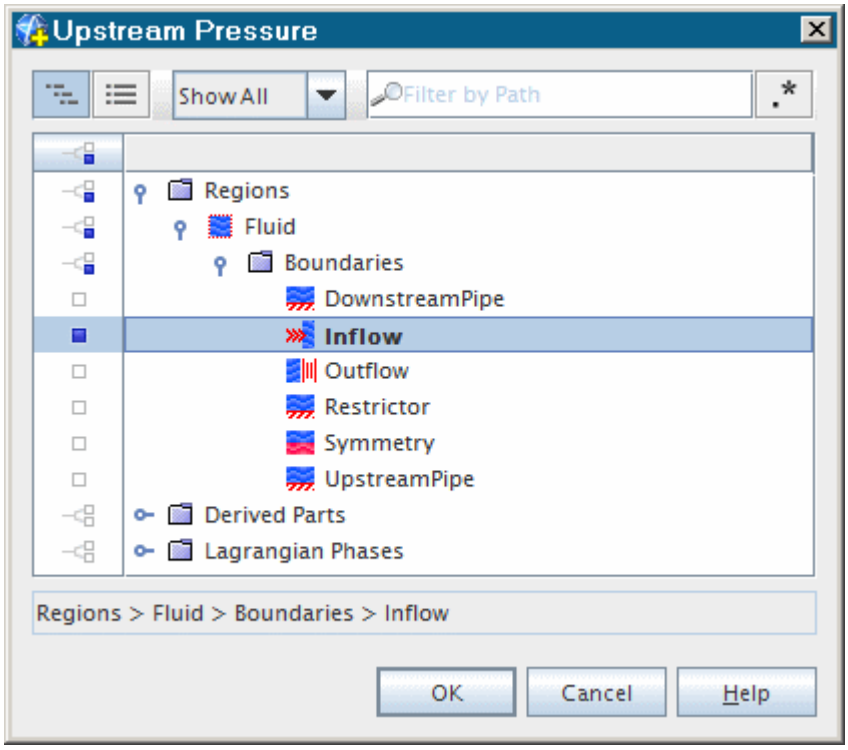
   A new part that is called **Plane Section** is added to the object tree inside the **Derived Parts** node.

Create the reports that are used to extract the static pressure:

6. Right-click the **Reports** node and select **New Report** > **Surface Average**.



7. Rename the **Surface Average 1** node to `Upstream Pressure`.
8. Select the **Upstream Pressure** node and set *Scalar Field Function* to **Pressure**.
9. Click the right side of the *Parts* property. In the dialog that appears, expand the **Regions** and **Fluid** nodes and select the **Inflow** boundary as shown below.

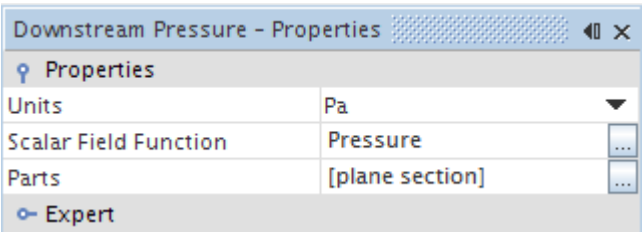


10. Click **OK**.

A similar report is created for the downstream pressure:

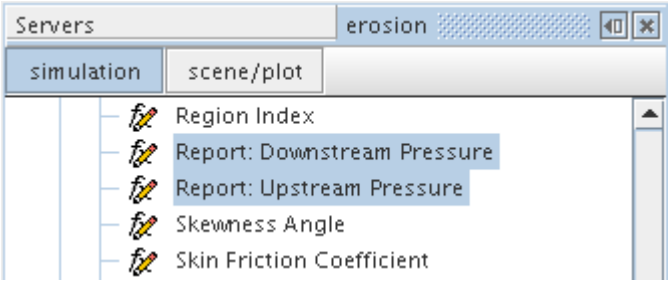
11. Right-click the **Reports** node and select **New Report** > **Surface Average**
12. Rename the **Surface Average 1** node to `Downstream Pressure`.
13. Select the **Downstream Pressure** node and set the *Scalar Field Function* to **Pressure** and the *Parts* to **Plane Section**.

    The completed *Properties* window appears as shown below.



Two new field functions corresponding to these reports have been automatically added to the object tree.

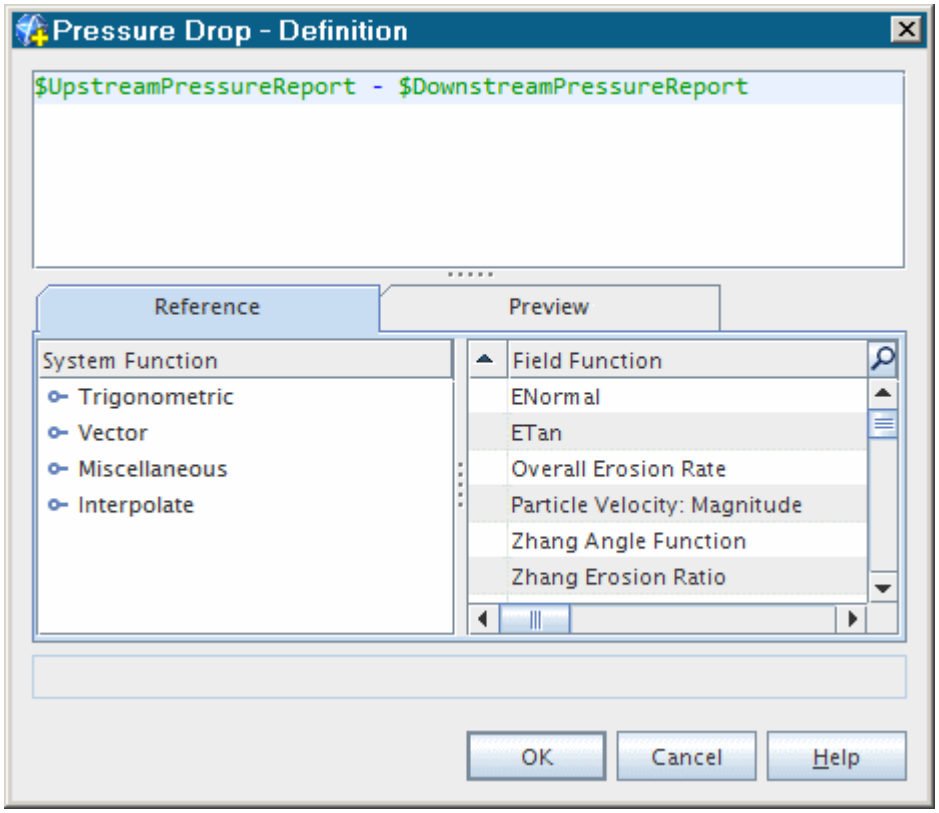
14. Open the **Tools** > **Field Functions** node and scroll down to see the new nodes:



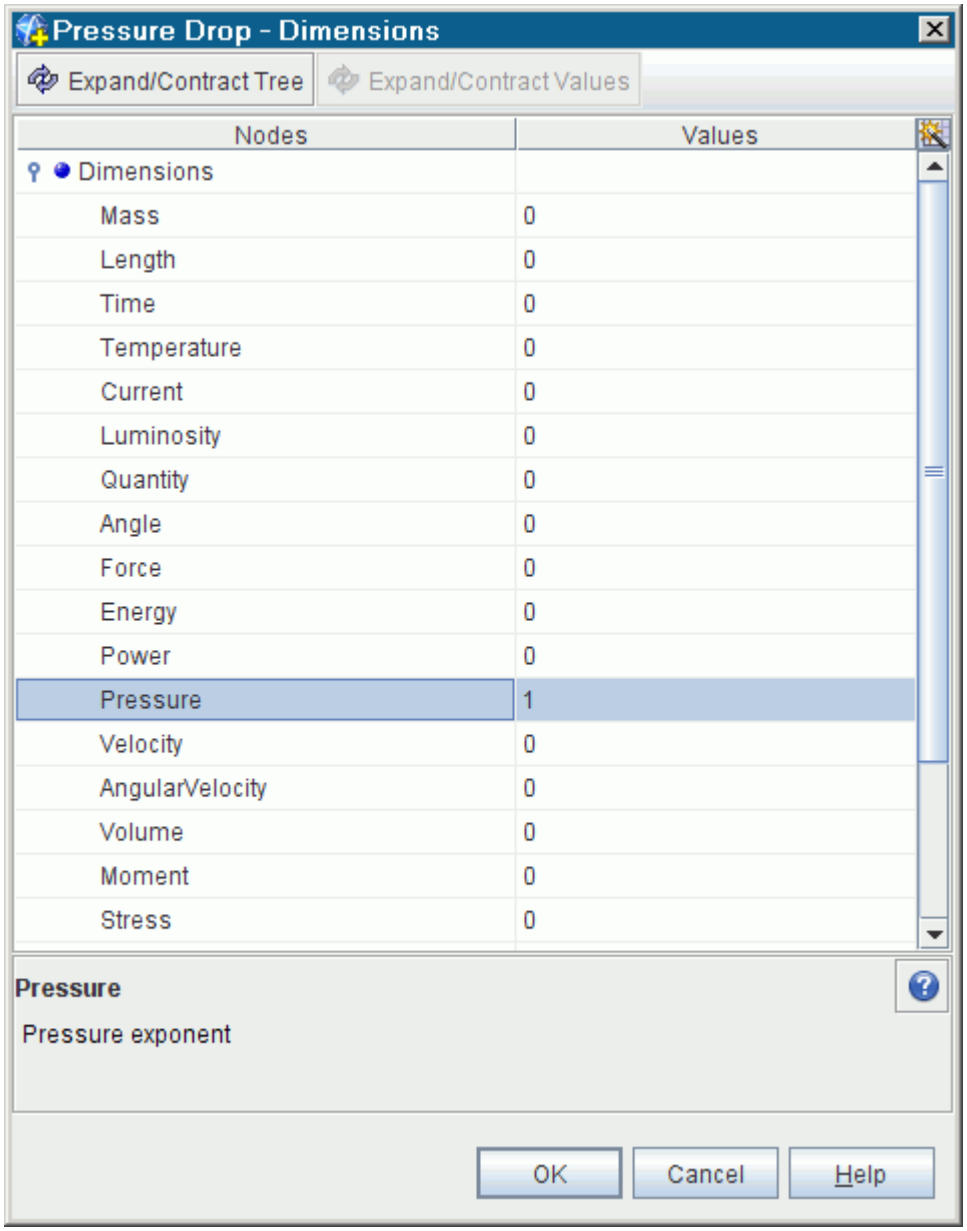
Examination of the properties for these nodes shows that the relevant field function names are **DownstreamPressureReport** and **UpstreamPressureReport** respectively. These function names are used to create an expression report for the pressure drop:

15. Right-click the **Reports** node and select **New Report** > **Expression**.
16. Rename the **Expression 1** node to `Pressure Drop`.
17. Select the **Pressure Drop** node.
18. Click ... (**Custom Editor**) to enter the *Definition* property as:

    `$UpstreamPressureReport - $DownstreamPressureReport`
19. Click **OK**.

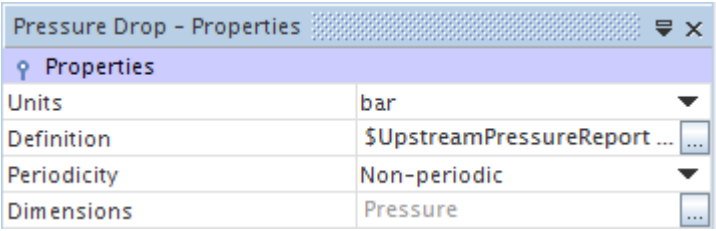


20. Click ... (**Custom Editor**) for the *Dimensions* property. In the *Pressure Drop - Dimensions* dialog, set the **Pressure** dimension to `1` as shown below.



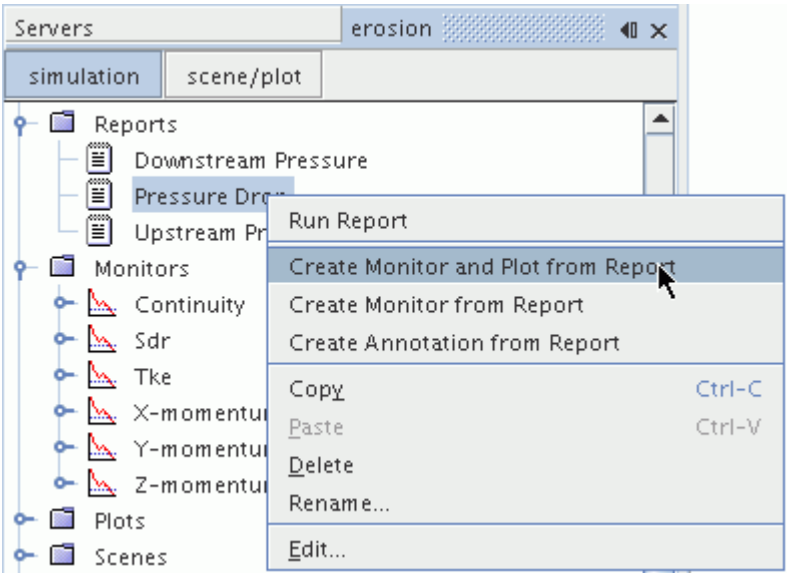
21. Click **OK**.
22. Finally, set the *Units* to `bar`.

    The *Properties* window for the **Pressure Drop** report appears as shown below.

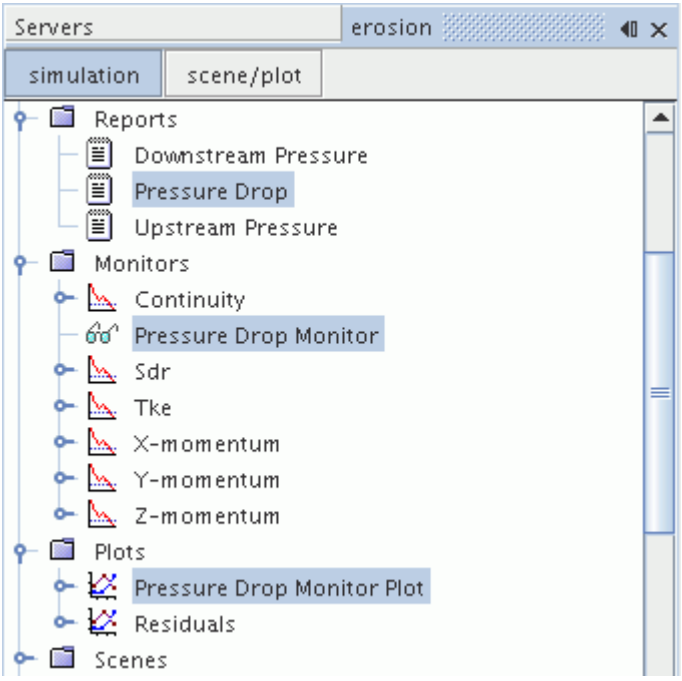


A monitor and plot is created from this report.

23. Right-click the **Reports** > **Pressure Drop** node and select **Create Monitor and Plot from Report**.



The resulting monitor and plot can be seen in the object tree:



24. Right-click the **Plots** > **Pressure Drop Monitor Plot** and select **Open** from the pop-up menu. The empty plot is displayed.
25. Save the simulation.

Initializing and Running the Simulation

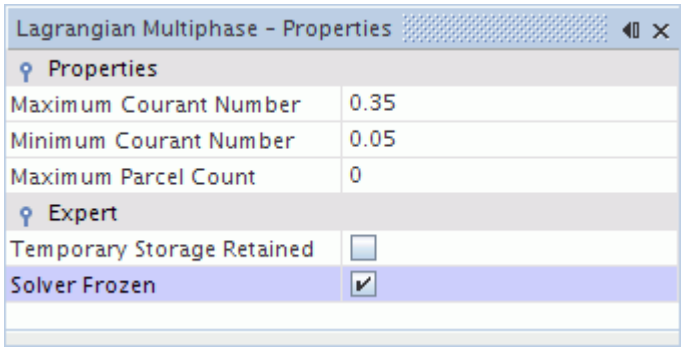
Preparation of the simulation is now complete, and the simulation can be run.

Run the simulation in two stages. A steady state solution for the fluid phase in the absence of particles is obtained first, and then the particle tracks are obtained with a single additional iteration. This is a reasonable approach as two-way coupling has not been activated for the Lagrangian phase.

Fluid Phase

To deactivate the Lagrangian solver while the fluid phase solution is obtained:

1. Select the **Solvers** > **Lagrangian Multiphase** node and activate the **Solver Frozen** expert property.



To initialize the solution and run the simulation:

2. Click (**Initialize Solution**) in the *Solution* toolbar or select **Solution** > **Initialize Solution** from the menu.

3. Click (**Run**) in the *Solution* toolbar.

   The solution progress is displayed in the *Output* window. You can ignore the warning message about smoothing in the *Output* window. The *Residuals* display is created automatically in the *Graphics* window and shows the progress of the solvers.

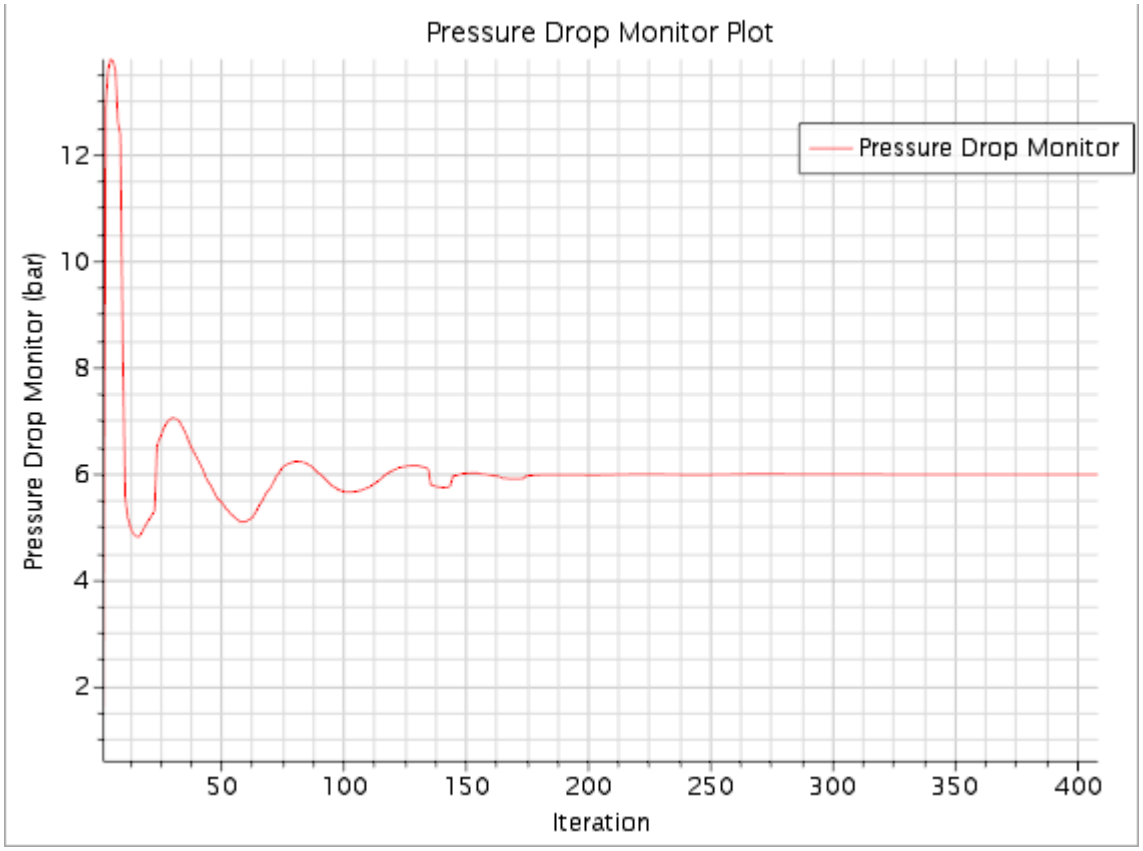
   While the simulation is running, you can click the tabs at the top of the *Graphics* window to view scenes and plots.

4. Select the *Scalar Scene 1* tab to view the pressure distribution as the solution develops.

   During the run, you can stop the process by clicking (**Stop**) on the toolbar. If you do stop the simulation, you can click (**Run**) to resume it.

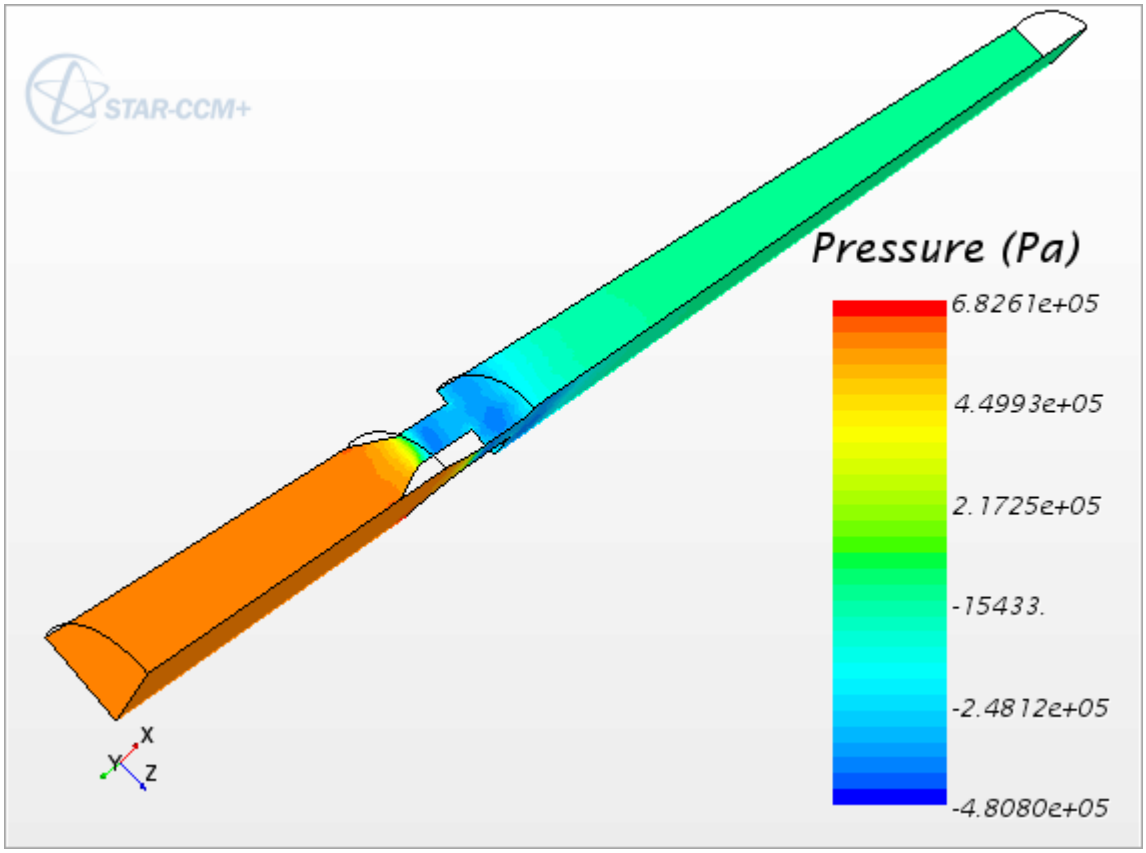
   If left alone, the simulation continues until 1000 iterations are complete or the continuity residual criterion, 0.001, is satisfied for all appropriate quantities.

   The *Pressure Drop Monitor Plot* appears as shown below at the end of the simulation.



5. Click the *Scalar Scene 1* tab at the top of the *Graphics* window.

   A scene similar to the one shown below appears.



6. When the simulation has finished running, save it.

Lagrangian Phase

The Lagrangian Multiphase solver is reactivated to obtain the particle tracks and erosion predictions.

7. Select the **Solvers** > **Lagrangian Multiphase** node.
8. In the *Properties* window, deselect **Solver Frozen**.
9. Click (**Step**) in the *Solution* toolbar to perform a single iteration.
   It can take several minutes for the solver to compute all the particle tracks.
10. When the particle tracking is complete, save the simulation.

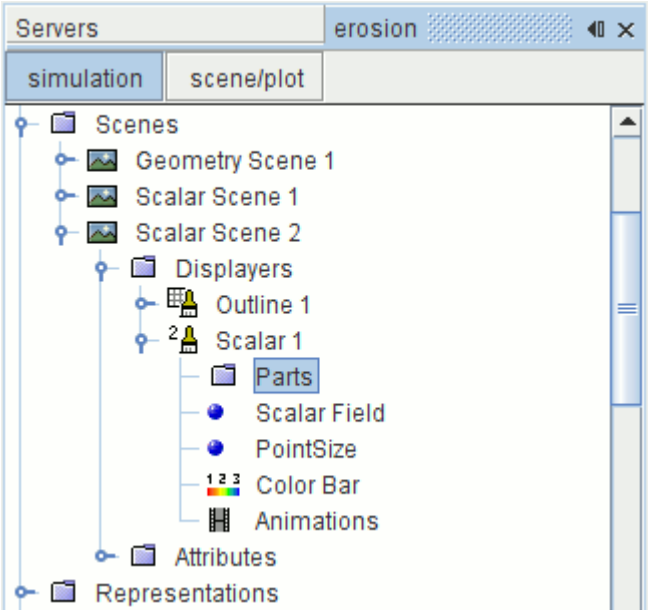
Visualizing the Results

Create a scalar scene to view the results of the simulation.

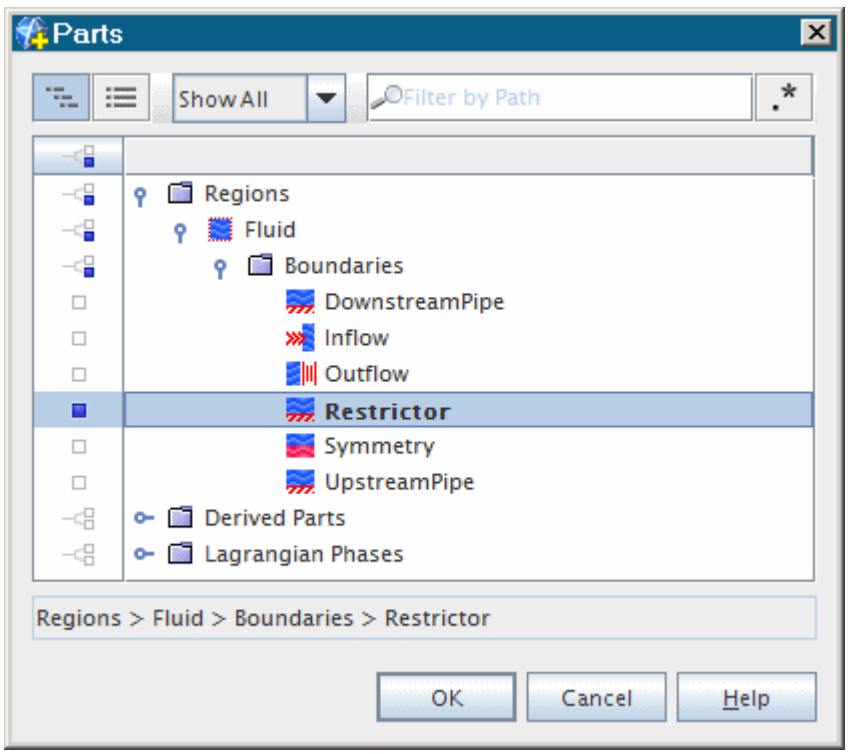
The main result of interest in this tutorial is the erosion rate across the restrictor surface. A scalar scene can be created showing this profile.

To visualize the results:

1. Right-click the **Scenes** node and select **New Scene** > **Scalar**.
2. Open the **Scenes** > **Scalar Scene 2** node and select the **Displayers** > **Scalar 1** > **Parts** node.



3. Set *Parts* to **Regions** > **Fluid** > **Restrictor**.

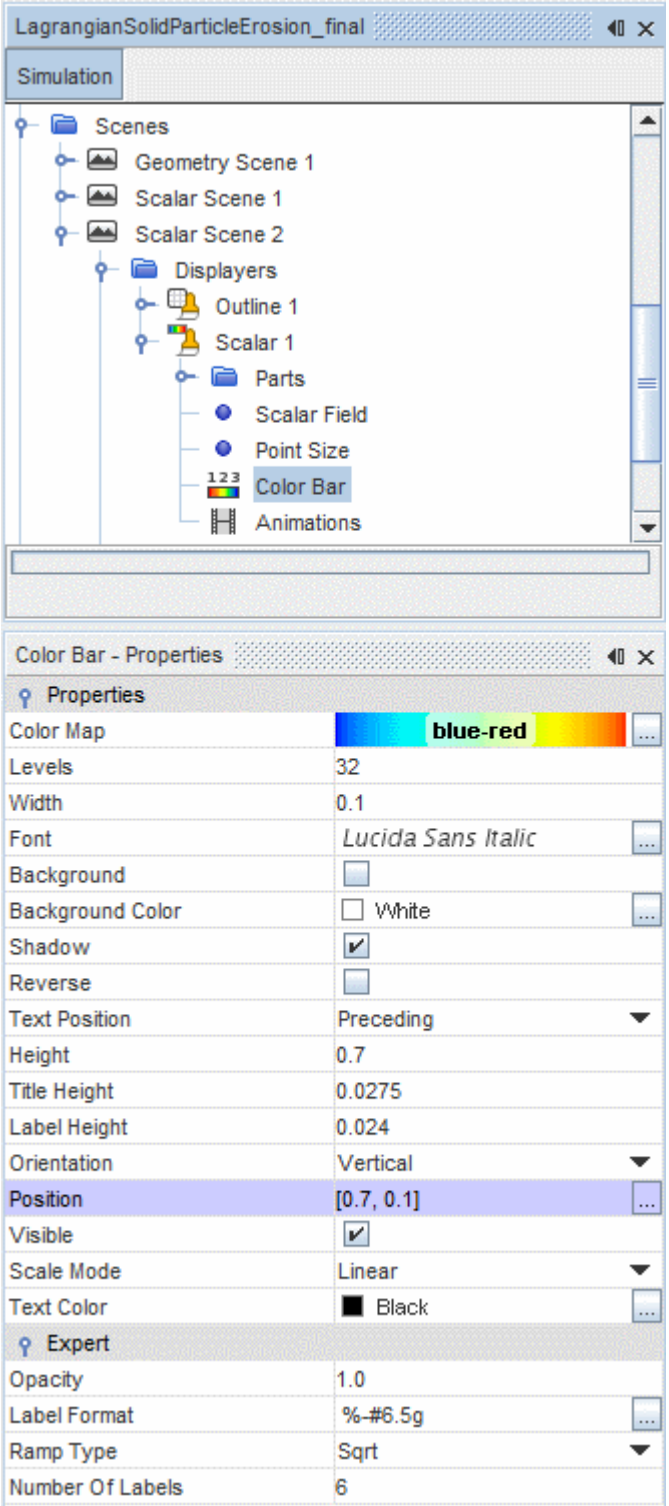


4. Continuing within the same **Displayers** > **Scalar 1** node, select the **Scalar Field** node.
5. Set *Function* to **Erosion Rate**.
6. Select the **Scenes** > **Scalar Scene 2** > **Displayers** > **Outline 1** node and clear the *Outline* property.

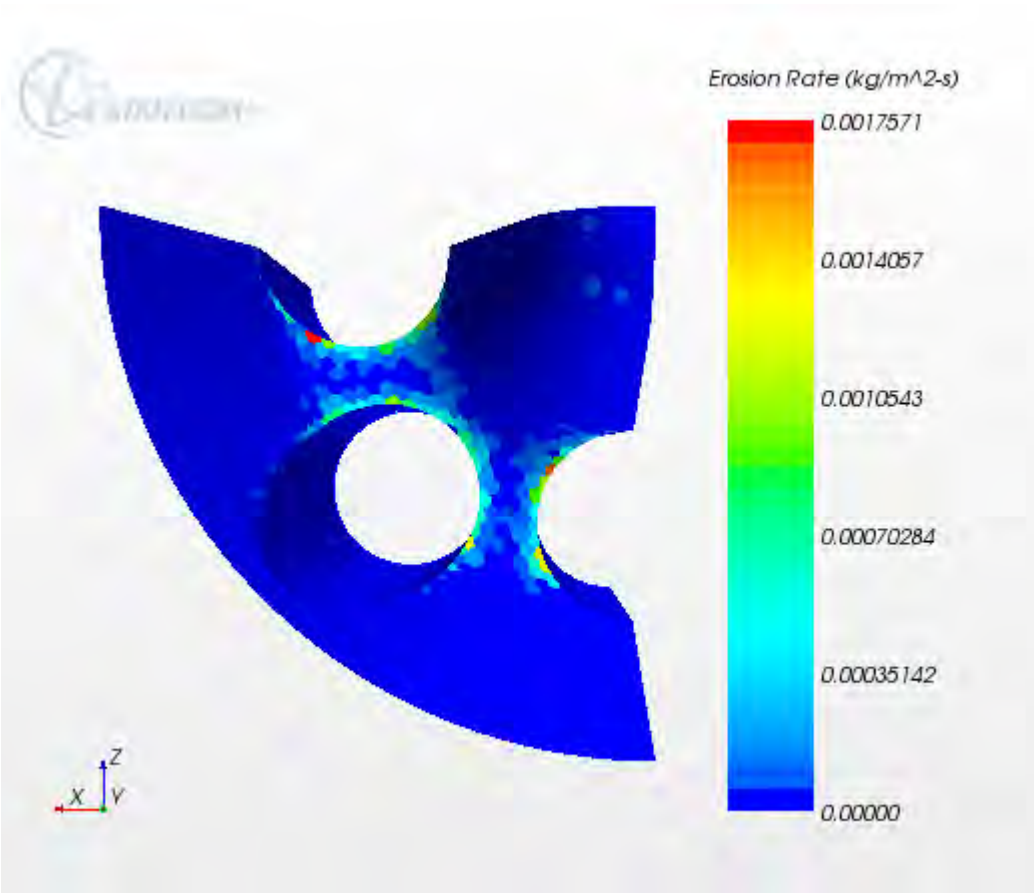
The view can be initialized using (**Save-Restore-Select Views**).

7. Click (**Save-Restore-Select Views**) and select **View** > **+Y** > **Up +Z** from the pop-up menus.
8. Select the **Color Bar** node and set the following:

Setting	Value
Orientation	**Vertical** (Set before setting *Width* and *Height*.)
Width	`0.1`
Height	`0.7`
Position	`[0.7, 0.1]`



9. Use (**Rubberband Zoom**) and mouse controls to zoom in on the eroded restrictor surface and position it as shown below.



The plot shows where the maximum erosion rates are expected.

Estimating the Overall Erosion Rate

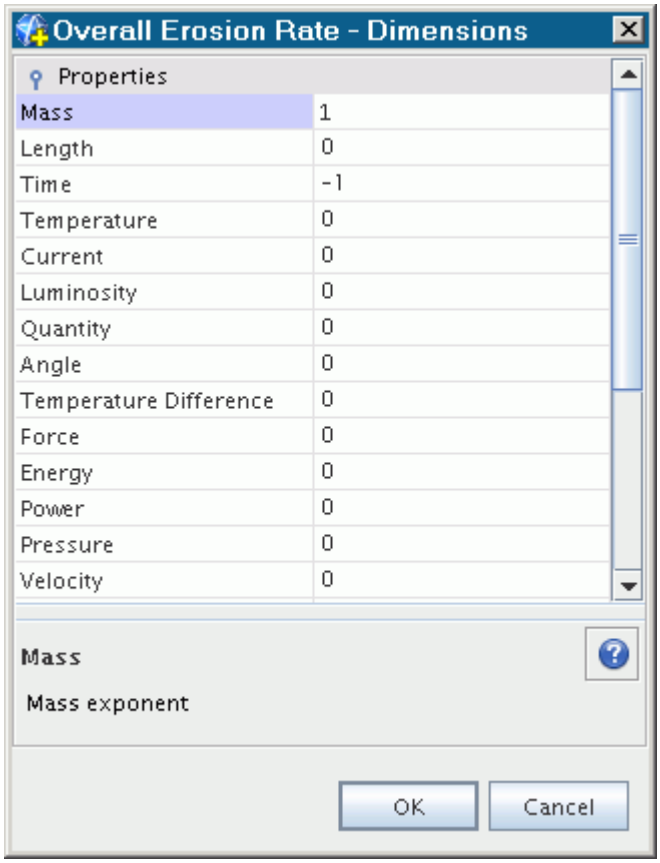
Create a field function and report to estimate the overall erosion rate.

To obtain an estimate of the erosion rate in g/hr for the whole restrictor, it is necessary to create a field function that multiplies the computed erosion rate in each boundary face by the face area.

To estimate the overall erosion rate:

Create the field function.

1. Right-click the **Tools** > **Field Functions** node and select **New** > **Scalar**.
2. Rename the **User Field Function 1** node to `Overall Erosion Rate`.
3. Edit the *Dimensions* property and set *Mass* to `1` and *Time* to `-1` as shown.

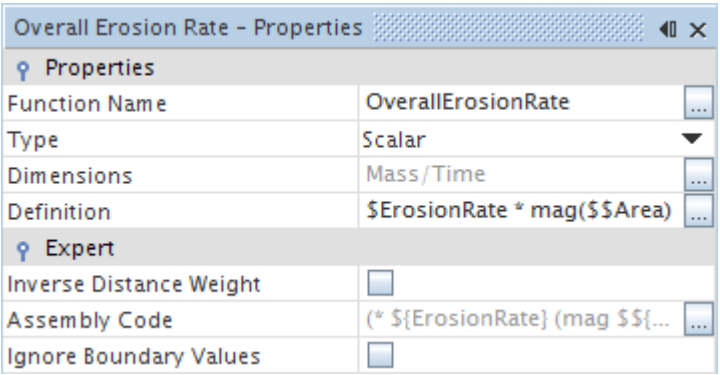


4. Click **OK**.
5. Set the *Function Name* to `OverallErosionRate`.
6. Edit the *Definition* property and enter the following definition in the dialog:

   ```
   $ErosionRate * mag($$Area)
   ```

7. Click **OK**.

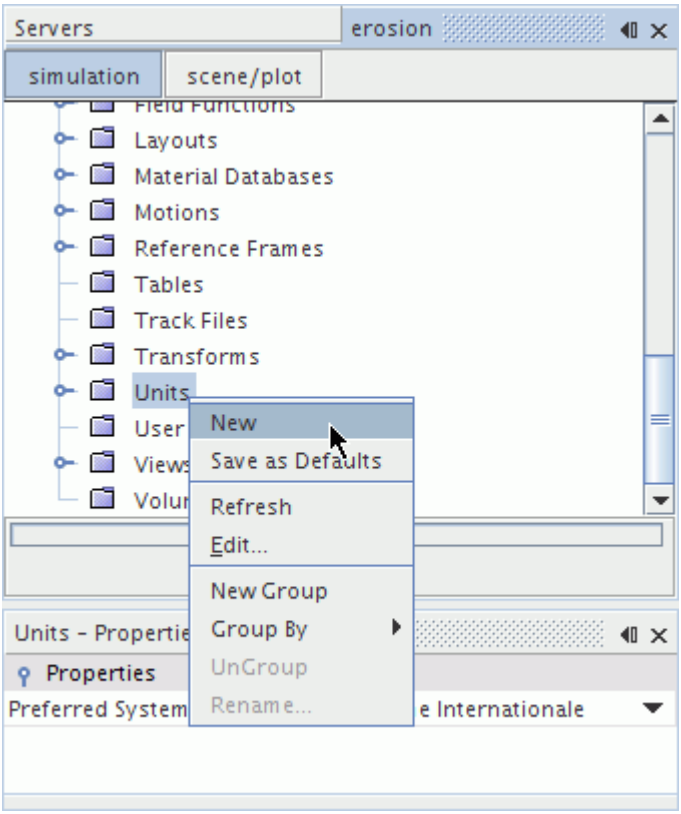
   The completed *Properties* window appears as shown below.



It is possible to report the overall erosion rate in units other than kg/s by creating a user-defined unit. This is demonstrated by creating a unit for g/hr. The conversion factor is:

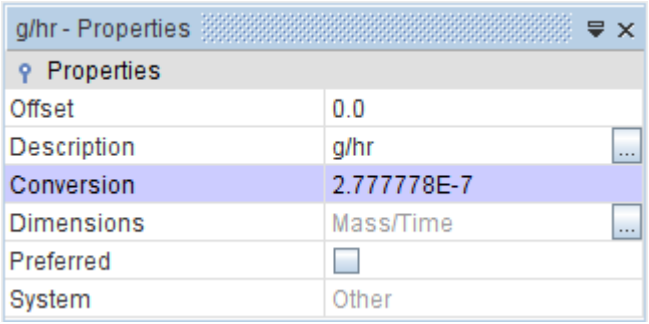
1 g/hr = 2.777778 x 10^{-7} kg/s

8. Right-click the **Tools** > **Units** node and select **New** from the pop-up menu.



9. Rename the **Units 1** node to `g/hr`.
10. Select the **g/hr** node and set *Description* to `g/hr`.
11. Edit the *Dimensions* property and set *Mass* to `1`.
12. Set *Time* to `-1`.
13. Click **OK**.
14. Enter the *Conversion* property as `2.777778E-7`.

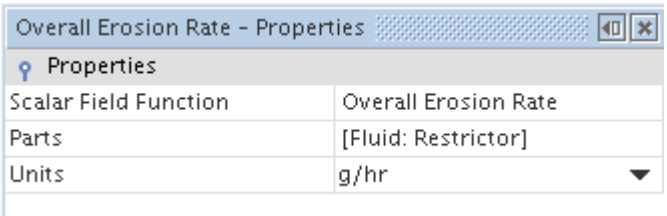
    The completed *Properties* window is as shown below.



Create a report that uses the field function and unit:

15. Right-click the **Reports** node and select **New Report** > **Sum**.
16. Rename the **Sum 1** node to `Overall Erosion Rate`.
17. Set the *Scalar Field Function* to **Overall Erosion Rate**.
18. Click the right half of the *Parts* property, expand **Regions** and **Fluid** nodes and select **Restrictor**.
19. Set the *Units* to **g/hr** using the drop-down menu.

    The completed *Properties* window appears as shown below.



20. Right-click the **Overall Erosion Rate** report node and select **Run Report** from the pop-up menu.

    See the *Output* window for the report result. A typical value is `1.858468e-01 g/hr` .

The predicted quantity is for a quarter annulus only. To get the full erosion rate for the real case, it is necessary to multiply by 4. This gives a predicted erosion rate of 0.75 g/hr for the whole component.

Comparison with Experimental Data

The simulation results are compared with experimental data.

Pressure Drop

For flow tests conducted at a flow rate of 28.09 liters/s, the pressure drop across the restrictor was found to be 5.44 bar. This value compares favorably with the predicted pressure drop of 5.99 bar at 28.57 liters/s obtained with the relatively coarse mesh that is used in this tutorial.

Erosion Rate

An erosion test for the restrictor device that is considered in this tutorial gave an average erosion rate of 0.58 g/hr over a 14 hour period. The flow rate and sand concentration that are used in this tutorial match those of the erosion test, and the predicted erosion rate using the Oka model is 0.75 g/hr.

Implementing Alternative Functions for Erosion Ratio

Erosion equations are often tuned for a particular material/sand combination that are based on experimental testing. In this section, implement the model that was developed by Zhang et al. at the Erosion/Corrosion Research Center in Tulsa as an example.

The model that is given by Zhang et al. ([796]) is:

$$ER = C(BH)^{-0.59} F_s V_p^n F\left(\Theta\right)$$

$$F(\Theta) = 5.40\Theta - 10.11\Theta^2 + 10.93\Theta^3 - 6.33\Theta^4 + 1.42\Theta^5$$

where:

- *ER* is the Erosion Ratio.
- *C* and *n* are empirical constants that are given as 2.17×10^{-7} and 2.41 respectively.
- *BH* is the Brinell hardness of the wall material. For carbon steel, this is taken as 120 after Zhang et al. ([796]).
- F_s is the particle shape coefficient, which is said to be 0.2 for fully rounded sand particles.
- V_p is the particle impact velocity.
- Θ is the particle impact angle in radians.

Implement the angle function in a field function, and reference this angle field function as a field function for erosion ratio:

1. Right-click the **Tools** > **Field Functions** node and select **New** > **Scalar**.
2. Rename the **User Field Function 1** node to `Zhang Angle Function`.
3. Select the **Zhang Angle Function** node and set the following properties:

Property	Setting
Function Name	`ZhangAngleFunction`
Dimensions	*Dimensionless*
Definition	`(5.4 * $ParticleIncidenceAngle) - (10.11 * pow($ParticleIncidenceAngle, 2)) + (10.93 * pow($ParticleIncidenceAngle, 3)) - (6.33 * pow($ParticleIncidenceAngle, 4)) + (1.42 * pow($ParticleIncidenceAngle, 5))`

If you insert the numeric constants in the equation:

$$ER = C(BH)^{-0.59} F_s V_p^n F(\Theta)$$

the final equation to be implemented is:

$$ER = 2.17 \times 10^{-7} \cdot (120)^{-0.59} \cdot 0.2 \cdot V_p^{2.41} F(\Theta)$$

$$ER = 2.57497 \times 10^{-9} V_p^{2.41} F(\Theta)$$

4. Right-click the **Tools** > **Field Functions** node and select **New** > **Scalar**.
5. Rename the **User Field Function 1** node to `Zhang Erosion Ratio`.
6. Select the **Zhang Erosion Ratio** node and set the following properties:

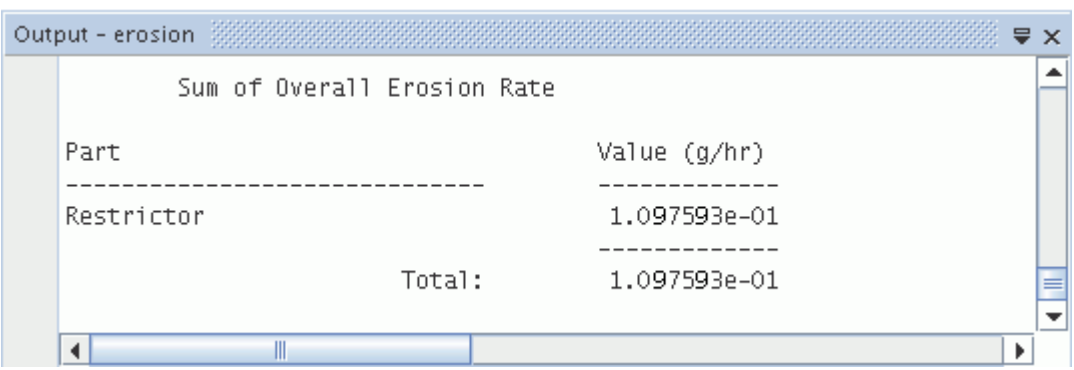
Property	Setting
Function Name	`ZhangErosionRatio`
Dimensions	*Dimensionless*
Definition	`2.57497E-9 * pow((mag($${ParticleVelocity})), 2.41) * $ZhangAngleFunction`

The field function must now be set in the Lagrangian model tree:

7. Select the **Continua** > **Physics 1** > **Models** > **Lagrangian Multiphase** > **Lagrangian Phases** > **Phase 1** > **Boundary Conditions** > **Wall** > **Physics Values** > **Impact Wear** node and set *Method* to **Field Function**.
8. Set *Scalar Function* to **Zhang Erosion Ratio**.

With the new field function in place, it is only necessary to run a single iteration to obtain a further prediction of erosion.

9. Click (**Step**) in the **Solution** toolbar to perform a single iteration. It can take several minutes for the solver to compute all the particle tracks.
10. Save the simulation.
11. Run the **Overall Erosion Rate** report to obtain the mass loss rate from the new result:



The erosion rate for the whole annulus is four times the predicted amount, which is 0.41 g/hr for the Zhang erosion model.

Summary

This tutorial demonstrated how to set up an erosion modeling analysis in STAR-CCM+.

This tutorial has introduced the following STAR-CCM+ features:

- Selecting models for a Lagrangian phase.
- Setting up an injector for Lagrangian Multiphase modeling.
- Defining material properties that are required for the selected models.
- Defining boundary conditions.

- Setting solver parameters.
- Initializing and running the solver until the residual magnitudes are sufficiently small.
- Analyzing results using the visualization facilities.
- Analyzing results using field functions.

Bibliography

This tutorial references the following papers.

[793] *McCulloch, S., 2005. United States Patent No. US 6,886,598 B2, “Choke Restrictor Devices and Methods”.*

[794] *Wallace, M.S., 2001. “CFD-based erosion modeling of simple and complex geometries.” PhD Thesis, University of Strathclyde.*

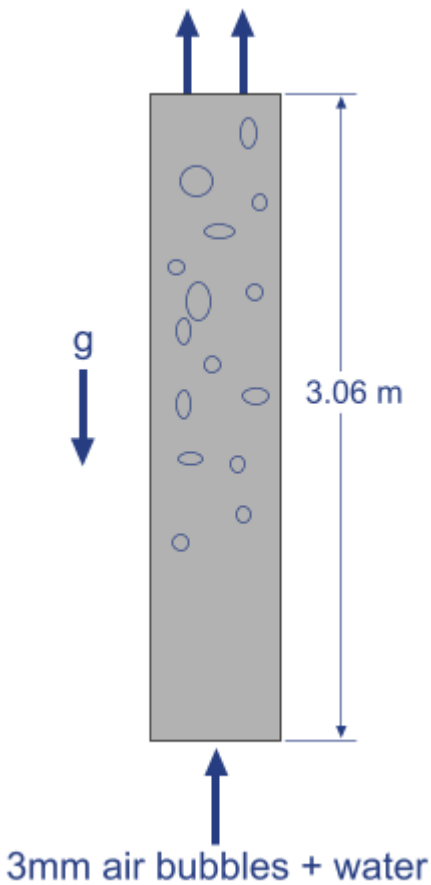
[795] *Forder, A., Thew, M., Harrison, D., 1998. A numerical investigation of solid particle erosion experienced within oilfield control valves. Wear, Vol. 216, pp. 184-193.*

[796] *Zhang, Y., Reuterfors, E.P., McLaury, B.S., Shirazi, S.A., Rybicki, E.R., 2007. “Comparison of computed and measured particle velocities and erosion in water and air flows.” Wear, Vol. 263, pp. 330-338.*

Eulerian: Hibiki's Bubble Column

This tutorial demonstrates how to model a Eulerian multiphase turbulent flow in STAR-CCM+.

This simulation models Hibiki's bubble flow column [797], which involves air and water phases being pumped through a vertical column. The geometry is shown in the diagram below.



The first part of the tutorial covers how to select the Eulerian multiphase and multiphase turbulence models, specify the properties of each phase and define the phase interaction.

In the second part of the tutorial, the S-Gamma model is activated to account for the size distribution of bubbles in the dispersed phase (air). The S-Gamma model predicts the Sauter mean diameter, which in turn is used to set the phase interaction length scale.

A validation section is also included where the results of the two simulations are compared with experimental data.

Contents:

Prerequisites
Importing the Mesh and Naming the Simulation

Visualizing the Mesh
Selecting the Physics Models
Creating Phases and Selecting Phase Models
Defining the Phase Interactions
Setting Initial Conditions and Reference Values
Setting Boundary Conditions
Setting Stopping Criteria and Solver Parameters
Setting Up the Plots
Running the Simulation
Exporting the Plot Data
Using the S-Gamma Model in the Simulation
Validating the Results
Summary
Bibliography

Prerequisites

The instructions in the Eulerian: Hibiki's Bubble Column tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Importing the Mesh and Naming the Simulation

Launch STAR-CCM+, import the supplied geometry, and save the simulation.

To import the mesh and name the simulation:

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh**.
4. In the *Open* dialog, choose `doc/startutorialsdata/multiphaseFlow/data/axisymmetricVerticalPipe.ccm`, then click **Open**.

   STAR-CCM+ provides feedback on the import process, which takes a few seconds, in the *Output* window.
5. Save the simulation as `verticalPipe.sim`.

Visualizing the Mesh

Create a mesh scene and examine the mesh.

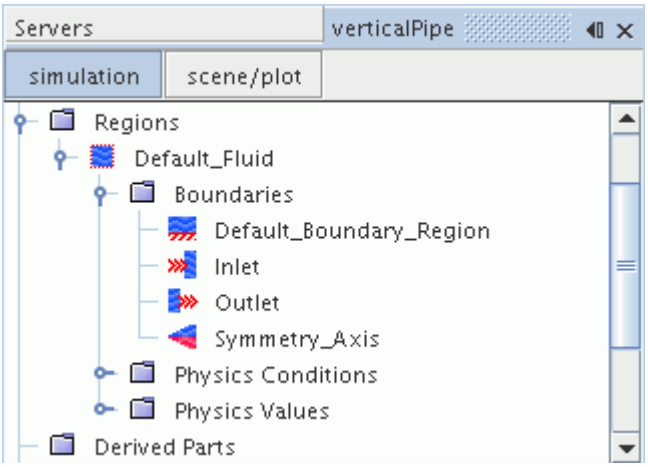
An axisymmetric hexahedral mesh of a vertical pipe has been prepared for this analysis, so further meshing is not required. The boundary names and types are pre-defined.

To visualize the mesh:

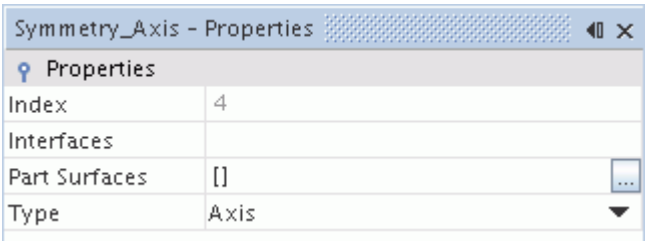
1. Create a mesh scene.

Examine the imported boundaries.

2. Expand the **Regions** > **Default Fluid** > **Boundaries** node and select the **Symmetry_Axis** node.



3. Make sure that the *Type* property is set to **Axis**.



4. Select the **Inlet** node and observe which part is highlighted in the scene. Make sure that the boundary *Type* is set to **Velocity Inlet**.
5. Select the **Outlet** boundary node and make sure that the *Type* is set to **Outlet**.

Selecting the Physics Models

Physics models define the primary variables of the simulation, including pressure, temperature, velocity, and what mathematical formulation is used to generate the solution.

The multiphase turbulence models are available for selection once the multiphase flow and material models have been selected.

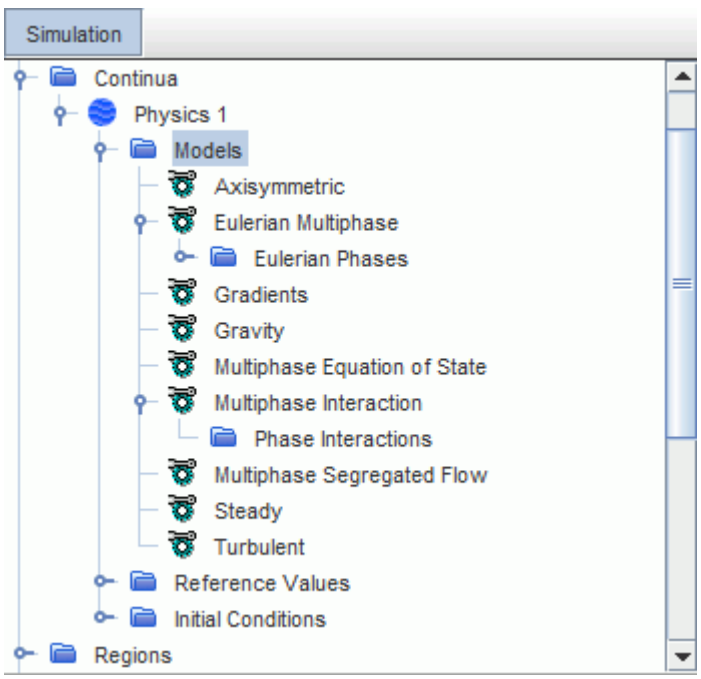
To select the physics models:

1. For the physics continuum, **Continua** > **Physics 1**, in the *Enabled Models* group box, deactivate **Two Dimensional**.

2. Select the following models in order:

Group Box	Model
Space	**Axisymmetric**
Time	**Steady**
Material	**Eulerian Multiphase** **Multiphase Interaction** (selected automatically)
Eulerian Multiphase Model	**Multiphase Segregated Flow** **Multiphase Equation of State** (selected automatically) **Gradients** (selected automatically)
Viscous Regime	**Turbulent**
Optional Models	**Gravity**

3. Click **Close**.
4. To review the models, open the **Physics 1** > **Models** node.



5. Save the simulation.

Creating Phases and Selecting Phase Models

Create the phases and select the appropriate phase models.

The two phases flowing through the section of vertical pipe are air and water. These phases are defined by creating new phases under the Eulerian physics model, and selecting the corresponding material properties. Create the liquid phase first:

To create the phases and select the phase models:

Create the liquid phase.

1. In the **Physics 1** continuum, right-click the **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and create a new phase.
2. Rename the **Phase 1** node to **Water**.
3. For the **Water** phase, select the following models:

Group Box	Model
Enabled Models	**Flow** (pre-selected) **Turbulent** (pre-selected)
Material	**Liquid** **Reynolds-Averaged Navier-Stokes** (selected automatically)

4. Deactivate **Auto-select recommended models**.
5. Select the following models in order:

Group Box	Model
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**
K-Epsilon Turbulence Models	**Realizable K-Epsilon Two-Layer** **Exact Wall Distance** (selected automatically)

6. Activate **Auto-select recommended models**.
7. Select the following models in order:

Group Box	Model
Enabled Models	**Two-Layer All y+ Wall Treatment** (selected automatically)
Equation of State	**Constant Density**

8. Click **Close**.

Create the gas phase.

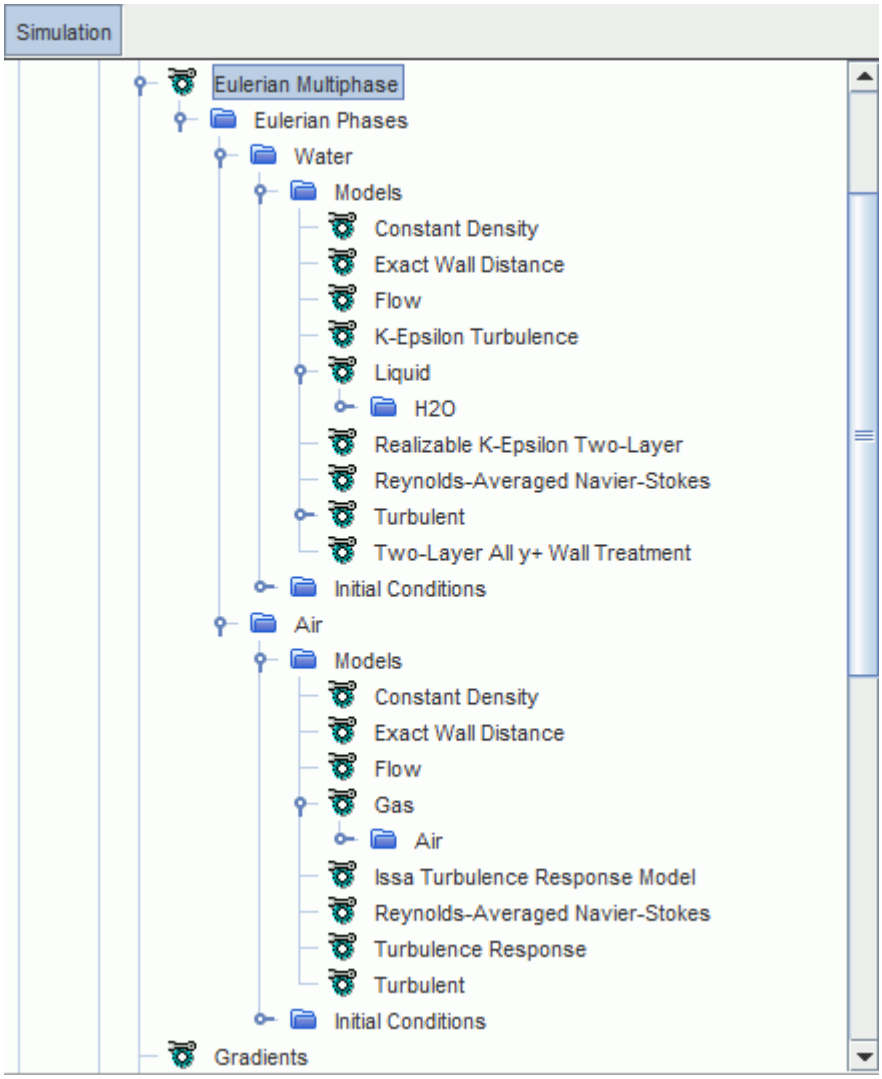
9. Create a second phase and rename it to **Air**.
10. For the **Air** phase, select the following models in order:

Group Box	Model
Enabled Models	**Flow** (pre-selected) **Turbulent** (pre-selected)
Material	**Gas**

Group Box	Model
	Reynolds-Averaged Navier-Stokes (selected automatically)
Reynolds-Averaged Turbulence	**Turbulence Response**
Turbulence Response Models	**Issa Turbulence Response Model** **Exact Wall Distance** (selected automatically)
Equation of State	**Constant Density**

11. Click **Close**.

    The **Eulerian Phases** node appears as shown below.



Specify the material properties for the liquid phase.

12. Edit the **Eulerian Phases** > **Water** > **Models** > **Liquid** > **H2O** > **Material Properties** node and set the following properties:

Node	Property	Setting
Density > **Constant**	**Value**	`1000 kg/m^3`
Dynamic Viscosity > **Constant**	**Value**	`0.001 Pa-s`

Specify the material properties for the gas phase.

13. Edit the **Eulerian Phases** > **Air** > **Models** > **Gas** > **Air** > **Material Properties** node and set the following properties:

Node	Property	Setting
Density > **Constant**	**Value**	`1 kg/m^3`
Dynamic Viscosity > **Constant**	**Value**	`1.8E-5 Pa-s`

14. Save the simulation.

Defining the Phase Interactions

Define the interactions between the phases.

Using the **Multiphase Interaction** model, you can define the interaction between the water and air phases.

To define the phase interactions:

1. In the **Physics 1** continuum, right-click the **Models** > **Multiphase Interaction** > **Phase Interactions** node and create a new phase interaction.
2. Rename the **Phase Interaction 1** node to **Water-Air**.
3. For the **Water-Air** phase interaction, deactivate **Auto-select recommended models**, and then select the following models in order:

Group Box	Model
Phase Interaction Topology	**Continuous-Dispersed Phase Interaction** **Drag Force** (selected automatically) **Multiphase Material** (selected automatically)
Interaction Length Scale	**Interaction Length Scale** (selected automatically)
Interaction Area Density	**Interaction Area Density** (selected automatically)
Optional Models	**Lift Force** **Virtual Mass Force** **Turbulent Dispersion Force**
Virtual Mass Coefficient	**Virtual Mass Coefficient**

4. Click **Close**.

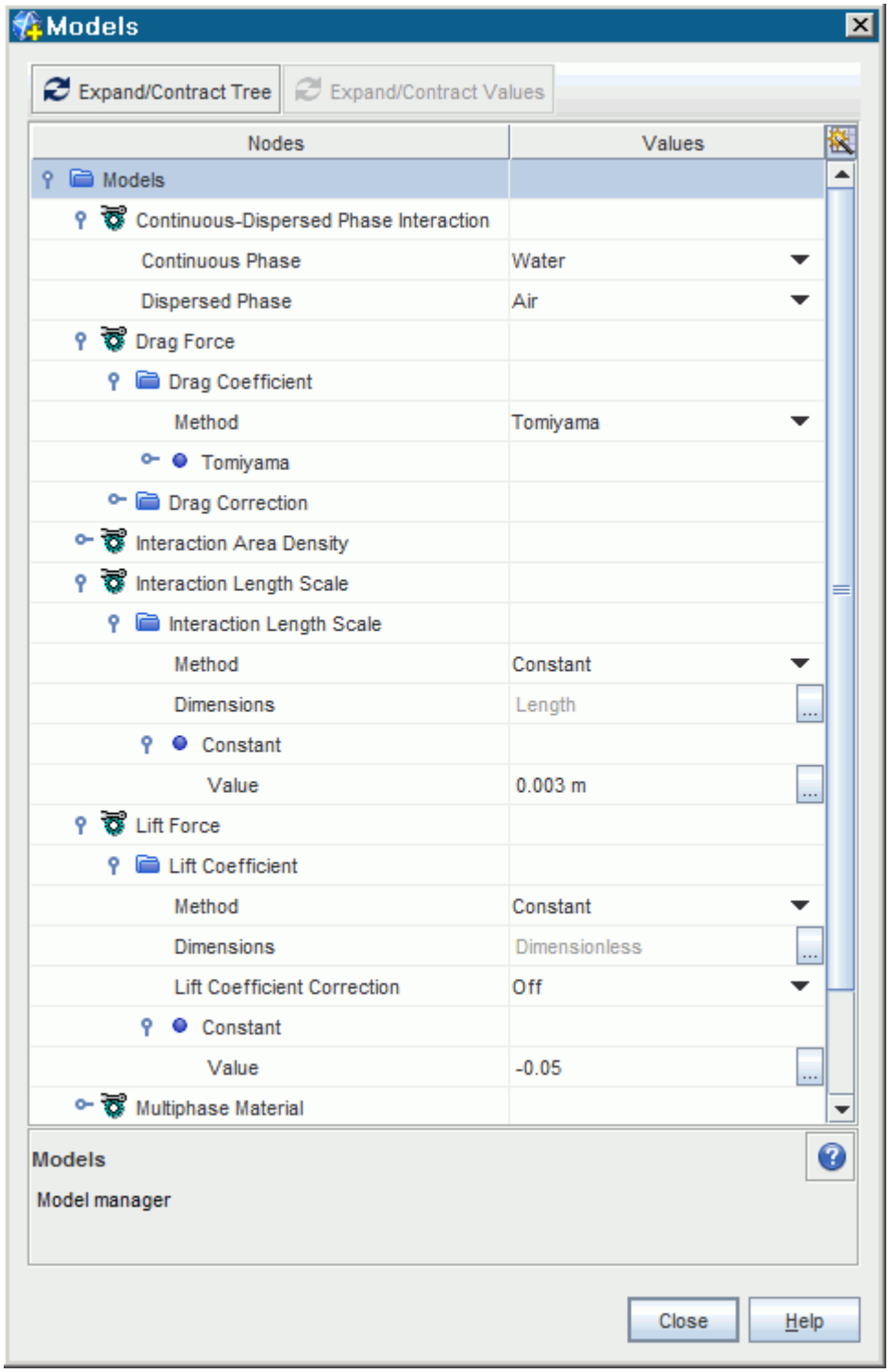
   The **Phase Interactions** > **Water-Air** > **Models** node appears as shown below.

Define the phase interaction properties.

5. Edit the **Water-Air** > **Models** node and set the following properties:

Node	Property	Setting
Continuous-Dispersed Phase Interaction	**Continuous Phase**	**Water**
	Dispersed Phase	**Air**
Drag Force > **Drag Coefficient**	**Method**	**Tomiyama**
Interaction Length Scale > **Interaction Length Scale** > **Constant**	**Value**	`0.003 m`
Lift Force > **Lift Coefficient** > **Constant**	**Value**	`-0.05`

The completed *Edit* dialog is shown below.



6. Edit the **Physics 1** > **Models** > **Eulerian Multiphase** > **Eulerian Phases** > **Air** > **Models** > **Issa Turbulence Response Model** node and set the *Phase Interaction* to **Water-Air**.
7. Save the simulation.

Setting Initial Conditions and Reference Values

Set the initial conditions and reference values for the physics continuum.

To set the initial conditions and reference values:

1. Edit the **Continua** > **Physics 1** > **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and set the following properties:

Node	Property	Setting
Water > **Initial Conditions**		
⌊ **Turbulence Specification**	**Method**	**Intensity + Length Scale**
⌊ **Turbulent Length Scale**	**Value**	`0.005 m`
⌊ **Velocity**	**Value**	`[1.23, 0.0] m/s`
⌊ **Volume Fraction**	**Value**	`0.8`
Air > **Initial Conditions**		
⌊ **Velocity**	**Value**	`[1.61, 0.0] m/s`
⌊ **Volume Fraction**	**Value**	`0.2`

Set the reference values for the **Physics 1** continuum.

2. Edit the **Reference Values** node and set the following properties:

Node	Property	Setting
Reference Density	**Value**	`1000 kg/m^3`
Gravity	**Value**	`[-9.81, 0.0] m/s^2`

3. Save the simulation.

Setting Boundary Conditions

Define the boundary types and specify the appropriate property values.

To set the boundary conditions:

1. Go to **Regions** > **Default_Fluid** > **Boundaries** > **Inlet** > **Phase Conditions**. Select **Water** > **Physics Conditions** > **Turbulence Specification** and set *Method* to **Intensity + Length Scale**.
2. Edit the **Regions** > **Default_Fluid** > **Boundaries** > **Inlet** > **Phase Conditions** node and set the following properties:

Node	Property	Setting
Water > **Physics Values**		
⌊ **Turbulent Length Scale**	**Value**	`0.005 m`

Node	Property	Setting
⌊ **Velocity Magnitude**	**Value**	`1.23 m/s`
⌊ **Volume Fraction**	**Value**	`0.8`
Air > **Physics Values**		
⌊ **Velocity Magnitude**	**Value**	`1.61 m/s`
⌊ **Volume Fraction**	**Value**	`0.2`

3. Save the simulation.

Setting Stopping Criteria and Solver Parameters

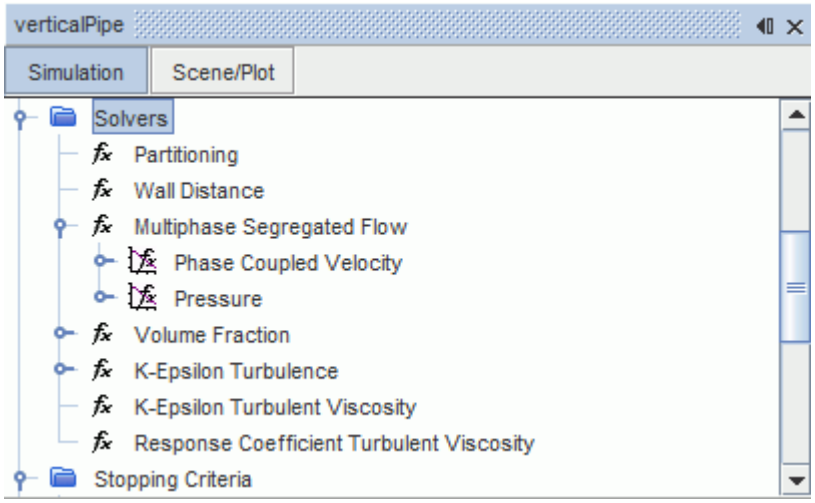
Set the appropriate solver parameters and stopping criteria for the simulation.

Decrease the under-relaxation factors for the solvers to aid solution convergence.

To set the solver parameters and stopping criteria:

1. Edit the **Solvers** node and set the following properties:

Node	Property	Setting
Multiphase Segregated Flow > **Phase Coupled Velocity**	**Implicit Under-Relaxation Factor**	`0.8`
	Explicit Under-Relaxation Factor	`0.5`
Multiphase Segregated Flow > **Pressure**	**Under-Relaxation Factor**	`0.2`
Volume Fraction	**Implicit Under-Relaxation Factor**	`0.6`
	Explicit Under-Relaxation Factor	`0.25`
K-Epsilon Turbulence	**Under-Relaxation Factor**	`0.8`
K-Epsilon Turbulent Viscosity	**Under-Relaxation Factor**	`1.0`



2. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Steps	**Enabled**	Activated
	Maximum Steps	`750`

3. Save the simulation.

Setting Up the Plots

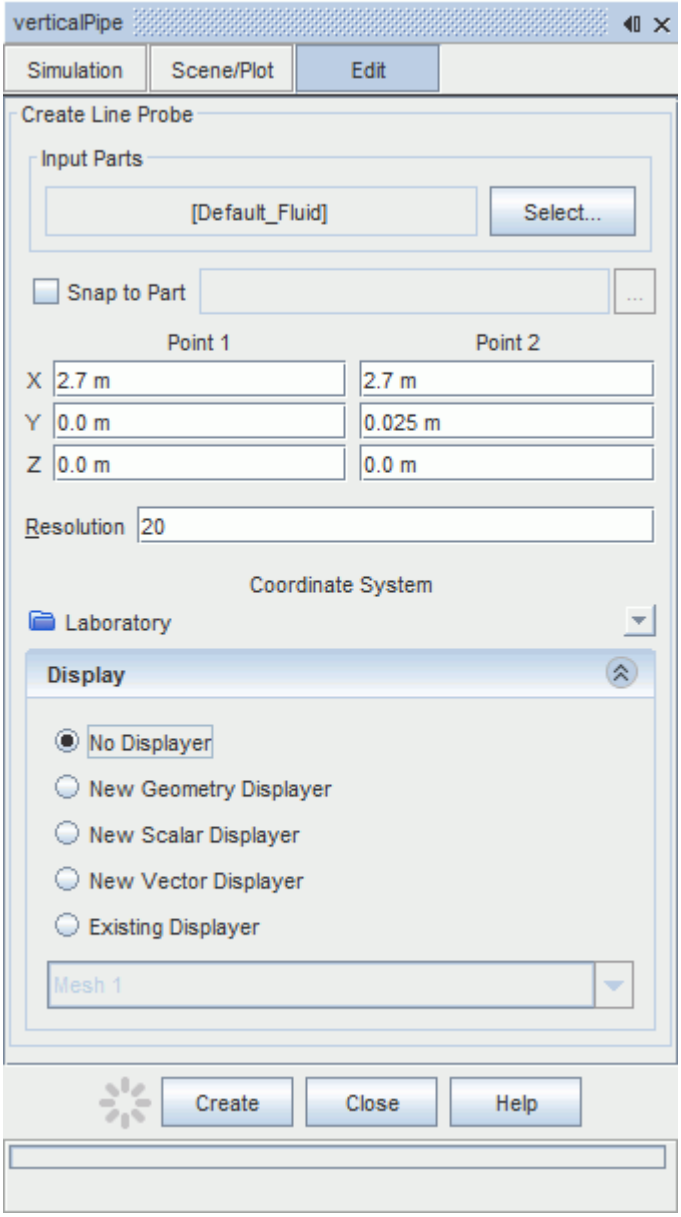
Create a plot to view the simulation results.

A line probe is created and used to extract data across the radius of the bubble column. When creating a derived part, STAR-CCM+ automatically uses the active scene to display a preview of the part.

To set up the plots:

1. Right-click the **Derived Parts** node and select **New Part** > **Probe** > **Line**.
2. In the *Create Line Probe* panel, do the following:
   a) Make sure that **Default_Fluid** is selected in the **Input Parts** box.
   b) Set the coordinates of **Point 1** to `[2.7, 0.0, 0.0] m`.
   c) Set the coordinates of **Point 2** to `[2.7, 0.025, 0.0] m`.
   d) In the **Display** box, select **No Displayer**.

   The panel appears as follows.

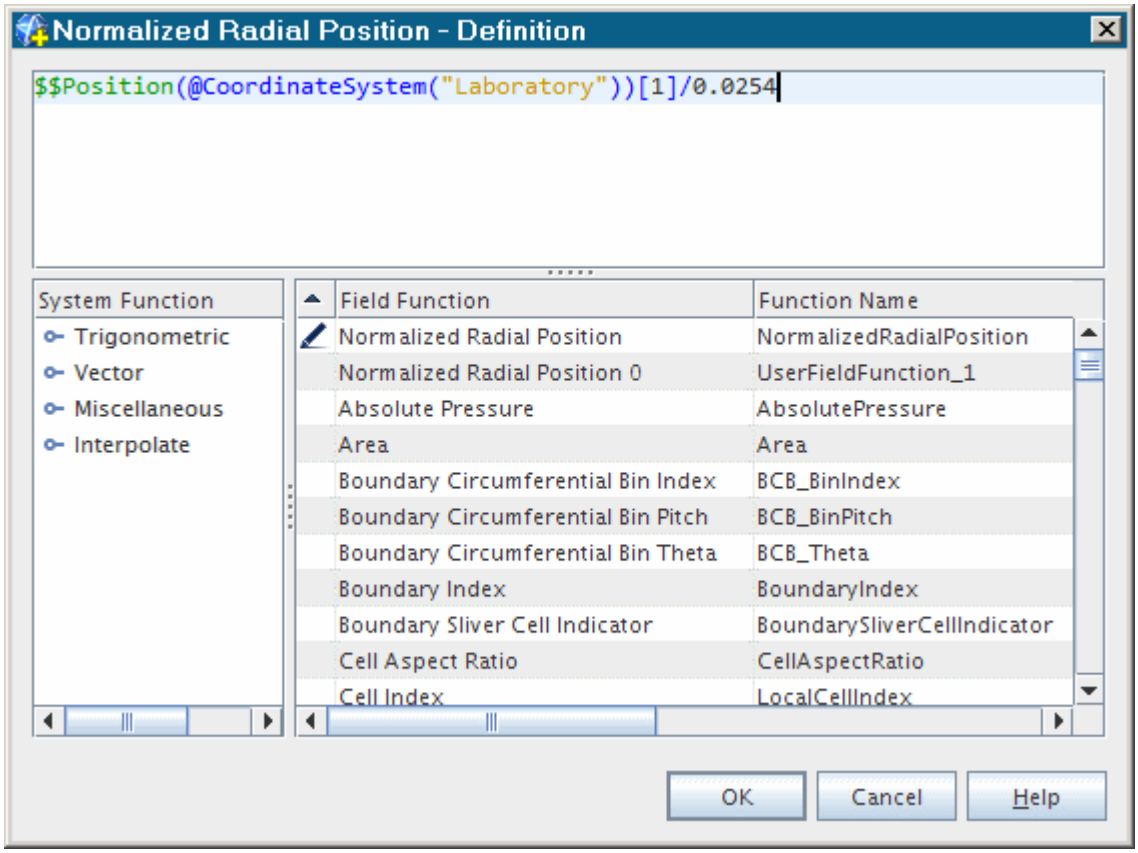


e) Click **Create** and then **Close**.

The x-axis in the plots indicates the radial position of the data points in the pipe. For convenience, use the normalized radial position (`r/R`), which can be specified using a field function:

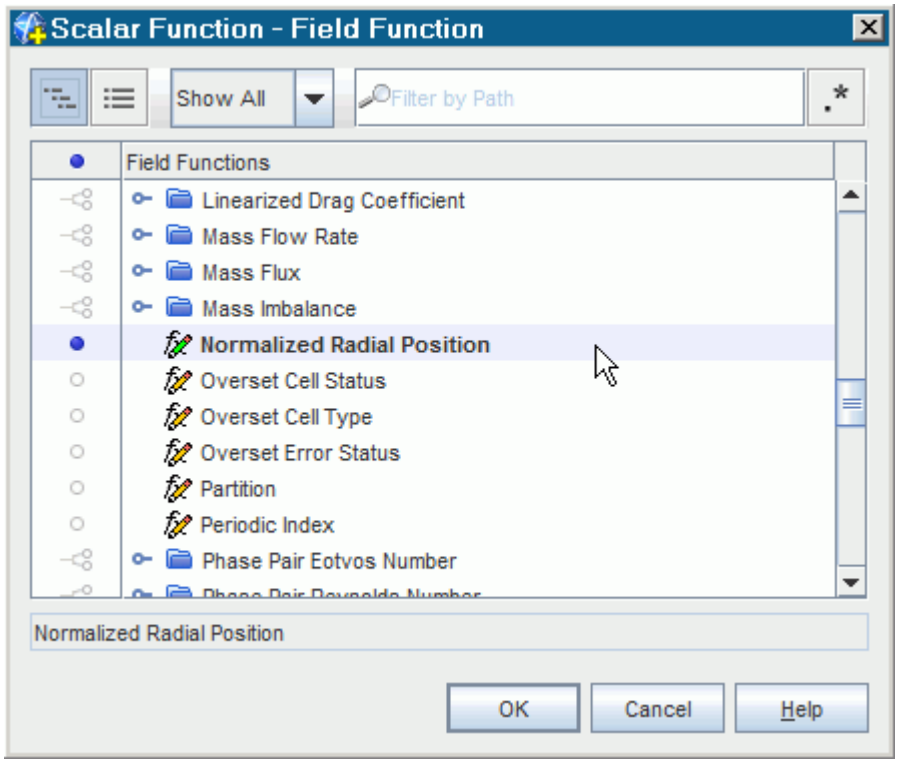
3. Right-click the **Tools** > **Field Functions** node and select **New** > **Scalar**.
4. Rename the **User Field Function 1** node to `Normalized Radial Position`.
5. Set the *Function Name* to `Normalized Radial Position`.
6. Set the *Definition* to `$$Position(@CoordinateSystem("Laboratory"))[1]/0.0254`

Note: Two dollar symbols (`$$`) are used when referencing a position field function.

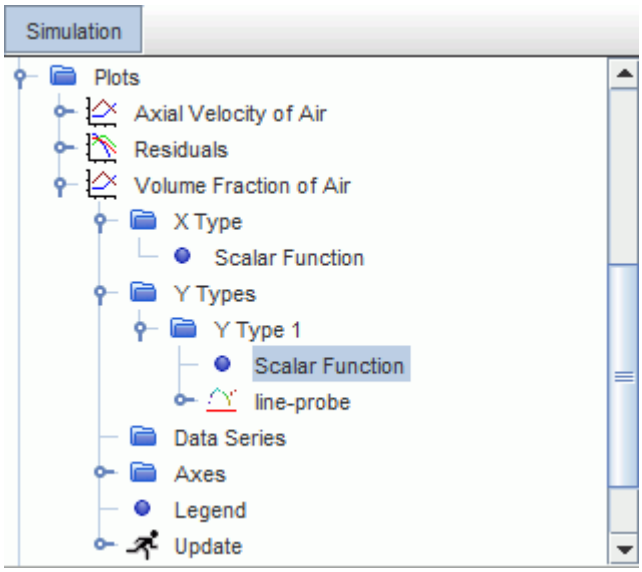


The plot can now be created.

7. Click (**Create/Open Plots**) in the **Plot** toolbar and select **XY Plot**.
8. Rename the **Plots** > **XY Plot 1** node to `Volume Fraction of Air`.
9. Click **[]** next to the *Parts* property. In the *Volume Fraction of Air - Parts* dialog, select **Derived Parts** > **Line Probe**.
10. Select the **Volume Fraction of Air** > **X Type** node and set the *Data Type* to **Scalar**.
11. Select the **X Type** > **Scalar Function** node and set the *Field Function* property to **Normalized Radial Position**.



12. Select the **Y Types** > **Y Type 1** > **Scalar Function** node.

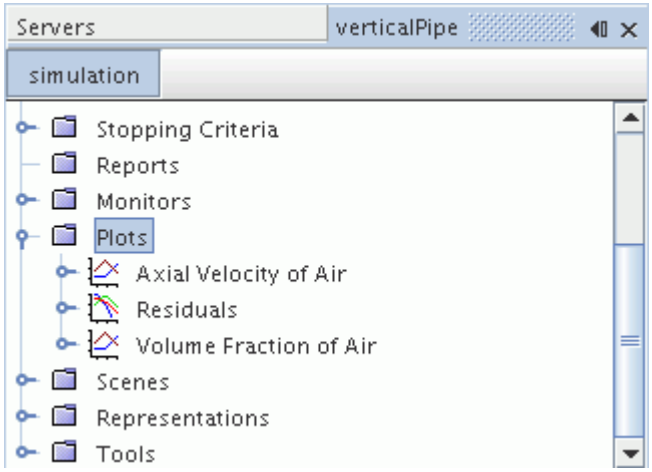


13. Set the *Field Function* property to **Volume Fraction** > **Volume Fraction of Air**.

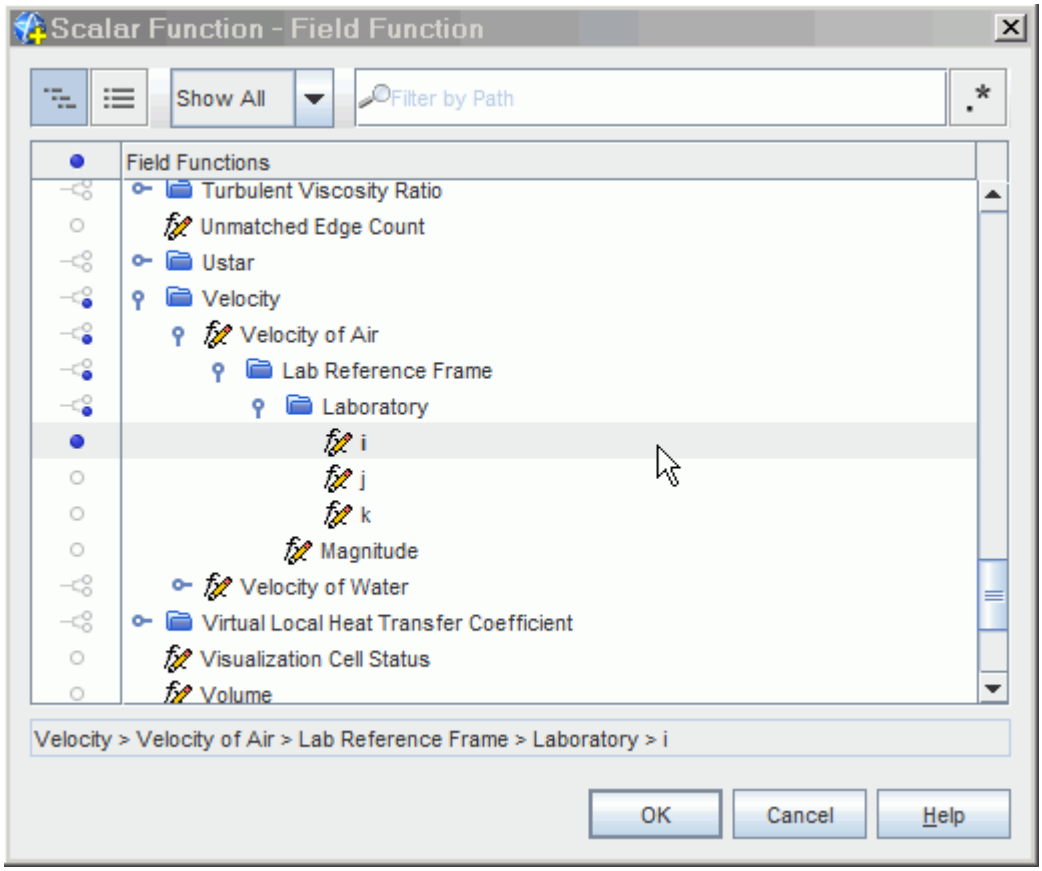
The remaining plot shares many properties with the first plot, so you can copy and paste the original plot to create the remaining plot and change some of the properties as necessary:

14. Right-click the **Plots** > **Volume Fraction of Air** node and select **Copy**, then right-click the **Plots** node and select **Paste**.
15. Rename the new plot `Axial Velocity of Air`.

16. Right-click the **Plots** node and select **Refresh**.



17. Select the **Axial Velocity of Air** > **Y Types** > **Y Type 1** > **Scalar Function** node and set the *Field Function* to **Velocity** > **Velocity of Air** > **Lab Reference Frame** > **Laboratory** > **i**.



18. To open the plots, right-click the **Plots** node and select **Open All Plots**. Use the tabs at the top of the *Graphics* window to display each plot.
19. Save the simulation.

Running the Simulation

Preparation of the simulation is now complete, and the simulation can be run.

To run the simulation:

1. Click (**Run**) in the *Solution* toolbar.

   The solution progress is displayed in the *Output* window. The *Residuals* display is created automatically in the *Graphics* window and shows the progress of the solvers.

   While the simulation is running, you can click the tabs at the top of the *Graphics* window to view scenes and plots.

   During the run, you can stop the process by clicking (**Stop**) on the toolbar. If you do stop the simulation, you can click (**Run**) to resume it.

   If left alone, the simulation continues until 750 iterations are complete. When the last iteration is complete, the following message is displayed in the *Output* window:

   `Stopping criterion Maximum Steps satisfied.`

2. When the simulation has finished running, save it.

Exporting the Plot Data

Export the data from the plots so that it can be compared with the S-Gamma model and experimental data in the second part of the tutorial.

To export the plot data:

1. Right-click the **Plots** > **Volume Fraction of Air** node and select **Export**.
2. In the *Save* dialog that appears, save the file under the name `nonSGamma_voidFraction.csv`.
3. Repeat the steps for the other plot, naming the exported data for the **Axial Velocity of Air** plot as `nonSGamma_vAir.csv`.

Using the S-Gamma Model in the Simulation

The S-Gamma model was developed to model the size distribution of dispersed bubbles with the consideration of breakup and coalescence effects.

In dispersed two-phase flows, the strength of interphase forces, and the quantity of heat and mass transfer, are functions of the bubble size. Bubble sizes can change continuously due to coalescence and breakup, therefore being able to account for this size distribution leads to an improved prediction of these quantities. In STAR-CCM+, the S-Gamma method of moments has been implemented to account for particle size distributions in dispersed two-phase flows.

- For breakup, the S-Gamma model considers the balance between disruptive forces (due to shear and turbulence) and restoring forces (due to surface tension) on the droplet.
- For coalescence, the S-Gamma model considers the probability of collisions of the droplets, the contact time of two colliding drops, and the drainage time of the liquid film between the drops.

Contents:

Activating and Setting Up the S-Gamma Model
Modifying the Phase Interaction Settings
Setting S-Gamma Boundary Values and Solver Properties
Clearing the Solution and Running the Simulation

Activating and Setting Up the S-Gamma Model

Activate the S-Gamma model and define the appropriate properties.

The S-Gamma model is activated in the *Phase Model Selection* dialog, where it can be selected from the *Optional Models* box.

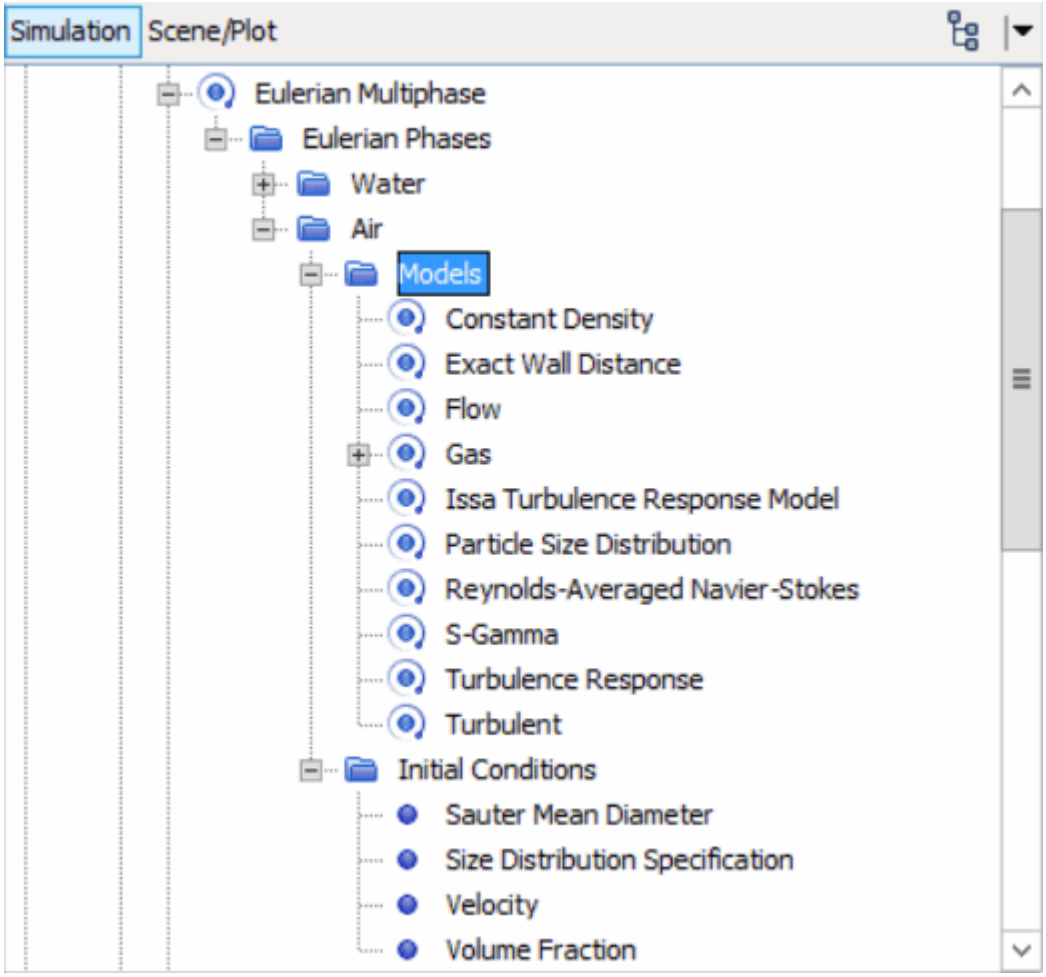
To activate the S-Gamma model:

1. Save a copy of the simulation. Name the copy `verticalPipeSGamma.sim`.
2. Right-click the **Physics 1** > **Models** > **Eulerian Multiphase** > **Eulerian Phases** > **Air** > **Models** node and choose **Select models...**.

In the *Air Model Selection* dialog:

3. Select the **Particle Size Distribution** model from the *Optional Models* group box.
4. Select the **S-Gamma** model from the *Particle Size Distribution* group box.
5. Click **Close**.

   The **Eulerian Phases** > **Air** node appears as shown below.



Define the S-Gamma properties.

6. Select the **Air** > **Models** > **S-Gamma** node and make sure that the *Number of S-Gamma Equations* property is set to **Two Equation**.

   This setting solves for both the transport of particle number density and the interfacial area density (zeroth and second moments).

7. Select the **Initial Conditions** > **Sauter Mean Diameter** node and set *Value* to `0.003 m`.

Modifying the Phase Interaction Settings

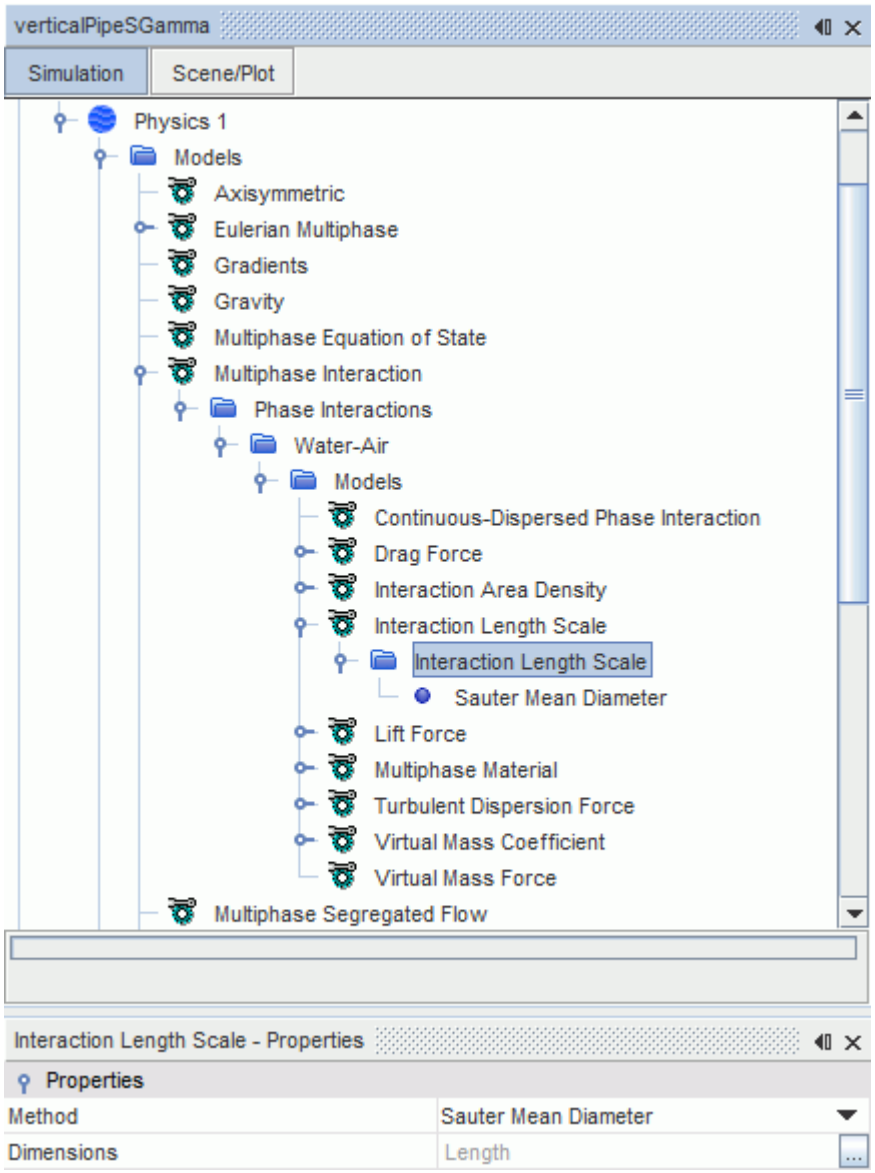
Define the appropriate phase interaction settings.

Additional models that are related to the S-Gamma model are activated in the phase interaction node. The Constant Interaction Length Scale model is also replaced with the User-Defined Length Scale model.

To modify the phase interaction settings:

1. Select the **Physics 1** > **Models** > **Multiphase Interaction** > **Phase Interactions** > **Water-Air** > **Models** node.
2. Select the **Interaction Length Scale** > **Interaction Length Scale** node, and then set **Method** to **Sauter Mean Diameter**.

The **Phase Interactions** > **Water-Air** > **Models** node appears as shown below.



3. Right-click the **Phase Interactions** > **Water-Air** > **Models** node and click **Select models**.
4. In the *Water-Air Model Selection* dialog, select the **S-Gamma Breakup** and **S-Gamma Coalescence** models from the *Optional Models* group box.
5. Click **Close**.

Define the phase interaction parameters that are related to the S-Gamma model:

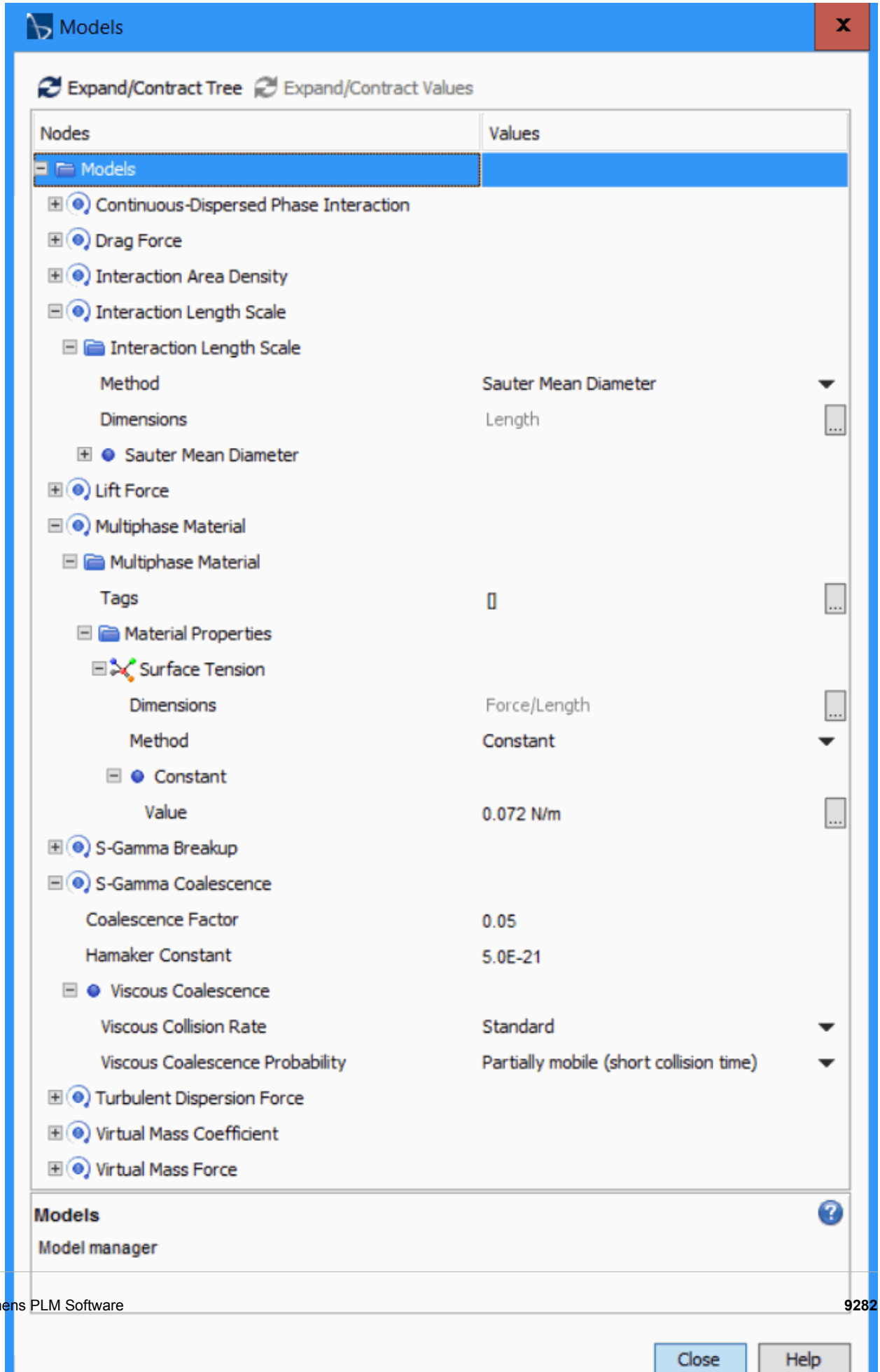
6. Right-click the **Water-Air** > **Models** node and select **Edit**.

In the *Models* dialog, do the following:

7. Expand the **Multiphase Material** > **Multiphase Material** > **Material Properties** > **Surface Tension** > **Constant** node and set the **Value** to `0.072 N/m`.
8. Expand the **S-Gamma Coalescence** node and make sure that the **Coalescence Factor** is set to `0.05`.

9. Expand the **S-Gamma Coalescence** > **Viscous Coalescence** node and set the **Viscous Coalescence Probability** to **Partially mobile (short collision time)**.

   The completed *Models* dialog is shown below:


Models
Expand/Contract Tree
Expand/Contract Values
Nodes
Values
Models
Continuous-Dispersed Phase Interaction
Drag Force
Interaction Area Density
Interaction Length Scale
Interaction Length Scale
Method
Sauter Mean Diameter
Dimensions
Length
Sauter Mean Diameter
Lift Force
Multiphase Material
Multiphase Material
Tags
[]
Material Properties
Surface Tension
Dimensions
Force/Length
Method
Constant
Constant
Value
0.072 N/m
S-Gamma Breakup
S-Gamma Coalescence
Coalescence Factor
0.05
Hamaker Constant
5.0E-21
Viscous Coalescence
Viscous Collision Rate
Standard
Viscous Coalescence Probability
Partially mobile (short collision time)
Turbulent Dispersion Force
Virtual Mass Coefficient
Virtual Mass Force
Models
Model manager
Close
Help

10. Click **Close**.

Setting S-Gamma Boundary Values and Solver Properties

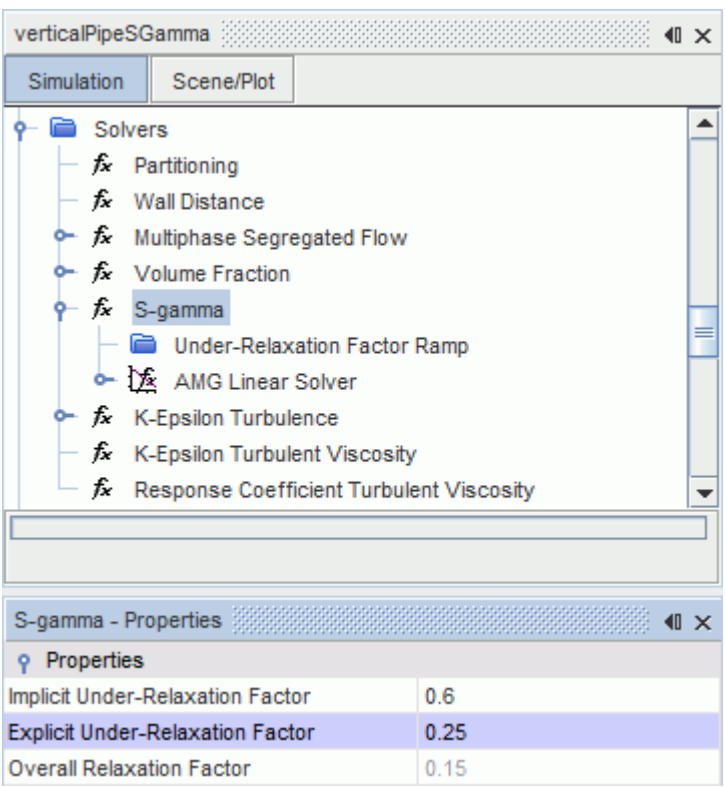
Define the appropriate boundary conditions and solver properties for the S-Gamma model.

Define the Sauter mean diameter for the inlet boundary.

To set the S-Gamma bounday values:

1. Select the **Regions** > **Default Fluid** > **Boundaries** > **Inlet** > **Phase Conditions** > **Air** > **Physics Values** > **Sauter Mean Diameter** node and set *Value* to `0.003 m`.
2. Edit the **Solvers** node and set the following properties:

Node	Property	Setting
S-Gamma	**Implicit Under-Relaxation Factor**	`0.6`
	Explicit Under-Relaxation Factor	`0.25`



3. Save the simulation.

Clearing the Solution and Running the Simulation

Clear the results of the simulation and run the simulation again.

The stopping criteria and solution controls remain unchanged.

To clear the solution and run the simulation again:

1. Clear the solution from the previous run.
   Accept all default settings in the dialog that appears.
2. Click (**Run**) in the **Solution** toolbar.
3. Save the simulation when the solver finishes.

Validating the Results

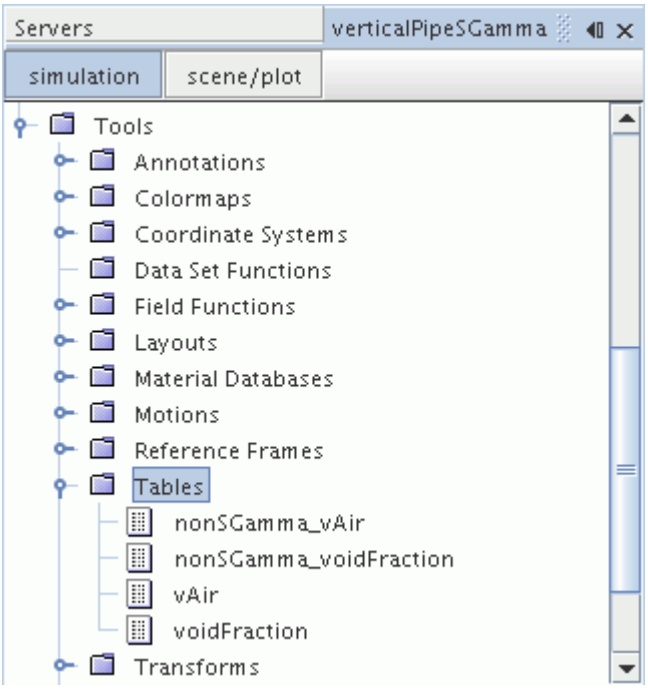
Compare the results against experimental data.

The S-Gamma and non-S-Gamma solutions are validated against experimental data. The experimental data has been provided in two comma-separated values (.csv) files. You import this data in the simulation as tables.

To validate the results:

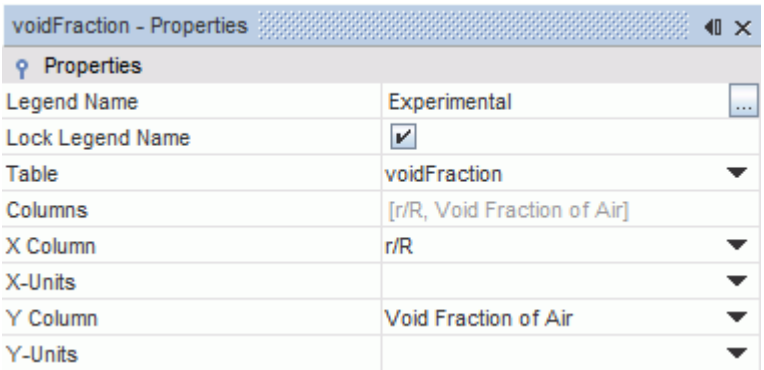
1. Right-click the **Tools** > **Tables** node and select **New Table** > **File Table**.
2. In the *Open* dialog, navigate to the `doc/startutorialsdata/multiphaseFlow/data` subdirectory of your STAR-CCM+ installation directory and multi-select `vAir.csv` and `voidFraction.csv`.
3. Click **Open**.
4. Using the same technique, import the .csv files that you exported in the first part of this tutorial:
   - `nonSGamma_vAir.csv`
   - `nonSGamma_voidFraction.csv`.

   The updated **Tables** node is shown below.



The tabular data is now added to the plots, starting with the **Volume Fraction of Air** plot:

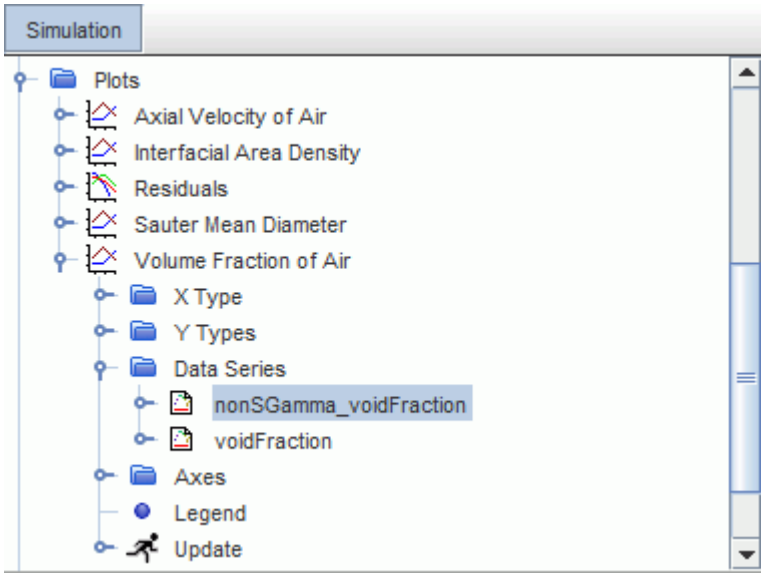
5. Select the **Plots** > **Volume Fraction of Air** node.
6. Right-click the **Data Series** node and select **Add Data**.
7. In the *Add Data Providers to Plot* window, select **voidFraction**.
8. Select the **Data Series** > **voidFraction** node and do the following:
   a) Set the *Legend Name* to `Experimental`.
   b) Set the *Table* to **voidFraction** using the drop-down menu.
   c) Set the *X Column* to **r/R**.
   d) Set the *Y Column* to **Void Fraction of Air**.



9. Select the **voidFraction** > **Symbol Style** node and do the following:
   a) Set the *Color* to **Black**.
   b) Set the *Shape* to **Filled Circle**.

Add the non-S-Gamma data set to the plot.

10. Add a tabular data set as described previously and select the **NonSGamma_voidFraction** data provider.



11. Select the **Data Series** > **NonSGamma_voidFraction** node.

Do the following:

12. Set the *Legend Name* to `NonSGamma`.
13. Set the *Table* to **nonSGamma_voidFraction** using the drop-down menu.
14. Set the *X Column* to **Line Probe: Normalized Radial Position**.
15. Set the *Y Column* to **Line Probe: Volume Fraction of Air**.
16. Select the **NonSGamma_voidFraction** > **Line Style** node and do the following:
    a) Set the *Color* to a color of your choice. In this tutorial, **Blue** is used for the non-S-Gamma data set.
    b) Set the *Style* to **Solid**.
    c) Select the **NonSGamma_voidFraction** > **Symbol Style** node and set the *Color* to **Blue**.

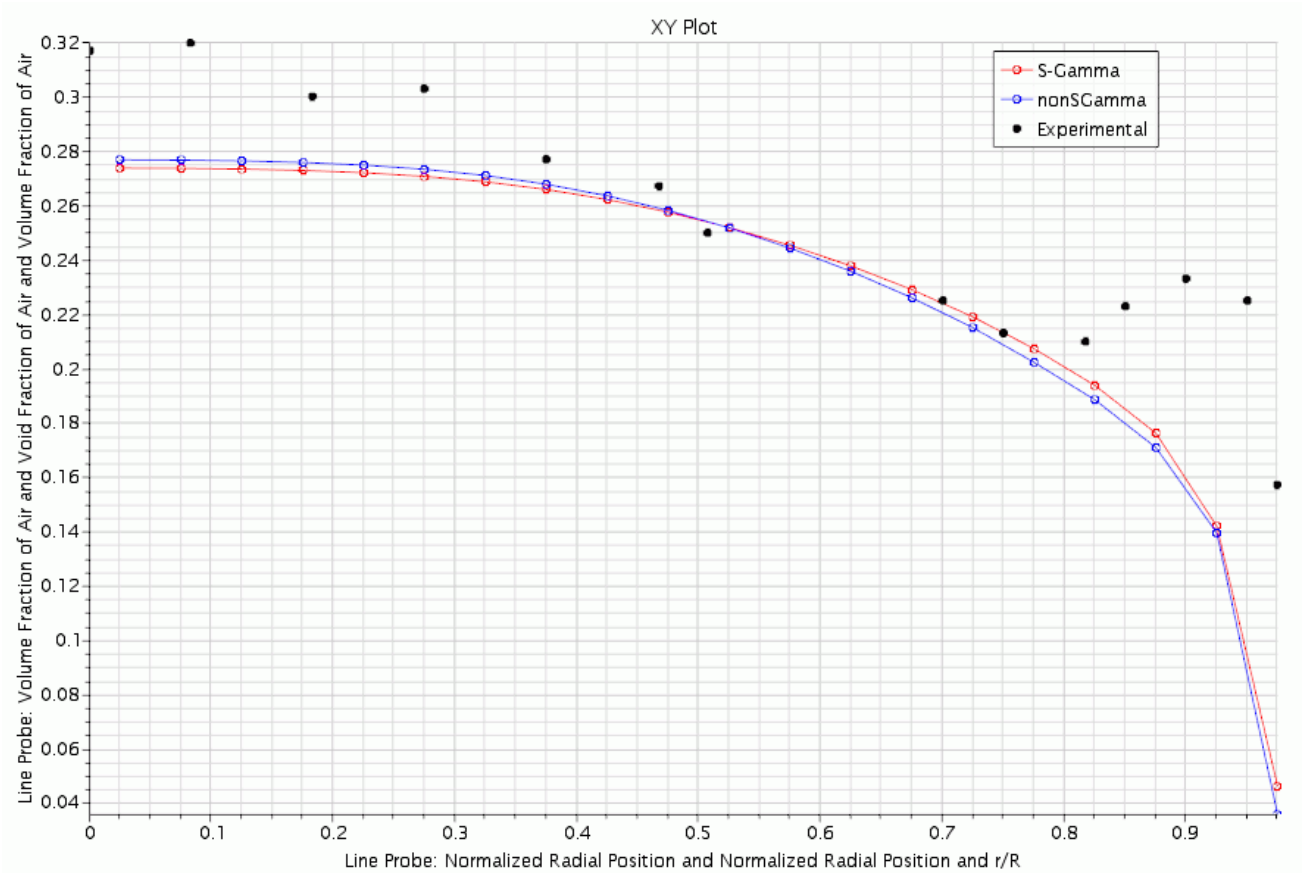
Rename the original data set for this simulation.

17. Select the **Volume Fraction of Air** > **Y Types** > **Y Type 1** > **Line Probe** node and set the *Legend Name* to `S-Gamma`.

Set the line type for the **S-Gamma** data set to **Solid**:

18. Select the **Line Probe** > **Line Style** node and set the *Style* to **Solid**.

The *Volume Fraction of Air* plot appears as shown below.



Add the remaining data sets to the plot using the techniques described above.

19. Add the two data sets to the **Data Series** node in the **Axial Velocity of Air** plot. Rename one of the data sets to `vAir` and the other to `NonSGamma_vAir`.
20. Right-click the **vAir** data set node and select **Edit**.
21. In the *Edit* dialog, set the properties as shown in the table below.

Node	Value
Table	vAir
X Column	r/R
Y Column	Axial Velocity of Air
Symbol Style > **Color**	Black
Symbol Style > **Shape**	Filled Circle

Do the same for the NonSGamma data set.

22. Right-click the **NonSGamma_vAir** data set node and select **Edit**.
23. In the *Edit* dialog, set the properties as shown in the table below.

Node	Value
Table	**nonSGamma_vAir**

Node	Value
X Column	**Line Probe: Normalized Radial Position**
Y Column	**Line Probe: Velocity of Air[i]**
Line Style > **Color**	Blue
Line Style > **Style**	Solid
Symbol Style > **Color**	Blue

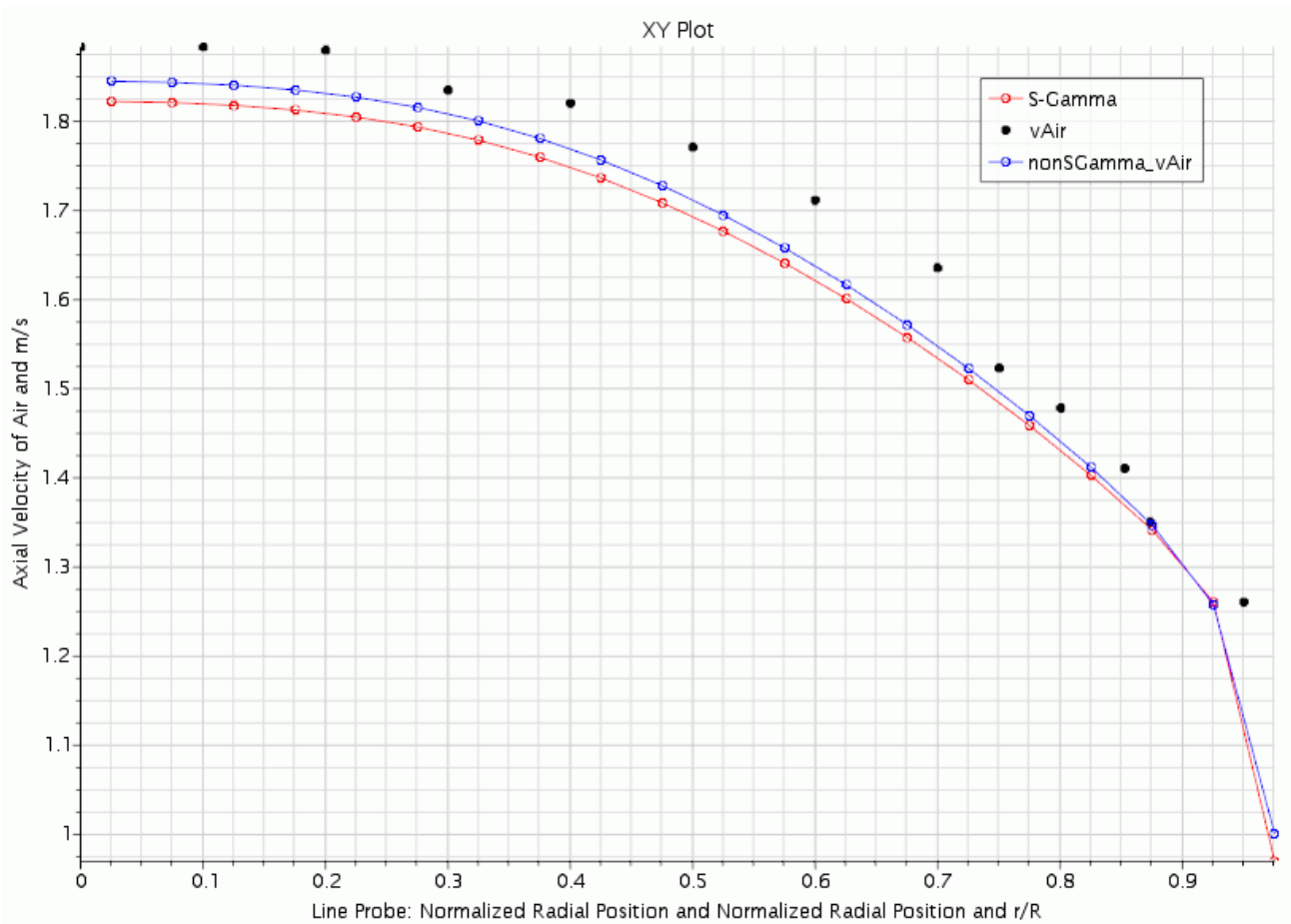
Rename the original data set for this simulation:

24. Select the **Axial Velocity of Air** > **Y Types** > **Y Type 1** > **Line Probe** node and set the *Legend Name* to `S-Gamma`.

Set the line type for the **S-Gamma** data set to **Solid**:

25. Select the **Line Probe** > **Line Style** node and set the *Style* to **Solid**.

The *Axial Velocity of Air* plot is shown below.



26. Save the simulation.

Contents:

Validating the S-Gamma Model

Validating the S-Gamma Model

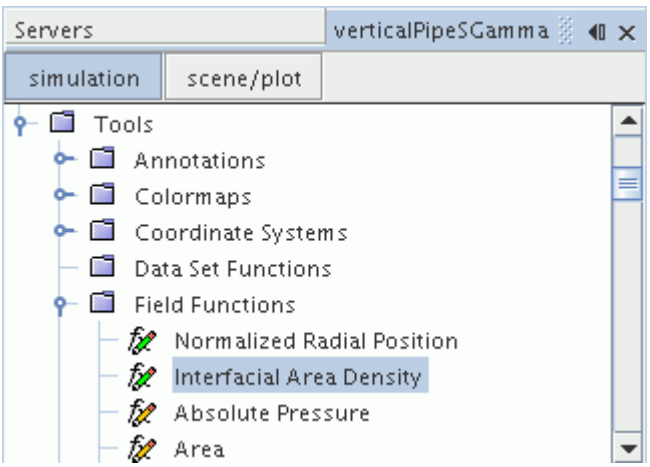
Compare the S-Gamma model results with experimental data.

The following section compares the plots of the Sauter mean diameter of the dispersed phase and the Interfacial Area Density to experimental data.

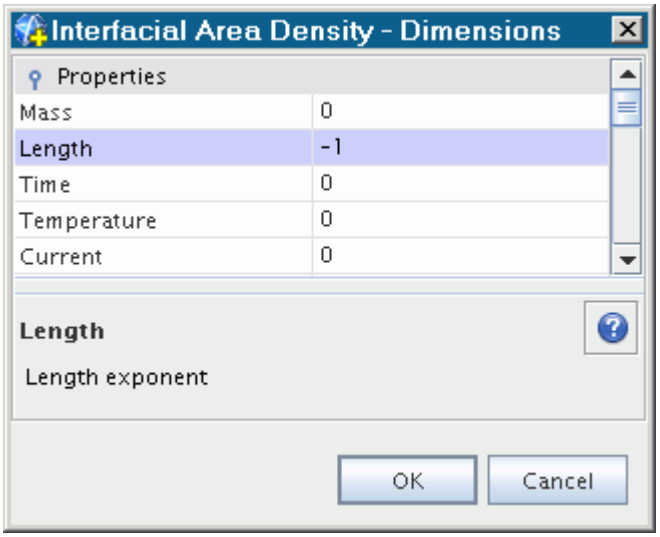
Create a field function that calculates the Interfacial Area Density. The Interfacial Area Density is defined by multiplying the **Second Size Distribution of Air** field function by π.

To validate the S-Gamma model:

1. Under the **Tools** > **Field Functions** node, create a scalar field function and rename it to `Interfacial Area Density`.



2. Select the **Interfacial Area Density** node and do the following:
   a) Click the area next to the *Dimensions* property. In the dialog that appears, set the *Length* to `-1`.



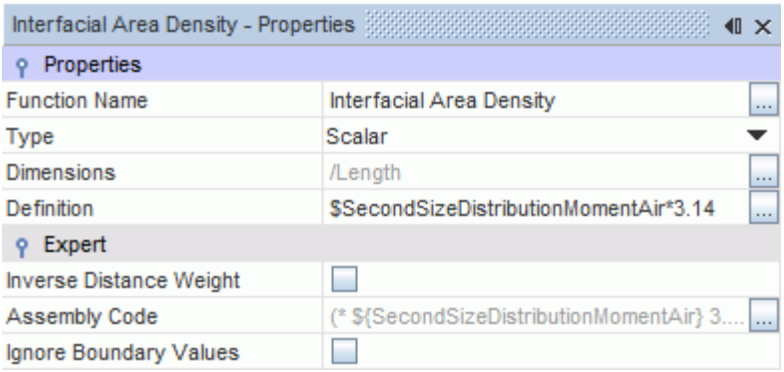
b) Click **OK**.

c) Set the *Function Name* to `Interfacial Area Density`.

d) Set the *Definition* to `$SecondSizeDistributionMomentAir*3.14`

Note: One dollar symbol (`$`) is used when referencing a scalar field function.

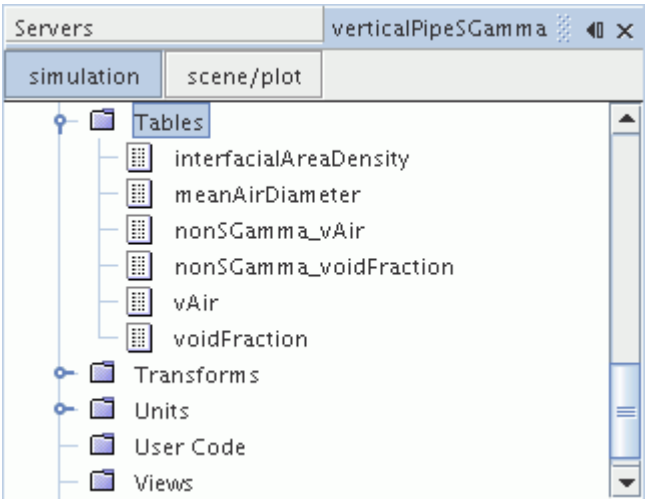
The completed *Properties* window is shown below.



Import the experimental data into STAR-CCM+.

3. Import the `interfacialAreaDensity.csv` and `meanAirDiameter.csv` files under the **Tools** > **Tables** node.

   These files can be found in the `doc/startutorialsdata/multiphaseFlow/data` subdirectory of your STAR-CCM+ installation directory.



The plots can now be created. The plot of the Interfacial Area Density is created first.

4. Create a **XY Plot** and rename it to `Interfacial Area Density`.

A new plot automatically appears in the *Graphics* window.

5. Right-click the **Interfacial Area Density** node and select **Edit**.
6. In the *Interfacial Area Density* dialog, set the properties as shown in the table below.

Node	Value
Parts	**Derived Parts** > **Line Probe**
X-Type > **Data Type**	**Scalar**
X-Type > **Scalar Function** > **Field Function**	**Normalized Radial Position**
Y-Types > **Y-Type 1** > **Scalar Function** > **Field Function**	**Interfacial Area Density**
Y-Types > **Y-Type 1** > **Line Probe** > **Legend Name**	**S-Gamma**
Y-Types > **Y-Type 1** > **Line Probe** > **Line Style** > **Color**	**Red**
Y-Types > **Y-Type 1** > **Line Probe** > **Line Style** > **Style**	**Solid**
Y-Types > **Y-Type 1** > **Line Probe** > **Symbol Style** > **Color**	**Red**

7. Click **Close**.
8. Right-click the **Interfacial Area Density** > **Data Series** node and click **Add Data**.
9. In the *Add Data Providers to Plot* window, select **interfacialAreaDensity**.
10. Right-click the **Data Series** > **interfacialAreaDensity** node and select **Edit**.
11. In the *interfacialAreaDensity* dialog, set the properties as shown in the table below.

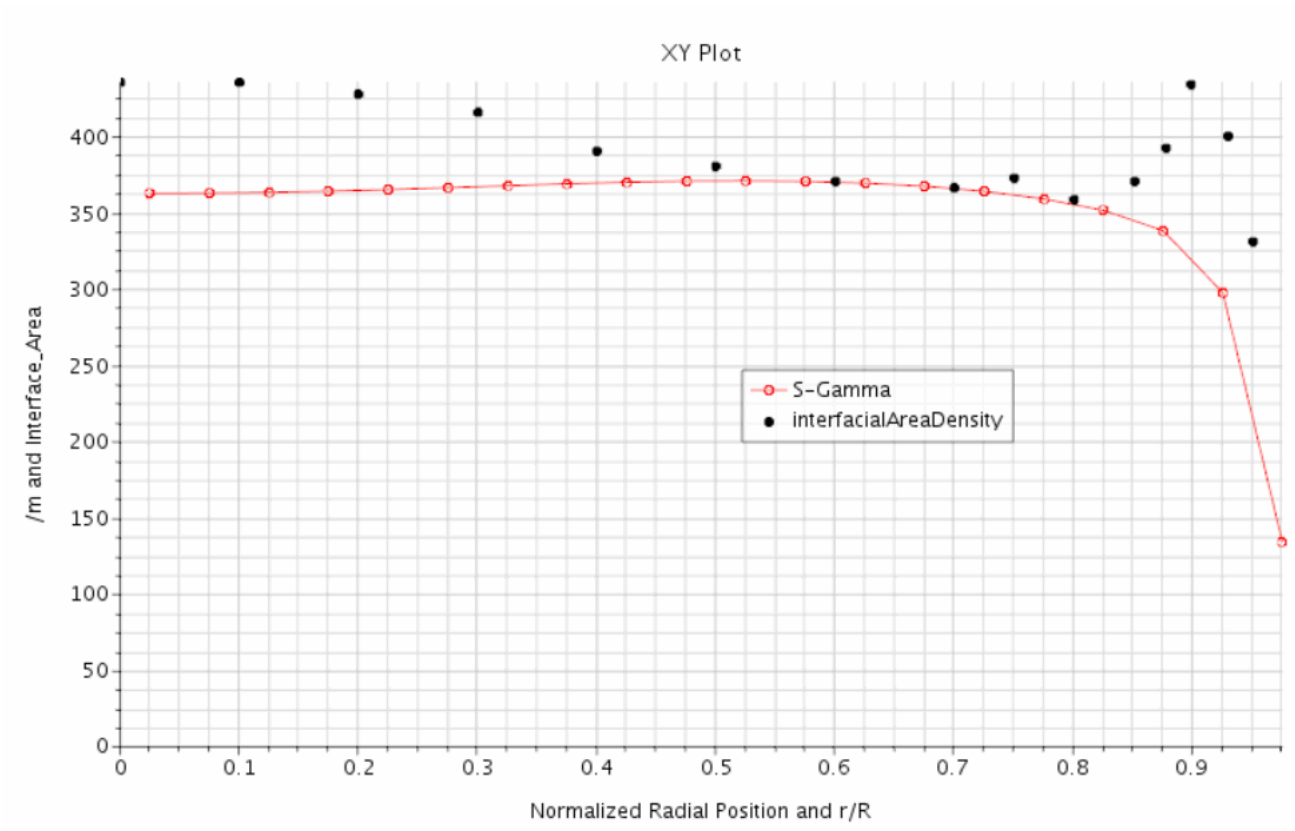
Node	Value
Table	**interfacialAreaDensity**
X Column	**r/R**
Y Column	**Interface_Area**
Symbol Style > **Color**	**Black**
Symbol Style > **Shape**	**Filled Circle**

12. Click **Close**.

Manually adjust the y-axis minimum value.

13. Select the **Interfacial Area Density** > **Axes** > **Left Axis** node, set the *Minimum* to `0.0`, and activate the checkbox next to *Lock Minimum*.

The *Interfacial Area Density* plot now has a minimum y value of 0, as shown in the following image:



Create and set up the plot of the Sauter Mean Diameter:

14. Create an **XY Plot** and rename it `Sauter Mean Diameter`.

A new plot automatically appears in the *Graphics* window.

15. Right-click the **Sauter Mean Diameter** node and select **Edit**.
16. In the *Sauter Mean Diameter* dialog, set the properties as shown in the table below.

Node	Value
Parts	**Derived Parts** > **Line Probe**
X-Type > **Data Type**	**Scalar**
X-Type > **Scalar Function** > **Field Function**	**Normalized Radial Position**
Y-Types > **Y-Type 1** > **Scalar Function** > **Field Function**	**Sauter Mean Diameter** > **SauterMeanDiameter of Air**
Y-Types > **Y-Type 1** > **Scalar Function** > **Units**	**mm**
Y-Types > **Y-Type 1** > **Line Probe** > **Legend Name**	**S-Gamma**
Y-Types > **Y-Type 1** > **Line Probe** > **Line Style** > **Color**	**Red**
Y-Types > **Y-Type 1** > **Line Probe** > **Line Style** > **Style**	**Solid**
Y-Types > **Y-Type 1** > **Line Probe** > **Symbol Style** > **Color**	**Red**

17. Click **Close**.
18. Right-click the **Sauter Mean Diameter** > **Data Series** node and click **Add Data**.
19. In the *Add Data Providers to Plot* window, select **meanAirDiameter**.
20. Right-click the **Data Series** > **meanAirDiameter** node and select **Edit**.
21. In the *meanAirDiameter* dialog, set the properties as shown in the table below.

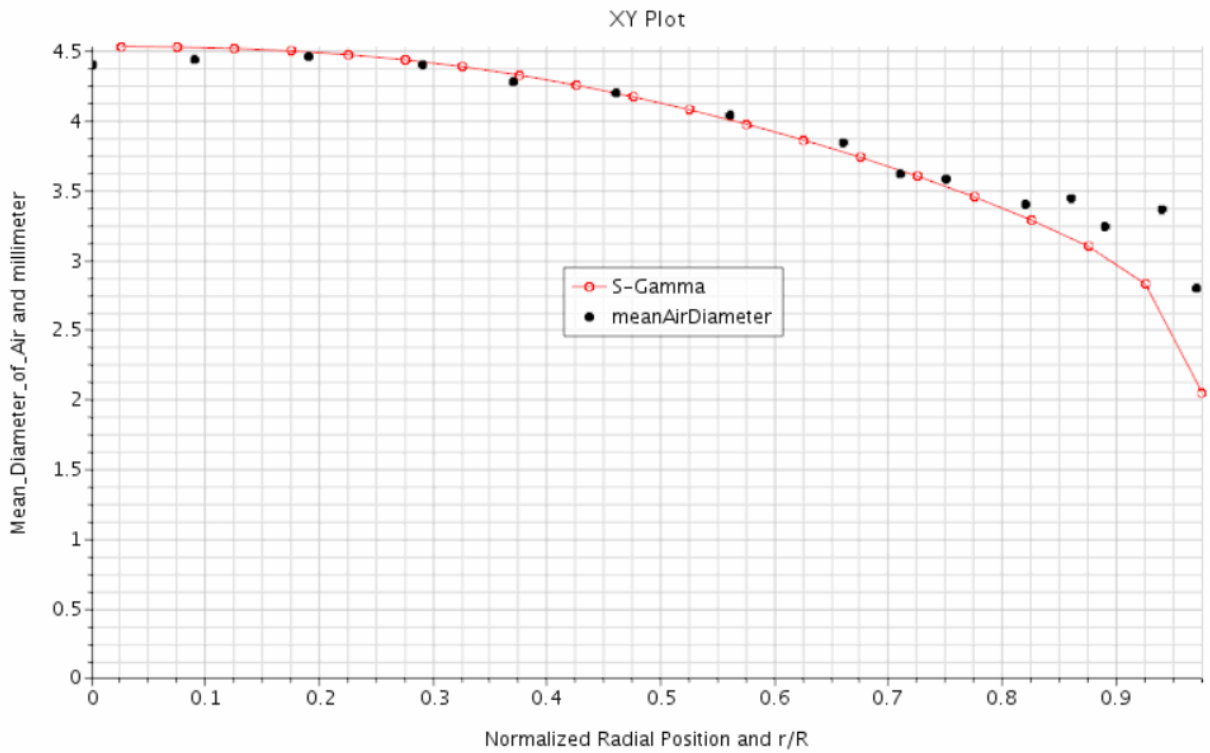
Node	Value
Table	**meanAirDiameter**
X Column	**r/R**
Y Column	**Mean_Diameter_of_Air**
Symbol Style > **Color**	**Black**
Symbol Style > **Shape**	**Filled Circle**

22. Click **Close**.

Manually adjust the y-axis minimum value.

23. Select the **Sauter Mean Diameter** > **Axes** > **Left Axis** node, set the *Minimum* to `0.0`, and activate the checkbox next to *Lock Minimum*.

    The *Sauter Mean Diameter* plot has a minimum y value of 0, as shown below:



Summary

This tutorial demonstrated how to model a Eulerian multiphase turbulent flow in STAR-CCM+.

This tutorial has introduced the following STAR-CCM+ features:

- Setting up the Eulerian Multiphase and Multiphase Turbulence models
- Creating phases and selecting phase models
- Defining phase interactions
- Exporting results
- Activating the S-Gamma model
- Validating the results.

Bibliography

This tutorial references the following paper:

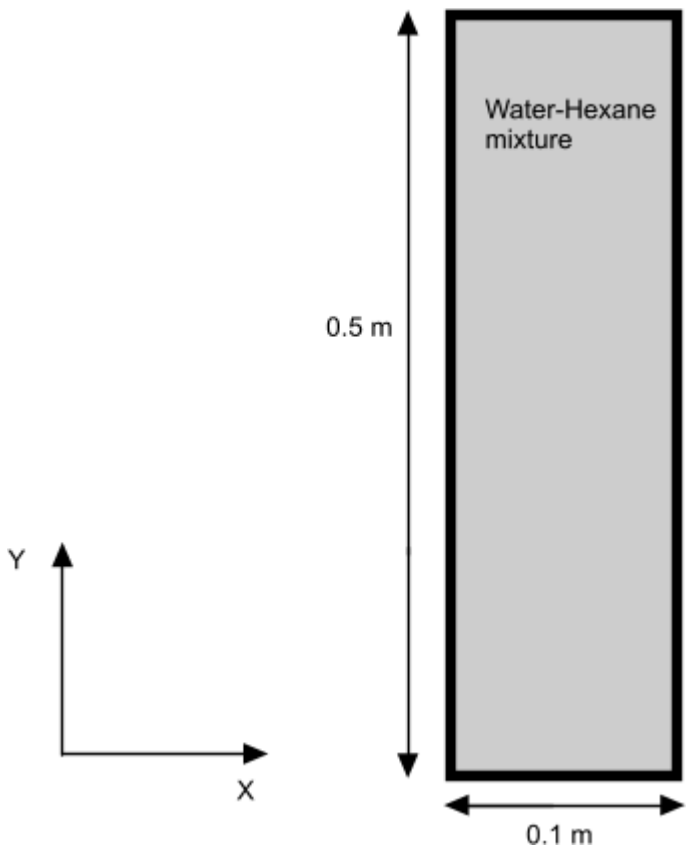
[797] *Hibiki, T., Ishii, M., Xiao, Z., 2001. "Axial interfacial area transport of vertical bubbly flows." International Journal of Heat and Mass Transfer, Vol. 44.*

Eulerian: Mixture Settling

This tutorial demonstrates how to model mixture settling in STAR-CCM+. The simulation models the Rayleigh-Taylor instability where the heavier water phase is above the lighter hexane phase in a rectangular tank.

This tutorial uses the Multiple Flow Regime model, which combines the benefits of the VOF model and the Eulerian Multiphase model in a single model. The Multiple Flow Regime model captures sharp free surfaces in a similar manner to the VOF model. However, when droplets or bubbles are smaller than the mesh can resolve, the Multiple Flow Regime model treats them as dispersed phases. This approach is analogous to the Large Eddy Simulation (LES) model where scales larger than the grid are resolved, and subgrid scales are modeled.

A two-dimensional mesh is used to represent the tank. The geometry that used in this simulation is shown below:



The boundaries representing the tank are assumed to be no-slip walls. In this simulation, the implicit unsteady solver is used to study the movement of two phases with different densities within the tank. The buoyancy effect causes the lighter hexane phase to rise and settle at the top of the heavier water phase.

Contents:

Prerequisites

Importing the Mesh and Naming the Simulation
Converting to a Two-Dimensional Mesh
Selecting the Physics Models
Creating Phases and Selecting Phase Models
Defining the Phase Interactions
Setting Initial Conditions and Reference Values
Setting Solver Parameters and Stopping Criteria
Setting up a Monitor
Creating a Scalar Scene
Running the Simulation
Visualizing the Results
Summary

Prerequisites

The instructions in the Eulerian: Mixture Settling tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Importing the Mesh and Naming the Simulation

Launch STAR-CCM+, import the supplied geometry, and save the simulation.

A three-dimensional, trimmed cell mesh has been prepared for this analysis, which is subsequently converted to a two-dimensional mesh.

To import the mesh and name the simulation:

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh**.
4. In the *Open* dialog, choose `doc/startutorialsdata/multiphaseFlow/data/tank.ccm`, then click **Open**.

   STAR-CCM+ provides feedback on the import process, which takes a few seconds, in the *Output* window.
5. Save the simulation as `settlingTank.sim`.

Converting to a Two-Dimensional Mesh

Convert the three-dimensional mesh to two dimensions.

To carry out a two-dimensional analysis, convert the imported mesh to two-dimensions using the **Convert to 2D** option. STAR-CCM+ has special requirements for converting three-dimensional meshes to two dimensions:

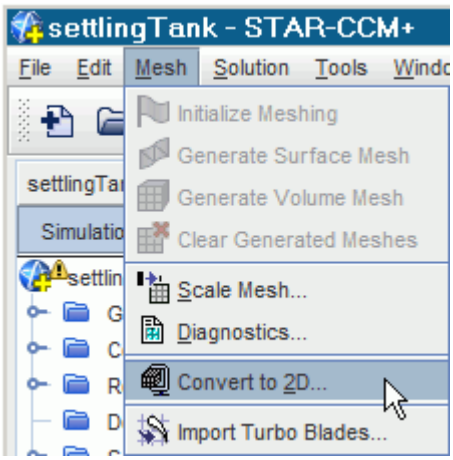
- The grid must be aligned with the X-Y plane.
- The grid must have a boundary plane at the Z = 0 location.

The imported mesh for this tutorial was built to meet these requirements.

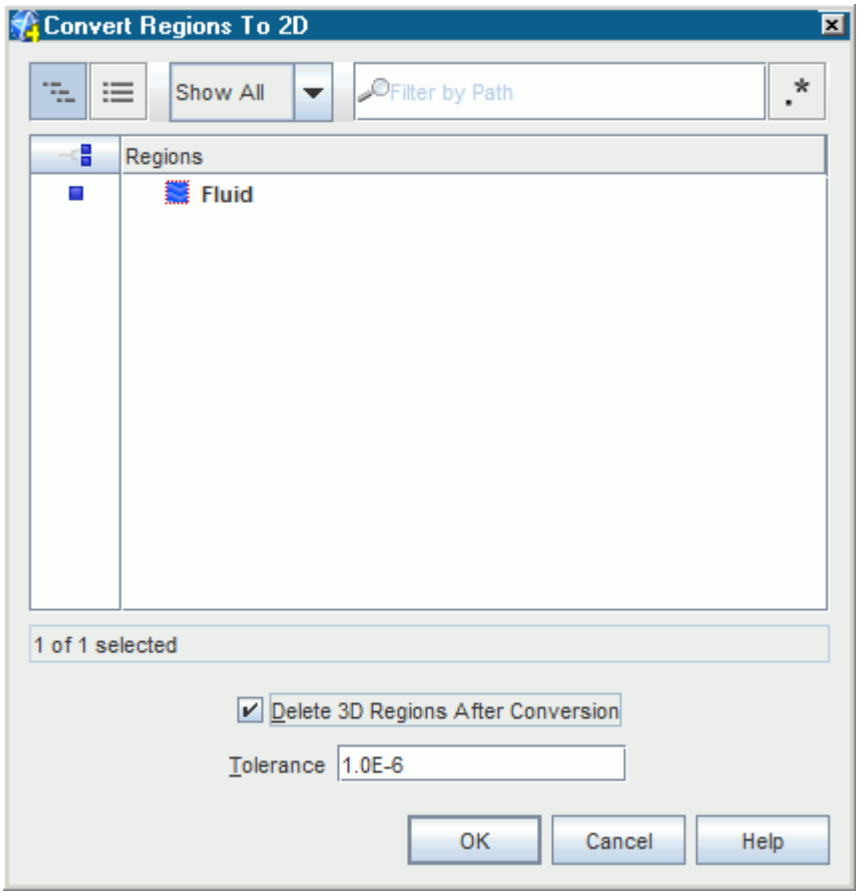
1. Rename the **Regions** > **Default_Fluid** node to **Fluid**.

To convert the mesh to two dimensions:

2. Select **Mesh** > **Convert to 2D**.



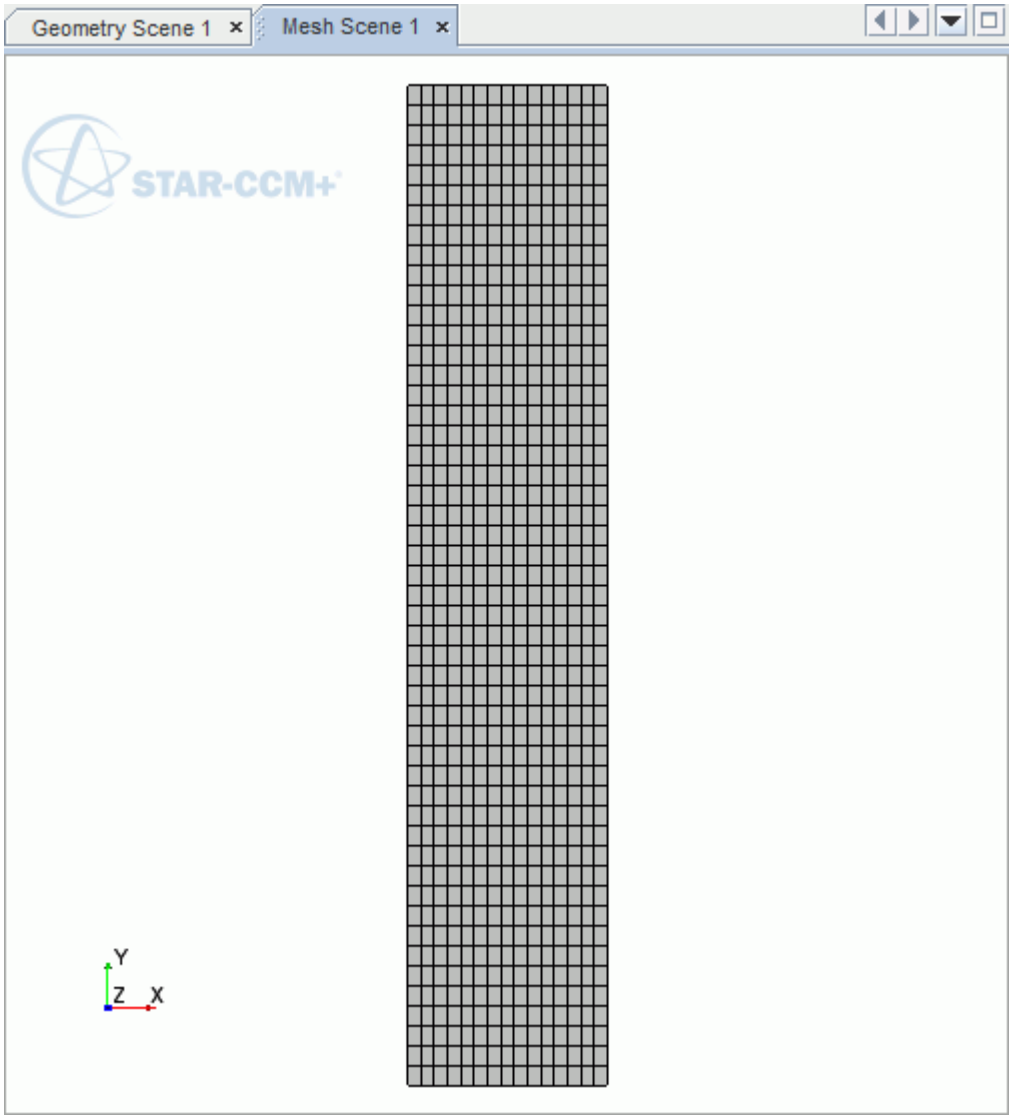
3. In the *Convert Regions To 2D* dialog, do the following:
   a) Check that the **Fluid** region is selected.
   b) Activate the **Delete 3D Regions After Conversion** option.



c) Click **OK**.

4. Right-click the **Scenes** node and then select **New Scene** > **Mesh**.

   The converted two-dimensional mesh is shown, viewed from the z-direction, in the *Mesh Scene 1* display. The mouse rotation option is suppressed for two-dimensional scenes.



You can perform the mesh diagnostics check by selecting **Mesh** > **Diagnostics** from the menu bar. The extent of the geometry is:

```
x: [-5.0000e-02, 5.0000e-02] m
y: [ 0.0000e+00, 5.0000e-01] m
z: [ 0.0000e+00, 0.0000e+00] m
```

The physics continuum that was associated with the deleted three-dimensional region is no longer required and can be deleted.

5. Expand the **Continua** node. Right-click the **Physics 1** continuum node and select **Delete**.
6. Click **Yes** in the *Confirm Object Deletion* dialog.

Selecting the Physics Models

Physics models define the primary variables of the simulation, including pressure, temperature, velocity, and what mathematical formulation is used to generate the solution.

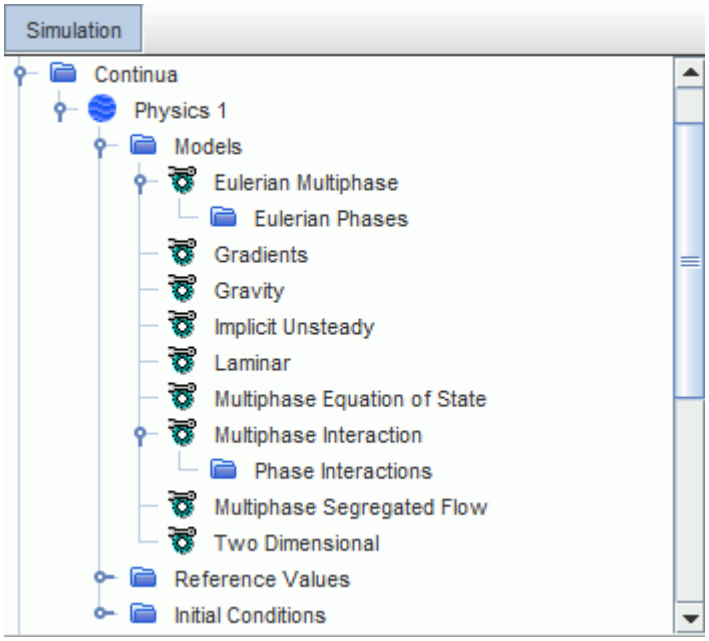
In this simulation, a simple laminar flow is set up to study the separation of two phases with different densities within a settling tank. Use the Eulerian Multiphase option to allow you to define the two phases, and the segregated multiphase option to define how the two phases interact.

To select the physics models:

1. Rename the **Continua** > **Physics 1 2D** node to **Physics 1**.
2. For the physics continuum, **Continua** > **Physics 1**, select the following models in order:

Group Box	Model
Space	**Two Dimensional** (selected automatically)
Time	**Implicit Unsteady**
Material	**Eulerian Multiphase** **Multiphase Interaction** (selected automatically)
Eulerian Multiphase Model	**Multiphase Segregated Flow** **Multiphase Equation of State** (selected automatically) **Gradients** (selected automatically)
Viscous Regime	**Laminar**
Optional Models	**Gravity**

3. Click **Close**.
4. To review the models, open the **Physics 1** > **Models** node.



In Large Scale Interface simulations, you are advised to use second order convection for the Volume Fraction as well as the Flow. This setting reduces numerical diffusion and helps to obtain a reasonably sharp interface.

5. Select the **Models** > **Multiphase Segregated Flow** node and set **Volume Fraction Convection** to **2nd-order**.
6. Save the simulation.

Creating Phases and Selecting Phase Models

Create the two phases and select the appropriate phase models.

Create two new phases and specify them as hexane and water. Select the models that are associated with each phase. The hexane and water phases are both modeled as liquid and are both defined as having a constant density. Once the phase models have been selected, you can set the material properties for each phase.

To create the phases and select the phase models:

Create the heavier liquid phase.

1. In the **Physics 1** continuum, right-click the **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and select **New**.
2. Rename the **Phase 1** node to **Water**.
3. For the **Water** phase, select the following models in order:

Group Box	Model
Enabled Models	**Laminar** (selected automatically) **Flow** (selected automatically)
Material	**Liquid**

Group Box	Model
Equation of State	**Constant Density**

4. Click **Close**.

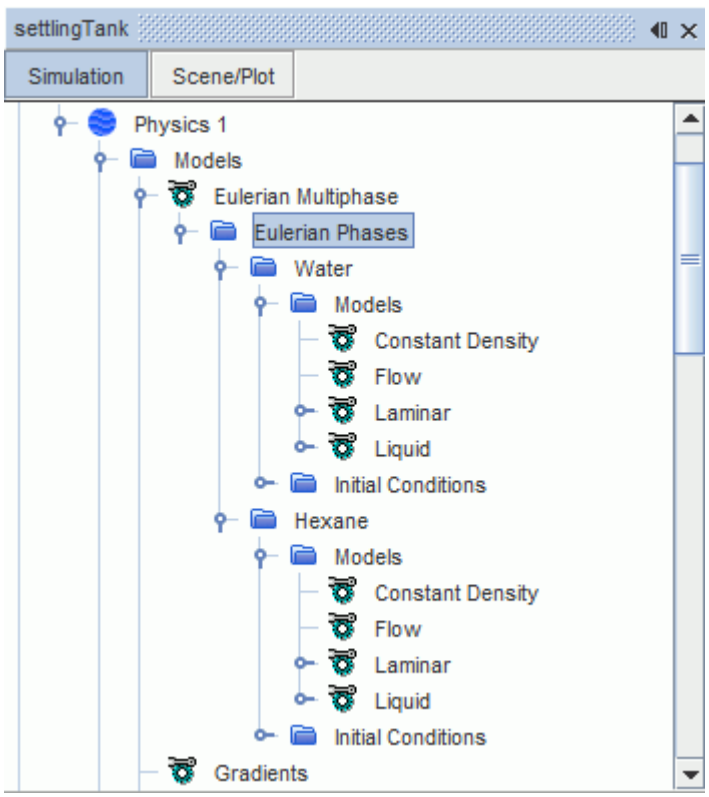
Create the lighter liquid phase:

5. Create a second phase and rename it to **Hexane**.
6. For the **Hexane** phase, select the following models in order:

Group Box	Model
Enabled Models	**Laminar** (selected automatically) **Flow** (selected automatically)
Material	**Liquid**
Equation of State	**Constant Density**

7. Click **Close**.

   The **Eulerian Phases** node appears as shown below.



For the lighter phase, replace the default fluid with Hexane.

8. Right-click **Eulerian Phases** > **Hexane** > **Models** > **Liquid** > **H2O** and click **Replace with...**.
9. In the *Replace Material* dialog, select **C6H14 (Hexane)**.

The default material properties of both fluids are suitable for this simulation.

10. Save the simulation.

Defining the Phase Interactions

Define the interactions between the heavy and light fluid phases.

Using the **Multiphase Interaction** model, you can define the interaction between the water and hexane phases. The water phase is specified as continuous, while the hexane phase is specified as dispersed.

To define the phase interactions:

1. In the **Physics 1** continuum, select the **Models** > **Multiphase Interaction** > **Phase Interactions** node and create a phase interaction.
2. Rename the **Phase Interaction 1** node to **Water-Hexane**.
3. For the **Water-Hexane** phase interaction, select the following models:

Group Box	Model
Phase Interaction Topology	**Multiple Flow Regime Phase Interaction**
Various Groups	**Drag Force** (selected automatically) **Multiphase Material** (selected automatically) **Interaction Length Scale** (selected automatically) **Interaction Area Density** (selected automatically)
Optional Models	**Surface Tension Force**

4. Click **Close**.
5. Edit the **Water-Hexane** > **Models** node and set the following properties:

Node	Property	Setting
Multiple Flow Regime Phase Interaction	**Primary Phase**	**Water**
	Secondary Phase	**Hexane**
Multiphase Material > **Multiphase Material** > **Material Properties** > **Surface Tension** > **Constant**	**Value**	`0.35 N/m`

6. Save the simulation.

Setting Initial Conditions and Reference Values

Define the initial volume fractions of each phase and set the reference density for the physics continuum.

Where there is large interface present at the start of the simulation, it is general practice to define the volume fraction of a phase at a cell to either 1.0 or 0.0 according to the location of the interface. This definition produces a 'sharp' volume fraction field: the transition of the volume fraction from 1.0 to 0.0 occurs within a single cell. However, in simulations that include the surface tension force, such a sharp volume fraction field can lead to numerical instabilities.

For simple interface shapes (such as linear or circular), you are advised to define a smooth volume fraction field by varying the volume fraction from 1.0 to 0.0 across two or three cells from the interface. For a box that has a horizontal interface at $y = y_{mean}$ with the primary phase on top, the procedure to define a smooth volume fraction across the interface is as follows:

1. Define a normal distance function, ϕ, from the interface.

   For instance, the distance function is defined as $\phi = y - y_{mean}$. Thus ϕ is positive in the primary fluid and negative in the secondary fluid. It is zero at the interface.
2. Define the thickness of the interface, 2ϵ, that is, the number of cells across which the transition occurs. Generally $2\epsilon = 3\Delta y$, which is the value that is used in this case. However, this value is a user input that depends on the specific details of the scenario that you are modeling.

   Once the interface thickness is selected, the smooth volume fraction of the primary phase is calculated as:

$$\alpha_p = \begin{cases} 0 & if \phi < -\epsilon \\ \frac{\phi+\epsilon}{2\epsilon} + \frac{1}{2\pi}\sin\left(\frac{\pi\phi}{\epsilon}\right) & if|\phi| \le \epsilon \\ 1 & if \phi > \epsilon \end{cases}$$

3. Thereafter, the secondary phase volume fraction is $\alpha_s = 1.0 - \alpha_p$

The physical interpretation of this definition is available in "

[798] *On derivation and physical interpretation of level set method based equations for two-phase flow simulations*

", Gada, Vinesh H., Sharma, Atul, Numerical Heat Transfer, Part B: Fundamentals, V 56(4), pp. 307–322, 2009.

This tutorial uses field functions to initialize the volume fraction for each of the phases within the tank. The field functions implement the volume fraction smoothing definition that is described above.

To set the initial conditions and reference values:

Create the field functions to define the initial volume fractions of each phase.

1. Define the field functions as follows:
   a) Right-click the **Tools** > **Field Functions** node and select **New** > **Scalar**.
   b) Rename the **User Field Function 1** node to `Initial_VF_Water`.
   c) Select the **Initial_VF_Water** node and set the following properties:

Property	Setting
Function Name	`Initial_VF_Water`
Definition	`1.0-${Initial_VF_Hexane}`

2. Use the same procedure to create the required field functions, using the details below:

Name / Function Name	Definition
`Initial_VF_Hexane`	`${HighVF_Anly}`

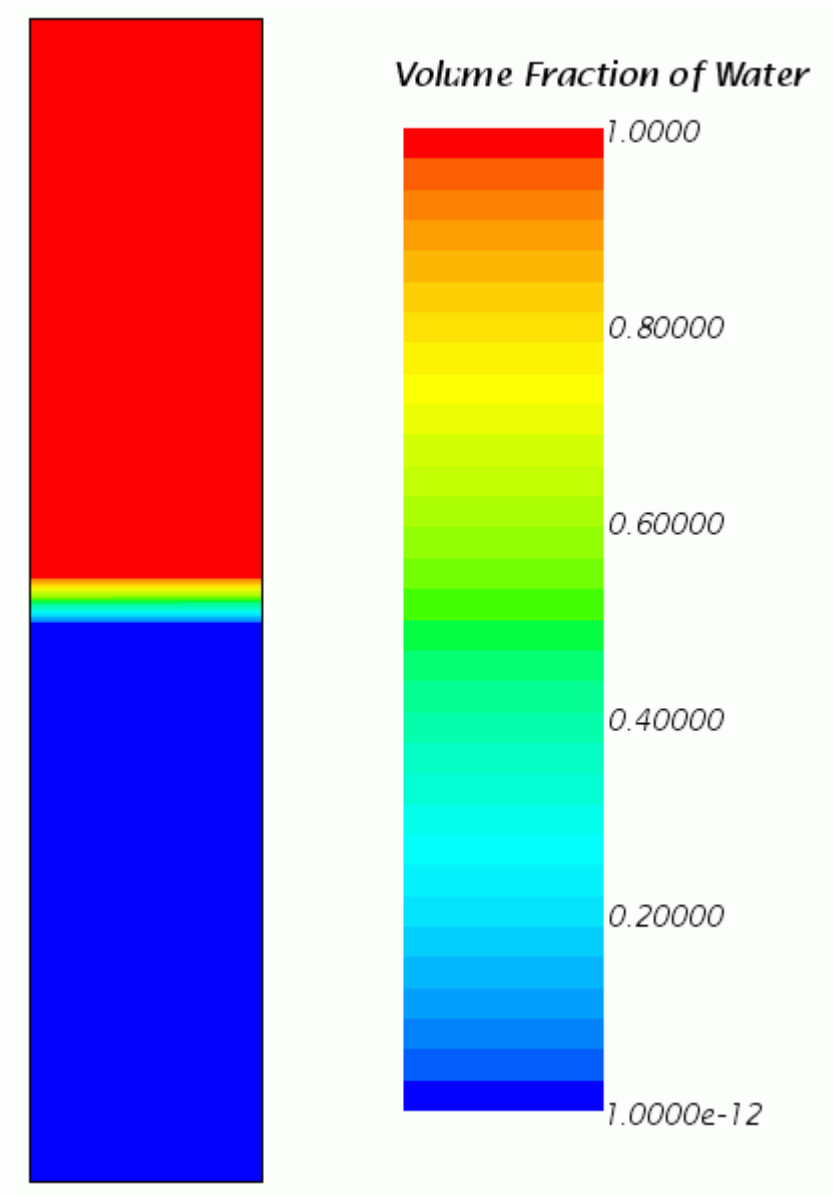
Name / Function Name	Definition
HighVF_Anly	(${Phi_Anly-LSF} > ${EpsilonPhi_Anly_HalfInterfaceThickness}) ? 1.0 : ${LowVF_Anly}
LowVF_Anly	(${Phi_Anly-LSF} < -${EpsilonPhi_Anly_HalfInterfaceThickness}) ? 0 : ${HSmooth_Anly}
EpsilonPhi_Anly_HalfInterfaceThickness	1*sqrt(${Volume})
Phi_Anly-LSF	${yMean}-${Profile}
HSmooth_Anly	(${EpsilonPhi_Anly_HalfInterfaceThickness}+${Phi_Anly-LSF})/(2*${EpsilonPhi_Anly_HalfInterfaceThickness}) + sin(${PI}*${Phi_Anly-LSF}/${EpsilonPhi_Anly_HalfInterfaceThickness})/(2*${PI})
Profile	$${Centroid}[1]-0.0001*sin($${Centroid}[0]*${PI}*1.0/0.2)
yMean	0.25
PI	3.14159265359

Set the initial conditions for the **Physics 1** continuum.

3. Edit the **Physics 1** > **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and set the following properties:

Node	Property	Setting
Water > **Initial Conditions** > **Volume Fraction**	**Method**	**Field Function**
	Scalar Function	Initial_VF_Water
Hexane > **Initial Conditions** > **Volume Fraction**	**Method**	**Field Function**
	Scalar Function	Initial_VF_Hexane

The diagram below shows the initial volume fraction of each phase with the heavier water phase lying above the lighter hexane phase.



Large Scale Interface phase interactions do not have a clear reference phase because either of the phases can be dispersed in the other phase. In this scenario, the density variation can be significant and can result in numerical instabilities. To obtain a robust and stable numerical solution, set the reference density to zero. This setting ensures that the full body force due to gravity is used in the momentum equation.

4. Edit the **Reference Values** node and set the following property:

Node	Property	Setting
Reference Density	**Value**	`0.0 kg/m^3`

5. Save the simulation.

Setting Solver Parameters and Stopping Criteria

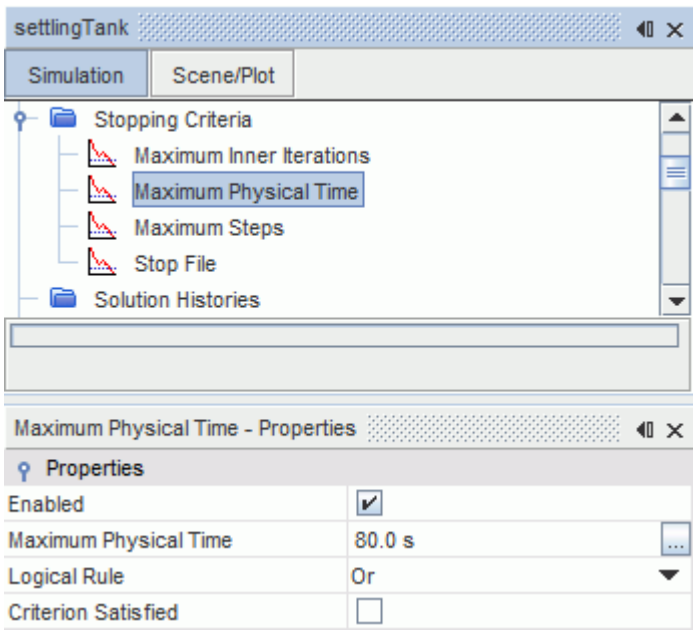
Set the appropriate solver parameters and stopping criteria for the simulation.

As you are solving an unsteady problem, you specify the time-step size and the elapsed simulation time. This calculation runs for a maximum physical time of 80 seconds with a time-step size of 0.01 s.

To set the solver parameters and stopping criteria:

1. Edit the **Solvers** > **Implicit Unsteady** node and set the **Time-Step** to `0.01 s`.
2. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Inner Iterations	**Maximum Inner Iterations**	`5`
Maximum Physical Time	**Maximum Physical Time**	`80.0 s`
Maximum Steps	**Enabled**	Deactivated



3. Save the simulation.

Setting up a Monitor

Create a monitor to help assess convergence of the simulation.

In this tutorial, you monitor the volume fractions at the center of the fluid domain.

To set up a monitor:

1. Right-click the **Derived Parts** node and select **New Part** > **Probe** > **Point**.
2. In the *Create Point Probe* dialog, make the following settings:

Setting	Value
Input Parts	**Fluid 2D**
Point	`[0.05, 0.5, 0.0]`
Display	**No Displayer**

3. Click **Create** and then **Close**.

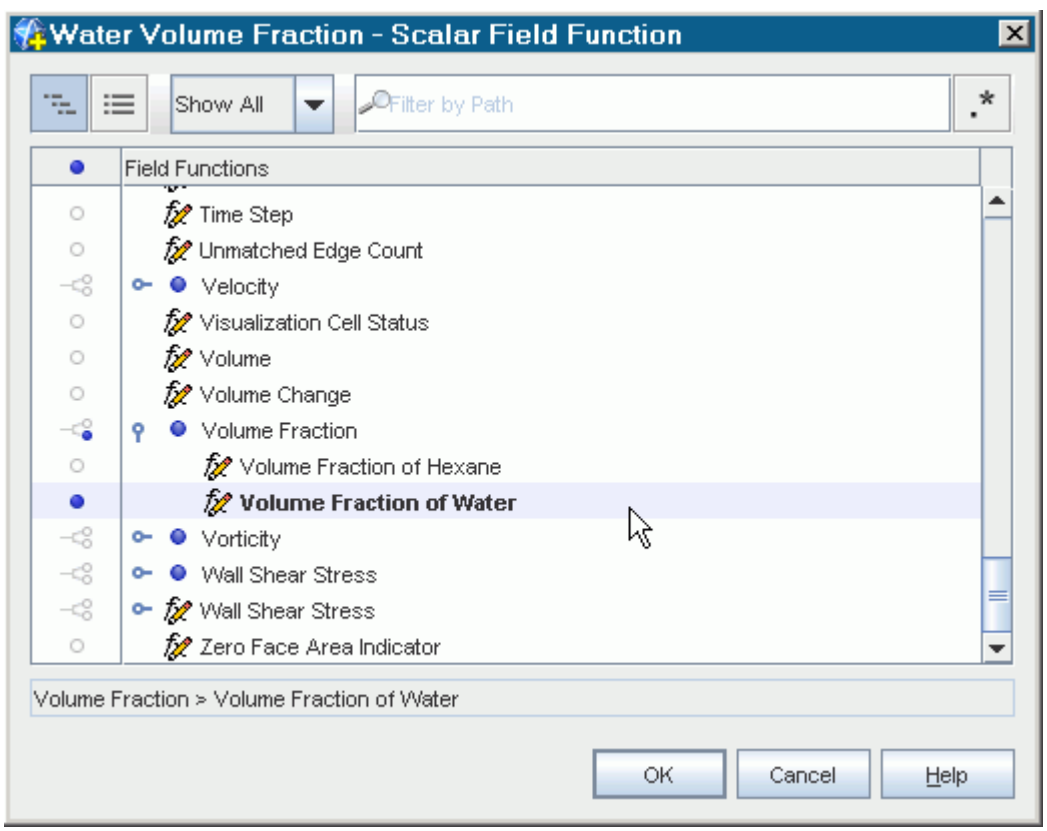
   A new derived part is created called **Point**.

This point is used to monitor the water and hexane volume fractions in the fluid. Before you can create the monitor, create the report upon which these monitors are based. In this case, you create a Maximum report.

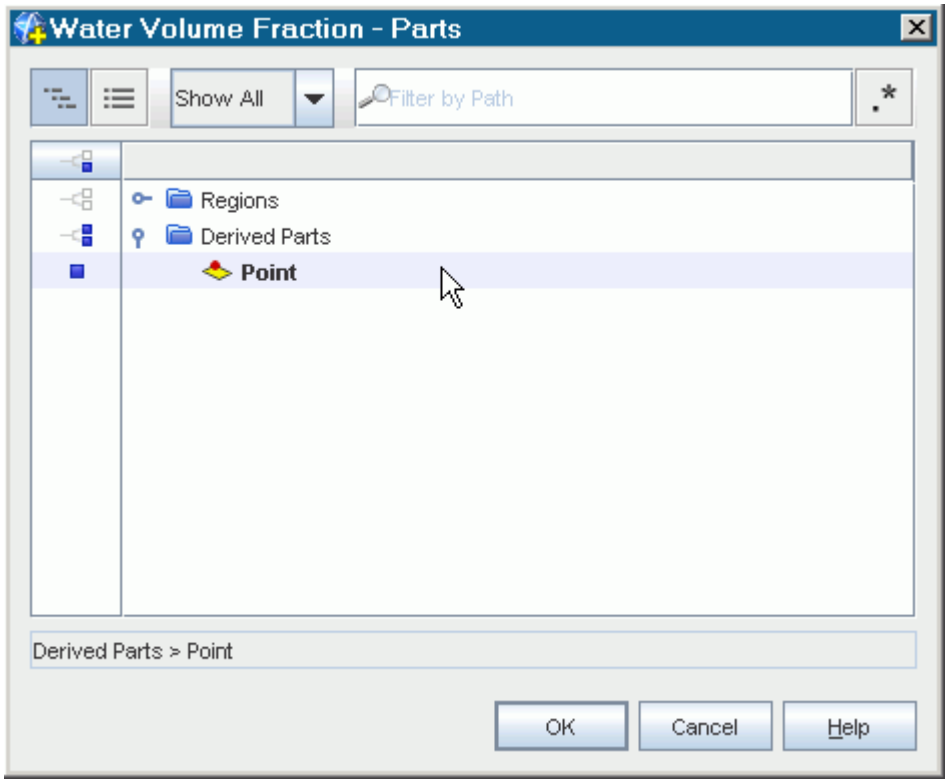
4. Right-click the **Reports** node and select **New Report** > **Maximum**.

A new report is created named **Maximum 1**, which is used to monitor the water volume fraction. Give the report a more suitable name.

5. Rename the **Maximum 1** node to `Water Volume Fraction`.
6. Select the **Reports** > **Water Volume Fraction** node and set *Scalar Field Function* to **Volume Fraction** > **Volume Fraction of Water**.



7. Set *Parts* to **Derived Parts** > **Point**.



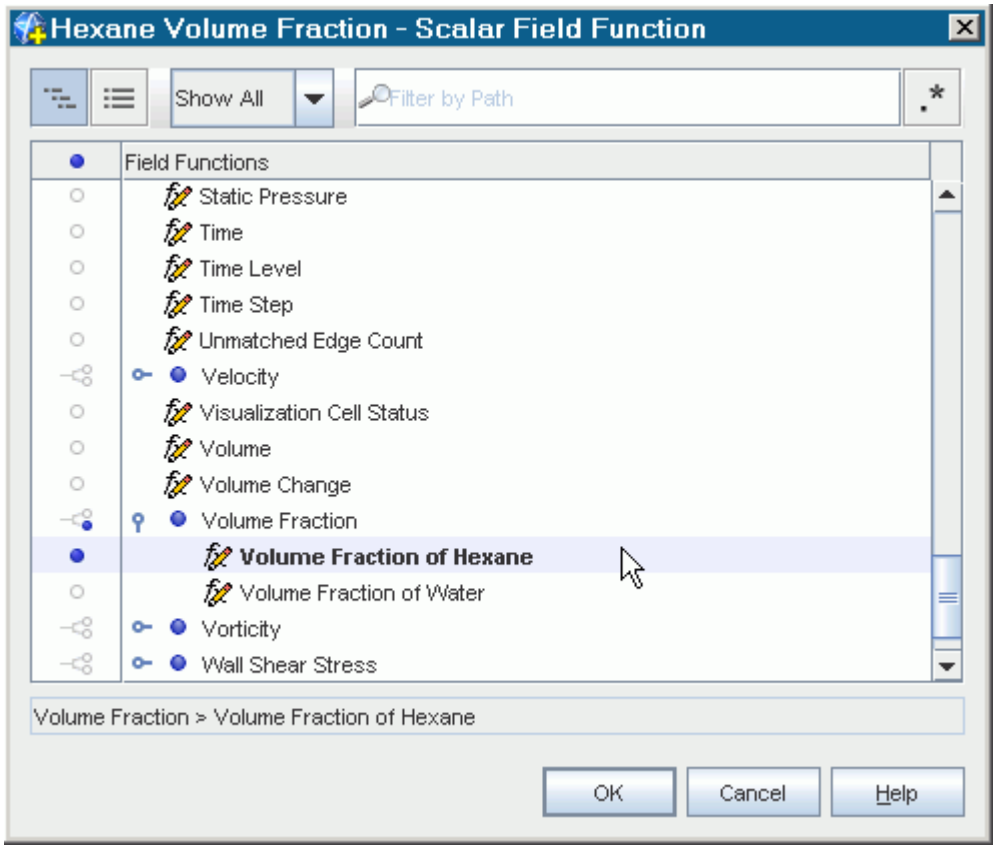
Create the report for the air volume fraction by copying and pasting the report for the water volume fraction.

8. Right-click the **Reports** > **Water Volume Fraction** node and select **Copy**.
9. Right-click the **Reports** node and select **Paste**.

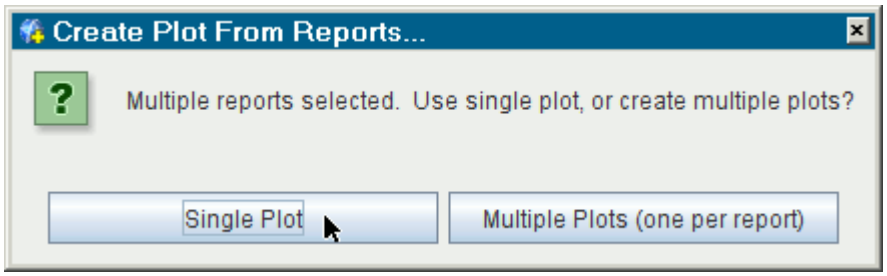
   A new report is created, named **Copy of Water Volume Fraction**.

Give the report a more suitable name.

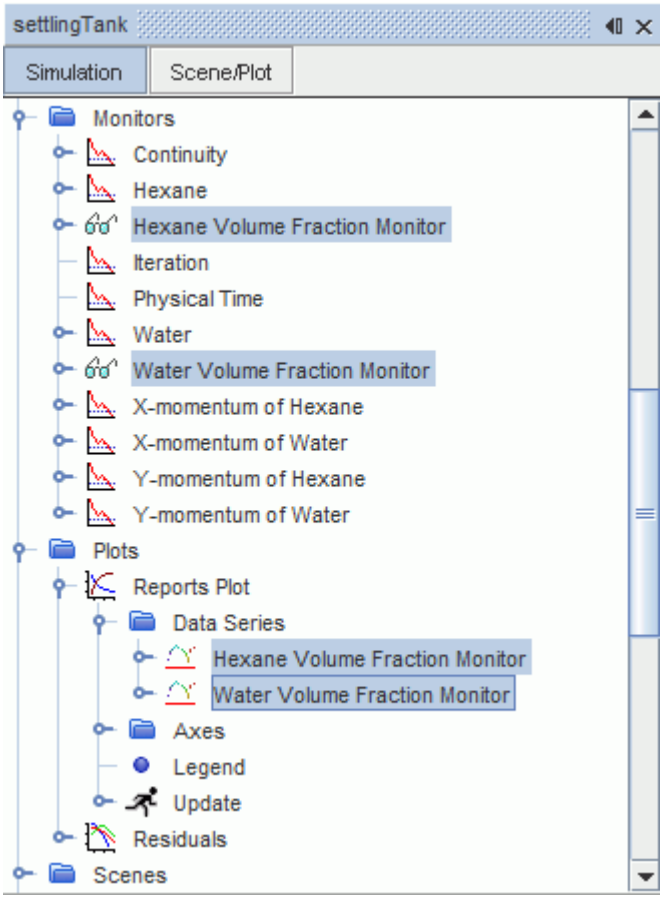
10. Rename the **Copy of Water Volume Fraction** node to `Hexane Volume Fraction`.
11. Select the **Reports** > **Hexane Volume Fraction** node and set *Scalar Field Function* to **Volume Fraction** > **Volume Fraction of Hexane**.



12. Multi-select the **Hexane Volume Fraction** and **Water Volume Fraction** nodes. Right-click the selected nodes and select **Create Monitor and Plot from Report**.
13. In the *Create Plot From Reports* dialog, click **Single Plot**.



A new monitor and plot are created.



Creating a Scalar Scene

Create a scalar scene to visualize the water volume fraction.

To create a scalar scene:

1. Right-click the **Scenes** node, and select **New Scene** > **Scalar**.
2. Click **Scene/Plot** above the object tree.
3. Select the **Displayers** > **Scalar 1** > **Scalar Field** node and set the *Function* to **Volume Fraction** > **Volume Fraction of Water**.
4. Select the **Scalar 1** node and set the *Contour Style* to **Smooth Filled**.
5. Select the **Color Bar** node and do the following:
   a) Set the *Orientation* to **Vertical**.
   b) Set the *Width* to `0.1`.
   c) Set the *Height* to `0.8`.
   d) Set the *Position* to `[0.7, 0.1]`.
6. Click the **Simulation** button to return to the simulation object tree.
7. Save the simulation.

Running the Simulation

Preparation of the simulation is now complete, and the simulation can be run.

To run the simulation:

1. Click (**Run**) in the *Solution* toolbar.

   The solution progress is displayed in the *Output* window. The *Residuals* display is created automatically in the *Graphics* window and shows the progress of the solvers.

   While the simulation is running, you can click the tabs at the top of the *Graphics* window to view scenes and plots.

   During the run, you can stop the process by clicking (**Stop**) on the toolbar. If you do stop the simulation, you can click (**Run**) to resume it.

   When the last iteration is complete, the following message is displayed in the *Output* window:

   `Stopping criterion Maximum Physical Time satisfied.`

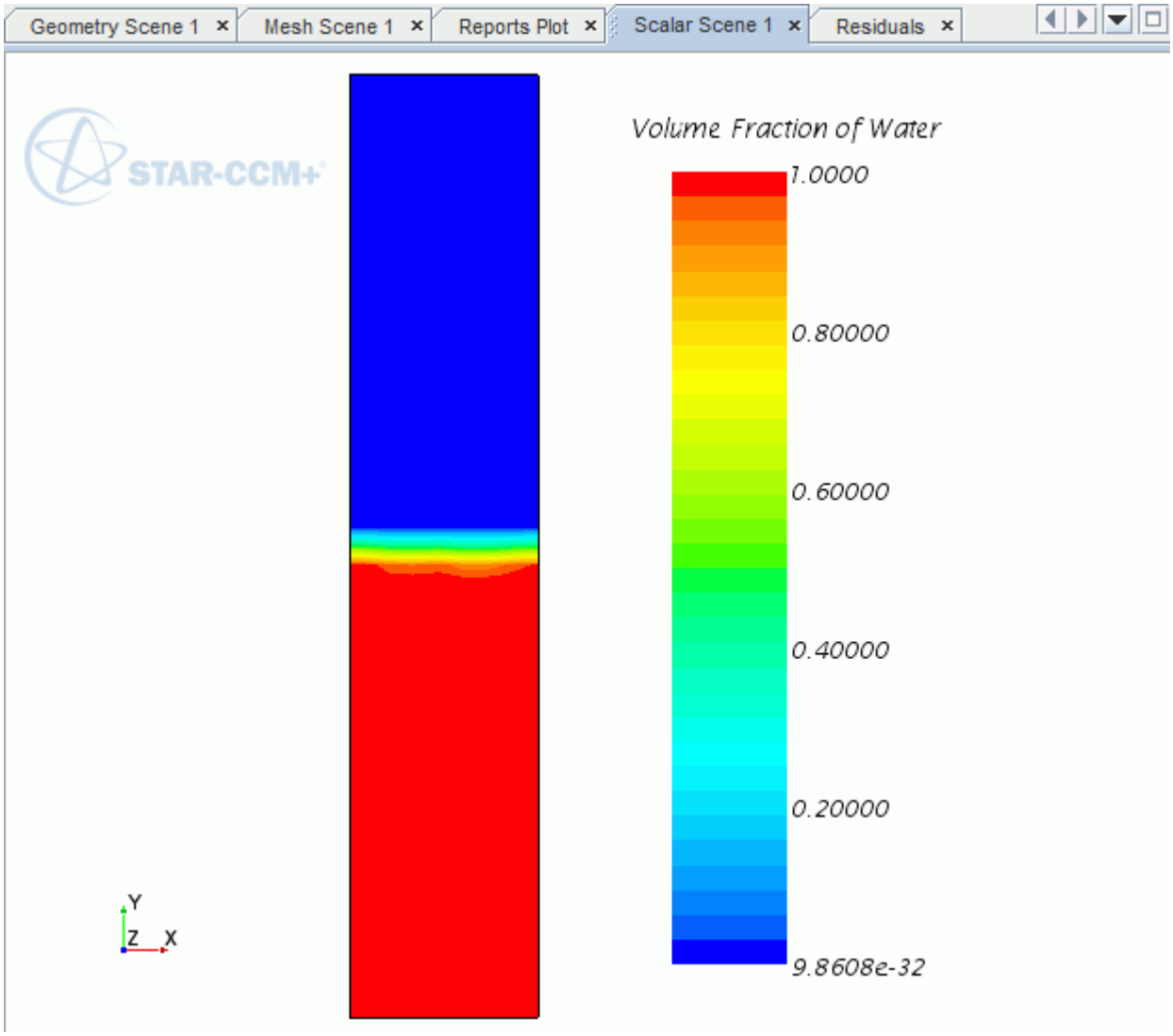
2. When the simulation has finished running, save it.

Visualizing the Results

View the results of the simulation.

To visualize the results:

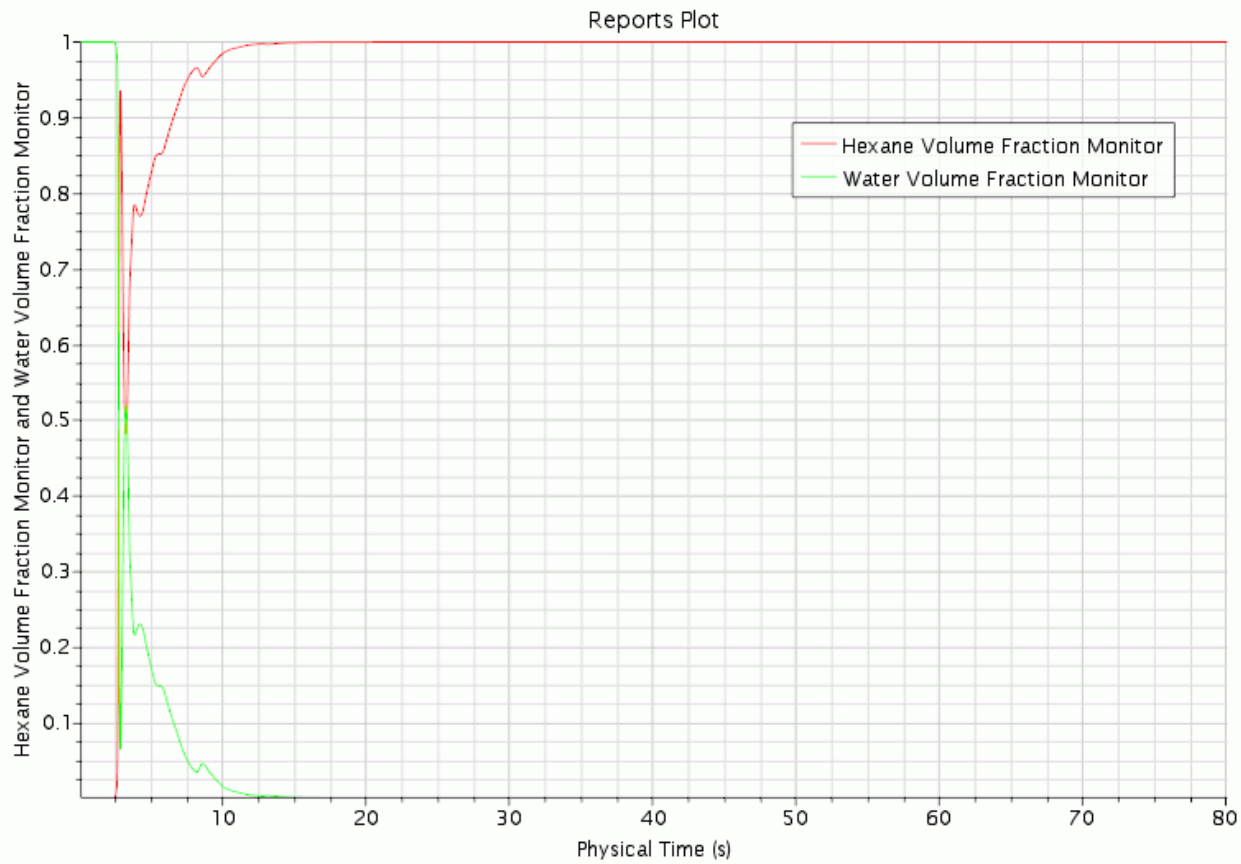
1. Click the *Scalar Scene 1* tab at the top of the *Graphics* window.



As expected, water settles down to the bottom of the tank. Hexane, being lighter than water, moves to the top of the tank.

2. Right-click the **Plots** > **Reports Plot** and select **Open**.

   The Reports Plot is shown below:



The volume fractions of water and hexane at the mid point of the fluid domain settle to the expected values of 0.0 and 1.0, respectively. This result clearly indicates that the solution is stable and is converged.

Summary

This tutorial demonstrated how to model mixture settling in STAR-CCM+.

This tutorial has introduced the following STAR-CCM+ features:

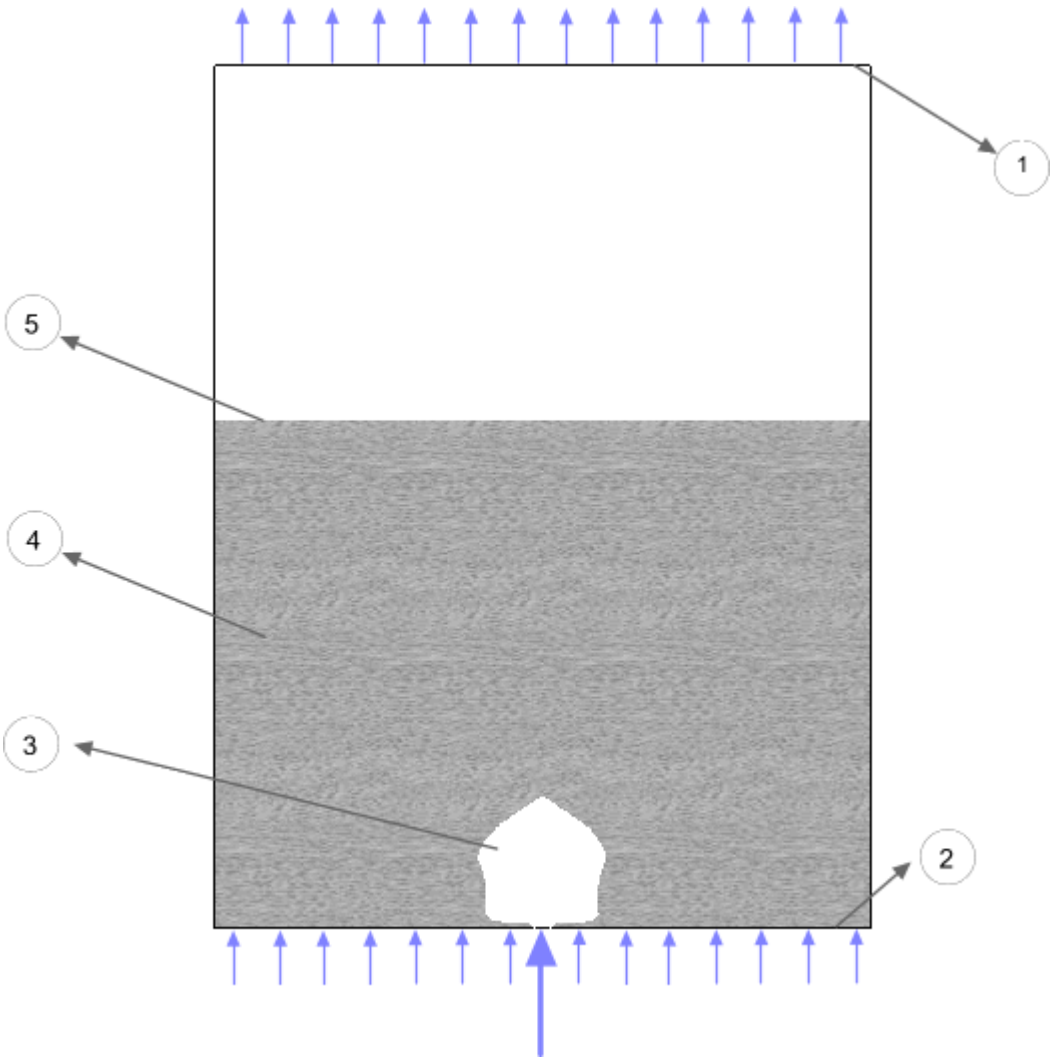
- Setting up the physics models.
- Creating phases and selecting phase models.
- Defining phase interaction.
- Setting model properties.
- Running the simulation.
- Visualizing the results.

Eulerian: Bubble Formation in a Fluidized Bed

This tutorial demonstrates how to model bubble formation in a fluidized bed in STAR-CCM+. It simulates the injection of gas into a fluidized bed and monitors the formation and growth of the bubble.

Fluidized beds are formed when a fluid is passed through a solid particulate substance to cause the mixture to flow like a fluid. The formation and propagation of bubbles within the bed affect its heat and mass transfer properties.

The shape of the bubble as well as its quantitative growth has been reported ([799]). The gas injection orifice is 15 mm wide. The superficial gas velocity at the orifice is 10 m/s. The rest of the inlet has a superficial gas velocity of 0.25 m/s; the minimum gas velocity to fluidize the bed. The geometry is shown below.



Index	Description
1	Air outlet

Index	Description
2	Inlet with air orifice
3	Bubble
4	Fluidized bed
5	Free surface

Contents:

Prerequisites
Importing the Mesh and Naming the Simulation
Setting Up the Physics Models
Creating Phases and Selecting Phase Models
Defining the Phase Interactions
Setting Initial Conditions
Setting Boundary Conditions
Setting Solver Parameters and Stopping Criteria
Setting Up a Monitor
Creating a Scalar Scene
Running the Simulation
Visualizing the Results
Summary
Bibliography

Prerequisites

The instructions in the Eulerian: Bubble Formation in a Fluidized Bed tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Importing the Mesh and Naming the Simulation

Launch STAR-CCM+, import the supplied geometry, and save the simulation.

A two-dimensional, trimmed cell mesh has been prepared for this analysis. This mesh is relatively coarse, suitable only for a tutorial.

To import the mesh and name the simulation:

1. Launch STAR-CCM+.

2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh**.
4. In the *Open* dialog, choose `doc/startutorialsdata/multiphaseFlow/data/fluidizedbed.ccm`, then click **Open**.

   STAR-CCM+ provides feedback on the import process, which takes a few seconds, in the *Output* window.
5. Save the simulation as `BubbleFormation.sim`.

Setting Up the Physics Models

Physics models define the primary variables of the simulation, including pressure, temperature, velocity, and what mathematical formulation is used to generate the solution.

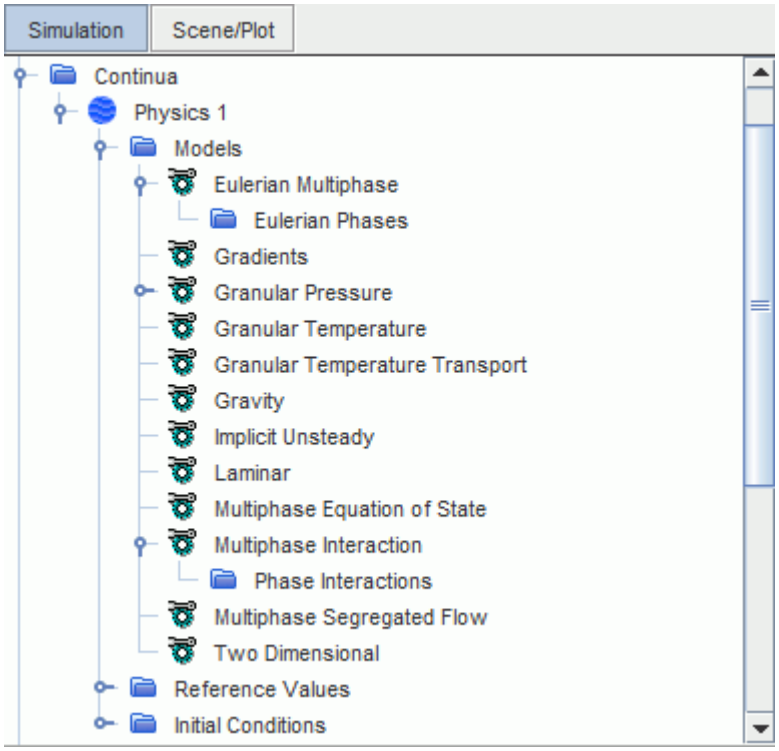
The main phenomenon to solve for is the interaction of a granular Eulerian phase with a fluid Eulerian phase.

To select the physics models:

1. For the physics continuum, **Continua** > **Physics 1**, select the following models in order:

Group Box	Model
Enabled Models	**Two Dimensional** (Pre-selected)
Time	**Implicit Unsteady**
Material	**Eulerian Multiphase** **Multiphase Interaction** (Selected automatically)
Eulerian Multiphase Model	**Multiphase Segregated Flow** **Multiphase Equation of State** (Selected automatically) **Gradients** (Selected automatically)
Viscous Regime	**Laminar**
Optional Models	**Gravity** **Granular Pressure** **Granular Temperature** (Selected automatically) **Granular Temperature Transport** (Selected automatically)

2. Click **Close**.
3. To review the models, open the **Physics 1** > **Models** node.



You now set the appropriate properties.

4. Select **Continua** > **Physics 1** > **Models** > **Granular Pressure** and set **Frictional Stress Formulation** to **Schaeffer**.
5. Select the **Granular Pressure** > **Schaeffer** node and set the **Angle of Internal Friction** to `25.0`.

Use second-order velocity convection.

6. Select the **Continua** > **Physics 1** > **Models** > **Multiphase Segregated Flow** node and set the **Volume Fraction Convection** property to **2nd-order**.
7. Make sure that the **Velocity Convection** is **2nd-order**.
8. Save the simulation.

Creating Phases and Selecting Phase Models

Create two new phases, specify them as air and particle, and set the appropriate material properties for each phase.

To create the phases and select the phase models:

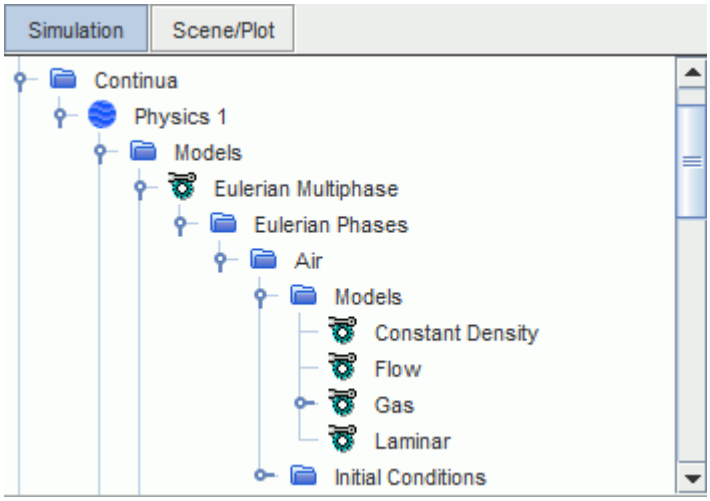
Create the gas phase.

1. In the **Physics 1** continuum, right-click the **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and create a new phase.
2. Rename the **Phase 1** node to **Air**.
3. For the **Air** phase, select the following models:

Group Box	Model
Enabled Models	**Laminar** (Selected automatically) **Flow** (Selected automatically)
Material	**Gas**
Equation of State	**Constant Density**

4. Click **Close**.

   The **Air** phase node appears as shown below:



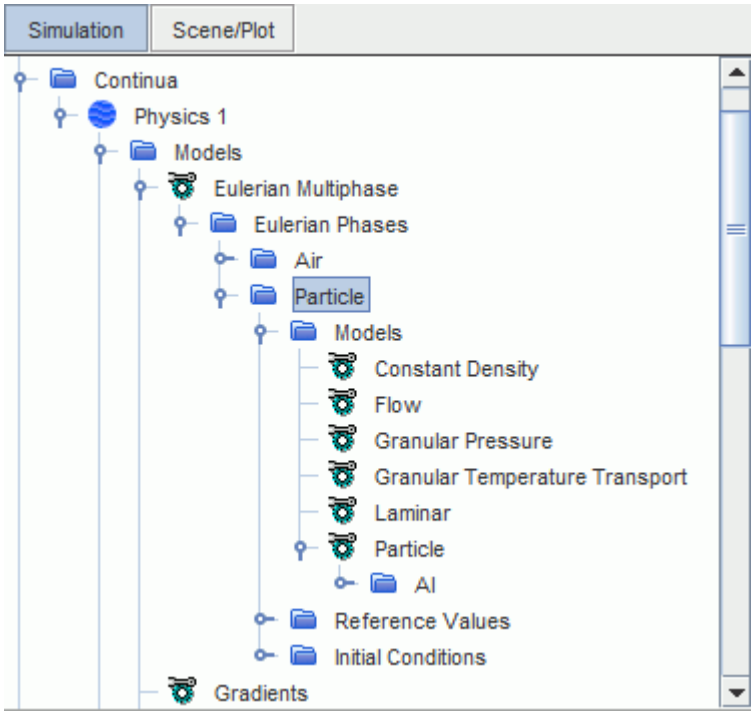
Create the particle phase.

5. Create a second phase and rename it to **Particle**.
6. For the **Particle** phase, select the following models:

Group Box	Model
Enabled Models	**Laminar** (Selected automatically) **Flow** (Selected automatically)
Material	**Particle**
Enabled Models	**Constant Density** (Selected automatically) **Granular Temperature Transport** (Selected automatically) **Granular Pressure** (Selected automatically)

7. Click **Close**.

   The **Particle** phase node appears as shown below:



Specify the material properties for the gas phase.

8. Edit the **Eulerian Phases** > **Air** > **Models** > **Gas** > **Air** > **Material Properties** node and set the following properties:

Node	Property	Setting
Density > **Constant**	**Value**	`1.28 kg/m^3`
Dynamic Viscosity > **Constant**	**Value**	`1.7E-5 Pa-s`

Specify the material properties for the particle phase.

9. Edit the **Eulerian Phases** > **Particle** > **Models** > **Particle** > **Al** > **Material Properties** node and set the following properties:

Node	Property	Setting
Density > **Constant**	**Value**	`2660.0 kg/m^3`
Particle Diameter > **Constant**	**Value**	`5.0E-4 m`

Set the maximum solid fraction.

10. Edit the **Eulerian Phases** > **Particle** > **Models** > **Granular Pressure** node and set the **Maximum Solid Fraction** to `0.65`.

Specify a reference value for the particle coefficient of restitution.

11. Edit the **Eulerian Phases** > **Particle** > **Reference Values** > **Coefficient of restitution** node and set the **Value** to `0.9`.
12. Save the simulation.

Defining the Phase Interactions

Define the interactions between the phases.

Using the **Multiphase Interaction** model, you can define the interaction between the air and particle phases.

To define the phase interactions:

1. In the **Physics 1** continuum, right-click the **Models** > **Multiphase Interaction** > **Phase Interactions** node and create a new phase interaction.
2. For the **Phase Interaction 1** phase interaction, select the following models:

Group Box	Model
Phase Interaction Topology	**Continuous-Dispersed Phase Interaction**
Enabled Models	**Drag Force** (Selected automatically) **Multiphase Material** (Selected automatically) **Interaction Length Scale** (Selected automatically) **Interaction Area Density** (Selected automatically)

3. Click **Close**.
4. Edit the **Phase Interaction 1** > **Models** node and set the following properties:

Node	Property	Setting
Continuous-Dispersed Phase Interaction	**Continuous Phase**	**Air**
	Dispersed Phase	**Particle**
Drag Force > **Drag Coefficient**	**Method**	**Gidaspow**
Drag Force > **Drag Coefficient** > **Gidaspow**	**Transition Volume Fraction**	`0.2`

5. Click **Close**.
6. Save the simulation.

Setting Initial Conditions

Set the initial conditions for the physics continuum.

Initially, the fluidized bed contains 40.2% particles and is 0.5 m deep. Before bubble formation, the particles are motionless and the air velocity sustains the fluidization of the bed. You can use field functions to set the initial volume fraction for each phase and the initial velocity of air.

To set the initial conditions:

Create the field functions for the initial volume fraction for each phase and the initial velocity of air.

1. Define the field function for the initial particle volume fraction as follows:
   a) Right-click the **Tools** > **Field Functions** node and select **New** > **Scalar**.
   b) Rename the **User Field Function 1** node to `InitialVFParticle`.

c) Select the **InitialVFParticle** node and set the following properties:

Property	Setting
Function Name	`InitialVFParticle`
Definition	`($$Centroid[1] < 0.5) ? 0.402 : 1e-6`

2. Use the same procedure to create field functions for the initial air volume fraction and the initial velocity of air, using the details below:

Name / Function Name	Definition
`InitialVFAir`	`($$Centroid[1] < 0.5) ? 0.598 : (0.999999)`
`InitialVelAir`	`0.25/$InitialVFAir`

Set the initial conditions for the **Physics 1** continuum.

3. Edit the **Continua** > **Physics 1** > **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and set the following properties:

Node	Property	Setting
Air > **Initial Conditions**		
⌊ **Velocity**	**Method**	**Composite**
⌊ **Composite** > **Y Component**	**Method**	**Field Function**
	Scalar Function	**InitialVelAir**
⌊ **Volume Fraction**	**Method**	**Field Function**
	Scalar Function	**InitialVFAir**
Particle > **Initial Conditions**		
⌊ **Volume Fraction**	**Method**	**Field Function**
	Scalar Function	**InitialVFParticle**

4. Save the simulation.

Setting Boundary Conditions

Define the boundary types and specify the appropriate property values.

The boundaries representing the bed are no-slip walls, a velocity inlet, and a pressure outlet.

To set the boundary conditions:

Check the **LHS** and **RHS** boundary conditions.

1. Edit the **Regions** > **Fluidized_Bed** > **Boundaries** > **LHS** node and set the following property:

Node	Property	Setting
Phase Conditions > **Particle** > **Physics Conditions** > **Granular Temperature Specification**	**Option**	**Extrapolation**

2. Similarly, edit the **Regions** > **Fluidized_Bed** > **Boundaries** > **RHS** node.

To model the bubble formation, the inlet boundary requires field functions for the particle and gas volume fractions. Only gas enters through the inlet, but the granular flow model also requires an estimate of the kinetic energy of the particles.

3. Create the following scalar field functions:

Name / Function Name	Definition
`InletVFAir`	`1-$InletVFParticle`
`InletVFParticle`	`($$Centroid[0] > 0.2775) && ($$Centroid[0] < 0.2925) ? 0 : 0.402`
`InletVelAir`	`($$Centroid[0] > 0.2775) && ($$Centroid[0] < 0.2925) ? (10/$InletVFAir) : (0.25/$InletVFAir)`

Set the boundary conditions.

4. Edit the **Regions** > **Fluidized_Bed** > **Boundaries** > **Bottom** > **Phase Conditions** node and set the following properties:

Node	Property	Setting
Air > **Physics Values**		
⌊ **Velocity Magnitude**	**Method**	**Field Function**
	Scalar Function	**InletVelAir**
⌊ **Volume Fraction**	**Method**	**Field Function**
	Scalar Function	**InletVFAir**
Particle > **Physics Values**		
⌊ **Volume Fraction**	**Method**	**Field Function**
	Scalar Function	**InletVFParticle**

Allow the particles to slip along the wall boundaries.

5. Edit the **Regions** > **Fluidized_Bed** > **Boundaries** node and set the following properties:

Node	Property	Setting
LHS > **Phase Conditions** > **Particle** > **Physics Conditions** > **Shear Stress Specification**	**Method**	**Slip**
RHS > **Phase Conditions** > **Particle** > **Physics Conditions** > **Shear Stress Specification**	**Method**	**Slip**

Allow gas to escape through the **Top** boundary.

6. Edit the **Regions** > **Fluidized_Bed** > **Boundaries** > **Top** node and set the following property:

Node	Property	Setting
Phase Conditions > **Particle**	**Type**	**Phase Impermeable**

7. Save the simulation.

Setting Solver Parameters and Stopping Criteria

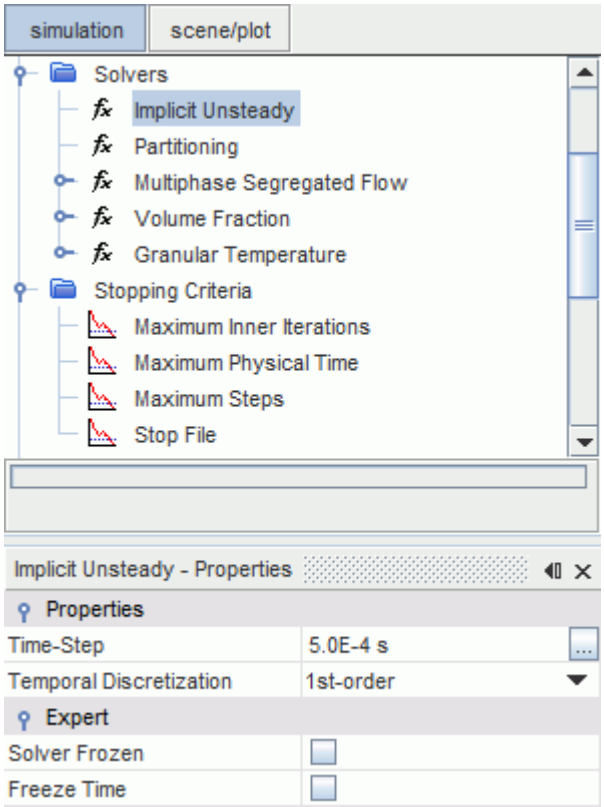
Set the appropriate solver parameters and stopping criteria for the simulation.

As you are solving an unsteady problem, it is necessary to specify the time-step size and the elapsed simulation time. This calculation runs for a maximum physical time of 0.2 seconds with a time-step size of 5.0E-4 s.

To set the solver parameters and stopping criteria:

1. Edit the **Solvers** node and set the following properties:

Node	Property	Setting
Implicit Unsteady	**Time-Step**	`5.0E-4 s`
Multiphase Segregated Flow > **Phase Coupled Velocity**	**Implicit Under-Relaxation Factor**	`0.1`
Multiphase Segregated Flow > **Pressure**	**Under-Relaxation Factor**	`0.1`
Volume Fraction	**Implicit Under-Relaxation Factor**	`0.1`



2. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Physical Time	**Enabled**	Activated
	Maximum Physical Time	`0.2 s`
Maximum Inner Iterations	**Maximum Inner Iterations**	`20`
Maximum Steps	**Enabled**	Deactivated

3. Save the simulation.

Setting Up a Monitor

Create a monitor and plot of the bubble diameter to help assess convergence of the simulation.

By default, STAR-CCM+ produces a residuals plot which can be used to monitor the convergence of the solution. It is good practice to create more monitors on physical quantities to help in assessing convergence.

The bubble edge is defined where an 85% air concentration appears in the fluidized bed.

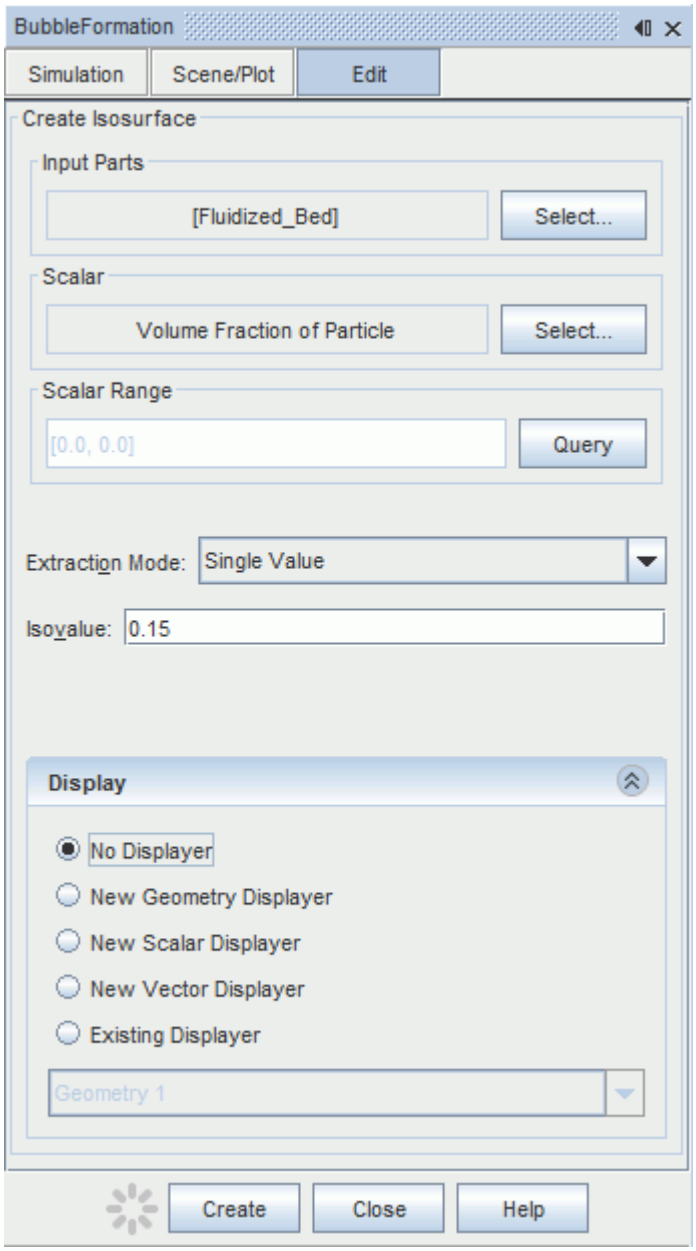
To monitor the bubble diameter:

1. Right-click the **Derived Parts** node and select **New Part** > **Isosurface**.

   The *Create Isosurface* dialog appears in the *Edit* window.

2. In the *Input Parts* box, make sure that **Fluidized_Bed** is selected.

3. Set the **Scalar** function to **Volume Fraction** > **Volume Fraction of Particle**.
4. Set the **Extraction Mode** to **Single Value**.
5. Set the **Isovalue** to `0.15`.
6. In the *Display* group box, select **No Displayer**.



7. Click **Create** and then click **Close**.

   A new derived part, **Isosurface**, is created.

However, the 85% concentration also occurs at the free surface at the top of the bed.

To remove the free surface at the top of the fluidized bed:

8. Right-click the **Derived Parts** node and select **New Part** > **Threshold**.

The *Create Threshold* dialog appears in the *Edit* window.

9. In the *Input Parts* box, select **Isosurface**.
10. Set the **Scalar** to **Centroid** > **Laboratory** > **Y**.
11. Set the **Extraction Mode** to **All below**.
12. Set the **Threshold Value** to `0.4 m`.
13. In the *Display* group box, select **No Displayer**.
14. Click **Create** and then click **Close**.

To create the field function for the line integral:

15. Right-click **Tools** > **Field Functions** and select **New** > **Scalar**.
16. Rename the **User Field Function 1** node to `Unity`.
17. In the `Properties` window, set the **Function Name** property to `Unity`.
18. Enter a **Definition** of `1`.

Create a Line Integral report along the circumference of the bubble:

19. Right-click the **Reports** node and select **New Report** > **Line Integral**.
20. Rename the **Line Integral 1** node to `Circumference`.
21. Select the **Reports** > **Circumference** node and set the **Field Function** to **Unity**.
22. Set the **Parts** property to **Derived Parts** > **Threshold**.

You can use the **Circumference** report to calculate the bubble diameter:

23. Right-click the **Reports** node and select **New Report** > **Expression**.
24. Rename the **Expression 1** node to `Bubble Diameter`.
25. Select the **Reports** > **Bubble Diameter** node and enter the definition:

    ```
    $CircumferenceReport /3.14
    ```

To create a monitor and plot of the bubble diameter at each time-step:

26. Right-click **Bubble Diameter** and set **Create Monitor and Plot from Report**.
27. Select the **Monitors** > **Bubble Diameter Monitor** node and set the **Trigger** property to **Time Step**.
28. Save the simulation.

Creating a Scalar Scene

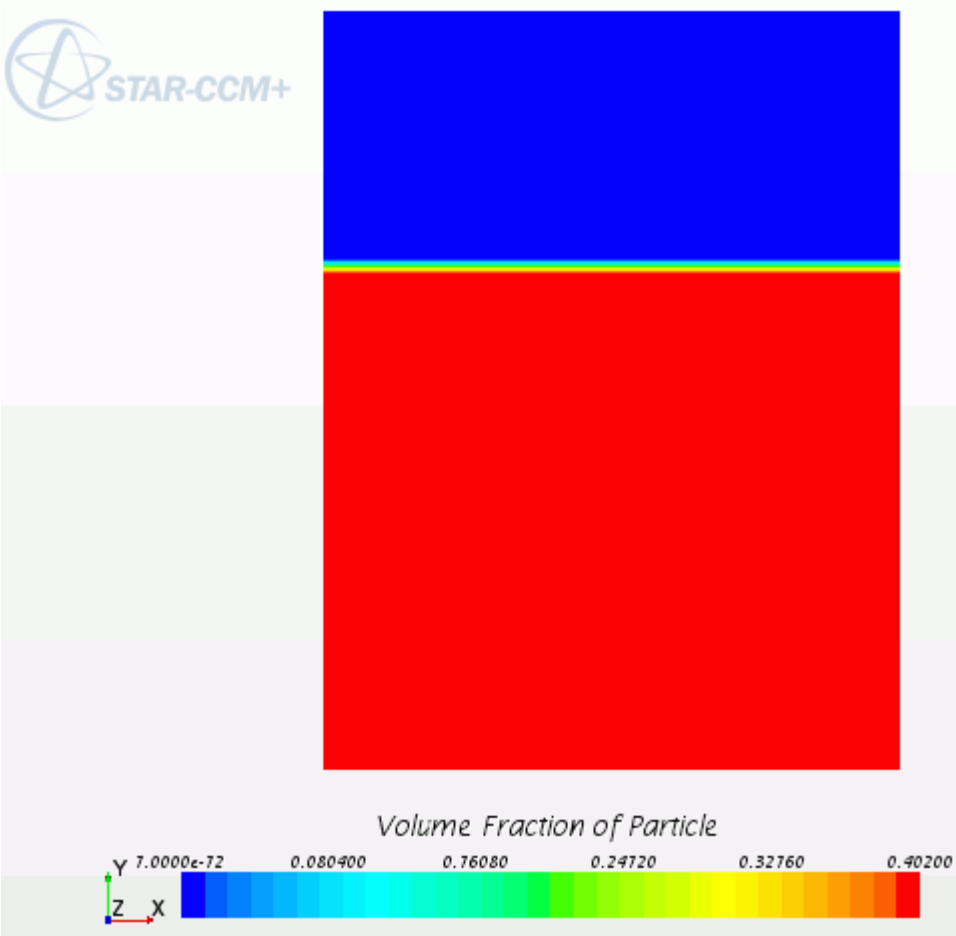
Create a scalar scene to display the particle volume fraction in the tank.

To create a scalar scene:

1. Right-click the **Scenes** node, and select **New Scene** > **Scalar**.
2. Rename the **Scalar Scene 1** node to `Bubble Shape`.
3. Click the **Scene/Plot** button above the object tree.
4. Select the **Displayers** > **Scalar 1** > **Scalar Field** node and set the **Function** property to **Volume Fraction** > **Volume Fraction of Particle**.
5. Select the **Scalar 1** node and set the **Contour Style** to **Smooth Filled**.

6. To return to the simulation object tree, click the **simulation** button.
7. Click the **Initialize Solution** button.

   The initial distribution of the phases appears in the *Graphics* window.



8. Save the simulation.

Running the Simulation

Preparation of the simulation is now complete, and the simulation can be run.

To run the simulation:

1. Click (**Run**) in the *Solution* toolbar.

   The solution progress is displayed in the *Output* window. The *Residuals* display is created automatically in the *Graphics* window and shows the progress of the solvers.

   While the simulation is running, you can click the tabs at the top of the *Graphics* window to view scenes and plots.

During the run, you can stop the process by clicking (**Stop**) on the toolbar. If you do stop the simulation, you can click (**Run**) to resume it.

When the last iteration is complete, the following message is displayed in the *Output* window:

```
Stopping criterion Maximum Physical Time satisfied.
```

2. When the simulation has finished running, save it.

Visualizing the Results

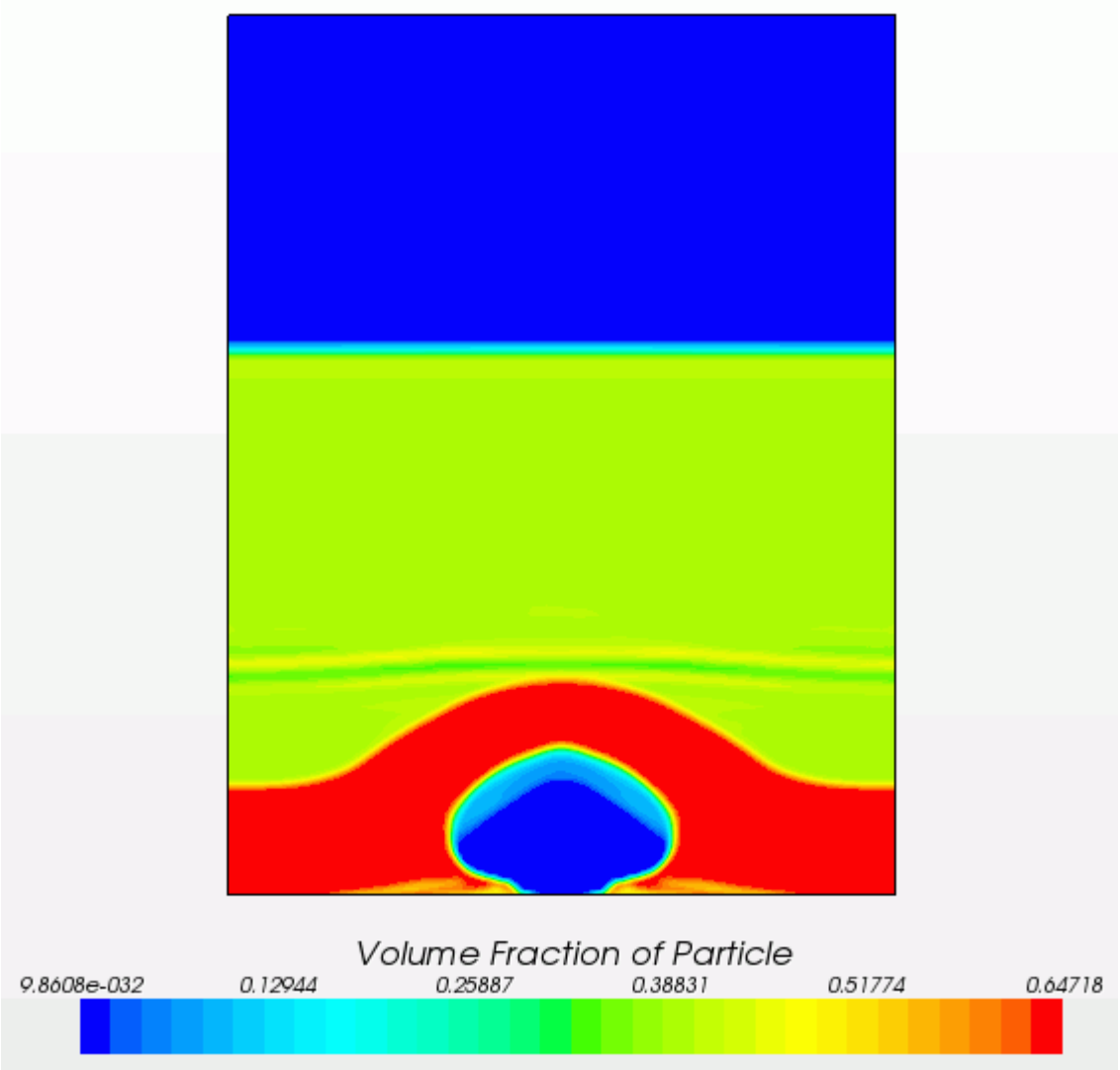
View the results of the simulation.

You can visualize the computed solution by viewing the scalar scene in the *Graphics* window.

To visualize the results of the simulation:

1. Click the *Bubble Shape* tab at the top of the *Graphics* window.

   A bubble has formed at the bottom of the fluidized bed.

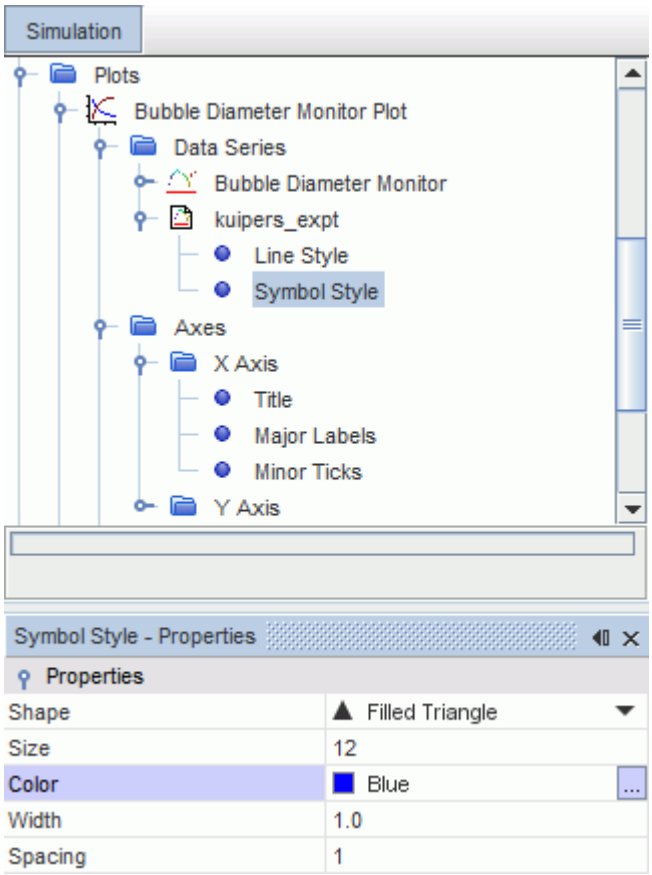


Validation data is available for this case. Import this data and use it to validate the prediction of bubble diameter.

To compare the bubble diameter with experimental results:

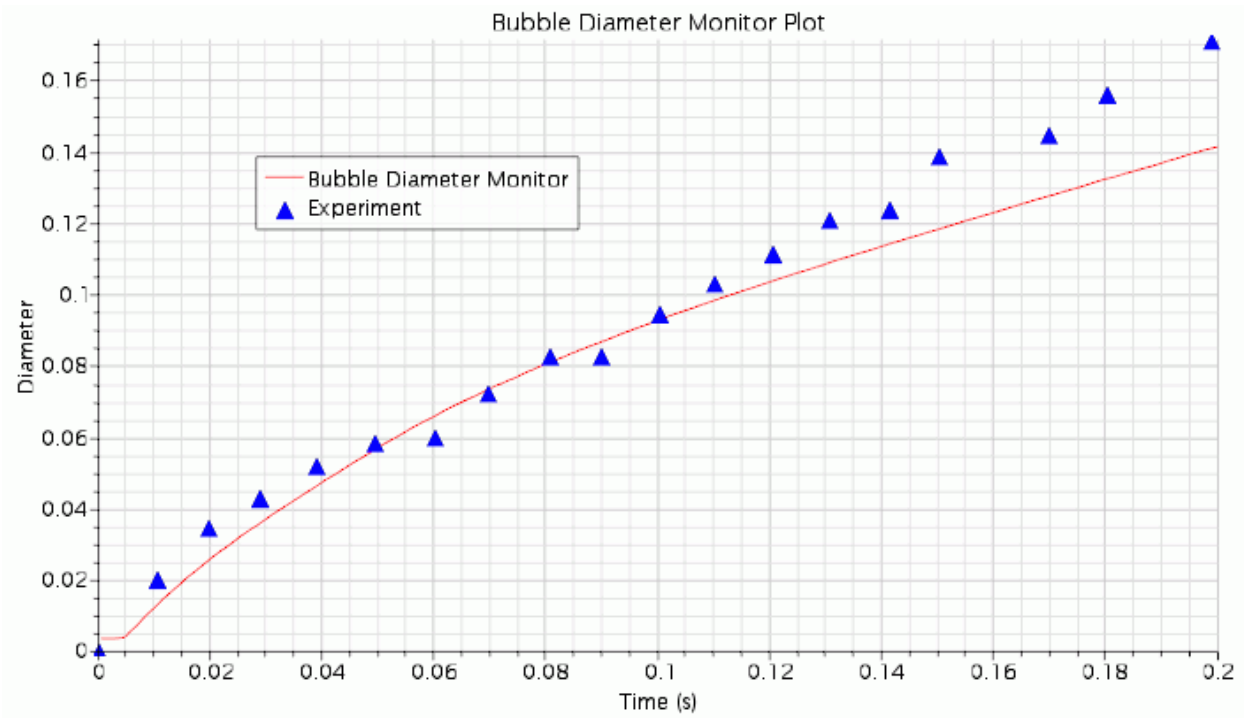
2. Right-click the **Tools** > **Tables** and select **New Table** > **File Table**.
3. In the *Open* dialog navigate to the `doc/startutorialsdata/multiphaseFlow/data` subdirectory of your STAR-CCM+ installation, and select the file `kuipers_expt.csv`.
4. Click **Open**.
5. Right-click the **Plots** > **Bubble Diameter Monitor Plot** > **Data Series** node and select **Add Data**.
6. In the *Add Data Providers to Plot* dialog, expand the **Tables** node and select **kuipers_expt**.

7. Select the **kuipers_expt** node and set the **Legend Name** to `Experiment`, the **X Column** to `time` and the **Y Column** to `dia`.
8. Select the **Line Style** node and make sure that the **Style** property is set to **None**.
9. Select the **Symbol Style** node and do the following:
   a) Change the **Size** property to `12`.
   b) Change the **Color** property to **Blue**.
   c) Change the **Shape** property to **Filled Triangle**.



10. Select the **Plots** > **Bubble Diameter Monitor Plot** > **Axes** > **Bottom Axis** node and set the **Maximum** to `0.2`.
11. Select the **Plots** > **Bubble Diameter Monitor Plot** > **Axes** > **Bottom Axis** > **Title** node and set the **Title** to `Time (s)`.
12. Select the **Plots** > **Bubble Diameter Monitor Plot** > **Axes** > **Left Axis** > **Title** node and set the **Title** to `Diameter`.

    The experimental data now appears in the same plot as the predicted bubble diameter. A reports plot scene appears as shown below:



    The results provide a reasonable agreement with the experiment.
13. Save the simulation.

Summary

This tutorial demonstrated how to model bubble formation in a fluidized bed in STAR-CCM+.

This tutorial has introduced the following STAR-CCM+ features:

- Setting up a simulation for granular flow using phase interaction models.
- Creating phases and selecting phase models.
- Defining phase interaction.
- Setting model properties.
- Creating a line integral report to monitor bubble diameter.

- Running the simulation.
- Visualizing the results.

Bibliography

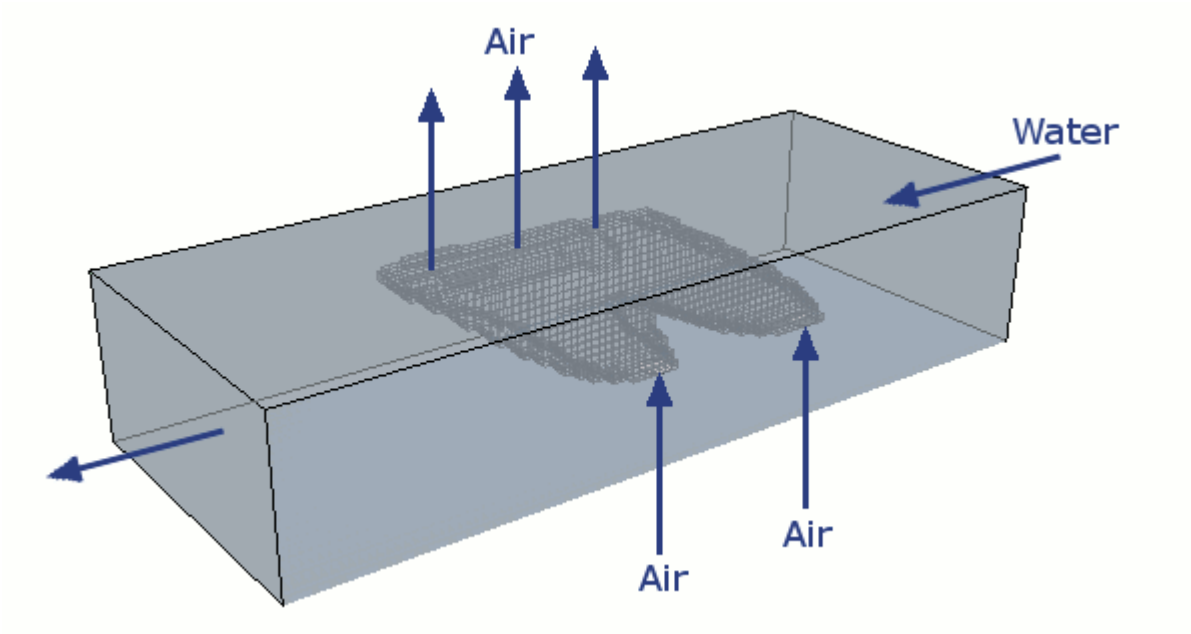
This tutorial references the following papers:

[799] *Kuipers, J.A.M. 1990. "A two-fluid micro balance model of fluidized beds", PhD Thesis, Univ. of Twente, Twente, the Netherlands*

Eulerian: Degassing Boundary

This tutorial demonstrates how to model a Eulerian multiphase turbulent flow with a degassing boundary in STAR-CCM+. It simulates a simple aeration tank, where air enters the tank through two gas injectors at the bottom of the tank and escapes through the top surface.

The geometry is shown below.



The degassing surface (the top boundary) is modeled using a frictionless phase-permeable wall. This wall can share a region with an outlet or pressure outlet boundary.

STAR-CCM+ automatically calculates the mass flow rate through the phase-permeable boundary (the degassing rate). This rate can be plotted using the predefined Mass-Flow Rate field function.

This tutorial also covers how to select the Eulerian multiphase and multiphase turbulence models, specify the properties of each phase and define the phase interaction.

Contents:

Prerequisites
Importing the Mesh and Naming the Simulation
Selecting the Physics Models
Creating Phases and Selecting Phase Models
Defining the Phase Interactions
Deactivating Secondary Gradients
Setting Initial Conditions and Reference Values
Setting Boundary Conditions

Setting Solver Parameters and Stopping Criteria
Visualizing the Solution
Running the Simulation
Visualizing the Results
Summary

Prerequisites

The instructions in the Eulerian: Degassing Boundary tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Importing the Mesh and Naming the Simulation

Launch STAR-CCM+, import the supplied geometry, and save the simulation.

A three-dimensional, trimmed cell mesh has been prepared for this analysis.

To import the mesh and name the simulation:

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh**.
4. In the *Open* dialog, choose `doc/startutorialsdata/multiphaseFlow/data/aerationTank.ccm`, then click **Open**.

   STAR-CCM+ provides feedback on the import process, which takes a few seconds, in the *Output* window.
5. Save the simulation as `aerationTank.sim`.

Create a mesh scene, and visualize the geometry:

6. Click (**Create/Open Scenes**) and select **Mesh**.
7. Use the mouse buttons to rotate and view the geometry.

Selecting the Physics Models

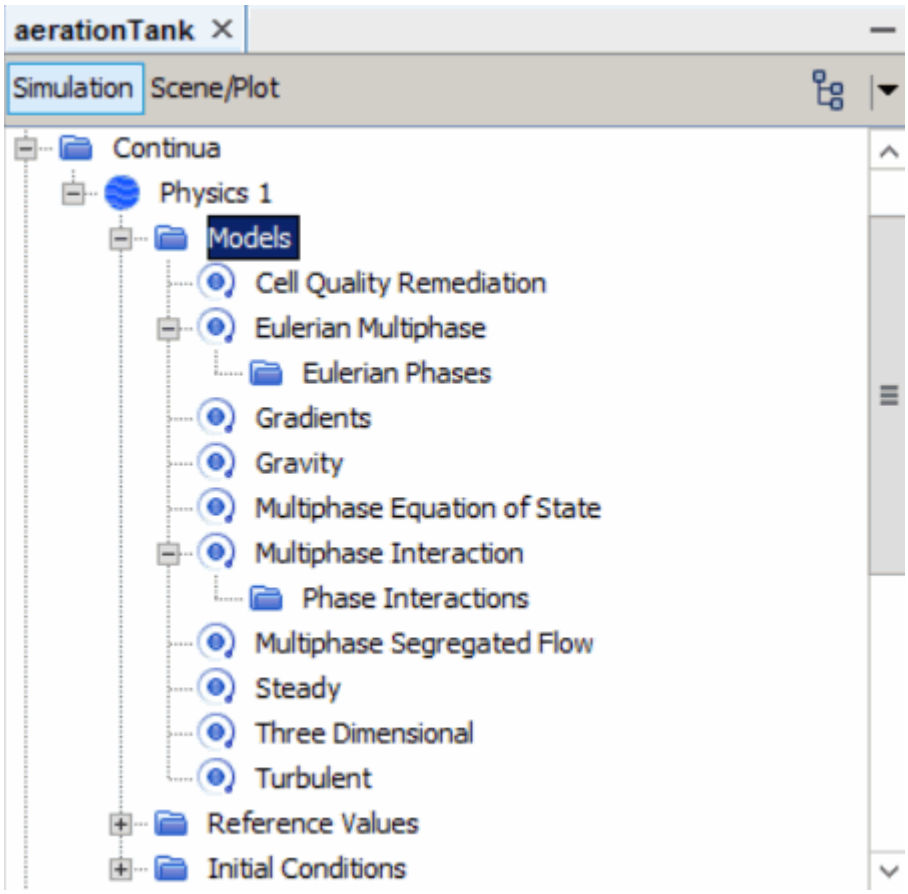
Physics models define the primary variables of the simulation, including pressure, temperature, velocity, and what mathematical formulation is used to generate the solution.

To select the physics models:

1. For the physics continuum, **Continua** > **Physics 1**, select the following models in order:

Group Box	Model
Enabled Models	**Three Dimensional** (Pre-selected)
Time	**Steady**
Material	**Eulerian Multiphase** **Multiphase Interaction** (Selected automatically)
Eulerian Multiphase Model	**Multiphase Segregated Flow** **Multiphase Equation of State** (Selected automatically) **Gradients** (Selected automatically)
Viscous Regime	**Turbulent**
Optional Models	**Gravity** **Cell Quality Remediation**

2. Click **Close**.
3. To review the models, open the **Physics 1** > **Models** node.



4. Save the simulation.

Creating Phases and Selecting Phase Models

Create the phases and select the appropriate phase models.

The two phases that are used in this case are air and water. Air bubbles are injected into a flowing stream of water. The phases are defined by creating new phases under the Eulerian Multiphase model.

To create the phases and select the phase models:

Create the liquid phase.

1. In the **Physics 1** continuum, right-click the **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and create a new phase.
2. Rename the **Phase 1** node to **Water**.
3. For the **Water** phase, select the following models:

Group Box	Model
Enabled Models	**Flow** (Pre-selected) **Turbulent** (Pre-selected)
Material	**Liquid** **Reynolds-Averaged Navier-Stokes** (Selected automatically)

4. Deactivate **Auto-select recommended models**.
5. Select the following models in order:

Group Box	Model
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**
K-Epsilon Turbulence Models	**Standard K-Epsilon** **Exact Wall Distance** (Selected automatically)

6. Activate **Auto-select recommended models**.
7. Select the following models in order:

Group Box	Model
Enabled Models	**High y+ Wall Treatment** (Selected automatically)
Equation of State	**Constant Density**

8. Click **Close**.

Create the gas phase.

9. Create a second phase and rename it to **Air**.
10. For the **Air** phase, select the following models:

Group Box	Model
Enabled Models	**Flow** (Pre-selected) **Turbulent** (Pre-selected)

Group Box	Model
Material	**Gas** **Reynolds-Averaged Navier-Stokes** (Selected automatically)

11. Deactivate **Auto-select recommended models**.
12. Select the following models in order:

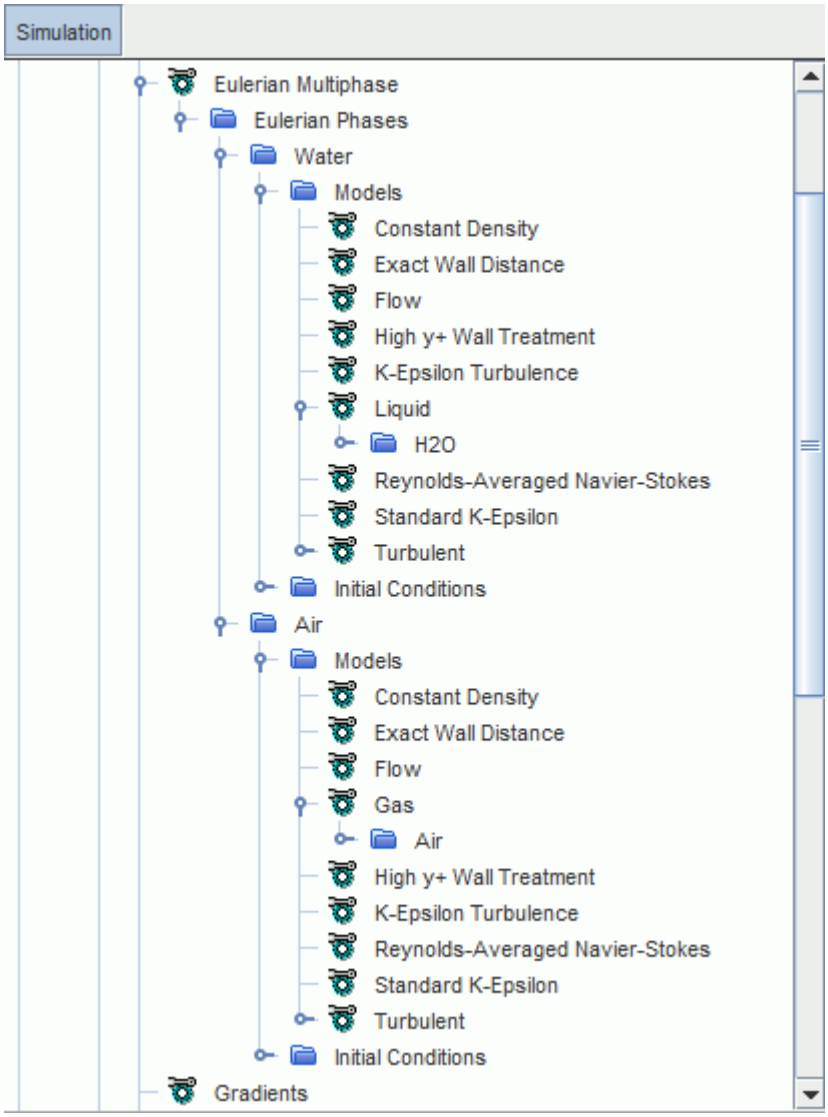
Group Box	Model
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**
K-Epsilon Turbulence Models	**Standard K-Epsilon** **Exact Wall Distance** (Selected automatically)

13. Activate **Auto-select recommended models**.
14. Select the following models in order:

Group Box	Model
Enabled Models	**High y+ Wall Treatment** (Selected automatically)
Equation of State	**Constant Density**

15. Click **Close**.

    The **Eulerian Phases** node appears as shown below.



16. Save the simulation.

Defining the Phase Interactions

Define the interactions between the phases.

Using the **Multiphase Interaction** model, you can define the interaction between the water and air phases. The Wang Curve Fit model is suited for modeling dilute concentrations of air bubbles in water. The air bubbles being modeled have a diameter of 1 mm.

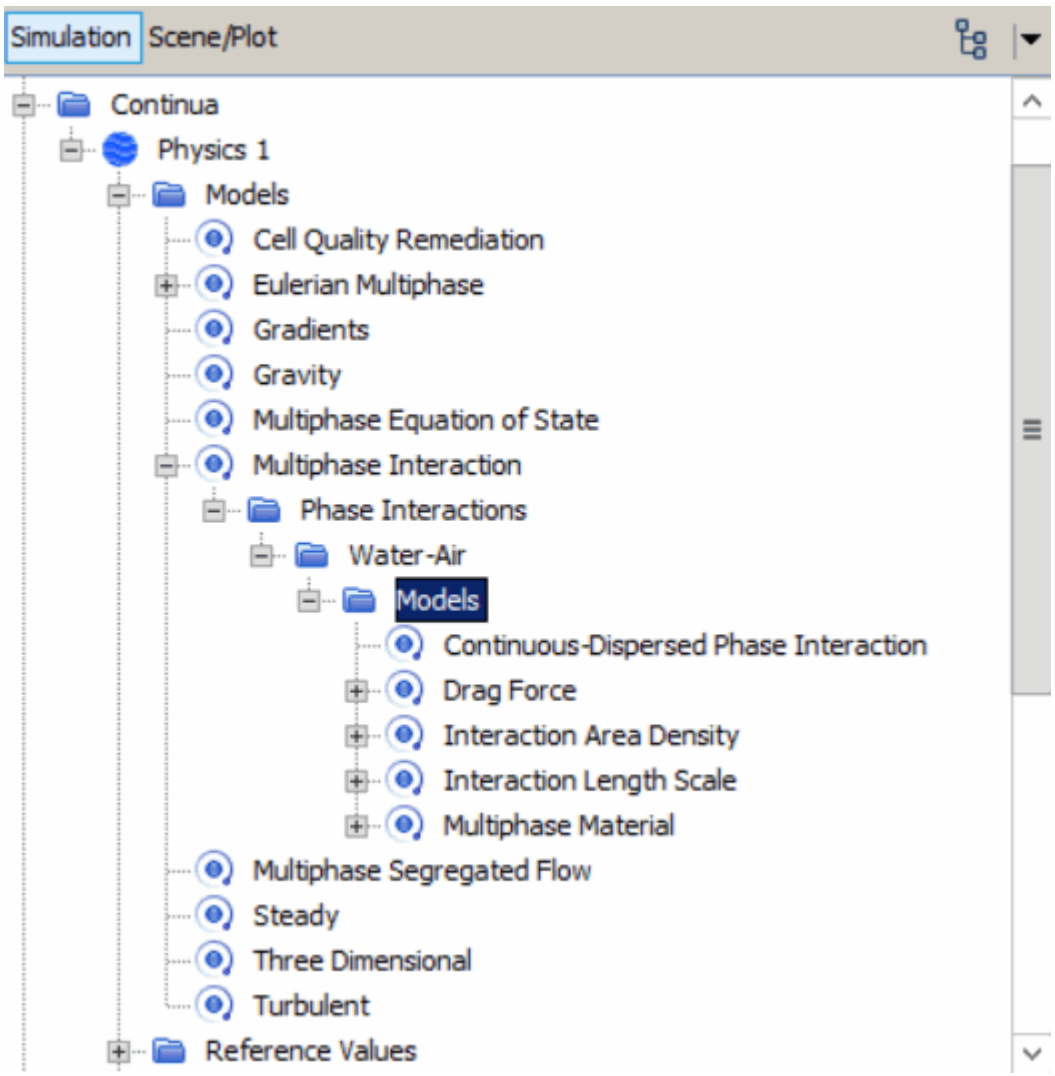
To define the phase interactions:

1. In the **Physics 1** continuum, right-click the **Models** > **Multiphase Interaction** > **Phase Interactions** node and create a new phase interaction.
2. Rename the **Phase Interaction 1** node to **Water-Air**.
3. For the **Water-Air** phase interaction, select the following models in order:

Group Box	Model
Phase Interaction Topology	**Continuous-Dispersed Phase Interaction**
Enabled Models	**Drag Force** (selected automatically) **Multiphase Material** (selected automatically) **Interaction Length Scale** (selected automatically) **Interaction Area Density** (selected automatically)

4. Click **Close**.

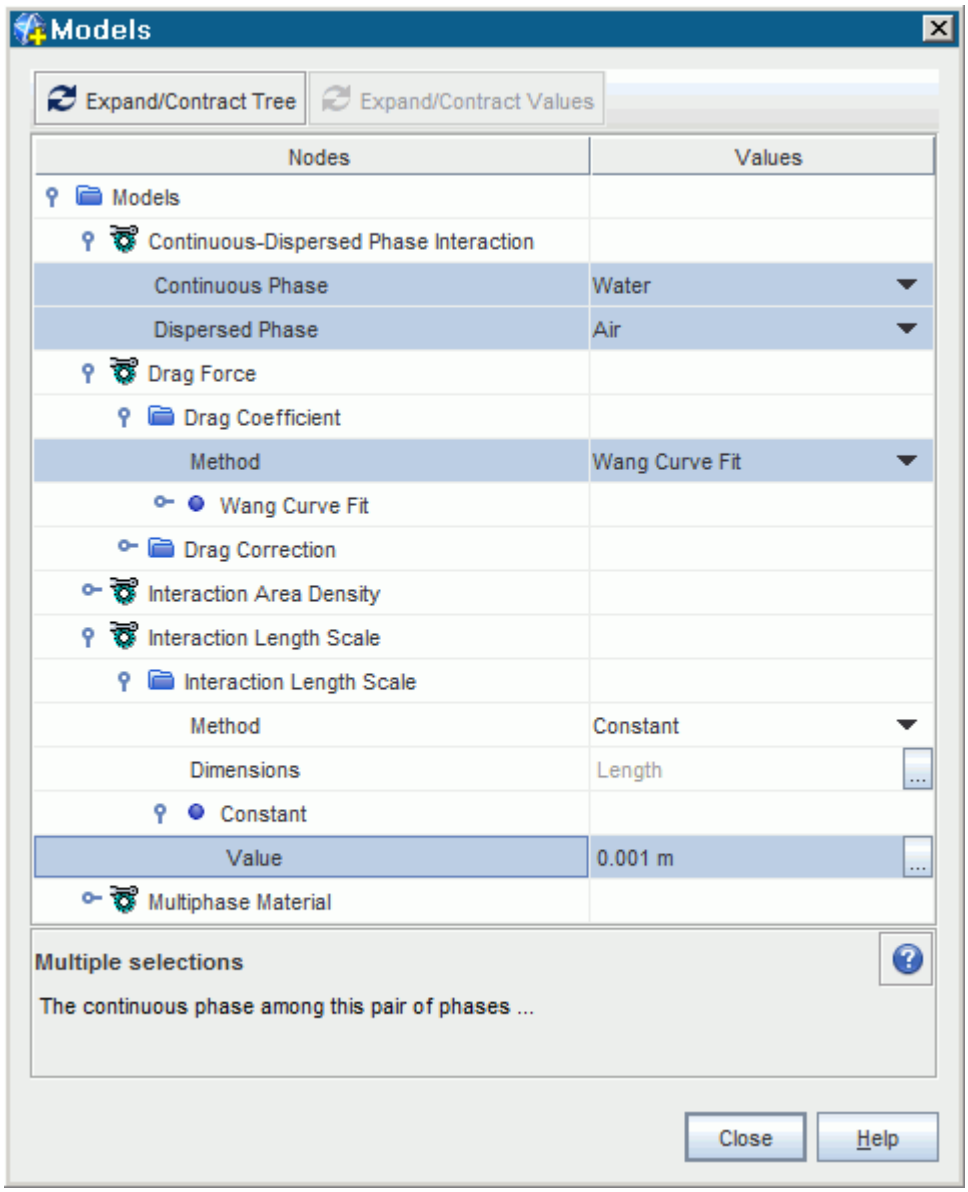
   The **Water-Air** > **Models** node is shown below:



5. Edit the **Water-Air** > **Models** node and set the following properties:

Node	Property	Setting
Interaction Length Scale > Interaction Length Scale > Constant	**Value**	`0.001 m`
Continuous-Dispersed Phase Interaction	**Continuous Phase**	**Water**
	Dispersed Phase	**Air**
Drag Force > Drag Coefficient	**Method**	**Wang Curve Fit**

The completed *Models* dialog is shown below.



6. Save the simulation.

Deactivating Secondary Gradients

Deactivate the secondary gradients option of the Multiphase Segregated Flow and Standard K-Epsilon models for each phase to help accelerate solution convergence.

In the following steps, you deactivate the secondary gradients option of the Multiphase Segregated Flow and Standard K-Epsilon models for each phase to help accelerate solution convergence.

To deactivate the secondary gradients:

Edit the Multiphase Segregated Flow model first.

1. Select the **Physics 1** > **Models** > **Multiphase Segregated Flow** node and set the *Secondary Gradients* to **Off**.

   The *Properties* window is shown below.

Edit the Multiphase Standard K-Epsilon model for each phase.

2. Select the **Physics 1** > **Models** > **Eulerian Multiphase** > **Eulerian Phases** > **Water** > **Models** > **Standard K-Epsilon** node and set the *Secondary Gradients* to **Off**.
3. Repeat for the *Secondary Gradients* property of the **Physics 1** > **Models** > **Eulerian Multiphase** > **Eulerian Phases** > **Air** > **Models** > **Standard K-Epsilon** node.

Setting Initial Conditions and Reference Values

Set the initial conditions and reference values for the **Physics 1** continuum.

To set the initial conditions and reference values:

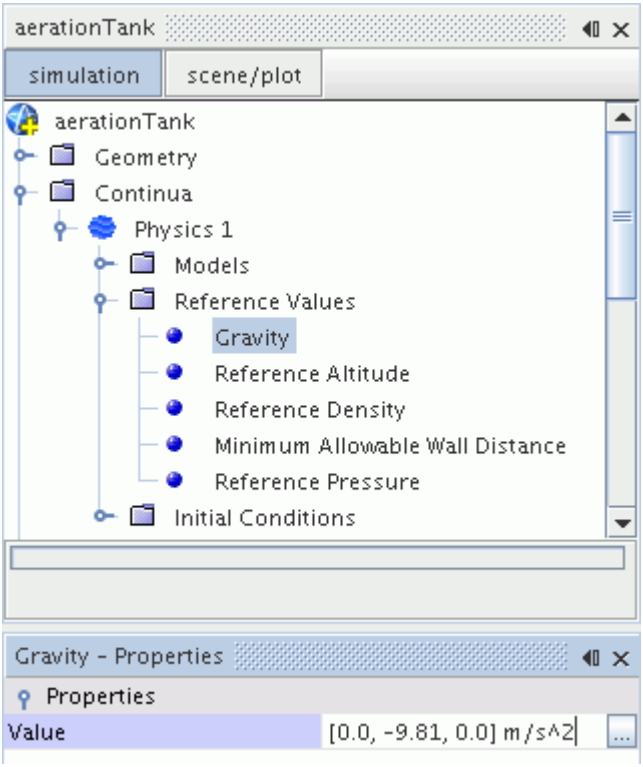
1. Edit the **Physics 1** > **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and set the following properties:

Node	Property	Setting
Water > **Initial Conditions**		
⌊ **Velocity**	**Value**	`[0.0, 0.0, 0.5] m/s`
⌊ **Volume Fraction**	**Value**	`1.0`
Air > **Initial Conditions**		
⌊ **Velocity**	**Value**	`[0.0, 0.0, 0.5] m/s`
⌊ **Volume Fraction**	**Value**	`0.0`

Set gravity to act in the -y direction. It is common practice to use the density of the continuous phase as the reference density.

2. Edit the **Reference Values** node and set the following property:

Node	Property	Setting
Gravity	**Value**	`[0.0, -9.81, 0.0] m/s^2`
Reference Density	**Value**	`997.561 kg/m^3`



3. Save the simulation.

Setting Boundary Conditions

Define the boundary types and specify the appropriate property values.

The boundary types have been defined before mesh generation.

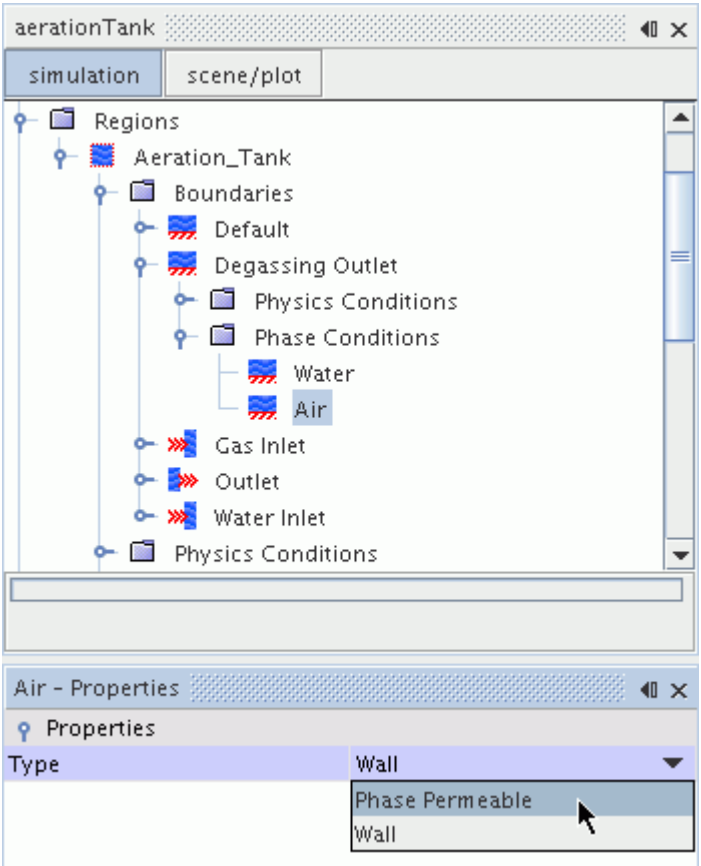
You begin by setting the phase-permeable **Degassing Outlet** boundary so that only the air bubbles can escape as they rise to the top of the aeration tank. STAR-CCM+ ensures that the total gas inflow is equal to the total gas outflow and computes the internal gas volume fraction according to continuity.

To set the boundary conditions:

Allow gas to escape through the **Degassing Outlet** boundary.

1. Edit the **Regions** > **Aeration_Tank** > **Boundaries** > **Degassing Outlet** node and set the following property:

Node	Property	Setting
Phase Conditions > **Air**	**Type**	**Phase Permeable**



The top boundary of the aeration tank now acts as an outlet for the air phase. The water phase treats this boundary as a wall.

Make sure that this wall does not affect the flow of the water.

2. Edit the **Degassing Outlet** node and set the following property:

Node	Property	Setting
Phase Conditions > **Water** > **Physics Conditions** > **Shear Stress Specification**	**Method**	**Slip**

Define the **Gas Inlet** boundary. The velocity of each phase at the gas injector boundary is specified under the **Velocity Magnitude** node. For air bubbles of 1-mm diameter, the terminal rise velocity in water is about 0.25 m/s. You can use the terminal velocity at the inlets as bubbles in most bubbly flows reach their terminal velocity within a short distance from the injector.

Set the volume fraction of the gas injectors. This inlet is essentially a perforated plate where 10% of the area corresponds to the air nozzle holes.

3. Edit the **Gas Inlet** > **Phase Conditions** node and set the following properties:

Node	Property	Setting
Water > **Physics Values**		
⌊ **Velocity Magnitude**	**Value**	`0.0 m/s`
⌊ **Volume Fraction**	**Value**	`0.9`
Air > **Physics Values**		
⌊ **Velocity Magnitude**	**Value**	`0.25 m/s`
⌊ **Volume Fraction**	**Value**	`0.1`

Define the **Water Inlet** boundary. This inlet boundary introduces water into the aeration tank at a steady rate.

Set the velocity of the air phase to match the water velocity and set the air phase volume fraction to 0.0.

4. Edit the **Water Inlet** > **Phase Conditions** node and set the following properties:

Node	Property	Setting
Water > **Physics Values**		
⌊ **Velocity Magnitude**	**Value**	`0.5 m/s`
⌊ **Volume Fraction**	**Value**	`1.0`
Air > **Physics Values**		
⌊ **Velocity Magnitude**	**Value**	`0.5 m/s`
⌊ **Volume Fraction**	**Value**	`0.0`

5. Save the simulation.

Setting Solver Parameters and Stopping Criteria

Set the appropriate solver parameters and stopping criteria for the simulation.

To set the solver parameters and stopping criteria:

1. Edit the **Solvers** node and set the following properties:

Node	Property	Setting
Multiphase Segregated Flow > **Phase Coupled Velocity**	**Implicit Under-Relaxation Factor**	`0.3`
Multiphase Segregated Flow > **Pressure**	**Under-Relaxation Factor**	`0.3`
Volume Fraction	**Implicit Under-Relaxation Factor**	`0.2`
K-Epsilon Turbulence	**Under-Relaxation Factor**	`0.5`

Node	Property	Setting
K-Epsilon Turbulent Viscosity	**Under-Relaxation Factor**	`0.5`

2. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Steps	**Enabled**	Activated
	Maximum Steps	`700`

3. Save the simulation.

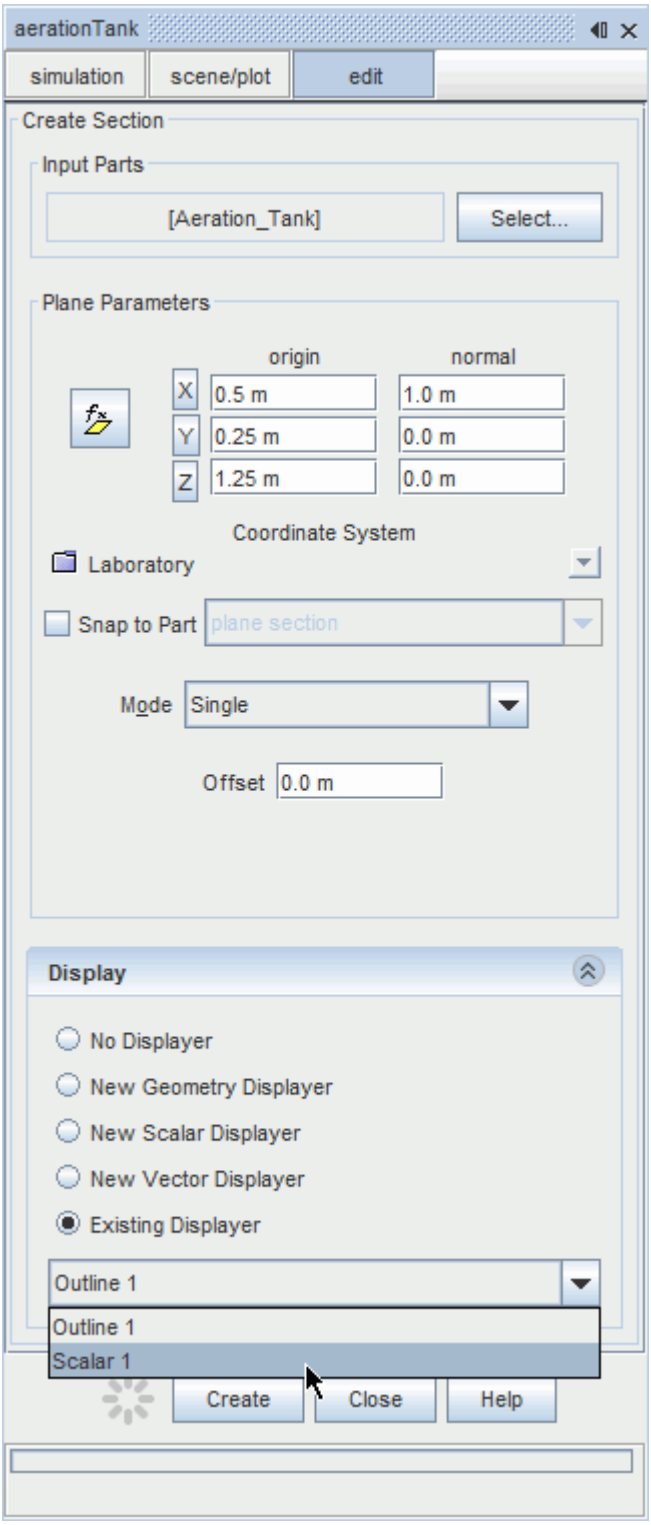
Visualizing the Solution

Create two scalar scenes to display the results of the simulation.

To help you visualize the solution, create two scalar scenes. One displays the volume fraction of air through a central section plane, and the other displays the mass flow rate of the air phase on the degassing boundary. Create the section plane that cuts through the center of the geometry, including the two gas injectors. Then display the volume fraction of air onto this plane.

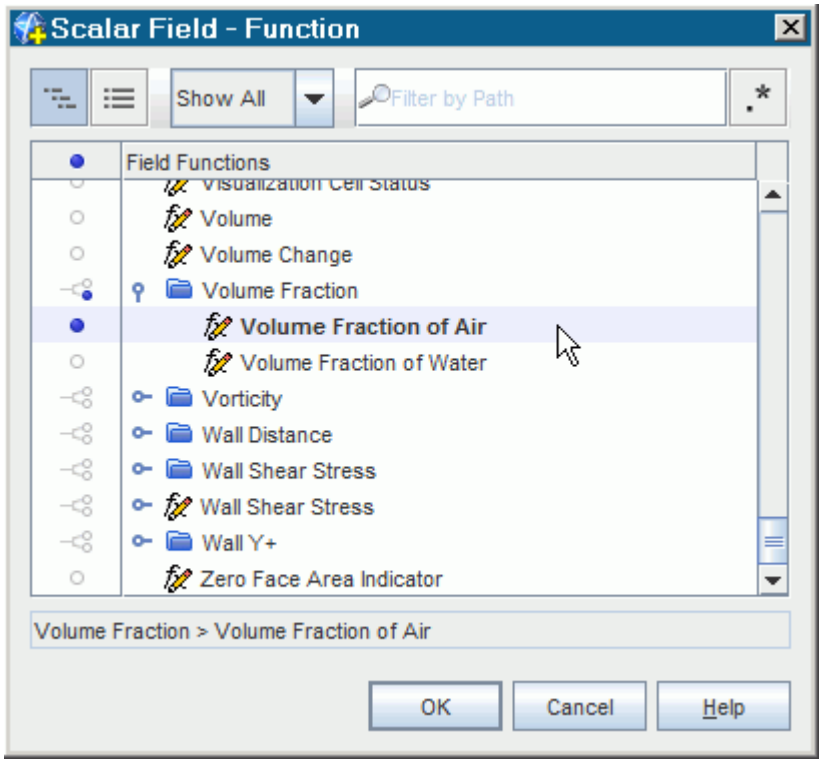
To visualize the solution:

1. Click (**Create/Open Scenes**) and select **Scalar**.
2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane**.
3. In the *Create Section* dialog that appears, select **Existing Displayer** > **Scalar 1** from the *Display* box.

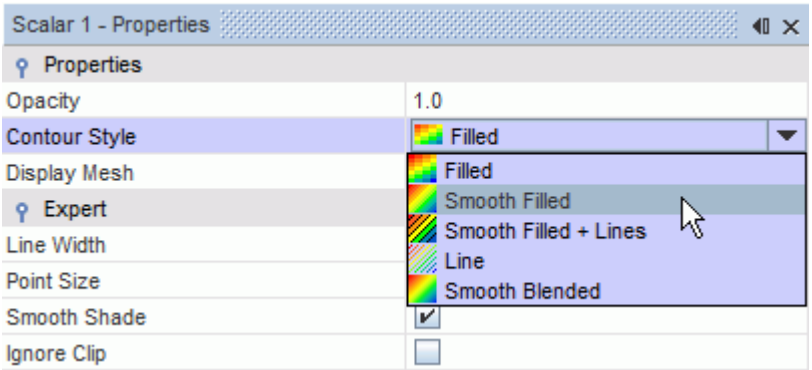


4. In the *Plane Parameters* group box, ensure that the **origin** is `0.5, 0.25, 1.25` and that the **normal** is `1.0, 0.0, 0.0`.
5. Leave the remaining settings at their default values and click **Create** and then **Close**.
6. Click the **scene/plot** button to show the object tree for the scalar scene.

7. Select the **Displayers** > **Scalar 1** > **Scalar Field** node and set the *Function* to **Volume Fraction** > **Volume Fraction of Air**.



8. Select the **Displayers** > **Scalar 1** node and set the *Contour Style* to **Smooth Filled**.

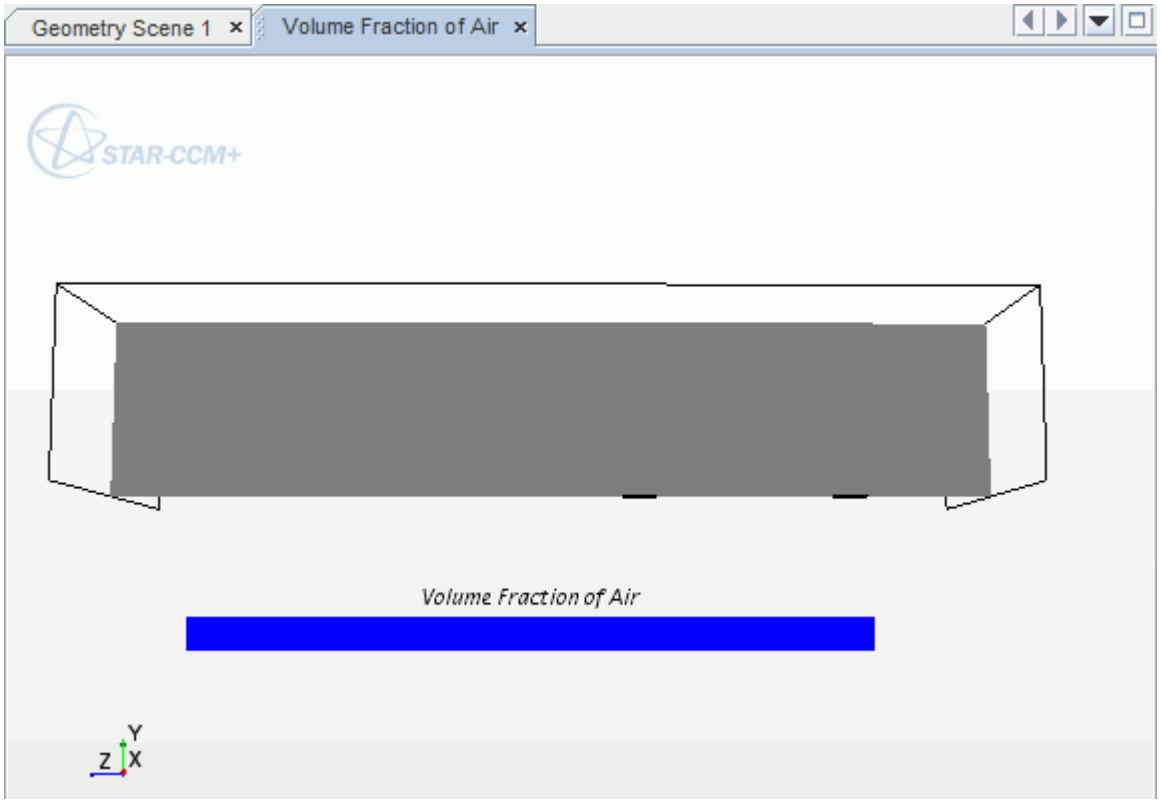


9. Click (**Save-Restore-Select views**) and select **Views** > **+X** > **Up +Y**.

   This action switches the view to the side of the tank.

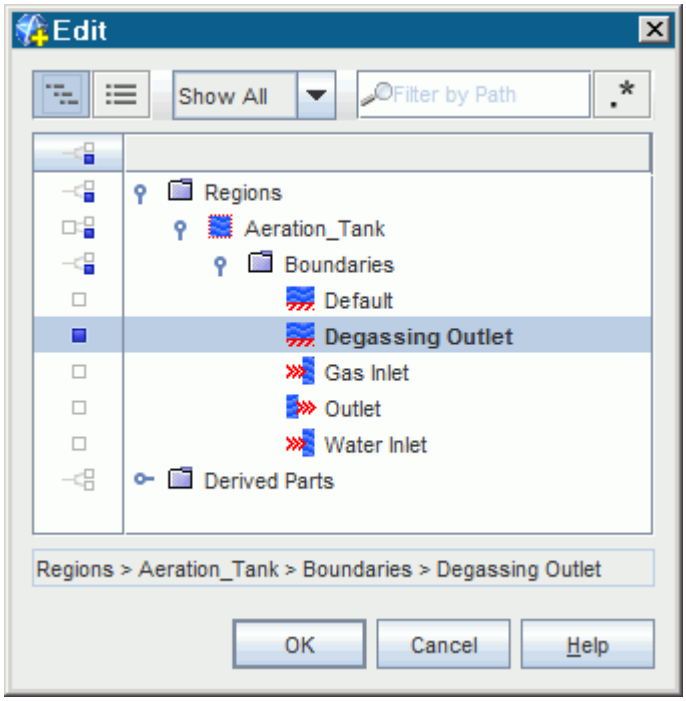
10. Select the **Scalar Scene 1** node at the top of the object tree and rename it to `Volume Fraction of Air`.

    The scene is shown below.



Create the second scalar scene that displays the mass flow rate of air through the degassing boundary.

11. Click (**Create/Open Scenes**) and select **Scalar**.
12. Right-click the **Displayers** > **Scalar 1** > **Parts** node and select **Edit**.
13. In the *Edit* dialog, select the **Regions** > **Aeration_Tank** > **Degassing Outlet** node, as shown below.



14. Click **OK**.
15. Select the **Scalar 1** > **Scalar Field** node and set the *Function* to **Mass Flow Rate** > **Air**.

    This action displays the mass flow of air across the degassing boundary. The mass per second through each boundary face is displayed, not the mass flow per unit area per second.
16. Select the **Displayers** > **Scalar 1** node and set the *Contour Style* to **Smooth Filled**.
17. Click (**Save-Restore-Select views**) and select **Views** > **+Y** > **Up -X**.

    This switches to the top view of the tank.
18. Select the **Scalar Scene 1** node at the top of the object tree and rename it to `Mass Flow Rate of Air`.
19. Save the simulation.

Running the Simulation

Preparation of the simulation is now complete, and the simulation can be run.

To run the simulation:

1. Click (**Run**) in the *Solution* toolbar.

   The solution progress is displayed in the *Output* window. The *Residuals* display is created automatically in the *Graphics* window and shows the progress of the solvers.

   While the simulation is running, you can click the tabs at the top of the *Graphics* window to view scenes and plots.

During the run, you can stop the process by clicking (**Stop**) on the toolbar. If you do stop the simulation, you can click (**Run**) to resume it.

If left alone, the simulation continues until 700 iterations are complete. When the last iteration is complete, the following message is displayed in the *Output* window:

```
Stopping criterion Maximum Steps satisfied.
```

2. When the simulation has finished running, save it.

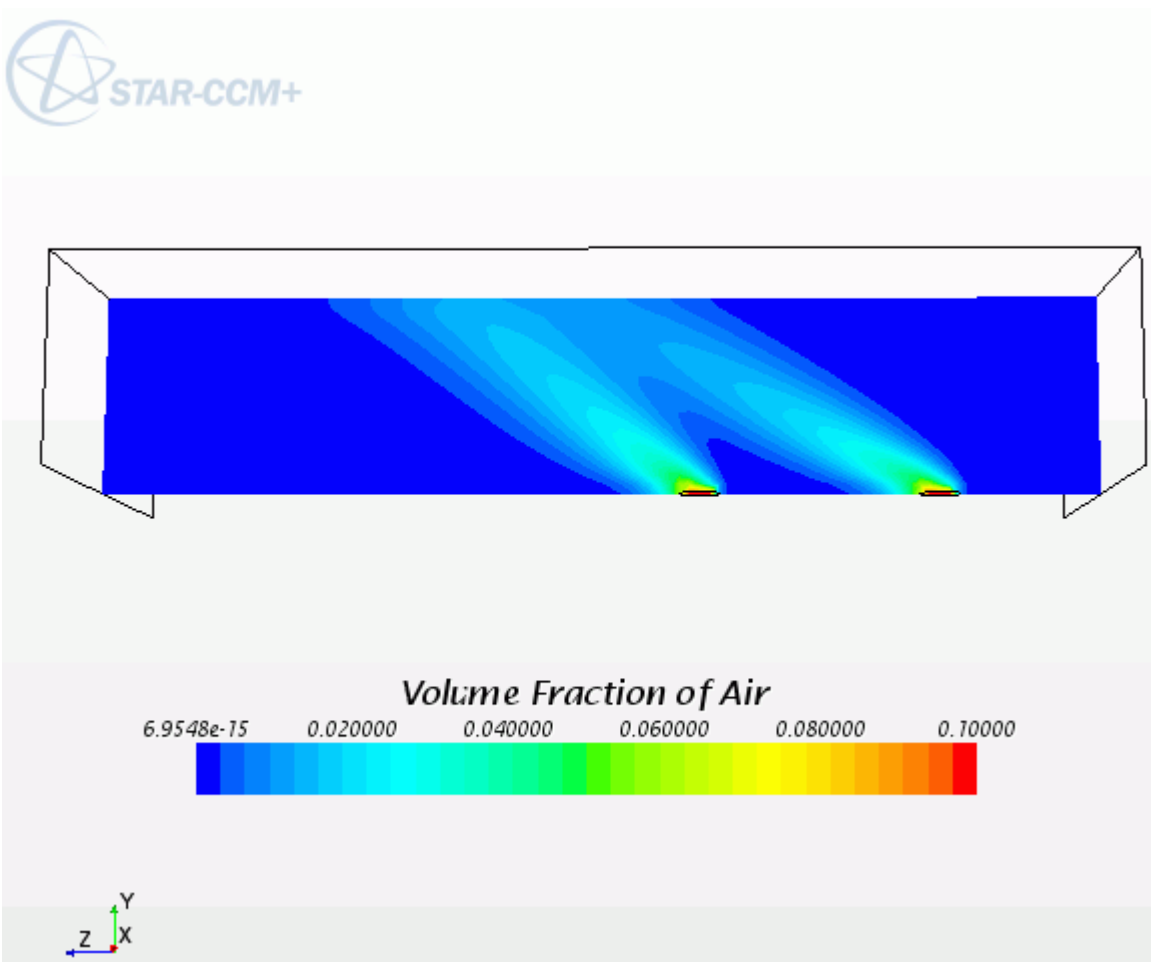
Visualizing the Results

View the results of the simulation using the scalar scenes that you prepared earlier.

The solution can be visualized by displaying each of the scalar scenes you prepared earlier.

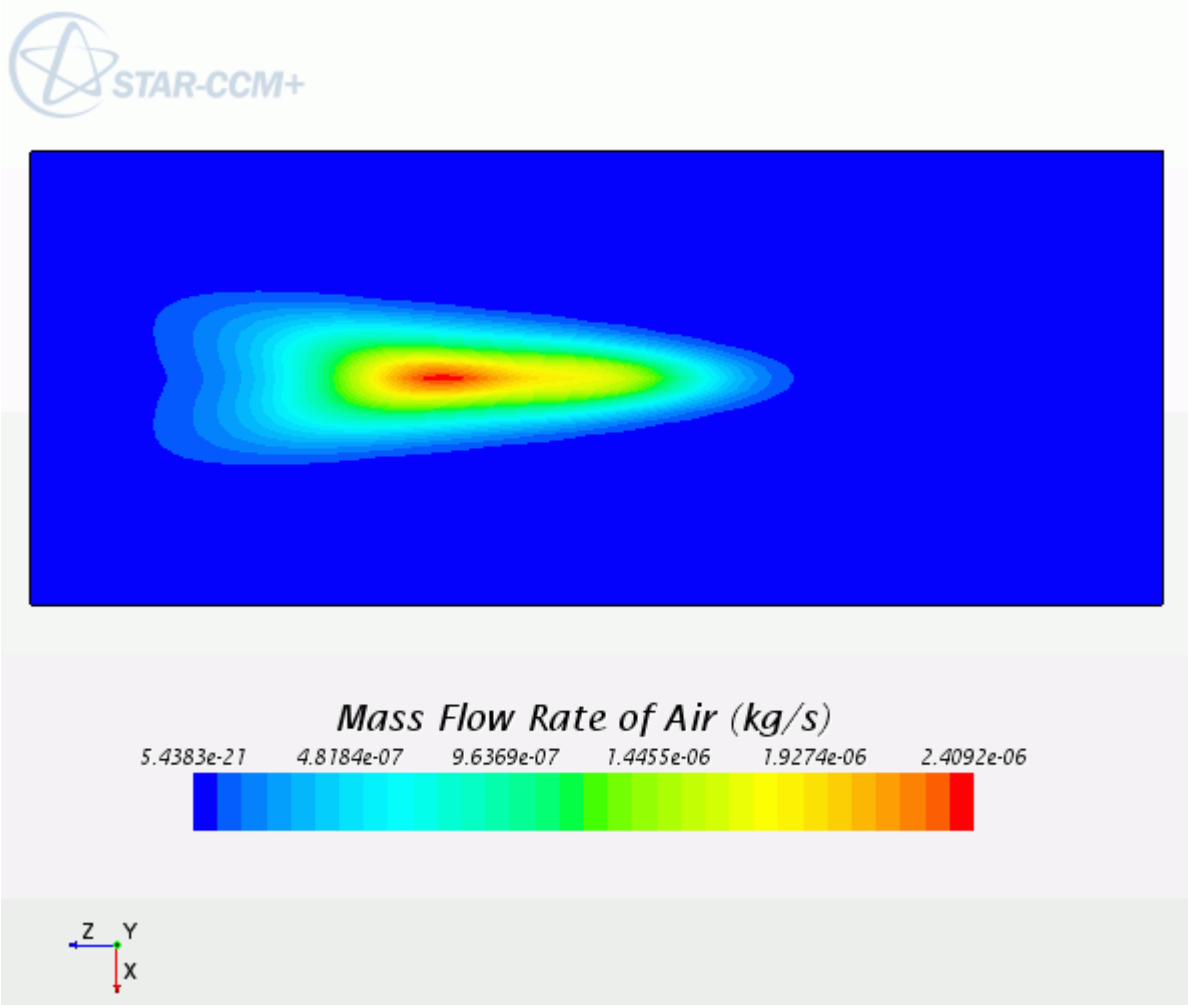
To visualize the results:

1. Select the *Volume Fraction of Air* tab at the top of the *Graphics* window.



The air bubbles are being carried downstream as they rise towards the degassing outlet, where they then escape through the phase-permeable wall. The water is unaffected by this boundary (due to the slip condition you applied) and is kept inside the tank.

2. Similarly, display the *Mass Flow Rate of Air* scene by selecting the corresponding tab.



This scene displays the mass flow rate of the escaping air bubbles through the phase-permeable wall. The faster moving bubbles are concentrated near the center of the bubble plume.

To make sure that no water is passing through the degassing outlet, you can do the following:

3. Right-click the color bar in the scene window and change the scalar field to display Mass Flow Rate of Water.

   No water flows through this boundary.

Summary

This tutorial demonstrated how to model a Eulerian multiphase turbulent flow with a degassing boundary in STAR-CCM+.

This tutorial has introduced the following STAR-CCM+ features:

- Setting up the Eulerian Multiphase model.
- Creating phases and selecting phase models.
- Define phase interaction.
- Setting reference values and initial conditions.
- Setting up boundary conditions, including a phase-permeable wall boundary to allow the selected phase to exit through it.
- Running the simulation.
- Visualizing the results.

Eulerian: Wall Boiling

This tutorial validates the wall boiling model in STAR-CCM+ against experimental void and temperature data for high-pressure forced convective boiling.

The 1967 paper of Bartolomei [800] provides 13 cases. The first case at 45 *bar* is of particular interest as it provides wall superheat data, and is the focus of this tutorial. Although the agreement between prediction and experiment is adequate rather than good, the same models can be used to obtain reasonable predictions for void in many of the further 29 cases spanning 30-150 *bar* given in the 1982 paper [801].

The geometry that is used is a Bartolomei boiling tube, where subcooled water flows upward through a 2 meter tall vertical pipe. The pressure at the outlet is 45 bar. Heat is applied uniformly at the pipe wall.

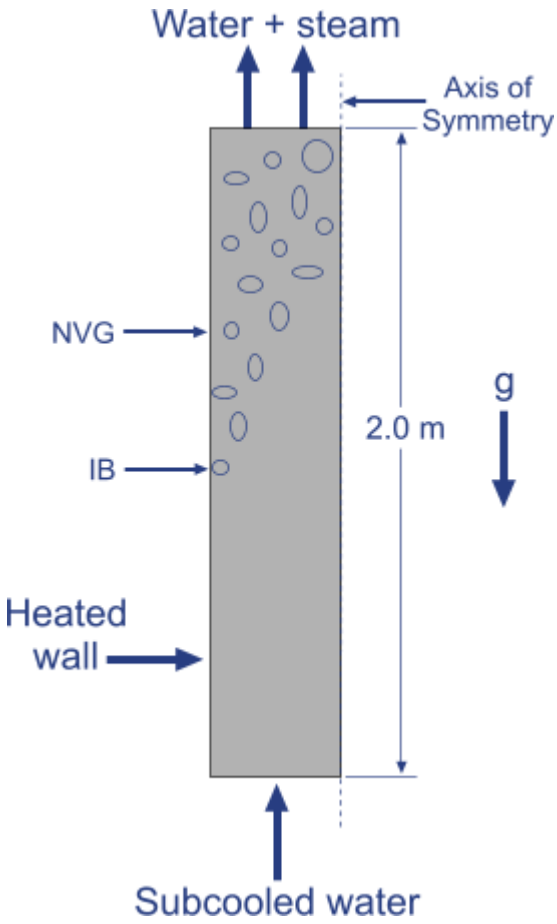
The basic geometry is a tube of 15.4-mm diameter and heated length of 2 m. To allow some flow development between the inlet and the heated section, a short unheated leader has been added. Similarly so that the outlet conditions in this developing flow do not influence the end of the heated section, a short trailer has also been added.

The conditions are defined as follows:

Length	*L*	2 m
Equivalent Diameter	*D*	0.0154 m
Heat Flux	*q*	570000 W/m^2
Mass Flux	*G*	900 kg/m^2
Inlet Temperature	T^{IN}	470.63 K
System Pressure	P^{SYS}	45.0 bar

Boiling starts when the wall temperature exceeds the water saturation temperature. This point is called the point of Inception of Boiling (IB) and small bubbles start to appear near the wall. When the degree of subcooling decreases, downstream of the point of IB, the volume fraction of the steam begins to grow. This point in the tube is called the point of Net Vapor Generation (NVG). Further downstream from the NVG point, the volume fraction of the steam continues to grow at an increasing rate and the bubbles start to detach from the wall and move into the liquid. As this detaching occurs, some of the bubbles condense, enhancing the heat transfer to the water. The mixture of steam and water exits at the top of the tube.

An axisymmetric, two-dimensional volume mesh of the Bartolomei boiling tube is provided, as shown below.



Contents:

Prerequisites
Importing the Mesh and Naming the Simulation
Setting Up User Field Functions
Selecting the Physics Models
Creating Phases and Selecting Phase Models
Defining the Phase Interactions
Defining the Phase Interaction Properties
Setting the Material Properties
Setting Reference Values
Setting Boundary Conditions
Setting Initial Conditions
Setting Solver Parameters and Stopping Criteria
Setting Up Monitoring Plots
Setting Up Analysis Plots
Visualizing the Solution
Running the Simulation
Final Mass Balance

Visualizing the Results
Comparison With Experiment
Turbulence Checks
Additional Information
Summary
Bibliography

Prerequisites

The instructions in the Eulerian: Wall Boiling tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Input Files

The input files are in the `doc/startutorialsdata/multiphaseFlow/data` subdirectory of your STAR-CCM+ installation directory. It is recommended that you copy the following input files to your working directory before you start this tutorial:

- `wallBoilingTutorialGrid.ccm`
- `wb_setupFieldFunctions.java`
- `wb_createAxialPlots.java`
- `wb_deleteAxialPlots.java`
- `wb_setupAnalysisPlots.java`
- `wb_setupMonitorPlots.java`
- `Bart1967_Figure3_AxisTemperature.csv`
- `Bart1967_Figure3_CalorimetricTemperature.csv`
- `Bart1967_Figure3_TrueVoid.csv`
- `Bart1967_Figure3_WallTemperature.csv`

Importing the Mesh and Naming the Simulation

Launch STAR-CCM+, import the supplied geometry, and save the simulation.

The basic geometry is a tube of 15.4-mm diameter and heated length of 2 m. To allow some flow development between the inlet and the heated section, a short unheated leader has been added. Similarly so that the outlet conditions in this developing flow do not influence the end of the heated section, a short trailer has also been added. These additions are helpful and reasonable computational practices when you consider the drawings of the original rig.

The overall grid resolution is 115 cells in the axial direction and 20 cells in the radial direction between the axis and the tube wall. Both the grid and the grid cells have a high aspect ratio, and it is helpful later to set up a simple transform to view the mesh or the results.

To import the mesh and name the simulation:

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh**.
4. In the *Open* dialog, choose `doc/startutorialsdata/multiphaseFlow/data/wallBoilingTutorialGrid.ccm`, then click **Open**.

   STAR-CCM+ provides feedback on the import process, which takes a few seconds, in the *Output* window.
5. Save the simulation as `wallBoilingTutorial.sim`.

Setting Up User Field Functions

Create the field functions that are required for this tutorial.

This tutorial makes extensive use of field functions to define case parameters, to set up boundary conditions, and to analyze and present results.

For convenience, these field functions can be set up by running a macro before you set up any models.

To create the user field functions:

1. Select **File** > **Macro** > **Play Macro**.
2. In the *Open* dialog, navigate to your working directory and select `wb_setupFieldFunctions.java`.
3. Click the **Open** button to run the macro.

The field functions are tabulated below:

Case Parameter Field Functions

EquivalentDiameter	
Type	**Scalar**
Definition	`0.0154`
Ignore Boundary Values	**Deactivated**

HeatedLength	
Type	**Scalar**
Definition	`2.0`
Ignore Boundary Values	**Deactivated**

SystemPressure	
Type	**Scalar**

SystemPressure	
Definition	4500000.0
Ignore Boundary Values	**Deactivated**

SaturationTemperature	
Type	**Scalar**
Definition	530.59
Ignore Boundary Values	**Deactivated**

InletDensity	
Type	**Scalar**
Definition	832.06
Ignore Boundary Values	**Deactivated**

InletMassFlux	
Type	**Scalar**
Definition	900.0
Ignore Boundary Values	**Deactivated**

InletTemperature	
Type	**Scalar**
Definition	470.63
Ignore Boundary Values	**Deactivated**

InletVelocity	
Type	**Vector**
Definition	[$InletMassFlux/$InletDensity, 0, 0]
Ignore Boundary Values	**Deactivated**

WallHeatFlux	
Type	**Scalar**
Definition	570000.0
Ignore Boundary Values	**Deactivated**

Model Field Functions

IterationsToFullPower	
Type	**Scalar**
Definition	`200.0`
Ignore Boundary Values	**Deactivated**

WallHeatFluxDistribution	
Type	**Scalar**
Definition	`($$Centroid[0] > 0 && $$Centroid[0] < $HeatedLength) ? $WallHeatFlux : 0.0`
Ignore Boundary Values	**Deactivated**

WallHeatFluxIterationRamp	
Type	**Scalar**
Definition	`( $Iteration < $IterationsToFullPower ? ($WallHeatFluxDistribution * $Iteration )/ $IterationsToFullPower : $WallHeatFluxDistribution )`
Ignore Boundary Values	**Deactivated**

dMax	
Type	**Scalar**
Definition	`0.012`
Ignore Boundary Values	**Deactivated**

dMin	
Type	**Scalar**
Definition	`1.0E-6`
Ignore Boundary Values	**Deactivated**

ClippedSauterMeanDiameter	
Type	**Scalar**
Definition	`( $SauterMeanDiameterVapor < $dMin ? $dMin : ( $SauterMeanDiameterVapor >`

ClippedSauterMeanDiameter	
	`$dMax ? $dMax : $SauterMeanDiameterVapor ) )`
Ignore Boundary Values	**Deactivated**

Analysis Field Functions

InletThermodynamicQuality	
Type	**Scalar**
Definition	`$SpecificHeatLiquid * ($InletTemperature - $SaturationTemperature)/ ${LatentHeatLiquid Vapor Interface}`
Ignore Boundary Values	**Deactivated**

ThermodynamicQuality	
Type	**Scalar**
Definition	`$InletThermodynamicQuality+ 4 * $WallHeatFlux * $$Centroid[0]/ ( $InletMassFlux * ${LatentHeatLiquid Vapor Interface} * $EquivalentDiameter )`
Ignore Boundary Values	**Deactivated**

LiquidSubcooling	
Type	**Scalar**
Definition	`${TemperatureLiquid Vapor Interface} - ${TemperatureLiquid}`
Ignore Boundary Values	**Activated**

WallSuperHeat	
Type	**Scalar**
Definition	`${TemperatureLiquid} - ${TemperatureLiquid Vapor Interface}`
Ignore Boundary Values	**Deactivated**

QuenchingHeatFlux	
Type	**Scalar**

QuenchingHeatFlux	
Definition	`(1-${WallDryoutAreaFractionLiquid Vapor Interface}) * ${BubbleInducedQuenchingHeatTransfer CoefficientLiquid Vapor Interface} * ${BubbleInducedQuenchingTemperatureF actorLiquid Vapor Interface} * ( $WallSuperHeat + $LiquidSubcooling )`
Ignore Boundary Values	**Deactivated**

WallEvaporationRate	
Type	**Scalar**
Definition	`-${WallInterphaseMassFluxLiquid Vapor Interface}`
Ignore Boundary Values	**Deactivated**

BulkCondensationRate	
Type	**Scalar**
Definition	`( ${InterphaseMassTransferRateLiquid Vapor Interface} > 0 ? ${InterphaseMassTransferRateLiquid Vapor Interface} : 0 )`
Ignore Boundary Values	**Deactivated**

BulkEvaporationRate	
Type	**Scalar**
Definition	`( ${InterphaseMassTransferRateLiquid Vapor Interface} < 0 ? -${InterphaseMassTransferRateLiquid Vapor Interface} : 0 )`
Ignore Boundary Values	**Deactivated**

Selecting the Physics Models

Physics models define the primary variables of the simulation, including pressure, temperature, velocity, and what mathematical formulation is used to generate the solution.

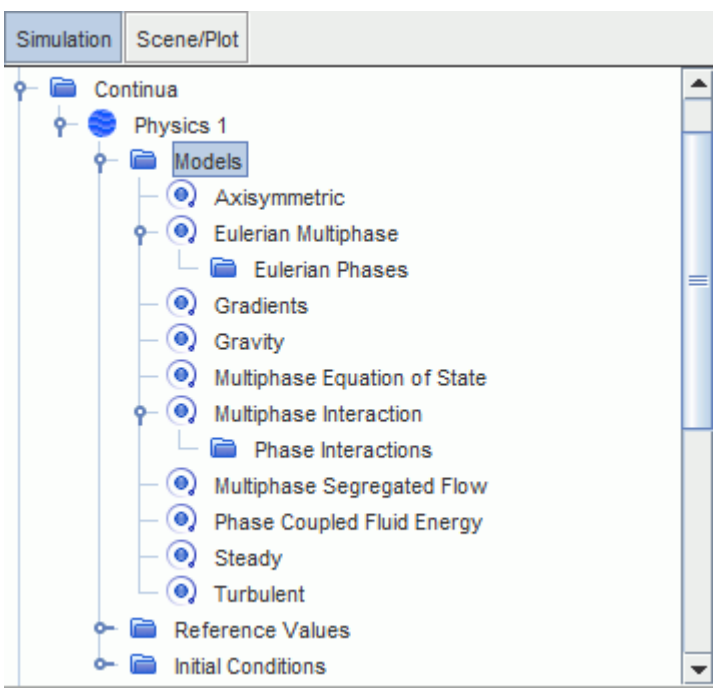
The multiphase turbulence models are available for selection once the multiphase flow and material models have been selected.

To select the physics models:

1. For the physics continuum, **Continua** > **Physics 1**, in the *Enabled Models* group box, deactivate **Two-Dimensional**.
2. Select the following models in order:

Group Box	Model
Space	**Axisymmetric**
Time	**Steady**
Material	**Eulerian Multiphase** **Multiphase Interaction** (Selected automatically)
Eulerian Multiphase Model	**Multiphase Segregated Flow** **Multiphase Equation of State** (Selected automatically) **Gradients** (Selected automatically)
Viscous Regime	**Turbulent**
Optional Models	**Phase Coupled Fluid Energy** **Gravity**

3. Click **Close**.
4. To review the models, open the **Physics 1** > **Models** node.



5. Save the simulation.

Creating Phases and Selecting Phase Models

Create the phases and select the appropriate phase models.

The two phases, Liquid and Vapor, that are flowing through the section of vertical pipe represent water and steam. These phases are defined by creating phases under the Eulerian physics model, but material properties are set later once all other models have been selected.

The main modeling choices that are made at this stage are:

- A K-Epsilon model with a two-layer wall treatment is selected for the continuous phase so that it can be used under a wide range of flow conditions.
- The Turbulence Response model is an appropriate model for the velocity fluctuations that are experienced in the dispersed phase for this bubbly flow.
- The S-gamma model is activated for the disperse phase so that it can track bubble size starting from bubble creation at the boiling wall, and including the effects of breakup, coalescence, and (most importantly) condensation.

To create the phases and select the phase models:

Create the liquid phase.

1. In the **Physics 1** continuum, right-click the **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and create a phase.
2. Rename the **Phase 1** node to **Liquid**.
3. For the **Liquid** phase, select the following models:

Group Box	Model
Enabled Models	**Flow** (Pre-selected) **Turbulent** (Pre-selected)
Material	**Liquid** **Reynolds-Averaged Navier-Stokes** (Selected automatically)
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence** **Realizable K-Epsilon Two-Layer** (Selected automatically) **Exact Wall Distance** (Selected automatically) **Two-Layer All y+ Wall Treatment** (Selected automatically)
Equation of State	**Constant Density**
Energy	**Segregated Fluid Temperature**

4. Click **Close**.

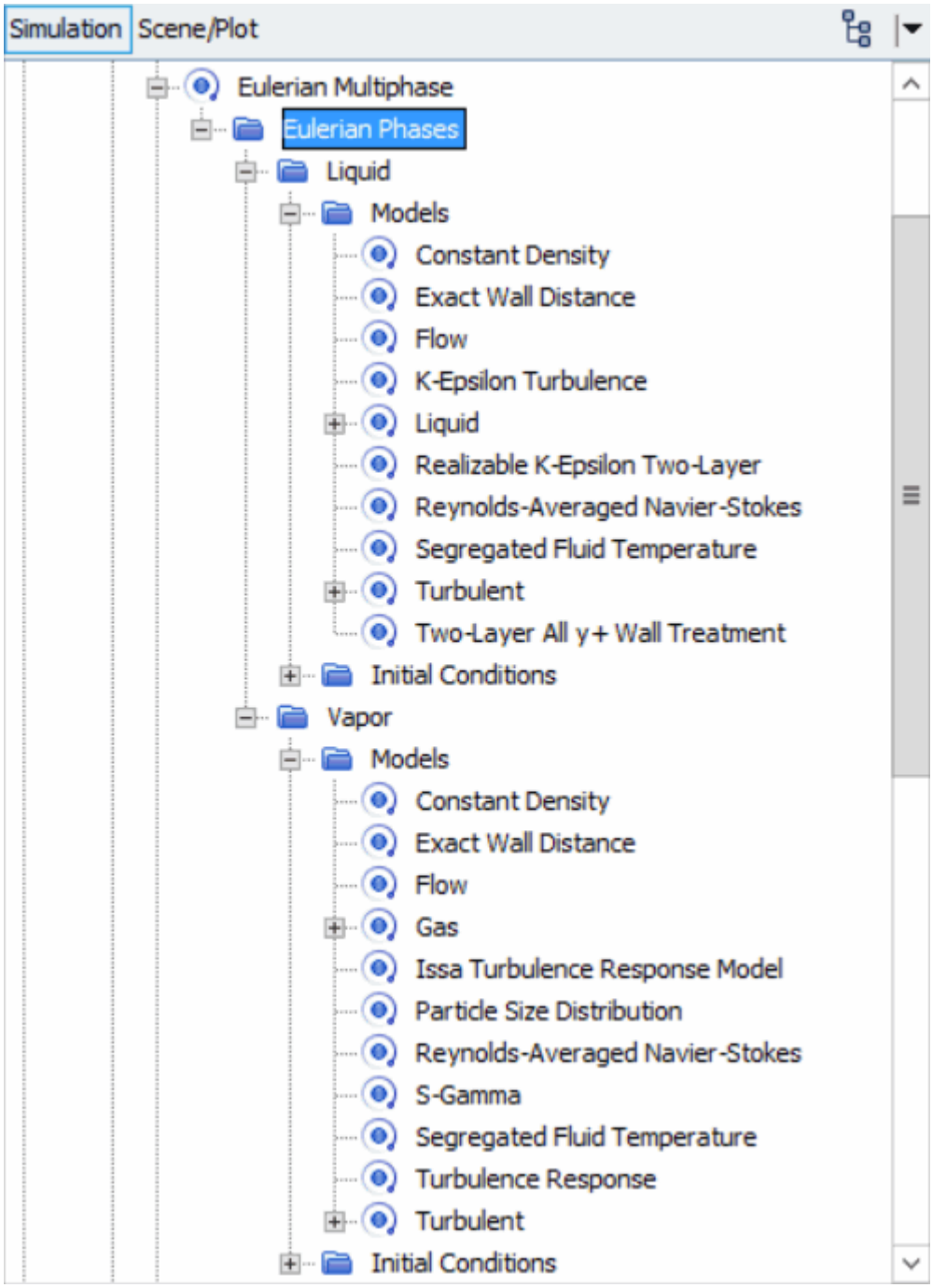
Create the gas phase.

5. Create a second phase and rename it to **Vapor**.
6. For the **Vapor** phase, select the following models:

Group Box	Model
Enabled Models	**Flow** (Pre-selected) **Turbulent** (Pre-selected)
Material	**Gas** **Reynolds-Averaged Navier-Stokes** (Selected automatically)
Reynolds-Averaged Turbulence	**Turbulence Response**
Turbulence Response Models	**Issa Turbulence Response Model** **Exact Wall Distance** (Selected automatically)
Equation of State	**Constant Density**
Energy	**Segregated Fluid Temperature**
Optional Models	**Particle Size Distribution**
Particle Size Distribution	**S-Gamma**

7. Click **Close**.
8. Rename the **Vapor** > **Models** > **Gas** > **Air** node to `Steam`.

   The **Eulerian Phases** node appears as shown.



9. Save the simulation.

Defining the Phase Interactions

Define the interactions between the phases.

Using the **Multiphase Interaction** model, you can define the interaction between the liquid and vapor phases.

The main modeling choice that is made here is the use of the combination of Hibiki-Ishii Nucleation Site Number Density and Kocamustafaogullari Bubble Departure Diameter. This combination of models is

applicable for water boiling under a wide range of pressures. The default models (Lemmert-Chawla and Tolubinsky) are much simpler, but can need testing and recalibration for use at different pressures.

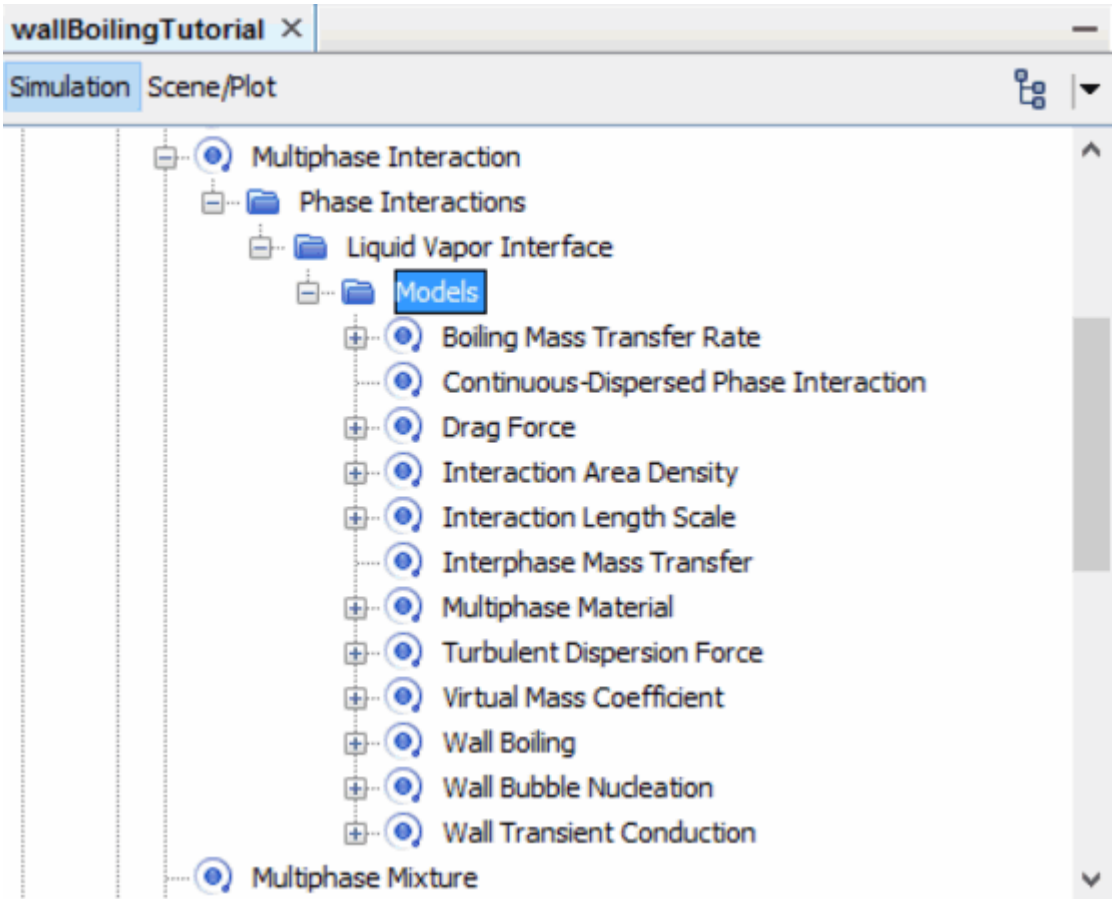
To define the phase interactions:

1. In the **Physics 1** continuum, right-click the **Models** > **Multiphase Interaction** > **Phase Interactions** node and create a phase interaction.
2. Rename the **Phase Interaction 1** node to **Liquid Vapor Interface**.
3. For the **Liquid Vapor Interface** phase interaction, deactivate **Auto-select recommended models**, and then select the following models in order:

Group Box	Model
Phase Interaction Topology	**Continuous-Dispersed Phase Interaction** **Drag Force** (selected automatically) **Multiphase Material** (selected automatically)
Interaction Length Scale	**Interaction Length Scale** (selected automatically)
Interaction Area Density	**Interaction Area Density** (selected automatically)
Optional Models	**Turbulent Dispersion Force**
Virtual Mass Coefficient	**Virtual Mass Coefficient**
Optional Models	**Interphase Mass Transfer**
Interphase Mass Transfer Rate	**Boiling Mass Transfer Rate**
Optional Models	**Wall Boiling**
Bubble Nucleation	**Wall Bubble Nucleation**
Transient Conduction	**Wall Transient Conduction**

4. Click **Close**.
5. As a wide range of volume fractions is expected, ensure that the **Interaction Area Density** model has the *Method* set to **Symmetric**.
6. Click **Close**.

The **Liquid Vapor Interface** > **Models** node is shown below:



Defining the Phase Interaction Properties

Define the interactions between the phases.

The phase interaction properties are now defined, starting with the definition of the continuous and dispersed phases.

To define the phase interaction properties:

1. Edit the **Liquid Vapor Interface** > **Models** node and set the following properties:

Node	Property	Setting
Continuous-Dispersed Phase Interaction	**Continuous Phase**	**Liquid**
	Dispersed Phase	**Vapor**

Once the Continuous and Dispersed phases have been defined, you select additional models:

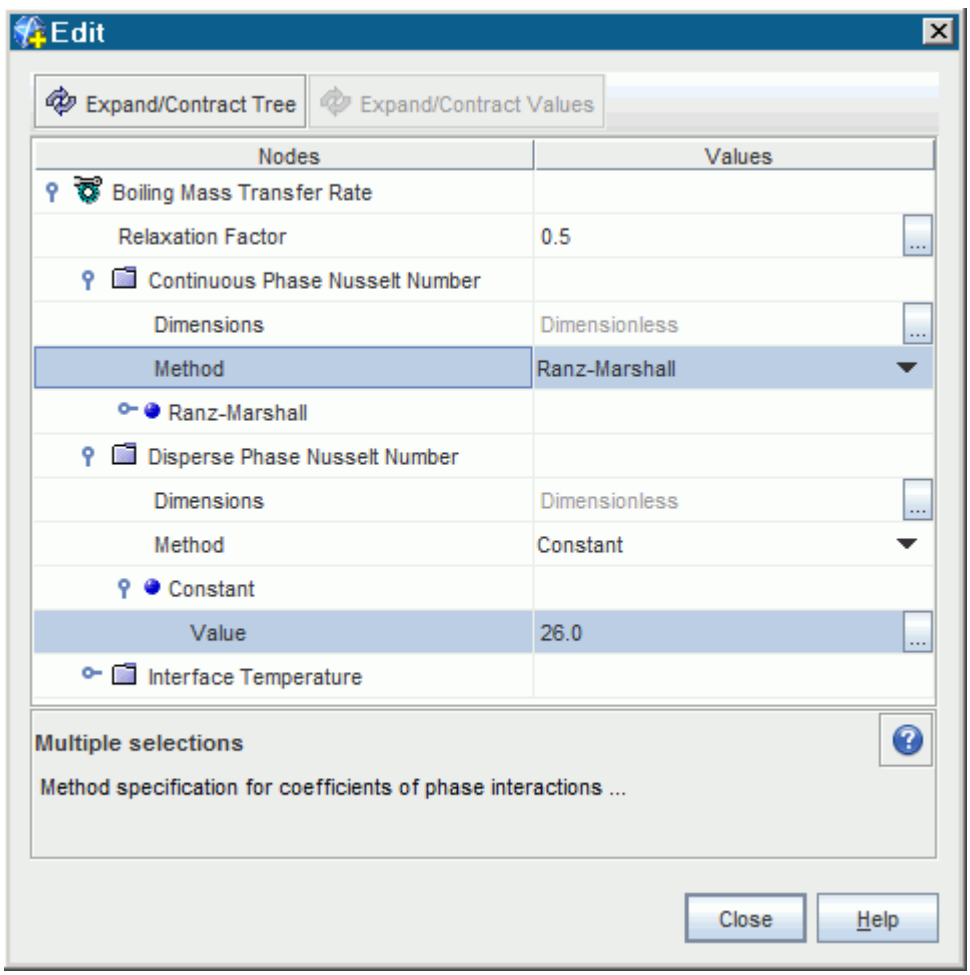
2. For the **Liquid Vapor Interface** phase interaction, select the following models:

Group Box	Model
Optional Models	**S-Gamma Breakup**
	S-Gamma Coalescence

Define the remainder of the phase interaction properties.

3. Edit the **Liquid Vapor Interface** > **Models** node and set the following properties:

Node	Property	Setting
Boiling Mass Transfer Rate > **Continuous Phase Nusselt Number**	**Method**	**Ranz-Marshall**
∟ **Disperse Phase Nusselt Number** > **Constant**	**Value**	`26.0`
Drag Force > **Drag Coefficient**	**Method**	**Tomiyama**
∟ **Drag Correction** > **Volume Fraction Exponent**	**Volume Fraction Exponent (continuous)**	`0.0`
Turbulent Dispersion Force	**Use Logarithmic Form**	Activated
Interaction Length Scale > **Interaction Length Scale**	**Method**	**Field Function**
∟ **Field Function**	**Scalar Function**	**ClippedSauterMeanDiameter**



Selecting the **Ranz-Marshall** option makes STAR-CCM+ calculate the **Continuous Phase Nusselt Number** using the Ranz-Marshall correlation. The Ranz-Marshall correlation, in turn, is used to calculate the heat transfer coefficient for the heat transfer from the bulk water to the bubble interface.

The **Disperse Phase Nusselt Number** is used when calculating the heat transfer coefficient for the heat transfer from the bubble interface to the bulk steam. The exact value is not critical because the vapor is going to be at, or close to, saturation temperature. A value of 2 represents a slow response with little internal circulation. A higher value such as 26 can represent either a fast superficial response or the effect of internal mixing within the vapor.

The **Tomiyama** model is used for modeling bubbly flows. Drag reduction is not expected for small spherical bubbles.

The **Turbulent Dispersion Force** setting is important for wall boiling calculations in the presence of condensation because it treats the differencing of rapidly changing volume fraction gradients more accurately. Without this setting, the simulation can get unusual velocities at the edge of the spreading bubble cloud as well as unnecessary convergence difficulties.

The **Interaction Length Scale** is based on the bubble size predicted by the S-Gamma model. The **ClippedSauterMeanDiameter** field function is one of the user-defined field functions that was created earlier. This setting allows the Sauter Mean Diameter that is predicted with the S-Gamma model to be used as the interaction length scale for drag and interface transfers. It also filters the calculations through a maximum and a minimum diameter so that early unconverged S-Gamma solutions do not upset convergence.

4. Edit the **Physics 1** > **Models** > **Eulerian Multiphase** > **Eulerian Phases** > **Vapor** > **Models** node and set the following properties:

Node	Property	Setting
Issa Turbulence Response Model	**Phase Interaction**	**Liquid Vapor Interface**
S-Gamma	**Number of S-Gamma Equations**	**One Equation**

5. Save the simulation.

Setting the Material Properties

Define the phases and set the appropriate material properties.

In this high-pressure calculation where wall dryout is not expected, the vapor remains at saturation temperature. Therefore it is valid to assume constant vapor properties and constant interface properties.

The liquid on the other hand can experience a significant change in density and in specific heat while it is heated from a subcooling of 60 K towards saturation temperature. Nevertheless a useful analysis can be made using constant properties for the liquid by respecting these two conditions:

- In order to obtain the correct heat balance, while also using the experimental inlet temperature, the following definition of liquid specific heat was chosen:

$$C_p = \frac{h_L^{SAT} - h_L^{IN}}{T^{SAT} - T^{IN}}$$

where *SAT* indicates saturation, *IN* indicates inlet, and *L* indicates liquid.

- The liquid density, viscosity, and conductivity are chosen at a temperature midway between inlet and saturation. The inlet velocity must be chosen such that inlet velocity times density matches the experimental mass flux.

In this tutorial, the process of setting the properties has been deferred until after all the models have been selected. Some properties appear only when demanded by particular models.

Specify the material properties for the liquid phase.

1. Edit the **Eulerian Multiphase** > **Eulerian Phases** > **Liquid** > **Models** > **Liquid** > **H20** > **Material Properties** node and set the following properties:

Node	Property	Setting
Density > **Constant**	**Value**	`$InletDensity`
Dynamic Viscosity > **Constant**	**Value**	`0.0001177 Pa-s`
Heat of Formation > **Constant**	**Value**	`1122143 J/kg`
Specific Heat > **Constant**	**Value**	`4648.3 J/kg-K`
Standard State Temperature > **Constant**	**Value**	`$SaturationTemperature`
Thermal Conductivity > **Constant**	**Value**	`0.6433 W/m-K`

Set the initial temperature of the liquid phase:

2. Select the **Liquid** > **Initial Conditions** > **Static Temperature** node and set the **Value** to `461.0 K`.

   The settings for the liquid phase are now complete.

Specify the material properties for the vapor phase.

In general the phase properties that are labeled as Heat of Formation and Standard State Temperature are used internally by STAR-CCM+ as a general Reference Enthalpy at a specified Reference Temperature. For this boiling application, these properties are chosen to be Saturation Enthalpy and Saturation Temperature respectively from standard steam tables.

Latent Heat is computed automatically from the phase enthalpy definitions.

3. Edit the **Vapor** > **Models** > **Gas** > **Steam** > **Material Properties** node and set the following properties:

Node	Property	Setting
Density > **Constant**	**Value**	`22.7 kg/m^3`
Dynamic Viscosity > **Constant**	**Value**	`1.78E-5 Pa-s`
Heat of Formation > **Constant**	**Value**	`2797997 J/kg`
Molecular Weight > **Constant**	**Value**	`18.0 kg/kmol`

Node	Property	Setting
Specific Heat > **Constant**	**Value**	`4227.9 J/kg-K`
Standard State Temperature > **Constant**	**Value**	`$SaturationTemperature`
Thermal Conductivity > **Constant**	**Value**	`0.0535 W/m-K`

4. Select the **Liquid Vapor Interface** > **Models** > **Multiphase Material** > **Multiphase Material** > **Material Properties** node and set the following properties:

Node	Property	Setting
Boiling Temperature	**Method**	**Field Function**
∟ **Field Function**	**Scalar Function**	`$SaturationTemperature`
Surface Tension > **Constant**	**Value**	`0.024292 N/m`

Setting Reference Values

Set the reference values for the physics continuum.

The reference gravity, density, and pressure for the simulation are set up. The main modeling choice here is to take the reference density for the buoyancy term from the liquid phase density. This choice allows the simulation to move smoothly from a zero initial field for pressure and vapor volume fraction, rather than requiring a hydrostatic initial pressure field.

To set the reference values:

Set the reference values for the **Physics 1** continuum.

1. Edit the **Reference Values** node and set the following properties:

Node	Property	Setting
Gravity	**Value**	`[-9.81, 0.0] m/s^2`
Reference Density	**Value**	`$InletDensity`
Reference Pressure	**Value**	`$SystemPressure`

2. Save the simulation.

Setting Boundary Conditions

Define the boundary types and specify the appropriate property values.

Specify conditions on the wall and inlet boundaries.

To set the boundary conditions:

1. Edit the **Regions** > **Boiling_Tube** > **Boundaries** > **Wall** node and set the following properties:

Node	Property	Setting
Physics Conditions > **Thermal Specification**	**Condition**	**Heat Flux**
Physics Values > **Heat Flux**	**Method**	**Field Function**
	Scalar Function	**WallHeatFluxIterationRamp**

Set the phase conditions for the liquid and vapor phases at the inlet. The turbulence intensity and turbulent viscosity ratio are modified so that the settings match the Initial Conditions.

2. Edit the **In** > **Phase Conditions** node and set the following properties:

Node	Property	Setting
Liquid > **Physics Values**		
⌊ **Static Temperature**	**Value**	`$InletTemperature`
⌊ **Turbulence Intensity**	**Value**	`0.01`
⌊ **Turbulent Viscosity Ratio**	**Value**	`200`
⌊ **Velocity Magnitude**	**Method**	**Field Function**
	Scalar Function	**InletVelocity** > **Laboratory** > **i**
⌊ **Volume Fraction**	**Value**	`1.0`

Node	Property	Setting
Vapor > **Physics Values**		
⌊ **Sauter Mean Diameter**	**Value**	`0.001`
⌊ **Static Temperature**	**Value**	`$SaturationTemperature`
⌊ **Velocity Magnitude**	**Method**	**Field Function**
	Scalar Function	**InletVelocity** > **Laboratory** > **i**
⌊ **Volume Fraction**	**Value**	`0.0`

Define the outlet boundary. The static temperature for the liquid and vapor phases at the outlet are set. These values are the same as for the inlet.

3. Edit the **Out** > **Phase Conditions** node and set the following properties:

Node	Property	Setting
Liquid > **Physics Values**		
⌊ **Static Temperature**	**Value**	`$InletTemperature`
⌊ **Turbulence Intensity**	**Value**	`0.01`

Node	Property	Setting
⌊ **Turbulent Viscosity Ratio**	**Value**	`10`
⌊ **Volume Fraction**	**Value**	`1.0`

Node	Property	Setting
Vapor > **Physics Values**		
⌊ **Sauter Mean Diameter**	**Value**	`0.001`
⌊ **Static Temperature**	**Value**	`$SaturationTemperature`
⌊ **Volume Fraction**	**Value**	`0.0`

4. Save the simulation.

Setting Initial Conditions

Set the global initial conditions and the initial conditions for each phase.

To set the initial conditions:

1. Edit the **Physics 1** > **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and set the following properties for the liquid phase:

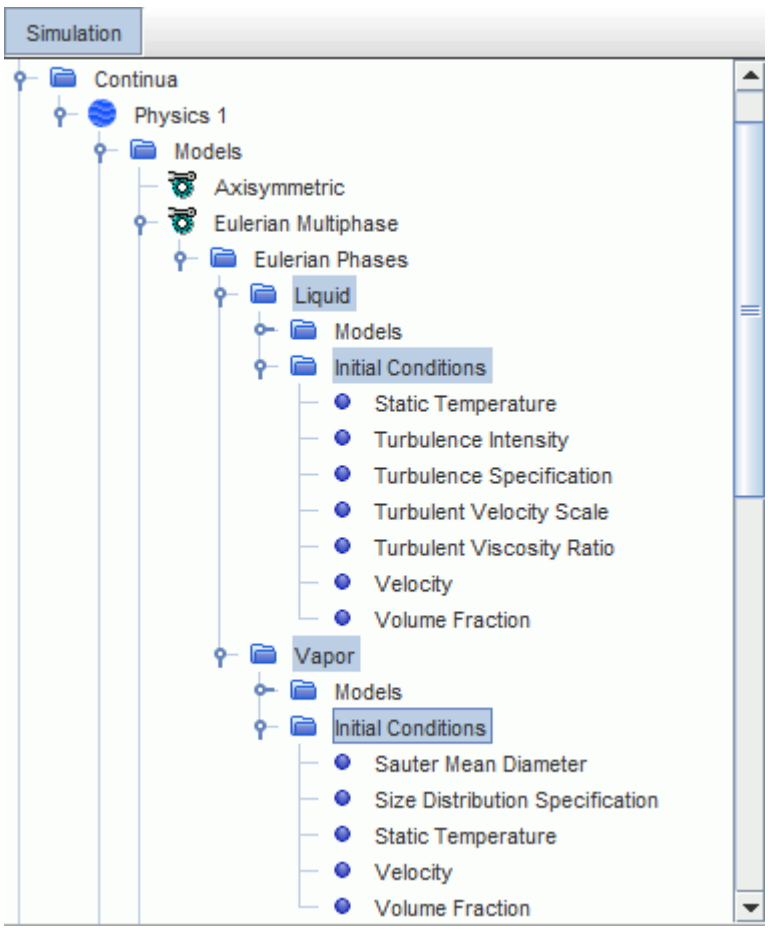
Node	Property	Setting
Liquid > **Initial Conditions**		
⌊ **Static Temperature**	**Value**	`$InletTemperature`
⌊ **Turbulence Intensity**	**Value**	`0.01`
⌊ **Turbulence Specification**	**Method**	**Intensity + Viscosity Ratio**
⌊ **Turbulence Velocity Scale**	**Value**	`1.0 m/s`
⌊ **Turbulent Viscosity Ratio**	**Value**	`200`
⌊ **Velocity**	**Method**	**Field Function**
	Vector Function	**InletVelocity**
⌊ **Volume Fraction**	**Value**	`1.0`

2. Edit the same node and set the following properties for the vapor phase:

Node	Property	Setting
Vapor > **Initial Conditions**		

Node	Property	Setting
∟ **Sauter Mean Diameter**	**Value**	`0.01 m`
∟ **Size Distribution Specification**	**Method**	**Sauter Mean Diameter**
∟ **Static Temperature**	**Value**	`$SaturationTemperature`
∟ **Velocity**	**Method**	**Field Function**
	Vector Function	**InletVelocity**
∟ **Volume Fraction**	**Value**	`0.0`

The phase equations cannot be solved when volume fraction is identically zero. Therefore, for boundary conditions and for initial conditions, the code uses 10E-12 for vapor volume fraction rather than 0.0.



3. Save the simulation.

Setting Solver Parameters and Stopping Criteria

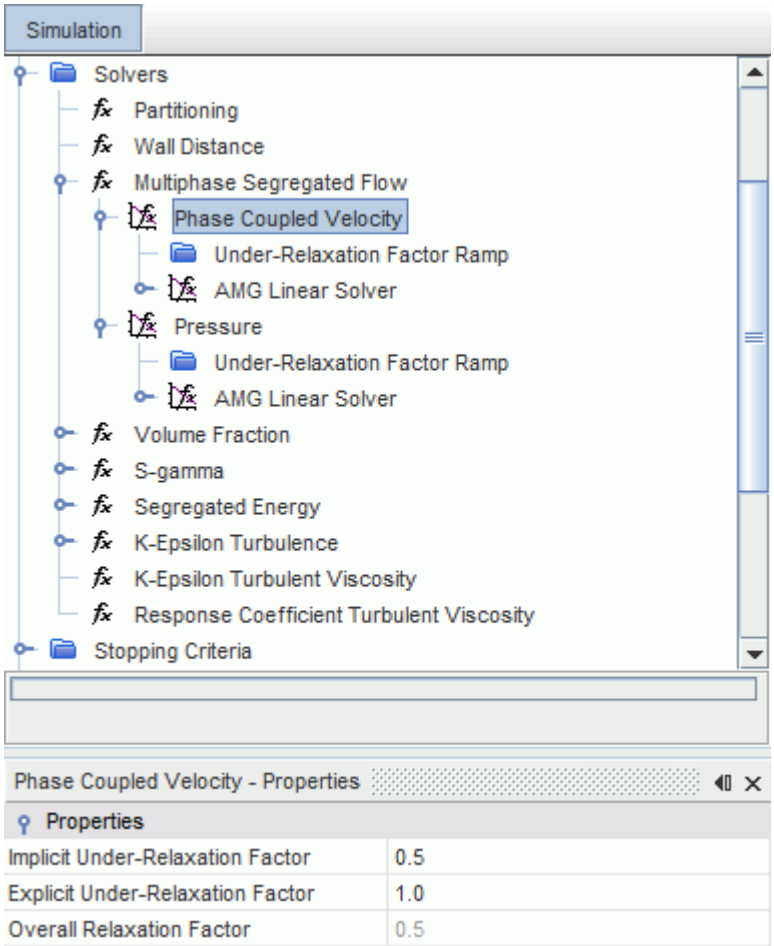
Set the appropriate solver parameters and stopping criteria for the simulation.

The under-relaxation factors for the solvers are decreased to help the solution converge. These relaxation factors have not been tuned for best performance in this one particular case. Rather, they have been chosen to also allow converged solutions to be obtained for many of the test conditions reported in Bartolemei (1982) ([801]).

To set the solver parameters and stopping criteria:

1. Edit the **Solvers** node and set the following properties:

Node	Property	Setting
Multiphase Segregated Flow > Phase Coupled Velocity	**Implicit Under-Relaxation Factor**	`0.5`
Multiphase Segregated Flow > Pressure	**Under-Relaxation Factor**	`0.1`
Volume Fraction	**Implicit Under-Relaxation Factor**	`0.1`
S-gamma	**Implicit Under-Relaxation Factor**	`0.1`
Segregated Energy	**Fluid Under-Relaxation Factor**	`0.5`
K-Epsilon Turbulence	**Under-Relaxation Factor**	`0.5`



The Sauter Mean Diameter is predicted from the ratio of S2 and volume fraction. Therefore the **Volume Fraction** and **S-gamma** solvers are given the same relaxation factors to ensure that these two fields are progressed together.

In addition to solver relaxation factors, there are further relaxation factors that are associated with some specific model terms.

2. Edit the **Continua** > **Physics 1** > **Models** > **Multiphase Interaction** > **Phase Interactions** > **Liquid Vapor Interface** > **Models** node and set the following properties:

Node	Property	Setting
Boiling Mass Transfer Rate	**Relaxation Factor**	`0.1`
Wall Boiling	**Relaxation Factor**	`0.1`
∟ **Wall Dryout Area Fraction**	**Bubbly Layer Relaxation Factor**	`0.5`

3. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Steps	**Enabled**	Activated

Node	Property	Setting
	Maximum Steps	`2500`

4. Save the simulation.

Setting Up Monitoring Plots

Create plots to monitor the progress of the simulation.

Since the simulation is starting with no boiling and with little vapor, the final residuals can be many orders of magnitude different from the initial levels.

To help determine when the simulation has converged, you can also monitor the progress of the following:

- maximum velocity
- maximum and minimum temperature
- maximum volume fraction
- mass balance

Any of these plots can be set up individually by applying the **Create Monitor and Plot from Report** action to selected groups of various reports operating on relevant fields or field functions.

For convenience, the complete set of monitoring plots for this tutorial can be set up by running a macro.

To set up the monitoring plots:

1. Select **File** > **Macro** > **Play Macro**.
2. In the *Open* dialog, navigate to your working directory and select `wb_setupMonitorPlots.java`.
3. Click **Open** to run the macro.

In this tutorial, the maximum and minimum plots are expected to confirm that the velocities and temperatures remain physically plausible throughout the convergence history. This result is due to the strategy of ramping up the power gradually as the flow field is established.

In more difficult or unfamiliar cases, maximum and minimum plots can be helpful to identify the first signs of when a simulation is starting to get out of hand and can need adjustments to relaxation factors.

The Mass Balance plot tracks three quantities, as follows:

- LiquidMassFlow, which computes the mass outflow of liquid less the mass inflow of liquid.
- VaporMassFlow, which computes the mass outflow of vapor less the mass inflow of vapor.
- NetEvaporationRateReport, which is constructed from the expression $WallEvaporationRateReport + $BulkEvaporationRateReport - $BulkCondensationRateReport.

  where:

  - BulkCondensationRateReport is the volume integral of the positive part of the interior IPMT rate from the vapor phase to the liquid phase.
  - BulkEvaporationRateReport is the volume integral of the negative part of the interior IPMT rate from the vapor phase to the liquid phase.
  - WallEvaporationRateReport is the surface integral of minus the wall IPMT rate from the vapor phase to the liquid phase.

The IPMT (Interphase Mass Transfer) rate is defined to be positive where mass is transferred from the second (dispersed) phase to the first (continuous) phase that is specified under the Eulerian Continuous-Dispersed phase interaction. For this tutorial, this definition means that a positive IPMT rate indicates condensation, and a negative value indicates evaporation.

On convergence, the mass balance is expected to be obeyed for each phase, at least to some finite precision:

- LiquidMassFlow + NetEvaporationRateReport = 0
- VaporMassFlow - NetEvaporationRateReport = 0

Setting Up Analysis Plots

Create plots to analyse the results of the simulation.

It is also instructive to observe quantities of interest as convergence is approached. You set up plots of Axis Subcooling and Wall Superheat which can be compared directly with data given by Bartolemei (1967) ([800]). An important feature of the model is the automatic partitioning of the applied wall heat flux into three components (liquid convection, evaporation, and vapor convection). This partitioning is presented as a Wall Heat Flux plot. A number of other submodel quantities, such as bubble size, are of interest to researchers in this field.

For convenience, the analysis plots for this tutorial can be set up by running a macro.

To set up the analysis plots:

1. Select **File** > **Macro** > **Play Macro**.
2. In the *Open* dialog, navigate to your working directory and select `wb_setupAnalysisPlots.java`.
3. Click **Open** to run the macro.

Many plots present results on the wall or on the axis against height. However, because of uncertainty in the exact location and conditions at the start of the heated section, a comparison with Bartolemei (1967) ([800]) or Bartolemei (1982) ([801]) is best done by using thermodynamic quality as a non-dimensional scale. Thermodynamic quality is easily related to height from the heat balance.

The definition of thermodynamic quality is:

$$x_{th} = \frac{h - h_L^{SAT}}{h_{LV}}$$

where *h* is the mixture enthalpy as calculated by this heat balance:

$$h = h_L^{IN} + \frac{4qz}{GD}$$

For convenience, the thermodynamic quality has been set up as a user field function.

The components of heat flux between the wall and the liquid, between the wall and the vapor, and between the wall and the evaporating interface, are available as standard field functions. These field functions are suffixed with the relevant phase or interaction name:

- Boundary Heat Flux of Liquid
- Boundary Heat Flux of Vapor
- Boundary Heat Flux of Liquid Vapor Interface

Boundary heat fluxes follow the sign convention in that they are positive when directed outwards. This sign convention is corrected for the presentation of the Wall Heat Flux plot.

Wall superheat is defined as the wall temperature excess over the saturation temperature. It is defined here as a user field function. The wall temperature itself is obtained by evaluating the temperature of either phase at the wall.

Liquid subcooling is defined as the saturation temperature less the liquid temperature. To ensure that this value can also be plotted at the wall, the Ignore Boundary Values option is activated when the field function is created. This setting ensures that the liquid temperature in the cell next to the wall is used rather than the wall temperature.

Visualizing the Solution

Create a scalar scene to display the results of the simulation.

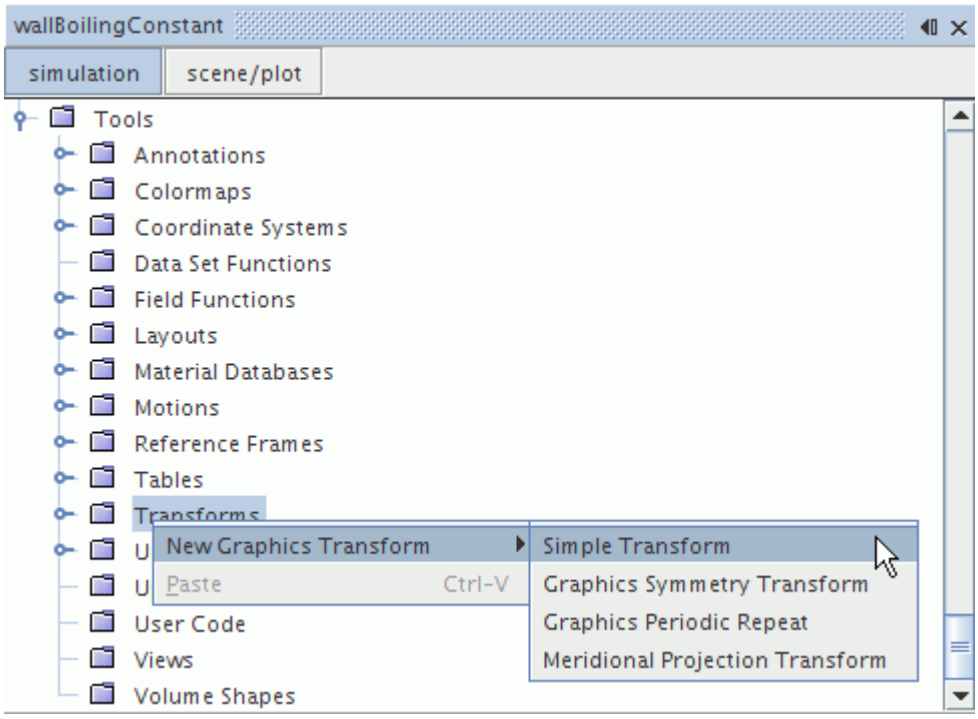
A scalar scene is created to display the volume fraction of vapor. This variable can be monitored while the solver is running.

To visualize the solution:

1. Create a scalar scene.

The view of the pipe in the scene is scaled using a simple transform so that the entire length of the pipe can be viewed in a smaller area. The transform also rotates the geometry so that it sits vertically.

2. Right-click the **Tools** > **Transforms** node and select **New Graphics Transform** > **Simple Transform**.

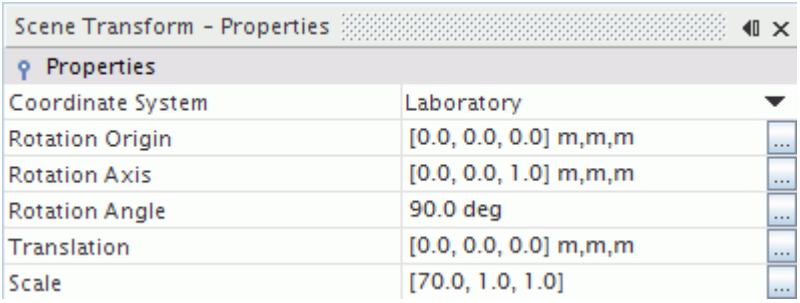


A new node, **Simple Transform 1**, is added to the **Transforms** node.

3. Rename the **Simple Transform 1** node to `Scene Transform`.

4. Select the **Scene Transform** node and do the following:
   a) Set the *Scale* to `[70.0, 1.0, 1.0]`.
   b) Set the *Rotation Angle* to `90 deg`.

   The completed *Properties* window is shown below.



5. Click **scene/plot**.
6. Select the **Scalar Scene 1** > **Displayers** > **Scalar 1** node and do the following:
   a) Set the *Contour Style* to **Smooth Filled**.
   b) Set the *Transform* to **Scene Transform**.

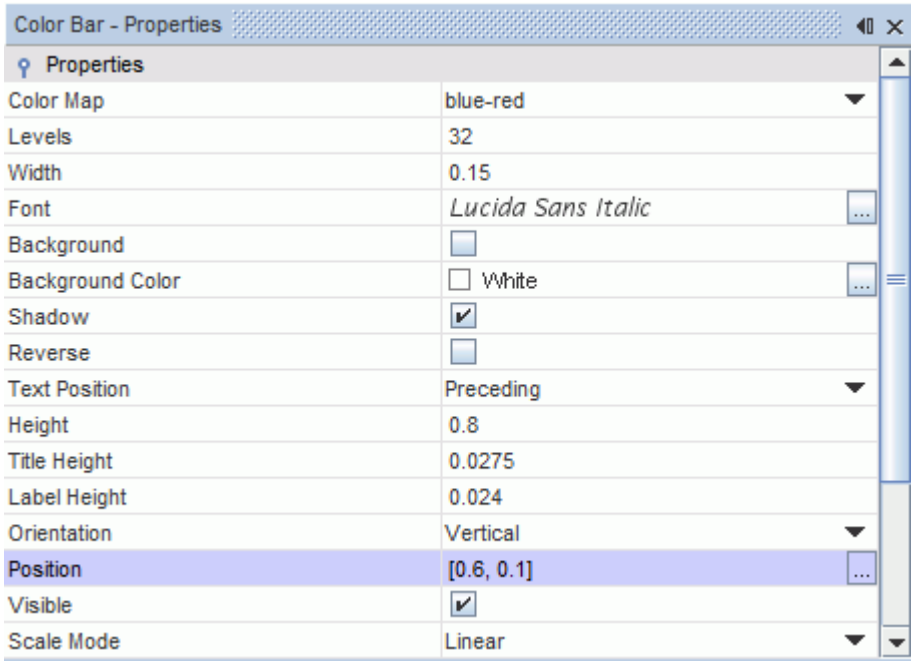
Set the field function for the scalar scene:

7. Select the **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Scalar Field** node and set the *Function* to **Volume Fraction** > **Volume Fraction of Vapor**.

The color bar is edited so that it is positioned beside the vertical pipe.

8. Select the **Displayers** > **Scalar 1** > **Color Bar** node and do the following:
   a) Set the *Orientation* to **Vertical**.
   b) Set the *Height* to `0.8`.
   c) Set the *Width* to `0.15`.
   d) Set the *Position* to `[0.6, 0.1]`.

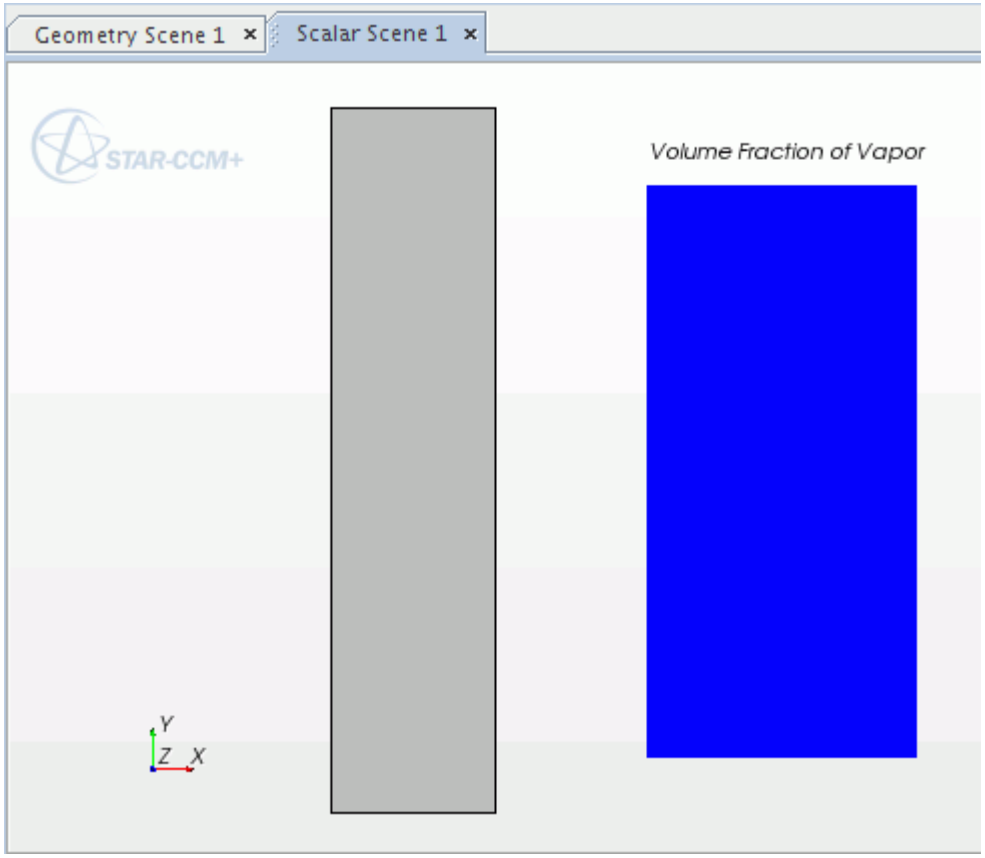
   The completed *Properties* window is shown below.



The **Scene Transform** is also applied to the **Outline 1** node.

9. Select the **Scalar Scene 1** > **Displayers** > **Outline 1** node and set the *Transform* to **Scene Transform**.

   The scalar scene appears as shown below.



10. Click **simulation** to return to the STAR-CCM+ simulation object tree.
11. Save the simulation.

Running the Simulation

Preparation of the simulation is now complete, and the simulation can be run.

To run the simulation:

1. Click (**Run**) in the *Solution* toolbar.

   The solution progress is displayed in the *Output* window. The *Residuals* display is created automatically in the *Graphics* window and shows the progress of the solvers.

   While the simulation is running, you can click the tabs at the top of the *Graphics* window to view scenes and plots.

   During the run, you can stop the process by clicking (**Stop**) on the toolbar. If you do stop the simulation, you can click (**Run**) to resume it.

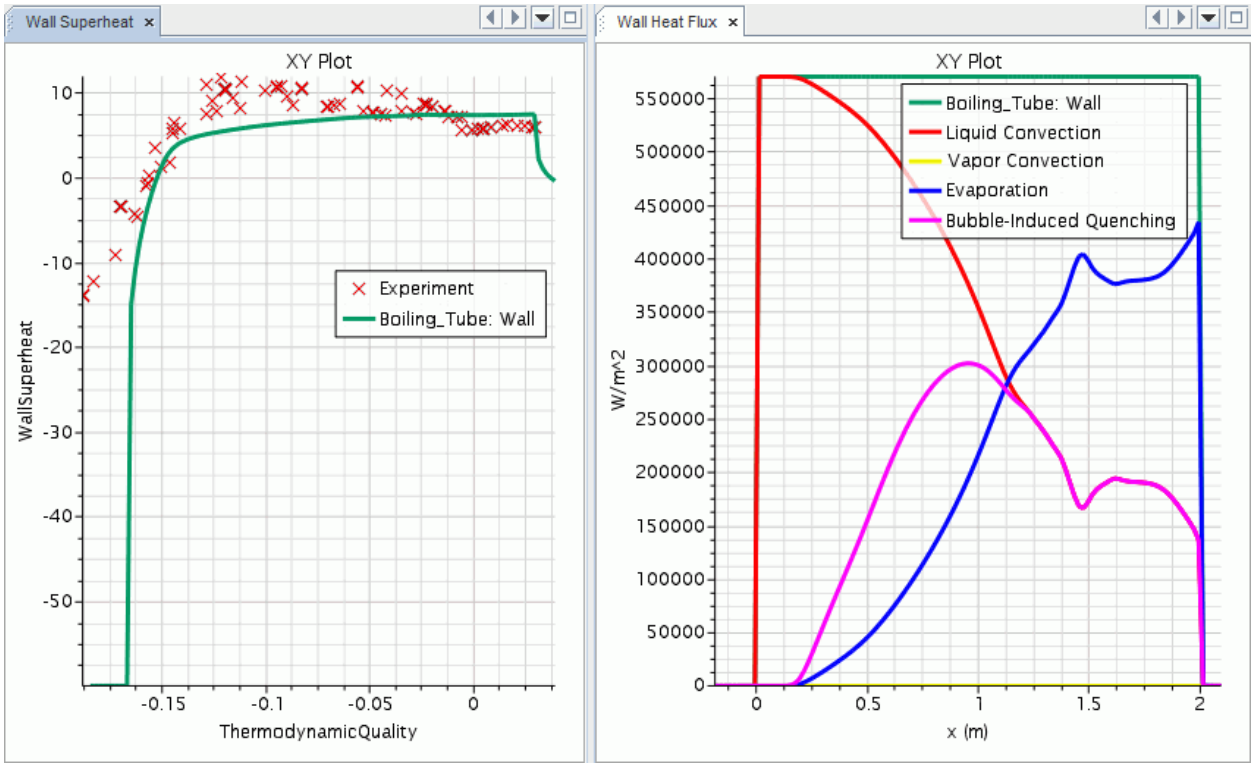
   If left alone, the simulation continues until 2500 iterations are complete. When the last iteration is complete, the following message is displayed in the *Output* window:

   ```
   Stopping criterion Maximum Steps satisfied.
   ```

As well as observing the progress of residuals during convergence, an interesting feature to follow is the development of the heat flux partitioning.

2. Expand the **Plots** node and then open the *Wall SuperHeat* and *Wall Heat Flux* plots.
3. In the *Graphics* window, click the tab of one of these plots and drag it to the left or right margin of the window.

   The two plots are then displayed side by side, as in the example below.



The wall superheat is initially negative (below boiling point), but rises as the wall power is ramped up, and becomes positive just before full power is achieved at 200 iterations. During this period, the whole of the heat flux is directed to the liquid only. Once boiling point is achieved, an evaporation heat flux takes up a share of the applied wall heat flux. As the calculation continues, the liquid convection in the boiling region is replaced with a quenching heat flux due to the agitation or mixing of the liquid by departing bubbles. Both evaporation (latent heat) and quenching (agitation) are major enhancers of heat transfer during nucleate boiling.

At the inception of boiling, single phase heat transfer is the only contributor to the wall heat flux. As the wall superheat rises, more nucleations sites become active, and both the evaporation and quenching contributions rise accordingly. However, the quenching contribution goes through a maximum, because it is also proportional to liquid subcooling which is falling as the liquid temperature rises.

The overall balance between quenching and evaporation fluxes is a function of operating pressure, since quenching flux varies as departure size squared and evaporation flux varies as departure size cubed. The bubble departure size is smaller at higher pressures.

The wall superheat profile is established early thanks to the sensitivity of nucleation site density to a large power of wall superheat. Net vapor production appears a little later, at about 400 iterations, once the condensation capacity of the subcooled water exceeds the wall evaporation rate. At the same time, the flow begins to accelerate, as can be seen from the maximum velocity monitor plot.

True convergence of mass balance and of residuals can be seen to start after 600 iterations. The maximum/minimum plots show small changes after iteration 2000. The calculation is stopped at 2500 iterations to get a good final mass balance.

4. When the simulation has finished running, save it.

Final Mass Balance

The final mass balance is summarized.

Liquid Inflow	2.6681E-2 kg/s
Wall Evaporation	1.9231E-3 kg/s
Bulk Evaporation	4.2551E-4 kg/s
Total Evaporation	2.3486E-3 kg/s
Less Bulk Condensation	1.5009E-3 kg/s
Net Evaporation	8.4769E-4 kg/s
Net Evaporation as % of Liquid Inflow	3.18%

Vapor Outflow less Inflow	8.4769E-4 kg/s
less Net Evaporation Rate	8.4769E-4 kg/s
Error	0.0000E+00 kg/s
Error as % of Liquid Inflow	0.00%

Liquid Outflow less Inflow	-8.4768E-4 kg/s
plus Net Evaporation Rate	8.4769E-4 kg/s
Error	3.4000E-9 kg/s
Error as % of Liquid Inflow	0.00%

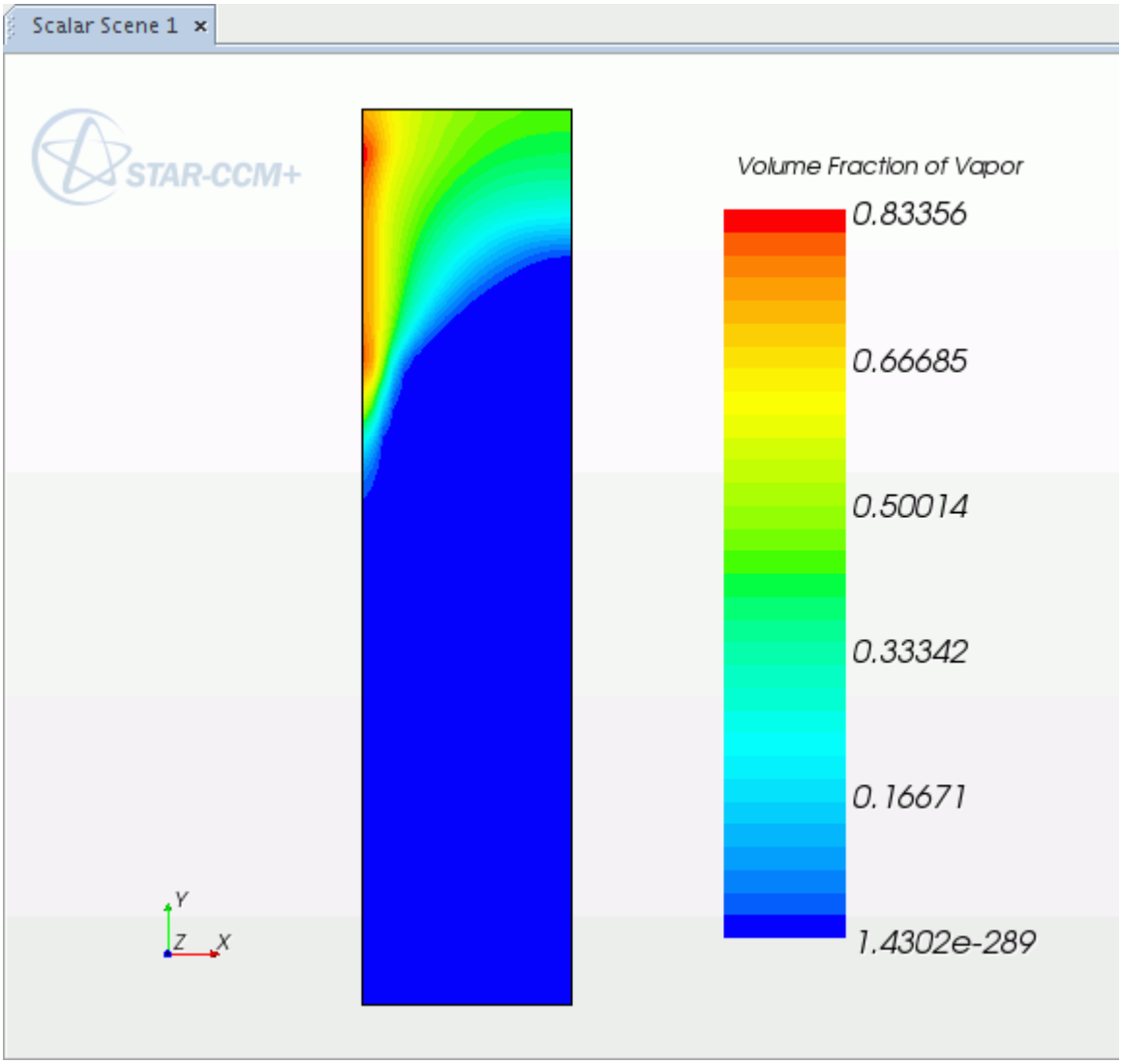
The net evaporation is 3.18% percent of the mass flow.

This figure also represents the true quality (mass fraction of steam) at the outlet.

Visualizing the Results

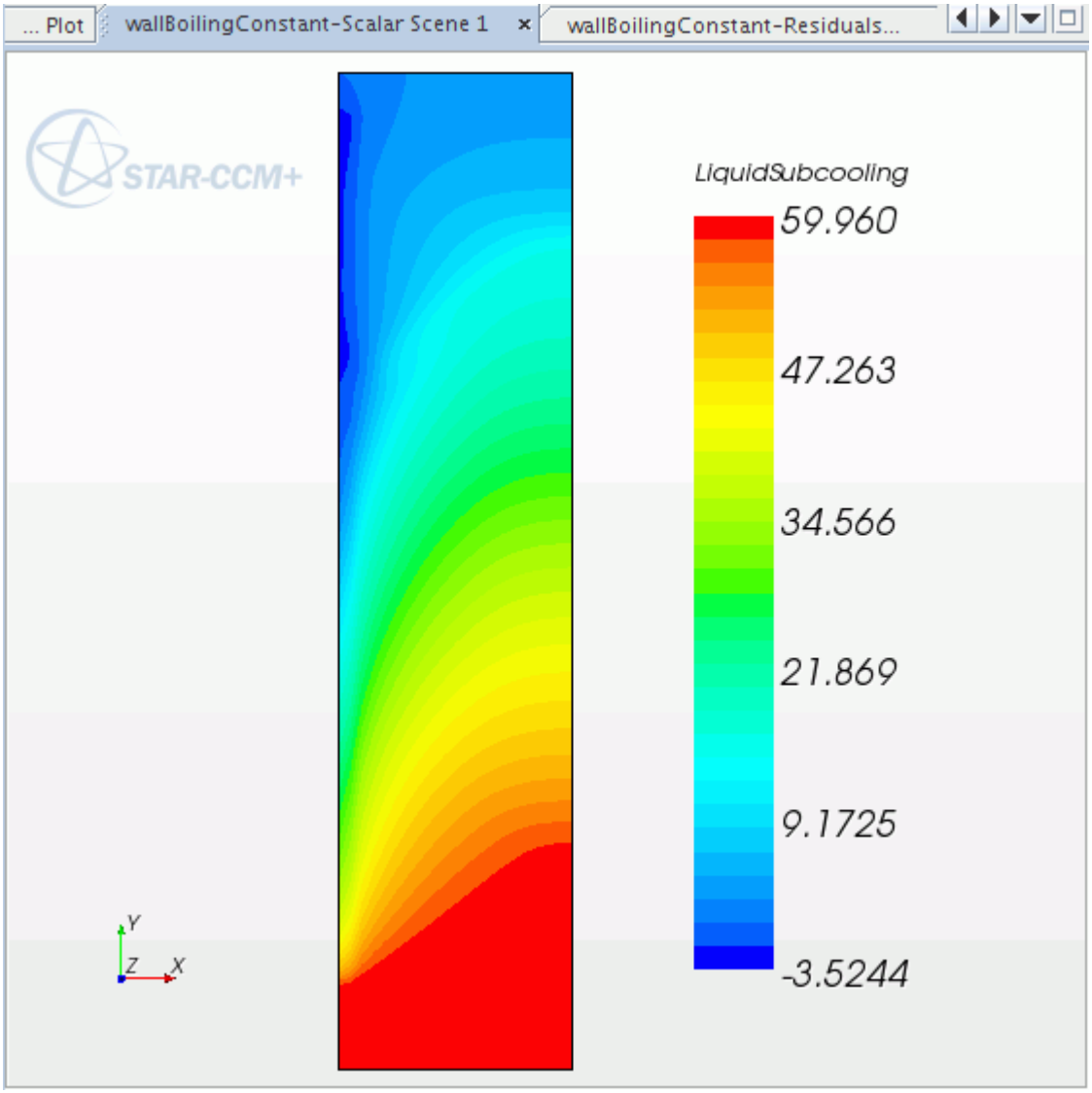
Examine the results of the simulation.

The scalar scene depicting the volume fraction of the vapor is shown below.



The first thing to look at is the vapor volume fraction distribution and to compare it with the liquid subcooling. This comparison can be done by switching the Scalar Field that is selected for the Scalar Scene.

The scalar scene depicting the liquid subcooling is shown below.



A small negative subcooling appears next to the wall, which in effect means that the liquid is superheated by a few degrees. The reason that it has not yet evaporated, is that the rate of evaporation is limited by the rate of heat transfer between the liquid and the liquid-vapor interface. Most of the vapor is confined to the region where the subcooling is not too high, below around 10 K. Elsewhere the vapor is quickly condensed as soon as it is produced on the wall. The vapor appears to spread quickly across the flow in this scaled scene, but in fact it is spread by the turbulent velocity fluctuations, modeled using the turbulent dispersion force.

Comparison With Experiment

Compare the results of the simulation with experimental data.

The main result of interest is the axial distribution of mean void. Since this result is a post processing calculation that is difficult to do without making a special assumption of a uniform grid, this calculation is made using a customized user macro:

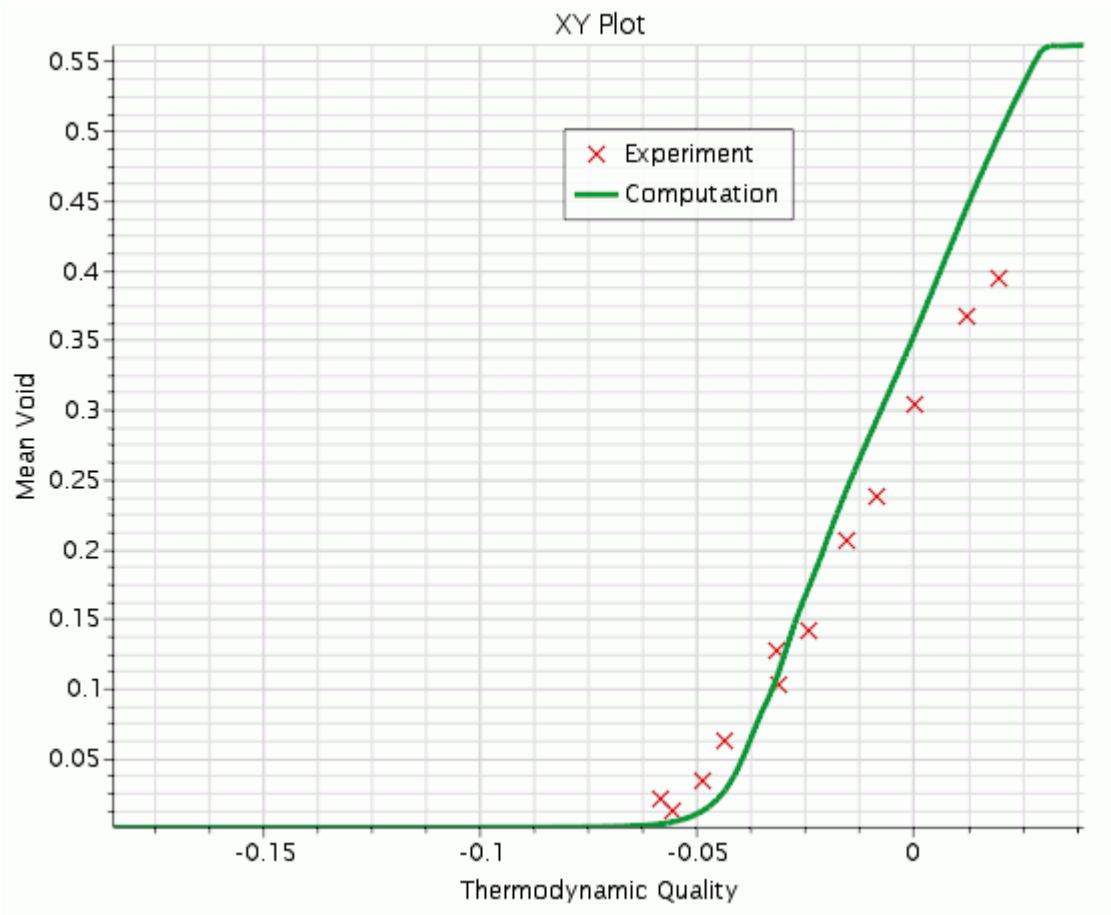
1. Select **File** > **Macro** > **Play Macro**.

2. In the *Open* dialog, navigate to your working directory and select `wb_createAxialPlots.java`.
3. Click **Open** to run the macro.

   The macro creates two new plots entitled *Compare Axial Void* and *Compare Calorimetric Temperature*. At the same time, the macro loads experimental data from file.

Note: To refresh these plots, run the `wb_deleteAxialPlots.java` macro first.

The *Compare Axial Void* plot is shown below:



The prediction of axial void is good near the start of net vapor generation and passes through the first few experimental points up to an intermediate level of void. Beyond this intermediate level, it over predicts the void by about 20% at the last experimental point. The agreement could be improved with model tuning. However, it is more satisfactory that this set of models and relaxation factors can provide solutions to many of the cases that are specified in Bartolemei (1982) [801] without any further adjustments.

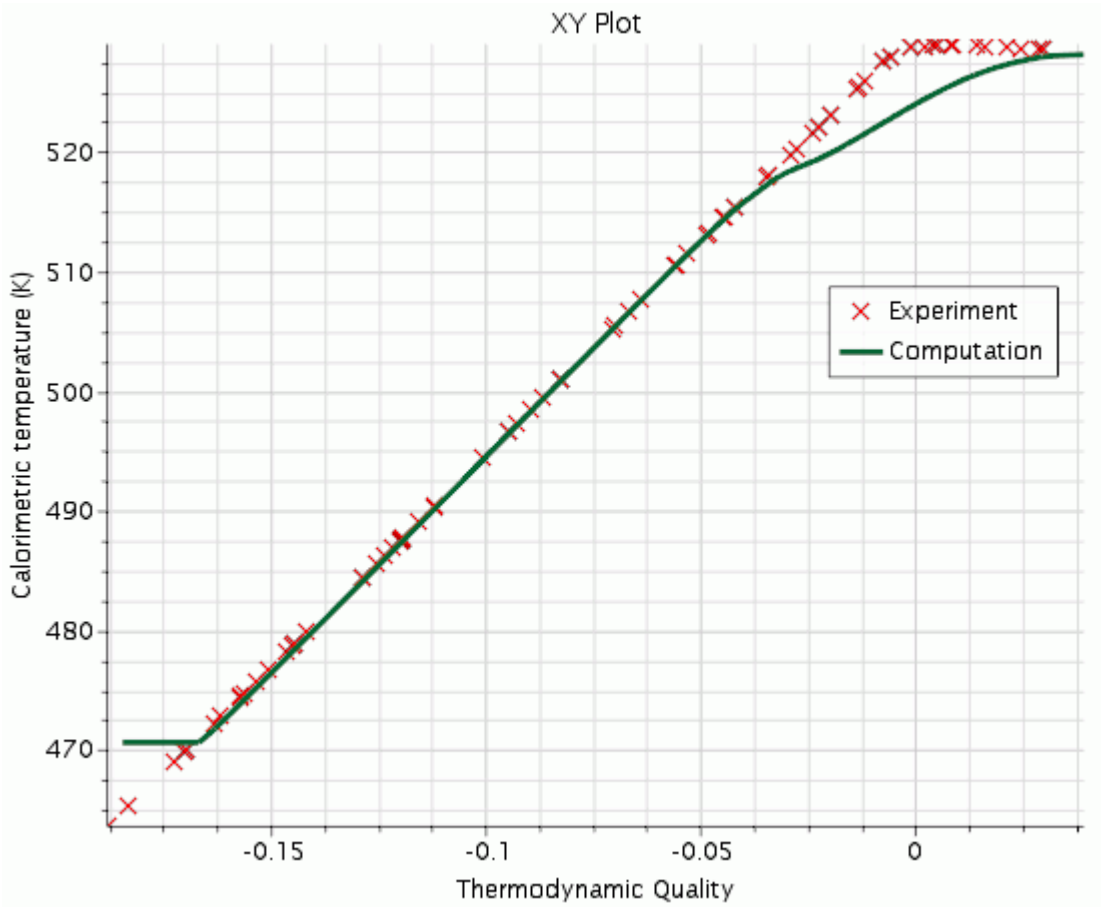
This wide range of applicability is largely due to calibrations of Kocamustafaogulari and of Hibiki-Ishii to a wide range of pressures up to 150 bar. At 150 bar the bubble departure size becomes very small (about 5 micron compared to 50 micron for this test at 45 bar). It is also due to the capability of the S-gamma model to track the bubble size in subcooled flow starting from the Kocamustaogulari model for departure diameter.

For more information, see the following sections:

- Hibiki Ishii Nucleation Site Number Density
- Kocamustafaogullari Bubble Departure Diameter

The Calorimetric Temperature that is provided by Bartolemei (1967) [800] is not fully explained in the paper, and is a little difficult to interpret. The name as translated from Russian suggests a velocity-weighted temperature averaged across the diameter, and the macro has prepared such an average from the computed results.

The *Compare Calorimetric Temperature* plot is shown below:



Bartolemei (1967) [800] does indeed provide a large number of points for Calorimetric Temperature as if these were experimental data. These points appear to lie on a straight line until a constant saturation temperature is reached exactly at zero thermodynamic quality.

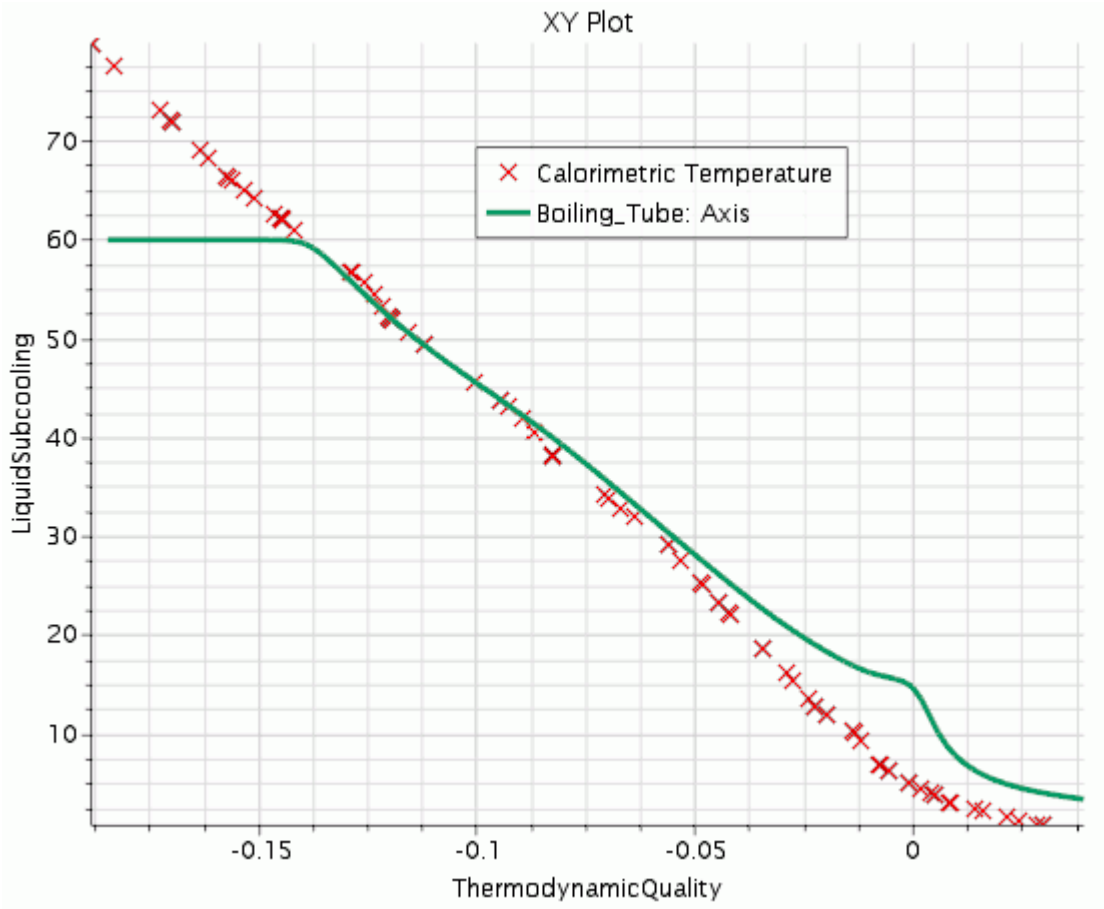
The STAR-CCM+ results deviate from this line when vapor is generated. This result is as expected from a heat balance, because if vapor coexists in subcooled conditions (thermodynamic quality < 0), liquid must be below the temperature that is expected for a homogeneous mixture.

Now consider the liquid subcooling along the axis. The experimental data indicates that there was already a temperature gradient upstream of the heated section.

The source reference [Bartolemei (1967) [800]] gives no details here, but you are able to proceed by assuming that the simulation can model a short length of thermal boundary layer development before any

boiling begins. This tutorial also uses the same thermodynamic quality scale to compare experiment and simulation, so that after some distance any differences due to inlet conditions become irrelevant.

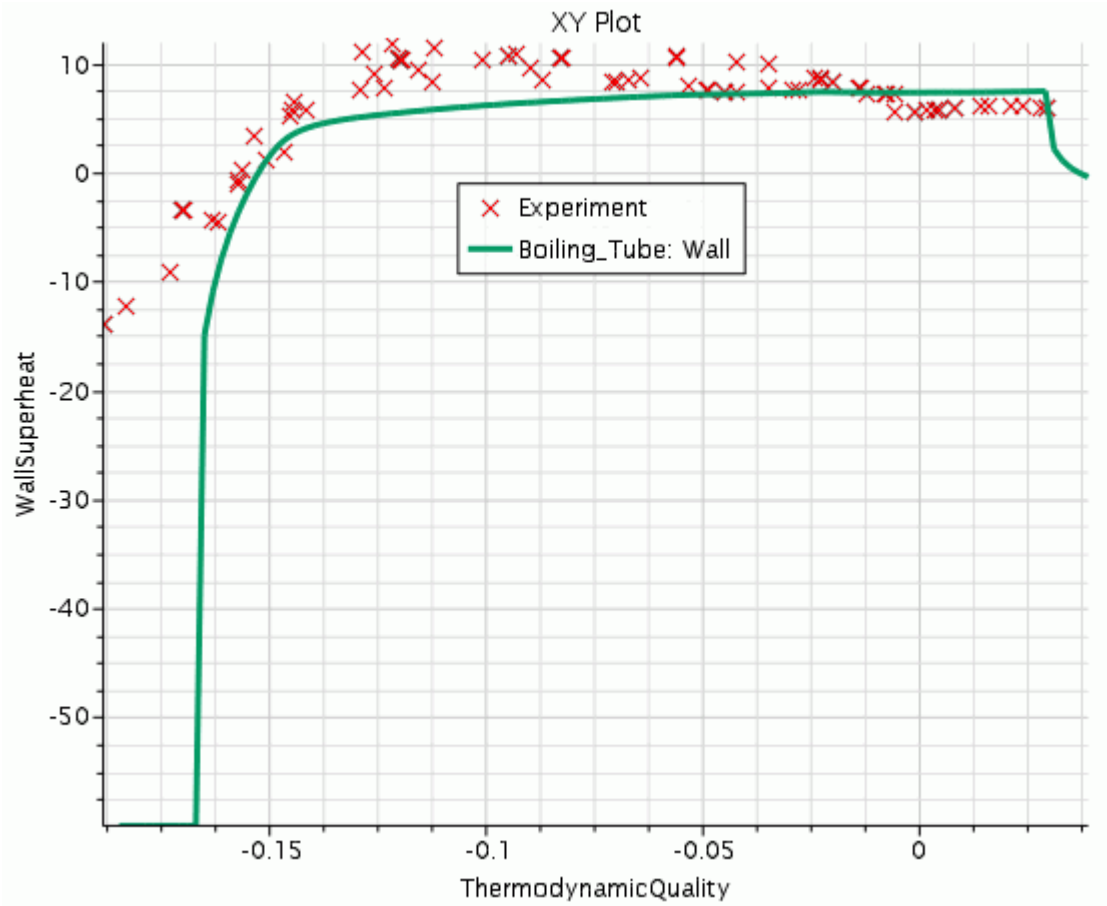
The *Axis Subcooling* plot is shown below:



Here, as expected, the liquid subcooling is still above zero as the zero thermodynamic quality line is crossed. Heat is going into vapor production near the wall instead of heating the liquid core. The over prediction that STAR-CCM+ makes of this core subcooling is consistent with its over prediction of outlet vapor.

Finally compare the predicted and experimental wall superheat. Bartolemei (1967) [800] temperature measurements are used on the outside of the boiling tube, converted to inner face temperatures using Bartolemei's own estimate of temperature drop in the tube wall.

The *Wall Superheat* plot is shown below:



It is clear that there was heating ahead of the specified heated section in the experiment, while STAR-CCM+ takes a little distance to develop a thermal boundary layer. After this distance, the agreement is good at the start of boiling (zero superheat) and at fully developed boiling nearer the outlet where the superheat is of order 6K. In between, in the mixed convection/evaporation region, the current model underestimates superheat.

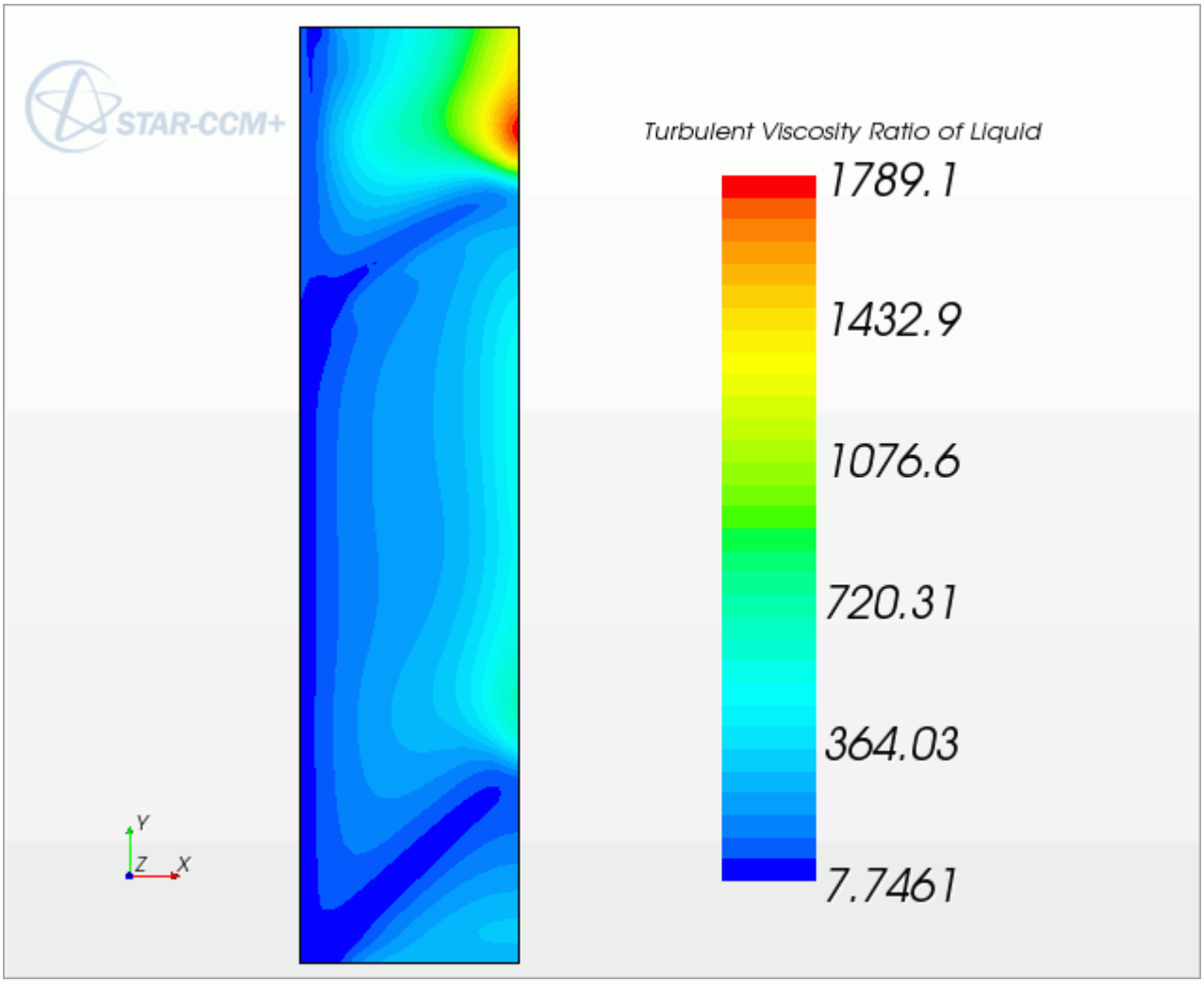
Again this model could be tuned to improve the superheat for this particular case (Bartolemei does not provide temperature data for any other conditions). However, it is good to see that this model could be useful for fixed-temperature applications as well as for fixed flux applications.

Turbulence Checks

Examine the results to check the turbulence profile.

If you use the Scalar Scene to examine the Turbulent Viscosity Ratio of Liquid, you see that there are two troughs in turbulence across the flow.

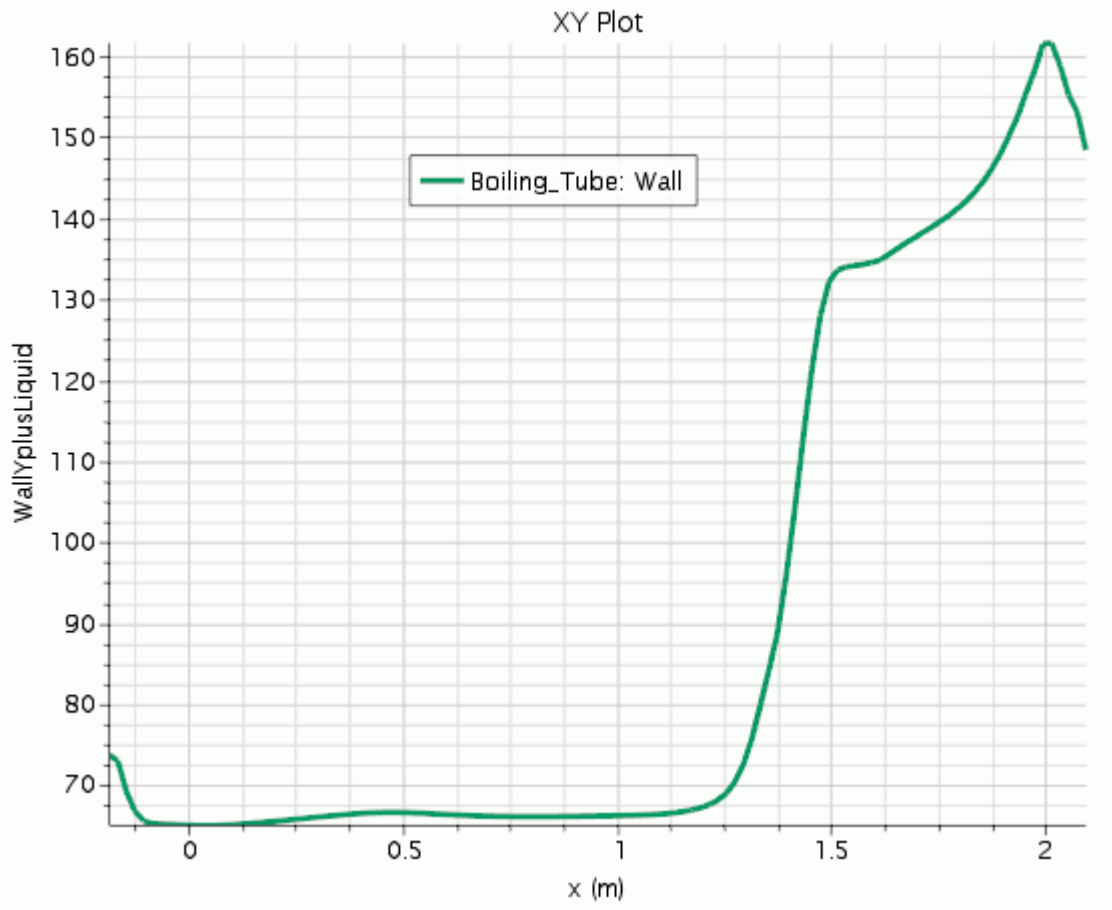
The Scalar Scene is shown below:



The first occurs at the inlet and is entirely due to the use of flat velocity and turbulence profiles at the inlet. The second is due to the acceleration of the flow due to the onset of vapor production. The point here is to check that the first artificial feature is not interfering with the prediction of onset of boiling nor with the onset of vapor production. This scenario is the case here since the artificial first feature is well downstream of the inception of wall boiling. Under different conditions, the right thing to do would be to replace the flat inlet profiles by fully developed profiles of velocity, turbulence, and perhaps temperature.

The second thing to check is that the predicted value of y+ at the first cell center next to the wall is well above 30.

The *Wall Yplus of Liquid* plot is shown below:



The computed y+ value confirms that a high-y+ wall treatment could have been used for this calculation. The two-layer treatment is retained so that this same model can be used under a wider range of conditions. The wall plot shows that y+ starts around 65 in the single phase region and then increases as the flow accelerates with increasing vapor content.

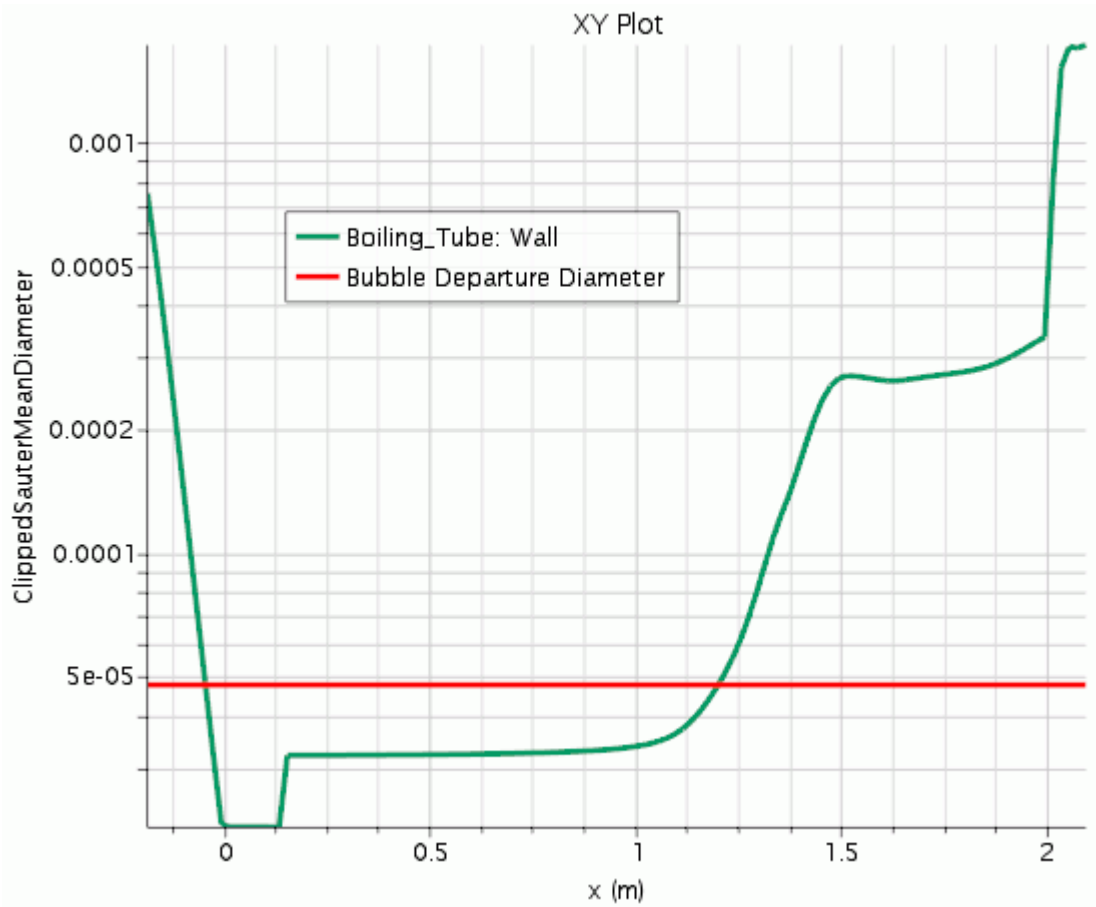
Additional Information

Examine the detailed results for specific models.

If you are interested, you can inspect the more detailed results for various submodels.

Perhaps the most important of these results is the bubble size prediction next to the wall shown in plot Near-Wall Bubble size. Bubbles depart with a size of about 50 micron under these conditions. Where the water is still heavily subcooled, the S-Gamma model predicts that bubble size next to the wall is immediately smaller due to condensation. Once the liquid subcooling is small enough (liquid temperature is approaching saturation) the local condensation rate can no longer balance wall evaporation rate. At this point on the wall, net vapor production begins, and also bubble size starts to rise due to coalescence.

The Near-Wall Bubble Size plot is shown below:



Some other points of interest to boiling modeling are:

- Near wall volume fraction reaches levels of 0.8 near the wall in this calculation. This result might not be physical, and could be modified with model adjustments. However, due to the difficulties of making measurements across narrow tubes operating at very high pressure, the Bartolemei (1967) ([800]) and Bartolemei (1982) ([801]) data provides no information to support such adjustments to radial distribution.
- Bubble departure frequency is a simple model that essentially depends on departure size, so is also constant in this constant property calculation.
- Bubble Influence Wall Area Fraction shows how the heat flux between the wall and the liquid phase is partitioned between fully developed convective heat transfer and bubble-induced quenching heat flux. Initially this fraction is small as bubble nucleation sites are far apart, but rises as the number of active sites increases with wall superheat.
- The nucleation site number density plot shows a rapid change in number of active sites with very small change in wall superheat. It is this behavior at the heart of the wall boiling model that both enhances heat transfer rapidly and keeps the wall superheat relatively uniform along a boiling wall.

Summary

This tutorial validated the wall boiling model in STAR-CCM+ against experimental void and temperature data for high-pressure forced convective boiling.

This tutorial has covered the following:

- Selection of suitable models for Eulerian simulation of forced convective boiling at high pressure.
- Definition of phase enthalpies that are based on steam table conventions.
- Selection of constant properties that give the same mass and energy balance as variable properties.
- Monitoring of min/max phase velocities, temperatures, and volume fractions as an indicator of relaxation needs in a sensitive calculation.
- Use of wall heat flux ramping during flow development to keep velocities and temperatures bounded.
- Verification of phase mass balances with interphase mass transfer.
- Methods for extracting results for axial void, liquid subcooling, wall superheat, and wall heat flux partitioning.
- Use of the thermodynamic quality scale to compare results against experiment.
- Extensive use of field functions and macros to scale and analyze the results.

Bibliography

This tutorial references the following papers.

[800] *Bartolomei, G. G. and Chanturiya, V. M. 1967 “Experimental Study of True Void Fraction When Boiling Subcooled Water in Vertical Tubes” in Thermal Engineering, 14(2).*

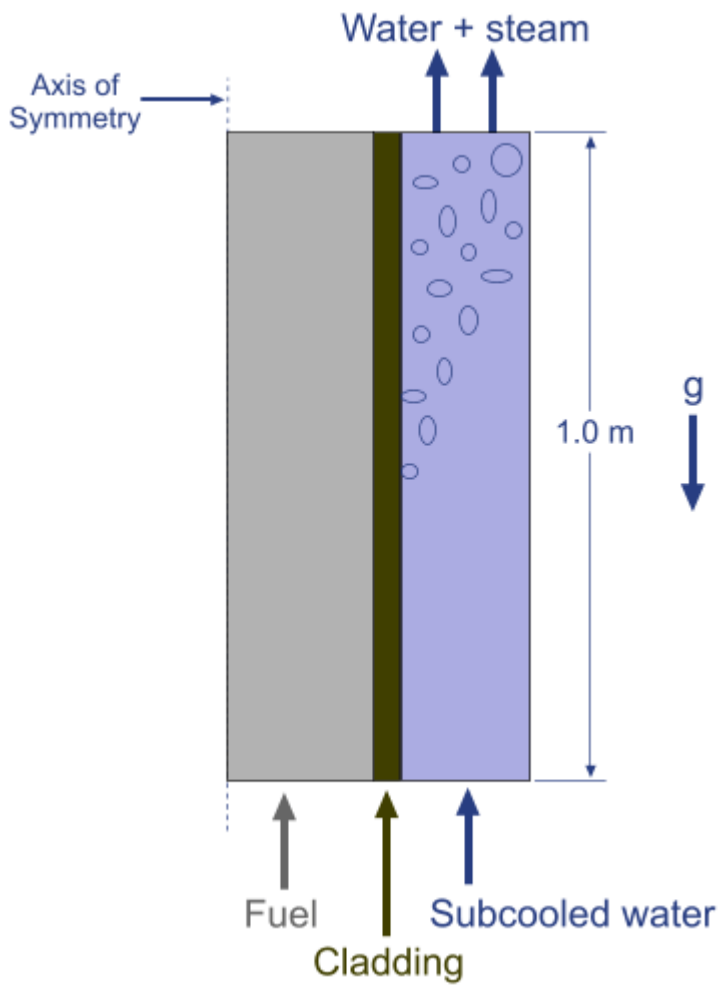
[801] *Bartolomei, G. G. et al 1982 “An Experimental Investigation of True Volumetric Vapor Content with Subcooled Boiling in Tubes” in Thermal Engineering, 29(3).*

Eulerian: Conjugate Heat Transfer Wall Boiling

This tutorial demonstrates how to set up a conjugate heat transfer wall boiling analysis in STAR-CCM+. It simulates a nuclear fuel pin that is cooled using subcooled boiling under flow conditions typical in the first meter of a 3.7-m long BWR6 reactor channel.

The channel operates at 71.7 bar with an average heat flux of 524900 W/m^2 at the cladding surface.

An axisymmetric, two-dimensional volume mesh of the boiling tube is provided, as shown below.



The fuel, cladding, and coolant are modeled as three finite regions. There is a gap between the fuel and the cladding. The properties of this gap vary as the fuel reacts and daughter constituents fill the space between the fuel and the cladding. Therefore, the gap is treated as an empirical contact resistance, with its thickness added to the fuel.

Contents:

Prerequisites

Importing the Mesh and Naming the Simulation
Visualizing the Mesh
Selecting the Physics Models
Setting the Solid Material Properties
Creating Phases and Selecting Phase Models
Defining the Phase Interactions
Setting the Fuel-Cladding Interface Properties
Setting Region Sources and Boundary Conditions
Setting Initial Conditions and Reference Values
Setting the Stopping Criteria
Visualizing the Temperature and Vapor Fraction
Running the Simulation
Visualizing the Results
Validating the Results
Summary

Prerequisites

The instructions in the Eulerian: Conjugate Heat Transfer Wall Boiling tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Importing the Mesh and Naming the Simulation

Launch STAR-CCM+, import the supplied geometry, and save the simulation.

To import the mesh and name the simulation:

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh**.
4. In the *Open* dialog, choose `doc/startutorialsdata/multiphaseFlow/data/WallBoilingCht.ccm`, then click **Open**.

   STAR-CCM+ provides feedback on the import process, which takes a few seconds, in the *Output* window.
5. Save the simulation as `wallBoilingCHT.sim`.

Visualizing the Mesh

Create a mesh scene to view the mesh.

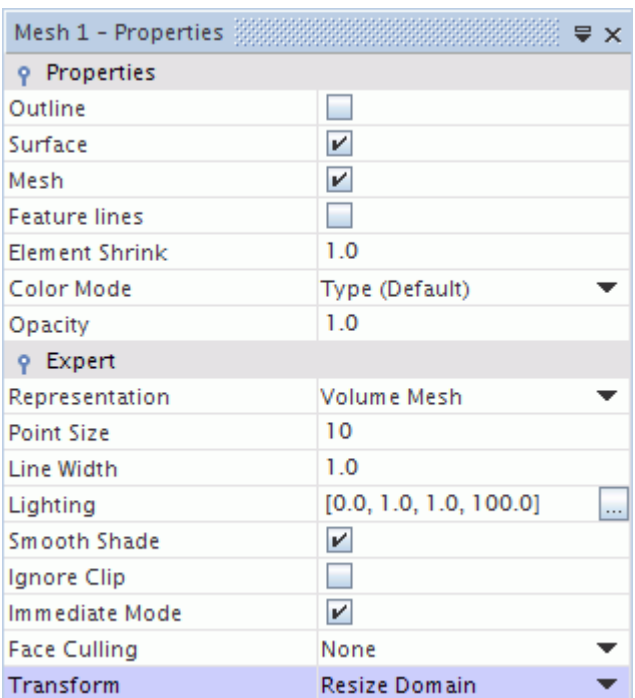
An axisymmetric mesh of the fuel pin, cladding, and coolant has been prepared for this analysis, so further meshing is not required. The boundary names and types are as defined in the imported mesh. View the mesh to examine these boundaries.

To visualize the mesh:

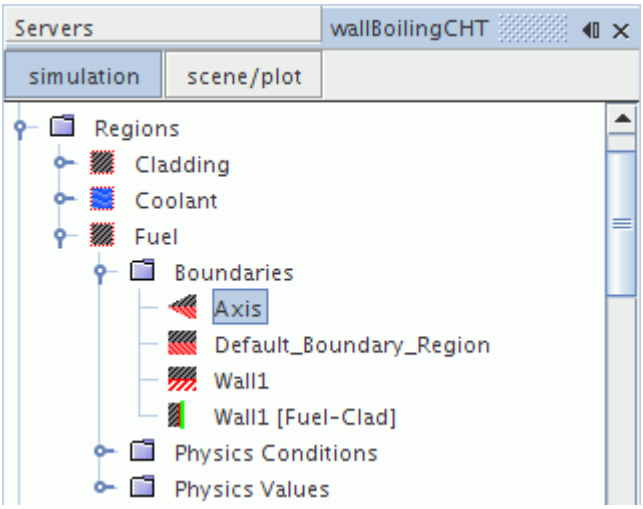
1. Create a mesh scene.

The fuel pin is 1 m long in the X direction and 0.01 m in the Y direction. To visualize the scene with an improved aspect ratio, create a simple transform to stretch the domain in the Y direction:

2. Right-click **Tools** > **Transforms** and select **New Graphics Transform** > **Simple Transform**.
3. Rename the newly created **Simple Transform 1** node to `Resize Domain`.
4. In the *Properties* window, set the **Scale** property to `[1.0, 20.0, 1.0]`.
5. To apply this transform to the mesh scene, select **Scenes** > **Mesh Scene 1** > **Displayers** > **Mesh 1** and set its **Transform** expert property to **Resize Domain**.

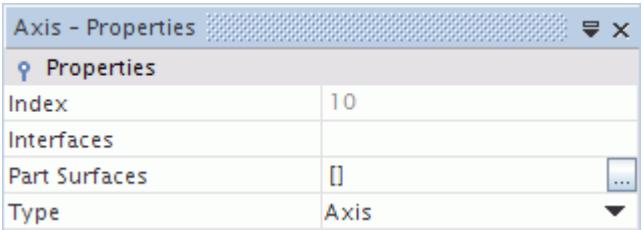


6. To examine the imported boundaries, expand the **Regions** > **Fuel** > **Boundaries** node and select the **Axis** node.

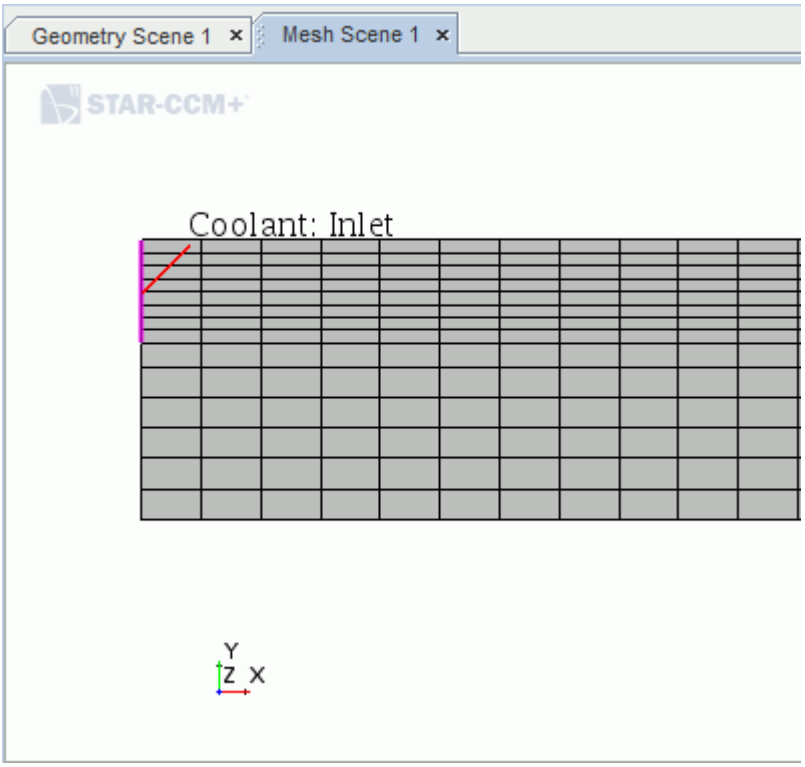


This boundary is highlighted in **Mesh Scene 1**.

7. In the *Properties* window, make sure that the **Type** is set to **Axis**.



8. Similarly, select the **Coolant** > **Boundaries** > **Inlet** node and observe which part is highlighted in the scene.



9. Save the simulation.

Selecting the Physics Models

Physics models define the primary variables of the simulation, including pressure, temperature, velocity, and what mathematical formulation is used to generate the solution.

Three continua are set up to model each of the three regions.

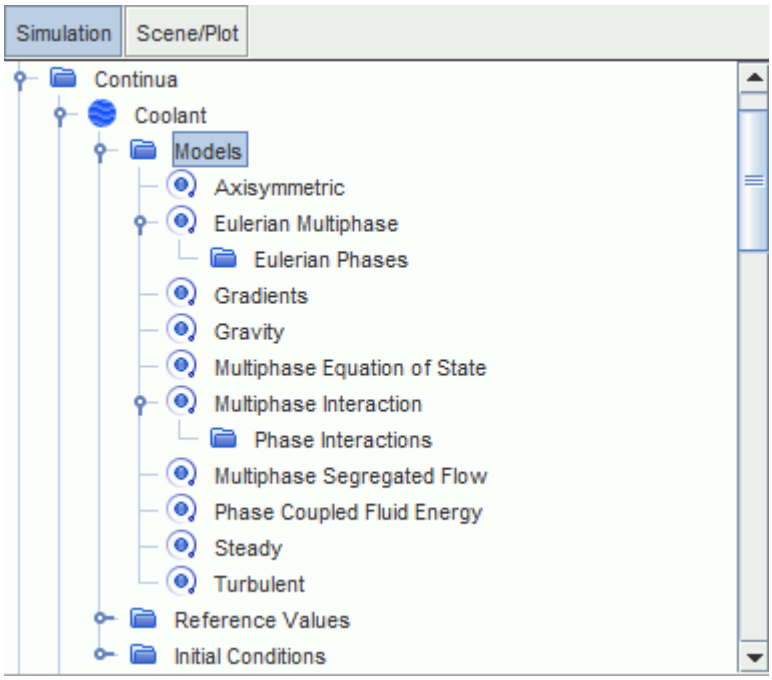
To set up the physics models for the coolant:

1. For the physics continuum, **Continua** > **Physics 1**, in the *Enabled Models* group box, deactivate **Two-Dimensional**.
2. Select the following models in order:

Group Box	Model
Space	**Axisymmetric**
Time	**Steady**
Material	**Eulerian Multiphase** **Multiphase Interaction** (Selected automatically)
Eulerian Multiphase Model	**Multiphase Segregated Flow** **Multiphase Equation of State** (Selected automatically) **Gradients** (Selected automatically)

Group Box	Model
Viscous Regime	**Turbulent**
Optional Models	**Gravity** **Phase Coupled Fluid Energy**

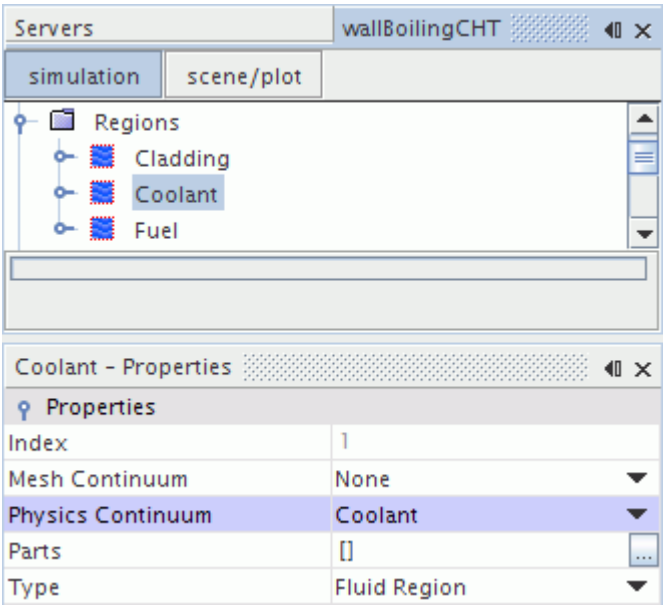
3. Click **Close**.
4. Rename the **Physics 1** node to **Coolant**.
5. To review the models, open the **Coolant** > **Models** node.



6. Save the simulation.

To assign the **Coolant** continuum to the **Coolant** region:

7. Select the **Regions** > **Coolant** node and set its **Physics Continuum** to **Coolant**.



To set up the physics models for the cladding and fuel:

8. For the physics continuum, **Continua** > **Physics 2**, in the *Enabled Models* group box, deactivate **Two-Dimensional**.
9. Select the following models in order:

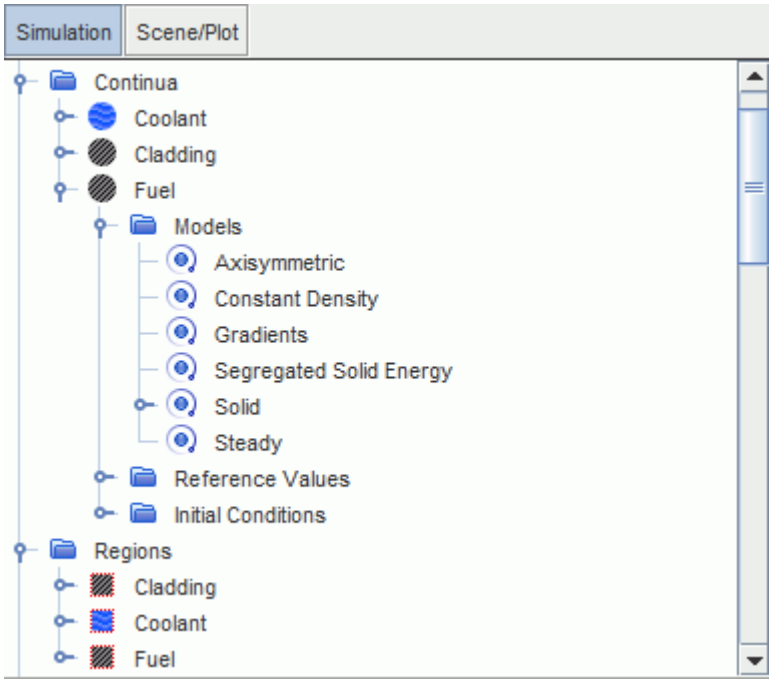
Group Box	Model
Space	**Axisymmetric**
Time	**Steady**
Material	**Solid**
Optional Models	**Segregated Solid Energy** **Gradients** (Selected automatically)
Equation of State	**Constant Density**

10. Click **Close**.

The physics models for the cladding and fuel are identical. However, since different material properties are used to model each region, two separate continua are created for the cladding and fuel:

11. Right-click **Continua** > **Physics 2** and select **Copy**.
12. Right-click **Continua** and select **Paste**.
13. Rename **Copy of Physics 2** to **Cladding**.
14. Rename **Physics 2** to **Fuel**.
15. Assign these two continua to their respective regions.

    The icons of the **Fuel** and **Cladding** regions turn to gray to indicate that they are both solid regions.



16. Save the simulation.

Setting the Solid Material Properties

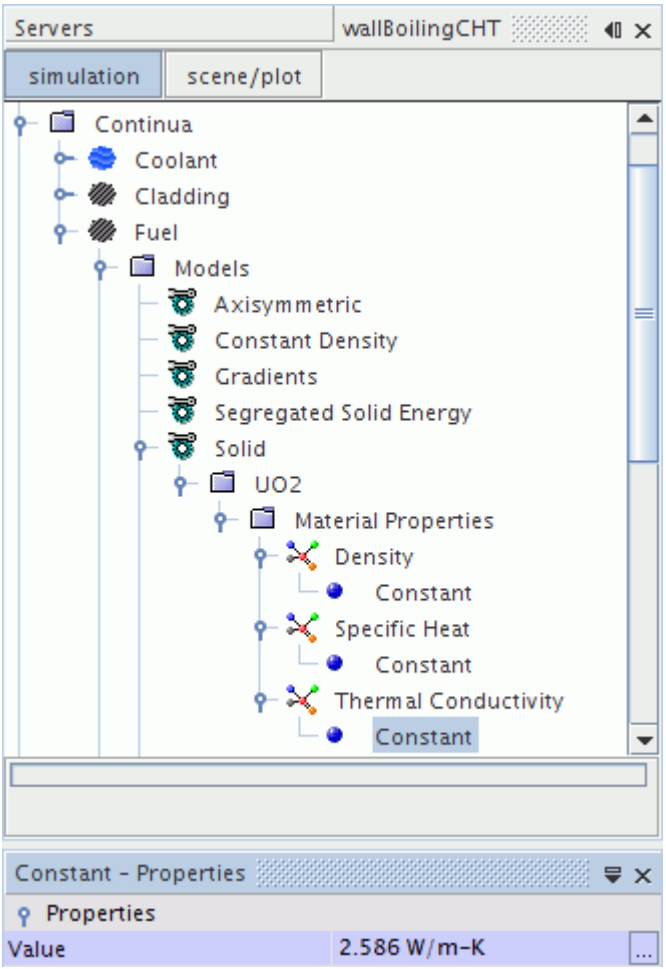
Define the properties for the solid materials.

To set the properties for the solid materials:

Specify the material properties for the fuel.

1. Rename the **Continua** > **Fuel** > **Models** > **Solid** > **Al** node to **UO2**.
2. Edit the **UO2** > **Material Properties** node and set the following properties:

Node	Property	Setting
Density > **Constant**	**Value**	`10420.0 kg/m^3`
Specific Heat > **Constant**	**Value**	`327.7 J/kg-K`
Thermal Conductivity > **Constant**	**Value**	`2.586 W/m-K`



Specify the material properties for the cladding.

3. Rename the **Cladding** > **Models** > **Solid** > **Al** node to **Zircaloy**.
4. Edit the **Zircaloy** > **Material Properties** node and set the following properties:

Node	Property	Setting
Density > **Constant**	**Value**	`6579.0 kg/m^3`
Specific Heat > **Constant**	**Value**	`314.5 J/kg-K`
Thermal Conductivity > **Constant**	**Value**	`15.61 W/m-K`

5. Save the simulation.

Creating Phases and Selecting Phase Models

Create the phases and select the appropriate phase models.

The two phases flowing through the coolant section of vertical pipe are water and steam. These phases are defined by creating them under the Eulerian Phases model, and selecting the corresponding material properties.

To create the phases and select the phase models:

Create the liquid phase.

1. In the **Coolant** continuum, right-click the **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and create a phase.
2. Rename the **Phase 1** node to **Liquid**.
3. For the **Liquid** phase, select the following models:

Group Box	Model
Enabled Models	**Flow** (Pre-selected) **Turbulent** (Pre-selected)
Material	**Liquid** **Reynolds-Averaged Navier-Stokes** (Selected automatically)

4. Deactivate **Auto-select recommended models**.
5. Select the following models in order:

Group Box	Model
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**
K-Epsilon Turbulence Modelss	**Standard K-Epsilon** **Exact Wall Distance** (Selected automatically)

6. Activate **Auto-select recommended models**.
7. Select the following models in order:

Group Box	Model
Enabled Models	**High y+ Wall Treatment** (Selected automatically)
Equation of State	**Constant Density**
Energy	**Segregated Fluid Temperature**

8. Click **Close**.
9. Rename the **Liquid** > **Models** > **Liquid** > **H2O** node to **H2O (l)**.

Create the gas phase.

10. Create a second phase and rename it to **Vapor**.
11. For the **Vapor** phase, select the following models:

Group Box	Model
Enabled Models	**Flow** (Pre-selected) **Turbulent** (Pre-selected)
Material	**Gas** **Reynolds-Averaged Navier-Stokes** (Selected automatically)

12. Deactivate **Auto-select recommended models**.
13. Select the following models in order:

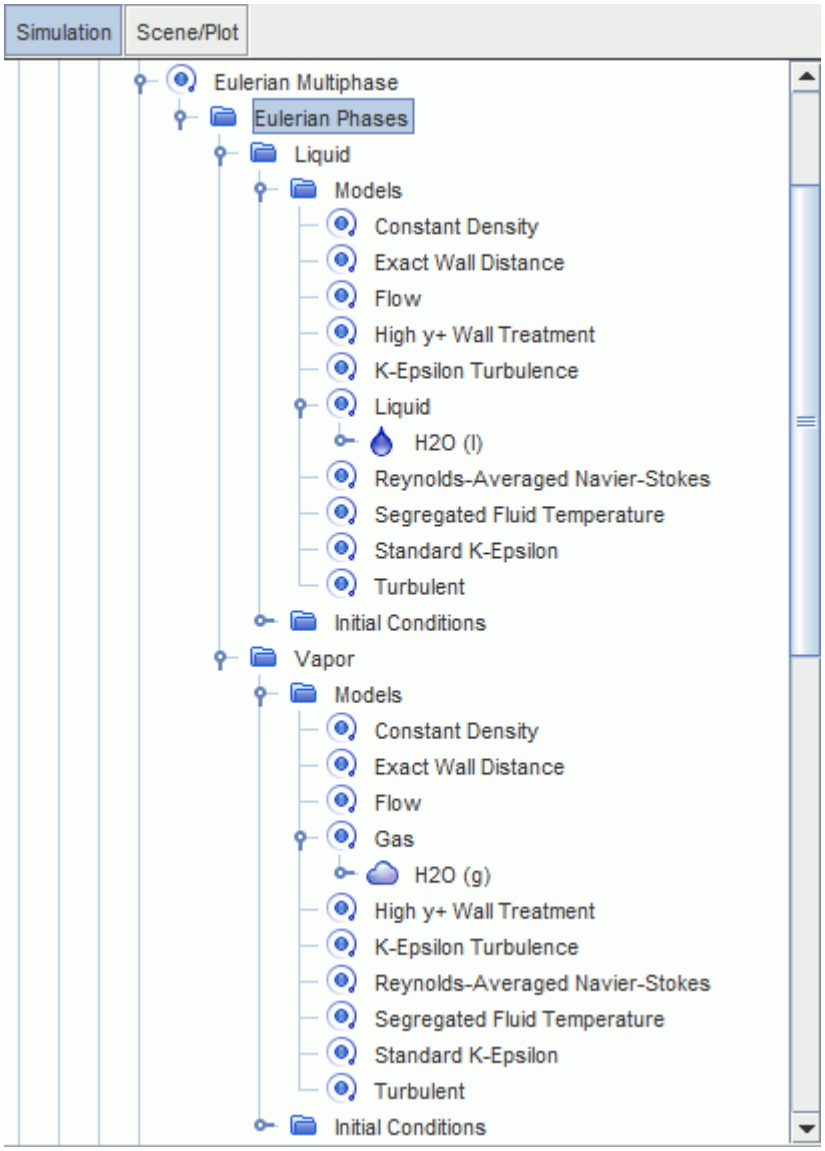
Group Box	Model
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**
K-Epsilon Turbulence Models	**Standard K-Epsilon** **Exact Wall Distance** (Selected automatically)

14. Activate **Auto-select recommended models**.
15. Select the following models in order:

Group Box	Model
Enabled Models	**High y+ Wall Treatment** (Selected automatically)
Equation of State	**Constant Density**
Energy	**Segregated Fluid Temperature**

16. Click **Close**.
17. Rename the **Vapor** > **Models** > **Gas** > **Air** node to **H2O (g)**.

    The **Eulerian Phases** node appears as shown.



The material properties for each phase are now modified, starting with the water phase.

18. Edit the **Liquid** > **Models** > **Liquid** > **H20 (l)** > **Material Properties** node and set the following properties:

Node	Property	Setting
Density > **Constant**	**Value**	`745.65 kg/m^3`
Dynamic Viscosity > **Constant**	**Value**	`9.253E-5 Pa-s`
Heat of Formation > **Constant**	**Value**	`1276100 J/kg`
Specific Heat > **Constant**	**Value**	`5310.0 J/kg-K`
Standard State Temperature > **Constant**	**Value**	`560.65 K`
Thermal Conductivity > **Constant**	**Value**	`0.574 W/m-K`

Next set the properties of the vapor phase:

19. Edit the **Vapor** > **Models** > **Gas** > **H2O (g)** > **Material Properties** node and set the following properties:

Node	Property	Setting
Density > **Constant**	**Value**	`37.52 kg/m^3`
Dynamic Viscosity > **Constant**	**Value**	`1.9034E-5 Pa-s`
Heat of Formation > **Constant**	**Value**	`2770300 J/kg`
Specific Heat > **Constant**	**Value**	`5440 J/kg-K`
Standard State Temperature > **Constant**	**Value**	`560.65 K`
Thermal Conductivity > **Constant**	**Value**	`0.06514 W/m-K`

20. Save the simulation.

Defining the Phase Interactions

Define the interactions between the phases.

Using the **Multiphase Interaction** model, you can define the interaction between the liquid and vapor phases.

To define the phase interaction:

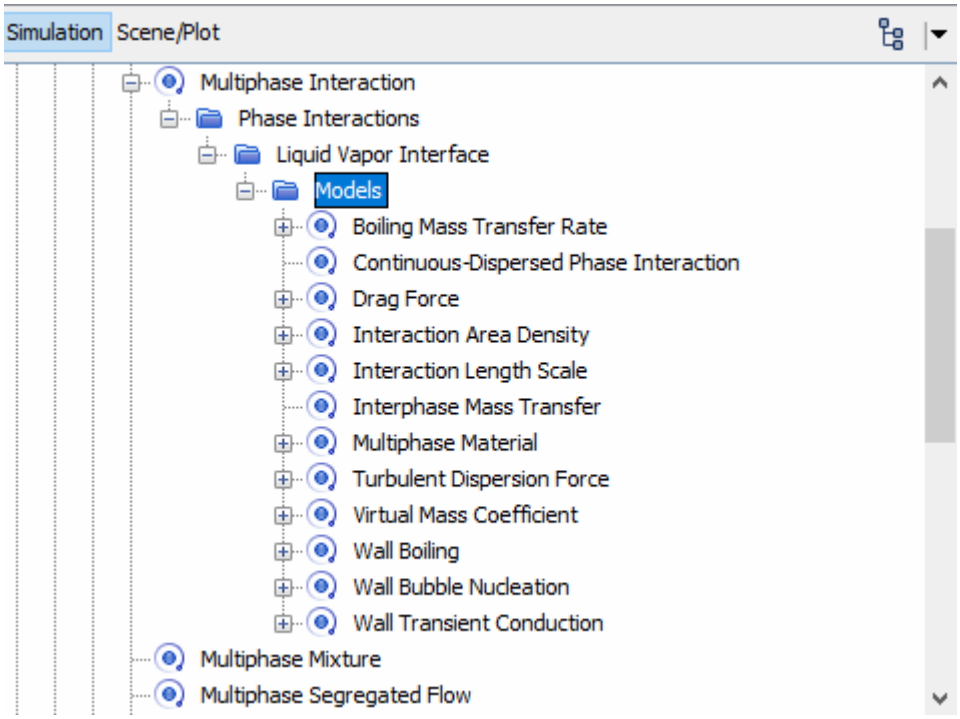
1. In the **Coolant** continuum, right-click the **Models** > **Multiphase Interaction** > **Phase Interactions** node and create a phase interaction.
2. Rename the **Phase Interaction 1** node to **Liquid Vapor Interface**.
3. For the **Liquid Vapor Interface** phase interaction, select the following models in order:

Group Box	Model
Phase Interaction Topology	**Continuous-Dispersed Phase Interaction**
Enabled Models	**Drag Force** (selected automatically) **Multiphase Material** (selected automatically) **Interaction Length Scale** (selected automatically) **Interaction Area Density** (selected automatically)
Optional Models	**Turbulent Dispersion Force** **Virtual Mass Coefficient** (selected automatically)
Optional Models	**Interphase Mass Transfer**
Interphase Mass Transfer Rate	**Boiling Mass Transfer Rate**
Optional Models	**Wall Boiling** **Wall Bubble Nucleation** (selected automatically)

Group Box	Model
	Wall Transient Conduction (selected automatically)

4. Click **Close**.

   The **Liquid Vapor Interface** > **Models** node is shown below:



Define the phase interaction properties, starting with the definition of the continuous and dispersed phases.

5. Edit the **Liquid Vapor Interface** > **Models** node and set the following properties:

Node	Property	Setting
Continuous-Dispersed Phase Interaction	**Continuous Phase**	**Liquid**
	Dispersed Phase	**Vapor**

Select a more appropriate interaction length scale model.

6. Select the **Liquid Vapor Interface** > **Models** > **Interaction Length Scale** > **Interaction Length Scale** node and set the **Method** property as **Kurul Podowski** .

Define the remainder of the phase interaction properties.

7. Edit the **Liquid Vapor Interface** > **Models** node and set the following properties:

Node	Property	Setting
Turbulent Dispersion Force	**Use Logarithmic Form**	Activated

Node	Property	Setting
Boiling Mass Transfer Rate > **Continuous Phase Nusselt Number**	**Method**	**Ranz-Marshall**
Multiphase Material > **Multiphase Material** > **Material Properties** > **Boiling Temperature** > **Constant**	**Value**	`560.65 K`
Multiphase Material > **Multiphase Material** > **Material Properties** > **Surface Tension** > **Constant**	**Value**	`0.01725 N/m`
Drag Force > **Drag Coefficient**	**Method**	**Tomiyama**

The logarithmic form of the **Turbulent Dispersion Force** is used to improve the convergence rate. When the volume fraction of one phase is low and varies rapidly, the logarithmic form is differenced more accurately than the standard form.

Selecting the **Ranz-Marshall** option makes STAR-CCM+ calculate the Nusselt number using the Ranz-Marshall correlation. The Nusselt number is used to calculate the heat transfer coefficient for the heat transfer from the bulk water to the bubble interface.

The interface temperature is the saturation temperature of the liquid phase.

The **Tomiyama** model is appropriate for modeling a small bubble in a low viscosity, air-water system.

8. Save the simulation.

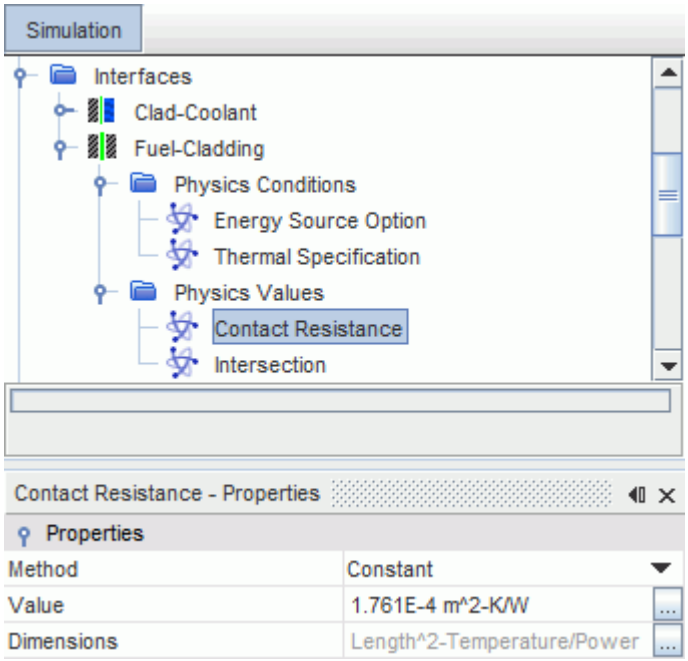
Setting the Fuel-Cladding Interface Properties

Define the Fuel-Cladding interface contact resistance.

You set the **Fuel-Cladding** interface contact resistance which models the gap between the two regions.

To set the Fuel-Cladding Interface properties:

1. Select the **Interfaces** > **Fuel-Cladding** > **Physics Values** > **Contact Resistance** node and set **Value** to `1.761E-4 m^2-K/W`.



2. Save the simulation.

Setting Region Sources and Boundary Conditions

Define the boundary types and specify the appropriate property values.

Conditions are specified in the fuel region, and on the inlet and outlet boundaries of the coolant.

To set the boundary conditions:

Specify the volume energy source in the fuel.

1. Edit the **Regions** > **Fuel** node and set the following properties:

Node	Property	Setting
Physics Conditions > **Energy Source Option**	**Method**	**Volumetric Heat Source**
Physics Values > **Volumetric Heat Source**	**Value**	`2.26412E8 W/m^3`

Set the **Inlet** boundary conditions.

2. Edit the **Regions** > **Coolant** > **Boundaries** > **Inlet** > **Phase Conditions** node and set the following properties:

Node	Property	Setting
Liquid > **Physics Values**		
∟ **Velocity Magnitude**	**Value**	`2.0076`

Node	Property	Setting
└ **Volume Fraction**	**Value**	0.9999
Vapor > **Physics Values**		
└ **Velocity Magnitude**	**Value**	2.0076
└ **Volume Fraction**	**Value**	1.0E-4

3. Edit the **Coolant** > **Boundaries** > **Inlet** node and set the following properties:

Node	Property	Setting
Phase Conditions > **Liquid** > **Physics Values** > **Static Temperature**	**Value**	551.55 K
└ **Vapor** > **Physics Values** > **Static Temperature**	**Value**	560.65 K
Phase Conditions > **Liquid** > **Physics Values** > **Turbulence Intensity**	**Value**	0.05
└ **Turbulent Viscosity Ratio**	**Value**	400
Phase Conditions > **Vapor** > **Physics Values** > **Turbulence Intensity**	**Value**	0.05
└ **Turbulent Viscosity Ratio**	**Value**	200

Set the **Outlet** boundary conditions. At the outlet boundary, the volume fraction for each phase is extrapolated from the interior of the domain.

4. Edit the **Coolant** > **Boundaries** > **Outlet** node and set the following properties:

Node	Property	Setting
Physics Conditions > **Volume Fraction Specification**	**Option**	**Extrapolation**
Phase Conditions > **Liquid** > **Physics Values** > **Static Temperature**	**Value**	560.65 K
└ **Vapor** > **Physics Values** > **Static Temperature**	**Value**	560.65 K
Phase Conditions > **Liquid** > **Physics Values** > **Turbulence Intensity**	**Value**	0.05
└ **Turbulent Viscosity Ratio**	**Value**	400

Node	Property	Setting
Phase Conditions > **Vapor** > **Physics Values** > **Turbulence Intensity**	**Value**	0.05
└ **Turbulent Viscosity Ratio**	**Value**	200

5. Save the simulation.

Setting Initial Conditions and Reference Values

Set the initial conditions and reference values for the physics continuum.

Set the reference gravity, density, and pressure for the coolant.

To set the initial conditions and reference values:

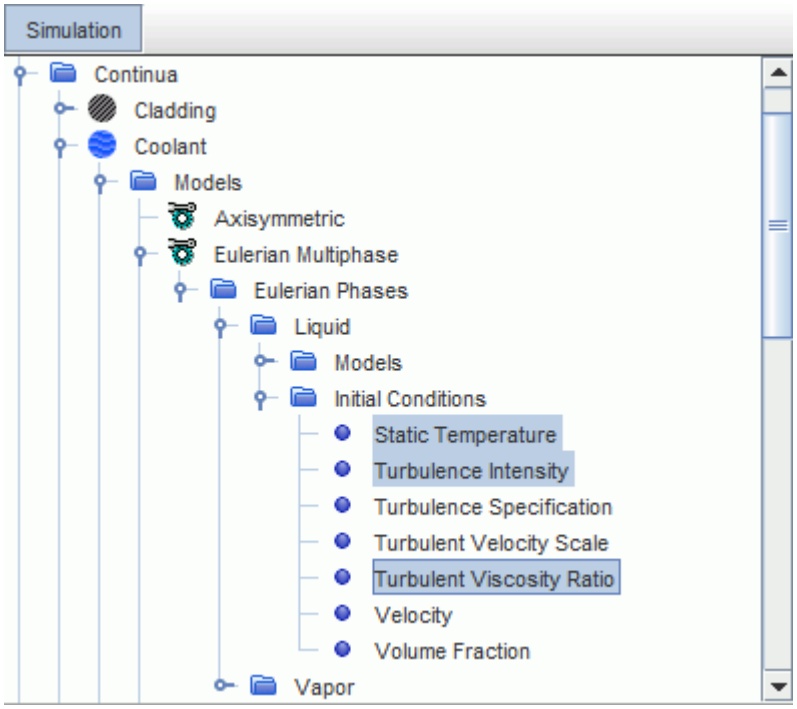
1. Edit the **Coolant** > **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and set the following properties:

Node	Property	Setting
Liquid > **Initial Conditions**		
└ **Velocity**	**Value**	[2.0076, 0.0] m/s
└ **Volume Fraction**	**Value**	0.9999
Vapor > **Initial Conditions**		
└ **Velocity**	**Value**	[2.0076, 0.0] m/s
└ **Volume Fraction**	**Value**	1.0E-4

Set the initial conditions for each phase.

2. Edit the **Models** > **Eulerian Multiphase** > **Eulerian Phases** > **Liquid** > **Initial Conditions** node and set the following properties:

Node	Property	Setting
Static Temperature	**Value**	551.55 K
Turbulence Intensity	**Value**	0.05
Turbulent Viscosity Ratio	**Value**	400



3. Edit the **Models** > **Eulerian Multiphase** > **Eulerian Phases** > **Vapor** > **Initial Conditions** node and set the following properties:

Node	Property	Setting
Static Temperature	**Value**	`560.65 K`
Turbulence Intensity	**Value**	`0.05`
Turbulent Viscosity Ratio	**Value**	`200`

Set the reference values for the **Coolant** continuum.

4. Edit the **Reference Values** node and set the following properties:

Node	Property	Setting
Gravity	**Value**	`[-9.81, 0.0] m/s^2`
Reference Density	**Value**	`745.65 kg/m^3`
Reference Pressure	**Value**	`7170000.0 Pa`

5. Save the simulation.

Setting the Stopping Criteria

Set the appropriate stopping criteria for the simulation.

The simulation runs for 200 iterations.

To set the stopping criteria:

1. Edit the **Stopping Criteria** node and set the following property:

Node	Property	Setting
Maximum Steps	**Enabled**	Activated
	Maximum Steps	200

2. Save the simulation.

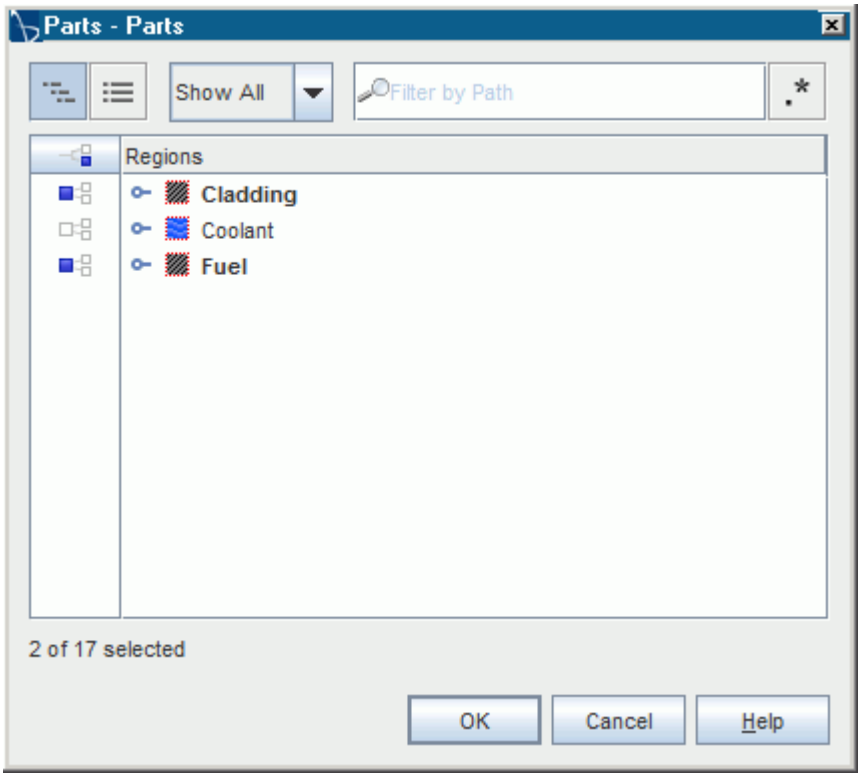
Visualizing the Temperature and Vapor Fraction

Create a scalar scene to view the results of the simulation.

You now create a scalar scene that displays the temperature of the fuel and cladding, and the coolant vapor volume fraction. Both field functions can be monitored while the solver is running.

To visualize the temperature and vapor fraction:

1. Create a scalar scene.
2. Rename **Scenes** > **Scalar Scene 1** to **Dual Scene**.
3. Display the temperature in the fuel and cladding regions:
   a) Select the **Dual Scene** > **Displayers** > **Scalar 1** > **Parts** node.
   b) In the *Properties* window, click ... (**Custom Editor**) for the **Parts** property.
   c) In the *Parts - Parts* dialog, right-click on a blank area and select **Deselect All**.
   d) Select the **Cladding** and **Fuel** regions.



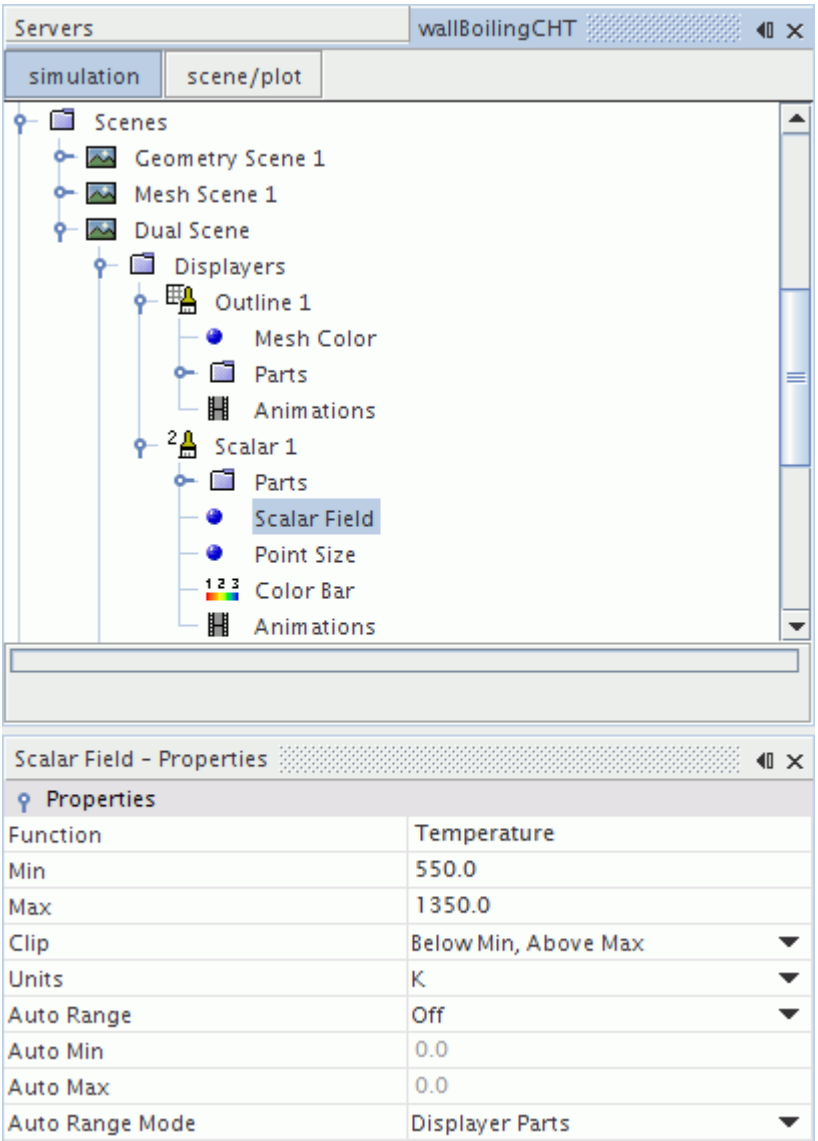
e) Click **OK**.

f) Select the **Scalar 1** > **Scalar Field** node and set its **Function** property to **Temperature**.

g) Set the **Auto Range** property to **Off**.

h) Set the **Min** property to `550.0`.

i) Set the **Max** property to `1350.0`.



4. Create a new scalar displayer for the coolant:

a) Right-click the **Dual Scene** > **Displayers** node and select **New Displayer** > **Scalar**.

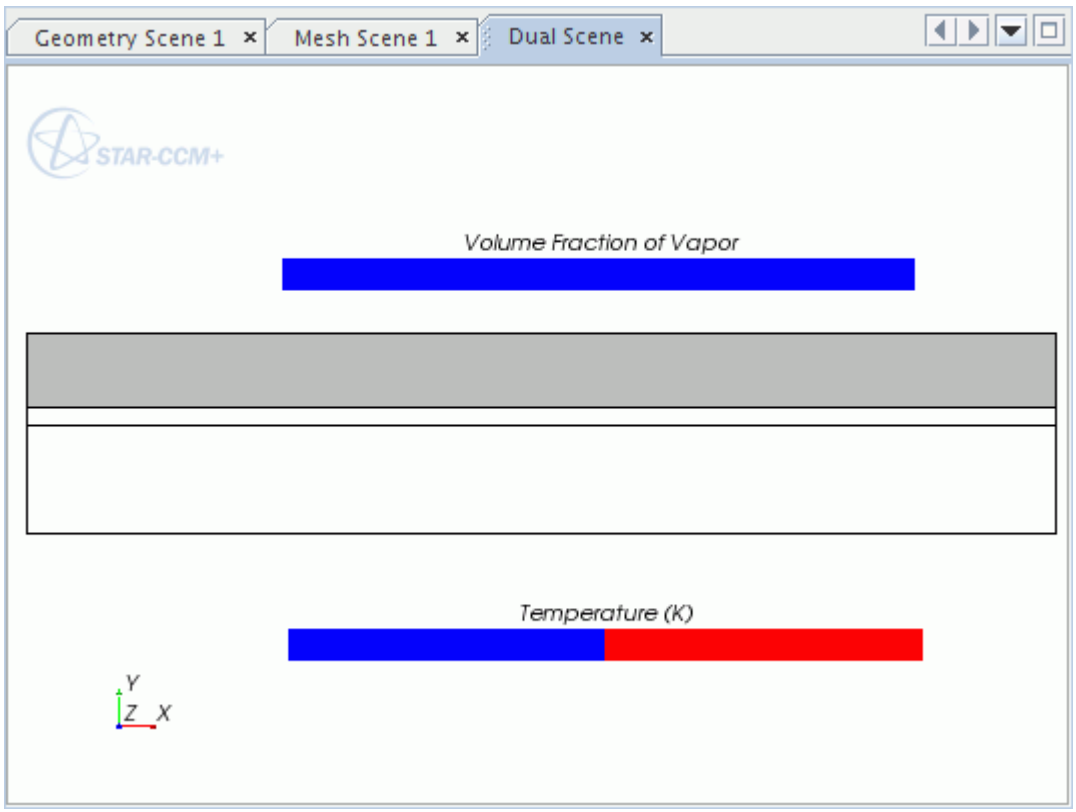
b) Select the **Scalar 2** > **Parts** node.

c) In the *Properties* window, click ... (**Custom Editor**) for the **Parts** property.

d) Select **Coolant** and click **OK**.

e) Select **Scalar 2** > **Scalar Field** and set its **Function** property to **Volume Fraction** > **Volume Fraction of Vapor**.

5. Apply the **Resize Domain** transform to **Outline 1**, **Scalar 1** and **Scalar 2** to improve the aspect ratio of the scene:
   a) Select the **Dual Scene** > **Displayers** > **Outline 1** node and set its **Transform** property to **Resize Domain**.
   b) Select the **Dual Scene** > **Displayers** > **Scalar 1** node and set its **Transform** property to **Resize Domain**.
   c) Select the **Dual Scene** > **Displayers** > **Scalar 2** node and set its **Transform** property to **Resize Domain**.
6. Using the mouse buttons, move the legends and adjust the scene so that it appears as shown.



7. Save the simulation.

Running the Simulation

Preparation of the simulation is now complete, and the simulation can be run.

To run the simulation:

1. Click (**Run**) in the *Solution* toolbar.

   The solution progress is displayed in the *Output* window. The *Residuals* display is created automatically in the *Graphics* window and shows the progress of the solvers.

   While the simulation is running, you can click the tabs at the top of the *Graphics* window to view scenes and plots.

During the run, you can stop the process by clicking (**Stop**) on the toolbar. If you do stop the simulation, you can click (**Run**) to resume it.

If left alone, the simulation continues until 200 iterations are complete. When the last iteration is complete, the following message is displayed in the *Output* window:

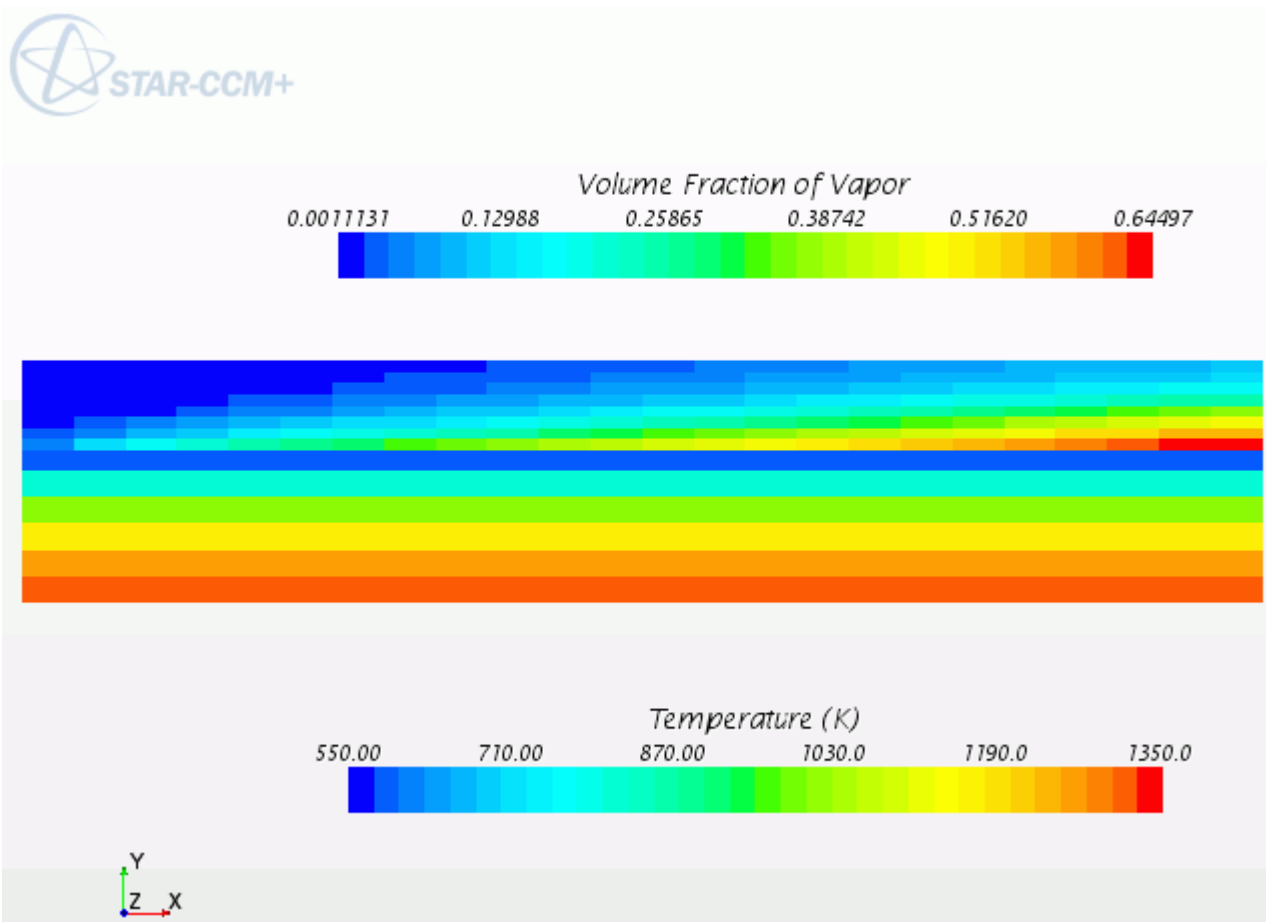
```
Stopping criterion Maximum Steps satisfied.
```

2. When the simulation has finished running, save it.

Visualizing the Results

Modify the scalar scene to better visualize the results of the simulation.

The scalar scene depicting the volume fraction of the vapor and the temperature of the cladding and fuel is shown below.

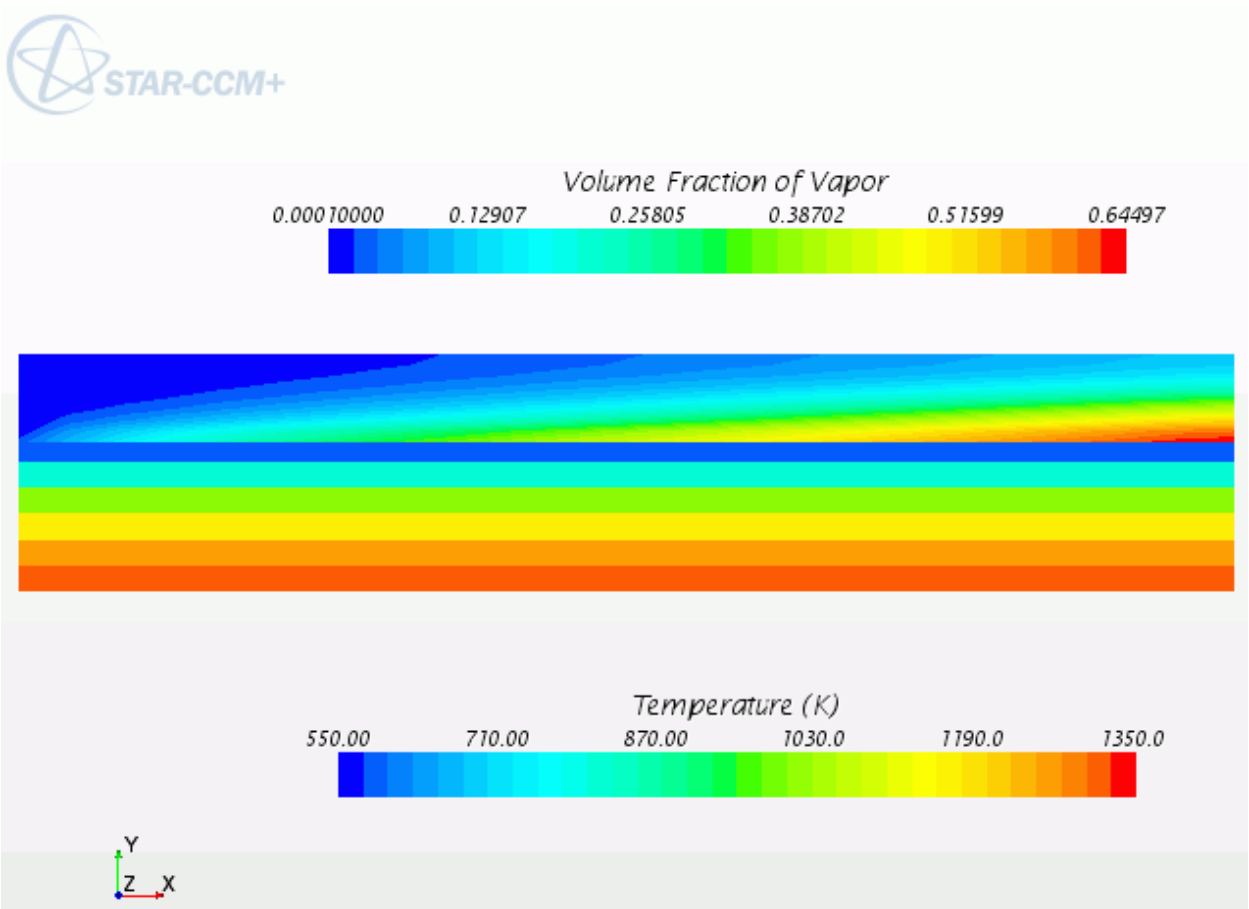


Make the vapor volume fraction representation smoother by changing the contour style of its scalar displayer.

To visualize the results:

1. Select **Scenes** > **Dual Scene** > **Displayers** > **Scalar 2**.
2. In the *Properties* window, set the **Contour Style** to **Smooth Filled**.

The *Dual Scene* updates as shown.



3. Save the simulation.

Validating the Results

Compare the results of the simulation with experimental data.

The computed temperature profile is validated by assuming that:

- Saturation conditions are obtained after a distance of 1 m.
- There are 6 K of superheat at the wall.
- An analytical solution for axisymmetric heat conduction can represent the heat transfer across the fuel and cladding.
- There is a contact resistance of 1.761E-4 (m^2K/W) at the cladding-fuel interface.
- The expected temperature at the center of the fuel is about 1000 C.

Import the expected radial temperature profile from a provided file:

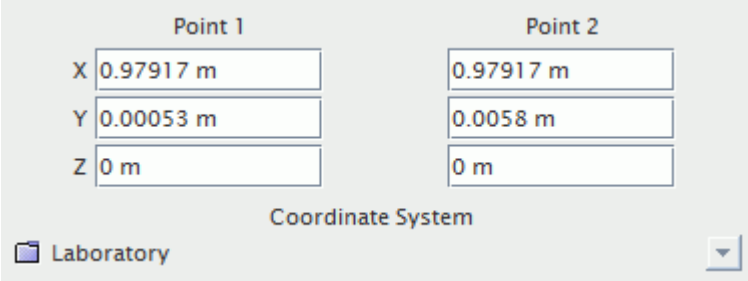
1. Right-click **Tools** > **Tables** and select **New Table** > **File Table**.
2. In the *Open* dialog, navigate to the `doc/startutorialsdata/multiphaseFlow/data` folder of your STAR-CCM+ installation and select file `WallBoilingCht.csv`.

3. Click **Open** to start the import.

   The **WallBoilingCht** child node is created under **Tables**.

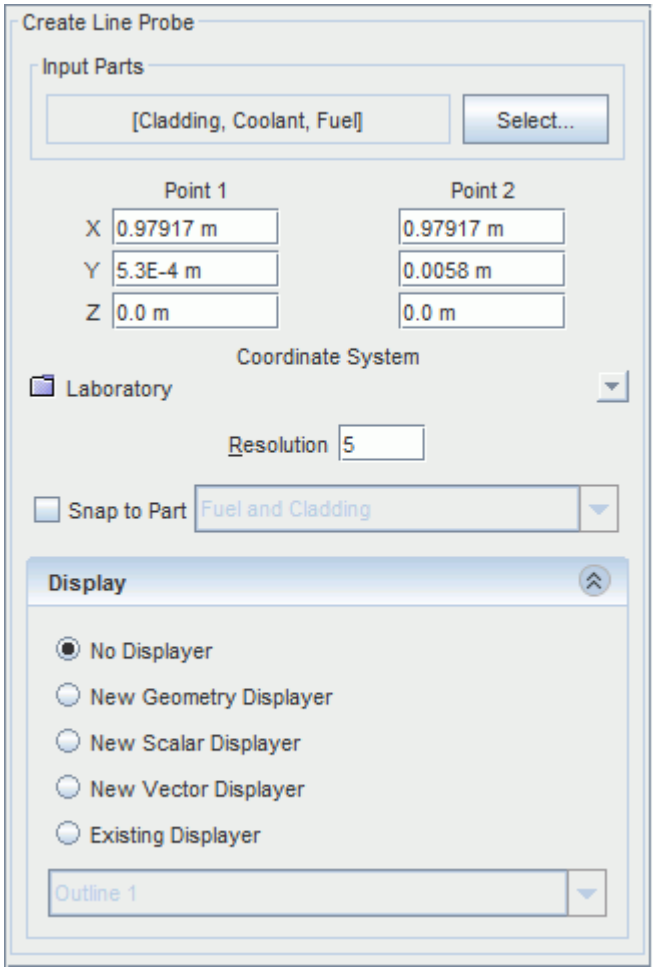
You create two derived parts that extract the radial temperature across the domain. Start with the fuel and cladding regions:

4. Right-click **Derived Parts** and choose **New Part** > **Probe** > **Line**.
5. Click the **[Cladding, Coolant, Fuel]** button in the *Input Parts* group-box.
6. Deactivate the **Coolant** node, leaving **Cladding** and **Fuel** activated.
7. Click **OK**.
8. Enter the coordinates of *Point 1* and *Point 2* as shown:

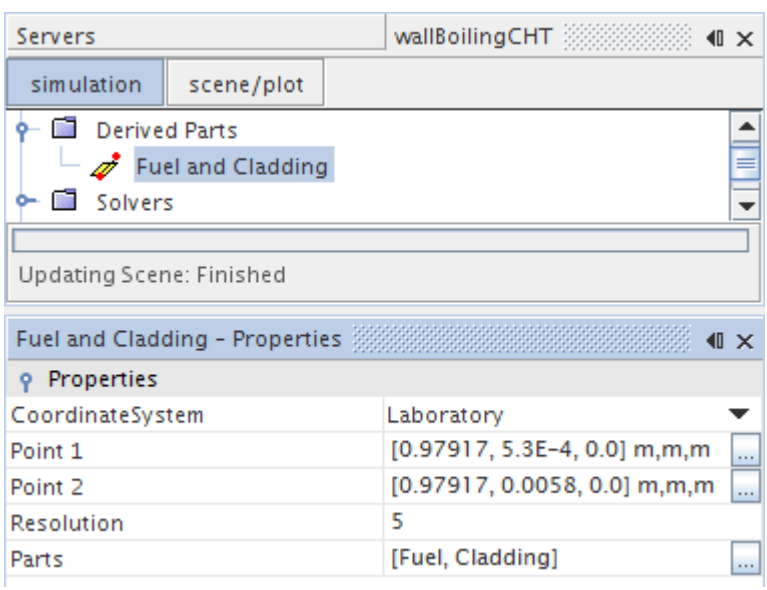


9. Leave the *Coordinate System* as **Laboratory**.
10. Enter a *Resolution* of 5.
11. Select **No Displayer** from the *Display* group box.

    The *Create Line Probe* window appears as shown.

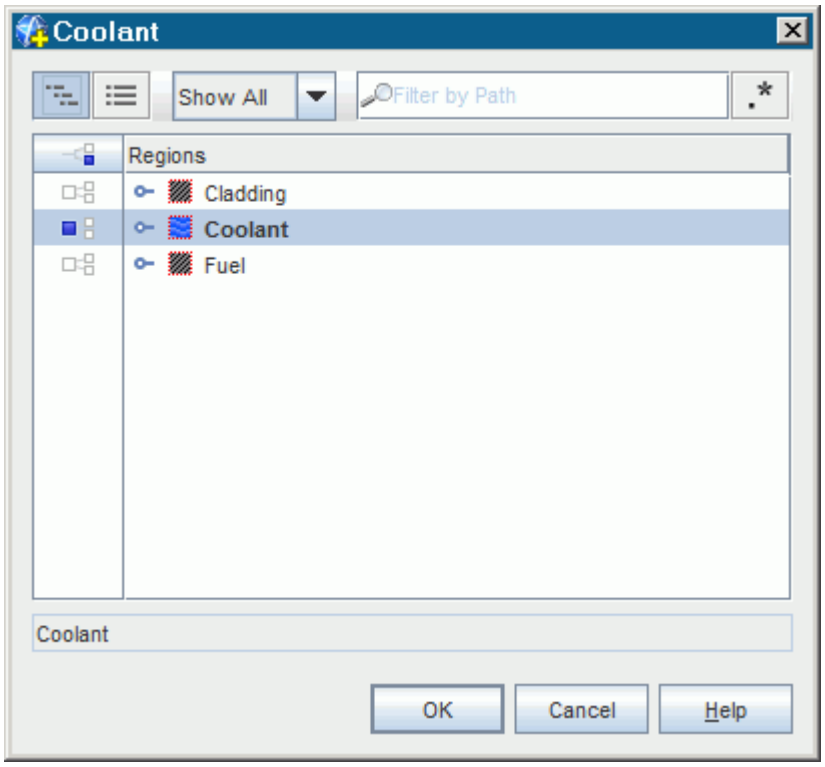


12. Click **Create**, and then **Close**.
13. Rename **Derived Parts** > **Line Probe** to **Fuel and Cladding**.



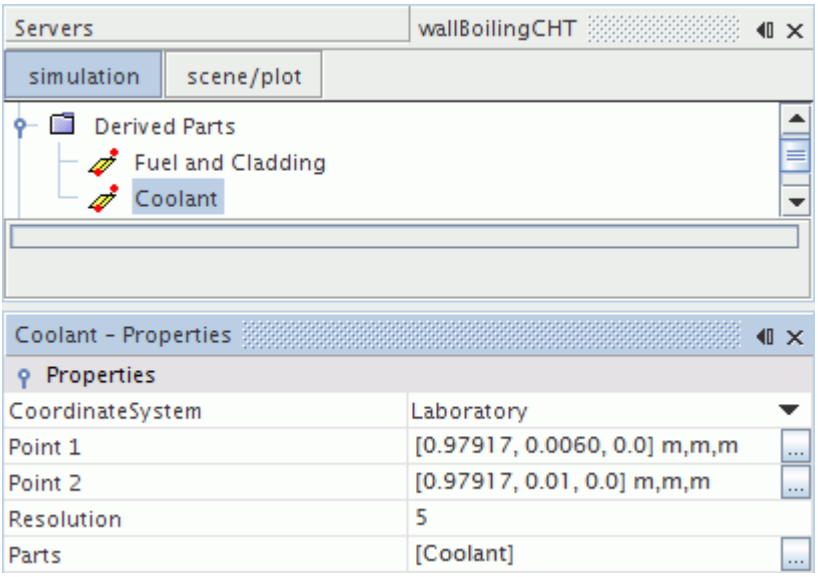
To create a derived part for the coolant region, copy the **Fuel and Cladding** line probe and edit its properties:

14. Right-click **Fuel and Cladding** and select **Copy**.
15. Right-click **Derived Parts** and select **Paste**.
16. Rename **Copy of Fuel and Cladding** to **Coolant**.
17. In the *Properties* window, set **Point 1** to `[0.97917, 0.0060, 0.0] m, m, m`.
18. Set **Point 2** to `[0.97917, 0.01, 0.0] m, m, m`.
19. Click ... (**Custom Editor**) for the **Parts** property.
20. In the *Coolant - Parts* dialog, right-click on a blank area and select **Deselect All**.
21. Activate **Coolant**.



22. Click **OK**.

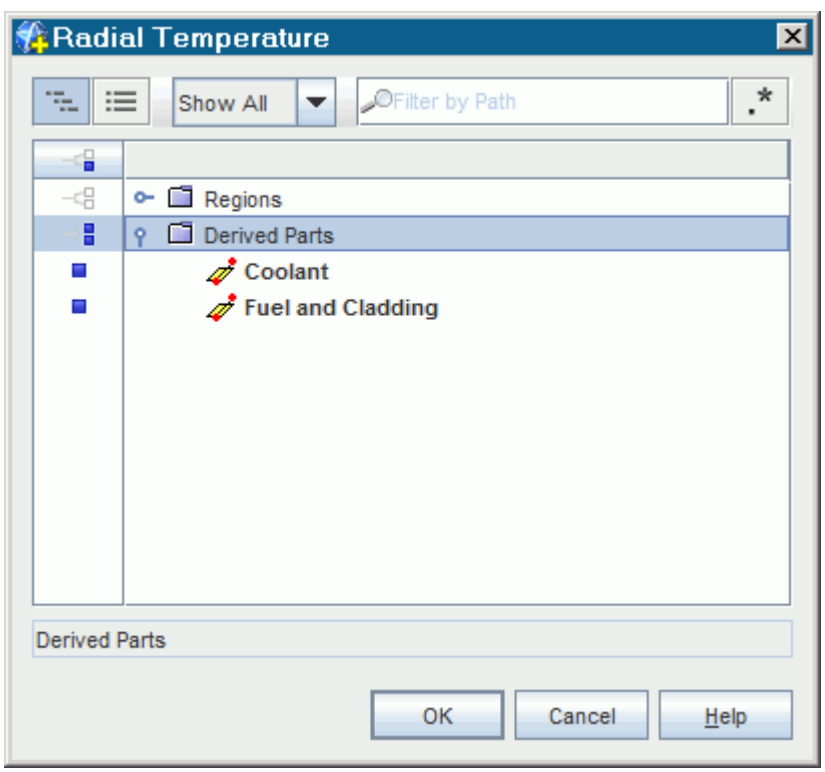
    The *Coolant - Properties* window appears as shown.



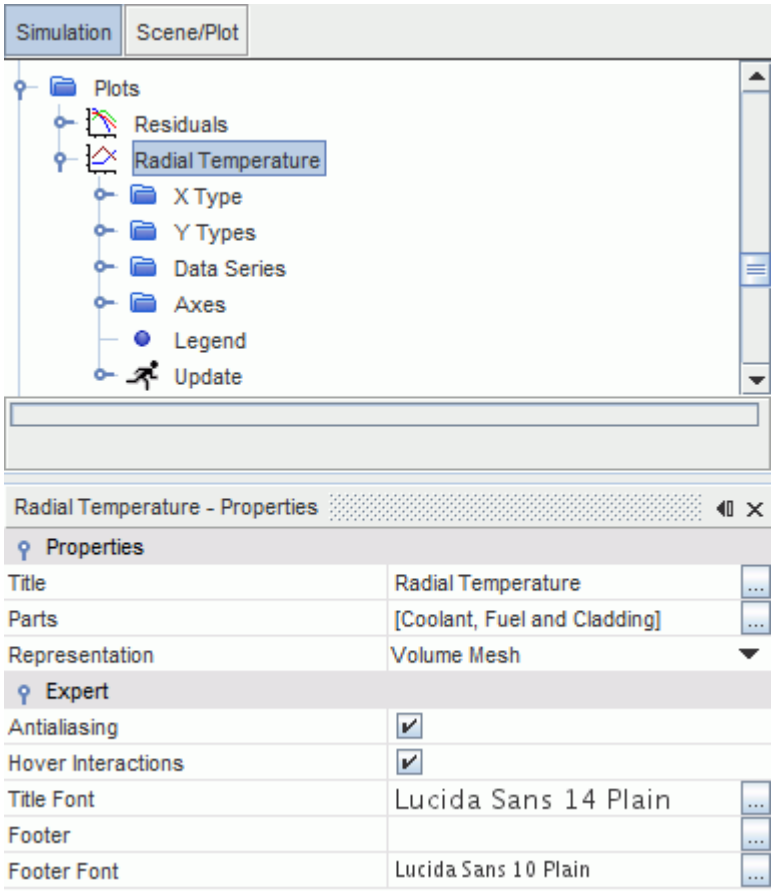
Create an X-Y plot that displays the radial temperature profile:

23. Right-click **Plots** and select **New Plot** > **XY Plot**.
24. Rename **XY Plot 1** to **Radial Temperature**.
25. In the *Properties* window, set its **Title** property to **Radial Temperature**.
26. Click ... (**Custom Editor**) for the **Parts** property.
27. In the *Radial Temperature - Parts* dialog, activate **Derived Parts**.

    Both **Coolant** and **Fuel and Cladding** parts are selected.



28. Click **OK**.



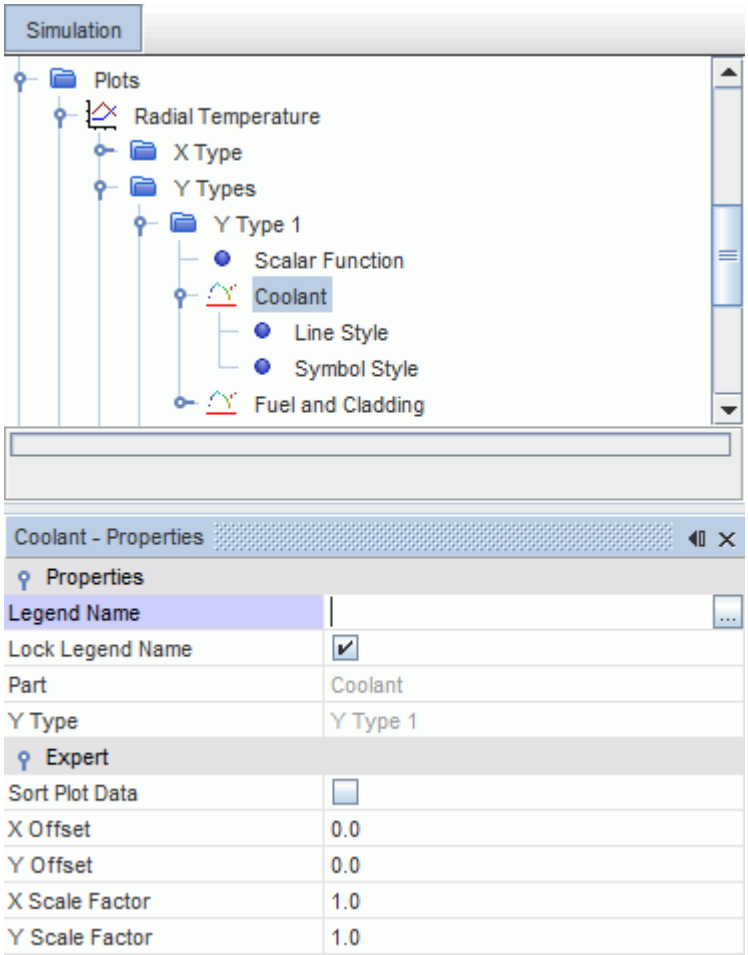
29. Select **Radial Temperature** > **X Type** and set its **Type** property to **Direction**.
30. Select **Radial Temperature** > **X Type** > **Vector Quantity** and set its **Value** property to `[0.0, 1.0, 0.0]`.
31. Select **Radial Temperature** > **Y Types** > **Y Type 1** > **Scalar Function** and set its **Field Function** property to **Temperature**.

Create a second Y Type that displays the temperature in the coolant:

32. Right-click **Radial Temperature** > **Y Types** and select **New**.
33. Select **Radial Temperature** > **Y Types** > **Y Type 2** > **Scalar Function** and set its **Field Function** property to **Temperature** > **Temperature of Liquid**.

The temperature profile in the coolant is not plotted. Remove it from the display:

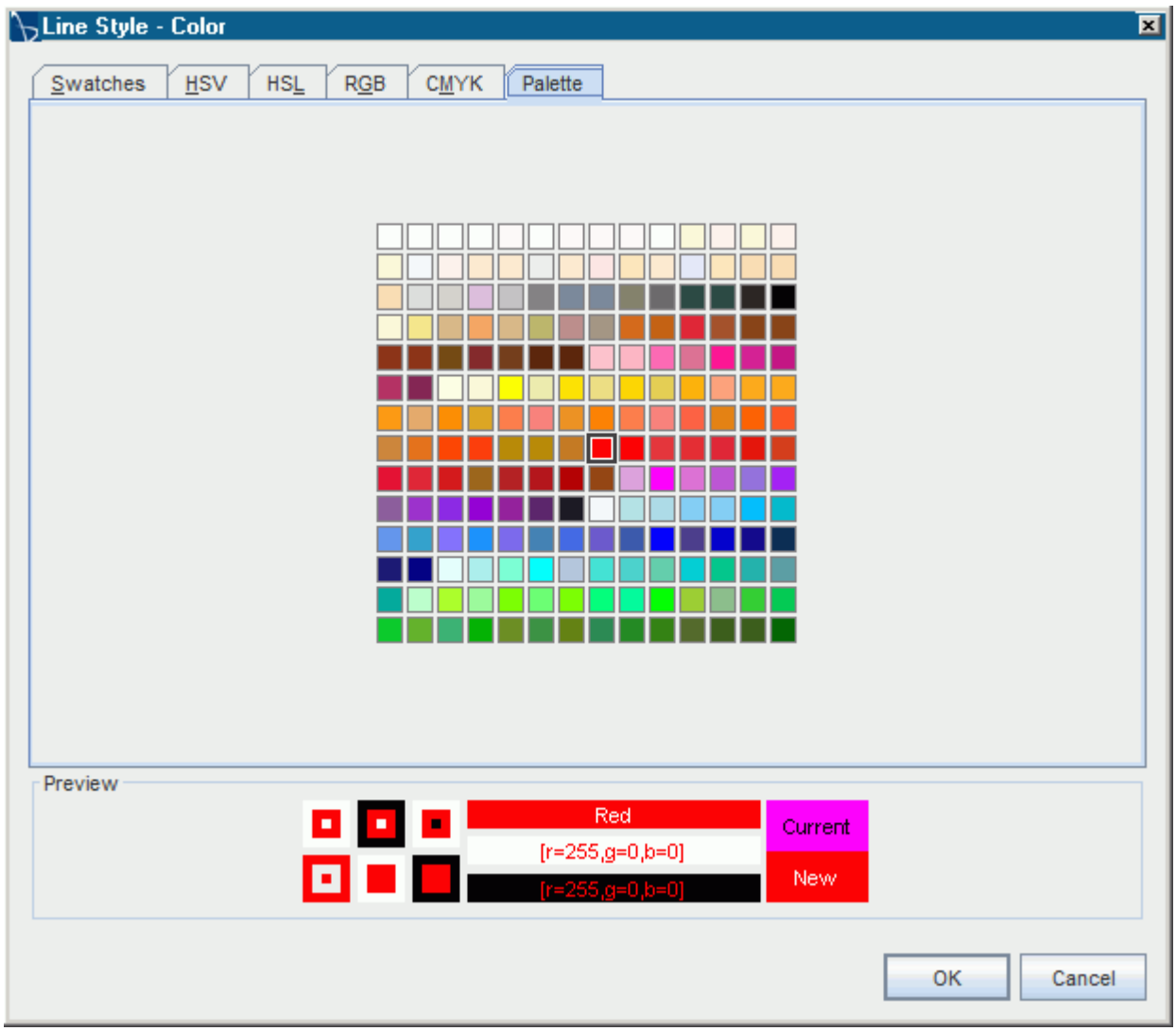
34. Select **Y Types** > **Y Type 1** > **Coolant** and, in the *Properties* window, deactivate **Lock Legend Name** and then erase the **Legend Name**.



35. Select **Y Types** > **Y Type 1** > **Coolant** > **Symbol Style** and set its **Shape** to **None**.
36. Select **Y Types** > **Y Type 2** > **Fuel and Cladding** and, in the *Properties* window, deactivate **Lock Legend Name** and then erase the *Legend Name*.
37. Select **Y Types** > **Y Type 2** > **Fuel and Cladding** > **Symbol Style** and set its **Shape** to **None**.

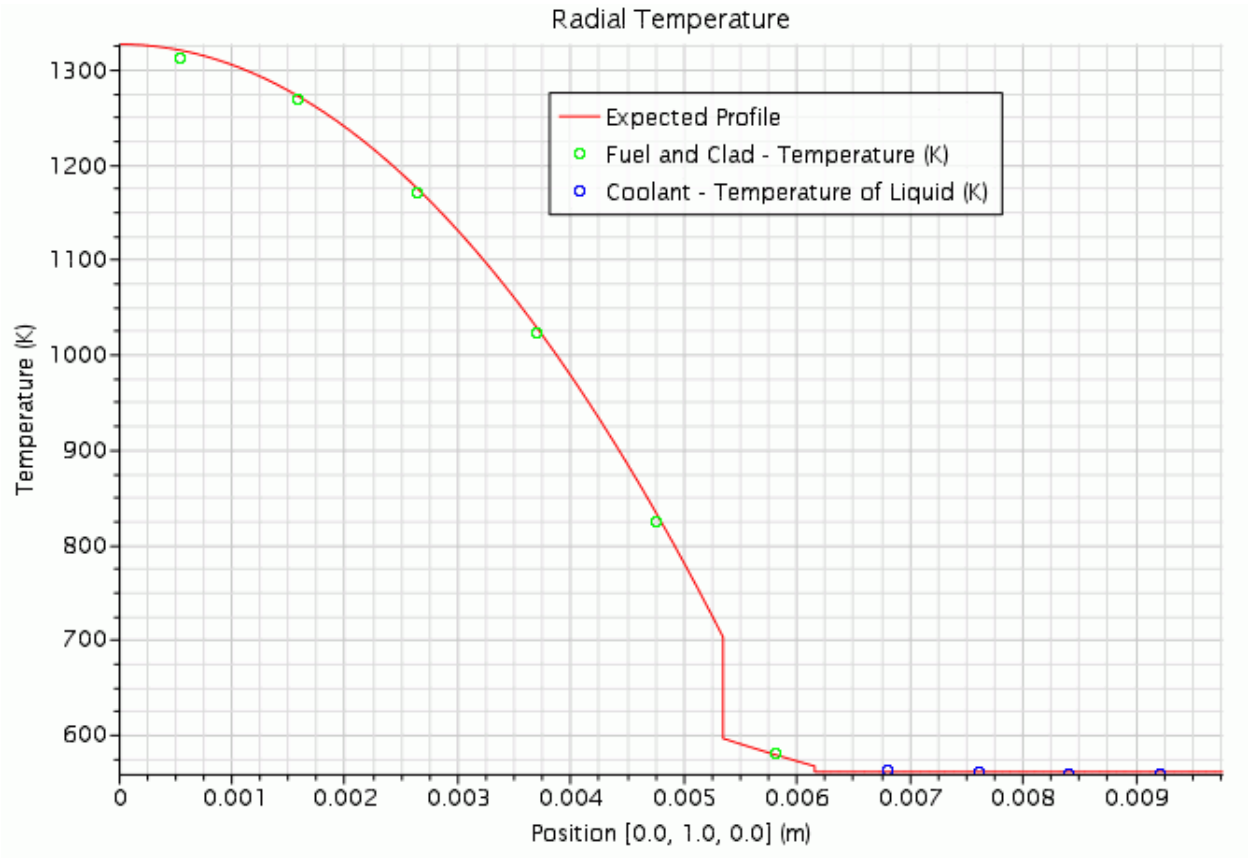
Import the expected profile as a table and add it to the plot:

38. Right-click **Radial Temperature** > **Data Series** and select **Add Data**.
39. In the *Add Data Providers to Plot* dialog, select **WallBoilingCht** and then click **OK**.
40. Select the **Plots** > **Radial Temperature** > **Data Series** > **WallBoilingCht** node and change the **Legend Name** to `Expected Profile`.
41. Set the **X Column** to **r**.
42. Set the **Y Column** to **T**.
43. Select **WallBoilingCht** > **Symbol Style** and set its **Shape** to **None**.
44. Select the **WallBoilingCht** > **Line Style** node.
45. In the *Properties* window, set the **Style** to **Solid**.
46. Click ... (**Custom Editor**) for the **Color** property.
47. In the *Line Style - Colour* dialog, click the *Palette* tab.
48. Select **Red**.



49. Click **OK**.
50. To add the X-axis title, select **Radial Temperature** > **Axes** > **Bottom Axis** > **Title** and, in the *Properties* window, set the **Title** to `Position [0.0, 1.0, 0.0] (m)`.
51. To add the Y-axis title, select **Radial Temperature** > **Axes** > **Left Axis** > **Title** and, in the *Properties* window, set the **Title** to `Temperature (K)`.

    The *Radial Temperature* plot appears as shown.



The computed temperature shows good agreement with the expected radial profile.

52. Save the simulation.

Summary

This tutorial demonstrates how to set up a conjugate heat transfer wall boiling analysis in STAR-CCM+.

This tutorial has introduced the following STAR-CCM+ features:

- Setting up the Eulerian Multiphase and Multiphase Turbulence models.
- Creating phases and selecting phase models for wall boiling.
- Defining phase interactions.
- Visualizing two field functions on the same scene.
- Plotting a radial temperature profile using two line probes.

Eulerian Multiple Flow Regimes: Pressurized Water Reactor

This tutorial demonstrates how to use the Multiple Flow Regimes model in STAR-CCM+ to simulate the counter-current air-water flow in a pressurized water reactor.

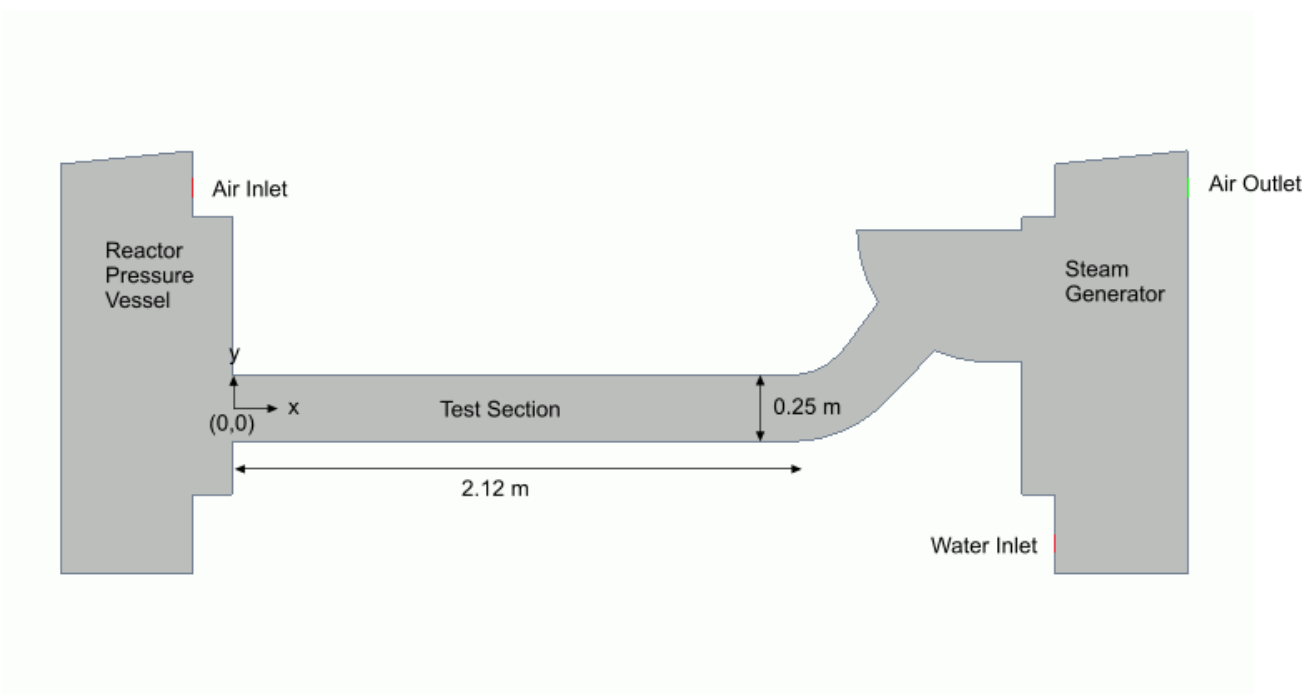
The Multiple Flow Regimes model combines the benefits of the VOF model and the Eulerian Multiphase model in a single model that can capture a number of different co-existing flow regimes:

- Stratified flow / free surfaces
- Dispersed sprays
- Dispersed bubbles

The Multiple Flow Regimes model captures sharp free surfaces in a similar manner to the VOF model. However, when droplets or bubbles are smaller than the mesh can resolve, the Multiple Flow Regimes model treats them as dispersed phases. This approach is analogous to the Large Eddy Simulation (LES) model where scales larger than the grid are resolved, and subgrid scales are modeled.

This tutorial uses the Multiple Flow Regimes model to simulate the counter-current air-water flow in a pressurized water reactor (PWR). In some scenarios, the flow rate of liquid coolant into the PWR can be highly reduced by a counter-flow of gas. This flow can be very unsteady and include all of the flow regimes that are listed above.

The geometry that is used is a 2D 1/3rd scale model of the hot leg PWR of a German Konvoi Pressurized Water Reactor. The geometry is shown below.



The reactor pressure vessel (RPV) on the left and steam generator (SG) separator on the right are connected by a test section. The test section is composed of a horizontal rectangular channel, a bend that connects it to

an upward inclined and expanded channel, and a quarter of a circle representing the steam generator inlet chamber. The horizontal part of the test section is 2.12 m long and 0.25 m high. The riser is 0.23 m long, has an inclination of 50 degrees to the horizontal plane, and an expansion angle of 7.5 degrees. The inner and outer bend radii of curvature are 0.25 m and 0.5 m, respectively.

In this simulation, a constant water flow rate is injected at the bottom of the steam generator simulator from where it flows through the test section to the RPV. Air is injected into the top of the RPV and flows through the test section to the SG separator in counter-current to the water flow.

Contents:

Prerequisites
Importing the Mesh and Naming the Simulation
Selecting the Physics Models
Creating Phases and Selecting Phase Models
Setting Initial Conditions and Reference Values
Defining the Multiple Flow Regime Phase Interaction
Setting Boundary Conditions
Setting Stopping Criteria and Solver Parameters
Setting up a Monitor
Creating a Scalar Scene to Visualize the Solution
Running the Simulation
Visualizing the Results
Animating the Results
Summary

Prerequisites

The instructions in the Eulerian Multiple Flow Regimes: Pressurized Water Reactor tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Importing the Mesh and Naming the Simulation

Launch STAR-CCM+, import the supplied geometry, and save the simulation.

To import the mesh and name the simulation:

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh**.

4. In the *Open* dialog, choose `doc/startutorialsdata/multiphaseFlow/data/pressurizedwaterreactor.ccm`, then click **Open**.

   STAR-CCM+ provides feedback on the import process, which takes a few seconds, in the *Output* window.
5. Save the simulation as `PWR_AirWaterFlow.sim`.

Selecting the Physics Models

Physics models define the primary variables of the simulation, including pressure, temperature, velocity, and what mathematical formulation is used to generate the solution.

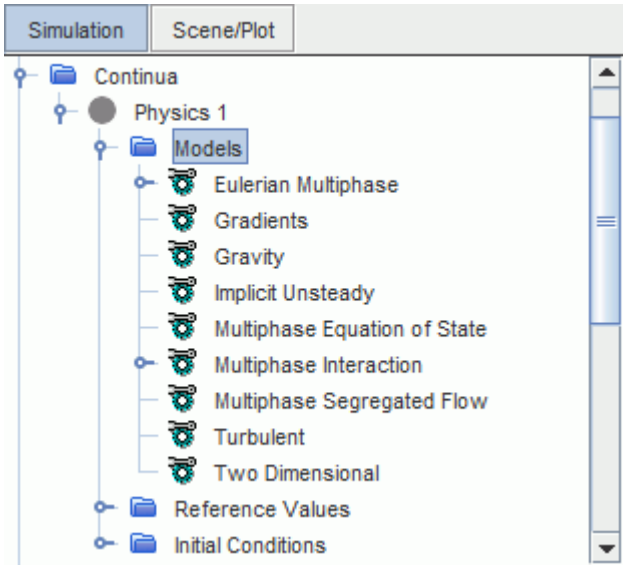
The core physics model that this tutorial requires is the Multiphase Segregated Flow model which provides a full Eulerian solution for the air-water flow. The Multiple Flow Regimes portion of the setup is defined separately in a phase interaction.

To select the physics models:

1. In the physics continuum, **Continua** > **Physics 1**, select the following models in order:

Group Box	Model
Space	**Two Dimensional** (pre-selected)
Time	**Implicit Unsteady**
Material	**Eulerian Multiphase** **Multiphase Interaction** (selected automatically)
Eulerian Multiphase Model	**Multiphase Segregated Flow** **Multiphase Equation of State** (selected automatically) **Gradients** (selected automatically)
Viscous Regime	**Turbulent**
Optional Models	**Gravity**

2. Click **Close**.
3. To review the models, open the **Physics 1** > **Models** node.



You now set the appropriate properties. In a Multiple Flow Regimes simulation, you are advised to use second order convection for the Volume Fraction as well as the Flow. This setting reduces numerical diffusion and helps to obtain a reasonably sharp interface. The minimum volume fraction setting increases the robustness of the simulation.

4. Select the **Multiphase Segregated Flow** node and set the following properties:

Property	Setting
Volume Fraction Convection	**2nd-order**
Minimum Volume Fraction	`1.0E-12`

5. Save the simulation.

Creating Phases and Selecting Phase Models

Create the phases and select the appropriate phase models.

The two phases flowing through the test section of the pressurized water reactor are air and water. You define these phases under the Eulerian Multiphase physics model node. This case uses the K-Omega Turbulence model as it gives the best results for confined flows. Additionally, the phase interaction uses the Interface Turbulence Damping model which is compatible only with the K-Omega Turbulence models.

To create the phases and select the phase models:

Create the liquid phase.

1. In the **Physics 1** continuum, right-click the **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and select **New**.
2. Rename the **Phase 1** node to **Water**.
3. For the **Water** phase, select the following models in order:

Group Box	Model
Enabled Models	**Flow** (pre-selected) **Turbulent** (pre-selected)
Material	**Liquid** **Reynolds-Averaged Navier-Stokes** (selected automatically)
Reynolds-Averaged Turbulence	**K-Omega Turbulence** **SST (Menter) K-Omega** (selected automatically) **Exact Wall Distance** (selected automatically) **All y+ Wall Treatment** (selected automatically)
Equation of State	**Constant Density**

4. Click **Close**.

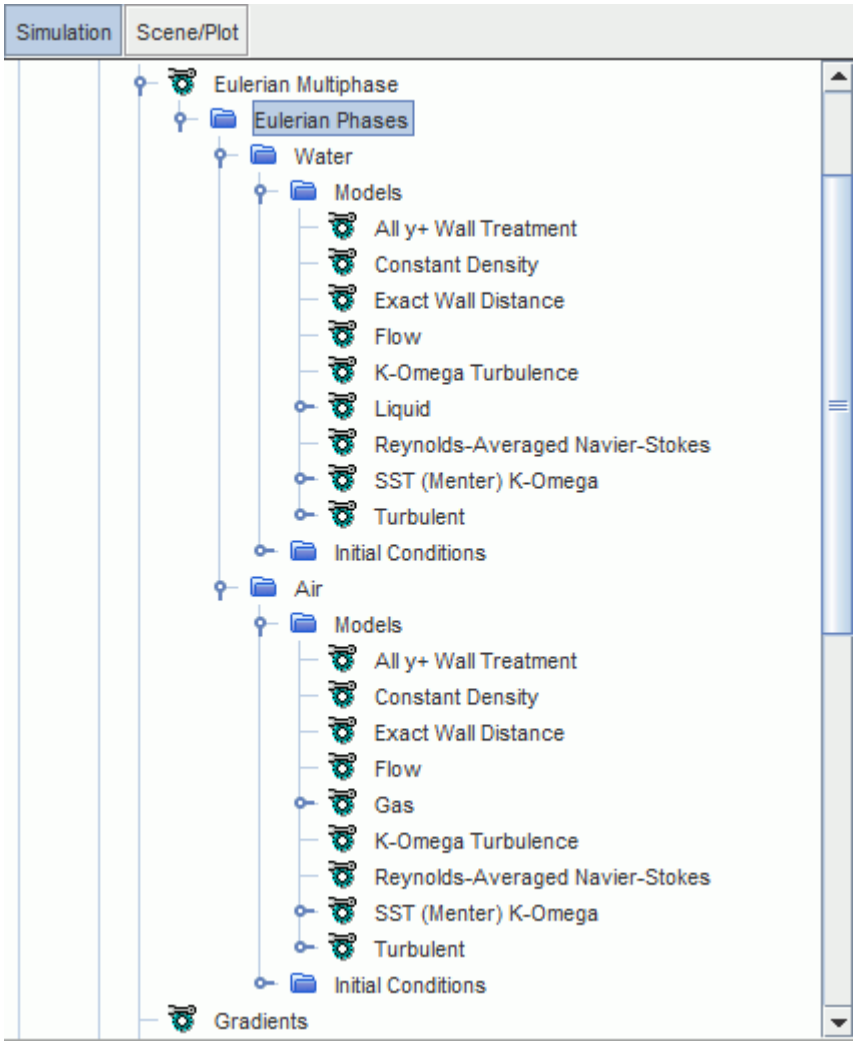
Create the gas phase.

5. Create a second phase and rename it to **Air**.
6. For the **Air** phase, select the following models in order:

Group Box	Model
Enabled Models	**Flow** (pre-selected) **Turbulent** (pre-selected)
Material	**Gas** **Reynolds-Averaged Navier-Stokes** (selected automatically)
Reynolds-Averaged Turbulence	**K-Omega Turbulence** **SST (Menter) K-Omega** (selected automatically) **Exact Wall Distance** (selected automatically) **All y+ Wall Treatment** (selected automatically)
Equation of State	**Constant Density**

7. Click **Close**.

   The **Eulerian Phases** node appears as shown below.

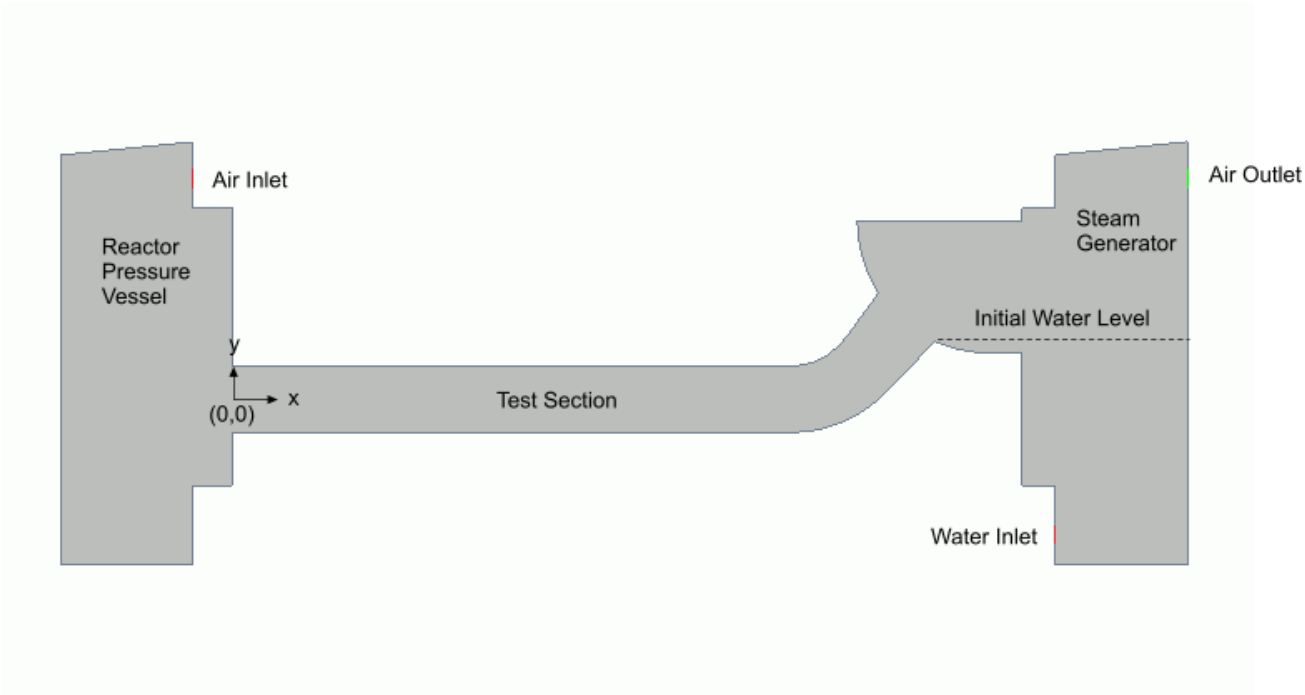


8. Save the simulation.

Setting Initial Conditions and Reference Values

Set the initial conditions and reference values for the physics continuum.

The initial volume fractions of water and air are specified as scalar functions. The steam generator is partially full of water, with the initial water level indicated in the diagram below:



To set the initial conditions and reference values:

1. Define the field function to use for specifying the initial volume fraction of water:
   a) Right-click the **Tools** > **Field Functions** node and select **New** > **Scalar**.
   b) Rename the **User Field Function 1** node to `Initial_VF_Water`.
   c) Select the **Initial_VF_Water** node and set the following properties:

Property	Setting
Function Name	`Initial_VF_Water`
Definition	`($${Centroid}[0]>2.67) ? (($${Centroid}[1]<0.230) ? 1.0 : 0.0 ) : 0.0`

2. Define the field function to use for specifying the initial volume fraction of air:
   a) Create another scalar field function and rename it to `Initial_VF_Air`.
   b) Select the **Initial_VF_Air** node and set the following properties:

Property	Setting
Function Name	`Initial_VF_Air`
Definition	`1.0-${Initial_VF_Water}`

3. Edit the **Continua** > **Physics 1** > **Models** > **Eulerian Multiphase** node and set the following properties:

Node	Property	Setting
Water > **Initial Conditions** > **Volume Fraction**	**Method**	**Field Function**
	Scalar Function	**Initial_VF_Water**

Node	Property	Setting
Air > **Initial Conditions** > **Volume Fraction**	**Method**	**Field Function**
	Scalar Function	**Initial_VF_Air**

Multiple Flow Regimes phase interactions do not have a clear reference phase because either of the phases can be dispersed in the other phase. In this scenario, the density variation can be significant (several orders of magnitude) and can result in numerical instabilities. To obtain a robust and stable numerical solution, set the reference density to zero. This setting ensures that the full body force due to gravity is used in the momentum equation.

Set the reference value for the **Physics 1** continuum:

4. Edit the **Continua** > **Physics 1** > **Reference Values** node and set the following property:

Node	Property	Setting
Reference Density	**Value**	`0.0 kg/m^3`

5. Save the simulation.

Defining the Multiple Flow Regime Phase Interaction

Use the Multiple Flow Regime Phase Interaction model to define the interaction between the air and water phases in the counter-current flow.

In cases that have a stratified two-phase flow that is combined with a dispersed two-phase flow, three concurrent regions of two-phase flow can be defined:

- The first regime: the region where the gas is dispersed in the liquid.
- The interface regime: the region where the gas and the liquid remain separated by a large scale interface.
- The second regime: the region where the liquid is dispersed in the gas.

Using the **Multiple Flow Regime Phase Interaction** model, you can define these flow regimes in the interaction between the water and air phases.

To define the phase interactions:

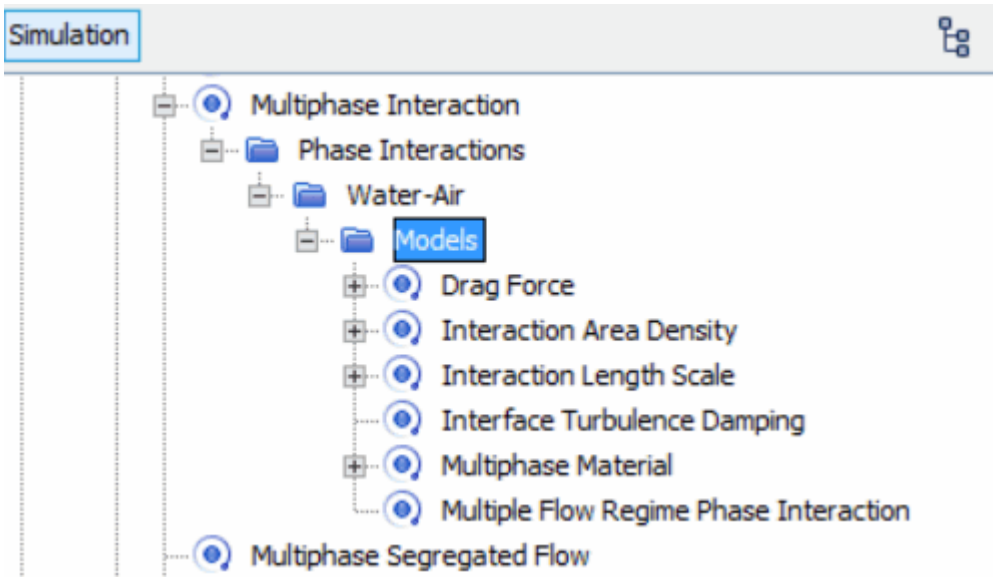
1. In the **Physics 1** continuum, right-click the **Models** > **Multiphase Interaction** > **Phase Interactions** node and select **New**.
2. Rename the **Phase Interaction 1** node to **Water-Air**.
3. For the **Water-Air** phase interaction, select the following models:

Group Box	Model
Phase Interaction Topology	**Multiple Flow Regime Phase Interaction** **Drag Force** (selected automatically) **Multiphase Material** (selected automatically)
Interaction Length Scale	**Interaction Length Scale** (selected automatically)
Interaction Area Density	**Interaction Area Density** (selected automatically)

Group Box	Model
Optional Models	**Interface Turbulence Damping** (selected automatically)

4. Click **Close**.

   The **Phase Interactions** > **Water-Air** > **Models** node appears as shown below.



Set the multiple flow regime phase interaction properties.

5. Select the **Water-Air** > **Models** > **Multiple Flow Regime Phase Interaction** node and set the following properties:

Property	Setting
Primary Phase	**Water**
Secondary Phase	**Air**

   The default values for the **First Regime Terminus** and **Second Regime Onset** are acceptable for this simulation.

   The first regime terminus is the value of the volume fraction of the secondary phase at which the first regime transits to the interface regime. In the first regime, the primary phase is treated as the continuous phase. The second regime onset is the value of the volume fraction of the secondary phase at which the interface regime transits to the second regime. In the second regime, the secondary phase is treated as the continuous phase.

Set the remaining phase interaction properties. In the vicinity of large scale interface, a boundary layer develops and the interface must be treated as a moving wall. The default properties for the Drag Force and the Interaction Area Density are appropriate for air-water flows. You set the Interaction Length Scale and Interface Turbulence Damping properties.

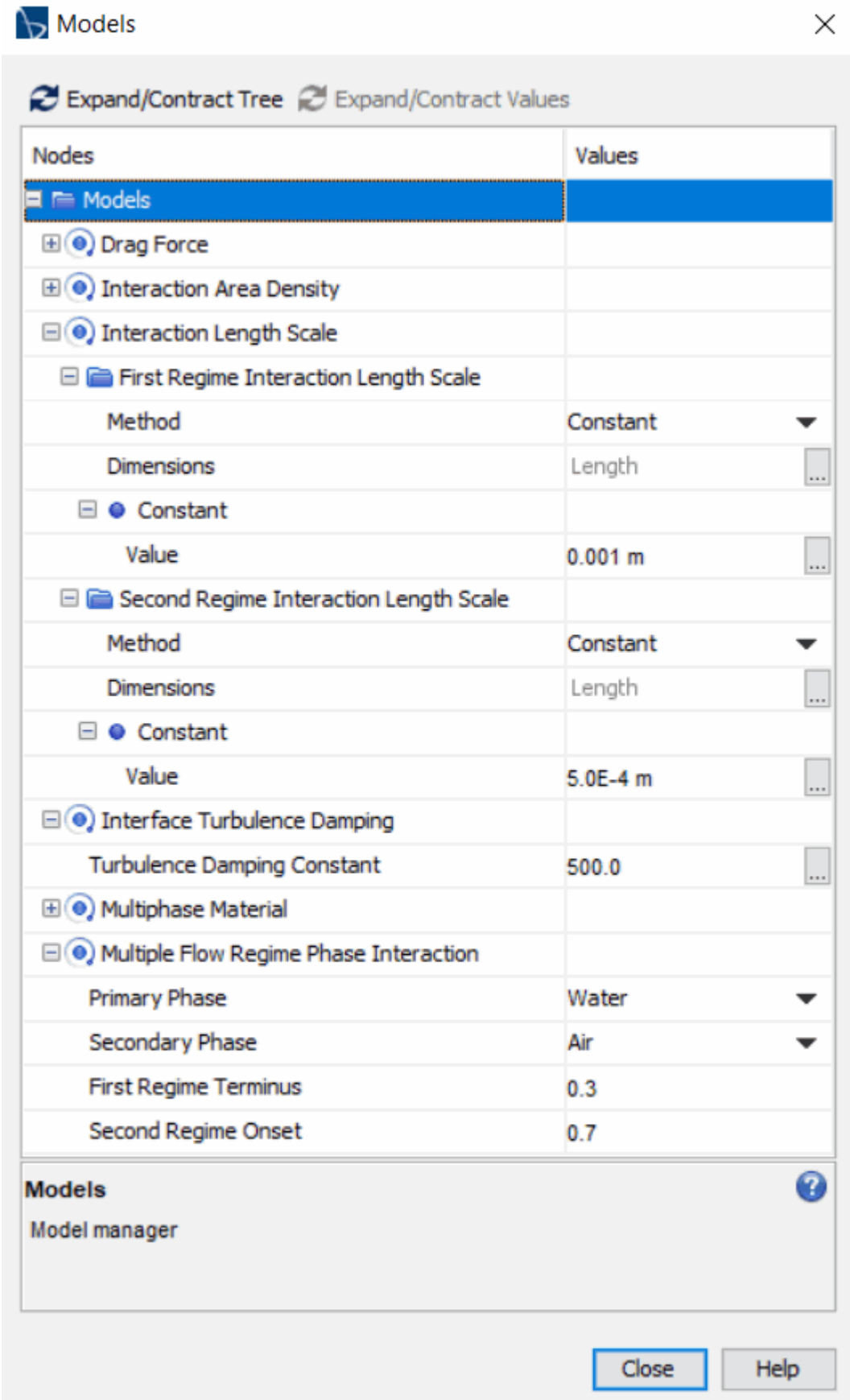
The First Regime Interaction Length Scale specifies the typical size of the air bubbles that are dispersed in the water. The Second Regime Interaction Length Scale specifies the typical size of the water droplets that are dispersed in the air.

For more information, see Modeling Interface Turbulence Damping.

6. Edit the **Water-Air** > **Models** node and set the following properties:

Node	Property	Setting
Interaction Length Scale > **First Regime Interaction Length Scale** > **Constant**	**Value**	`0.001 m`
⌊ **Second Regime Interaction Length Scale** > **Constant**	**Value**	`5.0E-4 m`
Interface Turbulence Damping	**Turbulence Damping Constant**	`500.0`

The completed *Edit* dialog is shown below.


Models
Expand/Contract Tree
Expand/Contract Values
Nodes
Values
Models
Drag Force
Interaction Area Density
Interaction Length Scale
First Regime Interaction Length Scale
Method
Constant
Dimensions
Length
Constant
Value
0.001 m
Second Regime Interaction Length Scale
Method
Constant
Dimensions
Length
Constant
Value
5.0E-4 m
Interface Turbulence Damping
Turbulence Damping Constant
500.0
Multiphase Material
Multiple Flow Regime Phase Interaction
Primary Phase
Water
Secondary Phase
Air
First Regime Terminus
0.3
Second Regime Onset
0.7
Models
Model manager
Close
Help

7. Save the simulation.

Setting Boundary Conditions

Specify the boundary conditions for the air inlet and water inlet boundaries.

To set the boundary conditions:

1. Edit the **Regions** > **PWR_2D** > **Boundaries** > **Air_Inlet** > **Phase Conditions** > **Air** > **Physics Values** node and set the following properties:

Node	Property	Setting
Velocity Magnitude	**Value**	`15.0 m/s`
Volume Fraction	**Value**	`1.0`

2. Edit the **Regions** > **PWR_2D** > **Boundaries** > **Water_Inlet** > **Phase Conditions** > **Water** > **Physics Values** node and set the following properties:

Node	Property	Setting
Velocity Magnitude	**Value**	`3.0 m/s`
Volume Fraction	**Value**	`1.0`

3. Save the simulation.

Setting Stopping Criteria and Solver Parameters

Set the appropriate solver parameters and stopping criteria for the simulation.

Modify the under-relaxation factors for the solvers to aid solution convergence.

As you are solving an unsteady problem, it is necessary to specify the time-step size and the elapsed simulation time. This calculation runs for a maximum physical time of 8.0 seconds with a time-step size of 0.01 s.

To set the solver parameters and stopping criteria:

1. Edit the **Solvers** node and set the following properties:

Node	Property	Setting
Implicit Unsteady	**Time-Step**	`0.01 s`
Multiphase Segregated Flow > **Phase Coupled Velocity**	**Implicit Under-Relaxation Factor**	`0.2`
	Explicit Under-Relaxation Factor	`0.5`
Multiphase Segregated Flow > **Pressure**	**Under-Relaxation Factor**	`0.1`

Node	Property	Setting
Volume Fraction	**Implicit Under-Relaxation Factor**	`0.2`
	Explicit Under-Relaxation Factor	`0.5`
K-Omega Turbulence	**Under-Relaxation Factor**	`0.2`
K-Omega Turbulent Viscosity	**Under-Relaxation Factor**	`0.2`
	Maximum Ratio	`1.0E8`

2. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Inner Iterations	**Maximum Inner Iterations**	`25`
Maximum Physical Time	**Maximum Physical Time**	`8.0 s`
Maximum Steps	**Enabled**	Deactivated

3. Save the simulation.

Setting up a Monitor

Create a monitor to visualize the water-air interface at a particular location within the test section.

By default STAR-CCM+ produces a residuals plot that you can use to monitor the convergence of the solution. Additionally it is useful to create other monitors to help in assessing the progress of the simulation. In this tutorial, you monitor the height of the water-air interface where it enters the horizontal region of the test section.

To set up a monitor:

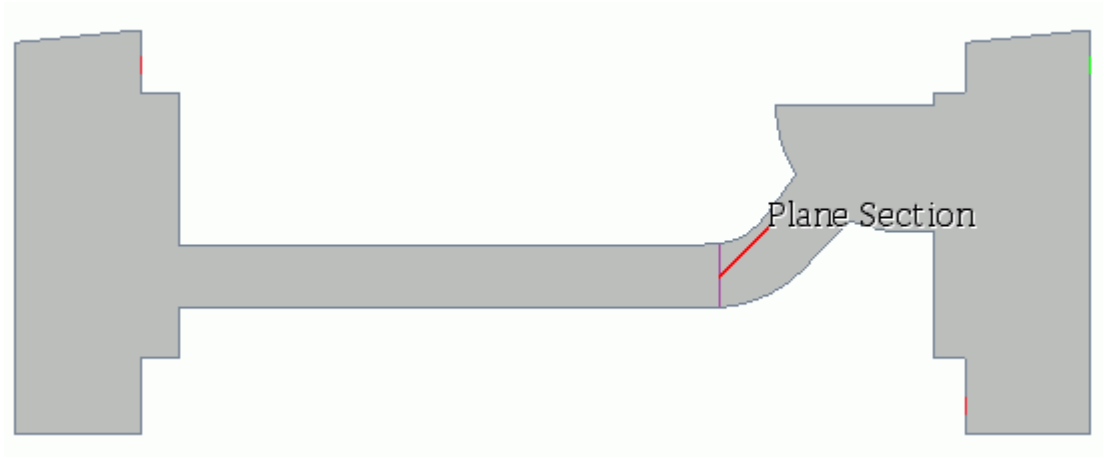
Create a plane section through the horizontal region of the test section:

1. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane**.

   The *Create Plane Section* dialog appears in the *Edit* window.
2. In the *Input Parts* box, make sure that **PWR_2D** is selected.
3. Set the *Origin* coordinates to `[2.15, 0.0, 0.0]` m.
4. In the *Display* box, select **No Displayer**.
5. Click **Create** and then **Close**.

   STAR-CCM+ creates a new derived part, **Plane Section**.

   The location of the new plane section is indicated in the screenshot below.



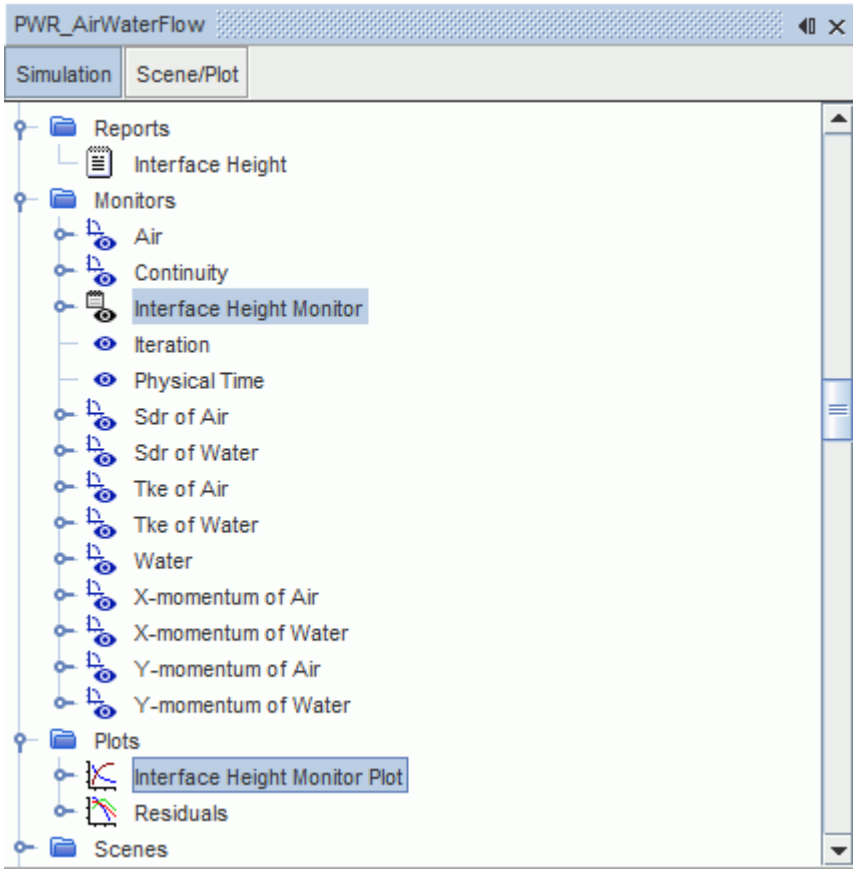
Before you can create the monitor, you create the report upon which the monitor is based. In this simulation, the height of the water-air interface is defined as the line integral of the volume fraction of water on the derived plane section. To monitor this value, you create a Line Integral report.

6. Right-click the **Reports** node and select **New Report** > **Line Integral**.

A new report is created named **Line Integral 1**, which is used to monitor the water volume fraction. Give the report a more suitable name.

7. Rename the **Line Integral 1** node to `Interface Height`.
8. Select the **Reports** > **Interface Height** node and set *Field Function* to **Volume Fraction** > **Volume Fraction of Water**.
9. Set *Parts* to **Derived Parts** > **Plane Section**.
10. Right-click the **Interface Height** node and select **Create Monitor and Plot from Report**.

    The **Interface Height Monitor** and **Interface Height Monitor Plot** are added to the simulation tree.



11. Save the simulation.

Creating a Scalar Scene to Visualize the Solution

Create a scalar scene to visualize the solution and define a trigger that saves the scene at regular intervals.

This scalar scene lets you visualize the volume fraction of water and provides qualitative information about the air-water interface. The scene is set up to write a .png file every five time steps. You can use the resulting files to create an animation that shows the volume fraction of water changing over the simulation run time.

To create the scalar scene:

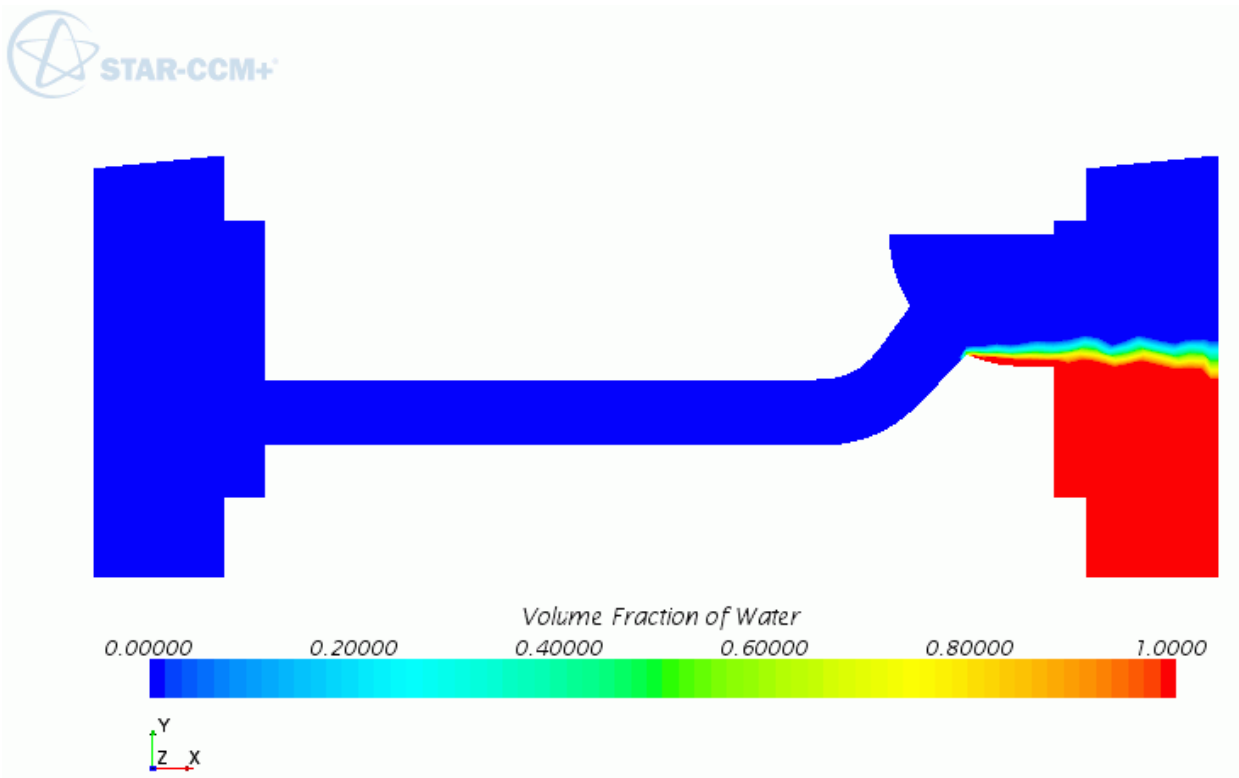
1. Right-click the **Scenes** node, and select **New Scene** > **Scalar**.
2. Click **Scene/Plot** above the object tree.
3. Select the **Displayers** > **Scalar 1** > **Scalar Field** node and do the following:
   a) Set the *Function* to **Volume Fraction** > **Volume Fraction of Water**.
   b) Set the *Auto Range* to **Off**.
   c) Set the *Min* to `0.0`.
   d) Set the *Max* to `1.0`.
4. Select the **Scalar 1** node and set the *Contour Style* to **Smooth Filled**.
5. Select the **Color Bar** node and set the *Levels* to **64**.
6. Select the **Attributes** > **Update** node and do the following:

a) Set *Save To File* to **Activated**.

b) Set *Trigger* to **Time Step**.

7. Select the **Update** > **Time Step Frequency** node and set *Frequency* to 5.
8. Click **Simulation** to return to the simulation object tree.
9. Click **Solution** > **Initialize Solution**.

   The scalar scene is updated to show the initial volume fraction of water:



10. Save the simulation.

Running the Simulation

Preparation of the simulation is now complete, and ready to run.

To run the simulation:

1. Click (**Run**) in the *Solution* toolbar.

   The solution progress is displayed in the *Output* window. The *Residuals* display opens automatically in the *Graphics* window and shows the progress of the solvers.

   While the simulation is running, you can click the tabs at the top of the *Graphics* window to view scenes and plots.

During the run, you can stop the process by clicking (**Stop**) on the toolbar. If you do stop the simulation, you can click (**Run**) to resume it.

If left alone, the simulation continues until 8 seconds of physical time is complete. When the last iteration is complete, the following message is displayed in the *Output* window:

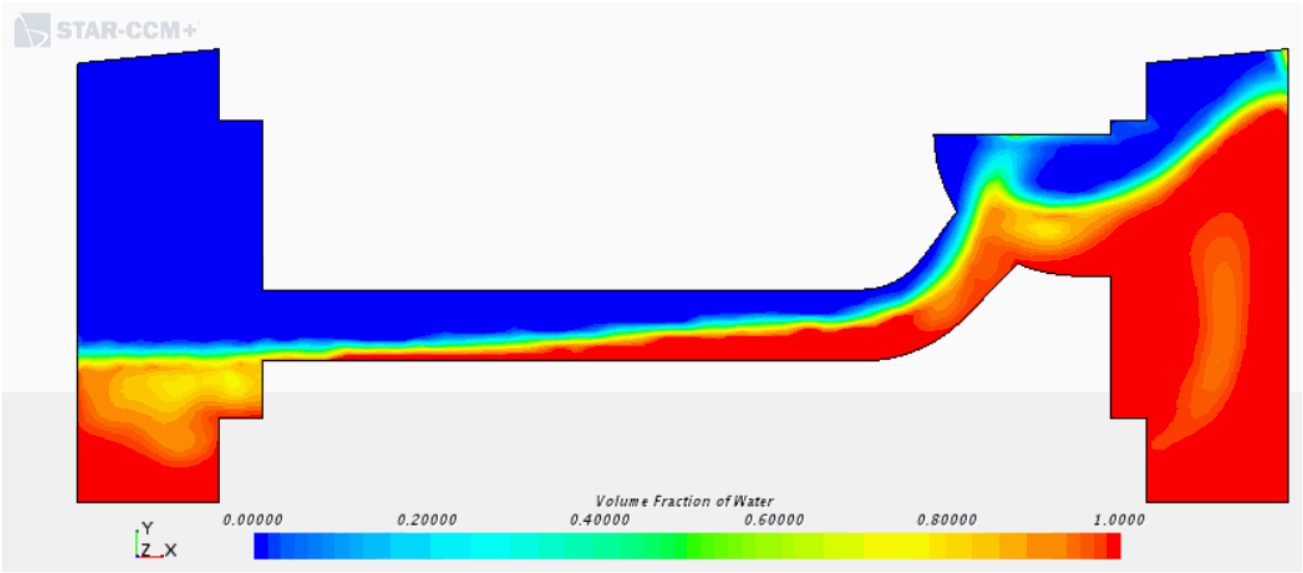
```
Stopping criterion Maximum Physical Time satisfied.
```

2. When the simulation has finished running, save it.

Visualizing the Results

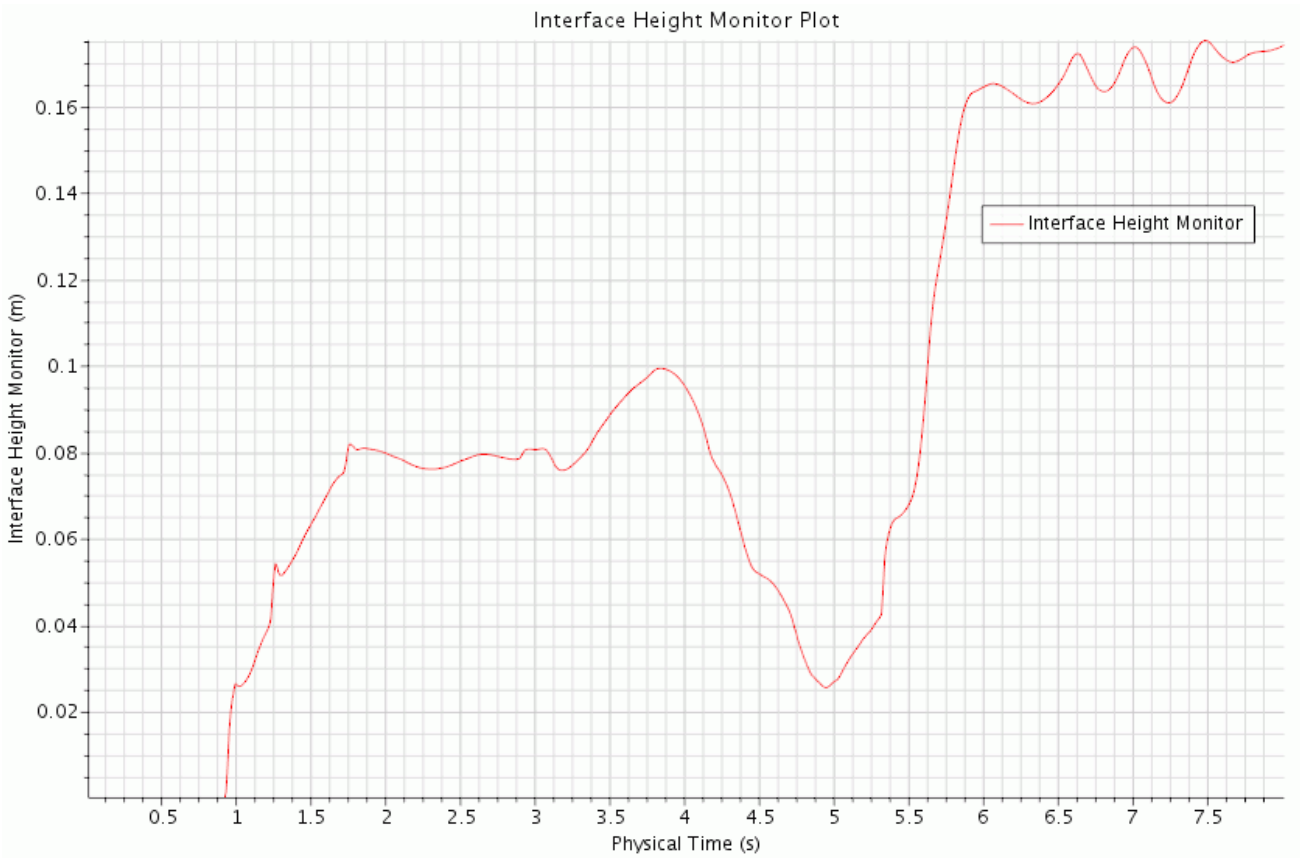
View the results of the simulation. You can visualize the multiple regimes and monitor the height of the Water-Air interface at a particular location to observe its periodic behavior.

The scalar scene shows the volume fraction of water at the end of the simulation.



This scene shows the coolant water flowing from right to left, and air flowing in the opposite direction. Regions of stratified flow, dispersed droplets, and dispersed bubbles are all present. Unresolved droplets and bubbles are clearly visible, and a roll wave can be seen where the test section joins the steam generator inlet chamber.

The *Interface Height Monitor Plot* shown below records the height of the Water-Air interface where the water flow enters the horizontal region of the test section. The plot illustrates the periodic behavior of the counter-current flow.

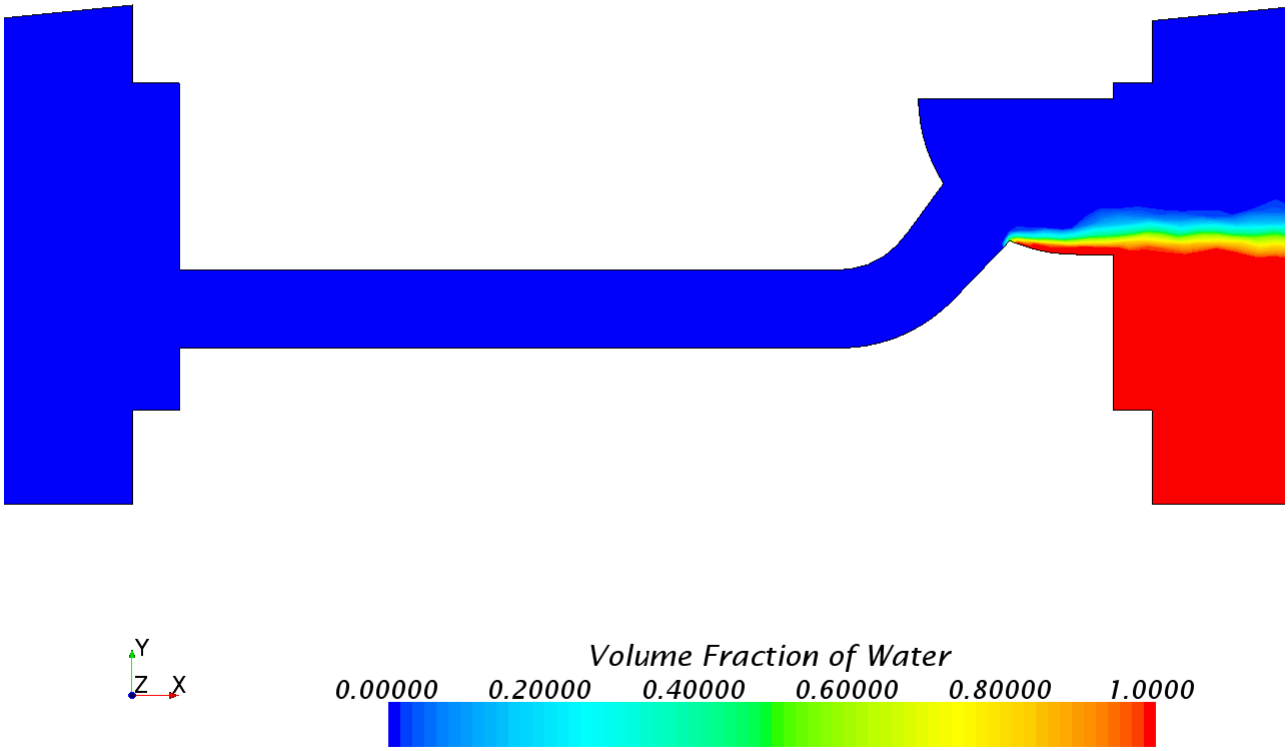


Animating the Results

Use the exported images of volume fraction to create an animation of flow within the channel.

An animated view lets you visualize the periodic blockage of the channel due to the build-up of water. This periodic behavior is a core feature of counter-current flows.

In the Creating a Scalar Scene to Visualize the Solution section, you instructed STAR-CCM+ to save the volume fraction of water profile every 5 time-steps. The resulting series of `.png` files can be used to produce an animation using an external package, such as the *animate* tool that is included with most Linux and Unix systems. An example of such an animation is shown below.



Summary

This tutorial demonstrated how to use the Multiple Flow Regimes model in STAR-CCM+ to simulate a counter-current two-phase flow.

The main feature of such flows is the presence of high shear near the large scale interface and the consequent generation of waves.

This tutorial has introduced the following STAR-CCM+ features:

- Setting up the Eulerian Multiphase and Turbulence models.
- Creating phases and selecting phase models.
- Defining Multiple Flow Regimes phase interactions.
- Setting up turbulence damping.
- Setting solver under-relaxation factors.
- Visualizing the results.

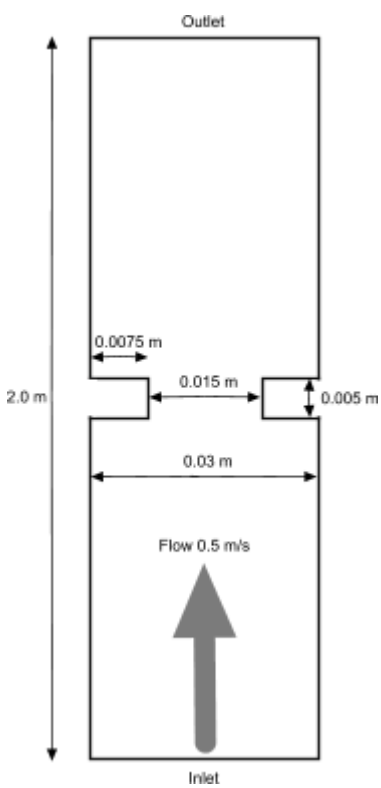
Eulerian: Pipe Flow with Multi-Speed AMUSIG

The Adaptive Multiple Size-Group (AMUSIG) model predicts the size distribution of particles (droplets or bubbles) in a dispersed multiphase flow regime. This tutorial demonstrates how to use the AMUSIG model for predicting the size distribution of hexane droplets in water. The water is flowing up a vertical pipe in which a concentric orifice increases turbulence and so accelerates droplet breakup.

The collection of droplets is modeled using five groups of representative computational particles. Each group consists of identical particles, and has its own size and its own velocity. A population balance is solved for each size-group in each cell.

The AMUSIG model is locally adaptive so that each size-group carries approximately the same share of the hexane phase. This method is efficient and benefits fully from the computational expense of multi-speed modeling. Multi-speed modeling is necessary when analyzing strong interactions between the size distribution and the flow.

The geometry of the pipe is shown below (not to scale). The flow enters the pipe at a velocity of 0.5 m/s and is composed of 99% water (by volume fraction) and 1% hexane.



Contents:

Prerequisites
Importing the Mesh and Naming the Simulation
Selecting the Physics Models

Creating Phases and Selecting Phase Models
Defining the Phase Interactions
Setting Initial Conditions and Reference Values
Setting Boundary Conditions
Setting Solver Parameters and Stopping Criteria
Setting Up the Plots
Visualizing the Solution
Running the Simulation
Visualizing the Results
Summary

Prerequisites

The instructions in the Eulerian: Pipe Flow with Multi-Speed AMUSIG tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

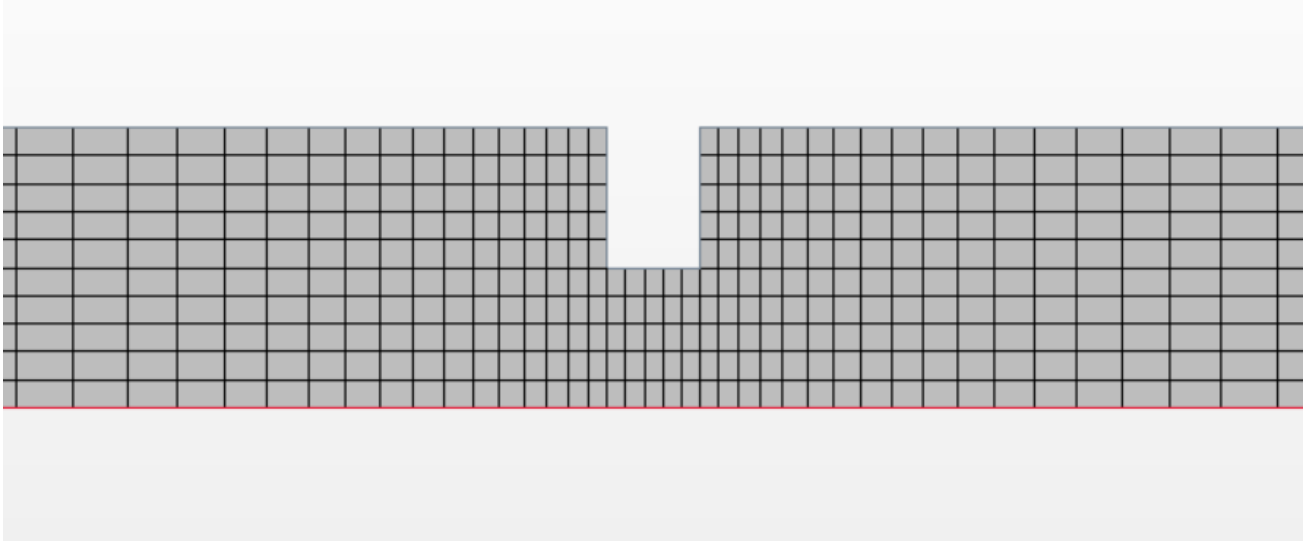
Before attempting this tutorial, make sure that you are familiar with the following:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Importing the Mesh and Naming the Simulation

A 2D axisymmetric mesh has been prepared for this analysis. Launch STAR-CCM+, import the mesh, and save the simulation.

This mesh represents an axisymmetric section of the pipe geometry. The mesh has a uniform spacing in the radial direction and a hyperbolic variation in the lengthwise direction. The minimum spacing is at the center of the pipe where the orifice is located.



To import the mesh and name the simulation:

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh**.
4. In the *Open* dialog, choose `doc/startutorialsdata/multiphaseFlow/data/Pipe_with_Restriction.ccm`, then click **Open**.

   STAR-CCM+ provides feedback on the import process in the *Output* window.
5. Save the simulation as `Pipe_Flow.sim`.

Selecting the Physics Models

Set up a Segregated Multiphase simulation. The geometry is axisymmetric and the flow is turbulent due to the restriction in the pipe. Gravity is included to model the buoyancy of the hexane droplets as they rise up the vertical pipe.

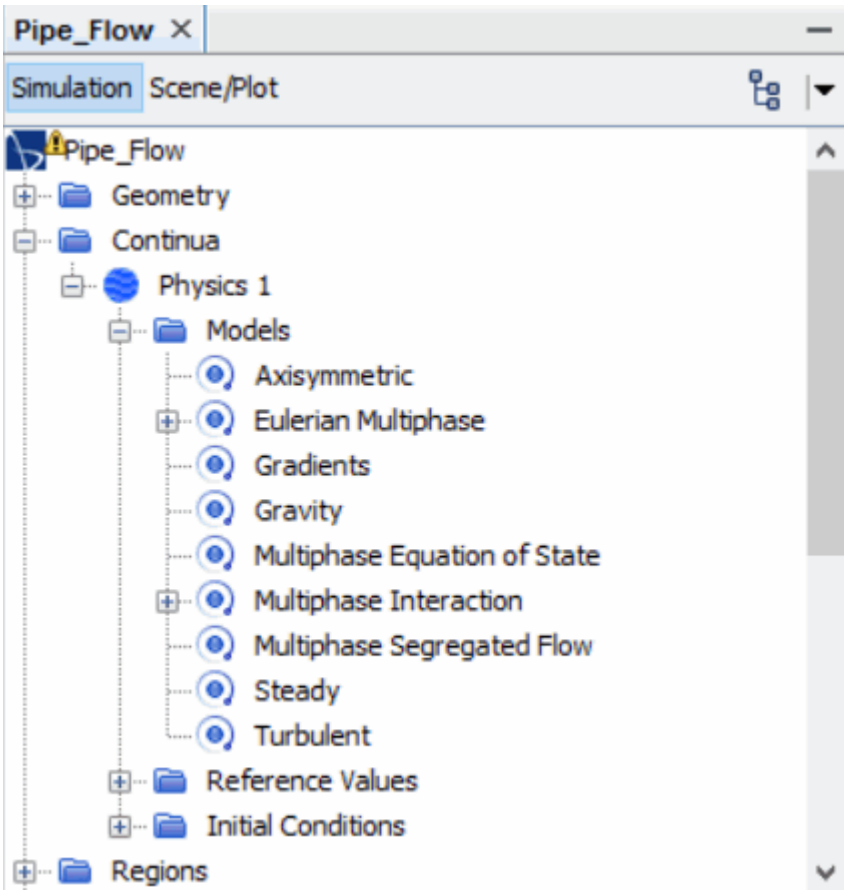
To select the physics models:

1. Right-click **Continua** > **Physics 1** and choose **Select Models**.
2. In the *Enabled Models* group box, deactivate **Two Dimensional**.
3. Select the following models in order:

Group Box	Model
Space	**Axisymmetric**
Time	**Steady**
Material	**Eulerian Multiphase** **Multiphase Interaction** (Selected automatically)
Eulerian Multiphase Model	**Multiphase Segregated Flow**

Group Box	Model
	Multiphase Equation of State (Selected automatically) **Gradients** (Selected automatically)
Viscous Regime	**Turbulent**
Optional Models	**Gravity**

4. Click **Close**.
5. To review the models, open the **Physics 1** > **Models** node.



6. Save the simulation.

Creating Phases and Selecting Phase Models

Here you create two phases and define the physics models for each. The continuous phase is water and the dispersed phase is hexane. The water and hexane phases are both modeled as liquid with constant density. The Adaptive Multiple Size-Group (AMUSIG) model is used to predict the size distribution of hexane droplets in the flow.

To create the phases and select the phase models:

Create the continuous liquid phase, water.

1. In the **Physics 1** continuum, right-click the **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and select **New**.
2. Rename the **Phase 1** node to **Water**.

The Reynolds Stress Turbulence model is applied as the turbulence in the water phase is anisotropic.

3. For the **Water** phase, right-click the **Models** node and select the following models in order:

Group Box	Model
Enabled Models	**Turbulent** (selected automatically) **Flow** (selected automatically)
Material	**Liquid** **Reynolds-Averaged Navier-Stokes** (selected automatically)
Equation of State	**Constant Density**
Reynolds-Averaged Turbulence	**Reynolds Stress Turbulence**
Reynolds Stress Turbulence Models	**Linear Pressure Strain Two-Layer** **Exact Wall Distance** (selected automatically) **Two-Layer All y+ Wall Treatment** (selected automatically)

4. Click **Close**.

To define the dispersed liquid phase, hexane:

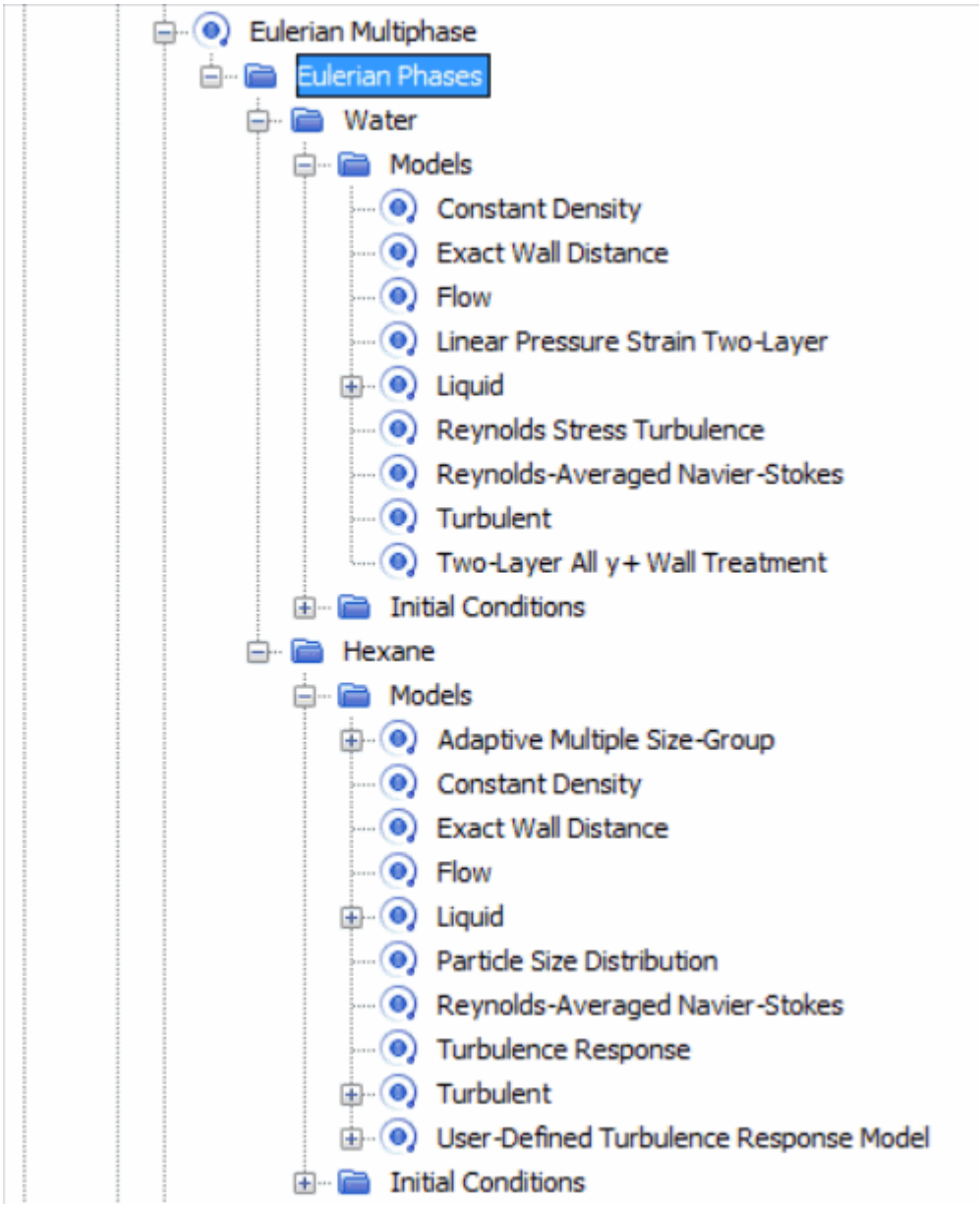
5. Create a second phase and rename it to **Hexane**.

6. For the **Hexane** phase, select the following models in order:

Group Box	Model
Enabled Models	**Turbulent** (selected automatically) **Flow** (selected automatically)
Material	**Liquid** **Reynolds-Averaged Navier-Stokes** (selected automatically)
Equation of State	**Constant Density**
Reynolds-Averaged Turbulence	**Turbulence Response**
Turbulence Response Models	**User-Defined Turbulence Response Model** **Exact Wall Distance** (selected automatically)
Optional Models	**Particle Size Distribution**
Particle Size Distribution	**Adaptive Multiple Size-Group**

7. Click **Close**.

   The tree below the **Eulerian Phases** node now appears as shown below.



For the dispersed phase, replace the default fluid with Hexane.

8. Right-click **Eulerian Phases** > **Hexane** > **Models** > **Liquid** > **H2O** and click **Replace with...**.
9. In the *Replace Material* dialog, select **C6H14 (Hexane)** and then click **OK**.

   The default material properties of both fluids are suitable for this simulation.

To specify the Adaptive Multiple Size-Group properties for the dispersed hexane phase:

10. Edit the **Eulerian Phases** > **Hexane** > **Models** node and set the following properties:

Node	Property	Setting
Adaptive Multiple Size-Group	*Number of Size-Groups*	`5`
	Velocity Option	**Multi-Speed**
	Size Distribution Diffusion Limiter	`1.0E9`

The *Size Distribution Diffusion Limiter* is set to a high value in order to mitigate diffusion caused by high size gradients on a coarse grid.

11. Save the simulation.

Defining the Phase Interactions

Using the **Multiphase Interaction** model, you define the interaction between the continuous water phase and the dispersed hexane phase. Phase interaction models account for droplet breakup and coalescence.

For turbulent breakup, two models are available: **Martinez-Bazan Fragments Breakage Rate** and **Tsouris and Tavlarides Breakage Rate**. The **Tsouris and Tavlarides Breakage Rate** is the most suitable model for this tutorial case as it does not have a minimum particle diameter for breakup. The model specifies a breakup probability that decreases exponentially with droplet diameter.

The **Luo Coalescence Efficiency** model is the only option available within the AMUSIG model for coalescence in turbulent conditions. The model specifies a higher probability of coalescence due to the longer contact time caused by turbulence fluctuations, and a shorter bubble deformation time.

To define the phase interactions:

1. In the **Physics 1** continuum, right-click the **Models** > **Multiphase Interaction** > **Phase Interactions** node and create a phase interaction.
2. Right-click **Phase Interaction 1** > **Models** and click **Select Models**. Choose the following models:

Group Box	Model
Phase Interaction Topology	**Continuous-Dispersed Phase Interaction**
Enabled Models	**Drag Force** (Selected automatically) **Multiphase Material** (Selected automatically) **Interaction Length Scale** (Selected automatically) **Interaction Area Density** (Selected automatically)

3. Click **Close**.
4. Select the **Phase Interaction 1** > **Models** > **Continuous-Dispersed Phase Interaction** node and set the following properties:

Property	Setting
Continuous Phase	**Water**
Dispersed Phase	**Hexane**

5. Again, right-click **Phase Interaction 1** > **Models** and click **Select Models**. Choose the following additional models:

Group Box	Model
Optional Models	**Turbulent Breakup** **Multiple Size-Group Breakup** (Selected automatically)
Breakage Rate	**Tsouris and Tavlarides Breakage Rate** **Tsouris and Tavlarides Daughter Particle Size Distribution** (Selected automatically)
Optional Models	**Turbulent Coalescence** **Multiple Size-Group Coalescence** (Selected automatically) **Turbulent Collision Rate** (Selected automatically) **Luo Coalescence Efficiency** (Selected automatically)
Optional Models	**Turbulent Dispersion Force** **Virtual Mass Coefficient** (Selected automatically)
Optional Models	**Virtual Mass Force**

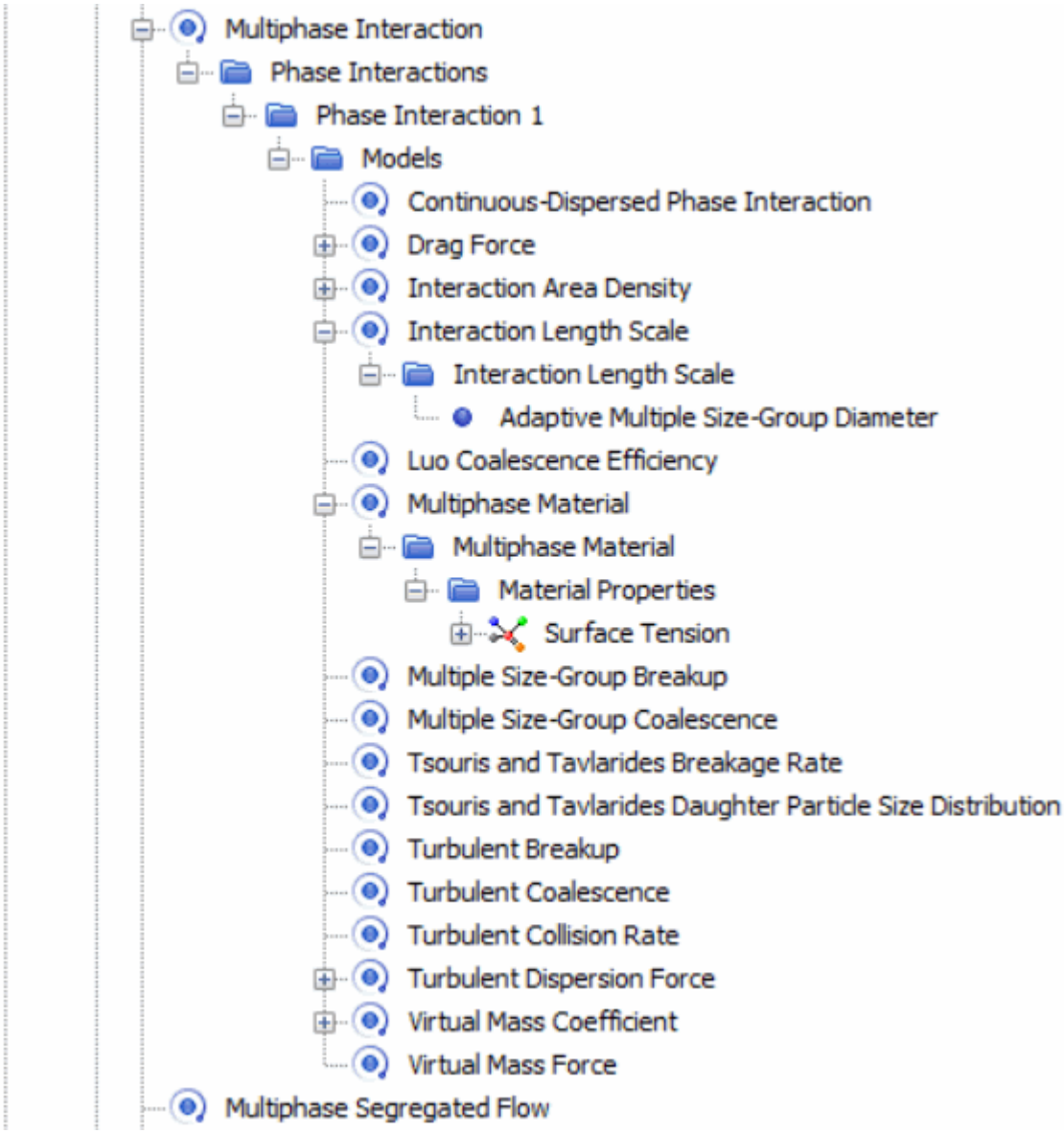
6. Click **Close**.

A length scale is required for interaction area and transfers. The **Adaptive Multiple Size-Group Diameter** method is the usual and default method for AMUSIG simulations. When this option is selected in a multi-speed simulation, each size-group automatically uses its own group diameter for the drag force and for other interaction models.

The interfacial surface tension between the hexane droplet and the surrounding water is also important. This value is used in the calculation of the Weber number—the ratio of dynamic pressure (due to turbulence) to surface tension. A high Weber number implies a higher breakup rate for the droplets.

7. Edit the **Phase Interaction 1** > **Models** node and set the following properties:

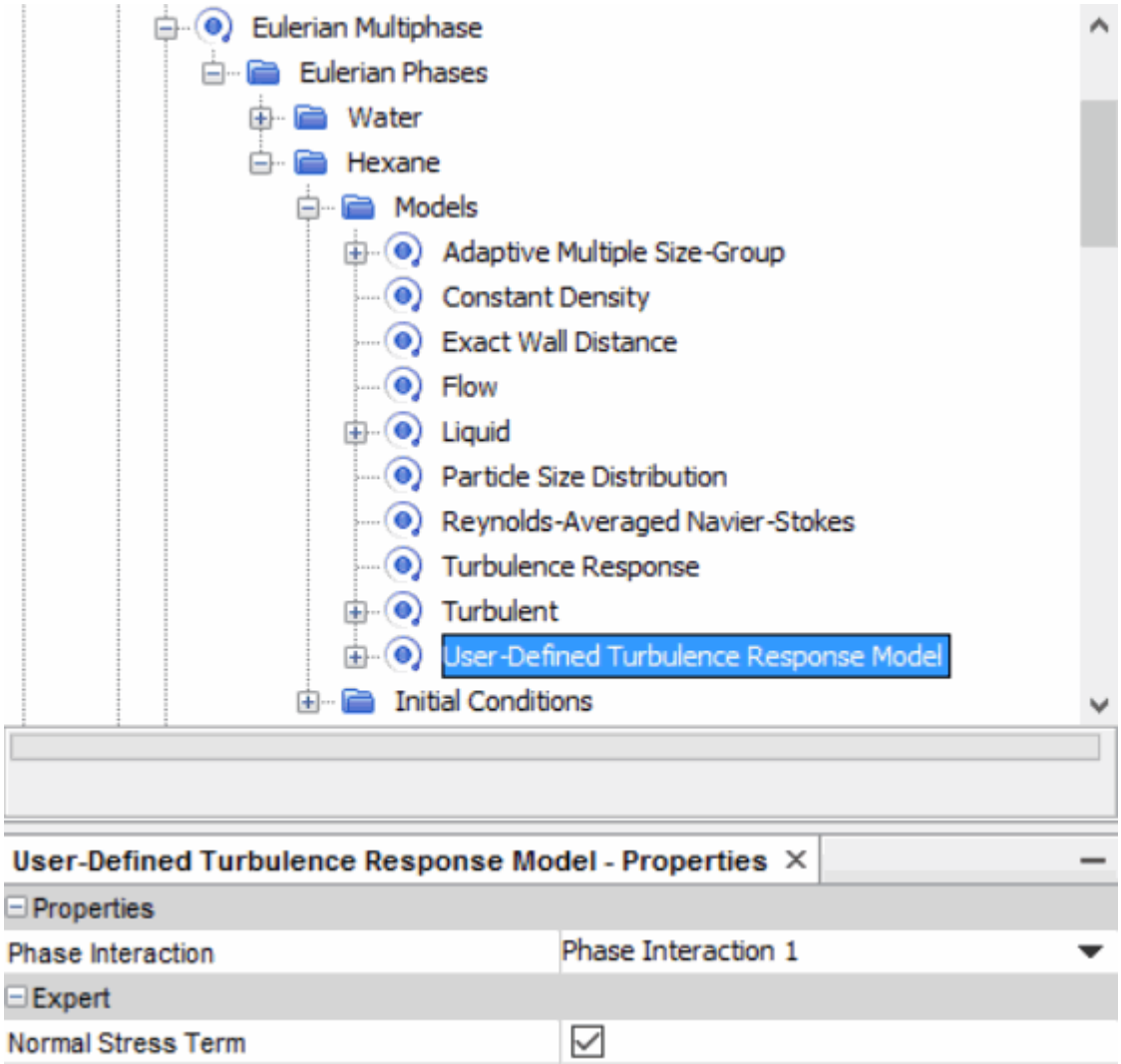
Node	Property	Setting
Interaction Length Scale > **Interaction Length Scale**	**Method**	**Adaptive Multiple Size-Group Diameter**
Multiphase Material > **Material Properties** > **Surface Tension** > **Constant**	**Value**	`0.051 N/m`



Configure the turbulence response of the Hexane phase. Specify the phase interaction and, as this simulation uses the Reynolds Stress Turbulence model, activate the *Normal Stress Term*.

8. Select the **Eulerian Phases** > **Hexane** > **Models** > **User-Defined Turbulence Response Model** node and set the following properties:

Property	Setting
Phase Interaction	**Phase Interaction 1**
Normal Stress Term	**Activated**



9. Save the simulation.

Setting Initial Conditions and Reference Values

The simulation requires initial conditions for each phase and reference values for gravity. The initial hexane droplet size distribution is uniform within a specified diameter range and zero outside the range.

To set the initial conditions and reference values:

1. Edit the **Continua** > **Physics 1** > **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and set the following properties:

Node	Property	Setting
Water > **Initial Conditions**		
∟ **Turbulence Intensity**	**Value**	`0.1`
∟ **Turbulence Specification**	**Method**	**Intensity + Length Scale**

Node	Property	Setting
⌊ **Volume Fraction**	**Value**	`0.99`
Hexane > **Initial Conditions**		
⌊ **Particle Size Distribution**	**Option**	**Uniform distribution**
⌊ **Maximum Diameter**	**Value**	`0.003 m`
⌊ **Minimum Diameter**	**Value**	`0.002 m`
⌊ **Volume Fraction**	**Value**	`0.01`

Set the reference values for the **Physics 1** continuum. To simulate a vertical pipe, set the gravity vector along the X-axis. As the reference density is used to evaluate the buoyancy source term, it is set to the density of the water-hexane mixture. The density of hexane is `655 kg/m^3` and the volume fraction of hexane in the mixture is 0.01, so the mixture density is `996.5 kg/m^3`.

2. Edit the **Reference Values** node and set the following properties:

Node	Property	Setting
Gravity	**Value**	`[-9.81, 0.0] m/s^2`
Reference Density	**Value**	`996.5 kg/m^3`

3. Save the simulation.

Setting Boundary Conditions

The boundaries that represent the pipe are a wall (the pipe exterior wall) and an axis of symmetry (the center of the pipe). One end of the pipe is a velocity inlet, and the other end of the pipe is a pressure outlet.

At the inlet, you set the flow velocity to 0.5 m/s and set the volume fraction to 0.99 water and 0.01 hexane. In a high-turbulence case, turbulence intensity values typically range from 0.05 to 0.2. This simulation uses a turbulence intensity value of 0.1.

To set the boundary conditions:

1. Edit the **Regions** > **Region_2D** > **Boundaries** > **Body 1.In** > **Phase Conditions** node and set the following properties:

Node	Property	Setting
Water > **Physics Conditions** > **Turbulence Specification**	**Method**	**Intensity + Length Scale**
Water > **Physics Values**		
⌊ **Turbulence Intensity**	**Value**	`0.1`
⌊ **Velocity Magnitude**	**Value**	`0.5 m/s`

Node	Property	Setting
⌊ **Volume Fraction**	**Value**	0.99
Hexane > **Physics Values**		
⌊ **Maximum Diameter**	**Value**	0.003 m
⌊ **Minimum Diameter**	**Value**	0.002 m
⌊ **Velocity Magnitude**	**Value**	0.5 m/s
⌊ **Volume Fraction**	**Value**	0.01

2. Edit the **Regions** > **Region_2D** > **Boundaries** > **Body 1.Out** > **Phase Conditions** node and set the following properties:

Node	Property	Setting
Water > **Physics Conditions** > **Turbulence Specification**	**Method**	**Intensity + Length Scale**
Water > **Physics Values**		
⌊ **Turbulence Intensity**	**Value**	0.1
⌊ **Volume Fraction**	**Value**	1.0

3. Save the simulation.

Setting Solver Parameters and Stopping Criteria

The **Multiple Size-Group** solver default under-relaxation settings give reliable convergence in one-speed cases. However, cases that have a high particle concentration, or multi-speed cases where each size-group can travel in a different direction, require additional relaxation. In this simulation, values of 0.7 and 0.5 respectively are appropriate.

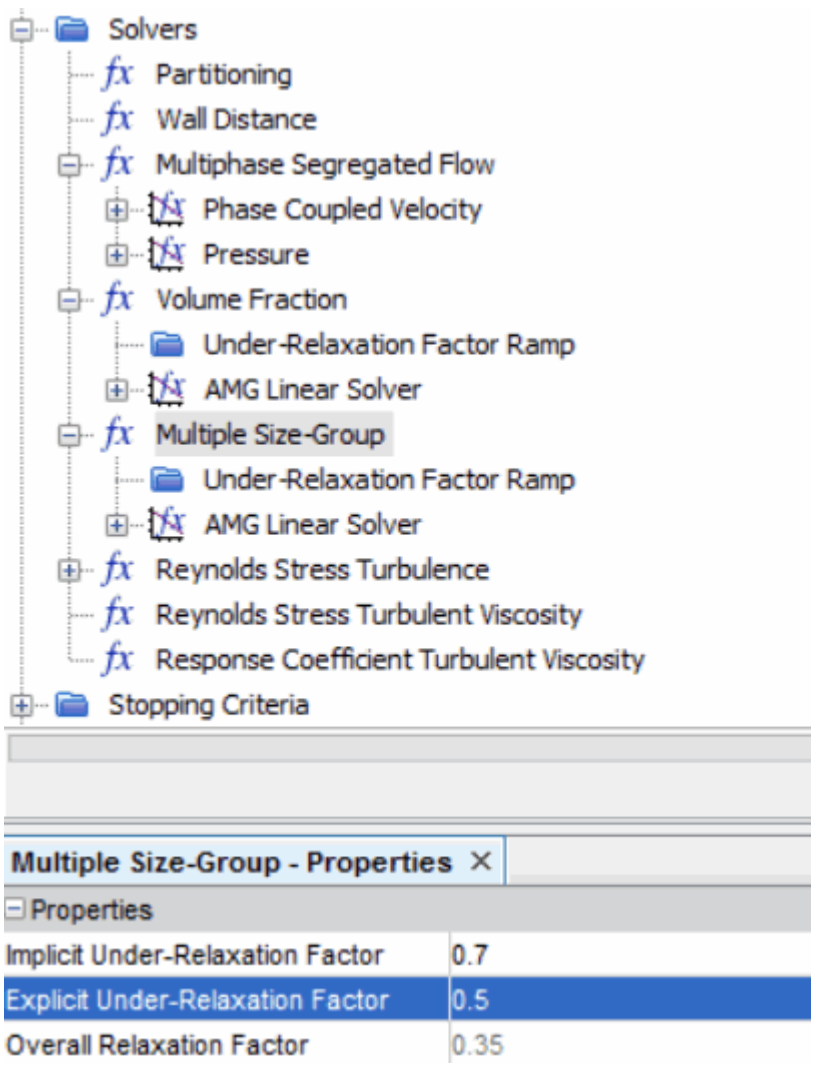
The under-relaxation factors for the other solvers are set as high as possible. In this case, most settings are lower than the default values.

To set the solver parameters and stopping criteria:

1. Edit the **Solvers** node and set the following properties:

Node	Property	Setting
Multiphase Segregated Flow > **Phase Coupled Velocity**	**Implicit Under-Relaxation Factor**	0.8
	Explicit Under-Relaxation Factor	0.4
Multiphase Segregated Flow > **Pressure**	**Under-Relaxation Factor**	0.1

Node	Property	Setting
Volume Fraction	**Implicit Under-Relaxation Factor**	0.8
	Explicit Under-Relaxation Factor	0.2
Multiple Size-Group	**Implicit Under-Relaxation Factor**	0.7
	Explicit Under-Relaxation Factor	0.5



This simulation requires about 1500 iterations to achieve reasonable convergence.

2. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Steps	**Enabled**	Activated
	Maximum Steps	`1500`

3. Save the simulation.

Setting Up the Plots

To monitor the progress of the simulation, you create plots of droplet diameter.

You create the following plots:

- Group Diameters plot

  Plots the droplet diameter of each size-group along the X-axis (the center of the pipe). As each size-group contains an equal volume fraction of hexane, this plot gives information about the development of the droplet size distribution as the flow progresses along the pipe.
- Mean Diameters plot

  Plots the mean diameter of the hexane droplets along the X-axis. Three different measures of mean diameter are plotted: Surface Mean Diameter, Volume Based Diameter, and Volume Mean Diameter. Differences in values between these diameters are indicators of non-uniformity in the size distribution.

For more information, see the Adaptive Multiple Size-Group (AMUSIG) theory.

Create the **Group_Diameters** plot:

1. Click (**Create/Open Plots**) in the **Plot** toolbar and select **XY Plot**.
2. In the simulation tree, rename the **Plots** > **XY Plot 1** node to `Group_Diameters`.
3. Select the **Group_Diameters** node and set **Parts** to **Region_2D** > **Boundaries** > **Body 1.axis**.
4. Right-click the **Group_Diameters** > **Y Types** node and click **New**.

   A new sub-node is added, **Y Type 2**.
5. Repeat the previous step three more times to create the sub-nodes **Y Type 3**, **Y Type 4**, and **Y Type 5**.
6. Select the **Group_Diameters** > **Y Types** > **Y Type 1** node and set the following properties:

Node	Property	Setting
Scalar Function	**Field Function**	**Group Diameter** > **GroupDiameter of Hexane SizeGroup 0**
Region_2D: Body 1.axis	**Sort Plot Data**	Activated
∟ **Line Style**	**Style**	**Solid**

7. Repeat the previous step for the other four Y Type sub-nodes.

   The settings for these nodes are the same, except for the **Field Function** property. Set this property on these nodes as follows:

Y Type Node	Property	Setting
Y Type 2	**Field Function**	**GroupDiameter of Hexane SizeGroup 1**
Y Type 3	**Field Function**	**GroupDiameter of Hexane SizeGroup 2**
Y Type 4	**Field Function**	**GroupDiameter of Hexane SizeGroup 3**
Y Type 5	**Field Function**	**GroupDiameter of Hexane SizeGroup 4**

8. Select the **Group_Diameters** > **Axes** node and set the following properties:

Node	Property	Setting
Bottom Axis > **Title**	**Title**	`Location on X-axis (m)`
Left Axis > **Title**	**Title**	`Group Diameter (m)`

Create the **Mean_Diameters** plot:

9. Create another XY plot and rename the **Plots** > **XY Plot 1** node to `Mean_Diameters`.
10. Select the **Mean_Diameters** node and set **Parts** to **Region_2D** > **Boundaries** > **Body 1.axis**
11. Right-click the **Mean_Diameters** > **Y Types** node and click **New**.

    A new sub-node is added, **Y Type 2**.
12. Repeat the previous step to create the sub-node **Y Type 3**.
13. Select the **Mean_Diameters** > **Y Types** > **Y Type 1** node and set the following properties:

Node	Property	Setting
Scalar Function	**Field Function**	**Surface Mean Diameter** > **Surface Mean Diameter of Hexane**
Region_2D: Body 1.axis	**Sort Plot Data**	Activated
⌊ **Line Style**	**Style**	**Solid**

14. Repeat the previous step for the other two Y Type sub-nodes.

    The settings for these additional nodes are the same, except for the **Field Function** property. Set this property on these nodes as follows:

Y Type Node	Property	Setting
Y Type 2	**Field Function**	**Volume Based Diameter** > **Volume Based Diameter of Hexane**
Y Type 3	**Field Function**	**Volume Mean Diameter** > **Volume Mean Diameter of Hexane**

15. Select the **Mean_Diameters** > **Axes** node and set the following properties:

Node	Property	Setting
Bottom Axis > **Title**	**Title**	`Location on X-axis (m)`
Left Axis > **Title**	**Title**	`Mean Diameter (m)`

16. Save the simulation.

Visualizing the Solution

To visualize the breakup rate and the coalescence rate of the hexane droplets, and the surface mean diameter of the droplets, you create scalar scenes. The pipe width (Y axis) is scaled by a factor of 2 in order to provide a better view of the results.

Create a transform to scale the Y axis by a factor of 2:

1. Right-click the **Tools** > **Transforms** node, and select **New Graphics Transform** > **Simple Transform**.
2. Rename the new transform to `Y Axis Scale`.
3. Select the **Y Axis Scale** node and set the *Scale* to `[1.0, 2.0, 1.0]`.

To visualize the breakup rate:

4. Right-click the **Scenes** node, and select **New Scene** > **Scalar**.
5. Rename the new scene to `Breakup Rate`.
6. Select the **Breakup Rate** > **Displayers** node and set the following properties:

Node	Property	Setting
Outline 1	**Transform**	**Y Axis Scale**
Scalar 1	**Contour Style**	**Smooth Filled**
	Transform	**Y Axis Scale**
∟ **Scalar Field**	**Function**	**Breakup Rate** > **Breakup Rate of Phase Interaction 1**

To visualize the coalescence rate:

7. Create another scalar scene and rename it to `Coalescence Rate`.
8. Select the **Coalescence Rate** > **Displayers** node and set the following properties:

Node	Property	Setting
Outline 1	**Transform**	**Y Axis Scale**
Scalar 1	**Contour Style**	**Smooth Filled**
	Transform	**Y Axis Scale**
∟ **Scalar Field**	**Function**	**Coalescence Rate** > **Coalescence Rate of Phase Interaction 1**

To visualize the surface mean diameter of hexane:

9. Create another scalar scene and rename it to `Surface Mean Diameter`.
10. Select the **Surface Mean Diameter** > **Displayers** node and set the following properties:

Node	Property	Setting
Outline 1	**Transform**	**Y Axis Scale**
Scalar 1	**Contour Style**	**Smooth Filled**
	Transform	**Y Axis Scale**
∟ **Scalar Field**	**Function**	**Surface Mean Diameter** > **Surface Mean Diameter of Hexane**

11. Save the simulation.

Running the Simulation

Preparation of the simulation is now complete, and you can run the simulation.

To run the simulation:

1. Click (**Run**) in the *Solution* toolbar.

   The solution progress is displayed in the *Output* window. The *Residuals* display is created automatically in the *Graphics* window and shows the progress of the solvers.

   While the simulation is running, you can click the tabs at the top of the *Graphics* window to view scenes and plots.

   During the run, you can stop the process by clicking (**Stop**) on the toolbar. If you do stop the simulation, you can click (**Run**) to resume it.

   If left alone, the simulation continues until 1500 iterations are complete and the following message is displayed in the *Output* window:

   `Stopping criterion Maximum Steps satisfied.`
2. When the simulation has finished running, save it.

Visualizing the Results

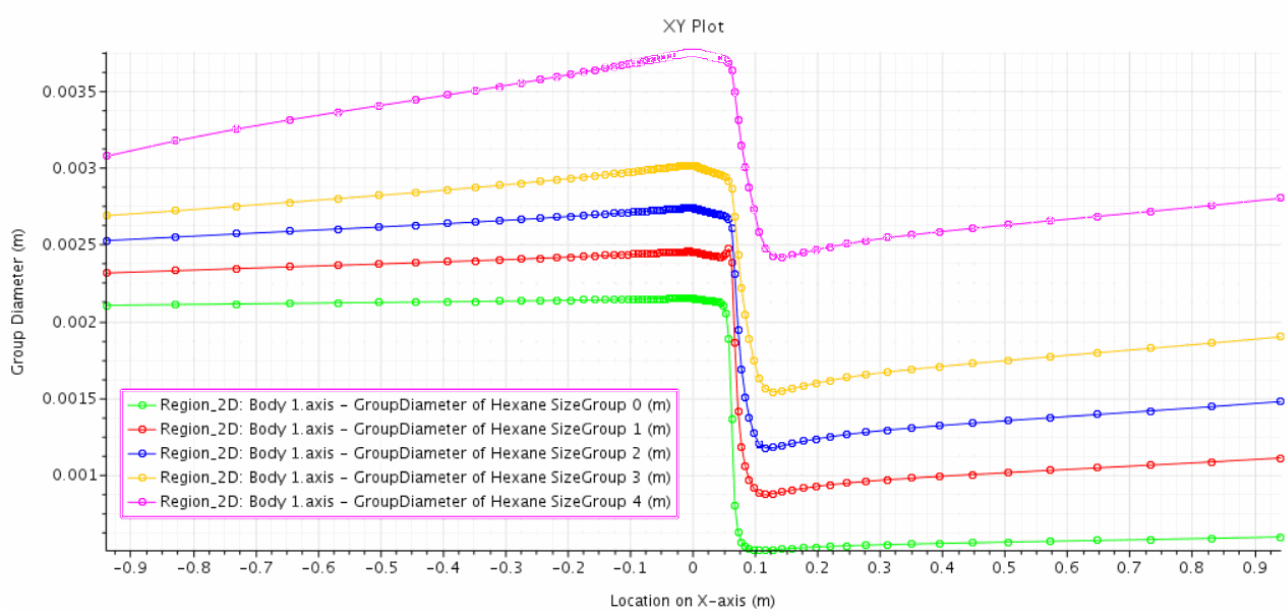
The resulting plots show the development of the size-group diameters as the hexane droplets progress along the pipe. The scenes show the regions where droplet breakup and coalescence are occurring, and the distribution of the droplet mean diameter values in the flow.

If necessary, adjust the scene to zoom in on the area that you want to view.

Group Diameters Plot

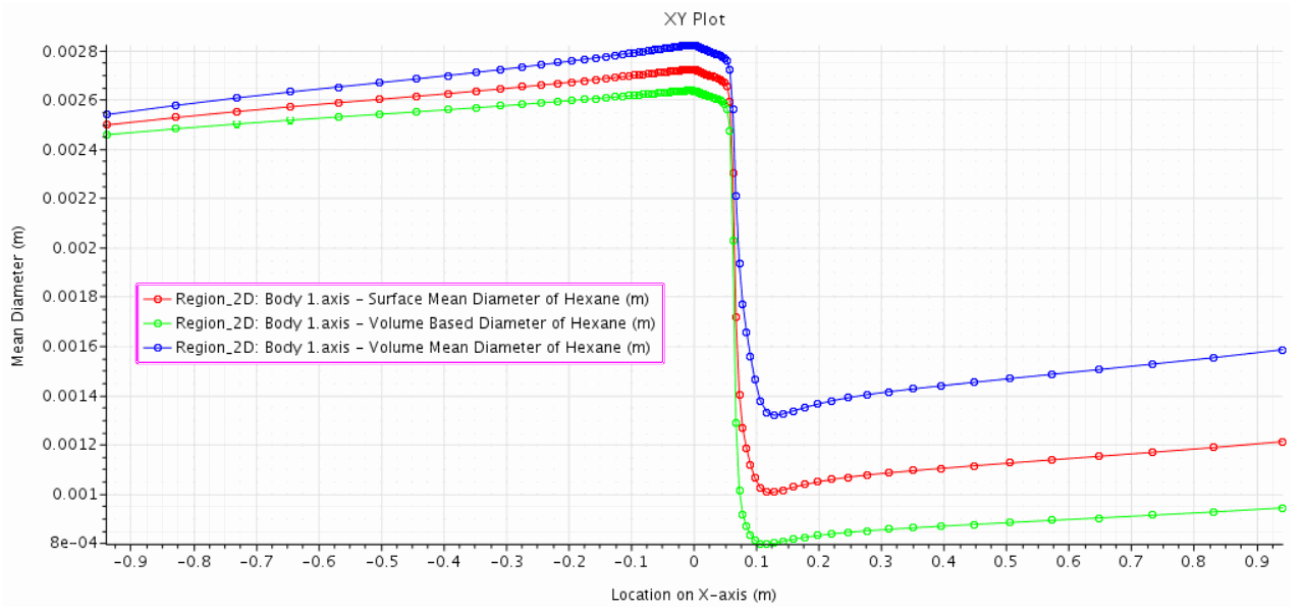
In this plot, some coalescence—especially for large diameters—is observed upstream of the restriction followed by significant breakup as the droplets pass through the region of intense turbulence downstream.

Beyond the turbulent region, droplet coalescence resumes. The droplet size distribution can be inferred from the spacing of the five size-group diameters. The line that is plotted for each group represents the mean volume fraction of that group, so the five lines that are shown in the plot can also be read as 10%, 30%, 50%, 70%, and 90% cumulative volume fractions of hexane.



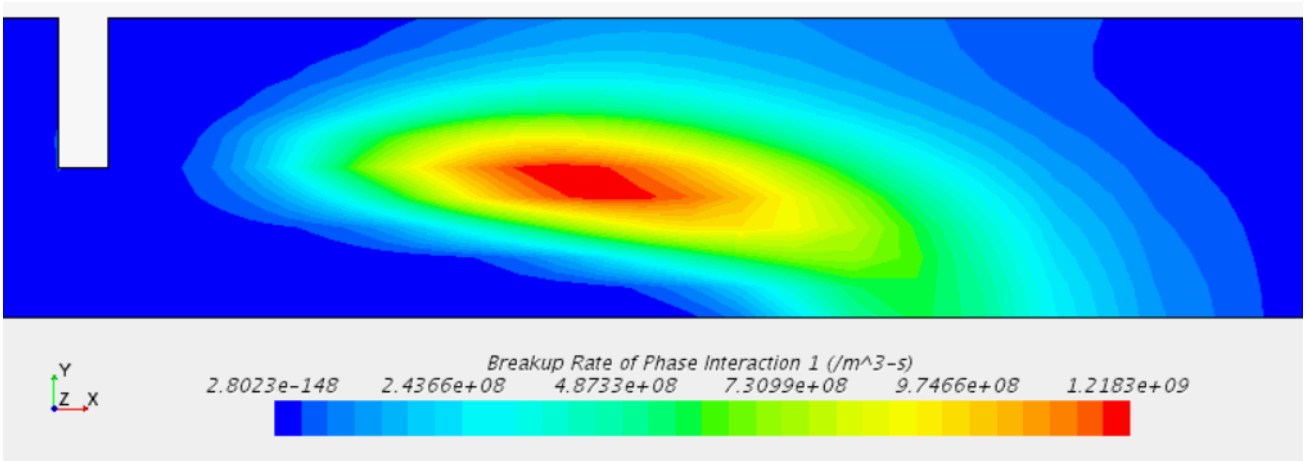
Mean Diameters Plot

In the Mean Diameters plot, the differences between diameters indicate polydispersity of the size distribution.



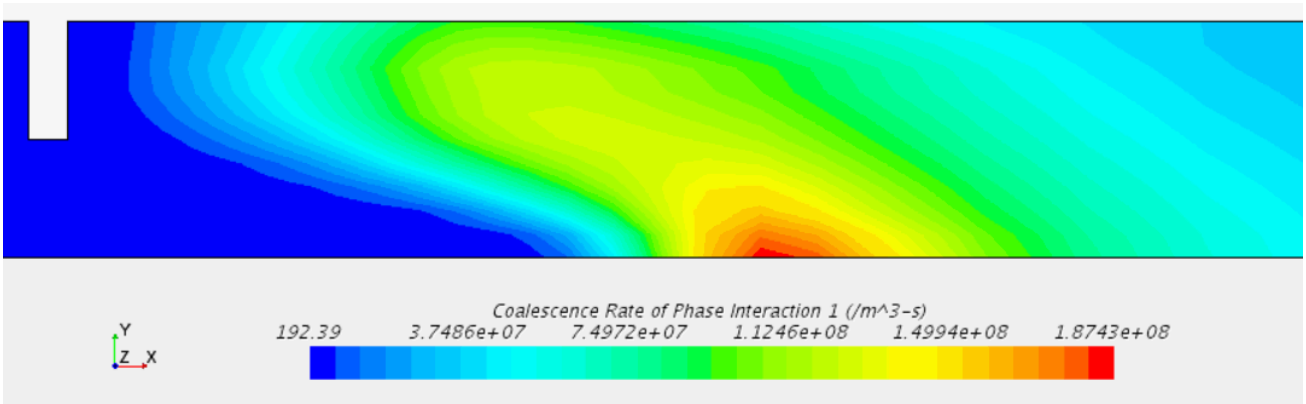
Breakup Rate Scene

Here, you can see a high breakup rate in a localized region of turbulence downstream of the restriction, but practically zero breakup elsewhere.



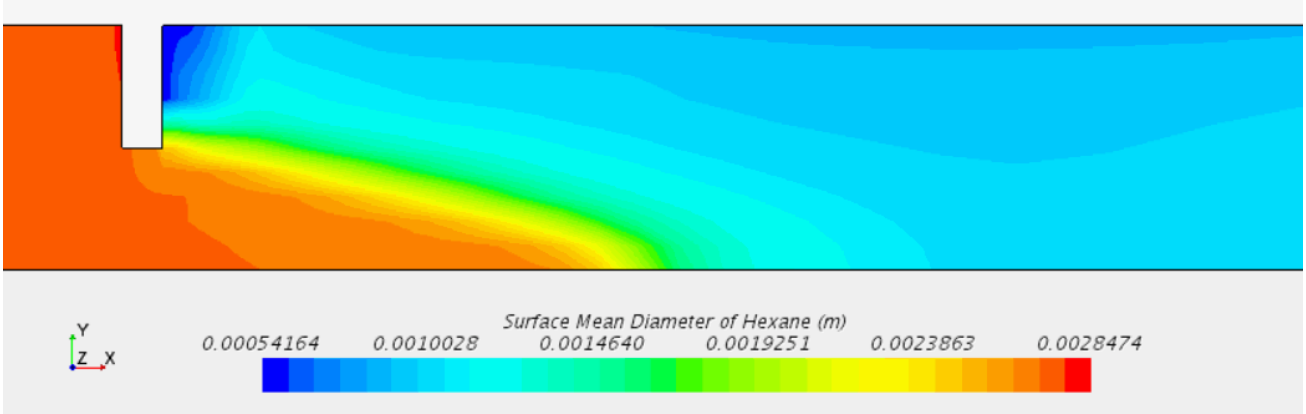
Coalescence Rate Scene

This scene shows a strong peak in the coalescence rate in the area downstream of the region of intense turbulence (where most of the droplet breakup occurs). This increase corresponds to the increased population of smaller droplets.



Surface Mean Diameter Scene

The Surface Mean Diameter of the droplets reduces by half as they pass through the turbulent region of the flow.



Summary

This tutorial demonstrated how to model droplet size distribution with the Adaptive Multiple Size-Group (AMUSIG) model in STAR-CCM+.

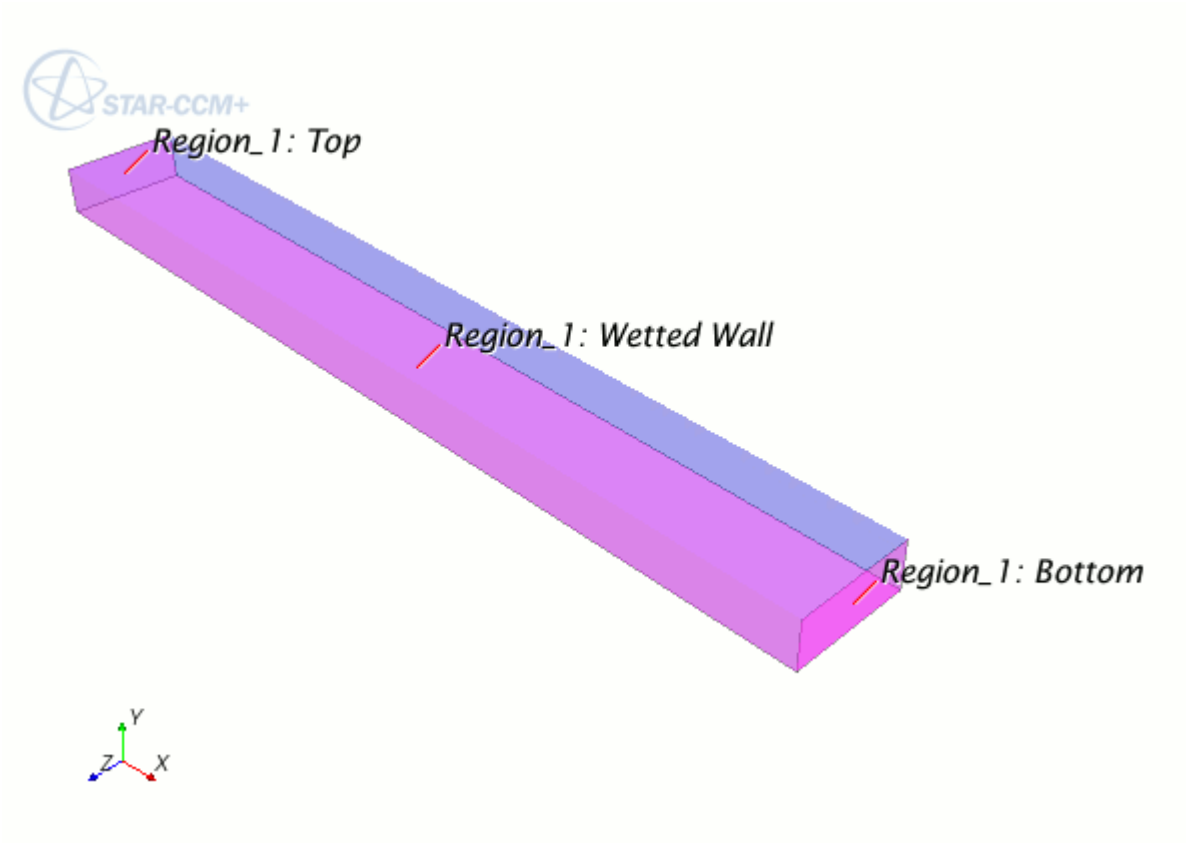
This tutorial has introduced the following STAR-CCM+ features:

- Setting up the physics models.
- Creating phases and selecting the AMUSIG phase models.
- Defining the phase interaction and selecting the AMUSIG breakup and coalescence models.
- Setting the appropriate initial conditions, reference values, and solver properties for an AMUSIG simulation.
- Setting up plots and scenes to visualize the results.
- Running the simulation.
- Viewing and interpreting the results.

Fluid Film: Liquid Film Flow

This tutorial demonstrates how to use the fluid film modeling capability of STAR-CCM+. It simulates the progress of a rectangular film of water flowing down the wall of a three-dimensional duct.

Water flows down the wall (wetted wall in the following figure) due to gravity, forming a liquid film in the process.



The initial water is 1 mm thick and covers a wall length and width of 250 mm and 200 mm, respectively, and is placed at the top wall. The surrounding gas is initially at rest. The simulation is run using the implicit unsteady solver for a simulated time of 10 seconds, which is long enough to satisfy the similarity condition. For the first 1000 time steps, the time step size is set to 0.001 seconds. Thereafter, a time step size of 0.01 seconds is used.

Contents:

Prerequisites
Importing the Mesh and Naming the Simulation
Visualizing the Volume Mesh
Selecting the Physics Models
Setting Fluid Film Models and Initial Conditions

Setting Up Shell Region and Boundary Conditions
Setting Reference Values
Setting Solver Parameters and Stopping Criteria
Visualizing the Solution
Running the Simulation
Visualizing the Results
Summary

Prerequisites

The instructions in the Fluid Film: Liquid Film Flow tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Importing the Mesh and Naming the Simulation

Launch STAR-CCM+, import the supplied geometry, and save the simulation.

To import the mesh and name the simulation:

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh**.
4. In the *Open* dialog, choose `doc/startutorialsdata/multiphaseFlow/data/duct.ccm`, then click **Open**.

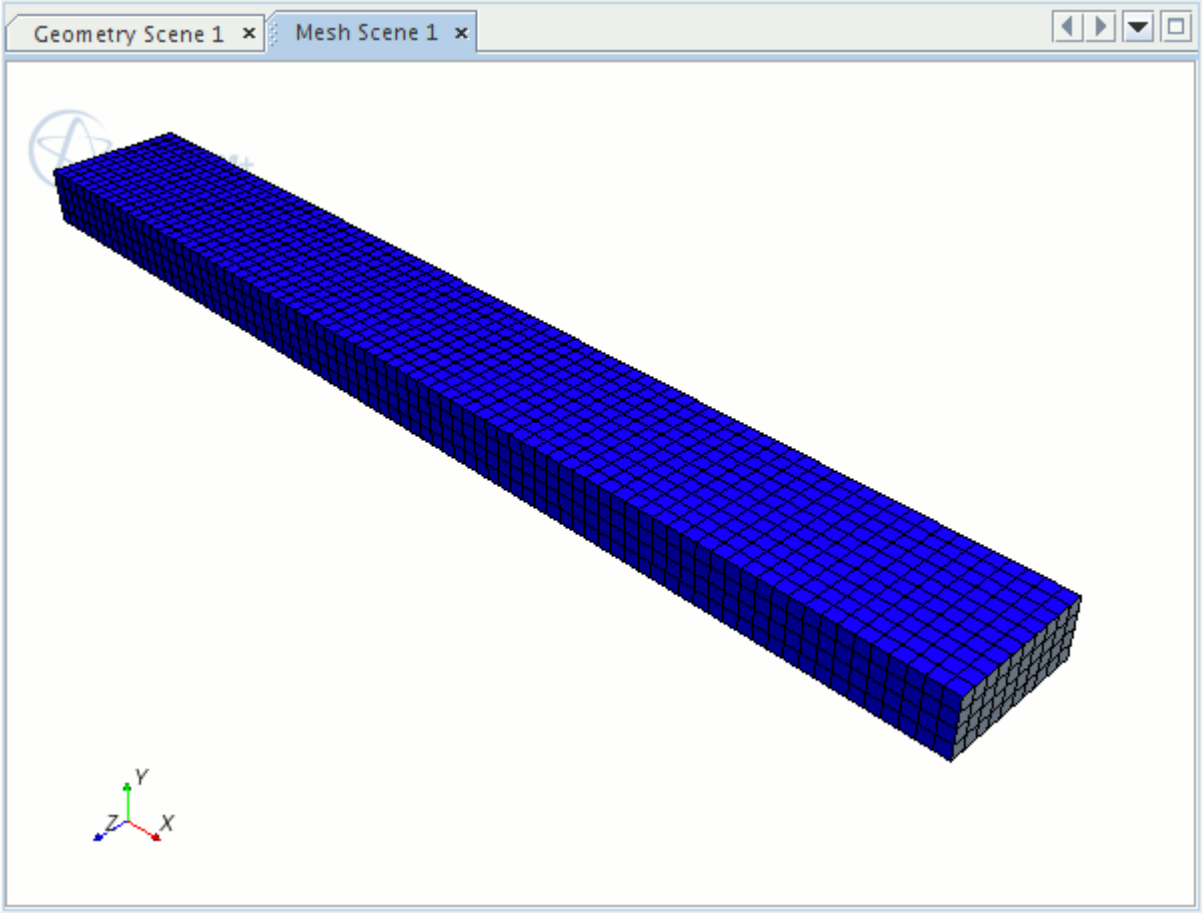
   STAR-CCM+ provides feedback on the import process, which takes a few seconds, in the *Output* window.
5. Save the simulation as `wallFilm.sim`.

Visualizing the Volume Mesh

Examine the imported volume mesh.

To visualize the volume mesh:

1. Right-click the **Scenes** node and select **New Scene** > **Mesh**.
2. Examine the *Mesh Scene 1* display in the *Graphics* window.
3. Click in the toolbar and select **Views** > **+X +Y +Z** > **Up +Y**.
4. Use the mouse buttons, zoom in and position the part so that the view is similar to the view shown below.



5. Click (**Save-Restore-Select views**) in the toolbar and select **Store Current View**.

Selecting the Physics Models

Physics models define the primary variables of the simulation, including pressure, temperature, velocity, and what mathematical formulation is used to generate the solution.

The main phenomena to solve for in this simulation is unsteady, laminar, incompressible, gaseous flow. Set up an implicit unsteady, laminar, incompressible, gaseous flow using the segregated flow and fluid film models.

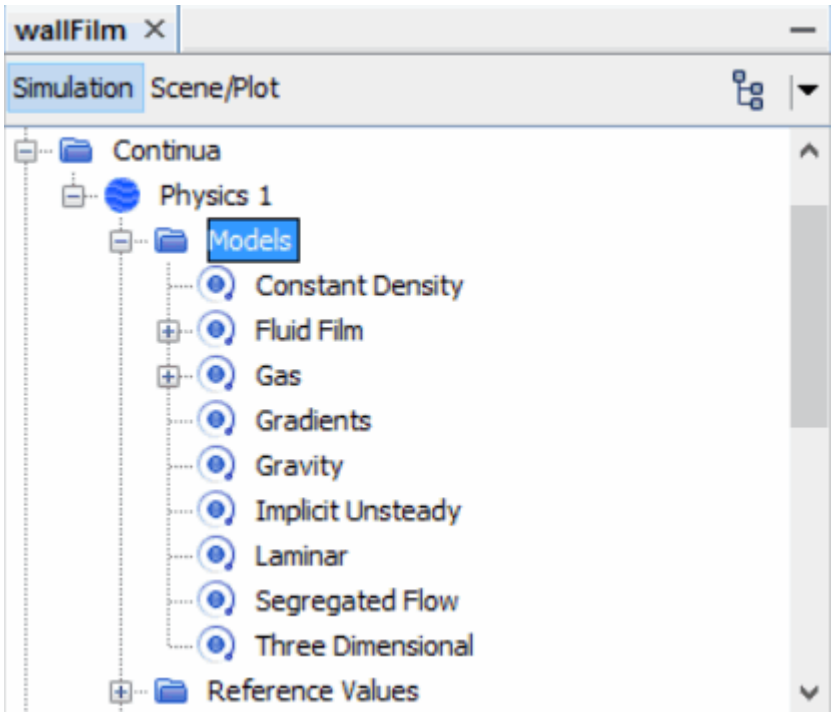
To select the physics models:

1. For the physics continuum, **Continua** > **Physics 1**, select the following models in order:

Group Box	Model
Enabled Models	**Three Dimensional** (selected automatically)
Time	**Implicit Unsteady**
Material	**Gas**
Flow	**Segregated Flow** **Gradients** (selected automatically)

Group Box	Model
Equation of State	**Constant Density**
Viscous Regime	**Laminar**
Optional Models	**Fluid Film** **Gravity**

2. Click **Close**.
3. To review the models, open the **Physics 1** > **Models** node.



4. Save the simulation.

Setting Fluid Film Models and Initial Conditions

Set the fluid film models and the initial conditions for the fluid film phases.

To set the fluid film models and initial conditions:

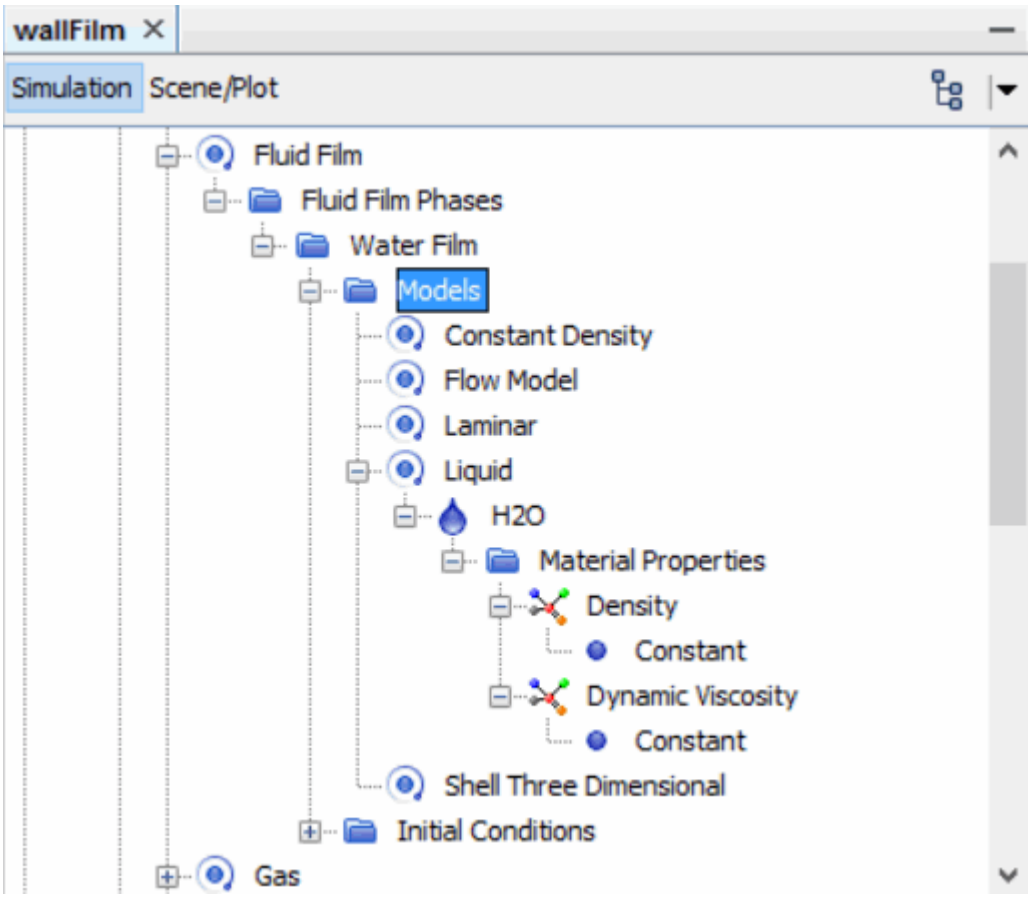
You first define the fluid film phases. In this tutorial, you define only one phase for water.

1. In the **Physics 1** continuum, right-click the **Models** > **Fluid Film** > **Fluid Film Phases** node and create a new phase.
2. Rename the **Phase 1** node to **Water Film**.
3. For the **Water Film** phase, select the following models in order:

Group Box	Model
Space	**Shell Three-Dimensional** **Flow Model** (selected automatically) **Laminar** (selected automatically)
Material	**Liquid**
Equation of State	**Constant Density**

4. Click **Close**.

The expanded **Water Film** > **Models** node is shown below.



You specify the initial water thickness with a field function, as you only want film to appear initially at the top of the geometry. Assume that the initial water is 1 mm thick and covers a wall length and width of 250 mm and 200 mm, respectively.

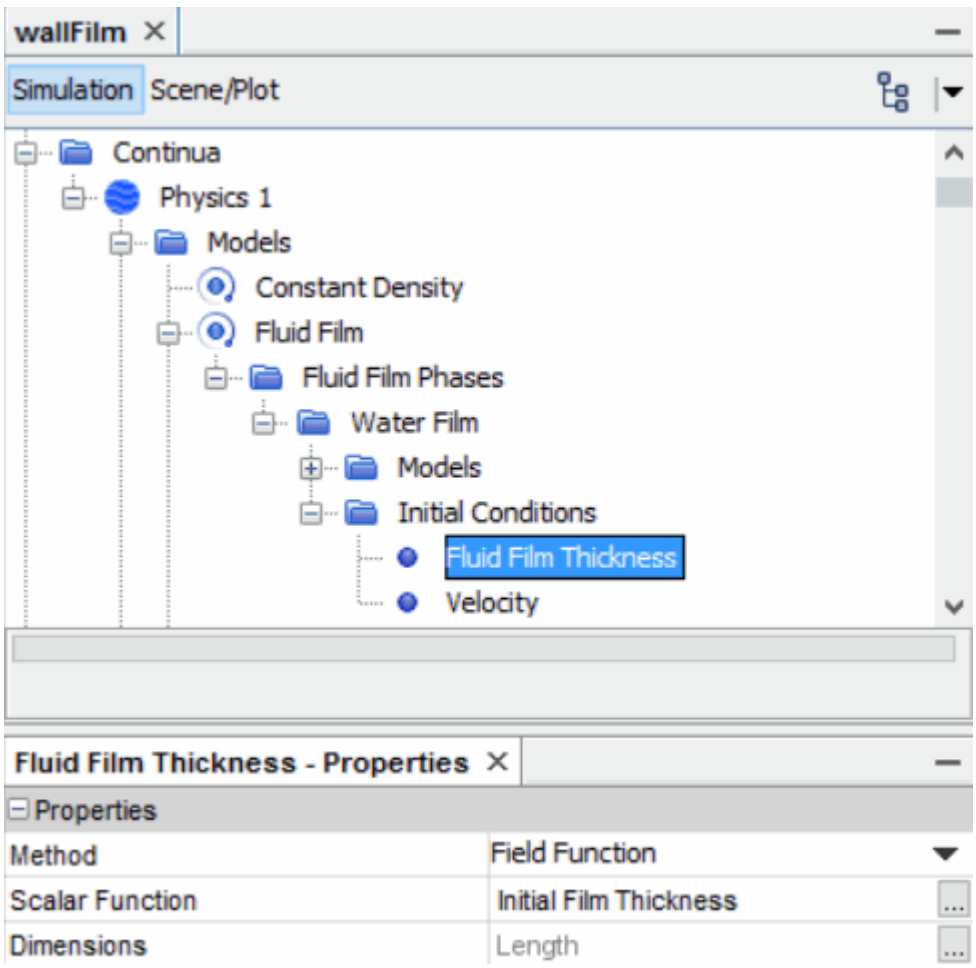
5. Define the field function to use for specifying the initial water thickness:
   a) Right-click the **Tools** > **Field Functions** node and select **New** > **Scalar**.
   b) Rename the **User Field Function 1** node to `Initial Film Thickness`.
   c) Select the **Initial Film Thickness** node and set the following properties:

Property	Setting
Dimensions	**Length**
Function Name	`InitialFilmThickness`
Definition	`(($$Centroid[0]<0.25) && ($$Centroid[2]>0.025) && ($$Centroid[2]<0.225)) ? 0.001: 0`

Set the initial conditions for the fluid film.

6. Edit the **Water Film** > **Initial Conditions** node and set the following properties:

Node	Property	Setting
Fluid Film Thickness	**Method**	**Field Function**
	Scalar Function	**Initial Film Thickness**



7. Save the simulation.

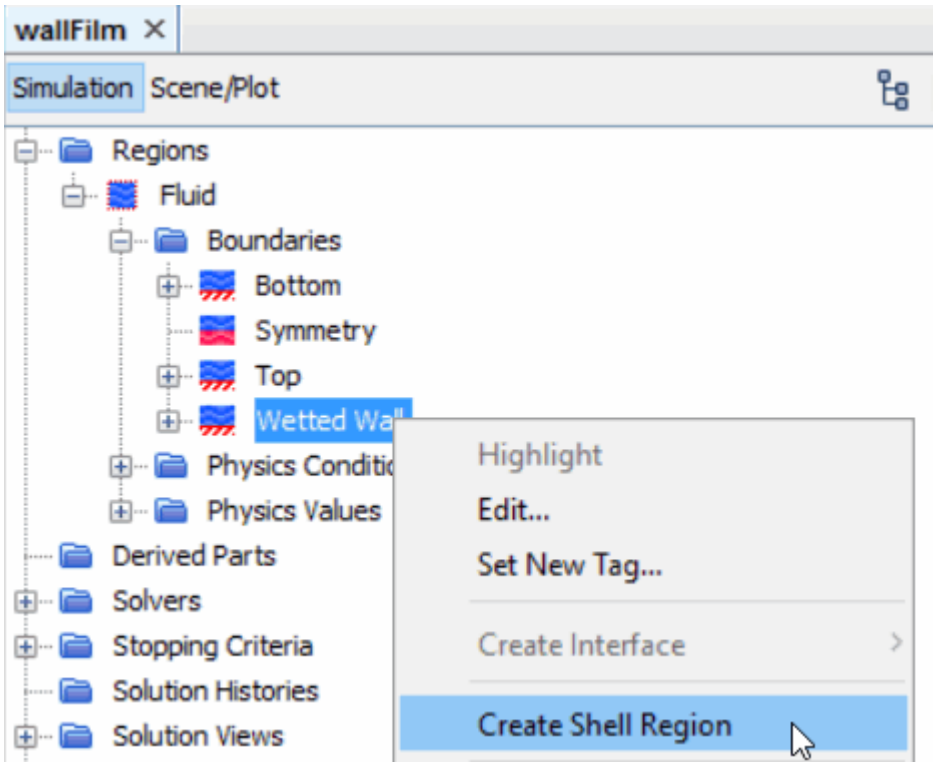
Setting Up Shell Region and Boundary Conditions

Create a shell region and define the boundary conditions.

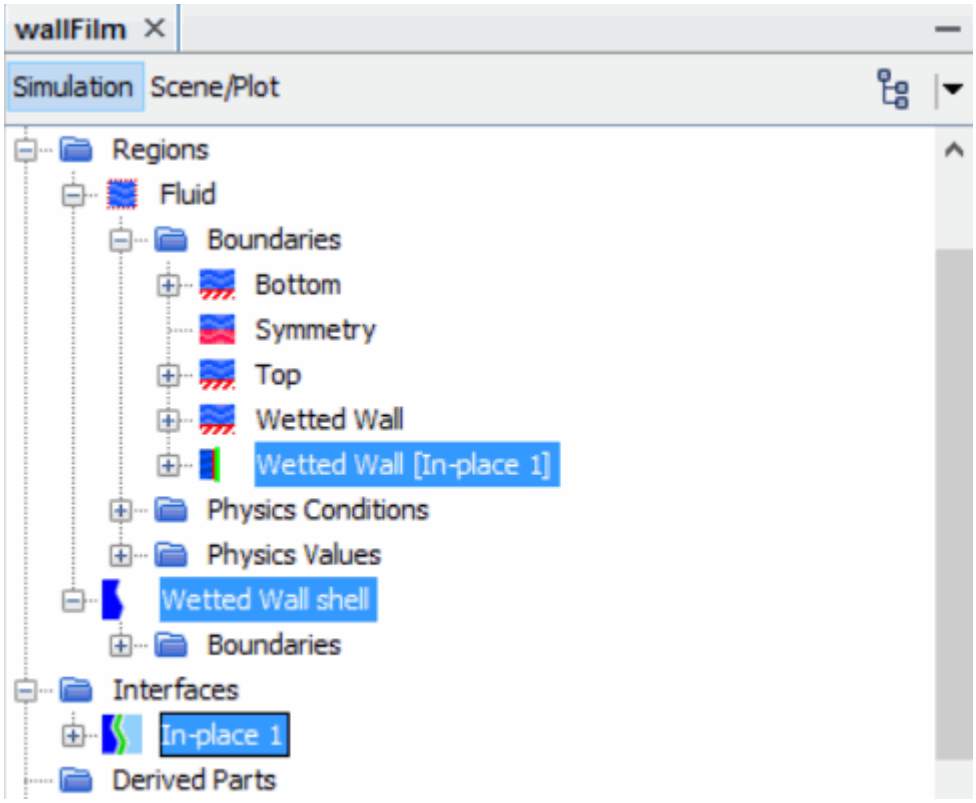
In this tutorial liquid film forms on the **Wetted Wall** boundary of the **Fluid** region. A shell region is required for all boundaries on which fluid film is to flow. This shell region is created from the Wetted Wall boundary.

To create a shell region and define the boundary conditions:

1. Right-click the **Regions** > **Fluid** > **Boundaries** > **Wetted Wall** node and select **Create Shell Region**.

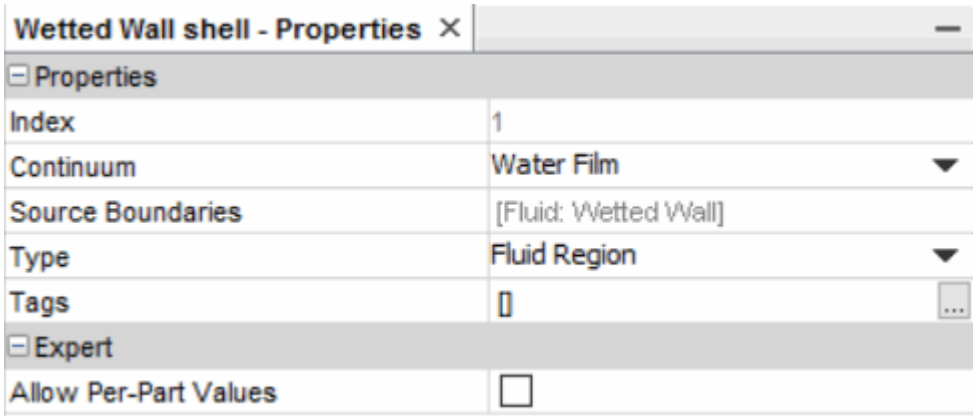


A new shell region is added to the simulation tree. In addition, an interface is formed between the fluid region and the shell region, which is based on the geometry of the Wetted Wall boundary surface. The original boundary therefore becomes an interface to the film shell region.



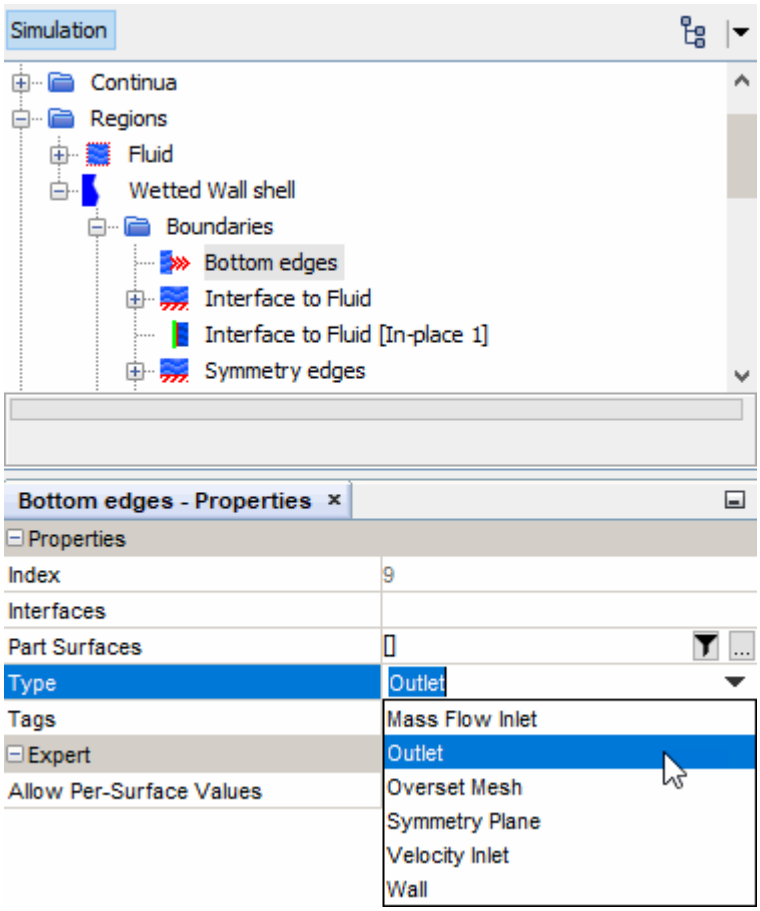
The water film phase must be assigned to the **Wetted Wall Shell** region.

2. Select the **Regions** > **Wetted Wall Shell** node and set the **Continuum** to **Water Film**.



Specify the actual fluid film boundary conditions using the nodes within the Wetted Wall Shell region. These boundaries are formed from the edges of the original Wetted Wall boundary from which the shell region was created. By default all boundaries are set to no-slip wall, except the Interface to Fluid [In-place 1], which is set to the Fluid-Film Boundary type. So that the fluid film can leave the domain, you change the boundary type of the Bottom edges boundary. Set the type for this boundary to Outlet.

3. Select the **Regions** > **Wetted Wall Shell** > **Boundaries** > **Bottom edges** node and set **Type** to **Outlet**.



4. Save the simulation.

Setting Reference Values

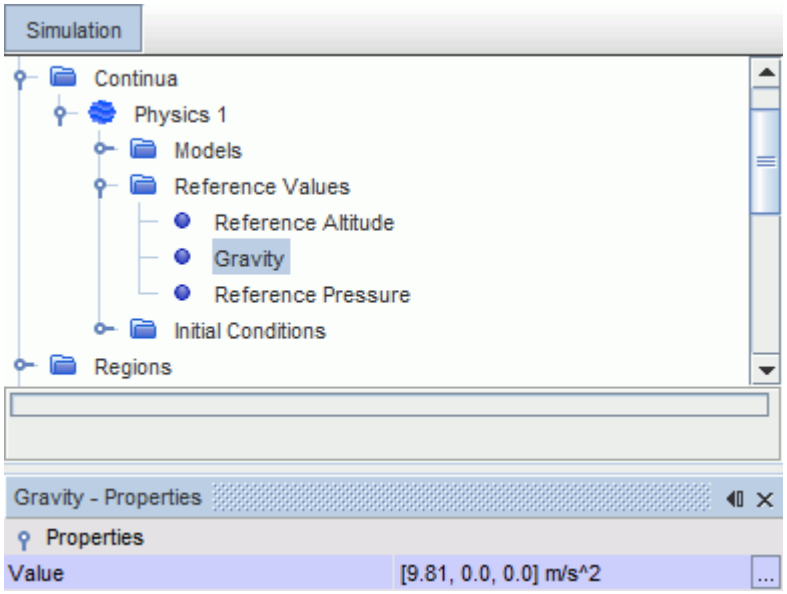
Set the reference values for the **Physics 1** continuum.

The reference values of the **Physics 1** continuum must be defined to make sure that the acceleration due to gravity acts in the expected direction. In this tutorial, gravity acts in the positive X direction.

To set the reference values:

1. Edit the **Physics 1** > **Reference Values** node and set the following property:

Node	Property	Setting
Gravity	**Value**	`[9.81, 0.0, 0.0] m/s^2`



Setting Solver Parameters and Stopping Criteria

Set the appropriate solver parameters and stopping criteria for the simulation.

By default the time step size is set to 1 s. Modify this setting and prescribe a variable time step size according to the following rule:

- When time < 1 s, time step = 0.001 s
- Otherwise, time step=0.01 s

To set the solver parameters and stopping criteria:

1. Edit the **Solvers** > **Implicit Unsteady** node and set the following properties:

Property	Setting
Time-Step	`($Time<1) ? 0.001 : 0.01`
Temporal Discretization	**2nd-order**

2. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Physical Time	**Enabled**	Activated
	Maximum Physical Time	`10 s`
Maximum Inner Iterations	**Maximum Inner Iterations**	`5`
Maximum Steps	**Enabled**	Deactivated

3. Save the simulation.

Visualizing the Solution

Create reports, monitors, and plots to view the results of the solution.

The fluid film thickness can be monitored in a number of ways. In this section, create a report, monitor, and plot of the maximum film thickness, and an X-Y plot of film thickness along the vertical surface.

Contents:

Creating a Maximum Report, Monitor, and Plot
Creating an X-Y Plot of Fluid Film Thickness
Creating a Scalar Scene

Creating a Maximum Report, Monitor, and Plot

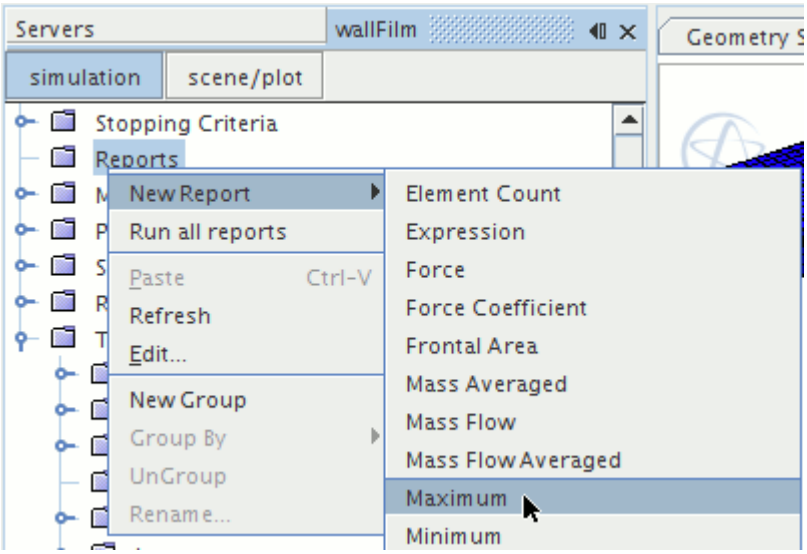
Create a report, monitor, and plot to monitor the maximum film thickness along the wetted wall.

STAR-CCM+ can dynamically monitor virtually any quantity while the solution develops. To monitor a quantity, you set up a report that defines the quantity of interest and the parts of the region to monitor. A monitor is then defined based on that report. This monitor can also be used to create an appropriate X-Y graph plot.

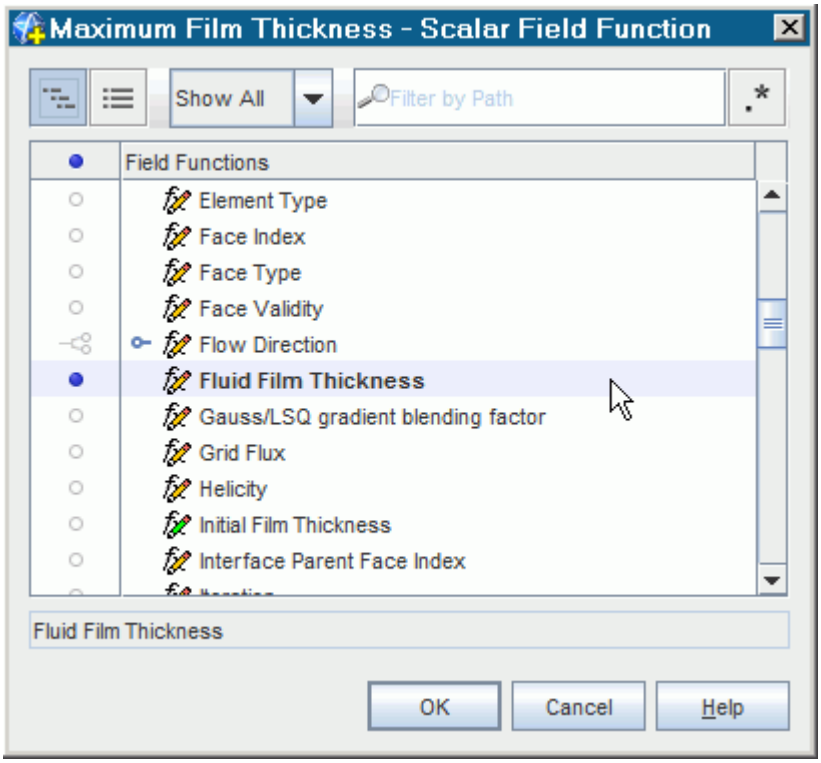
To create a maximum report, monitor, and plot:

Set up a report to monitor the maximum film thickness along the wetted wall.

1. Right-click the **Reports** node and select **New Report** > **Maximum**.

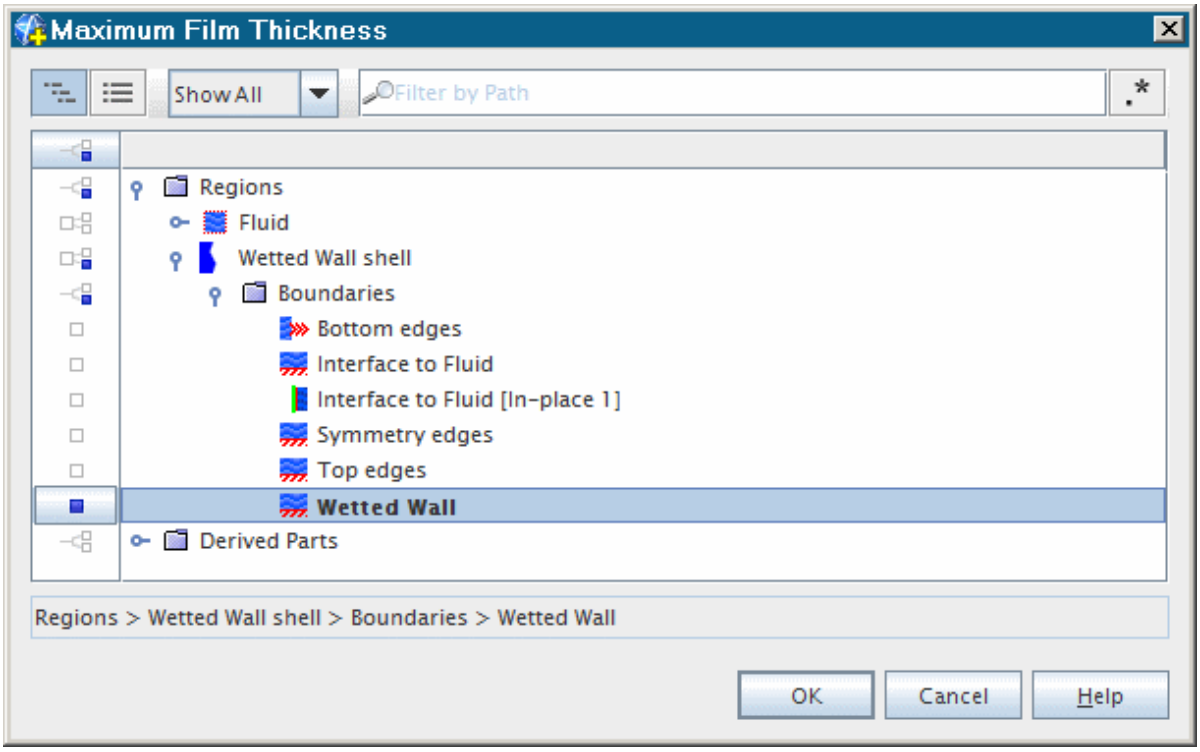


2. Rename the **Reports** > **Maximum 1** node to `Maximum Film Thickness`.
3. Select the **Maximum Film Thickness** node and do the following:
   a) Set the *Scalar Field Function* to **Fluid Film Thickness**.



b) Click to the right of the *Parts* property.

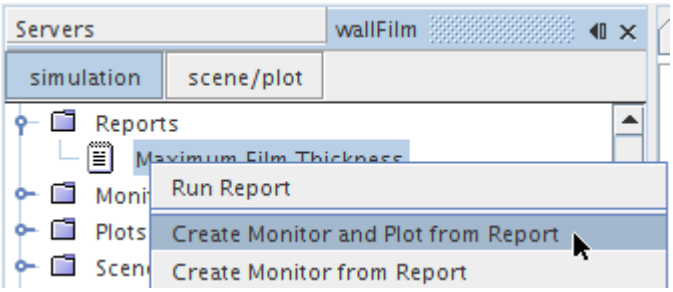
c) In the dialog that appears, expand the **Regions** > **Wetted Wall shell** node and select **Wetted Wall**.



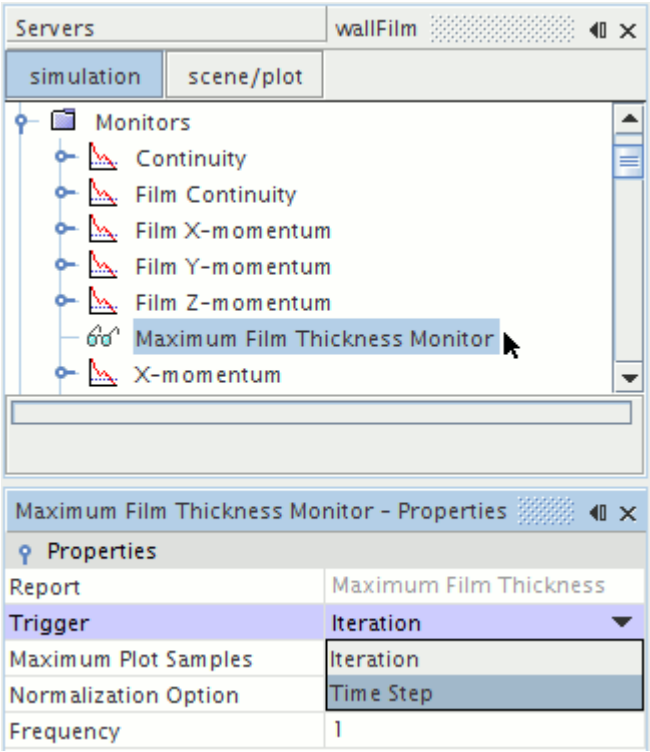
d) Click **OK**.

Create a monitor and a plot for the report.

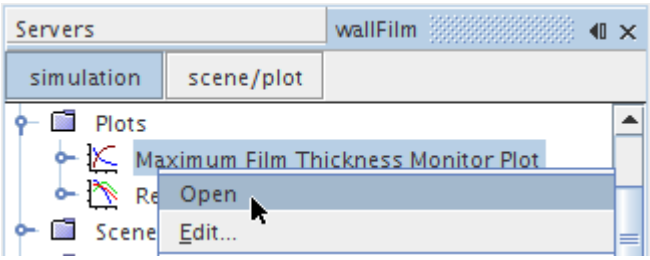
4. Right-click the **Maximum Film Thickness** node and select **Create Monitor and Plot from Report**.



5. Select the **Monitors** > **Maximum Film Thickness Monitor** node and check that the *Trigger* property is set to **Time Step**.



6. Right-click the **Plots** > **Maximum Film Thickness Monitor Plot** node and select **Open**.



Creating an X-Y Plot of Fluid Film Thickness

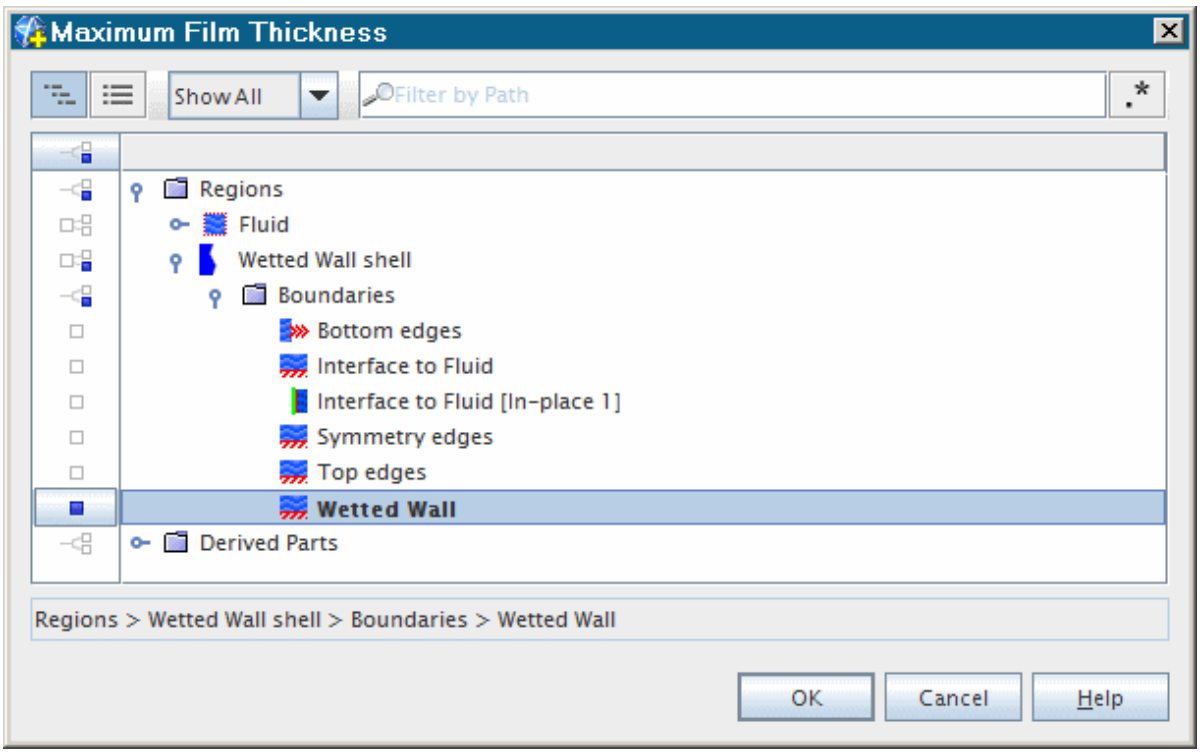
Create an X-Y plot of the fluid film thickness along a plane section through the center of the wetted wall.

To create an X-Y plot of the fluid film thickness:

Create the plane section.

1. Activate the *Mesh Scene 1* in the *Graphics* window.
2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane**.
3. In the *Create Section* window, do the following:

   a) In the *Input Parts* box, click the drop-down menu. In the dialog that appears, use (**Column Selection**) to clear selections, then expand the **Regions** > **Wetted Wall shell** node and select **Wetted Wall**.



   b) Click **OK**.

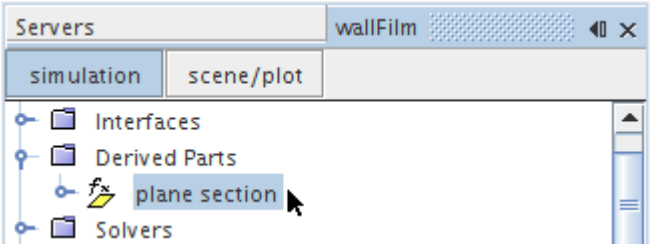
   c) Set *Normal* to `[0.0, 0.0, 1.0]` m.

   d) Under *Display*, select the **No Displayer** radio button.

   e) Click **Create** and then **Close**.

   A new derived part is created called **plane section**.

4. To view this part in the *Graphics* window, open the **Derived Parts** node and select **plane section**.

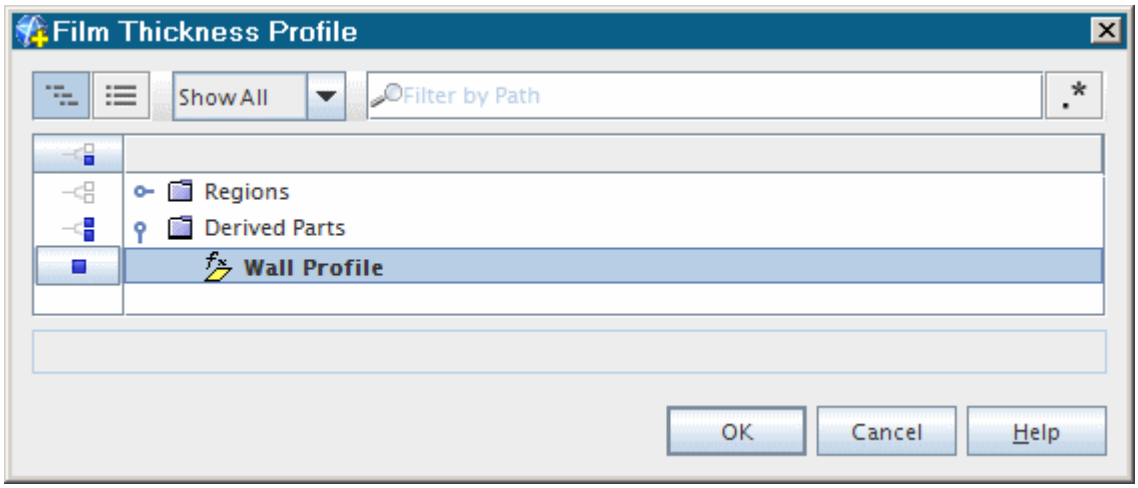


The location of the line running down the surface is highlighted in the *Graphics* window.

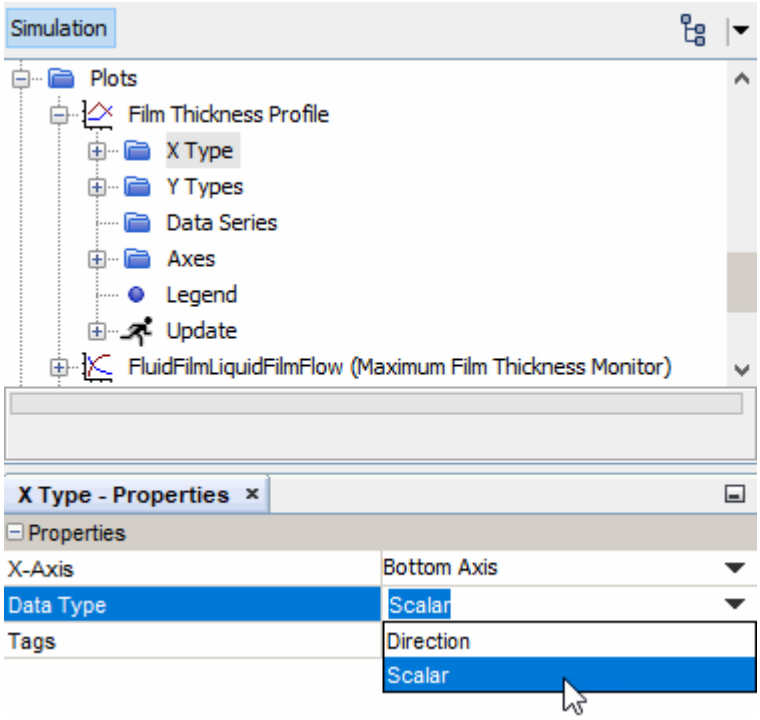
5. Rename the **Derived Parts** > **plane section** node to `Wall Profile`.

Create an X-Y plot for the fluid film thickness using the plane section.

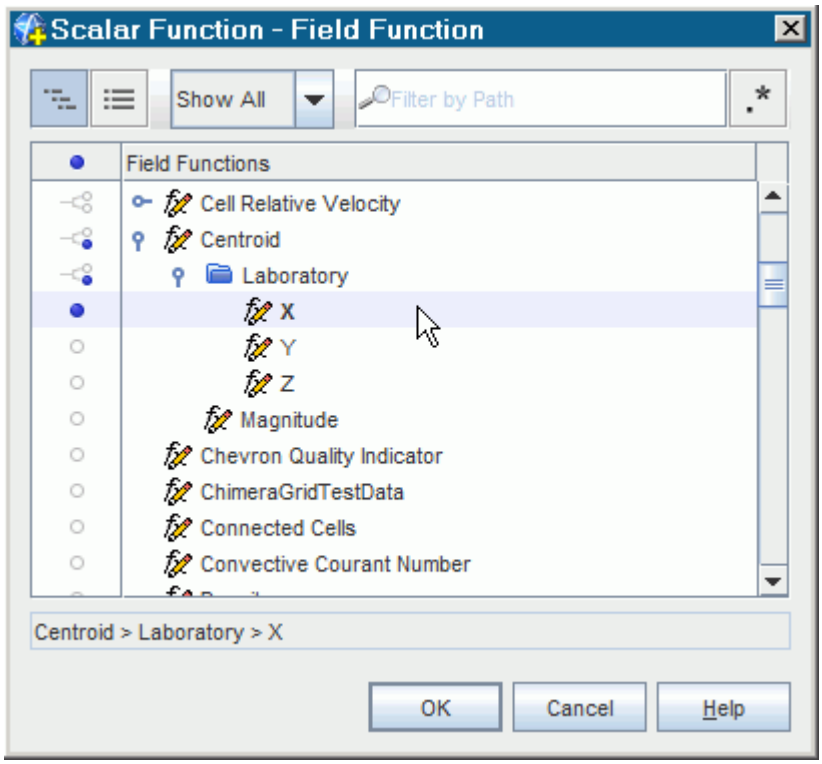
6. Right-click the **Plots** node and select **New Plot** > **X-Y**.
7. Rename the **Plots** > **XY Plot 1** node to `Film Thickness Profile`.
8. With the **Film Thickness Profile** node selected, click to the right of the *Parts* property. In the dialog that appears, expand the **Derived Parts** node and select **Wall Profile**.



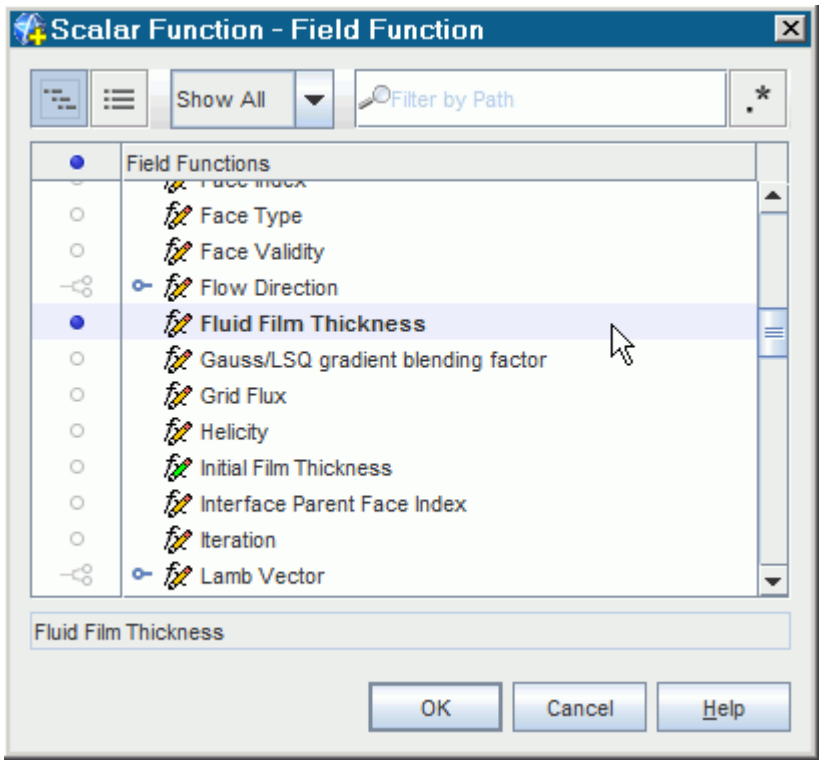
9. Click **OK**.
10. Select the **Film Thickness Profile** > **X Type** node and set the *Data Type* property it to **Scalar**.



11. Select the **Film Thickness Profile** > **X Type** > **Scalar Function** node and click to the right of the *Field Function* property and, in the *Scalar Function - Field Function* dialog, select **Centroid** > **Laboratory** > **X**.



12. Select the **Film Thickness Profile** > **Y Types** > **Y Type 1** > **Scalar Function** node and click to the right of the *Field Function* property and in the *Scalar Function - Field Function* dialog, select **Fluid Film Thickness**.



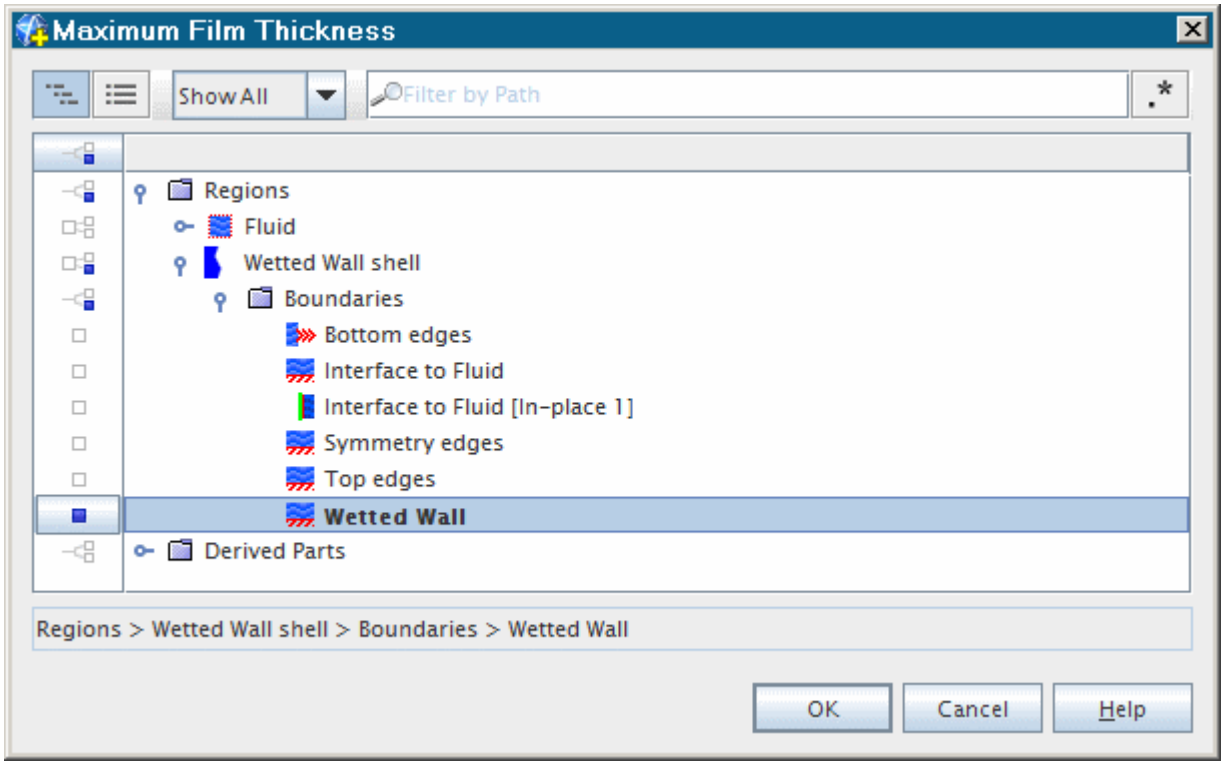
Creating a Scalar Scene

Create a scalar scene for fluid film thickness.

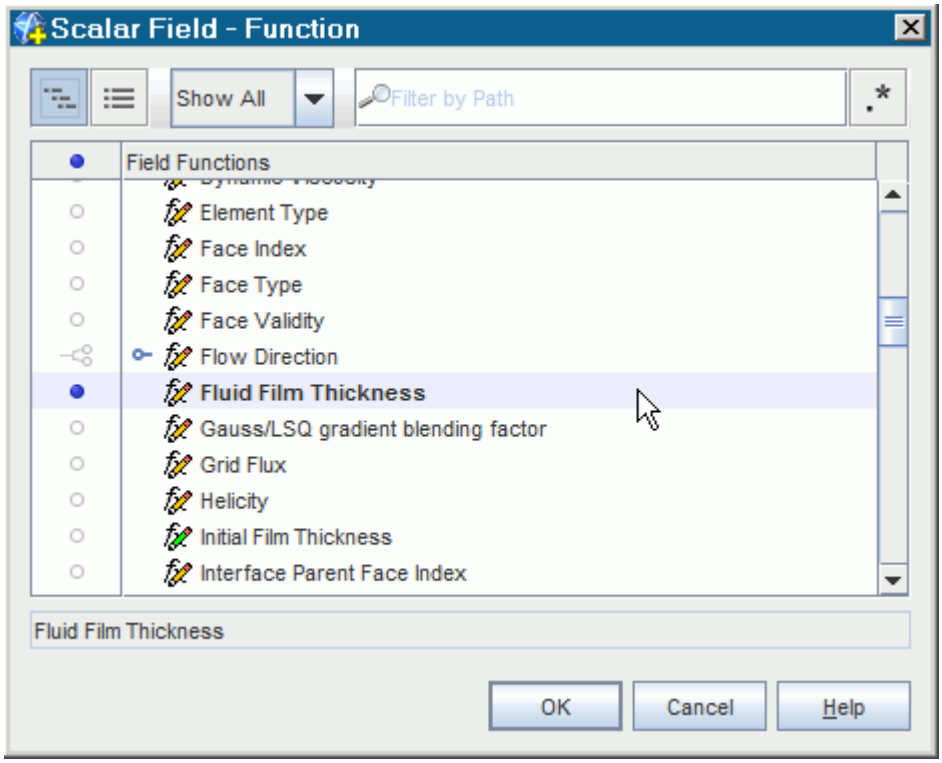
To create a scalar scene:

1. Right-click the **Scenes** node and select **New Scene** > **Scalar**.

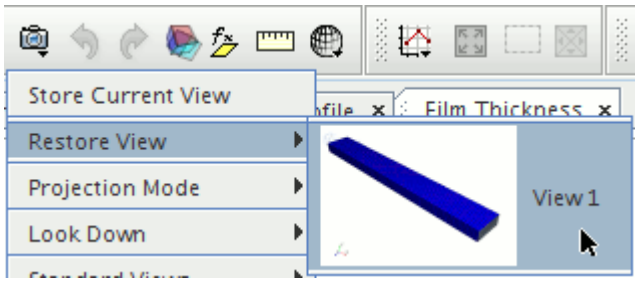
   A new node, **Scalar Scene 1**, is created under the **Scenes** node.
2. Rename the **Scenes** > **Scalar Scene 1** node to `Film Thickness`.
3. Above the object tree, click **scene/plot**.
4. Select the **Film Thickness** > **Displayers** > **Scalar 1** > **Parts** node, and click to the right of the *Parts* property.
5. In the dialog that appears, expand the **Regions** > **Wetted Wall shell** node and select **Wetted Wall**.



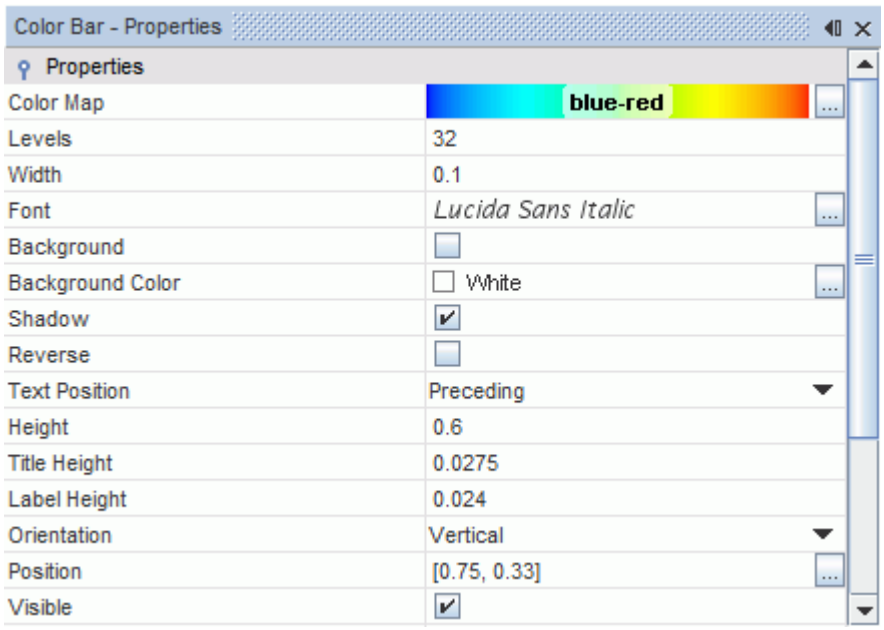
6. Click **OK**.
7. Select the **Film Thickness** > **Displayers** > **Scalar 1** > **Scalar Field** node.
8. Click to the right of the *Function* property and, in the *Scalar Field - Function* window, select **Fluid Film Thickness**.



9. Click (**Save-Restore-Select views**) in the toolbar and select **Restore View** > **View 1**.



10. Select the **Film Thickness** > **Displayers** > **Scalar 1** > **Color Bar** node and set *Width* to `0.1`, *Height* to `0.6`, *Orientation* to **Vertical** and *Position* to `[0.75, 0.33]`.



11. Save the simulation.

Running the Simulation

Preparation of the simulation is now complete, and the simulation can be run.

To run the simulation:

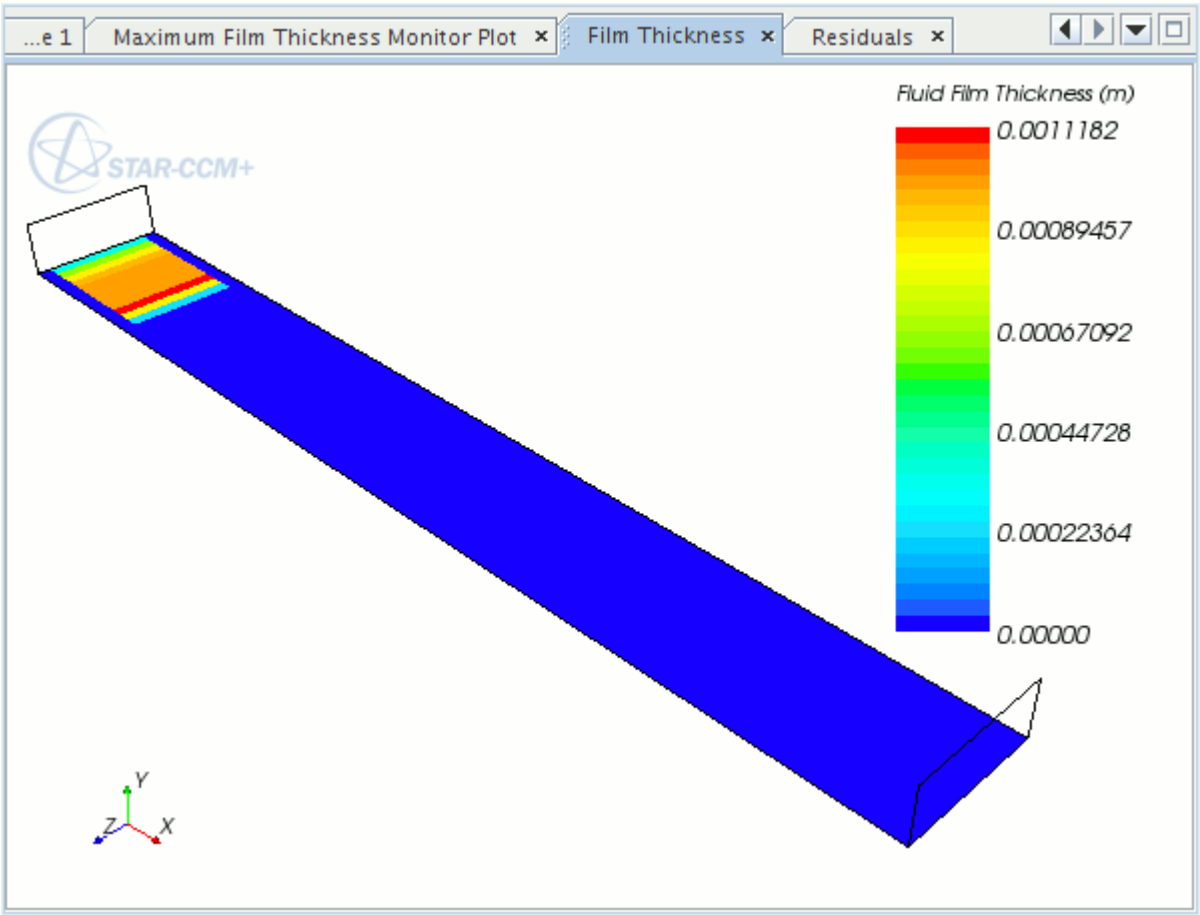
1. Click (**Run**) in the *Solution* toolbar.

   The solution progress is displayed in the *Output* window. The *Residuals* display is created automatically in the *Graphics* window and shows the progress of the solvers.

While the simulation is running, you can click the tabs at the top of the *Graphics* window to view scenes and plots.

During the run, you can stop the process by clicking (**Stop**) on the toolbar. If you do stop the simulation, you can click (**Run**) to resume it.

2. Activate the film thickness scene and observe the formation of fluid film as the solution progresses.



The example above shows the fluid film thickness at early stages of the run. The simulation continues until 10 seconds of physical time is complete.

When the last iteration is complete, the following message is displayed in the *Output* window:

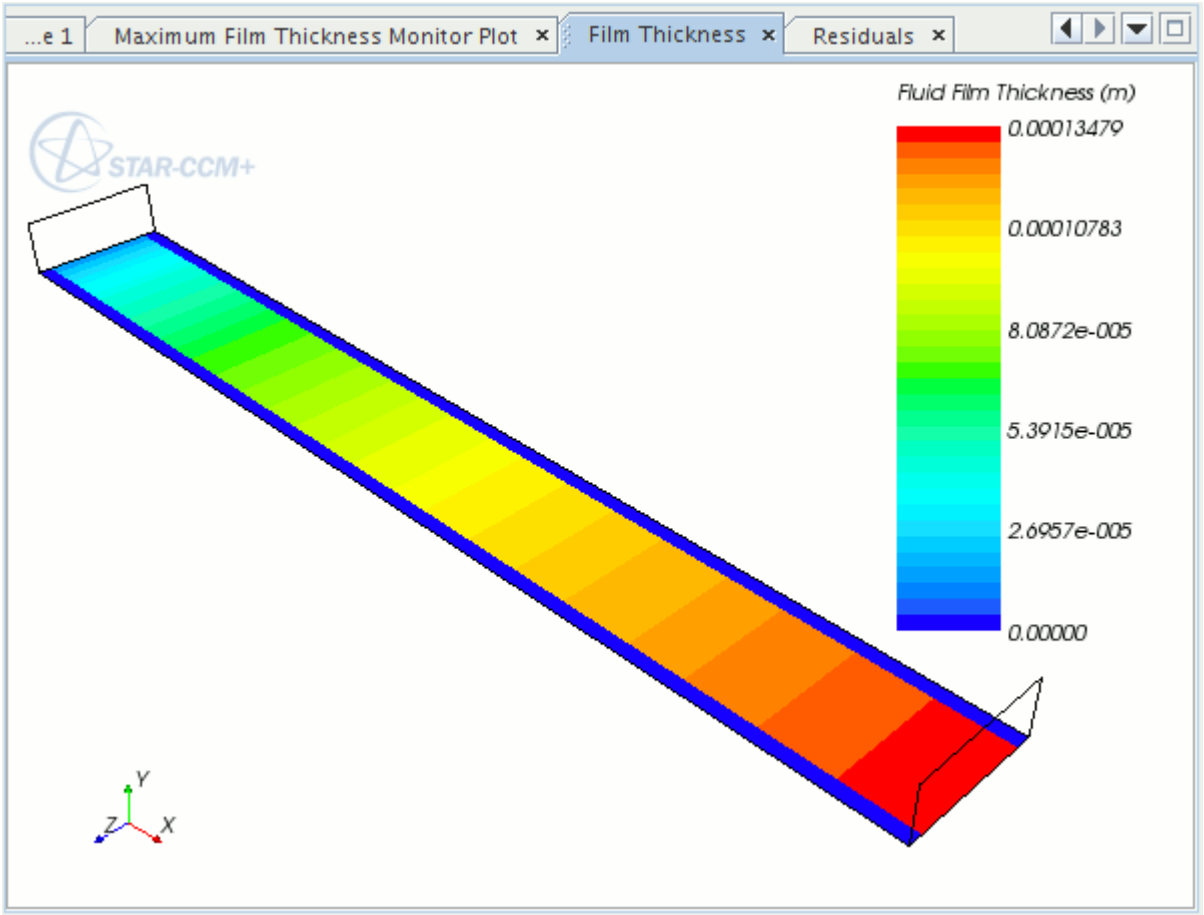
```
Stopping criterion Maximum Physical Time satisfied.
```

3. When the simulation has finished running, save it.

Visualizing the Results

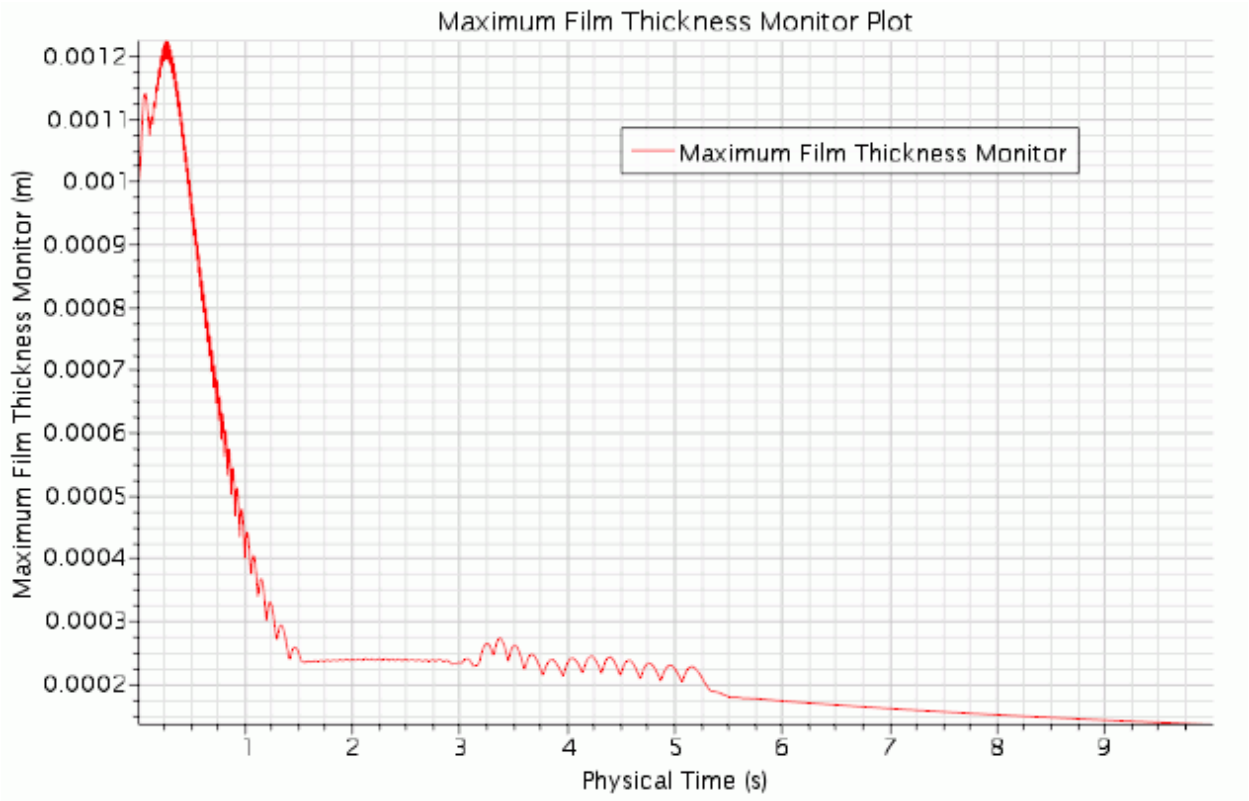
Examine the results of the simulation.

The film thickness contour for the converged solution appears as shown below.



1. Activate the maximum film thickness monitor plot.

   The maximum film thickness plot for the converged solution appears as shown below.



Summary

This tutorial demonstrated how to use the fluid film modeling capability of STAR-CCM+.

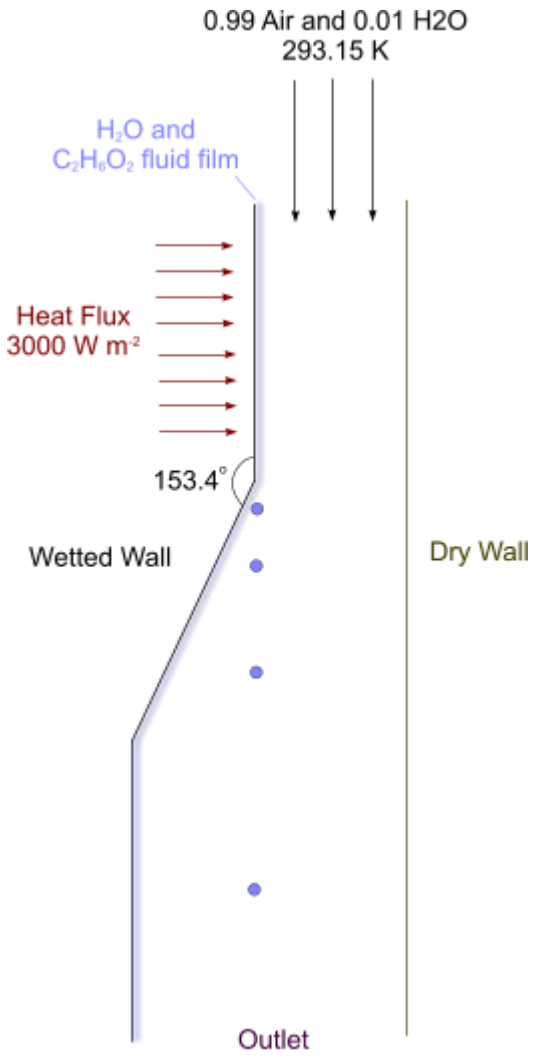
This tutorial has introduced the following STAR-CCM+ features:

- Defining models for fluid film model.
- Setting fluid film models and initial conditions using field function.
- Setting up shell region and boundary conditions.
- Setting up variable time-step size for the implicit solver.
- Running the solver for a set physical time.

Fluid Film: Binary Liquid Film Flow with Evaporation and Edge Stripping

This tutorial demonstrates how to use the fluid film modeling capability of STAR-CCM+ to set up a multi-component film scenario with evaporation and edge stripping. It simulates the progress of a film mixture of water and ethylene glycol flowing down a partially heated wall.

The liquid mixture flows down the wetted wall due to gravity, forming a film. The film evaporates and edge stripping occurs at the 153.4° bend.



The initial liquid film is at 293.15 K and covers the whole wetted wall with a 0.343 mm thickness. The surrounding multi-component gas is initially at rest and is composed of 99% dry air and 1% water vapor at 293.15 K. The case is run using the implicit unsteady solver for a simulated time of 5.0 seconds. The time step size is set to 0.01 seconds.

Contents:

Prerequisites
Importing the Mesh and Naming the Simulation
Visualizing the Volume Mesh
Selecting the Physics Models
Setting the Multi-Component Gas Properties and Initial Conditions
Setting Fluid Film Models and Initial Conditions
Setting the Fluid Film Evaporation and Condensation Model
Setting the Lagrangian Model
Setting the Gas Boundary Conditions
Setting Up Shell Region and Boundary Conditions
Setting Up the Injector
Setting Up the Film Stripping Model
Setting Reference Values
Setting Solver Parameters and Stopping Criteria
Setting Up Evaporation Rate Reports, Monitors, and Plots
Setting Up a Scalar Scene
Running the Simulation
Visualizing the Results
Summary

Prerequisites

The instructions in the Fluid Film: Binary Liquid Film Flow with Evaporation and Edge Stripping tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Importing the Mesh and Naming the Simulation

Launch STAR-CCM+, import the supplied geometry, and save the simulation.

To import the mesh and name the simulation:

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh**.
4. In the *Open* dialog, choose `doc/startutorialsdata/multiphaseFlow/data/stripping.ccm`, then click **Open**.

STAR-CCM+ provides feedback on the import process, which takes a few seconds, in the *Output* window. A geometry scene is created in the *Graphics* window.

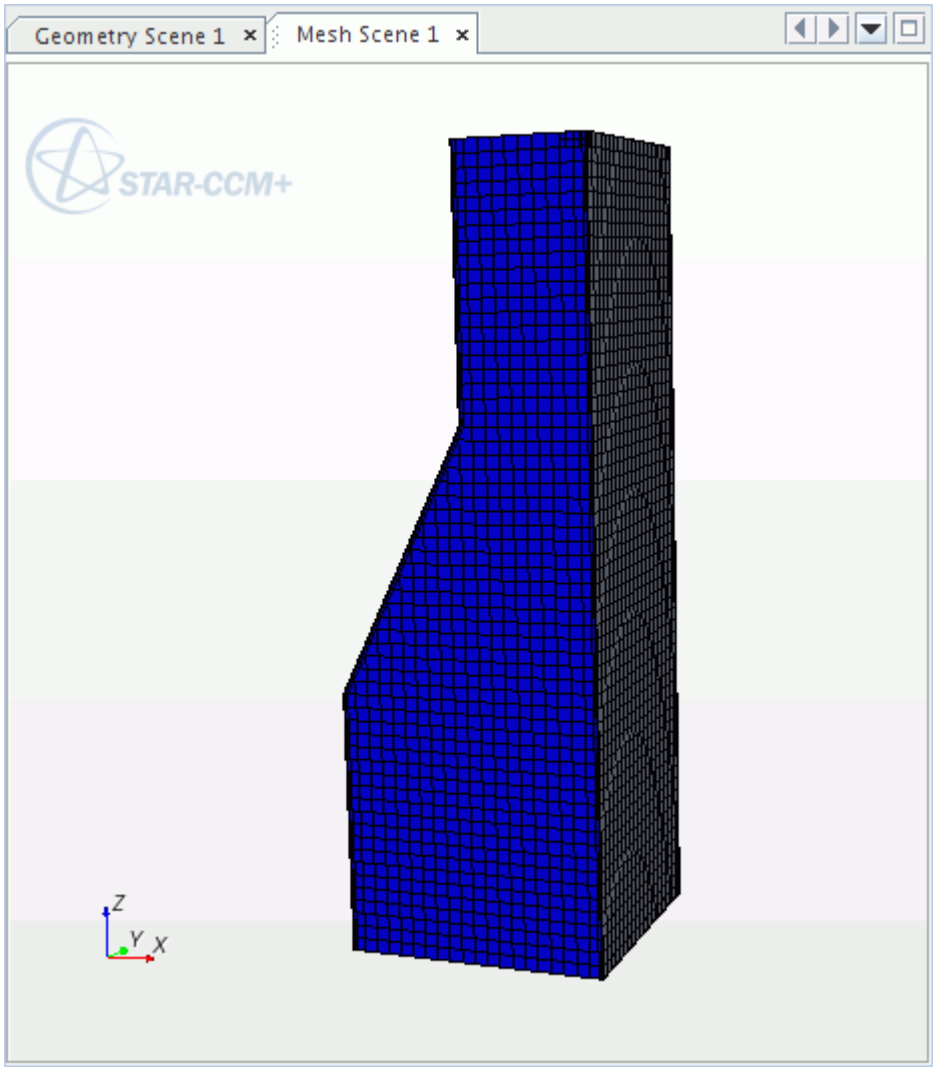
5. Save the simulation as `strippingFilm.sim`.

Visualizing the Volume Mesh

Examine the imported volume mesh.

To visualize the volume mesh:

1. Create a mesh scene.
2. Position the part to obtain a similar view to the one shown below.



3. Click (**Save-Restore-Select views**) in the toolbar and select **Store Current View**.

Selecting the Physics Models

Physics models define the primary variables of the simulation, including pressure, temperature, velocity, and what mathematical formulation is used to generate the solution.

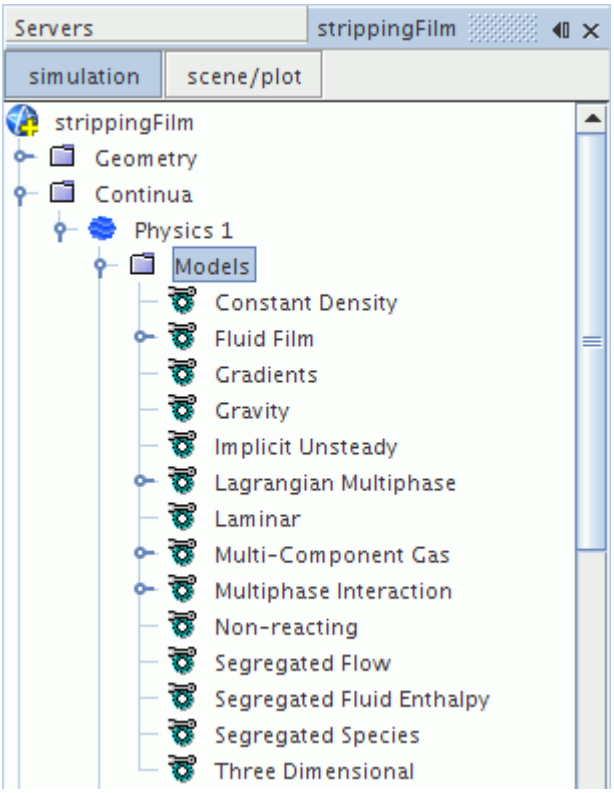
The main phenomena to solve for in this simulation is laminar, incompressible, multi-component gaseous flow. You use the segregated flow and fluid enthalpy models, and additionally select the multi-component fluid film, multiphase interaction, and Lagrangian multiphase models.

To select the physics models:

1. For the physics continuum, **Continua** > **Physics 1**, select the following models in order:

Group Box	Model
Enabled Models	**Three Dimensional** (Pre-selected)
Time	**Implicit Unsteady**
Material	**Multi-Component Gas**
Reaction Regime	**Non-reacting**
Flow	**Segregated Flow** **Segregated Species** (selected automatically) **Gradients** (selected automatically)
Equation of State	**Constant Density**
Viscous Regime	**Laminar**
Optional Models	**Fluid Film** **Gravity** **Segregated Fluid Enthalpy** **Multiphase Interaction** **Lagrangian Multiphase**

2. Click **Close**.
3. To review the models, open the **Physics 1** > **Models** node.



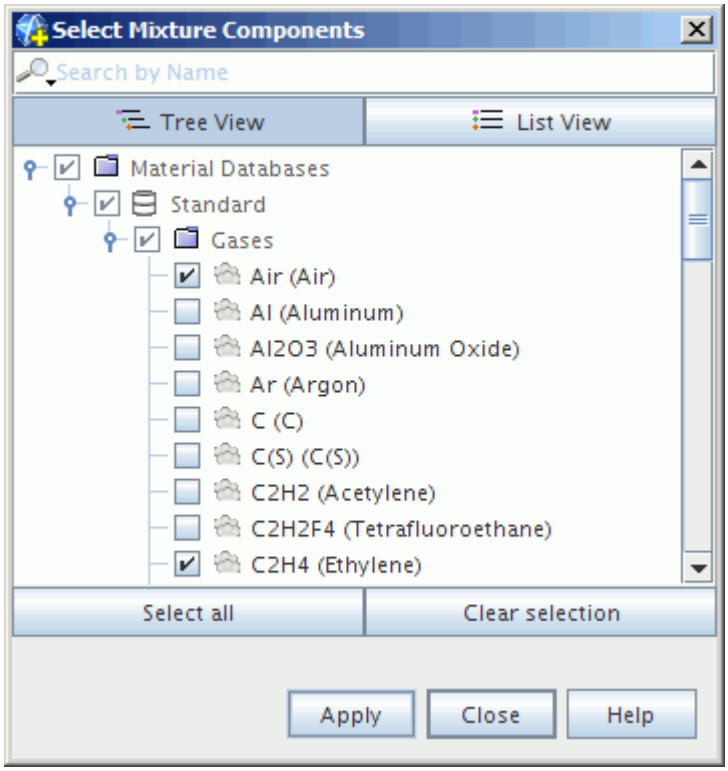
4. Save the simulation.

Setting the Multi-Component Gas Properties and Initial Conditions

Define the gas components in the gas continuum and set their properties.

To set the multi-component gas properties and initial conditions:

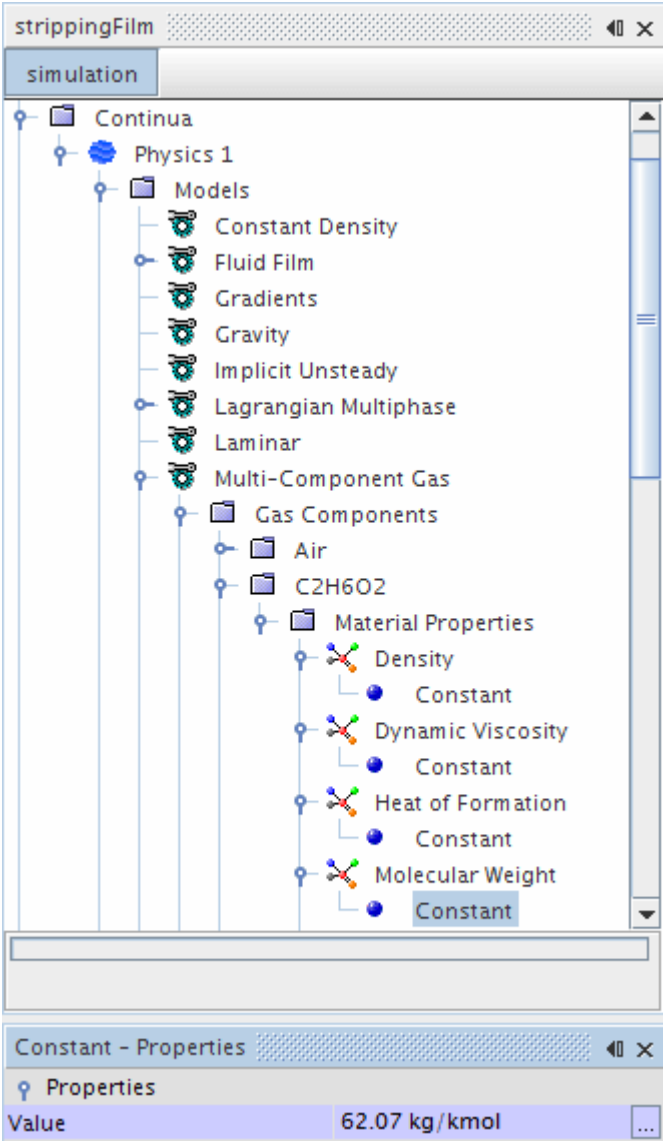
1. Right-click **Multi-Component Gas** > **Gas Components** and choose **Select Mixture Components**.
2. In the *Select Mixture Components* dialog, expand the **Material Databases** > **Standard** > **Gases** node and select the following gases: **H2O (Water)**, **C2H4 (Ethylene)**, and **Air (Air)**.



3. Click **Apply**, and then **Close**.

Specify custom properties to model ethylene glycol vapor.

4. Rename the **Gas Components** > **C2H4** node to **C2H6O2**, the chemical symbol for ethylene glycol.
5. Select the **Gas Components** > **C2H6O2** > **Material Properties** > **Molecular Weight** > **Constant** node and change its *Value* to `62.07 kg/kmol`.

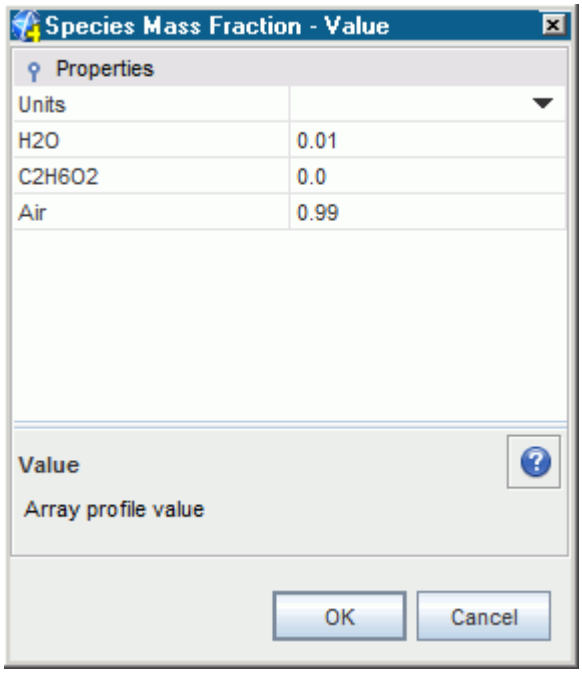


Set the initial conditions for the multi-component gas continuum. The gas is initially at 293.15 K.

6. Select **Continua** > **Physics 1** > **Initial Conditions** > **Static Temperature** and set *Value* to `293.15 K`.

The gas initially consists of 99% dry air and 1% water vapor.

7. Select **Continua** > **Physics 1** > **Initial Conditions** > **Species Mass Fraction** and click ... (**Custom Editor**) for the *Value* property.
8. In the *Species Mass Fraction - Value* dialog, set the values of *H2O* and *Air* to `0.01` and `0.99` respectively.



The sum of all mass fractions must be equal to `1.0`.

9. Click **OK**.

Setting Fluid Film Models and Initial Conditions

Set the fluid film models and the initial conditions for the fluid film phases.

You define the fluid film phases for H2O and C2H6O2 and set the initial conditions for the fluid film. The film is initially at 293.15 K and has a constant thickness of 0.343 mm. The mass fractions of water and ethylene glycol are initially equal.

To set the fluid film models and initial conditions:

Define the fluid film phases for H2O and C2H6O2.

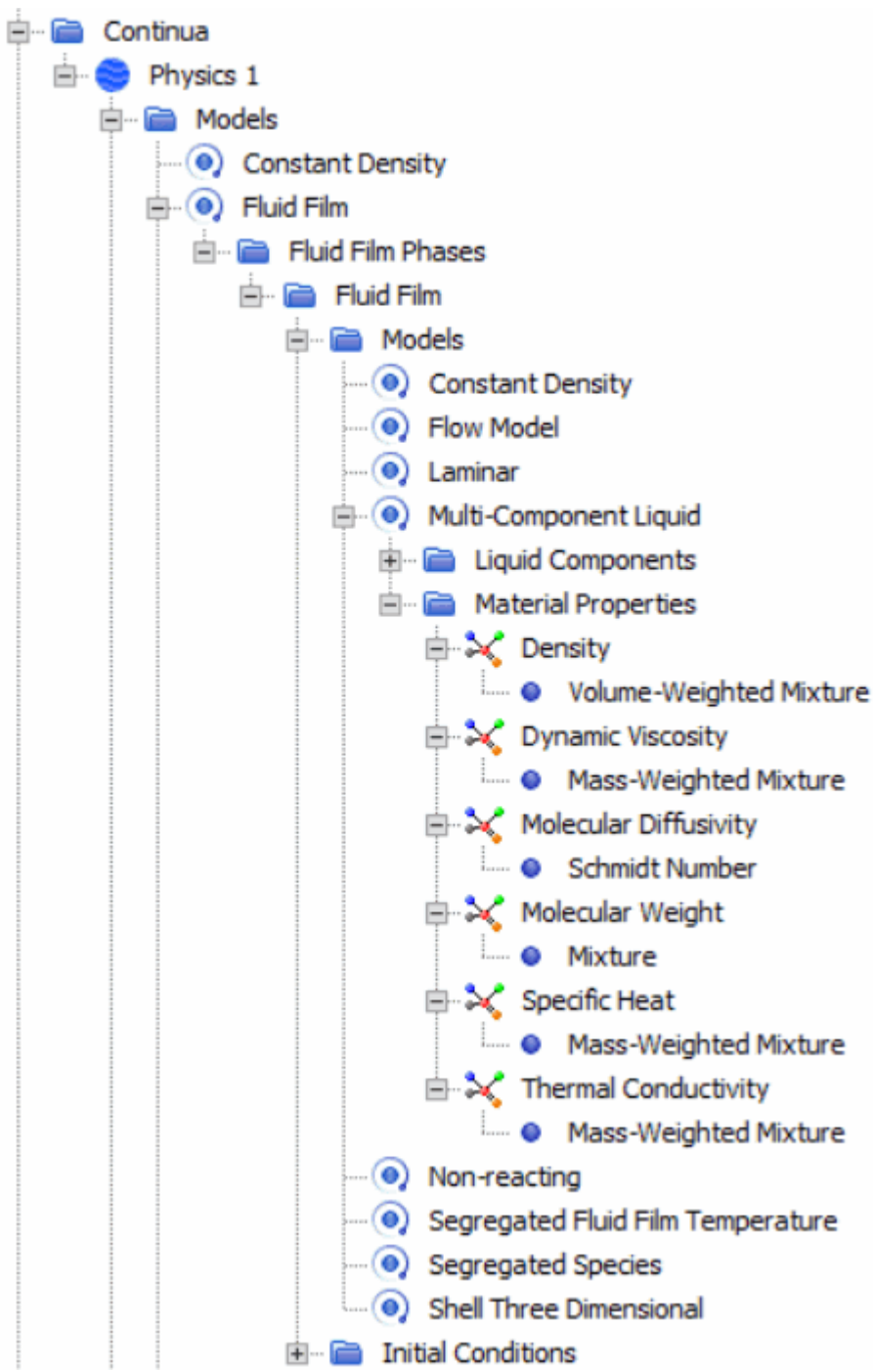
1. In the **Physics 1** continuum, right-click the **Models** > **Fluid Film** > **Fluid Film Phases** node and create a new phase.
2. Rename the **Phase 1** node to **Fluid Film**.
3. For the **Fluid Film** phase, select the following models in order:

Group Box	Model
Space	**Shell Three-Dimensional** **Flow Model** (selected automatically) **Laminar** (selected automatically) **Segregated Fluid Film Temperature** (selected automatically)
Material	**Multi-Component Liquid**

Group Box	Model
	Segregated Species (selected automatically)
Reaction Regime	**Non-reacting**
Equation of State	**Constant Density**

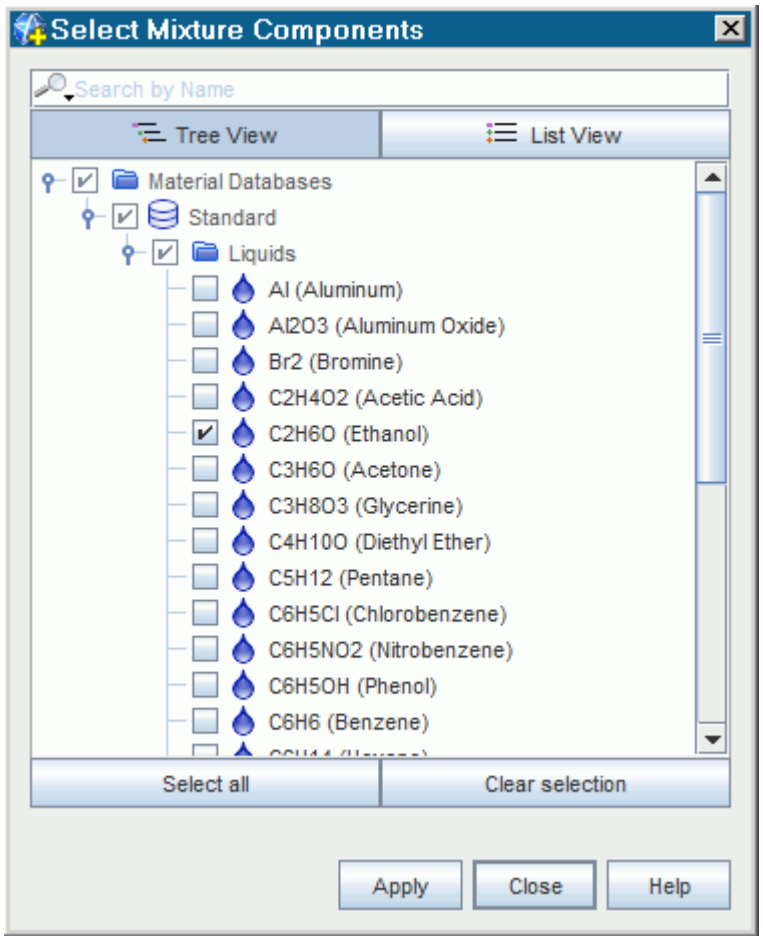
4. Click **Close**.

   The simulation tree appears as follows:



Select the liquid film components in the same way as for the multi-component gas continuum. Ethylene glycol is not listed in the liquid database; select ethanol and modify its name and properties to match the name and properties of ethylene glycol.

5. Right-click **Fluid Film** > **Models** > **Multi-Component Liquid** > **Liquid Components** and choose **Select Mixture Components**.
6. Expand the **Material Databases** > **Standard** > **Liquids** node and select **H2O (Water)** and **C2H6O (Ethanol)**.

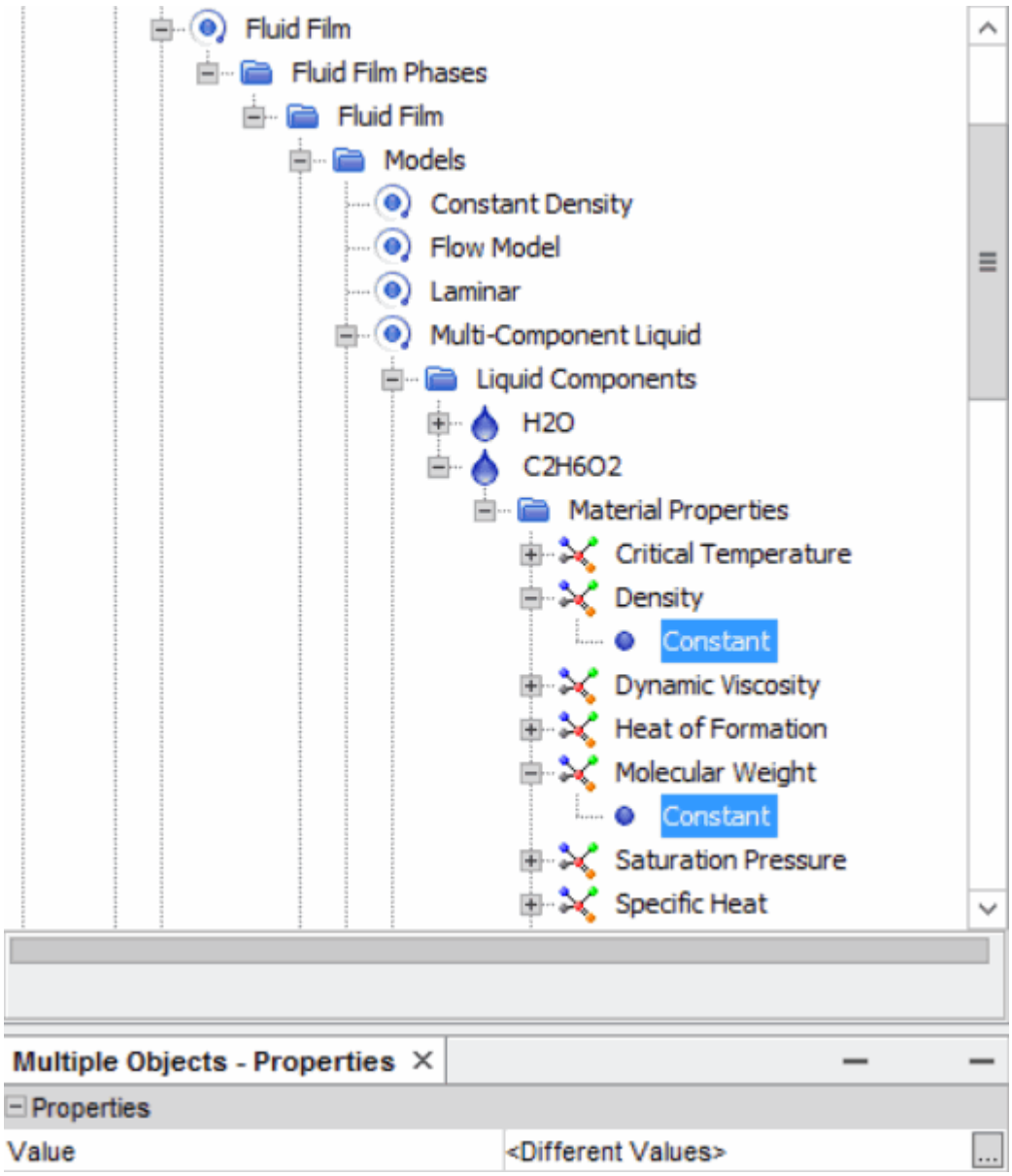


7. Click **Apply**, and then **Close**.
8. Rename the **Liquid Components** > **C2H6O** node to **C2H6O2**.

Set the density and molecular weight values of ethylene glycol.

9. Edit the **Liquid Components** > **C2H6O2** > **Material Properties** node and set the following properties:

Node	Property	Setting
Density > **Constant**	**Value**	`1113.2 kg/m^3`
Molecular Weight > **Constant**	**Value**	`62.07 kg/kmol`



Set the initial conditions for the fluid film.

10. Edit the **Fluid Film** > **Initial Conditions** node and set the following properties:

Node	Property	Setting
Fluid Film Thickness	**Value**	`3.43E-4 m`
Static Temperature	**Value**	`293.15 K`
Species Mass Fraction	**Value**	`[0.5, 0.5]`

11. Save the simulation.

Setting the Fluid Film Evaporation and Condensation Model

Set up the fluid film evaporation and condensation model.

The fluid film evaporation and condensation model is set as a Film-Physics Continuum phase interaction in STAR-CCM+.

To create this phase interaction:

1. Right-click **Continua** > **Physics 1** > **Models** > **Multiphase Interaction** > **Phase Interactions** and select **New**.

   A new **Phase Interaction 1** node is added to the **Phase Interactions** folder.
2. Rename the **Multiphase Interaction** > **Phase Interactions** > **Phase Interaction 1** node to **Film Mixture - Gas**.

To set up the Film-Physics Continuum phase interaction:

3. Right-click **Multiphase Interaction** > **Phase Interactions** > **Film Mixture - Gas** > **Models** and choose **Select Models...**.

In the *Film Mixture - Gas Model Selection* dialog:

4. Select **Film-Physics Continuum Interaction** from the *Phase Interaction Topology* box.
5. Click **Close**.
6. Select **Film Mixture - Gas** > **Models** > **Film-Physics Continuum Interaction** and set the *Fluid Film Phase* to **Fluid Film**.

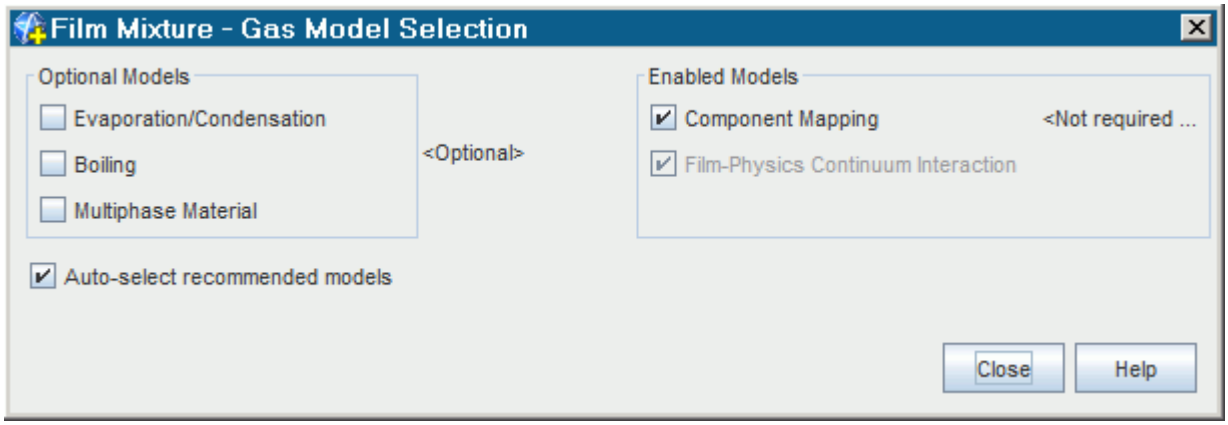
   The **Component Mapping** model is activated automatically.

To select the evaporation and condensation model:

7. Right-click **Multiphase Interaction** > **Phase Interactions** > **Film Mixture - Gas** > **Models** and choose **Select Models...**.

In the *Film Mixture - Gas Model Selection* dialog:

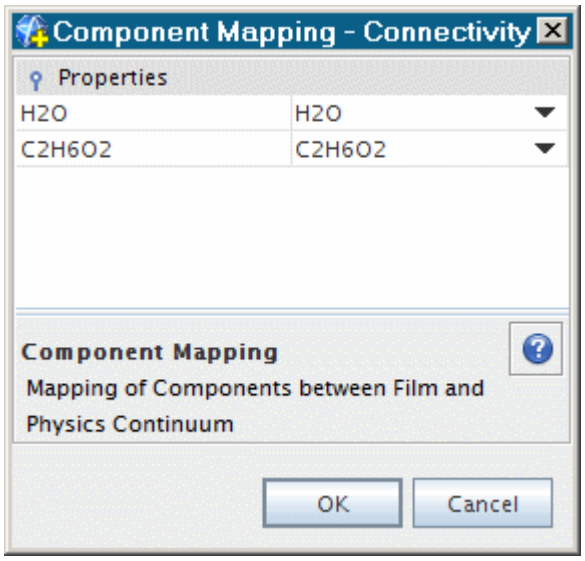
8. Select **Evaporation/Condensation** and **Multiphase Material** from the *Optional Models* box.



9. Click **Close**.

Specify the connection between components in the gas phase and components in the fluid film phase for evaporation or condensation:

10. Select **Multiphase Interaction** > **Phase Interactions** > **Film Mixture - Gas** > **Models** > **Component Mapping** and click ... (**Custom Editor**) for the *Connectivity* property.
11. In the *Connectivity* dialog, map the fluid film components to their corresponding gas components, as shown below.



12. Click **OK**.

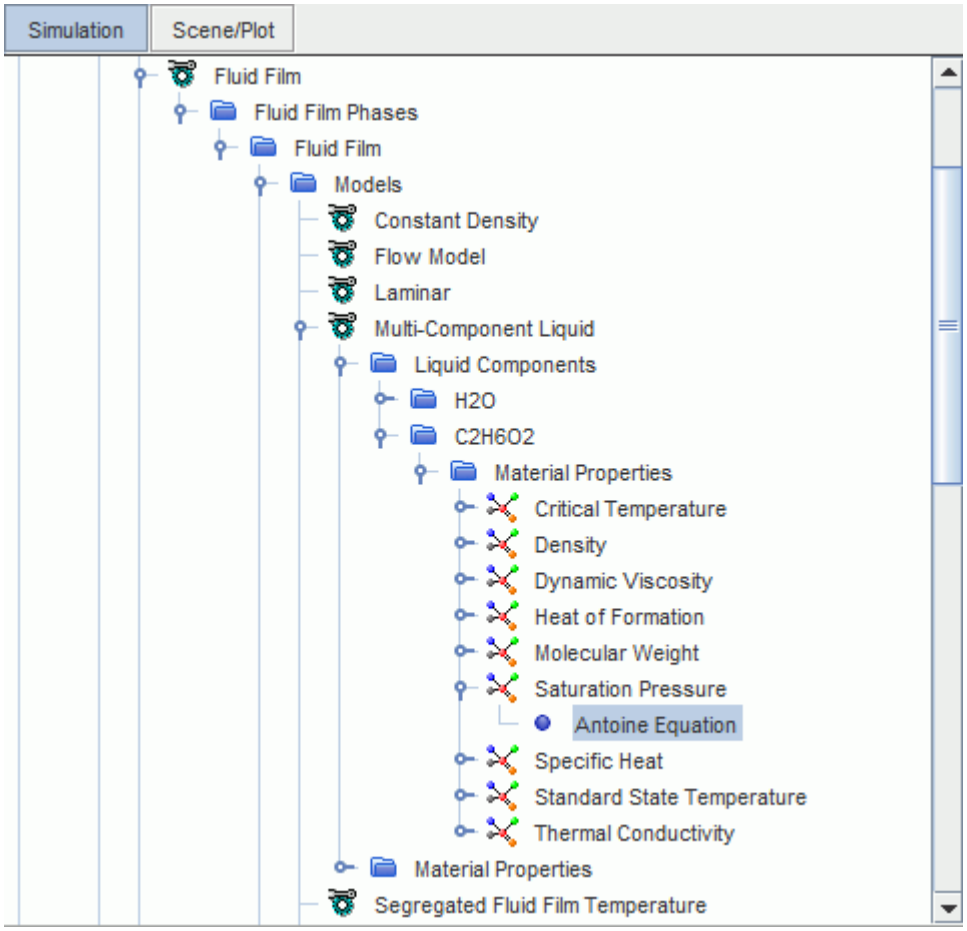
To model realistic saturation pressure behavior, use the **Antoine Equation** method.

13. Select **Continua** > **Physics 1** > **Models** > **Fluid Film** > **Fluid Film Phases** > **Fluid Film** > **Models** > **Multi-Component Liquid** > **Liquid Components** > **H2O** > **Material Properties** > **Saturation Pressure** and set *Method* to **Antoine Equation**.

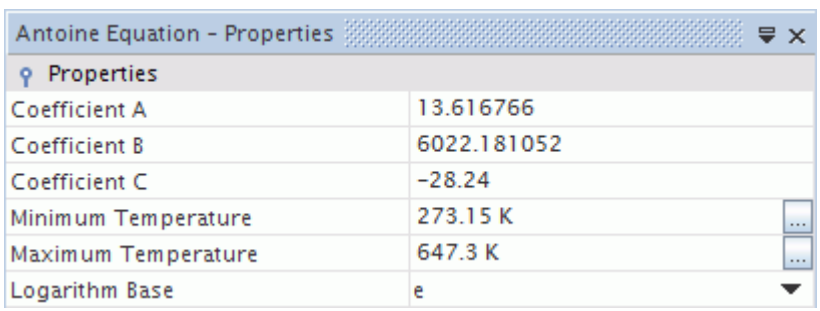
    The default values for H2O are appropriate for this simulation, so no changes are required.

To input the parameters of the *Antoine Equation* for C2H6O2:

14. Continuing with the **Liquid Components** node, select **C2H6O2** > **Material Properties** > **Saturation Pressure** and set *Method* to **Antoine Equation**.



15. Select the **Antoine Equation** node and set the properties to the values shown below:



Setting the Lagrangian Model

Set up a Lagrangian multiphase model to simulate stripping droplets.

To set the Lagrangian model:

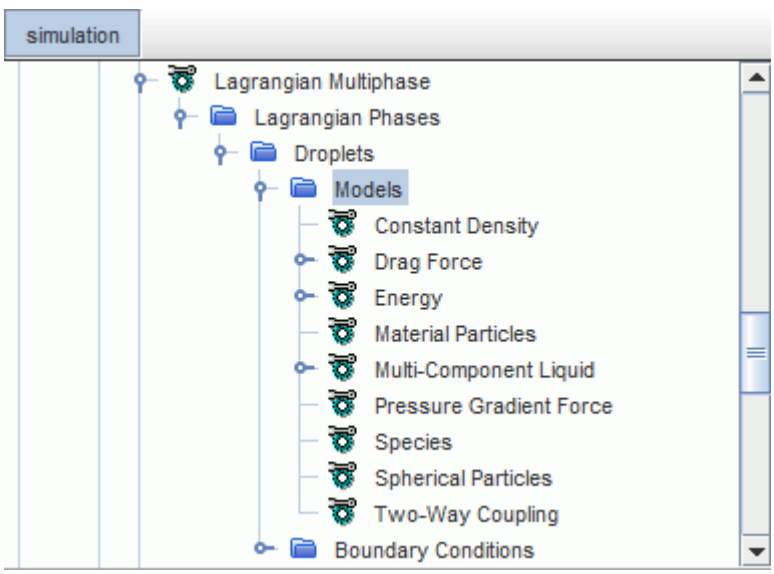
1. Right-click **Continua** > **Physics 1** > **Models** > **Lagrangian Multiphase** > **Lagrangian Phases** and select **New**.
2. Rename the newly created **Phase 1** to **Droplets**.

3. For the Lagrangian phase, **Droplets**, select the following models in order:

Group Box	Model
Particle Type	**Material Particles** **Spherical Particles** (selected automatically) **Pressure Gradient Force** (selected automatically)
Material	**Multi-Component Liquid** **Species** (selected automatically)
Equation of State	**Constant Density**
Optional Particle Forces	**Drag Force**
Optional Models	**Two-Way Coupling** **Energy**

4. Click **Close**.

   The simulation tree appears as shown below.



To set up the droplet components:

5. Right-click **Multi-Component Liquid** > **Droplet Components** and choose **Select Mixture Components**.
6. In the *Select Mixture Components* dialog, expand the **Material Databases** > **Standard** > **Liquids** node and select **H2O (Water)** and **C2H6O (Ethanol)**.
7. Click **Apply**, and then **Close**.
8. Rename the **Droplets Components** > **C2H6O** node to **C2H6O2**.

To set the density and molecular weight values for ethylene glycol:

9. Select **Droplet Components** > **C2H6O2** > **Material Properties** > **Density** > **Constant** and set *Value* to `1113.2 kg/m^3`.

10. Select **Droplet Components** > **C2H6O2** > **Material Properties** > **Molecular Weight** > **Constant** and set *Value* to `62.07 kg/kmol`.

Setting the Gas Boundary Conditions

Define the boundary conditions and specify the appropriate property values.

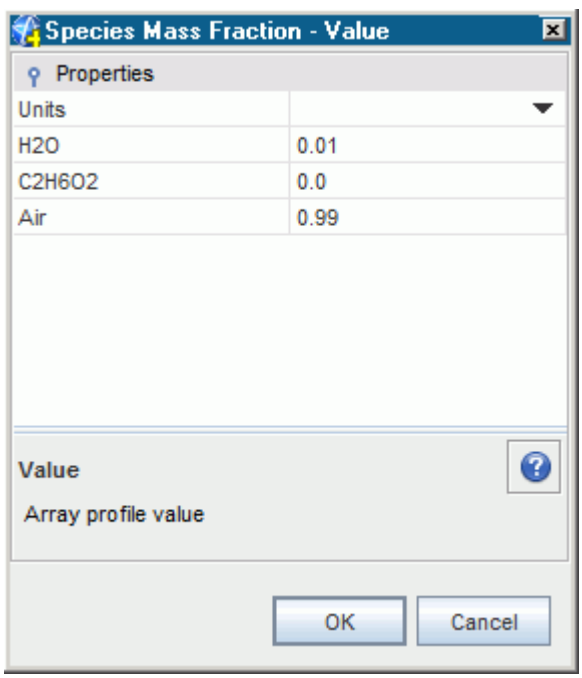
The gas enters the domain at 1.0 m/s with a temperature of 293.15 K, and a composition of 99% dry air and 1% H2O.

To set the gas boundary conditions:

Set up the inlet.

1. Edit the **Regions** > **Fluid** > **Boundaries** > **Top** > **Physics Values** node and set the following properties:

Node	Property	Setting
Species Mass Fraction	**Value**	`[0.01, 0.0, 0.99]`
Static Temperature	**Value**	`293.15 K`



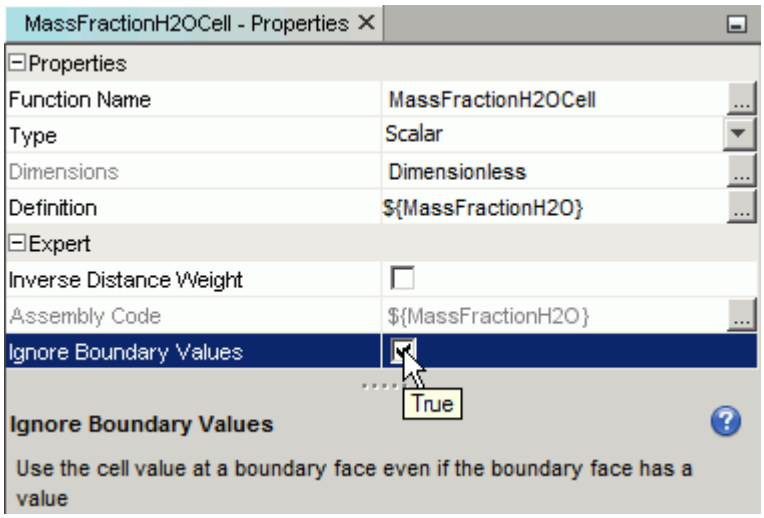
Leave the default velocity magnitude to 1.0 m/s.

Set up an outlet for the gas to leave at the bottom of the domain. The outlet gas composition is unknown and cannot be specified. The outlet gas composition is therefore matched to the composition inside the domain. Create user field functions to specify the mass fraction of each species. Set the outlet temperature to 293.15 K.

2. Right-click the **Tools** > **Field Functions** node and select **New** > **Scalar** from the pop-up menu.

3. Rename the **User Field Function 1** to `MassFractionH2OCell`.
4. Select the **MassFractionH2OCell** node and set the following properties:

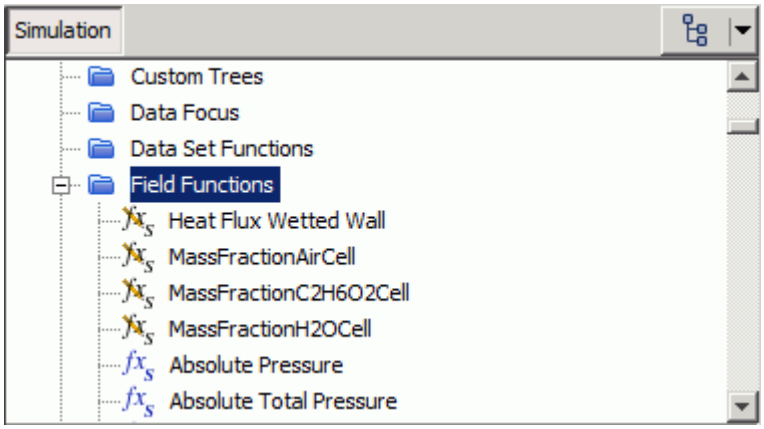
Property	Setting
Function Name	`MassFractionH2OCell`
Definition	`${MassFractionH2O}`
Ignore Boundary Values	**Activated**



5. In the same way, create field functions for the mass fraction of C2H6O2 and the mass fraction of air.

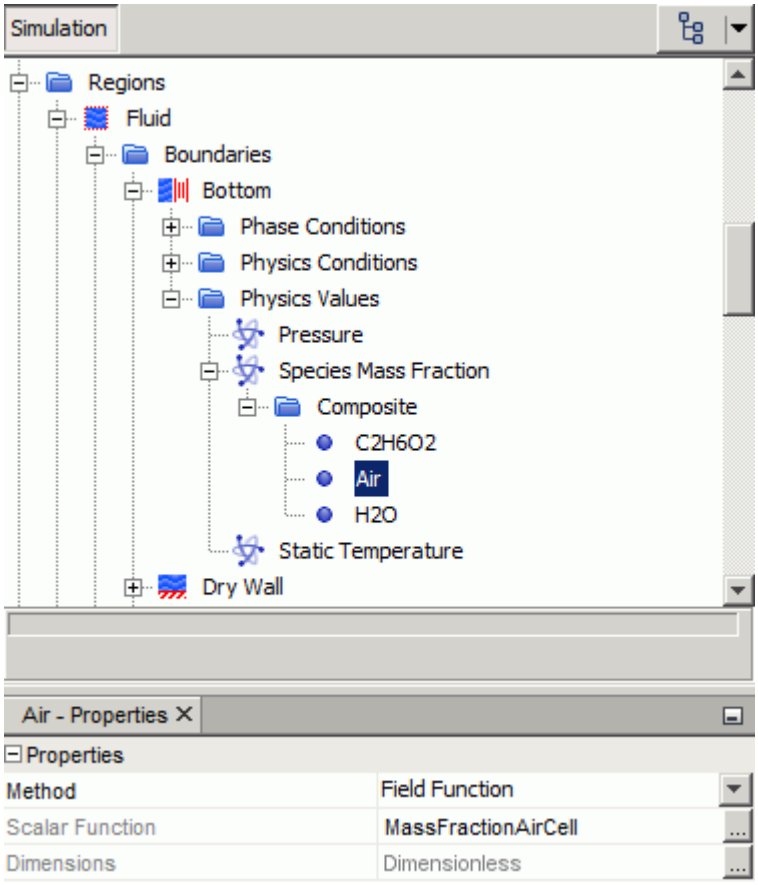
Property	Setting
Function Name	`MassFractionC2H6O2Cell`
Definition	`${MassFractionC2H6O2}`
Ignore Boundary Values	**Activated**

Property	Setting
Function Name	`MassFractionAirCell`
Definition	`${MassFractionAir}`
Ignore Boundary Values	**Activated**



6. Edit the **Regions** > **Fluid** > **Boundaries** > **Bottom** node and set the following properties:

Node	Property	Setting
Bottom	**Type**	**Pressure Outlet**
Physics Values > **Species Mass Fraction**	**Method**	**Composite**
└ **Composite** > **H2O**	**Method**	**Field Function**
	Scalar Function	**MassFractionH2OCell**
└ **Composite** > **C2H6O2**	**Method**	**Field Function**
	Scalar Function	**MassFractionC2H6O2Cell**
└ **Composite** > **Air**	**Method**	**Field Function**
	Scalar Function	**MassFractionAirCell**
Physics Values > **Static Temperature**	**Value**	`293.15 K`



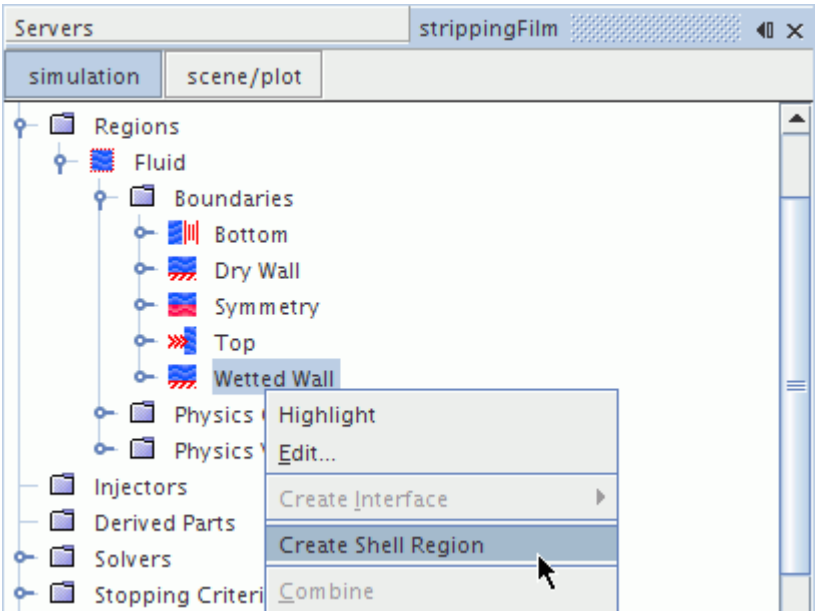
Setting Up Shell Region and Boundary Conditions

Create a shell region and define the boundary conditions.

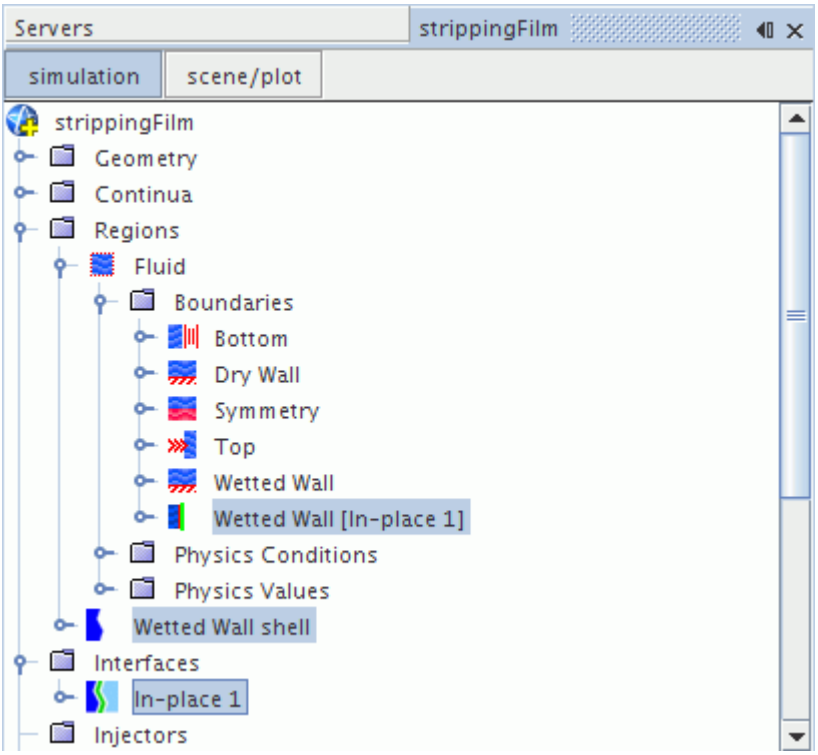
In this tutorial, the binary liquid film forms on the **Wetted Wall** boundary of the **Fluid** region. A shell region is required for all boundaries on which the fluid film is to flow. You create this shell region from the **Wetted Wall** boundary.

To create a shell region and define the boundary conditions:

1. Right-click the **Regions** > **Fluid** > **Boundaries** > **Wetted Wall** node and select **Create Shell Region**.

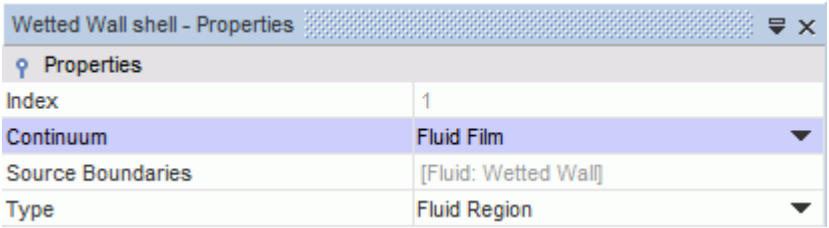


A new shell region is added to the simulation tree. In addition, an interface is formed between the fluid region and the shell region, based on the geometry of the **Wetted Wall** boundary surface.



Assign the fluid film phase to the **Wetted Wall Shell** region:

2. Select the **Regions** > **Wetted Wall Shell** node and set the **Continuum** to **Fluid Film**.



Specify the actual fluid film boundary conditions using the nodes within the **Wetted Wall Shell** region. These boundaries are formed from the edges of the original **Wetted Wall** boundary from which the shell region was created. By default all boundaries are set to no-slip wall, except the **Interface to Fluid [In-place 1]**, which is set to the **Fluid-Film Boundary** type.

Change the boundary type of the **Bottom edges**, **Symmetry edges**, and **Top edges** boundaries to their appropriate types.

3. Edit the **Regions** > **Wetted Wall Shell** > **Boundaries** node and set the boundary types as follows:

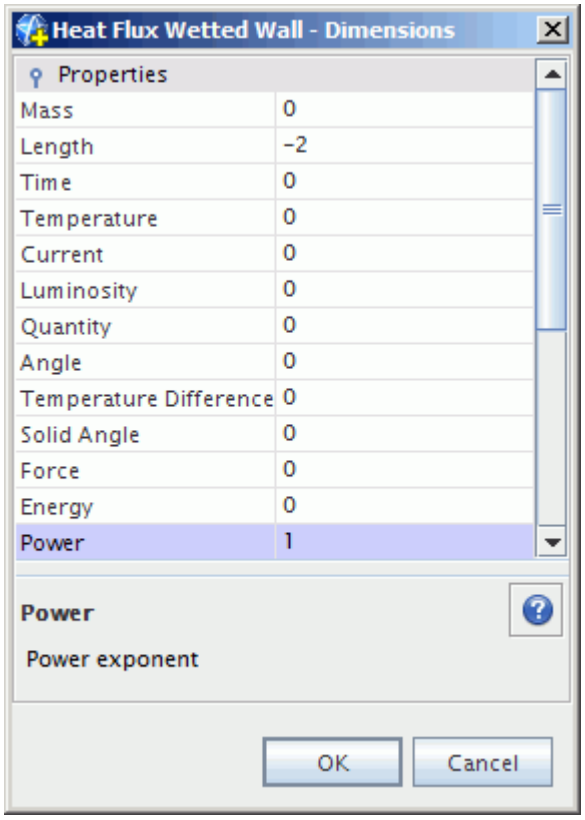
Boundary	Type
Bottom edges	**Outlet**
Symmetry edges	**Symmetry Plane**
Top edges	**Mass Flow Inlet**

The **Wetted Wall** is heated in one section only. You can define the heat flux in a field function.

4. Right-click the **Tools** > **Field Functions** node and select **New** > **Scalar** from the pop-up menu.
5. Rename the **User Field Function 1** to `Heat Flux Wetted Wall`.
6. Select the **Heat Flux Wetted Wall** node and set the following properties:

Property	Setting
Function Name	`heatFluxWettedWall`
Definition	`($$Position[2] <= 0.2)? 0 : 3000`

7. Set **Dimensions** as **Length** = `-2` and **Power** = `1`.



Specify the **Wetted Wall** heat flux.

8. Edit the **Regions** > **Wetted Wall Shell** > **Boundaries** > **Wetted Wall** node and set the following properties:

Node	Property	Setting
Physics Conditions > **Thermal Specification**	**Condition**	**Heat Flux**
Physics Values > **Heat Flux**	**Method**	**Field Function**
	Scalar Function	**Heat Flux Wetted Wall**

Set the boundary conditions for the inlet of the fluid film shell.

9. Edit the **Boundaries** > **Top edges** > **Physics Values** node and set the following properties:

Node	Property	Setting
Fluid Film Thickness	**Value**	`0.001 m`
Mass Flow Rate	**Value**	`0.015 kg/s`
Species Mass Fraction	**Value**	`[0.5, 0.5]`
Total Temperature	**Value**	`293.15 K`

10. Save the simulation.

Setting Up the Injector

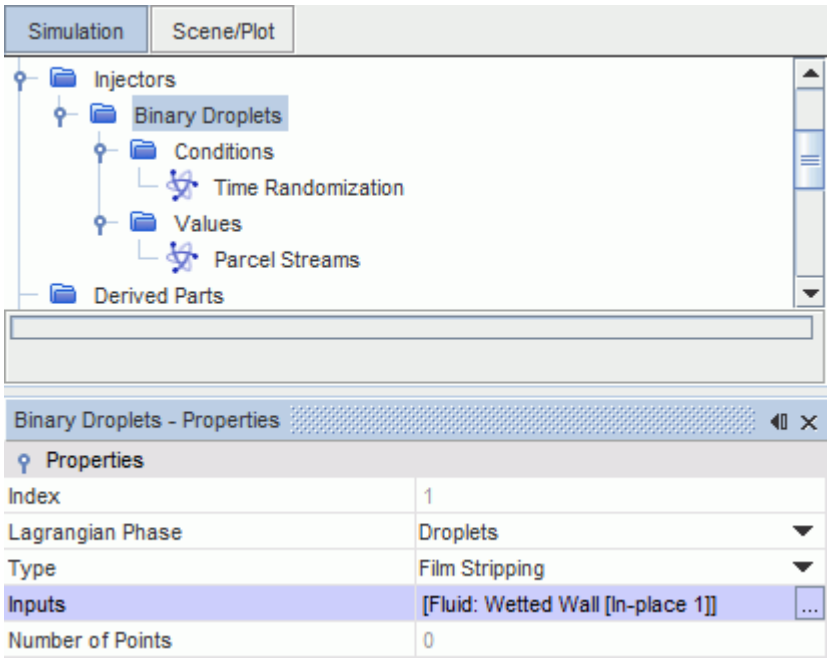
Set up a Lagrangian injector to model the film surface from which droplets leave the film and are injected into the bulk flow.

A special type of injector is used to signify a film surface from which droplets leave the film and are injected into the bulk flow.

To set up the injector:

1. Right-click **Injectors** and select **New**.
2. Rename the **Injectors** > **Injector 1** node to **Binary Droplets**.
3. Set the injector *Type* to **Film Stripping**.
4. Set the *Lagrangian Phase* to **Droplets**.
5. Click ... (**Custom Editor**) for the *Inputs* property.
6. In the selection dialog, select **Regions > Fluid > Boundaries > Wetted Wall [In-place 1]** and click **OK**.

   The *Properties* window appears as follows:



7. Save the simulation.

Setting Up the Film Stripping Model

Define the film stripping model as a Film-Lagrangian phase interaction.

The edge film stripping model is set as a Film-Lagrangian phase interaction in STAR-CCM+.

To set up the film stripping model:

Create a phase interaction.

1. Right-click **Continua** > **Physics 1** > **Models** > **Multiphase Interaction** > **Phase Interactions** and select **New**.
2. Rename the newly created **Phase Interaction 1** node to **Film Mixture - Droplets**.

To set up the Film-Lagrangian phase interaction:

3. Right-click **Multiphase Interaction** > **Phase Interactions** > **Film Mixture - Droplets** > **Models** and choose **Select Models...**

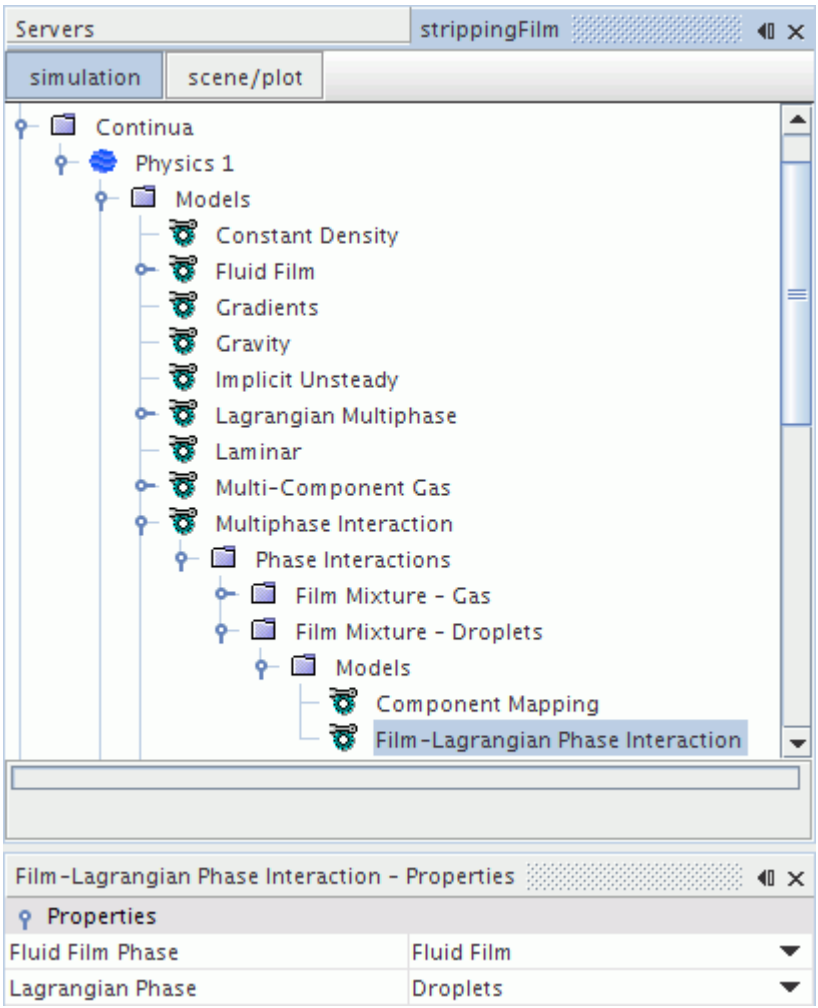
In the *Film Mixture - Droplets Model Selection* dialog:

4. Select **Film-Lagrangian Phase Interaction** from the *Phase Interaction Topology* box.
5. Click **Close**.

Set the phases for the Film-Lagrangian interaction:

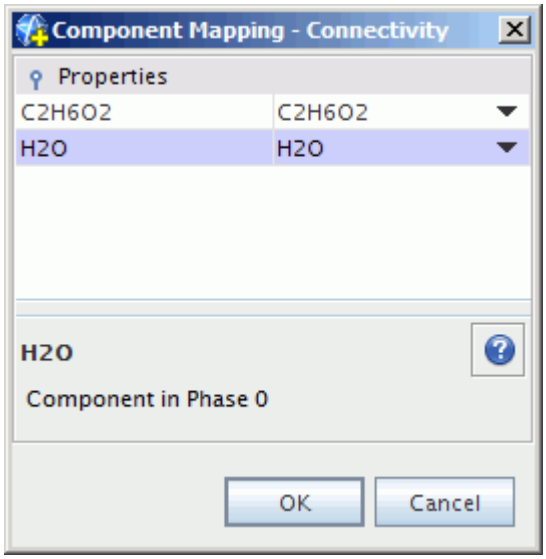
6. Select **Film Mixture - Droplets** > **Models** > **Film-Lagrangian Phase Interaction** and set the *Fluid Film Phase* to **Fluid Film** and the *Lagrangian Phase* to **Droplets**.

The **Component Mapping** node appears in the simulation tree to indicate that mapping is required between components of the film and droplet phases.



To set up the mapping between the phases:

7. Select **Multiphase Interaction** > **Phase Interactions** > **Film Mixture - Droplets** > **Models** > **Component Mapping** and click ... (**Custom Editor**) for the *Connectivity* property.
8. In the *Connectivity* dialog, map the fluid film components to their corresponding gas components, as shown below.



9. Click **OK**.

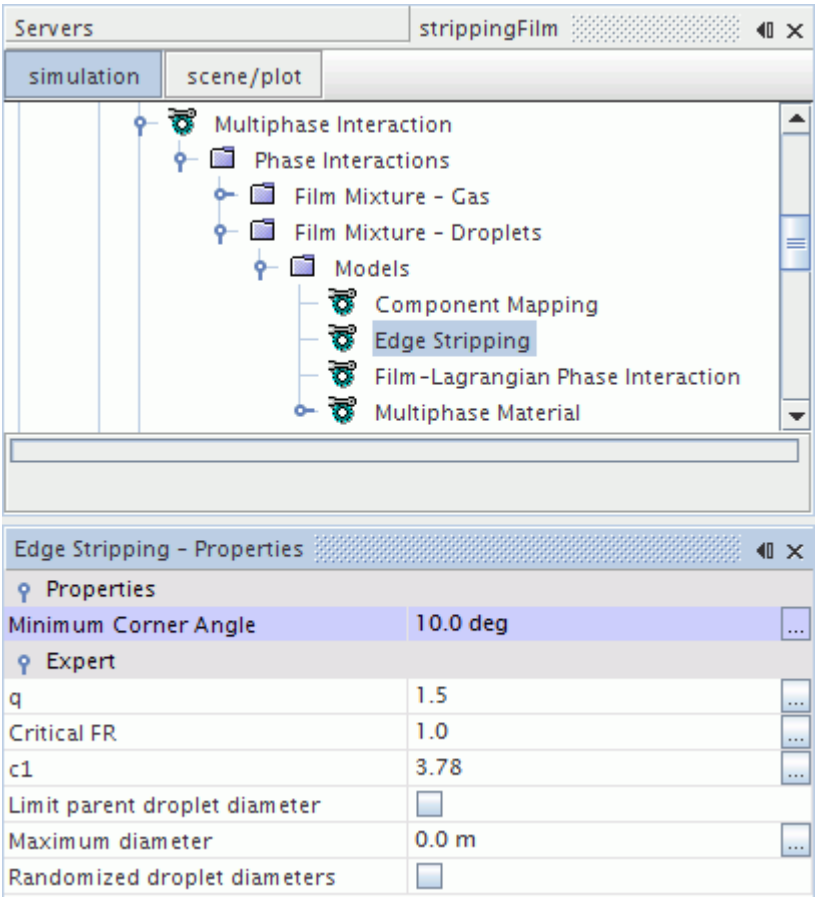
Select the edge stripping model and set its properties:

10. Right-click **Multiphase Interaction** > **Phase Interactions** > **Film Mixture - Droplets** > **Models** and choose **Select Models**.

In the *Film Mixture - Droplets Model Selection* dialog:

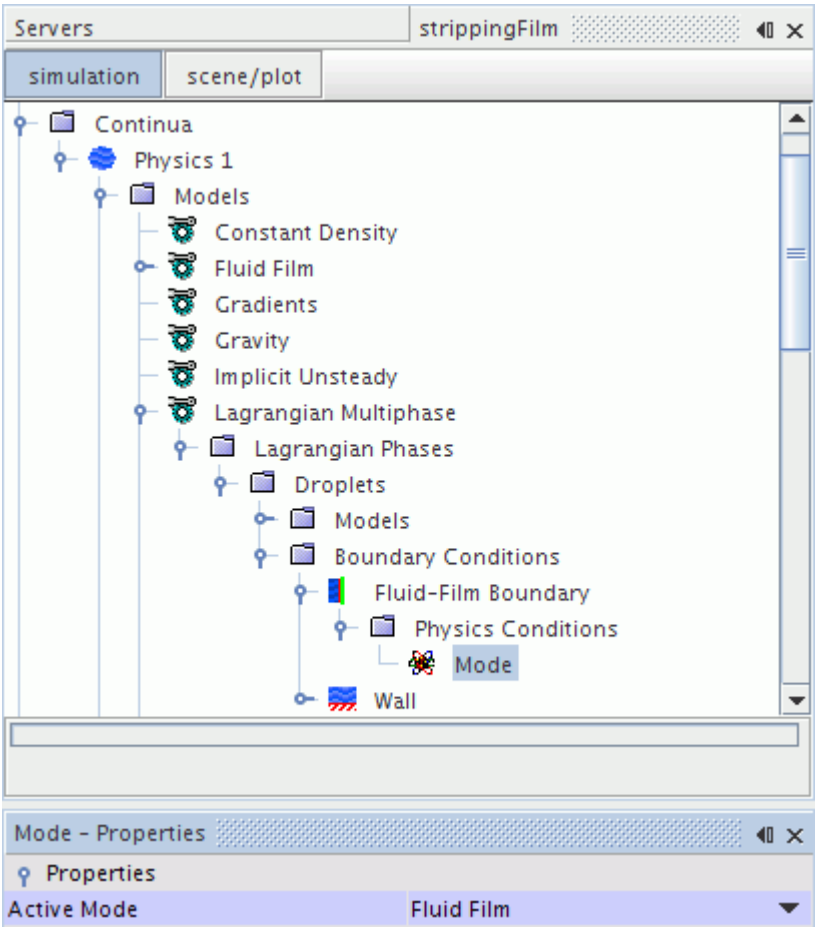
11. Select **Edge Stripping** from the *Optional Models* box.

    The **Multiphase Material** option is automatically selected.
12. Click **Close**.
13. Select the **Film Mixture - Droplets** > **Models** > **Edge Stripping** node.
14. To make sure that stripping occurs at the bend, set the *Minimum Corner Angle* to `10.0 deg`.



Having defined the **Film Mixture - Droplets** phase interaction, you can now set the boundary interaction mode for the fluid film boundary by editing the global Lagrangian phase boundary specification:

15. Select **Continua** > **Physics 1** > **Models** > **Lagrangian Multiphase** > **Lagrangian Phases** > **Droplets** > **Boundary Conditions** > **Fluid-Film Boundary** > **Physics Conditions** > **Mode** and set the *Active Mode* to **Fluid Film**.



The **Fluid Film** mode is available only when there is a correct match between the fluid film continuum and the Lagrangian phase. This correct match occurs when components have been correctly mapped and the same models are used in both fluid film and Lagrangian phases. If you make any changes to the fluid film or the Lagrangian phase models that invalidate the **Film Mixture - Droplets** phase interaction, the *Active Mode* reverts to **Rebound**. The **Fluid Film** option is not available until you validate the phase interaction. You must set the *Active Mode* back to **Fluid Film**.

16. Save the simulation.

Setting Reference Values

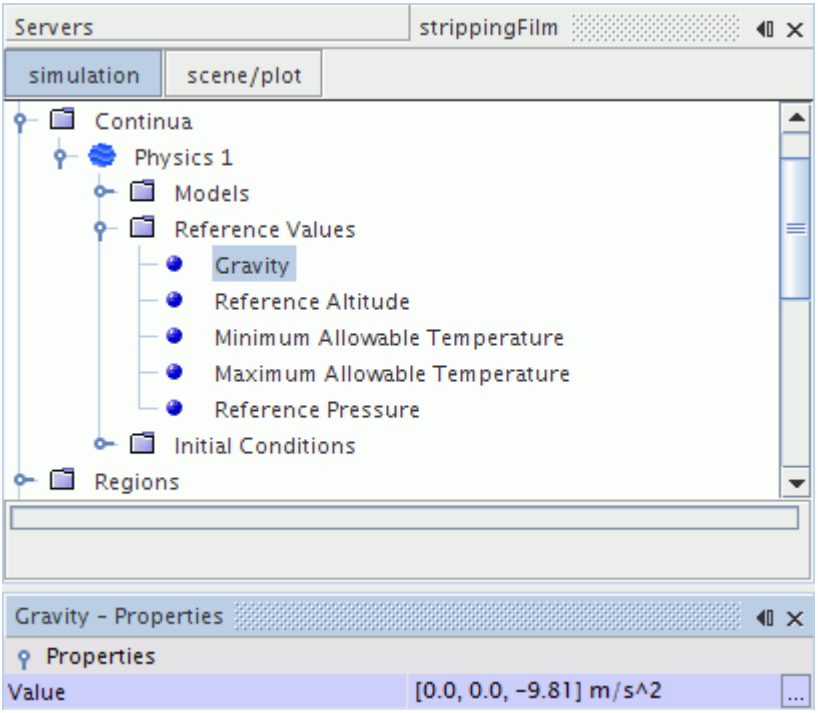
Set the reference value for the physics continuum.

Define the reference values of the **Physics 1** continuum to make sure that acceleration due to gravity acts in the expected direction. In this tutorial, the gravity acts in the negative Z direction.

To set the reference value:

1. Select the **Physics 1** > **Reference Values** node and ensure that the following property is set:

Node	Property	Setting
Gravity	**Value**	`[0.0, 0.0, -9.81] m/s^2`



2. Save the simulation.

Setting Solver Parameters and Stopping Criteria

Set the appropriate solver parameters and stopping criteria for the simulation.

To set the solver parameters and stopping criteria:

1. Edit the **Solvers** node and set the following properties:

Node	Property	Setting
Implicit Unsteady	**Time-Step**	`0.01 s`
	Temporal Discretization	**2nd-order**

2. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Physical Time	**Enabled**	Activated
	Maximum Physical Time	`5 s`
Maximum Inner Iterations	**Maximum Inner Iterations**	`5`

Node	Property	Setting
Maximum Steps	**Enabled**	Deactivated

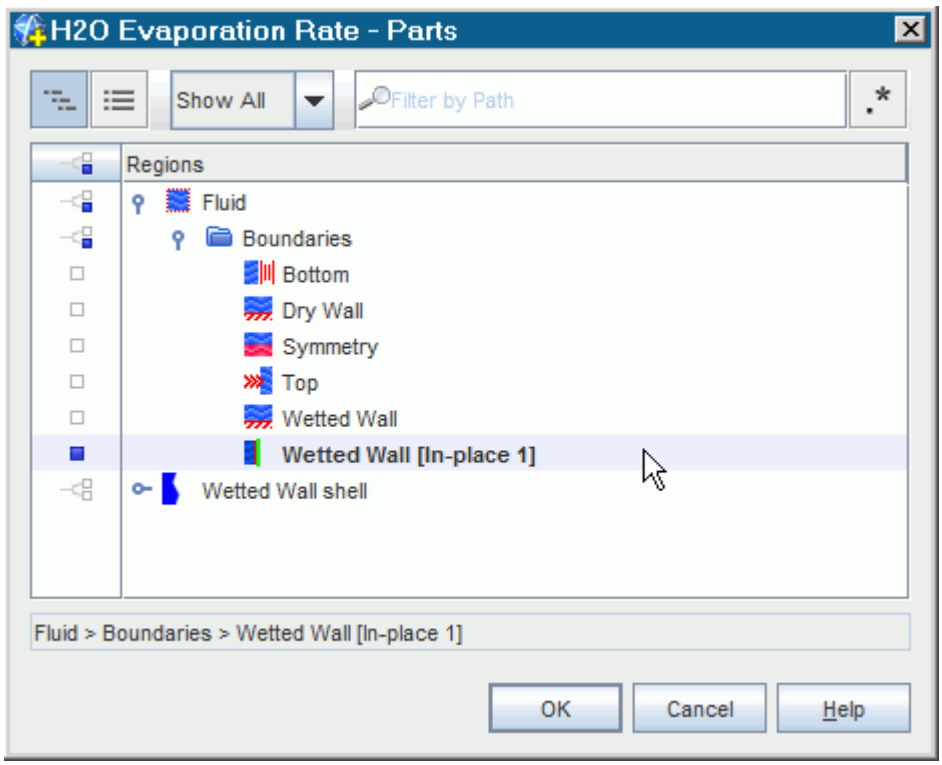
3. Save the simulation.

Setting Up Evaporation Rate Reports, Monitors, and Plots

Set up reports, monitors, and plots to monitor the film evaporation rate along the fluid film boundary.

To set up evaporation rate reports, monitors, and plots:

1. Right-click the **Reports** node and select **New Report** > **Surface Average**.
2. Rename the newly created **Surface Average 1** node to `H2O Evaporation Rate`.
3. Set the *Scalar Field Function* to **Film Evaporation Rate** > **Film Evaporation Rate of H2O**.
4. Click the ellipsis to the right of the *Parts* property.
5. In the dialog that appears, expand the **Regions** > **Fluid** node and select **Wetted Wall [In-place 1]**.



6. Click **OK**.

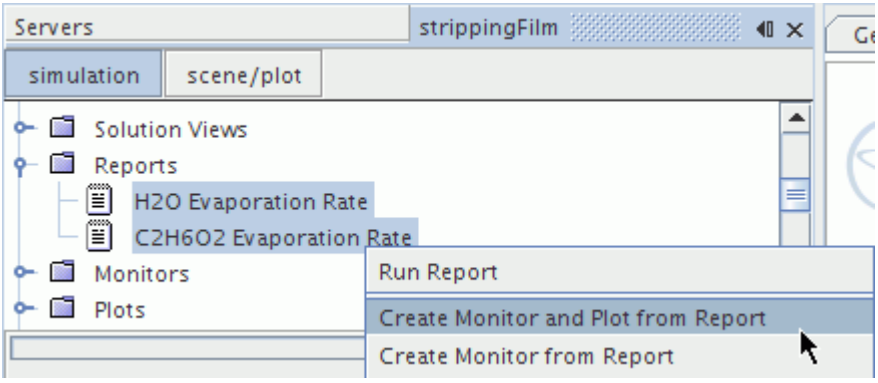
The *H2O Evaporation Rate - Properties* window looks as follows:

Copy this report to create a second one for the ethylene glycol evaporation rate:

7. Right-click **Reports** > **H2O Evaporation Rate** and select **Copy**.
8. Right-click **Reports** and select **Paste**.
9. Rename **Copy of H2O Evaporation Rate** to **C2H6O2 Evaporation Rate**.
10. Select **Reports** > **C2H6O2 Evaporation Rate** and, set the *Scalar Field Function* to **Film Evaporation Rate** > **Film Evaporation Rate of C2H6O2**.

Create a monitor and a plot for both reports.

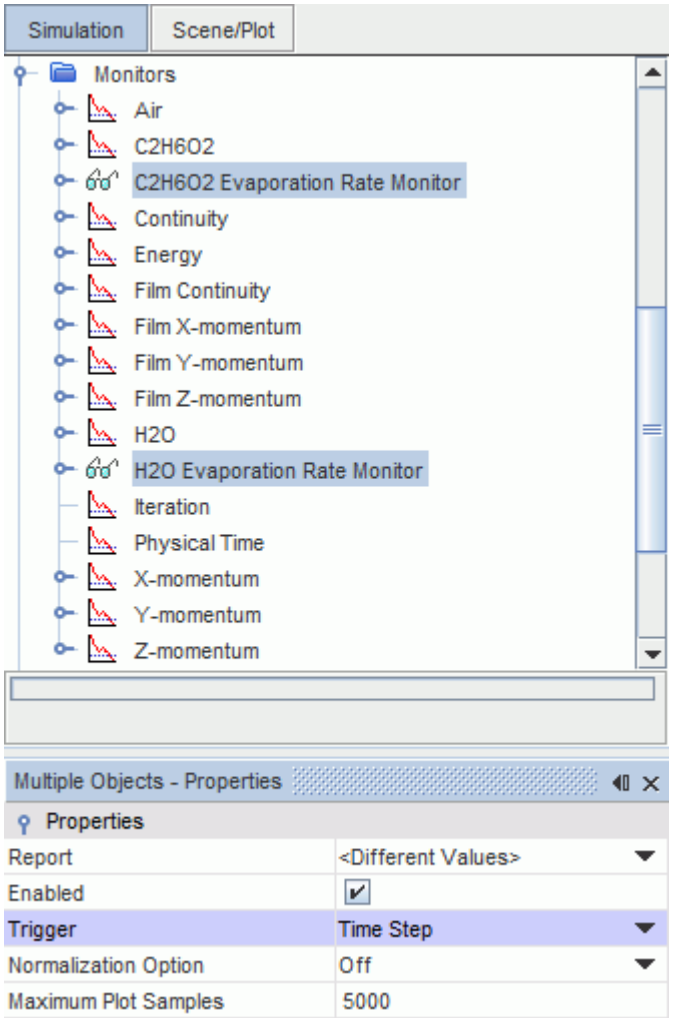
11. Select both the **H2O Evaporation Rate** and **C2H6O2 Evaporation Rate** nodes, right-click either and select **Create Monitor and Plot from Report**.



12. Click **Single Plot** to display both evaporation rates on the same plot.

Set the monitors to update at each time step:

13. Select both the **Monitors** > **C2H6O2 Evaporation Rate Monitor** and the **H2O Evaporation Rate Monitor** nodes, and set *Trigger* to **Time Step**.



Format the monitor plot appearance:

14. Rename **Plots** > **Reports Plot** to **Evaporation Rates**.
15. Right-click the **Plots** > **Evaporation Rates** node and select **Open**.
16. Select **Plots** > **Evaporation Rates** > **Axes** > **Bottom Axis** > **Title** and set the *Title* to `Time (s)`.
17. Select **Plots** > **Evaporation Rates** > **Axes** > **Left Axis** > **Title** and set the *Title* to `Surface Average of Film Evaporation Rate (kg/m^2-s)`.

Setting Up a Scalar Scene

Set up a scalar scene to display the fluid film temperature.

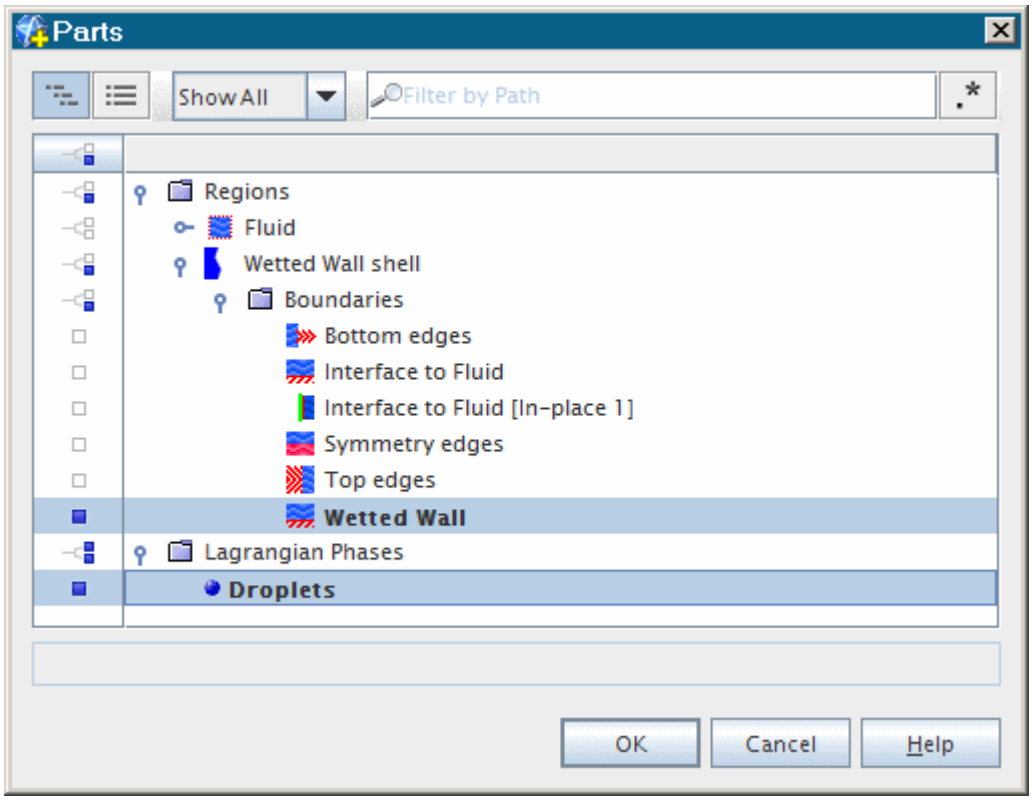
To set up a scalar scene:

Create a scalar scene for fluid film temperature.

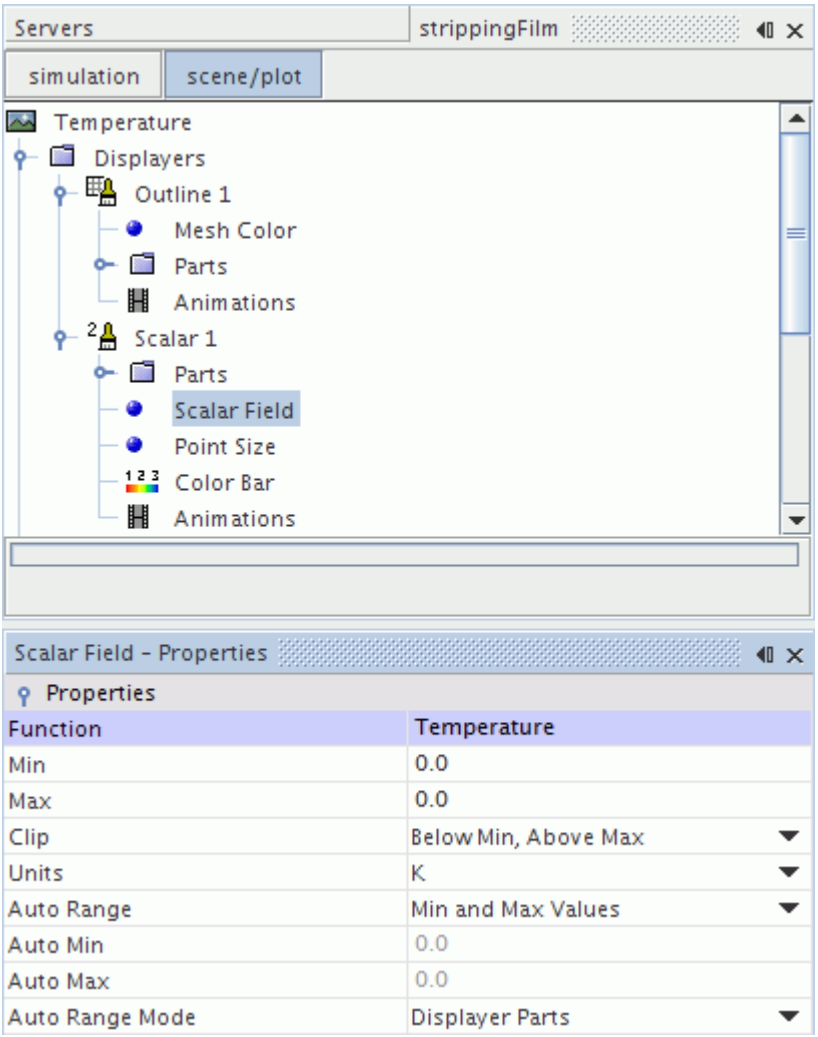
1. Right-click the **Scenes** node and select **New Scene** > **Scalar**.

A new sub-node **Scalar Scene 1** is created under the **Scenes** node.

2. Rename the **Scenes** > **Scalar Scene 1** node to `Temperature`.
3. Click **scene/plot**.
4. Select the **Temperature** > **Displayers** > **Scalar 1** > **Parts** node and then click the ellipsis to the right of the *Parts* property. In the dialog that appears, expand the **Regions** > **Wetted Wall shell** node and select **Wetted Wall**.
5. Expand the **Lagrangian Phases** node and select **Droplets** to visualize the edge film stripping.



6. Click **OK**.
7. Select the **Temperature** > **Displayers** > **Scalar 1** > **Scalar Field** node and click to the right of the *Function* property and set it to **Temperature**.



8. Click (**Save-Restore-Select views**) in the toolbar and select **Restore View** > **View 1**.
9. Save the simulation.

Running the Simulation

Preparation of the simulation is now complete, and the simulation can be run.

To run the simulation:

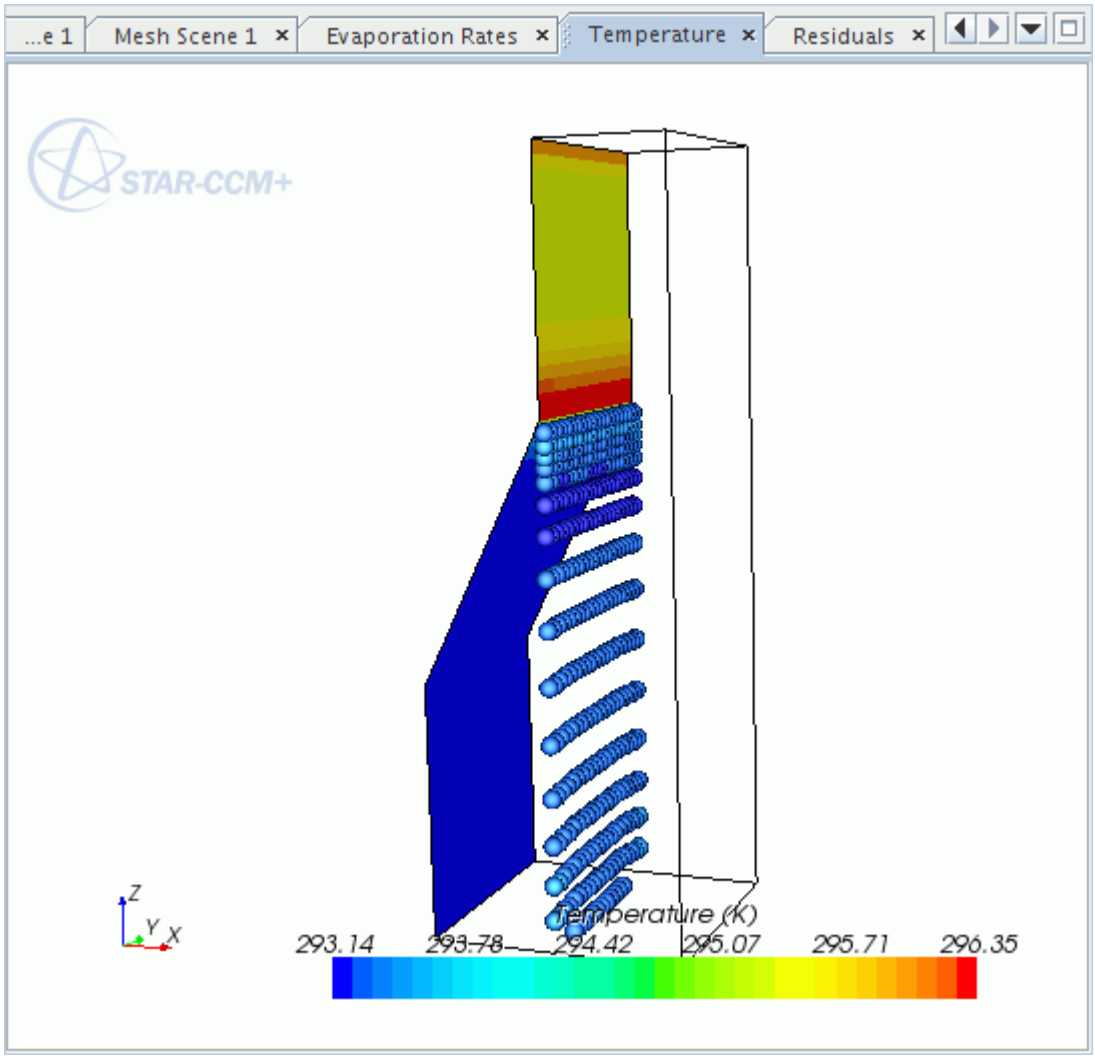
1. Click (**Run**) in the *Solution* toolbar.

   The solution progress is displayed in the *Output* window. The *Residuals* display is created automatically in the *Graphics* window and shows the progress of the solvers.

   While the simulation is running, you can click the tabs at the top of the *Graphics* window to view scenes and plots.

   During the run, you can stop the process by clicking (**Stop**) on the toolbar. If you do stop the simulation, you can click (**Run**) to resume it.

2. Activate the temperature scene and observe the change in wall and droplets temperature as the solution progresses.



The example above shows the temperature in the early stages of the run. The simulation continues until 5 seconds of physical time is complete.

When the last iteration is complete, the following message is displayed in the *Output* window:

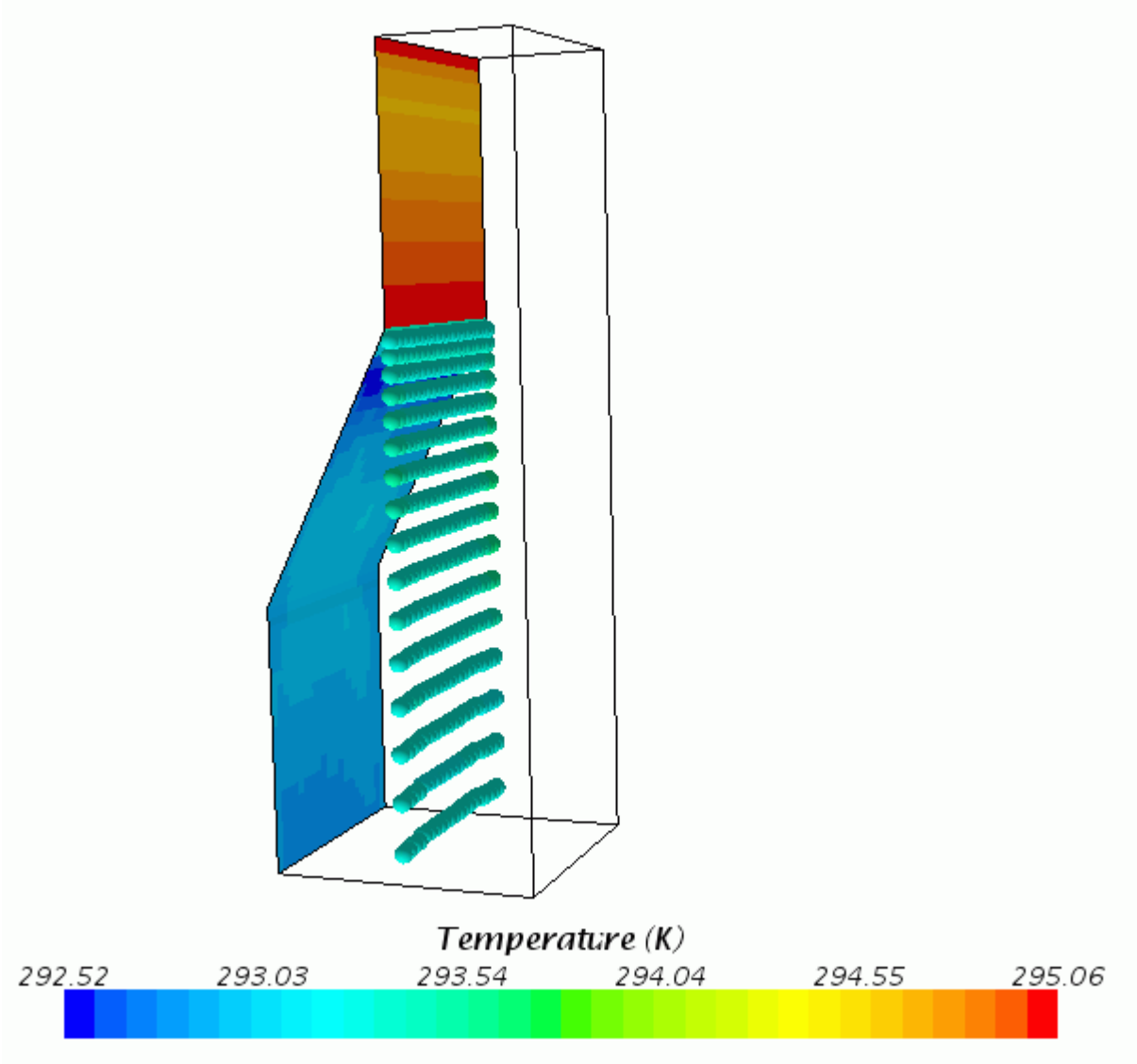
```
Stopping criterion Maximum Physical Time satisfied.
```

3. When the simulation has finished running, save it.

Visualizing the Results

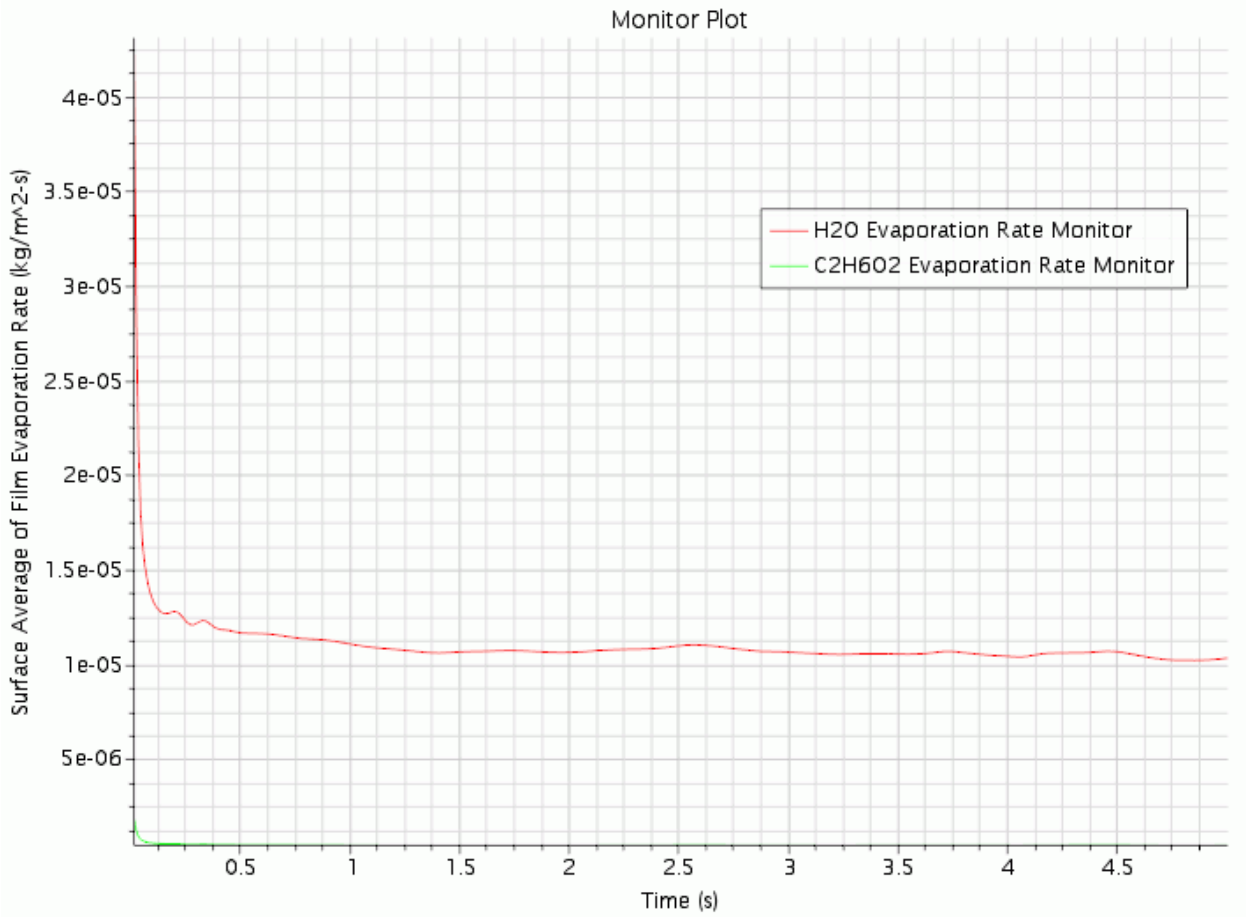
Examine the results of the simulation.

The temperature contour for the solution at the final time step is as shown below.



1. Activate the evaporation rates monitor plot.

   The evaporation rates plot for the converged solution is as shown below.



Summary

This tutorial demonstrated how to use the fluid film modeling capability of STAR-CCM+ to set up a multi-component film scenario with evaporation and edge stripping.

This tutorial has introduced the following STAR-CCM+ features:

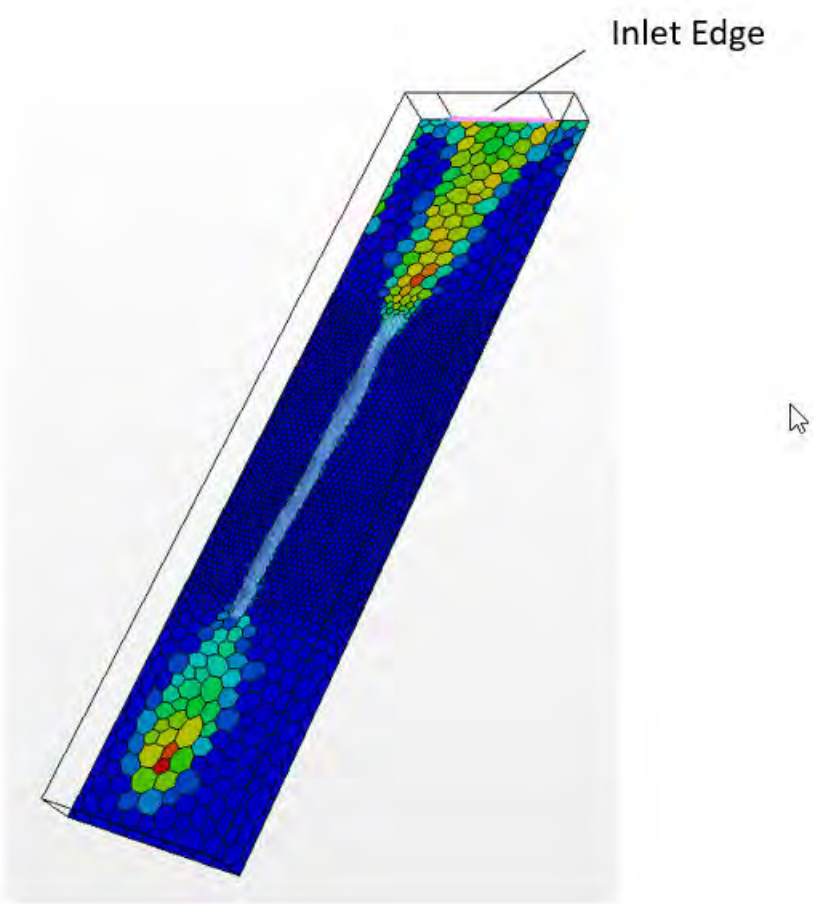
- Defining models for fluid film model.
- Setting fluid film models.
- Setting Film-Physics Continuum and Film-Lagrangian phase interactions.
- Setting the fluid film evaporation and condensation model.
- Setting up shell region and boundary conditions.
- Setting up the edge stripping model.
- Running the solver for a set physical time.

Fluid Film - VOF: Rivulets on an Inclined Plane

Situations occur where a thin film of liquid forms on a solid surface, for example, rainwater on the windshield of a car. This film can flow and accumulate, and gradually grow into a "thick" film, or a pool of liquid that can no longer be described as a fluid film. These situations are multi-scale problems which, if resolved on all scales (using the VOF multiphase model), would require significant computer resources.

STAR-CCM+ offers a modeling approach to solve these types of problems in an efficient way. You can combine the VOF model and the thin fluid film model, such that the VOF model is used in areas where the mesh is sufficiently fine to resolve the flow of liquid, and the fluid film model is used in areas where the mesh size is larger than the film thickness.

This tutorial models a thin stream of water flowing down a surface that is inclined at an angle of 45° with the horizontal. The inclined plane is 0.2 meters long and 0.05 meters wide. The computational domain is 0.01 meters high. Water enters the domain through the inlet edge at the top at a rate of 0.005 kg/s and flows down the incline as a thin film or rivulet. The simulation domain is shown below:



The mesh resolution is not uniform. The upper and lower regions of the inclined plane use a coarse mesh, but the central region uses a finer mesh. In this case, the mesh in the central region is sufficiently fine to resolve the flow of liquid, so the VOF model is used. The fluid film model is used in the upper and lower regions.

Contents:

Prerequisites
Importing the Mesh and Naming the Simulation
Selecting the Physics Models
Defining the Water Film and Air Phases
Defining the Phase Interactions
Setting Initial Conditions and Gravity
Setting Boundary Conditions
Specifying the Transition Volume Fraction
Setting Solver Parameters and Stopping Criteria
Visualizing the Water Film
Running the Simulation
Visualizing the Results
Summary

Prerequisites

The instructions in the Fluid Film - VOF: Rivulets on an Inclined Plane tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Importing the Mesh and Naming the Simulation

Launch STAR-CCM+, import the supplied geometry, and save the simulation.

To import the mesh and name the simulation:

1. Launch STAR-CCM+ and start a new simulation.
2. Select **File** > **Import** > **Import Volume Mesh**.
3. In the *Open* dialog, choose `doc/startutorialsdata/multiphaseFlow/data/rivuletsOnInclinedPlane.ccm`, then click **Open**.

   STAR-CCM+ provides feedback on the import process, which takes a few seconds, in the *Output* window.
4. Save the simulation as `RivuletsOnInclinedPlane.sim`.

Selecting the Physics Models

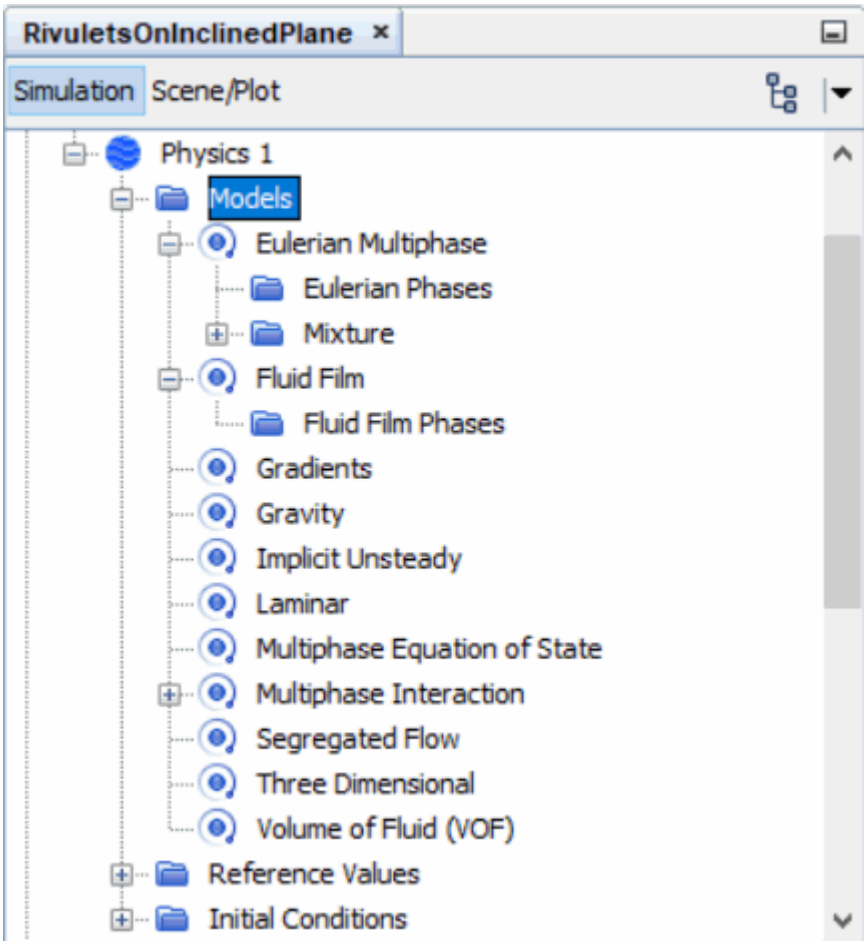
The water flows down the inclined plane due to gravity. The two-phase flow of water film and air is modeled using a hybrid Film-VOF approach.

To select the physics models:

1. For the physics continuum, **Continua** > **Physics 1**, select the following models in order:

Group Box	Model
Enabled Models	**Three Dimensional** (Pre-selected)
Time	**Implicit Unsteady**
Material	**Eulerian Multiphase** **Multiphase Interaction** (Selected automatically)
Eulerian Multiphase Model	**Volume of Fluid (VOF)** **Multiphase Equation of State** (Selected automatically) **Segregated Flow** (Selected automatically) **Gradients** (Selected automatically)
Viscous Regime	**Laminar**
Optional Models	**Fluid Film** **Gravity**

2. Click **Close**.
3. To review the models, open the **Physics 1** > **Models** node.



4. Save the simulation.

Defining the Water Film and Air Phases

As the water phase can be represented in this simulation with either the VOF model or the Fluid Film model, the material properties of the Eulerian (VOF) water phase and the Fluid Film water phase must be the same.

To create the phases and select the phase models:

Create the VOF gas phase.

1. In the **Physics 1** continuum, right-click the **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and select **New**.
2. Rename the **Phase 1** node to **Air**.
3. For the **Air** phase, select the following models:

Group Box	Model
Enabled Models	**Laminar** (selected automatically)
Material	**Gas**
Equation of State	**Constant Density**

4. Click **Close**.

Create the VOF liquid phase.

5. Create a second Eulerian phase and rename it to **Water**.
6. For the **Water** phase, select the following models:

Group Box	Model
Enabled Models	**Laminar** (selected automatically)
Material	**Liquid**
Equation of State	**Constant Density**

7. Click **Close**.

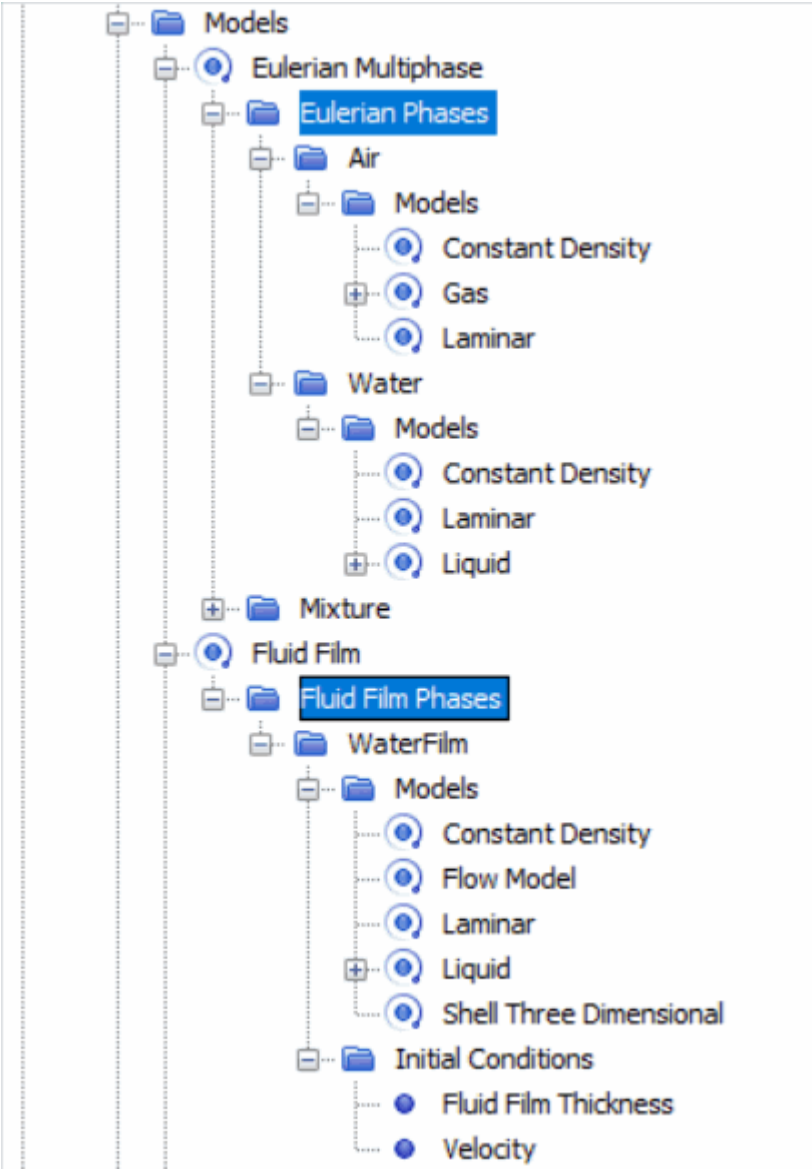
Create the Fluid Film phase.

8. In the **Physics 1** continuum, right-click the **Models** > **Fluid Film** > **Fluid Film Phases** node and select **New**.
9. Rename the **Phase 1** node to **WaterFilm**.
10. For the **WaterFilm** phase, select the following models in order:

Group Box	Model
Space	**Shell Three Dimensional** **Laminar** (selected automatically) **Flow Model** (selected automatically)
Material	**Liquid**
Equation of State	**Constant Density**

11. Click **Close**.

    The **Eulerian Phases** node and the **Fluid Film Phases** node appear as shown below:



12. Save the simulation.

Defining the Phase Interactions

You require phase interactions to describe how the phases – water and air – mutually affect each other. The shape and position of the interface between the water film and air is influenced significantly by surface tension effects.

For this hybrid VOF-Fluid Film simulation, the water film is defined both as a Eulerian (VOF) phase and as a Fluid Film phase with the same material properties. Therefore, you require two separate phase interactions both with surface tension activated to describe the interdependency between the water film and air:

- Film-VOF phase interaction between the air VOF phase and the water fluid film phase
- VOF-VOF phase interaction between the air VOF phase and the water VOF phase

To model the transition of the water fluid film phase to the water VOF phase depending on mesh resolution, a third phase interaction is needed:

- Film-VOF phase interaction between the water fluid film phase and the water VOF phase. This phase interaction invokes the Resolved Fluid Film model.

  The **Resolved Fluid Film** model automatically uses the model (fluid film or VOF) that is the most appropriate. The fluid film model is used in areas where a liquid film is present and where the numerical mesh is not sufficiently fine to resolve the flow and the distribution of phases with the VOF model. As the fluid film gets thicker, or as the film enters a region with a finer mesh, the flow of liquid film is modeled using the VOF model. Similarly, if a VOF liquid phase enters a region with a coarse mesh, or if the film thickness gets smaller, the flow of liquid is modeled using the fluid film model.

To define the phase interactions:

1. Create the Film-VOF phase interaction between the *Air* and the *WaterFilm* phases.
   a) In the **Physics 1** continuum, right-click the **Models** > **Multiphase Interaction** > **Phase Interactions** node and create a new phase interaction.
   b) Rename the **Phase Interaction 1** node to `Fluid Film Surface Tension`.
   c) For the **Fluid Film Surface Tension** phase interaction, open the *Fluid Film Surface Tension Model Selection* dialog and, in the *Phase Interaction Topology* group box, select **Film-VOF Phase Interaction**.
   d) Click **Close**.
   e) Select the **Fluid Film Surface Tension** > **Models** > **Film-VOF Phase Interaction** node and set the following properties:

Property	Setting
VOF Phase	**Air**
Fluid Film Phase	**WaterFilm**

   f) Reopen the *Fluid Film Surface Tension Model Selection* dialog and select the following models:

Group Box	Model
Optional Models	**Surface Tension** **Multiphase Material** (selected automatically)

   g) Click **Close**.

   A minimum film thickness of `1.0E-5 m` and the default surface tension value for the air-water interface, `0.072 N/m`, are appropriate for this simulation.

   h) Edit the **Fluid Film Surface Tension** > **Models** node and set the following properties:

Node	Property	Setting
Multiphase Material > **Multiphase Material** > **Material Properties** > **Surface Tension** > **Constant**	**Value**	`0.072 N/m`
Surface Tension	**Minimum Film Thickness**	`1.0E-5 m`

2. Create the VOF-VOF phase interaction between the *Water* and the *Air* phases.

a) Create another new phase interaction and rename it to `VOF Water Surface Tension`.

b) For the **VOF Water Surface Tension** phase interaction, open the *VOF Water Surface Tension Model Selection* dialog and, in the *Phase Interaction Topology* group box, select **VOF-VOF Phase Interaction**.

c) Click **Close**.

d) Select the **VOF Water Surface Tension** > **Models** > **VOF-VOF Phase Interaction** node and set the following properties:

Property	Setting
Primary Phase	**Water**
Secondary Phase	**Air**

e) Reopen the *VOF Water Surface Tension Model Selection* dialog and select the following models:

Group Box	Model
Optional Models	**Surface Tension Force** **Multiphase Material** (selected automatically)

f) Click **Close**.

The surface tension for the air-water interface must be set to the same value for both phase interactions.

g) Edit the **VOF Water Surface Tension** > **Models** > **Multiphase Material** node and set the following property:

Node	Property	Setting
Multiphase Material > **Material Properties** > **Surface Tension** > **Constant**	**Value**	`0.072 N/m`

3. Create the Film-VOF Phase Interaction between the *Water* and the *WaterFilm* phases.

a) Create another new phase interaction and rename it to `Resolved Fluid Film`.

b) For the **Resolved Fluid Film** phase interaction, open the *Resolved Fluid Film Model Selection* dialog and, in the *Phase Interaction Topology* group box, select **Film-VOF Phase Interaction**.

c) Click **Close**.

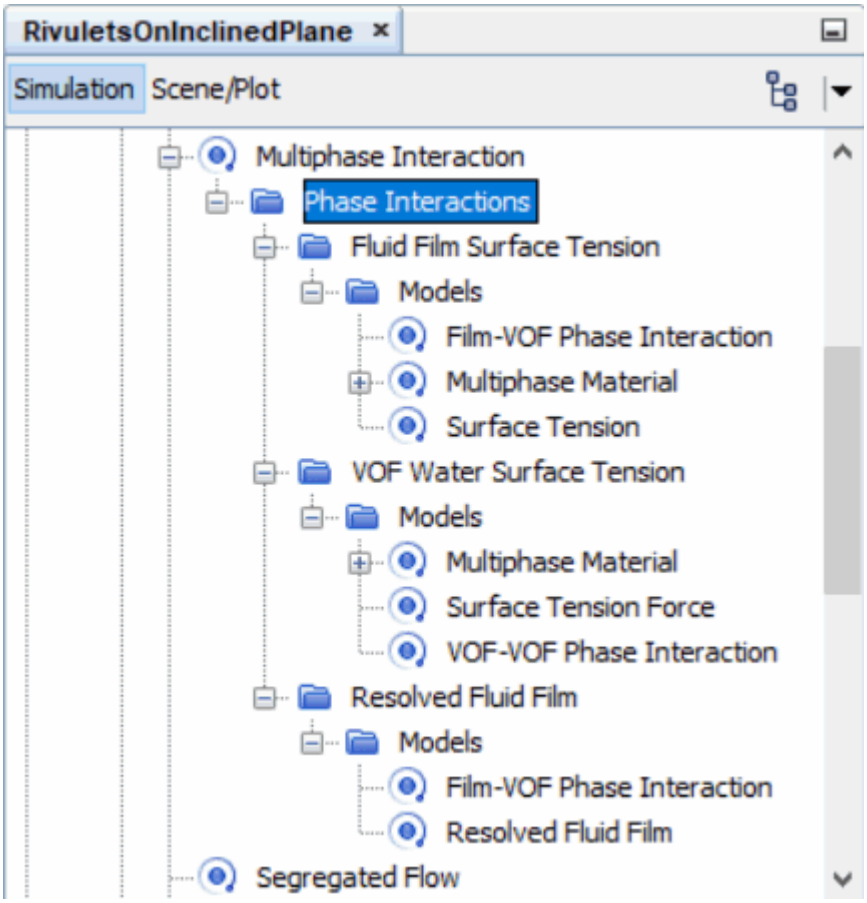
d) Select the **Resolved Fluid Film** > **Models** > **Film-VOF Phase Interaction** node and set the following properties:

Property	Setting
VOF Phase	**Water**
Fluid Film Phase	**WaterFilm**

e) Reopen the *Resolved Fluid Film Model Selection* dialog and, in the *Optional Models* group box, select **Resolved Fluid Film**.

f) Click **Close**.

The **Phase Interactions** node appears as shown below:



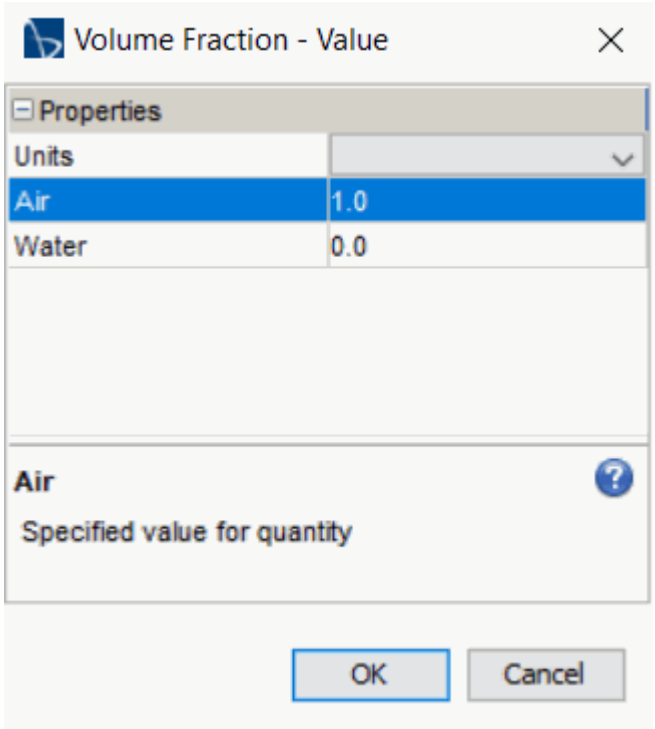
4. Save the simulation.

Setting Initial Conditions and Gravity

At the start of the simulation, the simulation domain contains air only. Set the gravity vector to operate in the negative Y direction.

To set the initial conditions and gravity for the **Physics 1** continuum:

1. Select the **Physics 1** > **Initial Conditions** > **Volume Fraction** node and set *Value* to `[1.0, 0.0]`.



2. Select the **Physics 1** > **Reference Values** > **Gravity** node and set *Value* to `[0.0, -9.81, 0.0]`.
3. Save the simulation.

Setting Boundary Conditions

The inclined plane is modeled to be in an open environment using slip walls at the sides of the computational domain and a pressure boundary at the front. In the case of inflow through the pressure boundary due to recirculation, only air enters the computational domain.

1. Edit the **Regions** > **Body_1** > **Boundaries** node and set the following properties:

Node	Property	Setting
Front > **Physics Values** > **Volume Fraction**	**Value**	`[1.0, 0.0]`
Top > **Physics Conditions** > **Shear Stress Specification**	**Method**	**Slip**
Sides > **Physics Conditions** > **Shear Stress Specification**	**Method**	**Slip**

Assign the fluid film continuum to the shell region.

2. Select the **Regions** > **PartiallyWetSurface_shell** node and set the *Continuum* to **WaterFilm**.

At the inlet, the fluid film enters the domain at a rate of `0.005 kg/s` and with a thickness of `5.0E-4 m`. In this tutorial, the contact angle is 90 degrees. This value is neutral – neither hydrophobic nor hydrophilic – and is reasonable for water on glass.

3. Edit the **Regions** > **PartiallyWetSurface_shell** > **Boundaries** node and set the following properties:

Node	Property	Setting
Inlet Edge > **Physics Values**		
⌊ **Fluid Film Thickness**	**Value**	`5.0E-4 m`
⌊ **Mass Flow Rate**	**Value**	`0.005 kg/s`
Sides Edges > **Phase Conditions** > **Fluid Film Surface Tension** > **Physics Values** > **Contact Angle**	**Value**	`90.0 deg`
PartiallyWetSurface > **Phase Conditions** > **Fluid Film Surface Tension** > **Physics Values** > **Contact Angle**	**Value**	`90.0 deg`

4. Save the simulation.

Specifying the Transition Volume Fraction

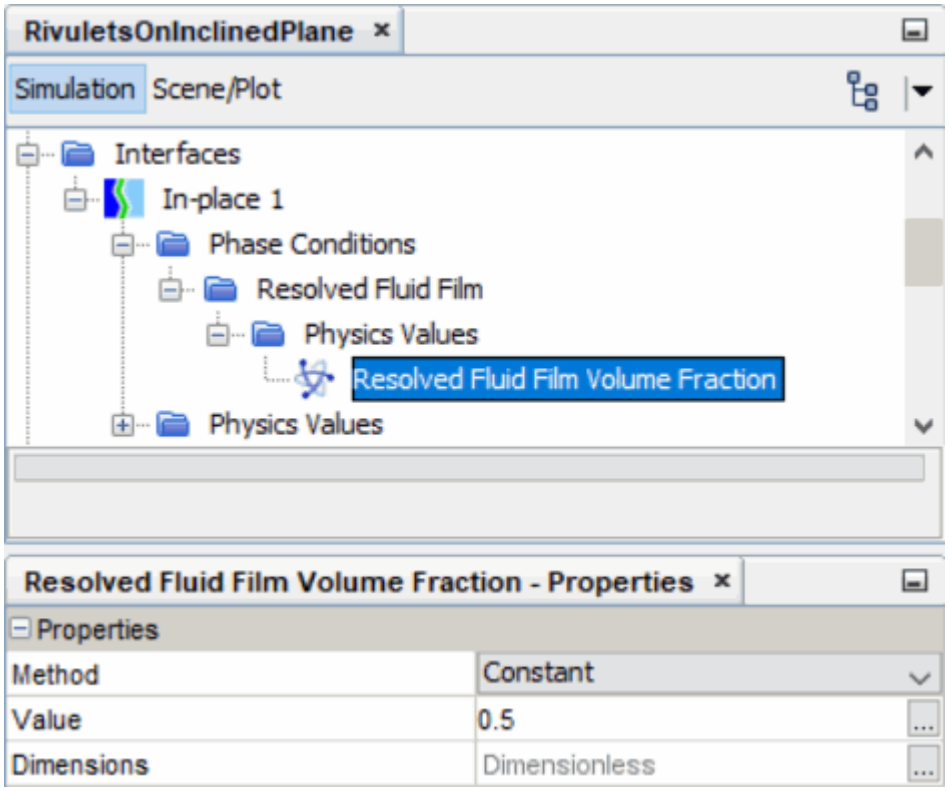
The transition between the fluid film and VOF phase models is based on the volume fraction of liquid (the total of the fluid film phase and the VOF phase combined) at the interface and the transition volume fraction α_{TRANS} that you specify for the interface.

Fluid film transitions into the VOF phase when the total volume fraction in the neighbor volume cell is greater than α_{TRANS}. The VOF phase transitions into fluid film when the total volume fraction is less than α_{TRANS}. To ensure stability, mass is transferred at a finite rate – the transition of mass from one phase to the other takes place over a few iterations rather than in a single time-step.

For more information, see Modeling a Fluid Film with Different Characteristic Length Scales.

The default transition volume fraction α_{TRANS} is a constant value of 0.5. This value is acceptable for this simulation.

1. Select the **Interfaces** > **In-place 1** > **Phase Conditions** > **Resolved Fluid Film** > **Physics Values** > **Resolved Fluid Film Volume Fraction** node and check that *Value* is set to `0.5`.



2. Save the simulation.

Setting Solver Parameters and Stopping Criteria

This hybrid fluid film-VOF simulation is a transient analysis that you run using the implicit unsteady solver with the default time-step of 0.001s. The simulation stops after 0.35s.

To set the stopping criteria:

1. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Inner Iterations	**Enabled**	Activated
	Maximum Inner Iterations	`10`
Maximum Physical Time	**Enabled**	Activated
	Maximum Physical Time	`0.35 s`
Maximum Steps	**Enabled**	Deactivated

2. Save the simulation.

Visualizing the Water Film

Create three scalar scenes to visualize the water flowing down the inclined plane. One scene displays the film thickness in the fluid film regions and the second scene displays the surface of the rivulet volume in the VOF region. The third scene displays both the fluid film and VOF components.

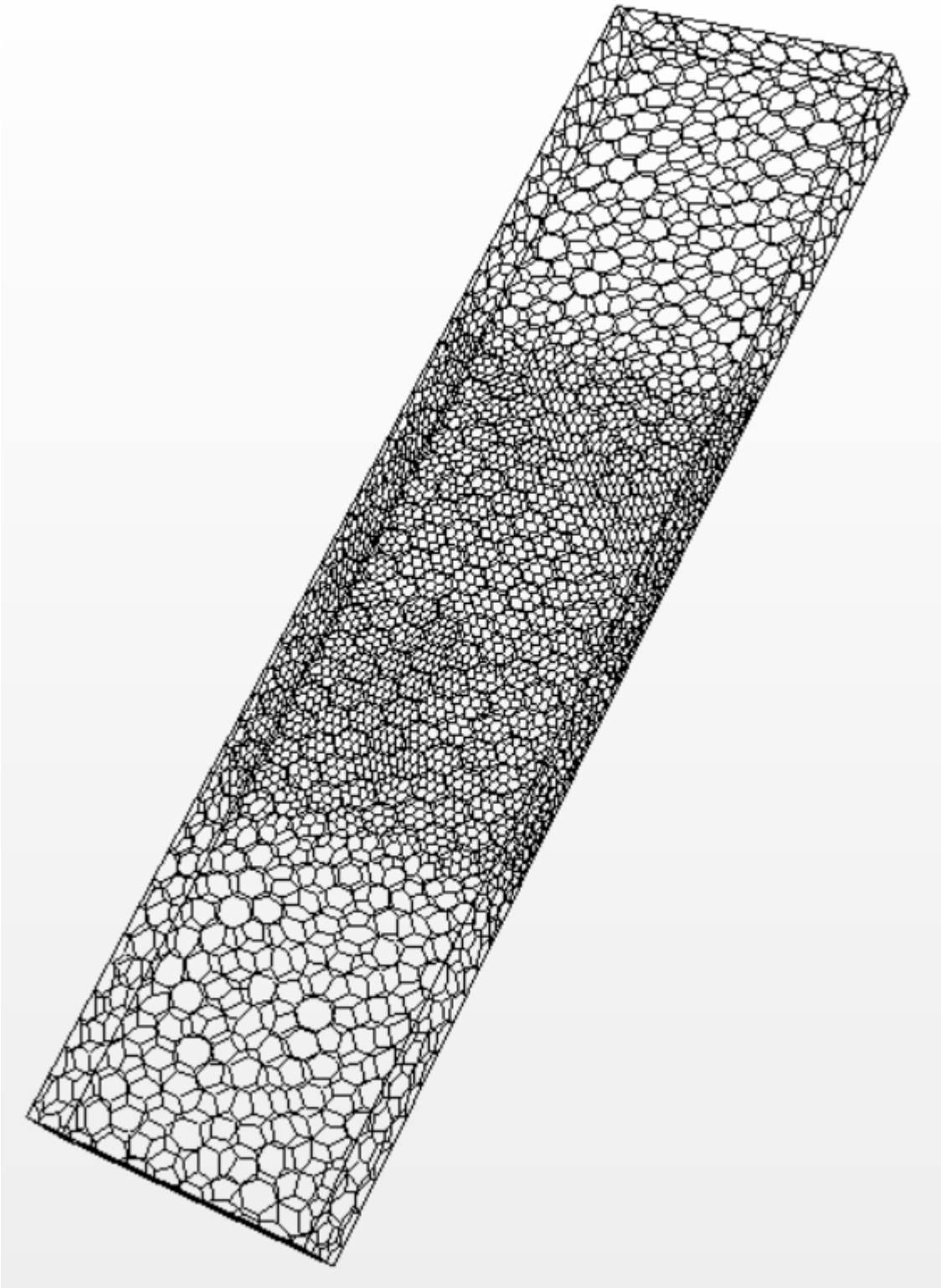
1. Create an isosurface with a volume fraction value of 0.5 that represents the VOF resolved water film surface:
   a) Right-click the **Derived Parts** node and select **New Part** > **Isosurface**.

      The *Create Isosurface* dialog appears in the *Edit* window.
   b) In the *Input Parts* box, select **Body_1**.
   c) Set the **Scalar** function to **Volume Fraction** > **Volume Fraction of Water**.
   d) Set the **Extraction Mode** to **Single Value**.
   e) Set the **Isovalue** to `0.5`.
   f) In the *Display* group box, select **No Displayer**.
   g) Click **Create** and then click **Close**.

      A new derived part, **Isosurface**, is created.

Create the scalar scenes.

2. Right-click the **Scenes** node, and select **New Scene** > **Scalar**.
3. Select the **Scalar Scene 1** node and then set **Mesh Override** to **Show All Meshes**.
4. Manipulate the scene to make the mesh orientation similar to the example below:



5. Use the Copy and Paste features to create two more copies of **Scalar Scene 1**.
6. Rename the scalar scenes as **Fluid Film Scene**, **Resolved Water Scene**, and **Combined Scene**.

Set up the first scalar scene to display the film thickness.

7. Select the **Fluid Film Scene** node and set the following properties:

Node	Property	Setting
Fluid Film Scene	**Mesh Override**	**Use Displayer Property**
⌊ **Displayers** > **Scalar 1**	**Contour Style**	**Filled**
	Display Mesh	Enabled
⌊ **Parts**	**Parts**	**Regions** > **PartiallyWetSurface shell** > **Boundaries** > **Interface to Body 1 [In-place 1]**
⌊ **Scalar Field**	**Function**	**Fluid Film Thickness**
⌊ **Color Bar**	**Orientation**	**Vertical**
	Position	`[0.05, 0.1]`

Set up the second scalar scene to display the mesh outline of the resolved rivulet in the VOF region.

8. Select the **Resolved Water Scene** > **Displayers** node and set the following properties:

Node	Property	Setting
Outline 1	**Opacity**	`0.6`
	Color Mode	**Constant**
⌊ **Mesh Color**	**Mesh Color**	**Sky Blue**
⌊ **Parts**	**Parts**	**Derived Parts** > **Isosurface**

Set up the third scalar scene to display both the liquid film and VOF components.

9. Select the **Combined Scene** > **Displayers** node and set the following properties:

Node	Property	Setting
Outline 1	**Opacity**	`0.6`
	Color Mode	**Constant**
⌊ **Mesh Color**	**Mesh Color**	**Sky Blue**
⌊ **Parts**	**Parts**	**Derived Parts** > **Isosurface**
Scalar 1	**Contour Style**	**Filled**
⌊ **Parts**	**Parts**	**Regions** > **PartiallyWetSurface shell** > **Boundaries** > **Interface to Body 1 [In-place 1]**
⌊ **Scalar Field**	**Function**	**Fluid Film Thickness**
⌊ **Color Bar**	**Orientation**	**Vertical**

Node	Property	Setting
	Position	`[0.05, 0.1]`

10. Save the simulation.

Running the Simulation

Preparation of the simulation is now complete, and the simulation can be run.

To run the simulation:

1. Click (**Run**) in the *Solution* toolbar.

   The solution progress is displayed in the *Output* window. The *Residuals* display is created automatically in the *Graphics* window and shows the progress of the solvers.

   While the simulation is running, you can click the tabs at the top of the *Graphics* window to view scenes and plots.

   During the run, you can stop the process by clicking (**Stop**) on the toolbar. If you do stop the simulation, you can click (**Run**) to resume it.

   When the last iteration is complete, the following message is displayed in the *Output* window:

   `Stopping criterion Maximum Physical Time satisfied.`

2. When the simulation has finished running, save it.

Visualizing the Results

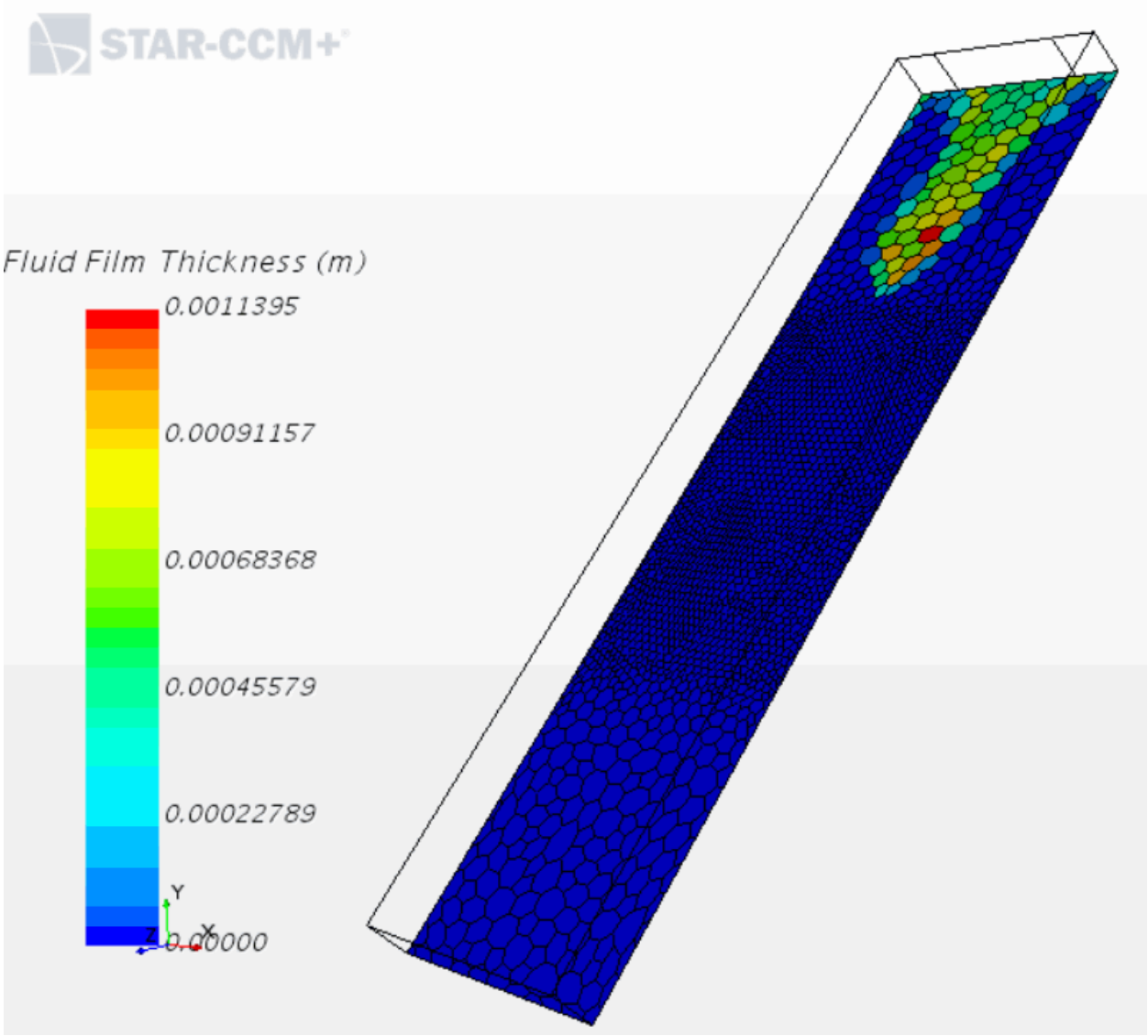
The water flowing down the inclined plane is a single physical situation. There is no physical change in the flow between the different mesh areas. However, the stream of water is modeled with two different models: the Fluid Film model in the coarse areas of the mesh (the upper and lower sections), and the VOF model in the middle area where the mesh is finer.

In order to obtain the most accurate description of the water flow, you need to plot the film using both models. A plot that uses only one model does not provide the best representation of the water flow.

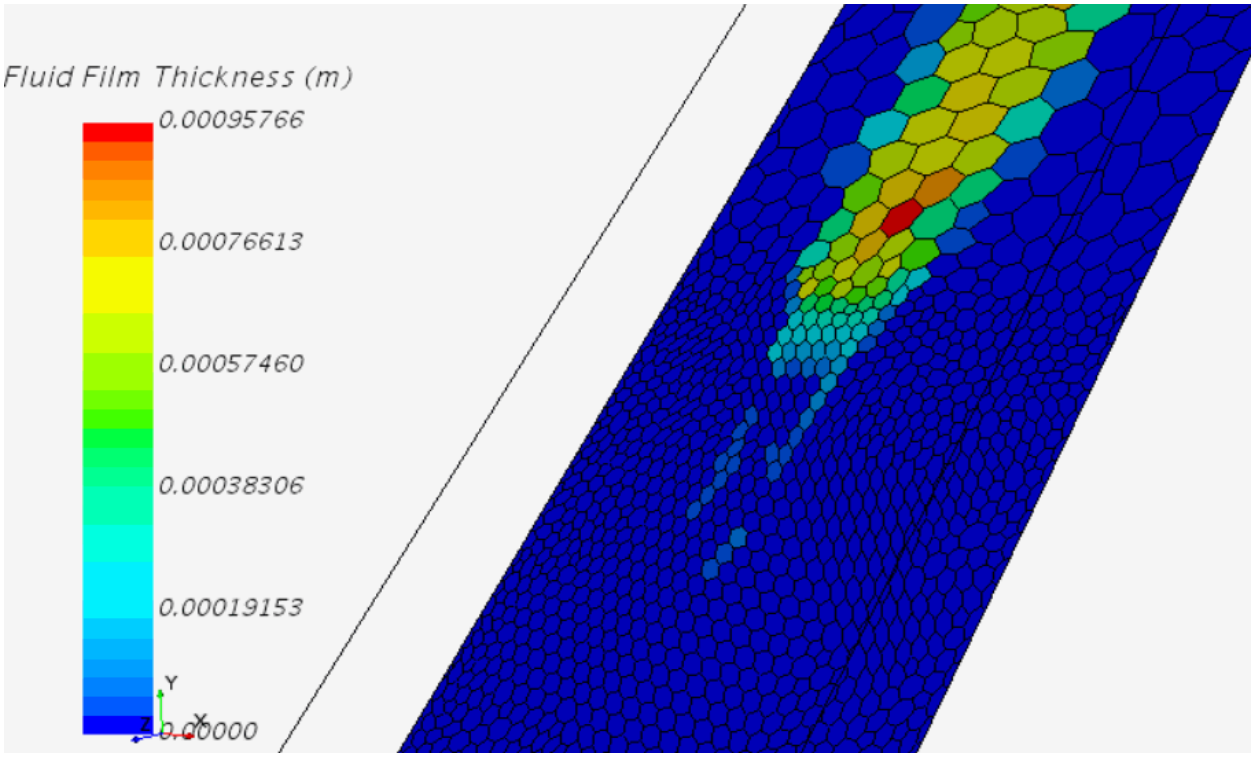
To visualize the results of the simulation:

1. Open the **Fluid Film Scene**.

   Water flows from the inlet and is modeled as a fluid film on the upper region of coarse mesh:



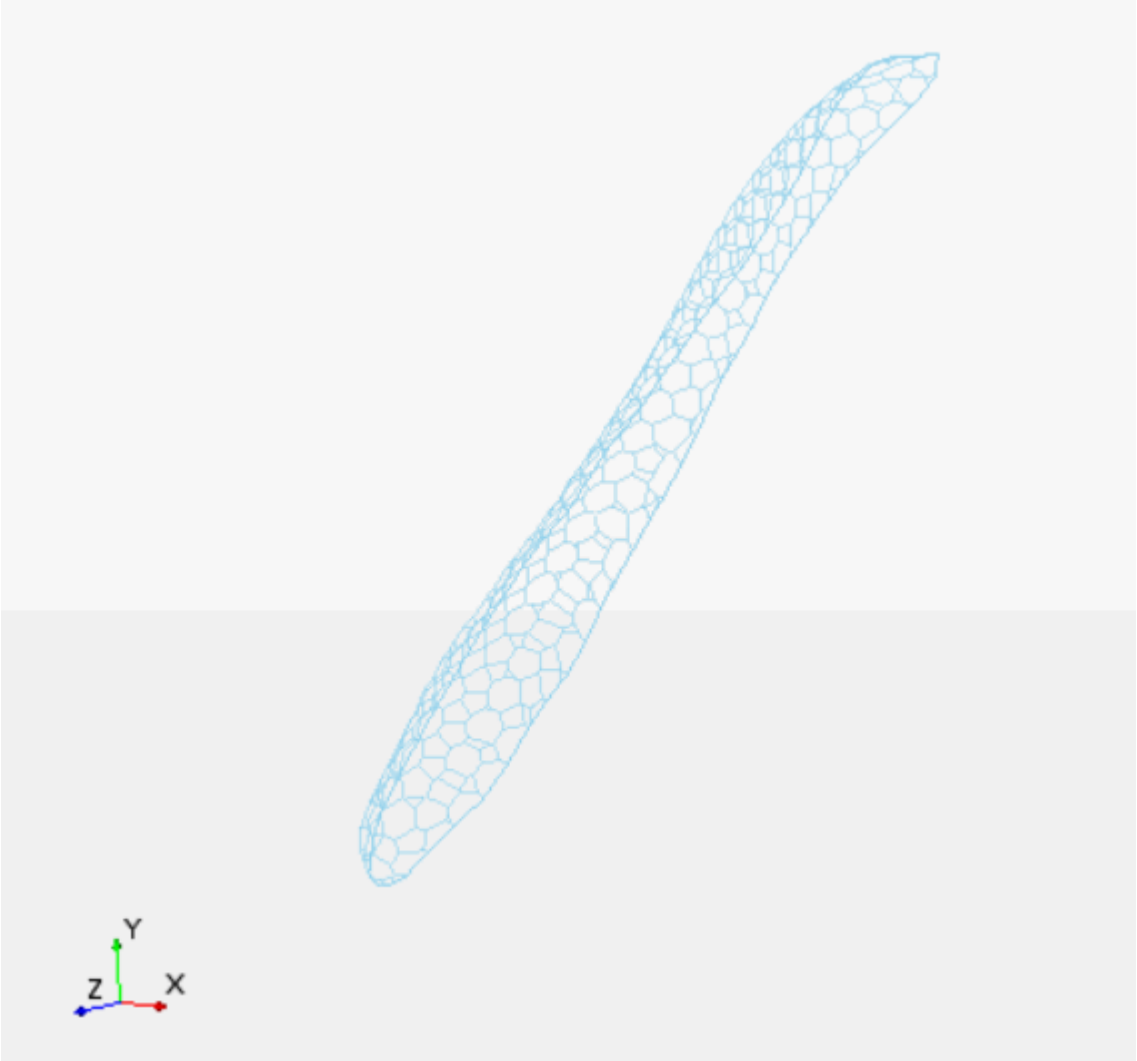
Where the mesh becomes finer, the water transitions from the fluid film phase to the VOF water phase.



To view the VOF water phase that is modeled in the fine mesh region:

2. Open the **Resolved Water Scene**.

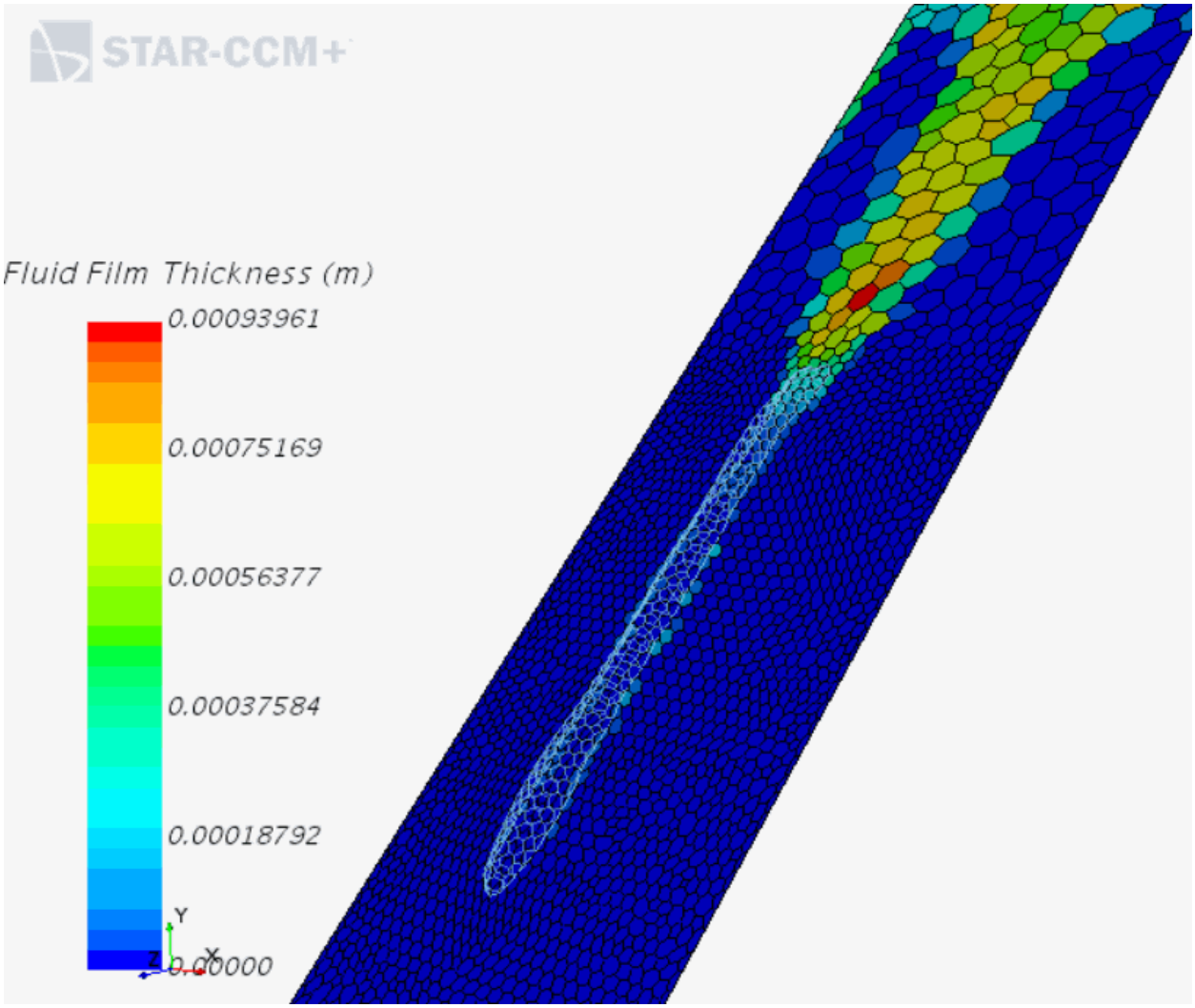
   The rivulet forms in the fine mesh region as water transitions from the fluid film phase to the VOF water phase.



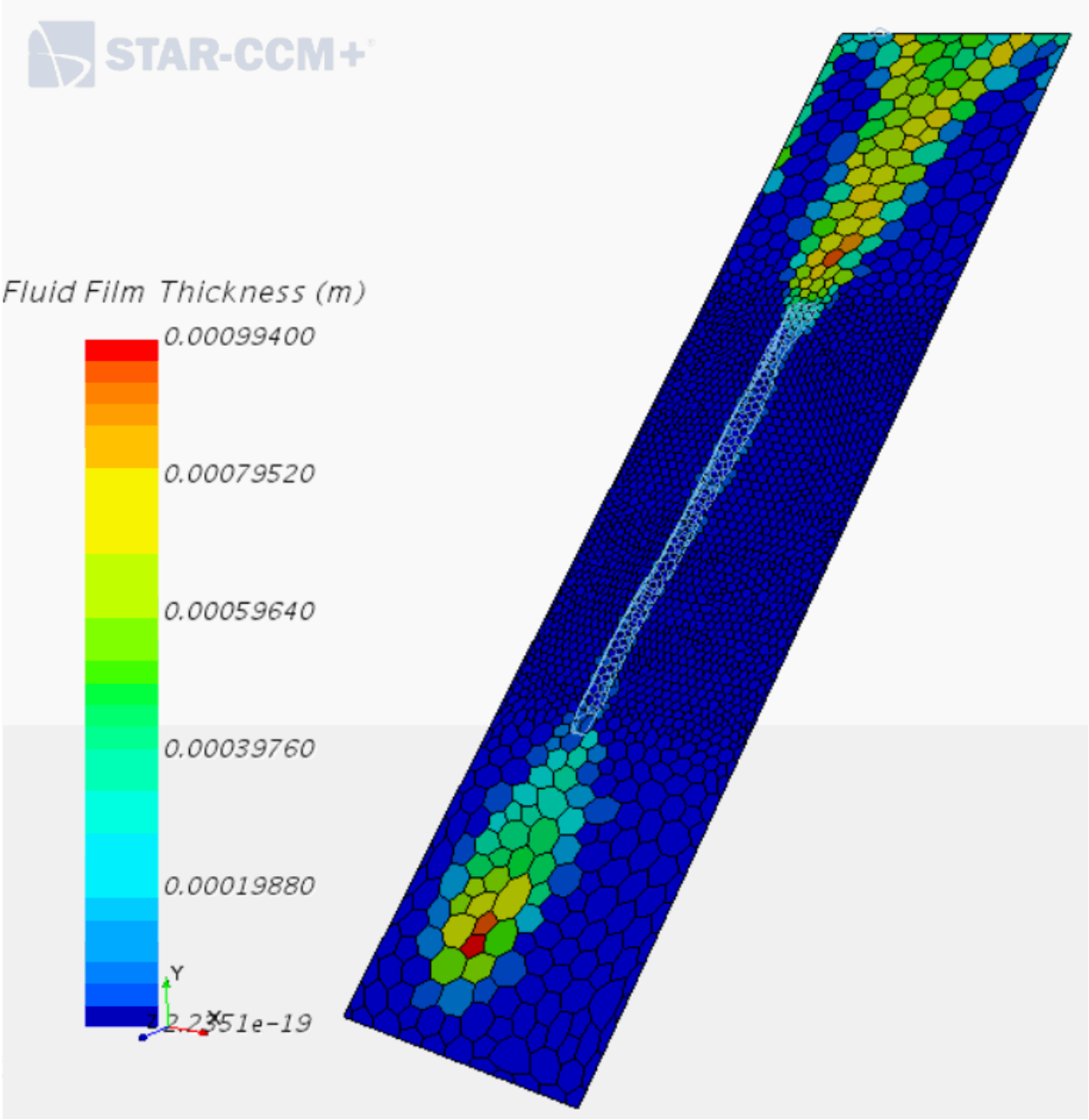
To view both the fluid film and VOF components:

3. Open the **Combined Scene**.

   The example below shows the flow of water when the simulation is in progress.



The example below shows the flow of water at the end of the simulation. When the flow reaches the lower region of coarse mesh, the water transitions from the VOF phase model to the fluid film model.



Summary

This tutorial demonstrated how to set up a hybrid Fluid Film-VOF simulation to model a water film flowing down an inclined plane.

This tutorial has introduced the following STAR-CCM+ features:

- Setting up the physics models.
- Creating phases and selecting the phase models.
- Defining the phase interaction between the VOF water phase and the fluid film phase.
- Setting the appropriate initial conditions, reference values, and boundary conditions.

- Setting the transition volume fraction at the interface.
- Setting up plots and scenes to visualize the results.
- Running the simulation.
- Viewing and interpreting the results.

Dispersed Multiphase: Airfoil Icing

When aircraft pass through air containing supercooled liquid water droplets, it is possible for the droplets to impact the aircraft surface and form layers of ice. Ice formation on an aircraft wing changes the aerodynamic properties of the wing, and leads to loss of lift, increase in drag, and a change in pressure distribution. STAR-CCM+ provides a technique for simulating the formation of ice on airfoils. This technique relies on the Dispersed Multiphase model to represent the liquid droplets in the air surrounding the airfoil.

The case simulated in this tutorial consists of a two-dimensional airfoil in an icing tunnel. The case is taken from a NASA study ([802]) on ice formation and icing effects for modern airfoils.

In this simulation, a multiphase mixture of air and water droplets flows over an idealized two-dimensional airfoil, in a low-temperature, low-turbulence regime. The mixture flows with initial velocity of 129 m/s in the x-direction. The droplets are water particles with diameter of 2.1E-5 m. The droplets volume fraction, with respect to air, is of 3.42E-7.

The geometry that is used in the tutorial is displayed below.

In the DMP method, the background air affects the water droplets through drag and heat transfer. The water droplets, however, do not have any impact on the air. The liquid content of the droplet phase is transferred to the surface of the airfoil as a liquid film. For this reason, a shell region with fluid film properties is created on the surface of the airfoil. Droplets impact the shell region and form a liquid film.

Liquid film on the airfoil surface turns to ice. As more and more ice is formed, the shape of the airfoil must change. STAR-CCM+ converts the volume of ice in the film to a corresponding deformation in the wing surface. After deforming the wing surface using the mesh morpher, STAR-CCM+ removes the volume of ice from the film layer. In this way, the shape of the shell region changes to reflect the buildup of ice on the airfoil surface.

Contents:

Prerequisites
Importing the Volume Mesh
Selecting Physics Models
Specifying Initial Conditions
Creating a Shell Region
Selecting Fluid Film Models

Tracking the Ice Layer
Selecting Dispersed Phase Models
Defining Phase Interactions
Setting Up the Morpher
Defining Boundary Conditions
Setting Up Monitors
Preparing the Scenes
Setting Stopping Criteria
Setting the Solver Parameters
Running the Simulation
Visualizing the Results
Summary
DMP: Airfoil Icing Tutorial Bibliography

Prerequisites

The instructions in the Dispersed Multiphase - Airfoil Icing tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Techniques	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

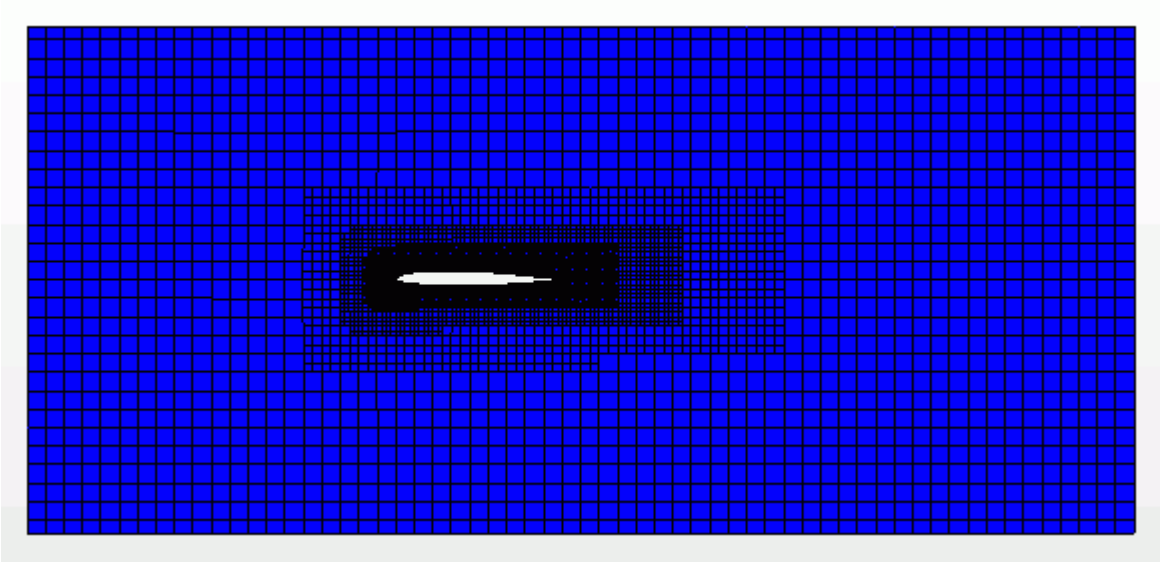
Importing the Volume Mesh

To set up the STAR-CCM+ simulation, launch a simulation and import the volume mesh for this tutorial.

1. Launch STAR-CCM+.
2. Start a new simulation.
3. Select **File** > **Import** > **Import Volume Mesh...**
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/multiphaseFlow/data`.
5. Select `dmpAirfoil.ccm`, then click **Open**.

   STAR-CCM+ provides feedback on the import process in the *Output* window, and a mesh region is created under the **Regions** node.
6. Save the simulation as `dmpAirfoil.sim`.

   You can visualize the imported volume mesh with a method of your choice, such as creating a mesh scene. The mesh appears as shown below.



Selecting Physics Models

In the DMP approach, single-phase flow models are used to simulate the continuous gas phase, which in this tutorial is air. The dispersed phase is water droplets.

By default, DMP is one-way coupled: the background air flow can influence the dispersed liquid phase (for example, through drag force and heat transfer) but not the reverse (that is, the air flow affects the water droplets, but the droplets have no effect on the air). The only way in which the dispersed phase can influence the physics continuum is through impingement onto a liquid film.

Two-way coupling is optional in a DMP simulation. When two-way coupling is activated, the dispersed phases can influence the physics continuum. Drag force and heat transfer contribute to all phases, and the volume of the dispersed phases is subtracted from the available volume for the background phase. You activate two-way coupling when you expect a dispersed phase to have a significant effect on the continuous phase. However, in this case, the influence of the dispersed phase on the continuous background phase is assumed to be insignificant and is neglected.

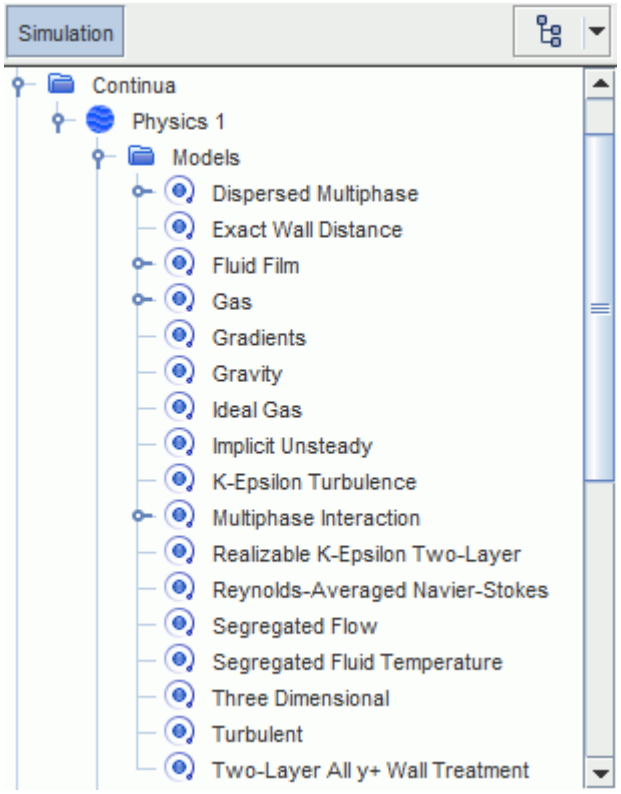
At this stage, select the single-phase models that represent the unsteady air flow as an ideal, turbulent gas, subject to gravity. Choose the Dispersed Multiphase model to solve for the water phase as droplets, and the Fluid Film to solve for the water phase on the surface of the airfoil. You also require the Multiphase Interaction model, to define the influence of the gas phase on the droplet motion and energy, and the impingement of the droplets onto the water film phase.

1. For the physics continuum, **Continua** > **Physics 1**, select the following models in order:

Group box	Model
Enabled Models	**Three Dimensional** (Selected automatically)
Time	**Implicit Unsteady**
Material	**Gas**
Flow	**Segregated Flow**

Group box	Model
Enabled Models	**Gradients** (Selected automatically)
Equation of State	**Ideal Gas**
Energy	**Segregated Fluid Temperature**
Viscous Regime	**Turbulent**
Enabled Models	**Reynolds-Averaged Navier-Stokes** (Selected automatically)
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**
Enabled Models	**Exact Wall Distance** (Selected automatically)
	Realizable K-Epsilon Two-Layer (Selected automatically)
	Two-Layer All y+ Wall Treatment (Selected automatically)
Optional Models	**Fluid Film**
	Dispersed Multiphase
	Multiphase Interaction
	Gravity

2. Click **Close**.
3. To review the models, open the **Physics 1** > **Models** node.



4. Save the simulation.

Specifying Initial Conditions

By default, gravity is set to act along the z-axis, whereas the geometry for this tutorial requires it to act along the y-axis. The simulation also requires initial values for fluid velocity and temperature.

To specify the initial conditions:

1. Edit the **Continua** > **Physics 1** node, and set the following properties:

Node	Property	Setting
Reference Values		
└ **Gravity**	*Value*	`[0,-9.81,0] m/s^2`
Initial Conditions		
└ **Static Temperature**	*Value*	`258.4 K`
└ **Velocity**	*Value*	`[129,0,0] m/s`

Creating a Shell Region

Define a shell region on the surface of the airfoil.

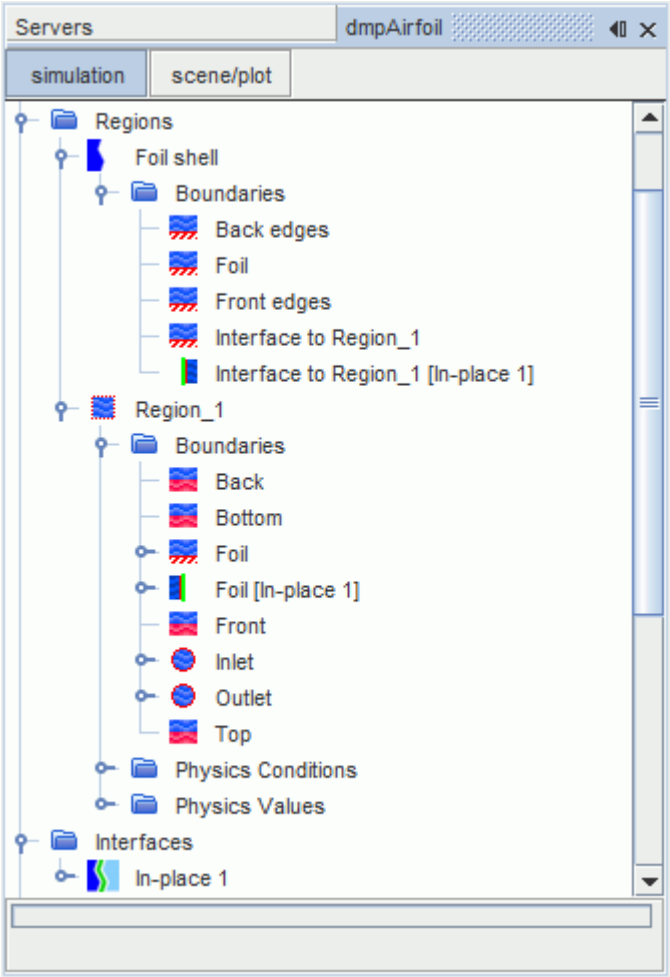
The DMP model permits mass within its liquid phase to transfer to a liquid film. In STAR-CCM+, liquid films exist only within a shell region. A shell region is a cell layer that is located on top of an existing boundary. Although the cell layer has zero-thickness in geometry, it contains a finite thickness relevant to the quantity being modeled. In the case of a liquid film, the thickness is the depth of the film layer.

In this tutorial, you define a shell region on the surface of the airfoil, so that the liquid film associated with the shell region can receive impinging droplets.

To create a shell region:

1. Right-click the **Regions** > **Region_1** > **Boundaries** > **Foil** node and select **Create Shell Region**.

   A shell region, **Foil shell**, is created within the **Regions** node. Also, an interface between the shell region and **Region 1** is created automatically within the **Interfaces** node.



2. Save the simulation.

Selecting Fluid Film Models

The set of models that solve for the liquid film phase is specified within the Fluid Film model node. Here, you first define a fluid film phase and select appropriate physics models. You then assign the film phase to the shell region on the surface of the airfoil.

To define a fluid film phase:

1. Expand the **Continua** > **Physics 1** > **Models** > **Fluid Film** node.
2. Right-click the **Fluid Film Phases** node and select **New**.

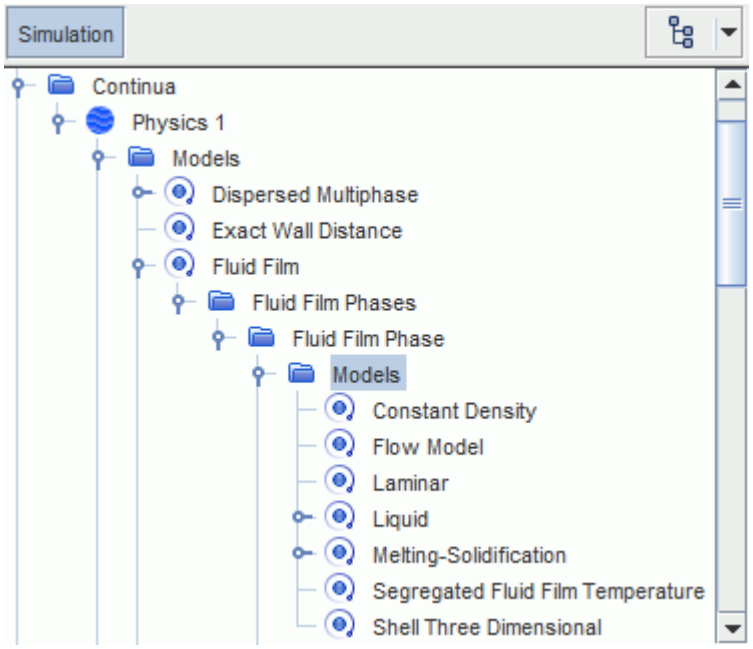
   A new node, **Phase 1**, is created within the **Fluid Film Phases** node.
3. Rename **Phase 1** to `Film Phase`.

Select the models for the film phase. Within this selection, activate the Melting-Solidification model that captures the solidification of water droplets into ice:

4. For the fluid film phase, **Film Phase**, select the following physics models:

Group box	Model
Space	**Shell Three Dimensional***
Enabled Models	**Segregated Fluid Film Temperature** (selected automatically)
	Laminar (selected automatically)
	Flow Model (selected automatically)
Material	**Liquid**
Equation of State	**Constant Density**
Optional Models	**Melting-Solidification**

5. To review the models, expand the **Fluid Film Phases** > **Film Phase** > **Models** node.



As an initial condition for the fluid film phase, specify the temperature:

6. Expand the **Film Phase** > **Initial Conditions** node.
7. Select the **Static Temperature** node and set *Value* to `258.4 K`.

Tracking the Ice Layer

In an ice-accretion simulation, STAR-CCM+ calculates the thickness of solidified film (ice) at every time-step. At the start of the next time-step, the solid film thickness increment is passed to the mesh morpher so that it can deform the surface of the airfoil to match the film thickness. The solidified film mass is then removed from the simulation.

To allow the Melting-Solidification model to track the liquid-solid interface, you must set its Solidified Film Option to Remove Solidified Film.

To set the properties for removing the solidified film:

1. Expand the **Continua** > **Physics 1** > **Models** > **Fluid Film** > **Fluid Film Phases** > **Film Phase** > **Models** node.
2. Set the following properties:

Node	Property	Setting
Flow Model	*Maximum Film Thickness*	`1.0E-5 m`
Melting-Solidification	*Solidified Film Option*	**Remove Solidified Film**
	Solid Time-step Factor	`10`

The *Solid Time-step Factor* allows you to decrease the time that is required to run the simulation. A *Solid Time-step Factor* of 10, allows ice to form 10 times faster in the simulation.

The ice accumulation, visible as deformation of the wing shape, corresponds to an exposure time given by the product of the *Solid Time-step Factor* and the simulated physical time.

Finally, assign the fluid film phase models to the shell region, **Foil shell**:

3. Select the **Regions** > **Foil shell** node and set *Continuum* to **Film Phase**.
4. Save the simulation.

Selecting Dispersed Phase Models

A dispersed phase is defined within the Dispersed Multiphase model node as an Eulerian phase.

By default, each dispersed phase is one-way coupled: the influence of the dispersed phase on the continuous background phase is assumed to be insignificant and is neglected. If you expect a dispersed phase to have a significant effect on the continuous phase, you can activate two-way coupling. However, one-way coupling is sufficient for this simulation.

To create a dispersed phase:

1. Expand the **Continua** > **Physics 1** > **Models** > **Dispersed Multiphase** node.
2. Right-click the **Eulerian Phases** node, and select **New**.

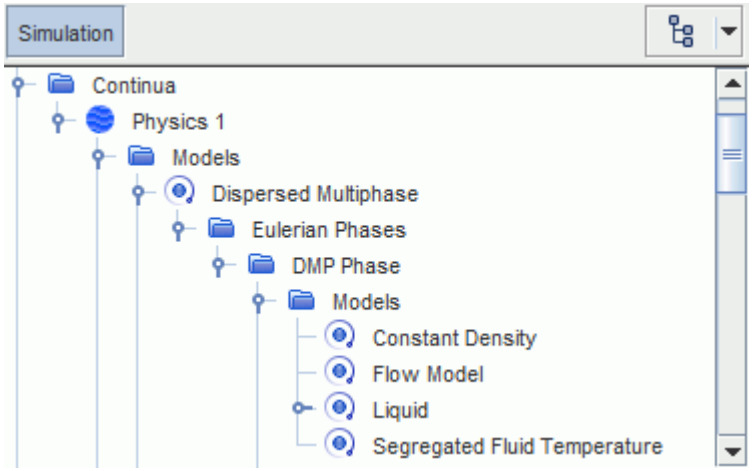
   A new node, **Phase 1**, is added within the **Eulerian Phases** node.
3. Rename the **Phase 1** node to `DMP Phase`.

Select the physics models for the dispersed phase:

4. For the dispersed phase, **DMP Phase**, select the following physics models:

Group box	Model
Material	**Liquid** **Constant Density** (selected automatically)
Optional Models	**Flow Model** **Segregated Fluid Temperature** (selected automatically)

5. To review the selected models, expand the **DMP Phase** > **Models** node.



For the dispersed phase, specify the droplets diameter and set their initial temperature and velocity to match the initial values that are set for the background fluid.

6. Expand the **DMP Phase** > **Models** > **Liquid** > **H2O** > **Material Properties** node.
7. Select the **Particle Diameter** > **Constant** node, and set *Value* to `2.1E-5 m`.
8. Edit the **DMP Phase** > **Initial Conditions** node, and set the following properties:

Node	Property	Setting
Static Temperature	*Value*	`258.4 K`
Velocity	*Value*	`[129, 0, 0] m/s`

9. Save the simulation.

Defining Phase Interactions

Interactions between phases are specified within the Multiphase Interaction Model node. In this case, the continuous gas phase interacts with the dispersed phase through drag force and heat transfer. The dispersed phase interacts with the fluid film phase through impingement.

To specify the interaction between the dispersed phase and the fluid film phase:

1. Expand the **Continua** > **Physics 1** > **Models** > **Multiphase Interaction** node.
2. Right-click the **Phase Interactions** node and select **New**.

   A new node, **Phase Interaction 1**, appears within the **Phase Interactions** node.
3. Rename **Phase Interaction 1** to `Film-DMP`.
4. For the phase interaction, **Film-DMP**, select the following physics models:

Group box	Model
Phase Interaction Topology	**DMP-Film Interaction**
Optional Models	**Impingement**

5. Select the **Film-DMP** > **Models** > **DMP-Film Interaction** node, and set the following properties:

Property	Setting
Fluid Film Phase	**Film Phase**
Dispersed Phase	**DMP Phase**

To specify the interaction between the dispersed phase and the continuous phase:

6. Repeat the steps to create a second phase interaction. Name this second interaction, `DMP-Continuum`.
7. For the phase interaction, **DMP-Continuum**, select the following physics models:

Group box	Model
Phase Interaction Topology	**DMP-Physics Continuum Interaction**
Optional Models	**Interaction Length Scale**
	Heat Transfer
	Drag Force

8. Select the **DMP-Continuum** > **Models** > **DMP-Physics Continuum Interaction** node, and set *Dispersed Phase* to **DMP Phase**.
9. Save the simulation.

Setting Up the Morpher

This case requires the morpher to apply mesh deformation for ice accretion. In STAR-CCM+, the morpher is defined as a motion.

To activate the morpher:

1. Right-click the **Tools** > **Motions** node and select **New** > **Morphing**.
2. Select the **Regions** > **Region_1** > **Physics Values** > **Motion Specification** node, and set *Motion* to **Morphing**.

Assign the morphing motion to the airfoil surface:

3. Expand the **Regions** > **Region_1** > **Boundaries** > **Foil [In-place 1]** > **Physics Conditions** node.
4. Select the **Morpher Specification** node and set *Specification* to **Displacement**.
5. Select the **Morpher Displacement Specification** node and set *Specification* to **Internal**.

   When the morpher method is set to Internal, the displacement fed to the morpher is computed by STAR-CCM+ internal models. This method is used by the Melting-Solidification model, that provides the solid film thickness increment to the morpher.

   Motion Specification for the **Foil shell** region is automatically set to follow **Region_1**.

Defining Boundary Conditions

Specify the temperature, the volume fraction of the dispersed phase, and the Mach number at the inlet and outlet.

To define the boundary conditions:

1. Expand the **Regions** > **Region_1** > **Boundaries** > **Inlet** node.
2. Edit the **Physics Values** node, and set the following properties:

Node	Property	Setting
Mach Number	*Value*	`0.4`
Static Temperature	*Value*	`258.4 K`

3. Edit the **Phase Conditions** > **DMP Phase** > **Physics Values** node, and set the following properties:

Node	Property	Setting
Volume Fraction	*Value*	`3.42E-7`

4. Repeat the previous steps to set the boundary conditions for the **Outlet**, using the same settings that you entered for the **Inlet**.

Specify boundary conditions at the interface:

5. Edit the **Regions** > **Region_1** > **Boundaries** > **Foil [In-place 1]** node, and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Wall Surface Specification**	*Method*	**Rough**
Physics Values		
└ **Roughness Height**	*Value*	`0.1 mm`

6. Save the simulation.

Setting Up Monitors

Monitor key quantities like the film thickness, the maximum film thickness, and the film surface area, as functions of time.

To set up the reports and monitors:

1. Create a **Sum** report with the following settings:

Property	Setting
Scalar Field Function	**Area: Magnitude**
Parts	**Foil shell** > **Boundaries** > **Foil**

2. Rename the **Sum 1** node to `Film Surface`.
3. Create a **Surface integral** report with the following settings:

Property	Setting
Scalar Field Function	**Fluid Film Thickness**

Property	Setting
Parts	**Foil shell** > **Boundaries** > **Foil**

4. Rename the **Surface integral 1** node to `Film Volume`.
5. Create a **Maximum** report with the following settings:

Property	Setting
Scalar Field Function	**Fluid Film Thickness**
Parts	**Foil shell** > **Boundaries** > **Foil**

6. Rename the **Maximum 1** node to `Maximum Film Thickness`.
7. Right-click the **Film Surface** node and select **Create Monitor and Plot from Report**.
8. Repeat this step for all the reports that you created.

Preparing the Scenes

Here you prepare scalar scenes to visualize the impingement rate, the volume fraction of the dispersed phase, and the total solid film thickness increment. You also create a vector scene to visualize the morpher displacement.

To visualize the impingement rate:

1. Create a scalar scene and rename it to `Impingement Rate`.
2. Expand the **Scenes** > **Impingement Rate** > **Displayers** node.
3. Edit the **Outline 1** > **Parts** node, and remove **Region_1** from the selected Parts.
4. For the node, **Scalar 1**, set the following properties:

Node	Property	Setting
Parts	*Parts*	**Region_1** > **Boundaries** > **Foil [In-place 1]**
Scalar Field	*Function*	**Film Impingement Rate** > **Film-Dispersed Impingement Rate of Film-DMP**

To visualize the volume fraction of the dispersed phase:

5. Create a scalar scene and rename it to `DMP Volume Fraction`.
6. Edit the **Scenes** > **DMP Volume Fraction** > **Displayers** node, and set the following properties:

Node	Property	Setting
Scalar 1 > **Parts**	*Parts*	**Region_1**
Scalar 1 > **Scalar Field**	*Function*	**Volume Fraction** > **Volume Fraction of DMP Phase**

The solid film thickness increment is computed at every time step. To visualize the total solid film thickness increment, accumulated during the total simulation time, you need to create a Field Sum monitor for the **Solid Film Thickness Increment** scalar function.

7. Within the **Monitors** node, create a new **Field Sum** monitor with the following settings:

Property	Setting
Field Function	**Solid Film Thickness Increment**
Parts	**Foil shell** > **Boundaries** > **Interface to Region_1 [In-place 1]**

8. Rename the Field Sum 1 monitor to `Total Solid Film Thickness Increment`.
9. Create a scalar scene and rename it to `Total Solid Film Thickness Increment`.
10. Expand the **Scenes** > **Total Solid Film Thickness Increment** > **Displayers** node.
11. Edit the **Outline 1** > **Parts** node, and remove **Region_1** from the selected Parts.
12. For the node, **Scalar 1**, set the following properties:

Node	Property	Setting
Parts	*Parts*	**Foil shell** > **Boundaries** > **Interface to Region_1 [In-place 1]**
Scalar Field	*Function*	**Sum of Solid Film Thickness Increment**

To visualize the morpher displacement:

13. Create a vector scene and rename it to `Morpher Displacement`.
14. Expand the **Scenes** > **Morpher Displacement** > **Displayers** node.
15. Select the **Outline 1** node and activate *Surface*.
16. Edit the **Outline 1** > **Parts** node, deselect all Parts, and then select **Region_1** > **Boundaries** > **Foil [In-place 1]**.
17. For the node, **Vector 1**, set the following properties:

Node	Property	Setting
Parts	*Parts*	**Region_1** > **Boundaries** > **Foil [In-place 1]**
Vector Field	*Function*	**Morpher Displacement**

18. Save the simulation.

Setting Stopping Criteria

Specify the maximum number of iterations and the maximum physical time.

To set the stopping criteria for this simulation:

1. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Inner Iterations	*Maximum Inner Iterations*	`20`
Maximum Physical Time	*Maximum Physical Time*	`34.2 s`

Setting the Solver Parameters

Set appropriate parameters for the Dispersed Phase Flow solver and the Implicit Unsteady solver.

To set the solver parameters:

1. Edit the **Solvers** > **Dispersed Multiphase** > **Dispersed Phase Flow** node.
2. Set the following properties:

Node	Property	Setting
Velocity	*Under-Relaxation Factor*	`0.2`
Volume Fraction	*Maximum Volume Fraction*	`1.0E-4`
	Minimum Volume Fraction	`1.0E-8`

3. Select the **Solvers** > **Implicit Unsteady** node and set *Time-Step* to `0.1 s`.
4. Save the simulation.

Running the Simulation

The simulation is now fully set up and ready to be run.

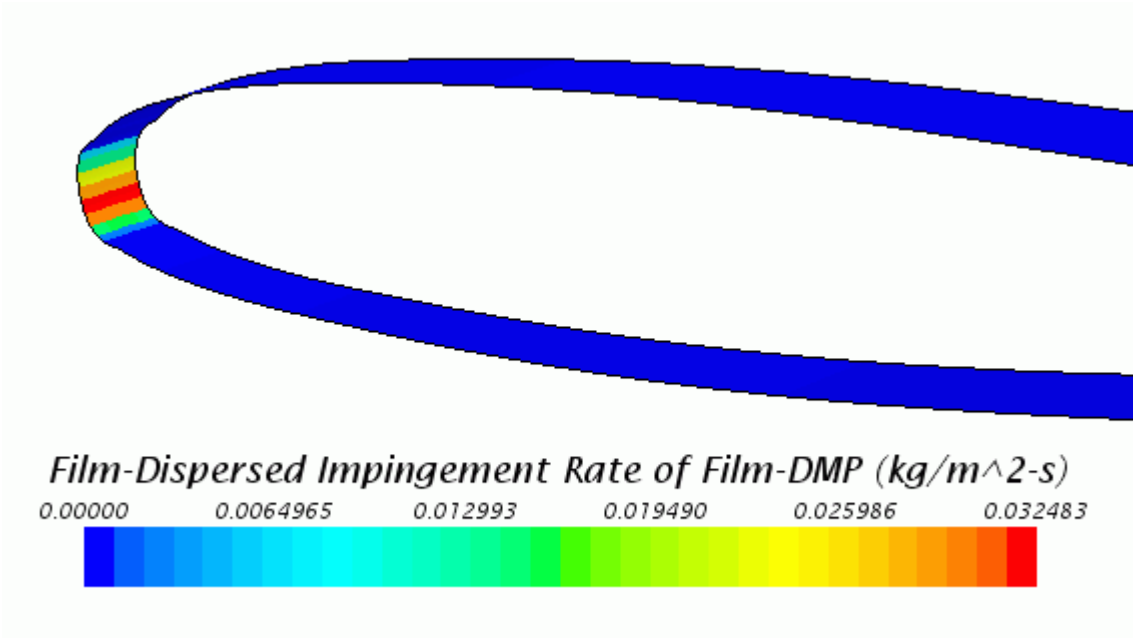
To run the simulation:

1. Click (**Initialize Solution**).
2. Click (**Run**).
3. When the simulation has finished running, save the simulation.

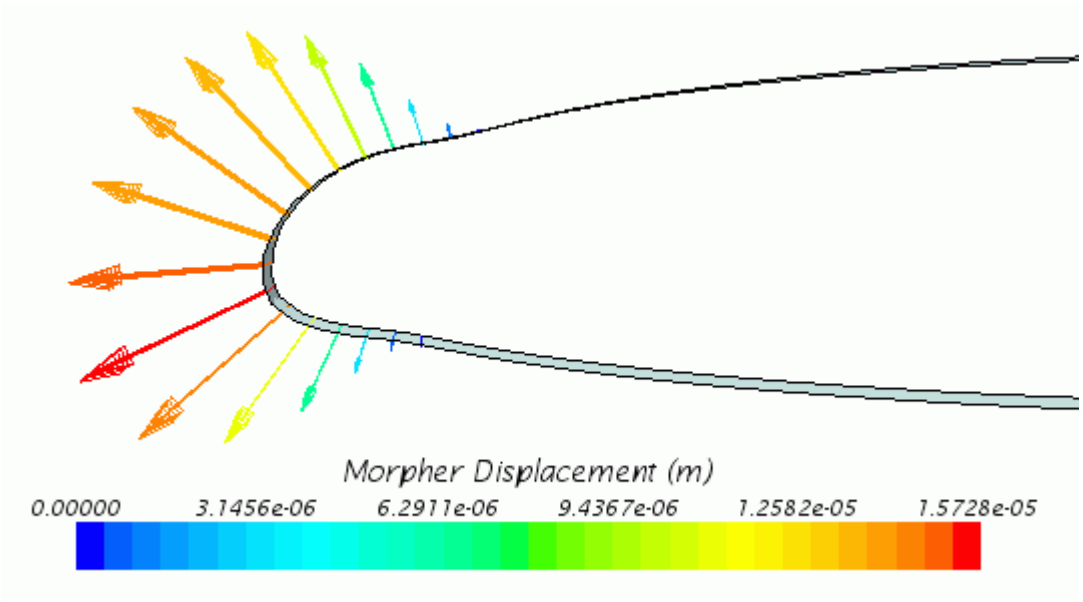
Visualizing the Results

Examine the scenes that you prepared earlier.

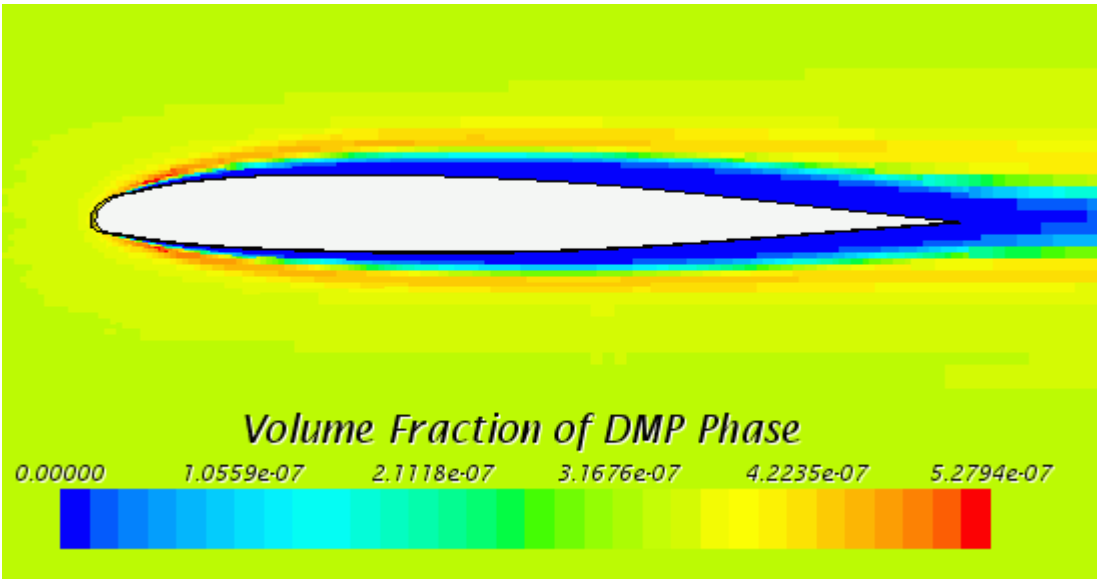
The Impingement Rate scene shows the rate at which water droplets hit the surface of the airfoil:



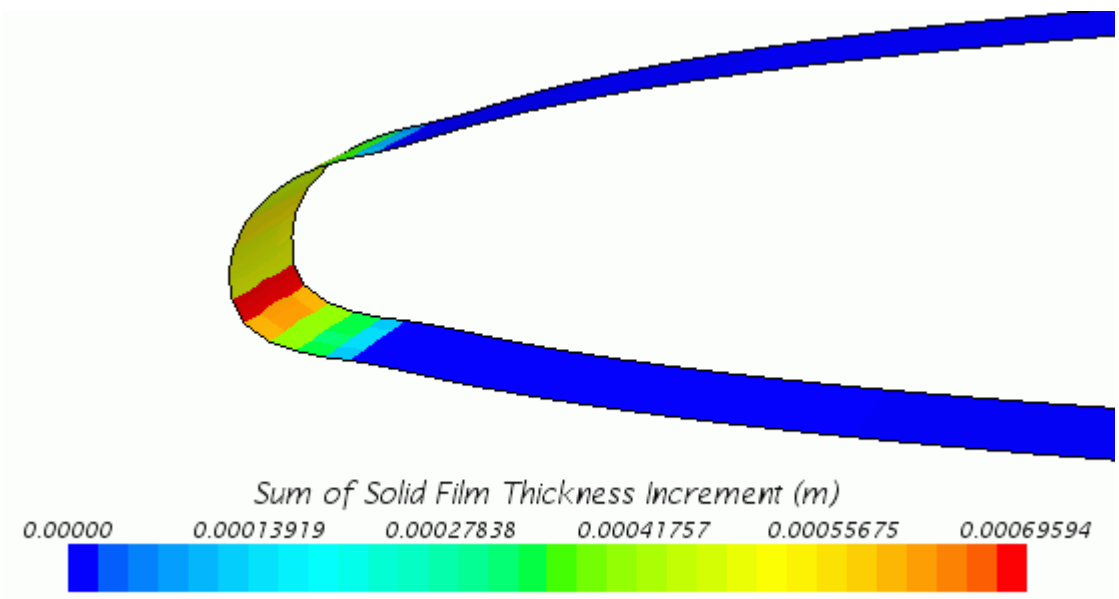
The amount by which the morpher displaces the airfoil surface in each time step is shown in the Morpher Displacement scene:



The concentration of water droplets in air is shown in the DMP Volume Fraction scene:



The total amount of ice accumulated on the airfoil surface is shown in the scene for Total Solid Film Thickness Increment:



Summary

This tutorial demonstrated how to simulate ice formation on two-dimensional airfoils in STAR-CCM+. In particular, the simulation focuses on the ice build-up process, showing the effects on the shape of the wing. Since changes in the wing shape affect the lift and drag forces, it is straightforward to investigate the effects of the accumulated ice on aerodynamic performance.

This tutorial has introduced the following STAR-CCM+ features:

- Selecting the continuum physics models in the DMP approach
- Setting up a fluid shell region as particles impact boundary surface
- Using the DMP model to define a dispersed phase
- Defining interactions between phases
- Using the Melting-Solidification model in combination with the morpher

DMP: Airfoil Icing Tutorial Bibliography

[802] *Addy Jr, H. E. "Ice Accretions and Icing Effects for Modern Airfoils", NASA/TP—2000-210031*

Discrete Element Method

The tutorials in this set illustrate various STAR-CCM+ features for simulating Discrete Element Method problems

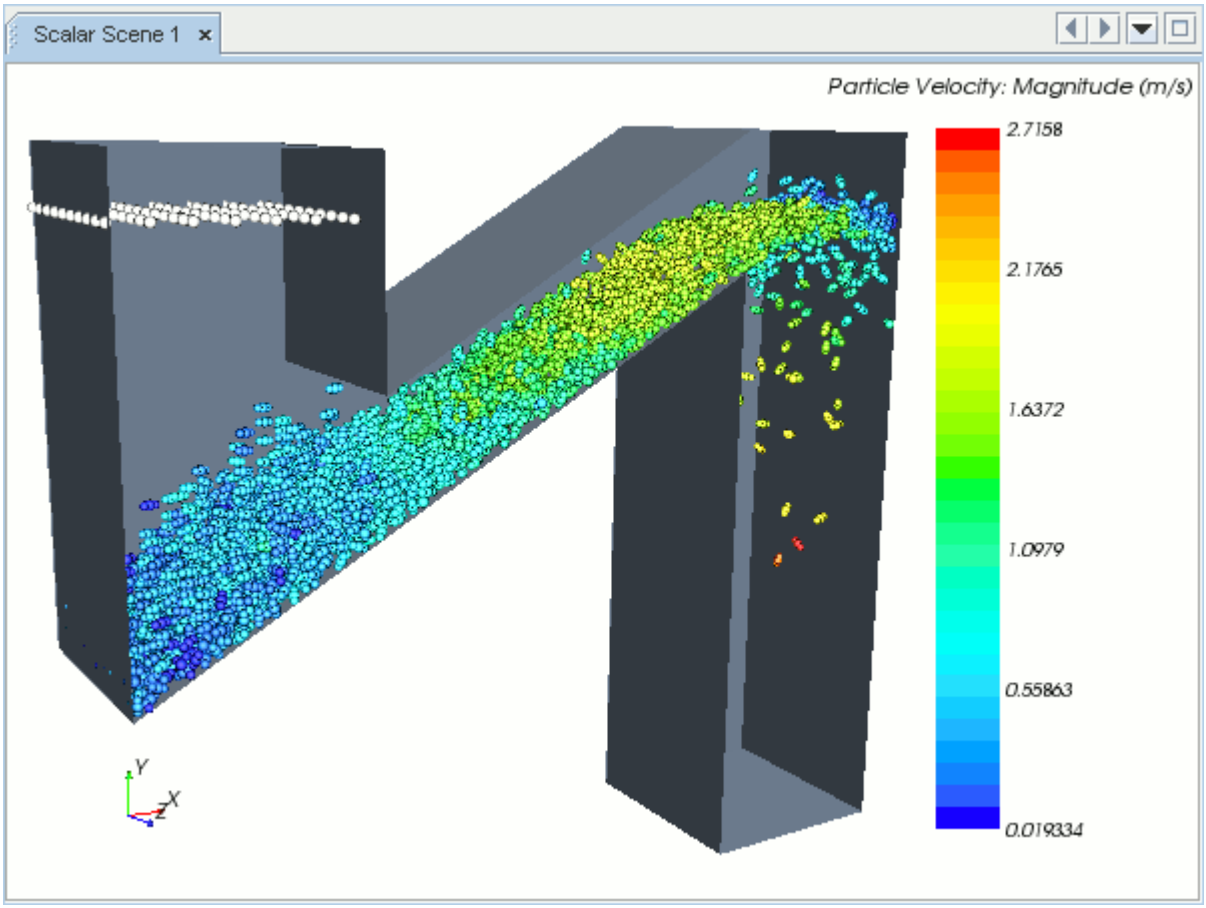
Contents:

DEM Particles in a Conveyor
DEM Particle Settling
Cylindrical Particles in a Rotating Drum

DEM Particles in a Conveyor

The purpose of this tutorial is to provide a guide to setting up a simulation using the Discrete Element Method.

The simulation consists of an injection chute, conveyor belt, and collection hopper. Grains of wheat are injected from points in the injection chute for 0.5 seconds. The conveyor then carries the grain up a slope, where it falls into the collection hopper. A visual representation of the transient solution is created, which shows the grain moving through the fluid region. The color of each piece of grain represents the velocity of that particle.



Contents:

Prerequisites
Importing the Surface Mesh and Naming the Simulation
Visualizing the Imported Geometry and Surface Mesh
Generating the Volume Mesh
Selecting Physics Models
Defining Lagrangian Phases
Defining the DEM Particle Interaction

Setting Reference Values
Setting the Moving Conveyor Wall Condition
Creating an Injector
Setting the Solver Parameters and Stopping Criteria
Setting Up a Visualization Scene
Running the Simulation
Visualizing Results
Summary

Prerequisites

The instructions in this tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

To complete this tutorial, familiarize yourself with the following:

Techniques	Associated Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Importing the Surface Mesh and Naming the Simulation

Create a new simulation and import the mesh from `grainConveyor.x_b`.

Navigate to the `doc/startutorialsdata/discreteElementMethod/data` subdirectory of the STAR-CCM+ installation directory. Copy the `grainConveyor.x_b` file to the current working directory.

Start upSTAR-CCM+ in the usual manner and select the New Simulation option from the menu bar.

Continue by importing the fluid surface mesh and saving the simulation.

1. Select **File** > **Import** > **Import Surface Mesh** from the menu bar. In the *Open* dialog, navigate to the working directory and select the file `grainConveyor.x_b`. Click **Open**.
2. In the *Import Surface Options* dialog:
   a) Make sure that **Create New Part** is activated.
   b) Maintain the remaining settings and click **OK**.

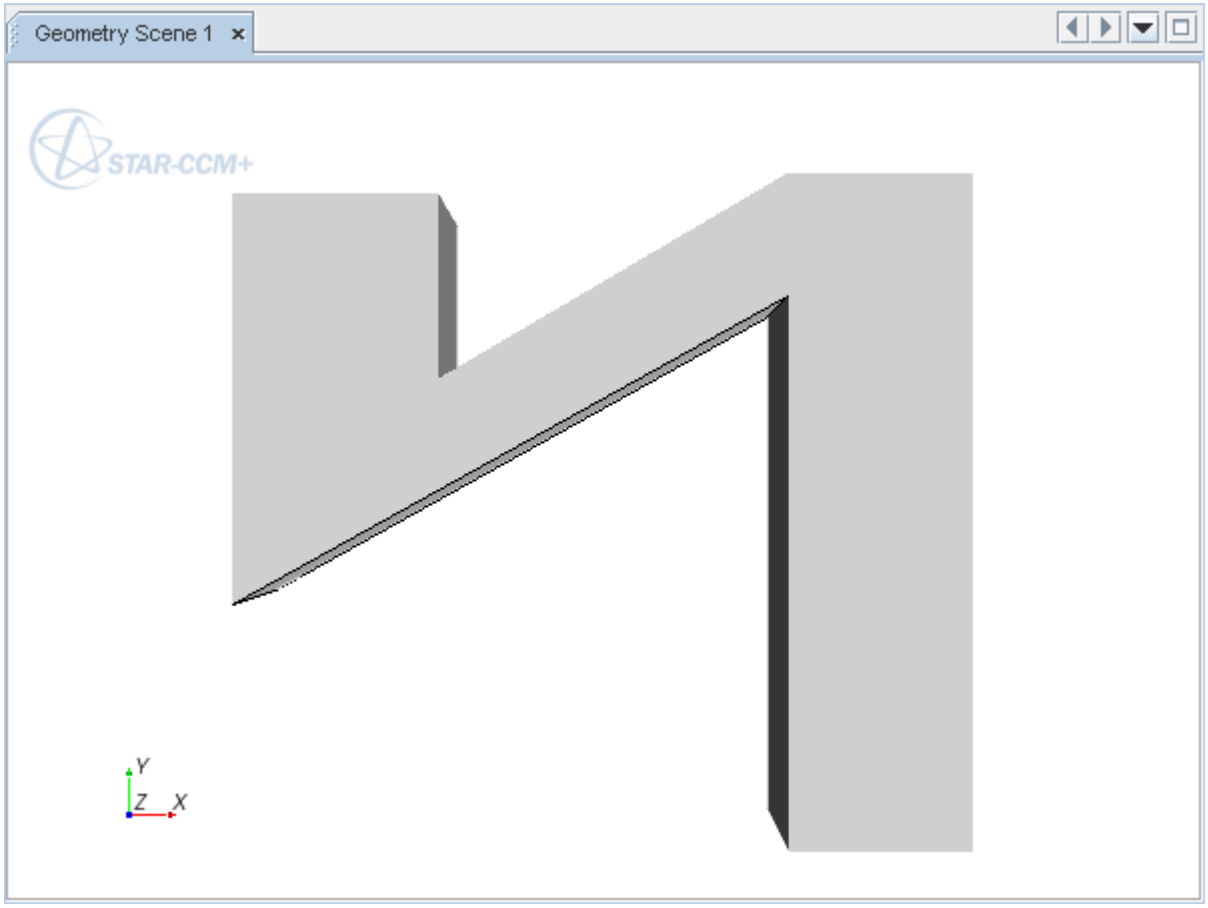
   STAR-CCM+ provides feedback on the import process, which takes a few seconds, in the *Output* window. A geometry scene is created in the *Graphics* window.
3. Save the new simulation to disk under file name `tutorialDEM.sim`.

Visualizing the Imported Geometry and Surface Mesh

Examine the *Geometry Scene 1* display in the *Graphics* window.

Use the following steps:

1. Use the mouse buttons to rotate, zoom in, and position the part to obtain a similar view as shown below.



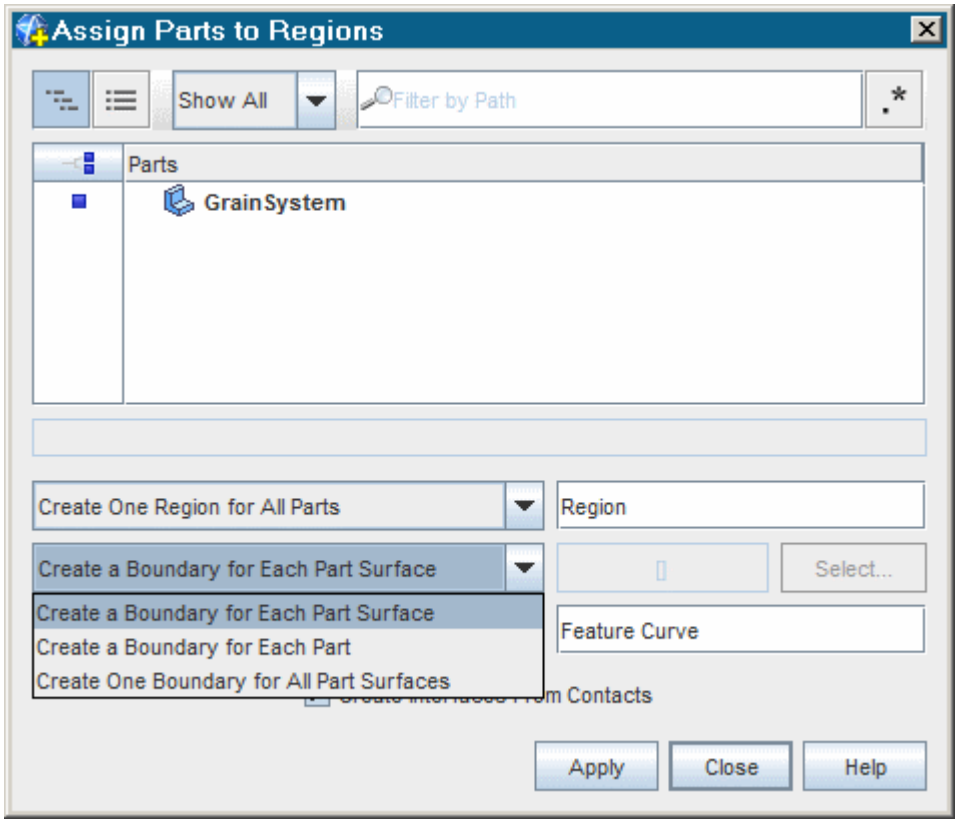
2. Save this view by clicking (**Save-Restore-Select views**) on the toolbar and choosing **Store Current View**.

Generating the Volume Mesh

The first step in generating a volume mesh is to create a fluid region from the imported geometry part.

Use the following steps:

1. Right-click the **Geometry** > **Parts** > **GrainSystem** node, and select **Assign Parts to Regions...**
2. In the *Assign Parts to Regions* dialog, make sure that the **GrainSystem** part is selected, and set the second dropdown to **Create a Boundary for Each Part Surface**.



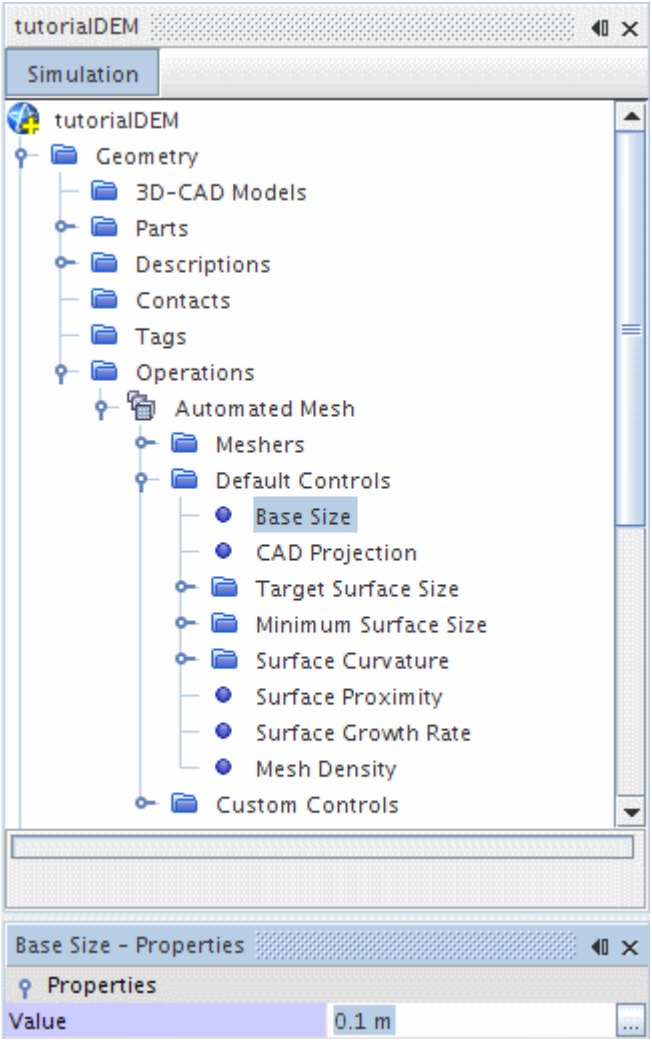
3. Click **Apply**, and then **Close**.

To create a meshing continuum with a base size of 0.1m:

4. Right-click **Geometry** > **Operations** and select **New** > **Mesh** > **Automated Mesh**.
5. In the *Create Automated Mesh Operation* dialog:
   a) In the *Parts* list, select **GrainSystem**
   b) In the *Select Meshers* box, select the following meshers:

Group	Mesher
Surface Meshers	**Surface Remesher**
Core Volume Meshers	**Polyhedral Mesher**

6. Select the **Automated Mesh** > **Default Controls** > **Base Size** node and set *Value* to 0.1 m.



7. Click (**Generate Volume Mesh**) in the **Mesh** toolbar.

   The output from the meshers is shown in the *Output* window. When the meshing process is complete, the message `Volume Meshing Pipeline Completed` is displayed in the *Output* window.

8. Select **Scenes** > **Geometry Scene 1** > **Displayers** > **Geometry 1** and activate *Mesh*.
9. In the *Geometry Scene*, right-click and select **Apply Representation** > **Volume Mesh** to see the result.

10. Save the simulation.

Selecting Physics Models

Physics models are required to define the behavior of the multiphase continuum.

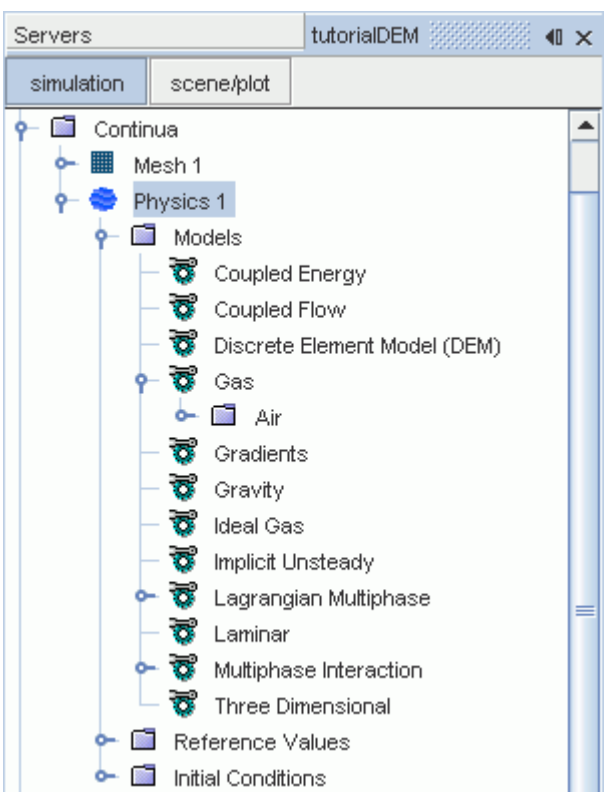
To set the physics models:

1. For the physics continuum **Continua** > **Physics 1**, select the following models, in order:

Group Box	Model
Enabled Models	**Three-Dimensional** (Selected automatically)
Time	**Implicit Unsteady**
Material	**Gas**
Flow	**Coupled Flow**
Enabled Models	**Gradients** (Selected automatically)
Equation of State	**Ideal Gas**
	Coupled Energy (Selected automatically)
Viscous Regime	**Laminar**
Optional Models	**Gravity**
	Lagrangian Multiphase
	Discrete Element Model (DEM)
	Multiphase Interaction (Selected automatically)

2. Click **Close**.
3. Open the **Physics 1** > **Models** manager.

The selected models appear as shown below:



4. Save the simulation.

Defining Lagrangian Phases

A Lagrangian phase is used to define the solid particles in the simulation. The phase requires specific physics models and properties to characterize its behavior. In this case, the solid particles are grains of wheat, with the material properties defined manually.

To define the Lagrangian phase:

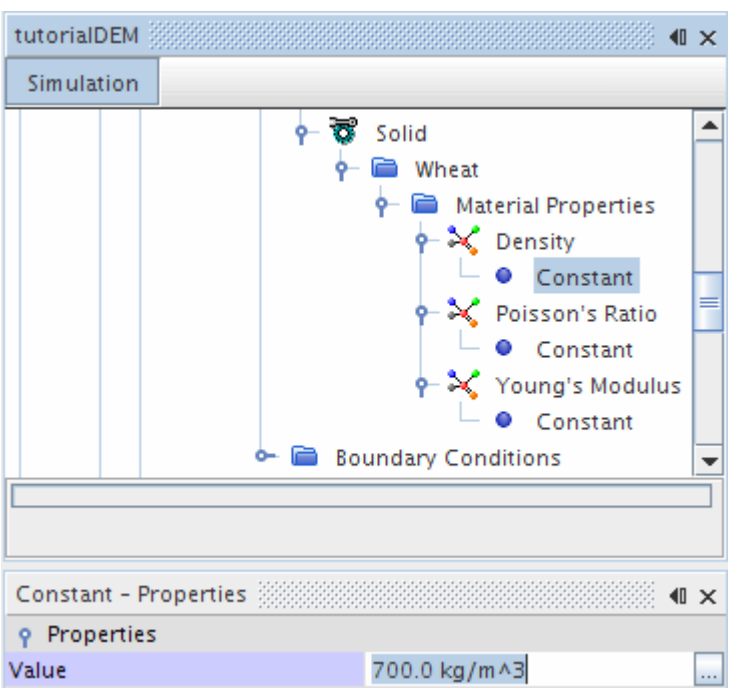
1. Right-click the **Lagrangian Multiphase** > **Lagrangian Phases** manager and select **New**.
2. Rename the phase as `Grain`.
3. Right-click the **Grain** > **Models** manager and select **Select models**.
4. In the *Grain Model Selection* dialog, select the following models, in order:

Group Box	Model
Particle Type	**DEM Particles**
Optional Particle Forces	**Pressure Gradient Force** (selected automatically)
Material	**Solid**
Equation of State	**Constant Density**
Particle Shape	**Composite Particles**

5. Click **Close**.

To define the material properties for the Grain particles:

6. Select **Models** > **Solid** > **Rubber** and rename the **Rubber** node to `Wheat`.
7. Select the **Material Properties** > **Density** > **Constant** node and set the *Value* to `700 kg/m^3`.

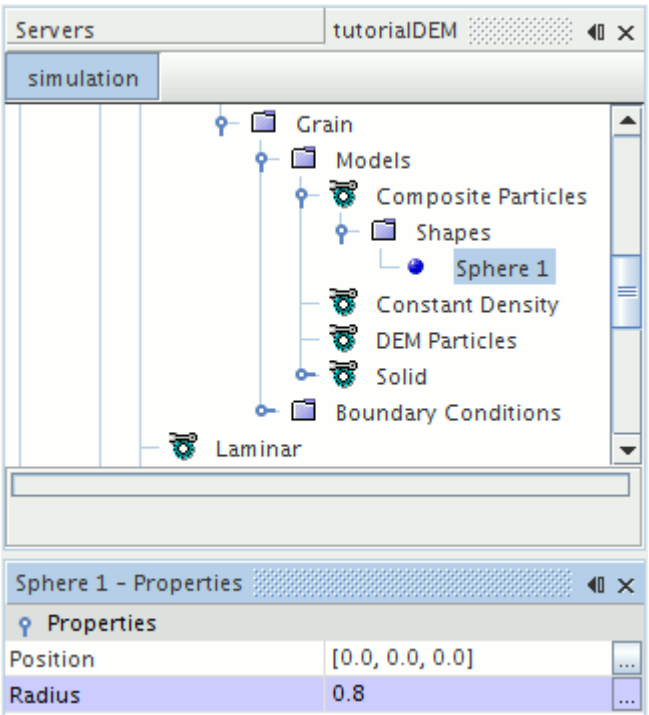


8. Set the remaining properties as follows:

Property	Setting
Poisson's Ratio	`0.5`
Young's Modulus	`0.2 MPa`

To define the physical shape of the grain particles:

9. Right-click the **Model** > **Composite Particles** > **Shapes** node and select **New**.
10. Select the **Sphere 1** node and set the *Radius* to `0.8`.

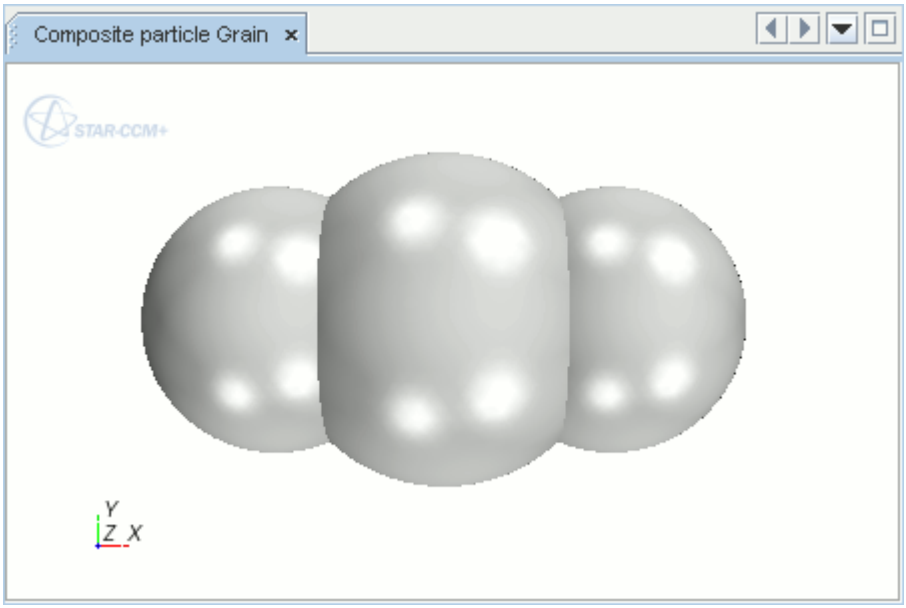


11. Create two more shapes with the following properties:

Node	Property	Setting
Sphere 2	*Position*	`[1.0, 0.0, 0.0]`
	Radius	`1.0`
Sphere 3	*Position*	`[2.0, 0.0, 0.0]`
	Radius	`0.8`

12. Right-click the **Composite Particles** node and select **Show model**.

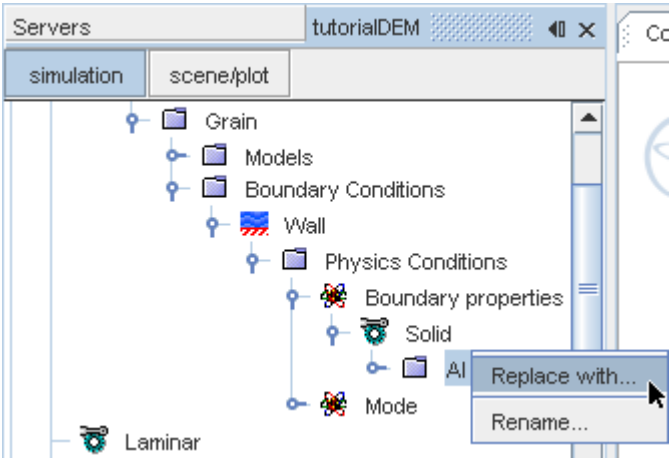
A representation of the grain particle is shown in the *Graphics* window.



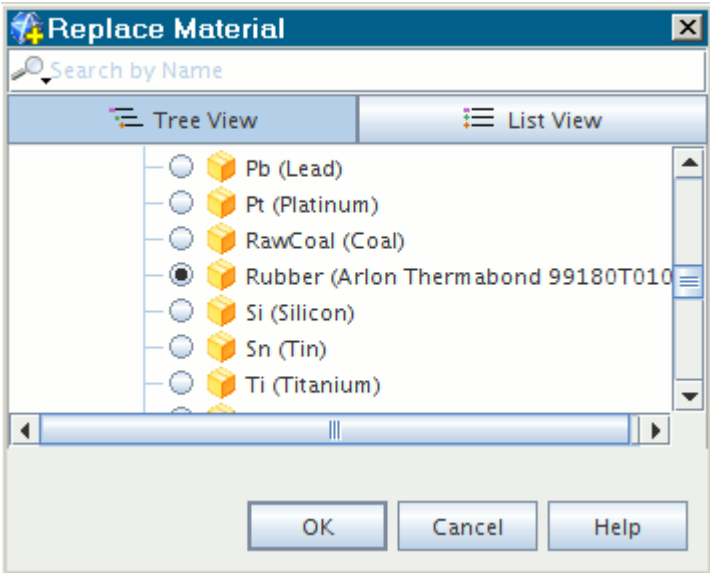
13. Close the *Composite particle Grain* scene.

To define the physics conditions on the particle boundary:

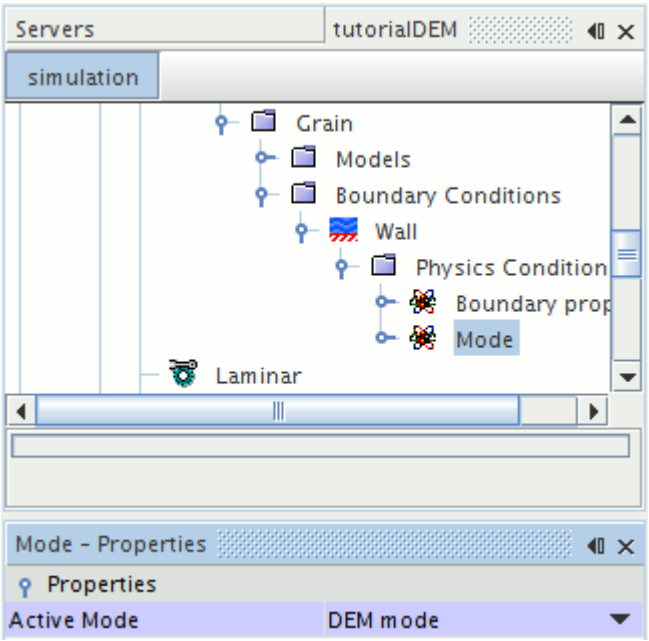
14. Right-click the **Grain** > **Boundary Conditions** > **Wall** > **Physics Conditions** > **Boundary Properties** > **Solid** > **Al** node and select **Replace with**.



15. In the *Replace Material* dialog, expand the **Material Databases** > **Standard** > **Solids** manager, select **Rubber (Arlon Thermabond 99180T010)** and click **OK**.



16. Select the **Physics Conditions** > **Mode** node and check that the *Active Mode* property is set to **DEM mode**.



17. Save the simulation.

Defining the DEM Particle Interaction

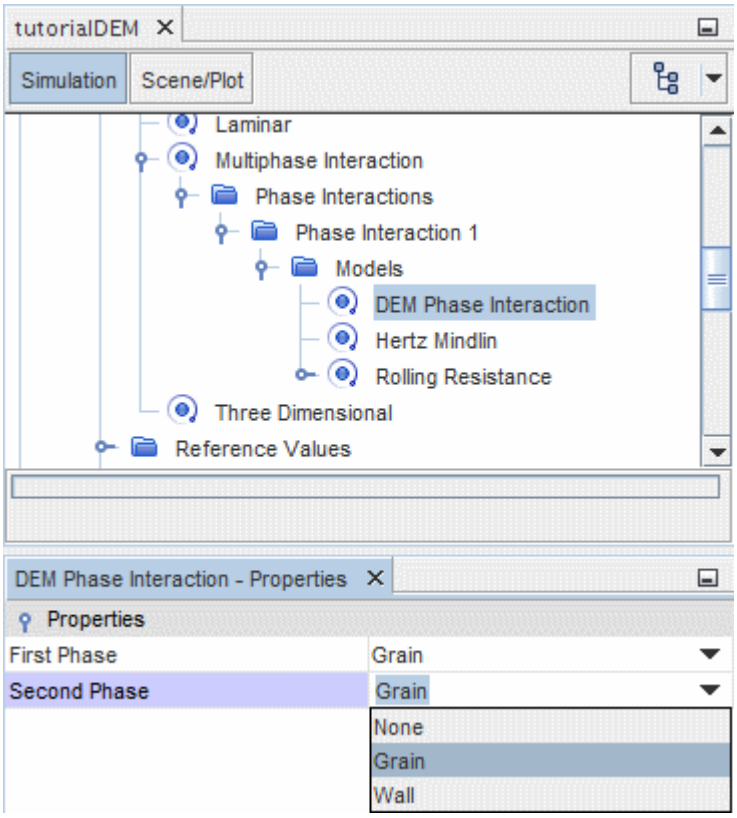
The DEM particle interaction is used to define how the particles behave when they collide with one another and the region walls.

To define the grain to grain interaction:

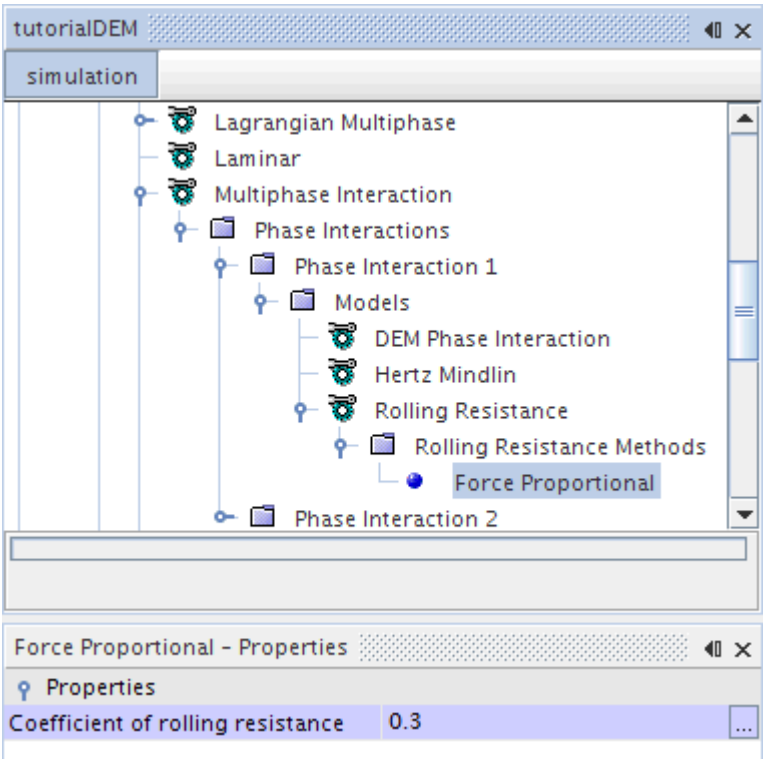
1. Right-click the **Physics 1** > **Models** > **Multiphase Interaction** > **Phase Interactions** manager and select **New**.
2. Right-click the **Phase Interaction 1** > **Models** node and select the following models, in order:

Group Box	Model
Phase Interaction Topology	**DEM Phase Interaction**
Optional Models	**Hertz Mindlin** **Rolling Resistance** (selected automatically)

3. Click **Close**.
4. Select the **Models** > **DEM Phase Interaction** manager and set both the *First phase* and *Second phase* properties to **Grain**.



5. Select the **Models** > **Rolling Resistance** > **Rolling Resistance Methods** > **Force Proportional** node and set the *Coefficient of rolling resistance* to `0.3`. The remaining properties can be left as default.



To define the Grain to Wall interaction, follow the steps above and create a particle interaction model, **Phase Interaction 2**, with the following settings:

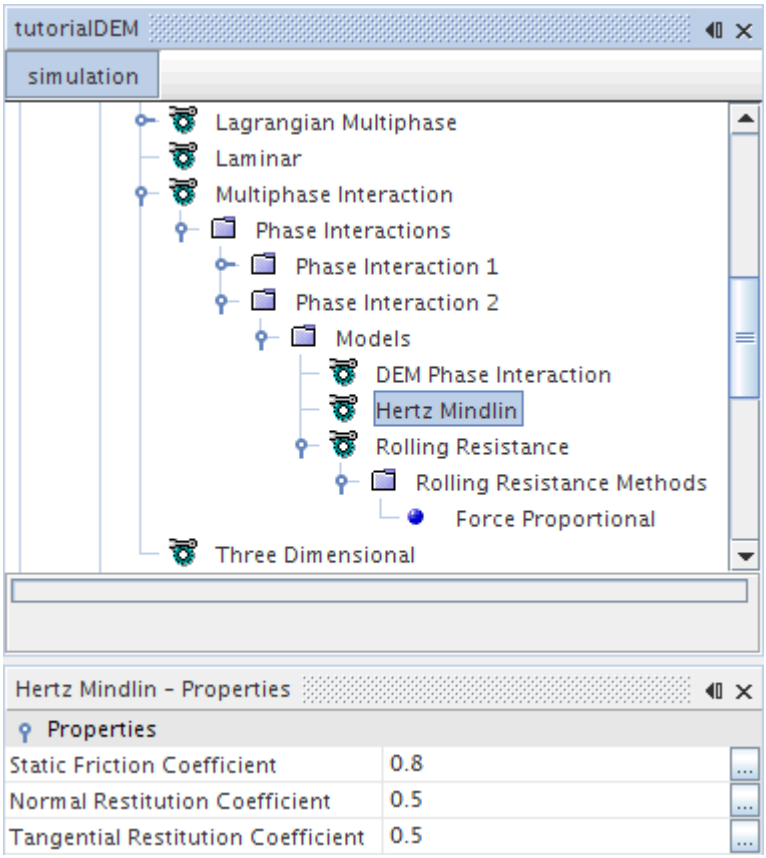
6. Right-click the **Phase Interaction 2** > **Models** node and select the following models, in order:

Group Box	Model
Phase Interaction Topology	**DEM Phase Interaction**
Optional Models	**Hertz Mindlin**

7. Set the following properties:

Node	Property	Setting
DEM Phase Interaction	*First Phase*	**Grain**
	Second Phase	**Wall**
Hertz Mindlin	*Static Friction Coefficient*	0.8
	Normal Restitution Coefficient	0.5
	Tangential Restitution Coefficient	0.5
Rolling Resistance > **Rolling Resistance Methods** > **Force Proportional**	*Coefficient of rolling resistance*	0.3

This part of the simulation tree along with the *Hertz Mindlin - Properties* window appears as shown below:



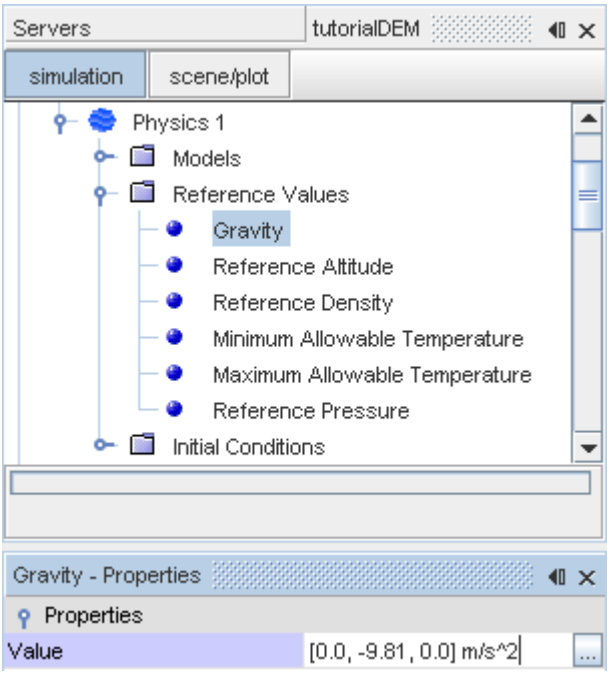
8. Save the simulation.

Setting Reference Values

Define the reference values to make sure that the acceleration due to gravity acts in the expected direction.

To set the gravity reference value:

- Select the **Physics 1** > **Reference Values** > **Gravity** node and set the *Value* to `[0.0, -9.81, 0.0] m/s^2`.

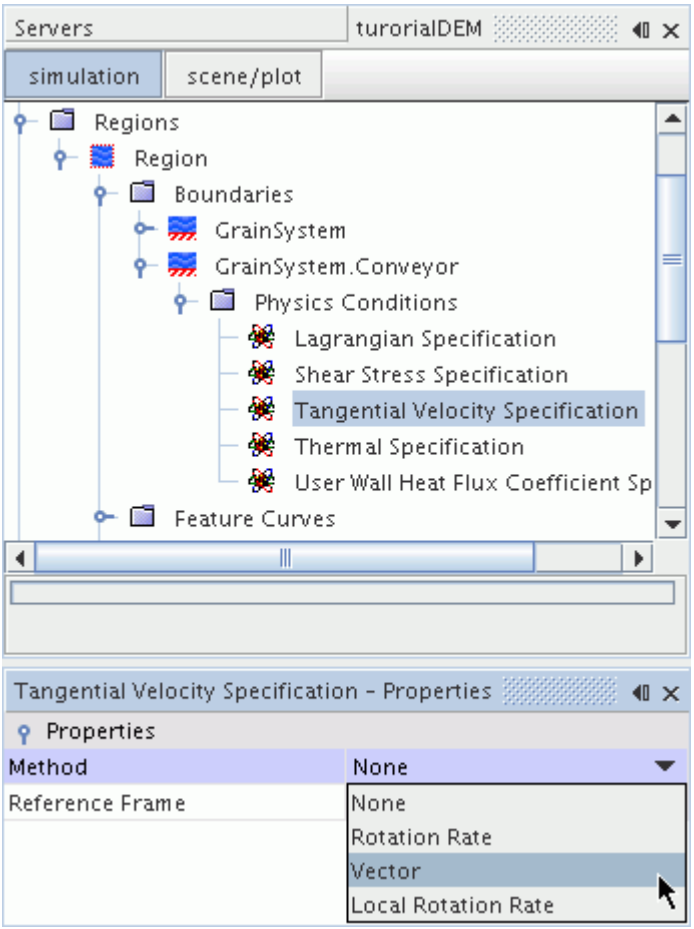


Setting the Moving Conveyor Wall Condition

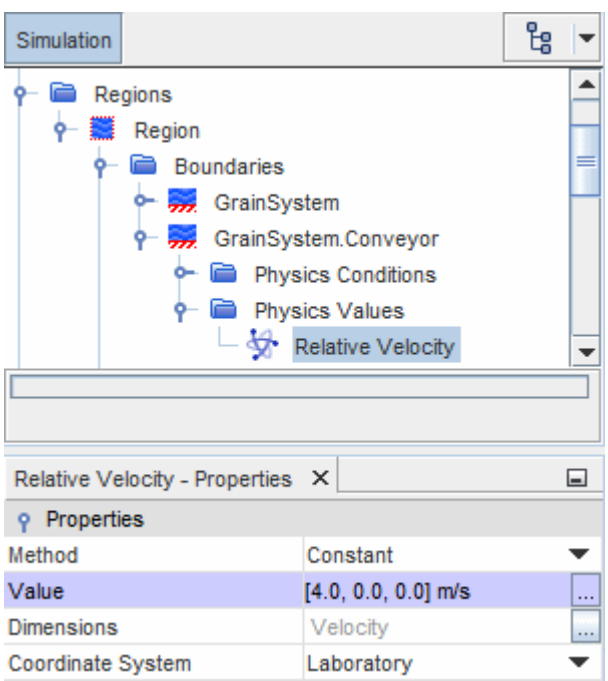
Define the conveyor boundary with a translational velocity to represent the movement of the belt.

To set the moving wall condition:

1. Expand the **Regions** > **Region** > **Boundaries** > **GrainSystem.Conveyor** > **Physics Conditions** manager, and select the **Tangential Velocity Specification** node and set the *Method* to **Vector**.



2. Select the **Physics Values** > **Relative Velocity** node and set *Value* to `[4.0, 0.0, 0.0] m/s`.



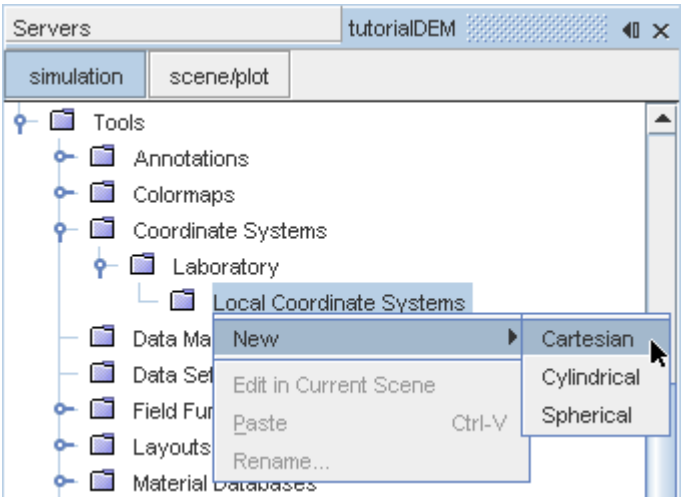
3. Save the simulation.

Creating an Injector

Create the local Cartesian coordinate system.

Use the following steps:

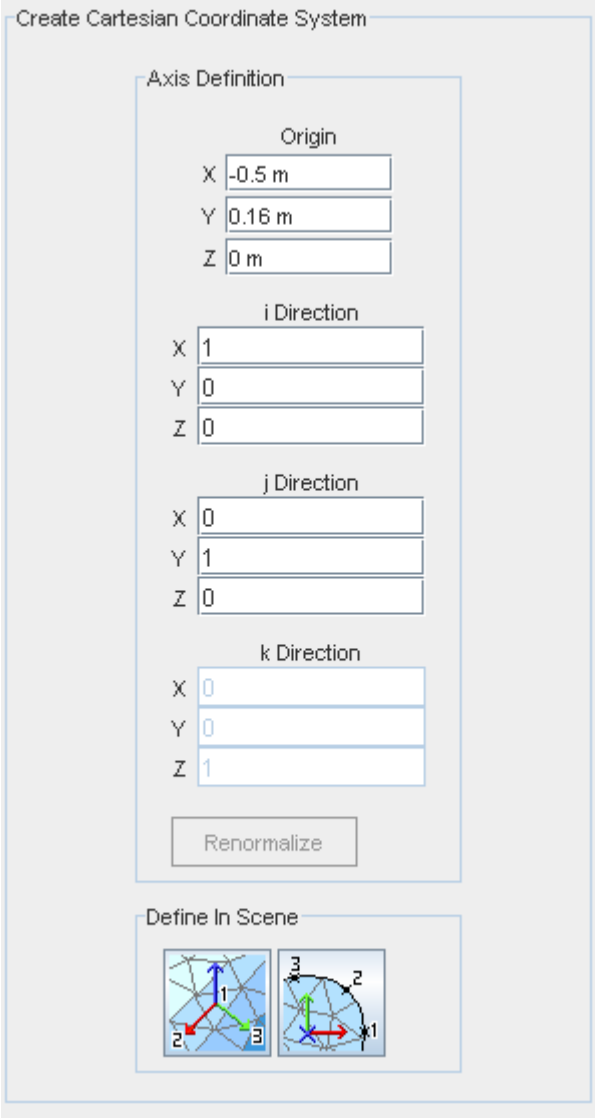
1. Right-click the **Tools** > **Coordinate System** > **Laboratory** > **Local Coordinate System** node and select **New** > **Cartesian**.



2. Using the in-place dialog, define the coordinate system with the following settings:

Property	Setting
Origin	`X: -0.5 m`
	`Y: 0.16 m`
	`Z: 0.0 m`
i Direction	`X: 1`
	`Y: 0`
	`Z: 0`
j Direction	`X: 0`
	`Y: 1`
	`Z: 0`

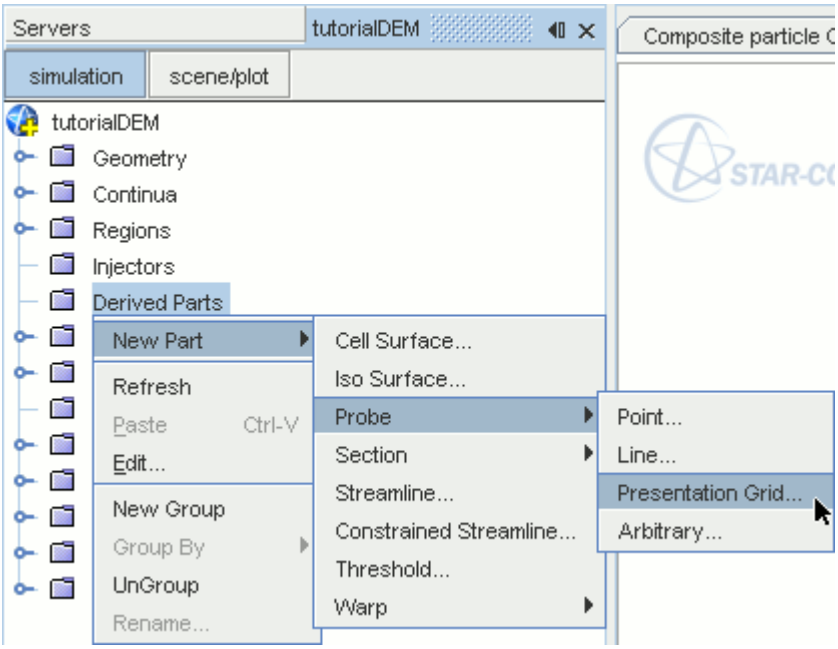
The dialog appears as follows.



3. Click **Create** then **Close**.

To create the presentation grid using the *Properties* window:

4. Right-click the **Derived Parts** manager and select **New Part** > **Probe** > **Presentation Grid**.



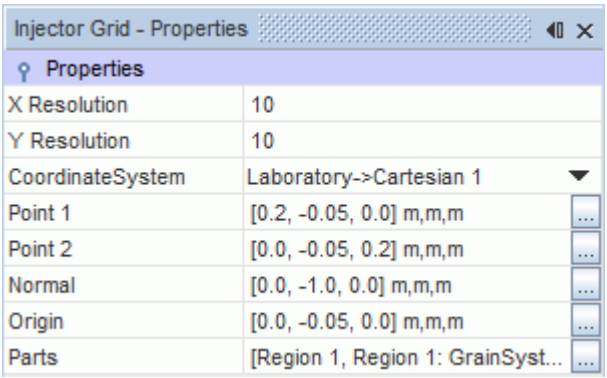
5. Using the in-place dialog, click **Create** and **Close**.

An arbitrary plane is created with the default settings.

To modify the settings:

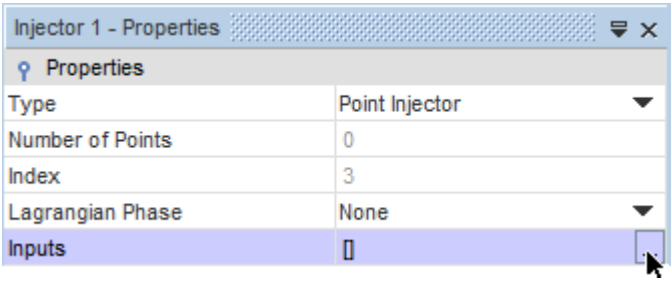
6. Rename the **presentation grid** node to `Injector Grid`.
7. Use the following settings to define the injector grid:

Property	Setting
X Resolution	`10`
Y Resolution	`10`
Coordinate System	`Laboratory->Cartesian 1`
Point 1	`[0.2, -0.05, 0.0] m,m,m`
Point 2	`[0.0, -0.05, 0.2] m,m,m`
Origin	`[0.0, -0.05, 0.0] m,m,m`

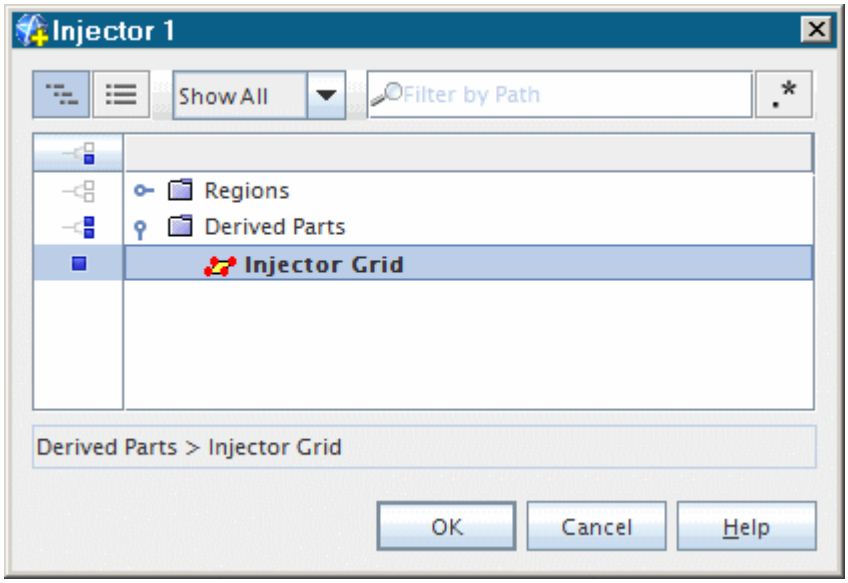


Finally, the particle injector is created.

8. Right-click the **Injectors** node and select **New**.
9. Select the **Injector 1** node and set the *Type* to **Part Injector**.
10. To assign the injector to the previously created injector grid, click the ellipsis next to the *Inputs* property.

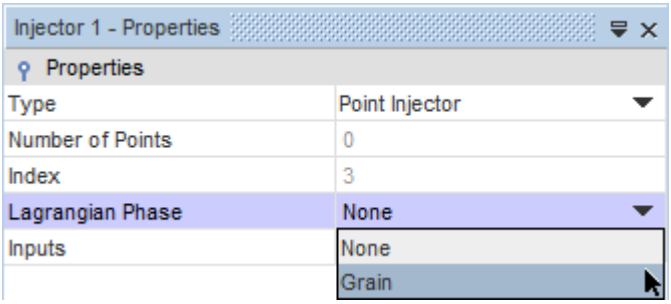


11. In the *Inputs* selection dialog, expand the **Derived Parts** node and tick the checkbox next to the **Injector Grid** node.



12. Click **OK** to close the dialog.

13. While still at the **Injector 1** node, set the *Lagrangian Phase* to **Grain**.



14. Expand the **Injector 1** node and set the following properties:

Node	Property	Setting
Conditions		
└ **Flow Rate Distribution**	**Method**	**Per Injector Point**
└ **Flow Rate Specification**	**Method**	**Particle Flow Rate**
Values		
└ **Particle Diameter** > **Constant**	**Value**	`0.01 m`
└ **Particle Flow Rate** > **Constant**	**Value**	`$Time < 0.5 ? 100 : 0`
└ **Velocity** > **Constant**	**Value**	`[0.0, -2.0, 0.0] m/s`

15. Save the simulation.

Setting the Solver Parameters and Stopping Criteria

As the simulation is transient, set the solver time-step and elapsed time.

To define the solver parameters:

1. To define the solver parameters, expand the **Solvers** node and set the following properties:

Node	Property	Value
Implicit Unsteady	**Time-Step**	`0.01`
DEM Solver	**Time Scale**	`0.1`

2. To define the stopping criteria, select the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Inner Iterations	*Maximum Inner Iterations*	`1`
Maximum Physical Time	*Maximum Physical Time*	`2.5 s`

Node	Property	Setting
Maximum Steps	*Enabled*	Deactivated

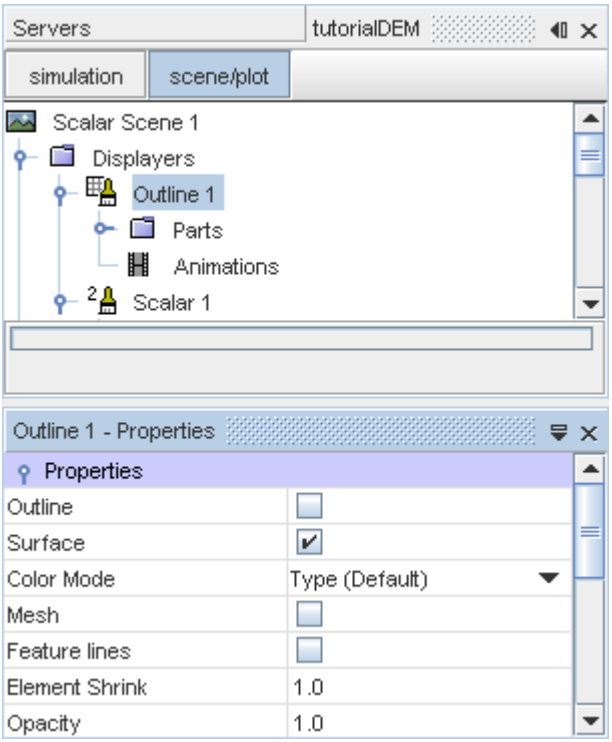
3. Save the simulation.

Setting Up a Visualization Scene

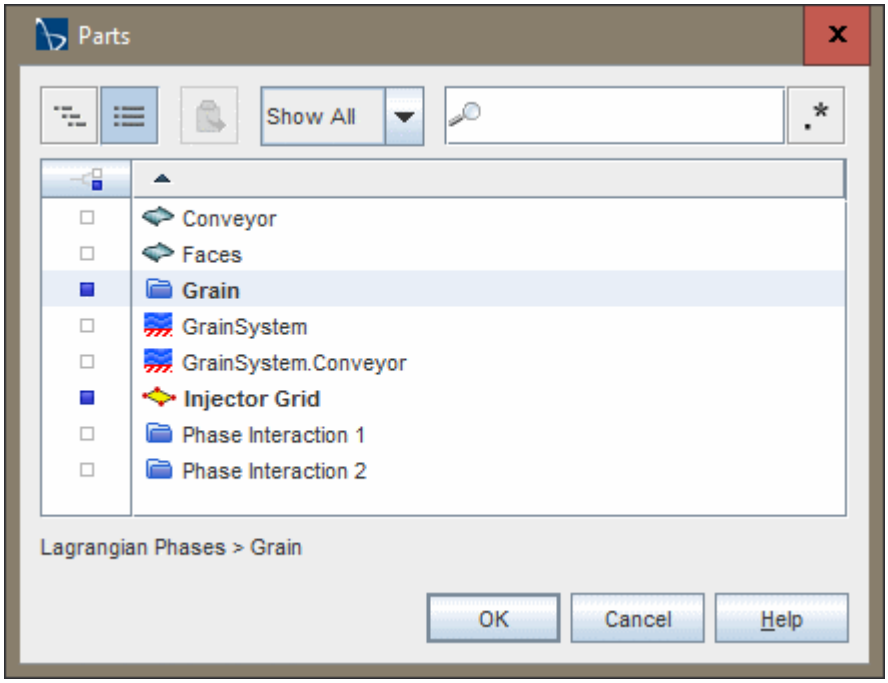
Create a scalar scene to provide a visual representation of the grain particles within the fluid system.

Use the following steps:

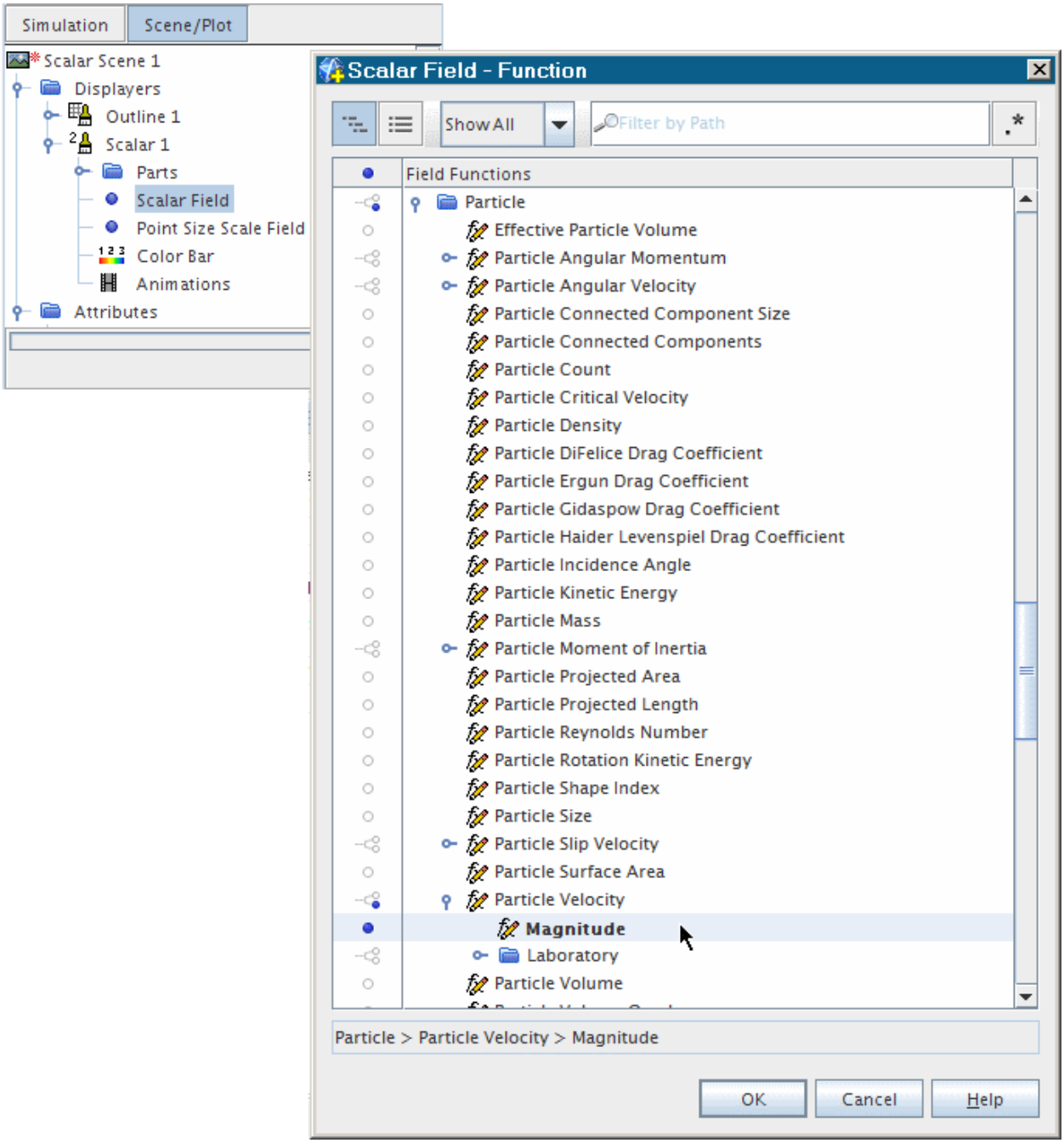
1. Right-click the **Scenes** node, and select **New Scene** > **Scalar**.
2. Click the **Scene/Plot** button above the object tree.
3. Select the **Outline 1** node.
   - Deactivate the *Outline* property.
   - Activate the *Surface* property.



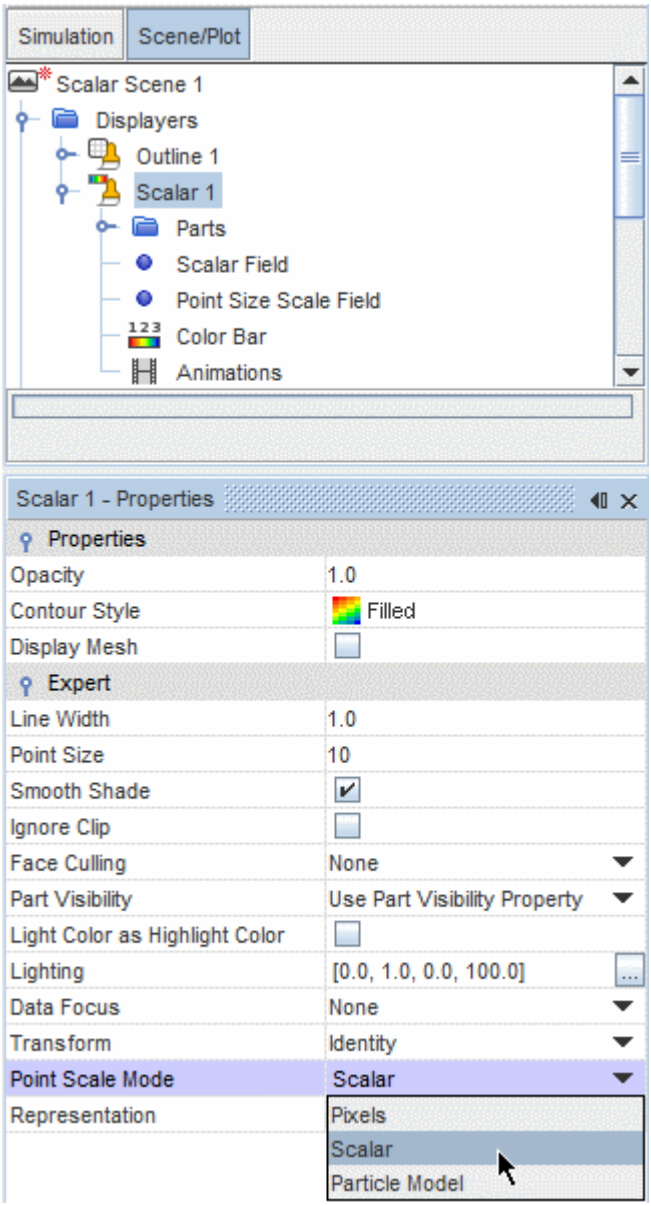
4. Right-click the **Scalar 1** > **Parts** manager and select **Edit**.
5. In the *Parts* dialog, select **Injector Grid** and **Grain**. Click **OK**.



6. Select the **Scalar Field** node and set the *Function* to **Particle** > **Particle Velocity** > **Magnitude**.

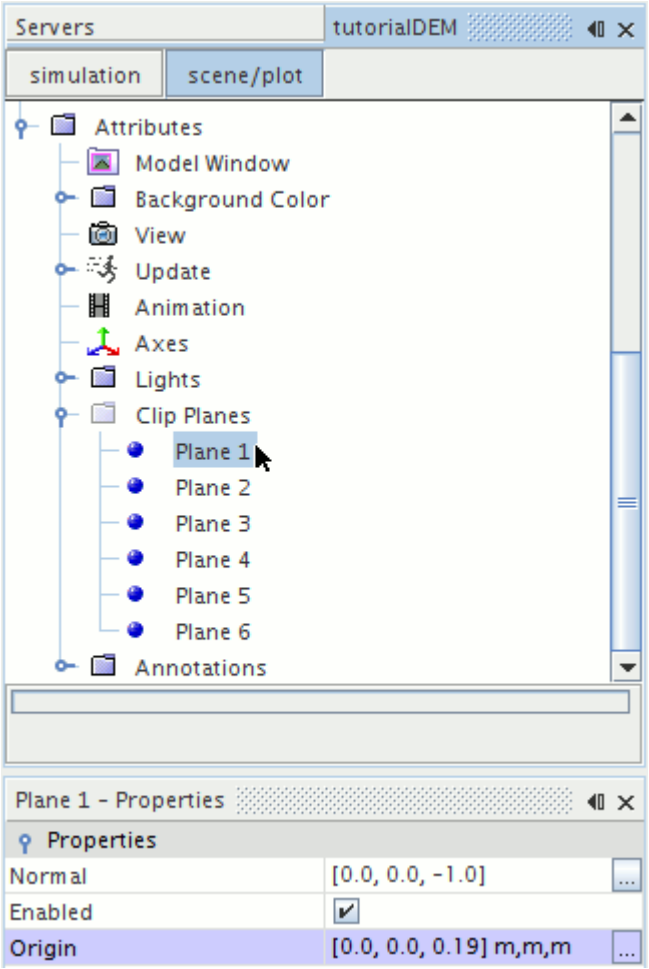


7. Select the **Scalar 1** node and set the *Point Scale Mode* to **Scalar**.

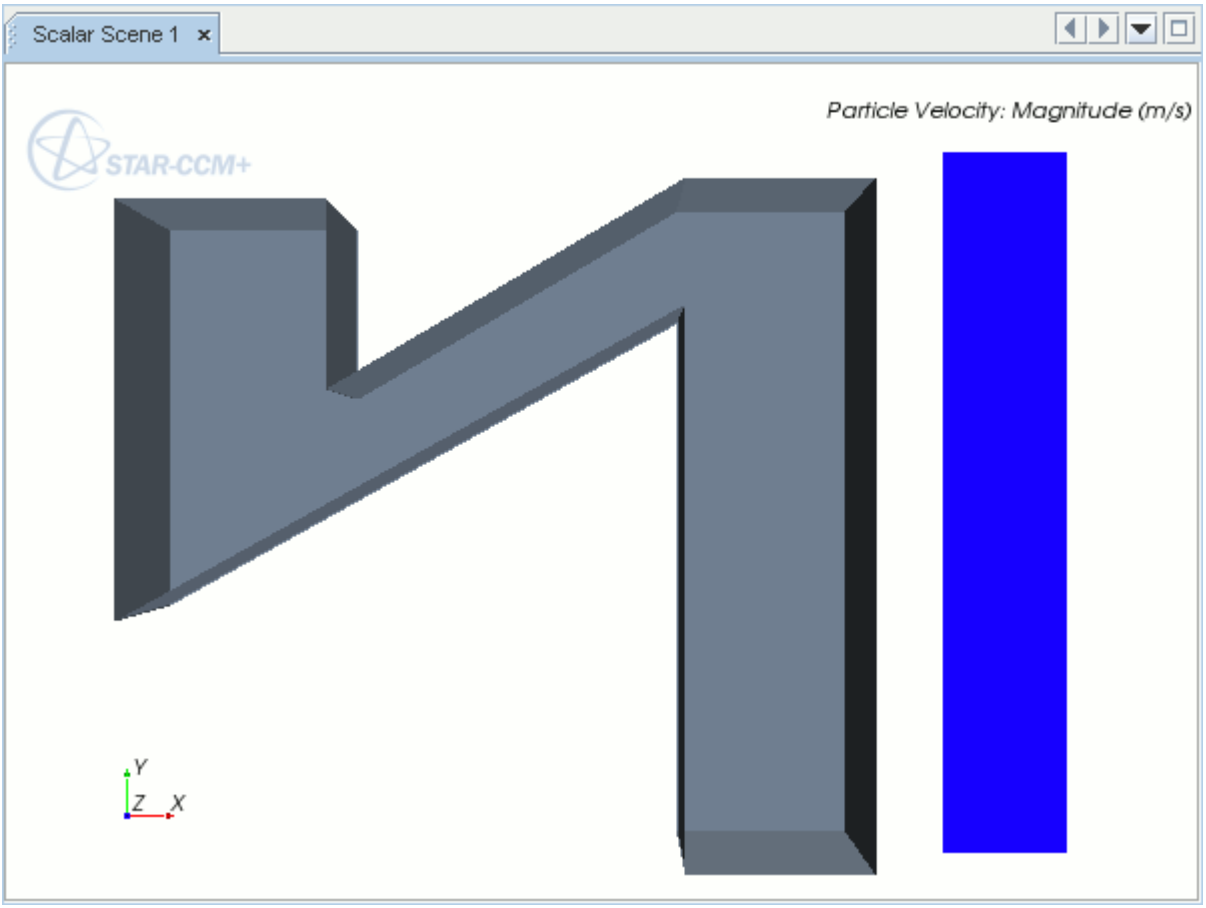


8. Select the **Attribute** > **Clip Plane** > **Plane 1** node and set the following properties:

Property	Setting
Normal	`[0.0, 0.0, -1.0]`
Enabled	Activated
Origin	`[0.0, 0.0, 0.19] m, m, m`

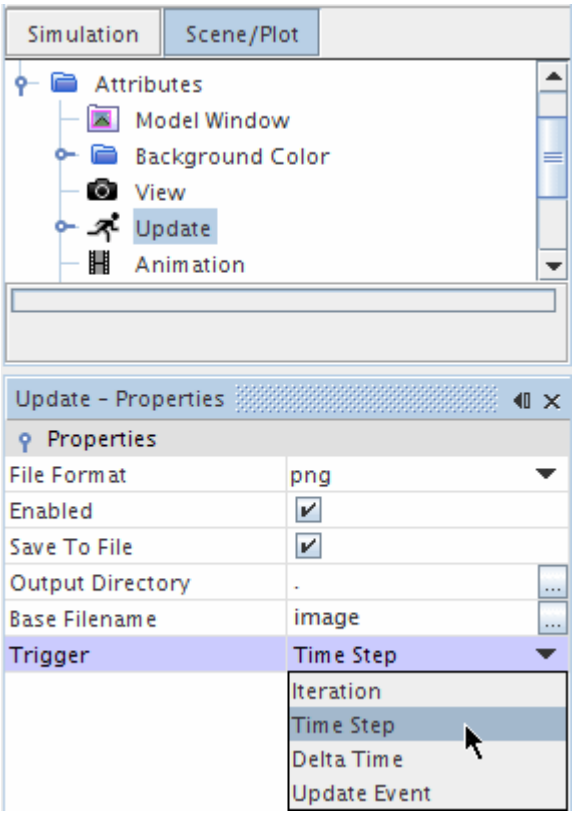


9. Rearrange the color bar and display to provide the following view.

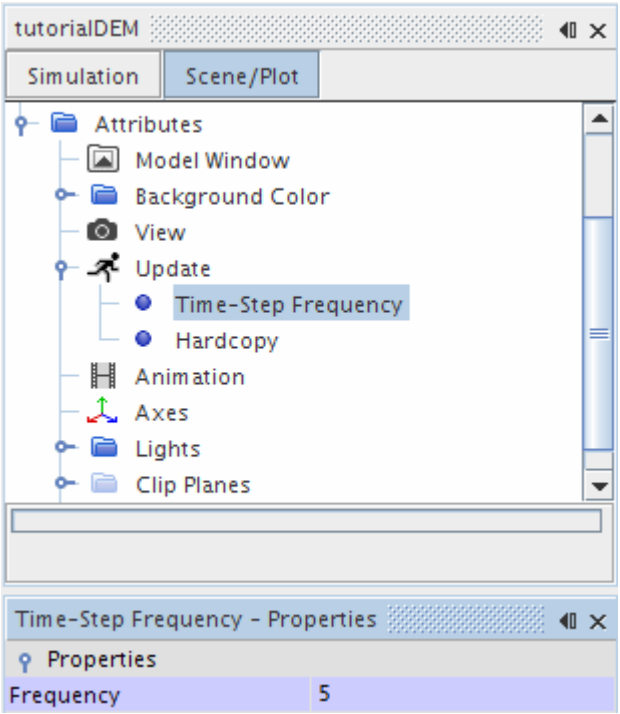


To export an image of the visualization at every time-step:

10. Select the **Attributes** > **Update** node and set the *Trigger* to **Time Step**.
11. Activate the checkbox for the *Save to File* property.



12. Expand the **Update** manager, select the **Update** > **Time-Step Frequency** node, and set the *Frequency* to `5`.



13. Save the simulation.

Running the Simulation

The simulation is now ready to be run.

Use the following steps:

1. Run the simulation by clicking (**Run**) in the **Solution** toolbar.

The iterative progress can be monitored in the *Output* window. The scalar scene displays the injection and movement of the grain particles within the fluid region as the solution proceeds.

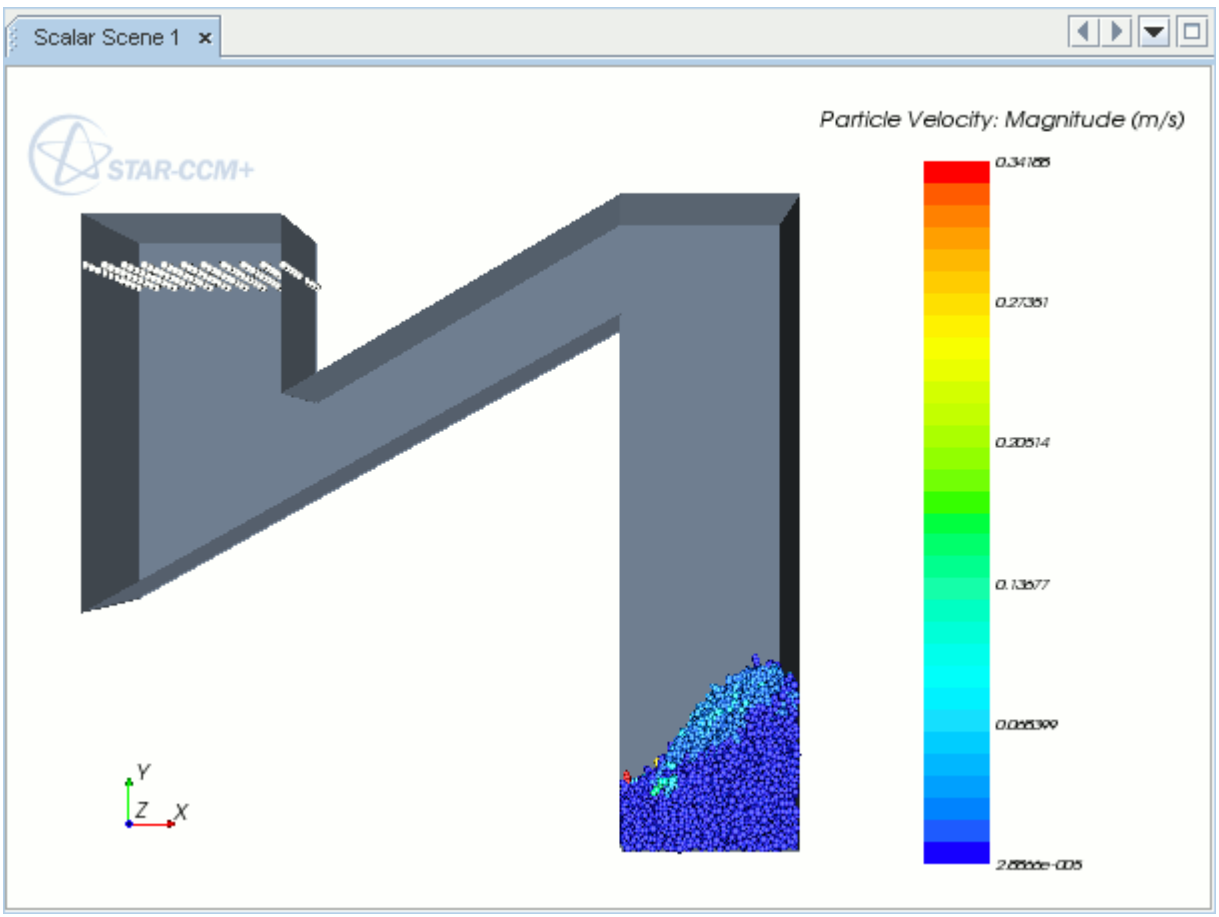
When the solution is complete, the following message is displayed in the *Output* window: `Stopping criterion Maximum Physical Time satisfied.`

2. Save the simulation.

Visualizing Results

When the solver is complete, the *Scalar Scene 1* in the *Visualization* window appears.

See below:



The scene shows the grain particles in the collection area, after being transported up the conveyor. A series of images have been created in the current working directory named **Scalar_Scene_1_image_xxxxx.png** , where **xxxxx** refers to the time-step number. These images show the particle movement through the region during the transient solution, and can be used to create an animation.

Summary

This tutorial provides a guide to setting up a simulation using the Discrete Element Method.

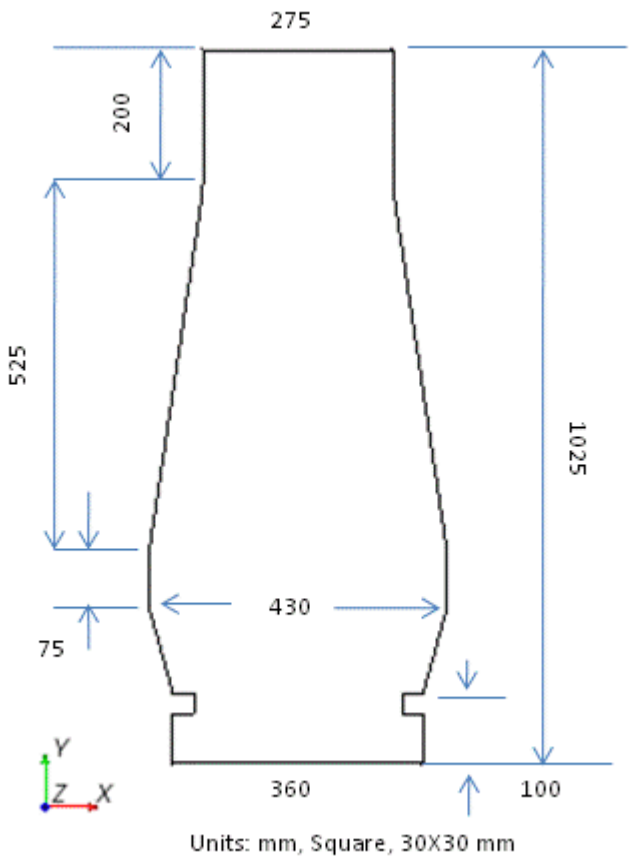
The tutorial has introduced the following STAR-CCM+ features:

- Selecting fluid physics models.
- Definition of Lagrangian phases.
- Definition of DEM particle interaction.
- Creation of a particle injector.
- Setting up visualization scenes.

DEM Particle Settling

This tutorial simulates a three-dimensional solid flow through a model blast furnace to predict the formation of the deadman region at the base of the furnace.

Based on the geometry and raceway size from Zhou *et al.*, a model of an iron making blast furnace is constructed as shown below.



The particles are consumed near the raceway in an actual furnace. In this case, particle loss is instead simulated by introducing a particle outlet at the raceway location. Changes in particle size and property are not taken into account.

The main aim of this tutorial is to present a comparison of the predicted deadman size due to solid descent (using particle residence time) with the laboratory scale experiments of Zhou *et al*.

The three stages of the simulation are:

1. Particle bed formation: In the first stage, the simulation is run for 11.5 s with the inlet open to allow the furnace to be filled with particles (glass beads) to create the initial bed.
2. Particle settlement: In this stage, the simulation is run for a further 2 s with the inlet closed to allow the bed to settle.

For the first two stages the outlets are closed to allow the build-up and settlement of the glass beads in the furnace.

3. Particle discharge: In the final stage, the simulation is run for a further 36.5 s with the inlet and outlets (one on each side) open.

Particle properties in this simulation are as follows:

- Spherical, 10 mm diameter particles of density 2500 kg/m^3.
- Sliding friction coefficient of 0.4 for both particle-particle and wall-particle interactions.
- Particle mass flow rate of 18.33 kg/s.

Contents:

Importing the Surface Mesh and Naming the Simulation
Generating the Volume Mesh
Selecting Physics Models
Defining Lagrangian Phases
Defining the DEM Particle Interaction
Setting Reference Values
Setting the Boundary Conditions
Creating an Injector
Setting the Solver Parameters and Stopping Criteria
Setting Up a Visualization Scene
Running the Simulation
Visualizing Results
Stage Two of the Simulation
Stage Three of the Simulation
Comparing the Results
Summary
Blast Furnace Tutorial Bibliography

Importing the Surface Mesh and Naming the Simulation

Start a new simulation and import `blastFurnaceSurfaceMesh.dbs`.

Use the following steps:

1. Navigate to the `doc/startutorialsdata/discreteElementMethod/data` subdirectory of the STAR-CCM+ installation directory. Copy the `blastFurnaceSurfaceMesh.dbs` file to the current working directory.
2. Start up STAR-CCM+ in the usual manner and select the New Simulation option from the menu bar.

A surface mesh for the blast furnace has been prepared for this analysis. Continue by importing the surface mesh and saving the simulation.

3. From the menu bar, select **File** > **Import** > **Import Surface Mesh...**

4. In the *Open* dialog, navigate to the working directory and select the file `blastFurnaceSurfaceMesh.dbs`.
5. To start the import, click **Open**.
6. To accept the default *Import Surface Options*, click **OK**.

   Note: The **One boundary per cell type** option results in a different boundary being created for each cell type, property ID or solid depending on the input file type.

7. To create a geometry scene, activate the **Open Geometry Scene After Import** checkbox.

STAR-CCM+ provides feedback on the import process, which takes a few seconds, in the *Output* window. A geometry scene is created in the *Graphics* window.

8. Save the new simulation to disk under file name `blastFurnace.sim`.

Generating the Volume Mesh

Use the surface remesher and polyhedral volume mesher to create a mesh with a base size of 0.01 m, as the fluid solution is not of particular importance for this tutorial.

Use the following steps:

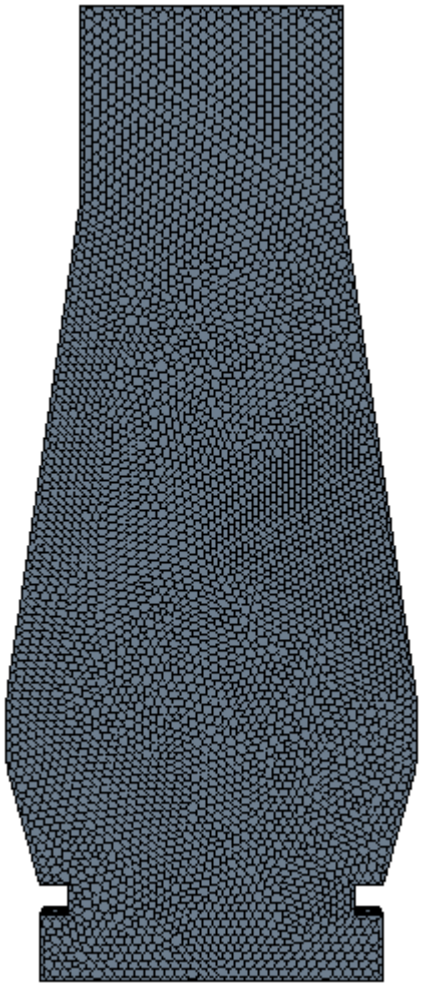
1. Right-click the **Geometry** > **Parts** > **blastFurnaceSurfaceMesh** node and select **Assign Parts to Regions...**.
2. In the *Assign Parts to Regions* dialog, make sure that the **blaskFurnaceSurfaceMesh** part is selected, then select **Create a Boundary for Each Part Surface** in the second pull-down menu.
3. Click **Apply**, and then **Close**.
4. Right-click **Geometry** > **Operations** and select **New** > **Mesh** > **Automated Mesh**.
5. In the *Create Automated Mesh Operation* dialog:
   a) In the *Parts* list, select **blastFurnaceSurfaceMesh**
   b) In the *Select Meshers* box, select the following meshers:

Group	Mesher
Surface Meshers	**Surface Remesher**
Core Volume Meshers	**Polyhedral Mesher**

6. Select the **Operations** > **Automated Mesh** > **Default Controls** > **Base Size** node and set **Value** to `0.01 m`.
7. Click (**Generated Volume Mesh**) in the **Mesh** toolbar.

   The output from the meshers is shown in the *Output* window. When the meshing process is complete, the message `Volume Meshing Pipeline Completed` is displayed in the *Output* window.
8. Select **Scenes** > **Geometry Scene 1** and set the *Mesh Override* property to **Show All Meshes** to see the surface mesh.

9. Right-click **Geometry Scene 1** and select **Apply Representation** > **Volume Mesh** to see the volume mesh.



10. Save the simulation.

Selecting Physics Models

Physics models are required to define the behavior of the multiphase continua.

To set the physics models:

1. For the **Continua** > **Physics 1**, select the following models, in order:

Group Box	Model
Enabled Models	**Three-Dimensional** (Selected automatically)
Time	**Implicit Unsteady**
Material	**Gas**

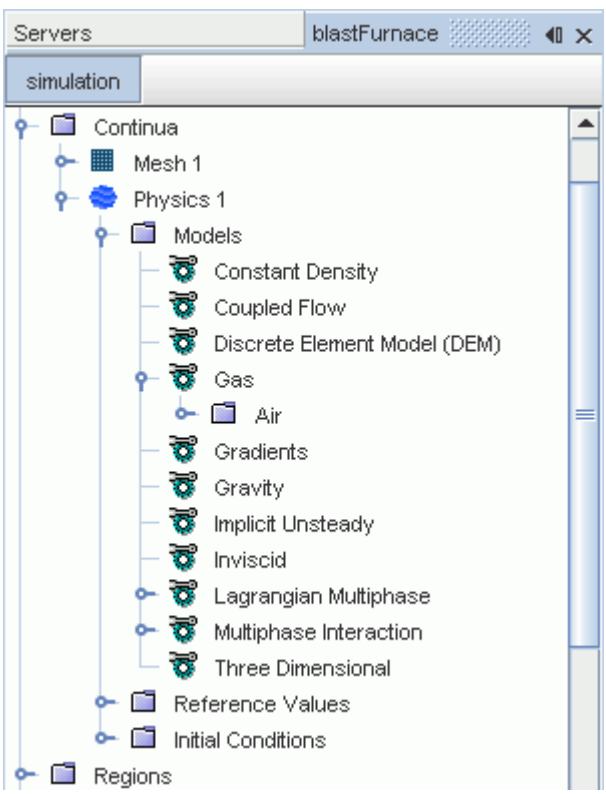
Group Box	Model
Flow	**Coupled Flow**
Enabled Models	**Gradients** (Selected automatically)
Equation of State	**Constant Density**
Viscous Regime	**Inviscid**
Optional Models	**Gravity**
	Lagrangian Multiphase
	Discrete Element Model (DEM)
Enabled Models	**Multiphase Interaction** (Selected automatically)

2. Click **Close**.

The color of the **Physics 1** node turns from gray to blue to indicate that fluid models have been activated and the continua is valid.

3. Open the **Physics 1** > **Models** node.

The selected models appear as shown below:



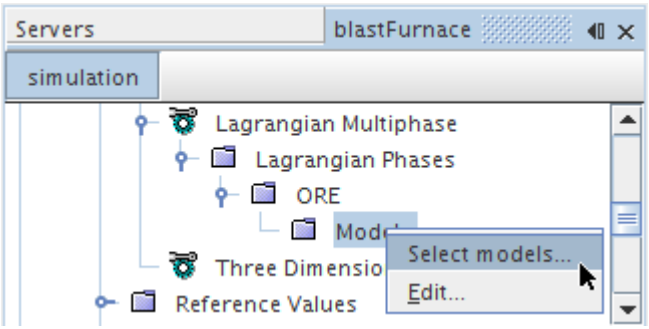
4. Save the simulation.

Defining Lagrangian Phases

A Lagrangian phase is used to define the solid particles in the simulation. The phase requires specific physics models and properties to characterize its behavior. In this case, the solid particles are glass beads, with the material properties defined manually.

To define the Lagrangian phase:

1. Right-click the **Lagrangian Multiphase** > **Lagrangian Phases** node and select **New**.
2. Rename the phase to `ORE`
3. Right-click the **ORE** > **Models** node and choose **Select models**



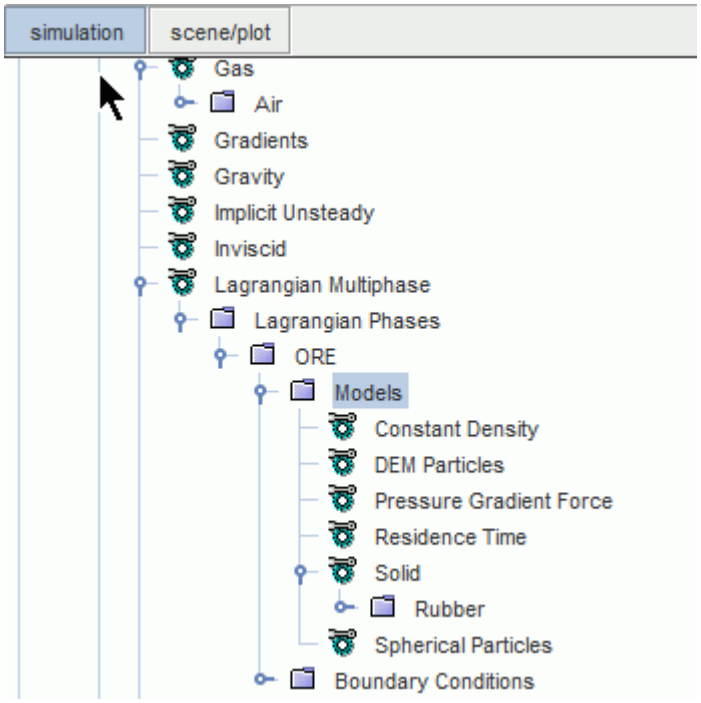
4. In the *ORE Model Selection* dialog, select the following models, in order:

Group Box	Model
Particle Type	**DEM Particles**
Optional Particle Forces	**Pressure Gradient Force** (selected automatically)
Particle Shape	**Spherical Particles**
Material	**Solid**
Equation of State	**Constant Density**
Optional Models	**Residence Time**

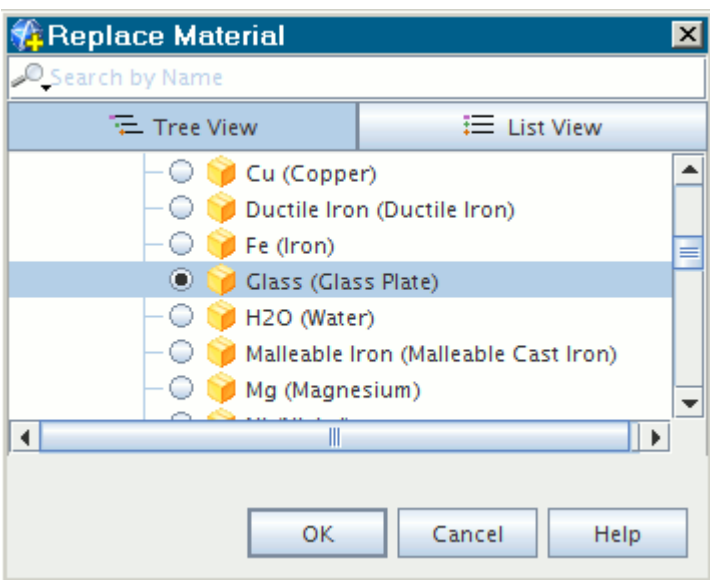
5. Click **Close**.

Now you can set an appropriate material for the Ore.

6. Right-click the **Rubber** node and select **Replace with...**

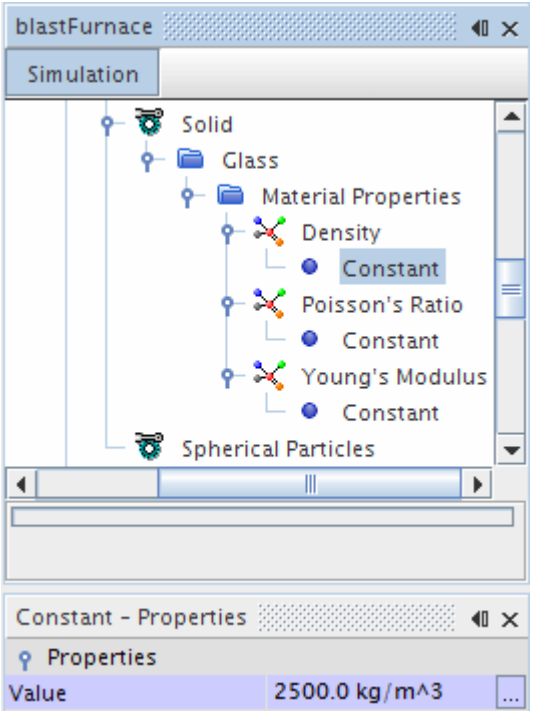


7. In the *Replace Material* dialog, expand the **Material Databases** > **Standard** > **Solids** node, select **Glass (Glass Plate)**, and click **OK**.



Now define the material properties for the glass beads:

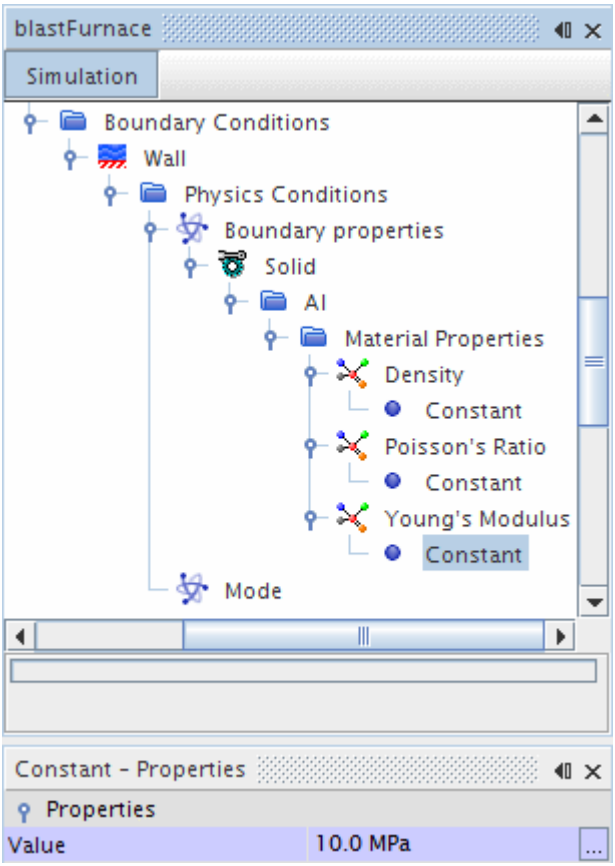
8. Select the **Material Properties** > **Density** > **Constant** node and make sure that the *Value* property is set to `2500 kg/m^3`.



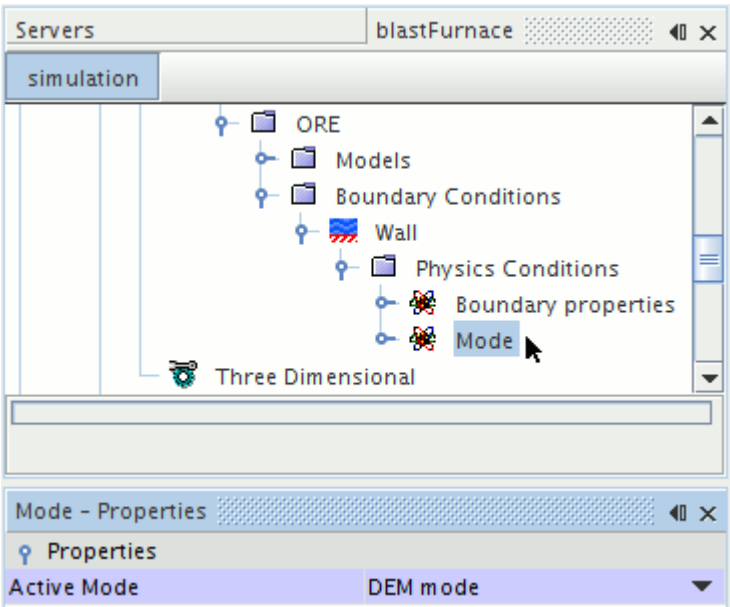
9. Select the **Material Properties** > **Poisson's Ratio** > **Constant** node and set *Value* to `0.3`
10. Select the **Material Properties** > **Young's Modulus** > **Constant** node and set *Value* to `10 MPa`

Define the physics conditions on the particle boundary using the surface material properties and interaction mode.

11. Select the **ORE** > **Boundary Conditions** > **Wall** > **Physics Conditions** > **Boundary properties** > **Solid** > **Al** > **Material Properties** > **Young's Modulus** > **Constant** node and set *Value* to `10 MPa`.



12. Select the **ORE** > **Boundary Conditions** > **Wall** > **Physics Conditions** > **Mode** node and set *Active Mode* to **DEM mode**.



13. Save the simulation.

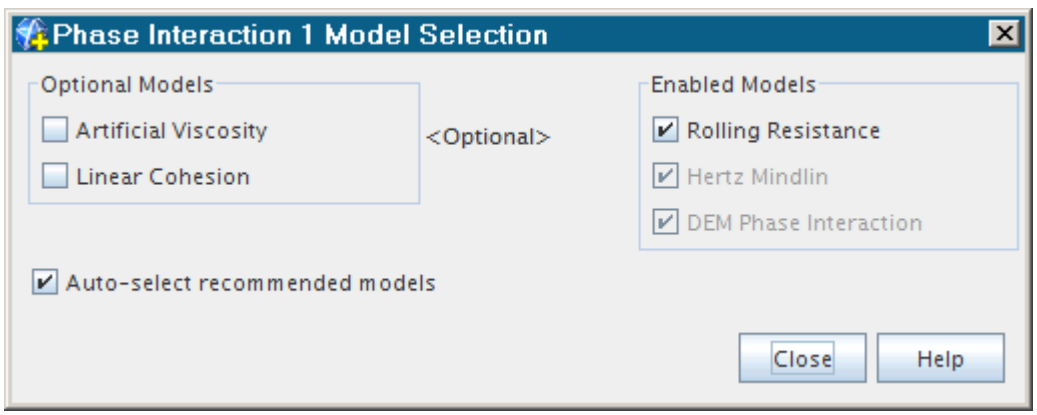
Defining the DEM Particle Interaction

The DEM particle interaction is used to define how the particles behave when they collide with one another and wall regions. The procedure to define the ORE to ORE interaction has been provided in detail, with a table summarizing the steps for ORE to wall interaction.

To define the ORE to ORE interaction:

1. Right-click the **Physics 1** > **Models** > **Multiphase Interaction** > **Phase Interactions** node and select **New**.
2. Right-click the **Phase Interaction 1** > **Models** node and select the following models, in order:

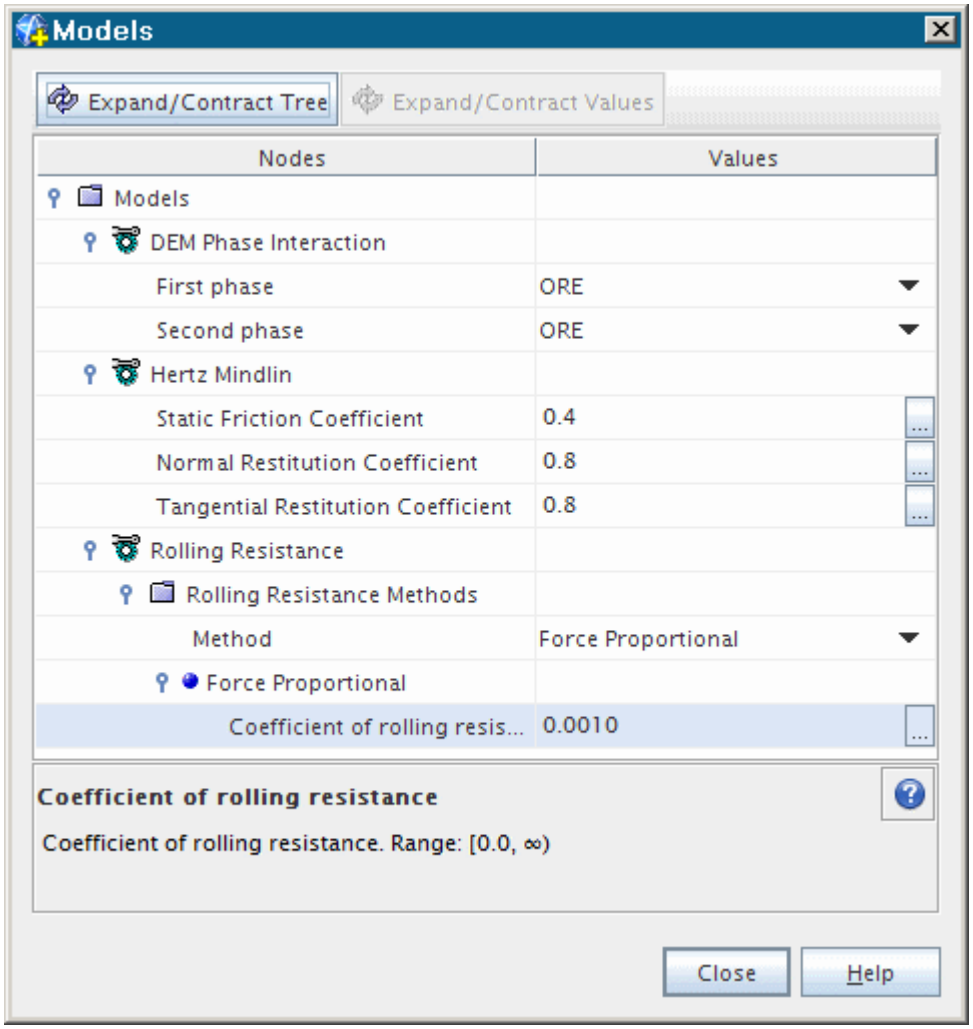
Group Box	Model
DEM Contact Model	**DEM Phase Interaction**
Optional Models	**Hertz Mindlin** **Rolling Resistance** (selected automatically)



3. Right-click the **Phase Interaction 1** > **Models** node and edit the node.
4. Click **Expand/Contract Tree** in the *Edit* dialog, and set the following nodes:

Node	Value
First Phase	**ORE**
Second Phase	**ORE**
Static Friction Coefficient	`0.4`
Normal Restitution Coefficient	`0.8`
Tangential Restitution Coefficient	`0.8`
Coefficient of Rolling Resistance	`0.0010`

The *Models* dialog looks as shown below:

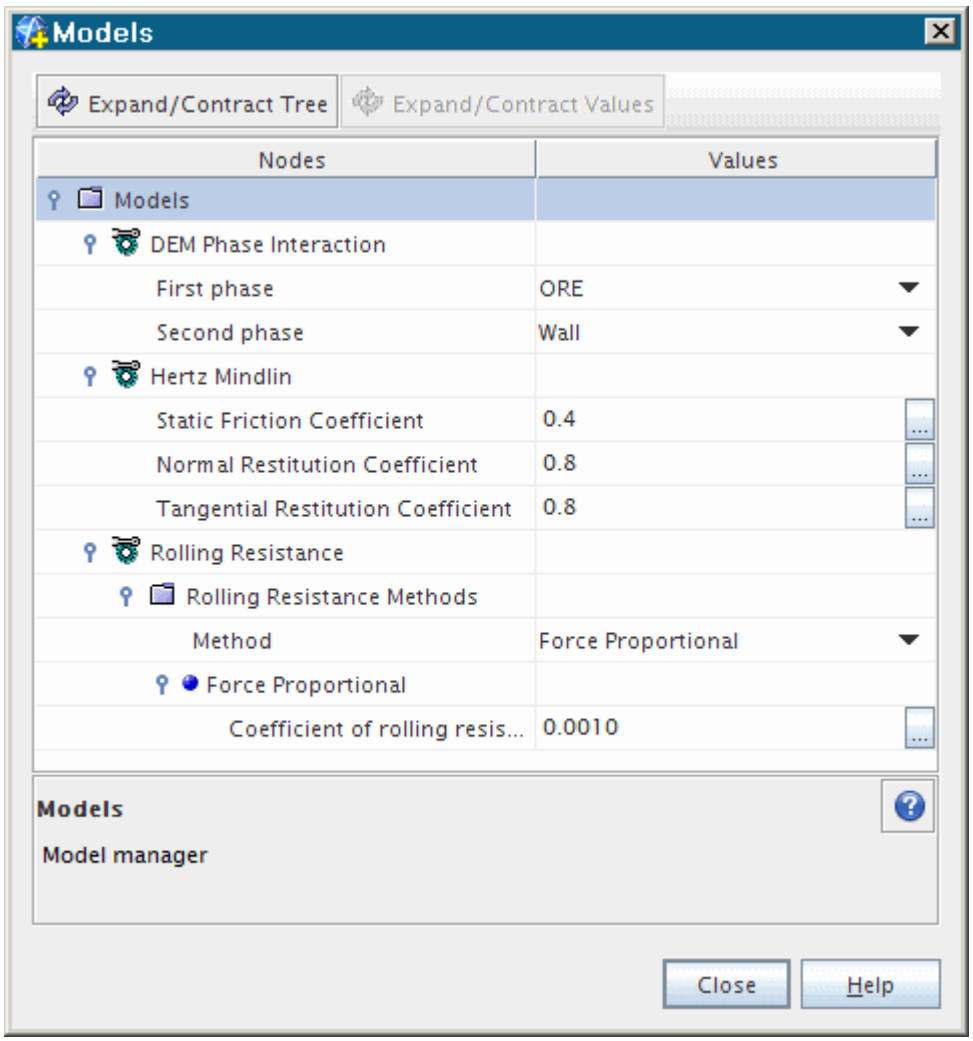


5. Rename the **Phase Interaction 1** node to `ORE-ORE`
6. Similarly, follow the above steps and create another phase interaction for the ORE to Wall interaction.

Group Box	Model
Phase Interaction Topology	**DEM Phase Interaction**
DEM Contact Models	**Hertz Mindlin** **Rolling Resistance** (selected automatically)

Node	Value
First Phase	**ORE**
Second Phase	**Wall**
Static Friction Coefficient	`0.4`
Normal Restitution Coefficient	`0.8`
Tangential Restitution Coefficient	`0.8`
Coefficient of Rolling Resistance	`0.0010`

The *Models* dialog looks as shown below



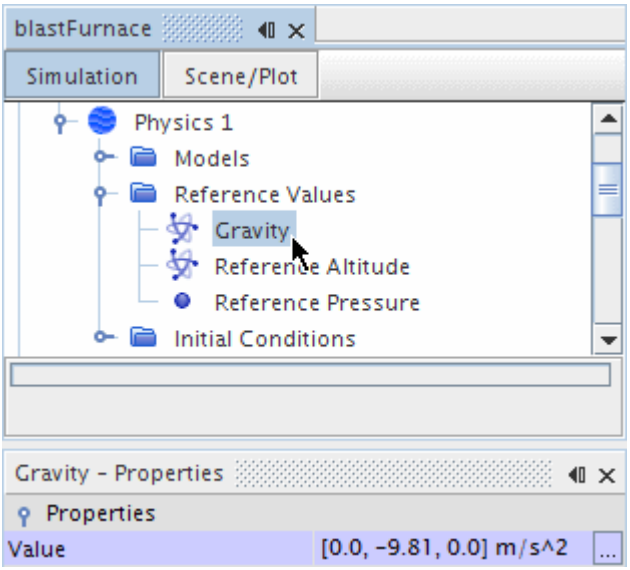
7. Rename the **Phase Interaction 2** node to `ORE-Wall`

Setting Reference Values

The reference values must be defined to make sure that the acceleration due to gravity acts in the expected direction.

To set the gravity reference value:

1. Select the **Continua** > **Physics 1** > **Reference Values** > **Gravity** node and set *Value* to `[0.0, -9.81, 0.0] m/s^2`.



Setting the Boundary Conditions

By default, the conditions for boundaries representing the furnace are all set to no-slip walls. You now change the **pln** boundary to **Outlet** with default **Split Ratio** of `1` to allow filling of the geometry with particles for the first 11.5 seconds of the simulation.

Use the following steps:

1. Navigate to the **Regions** > **Region** > **Boundaries** node.
2. Edit **blastFurnaceSurfaceMesh.pln** and set the following properties:

Node	Property	Setting
blastFurnaceSurfaceMesh.pln	**Type**	**Outlet**
└ **Physics Values**	**Split Ratio**	`1.0`

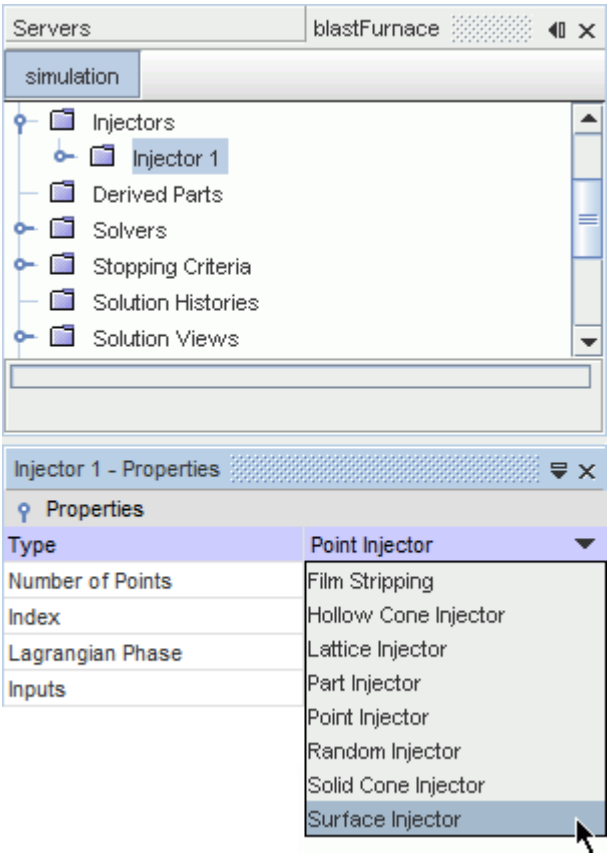
3. Save the simulation.

Creating an Injector

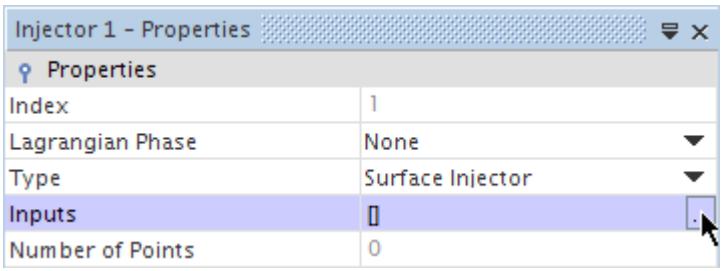
Particles are injected into the fluid region from the top of the blast furnace. The particles, 0.01 m in diameter, are injected at a mass flow rate of 18.33 kg/s with a velocity of 0.01 m/s in the negative Y-direction.

Create the particle injector:

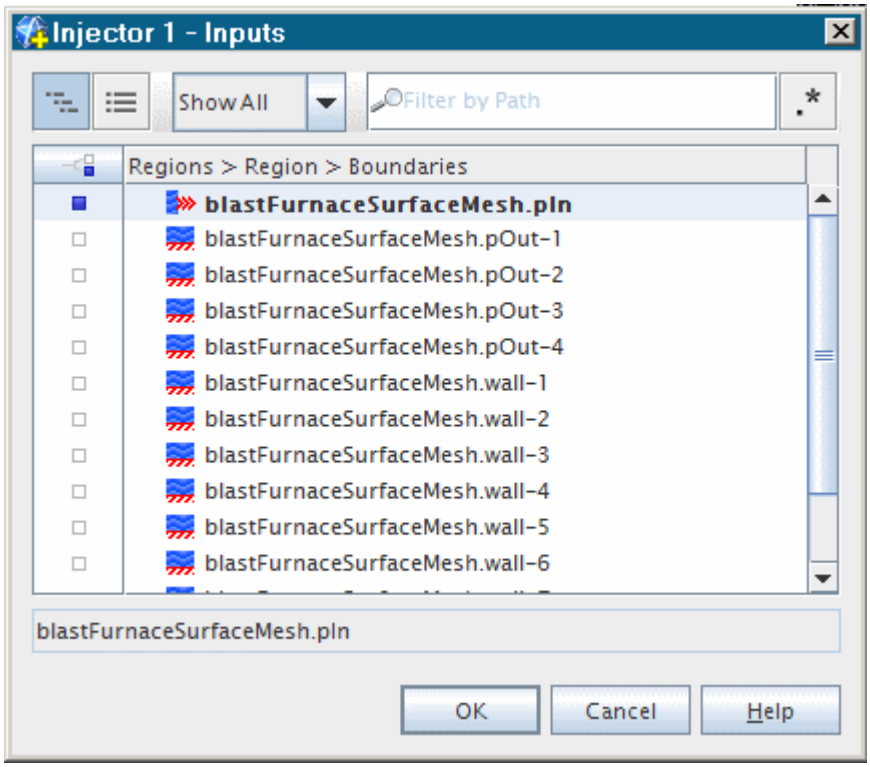
1. Right-click the **Injectors** node and select **New**.
2. Select the **Injector 1** node and set *Type* to **Surface Injector**.



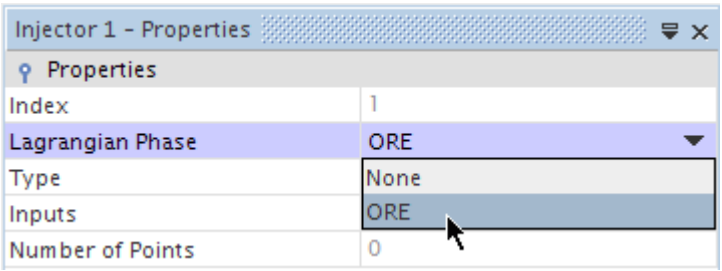
Now assign the injector to the **blaskFurnaceSurfaceMesh.pln** region of the blast furnace geometry and click the ellipsis next to the *Inputs* property.



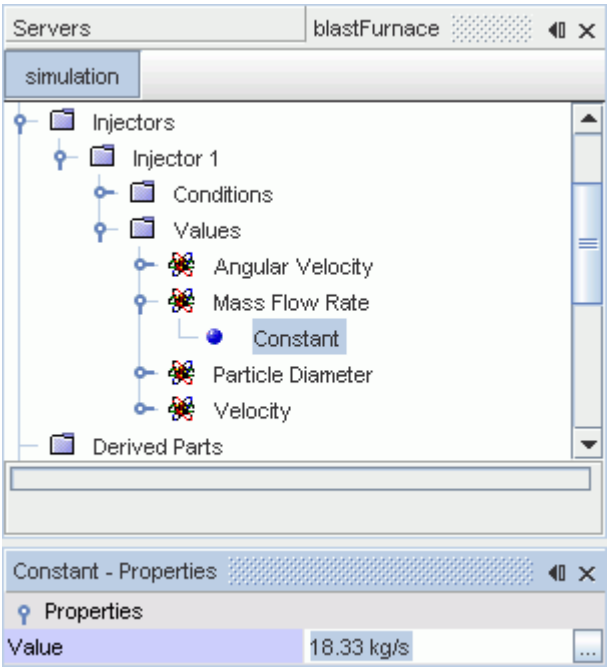
3. In the *Injector 1 - Inputs* dialog, select **blastFurnaceSurfaceMesh.pln** and click **OK**.



4. While Injector 1 is still selected, set the *Lagrangian Phase* to **ORE**.



5. Expand the **Injector 1** > **Values** node, and select the **Mass Flow Rate** > **Constant** node and set *Value* to `18.33 kg/s`.



6. Set the remaining properties as follows:

Property	Setting
Particle Diameter	0.01 m
Velocity	[0, -0.01, 0] m/s

7. Save the simulation.

Setting the Solver Parameters and Stopping Criteria

As the simulation is transient, the solver time-step and actual run time must be set. In this case, the time-step is 0.01 seconds, with a physical time of 11.5 seconds for the first stage of the simulation. Also as no fluid flow effect is taken into account, freeze the coupled implicit solver.

To define the solver parameters:

1. Expand the **Solvers** node and set the following properties:

Node	Property	Setting
Implicit Unsteady	*Time-Step*	`0.01 s`
Coupled Implicit	*Freeze Flow*	Activated

To define the stopping criteria:

2. Expand the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Inner Iterations	*Maximum Inner Iterations*	`1`

Node	Property	Setting
Maximum Physical Time	*Maximum Physical Time*	`11.5 s`
Maximum Steps	*Enabled*	Deactivated

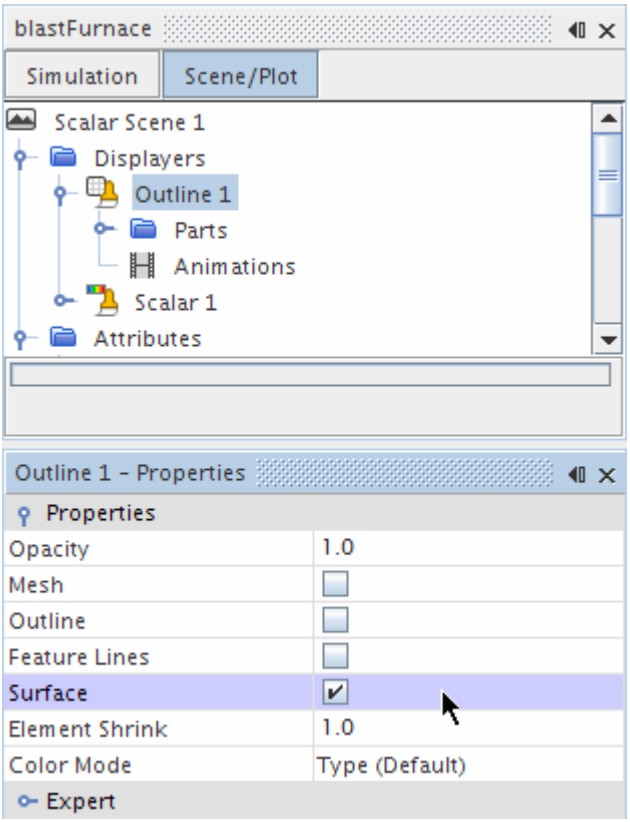
3. Save the simulation.

Setting Up a Visualization Scene

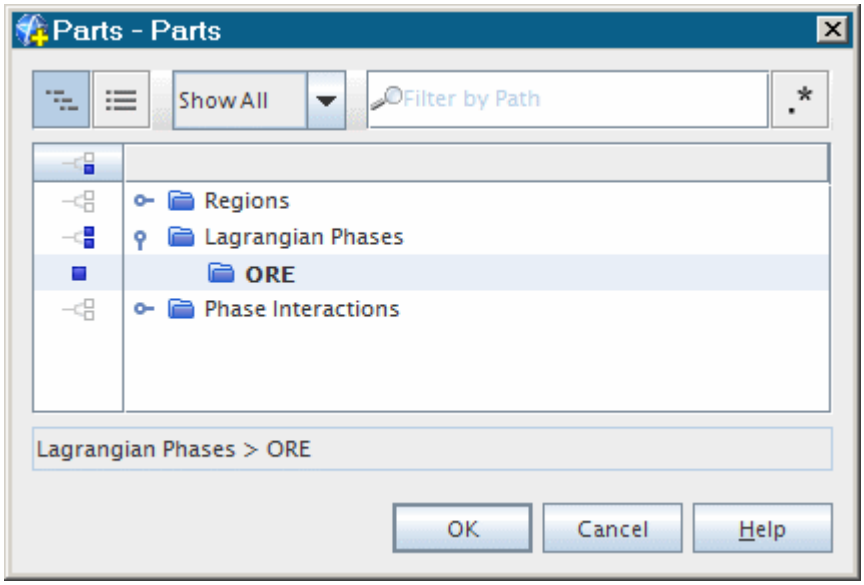
A scalar scene is created to provide a visual representation of the glass beads within the fluid system.

Use the following steps:

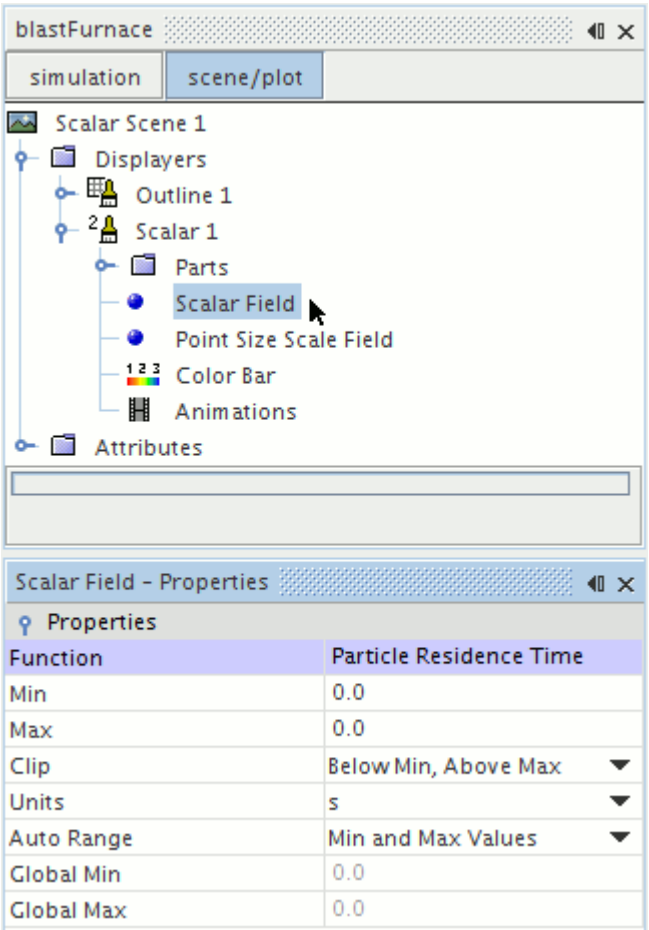
1. Right-click the **Scenes** node, and select **New Scene** > **Scalar**.
2. Click the **Scene/Plot** button above the object tree.
3. Select the **Outline 1** node, deactivate *Outline*, and activate *Surface*.



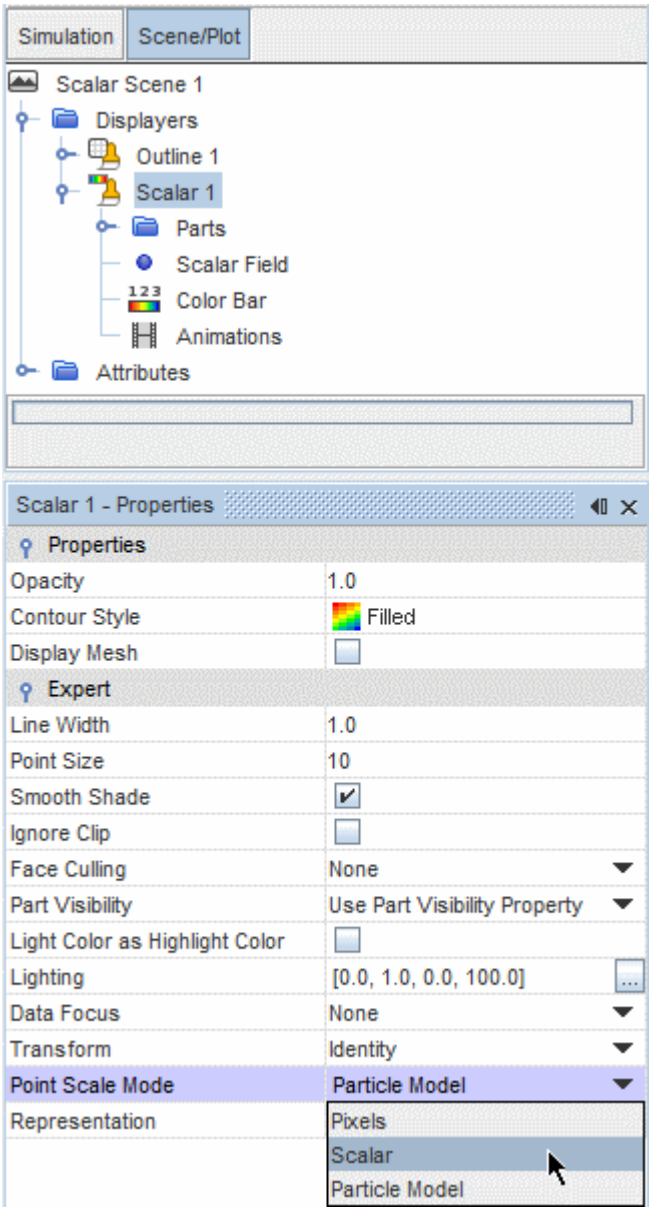
4. Edit the **Scalar 1** > **Parts** node.
5. In the *Parts - Parts* dialog, expand the **Lagrangian Phases** node, select **ORE**, and click **OK**.



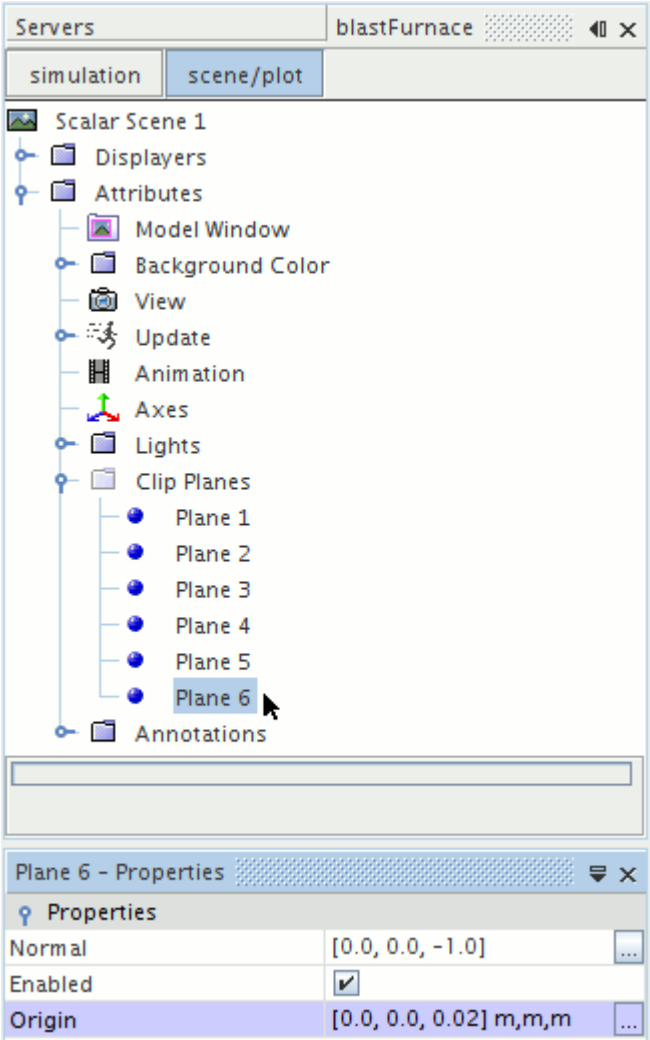
6. Select the **Scalar Field** node and set *Function* to **Particle** > **Particle Residence Time**.



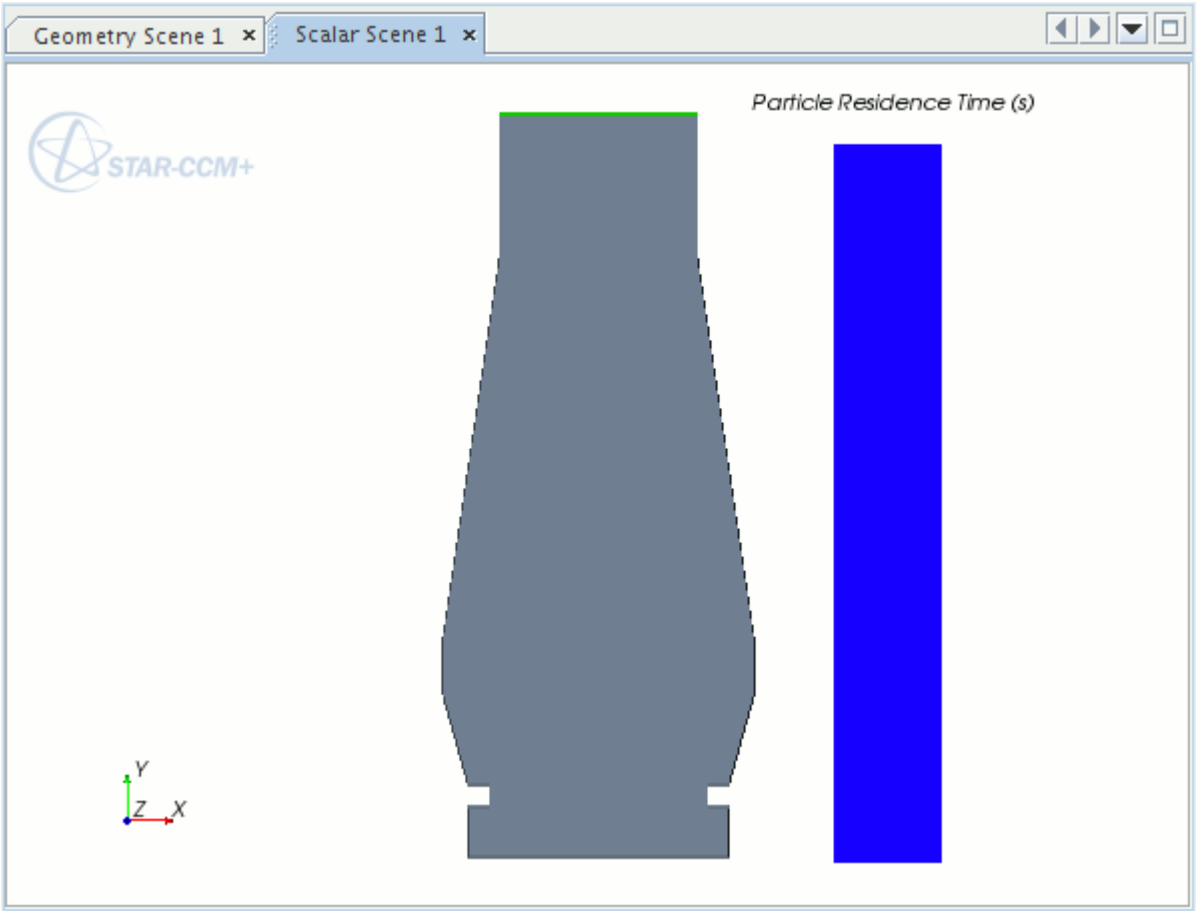
7. Select the **Scalar 1** node and set the *Point Scale Mode* to **Scalar**.



8. Select the **Attribute** > **Clip Planes** > **Plane 6** node and set the following properties:
   - Enabled: `Activated`
   - Origin: `[0.0, 0.0, 0.02]` m, m, m



9. Rearrange the color bar and display to provide the following view.



Running the Simulation

The simulation is now ready to be run.

Use the following steps:

1. Run the simulation by clicking (**Run**) in the **Solution** toolbar.

The iterative progress can be monitored in the *Output* window. The scalar scene displays the injection and movement of the glass beads within the fluid region as the solution proceeds.

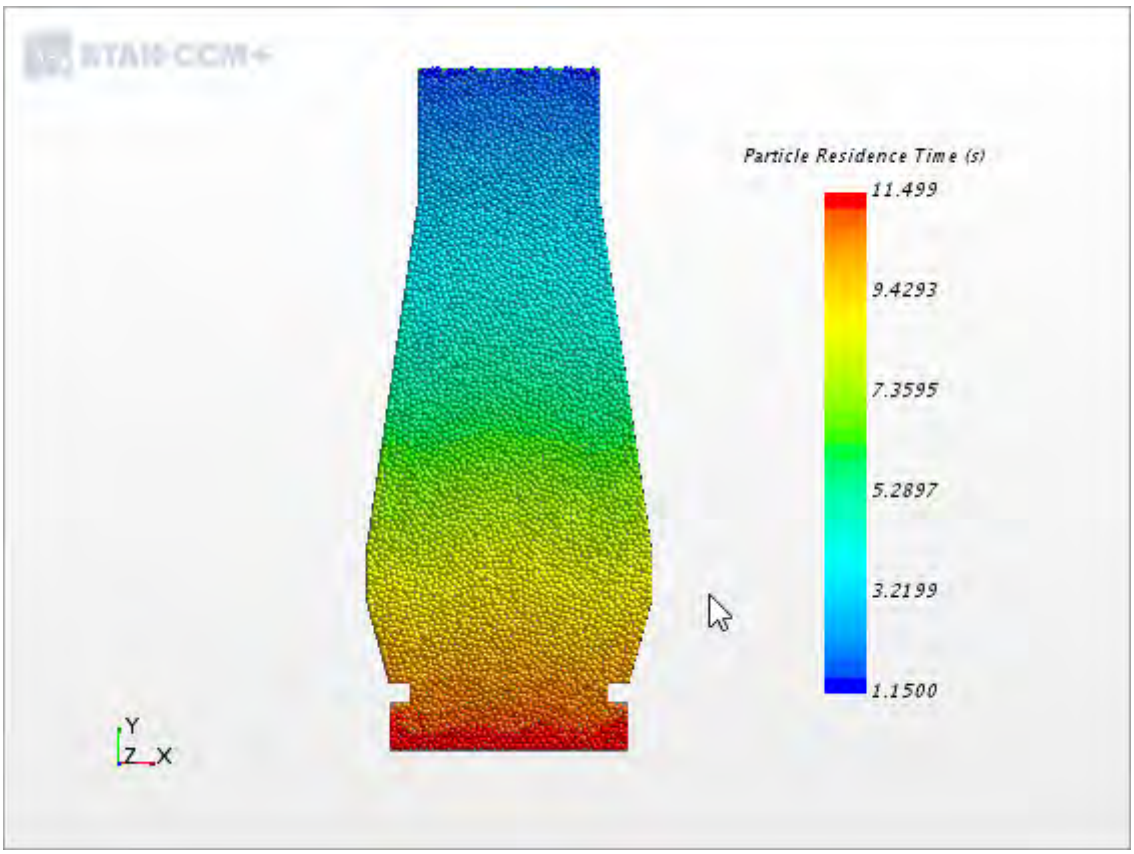
When the solution is complete, the following message is displayed in the *Output* window: `Stopping criterion Maximum Physical Time satisfied.`

2. Save the simulation.

Visualizing Results

When the solver is complete, the *Scalar Scene 1* in the *Graphics* window appears.

See below.



Stage Two of the Simulation

Having filled the furnace with beads in stage 1, it is now necessary to allow a period for the beads to settle before opening the outlets and allowing the beads to escape.

Change the **pIn** boundary to **Wall** to prevent the flow of particles into the furnace and to allow the glass beads to settle and come to rest within the furnace.

1. Select the **Regions** > **Region** > **Boundaries** > **blastFurnaceSurfaceMesh.pIn** node and set *Type* to **Wall**.
2. Select the **Stopping Criteria** > **Maximum Physical Time** node and set *Maximum Physical Time* to `13.5 s`.

Contents:

Running the Simulation
Visualizing Results

Running the Simulation

The simulation is now ready to be run.

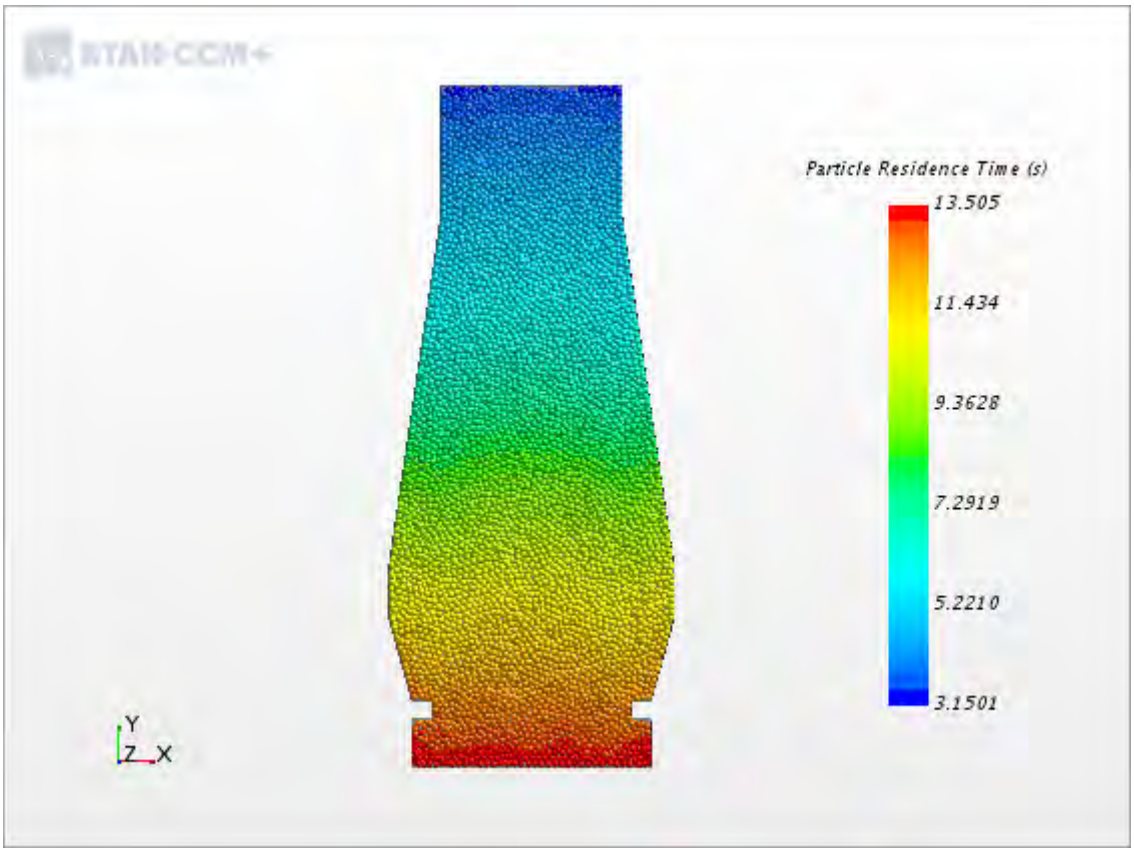
Use the following steps:

1. Run the simulation by clicking (**Run**) in the **Solution** toolbar.
2. Save the simulation.

Visualizing Results

When the solver is complete, the *Scalar Scene 1* in the *Graphics* window appears.

See below:



Stage Three of the Simulation

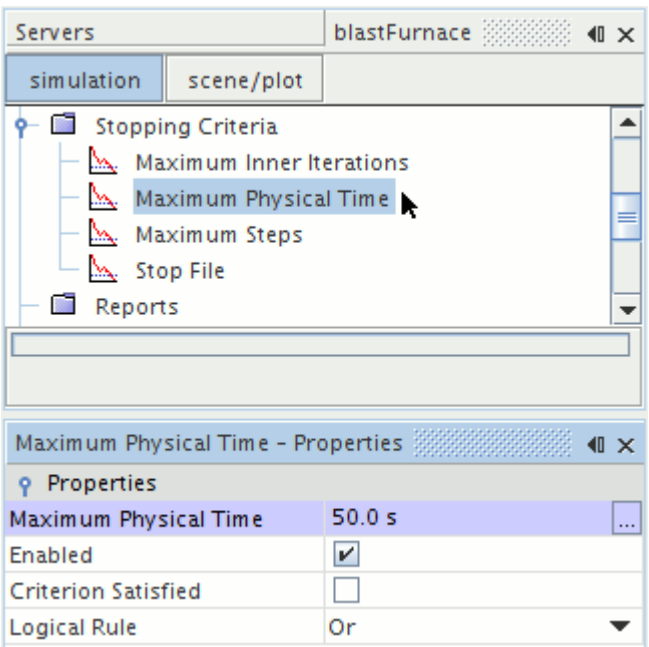
For this final stage of the simulation, open the outlets (one on each side) to allow the particles to discharge. Also, the inlet to allow new particles to enter the furnace to supplement the exiting ones already in the furnace.

Use the following steps:

1. Edit the **Regions** > **Region** > **Boundaries** node.
2. Set the following properties:

Node	Property	Setting
blastFurnaceSurfaceMesh.pIn	*Type*	**Outlet**
└ **Physics Values** > **Split Ratio**	*Split Ratio*	`1`
blastFurnaceSurfaceMesh.pOut-2	*Type*	**Velocity Inlet**
└ **Physics Values** > **Velocity Magnitude** > **Constant**	*Value*	`1e-4 m/s`
blastFurnaceSurfaceMesh.pOut-3	*Type*	**Velocity Inlet**
└ **Physics Values** > **Velocity Magnitude** > **Constant**	*Value*	`1e-4 m/s`

3. Select the **Stopping Criteria** > **Maximum Physical Time** node and set *Maximum Physical Time* to `50 s`.



Contents:

Running the Simulation
Visualizing Results

Running the Simulation

The simulation is now ready to be run.

Use the following steps:

1. Run the simulation by clicking (**Run**) in the **Solution** toolbar.
2. Save the simulation.

In rare occasions, clumps of particles block the flow of particles at an outlet. This causes the scene to show a skewed distribution of particles. If blocking occurs, use the following procedure:

3. Stop the simulation by clicking (**Stop**) in the **Solution** toolbar.
4. Close the simulation without saving.
5. Reload the last saved simulation.
6. Select the **Stopping Criteria** > **Maximum Physical Time** node and set *Maximum Physical Time* to `13.6 s`.
7. Select **Continua** > **Physics 1** > **Models** > **Multiphase Interactions** > **Phase Interactions** > **ORE-ORE** > **Models** > **Hertz-Mindlin** and set *Static Friction Coefficient* to `0`.
8. Run the simulation by clicking (**Run**) in the **Solution** toolbar.

   When the simulation stops, both outlets are unblocked.

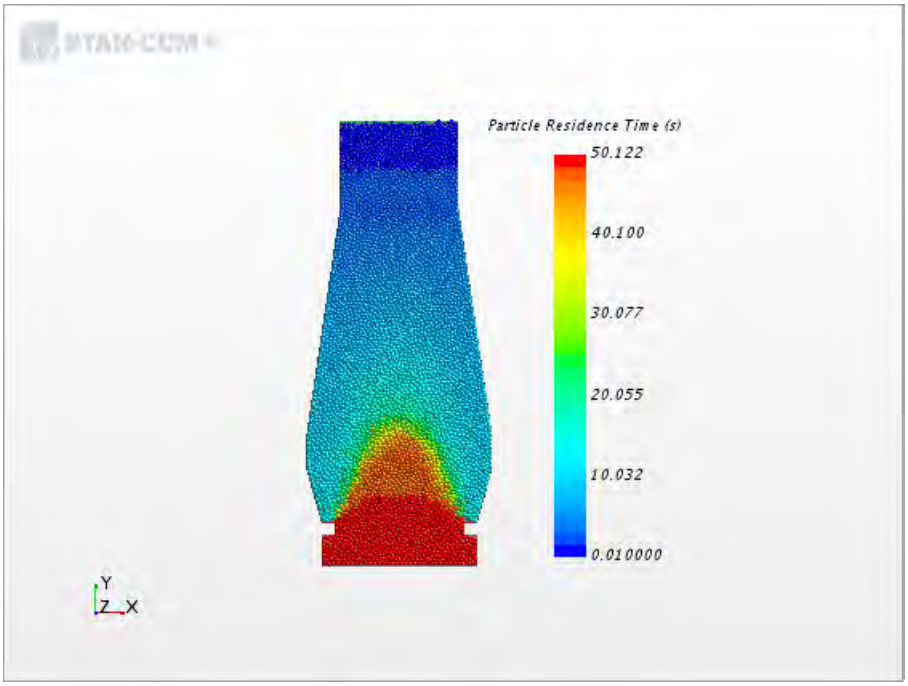
Return to solving the case as before the blockage occurred:

9. Select the **Stopping Criteria** > **Maximum Physical Time** node and set the *Maximum Physical Time* to `50 s`.
10. Select **Continua** > **Physics 1** > **Models** > **Multiphase Interactions** > **Phase Interactions** > **ORE-ORE** > **Models** > **Hertz-Mindlin** and set the *Static Friction Coefficient* to `0.4`.
11. Run the simulation by clicking (**Run**) in the **Solution** toolbar.
12. Save the simulation.

Visualizing Results

When the solver is complete, the *Scalar Scene 1* in the *Graphics* window appears.

See below:



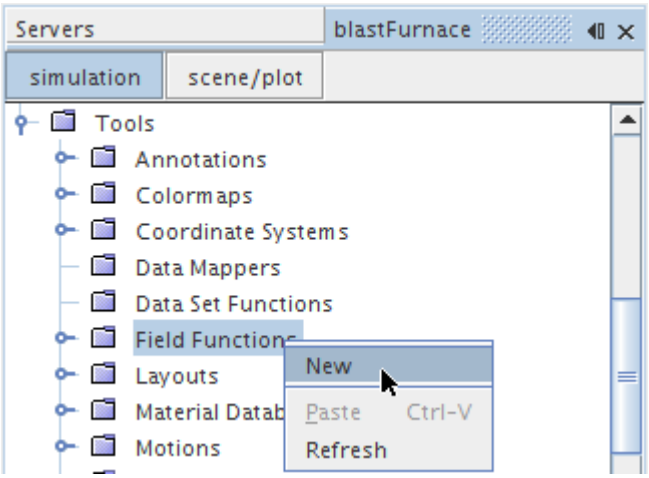
Particles that have residence time of more than 36.5 s constitute the deadman region. In this region, particles move rather slowly and hence have high residence times.

Comparing the Results

In this section, compare the height and shape of the deadman region with the experimental image of Zhou *et al*. To show the deadman region more clearly, you create the user-defined field function, *oldParticles*, which displays the particles with residence times of less than 36.5 s all in the same color.

To define the region in a field function:

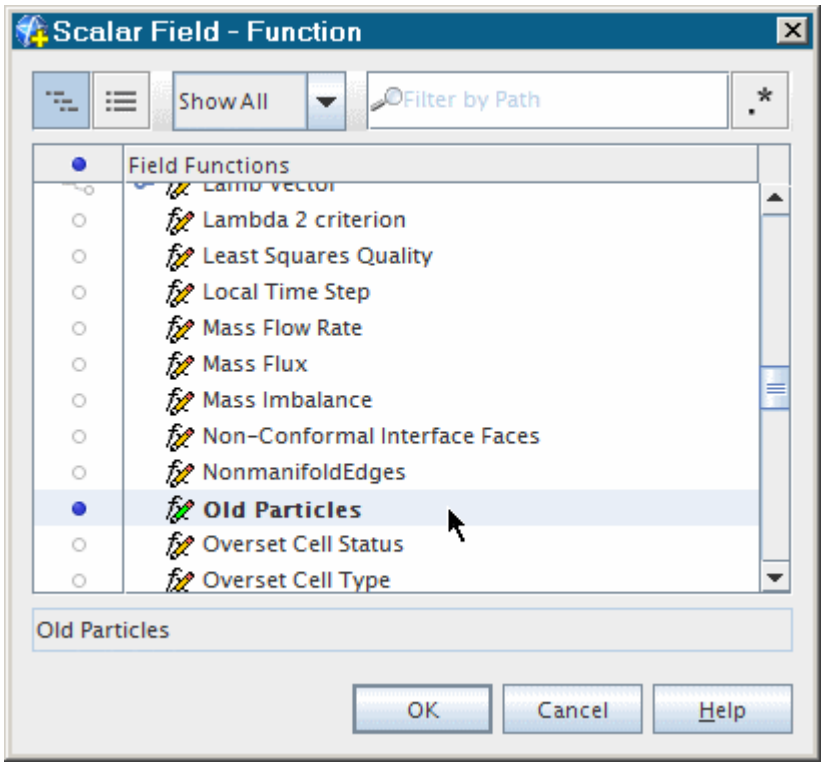
1. Open the **Tools** node at the bottom of the main object tree. Right-click the **Field Functions** node and select **New** > **Scalar**.



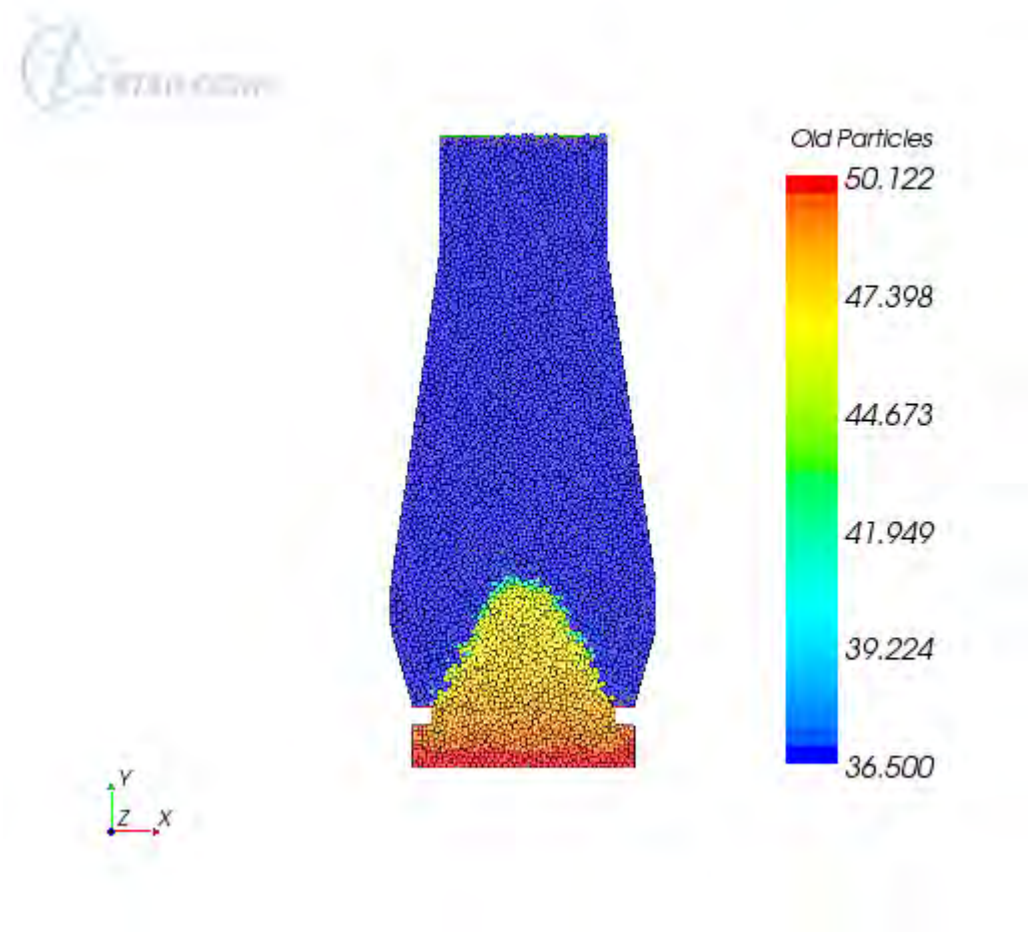
2. Rename the **User Field Function 1** node to `Old Particles`
3. Select the **Old Particles** node and set the following properties:

Property	Setting
Function Name	`OldParticles`
Definition	`max(0,$ParticleResidenceTime - ($Time-13.5)) + ($Time-13.5)`

4. Select the **Scenes** > **Scalar Scene 1** > **Displayers** > **Scalar 1**> **Scalar Field** node and set the *Function* to **Old Particles**.

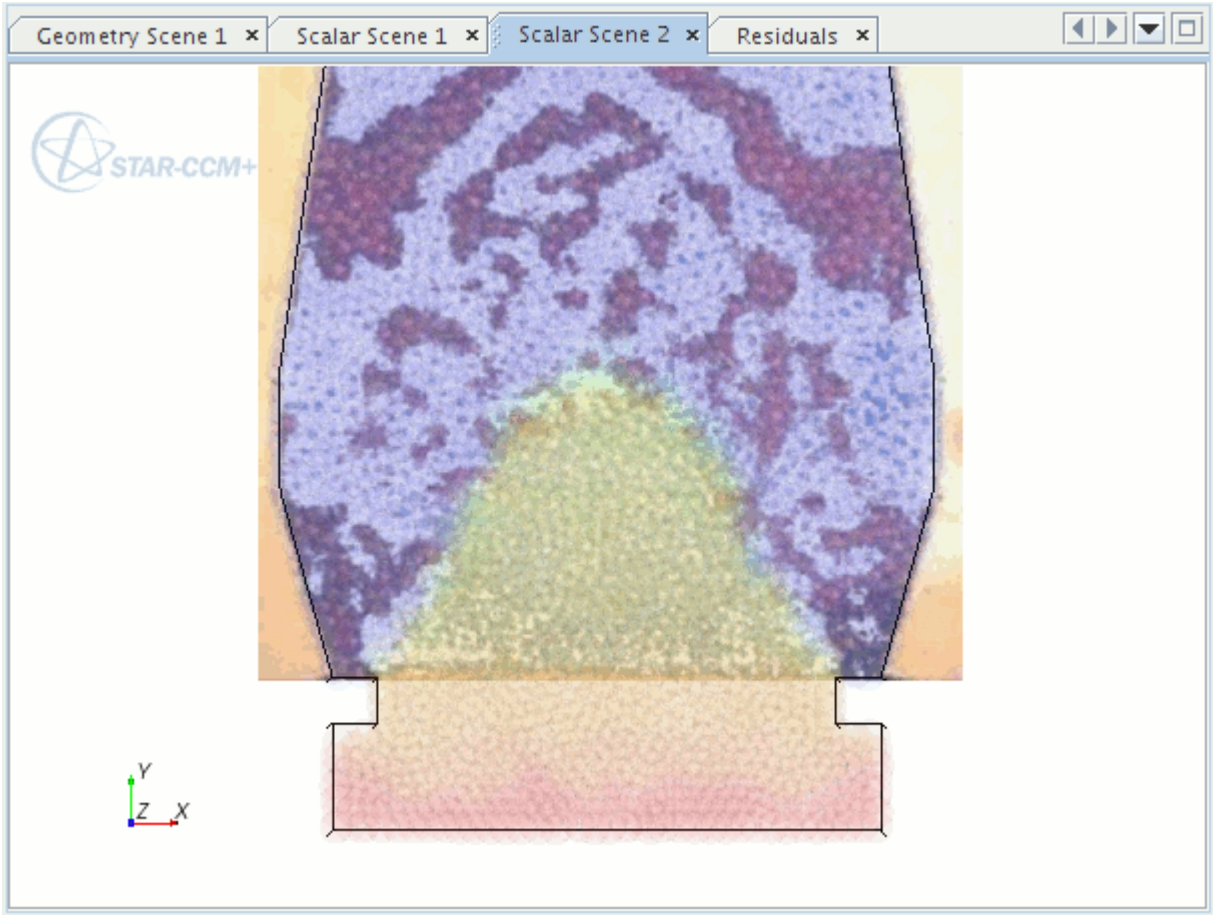


The *Scalar Scene 1* in the *Graphics* window appears as shown below:



Deep blue colored particles are newer particles, injected after time t=13.5 s. The remaining particles are the ones that had come into the domain before 13.5 s.

The following screenshot shows the comparison, with the experimental image superimposed on top of the Old Particles scene above and zoomed in using the mouse buttons:



The deadman size, height, and shape are well-predicted.

Summary

This tutorial simulated a three-dimensional solid flow through a model blast furnace.

This tutorial has introduced the following STAR-CCM+ features:

- Selecting fluid physics models.
- Definition of Lagrangian phases.
- Definition of DEM particle interaction.
- Creation of a particle injector.
- Setting solver and stopping criteria.
- Setting up visualization scenes.
- Running the simulation.
- Visualizing the results.

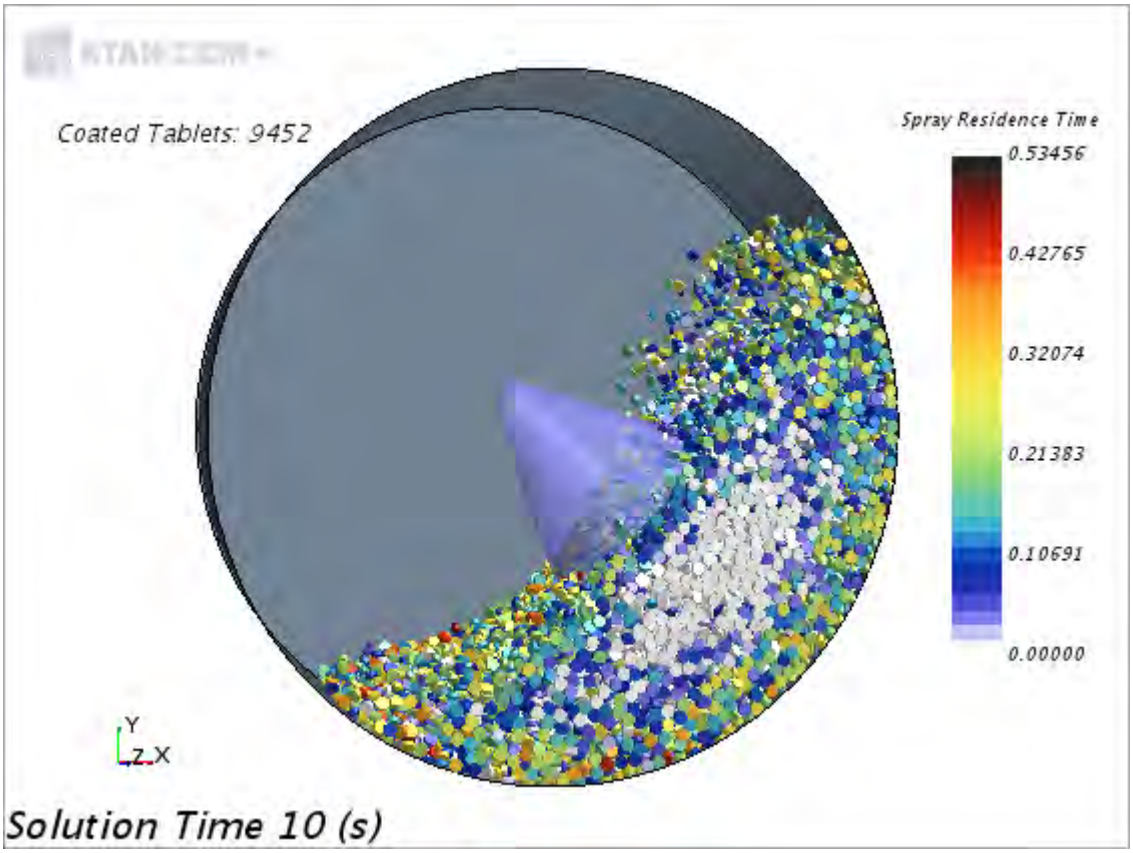
Blast Furnace Tutorial Bibliography

Zhou, Z., Zhu, H., Yu, A., Wright, B., Pinson, D. and Zulli, P. 2005, “Discrete Particle Simulation of Solid Flow in a Model Blast Furnace.” *ISIJ International* , vol 45, no 12, pp 1828 – 1837.

Cylindrical Particles in a Rotating Drum

This tutorial simulates a large number of cylindrical-shaped tablets tumbling in a rotating drum and being sprayed with a protective coating.

The tablets are simulated by cylindrical DEM particles. The tablets are injected into the rotating drum and, once they have settled, their residence time within the conical spray is tracked. The complexity of the coating process is avoided by assuming that all tablets are coated in a uniform manner when within the spray, regardless of their angle relative to the jet. The transfer of the coating to, or from, tablets and mixer walls is ignored, as is the effect of the spray jet and drag on the tablets due to the air in the drum. However, the impact between tablets and drum-walls (including frictional effects), is included via a set of phase-interactions.



This tutorial consists of two simulation runs. In the first run, the tablet particles are injected into the drum, which rotates for three seconds (approximately one full drum reveolution) to settle the particles. In the second run, the passive scalar is activated and the drum rotates for a further seven seconds to track the coating process.

Note that, although the laminar flow solver is selected in this example, the fluid flow solver is switched off (frozen). Thus, the air in the drum is assumed to be still and so to have negligible effect on the movement of the tablets.

Contents:

Prerequisites
Load the Base Simulation
Selecting the Physics Models
Defining Lagrangian Phases
Defining DEM Particle Interactions
Setting the Rotating Drum Wall Condition
Creating an Injector
Setting the Solver Parameters and Stopping Criteria
Setting Up a Visualization Scene
Running the Injection Simulation
Visualizing the Results of the Injection
Initiate the Passive Scalar Spray
Setting the Stopping Criteria and the Solution History File
Running the Coating Spray Simulation
Summary

Prerequisites

The instructions in this tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

To complete this tutorial, familiarize yourself with the following:

Techniques	Associated Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+
Creating .simh and animation files	Simulation History: Lagrangian DEM on an Auger
	Solution Recording and Playback: Vortex Shedding

Load the Base Simulation

A base simulation file has been prepared for this simulation. Continue by loading the base simulation file and saving it under a different file name.

1. Create a directory for the tutorial called `rotatingdrum`.
2. In this directory, copy the file containing the problem geometry (`rotatingDrumBase.sim`) from the `doc/startutorialsdata/discreteElementMethod/data` subdirectory of your STAR-CCM+ installation directory.
3. Start up STAR-CCM+ in a manner that is appropriate to your working environment.
4. Click (**Load Simulation**) from the tool bar and open `rotatingDrumBase.sim`.
5. Save the new simulation to disk under file name `rotatingDrum.sim`.

Selecting the Physics Models

Select the physics models required to define the behavior of the multiphase continuum.

To set the physics models:

1. For the physics continuum **Continua** > **Physics DEM**, select the following models, in order:

Group Box	Model
Space	**Three-Dimensional** (preselected)
Time	**Implicit Unsteady**
Material	**Gas**
Flow	**Coupled Flow**
Enabled Models	**Gradients** (selected automatically)
Equation of State	**Constant Density**
Viscous Regime	**Laminar**
Optional Models	**Gravity**
	Lagrangian Multiphase
	Discrete Element Model (DEM)
	Multiphase Interaction (selected automatically)

2. Click **Close**.

By default, gravity acts in the -Z direction. Change this direction to -Y:

3. Select the **Physics DEM** > **Reference Values** > **Gravity** node and set *Value* to `[0.0, -9.81, 0.0] m/s^2`.
4. Save the simulation.

Defining Lagrangian Phases

A Lagrangian phase is used to define the solid particles in the simulation. The phase requires specific models and properties to characterize its behavior. In this case, the solid particles are cylindrical tablets, with the material properties set by the user. In a later step, you define interactions between tablets and the mixer walls and between other tablets.

To define the Lagrangian phase:

1. Right-click the **Lagrangian Multiphase** > **Lagrangian Phases** node and select **New**.
2. Rename the phase as `tablet`.
3. Right-click the **tablet** > **Models** node, click **Select Models**, and select the following models in order:

Group Box	Model
Particle Type	**DEM Particles**

Group Box	Model
Optional Particle Forces	Deactivate **Pressure Gradient Force** (selected automatically)
Particle Shape	**Cylindrical Particles**
Material	**Solid**
Equation of State	**Constant Density**
Optional Models	**Passive Scalar**

4. Click **Close**.
5. Expand the **Lagrangian Multiphase** > **Lagrangian Phases** > **tablet** > **Models** > **Solid** node.
6. Select the **Rubber** node and rename it as `tablet`.
7. Expand the **tablet** > **Material Properties** > **Density** node.
8. Select the **Constant** node and set *Value* to `1600.0 kg/m^3`.
9. Right-click the **Passive Scalar** > **Passive Scalars** node, and select **New**.
10. Rename the newly created **Passive Scalar** sub-node `Spray Residence Time`.

Note: Although defined, the passive scalar remains inactive in this first phase of the tutorial.

A default boundary interaction is specified for all wall boundaries in the Lagrangian phase itself. For details, see the node **Lagrangian Multiphase** > **Lagrangian Phases** > **tablet** > **Boundary Conditions** > **Wall**.

Defining DEM Particle Interactions

The DEM particle interactions define how the particles behave when they collide with one another and with the drum walls.

To define the tablet-to-tablet interaction:

1. Right-click the **Physics DEM** > **Models** > **Multiphase Interaction** > **Phase Interactions** node and select **New**.
2. Rename the node **Phase Interaction 1** to `tablet-tablet`.

Specify the models and properties for the tablet-tablet interaction:

3. Right-click the **tablet-tablet** > **Models** node and select the following models, in order:

Group Box	Model
Phase Interaction Topology	**DEM Phase Interaction**
Optional Models	**Linear Spring**
	Deactivate **Rolling Resistance** (in the *Enabled Models* group box)

4. Click **Close**.
5. Right-click the **tablet-tablet** interaction node and select **Copy** (or use **Ctrl-C**).

6. Right-click the **Phase Interactions** node and select **Paste** (or use **Ctrl-V**). Repeat this this operation one further time.

   This action creates two copies of the **tablet-tablet** interaction node.

7. Rename the copies to `tablet-side` and `tablet-front & back`.

Specify the models and properties for the interaction between tablets:

8. Edit the **tablet-tablet** > **Models** node and set the following model properties:

Node	Property	Setting
DEM Phase Interaction	*First Phase*	**tablet**
	Second Phase	**tablet**
Linear Spring	*Static Friction Coefficient*	`0.4`
	Normal Restitution Coefficient	`0.1`
	Tangential Restitution Coefficient	`0.1`
└Normal Spring Stiffness Method > **Constant**	*Value*	`10000`
└Tangential Spring Stiffness Method > **Constant**	*Value*	`10000`

In this tutorial, different frictional properties are required on the side walls of the cylinder from the front and back walls. To expose the side wall boundary as an item in the *Second Phase* drop-down list, you must set the **Lagrangian Specification** on the side wall boundary itself to **Specify for Boundary**:

9. Select the **Regions** > **mixer** > **Boundaries** > **side** > **Physics Conditions** > **Lagrangian Specification** node and set **Method** to **Specify for Boundary**.

   A new sub-node **Phase Conditions** appears under the boundary node.

Return to setting up the phase interactions within the **Physics DEM** continuum.

Within the **tablet-side** interaction, specify the models and properties for the interaction between tablets and the cylindrical wall of the drum:

10. Edit the **tablet-side** > **Models** node and set the following model properties:

Node	Property	Setting
DEM Phase Interaction	*First Phase*	**tablet**
	Second Phase	**mixer: side**
Linear Spring	*Static Friction Coefficient*	`0.9`
	Normal Restitution Coefficient	`0.1`
	Tangential Restitution Coefficient	`0.1`
└Normal Spring Stiffness Method > **Constant**	*Value*	`10000`

Node	Property	Setting
└ Tangential Spring Stiffness Method > **Constant**	*Value*	`10000`

Specify the models and properties for the interaction between tablets and the back and front of the drum:

11. Edit the **tablet-front & back** > **Models** node and set the following properties:

Node	Property	Setting
DEM Phase Interaction	*First Phase*	**tablet**
	Second Phase	**Wall** (the global wall boundary condition for the Lagrangian phase)
Linear Spring	*Static Friction Coefficient*	`0.0`
	Normal Restitution Coefficient	`0.6`
	Tangential Restitution Coefficient	`0.6`
└ Normal Spring Stiffness Method > **Constant**	*Value*	`10000`
└ Tangential Spring Stiffness Method > **Constant**	*Value*	`10000`

12. Save the simulation.

Setting the Rotating Drum Wall Condition

Define the rotational velocity of the drum wall.

1. Select **Tools** > **Motions** > **Rotation** and set the following properties:

Property	Value
Axis Direction	`[0.0, 0.0, 1.0]`
Axis Origin	`[0.0, 0.0, 0.0] m`
Rotation Rate	`20.0 rpm`
Coordinate System	**Laboratory**

2. Select the **Regions** > **mixer** > **Physics Values** > **Motion Specification** node and set *Motion* to **Rotation**.
3. Save the simulation.

Creating an Injector

Create an injector to deliver the tablets into the drum at the beginning of the simulation.

1. Right-click the **Injectors** node and select **New**.
   This creates the **Injector 1** node.
2. Select the **Injector 1** node and set the *Type* to **Random Injector**.
3. Set *Lagrangian Phase* to **tablet**.
4. Set *Inputs* to **mixer** (region).
5. Edit the **Injector 1** node and set the following properties:

Node	Property	Setting
Conditions		
└ **Particle Packing Specification**	*Method*	**Particle Count**
Values		
└ **Number of Particles**	*Value*	`24000.0`
└ **Cylinder Height**	*Method*	**Constant**
	Value	`0.0132 m`
└ **Cylinder Radius**	*Method*	**Constant**
	Value	`0.01825 m`
└ **Number of Seeds**	*Value*	`288000`
└ **Particle orientation**	*Method*	**Uniformly distributed**
└ **Velocity**	*Method*	**Constant**
	Value	`[0.0, -2.0, 0.0] m/s`

6. Save the simulation.

Setting the Solver Parameters and Stopping Criteria

Set the solver time-step and elapsed time.

1. To define the solver parameters, edit the **Solvers** node and set the following properties:

Solver Node	Property	Value
Implicit Unsteady	*Time-Step*	`0.01 s`
Lagrangian Multiphase > **Load Balancing**	*Enabled*	Activated
DEM Solver	*Time Scale*	`1.0`

Solver Node	Property	Value
Coupled Implicit	*Freeze Flow*	Activated

2. To define the stopping criteria, edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Inner Iterations	*Maximum Inner Iterations*	`1`
Maximum Physical Time	*Maximum Physical Time*	`3.0 s`
Maximum Steps	*Enabled*	Deactivated

3. Save the simulation.

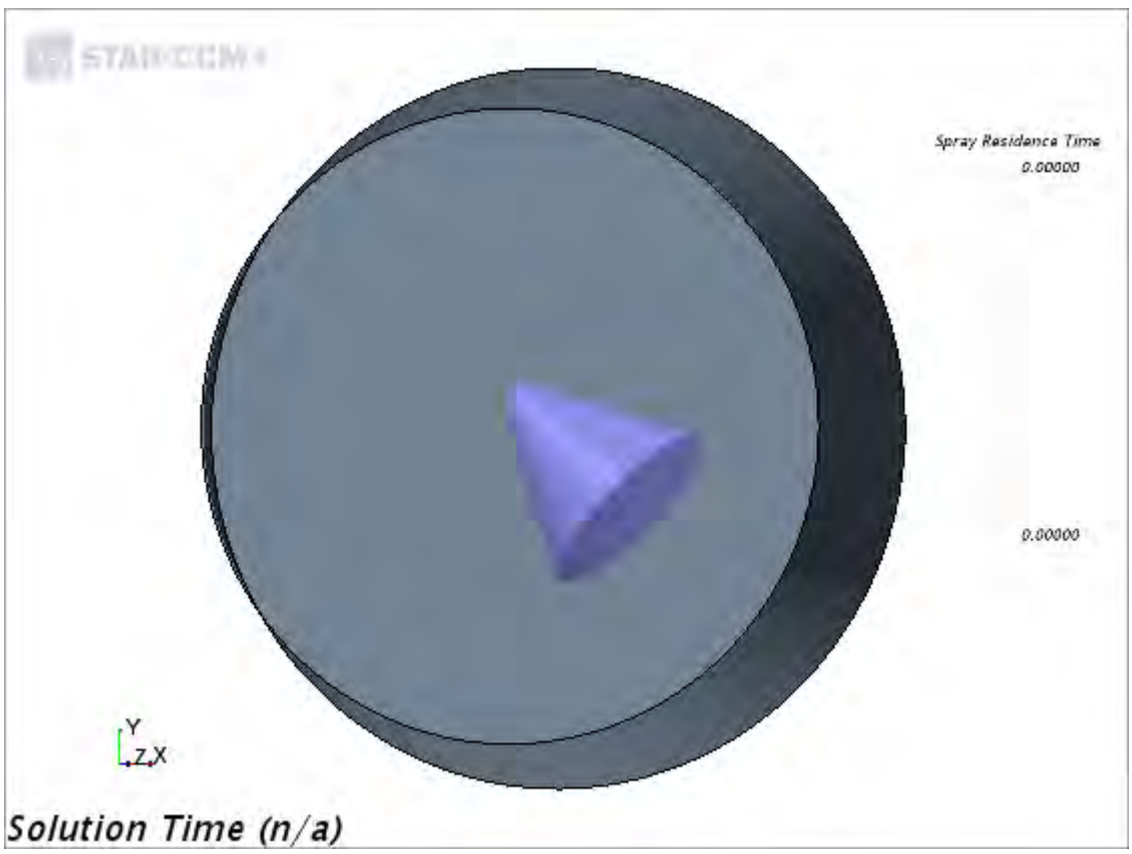
Setting Up a Visualization Scene

Set up the Scalar Drum Scene in preparation for display of the tablets.

To view the pre-defined scene:

1. Right-click the **Scenes** > **Scalar Scene Drum** node and select **Open**.

   A view similar to the one shown below appears (note that the tablets have yet to be injected).



2. Select **Scalar Scene Drum** > **Displayers** > **Tablets** > **Scalar Fields** and set *Function* to **Particle: Spray Residence Time**.

The "Spray Residence Time" caption is added to the scene.

3. Select **Scenes** > **Scalar Scene Drum** > **Displayers** > **Tablets** > **Parts** and set *Parts* to **Lagrangian Phases** > **tablet**.

To save results images to files every 0.04 seconds:

4. Select **Scalar Scene Drum** > **Attributes** > **Update** > **Time-Step Frequency** and set *Frequency* to `4`.

For the final scene, a report annotation provides a count of coated particles. To supply the input part for this report, you create a derived part based on the threshold value of the scalar *Spray Residence Time*.

To create the threshold part:

5. Right-click **Derived Parts** and select **New Part** > **Threshold...**
6. In the *Edit* dialog:
   a) Set **Input Parts** to **Lagrangian Phases** > **tablet**.
   b) Set **Scalar** to **Particle** > **Spray Residence Time**.
   c) Set **Extraction Mode** to **All Above** and ensure the **Threshold Value** is set to `0.0`.
   d) Set to **No Displayer**.
7. Click **Create** and **Close**.
8. Rename the **Threshold** node to `Color Sprayed Tablets`.
9. Select the **Reports** > **SprayedTablets** node and set *Parts* to **Derived Parts** > **Color Sprayed Tablets**.
10. Save the simulation.

Running the Injection Simulation

The simulation is now ready to be run.

Use the following steps:

1. Run the simulation by clicking (**Run**) in the **Solution** toolbar.

The iterative progress can be monitored in the *Output* window. The scalar scene displays the injection and movement of the cylindrical-shaped tablets within the fluid region as the solution proceeds.

When the solution is complete, the following message is displayed in the *Output* window: `Stopping criterion Maximum Physical Time satisfied.`

2. Save the simulation.

The simulation injects 24,000 cylindrical particles (tablets) and rotates the drum for three seconds to settle them into position. The movement of the tablets can be viewed as the solution progresses in the *Scalar Scene Drum* window.

Running the simulation takes approximately 30 minutes on a typical workstation with eight processors running in parallel.

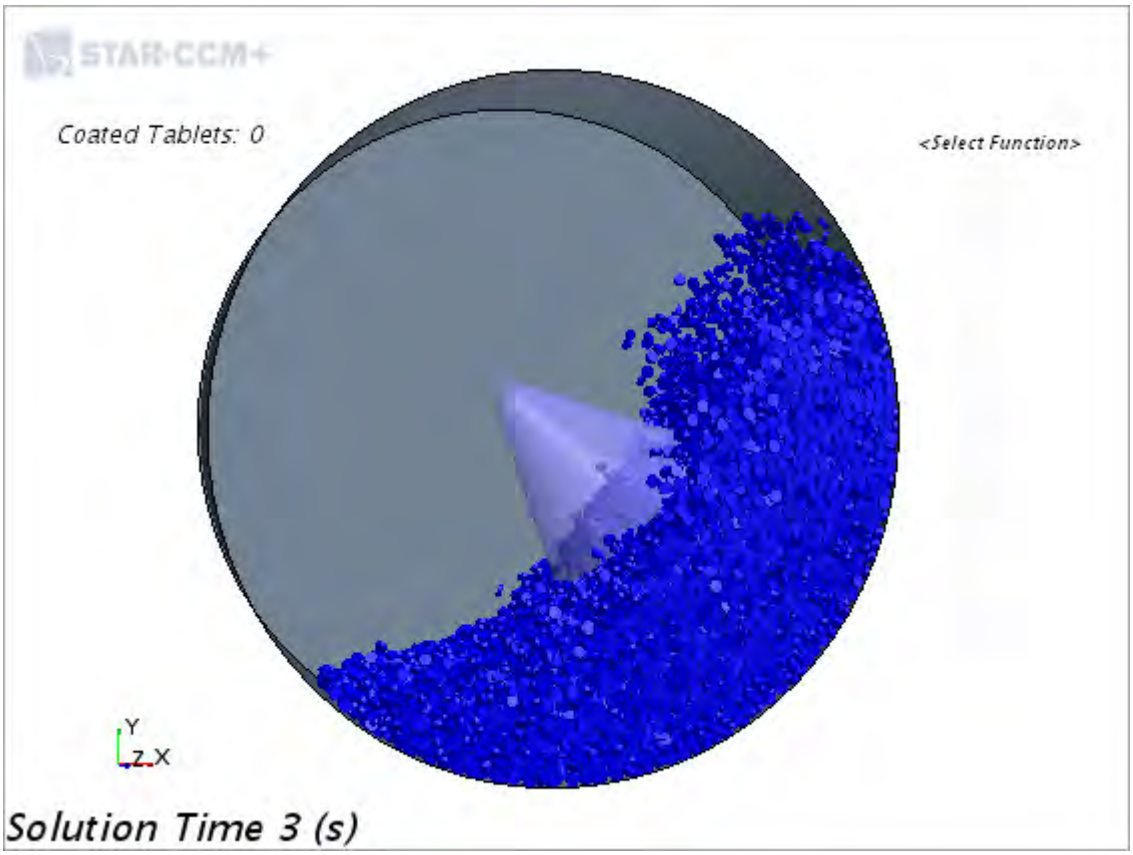
Visualizing the Results of the Injection

View the results of the initial simulation.

The most recent result is displayed in the scalar scene window and a set image files has been saved to the directory. An animation of the particles dropping into the revolving drum can be produced from these images. Another approach that can be used is to set up a simulation history file, from which a solution view and animation can be created within STAR-CCM+. Refer to Simulation History: Lagrangian DEM on an Auger for more details on how to do this.

The result of the first simulation run appears in the *Visualization* window of *Scalar Scene Drum* and should look similar to the following image.

Note: The result will not produce exactly the same image owing to the random injection algorithm, but the angle of repose should be similar.



The scene shows the cylindrical particles in the rotating drum, after three seconds of rotation. The tablets have reached an equilibrium position.

Initiate the Passive Scalar Spray

Set up and activate the passive scalar used to represent the amount of coating that is sprayed onto the tablets.

The passive scalar was set up previously, but must now be activated so that it tracks the tablet residence time within the conical spray.

1. Select the **Continua** > **Physics DEM** > **Models** > **Lagrangian Multiphase** > **Lagrangian Phases** > **tablet** > **Models** > **Passive Scalar** node and set *User Source Option* to **Scalar rate with inferred density**.

   STAR-CCM+ adds the **Composite** node and its accompanying **Spray Residence Time** sub-node (based on the name of the Passive Scalar already set previously).
2. Select the **Passive Scalar** > **Passive Scalar Source Density Inferred** node and set *Method* to **Composite**.
3. Select the **Composite** > **Spray Residence Time** node and set the following properties:

Property	Setting
Method	**Field Function**
Scalar Function	**zoneSprayZone_Particles**

   The field function **zoneSprayZone_Particles** has been specially prepared for this tutorial. **zoneSprayZone_Particles** specifies that the passive scalar is applied to tablets that pass through the conical spray volume:

```
($$ParcelCentroid("CylSprayZone")[2] > 0.0 &&
$$ParcelCentroid("CylSprayZone")[2] < 0.49517673612559787 &&
$$ParcelCentroid("CylSprayZone")[0] < 0.05+0.5048702448262621*
$$ParcelCentroid("CylSprayZone")[2] ) ? 1.0 : 0.0
```

   A tablet receives spray for as long as its centroid is within the conical volume. The tablet can enter the volume multiple times. The passive scalar value increases for each time-step in which the tablet remains within the spray volume.

Setting the Stopping Criteria and the Solution History File

Resume the simulation and set the maximum time to ten seconds. Create a solution history file to record the particle (tablet) velocities and the Spray Residence Time

1. Select the **Stopping Criteria** > **Maximum Physical Time** node and set *Maximum Physical Time* to `10.0 s`.
2. Right-click **Solution Histories** and select **New**.
3. Enter `rotatingDrumData` as the name of the solution history file and click **Save**.

   A new node containing the name of the solution history file is added below the **Solution Histories** node. The red asterisk next to this node means that data is actively written to the file when the simulation runs.
4. To choose what data to write to the solution history file, and the frequency of writing it, edit the **Solution Histories** > **rotatingDrumData** node and set the following properties:

Node	Property	Setting
rotatingDrumData	*Auto-record*	Activated
	Export for reporting	Activated

Node	Property	Setting
	Regions	To avoid writing the mesh each time-step, remove the **mixer** region.
	Inputs	**Lagrangian Phases** > **tablet**
	Functions	**Particle** > **[Angular Momentum...Velocity]** > **Spray Residence Time**
		Particle > **[Angular Momentum...Velocity]** > **Velocity** > **Magnitude**
		Particle > **[Angular Momentum...Velocity]** > **Velocity**
└ **Update**	*Trigger*	**Time Step**
└ **Time-Step Frequency**	*Frequency*	`1`

5. Save the simulation.

Running the Coating Spray Simulation

The simulation is now ready to be run to determine the residence time of the tablets within the spray zone.

First, update the scalar scene so that residence time is shown for the tablets:

1. Select **Scenes** > **Scalar Scene Drum** > **Displayers** > **Tablets** > **Scalar Field** and set *Function* to **Particle** > **Spray Residence Time**.

Now run the simulation:

2. Click (**Run**) in the **Solution** toolbar.

   The simulation rotates the drum for a further seven seconds, passing particles through the spray. This spray 'coats' the particles by assigning a passive scalar value. The residence time of each individual tablet is indicated by the color change and the range of residence time is given in the scale to the right of the scene.

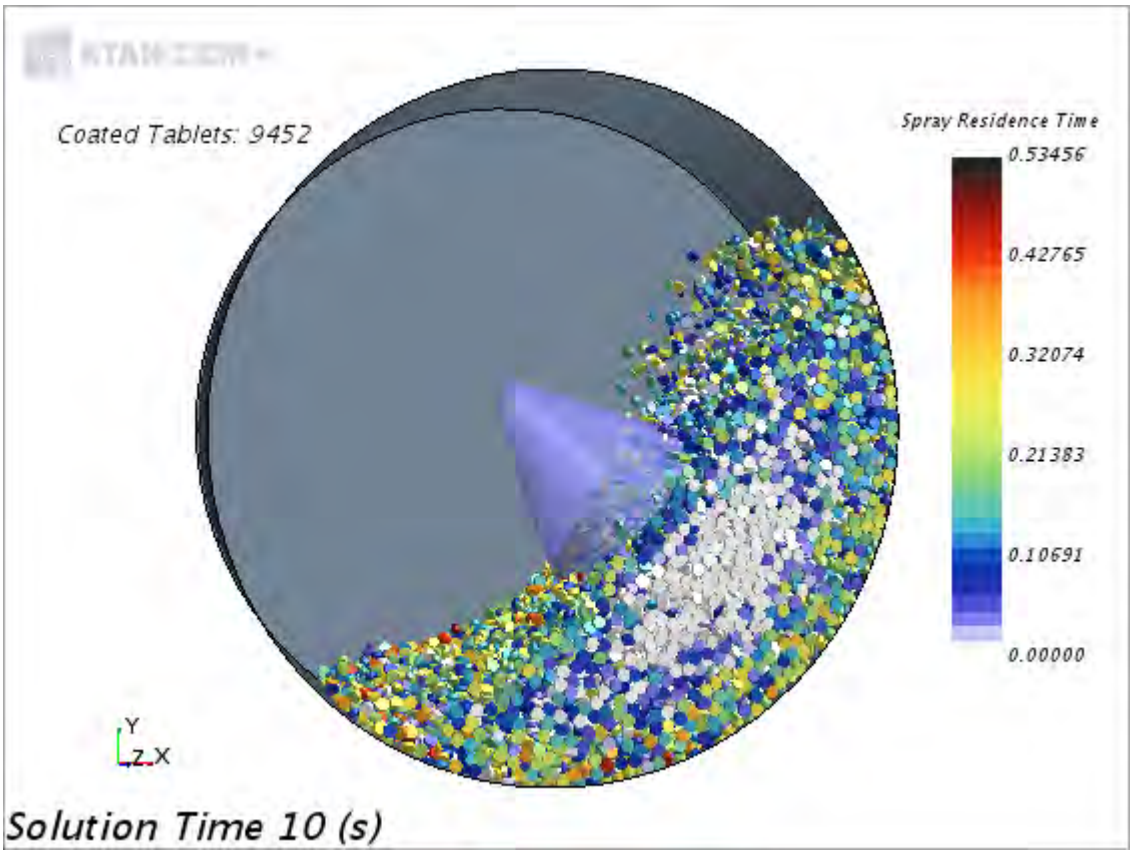
   Running the simulation takes approximately five hours on a workstation with eight processors.

You can monitor the iterative progress in the *Output* window. The scalar scene displays the tablets tumbling in the drum. The residence time increments each time a tablet passes through the conical volume of the spray.

When the solution is complete, the following message is displayed in the *Output* window: `Stopping criterion Maximum Physical Time satisfied.`

The final scalar scene should resemble the screenshot below.

Note: STAR-CCM+ does not produce exactly the same image owing to the random injection algorithm which affects the initial conditions.



3. Save the simulation.

Summary

This tutorial simulates the injection of cylindrical particles (tablets) into a rotating drum and being coated by a conical spray.

This tutorial introduced the following STAR-CCM+ features:

- Definition of Lagrangian phases.
- Definition of DEM particle interaction.
- Creation of a particle injector.
- Definition of passive scalars.
- Setting solver and stopping criteria.
- Running the simulation.
- Visualizing the results.

Motion

The tutorials in this set illustrate various STAR-CCM+ features for simulating problems with moving geometries and meshes, dynamic fluid body interaction, and rigid body motion:

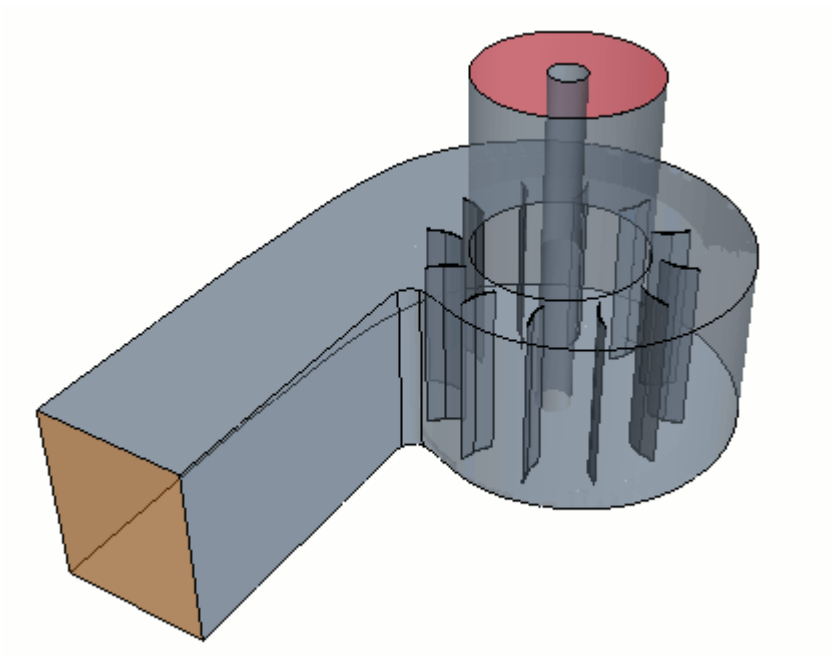
Contents:

Moving Reference Frames: Rotating Fan
Rigid Body Motion: Rotating Fan
DFBI: Boat in Head Waves
DFBI: Lifeboat with Overset Mesh
Marine Resistance Prediction: KCS Hull with a Rudder
Moving Reference Frames: Marine Propeller in Open Water
Body Force Propeller Method: Marine Self-Propulsion
Blade Element Method: Helicopter Rotor-Fuselage Interaction
Morphing: Cylinder with Boundary Motion

Moving Reference Frames: Rotating Fan

This tutorial outlines the steps that are required to set up and run a rotating radial fan analysis. It uses the Moving Reference Frame model, a steady-state approach that involves two or more frames of reference that can be stationary or moving relative to each other.

The problem involves an idealized fan assembly whose axis and 12 equally spaced blades are rotating at 2,000 rpm while the external body remains stationary. Air enters through the inlet, that is shown in red below, at 5 m/s and exits through a pressure boundary, that is shown in orange.



In this case, two reference frames are used for the analysis — a spinning one representing the fan and blades and a stationary one representing the fan body and inlet section.

Contents:

Importing the Mesh
Scaling the Mesh
Setting up Physics Models
Creating a Moving Reference Frame
Setting Initial Conditions
Creating Interfaces
Setting Boundary Conditions
Setting Solver Parameters and Stopping Criteria
Visualizing the Solution
Reporting Monitoring and Plotting
Running the Simulation

Visualizing the Results
Summary

Importing the Mesh

To set up the STAR-CCM+ simulation, launch a simulation and import the supplied volume mesh.

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh...**.
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/motion/data`.
5. Select `fan.ccm` then click **Open**.

   STAR-CCM+ provides feedback on the import process in the *Output* window. Two mesh regions named **Fluid** and **Rotating** are created under the **Regions** node.
6. Save the simulation as `fan_mrf.sim`.

Scaling the Mesh

For the input file supplied, STAR-CCM+ interprets the length units in meters instead of millimeters. You can restore the correct dimensions by scaling the mesh.

The original grid was built using millimeters as the length unit, but no scaling factor was applied when the grid was saved in the .ccm file. You can refer to the output that is produced at import for information on the mesh regions. The mesh regions extend from roughly -113 to 240 units in the x-direction, -133 to 98 in the y-direction and -16 to 184 in the z-direction. Without any further action, STAR-CCM+ interprets these values as meters, which would make the fan rather large. Therefore, scale the region by a factor of 0.001 so that the dimensions are correctly specified:

1. Select **Mesh** > **Scale Mesh...** from the menu bar.
2. In the *Scale Mesh* dialog, make sure that the **Fluid** and **Rotating** regions are selected.
3. Enter a value of `0.001` for the **Scale Factor**.

4. Click **Apply** to scale the domain and then click **Close**.

   When the Apply button is clicked, the mesh region is reduced in size.
5. To restore the previous viewing distance from the scaled domain in the display, click the **Reset View** button in the toolbar.

To verify that scaling has in fact been applied, you can perform the mesh diagnostics check and review the output values.

Setting up Physics Models

As previously noted, the mesh consists of two regions: Fluid and Rotating. However, only one continuum is needed for the simulation since the two regions contain the same material. The physical models for this continuum are defined next.

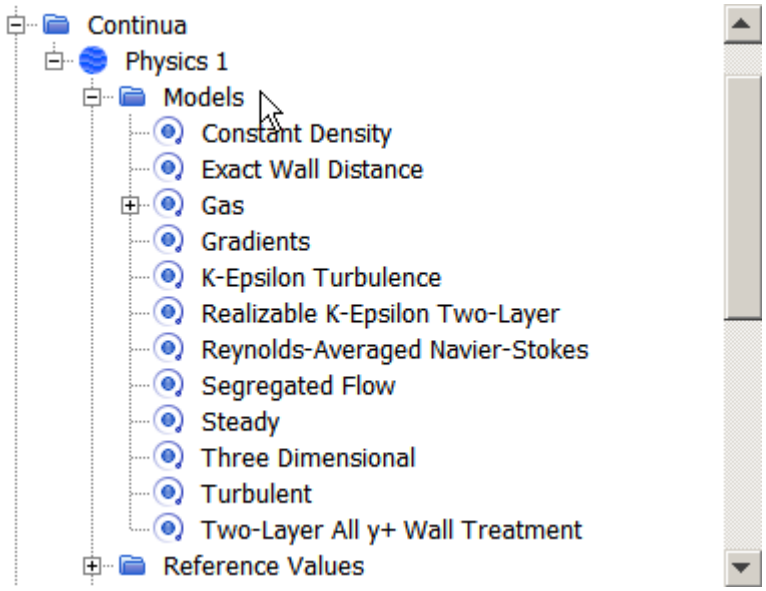
To set the physics models:

1. For the physics continuum, **Continua** > **Physics 1** , select the following models, in order:

Group Box	Models
Time	Steady
Space	Three Dimensional
Material	Gas
Flow	Segregated Flow Gradients (selected automatically)
Equation of State	Constant Density
Viscous Regime	Turbulent Reynolds-Averaged Navier-Stokes (selected automatically)
Reynolds-Averaged Turbulence	K-Epsilon Turbulence Two-Layer All y+ Wall Treatment (selected automatically) Exact Wall Distance (selected automatically) Realizable K-Epsilon Two-Layer (selected automatically)

2. Click **Close**.
3. To review the models, open **Physics 1** > **Models** .

   The selected models appear as shown:



4. Save the simulation.

Creating a Moving Reference Frame

Create a moving reference frame for the rotating region.

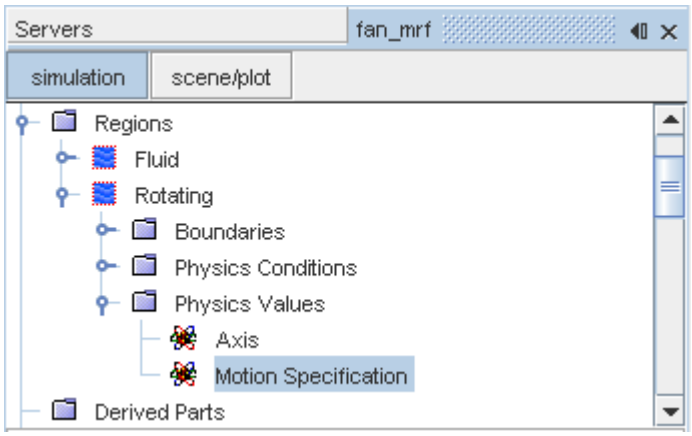
Reference frames are created and managed within the **Tools** > **Reference Frames** node, and are applied to regions after they are created.

1. Right-click the **Tools** > **Reference Frames** node and select **New** > **Rotating**.
2. Select the **Rotating** node and set the **Rotation Rate** to `2000 rpm`.

   The other properties are left unchanged as the default definition for the axis of rotation is suitable for this problem.

Assign the reference frame to the **Rotating** region:

3. Select the **Regions** > **Rotating** > **Physics Values** > **Motion Specification** node.



4. In the *Properties* window, set **Reference Frame** to **Rotating**.

   The **Physics Values** > **Axis** node of the region is removed, as the axis specification is now taken from the Rotating reference frame.

Setting Initial Conditions

Change the turbulence parameters that are set as part of the initial conditions.

To set the initial conditions:

1. Edit the **Continua** > **Physics 1** > **Initial Conditions** node and set the following properties:

Node	Property	Setting
Turbulence Specification	*Method*	**K + Epsilon**
Turbulent Dissipation Rate	*Value*	`150.0 m^2/ s^3`
Turbulent Kinetic Energy	*Value*	`1.5 J/kg`

2. Save the simulation.

Creating Interfaces

All regions in a STAR-CCM+ simulation require an interface to be present between them that transfers the appropriate mass and energy quantities during the calculation.

In this example, create two interfaces between the **Fluid** and **Rotating** regions. This process involves creating so-called “In-place” interfaces between the default wall boundaries enclosing each region.

The first interface is created between the inlet and rotating section:

1. Open the two **Boundaries** nodes inside the **Fluid** and **Rotating** nodes.
2. Multi-select the two nodes named **Interface1** from each boundary grouping.
   These two boundaries represent the interface between the inlet section of the fan and the rotating region and are now highlighted in pink in the Geometry Scene display.
3. Right-click either of the selected nodes and then select **Create Interface**.

   The above step creates a node within each **Boundaries** node called **Interface1 [Interface 1]**.

The second interface is created between the fan body and the rotating region.

4. Multi-select the two nodes named **Interface2** from each boundary grouping.
   These two boundaries represent the interface between the rotating region and the rest of the stationary part of the fan and are now highlighted in pink in the *Geometry Scene* display.
5. Right-click either of the selected nodes and then select **Create Interface** to create another in-place interface called **Interface2 [Interface 2]**.
6. Save the simulation.

In addition to the in-place boundaries, an **Interfaces** node appears in the simulation tree. Within it, two interface nodes, **Interface 1** and **Interface 1** are created. This forms the connection between the **Fluid** and

Rotating regions once the flow initialization step has been performed and supersedes the original wall boundaries forming the connection.

Setting Boundary Conditions

The fluid velocity at the inlet and the rotation conditions at the wall boundaries are specified.

STAR-CCM+'s default rotation setting for all wall boundaries is for them to remain stationary relative to the mesh. Therefore, in absolute terms, the default setting for the wall boundaries of the Fluid region is zero rotation while the default setting for the wall boundaries of the Rotating region is 2000 rpm. This setting is inappropriate for one wall boundary in each region:

- The **Axis** boundary in the **Fluid** region, which is rotating as it forms part of the axle rotating the blades.
- The **Walls** boundary in the **Rotating** region, which is stationary as the walls form part of the casing.

To set the boundary conditions for the **Fluid** region:

1. Edit the **Regions** > **Fluid** > **Boundaries** > **Inlet** node and set the following properties:

Node	Property	Setting
Physics Values		
└ **Velocity Magnitude**	*Method*	**Constant**
	Value	`5.0 m/s`

2. Edit the **Regions** > **Fluid** > **Boundaries** > **Axis** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Tangential Velocity Specification**	*Method*	**Rotation Rate**
Physics Values		
└ **Wall Relative Rotation**	*Method*	**Constant**
	Value	`2000 rpm`

To set the boundary conditions for the **Rotating** region:

3. Edit the **Regions** > **Rotating** > **Boundaries** > **Walls** node and set the following properties.

Node	Property	Setting
Physics Conditions		
└ **Tangential Velocity Specification**	*Method*	**Fixed**
Reference Frame Specification	*Option*	**Lab Frame**

Setting Solver Parameters and Stopping Criteria

To make sure that a reasonable solution is obtained, alter some of the solver parameters.

To set the solver parameters and the stopping criteria:

1. Select the **Solvers** > **Segregated Flow** > **Velocity** node and change the **Under-Relaxation Factor** to `0.5`.
2. Similarly, change the **Pressure Under-Relaxation Factor** to `0.2`.
3. Select the **Pressure** > **AMG Linear Solver** node and change the **Convergence Tolerance** to `0.05`.

The simulation runs for 600 iterations. This number can be specified using a stopping criterion, as follows:

4. Select the **Stopping Criteria** > **Maximum Steps** node and change the **Maximum Steps** to `600`.
5. Save the simulation.

Visualizing the Solution

Visualize the solution while it is being updated by plotting velocity vectors on an x-y section bisecting the problem geometry.

1. Create the section:
   a) Right-click the Derived Parts node and select **New Part** > **Section** > **Plane**.
   b) In the *Create Section* dialog, set the following properties:

Property	Setting
Input Parts	**[Fluid, Rotating]**
Origin	`[0.0, 0.0, 0.0] m,m,m`
Normal	`[0.0, 0.0, 1.0] m,m,m`
Display	**New Geometry Displayer**

   c) Click **Create** and then **Close**.
2. Create the vector scene:
   a) Right-click the **Scenes** node and select **New Scene** > **Vector**.
   b) Select the **Scenes** > **Vector Scene 1** > **Displayers** > **Vector 1** > **Parts** node and add plane section to the Parts list.
   c) Repeat this process so that the only part in the **Parts** list in the **Outline** displayer is the plane section.
   d) Click the camera icon in the Visualization toolbar and select **Projection Mode** > **Parallel**.
   e) Click the camera icon in the Visualization toolbar and select **Views** > **-Z** > **Up -Y**.
3. To show the entire section more clearly, use the middle mouse to zoom in a little.

Reporting Monitoring and Plotting

STAR-CCM+ can dynamically monitor virtually any quantity while the solution develops. The process involves setting up a report which defines the quantity of interest and the parts of the region to monitor. A monitor is

then defined based on the report which controls the update frequency and normalization characteristics. An X-Y graph plot can then be made from the monitor.

To track the area-averaged pressure at the inlet boundary, use a report monitor. This value is a useful measure of whether the solution has converged.

1. In the simulation tree, right-click the **Reports** node and select **New Report** > **Surface Average**.
2. Rename the **Reports** > **Surface Average 1** node to `Area Averaged Inlet Pressure`.
3. Select the **Area Averaged Inlet Pressure** node and add **Fluid: Inlet** to the **Parts** list and change the **Scalar Field Function** to **Pressure**.

4. To create the monitor and plot that are based on this report, right-click the **Area Averaged Inlet Pressure** node and select **Create Monitor and Plot from Report.**

   A new node named **Area Averaged Inlet Pressure Monitor Plot** appears under the **Plots** node.
5. Save the simulation.

Running the Simulation

You can now run the simulation.

To start the analysis:

1. Click (Run) in the toolbar.

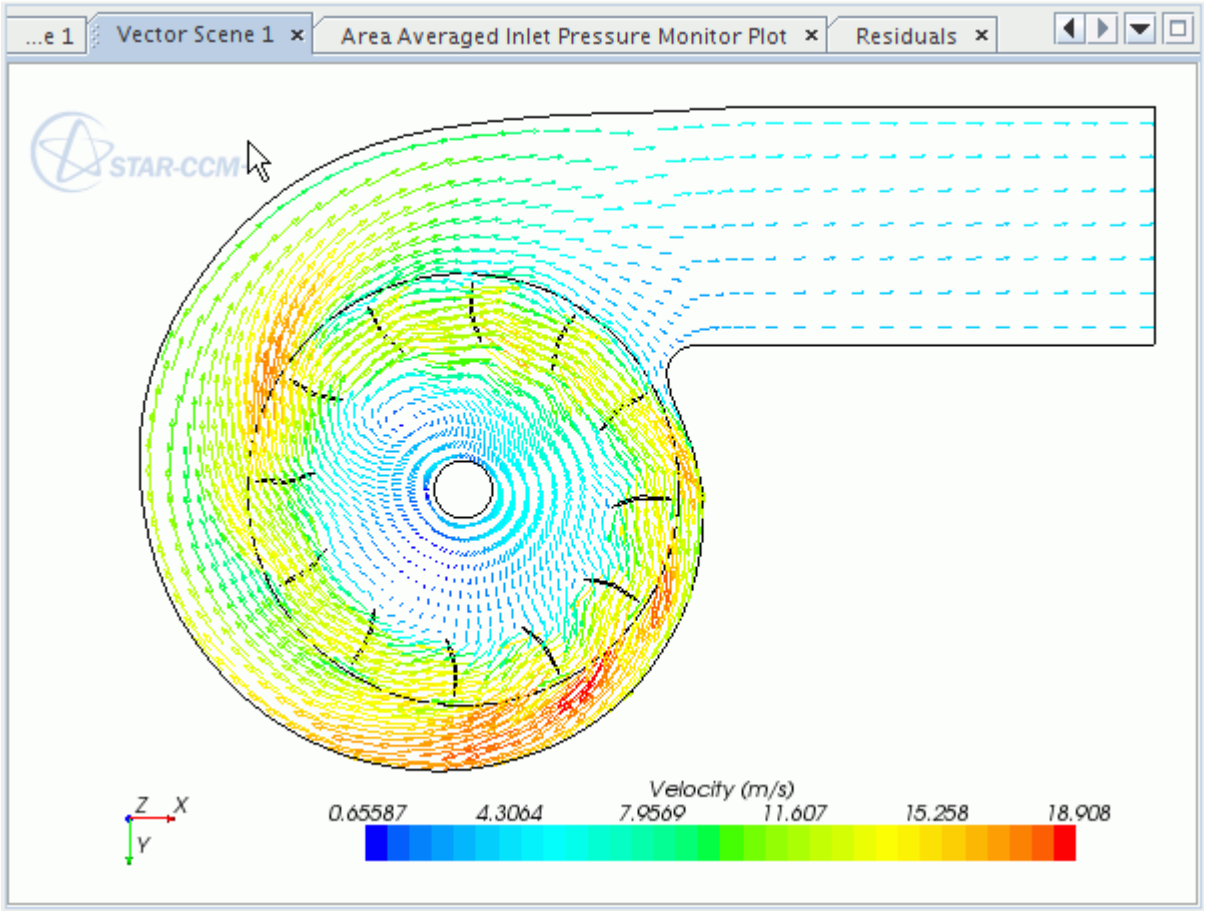
   The *Residuals* display is automatically created and shows the progress of the solver.

For this case, however, the *Area Averaged Inlet Pressure Monitor Plot* is a more useful plot to view during the run. The tabs at the top of the Graphics window make it possible to select any active display for viewing.

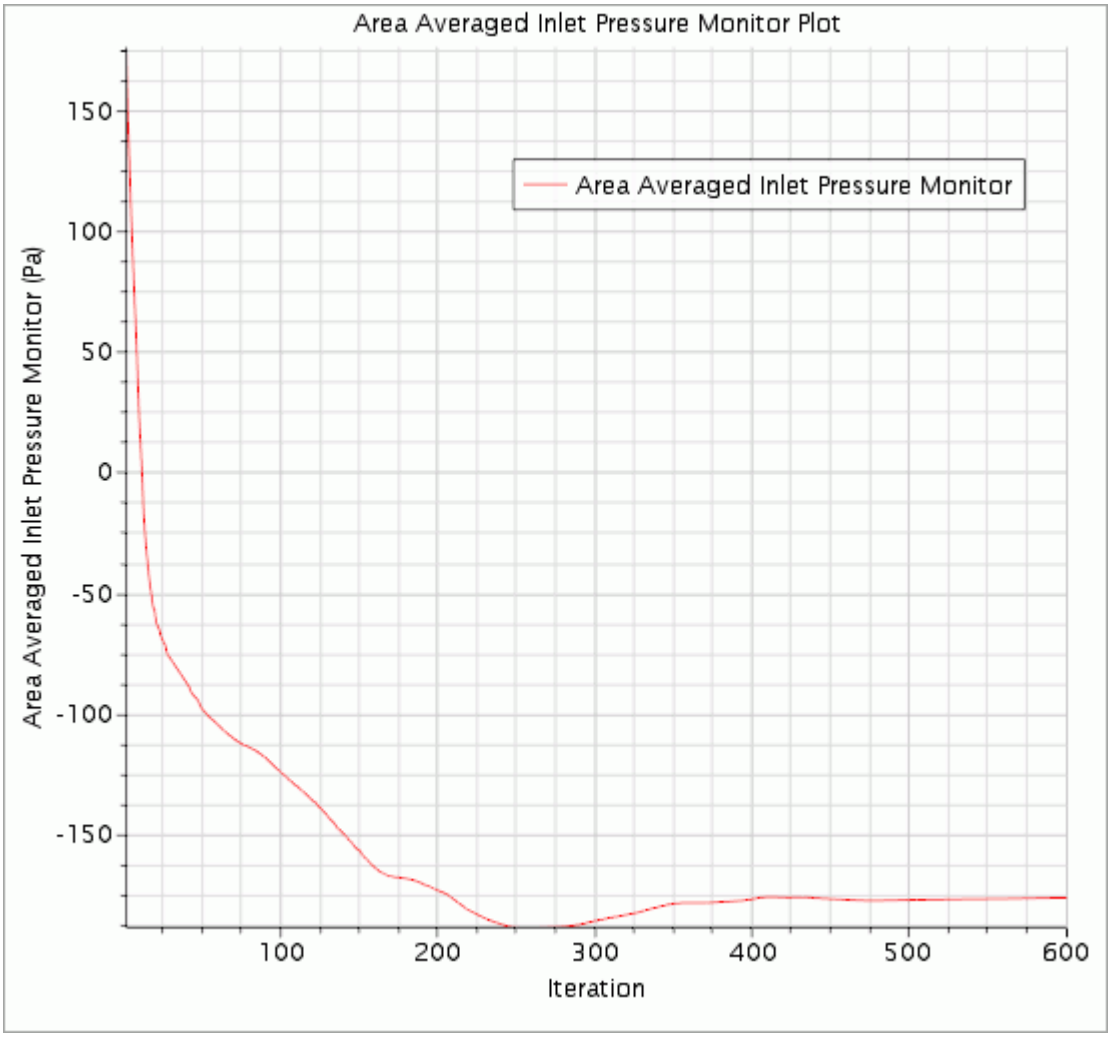
During the run, it is possible to stop the solution by clicking the (Stop) button in the toolbar. If you do halt the simulation, it can be continued by clicking the (Run) button. If left alone, the simulation continues until 600 iterations are completed.

To watch the solution as it is updated:

2. Click the tab for the *Vector Scene 1* display.



At the end of the calculation, the Area Averaged Inlet Pressure Monitor Plot looks as shown below



The pressure at the inlet has almost converged to a steady value. However, it is unlikely to do so completely as the solution to this problem is not truly steady-state.

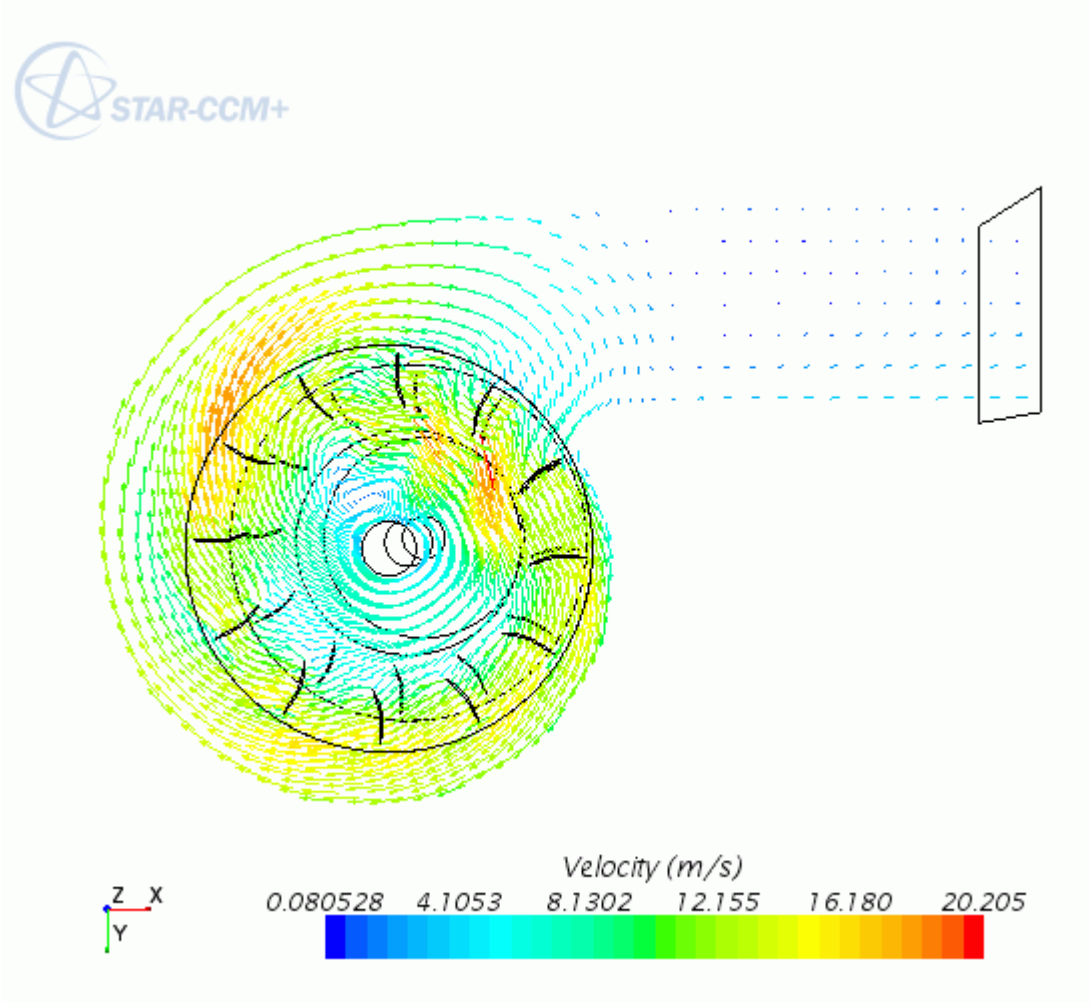
3. Save the simulation.

Visualizing the Results

You can visualize the velocity vector results for the converged solution.

To visualize the results:

1. Click the tab for the *Vector Scene 1* display.



2. Save the simulation.

Summary

This tutorial demonstrated how to set up a moving reference simulation in STAR-CCM+.

This tutorial introduced the following STAR-CCM+ features:

- Defining a moving reference frame model.
- Setting initial conditions.
- Creating interfaces between different mesh regions.
- Specifying region-wise conditions for moving reference frames.
- Defining boundary conditions.
- Setting solver parameters.
- Setting up a monitoring report and plot.
- Running the solver for a set number of iterations.
- Analyzing the results.

Rigid Body Motion: Rotating Fan

This tutorial models the same radial fan problem as the Moving Reference Frames: Rotating Fan. However, instead of using the steady-state moving reference frame model, this tutorial uses the transient Rigid Body Motion model to solve the problem.

To avoid having to set up the entire case, this tutorial restarts from the `.sim` file that was created in the Moving Reference Frames: Rotating Fan tutorial.

Contents:

Loading an Existing Simulation
Setting up the Rigid Body Motion Model
Setting Solver Parameters and Stopping Criteria
Monitoring the Solution
Initializing and Running the Simulation
Checking the Results
Summary

Loading an Existing Simulation

For this tutorial, you start from a previously saved simulation file.

Note: To start this tutorial, you can open one of two versions of the simulation file: the one that you created locally (if applicable), or a previously solved simulation file that is available online. The following instructions apply to a locally created simulation file.

1. Launch STAR-CCM+.
2. Select **File** > **Load...**.
3. In the *Load a File* dialog, navigate to the `fan_mrf.sim` file that was created in the Moving Reference Frames: Rotating Fan tutorial.
4. Select `fan_mrf.sim` then click **Open**.
5. Click **OK**.
6. Save the simulation as `fan_rbm.sim`.

Setting up the Rigid Body Motion Model

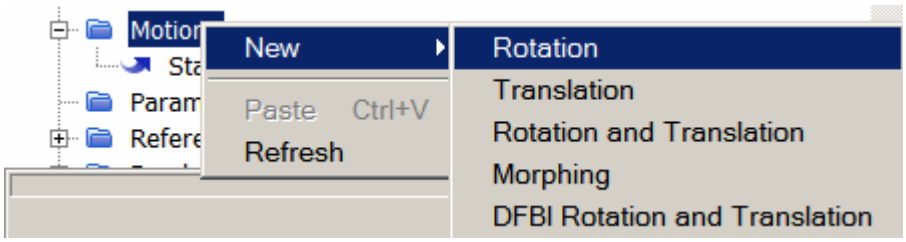
The first step in setting up the Rigid Body Motion model is to switch the analysis type to unsteady.

For the physics continuum, **Physics 1**:

1. Deactivate the **Steady** model from the *Enabled Models* box.
2. Select **Implicit Unsteady** from the *Time* box, then click **Close**.

Create a motion that uses the rigid body motion method and apply this new motion to the **Motion Specification** node of the rotating region.

3. Right-click the **Tools** > **Motions** node and select **New** > **Rotation**.



A new **Rotation** node is added to the **Motions** node.

4. Select the **Rotation** node and set the following properties:

Property	Setting
Axis Direction	`[0.0, 0.0, 1.0]`
Rotation Rate	`2000 rpm`

Apply this motion to the rotating region, and change the reference frame for the region to the stationary default frame.

5. Select the **Regions** > **Rotating** > **Physics Values** > **Motion Specification** node and set **Motion** to **Rotation**.

   Selecting this motion results in the **Reference Frame** property automatically reverting to the default **Lab Reference Frame**.

6. Save the simulation.

Setting Solver Parameters and Stopping Criteria

As the analysis is now unsteady, the time-step size is specified. For the specified rotational speed of 2000 rpm, it takes 0.03 s to complete one full revolution.

In a typical case, it is suggested to use 1 degree per time-step. This setting gives a time-step size of 0.03s/360 or 8.33E-5 s. For accuracy, a 2nd-order temporal discretization scheme is used. For demonstration purposes, you solve three solution cycles, 0.09 s, only.

1. Edit the **Solvers** node and set the following properties:

Node	Property	Setting
Implicity Unsteady	**Time-Step**	`8.33333333e-5 s`
	Temporal Discretization	**2nd-order**
Segregated Flow > **Velocity**	**Under-Relaxation Factor**	`0.9`
Segregated Flow > **Pressure**	**Under-Relaxation Factor**	`0.9`

You can specify the maximum physical time to be 0.09 s. The maximum inner iterations value must be selected carefully: if the value is too low, the solution does not converge within a time-step; if it is too high the solution takes a long time to complete. Select a value of 7 and add a runtime monitor later.

2. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Physical Time	**Enabled**	Activated
	Maximum Physical Time	`0.09 s`
Maximum Inner Iterations	**Maximum Inner Iterations**	`7`
Maximum Steps	**Enabled**	Deactivated

3. Save the simulation.

Monitoring the Solution

Create a report, monitor, and plot to track the moment of force on the surface of the rotating blades during the simulation. Monitoring this value helps to make sure that the solution is converging within each time-step.

1. Right-click the **Reports** node and select **New Report** > **Moment**.

   A node that is called **Moment 1** is added to the **Reports** node.
2. Select the **Moment 1** node, and set **Parts** to **Rotating** > **Blades**.

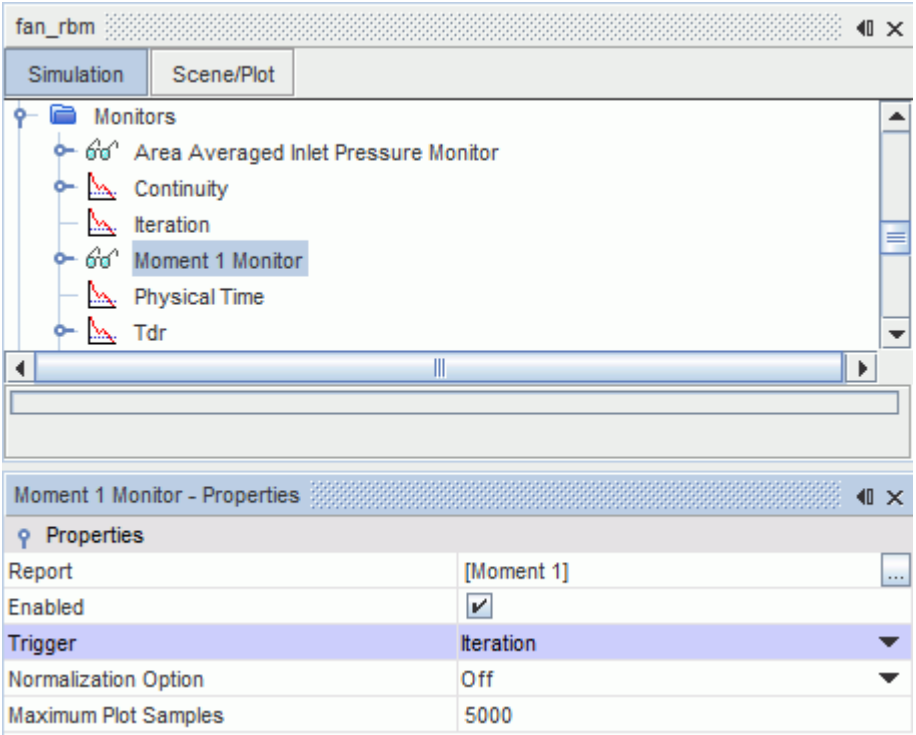
To create the monitor and plot that are based on this report:

3. Right-click the **Reports** > **Moment 1** node and select **Create Monitor and Plot from Report**.

   A new node that is named **Moment 1 Monitor** appears under the **Monitors** node.

   A new node that is named **Moment 1 Monitor Plot** also appears under the **Plots** node.
4. Select the **Monitors** > **Moment 1 Monitor** node and set *Trigger* to **Iteration**.

   Monitoring the moment during time-steps provides the data for intra time-step convergence checks.



Normally, it is important to monitor convergence between time-steps. However, for most plots in unsteady cases it is more useful to examine the solution progress by reporting a quantity every time-step rather than every iteration. Modify the **Area Averaged Inlet Pressure Monitor** plot and the vector scene to update every time-step.

For the vector scene:

5. Select the **Scenes** > **Vector Scene 1** > **Attributes** > **Update** node and set *Trigger* to **Time Step**.

Next, set up the maximum wall pressure as the monitoring quantity:

6. Select the **Monitors** > **Area Averaged Inlet Pressure Monitor** node and set *Trigger* to **Time Step**.

To preview both of the plots:

7. Right-click the **Area Averaged Inlet Pressure Monitor Plot** node and select **Open**.

Make sure that the *X-Axis Monitor* is set to the correct type for each plot:

8. Select **Plots** > **Area Average Inlet Pressure Monitor Plot** and set *X-Axis Monitor* to **Physical Time**.
9. Select **Plots** > **Moment 1 Monitor Plot** and set *X-Axis Monitor* to **Iteration**.

   The selected plots appear in the *Graphics* window.
10. Save the simulation.

Initializing and Running the Simulation

The analysis starts from the same stationary initial flow field that was set up in the Moving Reference Frames: Rotating Fan tutorial, therefore the existing solution must be cleared first.

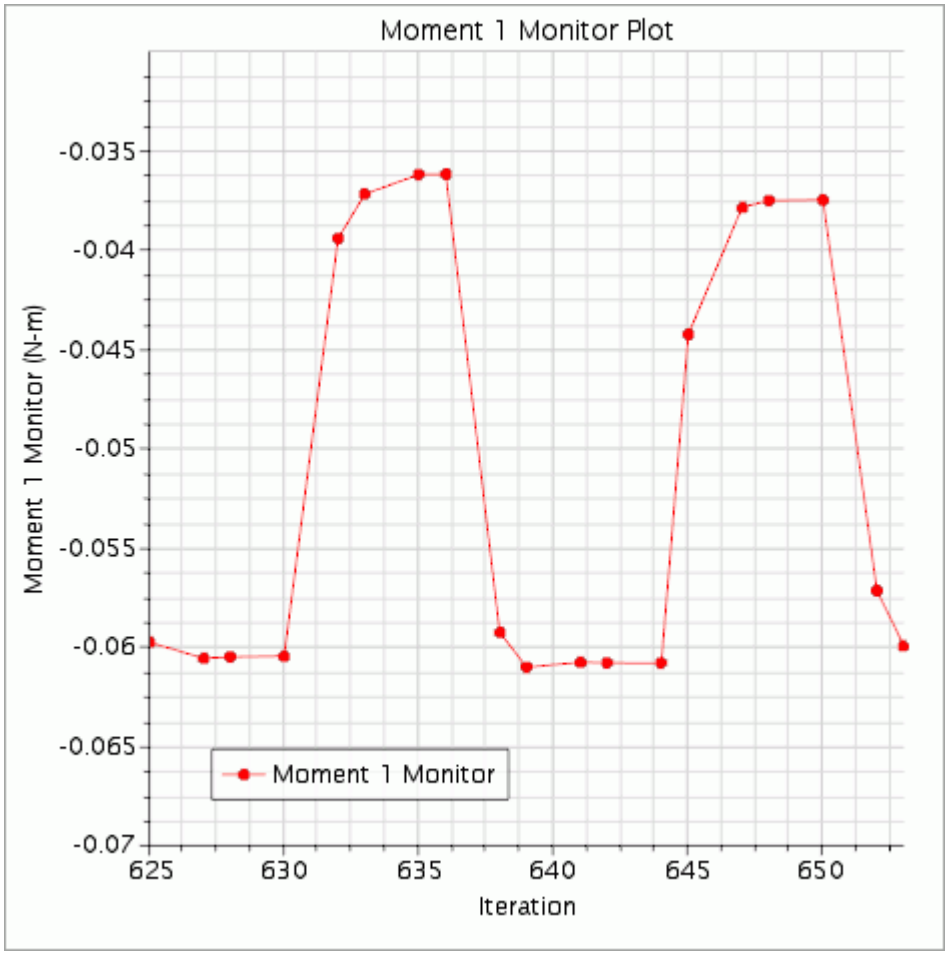
1. In the menu, select **Solution** > **Clear Solution...** and then click **OK** in the *Clear Solution* dialog.
2. Click (**Run**) in the toolbar.

   The *Residuals* display is created automatically and shows the progress of the solver.

During the run, make sure that the number of inner iterations that was selected is sufficient to achieve convergence.

3. Click the *Moment 1 Monitor Plot* tab at the top of the *Graphics* window.
4. Select the **Plots** > **Moment 1 Monitor Plot** > **Data Series** > **Moment 1 Monitor** > **Symbol Style** node and set **Shape** to **Filled Circle**.
5. In the Plot toolbar, click (**Zoom Selected Plot**) and then zoom into the graph over approximately the last 20 iterations.

   The plot appears similar to the example below. (Note: the example plot that is shown below is for demonstration purposes only, results can differ based on the iterations selected).



From the above profile, it is clear the solution is indeed converging within each time-step: the profile levels off in each time-step.

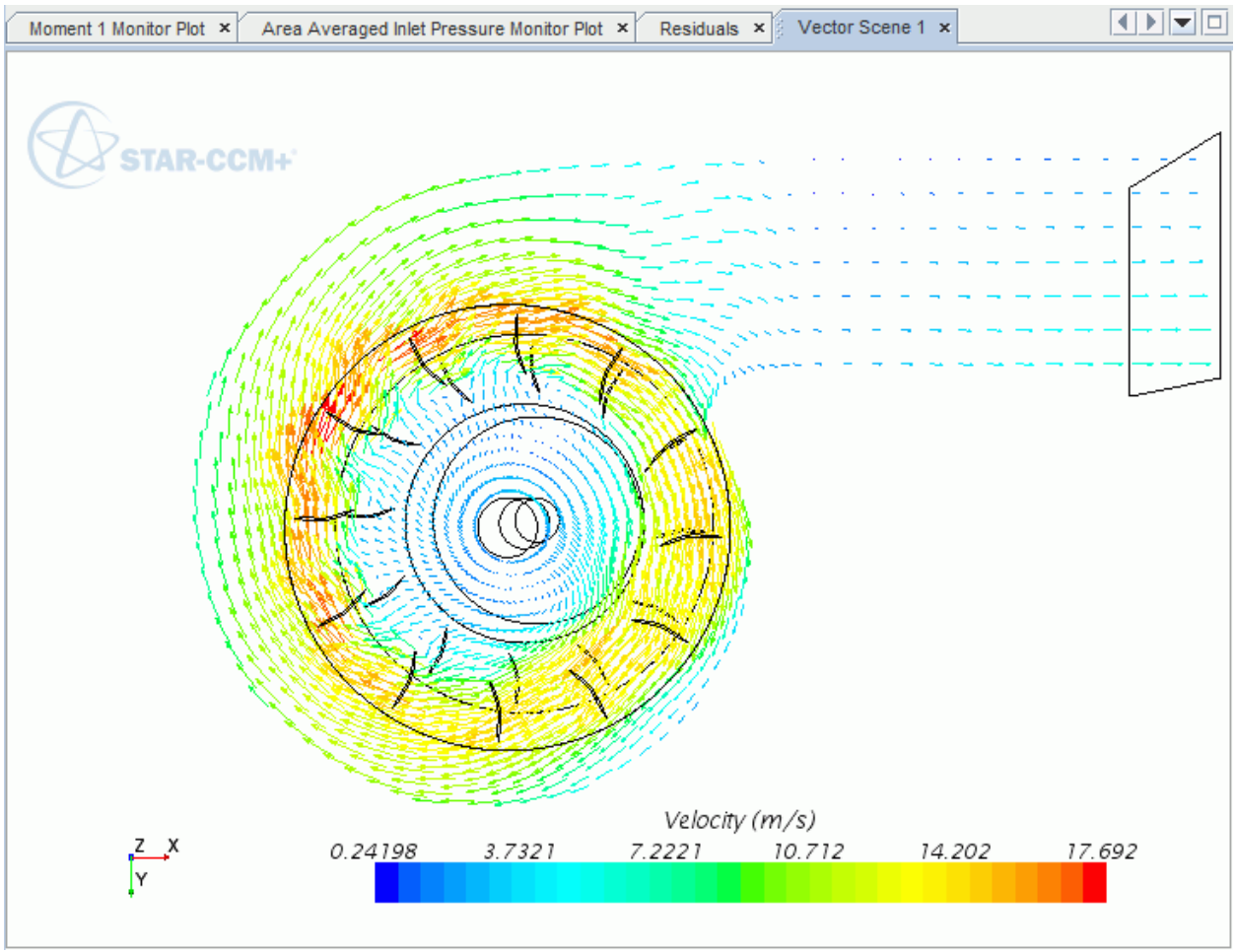
The simulation continues until all time-steps are complete.

6. After the run is finished, save the simulation.

Checking the Results

The vector plot is compared with the results for the Moving frame of Reference case.

1. Right-click the **Scene** > **Vector Scene 1** node and select **Open**.



These velocity vectors are similar to the vectors that were created during the Moving Reference Frames: Rotating Fan tutorial.

2. Click the *Area Averaged Inlet Pressure Monitor Plot* tab at the top of the *Graphics* window.

The large scale that is used for the y-axis makes it difficult to assess the inlet pressure variation over the solution cycles.

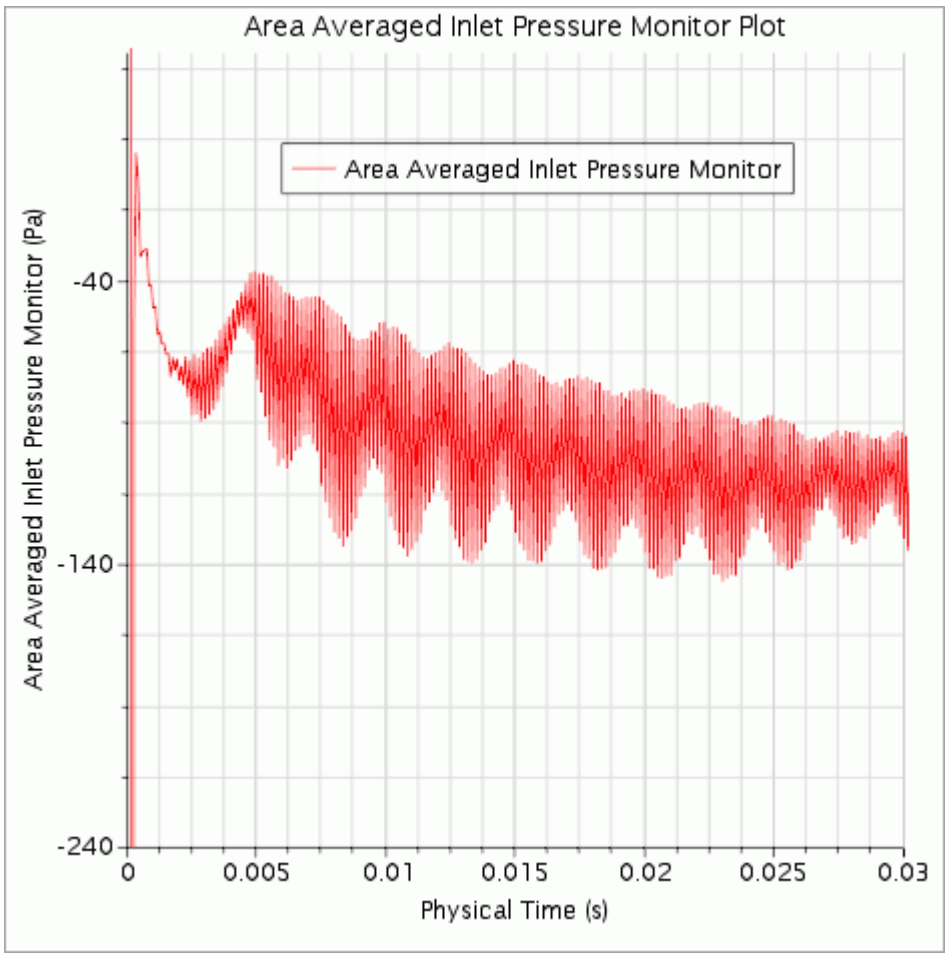
To update the plot:

3. Edit the **Plots** > **Area Averaged Inlet Pressure Monitor Plot** > **Axes** node and set the following properties:

Node	Property	Setting
Left Axis	*Minimum*	`-240`
	Lock Minimum	**Activated**

Node	Property	Setting
	Maximum	40
	Lock Maximum	**Activated**
└ **Major Labels**	*Spacing*	100.0
	Precision	-1
Bottom Axis	*Maximum*	0.03
	Lock Minimum	**Activated**
└ **Major Labels**	*Spacing*	0.005

The plot now appears as shown:



Note: The pressure at the inlet is oscillating with a period of 0.0025 s, that is 1/12 of a revolution, as expected. Over time the amplitude of these oscillations become regular as the flow cycle settles down.

4. Save the simulation.

Summary

This tutorial demonstrated how to set up a rigid body motion simulation in STAR-CCM+.

This tutorial introduced the following STAR-CCM+ features:

- Defining a rigid body motion model.
- Setting solver parameters and stopping criteria for an unsteady analysis.
- Monitoring the solution progress using plots and vector scenes.
- Running the solver for a set time.
- Analyzing the results using STAR-CCM+’s visualization and XY-plot facilities.

DFBI: Boat in Head Waves

This tutorial demonstrates use of the Six Degrees Of Freedom solver with a free surface flow to model the motion of a boat hull facing oncoming waves.

The appropriate features in STAR-CCM+ provide initial conditions and wave generation.

Contents:

Importing the Mesh and Naming the Simulation
Setting Up the Models
Setting the Material Properties
Selecting the 6-DOF Motion Model
Setting the Body Initial Coordinate System
Creating the 6-DOF Body
Defining the VOF Waves
Setting Initial Conditions
Setting the Boundary Conditions
Setting Up Solver Parameters and Stopping Criteria
Creating the Free Surface Isosurface
Creating a Custom Colormap
Visualizing the Free Surface
Creating DFBI Displayers
Exporting Images for Scene Animation
Reporting, Monitoring, and Plotting
Initializing and Running the Simulation
Visualizing the Results
Summary

Importing the Mesh and Naming the Simulation

You launch STAR-CCM+, import the supplied mesh, and save the simulation.

To import the mesh:

1. Start up STAR-CCM+ in a manner that is appropriate to your working environment and select the New Simulation option from the menu bar.
2. Import the three-dimensional hexahedral cell mesh, `boat.ccm` from the `doc/startutorialsdata/motion/data` subdirectory of your STAR-CCM+ installation.
3. Save the new simulation as `boat.sim`.

Setting Up the Models

Several physical models are activated to simulate the forces acting on the hull. The simulation models the behavior of two fluids (air and water) within the same continuum, and uses the Volume of Fluid model to do so.

As there are two fluids in different phases, the Eulerian Multiphase model is activated, and the effect of gravity acting on both is included using the Gravity model.

The effect of turbulence on the fluid is modeled using the default K-Epsilon turbulence model.

A default Physics Continuum was created when the mesh was imported. All necessary models are defined using this continuum.

1. For the physics continuum, **Continua** > **Physics 1**, select the following models in order:

Group Box	Model
Enabled Models	**Three-Dimensional** (Selected automatically)
Time	**Implicit Unsteady**
Material	**Eulerian Multiphase**
Multiphase Model	**Volume of Fluid (VOF)**
Viscous Regime	**Turbulent**
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**
Optional Models	**Gravity** **VOF Waves**

2. Click **Close**.
3. Save the simulation .

Setting the Material Properties

You set the material properties for the Eulerian phases.

In the boat window, open the **Continua** node. The color of the **Physics 1** node has turned from gray to blue to indicate that models have been activated for that continuum.

1. Open the **Physics 1** > **Models** node.

   The selected models now appear within that node.
2. Open the **Eulerian Multiphase** node.

Define the materials corresponding to each of the mixture phases under the **Eulerian Phases** node.

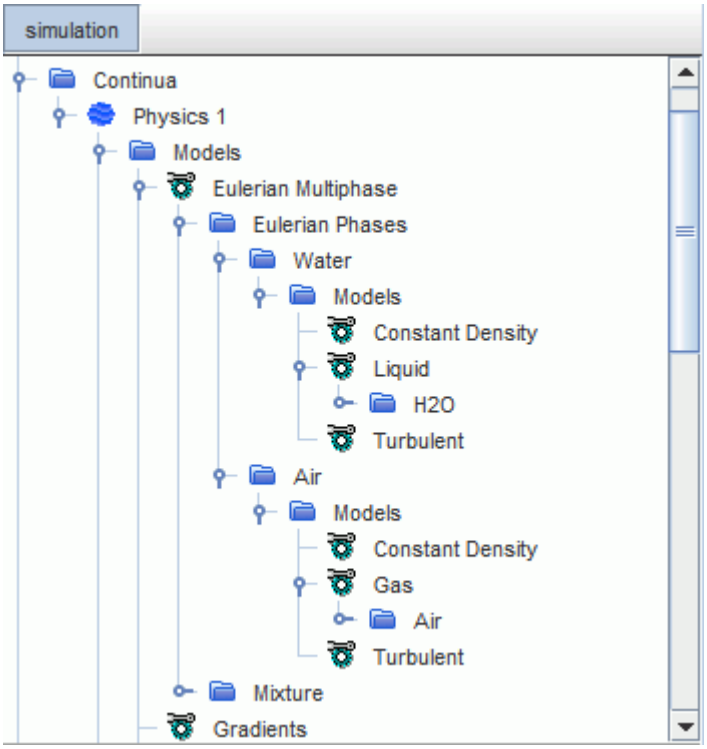
3. Right-click the **Eulerian Phases** node and select **New** to create a phase node that is called **Phase 1**
4. Rename the **Phase 1** node to `Water`

It is now necessary to set the models that determine the behavior of the *Water* phase.

5. Right-click the **Eulerian Phases** > **Water** > **Models** node and choose **Select models...**

6. Select **Liquid** from the *Material* group box.
7. Select **Constant Density** from the *Equation of State* group box.
8. Repeat the steps for a second phase except this time select **Gas** rather than **Liquid**, and specify **Constant Density**.
9. Rename the phase to `Air`

In both instances, you are using the default material appropriate to each phase (water for the liquid and air for the gas), so no further changes are necessary to material properties. The material selections can be seen by expanding the **Water** and **Air** nodes within the **Eulerian Phases** parent node.



10. Save the simulation.

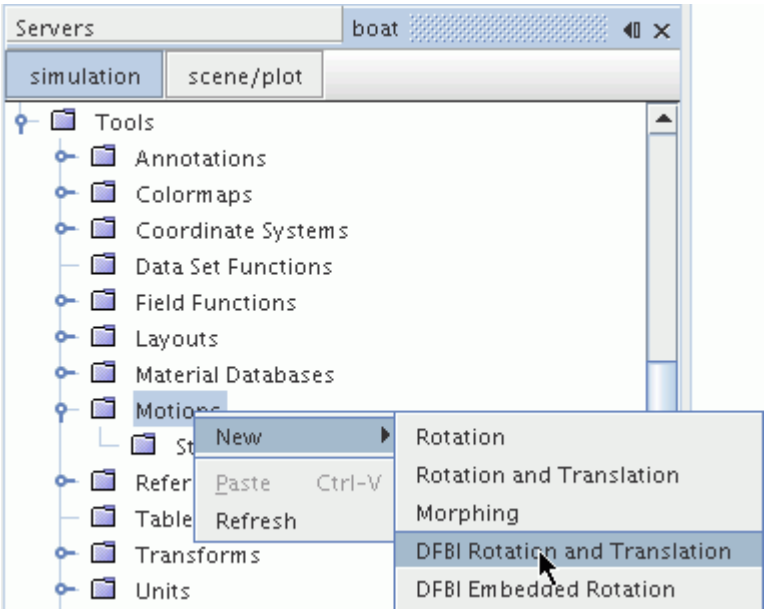
Selecting the 6-DOF Motion Model

The 6-DOF motion solver is activated by creating the relevant DFBI motion object and assigning it to the fluid region. In this case, the DFBI rotation and translation motion is required.

To select the 6-DOF motion model:

1. Open the **Tools** > **Motions** node.

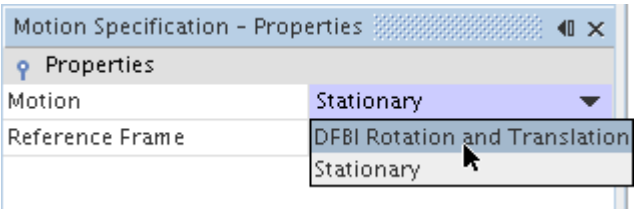
   The default **Stationary** motion is displayed.
2. Right-click the **Motions** node and choose **New** > **DFBI Rotation and Translation** as shown below.



3. A new **DFBI Rotation and Translation** node is added to the **Motion** manager.

Select this motion within the **Motion Specification** node for the fluid region.

4. Select the **Regions** > **Region 1** > **Physics Values** > **Motion Specification** node and set *Motion* to **DFBI Rotation and Translation**.



The action of setting the DFBI motion on the region exposes the **DFBI** manager node on the object tree, and adds the **6-DOF Solver** and **6-DOF Motion** nodes to the **Solvers** manager.

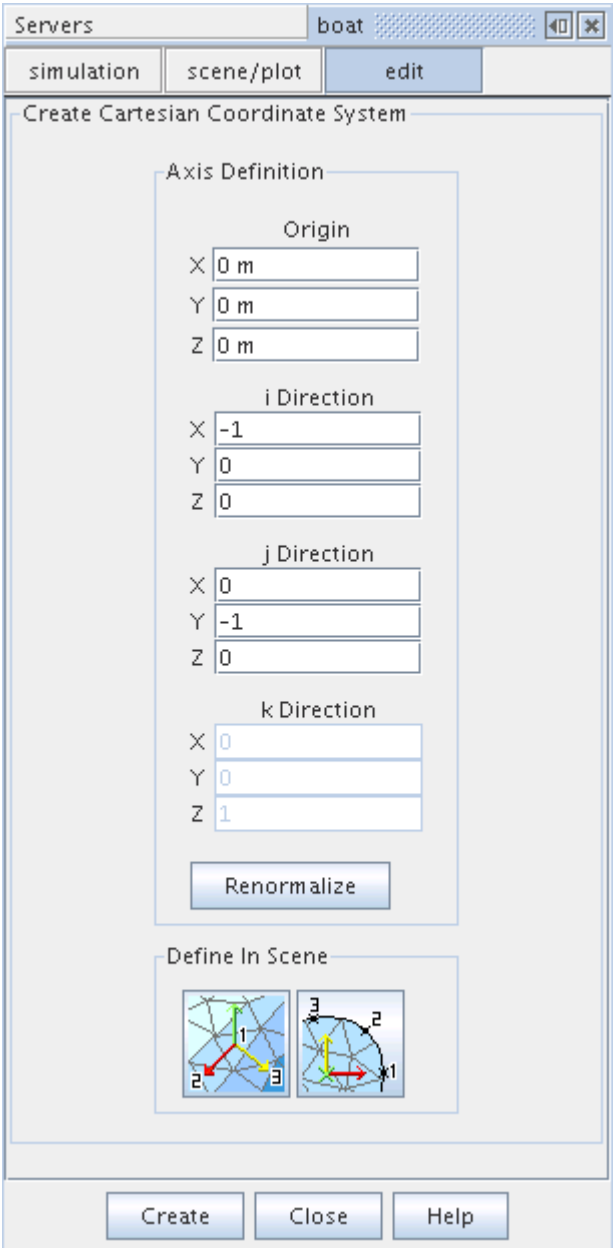
Setting the Body Initial Coordinate System

The 6-DOF Solver must know the initial orientation of the body relative to the laboratory coordinate system. Therefore, define a local coordinate system in which the x-axis is aligned with the forward direction of the body and the z-axis in the vertical direction.

To set the body initial coordinate system:

1. Make sure that *Geometry Scene 1* is still open.
2. Open the **Tools** > **Coordinate Systems** node and right-click the **Laboratory** > **Local Coordinate Systems** node. Choose **New** > **Cartesian** from the pop-up menu

   The *Create Cartesian Coordinate System* window is displayed in the Explorer Pane.
3. Set the *X* value of the *i Direction* to `-1` and the *Y* value of the *j Direction* to `-1` as shown below.

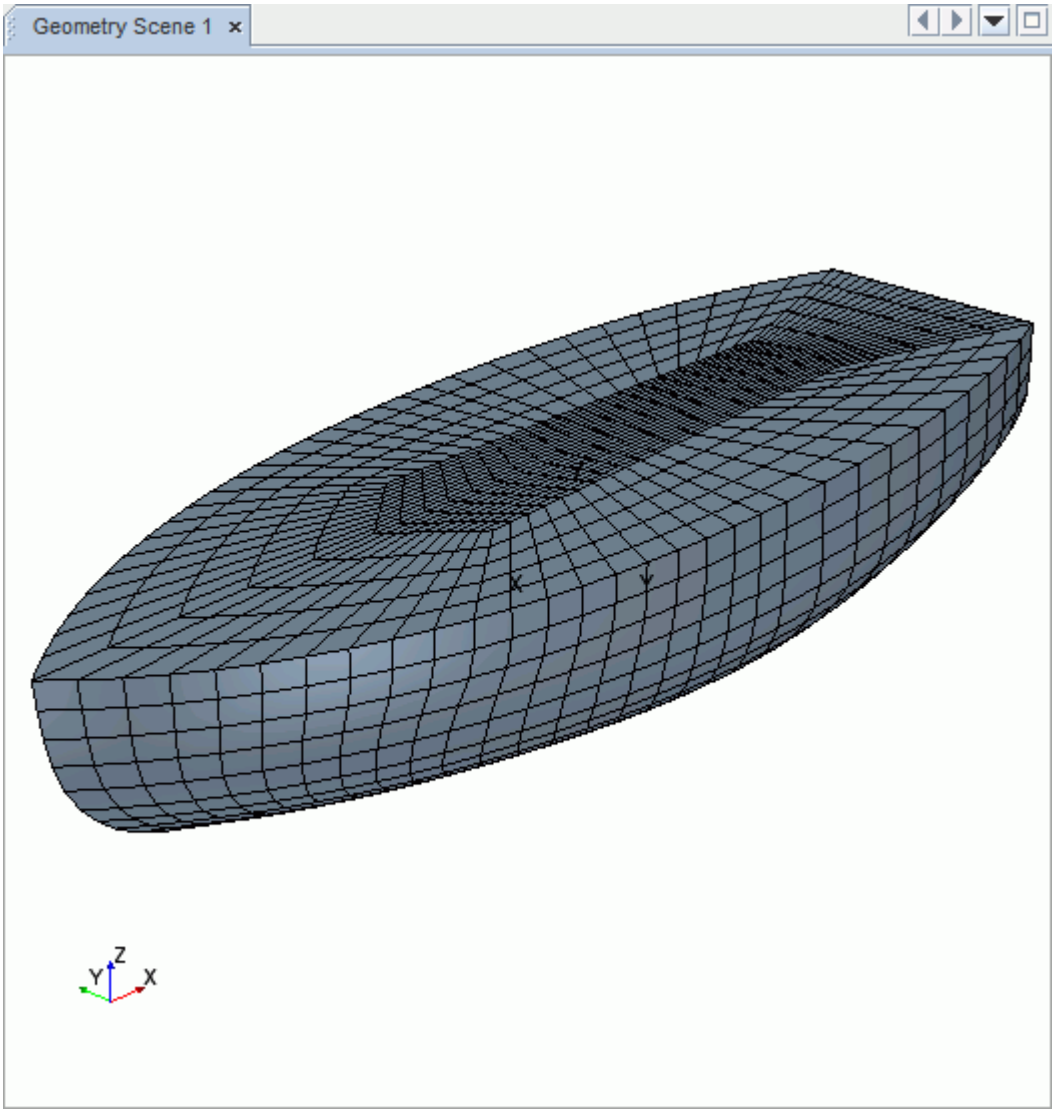


4. Click **Create** and then **Close**.

The node is shown as **Cartesian 1** in the object tree

5. Rename the **Cartesian 1** node to `Initial ship orientation`

Note: The location and orientation of the coordinate system can be checked visually in *Geometry Scene 1*. Select the **Initial ship orientation** node in the object tree and rotate the model in the scene as required.

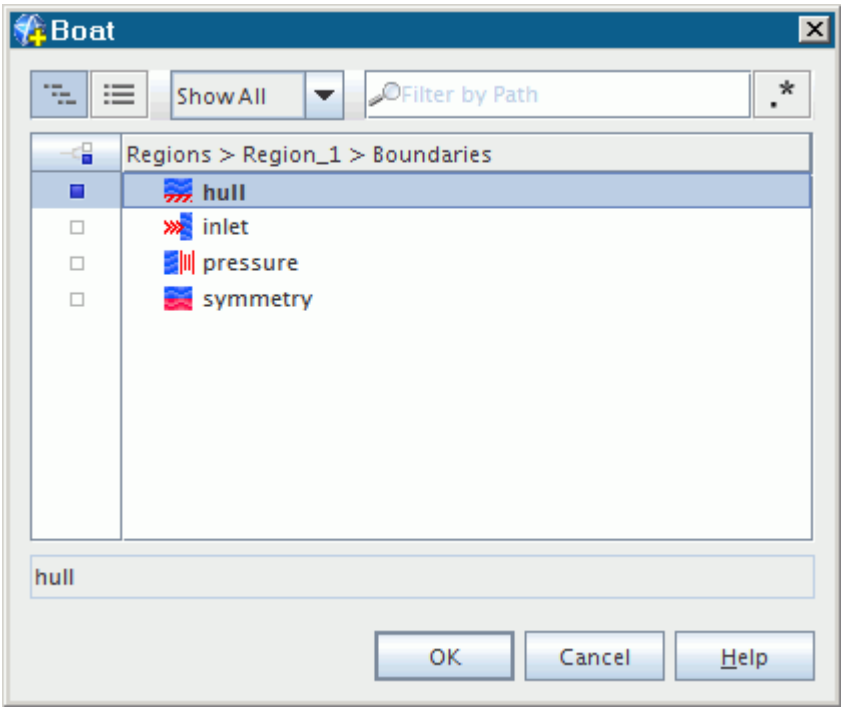


Creating the 6-DOF Body

The next step is to define the surface of the floating body and its properties.

To define the surface and its properties, first create a 6-DOF body to which the boundary representing the boat hull is assigned:

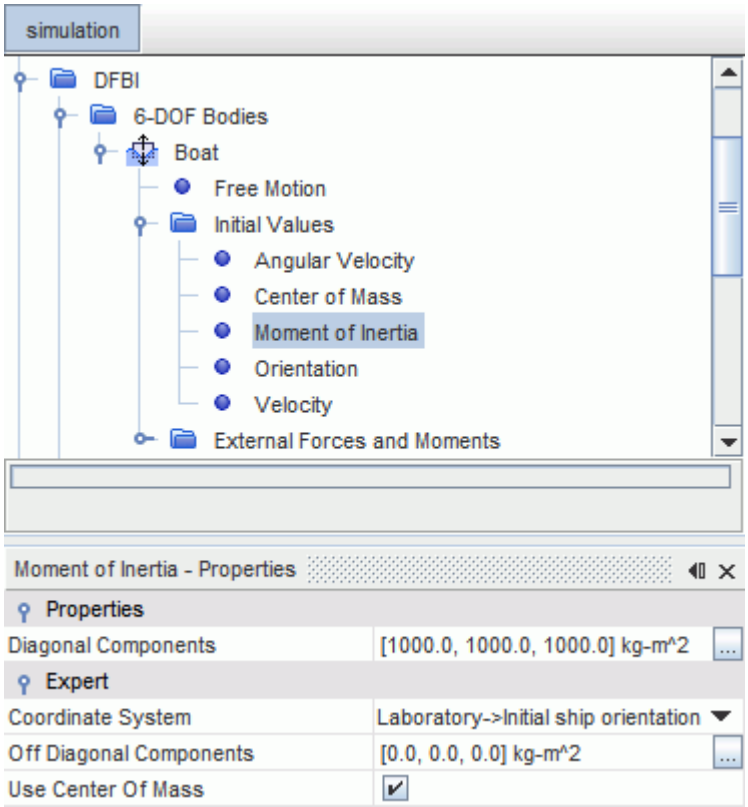
1. Right-click the **DFBI** > **6-DOF Bodies** node and select **New** > **3D Body** from the pop-up menu.
2. Rename the **Body 1** node to `Boat`
3. Select the **Boat** node and click the custom editor button ... to the right of the *Body Surface* property. The *Select Objects* dialog appears.
4. Expand the **Regions** and **Region_1** nodes and tick the checkbox for the **hull** node. Click **OK** to close the dialog.



5. Set the *Body Mass* to `400.0 kg`.
6. So that the fluid solution can develop before starting the body motion, set the *Release Time* property to `0.5 s`.

In this simulation, the body is allowed to rotate about the y-axis only, to simulate motion in the head waves. This means that the moment of inertia about the y-axis must be defined.

7. Select the **Boat** > **Initial Values** > **Moment of Inertia node:**
   a) Set *Diagonal Components* to `[1000.0, 1000.0, 1000.0] kg-m^2`.
   b) Set *Coordinate System* to **Laboratory->Initial ship orientation**.
   c) Activate *Use Center of Mass*.

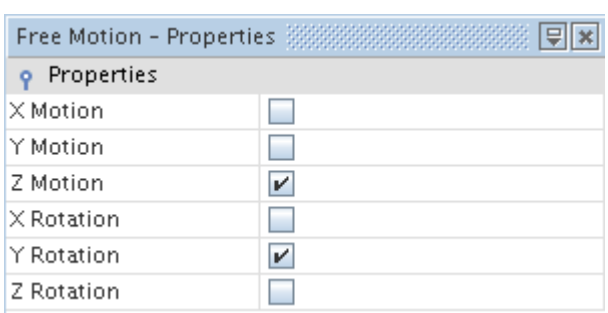


The previously created coordinate system defines the initial orientation of the body.

8. Select the **Boat** > **Initial Values** > **Orientation** node and set *Coordinate System* to **Laboratory->Initial ship orientation**.

Although the 6-DOF body has now been created with the Free Motion option, it is still necessary to specify the manner in which the body can move. In this case, the body must be free to translate in the Z direction (heave) and to rotate about the Y axis (pitch).

9. Select the **DFBI** > **6-DOF Bodies** > **Boat** > **Free Motion** node and tick the *Z Motion* and *Y Rotation* properties.



10. Save the simulation.

Defining the VOF Waves

It is possible to define transient boundary conditions that simulate waves on the free surface using the VOF Wave physics model in STAR-CCM+.

The VOF Wave model generates a number of field functions that are based on wave parameters that you define. These field functions can be used as boundary and initial conditions in appropriate nodes on the object tree.

1. To create the VOF Wave, right-click the **Continua** > **Physics 1** > **Models** > **VOF Waves** > **Waves** node and select **New** > **Fifth Order**.
2. Rename the **Fifth Order Vof Wave 1** node to `Head Wave`.
3. Edit the **Head Wave** node and set the following properties:

Property	Setting
Current	`[2.5, 0.0, 0.0] m/s`
Wind	`[2.5, 0.0, 0.0] m/s`
Wave Height	`0.2 m`
Water Depth	`10 m`
Specification Type	**Wave Length**

4. Select the **Head Wave** > **Wave Length Specification** node and set *Wave Length* to `6.0 m`.

Setting Initial Conditions

Set the initial conditions using field functions that are associated with the previously defined VOF Wave.

Initial conditions that are set are the free-surface wave profile, the velocity profile within the phases and the hydrostatic pressure.

To set the initial conditions:

1. Edit the **Continua** > **Physics 1** > **Initial Conditions** node and set the following properties:

Node	Property	Setting
Volume Fraction	*Method*	**Composite**
└ **Composite** > **Water**	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Heavy Fluid of Head Wave**
└ **Composite** > **Air**	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Light Fluid of Head Wave**
Velocity	*Method*	**Field Function**
	Vector Function	**Velocity of Head Wave**

Node	Property	Setting
Pressure	*Method*	**Field Function**
	Scalar Function	**Hydrostatic Pressure of Head Wave**

2. Save the simulation.

Setting the Boundary Conditions

There are four boundaries for this geometry:

1. No-slip wall boundary on the hull of the boat
2. Symmetry boundary on the Z-X plane at Y = 0.0
3. Pressure outlet on the downstream face
4. Velocity inlet on all remaining boundaries

The boundary types were correctly set during the initial mesh import, so it only remains to set the conditions corresponding to each type. Default conditions are suitable for the symmetry plane and the no-slip wall boundary on the boat hull. Wave conditions are applied to the pressure outlet and velocity inlet.

Contents:

Inlet Boundary
Pressure Boundary

Inlet Boundary

The head wave enters the simulation through the inlet boundary. You use field functions to define this oncoming wave.

To set the inlet boundary:

1. Edit the **Regions** > **Region_1** > **Boundaries** > **inlet** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Velocity Specification**	*Method*	**Components**
Physics Values		
└ **Velocity**	*Method*	**Field Function**
	Vector Function	**Velocity of Head Wave**
└ **Volume Fraction**	*Method*	**Composite**
└ **Composite** > **Water**	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Heavy Fluid of Head Wave**

Node	Property	Setting
└ **Composite** > **Air**	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Light Fluid of Head Wave**

Pressure Boundary

The head wave leaves the domain through the pressure boundary. You use field functions to define the wave at the boundary.

To set the pressure boundary:

1. Edit the **Regions** > **Region_1** > **Boundaries** > **pressure** node and set the following properties:

Node	Property	Setting
Physics Values		
└ **Pressure**	*Method*	**Field Function**
	Scalar Function	**Hydrostatic Pressure of Head Wave**
└ **Volume Fraction**	*Method*	**Composite**
└ **Composite** > **Water**	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Heavy Fluid of Head Wave**
└ **Composite** > **Air**	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Light Fluid of Head Wave**

Setting Up Solver Parameters and Stopping Criteria

As this simulation is transient, set the time step, the maximum number of inner iterations that are permitted in each time step, and the overall physical time for which a solution is obtained.

1. Select the **Solvers** > **Implicit Unsteady** node and set *Time-Step* to `0.01 s`
2. Select the **Stopping Criteria** > **Maximum Inner Iterations** node and make sure that the *Maximum Inner Iterations* is set to `5`
3. Select the **Stopping Criteria** > **Maximum Physical Time** node and set *Max Physical Time* to `2.0 s`

Creating the Free Surface Isosurface

It is instructive to visualize the motion of the boat hull in response to the oncoming wave over time.

This motion can be visualized by creating a scalar scene showing the boat hull with the free surface, and writing out this scene as a graphics file at each time step of the simulation. The resulting collection of files can be used to form an animation of the free surface and hull motion.

To plot the surface of the water in a scalar scene, first create an isosurface that is based on the volume fraction of water in a cell:

1. Right-click the **Derived Parts** node and select **New Part** > **Isosurface...**.
2. In the *Create Isosurface* dialog, set *Scalar* to **Volume Fraction** > **Volume Fraction of Water**.
3. Leave the *Extraction Mode* set to `Single Value`.
4. Set *Isovalue* to `0.5`.
5. Select **Existing Displayer** > **Geometry 1** in the *Display* panel.
6. Click **Create**, and then **Close**
7. Rename the **Isosurface**node to `Water Surface`

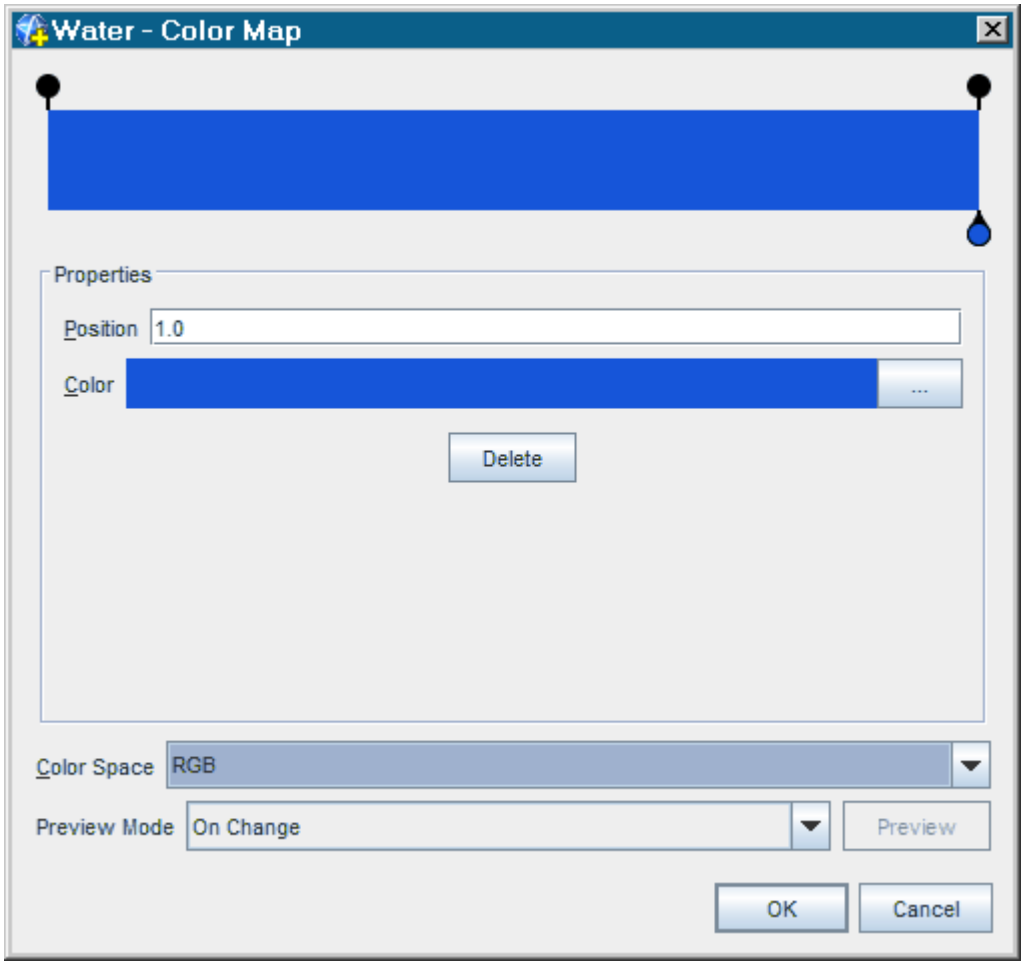
Creating a Custom Colormap

To make the water surface appear in blue for visualization, you can create a custom colormap.

Set the color of the free surface artificially to a uniform blue. First create a custom **Colormap** within the **Tools** node as follows:

1. Right-click the **Tools** > **Colormaps** node and select **New**.
2. In the *Create Color Map* dialog, click **OK**.
3. Rename the **user 1** node to `Water`.
4. Select the **Tools** > **Colormaps** > **Water** node and open the *Color Map* property for editing by clicking the editor icon.
5. Click and drag the blue control point all the way to the right so that the color bar is entirely blue.
6. Set *Color Space* to **RGB**.

   The completed dialog is shown below.

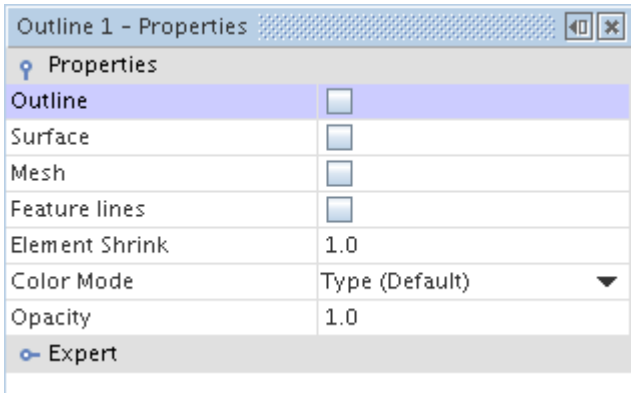


7. Click **OK**.

Visualizing the Free Surface

The **Water Surface** part is now used in creating the scalar scene.

1. Right-click the **Scenes** node, and select **New Scene** > **Scalar**.
2. Select the **Scenes** > **Scalar Scene 1** > **Displayers** > **Outline 1** node and deactivate *Outline*.



3. Select the **Scalar 1** > **Parts** node and set **Parts** to **Derived Parts** > **Water Surface**.
4. Select the **Scalar Field** node and set *Function* to **Volume Fraction** > **Volume Fraction of Water**.
   a) Set *Clip* to **Off**.
   b) Set *Auto Range* to **Off**.
   c) Set *Min* to `0.5`.
   d) Set *Max* to `0.5`.

You can now return to the **Scenes** > **Scalar Scene 1** node to finish setting its properties.

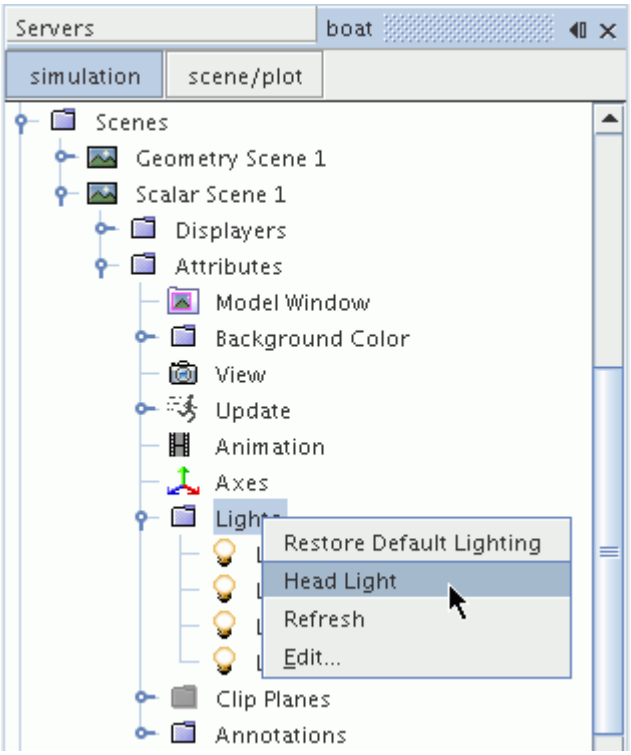
5. Select the **Scenes** > **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Color Bar** node and set *Color Map* to **Water**.
6. Deactivate *Shadow*.
7. Set *Levels* to `2`(as there are only two points on the **Water** color map).
8. Deactivate *Visible* because you are intending to display the free surface as a uniform color: the color bar is not necessary.

Some further changes are required to provide a complete representation of the boat hull.

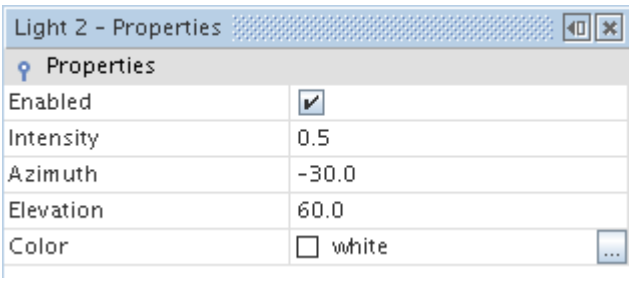
9. Select the **Scenes** > **Scalar Scene 1** > **Displayers** > **Scalar 1** node and set *Contour Style* to **Smooth Filled**.
10. Open the *Expert* properties and set *Transform* to **symmetry 1**.

Some adjustments are made to the scalar scene lighting to improve the appearance.

11. Right-click the **Scenes** > **Scalar Scene 1** > **Attributes** > **Lights** node and select the item **Head Light**. Only the first light remains active.



12. Open the **Lights** node and select the **Light 2** node.
13. Activate *Enabled* for the **Light 2** node.



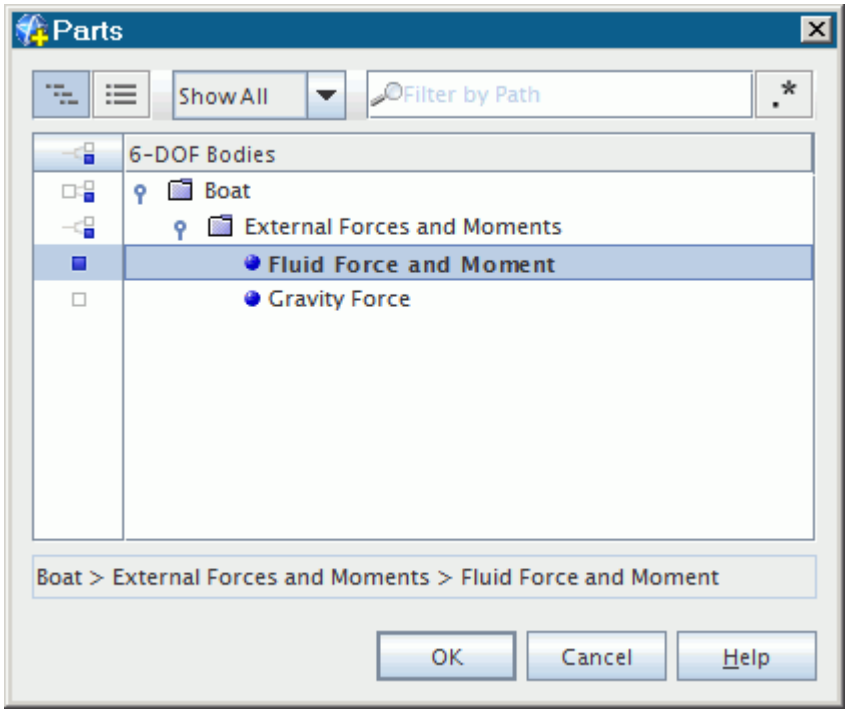
Creating DFBI Displayers

Two displayers are added to the scene to display the center of mass of the boat and the forces and moments acting on it.

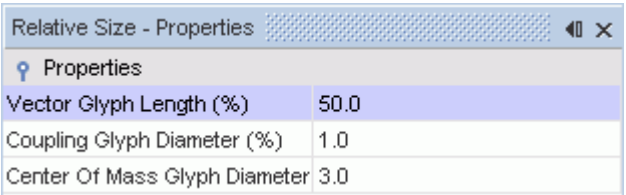
1. Right-click **Scenes** > **Scalar Scene 1** > **Displayers** and select **New Displayer** > **DFBI**
2. Select **DFBI 1** > **Parts** node and set *Parts* to **6-DOF Bodies** > **Boat**.
3. Select the **DFBI 1** node and set *Opacity* to `0.5`
4. Set *Transform* to **symmetry 1**.

The boat and an icon showing the location of the center of mass appear.

5. Right-click **Scenes** > **Scalar Scene 1** > **Displayers** and select **New Displayer** > **DFBI**
6. Select the **DFBI 2** > **Parts** node and set *Parts* to**6-DOF Bodies** > **Boat** > **External Forces and Moments** > **Fluid Force and Moment**.



7. Select **DFBI 2** > **Relative Size** node and set *Vector Glyph Length (%)* to `50`



8. Select **DFBI 2** node and set *Vector Length* to *Vector Magnitude*.

The force and moment are calculated during the run.

Exporting Images for Scene Animation

The final step in preparing for the free surface animation is to have STAR-CCM+ create a hardcopy of *Scalar Scene 1* each time-step.

1. Select the **Scenes** > **Scalar Scene 1** > **Attributes** > **Update** node and set *Trigger* to **Time-Step**.
2. Set the *Output Directory* to point to a suitable folder on your file system where multiple images can be saved.
3. Enter `wave_surface` for the *Base Filename* property. Image files that are created from this scene have `wave_surface` contained in the filename.
4. Activate *Save To File*.

   A red asterisk appears next to the **Scalar Scene 1** node in the object tree.

5. Select the **Update** > **Hardcopy** node and set the following properties:

Property	Setting
Output Width	`800`
Output Height	`600`

6. Save the simulation.

Reporting, Monitoring, and Plotting

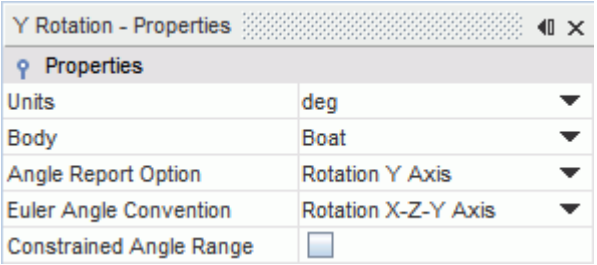
Use report monitors in this simulation to plot the dynamic response of the motion of the boat to the incident waves over time.

1. Right-click the **Reports** node and select **New Report** > **6-DOF Body Translation**.

   A node that is called **6-DOF Body Translation 1** appears within the **Reports** node.
2. Rename the **6-DOF Body Translation 1** node to `Z Translation`
3. Select the **Z Translation** node and set *Direction* to `[0.0, 0.0, 1.0]`.

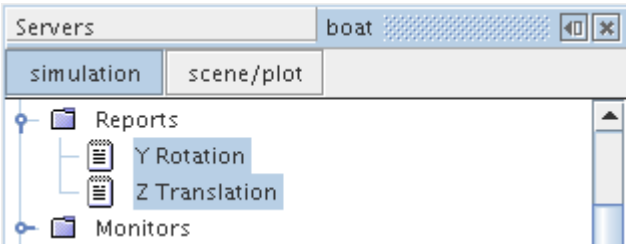
A second report is created to monitor the angle of rotation about the Y axis.

4. Right-click the **Reports** node in the object tree and select **New Report** > **6-DOF Body Orientation**.
5. Rename the **6-DOF Body Angle 1** node to `Y Rotation`
6. Select the **Y Rotation** node and set *Angle Report Option* to `Rotation Y Axis`.



Create monitors and plots that are based on these reports.

7. First select the **Reports** > **Y Rotation** node and then, while pressing down the `Ctrl` key, also select the **Z Translation** node.

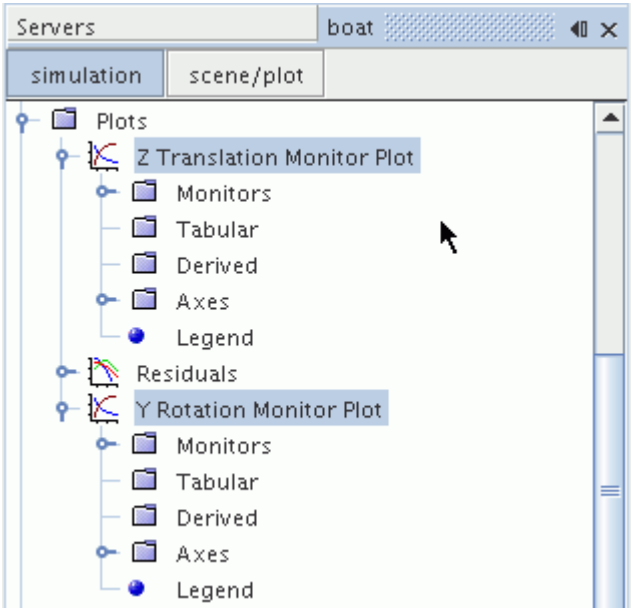


8. Right-click the nodes and select item **Create Monitor and Plot from Report**

The *Create Plot From Reports* dialog appears.

9. Select the **Multiple Plots (one per report)** option.

Two new nodes named **Z Translation Monitor Plot** and **Y Rotation Monitor Plot** appear under the **Plots** node.



Open the plots corresponding to these nodes, to preview.

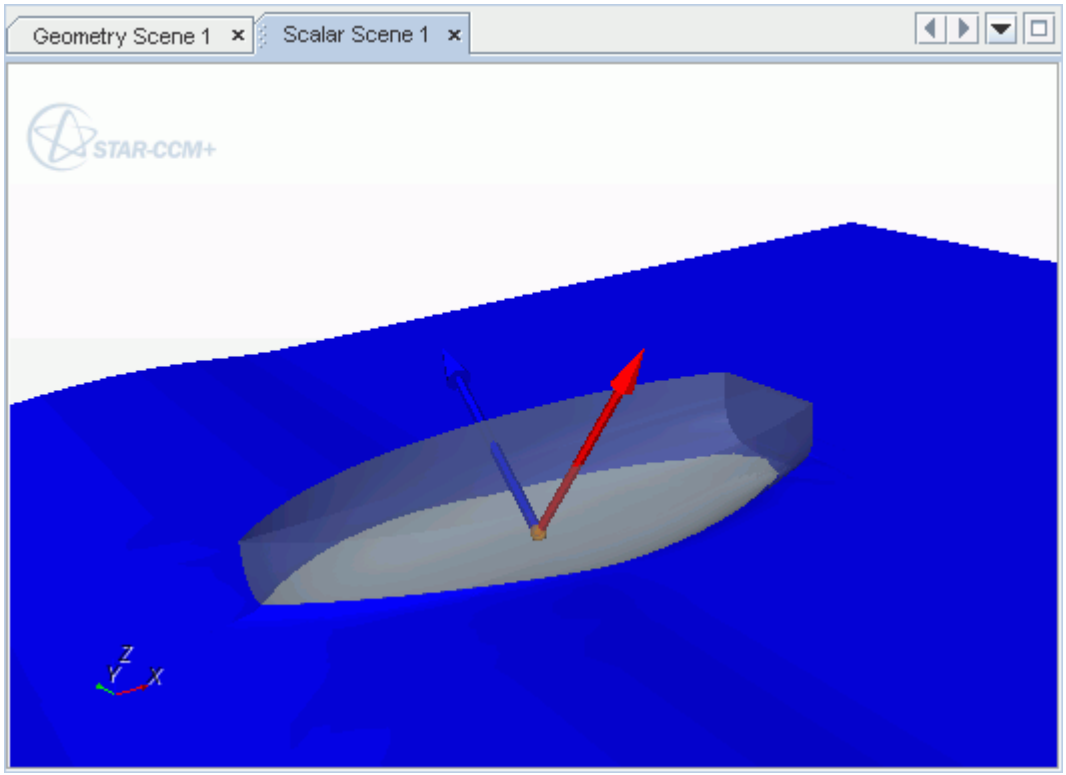
The monitors that provide data for these plots are set to update every iteration by default. As this simulation is unsteady, it is better to update the monitors once per time-step instead at every iteration.

10. Open the **Monitors** node in the simulation tree and select both the **Y Rotation Monitor** and the **Z Translation Monitor** nodes using the `Ctrl Click` method.
11. In the *Properties* window, set *Trigger* to `Time Step`.
12. Save the simulation.

Initializing and Running the Simulation

To verify that the initial conditions applied to the regions are correct, STAR-CCM+ allows you to perform just solver initialization to start with and then view the results using an appropriate scene.

1. To initialize, click the (**Initialize Solution**) button in the toolbar or use the **Solution** > **Initialize Solution** menu item
2. Select the *Scalar Scene 1* window to see the initialization result. Use the mouse buttons to orient and size the model as shown below.



The force (red) and moment (blue) vectors for half of the boat are displayed during the run. To display the vectors for other half, enable the symmetry transform:

3. Select the **DFBI 2** node and set *Transform* to **symmetry 1**.

The force vector, a polar vector, is mirrored at the symmetry plane. The moment vector, an axial vector, maintains its magnitude but changes direction due to reflection. To clear the forces or moments on the body untick the appropriate box in the **DFBI 2**node properties.

To reduce cluttering the scene, view the force and moment for one half of the boat only for this run.

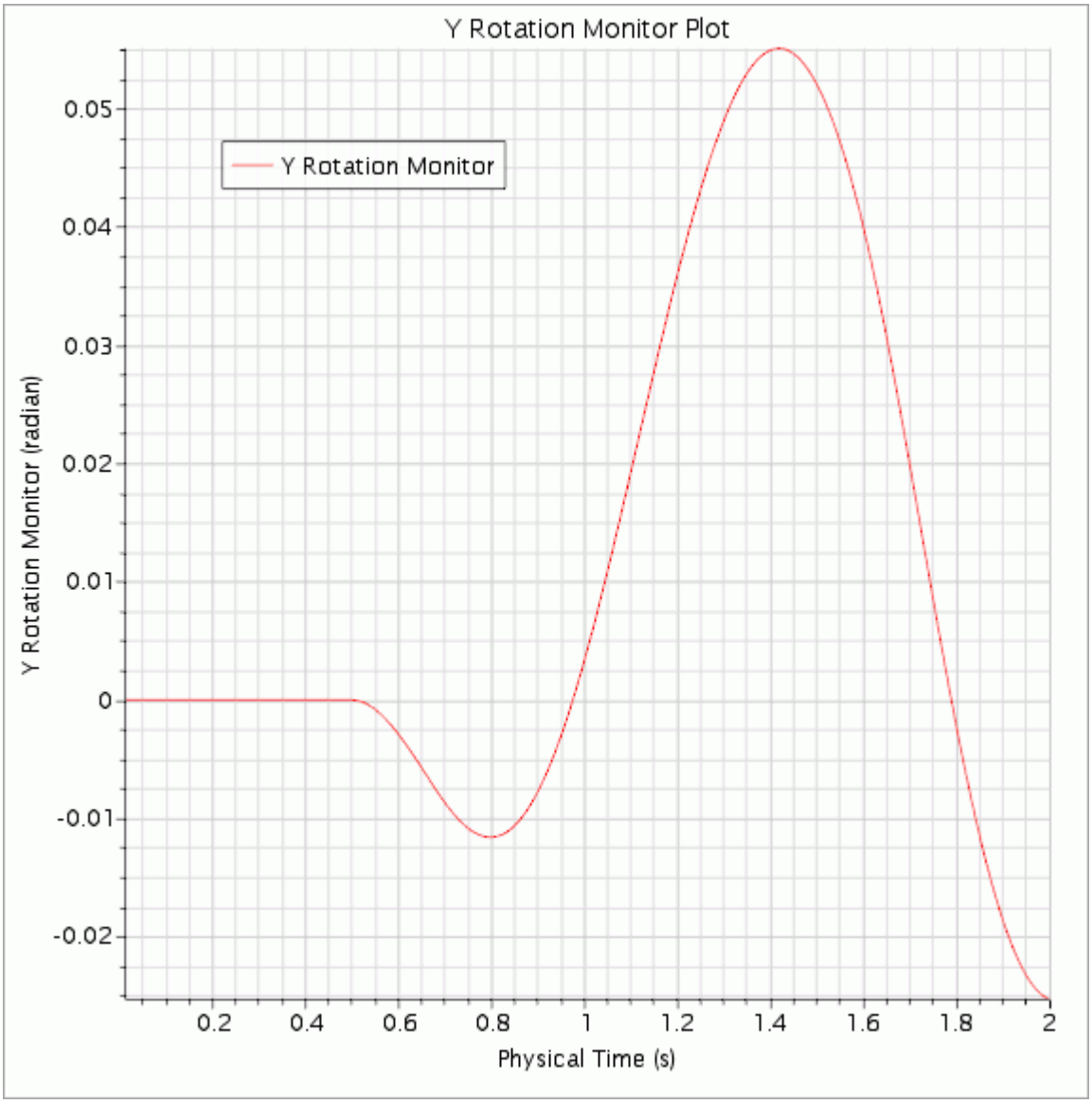
4. Select the **DFBI 2** node:

   Set *Transform* to **Identity**.

   Activate **Forces** and **Moments**.

5. Click the (**Run**) button in the toolbar.

The *Y Rotation Monitor Plot* at the end of the simulation is shown below.



6. After the run is finished, save the simulation.

Visualizing the Results

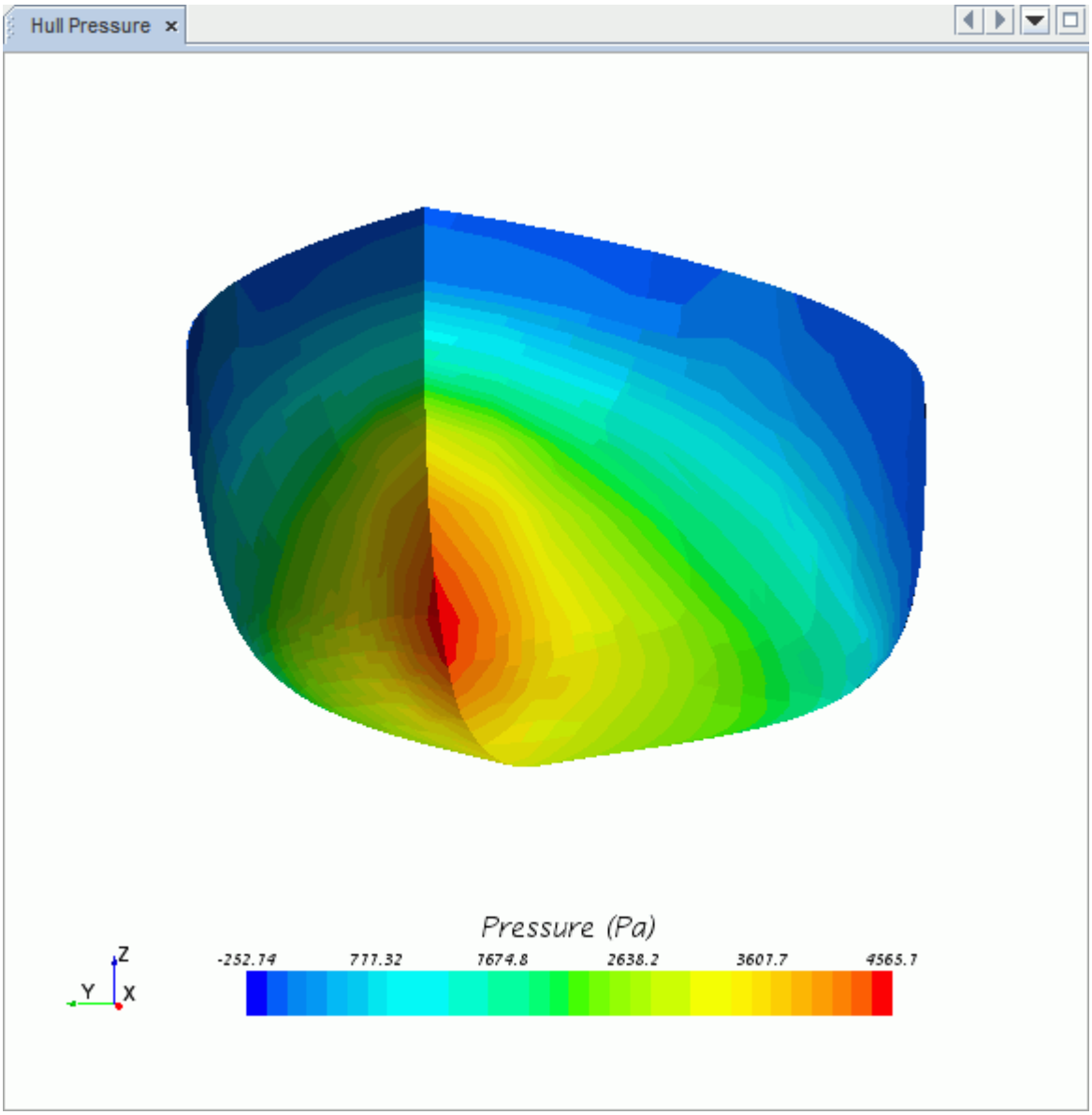
The pressure distribution on the boat hull after 2.0 s is plotted in a scalar scene.

1. Right-click the **Scenes** node and select **New Scene** > **Scalar** from the pop-up menu.
2. Rename the **Scalar Scene 2** node as **Hull Pressure**
3. Select the **Scenes** > **Hull Pressure** > **Displayers** > **Outline 1** node and deactivate *Outline*.

   All visible outlines are removed from the *Hull Pressure* display.
4. Select the **Hull Pressure** > **Displayers** > **Scalar 1** > **Parts** node and click the right-hand side of the *Parts* property.
5. In the dialog that appears, expand the **Regions** and **Region_1** nodes. Select the **hull** node and click **OK**.

6. Select the **Hull Pressure** > **Displayers** > **Scalar 1** > **Scalar Field** node and set *Function* to **Pressure**.
7. Select the **Hull Pressure** > **Displayers** > **Scalar 1** node and set *Contour Style* to **Smooth Filled**.
8. Expand the **Expert** properties and set *Transform* to **symmetry 1**.

Use the mouse controls to zoom and orient the scene as desired. A close-up of the bow is shown below.



9. Save the simulation.

Summary

This tutorial demonstrated use of the Six Degrees Of Freedom solver with a free surface flow to model the motion of a boat hull facing oncoming waves.

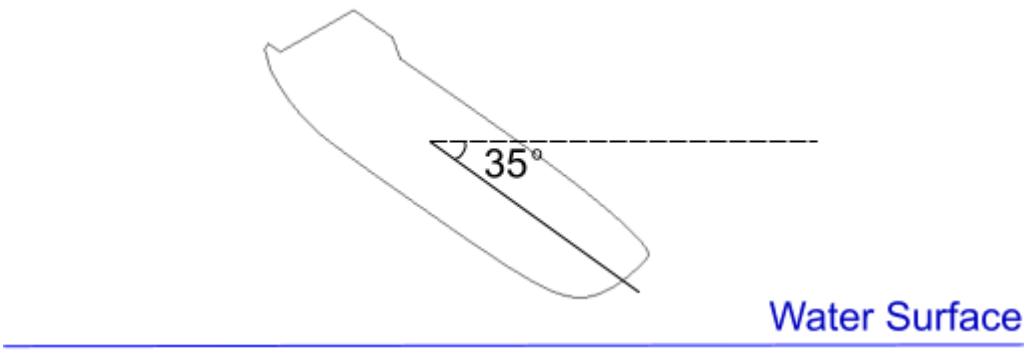
This tutorial introduced the following STAR-CCM+ features:

- Activating the 6-DOF model.
- Creating a VOF wave and using it for initial and boundary conditions.
- Obtaining a visualization of the free-surface flow using a derived part.
- Writing successive images of a scalar scene to disk during a transient simulation.

DFBI: Lifeboat with Overset Mesh

This tutorial demonstrates use of the Overset Mesh feature with free surface flow and DFBI to model the motion of a lifeboat falling into water. STAR-CCM+ automates the grid overlapping process.

The case is inspired by an experiment that is described in the paper “Simulation of Lifeboat Launching under Storm Conditions” by H.J. Morch et al. (2008). A lifeboat that is launched from a ramp undergoes linear and angular acceleration before hitting the water surface.



You model motion with three degrees of freedom: vertical and horizontal translation, and pitching rotation. This setup allows you to use symmetry conditions and include only half the geometry, thus reducing the computational cost of the simulation.

The full fall of the boat from rest on the ramp need not be modeled. Boat motion through the air can be pre-computed using simplified classical physics equations. The simulation is started with the boat initially positioned just above the water surface, with the following properties:

- Mass: 10,000 kg
- Moment of Inertia about axes passing through center of mass: 16000.0 kg m2
- Initial downward velocity: 22 m/s
- Initial angular velocity: 0 rad/s
- Initial inclination: 35

Contents:

Importing the Mesh
Creating Regions and Setting Boundary Types
Selecting Physics Models
Coupling the Regions
Generating the Mesh
Transforming the Overset Region
Creating Eulerian Phases
Setting the DFBI Motion and 6-DOF Body

Defining the VOF Waves
Setting Initial Conditions
Setting the Boundary Conditions
Setting Up Solver Parameters and Stopping Criteria
Setting Scenes to Visualize the Solution
Initializing and Running the Simulation
Summary

Importing the Mesh

You import a CAD model consisting of two parts: a large block for the background region, and a smaller part from which the boat was subtracted to represent the overset region.

1. Start up STAR-CCM+ in a manner that is appropriate to your working environment and create a new simulation.

Import the CAD model:

2. Select **File** > **Import** > **Import Surface Mesh...** and select `lifeBoatOverset.x_b` from the `doc/startutorialsdata/motion/data` subdirectory of your STAR-CCM+ installation.

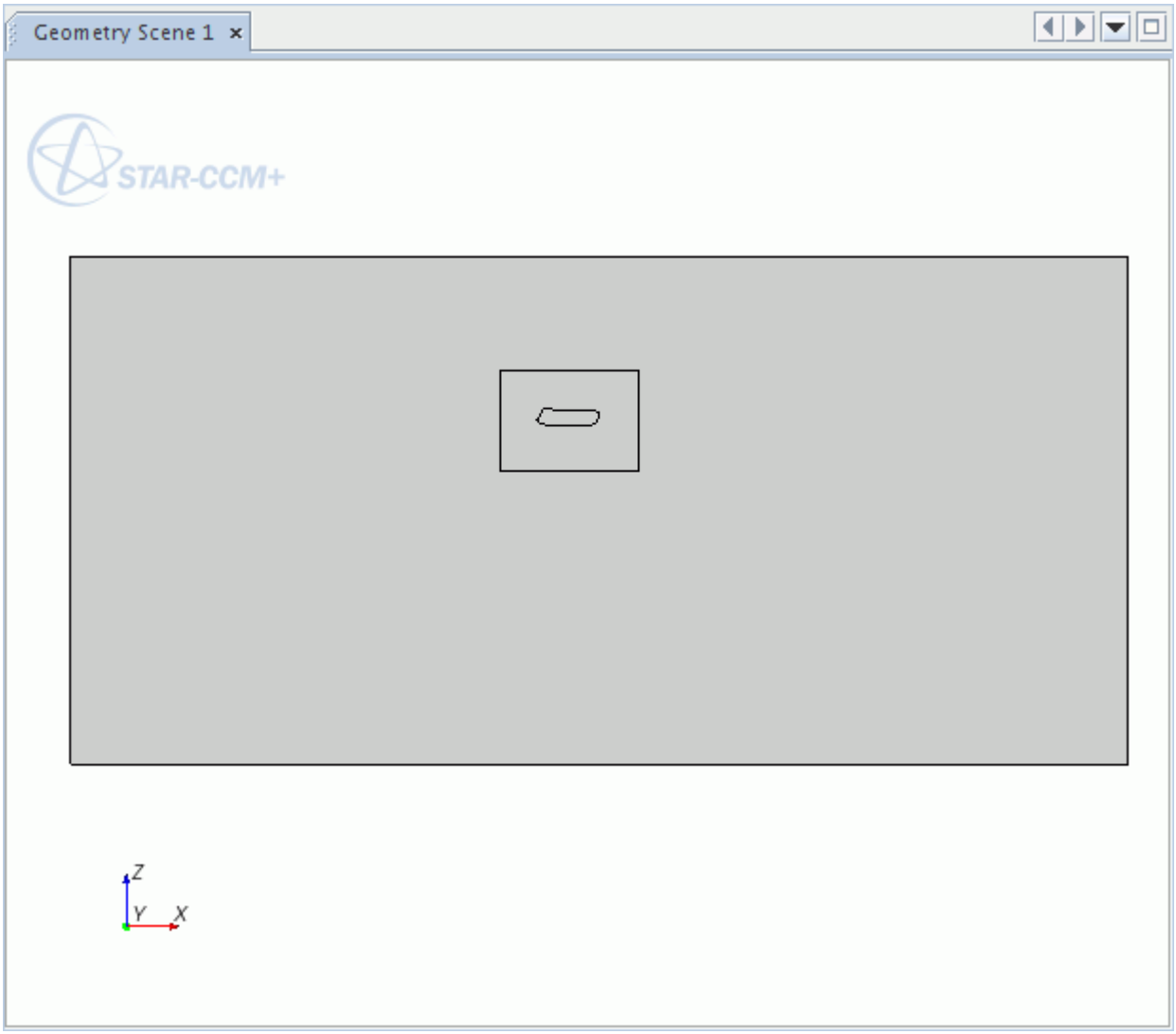
   The *Import Surface Options* dialog appears.

   The default options are appropriate for the case.
3. Click **OK**.

Initially all parts of the imported geometry are shown as solid, gray surfaces with the boat hull hidden from sight. To see the hull shape in its imported orientation:

4. Click (**Save-Restore-Select Views**) and select **Views** > **-Y** > **Up +Z**.

Geometry Scene 1 displays the hull shape together with the outlines of the **Overset** and **Background** parts as shown below.



5. Save the new simulation as `lifeBoat.sim`.

Creating Regions and Setting Boundary Types

Create regions from the **Background** and **Overset** parts.

1. Multi-select the **Geometry** > **Parts** > **Background** and **Overset** nodes.
2. Right-click either selected node and select **Assign Parts to Regions...**.
3. In the *Assign Parts to Regions* dialog:
   a) Set **Region Mode** to **Create a Region for Each Part**.
   b) Set **Boundary Mode** to **Create a Boundary for Each Part Surface**.
   c) Select **Do Not Create Interfaces From Contacts**.
   d) Click **Apply**, and then **Close**.

To be able to create an overset interface between the two regions, assign the `Overset Mesh` type to at least one boundary in the **Overset** region. This boundary type applies to all the boundary surfaces of the

Overset region which are embedded inside the **Background** region, and do not form part of the boat hull. Boundaries of the **Overset** region which are coplanar with boundaries of the **Background** region (in this case, `Symmetry plane`) must be assigned the same boundary type.

Assign the boundary of each region to the correct type:

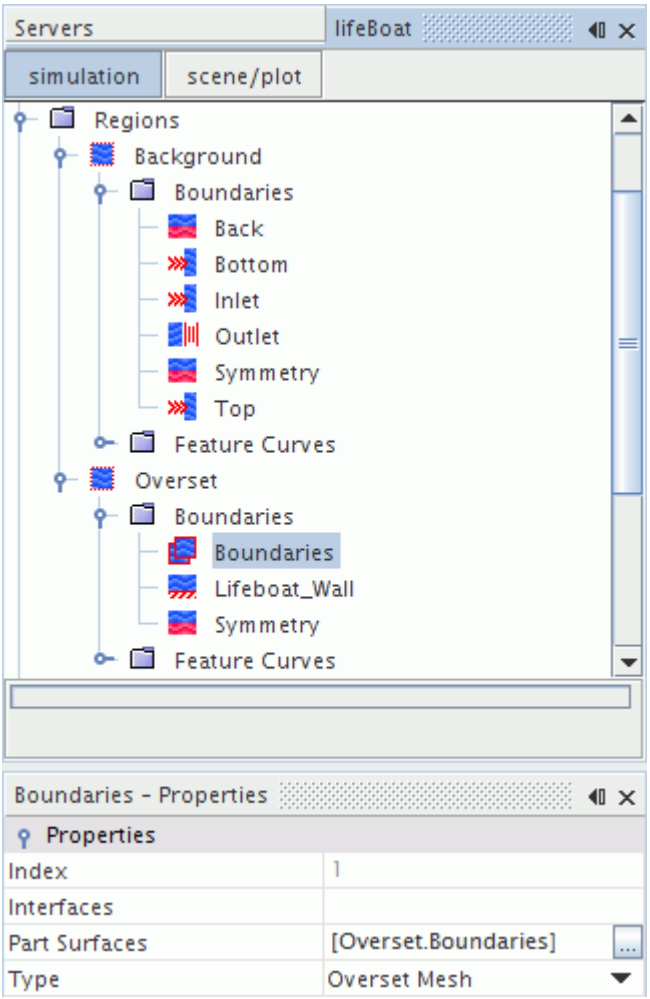
4. Edit the **Regions** > **Background** > **Boundaries** node and set the boundary type for each surface as follows:

Node	Boundary Type
Back	**Symmetry Plane**
Bottom	**Velocity Inlet**
Inlet	**Velocity Inlet**
Outlet	**Pressure Outlet**
Symmetry	**Symmetry Plane**
Top	**Velocity Inlet**

5. Edit the **Regions** > **Overset** > **Boundaries** node and set the boundary type for each surface as follows:

Node	Boundary Type
Boundaries	**Overset Mesh**
Lifeboat_Wall	**Wall**
Symmetry	**Symmetry Plane**

The simulation tree appears as follows:



6. Save the simulation .

Selecting Physics Models

A physics continuum must be present, and assigned to both regions, before the **Overset** region can be coupled to the **Background** region. You create the physics continuum and activate the several physical models that are required to simulate the forces acting on the hull.

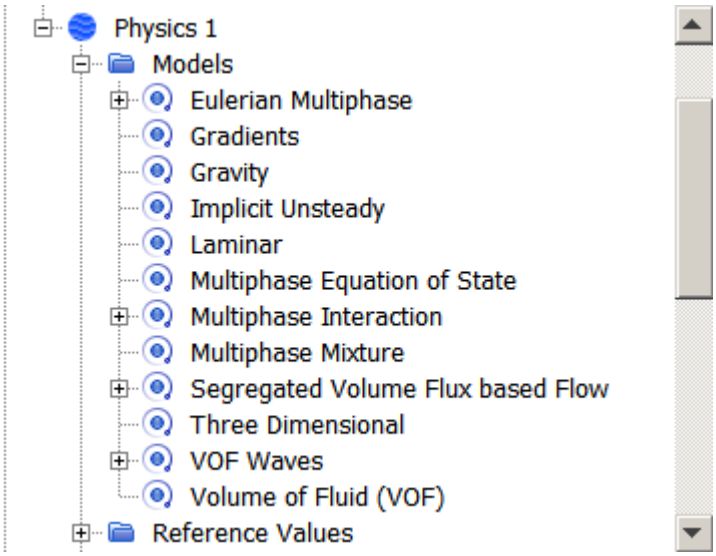
The simulation models the behavior of two fluids (air and water) within the same continuum, and uses the Volume of Fluid model to do so. As there are two fluids in different phases, the Eulerian Multiphase model is activated, and the effect of gravity acting on both is included using the Gravity model. The flow is assumed to be laminar since you are only interested in simulating water entry and impact forces. Simulation of real lifeboat cases over a longer distance would require a turbulence model.

To select the physics models:

1. Create a new physics continuum.
2. For the physics continuum, **Continua** > **Physics 1**, select the following models in order:

Group Box	Model
Space	**Three-Dimensional**
Time	**Implicit Unsteady**
Material	**Eulerian Multiphase Multiphase Interaction** (Selected automatically)
Eulerian Multiphase Model	**Volume of Fluid (VOF)** **Gradients** (Selected automatically) **Segregated Volume Flux based Flow** (Selected automatically) **Multiphase Equation of State** (Selected automatically)
Viscous Regime	**Laminar**
Optional Models	**Gravity** **VOF Waves**

3. Click **Close**.
4. To review the models, open the **Physics 1** > **Models** node.



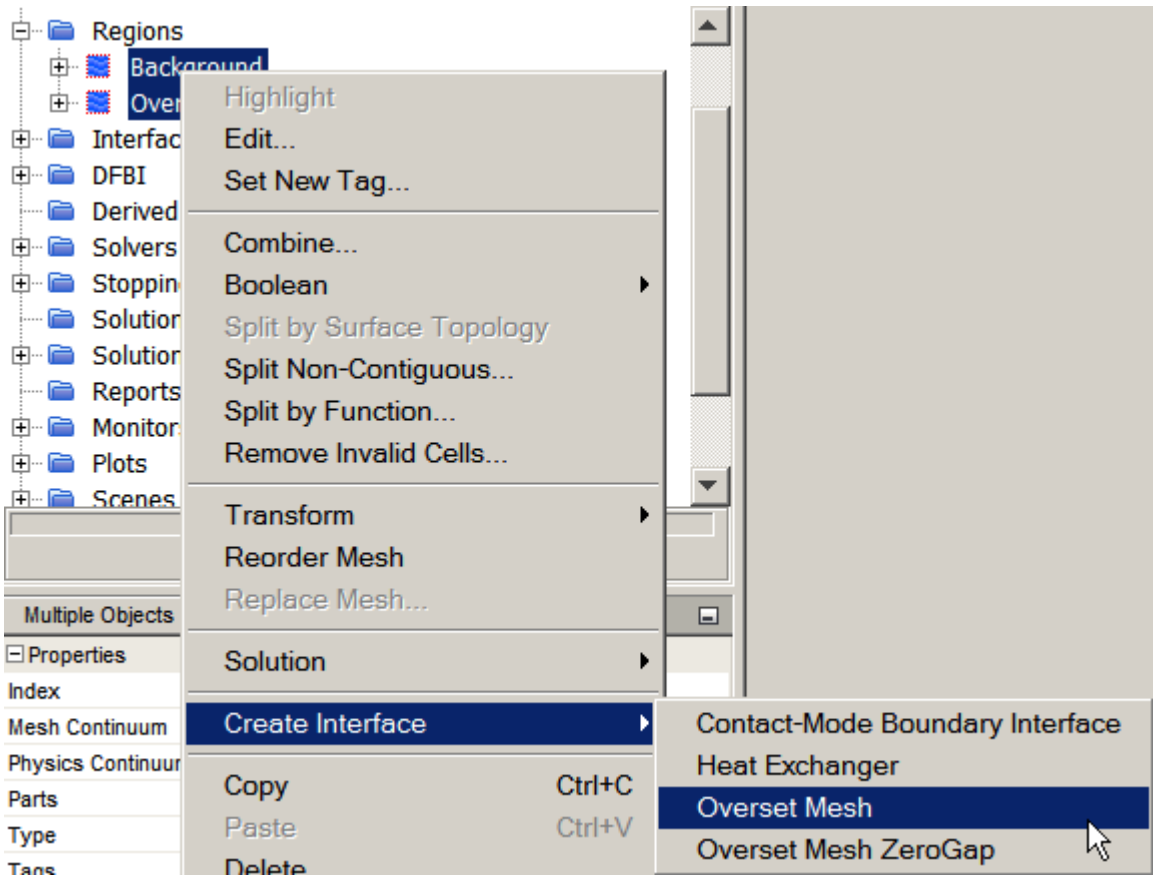
5. Save the simulation .

Coupling the Regions

You must create an interface between the background and the overset regions.

To create the Overset Interfaces:

1. Multi-select the **Regions** > **Background** and **Overset** nodes.
2. Right-click either selected node and select **Create Interface** > **Overset Mesh**.



A new **Interfaces** node appears in the simulation tree, with **Overset Mesh 1** as child node. This volume-type interface provides coupling of solutions on the domains using automatically generated sets of acceptor cells on one mesh and donor cells on the other mesh. Variable values at donor cells express variable values at acceptor cells through interpolation.

The interpolation method can be set as follows:

3. Select the **Interfaces** > **Overset Mesh 1** > **Physics Conditions** > **Interpolation Option** node and set *Option* to `Linear`.

4. Save the simulation .

Generating the Mesh

Proceed with the mesh creation for the case.

1. Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
2. In the *Create Automated Mesh Operation* dialog, select **Background** from the *Parts* list.
3. Select the following meshers in order:

Group Box	Mesher
Surface Meshers	**Surface Remesher**

Group Box	Mesher
Core Volume Meshers	**Trimmed Cell Mesher**

4. Click **OK**.
5. Rename the new **Automated Mesh** node to **Background**.

Create a second automated mesh operation that is called **Overset** by copying the **Background** automated mesh operation.

6. Right-click the **Operations** > **Background** node and select **Copy**.
7. Right-click the **Operations** node and select **Paste**.
8. Rename **Copy of Background** to **Overset**
9. Select **Operations** > **Overset** and set *Input Parts* to **Overset**.

When generating the mesh, pay particular attention to the following:

- To minimize errors in interpolating variables between the two meshes, use the same order of magnitude of grid density in the overlapping region of the **Overset** and **Background** meshes.
- To align the overset mesh with the flow around the lifeboat as it moves through water, generate the mesh with the lifeboat in horizontal position. Rotate the region to its initial position for simulation after the mesh creation.
- The grid must be finer around the boat hull and the water-air interface to capture relevant flow detail.

Set up volumetric controls for each mesh to help you achieve these goals. First create blocks inside which finer meshes are generate

10. Right-click the **Geometry** > **Parts** node and select **New Shape Part** > **Block**.
11. Enter the following coordinates:

	Corner 1	*Corner 2*
X	-28.0m	50.0 m
Y	0.0 m	12.0 m
Z	-45.0 m	0.0 m

12. Click **Create**, and then **Close**.
13. Rename **Geometry** > **Parts** > **Block** to **Overlap**
14. Using the same procedure, create another block that is called **Water Surface** with the following coordinates:

	Corner 1	*Corner 2*
X	-125.0 m	125.0 m
Y	0.0 m	50.0 m
Z	-34.0 m	-24.0 m

15. Create another block that is called **Boat** with the following coordinates:

	Corner 1	*Corner 2*
X	-18.0 m	4.0 m

	Corner 1	*Corner 2*
Y	0.0 m	5.0 m
Z	-4.0 m	8.0 m

Set the base size for the **Background** mesh operation and apply volumetric controls:

16. Select the **Operations** > **Background** > **Default Controls** > **Base Size** node and set *Value* to `5.0 m`
17. Right-click the **Operations** > **Background** > **Custom Controls** node and select **New** > **Volumetric Control**.
18. Rename **Volumetric Control** to **Overlap**.
19. Select the **Custom Controls** > **Overlap** node and set *Parts* to **Overlap**.
20. Select the **Custom Controls** > **Overlap** > **Controls** > **Surface Remesher** node and activate **Customize Size**.
21. Select **Overlap** > **Values** > **Custom Size** and set *Relative to base* to `25.0`, resulting in an *Absolute Size* of `1.25 m`.

    This setting makes sure that the overset and background meshes are of similar order of magnitude where they overlap.
22. Select the **Custom Controls** > **Overlap** > **Controls** > **Trimmer** node and activate **Customize Isotropic Size**.

Create the volumetric control for the zone along the expected water surface:

23. Continuing with settings for the **Background** operation, create a custom Volumetric Control and rename it **Water Surface**.
24. Select the **Custom Controls** > **Water Surface** node and set *Parts* to **Water Surface**.
25. Activate the **Custom Size** selection option for the Surface Remesher and set the **Relative Size** to a *Percentage of Base* of `100.0`.
26. Select the **Water Surface** > **Controls** > **Trimmer** node and activate **Customize Anisotropic Size**.
27. Select the **Water Surface** > **Values** > **Trimmer Anisotropic Size** node and activate **Custom X Size**, **Custom Y Size**, and **Custom Z Size**.
28. Select the **Trimmer Anisotropic Size** > **Relative Z Size** node and set *Percentage of Base* to `50.0`.

Align the mesh to the background region and change the mesh growth settings from the volumetric controls:

29. Select the **Background** > **Meshers** > **Trimmed Cell Mesher** node and activate **Perform Mesh Alignment**.
30. Select the **Background** > **Default Controls** > **Mesh Alignment Location** node and set *Location* to `[-125.0, 0.0, 0.0] m`
31. Select the **Background** > **Default Controls** > **Volume Growth Rate** node and set *Default Growth Rate* to **Slow**.
32. Set *Surface Growth Rate* to **Slow**.

Set the base size for the **Overset** mesh operation and add a volumetric control to refine the region around the lifeboat hull.

33. Select the **Operations** > **Overset** > **Default Controls** > **Base Size** node and set *Value* to `1.0 m`

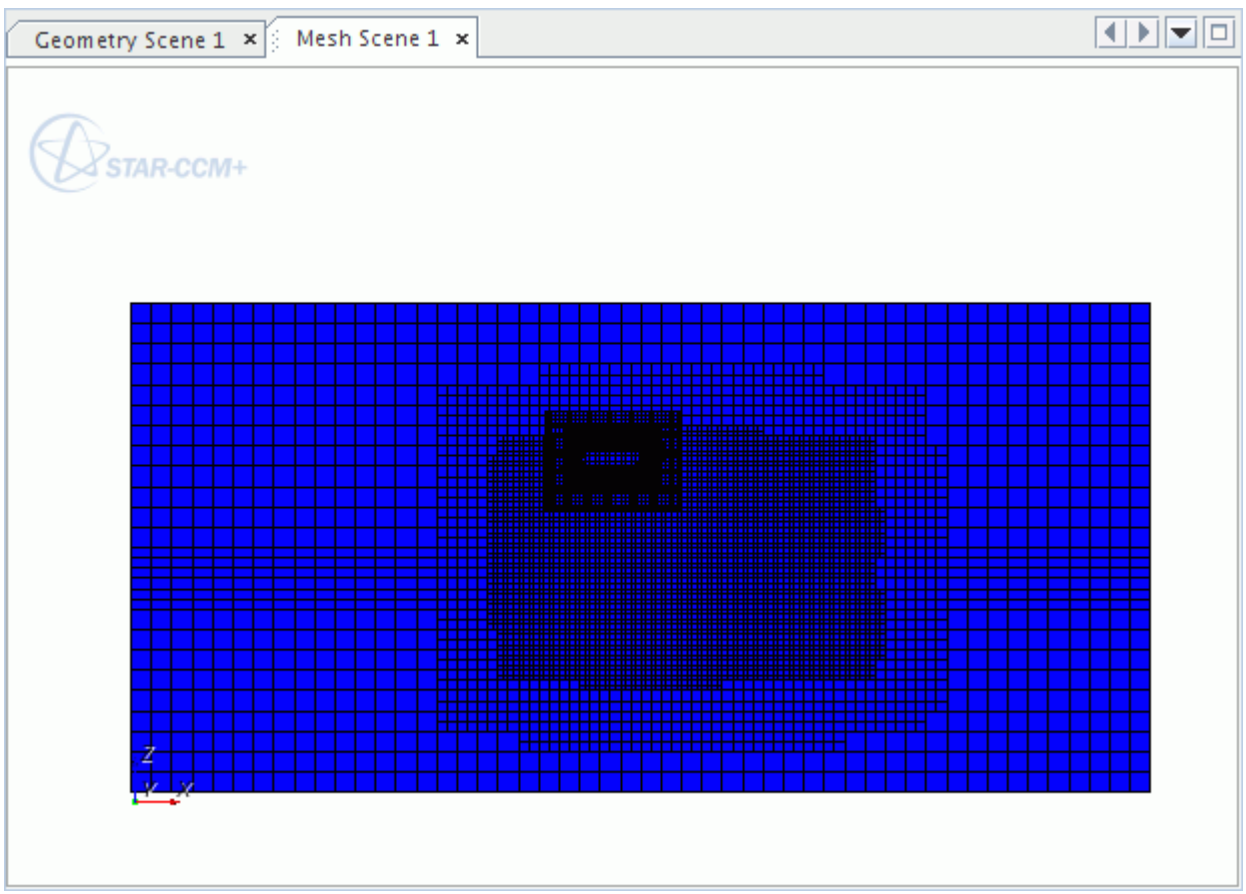
34. Create a custom Volumetric Control for the **Overset** mesh operation and rename it **Boat**.
35. Select **Custom Controls** > **Boat** and set *Parts* to **Boat**.
36. Select the **Custom Controls** > **Boat** > **Controls** > **Surface Remesher** node and activate **Customize Size**.
37. Select the **Boat** > **Values** > **Custom Size** node and set *Relative to base* to `50.0`.

Generate the mesh:

38. Click (**Generate Volume Mesh**) in the toolbar or select the **Generate Volume Mesh** item in the **Mesh** menu.

The mesh can be visualized by creating a Mesh Scene:

39. Right-click **Scenes** and select **New Scene** > **Mesh**.
40. Click (**Save-Restore-Select Views**) and select **Views** > **-Y** > **Up +Z** to display the mesh.



The current mesh settings have been chosen for the tutorial to run in about 30 minutes on a 2.5 GHz processor. If you want to resolve wave propagation accurately, use a finer mesh along the water surface.

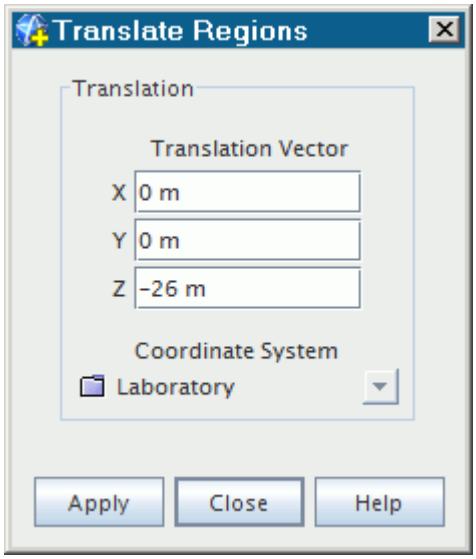
41. Save the simulation .

Transforming the Overset Region

Translate the lifeboat nearer to the water surface to decrease the simulation time and give it an initial tilt with a horizontal angle of 35.

1. Right-click **Regions** > **Overset** and select **Transform** > **Translate...**.

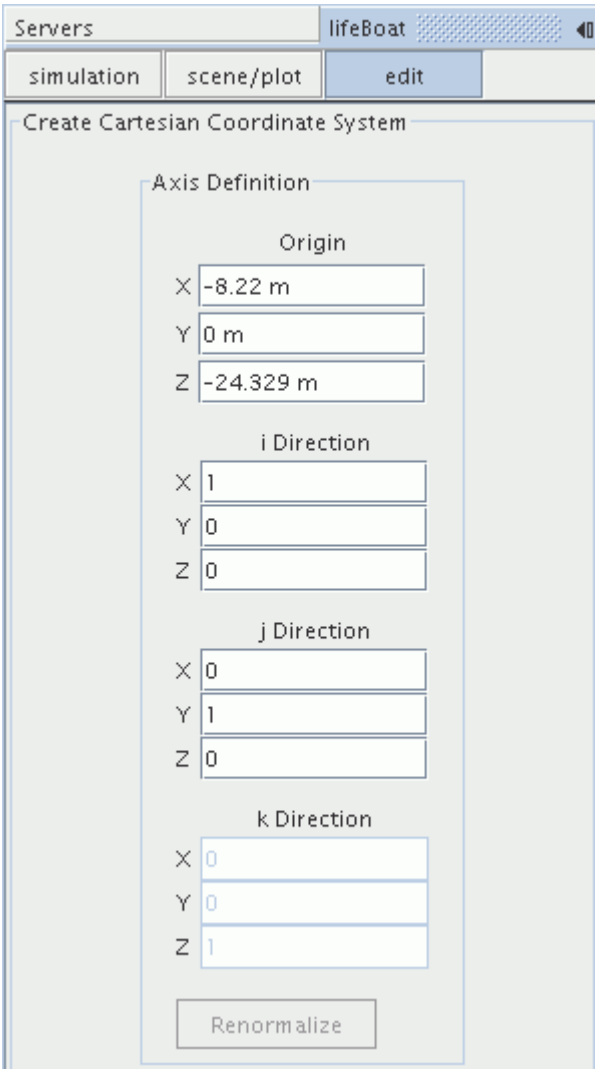
   The *Translate Regions* dialog appears.
2. Set the **Translation Vector** to `[0 m, 0 m, -26 m]` as shown below, and click **Apply** and then **Close**.



Before rotating the lifeboat, specify the coordinate system in which the boat is rotated. Create a coordinate system whose origin is taken as the center of mass of the boat after the translation.

3. Right-click the **Tools** > **Coordinate Systems** > **Laboratory** > **Local Coordinate Systems** node and select **New** > **Cartesian**.

   The *Create Cartesian Coordinate System* panel appears.
4. Set **Origin** to `[-8.22, 0.0 -24.329] m, m, m`.



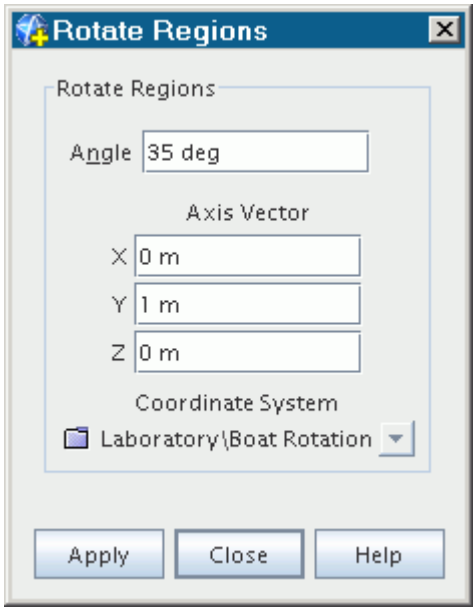
5. Click **Create** and then **Close**.
6. Rename the **Cartesian 1** node to **Boat Rotation**.

Rotate the **Overset** region:

7. Right-click **Regions** > **Overset** and select **Transform** > **Rotate...**.

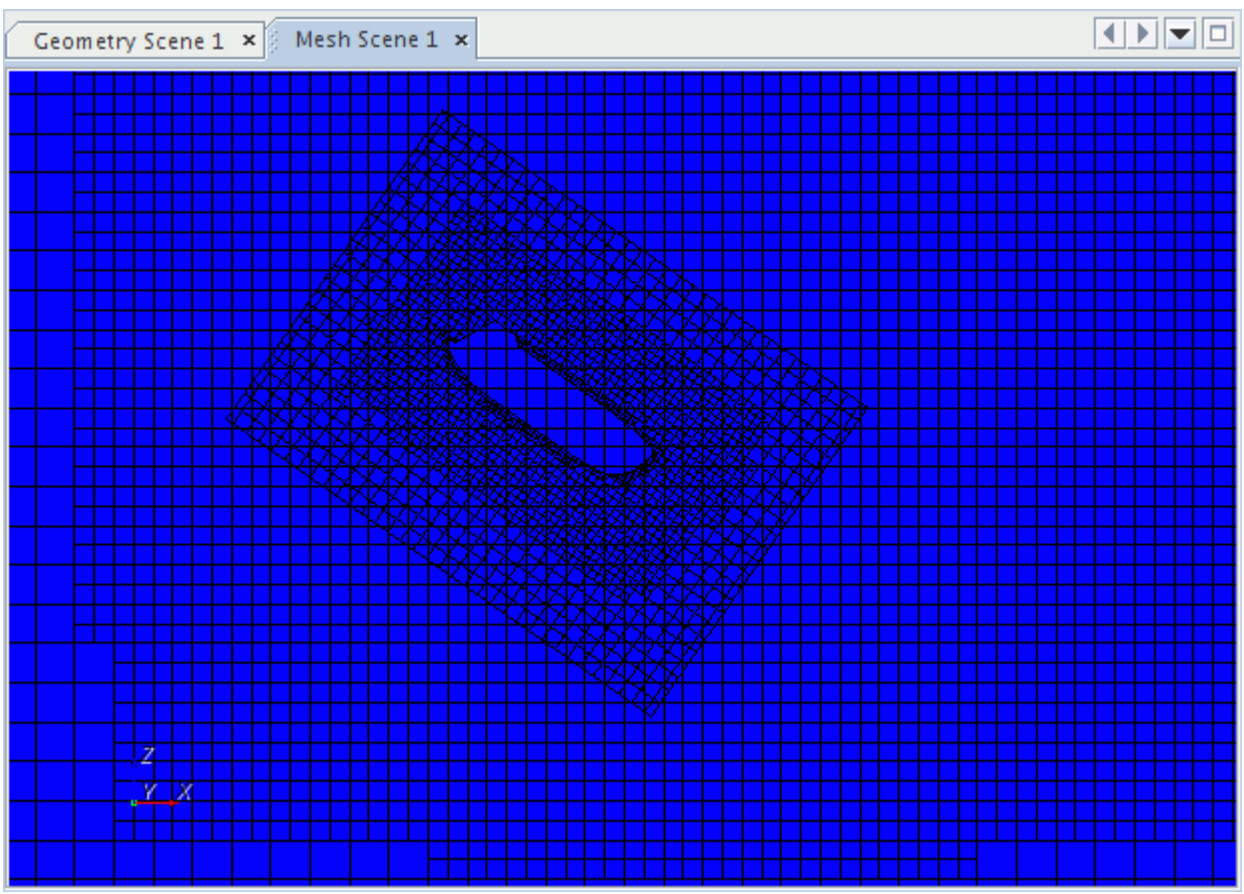
In the *Rotate Regions* dialog, do the following:

8. Set the **Angle** to `35 deg`.
9. Set the **Axis Vector** to `[0 m, 1 m, 0 m]`.
10. Set the **Coordinate System** to `Laboratory\Boat Rotation`.



11. Click **Apply** and then **Close**.

The mesh now appears as follows:



12. Save the simulation .

Creating Eulerian Phases

The materials corresponding to each of the mixture phases are defined as Eulerian phases.

Create the Eulerian phases that are required for the simulation and set their material properties:

1. Right-click the **Continua** > **Physics 1** > **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and select **New**.

   A new node, **Phase 1**, is created.
2. Rename the **Phase 1** node to `Water`.

Define the models that determine the behavior of the *Water* phase.

3. Right-click the **Eulerian Phases** > **Water** > **Models** node and choose **Select models...**

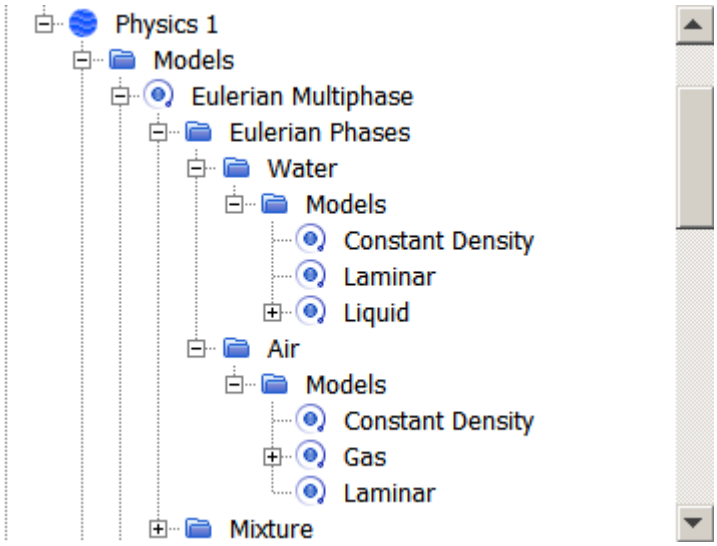
In the *Phase Model Selection* dialog:

4. Select **Liquid** from the *Material* group box.
5. Select **Constant Density** from the *Equation of State* group box, then click **Close**.

Repeat the above steps to create a second Eulerian phase.

6. Rename the second phase to **Air**.
7. Choose **Gas** rather than **Liquid** for the material model. Select **Constant Density** as above, then click **Close**.

In both instances, you are using the default material appropriate to each phase (water for the liquid and air for the gas), so no further changes are necessary to material properties. The model selections can be seen by expanding the **Water** and **Air** nodes within the **Eulerian Phases** parent node.



8. Save the simulation .

Setting the DFBI Motion and 6-DOF Body

The 6-DOF motion solver is activated when creating the relevant DFBI motion object and assigning it to the fluid region. For this tutorial, you require the DFBI rotation and translation motion.

1. Open the **Tools** > **Motions** node.

   The default **Stationary** motion is shown.
2. Right-click the **Motions** node and select **New** > **DFBI Rotation and Translation**.

A new **DFBI Rotation and Translation** node is added to the **Motion** manager.

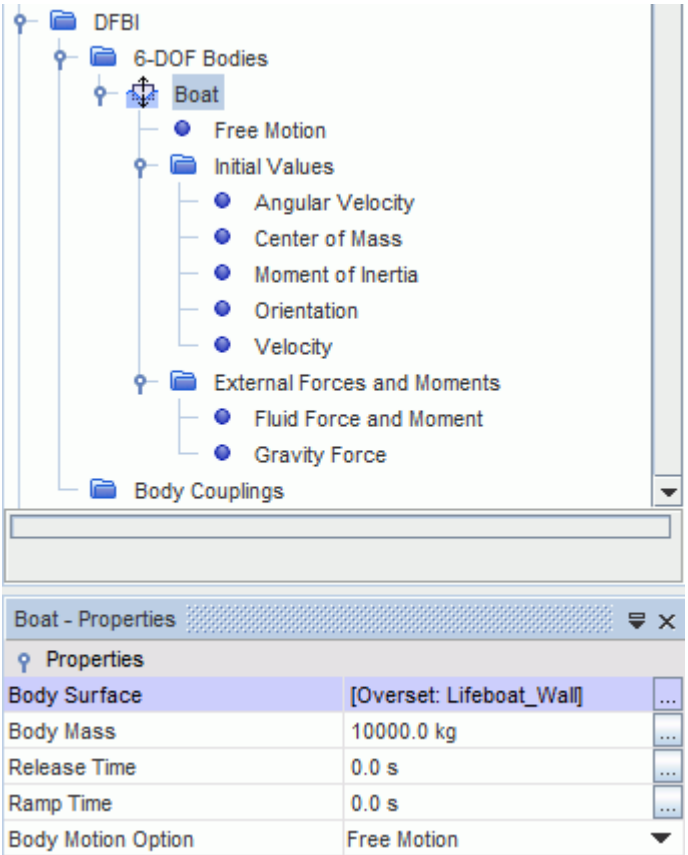
Select this motion within the **Motion Specification** node for the Overset region.

3. Select the **Regions** > **Overset** > **Physics Values** > **Motion Specification** node and set *Motion* to **DFBI Rotation and Translation**.

Create the **Boat** body and set its properties.

4. Right-click the **DFBI** > **6-DOF Bodies** node and select **New** > **3D Body**.
5. Rename **Body 1** to **Boat**.
6. Select the **DFBI** > **6-DOF Bodies** > **Boat** node and click ... (**Custom Editor**) for *Body Surface*.

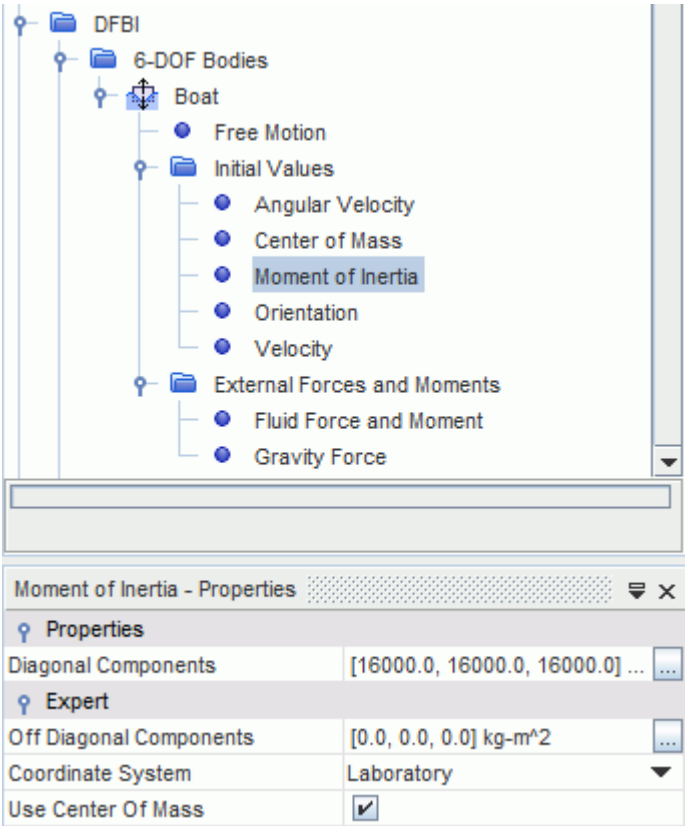
   The *Boat - Body Surface* dialog appears.
7. Select **Overset** > **Boundaries** > **Lifeboat_wall**, then click **OK**.
8. Set Body Mass to `10000.0 kg`.
9. Keep the default values for all other properties.



In this simulation, the body is allowed to rotate about the y-axis only, to simulate motion in the head waves. This means that the moment of inertia about the y-axis must be defined.

10. Select the **DFBI** > **6-DOF Bodies** > **Boat** > **Initial Values** > **Moment of Inertia** node and do the following:
    a) Set *Diagonal Components* to `[16000.0, 16000.0, 16000.0] kg-m^2`
    b) Activate **Use Center Of Mass**.

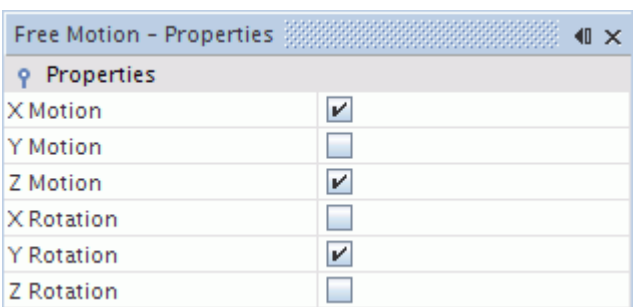
    The moments of inertia are evaluated with respect to the center of mass.



11. Select the **Initial Values** > **Center of Mass** node and set *Position* to `[-8.22, 0.0, -24.329] m`.
12. Select the **Initial Values** > **Velocity** node and set *Value* to `[0.0, 0.0, -22.0] m/s`.

Although the 6-DOF body has now been created with the Free Motion option, it is still necessary to specify the manner in which the body can move. In this case, the body is free to translate in the X direction (surge), Z direction (heave), and to rotate about the Y axis (pitch).

13. Select the **DFBI** > **6-DOF Bodies** > **Boat** > **Free Motion** node and activate the *X Motion*, *Z Motion*, and *Y Rotation* properties.



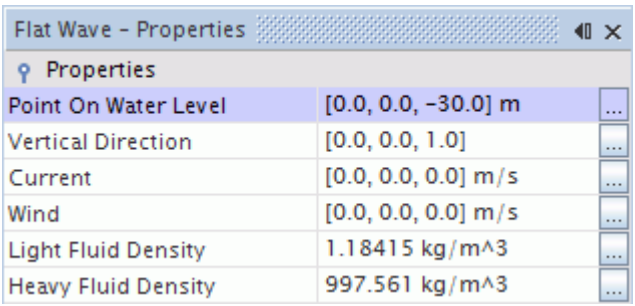
14. Save the simulation .

Defining the VOF Waves

The VOF Wave model generates a number of field functions that are based on the wave parameters that you define. These field functions can be used as boundary and initial conditions in appropriate nodes on the object tree.

To create the VOF Wave:

1. Right-click the **Continua** > **Physics 1** > **Models** > **VOF Waves** > **Waves** node and select **New** > **Flat**.
2. Rename the **Flat Vof Wave 1** node to `Flat Wave`.
3. Select the **Flat Wave** node and set *Point on Water Level* to `[0.0, 0.0, -30] m`.



Note: The *Light Fluid Density* and *Heavy Fluid Density* properties must be the same as those specified for the Eulerian phases **Water** and **Air**.

Setting Initial Conditions

The initial conditions are set using field functions that are associated with the previously defined VOF Wave. Initial conditions that are set are the free-surface wave profile, the velocity profile within the phases and the hydrostatic pressure.

1. Edit the **Continua** > **Physics 1** > **Initial Conditions** node and set the following properties:

Node	Property	Setting
Volume Fraction	*Method*	**Composite**
└ **Composite** > **Water**	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Heavy Fluid of Flat Wave**
└ **Composite** > **Air**	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Light Fluid of Flat Wave**
Velocity	*Method*	**Field Function**
	Vector Function	**Velocity of Flat Wave**

Node	Property	Setting
Pressure	*Method*	**Field Function**
	Scalar Function	**Hydrostatic Pressure of Flat Wave**

2. Save the simulation .

Setting the Boundary Conditions

The boundary types have been correctly set previously, so it only remains to set the conditions corresponding to each type.

There are four boundaries for this geometry:

- No-slip wall boundary on the hull of the boat
- Symmetry boundary on the Z-X plane at Y = 0.0 and Y = 50.0
- Pressure outlet on the right face
- Velocity inlet on all remaining boundaries

Default conditions are suitable for the symmetry plane and the no-slip wall boundary on the boat hull. Wave conditions are applied to the pressure outlet and velocity inlet. Nothing is prescribed at the Overset Mesh boundaries.

Contents:

Inlet Boundary
Outlet Boundary
Bottom Boundary
Top Boundary

Inlet Boundary

You specify the velocity and the volume fraction at the inlet boundary.

To specify the inlet boundary:

1. Edit the **Regions** > **Background** > **Boundaries** > **Inlet** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Velocity Specification**	*Method*	**Components**
Physics Values		
└ **Velocity**	*Method*	**Field Function**
	Vector Function	**Velocity of Flat Wave**

Node	Property	Setting
└ **Volume Fraction**	*Method*	**Composite**
└ **Composite** > **Water**	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Heavy Fluid of Flat Wave**
└ **Composite** > **Air**	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Light Fluid of Flat Wave**

Outlet Boundary

At the oulet boundary, you set the pressure and the volume fraction to correspond to the outgoing wave by using field functions.

To specify the outlet boundary:

1. Edit the **Regions** > **Background** > **Boundaries** > **Outlet** node and set the following properties:

Node	Property	Setting
Physics Values		
└ **Pressure**	*Method*	**Field Function**
	Scalar Function	**Hydrostatic Pressure of Flat Wave**
Volume Fraction	*Method*	**Composite**
└ **Composite** > **Water**	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Heavy Fluid of Flat Wave**
└ **Composite** > **Air**	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Light Fluid of Flat Wave**

Bottom Boundary

The following boundary conditions are appropriate for deep water conditions. A bottom boundary can also be a wall for shallow waters.

To specify the bottom boundary:

1. Edit the **Regions** > **Background** > **Boundaries** > **Bottom** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Velocity Specification**	*Method*	**Components**
Physics Values		
└ **Velocity**	*Method*	**Field Function**
	Vector Function	**Velocity of Flat Wave**
└ **Volume Fraction**	*Value*	`[1.0, 0.0]`

Top Boundary

You specify the velocity and the volume fraction at the top boundary.

To specify the top boundary:

1. Edit the **Regions** > **Background** > **Boundaries** > **Top** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Velocity Specification**	*Method*	**Components**
Physics Values		
└ **Velocity**	*Method*	**Field Function**
	Vector Function	**Velocity of Flat Wave**
Volume Fraction	*Value*	`[0.0, 1.0]`

Setting Up Solver Parameters and Stopping Criteria

As this simulation is transient, it is necessary to set the time step, the maximum number of inner iterations that are permitted in each time step, and the overall physical time for which a solution is obtained.

The time step value is selected as 0.0025 s so that, for any time step, the overset mesh moves by a distance smaller than the height of the smallest cell in the **Background** region. A finer mesh would require an even smaller time step.

1. Select the **Solvers** > **Implicit Unsteady** node and set the *Time-Step* to `0.0025 s`.

A second order time discretization is required for simulations requiring accurate wave propagation. If modeling wave propagation is not important, a first order time method can be used. Second order methods require smaller time steps.

2. Select the **Stopping Criteria** > **Maximum Inner Iterations** node and make sure that *Maximum Inner Iterations* is set to `5`.
3. Select the **Stopping Criteria** > **Maximum Physical Time** node and set *Max Physical Time* to `0.5 s`.

4. Select the **Stopping Criteria** > **Maximum Steps** node and deactivate *Enabled*.
5. Save the simulation .

Setting Scenes to Visualize the Solution

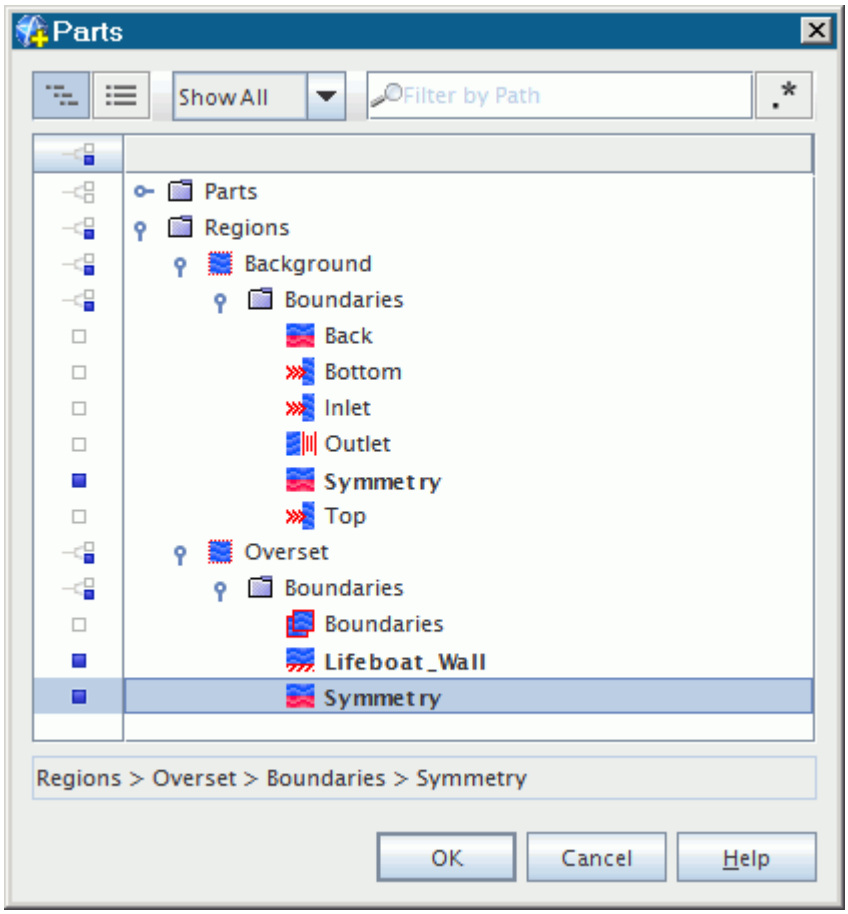
Create scenes to visualize the results as the simulation progresses.

To create a scalar scene:

1. Right-click the **Scenes** node and select **New Scene** > **Scalar**.
2. Rename the **Scalar Scene 1** node to **Volume Fraction of Water**.
3. Click (**Save-Restore-Select Views**) and select **Views** > **-Y** > **Up +Z**.
4. Select the **Scenes** > **Volume Fraction of Water** > **Displayers** > **Scalar 1** > **Parts** node and click custom editor ... to the right of the *Parts* property.

The *Parts-Parts* dialog appears.

5. Select the **Regions** > **Background** > **Symmetry**, **Regions** > **Overset** > **Lifeboat_Wall** and **Regions** > **Overset** > **Symmetry** nodes.



6. Click **OK** to close the dialog.

7. Expand **Scenes** > **Volume Fraction of Water** > **Displayers** > **Scalar 1**, select **Scalar Field**, and then set **Volume Fraction** > **Volume Fraction of Water** for the *Function* property.

To display overset cells, create an additional scene:

8. Click **Scenes** > **Volume Fraction of Water** and while holding the left mouse button, press `CTRL` and drag the **Volume Fraction of Water** node onto the **Scenes** node.
9. Rename the new node to `Overset Cells`.
10. Expand **Scenes** > **Overset Cells** > **Displayers** > **Scalar 1** and select **Scalar Field** and set `Overset Cell Type` for the *Function* property.

You are now ready to run the simulation.

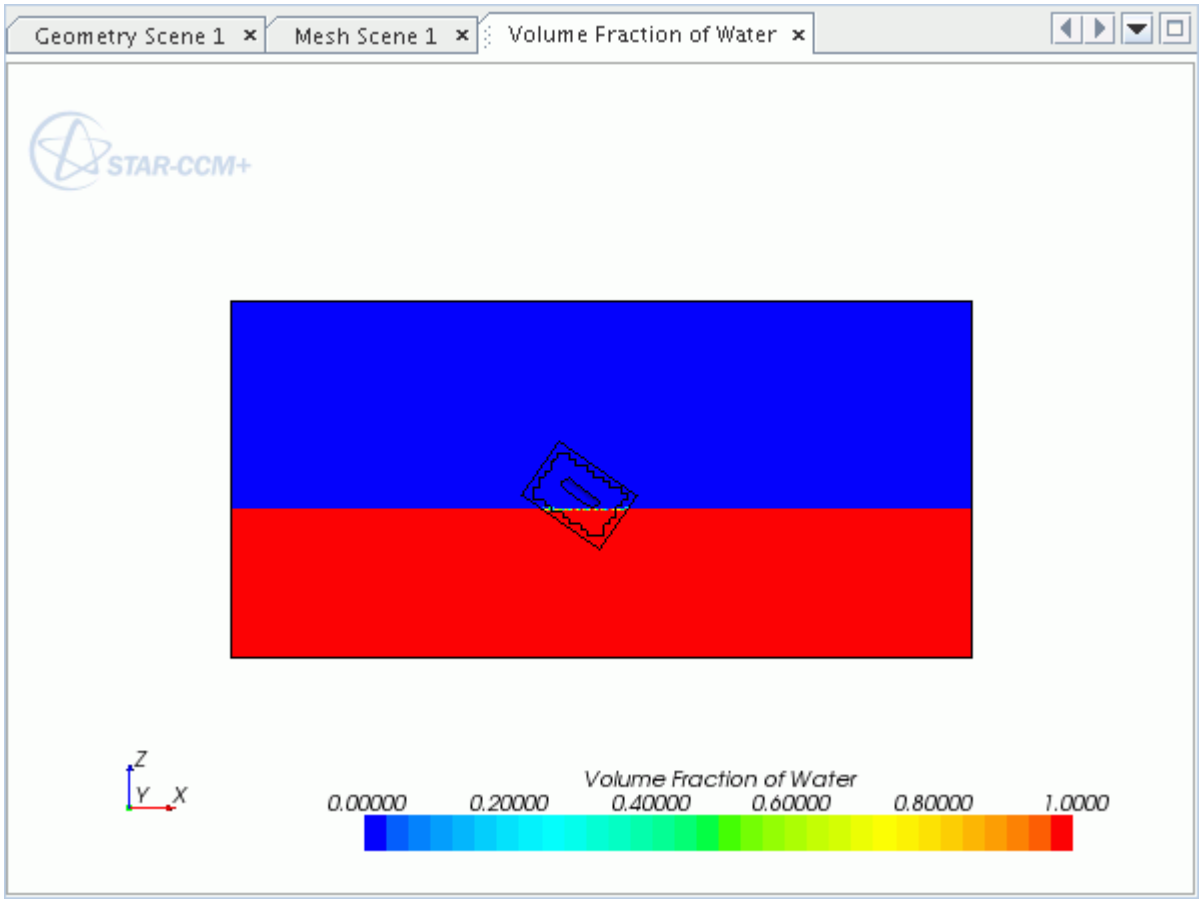
11. Save the simulation .

Initializing and Running the Simulation

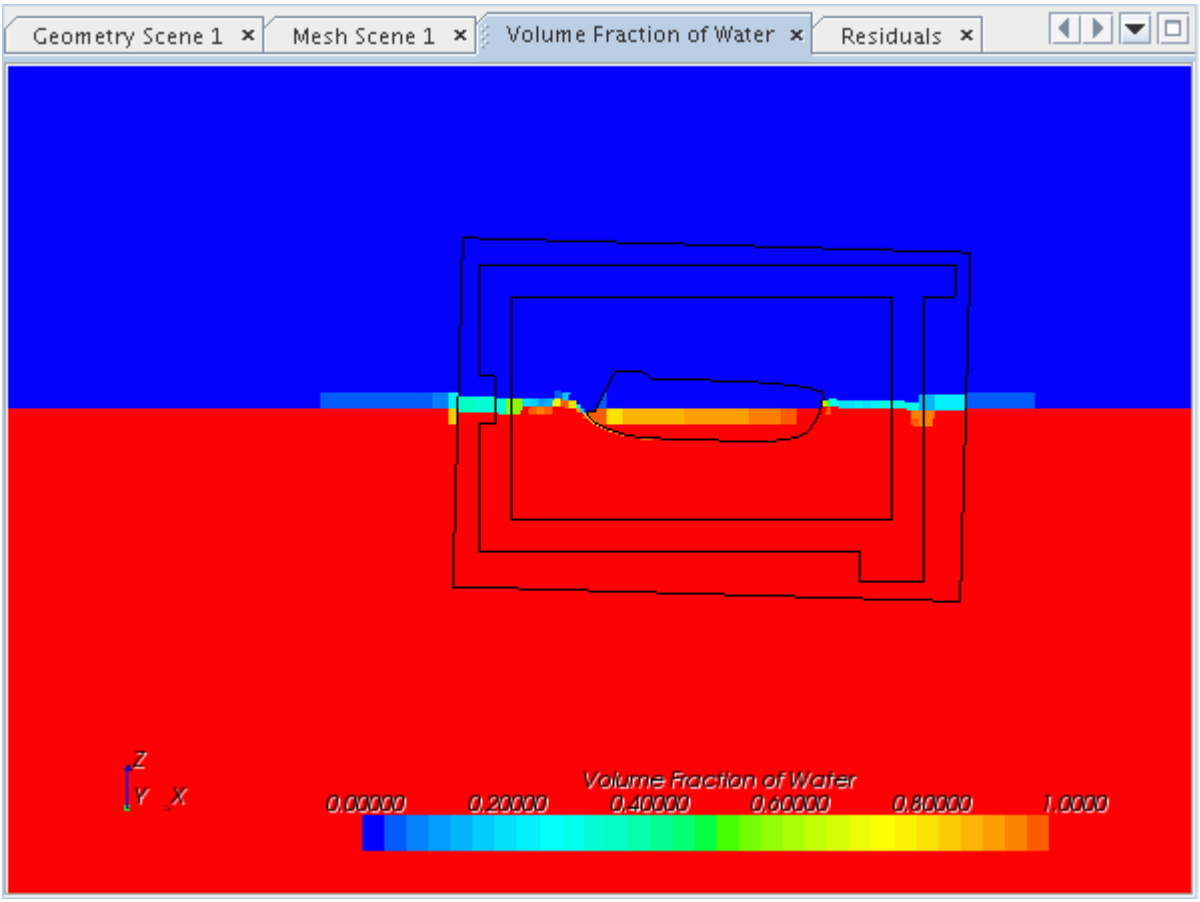
To verify that the initial conditions applied to the regions are correct, STAR-CCM+ allows you to perform just solver initialization to start with and then view the results using an appropriate scene.

To initialize:

1. Click (**Initialize Solution**) in the toolbar or use the **Solution** > **Initialize Solution** menu item.
2. Select the *Volume Fraction of Water* scene to see the initialization result.

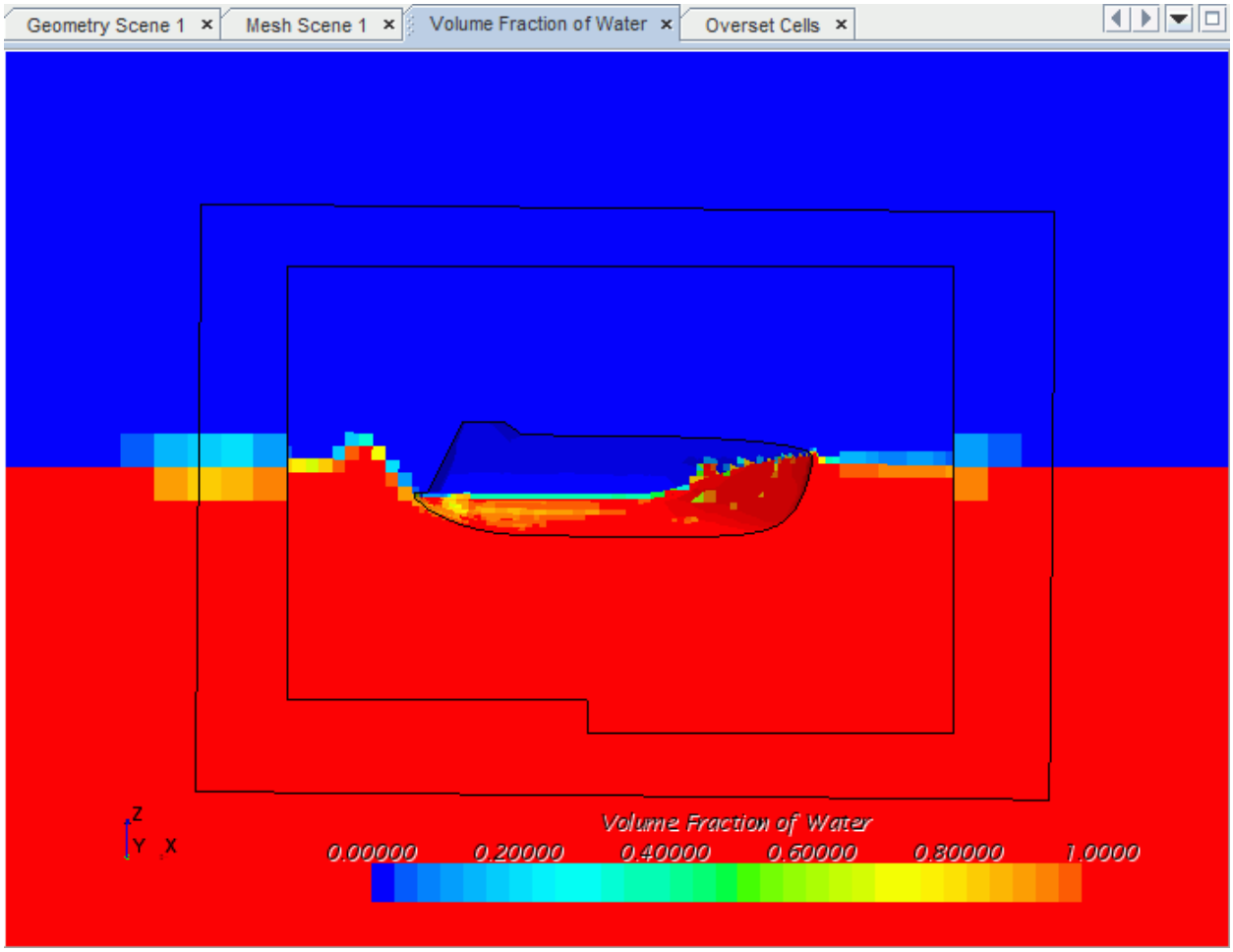


3. Click (**Run**) in the toolbar.
4. View the evolution of the *Volume Fraction of Water* scene while the simulation is running.



5. After the run is finished, save the simulation.

   The example below shows the *Volume Fraction of Water* scene at the end of the simulation:



Summary

This tutorial introduced the following STAR-CCM+ features:

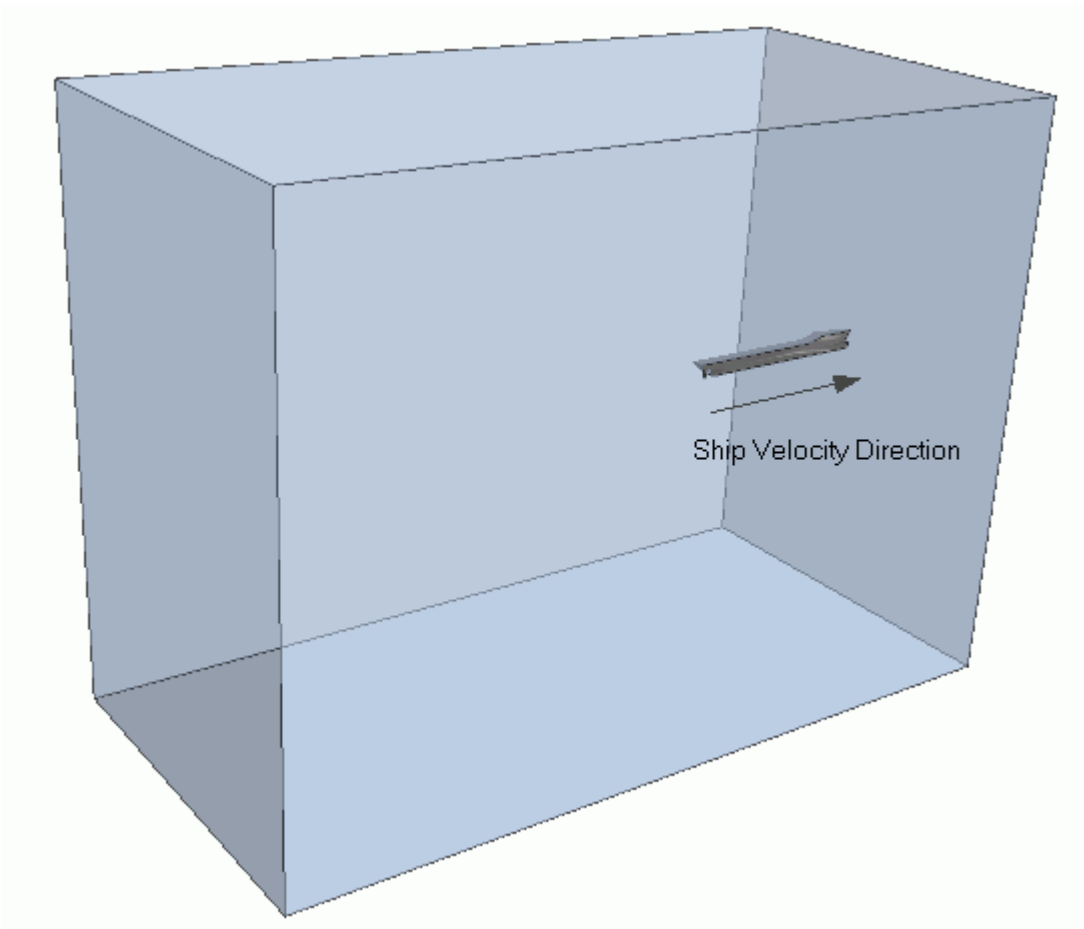
- Using the Overset Grids model.
- Activating the 6-DOF model.
- Creating a VOF wave and using it for initial and boundary conditions.

Marine Resistance Prediction: KCS Hull with a Rudder

This tutorial demonstrates the workflow for setting up a resistance prediction simulation for a marine application.

The geometry that is used is the [803]. This geometry was used as part of a series of simulations that were performed for the Gothenburg Workshop 2010 on Numerical Ship Hydrodynamics. The STAR-CCM+ simulation results were compared with experimental towing tank data.

The KCS ship hull with rudder is placed within a virtual towing tank as shown below:



The simulation is performed in model scale. The hull length is approximately 7.7 m (length between perpendiculars L_{pp}= 7.2786 m) with a scale factor of 31.6. The coordinate origin is at aft perpendicular and still water surface. The still water surface is at 0.341772 m above keel.

Only half of the geometry is modeled due to symmetry conditions. The computational domain extends from:

- -18 m to 18 m in the flow direction
- -18 m to 9 m in the vertical direction
- 0 m to 18 m in the lateral direction

The following boundary conditions are applied:

- Symmetry plane at the hull midplane
- Symmetry plane at the side of the tank
- Hydrostatic pressure corresponding to the undisturbed water surface at the downstream boundary of the tank
- Inlets with prescribed velocity and volume fraction at the upstream, top, and bottom boundaries.

To obtain resistance data experimentally, the ship hull is towed through the towing tank. For the equivalent simulation, the ship remains in a fixed position, and an inlet velocity is prescribed at the upstream inlet boundary. This inlet velocity produces the relative movement. The ship is free to heave and pitch. The simulation starts from a flat water surface with the bottom of the hull being parallel to the free surface. The computation is performed for a Froude number of Fn = 0.2599.

You can work through the tutorial in three major steps:

- Mesh Generation
- Physical and Numerical Setup
- Visualization and Data Analysis

Contents:

Mesh Generation
Physical and Numerical Setup
Visualization and Data Analysis
Summary
Bibliography

Mesh Generation

In order to achieve accurate resistance predictions, you are recommended to use trimmed hexahedral meshes with local refinements and prism layers along the walls of the wetted surface. In order to refine the computational mesh only where it is necessary, you create volumetric controls with particular shapes. A particular cell size is then assigned to these volumetric controls.

Contents:

Importing the Geometry
Creating the Towing Tank
Assigning Parts to Regions
Selecting the Meshers
Specifying the Mesh Settings
Setting up Volumetric Mesh Refinements
Specifying the Boundary Types
Generating the Volume Mesh

Importing the Geometry

The meshing process starts from a CAD geometry in Parasolid format.

To import the hull of the ship, including the rudder:

1. Create a directory for the tutorial called, `hullWithRudder`.
2. Start up STAR-CCM+ and create a Simulation by clicking **File** > **New Simulation...**
3. Select **File** > **Import** > **Import Surface Mesh...** Navigate to the `doc/startutorialsdata/motion/data` subdirectory of your STAR-CCM+ installation directory. Select the file `KCS_geometry.x_t` from the selection list and click **Open**.
4. In the *Import Surface Options* dialog, click **OK**.

   The imported geometry appears in the *Geometry Scene 1* window. The imported geometry resides under the **Geometry** > **Parts** node.
5. Open the **Parts** > **KCS_Geometry** > **Surfaces** node and inspect the imported surfaces.
6. Select **File** > **Save As** and save the simulation under the name `KCS_resistancePrediction.sim`.

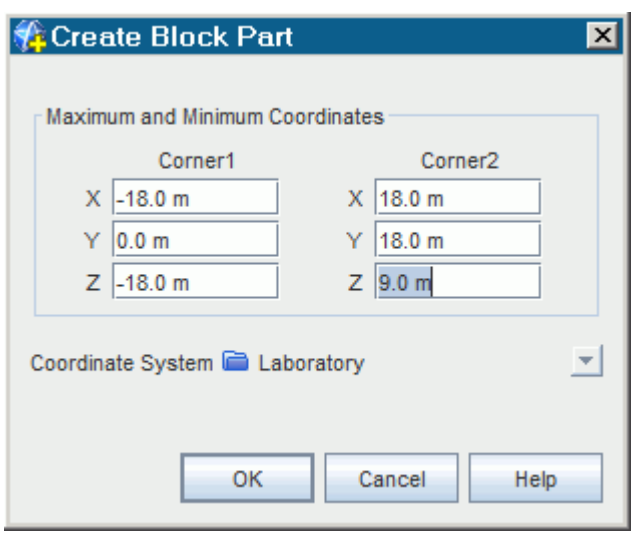
Creating the Towing Tank

The imported geometry contains only the ship hull with the rudder. To perform a resistance prediction analysis, create a towing tank around the ship.

For the towing tank, create a block shaped geometry around the ship, and then subtract the ship hull from the block. The resulting volume is a contiguous and closed virtual towing tank that contains the ship.

To create a block shaped part forming the towing tank:

1. Right-click the **Geometry** > **Parts** node and select **New Shape Part** > **Block**.
2. In the *Create Block Part* dialog, enter the following Maximum and Minimum Coordinates for the block:
   a) Set **Corner 1** to [`-18 m, 0.0 m, -18.0 m`].
   b) Set **Corner 2** to [`18.0 m, 18.0 m, 9.0 m`].



3. Click **Create** and then **Close**.
4. Rename the newly created **Geometry** > **Parts** > **Block** node to `Tank`.

The created block consists of one surface encompassing the entire tank.

In order to define different boundary conditions on each tank surface, split the tank surface:

5. Right-click the **Geometry** > **Parts** > **Tank** > **Surfaces** > **Block Surface** node and select **Split by Patch...**
6. In the *Split Part Surface by Patch* dialog, select **725** in the *Part Surface Patches* group box.

   The corresponding surface is highlighted in the *Graphics* window.
7. In the **Part Surface Name** text box, enter `Top`.
8. Click **Create**.
9. Split and rename the remaining block surfaces as described in the previous steps. Assign the following names to the different patches:

Patch	Part Surface Name
726	`Symmetry`
727	`Outlet`
728	`Side`
729	`Bottom`

Rename the final Patch 6 in the simulation tree:

10. Right-click the **Geometry** > **Parts** > **Tank** > **Surfaces** > **Block Surface** node and **Rename**... it `Inlet`.

To create the virtual towing tank, perform a Boolean subtract mesh operation:

11. Multi-select the **Tank** node and the **KCS_Geometry** node.
12. Right-click one of the highlighted items and select**Create Mesh Operation** > **Boolean** > **Subtract Parts.**
13. In the *Create Subtract Operation* dialog, click **OK**.

    The **Operations** > **Subtract** node is created.
14. Select the **Operations** > **Subtract** node and set *Target Part* to **Tank**.
15. Right-click the **Operations** > **Subtract** node and select **Execute**.
16. Rename the newly created **Parts** > **Subtract** part to `Virtual Towing Tank`.

To visualize the virtual towing tank:

17. Right-click in the empty space of the *Graphics* window and select **Hide All Parts**.
18. Select the **Geometry** > **Parts** > **Virtual Towing Tank** node and drag and drop it into the open geometry scene in the *Graphics* window.
19. In the pop-up menu, select **Part Displayers** > **Add to Geometry 1.**
20. Color the part surfaces.
    a) Select the **Scenes** > **Geometry Scene 1** > **Displayers** > **Geometry 1** node.
    b) In the *Properties* window, set *Color Mode* to **Distinguish Inputs**.

You can now inspect the combined geometry.

Assigning Parts to Regions

Before you create the volume mesh, transfer the virtual towing tank geometry to the regions level.

To assign the virtual towing tank part to the regions level:

1. Right-click the **Geometry** > **Parts** > **Virtual Towing Tank** node and select **Assign Parts to Regions...**
2. In the *Assign Parts to Regions* dialog:
   a) Select **Create a Region for Each Part**.
   b) Select **Create a Boundary for Each Part Surface**.
   c) Select **Do Not Create Interfaces From Contacts**.
   d) Click **Apply** and then **Close**.

To view the transferred geometry in the simulation tree:

3. Expand the **Regions** node.
   You can see the newly created **Virtual Towing Tank** node.
4. Expand the **Virtual Towing Tank** > **Boundaries** node.
   You can see all of the transferred boundary surfaces.

Selecting the Meshers

You select the required meshers for the surface mesh, the volume mesh, and the prism layers through the creation of an Automated Mesh operation.

To select the required mesh models:

1. Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
2. In the *Create Automated Mesh Operation* dialog:
   a) Select **Virtual Towing Tank** from the *Parts* list.
   b) Select the following meshers, in order:

Group	Mesher
Surface Meshers	**Surface Remesher**
Optional Surface Meshers	**Automatic Surface Repair**
Core Volume Meshers	**Trimmed Cell Mesher**
Optional Boundary Layer Meshers	**Prism Layer Mesher**

Specifying the Mesh Settings

Resistance analyses are generally performed using a trimmed volume mesh with prism layers on the wetted surface of the hull. Using trimmed cells means that the mesh is aligned with the undisturbed free surface.

To adjust the Prism Layer mesher settings:

1. Edit the **Operations** > **Automated Mesh** > **Meshers** node and set the following properties:

Node	Property	Setting
Prism Layer Mesher	*Minimum Thickness Percentage*	`1.0`
	Boundary March Angle	`75.0`

To specify global mesh settings:

2. Edit the **Automated Mesh** > **Default Controls** node and set the following properties:

Node	Property	Setting
Base Size	*Value*	`0.1125 m`
Target Surface Size	*Percentage of Base*	`50`
Minimum Surface Size	*Percentage of Base*	`6.25`
Surface Curvature > **Basic Curvature**	*# Pts./circle*	`36`
Auto-Repair Minimum Proximity >	*Minimum Proximity*	`0.001`
Number of Prism Layers	*Number of Prism Layers*	`6`
Prism Layer Total Thickness	*Absolute*	`0.02 m`
Volume Growth Rate	*Default Growth Rate*	**Slow**
	Surface Growth Rate	**Slow**
Maximum Cell Size	*Percentage of Base*	`1600`

In addition to the globally applied mesh settings, you specify local mesh settings for the boundary surfaces.

To specify local mesh settings on boundaries:

3. Right-click the **Automated Mesh** > **Custom Controls** node and create a new **Surface Control**.
4. Rename the **Surface Control** node to `Deck`.
5. Edit the **Deck** node and set the following properties:

Node	Property	Setting
Deck	*Part Surfaces*	**[Virtual Towing Tank.KCS_Geometry.Deck]**
└ **Controls**		
└ **Target Surface Size**	*Target Surface Size*	**Specify Custom Value**
└ **Minimum Surface Size**	*Minimum Surface Size*	**Specify Custom Value**
└ **Prism Layers**	*Prism Layers*	**Disable**
└ **Values**		
└ **Target Surface Size**	*Percentage of Base*	`200`

Node	Property	Setting
└ **Minimum Surface Size**	*Percentage of Base*	`100`

6. Copy and paste the **Deck** node into the **Custom Controls** node and rename it to `Hull`.
7. Edit the **Hull** node and set the following properties:

Node	Property	Setting
Hull	*Part Surfaces*	**[Virtual Towing Tank.KCS_Geometry.Hull]**
└ **Controls**		
└ **Prism Layers**	*Prism Layers*	**Use Parent Values**
└ **Values**		
└ **Target Surface Size**	*Percentage of Base*	`25`
└ **Minimum Surface Size**	*Percentage of Base*	`25`

8. Copy and paste the **Hull** node into the **Custom Controls** node and rename it to `Rudder`.
9. Edit the **Rudder** node and set the following properties:

Node	Property	Setting
Rudder	*Part Surfaces*	**[Virtual Towing Tank.KCS_Geometry.Rudder]**
└ **Values**		
└ **Target Surface Size**	*Percentage of Base*	`12.5`
└ **Minimum Surface Size**	*Percentage of Base*	`6.25`

10. Create a new **Surface Control** and rename it to `RudderEdgeShaftStern`.
11. Edit the **RudderEdgeShaftStern** node and set the following properties:

Node	Property	Setting
RudderEdgeShaftStern	*Part Surfaces*	**[Virtual Towing Tank.KCS_Geometry.RudderEdge,Virtual Towing Tank.KCS_Geometry.Shaft,Virtual Towing Tank.KCS_Geometry.Stern]**
└ **Controls**		
└ **Prism Layers**	*Prism Layers*	**Disable**

12. Create a new **Surface Control** and rename it to `Tank`.
13. Edit the **Tank** node and set the following properties:

Node	Property	Setting
Tank	*Part Surfaces*	**[Virtual Towing Tank.Tank.Bottom,Virtual Towing Tank.Tank.Inlet,Virtual Towing Tank.Tank.Outlet,Virtual Towing Tank.Tank.Side,Virtual Towing Tank.Tank.Symmetry,Virtual Towing Tank.Tank.Top]**
└ **Controls**		
└ **Target Surface Size**	*Target Surface Size*	**Specify Custom Value**
└ **Minimum Surface Size**	*Minimum Surface Size*	**Specify Custom Value**
└ **Values**		
└ **Target Surface Size**	*Percentage of Base*	`1600`
└ **Minimum Surface Size**	*Percentage of Base*	`1600`

Setting up Volumetric Mesh Refinements

In order to refine sharp corners or edges of your geometry, for example at the bow, or to improve the resolution of flow features, for example flow separation or the Kelvin wake, you can apply volumetric refinements to your volume mesh. For this application, it is particularly important to refine the free water surface area, that is, the interface area between water and air.

By using trimmed cells, you can take advantage of anisotropic refinement in the normal direction of the free surface area. You define the geometrical shapes and dimensions of these volumetric controls in the form of part shapes.

To generate volumetric refinements:

1. Generate a part shape at the bow:
   a) Right-click the **Geometry** > **Parts** node and select **New Shape Part** > **Block**.
   b) In the *Create Block Part* dialog, enter the following Maximum and Minimum Coordinates for the block:
      - Set **Corner 1** to [`7.17 m, -0.087 m, -0.34 m`].
      - Set **Corner 2** to [`7.54 m, 0.087 m, 0.059 m`].
   c) Click **Create** and **Close**.
   d) Rename the **Block** part as `BOW`.
2. Generate a part shape at the stern:
   a) Right-click the **Geometry** > **Parts** node and select **New Shape Part** > **Block**.

b) In the *Create/Edit Block Part* dialog, enter the following Maximum and Minimum Coordinates for the block:
   - Set **Corner 1** to [`-0.12 m, -0.09 m, -0.34 m`].
   - Set **Corner 2** to [`0.28 m, 0.05 m, 0.008 m`].

c) Click **Create** and **Close**.

d) Rename the **Block** part as `STERN RUDDER - fine`.

3. Right-click the **Operations** > **Automated Mesh** > **Custom Controls** node and select **New** > **Volumetric Control**.
4. Select the **Volumetric Control** node and rename it `BOW, STERN - fine`.
5. Edit the **BOW, STERN - fine** node and set the following properties:

Node	Property	Setting
BOW, STERN - fine	*Parts*	**[BOW, STERN RUDDER - fine]**
└ **Controls** > **Surface Remesher**	*Customize Size*	Activated
└ **Values** > **Custom Size**	*Size Type*	**Relative to base**
	Percentage of Base	`12.5`

In order to resolve flow features adequately around the hull, this tutorial specifies several part shapes.

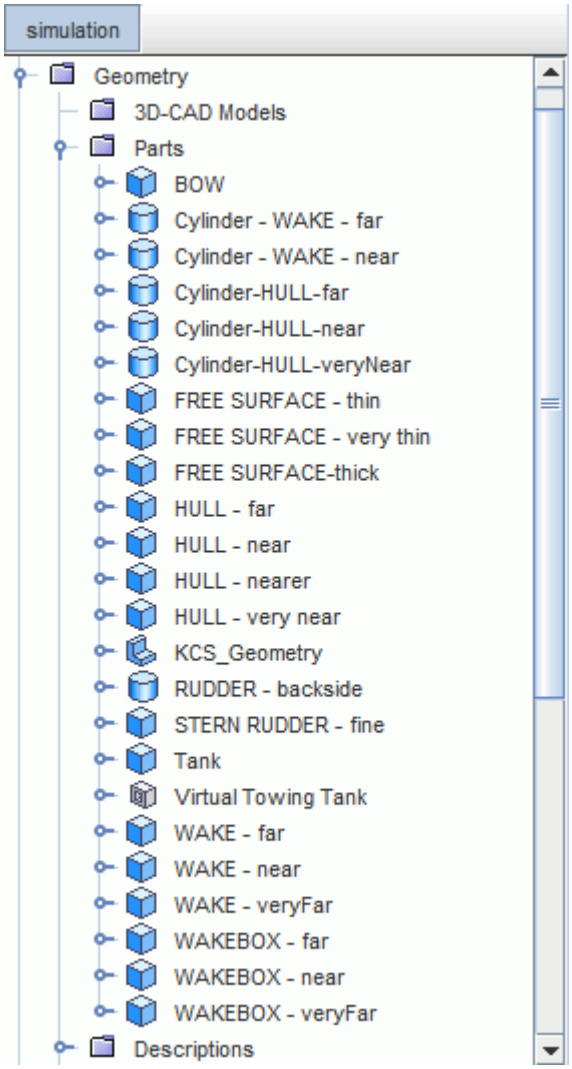
To run the macro for creating part shapes and volumetric controls:

6. Select **File** > **Macro** > **Play Macro...**
7. In the *Open* dialog, navigate to the `doc/startutorialsdata/motion/data` subdirectory of your STAR-CCM+ installation directory. Select the file `partShapes.java` from the selection list and click **Open**.

   The macro is executed and creates the required part shapes and volumetric controls.

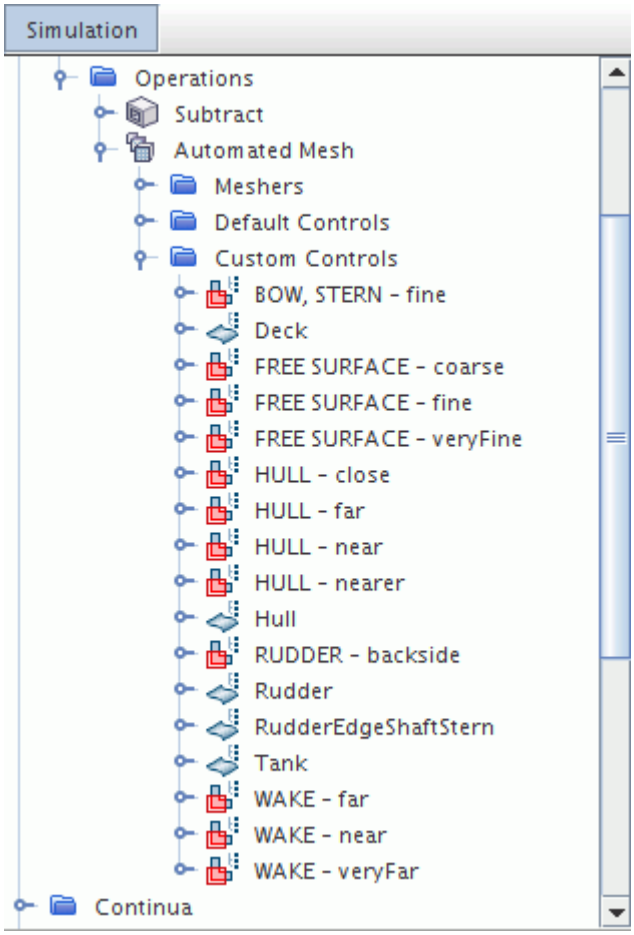
To view the created part shapes:

8. Expand the **Geometry** > **Parts** node.



To view the created volumetric controls:

9. Expand the **Geometry** > **Operations** > **Automated Mesh** > **Custom Controls** node.



Specifying the Boundary Types

Before you create the volume mesh, set the appropriate boundary types. By default, no prism layers are created for inflow, outflow, and symmetry boundaries.

To set the boundary types:

1. Select the **Regions** > **Virtual Towing Tank** > **Boundaries** > **Tank.Bottom** node and set *Type* to **Velocity Inlet**.
2. Similarly, set the boundary types for the other boundaries as listed in the following table:

Boundary Name	Boundary Type
Tank.Inlet	Velocity Inlet
Tank.Outlet	Pressure Outlet
Tank.Side	Symmetry Plane
Tank.Symmetry	Symmetry Plane
Tank.Top	Velocity Inlet

All boundaries with the **KCS_Geometry** prefix remain as **Wall**.

Generating the Volume Mesh

You can now proceed to generate the volume mesh.

To generate the volume mesh:

1. Select **Mesh** > **Generate Volume Mesh**.

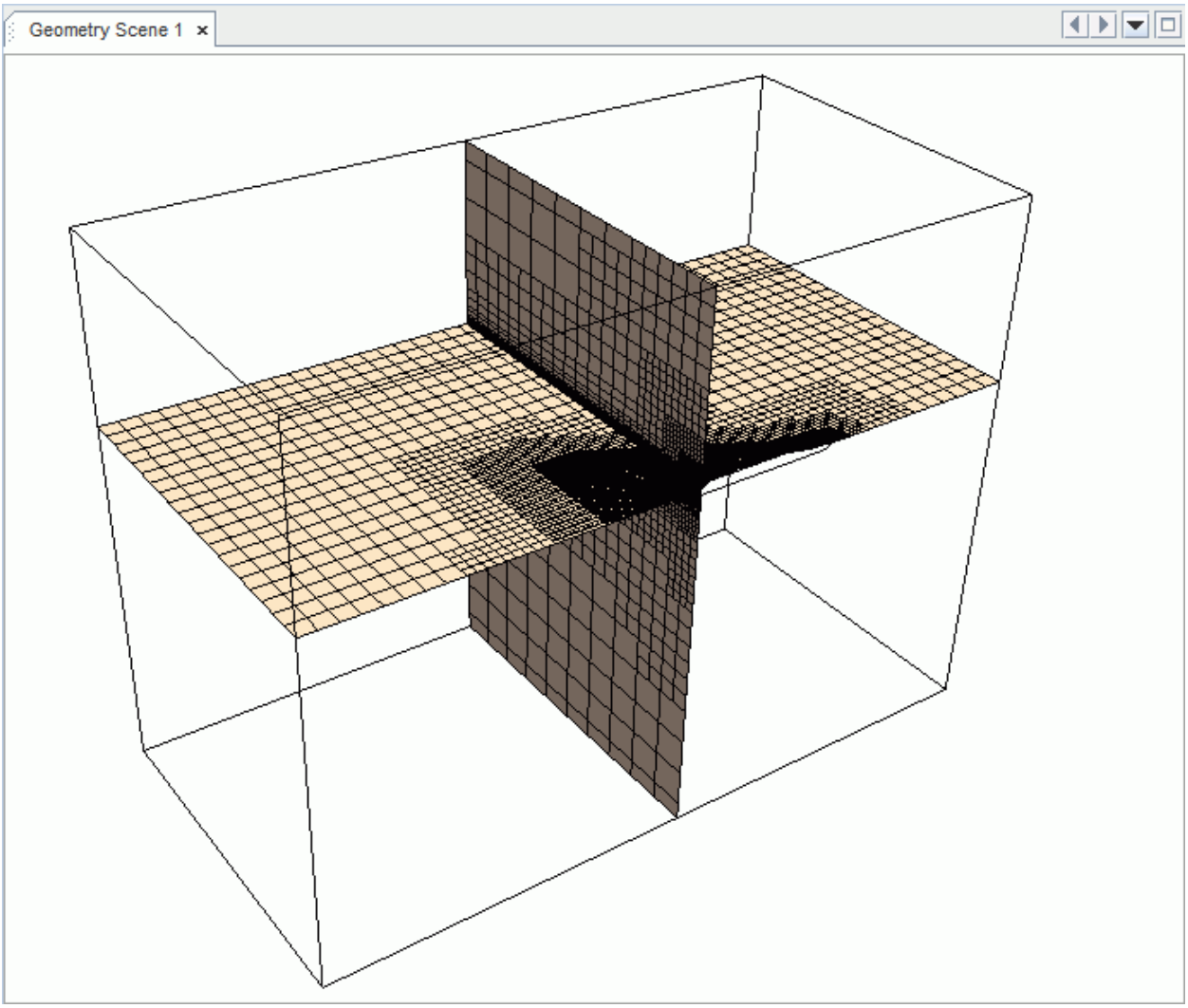
   The meshing process starts. You can follow the meshing progress in the *Output* window.

To visualize the final volume mesh:

2. Create two planar sections:
   a) Right-click **Derived Parts** and select **New Part** > **Section** > **Plane...**
   b) In the *Create Section* dialog:
      - In the Input Parts group box, click **Select** and choose **Regions** > **Virtual Towing Tank** in the *Select Objects* dialog.
      - Click **OK**.
   c) In the *Plane Parameters* group box:
      - Set **origin** to [`0.0 m, 0.0 m, -0.118 m`].
      - Set **normal** to [`0.0 m, 0.0 m, 1.0 m`].
      - Click **Create**.
   d) For the second planar section, in the *Plane Parameters* group box:
      - Set **origin** to [`0.5 m, 0.0 m, 0.0 m`].
      - Set **normal** to [`1.0 m, 0.0 m, 0.0 m`].
      - Click **Create** and then **Close**.

   Two new **Derived Parts** nodes were created: **Plane Section** and **Plane Section 2**.
3. Click *Scene/Plot*.
4. Select the **Displayers** > **Geometry 1** > **Parts** node and, in the *Properties* window, click ... (**Custom Editor**) next to *Parts*.
5. In the *Parts - Parts* dialog, deselect all and click **OK**.
6. Select the **Displayers** > **Outline 1** > **Parts** node and, in the *Properties* window, click ... (**Custom Editor**) next to *Parts.*
7. In the *Parts - Parts* dialog, select **Regions** > **Virtual Towing Tank** and click **OK**.
8. Multi-select the **Section Geometry 1** and **Section Geometry 2** nodes and set *Mesh* to **Activated**.
9. Right-click the empty space in the *Graphics* window and select **Apply Representation** > **Volume Mesh**.

   The inner trimmed volume mesh is displayed on the two planar sections.



Note the embedded refinement at an angle with respect to the track of the ship. This refinement is created to capture the Kelvin wake that is produced by the ship.

Physical and Numerical Setup

Flow resistance simulation is a complex transient multi-physics problem.

In STAR-CCM+, setting this simulation up involves:

- Initial conditions
- Boundary conditions
- VOF multiphase
- VOF Waves and wave damping
- Dynamic Fluid Body Interaction
- Transient numerical parameters

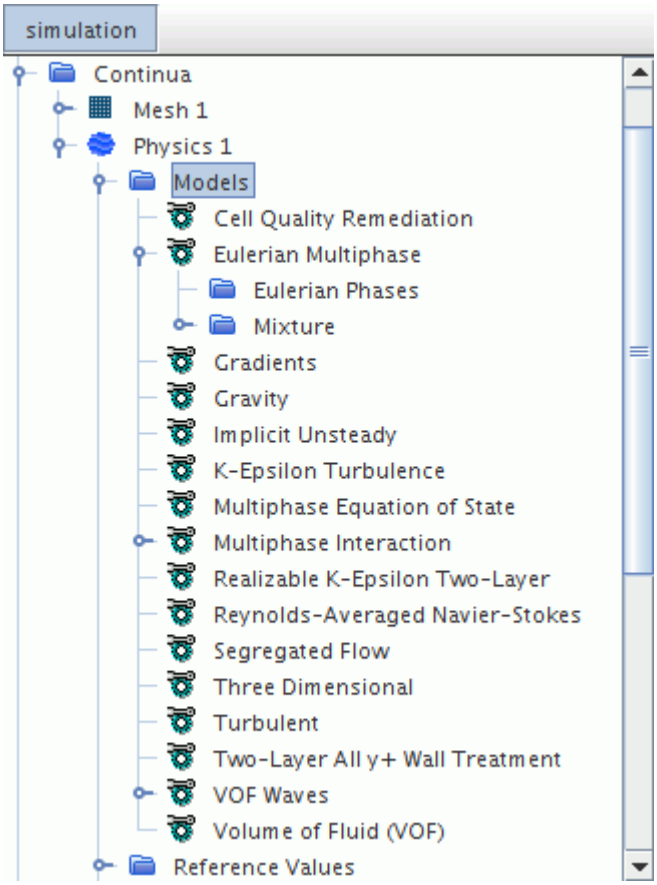
Contents:

Selecting the Physical Models
Defining the Eulerian Phases
Defining the VOF Wave
Setting Initial Conditions
Damping Wave Reflections
Setting Boundary Conditions
Defining Dynamic Fluid Body Interaction (DFBI)
Setting Solver Parameters and Stopping Criteria

Selecting the Physical Models

Select the physical models that are required for this simulation.

1. Return to *simulation*.
2. Right-click the **Continua** > **Physics 1** > **Models** node and choose **Select Models**...
3. In the *Physics 1 Model Selection* dialog:
   a) Select **Implicit Unsteady** from the *Time* group box.
   b) Select **Eulerian Multiphase** from the *Material* group box.
   c) Select **Volume of Fluid (VOF)** from the *Multiphase Model* group box.
   d) Select **Turbulent** from the *Viscous Regime* group box.
   e) Select **K-Epsilon Turbulence** from the *Reynolds-Averaged Turbulence* group box.
   f) From the *Optional Models* group box, select **Gravity, Cell Quality Remediation,** and **VOF Waves**.



g) Click **Close**.

Defining the Eulerian Phases

Using the VOF multiphase model requires that you create Eulerian phases for water and air.

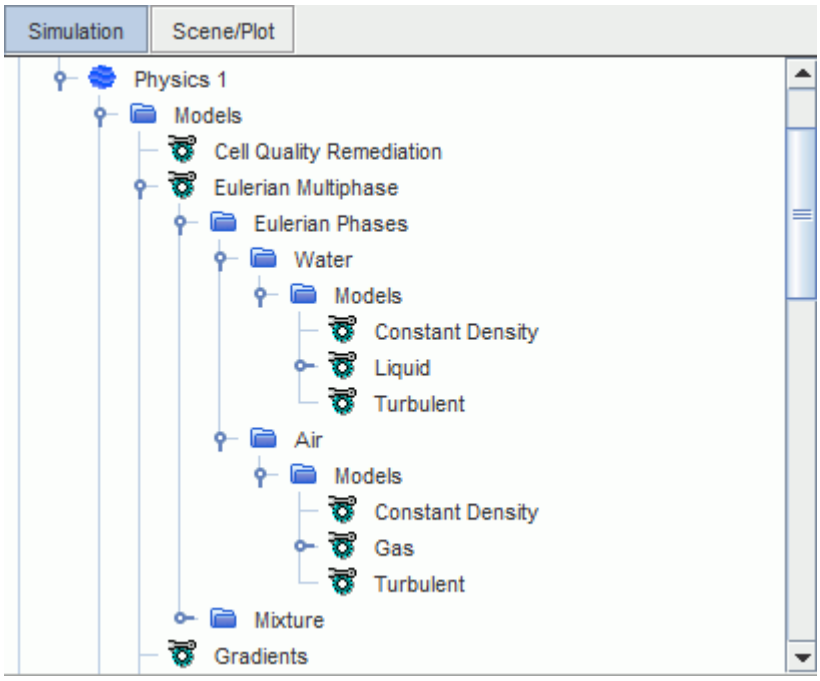
To define the Eulerian phases:

1. Right-click the **Continua** > **Physics 1** > **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and select **New**.
2. Rename **Phase 1** as `Water`.
3. Right-click the **Eulerian Phases** > **Water** > **Models** node and choose **Select Models...**
4. In the *Water Model Selection* dialog:
   a) Select **Liquid** from the *Material* group box.
   b) Select **Constant Density** from the *Equation of State* group box.
   c) Click **Close**.
5. Right-click the **Continua** > **Physics 1** > **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and select **New**.
6. Rename **Phase 1** as `Air`.
7. Right-click the **Eulerian Phases** > **Air** > **Models** node and choose **Select Models...**
8. In the *Air Model Selection* dialog:

a) Select **Gas** from the *Material* group box.

b) Select **Constant Density** from the *Equation of State* group box.

9. Click **Close**.



Defining the VOF Wave

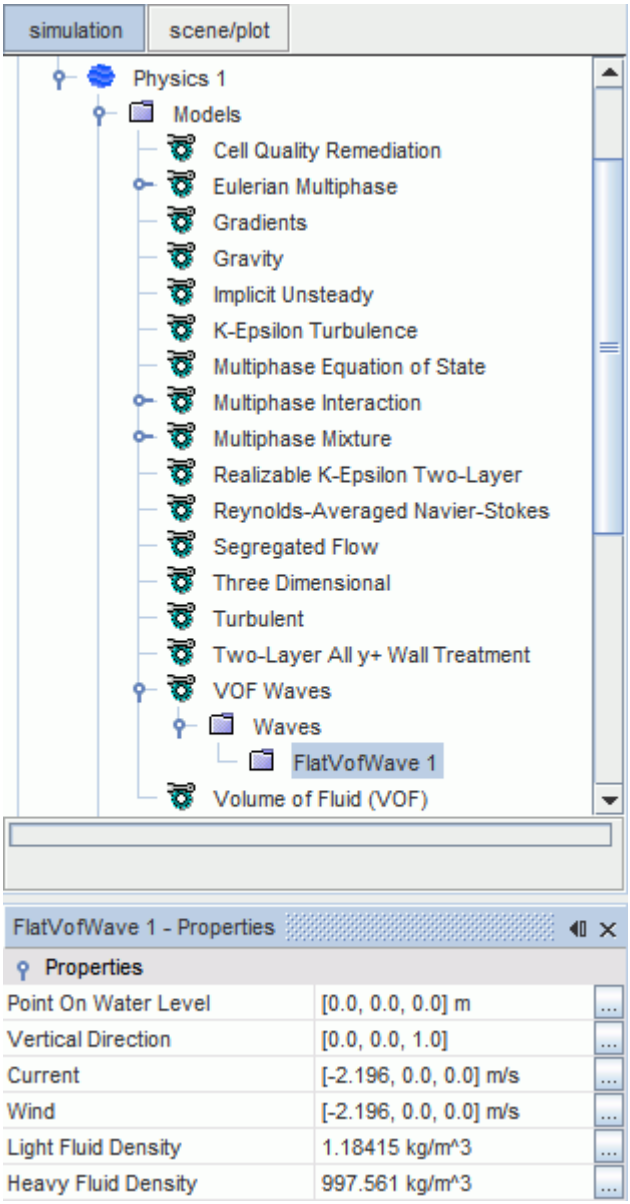
The free surface water level changes over time during the simulation. STAR-CCM+ provides the VOF Waves model that lets you specify wave initial and boundary conditions. Here, the ship is towed through calm water. Therefore, define a flat VOF Wave.

To define a flat VOF Wave:

1. Right-click the **Continua** > **Physics 1** > **Models** > **VOF Waves** > **Waves** node and select **New** > **Flat.**
2. Select the **Flat Vof Wave 1** node.
3. In the *Flat Vof Wave 1 - Properties* window:

   a) Set *Current* to [`-2.196, 0.0, 0.0`] m/s.

   b) Set *Wind* to [`-2.196, 0.0, 0.0`] m/s.



Setting Initial Conditions

Set the initial conditions for pressure, velocity, and volume fraction.

To set the initial conditions:

1. Edit the **Continua** > **Physics 1** > **Initial Conditions** node and set the following properties:

Node	Property	Setting
Pressure	*Method*	**Field Function**
	Scalar Function	**Hydrostatic Pressure of Flat Vof Wave 1**
Velocity	*Method*	**Field Function**

Node	Property	Setting
	Vector Function	**Velocity of Flat Vof Wave 1**
Volume Fraction	*Method*	**Composite**
└ **Composite** > **Water**	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Heavy Fluid of Flat Vof Wave 1**
└ **Composite** > **Air**	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Light Fluid of Flat Vof Wave 1**

Damping Wave Reflections

Unless appropriate measures are applied, wave reflections can occur during a flow resistance simulation. These wave reflections must be avoided, because they interact with the true wave field and can lead to invalid results.

There are two sources for wave reflections:

- wave reflections from boundaries
- wave reflections due to abrupt mesh transitions

The second cause was addressed by changing the *Template Default Growth Rate* from **Fast** (default) to **Slow** during mesh generation (see Generating the Volume Mesh).

To remedy wave reflections from boundaries, STAR-CCM+ provides the VOF Wave Damping functionality.

To activate VOF Wave damping at boundaries:

1. Select the **Regions** > **Virtual Towing Tank** > **Physics Conditions** > **VOF Wave Zone Option** node and activate **Damping**.
2. Select the **Regions** > **Virtual Towing Tank** > **Physics Values** > **VOF Wave Damping Length** node and set *Value* to `9.0 m`.

Setting Boundary Conditions

Leave the boundary conditions for the boundaries labelled with the prefix **KCS_Geometry** at their default wall boundary condition settings.

To set the boundary conditions:

1. Edit the **Regions** > **Virtual Towing Tank** > **Boundaries** > **Tank.Inlet** and the node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Velocity Specification**	*Method*	**Components**

Node	Property	Setting
└ **VOF Wave Damping Boundary Option**	*Method*	**Yes**
Physics Values		
└ **Velocity**	*Method*	**Field Function**
	Vector Function	**Velocity of Flat Vof Wave 1**
└ **Volume Fraction**	*Method*	**Composite**
└ **Composite** > **Water**	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Heavy Fluid of Flat Vof Wave 1**
└ **Composite** > **Air**	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Light Fluid of Flat Vof Wave 1**

2. Multi-edit the **Tank.Bottom** and the **Tank.Top** nodes and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Velocity Specification**	*Method*	**Components**
└ **VOF Wave Damping Boundary Option**	*Method*	**No**
Physics Values		
└ **Velocity**	*Method*	**Field Function**
	Vector Function	**Velocity of Flat Vof Wave 1**
└ **Volume Fraction**	*Method*	**Composite**
└ **Composite** > **Water**	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Heavy Fluid of Flat Vof Wave 1**
└ **Composite** > **Air**	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Light Fluid of Flat Vof Wave 1**

3. Edit the **Tank.Outlet** node and set the following properties:

Node	Property	Setting
Physics Conditions		

Node	Property	Setting
└ **VOF Wave Damping Boundary Option**	*Method*	**Yes**
Physics Values		
└ **Pressure**	*Method*	**Field Function**
	Scalar Function	**Hydrostatic Pressure of Flat Vof Wave 1**
└ **Volume Fraction**	*Method*	**Composite**
└ **Composite** > **Water**	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Heavy Fluid of Flat Vof Wave 1**
└ **Composite** > **Air**	*Method*	**Field Function**
	Scalar Function	**Volume Fraction of Light Fluid of Flat Vof Wave 1**

4. Select the **Tank.Side** > **Physics Conditions** > **VOF Wave Damping Boundary Option** node and set *Method* to **Yes**.
5. Save the simulation.

Defining Dynamic Fluid Body Interaction (DFBI)

The Dynamic Fluid Body Interaction (DFBI) model simulates the motion of the ship according to the forces acting on it induced by the flow. For this simulation, you allow the ship is to move with 2 degrees of freedom to account for sinkage and trim. You employ the equilibrium motion option to achieve a quasi steady-state equilibrium position of the ship. This option reduces the physical runtime of the simulation.

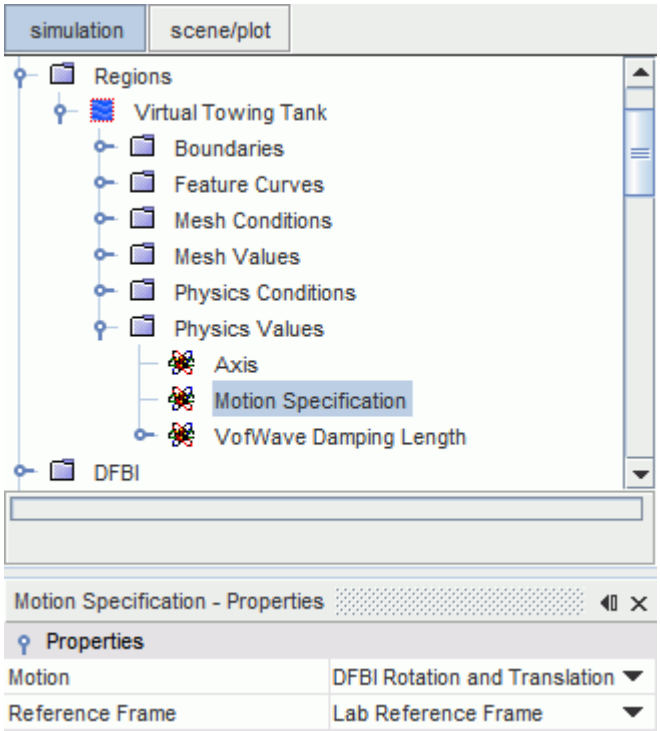
To create the DFBI motion:

1. Right-click the **Tools** > **Motions** node and select **New** > **DFBI Rotation and Translation**.

Assign this motion to the Virtual Towing Tank region:

2. Select the **Regions** > **Virtual Towing Tank** > **Physics Values** > **Motion Specification** node and set *Motion* to **DFBI Rotation and Translation**.

   A new **DFBI** node appears in the simulation tree.



To define the 6-DOF body properties of the ship:

3. Right-click the **DFBI** > **6-DOF Bodies** node and select **New** > **3D Body**.
4. Rename Body 1 as `Ship`.
5. Select the **6-DOF Bodies** > **Ship** node and set the following properties:

Property	Setting
Body Surface	All boundaries with the prefix **KCS_Geometry**
Body Mass	`823.0451 kg`
Release Time	`1.0 s`
Ramp Time	`5.0 s`
Body Motion Option	**Equilibrium**

To define the initial conditions of the 6-DOF body:

6. Expand the **6-DOF Bodies** > **Ship** > **Initial Values node.**
   a) Select the **Center of Mass** node and set *Position* to `[3.53151899, 0.0, -0.1113924] m`.
   b) Select the **Moment of Inertia** node and set *Diagonal Components* to `[136.7, 2725.117, 2725.117] kg/m²`.
   c) Activate **Use Center of Mass**.
7. Create a new local initial coordinate system at the center of mass of the ship:
   a) Right-click **Tools** > **Coordinate Systems** > **Laboratory** > **Local Coordinate Systems** and select **New** > **Cartesian**.

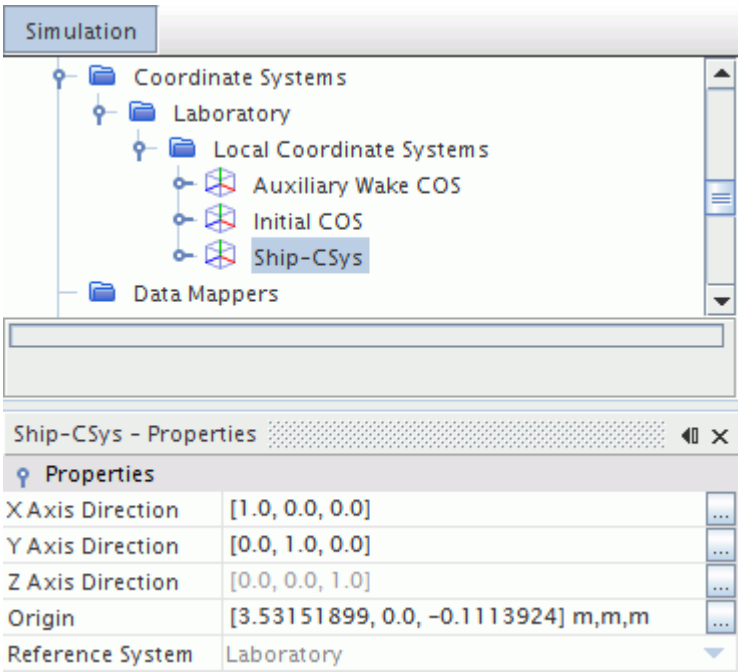
b) In the *Create Cartesian Coordinate System* dialog, set **Origin** to [`3.53151899 m, 0.0 m, -0.1113924 m`].

c) Click **Create** and then **Close**.

d) Rename **Cartesian 1** as `Initial COS`.

8. Select the **6-DOF Bodies** > **Ship** > **Initial Values node** > **Orientation** node and set *Coordinate System* to **Laboratory** > **Initial COS**.

The creation of the new 6-DOF body automatically creates a body local coordinate system, here called, **Ship-Csys**. This body local coordinate system is at the center of mass of the body and moves with the body as the simulation progresses.



To define the equilibrium motion of the ship:

9. Select the **DFBI** > **6-DOF Bodies** > **Ship** > **Equilibrium** node and set the following properties:

Property	Setting
Z Motion	Activated
Y Rotation	Activated

These settings allow motion in the Z direction (sinkage) and rotation around the Y axis (trim).

Setting Solver Parameters and Stopping Criteria

A resistance prediction analysis is a transient simulation for which you must specify the time-step, the number of inner iterations per time-step, and the physical time. Additionally, you can set numerical parameters such as under-relaxation factors to enhance convergence per time-step.

To set the solver parameters:

1. Select the **Solvers** > **Implicit Unsteady** node and set *Time Step* to `0.04 s`.
2. Select the **Solvers** > **6-DOF Solver** node and set *Maximum Number Of Iterations* to `5`.
3. Select the **Solvers** > **Segregated Flow** > **Velocity** node and set *Under-Relaxation Factor* to `0.7`.
4. Select the **Solvers** > **Segregated Flow** > **Pressure** node and set *Under-Relaxation Factor* to `0.4`.
5. Select the **Solvers** > **Segregated VOF** > **Single-Step** node and set *Under-Relaxation Factor* to `0.8`.

To set the stopping criteria:

6. Select the **Stopping Criteria** > **Maximum Inner Iterations** node and set *Maximum Inner Iteration* to `10`.
7. Select the **Stopping Criteria** > **Maximum Physical Time** node and set *Maximum Physical Time* to `200.0 s`.
8. Select the **Stopping Criteria** > **Maximum Steps** node and deactivate *Enabled*.
9. Save the simulation.

Visualization and Data Analysis

Here, you create several plots and graphs for visualizing the results.

The following items are analyzed:

- Visualizing the Propagation of the Free Water Surface
- Visualizing the Wave Pattern
- Monitoring and Plotting Resistance Data
- Monitoring and Plotting Trim and Sinkage Data

Contents:

Visualizing the Propagation of the Free Water Surface
Visualizing the Wave Pattern
Monitoring and Plotting Resistance Data
Monitoring and Plotting Trim and Sinkage Data
Monitoring and Plotting Equilibrium Forces and Moments
Running the Simulation
Visualizing the Results
Comparing with Experimental Results

Visualizing the Propagation of the Free Water Surface

You can plot the free water surface to get a good visual representation of the effect of hull resistance on the surrounding flow.

To visualize the propagation of the free water surface:

1. Create a contour plot of the volume fraction on the symmetry plane of the domain:
   a) Right-click the **Scenes** node and select **New Scene** > **Scalar**.

b) Rename **Scalar Scene 1** as `FreeSurfaceHull`.

c) Click **Scene/Plot**.

d) Select the **Displayers** > **Scalar 1** node and set *Contour Style* to **Smooth Filled**.

e) Select the **Scalar 1** > **Parts** node and set *Parts* to **Regions** > **Virtual Towing Tank** > **Tank.Symmetry**.

f) Select the **Scalar 1** > **Scalar Field** node and set *Function* to **Volume Fraction** > **Volume Fraction of Air**.

2. Click anywhere in the displayed scene and then press `s` on your keyboard to obtain a side view of the ship and symmetry plane.

Visualizing the Wave Pattern

You can plot the elevation of the free water surface to get a good visual representation of the wake that the hull produces as it moves through the water.

To create a contour plot of the elevation of the water surface:

1. Right-click the **Scenes** node and select **New Scene** > **Scalar**.
2. Rename **Scalar Scene 1** as `Waves`.
3. Create an iso-surface for the volume fraction of water at a value of 0.5:

   A value of 0.5 for the water volume fraction means that a computational cell is filled with 50% of water and, consequently, 50% of air. Therefore, this value gives the position of the water-air-interface corresponding to the free surface.

   a) Right-click the **Derived Parts** node and select **New Part** > **Isosurface...**

      The *Create Isosurface* dialog appears.

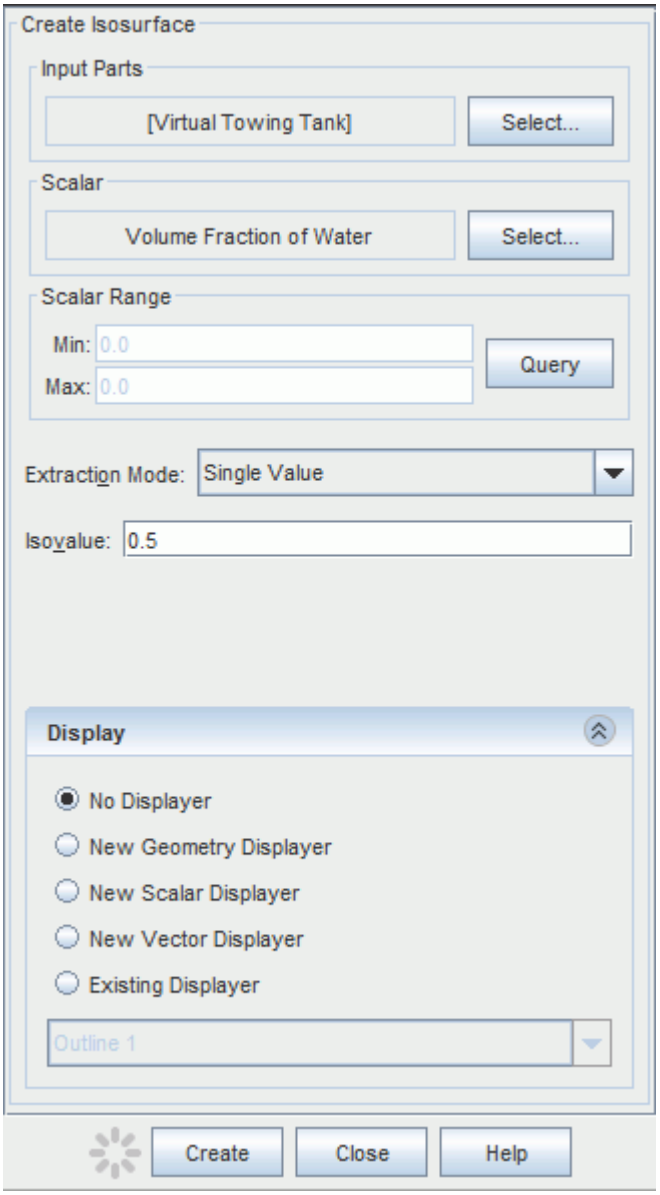
   b) In the *Input Parts* group box, select **Regions** > **Virtual Towing Tank**.

   c) In the *Scalar* group box, select **Volume Fraction** > **Volume Fraction of Water**.

   d) Set *Isovalue* to `0.5`.

   e) Select **No Displayer**.

   f) Click **Create** and **Close**.

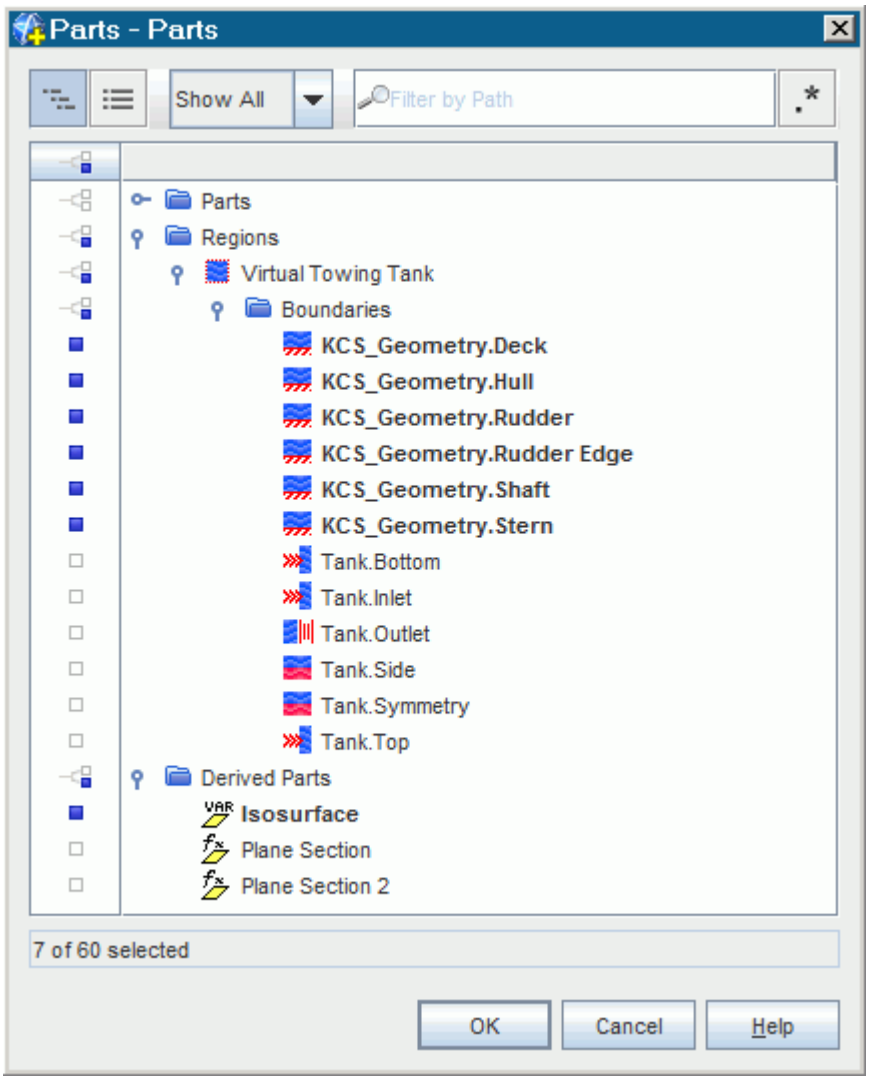


4. Click **Scene/Plot**.
5. Select the **Scalar 1** node.
6. In the *Scalar 1 - Properties* window:
   a) Set *Contour Style* to **Smooth Filled + Lines**.
   b) Set *Transform* to **Tank.Symmetry 1**.
7. Select the **Scalar 1** > **Parts** node and set *Parts* to **Derived Parts** > **Isosurface**. Click **OK**.
8. Select the **Scalar 1** > **Scalar Field** node and set *Function* to **Position** > **Laboratory** > **Z.**
9. Select the **Displayers** > **Outline 1** node and de-activate *Outline*.
10. Select the **Outline 1** > **Parts** node.
11. In the *Parts - Properties* window:
    a) Click ... (**Custom Editor**) next to *Parts.*

b) Expand **Regions** > **Virtual Towing Tank** > **Boundaries** and select all boundaries with the prefix **KCS**.

c) Expand **Derived Parts** and select **Isosurface**.

d) Click **OK**.



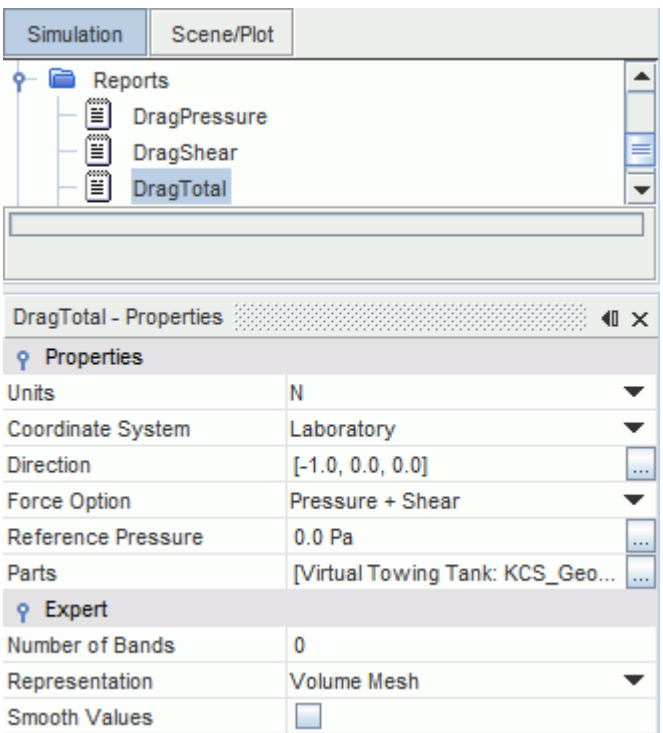
12. Click anywhere in the scene display area and then press `t` on your keyboard to obtain a top view of the ship and the wave pattern.
13. Return to *Simulation*.

Monitoring and Plotting Resistance Data

The resistance of a ship hull is defined as the force required to tow the ship in calm water at a constant velocity. Generally, a ship is designed to move through the water with as little force as possible. You can create a graph of the hull resistance over time to extract the value of this design parameter of your ship from the simulation. The drag force can be split into the two components: pressure drag and shear drag.

To analyze the drag of the ship hull:

1. Create a report for the shear drag over time:
   a) Right-click the **Reports** node and select **New Report** > **Force**.
   b) Rename **Force 1** as `DragShear`.
   c) In the *DragShear - Properties* window:
      - Set *Force Option* to **Shear**.
      - Set *Parts* to **Regions** > **Virtual Towing Tank** > **Boundaries** [all boundaries with the prefix **KCS_Geometry**] and click **OK**.
      - Set *Direction* to `[-1.0, 0.0, 0.0]`.
2. Create a report for the pressure drag over time:
   a) Copy and paste the **DragShear** node.
   b) Rename **Copy of DragShear** as `DragPressure`.
   c) In the *DragPressure - Properties* window, set *Force Option* to **Pressure**.
3. Create a report for the total drag over time:
   a) Copy and paste the **DragShear** node.
   b) Rename **Copy of DragShear** as `DragTotal`.
   c) In the *DragTotal - Properties* window, set *Force Option* to **Pressure + Shear**.



4. Create a monitor and plot for a combined display of shear and pressure drag:
   a) Multi-select **DragPressure** and **DragShear**.
   b) Right-click on any of the highlighted nodes and select **Create Monitor and Plot from Report**.
   c) In the pop-up window, click **Single Plot**.
   d) Rename **Plots** > **Reports Plot** as `ShearAndPressureDrag Plot`.

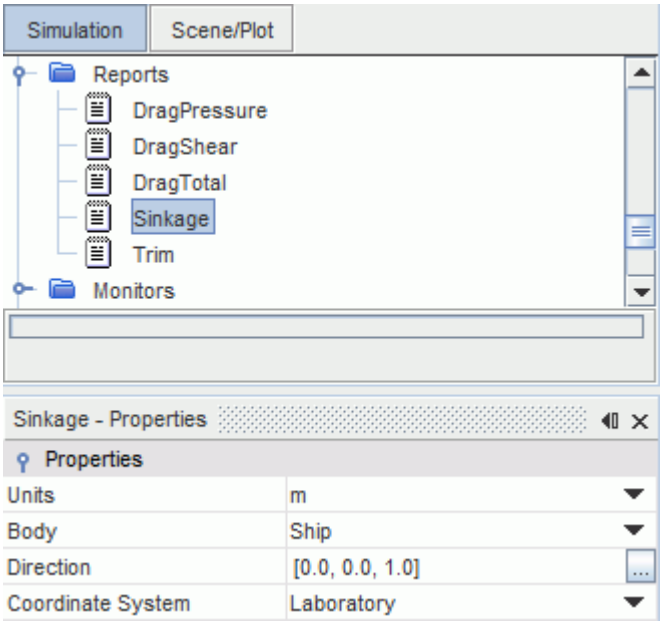
   e) Right-click the **Plots** > **ShearAndPressureDrag Plot** node and select **Open**.
5. Adjust the scale of the plot:
   a) Select the **Plots** > **ShearAndPressureDrag Plot** > **Axes** > **Left Axis** node and set *Maximum* to `400.0`.
6. Create a monitor and plot for the total drag:
   a) Right-click the **Reports** > **DragTotal** node and select **Create Monitor and Plot from Report**.
   b) Rename **Plots** > **DragTotal Monitor Plot** as `DragTotal Plot`.
   c) Right-click the **Plots** > **DragTotal Plot** node and select **Open.**
7. Adjust the scale of the plot:
   a) Select the **Plots** > **DragTotal Plot** > **Axes** > **Left Axis** node and set *Maximum* to `400.0`.

Monitoring and Plotting Trim and Sinkage Data

A ship advancing through water experiences sinkage and trim due to the hydrodynamic forces acting on the ship hull. A towing tank ship model free to sink and trim can experience an increase in resistance. You can create a graph of the trim and sinkage over time to extract the value of this design parameter of your ship from the simulation.

To analyze trim and sinkage of the ship hull:

1. Create a report for the trim:
   a) Right-click the **Reports** node and select **New Report** > **6-DOF Body Orientation**.
   b) Rename **Rigid Body Angle 1** to `Trim`.
   c) In the *Trim - Properties* window:
      - Set Angle Report Option to Rotation Y Axis.
      - Activate *Constrained Angle Range*.
2. Create a report for the sinkage:
   a) Right-click the **Reports** node and select **New Report** > **6-DOF Body Translation**.
   b) Rename **Rigid Body Translation 1** to `Sinkage`.
   c) In the *Sinkage - Properties* window, set *Direction* to `[0.0, 0.0, 1.0]`.



3. Create a monitor and plot for trim and sinkage:
   a) Multi-select the **Reports** > **Sinkage** and **Trim** nodes.
   b) Right-click on any of the highlighted nodes and select **Create Monitor and Plot from Report**.
   c) In the pop-up window, click **Single Plot**.
   d) Rename **Plots** > **Reports Plot** as `Motion Plot`.
   e) Right-click the **Plots** > **Motion Plot** node and select **Open**.
4. Save the simulation.

The setup is now complete and the simulation is ready to run.

Monitoring and Plotting Equilibrium Forces and Moments

When you employ the equilibrium motion, it is good practice that you monitor the forces and moments in the specified directions of the degrees of freedom acting on the ship hull. These forces and moments approach zero when an equilibrium is reached numerically. Monitoring these forces and moments lets you verify the quality of the equilibrium.

To monitor the equilibrium forces in Z direction:

1. Right-click the **Reports** node and select **New Report** > **6-DOF Body Force**.
2. Rename the **6-DOF Body Force 1** node to `6-DOF Body Force Z Direction`.
3. Select the **6-DOF Body Force Z Direction** node and set *Direction* to `[0.0, 0.0, 1.0]`.
4. Right-click the **6-DOF Body Force Z Direction** node and select **Create Monitor and Plot from Report**.
5. Select the **Monitors** > **6-DOF Body Force Z Direction Monitor** node and set *Trigger* to **Iteration**.

To monitor the equilibrium moment around the Y axis:

6. Right-click the **Reports** node and select **New Report** > **6-DOF Body Moment**.
7. Rename the **6-DOF Body Moment 1** node to `6-DOF Body Moment Y Axis`.

8. Select the **6-DOF Body Moment Y Axis** node and set *Direction* to `[0.0, 1.0, 0.0]`.
9. Right-click the **6-DOF Body Moment Y Axis** node and select **Create Monitor and Plot from Report**.
10. Select the **Monitors** > **6-DOF Body Moment Y Axis Monitor** node and set *Trigger* to **Iteration**.
11. Save the simulation.

Running the Simulation

You are advised to run this tutorial on a multi-core machine with four cores or more. Depending on the specifications of your computer, this tutorial takes one to two days to complete.

To run the simulation:

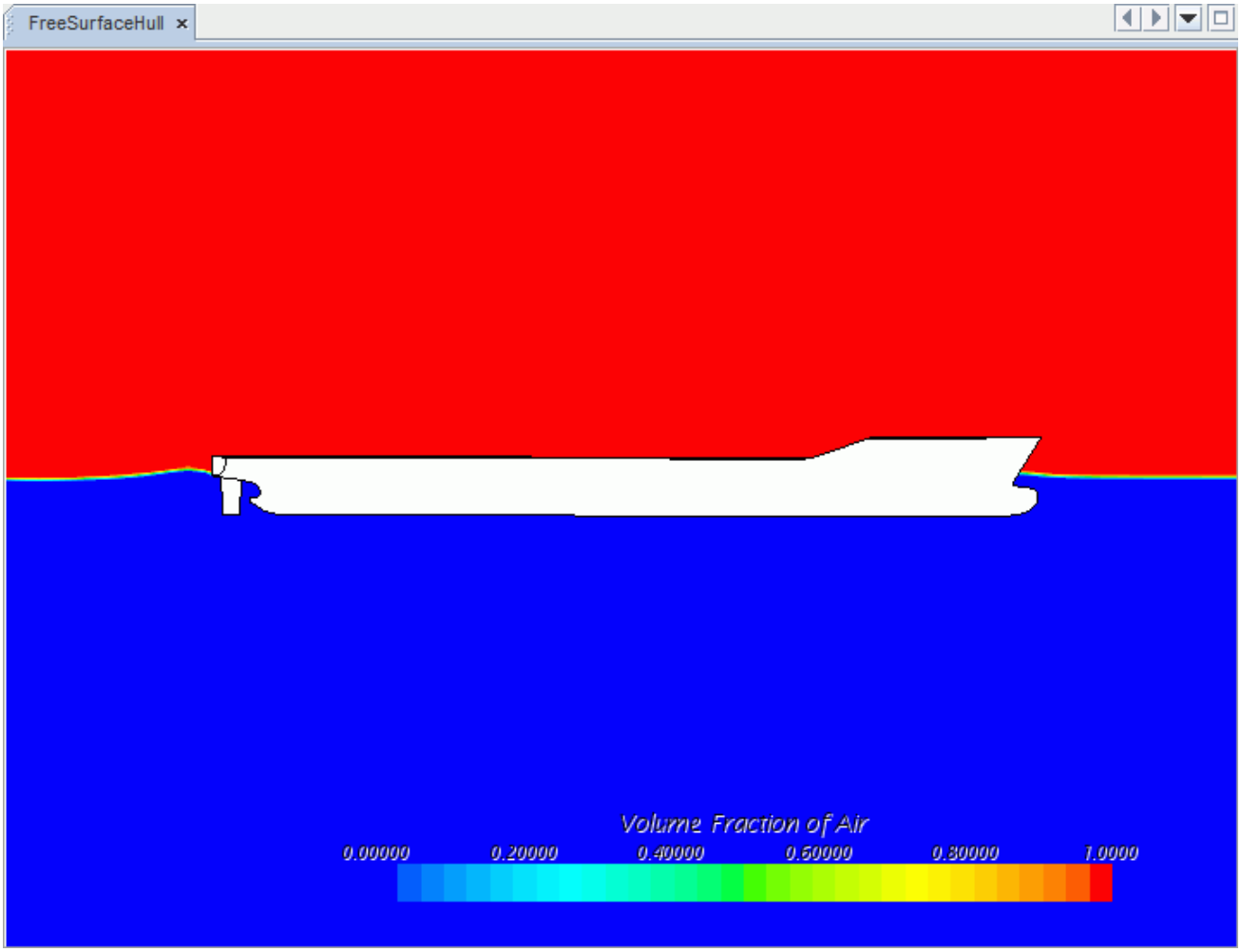
1. Click the (**Run**) button on the STAR-CCM+ toolbar.

While the simulation is running, you can click the tabs at the top of the *Graphics* window to view each of the scenes. The Residuals display is created automatically and shows the progress of the solvers.

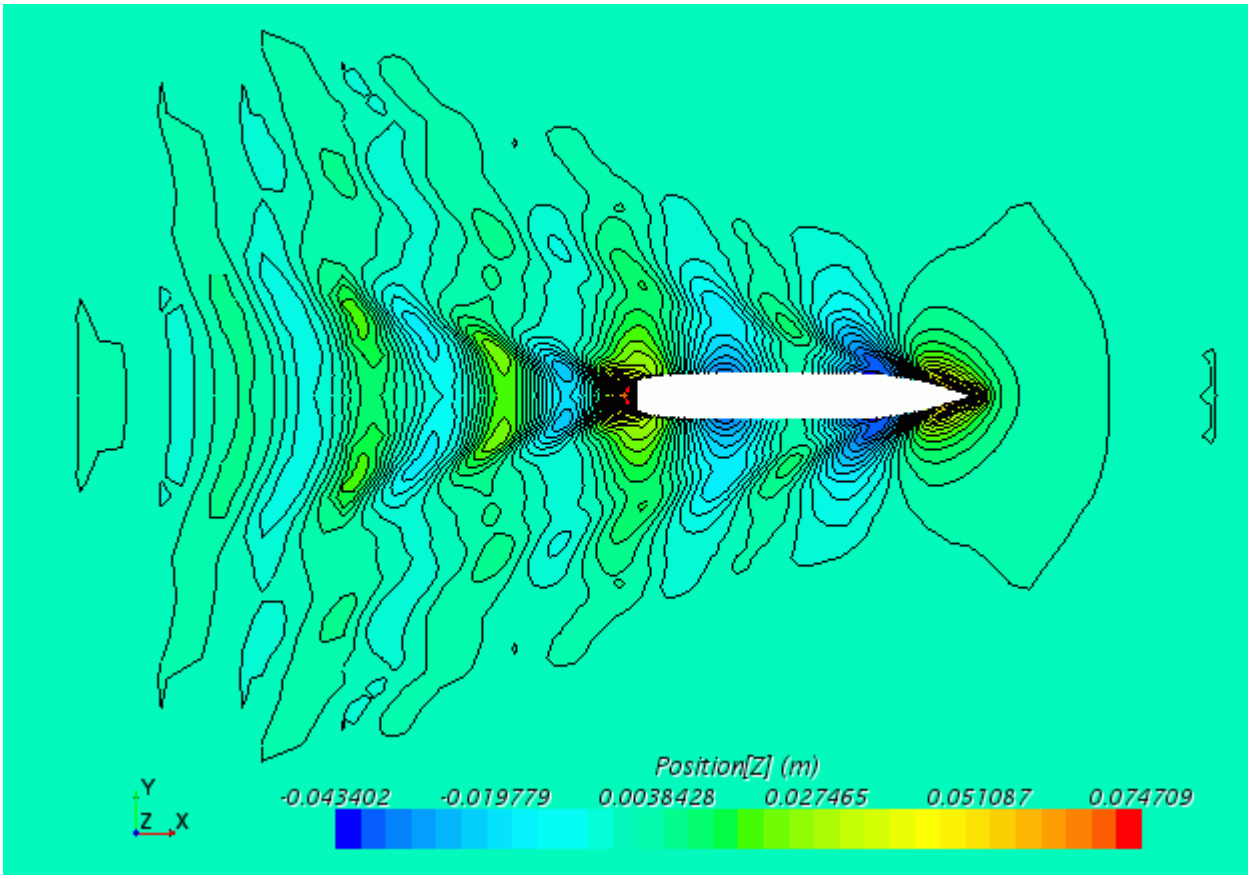
Visualizing the Results

You can follow the progress of the solution using each of the scenes that you created previously.

The following screenshot shows a close-up of the free water surface around the ship hull on the symmetry plane at the end of the simulation:



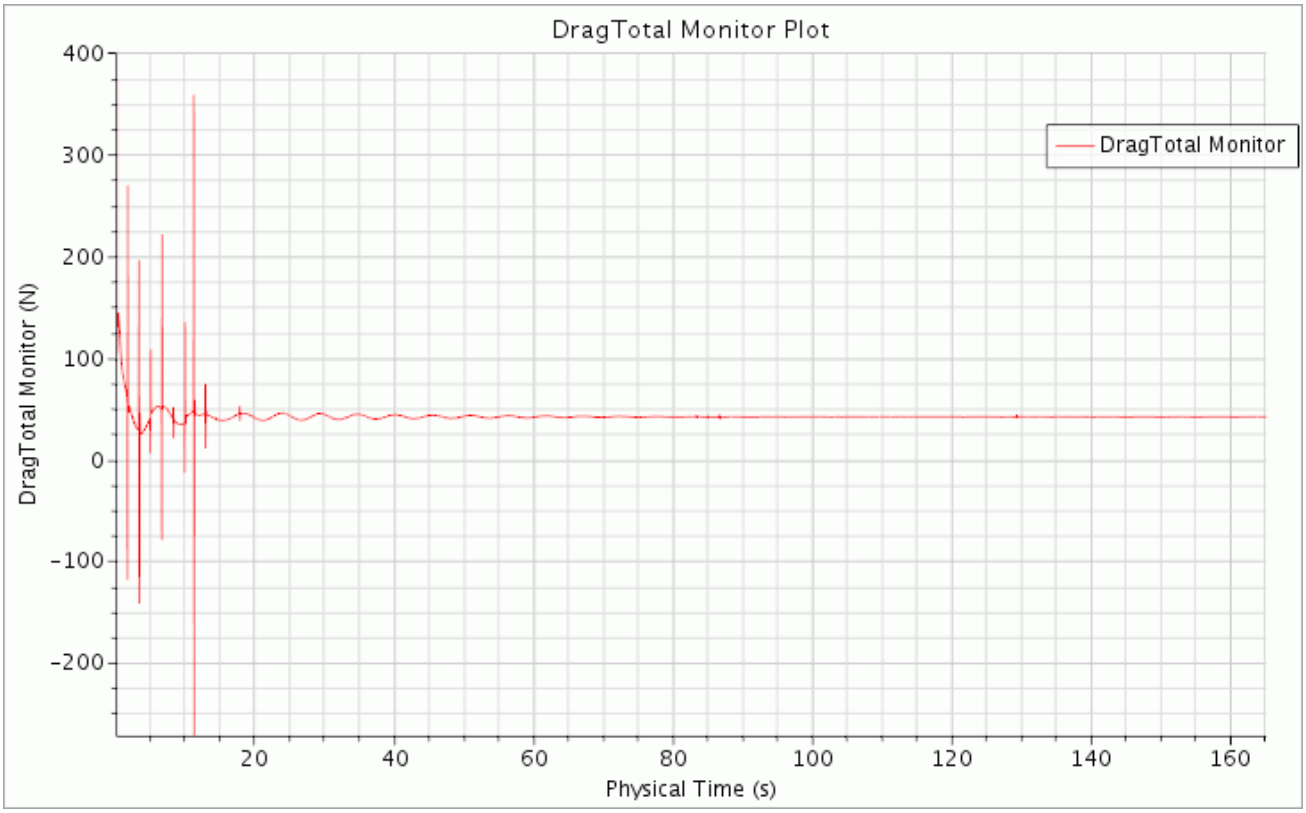
The following screenshot shows the wave pattern around the ship hull at the end of the simulation:



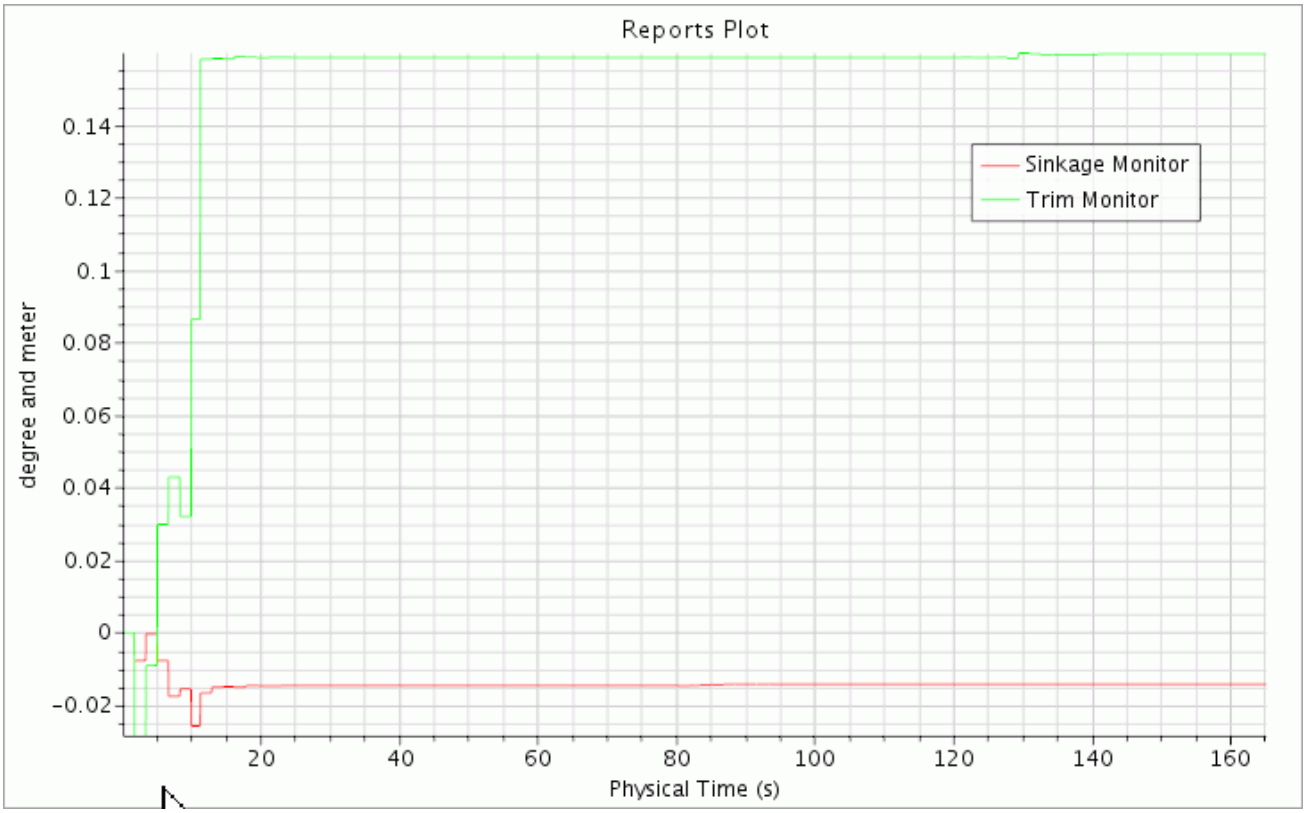
The following screenshot shows a plot of shear and pressure drag over time:

Here, you can see strong oscillations at the beginning of the simulation. These oscillations level out as the simulation progresses to reach a quasi steady-state solution after approximately 100 s.

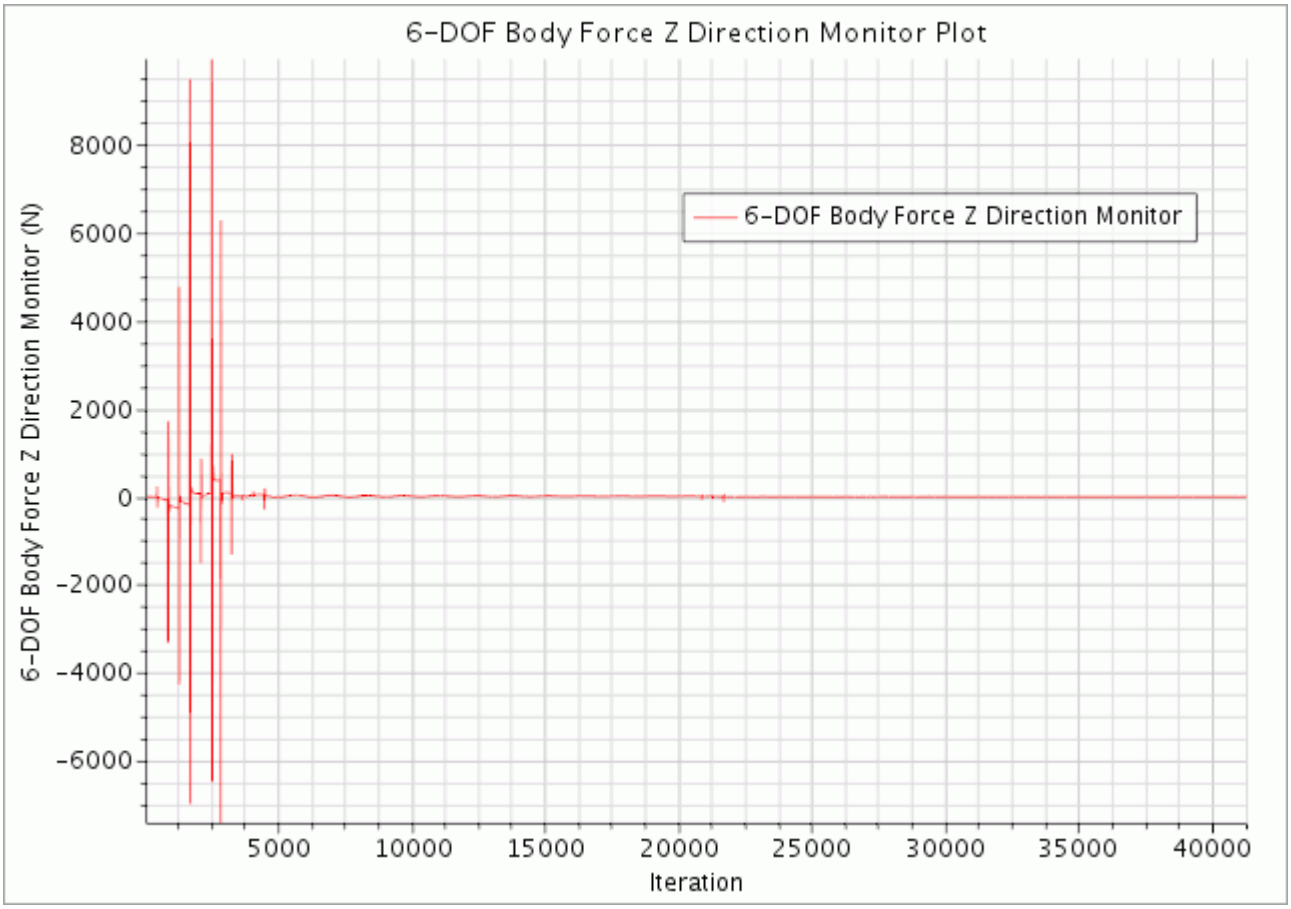
The following screenshot shows a plot of total drag over time:



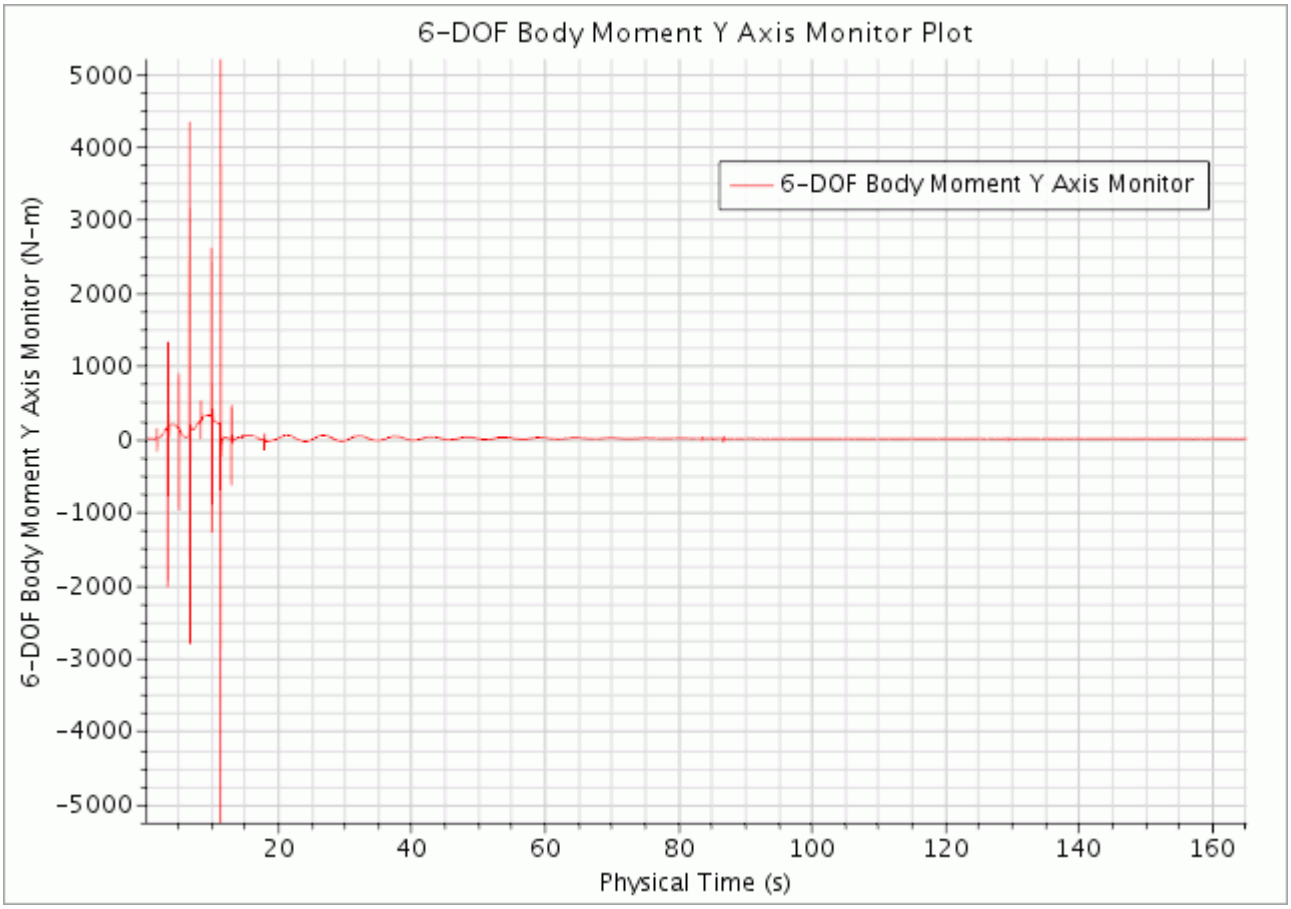
The following screenshot shows a plot of trim and sinkage over time:



The following screenshot shows the force in Z direction acting on the ship hull:



The following screenshot shows the moment around the Y axis acting on the ship hull:



Comparing with Experimental Results

For the Gothenburg Workshop 2010, experimental results were provided for comparison with simulations.

Here, the resistance coefficient is investigated:

$$c_d = \frac{F_d}{\frac{\rho}{2} v^2 A}$$

where:

Symbol	Description	Value
c_d	Resistance coefficient	
F_d	Total drag force (N)	41.876
v	Ship velocity (m/s)	2.196
A	Scaled area of hull and rudder (m²)	9.55275

The total drag force in the table above is obtained from the simulation.

1. Export the total drag over time plot:

a) Right-click the **Plots** > **DragTotal Plot** node and select **Export...**.

b) Set **File Name** to `TotalDrag.csv` and click **Save**.

2. Import the table into a spreadsheet program of your choice and perform a numerical averaging (arithmetic mean) of the total drag values over time from 135 – 165 s.
3. Calculate the resistance coefficient according to the equation above.

   Note that the simulation models only half of the ship. To match the experimental data, either the drag or the resistance coefficient must be multiplied by 2.

Resistance Coefficient (Predicted)	Resistance Coefficient (Experimental)	Deviation
0.003645	0.003711	1.77%

This result is obtained on a grid that is rather coarse for this type of simulation. It is expected that the deviation decreases with increasing mesh refinement.

Summary

This tutorial has introduced the following STAR-CCM+ features:

- Importing a ship hull geometry.
- Creating a towing tank around the ship.
- Generating a locally refined volume mesh using volumetric controls.
- Setting up a VOF multiphase simulation.
- Setting up a flat VOF Wave.
- Setting up boundary-wise wave damping.
- Setting up a DFBI rotation and translation motion.
- Setting up and running a transient simulation.
- Creating specific post-processing items for flow resistance prediction analysis.

Bibliography

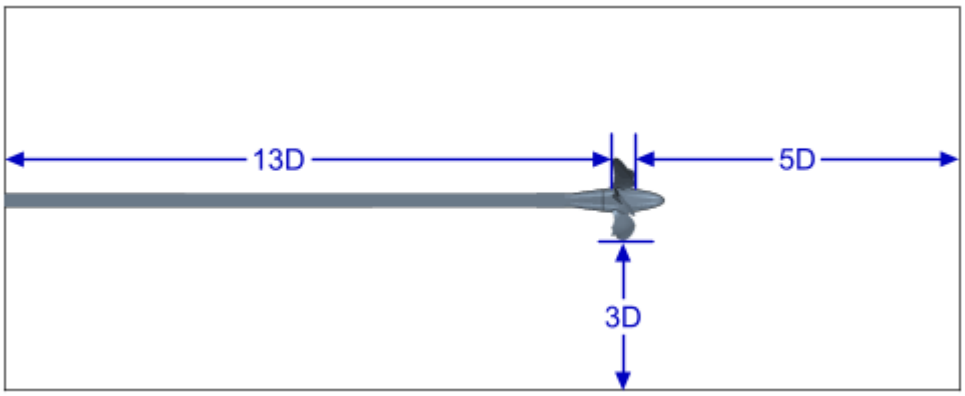
[803] *http://www.simman2008.dk/KCS/kcs_geometry.htm. Downloaded July 2013.*

Moving Reference Frames: Marine Propeller in Open Water

This tutorial demonstrates the workflow for setting up an open water simulation for a marine propeller.

The geometry that is used in this simulation is the Potsdam Propeller Test Case (PPTC) VP1304. The results that you obtain in STAR-CCM+ are compared with experimental data that is provided by SVA. This data was presented at the workshop on propeller performance during the International Symposium on Marine Propulsors held in Hamburg in June 2011.

The PPTC VP1304 propeller is placed in a virtual water tank as shown below:



The PPTC VP1304 propeller is a controllable pitch propeller that includes a 0.3 mm gap between the hub and the propeller blade at the leading and trailing edges. This gap is maintained in the simulation.

The rotation of the propeller is modelled using MRF.

The simulation is run for a range of advance coefficient values J (ranging from 0.6 to 1.4). The operating conditions for the simulation are listed as follows:

Density of Water (at 17.5°C)	998.67 kg/m^3
Dynamic Viscosity of Water (at 17.5°C)	0.001068 Pa s
Propeller Diameter	0.25 m
Number of Revolutions	15 rps

You are provided with the propeller performance data plot. The thrust coefficient K_T, the torque coefficient K_Q, and the open water efficiency η_O are calculated for each value of J and are plotted in the propeller performance plot for comparison.

Contents:

Prerequisites
Loading the Simulation File
Reviewing the Geometry
Selecting the Mesh Models

Setting Up Boundary Conditions and Values
Creating the Reports for the Performance Data
Monitoring the Propeller Performance Plot
Running the Simulation
Plotting the Propeller Performance Data
Summary
Bibliography

Prerequisites

The instructions in the Open Water tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Techniques	Associated Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Loading the Simulation File

For this tutorial, you are provided with a simulation file that contains pre-defined simulation objects. Due to the large number of cells, and the time it takes the solver to run, it is recommended that you run this tutorial in parallel on at least 10 cores.

Load the initial simulation:

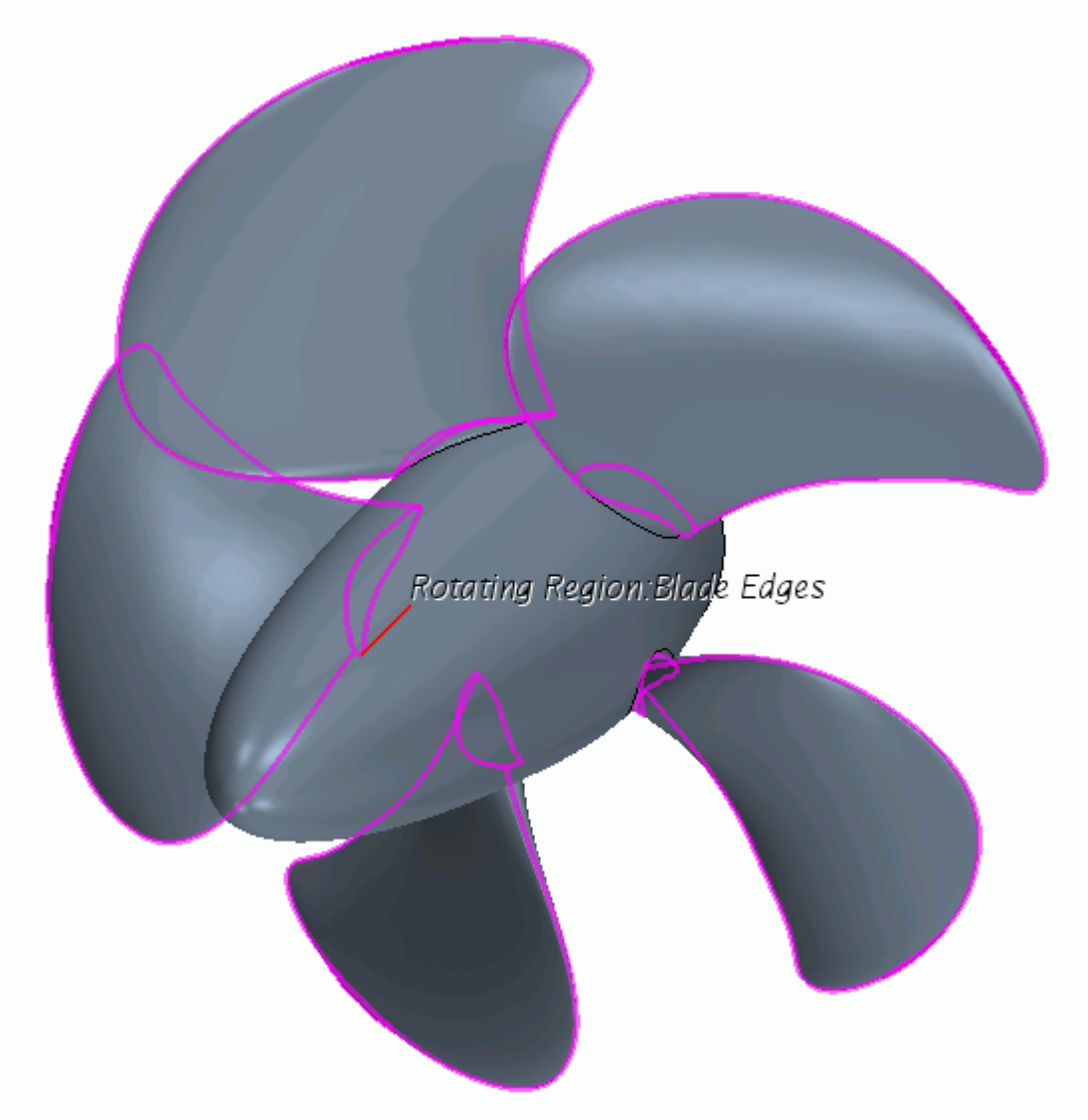
1. Launch STAR-CCM+.
2. Select **File** > **Load...**
3. In the *Load a File* dialog, click **Browse**.
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/motion/data`
5. Select `openWaterPropeller_start.sim`, then click **Open**.
6. In the *Load a File* dialog, click **OK**.
7. Save the simulation as `openWaterPropeller_run.sim`.

Reviewing the Geometry

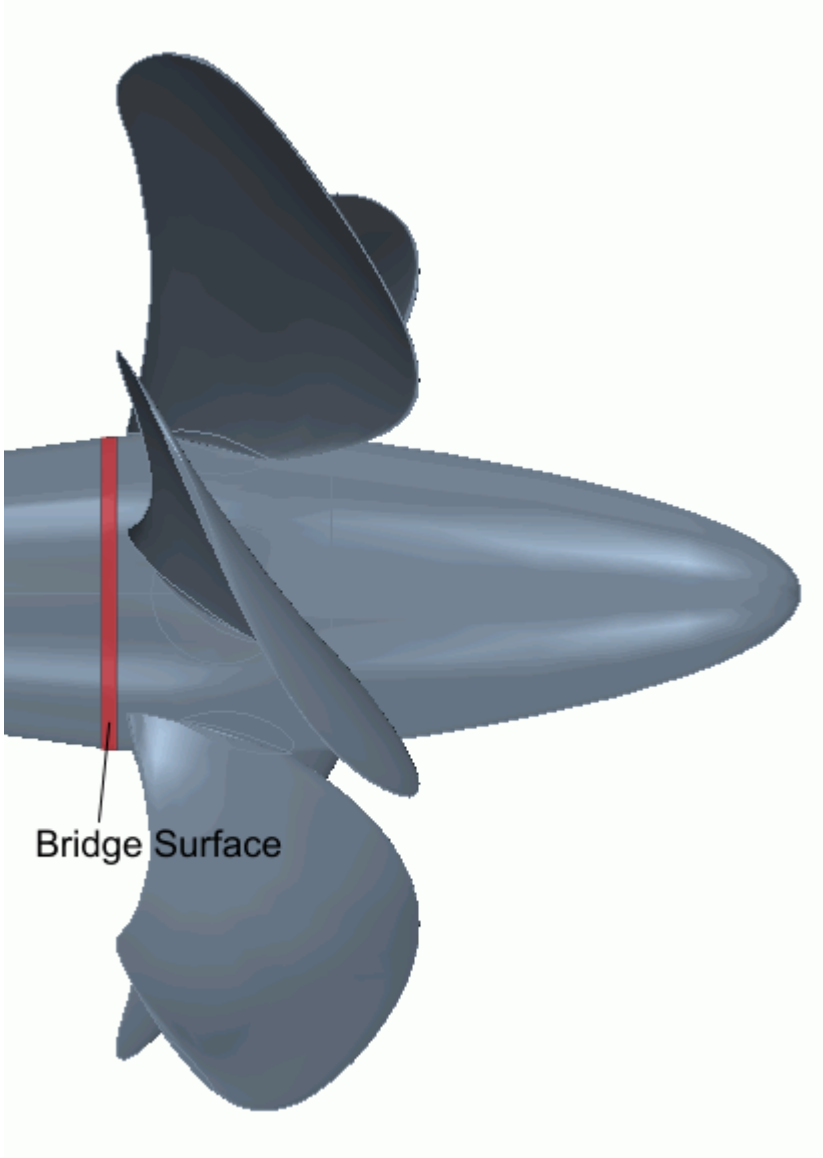
For open water simulations, it is recommended that your geometry model exhibits certain characteristics.

The geometry that is provided in the starting simulation for this tutorial demonstrates several best practices for this type of simulation:

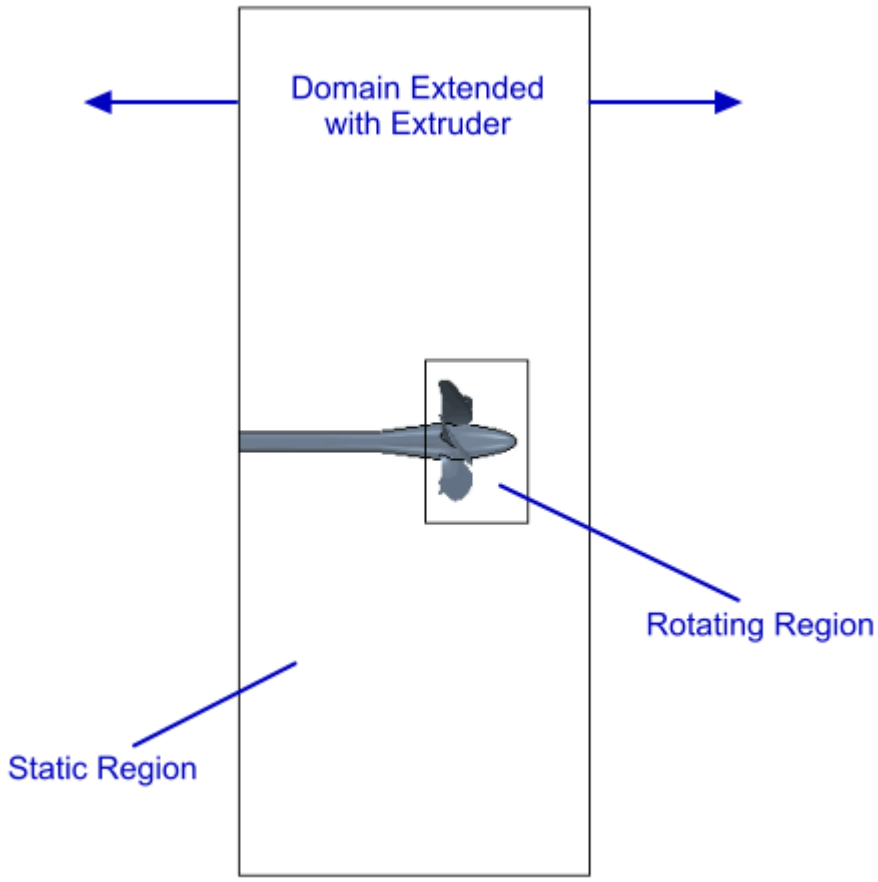
- There is a feature curve along the leading and trailing edge of the propeller to demarcate the pressure and suction sides of the propeller. During the meshing process, this feature curve allows you to capture the shape of the blade with a relatively coarse mesh.



- Small details in the geometry are closed off. In this simulation, the clearance gap between the propeller hub and the adjacent shaft has been bridged with an additional surface, as indicated in the following screenshot.



The motion of the propeller is modelled using MRF. The starting simulation file contains two regions, one for the rotating propeller, and the other for the static fluid domain. The static region is extended using the extruder mesh model.



Selecting the Mesh Models

For this type of simulation, the quality of the result depends to a large extent on how successful the mesh is at capturing features of the flow around the propeller.

In this simulation, you use the trimmer mesh model for the rotating region and its surroundings. The static region surrounding the shaft is meshed using the extruder mesher as this minimizes the computational costs.

An important criteria to consider when choosing mesh sizes is the wall y+ value. In general, you are advised to keep the wall y+ values outside the buffer layer, that is values from 5 to 30. In this tutorial, you aim for wall y + values greater than 30.

The meshing strategy adopted in this tutorial adheres to the Parts-Based Meshing (PBM) approach. This meshing strategy executes a pipeline of mesh operations on geometry parts; hence, you can modify an input part and the changes are propagated through the pipeline to the volume mesh. The extruder mesh also forms part of the pipeline operations.

To create the automated mesh operation:

1. Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
2. In the *Create Automated Mesh Operation* dialog, select **Parts** > **Rotating Region** and **Static Region**.
3. Activate the following meshing models:

Group box	Model
Surface Meshers	**Surface Remesher**
Core Volume Meshers	**Trimmed Cell Mesher**
Optional Boundary Layer Meshers	**Prism Layer Mesher**

4. Click **OK**.

To set the controls for the operation:

5. Select the **Operations** > **Automated Mesh** node and activate *Per-Part Meshing*.
6. Select the **Automated Mesh** > **Meshers** > **Trimmed Cell Mesher** node and activate *Perform Mesh Alignment*.

Contents:

Specifying the Default Controls
1Refining the Propeller Blades
Controlling the Prism Layer Size in the Clearance Gap
Specifying Prism Layer Size on the Interface
Specifying the Far Field Surface Size
Refining the Area Around the Propeller
Generating the Core Mesh
Generating the Extruder Mesh

Specifying the Default Controls

Specify the default mesh controls for this simulation:

1. Edit the **Operations** > **Automated Mesh** > **Default Controls** node and set the following properties:

Node	Property	Setting
Base Size	*Value*	`9 mm`
Minimum Surface Size	*Percentage of Base*	`0.5`
Surface Growth Rate	*Surface Growth Rate*	`1.3`
Number of Prism Layers	*Number of Prism Layers*	`3`
Prism Layer Total Thickness	*Percentage of Base*	`20`
Mesh Alignment Location	*Location*	`[0,0,0]`

1Refining the Propeller Blades

Set up custom mesh conditions to refine the mesh on the propeller blades.

1. Right-click the **Operations** > **Automated Mesh** > **Custom Controls** node, and select **New** > **Surface Control**.

2. Rename the **Surface Control** node to `Blade Surfaces`.
3. Select the **Blade Surfaces** node and set *Part Surfaces* to **Rotating Region** > **Blades**.
4. Edit the **Custom Controls** > **Blade Surfaces** > **Controls** node and set the following properties:

Node	Property	Setting
Target Surface Size	*Target Surface Size*	**Custom**
Minimum Surface Size	*Minimum Surface Size*	**Custom**
Surface Curvature	*Curvature*	**Custom**
Trimmer Surface Growth Rate	*Trimmer Surface Growth Rate*	**Custom**

5. Edit the **Blade Surfaces** > **Values** node and set the following properties:

Node	Property	Setting
Target Surface Size	*Percentage of Base*	`25`
Minimum Surface Size	*Percentage of Base*	`5`
Surface Curvature	*# Pts/circle*	`72`
Trimmer Surface Growth Rate	*Trimmer Surface Growth Rate*	**Fast**

Use the feature curve that splits the pressure side and suction side of the propeller to refine the mesh on the blade edges:

6. Right-click the **Operations** > **Automated Mesh** > **Custom Controls** node and select **New** > **Curve Control**.
7. Rename the **Curve Control** node as `Blade Edges`.
8. Select the **Blade Edges** node and set *Part Curves* to **Rotating Region** > **Blade Edges**.
9. Edit the **Blade Edges** > **Controls** node and set the following properties:

Node	Property	Setting
Target Surface Size	*Target Surface Size*	**Custom**
Minimum Surface Size	*Minimum Surface Size*	**Custom**

10. Edit the **Blade Edges** > **Values** node and set the following properties:

Node	Property	Setting
Target Surface Size	*Percentage of Base*	`5`
Minimum Surface Size	*Percentage of Base*	`2`

Controlling the Prism Layer Size in the Clearance Gap

The propeller in this simulation is a controllable pitch propeller, which means that there is a small clearance gap between each blade and the hub. In order to accurately resolve the boundary layer in this gap, modify the prism layer settings.

1. Select the **Operations** > **Automated Mesh** > **Meshers** > **Prism Layer Mesher** node.
2. Set the following in the *Prism Layer Mesher - Properties* Window.

Node	Property	Value
	Minimum Thickness Percentage	1
	Layer Reduction Percentage	5
	Boundary March Angle	25

Specifying Prism Layer Size on the Interface

The interface between the rotating region and the surrounding static region is cylindrical in shape. It is good practice to have cells that are normal to each other on either side of an interface.

To achieve this result, grow a single prism layer cell on from either side of the interface:

1. Right-click the **Operations** > **Automated Mesh** > **Custom Controls** node, and select **New** > **Surface Control**.
2. Rename this Surface Control to `Interface Prism Layer`.
3. Select the **Interface Prism Layer** node and set the *Part Surfaces* to **Rotating Region** > **Interface**, and **Static Region** > **Interface** respectively.
4. Select the **Interface Prism Layer** > **Controls** > **Prism Layers** node, and set **Prism Layers** node to **Custom.**
5. Select the **Controls** > **Prism Layers** > **Customize** node, and set the following properties:

Property	Setting
Customize Number of Layers	**Activated**
Customize Total Thickness	**Activated**
Override Boundary Defaults	**Activated**

6. Under the **Values** > **Custom Prism Values** node, set the following properties:

Node	Property	Setting
Number of Prism Layers	*Number of Prism Layers*	1
Override Boundary Defaults	*Override Boundary Defaults*	**Activated**
Prism Layer Total Thickness	*Percentage of Base*	50

Specifying the Far Field Surface Size

As flow behavior far from the propeller region is not particularly important for these simulations, you can specify a coarser mesh on the far field, inlet, and outlet boundaries.

1. Right-click the **Operations** > **Automated Mesh** > **Custom Controls** node, and select **New** > **Surface Control**.

2. Rename this Surface Control to **Far Field**.
3. Select the **Far Field** node and set the *Part Surfaces* to **Static Region** > **Far Field**, **Inlet** and **Outlet** respectively.
4. Select the **Custom Controls** > **Far Field** node, and set the following properties:

Node	Property	Setting
Controls > **Target Surface Size**	*Target Surface Size*	**Custom**
Values > **Target Surface Size**	*Percentage of Base*	`1600`

Refining the Area Around the Propeller

The area around the propeller requires a finer mesh to make sure that the flow characteristics are adequately captured.

Create a volumetric control region around the propeller:

1. Right-click the **Geometry** > **Parts** node, and choose **New Shape Part** > **Cylinder**.
2. Set the following parameters in the *Create Cylinder Part* dialogue:

Property	Setting
Start Coordinate	`[0.2, 0.0, 0.0] m`
End Coordinate	`[-0.25, 0.0, 0.0] m`
Cylinder Radius	`0.18 m`

3. Click **OK**.

   A new *Cylinder* shape part is added to the **Geometry** > **Parts** node.
4. Right-click the **Geometry** > **Operations** > **Automated Mesh** > **Custom Controls** node and select **New** > **Volumetric Control**.
5. Select **Volumetric Control** and set *Parts* to **Cylinder**.
6. Edit the **Volumetric Control** node and set the following properties:

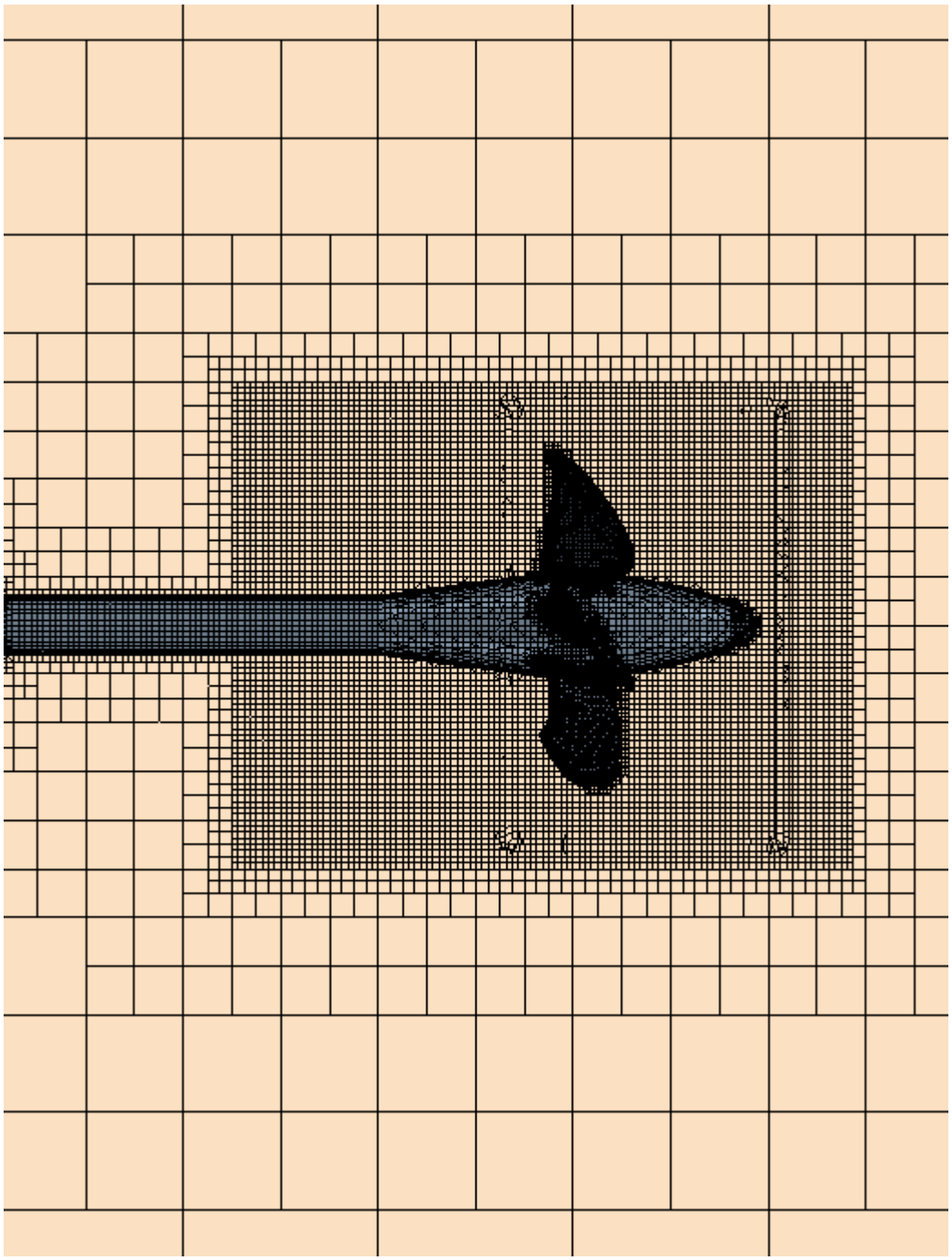
Node	Property	Setting
Controls		
└ **Trimmer**	*Customize isotropic size*	**Activated**
Values		
└ **Custom Size**	*Percentage of Base*	`50`

Generating the Core Mesh

The core regions are set up, ready for meshing. The mesh generation process takes roughly 10 minutes to complete, depending on the hardware you are using.

1. In the *Mesh Generation Toolbar*, click **Generate Volume Mesh**.
2. Open **Mesh Scene 1**.
3. Right-click a blank area in the scene and select **Apply Representation** > **Volume Mesh**. You may need to refresh the scene for the section plane to display correctly.

Note: There may be small differences in the mesh depending on the version of STAR-CCM+ used.



4. Save the simulation.

Generating the Extruder Mesh

Use the extruder mesh to extend the domain in the inflow and outflow directions. As the flow in these directions is generally normal to the cell faces, it is acceptable to generate stretched cells. In this tutorial, the largest cells have an aspect ratio of approximately 300.

First, extend the domain in the upstream direction:

1. Right-click the **Geometry** > **Operations** node and select **New** > **Surface Preparation** > **Surface Extruder**.
2. In the *Create Surface Extruder* dialog, set:
   a) *Input Parts* to **Static Region**.
   b) *Input Part Surfaces* to **Static Region** > **Inlet**.
   c) *Create New Part* to **Single New Part**.
   d) *Output Part Surfaces* to **Single Side**.
   e) *Extrusion Distance* to `1.0 m`.
   f) Activate **Create Volume Extruder from Output Parts** and **Execute Surface Extruder Operation Upon Creation**.
3. Click **OK**.
4. Expand the **Geometry** > **Operations** > **Volume Extruder** > **Controls** node and set the following properties:

Node	Property	Setting
Number of Layers	*Number of Layers*	`10`
└ **Stretching Function**	*Stretching Method*	**One Sided Geometric**
└ **One Sided Options**	*Stretching Mode*	**Thickness Ratio**
└ **One Sided Options** > **Thickness Ratio**	*Thickness Ratio*	`2`

Now extend the domain in the downstream direction:

5. Right-click the **Geometry** > **Operations** node and select **New** > **Surface Preparation** > **Surface Extruder**.
6. In the *Create Surface Extruder* dialog, set:
   a) *Input Parts* to **Static Region**.
   b) *Input Part Surfaces* to **Static Region** > **Outlet**.
   c) *Create New Part* to **Single New Part**.
   d) *Output Part Surfaces* to **All Pertinent Sides**.
   e) *Extrusion Distance* to `3.0 m`.
   f) Activate **Create Volume Extruder from Output Parts** and **Execute Surface Extruder Operation Upon Creation**.
7. Click **OK**.

8. Expand the **Geometry** > **Operations** > **Volume Extruder 2** > **Controls** node and set the following properties:

Node	Property	Setting
└ **Number of Layers**	*Number of Layers*	`40`
└ **Stretching Function**	*Stretching Function*	**One Sided Geometric**
└ **One Sided Options**	*Stretching Function*	**Thickness Ratio**
└ **One Sided Options** > **Thickness Ratio**	*Thickness Ratio*	`2`

Note: Two part placeholders appear under the **Geometry** > **Parts** node: *Surface Extruder* (upstream extension) and *Surface Extruder 2* (downstream extension).

9. Within **Geometry** > **Parts** , rename these parts to `Upstream Extension` and `Downstream Extension` respectively.
10. Multi-select the **Upstream Extension** and **Upstream Extension** nodes, right-click and select **Assign Parts to Regions**.
11. In the *Assign Parts to Regions* dialog:
    a) Select **Create a Region for Each Part**.
    b) Select **Create a Boundary for Each Part Surface**.
    c) Select **Create a Feature Curve for Each Part Curve**.
    d) Click **Apply**, then **Close**.
12. In the *Mesh Generation Toolbar*, click **Generate Volume Mesh**.
13. Create a new Mesh Scene: Right-click **Scenes** > **New Scene** > **Mesh** and examine the mesh.
14. Save the simulation.

Note: The boundaries **Static Region.Inlet** and **Static Region.Outlet** now form interfaces between the **Static Region** and the **Upstream Extension** and **Downstream Extension** regions respectively. The propeller shaft extension is named **Static Region.Shaft** as default, and the outer boundaries on the new regions are named **Default** and **Static Region.Far Field** respectively.

Setting Up Boundary Conditions and Values

The inlet, outlet, wall and symmetry boundary conditions must be defined on the boundaries of the newly created regions. The velocity of the flow, V_A, must also be set at the inlet, and as an initial condition in the fluid domain.

This velocity is defined by the advance coefficient, J:

$$J = \frac{V_A}{n \cdot D}$$

where:

- V_A is the advance velocity
- n is the rate of revolution
- D is the propeller diameter

Contents:

Defining the Advance Coefficient
Specifying Boundary Conditions

Defining the Advance Coefficient

In this steady state simulation, you specify a range of J values using two field functions: J and i. The field function, i, specifies the iteration on which the value of J changes.

In this tutorial, the starting value of J is 0.6. The value of J increases by 0.2 every 501 iterations. This number of iterations is sufficient for the solution to converge for each value of J.

First, define the field function **i**:

1. Create a custom scalar field function and rename it to `i`.
2. Select the **i** node and set the following properties:

Property	Setting
Definition	`floor ($Iteration/501)`
Function Name	`i`

Create and define the field function for **J**:

3. Create a custom scalar field function and rename it to `J`.
4. Select the **J** node and set the following settings:

Property	Setting
Definition	`0.6 + 0.2 * $i`
Function Name	`J`

Proceed to create field functions that supply the parameters for the advance velocity:

5. Create three custom scalar field functions and name them `n`, `D`, and `VA` respectively.
6. Set the following properties for the field functions:

Field Function	Property	Setting
n	*Definition*	`15`
	Function Name	`n`
D	*Definition*	`0.25`
	Function Name	`D`

Field Function	Property	Setting
VA	*Definition*	`$J * $n * $D`
	Function Name	`VA`

Specifying Boundary Conditions

Here, you define the boundary types on the extruded mesh regions: Upstream Extension and Downstream Extension, and apply the advance velocity to the inlet boundary. The boundaries on these new regions are set as walls by default.

1. Expand **Regions** > **Upstream Extension** > **Boundaries** node and rename **Default** boundary to **Far Field**
2. Set the following properties:

Node	Property	Setting
Far Field	*Type*	**Symmetry Plane**
Inlet	*Type*	**Velocity Inlet**
└ **Physics Values**		
└ **Velocity Magnitude**	*Method*	**Field Function**
	Scalar Function	**VA**

3. Edit the **Regions** > **Downstream Extension** > **Boundaries** node and set the following properties:

Node	Property	Setting
Outlet	*Type*	**Pressure Outlet**
Static Region.Far Field	*Type*	**Symmetry Plane**

Use the advance velocity as the initial velocity for the fluid domain:

4. Expand the **Continua** > **Physics 1** > **Initial Conditions** node.
5. Select the **Velocity** node and set *Value* to `[-1 * $VA, 0.0, 0.0]`.

Creating the Reports for the Performance Data

The propeller performance data consists of the advance coefficient, the thrust coefficient, the torque coefficient, and the open water efficiency.

For this case, a custom field function that specifies the value of density is provided for you.

Thrust Coefficient

The thrust coefficient, K_T, is defined as:

$$K_T = \frac{T}{\rho \cdot n^2 \cdot D^4} \tag{4164}$$

where:

- T is the thrust that is generated by the propeller
- ρ is the density of water
- n is the rate of revolution of the propeller
- D is the propeller diameter

In order to calculate this value, you must first calculate the thrust that the propeller generates. Create the report for the thrust:

1. Right-click **Reports** and select **New Report** > **Force**.
2. Rename this report to `T`.
3. Select the **T** report and set the following properties:

Property	Setting
Direction	`[1.0, 0.0, 0.0]`
Parts	**[Rotating Region: Blades, Rotating Region: Hub]**

4. Right-click **Reports** and select **New Report** > **Expression**.
5. Rename this report to `KT`.
6. Select the **KT** report and set *Definition* to `$TReport/($rho*pow($n,2)*pow($D,4)).`

Torque Coefficient

The torque coefficient, K_Q, is defined as:

$$K_Q = \frac{Q}{\rho \cdot n^2 \cdot D^5} \tag{4165}$$

where:

- Q is the torque that is generated by the propeller
- ρ is the density of water
- n is the rate of revolution of the propeller
- D is the propeller diameter

In order to calculate this value, you must first calculate the torque that the propeller generates. Create the report for the torque:

1. Right-click **Reports** and select **New Report** > **Moment**.
2. Rename this report to `Q`.
3. Select the **Q** report and set the following properties:

Property	Setting
Axis	`[-1.0, 0.0, 0.0]`

Property	Setting
Parts	**[Rotating Region: Blades, Rotating Region: Hub]**

4. Right-click **Reports** and select **New Report** > **Expression**.
5. Rename this report to `KQ`.
6. Select the **KQ** report and set *Definition* to `$QReport/($rho*pow($n,2)*pow($D,5))`.

Open Water Efficiency

The open water efficiency, η_O, is defined as:

$$\eta_O = \frac{J}{2\pi} \cdot \frac{K_T}{K_Q} \tag{4166}$$

where:

- J is the advance coefficient
- K_T is the thrust coefficient
- K_Q is the torque coefficient

Create the report for the open water efficiency:

1. Right-click **Reports** and select **New Report** > **Expression**.
2. Rename this report to `J`.
3. Select the **J** report and set *Definition* to `$J`.
4. Right-click **Reports** and select **New Report** > **Expression**.
5. Rename this report to `hO`.
6. Select the **hO** report and set *Definition* to `($KTReport/$KQReport)*($JReport/(2*3.1428571429))`.

Monitoring the Propeller Performance Plot

The propellor performance plot contains values of J and 10KQ.

1. Create an expression report and rename it to `10KQ`.
2. Select the **10KQ** report and set *Definition* to `10*$KQReport`.

The propeller performance plot includes the values of J, KT, 10KQ, and ηO. In the following steps you create monitors from these reports and modify the update frequency. The value of J changes every 501 iterations. The converged values of these reports are saved every 500 iterations.

3. Multi-select the **J**, **10KQ**, **KT**, and **hO** reports.
4. Right-click one of the nodes and choose **Create Monitor from Report**.
5. Expand the **Monitors** node and multi-select the **10KQ Monitor**, **KT Monitor**, **hO Monitor**, and **J Monitor** nodes.

6. Right-click one of the nodes and choose **Edit...**.
7. Select the **Iteration Frequency** node and set *Iteration Frequency* to `500`.
8. Click **Close**.
9. Right-click **Monitors** > **J Monitor** and select **Create Plot from Monitor**.
10. Save the simulation.

Running the Simulation

Set the stopping criteria and run the simulation.

1. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to `2500`.
2. Run the simulation.

   The simulation can take a few hours to complete, depending on the number of cores you are running.
3. Save the simulation.

Plotting the Propeller Performance Data

When complete, the simulation contains all the data that is required to plot the propeller performance data and compare it with the experimental results. The tutorial starting simulation already has a basic plot for you to use.

Begin by exporting the data to a `.csv` file:

1. Expand the **Monitors** node.
2. Multi-select the **10KQ Monitor**, **KT Monitor**, **hO Monitor**, and **J Monitor** nodes.
3. Right-click on one of the selections and select **Export...**
4. In the *Save* dialog, save the file as `propellerPerformanceData.csv` in the working directory.

   The data is exported into a single file that contains five columns of data.
5. Right-click **Tools** > **Tables** and select **New Table** > **File Table**.
6. In the *Open* dialog, select `propellerPerformanceData.csv`.

Update the plot:

7. Expand the **Plots** > **Propeller Performance Data** node.
8. Right-click **Data Series** and select **Add Data**.
9. In the *Add Data Providers to Plot* dialog, select **propellerPerformanceData** and click **OK**.
10. Edit the **propellerPerformanceData** node and set the following properties:

Node	Property	Setting
propellerPerformanceData	*Legend Name*	`KT`
	X Column	**J Monitor: Expression**
	Y Column	**KT Monitor: Expression**

Node	Property	Setting
└ **Line Style**	*Color*	**Black**
	Style	**Solid**
	Width	2
└ **Symbol Style**	*Size*	10
	Color	**Black**
	Shape	**Filled Triangle**

11. Create two more data sets from the **propellerPerformanceData** data, one for the `10KQ` data, and the other for the `hO` data.
12. Edit the **propellerPerformanceData 2** node and set the following properties:

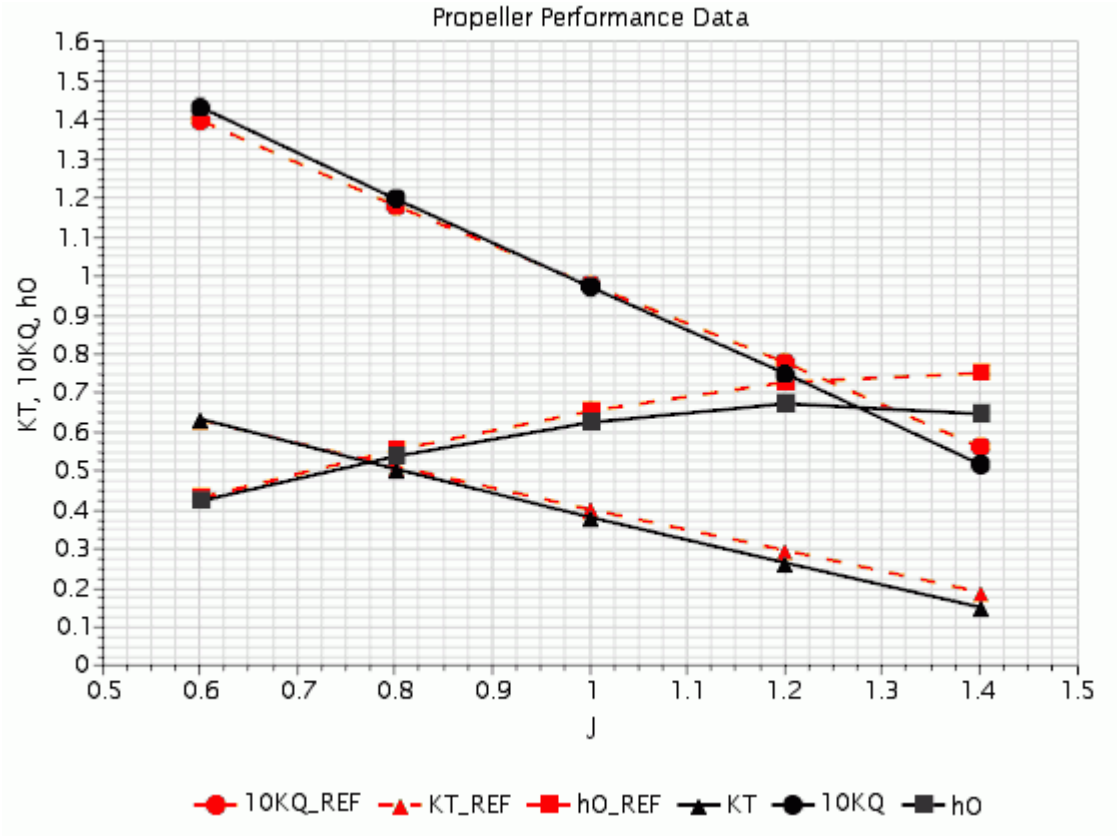
Node	Property	Setting
propellerPerformanceData 2	*Legend Name*	10KQ
	X Column	**J Monitor: Expression**
	Y Column	**10KQ Monitor: Expression**
└ **Line Style**	*Color*	**Black**
	Style	**Solid**
	Width	2
└ **Symbol Style**	*Size*	10
	Color	**Black**
	Shape	**Filled Circle**

13. Edit the **propellerPerformanceData 3** node and set the following properties:

Node	Property	Setting
propellerPerformanceData 3	*Legend Name*	hO
	X Column	**J Monitor: Expression**
	Y Column	**hO Monitor: Expression**
└ **Line Style**	*Color*	**Black**
	Style	**Solid**
	Width	2
└ **Symbol Style**	*Size*	10
	Color	**Black**
	Shape	**Filled Square**

14. Open the **Propeller Performance Data** plot.

The following plot displays the STAR-CCM+ calculations as black solid lines, and the experimental measurements as red long dashed lines.



Summary

This tutorial demonstrated the workflow for setting up an open water simulation for a marine propeller.

This tutorial has introduced the following STAR-CCM+ features:

- Importing a propeller geometry and checking that it contains recommended features for marine simulations.
- Extending the fluid domain using the extruder mesher.
- Setting up a steady state simulation to run with varying advance coefficent values.
- Setting up and exporting propeller performance data to a .csv file.
- Plotting the propeller performance data that is calculated in STAR-CCM+ against experimental data.

Bibliography

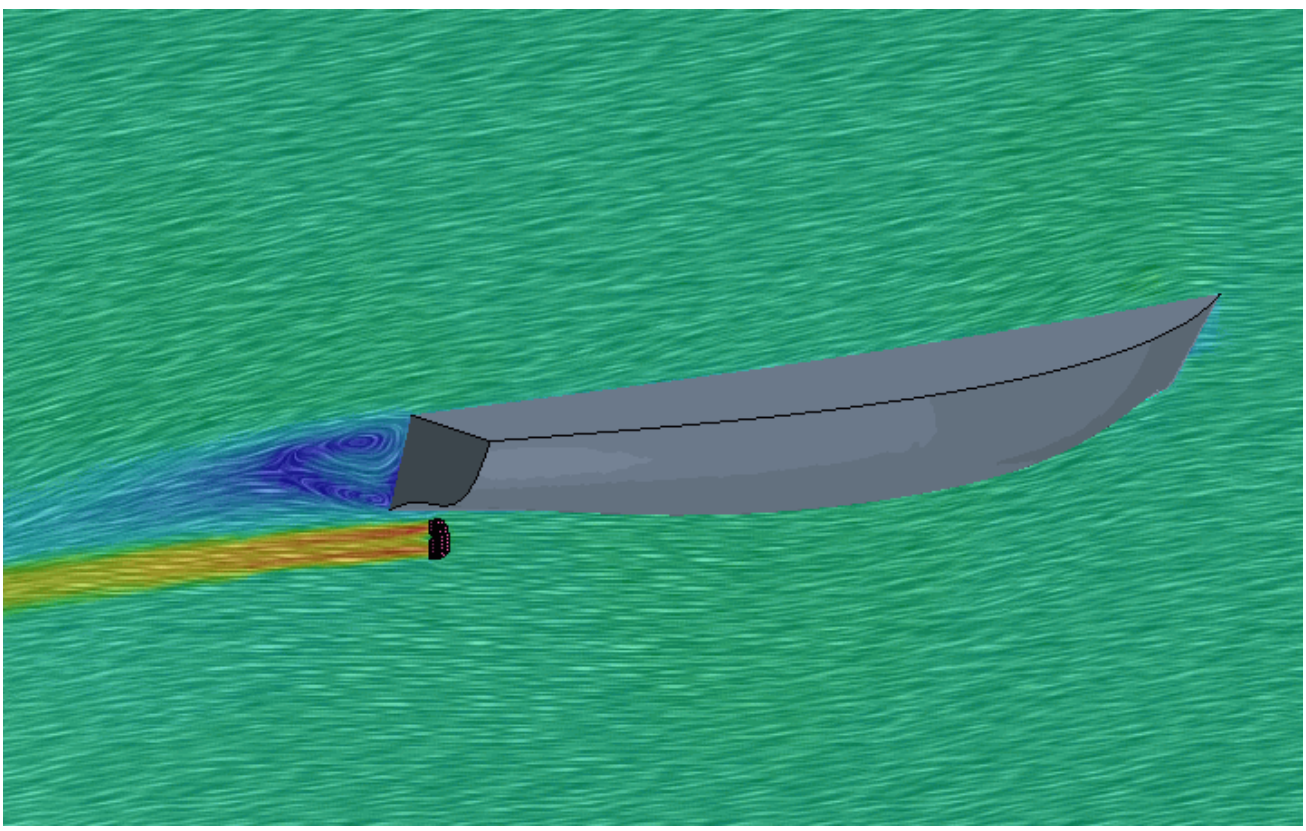
[804] *http://www.sva-potsdam.de/pptc_data.html#geometry. Downloaded October 2013.*

Body Force Propeller Method: Marine Self-Propulsion

One of the challenges in marine engineering is to predict the speed with which a ship hull moves through water in response to the thrust supplied by a spinning propeller. STAR-CCM+ provides a methodology by which you can predict this speed.

This tutorial simulates the motion of a ship through calm water in response to the thrust produced by its propeller. The propeller is placed centrally at the stern of the ship and rotates at 2300 rpm. Propeller effects are important in these type of simulations, as they can significantly affect the performance of the hull due to the swirl and pressure gradients.

To model self-propulsion, the ship is placed inside a virtual towing tank in full scale. In this tutorial, you apply the Body Force Propeller method of the Virtual Disk model to represent the effect of the propeller on the ship. The Body Force Propeller method allows you to model the propeller and hull interaction without the added cost of modeling the propeller geometry with a fine mesh and smaller time scales. For more information, see Body Force Propeller Method in the User Guide.



Specific models used in this tutorial are:

- Volume of Fluid (VOF) for the water and air. The simulation starts from a flat water surface.
- Dynamic Fluid Body Interaction (DFBI) for calculating the ship motion in response to fluid forces acting on it. The ship is not allowed to move forwards or backwards, but is free to heave and pitch.

In this simulation, the ship moves at a steady velocity with no acceleration. To achieve this condition in STAR-CCM+, the thrust induced by the propeller must balance the drag forces on the ship hull. However, you do not know in advance the inlet water velocity that is required to generate drag forces that balance the thrust. This tutorial describes a process for discovering the inlet velocity that results in an overall force balance of zero. Obtaining this balancing inlet velocity is equivalent to obtaining the hull speed as it represents the speed of the ship traveling through calm water relative to a fixed reference frame.

The model uses the full geometry of the ship and does not employ symmetry conditions.

The following boundary conditions are applied:

- inlets with prescribed velocity and volume fraction at the upstream, top, bottom, and side boundaries
- hydrostatic pressure corresponding to the undisturbed water surface at the downstream boundary of the tank
- walls on the ship hull surface

Contents:

Prerequisites
Loading the Simulation File
Selecting the Physics Models
Setting Up Dynamic Fluid Body Interaction (DFBI)
Modeling the Propeller by Using the Virtual Disk Model
Applying the Thrust Force to the Ship
Creating Mesh Refinement for the Virtual Disk
Generating the Volume Mesh
Defining the Water and Air Phases
Defining the VOF Wave
Setting Initial Conditions
Damping Wave Reflections
Setting Up Boundary Conditions
Setting Solver Parameters and Stopping Criteria
Monitoring and Plotting the Net Force in X Direction
Visualizing the Pressure Distribution
Running the Simulation
Visualizing the Results
Obtaining the Balancing Inlet Velocity
Summary

Prerequisites

The instructions in this tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Techniques	Associated Tutorial
The STAR-CCM+ workflow	Introduction
Using visualization tools, scenes, and plots	Introduction
Meshing and visualization of virtual towing tank applications	Marine Resistance Prediction: KCS Hull with a Rudder

Loading the Simulation File

For this tutorial, you are provided with a simulation file that contains pre-defined objects and settings.

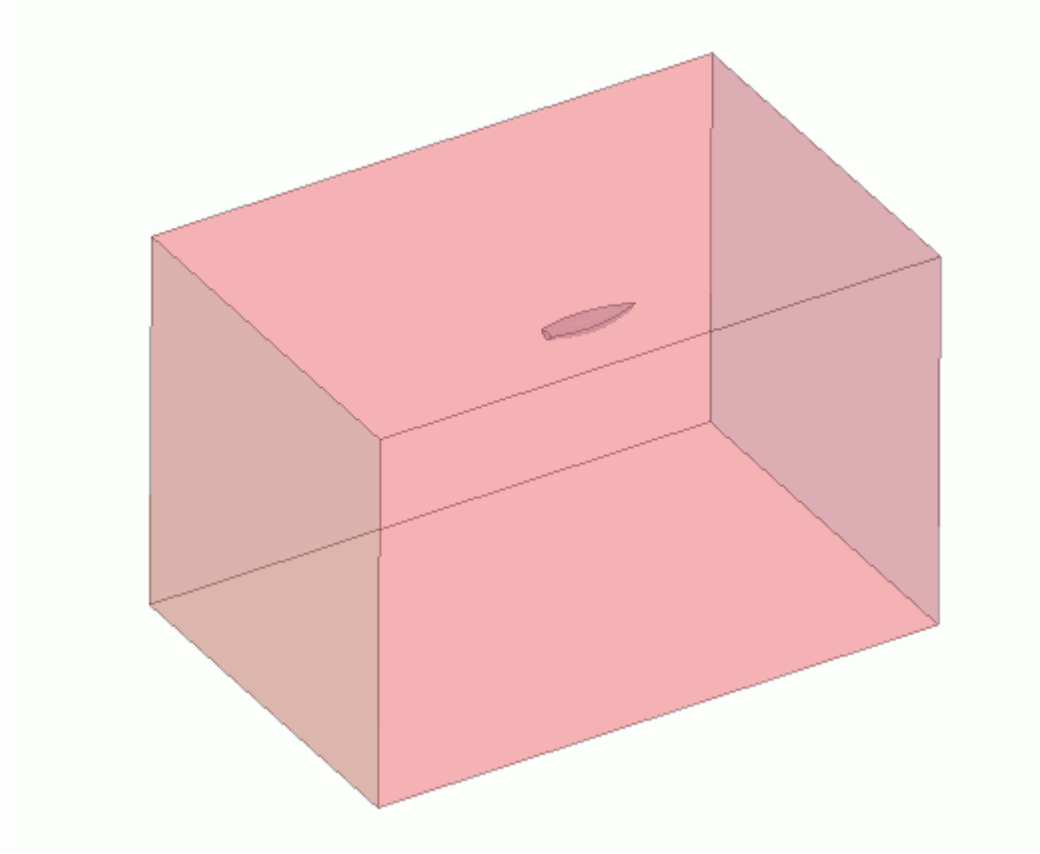
The pre-defined objects and settings include:

- geometry
- mesh operations
- boundary types

Due to the large number of cells, and the time it takes the solver to run, launch this tutorial in parallel on at least 4 cores.

To load the initial simulation:

1. Launch STAR-CCM+.
2. Select **File** > **Load Simulation...**
3. In the *Process Options* box, select an appropriate parallel option and set the number of *Compute Processes* to an amount not less than `4`.
4. In the *Load Simulation* dialog, click **Browse...**
5. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/motion/data`.
6. Select `marineSelfPropulsion_start.sim` then click **Open**.
7. In the *Load Simulation* dialog, click **OK**.
8. To visualize the imported geometry, open **Geometry Scene 1**.



9. Save the simulation as `marineSelfPropulsion_run.sim`.

Selecting the Physics Models

This marine self-propulsion simulation is a transient complex multi-physics problem for which you select a variety of physics models.

The simulation involves two fluid phases - water and air - that are modeled using the VOF multiphase model. The flow field is transient and turbulent. To model the effects of the propeller, you select the virtual disk model. To account for the hydrostatic pressure distribution under water, you activate gravity. You also use VOF waves that specify initial and boundary conditions for VOF phases as the simulation progresses.

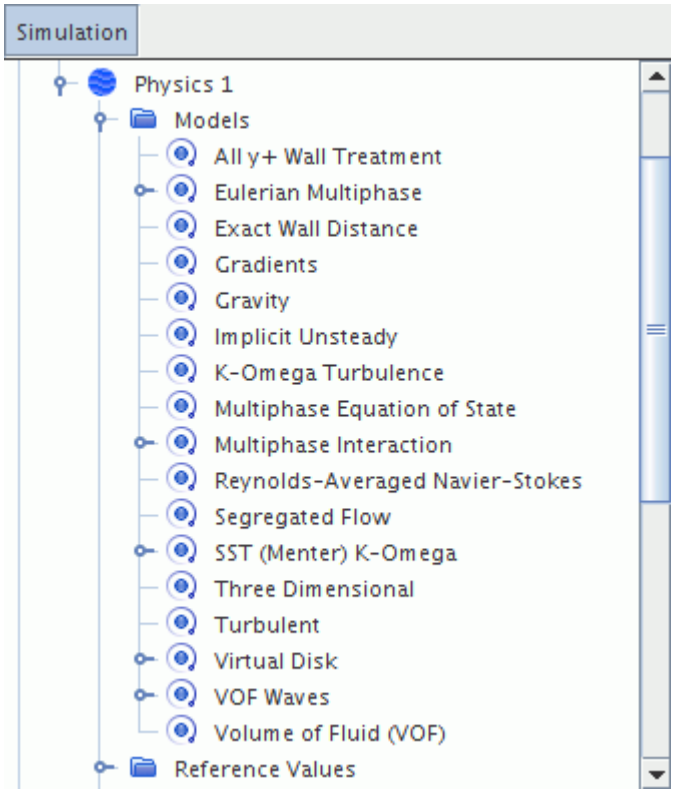
To select the physics models:

1. For the physics continuum, **Continua** > **Physics 1**, select the following models in order:

Group Box	Model
Enabled Models	**Gradients** (selected automatically) **Three Dimensional** (selected automatically)
Time	**Implicit Unsteady**
Material	**Eulerian Multiphase** **Multiphase Interaction** (selected automatically)

Group Box	Model
Eulerian Multiphase Model	**Volume of Fluid (VOF)** **Multiphase Equation of State** (selected automatically) **Segregated Flow** (selected automatically)
Viscous Regime	**Turbulent**
Reynolds-Averaged Turbulence	**K-Omega Turbulence** **Exact Wall Distance** (selected automatically) **SST (Menter) K-Omega** (selected automatically) **All y+ Wall Treatment** (selected automatically)
Optional Models	**Virtual Disk** **VOF Waves** **Gravity**

2. To review the models, open the **Physics 1** > **Models** node.



3. Save the simulation.

Setting Up Dynamic Fluid Body Interaction (DFBI)

The Dynamic Fluid Body Interaction (DFBI) model simulates the motion of the ship according to the forces acting on it. For this simulation, the ship is allowed to move with two degrees of freedom to account for sinkage and trim.

To set up the dynamic fluid body interaction:

1. Right-click the **Tools** > **Motions** node and select **New** > **DFBI Rotation and Translation**.
2. Select the **Regions** > **Domain** > **Physics Values** > **Motion Specification** node and set **Motion** to **DFBI Rotation and Translation**.

   The DFBI Rotation and Translation motion is assigned to the Domain region and a new **DFBI** node appears in the simulation tree..
3. Right-click the **DFBI** > **6-DOF Bodies** node and select **New** > **3D Body**.
4. Rename **Body 1** to `Ship`.
5. Edit the **6-DOF Bodies** node and set the following properties:

Node	Property	Setting
Ship	**Body Surface**	**Domain: Deck, Domain: Hull, Domain: Stern**
	Body Mass	`13775 kg`
	Ramp Time	`3.0 s`
└ **Free Motion**	**Z Motion**	Activated
	Y Rotation	Activated
└ **Initial Values**		
└ **Center of Mass**	**Position**	`[6.807, 0.0, 0.9] m,m,m`
└ **Moment of Inertia**	**Diagonal Components**	`[1.0, 181000.0, 1.0] kg-m^2`
	Use Center of Mass	Activated

6. Save the simulation.

Modeling the Propeller by Using the Virtual Disk Model

For this simulation, you model the effects of the propeller on the ship using the body force propeller method. This method belongs to the virtual disk model.

Using the virtual disk for a propeller requires you to define the following items:

- dimensions and location of the disk
- orientation of the disk axis. This orientation defines the direction of the thrust force.

- propeller performance data, where the thrust coefficient K_T and the torque coefficient K_Q are provided as a function of advance ratio J
- velocity plane on which to obtain an average velocity and density for the inflow surface of the virtual disk

To set up the virtual disk model:

1. Right-click the **Continua** > **Physics 1** > **Models** > **Virtual Disk** > **Virtual Disks** node and select **New**.
2. Edit the **Virtual Disk** node and set **Method** to **Body Force Propeller Method**.

Before you can specify the propeller curve, you must import the propeller table data file (here in .csv format) into STAR-CCM+.

3. Right-click **Tools** > **Tables** and select **New Table** > **File Table**.
4. Navigate to [INSTALL_DIR]/doc/startutorialsdata/motion/data/BSeries_Z5_BAR1p05_PD0p723.csv and click **Open**.

You can now specify the propeller curve and set up the virtual disk. You set the *Inflow Plane Radius* to a value that is 10% larger than the virtual disk radius. The **Inflow Plane Offset** is 10% of the virtual disk diameter.

5. Edit the **Virtual Disks** > **Virtual Disk** node and set the following properties:

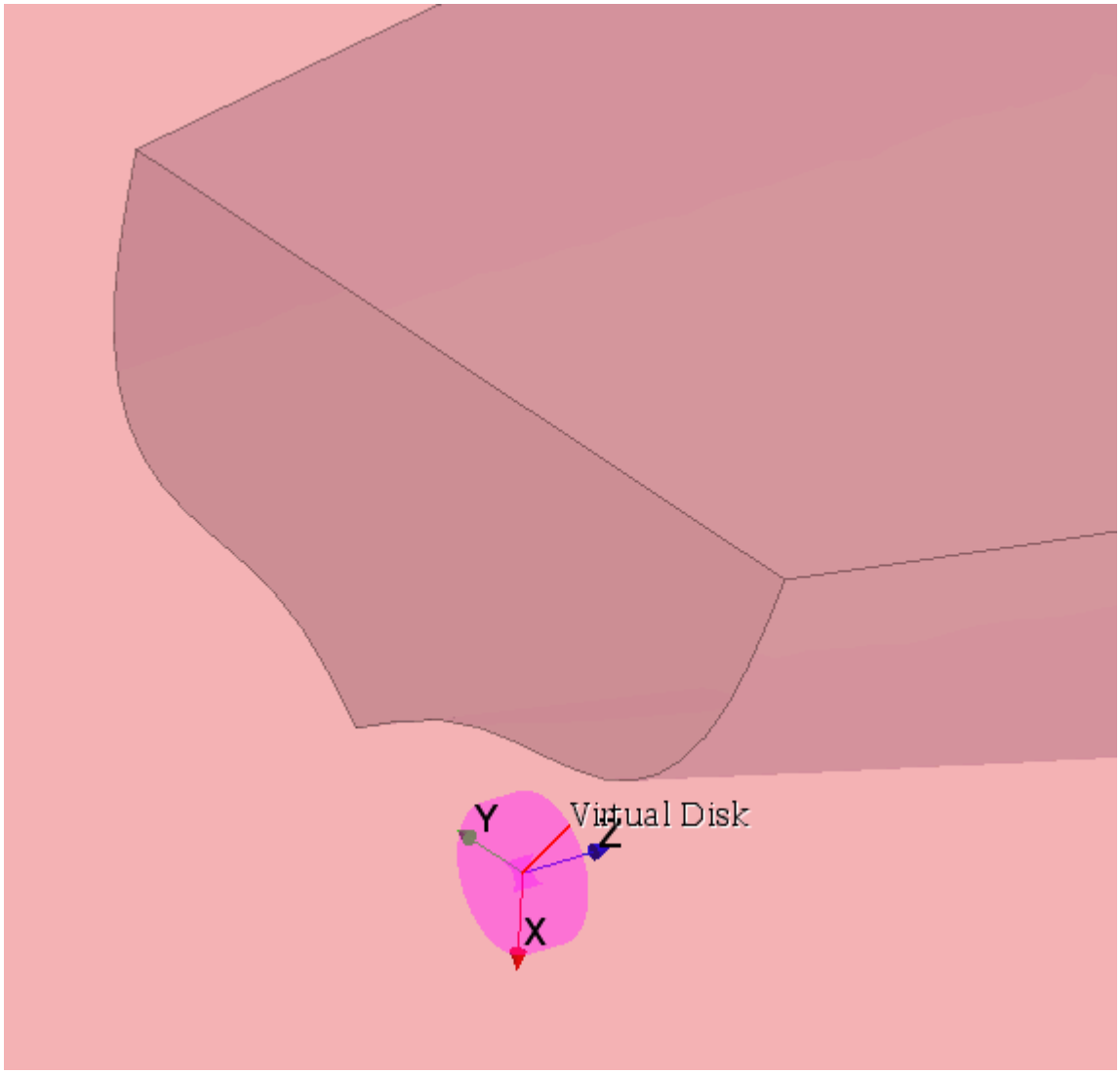
Node	Property	Setting
Propeller Curve	**Method**	**Table (J)**
└ **Table (J)**	**Table**	**BSeries_Z5_BAR1p05_PD0p723**
	Table: J	**J**
	Table: Kt	**KT**
	Table: Kq	**KQ**
	Table: Eta	**ETA**
Disk Geometry	**Inner Radius**	`0.036 m`
	Outer Radius	`0.18 m`
	Origin	`[-6.307, 0.0, -1.25] m,m,m`
└ **Normal And Coordinate System**	**Coordinate System**	**Laboratory -> Ship-CSys**
	Disk Normal	`[1.0, 0.0, 0.0]`
Inflow Specification	**Method**	**Inflow Velocity Plane**
└ **Inflow Velocity Plane**	**Inflow Plane Radius**	`0.198 m`
	Inflow Plane Offset	`0.018 m`
RotationRate (n)	**Rotation Rate**	`2300.0 rpm`

Note: You must associate the **Coordinate System** property that defines the direction of thrust with the local coordinate system of the ship body, here identified as **Ship-CSys**. This association establishes the connection between the thrust produced by the virtual disk and the DFBI motion of the ship. The ship coordinate system moves with the ship according to the fluid forces calculated by the DFBI model.

6. Expand the **Tools** > **Coordinate Systems** > **Laboratory** > **Local Coordinate Systems** and drag and drop the **Virtual Disk-CSys 1** node into the currently active *Geometry 1* scene.

   The virtual disk local coordinate system is displayed in the scene. The thrust produced by the virtual disk acts in the Z direction of this coordinate system.

7. To visualize the virtual disk, select the **Virtual Disk** node within **Physics Continuum 1** and zoom in on the virtual disk.



8. Save the simulation.

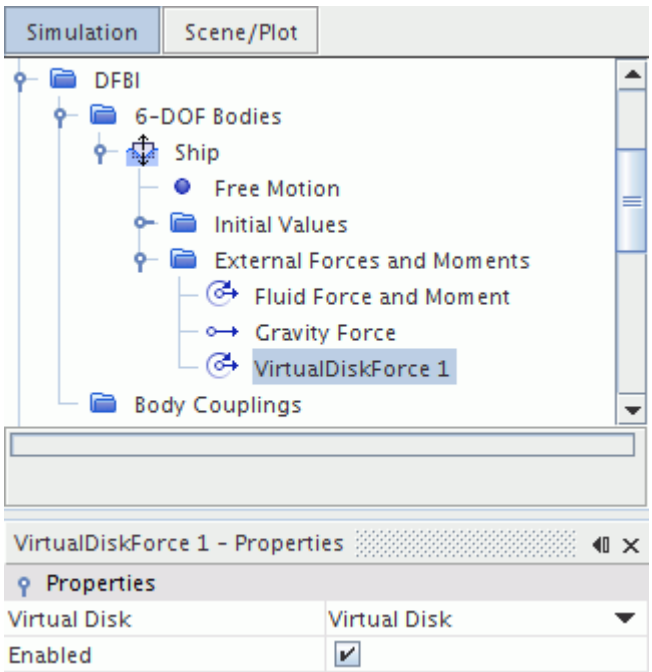
Applying the Thrust Force to the Ship

To apply the thrust force produced by the virtual disk to the ship, you must create an external force for the 6-DOF ship body.

To apply the virtual disk thrust to the ship:

1. Right-click the **DFBI** > **6-DOF Bodies** > **Ship** > **External Forces and Moments** node and select **New** > **Virtual Disk Force**.
2. Select the **External Forces and Moments** > **VirtualDiskForce 1** node and set **Virtual Disk** to **Virtual Disk**.

   The force produced by **Virtual Disk** is now associated with the **Ship** body.



3. Save the simulation.

Creating Mesh Refinement for the Virtual Disk

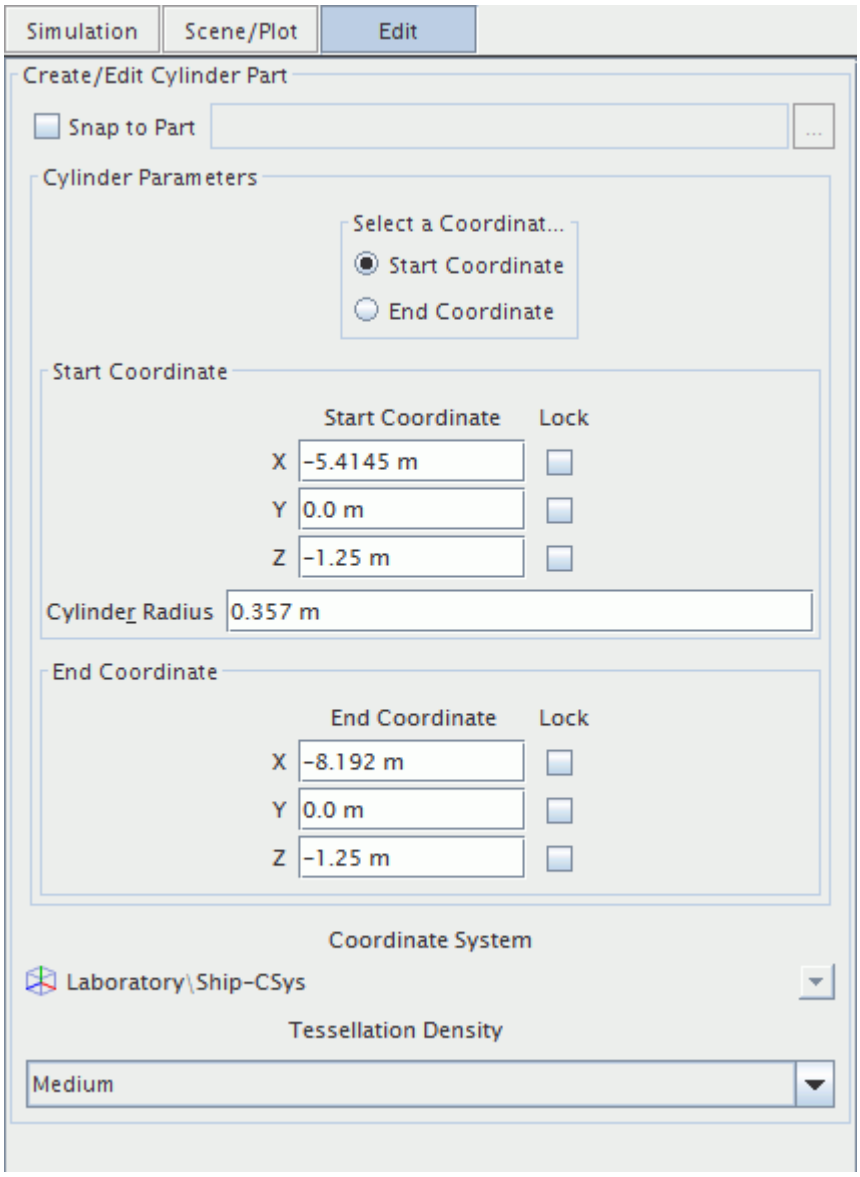
To obtain a good resolution of the flow field around the virtual disk, you are advised to create a cylindrical volumetric refinement. You create the refinement using part shapes and mesh custom controls.

To create the mesh refinement for the virtual disk:

1. Right-click the **Geometry** > **Parts** node and select **New Shape Part** > **Cylinder**.
2. In the *Create/Edit Cylinder Part* dialog set the following parameters:

	Start Coordinate	End Coordinate	Cylinder Radius	Coordinate System
X	-5.4145 m	-8.192 m	0.357 m	**Laboratory\Ship-CSys**

	Start Coordinate	End Coordinate	Cylinder Radius	Coordinate System
Y	`0.0 m`	`0.0 m`		
Z	`-1.25 m`	`-1.25`		



You create the cylindrical shape part with respect to the local body coordinate system of the ship.

3. Click **Create**, then **Close**.
4. Rename the **Parts** > **Cylinder** node as `Disk Refinement`.

Continue by assigning custom mesh controls to the **Disk Refinement** part shape:

5. Right-click the **Geometry** > **Operations** > **Automated Mesh** > **Custom Controls** node and select **New** > **Volumetric Control**.

6. Rename the **Volumetric Control** node as `Disk Refinement`.
7. Edit the **Disk Refinement** node and set the following properties:

Node	Property	Setting
Disk Refinement	**Parts**	**Disk Refinement**
└ **Controls** > **Trimmer**	**Customize Isotropic Size**	Activated
└ **Values** > **Custom Size**	**Size Type**	**Relative to base**
└ **Relative Size**	**Percentage of Base**	`10.0`

8. Save the simulation.

Generating the Volume Mesh

Suitable for this type of application, you generate a trimmed hexahedral mesh with local refinements and prism layers along the walls of the wetted surface.

In the initial simulation file, the required mesh models and corresponding settings are already set up. The wake pattern of the boat is resolved through cone-shaped volumetric refinements.

To generate the volume mesh:

1. Select **Mesh** > **Generate Volume Mesh**

   The mesh generation process several seconds to complete.
2. Open **Mesh Scene 1**.
3. Right-click a blank area in the scene and select **Apply Representation** > **Volume Mesh**

   You may need to refresh the scene for the section plane to display correctly.

4. Save the simulation.

Defining the Water and Air Phases

Using the VOF multiphase model requires that you create Eulerian phases for water and air.

To define the water and air phases:

1. In the **Physics 1** continuum, right-click the **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and create a new phase.
2. Rename the **Phase 1** node to `Water`.
3. For the new **Water** phase, select the following models:

Group Box	Model
Material	**Liquid**

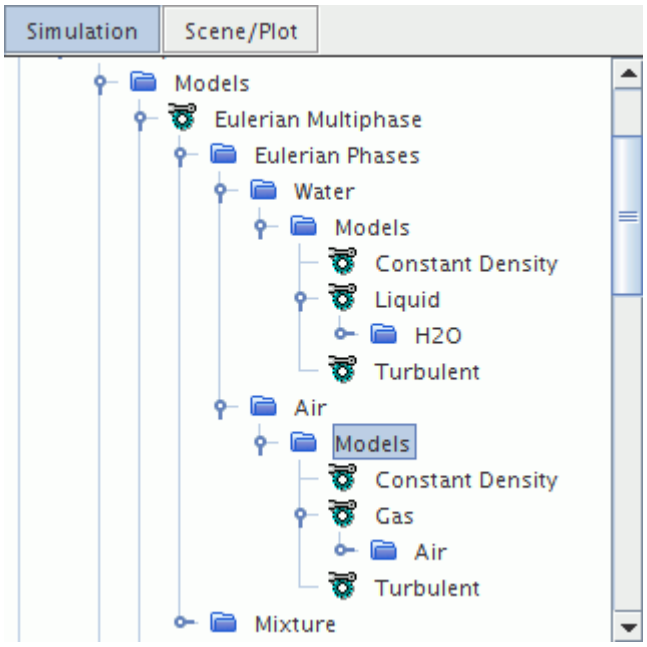
Group Box	Model
Equation of State	**Constant Density**

4. Create a second phase and rename it to `Air`.
5. For the **Air** phase, select the following models:

Group Box	Model
Material	**Gas**
Equation of State	**Constant Density**

You leave the material properties for water and air at their default values.

The **Eulerian Multiphase** node contains the following models:



6. Save the simulation.

Defining the VOF Wave

The free surface water level changes over time during the simulation. STAR-CCM+ provides the VOF Waves model that provides wave initial and boundary conditions. Here, you define a flat VOF wave as the ship is moving through calm water.

To define a flat VOF Wave:

1. In the **Physics 1** continuum, right-click the **VOF Waves** > **Waves** node and select **New** > **Flat**.
2. Select the **Flat Vof Wave 1** node and set the following properties:

Property	Setting
Current	`[-4.57, 0.0, 0.0] m/s`

Property	Setting
Wind	`[-4.57, 0.0, 0.0] m/s`

The inlet velocity that you specify here results from the procedure for obtaining the inlet velocity to balance the forces in X direction.

3. Save the simulation.

Setting Initial Conditions

You set the initial conditions for pressure, velocity, and volume fraction.

To set the initial conditions:

1. In the **Physics 1** continuum, edit the **Initial Conditions** node and set the following properties:

Node	Property	Setting
Pressure	**Method**	**Field Function**
	Scalar Function	**Hydrostatic Pressure of Flat Vof Wave 1**
Velocity	**Method**	**Field Function**
	Vector Function	**Velocity of Flat Vof Wave 1**
Volume Fraction	**Method**	**Composite**
└ **Composite** > **Water**	**Method**	**Field Function**
	Scalar Function	**Volume Fraction of Heavy Fluid of Flat Vof Wave 1**
└ **Composite** > **Air**	**Method**	**Field Function**
	Scalar Function	**Volume Fraction of Light Fluid of Flat Vof Wave 1**

2. Save the simulation.

Damping Wave Reflections

Unless appropriate measures are applied, waves can be reflected from the surrounding boundaries due to the nature of the numerical methods employed. These non-physical wave reflections must be avoided as they interact with the true wave field and can lead to invalid results.

STAR-CCM+ provides the VOF Wave damping model to minimize the risk of wave reflection.

To activate VOF Wave damping at boundaries:

1. Edit the **Regions** > **Domain** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **VOF Wave Zone Option**	**VofWave Zone Option**	**Damping**
Physics Values		
└ **VOF Wave Damping Length**	**Method**	**Constant**
	Value	`15.0 m`

2. Save the simulation.

VOF wave damping using the specified damping length is later activated for each boundary.

Setting Up Boundary Conditions

To specify wave damping, velocities, and variable water surface level at the surrounding boundaries of the ship, you apply field functions from the flat VOF wave.

To set up the boundary conditions:

1. Edit the **Regions** > **Domain** > **Boundaries** > **Aft** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **VOF Wave Damping Boundary Option**	**Method**	**Yes**
Physics Values		
└ **Pressure**	**Method**	**Field Function**
	Scalar Function	**Hydrostatic Pressure of Flat Vof Wave 1**
└ **Volume Fraction**	**Method**	**Composite**
└ **Composite** > **Water**	**Method**	**Field Function**
	Scalar Function	**Volume Fraction of Heavy Fluid of Flat Vof Wave 1**
└ **Composite** > **Air**	**Method**	**Field Function**
	Scalar Function	**Volume Fraction of Light Fluid of Flat Vof Wave 1**

2. Multi-edit the **Forward**, **Port**, and **Starboard** nodes and set the following properties:

Node	Property	Setting
Physics Conditions		

Node	Property	Setting
└ **Velocity Specification**	**Method**	**Components**
└ **VOF Wave Damping Boundary Option**	**Method**	**yes**
Physics Values		
└ **Velocity**	**Method**	**Field Function**
	Vector Function	**Velocity of Flat Vof Wave 1**
└ **Volume Fraction**	**Method**	**Composite**
└ **Composite** > **Water**	**Method**	**Field Function**
	Scalar Function	**Volume Fraction of Heavy Fluid of Flat Vof Wave 1**
└ **Composite** > **Air**	**Method**	**Field Function**
	Scalar Function	**Volume Fraction of Light Fluid of Flat Vof Wave 1**

3. Multi-edit the **Bottom** and **Top** nodes and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Velocity Specification**	**Method**	**Components**
Physics Values		
└ **Velocity Method**		**Field Function**
	Vector Function	**Velocity of Flat Vof Wave 1**
└ **Volume Fraction**	**Method**	**Composite**
└ **Composite** > **Water Method**		**Field Function**
	Scalar Function	**Volume Fraction of Heavy Fluid of Flat Vof Wave 1**
└ **Composite** > **Air Method**		**Field Function**
	Scalar Function	**Volume Fraction of Light Fluid of Flat Vof Wave 1**

Leave all wall boundaries at their default settings.

4. Save the simulation.

Setting Solver Parameters and Stopping Criteria

This self-propulsion simulation is a transient analysis for which you must specify a set of numerical parameters. The choice of parameters has a strong influence on the performance of STAR-CCM+ for these type of analyses..

To set the solver parameters and stopping criteria:

1. Edit the **Solvers** node and set the following parameter:

Node	Property	Setting
Implicit Unsteady	*Time-Step*	`0.02 s`
6-DOF Solver	*Maximum Number of Iterations*	`7`

2. Edit the **Stopping Criteria** node and set the following parameter:

Node	Property	Setting
Maximum Inner Iterations	*Maximum Inner Iterations*	`10`
Maximum Physical Time	*Maximum Physical Time*	`80.0 s`
Maximum Steps	*Enabled*	**Deactivated**

3. Save the simulation.

Monitoring and Plotting the Net Force in X Direction

You create several reports for the different forces such as drag, trim, and the virtual disk force components. The final expression report for the net force in X direction uses these reports.

To analyze the net force in the X direction acting on the ship hull:

1. Create a report for the total drag over time:
   a) Right-click the **Reports** node and select **New Report** > **Force**.
   b) Rename the **Force 1** node to `Drag`.
   c) Select the **Drag** report and set the following properties:

Property	Setting
Direction	`[-1.0, 0.0, 0.0]`
Parts	**(Regions)[Domain:Deck, Domain:Hull, Domain:Stern]**

2. Create a report for the trim:
   a) Right-click the **Reports** node and select **New Report** > **6-DOF Body Orientation**.
   b) Rename this report to `Trim Down.`
   c) Select the **Trim Down** report and set *Angle Report Option* to **Rotation Y Axis**.
3. Create a report for the thrust created by the virtual disk:
   a) Right-click the **Reports** node and select **New Report** > **Virtual Disk Force**.
   b) Rename this report to `Longitudinal Thrust.`

c) Select the **Longitudinal Thrust** report and set the following properties:

Property	Setting
Virtual Disk	**Virtual Disk**
Force Component Option	**Z-Axis Force**

4. Create a report for the X component of the thrust created by the virtual disk:
   a) Right-click the **Reports** node and select **New Report** > **Expression**.
   b) Rename this report to `X Component of Thrust`.
   c) Select the **X Component of Thrust** report and set *Definition* to `${LongitudinalThrustReport}*cos(abs($TrimDownReport))`.
5. Create a report for the net force in the X direction:
   a) Right-click the **Reports** node and select **New Report** > **Expression**.
   b) Rename this report to `X Component of Net Force`.
   c) Select the **X Component of Net Force** report and set *Definition* to `${XComponentofThrustReport} - $DragReport`.
6. Create monitors and plots from all reports:
   a) Multi-select all reports that you just created.
   b) Right-click on any one of the high-lighted items and select **Create Monitor and Plot from Report**.
   c) In the *Create Plot From Reports...* dialog, select **Multiple Plots (one per report)**.
7. Right-click the **Plots** > **X Component of Net Force Monitor Plot** node and select **Open**.
8. Save the simulation.

If you want to obtain an averaged value of the net force in X direction, right-click the **X Component of Net Force Monitor Plot** node and **Export...** the force values over time to a file in .csv format. You can then import this file into a spreadsheet software of your choice and average the force values. You can use this value to obtain the balancing inlet velocity.

Visualizing the Pressure Distribution

You can obtain a good visualization of the pressure distribution around the ship hull by subtracting the hydrostatic pressure component.

To visualize the pressure distribution:

1. Create a custom field function for pressure:
   a) Right-click the **Tools** > **Field Functions** node and create a new scalar field function.
   b) Rename this field function to `Pdynamic`.
   c) Select the **Pdynamic** node and set the following properties:

Property	Setting
Function Name	`Pdynamic`
Definition	`$Pressure + $Density * 9.81 * $ $Centroid[2]`

2. Create a scalar scene.
3. Edit the **Scenes** > **Scalar Scene 1** > **Displayers** node and set the following properties:

Node	Property	Setting
Scalar 1 > **Parts**	*Parts*	**(Regions) [Domain: Deck, Domain: Hull, Domain: Stern, (Derived Parts) Centerplane**
Scalar Field	*Function*	**Pdynamic**
Scalar 1	*Contour Style*	**Smooth Filled**

4. Select the **Scalar Scene 1** > **Attributes** > **Update** node and set *Trigger* to **Time Step**.
5. Save the simulation.

Running the Simulation

You are advised to run this tutorial on a multi-core machine with four cores or more. Depending on the specifications of your computer, this tutorial takes several hours or up to a day to complete.

To run the simulation:

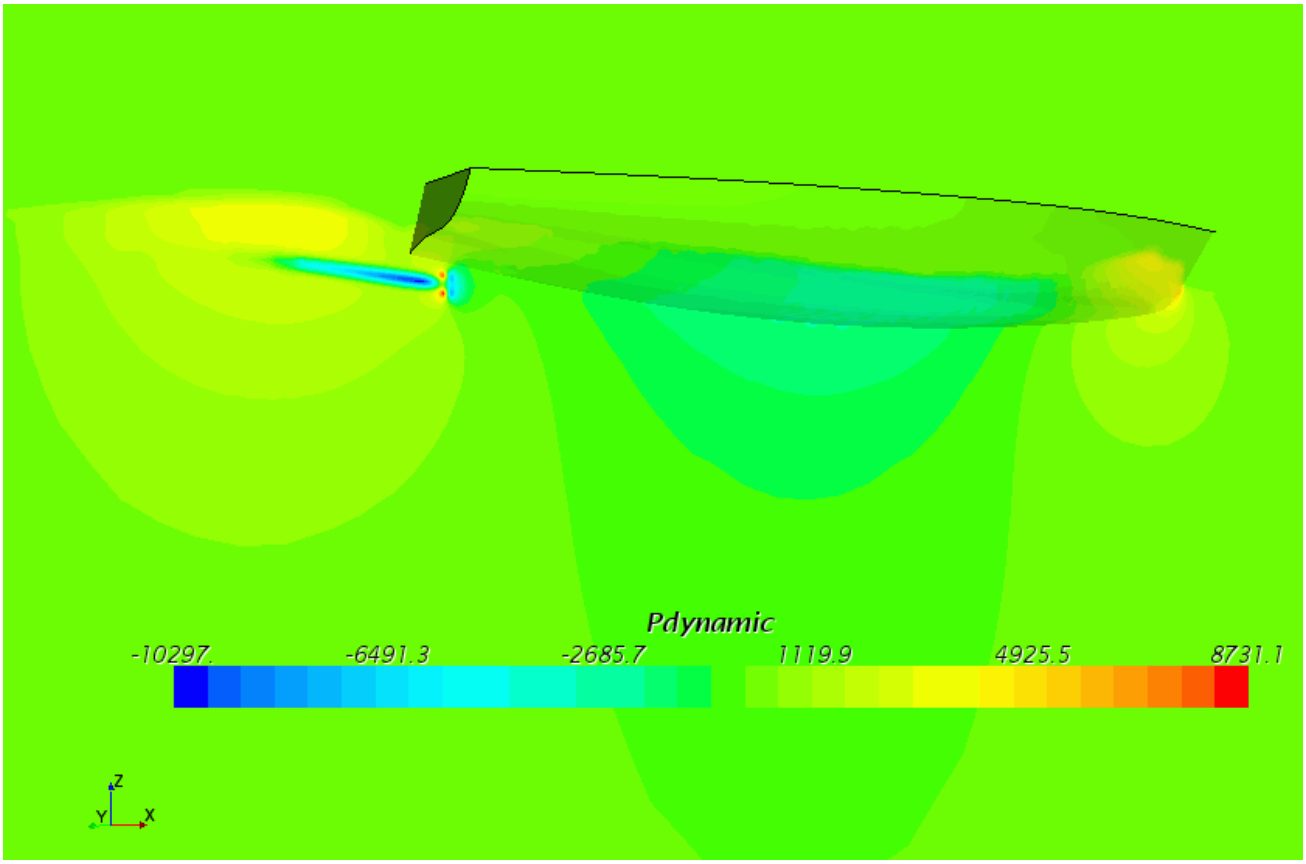
1. Click the (**Run**) button on the STAR-CCM+ toolbar.

   While the simulation is running, you can click the tabs at the top of the *Graphics* window to view each of the scenes. The *Residuals* display is created automatically and shows the progress of the solvers.

Visualizing the Results

You can follow the progress of the solution using each of the scenes that you created previously.

The following screenshot shows a close-up of the pressure distribution without the hydrostatic pressure component:



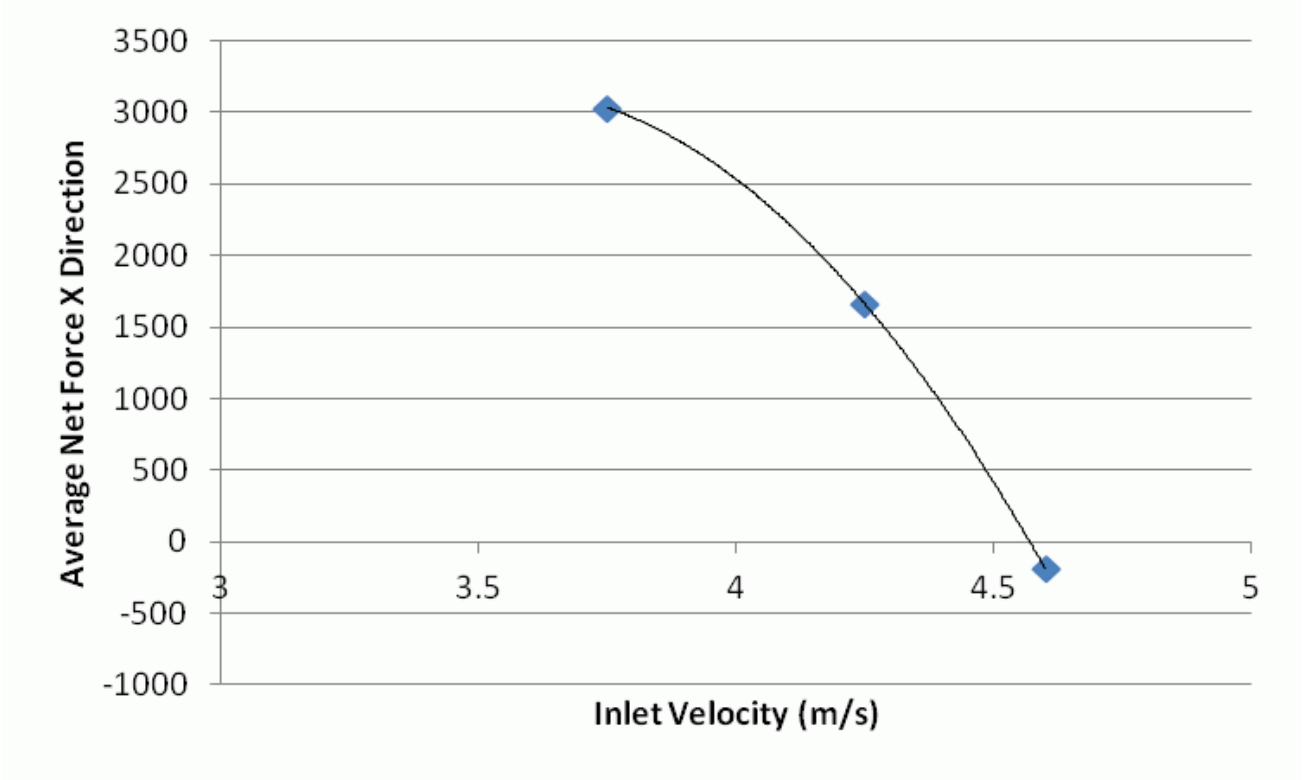
Obtaining the Balancing Inlet Velocity

You specify the inlet velocity such that the forces acting on the ship hull are balanced in the forward direction.

In this simulation, the propeller rotates at a fixed rate. This rotation rate produces a certain thrust and the ship moves forward. The movement of the ship through water induces a resistance due to the viscous and pressure forces acting on the ship hull—the drag force.

For this simulation setup, the ship is not moving freely in the forward (X) direction, but remains in a fixed position. Instead, you prescribe an inlet velocity that produces a relative motion of the ship.

The inlet velocity is not known in advance. You can determine it by running, for example, three simulations with three different inlet velocities. For each simulation, you calculate the average net force acting on the ship hull in the X direction. By plotting the net force against inlet velocity for each of the three simulations, you can find the velocity for which the net force becomes zero. An example is shown in the following plot:



For this simulation, the inlet velocity at which thrust and drag are balanced is 4.57 m/s. You set this value for the wave velocity in Defining the VOF Wave.

To set up the reports, monitors, and plots to obtain the averaged X net force, see Monitoring and Plotting the Net Force in X Direction.

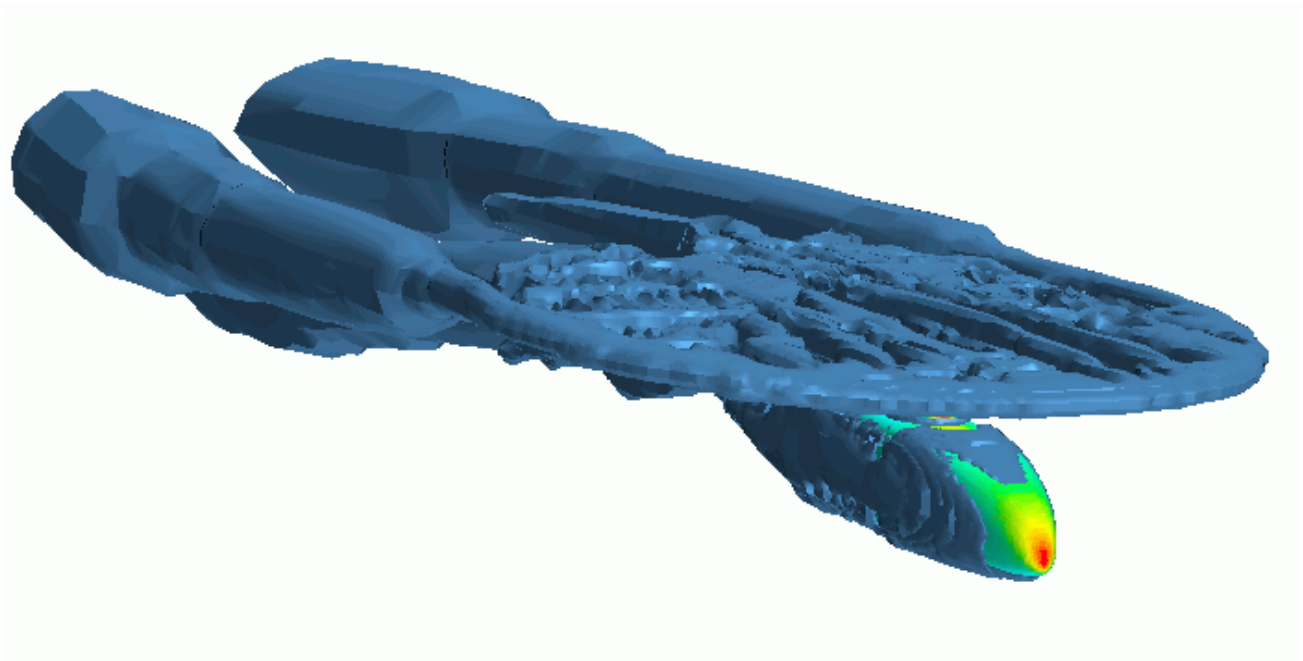
Summary

This tutorial has introduced the following STAR-CCM+ features:

- Setting up a marine self-propulsion simulation.
- Modeling a marine propeller by using the Virtual Disk model.
- Setting up the Virtual Disk model in conjunction with the DFBI model.
- Obtaining the unknown inlet velocity to balance the thrust and drag forces on the ship hull.
- Setting up a VOF multiphase simulation.
- Setting up a flat VOF Wave.
- Setting up boundary-wise wave damping.
- Setting up and running a transient simulation.

Blade Element Method: Helicopter Rotor-Fuselage Interaction

The rotating main rotor blades of a helicopter generate a complex flow field that leads to periodic aerodynamic loads on the airframe. Because of these aerodynamic loads, the passenger cabin can be subject to noise and vibration. These aerodynamic interactions affect the performance of helicopters and they can cause structural damage. STAR-CCM+ provides a method for simulating the flow generated by rotor blades that does not require you to mesh the blades individually. This method allows you to predict the complex flow field around a helicopter taking into account the fuselage and rotor flow field interaction.



This tutorial simulates the flight of a generic helicopter fuselage using a virtual wind tunnel in STAR-CCM+. The ROBIN (ROtor Body INteraction) configuration is used for the airframe where the geometry and operating condition are taken from [805]. The parameter values correspond to Table 17 in this paper.

The following list gives a summary of the operating condition used for this tutorial:

- Rotation rate: $\Omega = 2000\ \mathrm{rpm}$
- Advance ratio: $\mu = 0.151$
- Rotor thrust coefficient: $C_T/\sigma = T/\rho_\infty A v_{Tip}^2 \sigma = 0.0403$
- Rotor shaft angle (pitch angle): $\alpha_s = -3°$
- Collective pitch angle: $\theta = 7.7°$
- Lateral cyclic pitch angle: $A_1 = -1.8°$
- Longitudinal cyclic pitch angle: $B_1 = 2.3°$

where $\sigma = 0.098$ is the rotor solidity, $\rho_\infty = 1.176\, kg/m^3$ is the free-stream density, T is the rotor thrust, $A = \pi R^2$ is the rotor disk area, $R = 0.860552\, m$ and $v_{Tip} = \left(\Omega\pi/30\right) \cdot R$ is the tip speed due to rotation.

To model the helicopter flight, the ROBIN body is positioned within a virtual wind tunnel. In the tutorial, you apply the blade element method of the virtual disk model to simulate the effect of the rotor on the helicopter fuselage. The advantage of the blade element method is that it does not require you to explicitly resolve the rotor blades geometry with a fine mesh. For more information, see Blade Element method in the User Guide.

Contents:

Prerequisites
Loading the Initial Simulation File
Defining the Operating Condition Using Field Functions
Selecting the Physics Models
Modeling the Rotor by Using the Virtual Disk Model
Creating the Virtual Disk Mesh Refinement
Generating the Volume Mesh
Setting up Boundary Conditions
Setting Solver Parameters and Stopping Criteria
Monitoring and Plotting Virtual Disk Thrust
Visualizing the Velocity Field
Visualizing the Pressure on the Fuselage
Running the Simulation
Visualizing the Results
Summary
Bibliography

Prerequisites

The instructions in the blade element method tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Techniques	Associated Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Loading the Initial Simulation File

For this tutorial, you are provided with a simulation file that contains pre-defined objects and settings.

The pre-defined objects and settings include:

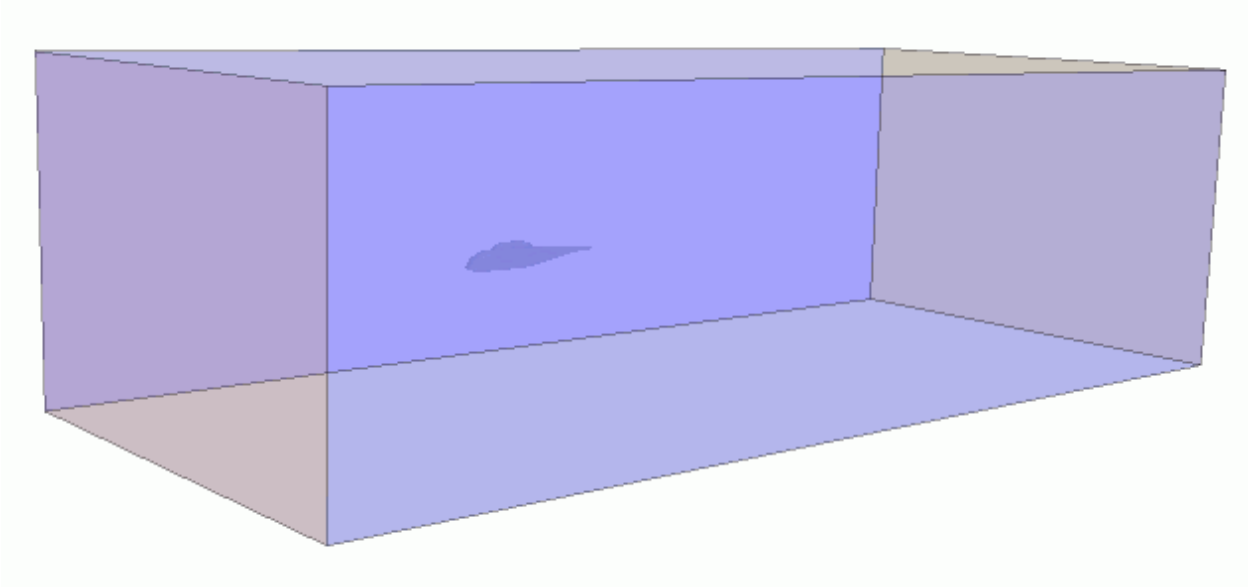
- geometry parts

- mesh operations
- boundary types

Due to the large number of cells, launch this tutorial in parallel on at least four cores.

To load the initial simulation:

1. Launch STAR-CCM+.
2. Select **File** > **Load...**
3. In the *Process Options* box, select an appropriate parallel option and set the number of **Compute Processes** to at least 4.
4. In the *Load a File* dialog, click **Browse...**
5. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/motion/data`.
6. Select `genericHelicopter_start.sim`, then click **Open**.
7. In the *Load a File* dialog, click **OK**.
8. To visualize the imported geometry, open **Geometry Scene 1**.



9. Save the simulation as `genericHelicopter_run.sim`.

Defining the Operating Condition Using Field Functions

Several variables define the operating condition that applies to the helicopter in the virtual wind tunnel. Defining these variables as field functions allows you to change the operating condition for further analyses.

For this tutorial, the operating condition is defined by the following parameters:

- Advance ratio

  The advance ratio determines the forward speed of the helicopter. This parameter is used to compute the inlet velocity at the inflow boundary.
- Thrust coefficient

The thrust coefficient is used to compute the target thrust of the rotor.

- Rotation rate

  The rotation rate is used to compute the rotational speed and the rotor tip velocity.

- Pitch angles

  You prescribe the various pitch angles later directly for the virtual disk properties.

To define the operating condition:

1. Create a field function for the advance ratio:
   a) Right-click the **Tools** > **Field Functions** node and select **New** > **Scalar**.
   b) Rename the **User Field Function 1** node to `Advance Ratio`.
   c) Select the **Advance Ratio** node and set the following properties:

Property	Setting
Function Name	`AdvanceRatio`
Definition	`0.151`

2. In the same way, create 11 more scalar field functions and name them `Disc Area`, `Inlet Velocity`, `Omega`, `Pi`, `Rotor Radius`, `Rotor Solidity`, `RPM`, `Thrust Coefficient`, `Thrust`, `Rho`, and `Tip Speed`, respectively.
3. Set the following properties for the field functions:

Field Function	Function Name	Dimensions	Definition
Rotor Radius	`RotorRadius`	**Length**	`0.860552`
Pi	`Pi`	Dimensionless	`atan(1)*4`
Disc Area	`DiscArea`	**Length^2**	`$RotorRadius* $RotorRadius*$Pi`
Rotor Solidity	`RotorSolidity`	Dimensionless	`0.098`
RPM	`RPM`	**1/Time**	`2000`
Omega	`Omega`	**AngularVelocity**	`$Pi*$RPM/30`
Tip Speed	`TipSpeed`	**Velocity**	`$Omega* $RotorRadius`
Inlet Velocity	`InletVelocity`	**Velocity**	`$AdvanceRatio* $TipSpeed`
Thrust Coefficient	`ThrustCoefficient`	Dimensionless	`0.0403`
Rho	`Rho`	**Mass/Volume**	`1.176`
Thrust	`Thrust`	**Force**	`$ThrustCoefficient*$Rho*$DiscArea* $TipSpeed*`

Field Function	Function Name	Dimensions	Definition
			`$TipSpeed* $RotorSolidity`

4. Save the simulation.

Selecting the Physics Models

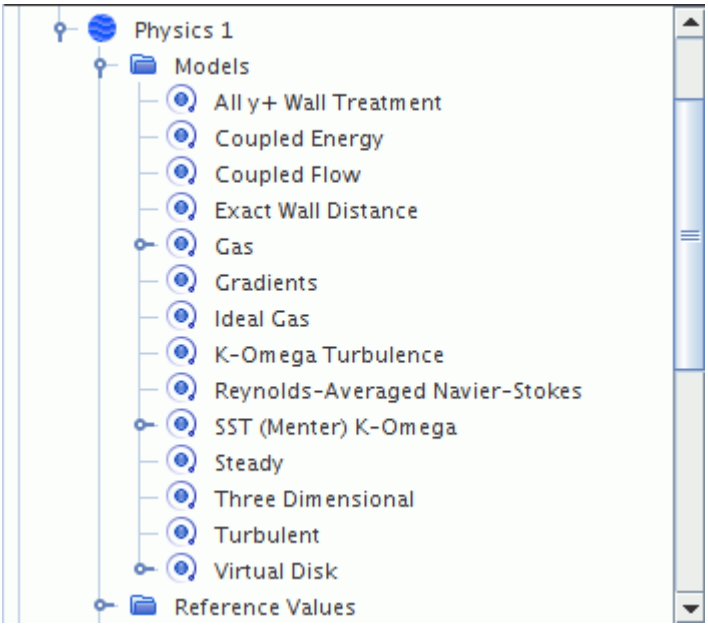
You compute a steady compressible flow of air. To model the turbulent flow, you use the $k - \omega$ turbulence model. For modeling the rotor, you select the virtual disk model.

To select the physics models:

1. For the physics continuum, **Continua** > **Physics 1**, select the following models in order:

Group Box	Model
Time	**Steady**
Material	**Gas**
Flow	**Coupled Flow**
	Gradients (selected automatically)
Equation of State	**Ideal Gas**
	Coupled Energy (selected automatically)
Viscous Regime	**Turbulent**
	Reynolds-Averaged Navier-Stokes (selected automatically)
Reynolds-Averaged Turbulence	**K-Omega Turbulence**
	SST (Menter) K-Omega (selected automatically)
	Exact Wall Distance (selected automatically)
K-Omega Wall Treatment	**All y+ Wall Treatment** (selected automatically)
Optional Models	**Virtual Disk**

2. To review the models, open the **Physics 1** > **Models** node.



3. Save the simulation.

Modeling the Rotor by Using the Virtual Disk Model

For this simulation, you model the effects of the rotor on the helicopter fuselage using the blade element method. This method belongs to the virtual disk model.

To model the rotor:

1. Right-click the **Continua** > **Physics 1** > **Models** > **Virtual Disk** > **Virtual Disks** node and select **New**.
2. Select the **Virtual Disks** > **Virtual Disk** node and set *Method* to **Blade Element Method**.

Before you can define the airfoil sections and the twist distribution, you must import two table data files (here in .csv format) into STAR-CCM+.

Note: STAR-CCM+ internally processes the provided angle data for the airfoil sections and the twist distribution in SI units, that is, in radians. In this context, it is important that you specify the tabular angle data correctly with respect to units.

First, you verify the preferred unit settings for angle data:

3. Select the **Tools** > **Units** > **radian** node and make sure that *Preferred* is activated.

   Radian is the preferred unit for angle data for this simulation.

Next, you verify the unit specification in your table data for twist distribution and the airfoil sections.

4. Using a text editor of you choice, open `[INSTALL_DIR]/doc/startutorialsdata/motion/data/NACA_0012.csv` and `[INSTALL_DIR]/doc/startutorialsdata/motion/data/twistTable.csv`.

   You can see that the angles in `twistTable.csv` are specified in radians. This specification corresponds to the preferred units settings and no further changes are required. In `NACA_0012.csv`, you can see that the angles are specified in degrees. In order to let STAR-CCM+ know that this angle

data is specified in degrees and must be converted to radians, a unit specification is added to the column header of the table data file:

```
"AoA (deg)","Cl","Cd","Mach"
-16,0,0,0
-16,-1.218,0.09183,0.2
-16,-1.316,0.08124,0.4
-16,-1.521,0.07389,0.6
...
```

5. Right-click **Tools** > **Tables** and select **New Table** > **File Table**.
6. Navigate to `[INSTALL_DIR]/doc/startutorialsdata/motion/data/NACA_0012.csv` and click **Open**.
7. Create another file table and import `twistTable.csv` from the same directory.

You can now proceed to set up the virtual disk.

8. Right-click the **Virtual Disk** > **Airfoil Sections** node and select **New**.
9. Edit the **Virtual Disk** node and set the following properties:

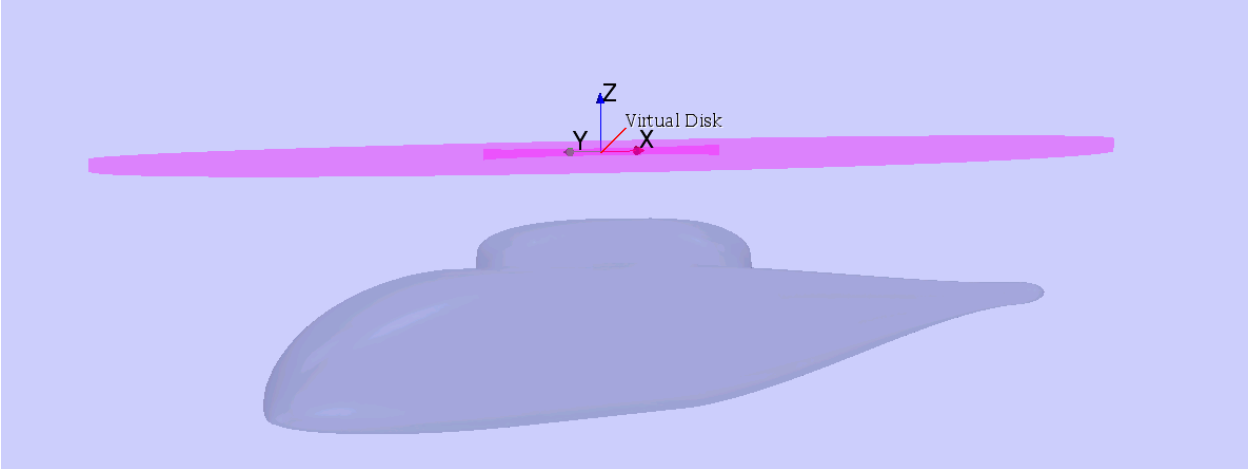
Node	Property	Setting
Airfoil Sections > **Airfoil Section 1**		
└ **Cl (AoA, Mach)**	*Input Table*	**NACA_0012**
	Table: Mach Number	**Mach**
	Table: Angle of Attack	**AoA**
	Table: Cl	**Cl**
└ **Cd (AoA, Mach)**	*Input Table*	**NACA_0012**
	Table: Mach Number	**Mach**
	Table: Angle of Attack	**AoA**
	Table: Cd	**Cd**
Chord Distribution > **Constant**	*Value*	`0.066294 m`
Twist Distibution	*Method*	**Table (r/R)**
└ **Table (r/R)** > **Table (r/R)**	*Input Table*	**twistTable**
	Table: Data	**twist**
	Table: Normalized Span	**r/R**
Disk Geometry	*Number of Blades*	`4`
	Inner Radius	`0.207 m`
	Outer Radius	`0.860552 m`
	Thickness	`0.02 m`

Node	Property	Setting
	Origin	`[0.696, 0.051, 0.322] m, m, m`
	Orientation Specification	**Pitch Angle and Lateral Angle**
└ **Lateral and Pitch Angle**	*Pitch Angle*	`-3 deg`
Rotation Rate	*Rotation Rate*	`$Omega`
	Ramp Method	**Linear Ramp**
└ **Linear Ramp**	*Start Iteration*	`1`
	End Iteration	`100`
	Initial Value	`1.0`
Disk Stick Specification	*Collective Pitch*	`7.7 deg`
	Cyclic Pitch: Cosine Component	`-1.8 deg`
	Cyclic Pitch: Sine Component	`2.3 deg`
	Virtual Disk Trim Option	**Trim Thrust Only**
Virtual Disk Resolution	*Radial Resolution*	`30`
	Azimuthal Resolution	`30`
Target Thrust	*Target Rotor Thrust*	`$Thrust`

Based on the *Origin* and the *Pitch Angle* that you provided, STAR-CCM+ creates a local virtual disk coordinate system. The thrust produced by the virtual disk acts in the Z direction of this coordinate system.

You can visualize the local virtual disk coordinate system and the virtual disk.

10. Expand the **Tools** > **Coordinate Systems** > **Laboratory** > **Local Coordinate Systems** and drag and drop the **Virtual Disk-CSys 1** node into the currently active *Geometry scene 1*.
11. Select the **Virtual Disk** node within **Physics 1** continuum and zoom in on the virtual disk.



You can see the dimensions, location, and the orientation of the virtual disk. The orientation determines the direction of the thrust that is produced by the virtual disk. This thrust acts in the Z direction of the local virtual disk coordinate system.

12. Save the simulation.

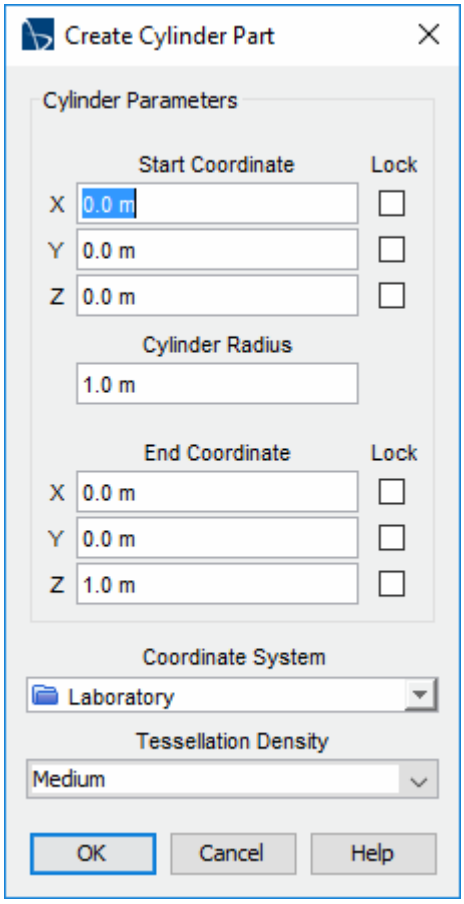
Creating the Virtual Disk Mesh Refinement

To obtain a good resolution of the flow field around the virtual disk, you are advised to create a cylindrical volumetric refinement. You create the refinement using part shapes and mesh custom controls.

To create the mesh refinement for the virtual disk:

1. Right-click the **Geometry** > **Parts** node and select **New Shape Part** > **Cylinder**.
2. In the *Create Cylinder Part* dialog, set the following parameters:

	Start Coordinate	End Coordinate	Cylinder Radius	Coordinate System
X	0.0	0.0	1.0 m	**Laboratory\Virtual Disk-CSys 1**
Y	0.0	0.0		
Z	-0.09 m	0.08		



You create the cylindrical shape part with respect to the local coordinate system of the virtual disk. By using this coordinate system, you automatically tilt the virtual disk refinement according to the pitch angle of the virtual disk.

3. Click **Create**, then **Close**.
4. Rename the **Parts** > **Cylinder** node to `Disk Refinement`.

Continue by assigning custom mesh controls to the Disk Refinement part shape:

5. Right-click the **Geometry** > **Operations** > **Automated Mesh** > **Custom Controls** node and select **New** > **Volumetric Control**.
6. Rename the **Volumetric Control** node as `Disk Refinement`.
7. Edit the **Disk Refinement** node and set the following properties:

Node	Property	Setting
Disk Refinement	*Parts*	**Disk Refinement**
└ **Controls** > **Trimmer**	*Customize Isotropic Size*	**Activated**
└ **Values** > **Custom Size**	*Relative to Base*	`10.0`

8. Save the simulation.

Generating the Volume Mesh

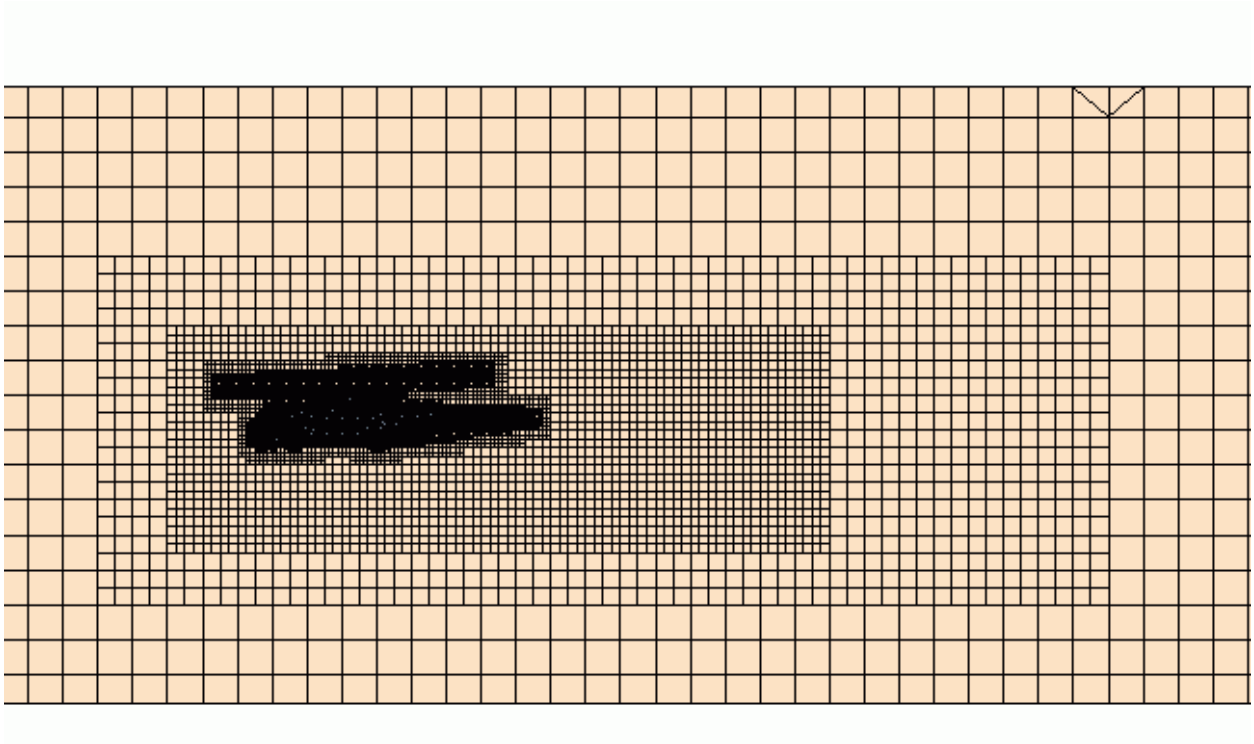
You can proceed to generate the volume mesh. For aerodynamics simulations, you typically use a trimmed cell mesh with refinements near the body and with embedded refinements for the near-field, the far-field, and the wake of the body.

The mesh you use here represents the recommended structure. However, to reduce the run-time of this tutorial, the mesh is coarser than for an industrial simulation.

To generate the volume mesh:

1. Select **Mesh** > **Generate Volume Mesh**.

   The mesh generation process takes several seconds to complete.
2. Open **Mesh Scene 1**.
3. Right-click a blank area in the scene and select **Apply Representation** > **Volume Mesh**.



4. Save the simulation.

Setting up Boundary Conditions

The boundary types are already set up in the starting simulation file that you imported at the beginning of the simulation. Here, you associate the inflow boundary condition with a corresponding field function.

By specifying an inlet velocity at the boundary in front of the helicopter, you produce a flow relative to the stationary helicopter body.

The inlet velocity is calculated from:

$$v_{Inlet} = \mu \cdot v_{Tip}$$

where $\mu = v_{Inlet}/v_{Tip}$ is the advance ratio and v_{Tip} is the tip velocity of the rotor.

1. Edit the **Regions** > **Wind Tunnel** > **Boundaries** > **Inlet** > **Physics Values** node and set the following properties:

Node	Property	Setting
Velocity Magnitude	*Method*	**Field Functions**
	Scalar Function	**Inlet Velocity**

The virtual wind tunnel must not impose floor or ceiling effects on the flow surrounding the fuselage. To model this, you specify a slip condition at these boundaries.

2. Multi-select the **Ceiling** and **Floor** nodes, right-click on any selection and select **Edit...**.
3. Select the **Physics Conditions** > **Shear Stress Specification** node and set *Method* to **Slip**.
4. Click **Close** and save the simulation.

Setting Solver Parameters and Stopping Criteria

The simulation is performed in steady-state using the coupled solver. You use the default numerical parameters for the coupled solver. The simulation stops after 1000 iterations.

To set stopping criteria:

1. Edit the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to `1000`.
2. Save the simulation.

Monitoring and Plotting Virtual Disk Thrust

The blade element method provides an algorithm that adjusts the initially specified collective pitch angle in order to reach the specified target thrust. This process is called trimming. Here, thrust trimming is activated in which only the collective pitch angle is adjusted. To follow this trimming process, you can plot the virtual disk thrust and the collective pitch angle as functions of iteration.

To monitor and plot the virtual disk thrust:

1. Right-click the **Reports** node and select **New Report** > **Virtual Disk Force**.
2. Rename the **Virtual Disk Force 1** node to `Virtual Disk Thrust`.
3. Edit the **Virtual Disk Thrust** node and set the following properties:

Property	Setting
Virtual Disk	**Virtual Disk**
Force Component Option	**Z-Axis Force**

4. Right-click the **Virtual Disk Thrust** node and select **Create Monitor and Plot from Report**.
5. Right-click the **Reports** node and select **New Report** > **Virtual Disk Angle**.
6. Rename the **Virtual Disk Angle 1** node to `Collective Pitch Angle`.

7. Edit the **Collective Pitch Angle** node and set the following properties:

Property	Setting
Units	**deg**
Virtual Disk	**Virtual Disk**
Angle Option	**Collective Pitch Angle**

8. Right-click the **Collective Pitch Angle** node and select **Create Monitor and Plot from Report**.
9. Save the simulation.

Visualizing the Velocity Field

You create a velocity contour plot on a section at the centerline of the helicopter.

To visualize the velocity field:

1. Right-click the **Scenes** node and select **New Scene** > **Scalar**.
2. Rename the **Scalar Scene 1** node to `Velocity Magnitude`.
3. Edit the **Velocity Magnitude** > **Displayers** node and set the following properties:

Node	Property	Setting
Scalar 1	*Contour Style*	**Smooth Filled**
└ **Parts**	*Parts*	**Derived Parts** > **Plane Section**
└ **Scalar Field**	*Function*	**Velocity** > **Lab Reference Frame** > **Magnitude**

4. Save the simulation.

Visualizing the Pressure on the Fuselage

To analyze the influence of the rotor on the helicopter fuselage, you plot the pressure contours on the helicopter surface.

To visualize the pressure on the fuselage:

1. Right-click the **Scenes** node and select **New Scene** > **Scalar**.
2. Rename the **Scalar Scene 1** node to `Pressure`.
3. Edit the **Pressure** > **Displayers** node and set the following properties:

Node	Property	Setting
Outline 1	*Part Visibility*	**Hide All Parts**
Scalar 1	*Contour Style*	**Smooth Filled**
└ **Parts**	*Parts*	**Regions** > **Wind Tunnel** > **Boundaries** > **ROBIN**

Node	Property	Setting
└ **Scalar Field**	*Function*	**Pressure**

4. Save the simulation.

Running the Simulation

You are advised to run this tutorial on a multi-core machine with four cores or more.

To run the simulation:

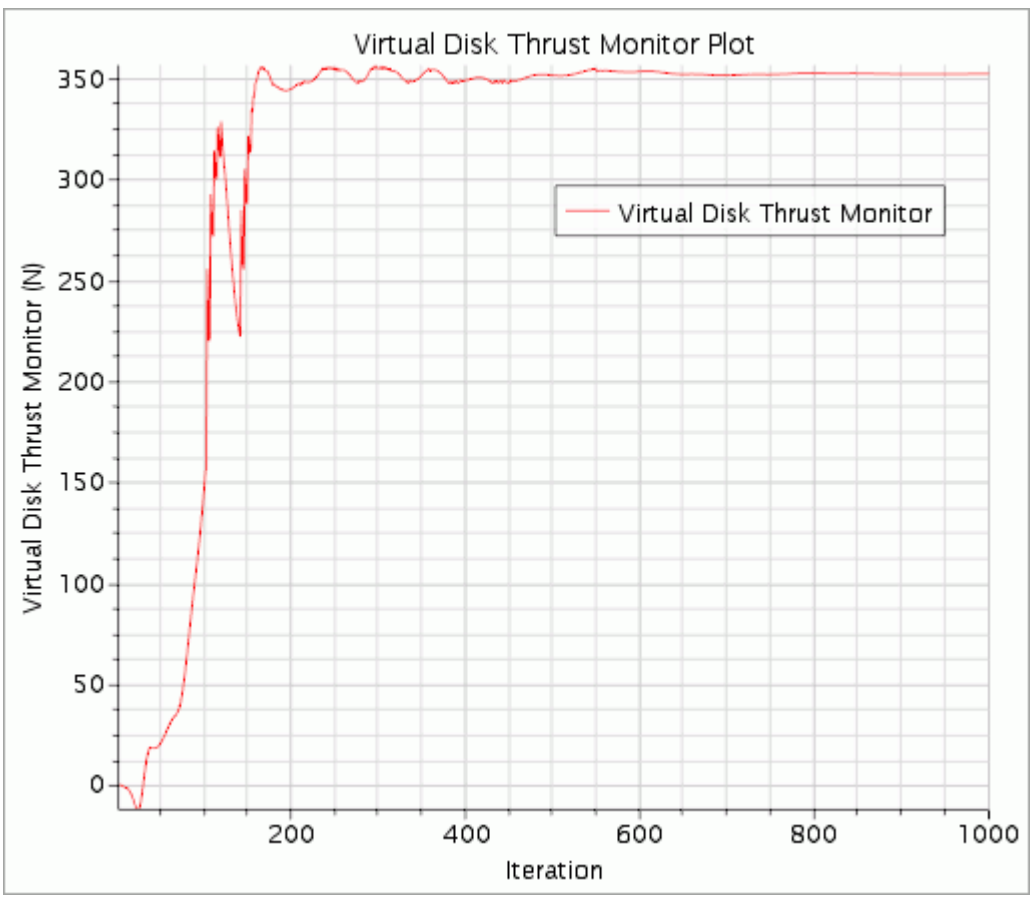
1. Click (**Run**) on the STAR-CCM+ toolbar.

   While the simulation is running, you can click the tabs at the top of the *Graphics* window to view each of the scenes. The *Residuals* display is created automatically and shows the progress of the solvers.

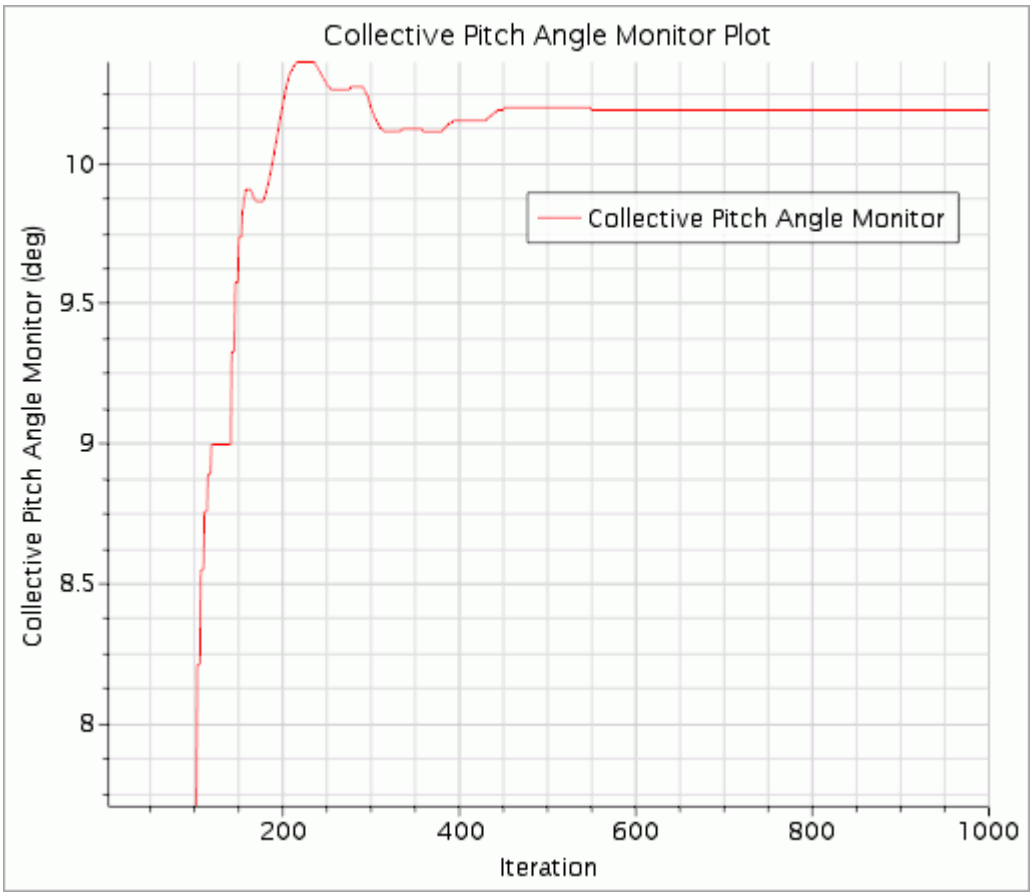
Visualizing the Results

You can follow the progress of the solution using each of the plots and scenes that you created previously.

In the following screenshot, you see a graph of the thrust created by the virtual disk as a function of iteration.

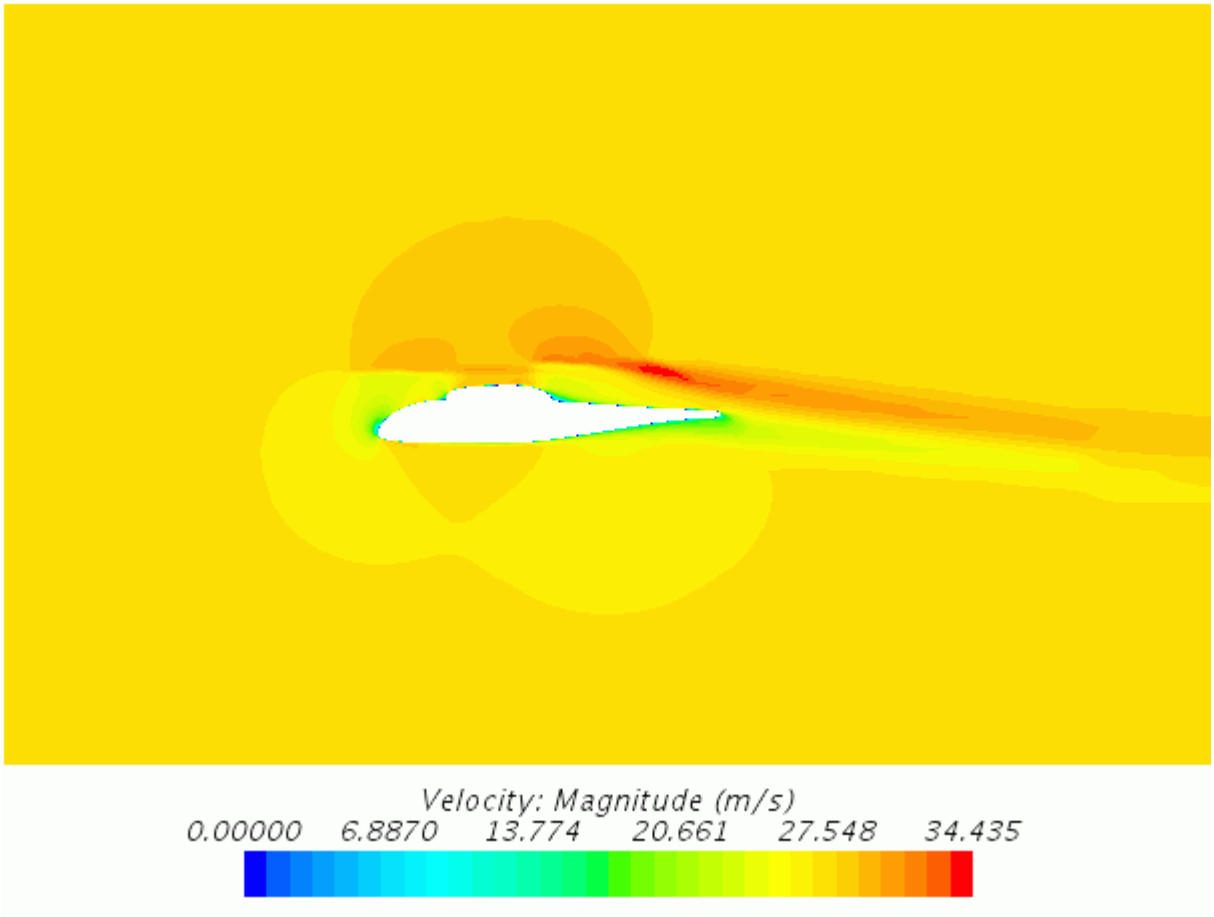


Here, you see the corresponding graph of the collective pitch angle as a function of iteration.

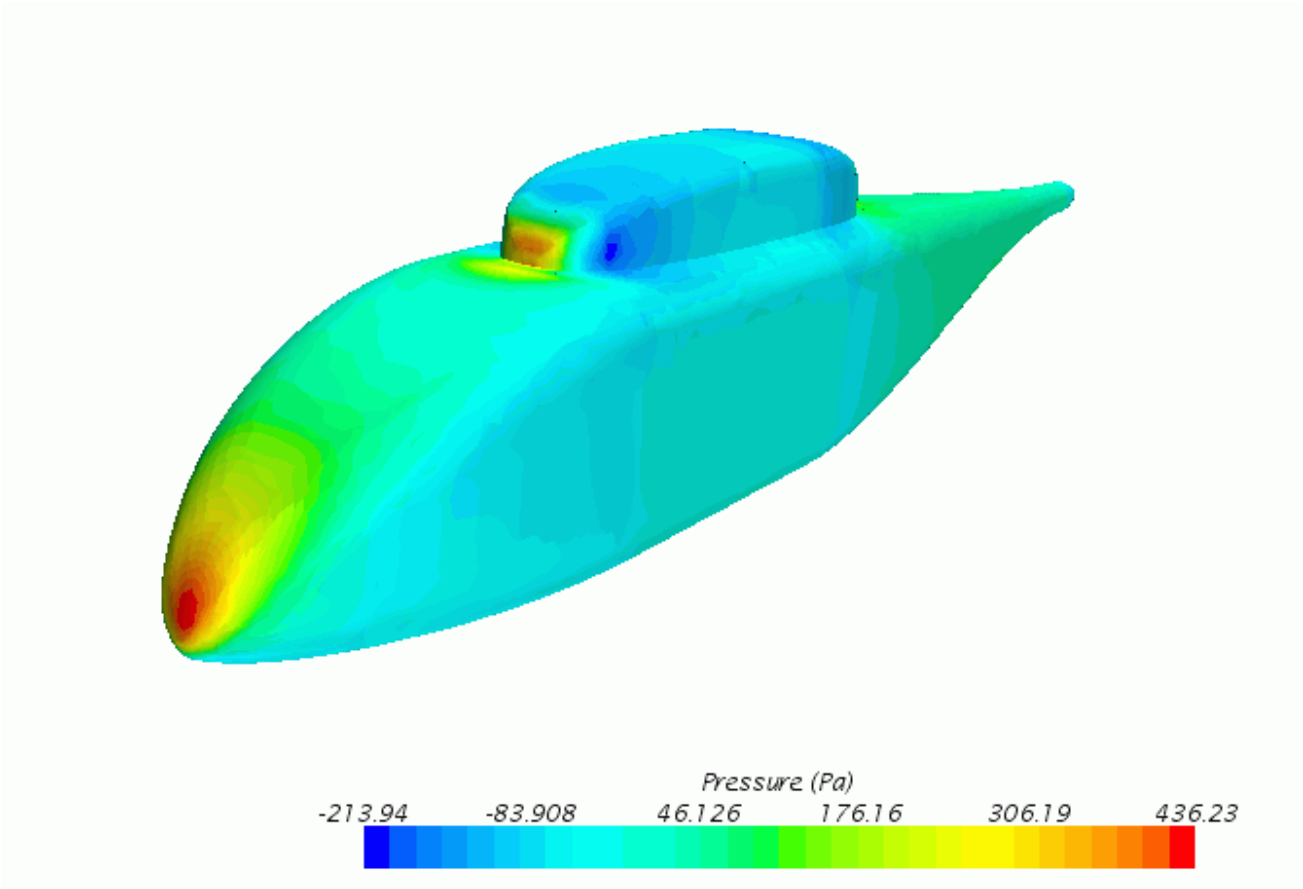


Using these two graphs, you can monitor the trimming process, that is, you can follow the change in collective pitch angle and its corresponding change of thrust. For the first 100 iterations, the rotation rate is ramped linearly from 1 to the specified value of 2000 rpm. During this period, the collective pitch angle remains constant at the specified value of 7.7°. After 100 iterations, STAR-CCM+ activates the trimming algorithm. You can see that the collective pitch angle increases non-linearly to reach a constant value after approximately 500 iterations as the run converges. When the trimming algorithm is active, the thrust is computed using the varying collective pitch angle. The collective pitch angle is adjusted until the specified target thrust is reached.

The following screenshot shows you a close-up of the velocity contours around the helicopter fuselage.



Next, you can see the pressure contours on the helicopter fuselage.



This plot allows you to analyse the aerodynamic loads on the helicopter body. You can also export this data to an external program for further structural analysis.

Summary

This tutorial has introduced the following STAR-CCM+ features:

- Modeling a helicopter rotor by using the Virtual Disk model.
- Parametrizing the variables for the operating condition by using field functions.
- Running a steady-state aerodynamics simulation.
- Plotting virtual disk specific reports.
- Analyzing results.

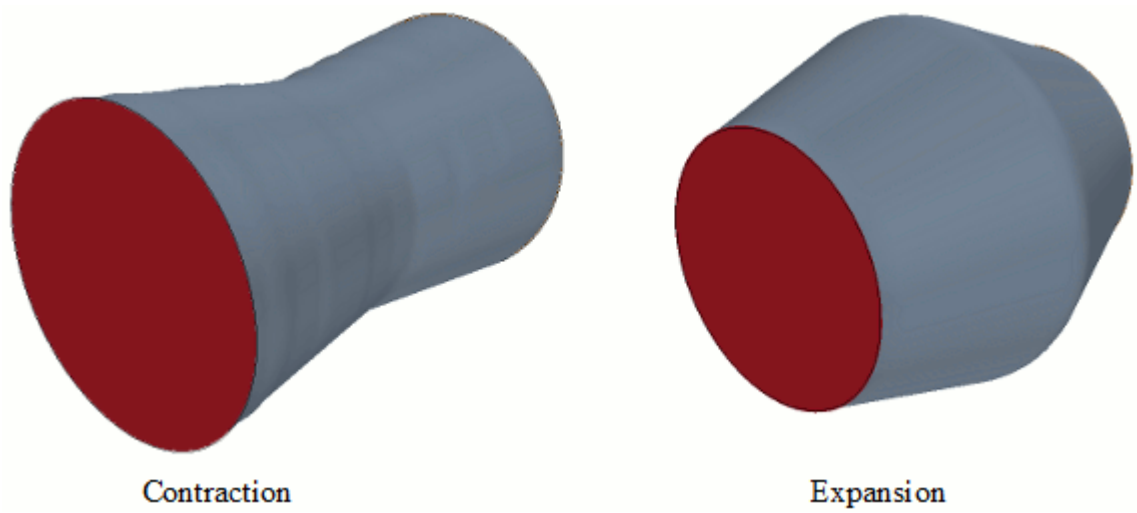
Bibliography

[805] *Mineck, R.E. and Gorton, S.A. 2000 "Steady and Periodic Pressure Measurements on a Generic Helicopter Fuselage Model in the Presence of a Rotor" , NASA/TM-2000-210286*

Morphing: Cylinder with Boundary Motion

STAR-CCM+ contains a Morphing Motion Model that enables you to define motion on boundary surfaces using several methods.

In this tutorial, two of the available methods are used to impose motion on the outer wall of a simple cylinder. A grid velocity is applied in the first part of the tutorial to cause contraction of the cylinder, and a boundary displacement is applied in the second part to cause expansion. These deformations are illustrated below.



Contents:

Importing the Geometry and Naming the Simulation
Preparing the Surfaces, Region, and Boundaries
Generating the Volume Mesh
Setting up the Physics
Setting up the Motion
Creating the Local Coordinate System
Creating a Field Function for the Grid Velocity
Setting Boundary Conditions and Values
Setting Initial Conditions
Setting Solver Parameters
Setting Stopping Criteria
Creating Additional Monitors
Visualizing the Solution
Setting Solver Verbosity
Running the Simulation
Visualizing the Results
Expanding the Cylinder

Summary

Importing the Geometry and Naming the Simulation

The first step in this tutorial is to import the cylinder geometry from the Parasolid file supplied.

1. Start up STAR-CCM+ in a manner that is appropriate to your working environment and select the New Simulation option from the menu bar. Click **OK** to accept the default options.
2. Select **File** > **Import** > **Import Surface Mesh** from the menu bar. In the *Open* dialog, navigate to the `doc/startutorialsdata/motion/data` subdirectory of your STAR-CCM+ installation directory and select file `cylinder.x_t`.
3. Click the **Open** button to start the import. The *Import Surface Options* dialog appears.
4. Select **Create New Part** from the **Import Mode** box.
5. Click **OK** to import the cylinder.

   STAR-CCM+ provides feedback on the import process in the **Output** window. A new geometry scene is created in the **Graphics** window showing the imported cylinder.
6. Save the new simulation to disk under file name `contraction.sim`.

Preparing the Surfaces, Region, and Boundaries

There is only one boundary surface at this stage, but as you need to have separate surfaces for the inlet and outlet boundaries, it is necessary to split the surface before creating the volume mesh.

Split the surface:

1. Right-click the **Geometry** > **Parts** > **Cylinder** > **Surfaces** > **Faces** node and select **Split By Angle**.
2. In the *Split Part Surfaces by Angle* dialog do the following:
   a) Make sure that the **Faces** surface is selected in the view panel.
   b) Accept the default angle.
   c) Click **OK**.

   Information on the splitting action is shown in the *Output* window. Three part surfaces are shown in the object tree.

Create a region and associated boundaries from the geometry part and its surfaces:

3. Right-click **Geometry** > **Parts** > **Cylinder** and select **Assign Parts to Regions...**.

   The *Assign Parts to Regions* dialog appears.
4. Select **Create One Region for All Parts**.
5. Select **Create a Boundary for Each Part Surface**.
6. Click **Apply**.
7. Close the dialog.

   In the object tree, all of the surfaces appear as individual boundaries within the region.

Rename the single region:

8. Rename the **Regions** > **Region** node to `Fluid`.

For this case generate a trimmed cell volume mesh aligned in the axial direction, with two layers of prism cells next to the cylinder wall. You do not wish to have prism cells on the inlet and outlet boundaries, however, and so to avoid this you must define the boundary types before you create the mesh. (The volume meshing tools do not generate prism layers on non-wall surfaces by default.)

9. Rename the **Regions** > **Fluid** > **Boundaries** > **cylinder.Faces 2** node to `Outlet`.
10. With the **Outlet** boundary still selected, set the *Type* to **Pressure Outlet**.
11. Rename **cylinder** to `Wall`.
12. Rename **cylinder.Faces 3** to `Inlet`.
13. Set the *Type* to **Velocity Inlet**.

Generating the Volume Mesh

The volume mesh is generated by creating an automated mesh operation, setting the appropriate sizing parameters, and running the mesh generator.

Create an automated mesh operation:

1. Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
2. In the *Create Automated Mesh Operation* dialog:
   a) Select **cylinder** from the Parts list.
   b) Select the following meshers, in order:

Group	Mesher
Surface Meshers	**Surface Remesher**
Core Volume Meshers	**Trimmed Cell Mesher**
Optional Boundary Layer Meshers	**Prism Layer Mesher**

   The **Automated Mesh** node appears in the **Operations** node.

Reference sizes must now be set at the global level to make sure cell sizes are appropriate for the simulation.

3. Select the **Automated Mesh** > **Default Controls** > **Base Size** node and set *Value* to `0.1 m`.

Several of the other meshing parameters are set relative to the base size by default, and so can be left unchanged. Change the Prism Layer Thickness, as the default setting results in a fairly abrupt change in cell size between the prism layer and the core mesh.

4. Select the **Default Controls** > **Prism Layer Total Thickness** node and set *Percentage of Base* to `65.0`.

The volume mesh can now be created.

5. Right-click the **Geometry** > **Operations** node and select **Execute All**.

Setting up the Physics

A default Physics Continuum, **Physics 1**, was created when the volume mesh was generated. This continuum is used to define the physical models required for the simulation.

1. For the **Physics 1** continuum, select the following models, in order:

Group Box	Model
Time	**Implicit Unsteady**
Material	**Liquid**
Flow	**Segregated Flow**
Equation of State	**Constant Density**
Viscous Regime	**Turbulent**
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**

2. Click **Close**.

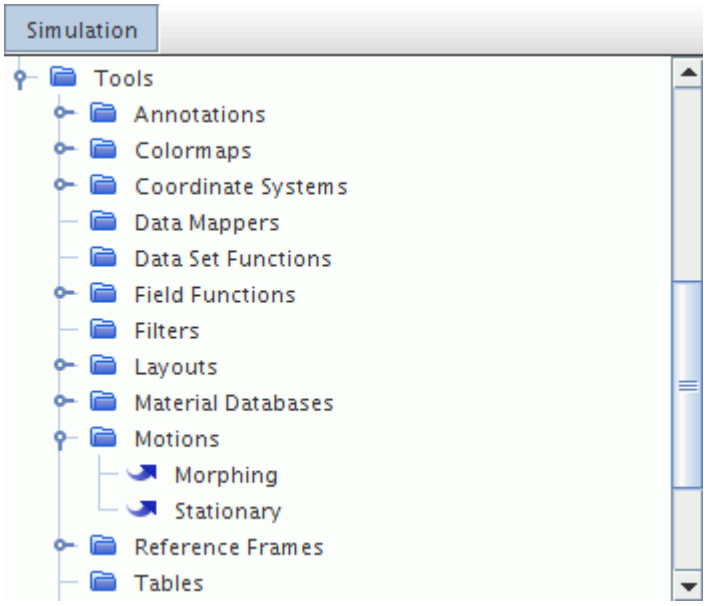
Setting up the Motion

Motion is defined in STAR-CCM+ using motion objects, which are managed by the Motions manager.

To define morphing motion, a Morpher motion object must be created and applied to the region within which morphing takes place.

1. Right-click the **Tools** > **Motions** node and select **New** > **Morphing**.

   A new motion node, called **Morphing**, is added to the **Motions** manager.



This motion is now applied to the **Fluid** region.

2. Select the **Regions** > **Fluid** > **Physics Values** > **Motion Specification** node and set *Motion* to **Morphing**.

   This change adds the mesh morpher solver to the **Solvers** manager.

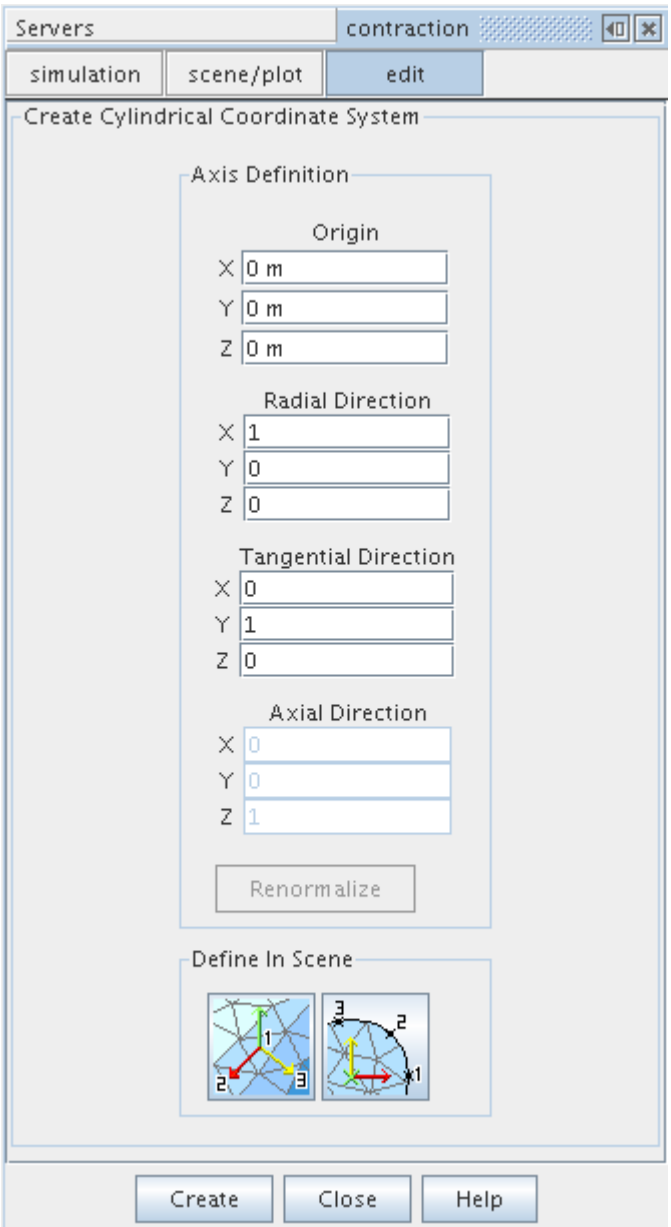
Creating the Local Coordinate System

Define the cylinder wall motion by a grid velocity in the radial direction. It is necessary to create a local cylindrical coordinate system in which to define the radial velocity as one does not exist by default.

1. Right-click the**Tools** > **Coordinate Systems** > **Laboratory** > **Local Coordinate Systems** node and select **New** > **Cylindrical**.

   The *Create Cylindrical Coordinate System* dialog appears in the **Edit** window.

   The default parameters are acceptable for this analysis, as the radial direction of the new system is in line with the x axis of the global cartesian system, and the axial direction of the new system is coincident with the z axis of the global system.

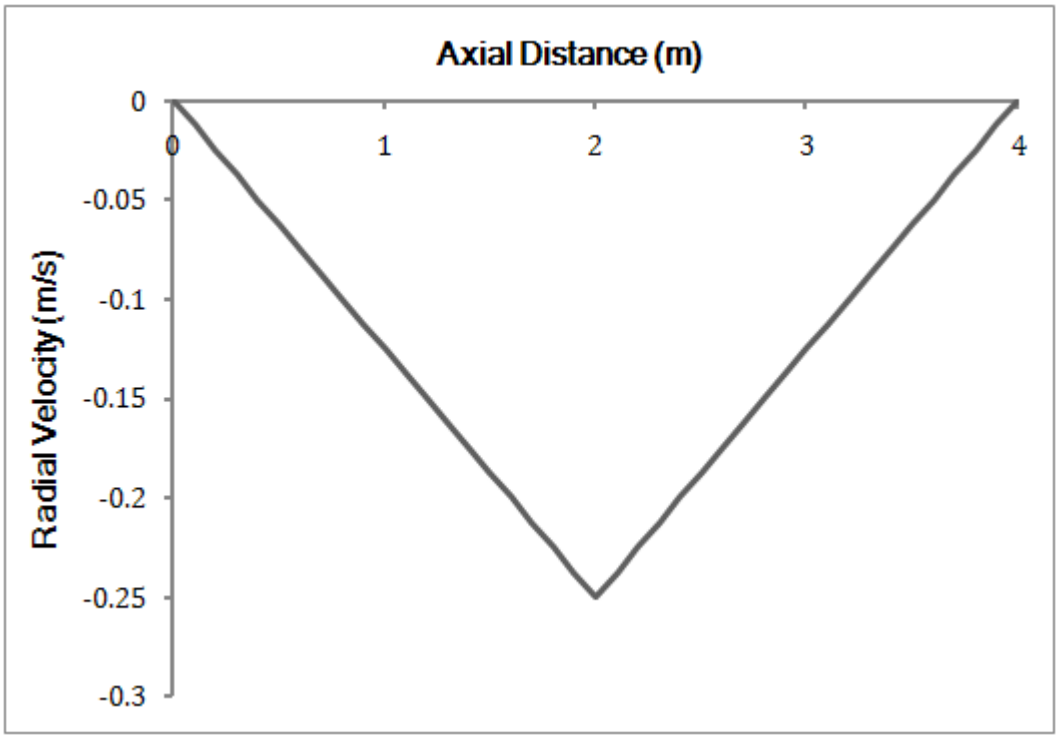


2. Click **Create** and then **Close**.

Creating a Field Function for the Grid Velocity

A radial velocity is to be set at the cylinder wall using a field function based on the axial position.

The variation of the radial velocity with axial distance is shown in the figure below:



The equation for this profile is:

$$v_r = \begin{cases} -\frac{1}{8}x_z & \text{for } x_z < 2 \\ -\frac{1}{8}\left(4 - x_z\right) & \text{for } x_z \geq 2 \end{cases}$$

where v_r is the radial velocity in m/s and x_z is the z (or axial) coordinate of the face centroid in m.

The field function for this profile is implemented as follows:

1. Right-click the **Tools** > **Field Functions** node and select **New** > **Vector**.
2. Rename the **User Field Function 1** node to `Cylinder Radial Velocity`.
3. Select the **Cylinder Radial Velocity** node and set the following properties:

Property	Setting
Function Name	`CylinderRadialVelocity`
Dimensions	**Velocity**
Definition	`[(($$Position("Cylindrical 1")[2] < 2) ? -0.125* $$Position("Cylindrical 1")[2] : -0.125*(4-$ $Position("Cylindrical 1")[2])), 0, 0]`

Note: the function `$$Position("Cylindrical 1")` returns a cartesian vector of the position coordinates in local coordinate system **Cylindrical 1**; the subscript operator `[]` is appended to access a single component of the position vector (with an index of `[0]`, `[1]`, or `[2]`).

Setting Boundary Conditions and Values

The boundary types were set before the volume meshing stage to make sure prism layers were obtained only on the wall.

Set physical values where necessary, and also specify conditions to control morphing of the mesh. Note: Morpher-specific conditions are required even on boundaries that are not explicitly deformed.

1. Edit the **Regions** > **Fluid** > **Boundaries** > **Inlet** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Morpher Specification**	*Specification*	**Constraint**
└ **Morpher Constraint Specification**	*Constraint*	**Boundary Plane**
Physics Values		
└ **Morpher Plane Options**	*Damping Factor*	`0.01 m`
└ **Velocity Magnitude**	*Value*	`10.0 m/s`

The **Boundary Plane** constraint permits vertices on the inlet boundary to move in the plane of the boundary surface itself, but not orthogonal to it.

This completes the settings for the **Inlet** boundary. The **Outlet** boundary has the same Morpher settings as the **Inlet** boundary.

2. Edit the **Regions** > **Fluid** > **Boundaries** > **Outlet** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Morpher Specification**	*Specification*	**Constraint**
└ **Morpher Constraint Specification**	*Constraint*	**Boundary Plane**
Physics Values		
└ **Morpher Plane Options**	*Damping Factor*	`0.01 m`

The **Wall** boundary is to be deformed during the simulation.

3. Edit the **Regions** > **Fluid** > **Boundaries** > **Wall** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Morpher Specification**	*Specification*	**Displacement**
└ **Morpher Displacement Specification**	*Specification*	**Grid Velocity**
Physics Values		
└ **Grid Velocity**	*Coordinate System*	**Laboratory-** > **Cylindrical 1**
	Method	**Field Function**
	Vector Function	**Cylinder Radial Velocity**

Setting Initial Conditions

For this tutorial, the only initial condition to set is the flow axial velocity.

Set the axial velocity to 10.0 m/s:

1. Select the **Continua** > **Physics 1** > **Initial Conditions** > **Velocity** node and set *Value* to `[0, 0, 10.0] m/s`.

Setting Solver Parameters

This is a simple transient simulation and so an appropriate time step must be set.

As there are no complex flow phenomena a reasonably large time step can be used. Make sure that the mesh motion proceeds at a pace that permits adequate convergence of the fluid solution in each time step.

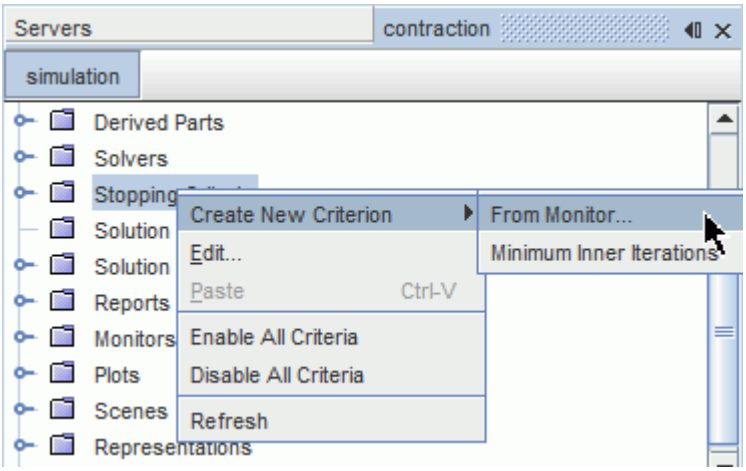
1. Select the **Solvers** > **Implicit Unsteady** node.
2. In the **Properties** window, set the *Time-Step* to `0.02 s`.

Setting Stopping Criteria

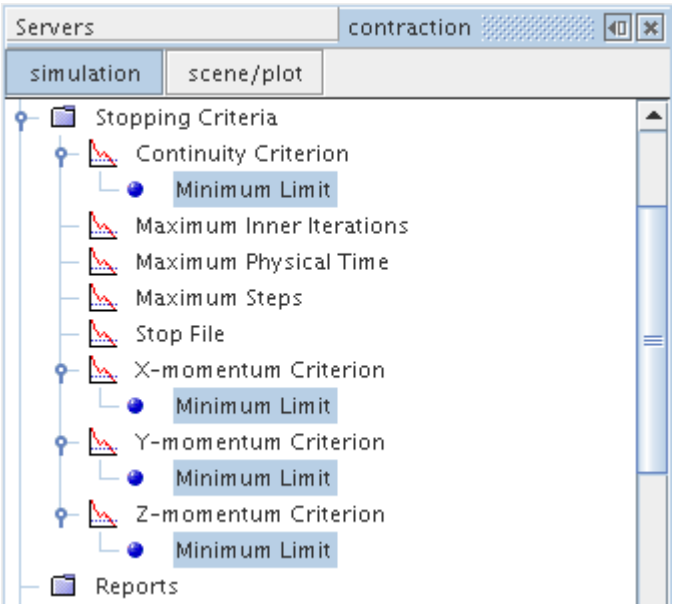
Stopping Criteria is created based on the Monitors for Continuity and Momentum to make sure that the solution does not iterate unnecessarily within a single time-step.

Set the Maximum Physical Time to 1.0 s.

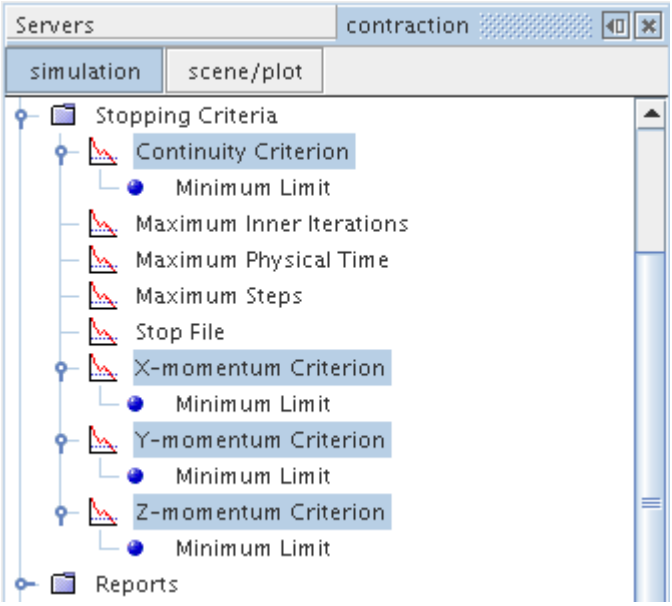
1. Right-click the **Stopping Criteria** node and select **Create New Criterion** > **From Monitor...**.



2. In the *Select Monitor* dialog, select the **Continuity** monitor and click **OK**.
3. Repeat this step to create criteria for **X-momentum**, **Y-momentum**, and **Z-momentum**.
4. Select the **Stopping Criteria** > **Continuity Criterion** > **Minimum Limit** node, and then use `Ctrl Click` to select the **Minimum Limit** nodes for the **X-momentum Criterion**, the **Y-momentum Criterion** and the **Z-momentum Criterion**.



5. Set the *Minimum Value* to `0.001`.
6. Select the **Stopping Criteria** > **Continuity Criterion** node, and then use `Ctrl Click` to select the **X-momentum Criterion**, **Y-momentum Criterion**, and **Z-momentum Criterion** nodes.



7. In the **Properties** window change the *Logical Rule* to **And**.

Having a *Logical Rule* of **And** means that the value of all four monitors (**Continuity**, **X-momentum**, **Y-momentum** and **Z-momentum**) must fall to 0.001 before the simulation assumes that an adequate level of convergence has been achieved and move on to the next time step. It is possible that one of the other Stopping Criteria could activate before reaching this level of convergence, in which case the simulation would move on to the next time step regardless.

8. Select the **Stopping Criteria** > **Maximum Inner Iterations** node and set the *Maximum Inner Iterations* to `10`.
9. Select the **Stopping Criteria** > **Maximum Physical Time** node and make sure that the *Max Physical Time* property is set to `1.0 s`.

Creating Additional Monitors

It is generally useful to create one or two additional monitors to help assess convergence of the simulation.

1. Right-click the **Reports** node and select **New Report** > **Mass Flow Averaged**.
2. Rename the **Mass Flow Averaged 1** node to `Inlet Pressure`.
3. In the **Properties** window, do the following:
   a) Set *Scalar Field Function* to **Pressure**.
   b) Set *Parts* to **Regions** > **Fluid** > **Boundaries** > **Inlet**.

Use the **Inlet Pressure** report to create a monitor and associated plot:

4. Right-click the **Reports** > **Inlet Pressure** node and select **Create Monitor and Plot from Report**.
5. Expand the **Reports**, **Monitors** and **Plots** nodes to see the related items.
6. Save the simulation.

Visualizing the Solution

A scalar scene is created to show the deformation of the cylinder wall as the simulation proceeds.

1. Right-click the **Scenes** node and select **New Scene** > **Scalar**.
2. Rename the **Scalar Scene 1** node to `Cylinder Wall`.
3. Select the **Cylinder Wall** node and use the **scene/plot** button at the top of the simulation tree to switch to the tree for this scene.
4. Select the **Cylinder Wall** > **Displayers** > **Outline 1** node.
5. In the **Properties** window, tick the *Surface* property, change the *Opacity* value to `0.1`, and set the *Representation* to **Geometry**.
6. Continuing within the **Cylinder Wall** tree, select the **Displayers** > **Scalar 1** > **Parts** node and set *Parts* to **Regions** > **Wall**.
7. Select the **Scalar 1** > **Scalar Field** node and set *Function* to **Centroid** > **Laboratory** > **Cylindrical 1** > **r**.
8. Click the **simulation** button to switch back to the simulation tree.

Setting Solver Verbosity

The Mesh Morpher solver has an option to display additional information in the **Output** window during runtime. This can be useful in gauging the proportion of CPU time devoted to the Morphing process.

1. Select the **Solvers** > **Mesh Morpher** node.
2. In the **Properties** window, set the *Verbosity* to **High**.
3. Save the simulation.

Running the Simulation

The setup is complete and you can now run the simulation.

To initialize the solution and run the analysis:

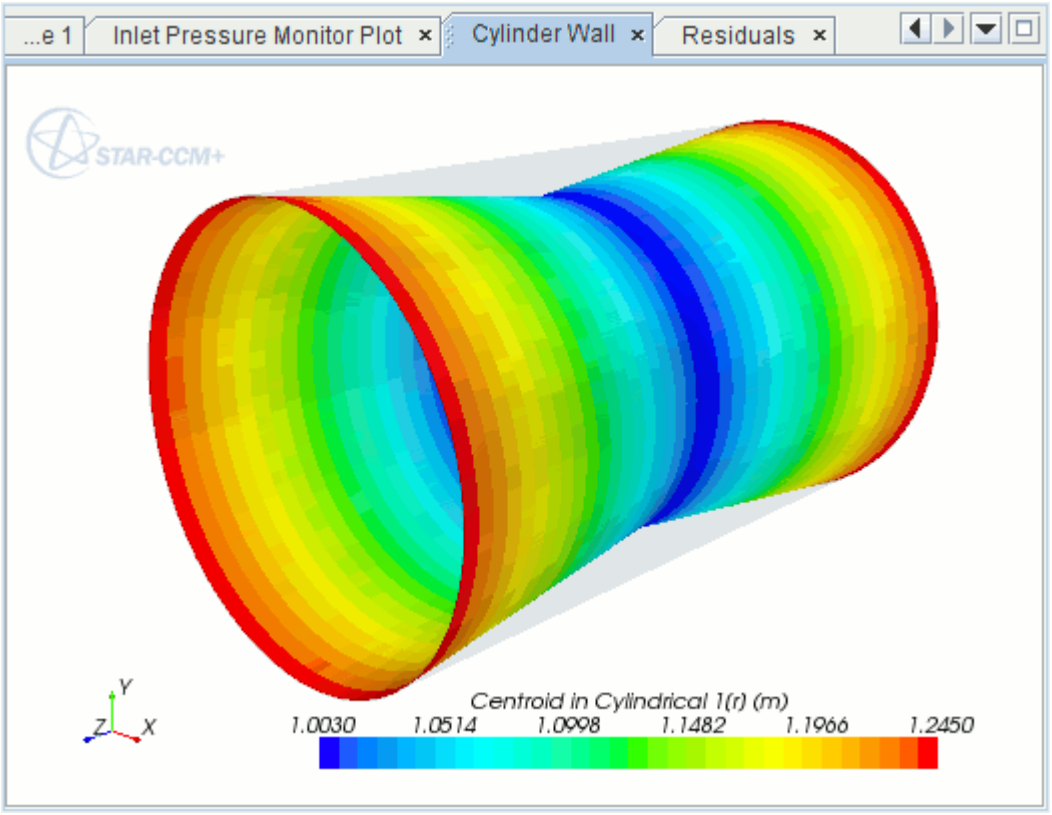
1. Click the (**Initialize Solution**) button in the **Solution** toolbar or use the **Solution** > **Initialize Solution** menu item.
2. Click the (**Run**) button in the **Solution** toolbar to run the analysis.

The *Residuals* display is automatically created and shows the solver's progress. If necessary, click the **Residuals** tab to bring the *Residuals* plot into view.

At the end of the simulation:

3. Click the **Cylinder Wall** tab at the top of the **Graphics** window.

   A cylinder wall scene similar to the one shown below appears.

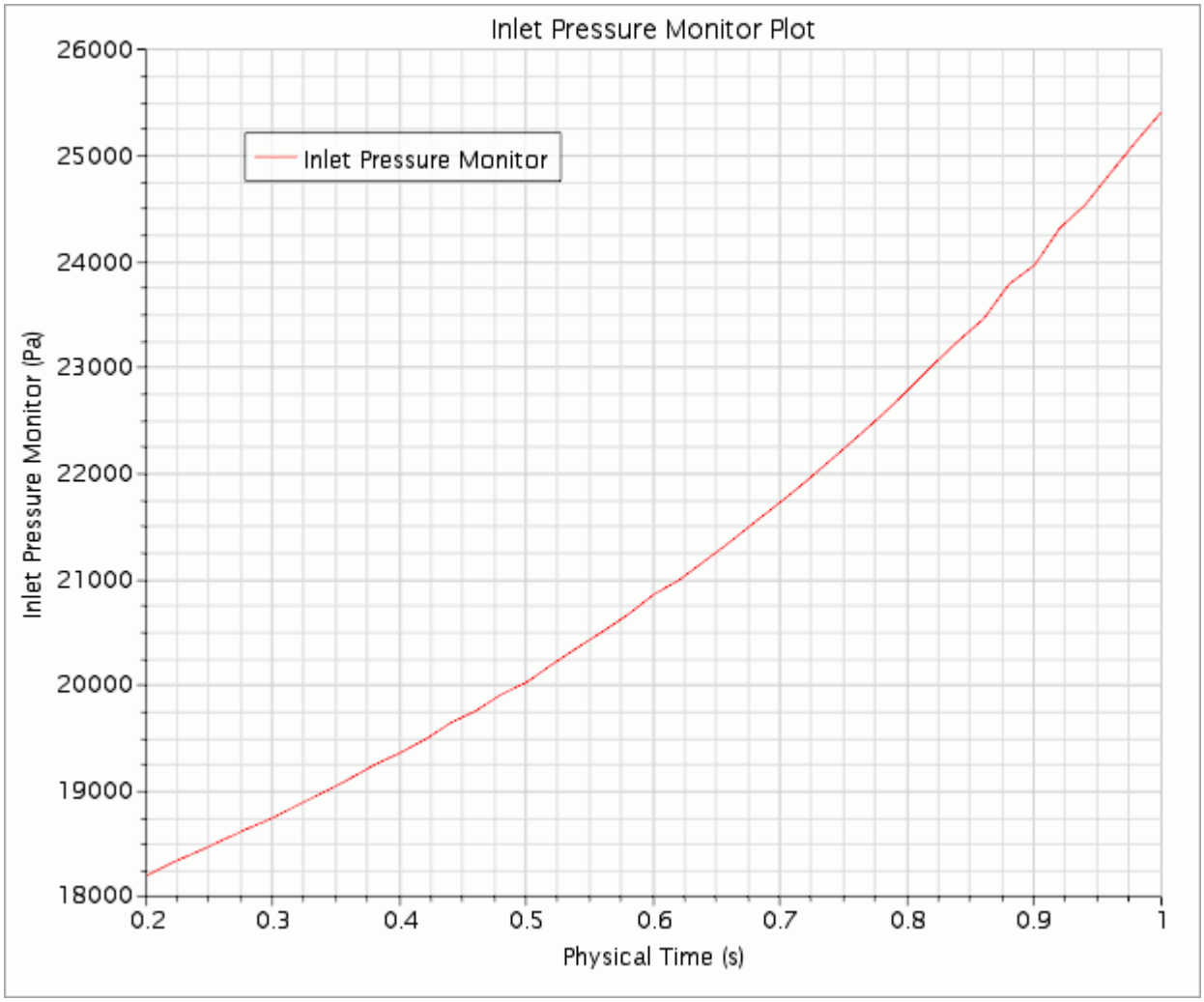


The cylinder would continue to contract if the simulation were run for longer, as no time limiting factor was built into the field function used to specify the grid velocity.

Modify the scales of the **Inlet Pressure Monitor Plot** to view the solution between 0.2 s and 1.0 s.

4. Select the **Plots** > **Inlet Pressure Monitor Plot** > **Axes** > **Bottom Axis** node and set *Minimum* to `0.2` and *Maximum* to `1.0`.
5. Select the **Axes** > **Left Axis** node and set *Minimum* to `18000.0` and *Maximum* to `26000.0`.

   The resulting **Inlet Pressure Monitor Plot** is shown below.



Visualizing the Results

A new scene is created to show streamlines of velocity magnitude running through the geometry.

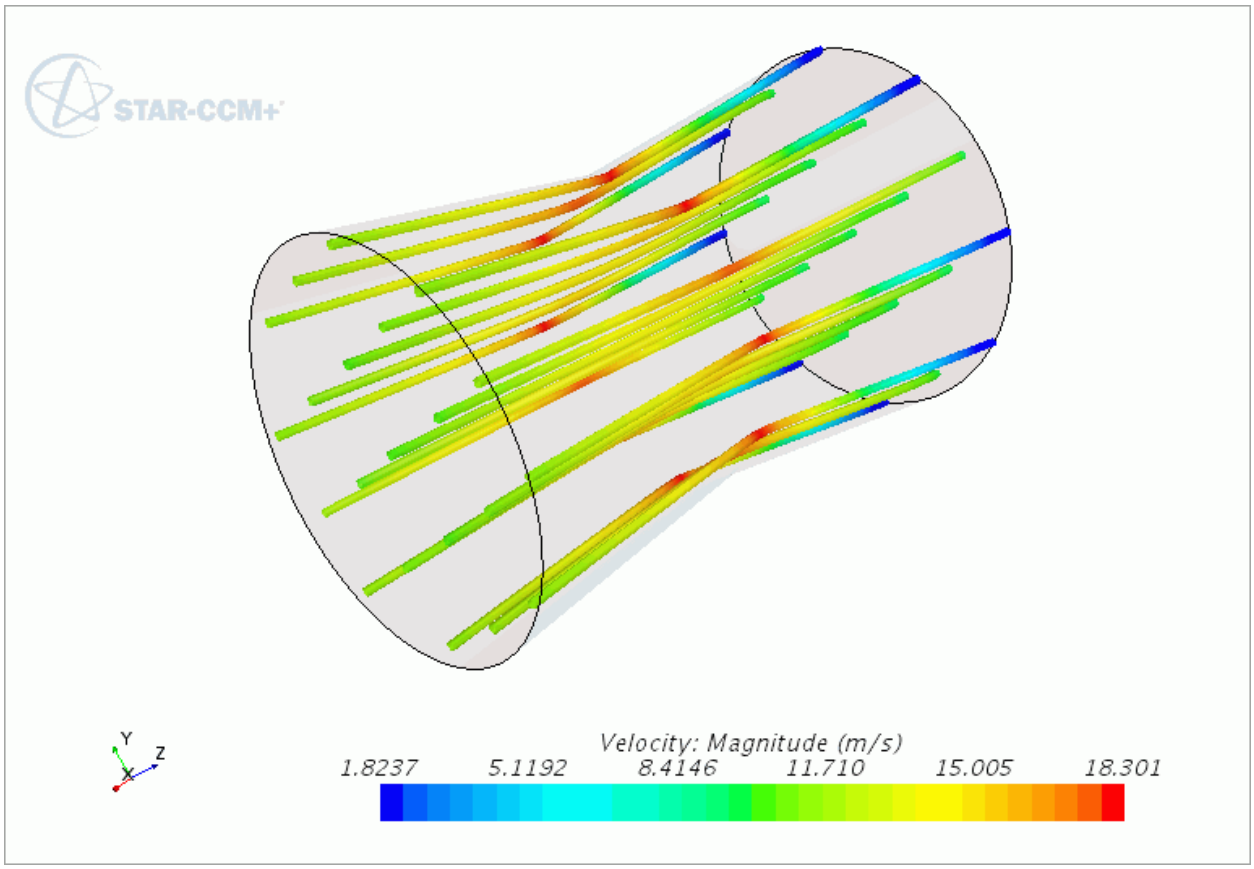
1. Right-click the **Scenes** node and select **New Scene** > **Geometry**.
2. Rename the **Scenes** > **Geometry Scene 2** node to `Streamlines`.

A derived part is created for the streamlines which can be added to the new scene.

3. Right-click the **Derived Parts** node and select **New Part** > **Streamline**.
4. In the *Create Streamline* panel, click **Select...** in the **Seed Parts** box. In the dialog that appears, select **Regions** > **Boundaries** > **Inlet** and click **OK**.
5. Click **Create** and then **Close**.
6. Select the **Scenes** > **Streamlines** node then click the **Scene/Plot** button to show only the relevant portion of the tree.
7. Select the **Streamlines** > **Displayers** > **Streamline Stream 1** > **Scalar Field** node and set *Function* to **Velocity: Magnitude**.

8. Select the **Displayers** > **Streamline Stream 1** node and set Mode to **Tubes** and Width to `0.03`.
9. Finally, select the **Displayers** > **Geometry 1** node.
10. Set the *Opacity* to `0.1`.

    A streamlines scene similar to the one shown below appears.



11. Switch back to the simulation tree using the **simulation** button.
12. Save the simulation.

Expanding the Cylinder

The same exercise can be repeated to model an expansion of the cylinder instead of a contraction. A displacement is specified at the wall boundary instead of a grid velocity.

1. Select **File** > **Save As** and save the simulation with a filename of `expansion.sim`.
2. Create a field function for the cylinder wall displacement with respect to the local cylindrical coordinate system:
   a) Right-click **Tools** > **Field Functions** and select **New** > **Vector**.
   b) Rename the vector field function to `Cylinder Wall Displacement`.
   c) Select the **Cylinder Wall Displacement** node and set the following properties:

Property	Setting
Function Name	`CylinderWallDisplacement`
Dimensions	**Length**
Definition	`[(($$Position("Cylindrical 1")[2] < 2) ? 0.00625*$$Position("Cylindrical 1")[2] : 0.00625*(4-$$Position("Cylindrical 1")[2])), 0, 0]`

The morpher conditions on the cylinder wall must be modified to use the new field function.

3. Edit the **Regions** > **Fluid** > **Boundaries** > **Wall** node and set the following properties:

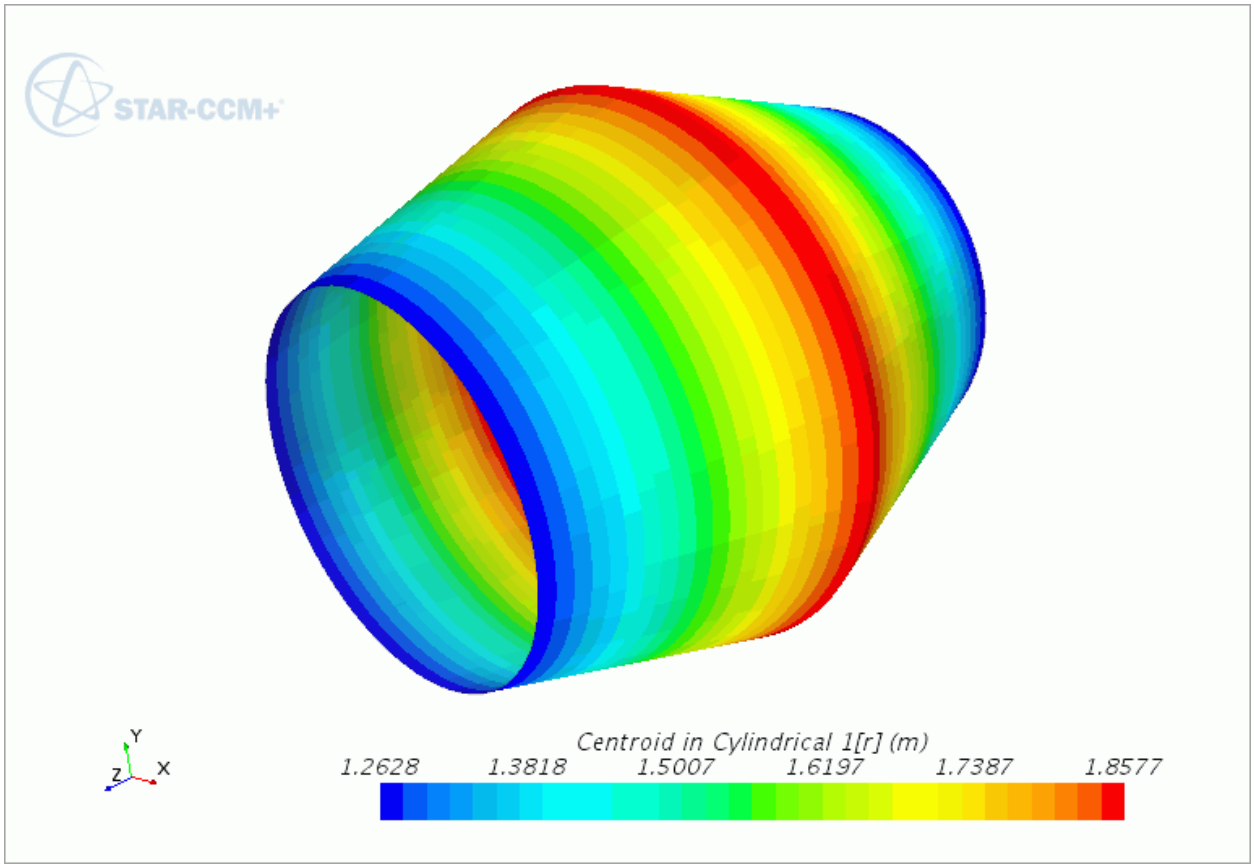
Node	Property	Setting
Physics Conditions		
└ **Morpher Specification**	*Specification*	**Displacement**
└ **Morpher Displacement Specification**	*Specification*	**Coordinate Offset**
Physics Values		
└ **Coordinate Offset**	*Method*	**Field Function**
	Vector Function	**Cylinder Wall Displacement**
	Coordinate System	**Laboratory->Cylindrical 1**

This completes the changes necessary for the expansion simulation.

4. Select **Solution** > **Clear Solution** from the menu bar and click **OK**.
5. Proceed to initialize and run the simulation as before using either the **Solution** > **Run** option or the (**Run**) button on the toolbar.

At the end of this second simulation:

6. Click the **Cylinder Wall** tab at the top of the **Graphics** window.
7. Use the mouse buttons to zoom out and position the part to obtain a view similar to that shown below.



Summary

This tutorial covered the following features:

- Import of a simple CAD geometry.
- Splitting boundaries and renaming.
- Generating a trimmed cell mesh with prism layers on the wall only.
- Selecting the Morpher motion and applying it to a region.
- Creating a field function to provide a grid velocity vector for the Morpher motion.
- Applying appropriate boundary conditions for the Morpher.
- Creating a scene to display the centroid on the deformed surface of the geometry.
- Creating a streamline derived part and using it to visualize the velocity field.
- Creating a field function to provide a displacement vector for the Morpher motion model.
- Using the Morpher motion model to simulate expansion as well as contraction.

Reacting Flow

The tutorials in this set illustrate various STAR-CCM+ features for simulating reacting flows such as combustion.

Contents:

3-Step Eddy Break-Up: Non-Premixed Propane
Complex Chemistry: Methane-Air Jet Flame
Spray Combustion: Jet A Fuel
Steady Laminar Flamelet: Non-Premixed Hydrogen
Surface Chemistry: Methane on Platinum Oxidation
Reacting Channels: Steam Cracking Furnace
Fire and Smoke Wizard: Steckler Room

3-Step Eddy Break-Up: Non-Premixed Propane

This tutorial models propane combustion in air using a 3-step Eddy Break-Up model.

For most steady combustion simulations which use only one fuel and one oxidizer stream, it is best to use one of the Flamelet models. Flamelet models are generally more accurate and easier to set up than the EBU model. However, in some situations, it is not always suitable to use a Flamelet model—such as in systems with three streams. This tutorial shows how to set up the Eddy Break-Up model using a simple set-up—which can alternatively be set up using a Flamelet model.

The following equations show the 3-step reaction that is modeled in this simulation.

$$C_3H_8 + 1.5O_2 \rightarrow 3CO + 4H_2$$

$$CO + 0.5O_2 \rightarrow CO_2$$

$$H_2 + 0.5O_2 \rightarrow H_2O$$

Air enters the combustion chamber through the three air inlets and propane gas enters through the fuel inlet. Both air and fuel are at a pressure of 1 bar and a temperature of 293 K at the inlets.

Contents:

Prerequisites
Importing the Mesh and Naming the Simulation
Setting up the Models
Setting Material Properties
Defining Reactions
Setting Initial Conditions and Reference Values
Creating Interfaces
Setting Boundary Conditions and Values
Setting Solver Parameters and Stopping Criteria
Visualizing the Solution
Reporting, Monitoring, and Plotting
Running the Simulation
Visualizing the Results
Summary

Prerequisites

The instructions in the 3-Step Eddy Break-Up: Non-Premixed Propane tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Importing the Mesh and Naming the Simulation

To set up the STAR-CCM+ simulation, launch a simulation and import the supplied volume mesh.

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh**.
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/reactingFlow/data`.
5. Select `combustor.ccm` then click **Open**.

   STAR-CCM+ provides feedback on the import process in the *Output* window and a mesh region is created under the **Regions** node.
6. Save the simulation as `ebuCombustion.sim`.

Setting up the Models

Models define the primary variables of the simulation, including pressure, temperature, velocity, and what mathematical formulation is used to generate the solution. In this example, the flow involves a turbulent, compressible, multi-component gas whose components are reacting chemically. The Segregated Flow model is used together with the Standard K-Epsilon turbulence model and the Eddy Break-Up reaction model.

1. For the physics continuum, **Continua** > **Physics 1**, select the following models in order:

   Note: **Auto-select recommended models** must be activated.

Group Box	Model
Enabled Models	**Three-Dimensional** (Selected automatically)
Time	**Steady**
Material	**Multi-Component Gas**
Reaction Regime	**Reacting**
Reacting Flow Models	**Reacting Species Transport**
Reacting Species Models	**Eddy Break-Up**
Flow	**Segregated Flow**
Enabled Models	**Gradients** (Selected automatically)
	Segregated Species (Selected automatically)

Group Box	Model
	Segregated Fluid Enthalpy (Selected automatically)
Equation of State	**Ideal Gas**
Viscous Regime	**Turbulent**
Enabled Models	**Reynolds-Averaged Navier-Stokes** (Selected automatically)

2. Deactivate **Auto-select recommended models**.
3. Select the following models in order:

Group Box	Model
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**
K-Epsilon Turbulence Models	**Standard K-Epsilon**
Enabled Models	**Exact Wall Distance** (Selected automatically)
Optional Models	**High y+ Wall Treatment**

4. Click **Close**.

   The Standard K-Epsilon model is preferable to the default Realizable Two-layer K-Epsilon model because the near-wall mesh is too coarse to resolve the boundary layer properly. Two-layer models are not recommended for meshes without refined near-wall layers.
5. To review the models, open the **Physics 1** > **Models** node.

To keep run times low, the mesh used in this tutorial is rather coarse. As a result, a solution can be obtained much faster if first-order convection term differencing is used for the momentum, species, energy, and turbulence equations. To change the differencing schemes:

6. Select the **Models** > **Segregated Flow** node and set *Convection* to **1st-order**.
7. Repeat this process for the **Segregated Species**, **Segregated Fluid Enthalpy**, and **Standard K-Epsilon** nodes.
8. Save the simulation.

Setting Material Properties

Define the components of the mixture and its material properties.

The physical properties of the air components (23.3% O2 and 76.7% N2, by mass) and the rest of the reaction components (C3H8, CO, H2, CO2, H2O) are defined as follows:

	O2	N2	C3H8	CO	H2	CO2	H2O
Molecular weights	32.0	28.008	44.1	28.01	2.018	44.01	18.02
Density	Ideal Gas						

Molecular viscosity	1.716 x 10^{-5}Pas
Specific Heat	Determined using thermodynamic polynomial functions
Thermal conductivity	Determined using the Lewis number

The options included within the **Material Properties** node define the material properties of the gas mixture as a whole. Specify these methods before setting up the individual mixture components.

1. Select the **Physics 1** > **Models** > **Multi-Component Gas** > **Material Properties** > **Dynamic Viscosity** node.
2. Set *Method* to **Constant**.

You are now ready to define the seven components of the mixture: C3H8, O2, CO, H2, CO2, H2O, and N2. All these species are included in the STAR-CCM+ materials database so detailed specification of their properties is not required.

3. Right-click the **Gas Components** node and choose **Select Mixture Components**.
4. In the *Select Mixture Components* dialog, expand the **Material Databases**, **Standard**, and **Gases** nodes to reveal the available gaseous components.
5. Select the required components: **C3H8 (Propane)**, **CO (Carbon Monoxide)**, **CO2 (Carbon Dioxide)**, **H2 (Hydrogen)**, **H2O (Water)**, **N2 (Nitrogen)**, and **O2 (Oxygen)**.
6. When complete, click **Apply** then **Close**.

   By default, STAR-CCM+ defines the specific heat of each component using a corresponding thermodynamic polynomial. These polynomials are retrieved from the materials database.
7. Save the simulation.

Defining Reactions

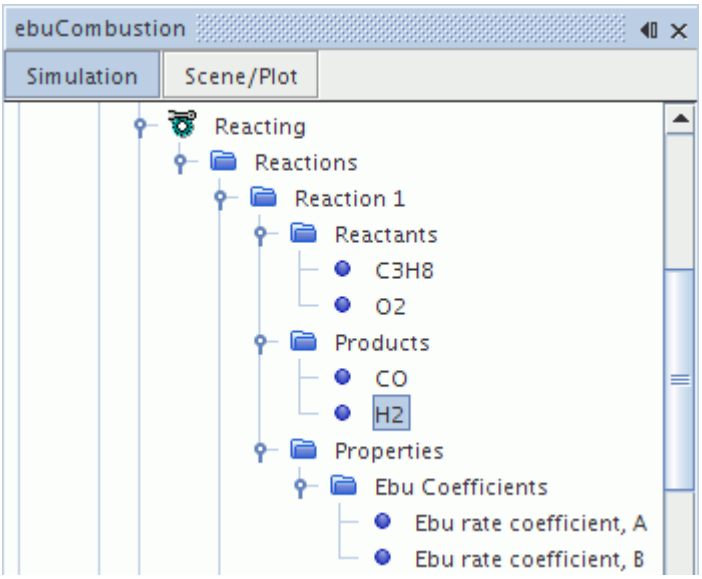
1. Select the **Models** > **Eddy Break-Up** node.
2. Set *Reaction Control* to **Standard EBU**.
3. Right-click the **Reacting** > **Reactions** node and select **New Reaction**.

   A **Reactions** > **Reaction 1** node appears. Three more nodes, named **Reactants**, **Products**, and **Properties**, are also added under the **Reaction 1** node.

Define the first reaction:

4. Right-click the **Reactants** node and select **Add Reactant** > **Gas Mixture** > **C3H8**.
5. Right-click the **Reactants** node and select **Add Reactant** > **Gas Mixture** > **O2**
6. Now add CO and H2 as products by right-clicking the **Products** node and selecting **Add Product** > **Gas Mixture**.

   This section of the simulation tree appears as shown in the following screenshot.



With all the necessary species added to **Reaction 1**, specify the stoichiometry of the reaction:

7. Select the **C3H8** node and see that the *Stoich. Coeff* is set to the default, `1.0`.
8. Set the *Stoich. Coeff* for O2, CO, and H2 to `1.5`, `3.0`, and `4.0`, respectively.

   The specification of **Reaction 1** is now complete.
9. Follow the same procedure to define the remaining two reactions of the chemical reaction scheme:

$$CO + 0.5O_2 \rightarrow CO_2$$

$$H_2 + 0.5O_2 \rightarrow H_2O$$

10. Save the simulation.

Setting Initial Conditions and Reference Values

Decrease the minimum allowable temperature to match the lower temperature limit for the polynomials.

The lower temperature limit for the specific heat polynomials that are imported from the materials database is 200 K. Although this temperature is much lower than you would expect to find in the converged solution, temperatures below this value could appear early in the run. For this reason, it is necessary to decrease the minimum allowable temperature to match the lower temperature limit for the polynomials.

1. Select the **Reference Values** > **Minimum Allowable Temperature** node and set *Value* to `200` K.

The initial condition of the combustor is a stationary flow field consisting entirely of air at 293 K.

2. Select the **Initial Conditions** > **Species Mass Fraction** node and set *Value* to:

O2	0.233
N2	0.767

   Maintain a *Value* of `0.0` for all other species.
3. Set the following properties:

Node	Property	Setting
Static Temperature	*Value*	`293` K
Turbulence Specification	*Method*	**Intensity + Length Scale**
Turbulence Intensity	*Value*	`0.05`
Turbulent Length Scale	*Value*	`0.2 m`

4. Save the simulation.

Creating Interfaces

All regions and boundaries already have suitable names so you can proceed to create the periodic interface linking the two plane, rectangular, cyclic boundaries.

1. Expand the **Regions** > **Default_Fluid** > **Boundaries** node.
2. Multi-select the **Cyclic1** and **Cyclic2** nodes.
3. Right-click one of the highlighted nodes and select **Create Interface**.

   Two internal interface boundary nodes appear under the **Boundaries** node and a node named **Interface 1** appears under the **Interfaces** node.
4. To change this interface to periodic, select the **Interfaces** > **Interface 1** node and set *Topology* to `Periodic`.

Setting Boundary Conditions and Values

All wall boundaries, including the baffle, are adiabatic, no-slip walls. Since **no-slip wall** is the default wall boundary type, no changes are required here. The default settings are also suitable for the outlet, so the four inlets are the only boundary conditions that you specify.

1. Expand the **Air_Inlet1** > **Physics Conditions** node and set the following properties:

Node	Property	Setting
Turbulence Specification	*Method*	**Intensity + Length Scale**
Velocity Specification	*Method*	**Components.**

The air and fuel are made to swirl on entry to the combustor by specifying inlet velocity vectors in a cylindrical coordinate system. To create the latter:

2. Right-click the **Tools** > **Coordinate Systems** > **Laboratory** > **Local Coordinate Systems** node and select **New** > **Cylindrical**.
3. In the *Create Cylindrical Coordinate System* in-place dialog, to accept the default properties, click **Create** then **Close**.

   A cylindrical coordinate system, **Cylindrical 1**, is created.
4. Expand the **Air_Inlet1** > **Physics Values** node and set the following properties:

Node	Property	Setting
Velocity	*Coordinate System*	**Laboratory -** > **Cylindrical 1**
	Value	`-11, -24, 42` m/s
Species Mass Fraction	O2 *Value*	0.233
	N2 *Value*	0.767
Static Temperature	*Value*	`293` K
Turbulence Intensity	*Value*	`0.1`

The required boundary conditions for **Air_Inlet1** are identical to the **Air_Inlet2** boundary conditions and similar to the **Air_Inlet3** and **Fuel_Inlet** boundary conditions. Therefore, it is convenient to copy the **Air_Inlet1** conditions to the other three inlet boundaries and change settings where necessary.

5. Right-click the **Air_Inlet1** node and select **Copy**.
6. Right-click the **Air_Inlet2**, **Air_Inlet3**, and **Fuel_Inlet** nodes, and select **Paste** each time.
7. Expand the **Air_Inlet3** > **Physics Values** node.
8. Select the **Velocity** node and set *Value* to `0,0,15` m/s.
9. Expand the **Fuel_Inlet** > **Physics Values** node.
10. Select the **Species Mass Fraction** node. Set the *Value* for C3H8 to 1.0 and set all other species to 0.0.
11. Click **OK**.
12. Select the **Velocity** node and set *Value* to `-28,-60,100` m/s.

    Specification of the boundary conditions is now complete.
13. Save the simulation.

Setting Solver Parameters and Stopping Criteria

The default under-relaxation factors for the flow and turbulence equations are suitable. However, reduce the under-relaxation fractions for the species and energy equations to make sure that the solution converges.

1. Expand the **Solvers** node and set the following properties:

Node	Property	Setting
Segregated Species	*Under-Relaxation Factor*	`0.8`
Segregated Energy	*Fluid Under-Relaxation Factor*	`0.8`

   It is important that the under-relaxation factors for the species and energy equations are the same to make sure that the two solutions remain synchronized. Other species and energy modeling settings, such as the choice of differencing scheme, are also kept the same.

The simulation is run for 500 iterations, which is sufficient to achieve a steady-state solution. This number can be specified using a stopping criterion.

2. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to `500`.

The solution does not run for more than 500 iterations, unless this stopping criterion is changed or disabled.

3. Save the simulation.

Visualizing the Solution

View the velocity vector field as the solution develops on a plane bisecting the mesh and passing through the axis of symmetry. A scene showing the temperature contours is created for the same plane.

1. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane**.
2. In the *Create Section* in-place dialog set the following properties:

Property	Setting
Input Parts	**Default_Fluid**
Plane Parameters	origin `0,0,0`
	vector `0,1,0`
Display	**No Displayer**

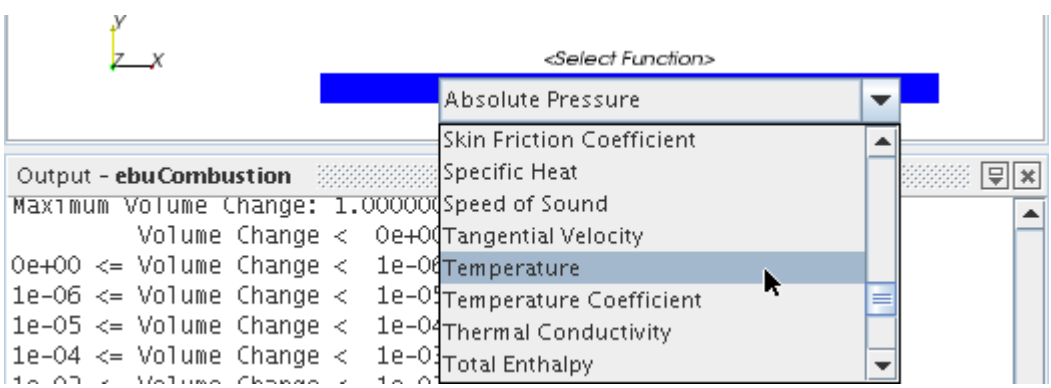
3. Click **Create** and **Close**.

To create the temperature contour scene:

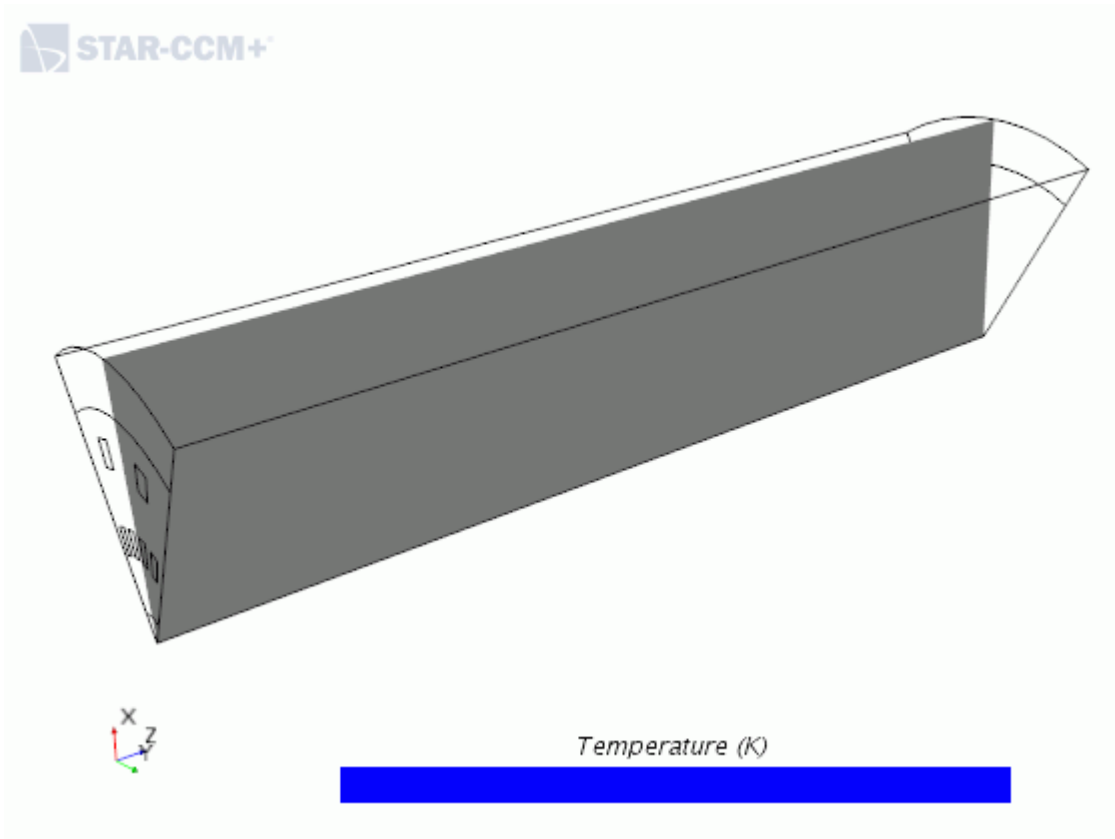
4. Create a Scalar Scene.
5. Expand the **Scalar Scene 1** > **Displayers** node and set the following properties:

Node	Property	Setting
Scalar 1	*Contour Style*	**Smooth Filled**
└ **Parts**	*Parts*	**Derived Parts** > **Plane Section**

6. Click **OK**.
7. In the *Graphics* window, right-click the scalar bar and select **Temperature**.



8. Rotate the scalar scene until the view is roughly perpendicular to the plane section and the inlet boundaries are on the left.



9. Save the simulation.

Reporting, Monitoring, and Plotting

STAR-CCM+ can dynamically monitor virtually any quantity while the solution develops. This procedure requires setting up a report defining the quantity of interest and the parts of the monitored region. A monitor is then defined based on that report. The former also helps to create an appropriate X-Y graph plot.

Report monitors are used in this case to compare the mass of carbon atoms flowing into the solution domain with the mass of such atoms flowing out of it. Once the two monitors reach the same value, it is reasonable to assume that the solution has converged also. To set up these reports, create a field function that calculates the mass flow rate of carbon. In view of the species that is involved in the 3-step reaction mechanism, this field function can be defined as follows:

$$M(C) = M\left(\frac{A_r(C)}{M_r(CO_2)}[CO2] + \frac{A_r(C)}{M_r(CO)}[CO] + \frac{3\,A_r(C)}{M_r(C_3H_8)}[C_3H_8]\right)$$

where $M(C)$ is the mass flow rate of carbon, M is the total mass flow rate, and $M_r(X)$, $A_r(X)$, and $[X]$ are the molecular weight, the atomic weight, and the concentration of species X, respectively. The corresponding field function is therefore:

$$M(C) = \frac{3}{77}M\left(7[CO2] + 11[CO] + 21[C_3H_8]\right)$$

To create the field function:

1. Right-click the **Tools** > **Field Functions** node, and select **New** > **Scalar**.
2. Rename the **User Field Function 1** node to `Carbon`.
3. Select the **Carbon** node and set the following properties:

Property	Setting
Function name	`Carbon`
Dimensions	**Dimensionless**
Definition	`(3/77)* $FaceFlux*($MassFractionCO2*7+ $MassFractionCO*11+ $MassFractionC3H8*21)`

The value of the Carbon field function is positive over outflow boundaries and negative over inflow boundaries. Therefore, to make a side-by-side comparison of carbon inflow and outflow, create another field function for the inflow boundaries that is the negative of the previously created one.

4. As before, right-click the **Field Functions** node and select **New** > **Scalar**.
5. Following the same procedure, change both the node name and function name of the field function to `MinusCarbon` and enter its definition as `0-$Carbon`.

You can now create the reports, monitors, and plots.

6. Create two **Sum** reports called **Carbon In** and **Carbon Out**.
7. Edit the **Reports** node and set the following properties:

Node	Property	Setting
Carbon In	*Parts*	**Air_Inlet1** **Air_Inlet2** **Air_Inlet3** **Fuel_Inlet**
	Field Function	**MinusCarbon**
Carbon Out	*Parts*	**Outlet**
	Field Function	**Carbon**

8. Multi-select the **Carbon In** and **Carbon Out** nodes.
9. Right-click one of the selected nodes and select **Create Monitor and Plot from Report**.
10. In the *Create Plot From Reports...* dialog, click **Single Plot**.

    With one plot of both reports, the convergence of the two quantities can be observed more easily.
11. Rename the **Reports Plot** node to `Carbon Balance`. Make the same entry for the *Title* property of this node.

To display the empty plot in the **Graphics** window:

12. Right-click the **Plots** > **Carbon Balance** node and select **Open**.
13. The analysis is now ready to be run.
14. Save the simulation.

Running the Simulation

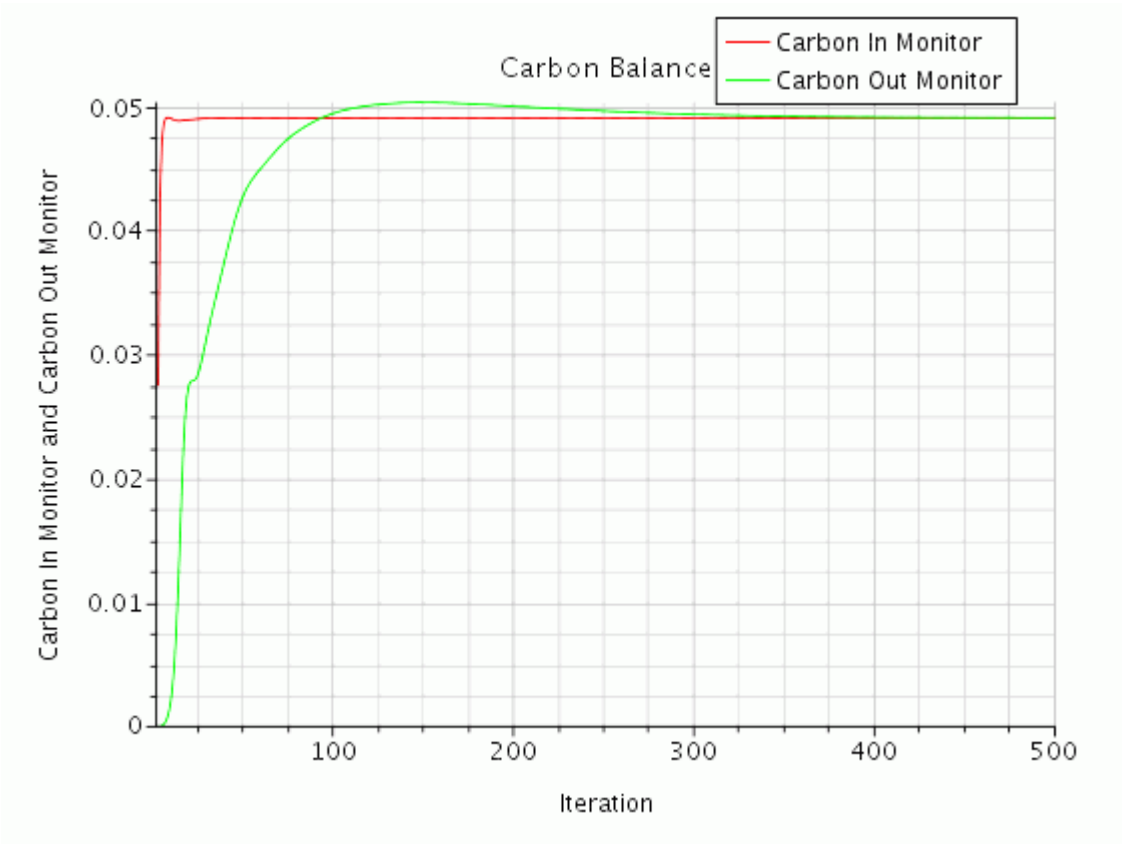
1. To run the simulation, click (**Run**).

The Residuals display is created automatically and shows the progress of the solver. If necessary, click the **Residuals** tab to bring the Residuals plot into view.

The tabs at the top of the **Graphics** window make it possible to select any of the active displays for viewing.

During the run, it is possible to stop the solution by clicking (**Stop**) on the toolbar. If you do halt the simulation, to continue the simulation later, click (**Run**). If left alone, the simulation continues until 500 iterations are complete.

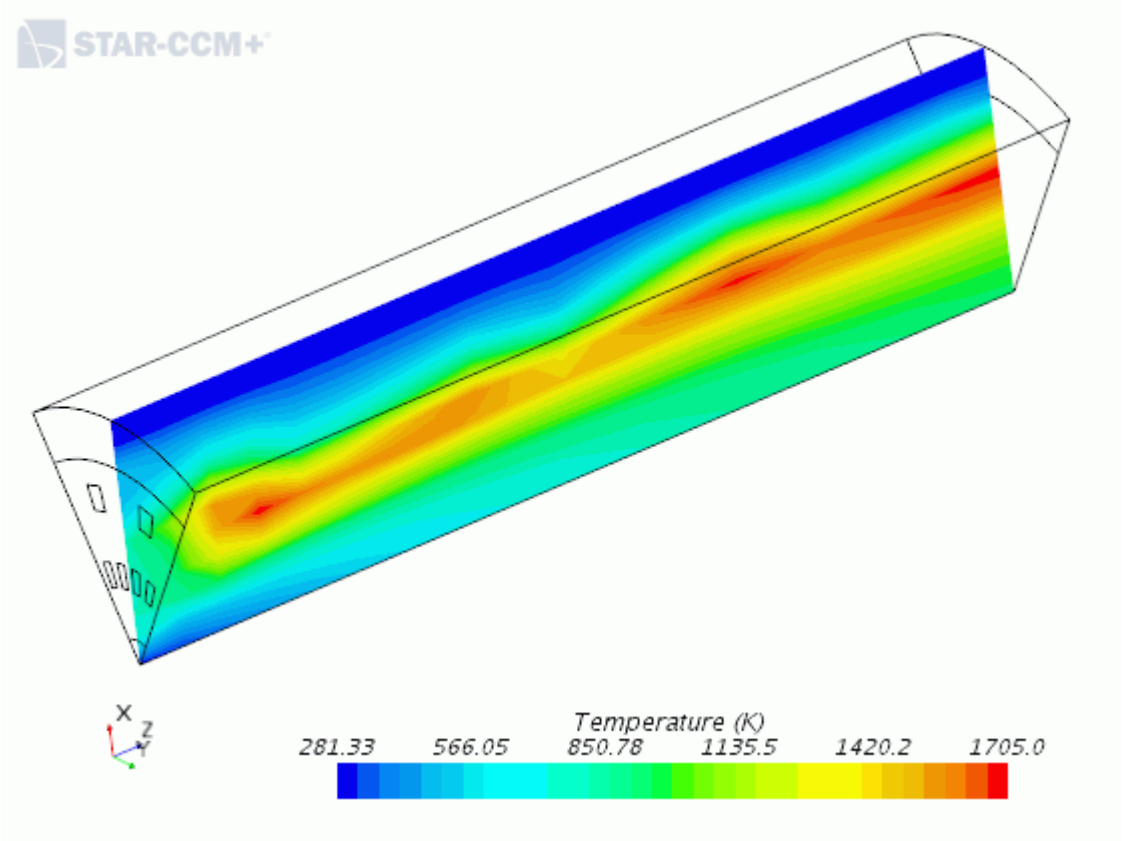
2. After the run has finished, click the **Carbon Balance** tab to make sure that the solution has indeed converged.



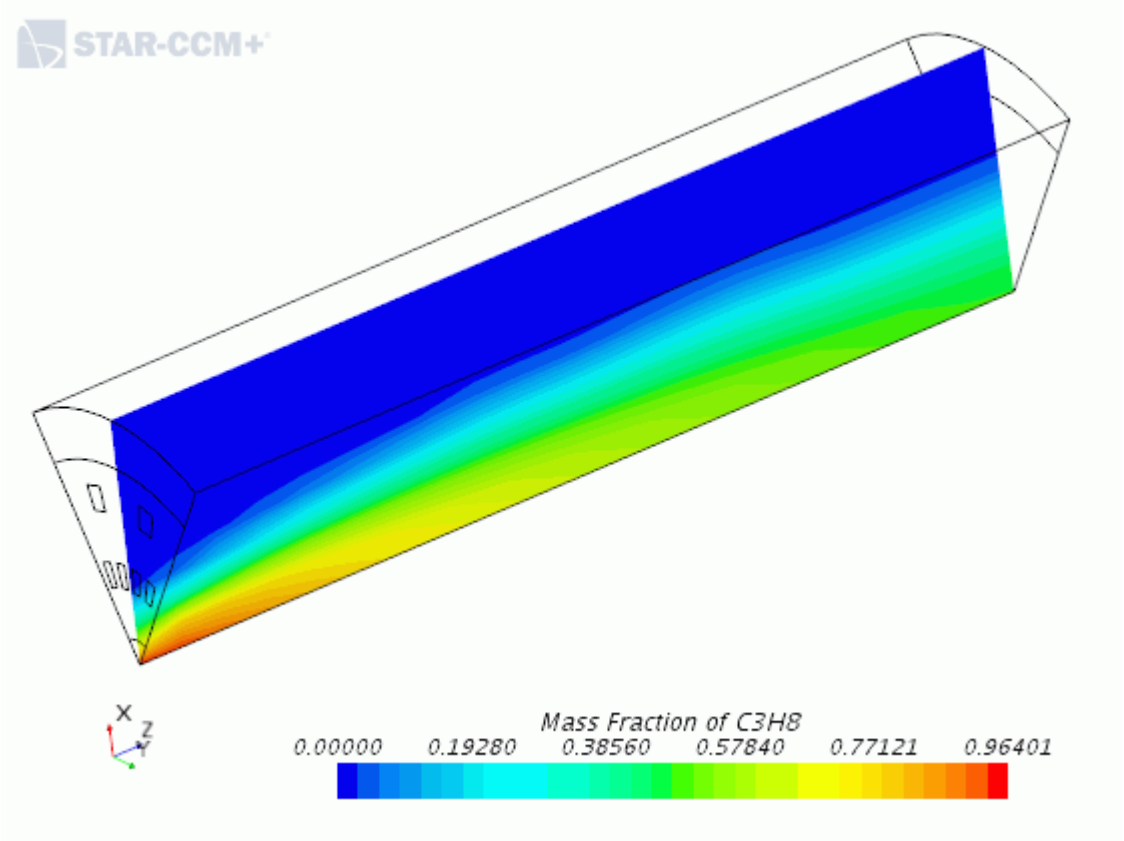
3. Save the simulation.

Visualizing the Results

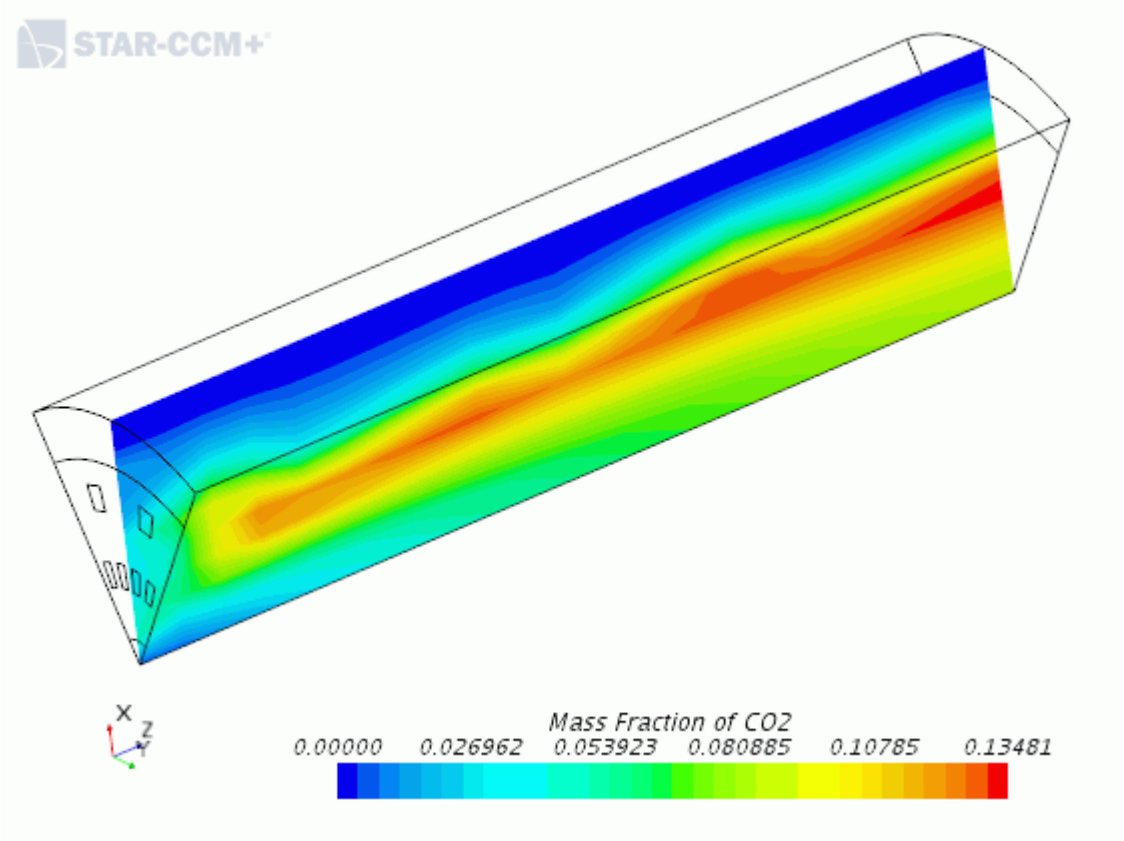
1. To view the temperature profile for the converged solution, select the *Scalar Scene 1* display.



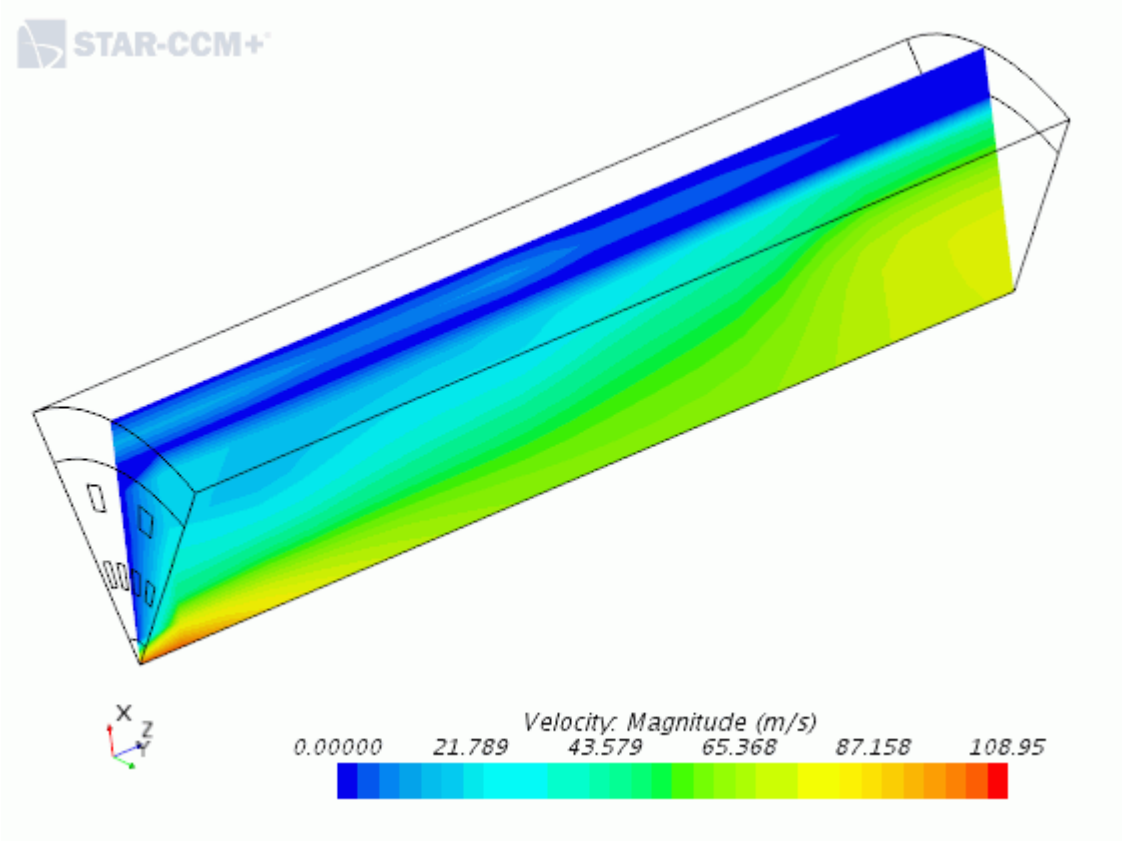
2. In the *Scalar Scene 1* display, right-click the scalar bar and select **Mass Fraction** > **C3H8**.



3. In the *Scalar Scene 1* display, right-click the scalar bar and select **Mass Fraction** > **CO2**.



4. In the *Scalar Scene 1* display, right-click the scalar bar and select **Velocity** > **Magnitude**.



Summary

This tutorial introduced the following STAR-CCM+ features:

- Defining models for Eddy Break-Up combustion.
- Defining material properties that are required for multi-component gases.
- Defining chemical reactions.
- Setting initial conditions and reference values.
- Creating interfaces.
- Defining boundary conditions.
- Running the solver for a set number of iterations.
- Analyzing the results.

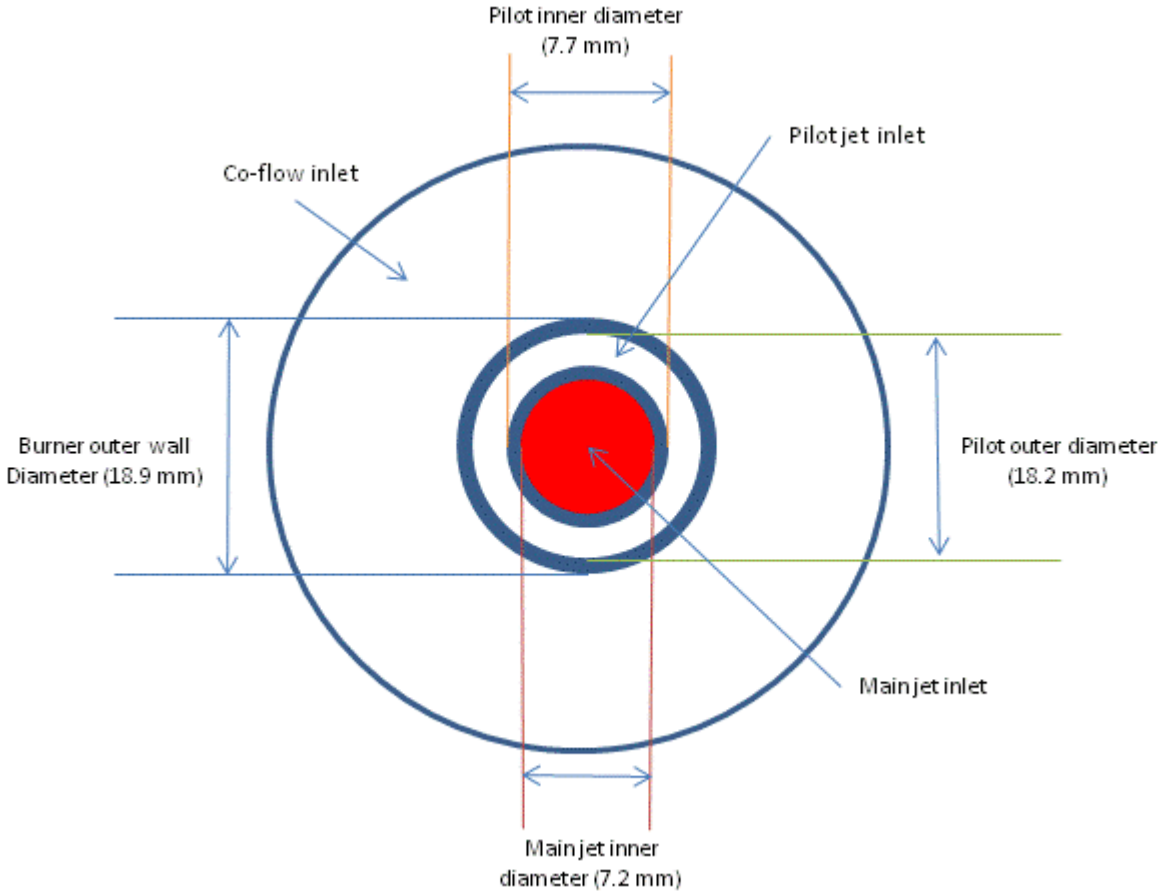
Complex Chemistry: Methane-Air Jet Flame

This tutorial demonstrates how to set up and solve a complex chemistry simulation to model finite-rate chemistry in a flame.

The Complex Chemistry model in STAR-CCM+ uses a detailed mechanism and a stiff ODE solver to capture the finite-rate kinetics. As solving detailed mechanisms is computationally expensive, STAR-CCM+ provides the options of using Clustering and ISAT to reconcile this expense. Clustering groups cells with similar thermal and chemical states before integrating the averaged state for the group. The reaction mapping is then interpolated back to the cells assuming the net reaction rate of all cells in the cluster is the same as the cluster average. Clustering usually provides a substantial speed-up as the number of clusters is less than the number of cells in the simulation. In this tutorial, you use clustering—which is activated by default. The chemistry computation with clustering is reduced to a few thousand points—independent of the number of cells in the simulation. Since a small and simple mesh is used in this tutorial, the speed-up from clustering is not substantial. However, for larger meshes in industrial simulations, the clustering performance increases with the number of cells.

When using the Complex Chemistry model, there are two solvers available to choose from. In this tutorial, the CVODE solver is used with a Chemkin mechanism to calculate the reaction rates and solve the complex chemistry calculations.

The Sandia piloted methane-air jet flame D [806] is simulated in this tutorial.



The two-dimensional axisymmetric geometry consists of the main jet with nozzle diameter of 7.2 mm burning a premixture of 25% methane and 75% dry air by volume. The pilot inlet surrounds the nozzle. The pilot nozzle has an outer diameter of 18.2 mm and is composed of a burnt stoichiometric mixture of the inlet jet and air.

The setup is simplified by using a constant velocity profile at the boundary where gas exits the main and pilot jet nozzles into the domain. The constant velocity and turbulence profiles at the nozzle exits is prone to flame blow-off. You can obtain a more accurate simulation by resolving the flow in the nozzles—by including an upstream mesh, or specifying a boundary layer profile for velocity, turbulent kinetic energy, and turbulent dissipation rate at the exit of the jet nozzle.

Contents:

Prerequisites
Importing the Mesh and Naming the Simulation
Setting up the Models
Importing the Complex Chemistry Definition
Setting Simulation Conditions and Controls
Preparing Analysis Objects
Running the Simulation and Reviewing the Results
Bibliography

Prerequisites

The instructions in the Complex Chemistry: Methane-Air Jet Flame tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

See the User Guide for more information on setting up complex chemistry simulations.

Importing the Mesh and Naming the Simulation

To set up the STAR-CCM+ simulation, launch a simulation and import the supplied volume mesh.

1. Launch STAR-CCM+.
2. Start a new simulation.
3. Select **File** > **Import** > **Import Volume Mesh**.
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/reactingFlow/data`.

5. Select `sandia.ccm` then click **Open**.

   STAR-CCM+ provides feedback on the import process in the *Output* window and a mesh region is created under the **Regions** node.

6. Save the simulation as `MethaneAirJetFlame.sim`.

Setting up the Models

The Complex Chemistry model is used with the Segregated Flow model to simulate the turbulent reacting flow of a compressible multi-component gas.

1. For the physics continuum, **Continua** > **Physics 1**:
   a) In the *Enabled Models* group box, deactivate **Two Dimensional**.
   b) Make sure that **Auto-select recommended models** is activated and select the following models in order:

Group Box	Model
Space	**Axisymmetric**
Material	**Multi-Component Gas**
Reaction Regime	**Reacting**
Reacting Flow Models	**Reacting Species Transport**
Reacting Species Models	**Complex Chemistry**
Turbulence Chemistry Interactions	**Laminar Flame Concept**
Flow	**Segregated Flow**
Enabled Models	**Gradients** (Selected automatically)
	Segregated Species (Selected automatically)
	Segregated Fluid Enthalpy (Selected automatically)
Equation of State	**Ideal Gas**
Time	**Steady**
Viscous Regime	**Turbulent**
Enabled Models	**Reynolds-Averaged Navier Stokes** (Selected automatically)

2. Deactivate **Auto-select recommended models**.
3. Select the following models in order:

Group Box	Model
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**
K-Epsilon Turbulence Models	**Standard K-Epsilon**
Enabled Models	**Exact Wall Distance** (Selected automatically)

Group Box	Model
K-Epsilon High y+ Wall Treatment	**High y+ Wall Treatment**

4. Click **Close**.
5. Save the simulation.

Importing the Complex Chemistry Definition

The ASCII Chemkin file set consists of a chemical mechanism file, a thermodynamics file and a transport file. The chemical mechanism file contains a list of the species and their reaction rates. The thermodynamic file contains the elemental composition of the species, as well as the thermodynamic properties to calculate the specific heat, enthalpy and entropy. The transport file contains the kinetic theory parameters to calculate molecular transport properties and can be optionally imported.

By default, the Complex Chemistry model uses the CVODE solver to integrate the chemistry over a time-step. However, this process is computationally expensive. To mitigate the expense, STAR-CCM+ automatically activates Clustering on the Chemistry Acceleration node that belongs to the Complex Chemistry model. In Situ Adaptive Tabulation (ISAT) is also available, but is not necessary for the case in this tutorial.

Import the complex chemistry definition [807] from the STAR-CCM+ repository.

1. Right-click the **Models** > **Complex Chemistry** node and select **Import Chemistry Definition (Chemkin format)**.
2. In the Import Chemkin Files dialog, deactivate **Import Fluid Transport Properties File**, and select the following files from the directory, `[INSTALL_DIR]/doc/startutorialsdata/reactingFlow/data/jetFlame/`:

Fluid Chemistry Reaction File	`chem.inp`
Fluid Thermodynamic Properties File	`therm.dat`

The **Multi-Component Gas** > **Gas Components** node is populated with all of the gas species in the mechanism and the **Reacting** > **Reactions** node is populated with all of the reactions in the mechanism.

Setting Simulation Conditions and Controls

In this section, you set the initial conditions, boundary conditions, solver settings, and stopping criteria.

Set the initial conditions for the start of the calculation: temperature, species mass fraction, the turbulence specification method, and the axial velocity of the gas.

1. Expand the **Continua** > **Physics 1** > **Initial Conditions** node and set the following properties:

Node	Property	Setting
Species Mass Fraction	*Value*	**O2**, `0.233`
		N2, `0.767`
Static Temperature	*Value*	`293`K

Node	Property	Setting
Turbulence Specification	*Method*	**K + Epsilon**
Turbulent Dissipation Rate	*Value*	50000 m²/s³
Turbulent Kinetic Energy	*Value*	30 J/kg
Velocity	*Value*	[0.9, 0.0] m/s

Set the boundary conditions for the inlets, outer boundaries, and outlet boundaries. For the outer and outlet boundaries, you set the backflow conditions to the extrapolation method, which extrapolates the flow direction from the interior of the domain. This case requires extrapolation because the flow is not necessarily normal to the boundary.

2. Expand the **Regions** > **fluid** > **Boundaries** node.
3. Edit the **inlet-1** > **Physics Values** node and set the following properties:

Node	Property	Setting
Species Mass Fraction	*Value*	**CH4**, 0.1563
		O2, 0.1965
		N2, 0.6472
Static Temperature	*Value*	293 K
Turbulence Intensity	*Value*	0.1
Turbulent Viscosity Ratio	*Value*	100
Velocity Magnitude	*Value*	49.6 m/s

4. Edit the **inlet-2** > **Physics Values** node and set the following properties:

Node	Property	Setting
Species Mass Fraction	*Value*	**H2O**, 0.095
		CO2, 0.117
		O2, 0.054
		N2, 0.734
Static Temperature	*Value*	1880 K
Turbulence Intensity	*Value*	0.1
Turbulent Viscosity Ratio	*Value*	100
Velocity Magnitude	*Value*	11.4 m/s

5. Edit the **inlet-3** > **Physics Values** node and set the following properties:

Node	Property	Setting
Species Mass Fraction	*Value*	**O2**, 0.233

Node	Property	Setting
		N2, `0.767`
Static Temperature	*Value*	`293` K
Turbulence Intensity	*Value*	`0.1`
Turbulent Viscosity Ratio	*Value*	`100`
Velocity Magnitude	*Value*	`0.9` m/s

6. Select the **outer** > **Physics Conditions** > **Backflow Specification** node and set *Direction* to **Extrapolated**.
7. Edit the **outer** > **Physics Values** node and set the following properties:

Node	Property	Setting
Species Mass Fraction	*Value*	**O2**, `0.233`
		N2, `0.767`
Static Temperature	*Value*	`293` K

8. Right-click the **outer** node, select **Copy**, then right-click the **outlet** node and select **Paste**.

   This action copies the settings from the outer node to the outlet node.

The default under-relaxation factors for the flow, species, energy, and turbulence equations are suitable. To ensure that the two solutions remain synchronized, it is important that the under-relaxation factors for the species and energy equations are the same.

9. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to `2200`.

   This value is sufficient to achieve a steady-state solution.
10. Save the simulation.

Preparing Analysis Objects

Create reports to monitor the axial temperature distribution and the mole fraction of CO in the fluid region.

- Create a plot for the axial temperature distribution.
- Create a plot for the mole fraction of CO.
- Create a plot to show the heat balance—which gives a good indication of convergence.

1. To plot the unnormalized residuals for each species, expand the **Monitors** node and multi-select all species monitor nodes.
2. In the *Properties* window, set *Normalization Option* to **Off**.

Create a plot for the axial temperature distribution:

3. Right-click the **Plots** node and select **New Plot** > **XY Plot**.
4. Rename the **Plots** > **XY Plot 1** node to `TAxis`.
5. Select the **TAxis** node and set the following properties:

Node	Property	Setting
TAxis	*Title*	**TAxis**
	Parts	**axis**
└ **Y Types** > **Y Type 1** > **Scalar Function**	*Field Function*	**Temperature**

Create a plot for the mole fraction of CO:

6. Right-click the **Plots** node and select **New Plot** > **XY Plot**.
7. Rename the **Plots** > **XY Plot 1** node to `Mole Fraction CO`.
8. Select the **Mole Fraction CO** node and set the following properties:

Node	Property	Setting
TAxis	*Title*	**Mole Fraction CO**
	Parts	**axis**
└ **Y Types** > **Y Type 1** > **Scalar Function**	*Field Function*	**Mole Fraction of CO**

Create a report and plot for the heat balance:

9. Right-click the **Reports** node and select **New Report** > **Heat Transfer**.
10. Rename the **Reports** > **Heat Transfer 1** node to `Heat Balance`.
11. Select the Heat Balance node and set parts to **inlet-1**, **inlet-2**, **inlet-3**, **outer**, and **outlet**.
12. Right-click the **Heat Balance** node and select **Create Monitor and Plot from report**.
13. Save the simulation.

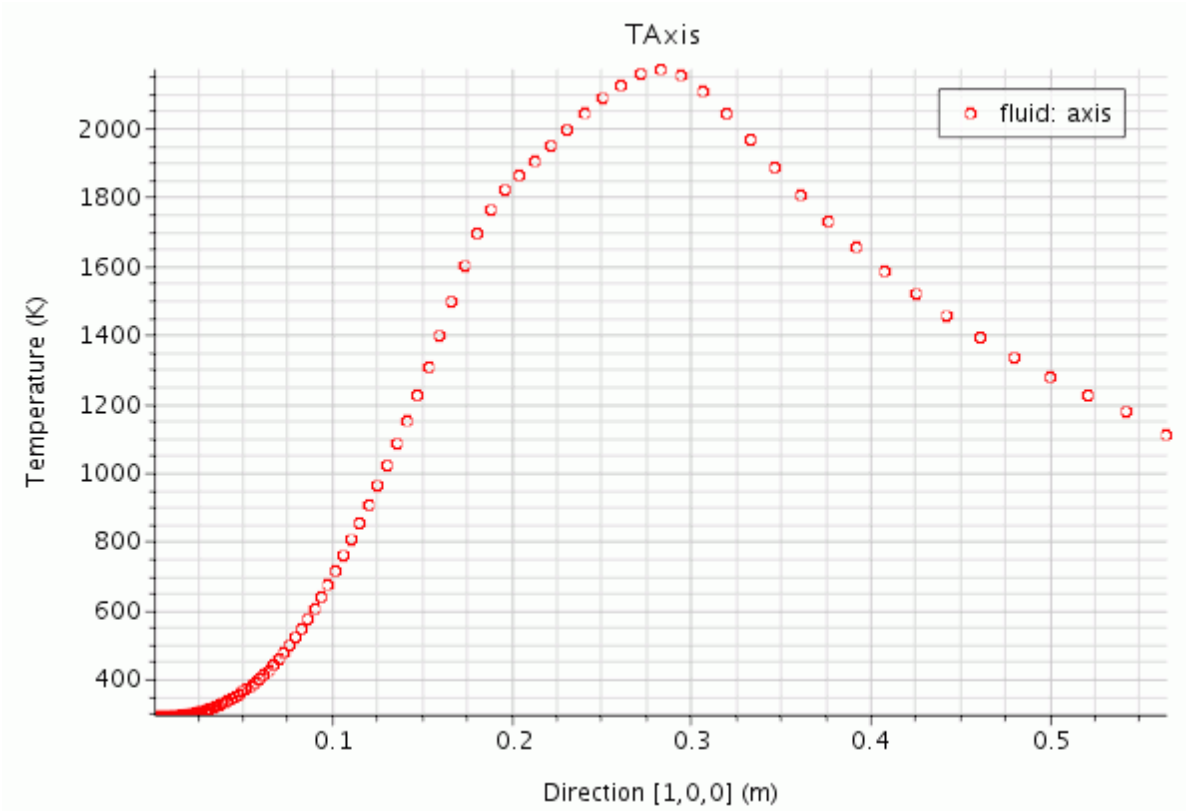
Running the Simulation and Reviewing the Results

The simulation is ready to run and you can review the results as the simulation is running.

1. To run the simulation, click (**Run**).

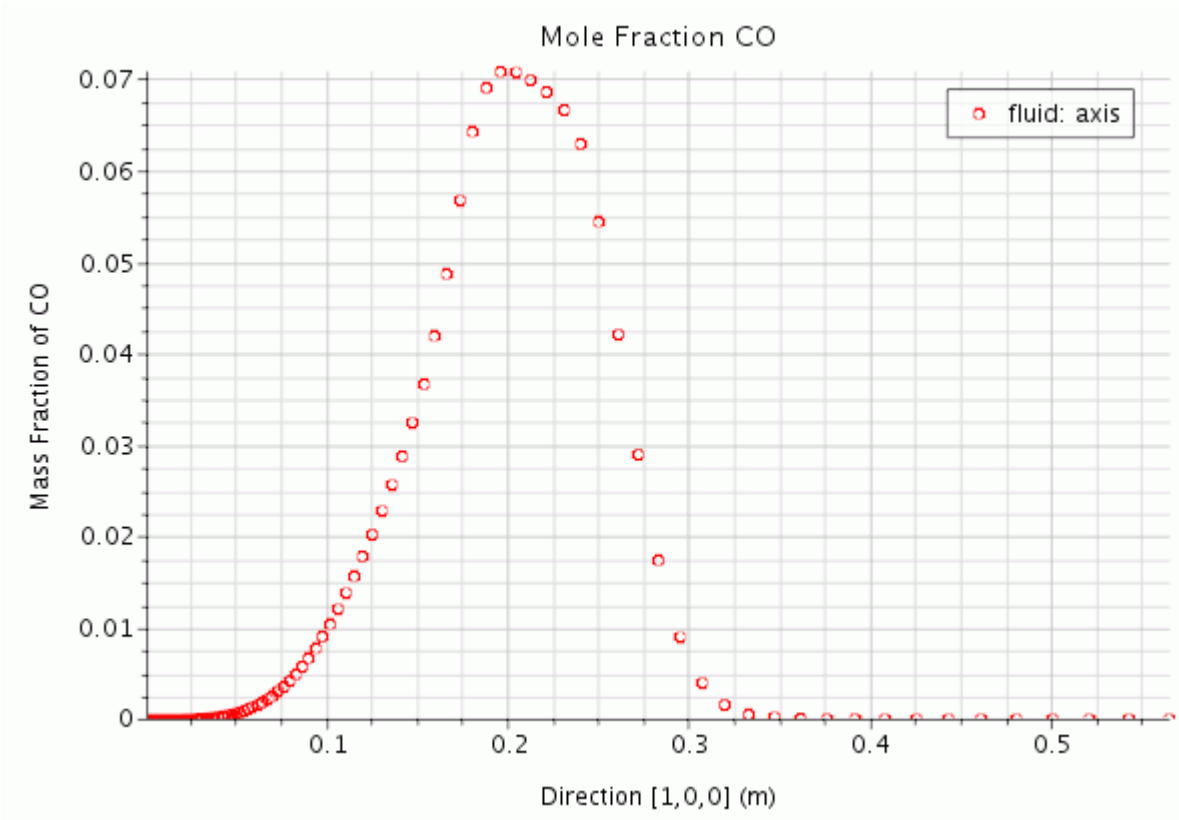
You can monitor the iteration progress in the *Output* window. The *Residuals* display is created automatically and shows the progress of the solver. If necessary, click the *Residuals* tab to bring the *Residuals* plot into view.

2. Save the simulation.
3. To view each of the following plots, click each tab:
   - *TAxis*:



The temperature is greatest at around 0.28 m along the X-axis of the geometry.

- *Mole Fraction CO*:



The mole fraction of CO is greatest around 0.2 m along the X-axis of the geometry. In the jet flame, the fuel mixture is rich to begin with and incomplete combustion occurs—which results in CO production instead of CO2:

$$CH_4 + 1\tfrac{1}{2}O_2 \rightarrow CO + 2H_2O$$

At 0.3 m along the X-axis of the geometry, the level of CO has dropped close to zero. As the jet flame develops, the fuel mixture becomes leaner and complete combustion occurs:

$$CH_4 + 2\,O_2 \rightarrow CO_2 + 2H_2O$$

Leaner combustion is associated with a hotter flame—which correlates with the previous plot which shows that the maximum temperature also occurs at 0.3 m along the X-axis.

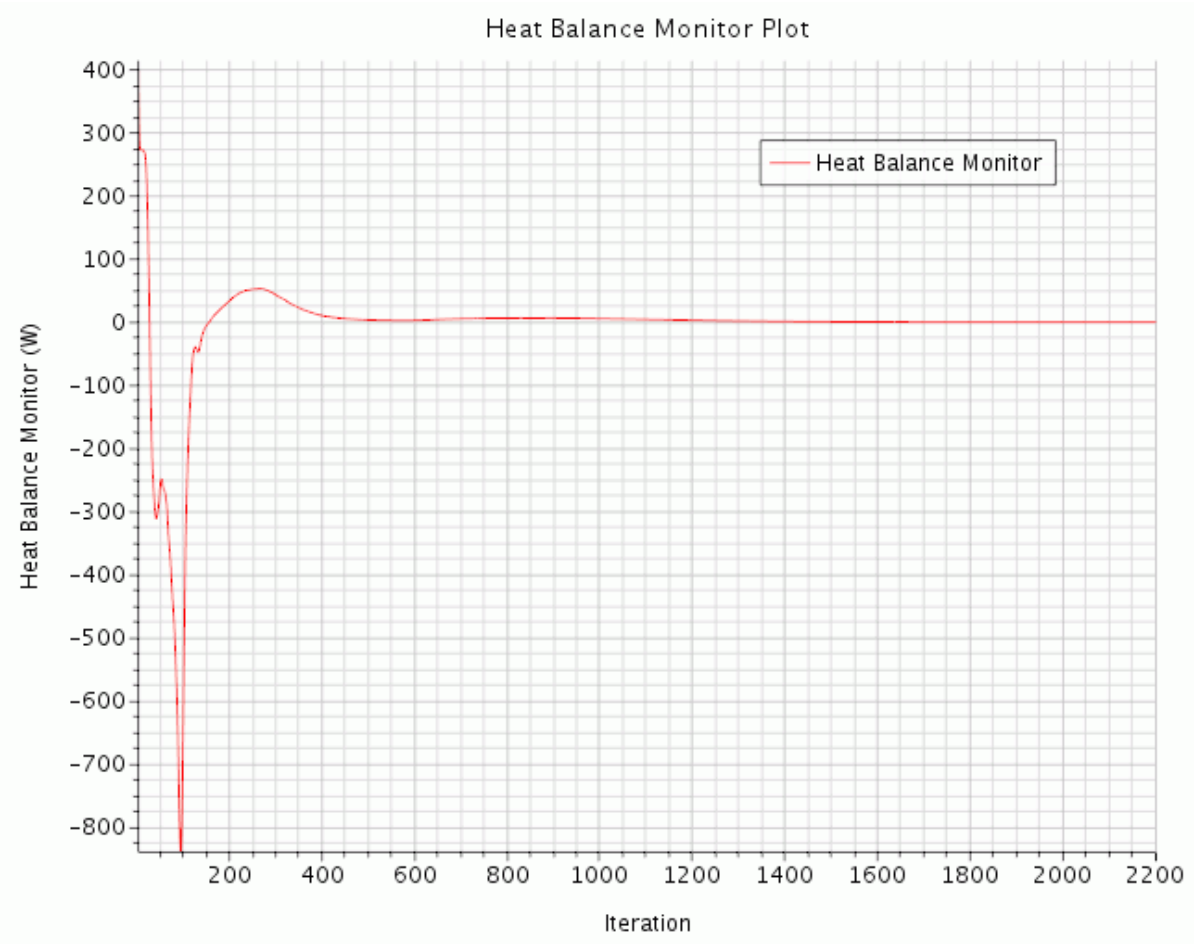
In lean flames, the CO is converted to CO2:

$$CO + \tfrac{1}{2}\,O_2 \rightarrow CO_2$$

Therefore, the mole fraction of CO after 0.56 m along the axis of the geometry is reduced to 1.47 E-8.

- *Heat Balance Monitor Plot*

  The heat balance plot monitors the balance of heat between all inlet boundaries and all outlet boundaries. Monitoring the heat balance provides a good indication of convergence. When the heat from the inlet boundaries is the same as the heat at the outlet boundaries, the value for the heat balance settles close to zero.



Bibliography

[806] *[411] Masri, A. R., Dibble, R. W., and Barlow, R. S. 1996 "The structure of turbulent nonpremixed flames revealed by Raman-Rayleigh-LIF measurements", Prog. Energy Combust. Sci. 22 (4) pp. 307-362.*

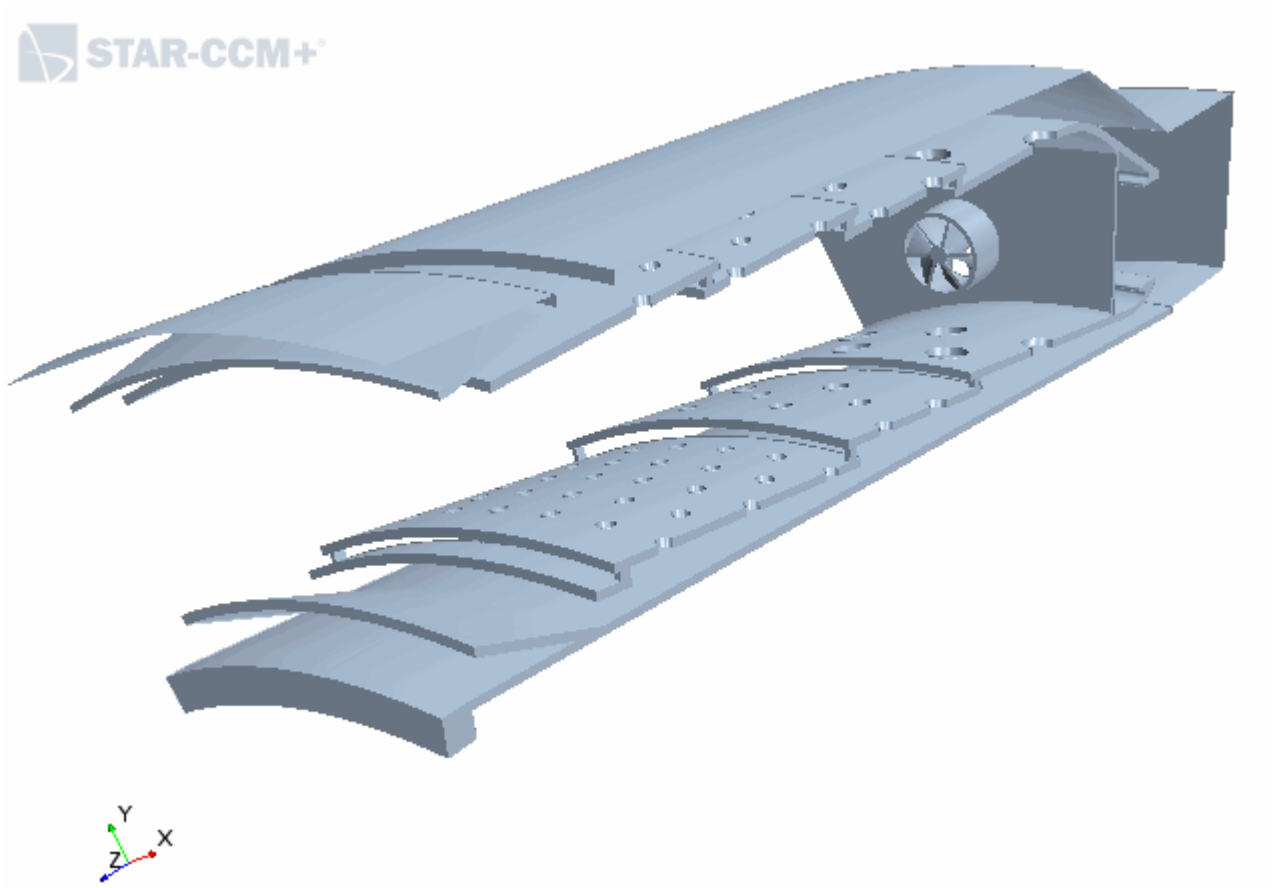
[807] *T.F. Lu and C.K. Law, "A criterion based on computational singular perturbation for the identification of quasi steady state species: A reduced mechanism for methane oxidation with NO chemistry," Combustion and Flame, Vol.154 No.4 pp.761-774, 2008.*

Spray Combustion: Jet A Fuel

STAR-CCM+ provides flamelet models which you can use to model combustion reactions that occur within gas combustors. These reactions are often modeled based on obtaining an infinitely fast chemical equilibrium.

In this tutorial, a spray combustion case is set up using the Non-Adiabatic Chemical Equilibrium flamelet model for non-premixed flames. The fuel is Jet A and is injected into the combustor in the form of droplets that then evaporate into combustible vapor.

The Chemical Equilibrium model in STAR-CCM+ assumes that the chemistry is infinitely fast and thermo-chemical equilibrium is reached instantaneously. Since the chemical equilibrium is a thermodynamic property and independent of the chemical mechanism, this tutorial simulation does not require a complex chemical mechanism. Instead, you define the chemical components of the reaction within a Chemical Equilibrium table.



Contents:

Loading the Starting File and Creating the Mesh
Defining the Physics and Chemistry
Setting up the Lagrangian Phase for the Droplets
Setting up Post-Processing

Running the Simulation
Reviewing the Results
Bibliography

Loading the Starting File and Creating the Mesh

You start this tutorial by loading a sim file in which a mesh, that is suitably refined for the purpose of this tutorial is ready to run. The mesh is created for the fluid volume within the combustor.

Note: Before starting this tutorial, make sure that you have at least 2 processes available to use.

1. Open STAR-CCM+ and select **Load Simulation**.
2. In the *Load Simulation* dialog:
   a) Set *Process Options* to **Parallel on Local Host** and *Compute Processes* to `2` or more—depending on the configuration of the machine that you are using.
   b) Click **Browse**.
   c) In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/reactingFlow/data`.
   d) Select `GeneralCombustor_StartingFile.sim` then click **Open**.
   e) Click **OK**.

Before generating the mesh, since the geometry in this tutorial represents a segment of a circular combustor, you need to define a periodic interface at each side of the fluid domain.

3. Open **Geometry Scene 1**.
4. To create an interface:
   a) Expand the **Regions** > **Fluid Volume** > **Boundaries** node.
   b) Multi-select the **Periodic1** and **Periodic2** nodes.
   c) Right-click one of the selected nodes and select **Create Interface**.
5. To define the interface as periodic, select the **Interfaces** > **Interface 1** node and set *Topology* to **Periodic**.

Now, you can generate the mesh.

6. From the main toolbar, select **Mesh** > **Generate Volume Mesh**.
7. Create a Mesh Scene.
8. To help visualize the mesh, click (**Make Scene Transparent**).
9. Save the file as `JetA_Combustor.sim`.

Defining the Physics and Chemistry

In this section, you select appropriate flow and combustion models and then set up the chemistry table that is used for the combustion reactions.

1. Right-click the physics continuum, **Fluid Volume**, choose **Select Models**, and select the following models in order. Make sure that **Auto-select recommended models** is activated.

Group Box	Model
Space	**Three Dimensional** (previously selected)
Time	**Steady**
Material	**Multi-Component Gas**
Reaction Regime	**Reacting**
Reacting Flow Models	**Flamelet**
Flamelet Models	**Chemical Equilibrium**
Enabled Models	**Ideal Gas** (selected automatically)
	Non-Adiabatic (selected automatically)
Flame Type	**Non-Premixed Flame**
Flow	**Segregated Flow**
Enabled Models	**Gradients** (selected automatically)
	Turbulent (selected automatically)
	Segregated Fluid Enthalpy (selected automatically)
	Reynolds-Averaged Navier-Stokes (selected automatically)
Reynolds-Averaged Turbulence	**K-Omega Turbulence**
Enabled Models	**SST (Menter) K-Omega** (selected automatically)
	Exact Wall Distance (selected automatically)
	All y+ Wall Treatment (selected automatically)
Optional Models	**Lagrangian Multiphase**
	NOx Emission
Specific NOx Models	**NOx Thermal**
Enabled Models	**NOx Zeldovich** (selected automatically)
Optional Models	**Radiation**
Radiation	**Participating Media Radiation (DOM)**
Radiation Spectrum (Participating)	**Gray Thermal Radiation**

2. Click **Close**.

In this tutorial, the simulation is run for a shorter number of iterations than would be used for a typical industrial case. Therefore, you also reduce the iteration at which STAR-CCM+ begins solving for NOx.

3. Select the **Models** > **NOx Emission** node and set *Begin* to `200`.

After selecting the appropriate physics models, you define the chemistry of the combustion, the composition of the fluid streams, and generate the chemical equilibrium table that the Chemical Equilibrium model requires.

4. To create the chemistry definition:
   a) Right-click the **Fluid Volume** > **Chemical Equilibrium Table Generator** > **Chemistry Definition** node and select **Select Mixture Components**.
   b) In the *Select Mixture Components* dialog, select the following components from the Standard material database:
      - C(S) (C(S))
      - C2H2 (Acetylene)
      - C12H23 (Jet-A)
      - CH (CH)
      - CH2 (CH2)
      - CO (Carbon Monoxide)
      - CO2 (Carbon Dioxide)
      - H (H)
      - H2 (Hydrogen)
      - H2O (Water)
      - N2 (Nitrogen)
      - O (O)
      - O2 (Oxygen)
      - OH (OH)
   c) Click **Apply** then click **Close**.
5. Expand the **Chemical Equilibrium Table Generator** > **Fluid Streams** node and set the following *Properties*:

Fuel	*Composition Specification*	**Mole Fraction**
	Fluid Stream Components	C12H23: `1.0`
	Temperature	`300.0 K`
Oxidizer	*Composition Specification*	**Mole Fraction**
	Fluid Stream Components	O2: `0.209`
		N2: `0.791`
	Temperature	`500.0 K`

To save computational memory and disk space, only species that are defined as species for post processing are stored in the flamelet table. By default, the species that are defined in the fluid streams are made available for post-processing. When the Radiation model is activated, CO2 and H2O are also included. You can also include other species that you are interested in monitoring—for example, carbon monoxide emissions.

6. To include CO for post-processing, select the **Chemical Equilibrium Table Generator** > **Parameters** > **Species for Tabulation** node and in addition to the current selection of *Species*, select **CO**.
7. To generate the chemical equilibrium table, right-click the **Chemical Equilibrium Table** node and select **Generate Chemical Equilibrium and Construct Table**.

   A progress bar appears below the tree.
8. Click **Save**.

Setting up the Lagrangian Phase for the Droplets

1. Within the **Continua** > **Fluid Volume** models, right-click the **Lagrangian Multiphase** > **Lagrangian Phases** node and select **New**.
2. Right-click the **Lagrangian Phases** > **Phase 1** > **Models** node and select **Select models**.
3. In the *Phase 1 Model Selection* dialog, select the following models in order. Make sure that **Auto-select recommended models** is activated.

Group Box	Model
Enabled Models	**Residence Time** (selected automatically)
Particle Type	**Material Particles**
Enabled Models	**Pressure Gradient Force** (selected automatically)
	Spherical Particles (selected automatically)
Material	**Liquid**
Equation of State	**Constant Density**
Mass Transfer	**Quasi-Steady Evaporation**
Enabled Models	**Energy** (selected automatically)
Track Sampling	**Track File**
Optional Particle Forces	**Drag Force**
Optional Models	**Turbulent Dispersion**
	Two-Way Coupling
	Parcel Depletion

4. Click **Close**.
5. Right-click the **Phase 1** > **Models** > **Liquid** > **H2O** node and select **Replace with**.
6. In the *Replace Material* dialog, select **C12H23 (Jet-A)** and click **OK**.
7. Expand the **Phase 1** > **Models** node and set the following *Properties*:

Node	Property	Setting
Quasi-Steady Evaporation	*Vapor Component*	**C12H23**
Track File	*Scalars*	In addition to default selections, select:

Node	Property	Setting
		• **Droplet.Latent Heat of Vaporization** • **Droplet.Saturation Pressure** • **Parcel.Mass Flow Rate** • **Particle.Density** • **Particle.Diameter** • **Particle.Mass** • **Particle.Temperature**
	Vectors	In addition to default selections, select, **Particle.Velocity**
Turbulent Dispersion	*Exact Eddy Interaction Time*	activated

Next, you adjust the Material Properties of C12H23.

8. Expand the **Liquid** > **C12H23** > **Material Properties** node.
   a) Set the following *Properties*:

Node	Property	Setting
Density		
⌊ **Constant**	*Value*	`796.0 kg/m³`
Latent Heat of Vaporization	*Method*	**Constant**
⌊ **Constant**	*Value*	`372000.0 J/Kg`
Saturation Pressure	*Method*	**Polynomial in T**
Specific Heat	*Method*	**Polynomial in T**

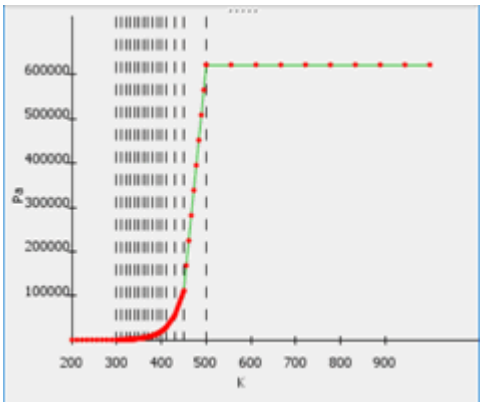
   b) Select the **Saturation Pressure** > **Polynomial in T** node, next to the *Polynomial* property, click ... (**Custom Editor**), and in the *Polynomial in T - Polynomial* dialog, set the following:

Note: You can copy and paste the values.

- *Use Real Exponents*: activated
- *Number of Intervals*: `16`
- *Interval Ranges*: `[200.0, 300.0, 310.0, 320.0, 330.0, 340.0, 350.0, 360.0, 370.0, 380.0, 390.0, 400.0, 410.0, 430.0, 450.0, 500.0, 1000.0]`
- *Number of Coefficients*: `[1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1]`
- *Coefficients*: `[642.0, 26.3, -7248.0, 37.2, -10627.0, 52.4, -15491.0, 73.9, -22586.0, 104.3, -32922.0, 147.1, -47902.0, 207.4, -69610.0, 292.6, -101134.0, 412.6, -146734.0, 582.0, -212800.0,`

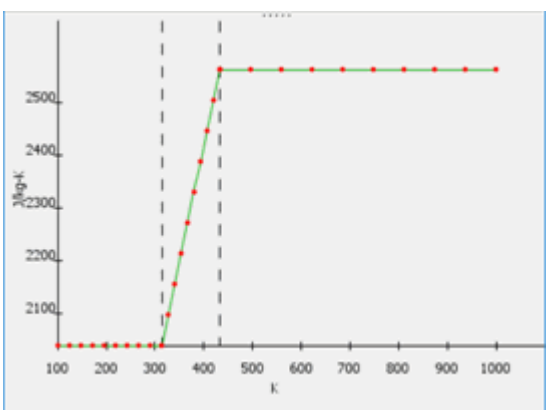
820.9, -308360.0, 1395.2, -543823.0, 2774.35, -1136857.5, 10214.0, -4484700.0, 622300.0]

- *Exponents*: [0.0, 1.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0, 0.0, 0.0]



c) Select the **Specific Heat** > **Polynomial in T** node, next to the *Polynomial* property, click ... (**Custom Editor**), and in the *Polynomial in T - Polynomial* dialog, set the following:

- *Use Real Exponents*: activated
- *Number of Intervals*: 3
- *Interval Ranges*: [100.0, 313.0, 433.0, 1000.0]
- *Number of Coefficients*: [1, 2, 1]
- *Coefficients*: [2038.0, 4.375, 668.625, 2563.0]
- *Exponents*: [0.0, 1.0, 0.0, 0.0]



The Lagrangian phase is injected into the main Fluid Volume by an injector. Before running the simulation, you create an injector in the Fluid Volume through which the Lagrangian phase is introduced.

9. In the main simulation tree, right-click the **Injectors** node and select **New**.
10. Select the **Injectors** > **Injector 1** node and set the following *Properties*:

Property	Setting
Lagrangian Phase	**Phase 1**
Type	**Hollow Cone Injector**
Inputs	**Fluid Volume**

11. Expand the **Injector 1** > **Values** node and set the following *Properties*:

Node	Property	Setting
Injector Diameter	*Value*	`0.005 m`
Inner Cone Angle	*Value*	`10 deg`
Outer Cone Angle	*Value*	`30 deg`
Parcel Streams	*Value*	`100`
Axis	*Origin*	`[0.0426, 0.1595, 0.01276]`
Mass Flow Rate	*Value*	`1.0E-4 kg/s`
Particle Diameter	*Value*	`3.0E-5 m`
Velocity Magnitude	*Value*	`5.0 m/s`

Setting up Post-Processing

Before running the simulation, you prepare scenes and plots to display the results.

1. Create a scalar scene to view the temperature on a cross-section of the Fluid Domain which intersects the inlet boundary.
   a) Create a scalar scene and rename it to `Temperature`.
   b) Expand the **Temperature** > **Displayers** > **Scalar 1** node and set the following *Properties*:

Node	Property	Value
Scalar 1	*Contour Style*	**Smooth Blended**
⌊ **Parts**	*Parts*	**Derived Parts** > **Plane Section**
⌊ **Scalar Field**	*Function*	**Temperature**

2. Create a scene to visualize the 3D flame using volume rendering.
   a) Create a scalar scene and re-name it to `3D Flame`.
   b) At the top of the object tree panel, click the *Scene/Plot* tab.
   c) Right-click the **3D Flame** > **Displayers** node and select **New Displayer** > **Geometry**.
   d) Expand the **Displayers** > **Geometry 1** node and set the following *Properties*:

Node	Property	Setting
Geometry 1	*Surface*	activated

Node	Property	Setting
	Color Mode	**Constant**
	Representation	**Geometry**
⌊ **Parts**	*Parts*	• **Parts** > **Exterior** • **Parts** > **Interior** • **Parts** > **Liner**

e) To toggle off the outline, double-click the **Displayers** > **Outline 1** node.

f) At the top of the object tree panel, click the *Simulation* tab.

g) With the *3D Flame* scene active in the *Graphics* window, in the main simulation tree, right-click the **Derived Parts** node and select **New Part** > **Resampled Volume**.

h) To confirm the default volume selection and settings, click **Create** then **Close**.

i) At the top of the object tree panel, click the *Scene/Plot* tab. Expand the **3D Flame** > **Displayers** > **Scalar 1** node and set the following Properties:

Node	Property	Setting
Scalar 1	*Opacity*	`0.1`
⌊ **Parts**	*Parts*	**Derived Parts** > **Resampled Volume**
⌊ **Scalar Field**	*Function*	**Temperature**
⌊ **Color Bar**	*Color Map*	**gray-darkred**
	Reverse	activated
⌊ **Volume Rendering Settings**	*Lighting Mode*	**Local Lighting**
	Quality	`1.0`

3. At the top of the object tree panel, click the *Simulation* tab.

You can use the many reports, monitors, and plots that are available to display the output from the simulation and assess how well the simulation is converging. For example, comparing the mass flow at the inlet with the mass flow at the outlet shows if the overall mass balance is equal—the overall mass balance equalizes as the simulation converges. The mass flow averaged reports also stabilise as the simulation reaches convergence. In industry, it is important to monitor NOx emissions from combustors. This tutorial provides custom field functions which you use to calculate the NOx emissions (dry emissions corrected at 15% O2 reference, [808]).

4. in the main simulation tree, right-click the **Reports** node and select **New Report** > **Mass Flow Averaged**. Then repeat the process to create one more mass flow averaged report, two mass flow reports, and one expression report.
5. Rename the Reports as follows:

Mass Flow Averaged 1	`Mass Flow Averaged Temperature`
Mass Flow Averaged 2	`Mass Flow Averaged NOx (ppmv)`
Mass Flow 1	`Mass Flow Inlet`

Mass Flow 2	`Mass Flow Outlet`
Expression 1	`Mass Flow Imbalance`

6. Set the following *Properties*:

Node	Property	Setting
Mass Flow Averaged Temperature	*Field Function*	**Temperature**
	Parts	**Regions** > **Fluid Volume** > **Outlet**
Mass Flow Averaged NOx (ppmv)	*Field Function*	**NOx (ppmv)**
	Parts	**Regions** > **Fluid Volume** > **Outlet**
Mass Flow Inlet	*Parts*	**Regions** > **Fluid Volume** > **Inlet**
Mass Flow Outlet	*Parts*	**Regions** > **Fluid Volume** > **Outlet**
Mass Flow Imbalance	*Definition*	`abs(${MassFlowOutletReport})-abs(${MassFlowInletReport})`

7. To create monitors and plots from the reports:
   a) Multi-select the following Report nodes:
      - **Mass Flow Averaged Temperture**
      - **Mass Flow Averaged NOx (ppmv)**
   b) Right-click one of the selected nodes and select **Create Monitor and Plot from Report**.
   c) In the *Create Plot From Reports* dialog, click **Multiple Plots (one per report)**.
   d) Right-click the **Reports** > **Mass Flow Imbalance** node and select **Create Monitor and Plot from Report**
8. Save the Simulation.

Running the Simulation

In this section, you adjust the solver settings and stopping criteria before running the simulation.

1. To adjust the solver settings, expand the **Solvers** node and set the following *Properties*:

Node	Property	Setting
Lagrangian Multiphase		
∟ **Steady**	*First Iteration*	`10`
	Update Frequency	`5`
∟ **Two-Way Coupling**	*Under-Relaxation Factor*	`0.5`

Node	Property	Setting
Segregated Flow		
∟ **Velocity**	*Under-Relaxation Factor*	`0.8`
Segregated Energy	*Fluid Under-Relaxation Factor*	`0.8`
DO Radiation		
∟ **Update Frequency** > **Steady**	*Iteration Update Frequency*	`10`
PPDF Combustion	*Under-Relaxation Factor*	`0.8`

In industrial cases, to relieve any instabilities, you can reduce the maximum and minimum courant numbers for the Lagrangian Multiphase solver.

2. To define the stopping criteria, select the **Stopping Criteria** > **Maximum Steps** node and set **Maximum Steps** to `500`.
3. **Save** the Simulation.
4. **Run** the Simulation.
5. When the Simulation is complete, click **Save**.

Reviewing the Results

1. To improve the residuals, you can turn off the normalization for the monitors.
   a) Right-click the **Tools** > **Filters** node and select **New**.
   b) Select and drag the **Monitors** > **Continuity** node into the *New Filter* panel.
   `Type is Residual Monitor` appears in the *Filter* panel.
   c) Rename the **Tools** > **Filters** > **Filter** node to `Residual Monitors`.
   d) Right-click the **Residual Monitors** node and select **New Custom Tree**.
   e) In the Residual Monitors custom tree, select all nodes and set *Normalization Option* to **Off**.
   f) Close the *Residual Monitors* windows.

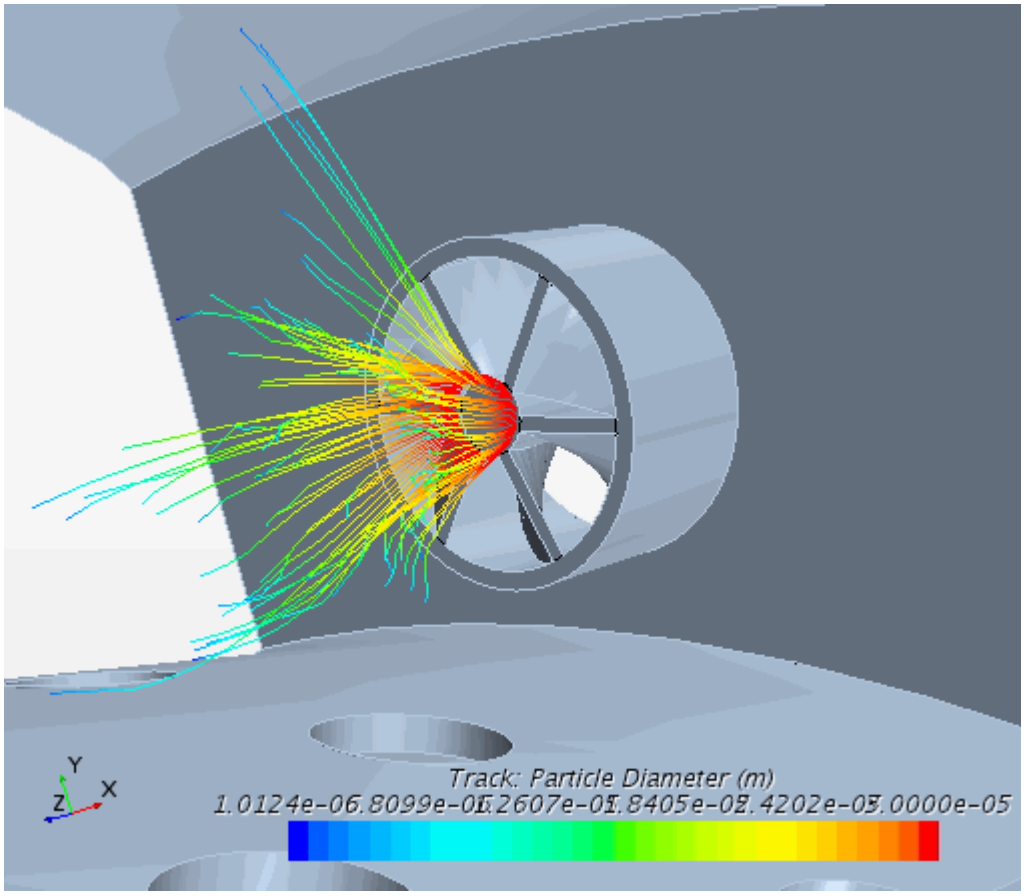
Jet A is a liquid fuel which enters the combustor as a spray of droplets which then evaporate before they combust. To optimize the efficiency of combustion, it is important to check that the fuel is evaporating effectively after entering the combustor. To visualize the droplets evaporating, you can track the particle diameter of the Jet A fuel droplets in a scalar scene.

2. Load the Track Files for the Lagrangian Phase.
   a) Right-click the **Tools** > **Track Files** node and select **Track File**.
   b) In the *Open* dialog, select `JetA_Combustor.trk` and click **Open**.
3. Create a scene to view the diameter of the Jet A fuel droplets as they enter the Fluid Domain.
   a) Create a copy of the **3D Flame** scene and rename it, `Particle Diameter`.
   b) Delete the **Particle Diameter** > **Displayers** > **Scalar 1** node.
   c) Right-click the **Particle Diameter** > **Displayers** node and select **New Displayer** > **Streamline**.
   d) Expand the **Displayers** > **Streamlines 1** node and set the following *Properties*:

Node	Property	Setting
Parts	*Parts*	**Phase 1 tracks**
Scalar Field	*Function*	**Track: Particle Diameter**

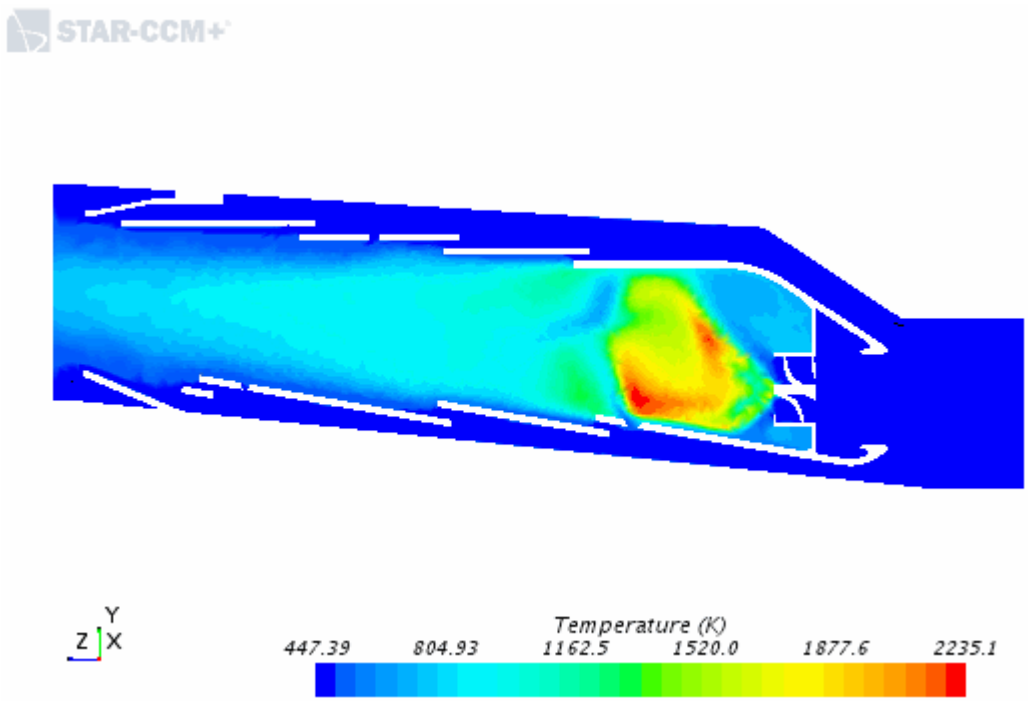
e) Open the **Particle Diameter** scene.

As the droplets evaporate, the diameter of the droplets decreases until all of the fuel becomes fully vaporized.

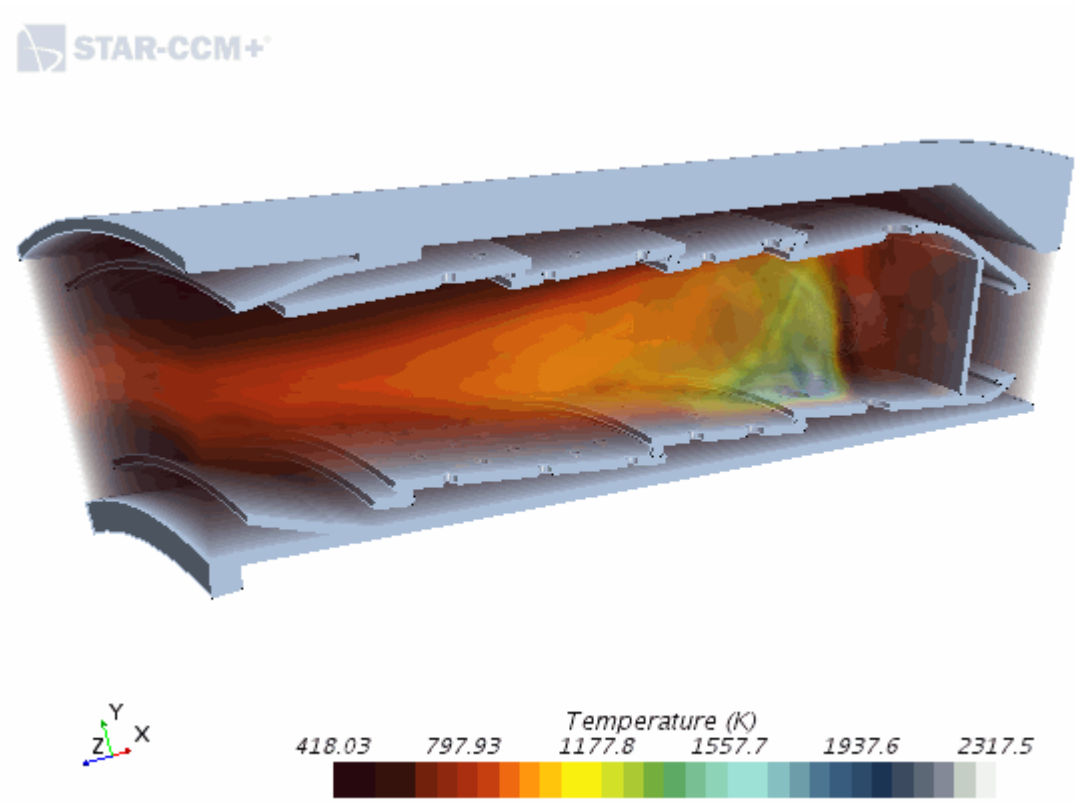


4. Open the **Temperature** and **3D Flame** scenes.

   The temperature scene shows the temperature on a cross section of the fluid domain. You can see that the temperature is highest in the area immediately after the droplets have evaporated—where the fuel combusts.

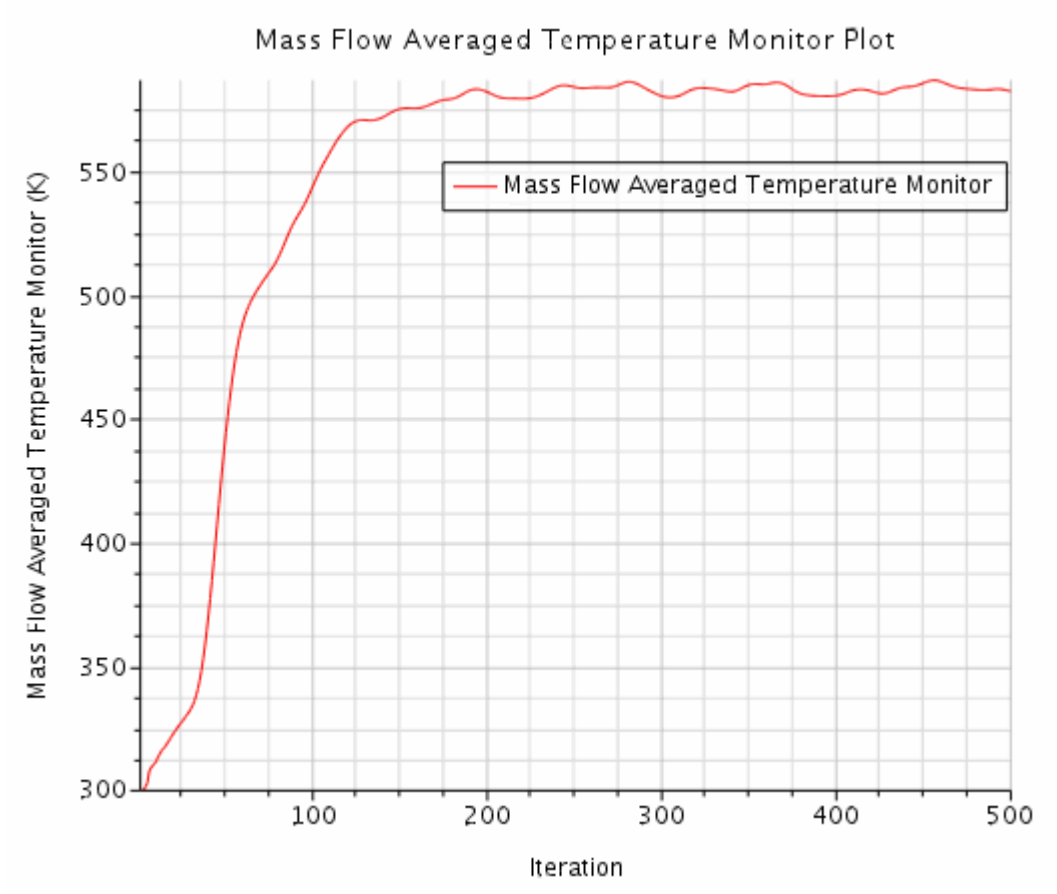


The 3D flame scene shows the temperature of the entire volume of the fluid domain.



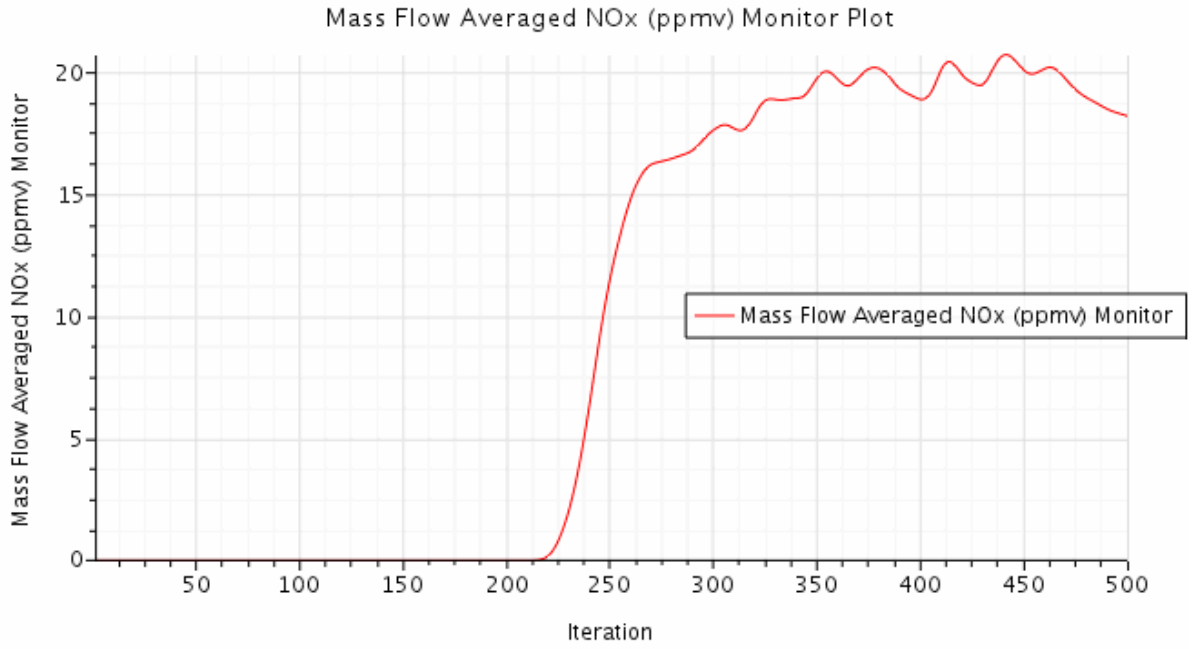
5. Open the following plots:

- **Plots** > **Mass Flow Averaged Temperature Monitor Plot**



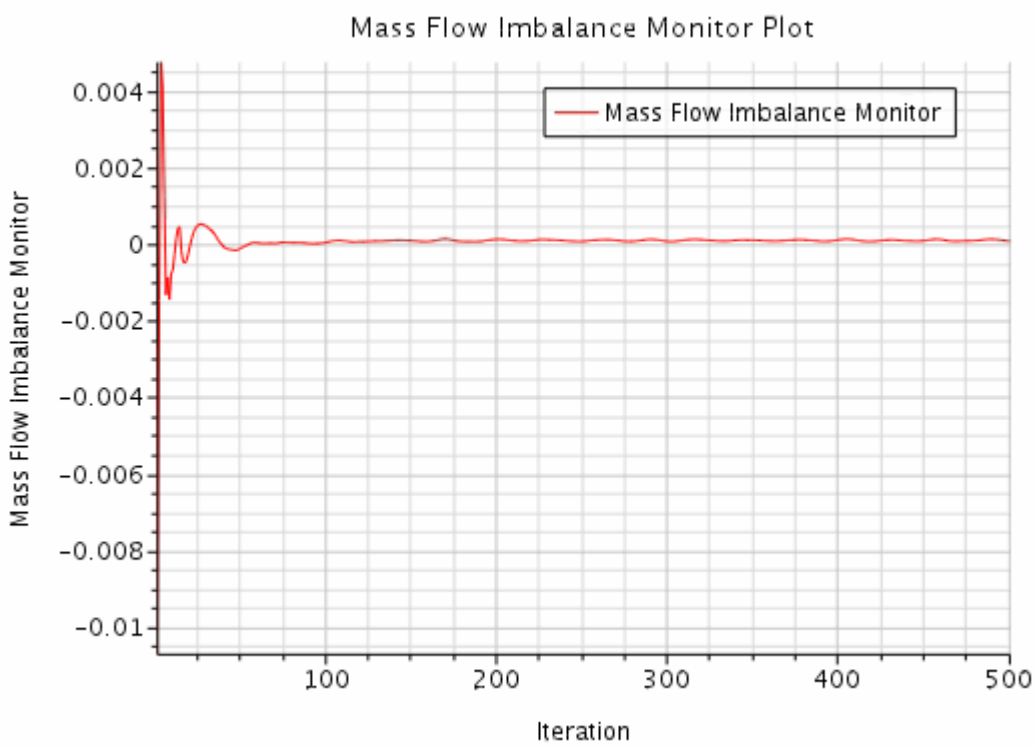
The average temperature at the outlet of the combustor in this simulation is less than 600 K. Industrial combustor simulations are likely to reach a temperature of around 1200 K at the outlet. The Mass Flow Averaged Temperature Monitor Plot is useful in showing that the simulation arrives at a steady temperature as it approaches convergence.

- **Plots** > **Mass Flow Averaged NOx (ppmv) Monitor Plot**



The solvers are set to begin solving for nitrogen oxide after 200 iterations, after which the mass flow of nitrogen oxide emissions at the outlet increases and fluctuates at about 20 ppmv. The custom field functions calculate the dry emissions corrected at 15% O2, as is common in industry. However, since the tutorial uses a simplified setup, the NOx levels in this simulation do not represent an industrial level.

- **Plots** > **Mass Flow Imbalance Monitor Plot**



This report compares the mass flow at the inlet with the mass flow at the outlet and shows that the overall mass imbalance equalizes as the simulation converges.

Bibliography

[808] *Combustion in Boilers and Furnaces 2015, Prof. Gordon E. Andrews, Energy Research Institute, School of Chemical and Process Engineering, University of Leeds*

Steady Laminar Flamelet: Non-Premixed Hydrogen

This tutorial demonstrates how to set up a turbulent hydrogen combustion simulation using the **Steady Laminar Flamelet (SLF)** reaction model for non-premixed flames.

The model assumes adiabatic conditions (no heat loss) and accounts for chemical non-equilibrium and finite-rate chemistry effects. To avoid having to set up the entire case, the simulation file from the 3-Step Eddy Break-Up tutorial is reloaded and modified.

The SLF model computes laminar opposed flow diffusion flames at a range of strain rates. These laminar flamelets are parameterized by mixture fraction and strain rate, and embedded in the 3D turbulent flame brush. Turbulent fluctuations are modeled with assumed shape β probability density functions (PDFs). The SLF model is appropriate for turbulent non-premixed (diffusion) flames.

Contents:

Prerequisites
Loading an Existing Simulation
Modifying Monitors, Plots, and Reports
Setting up the Models
Constructing the Steady Laminar Flamelet Table
Setting Boundary Conditions and Values
Setting Solver Parameters and Stopping Criteria
Running the Simulation
Visualizing the Results
Summary

Prerequisites

The instructions in the Steady Laminar Flamelet: Non-Premixed Hydrogen tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Loading an Existing Simulation

For this tutorial, start with a simulation file that contains pre-defined objects. Although some irrelevant settings are deleted, the boundary, interface, and initial conditions settings are preserved, which saves preparation time for similar cases.

Note: To start this tutorial, you can open one of two versions of the simulation file: the one that you created locally, or a previously solved simulation file that is available online. The following instructions apply to a locally created simulation file.

1. Launch STAR-CCM+.
2. Select **File** > **Load Simulation...**
3. In the *Load Simulation* dialog, click **Browse...**
4. In the *Open* dialog, navigate to the directory that contains the `ebuCombustion.sim` file.
5. Select `ebuCombustion.sim` then click **Open**.
6. In the *Load Simulation* dialog, click **OK**.
7. Save the simulation as `steadyLaminarFlamelet.sim`.

Modifying Monitors, Plots, and Reports

Remove the monitors, plots, and field functions that are not required for this tutorial.

The user field functions **Carbon** and **MinusCarbon** were created using Propane as an input fuel in the 3-Step Eddy Break-Up tutorial. Remove them and the reports, monitors, and plots that depend on them. In STAR-CCM+ dependent objects must be removed first.

1. Delete the following plots, monitors, and field functions nodes:
   - **Plots** > **Carbon Balance**.
   - **Monitors** > **Carbon In Monitor**.
   - **Monitors** > **Carbon Out Monitor**.
   - **Reports** > **Carbon In.**
   - **Reports** > **Carbon Out.**
   - **Tools** > **Field Functions** > **MinusCarbon**.
   - **Tools** > **Field Functions** > **Carbon**.

Setting up the Models

Models define the primary variables of the simulation. In this example, the flow involves a turbulent, compressible, multi-component gas whose components are reacting chemically.

The Segregated Flow model is used together with the standard K-Epsilon turbulence model and the Steady Laminar Flamelet model.

1. For the physics continuum, **Continua** > **Physics 1**:
   a) In the *Physics 1 Model Selection* dialog *Enabled Models* box, deactivate **Eddy Break-Up**, **Reacting Species Transport**, **Segregated Fluid Enthalpy**, **Ideal Gas**, and **Segregated Species**.
   b) Deactivate **Auto-select recommended models** then select the following models in order:

Group Box	Model
Enabled Models	**Three-Dimensional**

Group Box	Model
	Gradients
	Multi-Component Gas
	Reacting
	Segregated Flow
	Steady
	Turbulent
	Reynolds-Averaged Navier-Stokes
	K-Epsilon Turbulence
	Standard K-Epsilon
	Exact Wall Distance
	High y+ Wall Treatment
Reacting Flow Models	**Flamelet**
Flamelet Models	**Steady Laminar Flamelet (SLF)**
Equation of State for Flamelet	**Ideal Gas**
Energy Option	**Adiabatic**
Flame Type	**Non-Premixed Flame**

2. Click **Close**.
3. To review the models, open the **Physics 1** > **Models** node.
4. Select the **Segregated Flow** node and set *Convection* to **1st-order**.
5. For the **Adiabatic** node, set *Convection* to **1st-order**.
6. Make sure that the **Standard K-Epsilon** node has *Convection* set to **1st-order**.
7. Save the simulation.

Constructing the Steady Laminar Flamelet Table

In STAR-CCM+, you can import the mechanism in Chemkin format and generate the flamelet. The mechanism contains the species and reactions, the thermodynamic properties, and optionally, transport properties. Or, if you have a DARS Flamelet library, you can read these files into STAR-CCM+ and construct the table. For this tutorial, a DARS Flamelet library is provided.

Before constructing the table, copy the directory containing the flamelet library files into your working directory:

1. Navigate to the `doc/startutorialsdata/reactingFlow/data` directory of your STAR-CCM+ installation.
2. Copy the directory named `slf` into the directory that contains your simulation file.
3. In STAR-CCM+, right-click the **Continua** > **Physics 1** > **SLF Table Generator** node and select **Read DARS Flamelet Library and Construct Table**.

4. In the *Import Flamelet Files* dialog, click **Select** next to the *Select Library Folder* text box.
5. In the *Open* dialog, select the `slf` directory and click **Open**.
6. In the *Import Flamelet Files* dialog, click **OK**.

   The Steady Laminar Flamelet table is constructed from the flamelet data files in the directory, `slf`.

The species that are stored in the flamelet libraries are now listed under the **Multi-Component Gas** > **Gas Components** node.

No further specification of species or reactions is required since all necessary information is contained within the table.

7. Save the simulation.

Setting Boundary Conditions and Values

Specify the appropriate boundary property values.

The only change that is now required is to set the mixture fraction at the fuel inlet region to one.

1. Expand the **Regions** > **Default_Fluid** > **Boundaries** > **Fuel_Inlet** > **Physics Values** node.
2. Select the **Fuel_Inlet** > **Physics Values** > **Mixture Fraction Profile** node and set *Value* to [`1.0`].

Setting Solver Parameters and Stopping Criteria

Set the appropriate solver parameters and stopping criteria for the simulation.

To ensure convergence, reduce the under-relaxation factor for the pressure equation:

1. Select the **Solvers** > **Segregated Flow** > **Pressure** node and set *Under-Relaxation Factor* to `0.15`.

Change the default stopping criterion:

2. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to `600`.

   The solution stops after 600 iterations, unless this stopping criterion is changed or disabled.
3. Save the simulation.

Running the Simulation

Run the simulation.

To monitor the solution:

1. Right-click the **Scenes** > **Scalar Scene 1** node and select **Open**.

Clear the previous solution:

2. In the menu bar, select **Solution** > **Clear Solution**.
3. In the *Clear Solution* dialog, click **OK**.

To run the simulation:

4. Click (**Run**).

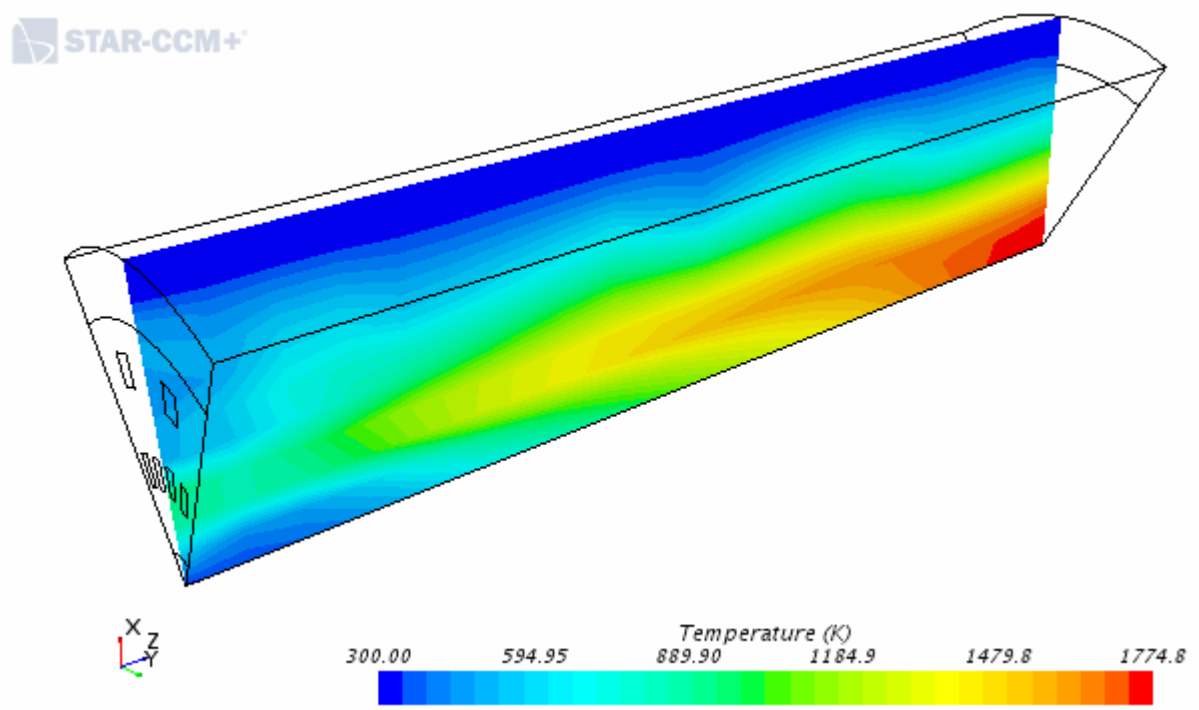
When the solution is complete:

5. Save the simulation.

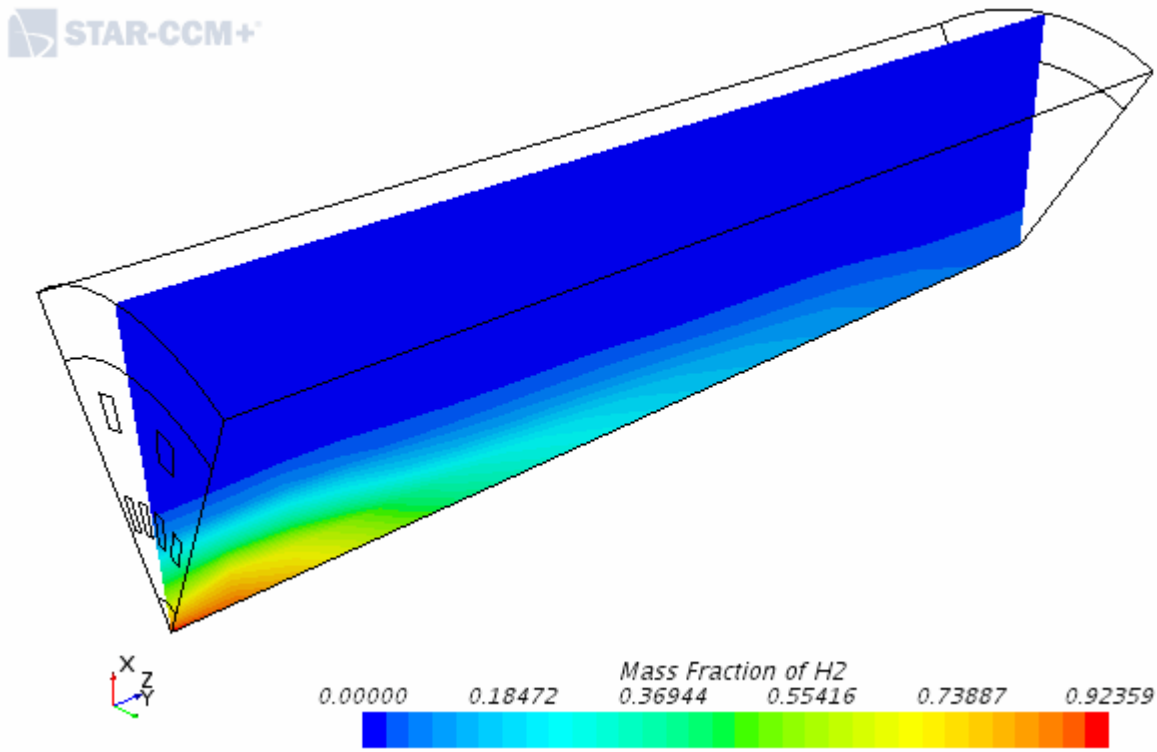
Visualizing the Results

This section of the tutorial covers post-processing.

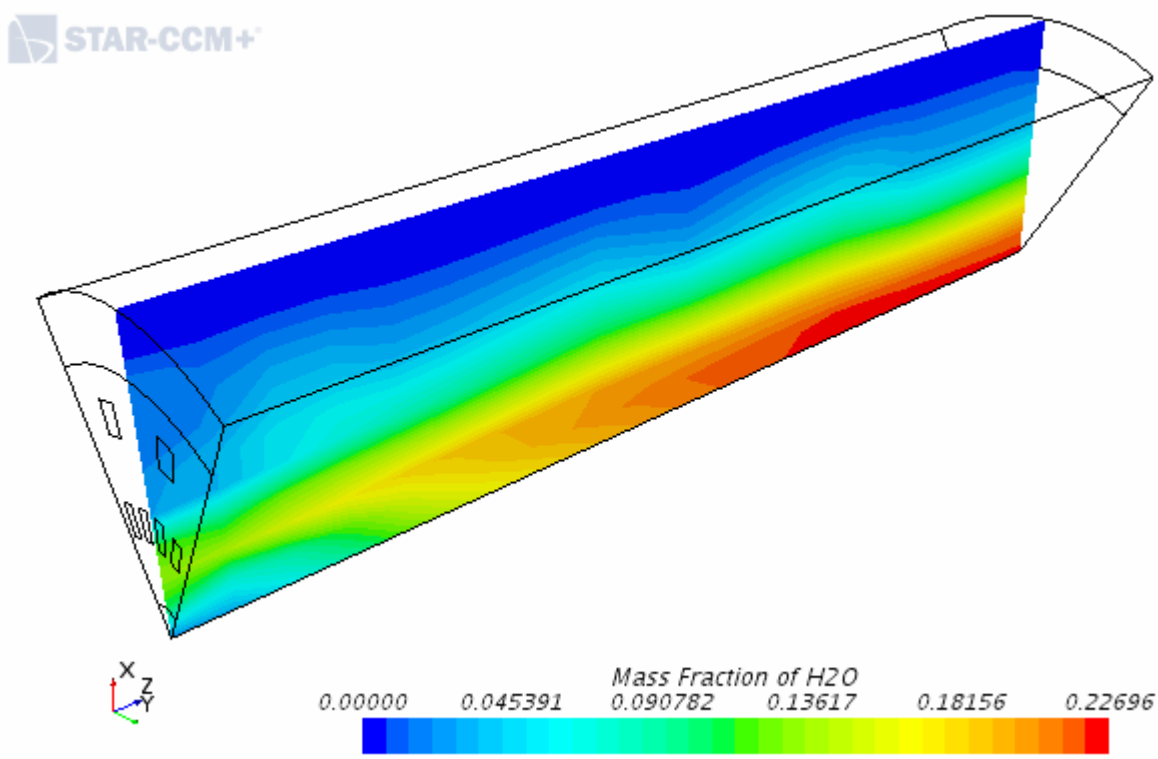
The temperature scene now appears as shown in the following screenshot:



1. In the *Scalar Scene 1* display, right-click the scalar bar and select **Mass Fraction** > **H2**.



2. In the *Scalar Scene 1* display, right-click the scalar bar and select **Mass Fraction** > **H2O**.



Summary

This tutorial demonstrated how to set up a hydrogen combustion simulation using the **Steady Laminar Flamelet** reaction model for non-premixed flames.

The steps covered were:

- Defining an adiabatic Steady Laminar Flamelet combustion model.
- Constructing a Steady Laminar Flamelet table.
- Setting stopping criteria.
- Running the solver for a set number of iterations.
- Analyzing the results using the visualization facilities of STAR-CCM+.

Surface Chemistry: Methane on Platinum Oxidation

This tutorial models methane on platinum oxidation using complex chemistry models.

The complex chemistry reactions are described in the Chemkin files `chem.inp` and `surf.inp`. In this simulation, a premixed mixture of methane and air flows over a platinum plate at a pressure of 1 bar and a temperature of 600 K.

Contents:

Prerequisites
Importing the Mesh and Naming the Simulation
Setting Up the Models
Importing the Complex Chemistry Definition
Setting Material Properties
Setting Initial Conditions
Setting Boundary Conditions and Values
Setting Solver Parameters and Stopping Criteria
Plotting Simulation Data
Reporting and Monitoring with Plots
Running the Simulation
Examining Plots

Prerequisites

The instructions in the Surface Chemistry: Methane on Platinum Oxidation tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Importing the Mesh and Naming the Simulation

To set up the STAR-CCM+ simulation, launch a simulation and import the supplied volume mesh.

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh**.
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/reactingFlow/data`.

5. Select `methaneOnPt.ccm` then click **Open**.

   STAR-CCM+ provides feedback on the import process in the *Output* window and a mesh region is created under the **Regions** node.

6. Save the simulation as `methaneOnPt.sim`.

Setting Up the Models

Models define the primary variables that are being solved in the simulation—such as pressure, velocity, temperature, and species mass fractions. In this example, the flow is a laminar, compressible, multi-component gas whose components are reacting chemically with the platinum surface.

In this tutorial, the Segregated Flow model is used together with the Complex Chemistry and Surface Chemistry models.

To select the physics models:

1. For the physics continuum, **Continua** > **Physics 1**, select the following models in order:

   Note: **Auto-select recommended models** must be activated.

Group Box	Model
Enabled Models	**Three Dimensional** (Selected automatically)
Time	**Steady**
Material	**Multi-Component Gas**
Reaction Regime	**Reacting**
Reactive Flow Models	**Reacting Species Transport**
Premixed Combustion Models	**Complex Chemistry**
Turbulence Chemistry Interactions	**Laminar Flame Concept**
Flow	**Segregated Flow**
Enabled Models	**Gradients** (Selected automatically)
	Segregated Species (Selected automatically)
	Segregated Fluid Enthalpy (Selected automatically)
Equation of State	**Ideal Gas**
Viscous Regime	**Laminar**
Optional Models	**Surface Chemistry**

2. Click **Close**.
3. To review the models, expand the **Physics 1** > **Models** node.
4. Save the simulation.

The **Complex Chemistry** model can use either the CVODE ODE solver or the DARS-CFD ODE solver. The default CVODE solver is used in this tutorial. In steady state simulations, you provide an iteration where

chemical reactions are activated. In this example, the cold flow simulation is run for the first 50 iterations to achieve a stable steady-state field.

5. Select the **Complex Chemistry** node and set the following properties:

Property	Setting
Source Enabled Trigger	`Iteration`
Begin	`51`
ODE Solver	`CVODE`

6. Save the simulation.

Importing the Complex Chemistry Definition

Import the complex chemistry definition from Chemkin-formatted files.

1. Copy the following files from `doc/startutorialsdata/reactingFlow/data/surfChemMethanePt/` to the directory where `methaneOnPt.sim` is located (your working directory):
   - `chem.inp`
   - `surf.inp`
   - `surf-therm.dat`
   - `therm.dat`
   - `tran.dat`

The two chemkin files `chem.inp` and `surf.inp` define the surface/gas chemistry mechanism. Thermodynamic data comes from `therm.dat` for species in the gas phase, and from `surf-therm.dat` for species in the surface phase. The file `tran.dat` has molecular transport data.

The next step is to bring the Chemkin-formatted files into the simulation.

2. Right-click the **Models** > **Complex Chemistry** node and select **Import Chemistry Definition (Chemkin format)**.
3. In the *Import Chemkin Files* dialog:
   a) Set the following properties:

Property	Setting
Fluid Chemistry Reaction File	`chem.inp`
Fluid Thermodynamic Properties File	`therm.dat`
Fluid Transport Properties File	`tran.dat`
Surface Chemistry Reaction File	`surf.inp`
Surface Thermodynamic Properties File	`surf-therm.dat`

   b) Click **OK**.
4. Save the simulation.

Setting Material Properties

Gas and surface species—as well as their reactions, thermodynamic properties, and molecular transport properties—are imported with the complex chemistry definition.

1. To see the species that are imported, expand the **Continua** > **Physics 1** > **Models** > **Multi-Component Gas** > **Gas Components** node.

The **Material Properties** > **Specific Heat** node for each species is automatically defined with the complex chemistry definition **Thermodynamic Polynomial Data**. The next step is to set mixture properties.

2. Open the **Multi-Component Gas** > **Material Properties** node and set the following properties:

Node	Property	Setting
Dynamic Viscosity	*Method*	**Mathur-Saxena Averaging**
Molecular Diffusivity	*Method*	**Kinetic Theory**
Thermal Conductivity	*Method*	**Mathur-Saxena Averaging**

3. Save the simulation.

Setting Initial Conditions

Specify appropriate values for the initial conditions.

The initial temperature of the domain is set to 600 K. Temperatures for this run are not expected to be low or high, so the temperature range between the minimum and maximum allowable temperatures is sufficient.

1. Expand the **Continua** > **Physics 1** > **Initial Conditions** node and set the following properties:

Node	Property	Setting
Static Temperature	*Value*	`600` K
Species Mass Fraction	*Value*	CH4: `0.02`
		N2: `0.752`
		O2: `0.228`

   Maintain the remaining default values.

2. Save the simulation.

Setting Boundary Conditions and Values

Specify boundary conditions for the inlet and outlet boundaries.

The boundaries **symmetry plane**, **symmetry plane-2**, and **symmetry plane-3** are set to `Symmetry Plane` for this problem. The boundary **Default Boundary Region** is set to a reactive wall and boundary conditions for surface species are specified. Also, boundary conditions for the inlet and outlet are specified.

For the outlet, the backflow conditions are provided, which are identical to the initial conditions that are specified above. For the inlet, a slightly faster velocity field is specified.

1. Expand the **Regions** > **Default_Fluid** > **Boundaries** > **pressure** node.
2. Set the following properties:

Node	Property	Setting
Physics Values		
└ **Species Mass Fraction**	*Value*	CH4: `0.02`
		N2: `0.752`
		O2: `0.228`

3. Select the **inlet** node and set the following properties:

Node	Property	Setting
Physics Values		
└ **Velocity Magnitude**	*Value*	`5.0` m/s
└ **Static Temperature**	*Value*	`600` K
└ **Species Mass Fraction**	*Value*	CH4: `0.02`
		N2: `0.752`
		O2: `0.228`

4. Expand the **Boundaries** > **Default_Boundary_Region** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Surface Mechanism Option**	*Mechanism*	**CATALYST**
└ **Thermal Specification**	*Condition*	**Temperature**
Physics Values		
└ **Surface Initial Bulk Activity Profiles** > **BULKPT_BULK Activity**	*Value*	Pt(B): `1.0`
└ **Surface Initial Site Fraction Profiles** > **SITEPT_SURFACE Fraction**	*Value*	Pt(S): `1.0`
└ **Static Temperature**	*Value*	`1000` K

The **Boundaries** > **Default_Boundary_Region** > **Physics Values** > **Surface Washcoat Factor** node has a *Value* of `1.0` by default. This setting specifies that the boundary has a smooth surface.

Specification of the boundary conditions is now complete.

5. Save the simulation.

Setting Solver Parameters and Stopping Criteria

Set the appropriate solver parameters and stopping criteria for the simulation.

The default solver settings are appropriate for this tutorial.

Run the simulation for 1100 iterations, which is sufficient to achieve a steady-state solution:

1. Select the **Stopping Criteria** > **Maximum Steps** node.
2. Set *Maximum Steps* to `1100`.

   The solution runs for 1100 iterations, unless this stopping criterion is changed or disabled.
3. Save the simulation.

Plotting Simulation Data

Create plots to help you visualize the solution.

Three plots are created: temperature, pressure, and mass fractions. These plots use a type of derived part that is called a line probe.

1. To create a line-probe, right-click the **Derived Parts** node and select **New Part** > **Probe** > **Line**.
2. In the *Create Line Probes* in-place dialog, set the following properties:

Property	Setting
Point 1	X: `2e-4`
	Y: `0.0`
	Z: `0.0`
Point 2	X: `2e-4`
	Y: `0.0`
	Z: `0.05`
Parts	**Default Fluid**

3. Click **Create** then **Close**.

Create the temperature plot:

4. Right-click the **Plots** node and select **New Plot** > **XY Plot**.
5. Rename the **XY Plot 1** node to `Temperature Plot`. Make the same entry for the *Title* property of this node.
6. Expand the **Temperature Plot** node and set the following properties:

Node	Property	Setting
Temperature Plot	*Title*	`Temperature Plot`
	Parts	**Line Probe**
X Type > **Vector Quantity**	*Value*	[`0.0,0.0,1.0`]
Y Types > **Y Type 1** > **Scalar Function**	*Field Function*	**Temperature**

Create a second plot for the CH4 mass fraction and set the properties.

7. Create an X-Y plot and rename the **XY Plot 1** node to `CH4 Plot`. Make the same entry for the *Title* property of this node.
8. Expand the **CH4 Plot** node and set the following properties:

Node	Property	Setting
CH4 Plot	*Title*	`CH4 Plot`
	Parts	**Line Probe**
X Type > **Vector Quantity**	*Value*	[`0.0,0.0,1.0`]
Y Types > **Y Type 1** > **Scalar Function**	*Field Function*	**Mass Fraction** > **Mass Fraction of CH4**

Create a third plot for the Site Surface Fraction of Platinum.

9. Create an X-Y plot and rename the **XY Plot 1** node to `Pt(S) Plot`. Make the same entry for the *Title* property of this node.
10. Expand the **Pt(S) Plot** node and set the following properties:

Node	Property	Setting
Pt(S) Plot	*Title*	`Pt(S) Plot`
	Parts	**Default_Boundary_Region**
X Type > **Vector Quantity**	*Value*	[`0.0,0.0,1.0`]
Y Types > **Y Type 1** > **Scalar Function**	*Field Function*	**Surface Site** > **Site Fraction** > **Surface Site Fraction of Pt(S)**

11. Save the simulation.

Reporting and Monitoring with Plots

In this tutorial, the CPU time that is needed to perform complex chemistry calculations is compared with the CPU time for the entire solver calculations.

STAR-CCM+ can dynamically monitor virtually any quantity while the solution develops. This requires setting up a report defining the quantity of interest and the parts of the region to be monitored. A monitor is then defined based on that report. The former also helps to create an appropriate X-Y graph plot. From the plots, it can be observed that complex chemistry calculations are time-consuming and that they take a large portion of

the total solver CPU time. However, complex chemistry calculations also offer the most detailed description of the underlying fluid chemistry, and for some problems these calculations are required to predict physical values of interest accurately.

1. Create two reports, one **Total Solver CPU Time** and one **Complex Chemistry Quantity Maximum**.
2. Select the **Complex Chemistry Quantity Maximum** node and set *Complex Chemistry Variable* to **Cpu Time**.
3. Multi-select both reports then right-click one of the reports and select **Create Monitor and Plot from Report**.
4. In the dialog that appears, click **Single Plot**.
5. Rename the **Plots** > **Reports Plot** node to `CPU Plot`.

   The analysis is now ready to be run.
6. Save the simulation.

Running the Simulation

Run the simulation.

1. To run the simulation, click (**Run**).

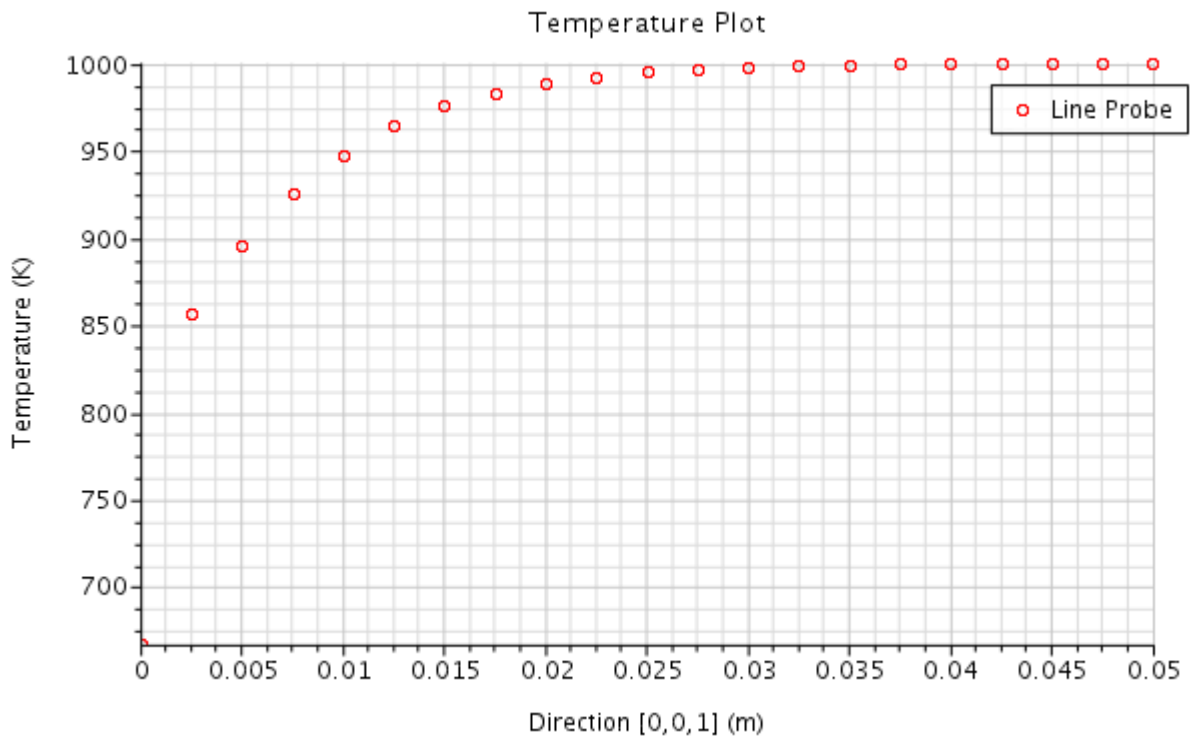
   The Residuals display is created automatically and shows the progress of the solver. This example looks different from the residuals that are generated in other simulations, since the plot depends on the models that are selected.

   The simulation continues until 1100 iterations are complete.
2. Save the simulation.

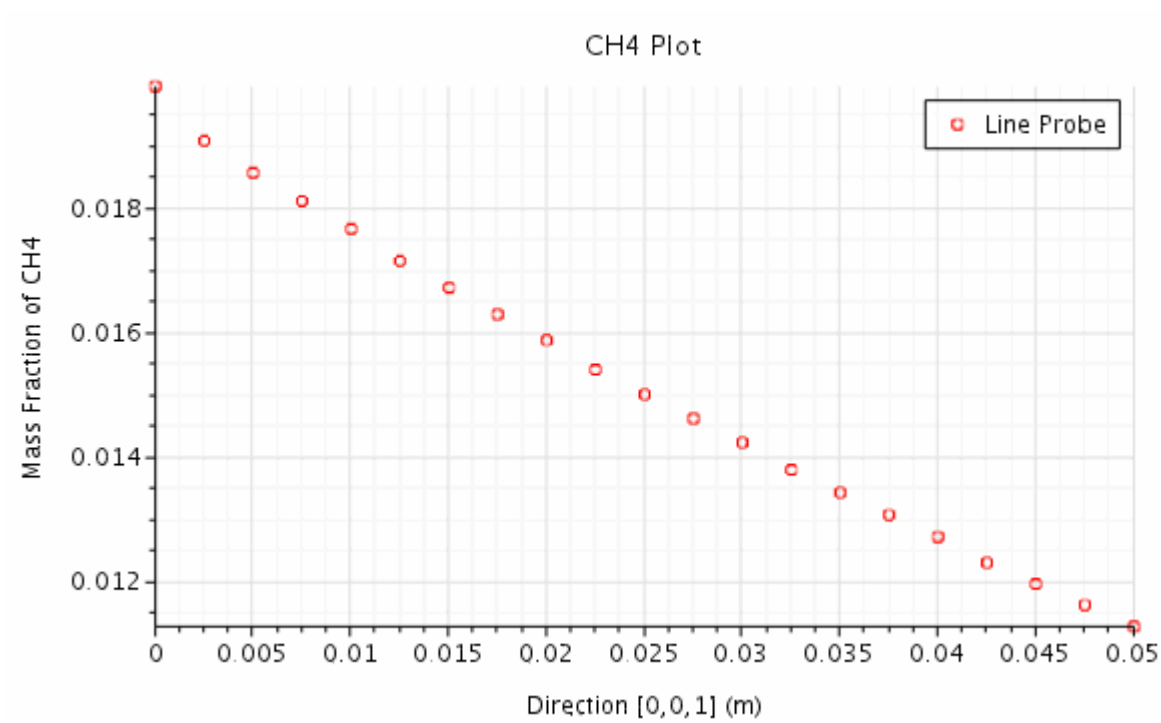
Examining Plots

Examine the results in the plots.

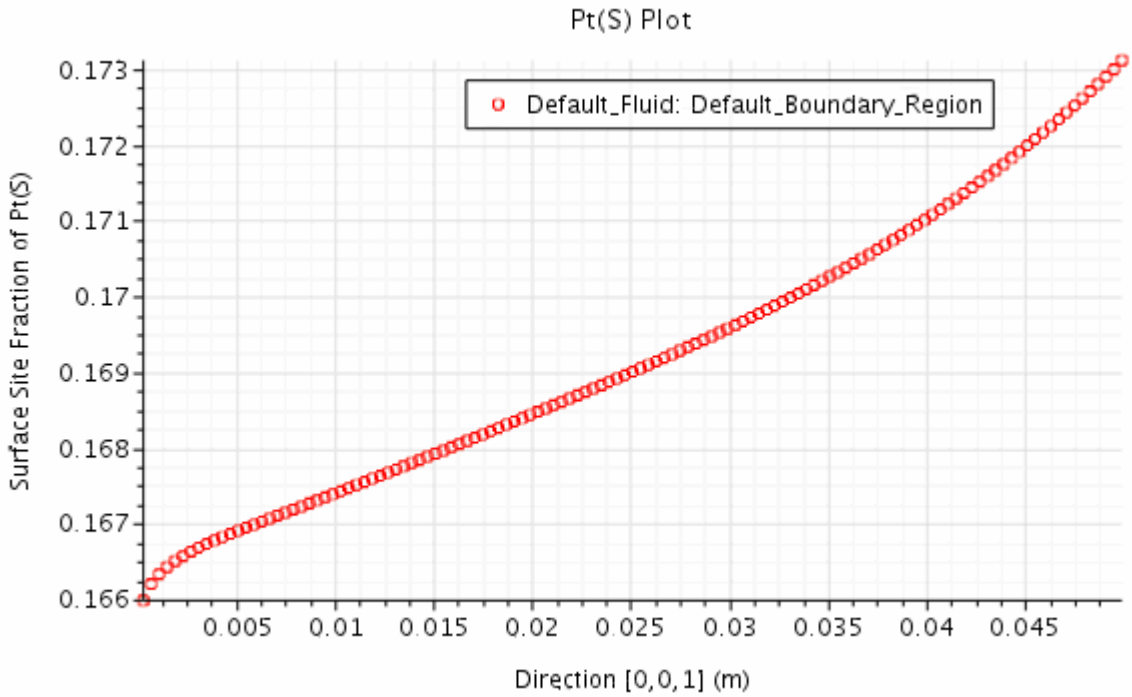
- Temperature



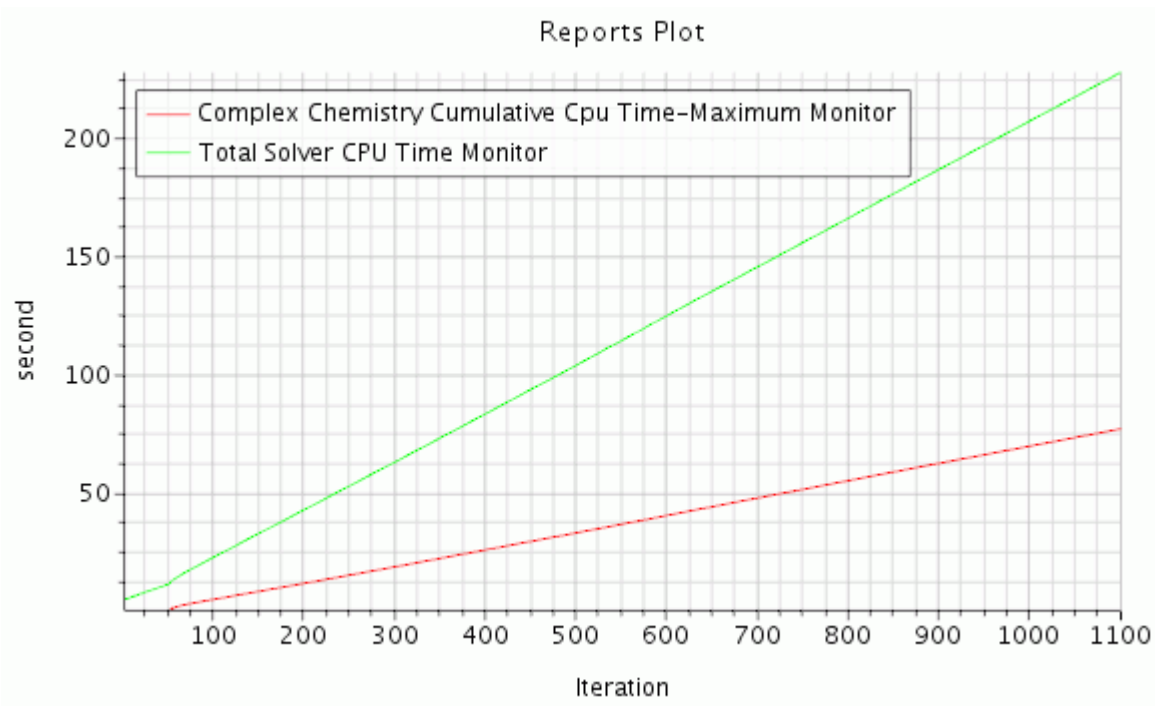
- Mass Fraction of CH4



- Site Fraction of Pt(S)


Pt(S) Plot
Default_Fluid: Default_Boundary_Region
Surface Site Fraction of Pt(S)
0.173
0.172
0.171
0.17
0.169
0.168
0.167
0.166
0.005
0.01
0.015
0.02
0.025
0.03
0.035
0.04
0.045
Direction [0,0,1] (m)

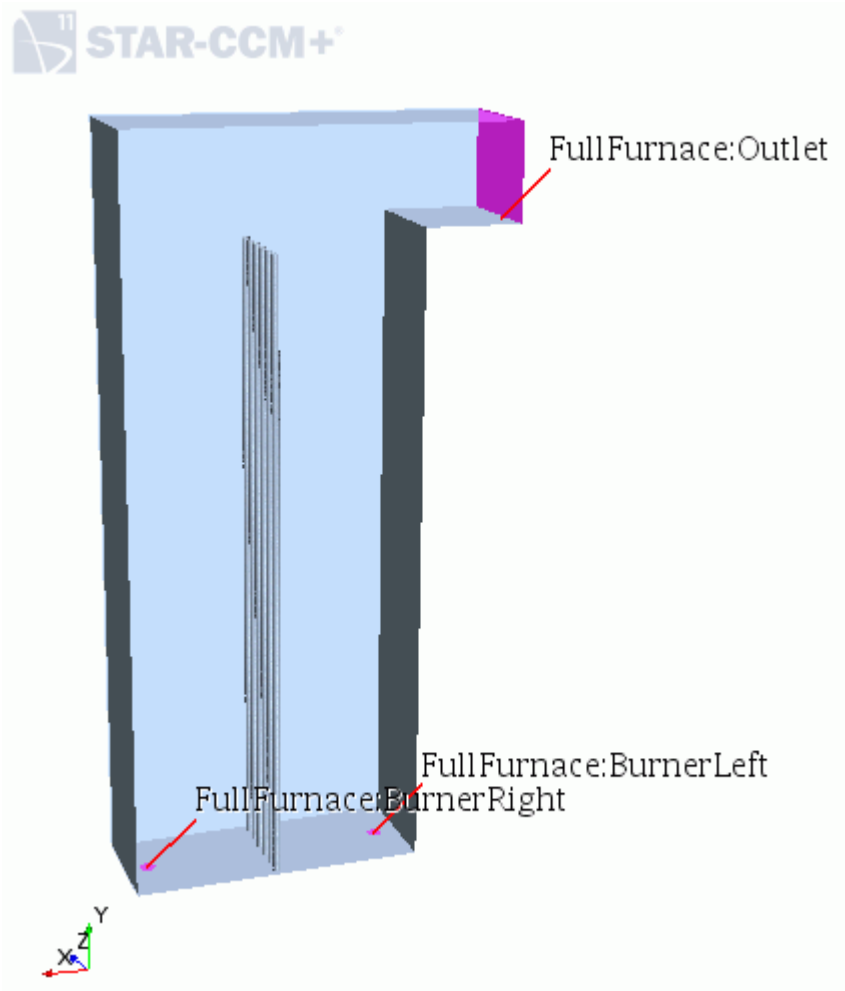
- CPU time


Reports Plot
Complex Chemistry Cumulative Cpu Time-Maximum Monitor
Total Solver CPU Time Monitor
second
200
150
100
50
100
200
300
400
500
600
700
800
900
1000
1100
Iteration

Reacting Channels: Steam Cracking Furnace

In this tutorial, you use the Reacting Channel Coupling co-simulation model in STAR-CCM+ to solve a reacting flow simulation using the same method as is required for systems such as: reformers, crackers, and process heaters.

There are two domains that are required in the geometry—the reacting channels (the tubes in which the processes occur) and the surrounding firebox.



The firebox is modeled as a three-dimensional calculation in STAR-CCM+. The physics for the processes in the reacting channels is modeled as a 1-D plug flow reactor using the DARS-CFD solver within the same STAR-CCM+ simulation. The co-simulation capability allows the exchange of information between the 3-D firebox and the 1-D process calculations, and also enforces the energy balance in the system. In the co-simulation, the information that is exchanged is the net heat flux of the reacting channels and the outer reacting channel wall temperatures. Additionally, detailed information on scalar and velocity profiles on the firebox and tubes is also available during the simulation.

1-D modeling of the reacting channel physics is a computationally efficient alternative to 3-D modeling of all reacting channels in the system. Especially if there are hundreds of tubes in industrial scale geometries. The geometry that is used in this tutorial consists of a simplified representation of a firebox that involves two burners with six reacting channels in-between the burners. Each burner has the same fuel composition and flow rate; the six tubes are the same length, diameter, and have the same properties. The objective of this tutorial is to set up the firebox and the physics for the reacting channels, and to evaluate the scalar profiles on the firebox and reacting channels.

The starting file for this tutorial contains pre-defined region and boundary settings. The mesh settings that are used are adequate to demonstrate the set-up, and obtain qualitative results from the simulation of reacting channels. For simulations that require more quantitative results, use finer mesh settings.

In the firebox, a partially premixed mixture of methane, air, and trace amounts of hydrogen is burned. The Eddy Break-up model is used and the following global reactions are defined:

$$\mathrm{CH_4} + 1\tfrac{1}{2}\mathrm{O_2} \rightarrow \mathrm{CO} + 2\mathrm{H_2O}$$

$$\mathrm{H_2} + \tfrac{1}{2}\mathrm{O_2} \rightarrow \mathrm{H_2O}$$

$$\mathrm{CO} + \tfrac{1}{2}\mathrm{O_2} \rightarrow \mathrm{CO_2}$$

In the reacting channels, Steam-Methane Reforming (SMR) reactions are simulated using a 3-step global mechanism which produces hydrogen. To account for the presence of catalysts inside the reacting channels, the following reactions are modeled in a packed bed system:

$$\mathrm{CH_4} + \mathrm{H_2O} \leftrightarrow \mathrm{CO} + 3\mathrm{H_2}$$

$$\mathrm{CO} + \mathrm{H_2O} \leftrightarrow \mathrm{CO_2} + \mathrm{H_2}$$

$$\mathrm{CH_4} + 2\mathrm{H_2O} \rightarrow \mathrm{CO_2} + 4\mathrm{H_2}$$

User-defined rates are required to describe the kinetics for this SMR mechanism. Since user-coding of reaction rates requires a Fortran compiler, customized Windows and Linux versions of the required Chemkin mechanisms are provided for this tutorial. For more information on compiling mechanisms for reacting channels, see the DARS User Guide.

Contents:

Loading the Starting File and Creating the Mesh
Defining the Firebox Physics
Setting up the Reacting Channels
Defining Physics Analysis Scenes and Plots
Specifying Initial Conditions and Boundary Conditions
Running the Simulation and Reviewing the Results
Bibliography

Loading the Starting File and Creating the Mesh

A suitably refined parts-based mesh is provided for geometry parts that represent a firebox with internal reacting channels.

Before starting this tutorial, copy one of the following reactions libraries from `[INSTALL_DIR]/doc/startutorialsdata/reactingFlow/data` to your working directory:

- If you are using a Linux system: `SMR_Chemistry_PFRLib_Linux`
- If you are using a Windows system: `SMR_Chemistry_PFRLib_Windows`

Note: Before starting this tutorial, make sure that you have at least 4 processes available to use.

1. Open STAR-CCM+ and select **Load Simulation**.
2. In the *Load Simulation* dialog:
   a) Set *Process Options* to **Parallel on Local Host** and *Compute Processes* to `4` or more—depending on availability.
   b) Click **Browse**.
   c) In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/reactingFlow/data`.
   d) Select `RC_StartingSim.sim` then click **Open**.
   e) Click **OK**.
3. Expand the **Regions** > **Firebox** > **Boundaries** node.

   The following boundary types are specified.

Boundary	Property	Setting
FullFurnace.BurnerRight	*Type*	*Mass Flow Inlet*
FullFurnace.BurnerLeft	*Type*	*Mass Flow Inlet*
FullFurnace.Outlet	*Type*	*Outlet*
FullFurnace.Walls	*Type*	*Symmetry Plane*

   All other boundaries are specified as *Wall* boundaries.

   Note: Specifying inlet, outlet, and symmetry boundaries before meshing ensures that prism layers are not created on those boundaries.

An automated mesh operation with various surface control mesh settings is set up and ready to mesh.

4. Right-click the **Geometry** > **Operations** > **Automated Mesh** node and select **Execute**.
5. When the mesh creation is complete, **create a mesh scene**.

   The mesh is refined suitably for this tutorial analysis. However, to save computational expense, this mesh is coarser than the mesh which would be used for realistic industrial scenarios.

   The simulation file also contains a pre-defined Temperature scene which you can use at a later stage to visualize the temperature on the reacting channel boundaries.
6. Save the file as `Firebox.sim`.

Defining the Firebox Physics

Reacting physics models simulate combustion in the firebox. Co-simulation models exchange data between the firebox and the plug flow reactor code for reacting channels.

Specify the physics models that are used to model hot flow within the Firebox. Also activate the co-simulation model.

1. For the continuum, **Continua** > **Physics 1**, select the following models in order:

   Note: Make sure that *Auto-select recommended models* is activated.

Group Box	Model
Space	**Three Dimensional**
Time	**Steady**
Material	**Multi-Component Gas**
Reaction Regime	**Reacting**
Reacting Flow Models	**Reacting Species Transport**
Reacting Species Models	**Eddy Break-up**
Flow	**Segregated Flow**
Enabled Models	**Gradients** (selected automatically) **Segregated Species** (selected automatically) **Segregated Fluid Enthalpy** (selected automatically)
Equation of State	**Ideal Gas**
Viscous Regime	**Turbulent**
Enabled Models	**Reynolds-Averaged Navier-Stokes** (selected automatically)
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**
Enabled Models	**Realizable K-Epsilon Two-Layer** (selected automatically) **Exact Wall Distance** (selected automatically) **Two-Layer All y+ Wall Treatment** (selected automatically)
Optional Models	**Radiation**
Radiation	**Participating Media Radiation (DOM)**
Radiation Spectrum (Participating)	**Gray Thermal Radiation**
Optional Models	**Co-Simulation**
Co-Simulation Models	**Reacting Channel**
Enabled Models	**Reacting Channel Steady Coupling** (selected automatically)
Optional Models	**Cell Quality Remediation**

2. Click **Close**.

3. Select the **Physics 1** > **Models** > **Eddy Break-up** node and set the following properties:

Property	Setting
Begin	`10`
Reaction Control	**Standard EBU**

Specify the gas components within the Firebox:

4. Right-click the **Models** > **Multi-Component Gas** > **Gas Components** node and select **Select Mixture Components**.
5. In the *Select Mixture Components* dialog:
   a) Expand the **Material Databases** > **Standard** > **Gases** node and select the following:
      - CH4 (Methane)
      - CO (Carbon Monoxide)
      - CO2 (Carbon Dioxide)
      - H2 (Hydrogen)
      - H2O (Water)
      - N2 (Nitrogen)
      - O2 (Oxygen)

   b) Click **Apply**, then **Close**.

   The **Multi-Component Gas** > **Gas Components** node is populated with the selected gas components.

To reflect the behavior of the gas components accurately, several material properties are specified. Since participating media radiation is used in this simulation, the optical path length must be specified to compute the absorption coefficient. The optical path length (OPL) was calculated previously using the result of two sum reports—Volume and Total Surface Area—in the relationship:

$$OPL = 3.6\frac{\text{Volume}}{\text{Total Surface Area}} \tag{4168}$$

6. Expand the **Multi-Component Gas** > **Material Properties** node and set the following properties:

Node	Property	Setting
Absorption Coefficient	*Method*	**Weighted Sum of Gray Gases**
⌊ **Weighted Sum of Gray Gases**	*OpticalPathLength*	`0.75 m`
Thermal Conductivity	*Method*	**Mass Weighted Mixture**
Turbulent Prandtl Number		
⌊ **Constant**	*Value*	`0.7`
Turbulent Schmidt Number		
⌊ **Constant**	*Value*	`0.7`

There are three combustion reactions that occur within the firebox domain:

- $CH_4 + 1\frac{1}{2}O_2 \rightarrow CO + 2H_2O$
- $H_2 + \frac{1}{2}O_2 \rightarrow H_2O$
- $CO + \frac{1}{2}O_2 \rightarrow CO_2$

To specify these reactions within the Reacting model:

7. Right-click the **Models** > **Reacting** > **Reactions** node and select **New Reaction**.

   A **Reactions** > **Reaction 1** node appears.
8. To create **Reaction 2** and **Reaction 3** nodes, repeat the above step twice more.
9. Set the Reactants and Products, and specify the stoichiometry for each reaction as follows:

Node	Action
Reaction 1 > **Reactants**	Right-click and select **Add Reactant** > **Gas Mixture** > **CH4**
	Right-click and select **Add Reactant** > **Gas Mixture** > **O2**
⌊ **O2**	Set **Stoich. Coeff** to `1.5`
Reaction 1 > **Products**	Right-click and select **Add Product** > **Gas Mixture** > **CO**
	Right-click and select **Add Product** > **Gas Mixture** > **H2O**
⌊ **H2O**	Set **Stoich. Coeff** to `2`
Reaction 2 > **Reactants**	Right-click and select **Add Reactant** > **Gas Mixture** > **H2**
	Right-click and select**Add Reactant** > **Gas Mixture** > **O2**
⌊ **O2**	Set **Stoich. Coeff** to `0.5`
Reaction 2 > **Products**	Right-click and select **Add Product** > **Gas Mixture** > **H2O**
Reaction 3 > **Reactants**	Right-click and select **Add Reactant** > **Gas Mixture** > **CO**
	Right-click and select **Add Reactant** > **Gas Mixture** > **O2**
⌊ **O2**	Set **Stoich. Coeff** to `0.5`
Reaction 3 > **Products**	Right-click and select **Add Product** > **Gas Mixture** > **CO2**

10. Rename the **Reactions** sub-nodes as follows:

Node	Rename as:
Reaction 1	`(CH4 + 1.5O2 --> CO + 2H2O)`
Reaction 2	`(H2 + 0.5O2 --> H2O)`
Reaction 3	`(CO + 0.5O2 --> CO2)`

Setting up the Reacting Channels

Reacting channel co-simulation is set up between the Plug Flow Reactor (PFR) in the reacting channels (where steam cracking reactions occur) and the firebox domain in STAR-CCM+ (where combustion reactions occur).

Set up the co-simulation:

1. In the simulation tree, expand the **Co-Simulations** > **Reacting Channel Co-Simulation** > **Conditions** node and set the following properties:

Node	Property	Setting
PFR Library Input Option	*Input Type*	**Binary**
PFR User Coding Option	*User Coding*	**Calculate Source Term**
PFR Library Run Time Compile (only if you are using a Linux System)	*Input Type*	**Precompiled**

2. Select the **Reacting Channel Co-Simulation** > **Values** > **PFR Library** node and next to *PFR Input*, click ...
3. In the *PFR-Library* dialog that appears, navigate to and select one of the following files:
   - If you are using a Windows system, use `SMR_Chemistry_PFRLib_Windows`
   - If you are using a Linux system, use `SMR_Chemistry_PFRLib_Linux`
4. Click **OK**.

   The *PFR Library - Properties*, *Files Loaded* and *Library Loaded*, are activated to confirm that the library and associated files are loaded.

   To see the species that are imported, select the **Reacting Channel Co-Simulation** > **Conditions** > **Reacting Channel Species** node.

Next, identify the boundaries that surround the reacting channels within the Firebox. These channels must be set to exchange data as part of the co-simulation.

5. Expand the **Regions** > **Firebox** > **Boundaries** node then:
   a) Multi-select all six of the **FullFurnace.Pipe #** nodes.
   b) Right-click one of the selected nodes and select **Edit**.
   c) Expand **Physics Conditions** > **External Code Coupling Specification** and set *External Code Coupling Specification* to **Reacting Channel Co-Simulation: Zone 1**.
   d) Click **Close**.

The **Co-Simulations** > **Reacting Channel Co-Simulation** > **Zones** > **Zone 1** > **Conditions** > **Coupled Model Parts** node updates to show the boundaries that are selected in STAR-CCM+ for coupling to the Plug Flow Reactor.

6. Select the **Co-Simulations** > **Reacting Channel Co-Simulation** > **Zones** > **Zone 1** > **Conditions** > **Reacting Channel Type** node and set *Correlation Type* to **Packed Bed**
   The reacting channels are specified as a packed bed reaction system. This setting signifies that each of the tubular channels is packed with porous material or particles through which the steam cracking reaction components flow.

To create a realistic set-up for modeling reacting channels, it is important to set the properties of each zone to reflect the actual parameters and physics of the reacting channel pipes within the firebox. In this tutorial, the inlet area and pipe length values are taken from the dimensions of the geometry that is provided in the starting file. Other conditions and values are arbitrary or are based on values used by G.D. Stefanidis and others in [809]. To determine values that are realistic for a given case, you can calculate some of the correlation values using STAR-CCM+. The wall discretization orientation is set as the direction of flow within the reacting channel.

7. Expand the **Zone 1** > **Values** node and set the following *Properties*:

Node	Property	Setting
Inlet Area	*Value*	`0.00785` m^2
Inlet Mass Fractions	*Initial Mass Fraction*	CH4: `0.2851`
		H2O: `0.7128`
		CO2: `1.1403E-3`
		N2: `9.597E-4`
Inlet Pressure	*Value*	`2797999.0` Pa
Inlet Temperature	*Value*	`857.0` K
Inlet Velocity	*Value*	`0.6` m/s
Packed Bed Heat Transfer Factor	*Value*	`1.68`
Packed Bed Particle Diameter	*Value*	`0.0075` m
Pipe Length	*Value*	`6.5` m
Wall Discretization	*Number of Axial Divisions*	`100`
	Orientation	`[0.0, 1.0, 0.0]`

Defining Physics Analysis Scenes and Plots

Update scenes and plots to display physics values on geometry parts when running the reacting channel simulation.

In reacting channel simulations, it is useful to check the temperature along the length of the reacting channel boundaries to make sure that it is consistent and sufficiently high for the steam cracking reactions which occur

in the reacting channels. You can also monitor the progress of the steam cracking reactions by plotting the consumption of reactants and production of products along the length of the reacting channels. It is also important to monitor the temperature and the release of CO2 at the outlet of the firebox.

To display the temperature on the reacting channels:

1. Open the Temperature scene.
2. Expand the **Scenes** > **Temperature** > **Displayers** node and set the following properties:

Node	Property	Setting
Scalar 1	*Representation*	**Volume Mesh**
⌊ **Scalar Field**	*Function*	**Temperature**

You can also plot the temperature within the reacting channels using a Reacting Channel X-Y plot.

3. Right-click the **Plots** node and select **New Plot** > **Reacting Channel XY Plot**.
4. Rename the **Reacting Channel XY Plot 1** node to `Reacting Channel Temperature` and set the following properties:

Property	Setting
Title	**Reacting Channel Temperature**
Parts	**FullFurnace.Pipe1 FullFurnace.Pipe2 FullFurnace.Pipe3 FullFurnace.Pipe4 FullFurnace.Pipe5 FullFurnace.Pipe6**

5. Multi-select all six of the **Plots** > **Reacting Channel Temperature** > **Data Series** > **Reacting Channel Co-Simulation: FullFurnace.Pipe#** nodes and set *Y Variable* to **Temperature**.

To monitor the progress of the steam cracking reactions within the reacting channels, plot the mass fractions of one of the reactants and one of the products, along the length of the reacting channel pipes:

6. Right-click the **Plots** node and select **New Plot** > **Reacting Channel XY Plot**.
7. Rename the **Reacting Channel XY Plot 1** node to `CH4 Mass Fraction` and set the following properties:

Property	Setting
Title	**CH4 Mass Fraction**
Parts	**FullFurnace.Pipe1 FullFurnace.Pipe2 FullFurnace.Pipe3 FullFurnace.Pipe4 FullFurnace.Pipe5 FullFurnace.Pipe6**

8. Multi-select all six of the **Plots** > **CH4 Mass Fraction** > **Data Series** > **Reacting Channel Co-Simulation: FullFurnace.Pipe#** nodes and set *Y Variable* to **CH4**.
9. Right-click the **Plots** node and select **New Plot** > **Reacting Channel XY Plot**.
10. Rename the **Reacting Channel XY Plot 1** node to `H2 Mass Fraction` and set the following properties:

Property	Setting
Title	**H2 Mass Fraction**
Parts	**FullFurnace.Pipe1 FullFurnace.Pipe2 FullFurnace.Pipe3 FullFurnace.Pipe4 FullFurnace.Pipe5 FullFurnace.Pipe6**

11. Multi-select all six of the **Plots** > **H2 Mass Fraction** > **Data Series** > **Reacting Channel Co-Simulation: FullFurnace.Pipe#** nodes and set *Y Variable* to **H2**.

To create reports that show the mass flow averaged temperature at the outlet and the mass of CO2 at the outlet:

12. Right-click the **Reports** node and select **New Report** > **Mass Flow Averaged**.
13. Rename the **Mass Flow Averaged 1** report to `Temperature at Outlet`.
14. Select the **Temperature at Outlet** node and set the following *Properties*:

Property	Setting
Field Function	**Temperature**
Parts	**Regions** > **Firebox** > **Boundaries** > **FullFurnace.Outlet**

15. Right-click the **Reports** > **Temperature at Outlet** node and select **Create Monitor and Plot from Report**.
16. Right-click the **Reports** node and select **New Report** > **Mass Flow Averaged**.
17. Rename the **Mass Flow Averaged 1** report to `Mass Fraction of CO2 at Outlet`.
18. Select the **Mass Fraction of CO2 at Outlet** node and set the following *Properties*:

Property	Setting
Field Function	**Mass Fraction** > **Mass Fraction of CO2**
Parts	**Regions** > **Firebox** > **Boundaries** > **FullFurnace.Outlet**

19. Right-click the **Reports** > **Mass Fraction of CO2 at Outlet** node and select **Create Monitor and Plot from Report**.
20. Save the simulation .

Specifying Initial Conditions and Boundary Conditions

Initially, before any combustion reactions occur in the firebox, the volume of the firebox contains air. These initial conditions are defined within the physics continuum.

1. Expand the **Continua** > **Physics 1** > **Initial Conditions** node and set the following:

Node	Property	Setting
Species Mass Fraction	*Value*	O2 = `0.233` N2 = `0.767`

2. Expand the **Regions** > **Firebox** > **Boundaries** node.
3. Multi-select the following boundary nodes:
   - **FullFurnace.BurnerLeft**
   - **FullFurnace.BurnerRight**
4. Right-click one of the selected nodes and select **Edit**. In the *Multiple Objects* dialog, expand **Physics Values** and set the following:

Node	Property	Setting
Mass Flow Rate	*Value*	`0.2113 kg/s`
Species Mass Fraction	*Value*	H2 = `0.003158` O2 = `0.220186` N2 = `0.724412` CH4 = `0.052244`

5. Click **Close**.
6. Multi-select the following boundary nodes:
   - **FullFurnace.CoffinWalls**
   - **FullFurnace.CoffinWalls2**
   - **FullFurnace.Floor**
   - **FullFurnace.Symmetry1**
   - **FullFurnace.Symmetry2**
7. Right-click one of the selected nodes and select **Edit**. In the *Multiple Objects* dialog, expand **Physics Values** > **Surface Emissivity** and set *Value* to `0.85`.
8. Click **Close**.
9. Multi-select all six of the **FullFurnace.Pipe#** nodes.
10. Right-click one of the selected nodes and select **Edit**. In the *Multiple Objects* dialog, expand **Physics Values** > **Surface Emissivity** and set *Value* to `0.6`.
11. Click **Close**.
12. Save the simulation .

Running the Simulation and Reviewing the Results

You can review the progress of the simulation as it runs by monitoring the scalar scenes and reports that are set up.

Before running the simulation, define the solver settings and stopping criteria:

1. Select the **Solvers** > **Segregated Flow** node and activate *Continuity Initialization*.
2. Expand the **Solvers** > **DO Radiation** node and set the following:

Node	Property	Setting
DO Radiation	*Maximum Iteration*	`25`

Node	Property	Setting
DO Radiation	*Solver Tolerance*	`0.001`
⌊ **Update Frequency**		
⌊ **Steady**	*Iteration Update Frequency*	`10`

3. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to `1000`.

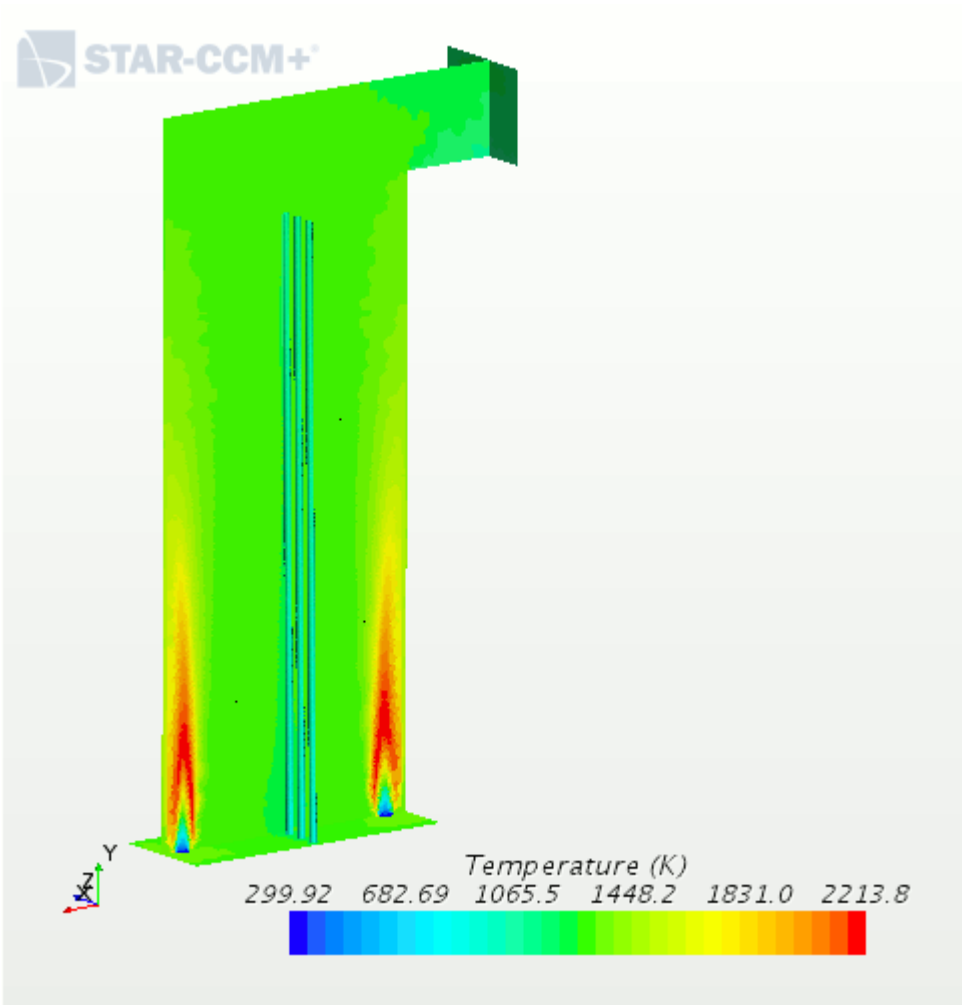
To review the progress of the simulation in the *Output* window as it runs, open the scenes and plots.

4. Open the following scenes and plots:
   - **Scenes** > **Temperature**
   - **Plots** > **CH4 Mass Fraction**
   - **Plots** > **H2 Mass Fraction**
   - **Plots** > **Mass Fraction of CO2 at Outlet**
   - **Plots** > **Reacting Channel Temperature**
   - **Plots** > **Temperature at Outlet**
5. Run the simulation .

Running the case can take a considerable time – depending on the number of processors that are being used. Using four processors can take several hours.

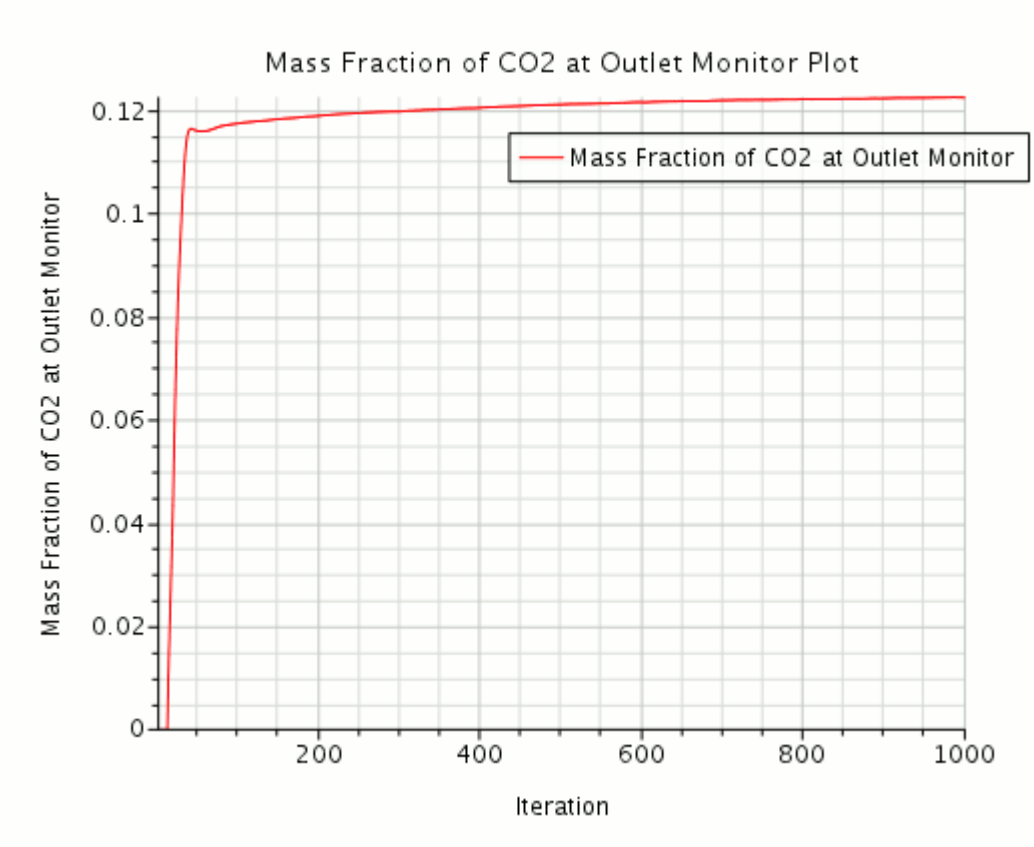
6. When the maximum number of steps is reached, save the simulation .
7. In the *Output* window, click the *Temperature* tab.

   The scene displays the temperature on an arbitrary plane surface through the centre of the furnace volume and on the surface of the reacting channels. You can see that the temperature along the length of the channels is distributed evenly as expected. An even temperature distribution ensures that the reaction components within the channels are exposed to a consistent temperature within the full length of the channels.



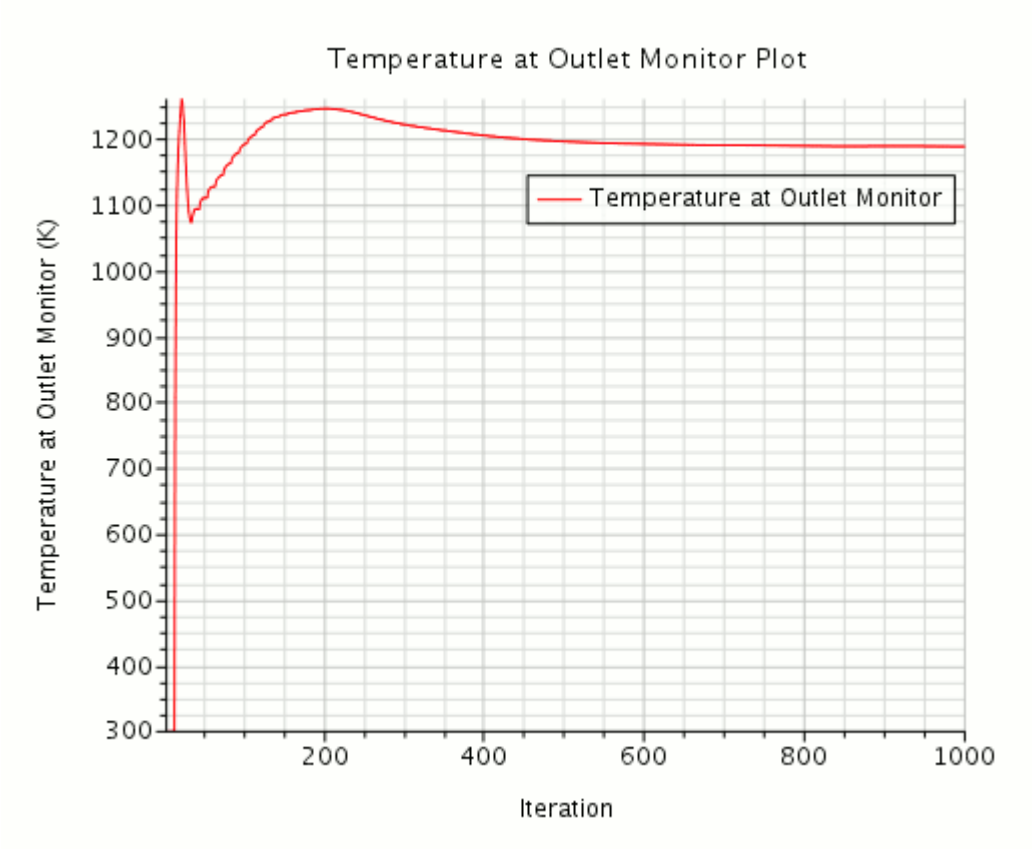
8. In the *Output* window, click the *Mass Fraction of CO2 at Outlet* tab.

   The mass fraction of CO2 at the Outlet reaches a constant value of 0.12.



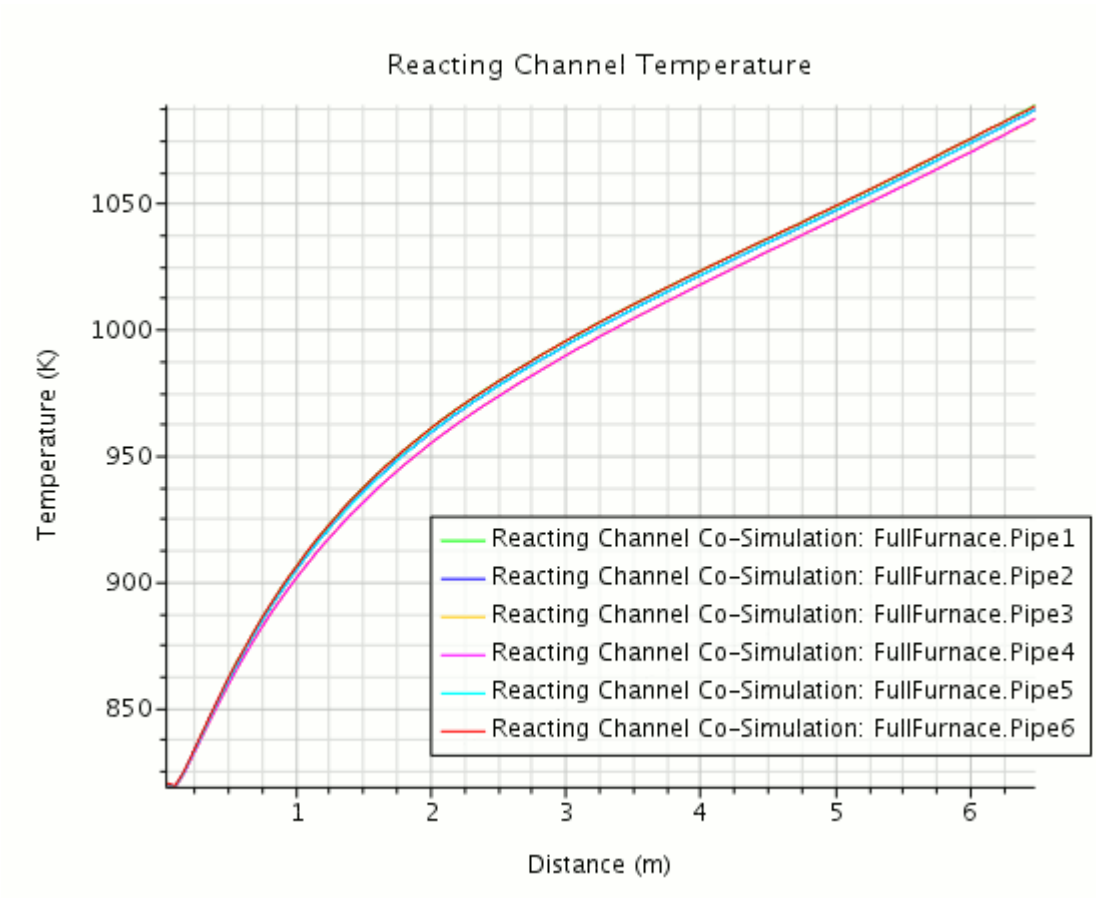
9. In the *Output* window, click the *Temperature at Outlet* tab.

   The temperature at the outlet reaches a constant temperature of around 1194 K.



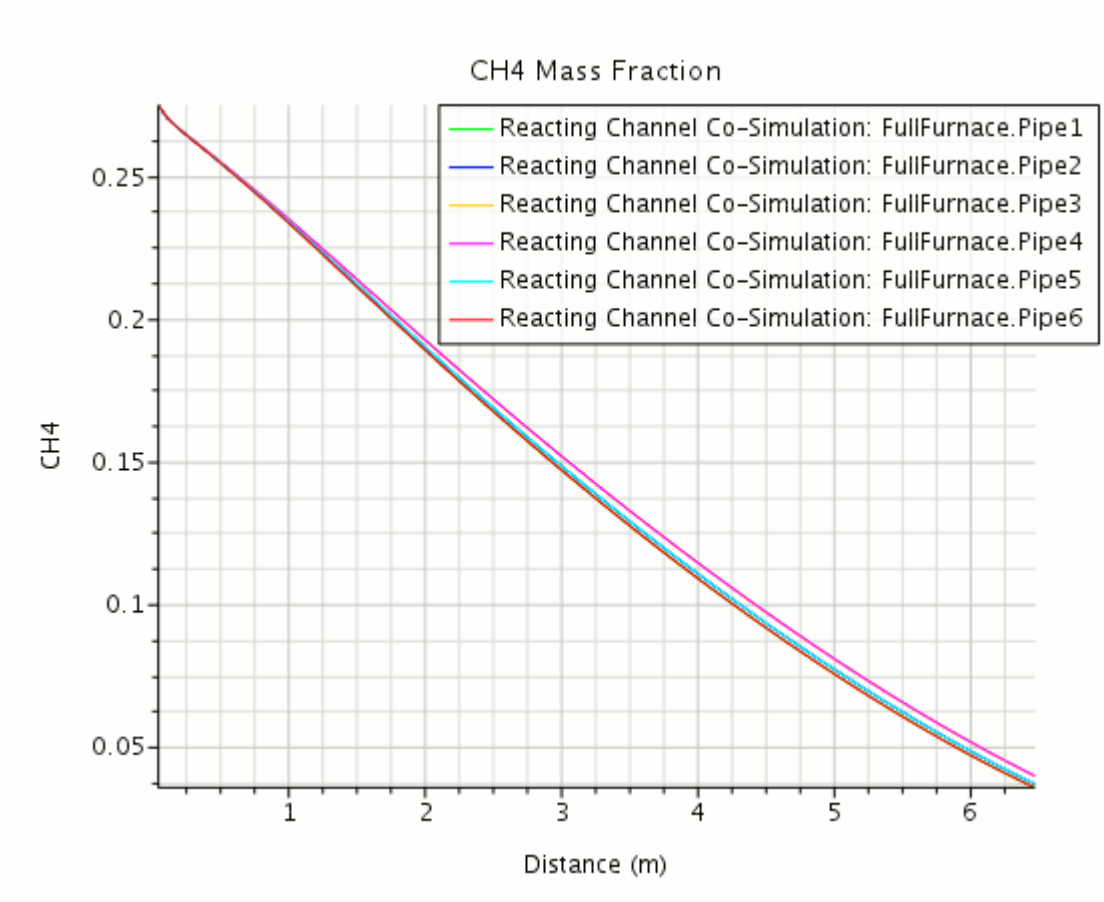
10. In the *Output* window, click the *Reacting Channel Temperature* tab.

    The temperature increases along the length of the reacting channels to around 820 K at the end of the pipes. In this simulation, long thin cylinders, which are not tapered at the end, represent the reacting channel pipes. If the pipes were tapered at the end, the temperature at the end of the pipes is more likely to stabilize or drop slightly.



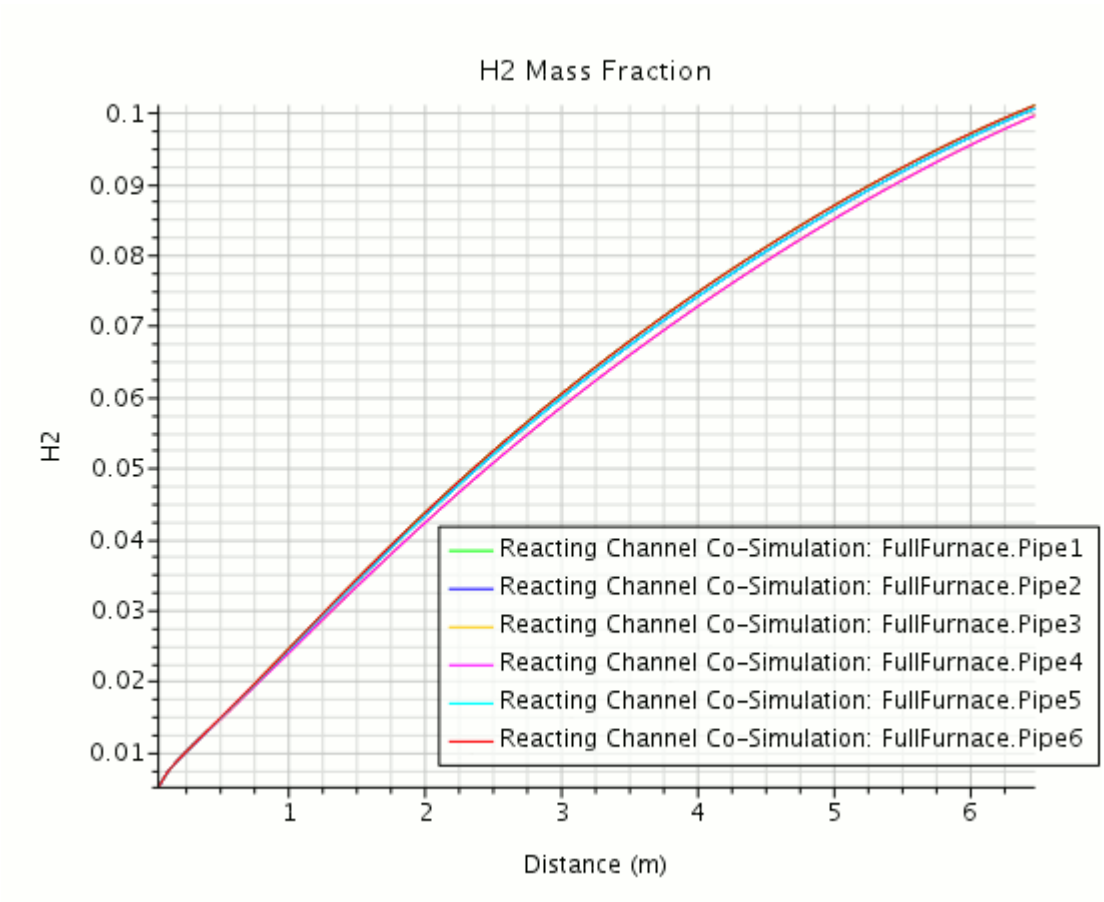
11. In the *Output* window, click the *CH4 Mass Fraction* tab.

    The mass fraction of CH4 drops along the length of the reacting channel tubes as the reactions within the channels consume CH4.



12. In the *Output* window, click the *H2 Mass Fraction* tab.

    The mass fraction of H2 increases along the length of the reacting channel tubes as the reactions within the channels produce H2.



Bibliography

[809] *CFD simulations of steam cracking furnaces using detailed combustion mechanisms. G.D. Stefanidis, B. Merci, G.J. Heynderickx, G.B. Marin. Computers and Chemical Engineering 30 (2006) 635–649.*

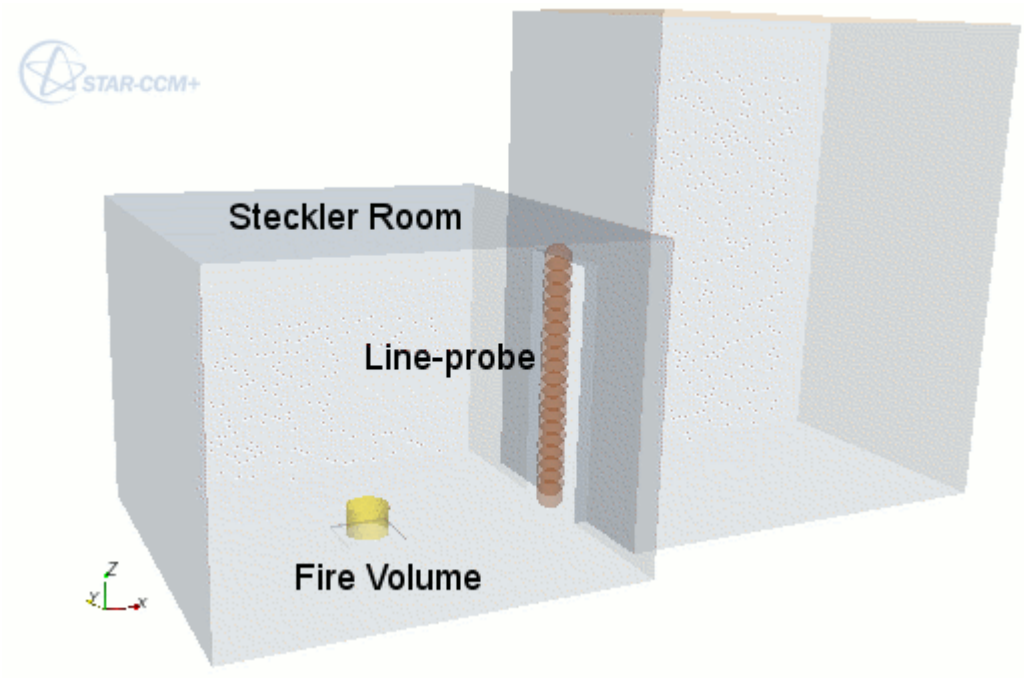
Fire and Smoke Wizard: Steckler Room

This tutorial demonstrates the Fire and Smoke Wizard. The geometry and fire settings closely resemble those Steckler reported, including radiation and wall heat absorption. This setup allows a preliminary comparison with experimental data.

Before you begin, take note of the following important Conventions in the **Fire and Smoke Wizard**:

- Gravity is in negative z-direction (z pointing up).
- Reference Density is 1.2 Kg/m^3.
- Reference Altitude is 0, 0, 0 m.

The problem geometry is shown below.



In this scene, the location of the line-probe corresponds to the experimental measurement stack.

Contents:

Prerequisites
Importing the Geometry
Assigning Parts to Regions
Generating the Volume Mesh
Setting up Fire Properties
Reviewing Model Settings
Setting up Solver Parameters and Stopping Criteria
Visualizing the Solution
Running the Simulation

Analyzing Results
Settings for Steckler Room Fire Validation
Comparison with Measurements
Adjusting Stopping Criteria and Continuing
Analyzing Validation Results
Summary

Prerequisites

The instructions in the Fire and Smoke Wizard: Steckler Room tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Importing the Geometry

To set up the STAR-CCM+ simulation, launch a simulation and import the supplied geometry.

1. Create a directory for the tutorial called `stecklerFire`.
2. Navigate to the `[INSTALL_DIR]/doc/startutorialsdata/reactingFlow/data` directory and copy the following files to your working directory:
   - `stecklerRoomFire.x_t`
   - `DoorUvelExpt.csv`
   - `DoorTempExpt.csv`
3. Launch STAR-CCM+
4. Start a simulation.
5. Select **File** > **Import** > **Import Surface Mesh...**.
6. In the *Open* dialog, navigate to your working directory, select `stecklerRoomFire.x_t`, and click **Open**.
7. In the *Import Surface Options* dialog, click **OK** to accept the default options.

   The *Geometry Scene 1* display appears in the *Graphics* window.
8. Save the simulation as `roomfire.sim`.

Assigning Parts to Regions

To define the computational domain, assign the parts to regions. In this tutorial, there are two regions, one for each part.

1. Expand the **Geometry** > **Parts** node and multi-select the **Fire** and **Room** nodes.
2. Right-click on one of the selections and choose **Assign Parts to Regions...**.
3. In the *Assign Parts to Regions* dialog, select:
   a) **Create a Region for Each Part**
   b) **Create a Boundary for Each Part Surface**
4. Click **Apply** then **Close**.

   Two new regions are created with boundaries that define the outlet and the interface between the two regions.

Set the outlet boundary as a Pressure Outlet:

5. Expand the **Regions** > **Room** > **Boundaries** node.
6. Select the **Outlet** node, and set *Type* to **Pressure Outlet**.

   In the *Graphics* window, the surface that corresponds to this boundary changes to an orange color.

Generating the Volume Mesh

Set up and generate the volume mesh.

1. Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
2. In the *Create Automated Mesh Operation* dialog:
   a) Select **Fire** and **Room** from the *Parts* list.
   b) Select the following meshers, in order:

Group	Mesher
Surface Meshers	**Surface Remesher**
Core Volume Meshers	**Polyhedral Mesher**
Optional Boundary Layer Meshers	**Prism Layer Mesher**

   An automated mesh operation is added to the **Operations** node.
3. Select the **Automated Mesh** > **Default Controls** > **Base Size** node and set *Value* to `0.1 m`.
4. Select the **Default Controls** > **Minimum Surface Size** node and set *Percentage of Base* to `50.0`.

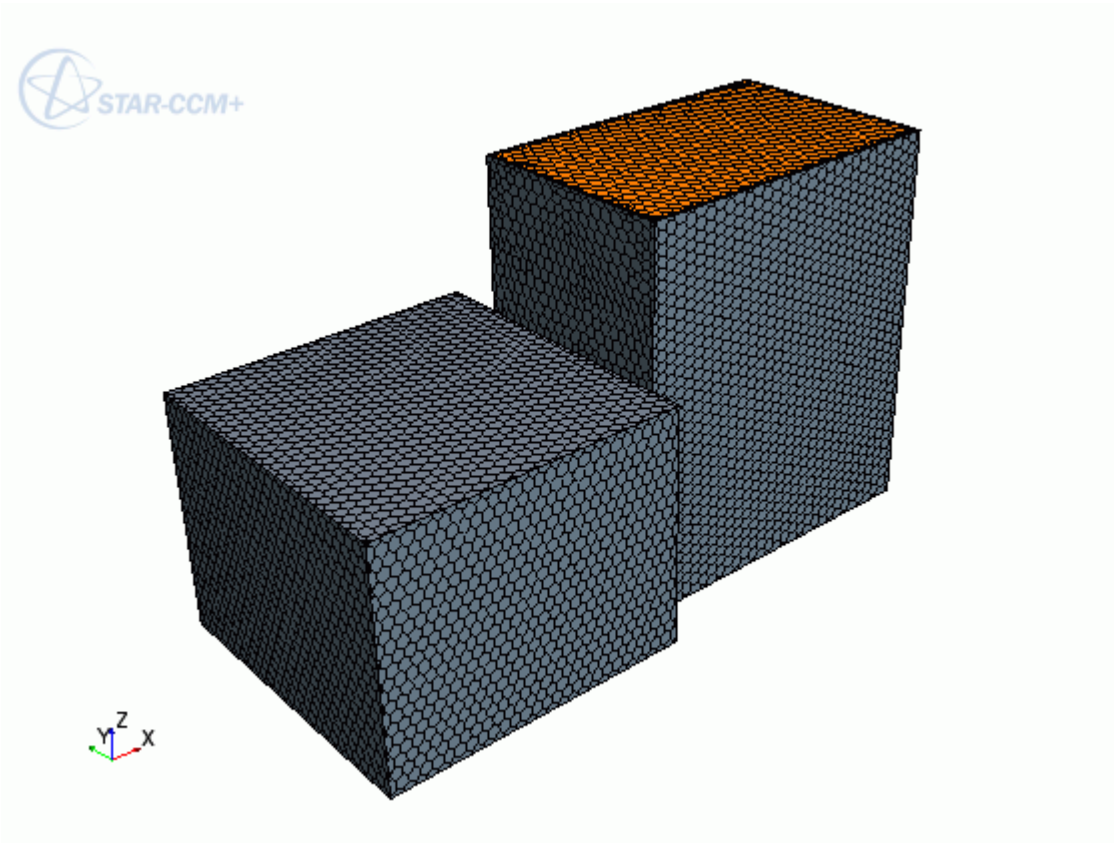
   This change prevents the mesher from creating small cells within the fire part.
5. Deactivate prism layers on the interface between the fire and the room:
   a) Right-click the **Automated Mesh** > **Custom Controls** node and select **New** > **Surface Control**.
   b) Select the **Custom Controls** > **Surface Control** node and set *Part Surfaces* to **Fire.Interface** and **Room.Interface**.
   c) Select the **Surface Control** > **Controls** > **Prism Layers** node and set *Prism Layers* to **Disable**.
6. Force the volume mesher to create cells within the fire part that are similar in size to cells in the rest of the mesh:
   a) Right-click the **Automated Mesh** > **Custom Controls** node and select **New** > **Part Control**.
   b) Select the **Custom Controls** > **Part Control** node and set *Parts* to **Fire**.
   c) Select the **Part Control** > **Controls** > **Mesh Density** node and set *Mesh Density* to **Custom**.

d) Select the **Part Control** > **Values** > **Mesh Density** node and set both *Density* and *Growth Factor* to `0.2`.

Generate the volume mesh:

7. Click **(Generate Volume Mesh)**.
   When the volume mesh is generated, the message `Volume Meshing Pipeline Completed` is displayed in the *Output* window. The number of Cells, Faces, and Vertices are also reported.
8. **Create a mesh scene.**

   The volume mesh is displayed below.

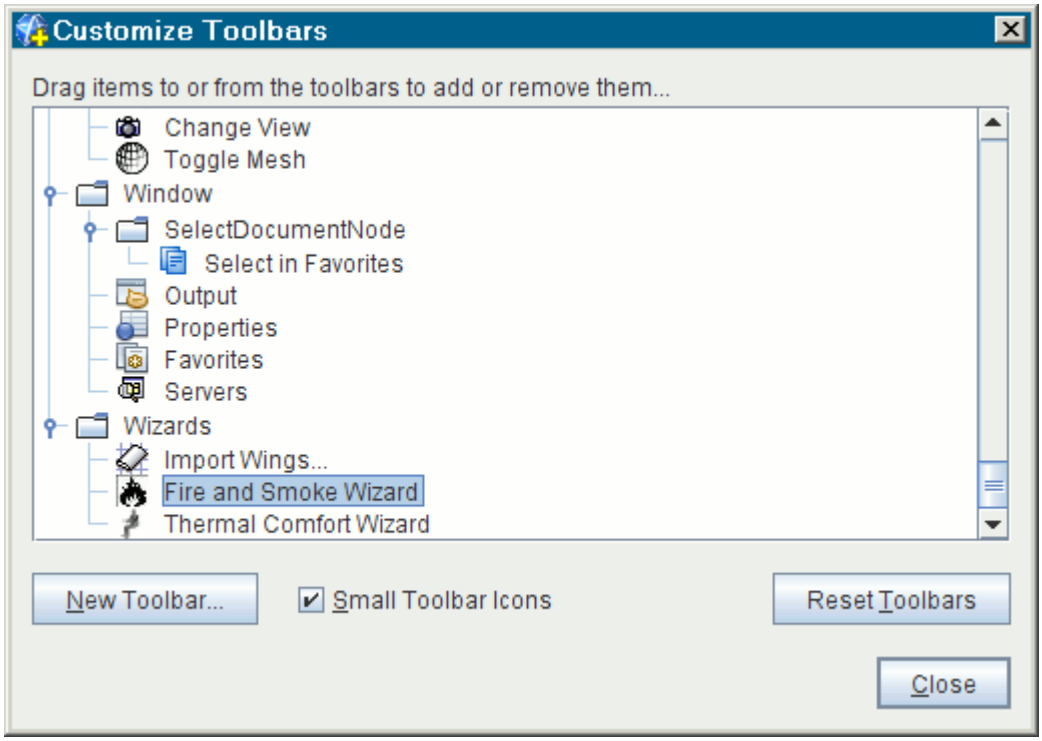


Setting up Fire Properties

Set up the properties of the fire using the **Fire and Smoke Wizard**.

To access the wizard:

1. In the menu bar, select **Window** > **Toolbars** > **Customize**.
2. In the *Customize Toolbars* dialog, scroll down to locate the **Fire and Smoke Wizard**.

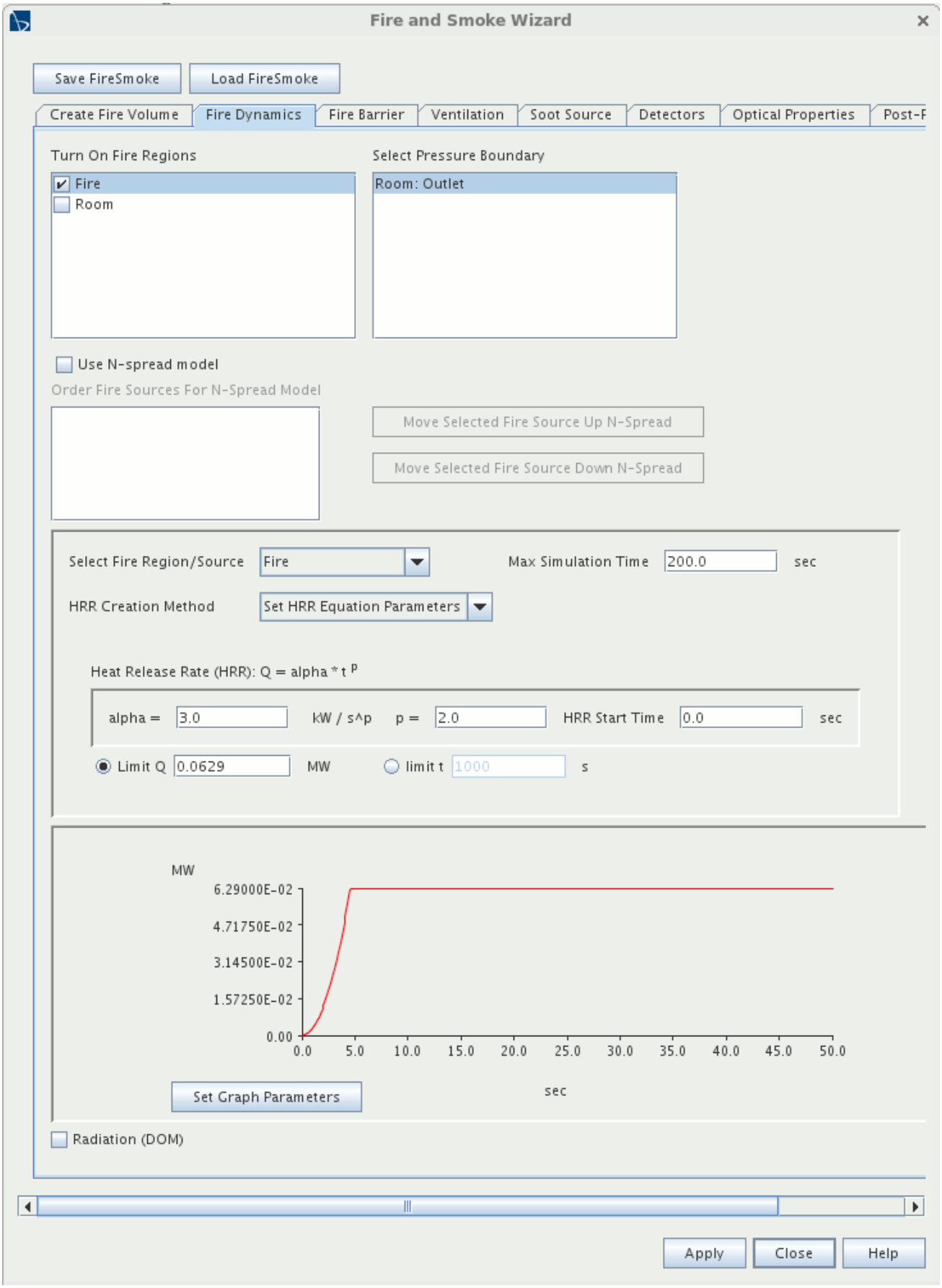


3. Drag and drop the **Fire and Smoke Wizard** icon onto the end of one of the toolbars near the top of the workspace.
4. In the *Customize Toolbars* dialog, click **Close**.
5. To open the *Fire and Smoke Wizard* dialog, click (**Fire and Smoke Wizard**).
6. Click the *Fire Dynamics* tab and set the following properties:

Property	Setting
Turn On Fire Regions	**Fire**
Select Pressure Boundary	**Room: Outlet**
Max Simulation Time	`200 sec`
alpha =	`3`
p =	`2`
Limit Q	`0.0629 MW`

7. Click **Set Graph Parameters**.
8. In the *Graph Parameters Dialog*, set *End X Range* to `50` and click **OK**.

When these steps are complete, the dialog appears as follows:


Fire and Smoke Wizard
Save FireSmoke
Load FireSmoke
Create Fire Volume
Fire Dynamics
Fire Barrier
Ventilation
Soot Source
Detectors
Optical Properties
Post-F
Turn On Fire Regions
Fire
Room
Select Pressure Boundary
Room: Outlet
Use N-spread model
Order Fire Sources For N-Spread Model
Move Selected Fire Source Up N-Spread
Move Selected Fire Source Down N-Spread
Select Fire Region/Source
Fire
Max Simulation Time
200.0
sec
HRR Creation Method
Set HRR Equation Parameters
Heat Release Rate (HRR): Q = alpha * t p
alpha =
3.0
kW / s^p
p =
2.0
HRR Start Time
0.0
sec
Limit Q
0.0629
MW
limit t
1000
s
MW
6.29000E-02
4.71750E-02
3.14500E-02
1.57250E-02
0.00
0.0
5.0
10.0
15.0
20.0
25.0
30.0
35.0
40.0
45.0
50.0
sec
Set Graph Parameters
Radiation (DOM)
Apply
Close
Help

9. Click **Apply**.

In this process, a fire source, height-varying pressure boundary setting, and scalar source are all set-up automatically.

10. Save the simulation.

Reviewing Model Settings

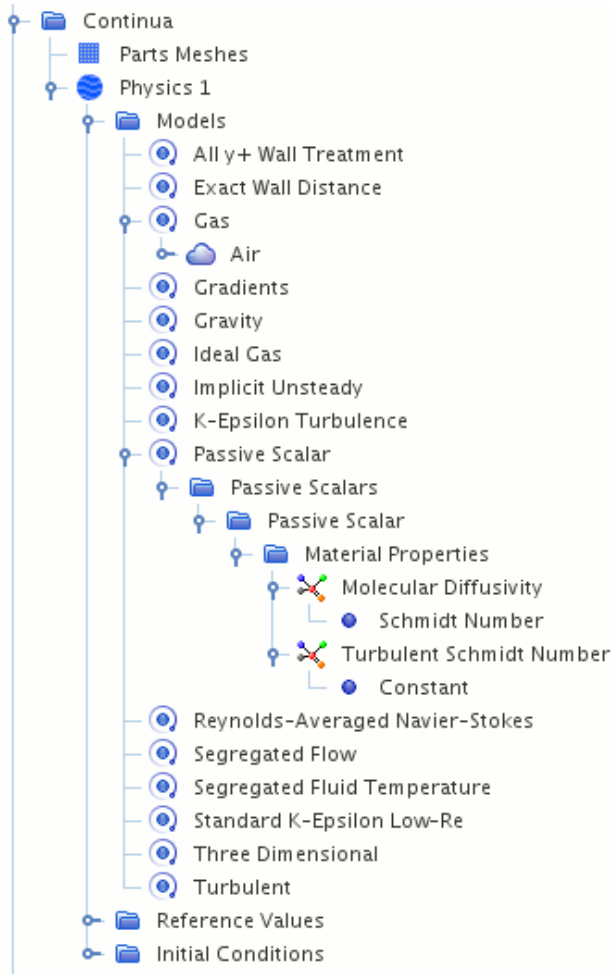
This tutorial describes an unsteady problem in which:

- A three-dimensional grid is employed.
- The fluid is a slightly compressible gas (air) that obeys the ideal gas equation of state.
- The flow is turbulent and non-isothermal.
- The $k - \varepsilon$ low Re model is used for representing turbulence effects.
- The segregated flow solver is used.
- Gravity effects are taken into account.

All of the above conditions are automatically set up as properties of the fire.

To make sure that the model setup is accurate, you can inspect the selected models and edit them as necessary:

1. Open the **Continua** > **Physics 1** > **Models** node.



In this case, the model setup is correct and no change is required.

Setting up Solver Parameters and Stopping Criteria

Before running the analysis, specify the maximum physical time over which the simulation is run.

Set the time step and the temporal (and spatial) discretization schemes. Also reduce the maximum number of iterations that are carried out for each time step.

1. Select the **Solvers** > **Implcit Unsteady** node and set *Time-Step* to `1.0 s`.
2. Select the **Stopping Criteria** > **Maximum Inner Iterations** node and set *Maximum Inner Iterations* to `5`.

Visualizing the Solution

To view the progress of the simulation as it runs, set up scalar scenes using a section plane.

Create a scalar scene to display the temperature on the plane section.

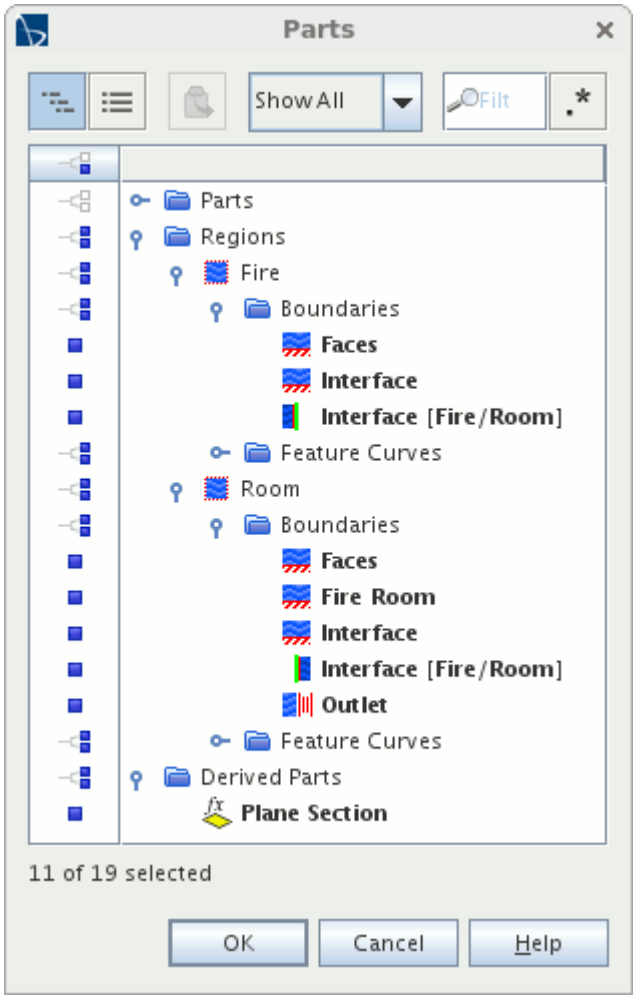
1. **Create a scalar scene.**

Create the plane section:

2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane...**, and set the following properties:

Property	Setting
Input Parts	**Fire** **Room**
origin	`[0.0, 0.0, 0.0] m`
normal	`[0.0, 1.0, 0.0] m`
Display	**Existing Displayer** > **Scalar 1**

3. Click **Create** then **Close**.
4. Expand the **Scalar Scene 1** > **Displayers** > **Outline 1** node.
5. Right-click the **Parts** node and select **Edit**.
6. In the *Edit* dialog, select all regions and derived parts.

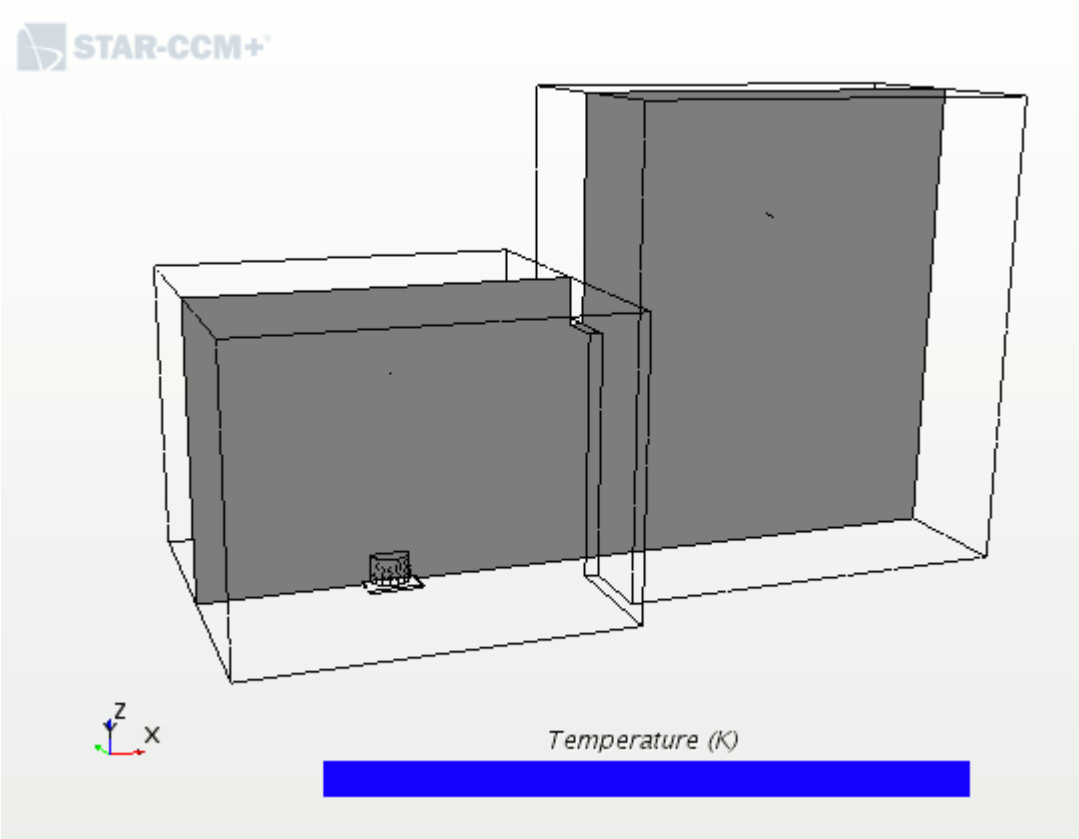


7. Click **OK**.
8. Select the **Scalar Scene 1** > **Displayers** > **Outline 1** node and activate **Feature Lines**.

9. Expand the **Displayers** > **Scalar 1** node and set the following properties:

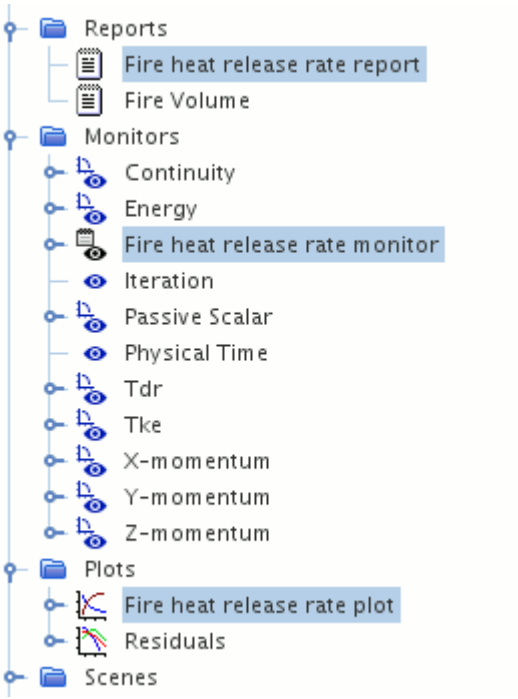
Node	Property	Setting
Scalar 1	*Contour Style*	**Smooth Filled**
└ **Scalar Field**	*Function*	**Temperature**

10. Use the mouse buttons to position the geometry as shown below:



During the Fire properties set up, reports, monitors, and plots for the fire heat release rate are all set up automatically.

11. Open the **Reports**, **Monitors**, and **Plots** nodes.



12. Save the simulation.

Running the Simulation

Run the simulation.

1. To start the solver, click **(Run)**.

   The analysis now starts and the results, in the form of a graph of residuals vs. iteration number, is displayed automatically in the *Graphics* window. An example of a residual plot is shown in a separate part of the User Guide.

   The solution stops after 200 time-steps.

Analyzing Results

Analyze the contours of temperature for the converged solution.

The contours of temperature on the plane section that is displayed in *Scalar Scene 1* are dynamically updated with each time-step.

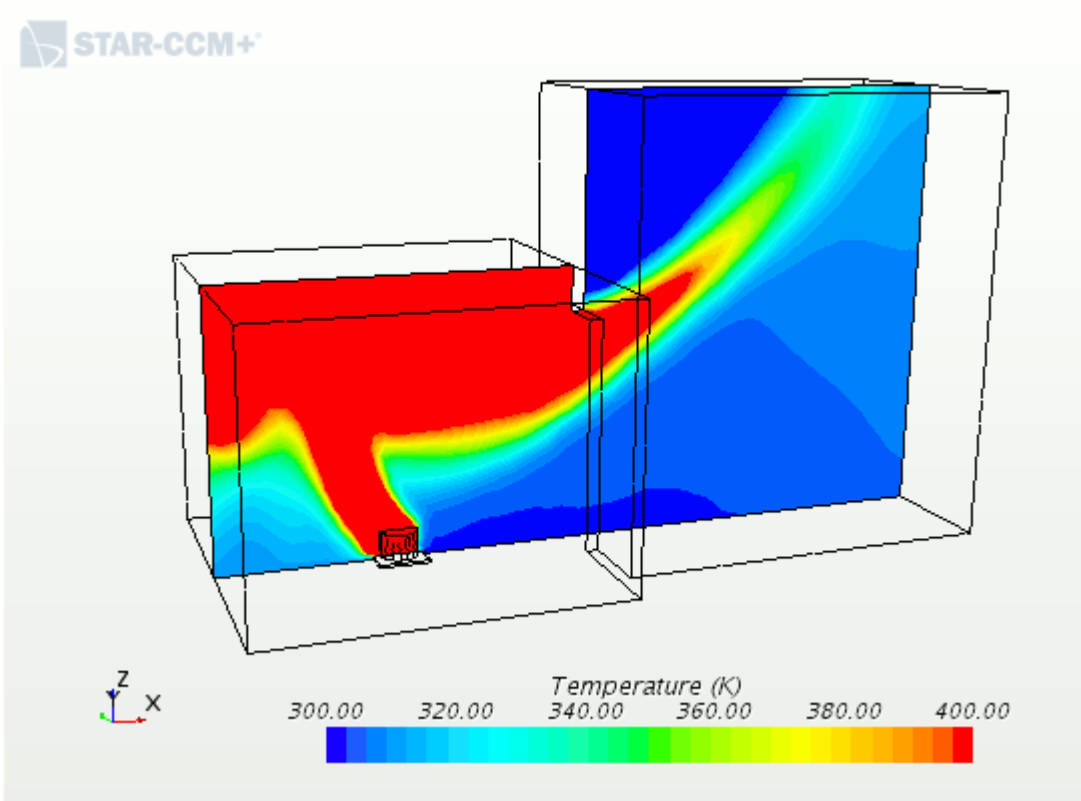
Adjust the temperature range that is displayed in the scalar scene to help you visualize the results.

1. Expand the **Scenes** > **Scalar Scene 1** > **Displayers** > **Scalar 1** node.
2. Select the **Scalar Field** node and set the following properties:

Property	Value
Min	300

Property	Value
Max	400
Clip	**Off**

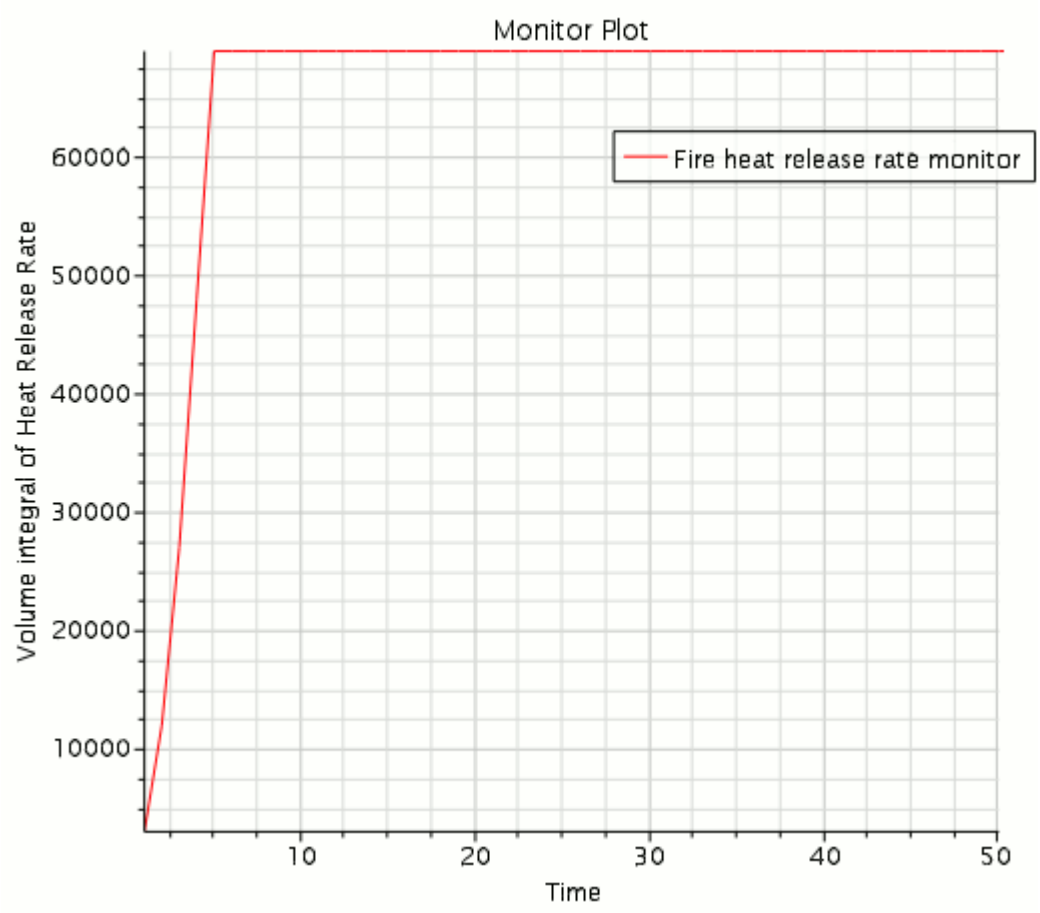
A scene similar to the one shown below is displayed.



Examine the Heat Release Rate in the Fire region:

3. Expand the **Plots** node.
4. Right-click the **Fire heat release rate plot** node and select **Open**.
5. Select the **Fire heat release rate plot** > **Axes** > **Bottom Axis** node and set *Maximum* to 50.

   A scene similar to the one shown below is now displayed.



6. Save the simulation.

Settings for Steckler Room Fire Validation

Set heat transfer coefficients for the surfaces that represent the walls and ceiling.

In the Steckler room fire experiment, the walls and ceiling are composed of conduction brick of thickness 0.1m and conductivity of 0.69W/m-K. Set an effective heat transfer coefficient of 6.9 W/m^2-K for each of the five shell surfaces representing the walls and ceiling of the room.

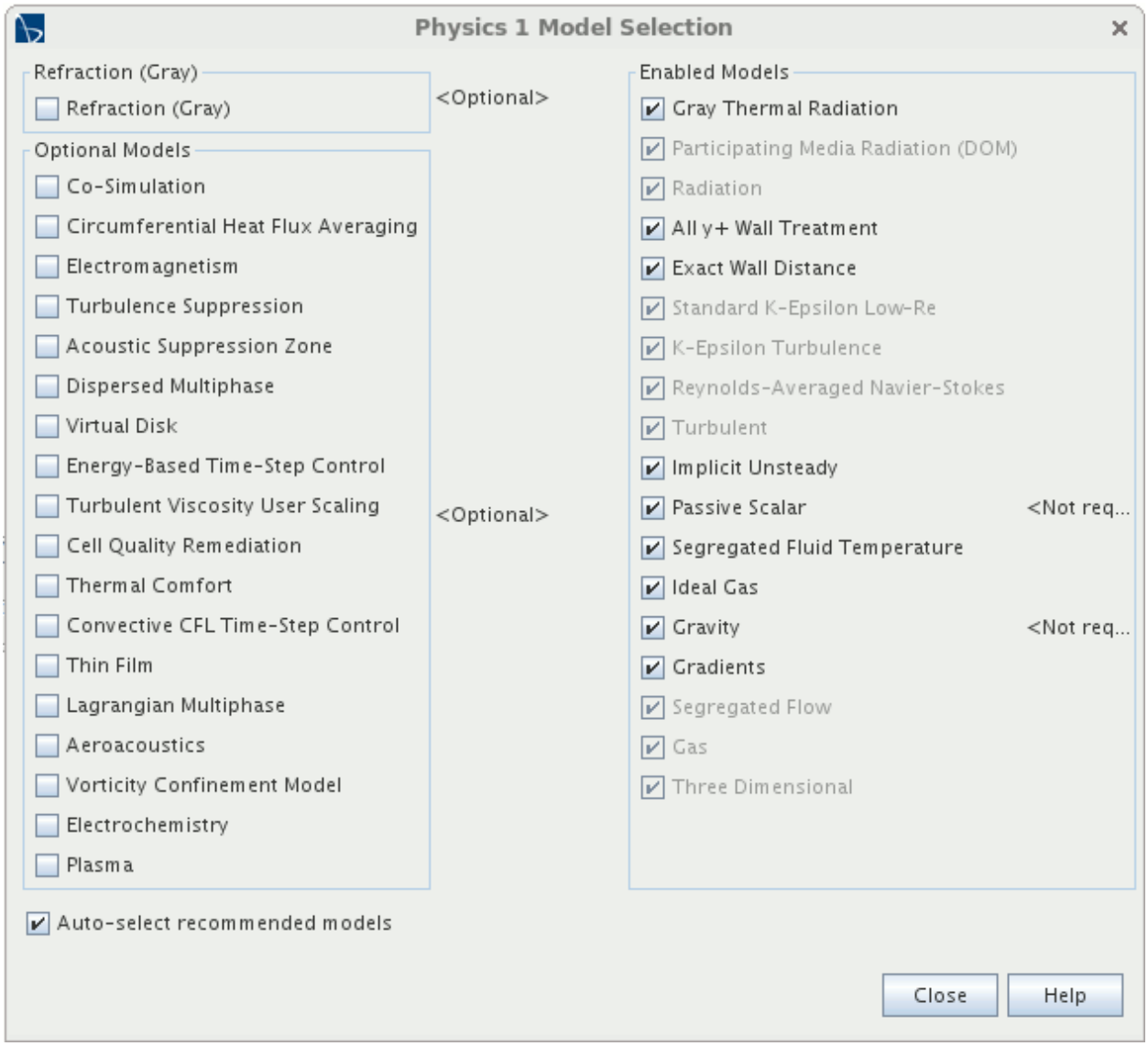
1. Expand the **Regions** > **Room** > **Boundaries** > **Fire Room** node.
2. Select the **Physics Conditions** > **Thermal Specification** node and set *Condition* to **Convection**.
3. Select the **Physics Values** > **Heat Transfer Coefficient** node and set *Value* to `6.9 W/m^2-K`.

Switch on the radiation model and set the gray-gas absorptivity to a field function based on temperature:

4. Open the **Continua** > **Physics 1** node.
5. Right-click the **Models** node and select **Select Models**.
6. In the *Physics Model Selection* dialog, select the following models in order:

Group Box	Model
Optional Models	**Radiation**
Radiation	**Participating Media Radiation (DOM)**
Radiation Spectrum (Participating)	**Gray Thermal Radiation**

The *Physics Model Selection* dialog has the following physics models enabled:



7. Click **Close**
8. Select the **Models** > **Gas** > **Air** > **Material Properties** > **Absorption Coefficient** node and set the following properties:

Node	Property	Setting
Absorption Coefficient	*Method*	**Field Function**

Node	Property	Setting
└ **Field Function**	*Scalar Function*	**AbsorTempFunction**

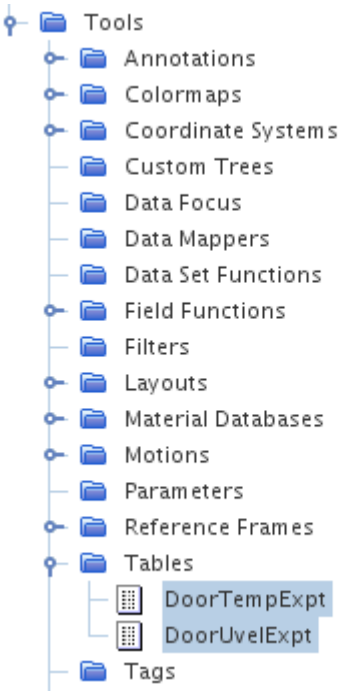
9. Save the simulation.

Comparison with Measurements

To compare the predictions with the experimental data, first load the files containing the vertical temperatures and horizontal velocities at the door.

1. Expand the **Tools** node.
2. Right-click the **Tables** node and select **New Table** > **File Table**.
3. In the *Open* dialog, navigate to your `stecklerFire` directory and select the `DoorTempExpt.csv` and `DoorUvelExpt.csv` files.
4. To read the data, click **Open**.

   The `DoorTempExpt.csv` and `DoorUvelExpt.csv` files are now listed under the **Tables** node.



Create a line probe against which to plot the predicted profiles:

5. Click the *Geometry Scene 1* tab at the top of the *Graphics* window to activate the scene.
6. Right-click the **Derived Parts** node and select **New Part** > **Probe** > **Line...**.
7. Set the following properties for the line probe:

Property	Setting
Input Parts	**Regions** > **Room**

Property	Setting
Point 1	`[1.4, 0.0, 0.0] m`
Point 2	`[1.4, 0.0, 1.83] m`
Resolution	`20`
Display	**New Geometry Displayer**

8. Click **Create** then **Close**.

   A **Derived Parts** > **Line Probe** node appears in the simulation tree.

Create a plot for temperature comparison with measurement using the following settings:

9. Right-click the **Plots** node and select **New Plot** > **X-Y Plot**.
10. Rename the **XY Plot 1** node to `Door Temperature`.
11. Edit the **Plots** > **Door Temperature** node and set the following properties:

Node	Property	Setting
Door Temperature	*Title*	`Door Temperature`
	Parts	**Derived Parts** > **Line Probe**
└ **X Type** > **Vector Quantity**	*Value*	`[0, 0, 1] m`
└ **Y Types** > **Y Type 1**	*Smooth Values*	Activated
└ **Scalar Function**	*Field Function*	**Temperature**
└ **Line Probe**	*Sort Plot Data*	Activated
└ **Line Style**	*Style*	**Solid**
└ **Axes**		
└ **Left Axis** > **Title**	*Title*	`Temperature [K]`
	Lock Title Name	Activated
└ **Bottom Axis** > **Title**	*Title*	`Position (m)`
	Lock Title Name	Activated
└ **Legend**	*x-Position*	`0.8`
	y-Position	`0.1`

12. To re-orient the plot axes:
    a) Right-click the **Axes** node and select **Orient Axes**.
    b) In the *Orient Axes* dialog, click ↔ then ↻.

       The Temperature axis appears on the bottom side of the plot and the Position axis appears on the left side of the plot.

13. Right-click the **Plots** > **Door Temperature** > **Data Series** node and select **Add Data**.
14. In the *Add Data Providers to Plot* dialog, select **DoorTempExpt** and click **OK**.
15. Edit the **Data Series** > **DoorTempExpt** node and set the following properties:

Node	Property	Setting
DoorTempExpt	*Legend Name*	`Experimental Temp.`
	Lock Legend Name	Activated
	Table	**DoorTempExpt**
	X Column	**Y**
	Y Column	**Temperature [K]**
	Sort Plot Data	Activated
└ **Symbol Style**	*Shape*	**Empty Square**
	Color	**Blue**

The temperature comparison graph setup is now complete. Create a similar plot for the door velocity.

16. Copy and paste the **Door Temperature** plot and rename as `Door Velocity`.
17. Edit the **Door Velocity** node and set the following properties:

Node	Property	Setting
Door Velocity	*Title*	`Door Velocity`
└ **Y Types**		
└ **Y Type 1** > **Scalar Function**	*Field Function*	**Velocity[i]**
└ **Axes**		
└ **Bottom Axis** > **Title**	*Title*	`Velocity (m/s)`

18. Expand the **Door Velocity** > **Data Series** > **DoorTempExpt** node and set the following properties:

Property	Setting
Legend Name	`Experimental Velocity`
Table	**DoorUvelExpt**
X Column	**Y**
Y Column	**V [m/s]**

The velocity comparison graph setup is now complete.

Adjusting Stopping Criteria and Continuing

Increase the maximum physical time of the simulation.

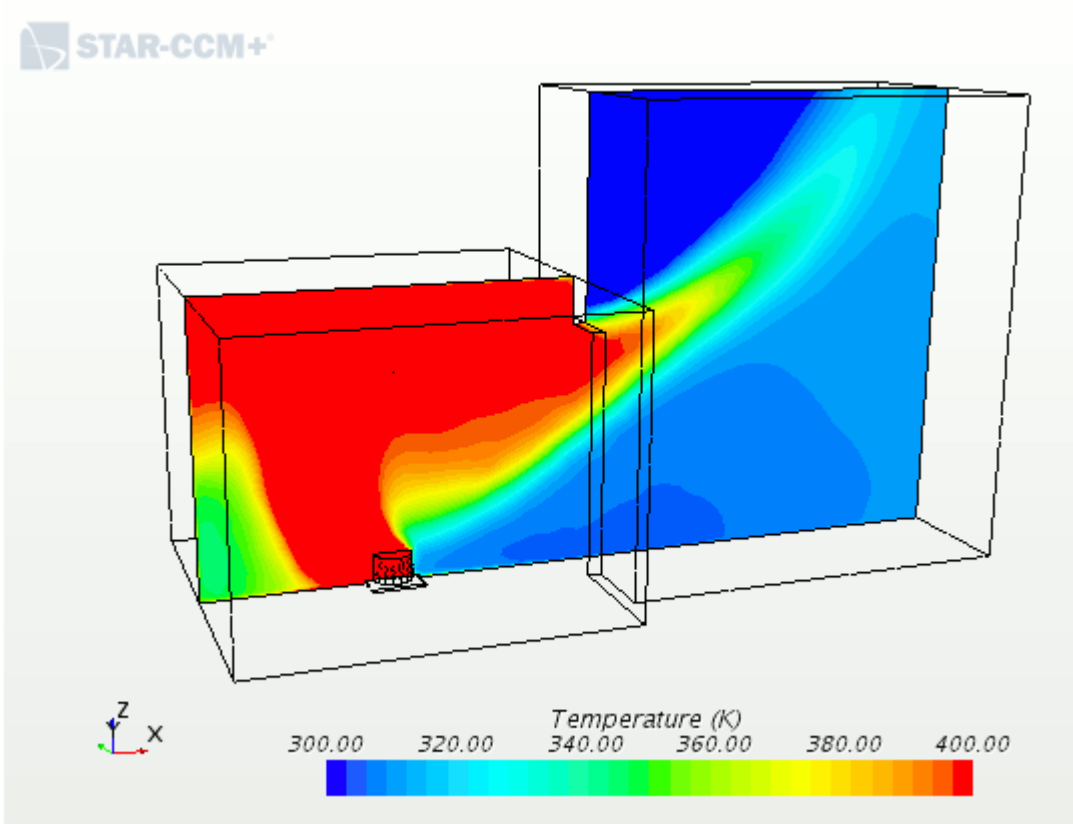
To allow you to solve for a further 200 time-steps, increase the maximum physical time to 400 seconds.

1. Select the **Stopping Criteria** > **Maximum Physical Time** node and set *Maximum Physical Time* to `400.0 s`.
2. Save the simulation as `roomFireValidation.sim`.
3. To start the solver, click **(Run)**.
   The solution stops after 400 time-steps.

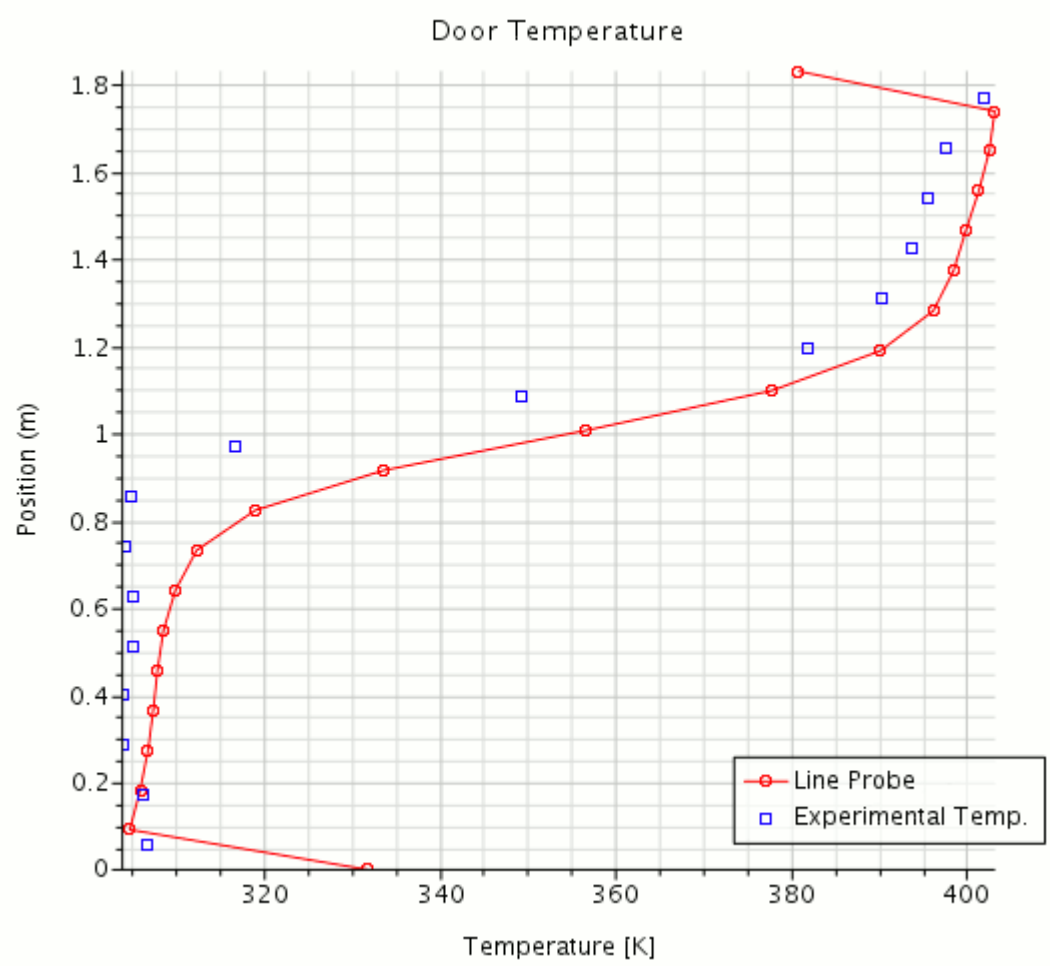
Analyzing Validation Results

This section of the tutorial covers the analysis of the validation results.

The following scene shows the solution after 400 time-steps, plotted in the temperature range of 300 K to 400 K.

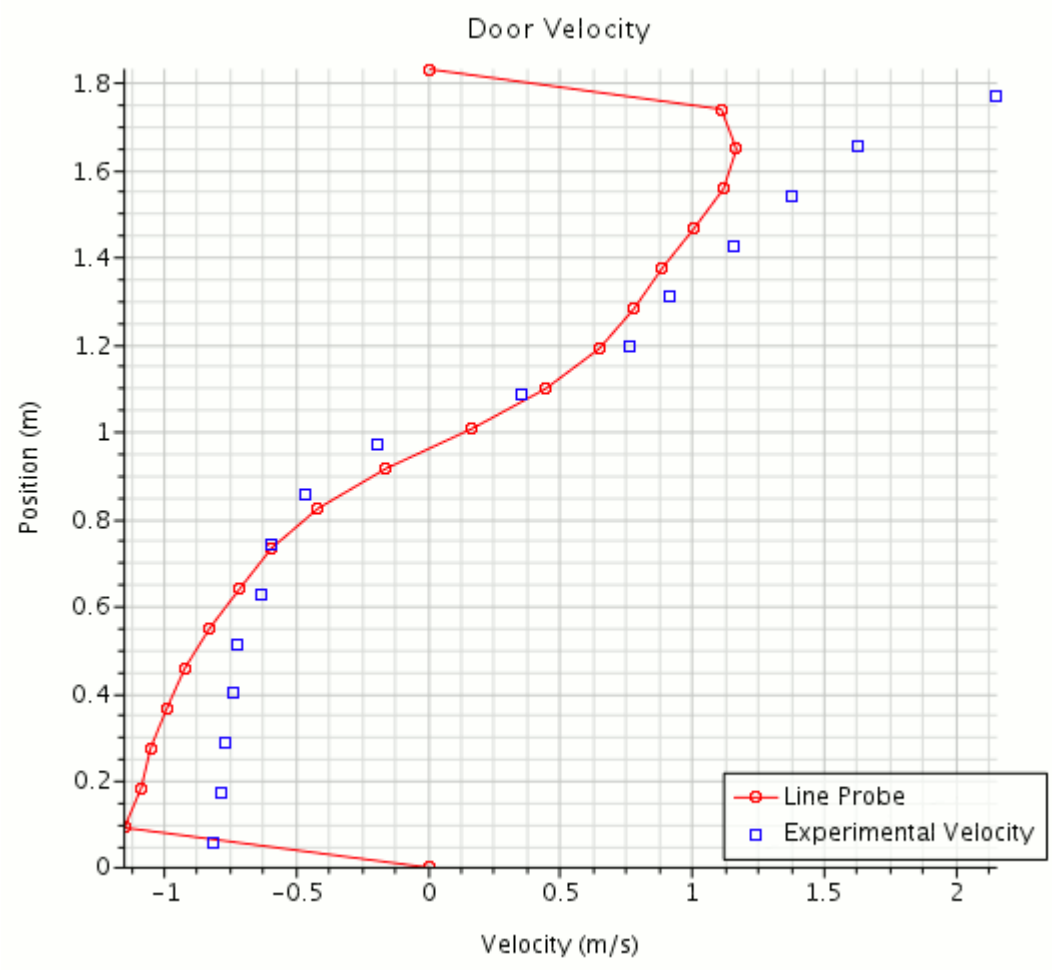


1. Open the *Door Temperature* plot which shows comparison at the defined line-probe location.



The temperature profile from the simulation is in good agreement with the profile based on experimental data.

2. Open the *Door Velocity* plot which shows comparison at the defined line-probe.



The simulated velocity profile shows good agreement with the experimental data up to a height of 1.5 m. After this point, the experimental data indicates that the maximum velocity in the boundary layer is higher than the simulation predicts. One cause for this difference could be an insufficient density of cells in the upper region of the doorway. Good simulation practice would be to reduce the volume mesh **Base Size** to 0.05 m, remesh the domain, and rerun the simulation using the finer mesh. Using a finer mesh size gives closer agreement between the simulated and experimental results.

Summary

This tutorial demonstrated how to use the Fire and Smoke Wizard.

The steps covered were:

- Invoking the Fire and Smoke Wizard.
- Defining the simulation models.
- Setting up radiation properties.
- Analyzing the results using the built-in plotting and visualization facilities, including the XY Plot, comparing the experimental data with the predictions.

Solid Stress

The tutorials in this set illustrate various STAR-CCM+ features for computing deformation, strain, and stresses in solid regions. They also show how such computations can be coupled to the fluid behavior in an analysis of fluid-structure interaction.

Contents:

Linear Stress Analysis: Cantilever I Beam
Plane Stress: Plate with Circular Hole in a Tensile Field
Thermal Strain: Exhaust Manifold
Fluid-Structure Interaction: Vibrating Pipe

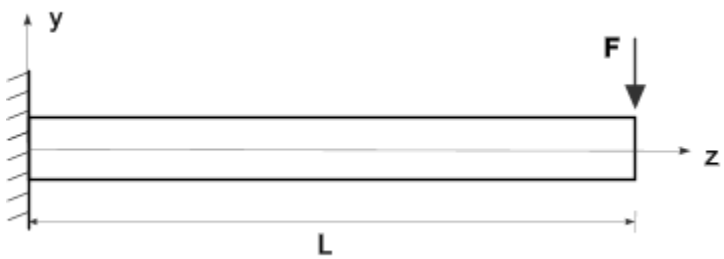
Linear Stress Analysis: Cantilever I Beam

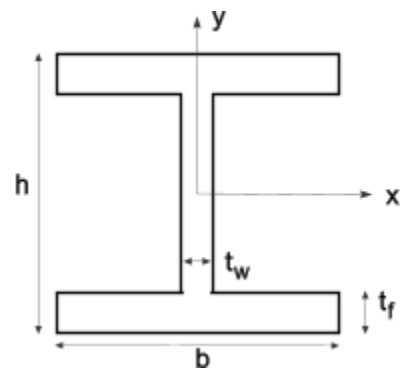
This tutorial demonstrates the basic concepts and workflow for linear FE (Finite Element) structural analyses in STAR-CCM+.

The case under study is a cantilevered I beam loaded at its free end. The displacements that result from loading are assumed to be small, so that the load-displacement relationship remains linear. For simplicity, the beam is assumed weightless.

In this tutorial, you compare the numerical solution, obtained with 3D solid hexahedral elements in STAR-CCM+, with the analytical solution given by Euler-Bernoulli beam theory. This theory is valid for slender beams with $\frac{L}{h} > 10$.

The mathematical model is schematized below.





Geometry		Material Properties (Ferretic Steel)/Load	
Flange thickness	$t_f = 8.5 \ \ \mathrm{mm}$	Load	$\lvert F \rvert = 800 \ \ \mathrm{N}$
Web thickness	$t_w = 5.6 \ \ \mathrm{mm}$	Young's Modulus	$E = 2 \cdot 10^5 \ \ \mathrm{MPa}$
Beam height	$h = 200 \ \ \mathrm{mm}$	Poisson Ratio	$\nu = 0.3$
Beam length	$L = 4000 \ \ \mathrm{mm}$	Density	$\rho = 7500 \ \ \mathrm{kg/m^3}$
Flange length	$b = 100 \ \ \mathrm{mm}$		

For this problem, Bernoulli-Euler's beam theory gives the values of the deflection at the free end, and the maximum value of the stress tensor component σ_{zz}, as:

$$\delta_{y,\ max} = \frac{FL^3}{3EI_x} \approx 4.63 \ \ \mathrm{mm}$$
$$\sigma_{zz,\ max} = \frac{M}{I_x} \cdot \frac{h}{2} \approx 17.3 \ \ \mathrm{MPa} \tag{4169}$$

where $I_x = \int_A y^2 \, dA \approx 1845.6 \cdot 10^4 \ \ \mathrm{mm}^4$ is the moment of inertia about the x axis, M is the bending moment, and all the remaining quantities are listed in the table above.

Contents:

Prerequisites
Importing the Geometry
Assigning the Imported Part to a Region
Selecting the Physics Models and Materials
Applying Loads and Constraints Using Segments
Generating the Hexahedral Mesh
Preparing Analysis Objects
Running the Simulation
Visualizing the Solution
Summary

Prerequisites

The instructions in the Linear Stress Analysis - Cantilever I Beam tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Techniques	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Importing the Geometry

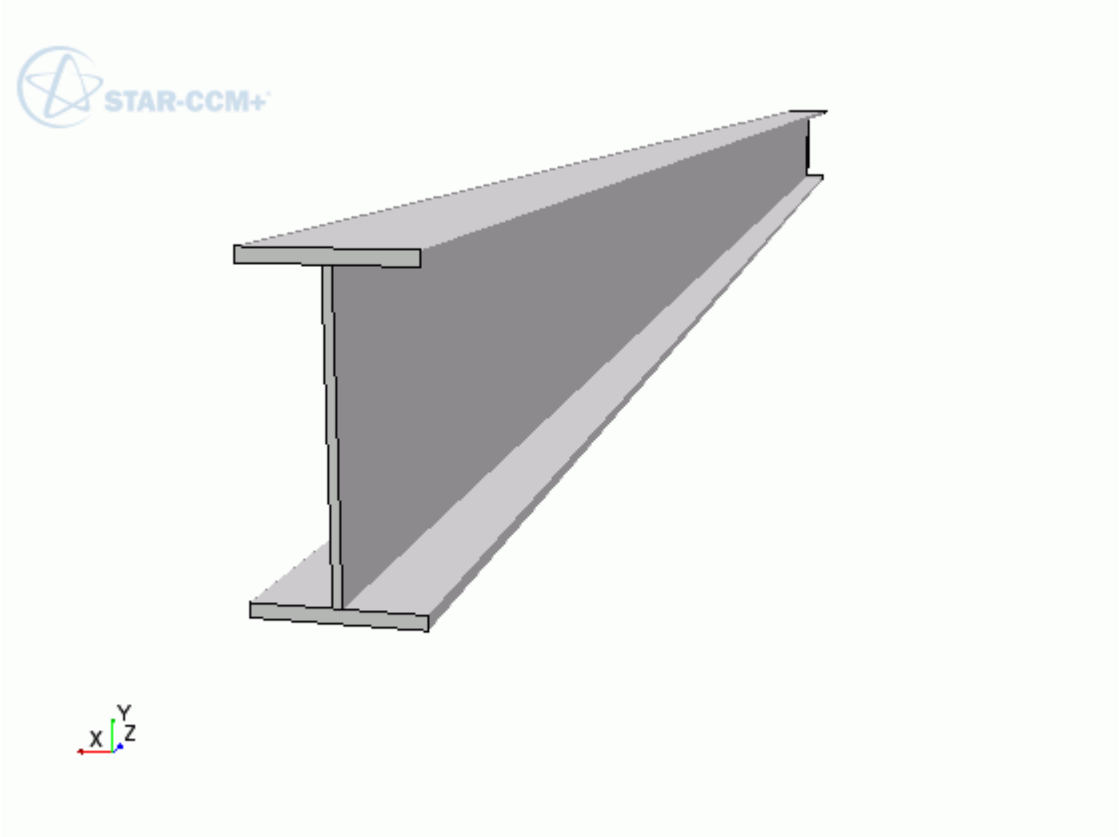
Launch STAR-CCM+ and import the 3D-CAD beam geometry that is provided for this tutorial.

1. Launch STAR-CCM+.
2. Start a new simulation.
3. Select **File** > **Import** > **Import Surface Mesh...**
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/solidStress/data`.
5. Select `Ibeam.x_b`, then click **Open**.
6. In the *Import Surface Options* dialog, click **OK**.

   The import process takes a few seconds, and STAR-CCM+ provides feedback on the process in the **Output** window. The imported geometry appears in a geometry scene in the **Graphics** window.

Save a convenient view of the geometry for later use:

7. Click (**Save-Restore-Select Views**) and select **Standard Views** > **(-Z) bottom**, then use the mouse to position the geometry as shown below.



8. Click (**Save-Restore-Select Views**) and select **Store Current View**.
9. Save the simulation as `IBeam.sim`.

Assigning the Imported Part to a Region

Before you generate a mesh for the imported part, the part must be assigned to a STAR-CCM+ region. As all the boundary conditions for this case are set directly on the part surfaces using segments, you can group all part surfaces under a single region boundary.

To assign the beam part to a region:

1. Right-click the **Geometry** > **Parts** > **Beam** node and select **Assign Parts To Regions...**

Instruct STAR-CCM+ to assign the beam part to a new region, and all the part surfaces to a single region boundary.

2. In the *Assign Parts to Regions* dialog, choose the following options:

Region option	**Create a Region for Each Part**
Boundary option	**Create a Boundary for Each Part** (default)

3. Click **Apply** then **Close**.

   STAR-CCM+ adds a region, **Beam**, under the **Regions** node.
4. Save the simulation.

Selecting the Physics Models and Materials

Select physics models to set up a static linear analysis and specify the beam material.

The physics models define the primary variables of the simulation and which mathematical formulation is used to generate the solution. In this simulation, the structure displacement and the internal stress distribution are computed using the linear **Sparse Direct Solver**, that is enabled when you select the **Solid Stress** model.

Note: As the **Solid Stress** model has specific mesh requirements, you are advised to select the physics models before you set up the mesh operations. Following this workflow prevents you from selecting incompatible meshers.

1. Create a physics continuum and rename it to `Beam Physics`.
2. For the physics continuum, **Beam Physics**, select the following models:

Group box	Model
Space	**Three Dimensional**
Time	**Steady**
Material	**Solid**
Optional Models	**Solid Stress**
	Material Law Models (selected automatically)

Select an appropriate material for the beam from the STAR-CCM+ material library:

3. Right-click the **Beam Physics** > **Models** > **Solid** > **Al** node and select **Replace with...**
4. In the *Replace Material* dialog, expand the **Material Databases** > **Standard** > **Solids** node and select **UNSS44600 (Ferretic Stainless Steel)**.

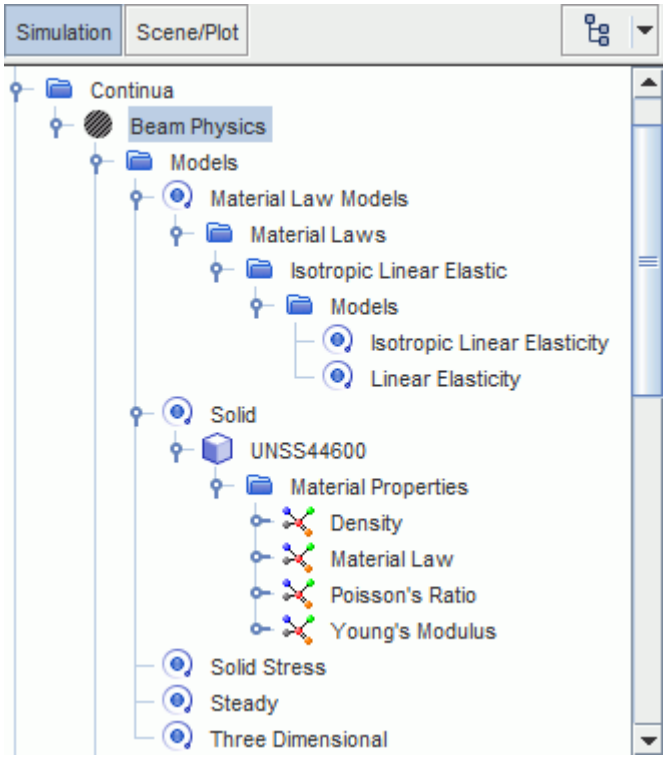
Define the material law that approximates the behavior of the material. In this tutorial, you assume that the material is isotropic linear elastic:

5. Expand the **Continua** > **Beam Physics** > **Models** > **Material Law Models** node.
6. Right-click the **Material Laws** node and select **New**.
7. Rename the **Material Laws** > **Material Law 1** node to `Isotropic Linear Elastic`.
8. For the **Isotropic Linear Elastic** law node, select the following models:

Group box	Model
Material Models	**Linear Elasticity**
Linear Elastic Material Models	**Isotropic Linear Elasticity**

Assign the isotropic linear elastic material law to the solid material:

9. Select the **Beam Physics** > **Models** > **Solid** > **UNSS44600** > **Material Properties** > **Material Law** node and set *Type* to **Isotropic Linear Elastic**.



Applying Loads and Constraints Using Segments

Using STAR-CCM+ segments, apply a shear load to the free end of the beam and constrain the end at the wall to prevent motion.

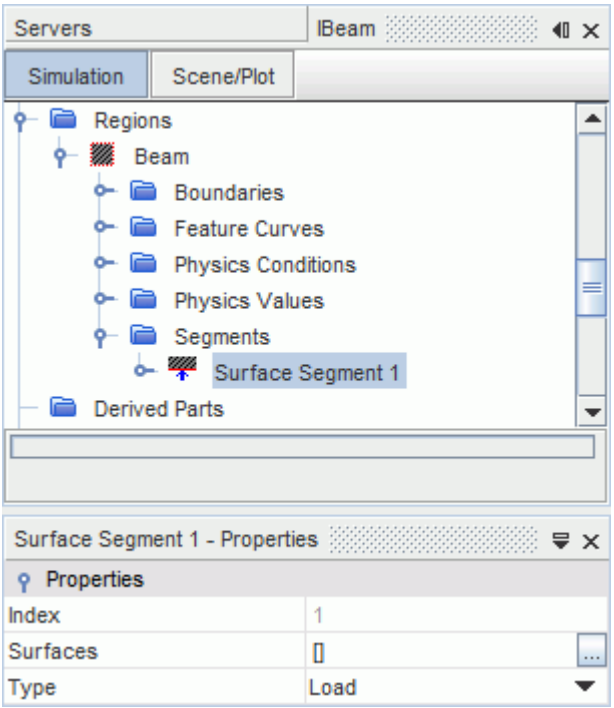
In the STAR-CCM+ stress analysis framework, loads and constraints are applied directly on part surfaces or vertices using segments. A segment is a collection of part surfaces or vertices subject to a common condition, such as a load or constraint. In this tutorial, you set up a segment to define a shear load on the free end of the beam, and two segments to constrain the end at the wall.

Start by creating a segment:

1. Right-click the **Regions** > **Beam** node and select **Create Segment**.

   STAR-CCM+ adds a new node, **Segments** > **Surface Segment 1**, under the **Beam** region node.

   All the segments that you create for a region are added within the **Segments** node for that region.



The segment *Surfaces* property allows you to specify a set of part surfaces. The *Type* property allows you to specify the type of condition (load or constraint) that is applied to the selected part surfaces.

Contents:

Applying a Shear Load
Applying Constraints

Applying a Shear Load

Set up the segment to apply a constant load on the free end surface of the beam.

Saint-Venant's Principle allows you to model the point load as a constant load distributed across the free end surface.

1. Rename the **Regions** > **Beam** > **Segments** > **Surface Segment 1** node to `Shear Load`.
2. Select the **Segments** > **Shear Load** node and set the properties as follows:

Property	Setting
Surfaces	**free side - lower flange**
	free side - upper flange
	free side - web
Type	**Load** (default)

The load method is defined through the **Solid Stress Loads** physics condition for the segment. The default method is set to **Force**, which is appropriate to this case.

To specify a shear force of 800 N in the negative y direction:

3. Select the **Shear Load** > **Physics Values** > **Force** node and set *Value* to `[0.0, -800.0, 0.0] N`.

   The segment is now fully defined.

   If you select the **Shear Load** segment node, the corresponding part surfaces are highlighted in the *Graphics* window.

Applying Constraints

Define two segments to apply constraints on the beam at the wall. Constrain the web in all directions, and the flanges in the normal direction only.

Stress analyses require the solid part to be fully constrained to prevent rigid body motions. Constraining the part so as to prevent any potential motion is essential for obtaining a numerical solution for displacements and stress throughout the structure.

When analyzing the computed stress distribution, examine the stress away from areas that are constrained in all degrees of freedom, as results in these areas may not give an accurate representation of the physical problem.

Constraining the beam wall end in all degrees of freedom would result in high stress concentrations at the flanges, that deviate from the maximum stress predicted by Euler-Bernoulli beam theory. To have a more accurate representation of the stress distribution at the flanges, constrain the web in all degrees of freedom, and the flanges in the axial direction only.

1. Create a segment and rename it to `Web Constraint`.
2. Edit the **Web Constraint** segment node and set the following properties:

Property	Setting
Surfaces	**wall side - web**
Type	**Constraint**

The constraining method is defined through the **Solid Stress Constraints** physics condition for the segment. The default method is **Fixed**, which is appropriate to this case, as it constrains the web in all directions.

3. Create a segment and rename it to `Flanges Constraint`.
4. Edit the **Flanges Constraint** segment node and set the following properties:

Property	Setting
Surfaces	**wall side - lower flange**
	wall side - upper flange
Type	**Constraint**

Specify a zero normal displacement constraint for this segment, to prevent movement normal to the plane of the flanges:

5. Select the **Flanges Constraint** > **Physics Conditions** > **Solid Stress Constraints** node and set *Method* to **Normal Displacement**.

   By default, the **Normal Displacement** value is set to zero.

The part is now fully constrained.

6. Save the simulation.

Generating the Hexahedral Mesh

In this tutorial, you mesh the beam with 3D solid hexahedral elements using the STAR-CCM+ Directed Mesher.

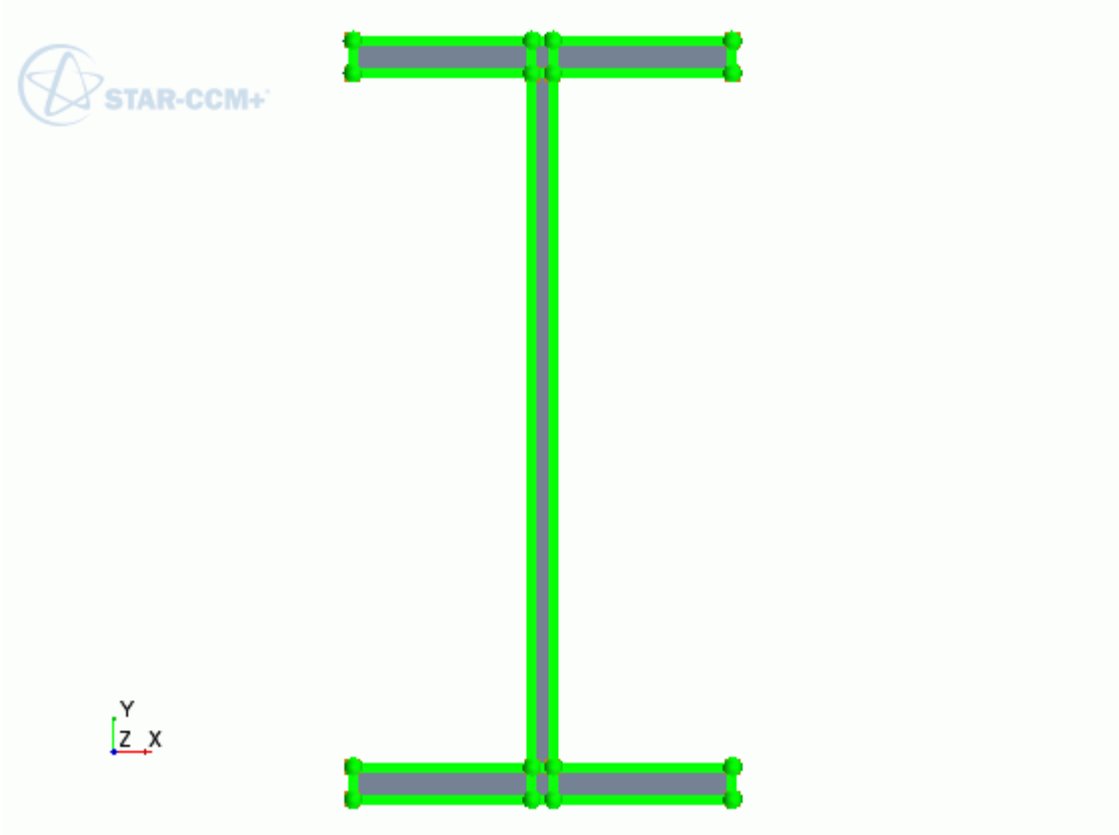
The Directed Meshing workflow requires you to manually create a quadrilateral patch mesh onto selected source surfaces. STAR-CCM+ then sweeps the surface mesh through the part, from the source surfaces to selected target surfaces, creating a specified number of layers.

By default, hexahedral elements are generated with 8 nodes (hex8), but you can add mid-side nodes to use hexahedral elements with 20 nodes (hex20).

1. Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Directed Mesh**.
2. In the *Create Directed Mesh Operation* dialog, select **Beam** from the *Parts* list then click **OK**.
3. Right-click the **Operations** > **Directed Mesh** node and select **Edit...**
4. In the *Directed Mesh* panel, set the following properties to specify the source and target surfaces for the directed mesh operation:

Node	Property	Setting
Source Surfaces	*Source Surfaces*	**wall side - lower flange**
		wall side - upper flange
		wall side - web
Target Surfaces	*Target Surfaces*	**free side - lower flange**
		free side - upper flange
		free side - web

To create a patch mesh on the beam surface at the wall, you are going to define the following patch topology:

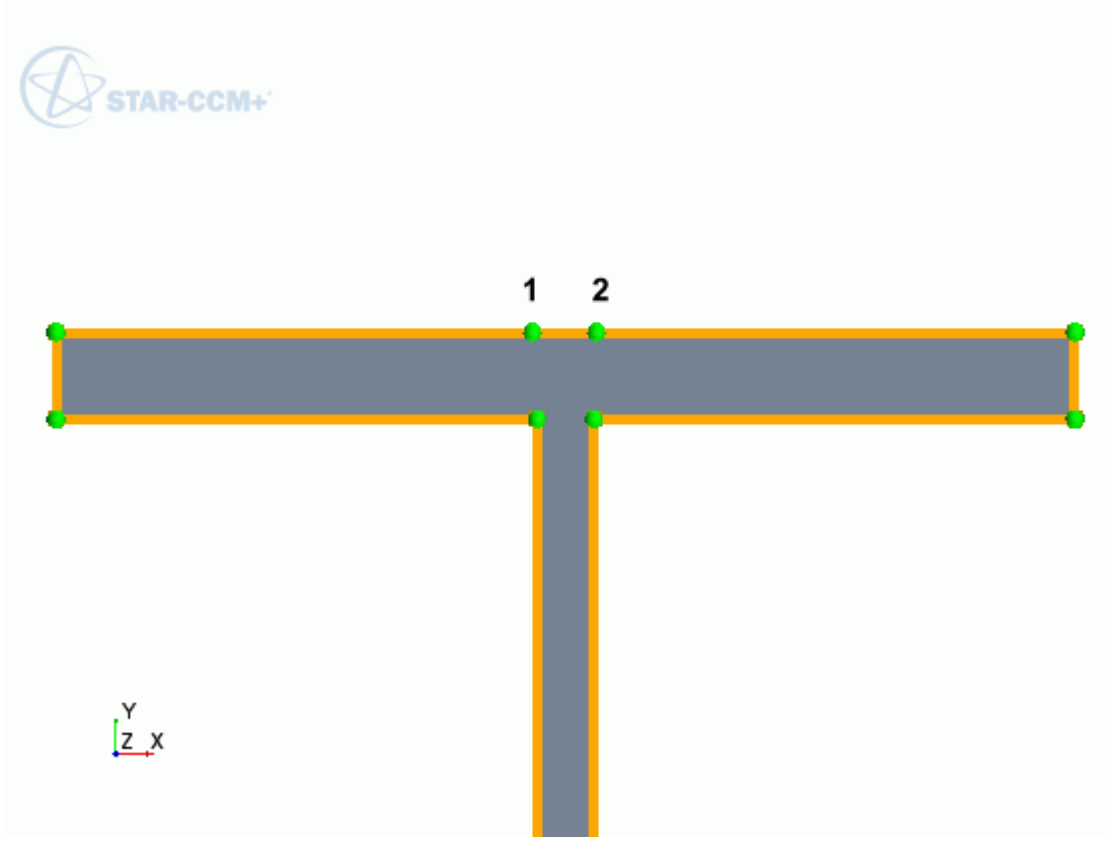


Start by adding a new source mesh:

5. In the *Directed Mesh* panel, right-click **Source Meshes** and select **New Source Mesh** > **Patch Mesh**.
6. In the *Part Collection For This Source Mesh* dialog, select **Beam** from the *Connected Parts* list then click **OK**.
7. Click anywhere in the *Directed Mesh 1* scene and press the <T> key to adjust the view.
8. In the in-place *Edit* dialog, click (**Auto-populate feature edges with patch curves**).

Split the surface of each flange into three patches:

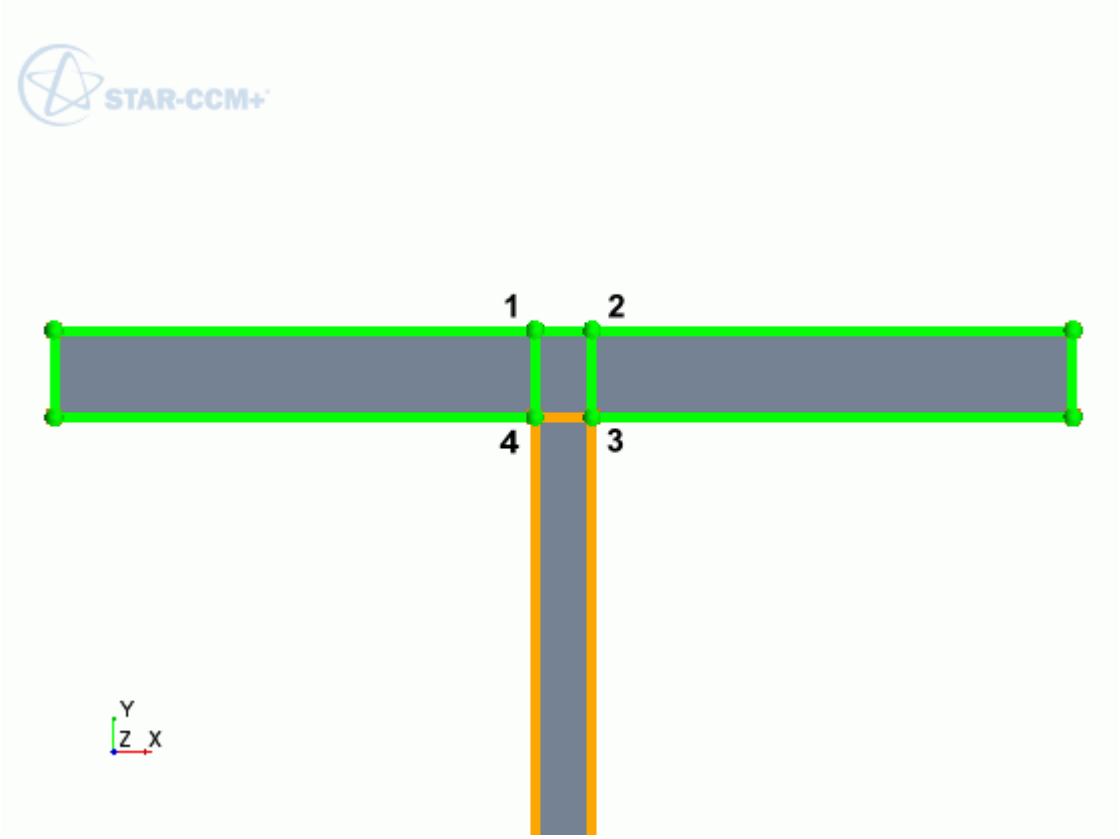
9. In the *Directed Mesh 1* scene, zoom in to focus on the upper flange.
10. Click (**Split the selected patch curve**) and maintain the split patch curve option, **At Picked Location**.
11. Click the external feature edge of the flange in two different locations, as shown below:



12. Press <Esc> to exit the split patch curve mode.

Adjust the coordinates of the new vertices as follows:

13. Click the vertex that is marked as **1** in the image above.
14. In the *Patch Vertex Location Options* group box, set the *X* coordinate to `-0.0028` and click **Apply**.
15. In a similar way, set the *X* coordinate of vertex **2** to `0.0028`.
16. Click (**Create Patch Curve**) then click the vertices 2, 3, 4 and 1 in order, as shown below:

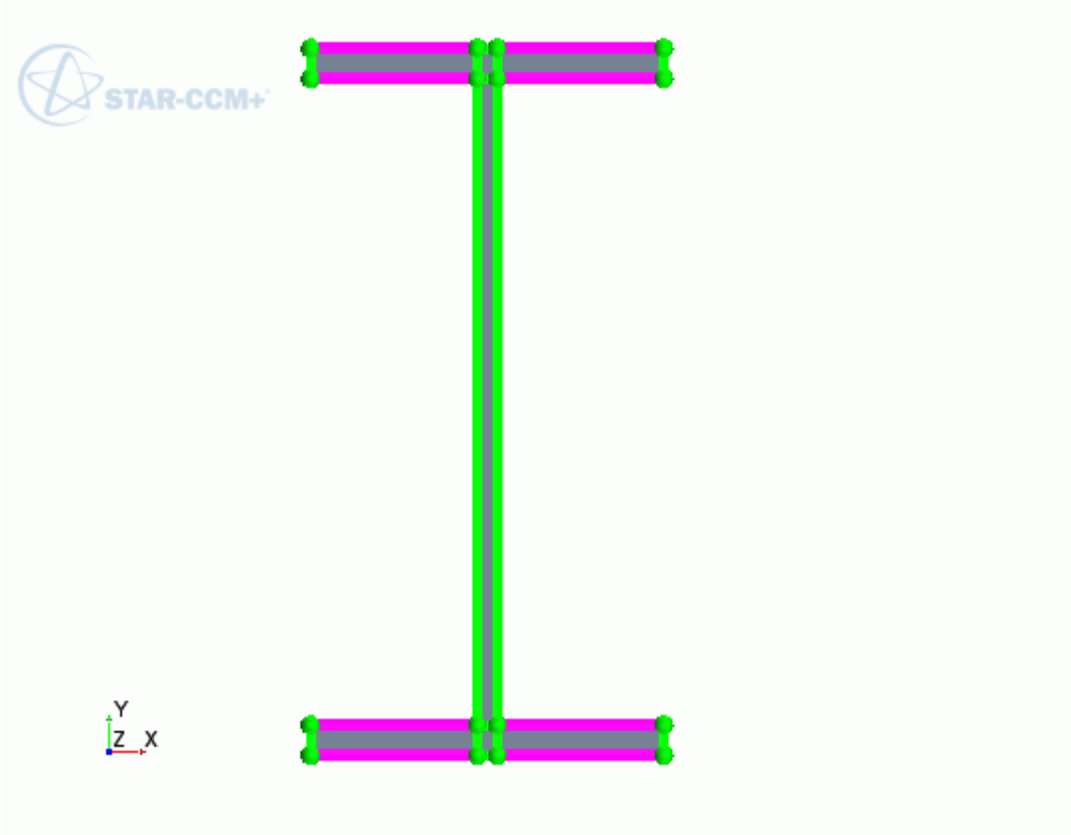


17. Following the same procedure, split the lower flange surface into three patches.

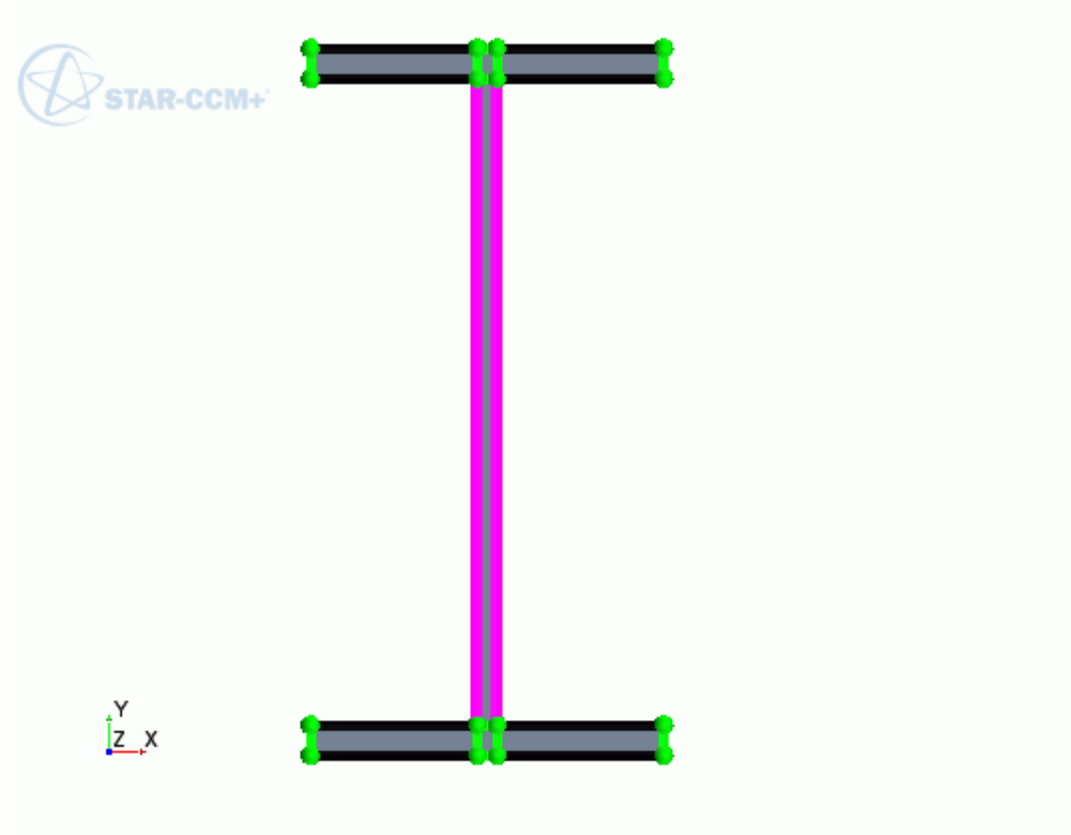
    The patch topology is now complete and looks as shown at the beginning of this section.

Complete the patch mesh by specifying the number of divisions within the patch curves:

18. In the *Edit* in-place dialog, switch the *Mode* to **Patch Mesh**.
19. In the *Directed Mesh 1* scene, multi-select the following patch curves:

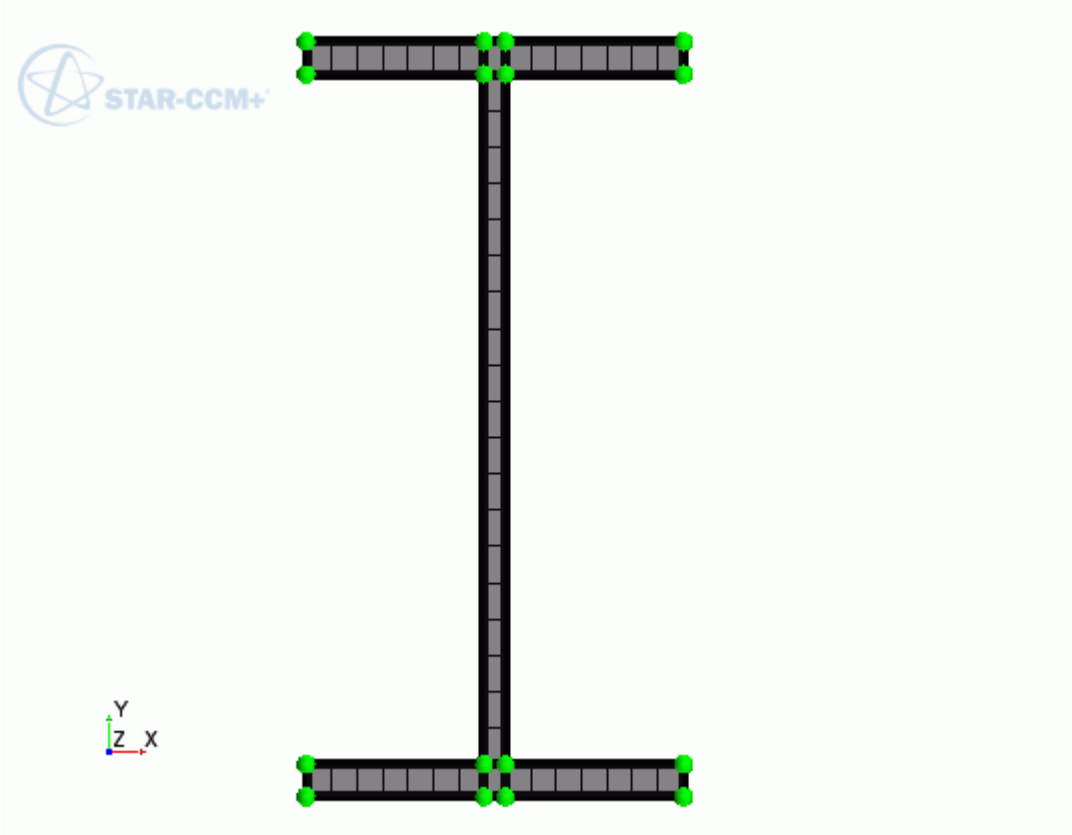


20. Set *Number of Divisions* to 7 then click **Apply**.
21. Multi-select the following patch curves:



22. Set *Number of Divisions* to `19` then click **Apply**.
23. For all the remaining patch curves, set *Number of Divisions* to `1`.

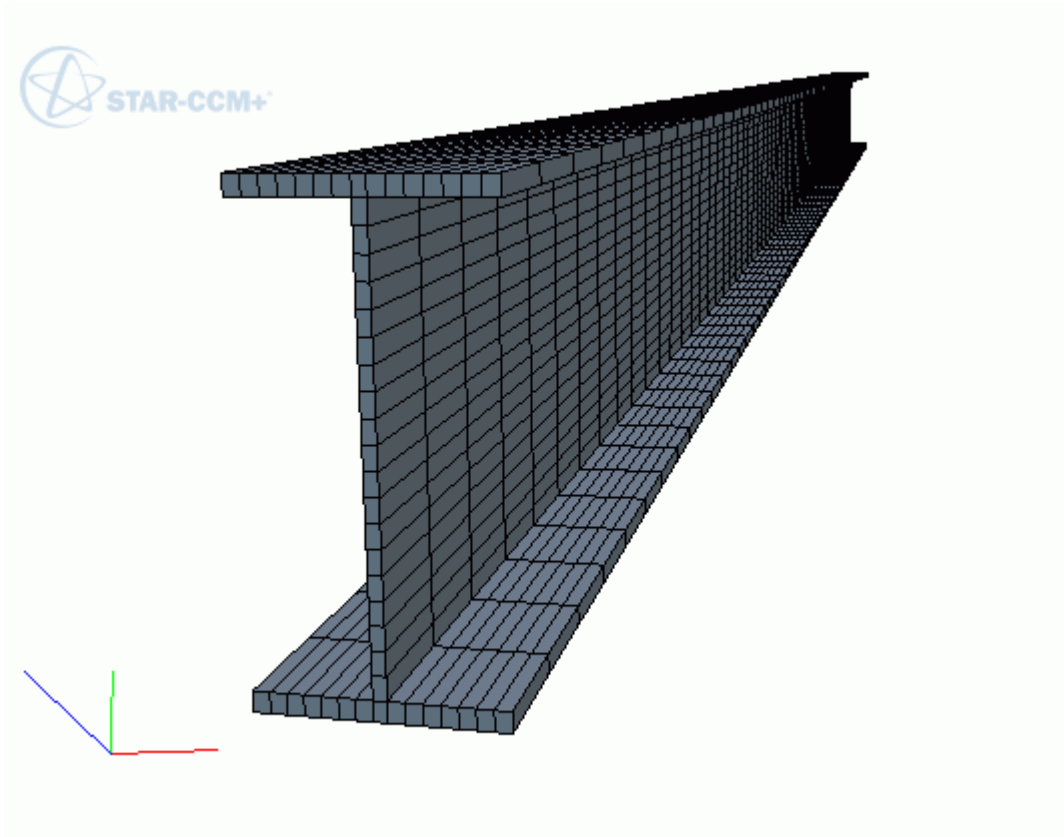
    The final patch mesh looks as shown below:



You are now ready to generate the volume hexahedral mesh.

24. Close the *Edit* in-place dialog.
25. In the *Directed Mesh* panel, right-click the **Mesh Distributions** node and select **New Volume Distribution**.
26. In the *Parts for creating a new mesh distribution* dialog, select **Beam** from the *Connected Parts* list then click **OK**.
27. Select the **Mesh Distributions** > **Volume Distribution** > **Default Controls** > **Number of Layers** node and set *Number of Layers* to `70`.
28. Click **Close Directed Mesh**.
29. In the simulation tree, right-click the **Geometry** > **Operations** > **Directed Mesh** node and select **Execute**.

    When the mesh generation is complete, you can visualize the mesh with a method of your choice, such as creating a mesh scene.



To add mid-side nodes:

30. Expand the **Regions** > **Beam** > **Physics Conditions** node.
31. Select the **Mid-side Vertex Option** and set *Mid-side Vertex* to **Add - Linear Interpolation**.

    The mid-side nodes are added when you run the simulation.
32. Save the simulation.

Preparing Analysis Objects

Set up scenes and reports to monitor the solution.

Set up a scalar scene to visualize the axial component of the stress tensor, σ_{zz}, throughout the beam:

1. Create a scalar scene and rename it to `Stress`.
2. Edit the **Scenes** > **Stress** > **Displayers** node and set the following properties:

Node	Property	Setting
Scalar 1	*Contour Style*	**Smooth Filled**
└ **Parts**	*Parts*	**Parts** > **Beam**
└ **Scalar Field**	*Function*	**Stress Tensor** > **Laboratory** > **k,k**

Node	Property	Setting
	Units	**MPa**

Create a scalar scene for the beam displacement, using a warp surface to visualize the deformed geometry. A warp surface is a derived part where the input surface point coordinates are moved along a selected vector multiplied by a scale factor.

3. Create a scalar scene and rename it to `Displacement`.
4. Right-click the derived parts node and select **New Part** > **Warp** > **Vector Warp...**
5. In the *Edit* in-place dialog, set the following properties:

Property	Setting
Input Parts	**Parts** > **Beam**
Vector	**Displacement**
Auto Scale	**Activated**
Display	**No Displayer**

6. Click **Create** then **Close**.

In the *Displacement* scalar scene, use a new geometry displayer to visualize the original geometry, and the existing scalar displayer to visualize the warp surface:

7. Right-click the **Scenes** > **Displacement** > **Displayers** node and select **New Displayer** > **Geometry**
8. Edit the **Displayers** node and set the following properties:

Node	Property	Setting
Geometry 1	*Opacity*	`0.2`
	Surface	**Activated**
└ **Parts**	*Parts*	**Parts** > **Beam**
Scalar 1	*Contour Style*	**Smooth Filled**
└ **Parts**	*Parts*	**Derived Parts** > **Vector Warp**
└ **Scalar Field**	*Function*	**Displacement** > **Laboratory** > **j**
	Units	**mm**

As Euler-Bernoulli beam theory predicts that the stress component σ_{zz} is maximum on the flanges at the wall, set up a report to evaluate the stress at a point located on the upper flange at the wall. Position the point probe sufficiently far from the area that was constrained in all degrees of freedom, as results in this area may not be accurate.

9. In the *Graphics* window, click the **Stress** scene tab.
10. Right-click the **Derived Parts** node and select **New Part** > **Probe** > **Point...**
11. In the *Create Point Probe* in-place dialog:
    a) Click **Select...** in the *Input Parts* box.

b) In the *Select Objects* dialog, right-click on a blank area and select **Deselect All**, then select the **Parts** > **Beam** > **Surfaces** node and click **OK**.

c) Set the point coordinates as follows:

Coordinate	Value
X	`50 mm`
Y	`100 mm`
Z	`0 mm`

d) Set *Display* to **No Displayer**.

e) Click **Create** then **Close**.

12. Rename the **Derived Parts** > **Point** node to `Corner-Point`.
13. Right-click the **Reports** node and select **New Report** > **Maximum**.
14. Rename the **Maximum 1** report to `Stress [k,k]` and set its properties as follows:

Property	Setting
Field Function	**Stress Tensor [k,k]**
Parts	**Derived Parts** > **Corner-Point**
Smooth Values	**Activated**

You can also set up a report to evaluate the maximum σ_{zz} where the beam is constrained in all degrees of freedom, to visualize how the stress computed in this area deviates from the analytical solution.

15. Copy the **Derived Parts** > **Corner-Point** node and paste it onto the **Derived Parts** node.
16. Rename the **Copy of Corner-Point** node to `Mid-Point`.
17. In the *Properties* window, set *Point* to `[0, 0.1, 0] m`.
18. Copy the **Reports** > **Stress [k,k]** node and paste it onto the **Reports** node.
19. Rename the **Copy of Stress [k,k]** node to **Stress [k,k] near Fixed Constraint**.
20. In the *Properties* window, set *Parts* to use **Derived Parts** > **Mid-Point** instead of **Derived Parts** > **Corner-Point**.
21. Save the simulation.

Running the Simulation

Specify appropriate running settings for this simulation. As this is a linear problem, you are likely to obtain a good solution within the first iteration. For a linear problem, it is good practice to make sure that the residuals drop at least 6 orders of magnitude within the second iteration.

1. Expand the **Stopping Criteria** node.
2. Select the **Maximum Steps** node and set *Maximum Steps* to `2`.
3. Select the **Force Criterion** > **Minimum Limit** node and set *Minimum Value* to `1.0E-6`.
4. Click (**Run**).

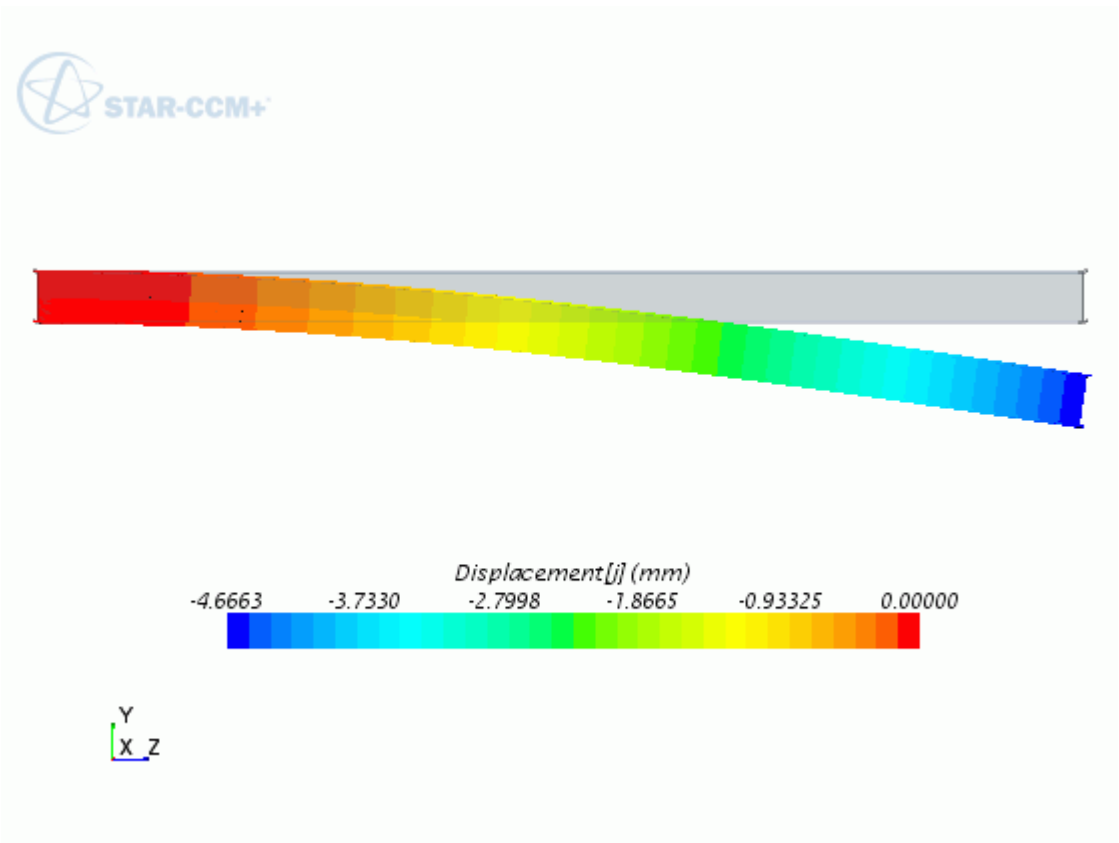
5. Save the simulation.

Visualizing the Solution

Visualize the solution for the displacement and stress.

1. Activate the *Displacement* scene in the *Graphics* window.

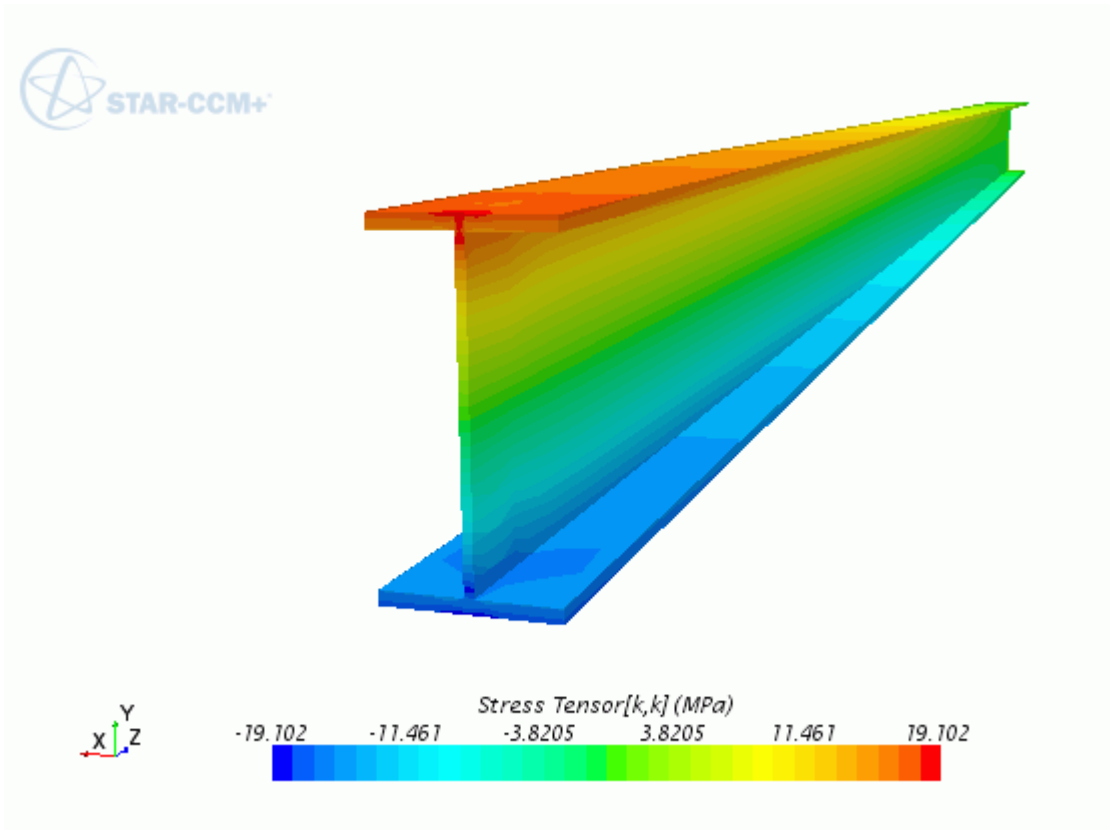
2. Click (**Save-Restore-Select Views**) and select **Views** > **-X** > **Up +Y**.



The computed maximum displacement is about 4.67 mm. Bernoulli-Euler beam theory predicts a displacement of about 4.63 mm, so the numerical solution differs from the theoretical result by less than 1%.

3. Activate the *Stress* scene in the *Graphics* window.

4. Click (**Save-Restore-Select Views**) and select **Restore View** > **View 1**.



As expected, the *Stress* scene shows high stress concentrations at the extremities of the web, that was constrained in all degrees of freedom. Examine the stress at the top flange corner, which is sufficiently far from the constrained areas:

5. Right-click the **Reports** > **Stress [k,k]** node and select **Run Report**.

   The report is printed to the output window. The computed stress at the top flange corner is about 17.4 MPa, which is in good agreement with the value of 17.3 MPa, given by Euler-Bernoulli beam theory.

To visualize how the stress computed near the fixed constraints deviates from Euler-Bernoulli beam theory:

6. Right-click the **Reports** > **Stress [k,k] near Fixed Constraint** node and select **Run Report**.

   The report is printed to the output window. The computed stress at the upper extremity of the web is about 18.61 MPa, which is higher than the result given by Euler-Bernoulli beam theory.

The results of this analysis are summarized in the table below. The first row shows the analytical solution from Euler-Bernoulli beam theory. The second row reports the numerical solution obtained in this tutorial. For comparison, the third row shows the numerical solution that you obtain if you apply a fixed constraint to both the flanges and the web.

Solution Method	Displacement [j]	σ_{zz} at flange center	σ_{zz} at flange corner
Euler-Bernoulli beam theory	- 4.63 mm	17.34 MPa	17.34 MPa
STAR-CCM+ analysis with:	- 4.67 mm	18.61 MPa	17.42 MPa

Solution Method	Displacement [j]	σ_{zz} at flange center	σ_{zz} at flange corner
• fixed constraint on web • normal displacement constraint on flanges			
STAR-CCM+ analysis with: • fixed constraint on web and flanges	-4.66 mm	20.67 MPa	24.08 MPa

Summary

This tutorial demonstrated the workflow for linear FE (Finite Element) structural analyses in STAR-CCM+.

This tutorial has presented the following procedures:

- Applying loads and constraints using STAR-CCM+ segments.
- Meshing with hexahedral elements using the Directed Mesher.
- Adding mid-side nodes to the elements.
- Setting up scenes and reports to visualize the structure displacement and the stress tensor components.
- Specifying stopping criteria appropriate to a linear analysis and running the simulation.

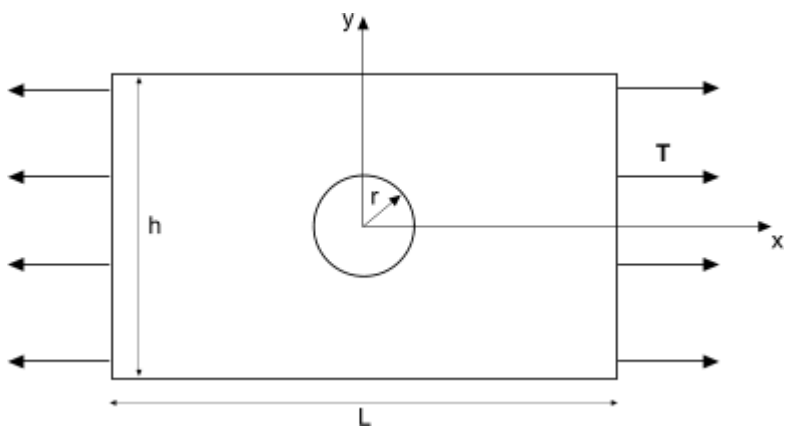
Plane Stress: Plate with Circular Hole in a Tensile Field

This tutorial demonstrates how to set up a linear elastic FE (Finite Element) stress analysis in STAR-CCM+.

The case under study is a rectangular plate under axial loads. A circular hole at the centre of the plate causes stress concentrations. The plate is in a plane stress state, so the stresses perpendicular to the plate are zero.

In this tutorial, you compute the stresses and the stress concentration factor at the hole and compare them with an analytical solution from Roark's [810].

The mathematical model is schematized below.



Length	L = 1.20 m
Height	h = 0.7 m
Thickness	t = 0.01 m
Hole Radius	r = 0.15 m
Axial Load	T = 1000 MPa
Material	Al
Stress condition	Linear elastic

As this problem has double symmetry, it is sufficient to model a quarter of the plate in STAR-CCM+.

Contents:

Prerequisites
Importing the Geometry
Defining the Simulation Topology
Selecting the Physics Models
Applying Loads
Generating the Mesh
Visualizing the Element Type

Preparing Analysis Objects
Running the Simulation
Visualizing the Solution
Summary
Bibliography

Prerequisites

The instructions in the Plane Stress - Plate with Circular Hole in a Tensile Field tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

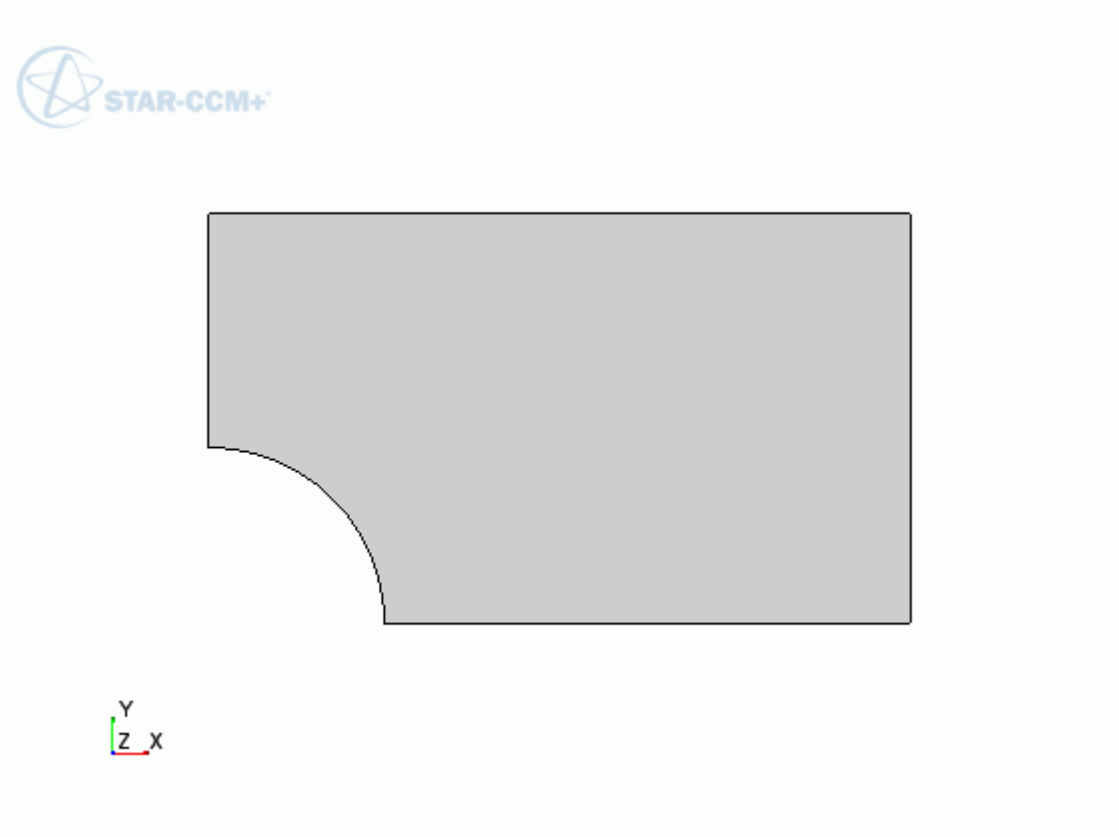
Techniques	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Importing the Geometry

Launch STAR-CCM+ and import the 3D-CAD plate geometry that is provided for this tutorial.

1. Launch STAR-CCM+.
2. Start a new simulation.
3. Select **File** > **Import** > **Import Surface Mesh...**
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/solidStress/data`.
5. Select `plateQuadrant.x_b`, then click **Open**.
6. In the *Import Surface Options* dialog, make sure that *Import Mode* is set to **Create New Part**, then click **OK**.

   The import process takes a few seconds, and STAR-CCM+ provides feedback on the process in the **Output** window. The imported geometry appears in a geometry scene in the **Graphics** window.



7. Save the simulation as `plate_planeStress.sim`.

Defining the Simulation Topology

Assign the part to a region and define the symmetry of the problem.

Before you generate a mesh for the imported part, the part must be assigned to a STAR-CCM+ region, and the part surfaces to region boundaries. In general, it is good practice to keep the number of boundaries to a minimum, and to have a single boundary definition for part surfaces that are subject to a common condition.

In this tutorial, you are going to apply a symmetry condition and a loading condition. Symmetry conditions are set on boundaries, whereas loading conditions are applied directly on part surfaces using segments.

As you are modeling only a quarter of the plate, you require two symmetry planes. Also, to fully constrain the part and to achieve a plane stress condition, you set a constraint on the front surface of the plate, to prevent movement in the normal direction. Since a zero normal displacement constraint is equivalent to a symmetry condition, you can fully constrain the part by specifying an additional symmetry plane.

Create a boundary for the part surfaces representing the symmetry planes, and a separate boundary for all the remaining part surfaces. The loading condition does not require an additional boundary definition, as it is set directly on the part surface.

Assign the plate part to a region:

1. Right-click the **Geometry** > **Parts** > **Plate** node and select **Set New Region...**.
2. In the *Input* dialog, set *New Region Name* to `Plate` and click **OK**.

3. Right-click the **Regions** > **Plate** > **Boundaries** > **Default** node and select **Delete**.

Create a boundary for the part surfaces where you apply symmetry conditions:

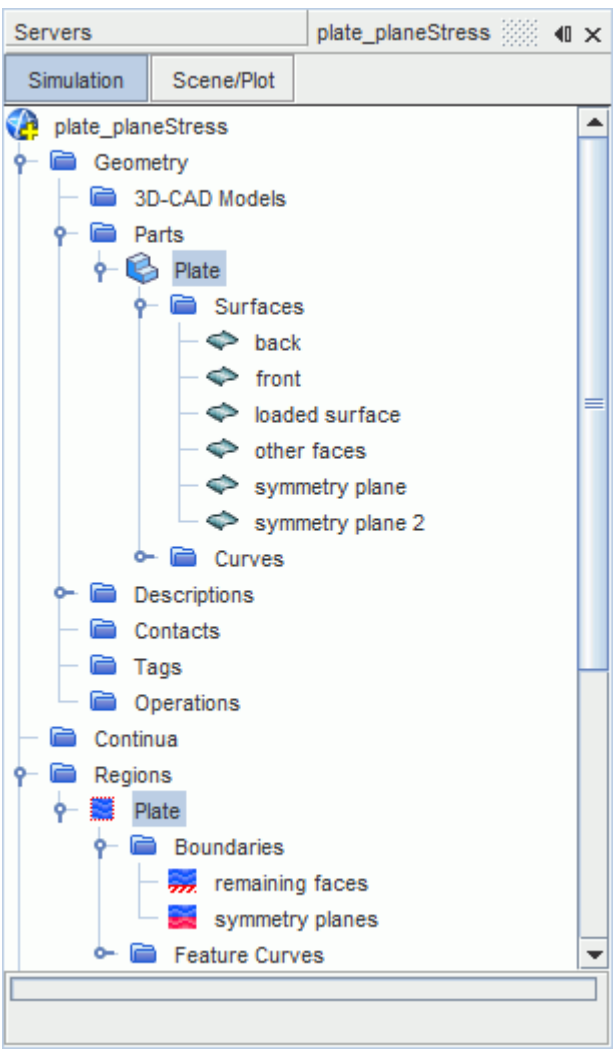
4. Expand the **Geometry** > **Parts** > **Plate** > **Surfaces** node.
5. Multi-select the **symmetry plane**, **symmetry plane 2**, and **front** nodes.
6. Right-click one of the selected nodes and select **Set New Boundary...**.
7. In the *Input* dialog, set *New Boundary Name* to `symmetry planes` and click **OK**.

Create a boundary for the remaining part surfaces:

8. Multi-select the **back**, **loaded surface** and **other faces** part surface nodes.
9. Right-click one of the selected nodes and select **Set Boundary** > **New...**.
10. In the *Input* dialog, set *New Boundary Name* to `remaining faces` and click **OK**.

Set a symmetry condition on the **symmetry planes** boundary:

11. Select the **Regions** > **Plate** > **Boundaries** > **symmetry planes** node and set *Type* to **Symmetry Plane**.
    The part and the region appear as shown below.



12. Save the simulation.

Selecting the Physics Models

Select physics models to set up a static linear analysis.

The physics models define the primary variables of the simulation and which mathematical formulation is used to generate the solution. In this simulation, the structure displacement and the internal stress distribution are computed using the **Sparse Direct Solver**, which is activated when you select the **Solid Stress** model.

1. Create a physics continuum and rename it to `Plate Physics`.
2. For the physics continuum, **Plate Physics**, select the following models:

Group box	Model
Space	**Three Dimensional**
Time	**Steady**
Material	**Solid**
Optional Models	**Solid Stress**
	Material Law Models (selected automatically)

The default solid material, **Al**, is appropriate to this case.

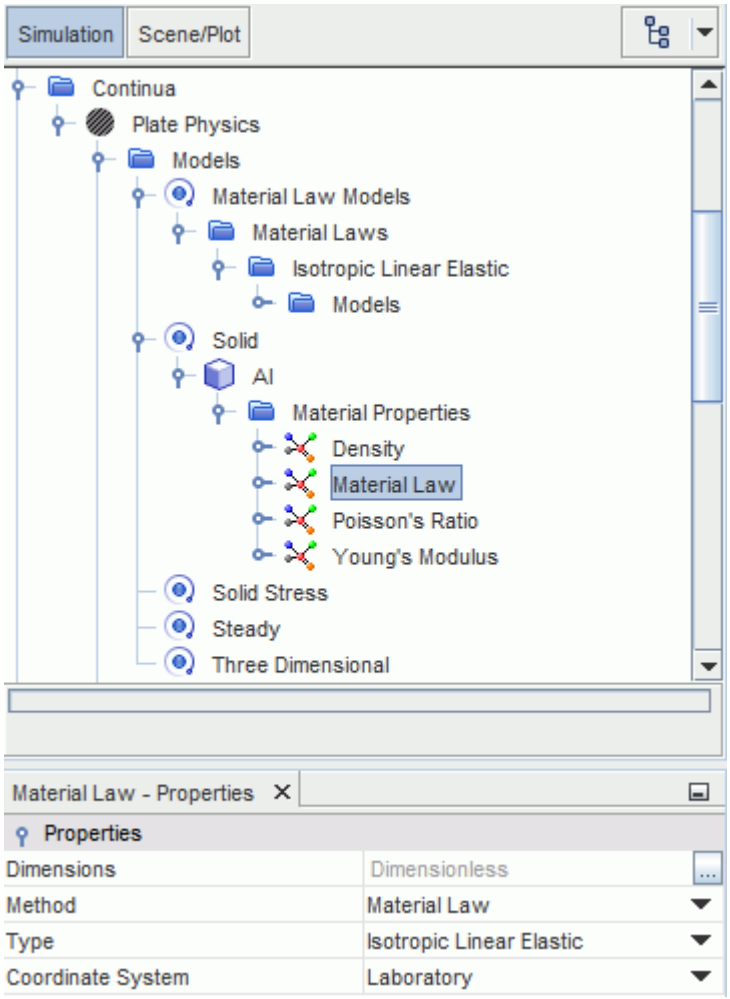
Define the material law that approximates the behavior of the material. In this tutorial, you model the material as isotropic linear elastic:

3. Expand the **Continua** > **Plate Physics** > **Models** > **Material Law Models** node.
4. Right-click the **Material Laws** node and select **New**.
5. Rename the **Material Laws** > **Material Law 1** node to `Isotropic Linear Elastic`.
6. For the **Isotropic Linear Elastic** law node, select the following models:

Group box	Model
Material Models	**Linear Elasticity**
Linear Elastic Material Models	**Isotropic Linear Elasticity**

Assign the isotropic linear elastic material law to the solid material:

7. Select the **Plate Physics** > **Models** > **Solid** > **Al** > **Material Properties** > **Material Law** node and set *Type* to **Isotropic Linear Elastic**.



Applying Loads

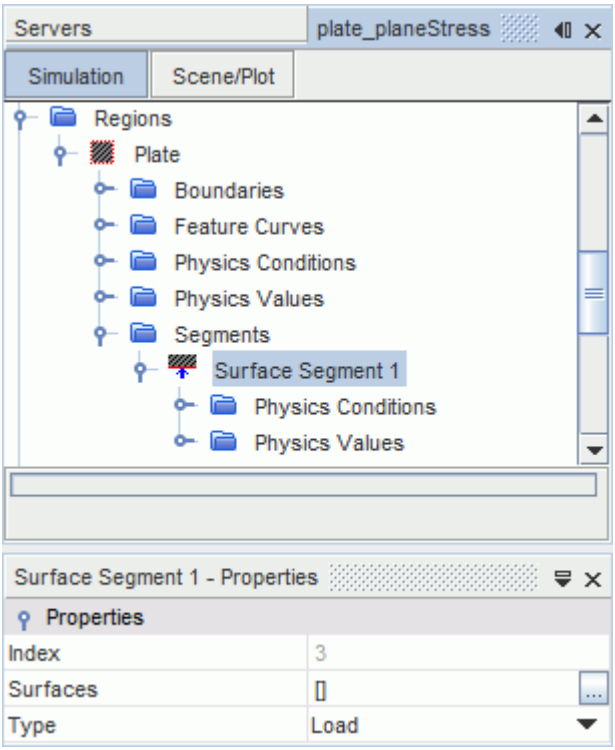
Using a STAR-CCM+ segment, apply a traction load to the side surface of the plate.

Start by adding a segment:

1. Right-click the **Regions** > **Plate** node and select **Create Segment**.

   STAR-CCM+ adds a node, **Segments** > **Surface Segment 1**, under the **Plate** region node.

   All the segments that you create for a region are added within the **Segments** node for that region.



The segment *Surfaces* property allows you to specify a set of part surfaces. The *Type* property allows you to specify the type of condition (load or constraint) that is applied to the selected part surfaces.

Set up the segment to apply a traction load on the side surface:

2. Rename the **Surface Segment 1** node to `Traction Load`.
3. Select the **Segments** > **Traction Load** node and set the properties as follows:

Property	Setting
Surfaces	**loaded surface**
Type	**Load** (default)

Specify a constant traction of 1000 MPa in the x direction:

4. Select the **Tools** > **Units** > **MPa** node and activate *Preferred*.
5. Edit the **Segments** > **Traction Load** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Solid Stress Loads**	*Method*	**Traction**
Physics Values		
└ **Traction**	*Value*	`[1000.0, 0.0, 0.0] MPa`

6. Save the simulation.

Generating the Mesh

In this tutorial, you mesh the imported part with 3D solid wedge elements using the **Tetrahedral Mesher** and the **Thin Mesher**.

Add an automated mesh operation and select the relevant meshers:

1. Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
2. In the *Create Automated Mesh Operation* dialog, select **Plate** from the *Parts* list, then select the following meshers:

Group	Mesher
Surface Meshers	**Surface Remesher**
Core Volume Meshers	**Tetrahedral Mesher**
Optional Volume Meshers	**Thin Mesher**

3. Click **OK**.

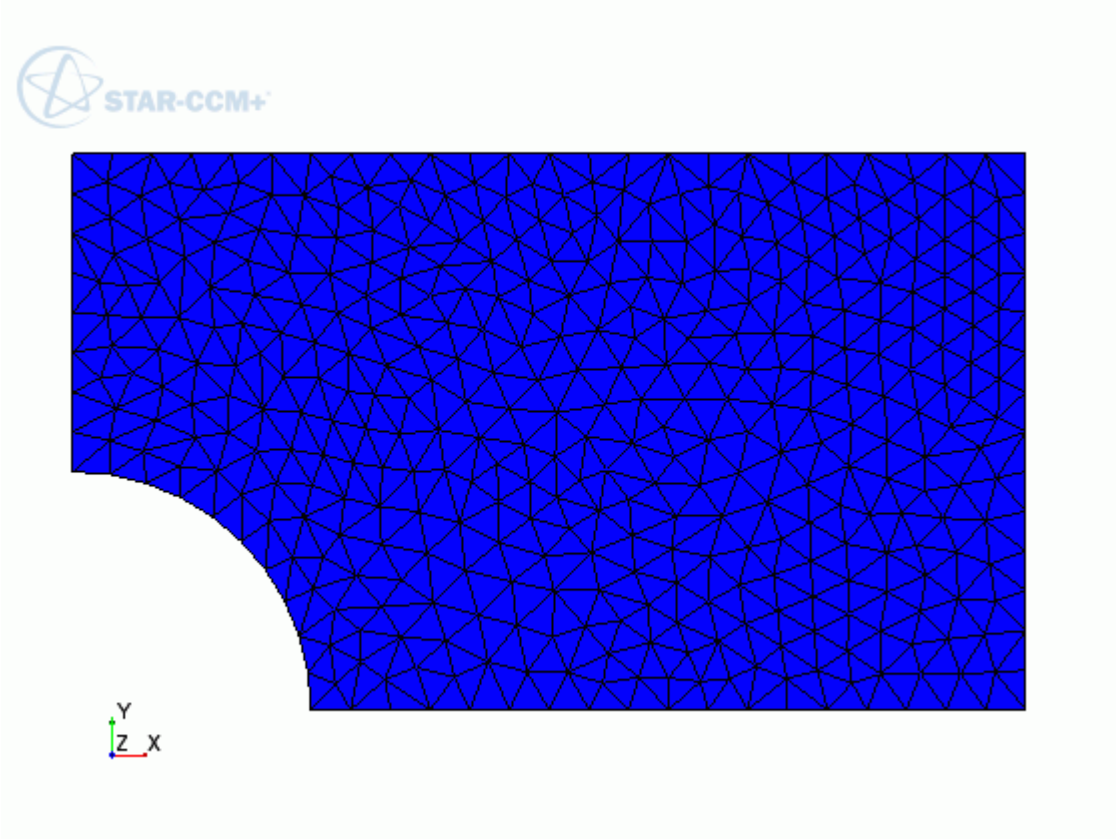
Specify reference values for the element surfaces and the number of thin layers that the **Thin Mesher** adds through the plate thickness:

4. Edit the **Operations** > **Automated Mesh** > **Default Controls** node and set the following properties:

Node	Property	Setting
Base Size	*Value*	`0.05 m`
Minimum Surface Size	*Percentage of Base*	`50`
Number of Thin Layers	*Number of Layers*	`1`

5. Execute the mesh operation.

   You can visualize the mesh with a method of your choice, such as creating a mesh scene.



Add mid-side vertices to the mesh:

6. Select the **Regions** > **Plate** > **Physics Conditions** > **Mid-side Vertex Option** and set *Mid-side Vertex* to **Add - Linear Interpolation**.

   STAR-CCM+ will generate the mid-side vertices when you run the simulation.

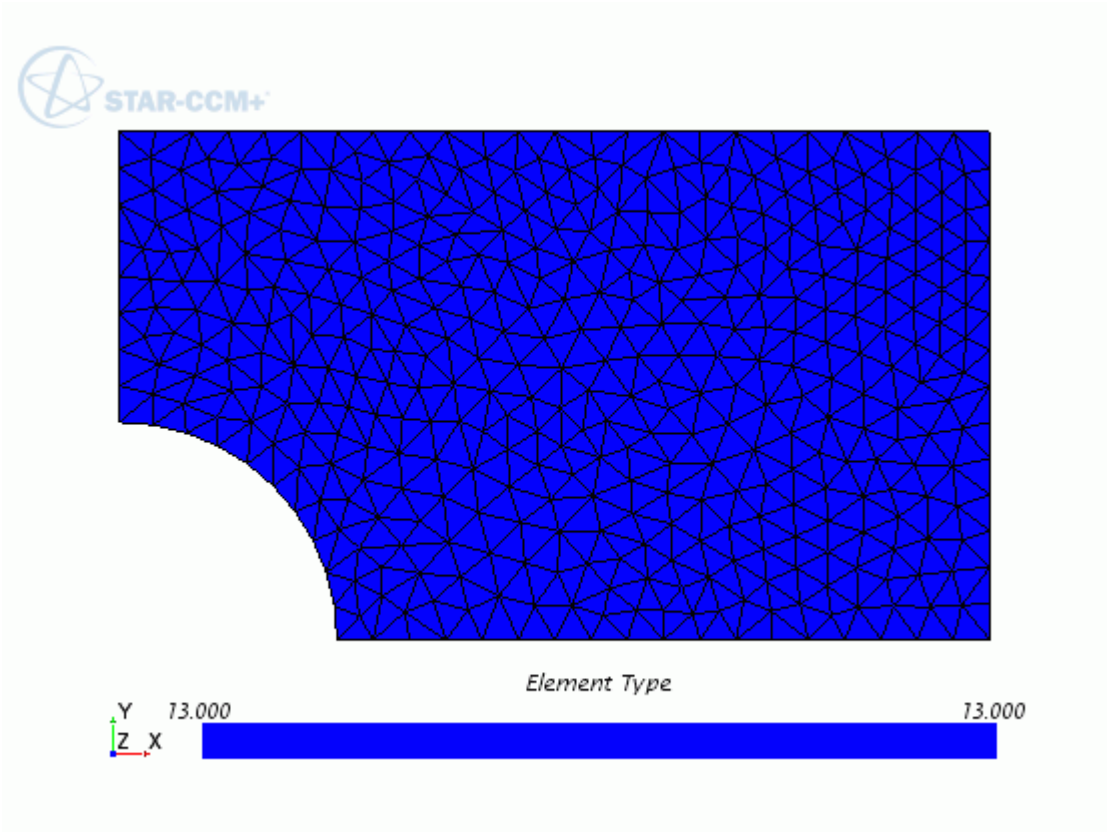
Visualizing the Element Type

Use the *Element Type* field function in STAR-CCM+ to display the element type in a scalar scene.

1. Create a scalar scene and rename it to `Element Type`.
2. Edit the **Element Type** > **Displayers** node and set the following properties:

Node	Property	Setting
Scalar 1	*Display Mesh*	**Activated**
└ **Parts**	*Parts*	**Parts** > **Plate**
└ **Scalar Field**	*Function*	**Element Type**

   The scene appears as shown below:



All the solid elements in the mesh are of the same type. The number 13 corresponds to wedge elements with 6 nodes.

When you run the simulation, STAR-CCM+ adds mid-side nodes to the mesh. As a consequence, the value that is displayed in the scene changes to 26, that corresponds to wedge elements with 15 nodes.

Preparing Analysis Objects

Set up scenes and reports to monitor the solution.

As this is a problem with cylindrical symmetry, it is useful to define a cylindrical coordinate system:

1. Expand the **Tools** > **Coordinate Systems** > **Laboratory** node.
2. Right-click the **Local Coordinate Systems** node and select **New** > **Cylindrical**.
3. In the *Edit* in-place dialog, click **Create** then **Close**.

Set up a scalar scene to visualize the hoop stress, or cylindrical stress, $\sigma_{\theta\theta}$:

4. Create a scalar scene and rename it to `Hoop Stress`.
5. Select the **Hoop Stress** node and set *Coordinate System* to **Laboratory** > **Cylindrical 1**.
6. Edit the **Hoop Stress** > **Displayers** node and set the following properties:

Node	Property	Setting
Scalar 1	*Contour Style*	**Smooth Filled**

Node	Property	Setting
└ **Parts**	*Parts*	**Parts** > **Plate**
└ **Scalar Field**	*Function*	**Stress Tensor** > **Laboratory** > **Cylindrical 1** > **theta, theta**

Create a report for the maximum value of σ_{xx}:

7. Right-click the **Reports** node and select **New Report** > **Maximum**.
8. Rename the report, **Maximum 1**, to `Stress tensor (ii)` and set the following properties:

Property	Setting
Scalar Field Function	**Stress Tensor** > **Laboratory** > **i,i**
Parts	**Parts**
Smooth Values	**Activated**

Also, set up a reaction report to display the applied force and the reaction force:

9. Right-click the **Reports** node and select **New Report** > **Sum Force**.
10. Select the **Sum Force 1** node and set *Parts* to **Plate**.

Create a scalar scene for the plate displacement along the x axis, using a warp surface to visualize the deformed geometry. A warp surface is a derived part where the input surface point coordinates are moved along a selected vector multiplied by a specified scale factor.

11. Create a scalar scene and rename it to `Displacement`.
12. Right-click the derived parts node and select **New Part** > **Warp** > **Vector Warp...**
13. In the *Edit* in-place dialog, set the following properties:

Property	Setting
Input Parts	**Parts** > **Plate**
Vector	**Displacement**
Auto Scale	**Activated**
Display	**No Displayer**

14. Click **Create** then **Close**.

In the *Displacement* scalar scene, use a new geometry displayer to visualize the original geometry, and the existing scalar displayer to visualize the warp surface:

15. Right-click the **Scenes** > **Displacement** > **Displayers** node and select **New Displayer** > **Geometry**
16. Edit the **Displayers** node and set the following properties:

Node	Property	Setting
Geometry 1	*Opacity*	`0.2`
	Surface	**Activated**

Node	Property	Setting
└ **Parts**	*Parts*	**Parts** > **Plate**
Scalar 1	*Contour Style*	**Smooth Filled**
└ **Parts**	*Parts*	**Derived Parts** > **Vector Warp**
└ **Scalar Field**	*Function*	**Displacement** > **Laboratory** > **i**
	Units	**mm**

Create a scalar scene to visualize the Von Mises stress.

17. Create a scalar scene and rename it to `Von Mises Stress`.
18. Edit the **Scenes** > **Von Mises Stress** > **Displayers** node and set the following properties:

Node	Property	Setting
Scalar 1	*Contour Style*	**Smooth Filled**
└ **Parts**	*Parts*	**Parts** > **Plate**
└ **Scalar Field**	*Function*	**Stress Von Mises**

19. Save the simulation.

Running the Simulation

Specify appropriate running settings for this simulation. As this is a linear problem solved with a direct solver, a solution is computed within the first iteration. For a linear problem, it is good practice to make sure that the residuals drop at least 6 orders of magnitude within the second iteration.

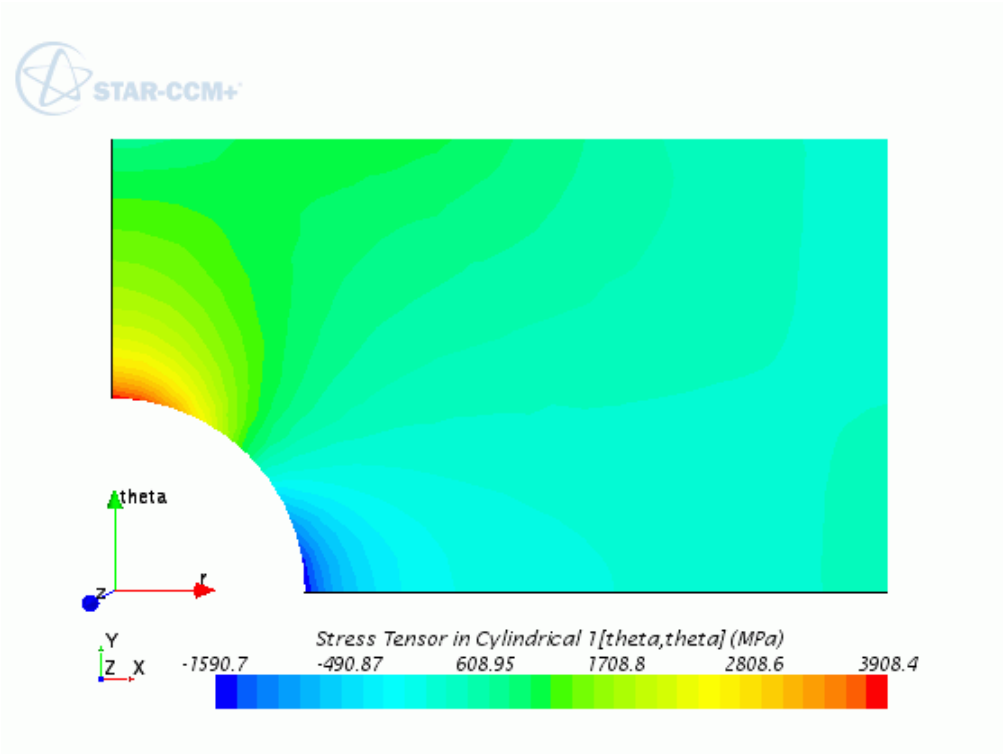
1. Expand the **Stopping Criteria** node.
2. Select the **Maximum Steps** node and set *Maximum Steps* to `2`.
3. Select the **Force Criterion** > **Minimum Limit** node and set *Minimum Value* to `1.0E-6`.
4. Click (**Run**).
5. Save the simulation.

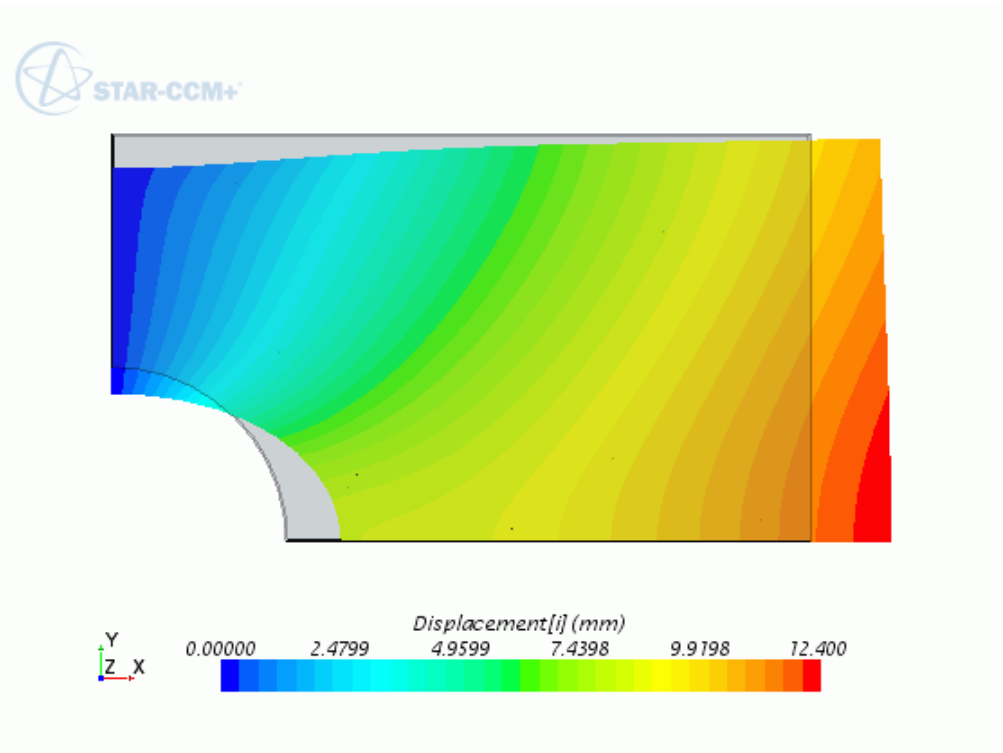
Visualizing the Solution

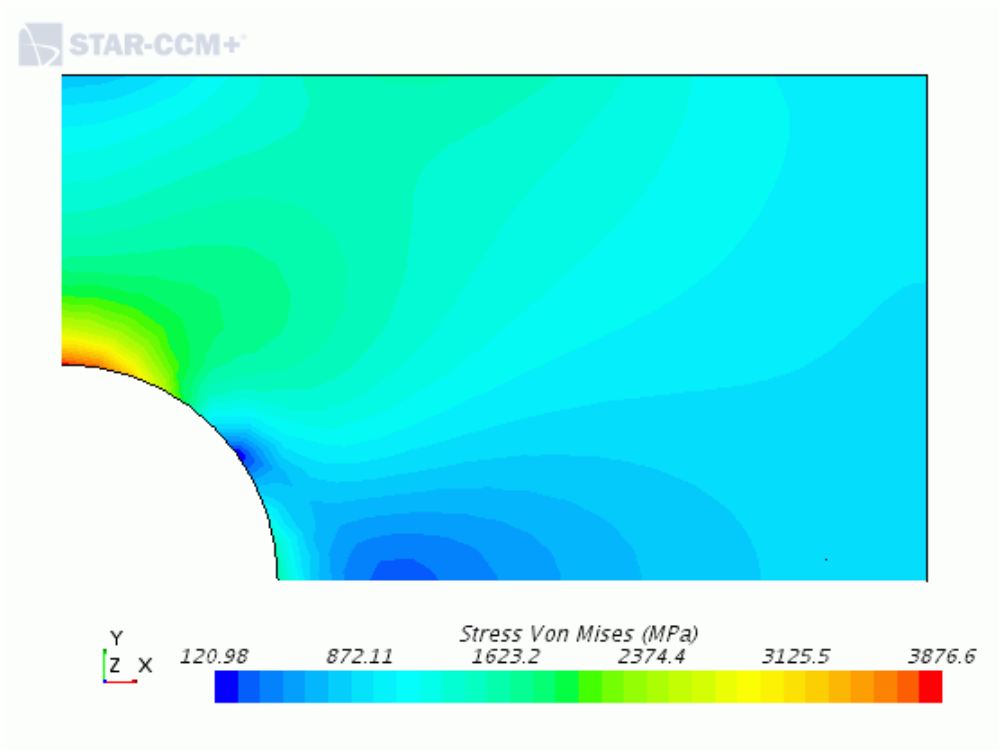
Visualize the solution and calculate the stress concentration factor from the computed stress.

1. Visualize the computed hoop stress, von Mises stress, and displacement using the scalar scenes that you created for this simulation.

   The scenes appear as shown below:


STAR-CCM+
theta
r
z
Y
Z X
Stress Tensor in Cylindrical 1[theta,theta] (MPa)
-1590.7
-490.87
608.95
1708.8
2808.6
3908.4


STAR-CCM+
Y
Z X
Displacement[i] (mm)
0.00000
2.4799
4.9599
7.4398
9.9198
12.400



To visualize the applied force and the reaction force:

2. Right-click the **Reports** > **Sum Force 1** node and select **Run Report**.
   The report is printed to the *Output* window. The resultant force is zero.

Run the report for the maximum value of σ_{xx}:

3. Right-click the **Reports** > **Stress Tensor (ii)** node and select **Run Report**.
   A value of $\sigma_{xx} \approx 3900 \quad \mathrm{MPa}$ is printed to the *Output* window.

Comparison with Analytical Solution from Roark's formulas

In Roark's formulas [810], the maximum stress σ_{xx} is related to the nominal stress σ_{nom} through the stress concentration factor k:

$$\begin{aligned} \sigma_{xx,\ \max} &= k\sigma_{\mathrm{nom}} \\ \sigma_{\mathrm{nom}} &= \sigma_i \frac{\mathrm{H}}{\mathrm{h}} \\ k &= 3.00 - 3.14\left(\frac{\mathrm{r}}{\mathrm{H}}\right) + 3.667\left(\frac{\mathrm{r}}{\mathrm{H}}\right)^2 - 1.527\left(\frac{\mathrm{r}}{\mathrm{H}}\right)^3 \end{aligned} \tag{4170}$$

where H is the height of the plate quarter, r is the hole radius, $\mathrm{h} = \mathrm{H} - \mathrm{r}$, and σ_i is the input stress, that is, the applied traction load.

The analytical solution therefore gives $\sigma_{xx,\ \max} = 3.86 \times 10^3 \quad \mathrm{MPa}$ and $\mathrm{k} = 2.21$.

The maximum σ_{xx} computed in STAR-CCM+ is $\sigma_{xx,\ \mathrm{report}} \approx 3.90 \times 10^3 \quad \mathrm{MPa}$, which gives a stress concentration factor of $k = \left(\sigma_{xx,\ \mathrm{report}}\right) / \sigma_{\mathrm{nom}} \approx 2.23$.

Note that the stress gradient within the elements at the top of the hole is very high, about 20 %. A better accuracy can be achieved with a finer mesh.

Summary

This tutorial demonstrated the workflow for linear FE (Finite Element) structural analyses in STAR-CCM+.

This tutorial has presented the following procedures:

- Apply symmetry conditions and constraining the part.
- Applying loads using STAR-CCM+ segments.
- Meshing with wedge elements using the Tetrahedral Mesher and the Thin Mesher.
- Adding mid-side nodes to the elements and visualize the element type.
- Setting up scenes to visualize the structure displacement, the hoop stress, and the Von Mises Stress.
- Specifying stopping criteria appropriate to a linear analysis and running the simulation.

Bibliography

[810] *Young, Warren C. "Roark's Formulas for Stress & Strain", 6th Edition, McGraw - Hill Inc, 1989, p.732*

Thermal Strain: Exhaust Manifold

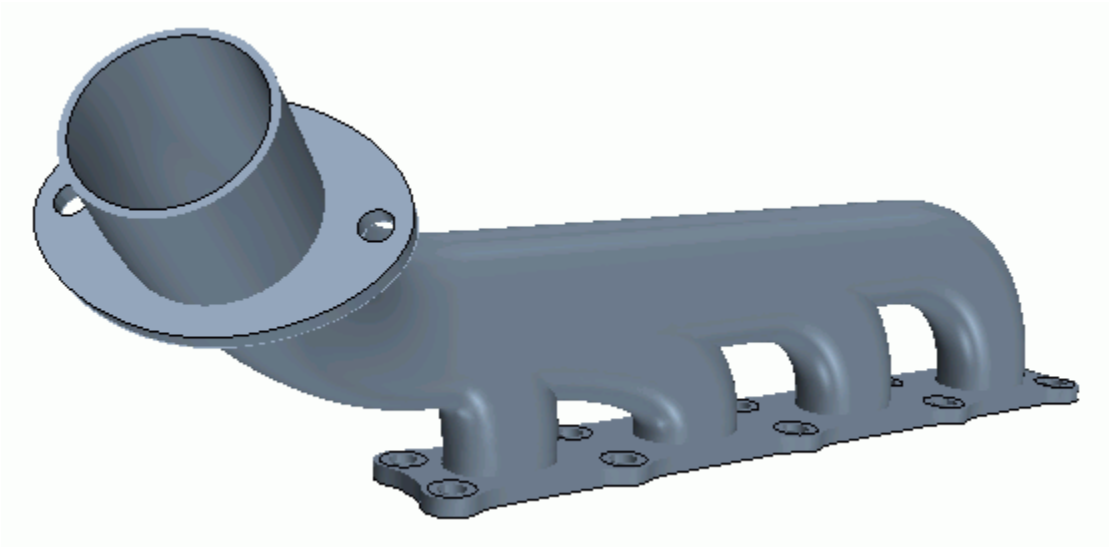
This tutorial demonstrates how to perform thermal stress analysis in STAR-CCM+.

In this simulation, you study the thermal expansion of an exhaust manifold in response to hot gas entering the inlets. The simulation incorporates both conjugate heat transfer (CHT) and stress analysis.

In STAR-CCM+, you can combine the advantages of the finite volume (FV) and the finite element (FE) methods together in the same simulation. In the first part of the tutorial, to compute the steady-state temperature distribution throughout the manifold, you run a finite volume conjugate heat transfer analysis. In the second part of the tutorial, you map the temperature data onto a finite element model of the manifold for stress analysis.

As you are seeking the static/steady-state solutions in the fluid and the solid, it is both convenient and accurate to initially solve for the fluid flow and the conjugate heat transfer, and then solve for the solid displacement while holding the thermal field fixed.

The simulation strategy, including the model geometry and assumptions, is summarized below.



Thermal Analysis

	Solid Domain	Fluid Domain
Geometry	**Inner diameter**: 0.04 m	Internal volume of the solid domain.
	Outer diameter: 0.042 m	
	Length: 0.4 m	
Assumptions and Models	• Material: Ductile Iron • Equation of State: Constant Density	• Material: Air • Equation of State: Ideal Gas • Flow Regime: Turbulent

	Solid Domain	Fluid Domain
Boundary Conditions	• Thermal specification for outer boundaries: Convection	• Temperature at inlets: 1000 K • Mass flow rate at inlets: 0.001 kg/s
Type of Analysis	Steady	Steady
Discretization and Solution Method	Finite Volume (FV)	Finite Volume (FV)
Mesh	Polyhedral Cells	Polyhedral Cells

Stress Analysis

	Solid Domain
Geometry	Same as FV Solid Domain
Assumptions and Models	• Material: Ductile Iron • Constitutive Law: Linear Elastic • Geometry: Linear
Boundary Conditions	• Constraints: Three point constraints at the manifold base • Thermal Loads: Temperature Loads are calculated during thermal analysis and then mapped onto the FE mesh
Type of Analysis	Static
Discretization and Solution Method	Finite Element (FE)
Mesh	Tet10 Elements

Contents:

Prerequisites
Loading the Initial Simulation
Running the Thermal Analysis
Preparing the Simulation for Stress Analysis
Specifying Physics Models and Material Properties
Generating the Mesh
Preparing Analysis Objects
Applying and Testing Constraints
Applying Thermal Loads
Solving for Solid Stress
Visualizing the Results
Summary

Prerequisites

The instructions in the Thermal Strain - Exhaust Manifold tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Techniques	Tutorial
The STAR-CCM+ workflow for stress analysis	Linear Stress Analysis: Cantilever I Beam
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+
Modeling Conjugate Heat Transfer (CHT)	Conjugate Heat Transfer: Heated Fin Introduction
Using the **Solid Stress** model	Linear Stress Analysis: Cantilever I Beam

Loading the Initial Simulation

For this tutorial, you are provided with a starting simulation that includes a predefined conjugate heat transfer analysis, with which you compute the steady-state temperature distribution in the fluid and the manifold.

To load the initial simulation:

1. Launch STAR-CCM+.
2. Select **File** > **Load Simulation...**
3. In the *Load Simulation* dialog, click **Browse...**
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/solidStress/data`.
5. Select `manifold_start.sim` and click **Open**.
6. In the *Load Simulation* dialog, click **OK**.

As the main focus of this tutorial is the application of thermal loads in FE stress analysis, the starting simulation file is already set up for FV conjugate heat transfer analysis. The starting simulation includes:

- geometry parts and corresponding FV regions, to represent the manifold and the fluid
- an automated mesh operation, to generate a polyhedral mesh for both parts
- a parts-based interface between the solid and fluid regions, for conjugate heat transfer
- physics models, thermal material properties, and boundary conditions for CHT analysis
- solver settings and stopping criteria
- scalar scenes to monitor the temperature throughout the manifold and the fluid

You can review the simulation settings by expanding the relevant nodes. Refer to the tutorial introduction for an overview of the thermal analysis, Thermal Strain: Exhaust Manifold.

7. Save the simulation to your working directory as `manifold_thermal.sim`.

Running the Thermal Analysis

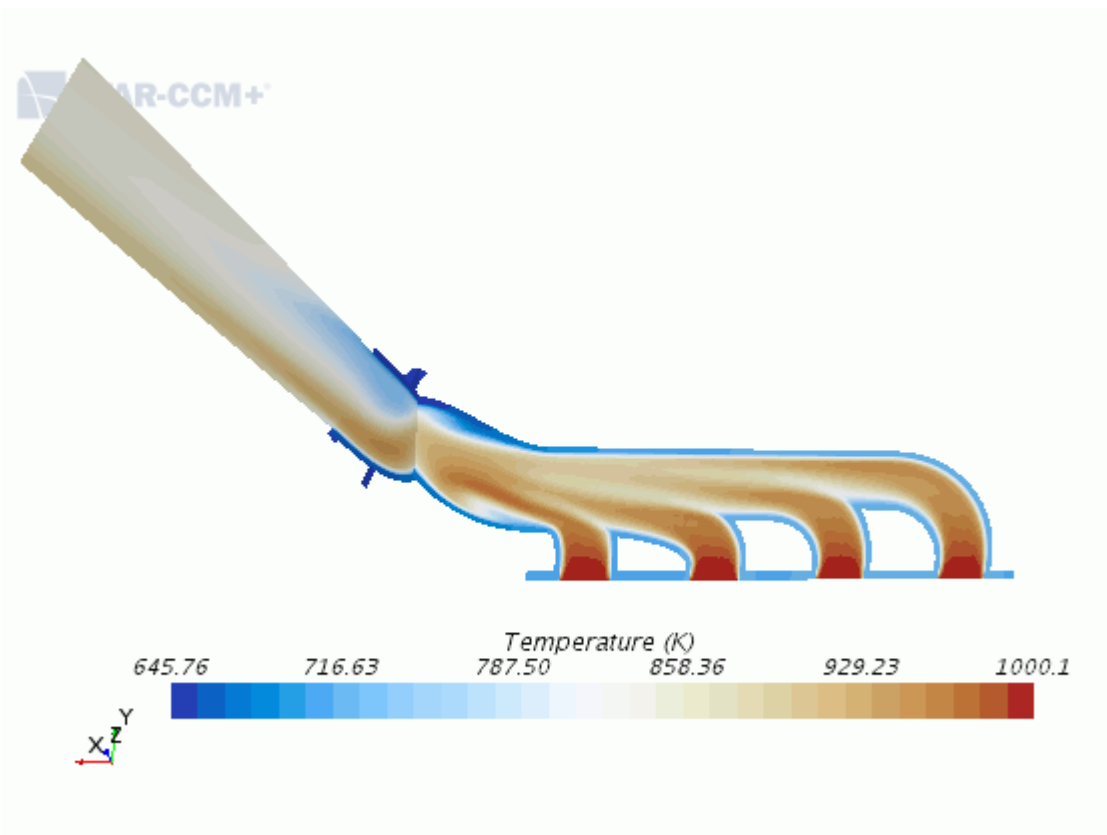
Run the predefined thermal analysis to obtain the steady-state temperature distribution. Later on, for thermal strain computation, you supply the computed temperature to the FE Stress solver.

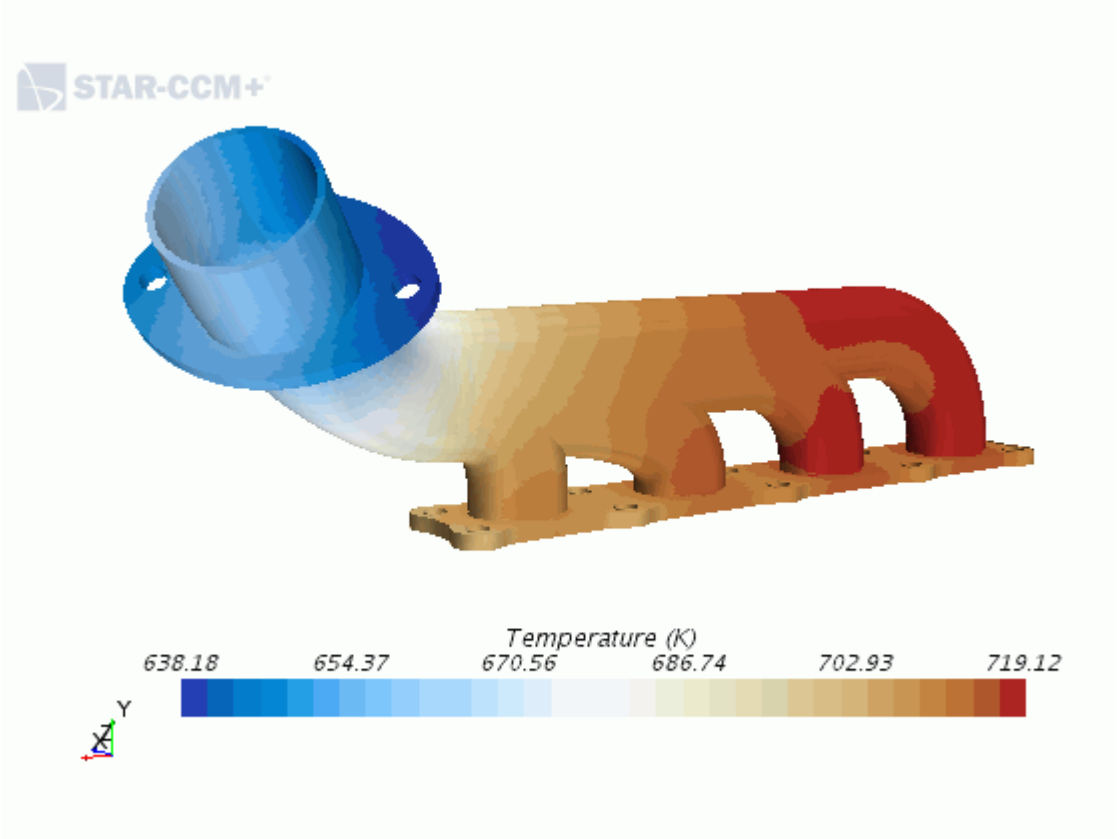
Generate the volume mesh and run the thermal analysis:

1. Click (**Generate Volume Mesh**).
   STAR-CCM+ executes the predefined automated mesh operation, **Thermal Analysis - Polyhedral Mesh**.
2. Click (**Run**).
   The simulation runs for 350 iterations.

Open the predefined temperature scenes to visualize the temperature distribution in the manifold and the gas:

3. Expand the **Scenes** node.
4. Open the **Fluid Temperature** and **Manifold Temperature** scenes.
5. In both scenes, right-click on a blank area and select **Apply Representation** > **Volume Mesh**.
   The temperature distribution appears as shown below:





6. Save the simulation.

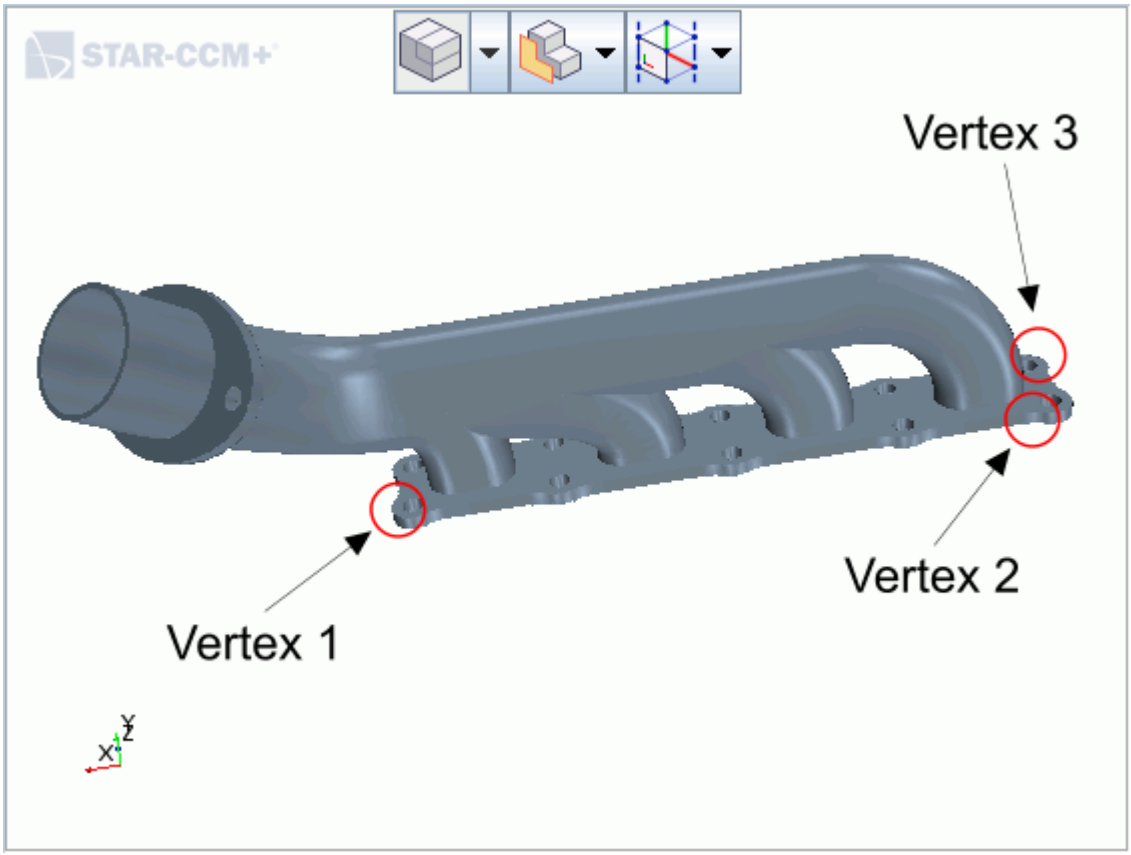
Preparing the Simulation for Stress Analysis

The thermal analysis is complete and you are now ready to set up the simulation for stress analysis. As the thermal and stress analyses use different discretization methods, FV and FE respectively, they require separate regions and continua.

To create the solid region for stress analysis, start by duplicating the manifold part:

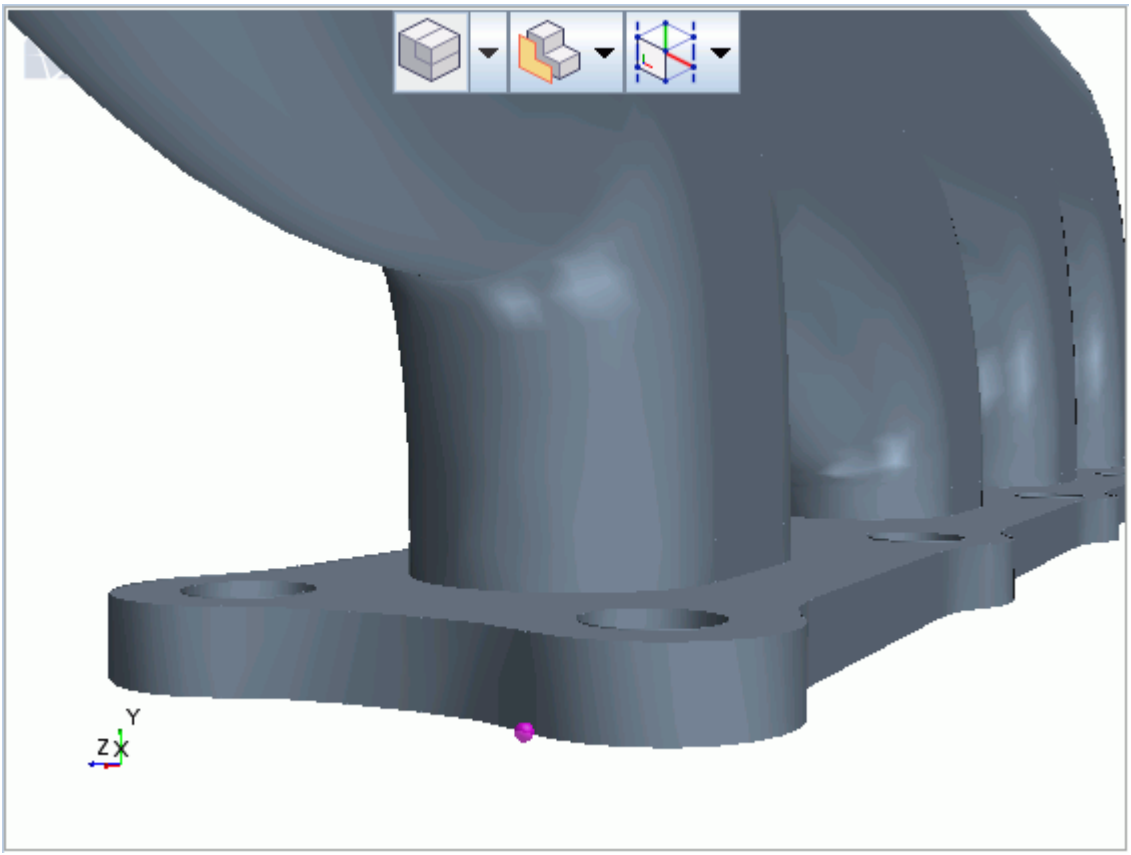
1. Right-click the **Geometry** > **Parts** > **Manifold (Thermal)** node and select **Duplicate**.
2. Rename the new part, **Manifold (Thermal) 2**, to `Manifold (Stress)`.

In this tutorial, you apply constraints at specific vertices on the manifold model. In order to apply constraints at vertices, the vertices must be available as part points. For this tutorial, you require three points at the following locations on the manifold base:



To create the required part points, mark the relevant vertices as feature vertices in 3D-CAD:

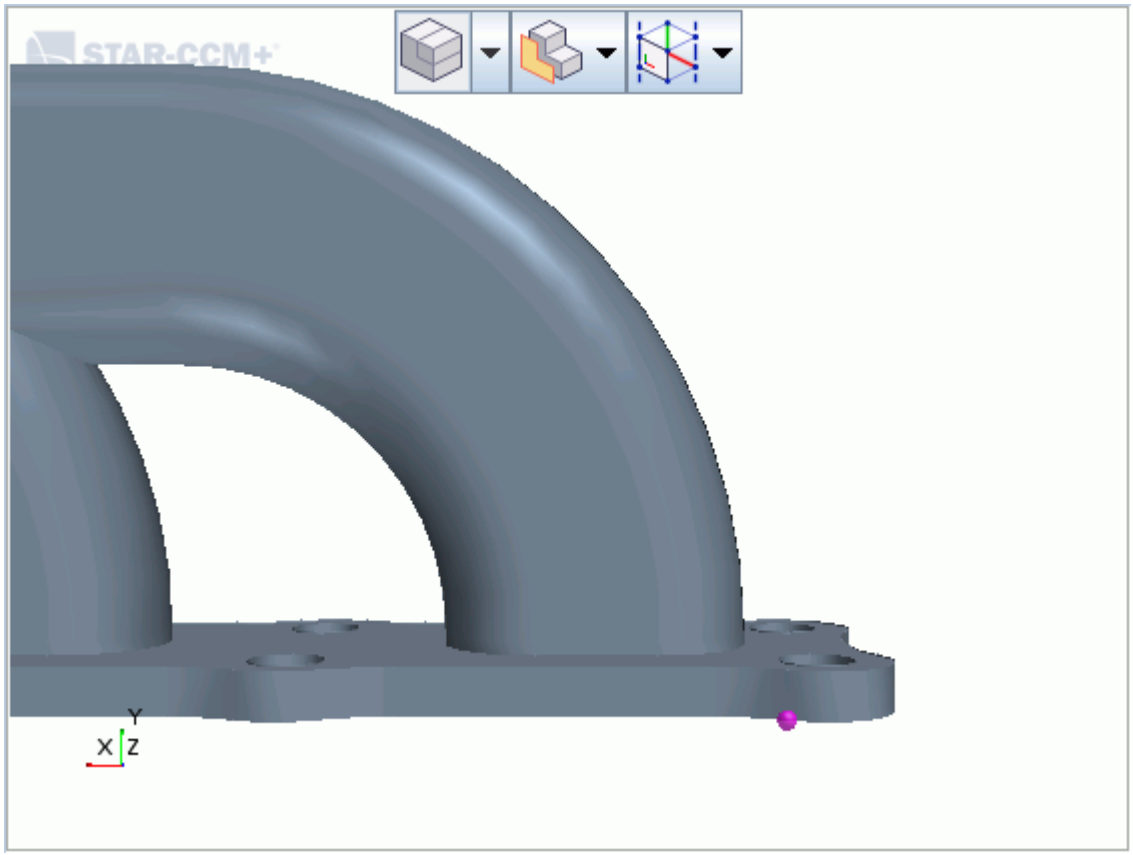
3. Right-click the **Manifold (Stress)** part and select **Edit in 3D-CAD**.
4. In 3D-CAD, rename the **3D-CAD Model 2** > **Bodies** > **Manifold (Thermal)** node to `Manifold (Stress)`.
5. In the *3D-CAD View 1* scene, identify the first vertex:



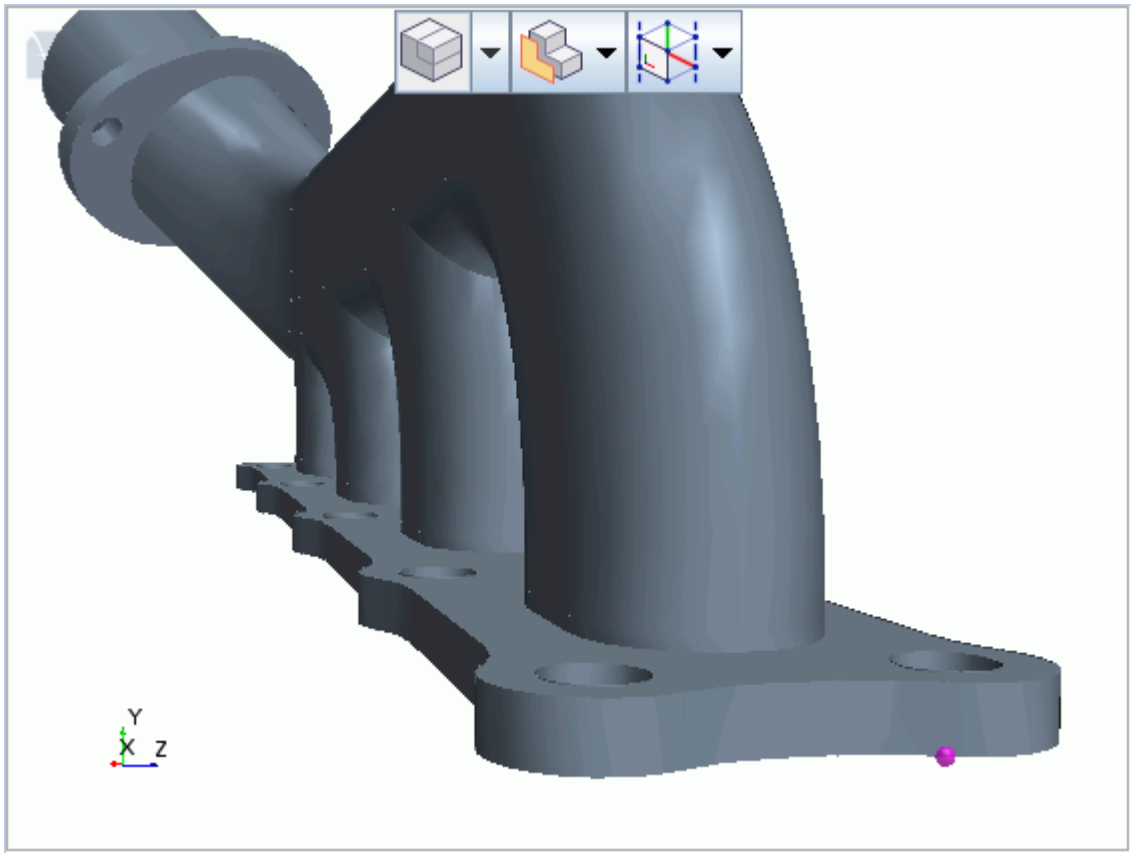
6. Right-click the vertex and select **Rename**. In the *Rename* dialog, enter `Vertex 1` and click **OK**.

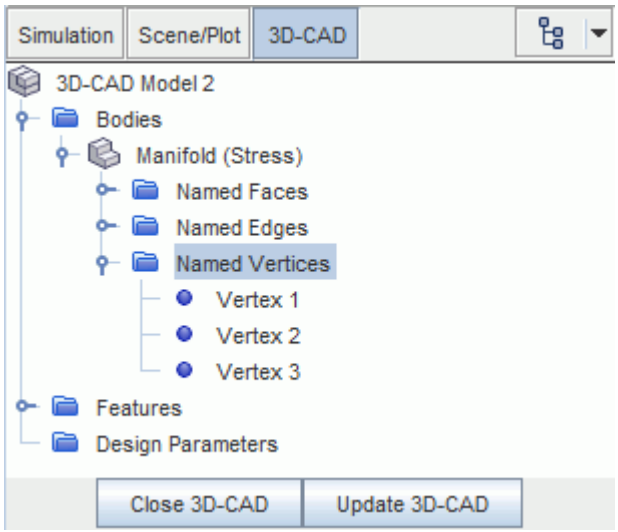
   STAR-CCM+ adds the vertex under the **Manifold (Stress)** > **Named Vertices** node.

7. Using the same procedure, identify the second vertex and rename it to `Vertex 2`.



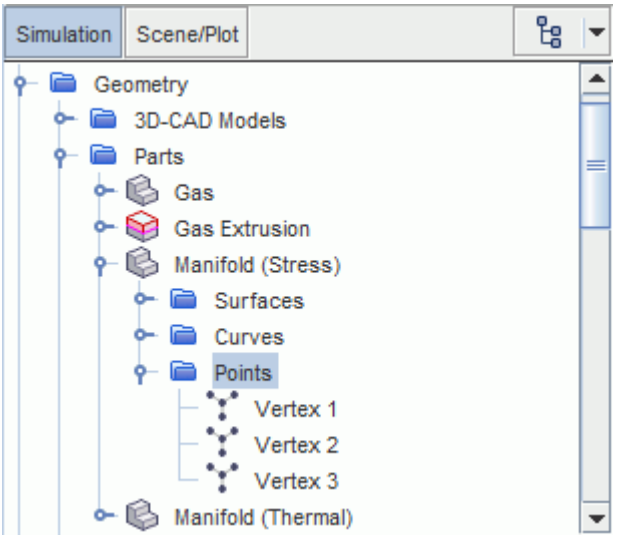
8. Finally, identify the third vertex and rename it to `Vertex 3`.





9. Close 3D-CAD.
10. Right-click the **Geometry** > **Parts** > **Manifold (Stress)** node and select **Update**.

    STAR-CCM+ adds part points corresponding to the feature vertices on the CAD model.

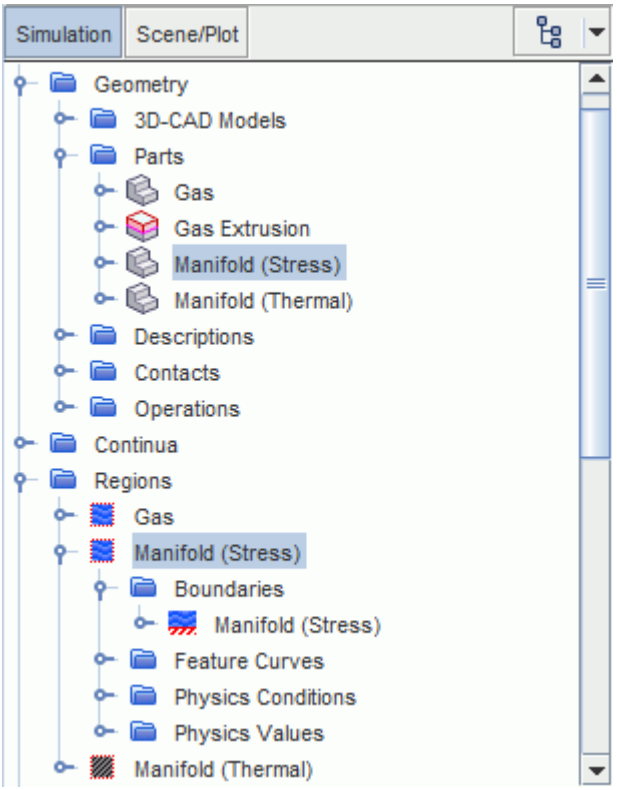


Assign the manifold part to a new region:

11. Right-click the **Manifold (Stress)** node and select **Assign Parts to Regions...**
12. In the *Assign Parts to Regions* dialog:
    a) Select the following options:

Property	Setting
Region option	**Create a Region for Each Part**
Interface option	**Do Not Create Interfaces From Contacts**

    b) Click **Apply** then **Close**.



STAR-CCM+ automatically assigns the **Manifold (Stress)** region to one of the existing physics continua. In the next section, create a physics continuum for stress analysis and assign it to the **Manifold (Stress)** region.

13. Save the simulation as `manifold_stress`.

Specifying Physics Models and Material Properties

Create a physics continuum for the solid FE region, including physics models that account for the manifold displacement in response to thermal loads. Also specify the solid material and adjust the material properties to suit the manifold case.

1. Create a physics continuum and name it `Stress Analysis - Manifold`.
2. For the physics continuum, **Stress Analysis - Manifold**, select the following models:

Group box	Model
Space	**Three Dimensional**
Time	**Steady**
Material	**Solid**
Optional Models	**Solid Stress**
	Material Law Models (selected automatically)
	Specified Temperature Load

The **Specified Temperature Load** model allows you to specify a temperature profile for the solid region, which is then used by the FE Stress solver to compute the resulting displacement.

Assign the physics continuum to the solid stress region:

3. Select the **Regions** > **Manifold (Stress)** node and set *Physics Continuum* to **Stress Analysis - Manifold**.

Define the material laws that approximate the behavior of the material. In this tutorial, you assume that:

- the stress-strain relationship in the material is isotropic linear elastic
- the thermal expansion in response to the applied thermal load is isotropic

4. Expand the **Continua** > **Stress Analysis - Manifold** > **Models** > **Material Law Models** node.
5. Right-click the **Material Laws** node and select **New**.
6. Rename the **Material Laws** > **Material Law 1** node to `Isotropic Linear Elastic`.
7. For the **Isotropic Linear Elastic** law node, select the following models:

Group box	Model
Optional Models	**Linear Elasticity**
Linear Elastic Material Models	**Isotropic Linear Elasticity**
Optional Models	**Thermal Expansion**
Thermal Expansion Material Models	**Isotropic Thermal Expansion**

Select an appropriate material for the manifold and define the material properties:

8. Expand the **Continua** > **Stress Analysis - Manifold** > **Models** > **Solid** node.
9. Replace **Al** with **Ductile Iron (Ductile Iron)**.
10. Expand the **Ductile Iron** > **Material Properties** node.
11. Select the **Material Law** node and set *Type* to **Isotropic Linear Elastic**.
12. Adjust the material properties as follows:

Node	Property	Setting
Poisson's Ratio > **Constant**	*Value*	`0.283`
Thermal Expansion Coefficient > **Constant**	*Value*	`1.38E-5 /K`
Young's Modulus > **Constant**	*Value*	`138000 MPa`
Zero Thermal Strain Reference Temperature > **Constant**	*Value*	`485 K`

13. Save the simulation.

Generating the Mesh

Generate a mesh of tetrahedral elements for the manifold. To improve accuracy of the solution, add mid-side nodes to the tet4 elements to generate quadratic tet10 elements.

1. Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
2. In the *Create Automated Mesh Operation* dialog:
   a) Select the following options:

Parts	**Manifold (Stress)**
Surface Meshers	**Surface Remesher**
Core Volume Meshers	**Tetrahedral Mesher**

   b) Click **OK**.
3. Rename the **Operations** > **Automated Mesh** node to `Stress Analysis - Tetrahedral Mesh`.
4. Edit the **Stress Analysis - Tetrahedral Mesh** > **Default Controls** node and set the following properties:

Node	Property	Setting
Base Size	*Value*	`0.005 m`
Minimum Surface Size	*Percentage of Base*	`30`

5. Right-click the **Stress Analysis - Tetrahedral Mesh** node and select **Execute**.

By default, mesh operations generate a mesh of linear elements. To add mid-side nodes to the linear tet4 elements:

6. Select the **Regions** > **Manifold (Stress)** > **Physics Conditions** > **Mid-side Vertex Option** node and set *Mid-side Vertex* to **Add - Linear Interpolation**.

STAR-CCM+ adds the mid-side nodes during the solution initialization. In this tutorial, you want to add the mid-side nodes before you run the simulation, so that the mid-side nodes are available on the FE mesh when you map the thermal loads to the mesh vertices. Initialize the solution to generate the mid-side nodes:

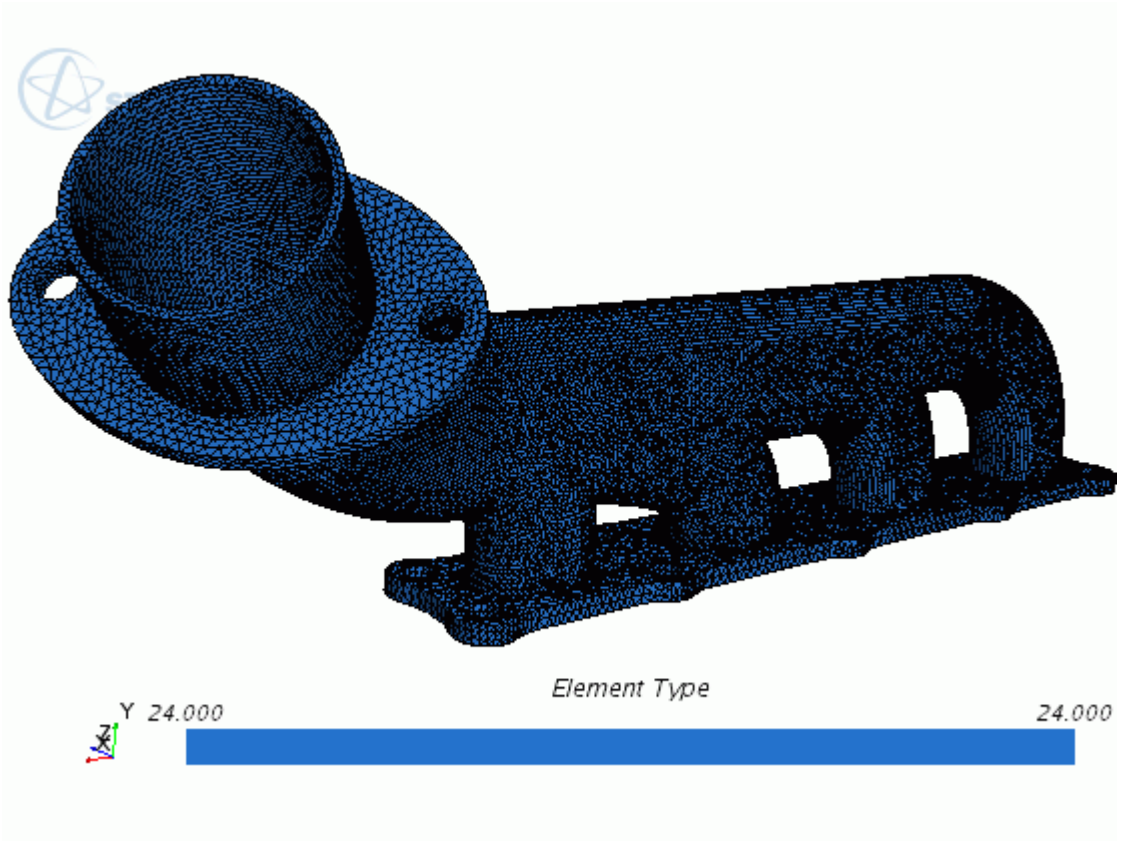
7. Click (**Initialize Solution**).

To visualize the volume mesh and element type:

8. Create a scalar scene and rename it to `Manifold Mesh (FE)`.
9. Edit the **Scenes** > **Manifold Mesh (FE)** > **Displayers** node and set the following properties:

Node	Property	Setting
Outline 1	*Outline*	**Deactivated**
Scalar 1	*Display Mesh*	**Activated**
└ **Parts**	*Parts*	**Parts** > **Manifold (Stress)**
└ **Scalar Field**	*Function*	**Element Type**
└ **Color Bar**	*Color Map*	**indigo-white**

   The mesh appear as shown below:



The number 24 identifies tetrahedral elements with mid-side nodes.

Preparing Analysis Objects

Prepare scalar scenes to visualize the displacement and the stress distribution throughout the manifold.

To visualize the displacement field:

1. Create a scalar scene and rename it to `Displacement`.

Create a vector warp to visualize the displacement field on the deformed geometry:

2. Right-click the **Derived Parts** node and select **New Part** > **Warp** > **Vector Warp...**
3. In the *Create Warp Surface* panel, set the following properties:

Property	Setting
Input Parts	**Parts** > **Manifold (Stress)**
Vector	**Displacement**
Scale Factor	`30`
Display	**Existing Displayer** > **Scalar 1**

4. Click **Create** then **Close**.
5. Edit the **Displacement** > **Displayers** node, and set the following properties:

Node	Property	Setting
Outline 1 > **Parts**	*Parts*	**Derived Parts** > **Vector Warp** (deselect all other parts/regions)
Scalar 1	*Opacity*	`0.3`
└ **Scalar Field**	*Function*	**Displacement** > **Magnitude**
	Units	**mm**
└ **Color Bar**	*Color Map*	**blue-red balanced**

Add a geometry displayer to the scene so you can also visualize the original geometry:

6. Right-click the **Displacement** > **Displayers** node, and select **New Displayer** > **Geometry**.
7. Select the **Displayers** > **Geometry 1** node and set the following properties:

Node	Property	Setting
Geometry 1	*Outline*	**Deactivated**
	Surface	**Activated**
└ **Parts**	*Parts*	**Parts** > **Manifold (Stress)**

To visualize the stress distribution:

8. Create a scalar scene and rename it to `von Mises Stress`.
9. Edit the **von Mises Stress** > **Displayers** node, and set the following properties:

Node	Property	Setting
Outline 1 > **Parts**	*Parts*	**Parts** > **Manifold (Stress)** (deselect all other parts/regions)
Scalar 1		
└ **Parts**	*Parts*	**Parts** > **Manifold (Stress)**
└ **Scalar Field**	*Function*	**Stress Von Mises**
	Units	**MPa**

10. Save the simulation.

Applying and Testing Constraints

Apply minimum constraints to allow for free thermal expansion.

When constraining a part for thermal-structural analysis, avoid applying constraints which are too restrictive and lead to unrealistic stress concentrations. In this tutorial, you apply constraints at three points on the manifold to prevent rigid body motion, while allowing free expansion of the solid structure.

Create a point segment to constrain one point on the manifold in all degrees of freedom:

1. Right-click the **Regions** > **Manifold (Stress)** node and select **Create Segment** > **Point Segment**.
2. Select the **Segments** > **Point Segment 1** node and set the following properties:

Node	Property	Setting
Point Segment 1	*Points*	**Vertex 1**
	Type	**Constraint**
└ **Physics Conditions** > **Solid Stress Constraint**	*Method*	**Fixed** (default)

Define a segment to constrain the second point along two axes:

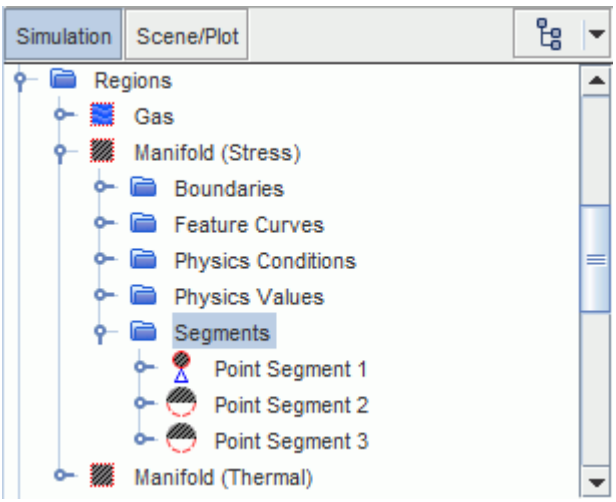
3. Right-click the **Segments** node and select **Create Segment** > **Point Segment**.
4. Select the **Point Segment 2** node and set the following properties:

Node	Property	Setting
Point Segment 2	*Points*	**Vertex 2**
	Type	**Constraint**
└ **Physics Conditions** > **Solid Stress Constraint**	*Method*	**Displacement**
└ **Physics Values** > **Displacement**	*Method*	**Composite**
└ **Composite**	*Constrain X*	**Deactivated**
	Constrain Y	**Activated** (default setting)
	Constrain Z	**Activated** (default setting)

Define a segment to constrain the third vertex along y:

5. Create another point segment.
6. Select the **Point Segment 3** node and set the following properties:

Node	Property	Setting
Point Segment 3	*Points*	**Vertex 3**
	Type	**Constraint**
└ **Physics Conditions** > **Solid Stress Constraint**	*Method*	**Displacement**
└ **Physics Values** > **Displacement**	*Method*	**Composite**
└ **Composite**	*Constrain X*	**Deactivated**
	Constrain Y	**Activated** (default setting)
	Constrain Z	**Deactivated**



In thermal strain analyses, it is important to define realistic constraints. A fixed condition on a surface segment can prevent thermal expansion and lead to unrealistic stress concentrations. To test your set of constraints, increase the temperature uniformly and run the analysis. The resulting stresses must be close to zero:

7. Edit the **Regions** > **Manifold (Stress)** node and set the following properties:

Node	Property	Setting
Physics Conditions > **Specified Temperature Option**	*Method*	**Specified**
Physics Values > **Specified Temperature**	*Method*	**Constant**
	Value	`650 K`

Freeze the flow and energy solvers and run the FE Stress solver for two iterations:

8. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to **352**.
9. Expand the **Solvers** node.
10. Multi-select the **Wall Distance**, **EB K-Epsilon Turbulence**, and **K-Epsilon Turbulent Viscosity** nodes and activate *Solver Frozen*.
11. Select the **Coupled Implicit** solver and activate *Freeze Flow*.
12. Click (**Run**).
13. When the stopping criterion has been satisfied, open the **von Mises Stress** scene and examine the stresses.

    As the stresses are approximately zero, the applied constraints are not too restrictive.
14. Save the simulation.

Applying Thermal Loads

Use the temperature field computed in the thermal analysis as a thermal load in the stress analysis.

Start by mapping the temperature data, available on the CHT analysis FV mesh, to the FE mesh used for stress analysis. To interpolate the temperature data from the Manifold (Thermal) part to the Manifold (Stress) part, create a volume data mapper:

1. Right-click the **Tools** > **Data Mappers** node and select **New Data Mapper** > **Volume Data Mapper**.
2. Rename the **Data Mappers** > **Volume Data Mapper 1** node to `Temperature Mapper`.

Specify the solid part of the CHT analysis, **Manifold (Thermal)**, as the source mesh from which to take temperature. Set the source stencil to specify the location where the solution is available on the source mesh. As the source mesh is a finite volume mesh, the solution is available at the cell centroids.

3. Edit the **Data Mappers** > **Temperature Mapper** node and set the following properties:

Property	Setting
Source Volumes	**Parts** > **Manifold (Thermal)**
Source Stencil	**Cell**

Complete the source settings by specifying temperature as the field to map. When you execute mapping operations, the mapped fields become available as field functions. Assign the mapped temperature field function a suitable name:

4. For the same node, **Temperature Mapper**, set the following properties:

Property	Setting
Scalar Field Functions	**Temperature**
Mapped Names	`CHT - Thermal Load`

Specify the target mesh for the mapping operation. As the target mesh is a finite element mesh, map the source temperature data to the mesh vertices:

5. Edit the **Temperature Mapper** > **Target Specifications** > **Volume 1** node and set the following properties:

Property	Setting
Target Volumes	**Parts** > **Manifold (Stress)**
Target Stencil	**Vertex** (default setting)

Execute the mapping operation:

6. Right-click the **Temperature Mapper** node and select **Map Data**.

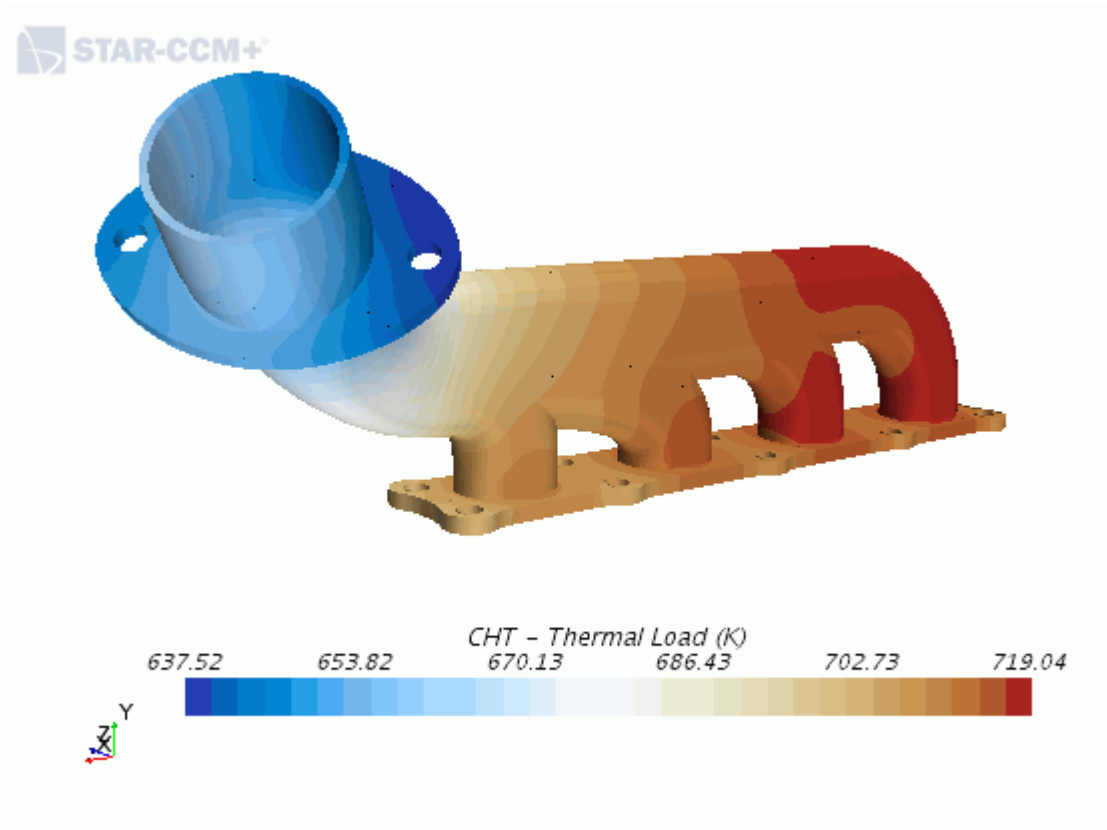
   STAR-CCM+ interpolates the temperature data to the target FE mesh, and generates the corresponding **CHT - Temperature Load** field function.

To visualize the mapped temperature in a scalar scene:

7. Create a scalar scene and rename it to `Thermal Load`.
8. Edit the **Scenes** > **Thermal Load** > **Displayers** node and set the following properties:

Node	Property	Setting
Outline 1	*Outline*	**Deactivated**

Node	Property	Setting
Scalar 1		
└ **Parts**	*Parts*	**Parts** > **Manifold (Stress)**
└ **Scalar Field**	*Function*	**CHT - Thermal Load**
└ **Color Bar**	*Color Map*	**cool-warm**



Finally, use the mapped data field function, **CHT - Temperature Load**, to specify the stress analysis thermal loads:

9. Edit the **Regions** > **Manifold (Stress)** > **Physics Values** node and set the following properties:

Node	Property	Setting
Specified Temperature	*Method*	**Field Function**
	Scalar Function	**CHT - Thermal Load**

10. Save the simulation.

Solving for Solid Stress

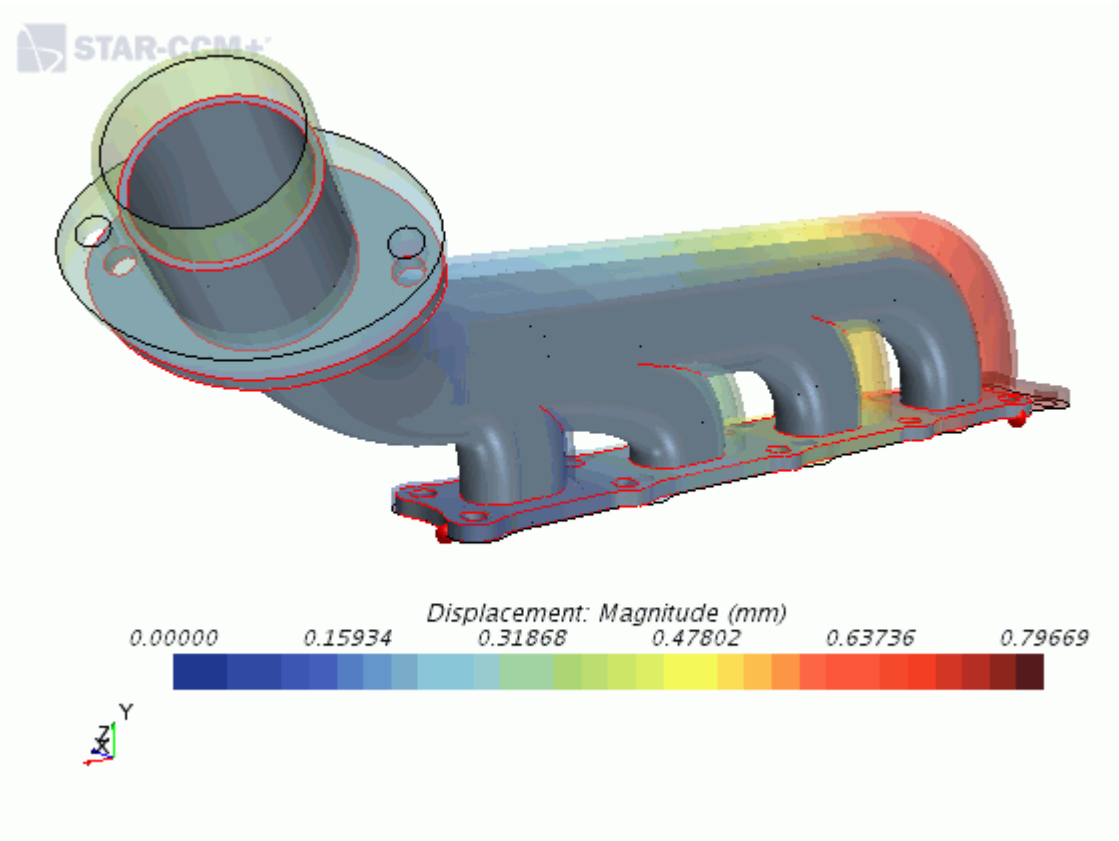
Run the simulation to solve for the manifold displacement in response to the thermal load.

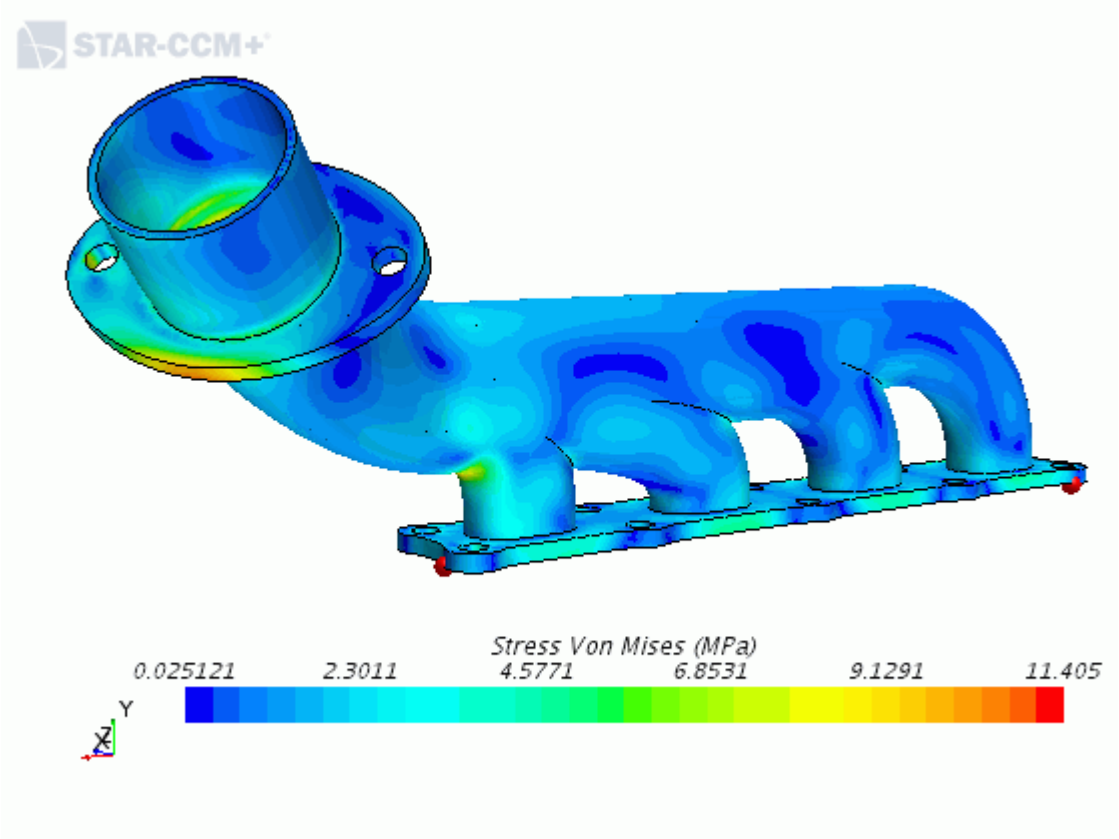
1. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to **354**.
2. Click (**Run**).
3. When the stopping criterion has been satisfied, save the simulation.

Visualizing the Results

Visualize the manifold displacement and the internal stresses.

1. Open the **Displacement** and **von Mises Stress** scenes to visualize the expansion of the manifold in response to the thermal loads, and the internal stresses.





As you left the manifold almost free to expand, by applying very mild constraints only to prevent rigid body motion, the von Mises stress scene shows stresses that are well below the yield stress of 200 MPa.

Summary

This tutorial demonstrated how to perform thermal stress analysis in STAR-CCM+.

This tutorial has introduced the following STAR-CCM+ features:

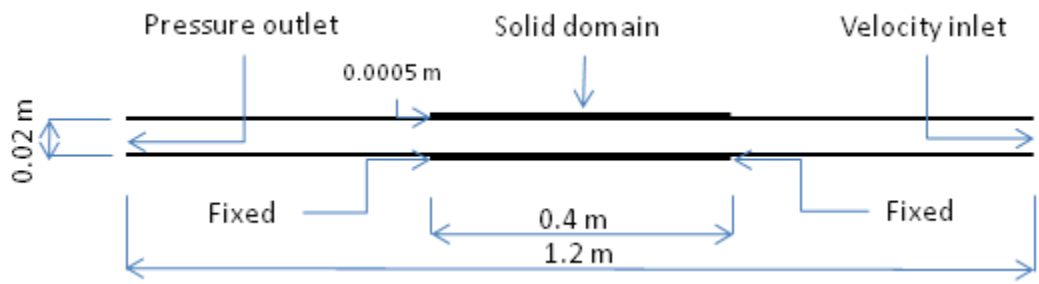
- Generating a tetrahedral mesh with mid-side nodes.
- Applying point constraints.
- Mapping volume data between different mesh representations.
- Applying thermal loads on a finite element mesh.
- Creating a vector warp to visualize the deformed geometry.

Fluid-Structure Interaction: Vibrating Pipe

This tutorial demonstrates how to set up fluid-structure interaction (FSI) problems in STAR-CCM+.

In this simulation, you study the dynamic vibration of a titanium pipe conveying water. As the mass of the pipe is small compared to the mass of the fluid, the physics of the fluid and the structure is coupled. In the first part of the simulation, you apply a body load to the pipe and run a static/steady analysis to seek the static equilibrium solution, obtaining an initial displacement for the pipe. In the second part of the simulation, you remove the body load and run a dynamic analysis, allowing the pipe to vibrate in real time.

The simulation strategy, including the model geometry and assumptions, is summarized below.



	Solid Domain	Fluid Domain
Geometry	**Inner diameter**: 0.02 m	**Diameter**: 0.02 m
	Outer diameter: 0.021 m	**Length**: 1.20 m
	Length: 0.40 m	**Pipe-contact:** only the middle section of the fluid domain is in contact with the pipe
	Symmetry: The problem is symmetric about the yz plane, allowing you to model 1/2 of the geometry.	**Symmetry**: The problem is symmetric about the yz plane, allowing you to model 1/2 of the geometry.
Assumptions and Models	• Material: Ti • Constitutive Law: Linear elastic • Loading Method: Static loading	• Material: H_2O • Equation of State: Compressible liquid with $\rho(p, c_s) = \rho_0 + p/c_s^2$ • Flow Regime: Laminar
	• 2-way FSI: The internal fluid flow influences the vibration modes of the pipe. The vibration of the pipe affects the fluid flow.	
Boundary Conditions	• pipe extremities: constrained in all degrees of freedom • outer surfaces: stress-free	• velocity inlet • pressure outlet

	Solid Domain	Fluid Domain
Type of Analysis	Quasi-static (first run) + Dynamic (second run)	Steady (first run) + Transient (second run)
Discretization and Solution Method	Finite Element (FE)	Finite Volume (FV)
Mesh	Hex8 Elements	Hexahedral Cells

Contents:

Prerequisites
Loading the Initial Simulation
Creating the Fluid-Structure Interface
Selecting the Physics Models
Specifying Material Properties
Generating the Mesh
Defining Motion
Defining Boundary Conditions and Body Load
Preparing Analysis Objects
Running a Static Analysis
Running a Dynamic Analysis
Visualizing the Solution
Summary

Prerequisites

The instructions in the Fluid Structure Interaction - Vibrating Pipe tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Techniques	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+
Using the **Solid Stress** model	Linear Stress Analysis: Cantilever I Beam

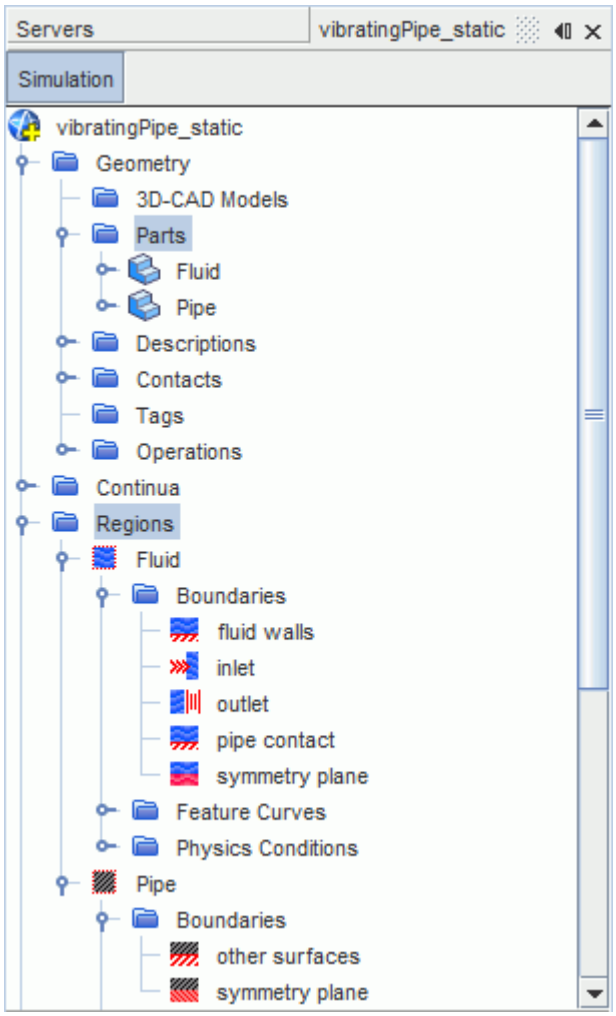
Loading the Initial Simulation

Load the starting simulation for this tutorial. The starting simulation file includes predefined geometry parts that represent the pipe and fluid geometry. Each part is assigned to a region, and part surfaces are assigned to region boundaries as appropriate. The boundary types and symmetry conditions are fully defined.

To load the initial simulation:

1. Launch STAR-CCM+.

2. Select **File** > **Load Simulation...**
3. In the *Load Simulation* dialog, click **Browse...**
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/solidStress/data`.
5. Select `vibratingPipe_start.sim` and click **Open**.
6. In the *Load Simulation* dialog, click **OK**.
7. You can review the predefined settings by expanding the relevant nodes.



The part representing the pipe is assigned to the solid region, **Pipe**. When using the **Solid Stress** model, you set all conditions other than symmetry directly on part surfaces, instead of region boundaries. Therefore, the **Pipe** region only requires two boundary definitions: one for the pipe symmetry plane, and one for all other part surfaces.

The part representing the fluid inside the pipe is assigned to the fluid region, **Fluid**. The fluid part requires a separate boundary definition for each part surface, to define the different boundary types and motion settings.

The boundary types and symmetry conditions are fully set.

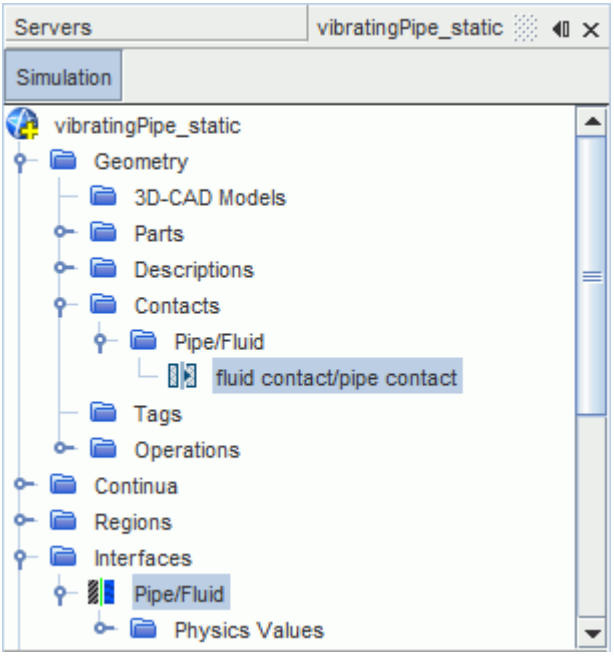
8. Save the simulation to your working directory as `vibratingPipe_static.sim`.

Creating the Fluid-Structure Interface

In fluid-structure interaction problems, the fluid and solid regions exchange field data across a shared interface. As the FE solid stress framework is completely based on parts, you create a fluid-structure interface from the contact between the fluid and solid parts.

To create a parts-based interface for fluid-structure interaction:

1. Expand the **Geometry** > **Contacts** > **Pipe/Fluid** node.
2. Right-click the **fluid contact/pipe contact** node and select **Set New Interface**.



In this simulation, you solve for fluid flow and solid displacements using different discretization methods, Finite Volume (FV) and Finite Element (FE) respectively. The different mesh topologies require an interface of type **Mapped Contact Interface**, which allows for data mapping between the FV and FE meshes.

3. Select the **Interfaces** > **Pipe/Fluid** node and set *Type* to **Mapped Contact Interface**.
4. Save the simulation.

Selecting the Physics Models

The starting simulation file also includes predefined physics continua, **Fluid Physics** and **Solid Physics**, that are assigned to the fluid and the solid region, respectively. For each continuum, activate appropriate physics models to specify the fluid and the pipe behavior, including coupling at the fluid-solid interface.

In this tutorial, you model the water flow as laminar and compressible. To specify the equation of state of the compressible fluid, activate the **User Defined EOS** model, that allows you to define the fluid density as a function of pressure and speed of sound.

1. For the physics continuum, **Continua** > **Fluid Physics**, select the following models:

Group Box	Model
Space	**Three Dimensional**
Time	**Implicit Unsteady**
Material	**Liquid**
Flow	**Segregated Flow**
	Gradients (Selected automatically)
Equation of State	**User Defined EOS**
Viscous Regime	**Laminar**

Specify the fluid initial velocity:

2. Select the **Fluid Physics** > **Initial Conditions** > **Velocity** node and set *Value* to `[0, 0, 0.001] m/s`.

In this simulation, you model the pipe as a linear elastic material. To solve for the displacement of the pipe in response to the applied body load, activate the **Solid Stress** model. To take into account the interaction between the fluid and the solid, activate the **Fluid Structure Coupling** model. This model allows for implicit data exchange at the fluid-solid interface.

3. For the physics continuum, **Solid Physics**, select the following models:

Group Box	Model
Space	**Three Dimensional**
Time	**Implicit Unsteady**
Material	**Solid**
Optional Models	**Solid Stress**
	Material Law Models (Selected automatically)
	Fluid Structure Coupling

Define the material law that approximates the behavior of the solid:

4. Expand the **Solid Physics** > **Models** > **Material Law Models** node.
5. Right-click the **Material Laws** node and select **New**.
6. Rename the **Material Laws** > **Material Law 1** node to `Isotropic Linear Elastic`.
7. For the **Isotropic Linear Elastic** law node, select the following models:

Group box	Model
Material Models	**Linear Elasticity**
Linear Elastic Material Models	**Isotropic Linear Elasticity**

8. Save the simulation.

Specifying Material Properties

Define the equation of state of the compressible fluid by specifying the fluid density in terms of pressure and speed of sound. Also, specify the pipe material.

For the definition of the fluid properties, you are provided with predefined user field functions, that specify the fluid density, the speed of sound in water, and the density pressure derivative as follows:

Field Function	Definition
H2O Speed Of Sound	$c_0 = 1440$
H2O Density	$\rho = 1000 + \frac{\mathrm{p}}{c_0^2}$
H2O Density Pressure Derivative	$\frac{d\rho}{d\mathrm{p}} = \frac{1}{c_0^2}$

These field functions do not require units specification, as the correct units are applied when you assign the field functions to the corresponding material properties.

1. Select the **Continua** > **Fluid Physics** > **Models** > **User Defined EOS** node and activate *Compressible*.

To specify the fluid material properties:

2. Expand the **Continua** > **Fluid Physics** > **Models** > **Liquid** > **H2O** node.
3. Edit the **Material Properties** node and set the following properties:

Node	Property	Setting
Speed of Sound	*Method*	**Field Function**
└ **Field Function**	*Scalar Function*	**H2O Speed Of Sound**
Density	*Method*	**Field Function**
└ **Field Function**	*Scalar Function*	**H2O Density**
Density pressure derivative	*Method*	**Field Function**
└ **Field Function**	*Scalar Function*	**H2O Density Pressure Derivative**

Select the pipe material from the STAR-CCM+ material library:

4. Right-click the **Continua** > **Solid Physics** > **Models** > **Solid** > **Al** node and select **Replace with...**
5. In the *Replace Material* dialog, select **Ti (Titanium)** from the material database, and click **OK**.

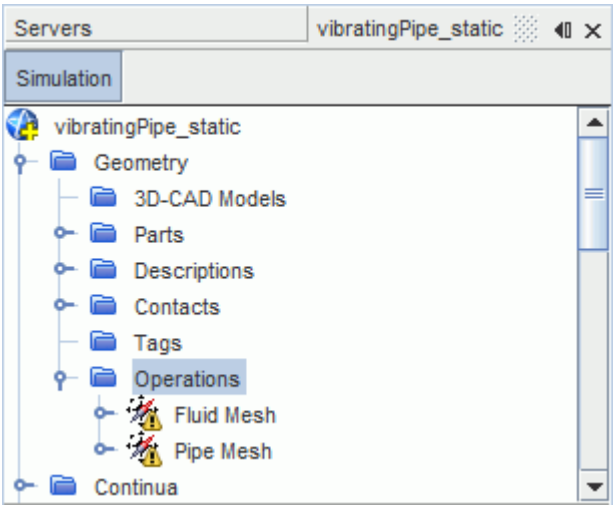
Assign the isotropic linear elastic material law to the solid material:

6. Select the **Ti** > **Material Properties** > **Material Law** node and set *Type* to **Isotropic Linear Elastic**.
7. Save the simulation.

Generating the Mesh

You are provided with predefined parts-based mesh operations, which are fully set up and ready for execution. These operations use the Directed Mesher to generate a hexahedral cell mesh for the fluid domain, and a mesh of linear hexahedral elements (hex8) for the solid domain.

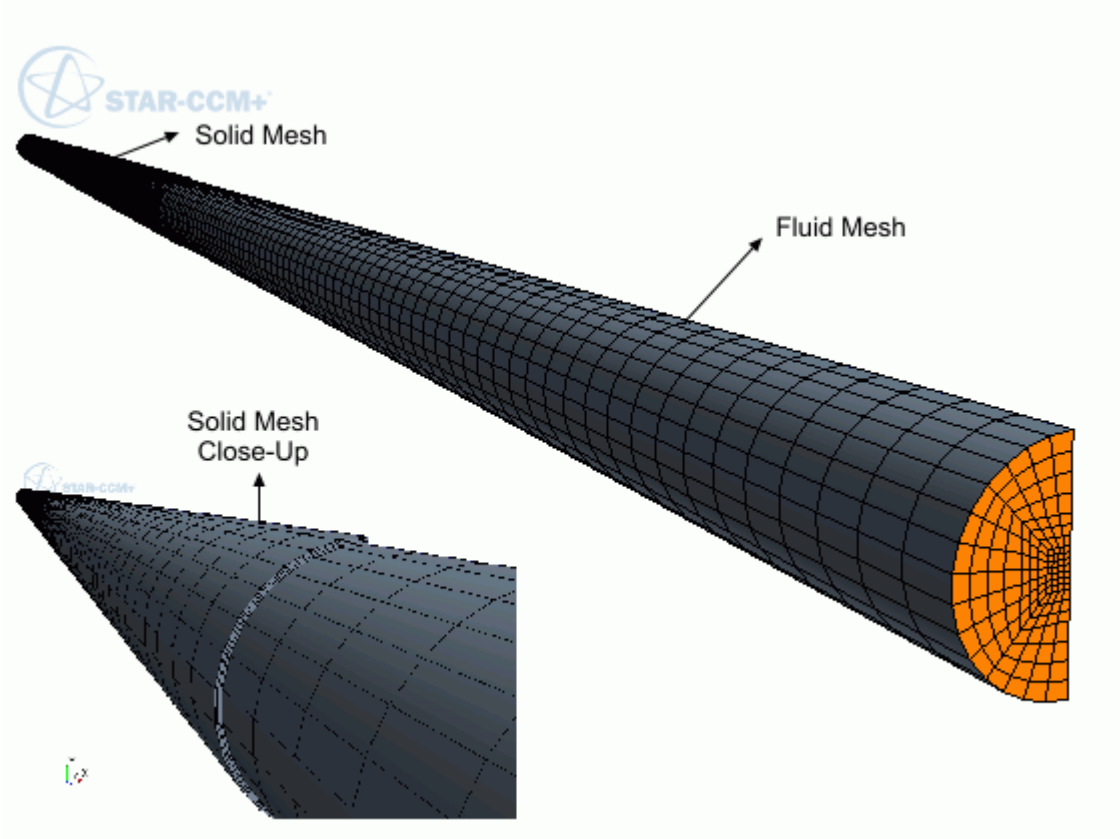
1. Expand the **Geometry** > **Operations** node.



Execute the predefined mesh operations sequentially:

2. Right-click the **Operations** node and select **Execute All**.

   The **Fluid Mesh** operation runs first, and sweeps the surface mesh that is defined on the outlet through the fluid volume, generating the fluid hexahedral mesh. The **Pipe Mesh** operation then sweeps the surface mesh available on the contacting fluid through the pipe volume, up to the pipe outer wall, generating the pipe hexahedral mesh. As the **Pipe Mesh** operation uses the existing fluid mesh, it is important that you execute the operations sequentially, without changing the order of the operations in the pipeline.
3. Visualize the volume mesh with a method of your choice, for example creating a mesh scene.



Defining Motion

Select the appropriate motion models to simulate the deformation of the fluid and solid regions under the external load on the pipe. For the solid region, define a **Solid Displacement** motion, that allows the pipe mesh to deform in real time based on the computed displacements. For the fluid region, define a **Morphing** motion, that allows the fluid mesh to deform based on the displacements mapped at the interface.

1. Right-click the **Tools** > **Motions** node and select **New** > **Morphing**.
2. Repeat the previous step and select **New** > **Solid Displacement**.

To assign the motion models to the fluid and solid regions:

3. Expand the **Regions** > **Fluid** > **Physics Values** node.
4. Select the **Motion Specification** node and set *Motion* to **Morphing**.
5. Expand the **Regions** > **Pipe** > **Physics Values** node.
6. Select the **Motion Specification** node and set *Motion* to **Solid Displacement**.

STAR-CCM+ automatically sets the morpher method for the fluid boundary representing the fluid-structure interface to **Solid Stress**. Set the **pipe contact** boundary to follow the interface motion:

7. Expand the **Regions** > **Fluid** > **Boundaries** node.
8. Select the **pipe contact** > **Physics Conditions** > **Morpher Specification** node and set *Specification* to **Floating**.

9. Save the simulation.

Defining Boundary Conditions and Body Load

Specify the fluid velocity at the inlet, and constrain the pipe extremities in all degrees of freedom. Also, apply a body load on the pipe to reach an initial deformed configuration.

For the fluid, you can specify boundary conditions under the relevant region boundary nodes. For the solid pipe, the parts-based approach of the STAR-CCM+ FE framework requires you to set constraints on part surfaces using segments.

To specify the fluid velocity at the inlet:

1. Expand the **Regions** node.
2. Select the **Fluid** > **Boundaries** > **Inlet** > **Physics Values** > **Velocity Magnitude** node and set *Value* to `0.001 m/s`.

To constrain the pipe:

3. Right-click the **Pipe** node and select **Create Segment**.
4. Rename the **Pipe** > **Segments** > **Surface Segment 1** to `Fixed Ends`, and set the following properties:

Property	Setting
Surfaces	**left end**
	right end
Type	**Constraint**

   The default constraining method is **Fixed** and is appropriate to this case, as it prevents the pipe extremities from moving in any directions.

The default settings for the fluid-structure interface are also appropriate to this case. To review the settings:

5. Expand the **Interfaces** > **Pipe/Fluid** > **Physics Conditions** node.

   The **FSI Added Mass Control Option** is set to **Auto**. This setting instructs STAR-CCM+ to automatically estimate the volume of displaced fluid per unit area, which is used as a preconditioner to the implicit algorithm, to speed up convergence.

To apply a static body load on the pipe:

6. Select the **Pipe** > **Physics Conditions** > **Body Load Option** node and set *Method* to **Specified**.
7. Select the **Physics Values** > **Body Load** node and set *Value* to `[0,-1.55E7,0]N/m^3`.
8. Save the simulation.

Preparing Analysis Objects

Prepare plots and scenes to monitor the pipe displacement during the solution progress.

Use the point probe provided in the starting simulation file to monitor the displacement in the pipe center:

1. Right-click the **Reports** node and select **New Report** > **Maximum**.

2. Rename the **Maximum 1** node to `Y-Displacement`.
3. Edit the **Reports** > **Y-Displacement** node, and set the following properties:

Property	Setting
Field Function	**Displacement** > **Laboratory** > **j**
Units	**mm**
Parts	**Derived Parts** > **Point**
Smooth Values	**Activated**

4. Right-click the **Y-Displacement** report and select **Create Monitor and Plot from Report**.

To visualize the deformation of the pipe in a scalar scene, scale the displacements using a vector warp:

5. Create a scalar scene and rename it to `Y-Displacement`.
6. Right-click the **Derived Parts** node and select **New Part** > **Warp** > **Vector Warp...**
7. In the *Create Warp Surface* in-place dialog, set the following properties:

Property	Setting
Input Parts	**Parts** > **Pipe**
Vector	**Displacement**
Auto Scale	**Activated**
Display	**Existing Displayer** > **Scalar 1**

8. Click **Create** then **Close**.
9. Expand the **Scenes** > **Y-Displacement** > **Displayers** > **Scalar 1** node and set the following properties:

Node	Property	Setting
Scalar Field	*Function*	**Displacement** > **Laboratory** > **j**
	Units	**mm**
Color Bar	*Color Map*	**blue-red balanced**

10. Click (**Save-Restore-Select views**) and select **Views** > **-X** > **Up +Y**.

To visualize the pressure distribution throughout the fluid:

11. Create a scalar scene and rename it to `Pressure`.
12. Edit the **Scenes** > **Pressure** > **Displayers** node and set the following properties:

Node	Property	Setting
Scalar 1	*Contour Style*	**Smooth Filled**
└ **Parts**	*Parts*	**Parts** > **Fluid**
└ **Scalar Field**	*Function*	**Pressure**

13. Click (**Save-Restore-Select views**) and select **Views** > **-X** > **Up +Y**.

14. Save the simulation.

Running a Static Analysis

Run a static analysis to compute the static equilibrium solution.

Use the **Implicit Unsteady** solver to reach a static solution. The first order temporal discretization scheme combined with a sufficiently large time-step has sufficient numerical damping to drive the simulation to the static solution quickly. In addition, in the static configuration the pipe velocity is zero, so you can neglect the flux related to the motion of the pipe.

1. Edit the **Solvers** node and set the following properties:

Node	Property	Setting
Implicit Unsteady	*Time-Step*	`1.0 s`
Solid Stress Solver > **Static Analysis**	*Static Analysis*	**Activated**
Mesh Morpher	*Recompute Interfaces*	**Deactivated**
	Ignore Gridflux Terms	**Activated**

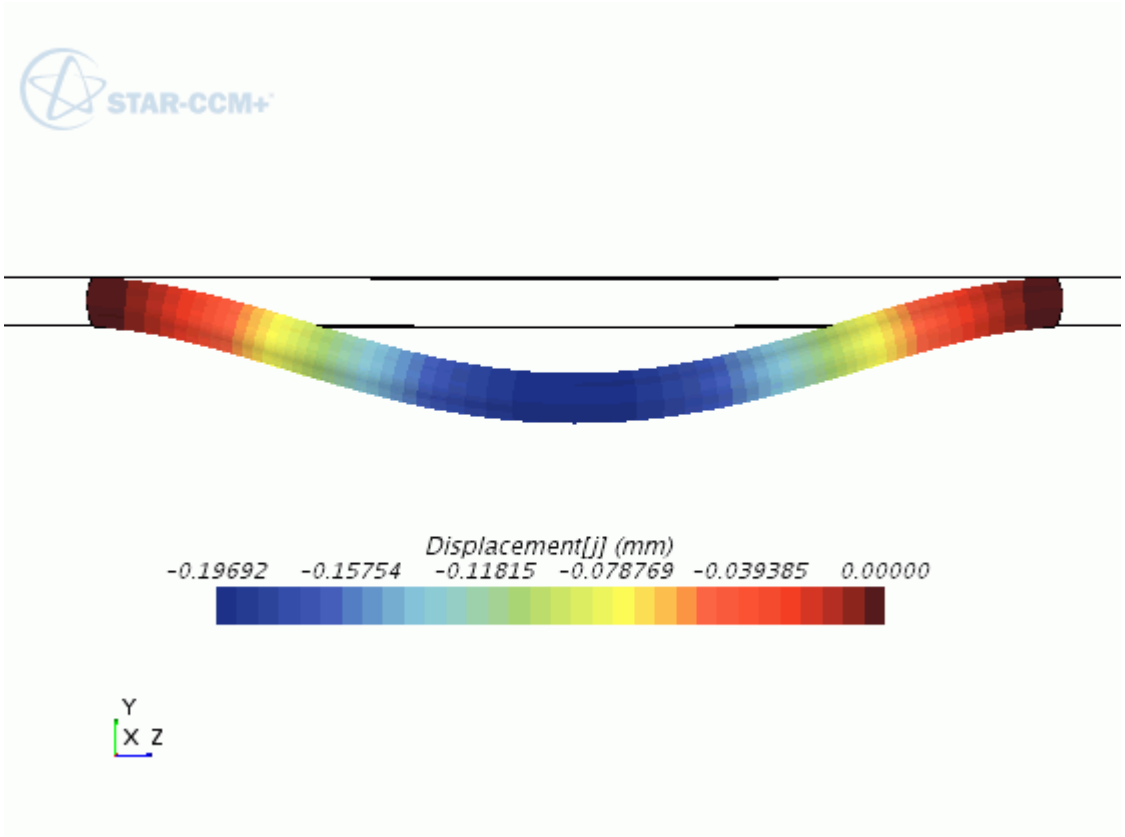
Note: If you do not deactivate *Recompute Interfaces*, the displacement solution on the interface is over-written at the end of each iteration.

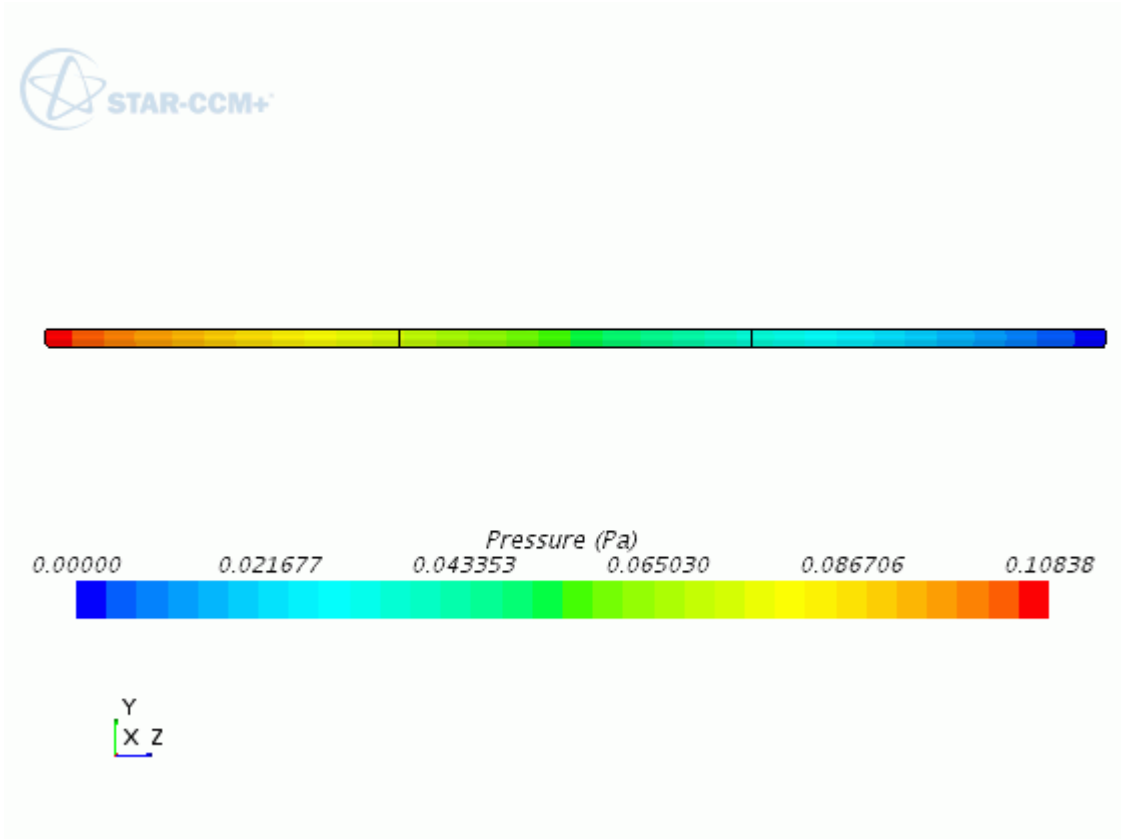
2. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Inner Iterations	*Maximum Inner Iterations*	`15`
Maximum Physical Time	*Maximum Physical Time*	`5 s`

You are now ready to run the static analysis:

3. Click (**Run**).
4. When the stopping criterion is satisfied, save the simulation.
5. Open the **Y-Displacement** and **Pressure** scalar scenes to visualize the displacement of the pipe due to the body load, and the fluid pressure.


STAR-CCM+
Displacement[j] (mm)
-0.19692 -0.15754 -0.11815 -0.078769 -0.039385 0.00000
Y
X Z


STAR-CCM+
Pressure (Pa)
0.00000 0.021677 0.043353 0.065030 0.086706 0.10838
Y
X Z

Running a Dynamic Analysis

Remove the body load from the pipe, and allow it to vibrate in real time.

To remove the body load:

1. Select the **Regions** > **Pipe** > **Physics Conditions** > **Body Load Option** node and set *Method* to **None**.

The pipe-water system has a vibration period of about 0.0023 s. For the dynamic analysis, choose a time-step size of about 1/100th of the natural period of the pipe/water system, which is sufficiently small to give high accuracy and stability. To reduce numerical damping, use the 2nd-order temporal discretization scheme:

2. Edit the **Solvers** node and set the following properties:

Node	Property	Setting
Implicit Unsteady	*Time Step*	`2.3e-5`
	Temporal Discretization	**2nd-Order**
Solid Stress Solver > **Static Analysis**	*Static Analysis*	**Deactivated**
Mesh Morpher	*Ignore Gridflux Terms*	**Deactivated**

3. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Physical Time	*Maximum Physical Time*	`0.0036 s`
Maximum Steps	*Enabled*	**Deactivated**

Optionally, you can export the displacement scene as a graphics file at specified time intervals. You can then use the graphics files to form animations of the pipe vibration using software of your choice:

4. Edit the **Scenes** > **Y-Displacement** > **Attributes** node and set the following properties:

Node	Property	Setting
Update	*Save To File*	**Activated**
	Trigger	**Time Step**
└ **Time-Step Frequency**	*Frequency*	`6`
└ **Hardcopy**	*Output Width*	`800`
	Output Height	`600`

   STAR-CCM+ saves the scene to the current directory every six time-steps, for a total of 26 `.png` files.

Before running the transient simulation, reset the iteration number to start from one:

5. From the **Solution** menu, select **Clear Solution...**
6. In the *Clear Solution* dialog, deactivate *Fields* and *Reset Mesh*, then click **OK**.

7. Click (**Run**).

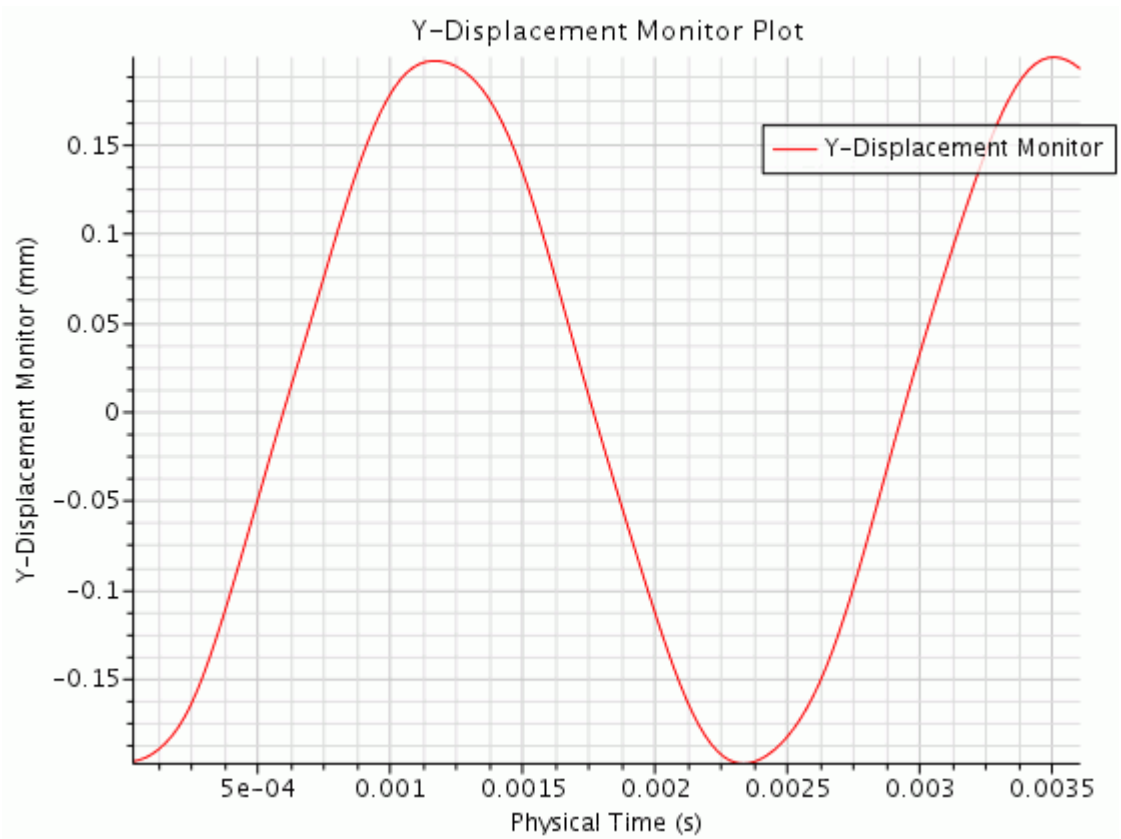
   The iteration progress can be monitored in the *Output* window. Monitor the convergence by viewing the **Residuals** using the visualization tabs in the *Graphics* window.

8. When the solution is complete, save the simulation as `vibratingPipe_dynamic.sim`.

Visualizing the Solution

Visualize the oscillation of the pipe-water system.

1. In the *Graphics* window, open the **Y-Displacement Monitor Plot**.



   The initial load determines the amplitude of oscillation, whereas the presence of fluid modifies the period of oscillation of the pipe. The natural period of the system pipe-water is about 0.0023 s.

   If you set up the simulation to save the displacement scene during the solution run, you can use the graphics files to form an animation of the pipe vibration.

Summary

This tutorial demonstrated how to set up fluid-structure interaction (FSI) problems in STAR-CCM+.

This tutorial has introduced the following STAR-CCM+ features:

- Creating contact-based interfaces.
- Setting up the fluid-structure interface.
- Specifying fluid and solid material properties.
- Defining motion.
- Applying constraints and body loads to the solid part.
- Running a static and dynamic analysis.
- Visualizing displacement.

Aeroacoustics

The tutorials in this set illustrate various STAR-CCM+ features for solving aeroacoustic simulations.

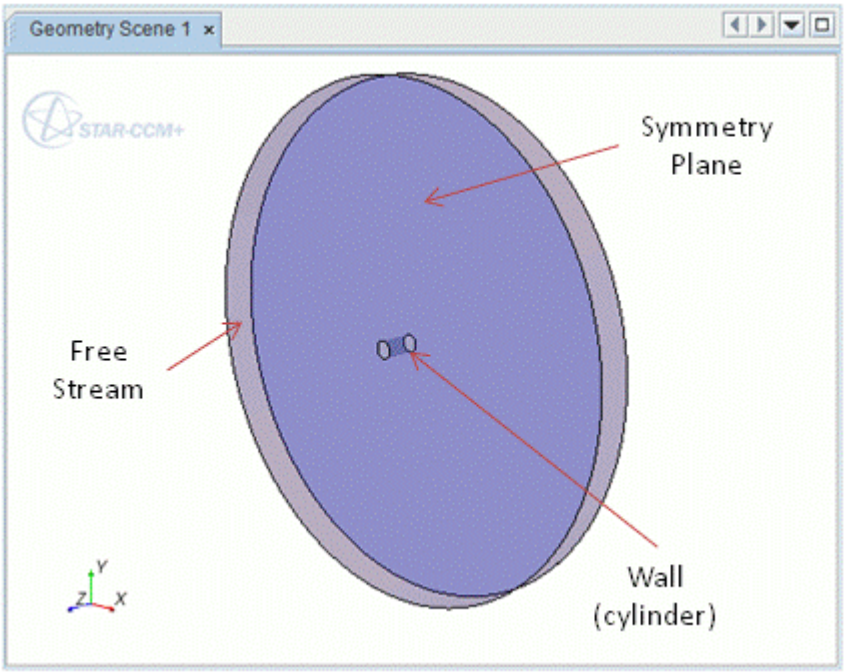
Contents:

Broadband Models: Noise from a Cylinder (Preparation)
DES and FW-H On-The-Fly: Noise from a Cylinder (Unsteady Analysis)
Ffowcs Williams-Hawkings: Sound Propagation
Signal Post-Processing: FFT and Wavenumber
Acoustic Wave Modeling: Noise from a Cylinder

Broadband Models: Noise from a Cylinder (Preparation)

This tutorial demonstrates how to set up aeroacoustics problems in STAR-CCM+. The flow around a cylinder is used in the simulation to describe the required steps.

The geometry is shown below.



The geometry consists of a cylinder, 20 mm in diameter, placed at the center of a three-dimensional circular computational domain, 0.6 m in diameter. The free-stream air velocity and temperature are set to be 50 m/s and 300K, respectively, corresponding to a Mach number of 0.144 (50.0/347.2) and a Reynolds number of 6.38E4 (1.184*50*0.02/1.85563E-5).

According to the recommended procedure, you first run in steady state to:

- Use the broadband models to understand where the aeroacoustics sources are located and to indicate where the mesh needs refining.
- Use the Mesh Frequency Cutoff estimator to prescribe the absolute cell size in the source region to satisfy frequency resolution requirements in transient flow.

Then, you activate an LES-based turbulence model in transient mode to capture the aeroacoustics sources directly and to:

- Use the post-processing tools to perform spectral analysis.
- Export the CFD fields for far-field noise propagation (suitable for non-compact sources).

Contents:

Loading the Base Simulation
Visualizing the Geometry

Generating the Volume Mesh
Selecting Fluid Physics Models
Setting Global Initial Conditions
Defining Boundary Conditions
Setting Up a Plane Section
Setting Up Scalar Scenes
Monitoring Force Coefficients
Setting Stopping Criteria
Running the Simulation
Viewing the Results
Applying the Broadband Noise Sources
Results Analysis and Consequences
Refining the Volume Mesh
Setting New Stopping Criteria
Running the Simulation
Viewing Results After Mesh Refinement
Summary

Loading the Base Simulation

A base simulation file has been prepared for this analysis. Continue by copying the file from the STAR-CCM+ installation directory, starting the simulation, loading the base simulation file and saving it under a different file name.

Note: Use the double precision version of STAR-CCM+ for all aeroacoustic and hydroacoustic analyses.

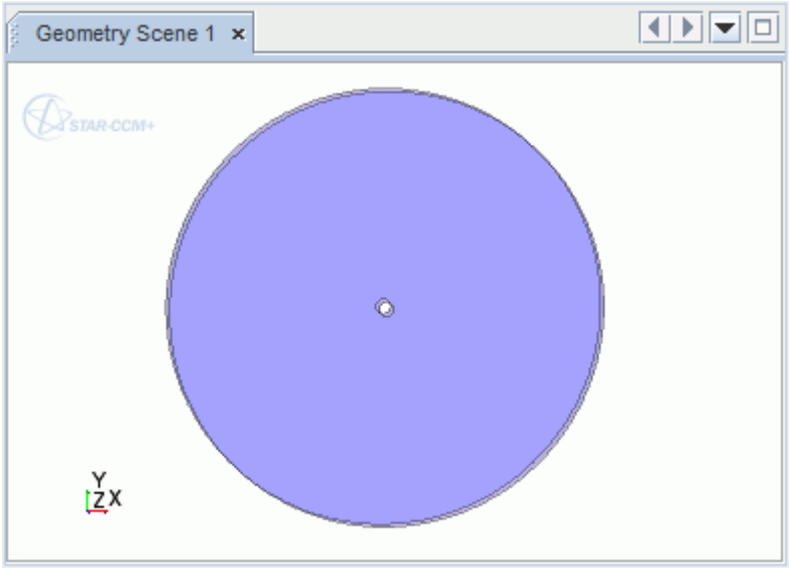
1. Create a directory for the tutorial called `aeroacoustics`.
2. In this directory, copy the file containing the problem geometry (`cylinderBase.sim`) from the `doc/startutorialsdata/aeroacoustics/data` subdirectory of your STAR-CCM+ installation directory. Also copy the two macro files, `compareRmsDragLiftCoefficient.java` and `compareRmsDragLiftCoefficientRefinedMesh.java`.
3. Start up a double precision version of STAR-CCM+ in a manner that is appropriate to your working environment.
4. Click (**Load Simulation**) from the tool bar and open `cylinderBase.sim`.
5. Save the new simulation to disk under file name `cylinder.sim`.

Visualizing the Geometry

Display the geometry scene.

1. Right-click the **Scenes** > **Geometry Scene 1** node and select **Open**.
2. Click (**Make Scene Transparent**) from the tool bar.

A similar view to the one shown below appears.



Store the current view.

3. Click (**Save-Restore-Select views**) in the toolbar and select **Store Current View**.
4. Examine the *Geometry Scene 1* display in the *Graphics* window by adjusting the viewing angle and zooming as appropriate. Click and select **Restore View** > **View 1**.

Generating the Volume Mesh

The meshing models and mesh reference values for this simulation have already been set up. A volumetric control has been created for mesh refinement, to capture the separation behind the cylinder, and to capture the vortex shedding.

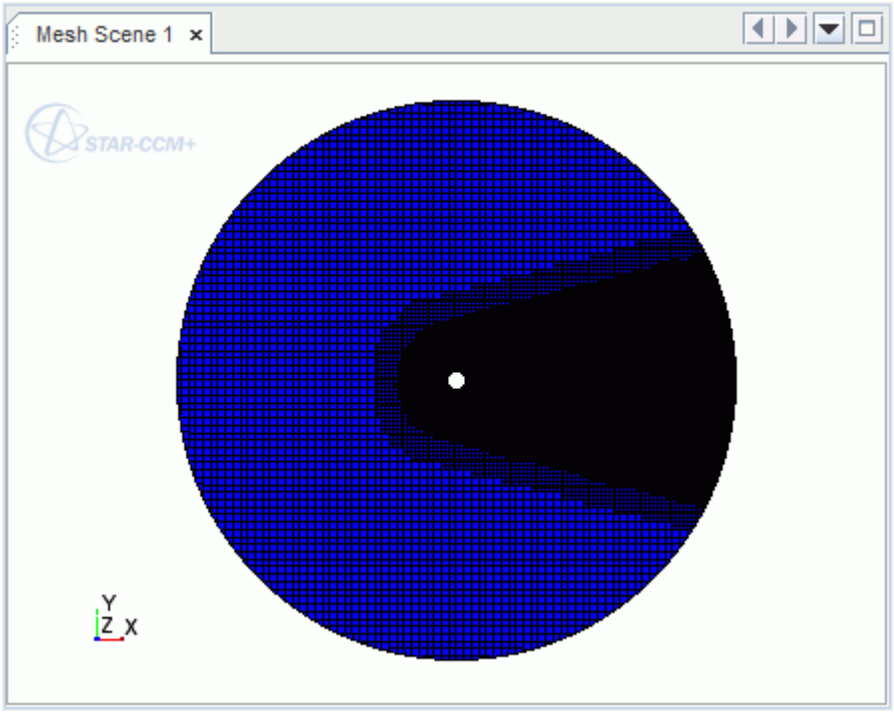
The mesh is ready to be generated.

1. Click (**Generate Volume Mesh**) in the **Mesh Generation** toolbar.

To visualize the volume mesh:

2. Right-click the **Scenes** node and select **New Scene** > **Mesh**.

The mesh appears as shown below.



Selecting Fluid Physics Models

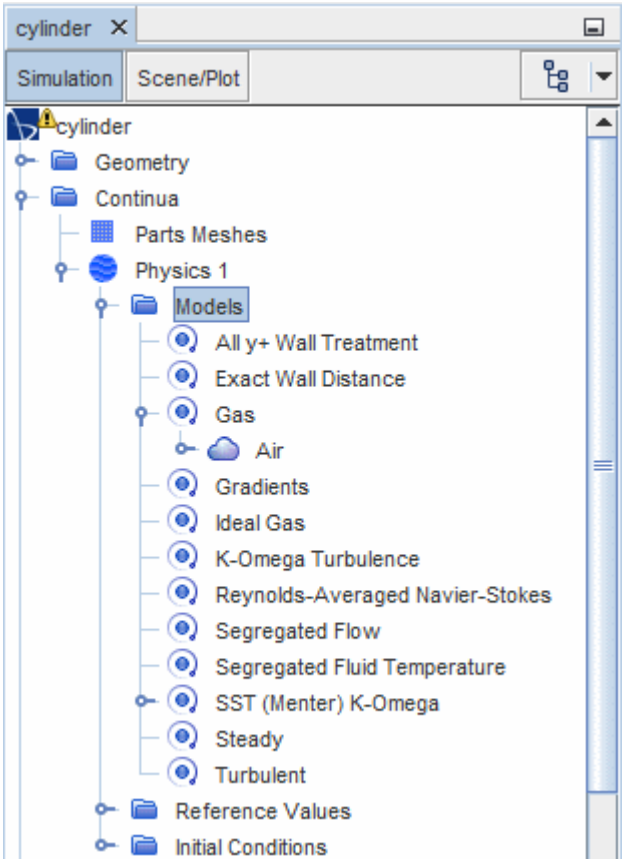
1. For the physics continuum,**Continua** > **Physics 1**, select the following models in order:

   The models define the primary variables of the simulation, including pressure, temperature and velocity. Specify the mathematical formulation that is used to generate the solution.

Group Box	Model
Time	**Steady**
Material	**Gas**
Flow	**Segregated Flow**
Equation of State	**Ideal Gas**
Energy	**Segregated Fluid Temperature**
Viscous Regime	**Turbulent**
Reynolds-Averaged Turbulence	**K-Omega Turbulence**
	Exact Wall Distance (selected automatically)

2. Click **Close**.

The selected models appear as shown below:



Setting Global Initial Conditions

To enhance convergence, initialize the flow field with a constant velocity of 50 m/s.

To set the initial conditions:

1. Select the **Continua** > **Physics 1** > **Initial Conditions** > **Velocity** node, make sure the *Method* property is set to **Constant**, and set *Value* to `[ 50, 0.0, 0.0]` m/s.

   All other values remain at default.
2. Save the simulation.

Defining Boundary Conditions

It is only necessary to set the Mach number of the freestream boundary, as all other settings have acceptable default values.

- Select the **Regions** > **Body 1** > **Boundaries** > **Freestream** > **Physics Values** > **Mach Number** node, make sure the *Method* property is set to **Constant**, and set the *Value* to `0.144`.

Setting Up a Plane Section

Create a plane section through the part with which to display the scalar and vector quantities while the solvers are running.

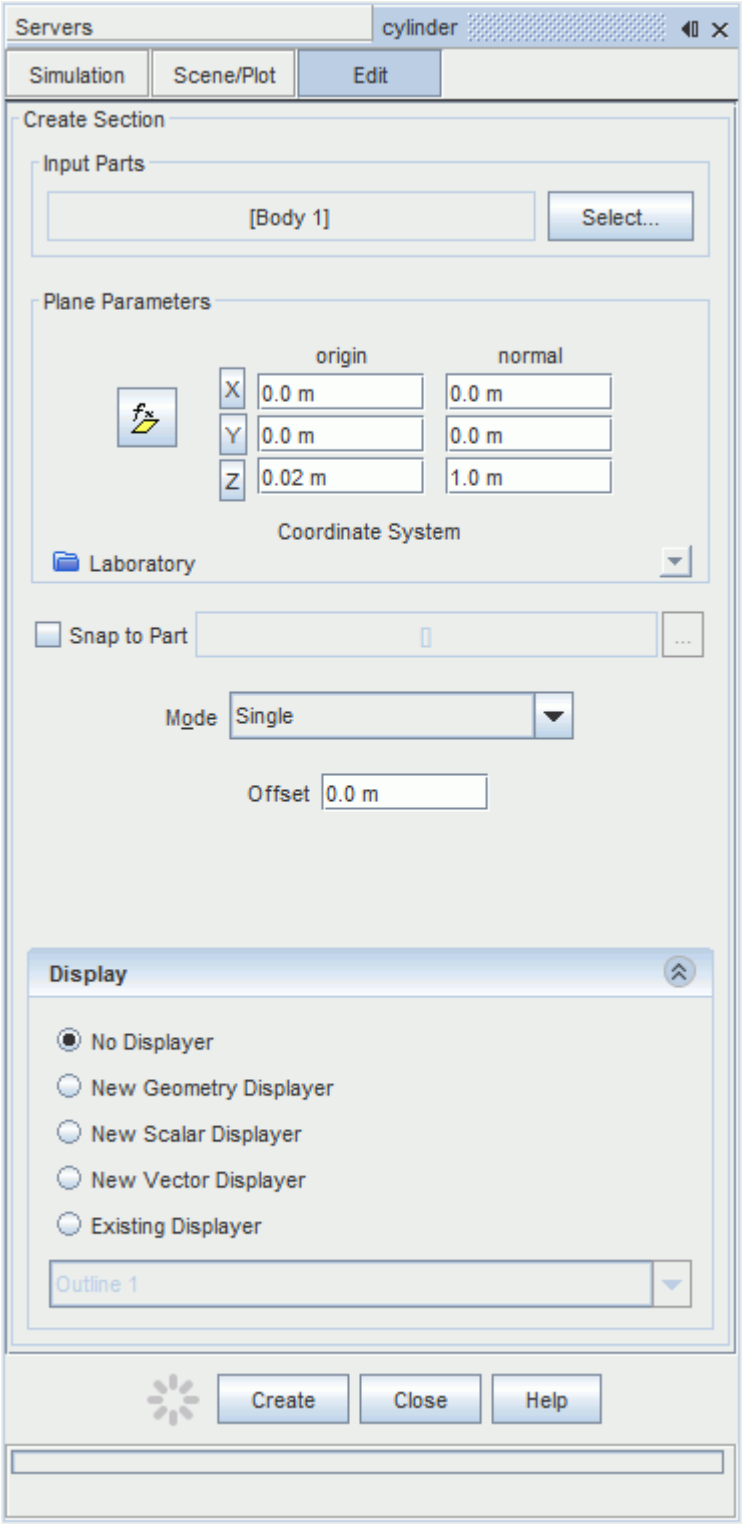
1. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane...**

   The *Create Section* dialog appears in the *cylinder* window.

2. Specify exact values to define the plane section:

Property	Value
Input Parts	**Body 1**
Plane Parameters	
⌊ *origin*	`[0, 0, 0.02]` m
⌊ *normal*	`[0, 0, 1]` m
Display	**No Displayer**

   The *Create Section* dialog now appears as shown below.



3. Click **Create**, then **Close**.

   A new derived part, **plane section**, is created under the **Derived Parts** node.

4. Save the simulation.

Setting Up Scalar Scenes

Two scalar scenes are created.

- Velocity Magnitude Contour
- Wall Y+ Contour

Contents:

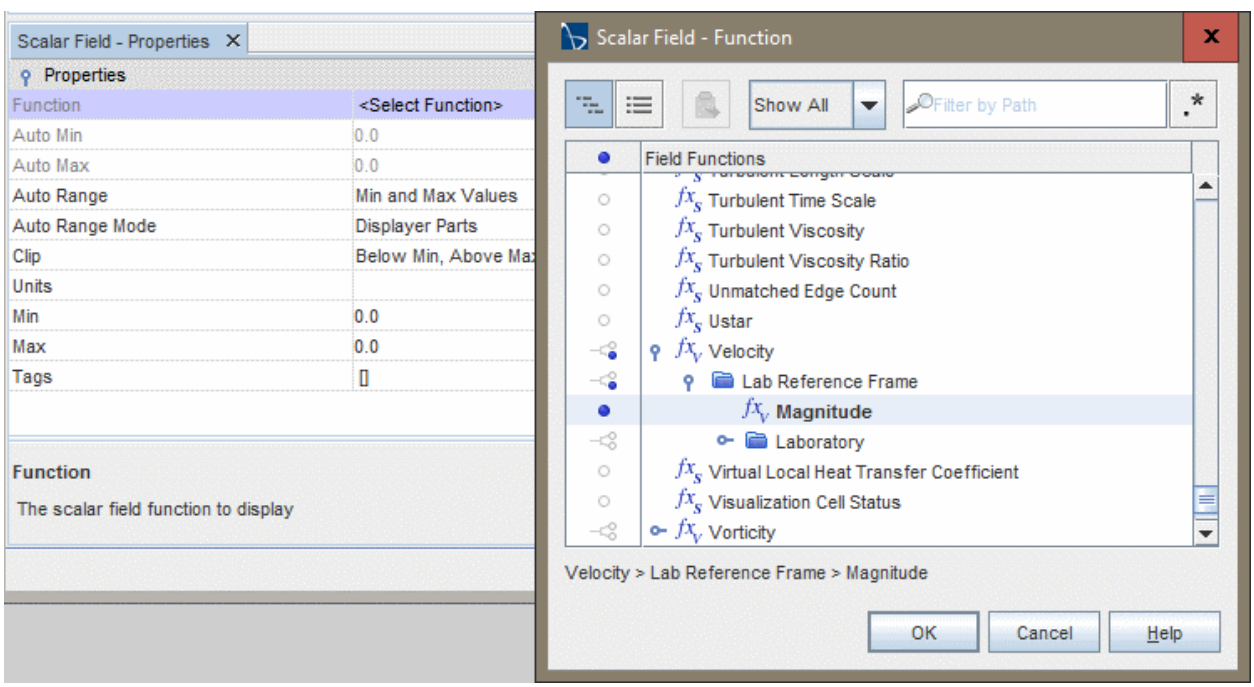
Velocity Magnitude Contour
Wall Y+ Contour

Velocity Magnitude Contour

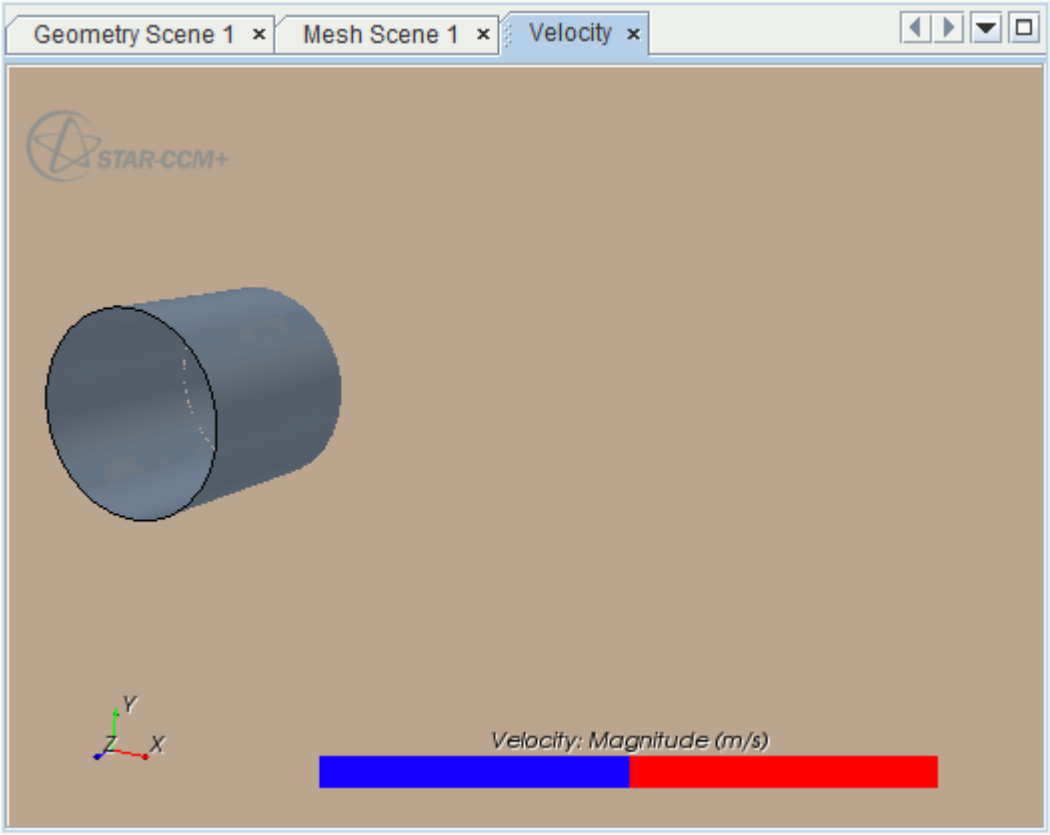
Create a scalar scene to display the velocity magnitude contour on the plane section.

Use the following steps:

1. Right-click the **Scenes** node and select **New Scene** > **Scalar**.
2. Rename the **Scalar Scene 1** node to `Velocity`.
3. Select the **Displayers** > **Outline 1** node and activate *Surface*.
4. Select the **Displayers** > **Outline 1** > **Parts** node and set *Parts* to **Regions** > **Body 1** > **Boundaries** > **Cylinderwall**.
5. Select the **Displayers** > **Scalar 1** node and set the *Contour Style* to `Smooth Filled`.
6. Select the **Displayers** > **Scalar 1** > **Parts** node and set *Parts* to **Derived Parts** > **plane section**.
7. Select the **Displayers** > **Scalar 1** > **Scalar Field** node and click to the right of *Function* and select **Velocity** > **Magnitude** from the list.



8. Set *Max* to `75`
9. Set *Clip* to `Off`.
10. Use the mouse buttons, as described previously, to adjust the viewing angle, zoom, and position the scene as shown below.

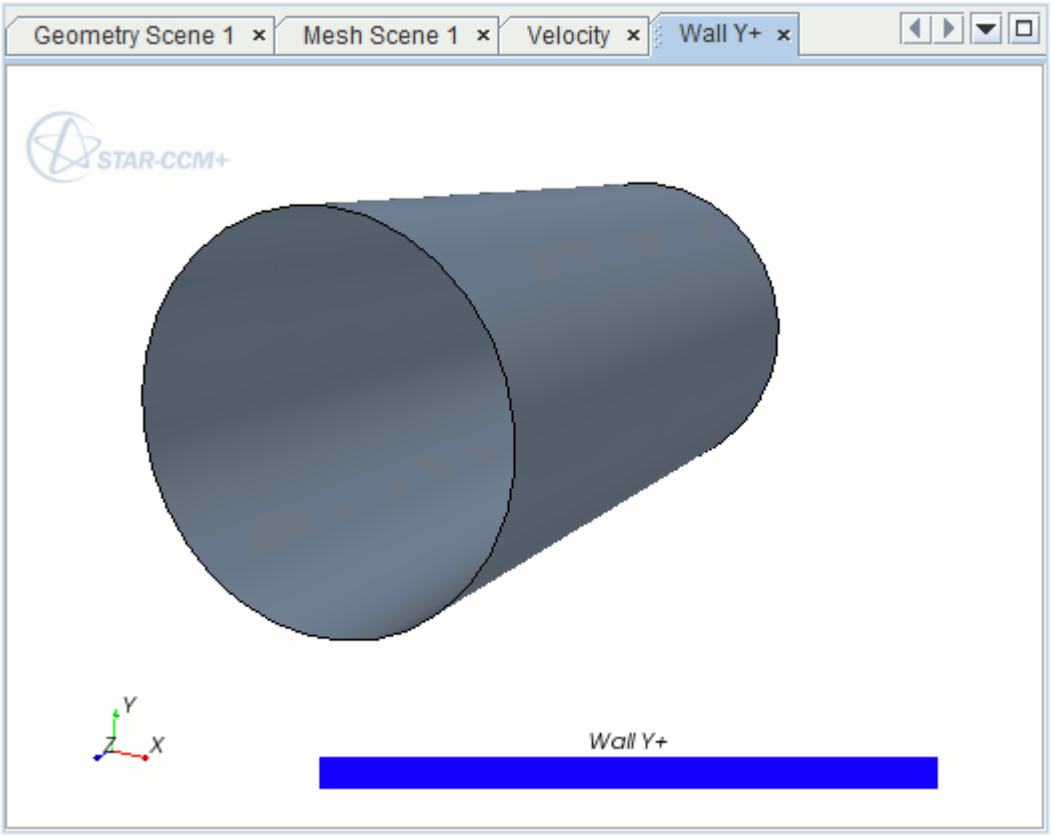


11. Store the current view. Click the (**Save-Restore-Select views**) button and select **Store Current View**.

Wall Y+ Contour

Create a scalar scene to display the Wall Y+ scalar field.

1. Create another scalar scene and rename it to **Wall Y+**.
2. Select the **Displayers** > **Outline 1** > **Parts** node and set *Parts* to **Regions** > **Body 1** > **Boundaries** > **Cylinderwall**.
3. Select the **Displayers** > **Scalar 1** > **Parts** node and set *Parts* to **Regions** > **Body 1** > **Boundaries** > **Cylinderwall**.
4. Select the **Displayers** > **Scalar 1** > **Scalar Field** node and set *Function* to **Wall Y+**.
5. Click the (**Save-Restore-Select views**) button, and select **Projection mode** > **Perspective**.
6. Use the mouse buttons, to zoom and position the scene as shown below.



Monitoring Force Coefficients

By default STAR-CCM+ provides a residuals plot which can be used to monitor the convergence of the solution. In addition to this, it is useful to create additional reports and monitors to help in assessing convergence.

Create two reports for monitoring force coefficients while the solvers are running:

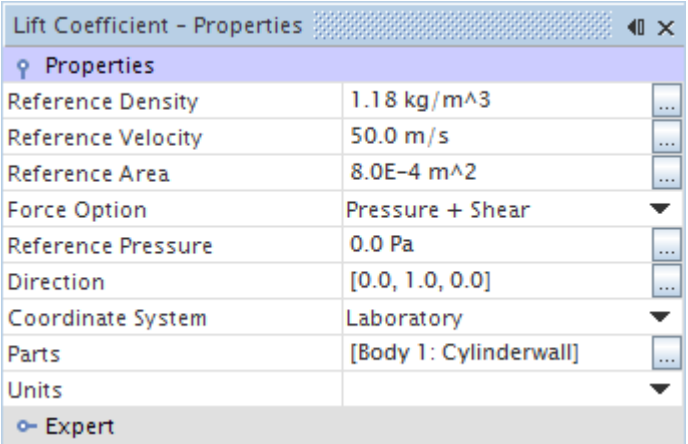
1. Right-click the **Reports** node and select **New Report** > **Force Coefficient**.
2. Rename the **Force Coefficient 1** node to `Drag Coefficient`.
3. In the *Properties* window, set the following properties:

Property	Setting
Reference Density	`1.18 kg/m^3`
Reference Velocity	`50 m/s`
Reference Area	`8E-4 m^2`
Parts	**Regions** > **Body 1** > **Boundaries** > **Cylinderwall**

This report can be copied and used as a basis for the lift coefficient report:

4. Copy and paste the **Drag Coefficient** node.
5. Rename the **Copy of Drag Coefficient** node to `Lift Coefficient`.

6. In the *Properties* window, set the *Direction* to `[0, 1, 0]`



Create a monitor and plot from the reports:

7. Multi-select the **Drag Coefficient** and **Lift Coefficient** nodes. Right-click the **Drag Coefficient** node and select **Create Monitor and Plot from Report.**
8. Click **Single Plot** to display the coefficients for both probes in a single graph.

Two new monitors have been created, with their associated plot.

9. Rename the **Plots** > **Reports Plot** to **Force Coefficients Plot**
10. Save the simulation.

Setting Stopping Criteria

Set the **Maximum Steps** as stopping criterion for this simulation.

Set the stopping criteria.

1. Edit the **Stopping Criteria** node and set the **Maximum Steps** to `200`.
2. Save the simulation.

Running the Simulation

Before running the simulation, reorder the mesh to optimize it for the solver.

1. Right-click **Regions** > **Body 1** and choose **Reorder Mesh**.
2. In the *Reorder Regions* dialog, activate **Use directional reordering**, then click **Apply**, then **Close**.

The simulation is now ready to be run.

3. Click (**Run**) in the *Solution* toolbar.

   The solution runs for 200 iterations. Observe that the residuals and other plots adopt an oscillatory pattern about a mean. This pattern is evidence that the steady state approach does not satisfy the real

physics of the problem; an unsteady approach is required. However, the purpose of this steady state solution is to qualify the mesh before committing resources to the full unsteady simulation.

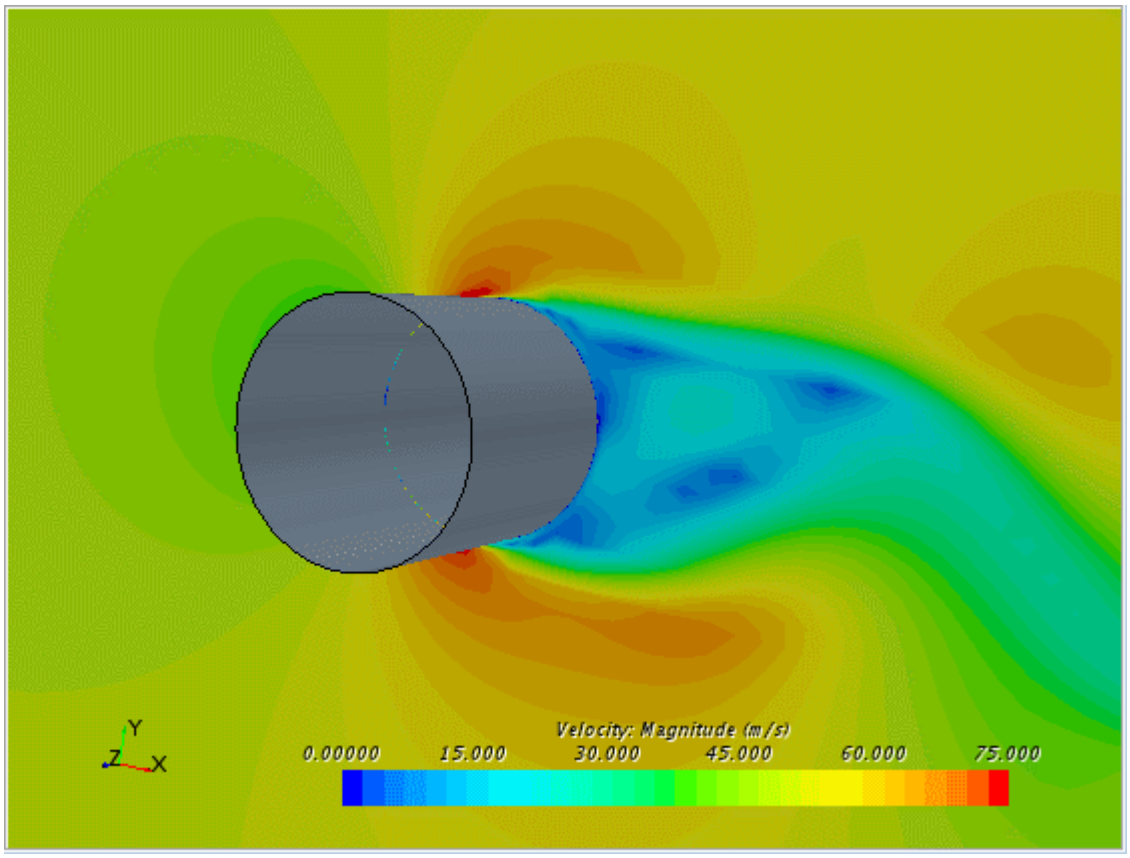
4. Save the simulation.

Viewing the Results

View the Velocity Magnitude Contour and Wall Y+ Contour scenes.

1. Click the *Velocity Magnitude Contour* tab.

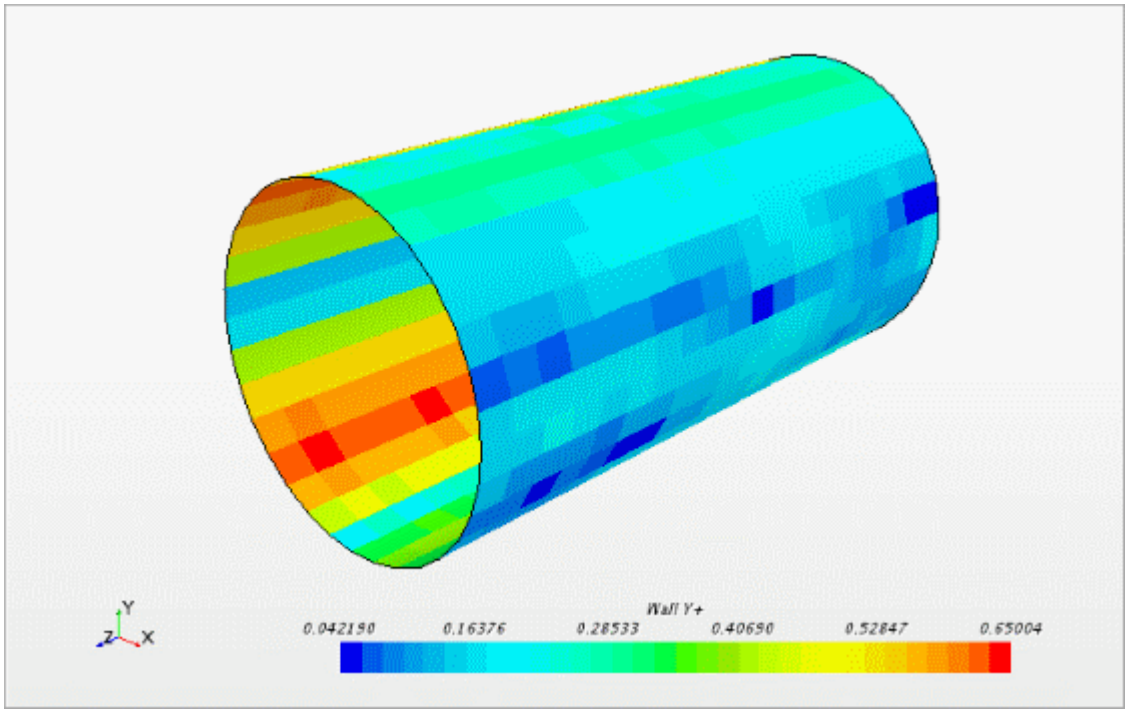
   A scene similar to the one shown below appears.



   The above plot illustrates the representative wake field.

2. Click the *Wall Y+ Contour* tab.

   A scene similar to the one shown below appears:



Wall Y+ values are everywhere below 1.

3. Close the *Wall Y+* scene in the *Graphics* window.

Contents:

Extracting Drag and Lift Coefficient RMS Values

Extracting Drag and Lift Coefficient RMS Values

Examine the root-mean-squares of drag and list coefficients.

1. Select **File** > **Macro** > **Play Macro**
2. In the *Open* dialog, navigate to your working directory and select `compareRmsDragLiftCoefficient.java`.
3. Click **Open** to run the macro.

The values appear in the *Output* window.

Applying the Broadband Noise Sources

The Broadband Noise Source models are intended to be used as tools for qualifying a volume mesh before a transient acoustic analysis. They are applied in the post-processing stage of a preliminary steady analysis.

Contents:

Selecting the Broadband Noise Sources
Visualizing the Curle Surface Acoustic Power dB Contour

Visualizing the Proudman Acoustic Power dB Contour
Visualizing the Mesh Frequency Cutoff Hz Contour

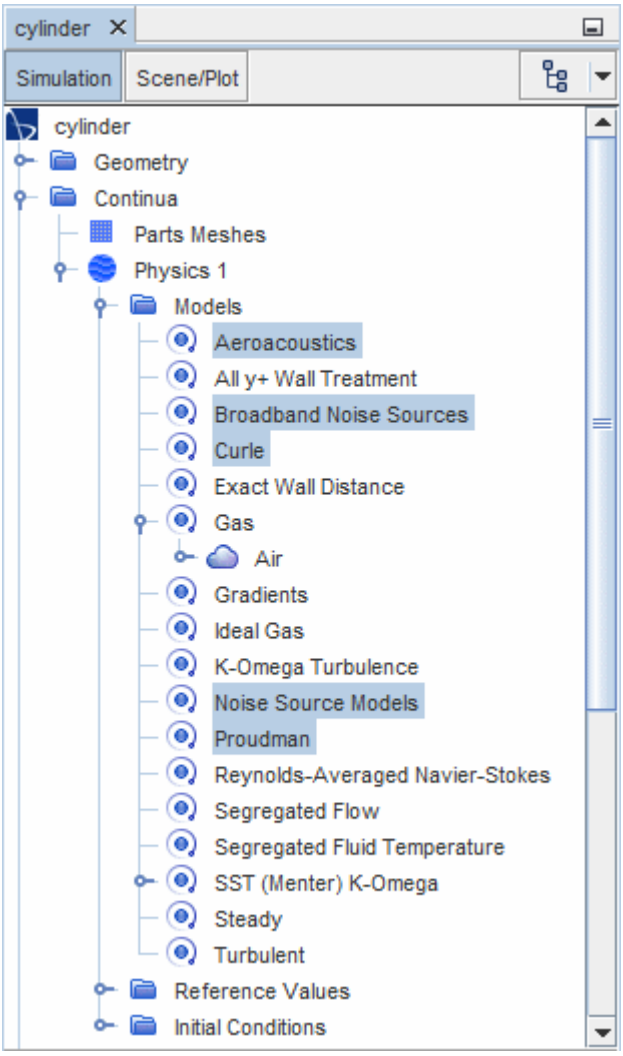
Selecting the Broadband Noise Sources

The broadband noise source models are activated from the model selection dialog in the **Continua** > **Physics 1** node.

- Open the **Continua** > **Physics 1** node and select the following models, in order:

Group Box	Model
Optional Models	**Aeroacoustics**
Aeroacoustics Models	**Broadband Noise Sources**
Broadband Noise Source Models	**Noise Source Models**
Noise Source Models	**Curle**
	Proudman

The selected models appear as shown below:

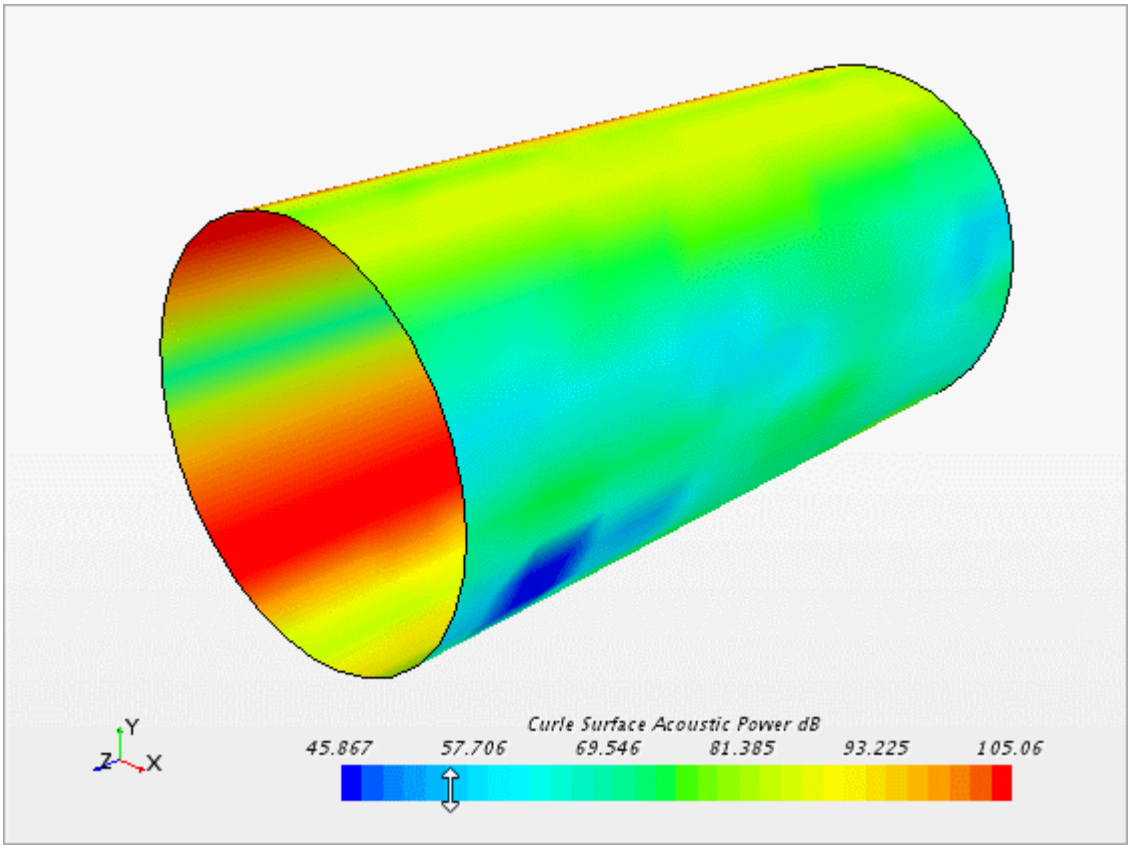


Visualizing the Curle Surface Acoustic Power dB Contour

Create a scalar scene to display the Curle surface acoustic power contour on the cylinder wall.

1. Copy and paste the **Scenes** > **Wall Y+** node.
2. Rename the new node to `Curle AC Power`.
3. Right-click the **Curle AC Power** node and choose **Open**.
4. Click the *scene/plot* tab.
5. Select the **Displayers** > **Scalar 1** > **Scalar Field** node and set *Function* to **Curle Surface Acoustic Power dB**.
6. Select the **Displayers** > **Scalar 1** node and set the *Contour Style* to **Smooth Blended**.
7. Click the (**Step**) button in the **Solution** toolbar.

A scene similar to the one shown below appears:



The Curle acoustic power model, which assumes an isotropic turbulent flow field, computes the sound that dipole sources generates. This model therefore represents the fluctuating surface pressure with which the solid boundary acts on the flow. The plot displays the noise level that the turbulent boundary layer emits over a surface at low Mach number.

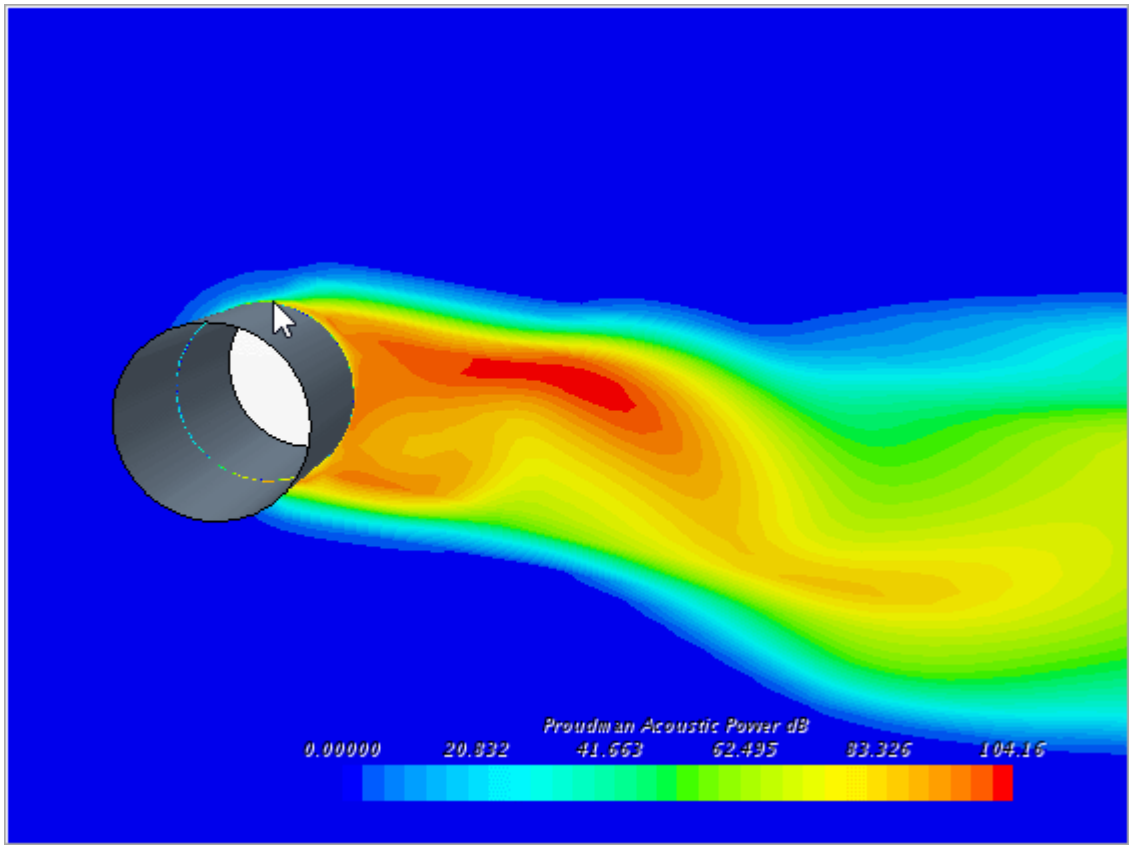
8. Close the *Curle AC Power* scene in the *Graphics* window.

Visualizing the Proudman Acoustic Power dB Contour

Create a scalar scene to display the Proudman surface acoustic power on the symmetry plane. This model indicates acoustic quadrupole sources.

1. Copy and paste the **Scenes** > **Velocity** node.
2. Rename the new node to `Proudman AC Power`.
3. Right-click the **Proudman AC Power** node and choose **Open**.
4. Select the **Displayers** > **Scalar 1** > **Scalar Field** node and set *Function* to **Proudman Acoustic Power dB**.

   A scene similar to the one shown below appears:



The Proudman acoustic power (in dB) estimates the local contribution of the quadrupole sources as generated by isotropic turbulence. It displays the acoustic power per unit volume as emitted by the turbulence structures, in the flow-field surrounding the cylinder.

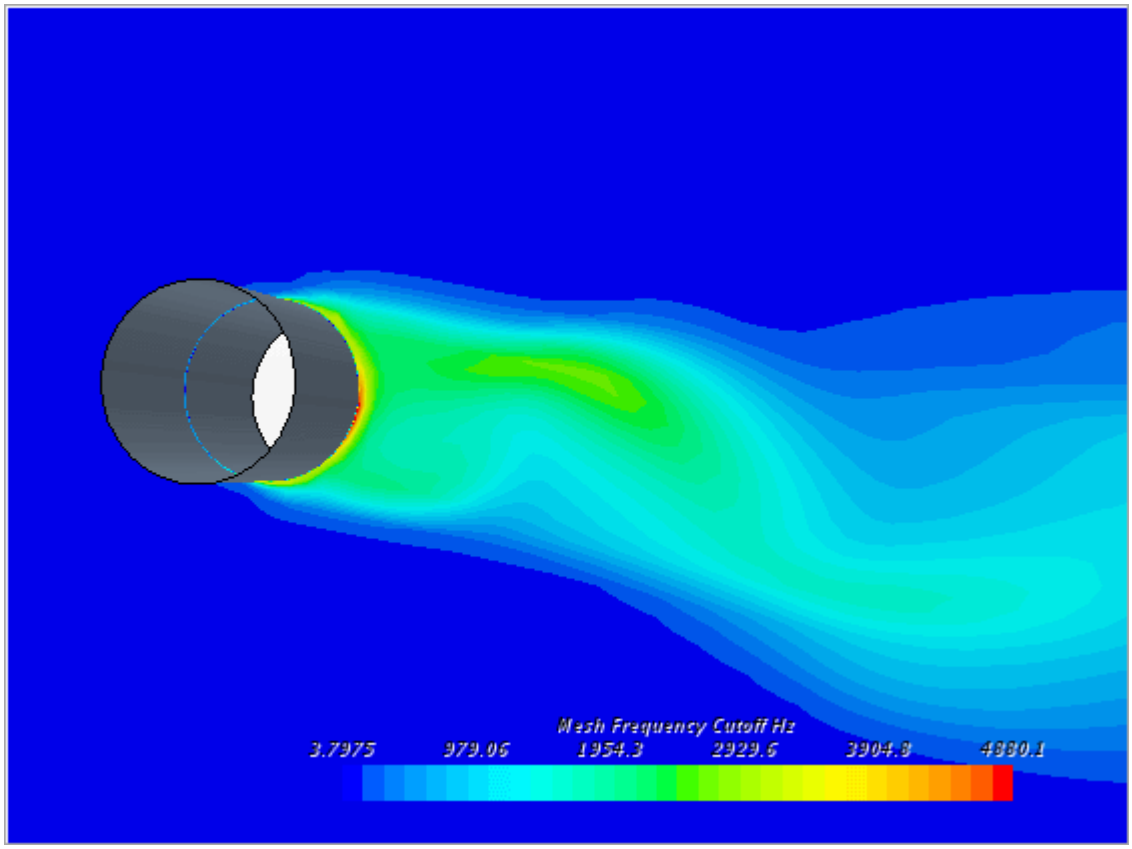
5. Close the *Proudman AC Power* scene in the *Graphics* window.

Visualizing the Mesh Frequency Cutoff Hz Contour

The mesh frequency cutoff field function becomes available when the Curle, Proudman, or Goldstein broadband noise source models are activated. In this case, Curle and Proudman have been selected. Create a scalar scene to display the mesh frequency cutoff contour on the symmetry plane.

1. Copy and paste the **Scenes** > **Proudman AC Power** node.
2. Rename the new node to `Mesh Frequency Cutoff`.
3. Right-click the **Mesh Frequency Cutoff** node and choose **Open**.
4. Select the **Displayers** > **Scalar 1** > **Scalar Field** node and set **Function** to **Mesh Frequency Cutoff Hz**.

   A scene similar to the one shown below appears:



This plot indicates that the mesh is able to resolve up to 2000 Hz, approximately, in the main volume source regions the Proudman correlation indicates. The mesh frequency cut-off estimate becomes less accurate in the boundary-extruded layer where the cells are non-isotropic.

5. Save the simulation.

Results Analysis and Consequences

The cylinder and cone refinements seem sufficient in terms of capturing the aeroacoustics sources behind the cylinder (as the Proudman source plot illustrates). Typically, however, the peak of the human hearing range is between 1-3 kHz. Therefore, refine the mesh further to increase the resolvable frequency.

Refining the Volume Mesh

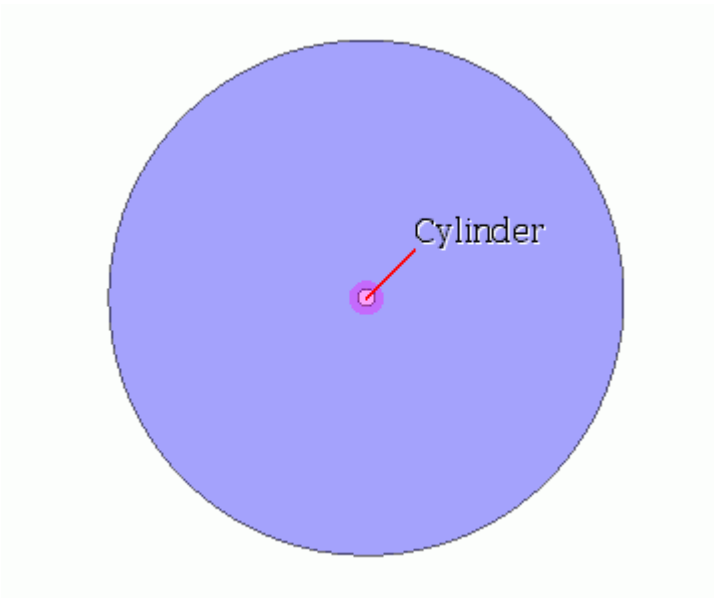
Continuing from the `cylinder.sim` file, you now create a cylindrical part, 0.02 m in radius, which defines the outer extent of the region to be refined.

1. Click the *Geometry Scene 1* tab in the *graphics* window.
2. Right-click the **Geometry** > **Parts** node and select **New Shape Part** > **Cylinder**.
3. In the *Create/Edit Cylinder Part* panel, set the following values:

Property	Setting
Start Coordinate	`[0, 0, 0] m`

Property	Setting
End Coordinate	`[0, 0, 0.041] m`
Cylinder Radius	`[0.02] m`

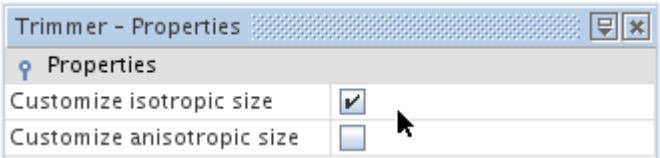
4. Cllick **Create** and **Close**.
5. Select the **Geometry** > **Parts** > **Cylinder** node, to display the cylinder part that you created in *Geometry Scene 1*.



You now create a volumetric control and select the shape that defines the source.

6. Expand the **Geometry** > **Operations** > **Automated Mesh** node.
7. Right-click the **Custom Controls** node and select **New** > **Volumetric Control**.

   A new node, **Volumetric Control 2**, is created in the simulation tree.
8. Select the **Volumetric Control 2** node and set *Parts* to **Cylinder**.
9. Expand the **Volumetric Controls 2** > **Controls** node and select **Trimmer**.
10. In the *Properties* window, enable *Customize isotropic size*.



   A new node,**Values**, is added to the simulation tree.
11. Select the **Volumetric Controls 2** > **Values** > **Custom Size** node and set *Size Type* to **Absolute**.
12. Set the *Value* to `8.0E-4 m`.
13. Click (**Generate Volume Mesh**) in the **Mesh Generation** toolbar.

The output from the meshers is shown in the *Output* window and the ongoing status is shown in the status bar. The solution obtained on the coarse mesh is interpolated onto the new mesh.

14. Save the simulation.

Setting New Stopping Criteria

Set a new value for **Maximum Steps**.

Use the following steps:

1. Open the **Stopping Criteria** node and select **Maximum Steps**.
2. In the *Properties* window, set the *Maximum Steps* to `400`.

Running the Simulation

The simulation is now ready to be run.

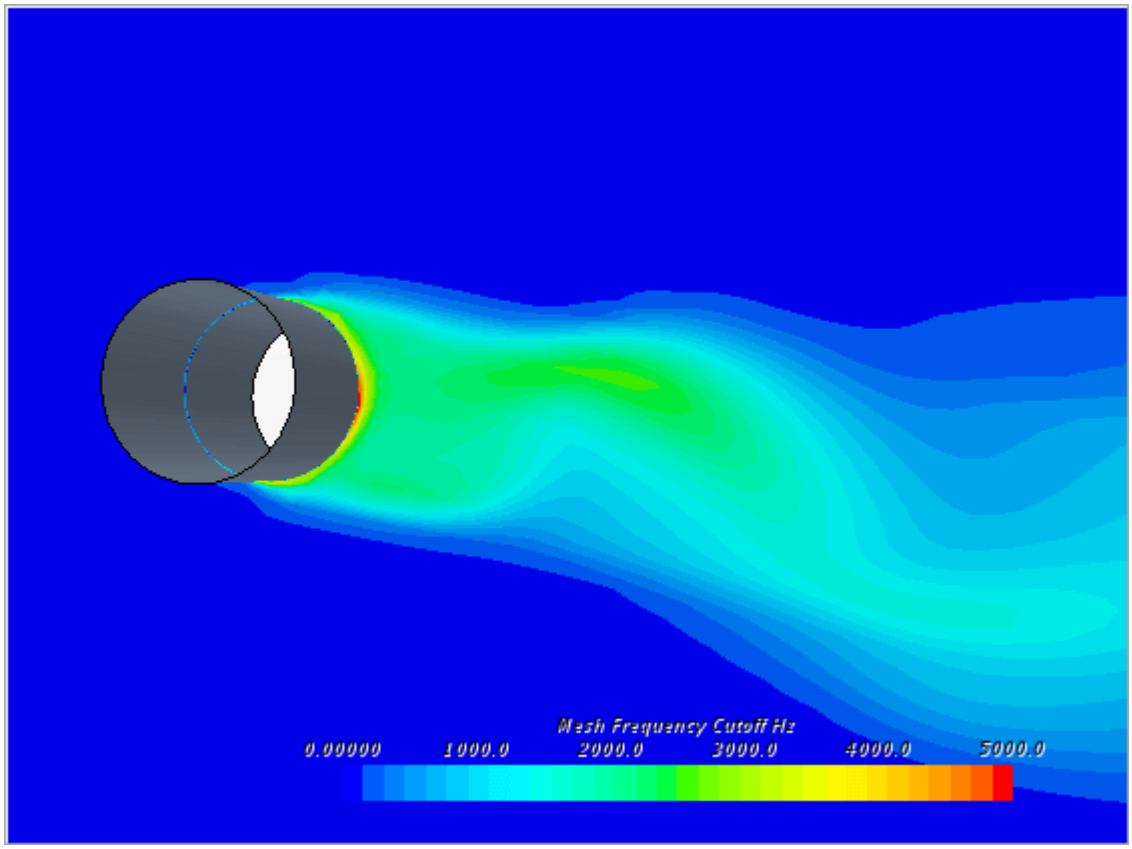
1. Run the simulation by clicking (**Run**) in the **Solution** toolbar.
2. Save the simulation on completion.

Viewing Results After Mesh Refinement

Visualize the updated *Mesh Frequency Cutoff* scene in the *Graphics* window.

1. Select the **Scenes** > **Mesh Frequency Cutoff** > **Displayers** > **Scalar 1** > **Scalar Field** node and set the *Min* to `0` and the *Max* to `5000`.

   A scene similar to the one shown below appears:



From the above plot, it is clear that the fine mesh adequately resolves higher frequencies close to the cylinder.

The steady-state analysis runs are now complete. Set up the final transient run in the next part of the tutorial.

2. Save the simulation.

Contents:

Extracting Drag and Lift Coefficient RMS Values

Extracting Drag and Lift Coefficient RMS Values

Examine the drag and list coefficients again, after the mesh has been refined.

1. Select **File** > **Macro** > **Play Macro**
2. In the *Open* dialog, navigate to your working directory and select `compareRmsDragLiftCoefficientRefinedMesh.java`.
3. Click **Open** to run the macro.

The values appear in the *Output* window.

Summary

This tutorial demonstrated how to set up aeroacoustics problems in STAR-CCM+, using the flow around a cylinder.

The tutorial introduced the following STAR-CCM+ features:

- Loading an existing base simulation.
- Visualizing the geometry.
- Generating the volume mesh.
- Selecting physics models.
- Setting up initial and boundary conditions.
- Setting up stopping criteria.
- Creating plane sectione and scenes to visualize the results.
- Applying the broadband noise sources.
- Running the solver for a set number of iterations.
- Refining the volume mesh and rerunning the solver.

DES and FW-H On-The-Fly: Noise from a Cylinder (Unsteady Analysis)

In the previous section of this tutorial, you completed the steady-state analysis runs for qualifying the volume mesh. In this section, continuing from the previous simulation file, you set up the final transient run for an unsteady analysis of the near field and an FW-H analysis of the far field.

Two point receivers are used for the far field analysis, each 1 m from the cylinder.

Note: Use the double precision version ofSTAR-CCM+ for all aeroacoustic and hydroacoustic analyses.

Note: Run this tutorial in parallel, with ten or more processors, to get a reasonable run time. This tutorial requires 4800 time-steps, with five inner iterations, on a test case with a large number of cells.

If you have closed the previous simulation file:

- Click (**Load Simulation**) from the tool bar and load `cylinder.sim`.

Contents:

Saving File for Transient Simulation
Selecting the Transient Calculation Physics Models
Setting the Solver Properties and Stopping Criteria
Visualizing the Mean Velocity
Monitoring Pressure
Monitoring Force Coefficients
Saving Transient Surface Data
Preparing the On-The-Fly FW-H Analysis
Visualizing the Solution
Running the Simulation
Visualizing the Results
Spectral Analysis
Results Analysis
Summary
DES and FW-H On-The-Fly Tutorial Bibliography

Saving File for Transient Simulation

The steady-state analysis runs are now complete. Set up the final transient run.

- Save a copy of the simulation file under the name `cylinderUnsteady.sim` for the transient calculation to follow.

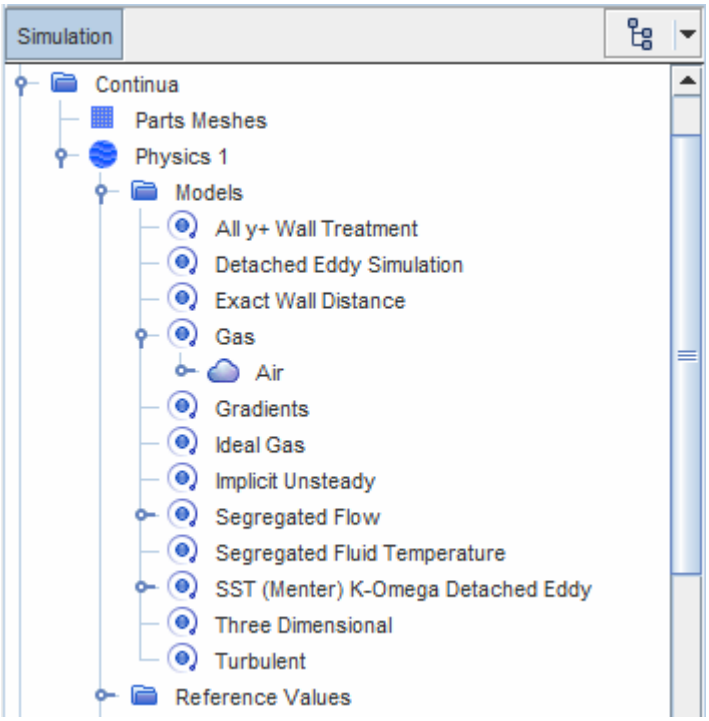
Selecting the Transient Calculation Physics Models

Modify the physics model selections to activate the unsteady solver.

1. For the physics continuum, **Continua** > **Physics 1**, deselect the following models from the *Enabled Models* list:
   - **Proudman**
   - **Noise Source Models**
   - **Curle**
   - **Broadband Noise Sources**
   - **Aeroacoustics**
   - **All y+ Wall Treatment**
   - **SST (Menter) K-Omega**
   - **K-Omega Turbulence**
   - **Reynolds-Averaged Navier-Stokes**
   - **Turbulent**
   - **Steady**
2. Deactivate the **Auto-select recommended models** option.
3. Select the following models, in order:

Group Box	Model
Time	**Implicit Unsteady**
Viscous Regime	**Turbulent**
Turbulence	**Detached Eddy Simulation**
Detached Eddy Simulation	**SST (Menter) K-Omega Detached Eddy**
K-Omega Wall Treatment	**All y+ Wall Treatment**

The physics models appear as below:



4. Open the **Continua** > **Physics 1** > **Models** node, select **Segregated Flow** and set *Convection* to **Hybrid**.
5. Open the **Continua** > **Physics 1** > **Models** node, select **SST (Menter) K-Omega Detached Eddy** and set the *Formulation Option* property to **DDES**.

Setting the Solver Properties and Stopping Criteria

Choosing the correct time-step is an important step in setting up an aeroacoustic simulation.

You typically evaluate the following criteria, and choose the smallest value:

- Convective Courant Number: Implicit solvers are usually stable at maximum values in the range 10-100 locally, but the overall mean value should be around 1. For this case, a time-step of 2.5E-5 s gives a Convective Courant Number whose mean value is approximately 1.
- Maximum resolvable frequency: For example, to obtain 10 points in a wave at 3000 Hz, (around the point at which human hearing is most sensitive), the time-step size should be 10 x 3000 Hz = 30000 Hz, or 3.33E-5 s.
- Local Strouhal shedding: At the Reynolds number observed in this tutorial, 63800, a cylinder sheds vortices at a Strouhal number of approximately 0.22. This number corresponds to a shedding frequency of 500 Hz. Capturing this shedding frequency requires a time-step size of 2.0E-4 s.

The smallest time-step size is the one required by the Convective Courant Number criterion, 2.5E-5 s.

To set the time-step size:

1. Select the **Solvers** > **Implicit Unsteady** node and set *Time-Step* to `2.5E-5 s`.
2. Set *Temporal-Discretization* to `2nd-order`.

Now set suitable under-relaxation factors:

3. Select the **Solvers** > **Segregated Flow** > **Velocity** node and set *Under-Relaxation Factor* to `0.9`.
4. Select the **Segregated Flow** > **Pressure** node and set *Under-Relaxation Factor* to `0.7`.

The case is run until 0.12 s and with 5 iterations per time-step:

5. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Inner Iterations	*Maximum Inner Iterations*	`5`
Maximum Physical Time	*Maximum Physical Time*	`0.12` s
Maximum Steps	*Enabled*	Deactivated

As the simulation is unsteady, update the trigger on the monitors for the lift and drag coefficients:

6. Within the **Monitors** node, multi-select the **Drag Coefficient Monitor** and **Lift Coefficient Monitor** nodes, and set *Trigger* to **Time Step**.
7. Save the simulation.

Visualizing the Mean Velocity

Create a monitor for displaying the mean velocity magnitude in a scalar scene.

1. Right-click the **Monitors** node and select **New Monitor** > **Field Mean**.

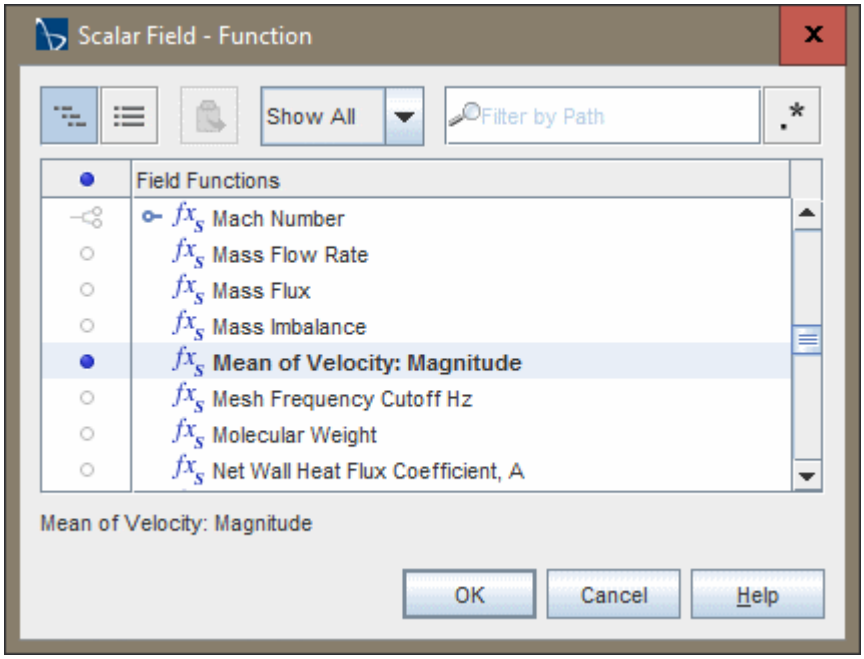
   Field Mean 1 is now added to the simulation tree.
2. Select the **Field Mean 1** node and set the following properties:

Property	Setting
Field Function	**Velocity Magnitude**
Parts	**Regions** > **Body 1**

3. Select the **Field Mean 1** > **Time-Step Frequency** node and set *Start Time-step* to `800`.

Create a scalar scene to display the mean velocity field:

4. Copy and paste the **Scenes** > **Velocity** node and paste it into the **Scenes** node.
5. Rename the **Copy of Velocity** node to `Mean Velocity Magnitude`.
6. Right-click the **Mean Velocity Magnitude** node and choose **Open**.
7. Select the **Displayers** > **Scalar 1** > **Scalar Field** node and set *Function* to **Mean of Velocity: Magnitude**.



Monitoring Pressure

Create three probes that are used to monitor pressure while the solvers are running.

Use the following steps:

1. Right-click the **Scenes** > **Geometry Scene 1** node and select **Open**.
2. In the simulation tree, right-click the **Derived Parts** node and select **New Part** > **Probe** > **Point**.
3. In the *Create Point* panel, set the following properties:

Group Box	Setting
Input Parts	**Body 1**
Point	`[0.0101, 0, 1.0E-4]` m, m, m
Display	**No Displayer**

   A new derived part is created, called **point**.
4. Rename the **Derived Parts** > **point** node to `Probe-back`.

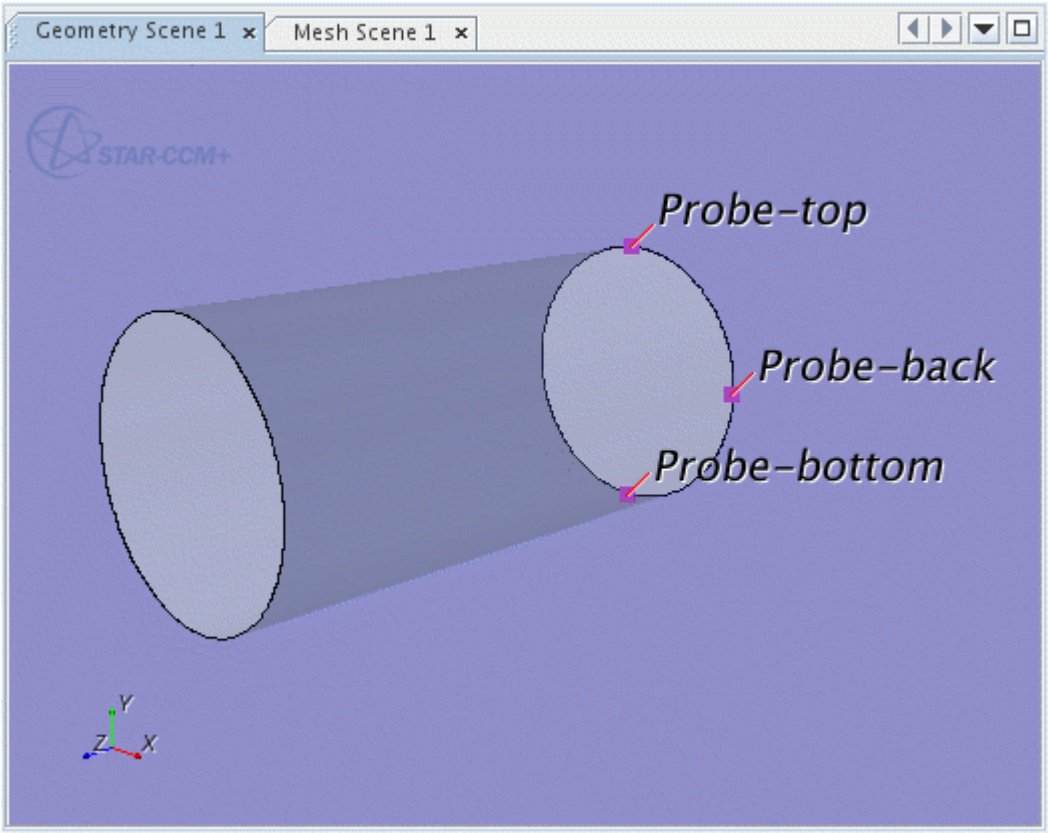
Create two more probes by copying and pasting:

5. Copy the **Probe-back** entry and paste it into the **Derived Parts** node.
6. Rename the **Derived Parts** > **Copy of Probe-back** node to `Probe-bottom`.
7. In the *Properties* window, set *Point* to [`0,-0.0101,1.0E-4`] m, m, m.
8. Copy the **Probe-back** node for a second time and paste it into the **Derived Parts** node.
9. Rename the **Derived Parts** > **Copy of Probe-back** node to `Probe-top`.
10. In the *Properties* window, set *Point* to [`0,0.0101,1.0E-4`] m, m, m.

Display the probes in the *Graphics* window:

11. Multi-select the **Probe-back**, **Probe-bottom**, and **Probe-top** nodes.

12. Click **(Make Scene Transparent)** and, using the mouse buttons, zoom and position the scene as shown below.



Create three reports, one for each probe:

13. Right-click the **Reports** node and select **New Report** > **Maximum**.
14. Rename the **Maximum 1** node to `Max Pressure-back`.
15. Select the **Max Pressure-back** node and set the following properties:

Property	Value
Scalar Field Function	**Pressure**
Parts	**Derived Parts** > **Probe-back**

16. Copy the **Max Pressure-back** node and paste it into the **Reports** node.
17. Rename the **Copy of Max Pressure-back**node to `Max Pressure-bottom`.
18. Select the **Max Pressure-bottom** node and set *Parts* to **Derived Parts** > **Probe-bottom**.
19. Create another copy of **Max Pressure-back.**
20. Rename the new node to `Max Pressure-top`.
21. Select the **Max Pressure-top** node and set *Parts* to **Derived Parts** > **Probe-top**.

Create a monitor and plot from the reports:

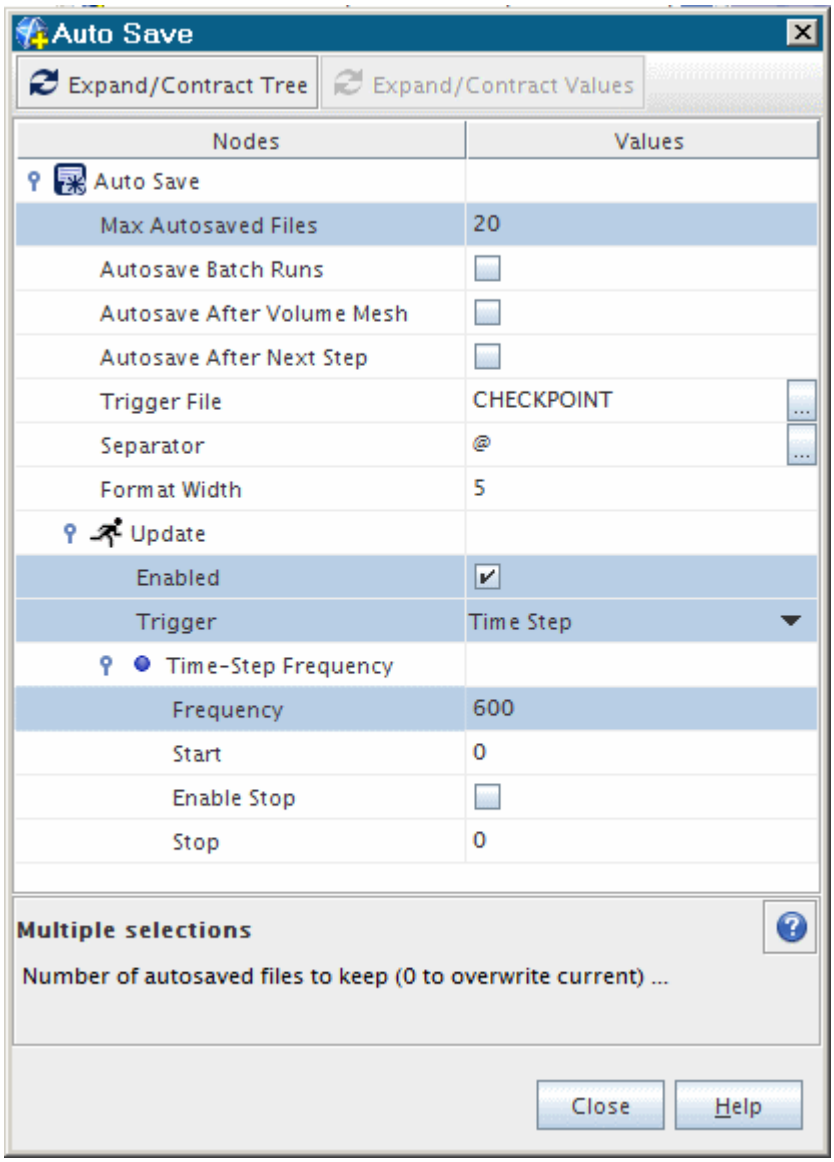
22. Multi-select the **Max Pressure-back**, **Max Pressure-bottom**, and **Max Pressure-top** nodes. Right-click the **Max Pressure-back** node and select **Create Monitor and Plot from Report**
23. Click **Single Plot** to display all three values in a single graph.

STAR-CCM+ creates three new monitors and the associated plot.

24. Rename the **Plots** > **Reports Plot** node to `Pressure Probes Plot`.

Change the auto save criteria to be based on time-step rather than iterations. Also specify the simulation to be saved every six hundred time steps.

25. Select **File** > **Auto Save...** .
26. In the *Auto Save* dialog, set *Max Autosaved Files* to `20`.
27. Under *Update*, activate the **Enabled** checkbox and select `Time Step` for the *Trigger* property.
28. Under *Time-Step Frequency*, set the *Frequency* to `600`.



29. Click **Close**.
30. Save the simulation.

Monitoring Force Coefficients

By default STAR-CCM+ provides a residuals plot which can be used to monitor the convergence of the solution.

Two monitor reports were created in the Broadband Models tutorial, for Drag Coefficient and Life Coefficient. Adapt the Force Coefficient Plot:

1. Select **Plots** > **Force Coefficient Plot** and set the *X-Axis Monitor* property to **Physical Time**.
2. Save the simulation.

Saving Transient Surface Data

Assuming that you have sufficient disk space (1.4 Gb), set up STAR-CCM+ to export volume or surface data at regular intervals throughout the transient simulation.

In this tutorial, you export pressure data on selected surfaces after each time-step, so that you can perform surface spectral analysis when the simulation completes.

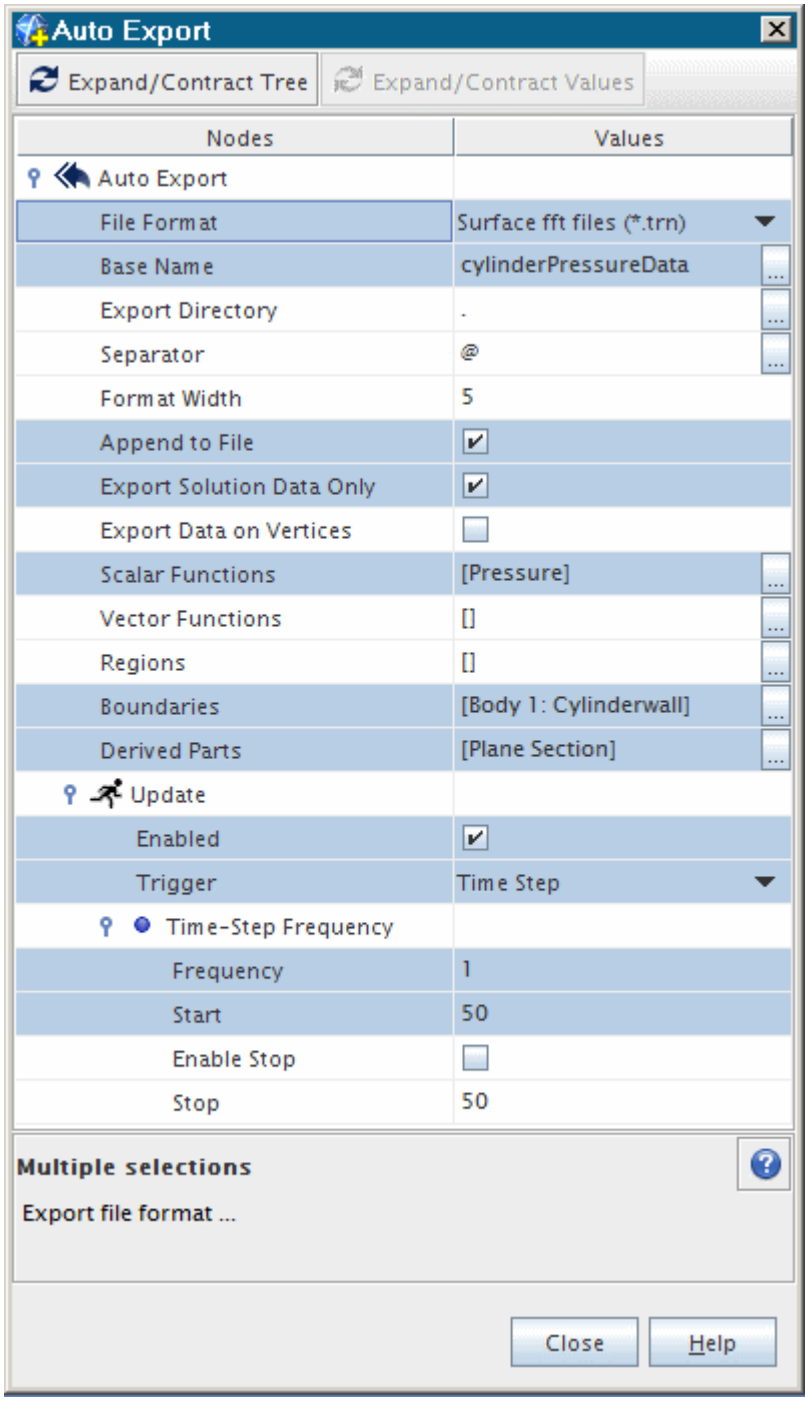
1. Select **File** > **Auto Export...**

   The *Auto Export* dialog appears.
2. Set the following values:

Nodes	Values
File Format	**Surface fft files (*.trn)**
Base Name	**cylinderPressureData**
Append to File	Activated
Export Solution Data Only	Activated
Scalar Functions	**Pressure**
Boundaries	**Body 1: Cylinderwall**
Derived Parts	**plane section**
Update	
⌊ *Enabled*	Activated
⌊ *Trigger*	**Time Step**
Time-Step Frequency	
⌊ *Frequency*	1

Nodes	Values
∟ *Start*	50

The *Auto Export* dialog looks as shown below.



3. Click **Close**.
4. Save the simulation.

Note: The Auto Export function can also be used to export data for use in third-party aeroacoustics propagation codes. When this is required, choose the **CD-adapco CCM Files (*.ccm)** option for the *File Format*, as some third-party software (Actran, VA One, and Virtual.Lab.Acoustics) are able to read the `.ccm` format directly. Follow the advice from these packages as to the data required. For volume data, select appropriate regions using the *Regions* property; for surface data, select appropriate boundaries or derived parts in the manner that is shown above.

Preparing the On-The-Fly FW-H Analysis

The two key steps in preparing STAR-CCM+ for a Ffowcs Williams-Hawkings analysis are to define the FW-H Surfaces and then define the FW-H Receivers. FW-H Receivers are points at specific distances from the noise sources.

To choose the FW-H models:

1. For the physics continuum, **Continua** > **Physics 1**, select the following models, in order:

Group Box	Model
Optional Models	**Aeroacoustics**
Aeroacoustics Models	**Ffowcs Williams-Hawkings Unsteady**
FW-H Unsteady Models	**On-The-Fly FW-H**

2. Select the **FW-H Receivers** node and set the following properties:

Property	Setting
Far-Field Density	`1.18 kg/m^3`
Far-Field Speed of Sound	`347.2 m/s`

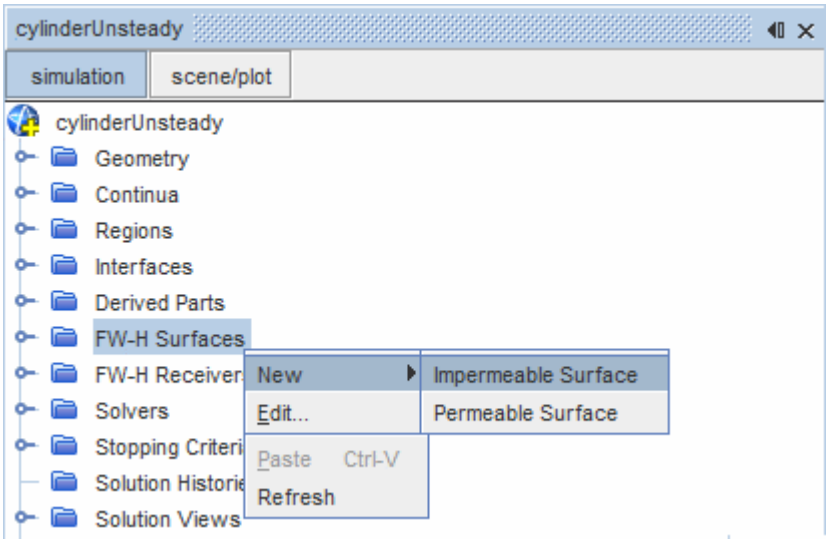
Contents:

Setting the FW-H Surfaces
Setting the FW-H Receivers
Checking FW-H Solver Properties

Setting the FW-H Surfaces

Two additional nodes were added to the simulation tree when the Ffowcs Williams-Hawkings model was selected. Set the FW-H Surfaces.

1. Right-click the **FW-H Surfaces** node and select **New** > **Impermeable Surface**.

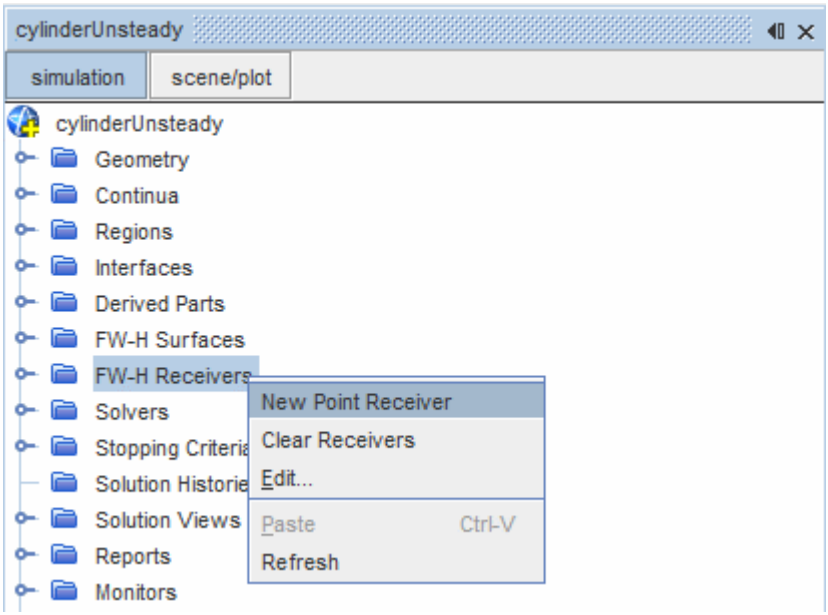


2. Select the **Impermeable Surface 1** node and set *Boundaries* to **Body 1:Cylinderwall**.

Setting the FW-H Receivers

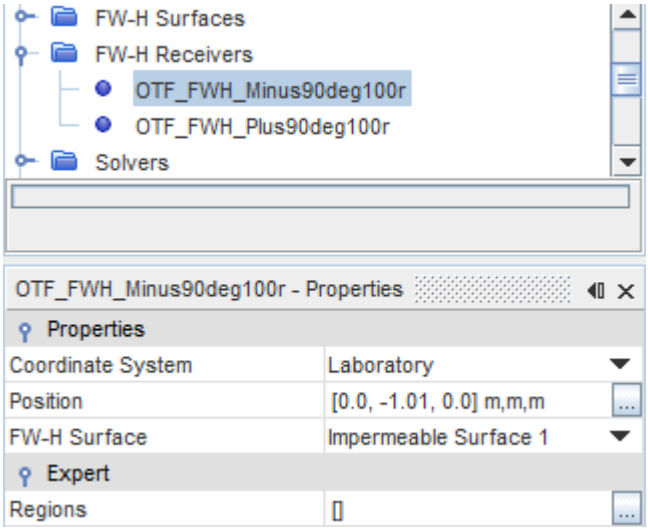
Set the point-receivers.

1. Right-click the **FW-H Receivers** node and select **New Point Receiver**.



2. Select **Point Receiver** and set *FW-H Surface* to **Impermeable Surface 1**.
3. Set *Position* to `[0.0, 1.01, 0.0] m,m,m`.
4. Rename the node to `OTF_FWH_Plus90deg100r`.
5. Similarly, create a second point receiver node.
6. Select the newly created node and set *FW-H Surface* to **Impermeable Surface 1**.

7. Set *Position* to `[0.0, -1.01, 0.0] m,m,m`.
8. Rename the node to `OTF_FWH_Minus90deg100r`.



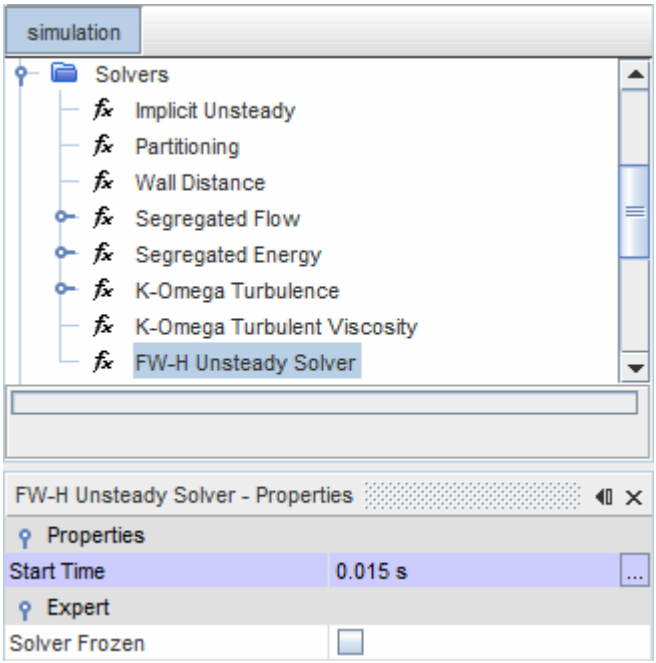
Checking FW-H Solver Properties

The **FW-H Solver** node has a *Start Time* property that defines the physical time when point-receivers start computing the sound pressure.

To set the start time:

- Select the **FW-H Unsteady Solver** node and set *Start Time* to `0.015 s`.

This setting is based on observed statistically steady flow of the 3D simulation, in which the mean level remains constant.



Visualizing the Solution

Add scenes for the noise emitted from the cylinder surface and received at each receiver.

The noise has a loading component for the unsteady motion of the force distribution. The body is stationary which means the thickness component is zero. The total surface noise is simply the summation of the two components.

To create the first total noise contour scene:

1. Create a scalar scene.
2. Rename the Scene to `TotalSurfaceTerm_OTF_FWH_Plus90deg100r`.
3. Select the **Displayers** node and set the following properties:

Node	Property	Setting
Scalar 1	*Contour Style*	**Smooth Filled**
⌊ **Scalar Field**	*Function*	**FW-H Noise Source** > **OTF_FWH_Plus90deg100r** > **Total Surface Term**
⌊ **Parts**	*Parts*	**Regions** > **Body 1** > **Boundaries** > **Cylinderwall**
Outline 1 > **Parts**	*Parts*	**Regions** > **Body 1** > **Boundaries** > **Cylinderwall**

4. In the main toolbar, click **Reset View** .

Only the cylinder is shown.

Add a second scalar scene for the other receiver similar to the first.

5. Copy the **Scenes** > **TotalSurfaceTerm_OTF_FWH_Plus90deg100r** node and paste it to the **Scenes** node.
6. Rename the **Copy of TotalSurfaceTerm_OTF_FWH_Plus90deg100r** to `TotalSurfaceTerm_OTF_FWH_Minus90deg100r`.
7. Select the **Scalar 1** > **Scalar Field** node and set *Function* to **FW-H Noise Source** > **OTF_FWH_Minus90deg100r** > **Total Surface Term**.
8. Save the simulation.

Running the Simulation

The simulation is now ready to be run. It is best that you run this simulation in parallel using several CPUs, as otherwise the run time is considerable.

1. Run the simulation by clicking the (**Run**) button in the **Solution** toolbar.

The solution is saved every six hundred time-steps. Save the final solution file (cylinderUnsteady@04800.sim) under the original file name.

2. Select **File** > **Save As**
3. In the *Save* dialog, select cylinderUnsteady.sim for the File Name.
4. Click **Save**.
5. Click **OK** to overwrite the previous file.

Contents:

Force Coefficients Time Plot
Maximum Pressure Time Plot
Mean Velocity Magnitude

Force Coefficients Time Plot

Modify the axis range of the force coefficients plot.

Use the following steps:

1. Right-click the **Plots** > **Force Coefficients Plot** node and select **Open**.
2. Select the **Plots** > **Force Coefficients Plot** > **Axes** > **Bottom Axis** node and set *Minimum* to `0.0` and *Maximum* to `0.12`.
3. Select the **Bottom Axis** > **Major Labels** node and set *Label Spacing* to `0.02`.
4. Select the **Bottom Axis** > **Title** node and set *Title* to `Time (s)`.
5. Select the **Left Axis** > **Title** node and set *Title* to `Force Coefficient`.

A scene similar to the one shown below is displayed.

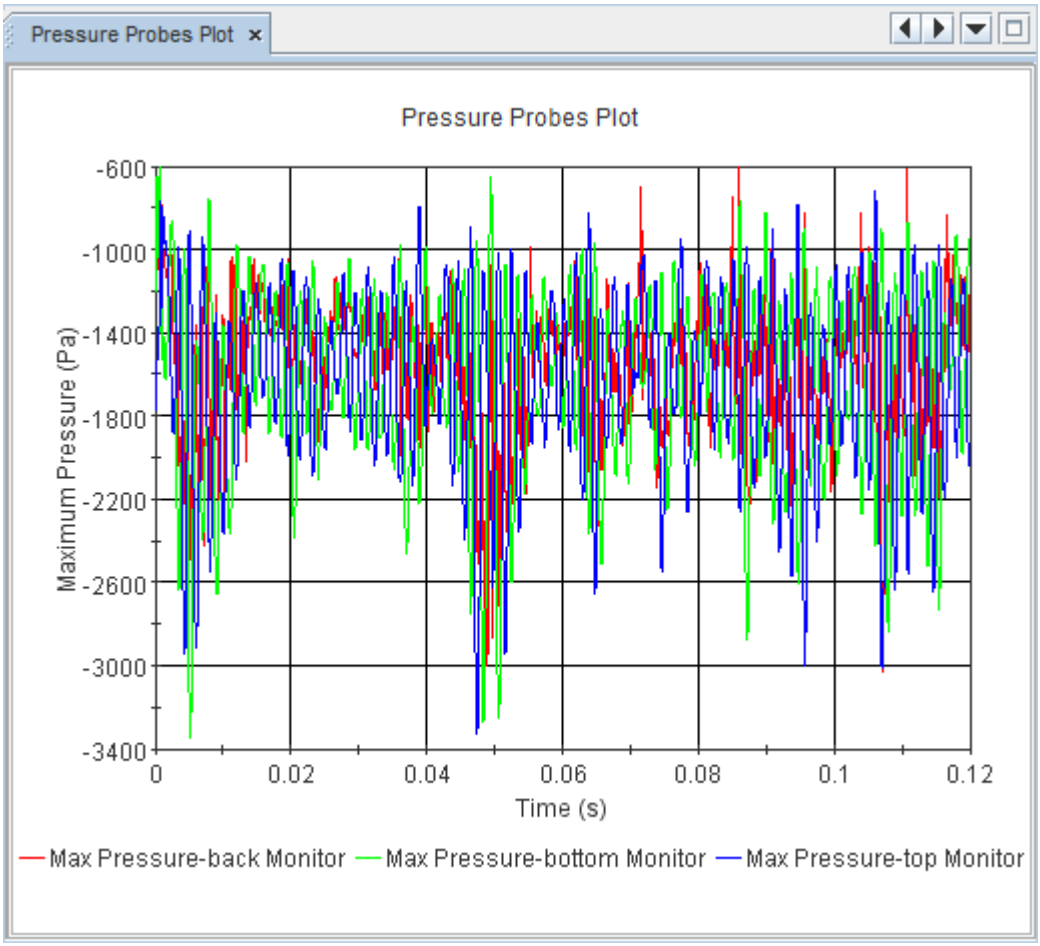
As expected, the mean lift is around zero and the mean drag is around 1 [452].

Maximum Pressure Time Plot

Create a plot of maximum pressure versus time.

1. Right-click the **Plots** > **Pressure Probes Plot** node and select **Open**.
2. Open the **Plots** > **Pressure Probes Plot** > **Axes** > **Bottom Axis** node and set *Minimum* to `0.0` and *Maximum* to `0.12`.
3. Select the **Bottom Axis** > **Major Labels** and set *Label Spacing* to `0.02`.
4. Select the **Bottom Axis** > **Title** node and set *Title* to `Time (s)`.
5. Select the **Left Axis** > **Title** node and set *Title* to `Maximum Pressure (Pa)`.

The Graphics window contains a scene similar to the following one.

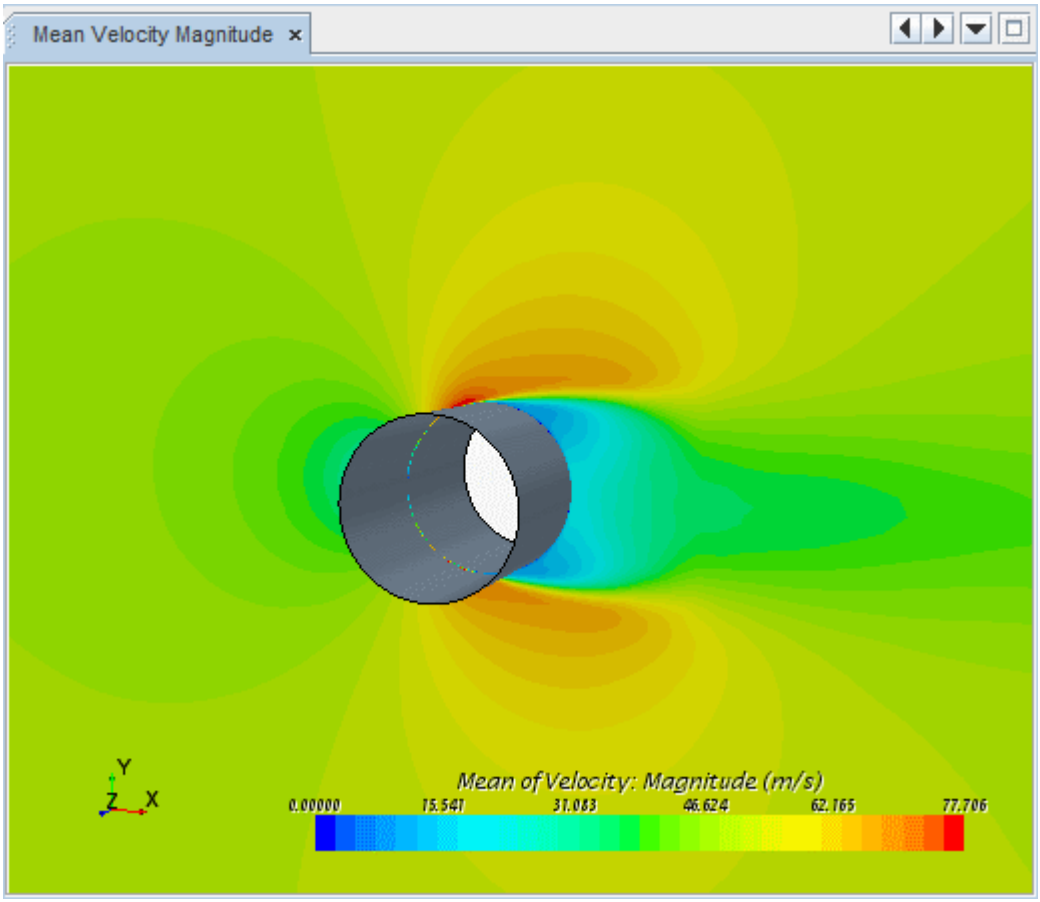


The mean pressure level remains constant while the top and bottom pressure probes identify anti-symmetric fluctuations.

Mean Velocity Magnitude

View the mean velocity magnitude in a scalar scene.

Examine the *Mean Velocity Magnitude* scene.

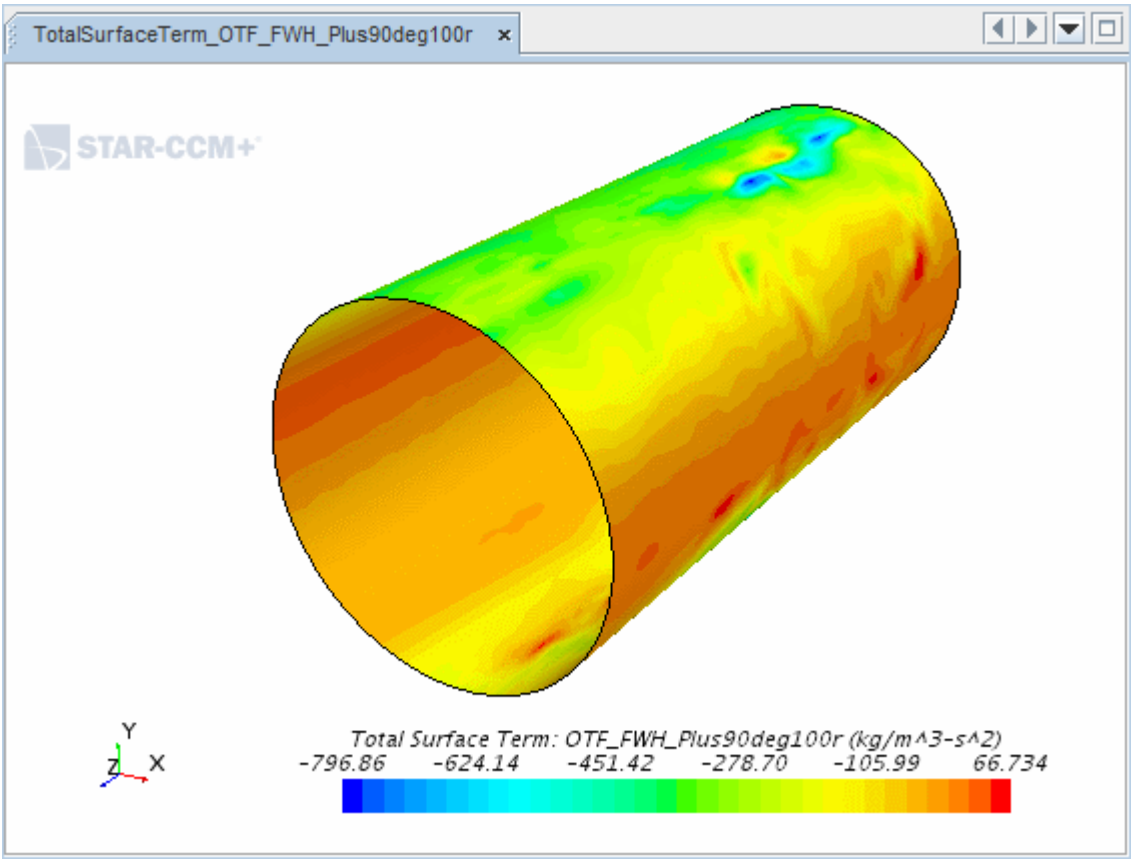


The mean velocity magnitude is symmetric around the cylinder. The difference in the wake is due to the refinement level.

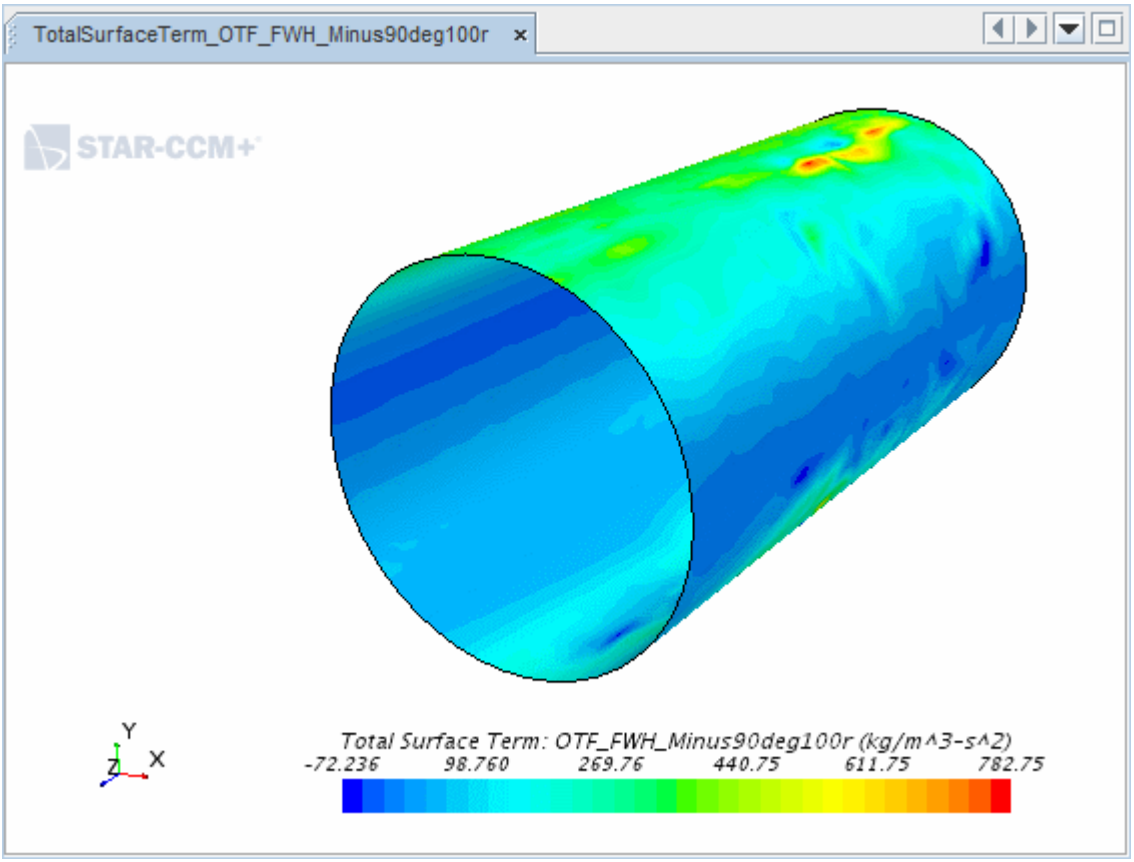
Visualizing the Results

You can now visualize the acoustic pressure on the cylinder, based on input from the two point-receivers.

1. Open the node **Scenes** > **TotalSurfaceTerm_OTF_FWH_Plus90deg100r**. A scene similar to the one shown below is displayed.



2. Open the node **Scenes** > **TotalSurfaceTerm_OTF_FWH_Minus90deg100r**. A scene similar to the one shown below is displayed.

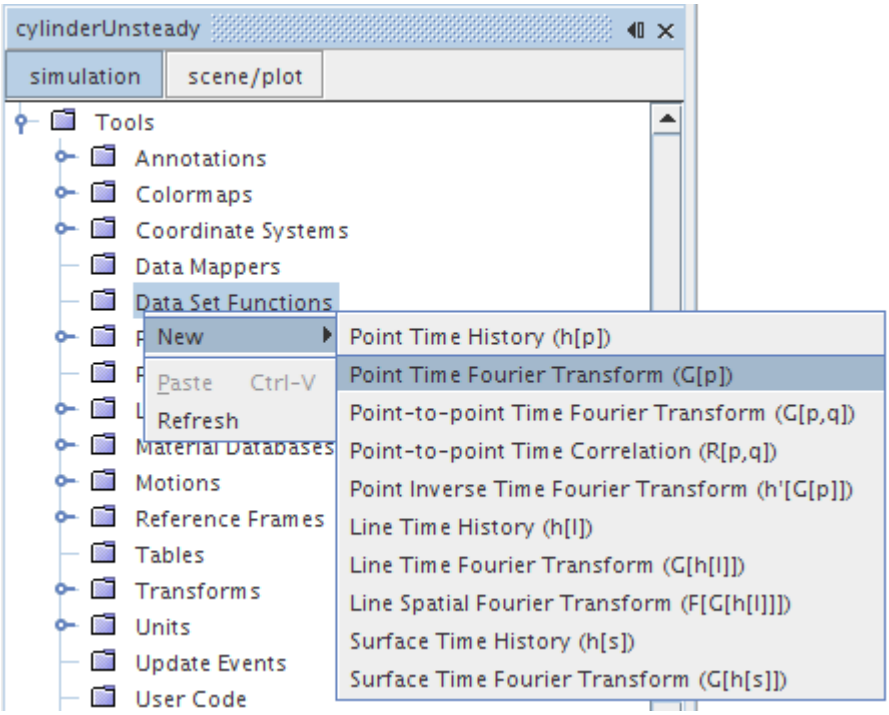


Spectral Analysis

Using Fast Fourier Transforms, you can determine sound pressure as a function of frequency at each of the point receivers.

Set up the spectral analysis.

1. Open the **Tools** node, right-click **Data Set Functions** and select **New** > **Point Time Fourier Transform (G(p))**.

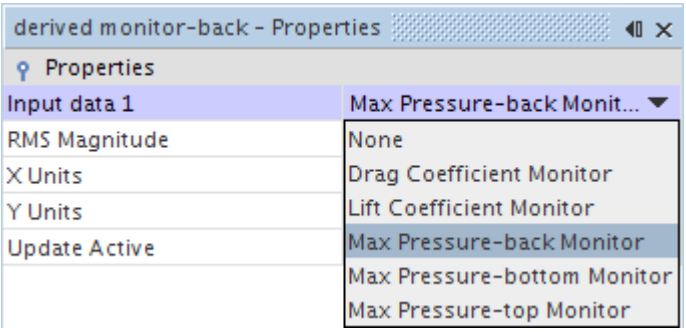


A new node, **G(p) 1**, is created in the simulation tree.

2. Set the following properties:

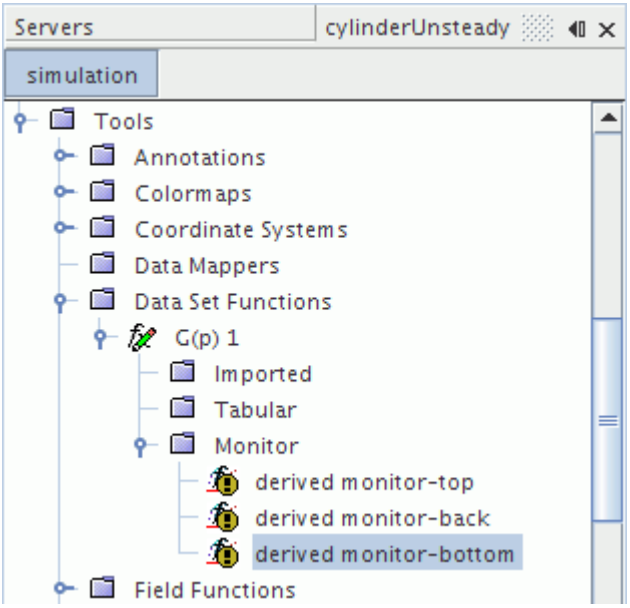
Property	Value
Start Time	`0.02 s`
Cut-off Time	`0.12 s`
Amplitude Function	**Sound Pressure Level**
Analysis Blocks	`3`
Overlap Factor	`0.5`
Window Function	**Hann**

3. Right-click the **G(p) 1** > **Monitor** node and select **New derived data from monitor**.
4. Rename the **Monitor** node to `derived monitor-back`.
5. In the *Properties* window, set *Input data 1* to **Max Pressure-back Monitor**.



6. Right-click the **G(p) 1** > **Monitor** node and select **New derived data from monitor** for a second time.
7. Rename the **Monitor** entry to `derived monitor-bottom`.
8. In the *Properties* window, set *Input data 1* to **Max Pressure-bottom Monitor**.
9. Right-click the **G(p) 1** > **Monitor** node and select **New derived data from monitor** for a third time.
10. Rename the **Monitor** entry to `derived monitor-top`.
11. In the *Properties* window, set *Input data 1* to **Max Pressure-top Monitor**.

    This portion of the object tree is shown in the following screenshot:



12. For each of the derived monitor nodes, activate the *Update Active* property.
13. Rename the **G(p) 1** node to `FFT-SPL`.

You now add the derived monitor nodes to a monitor plot.

14. Right-click the **Plots** node and select **New Plot** > **Monitor**.
15. Rename the **Monitor Plot 1 node** to `Monitor Plot - SPL`.
16. Select the **Plots** > **Monitor Plot - SPL** > **Data Series** node, right-click, and select **Add Data**.

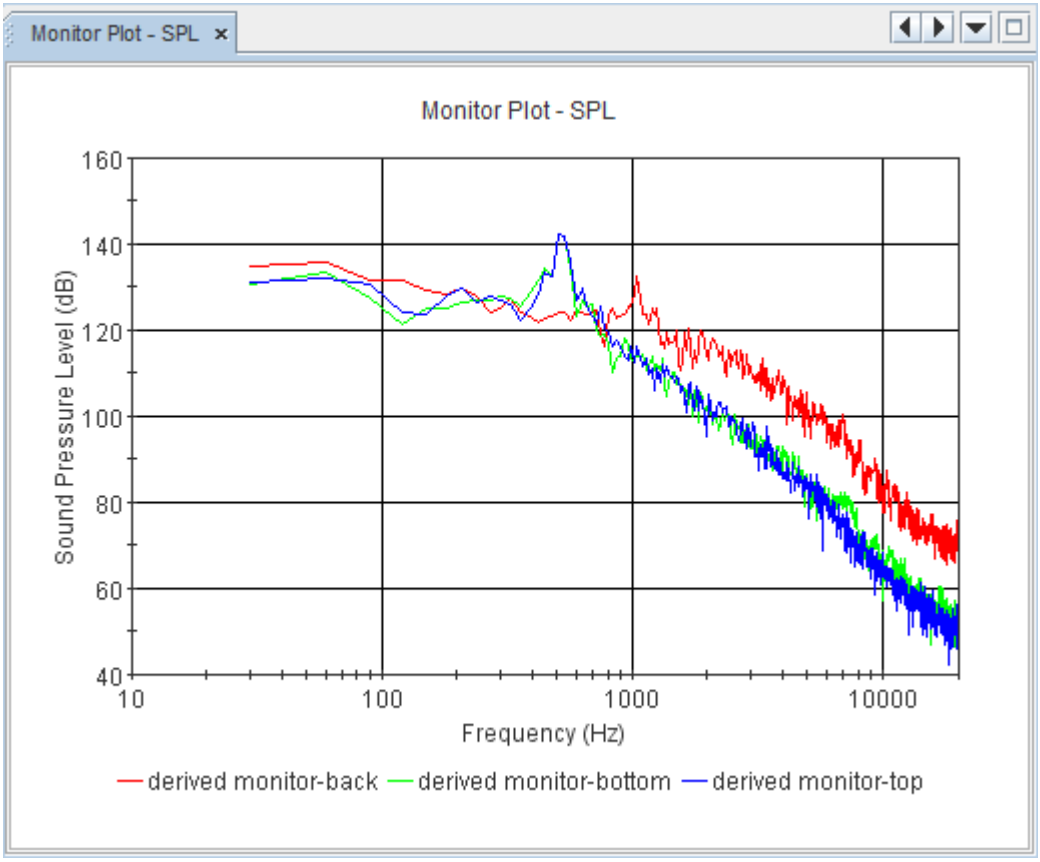
    This opens the *Add Data Providers to Plot* dialog.
17. Expand the Derived Data node and select the following:
    - derived monitor-back
    - derived monitor-bottom
    - derived monitor-top

18. Select the **Plots** > **Monitor Plot - SPL** > **Axes** > **X Axis** node.
19. In the *Properties* window, activate the *Logarithmic* property. Set the following properties:

Property	Value
Logarithmic	Activated
Minimum	`10`
Maximum	`20000`

20. Select the **X Axis** > **Title** node and set *Title* to `Frequency (Hz)`.
21. Select the **Y Axis** > **Title** node and set *Title* to `Sound Pressure Level (dB)`.

A scene similar to the one shown below is displayed.



From the above monitor plots, it is seen that there are two peaks: one at 500 Hz for the top and bottom probes (lift mode) and one at 1000 Hz for the back probe (drag mode).

22. Save the simulation.

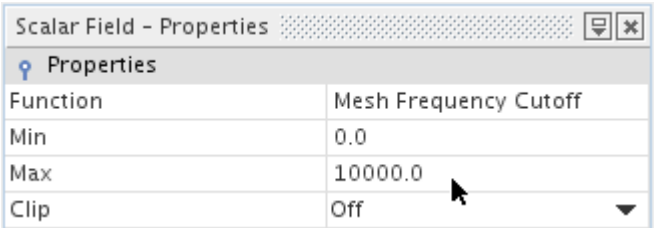
Results Analysis

Compare the peaks in the spectra with the Mesh Frequency Cutoff plot from the final steady-state simulation.

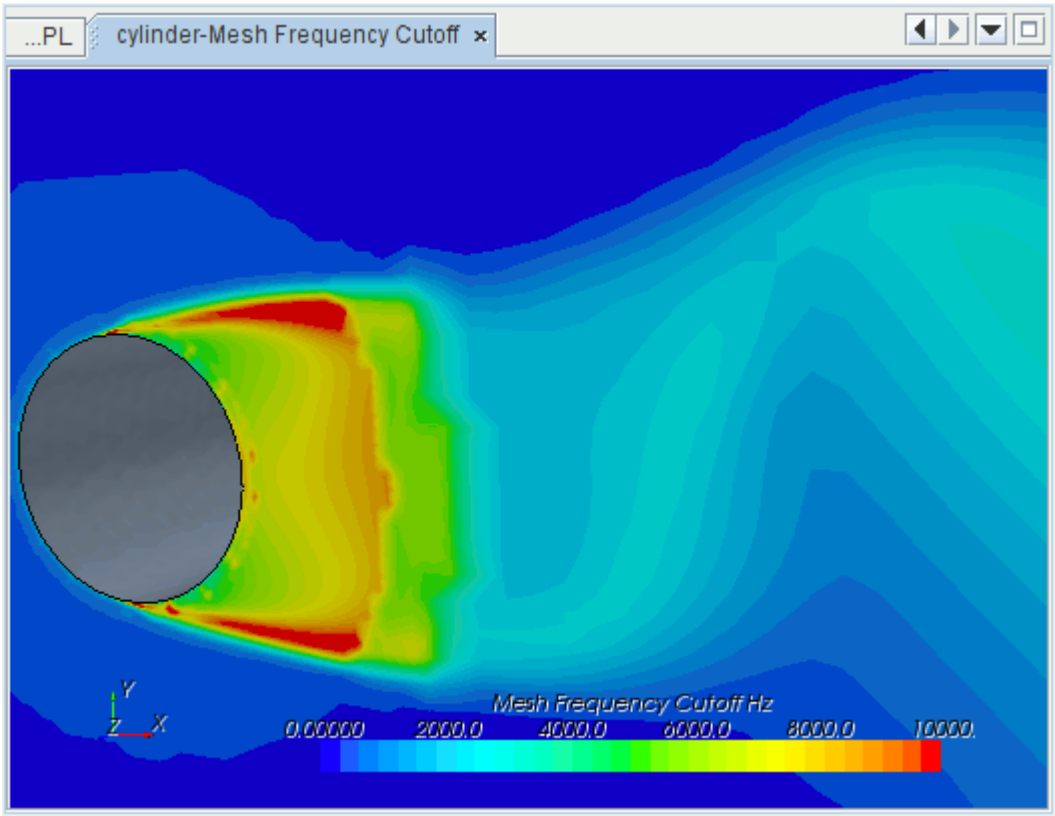
Use the following steps:

1. Open the simulation file for the steady-state calculation after 400 iterations (that is, the file `cylinder.sim` containing the refined mesh) alongside the existing `cylinderUnsteady` simulation.

2. In the same *Simulation* window, open the **Scenes** node.
3. Right-click **Mesh Frequency Cutoff** and select **Open.**
4. Select the **Mesh Frequency Cutoff** > **Displayers** > **Scalar 1** > **Scalar Field** node and set *Max* to `10000`.



Use the mouse controls to obtain a plot similar to the following:



Compare the predicted spectra with the Mesh Frequency Cutoff plot from the steady-state simulation. You can see that there is a close correlation between the steady-state estimated cut-off frequency and the high-end frequency resolution from the DES calculation. All spectra show a change of gradient at ~6000 Hz. The change in gradient represents a change in the energy cascade to the higher frequencies, resulting artificially from inadequate resolution of the turbulent structures. These gradient kinks correspond with the fine-mesh steady-state Mesh Frequency Cutoff values local to those points. The spectral peak at ~500 Hz is the predicted vortex shedding frequency.

Many researchers (for example, Strouhal, Oertel, and Williamson, among others) have studied the sound that is generated by the flow around a circular cylinder.

Strouhal found that the frequency f of the sound that is radiated from a cylinder of diameter D is related to the free-stream velocity U_∞ of a uniform flow. This relation is known as the Strouhal number $St = fD/U_\infty$. The Strouhal number is 0.2 to 0.22 for the Reynolds number range $300 \leq \mathrm{Re} \leq 10^4$(Inoue). For this case of a cylinder 20 mm in diameter and a free-stream air velocity of 50 m/s, the Reynolds number is 6.38 × 10^4 and the Strouhal number is 0.2. The frequency f of the sound is 500 Hz. This predicted frequency is identical to the shedding frequency of vortices behind the cylinder.

The SPL vs. frequency plot can numerically predict the frequency peak of 500 Hz generated by vortex shedding from the cylinder surface into its wake. This suggests that DES predicts the vortex shedding well and that the Strouhal number is close to the experimental value.

5. Save and close the simulation.

Summary

This tutorial continued from the previous one, doing an unsteady analysis of the near field and an FW-H analysis of the far field.

The tutorial introduced the following STAR-CCM+ features:

- Setting up a transient calculation.
- Modifying solver and stopping criteria parameters.
- Setting up monitoring report and plot.
- Saving transient surface data.
- Setting up spectral analysis.

DES and FW-H On-The-Fly Tutorial Bibliography

Strouhal, V. 1878, “Ueber eine besondere art der tonerregung.” *Annu. Phys. Chem. (Wied. Annu. Phys.)* Vol. 5, pp. 216-251.

Oertel, Jr. H., 1990, “Wakes behind blunt bodies” *Annu. Rev. Fluid Mech.* Vol 22, pp. 539-564.

Williamson, C. H., K., 1996, “Vortex dynamics in the cylinder wake” *Annu. Rev. Fluid Mech.* Vol 28, pp. 477-539.

Inoue O., Hatakeyama, N., 2002 “Sound generation by a two-dimensional circular cylinder in a uniform flow” *J. Fluid Mech.* Vol. 471, pp. 285-314.

Travin, A., Shur, M., 2000 “Detached-Eddy Simulations Past a Circular Cylinder” *Combustion*, Vol 63(1), pp. 293-313

Ffowcs Williams-Hawkings: Sound Propagation

This tutorial compares the Post FW-H model with the FW-H On-The-Fly model.

In the previous tutorial, DES and FW-H On-The-Fly: Noise from a Cylinder (Unsteady Analysis), you completed the DES and FW-H On-the-Fly transient runs. In this tutorial, you first import the transient surface data file that was generated during the transient runs. You then use the Post FW-H option of the FW-H model to calculate sound pressure levels at locations corresponding to the far field monitor points for which transient data was collected before. By comparing the Post FW-H model predictions with the results of the FW-H On-The-Fly analysis, you can verify that the results obtained from the `.trn` file with the Post FW-H model are identical to those obtained while actually running the unsteady simulation.

Contents:

Importing the Transient Surface Data
Preparing for Analysis with the Post FW-H Model
Plotting Sound Pressure Levels from FW-H Data
Summary
Using the Transient Surface Data Tutorial Bibliography

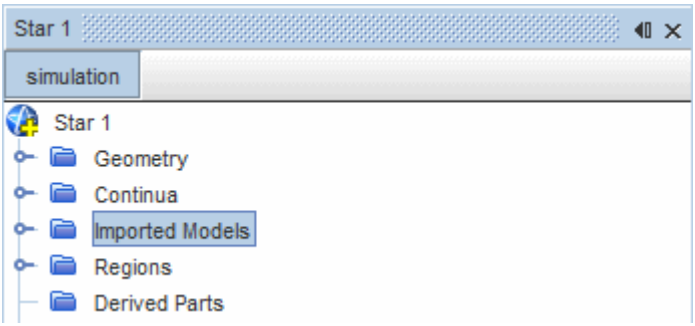
Importing the Transient Surface Data

If the transient surface data file has been saved as instructed previously, it can be loaded in the last saved simulation, cylinderUnsteady@04800.sim. This section describes the steps for doing so.

Use the following steps:

1. Start up a double-precision version of STAR-CCM+ in a manner that is appropriate to your working environment and load cylinderUnsteady@04800.sim.
2. Select **File** > **Import** > **Import CAE Model.**
3. Select `CylinderPressureData.trn` and click **Open**.
4. In the *Import Options* dialog, deactivate **Open geometry scene after import** and click **OK**.

   This action adds the **Imported Models** node to the object tree.



5. Expand the **Imported Models** node and select the **trn:cylinderPressureData** node.

Note the properties *Transient Display Time* and *Transient End Time*. These are used to examine the pressure data at different points in time.

6. Save the simulation as `3Dcylinder_FWH.sim`.

Preparing for Analysis with the Post FW-H Model

In this section, you select the Post FW-H model, set up two Post FW-H receivers, then run the Post FW-H solver. Receivers are points at specific distances from the noise sources where sound pressure is calculated. The receivers take their data from the imported CylinderPressureData.trn file.

Contents:

Choosing the Post FW-H Model
Creating Two Derived Point Parts
Setting the Post FW-H Receivers
Running the Post FW-H Solver

Choosing the Post FW-H Model

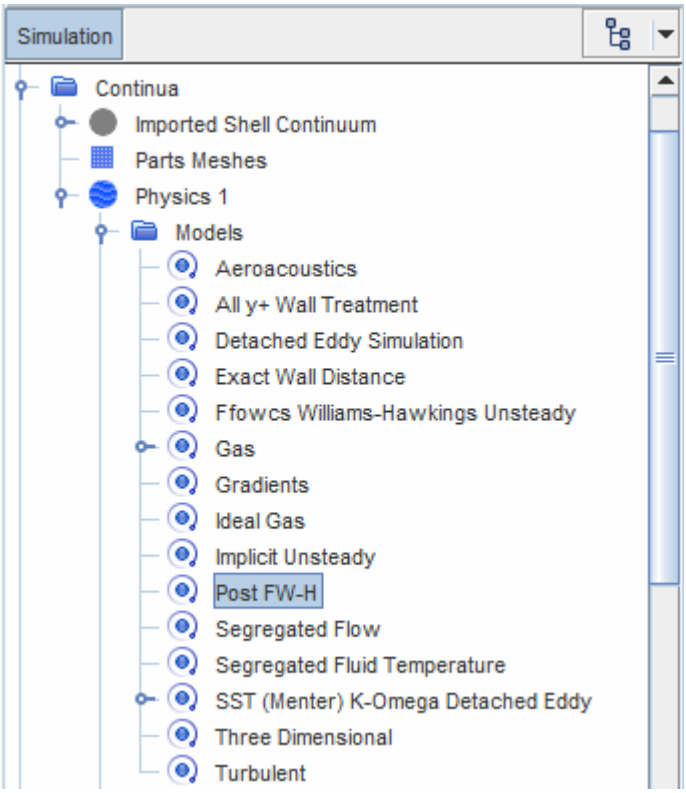
Add the Post FW-H model to the physics models.

To select the Post FW-H model:

1. Right-click the **Continua** > **Physics 1** > **Models** node and choose **Select Models.**

In the *Physics 1 Model Selection* dialog:

2. Deselect the **On-The-Fly FW-H** model.
3. Select **Post FW-H** from the *FW-H Unsteady Models* box.



4. Click **Close**.
5. Select the **Post FW-H Receivers** node and set the following properties:

Property	Setting
Far-Field Density	`1.18 kg/m^3`
Far-Field Sound Speed	`347.2 m/s`

Creating Two Derived Point Parts

Create two point parts at the positions of the point receivers.

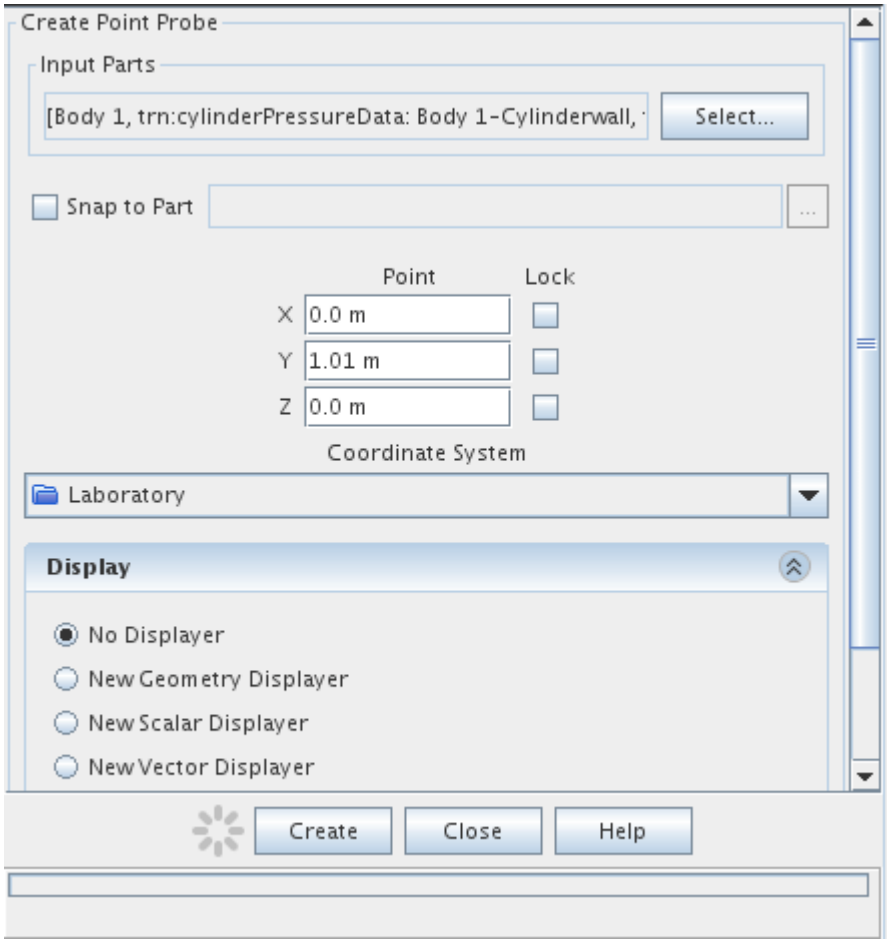
Use the following steps:

1. Right-click the **Scenes** > **Geometry Scene 1** node and select **Open**.
2. In the *Simulation* window, right-click the **Derived Parts** node and select **New Part** > **Probe** > **Point**.
3. In the *Create Point* panel, do the following:

   In the *Input Parts* box, select **Body 1** and **Imported models** > **trn:cylinderressureData** > **Imported Surfaces** > **Body 1-Cylinder wall**.

   Set the *Point* to `[0, 1.01, 0]` m, m, m.

   Under *Display*, select **No Displayer**.



Click **Create** and then **Close**.

4. Rename the **Derived Parts** > **point** node to `Plus90deg100r`.

Create another probe by copying and pasting the first one:

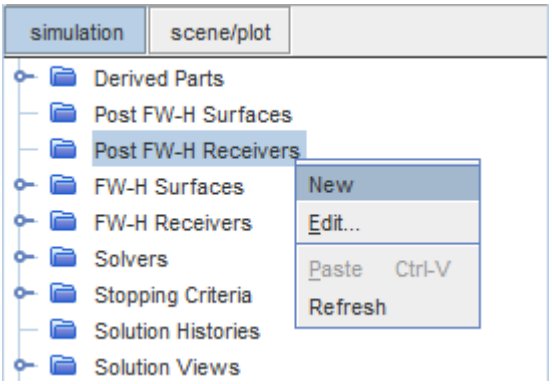
5. Right-click the **Plus90deg100r** entry and select **Copy**.
6. Right-click the **Derived Parts** node and select **Paste**.
7. Rename the **Derived Parts** > **Copy of Plus90deg100r** node to `Minus90deg100r`.
8. In the *Properties* window, set *Point* to [`0,-1.01,0`] m, m, m.

Setting the Post FW-H Receivers

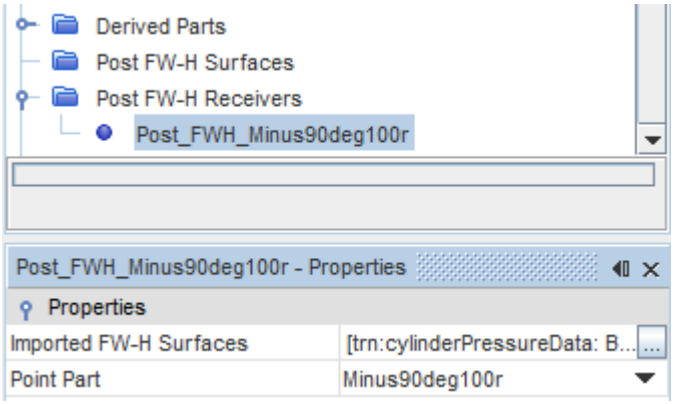
Two point-receivers are created for this analysis, both at a distance of 1.01 m from the cylinder surface. These point receivers take the point probes created previously as an input.

Set the point-receivers:

1. Right-click the **Post FW-H Receivers** node and choose **New** to create a new Point Receiver.



2. Select the **Post Point Receiver 1** node and set *Imported FW-H Surfaces* to **Body 1-Cylinderwall**.
3. For the same node, set *Point Part* to **Minus90deg100r**.
4. Rename the node to `Post_FWH_Minus90deg100r`.

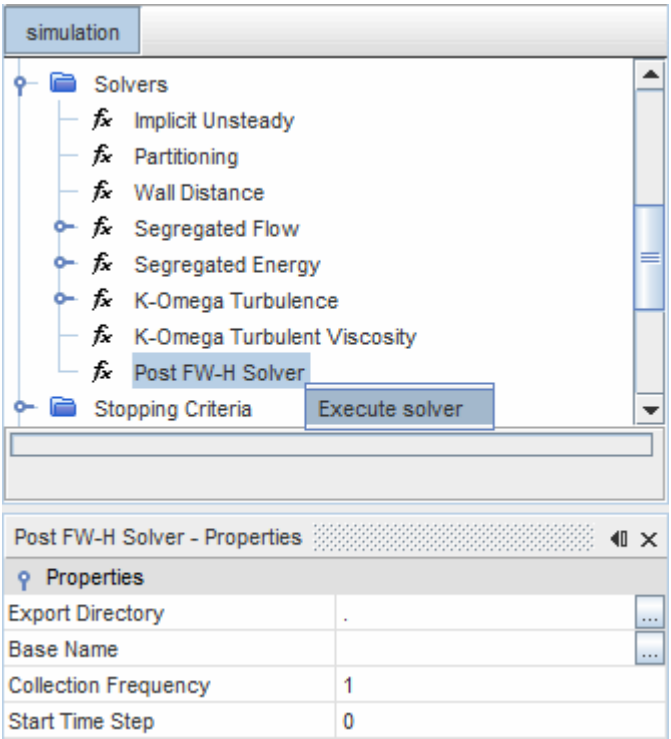


5. Using copy and paste, create a second point receiver node from the first one.
6. For this second node, set *Point Part* to **Plus90deg100r**.
7. Rename the node to `Post_FWH_Plus90deg100r`.

Running the Post FW-H Solver

You are now ready to run the Post FW-H solver.

1. Right-click on the **Post FW-H Solver** node and select **Execute Solver**.



2. A prompt displays, reading `Reset Transient Display Time?` Click **Yes**.

Plotting Sound Pressure Levels from FW-H Data

While generating the .trn file for the DES and FW-H On-The-Fly tutorial, STAR-CCM+ collected sound pressure data from two point receivers for the FW-H On-The-Fly model. In the following sections, you prepare plots of the sound pressure levels using data from both the On-The-Fly FW-H model and the Post FW-H model.

By plotting both sets of data on the same plot, you can verify that the Post FW-H model gives identical results to the On-The-Fly FW-H model. To use this data, the following steps are necessary:

1. Export the data from the point-receivers.
2. Import this data as a table.
3. Create a Fast Fourier Transform function that performs a spectral analysis using the sound pressure level data from the table created.
4. Plot this data with appropriate scales.

Contents:

Exporting Sound Pressure Data from the On-The-Fly FW-H Receivers
Exporting Sound Pressure Data from the Post FW-H Receivers
Importing Sound Pressure Data
Applying Fast Fourier Transform to Table Data
Creating a Plot of Sound Pressure Level

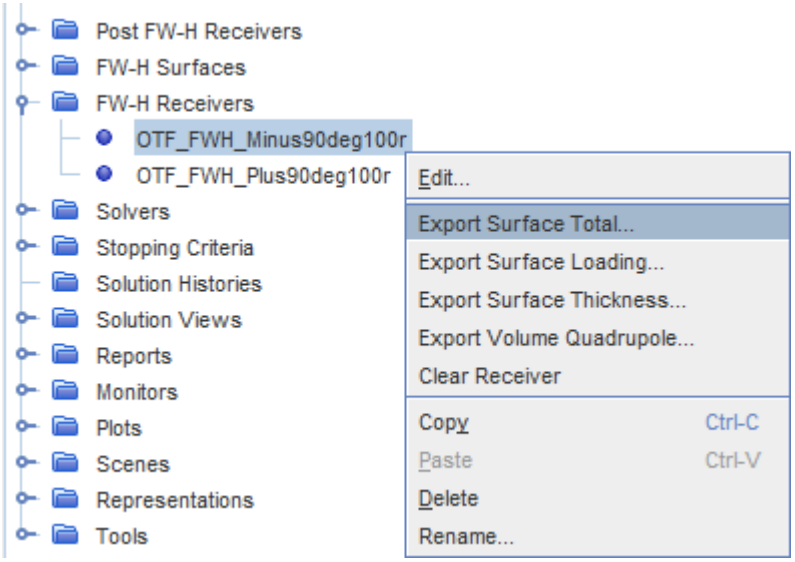
Exporting Sound Pressure Data from the On-The-Fly FW-H Receivers

Export the sound pressure data to external files.

The sound pressure data from the unsteady simulation is contained within the FW-H point-receivers.

Use the following steps:

1. Right-click the **FW-H Receivers** > **OTF_FWH_Minus90deg100r** node and choose **Export Surface Total**.



   The *Save* dialog appears.

2. For **File Name**, enter `OTF_FWH_Minus90deg100r`. Click **Save**.
3. Similarly, right-click the **FW-H Receivers** > **OTF_FWH_Plus90deg100r** node, choose **Export Surface Total**, and set **File Name** to `OTF_FWH_Plus90deg100r`. Click **Save**.

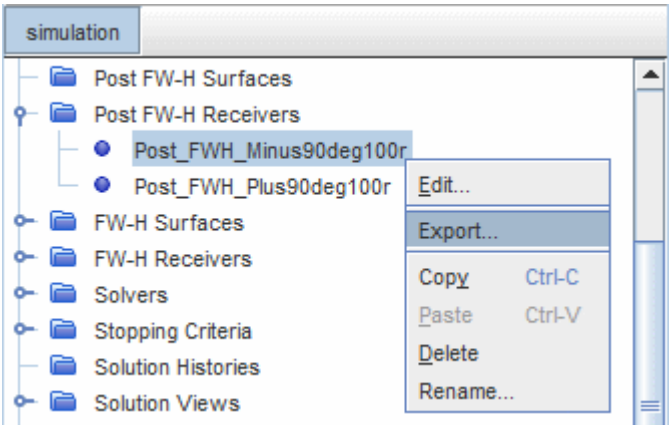
Two files are created in the working directory.

Exporting Sound Pressure Data from the Post FW-H Receivers

Export the sound pressure data to external files.

The sound pressure data generated by the Post FW-H model is contained within the Post FW-H point receivers.

1. Right-click the **Post FW-H Receivers** > **Post_FWH_Minus90deg100r** node and choose **Export...**



The *Save* dialog appears.

2. Enter `Post_FWH_Minus90deg100r` in the **File Name** entry box of the dialog and click **Save** to export the data.
3. Similarly, right-click the **Post FW-H Receivers** > **Post_FWH_Plus90deg100r** node, choose **Export...**, and enter `Post_FWH_Plus90deg100r` in the **File Name** entry box of the *Save* dialog. Click **Save** to export the data.

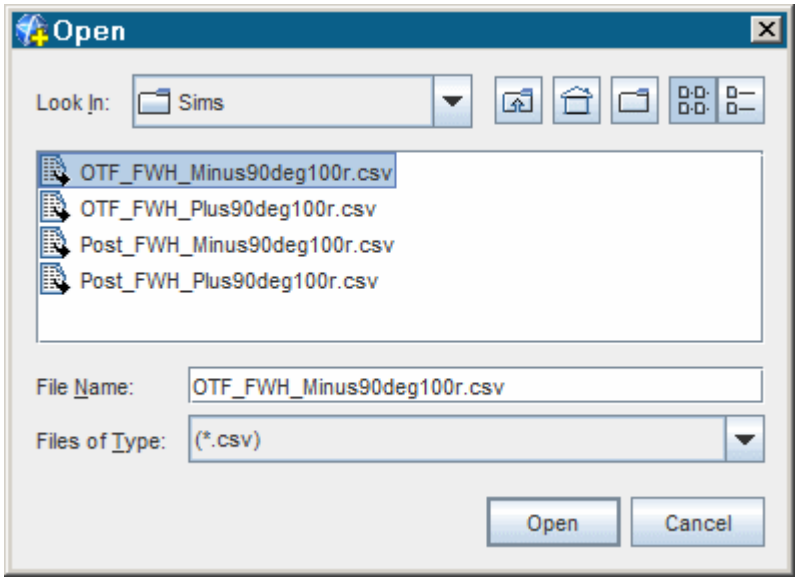
Two files are created in the working directory.

Importing Sound Pressure Data

Import the sound pressure data as tables.

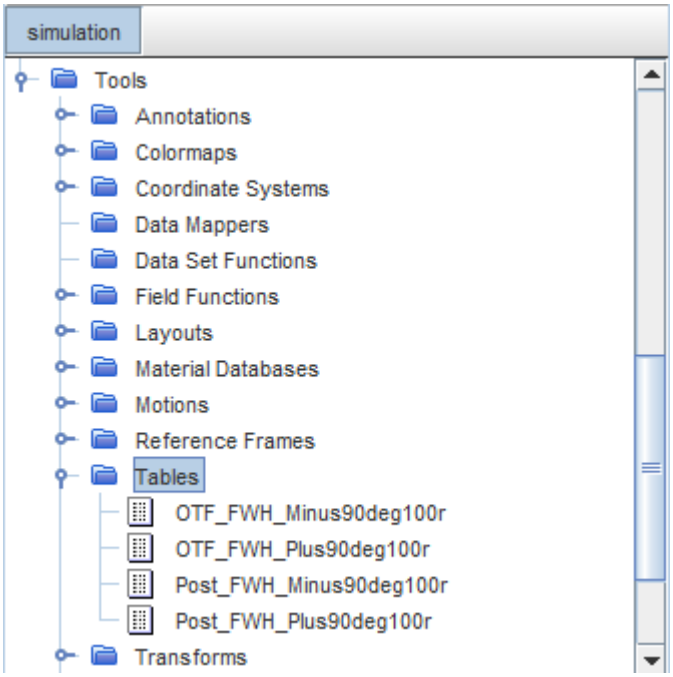
To import the sound pressure data from the four files:

1. Right-click the **Tools** > **Tables** node and select **New Table** > **File Table**.
2. In the *Open* dialog, select the **OTF_FWH_Minus90deg100r.csv** file and click **Open**.



3. Repeat these steps for the following data files:
   - **OTF_FWH_Plus90deg100r.csv**
   - **Post_FWH_Minus90deg100r.csv**
   - **Post_FWH_Plus90deg100r.csv**

Four additional nodes are created beneath the **Tables** manager.

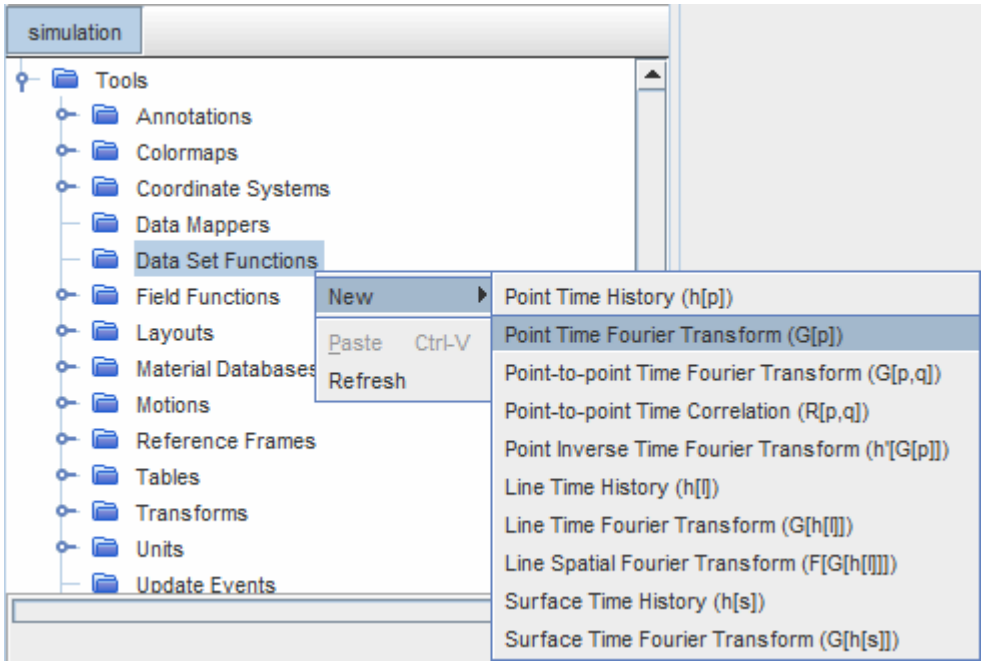


Applying Fast Fourier Transform to Table Data

Use a fast Fourier transform data set to analyze the sound pressure level at different frequencies.

Use the following steps:

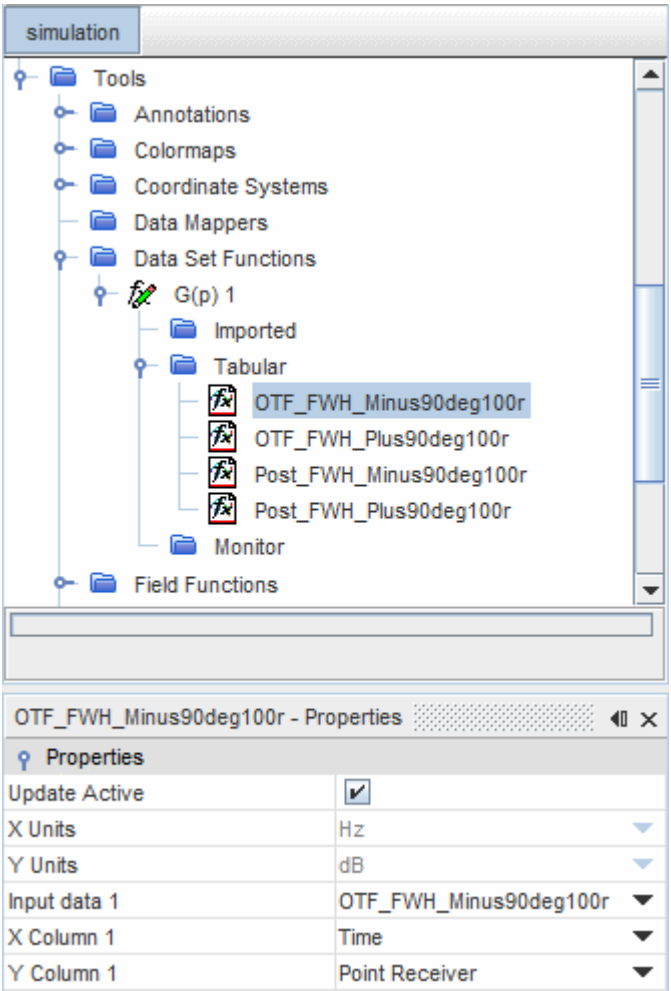
1. Right-click the **Tools** > **Data Set Functions** node and choose **New** > **Point Time Fourier Transform (G(p))**.



A node, **G(p) 1**, is added to the **Data Set Functions** manager node.

2. Right-click the **G(p) 1** > **Tabular** node and select **New derived data from table**.
3. Rename the node to `OTF_FWH_Minus90deg100r`.
4. For this node, set the following properties:

Property	Setting
Input data	**OTF_FWH_Minus90deg100r**
X Column 1	**Time**
Y Column 1	**Point Receiver (0 -1.01 0)**



5. Repeat this process to create three more Tabular nodes, for a total of four. The four nodes have identical properties, except for names and *Input data*:
6. Select the **G(p) 1** node and set the following properties:

Property	Setting
Update Interval	`1`
Start Time	`0.02 s`
Cut-off Time	`0.12 s`
Amplitude Function	**Sound Pressure Level**
Analysis Blocks	`3`
Overlap Factor	`0.5`
Window Function	**Hann**

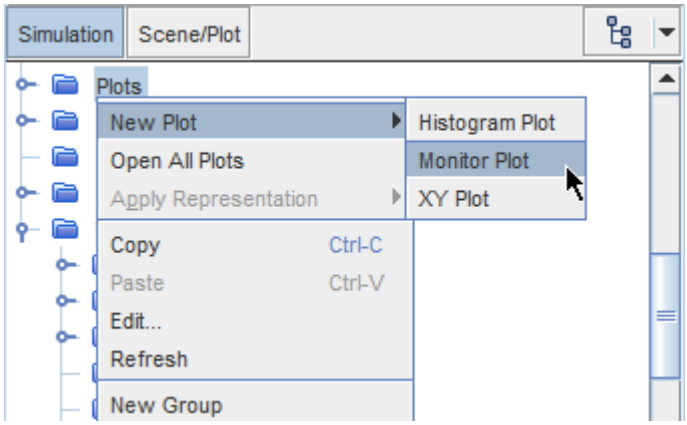
7. Select the *Reference Sound Pressure* node below the **G(p) 1** node and set *Value* to `2.0E-5 Pa`.
8. Select each of the four nodes under the **Tabular** node and activate *Update Active* in each one.

Creating a Plot of Sound Pressure Level

The spectral analysis plot takes its data from the fast Fourier transform created previously.

Use the following steps:

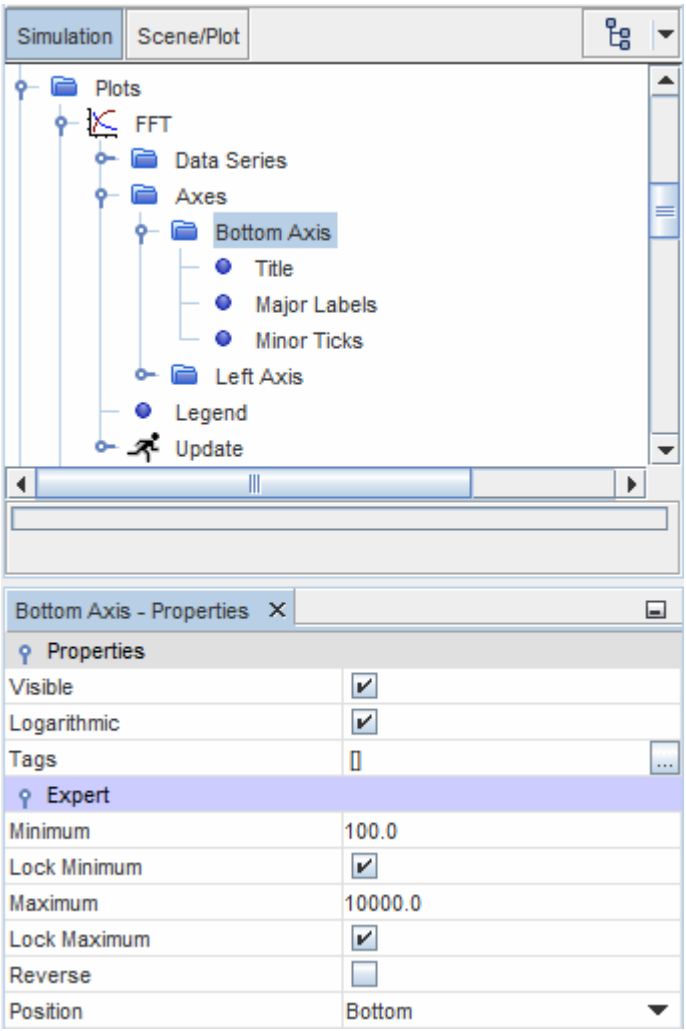
1. Right-click the **Plots** node and choose **New Plot** > **Monitor Plot**.



2. Rename the **Monitor Plot 1** node to `FFT`.
3. Open the **FFT** node, right-click on the **Data Series** node, and select Add Data.
   The *Add Data Providers to Plot* dialog opens.
4. In this dialog, select and expand Derived Data, then select the following items:
   - OTF_FWH_Minus90deg100r
   - OTF_FWH_Plus90deg100r
   - Post_FWH_Minus90deg100r
   - Post_FWH_Plus90deg100r

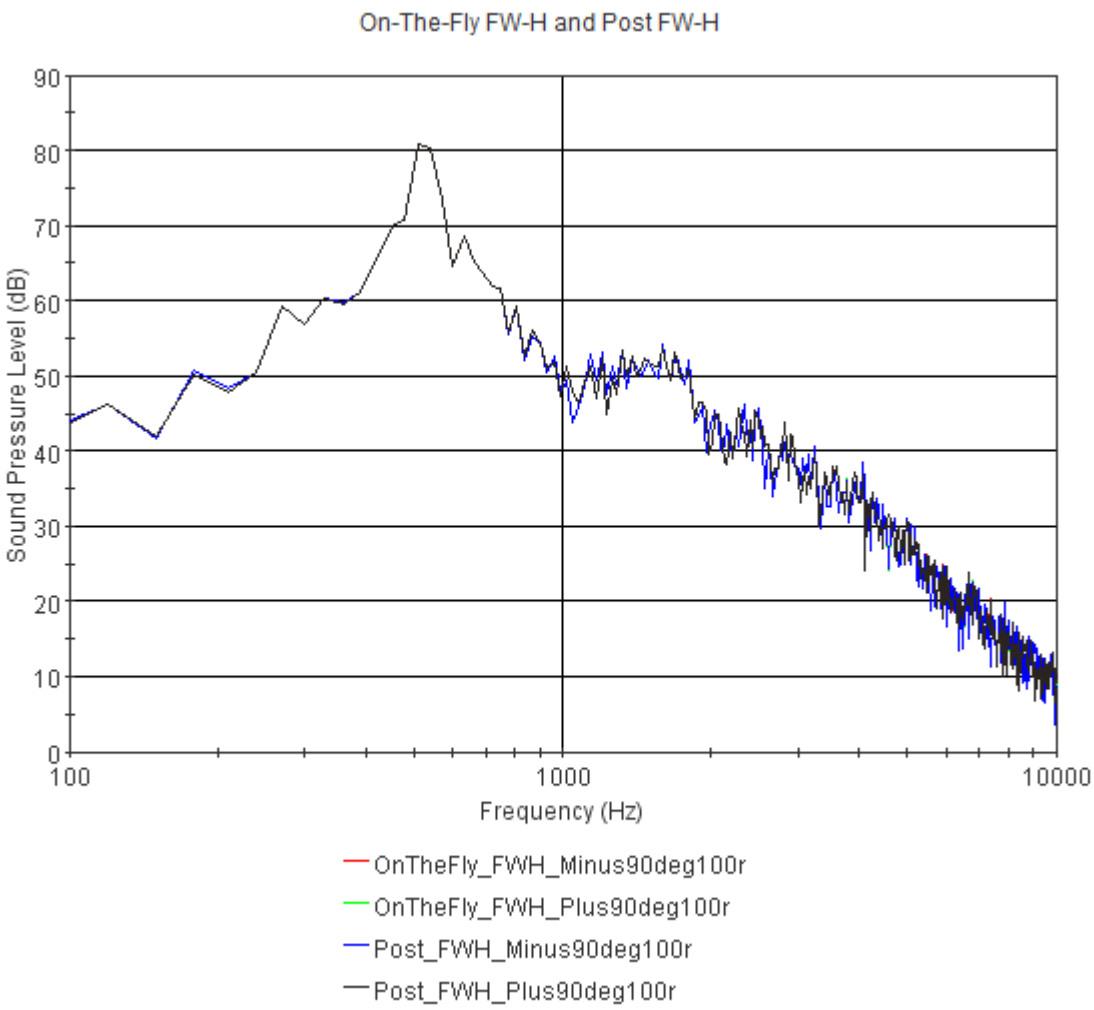
Change the X-axis:

5. Select the **FFT** > **Axes** > **Bottom Axis** node and activate *Logarithmic*.
6. Set *Minimum* to `100` and *Maximum* to `10000`.



7. Select the **FFT** node and set the *Title* property to `On-The-Fly FW-H and Post FW-H`.

The final plot appears similar to the one shown below.



The Post FW-H model allows you to obtain noise prediction at receiver locations not defined in the original unsteady simulation. As the computational cost of running a Post FW-H analysis is much lower than the cost of running an unsteady analysis, you can continue to explore the solution space for little extra effort.

Summary

This tutorial compared the results of the previous tutorial with those of the Post FW-H solver.

This tutorial introduced the following STAR-CCM+ features:

- Post-processing transient surface data
- Applying fast Fourier transforms
- Plotting sound pressure levels

Using the Transient Surface Data Tutorial Bibliography

Strouhal, V. 1878, “Ü ber eine besondere art der tonerregung.” *Annu. Phys. Chem. (Wied. Annu. Phys.)* Vol. 5, pp. 216-251.

Oertel, Jr. H., 1990, “Wakes behind blunt bodies” *Annu. Rev. Fluid Mech.* Vol 22, pp. 539-564.

Williamson, C. H., K., 1996, “Vortex dynamics in the cylinder wake” *Annu. Rev. Fluid Mech.* Vol 28, pp. 477-539.

Inoue O., Hatakeyama, N., 2002 “Sound generation by a two-dimensional circular cylinder in a uniform flow” *J. Fluid Mech.* Vol. 471, pp. 285-314.

Travin, A., Shur, M., 2000 “Detached-Eddy Simulations Past a Circular Cylinder” *Combustion*, Vol 63(1), pp. 293-313

Signal Post-Processing: FFT and Wavenumber

Several further analyses are possible using the transient surface data file from the earlier tutorial, DES and FW-H On-The-Fly: Noise from a Cylinder (Unsteady Analysis). These analyses are covered in this tutorial.

Contents:

Importing the Transient Surface Data
Plotting Pressure at Different Time-Steps
Plotting the Sound Pressure Level at Different Frequencies
Determining the Group Velocity of the Disturbances
Summary

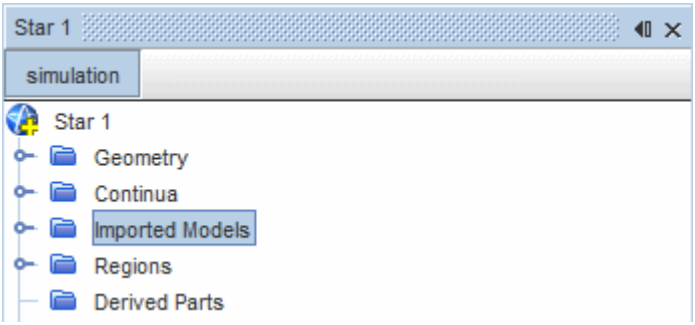
Importing the Transient Surface Data

Create a new simulation, based on the transient surface data that you obtained previously from the tutorial, DES and FW-H On-The-Fly: Noise from a Cylinder (Unsteady Analysis).

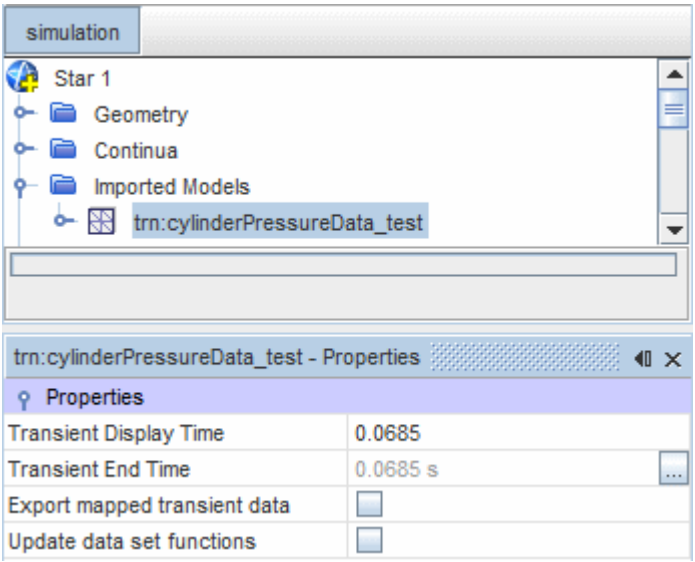
Use the following steps:

1. Start up a double-precision version of STAR-CCM+ in a manner that is appropriate to your working environment.
2. Select the New Simulation option from the menu bar.
3. Select **File** > **Import** > **Import CAE Model**.
4. Select `cylinderPressureData.trn` and click **Open**.
5. In the *Import Options* dialog, make sure that *Units* are set to **m** and that **Open geometry scene after import** is deactivated, then click **OK**.

   This action adds the **Imported Models** node to the object tree.



6. Expand the **Imported Models** node and select the **trn:cylinderPressureData** node.



Note the properties. These are used to examine the pressure data at different points in time.

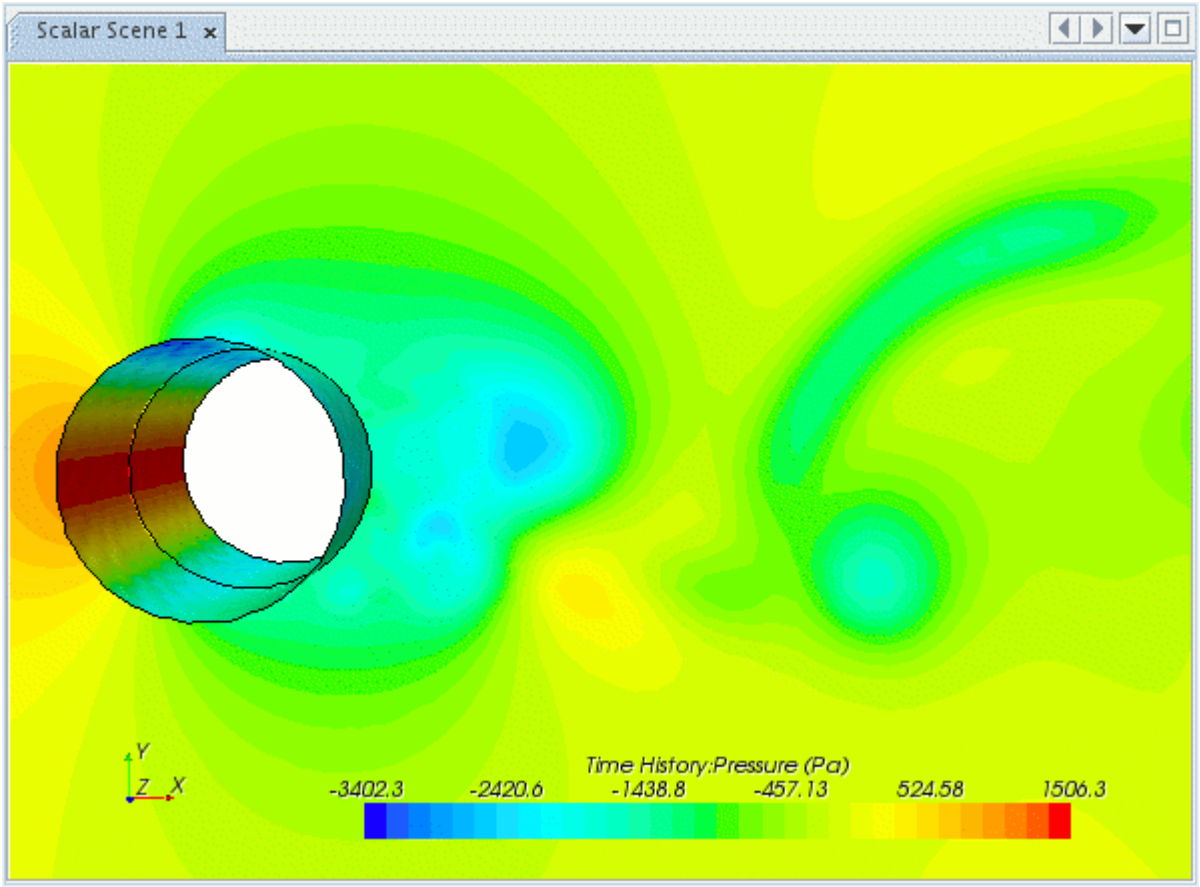
7. Save the simulation as `cylinderPressureData.sim`.

Plotting Pressure at Different Time-Steps

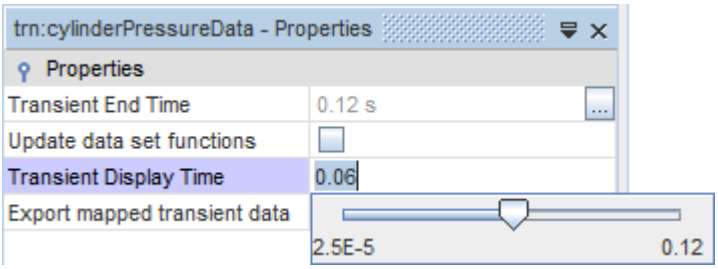
A scalar scene showing pressure is created; the **trn:cylinderPressureData** node is used to change the time for which the data is shown.

1. Create a scalar scene.
2. Select the **Scalar Scene 1** > **Displayers** > **Scalar 1** node and set *Contour Style* to **Smooth Filled**.
3. Select the **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Parts** node and click to the right of the *Parts* property. In the object selection dialog that opens, select all the nodes under the **Imported Models** > **trn:cylinderPressureData** node.
4. Select the **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Scalar Field** node.
5. Click to the right of the *Function* property and choose **Time History:Pressure**.

The scalar scene is updated to show the pressure field at the point in time specified by the *Transient Display Time* property of the **Imported Models** > **trn:cylinderPressureData** node. In this instance, the initial value is set to 0.12 s.



6. Select the **Imported Models** > **trn:cylinderPressureData** node and click to the right of *Transient Display Time*. Use the slider bar (or its accompanying edit box) to set the value to `0.06`.



The scalar scene updates to show the pressure field at 0.06 s.

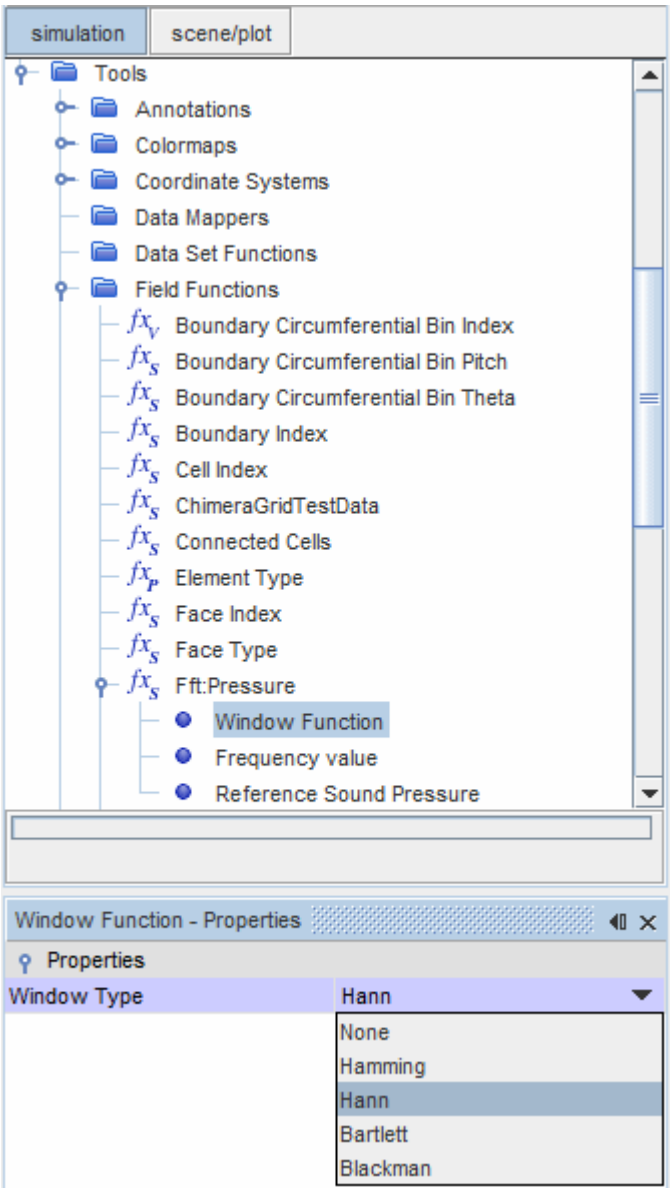
Plotting the Sound Pressure Level at Different Frequencies

The entire transient surface data file can be used in a Fast Fourier Transform to plot the sound pressure level across the stored surfaces at selected frequencies.

1. Expand the **Tools** > **Field Functions** node and select the **Fft:Pressure** node.
2. In the *Properties* window, set the the following properties:

Property	Setting
Amplitude Function	**Sound Pressure Level**
Analysis Blocks	`3`
Overlap Factor	`0.5`
Fft Input Start Time	`0.02 s`
Fft Input Cut-off Time	`0.12 s`

3. Select the **Fft:Pressure** > **Window Function** node and set the *Window Type* to **Hann**.

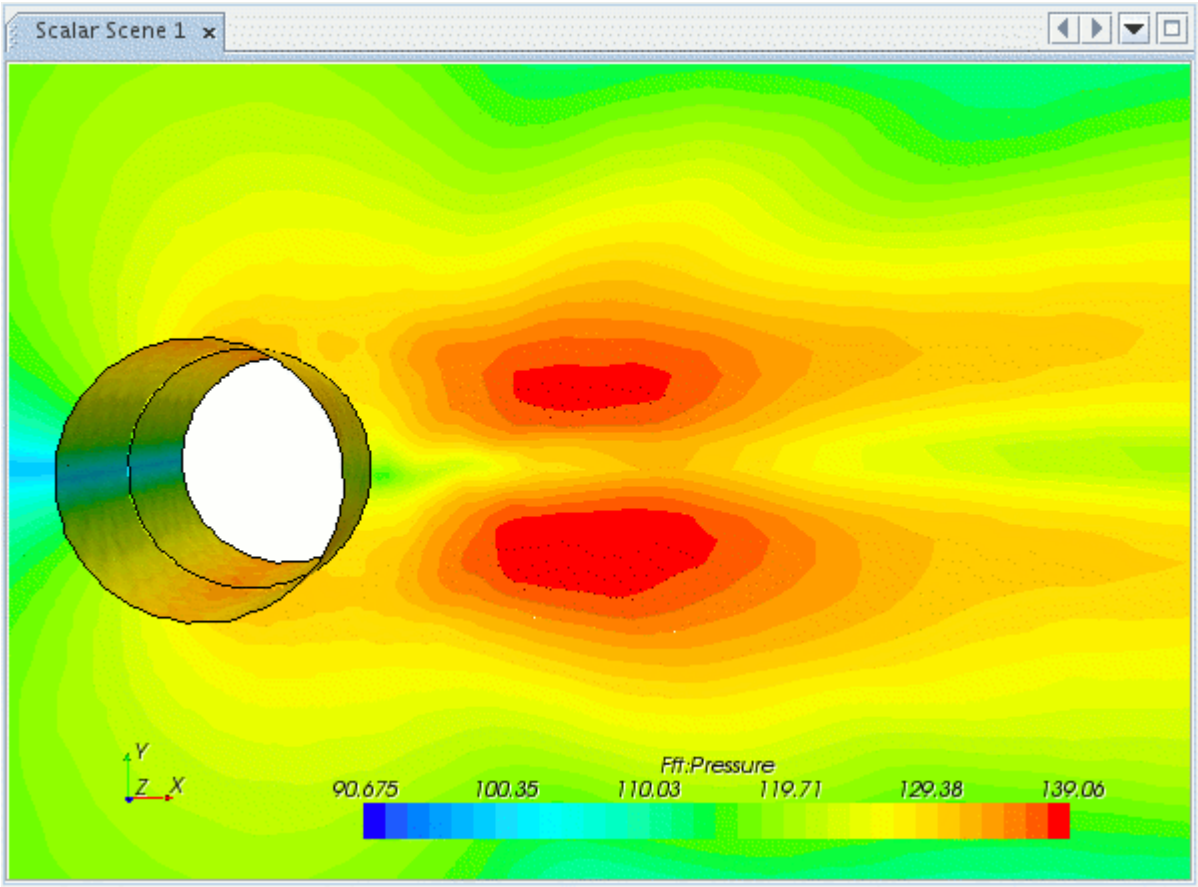


4. Select the **Fft:Pressure** > **Frequency Value** node and set the Value to `500 Hz`.

Plot the sound pressure level using *Scalar Scene 1*.

5. Make sure *Scalar Scene 1* is open and visible in the *Graphics* window.
6. Select the **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Scalar Field** node and set the *Function* to **Fft:Pressure**.

   STAR-CCM+ calculates the Fast Fourier Transform for the specified frequency. When the calculation completes, the scene updates.
7. Manipulate the scene using the mouse controls to obtain something similar to the following:



The sound pressure level at different frequencies can be viewed by modifying the **Frequency Value** of the **Fft:Pressure** field function. The SPL (dB) values at the top and bottom of the cylinder exactly match the point spectra dB levels (plotted previously) at 500 Hz.

8. Close the scene.

Determining the Group Velocity of the Disturbances

A wavenumber-frequency diagram (or fk plot) is used to determine the group velocity at which disturbances propagate along the domain.

To determine the group velocity for this simulation, first compute cross spectral densities or auto spectral densities using the data from the `.trn` file. Determine the range for which the data is meaningful for analysis and use this range to compute a line spatial Fourier transform. By plotting the spectra from this Fourier transform, you can compute the group velocity of the disturbances.

This section describes the steps for doing so:

1. Creating a line probe along which transient data is extracted.
2. Creating a line time history to record data from the line probe.
3. Calculating the spectral density along the line probe.
4. Calculating the line spatial Fourier transform and using it to determine the group velocity.

Contents:

Creating a Line Probe
Calculating the Spectral Density
Calculating the Line Spatial Fourier Transform

Creating a Line Probe

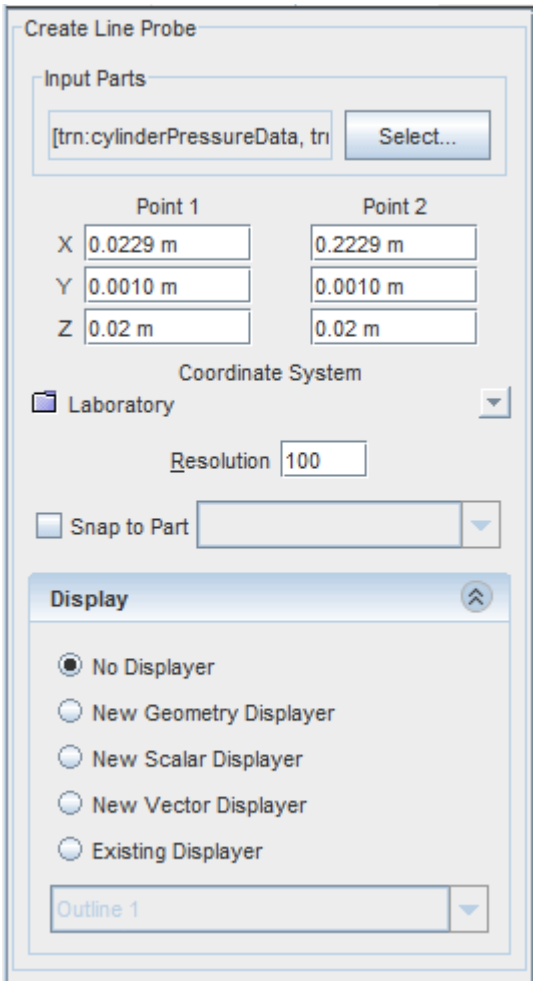
Create a line probe to sample the pressure time history.

1. Create a geometry scene.
2. Right-click the **Derived Parts** node and select **New Part** > **Probe** > **Line...**.

   The *Create Line Probe* window appears.
3. Click the **Select** button under *Input Parts* and activate **Imported Models** > **plane section** in the dialog box.
4. Click **OK**.
5. In the *Create Line Probe* window, set the following properties:

Property	Setting
Point 1	[`0.0229, 0.001, 0.02`] m, m, m
Point 2	[`0.2229, 0.001, 0.02`] m, m, m
Resolution	`100`
Display	**No Displayer**

   The *Create Line Probe* window appears as shown.



6. Click **Create**, and then click **Close**.

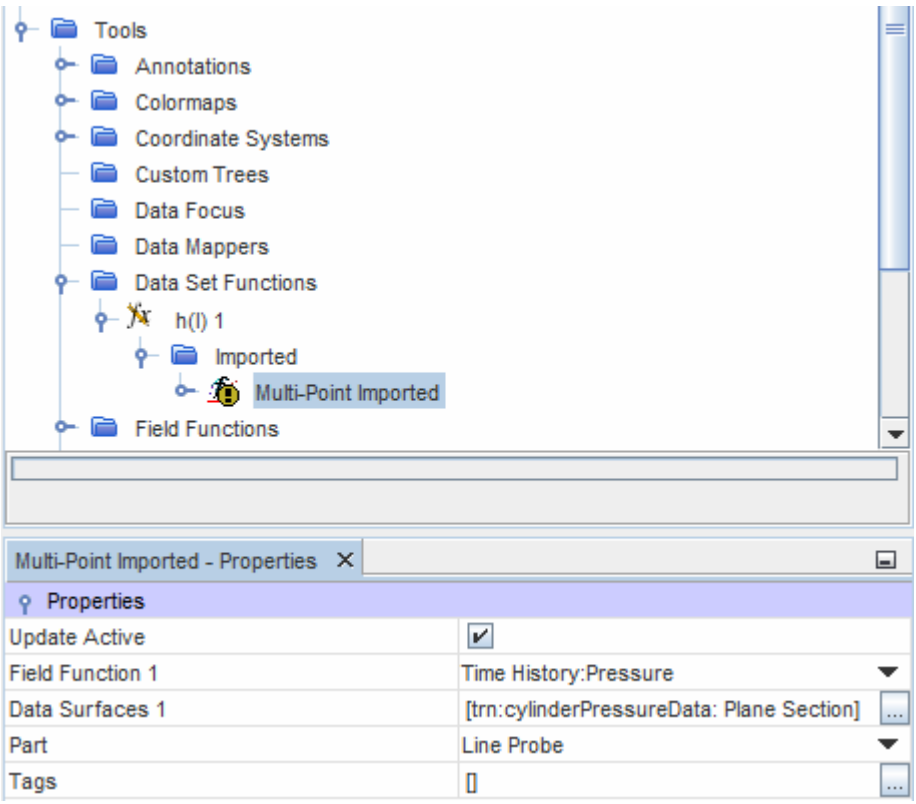
Calculating the Spectral Density

To calculate the spectral density, you must create a line time history, a line time Fourier transform, and a derived data set. You can then visualize the spectral density.

To create a line time history to record data from the line probe:

1. Right-click the **Tools** > **Data Set Functions** node and select the **New** > **Line Time History (h[l])** node.
2. Right-click the **h(l) 1** > **Imported** node and select **New derived data from imported surface and derived part**. The node **Multi-Point Imported** opens below.
3. Click on **Multi-Point Imported** and set the following properties:

Property	Setting
Field Function 1	**Time History:Pressure**
Data Surfaces 1	**plane section**
Part	**line-probe**
Update Active	Activated

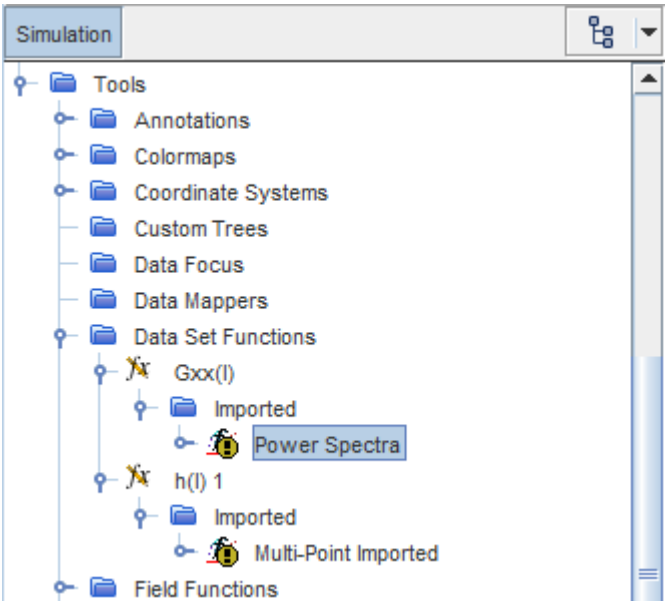


To create a line time Fourier transform:

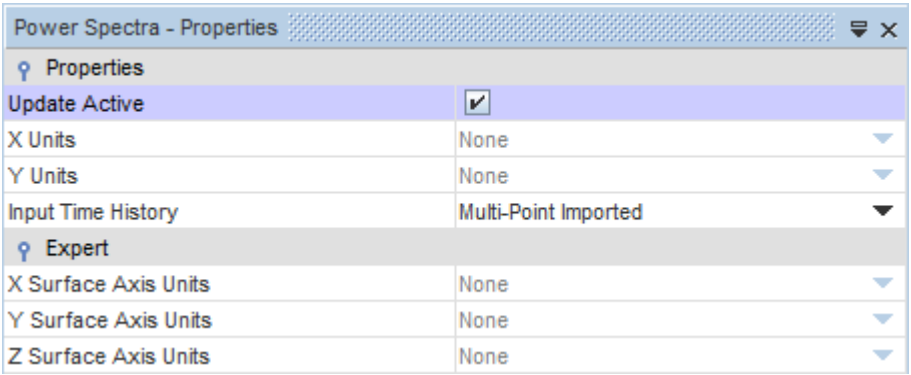
4. Right-click the **Tools** > **Data Set Functions** node and select **New** > **Line Time Fourier Transform (G[h[l]])**.
5. Rename the newly created **G(l) 1** node to `Gxx(l)`.
6. Select the **Tools** > **Data Set Functions** > **Gxx(l)** node and make sure the *Amplitude Function* is set to **Power Spectral Density**.

To create the derived data set node into which you import the data:

7. Right-click the **Gxx(l)** > **Imported** node and select **New derived data from derived data**.
8. Rename the newly created **Multi-Point Imported 2** node to `Power Spectra`.



9. Select the **Gxx(I)** > **Imported** > **Power Spectra** node.
10. Set *Input time history* to **Multi-Point Imported**.
11. Activate *Update Active*. All other properties are disabled for editing and updated automatically after you activate *Update Active*.



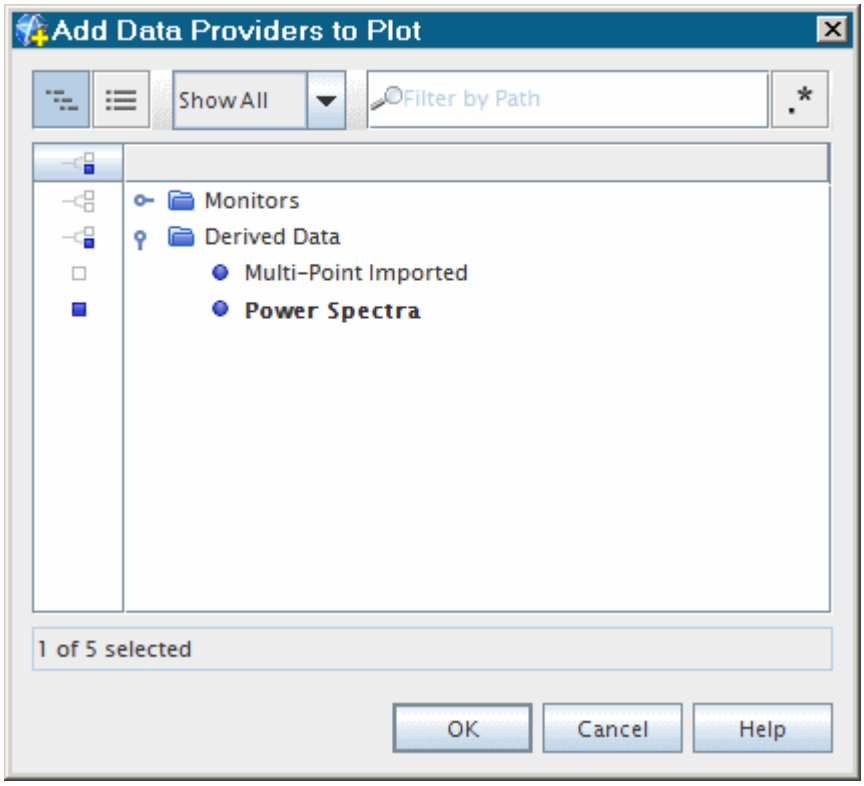
To visualize the spectral density:

With the *Update Active* property activated, the Fourier transform is automatically calculated once you assign the last property. This process takes a few minutes, depending on the processor speed of your computer.

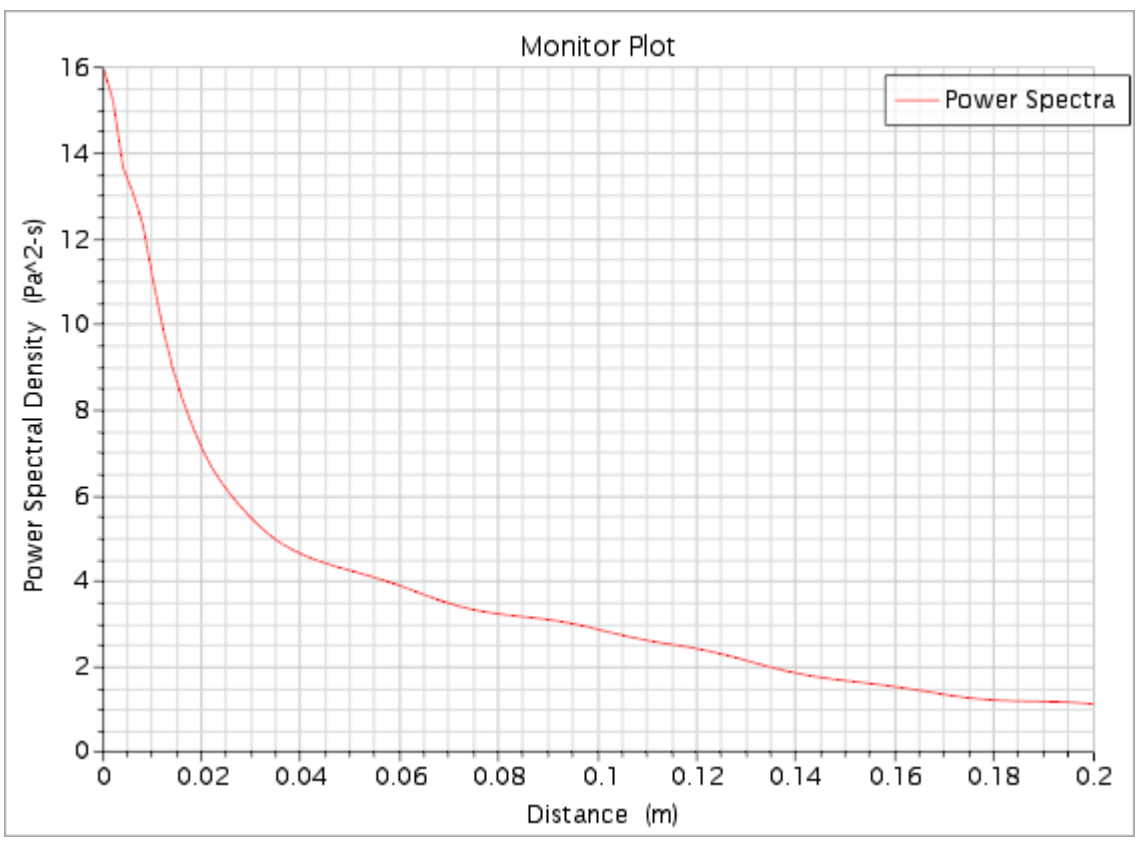
When the Fourier transform has been computed, visualize the spectral density as a function of distance. Use a logarithmic scale for the spectral density axis.

12. Create a monitor plot and rename it to `Power Spectra`.
13. Select the **Plots** > **Power Spectra** > **Data Series** node, right-click, and select Add Data.

    The *Add Data Providers to Plot* dialog opens.
14. In the dialog, expand Derived Data and select Power Spectra.



The plot of Power Spectral Density against Distance appears in the *Graphics* window.

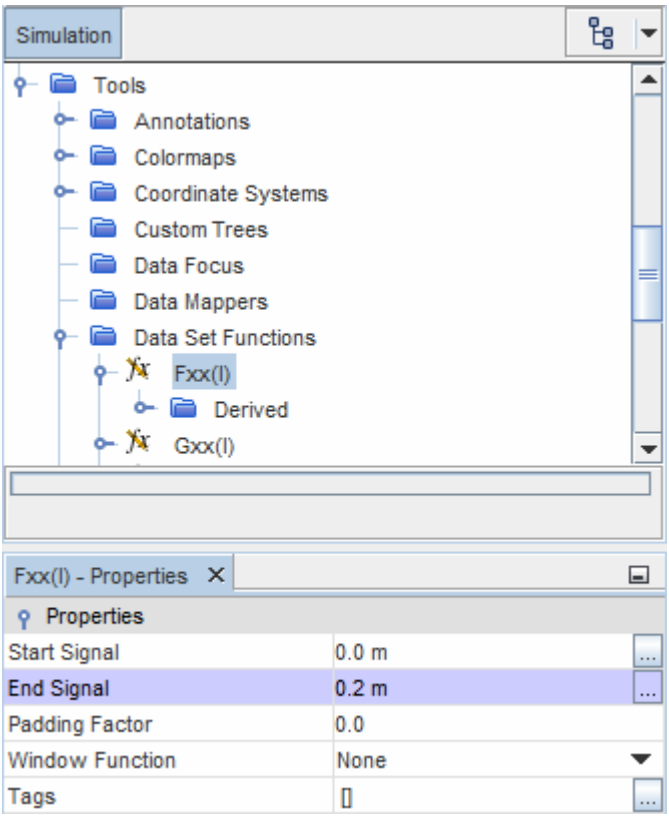


The expected profile is obtained. The average correlation is maximum at zero where the separation is minimum.

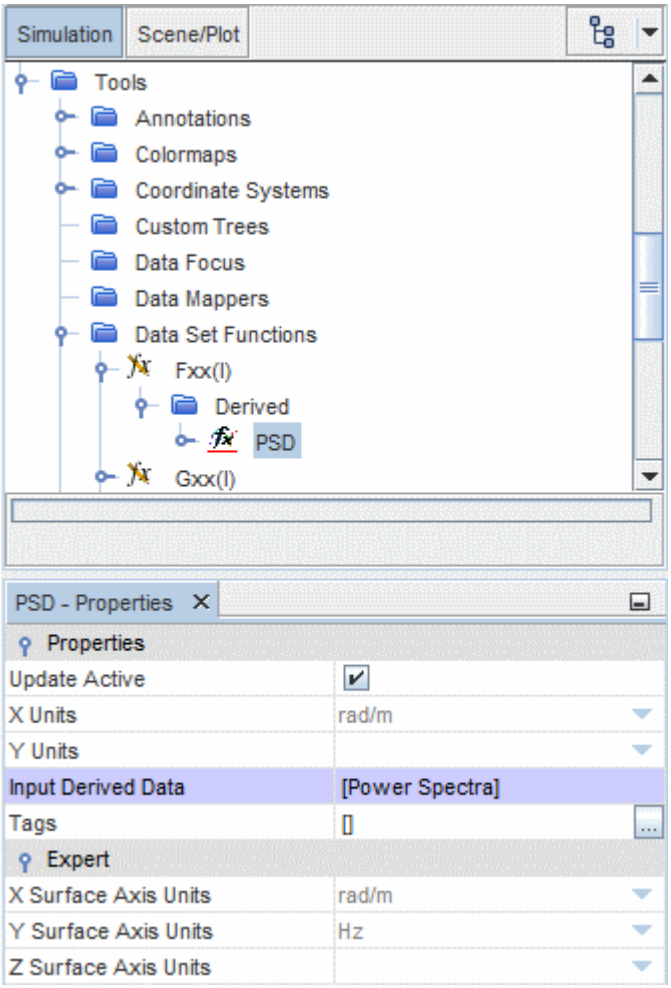
Calculating the Line Spatial Fourier Transform

Use the values in this range to create a line spatial Fourier transform. The velocity of propagation is calculated from this transform.

1. Right-click the **Tools** > **Data Set Functions**node and select **New** > **Line Spatial Fourier Transform (F[G[h[l]]])**.
2. Rename the newly created **F(G(l)) 1** node to `Fxx(l)`.
3. Select the **Data Set Functions** > **Fxx(l)** node and set *Start Signal* to `0.0 m`.
4. Set *End Signal* to `0.2 m`.



5. Right-click **Fxx(l)** > **Derived** and select **New derived data from derived data**.
6. Rename the newly created **Line Derived** node to `PSD`.
7. Select **Fxx(l)** > **Derived** > **PSD** and set *Input Derived Data* to **[Power Spectra]**.
8. Activate *Update Active*.



With the *Update Active* property activated, the line spatial Fourier transform is automatically calculated. This process takes a few minutes, depending on the processor speed of your computer.

When the Fourier transform has been computed, visualize the spectra:

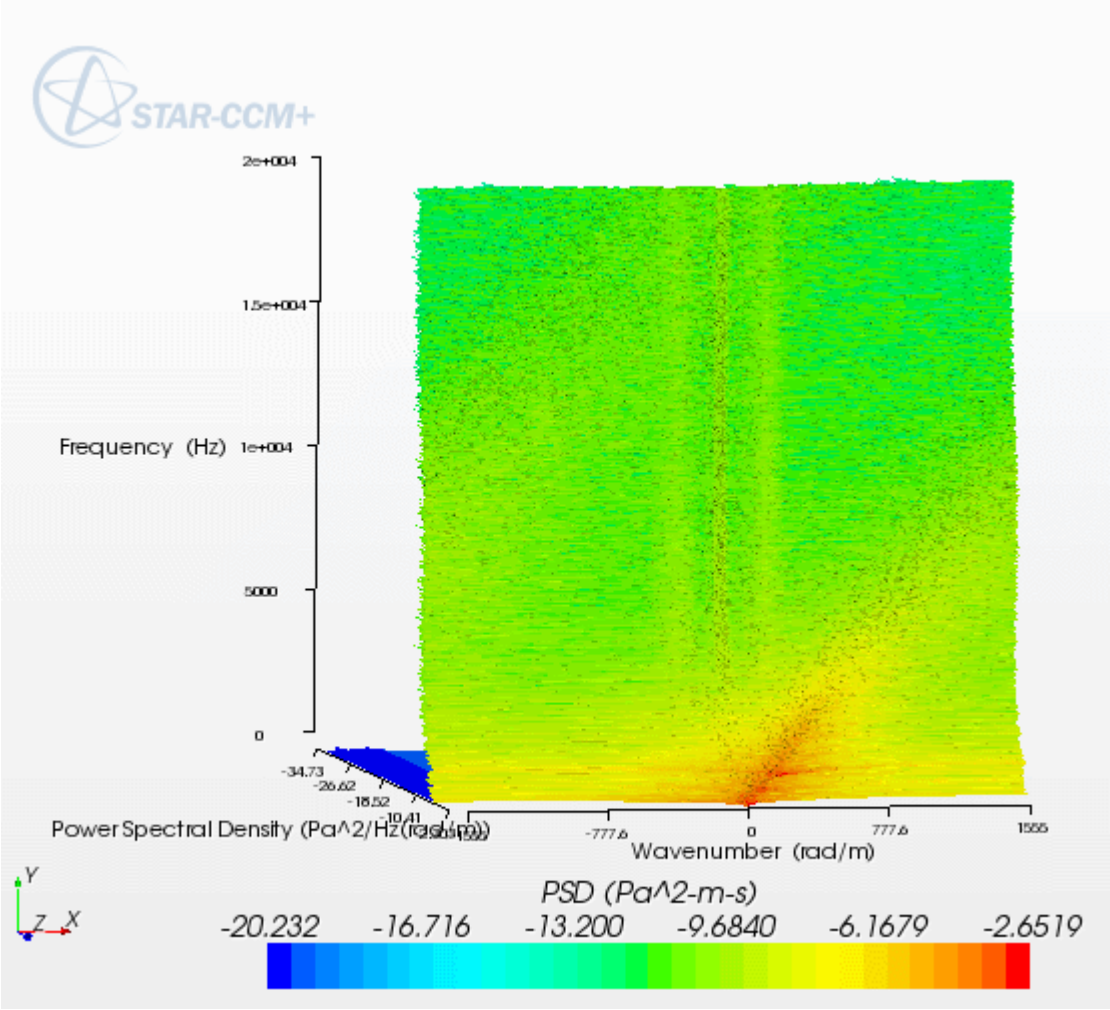
9. Select **PSD** > **Output control** and activate *Log data*.

Create a new scalar scene and rename it to `PSD`.

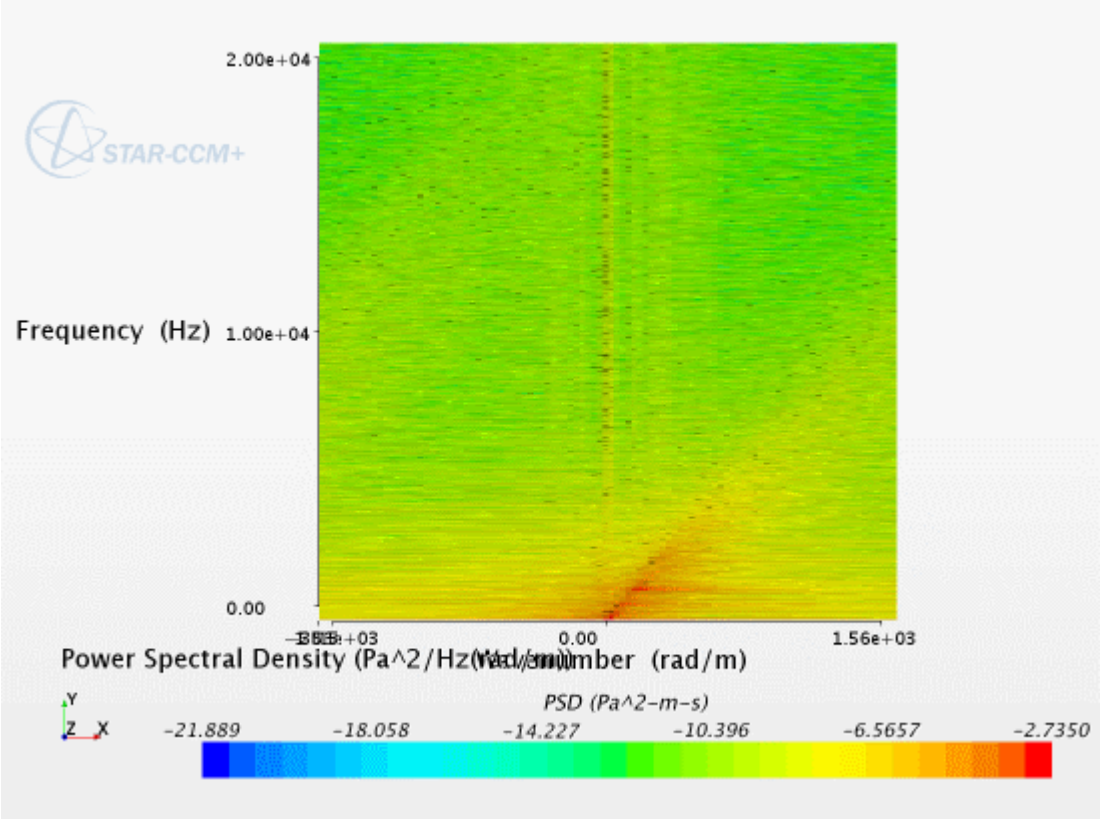
10. Select **PSD** > **Displayers** > **Outline 1** and deactivate *Outline*.
11. Select **PSD** > **Displayers** > **Scalar 1** > **Parts** and select **Derived Data > PSD**.
12. Click **OK**.

The power spectral density plot appears in the *Graphics* window. Use the mouse to adjust the view.

13. To color the plot, select **PSD** > **Displayers** > **Scalar 1** > **Scalar Field** and set *Function* to **Derived Data** > **PSD**.



14. To visualize the Power Spectral Density as a two-dimensional contour, click (**Save-Restore-Select views**) in the toolbar and select **Projection Mode** > **Parallel**.
15. To align with the X-Y plane, press the <T> key.



The slope of the distinctive line in the wavenumber-frequency plot is used to calculate the group velocity of the disturbances traveling along the domain. For this case, the group velocity is estimated as

$2\pi\frac{10,000}{1560}$ m/s = 40.3 m/s

which is, as expected, roughly 80% of the inlet velocity.

Summary

This tutorial covered post-processing for spectral and wavenumber analysis.

The tutorial introduced the following STAR-CCM+ features:

- Setting up a transient calculation.
- Modifying solver and stopping criteria parameters.
- Setting up monitoring report and plot.
- Saving transient surface data.
- Setting up spectral analysis.
- Post-processing transient surface data.

Acoustic Wave Modeling: Noise from a Cylinder

This tutorial demonstrates how to set up an acoustic wave simulation in STAR-CCM+. The flow around a cylinder is used in the simulation to describe the required steps.

The Acoustic Wave model uses the theory of Acoustic Perturbation Equations (APE) to predict the noise generated by the fluid flow (which is typically incompressible flow). The Acoustic Wave solver is one-way coupled with the flow solver through this APE-based noise source term, and models the acoustic pressure field inside a specified zone of interest. Using the Acoustic Wave model in a hybrid CFD/CAA simulation produces better results (with fewer spurious effects) than a pure compressible flow simulation.

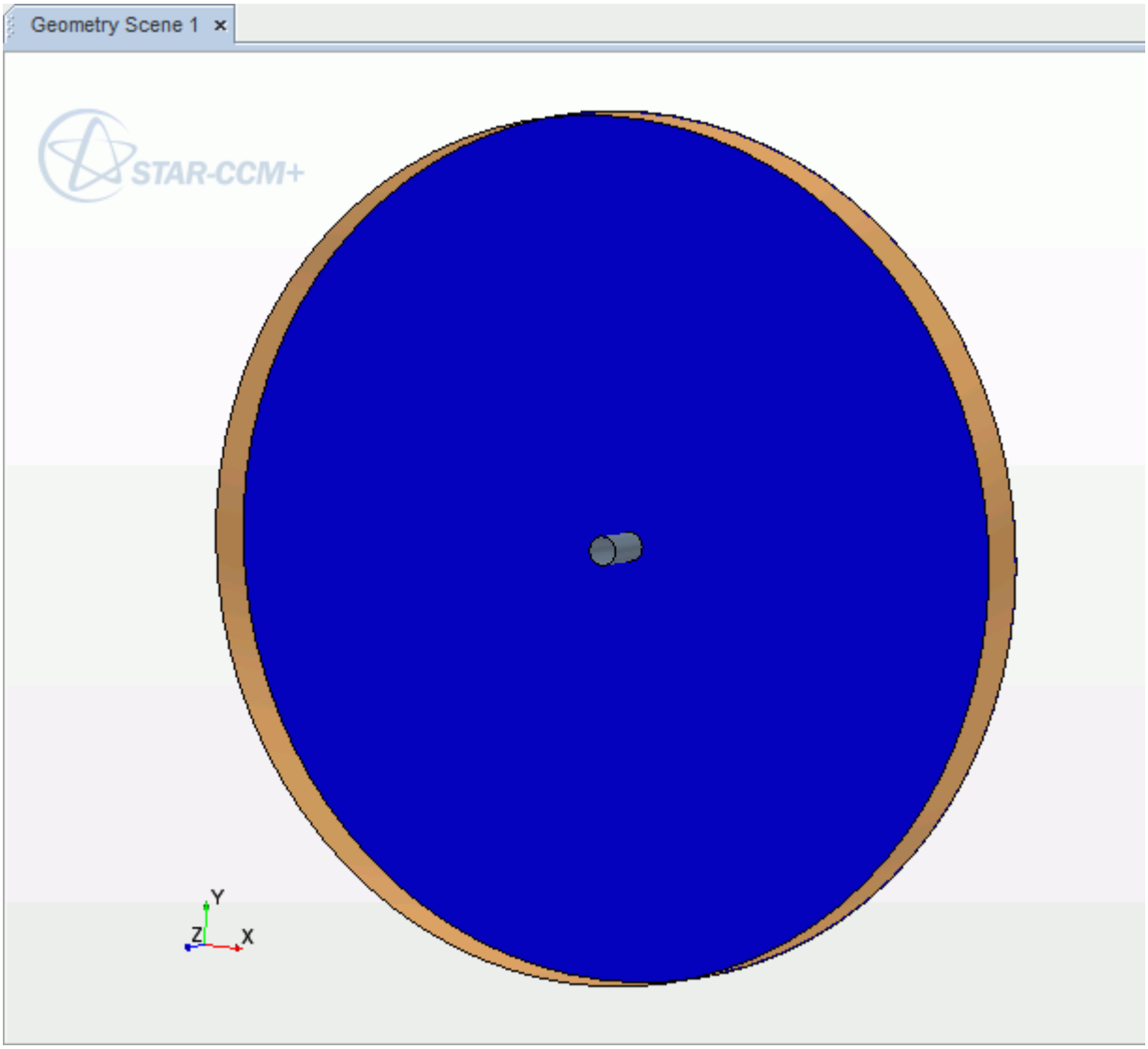
Note: The simulation in this tutorial is an example of an application where the flow past a bluff body causes the shedding of unsteady vortices. The Reynolds number in this tutorial, 6.38E4, is in the subcritical range 300 < Re <= 1.5E5. This indicates that the vortex shedding will be strong and periodic, exhibiting the familiar Karman vortex street. The onset of this phenomenon is inherently unpredictable.

The process for performing an acoustic wave simulation is as follows:

1. Perform a steady turbulent simulation to get an initial flow solution.
2. Perform an unsteady simulation to get a fully developed flow solution.
3. From the unsteady solution, compute the mean flow pressure.
4. Activate the Acoustic Wave model in the unsteady simulation.
5. Specify the noise source (the zone of interest in the flow) and the acoustic damping function to use in the acoustic wave simulation.
6. Run the acoustic wave simulation to obtain the acoustic pressure values throughout the flow region.

You can perform post-processing of the acoustic data to compare the simulation results against other data. STAR-CCM+ provides built-in data set functions, such as Fourier Transforms, that you can use to apply signal processing to the acoustic data.

The geometry that is used in this tutorial is shown below.



The geometry consists of a cylinder, 20 mm in diameter, placed at the center of a three-dimensional circular computational domain, 0.6 m in diameter. The air velocity is set to 50 m/s.

Contents:

Prerequisites
Loading the Base Simulation
Maintaining the Appropriate Acoustic CFL in the Wave Equation
Generating the Volume Mesh
Obtaining a Steady Solution
Obtaining an Unsteady Solution
Obtaining the Acoustic Wave Solution
Summary

Prerequisites

The instructions in the Acoustic Wave Solver tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Input Files

The input files that you require for this tutorial are in the `doc/startutorialsdata/aeroacoustics/data` subdirectory of your STAR-CCM+ installation directory. You are advised to copy the following input files to your working directory before you start this tutorial:

- `cylinderBase.sim` — contains the problem geometry.
- `CreateUserFieldFunctions.java` — sets up the field functions that you require to specify the acoustic noise source and damping regions.

Loading the Base Simulation

A base simulation file has been prepared for this analysis. Continue by starting the simulation, loading the base simulation file, and saving it under a new file name.

To load the base simulation:

1. Start up a double precision version of STAR-CCM+ in a manner that is appropriate to your working environment.
2. Load the base simulation file:
   a) Click (**Load Simulation**) from the tool bar.
   b) In the *Load Simulation* dialog, under **Process Options**, select **Parallel on Local Host**.

      You are strongly advised to run acoustic wave simulations in parallel. This tutorial case has a high number of cells and uses 10 inner iterations, so can be slow to run.
   c) In the **Compute Processes** box, set an appropriate number for the machine on which you are working.

      To make the run time reasonable, 10 processors are recommended.
   d) Navigate to your working directory and open `cylinderBase.sim`.
   e) Click **OK**.
3. Save the simulation to disk under file name `acoustic_wave.sim`.

Maintaining the Appropriate Acoustic CFL in the Wave Equation

To maintain the accuracy of acoustic predictions, apply criteria for points per wavelength (PPW) and the Acoustic CFL. This section outlines the requirements and calculations for this case.

The mesh size in the acoustic zone affects two criteria that are important for accurate acoustic predictions: the Acoustic CFL and the points per wavelength (PPW). The appropriate calculations for this tutorial are shown below.

Points Per Wavelength (PPW)

The recommended PPW for an acoustic calculation is PPW ≥ 30-40, maintained throughout the undamped zone. You can calculate the expected wavelength for this simulation from the vortex shedding frequency across the cylinder. This frequency comes from the Strouhal number, St:

$$St = \frac{fD}{v} \tag{4171}$$

Assuming a Strouhal number of 0.235 and rearranging the equation gives the expected frequency:

$$f = St\frac{v}{D} = 0.235 \times 50/0.02 = 587 \text{ Hz}$$

The wavelength in turn is calculated as c/f where c is the speed of sound (340 m/s). As $\text{PPW} = (c/f)/\Delta x$, the required mesh size for PPW of 40 is:

$$\Delta x = \frac{(340/587)}{40} = 0.0145 \text{ m}$$

Acoustic CFL

For the recommended PPW of 40, aim to have Acoustic CFL ≤ 4 in the zone of interest. Acoustic CFL is defined as:

$$\text{Acoustic CFL} = c\frac{\Delta t}{\Delta x} \tag{4172}$$

Rearranging and inserting a time-step Δt of 2.5e-5 s gives a candidate mesh size of:

$$\Delta x = 340 \times \left(2.5 \times 10^{-5}\right)/4 = 0.002125\text{m}$$

Overall Mesh Size

Taking the lower computed amount of 0.002125 m and rounding down gives a final target mesh size of 0.002 m in the acoustic zone.

Generating the Volume Mesh

The meshing models and mesh reference values for this simulation have already been set up. A volumetric control has been created for mesh refinement, to capture the separation behind the cylinder, and to capture the vortex shedding.

The mesh size is 0.008 m in the coarse zone and 0.002 m in the fine zone.

To generate the volume mesh:

1. In the **Mesh Generation** toolbar, click (**Generate Volume Mesh**).

To visualize the volume mesh:

2. Right-click the **Scenes** node and select **New Scene** > **Mesh**.

   The mesh appears as shown below.

Obtaining a Steady Solution

Run a steady turbulent simulation to get an initial flow solution. This solution is used as a starting point for the unsteady simulation.

The free-stream air velocity is set to 50 m/s.

To obtain an initial flow solution, you do the following:

1. Set up the steady physics continuum.
2. Set the appropriate initial conditions.
3. Set the appropriate boundary conditions.
4. Set the appropriate solver parameters and stopping criteria.
5. Run the steady simulation.

Contents:

Selecting the Physics Models for the Steady Solution
Setting the Initial Conditions
Setting the Boundary Conditions
Setting the Solver Parameters and Stopping Criteria for the Steady Solution
Running the Simulation for the Steady Solution

Selecting the Physics Models for the Steady Solution

Set up the physics models for the initial steady simulation.

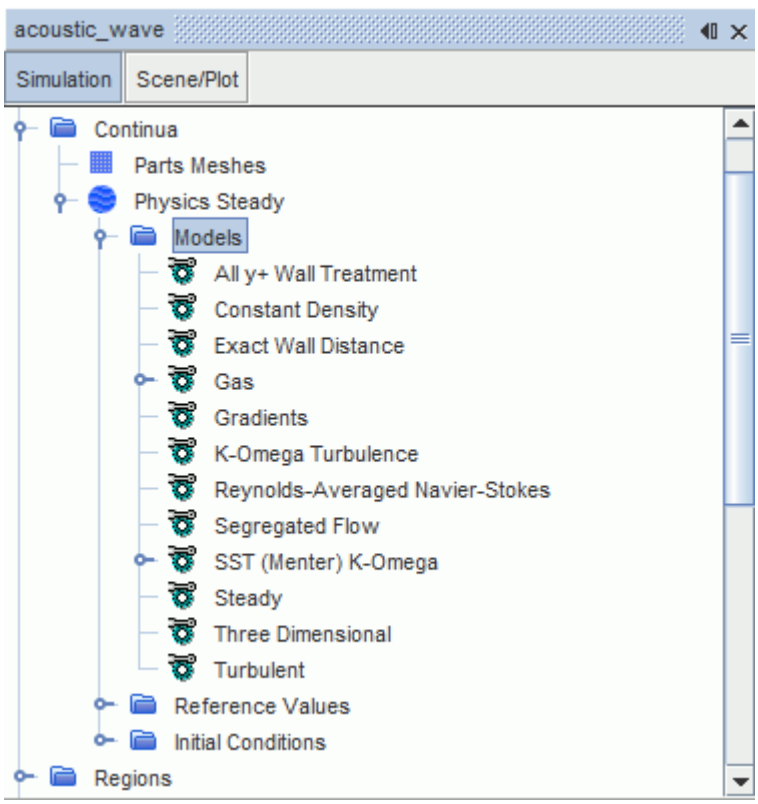
The result of the steady simulation provides a starting point for the next phase: an unsteady simulation from which you compute the mean flow pressure of the fully developed flow. To ensure good noise source predictions, the physics continuum requires an incompressible flow and a high-resolution turbulence model.

To select the physics models:

1. Rename the physics continuum, **Continua** > **Physics 1**, as **Continua** > **Physics Steady**.
2. For this physics continuum, select the following models in order:

Group Box	Model
Space	**Gradients** (Selected automatically) **Three Dimensional** (Selected automatically)
Time	**Steady**
Material	**Gas**
Flow	**Segregated Flow**
Equation of State	**Constant Density**
Viscous Regime	**Turbulent** **Reynolds-Averaged Navier-Stokes** (Selected automatically)
Reynolds-Averaged Turbulence	**K-Omega Turbulence** **SST (Menter) K-Omega** (Selected automatically) **Exact Wall Distance** (Selected automatically) **All y+ Wall Treatment** (Selected automatically)

3. Click **Close**.
4. To review the models, open the **Physics Steady** > **Models** node.



5. Save the simulation.

Setting the Initial Conditions

To enhance convergence, initialize the flow field with a constant velocity of 50 m/s.

To set the initial conditions for the **Physics Steady** continuum:

1. Edit the **Initial Conditions** node and set the following property:

Node	Property	Setting
Velocity	**Value**	`[50.0, 0.0, 0.0]`

   All other values remain at default.
2. Save the simulation.

Setting the Boundary Conditions

Set the flow velocity at the outer boundary; all other settings have acceptable default values.

To set the boundary condition:

1. Select the **Regions** > **Body 1** > **Boundaries** > **Freestream** node and rename it to **Outer**.
2. Select this node and set *Type* to **Velocity Inlet**.
3. Select the **Outer** > **Physics Values** node and set the following property:

Node	Property	Setting
Velocity Magnitude	**Value**	`50 m/s`

Setting the Solver Parameters and Stopping Criteria for the Steady Solution

Specify the Segregated Flow solver under-relaxation factors and set the appropriate stopping criteria for the steady simulation.

The under-relaxation factors for the Velocity and Pressure solvers are modified to enhance convergence.

To set the solver parameters and stopping criteria for the steady simulation:

1. Edit the **Solvers** node and set the following properties:

Node	Property	Setting
Segregated Flow		
└ **Velocity**	**Under-Relaxation Factor**	`0.9`
└ **Pressure**	**Under-Relaxation Factor**	`0.1`

2. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Steps	**Enabled**	Activated
	Maximum Steps	`500`

3. Save the simulation.

Running the Simulation for the Steady Solution

Preparation of the steady simulation is now complete, and you can run the simulation.

To run the steady simulation:

1. Click (**Run**) in the *Solution* toolbar.

   The solution progress is displayed in the *Output* window. The *Residuals* display is created automatically in the *Graphics* window and shows the progress of the solvers.

   During the run, you can stop the process by clicking (**Stop**) on the toolbar. If you do stop the simulation, you can click (**Run**) to resume it.

   If left alone, the simulation continues until 500 iterations are complete. When the last iteration is complete, the following message is displayed in the *Output* window:

   ```
   Stopping criterion Maximum Steps satisfied.
   ```

2. When the simulation has finished running, save it.

Obtaining an Unsteady Solution

Starting from the solution of the steady simulation, perform an unsteady simulation until the flow reaches a fully developed state.

The Acoustic Wave solver requires the mean of flow pressure in order to compute the instantaneous flow pressure fluctuations that are used in computing the noise source. You pre-compute the mean of flow pressure before you start the Acoustic Wave solver.

To compute the mean flow pressure, perform an unsteady simulation until the flow reaches a fully developed state. Use implicit time marching with sufficient inner iterations to ensure convergence.

To obtain a fully developed flow solution and compute the mean flow pressure:

1. Set up the unsteady physics continuum.
2. Set the appropriate solver parameters and stopping criteria.
3. Create a Mean of Pressure monitor.
4. Create a scalar scene to visualize the mean flow pressure for the fully developed flow.
5. Run the unsteady simulation.

Contents:

Selecting the Physics Models for the Unsteady Solution

Setting the Solver Parameters and Stopping Criteria for the Unsteady Solution
Visualizing the Mean Flow Pressure Field
Running the Simulation for the Unsteady Solution

Selecting the Physics Models for the Unsteady Solution

Create a physics continuum for the unsteady simulation and select the appropriate turbulence models.

To select the physics models:

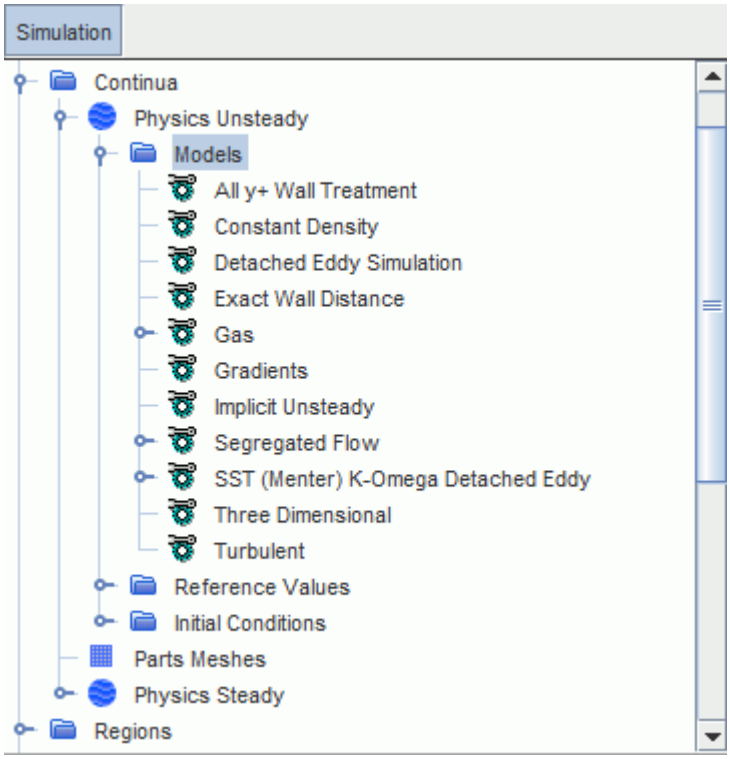
1. Make a copy of the physics continuum, **Continua** > **Physics Steady**, and rename the copy as **Continua** > **Physics Unsteady**.
2. For the physics continuum **Continua** > **Physics Unsteady**, deactivate the following models:

Group Box	Model
Enabled Models	**All y+ Wall Treatment** **Exact Wall Distance** **SST (Menter) K-Omega** **K-Omega Turbulence** **Reynolds-Averaged Navier-Stokes** **Steady** **Turbulent**

3. Deactivate the **Auto-select recommended models** option.
4. Select the following models in order:

Group Box	Model
Time	**Implicit Unsteady**
Viscous Regime	**Turbulent**
Turbulence	**Detached Eddy Simulation**
Detached Eddy Simulation	**SST (Menter) K-Omega Detached Eddy** **Exact Wall Distance** (Selected automatically)
K-Omega Wall Treatment	**All y+ Wall Treatment**

5. Click **Close**.
6. To review the models, open the **Physics Unsteady** > **Models** node.



7. Edit the **Regions** > **Body 1** node and set the following property:

Property	Value
Physics Continuum	**Physics Unsteady**

8. Save the simulation.

Setting the Solver Parameters and Stopping Criteria for the Unsteady Solution

Modify the Segregated Flow solver under-relaxation factors to values appropriate for an unsteady simulation. Also set the necessary stopping criteria.

To ensure convergence, use 10 inner iterations per time step.

To set the solver parameters and stopping criteria for the unsteady solution:

1. Edit the **Solvers** node and set the following properties:

Node	Property	Setting
Segregated Flow		
└ **Velocity**	*Under-Relaxation Factor*	`0.9`
└ **Pressure**	*Under-Relaxation Factor*	`0.7`
Implicit Unsteady	*Time-Step*	`2.5e-5 s`

Node	Property	Setting
	Temporal Discretization	**2nd-order**

2. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Inner Iterations	*Enabled*	Activated
	Maximum Inner Iterations	`10`
Maximum Physical Time	*Enabled*	Activated
	Maximum Physical Time	`0.01 s`
Maximum Steps	*Enabled*	Deactivated

3. Save the simulation.

Visualizing the Mean Flow Pressure Field

Create a scalar scene to visualize the mean flow pressure field from the fully developed flow solution.

First create a plane section:

1. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane**.
2. Edit the **Plane Section** node and set the following properties:

Property	Setting
Origin	`[0.0, 0.0, 0.02] m, m, m`
Normal	`[0.0, 0.0, 1.0] m, m, m`
Parts	**Regions** > **Body 1**

Next, create a Field Mean Pressure monitor:

3. Right-click the **Monitors** node and select **New Monitor** > **Field Mean**.
4. Edit the **Field Mean 1** node and set the following properties:

Property	Setting
Parts	**Regions** > **Body 1**
Field Function	**Pressure**

Finally, create a scalar scene to display the Mean of Pressure:

5. Right-click the **Scenes** node and select **New Scene** > **Scalar**.
6. Select the **Scenes** > **Scalar Scene 1** node and rename it as **Mean of Pressure**.
7. Edit the **Scenes** > **Mean of Pressure** > **Displayers** node and set the following properties:

Node	Property	Setting
Scalar 1	*Contour Style*	**Smooth Filled**

Node	Property	Setting
Parts	*Parts*	**Derived Parts** > **Plane Section**
Scalar Field	*Function*	**Mean of Pressure**

8. Save the simulation.

Running the Simulation for the Unsteady Solution

Preparation of the unsteady simulation is now complete, and you can run the simulation.

To run the unsteady simulation:

1. Click (**Run**) in the *Solution* toolbar.

   You can click the *Mean of Pressure* tab at the top of the *Graphics* window to view the development of the mean flow pressure field as the simulation progresses.

   The mean flow pressure reference is for all cells in the flow region.

   The simulation continues until 400 time-steps (which represents 0.01 seconds of physical time) are complete. The numbers that are displayed in the *Output* window represent the inner iterations of the solver, not time-steps. As there are 10 inner iterations per time-step, you expect the run to perform 4000 inner iterations.

   When the last iteration is complete, the following message is displayed in the *Output* window:

   ```
   Stopping criterion Maximum Physical Time satisfied.
   ```

2. When the simulation has finished running, save it.

Obtaining the Acoustic Wave Solution

Starting from the solution of the unsteady simulation (a fully developed flow solution), activate the Acoustic Wave model, specify the noise source region, and then run the simulation to compute the acoustic pressure.

You use the mean flow pressure that was computed in the unsteady simulation. This value is available as a field function.

This simulation models acoustic waves that are generated by turbulent flow. You specify the noise source with a field function that defines a volume of interest within the flow region. To suppress acoustic reflections from outside the noise source volume, you specify an acoustic damping function. Physical damping is required in this case, since the mesh becomes coarse away from the region of interest.

To obtain the acoustic wave solution, you do the following:

1. Activate the Acoustic Wave model.
2. Specify how the Mean Flow Pressure is obtained.
3. Create the appropriate field functions to define the noise source filtering and acoustic damping zones.

4. Set up the noise source to use in the Acoustic Wave model.
5. Create a scalar scene to visualize the acoustic pressure throughout the flow region.
6. Run the simulation with both the flow and the Acoustic Wave models activated.

Contents:

Setting Up the Acoustic Wave Model
Defining the Noise Source Region and Damping Zone
Setting the Solver Parameters and Stopping Criteria for the Acoustic Wave Solution
Visualizing the Velocity Magnitude
Visualizing the Acoustic Pressure
Running the Simulation for the Acoustic Wave Solution
Visualizing the Results

Setting Up the Acoustic Wave Model

To model the acoustic waves that are generated by the fluid flow in the physics continuum, you select the Acoustic Wave model and specify the appropriate acoustic perturbation-based noise source.

You specify the following:

- The method by which the acoustic waves are generated in the flow region.

  This simulation models the acoustic waves that are generated by the fluid flow in the physics continuum.
- The method by which the mean pressure field is obtained.

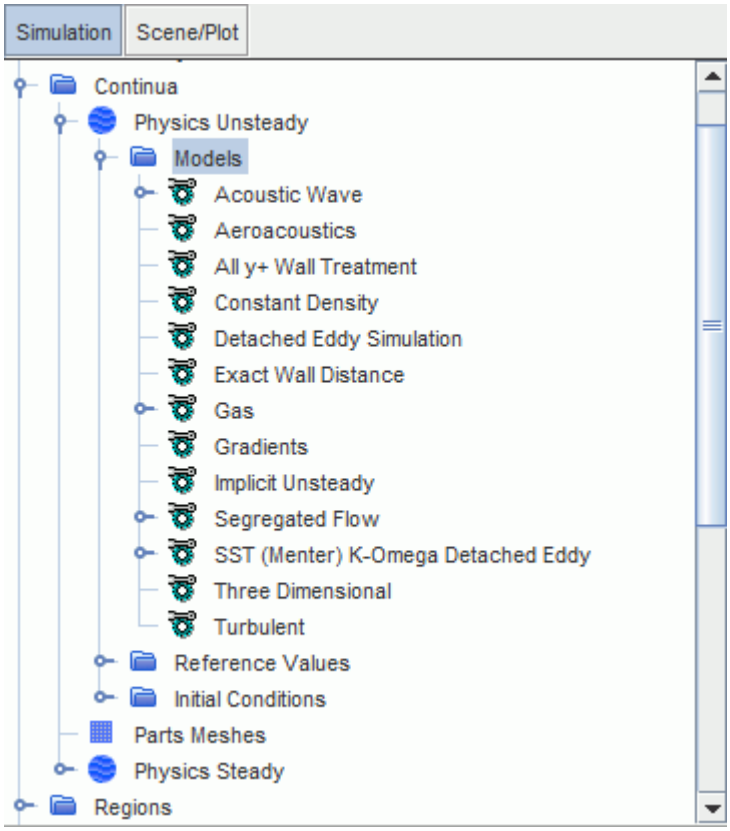
  This tutorial uses the Mean of Pressure field function that was calculated previously from the fully developed unsteady flow.

To set up the Acoustic Wave model:

1. For the physics continuum **Continua** > **Physics Unsteady**, select the following models in order:

Group Box	Model
Optional Models	**Aeroacoustics**
Aeroacoustics Models	**Acoustic Wave**

2. Click **Close**.
3. To review the models, open the **Physics Unsteady** > **Models** node.



4. Edit the **Continua** > **Physics Unsteady** > **Models** > **Acoustic Wave** node and set the following properties:

Node	Property	Setting
Acoustic Wave	*Noise Source*	**Acoustic Perturbation Based Source**
	Mean Flow Reference	**Specified**
└ Mean Flow Pressure Reference	*Method*	**Field Function**
	Scalar Function	**Mean of Pressure**

5. Save the simulation.

Defining the Noise Source Region and Damping Zone

When you use the Acoustic Wave model, you specify the noise source and the acoustic damping by setting the appropriate physics conditions and physics values of the flow region.

The Acoustic Wave model requires you to define field functions for two coefficients:

- The *Noise Source Weighting Coefficient*, which defines the space within which noise is generated.
- The *Acoustic Damping Coefficient*, which STAR-CCM+ uses to suppress acoustic reflections outside the noise source enclosure space.

In this tutorial, the `NoiseZone` and `DampingZone` field functions define these quantities, respectively. Both of these field functions use a Hanning window for the transition between the values of 0 and 1.

For convenience, in this tutorial, you run a macro to set up the appropriate field functions. The macro, `CreateUserFieldFunctions.java`, is available as an input file in the `doc/startutorialsdata/aeroacoustics/data` subdirectory of the STAR-CCM+ installation directory.

To set up the user field functions:

1. Select **File** > **Macro** > **Play Macro**.
2. In the *Open* dialog, navigate to your working directory and select `CreateUserFieldFunctions.java`.
3. Click **Open** to run the macro.

To specify the noise source region and the acoustic damping function:

4. Edit the **Regions** > **Body 1** > **Physics Conditions** node and set the following properties:

Node	Property	Setting
Acoustic Damping Option	*Acoustic Damping Option*	**Specified**
Noise Source Weighting Option	*Noise Source Region Option*	**Specified**

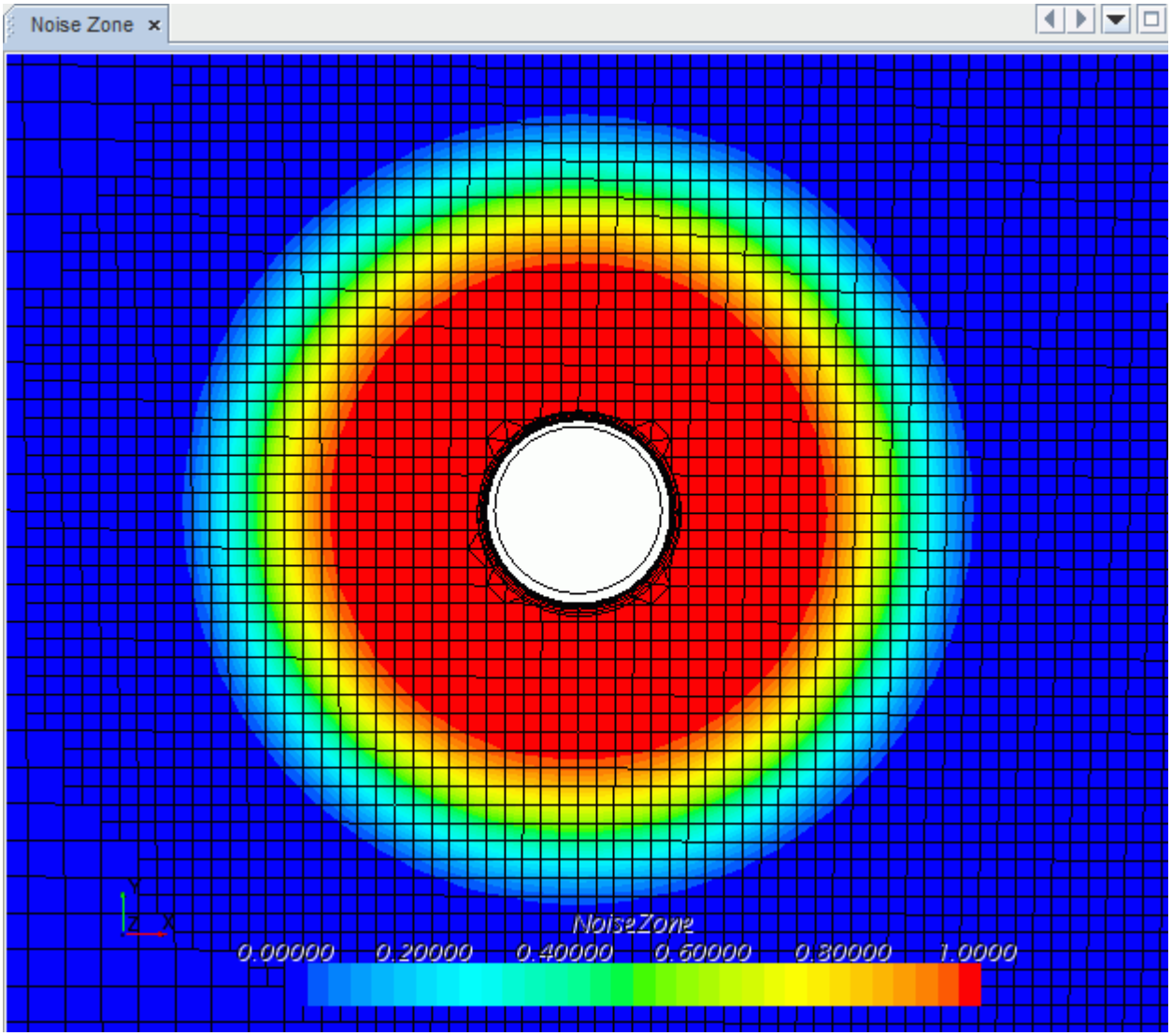
5. Edit the **Body 1** > **Physics Values** node and set the following properties:

Node	Property	Setting
Noise Source Weighting Coefficient	*Method*	**Field Function**
	Scalar Function	**NoiseZone**
Acoustic Damping Coefficient	*Method*	**Field Function**
	Scalar Function	**DampingZone**

6. Save the simulation.

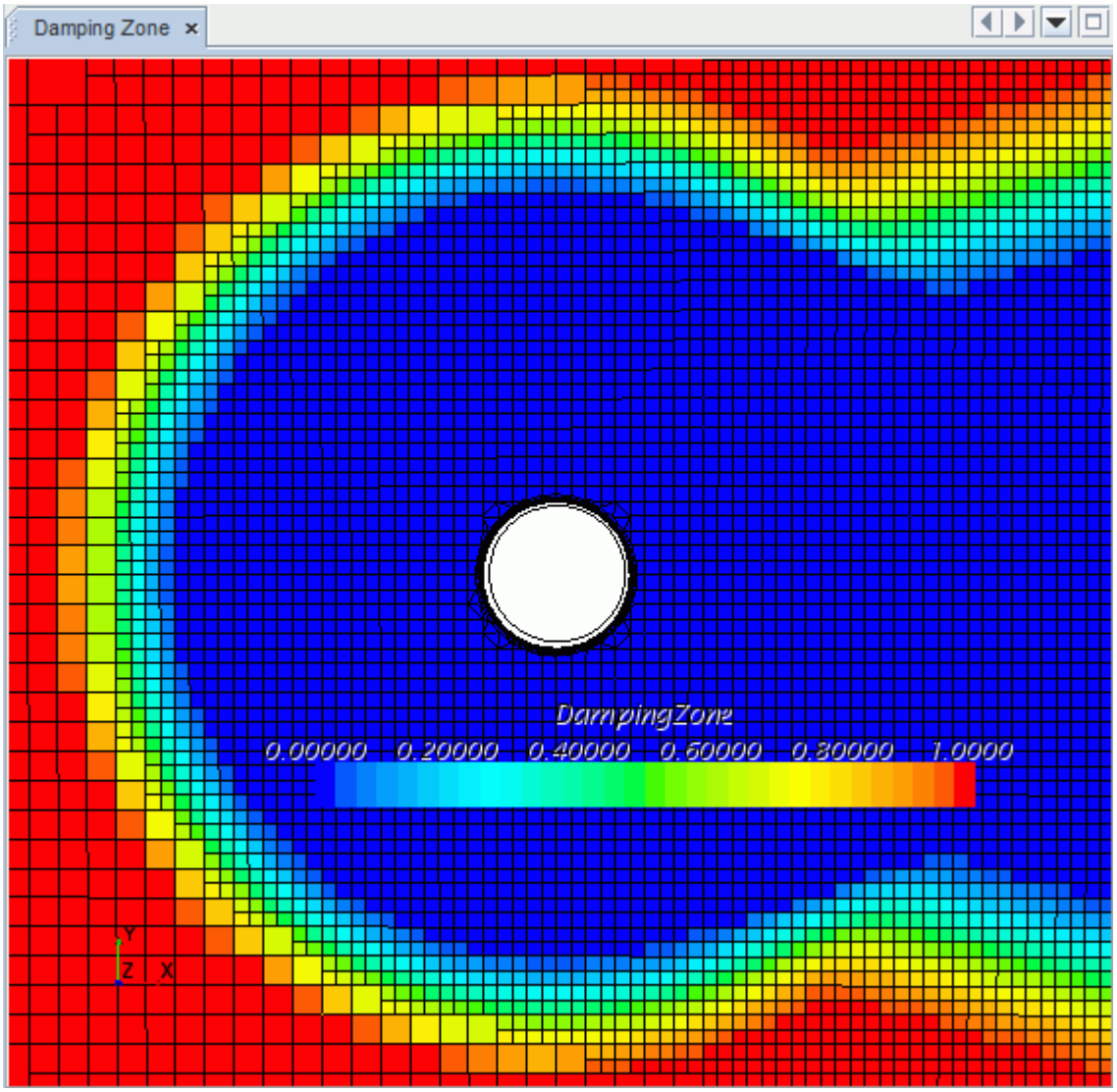
Visualizing the Noise Source Region and Damping Zone

The noise source region, as defined by the `NoiseZone` field function, is shown below:



The noise source is the region (in red) around the cylinder, where the airflow (moving from left to right) generates acoustic waves by vortex shedding from the cylinder. Note the fine mesh throughout the noise source zone.

The acoustic damping zone, as defined by the `DampingZone` field function, is shown below:



The region of interest (in blue) is undamped and everywhere else is fully damped (in red). Note the fine mesh throughout the undamped zone and the damping that is applied at the mesh transitions.

Field Function Definitions

The field functions that are used in this tutorial are tabulated below:

Function Name	Type	Definition
CombinedShape	**Scalar**	`(${JoinZones} > 1) ? 1 : ${JoinZones}`
DampingZone	**Scalar**	`1-${CombinedShape}`
dnSlope	**Scalar**	`${ydn_up}+($$Position[0]*tan(($${ydn_dn}-${ydn_up})/($${xdn}-${xup})))`

Function Name	Type	Definition
HanningCircle	**Scalar**	`1-(0.5*(1-cos($pi*($Radius-$Rm)/$thk)))`
HanningLong	**Scalar**	`($$Position[1] < ${upSlope}+${thkLong} && $$Position[1] > ${upSlope} ) ? 1-(0.5*(1-cos($pi*($$Position[1]+${thkLong}-${upSlope})/${thkLong}))) : 1-(0.5*(1-cos($pi*($$Position[1]+${thkLong}-${dnSlope})/${thkLong})))`
HanningSrc	**Scalar**	`1-(0.5*(1-cos($pi*($Radius-${Rsrc})/${thkSrc})))`
JoinZones	**Scalar**	`sqrt( (${ZoneLong}*${ZoneLong}) + (${ZoneCircle}*${ZoneCircle}) )`
NoiseZone	**Scalar**	`(${Radius} < ${Rsrc}) ? 1 : ((${Radius} > ${Rsrc} && ${Radius} < ${Rsrc}+${thkSrc}) ? ${HanningSrc} : 0)`
pi	**Scalar**	`4*atan(1)`
Radius	**Scalar**	`sqrt(($$Position[0]*$$Position[0])+($$Position[1]*$$Position[1]))`
Rm	**Scalar**	`0.05`
Rp	**Scalar**	`$Rm+$thk`
Rsrc	**Scalar**	`0.025`
thk	**Scalar**	`0.02`
thkLong	**Scalar**	`0.03`
thkSrc	**Scalar**	`0.02`

Function Name	Type	Definition
upSlope	**Scalar**	${yup_up}+($$Position[0]*tan(($${yup_dn}-${yup_up})/($${xdn}-${xup})))
xdn	**Scalar**	0.3
xup	**Scalar**	0.0065
ydn_dn	**Scalar**	-0.1
ydn_up	**Scalar**	-0.02
yup_dn	**Scalar**	0.1
yup_up	**Scalar**	0.02
ZoneCircle	**Scalar**	(${Radius} < ${Rm}) ? 1 : ((${Radius} > ${Rm} && ${Radius} < ${Rp}) ? ${HanningCircle} : 0)
ZoneLong	**Scalar**	($$Position[0] > ${xup}) ? ((abs($$Position[1]) > ${upSlope}+${thkLong}) ? 0 : ((abs($$Position[1]) < ${upSlope}) ? 1 : 1-${HanningLong})) : 0

Setting the Solver Parameters and Stopping Criteria for the Acoustic Wave Solution

Check that the solver parameters are correct and set the stopping criteria to give the simulation run another 0.005 seconds of physical time.

The Segregated Flow solver and Implicit Unsteady solver parameters are unchanged from the previous unsteady simulation. The default settings for the Acoustic Wave solver are acceptable for this simulation.

To set the solver parameters and stopping criteria:

1. Select the **Solvers** node and check that the following properties are set:

Node	Property	Setting
Acoustic Wave	*Start-time*	0.0 s

Node	Property	Setting
Newmark Delta Parameter	*Value*	`0.5`
Segregated Flow		
Velocity	*Under-Relaxation Factor*	`0.9`
Pressure	*Under-Relaxation Factor*	`0.7`
Implicit Unsteady	*Time-Step*	`2.5e-5 s`
	Temporal Discretization	**2nd-order**

2. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Physical Time	*Enabled*	Activated
	Maximum Physical Time	`0.015 s`

3. Save the simulation.

Visualizing the Velocity Magnitude

Create a scalar scene to display the velocity magnitude throughout the flow region.

To visualize the velocity magnitude:

1. Right-click the **Scenes** node and select **New Scene** > **Scalar**.
2. Select the **Scalar Scene 1** node and rename it as **Velocity**.
3. Edit the **Velocity** > **Displayers** node and set the following properties:

Node	Property	Setting
Scalar 1	*Contour Style*	**Smooth Filled**
Parts	*Parts*	**Derived Parts** > **Plane Section**
Scalar Field	*Function*	**Velocity** > **Lab Reference Frame** > **Magnitude**

4. Save the simulation.

Visualizing the Acoustic Pressure

Create a scalar scene to display the acoustic pressure throughout the flow region.

To visualize the acoustic pressure:

1. Right-click the **Scenes** node and select **New Scene** > **Scalar**.

2. Select the **Scalar Scene 1** node and rename it as **Acoustic Pressure**.
3. Edit the **Acoustic Pressure** > **Displayers** node and set the following properties:

Node	Property	Setting
Scalar 1	*Contour Style*	**Smooth Filled**
Parts	*Parts*	**Derived Parts** > **Plane Section**
Scalar Field	*Function*	**Acoustic Pressure**

4. Save the simulation.

Running the Simulation for the Acoustic Wave Solution

Preparation of the acoustic wave simulation is now complete, and the simulation can be run.

To run the acoustic wave simulation:

1. Click (**Run**) in the *Solution* toolbar.

   You can click the *Acoustic Pressure* tab at the top of the *Graphics* window to view the development of the acoustic pressure field as the simulation progresses.

   The simulation continues until 0.015 seconds of physical time is complete. When the last iteration is complete, the following message is displayed in the *Output* window:

   ```
   Stopping criterion Maximum Physical Time satisfied.
   ```

2. When the simulation has finished running, save it.

Visualizing the Results

When the simulation is complete, examine the acoustic pressure prediction.

The flow over the cylinder is illustrated by the velocity magnitude scene that is shown below:

The acoustic prediction (below) shows a field that is typically observed in laminar vortex shedding from a cylinder.

Summary

This tutorial demonstrated how to set up an acoustic wave simulation in STAR-CCM+, using the flow around a cylinder.

This tutorial demonstrated how a flow-generated noise source can be identified and propagated through the zone of interest using a hybrid CFD/CAA modeling technique. In this case, noise that is generated by flow over the cylinder is captured and the noise source information is passed onto the acoustic wave solver.

This tutorial has covered the following:

- Obtaining the mean flow pressure from an unsteady flow solution.
- Selecting the Acoustic Wave model.
- Specifying the noise source zone to enclose all of the important noise sources that are generated in the flow.
- Minimizing spurious acoustics: specifying the acoustic damping zone to ensure damping in the fine-to-coarse mesh transition regions away from the zone of interest.
- Setting the solver parameters and stopping criteria.
- Displaying the acoustic pressure in a scalar scene.

Electromagnetism

The following tutorials illustrate features for solving problems that involve an electric field.

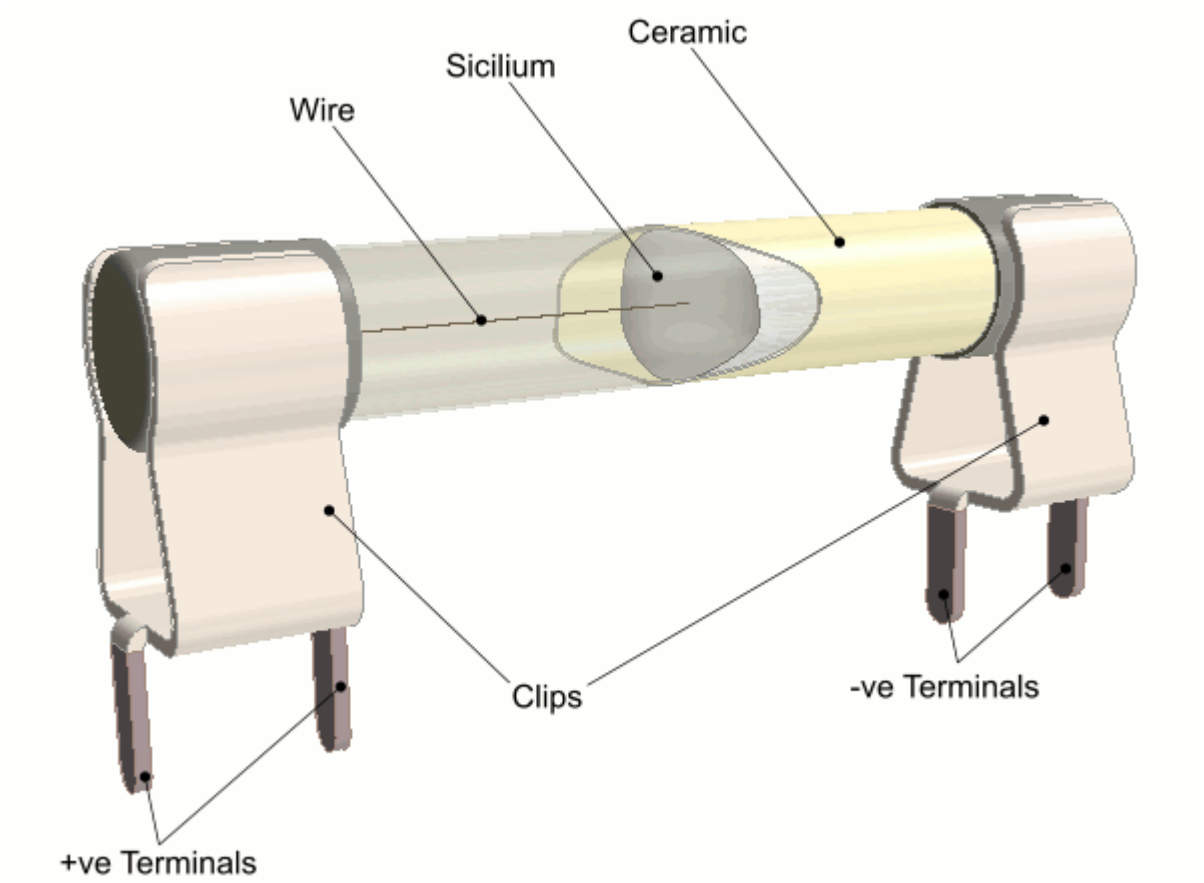
Contents:

Ohmic Heating: Domestic Fuse
Magnetic Vector Potential: Magnetic Flux

Ohmic Heating: Domestic Fuse

This tutorial demonstrates some of the capabilities of the Ohmic Heating model available in STAR-CCM+.

A simple domestic fuse geometry is used in the simulation to demonstrate the required steps. The geometry is shown below.



A fuse is a safety device to protect electrical equipment against unexpected high currents. Because of the Joule effect, the current going through the thin wire inside the fuse increases its temperature. If the current hits the critical current, the wire temperature exceeds its melting point, and breaks.

The fuse assembly in this tutorial is 30 mm long and is composed of metal, ceramic, and silicium. The wire inside the ceramic enclosure is 27.5 µm in diameter and has a critical current of 7 A with ±0.7% error.

A CAD model has been prepared in which various parts are imprinted correctly on adjacent parts. When this model is imported into STAR-CCM+ (as geometry parts), coincident faces are automatically detected and used to generate part contact data. The scenario being simulated is rather simple: to increase the potential difference between the two terminals of the wire, thus inducing a current increase until it reaches the critical point.

The post-processing tools in STAR-CCM+ are used to visualize the results.

Contents:

Prerequisites
Importing the Surface Mesh and Naming the Simulation
Examining the Imported Geometry
Increasing the Tessellation Density of the Fuse Wire
Assigning Geometry Parts to Regions
Setting up the Mesh
Adding a Volumetric Control
Generating the Mesh
Specifying Physics Models and Material Properties
Assigning Physics Continua
Setting Global Initial Conditions
Defining Boundary Conditions
Setting up Reports, Monitors, and Plots
Setting up a Plane Section
Setting Up a Scalar Scene
Setting Solver Parameters
Setting Stopping Criteria
Running the Simulation
Summary

Prerequisites

The instructions in the Ohmic Heating: Domestic Fuse tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	

Importing the Surface Mesh and Naming the Simulation

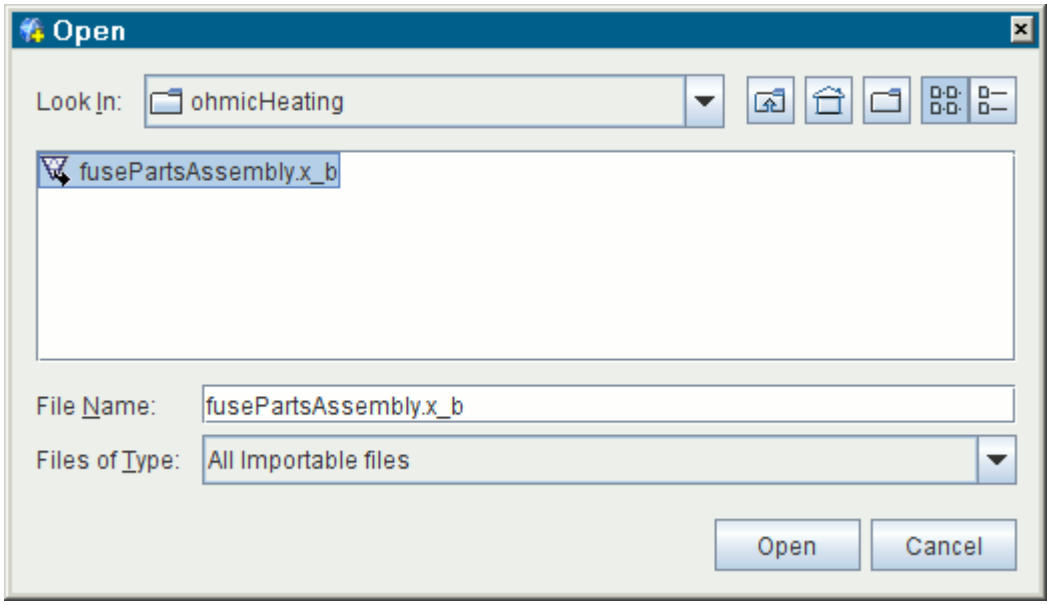
A CAD file containing the fuse assembly has been prepared for this analysis.

Begin by copying the file from the STAR-CCM+ installation directory, starting the simulation, importing the CAD model and saving the simulation.

1. Create a directory for the tutorial called `ohmicHeating`.
2. In this directory, copy the files containing the problem geometry (`fusePartsAssembly.x_b`) and the potential levels data (`potentialLevels.csv`) from the `doc/startutorialsdata/electromagnetism/data` subdirectory of your STAR-CCM+ installation directory.

3. Start up STAR-CCM+ in a manner that is appropriate to your working environment and create a New Simulation.

4. Click (**Import Surface Mesh**) from the tool bar.
5. In the *Open* dialog, select file `fusePartsAssembly.x_b`.



6. Click **Open** to start the import.
7. In the *Import Surface Options* dialog that appears, do the following:
   a) Select the **Create New Part** from the **Import Mode** box.
   b) Make sure that **Create Part Contacts from Coincident Entities** option is selected.
   c) Click **OK** to retain the default options.

   STAR-CCM+ provides feedback on the import process in the **Output** window. In particular, note the message `Created 8 new Part Contacts from 10 pairs of coincident faces.` A geometry scene is created in the **Graphics** window.
8. Finally, save the simulation to disk under file name `fuseWire.sim`.

Examining the Imported Geometry

The imported CAD parts, which represent the different parts of the fuse and its holder, are contained in the **Geometry** > **Parts** manager node.

The various parts were previously imprinted on each other to provide pairs of matching faces on the surfaces that are in contact. STAR-CCM+ detects the matching faces on import, and creates part/part contact data that are used later to define interfaces between regions.

These contacting surfaces can be viewed successively as follows:

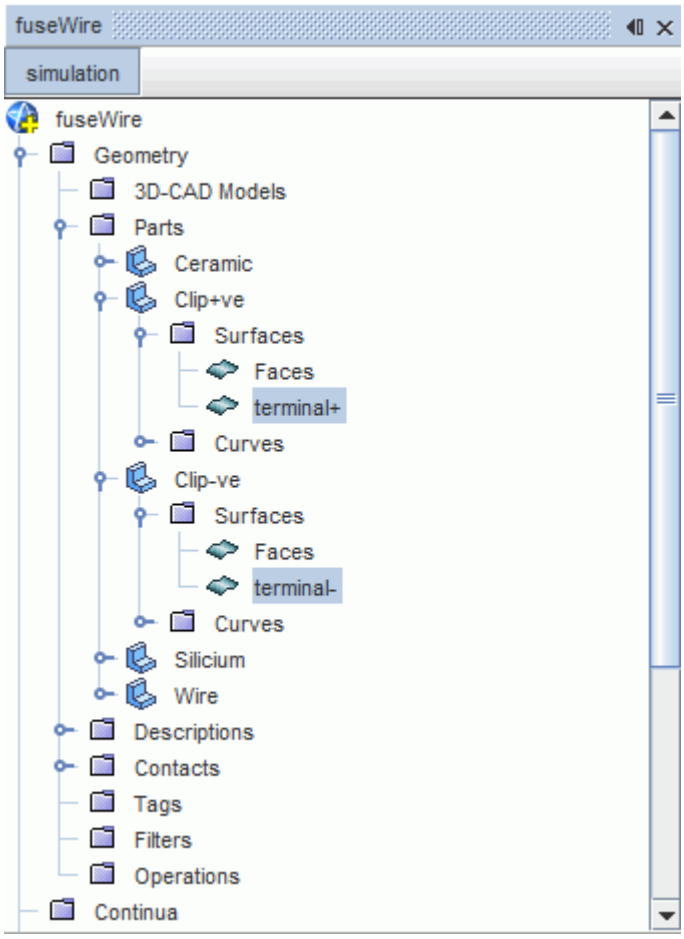
1. Right-click the **Geometry** > **Contacts** node and select **Browse Part/Part Contacts**.
2. In the *Part/Part Contact Browser Options* dialog, make sure that all the parts are selected and click **OK**.

The *Part/Part Contact Browser* dialog opens in the **edit** window and a scene in the **Graphics** window.

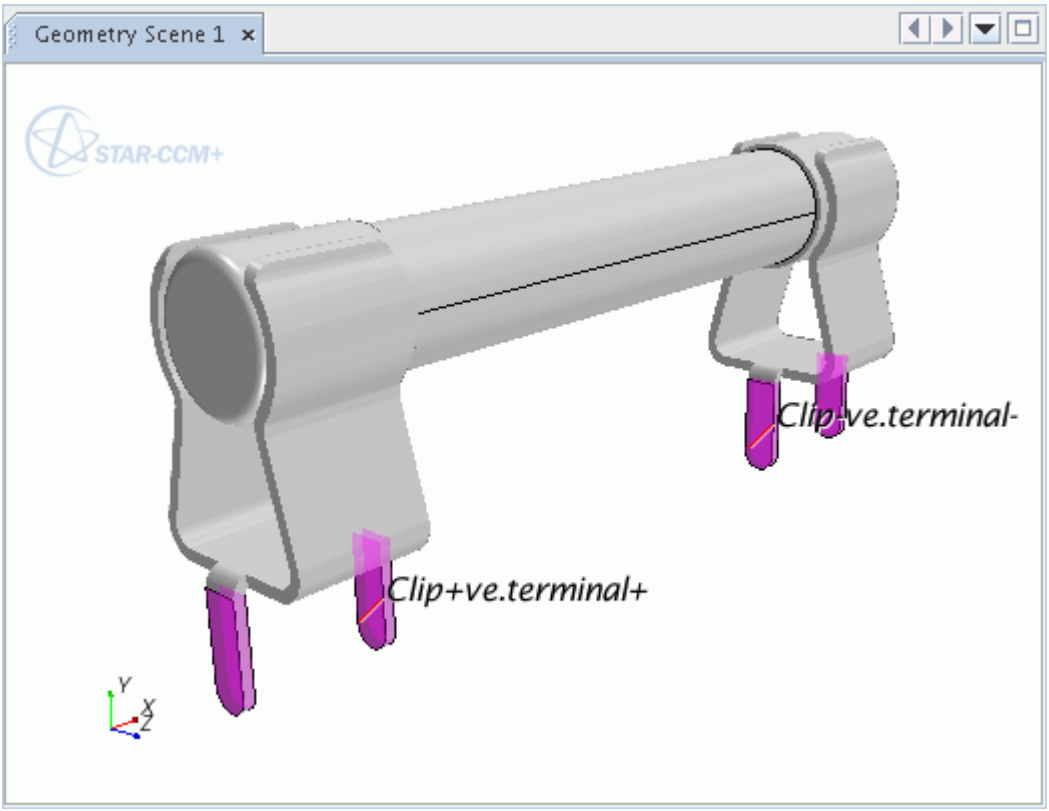
3. Select **Wire** from the **Source Parts** menu. Use the navigation buttons under the **Target Parts** menu to step through the **Clip+ve**, **Clip-ve**, and **Silicium** parts with which the wire is in contact.
4. Click **Close**.

The surfaces and feature curves of geometry parts are stored as individual entities on which operations can be performed if necessary. No further operations are required in this tutorial, but it is instructive to examine the surfaces for the electric potential.

5. Open the **Parts** > **Clip+ve** > **Surfaces** and **Parts** > **Clip-ve** > **Surfaces** nodes. Multi-select the **terminal+** and **terminal-** nodes.



These surfaces are highlighted in the **Graphics** window.



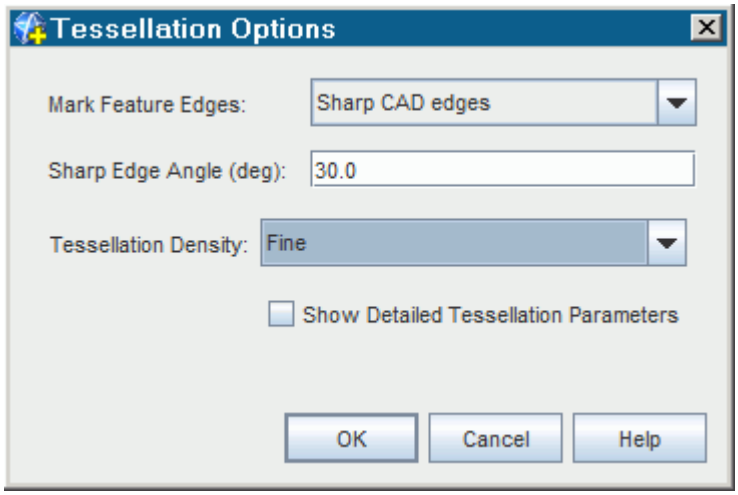
6. Use the mouse buttons to zoom in and position the part so that the view appears as shown.

7. Click (**Save-Restore-Select Views**) in the *Vis* toolbar and select **Store Current View**.

Increasing the Tessellation Density of the Fuse Wire

To ensure that the thin fuse wire has a well-defined surface, increase the tessellation density of the imported surface to represent the surface with a higher density of triangles.

1. Right-click the **Geometry** > **Parts** > **Wire** node and select **Re-tessellate**.
2. In the *Tessellation Options* dialog, set **Tessellation Density** to **Fine**.



3. Click **OK**.

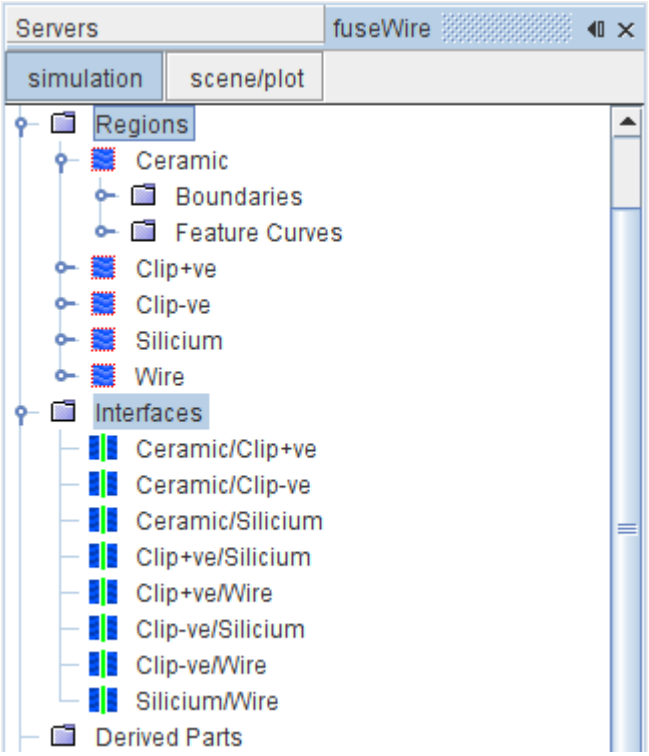
Assigning Geometry Parts to Regions

Assign geometry parts to regions before surface or volume meshing takes place. STAR-CCM+ allows you to create regions and boundaries that are based on the parts and their surfaces.

1. Multi-select all of the parts. Right-click the selected nodes and select **Assign Parts to Regions**.
2. In the *Assign Parts to Regions* dialog:
   a) Set the Region mode to **Create a Region for Each Part**.
   b) Set the Boundary mode to **Create a Boundary for Each Part Surface**.
   c) The Feature Curve mode can be left at **Create One Feature Curve for All Part Curves**. The **Create Boundary-mode Interfaces From Contacts** option remains selected.
   d) Click **Apply** and then **Close**.

The necessary regions and boundaries are created, along with interfaces corresponding to the part/part contact information reviewed earlier.

3. Expand the **Regions** and **Interfaces** nodes.



There are eight interfaces corresponding to the eight sets of part/part contact data.

Setting up the Mesh

Use the surface remesher with the polyhedral mesher to create the volume mesh.

To create the mesh:

1. Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
   a) From the *Parts* list, select all parts.
   b) Select the following meshers:

Group Box	Model
Surface Meshers	**Surface Remesher**
Volume Meshers	**Polyhedral Mesher**
Optional Surface Meshers	**Automatic Surface Repair**

   c) Click **OK**.
2. Edit the **Automated Mesh** > **Default Controls** node and set the following properties:

Node	Property	Setting
Base Size	*Value*	`1 mm`
Minimum Surface Size	*Percentage of Base*	`25`
Surface Curvature		

Node	Property	Setting
⌊ **Basic Curvature**	*# Pts/Circle*	`72.0`
Mesh Density	*Growth Factor*	`1.5`

To have adequate resolution on the surface of the fine wire, set a custom surface size on the interface between the wire and the surrounding solid.

3. Right-click **Automated Mesh** > **Custom Controls** and select **New** > **Surface Control**.
4. Select the **Custom Controls** > **Surface Control** node and set *Part Surfaces* to **Wire** > **Surfaces** > **Faces**.
5. Expand the **Surface Control** node and set the following properties:

Node	Property	Setting
Controls		
⌊ **Target Surface Size**	*Target Surface Size*	**Specify Custom Value**
⌊ **Minimum Surface Size**	*Minimum Surface Size*	**Specify Custom Value**
Values		
⌊ **Target Surface Size**	*Percentage of Base*	`5.0`
⌊ **Minimum Surface Size**	*Percentage of Base*	`2.5`

Adding a Volumetric Control

Add a volumetric control to provide adequate resolution of the volume mesh within the wire. When adding volumetric controls, it is first necessary to create the volume shape within which custom mesh control applies, and thereafter to select this shape within a volumetric control itself.

To add a volumetric control:

1. Create the Volume Shape:
   a) Right-click **Geometry** > **Parts** and select **New Shape Part** > **Cylinder**.
   b) In the *Create/Edit Cylinder Part* in place dialog, set the following:

Property	Setting
Start Coordinate	`[-0.01, 0.0065, 0.0] m`
End Coordinate	`[0.01, 0.0065, 0.0] m`
Cylinder Radius	`0.000028 m`

   c) Click **Create**, then **Close**.

   A **Parts** > **Cylinder** node appears in the tree.
2. Add the Volumetric Control:

a) Right-click the **Operations** > **Automated Mesh** > **Custom Controls** node and select **New** > **Volumetric Control**.

b) Rename the **Volumetric Control** node to `Cylinder`.

c) Select the **Custom Controls** > **Cylinder** node and set *Parts* to **Cylinder**.

In the simulation tree, the **Custom Controls** > **Cylinder** node changes color to show that a volume shape has been selected.

d) Expand the **Custom Controls** > **Cylinder** node and set the following:

Node	Property	Setting
Controls		
⌊ **Surface Remesher**	*Customize Size*	Activated
⌊ **Polyhedral Mesher**	*Customize Polyhedral Mesher*	Activated
Values		
⌊ **Custom Size**	*Percentage of Base*	`1.5`

Generating the Mesh

You can now create the volume mesh for the simulation.

To generate the volume mesh:

1. Click (**Generate Volume Mesh**) in the **Mesh Generation** toolbar.

   The mesh generation process takes some time depending on the speed of your computer. The output from the surface and volume meshers are shown in the *Output* window and the ongoing status is shown in the status bar.

2. Save the simulation .

Specifying Physics Models and Material Properties

You can now specify the physical models and the material properties for this simulation.

Three physics continua are required for this simulation:

- Copper (Ohmic Heating): For the Clip+ve, Clip-ve, and Wire regions through which electric current passes.
- Ceramic: For the Ceramic region, which is the outer casing of the central portion of the fuse.
- Silicium: Fills the annulus between the wire and the ceramic casing.

Contents:

Selecting Copper (Ohmic Heating) Physics Models
Specifying Copper (Ohmic Heating) Material Properties
Selecting Ceramic Physics Models

Specifying Ceramic Material Properties
Selecting Silicium Physics Models
Specifying Silicium Material Properties

Selecting Copper (Ohmic Heating) Physics Models

Define the models for the Copper (Ohmic Heating) continuum.

By default STAR-CCM+ has created a **Physics 1** continuum.

To select the physics models:

1. Rename the **Continua** > **Physics 1** node to `Copper (Ohmic Heating)`.
2. For the physics continuum, **Continua** > **Copper (Ohmic Heating)**, select the following models in order:

Group Box	Model
Space	**Three-Dimensional** (Selected automatically)
Time	**Steady**
Material	**Solid**
Optional Models	**Segregated Solid Energy**
	Gradients (Selected automatically)
Equation of State	**Constant Density**
Optional Models	**Electromagnetism**
Electromagnetism	**Electrodynamic Potential**
Optional Models	**Ohmic Heating**

3. Click **Close**.

Specifying Copper (Ohmic Heating) Material Properties

The properties of pure aluminum are set on a solid continuum by default. Replace this material with Copper.

To specify the material properties:

1. Right-click the **Models** > **Solid** > **Al** node and select **Replace with**.
2. In the *Replace Material* dialog, expand the **Material Databases** > **Standard** > **Solids** nodes and select **Cu (Copper)**.
3. Click **OK**.

Define a field function to relate the electrical conductivity of copper to its temperature.

4. Select the **Cu** > **Material Properties** > **Electrical Conductivity** node and set the **Method** to **Field Function**.

The electrical conductivity expression depends on the temperature as follows:

$$\text{Electrical Conductivity} = 26.08 \cdot 10^{9} \cdot T^{-27/25}$$

5. Right click the **Tools** > **Field Functions** node and select **New** > **Scalar**.
6. Rename the **User Field Function 1** node to `Copper Electrical Conductivity`.
7. Select the **Copper Electrical Conductivity** node and set the following properties:

Property	Setting
Dimensions	**ElectricConductance**
Function Name	`CopperElectricalConductivity`
Definition	`26.08e9 * pow($Temperature, (-27/25))`

8. Return to the **Electrical Conductivity** node, select **Field Function**, and set *Scalar Function* to **Copper Electrical Conductivity**.

Selecting Ceramic Physics Models

Define the models for the Ceramic continuum.

To select the physics models:

1. Create a physics continuum and rename the **Physics 1** node to `Ceramic`.
2. For the physics continuum, **Continua** > **Ceramic**, select the following models in order:

Group Box	Model
Space	**Three-Dimensional**
Time	**Steady**
Material	**Solid**
Optional Physics Models	**Segregated Solid Energy** **Gradients** (selected Automatically)
Equation of State	**Constant Density**

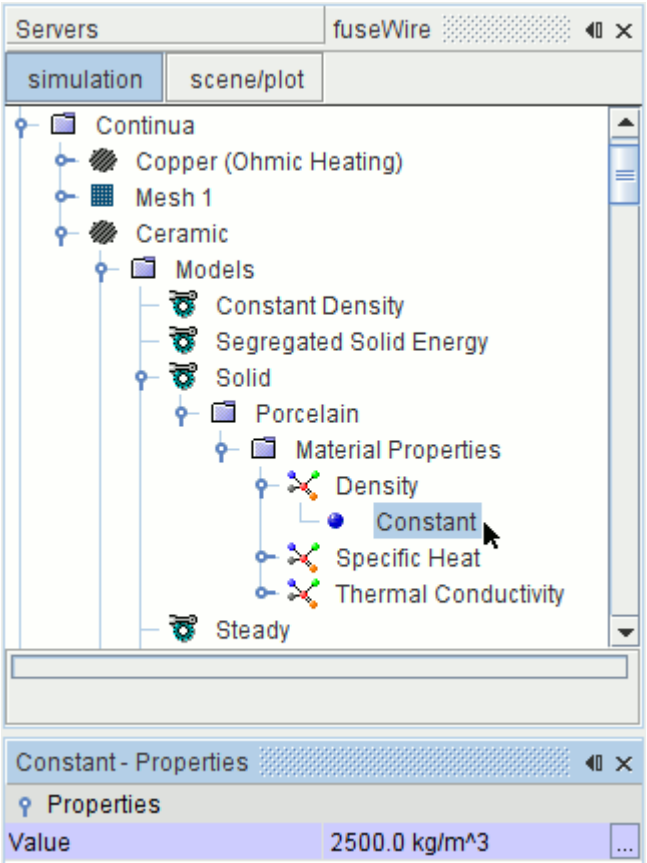
3. Click **Close**.

Specifying Ceramic Material Properties

You now specify the material properties for the ceramic.

To specify the material properties:

1. Rename the **Ceramic** > **Models** > **Solid** > **Al** node to `Porcelain`.
2. Select the **Porcelain** > **Material Properties** > **Density** > **Constant** node and set the **Value** to `2500 kg/m^3`.



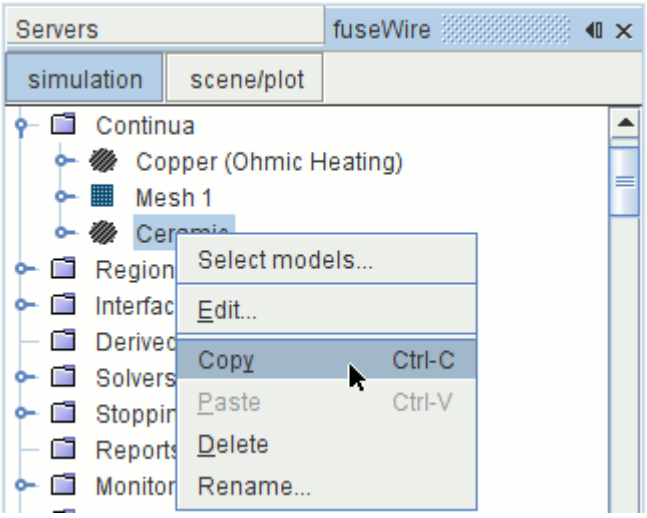
3. Select the **Specific Heat** > **Constant** node and set the **Value** to `1080 J/kg-K`.
4. Select the **Thermal Conductivity** > **Constant** node and set the **Value** to `0.928 W/m-K`.

Selecting Silicium Physics Models

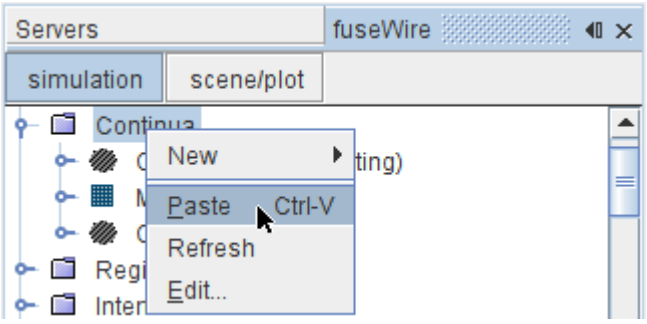
Define the models for the Silicium continuum by copying and pasting the Ceramic continuum.

To select the Silicium physics models:

1. Right-click the **Continua** > **Ceramic** node and select **Copy**.



2. Right-click the **Continua** node and select **Paste**.



An entry **Copy of Ceramic** is added to the object tree.

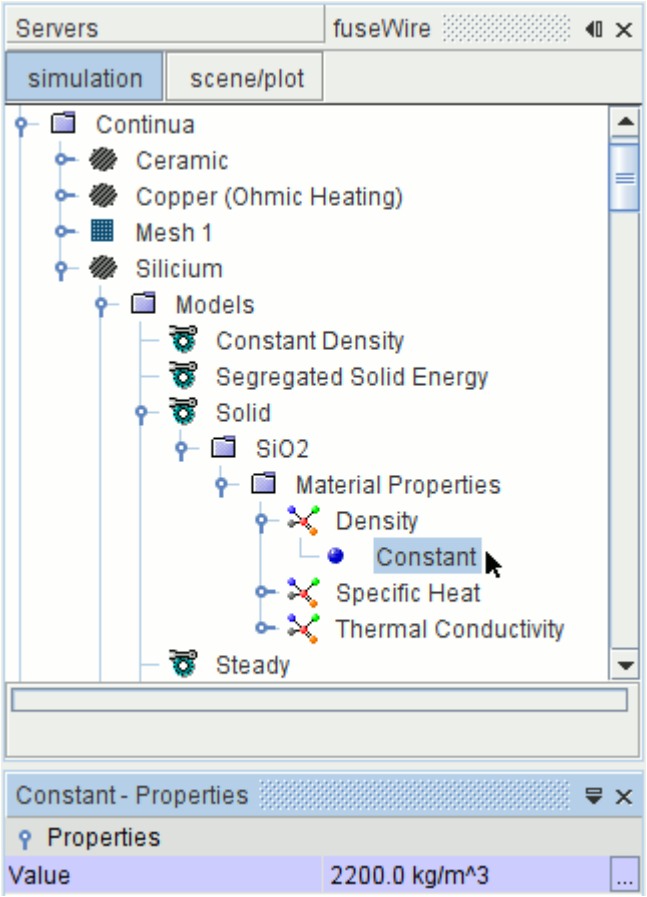
3. Rename the **Continua** > **Copy of Ceramic** node to `Silicium`.

Specifying Silicium Material Properties

You now specify the material properties for silicium.

To specify the material properties:

1. Rename the **Silicium** > **Models** > **Solid** > **Porcelain** node to `SiO2`.
2. Select the **SiO2** > **Material Properties** > **Density** > **Constant** node and set the **Value** to `2200 kg/m^3`



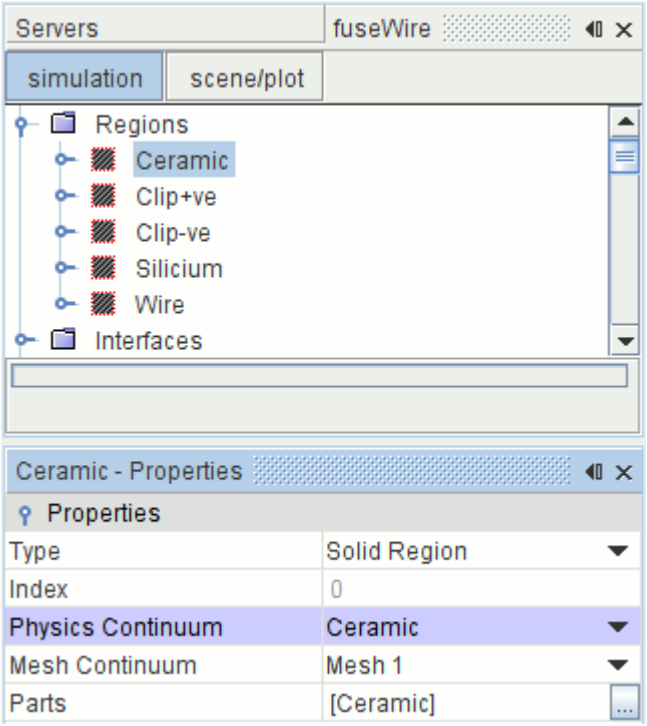
3. Select the **Specific Heat** > **Constant** node and set the **Value** to `740 J/kg-K`.
4. Select the **Thermal Conductivity** > **Constant** node and set the **Value** to `1.4 W/m-K`.

Assigning Physics Continua

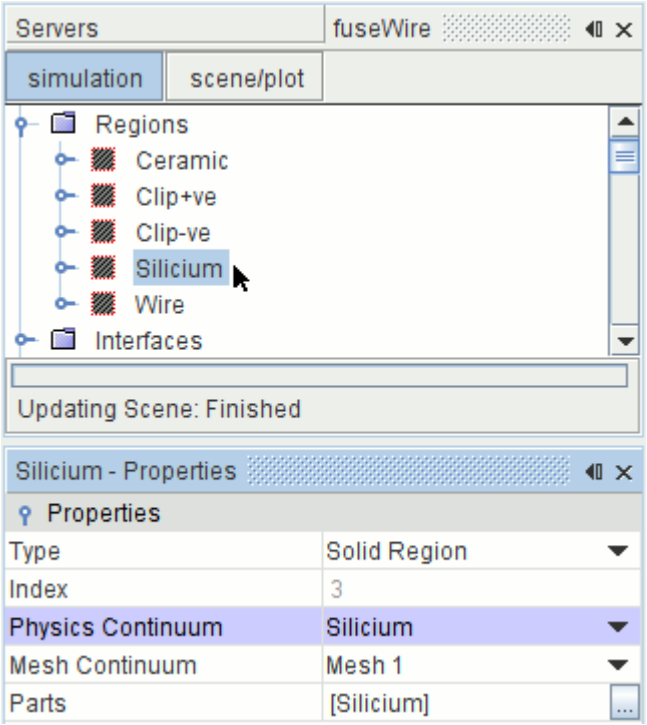
For a multi-region simulation, you must assign a physics continuum to each region. Through this assignment, the different physical models and material properties are assigned to the respective region.

To assign physics continua to regions:

1. Select the **Regions** > **Ceramic** node and set the *Physics Continuum* to **Ceramic**.



2. Select the **Regions** > **Silicium** node and set the *Physics Continuum* to **Silicium**.



The other three regions are associated with the Copper (Ohmic Heating) continuum by default.

3. Save the simulation .

Setting Global Initial Conditions

Specify the initial electrical potential value for the Copper (Ohmic Heating) model.

To set the initial conditions:

1. Select the **Continua** > **Copper (Ohmic Heating)** > **Initial Conditions** > **Electric Potential** node and set *Value* to `0.2 V`.

Defining Boundary Conditions

Set the boundary conditions for the Ceramic, Clip+ve, and Clip-ve regions.

To define the boundary conditions:

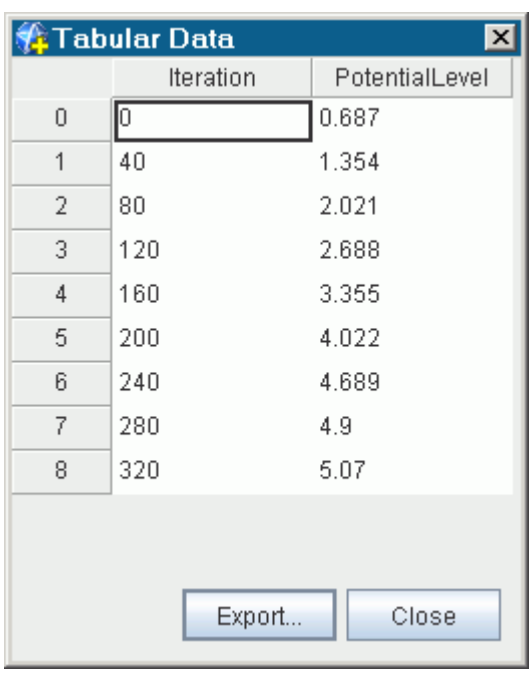
1. Select the **Regions** > **Ceramic** > **Boundaries** > **outerceramic** > **Physics Conditions** > **Thermal Specification** node and set the *Condition* to **Temperature**.
2. Select the **Physics Values** > **Static Temperature** node and set *Value* to `310` K.

Set the boundary conditions for the Clip+ve region to have a stepped electrical potential, read from a table of Iteration vs. Potential Levels. The table is supplied as a CSV file, `potentialLevels.csv`.

3. Right-click the **Tools** > **Tables** node and select **New Table** > **File Table**.
4. In the *Open* dialog, select `potentialLevels.csv` and click **Open**.

   potentialLevels is added to the object tree under the **Tables** node.
5. Right-click the **Tables** > **potentialLevels** node and select **Tabulate**.

   The content of the file is displayed in the *Tabular Data* dialog as shown below.



	Iteration	PotentialLevel
0	0	0.687
1	40	1.354
2	80	2.021
3	120	2.688
4	160	3.355
5	200	4.022
6	240	4.689
7	280	4.9
8	320	5.07

6. Click **Close**.

7. Select the **Regions** > **Clip+ve** > **Boundaries** > **terminal+** > **Physics Conditions** > **Electric Potential Specification** node and set the *Method* to **Electric Potential**.
8. Expand the **Physics Values** node and set the following:

Node	Property	Setting
Electric Potential	*Method*	**Table (iteration)**
∟ **Table (iteration)**	*Table*	**potentialLevels**
	Table: Data	**PotentialLevel**
	Interpolation	**Step**

9. Select the **Regions** > **Clip-ve** > **Boundaries** > **terminal-** > **Physics Conditions** > **Electric Potential Specification** node and set the *Method* to **Electric Potential**.
10. Select the **Physics Values** > **Electric Potential** node and set *Value* to `0.02 V`.
11. Save the simulation.

Setting up Reports, Monitors, and Plots

By default STAR-CCM+ produces a residuals plot which can be used to monitor the convergence of the solution. It is useful to create more monitors to help in assessing convergence.

Create two sets of reports, monitors and plots for the current and maximum temperature in the wire.

For the current-in-wire report, generate a plane section normal to the X direction using the wire as the input part. The plane section, is placed half way along the wire. It is used to create a surface integral of the current density in the Y direction (A/m^2).

1. Click the **Geometry Scene 1** tab in the **Graphics** window.
2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane**.

In the *Create Section* dialog that appears in the **edit** window, do the following:

3. In the **Input Parts** box, select **Regions** > **Wire**.
4. Set **Origin** to `[0.0, 0.00242566, 0.0] m` and **Normal** to `[1.0, 0.0, 0.0] m`.
5. Click **Create**, then click **Close**.

   A **Plane Section** derived part is created.
6. Rename the **Derived Parts** > **Plane Section** node to `Wire Cross-section`.
7. Right-click the **Reports** node and select **New Report** > **Surface Integral**.
8. Rename the **Surface Integral 1** node to `Current in Wire`.

Reports can be created for all derived parts and boundaries.

9. Select the **Reports** > **Current in Wire** node and set the following:

Property	Setting
Scalar Field Function	**Electric Current Density** > **Magnitude**
Parts	**Derived Parts** > **Wire Cross-section**

10. Click **OK**.

Create a report for the maximum temperature in the wire.

11. Right-click the **Reports** node and select **New Report** > **Maximum**.
12. Rename the **Maximum 1** node to `Maximum Temperature in Wire`.
13. Select the **Reports** > **Maximum Temperature in Wire** node and set the following *Properties*:

Property	Setting
Scalar Field Function	**Temperature**
Parts	**Regions** > **Wire**

Monitors and plots can be created from reports, and used to judge convergence while the simulation is running.

14. Multi-select the **Current in Wire** and the **Maximum Temperature in Wire** report nodes.
15. Right-click one of the selected nodes and select **Create Monitor and Plot from Report**.
16. Click **Multiple Plots (one per report)** to display the plots in separate graphs.

    Two monitors and plots are created.

    The **Current in Wire Monitor Plot** and **Maximum Temperature in Wire Monitor Plot** are opened automatically.

Setting up a Plane Section

You can use a plane section to visualize the temperature within the parts.

Generate a plane section normal to the Z direction using the ceramic, silicium, and wire as the input parts. The plane section is placed at the center of the fuse assembly. Use the plane in a scalar scene to display the temperature within the parts.

1. Open *Geometry Scene 1*.
2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane**.
3. In the *Create Plane Section* in-place dialog, set the following *Properties*:

Property	Setting
Parts	• **Regions** > **Ceramic** • **Regions** > **Silicium** • **Regions** > **Wire**
Origin	`[0.0, 0.0024209, 0.0] m`
Normal	`[0.0, 0.0, 1.0] m`

4. Click **Create**, then click **Close**.
5. Rename the **Plane Section** node to `Fuse Cross-section`.
6. Save the simulation.

Setting Up a Scalar Scene

To display the electric potential for the two clip regions and the temperature for the fuse cross-section derived part, you create a single scalar scene.

To create a scalar scene with two scalar displayers:

1. Create a Scalar Scene.
2. Expand the **Scalar Scene 1** > **Displayers** > **Scalar 1** node and set the following:

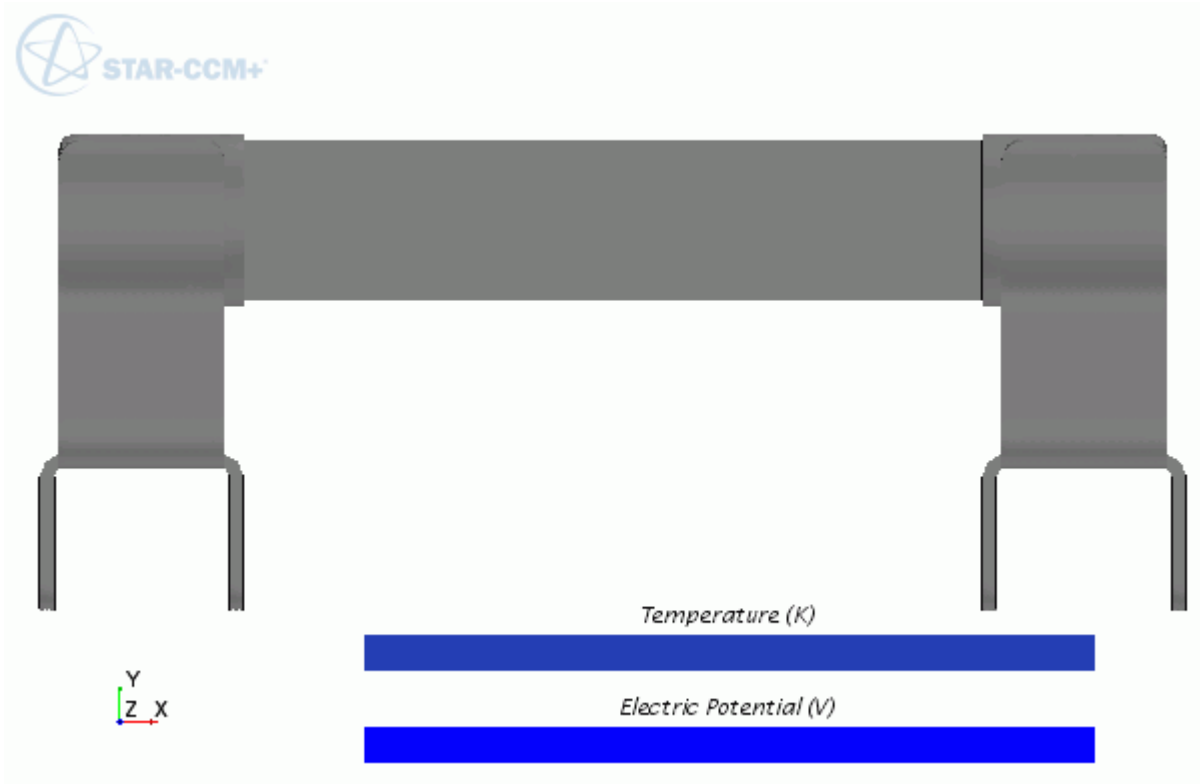
Node	Property	Value
Parts	*Parts*	**Regions** > **Clip+ve**
		Regions > **Clip-ve**
Color Bar	*Levels*	`2`
	Number of Labels	`3`
Scalar Field	*Function*	**Electric Potential**

Add another scalar displayer for the temperature of the fuse cross-section derived part:

3. Right-click the **Scalar Scene 1** > **Displayers** node and select **New Displayer** > **Scalar**.
4. Expand the **Displayers** > **Scalar 2** node and set the following:

Node	Property	Value
Parts	*Parts*	**Derived Parts** > **Fuse Cross-section**
Color Bar	*Color Map*	`cool-warm`
Scalar Field	*Function*	**Temperature**

5. Use the mouse buttons to zoom in and position the part as shown:



6. Save the simulation .

Setting Solver Parameters

You change the solver settings to help the solution converge quickly and maintain stability.

To set the solver parameters:

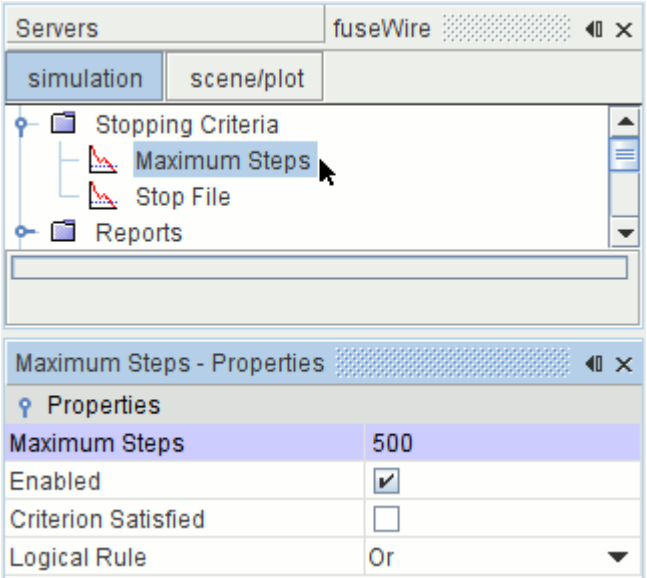
1. Expand the **Solvers** node.
2. Select the **Solvers** > **Segregated Energy** node and set the *Solid Under-Relaxation Factor* to `1` .

Setting Stopping Criteria

You now specify for how long the simulation runs.

Set the simulation to run for 500 iterations.

1. Select the **Stopping Criteria** > **Maximum Steps node** and set the *Maximum Steps* to `500`.



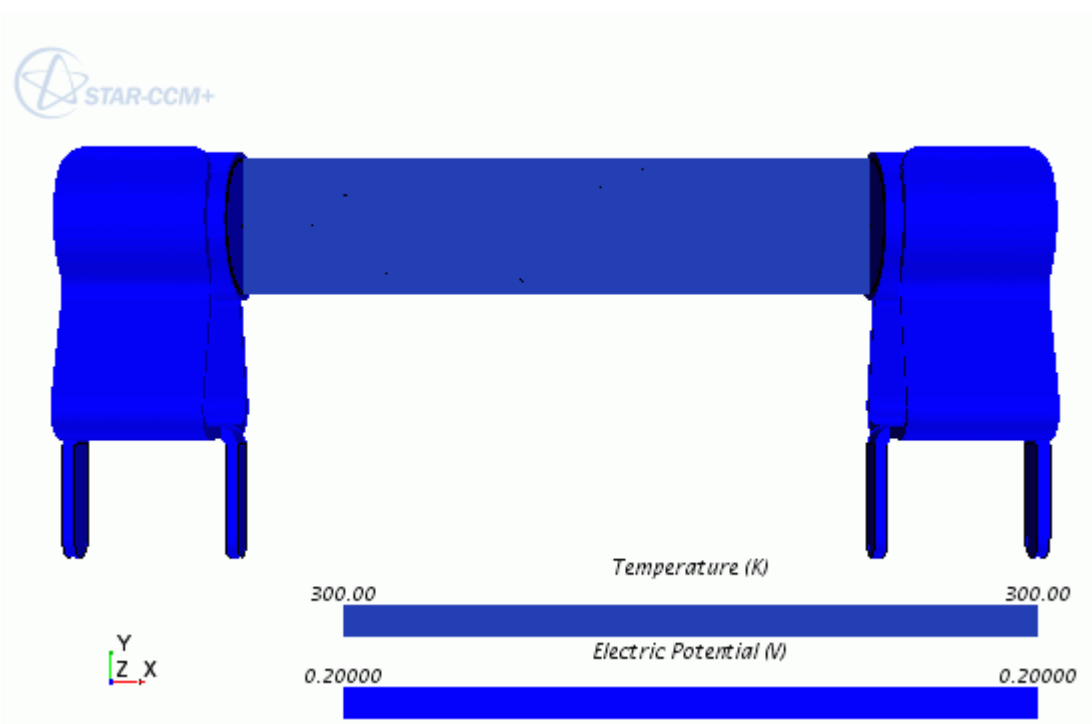
2. Save the simulation.

Running the Simulation

The simulation is now ready to be run.

To run the simulation:

1. To initialize the solution, click (**Initialize Solution**).
   The temperature and electric potential within the fuse assembly are initialized to the predefined values.



2. To run the simulation, click (**Run**).

   The iteration progress can be monitored in the *Output* window. You can monitor the convergence by viewing the Residuals using the visualization tabs in the *Graphics* window.

   When the last iteration has been completed, the message `Stopping Criterion Maximum Steps satisfied` is displayed in the *Output* window.

3. Save the simulation.

Contents:

Temperature and Electric Potential Plot
Report Plots

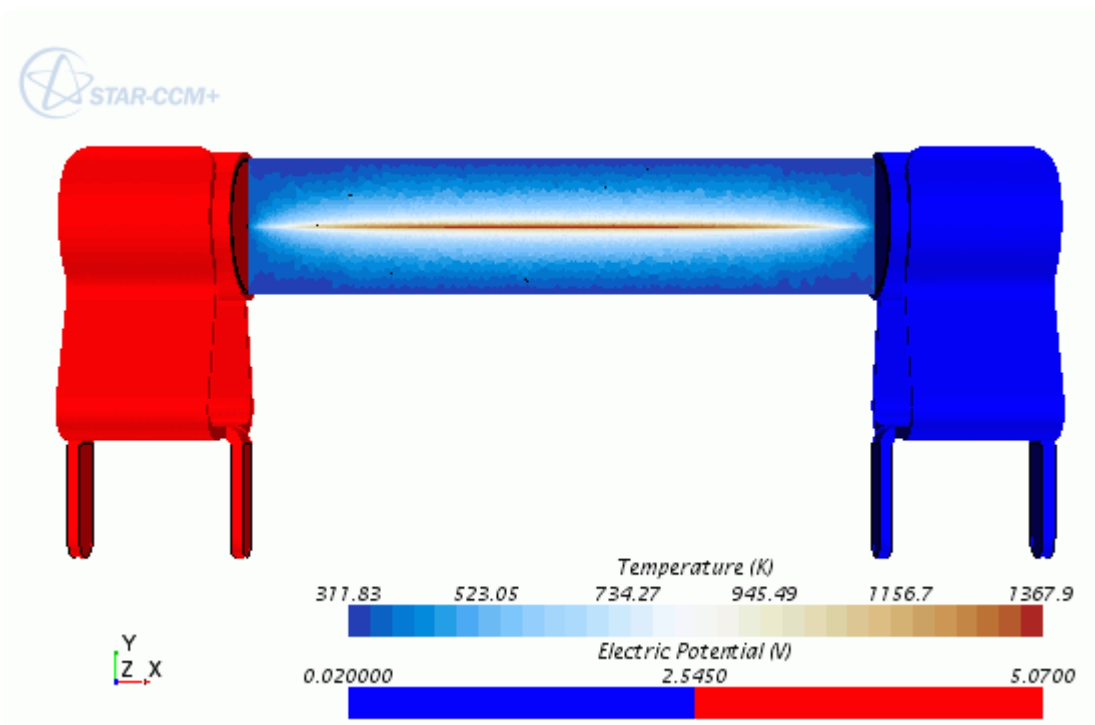
Temperature and Electric Potential Plot

You can now visualize the final temperature and electric potential distribution.

To visualize the temperature and electric potential plot:

1. Click the **Scalar Scene 1** tab at the top of the **Graphics** window.

   A scene appears as shown:

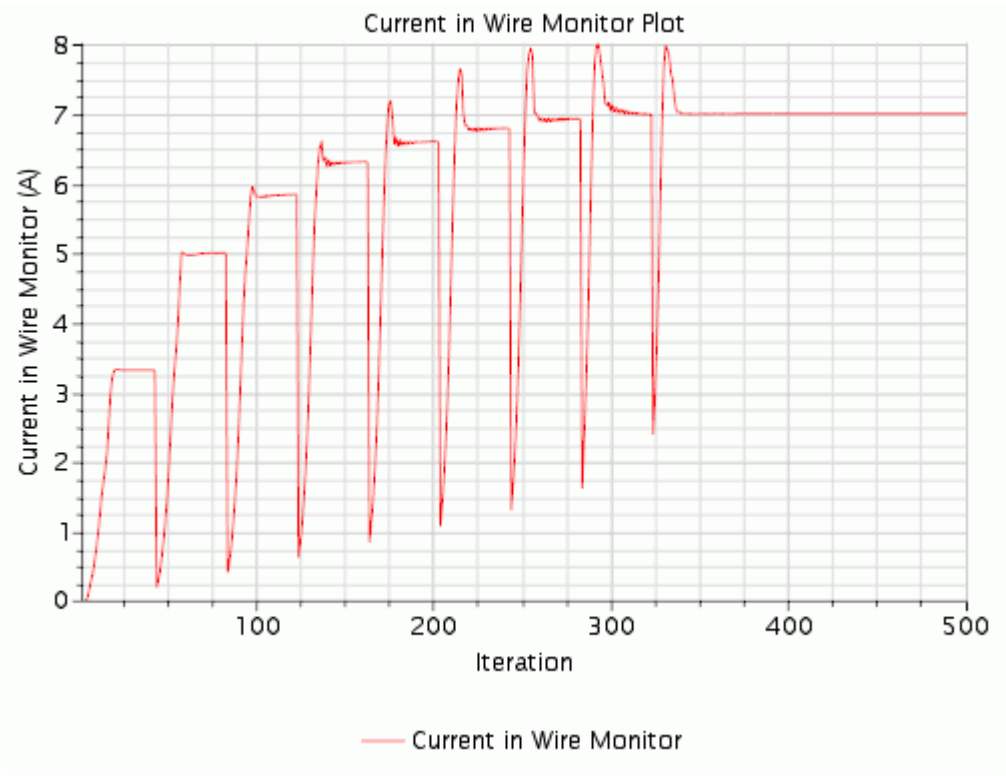


Report Plots

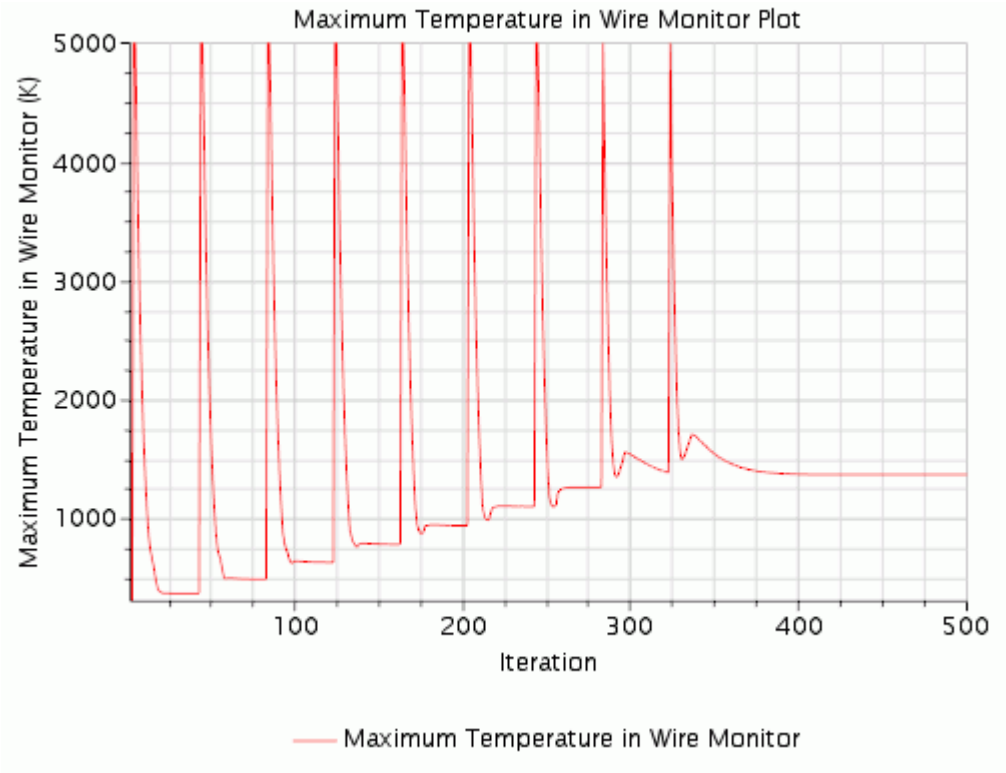
You visualize the current and the maximum temperature in the wire in the form of x-y plots.

To visualize the plots:

1. Click the **Current In Wire Monitor Plot** tab at the top of the **Graphics** window.
   A scene appears as shown



2. Click the **Maximum Temperature Monitor Plot** tab at the top of the **Graphics** window.
   A scene similar to the one shown appears



Summary

This tutorial demonstrated how to set up a simulation involving Ohmic Heating in STAR-CCM+.

This tutorial has introduced the following STAR-CCM+ features:

- Importing a surface mesh, creating a part and naming the simulation.
- Visualizing and examining the imported geometry.
- Generating a volume mesh using a surface remesher and polyhedral mesher.
- Setting the physics models for an Ohmic heating analysis.
- Implementation of user field functions.
- Setting up report, monitors, and plots.
- Post-processing electric potential and temperature.

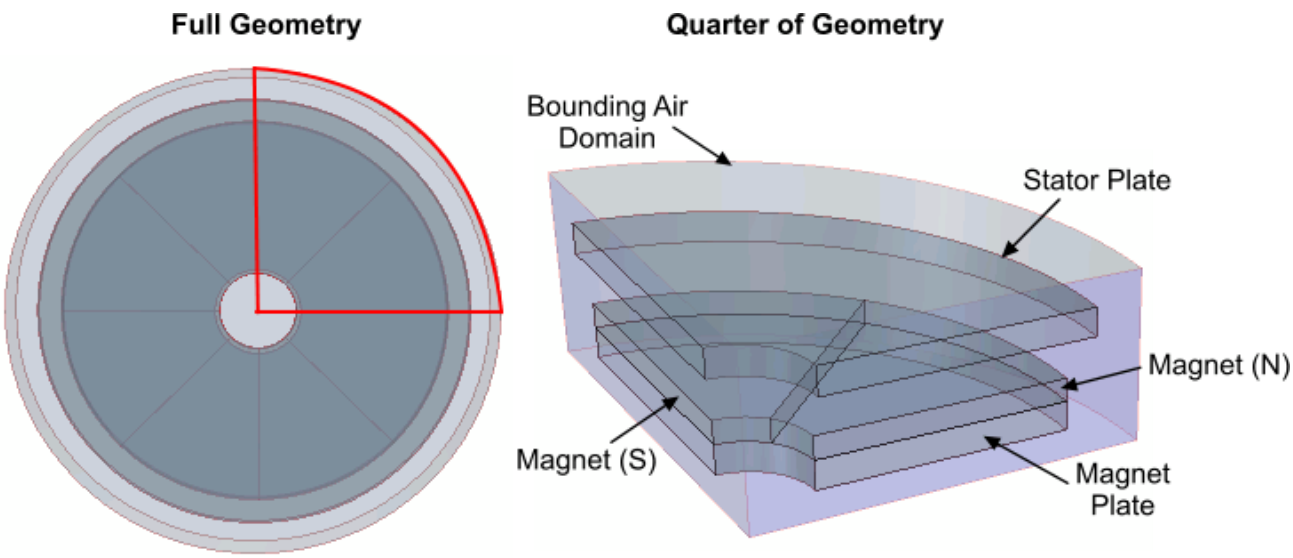
Magnetic Vector Potential: Magnetic Flux

This tutorial demonstrates how to use the Finite Element Magnetic Potential Vector model to solve for the magnetic flux produced by the magnets of an axial-flux motor.

An axial-flux, or “pancake”, motor is a disc-type electric motor where the magnets are arrayed on a rotor around a central shaft. This type of motor is typically used in applications that require a short axial length, for example in elevators where the lifting gear is located inside the elevator shaft.

The main flux in an axial-flux motor passes through an air gap between the rotor and stator in a direction parallel to the axis of rotation. The three-dimensional magnetic flux distribution the motor produces is difficult to calculate using analytical methods.

This tutorial models a quarter of a slotless permanent-magnet brushless motor, surrounded by air:



The windings, which would be located in the gap between the rotor and the stator, are not shown, as they are not used in the simulation.

Analysis Overview

Geometry	1/4 of the full geometry
Mesh	• Linear tetrahedral elements • Conformal interfaces
Materials and Physics	Materials: • Bounding domain: Air • Motor Components: Steel with constant magnetic permeability (linear magnetization assumption) Main Physics Models:

	• **Finite Element Magnetic Potential Vector** to solve for magnetic flux density • **Permanent Magnet** to model the effects of a permanent magnet
Boundary Conditions	Anti-Symmetry Perfect Electric Conductor (PEC) assumption for all boundaries
Interfaces	Boundary-mode interfaces that include: • internal interfaces between the contacting regions • periodic interfaces, to reproduce the behavior of the full geometry model
Discretization and Solution Method	Finite Element (FE) method

Contents:

Prerequisites
Loading the Initial Simulation
Preparing the Regions and Generating the Mesh
Selecting the Physics Models
Defining the Materials and Magnetic Properties
Preparing for Analysis
Running the Simulation
Visualizing the Results
Summary

Prerequisites

To complete this tutorial, you require a double-precision version of STAR-CCM+. Also, the instructions in this tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Techniques	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Loading the Initial Simulation

For this tutorial, you are provided with a starting simulation that includes predefined geometry parts representing the motor components and the air domain.

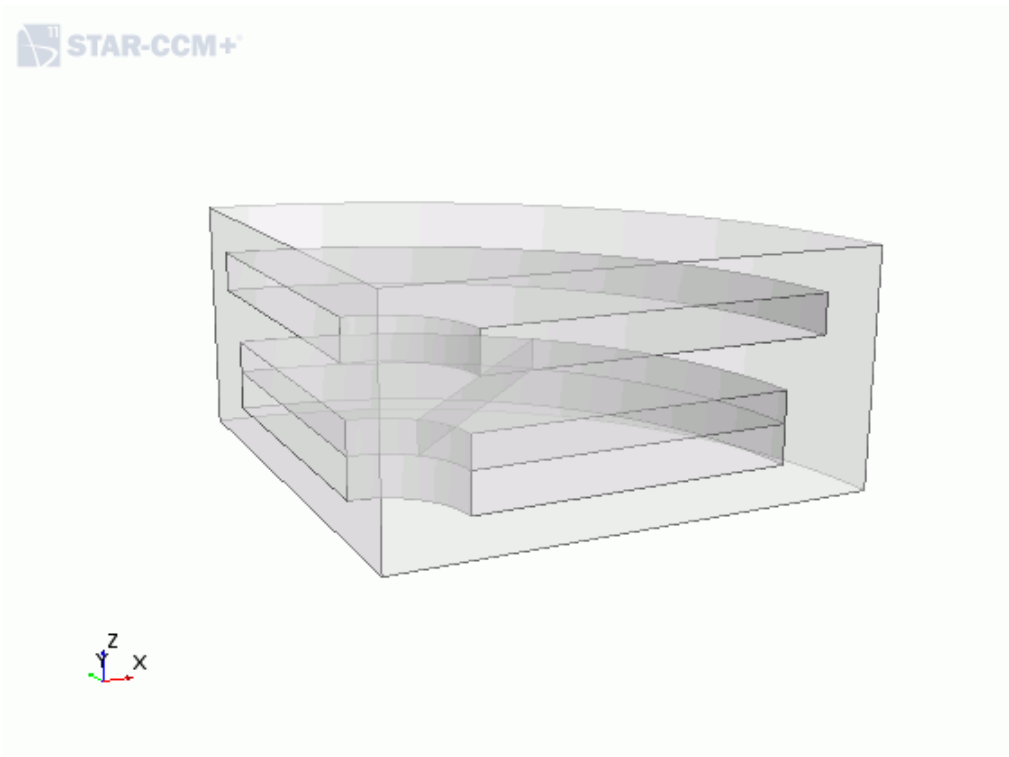
To launch the initial simulation:

1. Launch STAR-CCM+ in double-precision.
2. Select **File** > **Load Simulation...**
3. In the *Load Simulation* dialog, click **Browse...**

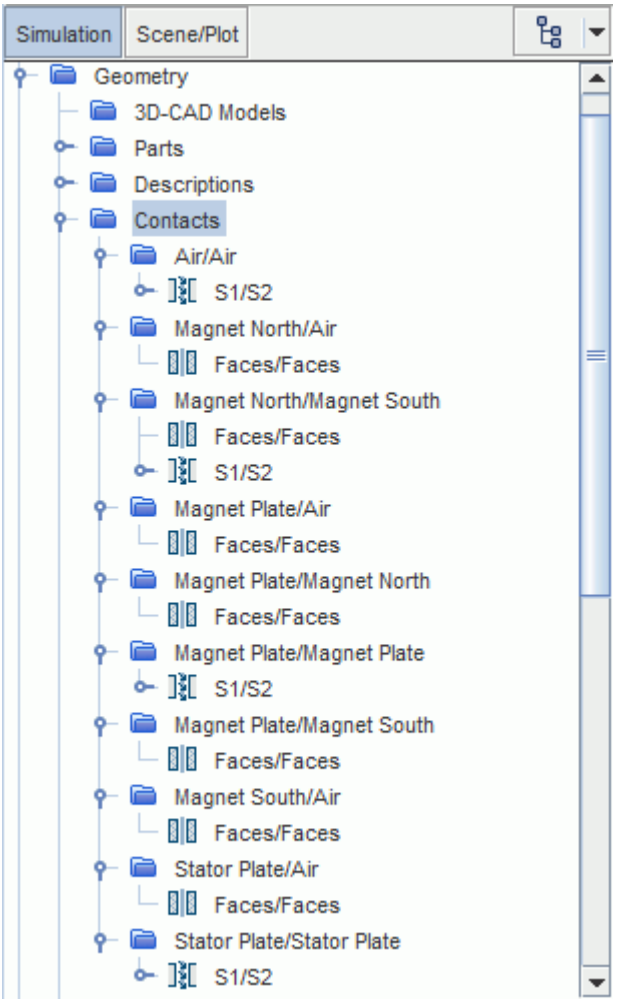
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/electromagnetism/data`.
5. Select `FEmagneticFlux_start.sim` and click **Open**.
6. In the *Load Simulation* dialog, click **OK**.
7. Save the simulation to your working directory as `FEmagneticFlux.sim`.

To visualize the predefined geometry:

8. Expand the **Scenes** node and open **Geometry Scene 1**.
9. Click (**Make Scene Transparent**).



The starting simulation file includes the required contacts between the parts. To reproduce the behavior of the full geometry model, you are provided with periodic contacts between the relevant part surfaces. You can visualize the predefined contacts in the geometry scene by selecting the corresponding nodes.



Preparing the Regions and Generating the Mesh

Assign the predefined geometry parts to regions and generate a mesh of tetrahedral elements.

Before you set up a mesh operation, create a region for each motor component and the air domain, and boundary-mode interfaces between the regions:

1. Expand the **Geometry** > **Parts** node and multi-select all parts.
2. Right-click one of the selected parts and select **Assign Parts to Regions**.
3. In the *Assign Parts to Regions* dialog, select the following options:

   Create a Region for Each Part
   Create a Boundary for Each Part Surface

   The default interface option is appropriate, as it instructs STAR-CCM+ to create boundary-mode interfaces. The **Finite Element Magnetic Vector Potential** model only supports boundary-mode interfaces.
4. Click **Apply** then **Close**.

STAR-CCM+ adds a region for each geometry part and creates internal boundary-mode interfaces between the regions. As the starting simulation file includes predefined periodic contacts between the parts, STAR-CCM+ automatically creates periodic interfaces between the regions.

In three dimensional simulations, the **FE Magnetic Vector Potential** model requires a mesh of tetrahedral, hexahedral, or wedge elements, with conformal interfaces. In this tutorial, you generate a mesh of tetrahedral elements for all parts using a single mesh operation. As the **Per-Part Meshing** option is off by default, STAR-CCM+ automatically produces a conformal mesh at all interfaces.

5. Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
6. In the *Create Automated Mesh Operation* dialog:
   a) Select all parts in the *Parts* group box.
   b) In the *Select Meshers* group box, choose the following meshers:

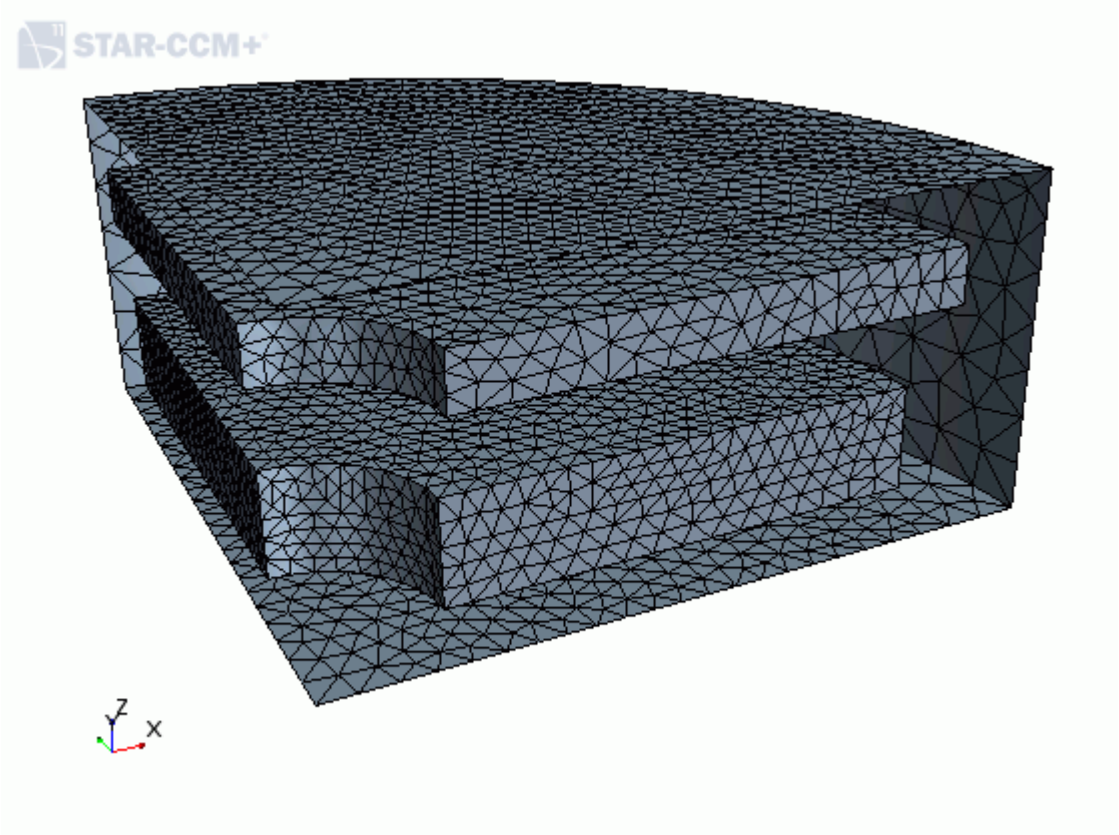
Surface Meshers	**Surface Remesher**
Core Volume Meshers	**Tetrahedral Mesher**

   c) Click **OK**.
7. Rename the **Operations** > **Automated Mesh** node to `Tetrahedral Mesh`.

Specify the size of the tetrahedral elements:

8. Expand the **Operations** > **Tetrahedral Mesh** > **Default Controls** node.
9. Select the **Base Size** node and set *Value* to `0.01 m`.
10. Click (**Generate Volume Mesh**).

    You can visualize the mesh with a method of your choice, such as creating a mesh scene. In the following example, two of the air domain boundaries are hidden, for better visualization of the internal mesh.



11. Save the simulation.

Selecting the Physics Models

In this simulation, you study the steady state distribution of the magnetic flux density throughout the motor components. This simulation requires two physics continua, one for the air region and one for the motor components.

You model the bounding air domain as an ideal gas in a laminar flow regime. To account for the magnetic flux throughout the air region, you activate the **Finite Element Magnetic Vector Potential** model:

1. Rename the **Continua** > **Physics 1** node to `Air`.
2. For the physics continuum, **Air**, select the following models:

Group box	Physics Model
Enabled Models	**Three-Dimensional** (selected automatically)
Time	**Steady**
Material	**Gas**
Flow	**Segregated Flow**
Enabled Models	**Gradients** (selected automatically)
Equation of State	**Ideal Gas**

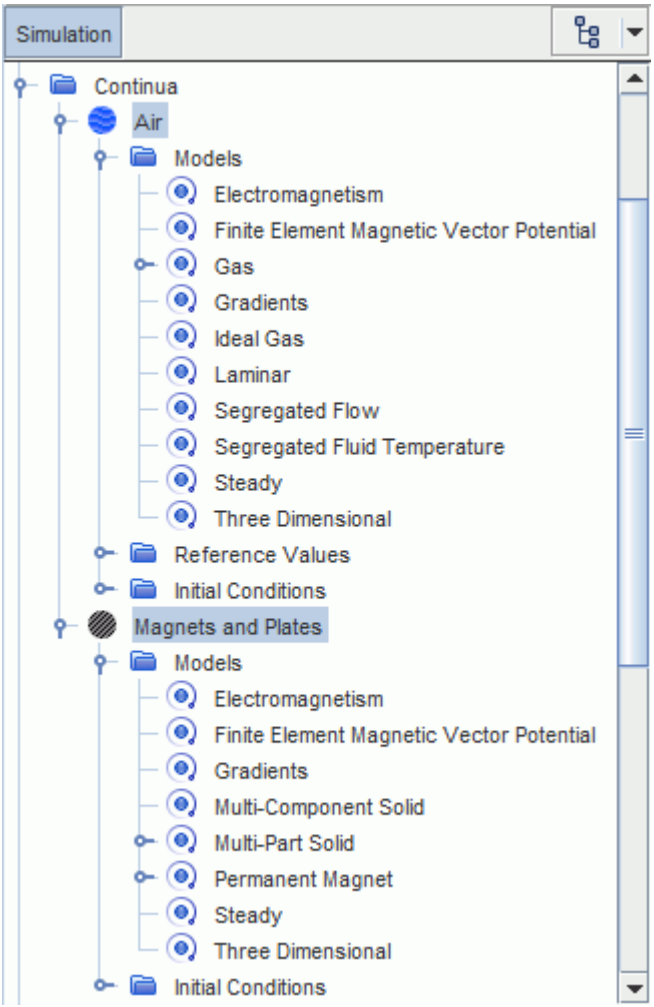
Group box	Physics Model
Energy	**Segregated Fluid Temperature**
Viscous Regime	**Laminar**
Optional Models	**Electromagnetism**
Electromagnetism	**Finite Element Magnetic Vector Potential**

You model the magnets, the magnet plate, and the stator plate using the same physics continuum. The **Multi-Part Solid** model allows you to specify different material properties for each solid region. The **Finite Element Magnetic Vector Potential** model accounts for the magnetic flux density in the motor components, whereas the **Permanent Magnet** model allows you to specify remanence magnetic flux density for the magnets:

3. Create a physics continuum named `Magnets and Plates` and select the following models:

Group box	Physics Model
Space	**Three-Dimensional**
Time	**Steady**
Material	**Multi-Component Solid**
Multi-Component Solid	**Multi-Part Solid**
Optional Models	**Electromagnetism**
Enabled Models	**Gradients** (selected automatically)
Electromagnetism	**Finite Element Magnetic Vector Potential**
Optional Models	**Permanent Magnet**

The physics continua appear as shown below:



Assign each region to the relevant physics continuum:

4. Expand the **Regions** node.
5. Multi-select the **Magnet North**, **Magnet Plate**, **Magnet South**, and **Stator Plate** region nodes and set *Physics Continuum* to **Magnets and Plates**.

   The **Air** region is already assigned to the **Air** physics continuum.
6. Save the simulation.

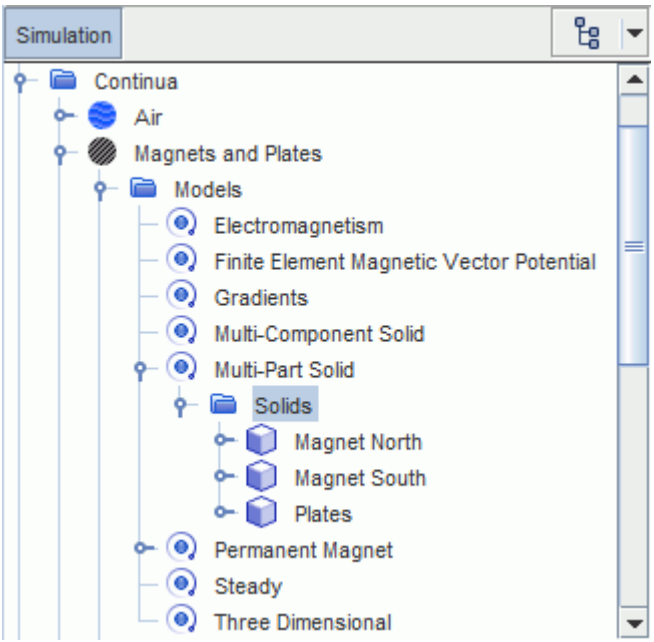
Defining the Materials and Magnetic Properties

In this simulation, you define the magnetic behavior of the motor by specifying the magnetic permeability of its components, and a residual magnetic flux density for the permanent magnets.

Although all motor components are made of steel, you require multiple material definitions to set the material properties of each component independently. As the magnets and plates have a different magnetic permeability, and magnet north and south have a different polarity, you require three material definitions, one for the plates, and one for each magnet.

1. Expand the **Continua** > **Magnets and Plates** > **Models** > **Multi-Part Solid** node.

2. Right-click the **Solids** node and select **Select Mixture Components...**
3. In the *Select Mixture Components...* dialog:
   a) Select **UNSS30200 (Stainless Steel)** from the standard material database.
   b) To add three material definitions, click **Apply** three times, then **Close**.
4. Rename the **Multi-Part Solid** > **Solids** > **UNSS30200** node to `Magnet North`, **UNSS30200 2** to `Magnet South`, and **UNSS30200 3** to `Plates`.



Specify the magnetic permeability of each material, and a remanence magnetic flux density for the permanent magnets. The remanence magnetic flux density is the magnetic field remaining in a permanent magnet after any external magnetic field is removed. For simplicity, this simulation models the plates and magnets using the same physics continuum. To deactivate the **Permanent Magnet** model for the plates, simply set the plates remanence magnetic flux density to zero.

5. Expand the **Magnet North**, **Magnet South**, and **Plates** nodes.
6. For each material, edit the **Material Properties** node and specify the following values:

Material	**Magnetic Permeability > Constant**	**Remanence Magnetic Flux Density > Constant**
Magnet North	`1.319493E-6 H/m`	`1.2 T`
Magnet South	`1.319493E-6 H/m`	`-1.2 T`
Plates	`6.2833E-4 H/m`	`0 T`

To specify the direction of the magnets remanence magnetic flux density:

7. Select the **Magnets and Plates** > **Models** > **Permanent Magnet** > **Magnetization Direction** node and set *Value* to `[0.0,0.0,1.0]`.

Associate each solid region to the relevant material definition:

8. Expand the **Regions** node.
9. Select the **Magnet North** > **Physics Values** > **Material Part Groups** > **Magnet North** node and set *Parts* to **Magnet North**.
10. Using the same procedure, assign **Magnet South** to the **Magnet South** material group.
11. Select the **Magnet Plate** > **Physics Values** > **Material Part Groups** > **Plates** node and set *Parts* to **Magnet Plate**.
12. Using the same procedure, assign **Stator Plate** to the **Plates** material group.
13. Save the simulation.

Preparing for Analysis

Create a scalar scene to display the Z-component of the magnetic flux density on the magnets. Also, create a plot to display the flux density along a line contour directed along the radius of the magnet plate.

1. Create a scalar scene and rename it to `Magnetic Flux Density [k]`.
2. Edit the **Magnetic Flux Density [k]** > **Displayers** node and set the following properties:

Node	Property	Setting
Outline 1	*Outline*	**Deactivated**
Scalar 1	*Display Mesh*	**Activated**
	Contour Style	**Smooth Filled**
└ **Parts**	*Parts*	**Regions** > **Magnet North**
		Regions > **Magnet South**
└ **Scalar Field**	*Function*	**Magnetic Flux Density** > **Laboratory** > **k**

3. Click (**Save-Restore-Select Views**) and select **Restore View** > **View 1**, then adjust the scene display as appropriate.

To plot the flux-density along a line contour, create a line probe that is directed along the radius of the magnet plate. Position the line origin, that is, the start point, in the center of the air gap. Since the center of the air gap is located 6 mm from the magnet surface, and the magnet length is 8 mm in the z-direction, position the line at z = 14 mm.

To create the line probe, it is convenient to use a cylindrical coordinate system:

4. Right-click the **Tools** > **Coordinate Systems** > **Laboratory** > **Local Coordinate Systems** node and select **New** > **Cylindrical**.
5. In the *Create Cylindrical Coordinate System* in-place dialog, click **Create** then **Close**.
6. Right-click the **Derived Parts** node and select **New Part** > **Probe** > **Line...**.
7. In the *Create Line Probe* in-place dialog:
   a) Click **Select** in the *Input Parts* group box. In the *Select Objects* dialog, click the **Regions** node to select all regions and boundaries, then click **OK**.
   b) Set the remaining properties as follows:

Property	Setting
Coordinate System	**Laboratory** > **Cylindrical 1**
Point 1	*r* = `0 m`
	theta = `0 rad`
	z = `0.014 m`
Point 2	*r* = `0.12 m`
	theta = `22.5 deg`
	z = `0.014 m`
Display	**Existing Displayer** > **Scalar 1**

c) Click **Create** then **Close**.

8. Right-click the **Plots** node and select **New Plot** > **XY Plot**.
9. Rename the **XY Plot 1** node to `Radial Magnetic Flux Density`.
10. Edit the **Radial Magnetic Flux Density** node and set the following properties:

Node	Property	Setting
Radial Magnetic Flux Density	*Parts*	**Derived Parts** > **Line Probe**
└ **Y Types** > **Y Type 1**	*Smooth Values*	**Activated**
└ **Scalar Function**	*Field Function*	**Magnetic Flux Density** > **Magnitude**
└ **Line Probe** > **Line Style**	*Style*	**Solid**

11. Save the simulation.

Running the Simulation

Freeze the flow solver and run the simulation to solve for the magnetic flux.

As this is a linear problem, the FE **Sparse Direct Solver** computes the solution in one iteration. For linear problems, it is good practice to make sure that the residuals drop at least 6 orders of magnitude within the second iteration.

1. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to `2`.

In this simulation, the only variable of interest is the magnetic flux. Therefore, you can freeze the flow solver to speed up the computation:

2. Select the **Solvers** > **Segregated Flow** node and activate **Freeze Flow**.

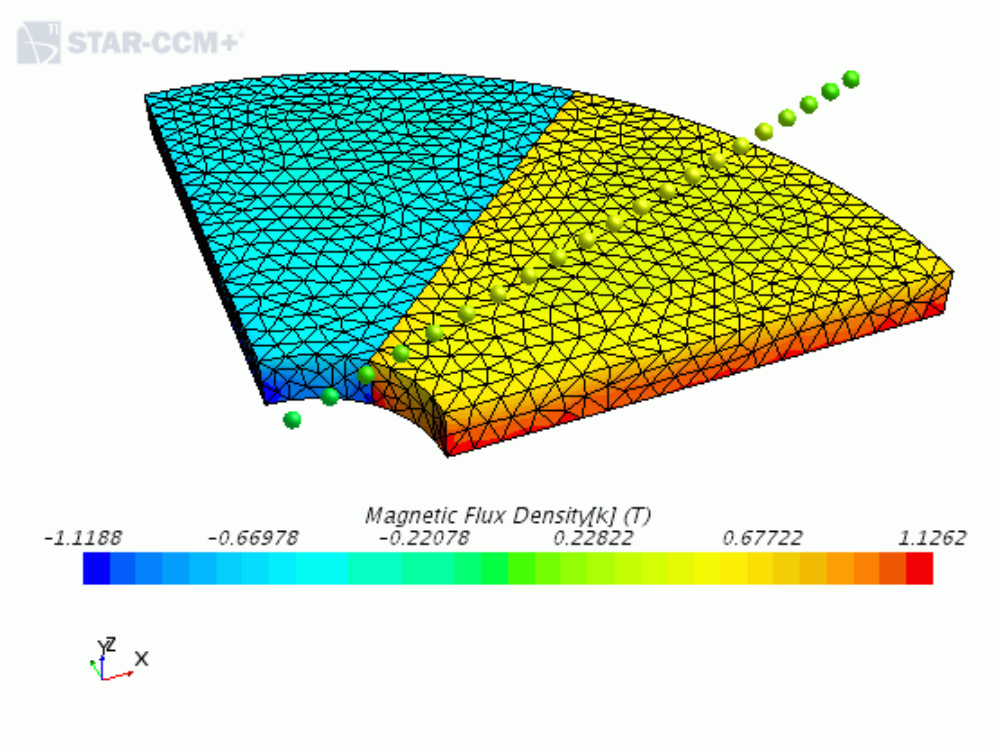
You are now ready to run the analysis:

3. Click (**Run**).
4. When the computation is complete, save the simulation.

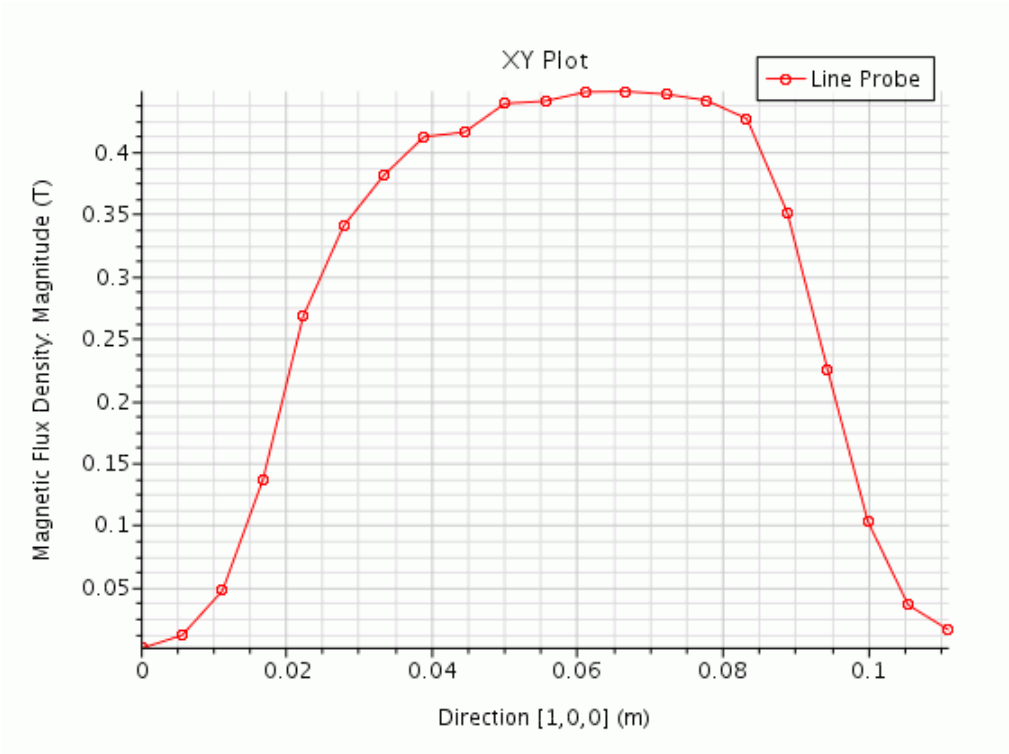
Visualizing the Results

You can visualize the magnetic flux density available for torque production in the scalar scene and XY plot that you created in the previous sections.

The **Magnetic Flux Density [k]** scene shows the distribution of the z-component of the magnetic flux density:



The **Radial Magnetic Flux Density** plot shows the distribution of the magnetic flux density along a line probe, that is directed along the radius of the magnet plate. The graph shows a profile that is typical for axial-flux motors. Linear motor topologies are characterized by a different distribution.



Summary

This tutorial demonstrated how to calculate a 3D magnetic flux field from magnets in a slotless permanent-magnet brushless motor.

Specifically, this tutorial has presented the following procedures:

- Using the **Finite Element Magnetic Vector Potential** and the **Permanent Magnet** models
- Defining materials and specifying the magnetic permeability
- Specifying a remanent magnetic flux to model permanent magnets
- Visualizing the computed magnetic flux in a scalar scene
- Visualizing the radial distribution of the magnetic flux using a line probe

Electrochemistry

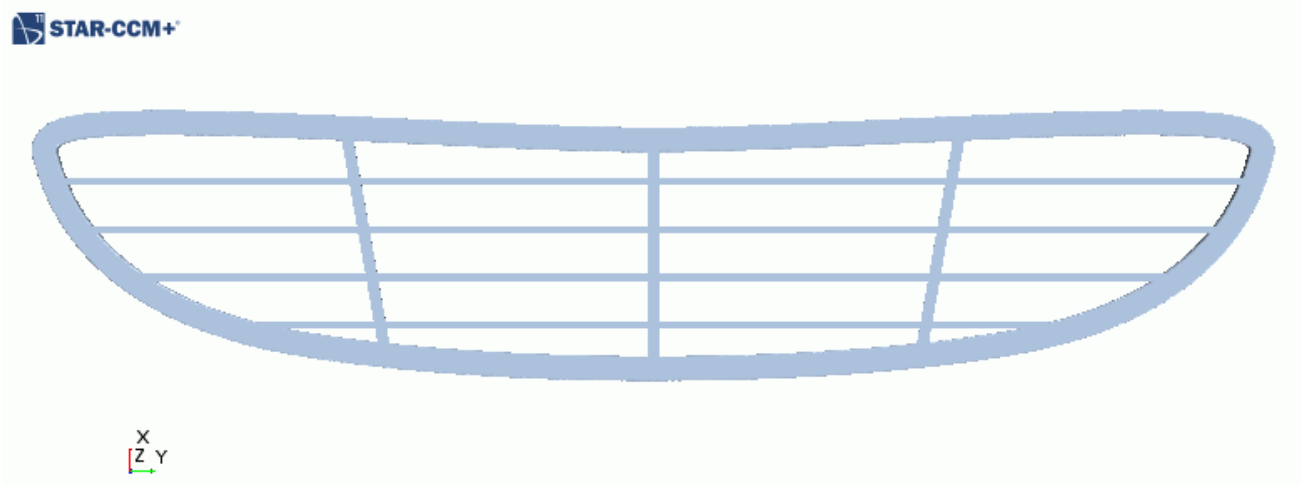
The following tutorial demonstrates chemical reactions induced by an electrical current.

Contents:

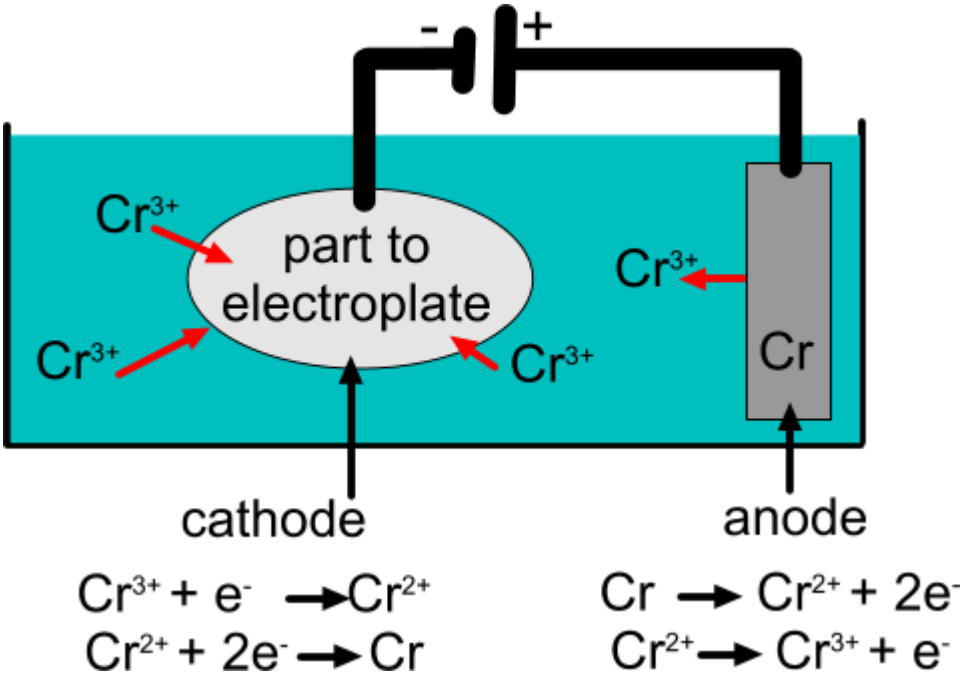
Electrochemistry: Electroplating
Electrochemistry: Solid Oxide Fuel Cell

Electrochemistry: Electroplating

Electroplating is a technique that is commonly used in industry to deposit a metal coating on parts. The part that requires coating sits within an electrolytic solution that contains metal ions. A current is then induced through the electrolyte, between the part acting as the cathode, and an anode that is made of the same metal as the metal ions. In this tutorial, you simulate electroplating of a plastic grille part, which is pre-coated with copper and nickel, using chromium ions.



In this tutorial, a voltage difference is applied between a chromium anode and a cathode that is formed from a plastic grille that already has a coating of copper and nickel. The voltage difference drives positively charged chromium ions (Cr3+) through the electrolyte from the anode to the cathode. When the Cr3+ ions reach the surface of the cathode, they gain electrons from the cathode and form chromium atoms. Since the metal chromium atoms are insoluble, they remain on the surface of the cathode to form a thin plate layer. As electrons move through the circuit from the anode to the cathode, metal chromium atoms at the anode lose electrons and become chromium ions in solution.



In this tutorial, the plastic grille part is held in the electrolyte solution by a metal rack which conducts electrons from the circuit to the grille part. The plastic grille is already coated with thin layers of copper and three types of nickel. The metal rack is coated in paint to prevent it becoming plated, although it is exposed near each point of contact with the grille. The conductivity of the metal rack is much greater than the conductivity of the electrolyte or the previously plated material. Therefore, ohmic losses are negligible and the electric potential in the rack is considered to be constant. A constant electric potential boundary condition is defined at the contact points between the rack and the grille.

Since the electrolyte solution is water-based, and hydrogen ions are reduced more easily than chromium ions, passing an electric current through the solution also creates hydrogen gas at the cathode surface.

$$2\mathrm{H}^{+} + 2\mathrm{e}^{-} \rightarrow \mathrm{H}_2 \tag{4173}$$

This unwanted side reaction consumes part of the electric current. In this tutorial, the detailed reaction mechanism is not modeled. Instead, the aggregate polarization curve is imported from a table [811], and a Faraday efficiency factor accounts for the fraction of current that is associated with the metal deposition reaction.

STAR-CCM+ ensures that current is conserved in the bulk electrolyte and metal shell using the following current conservation equations:

$$\vec{\nabla} \cdot \vec{J} = 0 \tag{4174}$$

$$\vec{J} = -\sigma \vec{\nabla} \phi \tag{4175}$$

where:

- $\vec{J} \equiv$ current density vector
- $\sigma \equiv$ electrical conductivity
- $\phi \equiv$ electric potential

At the interface between fluid and solid domains, where electrochemical reactions occur, STAR-CCM+ incorporates a direct linearization of the following term:

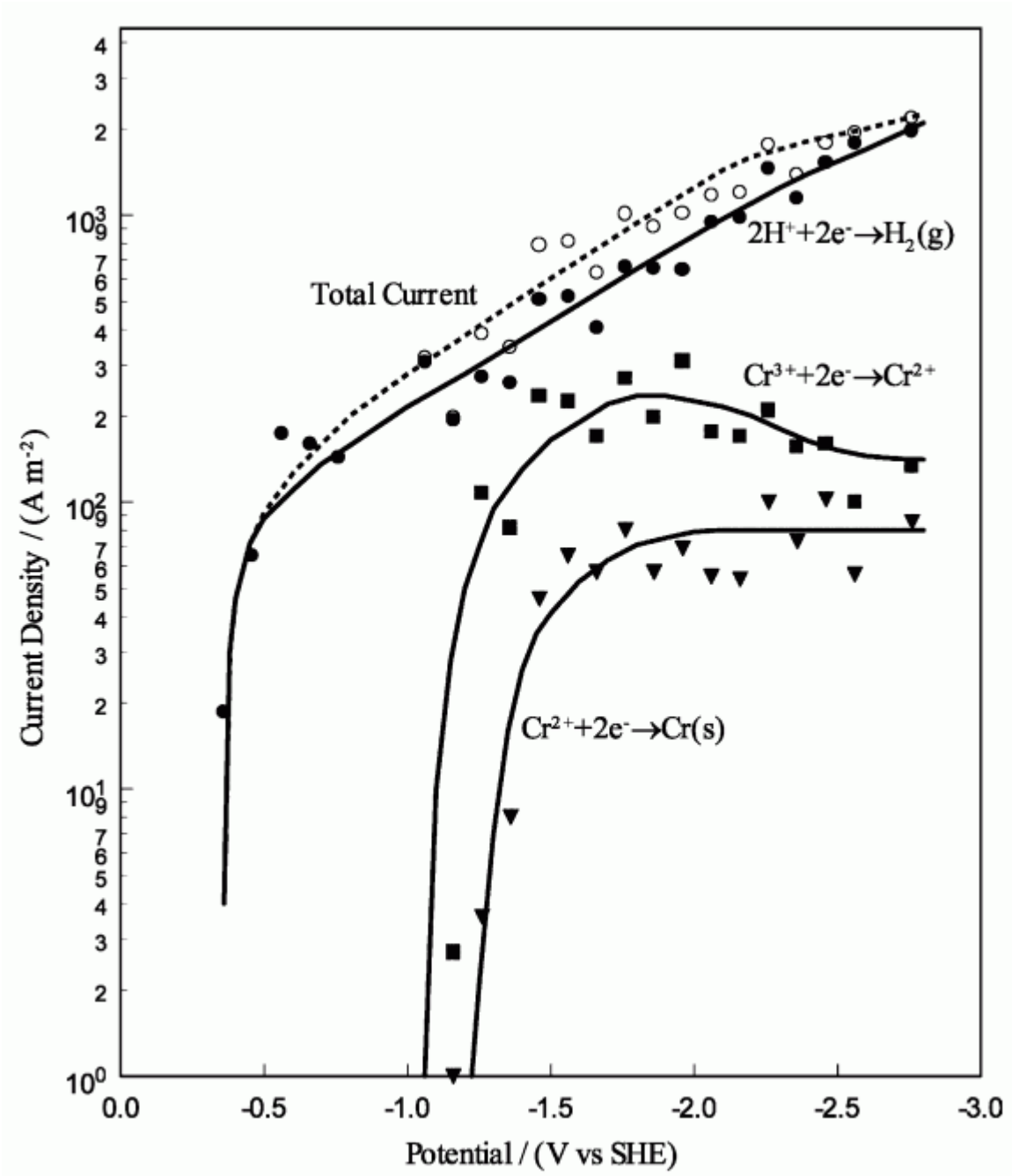
$$\vec{J} \cdot \vec{n} = j\left(\eta\right) \tag{4176}$$

$$\eta = \phi_{solid} - \phi_{fluid} - U_{eq} \tag{4177}$$

$U_{eq} \equiv$ equilibrium potential at zero current

$j(\eta)$ is a non-linear function which the Butler-Volmer, Tafel, or Tabular Polarization Curve describe.

The current density at the interface represents the total current for all electrochemical reactions at the surface. Side reactions, such as the evolution of hydrogen gas, result in a decrease in efficiency. Therefore, the efficiency of the surface mechanism that is used represents the amount of current which goes towards the plating reaction. This mechanism is based on the current densities that are calculated at the cathode, according to Song and Chin [811], at electric potentials that are relative to the standard hydrogen electrode (SHE).



Depending on the reason for electroplating parts, the thickness of the metal coating can become significant. If the coating is intended to protect the part from corrosion, then a thicker coating is required than if the coating is for decorative purposes. In this tutorial, the thickness of the pre-coated layer is approximated as the constant thickness of the shell region that represents the coating. To model a more accurate representation, you can run a transient simulation which accounts for the changing thickness at each time-step.

The thickness of the chromium plating is calculated using the following equation:

$$d = \theta \frac{M}{zF\rho} j \Delta t \tag{4178}$$

where:

- d is the plating thickness [m]
- M is the molecular mass of the metal plating, which in this case is 51.99 for chromium [kg/kmol]
- z is the charge of metal ion, which in this case is 3.
- F is the Faraday constant, which is 96,485,336.5 [s A/kmol].
- ρ is the plated metal mass density, which in this case is 7.19×10^3 [kg/m³]
- Δt is the plating time, which in this case is 300 [s]
- j is the current density (Boundary Specific Electric Current) which is returned from the simulation by a field function [A / m²]
- θ is the plating efficiency. The `Song_Cr3+Polarization` table contains lists of plating efficiencies at various current densities. This data is interpolated to return plating efficiencies at the local values of current density (Boundary Specific Electric Current).

When running the tutorial, the thickness of the chromium coating and the electric potential of the cathode (grille) are monitored using scalar scenes. If the current at the cathode drops, the rate of electrodeposition of chromium also drops. Therefore, you also plot the electric current at the anode and the cathode.

Contents:

Preparing the Simulation Objects and Mesh
Setting up the Physics
Preparing Post Processing
Running the Simulation and Reviewing the Results
Bibliography

Preparing the Simulation Objects and Mesh

The geometry is already imported and the mesh operation is ready to execute.

Make sure that you have two processes available on the local host machine that you are using.

1. Launch a double precision version of STAR-CCM+ and click (**Load Simulation**).
2. In the *Load Simulation* dialog:
   a) In the *Process Options* group box, select **Parallel on Local Host** and set *Compute Processes* to `2`.
   b) Next to *File*, click **Browse** and navigate to `[INSTALL_DIR]/doc/startutorialsdata/electrochemistry/data`
   c) Select `ElectroplatingTutorial_StartFile.sim` then click **Open**.
   d) Click **OK**.
3. Save the simulation file as `GrilleElectroplating.sim`.

The pre-defined simulation file already contains the following items:

- an automated mesh operation with the relevant settings
- the fluid region with parts pre-assigned, and whose boundaries are set to the correct types

A tabular polarization curve is required for the reaction mechanism. To import the table:

4. Right-click the **Tools** > **Tables** node and select **New Table** > **File Table**.
5. In the **Open** dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/electrochemistry/data`.
6. Select `Song_Cr3+Polarization.csv` and click **Open**.

   The table is imported and appears under the **Tools** > **Tables** node.

To create the volume mesh:

7. Right-click the **Geometry** > **Operations** > **Automated Mesh** node and select **Execute**.

   STAR-CCM+ generates a volume mesh containing around 680,000 cells.
8. Right-click the **Scenes** > **Mesh Scene 1** node and choose **Open**.

   The grille is displayed showing the supporting rack.

Setting up the Physics

Set up physics continua for both the fluid electrolyte and the solid shell plating on the grille.

To define the physics continua:

1. For the physics continuum, **Continua** > **Electrolyte**, select the following models in order:

Group Box	Model
Enabled Models	**Three Dimensional** (selected automatically)
Time	**Steady**
Material	**Liquid**
Flow	**Segregated Flow**
Enabled Models	**Gradients** (selected automatically)
Equation of State	**Constant Density**
Viscous Regime	**Laminar**
Optional Models	**Electromagnetism**
Electromagnetism	**Electrodynamic Potential**
Optional Models	**Electrochemistry**
Electrochemistry	**Electrochemical Reactions**

2. Click **Close**.

Set up the reaction mechanism at the surface:

3. Expand the **Continua** > **Electrolyte** > **Models** node.
4. Right-click the **Electrochemical Reactions** > **Surface Mechanism Manager** node and select **New Surface Mechanism**.
5. Rename the **Surface Mechanism Manager** > **Surface Mechanism** node to `Chrome`.
6. Right-click the **Surface Mechanism Manager** > **Chrome** > **Models** > **Reacting Surface** > **Reactions** node and select **New Reaction**.

7. Rename the **Reactions** > **Reaction 1** node to `Cr = Cr3+ + 3e-`
8. Select the **Reactions** > **Cr = Cr3+ + 3e-** > **Properties** > **Reaction Formulation** node and set *Select Method* to **Tabular**.
9. Select the **Reaction Formulation** > **Tabular Polarization Curve** node and set the following properties:

Property	Setting
Table	`Song_Cr3+Polarization`
Electric Potential	**potential**
Specific Electric Current	**current density**

Since the electrolyte is an aqueous solution, most of the material properties for water are suitable for the simulation. However, to account for the conducting ions in solution, the Electrical Conductivity must be adjusted.

10. Within **Electrolyte** > **Models**, select the **Liquid** > **H2O** > **Material Properties** > **Electrical Conductivity** > **Constant** node and set *Value* to `34.1 S/m`.
11. Select the **Continua** > **Electrolyte** > **Initial Conditions** > **Electric Potential** node and set *Value* to **1.0 V**.

Create a physics continuum for the pre-coated layers of copper and nickel onto which the chromium is electroplated. In this tutorial, default initial conditions are maintained for this physics continuum.

12. Right-click the **Continua** node and select **New** > **Physics Continuum**. Rename the resulting **Physics 1** node to **Plating**.
13. For the physics continuum, **Continua** > **Plating**, select the following models in order:

Group Box	Model
Space	**Shell Three Dimensional**
Time	**Steady**
Optional Models	**Solid**
	Electromagnetism
Enabled Models	**Gradients** (selected automatically)
	Electrodynamic Potential (selected automatically)
Optional Models	**Electrochemistry**
Enabled Models	**Electrochemical Reactions** (selected automatically)

14. Click **Close**.

The electrochemical reactions occur in the electrolyte at the boundaries to the anode and cathode. Therefore, conditions and values are specified at the anode and cathode boundaries. The thin layer of pre-plated copper and nickel is defined as a separate shell region which occurs at the boundary between the electrolyte and the grille (cathode) part. The chrome ions from the electrolyte are then deposited as chrome atoms on top of the shell region.

15. Expand the **Regions** > **Electrolyte** > **Boundaries** > **Anode** node and set the following Properties:

Node	Property	Setting
Physics Conditions		
⌊ **Electric Potential Specification**	*Method*	**Specific Electric Current**
Physics Values		
⌊ **Specific Electric Current**	*Value*	`165.25 A/m^2`

16. Right-click the **Regions** > **Electrolyte** > **Boundaries** > **Grille** node and select **Create Shell Region**.
17. Rename the **Regions** > **Grille shell** node to `Grille` and set its *Continuum* property to **Plating**.

    The *Type* is set to **Solid Region** automatically since a solid continuum is specified.
18. Select the **Regions** > **Grille** > **Physics Values** > **Shell Thickness** node and set *Value* to `1.0E-6 m`:

    This shell thickness represents the thickness of the copper and nickel layer on the plastic part.
19. Expand the **Grille** > **Boundaries** > **Rack edges** node and set the following properties:

Node	Property	Setting
Physics Conditions		
⌊ **Electric Potential Specification**	*Method*	**Electric Potential**
Physics Values		
⌊ **Electric Potential**	*Value*	`0.0 V`

20. Select the **Interfaces** > **In-place 1** > **Physics Conditions** > **Surface Mechanism Option** node and set *Mechanism* to **Chrome**.
21. Click **(Save)**.

Preparing Post Processing

To monitor and visualize the results from the simulation, set up reports and scenes using appropriate parts and specify the relevant physics entities to monitor.

This section describes how to set up the following scenes and reports:

- Reports that monitor the current at the anode and the cathode
- A scalar scene that displays electric potential on the surface of the grille
- A scalar scene that shows thickness of the Cr3+ coating that is covering the grille

1. Right-click the **Reports** node and select **New Report** > **Surface Integral**.
2. Rename the **Reports** > **Surface Integral 1** node to `Anode Current`.
3. Select the **Reports** > **Anode Current** node and set the following Properties:

Property	Setting
Field Function	**Boundary Specific Electric Current**
Parts	**Regions** > **Electrolyte** > **Boundaries** > **Anode**

4. Right-click the **Reports** > **Anode Current** node and select **Copy**, then right-click the **Reports** node and select **Paste**.
5. Rename the **Copy of Anode Current** node to `Cathode Current` then select the **Cathode Current** node and set the following properties:

Property	Setting
Field Function	**Electrochemical Reaction Rate** > **Electrochemical Reaction Rate of Cr = Cr3+ + 3e- of Chrome of Electrolyte**
Parts	**Regions** > **Electrolyte** > **Grille [In-place 1]**

6. Multi-select the **Reports** > **Anode Current** node and the **Reports** > **Cathode Current** node, then right-click one of the selected nodes and select **Create Monitor and Plot From Report**.
7. In the *Create Plot From Reports* dialog, click **Single Plot**.

   The **Reports Plot** appears in the *Graphics* window.
8. Select the **Plots** > **Reports Plot** > **Axes** > **Left Axis** node and set *Minimum* to `-200`.

Create a scalar scene that displays the electric potential on the surface of the grille.

9. Create a Scalar scene and rename it `Grille Electric Potential`.
10. To hide the outline in the displayer, double-click the **Grille Electric Potential** > **Displayers** > **Outline 1** node.
11. Expand the **Grille Electric Potential** > **Displayers** > **Scalar 1** node and set the following properties:

Node	Property	Setting
Parts	*Parts*	**Regions** > **Grille**
Scalar Field	*Function*	**Electric Potential**

Before setting up the Cr3+ Plating Thickness scalar scene, create the field function that is used to calculate the thickness of the plating. This field function comes from Eqn. (4178).

12. Right-click the **Tools** > **Field Function** node and select **New** > **Scalar**.
13. Rename the **Field Functions** > **User Field Function 1** node to `Cr3+ Plating Thickness` and set the following properties:

Note: When copying and pasting the definition for this field function, make sure to maintain the space between the words in the term `"current density"`.

Property	Setting
Function Name	`Cr3+ Plating Thickness`
Dimensions	**Length**

Property	Setting
Definition	`51.99*($BoundarySpecificElectricCurrent)*300* interpolateTable(@Table("Song_Cr3+Polarization"), "current density",LINEAR,"efficiency", -$BoundarySpecificElectricCurrent)/ (3*96485336.5*7.19e3)`

14. Create a Scalar scene and rename it `Cr3+ Plating Thickness`.
15. Hide the outline in the displayer.
16. Expand the **Cr3+ Plating Thickness** > **Displayers** > **Scalar 1** node and set the following properties:

Node	Property	Setting
Parts	*Parts*	**Regions** > **Electrolyte** > **Boundaries** > **Grille [In-place 1]**
Scalar Field	*Function*	**Cr+3 Plating Thickness**
	Units	**um**

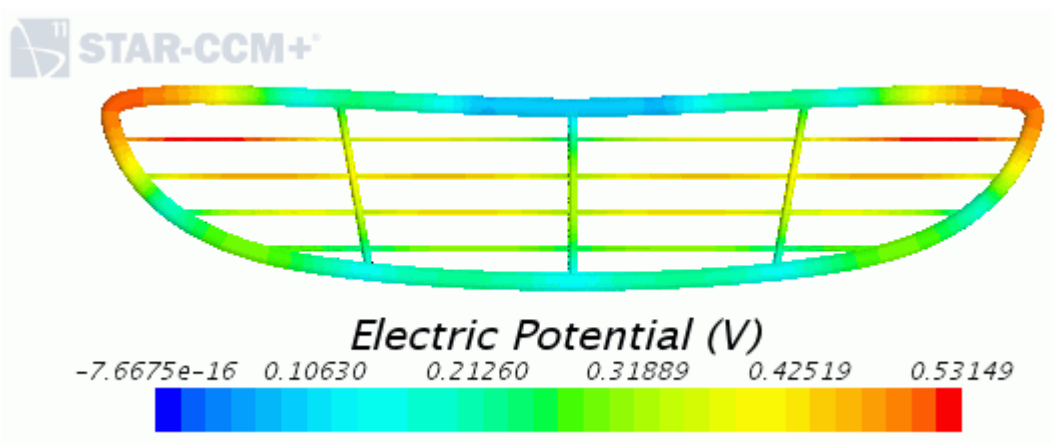
Running the Simulation and Reviewing the Results

After running the simulation, you can view the results using the scalar scenes and plots that are set up.

The default Solver settings are suitable for this simulation.

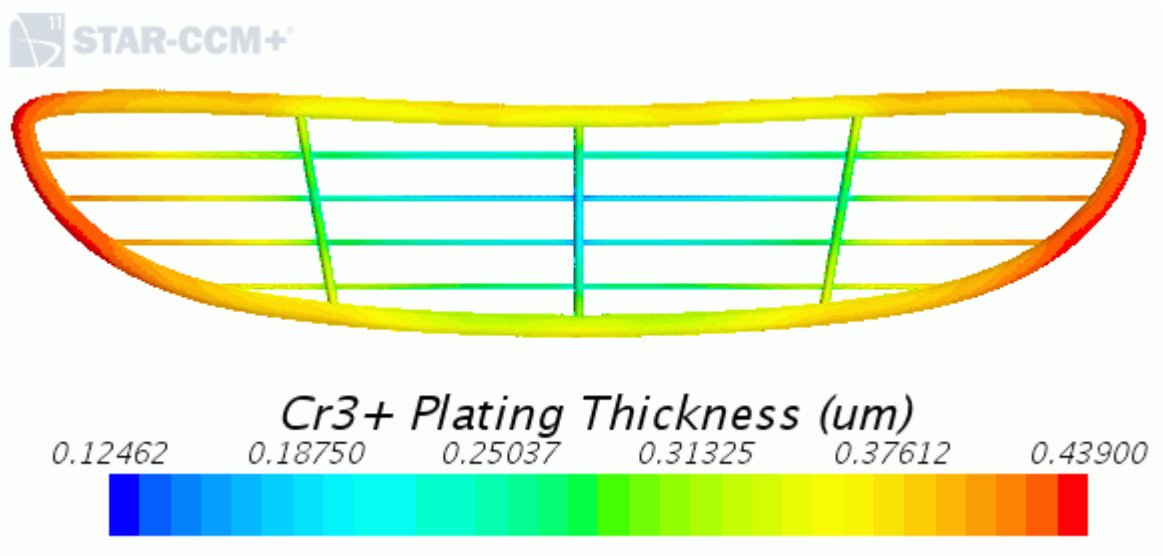
1. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to `50`.
2. Click (**Run**).
3. When the simulation is complete, open the **Grille Electric Potential** scene.

   The *Graphics* window displays the electric potential values on the grille part.



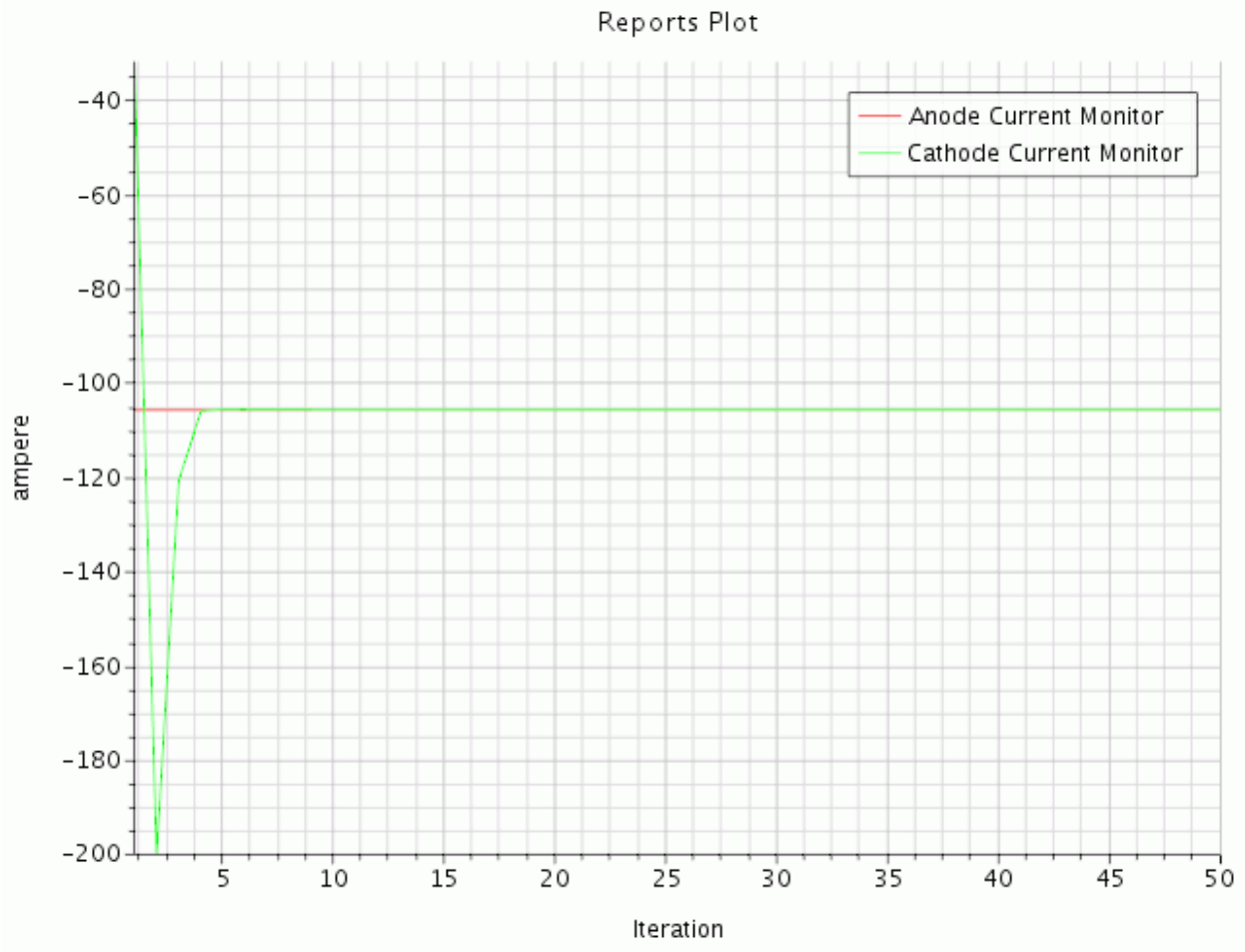
4. Open the Cr3+ Plating Thickness scene.

The *Graphics* window displays the thickness of the chromium plating on the grille part.



5. To check that the current is conserved between the anode and the cathode, open the **Plots** > **Reports Plot** node.

   The *Graphics* window displays the current at the anode and the cathode.



Ohmic losses and variations in electric potential within the pre-plated shell can cause the plating of the chromium to become more localized near the points of contact between the rack and grille. Further studies with this simulation could include observations of how the thickness of the copper and nickel shell affects the plating process. A thinner shell results in wider variations in electric potential values throughout the shell, which affects the amount of chromium that forms on the surface of the shell. For instability troubleshooting, modify the electric potential initial conditions and/or decrease the electric potential under-relaxation factor in the solver settings.

Bibliography

[811] *Song, Y. B., and D-T. Chin. "Current efficiency and polarization behavior of trivalent chromium electrodeposition process." Electrochimica Acta 48.4 (2002): 349-356.*

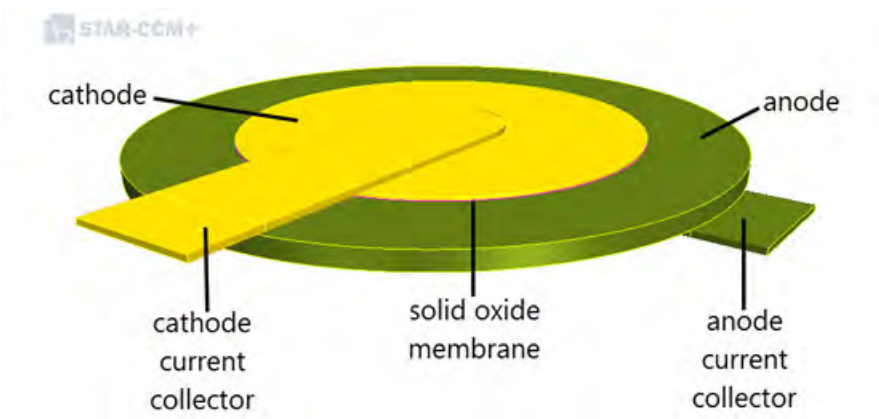
Electrochemistry: Solid Oxide Fuel Cell

Solid Oxide Fuel Cells (SOFCs) harness the chemical energy that is released by electrochemical reactions and convert it to electrical energy. SOFCs typically consume hydrogen and oxygen and only produce water as a waste product.

SOFCs are electrochemical cells (or stacks of cells) which are composed of electrodes (an anode and a cathode), a solid oxide electrolyte, and current collectors.

This tutorial demonstrates the workflow for setting up and simulating a single planar button solid oxide fuel cell. The mole fraction of oxygen is visualized in relation to the reaction flux. It is important to check that the transport of oxygen through the cathode is sufficient enough to avoid oxygen starvation—which would limit the reaction rate and damage the fuel cell components. Setting up a case in this way with a single cell is useful to first determine the optimal set-up, transport mechanism, and reaction mechanism before scaling-up to simulate multiple cells in a stack.

The following diagram shows the different components of the planar button SOFC.

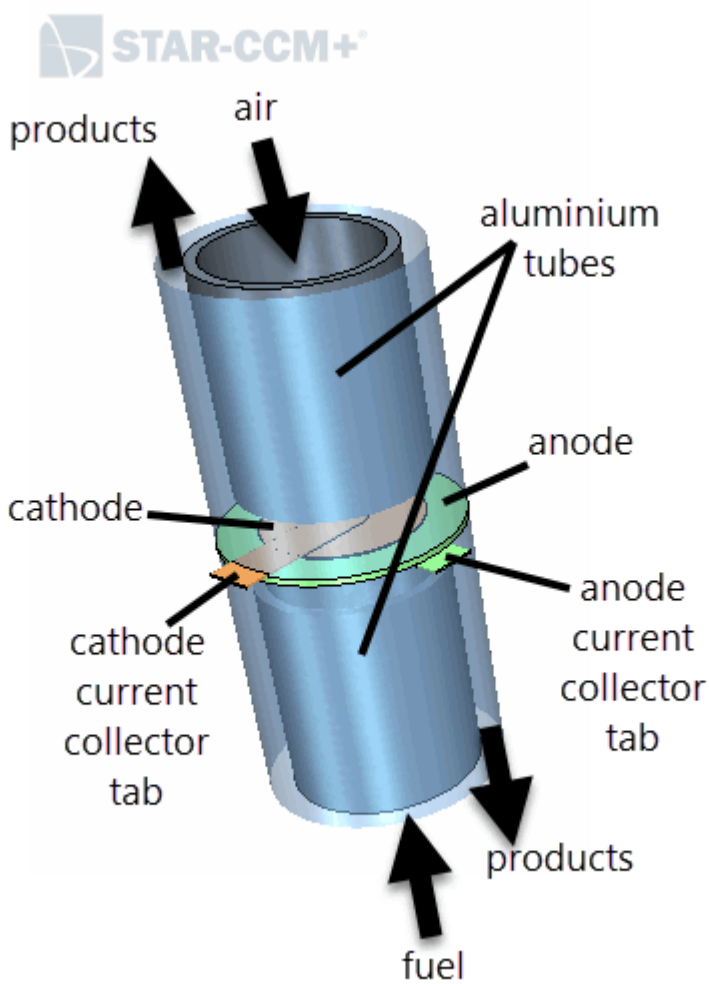


The planar button SOFC is placed inside a cylindrical casing. The complete geometry consists of an external casing which is intersected by the solid oxide electrolyte membrane and the anode and cathode current collectors. Within the volume of the external casing, there are internal aluminium tubes. Products of the reactions leave the SOFC from the space between the external casing and the tubes.

Components	Key Features
Solid Oxide Electrolyte Membrane	• Solid Shell Region • Thickness 0.06mm • Uses Solid Ion Model (identifies ions in the solid which participate in electrochemical reactions) • Uses Electrochemical Reactions Model (solid ions participate in reactions)
Anode and Cathode	Porous Regions (reactive species are not modeled in here—so no Electrochemical Reactions model required)

Components	Key Features
Anode and Cathode Current Collectors	Solid Regions which can conduct an electric current
Air and Fuel	• Fluid Regions • Multi-Component Gas Mixtures • Use Electrochemical Reactions Model (air and fuel species participate in reactions)
Tubes	Solid Regions

A gas phase fuel mixture (97% hydrogen and 3% water vapour) enters through the centre of the inner aluminum tube that is coincident with the anodic side of the button cell. A gas phase air mixture (21% oxygen and 79% nitrogen) enters through the centre of the inner aluminum tube that is coincident with the cathodic side of the button cell. The hydrogen in the fuel stream diffuses through the porous anode and reacts at the interface to the solid oxide electrolyte membrane. This interface represents the catalyst layers which include the triple phase boundaries, and will therefore be referred to as the TPB.



In STAR-CCM+, the direction of the electrochemical reaction current is positive in the direction of the solid to the fluid. In many electrochemical set-ups, this direction of reaction current is suitable as the solid is the electrode and the fluid is the electrolyte. However, in SOFCs, the solid oxide membrane is the electrolyte and the fluid is within the porous electrode. The Solid Ion model in STAR-CCM+ is in reduction current convention, leading to negative SOFC anode current density (electrons move from the anode to the cathode). Therefore, the anodic reaction—which is an oxidation reaction:

$$\mathrm{H_2 + O^{2-} \rightleftharpoons H_2O + 2e^-} \tag{4179}$$

is given in the reduction form:

$$\mathrm{H_2O + 2e^- \rightleftharpoons H_2 + O^{2-}} \tag{4180}$$

Prior to participating in the reaction at the anode, the oxide ions are transported across the solid oxide membrane. These oxide ions are created by the reaction that occurs at the surface between the cathode and the solid oxide electrolyte membrane:

$$\frac{1}{2}\mathrm{O_2} + 2e^- \rightleftharpoons \mathrm{O^{2-}} \tag{4181}$$

Contents:

Prerequisites
Loading an Existing Simulation
Defining Electrochemical Reaction Properties Using Field Functions
Selecting Physics Models
Setting Material Properties
Setting Initial Conditions
Defining Surface Reactions
Setting Up the Regions
Assigning the Surface Reactions
Setting Up the Boundary Conditions
Plotting the Conservation of Current, Energy, and Mass
Vizualizing Concentration and Molar Flux Distribution of Oxygen
Running the Simulation
Analyzing the Results
Summary
Appendix: User Field Function and Parameters Properties
Bibliography

Prerequisites

The instructions in this tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Loading an Existing Simulation

For this tutorial, you use an existing simulation file that contains pre-prepared geometry, mesh, regions, and interfaces. A selection of field functions are also supplied as a java macro file.

1. Copy the following files from the `[INSTALL_DIR]/doc/startutorialsdata/electrochemistry/data` directory to your working directory:
   - `SOFCbuttonCell_start.sim`
   - `sofc_setupFieldFunctions.java`
2. Start up a double-precision version of STAR-CCM+ in a manner that is appropriate to your working environment.
3. Load `SOFCbuttonCell_start.sim`.

   STAR-CCM+ provides feedback on the loading process in the *Output* window.
4. Open the Geometry scene and click (**Show All Meshes**).
5. Save the simulation as `SOFCbuttonCell.sim`.

Defining Electrochemical Reaction Properties Using Field Functions

You use custom field functions to define enthalpies, thermodynamic polynomial data, the equilibrium potential, and the equilibrium potential temperature derivative for all reactions within the fluid domains of the fuel cell.

For convenience, these field functions can be set up by running a macro.

To create the user field functions:

1. Select **File** > **Macro** > **Play Macro**.
2. In the *Open* dialog, select `sofc_setupFieldFunctions.java`.
3. To run the macro, click **Open**.
4. Save the simulation.

For detailed information about these field functions, see Appendix: User Field Function and Parameters Properties.

Selecting Physics Models

You define separate solid continua for the anode and cathode current collectors, the solid oxide electrolyte membrane (which separates the anode and cathode), and the tubes. Then you define separate fluid continua for the air and fuel domains.

It is necessary to maintain a continuous supply of fuel at the anode and air (oxygen) at the cathode. A solid continuum is set up for the aluminium tubes which supply the fuel and air.

1. Rename the **Continua** > **Physics 1** node to `Tubes`.
2. For the physics continuum, **Continua** > **Tubes**, select the following models in order:

Group Box	Model
Space	**Three Dimensional** (selected automatically)
Time	**Steady**
Material	**Solid**
Optional Models	**Segregated Solid Energy**
	Gradients (selected automatically)
Equation of State	**Constant Density**

3. Click **Close**.

The solid oxide electrolyte membrane separator is modeled as a solid shell region that separates the anode and cathode reactions.

4. Right-click the **Continua** node and select **New** > **Physics Continuum**. Rename the resulting **Physics 1** node to `Mem`. For this continuum, select the following models in order:

Group Box	Model
Space	**Shell Three Dimensional**
Time	**Steady**
Material	**Solid**
Optional Models	**Segregated Solid Energy**
	Gradients (selected automatically)
Equation of State	**Constant Density**
Optional Models	**Electrochemistry**
	Electrochemical Reactions (selected automatically)
	Electromagnetism (selected automatically)
	Electrodynamic Potential (selected automatically)
Optional Models	**Ohmic Heating**
	Electrochemical Reaction Heating
	Solid Ion

5. Click **Close**.
6. Select the **Mem** > **Models** > **Solid** > **Al** node and rename it `MEM`.

The anode is a porous solid ceramic material on the surface of which the electrochemical reactions take place.

7. Create a physics continuum named **CCanode**, and for it select the following models in order:

Group Box	Model
Space	**Three Dimensional**
Time	**Steady**
Material	**Solid**
Optional Models	**Segregated Solid Energy**
	Gradients (selected automatically)
Equation of State	**Constant Density**
Optional Models	**Electromagnetism**
Electromagnetism	**Electrodynamic Potential**
Optional Models	**Ohmic Heating**

8. Click **Close**.

Air is modeled as a multi-component gas mixture of oxygen and nitrogen. The Air continuum supplies oxygen to the porous cathode surface.

9. Create a physics continuum named **Air**, and for it select the following models in order:

Group Box	Model
Space	**Three Dimensional**
Time	**Steady**
Material	**Multi-Component Gas**
Reaction Regime	**Non-reacting**
Flow	**Segregated Flow**
	Gradients (selected automatically)
	Segregated Species (selected automatically)
Equation of State	**Constant Density**
Viscous Regime	**Laminar**
Optional Models	**Segregated Fluid Temperature**
	Electrochemistry
Electrochemistry	**Electrochemical Reactions**
	Electromagnetism (selected automatically)
	Electrodynamic Potential (selected automatically)
Optional Models	**Electrochemical Reaction Heating**
	Ohmic Heating

10. Click **Close**.

Fuel is modeled as a multi-component gas mixture of hydrogen and water vapour. The Fuel continuum supplies hydrogen to the porous anode surface.

11. Make a copy of the **Air** node and rename it `Fuel`.

The mass fraction gradient is high at the flow boundaries into the fuel and air continua. Since the flow boundary diffusion affects the mass flux of the components, it is necessary to turn off the flow boundary diffusion.

12. Multi-select the following models in the **Air** and **Fuel** continua:
    - **Air** > **Segregated Flow**
    - **Air** > **Segregated Fluid Temperature**
    - **Air** > **Segregated Species**
    - **Fuel** > **Segregated Flow**
    - **Fuel** > **Segregated Fluid Temperature**
    - **Fuel** > **Segregated Species**
13. Deactivate *Flow Boundary Diffusion*.

    The open-boundary fluxes $\dot{m}$ are now calculated as:

$$\dot{m} = \rho_i v\, A = Y_i \rho\, v\, A \tag{4182}$$

    - ρ_i is the density of species i
    - v is the velocity
    - A is the surface area of the boundary
    - Y_i is the mass fraction of species i
14. Save the simulation.

Setting Material Properties

For all continua, select material components and adjust material properties.

1. Select the material components for the **Air** continuum:
    a) Right-click the **Models** > **Multi-Component Gas** > **Gas Components** node and choose **Select Mixture Components**.
    b) In the *Select Mixture Components* dialog, expand the **Material Databases** > **Standard** > **Gases** node and select **N2 (Nitrogen)** and **O2 (Oxygen)**.
    c) Click **Apply** then **Close**.
2. For the **Fuel** continuum, follow the same steps as above, but specify the gas components as **H2 (Hydrogen)** and **H2O (Water)**.
3. Expand the following nodes:
    - **Air** > **Models** > **Multi-Component Gas** > **Gas Components**
    - **Fuel** > **Models** > **Multi-Component Gas** > **Gas Components**
4. Multi-select the following nodes in the **Air** and **Fuel** continua and set *Method* to **Thermodynamic Polynomial Data**:
    - **N2** > **Material Properties** > **Specific Heat**

- **O2** > **Material Properties** > **Specific Heat**
- **H2** > **Material Properties** > **Specific Heat**
- **H2O** > **Material Properties** > **Specific Heat**

Since air and fuel are not electrically conducting, their conductivity is set to zero. The solid parts of the porous anode and cathode, which are conducting, are defined later when setting up the porous regions.

5. Multi-select the following nodes in the **Air** and **Fuel** continua and set *Value* to `0.0` S/m:
   - **Air** > **Models** > **Multi-Component Gas** > **Material Properties** > **Electrical Conductivity** > **Constant**
   - **Fuel** > **Models** > **Multi-Component Gas** > **Material Properties** > **Electrical Conductivity** > **Constant**

The anode and cathode are made of porous ceramic materials. It is important to make sure that the correct electrical and thermal conductivities are specified for the material properties.

6. Rename the **CCanode** > **Models** > **Solid** > **Al** node `Anode`.
7. Edit the **Solid** > **Anode** > **Material Properties** node and set the following values:

Node	Property	Setting
Density		
└ **Constant**	*Value*	`7450.0` kg/m^3
Electrical Conductivity		
└ **Constant**	*Value*	`9400000.0` S/m
Specific Heat		
└ **Constant**	*Value*	`600.0` J/kg-K
Thermal Conductivity		
└ **Constant**	*Value*	`27.0` W/m-K

The anode current collector and cathode current collector have the same material properties.

8. Make a copy of the **CCanode** node and rename it `CCcathode`.
9. Rename the **CCcathode** > **Models** > **Solid** > **Anode** node `Cathode`.
10. Edit the **Continua** > **Mem** > **Models** > **Solid** > **MEM** > **Material Properties** node and set the following values:

Node	Property	Setting
Density		
└ **Constant**	*Value*	`6000.0` kg/m^3
Electrical Conductivity	*Method*	**Field Function**
└ **Field Function**	*Scalar Function*	**ElectrolyteConductivity**
Specific Heat		

Node	Property	Setting
└ **Constant**	*Value*	`400.0` J/kg-K
Thermal Conductivity		
└ **Constant**	*Value*	`2.7` W/m-K

Define the electrical species components within the SOFC membrane:

11. Expand the **Mem** > **Models** > **Solid Ion** node:
    a) Right-click the **Electrochemical Species Components** node and choose **Select Mixture Components**.
    b) In the *Select Mixture Components* dialog, expand the **Material Databases** > **Standard** > **ElectrochemicalSpecies** node and select **OH- (Hydroxide)**.
    c) Click **Apply**, then **Close**.
    d) Rename the **Electrochemical Species Components** > **OH-** node `O-2`.
    e) Select the **O-2** > **Material Properties** > **Charge Number** > **Constant** node and set *Value* to `-2.0`.
    f) Expand the **Elemental Composition** > **Elemental Composition Method** > **Atoms** node and delete the **H** sub-node.
12. Save the simulation.

Setting Initial Conditions

Set the initial temperature, electric potential values, and mole fractions of the fuel and air streams.

1. Multi-select the **Initial Conditions** > **Static Temperature** node of all continua and set *Value* to `973.15` K.
2. Select the **CCcathode** > **Initial Conditions** > **Electric Potential** node and set *Value* to `0.62 V`.
3. In addition, for the **Air** and **Fuel** continua, set the following initial conditions:
   - **Air**:

Node	Property	Setting
Initial Conditions		
└ **Electric Potential**	*Value*	`0.62 V`
└ **Species Specification**	*Method*	**Mole fraction**
└ **Species Mole Fraction**	*Value*	Click ... (Custom Editor) and set the following values: • N2: `0.79` • O2: `0.21`

   - **Fuel**:

Node	Property	Setting
Initial Conditions		
└ **Species Specification**	*Method*	**Mole fraction**
└ **Species Mole Fraction**	*Value*	Click ... (Custom Editor) and set the following values: • H2: `0.97` • H2O: `0.03`

4. Save the Simulation.

Defining Surface Reactions

Electrochemical reactions that occur at the surfaces between the electrodes and the electrolyte are defined manually.

For the **Air** continuum, define the reaction which occurs at the surface between the cathode and the electrolyte, Eqn. (4181).

1. Expand the **Continua** > **Air** > **Models** node.
2. Right-click the **Electrochemical Reactions** > **Surface Mechanism Manager** node and select **New Surface Mechanism**.
3. Rename the **Surface Mechanism Manager** > **SurfaceMechanism** node to `Cathodic`.
4. Right-click the **Cathodic** > **Models** > **Reacting Surface** > **Reactions** node and select **New Reaction**.
5. Rename the **Reactions** > **Reaction 1** node to `O2 + 4e- <=> 2O-2`.
6. Set up the sub-nodes of the **O2 + 4e- <=> 2O-2** node as follows:

Node	Action
Reactants	1. Right-click and select **Add Reactant** > **Gas Mixture** > **O2**. 2. Right-click and select **Add Reactant** > **e-**.
└ **O2 (Gas Mixture)**	Set *Rate Exponent* to `0.5`.
└ **e-**	Set *Stoich. Coeff* to `4.0` .
Products	Right-click and select **Add Product** > **Electrochemical Species Mixture** > **O-2**.
└ **O-2 (Electrochemical Species Mixture)**	Set *Stoich. Coeff* to `2.0`.
Properties	
└ **Equilibrium Potential**	Set *Select Method* to **Nernst Equilibrium Potential**.
└ **Nernst Standard Potential**	1. Set *Method* to *Field Function*.

Node	Action
	2. Set *Scalar Function* to **CathodeEquilibriumPotential-Nernst**.
└ **Equilibrium Potential T Derivative**	
└ **Nernst Standard Potential T Derivative**	1. Set *Method* to **Field Function**. 2. Set *Scalar Function* to **CathodeEquilibriumPotentialTder**
└ **Reaction Formulation**	
└ **Butler-Volmer** > **Alpha A**	Set *Value* to `1.0`.
└ **Butler-Volmer** > **Alpha C**	Set *Value* to `1.0`.
└ **Specific Exchange Current**	
└ **User Specific Exchange Current**	1. Set *Method* to **Field Function**. 2. Set *Scalar Function* to **CathodeExchangeCurrent**

For the **Fuel** continuum, set up the reaction that occurs at the surface between the anode and the electrolyte Eqn. (4179).

7. Copy the **Air** > **Models** > **Electrochemical Reactions** > **Surface Mechanism Manager** > **Cathodic** node.
8. Right-click the **Fuel** > **Models** > **Electrochemical Reactions** > **Surface Mechanism Manager** node and select **Paste**.
9. Rename the **Fuel** > **Models** > **Electrochemical Reactions** > **Surface Mechanism Manager** > **Cathodic** node to `Anodic`.
10. Rename the **Anodic** > **Models** > **Reacting Surface** > **Reactions** > **O2 + 4e- <=> 2O-2** node to `H2O + 2e- <=> H2 + O-2`.

    Note: The reactant species, O2, is not copied over with the mechanism because it is not specified as a gas component in the continuum that is receiving the copy.

11. Set up the sub-nodes of the **H2O + 2e- <=> H2 + O-2** node as follows:

Node	Action
Reactants	Add the reactant **Gas Mixture** > **H2O**.
└ **H2O**	Set *Rate Exponent* to `0.5`.
└ **e-**	Set *Stoich. Coeff* to `2.0`.
Products	Add the product **Gas Mixture** > **H2**.
└ **H2 (Gas Mixture)**	Set *Rate Exponent* to `0.5`.
└ **O-2 (Electrochemical Species Mixture)**	Set *Stoich. Coeff* to `1.0`.

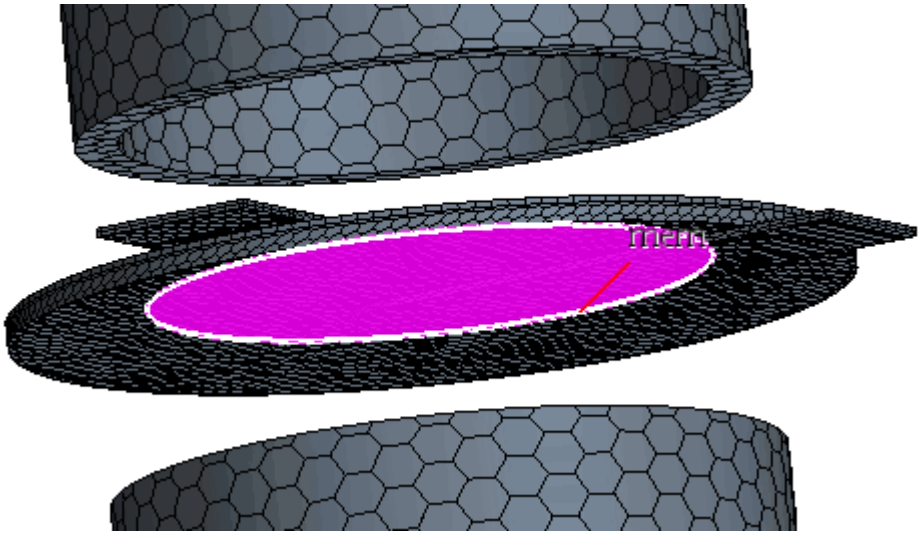
Node	Action
Properties	
└ **Equilibrium Potential**	
└ **Nernst Standard Potential**	Set *Scalar Function* to **AnodeEquilibriumPotential-Nernst**.
└ **Equilibrium Potential T Derivative**	
└ **Nernst Standard Potential T Derivative**	Set *Scalar Function* to **AnodeEquilibriumPotentialTder**.
└ **Specific Exchange Current**	
└ **User Specific Exchange Current**	Set *Scalar Function* to **AnodeExchangeCurrent**.

12. Save the simulation.

Setting Up the Regions

Create the shell region for the solid electrolyte membrane, specify which continua each of the regions use, and define the properties and values for each region.

1. Right-click the **Interfaces** > **anode/cathode** node and select **Insert Shell Region Interface**.
2. Rename the **Regions** > **mem-Interface-mem-Interface shell interface** node to `mem`.



3. Edit the **Regions** node and set the following properties:

Node	Property	Setting
air	*Physics Continuum*	**Air**
anode	*Physics Continuum*	**Fuel**

Node	Property	Setting
cathode	*Physics Continuum*	**Air**
currentcollector-anode	*Physics Continuum*	**CCanode**
currentcollector-cathode	*Physics Continuum*	**CCcathode**
fuel	*Physics Continuum*	**Fuel**
mem	*Continuum*	**Mem**
tube-anode	*Physics Continuum*	**Tubes**
tube-cathode	*Physics Continuum*	**Tubes**

4. Select the **Regions** > **mem** > **Physics Values** > **Shell Thickness** node and set *Value* to `6.0E-5` m.
5. Expand the **Regions** node, then multi-select the **anode** and **cathode** nodes and set *Type* to **Porous Region**.

The solid parts of the porous anode and cathode are conducting, therefore, values must be specified for the solid electrical and thermal conductivities.

6. Edit the **Regions** > **anode** > **Physics Values** node and set the following properties:

Node	Property	Setting
Porosity	*Value*	`0.3`
Porous Viscous Resistance	*Method*	**Isotropic Tensor**
└**Isotropic Tensor** > **Isotropic Component**	*Value*	`887.789` kg/m^3-s
Solid Electrical Conductivity	*Method*	**Isotropic Tensor**
└**Isotropic Tensor** > **Isotropic Component**	*Value*	`30300.0` S/m
Solid Thermal Conductivity		
└**Isotropic Tensor** > **Isotropic Component**	*Value*	`6.0` W/m-K

7. Edit the **Regions** > **cathode** > **Physics Values** node and set the following properties:

Node	Property	Setting
Porosity	*Value*	`0.7`
Porous Viscous Resistance	*Method*	**Isotropic Tensor**
└**Isotropic Tensor** > **Isotropic Component**	*Value*	`1855.08` kg/m^3-s
Solid Electrical Conductivity	*Method*	**Isotropic Tensor**
└**Isotropic Tensor** > **Isotropic Component**	*Value*	`12800.0` S/m

Node	Property	Setting
Solid Thermal Conductivity		
└ **Isotropic Tensor** > **Isotropic Component**	*Value*	`10.0` W/m-K

8. Save the Simulation.

Assigning the Surface Reactions

The cathodic and anodic reactions take place at the cathodic and anodic triple phase boundaries respectively. The triple phase boundaries are represented by interfaces at both sides of the solid oxide electrolyte membrane—between the porous anode and the porous cathode.

1. Rename the interface nodes **In-place 1** and **In-place 2** to `anode-TPB` and `cathode-TPB`, respectively.
2. Select the **anode-TPB** > **Physics Conditions** > **Surface Mechanism Option** node and set *Mechanism* to **Anodic**.
3. Select the **cathode-TPB** > **Physics Conditions** > **Surface Mechanism Option** node and set *Mechanism* to **Cathodic**.

Setting Up the Boundary Conditions

Each region has specific boundary conditions to define.

1. Edit the **Regions** > **air** > **Boundaries** > **inlet-air** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Species Specification**	*Method*	**Mole fraction**
Physics Values		
└ **Species Mole Fraction**	*Value*	• N2: `0.79` • O2: `0.21`
└ **Static Temperature**	*Value*	`973.15` K
└ **Velocity Magnitude**	*Value*	`0.0869639` m/s

2. Edit the **Regions** > **air** > **Boundaries** > **outlet-air** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Species Specification**	*Method*	**Mole fraction**
Physics Values		

Node	Property	Setting
└ **Species Mole Fraction**	*Value*	• N2: 0.79 • O2: 0.21
└ **Static Temperature**	*Value*	973.15 K

3. Edit the **Regions** > **fuel** > **Boundaries** > **inlet-hydrogen** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Species Specification**	*Method*	**Mole fraction**
Physics Values		
└ **Species Mole Fraction**	*Value*	• H2: 0.97 • H2O: 0.03
└ **Static Temperature**	*Value*	973.15 K
└ **Velocity Magnitude**	*Value*	0.0869639 m/s

4. Edit the **Regions** > **fuel** > **Boundaries** > **outlet-hydrogen** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Species Specification**	*Method*	**Mole fraction**
Physics Values		
└ **Species Mole Fraction**	*Value*	• H2: 0.97 • H2O: 0.03
└ **Static Temperature**	*Value*	973.15 K

5. Edit the **Regions** > **currentcollector-cathode** > **Boundaries** > **contact** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Electric Potential Specification**	*Method*	**Electric Potential**
Physics Values		
└ **Electric Potential**	*Value*	0.62 V

6. Edit the **Regions** > **currentcollector-anode** > **Boundaries** > **contact** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Electric Potential Specification**	*Method*	**Electric Potential**
Physics Values		
└ **Electric Potential**	*Value*	0.0 V

The zero value for electric potential corresponds to grounded conditions.

7. Save the simulation.

Plotting the Conservation of Current, Energy, and Mass

In this part of the tutorial, you set up reports and generate monitors and plots. These plots check the balance of current, energy, and mass, respectively, throughout the solid oxide fuel cell (SOFC).

You display the results on the following plots:

- Current Conservation

  The current conservation plot compares the electric current on the anode and cathode current collector boundaries to the current on the anode and cathode triple phase boundaries (TPB).

- Energy Conservation

  The energy that is generated from the SOFC is not entirely converted to external electrical power. Some energy is lost as heat at the outlet. The energy conservation plot balances the external power that is generated Eqn. (4186) with the flow boundary enthalpy losses Eqn. (4185).

  The non-isothermal energy conservation equation is:

$$\sum_{i,IN} \widehat{H}_i\left(T_{IN}\right) Y_i \dot{m} - \sum_{i,OUT} \widehat{H}_i\left(T_{OUT}\right) Y_i \dot{m} + \sum_{i,\mathrm{Reaction}} \widehat{H}_i\left(T_R\right) \dot{m}_i + Q_O + Q_E = 0 \tag{4183}$$

  where T_{IN} and T_{OUT} are the temperatures at the inlet and outlet respectively. $\dot{m}$ is the mass flux and Y_i is the mass fraction of species i. The enthalpy of formation, $\widehat{H}_i$ of species i, that is associated with the mass that is consumed by the reaction, is converted to internal electrochemical heat (Q_E), internal Ohmic heat (Q_O), and external electric power (P):

$$\sum_{i,\mathrm{Reaction}} \widehat{H}_i\left(T_R\right) \dot{m}_i + Q_O + Q_E + P = 0 \tag{4184}$$

  combining Eqn. (4183) and Eqn. (4184), gives:

$$P = \sum_{i,IN} \widehat{H}_i\left(T_{IN}\right) Y_i \dot{m} - \sum_{i,OUT} \widehat{H}_i\left(T_{OUT}\right) Y_i \dot{m} \tag{4185}$$

  The electrical power that is available to the external circuit is calculated by the anode-cathode potential difference, U, and the total current that is available to the external circuit, I:

$$P = IU \tag{4186}$$

  STAR-CCM+ calculates the ohmic heating by accounting for electric potential losses throughout every electrically or ionically conductive substrate:

$$Q_O = \iiint \sigma \left| \nabla \phi \right|^2 dV \tag{4187}$$

where σ is the electrical conductivity and $\nabla\phi$ is the electric potential gradient.

STAR-CCM+ calculates the electrochemical heat that is released from a reaction, i, using:

$$Q_E = \iint i\left(\eta + T\frac{\partial E_{eq}^{ox}}{\partial T}\bigg|_P\right) \tag{4188}$$

where η is the overpotential and $\frac{\partial E_{eq}^{ox}}{\partial T}\bigg|_P$ is the equilibrium potential temperature derivative.

- Mass Conservation

  The mass conservation plot compares the mass that is gained by the gas streams to the mass that is produced by electrochemical reactions at the TPB.

Create the reports:

1. Right-click the **Reports** node and select **New Report** > **Surface Integral**.
2. Rename the **Surface Integral 1** node to `Current_Anode`.
3. Select **Current_Anode** and set the following properties:

Property	Setting
Field Function	**Boundary Specific Electric Current**
Parts	**Regions** > **currentcollector-anode** > **Boundaries** > **contact**

4. In the same way, create 11 more *Surface Integral* reports and name them according to the table in the next step.
5. Set the following properties for the reports:

Report	Field Function	Parts
Current_AnodeTPB	**Boundary Specific Electric Current**	**Regions** > **anode** > **Boundaries** > **mem-Interface [anode-TPB]**
Current_Cathode	**Boundary Specific Electric Current**	**Regions** > **currentcollector-cathode** > **Boundaries** > **contact**
Current_CathodeTPB	**Boundary Specific Electric Current**	**Regions** > **cathode** > **Boundaries** > **mem-Interface [cathode-TPB]**
Electric_Power	**ElectricPowerFlux**	• **Regions** > **currentcollector-anode** > **Boundaries** > **contact** • **Regions** > **currentcollector-cathode** > **Boundaries** > **contact**
Enthalpy_Gain	**Boundary Heat Flux**	• **Regions** > **air** > **Boundaries** > **inlet-air** • **Regions** > **air** > **Boundaries** > **outlet-air**

Report	Field Function	Parts
		• **Regions** > **fuel** > **Boundaries** > **inlet-hydrogen** • **Regions** > **fuel** > **Boundaries** > **outlet-hydrogen**
Mass_Gain_H2	**Mass Flux H2**	• **Regions** > **fuel** > **Boundaries** > **inlet-hydrogen** • **Regions** > **fuel** > **Boundaries** > **outlet-hydrogen**
Mass_Gain_H2O	**Mass Flux H2O**	• **Regions** > **fuel** > **Boundaries** > **inlet-hydrogen** • **Regions** > **fuel** > **Boundaries** > **outlet-hydrogen**
Mass_Gain_O2	**Mass Flux O2**	• **Regions** > **air** > **Boundaries** > **inlet-air** • **Regions** > **air** > **Boundaries** > **outlet-air**
Mass_Reaction_H2	**Boundary Species Electrochemical Reaction Flux** > **Boundary Species Electrochemical Reaction Flux of H2**	**Regions** > **anode** > **Boundaries** > **mem-Interface [anode-TPB]**
Mass_Reaction_H2O	**Boundary Species Electrochemical Reaction Flux** > **Boundary Species Electrochemical Reaction Flux of H2O**	**Regions** > **anode** > **Boundaries** > **mem-Interface [anode-TPB]**
Mass_Reaction_O2	**Boundary Species Electrochemical Reaction Flux** > **Boundary Species Electrochemical Reaction Flux of O2**	**Regions** > **cathode** > **Boundaries** > **mem-Interface [cathode-TPB]**

Create monitors and plots for the reports:

6. Multi-select the **Current_Anode**, **Current_AnodeTPB**, **Current_Cathode**, and **Current_CathodeTPB** nodes.
7. Right-click one of the selected nodes and select **Create Monitor and Plot from Report**.
8. In the *Create Plot From Reports* dialog, click **Single Plot**.

9. Rename the **Reports Plot** node to `Current Conservation`.
10. In the same way, create monitors and a plot for each of the following groups of reports:
    - For **Electric_Power** and **Enthalpy_Gain**, create a plot named `Energy Conservation`.
    - For **Mass_Gain_H2**, **Mass_Gain_H2O**, **Mass_Gain_O2**, **Mass_Reaction_H2**, **Mass_Reaction_H2O**, and **Mass_Reaction_O2**, create a plot named `Mass Conservation`.
11. For each of the three conservation plots, update the *Title* property to match the name of the plot.
12. Expand the **Monitors** node, then multi-select the following nodes and set *Normalization Option* to **Manual**:
    - **Current_Anode Monitor**
    - **Current_CathodeTPB Monitor**
    - **Enthalpy_Gain Monitor**
    - **Mass_Gain_H2 Monitor**
    - **Mass_Gain_H2O Monitor**
    - **Mass_Gain_O2 Monitor**
    - **Mass_Reaction_H2 Monitor**
    - **Mass_Reaction_H2O Monitor**
    - **Mass_Reaction_O2 Monitor**
13. Expand the following nodes, multi-select each of the **Manual Normalization** sub-nodes, and set *Normalization Factor* to `-1.0`:
    - **Current_Anode Monitor**
    - **Current_CathodeTPB Monitor**
    - **Enthalpy_Gain Monitor**
    - **Mass_Gain_H2 Monitor**
    - **Mass_Gain_H2O Monitor**
    - **Mass_Gain_O2 Monitor**

    Setting the normalization factor to -1 inverts the monitor values.
14. For the following monitor nodes, set *Normalization Factor* of the **Manual Normalization** sub-nodes as follows:
    - **Mass_Reaction_H2 Monitor**: `-0.496032`
    - **Mass_Reaction_H2O Monitor**: `-0.0555093`
    - **Mass_Reaction_O2 Monitor**: `-0.03125`
15. Save the simulation.

Vizualizing Concentration and Molar Flux Distribution of Oxygen

To compare the mole fraction of oxygen alongside the consumption of oxygen on the cathode triple-phase boundary (TPB), you set up a scalar scene for each quantity.

1. Create two scalar scenes and rename them `Mole Fraction O2` and `Reaction Flux O2`.
2. Expand the **Mole Fraction O2** > **Displayers** > **Scalar 1** node and set the following properties:

Node	Property	Setting
Parts	*Parts*	**Regions** > **cathode** > **Boundaries** > **mem-Interface [cathode-TPB]**
Scalar Field	*Function*	**Mole Fraction** > **Mole Fraction of O2**
Color Bar	*Orientation*	**Vertical**

3. Right-click the **Mole Fraction O2** > **Displayers** node and select **New Displayer** > **Geometry**.
4. Edit the **Mole Fraction O2** > **Displayers** > **Geometry 1** node and set the following properties:

Node	Property	Setting
Geometry 1	*Surface*	Activated
	Color Mode	**Constant**
└ **Parts**	*Parts*	All boundaries and feature curves in the following regions: • **air** • **currentcollector-anode** • **currentcollector-cathode** • **tube-anode** • **tube-cathode**

5. Copy the **Geometry 1** node and paste it in the **Reaction Flux O2** > **Displayers** node.

The **Boundary Species Electrochemical Reaction Flux of O2** field function is used to display the consumption of oxygen in kmoles / area at the cathode TPB.

6. Expand the **Reaction Flux O2** > **Displayers** > **Scalar 1** node and set the following properties:

Node	Property	Setting
Parts	*Parts*	**Regions** > **cathode** > **Boundaries** > **mem-Interface [cathode-TPB]**
Scalar Field	*Function*	**Boundary Species Electrochemical Reaction Flux** > **Boundary Species Electrochemical Reaction Flux of O2**
Color Bar	*Orientation*	**Vertical**

7. Save the Simulation.

Running the Simulation

For highly diffusive problems like this, increasing the under-relaxation factors of the solvers helps accelerate convergence. With electromagnetic forces being strong forces in their nature, it is important to have well converged simulation results for the electrodynamic potential before converging the other physical quantities like the flow velocities, or species concentrations. In STAR-CCM+, convergence can be achieved by using under-relaxation factor ramps for the latter models.

Specify solver settings and stopping criteria, and run the simulation:

1. Edit the **Solvers** node and set the following properties:

Node	Property	Setting
Segregated Flow		
⌊ **Velocity**		
⌊ **Under-Relaxation Factor Ramp**	*Ramp Method*	**Linear Ramp**
⌊ **Linear Ramp**	*Start Iteration*	4
	End Iteration	5
	Initial Value	0.0
Segregated Species	*Under-Relaxation Factor*	0.995
⌊ **Under-Relaxation Factor Ramp**	*Ramp Method*	**Linear Ramp**
⌊ **Linear Ramp**	*Start Iteration*	8
	End Iteration	10
	Initial Value	0.0
Segregated Energy	*Fluid Under-Relaxation Factor*	1.0
	Solid Under-Relaxation Factor	1.0
⌊ **Fluid Under-Relaxation Factor Ramp**	*Ramp Method*	**Linear Ramp**
⌊ **Linear Ramp**	*Start Iteration*	8
	End Iteration	10
	Initial Value	0.0
⌊ **Solid Under-Relaxation Factor Ramp**	*Ramp Method*	**Linear Ramp**
⌊ **Linear Ramp**	*Start Iteration*	8

Node	Property	Setting
	End Iteration	`10`
	Initial Value	`0.0`

Note that velocity and pressure solvers must generally operate with consistent under-relaxation parameters, see section Segregated Flow Guidelines. The above settings avoid specialized ramps for the pressure solver by ramping the velocity solver discontinously from 0 to the final (default) value, where a zero under-relaxation implies a frozen flow solver.

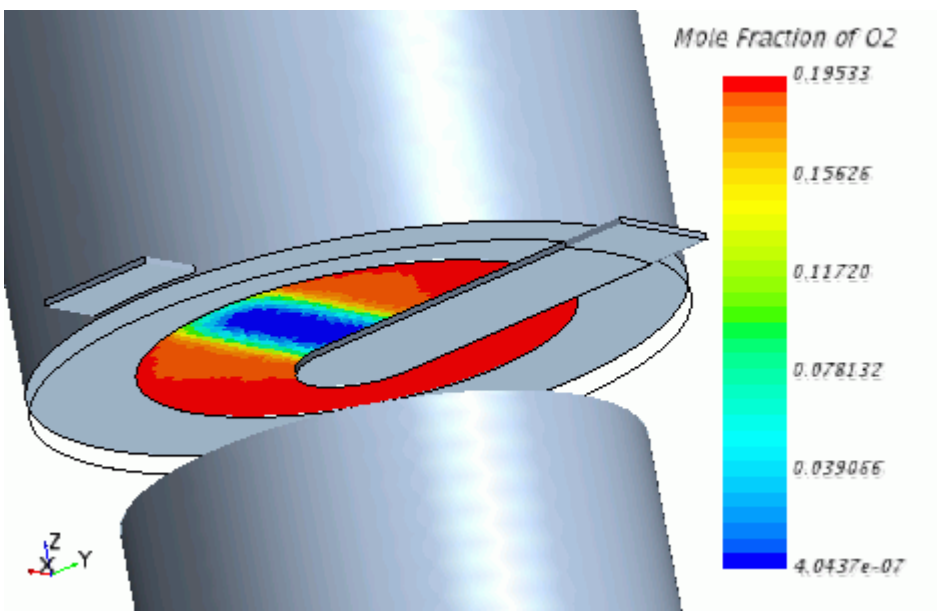
2. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to `500`.

3. Click (**Run**).

Analyzing the Results

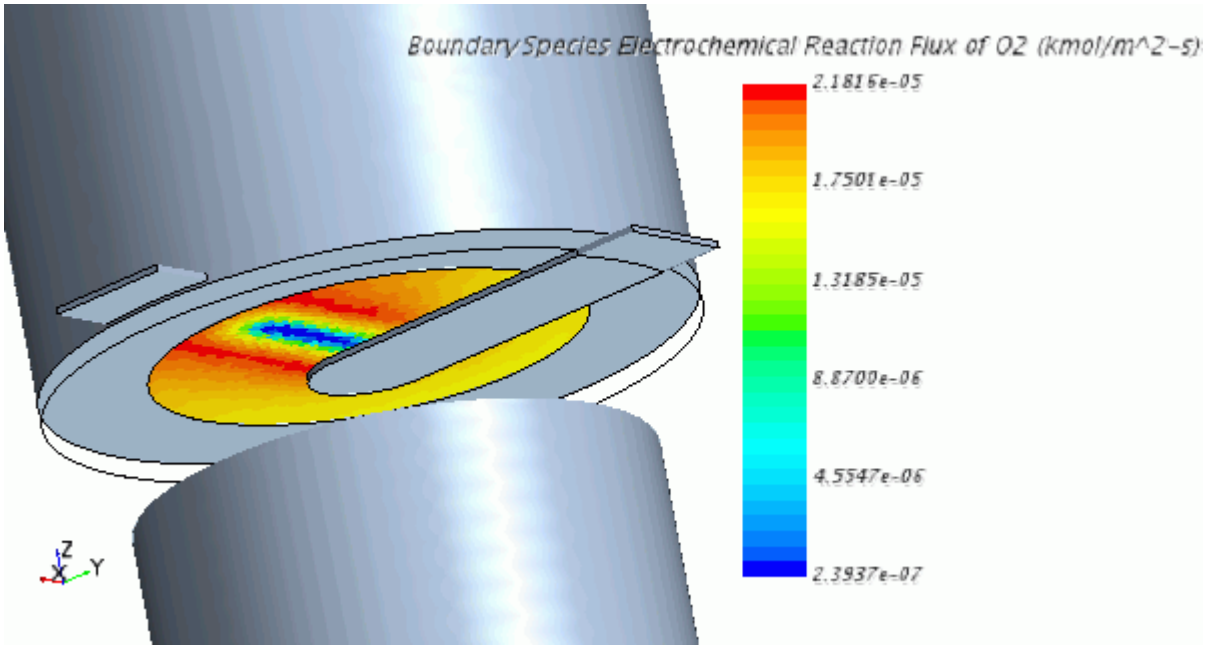
When the simulation is complete, review the scenes and plots.

1. Open the **Mole Fraction O2** scene.



The concentration of oxygen is low across a significant area of the porous cathode.

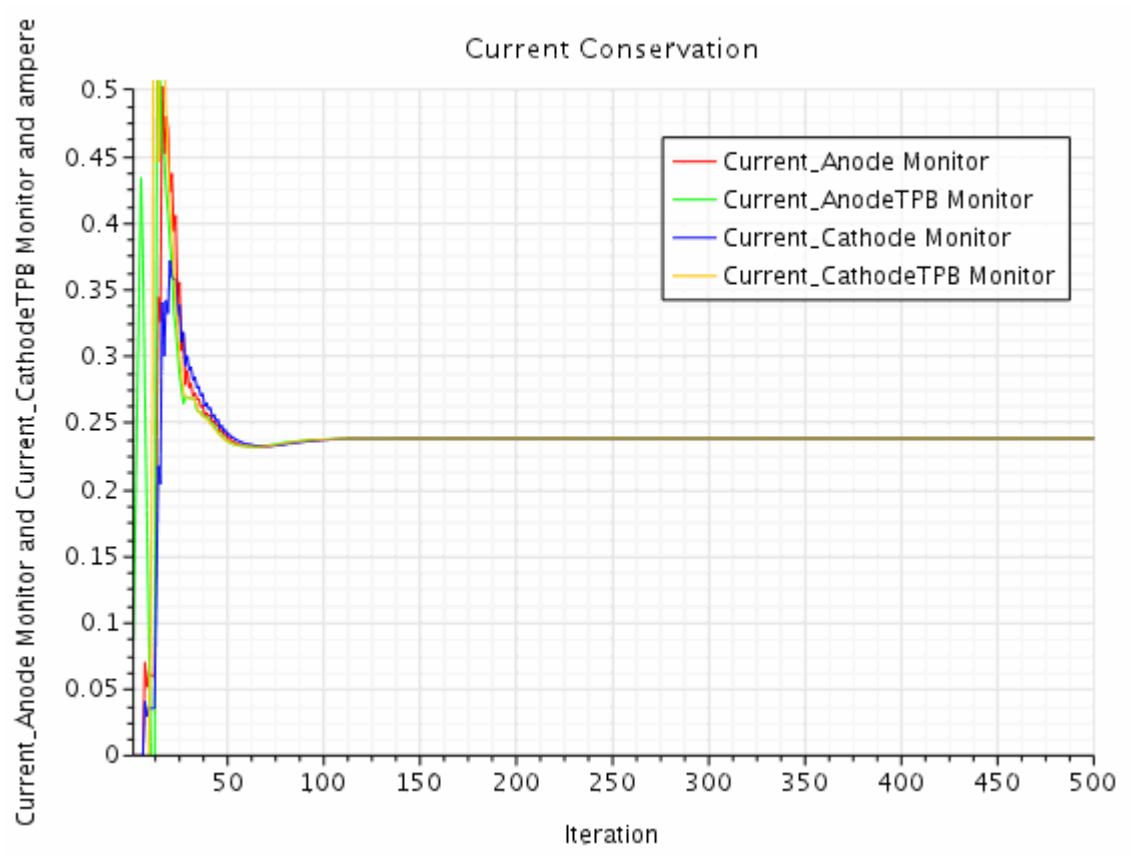
2. Open the **Reaction Flux O2** scene.



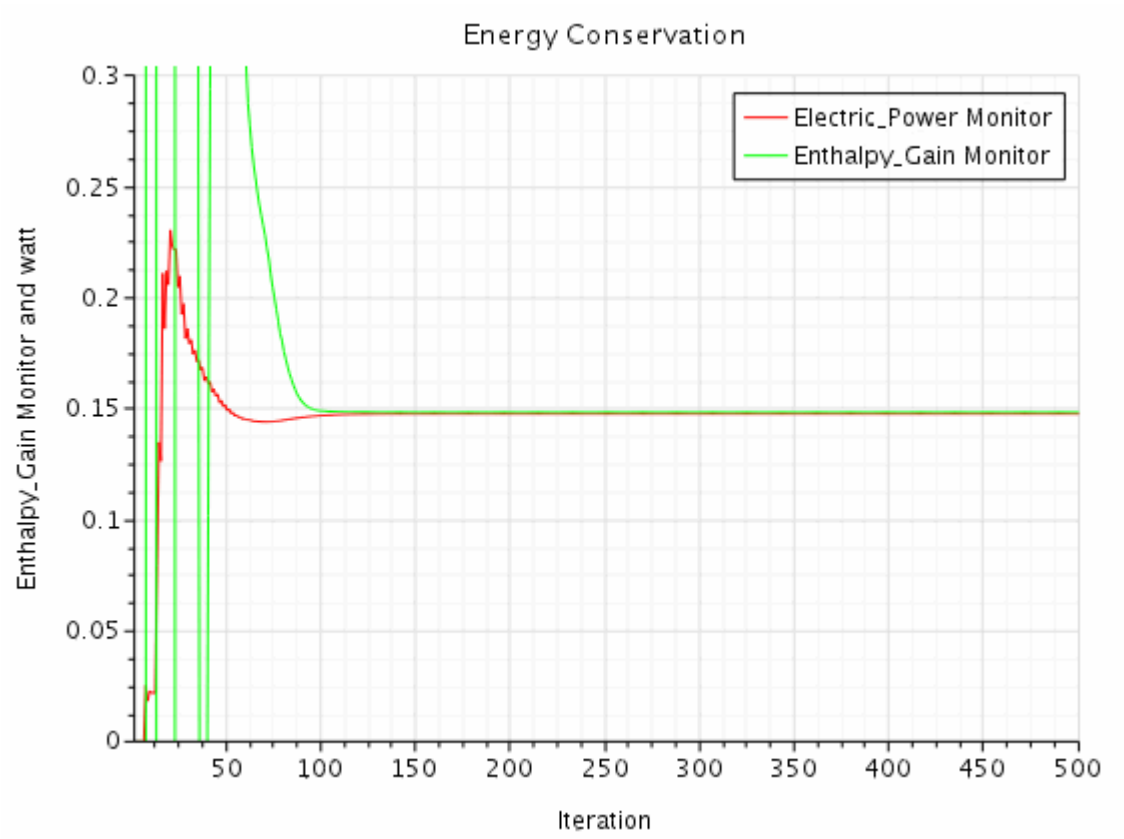
In the area of the cathode with a low concentration of oxygen, the reaction rate with respect to oxygen is decreased. The decreased reaction rate shows that the rate is limited by the concentration of oxygen, which is limited by the diffusion rate of oxygen through the porous cathode.

3. Open the **Current Conservation**, **Energy Conservation**, and **Mass Conservation** plots.
4. To improve the visibility of the results on the conservation plots, adjust the Y-axis limits.
   a) Expand the **Plots** node.
   b) Select the **Current Conversion** > **Axes** > **Left Axis** node and set *Minimum* to `0.0` and *Maximum* to `0.5`.
   c) Select the **Energy Conversion** > **Axes** > **Left Axis** node and set *Minimum* to `0.0` and *Maximum* to `0.3`.
   d) Select the **Mass Conversion** > **Axes** > **Left Axis** node and set *Minimum* to `-5e-8` and *Maximum* to `5e-8`.

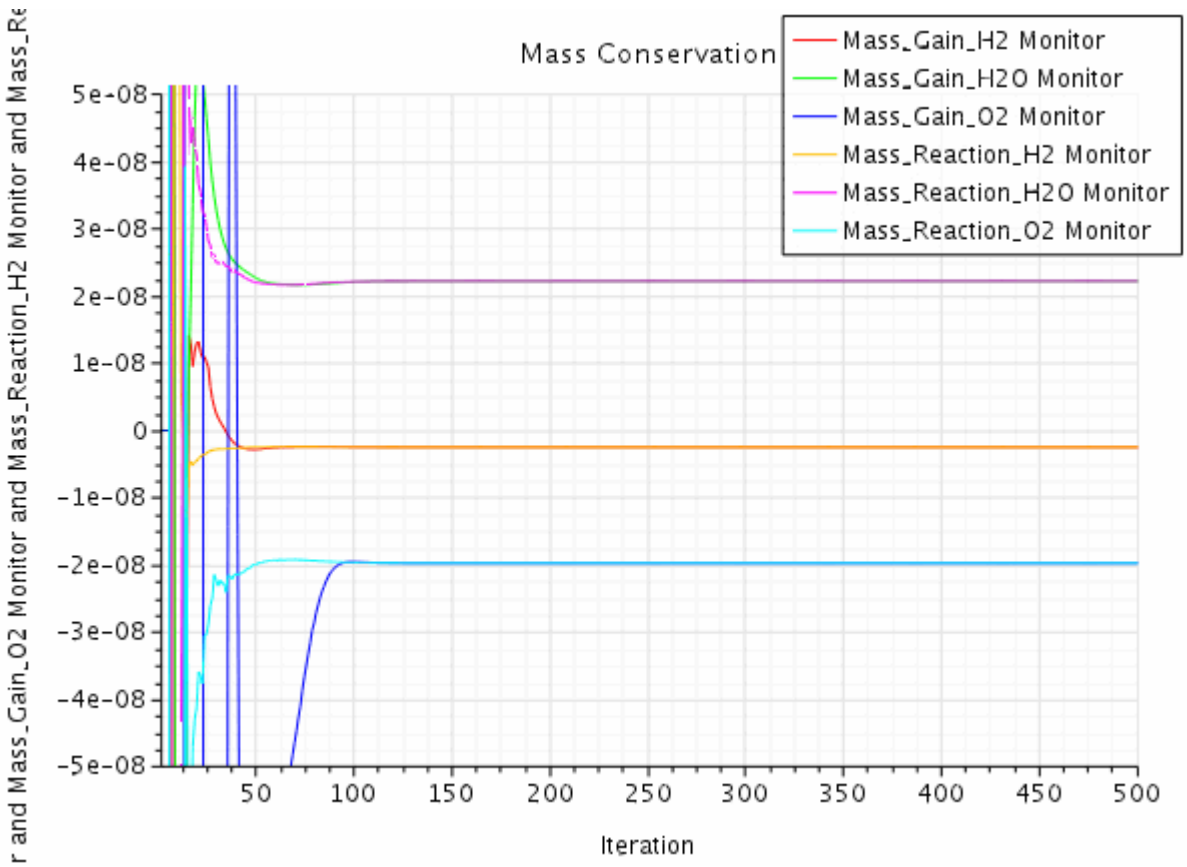
   The conservation plots appear as shown:



The current through all cross sections displays an equal value, indicating that a good conservation of electrical currents is warranted.



The external power that is generated (Eqn. (4186)) balances with the flow boundary enthalpy losses (Eqn. (4185)).



For each of the species, H2, H2O, and O2, their respective masses created by the electrochemical reactions (shown by the mass reaction monitors) balance with the masses of each respective species entering the fluid streams (shown by the mass gain monitors).

5. Save the Simulation.

Summary

This tutorial demonstrated electrochemistry for a solid oxide fuel cell.

The features that were demonstrated include:

- Defining the SOFC specific user field functions for electrochemical reaction properties.
- Selecting electrochemical surface reaction models.
- Defining and assigning electrochemical surface reactions.
- Specifying region and boundary conditions.
- Preparing reports and monitor plots.
- Examining the results in the plots.

Appendix: User Field Function and Parameters Properties

Names, dimensions, definitions, and formulation of the user-defined field functions and parameters that are used in the Solid Oxide Fuel Cell tutorial.

Global Parameters

Faraday Constant

Dimensions: **ElectricCharge/Quantity**

Value: $F = 9.6485 \qquad [\mathrm{sA/kmol}]$

UniversalGasConstant

Dimensions: **Energy/(Temperature Quantity)**

Value: $R_u = 8314.4621 \qquad [\mathrm{J/K\ mol}]$

User-Defined Field Functions

Thermodynamic Polynomial Data: Enthalpy Terms

Description: Terms representing thermodynamic polynomial data for the enthalpies of hydrogen, water, and oxygen.

Dimensions: Dimensionless

Anode_HpolyH2

Formulation:

$$H_{poly,\ H_2} = \frac{H_{H_2}}{R_u T}\ W_{H_2} \tag{4189}$$

Definition: `${EnthalpyH2} / ${UniversalGasConstant} / $Temperature * $MolecularWeightH2`

Anode_HpolyH2O

Formulation:

$$H_{poly,\ H_2O} = \frac{H_{H_2O}}{R_u T}\ W_{H_2O} \tag{4190}$$

Definition: `${EnthalpyH2O} / ${UniversalGasConstant} / $Temperature * $MolecularWeightH2O`

Cathode_HpolyO2

Formulation:

$$H_{poly,\ O_2} = \frac{H_{O_2}}{R_u T}\ W_{O_2} \tag{4191}$$

Definition: `${EnthalpyO2} / ${UniversalGasConstant} / $Temperature * $MolecularWeightO2`

Thermodynamic Polynomial Data: Temperature Derivatives of Enthalpy Terms

Description: Partial derivative of the corresponding enthalpy term for each species, with respect to temperature. In this case, the derivative of enthalpy with respect to temperature is the specific heat.

Dimensions: Dimensionless

AnodeHpolyH2derivative

Formulation:

$$\frac{\partial\left(H_{poly,H_2}T\right)}{\partial T} = \frac{C_{p,\,H_2}}{R_u}W_{H_2} \tag{4192}$$

Definition: `${SpecificHeatH2} / ${UniversalGasConstant} * $MolecularWeightH2`

AnodeHPolyH2Oderivative

Formulation:

$$\frac{\partial\left(H_{poly,H_2O}T\right)}{\partial T} = \frac{C_{p,H_2O}}{R_u}W_{H_2O} \tag{4193}$$

Definition: `${SpecificHeatH2O} / ${UniversalGasConstant} * $MolecularWeightH2O`

CathodeHpolyO2derivative

Formulation:

$$\frac{\partial\left(H_{poly,\,O_2}T\right)}{\partial T} = \frac{C_{p,\,O_2}}{R_u}W_{O_2} \tag{4194}$$

Definition: `${SpecificHeatO2} / ${UniversalGasConstant} * $MolecularWeightO2`

Thermodynamic Polynomial Data: Entropy Terms

Description: Thermodynamic polynomial data for the entropies of hydrogen, water, and oxygen. Parameters are given for temperatures above and below 1000K.

Dimensions: Dimensionless

Anode_SpolyH2

Formulation:

$$\frac{S_{H_2}}{R_u} = a_1 \ln T + a_2 T + \frac{a_3 T^2}{2} + \frac{a_4 T^3}{3} + \frac{a_5 T^4}{4} + a_7 \tag{4195}$$

Definition: `($Temperature<1000) ? 3.29812 * log($Temperature) + 8.24944E-4 * $Temperature -8.14302E-7 * pow($Temperature,2) / 2 -9.47543E-11 * pow($Temperature,3) / 3 + 4.13487E-13 * pow($Temperature,4) / 4 -3.29409: 2.99142 * log($Temperature) + 7.00064E-4 * $Temperature -5.63383E-8 * pow($Temperature,2) / 2 -9.23158E-12 * pow($Temperature, 3) / 3 + 1.58275E-15 * pow($Temperature,4) / 4 -1.35511`

Anode_SpolyH2O

Formulation:

$$\frac{S_{H_2O}}{R_u} = a_1 \ln T + a_2 T + \frac{a_3 T^2}{2} + \frac{a_4 T^3}{3} + \frac{a_5 T^4}{4} + a_7 \tag{4196}$$

Definition: `($Temperature<1000) ? 3.38684 * log($Temperature) + 0.00347498 * $Temperature -6.3547E-6 * pow($Temperature,2) / 2 + 6.96858E-9 * pow($Temperature,3) / 3 -2.50659E-12 * pow($Temperature,4) / 4 + 2.59023: 2.67215 * log($Temperature) + 0.00305629 * $Temperature -8.73026E-7 * pow($Temperature,2) / 2 + 1.201E-10 * pow($Temperature, 3) / 3 -6.39162E-15 * pow($Temperature,4) / 4 + 6.86282`

Cathode_SpolyO2

Formulation:

$$\frac{S_{O_2}}{R_u} = a_1 \ln T + a_2 T + \frac{a_3 T^2}{2} + \frac{a_4 T^3}{3} + \frac{a_5 T^4}{4} + a_7 \tag{4197}$$

Definition: `($Temperature<1000) ? 3.21294 * log($Temperature) + 0.0011274899999999999 * $Temperature -5.75615E-7 * pow($Temperature, 2) / 2 + 1.31388E-9 * pow($Temperature,3) / 3 -8.76855E-13 * pow($Temperature,4) / 4 + 6.03474: 3.69758 * log($Temperature) + 6.1352E-4 * $Temperature -1.25884E-7 * pow($Temperature,2) / 2 + 1.77528E-11 * pow($Temperature,3) / 3 -1.13644E-15 * pow($Temperature,4) / 4 + 3.`

Thermodynamic Polynomial Data: Temperature Derivatives of Entropy Terms

Description: Partial derivatives of the entropies of hydrogen, water, and oxygen, with respect to temperature.

Dimensions: **/Temperature**

AnodeSpolyH2derivative

Formulation:

$$\frac{\partial\left(\frac{S_{H_2}}{R_u}\right)}{\partial T} = \frac{a_1}{T} + a_2 + a_3 T + a_4 T^2 + a_5 T^3 \tag{4198}$$

Definition: `($Temperature<1000) ? 3.29812 * 1. / $Temperature + 8.24944E-4 -8.14302E-7 * $Temperature -9.47543E-11 * pow($Temperature,2) + 4.13487E-13 * pow($Temperature,3): 2.99142 * 1. / $Temperature + 7.00064E-4 -5.63383E-8 * $Temperature -9.23158E-12 * pow($Temperature,2) + 1.58275E-15 * pow($Temperature,3)`

AnodeSpolyH2Oderivative

Formulation:

$$\frac{\partial\left(\frac{S_{H_2O}}{R_u}\right)}{\partial T} = \frac{a_1}{T} + a_2 + a_3T + a_4T^2 + a_5T^3 \tag{4199}$$

Definition: `($Temperature<1000) ? 3.38684 * 1. / $Temperature + 0.00347498 -6.3547E-6 * $Temperature + 6.96858E-9 * pow($Temperature,2) -2.50659E-12 * pow($Temperature,3): 2.67215 * 1. / $Temperature + 0.00305629 -8.73026E-7 * $Temperature + 1.201E-10 * pow($Temperature,2) -6.39162E-15 * pow($Temperature,3)`

CathodeSpolyO2derivative

Formulation:

$$\frac{\partial\left(\frac{S_{O_2}}{R_u}\right)}{\partial T} = \frac{a_1}{T} + a_2 + a_3T + a_4T^2 + a_5T^3 \tag{4200}$$

Definition: `($Temperature<1000) ? 3.21294 * 1. / $Temperature + 0.0011274899999999999 -5.75615E-7 * $Temperature + 1.31388E-9 * pow($Temperature,2) -8.76855E-13 * pow($Temperature,3): 3.69758 * 1. / $Temperature + 6.1352E-4 -1.25884E-7 * $Temperature + 1.77528E-11 * pow($Temperature,2) -1.13644E-15 * pow($Temperature,3)`

Equilibrium Potentials

Description: The solution-dependent contributions that are not considered by the concentration-dependent Nernst equilibrium potential term as given by Eqn. (3665).

Dimensions: **ElectricPotential**

AnodeEquilibriumPotential-Nernst

Formulation:

$$N_{E,a} = \frac{R_uT}{2F}\left(H_{poly,\ H_2} - H_{poly,\ H_2O} + \frac{S_{H_2O}}{R_u} - \frac{S_{H_2}}{R_u}\right) \tag{4201}$$

Definition: `${UniversalGasConstant} * $Temperature / 2 / ${FaradayConstant} * (${Anode_HpolyH2}-${Anode_HpolyH2O}+${Anode_SpolyH2O}-${Anode_SpolyH2})`

CathodeEquilibriumPotential-Nernst

Formulation:

$$N_{E,c} = -\frac{R_u T}{4F}\left[H_{poly,\,O_2} - S_{O_2} + \ln\left(\frac{X_{O_2}W_{O_2} + X_{N_2}W_{N_2}}{\rho}\right) + \ln\left(\frac{P}{\rho_{ref}}\right)\right] \tag{4202}$$

Definition: `-${UniversalGasConstant} * $Temperature / 4 / ${FaradayConstant} * (${Cathode_HpolyO2}-${Cathode_SpolyO2} + log(($MoleFractionO2 * $MolecularWeightO2+$MoleFractionN2 * $MolecularWeightN2) / $Density) +log(${AbsolutePressure} / 101325))`

Equilibrium Potential Temperature Derivatives

Description: The temperature derivatives of the Nernst equilibrium potentials for the anode and cathode.

Dimensions: **ElectricPotential/Temperature**

AnodeEquilibriumPotentialTder

Formulation:

$$\left.\frac{\partial E_{eq,a}^{ox}}{\partial T}\right|_P = \left.\frac{\partial E_{0,a}^{ox}}{\partial T}\right|_P + \frac{R_u T}{2F}\ln\left(\frac{X_{H2}}{X_{H2,ref}}\right) - \frac{R_u T}{2F}\ln\left(\frac{X_{H2O}}{X_{H2O,ref}}\right) \tag{4203}$$

Definition: `${UniversalGasConstant} / 2 / ${FaradayConstant} * (${AnodeHPolyH2derivative}-${AnodeHPolyH2Oderivative}+${Anode_SpolyH2O}-${Anode_SpolyH2} + $Temperature * (${AnodeSPolyH2Oderivative}-${AnodeSPolyH2derivative}))`

CathodeEquilibriumPotentialTder

Formulation:

$$\left.\frac{\partial E_{eq,c}^{ox}}{\partial T}\right|_P = \left.\frac{\partial E_{0,c}^{ox}}{\partial T}\right|_P - \frac{R_u T}{4F}\ln\left(\frac{X_{O2}P}{101325\,[Pa]}\right) \tag{4204}$$

Definition: `-${UniversalGasConstant} / 4 / ${FaradayConstant} * (${CathodeHPolyO2derivative} -${Cathode_SpolyO2} + log(($MoleFractionO2 * $MolecularWeightO2+$MoleFractionN2 * $MolecularWeightN2) / $Density) -$Temperature * ${CathodeSPolyO2derivative}+log(${AbsolutePressure} / 101325))`

Exchange Current Densities

Description: The exchange current densities of the anode and cathode, as required by the Butler-Volmer equation.

Dimensions: **Current/Length^2**

AnodeExchangeCurrent

Formulation:

$$j_{0,a} = j^*_{0,a}\left(\frac{X_{H2}}{0.8}\right)^{0.5}\left(\frac{X_{H2O}}{0.2}\right)^{0.5} \tag{4205}$$

Definition: `7746 / pow($AnodeXToCconversion * $AnodeXToCconversion * .8 * .2,.5)`

CathodeExchangeCurrent

Formulation:

$$j_{0,c} = j^*_{0,c}\left(\frac{X_{O2}}{0.21}\right)^{0.5} \tag{4206}$$

Definition: `10010 / pow($CathodeXToCconversion * .21,.5)`

Mass Fluxes

Description: Mass fluxes of hydrogen, nitrogen, oxygen, and water.

Dimensions: **Mass/Length^2-Time**

Mass Flux H2

Formulation:

$$\dot{m}_{H_2} = W_{H_2}\frac{\upsilon_{H_2} j}{nF} = Y_{H_2}\dot{m} \tag{4207}$$

Definition: `${MassFractionH2} * ${MassFlux}`

Mass Flux N2

Formulation:

$$\dot{m}_{N_2} = W_{N_2}\frac{\upsilon_{N_2} j}{nF} = Y_{N_2}\dot{m} \tag{4208}$$

Definition: `${MassFractionN2} * ${MassFlux}`

Mass Flux O2

Formulation:

$$\dot{m}_{O_2} = W_{O_2}\frac{\upsilon_{O_2} j}{nF} = Y_{O_2}\dot{m} \tag{4209}$$

Definition: `${MassFractionO2} * ${MassFlux}`

Mass Flux H2O

Formulation:

$$\dot{m}_{H_2O} = W_{H_2O}\frac{\upsilon_{H_2O} j}{nF} = Y_{H_2O}\dot{m} \tag{4210}$$

Definition: `${MassFractionH2O} * ${MassFlux}`

Conversion Factors

Description: The conversion factors (CF) from mole fraction to molar concentration (required by STAR-CCM+) for the species in the respective mixture.

Dimensions: **Quantity/Length^3**

AnodeXToCconversion

Formulation:

$$CF_{an} = \frac{\rho}{X_{H_2} W_{H_2} + X_{H_2O} W_{H_2O}} \tag{4211}$$

Definition: `$Density / (${MoleFractionH2} * $MolecularWeightH2 + $MoleFractionH2O * $MolecularWeightH2O)`

CathodeXToCconversion

Formulation:

$$CF_{ca} = \frac{\rho}{X_{N_2} W_{N_2} + X_{O_2} W_{O_2}} \tag{4212}$$

Definition: `$Density / ($MoleFractionN2 * $MolecularWeightN2 + $MoleFractionO2 * $MolecularWeightO2)`

ElectricPowerFlux

Description: Electric power.

Dimensions: **Power/Length^2**

Formulation:

$$P = UI \tag{4213}$$

Definition: `${ElectricPotential} * ${BoundarySpecificElectricCurrent}`

ElectrolyteConductivity

Description: The conductivity of the solid oxide electrolyte.

Dimensions: **ElectricConductance/Length**

Formulation:

$$\sigma_m = \frac{1}{\left(0.3685 + 0.002838 \times e^{\frac{10300}{T}}\right) \cdot 100} \tag{4214}$$

Definition: `1. / ((0.3685 + 0.002838 * exp(10300 / $Temperature)) / 100.0)`

SOFC Tutorial Nomenclature

Nomenclature	
C_p	Heat capacity

Nomenclature	
CF	Conversion factor
E_{eq}	Equilibrium potential
E_0	Nernst standard potential
F	Faraday constant, 96485339.924 C/kmol
H	Enthalpy
I	Total current that is available to the external circuit, J
j	Reaction current density, A/m²
$\dot{m}_i$	Mass flux of species i
N_E	The solution-dependent contributions that are not considered by the concentration-dependent Nernst equilibrium potential.
P	Pressure (or Power, J/s)
R_u	Universal gas constant, 8314.4621 J / K $kmol^{-1}$
ρ	Density
σ	Conductivity, S/cm
S	Entropy
T	Temperature, K
U	Anode-cathode potential difference
v_i	Stoichiometric coefficient of species i
W	Molecular weight
X	Mole fraction
Y	Mass fraction

Bibliography

[812] *O. R. J. G. a. J. M. V. Costa-Nunes, "Comparison of the performance of Cu–CeO 2–YSZ and Ni–YSZ composite SOFC anodes with H 2, CO, and syngas," Journal of power sources, vol. 141, no. 2, pp. 241-249, 2005.*

[813] *F. Scholz, "Thermodynamics of electrochemical reactions," Electroanalytical Methods. Springer Berlin Heidelberg, pp. 11-31, 2010.*

[814] *A. D. S. B. B. a. J. G. P. Le, "Validation of a Solid Oxide Fuel Cell Model on the International Energy Agency Benchmark Case with Hydrogen Fuel," Fuel Cells, vol. 15, no. 1, pp. 27-41, 2015.*

[815] *S. G. Bratsch, "Standard electrode potentials and temperature coefficients in water at 298.15 K," Journal of Physical and Chemical Reference Data , vol. 18, no. 1, pp. 1-21, 1989.*

Battery

The tutorials in this set illustrate various STAR-CCM+ features for setting up a battery model:

Contents:

Battery Simulation Module: Cell Thermal Analysis
Cylindrical Cells: Cell Thermal Analysis
Li-Ion Battery Cell Model: Cell Electrochemistry Analysis

Battery Simulation Module: Cell Thermal Analysis

This tutorial introduces the battery modeling capabilities of STAR-CCM+. It simulates the battery heat generation that occurs during the first 90 seconds of a discharge cycle.

The tutorial takes you through the steps that are required to run a battery analysis in STAR-CCM+ using files that are created in Battery Design Studio® (BDS).

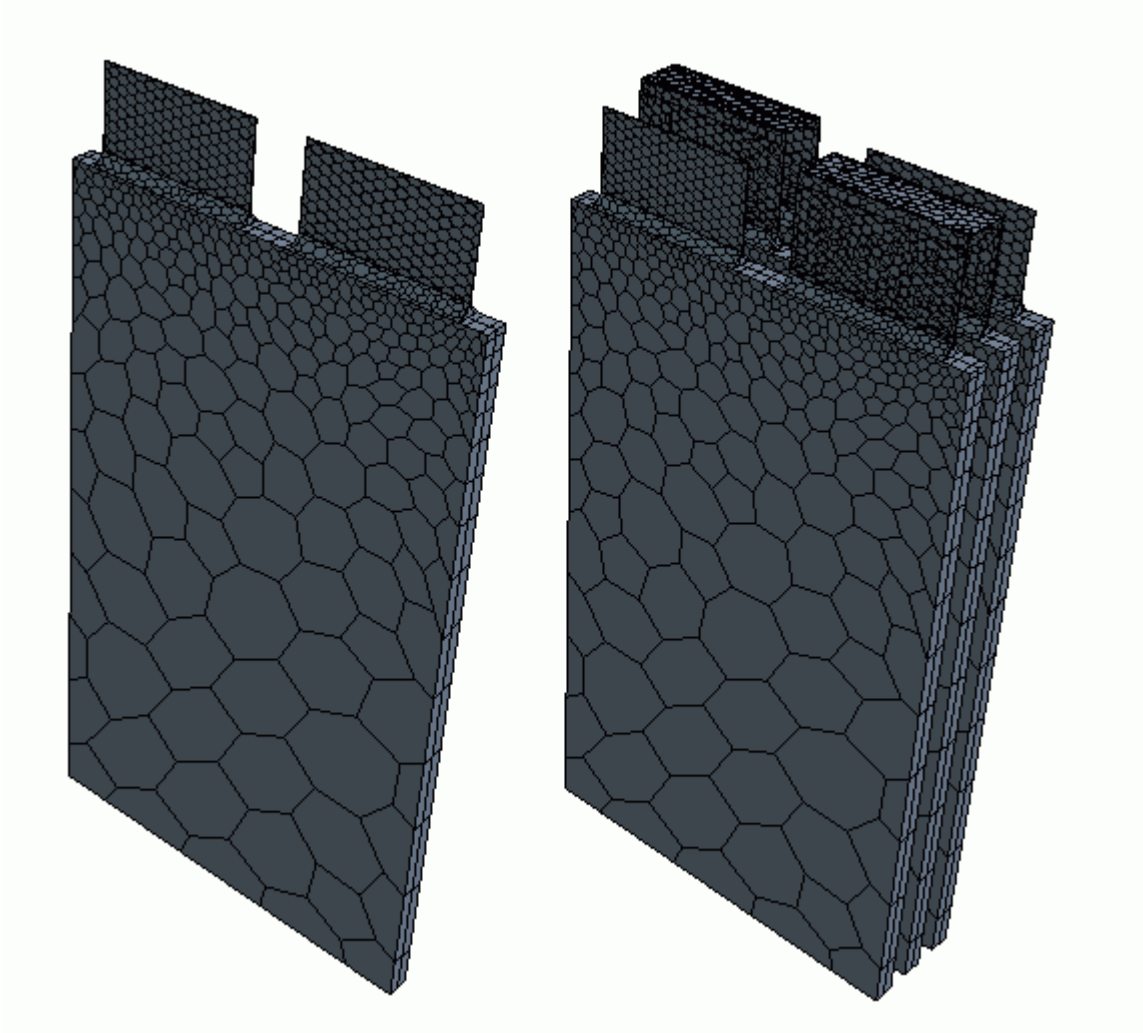
Additional licenses are required to use the battery modeling capabilities:

- For battery modeling, STAR-CCM+ requires a `batterysim` license in addition to the other license features.
- BDS, which is supplied for use with the STAR-CCM+ battery modeling capability, requires either a `batterysim` or a `batterydesignstudio` license.

The tutorial is divided into three parts:

1. Preparing files in BDS
2. Running a STAR-CCM+ simulation with a Single Cell
3. Running a STAR-CCM+ simulation with 3 Cells in Series

The battery heat generation that occurs during the first 90 seconds of a discharge cycle is simulated. A simple scenario with a single cell battery is run first. The simulation is then rerun after modifying the battery, adding two more cells in series.



Both simulations use a voltage plot to monitor the battery discharge during the cycle. A scalar scene displays how battery heat generation is distributed on the battery cells.

Contents:

Prerequisites
Preparing Files in Battery Design Studio®
Running a STAR-CCM+ Simulation with a Single Cell
Running a STAR-CCM+ Simulation with 3 Cells in Series
Summary

Prerequisites

Before attempting this tutorial, make sure that you are familiar with the following techniques.

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+

Technique	Tutorial
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Preparing Files in Battery Design Studio®

Prepare battery data in BDS and import it into STAR-CCM+.

Battery Design Studio (BDS) is a tool for the design, simulation, and analysis of batteries. Battery data from BDS is imported into STAR-CCM+ and can be used to set up complex scenarios, such as simulating cooling by including a fluid region around the battery cells.

To use a battery cell in a STAR-CCM+ simulation, the following files must be generated in BDS:

- `.tbm`—a text battery model file, containing all the information about the battery. This information includes construction, sizing, materials, and physics models.
- `.prg`—a program file, containing details of the battery load cycle

With STAR-CCM+, it is possible to import battery data from BDS and set up complex scenarios that include a fluid region around the battery cells to simulate cooling.

In the following topics, `tbm` and `prg` files are generated for a lithium-ion battery cell in BDS.

If you do not wish to work through the steps involving BDS, you can jump directly to Running a STAR-CCM+ simulation with a Single Cell.

Contents:

Creating a Battery Cell
Creating a Program File

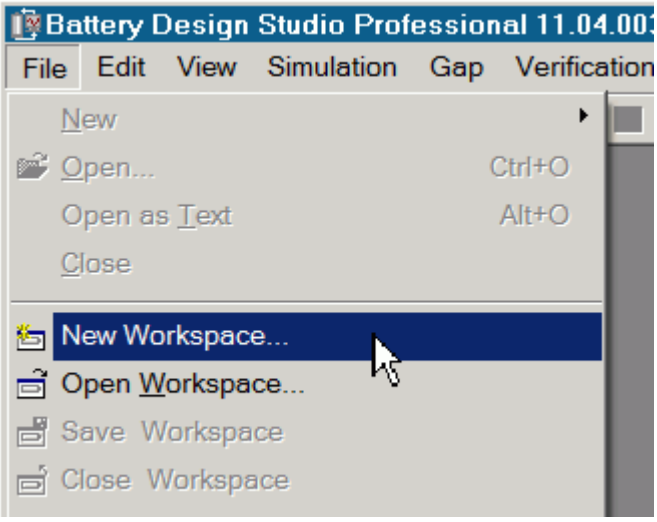
Creating a Battery Cell

Create a battery cell in BDS.

Use the Li-Ion stack cell that is available in BDS as a starting point.

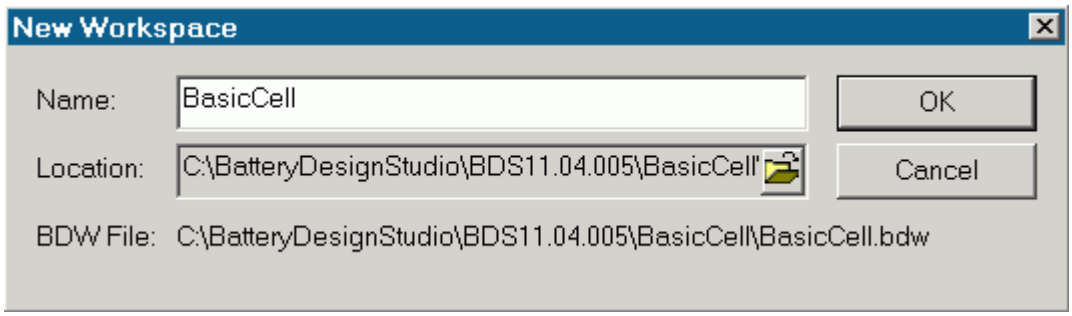
To create a battery cell in BDS:

1. Launch BDS.
2. Create a workspace by selecting **File** > **New Workspace**.



The *New Workspace* dialog appears.

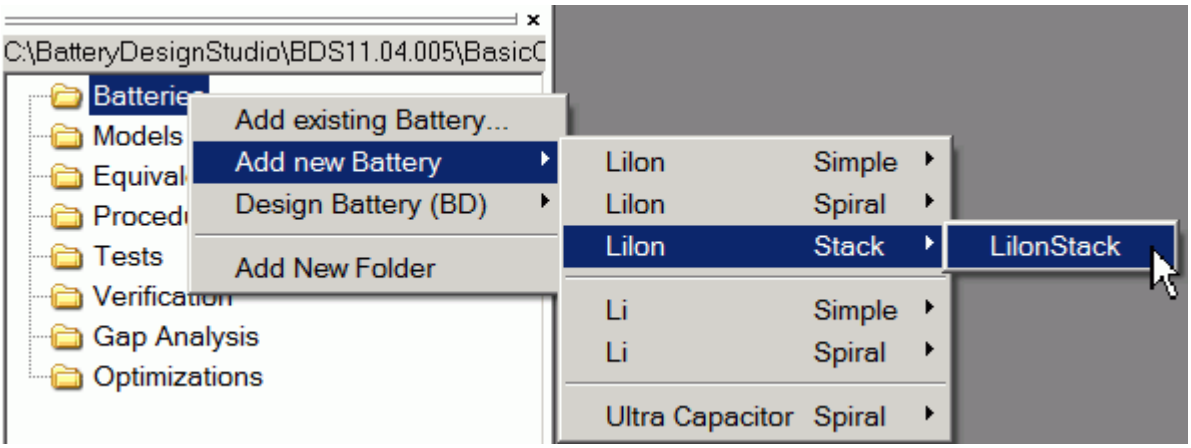
3. Enter `BasicCell` as the *Name* in the textbox, and select an appropriate location.



4. Click **OK**.
5. If the location does not exist, confirm the directory creation.

Components of the battery are created using the tree on the left of the BDS interface. Create a battery cell.

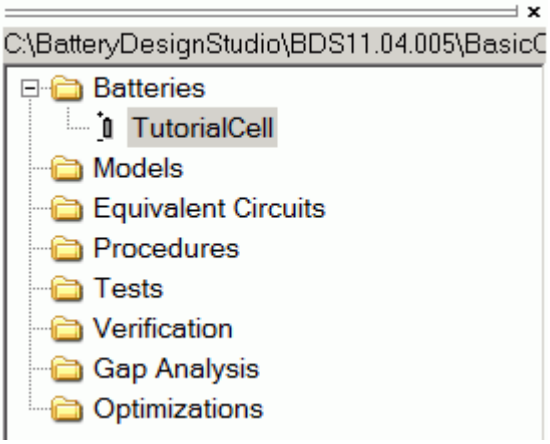
6. Right-click the **Batteries** folder and select **Add New Battery** > **Lilon Stack** > **LilonStack**.



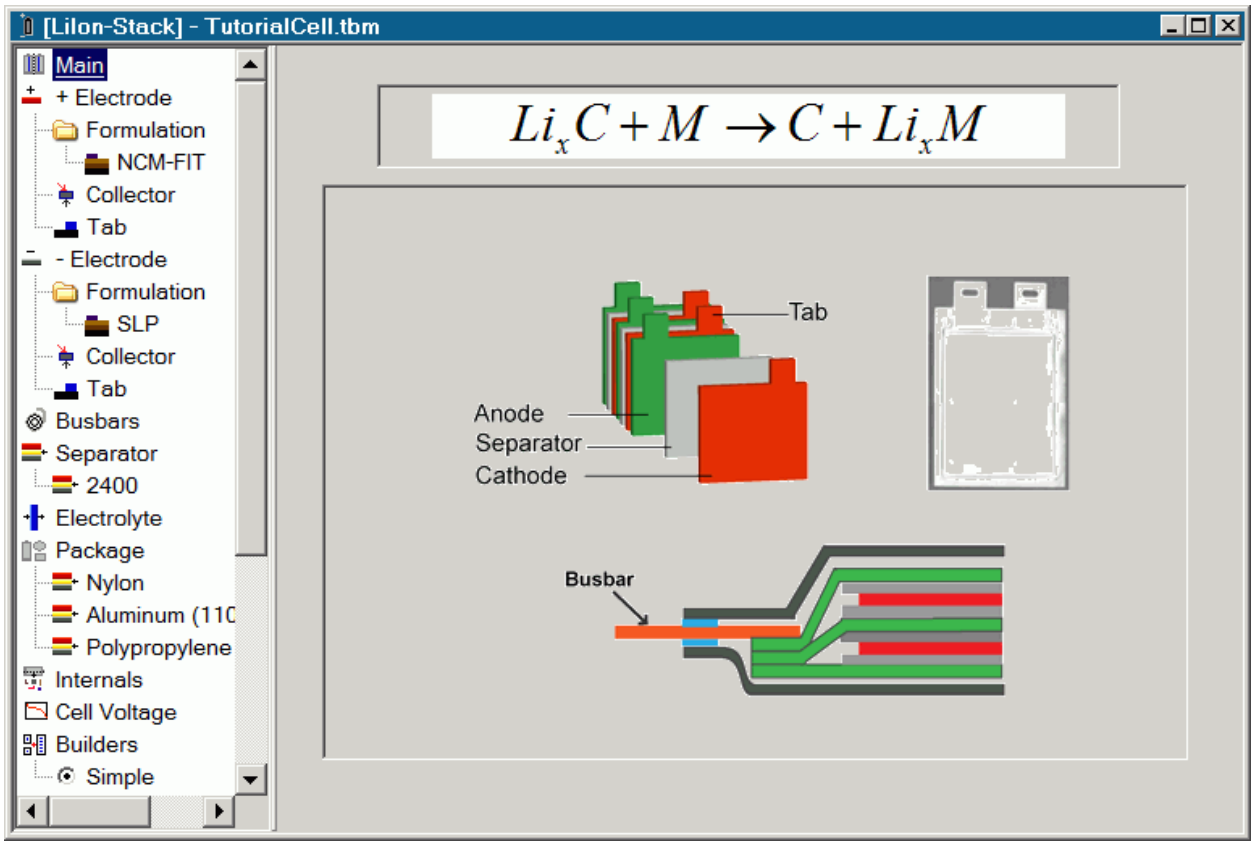
A new node, **LiIonStack**, is added below the **Batteries** node folder, and a corresponding `.tbm` file is created in the workspace directory.

The cell can be renamed immediately after it is created.

7. Rename the battery cell `TutorialCell`.

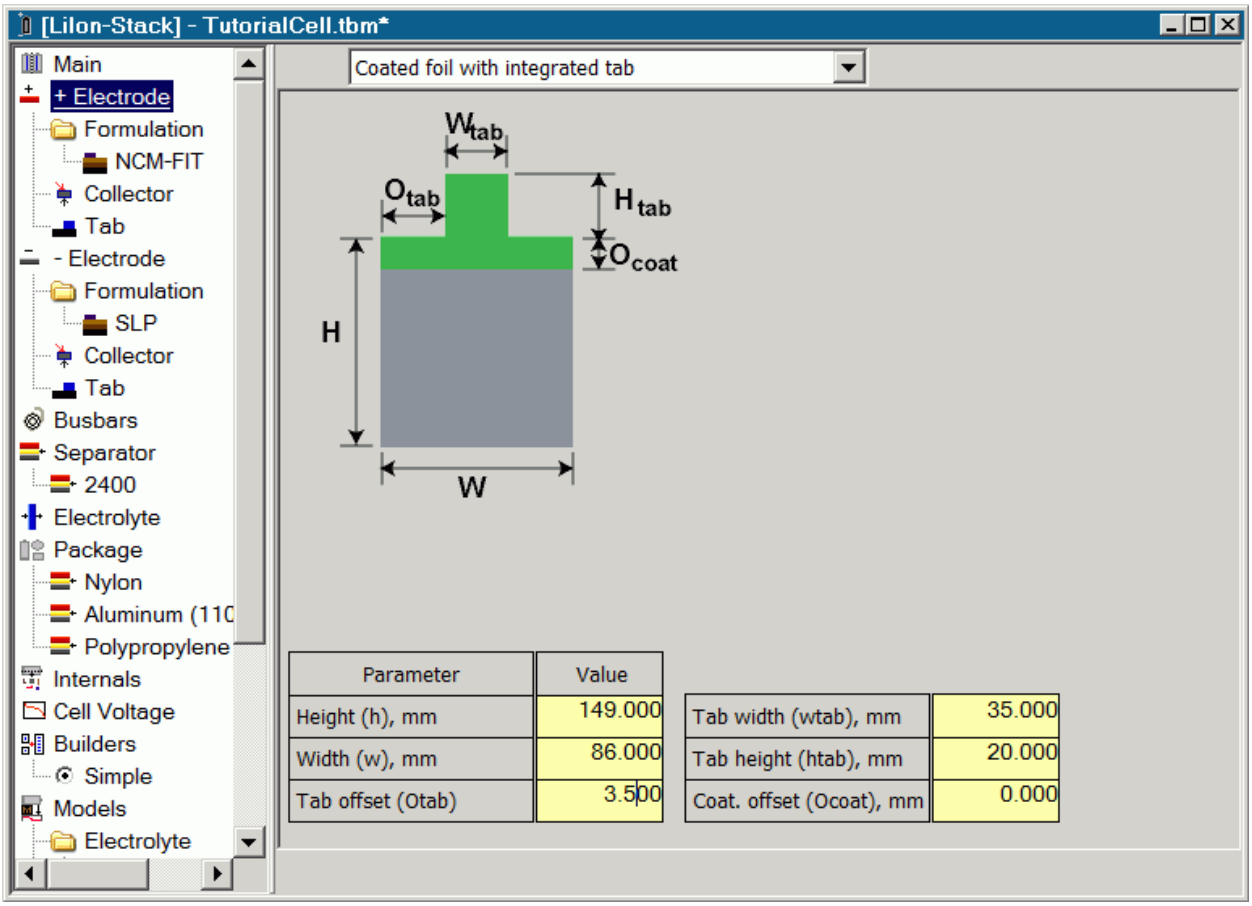


8. To see a diagram of the battery cell, select **Main** in the *TutorialCell.tbm* feature tree.

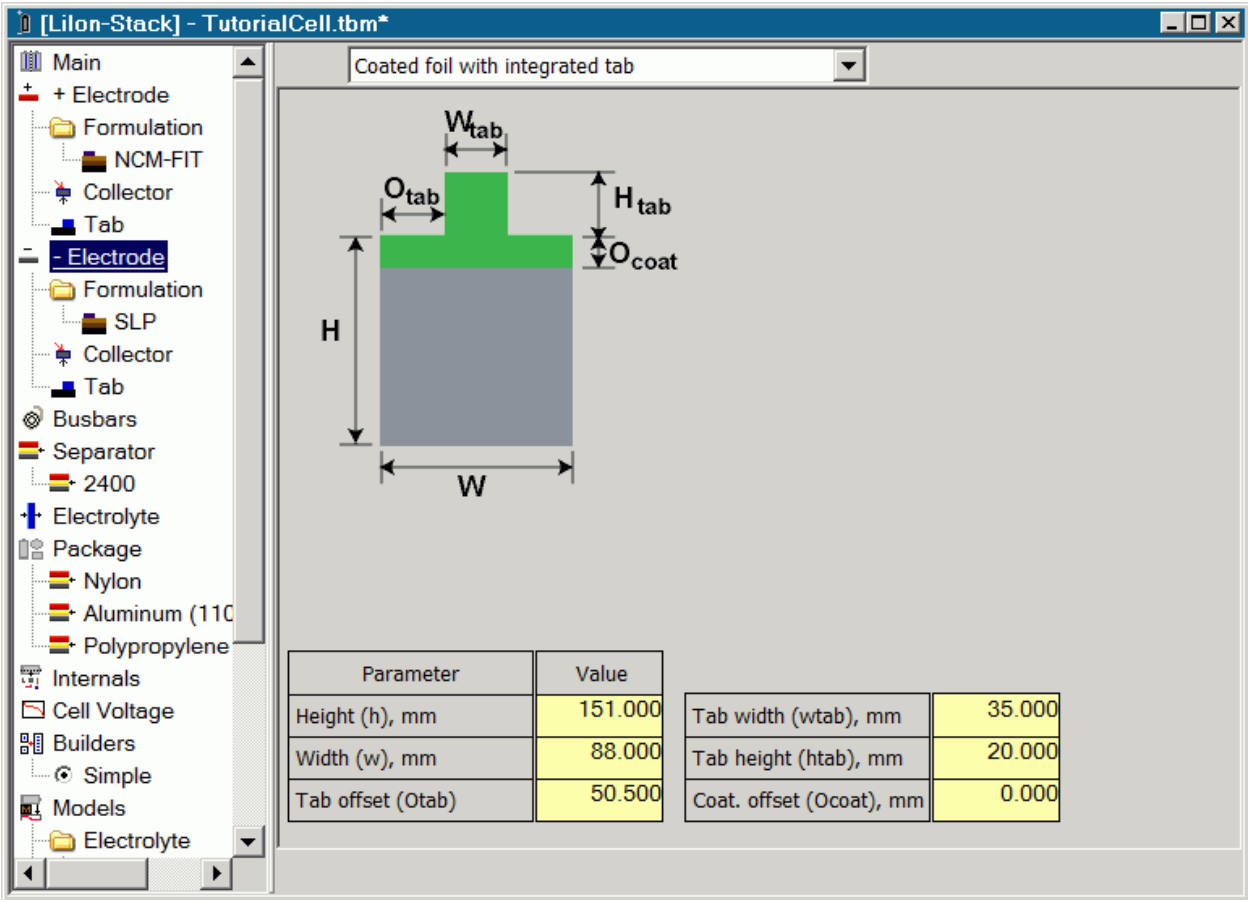


The position of the tabs on each electrode is specified using an offset value. To reposition the tabs, change the offset values.

9. Select the **+ Electrode** node in the *TutorialCell.tbm* window.
10. Set *Tab offset (Otab)* to `3.500`.

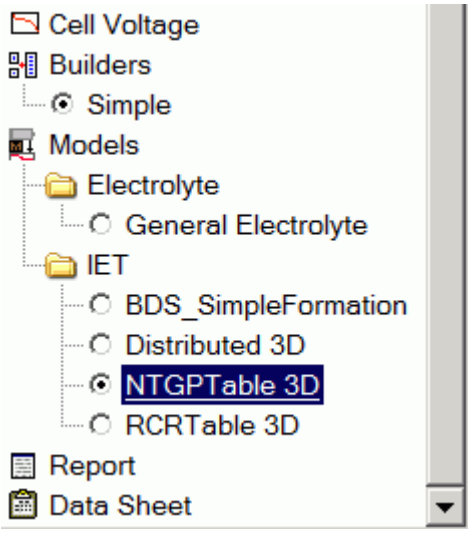


11. Select the **- Electrode** node in the *TutorialCell.tbm* window.
12. Set *Tab offset (Otab)* to `50.500`.



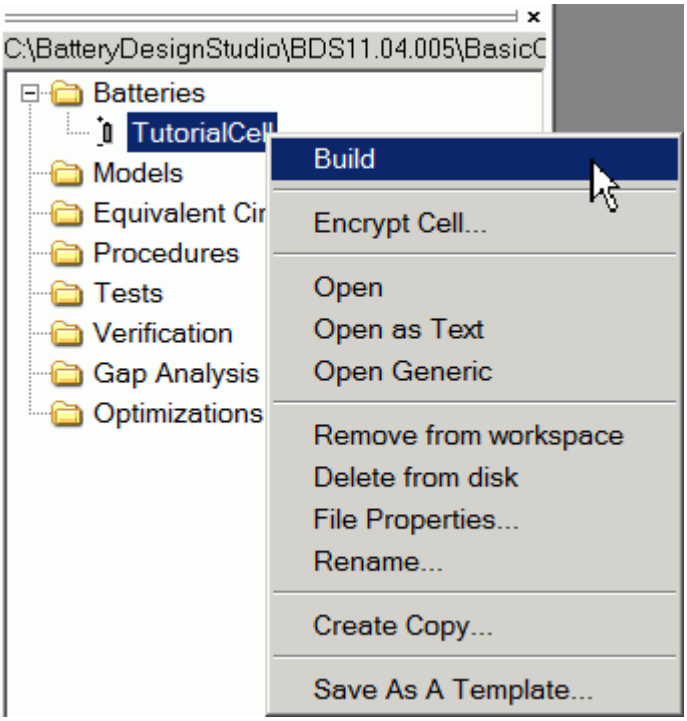
Select a battery model. This model defines how the battery behaves in response to different electrical loads. In this tutorial, the NTGPTable 3D model is used.

13. Scroll down to the **Models** node in the *TutorialCell.tbm* window, and select the NTGPTable 3D model from the **IET** folder.



The addition of a * to the *TutorialCell.tbm* window title indicates that you have changed the cell definition. Once the asterisk appears, **Build** the cell. This process rewrites the `.tbm` file in the workspace directory and prompts BDS to check the cell definition for any invalid changes.

14. Right-click the **Batteries** > **TutorialCell** node in the BDS feature tree and select **Build**.



15. If prompted to save, accept.

    A message box appears informing you that the build was successful.

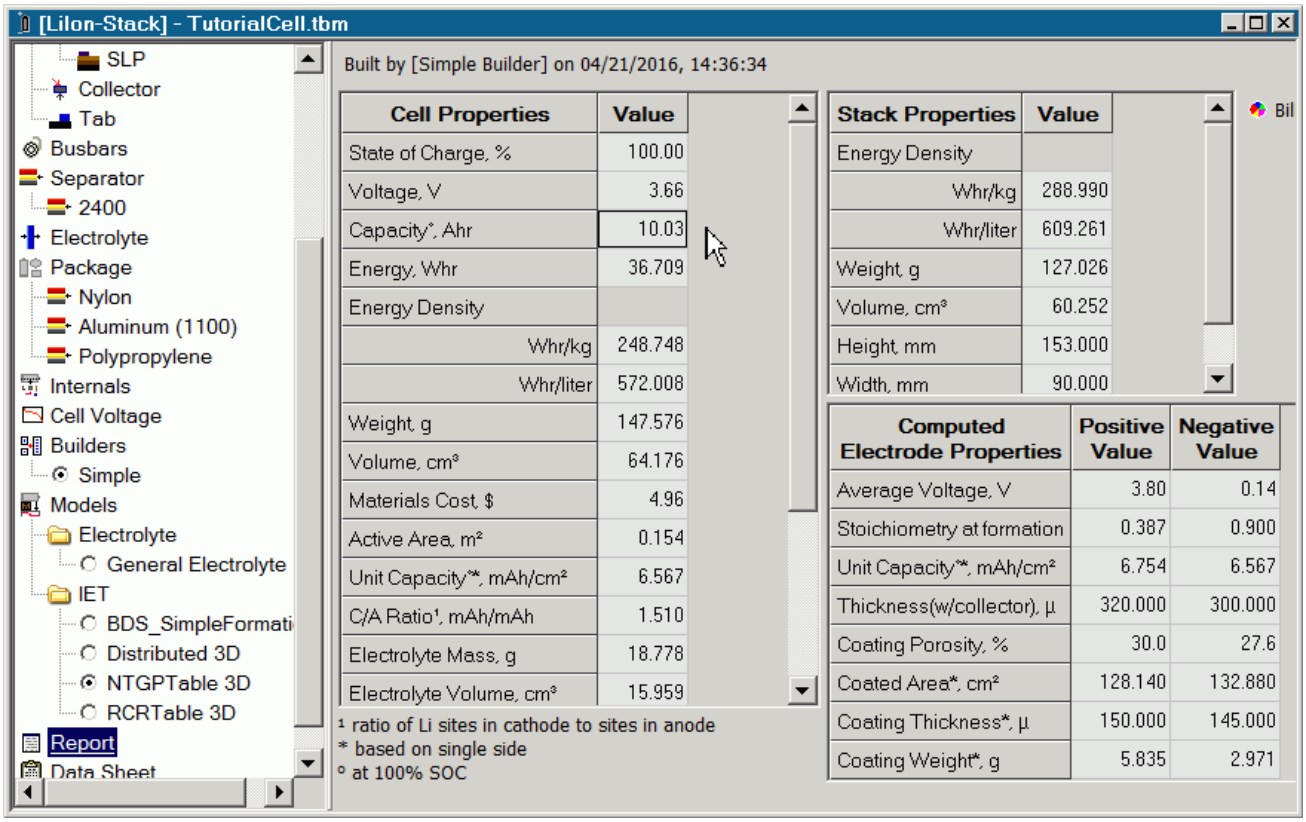
16. Click **OK** to close the message box.

The cell definition is complete.

Creating a Program File

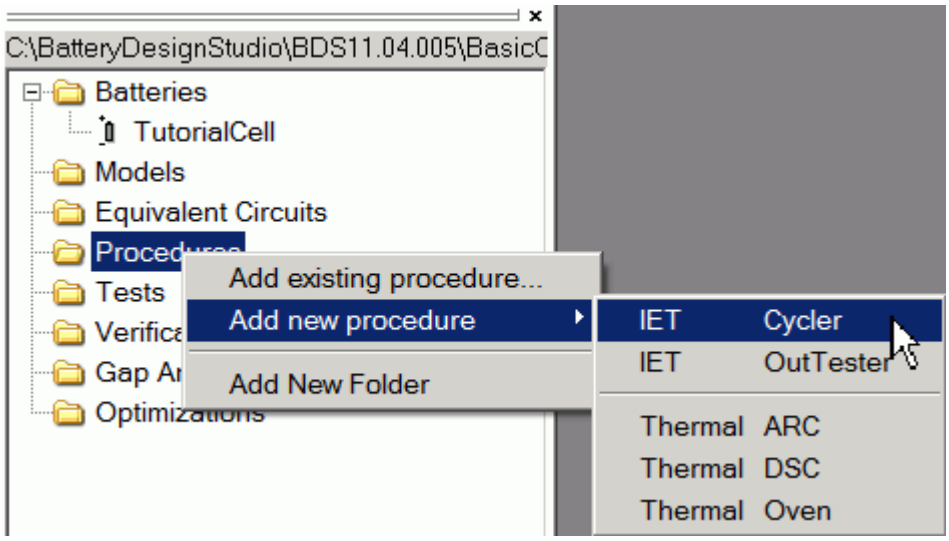
Create a program file in BDS.

The next step is to define a procedure for cycling the battery. In this case, specify a constant current discharge of 10 A, for 1 hour. The capacity of the cell is 10 Ah, so this procedure is equivalent to a 1 C discharge. You can find out the cell capacity by selecting the **Report** node in the *TutorialCell.tbm* window, and reading the value from the **Cell Properties** table.



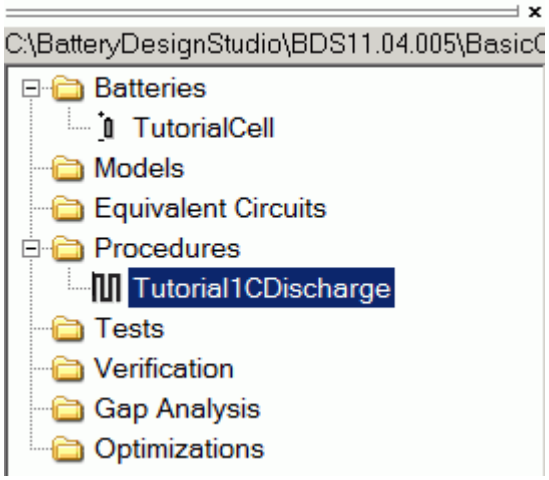
To create a procedure:

1. Right-click the **Procedures** node and select **Add new procedure** > **IET Cycler**.



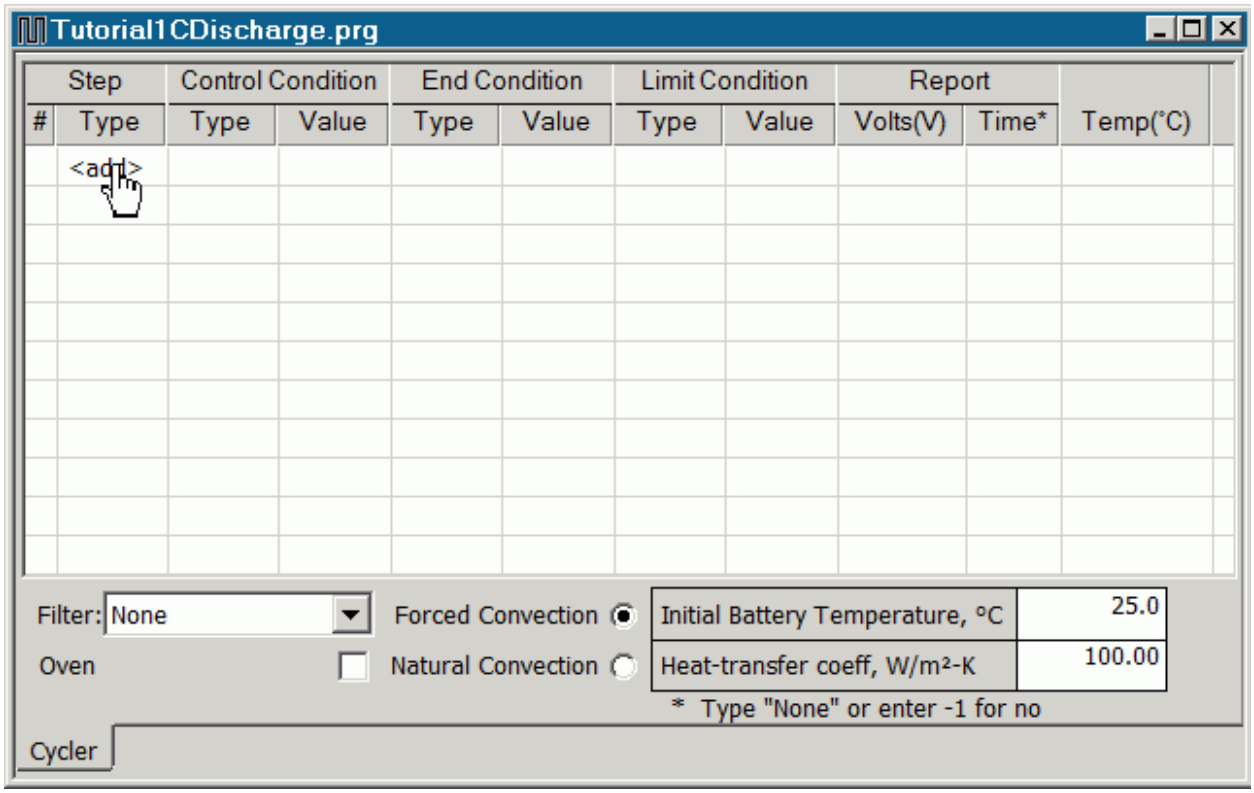
The **IET Cycler** is a simulation model that allows you to define the Current (I), Voltage (E), and Temperature (T). A **Cycler** node is added under the **Procedures** folder, and a corresponding program file (.prg) is created in the workspace directory. A new window that is called *Cycler.prg* appears in the work area.

2. Rename the program file `Tutorial1CDischarge`.

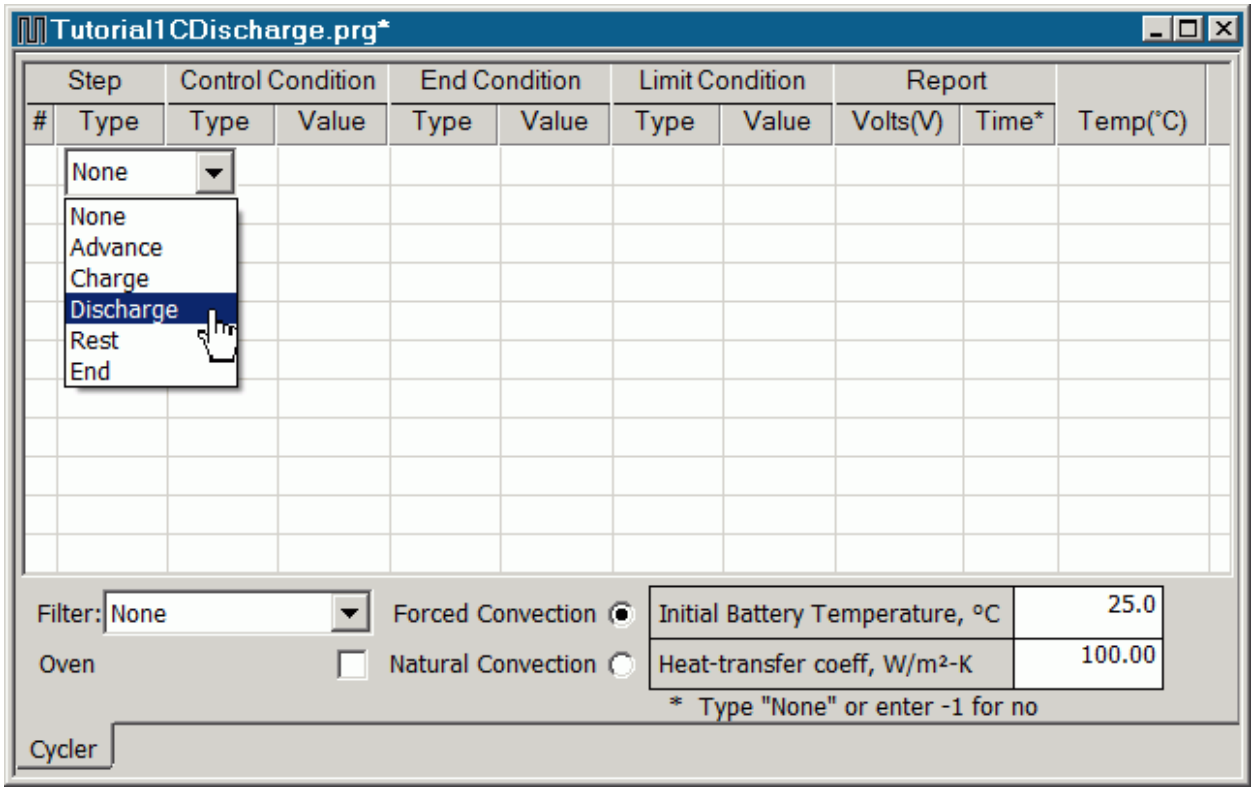


The name of the *Cycler.prg* window changes to *Tutorial1CDischarge.prg*.

3. Click **<add>** under the *Type* column for the **Step** in the *Tutorial1CDischarge.prg* window.

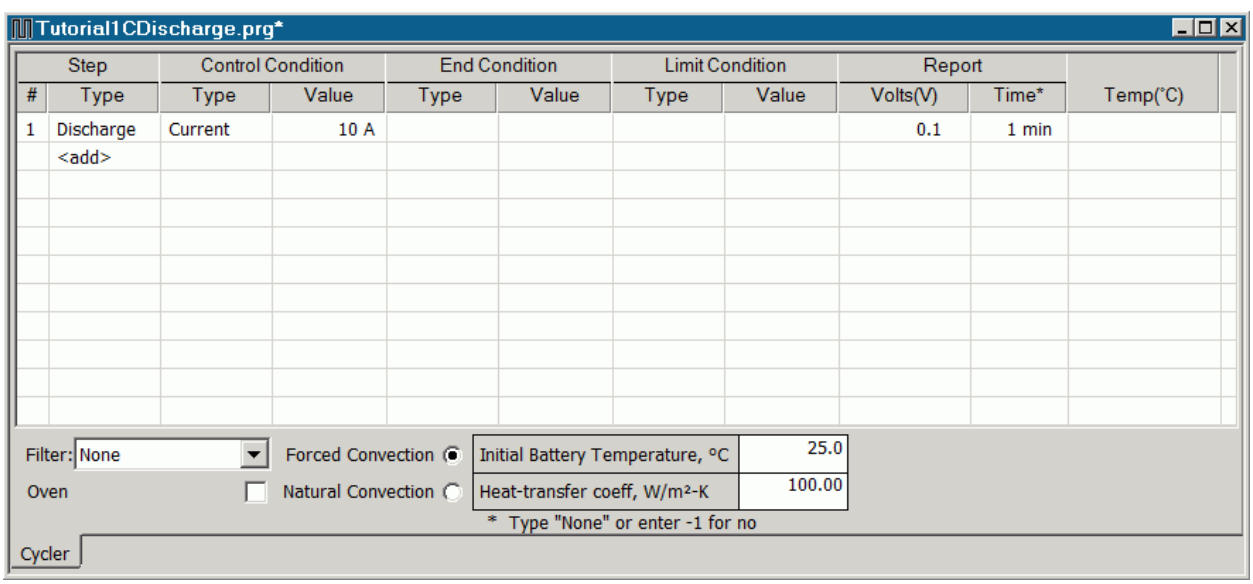


4. Click the cell twice for the drop-down menu to appear and select **Discharge**.



The constant current discharge is specified as a **Control Condition**.

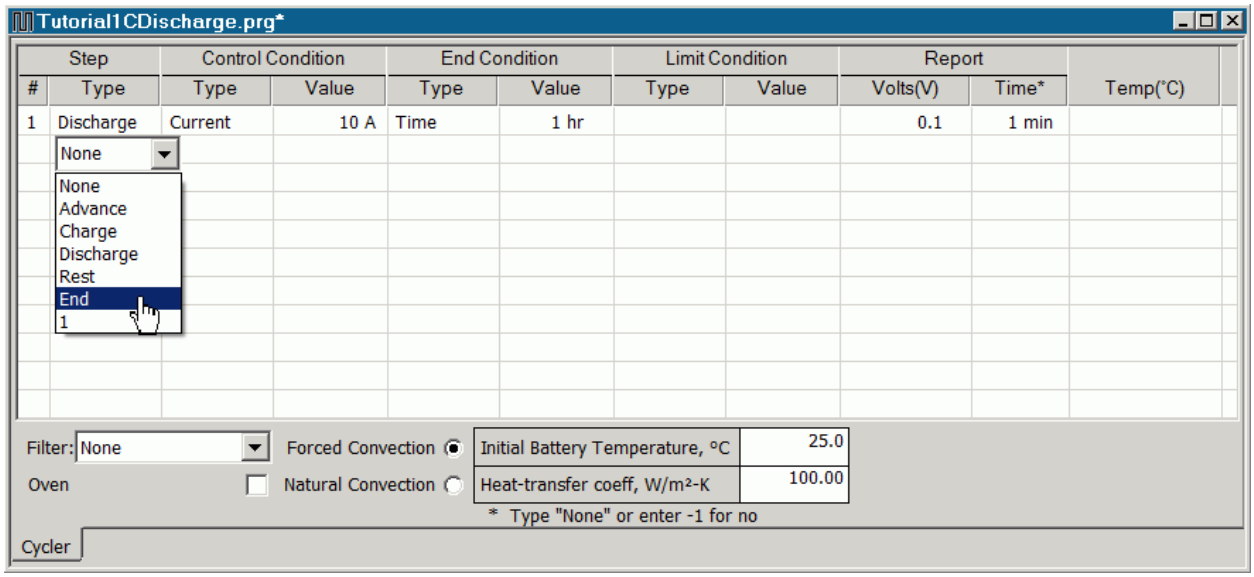
5. Click in the first cell of the *Type* column for the **Control Condition** and select **Current**.
6. Enter `10A` as the *Value*.



The discharge finishes after 1 hour, which is specified as an **End Condition**.

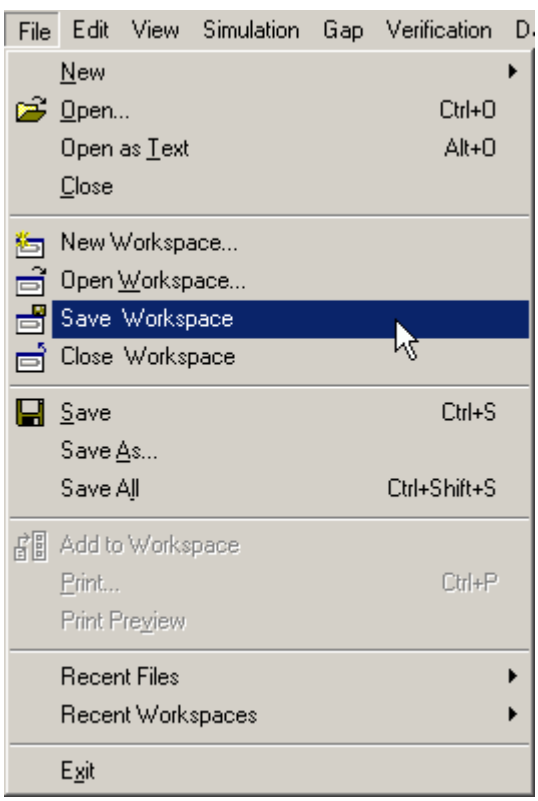
7. Click in the *Type* column for the **End Condition** and select **Time** from the drop-down list.

8. Set the *Value* of **Time** to `3600s`.
9. Click **<add>** under the *Type* column for the **Step** and select **End**.



The definition of the battery cell and program is now complete.

10. Save the workspace by selecting **File** > **Save Workspace** from the menu.



11. Click **Yes**.

Running a STAR-CCM+ Simulation with a Single Cell

Import the files from BDS into STAR-CCM+ and run a simple single cell simulation.

In this section of the tutorial, the files that you created in BDS are imported into STAR-CCM+, and a simple single cell simulation is run.

To create the simulation:

If you skipped the section involving BDS, create a working directory and copy the files that are required for this tutorial.

1. Create a working directory called `BasicCell`.
2. Copy the following files from the `doc/startutorialdata/battery/data` subdirectory of your STAR-CCM+ installation directory into the `BasicCell` directory:
   - `TutorialCell.tbm`
   - `Tutorial1CDischarge.prg`

Create the simulation.

3. Launch STAR-CCM+.
4. Start a simulation.
5. Save the simulation as `batteryTutorial.sim` in the `BasicCell` directory.

Contents:

Selecting the Physics Models
Importing the Battery Cell
Generating a Battery Module
Setting Up the Battery Mesh
Loading the Program Data
Creating a Circuit
Preparing a Scalar Scene
Setting Up a Voltage Monitor
Setting Solver Parameters and Stopping Criteria
Running the Simulation
Visualizing the Results

Selecting the Physics Models

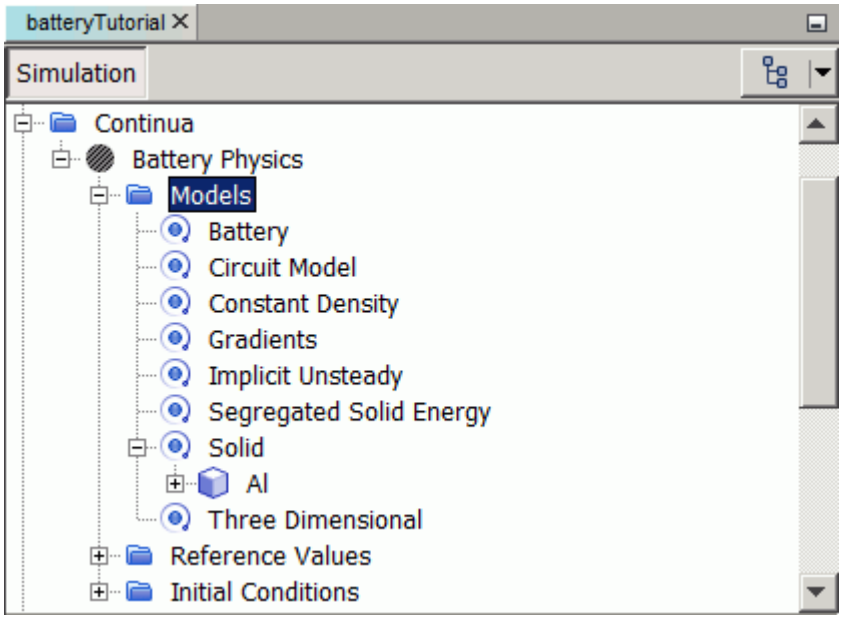
Physics models define the primary variables of the simulation and what mathematical formulation is used to generate the solution.

To select the physics models:

1. Create a physics continuum and rename the **Continua** > **Physics 1** node to **Battery Physics**.
2. For the physics continuum, **Continua** > **Battery Physics**, select the following models in order:

Group Box	Model
Space	**Three-Dimensional**
Time	**Implicit Unsteady**
Material	**Solid**
Optional Models	**Battery** **Circuit Model** (Selected automatically)
Energy	**Segregated Solid Energy**
Equation of State	**Constant Density**

3. Click **Close**.
4. To review the models, open the **Battery Physics** > **Models** node.



The **Batteries** node appears in the STAR-CCM+ feature tree. The battery pre-processing is carried out primarily under this node.

5. Save the simulation.

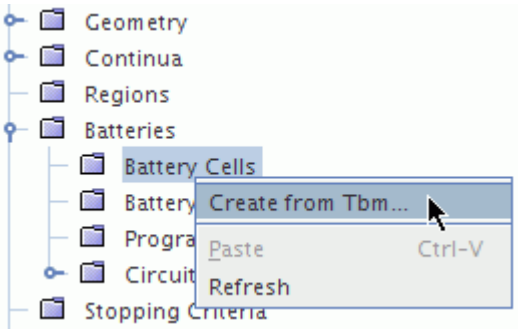
Importing the Battery Cell

The battery file created in BDS is imported into STAR-CCM+ as a battery cell.

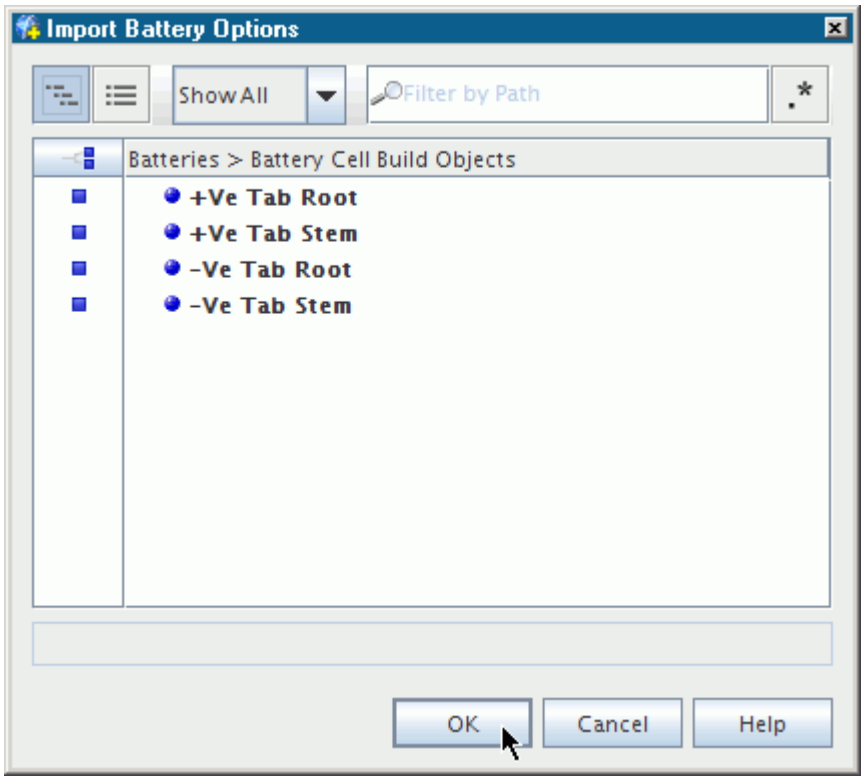
The battery (.tbm) file that you created in BDS is imported into STAR-CCM+ as a battery cell.

To import the battery cell:

1. Expand the **Batteries** node.
2. Right-click **Batteries** > **Battery Cells** and select **Create from Tbm...**.



3. In the *Import Battery Data From File* dialog, navigate to the BDS workspace directory, `BasicCell`, and select the file `TutorialCell.tbm`.
4. Click **Open**.
5. In the *Import Battery Options* dialog, ensure that all objects are selected, then click **OK**.



During the import process, STAR-CCM+ builds an idealized model of the cell using the 3D-CAD module. The model is displayed in the *3D-CAD View 1* scene in the *Graphics* window.

A battery module comprises one or more battery cells. The cells can be arranged in series and parallel. You specify the arrangement and number of cells. As a module typically contains multiple instances of the same cell, STAR-CCM+ automatically copies the imported cell for more instances.

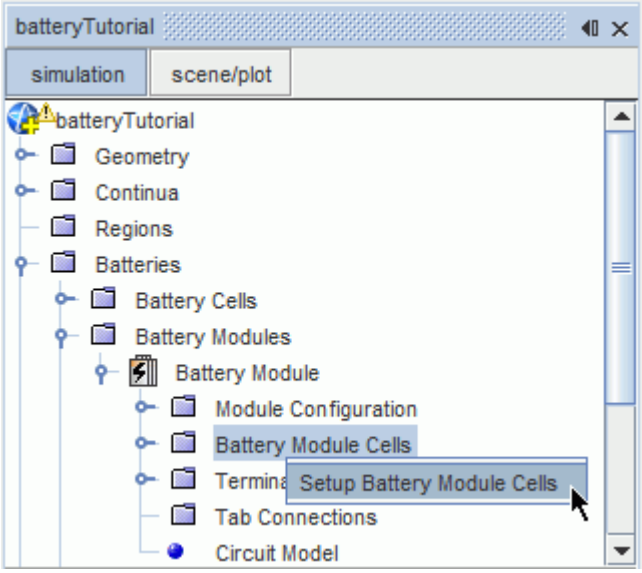
In this section of the tutorial, the module contains a single cell.

6. Right-click **Batteries** > **Battery Modules** and select **New**.

   A new **Battery Module** node is added to the tree. In the *Battery Module - Properties* window, the *N Series* and *N Parallel* properties define the number of cells in the module. By default these values are both set to `1`.

The next step is to set up the module cells.

7. Right-click **Battery Module** > **Battery Module Cells** and select **Setup Battery Module Cells**.

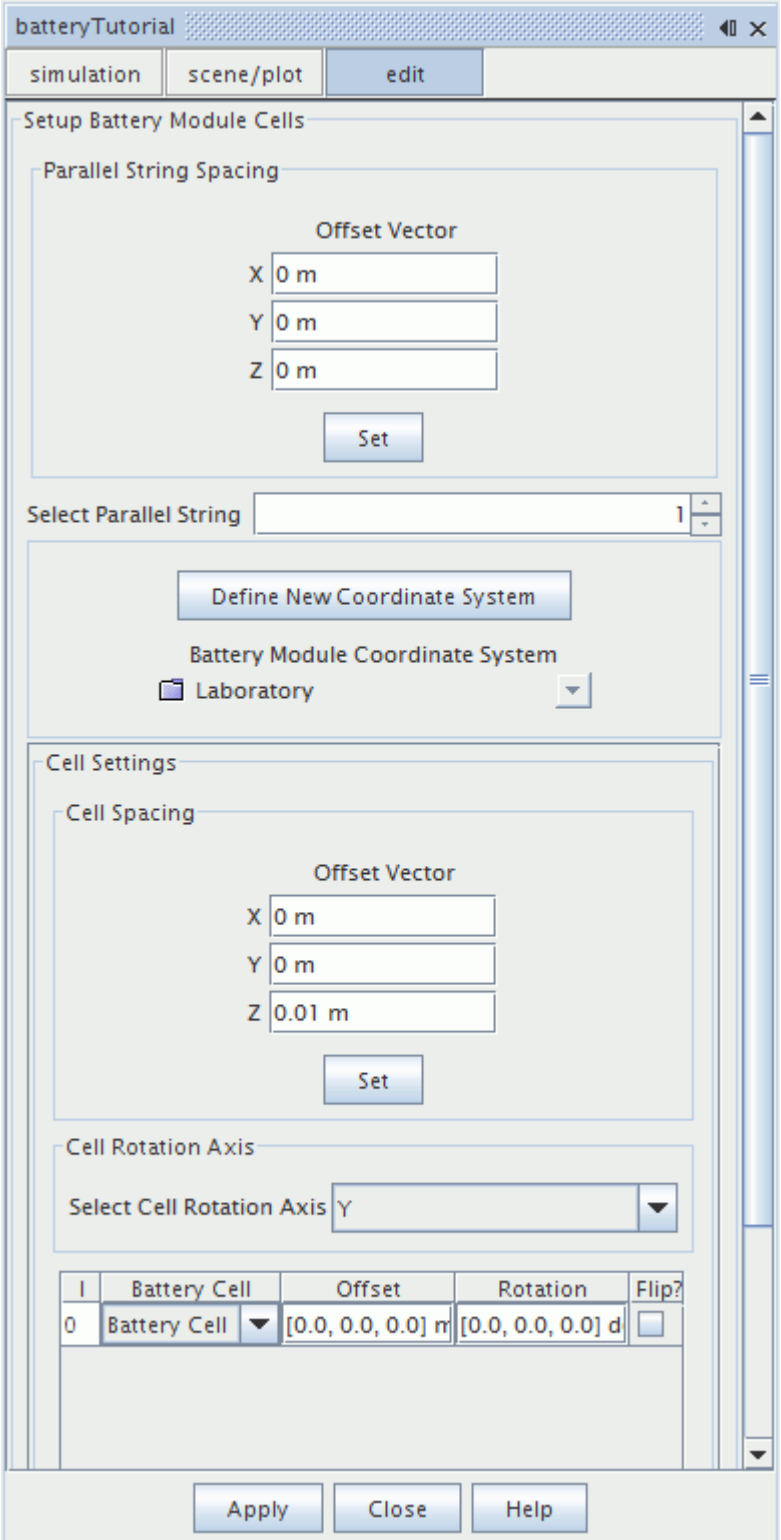


The panel that appears allows you to:

- Define the offset of battery cells in relation to each other.
- Flip or rotate cells.
- Assign different cells to each position in the battery module.

In this case, the default settings are suitable.

8. Click **Apply** to assign the imported cell to the battery module.



9. Click **Close**.

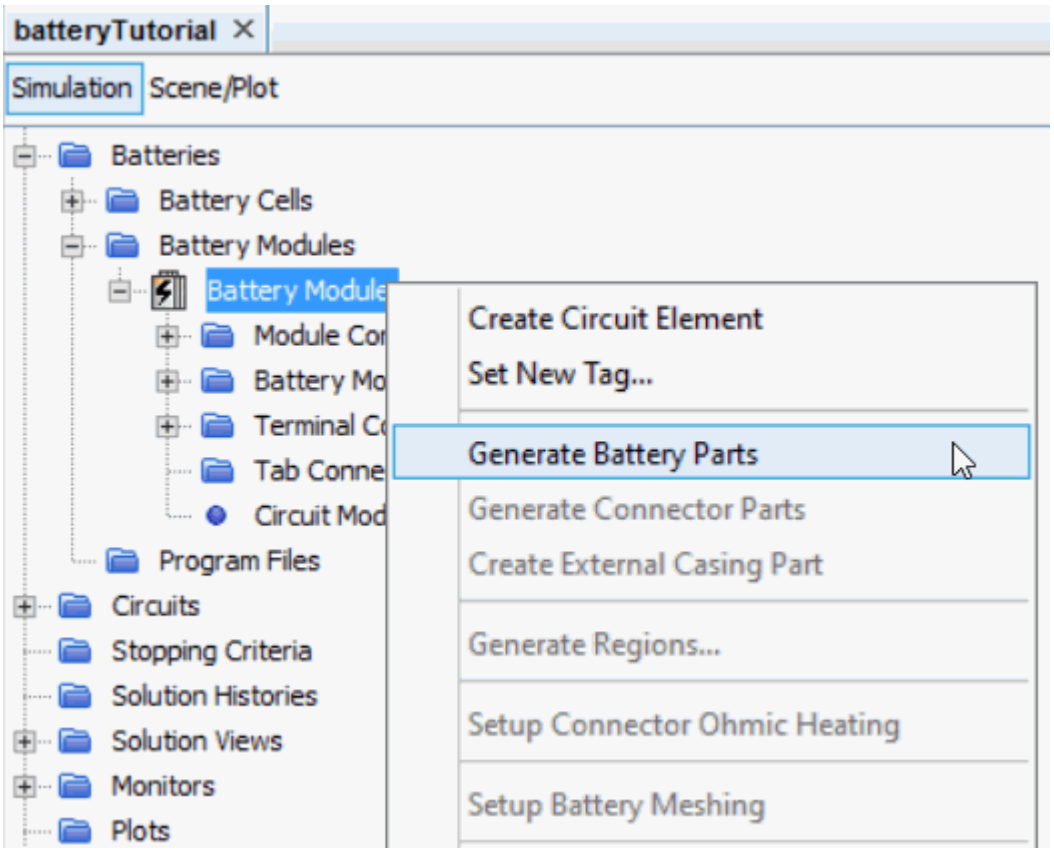
Generating a Battery Module

Generate the battery module in STAR-CCM+.

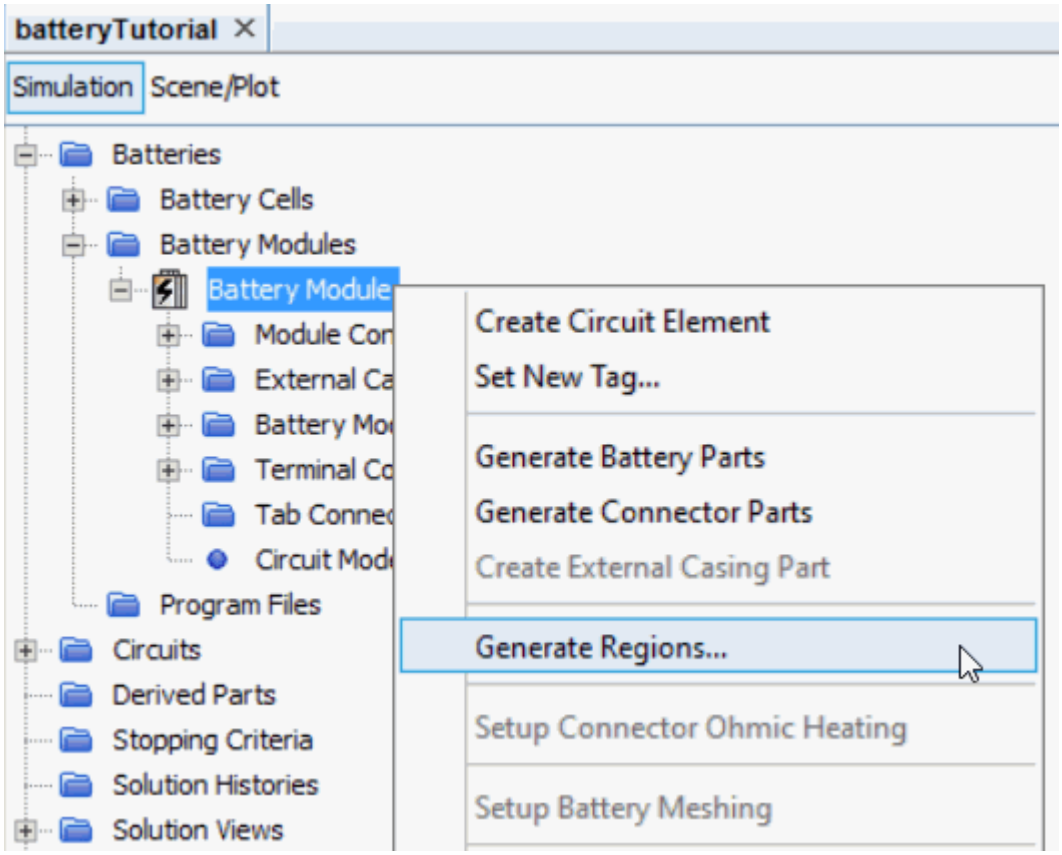
You are now ready to generate the battery module. This process involves generating parts, and then generating regions. For modules that contain several cells, this process also duplicates the battery cells according to the battery module setup.

To generate the battery module:

1. Right-click the **Battery Modules** > **Battery Module** node and select **Generate Battery Parts.**.

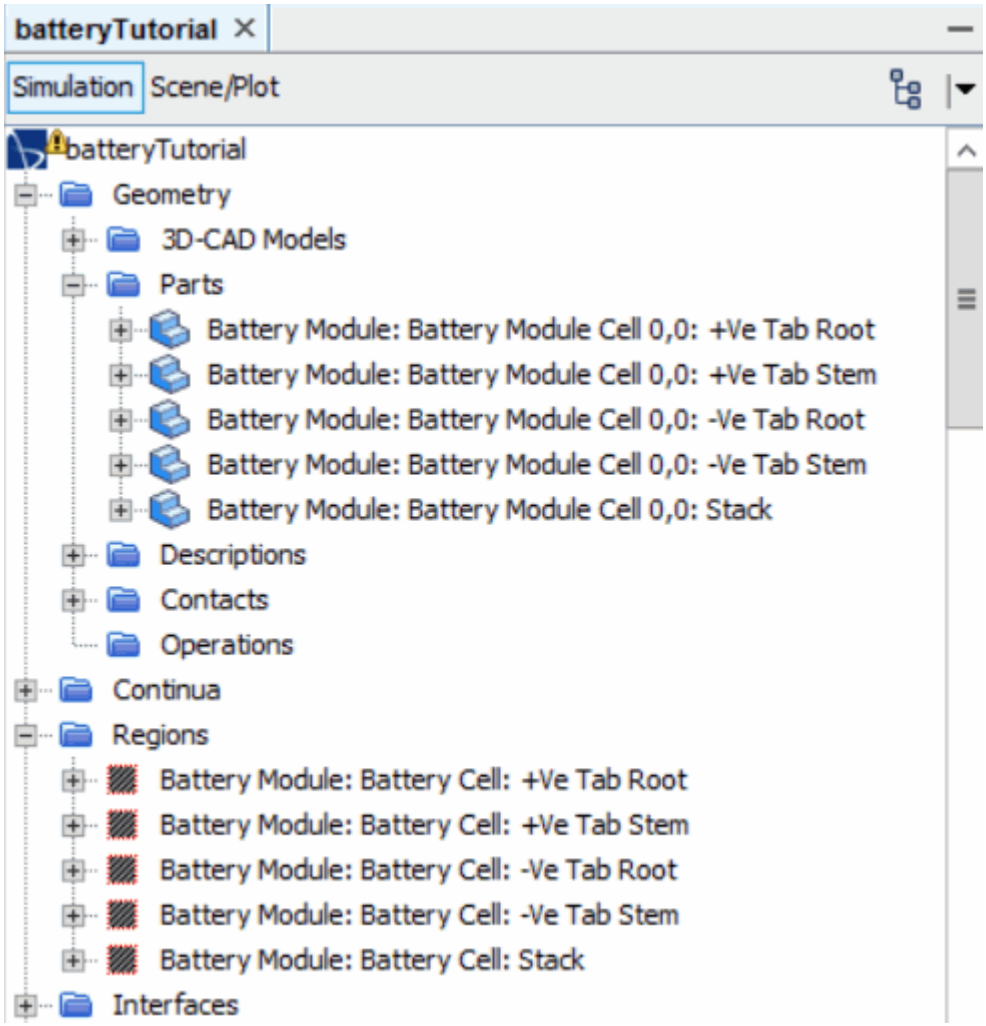


2. Right-click the **Battery Modules** > **Battery Module** node and select **Generate Regions...**.



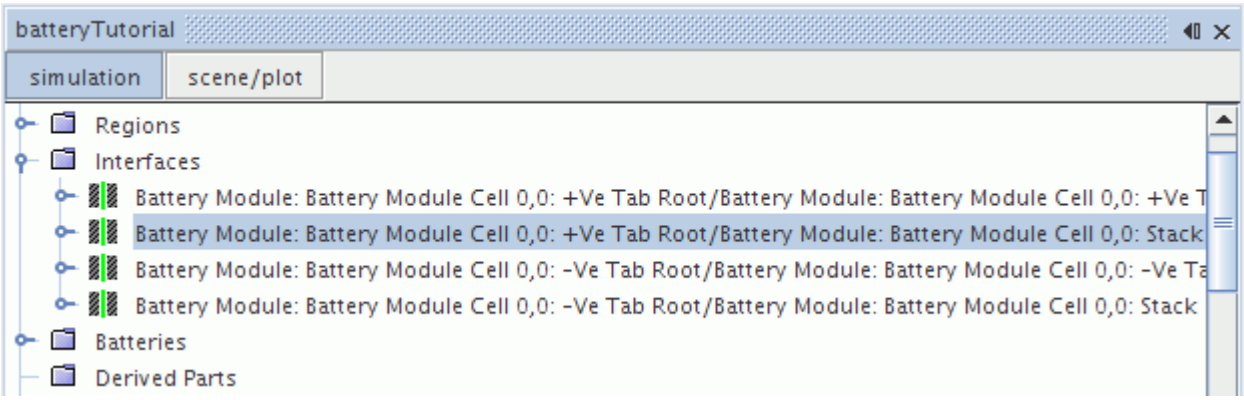
An *Additional (non-Battery) Parts to Generate Regions* dialog appears. This dialog is empty because you do not have any non-battery parts in this simulation.

3. Click **OK**.
4. Expand the **Geometry** > **Parts** node, and the **Regions** node, to see the parts and regions that were created during the process.

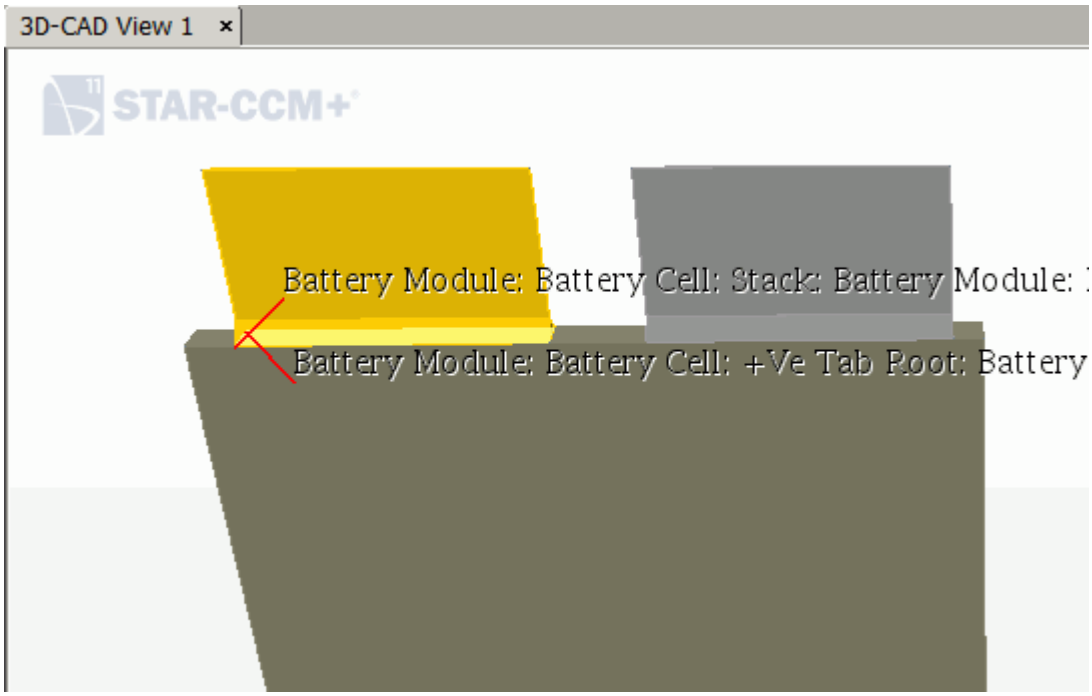


Interfaces between the stack, tab roots, and tab stems have automatically been created during the generate regions action.

5. Select the **Interfaces** > **Battery Module: Battery Module Cell 0,0: +Ve Tab Root/Battery Module: Battery Module Cell 0,0: Stack** node.



The selected interface is highlighted in *3D-CAD View 1*.



6. Save the simulation.

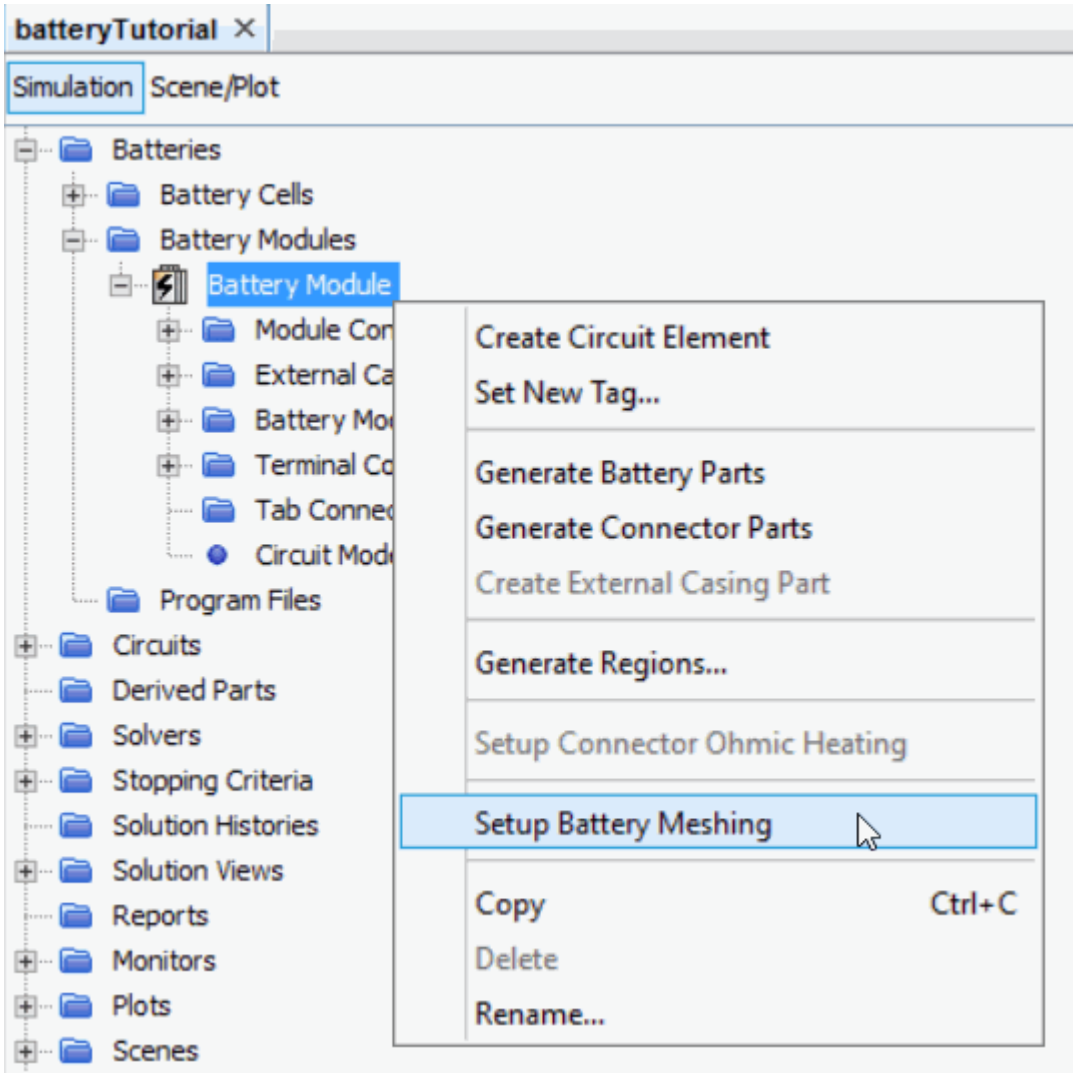
Setting Up the Battery Mesh

Define the battery mesh and set the appropriate properties.

The battery mesher generates a thin mesh on the tab stems, tab roots, and stack for thermal mesh and also generates a separate Cartesian electrical mesh on the stack. The mesh parameters for the tabs and thermal mesh are specified in a mesh continuum, while the number of cells in the electrical mesh are specified on the battery cell. Define the electrical mesh first.

To set up the mesh continuum:

1. Right-click the **Batteries** > **Battery Modules** > **Battery Module** node.
2. Select **Setup Battery Meshing**.



A new **Mesh 1** node appears under the **Continua** node. This new battery mesh continuum already contains the **Surface Remesher** and **Thin Mesher** models selected.

3. Rename the new Mesh 1 node to `Battery Mesh`.

Confirm that the **Battery Mesh** continuum contains the correct meshing models.

4. Edit the **Continua** > **Battery Mesh** node.
5. Ensure that the **Surface Remesher** and **Thin Mesher** models are selected.
6. Click **Close**.

Modify the **Battery Mesh** continuum **Reference Values**.

7. Expand the **Battery Mesh** > **Reference Values** node.
8. Select the **Base Size** node and set the **Value** to `5 mm`.

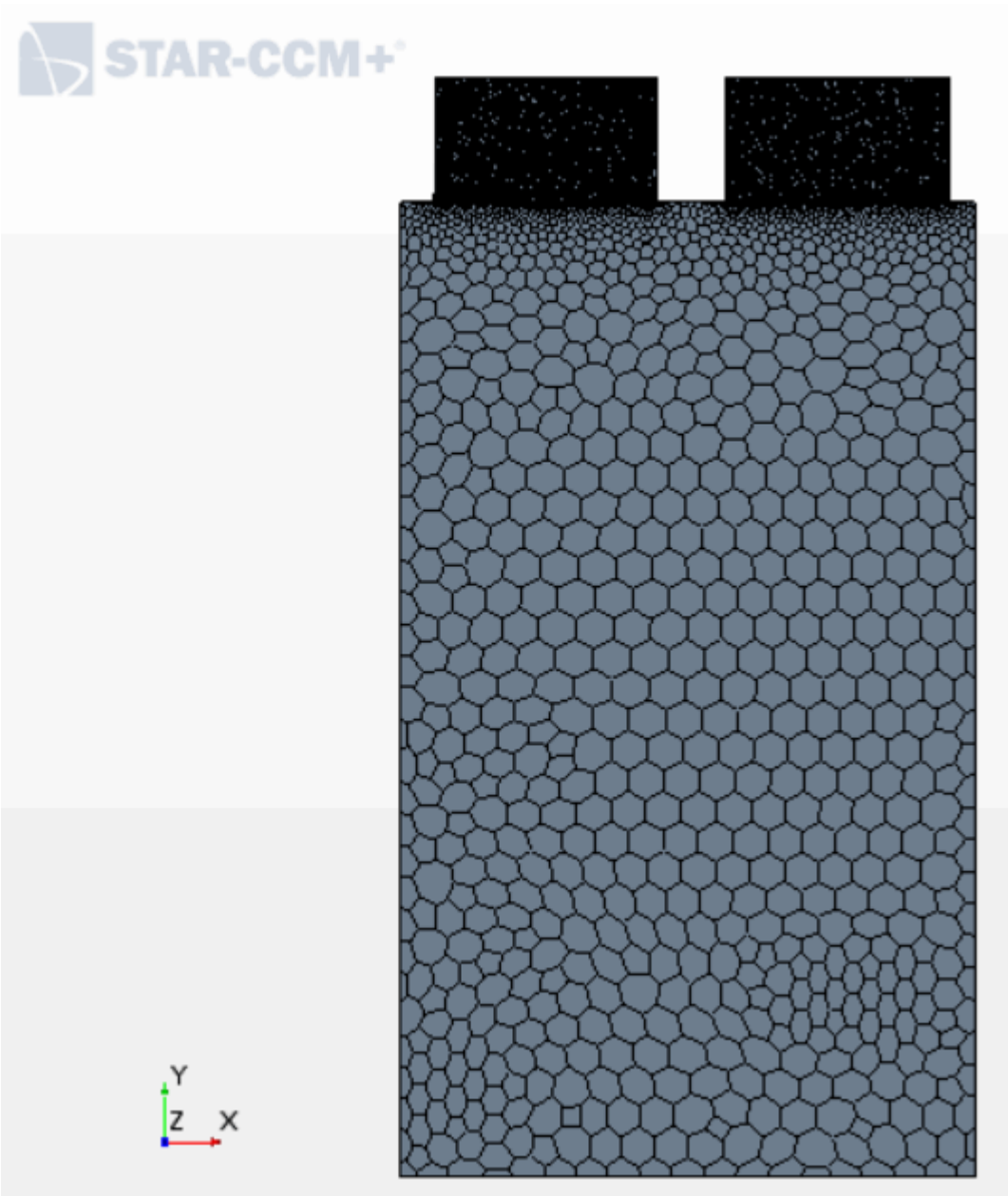
The properties of the **Battery Cell** node include the number of cells in the X, Y, and Z directions for the electrical stack. Initially, these parameters are defined based on the values in the `.tbm` file, but they can be changed to modify the stack electrical mesh resolution. Further information on setting up the mesh is provided in the STAR-CCM+ User Guide. You cannot view the electrical mesh.

To change the number of cells in the electrical mesh:

9. Select the **Batteries** > **Battery Cells** > **Battery Cell** node.
10. In the *Battery Cell Properties* window, set the *Electrical Mesh Dimensions* to `[5,5,1]`.
11. Click (**Generate Volume Mesh**) to build the volume mesh.
12. Create a Mesh Scene.

    Mesh Scene 1 is displayed in the *Graphics* window.

    The volume mesh is shown below.



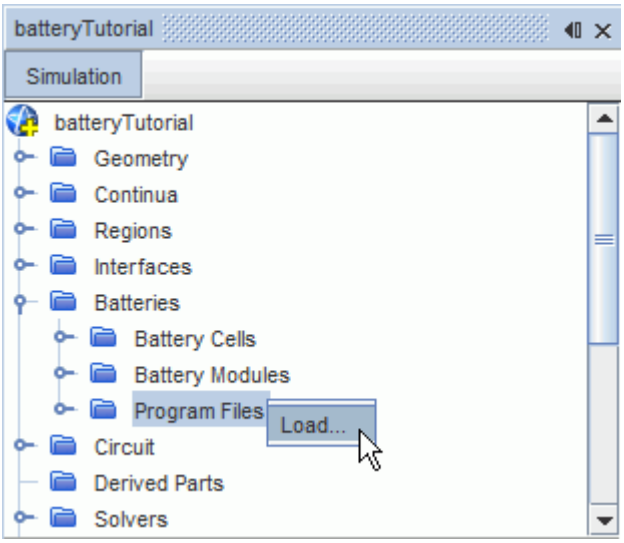
Loading the Program Data

Load the discharge procedure that you created in BDS into STAR-CCM+.

Load the discharge procedure that you created in BDS into STAR-CCM+ using the **Program Files** node in the STAR-CCM+ tree.

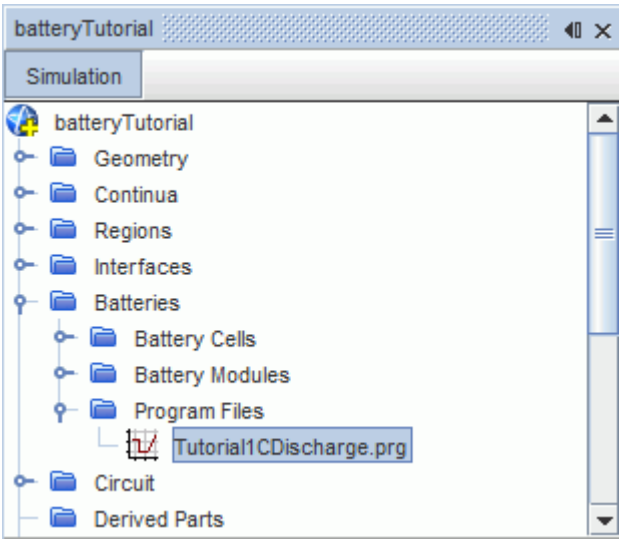
To load the program data:

1. Right-click **Batteries** > **Program Files** and select **Load**.



2. In the *Open* dialog, navigate to the `BasicCell` directory and select the `Tutorial1CDischarge.prg` file.
3. Click **Open**.

   The imported file is displayed under the **Program Files** node.



Creating a Circuit

Create a circuit to simulate the response of the battery.

The program file that you loaded in the previous step represents an electrical load on the battery. To simulate the response of the battery, the electrical load must be connected to the battery in a circuit.

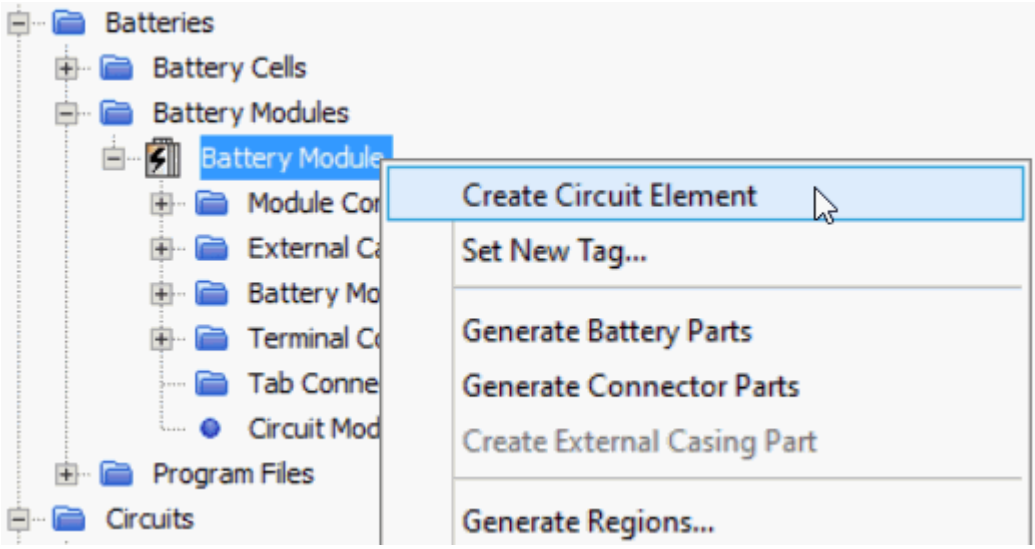
Connect the electrical load using the **Circuits** tool.

1. Expand the **Circuits** > **Circuit 1** node.

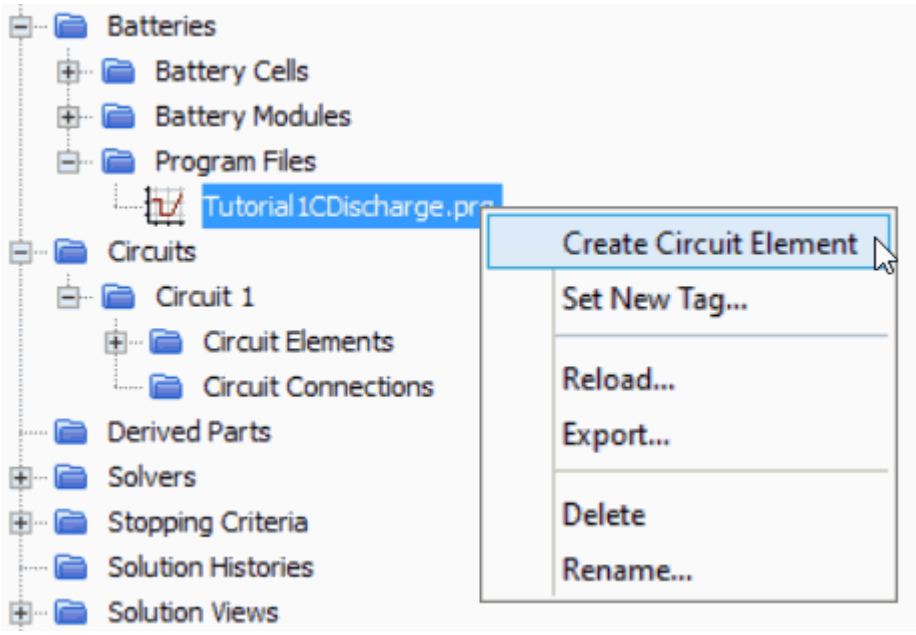
The **Circuits** tool uses Elements and Connections. Each element is either a battery module or a program, and a connection joins elements together in the circuit.

First, you define the battery module and program as elements of the circuit.

2. Right-click the **Batteries** > **Battery Modules** > **Battery Module** node and select **Create Circuit Element**.

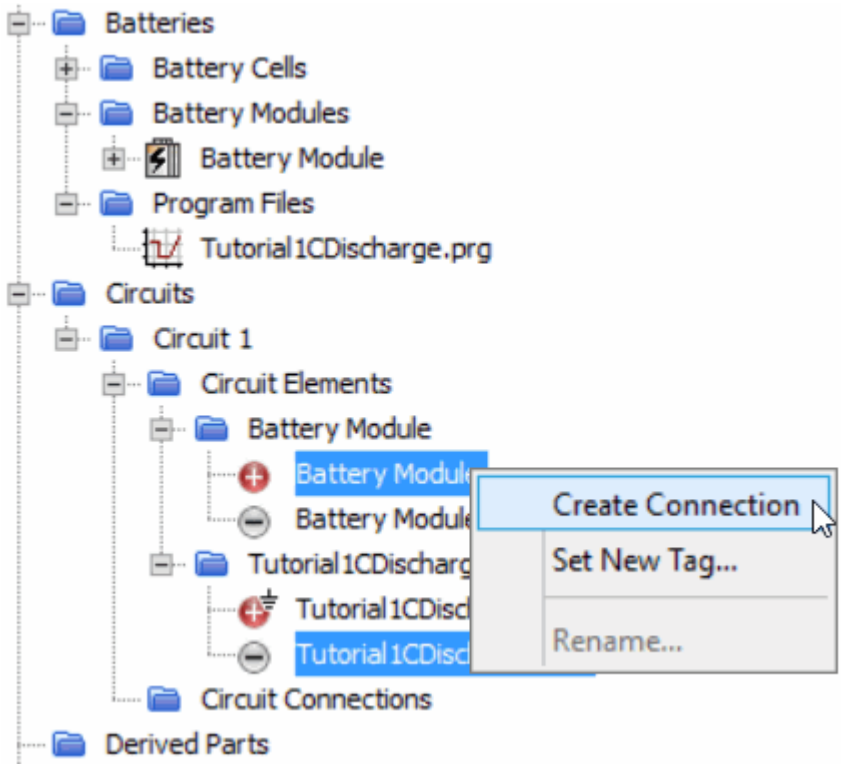


3. Right-click the **Program Files** > **Tutorial1CDischarge.prg** node and select **Create Circuit Element**.



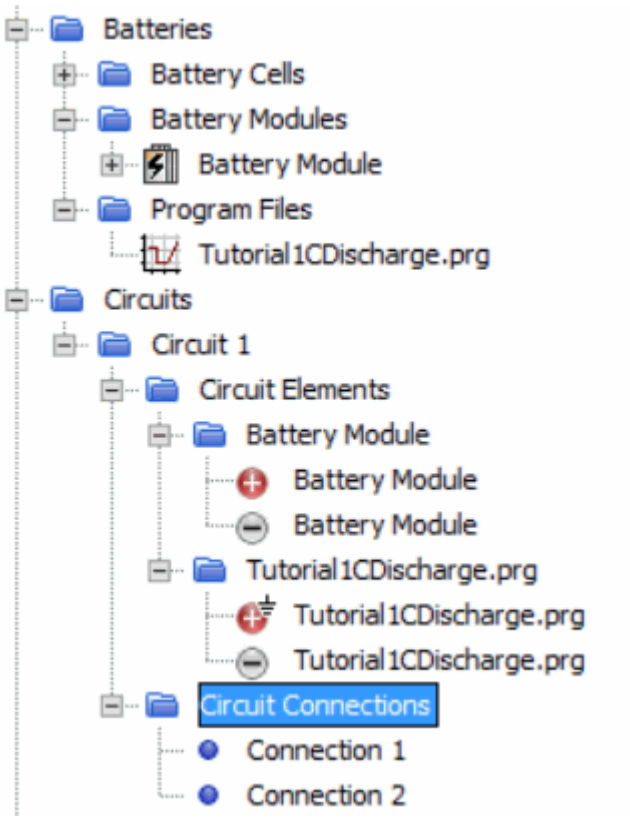
Next, the positive and negative terminals of the elements are connected.

4. Expand the **Circuits** > **Circuit 1** > **Circuit Elements** node.
5. Use `[Ctrl-Click]` to select the **Battery Module** > **(+) Battery Module** node and the **Tutorial1CDischarge.prg** > **(-) Tutorial1CDischarge.prg** node.
6. Right-click one of the nodes and select **Create Connection**.



7. Repeat for the remaining terminals: **(-) Battery Module** and **(+) Tutorial1CDischarge.prg**.

   Each connection is shown under the **Circuit Connections** node.



Preparing a Scalar Scene

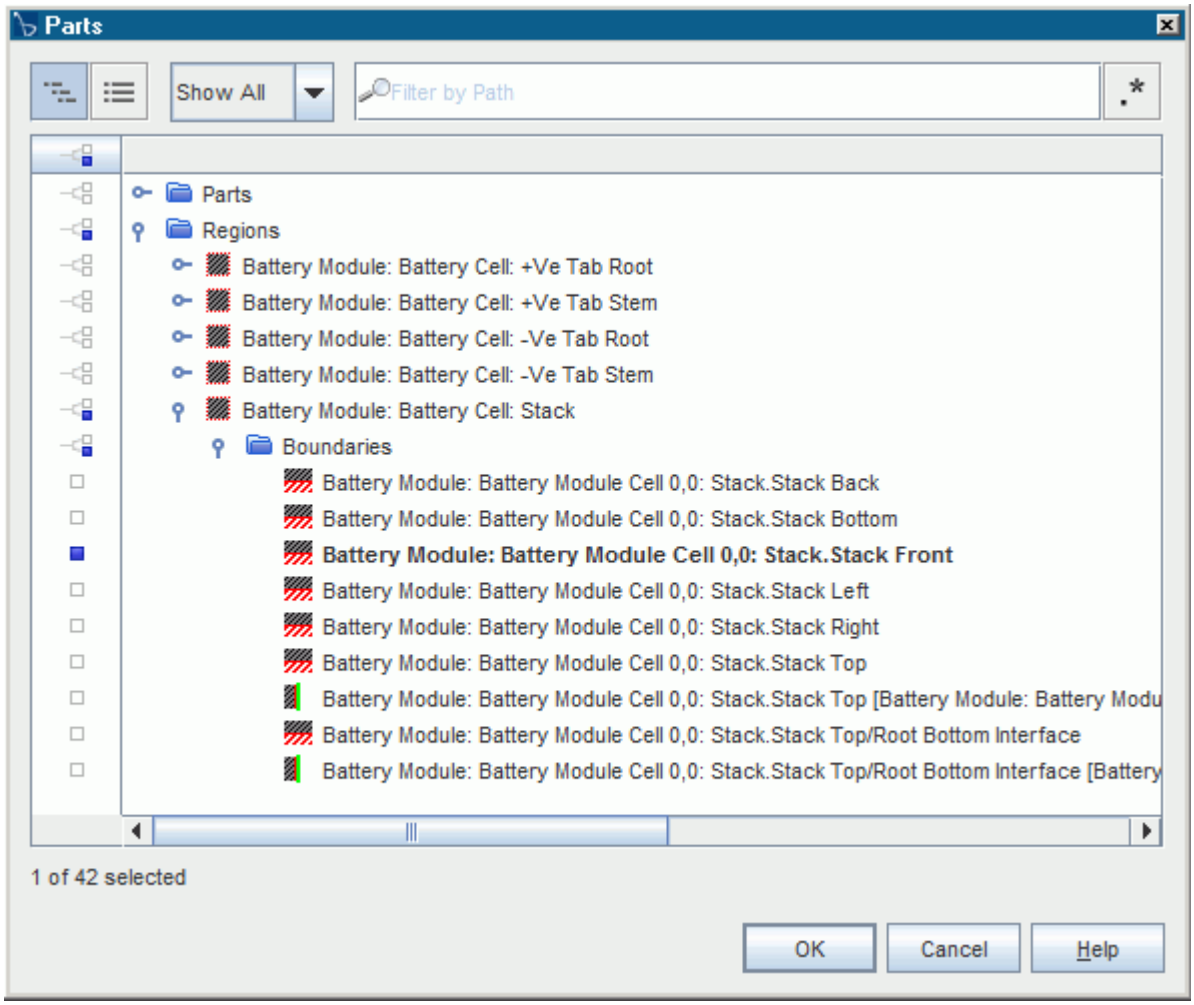
Create a scalar scene to display the results of the simulation.

A scalar scene is created to display the volumetric heat of the battery during the simulation.

1. Create a scalar scene.
2. Click **Scene/Plot**.
3. Right-click **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Parts** and select **Edit**.

   The *Parts* dialog appears.

4. Click (**Column Selection**) until all parts are deselected.
5. Expand the **Regions** node and the **Battery Module: Battery Cell: Stack** node.
6. Select **Battery Module: Battery Cell 0,0: Stack. Stack Front**.



7. Click **OK**.
8. Select the **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Scalar Field** node.
9. In the *Scalar Field - Properties* window, set *Function* to **Battery** > **Battery Volumetric Heat**.
10. Click the **Simulation** tab to return to the STAR-CCM+ simulation object tree.

Setting Up a Voltage Monitor

Define a voltage monitor to show the voltage of the battery over the course of the simulation.

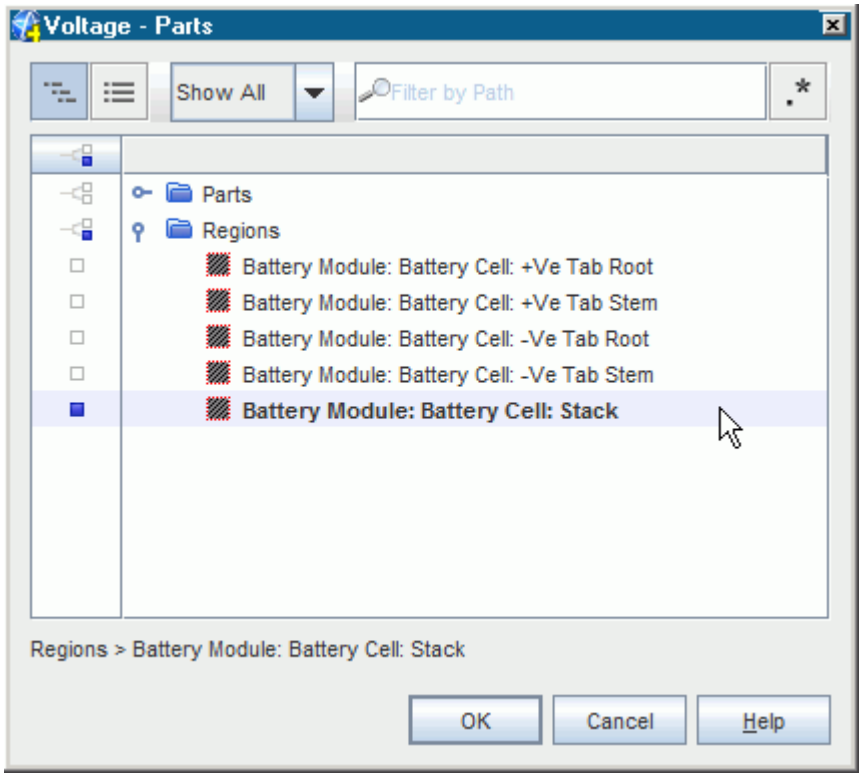
To set up a voltage monitor:

1. Right-click **Reports**.
2. Select **New Report** > **Volume Average**.

   A new **Volume Average 1** node appears under the **Reports** node.
3. Rename the **Reports** > **Volume Average 1** node to `Voltage`.
4. Select the **Reports** > **Voltage** node.
5. In the Voltage *Properties* window, set **Field Function** to **Battery** > **Battery Volts**.

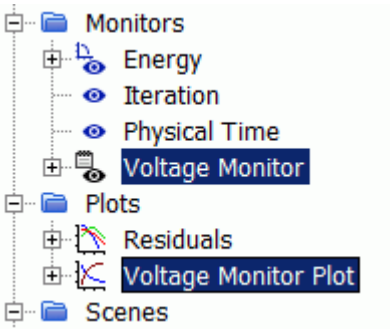
6. Click ... (**Custom Editor**) to the right of the *Parts* property.

   The *Voltage - Parts* dialog appears.
7. Expand the **Regions** node.
8. Click **Battery Module: Battery Cell: Stack**.



9. Click **OK** to close the dialog.
10. Right-click the **Reports** > **Voltage** node and select **Create Monitor and Plot from Report**.

    A new **Voltage Monitor** node appears under the **Monitors** node and a new **Voltage Monitor Plot** node appears under the **Plots** node.



The **Voltage Monitor** node lets you monitor the voltage of the battery with time.

11. Select the **Monitors** > **Voltage Monitor** node.

12. In the *Voltage Monitor - Properties* window, make sure that *Trigger* is set to **Time Step**.
13. To open the *Voltage Monitor Plot*, double-click the **Plots** > **Voltage Monitor Plot** node.

Setting Solver Parameters and Stopping Criteria

Set the appropriate solver parameters and stopping criteria for the simulation.

To ensure adequate convergence for the energy equation, you increase the number of inner iterations per time-step.

To set the solver parameters and stopping criteria:

1. Edit the **Solvers** node and set the following properties:

Node	Property	Setting
Implicit Unsteady	*Time-Step*	`1.0 s`
Segregated Energy	*Solid Under-Relaxation Factor*	`0.999`

2. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Physical Time	*Enabled*	Activated
Maximum Physical Time	*Maximum Physical Time*	`90 s`
Maximum Inner Iterations	*Maximum Inner Iterations*	`15`

3. Save the simulation.

Running the Simulation

Preparation of the simulation is now complete, and the simulation can be run.

To run the simulation:

1. Click (**Run**) in the *Solution* toolbar.

   The solution progress is displayed in the *Output* window. The *Residuals* display is created automatically in the *Graphics* window and shows the progress of the solvers.

   While the simulation is running, you can click the tabs at the top of the *Graphics* window to view scenes and plots.

   During the run, you can stop the process by clicking (**Stop**) on the toolbar. If you do stop the simulation, you can click (**Run**) to resume it.

   If left alone, the simulation continues until 90 seconds of physical time are complete. When the last iteration is complete, the following message is displayed in the *Output* window:

   ```
   Stopping criterion Maximum Physical Time satisfied.
   ```

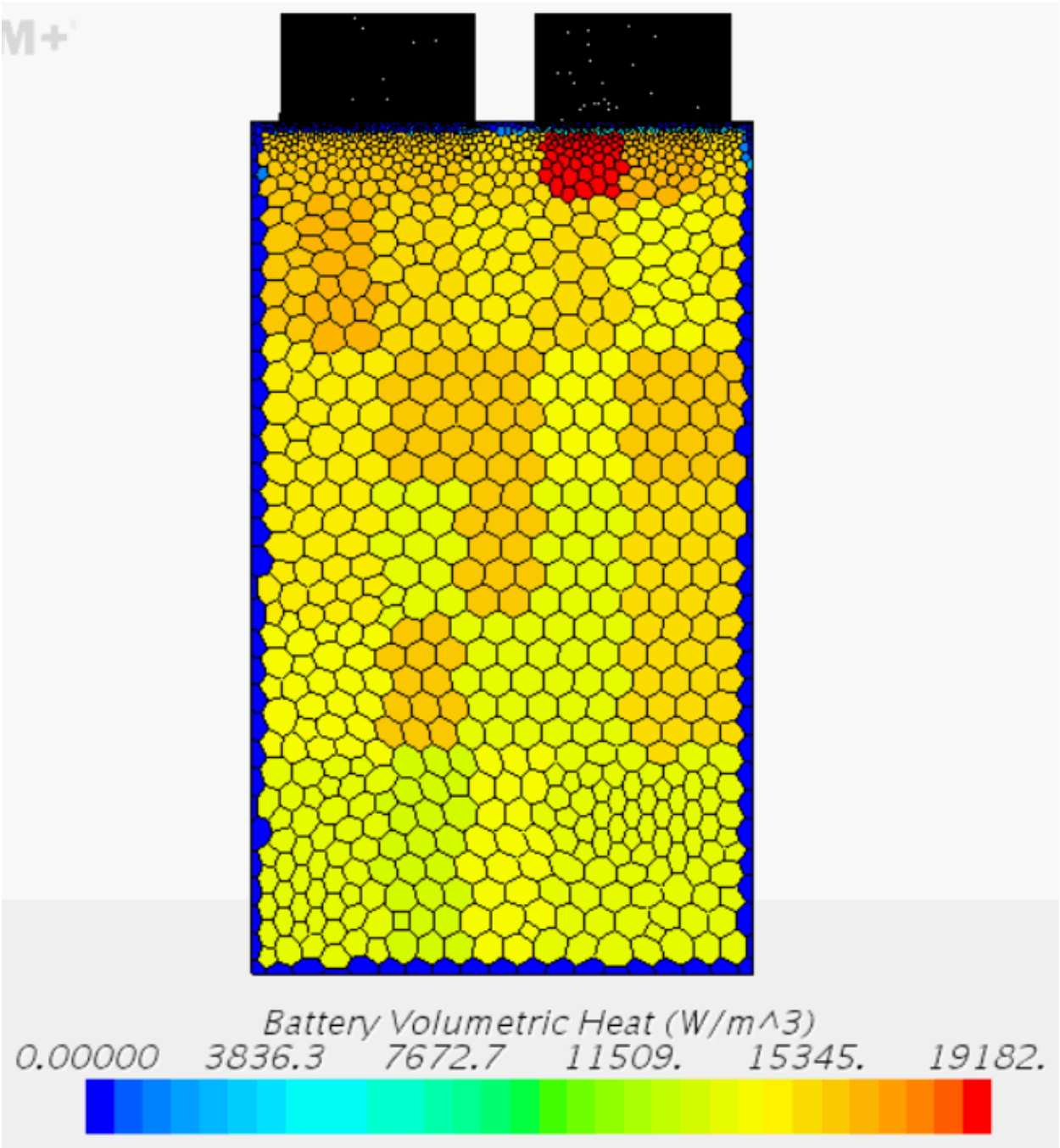
2. When the simulation has finished running, save it.

Visualizing the Results

View the results of the simulation.

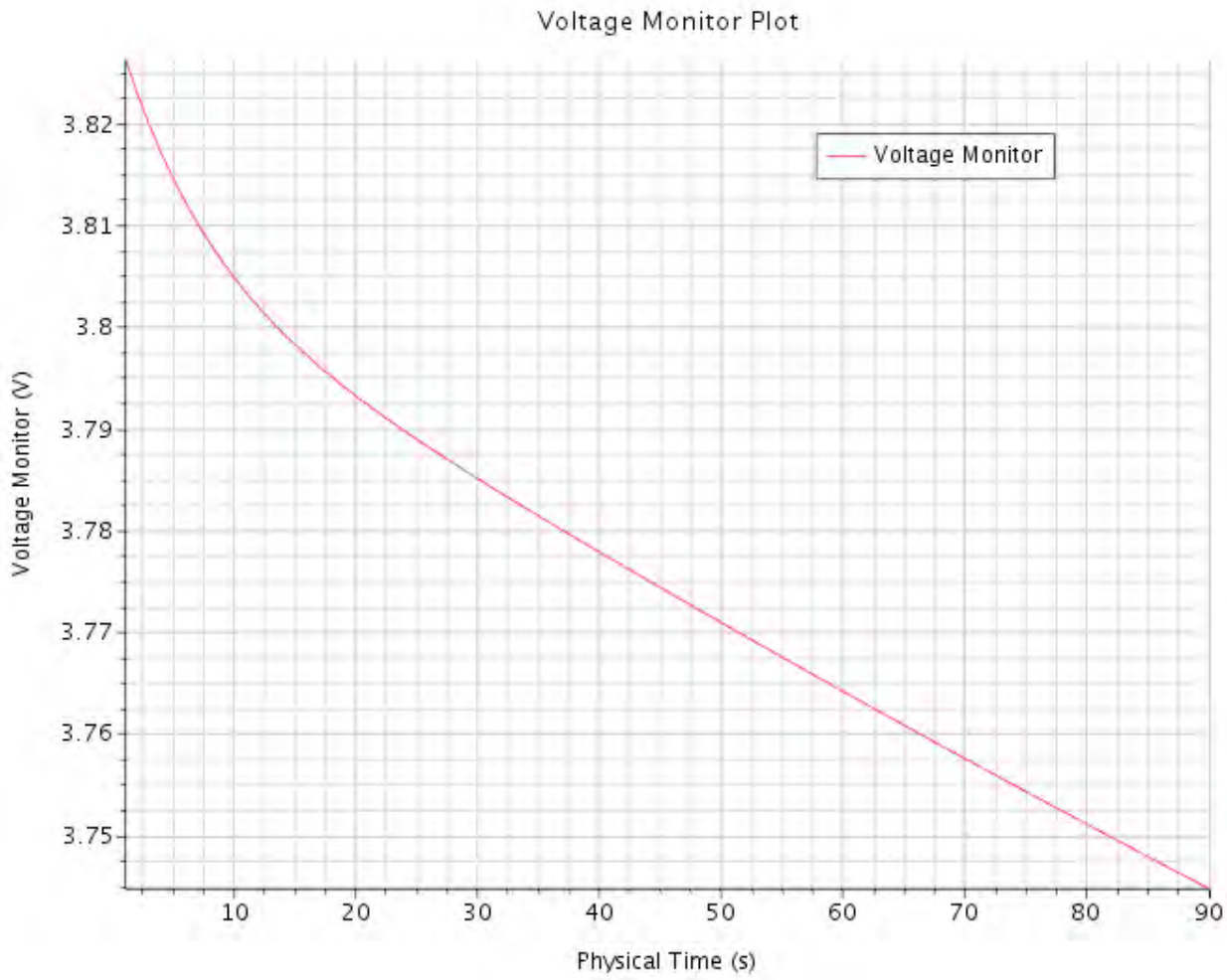
To visualize the results:

1. Select *Scalar Scene 1*.



The scalar scene is shown with the mesh turned on. It shows you that the highest concentration of battery heat generation occurs near the tab roots area in the battery cell, as expected.

2. Select the *Voltage Monitor Plot*.



The voltage plot displays the steady discharge of the battery during the first 90 seconds.

Running a STAR-CCM+ Simulation with 3 Cells in Series

Use the battery cell from the previous simulation to model a battery with three cells in series.

Model a battery with three cells in series, also called a string. A tab connector is imported into STAR-CCM+ to model the physical connections between the cells. Complete Part 2 of the tutorial (Running a STAR-CCM+ Simulation with a Single Cell) before attempting this part.

1. Copy the tab connector geometry file `tabConnector.x_b` from the `doc/startutorialdata/battery/data` subdirectory of your STAR-CCM+ installation directory into the `BasicCell` directory.

Clear the solution from the single cell simulation.

2. Select **Solution** > **Clear Solution** in the menu bar.
3. In the *Clear Solution* dialog, click **OK**.

4. Save the simulation.

Contents:

Setting Up the Battery Module
Setting Up Tab Connections
Generating the Battery Module
Generating the Volume Mesh
Running the Simulation
Visualizing the Results

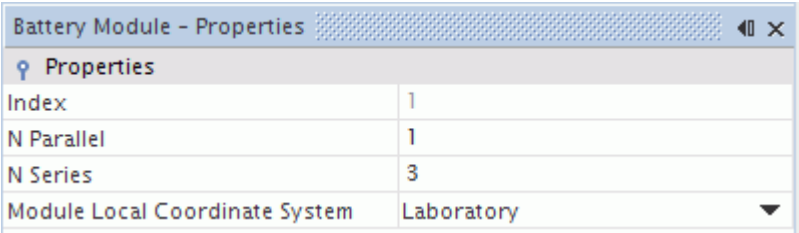
Setting Up the Battery Module

Set up the battery module with three cells in series.

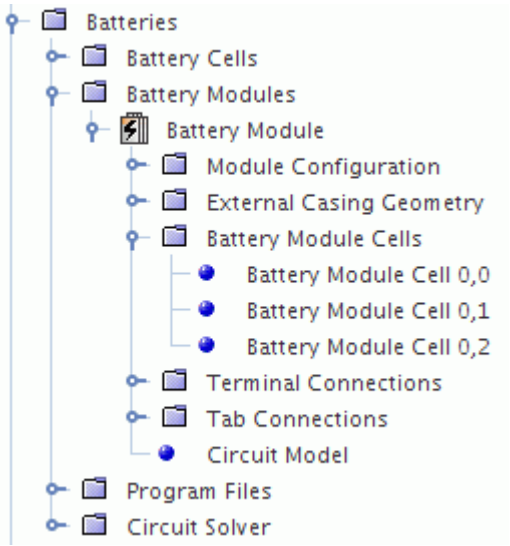
To set up the battery module:

Increase the number of cells in the battery module from 1 to 3.

1. Select the **Batteries** > **Battery Modules** > **Battery Module** node.
2. In the *Battery Module - Properties* window, set *N Series* to 3.



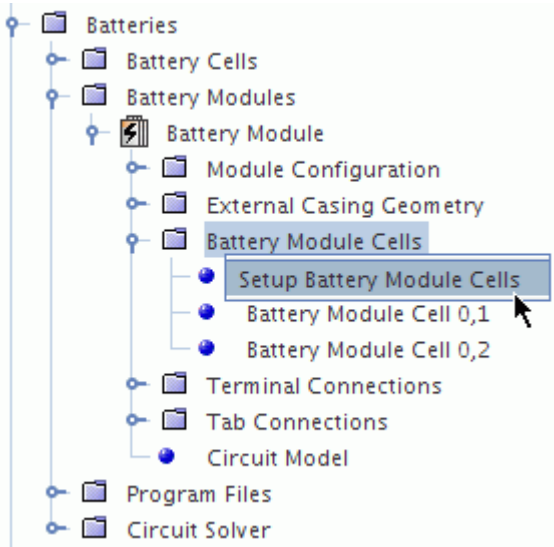
3. Expand the **Battery Module** > **Battery Module Cells** node.



Three cells are shown. The cells are named according to their position in the battery module matrix.

To define the position and orientation of the cells in the module:

4. Right-click the **Battery Module Cells** node and select **Setup Battery Module Cells**.



The **Edit** tab opens which contains the **Setup Battery Module Cells** panel.

In the *Setup Battery Module Cells* panel, there are two places to specify cell spacing values:

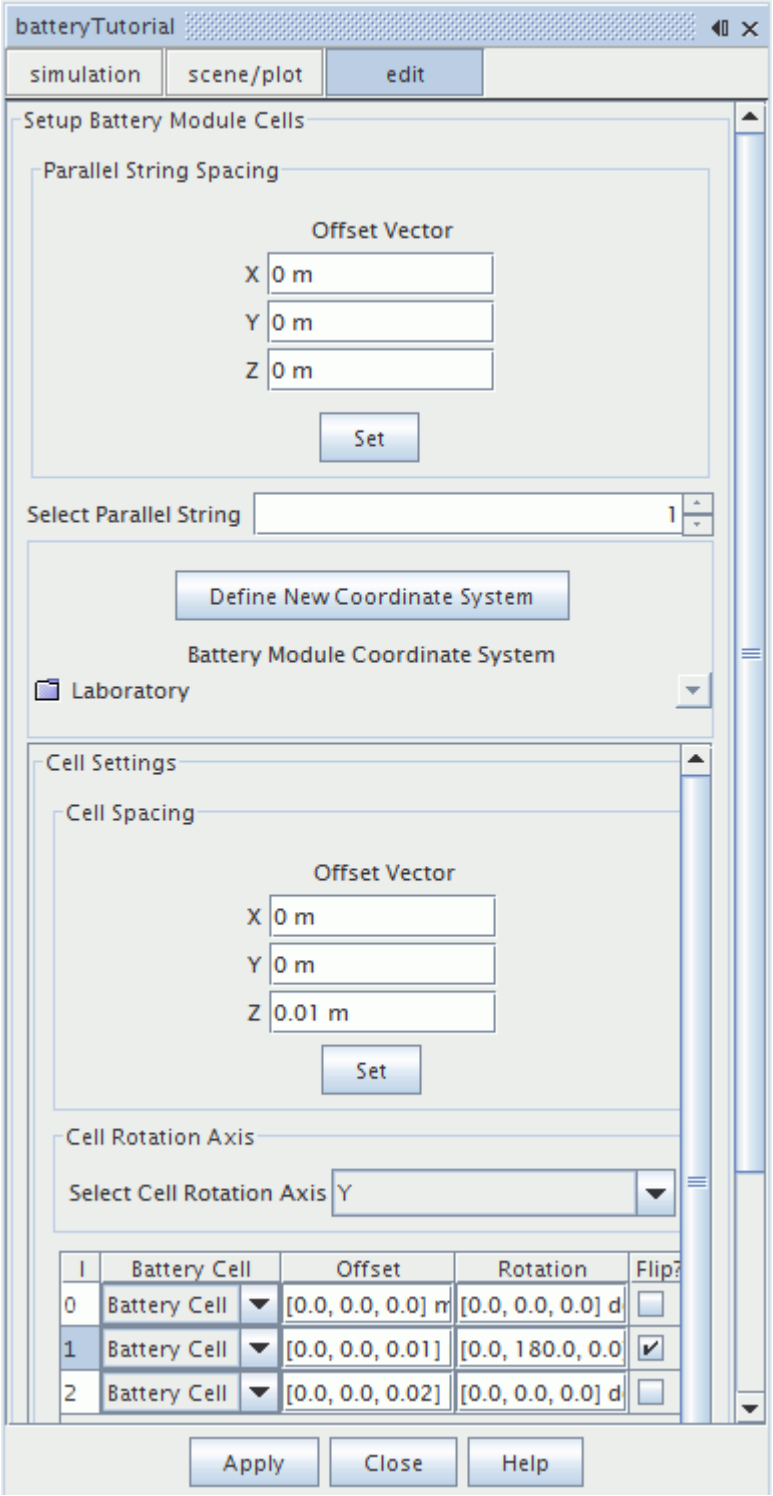
- The *Parallel String Spacing* group box allows you to define the relative position of multiple strings that are connected in parallel.
- The *Cell Spacing* group box allows you to define the relative position of cells that are connected in series within a particular string.

The cells are arranged in a stack formation, with an offset from one cell to the next of 0.01m. Use the default *Offset Vector* in the **Cell Spacing** box to define the position of the 3 cells. The **Parallel String Spacing** is not required.

5. In the **Cell Spacing** group box, make sure that the *Offset Vector* is set to `[0, 0, 0.01]m`.
6. Click **Set**.

At the bottom of the panel, notice that each cell in the table now has a different offset position. In addition to defining the cell spacing, flip the middle cell in the series. This positions the positive tab of the middle cell next to the negative tab of the first cell, which is necessary for the tab connections in this case.

7. Activate **Flip?** for the middle cell.



8. Click **Apply** and then click **Close**.

Setting Up Tab Connections

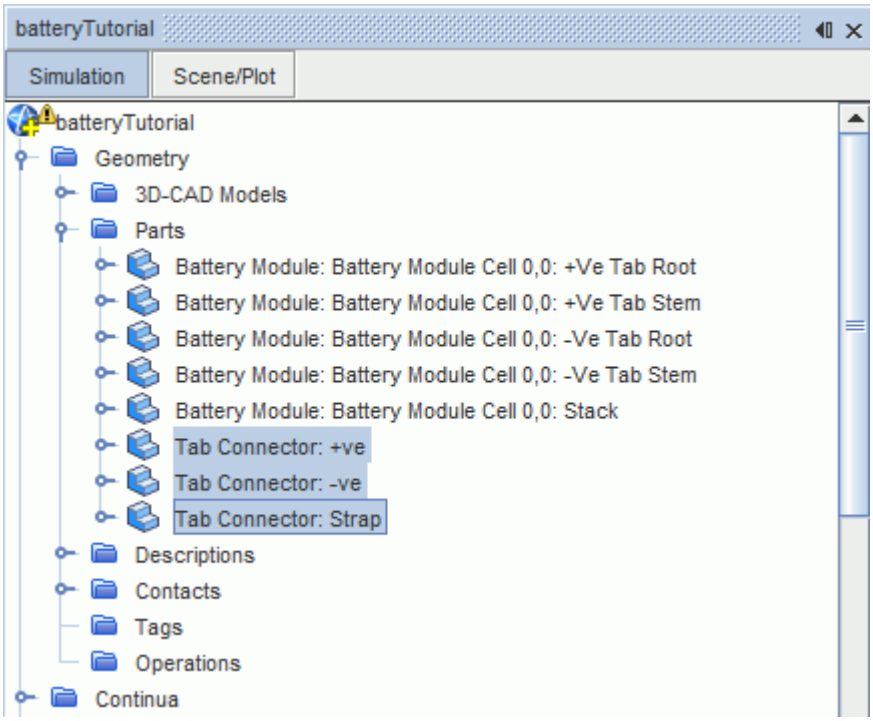
Set up tab connectors to define the connection between the positive and negative tabs of the battery cells.

A physical connection between the positive and negative tabs of the battery cells can be modeled using tab connectors. For a regular formation of cells, a single tab connector can be used, and STAR-CCM+ uses copies of the connector to join all the tabs in the battery module. A simple tab connector has been prepared for this tutorial.

To set up tab connectors:

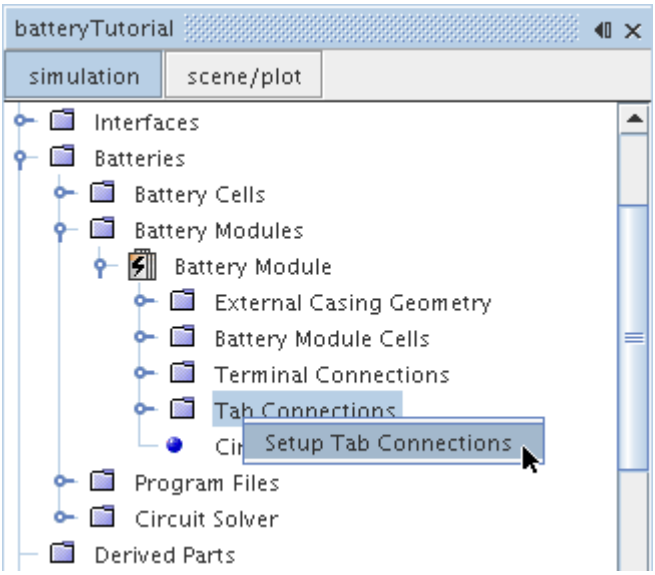
1. Select **File** > **Import** > **Import Surface Mesh** from the menu bar.
2. In the *Open* dialog, select the `tabConnector.x_b` file.
3. Click **Open**.
4. In the *Import Surface Options* dialog, do the following:
   a) Set **Import Mode** to **Create New Part**.
   b) Deactivate **Open Geometry Scene After Import** under **Import Options**.
5. Click **OK**.

The imported geometry consists of 3 parts: a positive post, a negative post, and a strap. These parts are added to the **Geometry** > **Parts** node in the STAR-CCM+ tree.



Use the imported tab connector parts in the battery module definition.

6. Right-click the **Batteries** > **Battery Modules** > **Battery Module** > **Tab Connections** node and select **Setup Tab Connections**.



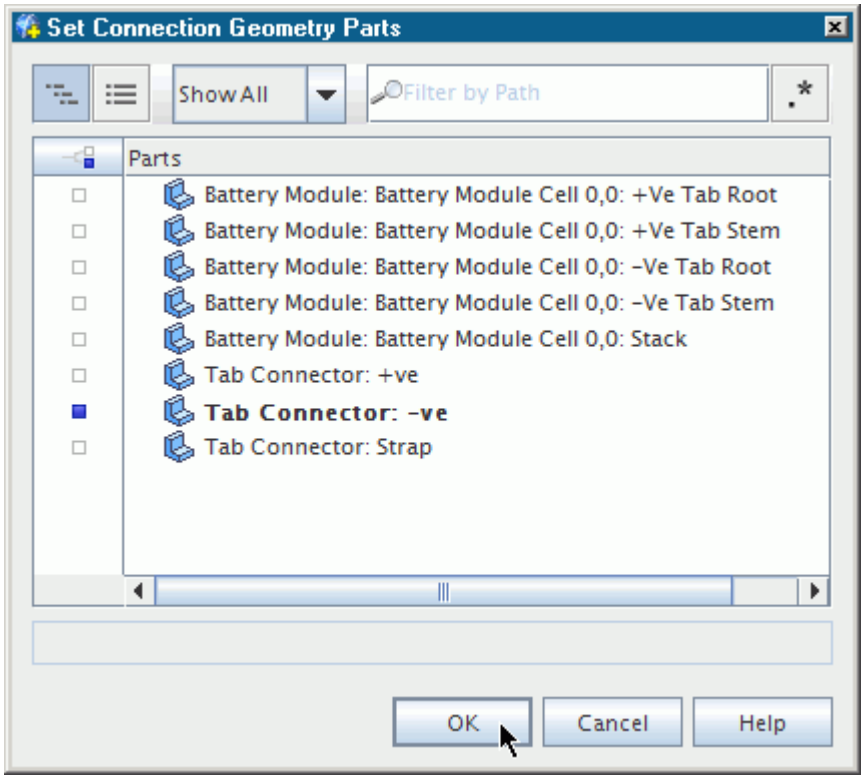
The *Setup Tabs Connections* in-place dialog appears.

7. Set **Connections Setup Method** to **Copy Single Tab Connector**.

The positive tab of the first cell in a string of cells that are connected in series generally becomes the positive terminal for the battery. Similarly, the negative tab on the last cell in a string of cells that are connected in series becomes the negative terminal for the battery. This means that the positive tab on the first cell and the negative tab on the last cell in the series remain open when the tab connectors are added.

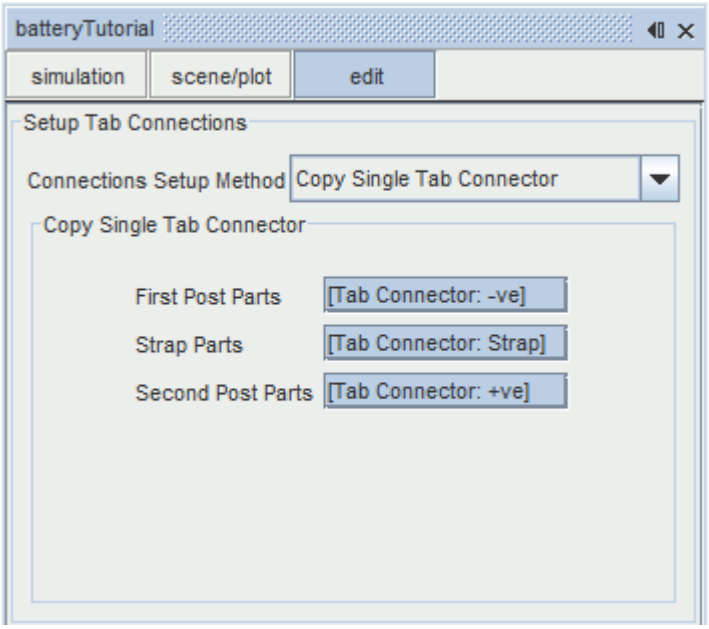
The tab connector parts are assigned using the *First Post Parts*, *Strap Parts*, and *Second Post Parts* options.

8. Click to the right of the *First Post Parts* property.
9. In the *Set Connection Geometry Parts* dialog, select **Tab Connector: -ve**.



10. Click **OK**.
11. Set the *Strap Parts* to **Tab Connector: Strap**.
12. Set the *Second Post Parts* to **Tab Connector: +ve**.

    The completed dialog is shown below:



13. Click **Apply**, then click **Close**.

Generating the Battery Module

Generate the battery module.

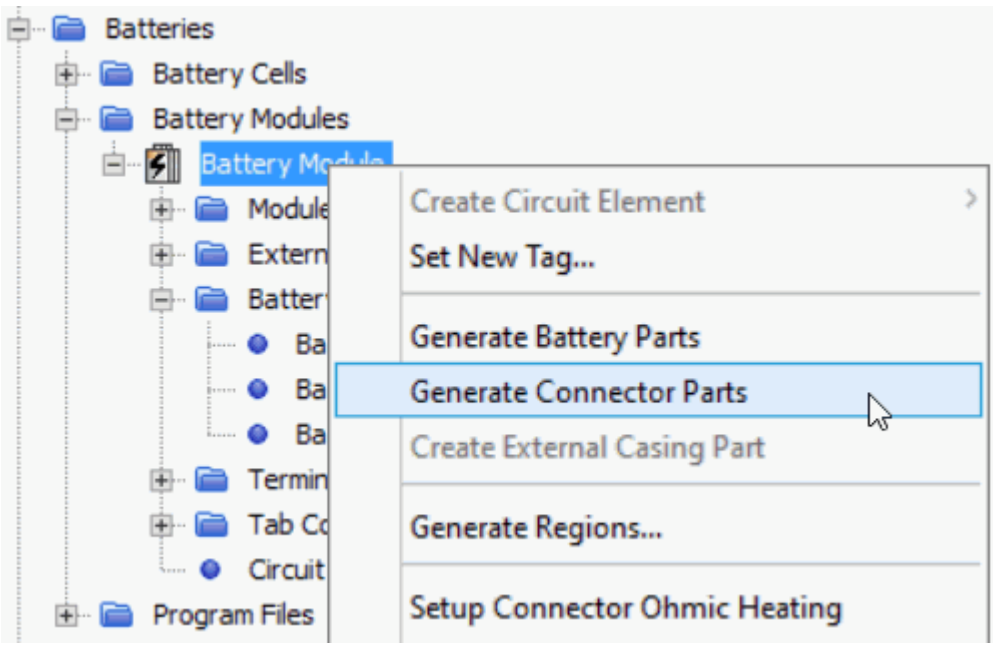
The three-cell battery is connected to the same electrical load, so no changes to the program file or circuit solver are necessary.

To generate the battery module:

1. To display the 3D-CAD View in the *Graphics* window, select the **3D-CAD View 1** tab.
2. Right-click the **Battery Modules** > **Battery Module** node and select **Generate Battery Parts**.

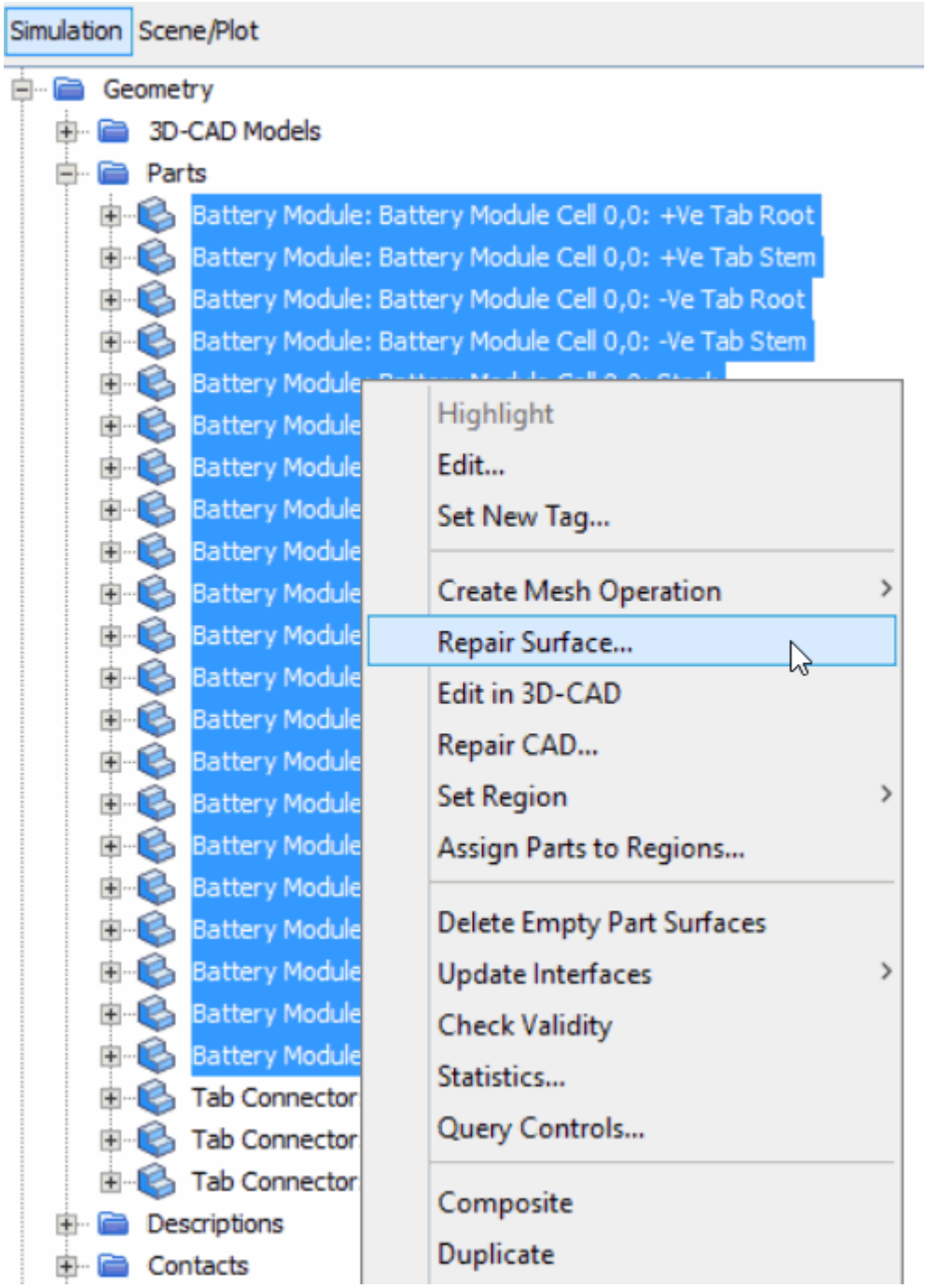
The battery module contains three cells. Parts have been created for each battery cell in the module.

3. Right-click the **Battery Modules** > **Battery Module** node and select **Generate Connector Parts**.



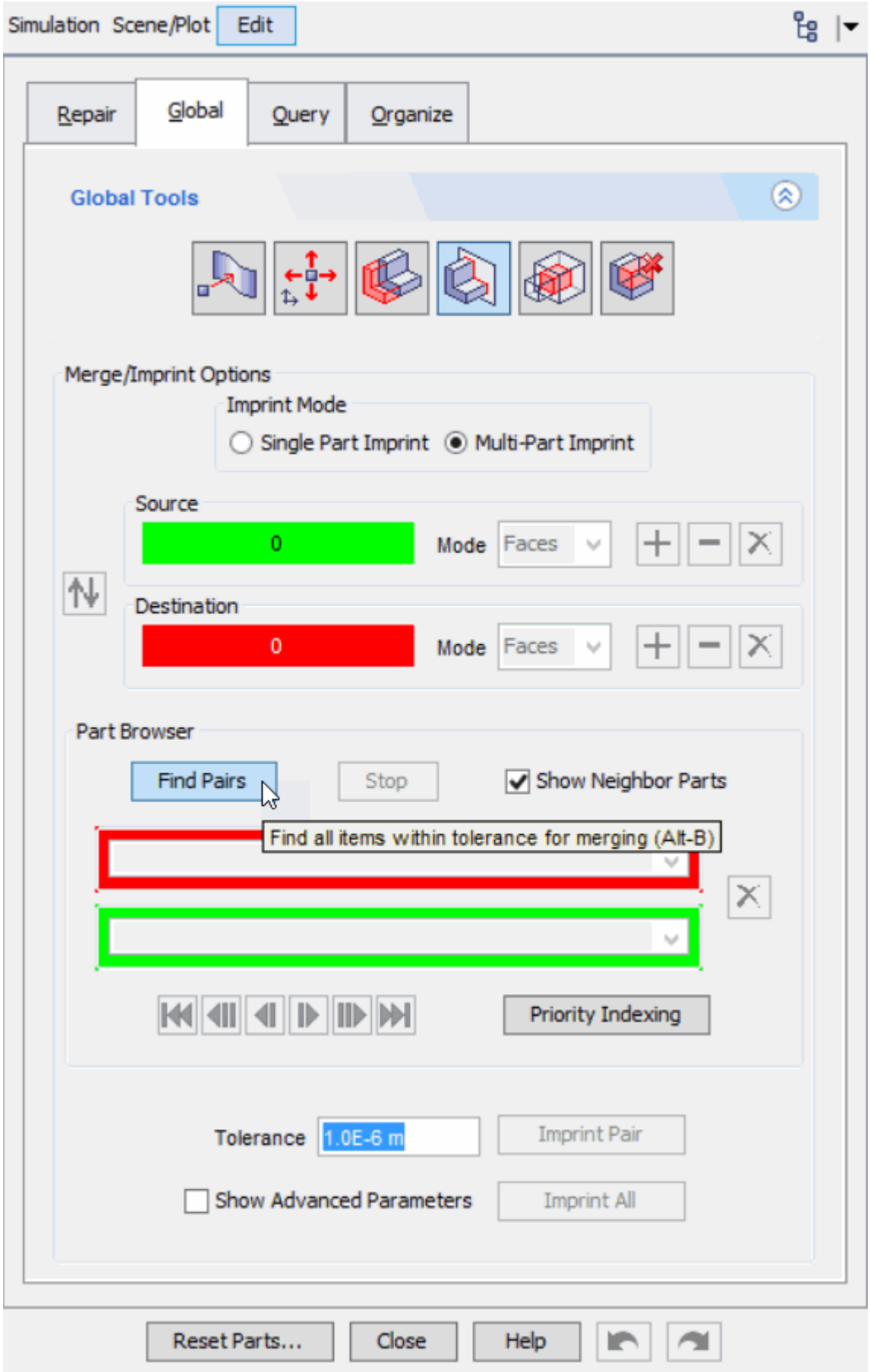
Parts are created for each tab connection. Before generating regions, imprint the tab connector and tab parts.

4. Expand the **Geometry** > **Parts** node.
5. Select all the **Battery Module** parts by holding the `Shift` key while selecting the first and last part.
6. Right-click one of the highlighted nodes and select **Repair Surface...**.

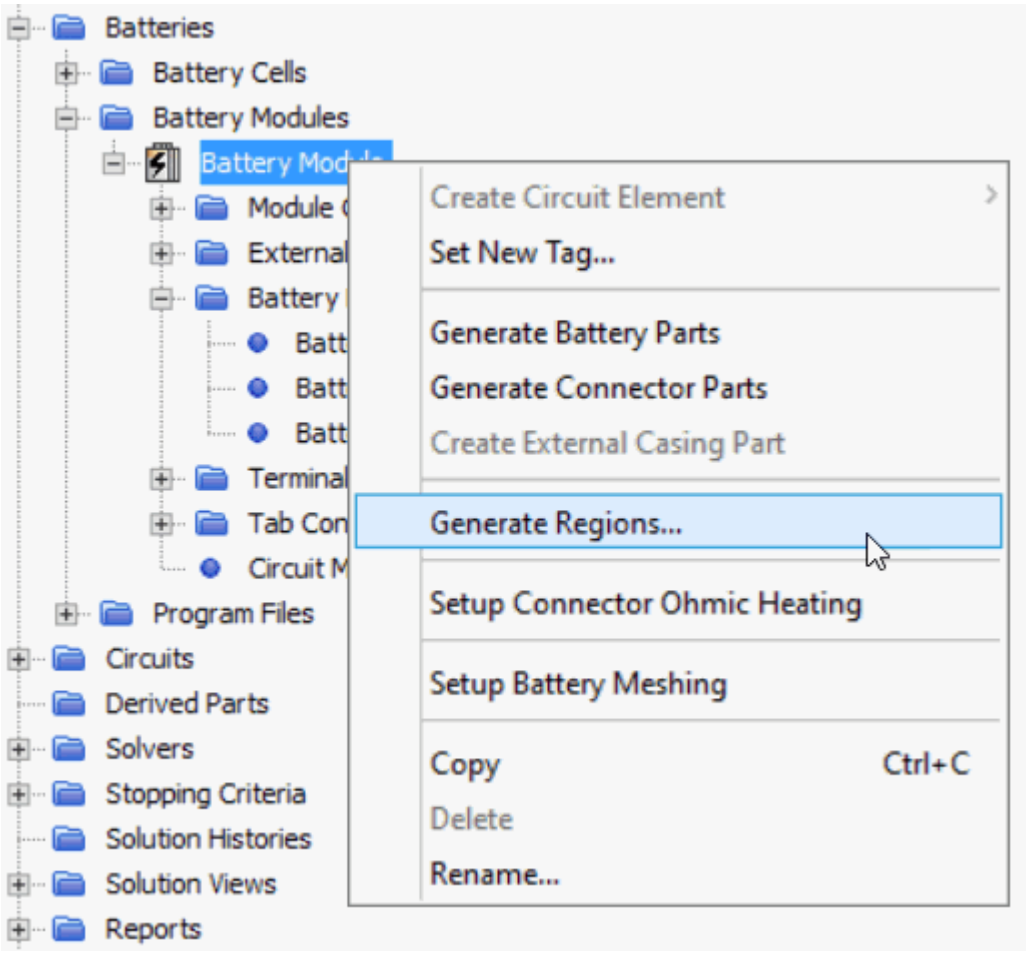


7. In the *Surface Repair Options* dialog, click **OK** to launch the surface repair tool for the selected parts.

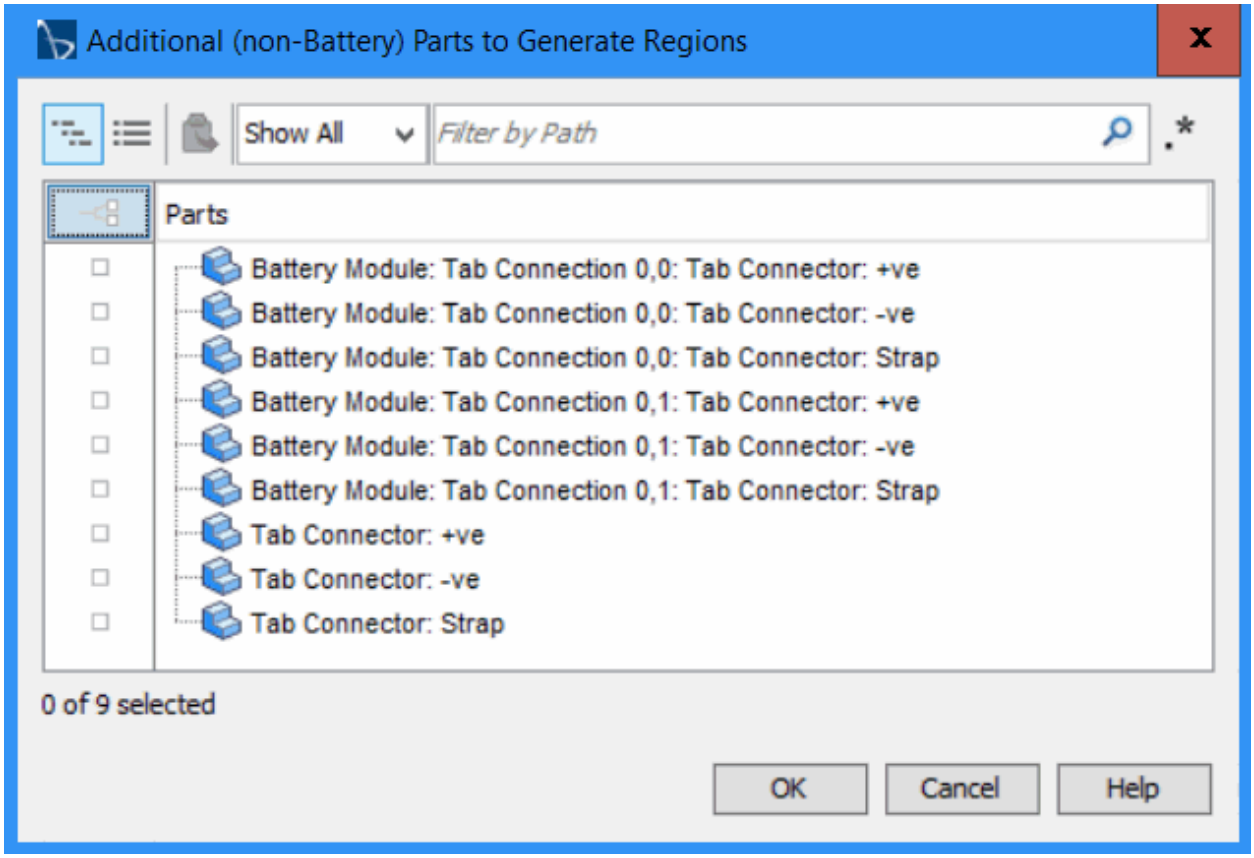
8. On the *Global* tab, click (**Merge/Imprint single or multiple parts**).
   The *Merge/Imprint Options* panel appears.
9. In the *Merge/Imprint Options* box, set the **Imprint Mode** to **Multi-Part Imprint**.
10. Set the **Tolerance** to `1.0E-6 m`.
11. In the *Part Browser* box, click **Find Pairs**.


Simulation
Scene/Plot
Edit
Repair
Global
Query
Organize
Global Tools
Merge/Imprint Options
Imprint Mode
Single Part Imprint
Multi-Part Imprint
Source
0
Mode
Faces
Destination
0
Mode
Faces
Part Browser
Find Pairs
Stop
Show Neighbor Parts
Find all items within tolerance for merging (Alt-B)
Priority Indexing
Tolerance
1.0E-6 m
Imprint Pair
Show Advanced Parameters
Imprint All
Reset Parts...
Close
Help

12. Once all the pairs are found, click **Imprint All** and then **Close**.
13. As with the parts and connector parts, right-click the **Batteries** > **Battery Modules** > **Battery Module** node and select **Generate Regions...**.



14. In the *Additional (non-Battery) Parts to Generate Regions* dialog, click (**Column Selection**) to deselect all parts.
15. Click **OK**.



STAR-CCM+ automatically creates interfaces between the tabs and the connectors, based on the imprinted parts.

16. Expand the **Interfaces** node and select each interface in turn.

    The selected interface is highlighted in the scene.

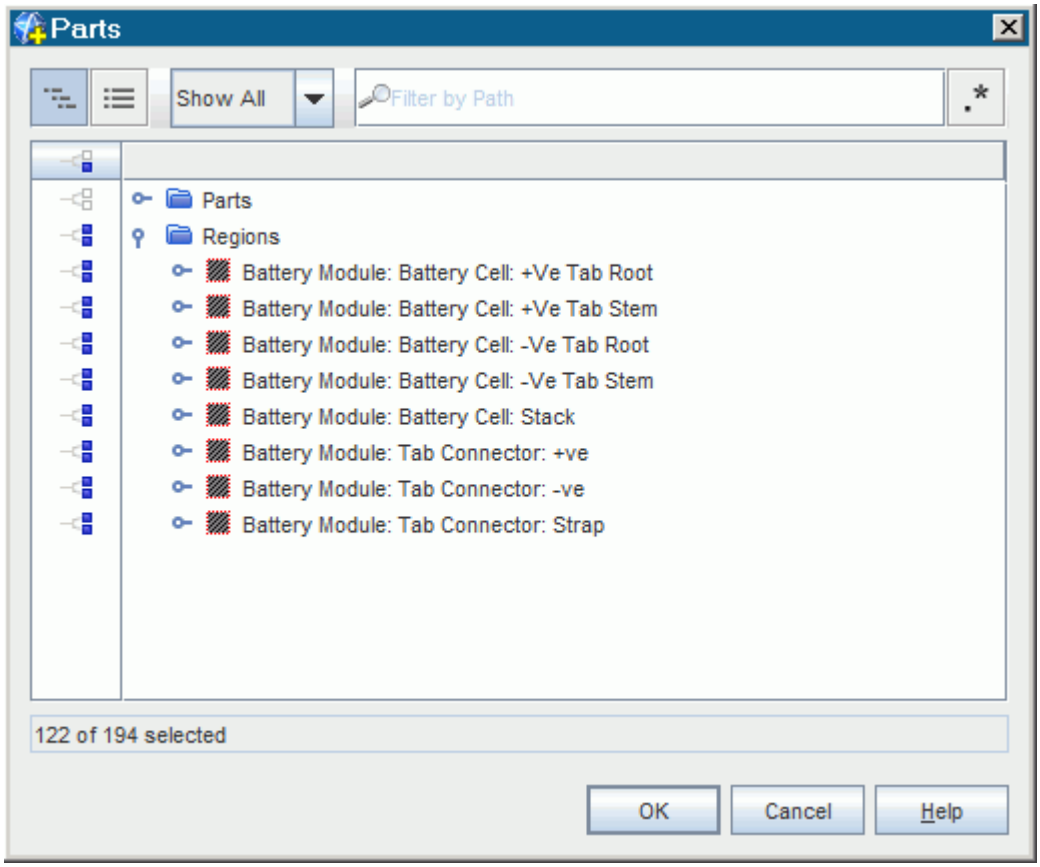
Generating the Volume Mesh

Define the battery mesh and set the appropriate properties.

Use the same mesh settings to generate the mesh for the three-cell battery.

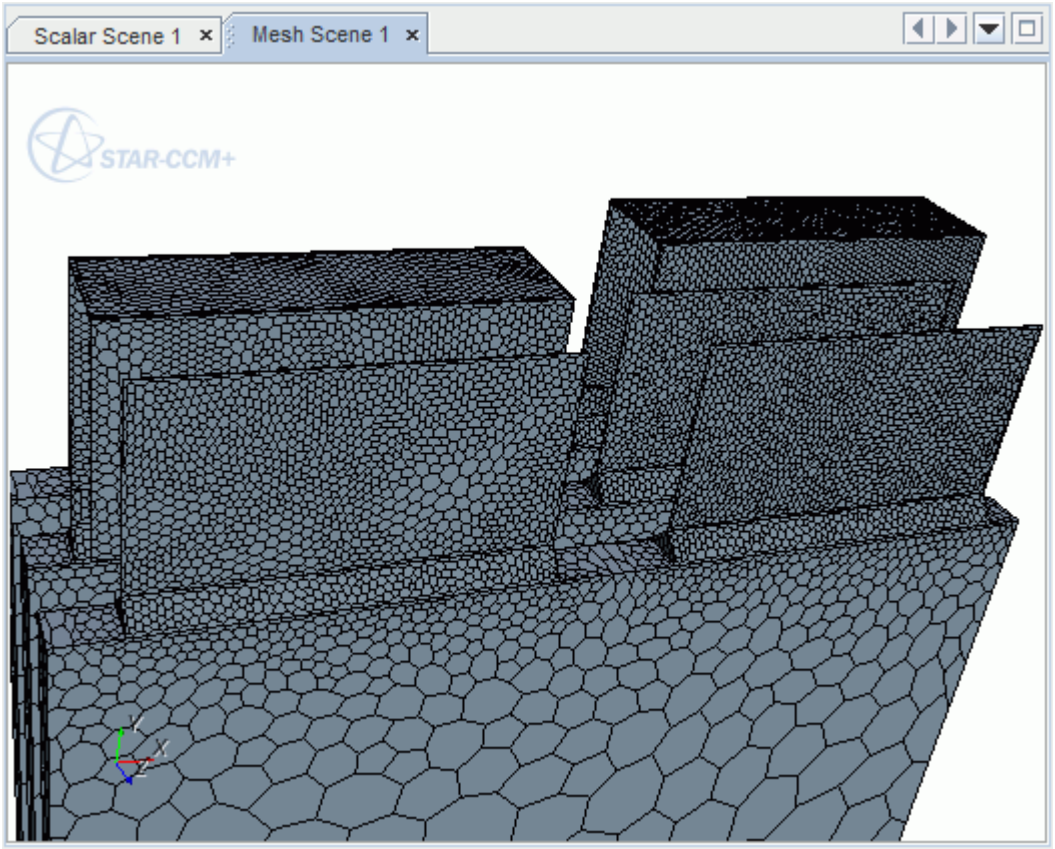
To generate the volume mesh:

1. Click (**Generate Volume Mesh**).
2. Display *Mesh Scene 1* in the *Graphics* window.
3. Click **Scene/Plot**.
4. Right-click **Mesh Scene 1** > **Displayers** > **Mesh 1** > **Parts** and select **Edit**.
5. In the *Parts* dialog, expand the **Regions** node then click **Select Children** on the left of the **Regions** node.



6. Click **OK**.

The mesh scene is shown below.



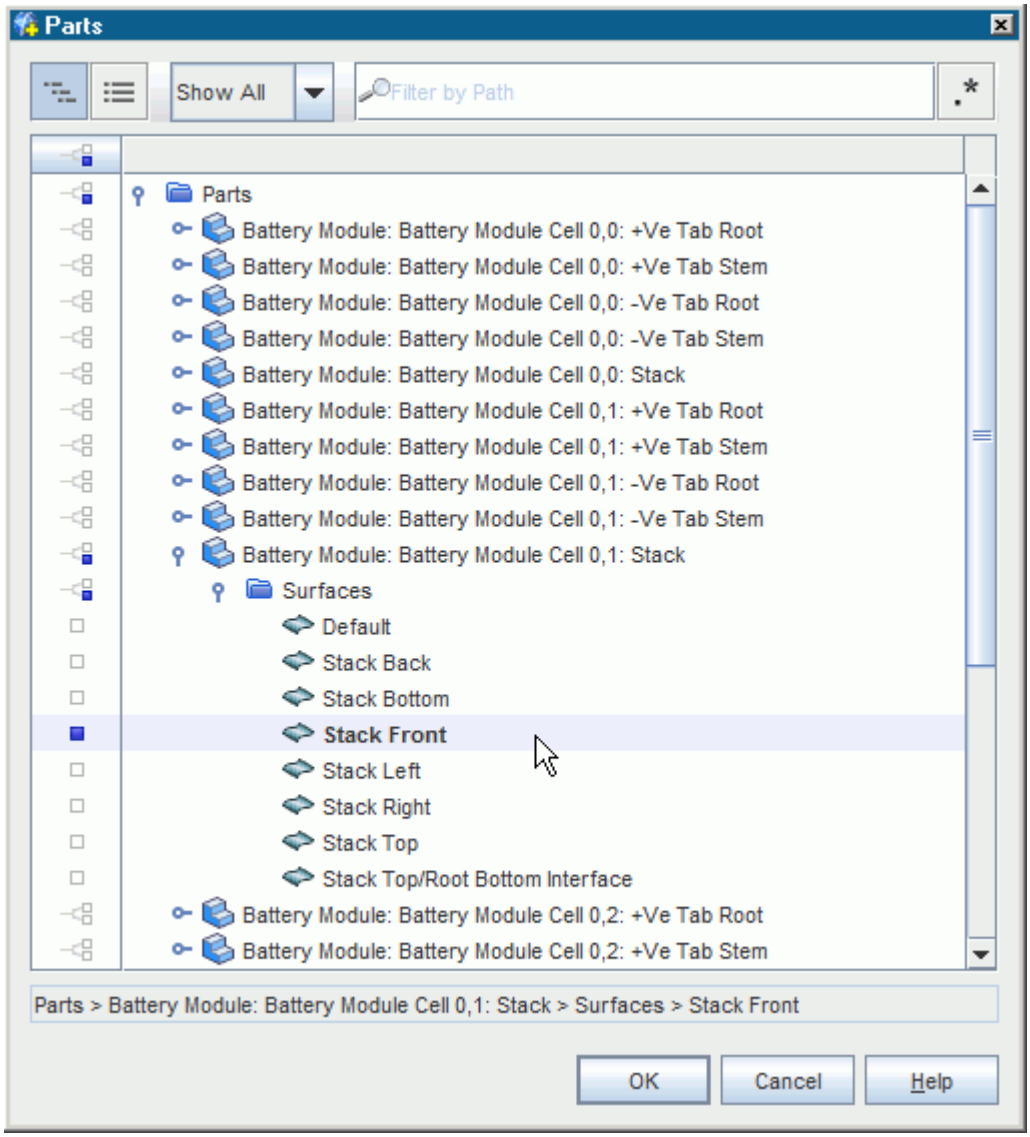
Running the Simulation

Modify the simulation and run it again.

Before running the simulation, modify the scalar scene to display the battery volumetric heat on the middle cell of the stack.

You make this modification by assigning geometry parts to the scalar displayer.

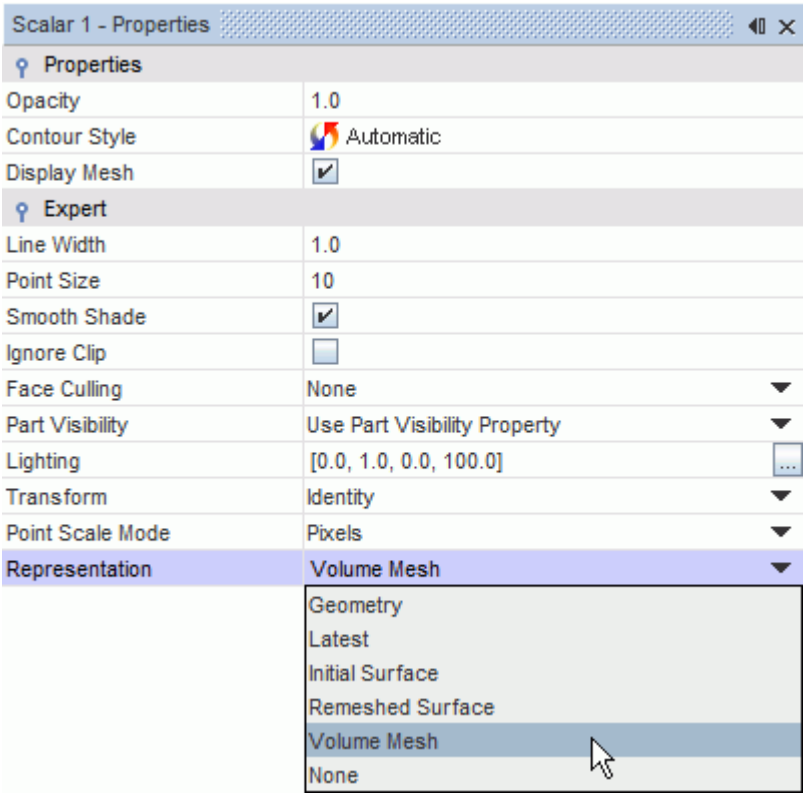
1. Show *Scalar Scene 1* in the *Graphics* window.
2. Under **Scene/Plot**, right-click the **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Parts** node and select **Edit**.
3. Click **Column Selection** to deselect all parts.
4. Expand the **Parts** > **Battery Module: Battery Module Cell 0,1: Stack** node and select **Stack Front**.



5. Click **OK**.

Make sure that the scene is displaying the volume mesh representation.

6. Select the **Scalar Scene 1** > **Displayers** > **Scalar 1** node and make sure that the **Representation** is set to **Volume Mesh**.



7. Save the simulation under the name `batteryTutorial_threeCells.sim`.

You are now ready to run the simulation.

8. Click (**Run**) in the *Solution* toolbar.
9. When the simulation has finished running, save it.

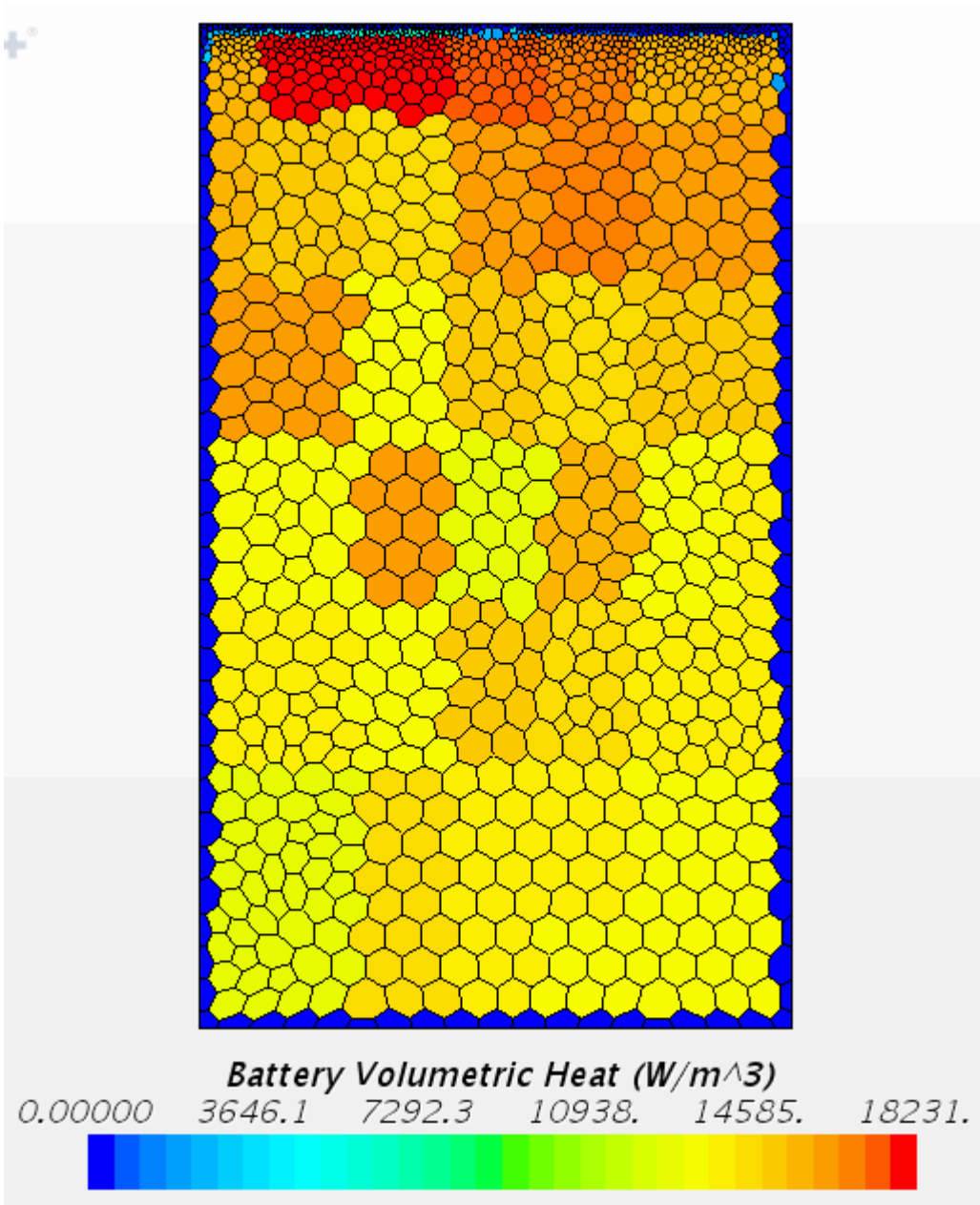
Visualizing the Results

View the results of the simulation.

To view the results of the simulation:

1. Select *Scalar Scene 1*.

   The Scalar Scene is shown below.



The scalar scene demonstrates that battery heat generation does not display any changes when more cells are added in series.

As seen in Part 2 (Running a STAR-CCM+ Simulation with a Single Cell), the highest concentration of battery heat generation is seen near the tab roots area.

Summary

This section outlines the STAR-CCM+ features that are introduced by the Battery Simulation Module: Cell Thermal Analysis tutorial.

This tutorial covered the following STAR-CCM+ features:

- Setting up the battery physics model.
- Importing a battery cell from Battery Design Studio®.
- Loading program data from Battery Design Studio®.
- Generating a battery module by creating parts, connector parts, and regions.
- Generating a volume mesh using the battery mesher.
- Creating a circuit.
- Setting up reports, monitors, and plots.
- Post-processing voltage and battery heat generation per unit volume.

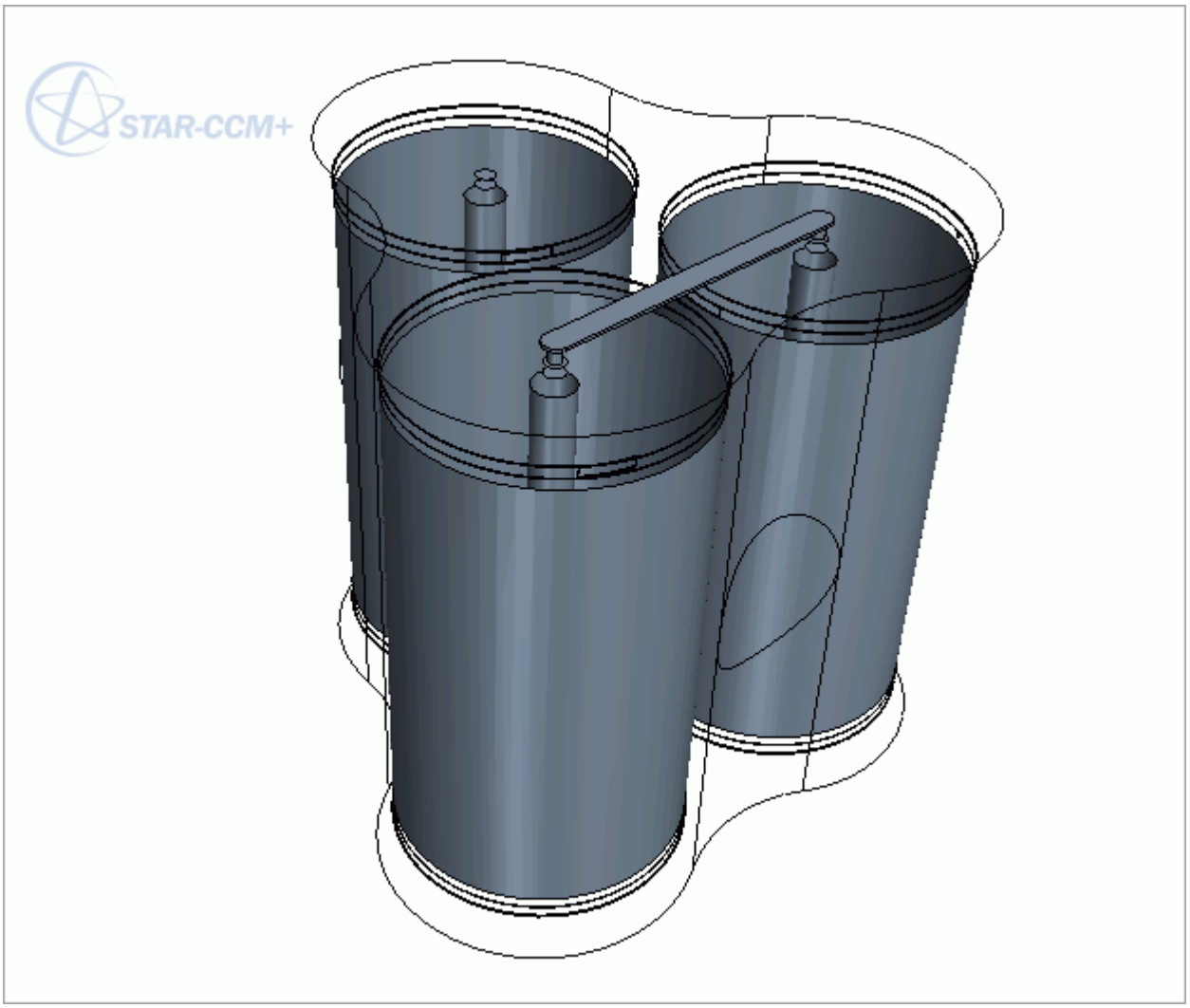
Cylindrical Cells: Cell Thermal Analysis

This tutorial demonstrates how to model a battery module in STAR-CCM+ using cylindrical cells. It simulates three cylindrical cells in series in a staggered configuration with an external casing.

The `tbm` file, `prg` file, a tab connector, and the external casing, are imported into STAR-CCM+.

Note: For battery modeling, STAR-CCM+ requires a `batterysim` license in addition to the other license features.

The three cylindrical cells are connected in series and are discharged for 200 seconds with a load of 22.11 A. The cells are cooled with a draft of 1 m/s. You model both the heat generation in the battery and the air flow in the external casing.



Contents:

Prerequisites
Creating the Simulation
Selecting the Physics Models
Importing the Cylindrical Cell
Defining the Cell Configuration
Setting Up Tab Connectors
Generating the Battery Module
Imprinting the Tab Connectors
Defining the External Casing
Generating Regions
Generating the Volume Mesh
Selecting the Physics Models for the External Casing
Creating a Circuit
Setting Solver Parameters and Stopping Criteria
Reporting, Monitoring, and Plotting
Preparing a Scalar Scene
Running the Simulation
Visualizing the Results
Summary

Prerequisites

The instructions in the Cylindrical Cells: Cell Thermal Analysis tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Creating the Simulation

Launch STAR-CCM+ and create a simulation.

To create the simulation:

Create a working directory and copy the files that are required for this tutorial.

1. Create a working directory called `CylindricalCell`.
2. Copy the following files from the `doc/startutorialdata/battery/data` subdirectory of your STAR-CCM+ installation directory into the `CylindricalCell` directory:
   - `TutorialCellCylindrical.tbm`

- `Tutorial3CDischargeCylindrical.prg`
- `post1Cylindrical.x_b`
- `post2Cylindrical.x_b`
- `strapCylindrical.x_b`
- `externalCasingCylindrical.x_b`

Create the simulation.

3. Launch STAR-CCM+.
4. Start a simulation.
5. Save the simulation as `batteryTutorialCylindrical.sim` in the `CylindricalCell` directory.

Selecting the Physics Models

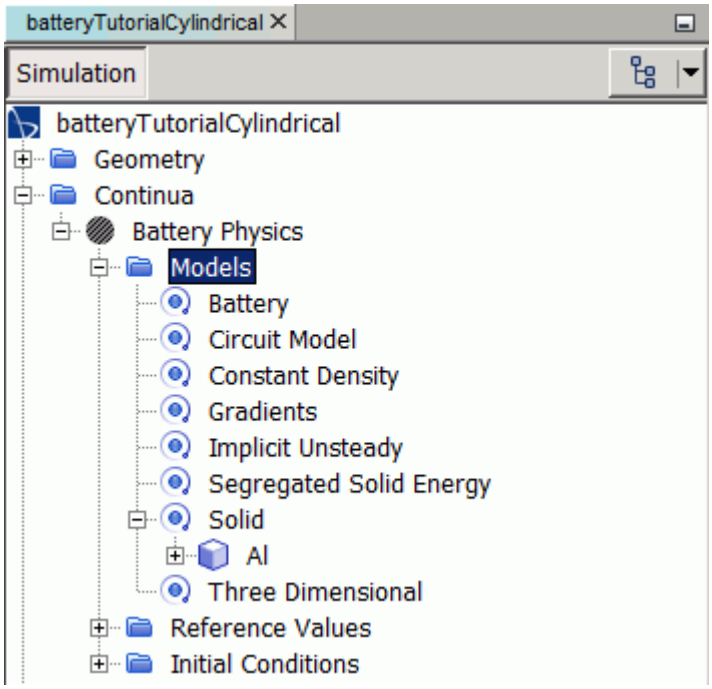
Physics models define the primary variables of the simulation and what mathematical formulation is used to generate the solution.

To select the physics models:

1. Create a physics continuum and rename the **Continua** > **Physics 1** node to **Battery Physics**.
2. For the physics continuum, **Continua** > **Battery Physics**, select the following models in order:

Group Box	Model
Space	**Three Dimensional**
Time	**Implicit Unsteady**
Material	**Solid**
Optional Models	**Battery** **Circuit Model** (selected automatically)
Energy	**Segregated Solid Energy** **Gradients** (selected automatically)
Equation of State	**Constant Density**

3. Click **Close**.
4. To review the models, open the **Battery Physics** > **Models** node.



The battery pre-processing is carried out primarily under this node.

5. Save the simulation.

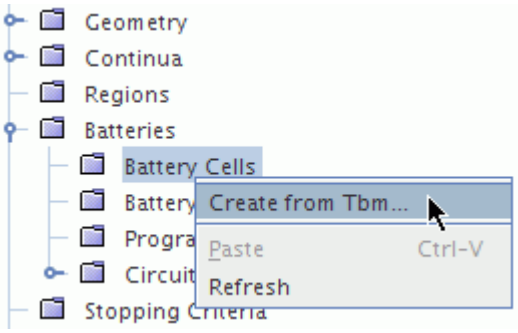
Importing the Cylindrical Cell

The battery file created in BDS is imported into STAR-CCM+ as a battery cell.

The battery file from Battery Design Studio®, `TutorialCellCylindrical.tbm`, is an idealized cylindrical cell. It uses the NTG3D model.

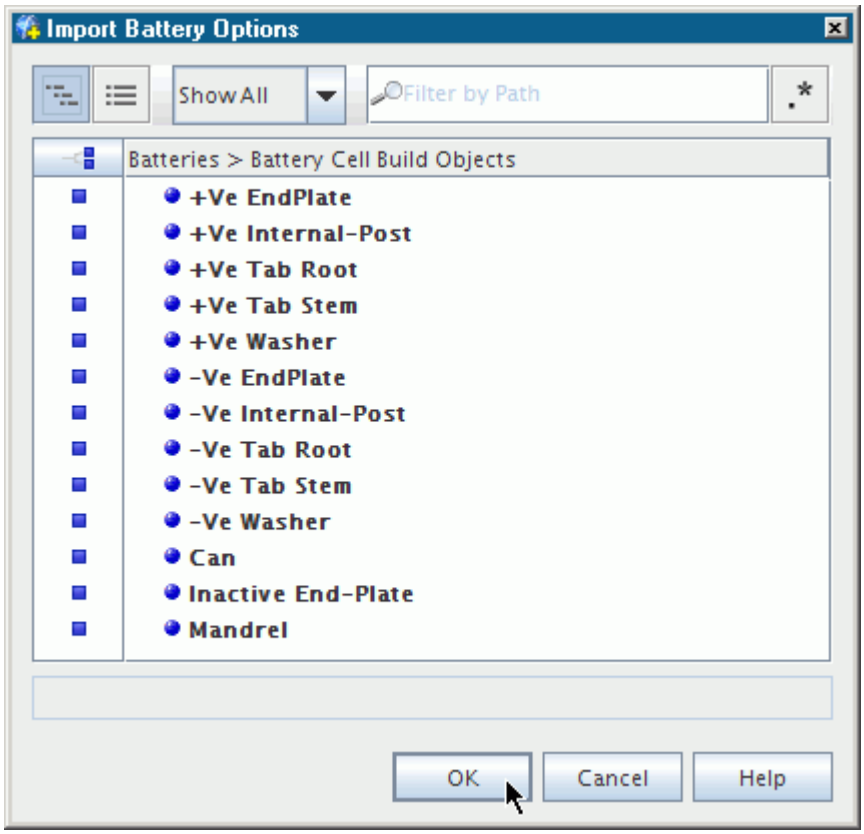
To import the cylindrical cell:

1. Expand the **Batteries** node.
2. Right-click **Batteries** > **Battery Cells** and select **Create from Tbm...**.

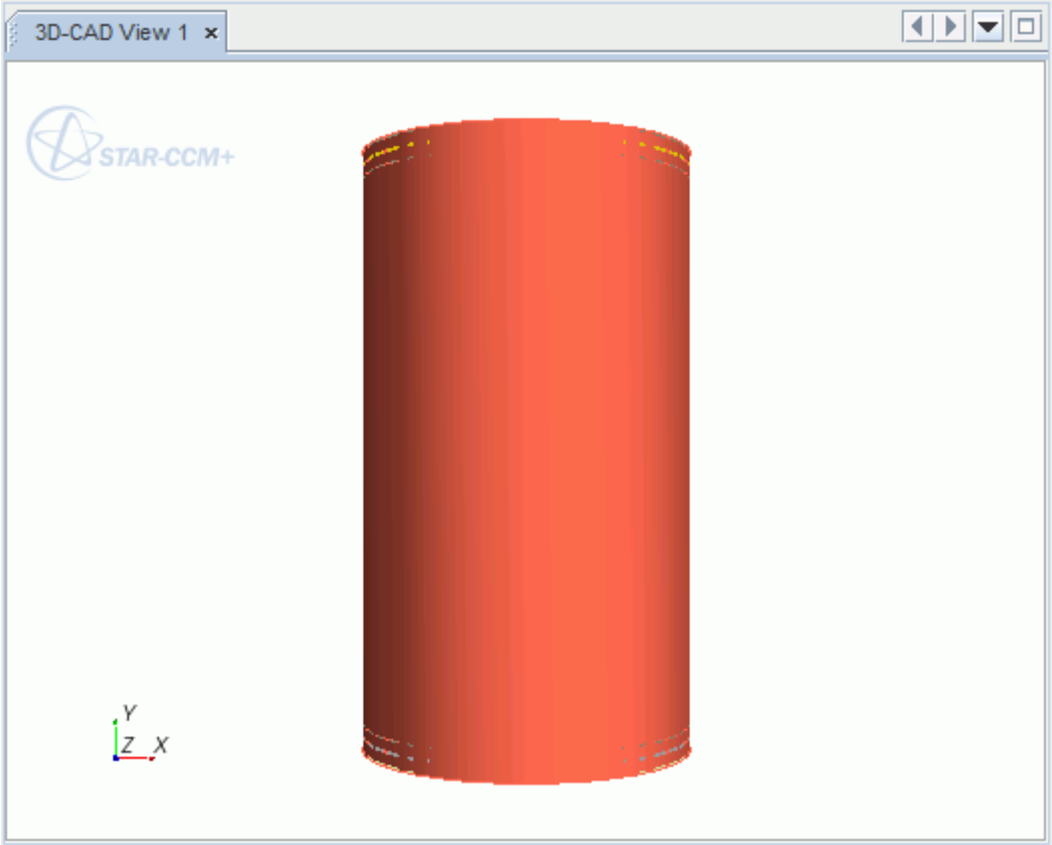


3. In the *Import Battery Data From File* dialog, navigate to your working directory, `CylindricalCell`, and select the file `TutorialCellCylindrical.tbm`.

4. Click **Open**.
5. In the *Import Battery Options* dialog, ensure that all objects are selected, then click **OK**.



During the import process, STAR-CCM+ builds an idealized model of the cell using the 3D-CAD module. The battery cell model appears in the *3D-CAD View 1* window.



Defining the Cell Configuration

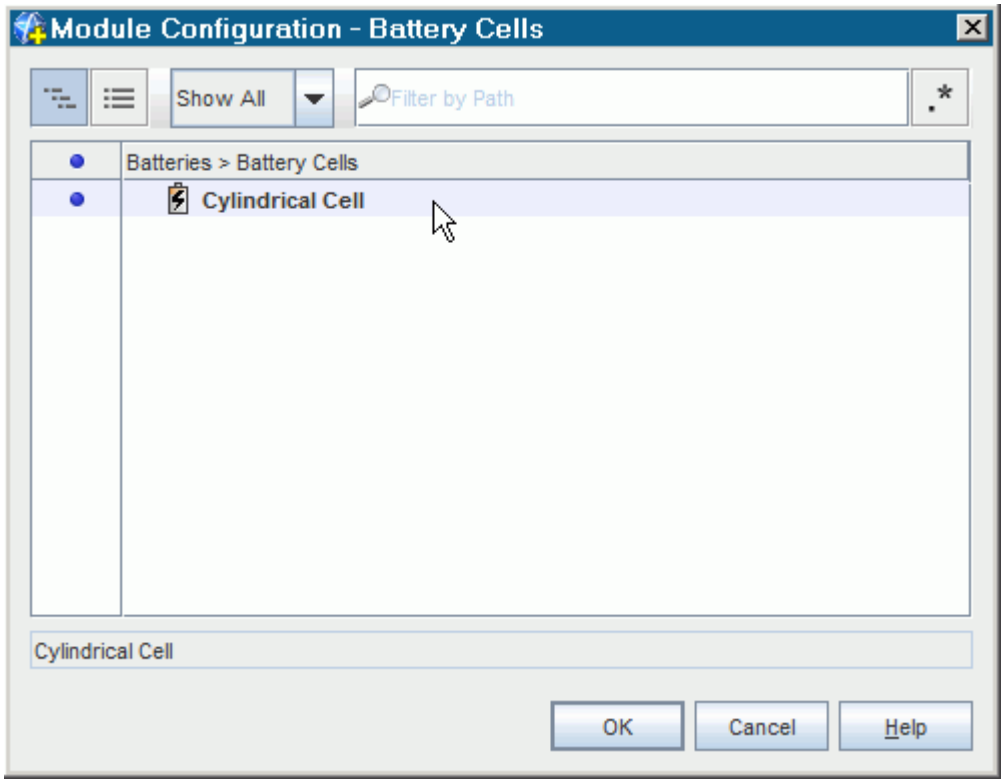
Define the configuration of the cells within the battery module.

Configure the cylindrical battery cells using a staggered configuration with uniform spacing.

To define the cell configuration:

Create a battery module:

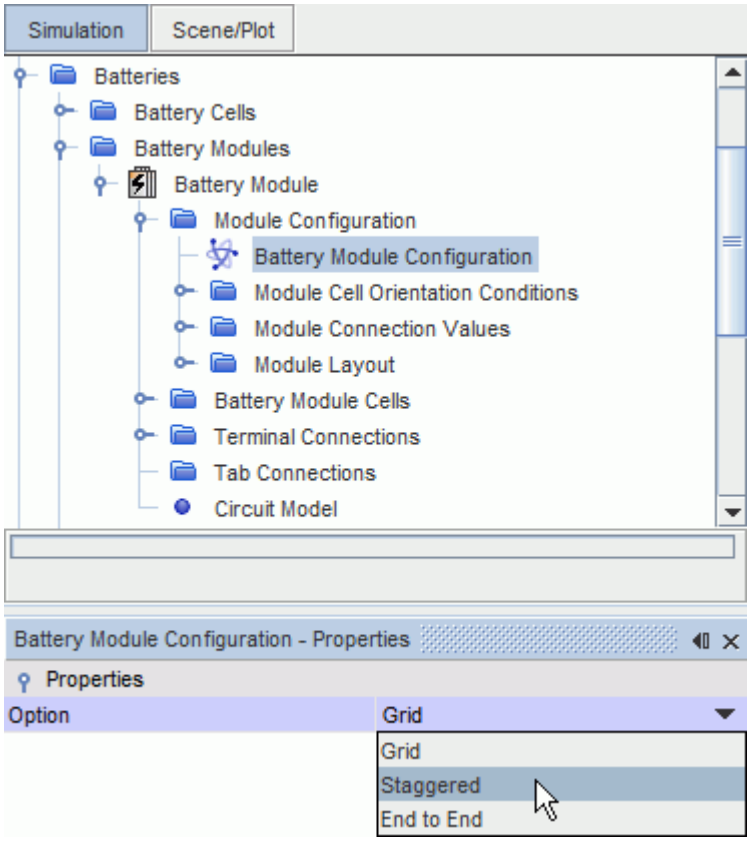
1. Right-click the **Batteries** > **Battery Modules** node and select **New**.
2. Select the **Batteries** > **Battery Modules** > **Battery Module** > **Module Configuration** node.
3. In the *Module Configuration - Properties* window to the right of **Battery Cells**, click ... (**Custom Editor**).
4. In the *Module Configuration - Battery Cells* dialog, select **Cylindrical Cell**.



5. Click **OK**.

Once you specify the battery cell to use for module configuration, STAR-CCM+ adds sub-nodes to the **Module Configuration** manager. These sub-nodes allow you to complete the setup.

6. Select the **Module Configuration** > **Battery Module Configuration** node.
7. In the *Battery Module Configuration - Properties* window, set *Option* to **Staggered**.

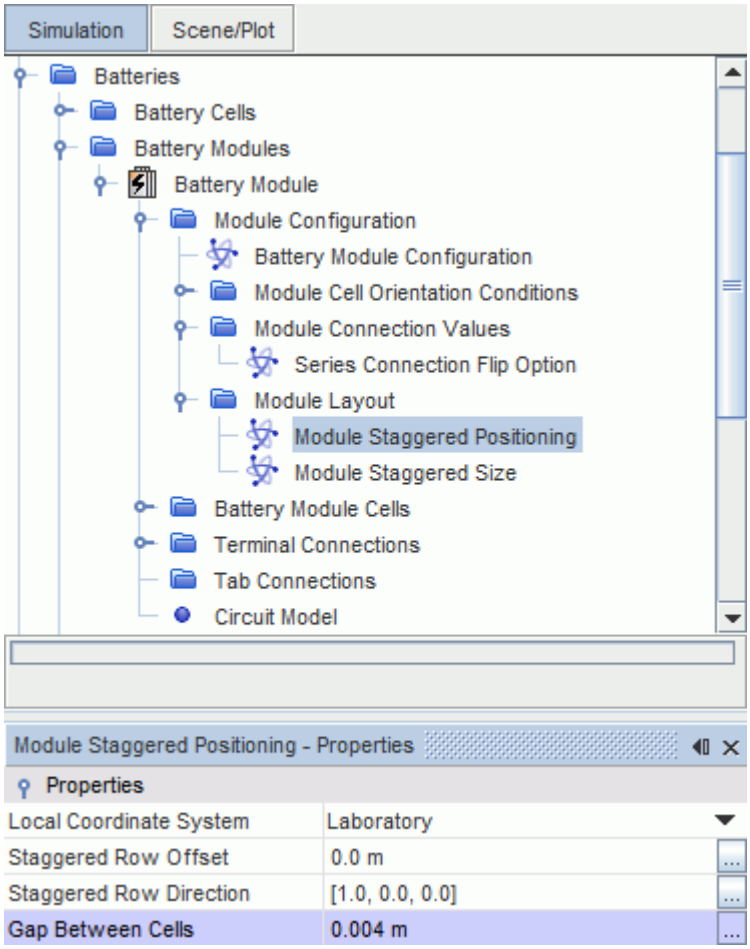


By default, the cell orientation is set to **Flip Every Other**, with the even cells flipped. Set the **Series Connection Flip Option**:

8. Select the **Module Configuration** > **Module Connection Values** > **Series Connection Flip Option** node.
9. In the *Series Connection Flip Option - Properties* window, make sure that *Option* is set to **Flip Starting With Second Cell**.

To set the number of cells in the module and the spacing between cells:

10. Select the **Module Configuration** > **Module Layout** > **Module Staggered Positioning** node.
11. In the *Module Staggered Positioning - Properties* window, set *Gap Between Cells* to `0.004 m`.



12. Select the **Module Configuration** > **Module Layout** > **Module Staggered Size** node and set *Number of Cells in Staggered Row* to `3`.

Setting Up Tab Connectors

Import and configure the tab connectors.

A tab connector has been prepared for this staggered case.

To import the tab connector into the simulation:

1. Select **File** > **Import** > **Import Surface Mesh** from the menu bar.
2. In the *Open* dialog, multi-select the `post1Cylindrical.x_b`, `post2Cylindrical.x_b`, and `strapCylindrical.x_b` files, and click **Open**.
3. In the *Import Surface Options* dialog, do the following:
   a) Make sure that the **Import Mode** is set to **Create New Part**.
   b) Deactivate **Open Geometry Scene After Import** under **Import Options**.
4. Click **OK**.

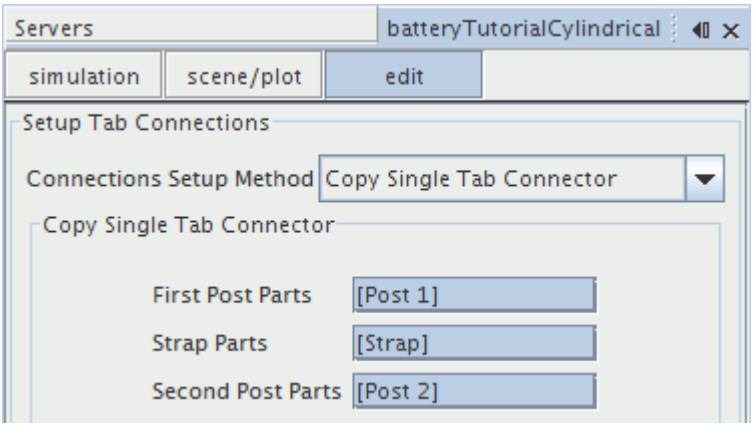
The imported tab connector includes: a strap, a positive post, and a negative post.

5. Right-click the **Batteries** > **Battery Modules** > **Battery Module** > **Tab Connections** node and select **Setup Tab Connections**.

   The *Setup Tab Connections Edit* tab appears.

The first post part is the negative post that connects the negative tab of the first battery cell to the positive tab of the second cell. The positive tab in the first battery cell always stays open. The tab connector is copied to connect the tabs of the second and third cell.

6. In the *Setup Tab Connections Edit* tab, click to the right of the *First Post Parts* property.
7. In the *Set Connection Geometry Parts* dialog, select **Post 1**.
8. Click **OK**.
9. Repeat the previous steps and assign the **Strap** part to **Strap Parts**, and the **Post 2** part to **Second Post Parts**.



10. Click **Apply** then **Close**.
11. Save the simulation.

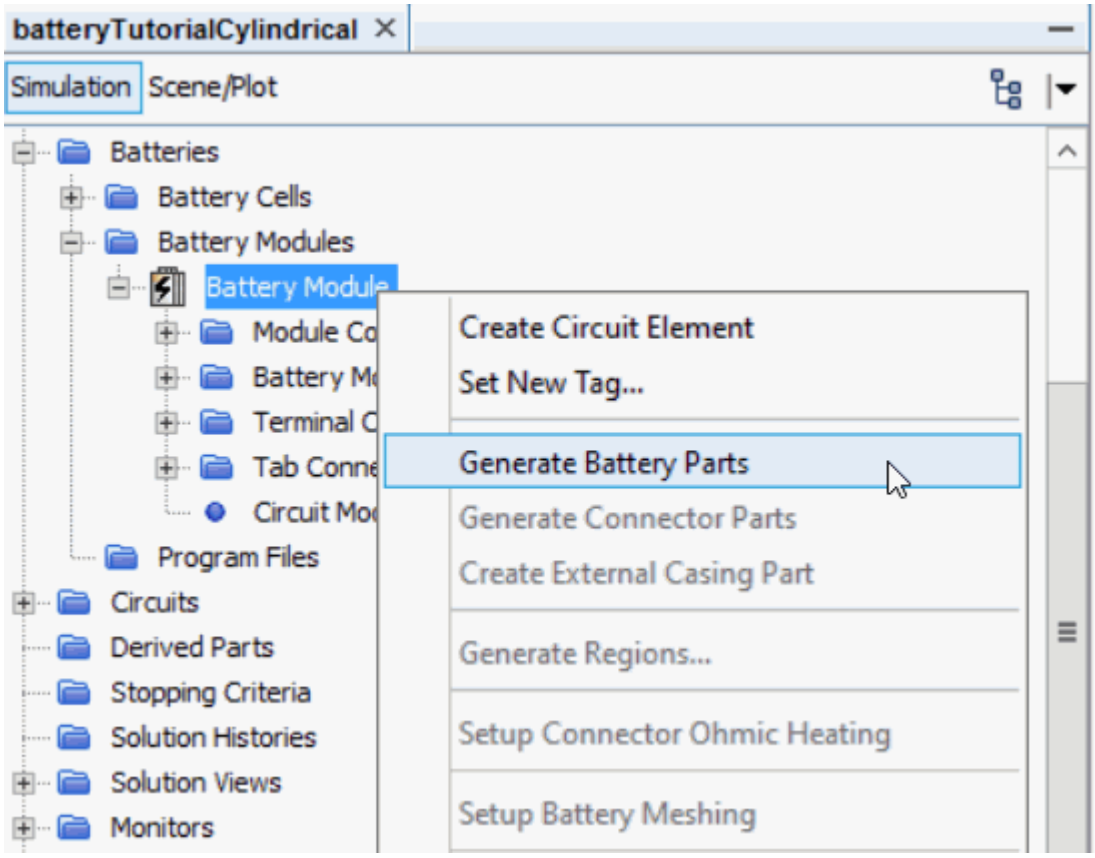
Generating the Battery Module

Generate the battery module parts.

Generate the battery module parts that you have configured in the previous steps.

To generate the battery module:

1. Right-click the **Batteries** > **Battery Modules** > **Battery Module** node and select **Generate Battery Parts**.

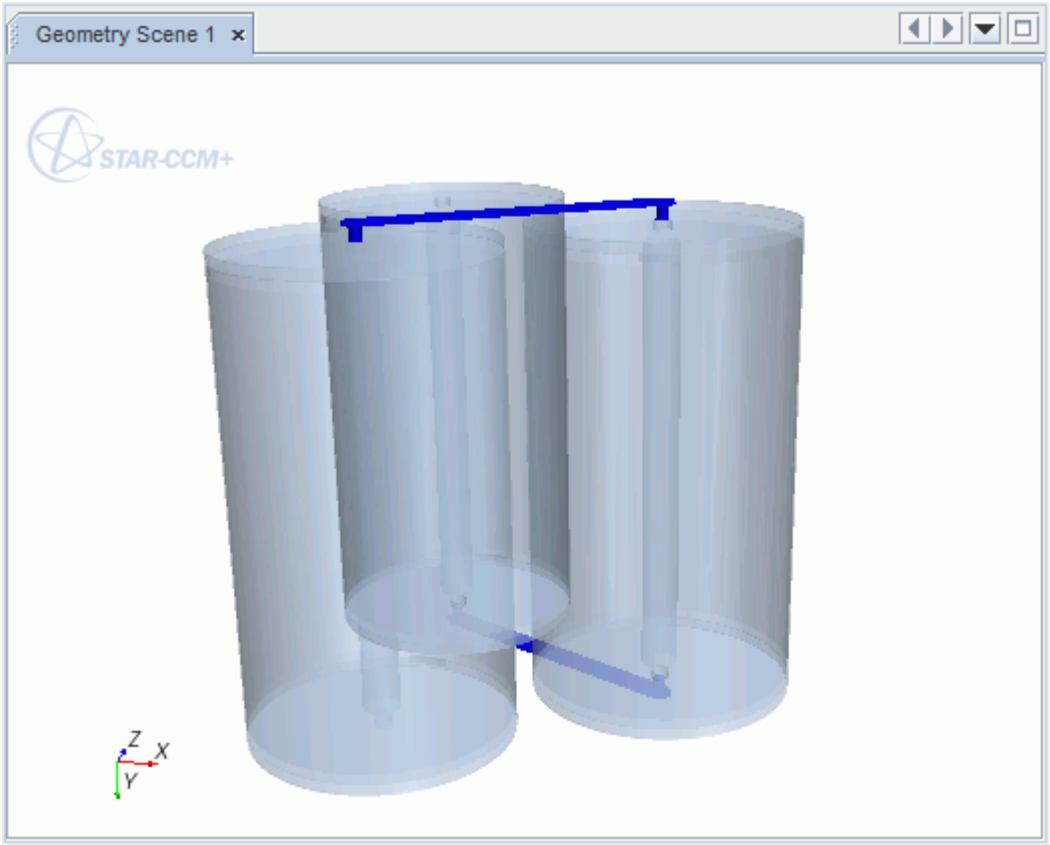


The following messages are printed in the *Output* window:

```
Generating Battery Module Cell Parts 0,0
Generating Battery Module Cell Parts 0,1
Generating Battery Module Cell Parts 0,2
```

2. Right-click the **Battery Modules** > **Battery Module** node and select **Generate Connector Parts**.
3. In the **Vis** tool bar, click the **Create / Open Scenes** button and select **Geometry**.

STAR-CCM+ creates two tab connectors on the opposite sides of the cells, as shown in the example below.



Imprinting the Tab Connectors

To ensure that all the interfaces are properly formed when you generate the battery module regions, perform an imprint operation between the battery cells and the tab connectors.

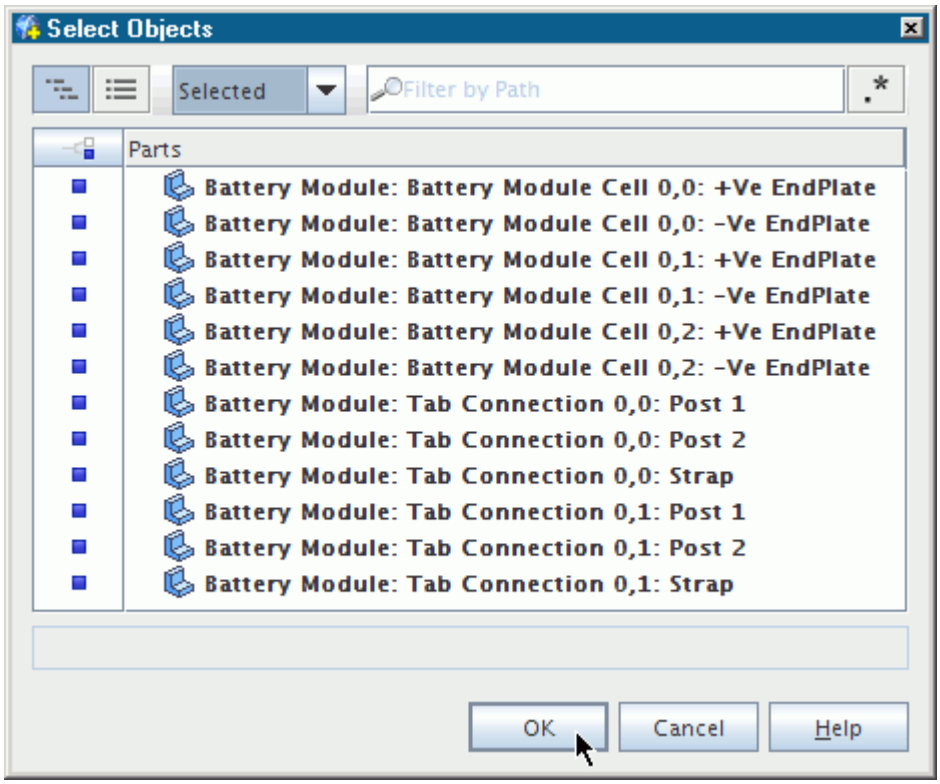
To imprint the tab connectors and the battery cells:

1. Click (**Launch Surface Repair**).
2. In the *Surface Repair Options* dialog next to **Geometry Parts**, click **Select**.

   The *Select Objects* dialog appears.
3. Click (**Column Selection**) until all parts are deselected.
4. Select the following:
   - **Battery Module: Battery Module Cell 0,0: +Ve EndPlate**
   - **Battery Module: Battery Module Cell 0,0: -Ve EndPlate**
   - **Battery Module: Battery Module Cell 0,1: +Ve EndPlate**
   - **Battery Module: Battery Module Cell 0,1: -Ve EndPlate**
   - **Battery Module: Battery Module Cell 0,2: +Ve EndPlate**
   - **Battery Module: Battery Module Cell 0,2: -Ve EndPlate**
   - **Battery Module: Tab Connection 0,0: Post 1**

- **Battery Module: Tab Connection 0,0: Post 2**
- **Battery Module: Tab Connection 0,1: Post 1**
- **Battery Module: Tab Connection 0,1: Post 2**
- **Battery Module: Tab Connection 0,0: Strap**
- **Battery Module: Tab Connection 0,1: Strap**

5. Click the **Show All** drop down menu and select **Selected**. Ensure your selection appears as shown below.

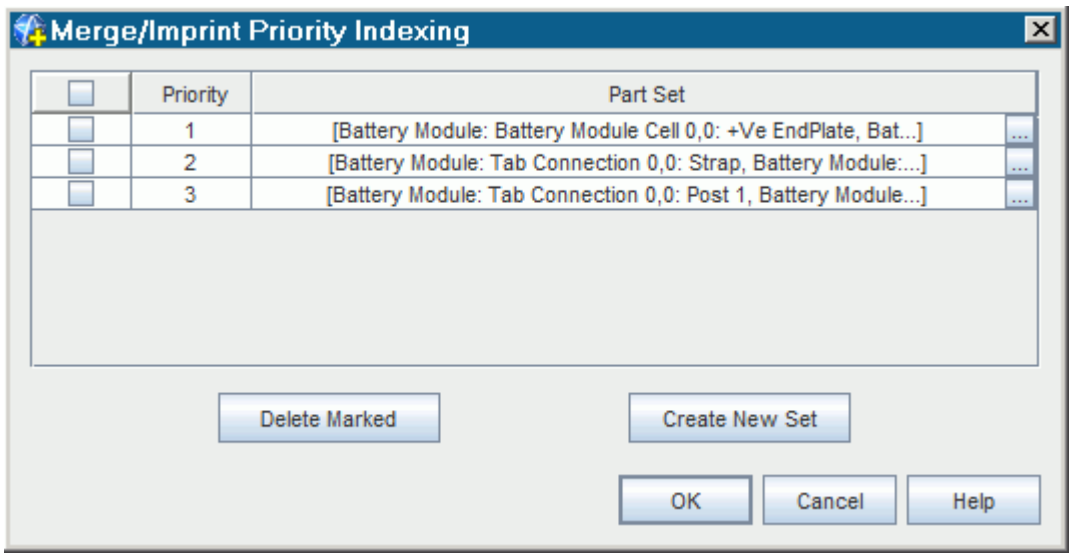


6. Click **OK**.
7. Click **OK** in the *Surface Repair Options* dialog.
8. In the surface repair *Edit* panel, select the **Global** tab and click (**Merge/Imprint single or multiple parts**).
9. Set the **Imprint Mode** to **Multi-Part Imprint**.
10. Set the **Tolerance** to `1.0E-6 m`.
11. Click **Priority Indexing**.

    The *Merge/Imprint Priority Indexing* dialog appears.
12. Click **Create New Set**.
13. Click (**Custom Editor**) for Priority 1.
14. In the *Select Parts* dialog, click **Column Selection** to select all parts.
15. Deselect the following:
    - **Post 1**

- **Post 2**
- **Strap**

16. Click **OK**.
17. Click **Create New Set**.
18. Click ... (**Custom Editor**) for Priority 2.
19. In the *Select Parts* dialog, select the following:
    - **Battery Module: Tab Connection 0,0: Strap**
    - **Battery Module: Tab Connection 0,1: Strap**
20. Click **OK**.
21. Click **Create New Set**.
22. Click ... (**Custom Editor**) for Priority 3.
23. In the *Select Parts* dialog, select the following:
    - **Battery Module: Tab Connection 0,0: Post 1**
    - **Battery Module: Tab Connection 0,0: Post 2**
    - **Battery Module: Tab Connection 0,1: Post 1**
    - **Battery Module: Tab Connection 0,1: Post 2**
24. Click **OK**.



25. To exit the *Merge/Imprint Priority Indexing* dialog, click **OK**.
26. In the **Part Browser** box, click **Find Pairs**.
27. When all the pairs are found, click **Imprint All**.
28. When all the pairs are imprinted, **Close** the *Edit* panel.
29. Save the simulation.

Defining the External Casing

Define the external casing for the battery module.

A simple external casing part is provided with your STAR-CCM+ installation.

To define the external casing:

Import the geometry into the simulation.

1. Select **File** > **Import** > **Import Surface Mesh...** from the menu bar.
2. In the *Open* dialog, select the `externalCasingCylindrical.x_b` file, and click **Open**.
3. In the *Import Surface Options* dialog, do the following:
   a) Make sure that the **Import Mode** is set to **Create New Part**.
   b) Set **Mark Feature Edges** to **All CAD Edges**.
   c) Deactivate **Open Geometry Scene After Import** under **Import Options**.
4. Click **OK**.

Create an inlet and outlet for the external casing.

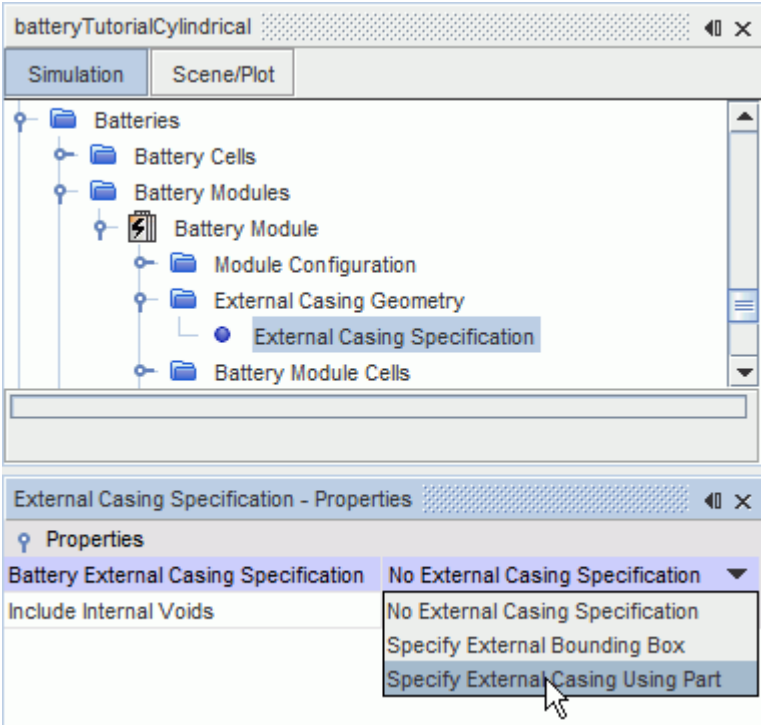
5. Right-click the **Geometry** > **Parts** > **Case** > **Surfaces** > **Faces** node and select **Split By Patch...**.

   Note: You must have a scene open to use this feature.

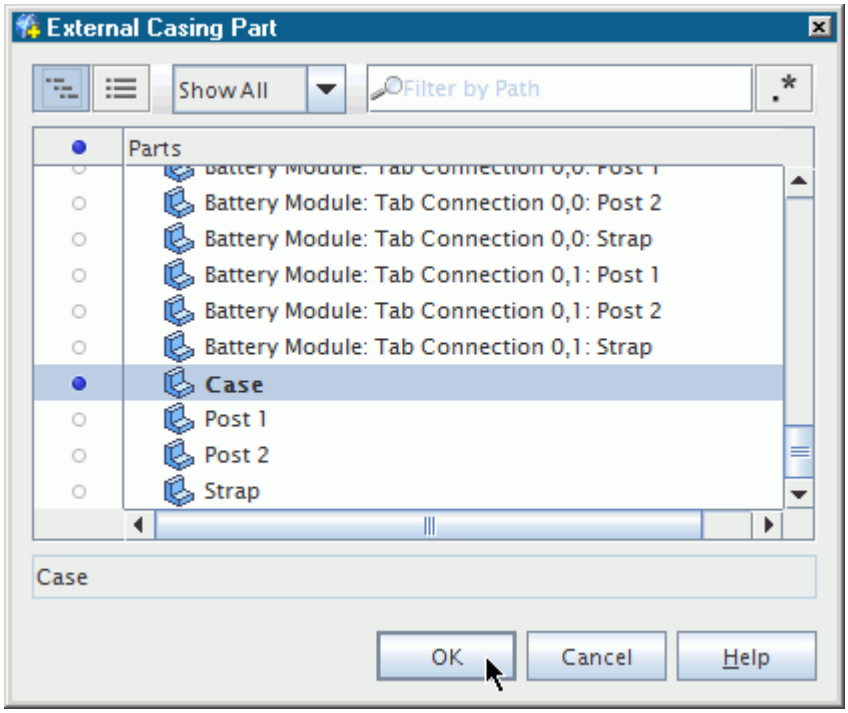
   The *Split Part Surface by Patch* panel appears.
6. In the **Part Surface Patches** box, select `3190`.
7. Set the **Part Surface Name** to `Outlet`.
8. Click **Create**.
9. Similarly, select `3188` and rename this part `Inlet`.
10. Click **Create**.
11. Click **Close**.

Select the external casing part.

12. Select the **Batteries** > **Battery Modules** > **Battery Module** > **External Casing Geometry** > **External Casing Specification** node.
13. In the *External Casing Specification - Properties* window, set *Battery External Casing Specification* to **Specify External Casing Using Part**.



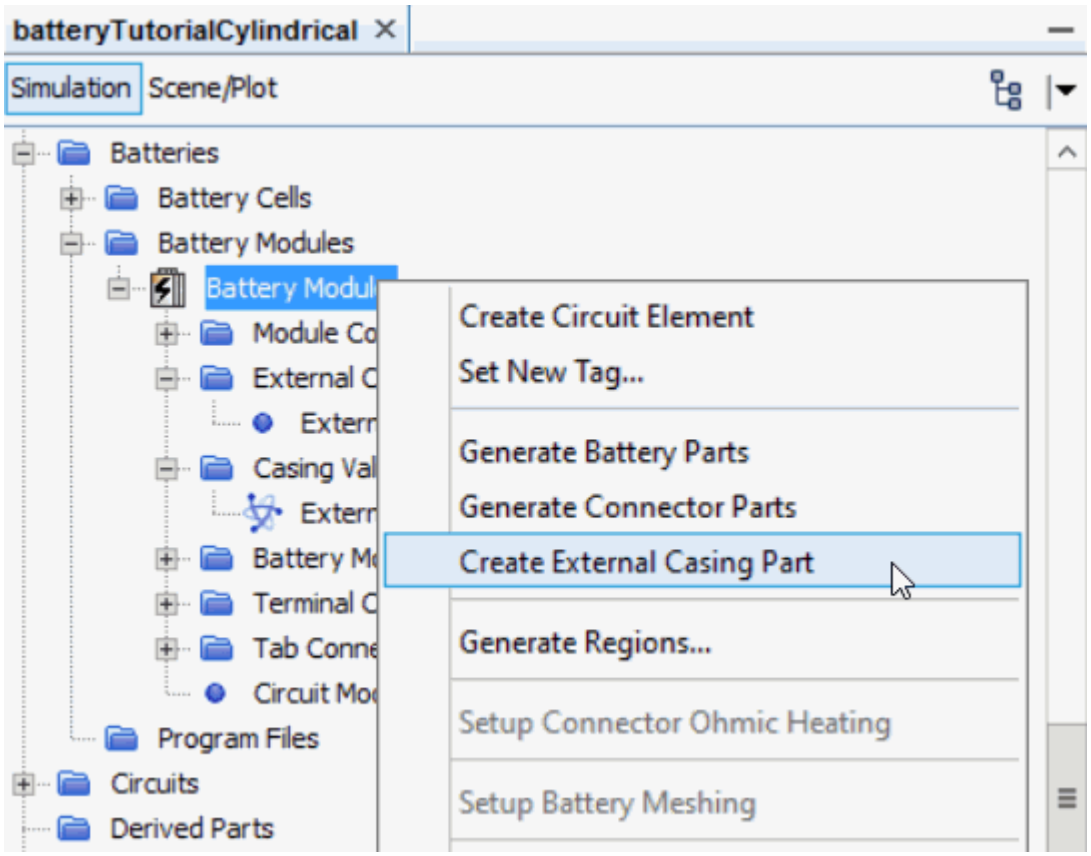
14. Select the **Battery Module** > **Casing Values** > **External Casing Part** node.
15. In the *External Casing Part - Properties* window, in the **External Casing Part** property, click ... (**Custom Editor**).
16. In the *External Casing Part* dialog, select **Case**.



17. Click **OK**.

You can now generate the external casing part.

18. Right-click the **Battery Modules** > **Battery Module** node and select **Create External Casing Part**.



STAR-CCM+ creates a **Battery Module: External Casing** part. The other battery module parts have been subtracted from it.

19. Save the simulation.

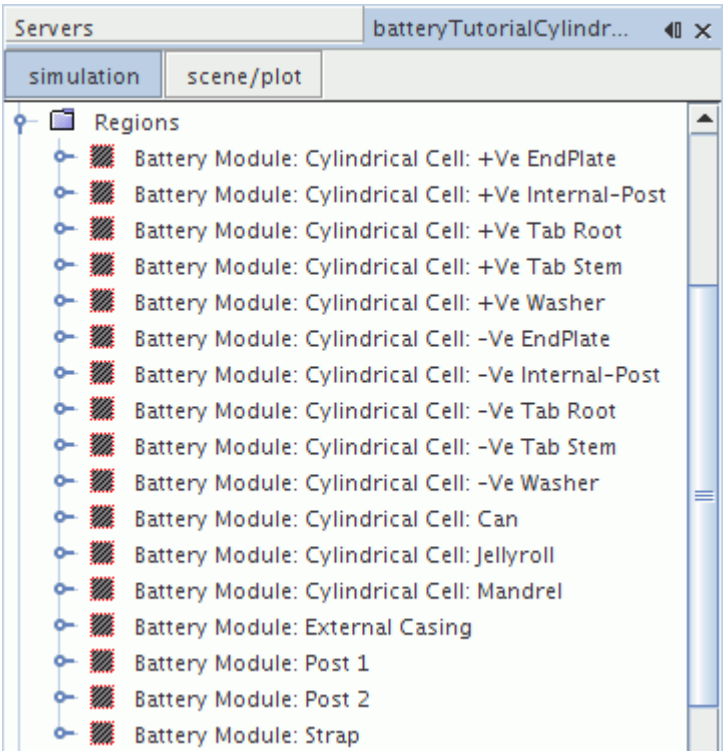
Generating Regions

Create the regions of the battery module.

To create the regions of the battery module:

1. Right-click the **Battery Modules** > **Battery Module** node and select **Generate Regions...**.
2. In the *Additional (non-Battery) Parts to Generate Regions* dialog, click (**Column Selection**) twice to deselect all parts.
3. Click **OK**.

   STAR-CCM+ automatically creates regions and interfaces for the battery module.



Generating the Volume Mesh

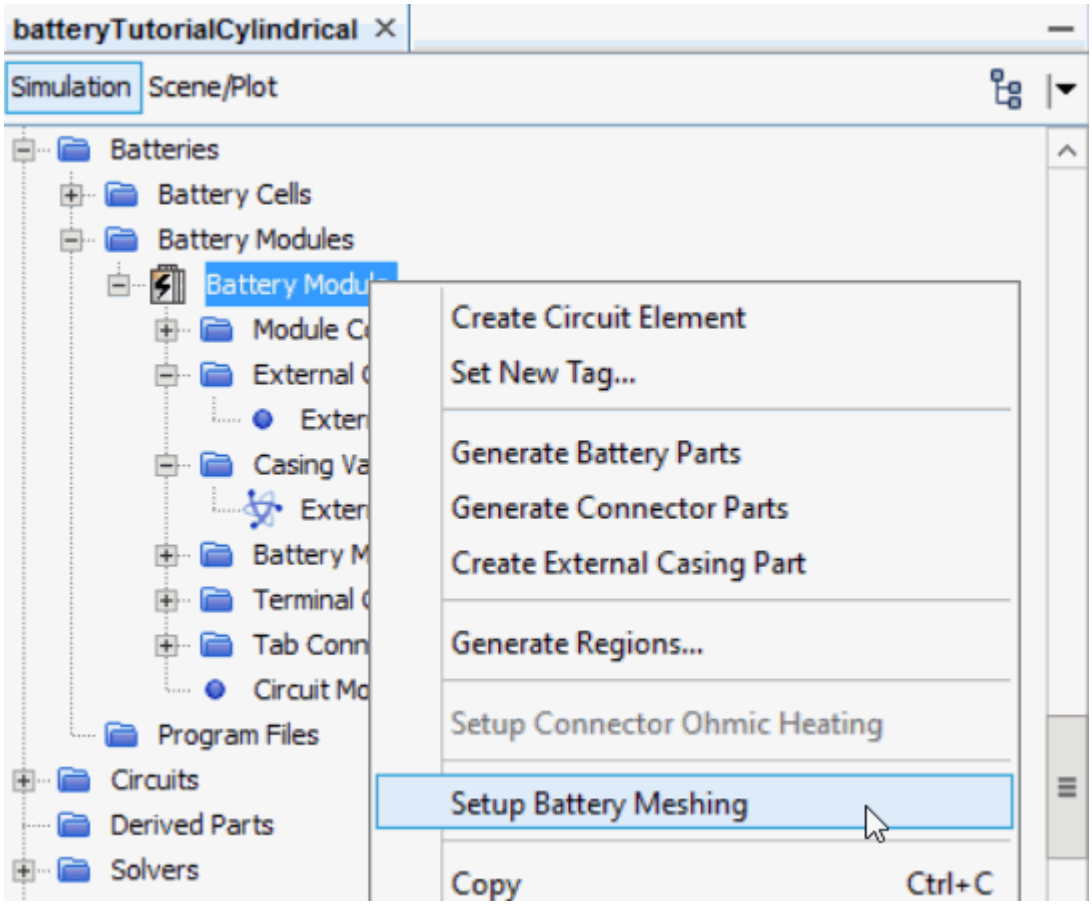
Create the volume mesh continua for the battery module.

Create two separate mesh continua—one for the battery cells and the tab connectors, then one for the external casing.

To generate the volume mesh:

Set up the battery mesh continuum.

1. Right-click the **Batteries** > **Battery Modules** > **Battery Module** node.
2. Select **Setup Battery Meshing**.



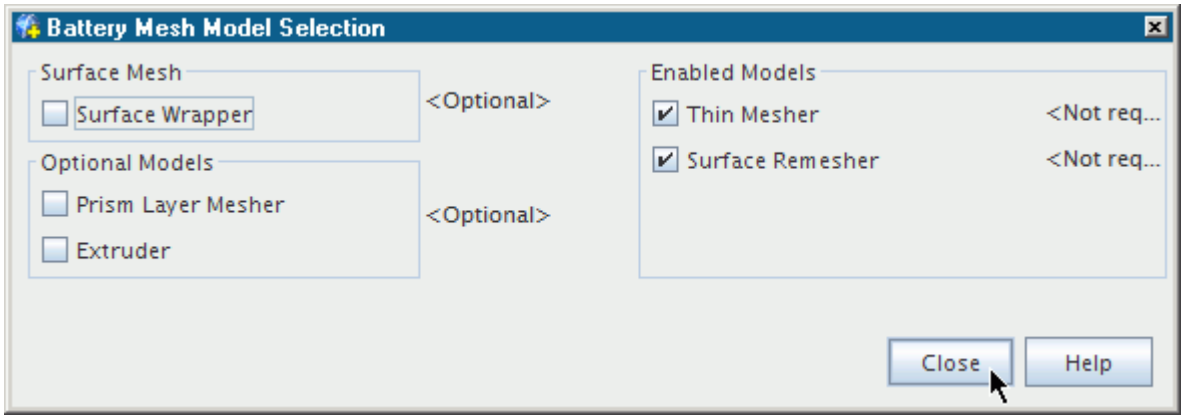
A new **Mesh 1** node appears under the **Continua** node. This new battery mesh continuum already contains the **Surface Remesher** and **Thin Mesher** models. The Battery Mesh Continuum also contains a volumetric control for the strap and post parts.

3. Rename the new **Mesh 1** node to `Battery Mesh`.

To confirm the **Battery Mesh** continuum contains the correct meshing models:

4. Right-click the **Continua** > **Battery Mesh** > **Models** node and select **Select Meshing Models**.

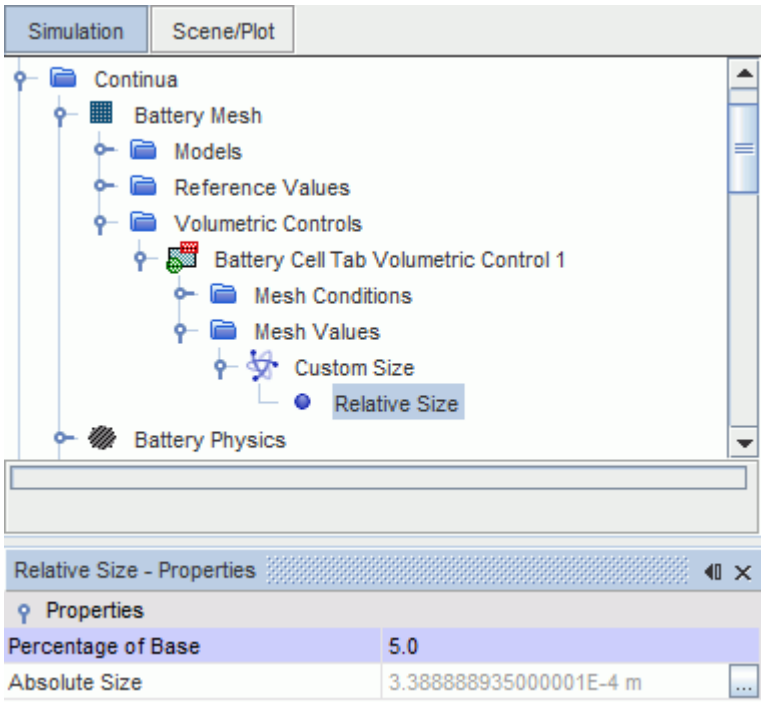
   The *Battery Mesh Model Selection* dialog appears as shown below.



5. Ensure the **Surface Remesher** and **Thin Mesher** models are selected.
6. Click **Close**.

To refine the volume mesh for the tab connectors:

7. Expand the **Battery Mesh** > **Volumetric Controls** > **Battery Cell Tab Volumetric Control 1** node.
8. Select the **Mesh Values** > **Custom Size** > **Relative Size** node.
9. In the *Properties* window, set *Percentage of Base* to `5.0`.

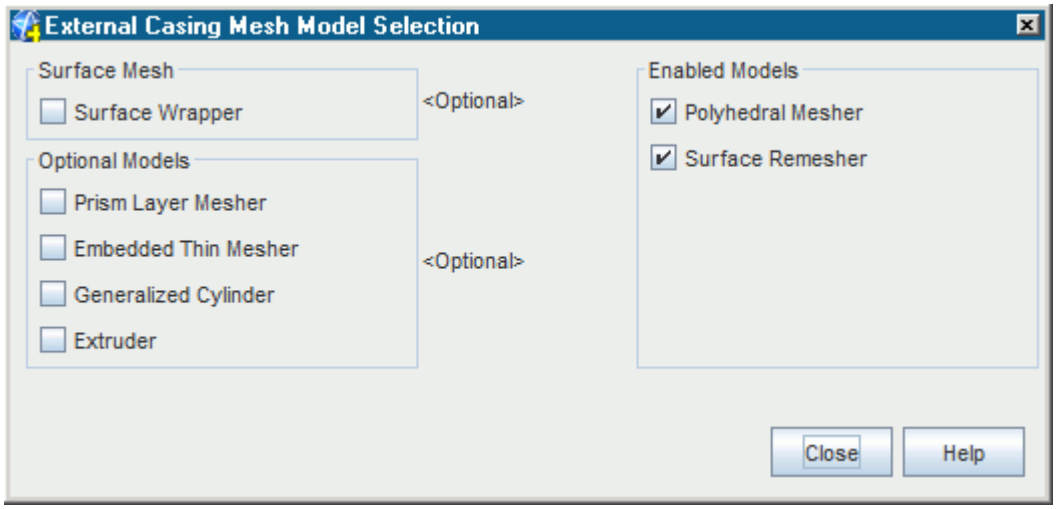


To create a second mesh continuum for the external casing:

10. Right-click the **Continua** node and select **New** > **Mesh Continuum**.
11. Rename the new **Mesh 1** node to `External Casing Mesh`.

To select the models for the external casing mesh:

12. Right-click the **Continua** > **External Casing Mesh** node and select **Select Meshing Models**.
13. In the *External Casing Mesh Model Selection* dialog, select the following:
    - **Surface Remesher** from the *Surface Mesh* group box.
    - **Polyhedral Mesher** from the *Volume Mesh* group box.



14. Click **Close**.
15. Select the **Continua** > **External Casing Mesh** > **Reference Values** > **Base Size** node and set *Value* to `0.005 m`.

To assign the external casing mesh continuum to the external casing region:

16. Select the **Regions** > **Battery Module: External Casing** node.
17. In the *Battery Module: External Casing - Properties* window, set the *Mesh Continuum* to **External Casing Mesh**.

To build the volume mesh:

18. Click (**Generate Volume Mesh**) in the tool bar.

Visualize the mesh.

19. In the **Vis** tool bar, click the **Create / Open Scenes** button and select **Mesh**.
20. Save the simulation.

Selecting the Physics Models for the External Casing

Physics models define the primary variables of the simulation and what mathematical formulation is used to generate the solution.

The external casing is modeled as an air continuum.

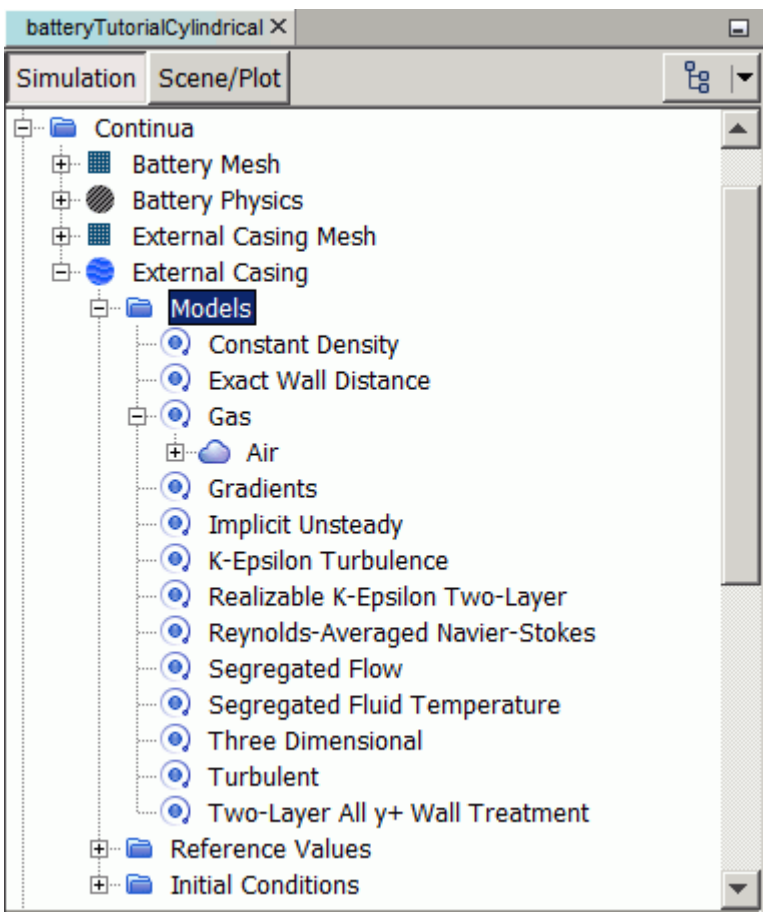
To select the physics models for the external casing:

1. Create a physics continuum and rename the **Continua** > **Physics 1** node to **External Casing**.
2. For the physics continuum, **Continua** > **External Casing**, select the following models in order:

Group Box	Model
Space	**Three-Dimensional**
Time	**Implicit Unsteady**

Group Box	Model
Material	**Gas**
Flow	**Segregated Flow** **Gradients** (selected automatically)
Equation of State	**Constant Density**
Viscous Regime	**Turbulent** **Reynolds-Averaged Navier-Stokes** (selected automatically)
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**
	Realizable K-Epsilon Two-Layer (Selected automatically) **Exact Wall Distance** (Selected automatically)
	Two-Layer All y+ Wall Treatment (Selected automatically)
Optional Models	**Segregated Fluid Temperature**

3. Click **Close**.
4. To review the models, open the **External Casing** > **Models** node.



5. Select the **Regions** > **Battery Module: External Casing** node and set **Physics Continuum** to **External Casing**.
6. Select the **Regions** > **Battery Module: External Casing** > **Boundaries** node and set the following properties:

Node	Property	Setting
Battery Module: External Casing.Case.Inlet	*Type*	**Velocity Inlet**
Battery Module: External Casing.Case.Outlet	*Type*	**Pressure Outlet**

7. Save the simulation.

Creating a Circuit

Connect the electrical load to the battery module to create a circuit.

A program file containing the electrical load on the battery is provided.

To create a circuit:

1. Right-click the **Batteries** > **Program Files** node and select **Load...**.
2. In the *Open* dialog, select the `Tutorial3CDischargeCylindrical.prg` file.
3. Click **Open**.

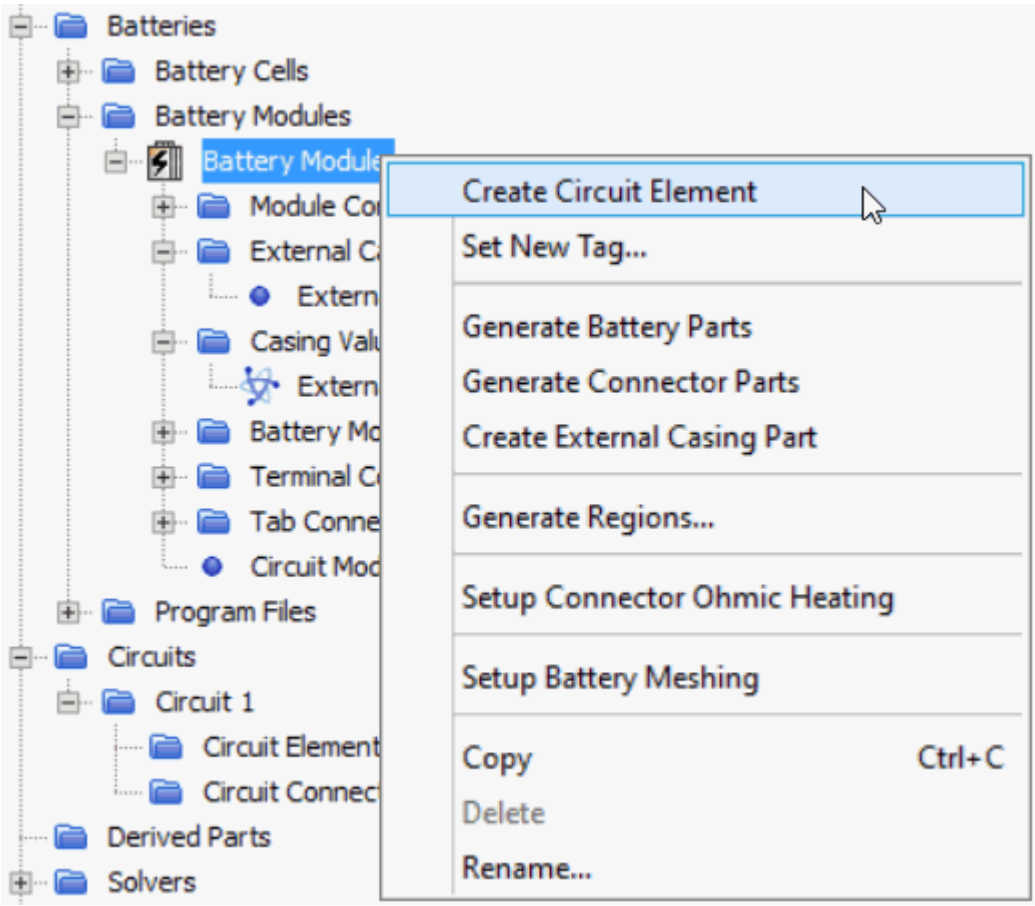
The program file that you loaded in the previous step represents an electrical load on the battery. To simulate the response of the battery, the electrical load must be connected to the battery in a circuit. Connect the electrical load using the Circuit Solver tool.

4. Expand the **Circuits** > **Circuit 1** > **Circuit Elements** node.

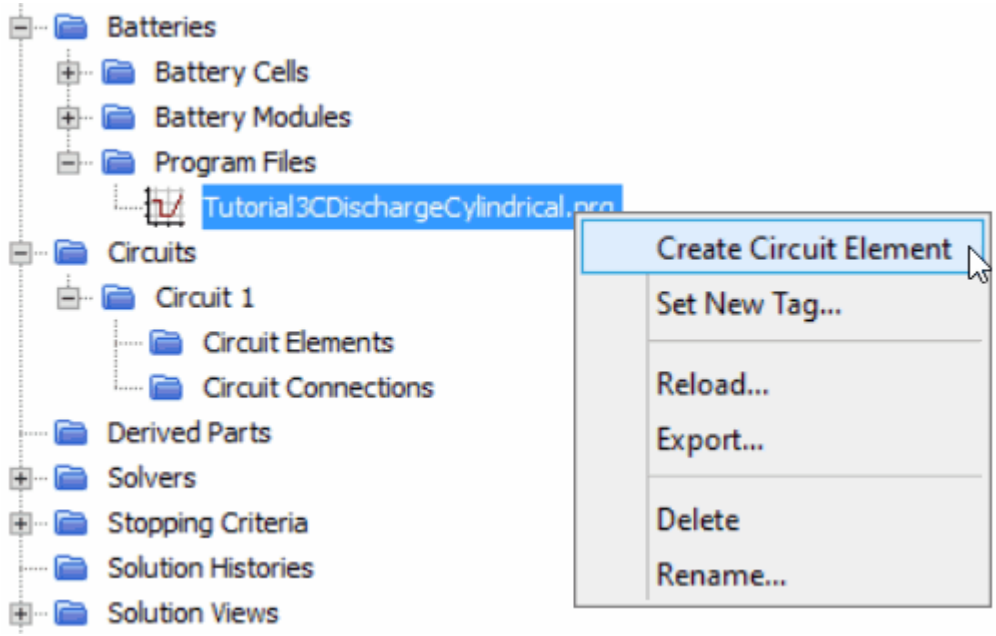
The Circuit Solver uses Elements and Connections. Each element is either a battery module or a program, and a connection joins elements together in the circuit.

First, you define the battery module and program as elements of the circuit.

5. Right-click the **Battery Modules** > **Battery Module** node and select **Create Circuit Element**.

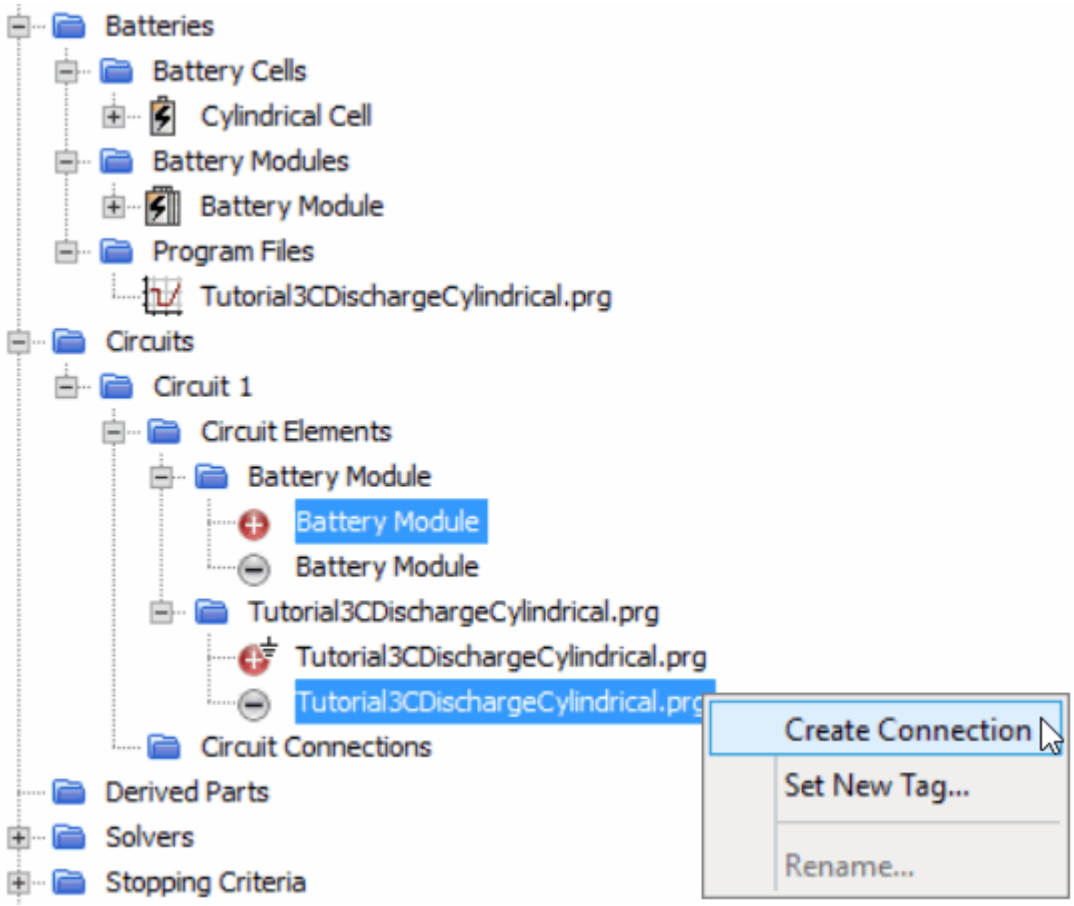


6. Right-click the **Program Files** > **Tutorial3CDischargeCylindrical.prg** node and select **Create Circuit Element**.



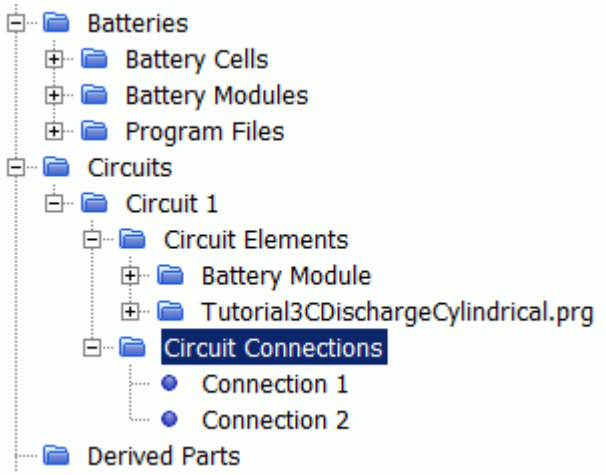
Next, the positive and negative terminals of the elements are connected.

7. Expand the **Circuits** > **Circuit 1** > **Circuit Elements** node.
8. Multi-select the **Battery Module** > **(+) Battery Module** node and the **Tutorial3CDischargeCylindrical.prg** > **(-) Tutorial3CDischarge.prg** node.
9. Right-click one of the nodes and select **Create Connection**.



10. Repeat for the remaining terminals: **(-) Battery Module** and **(+) Tutorial3CDischargeCylindrical.prg**.

    Each connection is shown under the **Circuit Connections** node.



To model the effect of ohmic heating, activate the ohmic heating models.

11. Right-click **Batteries** > **Battery Modules** > **Battery Module** and then select **Setup Connector Ohmic Heating**.

    The **Continua** > **Battery Module Connectors: Ohmic Heating** physics continuum node is added with the appropriate models activated.

12. Save the simulation.

Setting Solver Parameters and Stopping Criteria

Set the appropriate solver parameters and stopping criteria for the simulation.

Run the simulation for 200 seconds, after which the state of charge of the battery is around 80%.

To set the solver parameters and stopping criteria:

1. Edit the **Solvers** node and set the following properties:

Node	Property	Setting
Implicit Unsteady	**Time-Step**	`1.0 s`

2. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Physical Time	*Enabled*	Activated
	Maximum Physical Time	`200 s`
Maximum Inner Iterations	*Enabled*	Activated
	Maximum Inner Iterations	`5`
Maximum Steps	*Enabled*	Deactivated

3. Save the simulation.

Reporting, Monitoring, and Plotting

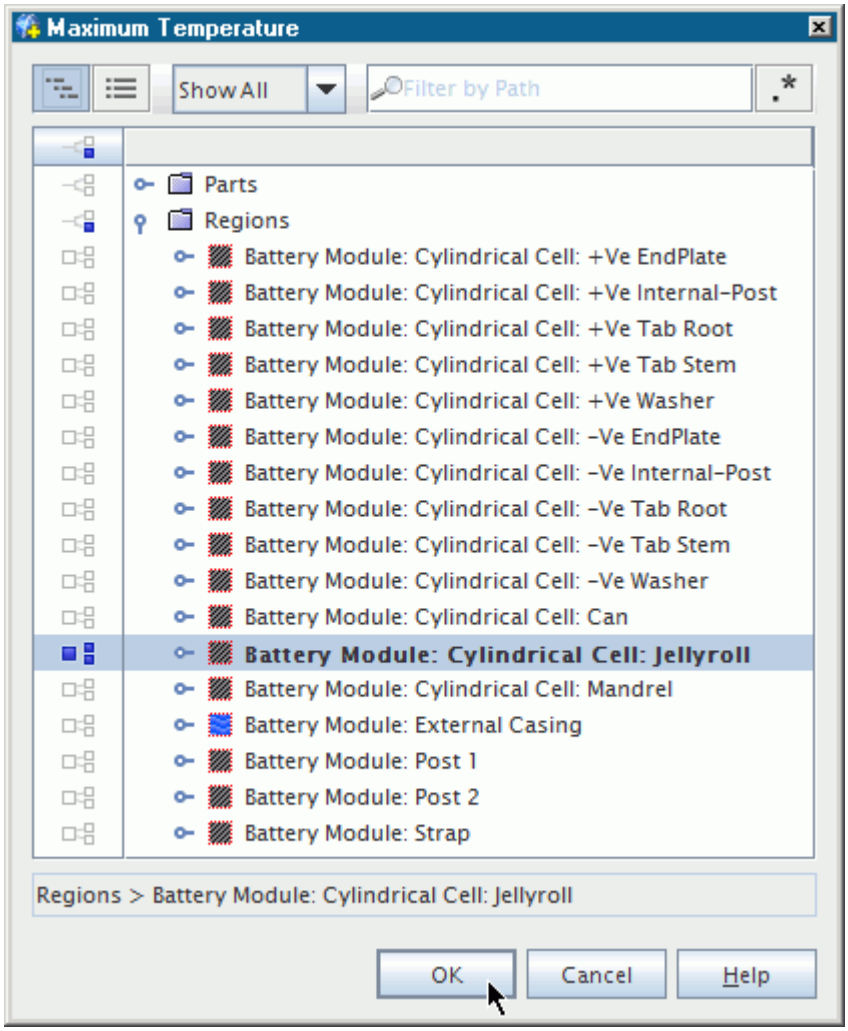
Create reports, monitors, and plots, to visualize the results of the simulation.

Create maximum and minimum temperature reports, monitors, and plots to visualize the temperature rise of the cells.

To create reports, monitors, and plots:

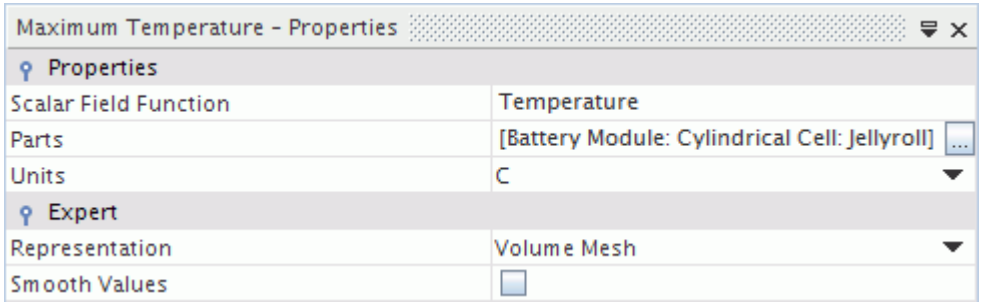
Start with the maximum temperature report.

1. Right-click the **Reports** node and select **New Report** > **Maximum**.
   A new report, **Maximum 1**, is created.
2. Rename the **Maximum 1** node to `Maximum Temperature`.
3. Select the **Reports** > **Maximum Temperature** node and set *Field Function* to **Temperature**.
4. Set *Units* to **C**.
5. Click ... (**Custom Editor**) for the *Parts* property.
   The *Maximum Temperature - Parts* dialog appears.
6. Expand the **Regions** node.
7. Click **Battery Module: Cylindrical Cell: Jellyroll** twice.



8. Click **OK**.

   The *Maximum Temperature Properties* window appears as shown:



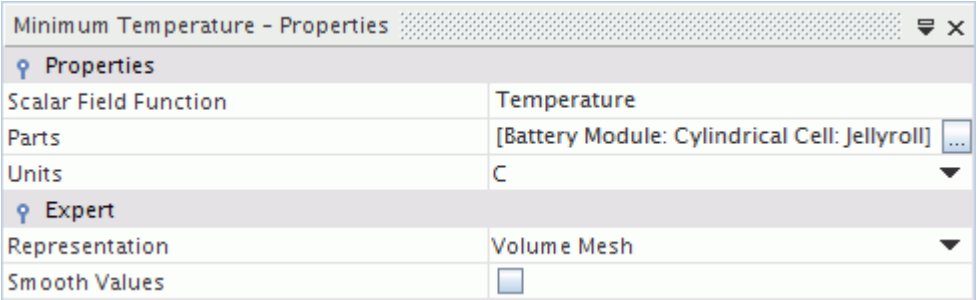
Set up the minimum temperature report.

9. Right-click the **Reports** node and select **New Report** > **Minimum**.
10. Rename the **Reports** > **Minimum 1** node to `Minimum Temperature`.
11. Select the **Reports** > **Minimum Temperature** node and set *Field Function* to **Temperature**.

12. Set *Units* to **C**.
13. Click ... (**Custom Editor**) for the *Parts* property.

    The *Minimum Temperature - Parts* dialog appears.
14. Expand the **Regions** node.
15. Click **Battery Module: Cylindrical Cell: Jellyroll** twice.
16. Click **OK**.

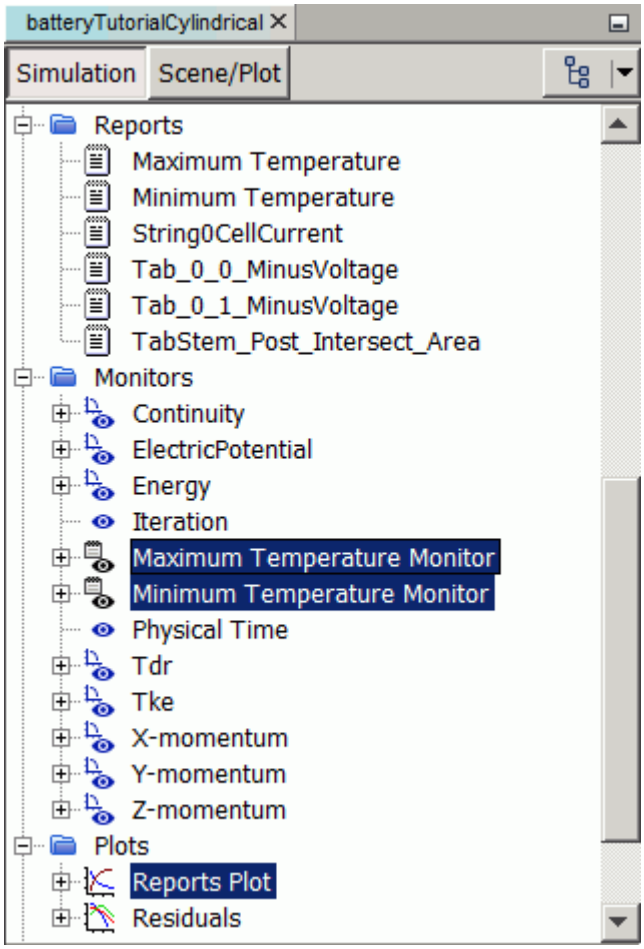
    The *Minimum Temperature Properties* window appears as shown:



Create a monitor and plot with the temperature reports:

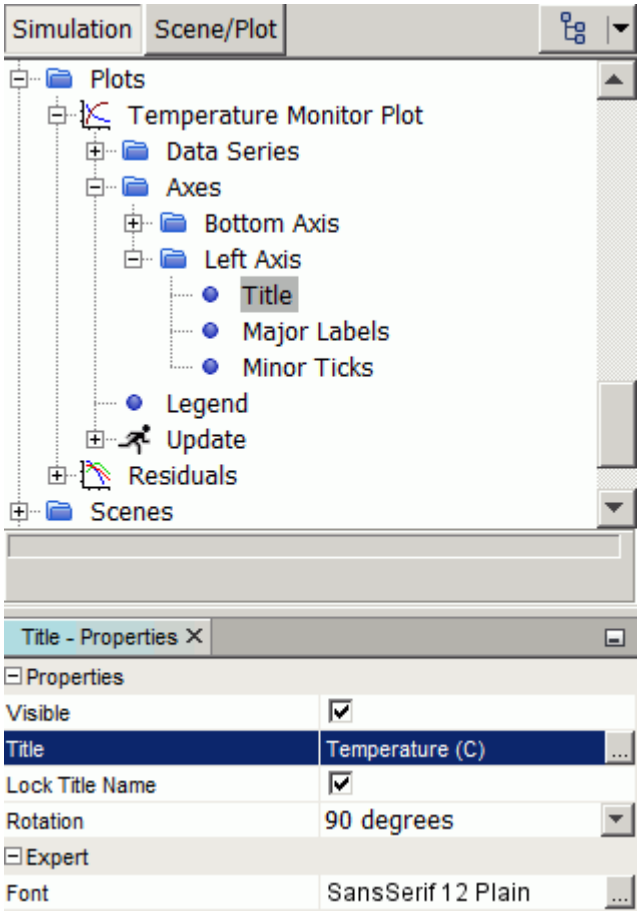
17. Multi-select the **Reports** > **Maximum Temperature** and **Reports** > **Minimum Temperature** nodes.
18. Right-click one of the selected nodes and select **Create Monitor and Plot from Report**.
19. In the *Create Plot From Reports...* dialog, click **Single Plot**.

    The **Maximum Temperature Monitor**, **Minimum Temperature Monitor**, and **Reports Plot** nodes appear in the simulation tree.



20. Rename **Plots** > **Reports Plot** to `Temperature Monitor Plot`.
21. Right-click the **Plots** > **Temperature Monitor Plot** node and select **Open**.

    The *Temperature Monitor Plot* window opens.
22. Select the **Plots** > **Temperature Monitor Plot** > **Axes** > **Left Axis** > **Title** node.
23. In the *Title - Properties* window, set the *Title* to `Temperature (C)`.



Create a state of charge report, monitor, and plot to visualize the discharging of the cells.

24. Right-click the **Reports** node and select **New Report** > **Volume Average**.

    A new report, **Volume Average 1**, is created.

25. Rename the **Volume Average 1** node to `State of Charge`.

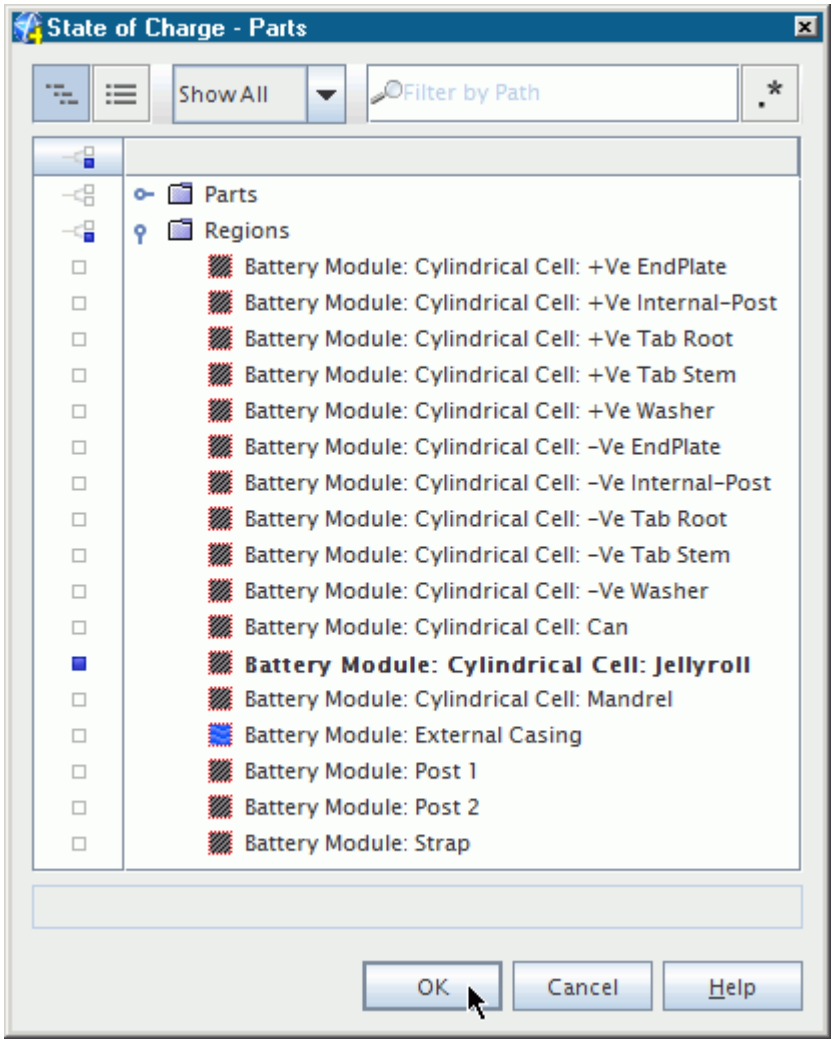
26. Select the **Reports** > **State of Charge** node and set *Field Function* to **Battery** > **Battery SOC**.

27. Click ... (**Custom Editor**) for the *Parts* property.

    The *State of Charge - Parts* dialog appears.

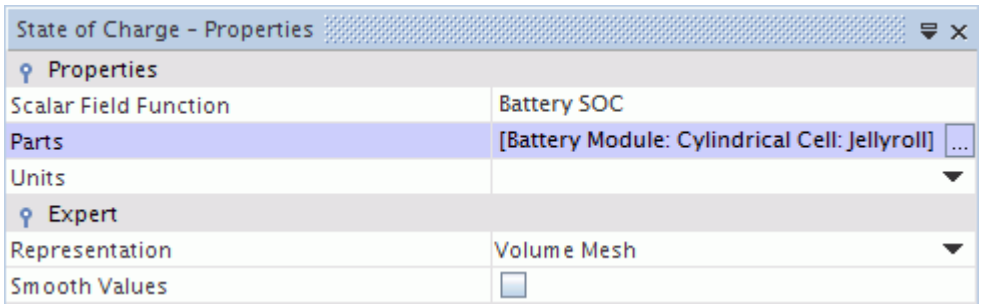
28. Expand the **Regions** node.

29. Select **Battery Module: Cylindrical Cell: Jellyroll**.



30. Click **OK**.

    The *State of Charge Properties* window appears as shown:



Create a monitor and plot with the state of charge report:

31. Right-click the **Reports** > **State of Charge** node and select **Create Monitor and Plot from Report**.
32. Right-click the **Plots** > **State of Charge Monitor Plot** node and select **Open**.
33. Save the simulation.

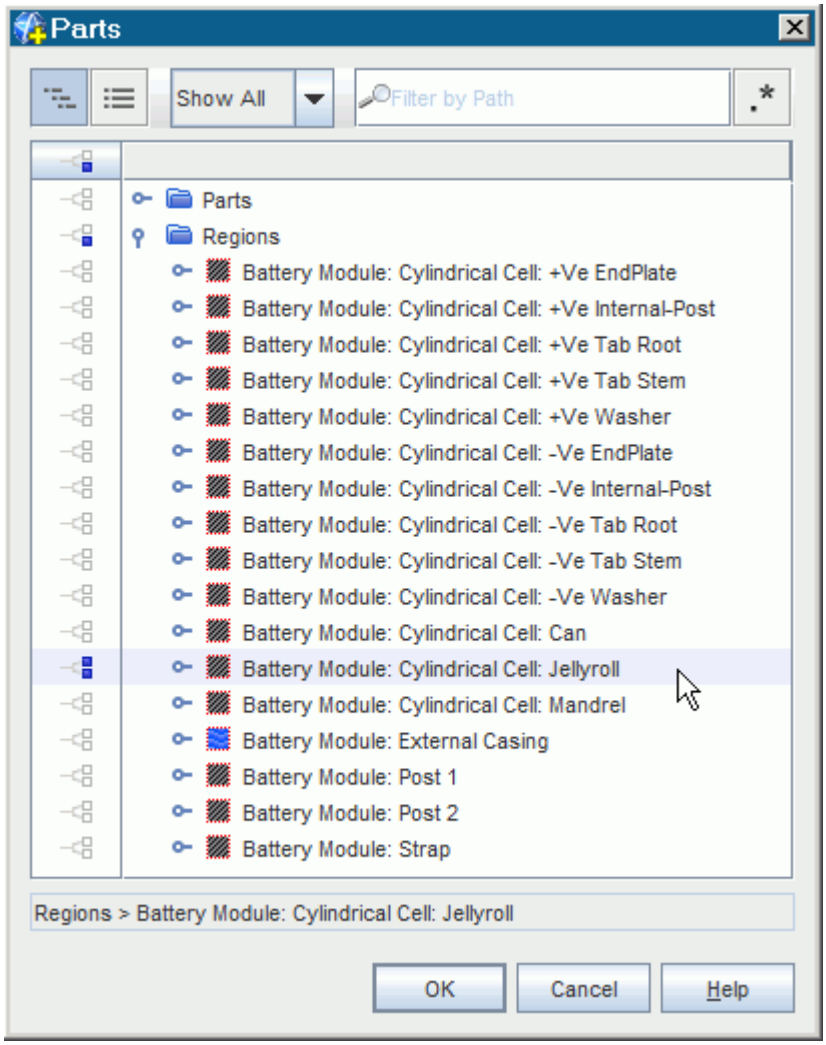
Preparing a Scalar Scene

Create a scalar scene to display the results of the simulation.

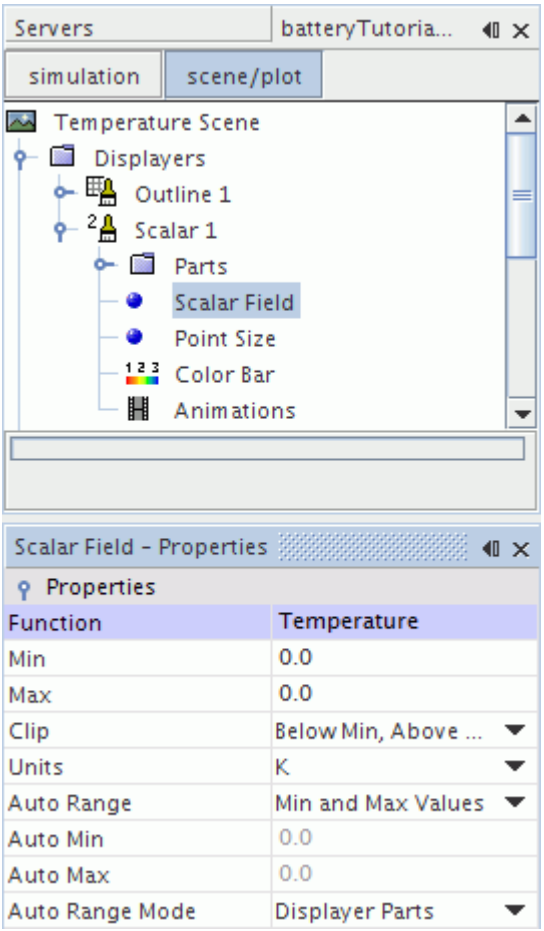
You create a scalar scene to display the temperature of the battery cans.

To prepare a scalar scene:

1. In the **Vis** tool bar, click the **Create / Open Scenes** button and select **Scalar**.
2. Rename the **Scenes** > **Scalar Scene 1** node to `Temperature Scene`.
3. Right-click the **Temperature Scene** > **Displayers** > **Scalar 1** > **Parts** node and select **Edit...**.
4. In the *Parts* dialog, expand the **Regions** node.
5. Select **Battery Module: Cylindrical Cell: Jellyroll**.



6. Click **OK**.
7. Select the **Temperature Scene** > **Displayers** > **Scalar 1** > **Scalar Field** node.
8. In the *Scalar Field - Properties* window, set **Function** to **Temperature**.



9. Save the simulation.

Running the Simulation

Preparation of the simulation is now complete, and the simulation can be run.

To run the simulation:

1. Click (**Run**) in the *Solution* toolbar.

   The solution progress is displayed in the *Output* window. The *Residuals* display is created automatically in the *Graphics* window and shows the progress of the solvers.

   While the simulation is running, you can click the tabs at the top of the *Graphics* window to view scenes and plots.

   During the run, you can stop the process by clicking (**Stop**) on the toolbar. If you do stop the simulation, you can click (**Run**) to resume it.

   If left alone, the simulation continues until 200 seconds of physical time are complete. When the last iteration is complete, the following message is displayed in the *Output* window:

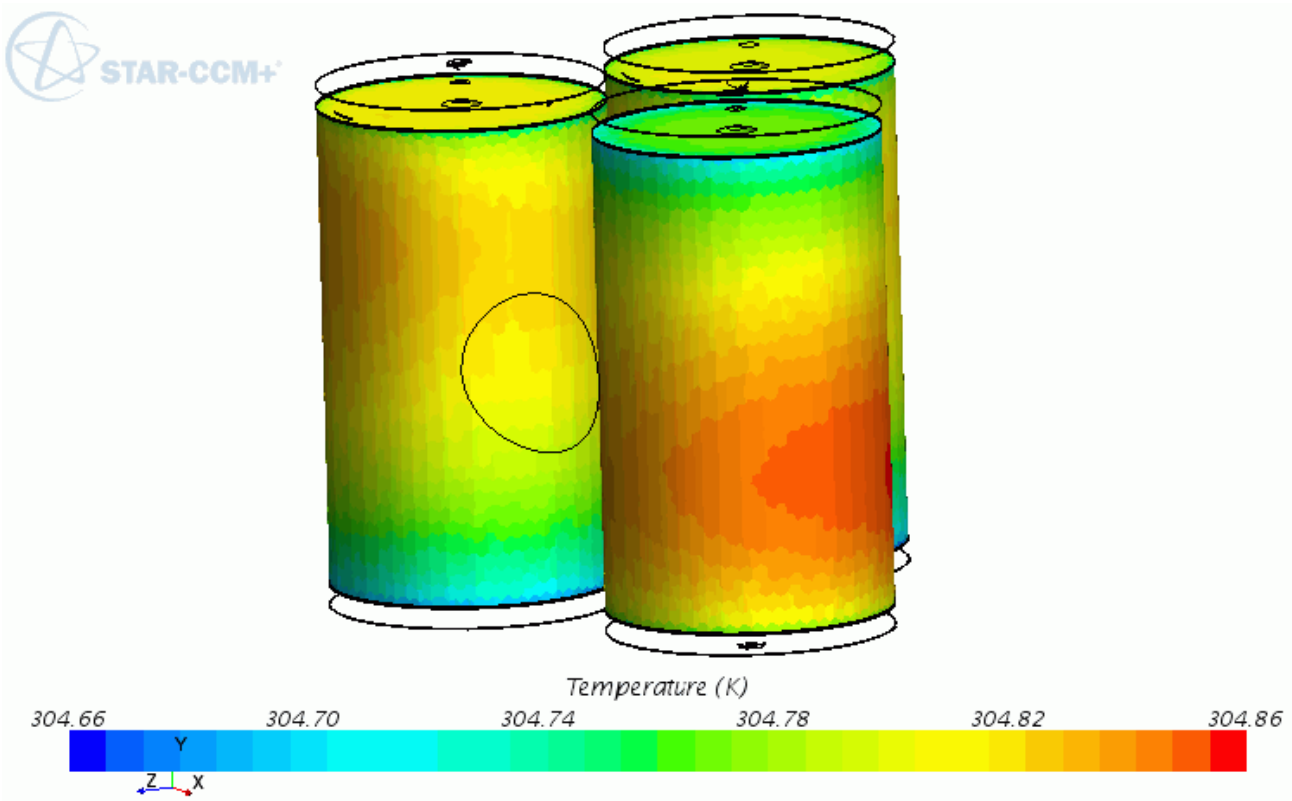
   ```
   Stopping criterion Maximum Physical Time satisfied.
   ```

2. When the simulation has finished running, save it.

Visualizing the Results

Visualize the effect of the draft in the external casing.

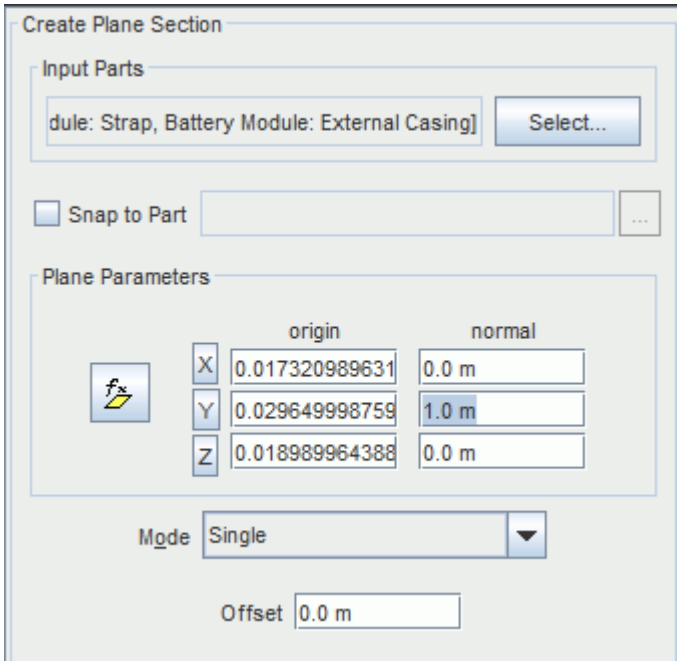
The scalar scene of temperature appears as follows.



Create a scalar scene to visualize the effect of the draft in the external casing.

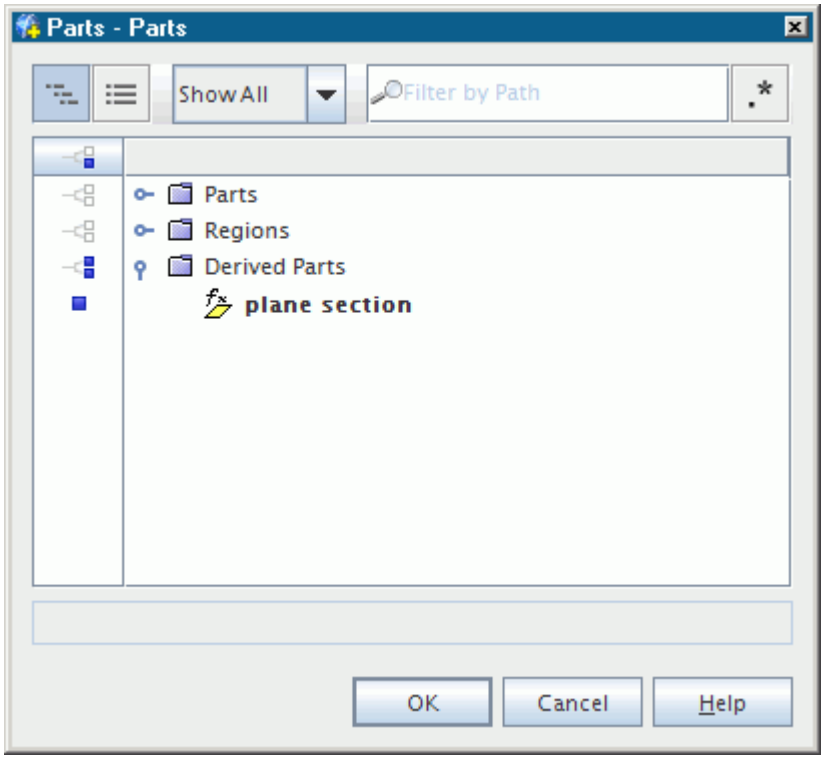
To visualize the results:

1. Right-click the **Scenes** > **Temperature Scene** node and click **Copy**.
2. Right-click the **Scenes** node and click **Paste**.
   A new scene, **Copy of Temperature Scene**, appears in the simulation tree.
3. Rename **Copy of Temperature Scene** to `Temperature Plane Section Scene`.
4. Open the *Temperature Plane Section Scene*.
5. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane...**.
   The *Create Plane Section* panel appears.
6. Set **normal** to `[0 m, 1 m, 0 m]` as shown below.



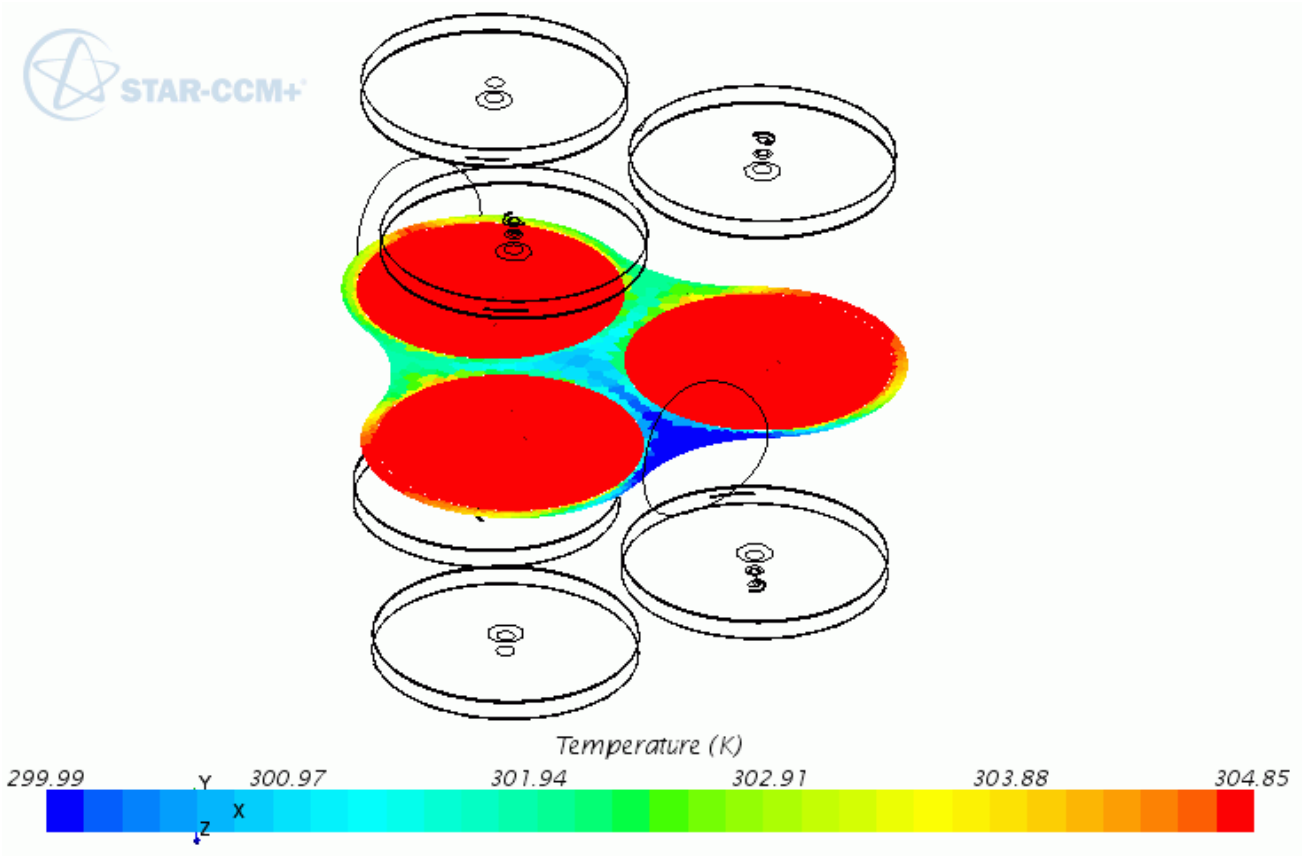
7. Set **Display** to **No Displayer**.
8. Click **Create**, and then **Close**.
9. Select **Scenes** > **Temperature Plane Section Scene** > **Displayers** > **Scalar 1** > **Parts** and click ... (**Custom Editor**) for the *Parts* property.

   The *Parts* dialog appears.
10. Click the **Column selection** button twice to deselect all parts.
11. Expand the **Derived Parts** node and select **Plane Section**.

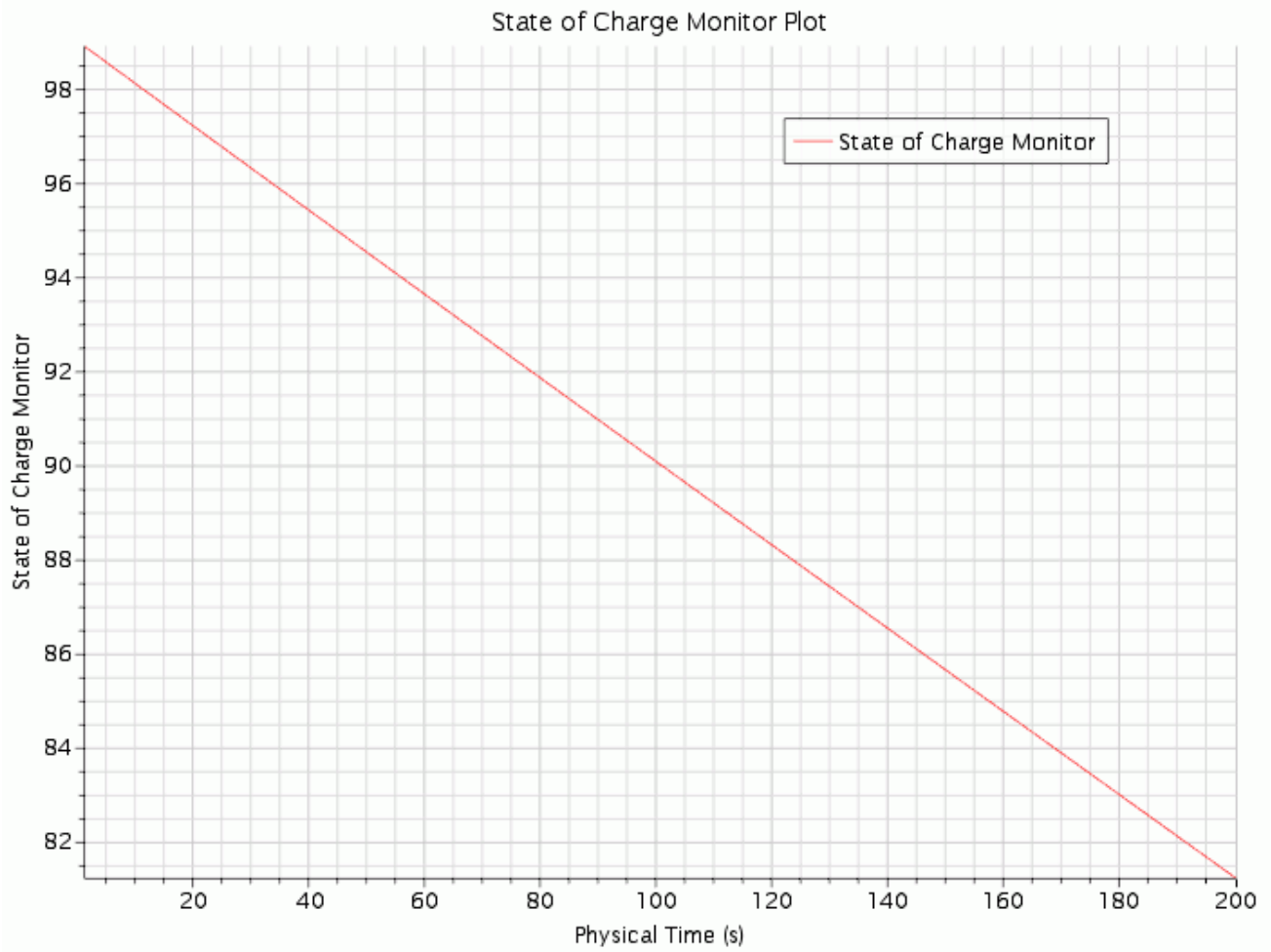


12. Click **OK**.

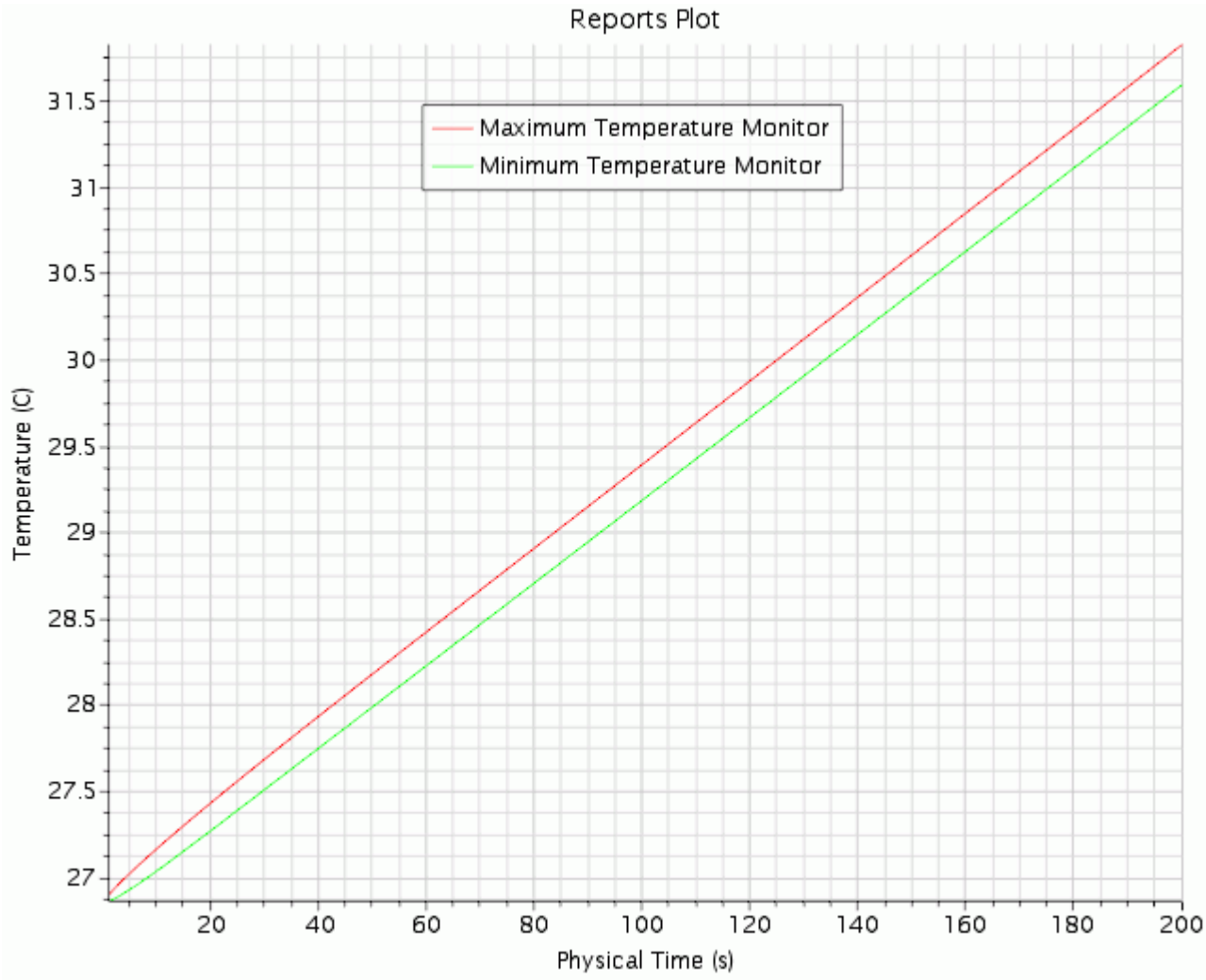
As expected, the temperature is lower near the inlet.



The battery state of charge has dropped to 80% as expected.



The temperature of the cells has risen by 5 C during the discharging process.



Summary

This tutorial demonstrated how to model a battery module in STAR-CCM+ using cylindrical cells.

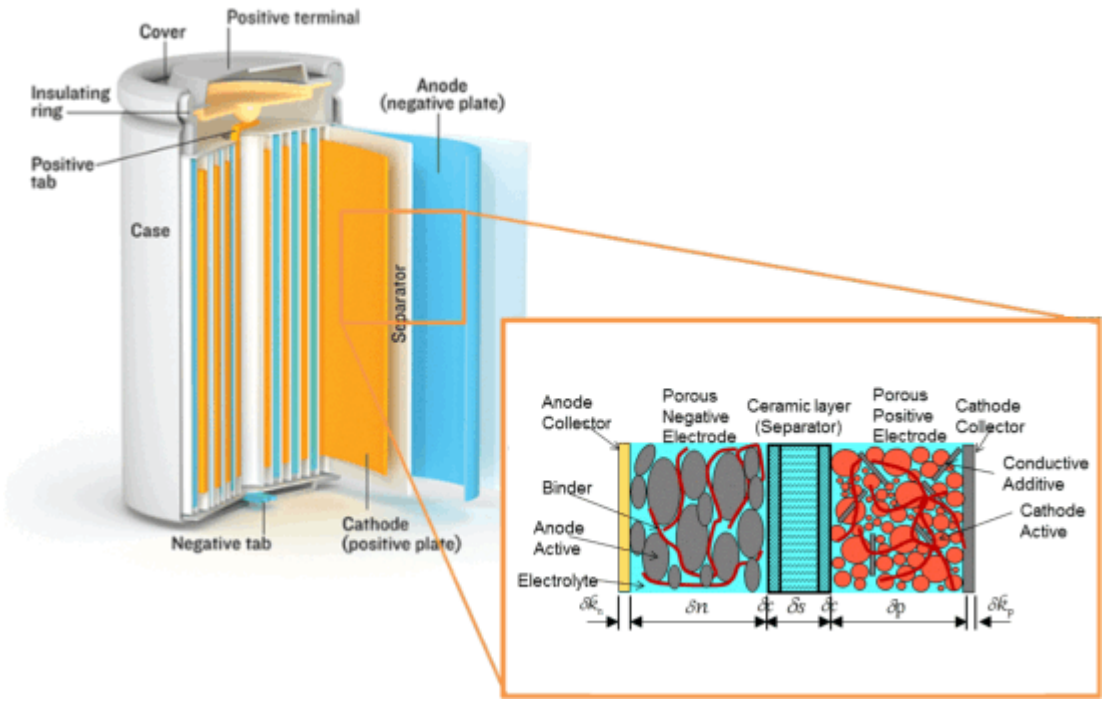
The steps covered were:

- Setting up the battery physics model.
- Imprinting parts.
- Generating a battery module by creating parts, connector parts, and regions.
- Generating a volume mesh using the battery mesher.
- Creating a circuit.

Li-Ion Battery Cell Model: Cell Electrochemistry Analysis

In this tutorial, you model a Li-Ion battery cell at the microscale level. The simulation introduces the method for the prediction of the three-dimensional microstructure of porous battery electrodes.

In a production environment, such analysis results in speeding up the design process of a battery cell, and assessing the safety and life-span of a design.



Generally, the structure of the active materials is irregular and complex, and varies from one active material to another. The structure of the solid particles is often represented by a simpler, regular structure. In this tutorial, you are given randomly distributed spherical particles that represent a single structure, and reflect the desired porosity.

In this tutorial, you model the following:

- electric potential in active materials, electrolytes, and current collectors
- concentration of lithium in active materials and electrolytes
- special conditions on the SEI (Solid Electrolyte Interface)

Note: To activate the Li-Ion battery cell, concentration, and electric potential models that are necessary for modeling the transport of Lithium ions, run a double-precision version of STAR-CCM+.

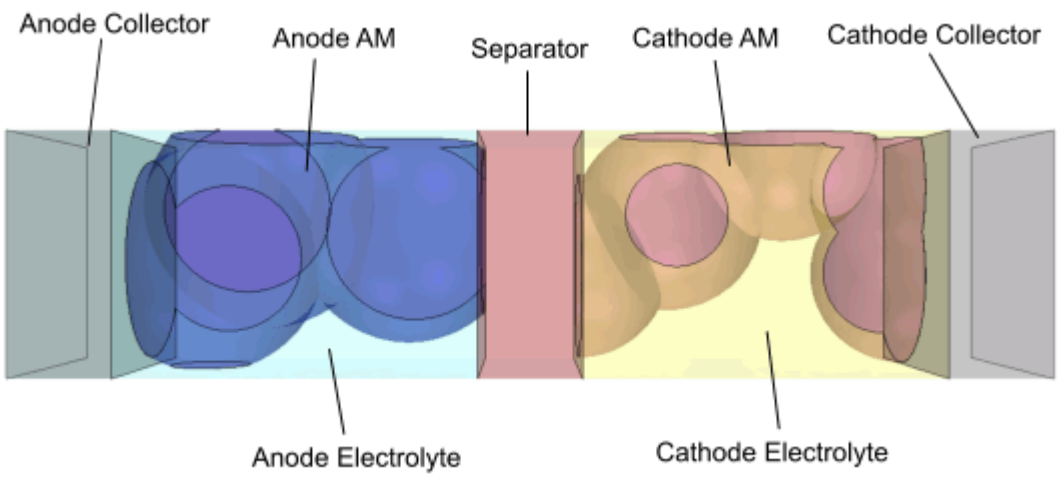
The geometry consists of a sample across a pair of electrodes from a typical Li-ion cell. The included parts of the geometry are:

- a cathode collector—10 m thick aluminum foil

- an anode collector—10 m thick copper foil
- electrolytes—a 50:50 mixture of ethylene carbonate/ethyl methyl carbonate, and $LiFP_6$ (lithium hexafluorophosphate—a common electrolyte salt in lithium batteries)
- a separator—10 m thick, with a porosity of 40%, and MacMullin number of 2.5.

  The MacMullin number is the ratio of the resistance of the separator when it is filled with electrolyte to the resistance of the electrolyte alone. This number describes the tortuosity when defining a homogeneous porous medium.
- a cathode AM (Active Material)—35 m thick $LiMn_2O_4$ (lithium manganese oxide), with particle diameter equal to 18 m, and a target porosity of 40%
- an anode AM—35 m thick graphite, with particle diameter equal to 20 m, and a target porosity of 40%

Each electrode is formed from a solid active material and the surrounding fluid electrolyte. These regions within an electrode are modeled separately. Chemical reactions occur at the SEI (Solid-Electrolyte Interfaces).



Note: The tutorial uses a simplified idealized microstructure of a $LiMn_2O_4$ electrode. The electrode structure is only represented by a few aggregated spherical particles and the electrolyte. The voluntary omission of the binders and conductivity aids is to speed up the computation for this tutorial. This case does not represent a commercially used electrode. However, it is relevant enough to compute a consistent solution and demonstrate the capabilities of the model.

Contents:

Prerequisites
Importing the Geometry and Naming the Simulation
Generating the Mesh
Setting Up the Physics Models
Assigning Physics Continua to Regions
Setting the Material Properties
Setting Initial Conditions
Setting Boundary and Interface Conditions
Setting Solver Parameters and Stopping Criteria

Preparing Scalar and Vector Scenes
Initializing the Solution and Running the Simulation
Visualizing the Results
Summary

Prerequisites

The instructions in the Li-Ion Battery Cell Model: Cell Electrochemistry Analysis tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Importing the Geometry and Naming the Simulation

Launch STAR-CCM+, import the supplied geometry, and save the simulation.

For this tutorial, you load a .sim file that contains the following objects:

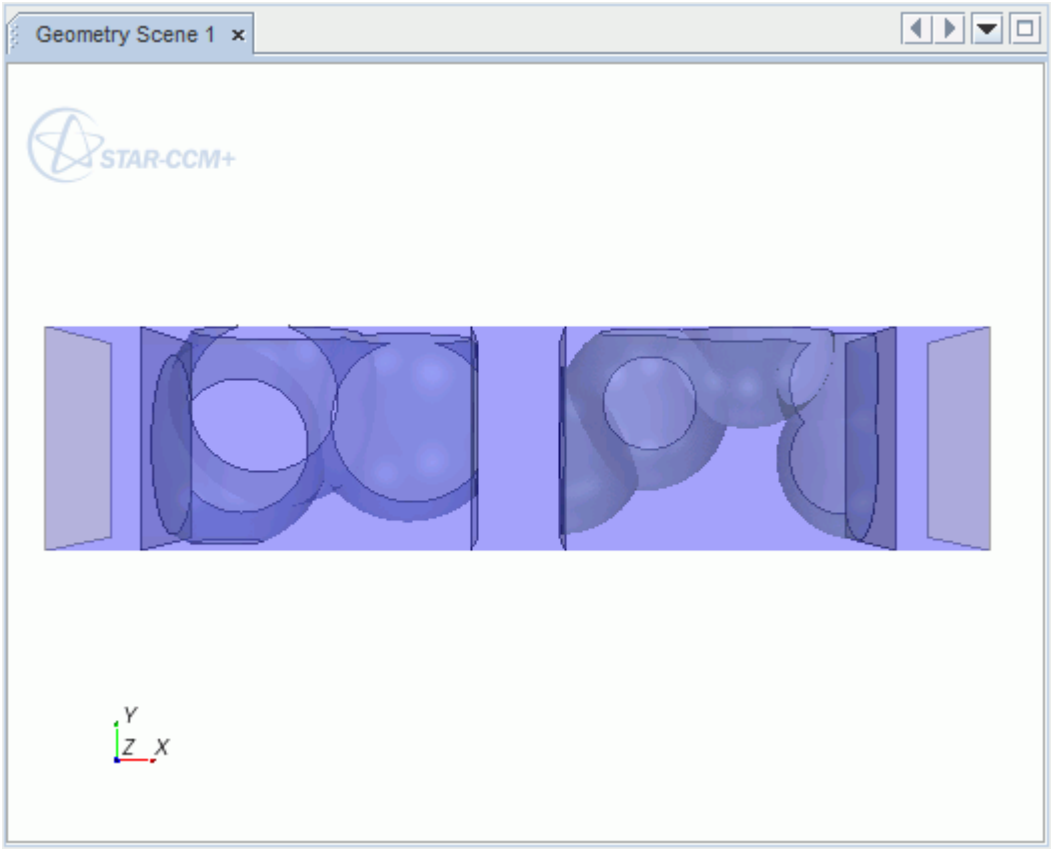
- Imprinted geometry parts: in order to create a matching (or conformal) mesh across an interface, the STAR-CCM+ meshers require that geometry faces on both sides of the interface are identical.
- Regions and interfaces.
- Boundaries with the type set correctly: as you are modeling only a section of a single layer cell, symmetry boundary conditions are specified on all outer walls of the cell cross-section.
- Three material property tables: one table contains the electrical and concentration properties of the electrolytes. Two tables contain data for the relationship between the equilibrium potentials and stoichiometries of the electrodes.

To import the geometry and name the simulation:

1. Launch a double precision version of STAR-CCM+.
2. Select **File** > **Load Simulation** in the menu bar.
3. In the *Load Simulation* dialog, click **Browse**.
4. Select `doc/startutorialsdata/battery/data/3DEChemGeometry.sim` and then click **OK**.

   STAR-CCM+ provides feedback on the import process, which takes a few seconds, in the *Output* window.
5. Save the simulation as `3DEChemTutorial.sim`.
6. Create a geometry scene.

The geometry with transparency turned on is shown below.



The active material spheres are represented by a single part for the anode, and a single part for the cathode. For each electrode, the ratio of the solid volume to the total volume is equivalent to the porosity.

Note: The imported geometry is based on a meter length scale instead of micrometer, which is the typical order of magnitude for these simulations. You scale the mesh in a later part of the tutorial.

Note: When using 3D-CAD to construct the geometry of a Li-Ion battery cell section, use the meter length scale. Due to the precision of the CAD tool (1.0E-8 m), you can experience difficulties while performing boolean operations and manipulating point coordinates if you use the micrometer scale.

Generating the Mesh

Use the polyhedral mesher to generate the volume mesh in all parts of the model.

As concentration gradients are highest across the SEI, you define prism layers to grow on either side of the interface. To ensure Li-ion/salt conservation on the interfaces between the electrodes and the separator, it is also best to designate a conformal mesh at these interfaces.

To generate the mesh:

1. Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
2. In the *Create Automated Mesh Operation* dialog:
   a) Select **Anode Collector** and **Cathode Collector** from the *Parts* list.

b) Select the following meshers, in order:

Group	Mesher
Surface Meshers	**Surface Remesher**
Core Volume Meshers	**Polyhedral Mesher**

c) Click **OK**.

3. Rename the **Automated Mesh** node to `Collector`.
4. Create another Automated Mesh operation.
   a) Select the following parts:
      - **Anode AM**
      - **Anode Electrolyte**
      - **Cathode AM**
      - **Cathode Electrolyte**
      - **Separator**

   b) Select the following meshing models:

Group	Mesher
Surface Meshers	**Surface Remesher**
Core Volume Meshers	**Polyhedral Mesher**
Optional Boundary Layer Meshers	**Prism Layer Mesher**

   c) Click **OK**.
5. Rename the **Automated Mesh** node to `AM Electrolyte Separator`.
6. Edit the **Operations** > **AM Electrolyte Separator** > **Default Controls** node and set the following:

Node	Property	Setting
Base Size	*Value*	`0.25 m`
Number of Prism Layers	*Number of Prism Layers*	`10`

7. Edit the **Operations** > **Collector** > **Default Controls** node and set the following:

Node	Property	Setting
Base Size	*Value*	`0.1 m`

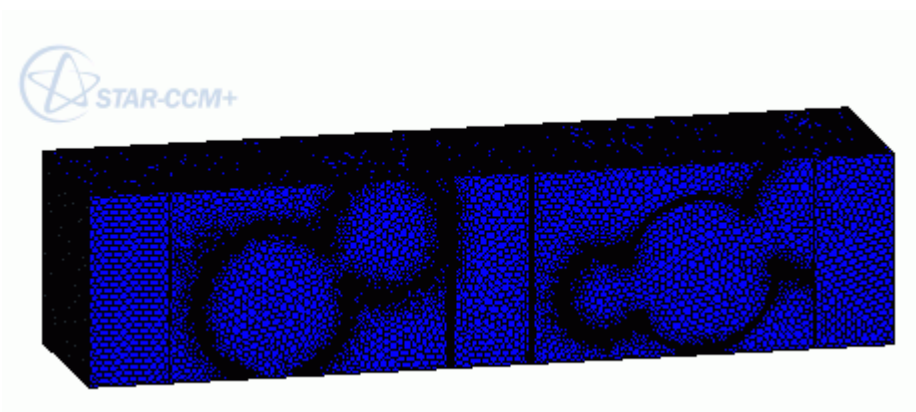
Prism layers are required only at the SEI. Deactivate prism layers on all other boundaries, and activate them on the SEI:

8. Right-click the **Operations** > **AM Electrolyte Separator** > **Custom Controls** node and select **New** > **Surface Control**.
9. Rename the **Surface Control** node to `Disabled Prism Layers`.
10. Expand the **Disabled Prism Layers** node and set the following:

Node	Property	Setting
Disabled Prism Layers	*Part Surfaces*	• **Anode AM** > **Surfaces** > **Anode AM/Anode Collector** • **Anode Electrolyte** > **Surfaces** > **Anode Electrolyte/Anode Collector** • **Cathode AM** > **Surfaces** > **Cathode AM/Cathode Collector** • **Cathode Electrolyte** > **Surfaces** > **Cathode Electrolyte/Cathode Collector**
Controls > **Prism Layers**	*Prism Layers*	**Disable**

You are ready to generate the volume mesh.

11. Right-click the **Geometry** > **Operations** node and select **Execute All**.
12. From the menu bar, select **Mesh** > **Scale Mesh...**.
13. In the *Scale Mesh* dialog, set the **Scale Factor** to `1E-5`.
14. Click **Apply** to scale down the mesh.
15. Click **Close**.
16. Create a Mesh Scene.



17. Save the simulation.

Setting Up the Physics Models

Physics models define the primary variables of the simulation and what mathematical formulation is used to generate the solution.

Physics continua are required for the following:

- anode and cathode active materials
- anode and cathode collectors
- electrolyte regions and separator, represented by one fluid continuum, with additional porosity defined for the separator.

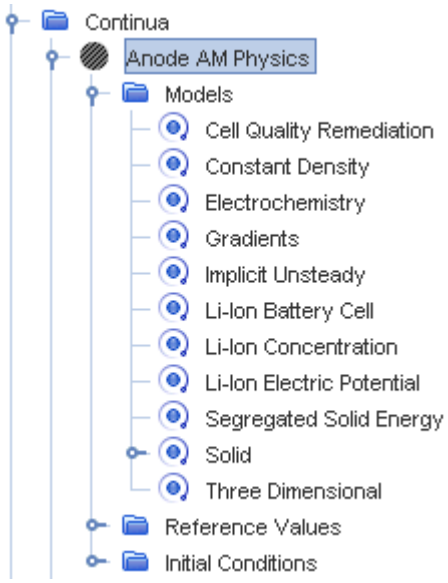
STAR-CCM+ creates a physics continuum, **Physics 1**, automatically when you generate a mesh.

To select the physics models:

1. Rename the **Continua** > **Physics 1** node to **Anode AM Physics**.
2. For the physics continuum, **Continua** > **Anode AM Physics**, make sure that **Auto-select recommended models** is activated and select the following models in order:

Group Box	Model
Space	**Three Dimensional** (selected automatically)
Time	**Implicit Unsteady**
Material	**Solid**
Optional Models	**Segregated Solid Energy** **Gradients** (selected automatically)
Equation of State	**Constant Density**
Optional Models	**Electrochemistry**
Electrochemistry	**Li-Ion Battery Cell**
Optional Models	**Li-Ion Concentration**
	Li-Ion Electric Potential
	Cell Quality Remediation

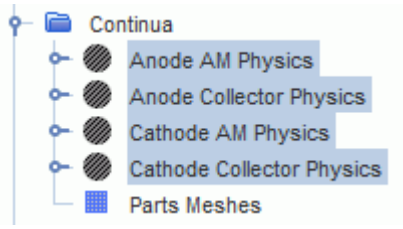
3. Click **Close**.
4. To review the models, open the **Anode AM Physics** > **Models** node.



To create physics continua for the other solid materials, copy the **Anode AM Physics** continuum:

5. Right-click the **Continua** > **Anode AM Physics** node and select **Copy**.
6. Right-click the **Continua** node and select **Paste**.

   STAR-CCM+ adds a **Copy of Anode AM Physics** node to the feature tree.
7. Repeat the paste process two more times.
8. Rename the physics continua nodes as follows:
   - **Copy of Anode AM Physics** to `Anode Collector Physics`
   - **Copy of Anode AM Physics 2** to `Cathode AM Physics`
   - **Copy of Anode AM Physics 3** to `Cathode Collector Physics`



Next, you create a fluid physics continuum for the electrolytes and separator:

9. Create a physics continuum.
10. Rename the **Continua** > **Physics 1** node to **Electrolyte Physics**.
11. For the physics continuum, **Continua** > **Electrolyte Physics**, select the following models in order:

Group Box	Model
Space	**Three-Dimensional**
Time	**Implicit Unsteady**

Group Box	Model
Material	**Liquid**
Flow	**Segregated Flow** **Gradients** (selected automatically)
Viscous Regime	**Laminar**
Equation of State	**Constant Density**
Optional Models	**Electrochemistry**
Electrochemistry	**Li-Ion Battery Cell** (**Li-Ion Concentration** and **Li-Ion Electric Potential** are selected automatically)
Energy	**Segregated Fluid Temperature**
Optional Models	**Cell Quality Remediation**

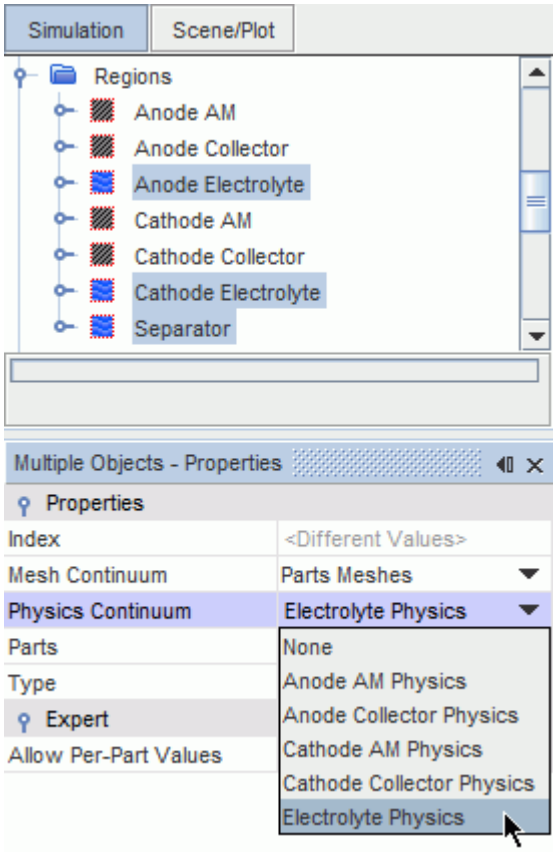
12. Click **Close**.

Assigning Physics Continua to Regions

Assign each physics continuum to its appropriate region.

1. Assign the physics continua to the appropriate regions, as follows:

Region	Physics Continuum
Anode AM	**Anode AM Physics**
Anode Collector	**Anode Collector Physics**
Cathode AM	**Cathode AM Physics**
Cathode Collector	**Cathode Collector Physics**
Anode Electrolyte	**Electrolyte Physics**
Cathode Electrolyte	
Separator	



The separator region acts as a porous region rather than a fluid one between the electrodes, with 40% porosity:

2. Select the **Regions** > **Separator** node and set *Type* to **Porous Region**.
3. Select the **Regions** > **Separator** > **Physics Values** > **Porosity** node and set *Value* to `0.4`.

The physics continua of the collectors do not use the Li-Ion concentration model:

4. For the **Anode Collector Physics** and **Cathode Collector Physics** continua, do the following:
   a) Right-click the node and choose **Select models...**.
   b) Deactivate the **Li-Ion Concentration** model from the *Enabled Models* group box.
   c) Click **Close**.
5. Save the simulation.

Setting the Material Properties

Define the material properties for each continuum.

To set the material properties:

Set the anode active material and current collector to graphite and copper, respectively.

1. Right-click the **Anode AM Physics** > **Models** > **Solid** > **Al** node and select **Replace with**.
2. In the *Replace Material* dialog, activate **C (Graphite)** and then click **OK**.

3. Repeat the process for the **Anode Collector Physics** continuum, but replace the material with **Cu (Copper)**.

Set the material properties for the cathode active material.

4. Expand the **Continua** > **Cathode AM Physics** > **Models** > **Solid** node.
5. Rename the **Al** node to **LiMn2O4**.
6. Select the **LiMn2O4** > **Material Properties** > **Electrical Conductivity** > **Constant** node and set *Value* to `0.01 S/m`.

   The Lithium/Salt Diffusivity for LiMn2O4 is similar to the value for aluminum: therefore, the default value is fine. The other material properties are not important for the accuracy of this tutorial and are left unaltered.

   The cathode collector remains as aluminum.

You treat the electrolyte as a mixture of LiPF6 salt in an equal mixture of ethylene carbonate and ethyl methyl carbonate.

Set the material properties for the electrical conductivity, Li+/salt diffusivity, and transference number using a material table. The simulation file contains this table. Each material property is interpolated using two independent variables: temperature, and salt concentration.

7. Rename **Continua** > **Electrolyte Physics** > **Models** > **Liquid** > **H2O** to **LiPF6**.
8. Select the **LiPF6** > **Material Properties** > **Electrical Conductivity** node and set *Method* to **Tabular**.
9. Select the **Electrical Conductivity** > **Tabular** > **Table** node and set *Input table* to **electrolyteLiPF6**.
10. Select the **Table** > **Interpolated** node and set *Column title* to **Spec.Cond**.
11. Right-click the **Table** > **Variables** node and select **New**.
12. Select the **Variables** > **Variable 1** node and set **Scalar Function** to **Temperature**.

    The **Variable 1** node is automatically renamed to **Temperature**.
13. Set *Column title* to **Temp**.
14. Create one more variable, then set the *Scalar Function* to **Lithium/Salt Concentration**, and the *Column Title* to **c2**—the salt concentration in the liquid phase.

Set the tabular material property for the **Lithium/Salt Diffusivity** and **Lithium Cation Transference Number** material properties.

15. Repeat the previous steps to set the tabular material property for the **Lithium/Salt Diffusivity** and **Lithium Cation Transference Number** material properties.

    The input table for both properties is also **electrolyteLiPF6**, and the first independent variable is **Temperature**.

    Note the following differences:

    - For **Lithium/Salt Diffusivity**, set *Column title* of the **Interpolated** node to **Diff.Coef**.
    - For **Lithium Cation Transference Number**, set *Column title* of the **Interpolated** node to **t+**.

    The remaining material properties remain at their default values.

Setting Initial Conditions

Set the initial conditions for the physics continua.

The initial conditions of the active materials represent a 50% utilized electrode.

To set the initial conditions:

Set the initial conditions for the **Anode AM Physics** continuum.

1. Edit the **Initial Conditions** node and set the following property:

Node	Property	Setting
Lithium/Salt Concentration	*Value*	`13 kmol/m^3`

Set the initial conditions for the **Cathode AM Physics** continuum.

2. Edit the **Initial Conditions** node and set the following property:

Node	Property	Setting
Lithium/Salt Concentration	*Value*	`13 kmol/m^3`

Set the initial conditions for the **Electrolyte Physics** continuum.

3. Edit the **Initial Conditions** node and set the following property:

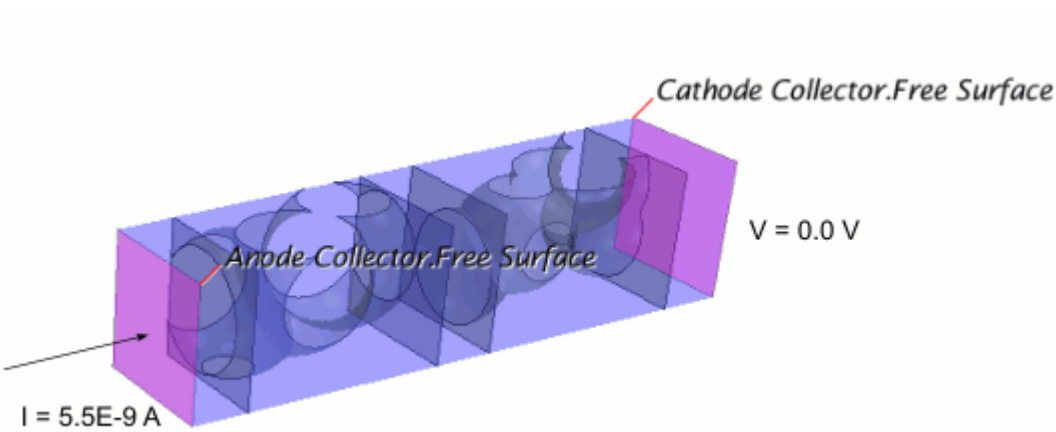
Node	Property	Setting
Lithium/Salt Concentration	*Value*	`1 kmol/m^3`

4. Save the simulation.

Setting Boundary and Interface Conditions

Define the boundary conditions at the free surfaces of the two current collectors.

To simulate an approximate discharge current of 10 A/m^2 across the cell segment, you define the current density at the anode collector, and a zero electric potential at the end of the cathode collector.



The discharge process is an electrochemical process: under an applied electric current, the lithium that is inserted in the anode active material is dissociated into a Li^+ ion and an electron. The Li^+ ion then migrates

through the separator and is inserted into the cathode active material. Hence, the concentration of Li^+ ion decreases on the anode side and increases on the cathode side over time.

You specify the current density, in A/m^2, at the anode since both the current and the anode areas are known. To allow a null net flux, you set the electrical potential at the outlet to 0 V.

To set the boundary and interface conditions:

1. Expand the **Regions** > **Anode Collector** > **Boundaries** > **Free Surface** node.
2. Select the **Physics Conditions** > **Electric Potential Specification** node and make sure that *Method* is set to **Specific Electric Current**.
3. Select the **Physics Values** > **Specific Electric Current** node and set *Value* to `10 A/m^2`.
4. Expand the **Regions** > **Cathode Collector** > **Boundaries** > **Free Surface** node.
5. Select the **Physics Conditions** > **Electric Potential Specification** node and set *Method* to **Electric Potential**.

   Leave the default `0.0 V` for the corresponding value of electric potential.

At the SEI, the Butler-Volmer relationship is used to compute the normal component of the electric current density at the SEI. The equilibrium potential of the active material term varies with the relative concentration in the electrode. Two tables are provided for you to use in the positive and negative electrodes.

6. Multi-select the **Interfaces** > **Anode AM/Anode Electrolyte** node and the **Anode AM/Separator** node.
7. Right-click any selected node and select **Edit...**.

   The *Multiple Objects* dialog appears.
8. Expand the **Physics Values** > **Butler-Volmer Relationship Parameters** node and set *Electrode Equilibrium Potential* to **Tabular**.

   The **Tabular Electrode Equilibrium Potential** node appears.
9. Expand the **Tabular Electrode Equilibrium Potential** node:
   - Set *Table* to **UeqNeg**.
   - Set *Relative Concentration in Electrode (y)* to **y**.
   - Set *Electrode Equilibrium Potential* to **- Ueq**.
10. Click **Close**.
11. Repeat the interface condition procedure for the **Cathode AM/Cathode Electrolyte** and **Cathode AM/Separator** interfaces, but set *Table* to **UeqPos** and *Electrode Equilibrium Potential* to **Ueq**.

Setting Solver Parameters and Stopping Criteria

Set the appropriate solver parameters and stopping criteria for the simulation.

In this tutorial, charge and discharge currents have low values, and all variable changes are slow. Therefore, you can increase the time-step of the implicit solver to a large value.

The solver requirements for three-dimensional electrochemistry are higher than usual, because of the large differences in electrical and species conductivities in the different regions. These differences in conductivities affect the linear system. Therefore, you can change the expert properties of the AMG Linear Solver for Electric Potential.

To set the solver parameters and stopping criteria:

1. Edit the **Solvers** node and set the following properties:

Node	Property	Setting
Implicit Unsteady	**Time-Step**	`10.0 s`
Electric Potential > **AMG Linear Solver**	*Enable Direct Solver*	Activated
	Maximum Direct Solver Equations	`1024`

2. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Physical Time	*Enabled*	Activated
	Maximum Physical Time	`100 s`
Maximum Inner Iterations	*Maximum Inner Iterations*	`10`

3. Save the simulation.

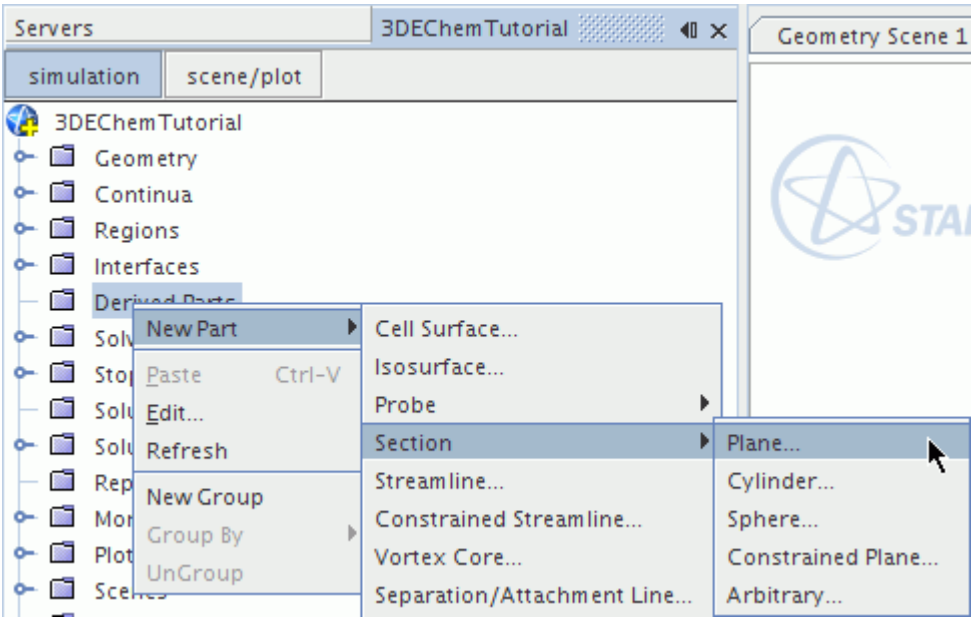
Preparing Scalar and Vector Scenes

Create scalar and vector scenes to display the simulation results.

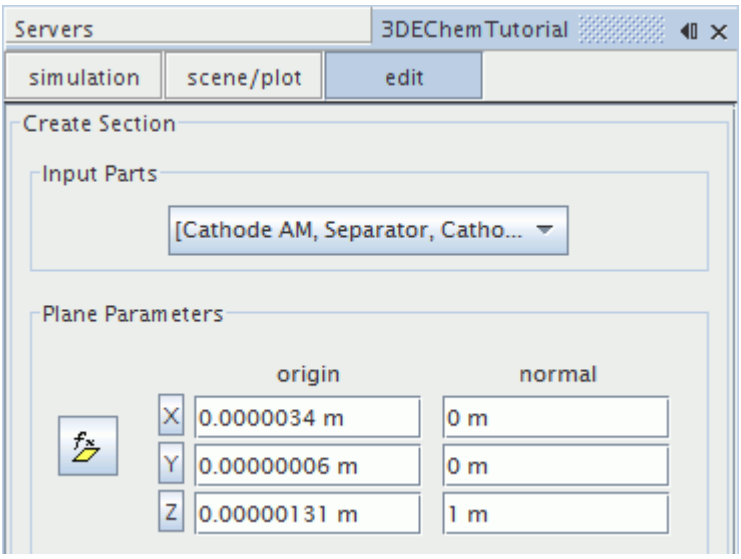
You create two scalar scenes to display the Li-Ion/salt concentration across a plane section of the battery cell, and the specific electric current on the SEI.

To prepare the scalar and vector scenes:

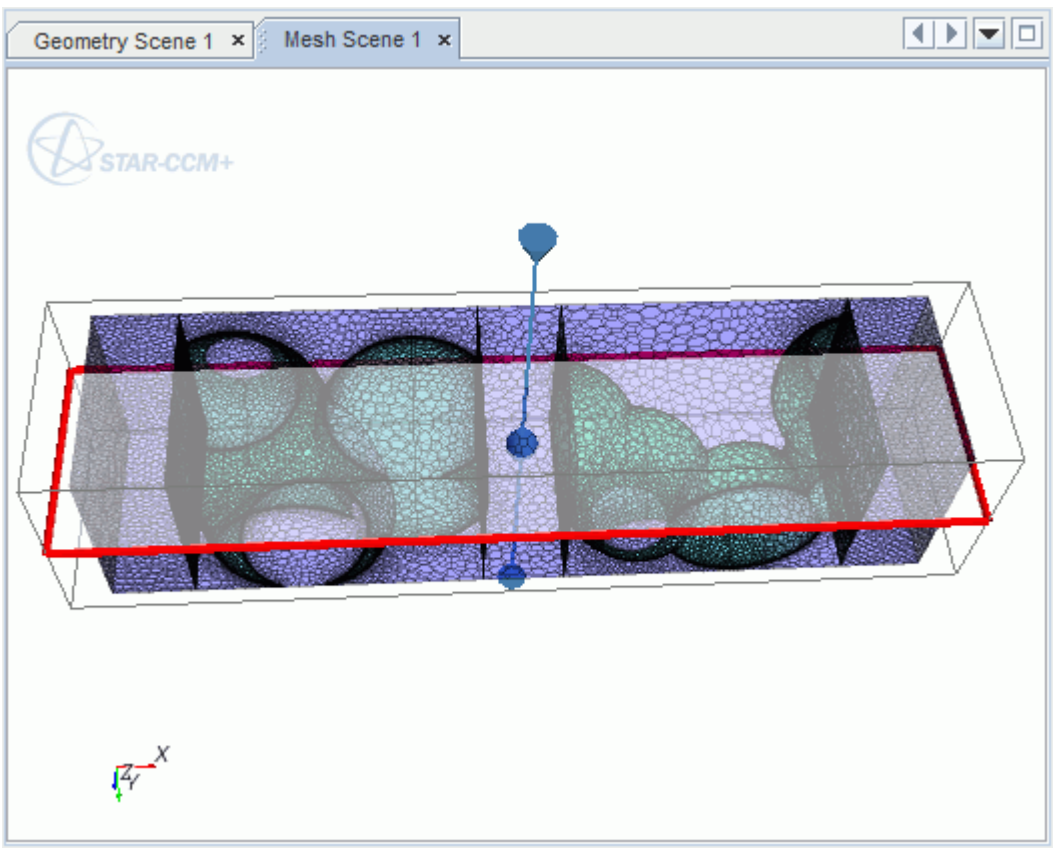
1. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane** .



2. In the *Edit* dialog, set the normal to `x = 0 m`, `y = 0 m`, and `z = 1 m`.

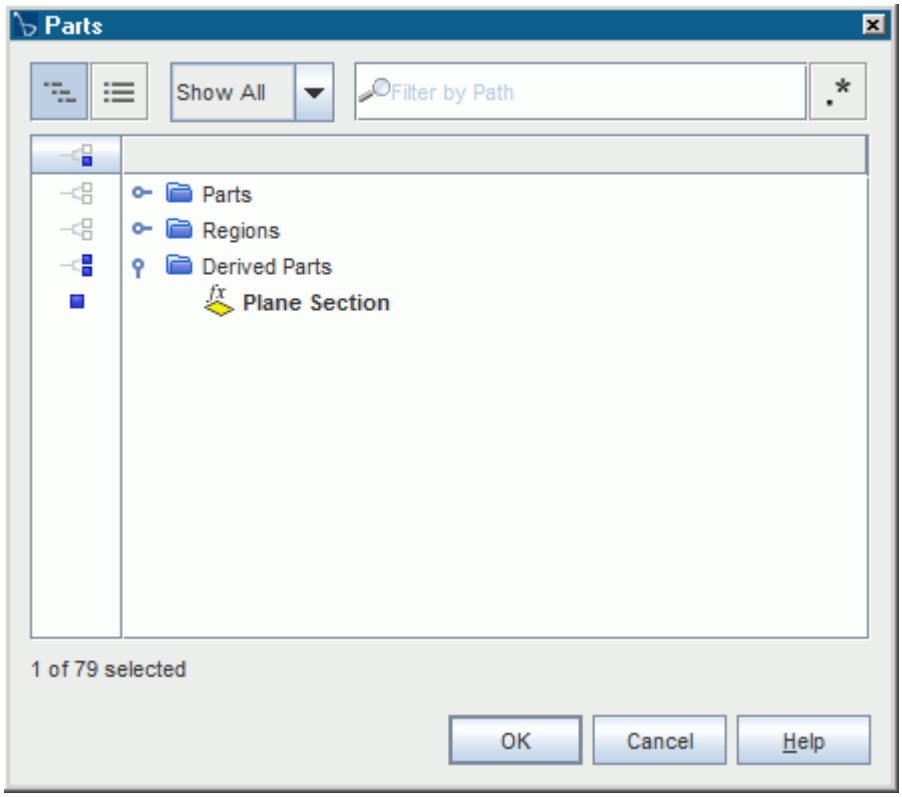


The plane scene looks as follows.

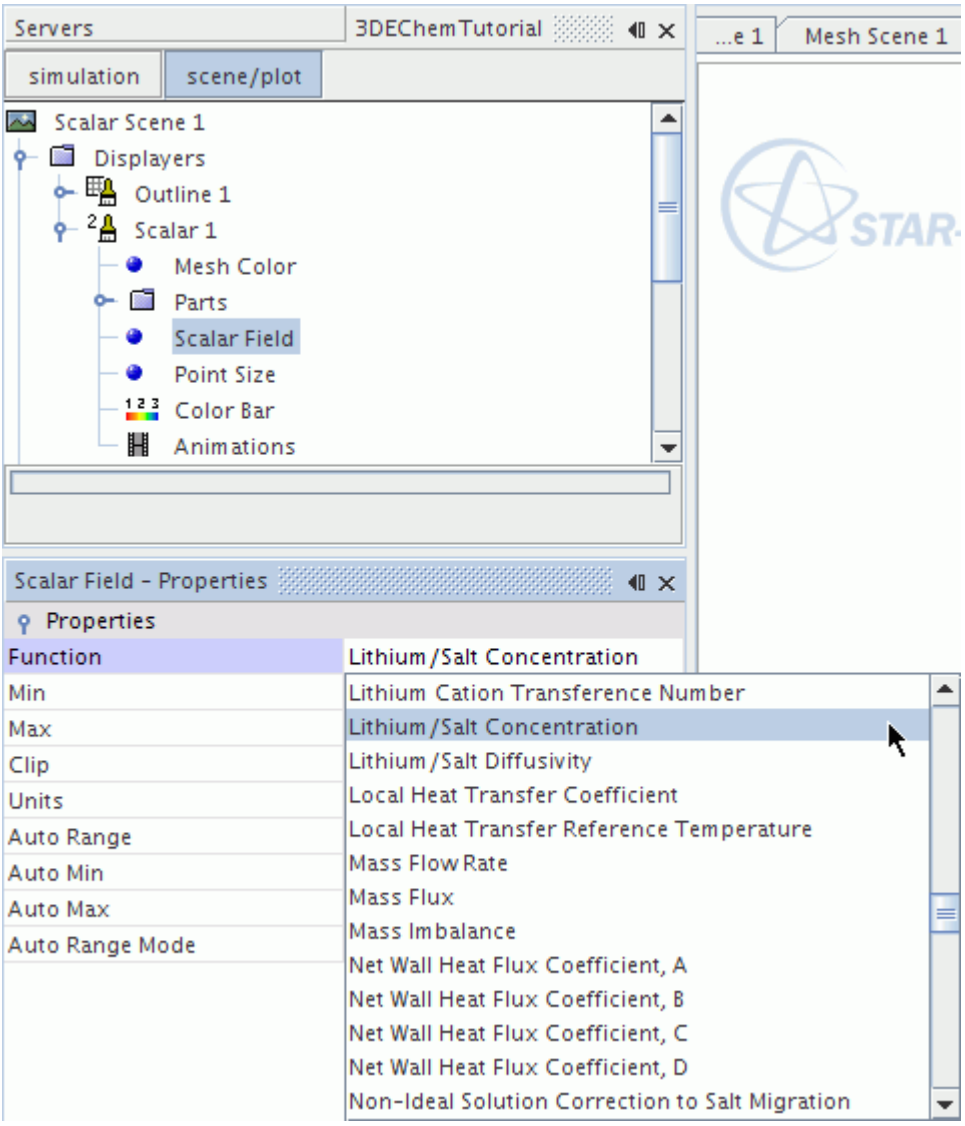


3. Click **Create**, and then **Close**.
4. Create a scalar scene.

5. Right-click the **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Parts** node and select **Edit...**.
6. In the *Parts* dialog, activate **Derived Parts** > **Plane Section**.



7. Click **OK**.
8. Click (**Show All Meshes**).
9. Select the **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Scalar Field** node and set *Function* to **Lithium/ Salt Concentration**.



Prepare a vector scene to visualize the electronic pathways.

10. Create a vector scene.
11. Select the **Vector Scene 1** > **Displayers** > **Vector 1** > **Parts** node and click to the right of the *Parts* property.
12. Activate **Derived Parts** > **Plane Section** and click **OK**.
13. Select the **Vector Scene 1** > **Displayers** > **Vector 1** > **Vector Field** node and set *Function* to **Electric Current Density**.

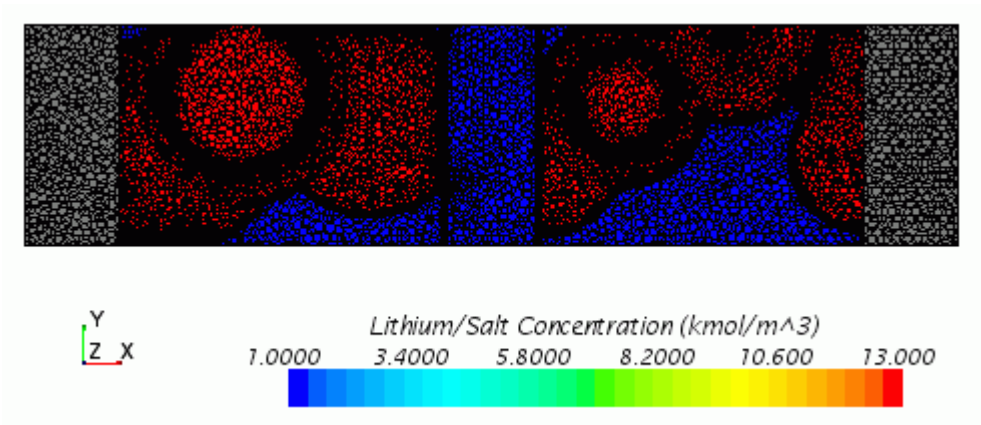
Initializing the Solution and Running the Simulation

Preparation of the simulation is now complete, and the simulation can be run.

To initialize the solution and run the simulation:

1. Click (**Initialize Solution**) in the *Solution* toolbar or select **Solution** > **Initialize Solution** from the menu.

   The initial Lithium/Salt Concentration looks as follows:



2. Click (**Run**) in the *Solution* toolbar.

   The solution progress is displayed in the *Output* window. The *Residuals* display is created automatically in the *Graphics* window and shows the progress of the solvers.

   While the simulation is running, you can click the tabs at the top of the *Graphics* window to view scenes and plots.

   During the run, you can stop the process by clicking (**Stop**) on the toolbar. If you do stop the simulation, you can click (**Run**) to resume it.

   If left alone, the simulation continues until 100 seconds of physical time are complete. When the last iteration is complete, the following message is displayed in the *Output* window:

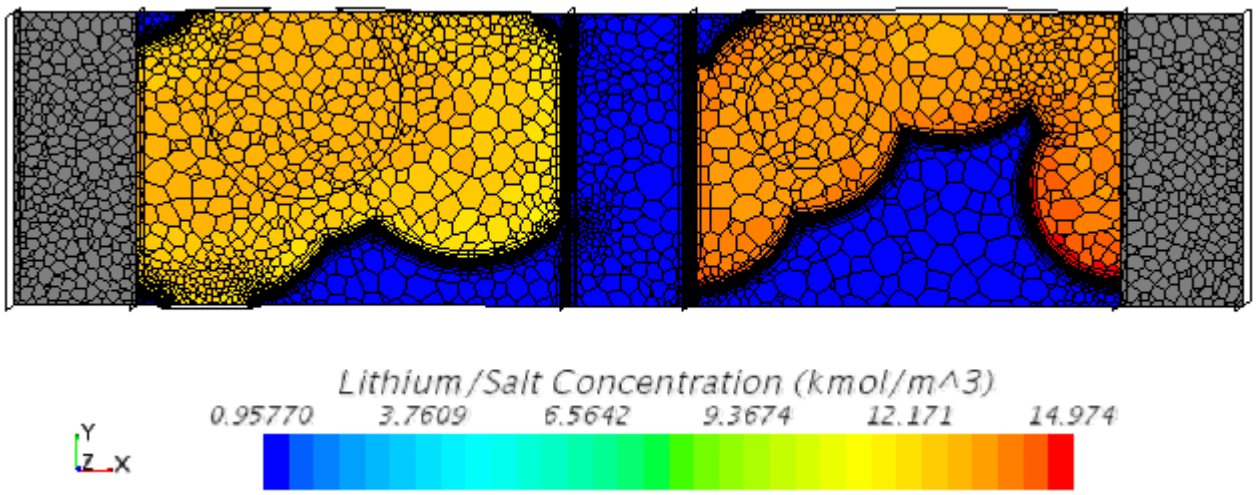
   ```
   Stopping criterion Maximum Physical Time satisfied.
   ```

3. When the simulation has finished running, save it.

Visualizing the Results

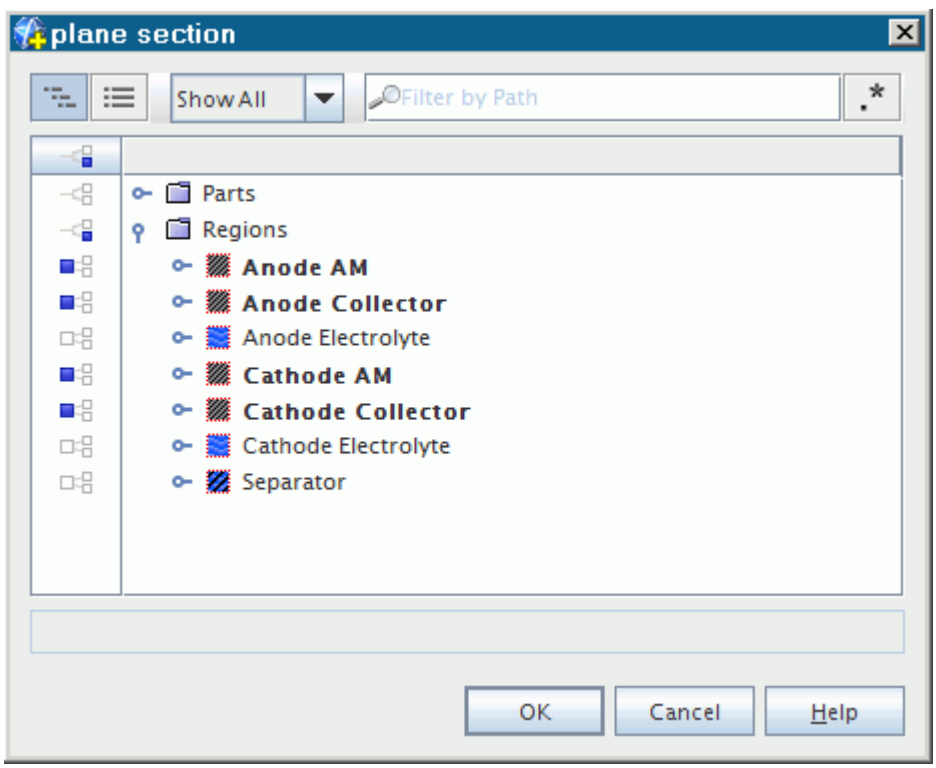
Visualize the Lithium/salt concentration in the active materials.

The scalar scene of the Li-Ion/salt concentration looks as follows.



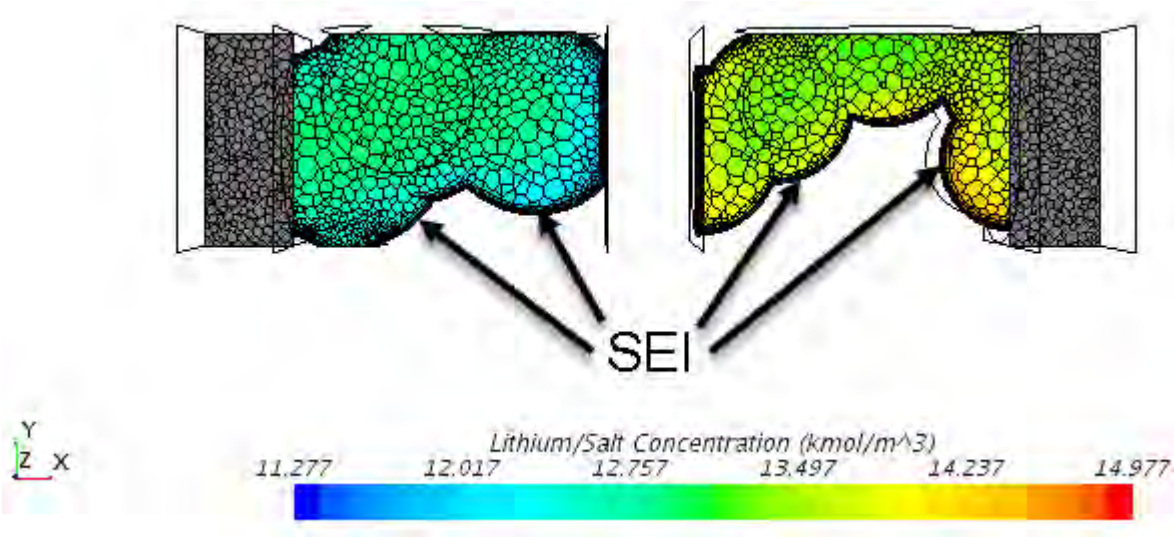
To visualize the Lithium/salt concentration in the active materials:

1. Select the **Derived Parts** > **plane section** node and click ... (**Custom Editor**) for the *Parts* property.
2. In the *plane section* dialog, expand the **Regions** node and deactivate **Anode Electrolyte**, **Cathode Electrolyte**, and **Separator**.



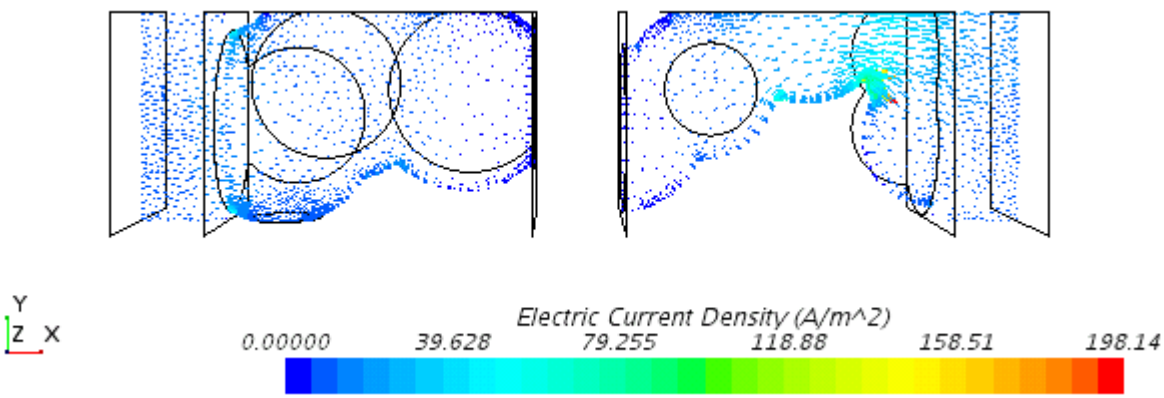
3. Click **OK**.

The Lithium/salt concentration in the active materials looks as follows.



On the anode side (left-hand side in the above screenshot), the concentration distribution in the active material indicates a discharge process. The lower level of concentration next to the SEI, compared with the core part, is consistent as the Li+ ions in this region naturally migrate in the electrolyte.

On the cathode side, higher levels of concentration are observed in the region next to the SEI, where the Li+ ions are inserted. The particle next to the current collector has the higher concentration level, which is a result of the high current density in this location.



By reviewing the Li+ ion concentration distribution on this structure, it is possible to observe whether large concentration gradients occur over time. These large concentration gradients are undesirable as they can lead to an increase in cell impedance, which results in large electric potential drop—hence adversely affecting the energy that is delivered by the cell.

Analyzing the current density distribution, especially on the anode side, can give good indications of potential occurrence of certain undesired side reactions, such as lithium deposition. These reactions are more likely to occur in regions of high current density.

This feature allows the cell manufacturer to understand the electrochemical process under a given load, and analyze the effects of various parameters—particle size, porosity, or total electrode thickness—on the cell performance.

Summary

This tutorial demonstrated how to model a Li-Ion battery cell at the microscale level.

The steps covered were:

- Setting up an electrochemistry model.
- Post-processing Lithium/Salt Concentration and SEI Specific Electric Current.

Casting

The tutorials in this set illustrate various STAR-CCM+ features for setting up casting:

Contents:

Basic Gravity Sand Casting
STAR-Cast: Investment Casting

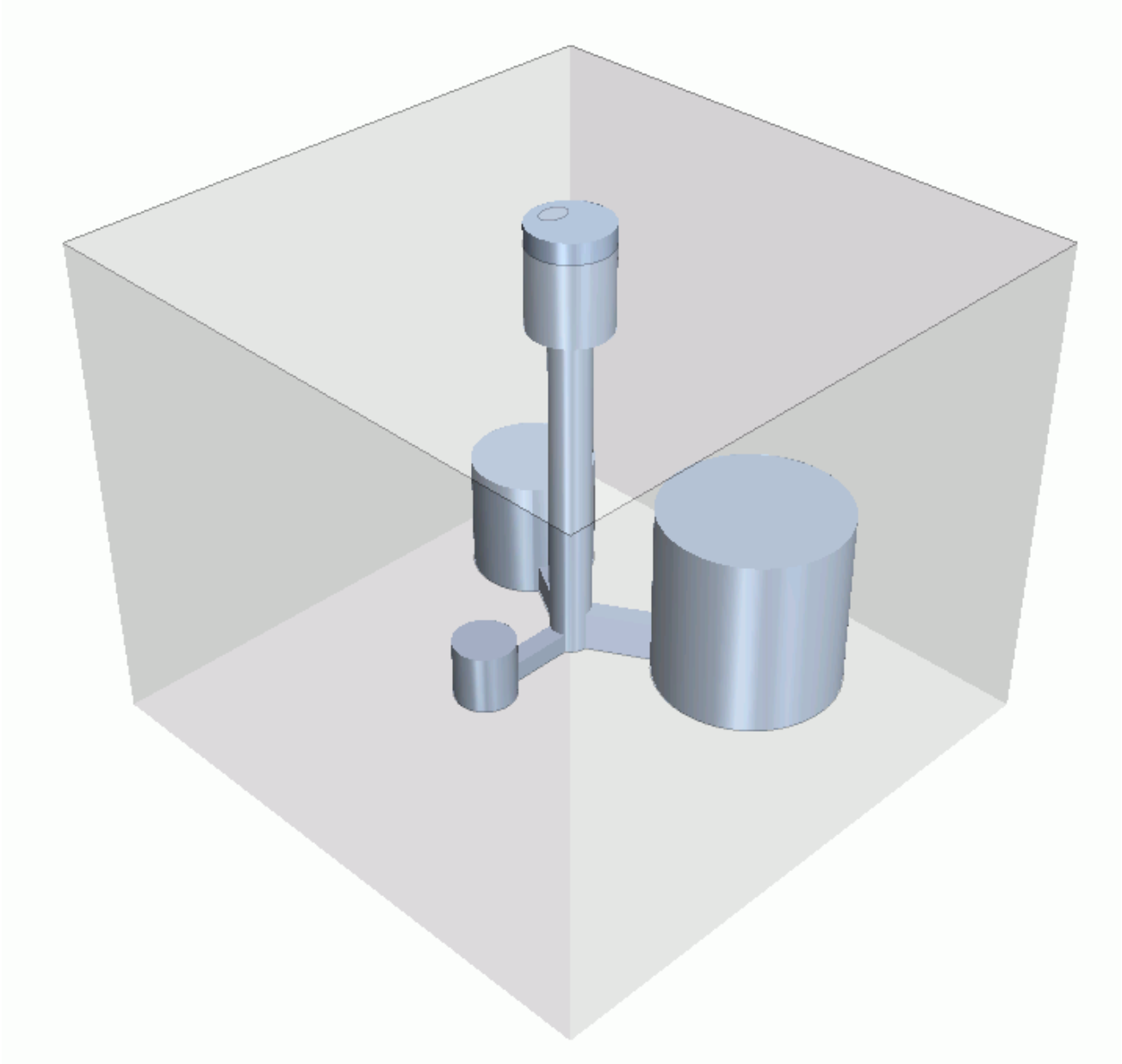
Basic Gravity Sand Casting

This tutorial demonstrates the setup, running, and post-processing of a basic gravity sand casting analysis using STAR-CCM+.

The mold is made of furan sand. The aluminum alloy AlSi7Mg0.3 is poured from a casting ladle into a sprue cup.

The casting scenario consists of a rectangular sand mold with a cavity composed of three cylindrical bodies and a gating system. The inflow condition assumes that the melt enters at constant pressure from the start of the simulation.

For this case, the sprue cup and casting regions are initially empty. The sprue cup outlet is set to atmospheric pressure. The mass flow into the pour cup corresponds to the flow rate that is required to fill the casting region in about 5–6 s.



This tutorial first simulates the filling of the casting region. You also simulate solidification after the mold is filled with the melt.

Depending on the specifications of your computer, the run time of this tutorial is of the order of several days. You are advised to run the tutorial in parallel to decrease the run time.

This tutorial uses the Criteria Functions model and therefore requires a double-precision version of STAR-CCM+.

Contents:

Prerequisites
Loading the Geometry
Generating the Mesh

Setting Up the Physics Models
Selecting the Casting Materials
Setting Up the Sand Mold Physics Properties
Creating Eulerian Phases and Setting Phase Models
Setting Up the Casting Properties
Setting Region and Boundary Conditions
Setting Solver Parameters and Stopping Criteria
Creating a Solution History File
Visualizing the Solution
Running the Simulation
Visualizing the Results
Creating a Melt Filling Animation
Summary

Prerequisites

The instructions in this tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Techniques	Associated Tutorial
The STAR-CCM+ workflow	Introduction
Using visualization tools, scenes, and plots	Introduction

Loading the Geometry

To set up the STAR-CCM+ simulation, launch a simulation and import the supplied geometry.

1. Start up a double precision version of STAR-CCM+ in a manner that is appropriate to your working environment.

A file has been prepared that contains geometry parts for:

- The sand mold
- The sprue cup
- The casting region

To load the file:

2. In the menu bar, select **File** > **Load...**.
3. In the *Load a File* dialog, click **Browse...** and then navigate to the `doc/startutorialsdata/casting/data` subdirectory of your STAR-CCM+ installation directory.
4. Select `basicGravitySandCastingGeo.sim` and click **Open**.
5. Click **OK**.

Visualize the geometry and save the simulation:

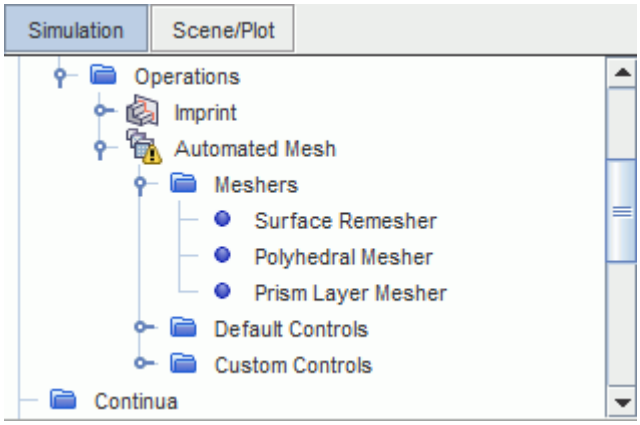
6. Right-click **Scenes** > **Geometry Scene 1** and select **Open**.
7. Select **File** > **Save As...** and save the simulation under the name `basicGravitySandCastingTutorial.sim`.

Generating the Mesh

The mesh models for this case are provided with the initial `.sim` file.

To review the mesh models for the **SprueCup**, **Casting**, and **SandMold** regions:

1. Expand the **Geometry** > **Operations** > **Automated Mesh** > **Meshers** node.



The **SprueCup**, **Casting**, and **SandMold** regions use a polyhedral mesh.

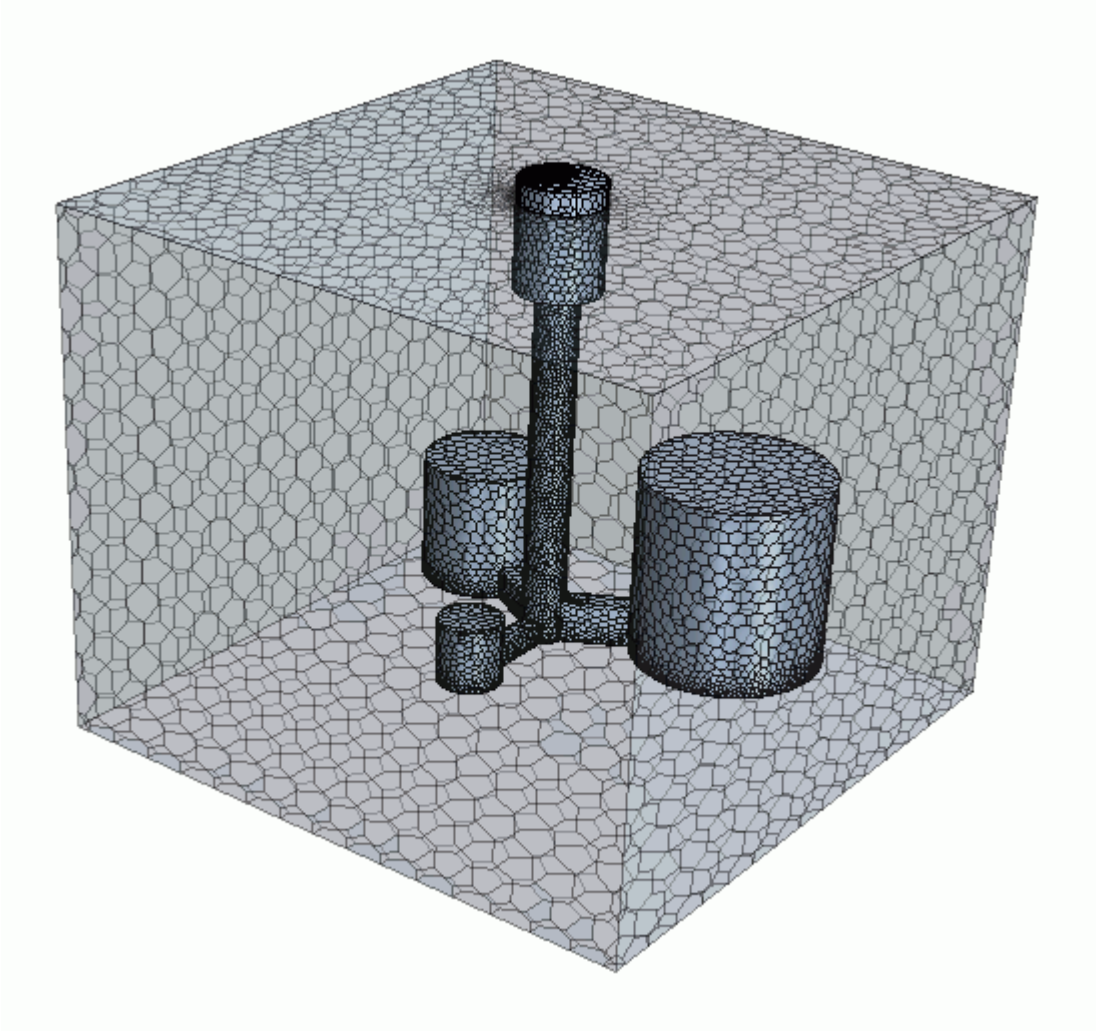
To generate the mesh:

2. Click (**Generate Volume Mesh**) in the **Mesh** toolbar.

   The mesh generation process takes several seconds, depending on the specifications of your computer.

Upon completion of the mesh generation process, make the mesh visible.

3. Right-click the **Scenes** > **Geometry Scene 1** node and select **Apply Representation** > **Volume Mesh**.
4. In the *Geometry Scene 1 - Properties* window, set *Mesh Override* to **Show All Meshes**.
5. Position the parts to obtain a similar view to the one shown below.



6. Save the simulation.

Setting Up the Physics Models

The casting process is transient. The flow of the melt, degassing of the air that is initially present in the casting region, and the solidification of the melt are all of interest for this case.

Melt Continuum

For the casting and sprue cup regions, you set up an implicit unsteady, laminar, multi-phase model using the segregated VOF flow and multiphase temperature models. To access the casting material database and criteria functions, you additionally select the casting model.

To select the physics models for the melt:

1. Rename the **Continua** > **Physics 1** node to `Casting Physics`.
2. For the **Continua** > **Casting Physics** continuum, select the following models in order:

Group Box	Model
Enabled Models	**Three Dimensional** (Selected automatically)
Time	**Implicit Unsteady**
Material	**Eulerian Multiphase**
	Multiphase Interaction (Selected automatically)
Eulerian Multiphase Model	**Volume of Fluid (VOF)**
	Gradients (Selected automatically)
	Multiphase Equation of State (Selected automatically)
	Segregated Flow (Selected automatically)
Viscous Regime	**Laminar**
Optional Models	**Segregated Multiphase Temperature**
	Gravity
	Casting
	Segregated Time Step
	Cell Quality Remediation

3. Click **Close**.

Mold Continuum

The heat transfer is of interest in the sand mold region. To activate the models for the mold:

4. Right-click the **Continua** node and select **New** > **Physics Continuum**.

   A new physics continuum, **Physics 1**, appears in the model tree.
5. Rename the **Continua** > **Physics 1** node to `Mold Physics`.
6. For the **Continua** > **Mold Physics** continuum, select the following models in order:

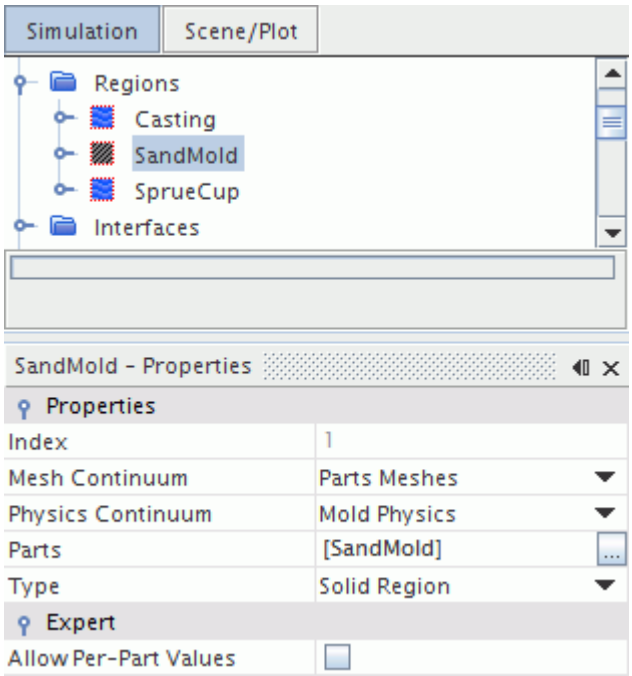
Group Box	Model
Space	**Three Dimensional**
Time	**Implicit Unsteady**
Material	**Solid**
Optional Models	**Segregated Solid Energy**
	Gradients (Selected automatically)
Equation of State	**User Defined EOS**
Optional Models	**Cell Quality Remediation**
	Radiation
Radiation	**Surface-to-Surface Radiation**

Group Box	Model
Radiation Spectrum	**Gray Thermal Radiation**

7. Click **Close**.

Make sure that the correct continuum is assigned to each region:

8. Multi-select the **Regions** > **Casting** and **Regions** > **SprueCup** nodes and make sure that *Physics Continuum* is set to **Casting Physics**.
9. Select the **Regions** > **SandMold** node and set *Physics Continuum* to **Mold Physics**.



10. Save the simulation .

Selecting the Casting Materials

You select the materials for the casting simulation from the casting material database. The aluminum alloy is EN AC-AlSi7Mg0.3 and the mold is made of furan sand.

To open the casting material database dialog:

1. Right-click the **Tools** > **Casting** > **Material Database Editor** node and select **Casting Material DataBase Editor...**

   The dialog can take several seconds to load.

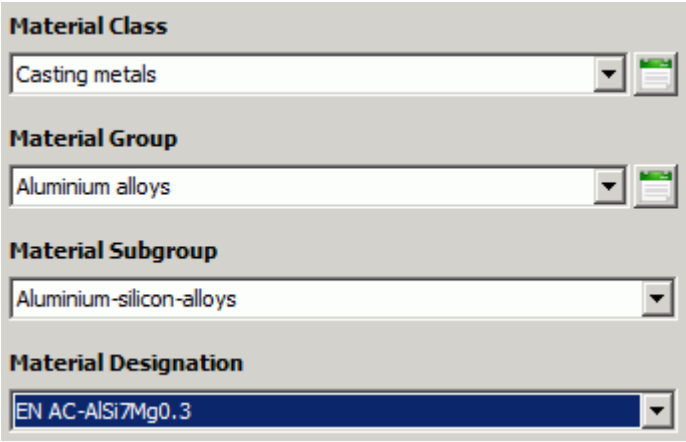
   In the *Disclaimer notice* dialog:
2. Read the disclaimer.
3. Activate **I fully and expressly agree to both sets of provisions specified above and the agreement concerning a contract penalty**.

4. Click **OK**.

   The *STAR-Cast mat* dialog opens with the *Material Selection* panel selected by default. The STAR-CCM+ GUI is not active when the casting material database dialog is opened.

5. Select the casting material:

Property	Setting
Material Class	**Casting metals**
Material Group	**Aluminium alloys**
Material Subgroup	**Aluminium-silicon-alloys**
Material Designation	**EN AC-AlSi7Mg0.3**



To add EN AC-AlSi7Mg0.3 to the collection of materials to be exported to STAR-CCM+:

6. Click the *Export Data* tab.
7. Click **Add Material to Collection**.

   EN AC-AlSi7Mg0.3 is added to the *Collection of Materials*.

Select the material for the sand mold:

8. Click the *Material Selection* tab.
9. Set *Material Class* to **Mould materials**.
10. Click the Icon List to the right of the Material Class menu.
11. In the *Available 'Mould materials'-materials* dialog, select **Furan Resin Sand O&H** and click **Select**.
12. Open the *Export Data* panel and click **Add Material to Collection**.

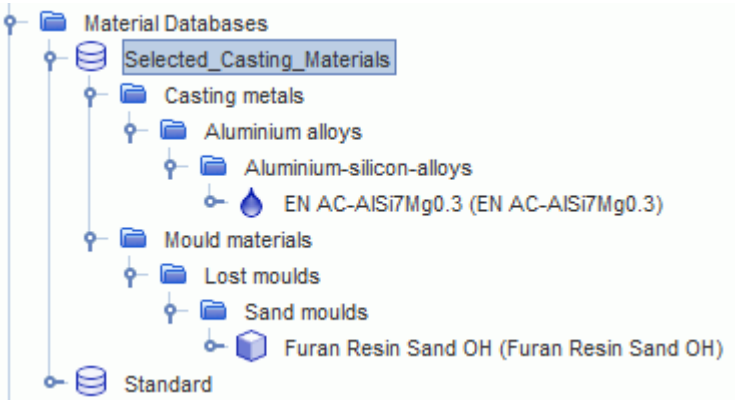
To export this material collection to STAR-CCM+:

13. Click **Send Collection to STAR-CCM+**.

    The casting material database dialog closes and STAR-CCM+ GUI becomes available.

14. Expand the **Tools** > **Material Databases** > **Selected_Casting_Materials** node.

    The materials that are required for this tutorial have been imported into STAR-CCM+.



15. Save the simulation .

Setting Up the Sand Mold Physics Properties

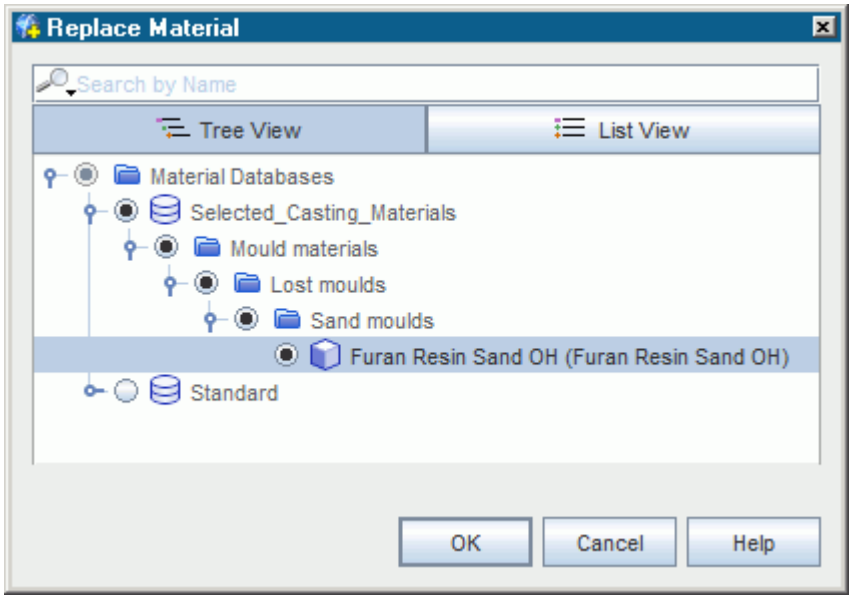
You must set material properties, physical model properties, and initial conditions for the sand mold.

To set up the properties of the sand mold:

1. Right-click the **Continua** > **Mold Physics** > **Models** > **Solid** > **Al** node and select **Replace with...**

   The *Replace Material* dialog appears.

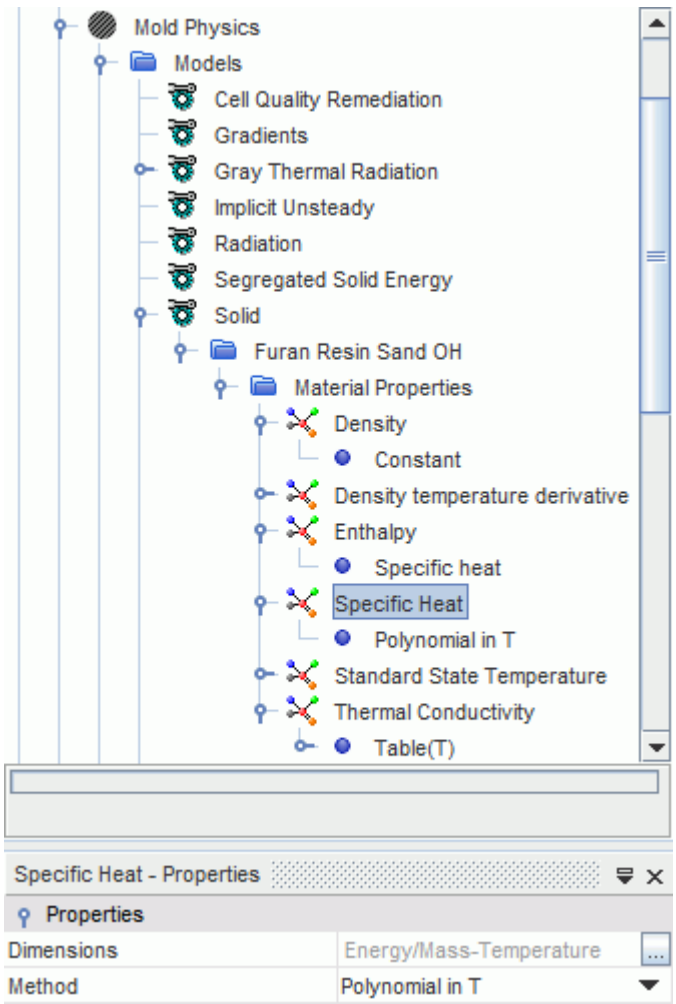
2. Expand the **Material Databases** > **Selected_Casting_Materials** > **Mould materials** > **Lost moulds** > **Sand moulds** node and select **Furan Resin Sand OH (Furan Resin Sand OH)**.



3. Click **OK**.

Set the enthalpy and thermal conductivity of the furan sand using the table from the imported material collection:

4. Select the **Furan Resin Sand OH** > **Material Properties** > **Enthalpy** node and make sure that *Method* is set to **Specific Heat**.
5. Select the **Material Properties** > **Specific Heat** node and set *Method* to **Polynomial in T**.
6. Select the **Furan Resin Sand OH** > **Material Properties** > **Thermal Conductivity** node and set *Method* to **Table(T)**.
7. Select the **Thermal Conductivity** > **Table(T)** > **Table(T)** node and make sure that *Input Table* is set to **Material Database Table**.
8. Make sure that *Table: Data* is set to **ThermalConductivity**.



Set the environment radiation temperature to 30 C:

9. Select the **Mold Physics** > **Models** > **Gray Thermal Radiation** > **Thermal Environments** > **Radiation Temperature** node and set *Radiation Temperature* to `30 C`.

Note: You can type the temperature value together with its unit.

Set the initial temperature of the sand mold to 30 C:

10. Select the **Mold Physics** > **Initial Conditions** > **Static Temperature** node and set *Value* to `30 C`.

11. Save the simulation .

Creating Eulerian Phases and Setting Phase Models

The two phases flowing through the casting and sprue cup regions are the melt and air.

These phases are defined by first creating them under the Eulerian phases model, and then selecting their corresponding material properties.

Melt Phase

The melt phase is created first:

1. Right-click the **Continua** > **Casting Physics** > **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and select **New**.

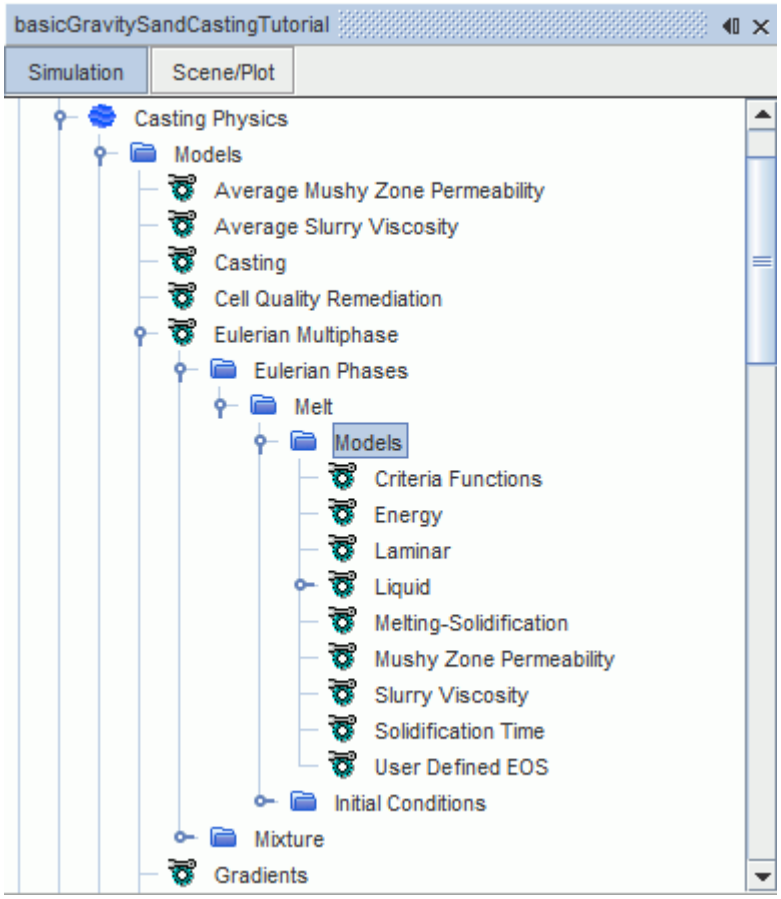
   A new phase, **Phase 1**, appears in the simulation tree.
2. Rename **Phase 1** to `Melt`.
3. Right-click the **Eulerian Multiphase** > **Eulerian Phases** > **Melt** > **Models** node and choose **Select models...**

   In the *Melt Model Selection* dialog, select the following models in order:

Group Box	Model
Enabled Models	**Laminar** (Selected automatically) **Energy** (Selected automatically)
Material	**Liquid**
Equation of State	**User Defined EOS**
Optional Models	**Melting-Solidification** **Slurry Viscosity** **Mushy Zone Permeability** **Solidification Time** When the melt is in a partly solidified state, it is modeled as a slurry in cells where the solid volume fraction is low, and as a mushy zone for higher solid volume fractions. **Criteria Functions** The Criteria Functions model is used solely for post-processing but has to be activated before running the simulation.

4. Click **Close**.

   The following screenshot shows the models that are used in the **Melt** phase.

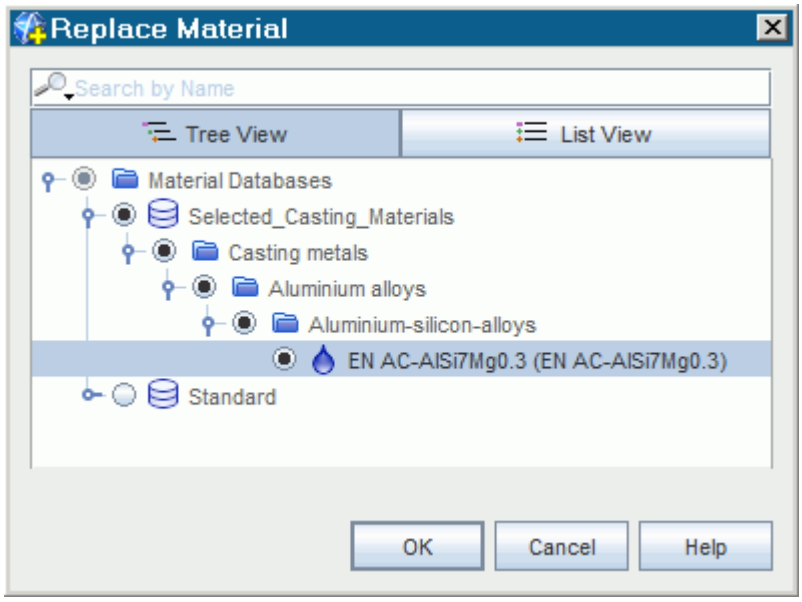


Set the material properties of the melt:

5. Right-click the **Melt** > **Models** > **Liquid** > **H2O** node and select **Replace with...**

   The *Replace Material* dialog appears.

6. Expand the **Material Databases** > **Selected_Casting_Materials** > **Casting metals** > **Aluminium alloys** > **Aluminium-silicon-alloys** node and activate **EN AC-AlSi7Mg0.3 (EN AC-AlSi7Mg0.3)**.



7. Click **OK**.

Use the imported material properties for EN AC-AlSi7Mg0.3.

To export the density table from the Selected_Casting_Materials database:

8. Right-click the **Tools** > **Material Databases** > **Selected_Casting_Materials** > **Casting metals** > **Aluminium alloys** > **Aluminium-silicon-alloys** > **EN AC-AlSi7Mg0.3 (EN AC-AlSi7Mg0.3)** > **Density** > **Table(T)** > **Material Database Table** node and select **Export...**.
9. In the *Save* dialog, enter `rho_T.csv` as **File Name** and click **Save**.
10. Right-click **Tools** > **Tables** and select **New Table** > **File Table**.
11. In the *Open* dialog, select `rho_T.csv` and click **Open**.

To enhance the stability of the simulation, an artificial compressibility of the melt phase is introduced. The density of EN AC-AlSi7Mg0.3 depends on pressure and temperature.

To make the melt phase compressible:

12. Select the **Melt** > **Models** > **User Defined EOS** node and activate *Compressible*.

You create a field function to set the density of EN AC-AlSi7Mg0.3:

13. Right-click **Tools** > **Field Functions** and select **New** > **Scalar**.
14. Rename **User Field Function 1** to **Variable Melt Density**.
15. Set *Function Name* to `VariableMeltDensity`.
16. Set *Dimensions* to **Mass/Volume** by clicking ... (**Custom Editor**).
17. In the *Variable Melt Density - Dimensions* dialog:
    a) Set *Mass* to `1`.
    b) Set *Volume* to `-1`.
    c) Click **OK**.

18. Set *Definition* to `max($MinimumMeltDensity, $InterpolatedMeltDensity + $MeltDensityPressureDerivative*$Pressure)`.

    This field function definition uses other user-defined field functions that you define next using the same procedure as described above:

Field Function Node	Dimensions	Function Name	Definition
Minimum Melt Density	Dimensionless	`MinimumMeltDensity`	1
Melt Density Pressure Derivative	**Mass/Pressure-Volume**	`MeltDensityPressureDerivative`	`1e-5`
Interpolated Melt Density	Dimensionless	`InterpolatedMeltDensity`	`interpolateTable(@Table("rho_T"), "Temperature", LINEAR, "Density", $Temperature)`
Melt Density Temperature Derivative	Dimensionless	`MeltDensityTemperatureDerivative`	`interpolateTable(@Table("rho_T"), "Temperature", LINEAR, "Density", $Temperature + 1) - $InterpolatedMeltDensity`

Note: You can copy and paste the field function, Function Names, and Definitions into STAR-CCM+.

Use this variable density for the melt phase.

19. Select the **Continua** > **Casting Physics** > **Models** > **Eulerian Multiphase** > **Eulerian Phases** > **Melt** > **Models** > **Liquid** > **EN AC-AlSi7Mg0.3** > **Material Properties** > **Density** node and set *Method* to **Field Function**.
20. Select the **Material Properties** > **Density** > **Field Function** node and set *Scalar Function* to **Variable Melt Density**.
21. Select the **Material Properties** > **Density Pressure Derivative** node and set *Method* to **Field Function**.
22. Select the **Material Properties** > **Density Pressure Derivative** > **Field Function** node and set *Scalar Function* to **Melt Density Pressure Derivative**.
23. Select the **Material Properties** > **Density Temperature Derivative** node and set *Method* to **Field Function**.
24. Select the **Material Properties** > **Density Temperature Derivative** > **Field Function** node and set *Scalar Function* to **Melt Density Temperature Derivative**.

To set the other material properties that depend on temperature:

25. Select the **Liquid** > **EN AC-AlSi7Mg0.3** > **Material Properties** > **Dynamic Viscosity** node and set *Method* to **Table(T)**.
26. Select the **Dynamic Viscosity** > **Table(T)** > **Table(T)** node and make sure that *Table: Data* is set to **DynamicViscosity**.
27. Similarly, set the following material properties to the **Table(T)** method: **Fraction Solid Curve** and **Thermal Conductivity**.
28. Select the **Entropy** node and set *Method* to **Isentropic**.

Activate the use of the switching function for the Carman-Kozeny Mushy Zone Permeability model and the Metzner Slurry Viscosity model:

29. Select the **Melt** > **Models** > **Liquid** > **EN AC-AlSi7Mg0.3** > **Material Properties** > **Mushy Zone Permeability** > **Carman-Kozeny** node and make sure that **Use Switching Function** is activated.
30. Select the **Melt** > **Models** > **Liquid** > **EN AC-AlSi7Mg0.3** > **Material Properties** > **Slurry Viscosity** > **Metzner** node and make sure that **Use Switching Function** is activated.

The interesting physics does not always happen when the melt is completely solidified, but rather when a large amount of the melt is solidified and feeding is blocked by the dendrite arm network. You set the critical temperature at which some quantities—temperature gradients, cooling rates, and local solidification times—are of interest. To set the critical temperature for EN AC-AlSi7Mg0.3:

31. Select the **Melt** > **Models** > **Criteria Functions** node and set *Input Units* to **C**.
32. Set *Critical Temperatures* to `[574.0]`

    This critical temperature is between the solidus temperature (526.804 C) and liquidus temperature (608.47 C) of the melt.

To stabilize the simulation, under-relax the solidification process:

33. Select the **Melt** > **Models** > **Melting-Solidification** node and set *Under-Relaxation Factor* to `0.7`.
34. Select the **Continua** > **Casting Physics** > **Models** > **Average Slurry Viscosity** node and set *Slurry Viscosity Under-Relaxation Factor* to `0.5`.
35. Select the **Continua** > **Casting Physics** > **Models** > **Average Mushy Zone Permeability** node and set *Mushy Zone Permeability Under-Relaxation Factor* to `0.5`.
36. Set the *Max. Porous Viscous Resistance* to `70000 kg/m^3-s`

The flow in the melt stops when the solid fraction is near 0.9. Set this flow stop value:

37. Select the **Melt** > **Models** > **Melting-Solidification** node and set *Flow Stop Relative Solid Fraction* to `0.95`.
38. Select the **Continua** > **Casting Physics** > **Models** > **Melting-Solidification Flow Stop** node and set *Flow Stop Solid Fraction* to `0.95`.

    The properties of the melt are determined the *Flow Stop Relative Solid Fraction*. The *Flow Stop Solid Fraction* is set to a higher value because the slurry viscosity and mushy zone permeability models already decelerate the flow for lower solid volume fractions. The flow stop functionality is only required when creeping flow is to be prevented, or when the solid needs to withstand very high pressure gradients. Hence, the *Flow Stop Solid Fraction* is generally set to a value close to 1.0.

Air Phase

Create the air phase:

39. Right-click the **Continua** > **Casting Physics** > **Models** > **Eulerian Multiphase** > **Eulerian Phases** node and select **New**.

    A new phase, **Phase 1**, appears in the simulation tree.

40. Rename **Phase 1** to `Air`.

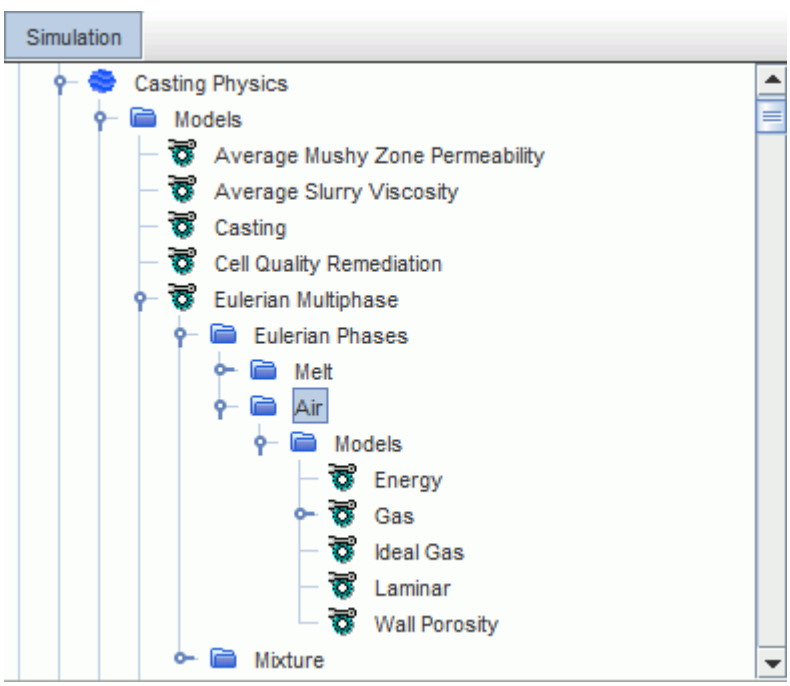
41. Right-click the **Eulerian Multiphase** > **Eulerian Phases** > **Air** > **Models** node and choose **Select models...**

    In the *Air Model Selection* dialog, select the following models.

Group Box	Model
Enabled Models	**Laminar** (Selected automatically) **Energy** (Selected automatically)
Material	**Gas**
Equation of State	**Ideal Gas**
Optional Models	**Wall Porosity** Degassing occurs when the melt flows into the casting region and air is pushed into the sand mold.

42. Click **Close**.

    The following screenshot shows the models that are used in the **Air** phase.



Activate Density Limiting for the **Air** phase:

43. Select the **Eulerian Phases** > **Air** > **Models** > **Ideal Gas** node and activate *Density Limiting*.

44. Select the **Air** > **Models** > **Ideal Gas** > **Density Limits** node and set *Cut-off Pressure* to `10000.0 Pa`.

Phase Interaction

Set up the phase interaction between the melt and the air:

45. Right-click the **Continua** > **Casting Physics** > **Models** > **Multiphase Interaction** > **Phase Interactions** node and select **New**.

    A new phase interaction, **Phase Interaction 1**, appears in the simulation tree.
46. Rename **Phase Interaction 1** to `Casting_Phase_Interaction`.
47. Right-click the **Casting_Phase_Interaction** > **Models** node and choose **Select models...**

    In the *Casting_Phase_Interaction Model Selection* dialog, select the following models.

Group Box	Model
Phase Interaction Topology	**VOF-VOF Phase Interaction**
Optional Models	**Surface Tension Force** **Multiphase Material** (activated automatically) **Interface Momentum Dissipation**

48. Click **Close**.
49. Select the **Casting_Phase_Interaction** > **Models** > **VOF-VOF Phase Interaction** node and set *Primary Phase* to **Melt**.
50. Set *Secondary Phase* to **Air**.

Create a field function to set a variable surface tension. The surface tension has a constant value of 0.9 N/m but approaches zero when the melt solidifies. Solidification of the melt occurs when the solid volume fraction reaches the flow stop solid volume fraction.

51. Right-click **Tools** > **Field Functions** and select **New** > **Scalar**.
52. Rename the new field function to `Surface Tension`.
53. Select the **Surface Tension** node and set the following properties:

Property	Setting
Function Name	`SurfaceTension`
Dimensions	**Force/Length**
Definition	`0.9 * (0.5 - atan(1000 * ($ {RelativeSolidVolumeFractionMelt} - 0.85)) / 3.14159265)`

54. Select the **Casting_Phase_Interaction** > **Models** > **Multiphase Material** > **Multiphase Material** > **Material Properties** node and set the following properties:

Node	Property	Setting
Surface Tension	**Method**	**Field Function**
└ **Field Function**	**Scalar Function**	**Surface Tension**

55. Save the simulation .

Setting Up the Casting Properties

Here, you set up model properties such as the direction of gravity and the initial conditions of the simulation.

To improve the prediction of the melt-air interface between the VOF phases, set the VOF sharpening factor:

1. Select the **Continua** > **Casting Physics** > **Models** > **Volume of Fluid (VOF)** node and set *Sharpening Factor* to `0.2`.

Gravity acts in the -Z direction.

2. Select the **Continua** > **Casting Physics** > **Reference Values** > **Gravity** node and make sure that *Value* is set to `[0.0, 0.0, -9.81] m/s^2`.

The casting region is initially at a temperature of 30 C and filled with air. To set up the initial properties of the casting continua:

3. Select the **Continua** > **Casting Physics** > **Initial Conditions** > **Static Temperature** node and set *Value* to `30 C`.
4. Select the **Continua** > **Casting Physics** > **Initial Conditions** > **Volume Fraction** node and set *Value* to `[0.0, 1.0]`

   Note: The first entry in the *Value* array denotes the melt fraction, and the second entry denotes the air fraction.

5. Save the simulation .

Setting Region and Boundary Conditions

Set the conditions in the SprueCup, Casting, and SandMold regions.

The sprue cup is initially filled with air. The outlet surface is set to atmospheric pressure and a constant temperature that is lower than the initial melt temperature, yet significantly hotter than room temperature. The mass flow into the pour cup yields a filling time of around 5–6 s.

Outlet

Specify the outlet properties in the sprue cup:

1. Edit the **Regions** > **SprueCup** > **Boundaries** > **Outlet** node and set the following properties:

Node	Property	Setting
Physics Values		
└ **Static Temperature**	*Value*	`30 C`
└ **Volume Fraction**	*Value*	`[0.0, 1.0]`

Mass Flow Inlet

During the first 0.05 s of the simulation, the mass flow rate is linearly ramped up to 0.35 kg/s and remains constant for 5.7 s. Then, the mass flow rate is linearly ramped down to 0 kg/s. This corresponds to an

approximate filling time of 5.8 s. The melt is not poured into the domain after this interval. This time-dependent inlet melt mass flow is provided as a table.

Import the inlet mass flow table:

2. Right-click the **Tools** > **Tables** node and select **New Table** > **File Table**.
3. In the *Open* dialog, select **inletMassFlow.csv** and click **Open**.

The volume fraction of melt at the inlet during the filling is 1.0. When the filling is complete, this fraction is set to 0.0. Use a table to specify this condition.

Import the inlet volume fraction table:

4. Right-click **Tools** > **Tables** and select **New Table** > **File Table**.
5. In the *Open* dialog, select **inletVolumeFraction.csv** and click **Open**.
6. Edit the **SprueCup** > **Boundaries** > **Inlet** node and set the following properties:

Node	Property	Setting
Physics Values		
└ **Total Temperature**	*Value*	`690 C`
└ **Mass Flow Rate**	*Method*	**Table (time)**
└ **Table (time)**	*Table*	**inletMassFlow**
	Interpolation	**Linear**
	Table: Data	**Mass Flow Rate**
└ **Volume Fraction**	*Method*	**Composite**
└ **Composite** > **Melt**	*Method* └	**Table (time)**
└ **Table (time)**	*Table*	**inletVolumeFraction**
	Interpolation	**Step**
	Table: Data	**Melt**
└ **Composite** > **Air**	*Method* └	**Table (time)**
└ **Table (time)**	*Table*	**inletVolumeFraction**
	Interpolation	**Step**
	Table: Data	**Air**

Interface Conditions

Set the interface conditions between the sprue cup and the sand mold as well as between the casting region and the sand mold. You specify the degassing conditions on both the interface and the wall. When the interface does not completely match the wall, the non-matching faces are impermeable to the gas.

7. Expand the **Regions** > **SprueCup** > **Boundaries** and the **Regions** > **Casting** > **Boundaries** nodes.

8. Multi-edit the **Walls**, the **Walls [SandMold/SprueCup]**, the **Interface_SandMold**, the **Interface_SandMold [Casting/SandMold]**, the **Bottom_Interface_SandMold**, and the **Bottom_Interface_SandMold [Casting/SandMold 2]** nodes and set the following properties:

Node	Property	Setting
Phase Conditions		
└ **Air** > **Physics Conditions** > **Wall Porosity Specification**	*Method*	**Specified**
└ **Air** > **Physics Values** > **Porous Viscous Resistance**	*Value*	`50000.0 m/s`
└ **Casting_Phase_Interaction** > **Physics Values** > **Contact Angle**	*Value*	`115 deg`

Sand Mold Region

Set the radiation conditions such that the hot sand mold only radiates to the environment of the sand mold and not within the fluid domain.

9. Edit the **Regions** > **SandMold** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Radiation Transfer Option**	*Internal Radiation Transfer*	Deactivated
Physics Values		
└ **Patch/Face Proportion**	*Patch/Face Proportion*	`20.0`

Sand Mold Wall Boundaries

Set the thermal conditions:

10. Expand the **Regions** > **SandMold** > **Boundaries** node.
11. Multi-edit the **Sides** and the **Top** nodes and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Thermal Specification**	*Condition*	**Environment**
Physics Values		
└ **Heat Transfer Coefficient**	*Value*	`10 W/m^2-K`

12. Save the simulation .

Setting Solver Parameters and Stopping Criteria

To enhance the convergence and stability of the simulation, you use the Segregated Time Step model. The Segregated Time Step model automatically adjusts the time step during run based on the CFL criteria that you specify.

To specify the properties of the Segregated Time Step model:

1. Select the **Continua** > **Casting Physics** > **Models** node and set the following properties:

Node	Property	Setting
Segregated Time Step	*Start Time Step*	`5.0E-5 s`
	Min. Time Step	`5.0E-6 s`
└ **Flow**	*Max. Condition Limit*	`0.35`
	Smoothing Steps	`6.0`
└ **VOF**	*Max. Condition Limit*	`0.25`
	Smoothing Steps	`4.0`

Deactivate the time step criteria based on the energy change rate and the transport of the phase change interface:

2. Multi-edit the **Energy** and **Melting-Solidification** nodes and set the *Max. Condition Limit* to `0.0`.

Set an upper limit for the Segregated Time Step and the under-relaxation factor for pressure:

3. Edit the **Solvers** node and set the following properties:

Node	Property	Setting
Implicit Unsteady	*Time-Step*	`0.1 s`
Segregated Flow > **Pressure**	*Under-Relaxation Factor*	`0.6`

Set the number of inner iterations per time step and the physical run time:

4. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Inner Iterations	*Maximum Inner Iterations*	`10`
Maximum Physical Time	*Maximum Physical Time*	`8.0 s`
Maximum Steps	*Enabled*	**Deactivated**

5. Save the simulation .

Creating a Solution History File

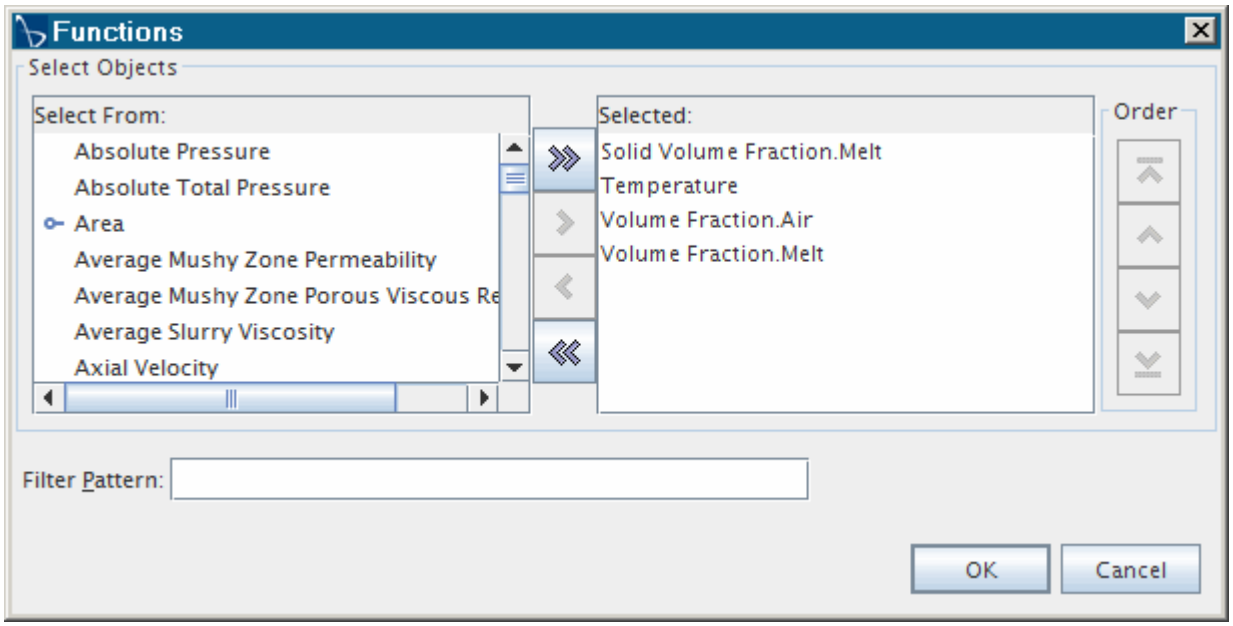
Create a solution history file to save the solved variables at different stages of the simulation.

You will use these saved states at a later stage for post-processing, the filling, and the solidification:

1. Right-click the **Solution Histories** node and select **New...**.
2. In the *Save* dialog, set File Name to `Cast`, and click **Save**.

   The **Solution Histories** > **Cast** node appears in the simulation tree.

3. Select the **Cast** node and click ... (**Custom Editor**) for the *Functions* property.
4. In the *Functions* dialog, select the following variables: **Solid Volume Fraction**, **Temperature**, and **Volume Fraction**.



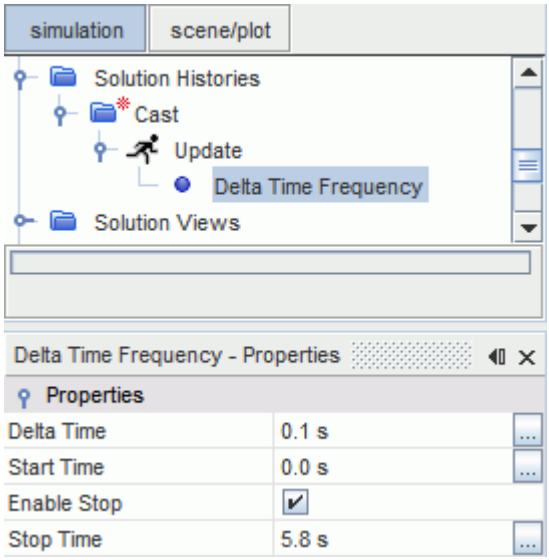
5. Click **OK**.

During the filling, the variables are saved at 0.1 s intervals:

6. Select the **Solution Histories** > **Cast** > **Update** node and set *Trigger* to **Delta Time**.
7. Select the **Update** > **Delta Time Frequency** node and set *Delta Time* to `0.1 s`.

   The filling starts at 0 s.

8. Set *Stop Time* to `5.8 s`.
9. Activate *Enable Stop*.



10. Save the simulation .

Visualizing the Solution

Create a scalar scene to display the melt volume fraction colored with temperature in the casting and sprue cup regions.

To create a melt volume fraction threshold:

1. Right-click **Derived Parts** and select **New Part** > **Threshold...**
2. In the *Create Threshold* dialog:
   a) Set *Scalar* to **Volume Fraction** > **Volume Fraction of Melt**.
   b) Set *Extraction Mode* to **All Above**.
   c) Set *Threshold Value* to `0.5`.
   d) Click **Create** and **Close**.

To create the melt fraction scene:

3. Create a scalar scene.
4. Rename **Scalar Scene 1** to `Melt Fraction Scene`.
5. Click **Scene/Plot**.
6. Select the **Melt Fraction Scene** > **Displayers** > **Outline 1** and set the following properties:
   a) Activate *Surface*.
   b) Set *Color Mode* to **Constant**.
   c) Set *Opacity* to `0.1`.
7. Select the **Melt Fraction Scene** > **Displayers** > **Outline 1** > **Parts** node and set *Parts* to **Regions** > **SandMold**.
8. Select the **Melt Fraction Scene** > **Displayers** > **Scalar 1** > **Parts** node and set *Parts* to **Derived Parts** > **Threshold**.

9. Select the **Melt Fraction Scene** > **Displayers** > **Scalar 1** > **Scalar Field** node and set **Function** to **Temperature**.
10. Set *Units* to **C**.
11. Save the simulation .

Running the Simulation

You are now ready to run the simulation.

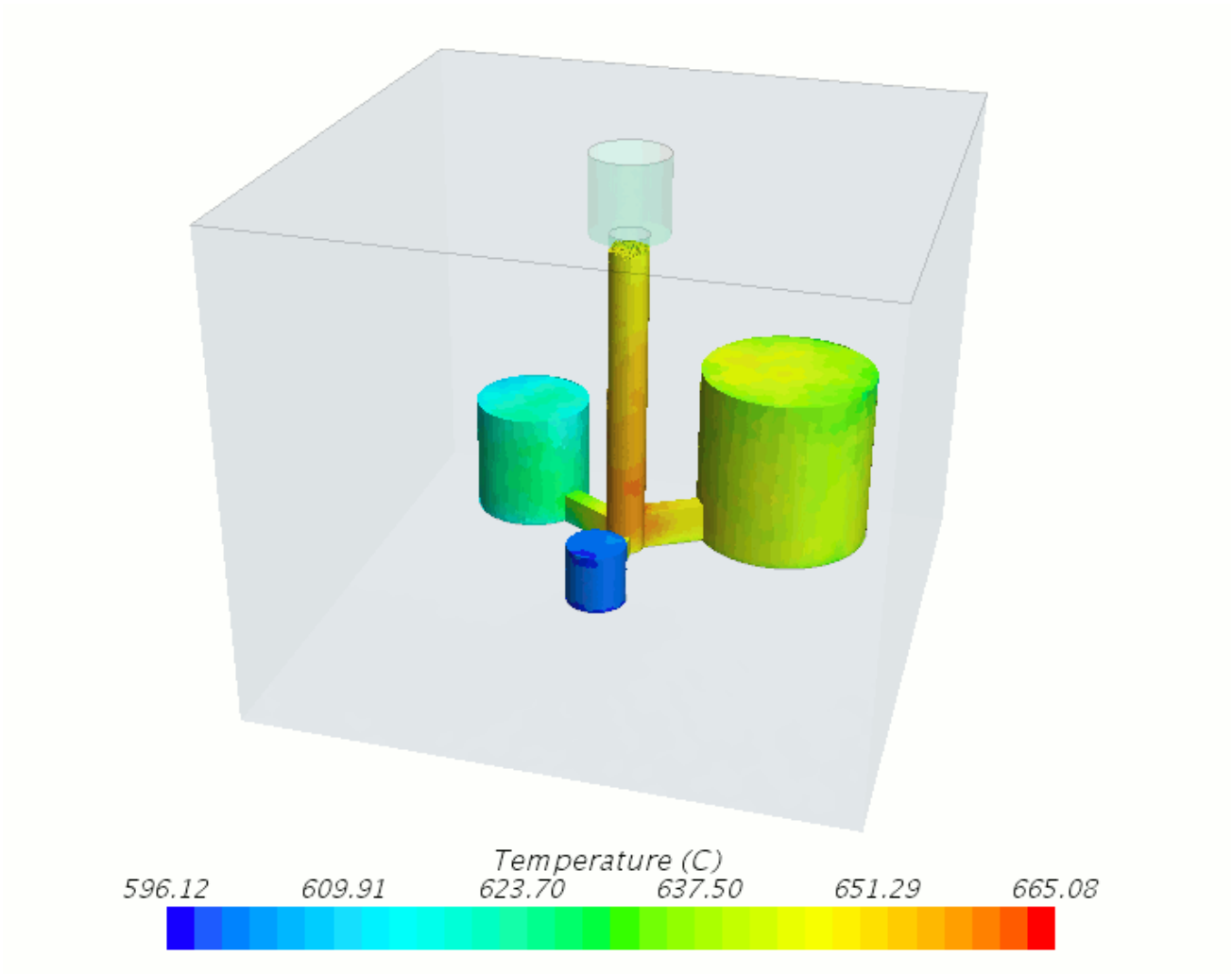
1. Click **Run** in the toolbar.

   A Residuals plot is created automatically, and shows the progress of the solver.
2. When the solver has finished running, save the simulation .

Visualizing the Results

Visualize the results after the simulation is completed.

The melt fraction scene after a simulation time of 8 s is shown below.



When the simulation stops after a physical time of 8 s, the melt has not completely solidified.

The tutorial stops here to keep the computer runtime to a minimum. If you wish, you can continue the run to achieve full solidification by increasing the physical run time of the simulation.

Creating a Melt Filling Animation

The solution history file contains the solution data from both the filling and solidification runs. Solution views are used to access this data for post-processing.

Add a solution time annotation to the scene to track the solution time as the scene animates:

1. Open the melt fraction scene to display the melt threshold.
2. Drag and drop the **Tools** > **Annotations** > **Solution Time** node onto the bottom left of the *Melt Fraction Scene*.

Create and set up a solution view for the animation:

3. Right-click the **Solution Histories** > **Cast** node and select **Create Recorded Solution View**.
4. In the opening dialog, select **Yes**.

The **Solution Views** > **Cast** node appears in the simulation tree.

5. Select the **Solution Views** > **Cast** > **Animation** node and set *Animation Mode* to **Solution Time**.

   The **Solution Time Animation** sub-node appears.

6. Select the **Solution Time Animation** node and set the following properties:

Property	Setting
Iteration Mode	**Cycle Time**
Start Solution Time	`0.0`
End Solution Time	`5.8`

7. Select the **Solution Time Animation** > **Cycle Time** node and set *Value* to `6.8 s`.
8. To set the *Representation* property of all displayers and solution history annotations of the melt fraction scene to **Cast**, drag and drop the **Solution Views** > **Cast** node onto the *Melt Fraction Scene*.

To create the melt filling animation:

9. Click (**Write a movie of the animation in the current scene**) in the **Animation** toolbar.
10. In the *Write animation* dialog: set *Animation Length* to `6.8`.
    a) Set *Resolution* to a resolution of your choice.
    b) Set *File Name* to `meltFilling.avi`.
    c) Click **Save** to write the animation to disk.
11. Play back the animation using a player of your choice.

Summary

This tutorial demonstrated the setup, running, and post-processing of a basic gravity sand casting analysis.

This tutorial has introduced the following STAR-CCM+ features:

- Setting up a basic sand casting simulation.
- Importing material properties from the casting material database.
- Interpolating tabular values with a field function.
- Creating a melt filling animation using a solution history file.

STAR-Cast: Investment Casting

This tutorial demonstrates the workflow for setting up an investment casting simulation.

Investment casting is an industrial process, which is also called lost-wax casting or lost-mold casting, because the cast mold is destroyed when the cast part is extracted. For precision casting, a wax model is dipped into a ceramic based slurry. The ceramic slurry dries and forms the mold for casting which is known as the shell. This process is repeated until the desired thickness is reached. The wax is then removed in an autoclave.

STAR-CCM+ provides a dedicated Offset Mesher model for the mesh that is required by this type of simulation. The Offset Mesher creates the geometry that represents the shell. The Offset Mesher mimics the manufacturing process and creates a mesh that is based on local curvature. By taking the shell into account for this type of simulation, you can simulate the full conjugate heat transfer of the melt interacting with the shell.

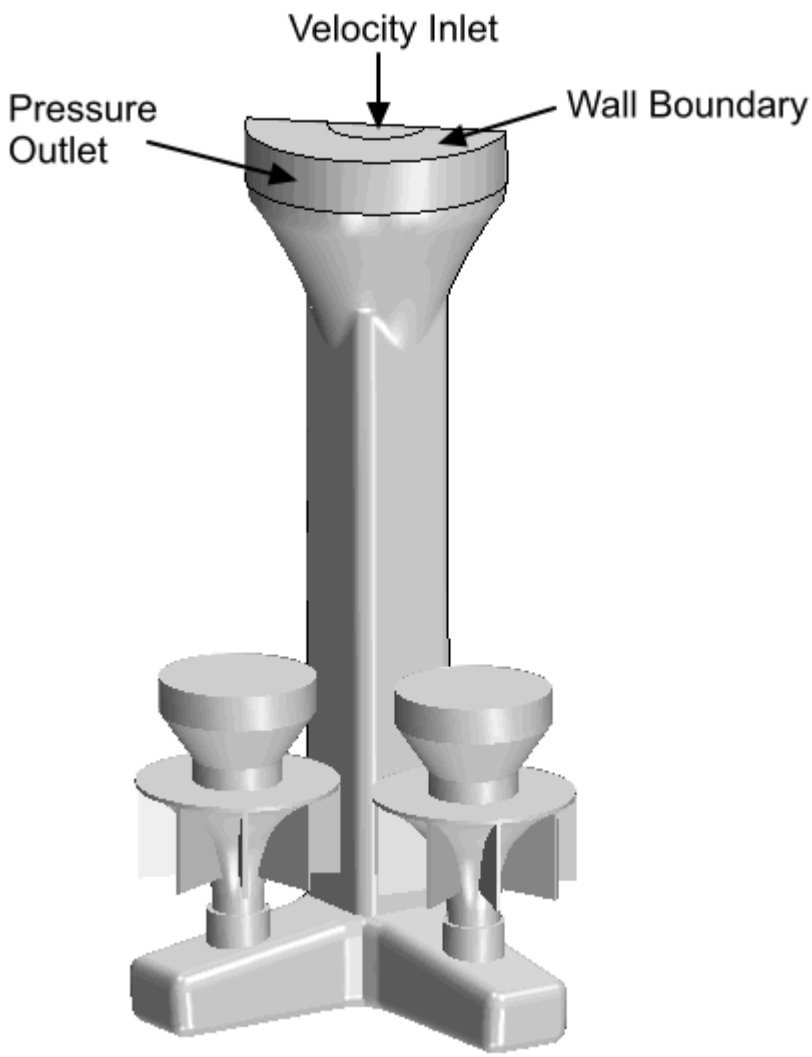
Note: The Offset Mesher is available using a macro. To obtain the macro, contact your local support representative.

The investment casting simulation is set up as a multiphase VOF flow simulation. The geometry consists of two regions, one represents the fluid cavity, and the other defines the top which contains the velocity inlet and pressure outlet. A third region is created to represent the shell.

Ideally, the multi-region volume mesh must have conformal interfaces between the regions. Conformal interfaces are created automatically when the parts are assigned to regions. All regions use the polyhedral volume mesher with the addition of the Offset Mesher for the shell region. To accurately compute the gradients at the wall and at the interfaces, create prism layers at these locations.

For the investment casting simulation here, the melt (a titanium alloy) enters the domain with a temperature of 1620 °C through a velocity inlet boundary. The inlet velocity is a function of time and is described using a table. When the melt impacts on the surface of the shell, it loses heat through the mold to the surroundings. The shell has an initial temperature of 900 °C and is bounded by convective wall boundaries. The hot melt cools off at the shell and solidifies.

The tutorial starts with the import of the investment casting geometry in Parasolid format. The geometry is shown below:



As the geometry is symmetric about a center plane, the simulation only requires half of the fluid cavity. The second half of cavity is accounted for by applying the symmetry plane boundary condition.

The melt is centrally poured into the mold at the top of the geometry. The air inside the mold escapes the geometry through the gap surrounding the melt. Simulating this condition would require an inflow boundary enclosed by an outflow boundary, on the same plane. Placing an inflow and outflow boundary together is numerically very unstable and must be avoided. Therefore, these two flow boundaries are separated by a wall boundary. The rounded shape is recommended as it avoids sharp corners and edges, which would reduce the robustness of the solution.

Note: You must use a double precision version of STAR-CCM+ for this tutorial.

Contents:

Prerequisites
Importing the Geometry
Generating the Mesh

Setting up the Investment Casting Simulation
Defining the Casting Process
Selecting the Casting Materials
Setting Initial Conditions
Setting Boundary Conditions
Setting Interface Properties
Setting Physics Models
Setting Solver Parameters
Setting up Criteria Functions
Setting Up Post Processing
Running the simulation
Visualizing the solution
Summary

Prerequisites

The instructions in this tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Techniques	Associated Tutorials
Using visualization tools, scenes, and plots	Introduction

Importing the Geometry

The meshing process starts from a CAD geometry in Parasolid format. The imported geometry represents the fluid cavity bounded by the inner walls of the shell, and the top, which contains the velocity inlet and pressure outlet.

To import the investment casting geometry of the sprue cup, the gating, the wheels, and the top:

1. Create a directory for the tutorial, called `investmentCasting`.
2. Start up a double precision version of STAR-CCM+ and create a new simulation by clicking **File** > **New Simulation...**
3. Select **File** > **Import** > **Import Surface Mesh...**. Navigate to the `doc/startutorialsdata/casting/data` subdirectory of your STAR-CCM+ installation directory. Select the file `castingAssembly.x_t` from the selection list and click **Open**.
4. In the *Import Surface Options* dialog, make sure that *Create Part Contacts from Coincident Entities* is activated. Leave all other settings at their defaults and click **OK**.

   The imported mesh appears in *Geometry Scene 1*, the imported geometry appears under the **Geometry** > **Parts** node, and strong in-place contacts are created between the geometry parts.
5. Expand the **Parts** > **Cast** > **Surfaces** node and inspect the imported surfaces.
6. Save the simulation as `investmentCastingTutorial.sim`.

Generating the Mesh

This simulation requires a volume mesh for the fluid interior, a volume mesh which contains the inlet and outlet boundaries, and a volume mesh for the solid shell.

The imported surface for the investment cast part defines the outer surface of the fluid interior. As the quality of this surface is not adequate for subsequent operations, you remesh the surface to improve its quality. The mesh for the shell region is then grown from the remeshed surface of the fluid interior.

The following sections describe the sequence of operations for setting up mesh continua, regions, and boundaries, then generating the volume meshes.

Before generating the mesh, load the Offset Mesher model macro:

1. Select **File** > **Macro** > **Play Macro...**.
2. Navigate to and select the macro, `OffsetMesher.java`. You can get this macro from your Siemens PLM Software support representative.
3. Click **Open**.

   A mesh continuum is created which has the following meshing models selected:

   - Polyhedral Mesher
   - Offset Mesher
   - Embedded Thin Mesher

Contents:

Creating the Cast Mesh
Creating the Top Mesh
Creating the Mesh for the Shell Region
Generating the Volume Mesh

Creating the Cast Mesh

Follow this procedure to modify the mesh continuum which represents the cast part.

The existing mesh continuum contains the Offset Mesher model which grows the mesh for the shell region from the outer surface of the cast mesh. Before setting up the mesh for the shell region, the cast mesh must be defined and mesh settings applied.

Before creating the cast mesh continuum, assign parts to regions:

1. Multi-select the **Geometry** > **Parts** > **Cast** and **Top** nodes, right-click one of the nodes and select **Assign Parts to Regions...**
2. In the *Assign Parts to Regions* dialog:
   a) Select **Create a Region for Each Part.**
   b) Select **Create a Boundary for Each Part Surface**.
   c) Accept **Create One Feature Curve for All Part Curves**.
3. Click **Apply** and then **Close**.

   Interfaces are automatically created from the part contacts.

Set up a cast mesh continuum.

4. Rename **Continua** > **Mesh 1** to `Cast`.
5. Right-click the **Cast** > **Models** node and select **Select Meshing Models...**.
6. In the *Cast Model Selection* dialog:
   a) Keep the meshing models that are already selected.
   b) Activate **Surface Remesher**.
   c) Activate **Prism Layer Mesher**.
   d) Click **Close**.
7. Select the **Cast** > **Models** > **Surface Remesher** node and deactivate **Do Curvature Refinement**.
8. Select the **Models** > **Embedded Thin Mesher** node and activate **Customize Thickness Threshold**.
9. Expand the **Cast** > **Reference Values** node and set the following properties:

Node	Property	Value
Base Size	*Value*	`0.0025 m`
Offset Values	*Surface Size*	`0.005 m`
	Thickness	`0.007 m`
	Concavity Factor	`30`
	Convexity Factor	`1`
Thin Solid Thickness	*Size Type*	**Absolute**
⌊ **Absolute Size**	*Value*	`15.0 mm`

Define the boundaries on the Cast region.

10. Expand the **Regions** > **Cast** > **Boundaries** node and set the following properties:

Node	Property	Value
Symmetry	*Type*	**Symmetry Plane**
Cast-Top Interface		
⌊ **Mesh Conditions**		
⌊ **Customize Prism Mesh**	*Customize Prism Mesh*	**Disable**

Creating the Top Mesh

Follow this procedure to create a separate mesh continuum for the Top part, then define the Top region boundaries.

The Top part does not use the Offset Mesher model and requires a separate mesh continuum from the cast part. In this case, the prism layer mesher–which is normally deactivated on inlet and outlet boundaries–is maintained. Maintaining a constant thickness prism layer along the coincident wall, inlet, and outlet boundaries ensures that good quality cells are created throughout the Top part.

To create a mesh continuum for the Top part:

1. Create a copy of the **Cast** mesh continuum and rename it **Top**.
2. Right-click the **Top** > **Models** node and select **Select Meshing Models...**.
3. In the *Top Model Selection* dialog, deactivate *Offset Mesher*.
   The *Embedded Thin Mesher* model is automatically deactivated.
4. Click **Close**.

Define the boundaries on the Top Region.

5. Select the **Regions** > **Top** node and set *Mesh Continuum* to **Top**.
6. Expand the **Top** > **Boundaries** node and set the following properties:

Node	Property	Setting
Inlet	*Type*	**Velocity Inlet**
⌊ **Mesh Conditions**		
⌊ **Custom Surface Size**	*Custom Surface Size*	Activated
⌊ **Customize Prism Mesh**	*Customize Prism Mesh*	**Specify Custom Values**
⌊ **Mesh Values**		
⌊ **Surface Size**		
⌊ **Relative Target Size**	*Percentage of Base*	`40.0`
Pressure Outlet	*Type*	**Pressure Outlet**
⌊ **Mesh Conditions**		
⌊ **Customize Prism Mesh**	*Customize Prism Mesh*	**Specify Custom Values**
Symmetry	*Type*	**Symmetry Plane**
Top-Cast Interface		
⌊ **Mesh Conditions**		
⌊ **Customize Prism Mesh**	*Customize Prism Mesh*	**Disable**

Creating the Mesh for the Shell Region

Follow this procedure to define the shell mesh, region, and boundaries.

The shell mesh is grown from the interface between the Shell region and the Cast region, and uses mesh values that you specify in the Cast mesh continuum.

Create a region for the shell mesh:

1. Create a region and rename it `Shell`.
2. Select the **Regions** > **Shell** node and set *Type* to **Solid Region**.

3. Expand the **Shell** > **Mesh Conditions** node and set the following Properties:

Node	Property	Setting
Offset Options	*Mark as Offset Region*	Activated. In the dialog, click **Yes**.
	Input Boundaries	**Cast-Shell Interface**
Customize Thin Mesher Parameters	*Customize Thin Mesher Parameters*	**Custom**

An **Interfaces** > **Offset interface on Cast-Shell Interface** node appears.

4. Expand the **Regions** > **Shell** > **Mesh Values** node and set the following Properties:

Node	Property	Setting
Thin Mesher Layers	*Number of Layers*	`3`
Thin Solid Thickness	*Size Type*	**Absolute**
∟ **Absolute Size**	*Value*	`12 mm`

Normally, prism layers are only required on the fluid side of an interface. In this case, due to the high temperature gradients at the interface, prism layers are created on both sides. Prism layers accurately predict the heat transfer at the interface between the liquid melt and the solid shell mesh. To create prism layers at the interface:

5. Expand the **Interfaces** > **Offset interface on Cast-Shell Interface** > **Mesh Conditions** node.
6. Select the **Interface Prism Layer Option** node and activate *Grow Prisms From Interface*.
7. Expand the **Regions** > **Shell** > **Boundaries** > **Offset Extrusion** > **Mesh Conditions** node.
8. Select the **Customize Prism Mesh** node and set *Customize Prism Mesh* to **Disable**.

Generating the Volume Mesh

You can now generate the volume mesh.

The **Cast** and **Top** mesh continua are set up and ready to mesh. The **Shell** region is grown from the **Offset interface on Cast-Shell Interface** on the outer surface of the **Cast** region.

Generate the volume mesh:

1. Select **Mesh** > **Generate Volume Mesh**.

To visualize the volume mesh:

2. Click *Scene/Plot*.
3. Select the **Displayers** > **Geometry 1** > **Parts** node.
4. In the *Parts - Properties* window, click **custom editor** ... next to *Parts*.
5. In the *Parts - Parts* dialog, select **Regions** and de-select **Parts**. Then click **OK**.
6. Return to *Simulation*.
7. Drag and drop the **Representations** > **Volume Mesh** node into the currently open *Geometry Scene 1*.
8. Save the simulation.

Setting up the Investment Casting Simulation

A dedicated Casting User Interface (CUI) guides you through the setup of the investment casting simulation.

Before you can use the CUI, you need to include it in the STAR-CCM+ toolbar.

For more information on how to access the CUI, see Accessing the Casting User Interface in the **User Guide** > **Modeling Physics** > **Modeling Casting** > **The Casting User Interface**.

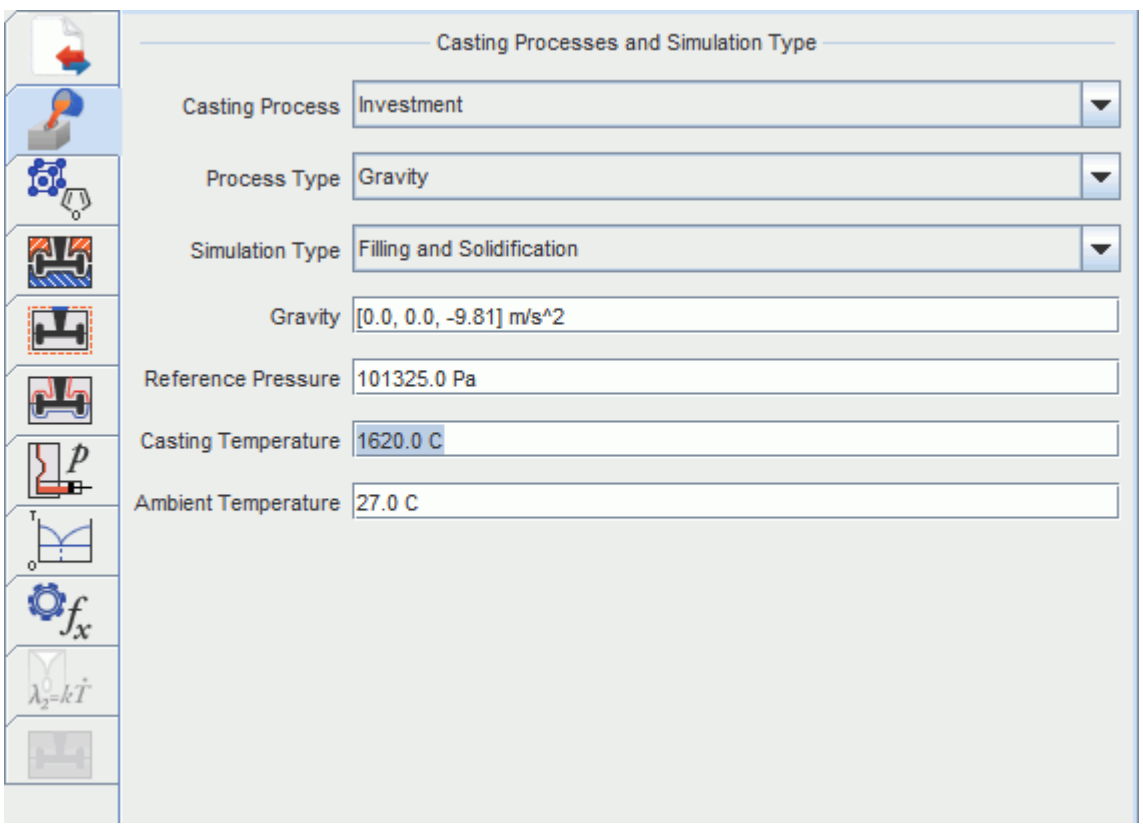
Defining the Casting Process

In the *Casting Processes and Simulation Type* panel of the CUI, you define the casting process.

Before setting up the casting process, close all open scenes in the *Graphics* window.

To specify the investment casting process including filling and solidification:

1. In the **Casting** toolbar, click the STAR-Cast User Interface.
2. The CUI opens up with the *Condition Settings Import/Export* panel. Select the *Casting Processes and Simulation Type* panel.
   a) Set *Casting Process* to **Investment**.
   b) Set *Process Type* to **Gravity**.
   c) Set *Simulation Type* to **Filling and Solidification**.
   d) Set *Casting Temperature* to `1620 C`.
   e) Leave the settings for *Gravity, Reference Pressure, and Ambient Temperature* unchanged.



Selecting the Casting Materials

You must select the materials for the melt and the shell mold.

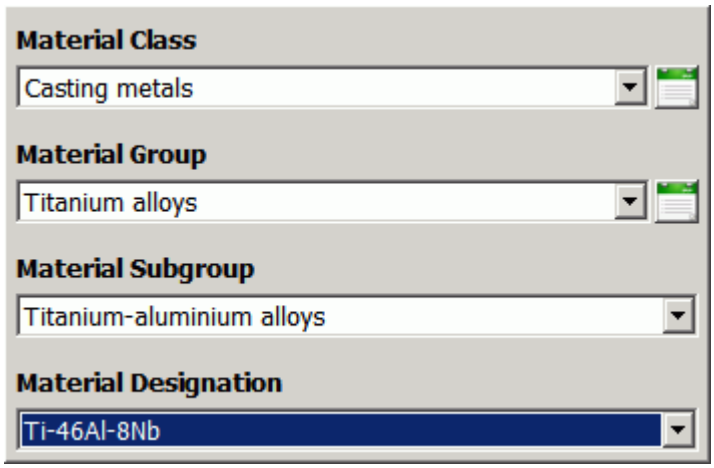
To open the casting material database:

1. Click (**Material Data**).
2. In the *Material Data* panel, select **Launch Database...**
3. Acknowledge the disclaimer and click **OK**.

The *STAR-Cast mat* dialog opens with the *Material Selection* panel selected by default. The STAR-CCM+ GUI is not active when the casting material database dialog is opened.

To select the material for the casting metal:

4. Set the *Material Class* to **Casting metals**.
5. Set the *Material Group* to **Titanium alloys**.
6. Set the *Material Subgroup* to **Titanium-aluminium alloys**.
7. Set the *Material Designation* to **Ti-46Al-8Nb**.



Selected materials (and the associated material properties) from the Material database must be exported to STAR-CCM+. You can create a collection of several materials and export them to STAR-CCM+ at the same time.

To add the selected Titanium alloy to the Collection of Materials:

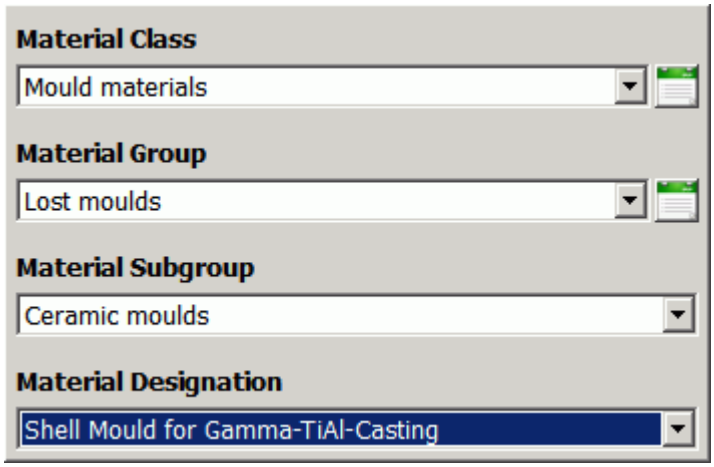
8. Click the *Export Data* tab.
9. Click **Add Material to Collection**.

   The Ti-46Al-8Nb alloy now appears in the list of Collection of Materials.

To select the material for the investment shell mold:

10. Click the *Material Selection* tab.

11. Set the *Material Class* to **Mould materials**.
12. Set the *Material Group* to **Lost moulds**.
13. Set the *Material Subgroup* to **Ceramic molds**.
14. Set the *Material Designation* to **Shell Mould for Gamma-TiAl-Casting**.



To add the investment shell mold material to the Collection of Materials:

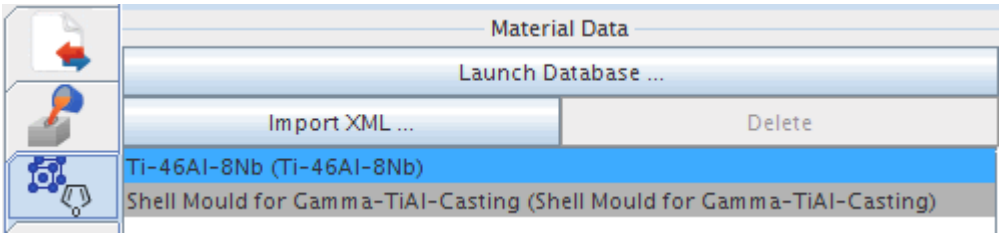
15. Click the *Export Data* tab.
16. Click **Add Material to Collection**.

The Shell Mould for Gamma-TiAl-Casting appears in the Collection of Materials list.

To export the Collection of Materials to STAR-CCM+:

17. Select **Send Collection to STAR-CCM+** in the *Export Data* tab.

This automatically closes the *STAR-Cast mat* dialog. In the CUI, you can now see the two selected materials from the Materials Database.



Setting Initial Conditions

Here, you set the initial conditions for the fluid InvestmentCast region and the solid Shell region.

1. Click the Regions symbol.
2. Within the *Initial Conditions* tab, you can define the initial conditions before you assign them to the appropriate regions. Set the following properties for the initial conditions of the fluid Cast and Top regions:

Properties	Options	Values
Name		`Cast_Init`
Type		**Fluid Region**
Temperature	**Constant**	`1100 °C.`
Velocity	**Constant**	`[0.0,0.0,0.0] m/s`
Volume Fraction	**Gas**	

3. Click **Create.** `Cast_Init` appears in the list of Initial Conditions.
4. Set the following properties for the initial conditions of the solid Shell region:

Properties	Options	Values
Name		`Shell_Init`
Type		**Solid Region**
Temperature	**Constant**	`900°C.`

5. Click **Create**.

To associate the initial conditions and the material properties with the appropriate regions:

6. Click the *Selections* tab. The three regions Cast, Shell, and Top are listed in the *Regions* column.
7. Click the empty *Materials* table cell next to the *Cast* table cell. Select **Ti-46Al-8Nb (Ti-46Al-8Nb)** as the material for this region.
8. Click the empty *Conditions* table cell for the Cast region. Select **Cast_Init** as initial condition.

   Cast_Init is offered as the only selection because it is the only fluid region condition defined.
9. Click the empty *Materials* table cell next to the *Top* table cell. Select **Ti-46Al-8Nb (Ti-46Al-8Nb)** as the material for this region.
10. Click the empty *Conditions* table cell for the Top region. Select **Cast_Init** as initial condition.
11. Click the empty *Materials* table cell next to the Shell region. Select **Shell Mould for Gamma-TiAl-Casting (Shell Mould for Gamma-TiAl-Casting)**.
12. Click the empty *Conditions* table cell for the Shell region. Select **Shell_Init** as the initial condition.

    Shell_Init is offered as the only selection because it is the only solid region condition that is defined.
13. Save the simulation.

Setting Boundary Conditions

For this investment casting simulation, you define five different types of boundary conditions.

To set the boundary conditions:

1. Click the Boundaries symbol in the CUI.
2. Within the *Settings* tab, you define the boundary conditions before associating them with the appropriate boundaries. At the melt inflow boundary set time-dependent profiles for velocity, temperature and the volume fraction of melt and air. Set the following properties:

Properties	Options	Values
Name		`Melt_Inlet`
Type		**Velocity Inlet Boundary**
Temperature	**Table (Time)**	
Velocity	**Table (Time)**	
Volume Fraction	**Table (Time)**	

3. For each of the time-dependent properties, do the following:
   a) Set **Tabulate** selector to **Import .csv**.
      A **File** button appears to the right.
   b) Click **File**, navigate to the `doc/startutorialsdata/casting/data` subdirectory, and select the file `MeltInletLinear.csv`.

   You can use a common text editor to open the file `MeltInletLinear.csv`. Inspect the format of the file. You can see that it is a multi-column file where all the inflow values are listed as columns depending on time as the first column. The headers in double quotes contain the keywords for the CUI to import the correct variable into the table editor.
4. Click **Create**.
5. Set the Pressure Outlet Boundary:

Properties	Options	Values
Name		`Gas_Outlet`
Type		**Pressure Outlet Boundary**
Temperature	**Constant**	`600 °C`
Pressure	**Constant**	`0.0 Pa`
Volume Fraction	**Gas**	

   Note that you can change the temperature units from K to °C by typing a C after the temperature value.

   The volume fraction setting ensures that in case of inflow through the pressure boundary (due to recirculation), only air enters the fluid domain.
6. Click **Create**.
7. Set the Adiabatic Fluid Wall Boundary:

Properties	Options	Values
Name		`Melt_Wall`
Type		**Adiabatic Fluid Wall Boundary**
Contact Angle	**Constant**	`90.0 deg`

8. Click **Create.**
9. Set the Solid Wall Boundary:

Properties	Options	Values
Name		`Shell_Wall`
Type		**Solid Wall Boundary**
Ambient Temperature	**Constant**	`20 °C`
Heat Transfer Coefficient	**Constant**	`10 W/(m²K)`
Emissivity	**Constant**	`0.5`

10. Click **Create.**
11. Set the Symmetry Plane Boundary:

Properties	Values
Name	`Symmetry`
Type	**Symmetry Plane Boundary**

12. Click **Create.**

Proceed to associate the boundary conditions with the appropriate boundary surfaces.

13. Click the *Selections* tab.
14. For each row in the table, click the empty *Conditions* cell and set the conditions as follows:

Boundaries	Conditions
Cast: Symmetry	**Symmetry**
Top: Inlet	**Melt_Inlet**
Top: NoSlipWall	**Melt_Wall**
Top: Symmetry	**Symmetry**
Top: PressureOutlet	**Gas_Outlet**
Shell: Offset Extrusion	**Shell_Wall**

Maintain the default *Conditions* for the remaining boundaries.

Setting Interface Properties

In this model, there are two interfaces: a contact interface between the fluid Cast region and the solid Shell region and an internal interface between the Top and the Cast region. At the contact interface, variable values are transferred between the two regions for a conjugate heat transfer analysis.

To specify the interface settings:

1. Click the Interfaces symbol.
2. Within the *Settings* tab, set the following properties for the interface between the melt and the shell:

Properties	Options	Values
Name		`Melt_Shell_Interface`

Properties	Options	Values
Type		**Fluid to Solid**
Contact Angle	**Constant**	`90 deg`
Heat Transfer Coefficient	**Table (Temp.)**	`HTC_T.csv`
Porous Inertia Coefficient	**Constant**	`0.0`
Porous Viscous Coefficient	**Constant**	`120000 m/s`

3. Click **Create**.
4. Set the following properties for the interface between the Cast and the Top region:

Properties	Options	Values
Name		`Cast_Top_Interface`
Type		**Flow Through**

5. Click **Create**.

Associate the interface condition with the appropriate interface surfaces of the model:

6. Click the *Selections* tab and set the following conditions:

Interfaces	Conditions
Cast/Top	**Cast_Top_Interface**
Offset interface on Cast-Shell Interface	**Melt_Shell_Interface**

7. Save the simulation.

Setting Physics Models

The CUI sets the basic physics models for a multiphase, conjugate heat transfer simulation. There are optional physical models available that allow you to specify the melting-solidification properties of the melt.

For this tutorial you use:

- Melting-solidification flow stop model
- Mushy zone permeability model
- Surface tension model

To specify the properties of the optional melting-solidification models:

1. Click the Physical Models symbol in the CUI.
2. Within the *Settings* tab, set the Melting-Solidification Flow Stop parameters:

Properties	Options	Values
Name		`FlowStop`
Model		**Melting and Solidification**
Flow Stop Relative Solid Fraction	**Constant**	`0.85`

Properties	Options	Values
Flow Stop Solid Fraction	**Constant**	`0.85`

3. Click **Create**.
4. Set the Mushy Zone model parameters:

Properties	Options	Values
Name		`MushyZone`
Model		**Mushy Zone**
Max. Viscous Resistance	**Constant**	`50000 kg/(m³s)`
Secondary Dendrite Arm Spacing	**Constant**	`30.0 um`
Critical Solid Fraction	activate **Use Switching**	`0.5`

Note that `um` represents `µm`.

5. Click **Create**.
6. Set the Surface Tension model parameters:

Properties	Options	Values
Name		`SurfaceTension`
Model		**Surface Tension**
Surface Tension	**Constant**	`0.75 N/m`

7. Click **Create**.

To assign the physical model conditions to the appropriate physical models:

8. Click the *Selections* tab.
9. For each row in the table, click the empty *Conditions* table cell and set the conditions as follows:
   a) Set *Melting and Solidification* to **FlowStop**.
   b) Set *Mushy Zone* to **MushyZone**.
   c) Set *Surface Tension* to **SurfaceTension**.
10. Save the simulation.

Setting Solver Parameters

Solver parameters must be set for the investment casting simulation.

To access the Solvers panel, click the Solvers symbol. The *Solvers* panel contains three tabs:

- Solver Settings
- Stopping Criteria
- Process Phases

Set the parameters moving from left to right.

1. Set the Solver Parameters:
   a) Click the *Solver Settings* tab.
   b) Enter `SolverSettings` as the *Name* for the condition.
   c) Set the time step size and the under-relaxation factors:

Properties	Values
Time Steps	**Automatic**
Volume of Fluid	**Condition On**
Velocity URF	`0.6`
Volume of Fluid URF	`0.75`
Melting/Solidification URF	`0.75`
Mushy Zone URF	`0.2`

   d) Select the **Time Steps** property and set the Automatic Time Step Controls at the bottom of the panel:

Properties	Values
Start Time	`0 s`
End Time	`3 s`
Start Time Step	`2.0E-5 s`
Minimum Time Step	`1.0E-5 s`
Max. Time Step Change Factor	`[1.3,2.0]`

   e) Select the **Volume of Fluid** property and set the following properties:

Properties	Values
Max. Condition Limit	`0.12`
Smoothing Steps	`4.0`

   f) Click **Create**.
2. Set the Stopping Criteria for the run:
   a) Click the *Stopping Criteria* tab.
   b) Enter `StopAfter3s` as the *Name* for the condition.
   c) Set the following properties:

Properties	Options	Values
Maximum Physical Time		`3 s`
Maximum Inner Iteration		`30`
Minimum Inner Iteration		`8`
Mass Error Rate	**Minimum**	`1.5 /s`

   a) Click **Create**.

3. Set the Process Phase:
   a) Click the *Process Phases* tab and enter `ProcessPhase` as *Process Phase Name*.
   b) Click **Insert**.
   c) Set the following conditions:

Process Phase	Solver Settings	Stopping Criteria
ProcessPhase	**SolverSettings**	**StopAfter3s**

Switch to the simulation tree to set the maximum time step:

4. Select the **Solvers** > **Implicit Unsteady** node and set *Time-Step* to `0.1 s`.
5. Save the simulation.

Setting up Criteria Functions

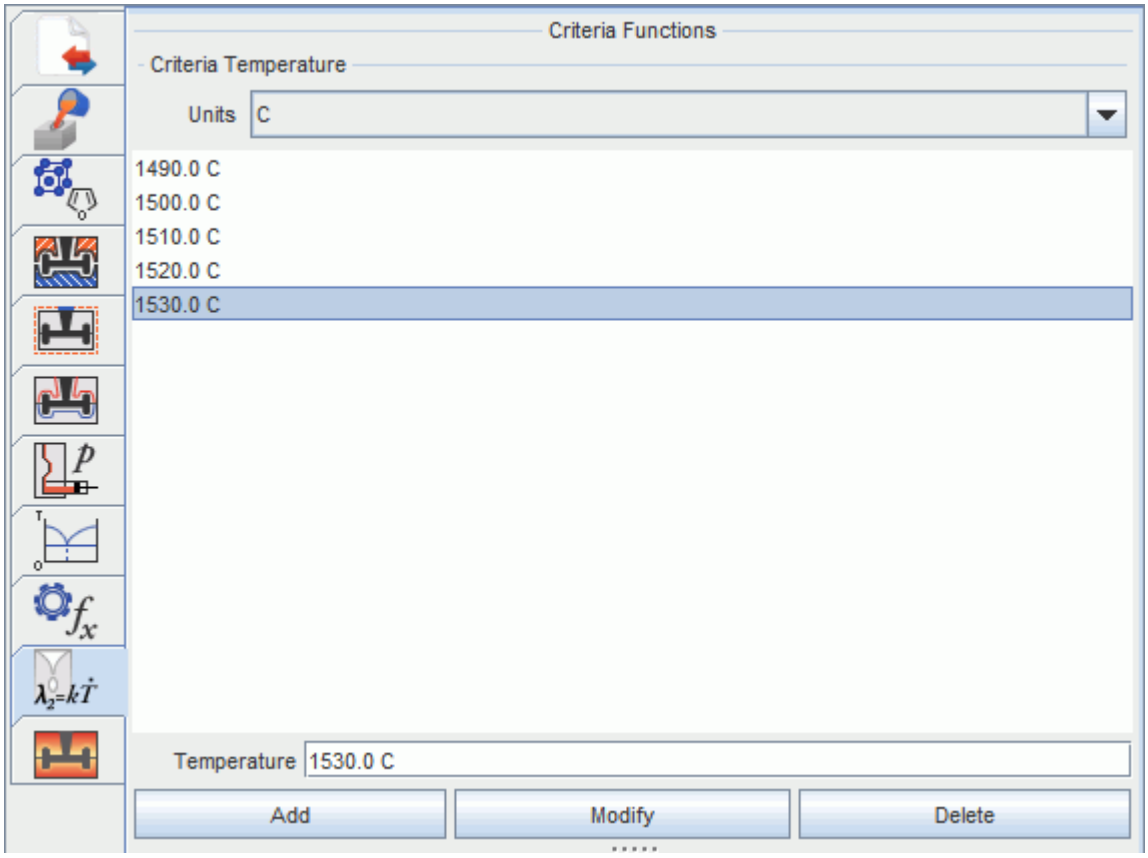
By default, the CUI activates all criteria functions except **Liquid Residence Time**, which is optional.

For more information, see Criteria Functions Definitions in the **User Guide** > **Modeling Physics** > **Modeling Casting** > **Using Criteria Functions**.

To set the critical temperatures for which criteria functions are evaluated:

1. Click the Criteria Functions symbol.
2. In the **Temperature** box, enter `1490 C` and click **Add**. This adds the first temperature to the list of critical temperatures.

   Note that the *Criteria Temperatures* have to be specified in °C.
3. In the same way, add temperatures `1500 C`, `1510 C`, `1520 C`, and `1530 C`.

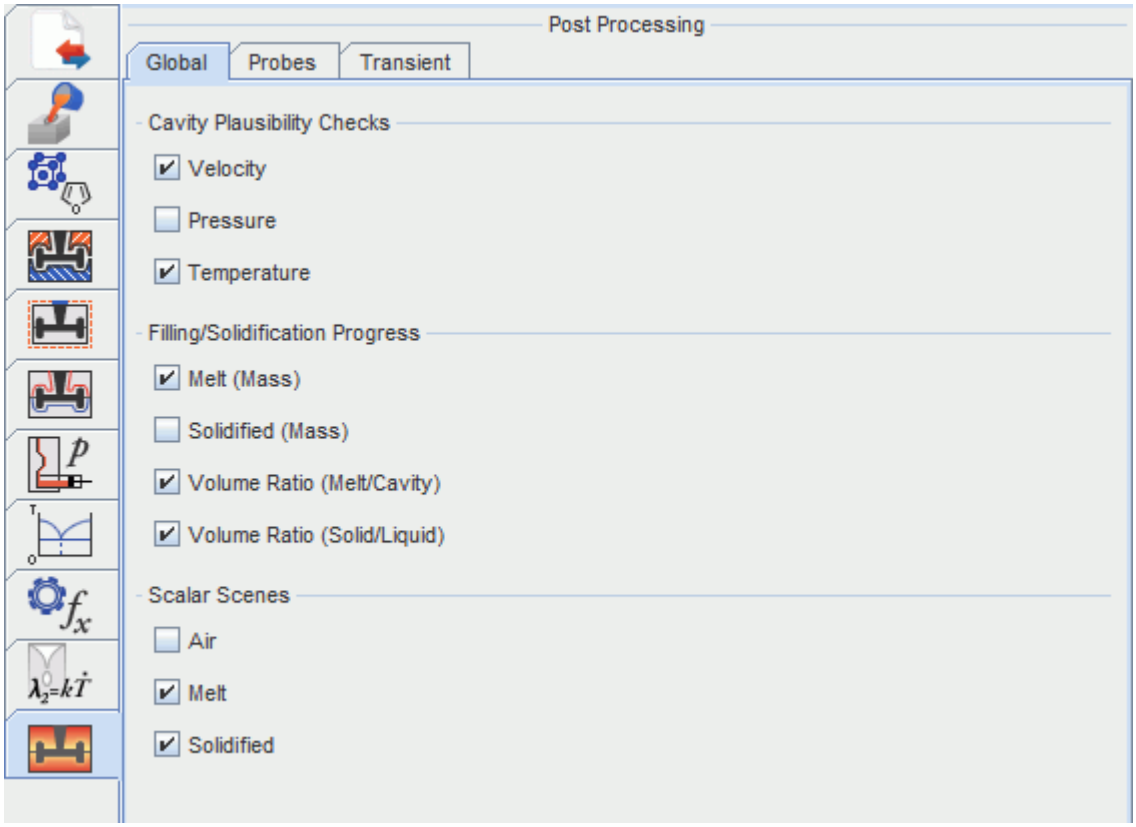


Setting Up Post Processing

You set up visualization of the results of the investment casting simulation from within the CUI.

To select the items for reporting and visualization:

1. Click the Post Processing symbol.
2. In the *Global* tab, activate the following post-processing items:
   - In the *Cavity Plausibility Checks* group box, select **Velocity** and **Temperature**.
   - In the *Filling/Solidification Progress group* box, select **Melt (Mass), Volume Ratio (Melt/Cavity)**, and **Volume Ratio (Solid/Liquid)**.
   - In the *Scalar Scenes* group box, select **Melt** and **Solidified**.



This completes the setup of the casting simulation using the CUI.

3. Save the simulation.

Running the simulation

After the filling process is complete, the flow stabilizes, and only melting/solidification takes place.

You are advised to run this tutorial on a multi-core machine with four cores or more. Depending on the specifications of the machine, this tutorial takes one to two days to complete.

Before you run the simulation, increase the interface artificial viscosity to reduce parasitic currents in the proximity of the free surface.

To increase the interface artificial viscosity:

1. Above the STAR-Cast panel you can see two buttons, **Simulation** and **STAR-Cast**. Click **Simulation**.

   The simulation tree appears.
2. In the simulation tree, select the **Continua** > **Melt** > **Models** > **Multiphase Interaction** > **Phase Interactions** > **Casting_Phase_Interaction** > **Models** > **Interface Momentum Dissipation** > **Interface Artificial Viscosity** > **Constant** node and set *Value* to `1.0`.
3. To switch back to the STAR-Cast panel, click **STAR-Cast**.

To run the simulation:

4. Click the Solvers symbol.

5. In the *Process Phases* tab, select **ProcessPhase** in the *Process Phase* table.
6. Click **Run**.

   The simulation starts running with the numerical settings that are defined for the process phase.

Visualizing the solution

The Casting User Interface (CUI) provides two scalar scenes by default.

These scenes are called:

- Melt

  The Melt scene shows a plot of the liquid melt in the computational domain.
- Solidified

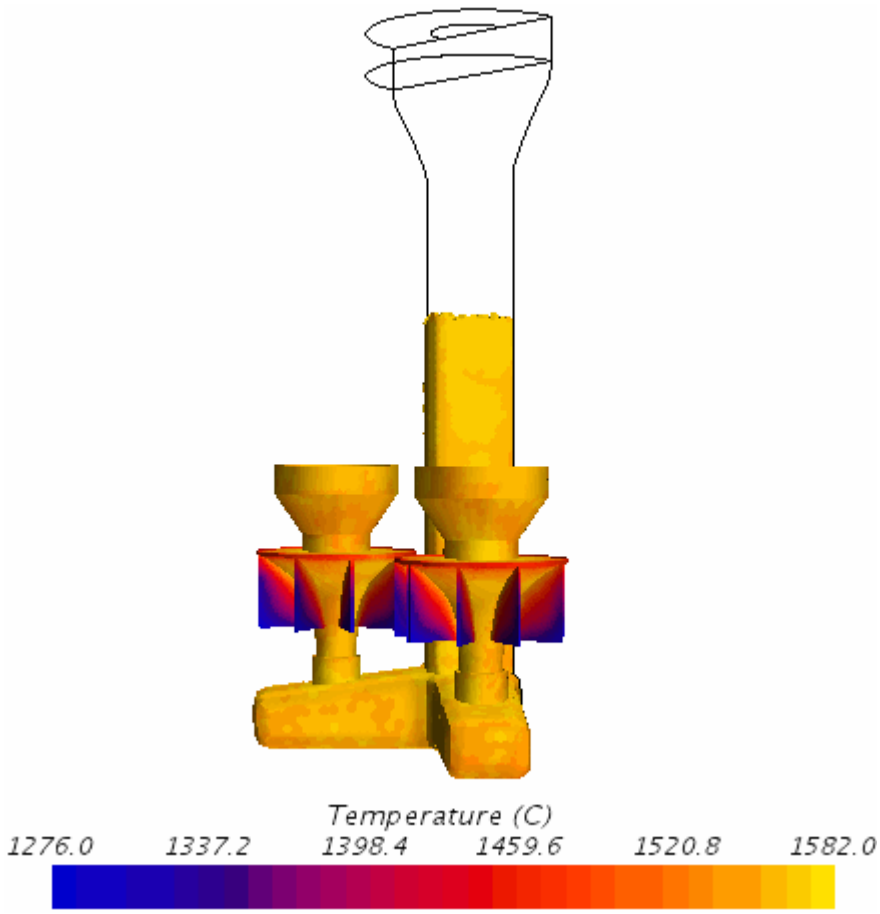
  The Solidified scene shows a plot of the melt that is already solidified.

You can use these scalar scenes to visualize the development of the solution while the simulation runs.

To visualize the liquid melt:

1. Right-click the **Scenes** > **Melt** node and choose **Open**.
2. Press the <f> key on your keyboard to obtain a front view of the plot.

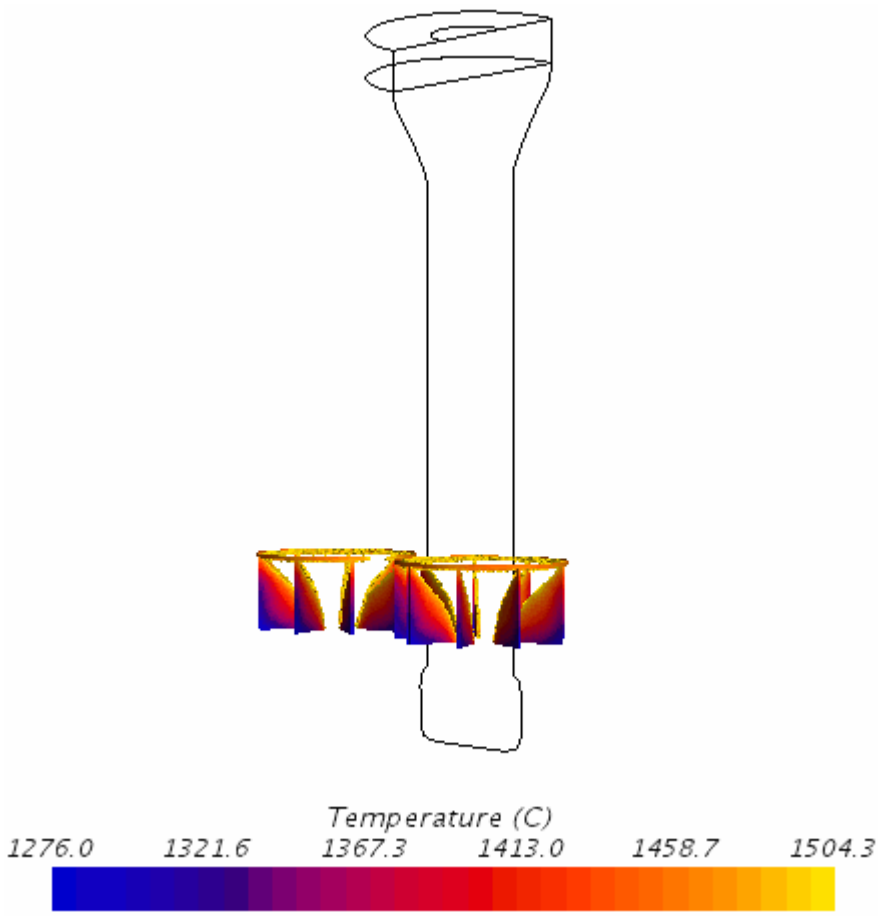
   The following screenshot shows the position and temperature of the liquid melt at the end of the simulation:



To visualize the solidified material in the domain:

3. Right-click the **Scenes** > **Solidified** node and choose **Open**.
4. Press the <f> key on your keyboard to obtain a front view of the plot.

   The following screenshot shows the position and temperature of the solidified melt at the end of the simulation:.

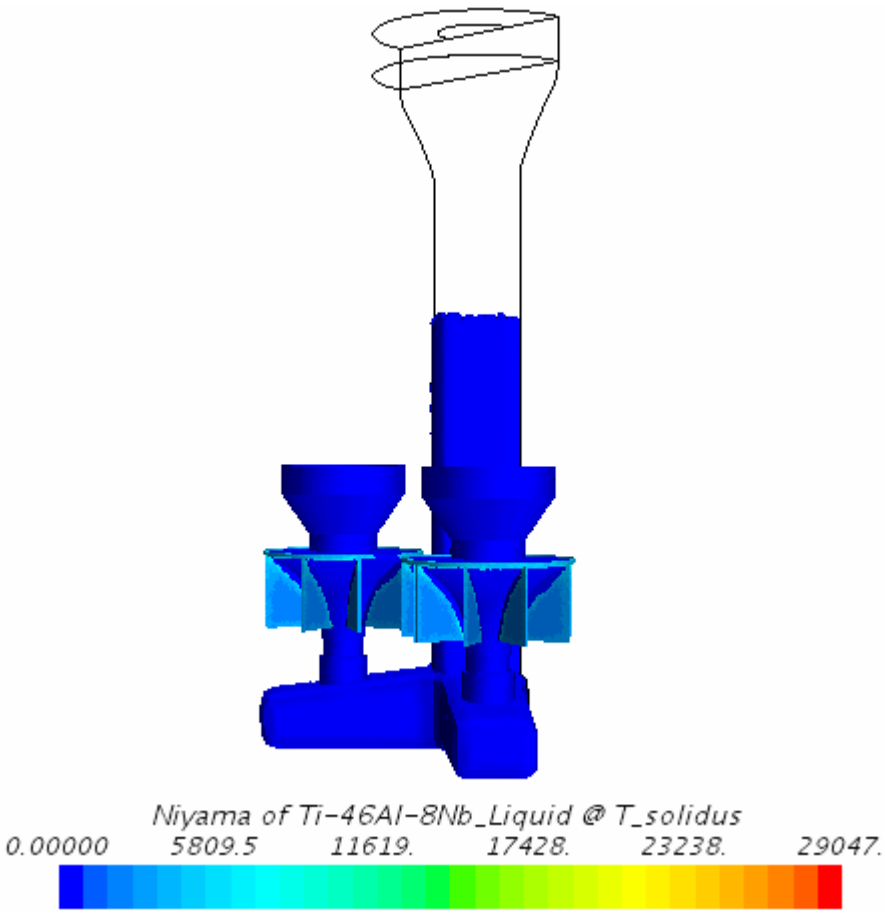


After the simulation is finished, you can visualize the criteria functions.

To visualize the Nyama criterion:

5. Create a scalar scene.
6. Select the **Displayers** > **Scalar 1** > **Parts** node.
7. Set *Parts* to **Derived Parts** > **VolumeFractionTi-46Al-8Nb_Liquid AboveMax Threshold.**
8. Select the **Displayers** > **Scalar 1** > **Scalar Field** node and set *Function* to **Niyama** > **Ti-46Al-8Nb_Liquid** > **Niyama of Ti-46Al-8Nb_Liquid @ T_solidus**.
9. Press the <f> key on your keyboard to obtain a front view of the plot.

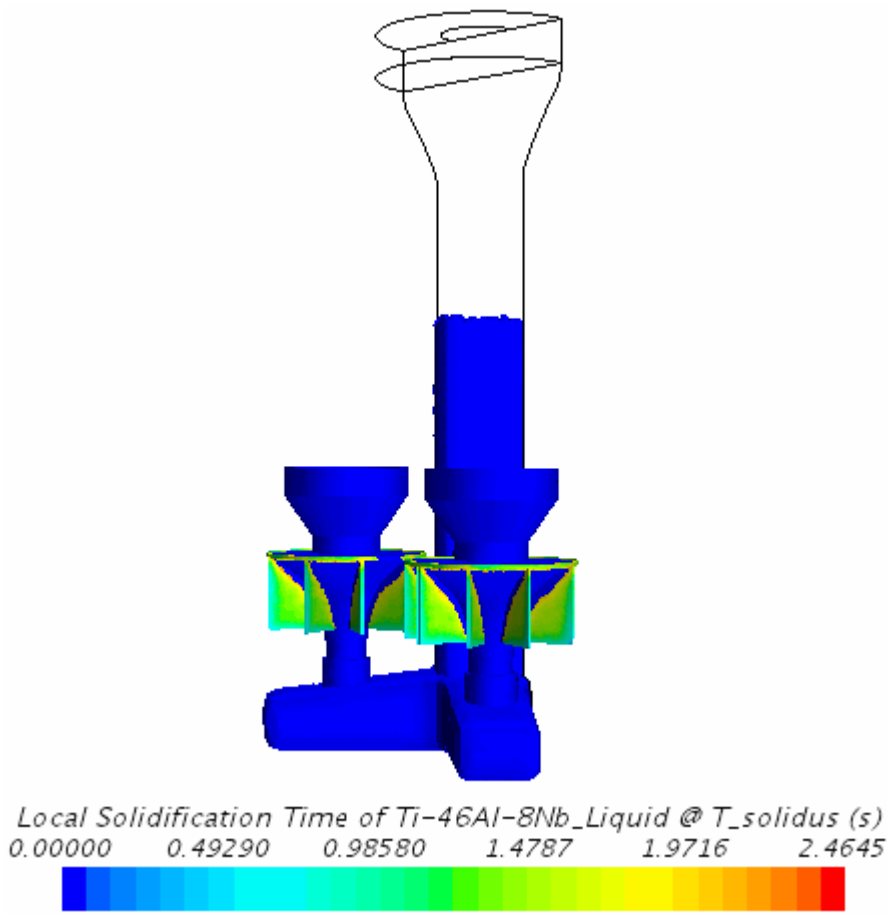
   The following screenshot shows the Niyama criterion after a physical time of 3 s:



To visualize the local solidification time at the solidus temperature:

10. Create a scalar scene.
11. Select the **Displayers** > **Scalar 1** > **Parts** node.
12. Set *Parts* to **Derived Parts** > **VolumeFractionTi-46Al-8Nb_Liquid AboveMax Threshold.**
13. Select the **Displayers** > **Scalar 1** > **Scalar Field** node and set *Function* to **Local Solidification Time** > **Ti-46Al-8Nb_Liquid** > **Local Solidification Time of Ti-46Al-8Nb_Liquid @ T_solidus**.
14. Press the <f> key on your keyboard to obtain a front view of the plot.

    The following screenshot shows the local solidification time criterion after a physical time of 3 s:



Summary

This tutorial demonstrates the workflow for setting up an investment casting simulation.

This tutorial has introduced the following STAR-CCM+ features:

- Importing an investment casting geometry.
- Using the Offset Mesher to generate a shell mold for investment casting simulations.
- Manually preparing the geometry for volume meshing.
- Setting up an investment casting simulation in the Casting User Interface.
- Running an investment casting simulation.
- Visualizing the simulation results.

Automation

The tutorials in this set illustrate various STAR-CCM+ automation and macro features:

Contents:

Simple Java Macros: Post-Processing Objects
Intermediate Java Macros: Run Multiple Simulations
Simulation Assistant: Internal Flow Assistant

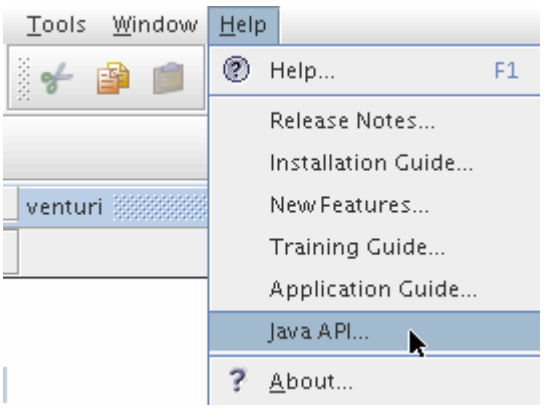
Simple Java Macros: Post-Processing Objects

This tutorial is intended to demonstrate the steps required in recording, editing, and replaying a Java macro in STAR-CCM+.

The case considered is that of an incompressible flow through a venturi flowmeter, and the objective in creating a macro is to generate multiple line probes throughout the solution domain to observe recovery of the flow profile.

The venturi flowmeter is symmetric around its central axis and can be readily modeled using a two-dimensional axisymmetric mesh. In STAR-CCM+ this requires you to import an initial three-dimensional mesh and convert it to two-dimensions using the relevant operations.

The object model that is used by the STAR-CCM+ Java Client is available for use by you in Java macros. Documentation for all classes and methods within this object model are available from the **Help** > **Java API** menu.



Contents:

Prerequisites
Importing the Volume Mesh
Converting the Mesh to Two-Dimensions
Selecting the Physics Models
Setting Boundary Conditions and Values
Setting Stopping Criteria
Visualizing the Simulation
Running the Simulation
Recording a Macro
Editing the Macro
Reviewing the Edited Macro
Running the Macro
Plotting Data Using Line Probes
Exporting XY Plot Data

Further Training
Summary

Prerequisites

The instructions in the Simple Java Macros tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+

Importing the Volume Mesh

To set up the STAR-CCM+ simulation, launch a simulation and import the supplied volume mesh.

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh**.
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/automation/data`.
5. Select `venturi.ccm` then click **Close**.

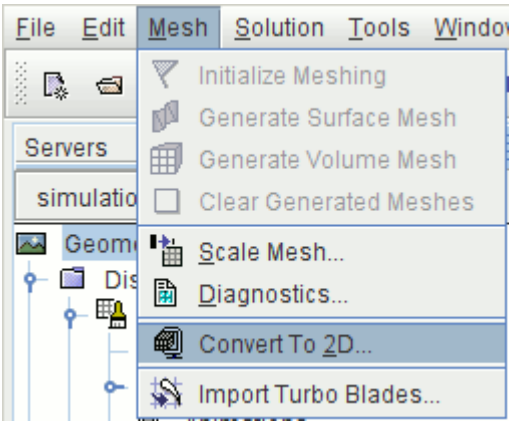
   STAR-CCM+ provides feedback on the import process in the *Output* window and a mesh region is created under the **Regions** node.
6. Save the simulation as `venturi.sim`.

Converting the Mesh to Two-Dimensions

STAR-CCM+ can use the surface of a three-dimensional mesh to create a two-dimensional mesh if the surface to be used lies on the X-Y plane at Z = 0.

Convert the mesh to two-dimensions:

1. Select **Mesh** > **Convert To 2D** from the menu.



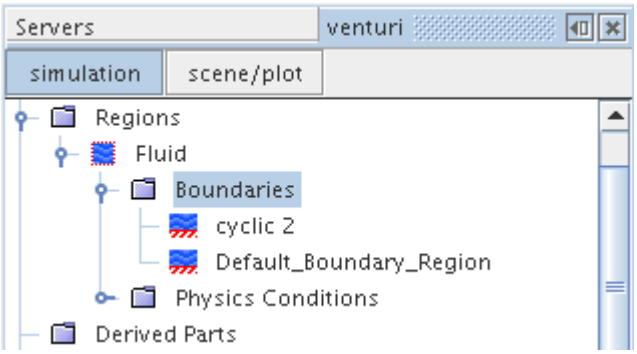
2. In the *Convert Regions To 2D* dialog, set the following properties:

Property	Setting
Delete 3D regions after conversion	**Activated**
Tolerance	`1.0E-18`

3. Click **OK**.

Some work is now required to clean up the region and associated boundaries.

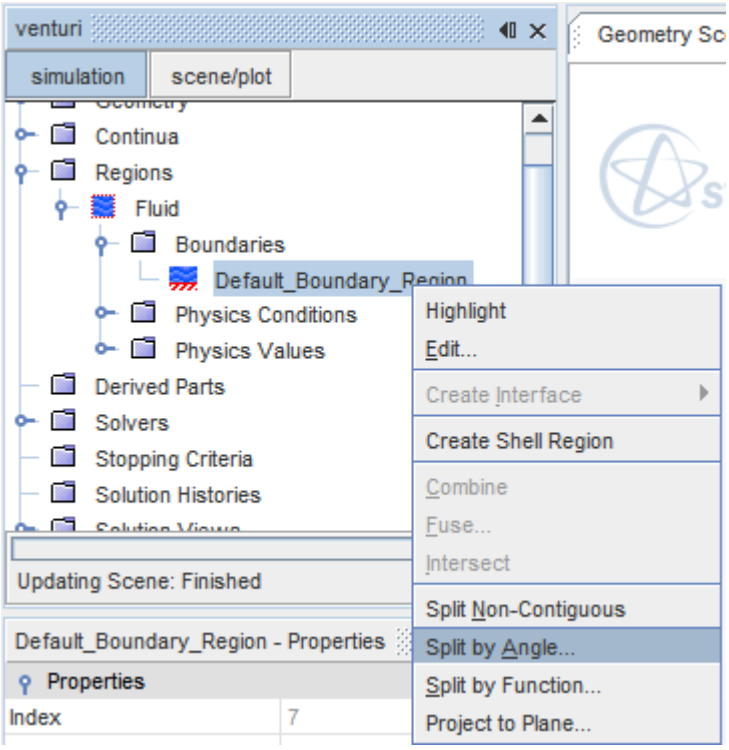
4. Expand the **Regions** node and rename the **fluid 2D** node to **Fluid**.
5. Expand the **Regions** > **Fluid** > **Boundaries** node to see the boundaries that remain after the conversion process.



Combine these boundaries to form a single boundary and then split it into four separate boundaries:

6. Multiselect the **Default_Boundary_Region** node and the **cyclic 2** node.
7. Right-click and select **Combine**.

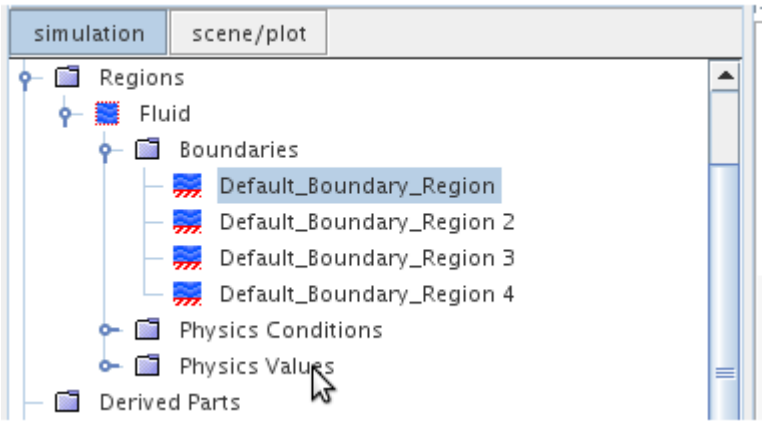
   There is now a single **Default_Boundary_Region** node.
8. Right-click the **Default_Boundary_Region** node and select **Split By Angle**.



The default settings in the **Split Boundaries by Angle** dialog are sufficient.

9. Click **OK**.

   Four boundaries appear as shown below:



10. Rename the boundaries as follows:

Old Name	New Name
Default_Boundary_Region	`Axis`
Default_Boundary_Region 2	`Outlet`
Default_Boundary_Region 3	`Inlet`
Default_Boundary_Region 4	`Wall`

At this stage, do not set the boundary types until you have selected the models for the analysis.

An additional physics continuum was created for the two-dimensional mesh during the mesh conversion process, which means that the original continuum for the three-dimensional mesh is not required.

11. Delete the **Continua** > **Physics 1** node.
12. Rename the **Physics 1 2D** node to `Physics 1`.
13. Save the simulation.

Selecting the Physics Models

Physics models define the physical variables and phenomena in the simulation. An incompressible isothermal turbulent flow is sufficient for this analysis. Use the segregated flow model with the k-omega turbulence model.

To select the physics models:

1. For the physics continuum, **Continua** > **Physics 1**, in the *Enabled Models* group box, deactivate **Two-Dimensional**.
2. Select the following models in order:

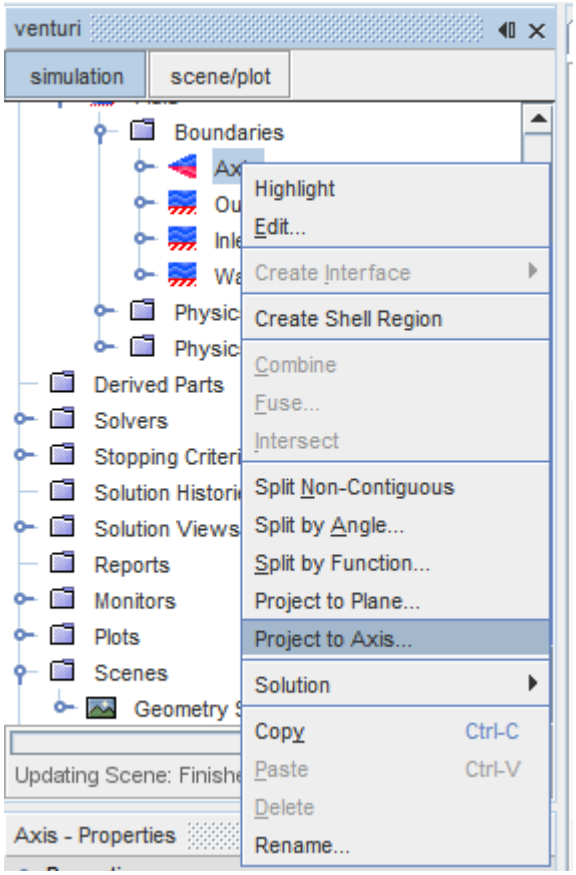
Group Box	Model
Space	**Axisymmetric**
Time	**Steady**
Material	**Liquid**
Flow	**Segregated Flow** **Gradients** (Selected automatically)
Equation of State	**Constant Density**
Viscous Regime	**Turbulent** **Reynolds-Averaged Navier-Stokes** (Selected automatically)
Reynolds-Averaged Turbulence	**K-Omega Turbulence** **SST (Menter) K-Omega** (Selected automatically) **Exact Wall Distance** (Selected automatically) **All y+ Wall Treatment** (Selected automatically)

3. Click **Close**.
4. Save the simulation.

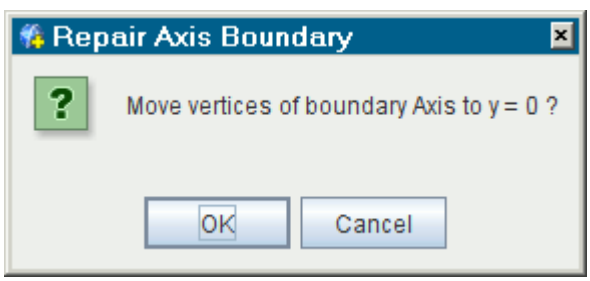
Setting Boundary Conditions and Values

Specify a constant velocity in the X direction at the inlet, and a constant pressure at the outlet. Set the centerline boundary to be of type axis. This axis boundary must lie exactly on the Y = 0 line; project the axis to the appropriate plane.

1. Select the **Regions** > **Fluid** > **Boundaries** > **Axis** node and set **Type** to **Axis**.
2. Right-click the **Boundaries** > **Axis** node and select **Project to Axis** from the pop-up menu.



3. In the `Repair Axis Boundary` dialog, click **OK**.



The vertices move along the axis to Y = 0.

Specify conditions for the inlet and outlet boundaries:

4. Set the boundary types for the inlet and outlet:

Boundary	Type
Inlet	**Velocity Inlet**
Outlet	**Pressure Outlet**

5. Edit the **Boundaries** > **Inlet** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Turbulence Specification**	*Method*	**Intensity + Length Scale**
└ **Velocity Specification**	*Method*	**Components**
Physics Values		
└ **Turbulence Intensity** > **Constant**	*Value*	`0.034`
└ **Turbulent Length Scale** > **Constant**	*Value*	`0.0107`
└ **Velocity** > **Constant**	*Value*	`[1.5, 0.0 0.0] m/s`

6. Edit the **Boundaries** > **Oultet** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Turbulence Specification**	*Method*	**Intensity + Length Scale**
Physics Values		
└ **Turbulence Intensity** > **Constant**	*Value*	`0.034`
└ **Turbulent Length Scale** > **Constant**	*Value*	`0.0107`

7. Save the simulation.

Setting Stopping Criteria

Set the stopping criteria for the simulation. By default, simulations are set to stop after 1000 iterations.

In this simulation, convergence is reached before then, so reduce the maximum number of iterations to 600.

1. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to `600`.
2. Save the simulation.

Visualizing the Simulation

Create a scalar scene to display the pressure field as the solution develops.

1. Create a scalar scene.

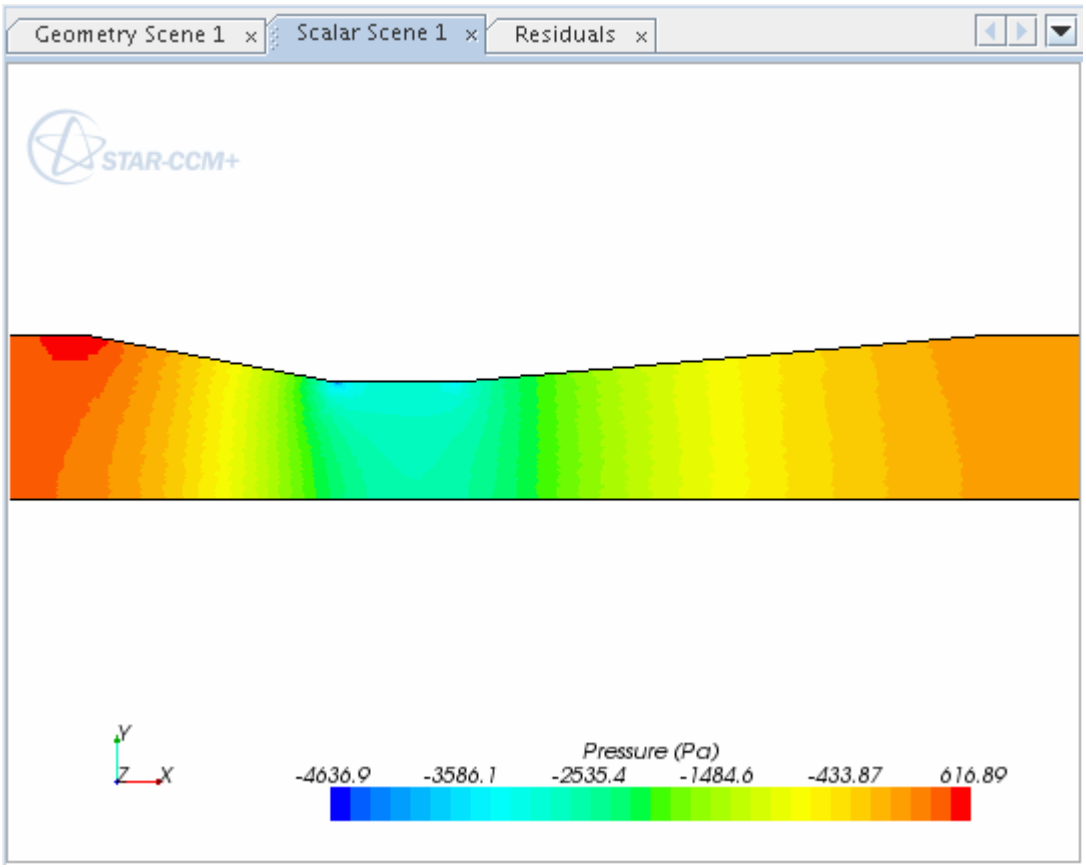
2. Navigate to the **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Scalar Field** and set *Function* to **Pressure**.
3. Save the simulation.

Running the Simulation

Preparation of the simulation is now complete, and the simulation can be run.

1. Click (**Initialize Solution**) in the *Solution* toolbar.
2. Click (**Run**) in the *Solution* toolbar.
3. While the simulation is running, select the *Scalar Scene 1* tab at the top of the *Graphics* window to visualize the solution.
4. When the simulation has finished running, save it.

The scalar scene appears as follows:



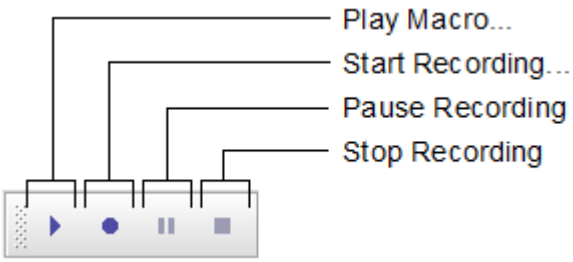
Recording a Macro

Create a macro to generate line probes at equally spaced intervals along the axis of the venturi tube. This enables you to extract profiles of axial velocity that can be used to observe the recovery of the flow after the venturi section.

Follow the procedure:

1. Initiate recording of the macro.
2. Manually work through the steps to create the first line probe.
3. Manually add the line probe to the scalar scene.
4. Stop recording the macro.

The **macro** toolbar contains four actions as shown below.



Start recording the macro:

1. Click (**Start Recording...**) to initiate the recording process.
2. In the *Save* dialog, enter `createLineProbes` for the name of the java file to which the macro is saved.
3. Click **Save**.

   An extra page called *createLineProbes* appears in the *Output* window.

   From this point onwards, any recordable action you make in STAR-CCM+ generates lines of Java code in this page. Clicking the (**Pause Recording**) button pauses generation of the Java code. Clicking the same button resumes generation of the Java code. Clicking the (**Stop Recording**) button ends all generation of the Java code and write the macro to file. Do not click either of these buttons for now.

Work through the required actions:

4. Right-click the **Derived Parts** node and select **New Part** > **Probe** > **Line** from the pop-up menu.
5. In the *Create Line Probe* dialog, set the following properties:

Property	Setting
Input Parts	**Fluid** (default)
Point 1	`[0.04, 0.0, 0.0] m`
Point 2	`[0.04, 0.0762, 0.0] m`
Resolution	`20`
Display	**New Geometry Displayer** (default)

6. Click **Create** then **Close**.
7. Rename the **Derived Parts** > **line-probe** node to `X = 0.04`.
8. Select the **Scenes** > **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Parts** node and set *Parts* to **X = 0.04**.

This completes the steps you wish to include in the macro.

To stop the macro:

9. Click the (**Stop Recording**) button in the macro toolbar.

Editing the Macro

Examine the recorded macro to identify where you can insert a loop for the creation of multiple line probes.

Note: The layout of the macro presented in the following sections may be different from what appears in your editor.

- Initial lines of the macro
- Initial sections of the execute0 method
- Code that obtains the Scene object
- Code to be repeated multiple times
- Adding the loop
- Displaying the new parts

Contents:

Initial Lines of the Macro
Initial Sections of the execute0 Method
Code that Obtains the Scene Object
Code to be Repeated Multiple Times
Adding the Loop
Displaying the New Parts

Initial Lines of the Macro

Open the file `createLineProbes.java` in an appropriate text editor. The first few lines of the macro specify the package to which the macro class belongs and the imported libraries that STAR-CCM+ use to compile the macro.

```
// STAR-CCM+ macro: createLineProbes.java
package macro;

import java.util.*;

import star.common.*;
import star.base.neo.*;
import star.vis.*;
```

After this initial section, see the class specifier.

```
public class createLineProbes extends StarMacro {
```

A macro recorded by STAR-CCM+ is a fully specified Java class derived from the `StarMacro` base class. The parts of this class specifier are briefly explained as follows:

- `public` informs the Java compiler that there are no access restrictions on this class.
- `class` is the keyword used to define a class.
- `createLineProbes` is the name of the class. This name must match the name of the file containing the class.
- `extends` indicates that the class is derived from another class (and inherits all the public and protected methods of that class).
- `StarMacro` is the name of the base class from which this one is derived.
- `{` is the opening brace of the class. It is matched by a closing brace at the end of the class.

The `createLineProbes` class contains one method called `execute`. All macros must contain this method, and this is the method that STAR-CCM+ calls to run the macro:

```
public void execute() {
```

The parts of the method signature are as follows:

- `public` is the access specifier and indicates that there are no restrictions on calling this method.
- `void` indicates that this method does not return any value.
- `execute()` is the name and argument list of the method. There are no arguments for this method.
- `{` is the opening brace of the method, and must be matched by a closing brace at the end.

The body of the `execute` method simply calls the private method, `execute0`. Consideration of `execute0` now follows.

Initial Sections of the execute0 Method

Examine the initial sections of the `execute0` method.

The first two sections of the method are:

```
Simulation simulation_0 =
  getActiveSimulation();

Units units_0 =
  simulation_0
  .getUnitsManager()
  .getPreferredUnits(
    new IntVector(
      new int[] {0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
  )
);
```

where:

```
simulation_0
.getUnitsManager()
.getPreferredUnits(
  new IntVector(
    /* etc */
```

is equivalent to

```
simulation_0.getUnitsManager().getPreferredUnits(new IntVector(/* etc
*/
```

The call to `getActiveSimulation()` returns the Java object that represents the simulation currently active in the STAR-CCM+ client window. This is the main object in the Java macro and provides access to many other object managers (for example `RegionManager` or `RepresentationManager`). The Java API documentation provides details on all methods available on the `Simulation` class.

The `simulation_0` object is used to obtain the `UnitsManager` object. The `UnitsManager` object is used to find the `Units` object that represents the default unit for length in the active simulation (through the call to `getPreferredUnits`). The default unit for length is required when specifying the two points used in creating line probes.

Code that Obtains the Scene Object

Examine the code that obtains the `scene` object.

There are several lines of code after the call on the `UnitsManager` object that obtains the `Scene` object (as `scene_0`) corresponding to *Scalar Scene 1* in the *Graphics* window. The **Fluid** region that is displayed in this scene is obtained using the `RegionManager` and is referred to in the code as `region_0` , which is a `Region` object.

```
Scene scene_0 =
  simulation_0
  .getSceneManager()
  .getScene("Scalar Scene 1");

Region region_0 =
  simulation_0
  .getRegionManager()
  .getRegion("Fluid");
```

Some additional code is present that defines the particular representation being used in the `PartDisplayer`:

```
PartDisplayer partDisplayer_7 =
    scene_1.getDisplayerManager().createPartDisplayer("Probe
Geometry", -1, 1);
```

Code to be Repeated Multiple Times

You now come to the code to be placed within a loop. Some Java comments have been added to help describe the operations taking place. Items to be replaced by variables are underlined, and sections to be removed are underlined.

```
// The LinePart object is the Derived Part you want to
// create. The PartManager on the simulation_0 object
// has the method that lets you do this.

LinePart linePart_0 =
      simulation_0.getPartManager().createLinePart(new
NeoObjectVector(new Object[] {}),
new DoubleVector(new double[] {0.0, 0.0, 0.0}),
new DoubleVector(new double[] {1.0, 0.0, 0.0}),
20);

// The LinePart object must now be told
// what coordinate system it is using. You retrieve the
// LabCoordinateSystem object from the
// CoordinateSystemManager on the simulation_0 object.

LabCoordinateSystem labCoordinateSystem_0 =
  ((LabCoordinateSystem)
    simulation_0
    .getCoordinateSystemManager()
    .getLabCoordinateSystem();

// As you now have the labCoordinateSystem_0 object we
// can tell linePart_0 to use it.

linePart_0
.setCoordinateSystem(labCoordinateSystem_0);

// The next two sections set the units on each axis,
// and the position, of the first Point of linePart_0

Coordinate coordinate_0 =
  linePart_0.getPoint1Coordinate();

coordinate_0
.setCoordinate(
  units_0,
  units_0,
  units_0,
  new DoubleVector(
    new double[] {0.04, 0.0, 0.0}
  )
```

```
);

// The next two sections set the units on each axis,
// and the position, of the second Point of linePart_0.

Coordinate coordinate_1 =
  linePart_0
  .getPoint2Coordinate();

coordinate_1
.setCoordinate(
  units_0,
  units_0,
  units_0,
  new DoubleVector(
    new double[] {0.04, 0.0762, 0.0}
  )
);

// The following two sections are updating the visible
// scene and are not required for creating line parts.
// Note: The part number may differ slightly in your
// recording.

PartDisplayer partDisplayer_5 =
     scene_1.getDisplayerManager().createPartDisplayer("Probe
Geometry", -1, 1);
     partDisplayer_5.initialize();

// Change the name of the linePart.

linePart_0.setPresentationName("X = 0.04");
```

This section is pruned to remove the redundant code and placed within a loop. Code shown as underlined is replaced by appropriate variables that are incremented within the loop.

The redundant code shown above is dealing with a `PartDisplayer` from `scene_0`, and has nothing to do with actually creating the line part. This can all safely be omitted.

Adding the Loop

Add a loop to the macro that renames the line probes that are created.

The additional code to be added to the macro is described in plain language as follows:

- Set the minimum X coordinate, XMIN.
- Set the maximum X coordinate, XMAX.
- Set the required X distance between line probes, XDIST.

- Initialize the current X location, XLOC = XMIN + XDIST.
- WHILE (XLOC is less than XMAX)
  - Create a line probe at XLOC
  - Rename the new line probe with XLOC in the title
  - XLOC = XLOC + XDIST
- END WHILE

To avoid having to set XMIN and XMAX directly in the macro, use the `getRegionExtents` method of the `Representation` class to set appropriate values. The method in the Java API documentation for the `Representation` class, shows that it requires the `Region` object for which extents are to be obtained.

In your editor window, replace the section of code considered previously with the modified code that now follows. Original code is shown in plain text and new or modified code is shown in bold. Enter the new code exactly as shown, as Java is a case-sensitive programming language.

```
// Start of execute0 function

Simulation simulation_0 =
  getActiveSimulation();

Units units_0 =
  simulation_0
  .getUnitsManager()
  .getPreferredUnits(
    new IntVector(
      new int[]
        {0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
    )
  );

Scene scene_0 =
  simulation_0
  .getSceneManager()
  .getScene("Scalar Scene 1");

PartDisplayer partDisplayer_1 =
  ((PartDisplayer) scene_0
  .getCreatorDisplayer());

FvRepresentation fvRepresentation_0 =
  ((FvRepresentation) simulation_0
  .getRepresentationManager()
  .getObject("Volume Mesh"));

partDisplayer_1
  .setRepresentation(fvRepresentation_0);

Region region_0 =
```

```
  simulation_0
  .getRegionManager()
  .getRegion("Fluid");

// New code begins here
// Define preliminary variables

double dXDIST = 0.1; // Distance between probes

// Use the FvRepresentation to find the extents of region_0
// (which was retrieved from simulation_0 earlier).
// FvRepresentation is derived from the Representation
// class and provides it's own version of the
// getRegionExtents method specifically for volume
// regions.
// getRegionExtents returns an array of six doubles:
// [xmin,xmax,ymin,ymax,zmin,zmax]

double[] dExtents =
  fvRepresentation_0
  .getRegionExtents(region_0);

// Retrieve xmin and xmax from dExtents

double dXMIN = dExtents[0];
double dXMAX = dExtents[1];

// Set the initial X location

double dXLOC = dXMIN + dXDIST;

// Start the loop
while(dXLOC < dXMAX) {
// Comes the section from the recorded macro
// Indent it further by two spaces and change
// all references to 0.04 with the variable dXLOC

LinePart linePart_0 =
  simulation_0
  .getPartManager()
  .createLinePart(
    new NeoObjectVector(
      new Object[] {region_0}
  ),
  new DoubleVector(
    new double[] {dXLOC, 0.0, 0.0}
  ),
  new DoubleVector(
```

```
  new double[] {dXLOC, 0.0762, 0.0}
  ),
  20
);

LabCoordinateSystem labCoordinateSystem_0 =
  simulation_0
  .getCoordinateSystemManager()
  .getLabCoordinateSystem();

linePart_0
  .setCoordinateSystem(labCoordinateSystem_0);

Coordinate coordinate_0 =
  linePart_0.getPoint1Coordinate();

coordinate_0
.setCoordinate(
  units_0,
  units_0,
  units_0,
  new DoubleVector(
    new double[] {dXLOC, 0.0, 0.0}
  )
);

// The next two sections set the units on each axis,
// and the position, of the second Point of linePart_0.

Coordinate coordinate_1 =
  linePart_0
  .getPoint2Coordinate();

coordinate_1
.setCoordinate(
  units_0,
  units_0,
  units_0,
  new DoubleVector(
    new double[] {dXLOC, 0.0762, 0.0}
  )
);

// Redundant sections removed from here

// Change the name of the linePart.
// The new name is created using the current X
// location. The String.format method ensures
```

```
// a consistent appearance of the number. See
// the Java documentation for details.

String strPartName =
  String.format("X = %6.4f",dXLOC);

linePart_0.setPresentationName(strPartName);

// Update dXLOC and end the loop with a closing brace
dXLOC = dXLOC + dXDIST;
} // End of Loop
```

This completes the main section for creating line probes.

Displaying the New Parts

The remaining change is to display all of the line probes in the scalar scene. This is achieved by obtaining all of the derived parts from the `PartManager` and setting them on the appropriate displayer of object `scene_0` obtained previously. Delete the redundant line of code which is underlined below:

```
// Following line is redundant: comment it out as shown
// or else delete it.

// partDisplayer_2.getParts().setObjects();

// Following section should remain unchanged; it is
// retrieving the relevant displayer from scene_0

ScalarDisplayer scalarDisplayer_0 =
  ((ScalarDisplayer)
    scene_0
    .getDisplayerManager()
    .getDisplayer("Scalar 1")
  );

// Add more code to retrieve all Derived Parts
// For more information on the Java technology, see
// the Java documentation for Collections and Generics

PartManager partManager_0 =
  simulation_0.getPartManager();

Collection<Part> allDerivedParts =
  partManager_0.getObjects();

scalarDisplayer_0
```

```
.getParts()
.setObjects(allDerivedParts);

} // End of execute0()
} // End of createLineProbes
```

This completes modification of the recorded macro.

- Save the macro in your editor.

Reviewing the Edited Macro

The final macro is shown below:

```
// Start of execute0 function

Simulation simulation_0 =
  getActiveSimulation();

Units units_0 =
  simulation_0
  .getUnitsManager()
  .getPreferredUnits(
    new IntVector(
      new int[]
        {0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
      )
    );

Scene scene_0 =
  simulation_0
  .getSceneManager()
  .getScene("Scalar Scene 1");

PartDisplayer partDisplayer_1 =
  ((PartDisplayer) scene_0
  .getCreatorDisplayer());

FvRepresentation fvRepresentation_0 =
  ((FvRepresentation) simulation_0
  .getRepresentationManager()
  .getObject("Volume Mesh"));

partDisplayer_1
.setRepresentation(fvRepresentation_0);

Region region_0 =
  simulation_0
```

```
  .getRegionManager()
  .getRegion("Fluid");

double dXDIST = 0.1; // Distance between probes

double[] dExtents =
  fvRepresentation_0
  .getRegionExtents(region_0);


double dXMIN = dExtents[0];
double dXMAX = dExtents[1];

double dXLOC = dXMIN + dXDIST;

// Start the loop
while(dXLOC < dXMAX) {
  LinePart linePart_0 =
    simulation_0
    .getPartManager()
    .createLinePart(
      new NeoObjectVector(
        new Object[] {region_0}
      ),
      new DoubleVector(
        new double[] {dXLOC, 0.0, 0.0}
      ),
      new DoubleVector(
        new double[] {dXLOC, 0.0762, 0.0}
      ),
      20
    );

LabCoordinateSystem labCoordinateSystem_0 =
  simulation_0
  .getCoordinateSystemManager()
  .getLabCoordinateSystem();

linePart_0
  .setCoordinateSystem(labCoordinateSystem_0);

Coordinate coordinate_0 =
  linePart_0.getPoint1Coordinate();

coordinate_0
.setCoordinate(
  units_0,
  units_0,
```

```
  units_0,
  new DoubleVector(
    new double[] {dXLOC, 0.0, 0.0}
  )
);

Coordinate coordinate_1 =
  linePart_0
  .getPoint2Coordinate();

coordinate_1
.setCoordinate(
  units_0,
  units_0,
  units_0,
  new DoubleVector(
    new double[] {dXLOC, 0.0762, 0.0}
  )
);

String strPartName =
  String.format("X = %6.4f",dXLOC);

linePart_0.setPresentationName(strPartName);

// Update dXLOC and end the loop with a closing brace
dXLOC = dXLOC + dXDIST;
} // End of Loop

// partDisplayer_2.getParts().setObjects();

ScalarDisplayer scalarDisplayer_0 =
  ((ScalarDisplayer)
    scene_0
    .getDisplayerManager()
    .getDisplayer("Scalar 1")
  );

PartManager partManager_0 =
  simulation_0.getPartManager();

Collection<Part> allDerivedParts =
  partManager_0.getObjects();

scalarDisplayer_0
.getParts()
.setObjects(allDerivedParts);
```

```
  } // End of execute0()
} // End of createLineProbes
```

Running the Macro

Run the macro in the STAR-CCM+ client.

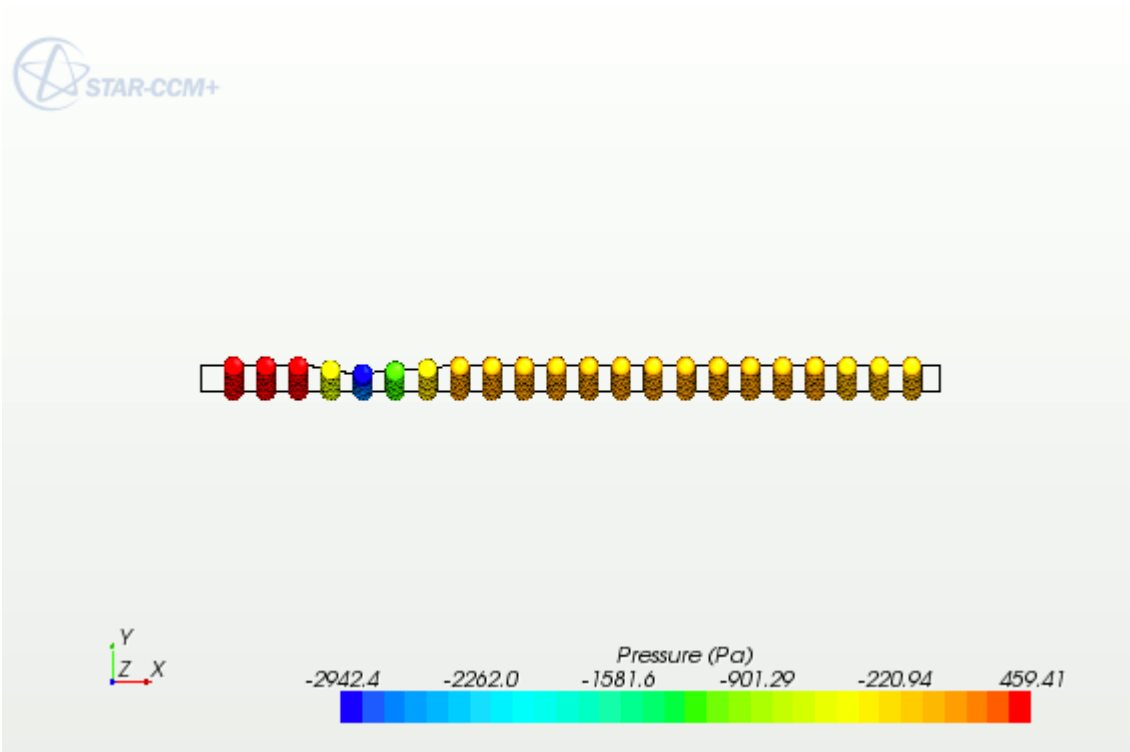
Before running the macro, delete the single-line probe created earlier:

1. Right-click the **Derived Parts** > **X = 0.04** node and select **Delete** from the pop-up menu.
2. Click **Yes** in the *Confirm Object Deletion* dialog that appears.

Run the macro:

3. In the *Macro* toolbar, click (**Play Macro...**).
4. In the *Open* dialog, select `createLineProbes.java`.
5. Click **Open**.

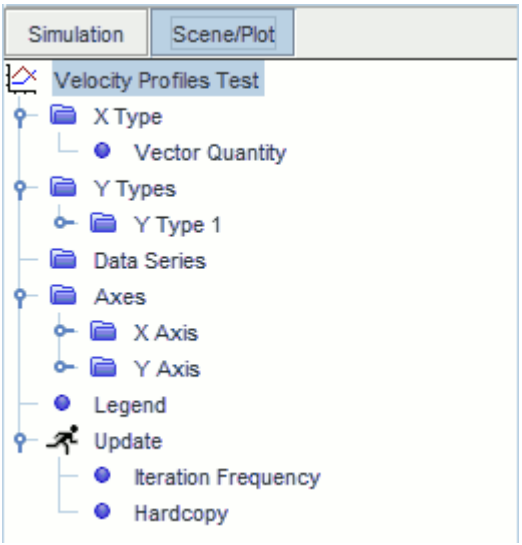
STAR-CCM+ first compiles the macro class, and if there are no errors reported, it calls the **execute** method. This results in the creation of 22 line probes, from **X = 0.1000** to **X = 2.2000**, which appear in *Scalar Scene 1*.



Plotting Data Using Line Probes

Use the line probes to create an X-Y plot of axial velocity profiles. Each data series presented on the plot corresponds to the data extracted along a single-line probe. The data contained in the plot are exported to a text file.

1. Right-click the **Plots** node and select **New Plot** > **X-Y** from the pop-up menu.
2. Rename the **Plots** > **XY Plot 1** node to `Velocity Profiles`.
3. Switch to the *scene/plot* pane by clicking the **scene/plot** button above the object tree.



The X-axis of the plot shows the Y position data from the line probe, and the Y-axis of the plot shows the axial velocity at those Y positions.

4. Edit the **Velocity Profiles** node and set the following properties:

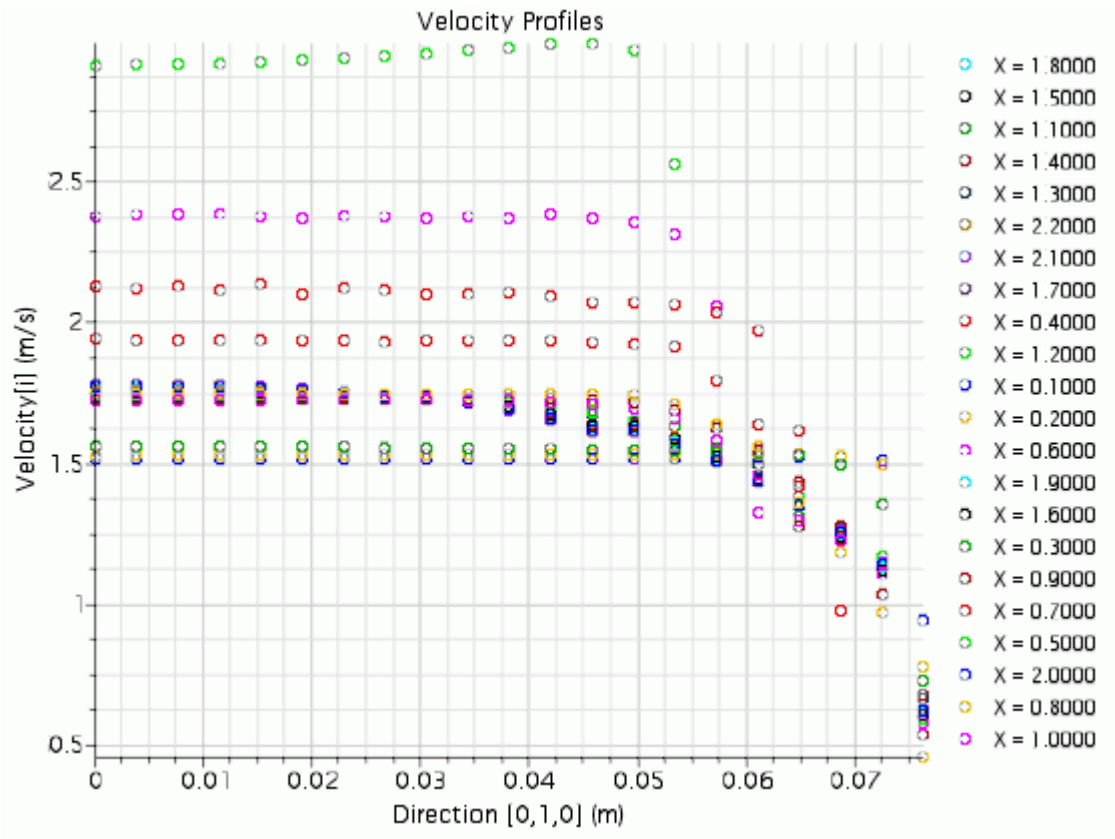
Node	Property	Setting
Velocity Profiles	*Title*	`Velocity Profiles`
└ **X Type** > **Vector Quantity**	*Value*	`[0.0, 1.0, 0.0] m`
└ **Y Type** > **Y Type 1** > **Scalar Function**	*Field Function*	**Velocity** > **Lab Reference Frame** > **Laboratory** > **i**
└ **Legend**	*Position*	**East**
	Layout	**Vertical**
	Font	**Lucida Sans Plain 8**

The final operation is to set the **Parts** property for the plot.

5. Select the **Velocity Profiles** node at the top of the tree.
6. In the *Properties* window, edit the *Parts* property.
7. In the *Velocity Profiles - Parts* dialog, select the **Derived Parts** node.
8. Expand the node and make sure that all parts are selected.

9. Click **OK**.

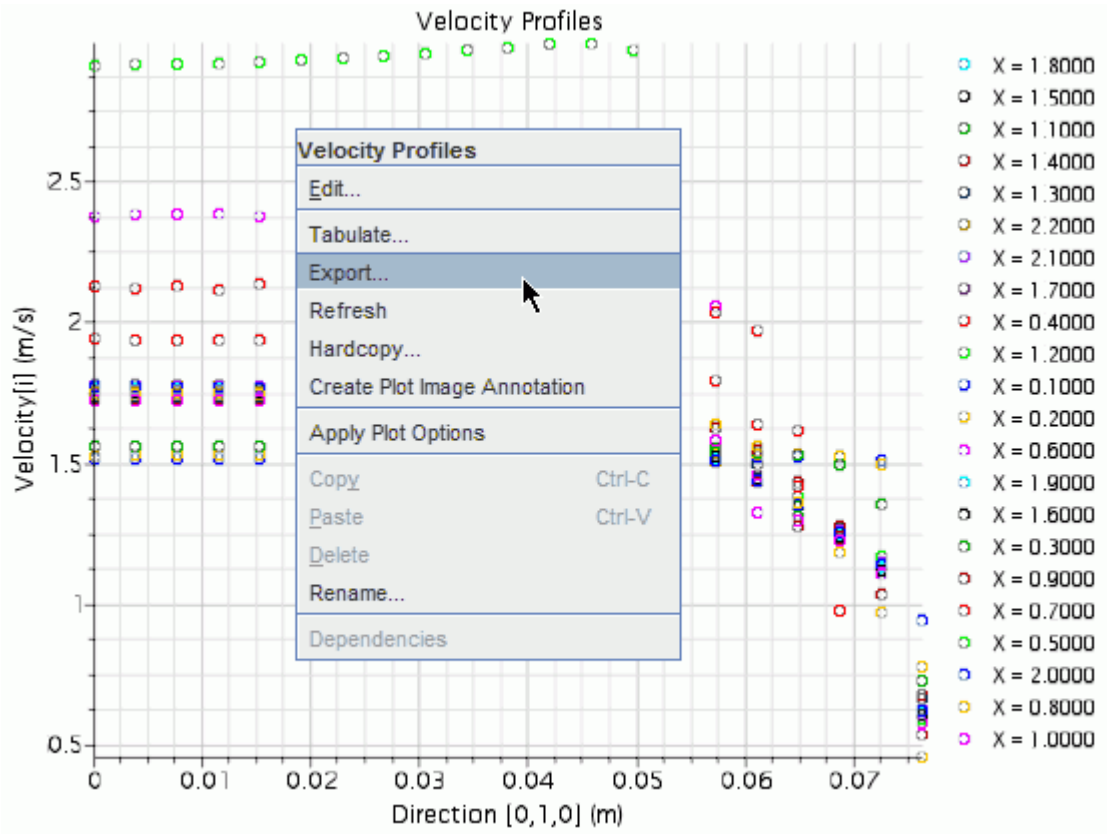
The final plot is displayed in the *Velocity Profiles* window.



Exporting XY Plot Data

The data contained in the *Velocity Profiles* plot can be exported to a text file.

1. Right-click the background of the *Velocity Profiles* plot and select **Export** from the pop-up menu.



2. In the *Save* dialog, set *File Name* to `exported_velocities`.
3. Click **Save**.

All data contained in the *Velocity Profiles* plot are written to a Comma-Separated Variable (CSV) file. This type of file can be imported into spreadsheet software for further analysis of the data.

Further Training

Siemens PLM Software provides an in-depth training course on the use of Java macros and the automation of tasks in STAR-CCM+. Contact your local support agent for details.

Summary

This tutorial has introduced the following STAR-CCM+ features:

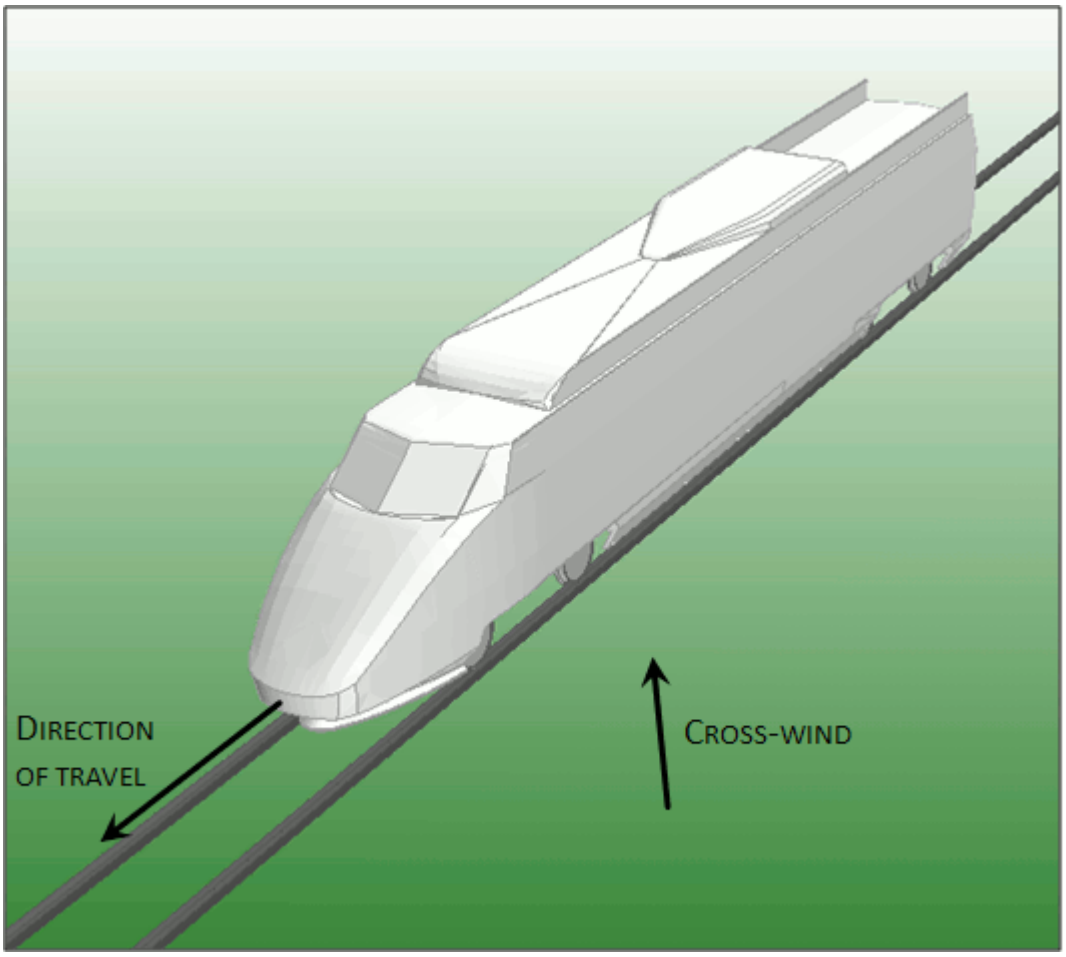
- Importing a three-dimensional mesh intended for use in a two-dimensional simulation.
- Converting a three-dimensional mesh to two-dimensions.
- Cleaning up regions and boundaries after the conversion.
- Splitting a boundary by angle and renaming the subsequent nodes.
- Selecting the axisymmetric space model.
- Setting an axial boundary type and projecting it to the Y axis.
- Recording a macro while creating a single-line probe and adding it to a scalar scene.

- Editing a previously recorded macro to add a loop.
- Including code to add all derived parts to a scalar scene.
- Running a macro.
- Creating an XY plot based on line probes.
- Exporting the data from an XY plot to a CSV file.

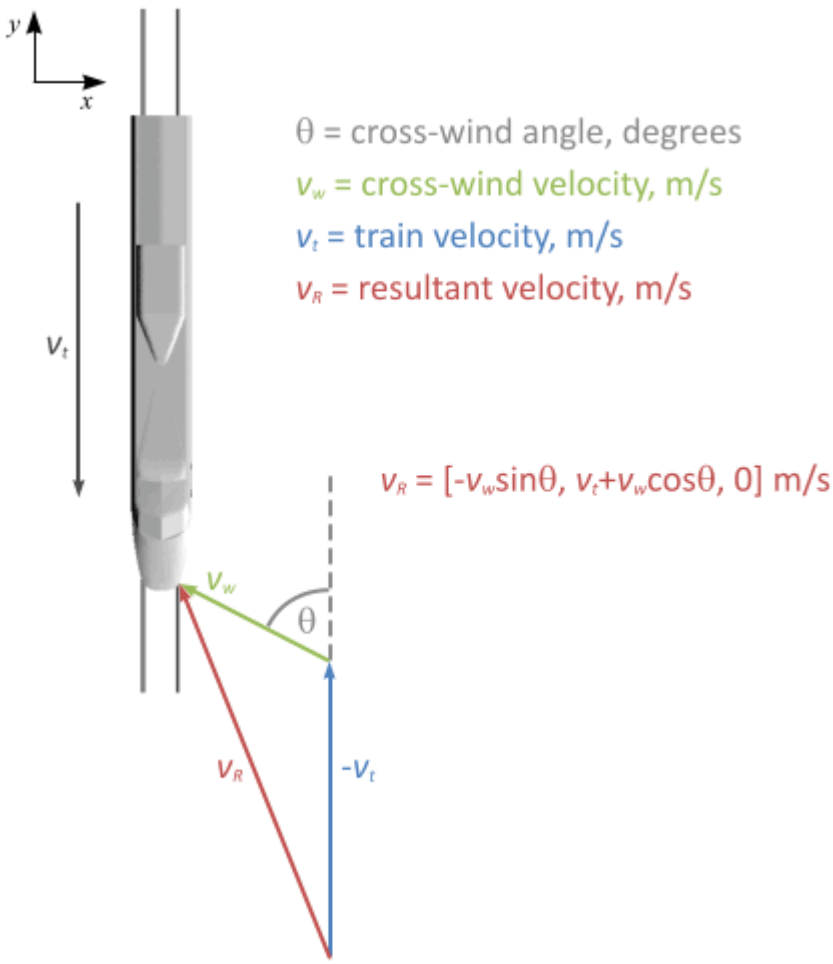
Intermediate Java Macros: Run Multiple Simulations

This tutorial simulates a train in a cross-wind, to calculate the drag coefficient on the train for different angles of attack.

One way in which Java macros can be used effectively is to run a number of simulations in which a single variable or a set of variables is altered in each one.



The train is modeled under field test conditions. The resultant velocity of the wind as seen from the train, is calculated based on a 22 m/s head wind and a 6 m/s cross-wind at angle of attack. In addition, you set a constant velocity on the track that is equivalent to the velocity of the train. The following diagram shows how the resultant velocity is calculated:



A Java macro runs a set of simulations consecutively, each with a different set of boundary conditions. The macro performs the following:

Read the boundary condition values from an external text file;

Set the new boundary conditions in each simulation;

Run the simulation;

Export data, scenes, and plots for post-processing.

The tutorial introduces the use of multiple classes to organize your macro and includes a fully commented Java file for reference purposes.

Contents:

Prerequisites
Importing the Surface Mesh and Naming the Simulation
Visualizing the Initial Surface
Assigning Parts to Regions
Setting Up the Mesh
Specifying Boundary Types

Generating the Mesh
Setting Up the Physics Continuum
Setting Global Initial Conditions
Specifying Inflow and Outflow Conditions
Creating a Drag Coefficient Report
Creating a Velocity Magnitude Scalar Scene
Creating a Scalar Scene with Streamlines
Designing the Macro
Recording the Initial Macro
Creating the Final Macro
Testing and Debugging
Running the Macro
Summary

Prerequisites

The instructions in the Intermediate Java Macros tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+
Recording and editing Java macros	Simple Java Macros: Post Processing

Importing the Surface Mesh and Naming the Simulation

To set up the STAR-CCM+ simulation, launch a simulation and import the supplied surface mesh.

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Surface Mesh...**
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/automation/data`.
5. Select `trainAndTrack.dbs` then click **Open**.
6. Click **OK**.
7. Save the simulation as `train.sim`.

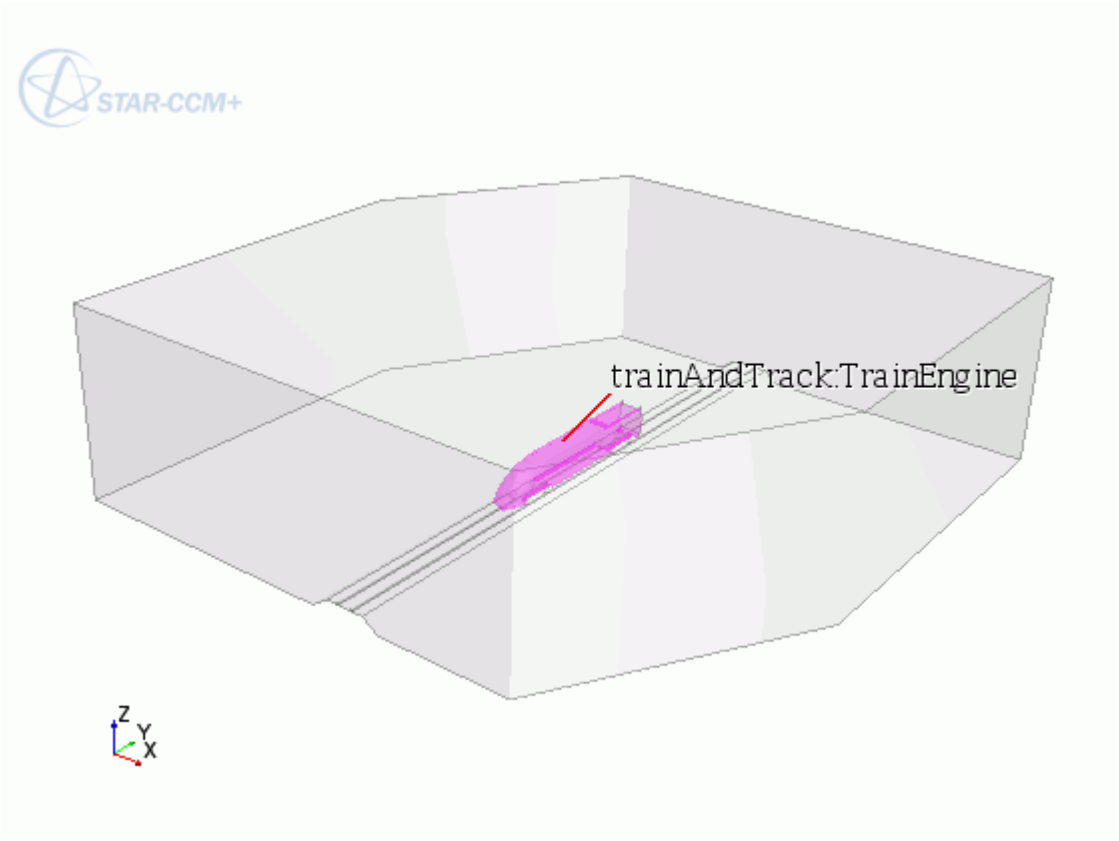
Visualizing the Initial Surface

Visualize the initial surface of the geometry.

In this example, all surfaces are contained within a single part, **trainAndTrack**. During the initial model preparation, the surface was divided into separate surfaces and these were given appropriate names.

1. Click (**Make Scene Transparent**).
2. Expand the **Geometry** > **Parts** > **trainAndTrack** > **Surfaces** node.
3. Select the **TrainEngine** node.

   The **TrainEngine** surface is highlighted on the model.



4. Select each of the surfaces and observe which part is highlighted.

Remove two of the surfaces from the current scene to view the mesh on the internal features.

5. Edit the **Displayers** > **Geometry 1** > **Parts** node.
6. In the *Edit* dialog, expand the **Surfaces** nodes and deactivate the **Inflow** and **Top** surfaces.
7. Click **OK**.

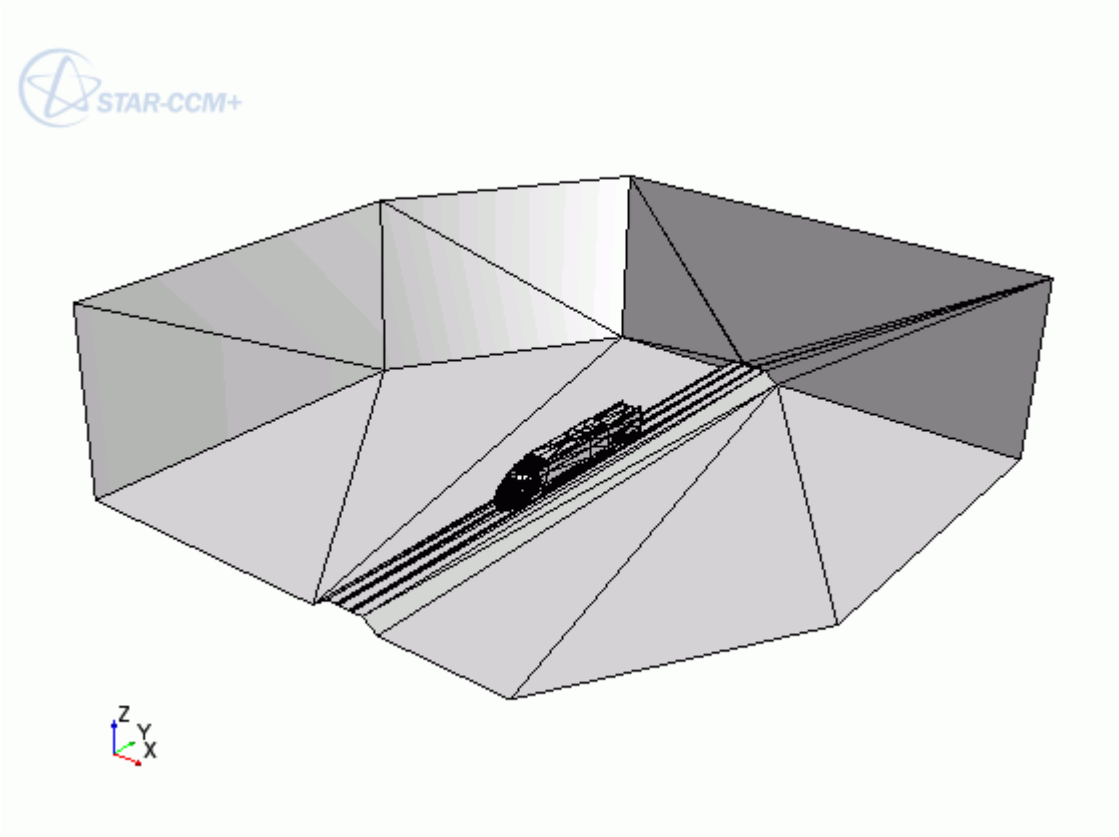
It is possible to toggle the visibility of any displayer in a scene to disable a particular displayer without deleting it from a scene.

8. Right-click the **Outline 1** node and select **Toggle Visibility**.

   The outline is no longer displayed in the scene.

9. In the *Vis* toolbar, click (**Show All Meshes**) to display the mesh.

10. In the *Vis* toolbar, click (**Make Scene Transparent**) to restore the scene transparency.



This type of mesh is generated when neutral format data or native CAD data is imported into STAR-CCM+, and is known as a tessellated surface mesh.

Assigning Parts to Regions

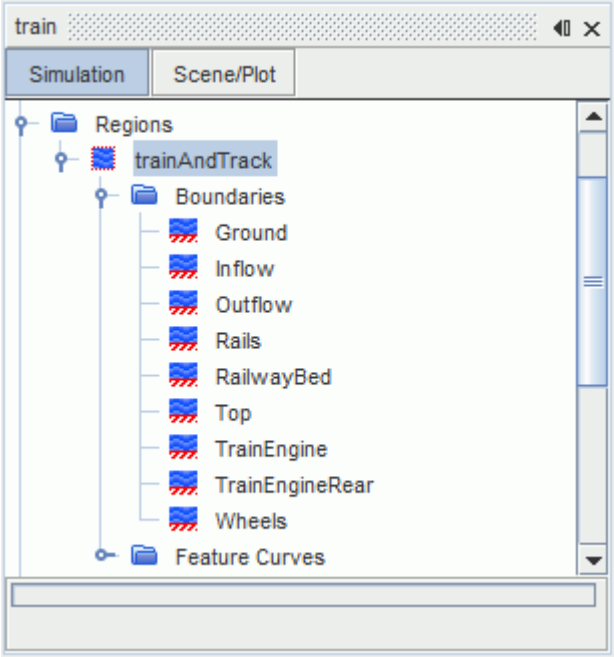
Before setting up the mesh for this tutorial, assign the imported part to a region.

1. Right-click the **Geometry** > **Parts** > **trainAndTrack** node and select **Assign Parts to Regions...**
2. In the *Assign Parts to Regions* dialog, set the modes as follows:

Mode	Setting
Region mode	**Create a Region for Each Part**
Boundary mode	**Create a Boundary for Each Part Surface**

3. Click **Apply** then **Close**.

   STAR-CCM+ adds a new region, **trainAndTrack**, within the **Regions** node. All the part surfaces are assigned to the corresponding region boundaries.



Setting Up the Mesh

To calculate the drag coefficient of the train as it is subjected to cross-winds at different angles accurately there must be enough cells near to the train, which requires a relatively fine mesh.

You can minimize the total number of cells in the mesh by generating a fine mesh around the train, and having a coarse mesh in the remainder of the fluid domain. Using different mesh densities is a good practice for mesh generation, promoting faster solution convergence, and effective use of processing power. In external aerodynamics cases, it is common to have a high mesh cell count (>1 million cells). However, the mesh generated in this tutorial is much lower to decrease the run-time.

Contents:

Selecting the Meshers
Setting the Mesh Default Controls
Setting up Wake Refinement
Specifying Custom Surface Controls

Selecting the Meshers

Create an Automated Mesh operation to select the meshers for the imported geometry.

Use the **Surface Remesher** to generate the surface mesh. Use the **Trimmed Cell Mesher** and **Prism Layer Mesher** to generate the volume mesh.

1. Right-click the **Geometry** > **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
2. In the *Create Mesh Operation* dialog:
   a) From the *Parts* list, select **trainAndTrack**.

b) From the *Select Meshers* group box, select the following meshers:

Group Box	Mesher
Surface Meshers	**Surface Remesher**
Core Volume Meshers	**Trimmed Cell Mesher**
Optional Boundary Layer Meshers	**Prism Layer Mesher**

c) Click **OK**.

Setting the Mesh Default Controls

Set the size of the coarser region of the mesh.

1. Edit the **Operations** > **Automated Mesh** > **Default Controls** node and set the following properties:

Node	Property	Setting
Base Size	*Value*	`5 m`
Prism Layer Total Thickness	*Percentage of Base*	`1`
Volume Growth Rate	*Default Growth Rate*	**Medium**
	Surface Growth Rate	**Medium**

Setting up Wake Refinement

Use the wake refinement tool to refine the mesh in the region behind the train by specifying a surface from which the wake extends. In this case, the surface is **TrainEngineRear**.

1. Right-click the **Operations** > **Automated Mesh** > **Custom Controls** node and select **New** > **Surface Control**.
2. Rename the **Surface Control** you just created to `TrainEngineRear_Wake`.
3. Select the **TrainEngineRear_Wake** node and set *Part Surfaces* to **trainAndTrack** > **Surfaces** > **TrainEngineRear**.

Specify the properties of the mesh in the region of the wake:

4. Expand the **TrainEngineRear_Wake** > **Controls** node.
5. Select the **Trimmer Wake Refinement** node and activate *Specify Trimmer Wake Refinement Options*.
6. Edit the **Values** node and set the following properties:

Node	Property	Setting
Wake Refinement	*Distance*	`5.0 m`
	Direction	`[0, 1, 0]`
└ **Relative Size**	*Percentage of Base*	`6`

Specifying Custom Surface Controls

Apply custom mesh settings to the train engine, wheels, and rails boundaries to make sure a suitably fine mesh around the train and the rails is generated.

Create a Surface Control to define custom mesh settings for the engine and wheels:

1. Right-click the **Automated Mesh** > **Custom Controls** node and select **New** > **Surface Control**.
2. Rename **Surface Control** to `Engine and Wheels`.
3. Edit the **Engine and Wheels** node and set the following properties:

Node	Property	Setting
Engine and Wheels	*Part Surfaces*	**TrainEngine**
		Wheels
		TrainEngineRear
└ **Controls**		
└ **Target Surface Size**	*Target Surface Size*	**Specify Custom Value**
└ **Minimum Surface Size**	*Minimum Surface Size*	**Specify Custom Value**
└ **Values**		
└ **Target Surface Size**	*Percentage of Base*	3
└ **Minimum Surface Size**	*Percentage of Base*	3

In a similar way, specify a custom surface size for the **Rails** surface:

4. Create a new **Surface Control** and rename it to `Rails`.
5. Edit the **Rails** node and set the following properties:

Node	Property	Setting
Rails	*Part Surfaces*	**Rails**
└ **Controls**		
└ **Target Surface Size**	*Target Surface Size*	**Specify Custom Value**
└ **Minimum Surface Size**	*Minimum Surface Size*	**Specify Custom Value**
└ **Values**		
└ **Target Surface Size**	*Percentage of Base*	7
└ **Minimum Surface Size**	*Percentage of Base*	7

Specifying Boundary Types

Specify the boundary types for the simulation.

It is good practice to set boundary types before generating the mesh. The default boundary type is Wall, and when the Prism Layer Mesher is used, prism layers grow from wall boundaries. Prism layers are not generated for non-wall boundaries unless they are explicitly chosen using the **Mesh Conditions** node for each boundary.

1. Set the following boundary types:

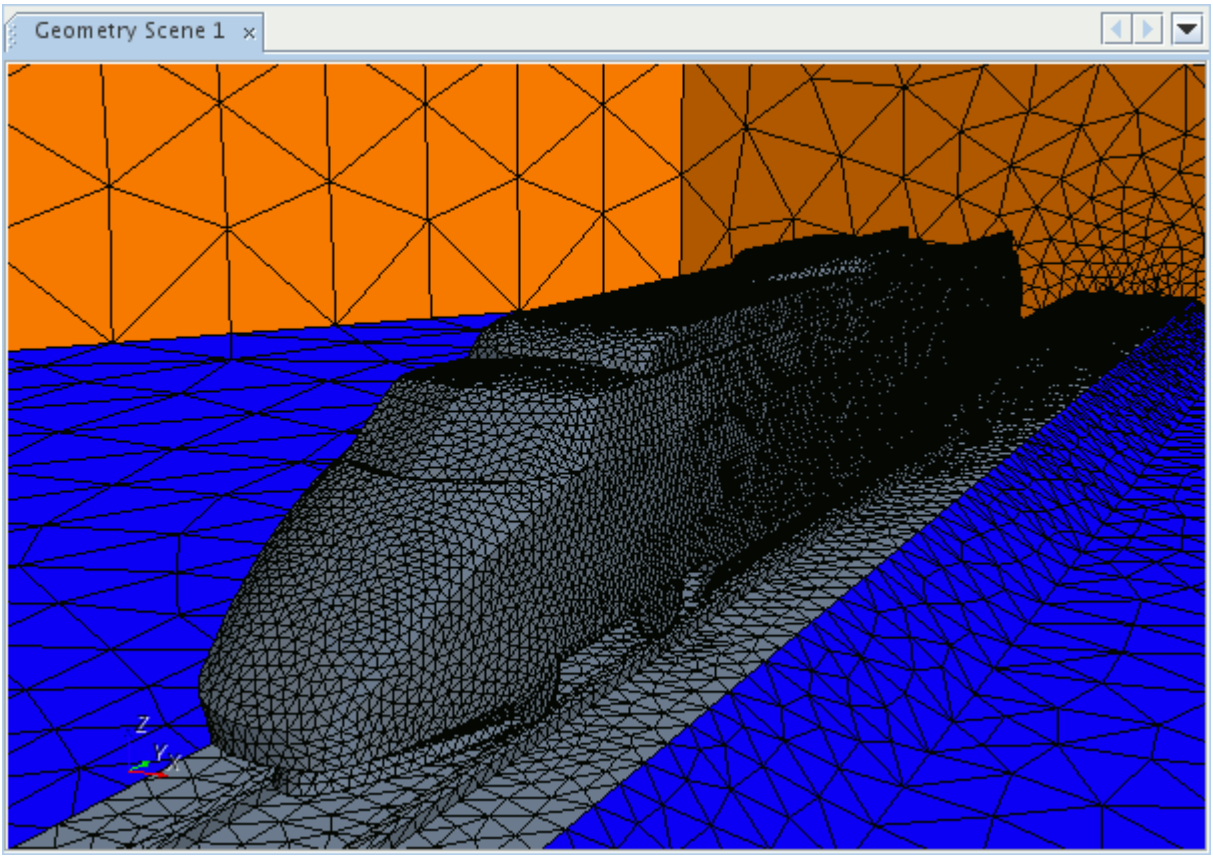
Boundary	Type
Ground	**Symmetry Plane**
Top	**Symmetry Plane**
Inflow	**Velocity Inlet**
Outflow	**Pressure Outlet**

2. Save the simulation.

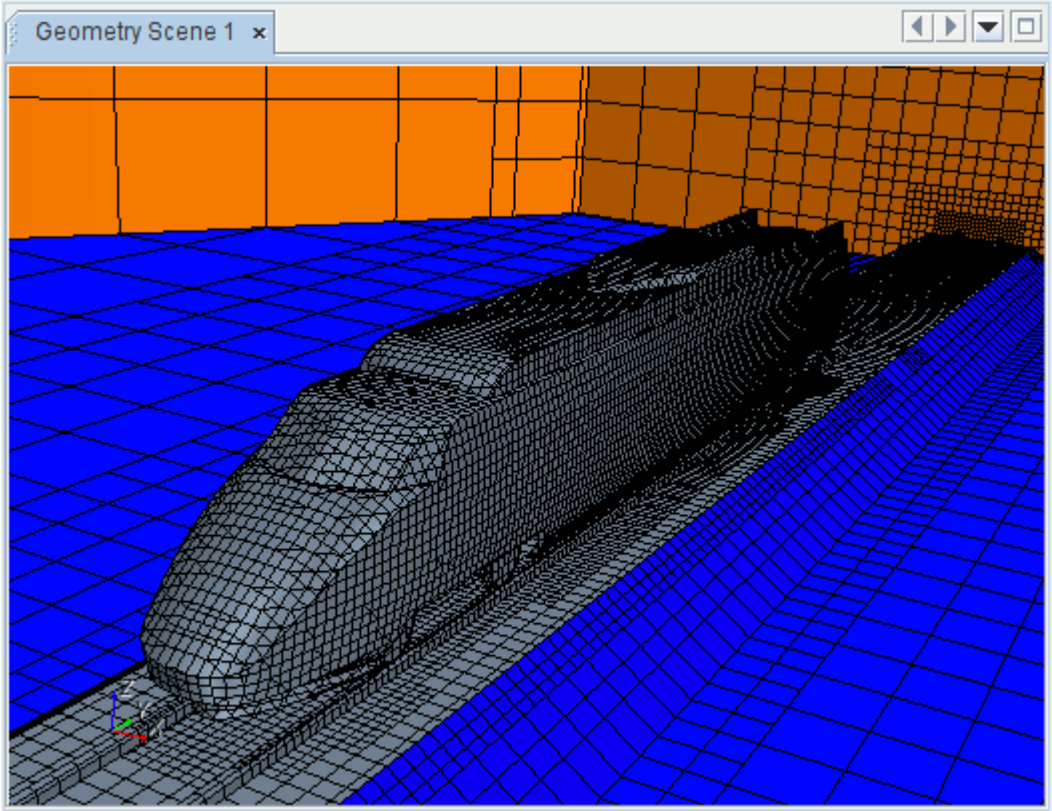
Generating the Mesh

The mesh parameters are fully defined and you can proceed to generate the mesh. It is good practice to generate the surface mesh and examine that before proceeding with the volume mesh.

1. Click (**Generate Surface Mesh**).
   The surface mesh near the train is shown.



2. Click (**Generate Volume Mesh**).
3. Right click the **Scenes** > **Geometry Scene 1** node and select **Apply Representation** > **Volume Mesh**.



4. Save the simulation.

Setting Up the Physics Continuum

Physics models define the physical variables and phenomena in the simulation. In this tutorial, model the flow as Steady and Turbulent. Use the Segregated Flow solver with the K-Epsilon Turbulence model.

1. For the physics continuum, **Continua** > **Physics 1**, select the following models in order:

Group Box	Model
Time	**Steady**
Material	**Gas**
Flow	**Segregated Flow**
Equation of State	**Constant Density**
Viscous Regime	**Turbulent**
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence** **Exact Wall Distance** (selected automatically)
Optional Physics Models	**Cell Quality Remediation**

2. Click **Close**.
3. Save the simulation.

Setting Global Initial Conditions

Set the initial flow field conditions. Retain the default value for the initial pressure condition, and specify custom values for turbulence.

1. Select the **Physics 1** > **Initial Conditions** > **Turbulence Specification** node and set *Method* to **Intensity + Length Scale**.
2. Select the **Turbulent Length Scale** > **Constant** node and set the *Value* to `2.5 m`.

Specifying Inflow and Outflow Conditions

Set the boundary conditions on the inflow, outflow, rails, and railway bed boundaries. The inflow velocity is different for each cross-wind flow angle and so is set in the Java macro.

1. Expand the **Regions** > **trainAndTrack** > **Boundaries** node.
2. Edit the **Inflow** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Turbulence Specification**	*Method*	**Intensity + Length Scale**
└ **Velocity Specification**	*Method*	**Components**
Physics Values		
└ **Turbulent Length Scale** > **Constant**	*Value*	`2.5 m`

Note: You set the velocity specification method at the inlet to record the steps to the macro for later use.

3. Edit the **Outlfow** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Turbulence Specification**	*Method*	**Intensity + Length Scale**
└ **Backflow Specification**	*Direction*	**Extrapolated**
Physics Values		
└ **Turbulent Length Scale** > **Constant**	*Value*	`2.5 m`

Note: You change the backflow direction specification method at the **Outflow** boundary to allow the fluid to flow parallel to the outlet boundary.

4. Edit the **Rails** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Tangential Velocity Specification**	*Method*	**Vector**
Physics Values		
└ **Relative Velocity** > **Constant**	*Value*	`[0, 22, 0] m/s`

5. Edit the **RailwayBed** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Tangential Velocity Specification**	*Method*	**Vector**
Physics Values		
└ **Relative Velocity** > **Constant**	*Value*	`[0, 22, 0] m/s`

6. Save the simulation.

Creating a Drag Coefficient Report

Create a drag coefficient report to obtain the value for the drag coefficient on the train at each cross-wind flow angle. Solution convergence can be monitored using this report also.

1. Right-click the **Reports** node and select **New Report** > **Force Coefficient**.
2. Rename the **Force Coefficient 1** node to `Drag Coefficient`.
3. Edit the **Drag Coefficient** node and set the following properties:

Property	Setting
Reference Velocity	`22 m/s`
Reference Area	`10 m^2`
Direction	`[0, 1, 0]`
Parts	**Regions** > **TrainEngine** **Regions** > **TrainEngineRear** **Regions** > **Wheels**

4. Right-click the **Drag Coefficient** node and select **Create Monitor and Plot from Report**.

A drag coefficient monitor is created along with a plot to show the drag coefficient calculated after each iteration.

Creating a Velocity Magnitude Scalar Scene

Create a scalar scene to visualize the velocity magnitude.

1. Create a section plane with the following properties:

Property	Setting
origin	`[0, 0, 0.8] m`
normal	`[0, 0, 1] m`
Display	**No Displayer**

2. Click **Create** then **Close**.
3. Create a scalar scene and rename it to `Velocity Magnitude`.
4. Click **scene/plot**.
5. Edit the **Velocity Magnitude** > **Displayers** node and set the following properties:

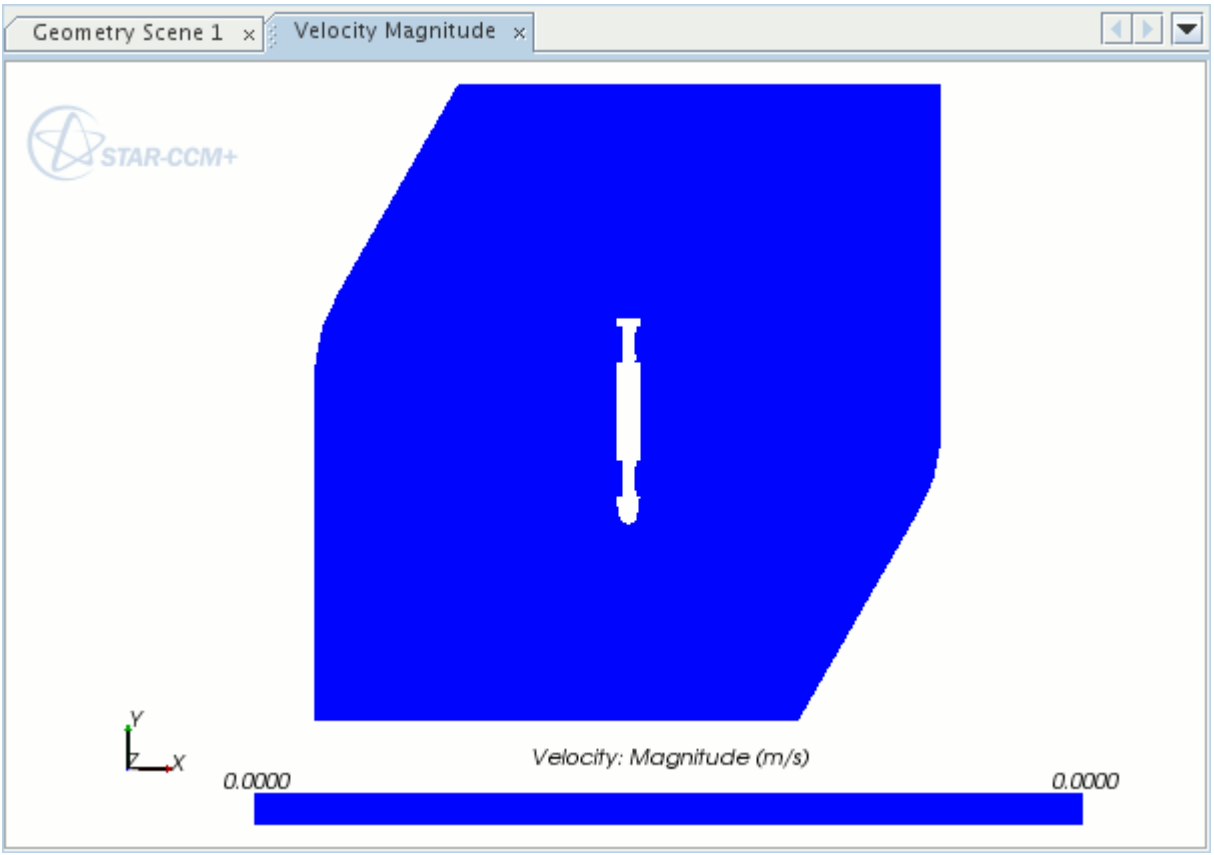
Node	Property	Setting
Outline 1	*Outline*	**Deactivated**
Scalar 1	*Contour Style*	**Smooth Filled**
└ **Parts**	*Parts*	**Derived Parts** > **plane section**
└ **Scalar Field**	*Function*	**Velocity Magnitude**

6. Click **Close**.
7. Initialize the solution.

   By initializing manually you are able to view the initialized solution, which helps in the following steps.

Adjust the position of the components in the scene. It is useful to adjust the scene at this stage before saving a hardcopy using the Java macro.

8. Position the scene to match the screenshot:



9. Click **simulation**.
10. Save the simulation.

Creating a Scalar Scene with Streamlines

Create a second scalar scene to display streamlines over the train.

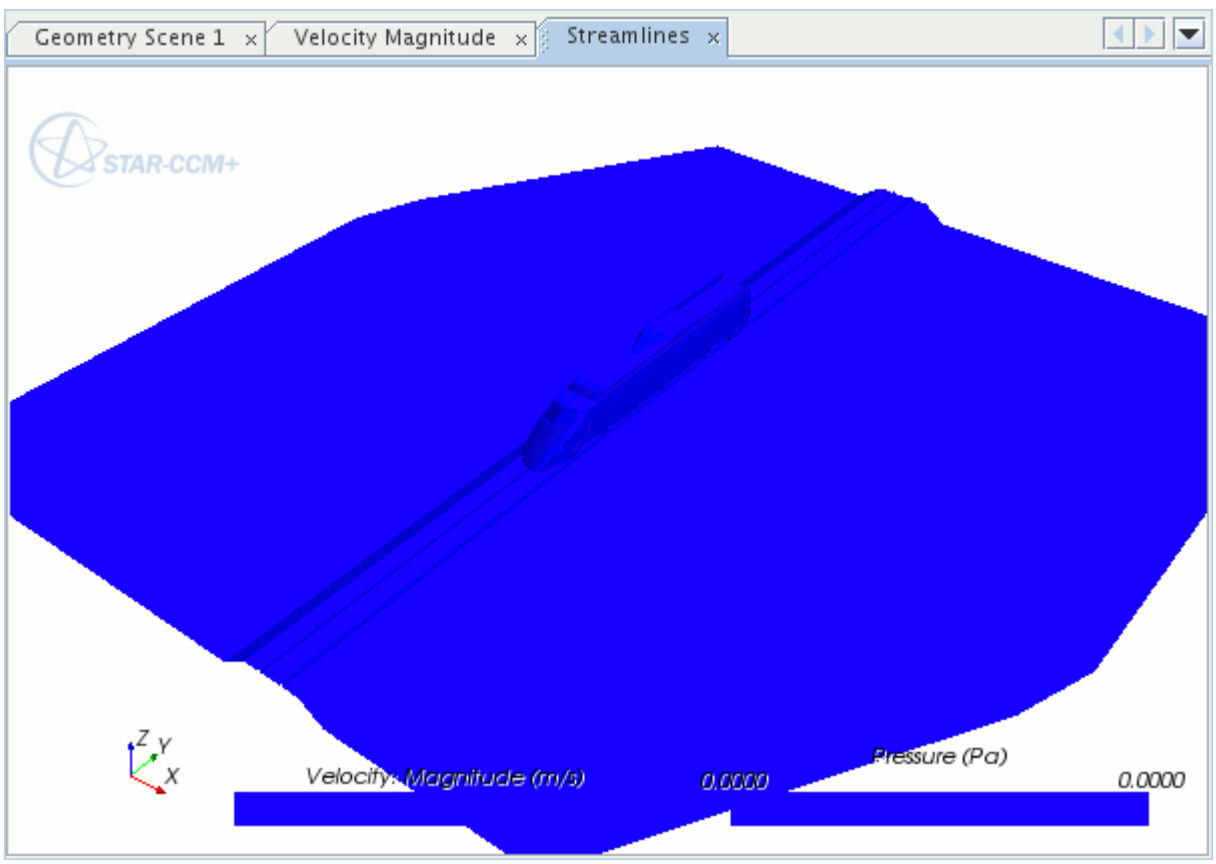
1. Create a scalar scene and rename it to `Streamlines`.
2. Create a streamline derived part with the following properties:

Property	Setting
Seed Mode	**Line Seed**
Point 1	`[-7, -73, 0.4] m`
Point 2	`[20, -52, 0.4] m`
Resolution	`50`
Direction	**Both**

3. Click **Create** then **Close**.
4. Click **scene/plot**.
5. Edit the **Streamlines** > **Displayers** node and set the following properties:

Node	Property	Setting
Outline 1	*Outline*	**Deactivated**
Scalar 1	*Contour Style*	**Smooth Filled**
└ **Parts**	*Parts*	**Ground** **Rails** **RailwayBed** **TrainEngine** **TrainEngineRear** **Wheels**
└ **Scalar Field**	*Function*	**Pressure**
└ **Color Bar**	*Color Map*	**grayscale**
Streamlines Stream 1 > **Scalar Field**	*Scalar Field*	**Velocity Magnitude**

6. Click **Close**.
7. Adjust the scene so that it appears as follows:



8. Close the *Streamlines* scene.

9. Click **simulation**.
10. Save the simulation.

The case is now set up correctly and you are ready to begin Designing the Java Macro.

Designing the Macro

Carry out the following steps when designing a macro:

- Define the aims of the macro
- Create a simple plan for the macro
- Plan the actions to be recorded in a macro
- Record a macro in STAR-CCM+
- Examine the recorded macro and pick out useful statements
- Finalize the structure of the macro
- Write the final macro

Contents:

Defining the Aims of the Macro
Creating a Simple Plan

Defining the Aims of the Macro

The first step in designing a Java macro is to define in one or two sentences, what you want the macro to do.

For the tutorial, the aim of the macro is to run a set of simulations with different cross-wind angles of attack, and provide the value of the drag coefficient on the train.

Creating a Simple Plan

With the aims of the macro clearly defined, you can start planning.

First, write down the steps required. For example, set the velocity to 5 m/s and run the simulation. These steps form the basis of a plan. Next, expand the plan to include other processes such as extracting data from an input file, looping through to repeat a set of actions, and opening a panel. Write the plan in simple terms.

To simplify the process of scripting the macro, you can record a simple Java macro in STAR-CCM+ and extract sections of the code.

To meet the aims that were defined previously, the following steps are required (processes that can be obtained by recording a macro in STAR-CCM+ are marked with a *):

- Read cross-wind data from an external source
- Clear any previous solution*
- Set the inlet velocity value*
- Set the initial velocity value*

- Run the simulation for some iterations*
- Calculate the drag coefficient on the train*
- Save the drag coefficient value for each cross-wind angle of attack in an external data file
- Save the simulation*
- Repeat for each cross-wind angle of attack

You have marked a number of actions for which the Java code can be obtained by recording a macro.

Recording the Initial Macro

Before you record the initial macro, it is good practice to have a clear plan of which actions you perform when recording the macro. The resulting Java file is clearer and less cluttered, and identifying which statements refer to a particular action is much easier.

The following sections cover:

- How to plan the actions to record
- Recording a macro
- Understanding the recorded macro

Contents:

Planning Actions to Record
Recording the Macro
Understanding the Initial Macro

Planning Actions to Record

At this stage you can consider any additional actions to perform in the final macro, such as exporting plots and scenes.

Include additional steps in the plan to help with post-processing the results of each simulation. Export hardcopies of the Residuals plot and *Velocity Magnitude* scalar scene, as well as a STAR-View+ file of the *Streamlines* scene. Using this technique, the results of each simulation can be compared without needing to open each one individually.

Record the following actions:

1. Set the initial velocity
2. Set the inflow velocity
3. Clear any previous solution
4. Run for x iterations
5. Get a value for the drag coefficient
6. Save a hardcopy of the *Velocity Magnitude* scene and Residuals plot
7. Save a STAR-View+ file of the *Streamlines* scene
8. Save the simulation

Recording the Macro

Record the planned actions.

Any command issued to STAR-CCM+ has an associated Java statement. The **macroRecording** tab of the **Output** window displays the statements. Do not record unnecessary commands to avoid the Java file being cluttered. Pause the recording to carry out actions without the associated statements being added to the macro.

1. In the *Macro* toolbar, click **(Start Recording...)**.
2. Save the file as `macroRecording.java`.

Set the initial velocity:

3. Expand the **Continua** > **Physics 1** > **Initial Conditions** > **Velocity** > **Constant** node.
4. Set *Value* to `[-10, 10, 0] m/s`.

Set the inflow velocity:

5. Select the **Regions** > **trainAndTrack** > **Boundaries** > **Inflow** > **Physics Values** > **Velocity** > **Constant** node.
6. Set the *Value* to `[-5, 5, 0] m/s`.

Clear the previous solution and run the simulation:

7. From the menu, select **Solution** > **Clear Solution**.
8. Run the simulation.
9. Click (**Stop Iterating**) to stop the simulation running.

Run the **Drag Coefficient** report. When you create the final macro, obtain a single value for the drag coefficient rather than running the entire report. To obtain some of the statements required later on, however, run the report.

10. Right-click the **Reports** > **Drag Coefficient** node and select **Run Report**.

Save hardcopies of the scene:

11. Select the **Velocity Magnitude** tab in the **Graphics** window.
12. Right-click in the white area of the scene and select **Hardcopy...**.
13. In the *Save To File* dialog, set the following properties:

Property	Setting
File Name	`velMag`
X Resolution	`1024`
Y Resolution	`768`

14. Click **Save**.
15. Repeat these steps to save a hardcopy of the *Residuals* plot; name the file `res`.
16. Right-click the **Scenes** > **Streamlines** node and select **Export 3D Visualization File**.

17. Name the file `Streamlines` and click **Save**.
18. Click the **Save As** button in the **File** menu and save the file under the name `trainMacro`.
19. Click **Save**.
20. Click the ■ (**Stop Recording**) button in the *Macro* toolbar to stop recording.

Understanding the Initial Macro

After recording the macro, it is good practice to open the file and noting which statements correspond to which actions.

To add notes to the Java file type `//"and your comment"` . The following steps examine the initial macro and describe which statements correspond to the actions you recorded in the previous section.

Open the `macroRecording.java` file in an appropriate text editor. Refer to the first section of the Java macro. The first set of statements in the macro conforms to the template of a standard STAR-CCM+ macro.

Code	Explanation

`// STAR-CCM+ macro: macroRecording.java`
`package macro;`
 	
`import java.util.*;`
 	
`import star.common.*;`
`import star.base.neo.*;`
`import star.vis.*;`
`import star.base.report.*;`
`import star.flow.*;`
 	
`public class macroRecording extends StarMacro {`
 	
`  public void execute() {`
`  execute0();`
`}`
 	
`  private void execute0() {`
	• The class is defined as belonging to the `macro` package.
	• A number of libraries are imported which are required to compile the macro.
	• A class called `macroRecording` is declared, and is specified as a subclass of the `StarMacro` class.
	• The main method, `execute()`, is declared. When the macro is played in STAR-CCM+ it uses `execute()`, which calls `execute0()`.

Look at the body of statements that were added to the macro during the record operation. Examine the body of the `execute0()` method. Find each statement as it is described, and add a comment in the text editor to describe its function.

Code	Explanation
`Simulation simulation_0 = getActiveSimulation();{`	• This statement is essential for STAR-CCM+ macros that interact with and manipulate a simulation. A variable of type `Simulation` is declared, called `simulation_0`. A reference to the location of the active simulation object is obtained using the `getActiveSimulation()` method and stored in the `simulation_0` variable. Add the comment: "`// Retrieve the current simulation`".

`PhysicsContinuum physicsContinuum_0 = ((PhysicsContinuum) simulation_0 .getContinuumManager() .getContinuum("Physics 1") );`
 	
`VelocityProfile velocityProfile_0 = physicsContinuum_0 .getInitialConditions() .get(VelocityProfile.class);`
 	
`velocityProfile_0 .getMethod(ConstantVectorProfileMethod.class) .getQuantity() .setComponents(-10.0, 10.0, 0.0);`
	• These statements set the initial velocity. Setting values using a macro follows the same logical sequence as expanding nodes in the object tree. First you find the physics continuum **Physics 1** in the active simulation, `simulation_0`, and you store a reference to its location in the variable `physicsContinuum_0`. Then you look in `physicsContinuum_0` for the initial velocity condition, and store a reference to its location in `velocityProfile_0`. Finally you set the new velocity condition using the `setComponents()` method.

Each of the entities in STAR-CCM+ is an object, for example boundaries, continua and even the active simulation. To interact with such objects, such as setting a constant value, methods exist which allow you to locate the object in question. To locate an object, follow the hierarchy of the object tree. For example, the initial velocity object is contained in a physics continuum object, which is contained in a simulation object. To interact with such an object, create a variable of the relevant type to store a reference to the location of the object.

Code	Explanation

`Region region_0 = simulation_0 .getRegionManager() .getRegion("trainAndTrack");`
 	
`Boundary boundary_0 =`
	• Here you set the inflow velocity. Find the **trainAndTrack** region in the active simulation, then find the **Inflow** boundary within that region, the **Velocity** condition within that boundary, and finally set the new velocity. To find the region you use a `RegionManager` , and to find a

Code	Explanation

`region_0`
`.getBoundaryManager()`
`.getBoundary("Inflow");`
 	
`VelocityProfile velocityProfile_1 =`
`boundary_0`
`.getValues()`
`.get(VelocityProfile.class);`
 	
`velocityProfile_1`
`.getMethod(ConstantVectorProfileMethod.class)`
`.getQuantity()`
`.setComponents(-5.0, 5.0, 0.0);`
	boundary in the region you use the `BoundaryManagerbelonging` to that region. A second variable of type `VelocityProfile` has been created and has been numbered incrementally to `1`.

`Solution solution_0 =`
`simulation_0`
`.getSolution();`
 	
`solution_0.clearSolution();`
	• These statements retrieve any solution belonging to the current simulation and clear it by calling the `clearSolution()` method.

`ResidualPlot residualPlot_0 =`
`((ResidualPlot) simulation_0`
`.getPlotManager()`
`.getObject("Residuals")`
`);`
	• This set of statements precedes the run simulation command, creating a *Residuals* plot in which to display the simulation progress.

`simulation_0.getSimulationIterator()` `.run();`	• This statement is used to run the simulation.

Code	Explanation

`ForceCoefficientReport forceCoefficientReport_0 =`
`((ForceCoefficientReport)`
`simulation_0.getReportManager()`
`.getReport("Drag Coefficient")`
`);`
 	
`forceCoefficientReport_0.printReport();`
	• Here you find the drag coefficient report, and store a reference to its location in the variable `forceCoefficientReport_0` . The `printReport()` method is then used to run the report within STAR-CCM+. You can use various alternative methods here, for example `getReportMonitorValue()` - this yields the value of the drag coefficient.

Code	Explanation

`Scene scene_0 =`
`  simulation_0`
`  .getSceneManager()`
`  .getScene("Velocity Magnitude");`
 	
`CurrentView currentView_0 =`
`  scene_0`
`  .getCurrentView();`
`_currentView_0.setInput(`
`  new DoubleVector(`
`  new double[]{-2.37, -46.74, -5.45}`
`  ), new DoubleVector(`
`  new double[]{-2.37, -46.74, 152.76}`
`  ), new DoubleVector(`
`  new double[]{0.0, 1.0, 0.0}`
`  ), -1.0, 0`
`);`
 	
`scene_0.printAndWait(`
`  resolvePath("/myTutorials/velMag.png`
`  ), 1, 1024, 768`
`);`
	• This set of statements was generated when the hardcopy of the *Velocity Magnitude* scene was saved. The first statement finds the scene and assigns it to the variable `scene_0`. The lines that follow set the current view of the scene, however they are not necessary in this case and can be removed. The next statement uses the `printAndWait()` method to print the image to file. Notice that the resolution 1024x768 that was entered when the macro was recorded is included in this statement.

`residualPlot_0.encode(`
`  resolvePath("/myTutorials/res.png"`
`  ), "png", 1024, 768`
`);`
	• This set of statements is used to save a copy of the *Residuals* plot.

`Scene scene_1 =`
`  simulation_0`
`  .getSceneManager()`
`  .getScene("Streamlines");`
 	
`scene_1.open(true);`
 	
`scene_1.export3DSceneFileAndWait(`
`  resolvePath("/myTutorials/streamlines.sce"`
	• Here you save a STAR-View+ scene file of the `Streamlines` scene by using the `export3DSceneFileAndWait()` method.

Code	Explanation
`  ), true` `);`	

`simulation_0.saveState(`
`  resolvePath("/myTutorials/`
`trainMacro.sim")`
`);`
	• Finally, the simulation is saved under the name `trainMacro.sim`.

To quickly find what you need when writing the final macro, add comments above each statement or set of statements at this stage.

Save the `macroRecording.java` file.

Creating the Final Macro

You are ready to write the final macro.

Writing the final macro can be approached in a number of different ways, and can be as complex as the task requires. Study the Java macro written to automate the tutorial case in the following steps, taking care to examine how the macro has been constructed. Proceed to the next section for a list of keywords relevant to this chapter, or jump to the following section to see how to finalize the macro structure.

Contents:

Keywords
Finalizing the Structure of the Macro
Class-by-Class Breakdown
Understanding the Main Method in the Macro

Keywords

This topic contains a list of key words that are used in this tutorial when describing elements in the Java code.

Keyword	Definition

Nested Class	A class that has been defined within another class. Nested classes take the following form in Java:
	`class OuterClass {`
	`  class NestedClass {`
	`  }`
	`}`

Keyword	Definition
Member Variable	A variable belonging to a class.
Method	A subroutine belonging to a class.
Instance Variable	A variable contained in an object.
Instance Method	A subroutine contained in an object.

Constructor	A constructor is used to create new objects. By default it allocates memory and initializes instance variables. Every class has at least one constructor; if a constructor definition has not been written in the class the system provides a default constructor. The following example shows a constructor that is used to set the values of the instance variables to the values of some input parameters.
	`public class MyClass {`
	`  private int m_num1;`
	`  private int m_num2;`
	`  public MyClass(int param1, int param2) {`
	`  m_num1 = param1;`
	`  m_num2 = param2;`
	`  }`
	`}`

Instantiate	The process of creating an object from a class is called instantiation, since an object is an instance of a class. You can say that to create an object you must instantiate the relevant class.

Finalizing the Structure of the Macro

It is good practice to finalize the macro structure before writing the final macro. Deciding how best to organize the macro, for example what methods to use, which actions to include in a loop, and the number of classes required. These decisions are based on necessity and your Java programming experience.

It is also useful at this stage to consider whether the macro needs to be modified in the future to include extra commands and processes, for example setting additional boundary conditions. If likely, it is recommended that the macro is divided into different subsections which carry out each task rather than having everything in a single continuous section. When you add extra processes to the macro, you can locate the section easily, and insert the command. The object-oriented nature of the Java language makes it suited to this kind of organization. In this tutorial, nested classes are used to distinguish between steps in the process.

The main processes are divided as follows:

- a class to read the input data
- a class from which to create objects which store input data
- a class to set values in STAR-CCM+ and run the simulation
- a class to carry out post-processing tasks
- an additional class to write any data to an external file, for example the drag coefficient

These classes are given the following names, respectively:

- `DataReader`
- `SimData`
- `SimRunner`
- `PostProcessor`
- `DataWriter`

One key advantage in organizing the macro in this way is that the main method in the macro is easy to understand as the statements follow a logical sequence.

Using an Input Data File

Consider is how the macro reads the wind data and passed to the simulation. You want to be able to alter the data as required, and run as many simulations as necessary. For this reason, the input data is saved in a simple text file, and read into the simulation using the macro.

- Navigate to the `doc/startutorialsdata/automation/data` subdirectory of your STAR-CCM+ installation directory. Copy the `trainInput.txt` file to your working directory.
- Open the `trainInput.txt` file using an appropriate text editor.

The file contains the following data:

- the angle of attack of the cross-wind in degrees, `AngDeg`
- the velocity of the train, `VelTrn`
- the velocity of the cross-wind, `VelWnd`
- the corresponding initial velocities, `InitVelTrn`, and `InitVelWnd`

Each line in the file represents a set of data. Use the macro to read each line and store the data in an object. Store a collection of these objects in an array. When each simulation is run, the data is read from an object and passed efficiently to the simulation, without the need to reopen the input file.

Class-by-Class Breakdown

A class-by-class breakdown is provided for you. The final macro is too complicated to describe how to write it from scratch, so it has been provided for you.

- Navigate to the `doc/startutorialsdata/automation/data` subdirectory of your STAR-CCM+ installation directory. Copy the `trainFlowAngles.java` file to your working directory.
- Open the `trainFlowAngles.java` file using an appropriate text editor.

A detailed description which includes Java coding follows in the following sections. The processes in each nested class are as follows:

SimData

The data passed to the `SimData` object includes the velocities of the train and wind. You must calculate the x and y components of the inflow velocity and initial velocity to set the velocities in the simulation. These calculations are made in the `SimData` class.

- Receive the data from `DataReader`.
- Calculate the velocity components.

DataReader

- Read a line of data from the input file
- Assign each value to the corresponding variable
- Create a `SimData` object for the set of variables
- Store the `SimData` object in a collection of `SimData` objects
- Repeat for each line in the input file.

DataWriter

- Create an output file and write the necessary table headings.
- Add a line to the file including the cross-wind angle and drag coefficient after each simulation has run.

SimRunner

- Retrieve the properties in STAR-CCM+related to the input data.
- Retrieve data from the `SimData` object and set the values of these properties.
- Clear any previous solution.
- Run the simulation.
- Obtain the value of the drag coefficient and store it in the `SimData` object.

PostProcessor

- Retrieve the scenes and plots in STAR-CCM+ to be exported as hardcopies.
- Save hardcopies of various scenes and plots.

The following sections describe each of these classes in detail, highlighting key features in the Java code.

Contents:

Understanding the SimData Nested Class
Understanding the DataReader Nested Class
Understanding the DataWriter Nested Class
Understanding the SimRunner Nested Class
Understanding the PostProcessor Nested Class

Understanding the SimData Nested Class

The SimData class defines objects that hold the data associated with a single simulation. Objects of this type are used throughout the macro.

Description	Code

• Find the `SimData` class declaration in the macro. In the initial lines of the class, declare the member variables . It is good practice for any member variable in a class to be defined as `private` to prevent unauthorized modification of the variable. If necessary, the class provides a "getter" function that returns the value of the member variable. Define the "getter" function as `public` or `protected` (with `public` being the most general).	`public class SimData {`
	
	`private double m_angDeg = 0.0;`
	`private double m_velX = 0.0;`
	`private double m_initVelY = 0.0;`
	`private double m_dragC = 0.0;`

• The constructor definition follows the declaration of the member variables. By writing the constructor definition in this way you have specified that when a `SimData` object is created, a specific set of input parameters is required. Statements in the body of the constructor are executed when a `SimData` object is created, and in this case the constructor calculates velocity components and sets the value of the member variables.	`public SimData(`
	`  double angDeg,`
	`  double velTrn,`
	`  double velWnd,`
	`  double initVelTrn,`
	`  double initVelWnd) {`
	
	`  m_angDeg = angDeg;`
	
	`  double angRad =`
	`  Math.toRadians(angDeg);`
	
	`  m_velX =`
	`  -1 * velWnd *`
	`Math.sin(angRad);`
	
	`}`

• This class exists as a data store, and requires "getter" and "setter" methods so that the data can be accessed and modified. Include a "getter" for each value that to be read. Include a "setter" for each modifiable value to be modified.	`public double getAngle() {`
	`  return m_angDeg;`
	`}`
	
	`public void setDrag(double`
	`dDrag) {`
	`  m_dragC = dDrag;`
	`}`

Proceed to the next section to examine the `DataReader` class.

Understanding the DataReader Nested Class

The `DataReader` class reads data from the input file and creates a `SimData` object for each set of wind data.

Each new `SimData` object is added to an `ArrayList` collection.

Description	Code

• In the first lines of the class a member variable is declared. The `SimData` objects populate the variable.
• No constructor has been defined, so the system supplies a default constructor when a `DataReader` object is created.
	`public class DataReader {`
	
	`  private List<SimData> m_flows =`
	`  new ArrayList<SimData>();`

• Specify a method to carry out the tasks. The path to the input file is passed as an argument to the method. A try-catch construct is used to handle any exceptions when the file cannot be found. If an exception is thrown, the class opens a MessageDialog showing the error string.	`public void readInput(String fileToRead) {`
	
	`  try {`
	
	`  FileReader fr =`
	`  new FileReader(fileToRead);`
	`  BufferedReader br =`
	`  new BufferedReader(fr);`
	`  Scanner sc =`
	`  new Scanner(br);`
	
	`  } catch (Exception e) {`
	
	`JOptionPane.showMessageDialog(`
	`  null, e.toString()`
	`  );`
	`  }`
	`}`

The `JOptionPane` statement is included to aid in debugging; as described in Testing and Debugging.

Description	Code

• Look at the `while` loop which slots into the `try-catch` brackets. The `Scanner` class allows you to read each value in the file as it is encountered, as well as read the file line-by-line. By default the scanner recognizes white spaces, such as tabs, spaces, and carriage returns, as	`while (sc.hasNextLine()) {`
	`  sc.nextLine();`
	`  if (sc.hasNextDouble()) {`
	`  double angDeg =`
	`  sc.nextDouble();`

Description	Code

breaks between values. You can use this functionality to read each value in a line, and pass it to a `SimData` object. Once a `SimData` object has been created and stored in the collection, repeat the process for the next line in the file. A loop repeats the process "while the file has another line". Also use an additional check in the form of an `if` statement. A line is only read out if it contains a number, and not if it.	`double initVelWnd = sc.nextDouble();`
	
	`SimData sd = new SimData( angDeg, , initVelWnd );`
	
	`m_flows.add(sd);`
	`}`
	`}`

• Finally, a getter method is provided to allow access to the collection of `SimData` objects from outside the `DataReaderclass`.	`public List<SimData> getFlowDetails() { return m_flows; }`

Proceed to the next section to examine the `DataWriter` class.

Understanding the DataWriter Nested Class

The `DataWriter` class is used to create an output text file containing the drag coefficient for each cross-wind angle.

Description	Code

• The class contains a member variable of type `String`, which is used to specify the name of the output file.
• The `FileWriter` and `BufferedWriter` used to write data to an external file, belong to the Java language rather than the STAR-CCM+ API. Use the constructor to create an output file and write a line of table headings.
	`public DataWriter(String fileToWrite) {`
	
	`m_outputFile = fileToWrite;`
	
	`try {`
	`FileWriter fw = new FileWriter(m_outputFile);`
	`BufferedWriter bw = new BufferedWriter(fw);`
	`bw.write("Angle (deg), DragCoefficient");`
	`bw.newLine();`
	`bw.close();`
	`} catch (Exception e) {`
	`}`
	`}`

Description	Code

• The `writeDataLine()` method opens the output file, adds a line that includes the cross-wind angle and the drag coefficient, then closes the file. The `newLine()` method is included to move the cursor to the start of the next line in the output file. As the output data is contained in the `SimData` object, the relevant object is passed to the method in the argument list.
• When the `FileWriter` object is created in the method, you can see that one of the arguments is `true`. This results in new data being appended to the end of the file.
	`public void writeDataLine(SimData sD) {`
	
	`  try {`
	`  FileWriter fw =`
	`  new FileWriter(m_outputFile, true);`
	`  BufferedWriter bw =`
	`  new BufferedWriter(fw);`
	`  bw.write(sD.getAngle() + ", " + sD.getDrag());`
	`  bw.newLine();`
	`  bw.close();`
	`  } catch (Exception e) {`
	`  }`
	`}`

Proceed to the next section to examine the `SimRunner` class.

Understanding the SimRunner Nested Class

In the `SimRunner` class, you implement the statements that were obtained when the macro was recorded.

Description	Code
• You declare the properties that are set in the simulation as member variables of the `SimRunner` class. Include only variables that are set using the input data, such as inlet velocity, and variables that are extracted, such as drag coefficient. The type of each of the member variables can be found by referring to the `macroRecording.java` file, and to the section Understanding the Recorded Macro .	`private VelocityProfile m_inflowVel = null;`

• Locating an object in STAR-CCM+ follows the same path as expanding nodes in the object tree. Therefore, to perform an operation such as setting the inlet velocity, locate each of the upper objects first. Here this action is performed in the constructor, as the location of the objects do not change, and you want to find the objects for every `SimRunner` object that is created.	`public SimRunner(Simulation theSim) {`
	
	`  m_sim = theSim;`
	
	`  PhysicsContinuum physics =`
	`  ((PhysicsContinuum) m_sim`
	`  .getContinuumManager()`
	`  .getContinuum("Physics 1"));`

Description	Code

• The statements for locating each object can be copied from the recorded macro, taking care the change the variable names as necessary.	`m_initVel = ((VelocityProfile) physics .getInitialConditions() .get(VelocityProfile.class));`
	
	`}`

• In the next method, `runCase()`, implement the statements that set the necessary properties in the simulation. First, you extract the values from the `SimData` object; then you use this data to set the values of each property. Notice that you use the getter methods that were defined in the `SimData` nested class to extract the data.	`public void runCase(SimData sD, int iterations) {`
	
	`double initX = sD.getInitVelX();`
	
	`double velY = sD.getVelY();`

Description	Code

• The variables are inserted into the `x` and `y` positions in the `setComponents()` method for the inflow and initial velocities; the variables have been underlined.	`((ConstantVectorProfileMethod) m_initVel .getMethod()) .getQuantity() .setComponents(initX, initY, 0.0);`
	
	`((ConstantVectorProfileMethod) m_inflowVel .getMethod()) .getQuantity() .setComponents(velX, velY, 0.0);`

• Each of the statements in the method is carried out in sequence. Once the new values have been set, you want to make sure that any previous solution has been cleared.	`m_sim.clearSolution();`
• You can now run the simulation. The number of iterations is specified by passing an integer to the `run()` method, the value of which set when the `runCase()` method is called.	`m_sim.getSimulationIterator() .run(iterations);`

Description	Code

• Finally, you obtain the drag coefficient, and store it in the `SimData` object by using the setter method defined earlier.	`double cdValue = m_forceReport.getReportMonitorValue();`
	`sD.setDrag(cdValue);`
	`}`

Proceed to the next section to examine the `PostProcessor` class.

Understanding the PostProcessor Nested Class

The `PostProcessor` class is used to export hardcopies of the *Velocity Magnitude* scene and the *Residuals* plot. You can also export the *Streamlines* scene as a STAR-View+ file.

Description	Code
• In a similar fashion to the previous class, you set member variables for each scene or plot that you want to export.	`public class PostProcessor {` `private Scene m_velMag = null;`

• Furthermore, you find the relevant objects in the simulation by placing the necessary statements in the constructor.	`public PostProcessor(Simulation theSim) {`
	`m_sim = theSim;`
	`m_velMag = ((Scene) m_sim .getSceneManager() .getScene("Velocity Magnitude"));`
	`}`

• Following are methods which are used to perform the action of saving the relevant scene or plot. As before, the statements for carrying out these operations can be copied from the recorded macro. You pass the path of the output file to the method as an argument, `String sceneToSave`.	`public void saveVelMagScene(String sceneToSave) {`
	`m_velMag.printAndWait( sceneToSave, 1, 1024, 768 );`
	`}`

Description	Code

• When a hardcopy is saved, it is automatically opened in the *Graphics* window in the STAR-CCM+ GUI. In the Streamlines scene, make sure that the scene is closed before the next simulation is run, otherwise the simulation run time is dramatically increased. Use the `Scene.close()` method.	`public void saveStreamlinesScene(String sceneToSave) {`
	
	`m_streamlines.export3DSceneFileAndWait(`
	`sceneToSave, true`
	`);`
	
	`m_streamlines.close(true);`
	
	`}`

Understanding the Main Method in the Macro

The main method in the macro is the `execute()` method, and is executed when the macro is run within STAR-CCM+.

Here you instantiate objects for each of the nested classes and use the methods in each, to carry out the required tasks.

Description	Code

• A try-catch operation surrounds the entire block of statements in the `execute()` method. The try-catch is useful in case any other errors are thrown when you run the macro.	`public void execute() {`
	
	`try {`
	
	`} catch (Exception e) {`
	
	`jOptionPane.showMessageDialog(`
	`null, e.toString()`
	`);`
	`}`
	`}`

Within the try-catch operation, you can divide the statements into two main sections. The first section contains statements that are carried out once, for example retrieving the active simulation. The second section contains a loop in which you place statements that are repeated for each set of wind data.

Description	Code

• In the first section you retrieve the active simulation and instantiate the `DataReader`, `DataWriter`, `SimRunner`, and `PostProcessor`. In addition, use the `DataReader` object to call the `readInput()` method that reads the input file and create a collection of `SimData` objects. The `getFlowDetails()` method is called to obtain a reference to the `SimData` collection and is supplied to the loop.	`Simulation theSim = getActiveSimulation();`
	
	`DataReader reader = new DataReader();`
	`reader.readInput(folder + "/trainInput.txt");`
	
	`List<SimData> listCases = reader.getFlowDetails();`
	
	`PostProcessor postP = new PostProcessor(theSim);`

• Next, use a loop to run the simulation for each set of wind data. The syntax used is that of a for-each loop, and reads: for each `SimData` object in the collection, do the following. This functionality came about in Java 5, and offers a convenient way to iterate over arrays and other collections. You use the reference to the `SimData` collection, `listCases` , to define the for-each loop.	`for (SimData sD : listCases) {`
	
	`}`

Description	Code
• Within the loop, the first process is to use the `SimRunner` object to carry out the tasks such as setting various conditions, clearing the previous solution, or running the case. Recall that these processes were defined in the method `runCase()`. You call this method, and pass it the current `SimData` object `sD` and an integer to define the number of iterations to carry out when each simulation is run, in this case 5.	`runner.runCase(sD, 5);`

• Next, following the logical process of running a simulation, you extract the drag coefficient and save hardcopies of scenes and a plot. As before, call the methods that were defined in the relevant nested classes. The cross-wind angle was retrieved from the current `SimData` object to save each scene under a suitable name.	`writer.writeDataLine(sD);`
	
	`postP.saveVelMagScene(`
	`folder + "/velMag" + sD.getAngle() + ".png"`
	`);`

Description	Code

• Finally, use another statement copied from the recorded macro to save the simulation after it has been run.	`theSim.saveState(`
	`  folder + "/train" +`
	`sD.getAngle() + ".sim"`
	`);`

Testing and Debugging

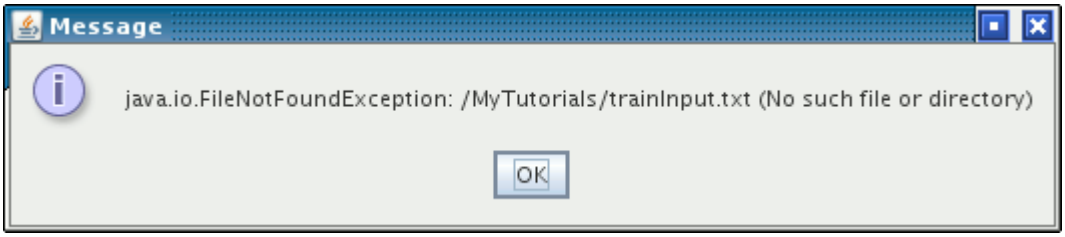
Test and debug your macro.

There are some useful tools that can help to debug the macro and find the source of the problems.

The first is used with the try-catch error handler, and has the following syntax:

```
try {

} catch (Exception e) {
  JOptionPane.showMessageDialog(
    null, e.toString()
  );
}
```

The purpose of this code is to create a window that displays the error message. The macro is paused until you accept the error by clicking the **OK** button. Afterwards, the macro attempts to continue running and if it cannot continue the process stops.



Another useful tool that can be used when debugging is the ability to print a line of text to the *Output* window in the STAR-CCM+GUI. Implement code that follows the following syntax:

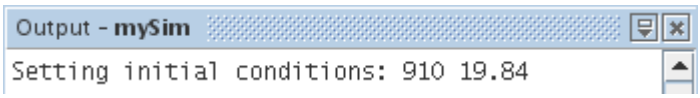
```
Simulation mySim =
  getActiveSimulation();

mySim.println("Output Text");
```

The output line is passed to the `println()` method as an `Object`, and so can include various parameters. The following example has inputs of type `String`, `int`, and `double`:

```
String myStr = "initial conditions";
double myDbl = 19.84;
```

```
int myInt = 910;

Simulation mySim =
  getActiveSimulation;

mySim.println(
  "Setting " + myStr + ": " + myInt + " " + myDbl
);
```

This code results in the following display in the *Output* window.



You can use this tool to print a line of text to the *Output* window when each key step in the macro has been performed. In this way, if the macro comes to a halt prematurely then you know that the code before the last printed message is not responsible for the error.

Keep the macro run-time to a minimum during development, as it is likely that several attempts are required to get everything correct. The number of iterations can be increased to their final amount at the end.

Running the Macro

Modify the macro to use your working directory to find the input file:

1. With the `trainFlowAngles.java` file open in an appropriate text editor, locate the following line. Change the section of code that is shown in bold.

   ```
   public static final String folder =
     "/myTutorials";
   ```

2. In place of `/myTutorials`, enter the absolute path to the location of the `trainInput.txt` file that was saved earlier in the tutorial.
3. When the new path has been set, save the file to your working directory. Retain the file name `trainFlowAngles.java`.
4. Close the file

You can now run the macro:

5. Return to the STAR-CCM+ GUI and make sure that the `trainMacro` simulation is active.
6. In the *Macro* toolbar, click **(Play Macro)** and select the `trainFlowAngles.java` file.

   If the macro does not run, make sure that:

   - The case has been set up correctly to receive the input data, for example the inlet velocity specification is set to components
   - The `trainInput.txt` file is located in the directory specified in the macro
   - Use the tools described in the Testing and Debugging section to identify the source of the problem.

7. Proceed to the next section if you wish to run the simulation for sufficient iterations to achieve convergence, or jump to the Summary to review the tutorial.

Contents:

Running the Simulation to Convergence

Running the Simulation to Convergence

The following steps demonstrate how to alter the macro to run the set of simulations until convergence has been reached.

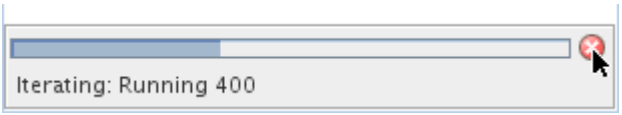
1. Open `trainFlowAngles.java` file using an appropriate text editor.
2. Locate the line in the macro where you call the `runCase()` method from the `SimRunner` object.

```
public void execute() {

  try {

    for (/**/) {

      runner.runCase(sD, 5);

    }
    } catch (/**/) {
    }
}
```

3. Change the number of iterations to `400`.

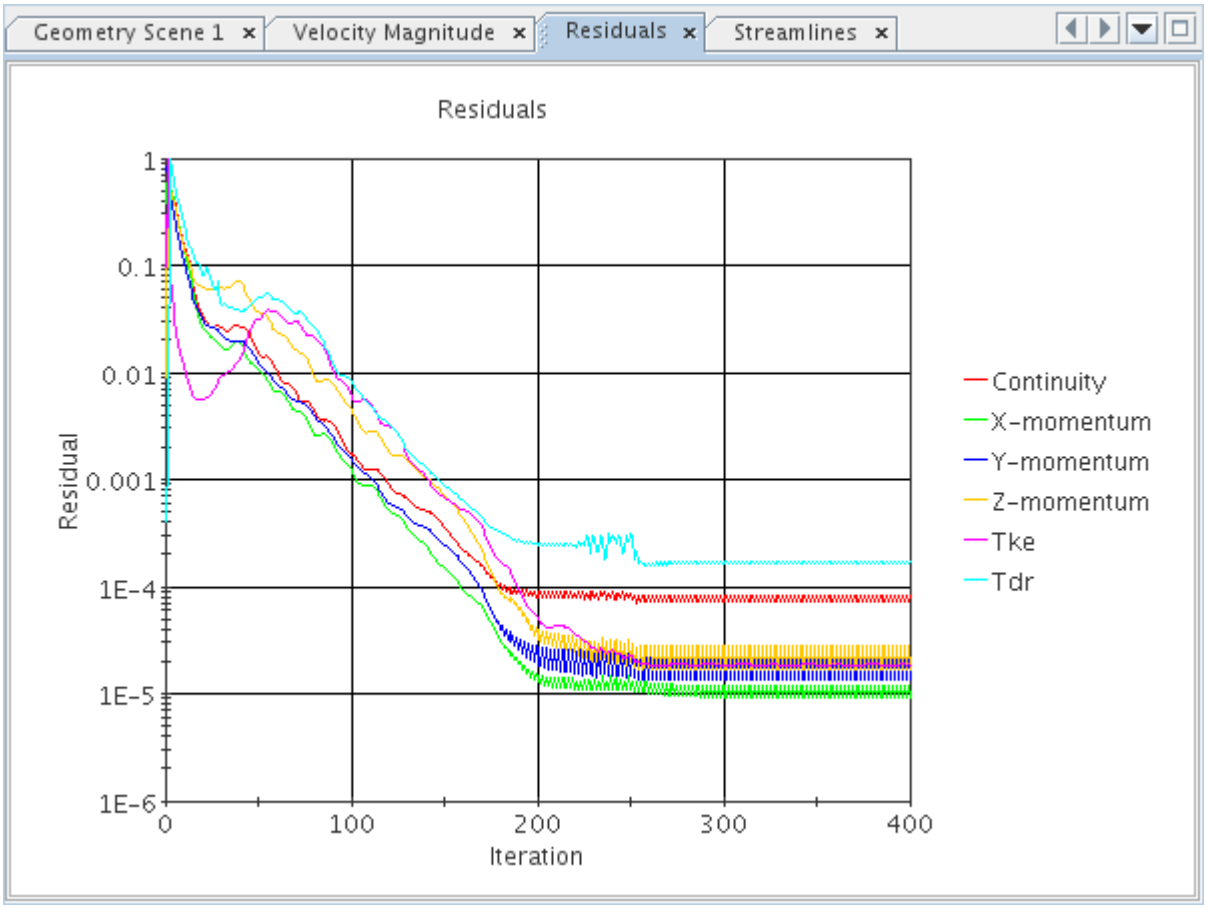
```
runner.runCase(sD, 400);
```

4. Save and close the file.
5. Make sure that the simulation `trainMacro.sim` is open.
6. Click the **(Play Macro)** button in the *Macro* toolbar and select the `trainFlowAngles.java` file.
7. If, at any time, the macro must be interrupted click the **(Abort)** button next to the status bar.

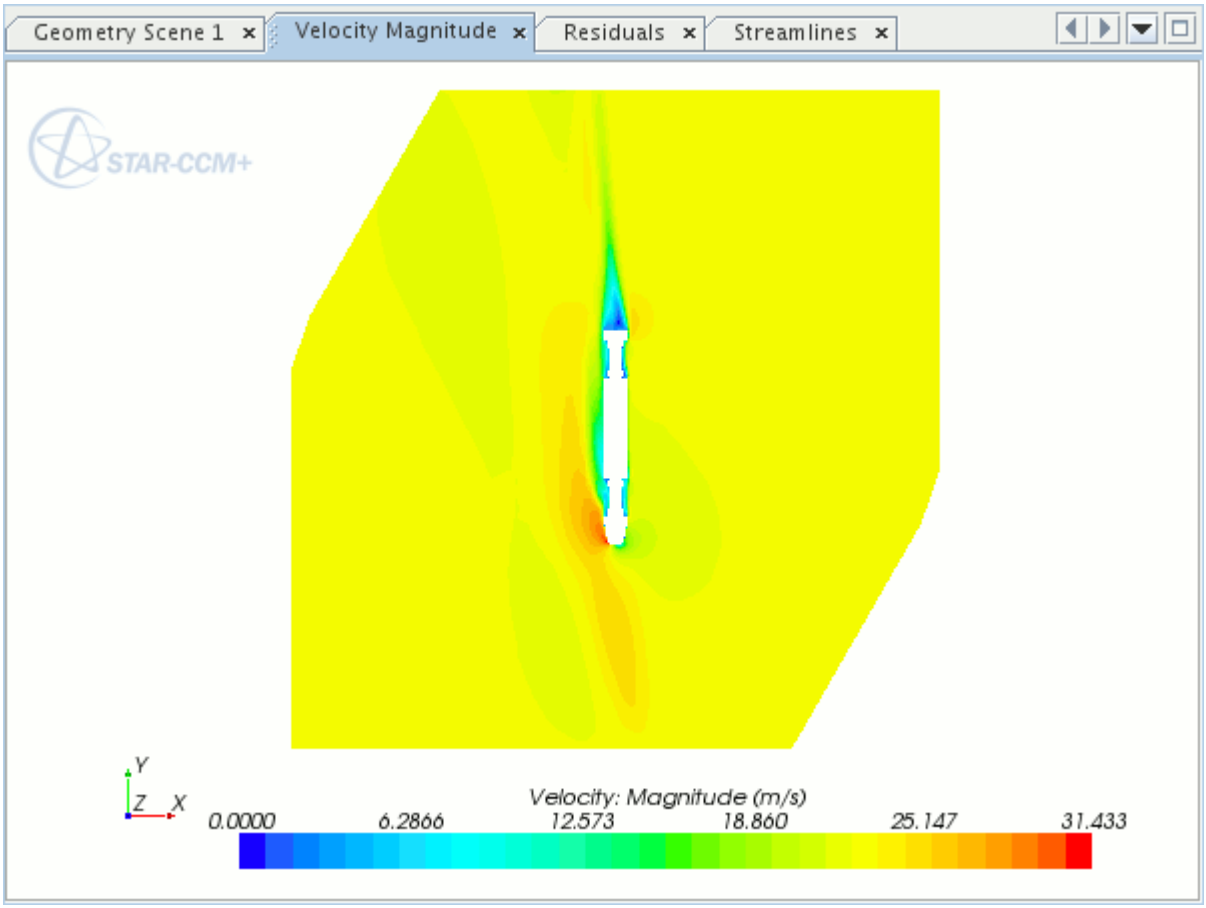


When the macro has finished running you get a set of drag coefficient data, as well as a collection of scenes and plots. The following are examples of the residual plot, and streamlines scene for the train with a cross-wind angle of 90 degrees, after 400 iterations.

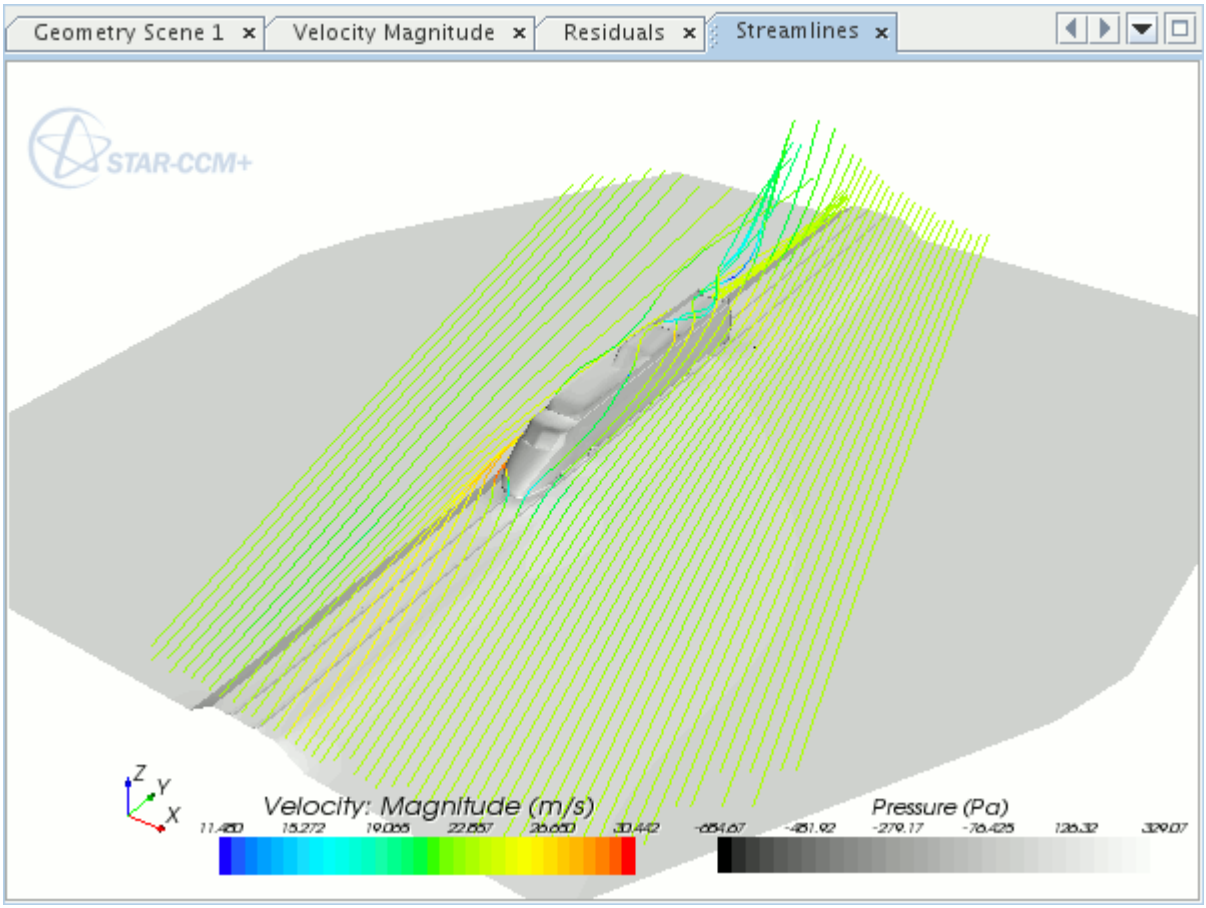
Residual Plot:



Velocity Magnitude Scene:



Streamlines Scene:



Summary

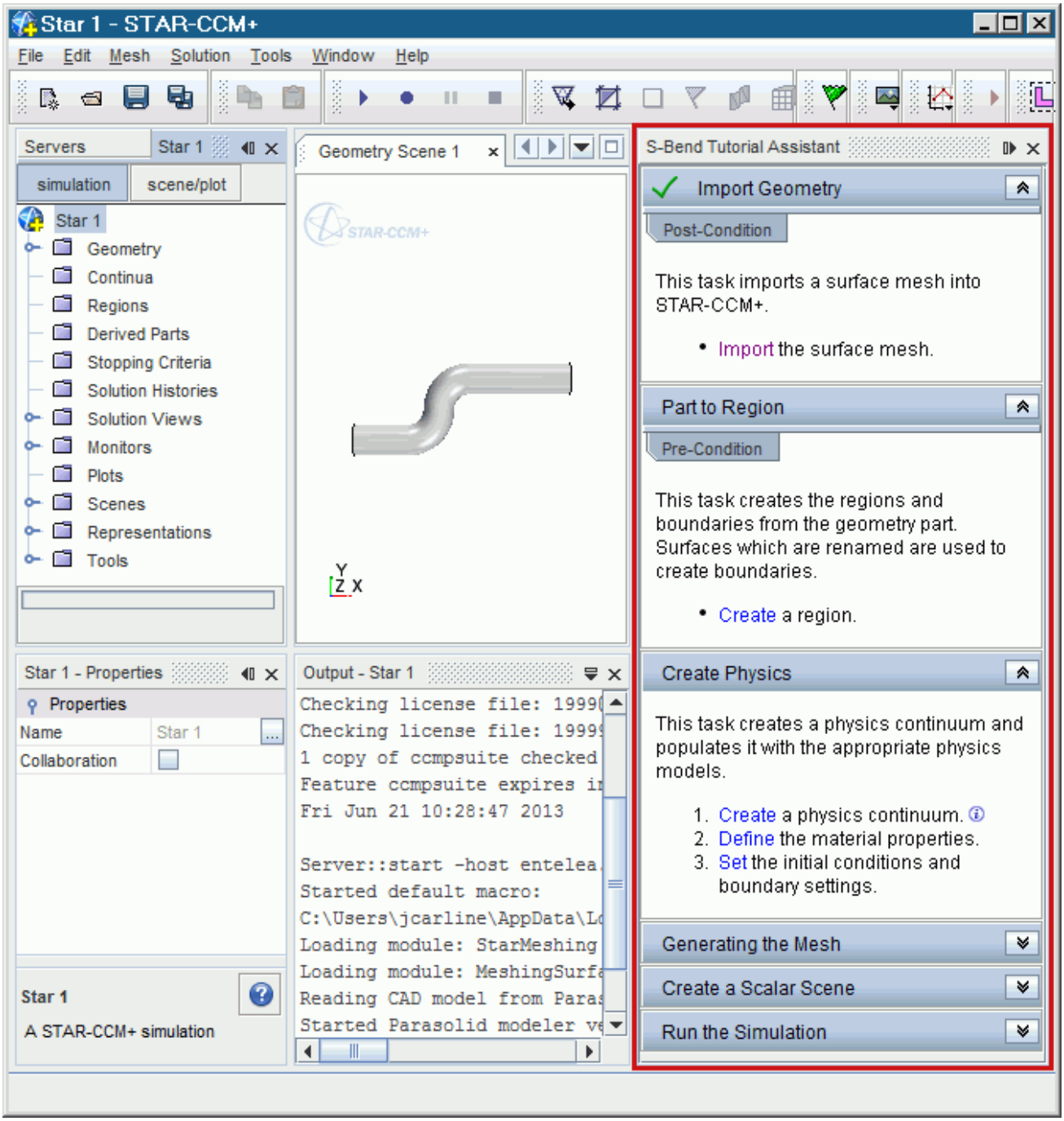
This tutorial has introduced the following STAR-CCM+ features:

- Setting up wake refinement
- Defining the motion of the track
- Creating a drag coefficient report
- Creating a velocity magnitude scalar scene
- Creating a scalar scene with streamlines
- Recording a macro;
- Understanding the recorded macro
- Designing a macro that uses nested classes
- Reading data into STAR-CCM+from a text file
- Using a macro to run consecutive simulations with different boundary conditions
- Writing data to an output text file
- Using a macro to export hardcopies of scenes and plots
- Testing and debugging the macro
- Running the macro

Simulation Assistant: Internal Flow Assistant

The Simulation Assistant provides you with an API framework that allows you to create workflows.

When loaded into STAR-CCM+, workflows appear on the right-hand side of the user interface, as shown highlighted in red in the following screenshot. An assistant project consists of Java classes and XHTML text.



When developing an assistant, it is recommended you use an IDE (Integrated Development Environment). This tutorial uses the NetBeans 7.3.1 IDE. You can get a free copy of NetBeans SE from the official website.

This tutorial presents the steps to create a Simulation Assistant for a simple workflow. The specific workflow that is created is for modeling an internal pipe flow. The existing S-Bend tutorial is an example of such an internal flow.

When creating an assistant, it is useful to break down the workflow into logical tasks, and to group related actions together. For example, actions that are related to creating and setting up the physics of a simulation can be grouped together in a single task.

The assistant in this tutorial has six tasks:

- Importing the geometry
- Creating regions from the parts
- Setting up the physics
- Generating the mesh
- Visualizing the solution
- Running the simulation

In general, there are two methods that you can use to create the initial Java code for each task:

- Record a Java macro while working through an example simulation in the STAR-CCM+ GUI. Copy and paste the relevant code from the recorded macro to the Simulation Assistant project.
- Write your own code for the assistant project.

In this tutorial you start by creating a simple assistant that has basic functionality. When the basic assistant is complete, you add more advanced features to it. These are:

- Conditions—add an extra layer of control and guidance to the assistant, and act as safeguards when you move between tasks
- Information Tags—allow you to add more information about a task or a particular step without cluttering the assistant panel. By default, additional information is hidden and only appears when required.
- Object Tree Focus—allows you to manipulate the STAR-CCM+ object tree. When you click on a step, you can control whether the corresponding object in the STAR-CCM+ object tree is highlighted. If the object is hidden inside manager nodes, these nodes are automatically expanded. To further focus on the action, you can choose to collapse all other expanded nodes in the object tree.

Contents:

Prerequisites
Creating the NetBeans Project
Creating the Assistant Class
Creating the First Task: Importing the Geometry
Building the Project
Loading the Assistant into STAR-CCM+
Creating the Second Task: Converting the Parts to Regions
Creating the Third Task: Continuum Physics
Creating the Remaining Tasks
Advanced Simulation Assistant Features

Expanding the Object Tree to Reveal Specific Objects
Adding an Information Tag to the Assistant
Summary

Prerequisites

The instructions in the 3D-CAD tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

You require an installation of NetBeans 7.3.1 or later to complete this tutorial. Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes and plots	Introduction to STAR-CCM+
Recording and editing Java macros	Simple Java Macros: Post Processing

Creating the NetBeans Project

Create a NetBeans project.

The first step in creating a Simulation Assistant is to create a project in NetBeans. Each Simulation Assistant typically consists of an assistant class and numerous task classes. The assistant class provides the overall order for the tasks in your assistant, while the task classes contain the steps that perform actions in STAR-CCM+.

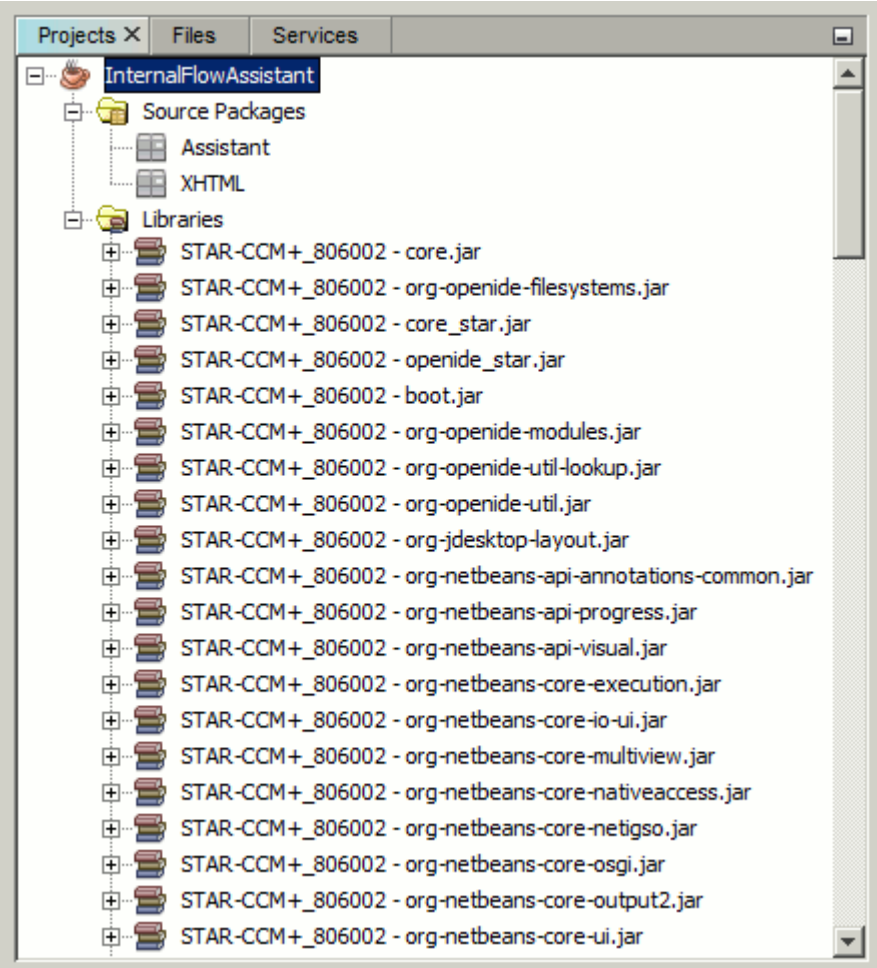
Note: If you require assistance in setting up NetBeans or your project, refer to *Creating a Simulation Assistant Project in NetBeans* in the STAR-CCM+ user guide.

Create and prepare a new project in NetBeans:

1. Launch NetBeans and create a new Java application project called `InternalFlowAssistant`.
2. While creating the project, do not create the main class.
3. Add two Java packages to the project's source package and name them:
   - `Assistant`
   - `XHTML`
4. Load the STAR-CCM+ libraries into your NetBeans project. You must import the `.jar` files that are located in the following directories:
   - `[STAR-CCM+_Installation]/star/lib/java/platform/core`
   - `[STAR-CCM+_Installation]/star/lib/java/platform/core/locale`
   - `[STAR-CCM+_Installation]/star/lib/java/platform/lib`
   - `[STAR-CCM+_Installation]/star/lib/java/platform/modules`
   - `[STAR-CCM+_Installation]/star/lib/java/platform/modules/ext`
   - `[STAR-CCM+_Installation]/star/lib/java/platform/modules/locale`

Note: If you are creating the project on a Windows platform, it is recommended that you copy and paste the STAR-CCM+ libraries to the root directory of your system, for example: `C:\STAR-CCM+ Libraries`. When building code, Windows has a limitation of how large the path to the libraries can be. In the default STAR-CCM+ installation directory, the STAR-CCM+ libraries would contain paths that are too long. When building the project, this would cause problems.

The project is displayed below.



Creating the Assistant Class

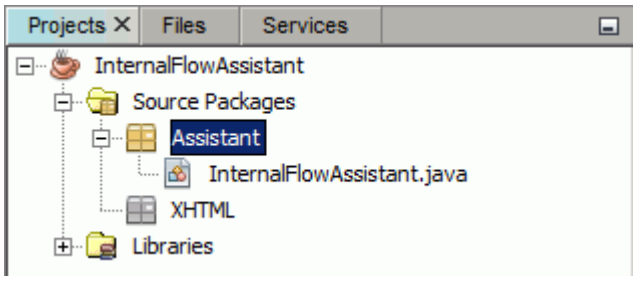
Create the assistant class.

The assistant class acts as the ‘table of contents’ for all the tasks in your assistant. It is good practise to create this class first, as it will allow you to test your tasks when you complete them. Initially, you create the ‘template’ of the class, and add the name of the task classes to this template as you complete them, in the order that you want them to appear.

Create the assistant class:

1. Right-click the Assistant package and select **New** > **Java Class**.

2. Add a new class to the **Assistant** package and name it `InternalFlowAssistant`.



3. Copy and paste the following code into the `InternalFlowAssitant` class, overwriting everything:

```
package Assistant;

import java.util.ArrayList;
import java.util.List;
import star.assistant.SimulationAssistant;
import star.assistant.Task;
import star.assistant.annotation.StarAssistant;

// Specifies the name of the Simulation Assistant in the GUI.
@StarAssistant(display = "Internal Flow Assistant")
public class InternalFlowAssistant extends SimulationAssistant {

    public InternalFlowAssistant() {
        // Creates a new array list for the list of tasks.
        List<Task> tasks = new ArrayList<Task>();
        // Contains the list of tasks.
        tasks.add(new Task1Name());
        setOutline(tasks);
    }
}
```

where:

- `tasks.add(new Task1Name());` is the placeholder for the first task.
- `Task1Name` is the name of the task class. When you complete the first task, replace it with the name of the class.

When you copy and paste from a web browser into NetBeans, the text loses its formatting. NetBeans can reformat this for you:

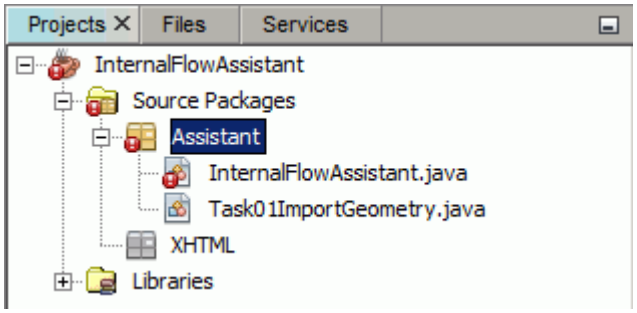
4. Select **Source** > **Format**.

Creating the First Task: Importing the Geometry

The first task imports the geometry into STAR-CCM+. This task is designed to be interactive, meaning that when you click the link in the task, an import dialog appears.

To begin:

1. Add a new class to the Assistant package and name it `Task01ImportGeometry`.



2. Copy and paste the following code into the `Task01ImportGeometry` class, overwriting everything:

```
package Assistant;

import java.io.File;
import java.util.Collection;

import javax.swing.JFileChooser;

import star.assistant.Task;
import star.assistant.ui.FunctionTaskController;
import star.common.GeometryPart;
import star.common.Simulation;
import star.common.SimulationPartManager;
import star.meshing.PartImportManager;

public class Task01ImportGeometry extends Task {

    public Task01ImportGeometry() {
    }

    public class ImportGeometryTaskController extends
FunctionTaskController {

        public void importSurfaceMeshDialog() {

            //open a file chooser
            JFileChooser fileChooser = new JFileChooser();
            if (fileChooser.showOpenDialog(null) !=
JFileChooser.APPROVE_OPTION) {
                return;
            }
            File cadFile = fileChooser.getSelectedFile();

            //import the part
```

```
            Simulation simulation_0 = getSimulation();
            PartImportManager partImportManager_0 =
                    simulation_0.get(PartImportManager.class);

            //use default import options
            partImportManager_0.importCadPart(cadFile.getPath(),
"SharpEdges", 30.0, 2, true, 1.0E-5, true, false);

            //add the new part to the lookup
            Collection<GeometryPart> new_parts =
simulation_0.get(SimulationPartManager.class).getParts();
            if (!new_parts.isEmpty()) {
                addToTaskLookup(new_parts.iterator().next());
            }

            //create a Geometry Scene

simulation_0.getSceneManager().createGeometryScene("Geometry
Scene", "Outline", "Geometry", 1);
            Scene scene_0 =

simulation_0.getSceneManager().getScene("Geometry Scene 1");

            scene_0.initializeAndWait();
            PartDisplayer partDisplayer_1 =
                    ((PartDisplayer) scene_0.getCreatorDisplayer());
            partDisplayer_1.initialize();
            PartDisplayer partDisplayer_0 =
                    ((PartDisplayer)
scene_0.getDisplayerManager().getDisplayer("Outline 1"));
            partDisplayer_0.initialize();
            PartDisplayer partDisplayer_2 =
                    ((PartDisplayer)
scene_0.getDisplayerManager().getDisplayer("Geometry 1"));
            partDisplayer_2.initialize();
            scene_0.open(true);
            CurrentView currentView_0 =
                    scene_0.getCurrentView();
            currentView_0.setInput(new DoubleVector(new double[]
{0.07000000029802322, 0.02000000048428774, 0.0}),
                                                        new
DoubleVector(new double[]{-0.14607243684130425,
0.11741414039370678, 0.17860456694301519}),
                                                        new
DoubleVector(new double[]{0.20083797287207372, 0.9415226723669483,
-0.2705534440210218}),

0.07747426272518879, 0);
```

```
            }
        }
    }
```

In this task, you use the standard Java function, `JFileChooser`, to launch an open dialog. The person using the assistant can then import the desired geometry file into STAR-CCM+.

Note: When you write your own code, NetBeans automatically suggests which classes to import. Lines that need your attention are flagged with a red exclamation mark. Press "`Alt`" + "`Enter`" on these lines to display the suggestions. For now, you can ignore any warnings.

Contents:

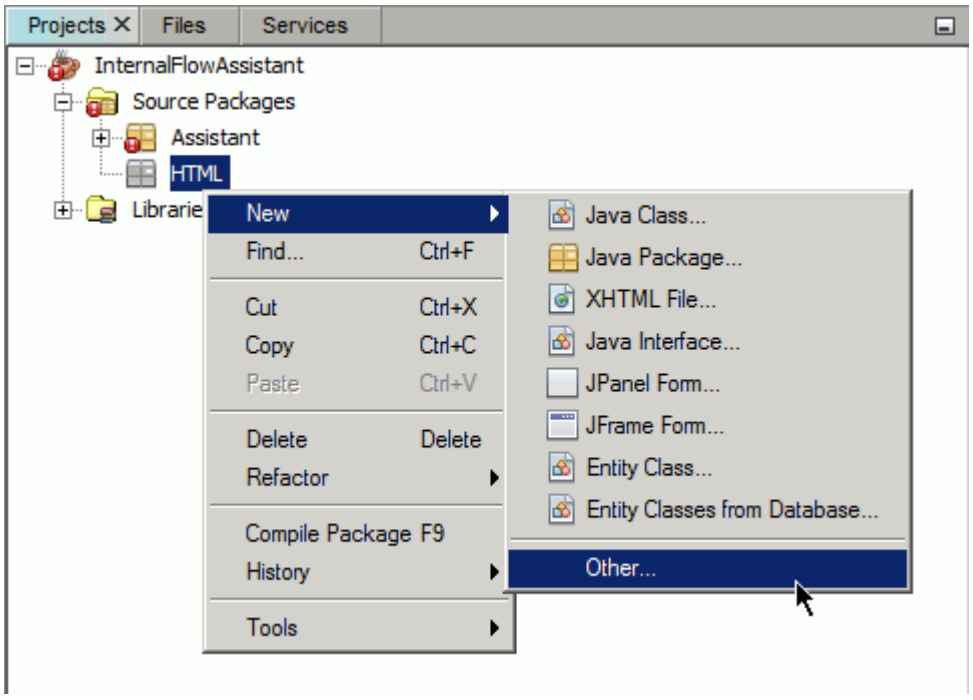
Adding a Description to the Task Panel
Linking the XHTML Content to the Task Class
Reviewing the Code for Task 1

Adding a Description to the Task Panel

You can add descriptions to each task panel. Furthermore, you can link a particular word or phrase in a step to a staraction. The word effectively becomes a link, and when clicked, performs a pre-defined action. It is recommended that you store all text in an external `.xhtml` file. It is also good practise to store all `.xhtml` files in a separate package to the source package. For this tutorial, you already created the `XHTML` package in an earlier section.

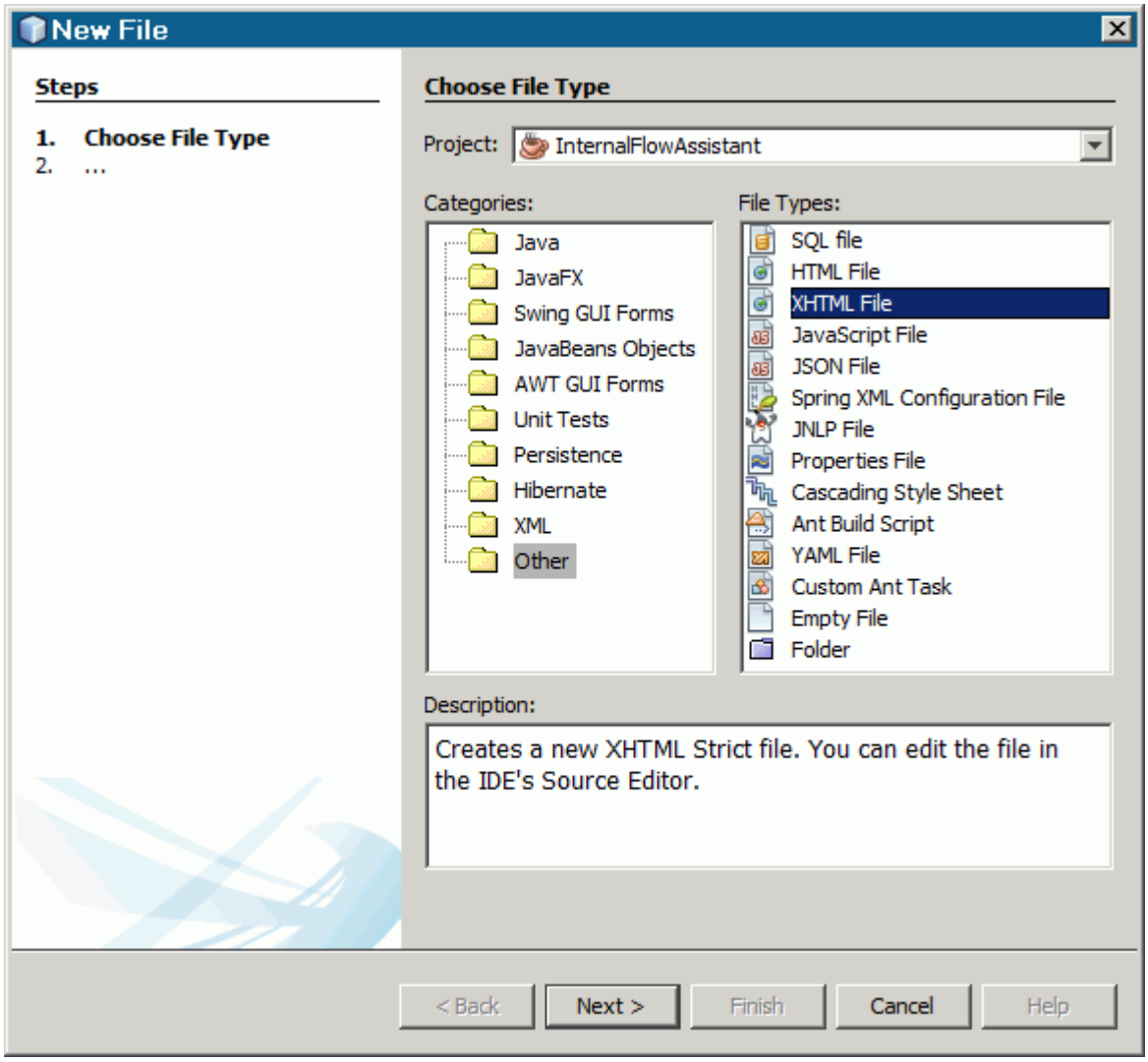
Add an XHTML file to your project:

1. Right-click on the **XHTML** package and select **New** > **Other...**.



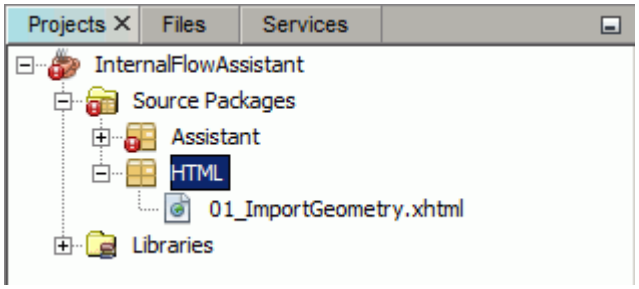
The *New File* dialog appears.

2. From the **Categories** list, select **Other**.
3. From the **File Types** list, select **XHTML**.



4. Click **Next** and enter `01_ImportGeometry` as the name.
5. Click **Finish**.

   The `.xhtml` file is added to the XHTML package.



6. Copy and paste the following code into the `01_ImportGeometry.xhtml` file:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://
www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmls="http://www.w3.org/1999/xhtml">
    <head>
        <meta http-equiv="CONTENT-TYPE" content="text/html;
charset=utf-8" />
    </head>
    <body>
        <p>
            <!--This is the description of the task that appears in
the task panel.-->
            This task imports a geometry part into STAR-CCM+.
            <ul>
                <!--This is a list item that appears in the task.
The word "Import" is linked to a staraction.-->
                <li><a
href="staraction:importSurfaceMeshDialog">Import</a> the surface
mesh.</li>
            </ul>
        </p>
    </body>
</html>
```

where:

- `staraction:importSurfaceMeshDialog` calls the `importSurfaceMeshDialog` function from the `Task01ImportGeometry` class.

7. Reformat the code in NetBeans.

Linking the XHTML Content to the Task Class

Link the XHTML content to the task class.

Now that the XHTML file is completed, link this file to the `Task01ImportGeometry` class:

1. Return to `Task01ImportGeometry` and locate the line beginning with `public class Task01ImportGeometry extends Task {`.

2. Insert the following code above that line:

```
@StarAssistantTask(display = "Import Geometry",
    contentPath = "XHTML/01_ImportGeometry.xhtml",
    controller =
Task01ImportGeometry.ImportGeometryTaskController.class)
```

Reviewing the Code for Task 1

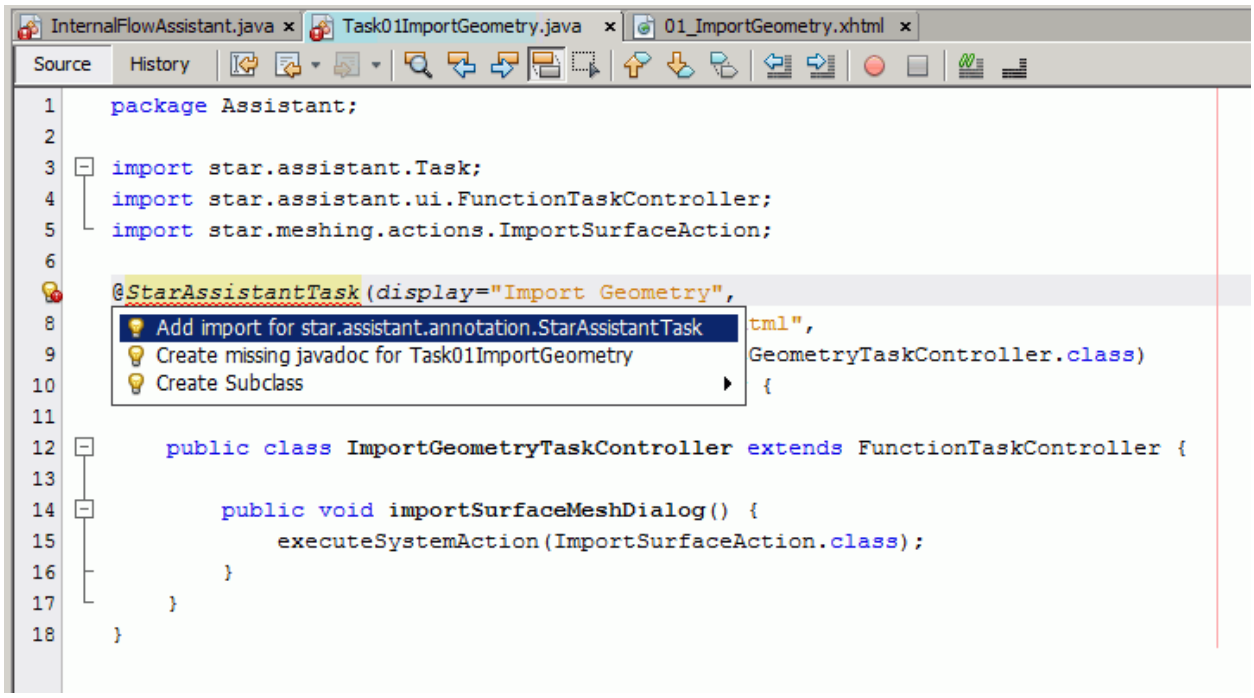
Review the code for the first task.

After linking the XHMTL file to the task class, NetBeans adds a warning on the line beginning with `@StarAssistantTask`. This error appears due to a missing import for the assistant task.

To resolve the issue, import the appropriate class:

1. Click on the line with the warning and press <`Alt`> + <`Enter`>.

   A pop-up appears with suggestions to fixing the problem.
2. Select **Add import for star.assistant.annotation.StarAssistantTask**.



   The following line is added to the code and the error flag is removed:

```
import star.assistant.annotation.StarAssistantTask;
```

3. Select **Source** > **Fix Imports** to automatically add all the missing imports to the class.
4. Select **Source** > **Format** to reformat the code in NetBeans.

The final code for the first task is displayed below:

```
package Assistant;

import java.io.File;
import java.util.Collection;

import javax.swing.JFileChooser;

import star.assistant.Task;
import star.assistant.annotation.StarAssistantTask;
import star.assistant.ui.FunctionTaskController;
import star.base.neo.DoubleVector;
import star.common.GeometryPart;
import star.common.Simulation;
import star.common.SimulationPartManager;
import star.meshing.PartImportManager;
import star.vis.CurrentView;
import star.vis.PartDisplayer;
import star.vis.Scene;

@StarAssistantTask(display = "Import Geometry",
    contentPath = "XHTML/01_ImportGeometry.xhtml",
    controller =
Task01ImportGeometry.ImportGeometryTaskController.class)
public class Task01ImportGeometry extends Task {

    public Task01ImportGeometry() {
    }

    public class ImportGeometryTaskController extends
FunctionTaskController {

        public void importSurfaceMeshDialog() {

            //open a file chooser
            JFileChooser fileChooser = new JFileChooser();
            if (fileChooser.showOpenDialog(null) !=
JFileChooser.APPROVE_OPTION) {
                return;
            }
            File cadFile = fileChooser.getSelectedFile();

            //import the part
            Simulation simulation_0 = getSimulation();
            PartImportManager partImportManager_0 =
                simulation_0.get(PartImportManager.class);
```

```
            //use default import options
            partImportManager_0.importCadPart(cadFile.getPath(),
"SharpEdges", 30.0, 2, true, 1.0E-5, true, false);
            //add the new part to the lookup
            Collection<GeometryPart> new_parts =
simulation_0.get(SimulationPartManager.class).getParts();
            if (!new_parts.isEmpty()) {
                addToTaskLookup(new_parts.iterator().next());
            }

            //create a Geometry Scene

simulation_0.getSceneManager().createGeometryScene("Geometry Scene",
"Outline", "Geometry", 1);
            Scene scene_0 =
                simulation_0.getSceneManager().getScene("Geometry
Scene 1");
            scene_0.initializeAndWait();
            PartDisplayer partDisplayer_1 =
                ((PartDisplayer) scene_0.getCreatorDisplayer());
            partDisplayer_1.initialize();
            PartDisplayer partDisplayer_0 =
                ((PartDisplayer)
scene_0.getDisplayerManager().getDisplayer("Outline 1"));
            partDisplayer_0.initialize();
            PartDisplayer partDisplayer_2 =
                ((PartDisplayer)
scene_0.getDisplayerManager().getDisplayer("Geometry 1"));
            partDisplayer_2.initialize();
            scene_0.open(true);
            CurrentView currentView_0 =
                scene_0.getCurrentView();
            currentView_0.setInput(new DoubleVector(new double[]
{0.07000000029802322, 0.02000000048428774, 0.0}),
                new DoubleVector(new double[]{-0.14607243684130425,
0.11741414039370678, 0.17860456694301519}),
                new DoubleVector(new double[]{0.20083797287207372,
0.9415226723669483, -0.2705534440210218}),
                0.07747426272518879, 0);
        }
    }
}
```

Building the Project

Build the NetBeans project.

At this stage, you must update the assistant class with the name of the class for Task 1:

1. Return to the assistant class, `InternalFlowAssistant`, and locate the line:

```
tasks.add(new Task1Name());
```

2. In that line, replace `Task1Name()` with the name of the Task 1 class:

```
Task01ImportGeometry()
```

3. Save the project.

You can now build the Simulation Assistant project:

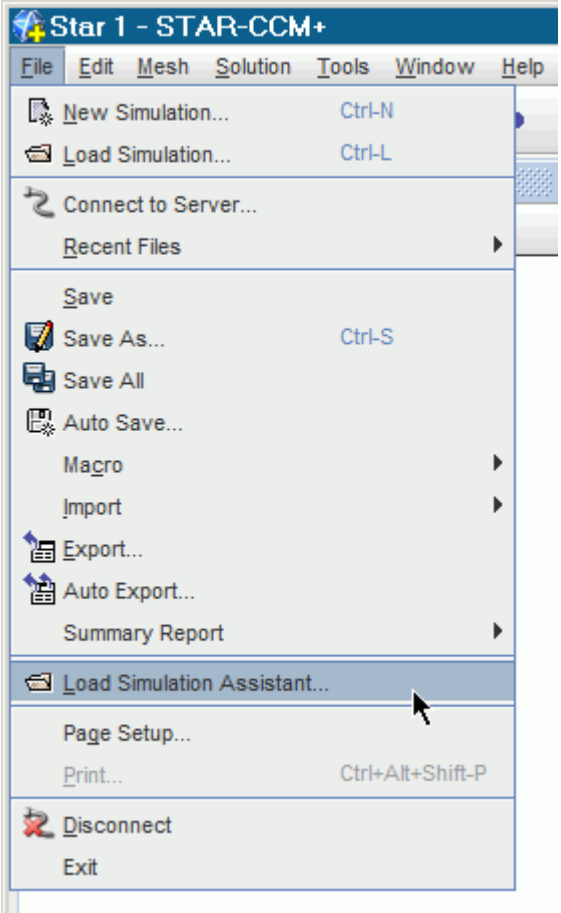
4. Select **Run** > **Build Project**.

When complete, `InternalFlowAssistant.jar` is placed in the `[Simulation_Assistant_Project]/dist` folder. This is the assistant file that you load into STAR-CCM+.

Loading the Assistant into STAR-CCM+

Load the assistant into STAR-CCM+:

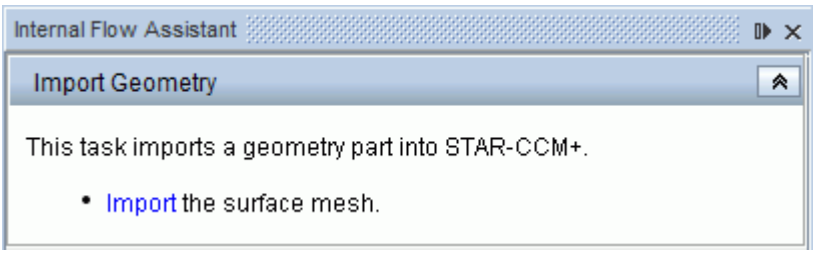
1. Load STAR-CCM+ and create a new simulation.
2. Select **File** > **Load Simulation Assistant...**.



The *Open* dialog appears.

3. Navigate to the `.jar` file for your assistant and open it.

   The assistant loads into STAR-CCM+ and appears on the right-hand side of the user-interface. At this stage, the assistant only has a single task in it.



It is recommended that you rebuild the assistant project after adding each new task. Doing so gives you the opportunity to test that each task works as expected.

Testing the Simulation Assistant

Test that the assistant behaves as expected:

1. In the *Import Geometry* panel, click **Import**.

The *Open* dialog appears.

2. Navigate to `[STAR-CCM+_Installation]/doc/startutorialsdata/simulationAssistant/data` and open `sBendGeometry.x_t`.

   The geometry is imported and placed in **Geometry** > **Parts**.

3. Close STAR-CCM+ without saving.

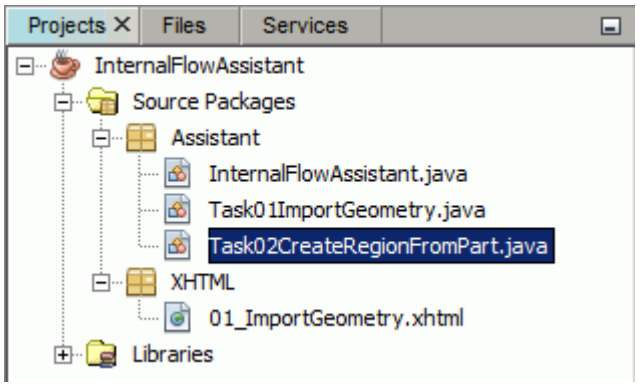
It is recommended that you rebuild the assistant project after adding each new task. Doing so gives you the opportunity to test that each task works as expected.

Creating the Second Task: Converting the Parts to Regions

The second task takes the imported part from the first task and creates a region and boundaries for it. Every new task has its own separate Java class.

For this tutorial, you are provided with the code for Task 2. The code makes sure that a geometry part exists in the simulation and, if found, creates a region for it. If no parts are found, nothing happens.

1. Add a new class to the Assistant package and name it `Task02CreateRegionFromPart`.



2. Copy the following code into the new class:

```
package Assistant;

import java.util.ArrayList;
import java.util.Collection;

import star.assistant.Task;
import star.assistant.annotation.StarAssistantTask;
import star.assistant.ui.FunctionTaskController;
import star.common.GeometryPart;
import star.meshing.CadPart;

@StarAssistantTask(display = "Create Region from Part",
    contentPath = "XHTML/02_CreateRegionFromPart.xhtml",
    controller =
Task02CreateRegionFromPart.RegionFromPartTaskController.class)
```

```
public class Task02CreateRegionFromPart extends Task {

    public Task02CreateRegionFromPart() {
    }

    public class RegionFromPartTaskController extends
FunctionTaskController {

        public void createRegion() {
            CadPart cadPart_1 = lookupObject(CadPart.class);
            if (cadPart_1 != null) {
                Collection<GeometryPart> list = new
ArrayList<GeometryPart>();
                list.add(cadPart_1);

getSimulation().getRegionManager().newRegionsFromParts(list,
                    "OneRegionPerPart", null,
"OneBoundaryPerPartSurface", null,
                    "OneFeatureCurve", null, true);
            }
        }
    }
}
```

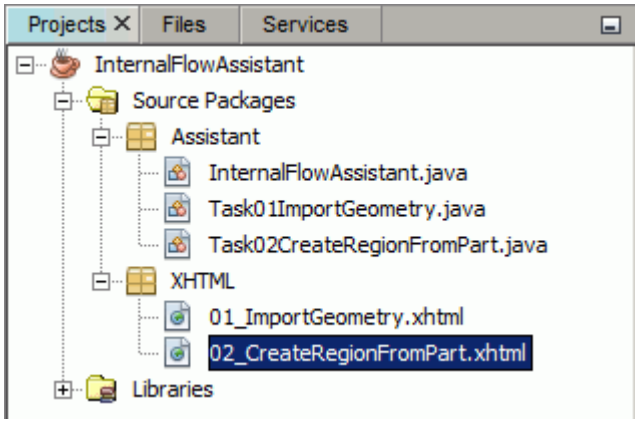
Contents:

Adding the Text to Task 2
Building and Testing the Second Task

Adding the Text to Task 2

Add the text for the second task:

1. Add an XHTML file to the HMTL package and name it `02_CreateRegionFromPart`.



2. Copy and paste the following code into the new `02_CreateRegionFromPart.xhtml`:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://
www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmls="http://www.w3.org/1999/xhtml">
    <head>
        <meta http-equiv="CONTENT-TYPE" content="text/html;
charset=utf-8" />
    </head>
    <body>
        <p>
            This task creates the regions and boundaries from the
geometry part. Renamed surfaces are used to create boundaries.
            <ul>
                <li><a href="staraction:createRegion">Create</a> a
region.</li>
            </ul>
        </p>
    </body>
</html>
```

Building and Testing the Second Task

Build and test the second task.

Before you can build the project, you must add the second task to the assistant class:

1. Return to `InternalFlowAssistant` and add the following line below the line corresponding to the first task:

```
tasks.add(new Task02CreateRegionFromPart());
```

   The final code is displayed below:

```
public InternalFlowAssistant() {

    List<Task> tasks = new ArrayList<Task>(); // Creates a new
array list for the list of tasks.
    // Contains the list of tasks.
    tasks.add(new Task01ImportGeometry());
    tasks.add(new Task02CreateRegionFromPart());
    setOutline(tasks);
    }
```

2. Save the project.
3. Rebuild the assistant project.
4. Load the rebuilt assistant into STAR-CCM+ and work through it.

   Task 1 imports the geometry into STAR-CCM+. Task 2 creates a region with the geometry from Task 1.

Creating the Third Task: Continuum Physics

Create the third task for the project.

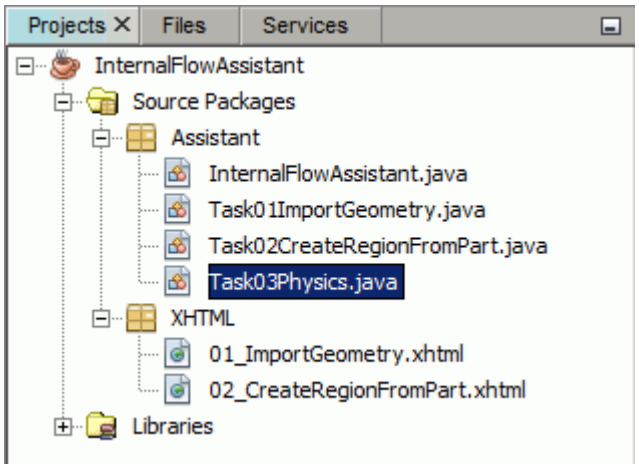
The third task contains three steps:

- The first step creates a physics continuum and populates it with physics models that are appropriate for the simulation.
- The second step modifies the material properties of the air.
- The third step sets up initial and boundary conditions, including the boundary types.

For this task, you record a Java macro and perform the desired actions in the STAR-CCM+ user interface. You then copy and paste the code for each step into your class, under the relevant function. As the task has three steps, the class will have three functions.

First, create the class for Task 3:

1. Add a new class to the Assistant package and name it `Task03Physics`.



2. Copy and paste the following code into the class:

```
package Assistant;

import star.assistant.annotation.StarAssistantTask;
import star.assistant.Task;
import star.assistant.ui.FunctionTaskController;

@StarAssistantTask(display = "Create Physics",
    contentPath = "XHTML/03_Physics.xhtml",
    controller = Task03Physics.PhysicsTaskController.class)
public class Task03Physics extends Task {

    public class PhysicsTaskController extends
FunctionTaskController {

        public void createPhysicsContinuum() {
            // code for Step 1: creating and defining the physics
```

```
continuum.
        }

        public void materialProperties() {
            // code for Step 2: modifying the material properties
of air.
        }

        public void initialConditionsAndBoundarySettings() {
            // code for Step 3: defining the initial conditions,
boundary type, and boundary conditions.
        }
    }
}
```

This code contains three functions, one function for each step:

- `createPhysicsContinuum()`
- `materialProperties()`
- `initialConditionsAndBoundarySettings()`

Contents:

Recording the Actions for Step 1
Recording the Actions for Step 2
Recording the Actions for Step 3
Reviewing the final Code for Task 3
Adding the Text to Task 3
Building and Testing the Third Task

Recording the Actions for Step 1

Record the actions for Step 1 in a macro:

1. Launch STAR-CCM+ and load the latest version of the assistant.
2. Work through the assistant until you reach the end.
3. Start recording a macro and name it `CreatingPhysicsContinuum.java`.
4. Create a physics continuum and rename it to `Physics`.
5. Edit the **Continuum** > **Physics** node and set the following properties:

Group Box	Model
Space	**Three Dimensional**
Time	**Steady**
Material	**Gas**
Flow	**Segregated Flow**

Group Box	Model
Equation of State	**Constant Density**
Viscous Regime	**Laminar**

6. Click **Close**.
7. Stop recording the macro.

   When you record a macro, a macro window appears next to the **Output** window. The code for the macro is displayed there. The following code is displayed:

```
// begin macro: CreatePhysicsContinuum
Simulation simulation_0 =
    getActiveSimulation();
PhysicsContinuum physicsContinuum_0 =

simulation_0.getContinuumManager().createContinuum(PhysicsContinuum.
class);
physicsContinuum_0.setPresentationName("Physics");
physicsContinuum_0.enable(ThreeDimensionalModel.class);
physicsContinuum_0.enable(SteadyModel.class);
physicsContinuum_0.enable(SingleComponentGasModel.class);
physicsContinuum_0.enable(SegregatedFlowModel.class);
physicsContinuum_0.enable(ConstantDensityModel.class);
physicsContinuum_0.enable(LaminarModel.class);
// end macro: CreatePhysicsContinuum
```

From here, you can copy and paste the code into the relevant function:

8. Return to NetBeans.
9. Copy and paste the code between `// begin macro: CreatePhysicsContinuum` and `// end macro` into the createPhysicsContinuum() function.
10. Reformat the macro code in NetBeans.
11. Select **Source** > **Fix Imports** to automatically add all the missing imports to the class.

Recording the Actions for Step 2

Record the actions for Step 2.

The second step involves modifying the material properties of air. Use the same technique as Step 1:

1. Return to STAR-CCM+ and record a macro named `ModifyMaterialProperties.java`.
2. Navigate to **Continua** > **Physics** > **Models** > **Gas** > **Air** > **Material Properties** and set the following properties:

Node	Property	Setting
Density	*Value*	`1.0 kg/m^3`
Dynamic	*Value*	`1.716E-5 Pa-s`

3. Stop recording the macro.

   The macro code for these actions is displayed below:

```
// begin macro: ModifyMaterialProperties
Simulation simulation_0 =
    getSimulation();
PhysicsContinuum physicsContinuum_0 =
    ((PhysicsContinuum)
simulation_0.getContinuumManager().getContinuum("Physics"));
SingleComponentGasModel singleComponentGasModel_0 =

physicsContinuum_0.getModelManager().getModel(SingleComponentGasMode
l.class);
Gas gas_0 =
    ((Gas) singleComponentGasModel_0.getMaterial());
ConstantMaterialPropertyMethod constantMaterialPropertyMethod_0 =
    ((ConstantMaterialPropertyMethod) gas_0.getMaterialProperties()
.getMaterialProperty(ConstantDensityProperty.class).getMethod());
constantMaterialPropertyMethod_0.getQuantity().setValue(1.0);
ConstantMaterialPropertyMethod constantMaterialPropertyMethod_1 =
    ((ConstantMaterialPropertyMethod) gas_0.getMaterialProperties()
.getMaterialProperty(DynamicViscosityProperty.class).getMethod());
constantMaterialPropertyMethod_1.getQuantity().setValue(1.716E-5);
// end macro: ModifyMaterialProperties
```

4. Copy and paste the code into the `materialProperties()` function.
5. Reformat the code in NetBeans.
6. Select **Source** > **Fix Imports**.

Recording the Actions for Step 3

Record the actions for Step 3.

The third step sets the initial conditions, defines the boundary types, and sets the velocity value on the Inlet boundary. The macro code for this step is provided for you:

1. Copy the following code into the `initialConditionsAndBoundarySettings()` function:

```
Simulation simulation_0 =
    getActiveSimulation();
PhysicsContinuum physicsContinuum_0 =
    ((PhysicsContinuum)
simulation_0.getContinuumManager().getContinuum("Physics"));
VelocityProfile velocityProfile_0 =

physicsContinuum_0.getInitialConditions().get(VelocityProfile.class)
;
velocityProfile_0.getMethod(ConstantVectorProfileMethod.class).getQu
antity().setComponents(0.429, 0.0, 0.0);
```

```
Region region_0 =
    simulation_0.getRegionManager().getRegion("Fluid");
Boundary boundary_0 =
    region_0.getBoundaryManager().getBoundary("Inlet");
boundary_0.setBoundaryType(InletBoundary.class);
VelocityMagnitudeProfile velocityMagnitudeProfile_0 =
    boundary_0.getValues().get(VelocityMagnitudeProfile.class);
velocityMagnitudeProfile_0.getMethod(ConstantScalarProfileMethod.cla
ss).getQuantity().setValue(0.429);
Boundary boundary_1 =
    region_0.getBoundaryManager().getBoundary("Outlet");
boundary_1.setBoundaryType(PressureBoundary.class);
```

2. Reformat the code in NetBeans.
3. Select **Source** > **Fix Imports**.
4. In the *Fix All Imports* dialog, click **OK**.

Reviewing the final Code for Task 3

Review the code for the third task.

The final code for Task 3 is displayed below:

```
package Assistant;

import star.assistant.annotation.StarAssistantTask;
import star.assistant.Task;
import star.assistant.ui.FunctionTaskController;
import star.common.Boundary;
import star.common.ConstantScalarProfileMethod;
import star.common.ConstantVectorProfileMethod;
import star.common.InletBoundary;
import star.common.PhysicsContinuum;
import star.common.PressureBoundary;
import star.common.Region;
import star.common.Simulation;
import star.common.SteadyModel;
import star.flow.ConstantDensityModel;
import star.flow.ConstantDensityProperty;
import star.flow.DynamicViscosityProperty;
import star.flow.LaminarModel;
import star.flow.VelocityMagnitudeProfile;
import star.flow.VelocityProfile;
import star.material.ConstantMaterialPropertyMethod;
import star.material.Gas;
import star.material.SingleComponentGasModel;
import star.metrics.ThreeDimensionalModel;
import star.segregatedflow.SegregatedFlowModel;
```

```
@StarAssistantTask(display = "Create Physics",
    contentPath = "XHTML/03_Physics.xhtml",
    controller = Task03Physics.PhysicsTaskController.class)
public class Task03Physics extends Task {

    public class PhysicsTaskController extends FunctionTaskController {

        public void createPhysicsContinuum() {

            // code for Step 1: creating and defining the physics
continuum.
            Simulation simulation_0 =
                getActiveSimulation();
            PhysicsContinuum physicsContinuum_0 =

simulation_0.getContinuumManager().createContinuum(PhysicsContinuum.cla
ss);
            physicsContinuum_0.setPresentationName("Physics");
            physicsContinuum_0.enable(ThreeDimensionalModel.class);
            physicsContinuum_0.enable(SteadyModel.class);
            physicsContinuum_0.enable(SingleComponentGasModel.class);
            physicsContinuum_0.enable(SegregatedFlowModel.class);
            physicsContinuum_0.enable(ConstantDensityModel.class);
            physicsContinuum_0.enable(LaminarModel.class);
        }

        public void materialProperties() {

            // code for Step 2: modifying the material properties of
air.
            Simulation simulation_0 = getSimulation();
            PhysicsContinuum physicsContinuum_0 =
                ((PhysicsContinuum)
simulation_0.getContinuumManager().getContinuum("Physics"));
            SingleComponentGasModel singleComponentGasModel_0 =

physicsContinuum_0.getModelManager().getModel(SingleComponentGasModel.c
lass);
            Gas gas_0 =
                Gas) singleComponentGasModel_0.getMaterial());
            ConstantMaterialPropertyMethod
constantMaterialPropertyMethod_0 =
                ConstantMaterialPropertyMethod)
gas_0.getMaterialProperties()

getMaterialProperty(ConstantDensityProperty.class).getMethod());
```

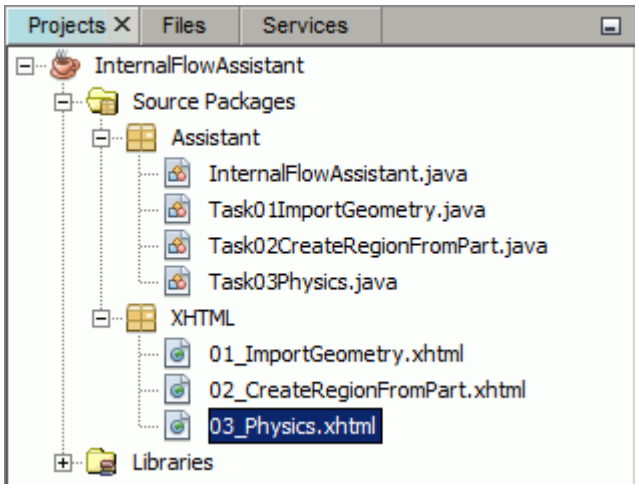
```
constantMaterialPropertyMethod_0.getQuantity().setValue(1.0);
            ConstantMaterialPropertyMethod
constantMaterialPropertyMethod_1 =
                ConstantMaterialPropertyMethod)
gas_0.getMaterialProperties()

getMaterialProperty(DynamicViscosityProperty.class).getMethod());

constantMaterialPropertyMethod_1.getQuantity().setValue(1.716E-5);
        }

        public void initialConditionsAndBoundarySettings() {

            // code for Step 3: defining the initial conditions,
boundary type, and boundary conditions.
            Simulation simulation_0 =
                getActiveSimulation();
            PhysicsContinuum physicsContinuum_0 =
                PhysicsContinuum)
simulation_0.getContinuumManager().getContinuum("Physics"));
            VelocityProfile velocityProfile_0 =

physicsContinuum_0.getInitialConditions().get(VelocityProfile.class);

velocityProfile_0.getMethod(ConstantVectorProfileMethod.class).getQuant
ity().setComponents(0.429, 0.0, 0.0);
            Region region_0 =
                simulation_0.getRegionManager().getRegion("Fluid");
            Boundary boundary_0 =
                region_0.getBoundaryManager().getBoundary("Inlet");
            boundary_0.setBoundaryType(InletBoundary.class);
            VelocityMagnitudeProfile velocityMagnitudeProfile_0 =

boundary_0.getValues().get(VelocityMagnitudeProfile.class);

velocityMagnitudeProfile_0.getMethod(ConstantScalarProfileMethod.class)
.getQuantity().setValue(0.429);
            Boundary boundary_1 =
                region_0.getBoundaryManager().getBoundary("Outlet");
            boundary_1.setBoundaryType(PressureBoundary.class);
        }
    }
}
```

Adding the Text to Task 3

Add the descriptions to the third task.

Now that Task 3 is complete, you can add the text and link the star actions to the relevant functions:

1. Add an XHTML file to your XHTML package and name it `03_Physics`.



2. Copy the following code into `03_Physics.xhtml`:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://
www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmls="http://www.w3.org/1999/xhtml">
    <head>
        <meta http-equiv="CONTENT-TYPE" content="text/html;
charset=utf-8" />
    </head>
    <body>
        <p>
            <!--Description in task panel.-->
            This task creates a physics continuum and populates it
            with the appropriate physics models.
            <ol>
                <!--First list item.-->
                <li><a
href="staraction:createPhysicsContinuum">Create</a> a physics
continuum.</li>
                <!--Second list item.-->
                <li><a
href="staraction:materialProperties">Define</a> the material
properties.</li>
                <!--Third list item.-->
                <li><a
href="staraction:initialConditionsAndBoundarySettings">Set</a>
                    the initial conditions and boundary settings.</
li>
            </ol>
        </p>
```

```
    </body>
</html>
```

Note: Each step is defined in a new list item. This task has three steps, so there are three list items, each of which is linked to a star action.

Building and Testing the Third Task

Build and test the third task.

Before building the project, add the third task to the assistant class:

1. Return to `InternalFlowAssistant` and add the following line below the line corresponding to the second task:

```
tasks.add(new Task03Physics());
```

2. Save the project.
3. Rebuild the project.
4. Test the assistant in STAR-CCM+.

Creating the Remaining Tasks

Create the remaining three tasks for the project.

The remaining tasks and associated XHTML files are already created for you to import into the project. An overview of what the tasks do is provided below.

Task 4

The fourth task deals with the meshing. This task is split into four steps:

- Step 1 creates a mesh continuum and populates it with the appropriate mesh models.
- Step 2 sets up the mesh parameters, such as base size.
- Step 3 generates the surface mesh.
- Step 4 generates the volume mesh.

Task 5

The fifth task creates a scalar scene that contains a plane-section along the center of the imported geometry.

Task 6

The sixth task is split into two steps:

- Step 1 sets the stopping criteria.
- Step 2 runs the simulation.

Contents:

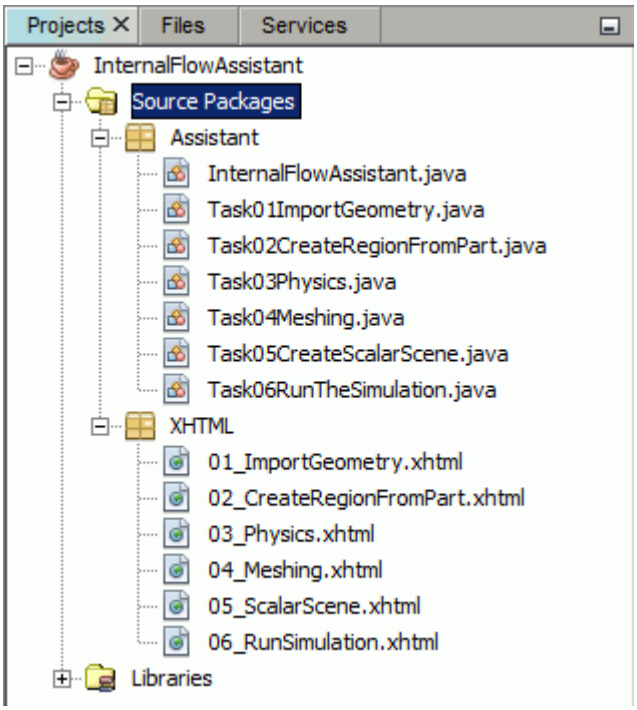
Adding the Remaining Tasks to Your Project
Test the Complete Assistant

Adding the Remaining Tasks to Your Project

Add the remaining tasks to your project:

1. In the file system, navigate to `[STAR-CCM+_Installation]/doc/startutorialsdata/simulationAssistant/data`.
2. Copy the following Java files into the `[Simulation_Assistant_Project]/src/Assistant` folder:
   - `Task04Meshing.java`
   - `Task05CreateScalarScene.java`
   - `Task06RunTheSimulation.java`
3. Copy and paste the following XHTML files into the `[Simulation_Assistant_Project]/src/XHTML` folder:
   - `04_Meshing.xhtml`
   - `05_ScalarScene.xhtml`
   - `06_RunSimulation.xhtml`
4. Save the project.

The tasks are automatically added to the NetBeans project.



Test the Complete Assistant

Perform a final test on the complete assistant:

1. Return to `InternalFlowAssistant` and add the missing tasks to the array list:

```
tasks.add(new Task04Meshing());
tasks.add(new Task05CreateScalarScene());
tasks.add(new Task06RunTheSimulation());
```

2. Save the project.
3. Rebuild the project.
4. Launch STAR-CCM+ and load the rebuilt assistant.
5. Work through the assistant and make sure that it behaves as expected:
   - Task 1 loads a geometry part into STAR-CCM+.
   - Task 2 creates a region from the imported geometry part.
   - Task 3 creates a physics continuum and sets up the physics models, initial conditions, and boundary conditions.
   - Task 4 creates a mesh continuum and sets up the mesh parameters. This task also covers the steps to generate the surface and volume meshes.
   - Task 5 creates a scalar scene.
   - Task 6 sets the stopping criteria and runs the STAR-CCM+ solver.

Advanced Simulation Assistant Features

There are multiple advanced Simulation Assistant features that you can use to improve the code. The following features are described in this section:

- Adding conditions to the task.
- Expanding the object tree to reveal specific objects.
- Adding information buttons within a task panel.

Contents:

Adding Conditions to the Assistant
Adding a Post-Condition to Task 1
Adding a Pre-Condition to Task 2
Testing the Conditions

Adding Conditions to the Assistant

Add a post-condition to Task 1 and a pre-condition to Task 2.

In a Simulation Assistant, conditions impose requirements that must be met at specific points in the workflow. There are two types of conditions:

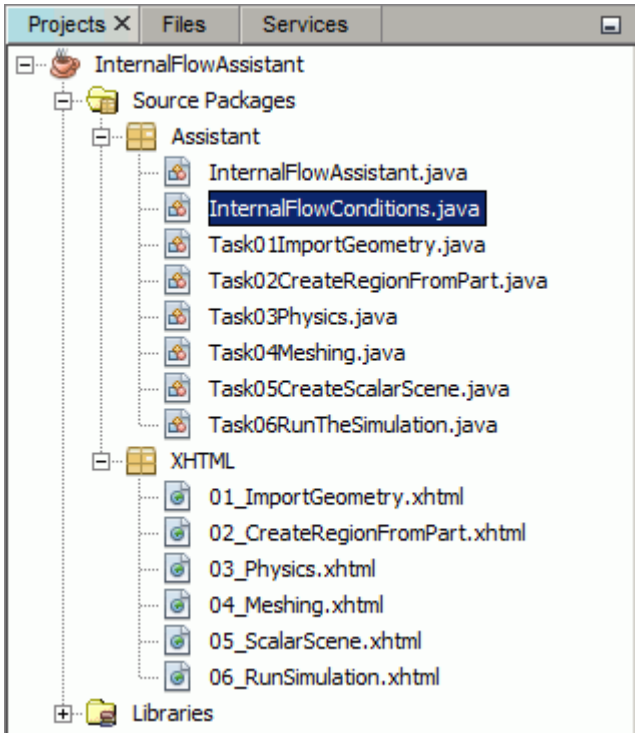
- Pre-conditions—define what is required before a task can begin.

- Post-conditions—check that the requirements of a task are met.

In this tutorial, you add a post-condition to Task 1, and a pre-condition to Task 2. The condition checks whether a geometry file part exists in the simulation. If no part exists, Task 1 is not complete and Task 2 is disabled.

It is good practice to add the code for all conditions in a separate class:

1. Add a new class to the Assistant package and name it `InternalFlowConditions`.



2. Copy the following code into the `InternalFlowConditions` class. Comments are included in the code:

```
package Assistant;

import java.util.Collections;
import star.assistant.CSOCondition;
import star.assistant.CSOLookupConditionTrigger;
import star.common.GeometryPart;
import star.common.filters.Predicate;

/**
 * This class contains all conditions used in the Internal Flow
Assistant.
 */
public class InternalFlowConditions {

    public static synchronized CSOCondition<GeometryPart>
```

```
createPartCondition() {

        // Creates a new condition
        CSOCondition<GeometryPart> partCondition = new
CSOCondition<GeometryPart>();
        // Sets the text description of the condition
        partCondition.setDesc("A geometry part must be present.");
        // Creates a new condition trigger that goes off when a
cadpart is added to the lookup
        CSOLookupConditionTrigger<GeometryPart>
partConditionTrigger = new
CSOLookupConditionTrigger<GeometryPart>(GeometryPart.class);
        // Sets the list of triggers to the one created above.

partCondition.setTriggers(Collections.singleton(partConditionTrigger
));
        // Creates a new predicate (true/false evaluation) with an
evaluate method and evaluates whether an object satisfies the
condition.
        partCondition.setPredicate(new Predicate<GeometryPart>() {
            @Override
            public boolean evaluate(GeometryPart part) {
                // You could check for specific attributes of the
part here.
                return true;
            }
        });
        return partCondition;
    }
}
```

Adding a Post-Condition to Task 1

The following code adds the post-condition to Task 1. The post-condition checks whether a geometry part exists in the **Geometry** > **Parts** manager node. If a part exists, the condition is satisfied.

1. Return to the `Task01ImportGeometry` class.
2. Locate the `public Task01ImportGeometry()` constructor.
3. Add the following code to this constructor:

```
// Makes sure that a geometry part is created.
setPostconditions(Collections.singleton(InternalFlowConditions.creat
ePartCondition()));
```

where:

- `setPostconditions()` sets the post-condition for the task.

- `Collections.singleton(InternalFlowConditions.createPartCondition())` is the condition that checks for a geometry part.

4. Add the missing import.

Adding a Pre-Condition to Task 2

Add a pre-condition to Task 2.

You cannot create a region from a part if there are no parts in the simulation, so it makes sense to add a pre-condition to Task 2, that disables that task if no geometry parts exist.

The following code adds a pre-condition to Task 2 to check whether a part exists in the `Geometry > Parts` manager node. If no part exists, the task is disabled (greyed out). If a part exists, the condition is satisfied and the task is enabled. Pre-conditions only affect the task that they are placed in.

1. Return to the `Task02CreateRegionFromPart` class.
2. Locate the `public Task02CreateRegionFromPart()` constructor.
3. Add the following code to this constructor:

```
// Makes sure that a geometry exists before enabling this task.
setPreconditions(Collections.singleton(InternalFlowConditions.create
PartCondition()));
```

   where:

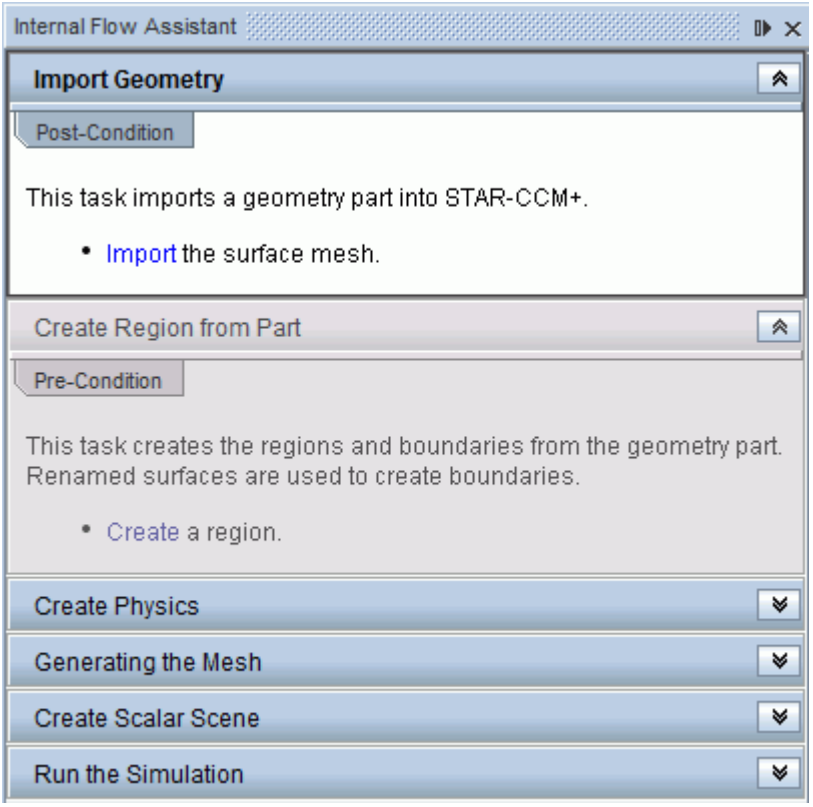
   - `setPreconditions()` sets the pre-condition for the task.
   - `Collections.singleton(InternalFlowConditions.createPartCondition())` is the condition that checks for a geometry part.

4. Add the missing import.
5. Save and build your project.

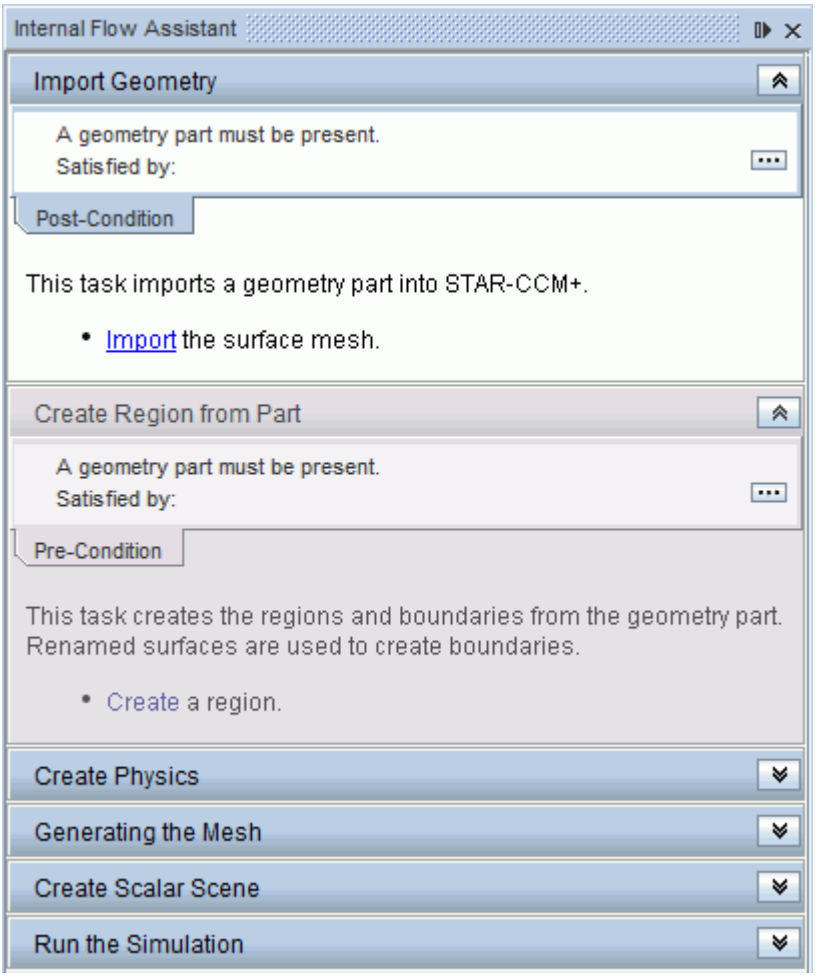
Testing the Conditions

Test the conditions to make sure that they behave as expected:

1. Load the assistant into STAR-CCM+.
2. Task 1 and Task 2 have condition tabs added to their panels.

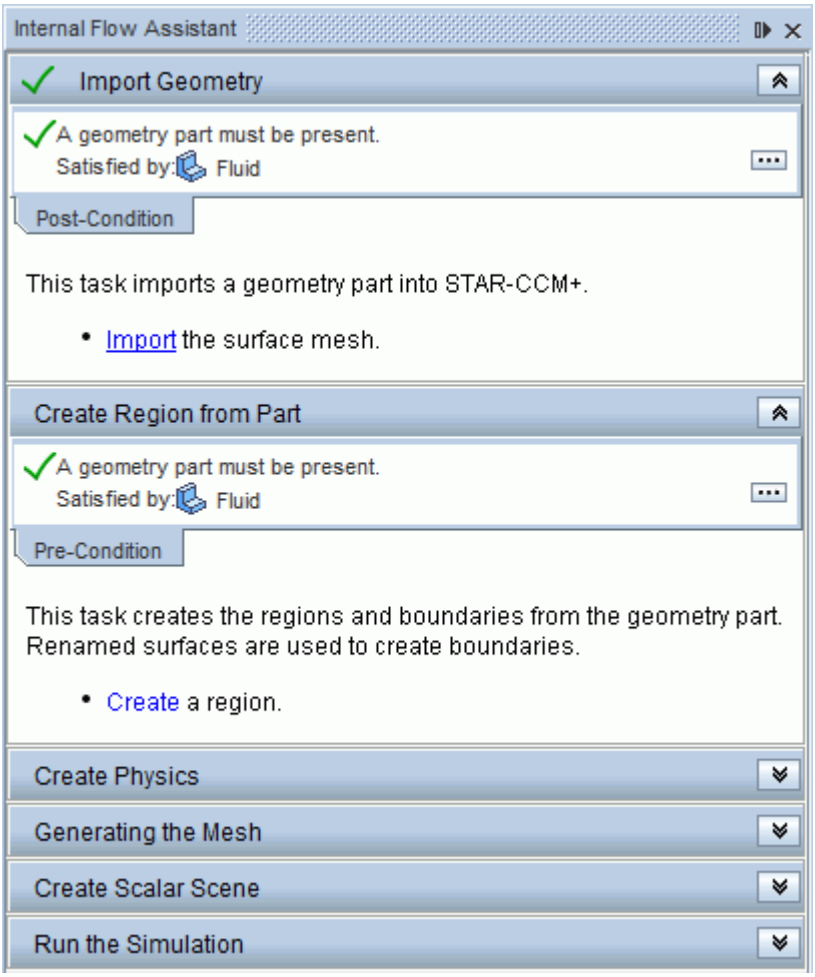


3. Click on the **Post-Condition** tab in Task 1 and the **Pre-Condition** tab in Task 2 to expand them and display more information about the condition requirements.



4. Import the geometry part into STAR-CCM+.

   The conditions for both tasks are satisfied and Task 2 becomes available. When a Post-Condition is satisfied, a green checkmark is displayed on the task panel title bar. The object that satisfies the condition is also displayed in the condition tab.



Expanding the Object Tree to Reveal Specific Objects

Add an exclusive and an inclusive focus to the assistant.

When working through the workflow of an assistant, it is helpful to see what part of the tree an action is associated with. When you click on a link in the assistant, you can cause the relevant node in the STAR-CCM+ object tree to expand.

There are two types of object tree focus:

- Exclusive focus: collapses all nodes in the simulation tree and only expands the node that you select.
- Inclusive focus: expands the node for the item that you select. All other nodes remain unchanged.

In this tutorial, you add an object tree focus to Task 3 > Step 1 and Task 3 > Step 2.

Adding an Exclusive Focus to Step 1

Add an object tree focus so that the **Continua** > **Physics** > **Models** manager node expands in the simulation tree when you click the link in the Simulation Assistant:

1. Return to the `Task03Physics` class.

2. Within the `createPhysicsContinuum()` method, insert the following code after the last statement:

```
// Object tree focus that expands the physics model manager node.
All other open nodes are collapsed.
selectAndExpandNodeExclusive(physicsContinuum_0.getModelManager());
```

The final code for the first step in Task 3 is displayed below:

```
public void createPhysicsContinuum() {

    // code for Step 1: creating and defining the physics continuum.
    Simulation simulation_0 =
        getActiveSimulation();
    PhysicsContinuum physicsContinuum_0 =

simulation_0.getContinuumManager().createContinuum(PhysicsContinuum.
class);
    physicsContinuum_0.setPresentationName("Physics");
    physicsContinuum_0.enable(ThreeDimensionalModel.class);
    physicsContinuum_0.enable(SteadyModel.class);
    physicsContinuum_0.enable(SingleComponentGasModel.class);
    physicsContinuum_0.enable(SegregatedFlowModel.class);
    physicsContinuum_0.enable(ConstantDensityModel.class);
    physicsContinuum_0.enable(LaminarModel.class);
    //Object tree focus that expands the physics model manager
node. All other open nodes are collapsed.

selectAndExpandNodeExclusive(physicsContinuum_0.getModelManager());
}
```

Adding an Inclusive Focus to Step 2

Add an object tree focus that so that the **Continua** > **Physics** > **Materials** > **Dynamic Viscosity** node expands in the simulation tree when you click the link in the Simulation Assistant:

1. Within the `materialProperties()` method, insert the following code after the last statement:

```
//Object tree focus that expands the Dynamic Viscosity node without
affecting other open nodes.
selectAndExpandNode(gas_0.getMaterialProperties()
    .getMaterialProperty(DynamicViscosityProperty.class).getMethod()
);
```

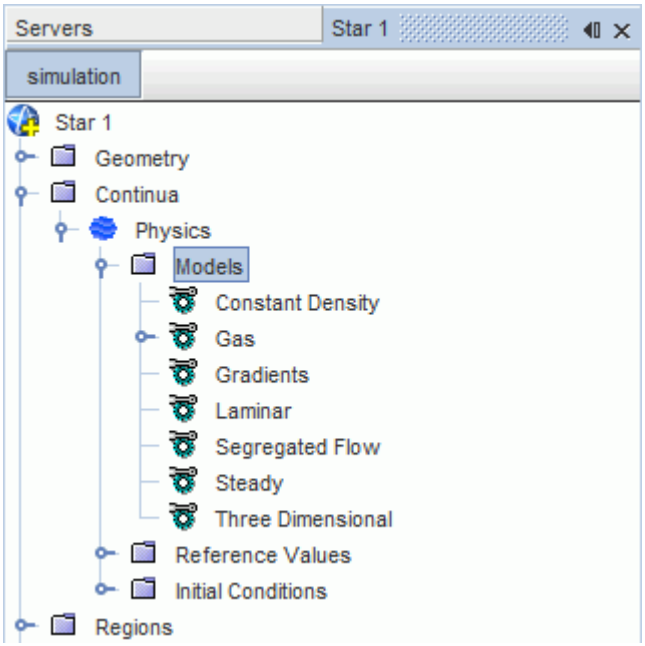
The final code for the first step in Task 3 is displayed below:

```
public void materialProperties() {

    // code for Step 2: modifying the material properties of air.
    Simulation simulation_0 = getSimulation();
```

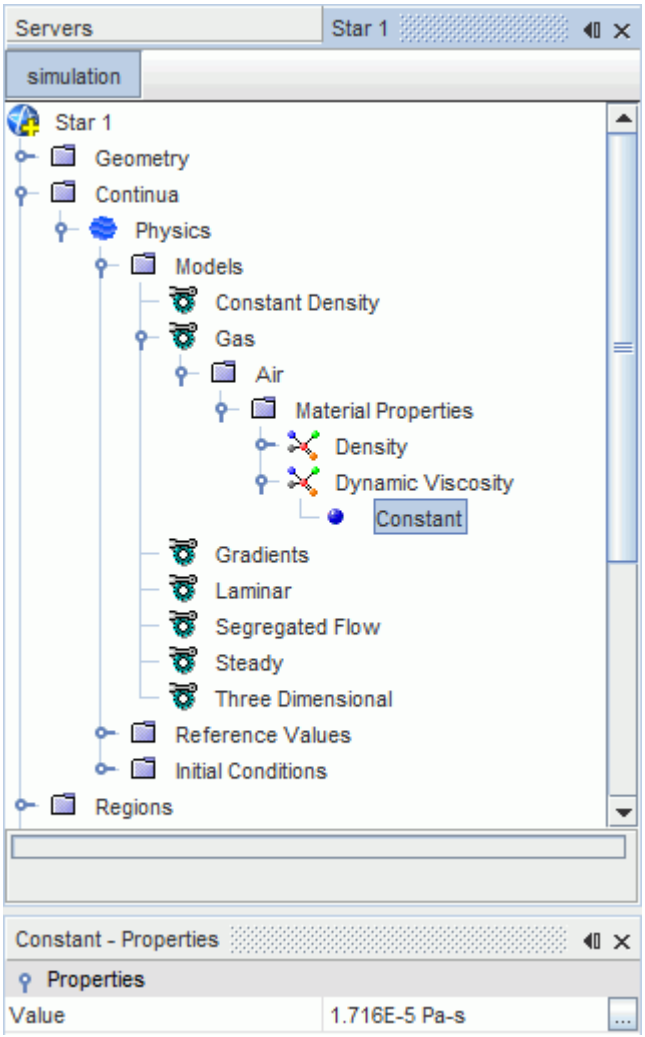
```
    PhysicsContinuum physicsContinuum_0 =
        PhysicsContinuum)
simulation_0.getContinuumManager().getContinuum("Physics"));
    SingleComponentGasModel singleComponentGasModel_0 =

physicsContinuum_0.getModelManager().getModel(SingleComponentGasMode
l.class);
    Gas gas_0 =
        Gas) singleComponentGasModel_0.getMaterial());
    ConstantMaterialPropertyMethod constantMaterialPropertyMethod_0
=
        ConstantMaterialPropertyMethod)
gas_0.getMaterialProperties()
        .getMaterialProperty(ConstantDensityProperty.class).getMetho
d());
    constantMaterialPropertyMethod_0.getQuantity().setValue(1.0);
    ConstantMaterialPropertyMethod constantMaterialPropertyMethod_1
=
        ConstantMaterialPropertyMethod)
gas_0.getMaterialProperties()
        .getMaterialProperty(DynamicViscosityProperty.class).getMeth
od());

constantMaterialPropertyMethod_1.getQuantity().setValue(1.716E-5);
    Object tree focus that expands the Dynamic Viscosity node
without affecting other open nodes.
    selectAndExpandNode(gas_0.getMaterialProperties()
        .getMaterialProperty(DynamicViscosityProperty.class).getMeth
od());
}
```

Testing the Object Tree Focus

Rebuild the project and test the modified assistant in STAR-CCM+. The first link in Task 3 expands the **Continua** > **Physics** > **Models** manager node.



The link in the second step expands the **Dynamic Viscosity** node without collapsing other open nodes.



Adding an Information Tag to the Assistant

Add an information tag to the assistant.

You can add information tags to a panel description or step. Information tags can contain more information about a specific step or task, but are not necessary for the task to be complete. Information tags are minimized by default, and can be expanded when required. The extra information is placed inside an information element in the XHTML file:

```
<information>Additional information goes here.</information>
```

For this tutorial, you add the names of the physics models which are created in the first step of Task 3.

To add an information tag:

1. Return to `03_Physics.xhtml`.

2. Add the information tag to the first list item:

```
<ol>
    <!--First list item.-->
    <li><a href="staraction:createPhysicsContinuum">Create</a> a
physics continuum.
        <!--Adds an information tag to first list item.-->
        <information>The following physics models are suitable for
this case:
            <ul>
                <li>Three Dimensional</li>
                <li>Gradients</li>
                <li>Steady</li>
                <li>Gas</li>
                <li>Segregated Flow</li>
                <li>Constant Density</li>
                <li>Laminar</li>
            </ul>
        </information></li>
```

3. Save your project and rebuild it.

Contents:

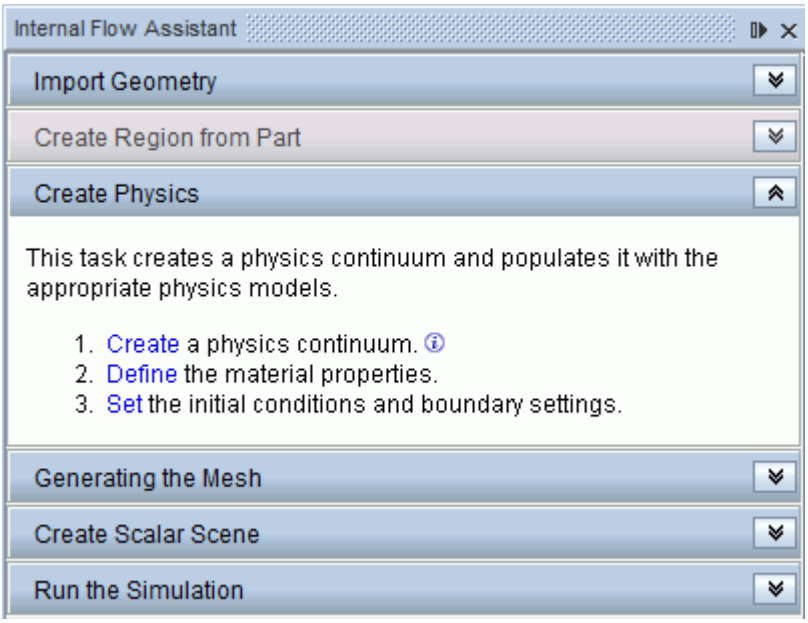
Testing the Information Tag

Testing the Information Tag

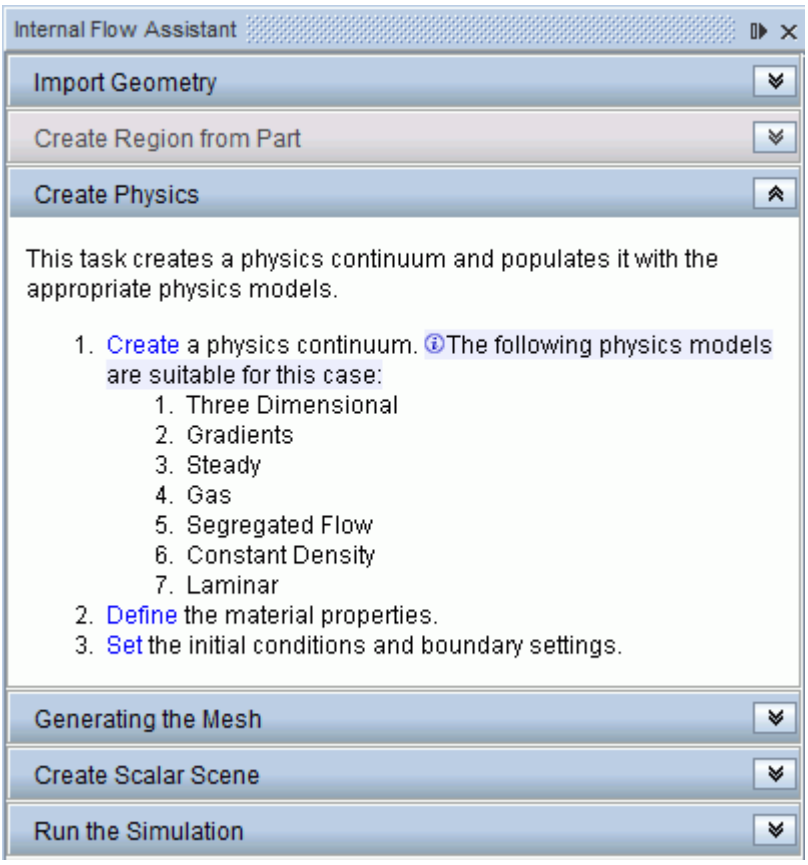
Test the information tag:

1. Launch STAR-CCM+ and test the new feature.

   A icon appears next to the first step in Task 3.



2. Click ⓘ to display the additional information.



3. Click ⓘ again to collapse the additional information.

Summary

This tutorial has introduced the following STAR-CCM+ features:

- Creating a Simulation Assistant project in NetBeans.
- Creating Simulation Assistant tasks, links and descriptions.
- Formatting the Simulation Assistant code and building the project.
- Loading and testing the Simulation Assistant code in STAR-CCM+.
- Adding conditions to the Simulation Assistant.
- Adding an object tree focus to the Simulation Assistant.
- Adding collapsible information to the Simulation Assistant.

Coupling with CAE Codes

The tutorials in this set illustrate various STAR-CCM+ features for coupling with CAE codes:

Contents:

STAR-CCM+ to STAR-CCM+ Coupling: Heat Transfer in a Chimney
LMS Amesim Co-Simulation: 1D Coupling
Abaqus File-Based Coupling: Exhaust Manifold
Abaqus Co-Simulation: Thermal Coupling
Abaqus Co-Simulation: Mechanical Coupling
GT-SUITE Co-Simulation: 1D Coupling
Co-Simulation API: Spindle Valve

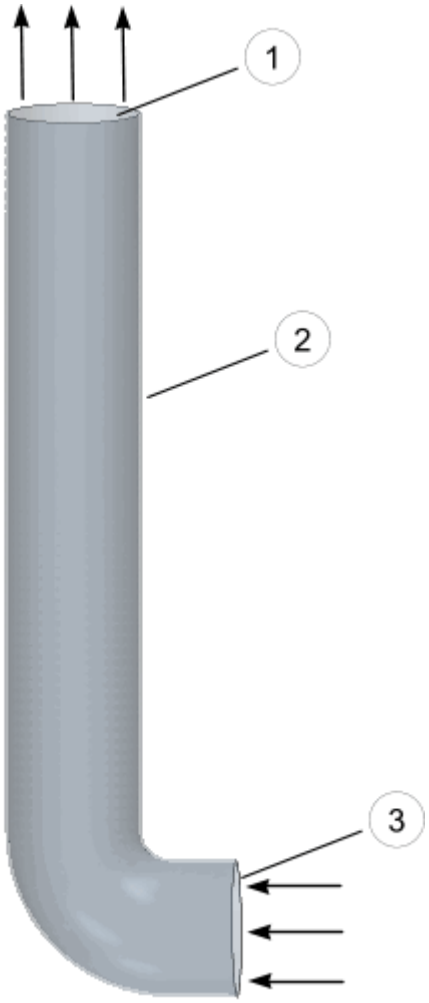
STAR-CCM+ to STAR-CCM+ Coupling: Heat Transfer in a Chimney

This tutorial demonstrates the process for modeling conjugate heat transfer using STAR-CCM+ to STAR-CCM+ thermal co-simulation.

Using STAR-CCM+ to STAR-CCM+ co-simulation to solve multi-physics problems has the advantages that:

- Regions can be developed independently.
- Each case can be modeled with a relevant time-scale.

The diagram shows the case to be modeled. Hot exhaust gas enters a metal chimney with a temperature of 600 K and an average velocity of 0.2 m/s. The chimney has internal diameter of 0.4 m, a wall thickness of 10 mm and is made of T316 austenitic stainless steel. The external environment is assumed to be a perfect heat sink at a temperature of 283 K.



The tutorial is divided into three sections:

1. In the first section, you create the solid model of the chimney.
2. In the second section, you prepare the fluid model.
3. In the third section, you couple both simulations to model the complete fluid-solid system.

Contents:

Prerequisites
Preparing the Solid Model Simulation
Preparing the Fluid Model Simulation
Coupling the Simulations
Summary

Prerequisites

The instructions in the STAR-CCM+ to STAR-CCM+ Coupling tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Techniques	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Preparing the Solid Model Simulation

Prepare the simulation for the solid chimney and run it to convergence.

Contents:

Importing the Solid Volume Mesh
Selecting Solid Physics Models
Defining the Solid Material
Defining the Solid Boundary and Initial Conditions
Preparing a Temperature Scene for the Solid
Setting Stopping Criteria for the Solid Simulation
Running the Solid Simulation
Visualizing the Decoupled Solid Results

Importing the Solid Volume Mesh

To set up the STAR-CCM+ simulation for the solid, launch a simulation and import the supplied volume mesh representing the chimney.

1. Launch STAR-CCM+.
2. Start a simulation.

3. Select **File** > **Import** > **Import Volume Mesh...**
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/couplingWithCaeCodes/data/`.
5. Select `chimney.ccm`, then click **Open**.

   STAR-CCM+ provides feedback on the import process in the *Output* window and a mesh region is created under the **Regions** node.
6. Save the simulation as `Chimney.sim`.

Selecting Solid Physics Models

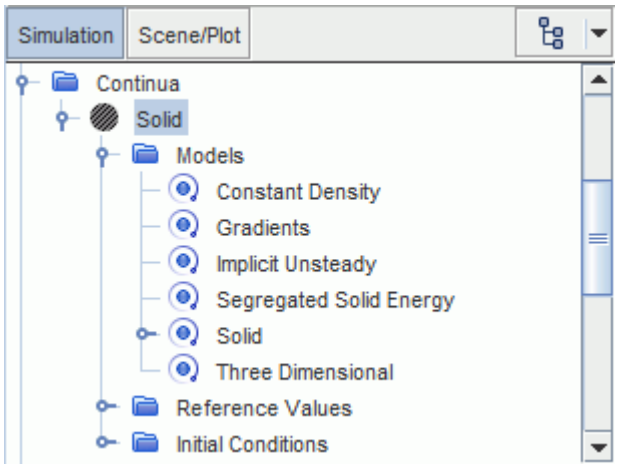
Select physics models to solve for the solid region.

To simulate heat conduction through a solid, choose an energy transport model and an equation of state.

1. Rename the **Continua** > **Physics 1** node to `Solid`.
2. For the physics continuum, **Continua** > **Solid**, select the following models in order:

Group Box	Model
Enabled Models	**Three Dimensional** (Selected automatically)
Time	**Implicit Unsteady**
Material	**Solid**
Optional Models	**Segregated Solid Energy**
	Gradients (Selected automatically)
Equation of State	**Constant Density**

3. Click **Close**.
4. To review the models, open the **Solid** > **Models** node.



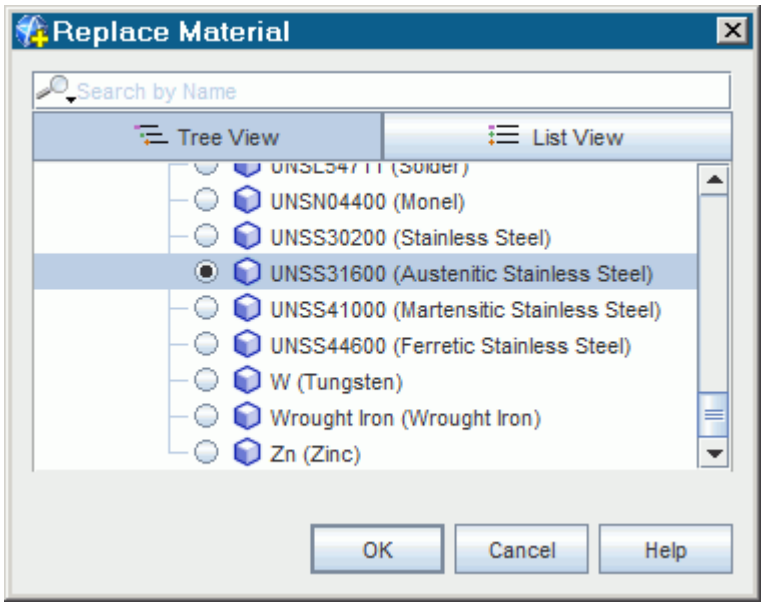
5. Save the simulation.

Defining the Solid Material

Set the solid material to stainless steel.

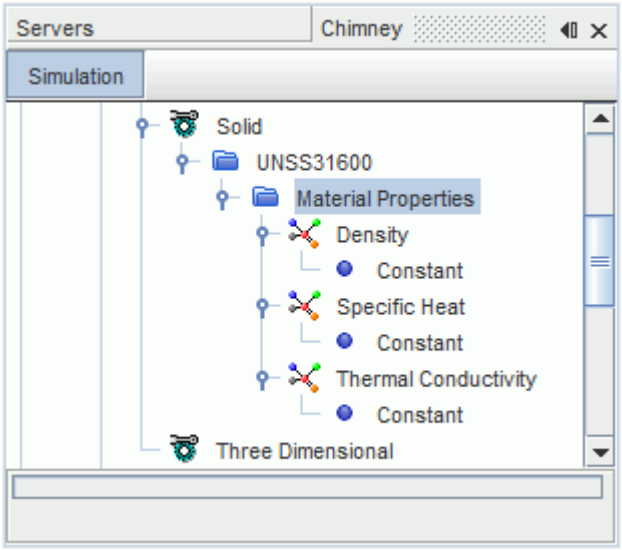
To set the solid material:

1. Right-click the **Continua** > **Solid** > **Models** > **Solid** > **AI** node and select **Replace with...**
2. In the *Replace Material* dialog, expand **Material Databases** > **Standard** > **Solids** and select *UNSS31600 (Austenitic Stainless Steel)*.



3. Click **OK**.

   The **AI** material node is replaced with **UNSS31600**.
4. Expand the **Material Properties** node to view the density, specific heat, and thermal conductivity properties.



The default values are appropriate for this tutorial.

Defining the Solid Boundary and Initial Conditions

Set boundary conditions for the solid region and the initial temperature.

For the independent simulation of the chimney, set the outer boundary to lose heat by convection, and the inner wall boundary to maintain a constant temperature. The **OuterSurf** boundary is exposed to air at 283 K with a heat transfer coefficient of 0.12 W/m^2-K.

1. Expand the **Regions** > **Chimney** > **Boundaries** node.

To apply the convection boundary:

2. Edit the **OuterSurf** node, and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Thermal Specification**	*Condition*	**Convection**
Physics Values		
└ **Ambient Temperature**	*Value*	`283.0 K`
└ **Heat Transfer Coefficient**	*Value*	`0.12 W/m^2-K`

To apply the constant temperature boundary:

3. Edit the **InnerSurf** node, and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Thermal Specification**	*Condition*	**Temperature**
Physics Values		
└ **Static Temperature**	*Value*	`380.0 K`

Leave the **Default** boundary as adiabatic.

4. Select the **Continua** > **Solid** > **Initial Conditions** > **Static Temperature** node and set *Value* to `283.0 K`.

Preparing a Temperature Scene for the Solid

Create a scene of the temperature for the entire chimney and create a file to record the transient solution history of the chimney.

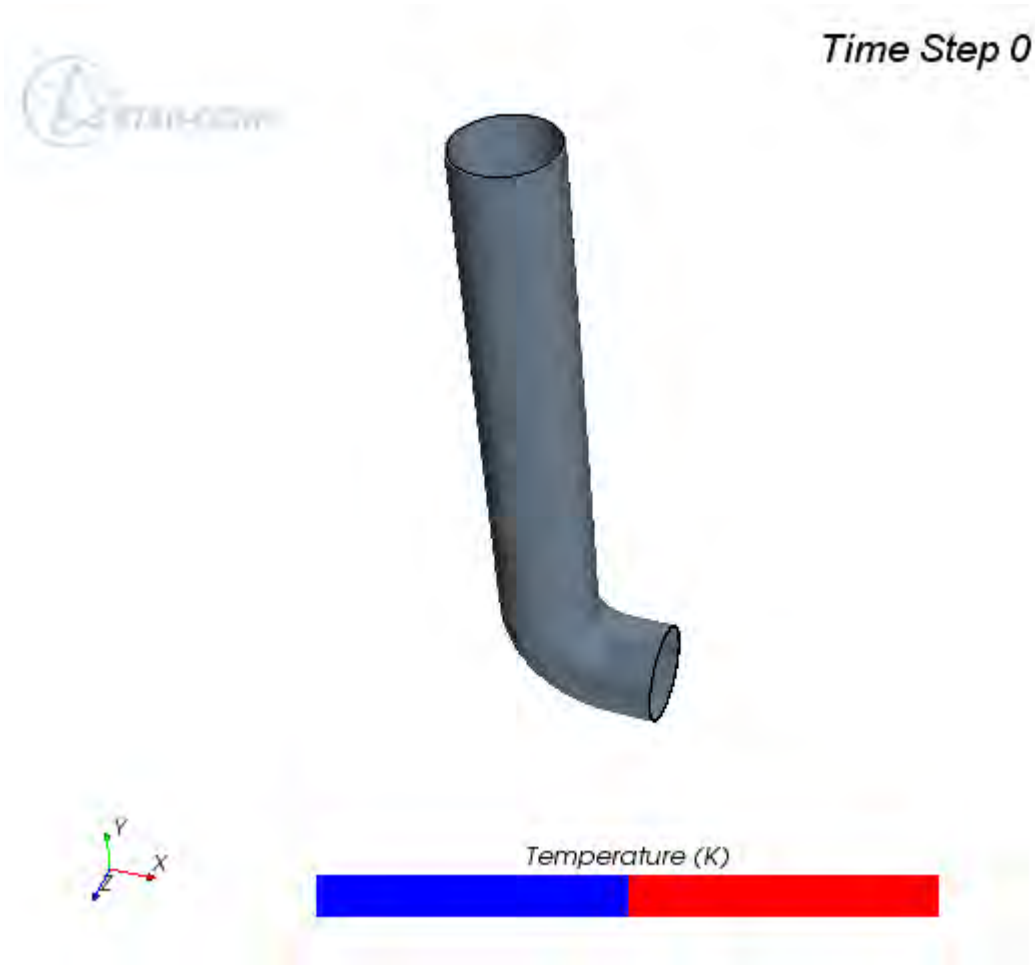
To create a temperature scene:

1. Create a scalar scene.
2. Rename the **Scalar Scene 1** node to `Wall Temperature`.
3. Edit the **Wall Temperature** > **Displayers** node and set the following properties:

Node	Property	Setting
Scalar 1		
└ **Parts**	*Parts*	**Default**
		InnerSurf
		OuterSurf
└ **Scalar Field**	*Function*	**Temperature**
	Clip	**Off**
	Min	`280`
	Max	`400`

Add an annotation to the scene to display each time step:

4. Select the **Tools** > **Annotations** > **Time Step** node and drag it onto the *Wall Temperature* scene.
5. Click the annotation text in the *Wall Temperature* scene and move it to the upper right corner of the solution scene.
6. Position the chimney as shown below:



Set update trigger to time step:

7. Select the **Scenes** > **Wall Temperature** > **Attributes** > **Update** node and set *Trigger* to **Time Step**.

Record the transient solution history of the chimney:

8. Right-click the **Solution Histories** node and select **New...**
9. In the *Save* dialog, save the solution history to your working directory as `ChimneyHistory.simh`.
10. Edit the **Solution Histories** node and set the following properties:

Node	Property	Setting
ChimneyHistory	*Functions*	**Temperature**
└ **Update**	*Trigger*	**Time Step**

11. Save the simulation.

Setting Stopping Criteria for the Solid Simulation

Set solver parameters and stopping criteria for the simulation.

Run the case for 5 minutes with a time step of 20 seconds:

1. Select the **Solvers** > **Implicit Unsteady** node and set *Time-Step* to `20 s`.
2. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Inner Iterations	*Maximum Inner Iterations*	`10`
Maximum Physical Time	*Maximum Physical Time*	`300 s`

Running the Solid Simulation

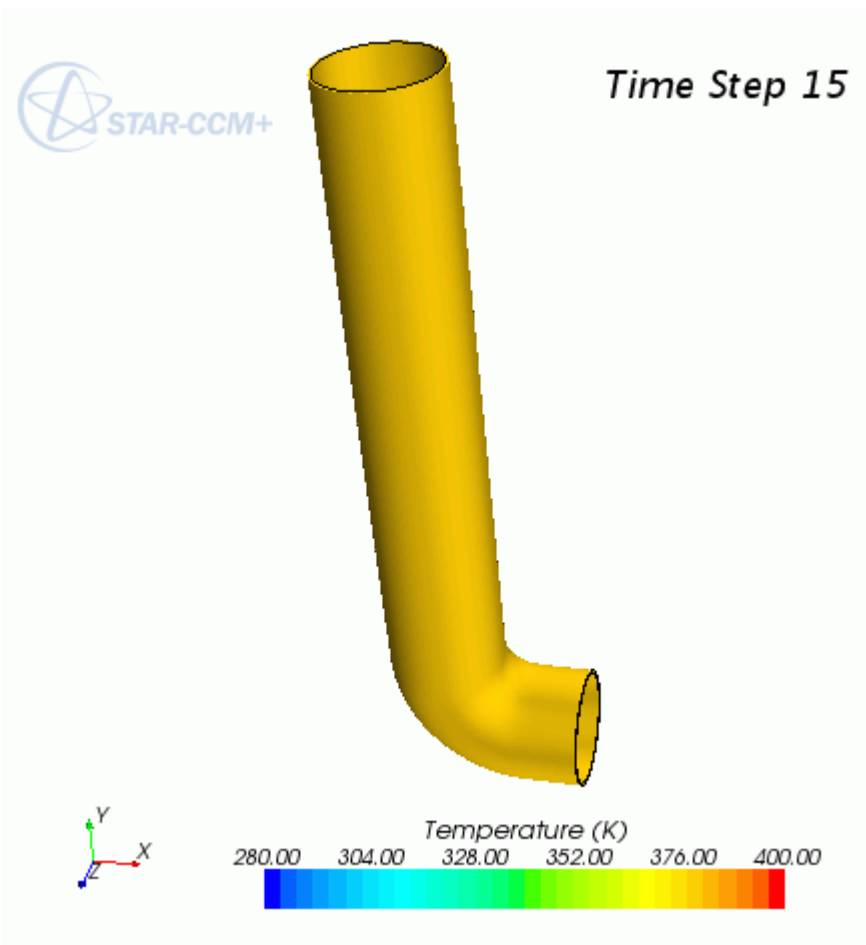
Run the simulation for the solid chimney.

1. Run the simulation.
2. When the solution finishes, save the simulation.

Visualizing the Decoupled Solid Results

Visualize the temperature distribution on the chimney outer wall.

The temperature distribution on the outer wall is uniform along the outer wall of the pipe.



These results can be compared against the coupled results later.

1. Save and close the simulation.

Preparing the Fluid Model Simulation

Prepare the simulation for the fluid inside the chimney and run it to convergence.

Contents:

Importing the Fluid Volume Mesh
Selecting Fluid Physics Models
Specifying Fluid Material Properties
Defining the Fluid Boundary and Initial Conditions
Preparing a Temperature Scene for the Fluid
Setting Stopping Criteria and Running the Fluid Simulation
Visualizing the Decoupled Fluid Results

Importing the Fluid Volume Mesh

To set up the STAR-CCM+ simulation for the fluid, launch a simulation and import the supplied volume mesh representing the fluid inside the chimney.

1. Launch STAR-CCM+.
2. Start a simulation.
3. Select **File** > **Import** > **Import Volume Mesh...**
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/couplingWithCaeCodes/data`.
5. Select `flue.ccm` and click **Open**.

   STAR-CCM+ provides feedback on the import process in the *Output* window and a mesh region is created under the **Regions** node.
6. Save the simulation as `Flue.sim`.

Selecting Fluid Physics Models

Select physics models to describe the fluid inside the chimney, and specify material properties.

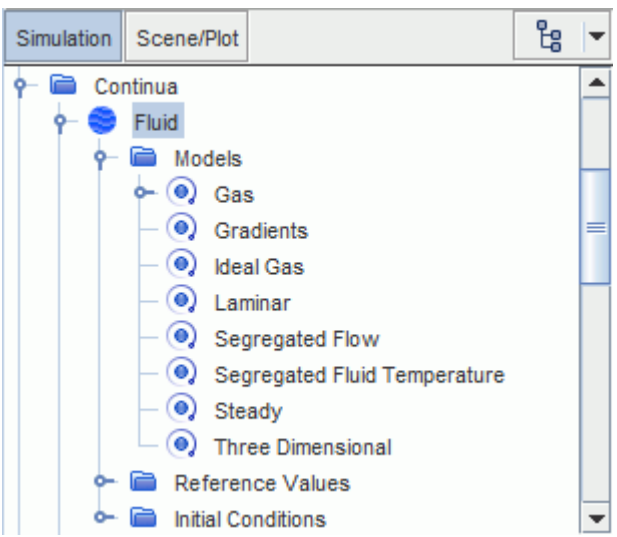
To select the physics models:

1. Rename the **Continua** > **Physics 1** node to `Fluid`.
2. For the physics continuum, **Continua** > **Fluid**, select the following models in order:

Group Box	Model
Enabled Models	**Three Dimensional** (Selected automatically)

Group Box	Model
Time	**Steady**
Material	**Gas**
Flow	**Segregated Flow**
	Gradients (Selected automatically)
Equation of State	**Ideal Gas**
Optional Models	**Segregated Fluid Temperature**
Viscous Regime	**Laminar**

3. Click **Close**.
4. To review the models, open the **Fluid** > **Models** node.



5. Save the simulation.

Specifying Fluid Material Properties

Specify the properties of the exhaust gas as having the same properties as air at 600 K.

To set the gas properties:

1. Expand the **Continua** > **Fluid** > **Models** > **Gas** > **Air** node.
2. Edit the **Material Properties** node and set the following properties:

Node	Property	Setting
Dynamic Viscosity > **Constant**	*Value*	`3.058E-5 Pa-s`
Molecular Weight > **Constant**	*Value*	`28.57 kg/kmol`
Specific Heat > **Constant**	*Value*	`1051.0 J/kg-K`

Node	Property	Setting
Thermal Conductivity > **Constant**	*Value*	`0.0469 W/m-K`

3. Save the simulation.

Defining the Fluid Boundary and Initial Conditions

Specify conditions for the boundaries of the fluid region. Also, set the fluid initial temperature and velocity.

The gas entering the chimney is traveling at 0.2 m/s with a temperature of 600 K. The Reynolds number for this scenario is 1518, thus justifying the laminar flow model. Specify these boundary conditions under the inlet boundary node:

1. Expand the **Regions** > **Flue** > **Boundaries** node.
2. Edit the **Inlet** node and set the following properties:

Node	Property	Setting
Physics Values		
└ **Static Temperature**	*Value*	`600 K`
└ **Velocity Magnitude**	*Value*	`0.2 m/s`

Set the outlet boundary to be at a static temperature of 283 K:

3. Select the **Outlet** > **Physics Values** > **Static Temperature** node and set *Value* to `283 K`.

For the decoupled case, assume that the inner wall is at a constant 380 K:

4. Edit the **HotGas** node and set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Thermal Specification**	*Condition*	**Temperature**
Physics Values		
└ **Static Temperature**	*Value*	`380 K`

Set initial conditions of 283 K and a constant velocity of - 0.2 m/s in the X direction:

5. Edit the **Continua** > **Fluid** > **Initial Conditions** node and set the following properties:

Node	Property	Setting
Static Temperature	*Value*	`283 K`
Velocity	*Value*	`[-0.2, 0.0, 0.0] m/s`

6. Save the simulation.

Preparing a Temperature Scene for the Fluid

Prepare a scalar scene to display the temperature within the fluid region.

To display the temperature within the fluid, first create a plane section through the fluid region:

1. Open *Geometry Scene 1*.
2. Within the **Derived Parts** node, create a plane section with the following properties:

Property	Setting
Input Parts	**Flue**
origin	`[0, 0, 0] m`
normal	`[0, 0, 1] m`
Display	**No Displayer**

3. Click **Create**, then **Close** the *Create Section* dialog.

Create a scalar scene to display the temperature on the plane section:

4. Create a scalar scene.
5. Rename the **Scalar Scene 1** node to `Fluid Temperature`.
6. Edit the **Fluid Temperature** > **Displayers** node and set the following properties:

Node	Property	Setting
Scalar 1	*Contour Style*	`Smooth Filled`
└ **Parts**	*Parts*	**Derived Parts** > **plane section**
└ **Scalar Field**	*Function*	**Temperature**

7. Save the simulation.

Setting Stopping Criteria and Running the Fluid Simulation

Set the stopping criteria, then run the fluid simulation.

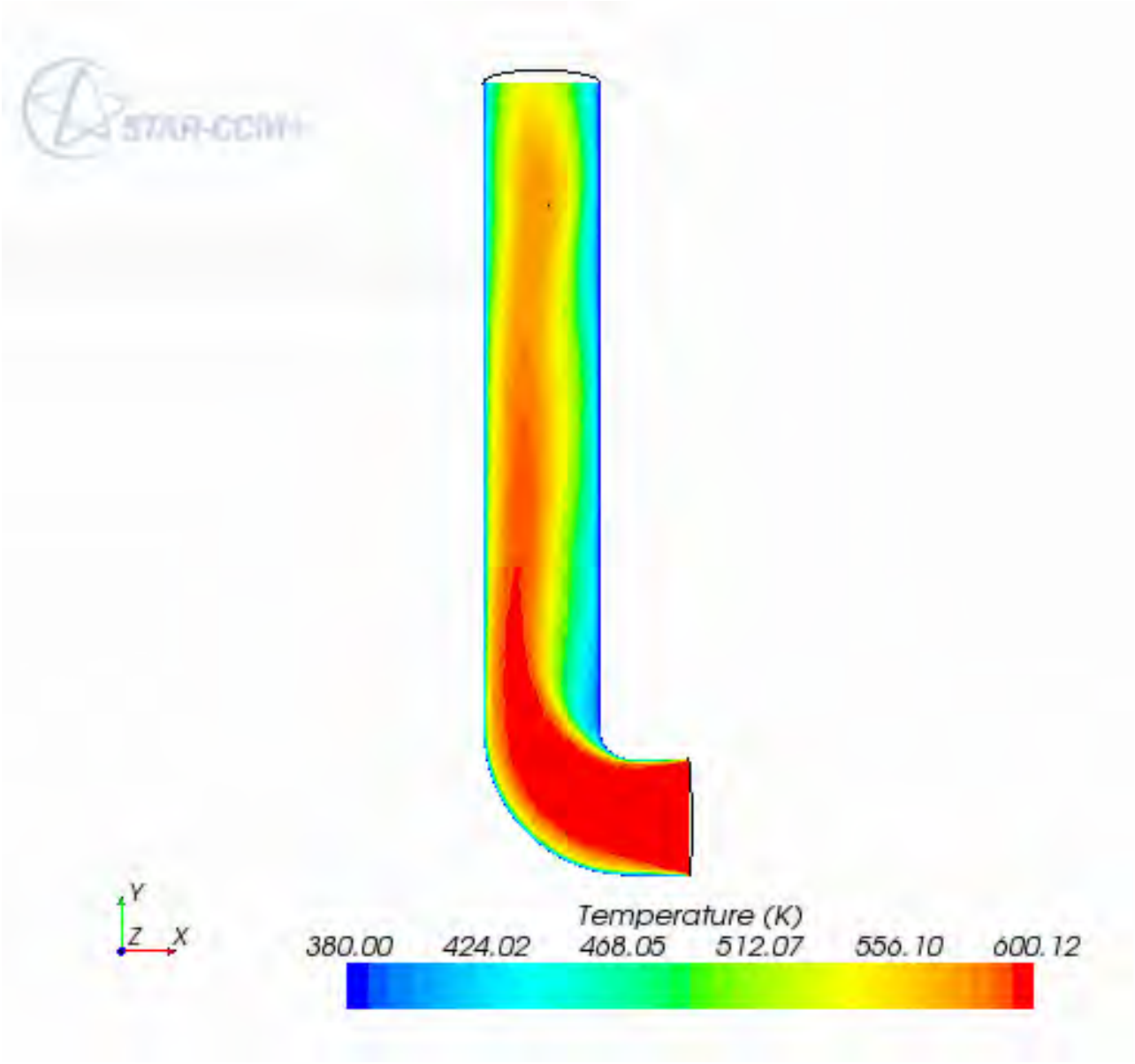
Reduce the maximum number of iterations stopping criteria to 100:

1. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to `100`.
2. Run the simulation.

   The solution converges within 100 iterations.
3. Save the simulation.

Visualizing the Decoupled Fluid Results

The temperature distribution in the gas is not uniform, even for this low speed flow. This means that the assumption of constant wall temperature used in `Chimney.sim` is not valid and an improved solution of the wall temperature distribution can be obtained by coupling the simulations.



These results can be compared against the coupled results later.

1. Save, then close the simulation.

Coupling the Simulations

Specify the co-simulation settings and run the co-simulation.

Contents:

Loading the Simulations
Preparing the Solid Simulation for Co-Simulation
Preparing the Fluid Simulation for Co-Simulation

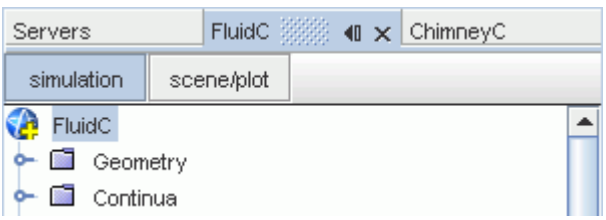
Running the Co-Simulation
Visualizing the Results

Loading the Simulations

Load both the simulations, that you created for the solid and fluid models, in the same instance of STAR-CCM+.

1. Launch STAR-CCM+.
2. Select **File** > **Load Simulation...**
3. In the *Load Simulation* dialog, click **Browse...**
4. In the *Open* dialog, navigate to the directory where you saved the simulation for the fluid model.
5. Select `Flue.sim`, then click **Open**.
6. In the *Load Simulation* dialog, click **OK**.
7. Save the simulation as `FluidC.sim`.
8. Repeat this process for the Chimney.sim file and save it as `ChimneyC.sim`.

   Both cases appear in the *Explorer Pane*:



Note: It is not necessary to have both simulations in the same instance of STAR-CCM+. It is equally valid to have the leading and lagging simulations in separate instances of STAR-CCM+ that can communicate with each other.

Preparing the Solid Simulation for Co-Simulation

In the solid simulation, activate co-simulation models, specify the coupled boundaries, and select the fields that are exchanged with the fluid simulation.

1. In the *Explorer Pane*, make sure that the ChimneyC simulation is highlighted.
2. For the physics continuum, **Continua** > **Solid**, select the following models in order:

Group box	Model
Optional Models	**Co-Simulation**
Co-Simulation Models	**STAR-CCM+**
STAR-CCM+ Coupling Models	**STAR-CCM+ Explicit Coupling**

3. Click **Close**.

   STAR-CCM+ creates additional physics continua that represent the partner simulation.

When you activate a co-simulation model, STAR-CCM+ adds a co-simulation link, **Co-Simulations** > **Link 1**, to the simulation tree. This node allows you to specify global co-simulation settings, such as concurrency and connection settings, and to define co-simulation zones.

A co-simulation zone defines a set of boundaries where fields are exchanged with the partner simulation. In this tutorial, you set up a single co-simulation zone that defines the solid-fluid interface and the thermal fields that are imported from and exported to the fluid at the solid-fluid interface.

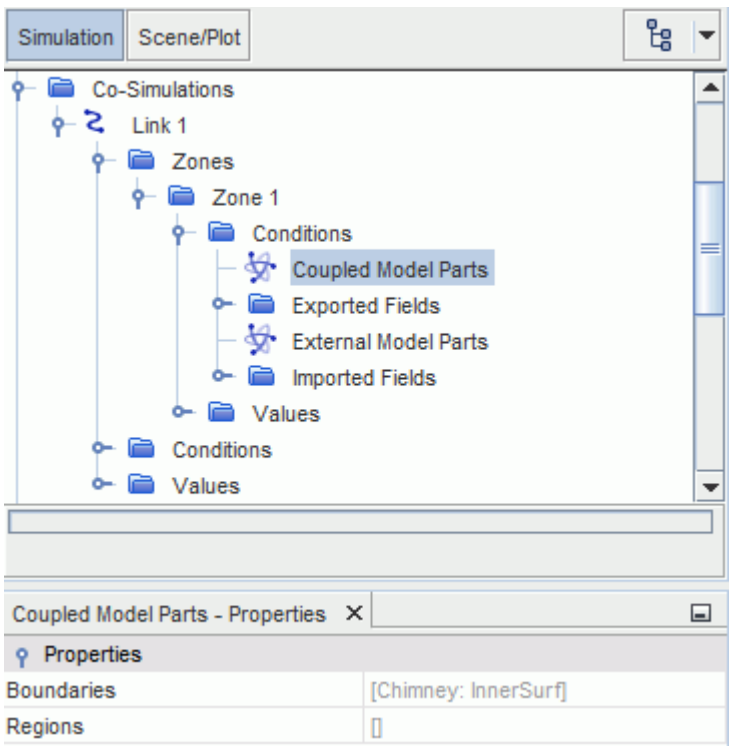
The name of a co-simulation link must match the name of the corresponding link in the partner simulation. In this tutorial, you leave the co-simulation link name as the default, **Link 1**.

Specify the inner surface of the chimney as the coupled boundary:

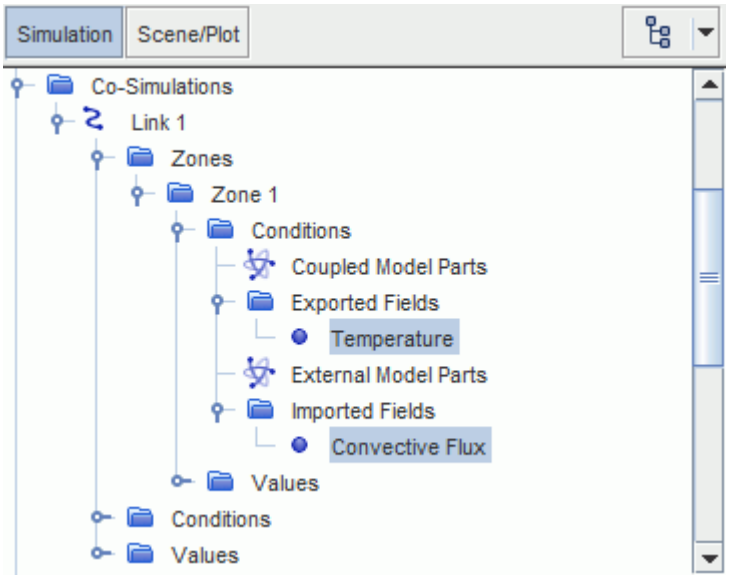
4. Expand the **Regions** > **Chimney** > **Boundaries** > **InnerSurf** > **Physics Conditions** node.
5. Select the **External Code Coupling Specification** node and set *External Code Coupling Specification* to **Link 1: Zone 1**.

   The *Method* for the **Physics Values** > **Ambient Temperature** node is automatically set to **Reference Temperature applied by Co-Simulation**.

   STAR-CCM+ updates the conditions for **Zone 1** by adding **InnerSurf** to the **Zone 1** coupled parts.



In this tutorial, the solid simulation imports convective fluxes from the fluid simulation and exports temperature. The default import/export settings for this simulation are appropriate:



Set the solid simulation to be the lagging simulation:

6. Expand the **Co-Simulations** > **Link 1** > **Conditions** node.

In this co-simulation, the fluid simulation leads while the solid simulation lags. That is, when you run the co-simulation, the fluid simulation runs first whereas the solid simulation waits for initial data from the fluid simulation before running.

7. Select the **Concurrency Option** and set *Method* to *Lag*.

Specify the time interval at which data is exchanged:

8. Select the **Link 1** > **Values** > **Coupling Time Interval** node and set *Coupling Time Interval* to `20 s`.

Increase the run time for this solution:

9. Select the **Stopping Criteria** > **Maximum Physical Time** node and set *Maximum Physical Time* to `1800 s`.

Clear the results from the previous simulation:

10. Select **Solution** > **Clear Solution**.
11. In the *Clear Solution* dialog, click **OK**.
12. Save the simulation.

Preparing the Fluid Simulation for Co-Simulation

In the fluid simulation, activate co-simulation models, specify the coupled boundaries, and select the fields that are exchanged with the solid simulation.

1. In the *Explorer Pane*, make sure that the FluidC simulation is highlighted.
2. For the physics continuum, **Continua** > **Fluid**, select the following models in order:

Group box	Model
Optional Models	**Co-Simulation**
Co-Simulation Models	**STAR-CCM+**
STAR-CCM+ Coupling Models	**STAR-CCM+ Steady Coupling**

3. Click **Close**.

   STAR-CCM+ creates additional continua, to represent the partner simulation.

To specify the coupled boundary:

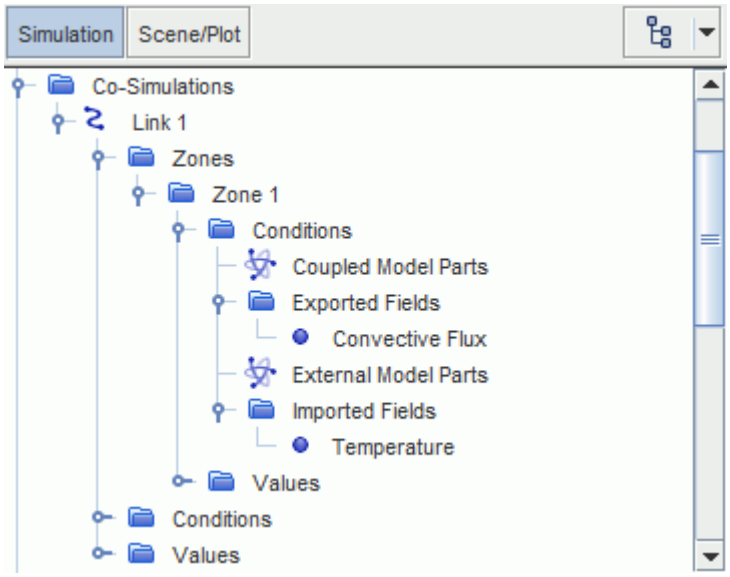
4. Expand the **Regions** > **Flue** > **Boundaries** > **HotGas** > **Physics Conditions** node.
5. Select the **External Code Coupling Specification** node and set *External Code Coupling Specification* to **Link 1: Zone 1**.

   The *Method* for the **Physics Values** > **Static Temperature** node is automatically set to **Temperature applied by Co-Simulation**.

   STAR-CCM+ automatically adds the **HotGas** boundary to the **Zone 1** coupled model parts, and updates the imported/exported field specification.

In this tutorial, the fluid simulation exports convective fluxes to the solid simulation and imports temperature:

6. Expand the **Co-Simulations** > **Link 1** > **Zones** node.
7. Right-click the **Zone 1** > **Conditions** > **Exported Fields** > **Pressure** node, select **Delete**, and confirm the selection.



Set the number of iterations between exchanges:

8. Select the **Link 1** > **Conditions** > **Field Exchange Controls** node and set *Iterations Per Exchange* to `100`.

The default setting for the **Concurrency Option** is appropriate, as the fluid simulation is set to be the leading simulation.

To connect to the partner simulation (that you already launched):

9. Select the **Co-Simulations** > **Link 1** > **Values** > **STAR-CCM+ Partner Host:Port** node.
10. In the *Properties* window, click ... (**Custom Editor**).
11. In the *STAR-CCM+ Partner Host:Port* dialog, select the **ChimneyC** server and click **OK**.

Increase the number of iterations for this solution:

12. Select the **Stopping Criteria** > **Maximum Steps** node and set *Maximum Steps* to `9000`.

    This is the maximum physical time for Chimney.sim (1800 s), divided by the time step (20 s), times the number of iterations per update (100).
13. Clear the results from the previous simulation.
14. Save the simulation.

Running the Co-Simulation

The simulations are fully set up and the co-simulation is ready to be run.

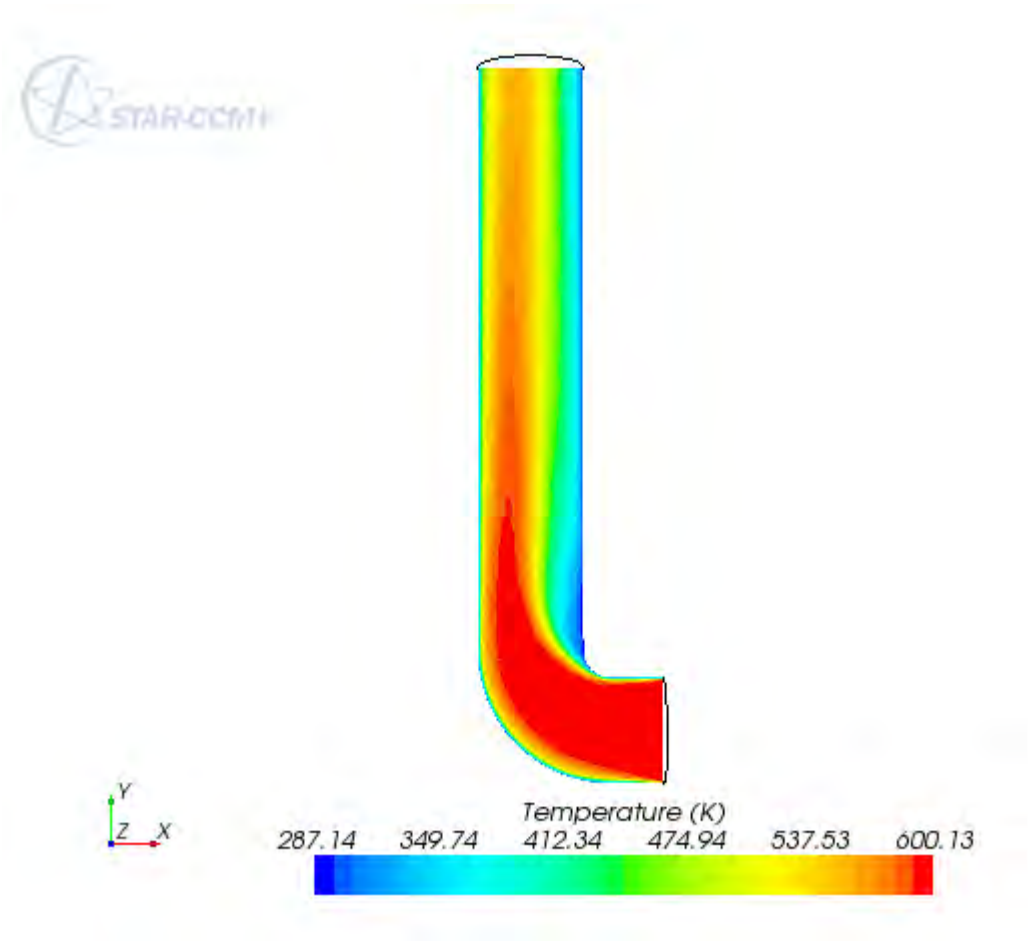
1. In FluidC.sim, click (**Run**).

   The fluid simulation connects to the solid simulation. The leading simulation runs first, whereas the lagging simulation waits for initial data. The simulations take turns solving their part of the problem and updating the boundary conditions at the coupled boundary. When its partner simulation is solving, the progress bar on the waiting partner displays the message *Waiting for partner code to finish*.
2. Open the *Residuals* plots for each case to check convergence behavior.
3. When the co-simulation is complete, save each simulation.

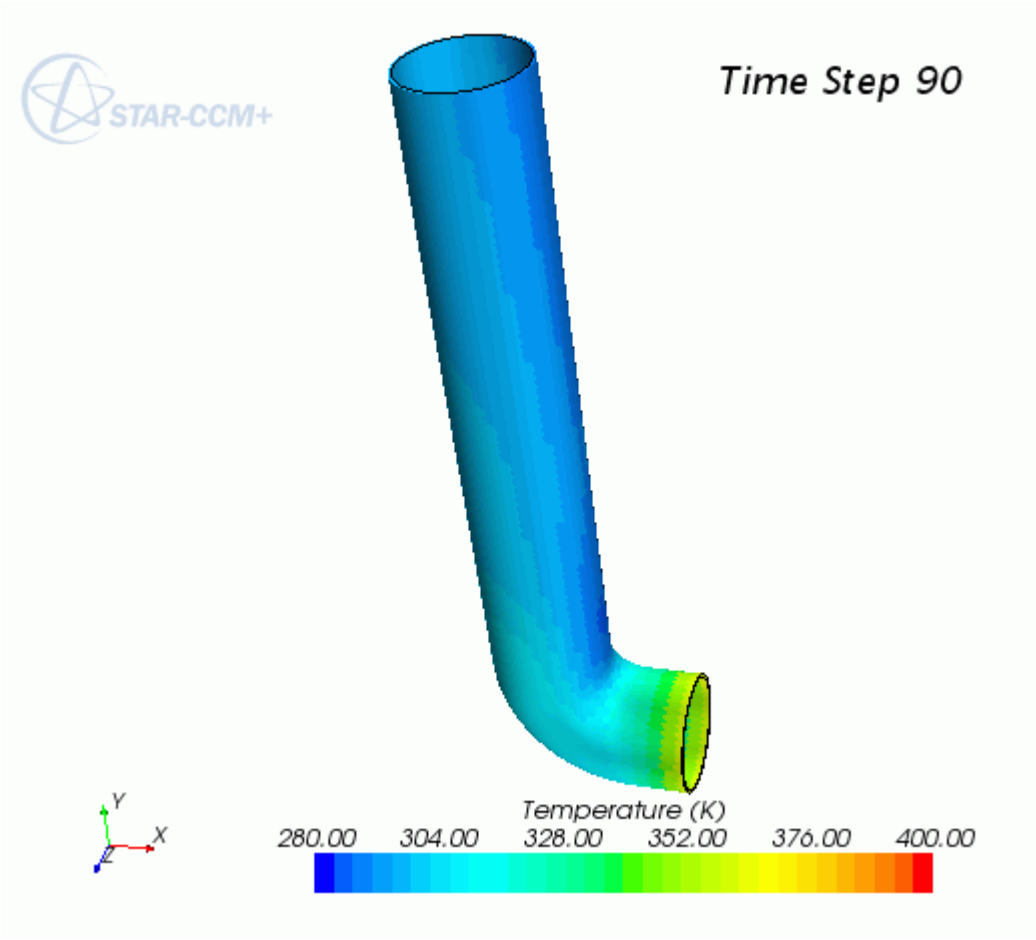
Visualizing the Results

Visualize the temperature scenes in the solid and the fluid simulation.

The final results for the FluidC case appear as shown below:



The final results for the ChimneyC case appear as shown below:

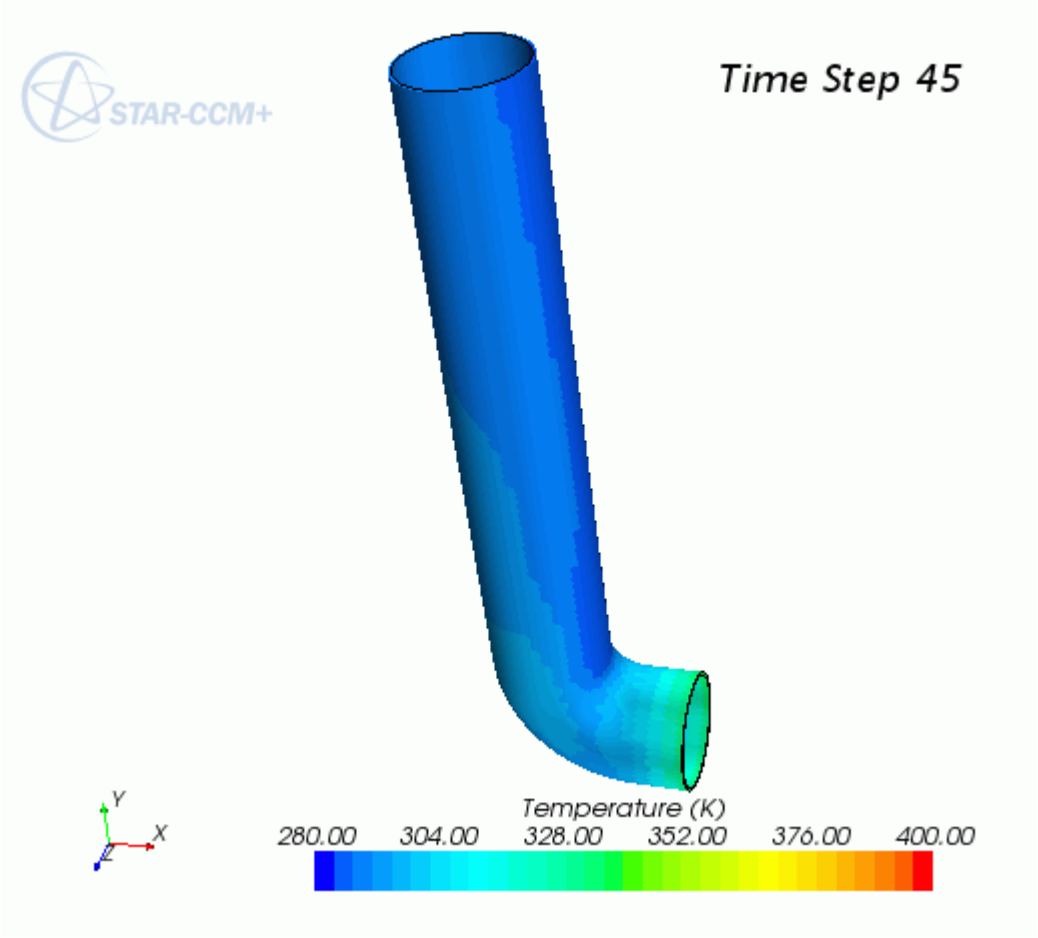


The coupling has improved the quality of the solution of both simulations. However, of the two, the temperature distribution in the metal chimney has changed the most. The chimney shows hot spots near the inlet and on the wall furthest away from the apex of the bend.

Display the ChimneyC solution at an interim time-step:

1. Right-click the **Solution Histories** > **ChimneyHistory** node and select **Create Recorded Solution View**.
2. In the dialog that appears, click **No**.
3. Click the *ChimneyC-Wall Temperature* tab in the *Graphics* window.
4. Drag and drop the **Solution Views** > **ChimneyHistory** node onto a blank area in the scene window.
5. Select the **Solution Views** > **ChimneyHistory** node and set *Time Step* to `45`.

   The scene updates, as shown below:



The difference between the temperature distribution at Time Step 45 and Time Step 90 shows that the metal is heating gradually as a result of the hot exhaust gases.

6. Save the simulation.

Summary

This tutorial demonstrated the process for modeling conjugate heat transfer using STAR-CCM+ to STAR-CCM+ thermal co-simulation.

This tutorial has demonstrated the following features:

- Setting up a STAR-CCM+ to STAR-CCM+ co-simulation.
- Creating the independent solid simulation and checking for convergence.
- Creating the independent fluid simulation and checking for convergence.
- Coupling two STAR-CCM+ simulations.
- Viewing intermediate results.

LMS Amesim Co-Simulation: 1D Coupling

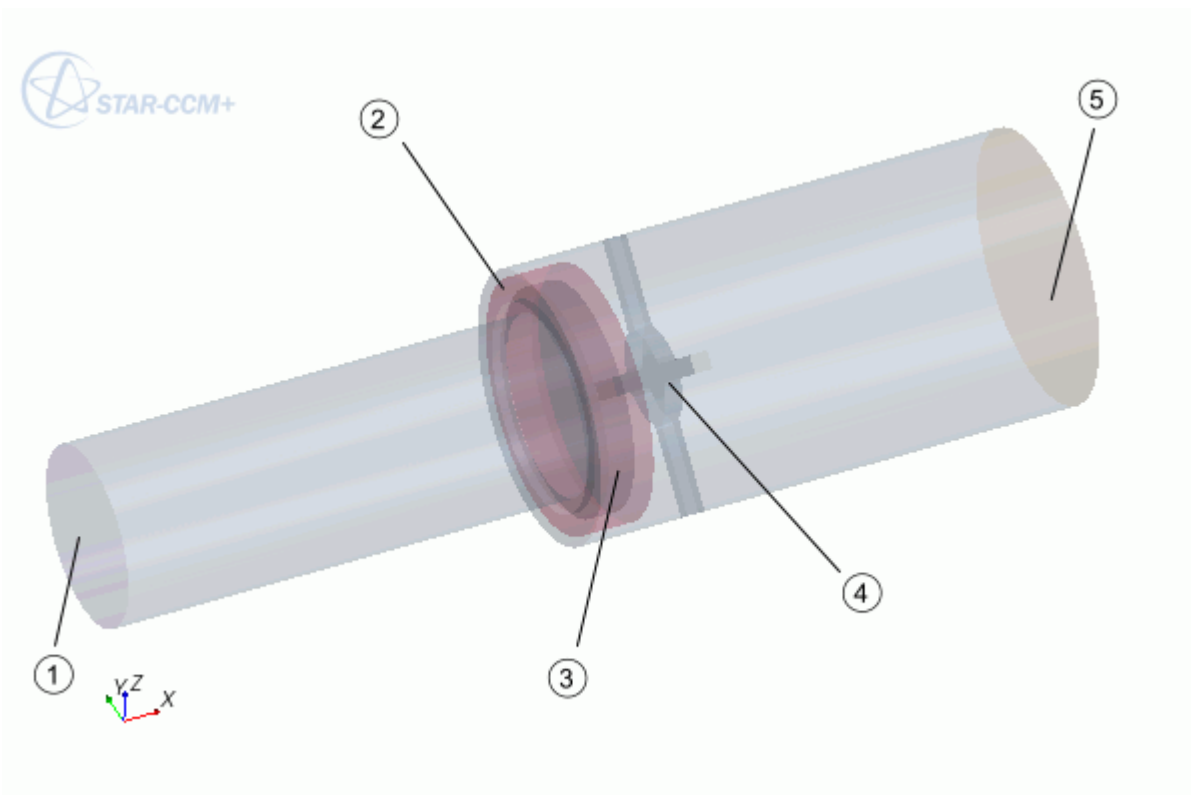
This tutorial demonstrates a typical workflow when running a coupled simulation with LMS Imagine.Lab Amesim.

With STAR-CCM+ to LMS Amesim co-simulation, you can obtain more accurate data on the amount of force acting on the plate to feed back into your LMS Amesim model. The data is transferred between the two codes every 5.0E-4 seconds. The total simulation time is 0.05 seconds.

STAR-CCM+ solves the fluid flow through the check valve during a transient simulation. The LMS Amesim model simulates a check valve, where a linear spring controls the plate displacement. The following list summarizes the characteristics of the check valve:

- Plate Diameter: 50 mm
- Plate Mass: 0.2 kg
- Plate Higher Displacement Limit: 10.21 mm. This property limits the amount that the plate can move in the specified direction, and simulates the distance to spring compression.
- Spring Rate: 2000 N/m

The following image shows the three-dimensional geometry and highlights the plate valve components:



Index	Description
1	**Inlet**

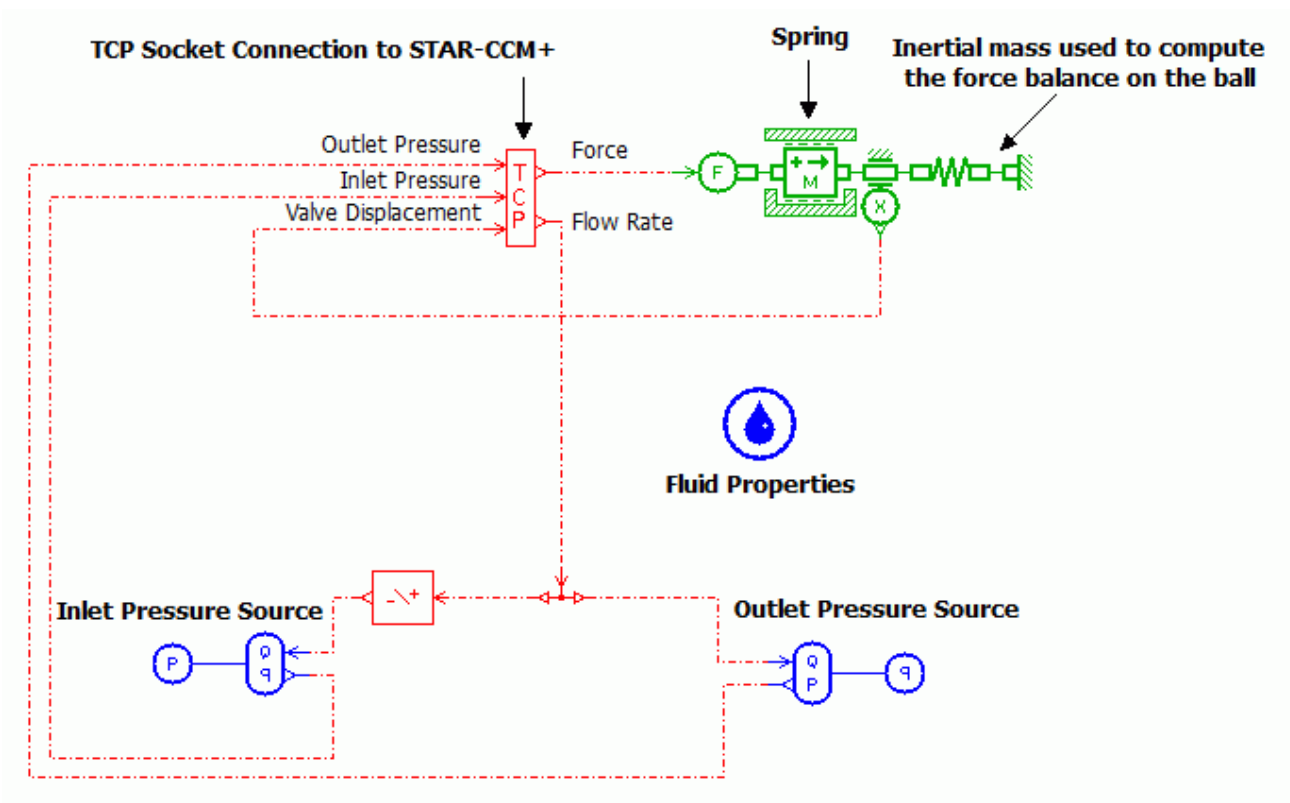
Index	Description
2	**Overset Region**
3	**Plate**
4	**Plate Support**
5	**Outlet**

For this tutorial, you are provided with a simulation file that contains predefined objects:

- mesh operations
- physics models not related to LMS Amesim
- initial conditions (in the physics continuum and at the inlet boundary)
- regions and boundaries

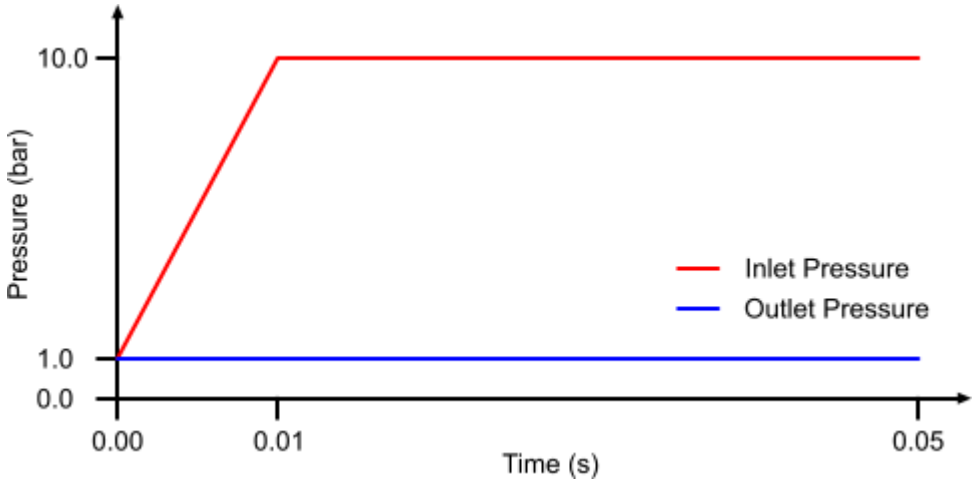
The LMS Amesim Model

In a co-simulation with LMS Amesim, you can exchange any scalar field between STAR-CCM+ and LMS Amesim. The LMS Amesim model for this tutorial is shown in the following diagram.



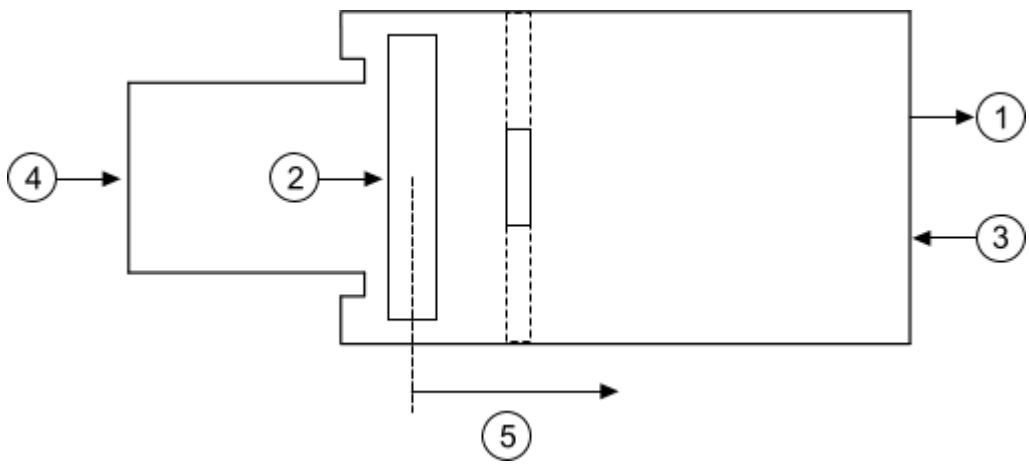
In this tutorial, STAR-CCM+ sends the mass flow data to LMS Amesim. LMS Amesim controls the pressure drop across the check valve. In LMS Amesim, the inlet pressure is set to climb from 1 bar to 10 bar for the first 0.01 seconds, and then is set to maintain 10 bar for 0.04 seconds. LMS Amesim maintains the outlet pressure at 1 bar for the entire simulation. LMS Amesim sends the calculated pressure data to STAR-CCM+, where it is

applied to the corresponding boundaries (the inlet and outlet). The following diagram illustrates these pressures:



The force that the plate experiences due to the flow is sent from STAR-CCM+ to LMS Amesim. LMS Amesim calculates the displacement of the plate using the force input signal (along with the specified plate mass and spring force), and sends the displacement output signal back to STAR-CCM+. STAR-CCM+ applies the displacement data to the overset region using a user-defined vertex motion. The new position of the plate affects the flow, and the new data is sent back to LMS Amesim. This cycle repeats for each time step.

The following diagram illustrates how data is exchanged between STAR-CCM+ and LMS Amesim:



Index	Description	Data Transfer
1	Outlet mass flow	STAR-CCM+ Field Function to LMS Amesim Port 1
2	Force on the plate	STAR-CCM+ Field Function to LMS Amesim Port 2
3	Outlet pressure	LMS Amesim Port 3 to STAR-CCM + Outlet Boundary

Index	Description	Data Transfer
4	Inlet pressure	LMS Amesim Port 4 to STAR-CCM + Inlet Boundary
5	Plate displacement	LMS Amesim Port 5 to STAR-CCM + Morpher

Contents:

Prerequisites
Loading the Initial Simulation
Generating the Volume Mesh
Selecting Physics Models
Specifying the Exported Fields
Specifying the Imported Fields
Preparing Analysis Objects
Setting the Time-Step and Stopping Criteria
Coupling to LMS Amesim
Running the Co-Simulation
Summary

Prerequisites

The instructions in the LMS Amesim Coupling tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Techniques	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

This tutorial describes the procedure using LMS Amesim v15.1. For a list of all supported versions of LMS Amesim, refer to the section *External Packages Support* of the STAR-CCM+ Release Notes.

Loading the Initial Simulation

For this tutorial, you are provided with a simulation file that contains pre-defined objects.

Load the initial simulation:

1. Launch STAR-CCM+.
2. Select **File** > **Load Simulation...**
3. In the *Load Simulation* dialog, click **Browse...**

4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/couplingWithCaeCodes/data`.
5. Select `checkValve-Amesim.sim`, then click **Open**.
6. In the *Load Simulation* dialog, click **OK**.

Generating the Volume Mesh

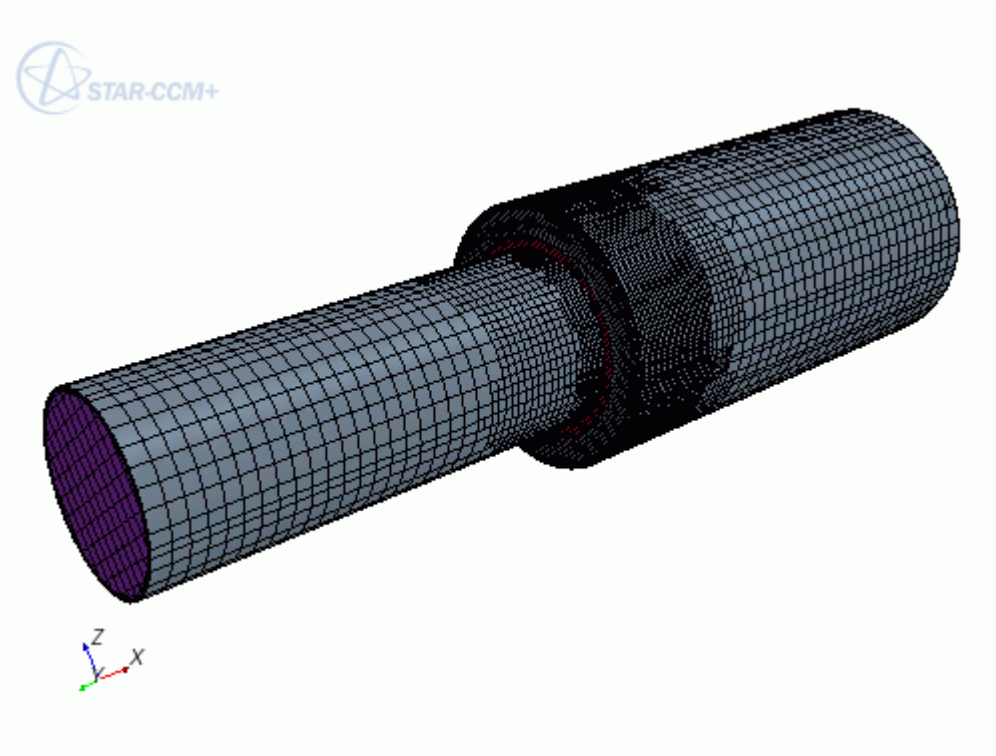
Generate the volume mesh using the predefined mesh settings.

To generate the mesh:

1. Click (**Generate Volume Mesh**).
   The surface mesh and volume mesh are generated with the predefined settings.

To visualize the mesh:

2. Open *Mesh Scene 1*.
3. Right-click in a blank area of the scene and select **Apply Representation** > **Volume Mesh**.
   The volume mesh is shown in the following screenshot.



4. Save the simulation.

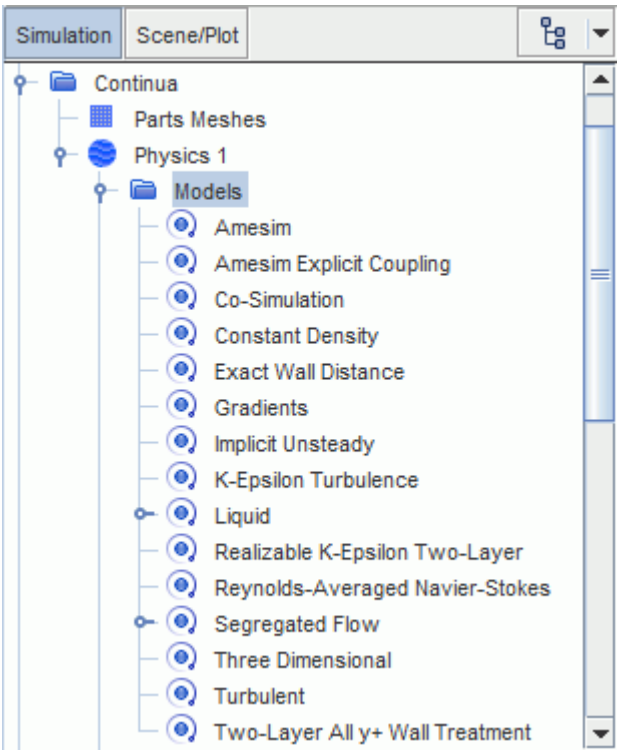
Selecting Physics Models

In this tutorial, you activate the physics models that enable coupling between STAR-CCM+ and LMS Amesim. All other physics models that are not related to LMS Amesim are preselected in the simulation.

1. For the physics continuum, **Continua** > **Physics 1**, select the following models in order:

Group Box	Model
Optional Models	**Co-Simulation**
Co-Simulation Models	**Amesim**
Amesim Coupling Models	**Amesim Explicit Coupling** (automatically selected)

2. Click **Close**.
3. To review the models, open the **Physics 1** > **Models** node.



The initial conditions for this simulation are predefined.

4. Save the simulation.

Specifying the Exported Fields

Specify the data that STAR-CCM+ exports to LMS Amesim. As LMS Amesim is a 1D code, STAR-CCM+ must send single-value fields to the LMS Amesim ports.

In this tutorial, you create reports to calculate the mass flow at the pipe outlet and the force acting on the plate, and export them to LMS Amesim ports 1 and 2, respectively.

To create the reports:

1. Right-click the **Reports** node and select **New Report** > **Mass Flow**.
2. Select the **Mass Flow 1** node and set *Parts* to **Regions** > **Pipe** > **Outlet**.
3. In a similar way, create a report of type **Force** and set its *Parts* to **Regions** > **Plate** > **Plate**.

Add the reports to the exported fields:

4. Expand the **Co-Simulations** > **Link 1** node.
5. Right-click the **Conditions** > **Exported Fields** node and select **Add Auxiliary Field** > **Scalar**.
6. Rename the **Exported Fields** > **Scalar Auxiliary Field 1** node to `Mass Flow` and set *Scalar Field Function* to **Reports** > **Report: Mass Flow 1**.
7. In a similar way, add another scalar auxiliary field named `Force` and set its scalar field function to **Reports** > **Report: Force 1**.

Make sure that each exported field is associated with the correct LMS Amesim port:

8. Expand the **Link 1** > **Conditions** > **Amesim Port Associations** node and set the following properties:

Node	Property	Setting
Export: Mass Flow	*Associated ID*	`1`
Export: Force	*Associated ID*	`2`

9. Save the simulation.

Specifying the Imported Fields

Specify the fields that STAR-CCM+ imports from LMS Amesim. Use the imported fields to define the boundary conditions and move the valve mesh.

In this tutorial, STAR-CCM+ imports the pressure at the pipe outlet, the total pressure at the pipe inlet, and the valve displacement from LMS Amesim ports 3, 4, and 5, respectively.

To specify the imported fields:

1. Expand the **Co-Simulations** > **Link 1** node.
2. Right-click the **Conditions** > **Imported Fields** node and select **Add Auxiliary Field** > **Scalar**.
3. Rename the **Imported Fields** > **Scalar Auxiliary Field 1** node to `Outlet Pressure`.
4. In a similar way, add two other scalar auxiliary fields named `Inlet Total Pressure` and `Valve Displacement`.

STAR-CCM+ automatically creates and assigns a field function to each imported field, based on the associated LMS Amesim port. Make sure that each imported field is associated with the correct LMS Amesim port:

5. Expand the **Link 1** > **Conditions** > **Amesim Port Associations** node and set the following properties:

Node	Property	Setting
Import: Outlet Pressure	*Associated ID*	3
Import: Inlet Total Pressure	*Associated ID*	4
Import: Valve Displacement	*Associated ID*	5

STAR-CCM+ automatically sets the *Scalar Field Function* property of the **Inlet Pressure**, **Outlet Pressure**, and **Valve Displacement** nodes to **Amesim Port 4**, **Amesim Port 3**, and **Amesim Port 5**, respectively.

Use the fields imported from LMS Amesim to set boundary conditions in STAR-CCM+. Specifically, define the inlet total pressure and the outlet pressure as follows:

6. Expand the **Regions** > **Pipe** > **Boundaries** node and set the following properties:

Boundary	Node	Property	Setting
Inlet	**Physics Values** > **Total Pressure**	*Method*	**Field Function**
		Scalar Function	**Amesim Port 4**
Outlet	**Physics Values** > **Pressure**	*Method*	**Field Function**
		Scalar Function	**Amesim Port 3**

Use the valve displacement imported from LMS Amesim port 5 to move the overset region, **Plate**, during the simulation:

7. Right-click the **Tools** > **Motions** node and select **New** > **User Defined Vertex**.
8. Select the **Regions** > **Plate** > **Physics Values** > **Motion Specification** node and set *Motion* to **User Defined Vertex**.
9. Edit the **Regions** > **Plate** node and set the following properties:

Node	Property	Setting
Physics Conditions > **User Defined Vertex Option**	*Method*	**Displacement**
Physics Values > **Linear Displacement**	*Total Displacement*	**Activated**
	Method	**Composite**
└ **Composite**		
└ **X Component**	*Method*	**Field Function**
	Scalar Function	**Amesim Port 5**

Note: The *Total Displacement* option specifies that the displacement data sent by LMS Amesim corresponds to the total amount the plate has moved. If this option is deactivated, STAR-CCM+ assumes that the displacement value corresponds to the amount the plate has moved in the current time-step.

10. Save the simulation.

Preparing Analysis Objects

Prepare analysis objects to monitor the solution while the co-simulation is running. In this tutorial, you monitor the plate displacement in response to the increased inlet pressure and the flow within the plate valve.

To plot the plate displacement:

1. Right-click the **Reports** node and select **New Report** > **Expression**.
2. Rename the **Expression 1** node to `Plate Displacement`.
3. Select the **Reports** > **Plate Displacement** node and set *Definition* to `${AmesimPort5}`.
4. Right-click the **Plate Displacement** node and select **Create Monitor and Plot from Report**.

Configure the plot to display the appropriate values:

5. Open the **Plate Displacement Monitor Plot**.
6. Expand the **Plot** > **Plate Displacement Monitor Plot** > **Axes** node and set the following properties:

Node	Property	Setting
Bottom Axis	*Maximum*	`0.05`
Left Axis	*Maximum*	`0.015`

Create a vector scene that shows the flow on a center plane within the plate valve. Use line integral convolution to visualize the flow velocity:

7. Create a new empty scene.
8. Within the **Derived Parts** node, create a new plane section with the following properties:

Property	Setting
origin	`[0, 0, 0] m`
normal	`[0, 0, 1] m`
Display	**New Vector Displayer**

9. Select the **Scenes** > **Scene 1** > **Displayers** > **Section Vector 1** node and set *Display Mode* to **Line Integral Convolution**.
10. Save the simulation.

Setting the Time-Step and Stopping Criteria

Match the time-step in STAR-CCM+ with the corresponding value in LMS Amesim.

In this tutorial, the time-step is 5E-4 seconds. This setting uses all the data that LMS Amesim sends to STAR-CCM+ to analyze the three-dimensional flow characteristics.

To set the time-step:

1. Select the **Solvers** > **Implicit Unsteady** node and set *Time-Step* to `5.0E-4 s`.

To set the stopping criteria:

2. Expand the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Inner Iterations	*Maximum Inner Iterations*	`5`
Maximum Physical Time	*Maximum Physical Time*	`0.05 s`
Maximum Steps	*Enabled*	**Deactivated**

The simulation is set up to run for a total of 0.05 seconds, which is enough time for the plate to settle after the increase in inlet pressure.

Note: When running a STAR-CCM+ to LMS Amesim co-simulation, the total time in LMS Amesim must be set to a higher value than the total time in STAR-CCM+. STAR-CCM+ keeps LMS Amesim data for two time-steps and interpolates between them.

3. Save the simulation.

Coupling to LMS Amesim

Start the LMS Amesim solver and connect to LMS Amesim.

Before STAR-CCM+ can communicate with LMS Amesim, you start the LMS Amesim solver in server mode.

When an LMS Amesim model contains a `dynamic_cosim_socket`, the LMS Amesim solver waits until the STAR-CCM+ solver runs before it begins solving the model.

To start the LMS Amesim solver:

1. Navigate to `[INSTALL_DIR]/doc/startutorialsdata/couplingWithCaeCodes/data/Amesim15.1`.
2. Copy the `Amesim_checkValve.ame` file to your current working directory.
3. Launch LMS Amesim and load the `Amesim_checkValve.ame` file that is located in your working directory.
4. Click to switch to simulation mode.
5. In the *System Compilation* dialog, click **Close**.
6. Click to start the LMS Amesim solver in server mode.

In STAR-CCM+, specify the host and port on which the LMS Amesim solver is running:

7. Select the **Link 1** > **Values** > **Connection Settings** node and set the following properties:

Property	Setting
Port Number	`60000`
Hostname	name or IP address of the machine where LMS Amesim is running.

To calculate an initial value for the force on the plate, instruct STAR-CCM+ to run two time steps before sending data to LMS Amesim:

8. Select the **Values** > **Couple From Step** node and set *Couple From Step* to `1`.

To couple to LMS Amesim:

9. Right-click the **Co-Simulations** > **Link 1** node and select **Connect**.

   The communication between LMS Amesim and STAR-CCM+ is established, and the *Output* window displays the number of input and output ports.

10. Save the simulation.

Running the Co-Simulation

Run the co-simulation and visualize the solution.

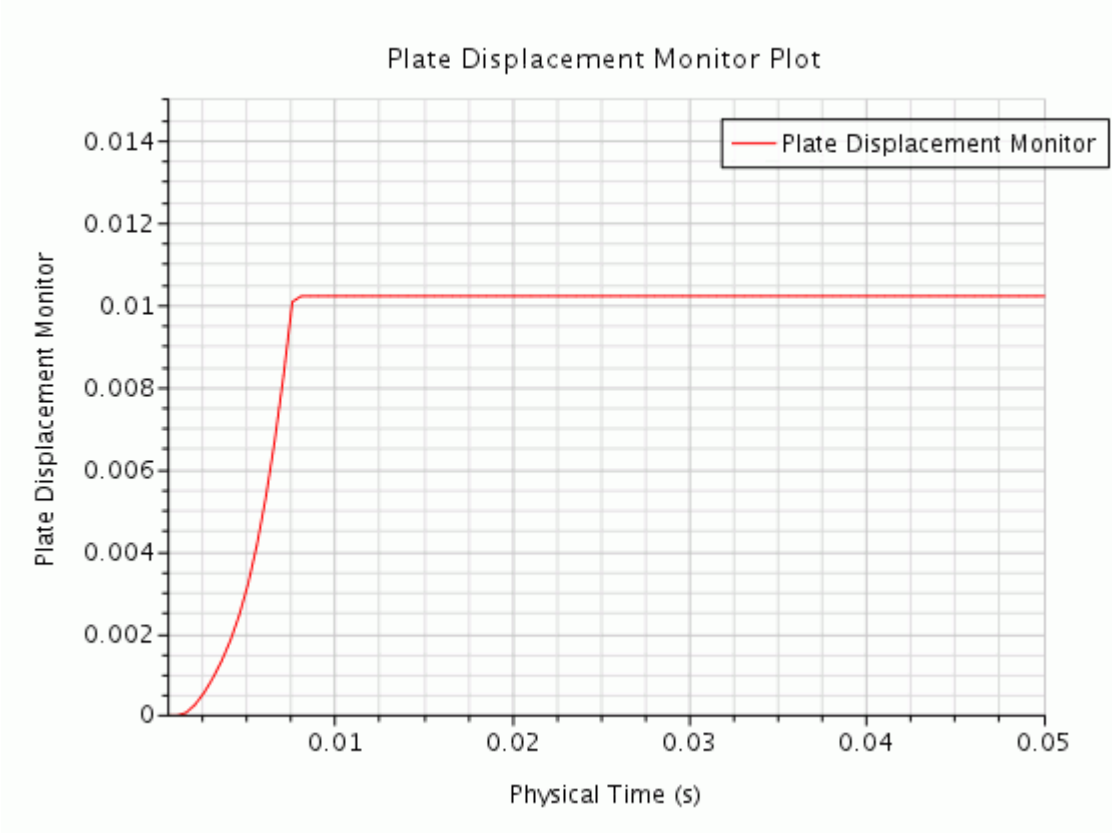
To run the co-simulation:

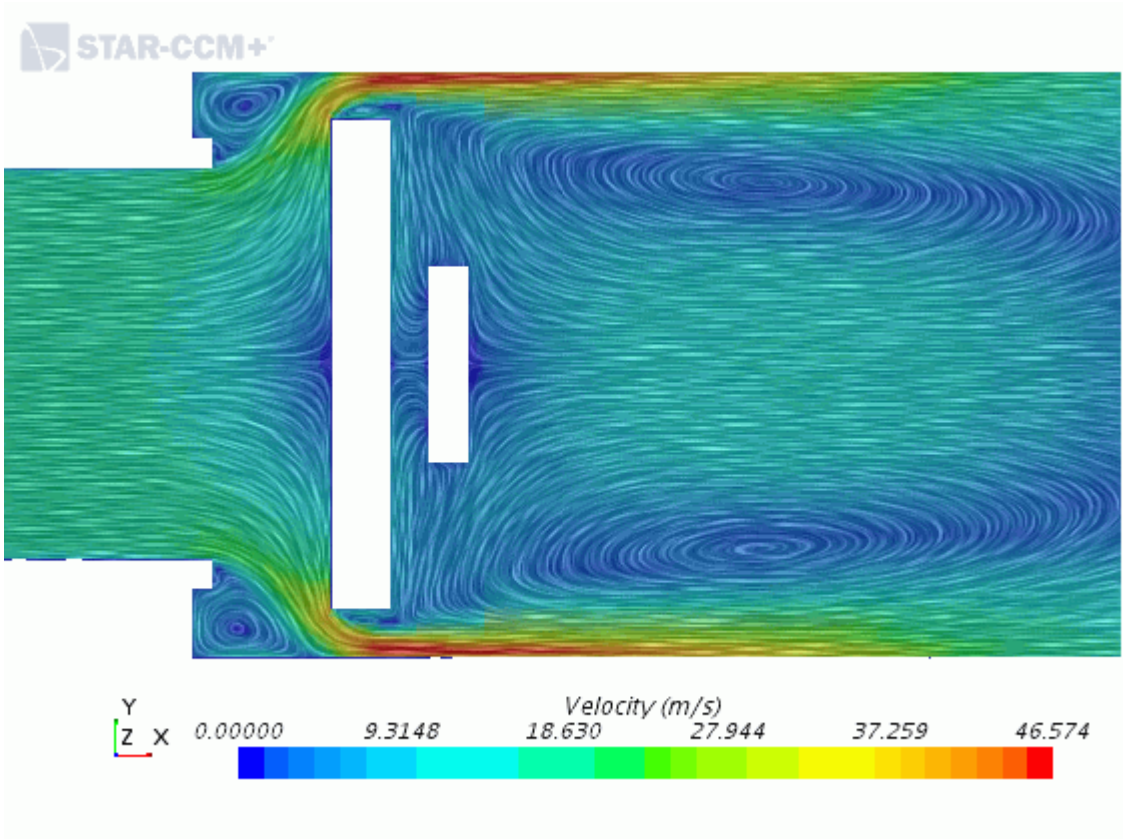
1. Click (**Run**) in the STAR-CCM+ simulation.

   Solution progress can be monitored in the *Output* window.

2. When the solver is complete, save the simulation.

   At the end of the simulation, the **Plate Displacement Monitor Plot** and **Scalar Scene 1** appear as follows.





Summary

This tutorial demonstrated a typical workflow when running a coupled simulation with LMS Amesim.

This tutorial has introduced the following STAR-CCM+ features:

- Selecting physics models for LMS Amesim co-simulation.
- Specifying the exchanged fields.
- Setting boundary conditions using the fields imported from LMS Amesim.
- Setting up a User Defined Vertex motion using the imported displacement.
- Coupling to LMS Amesim and running the co-simulation.

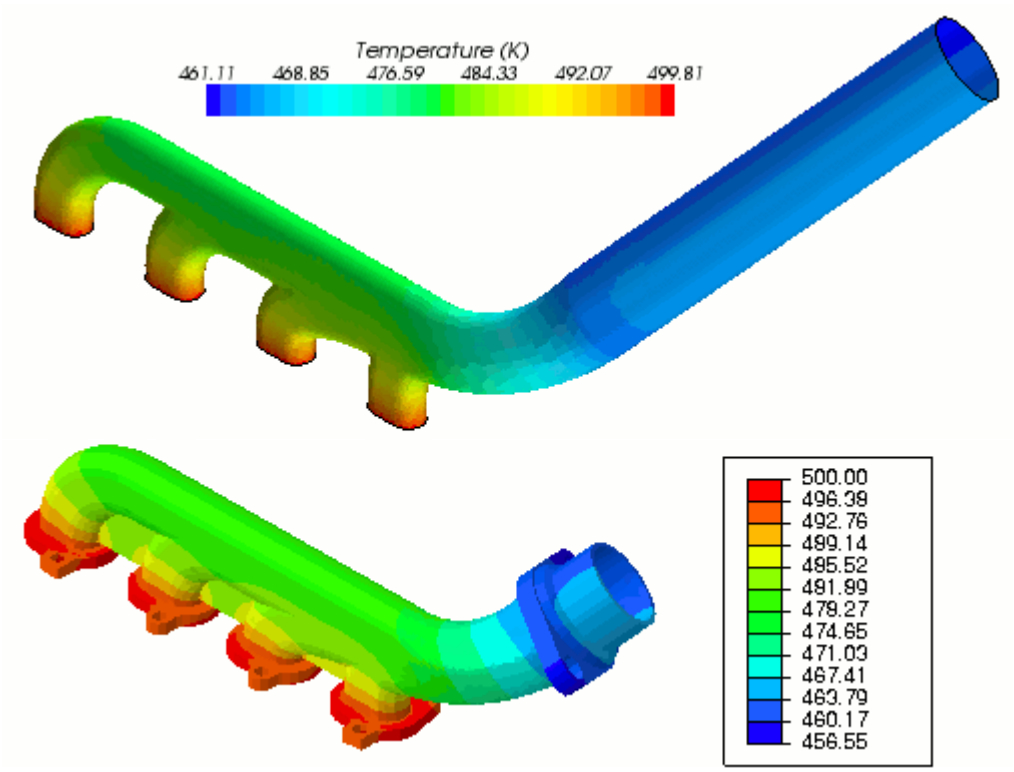
Abaqus File-Based Coupling: Exhaust Manifold

The purpose of this tutorial is to demonstrate the typical workflow in running a thermal fluid-structure interaction case using the file-based coupling method with Abaqus.

In the file-based coupling method, the thermal conduction in the solid structure is computed using Abaqus and the two separate simulations in Abaqus and STAR-CCM+ are coupled by exchanging data for the fluid-structure interface using files on the hard drive.

STAR-CCM+ passes local heat transfer coefficients and reference temperatures to Abaqus and Abaqus supplies the wall temperature to STAR-CCM+. In Abaqus terminology, the heat transfer coefficient is referred to as film coefficient and the reference temperature is referred to as sink temperature. The goal is to exchange data between the two codes for enough times so that you reach a converged solution. This can be accomplished using the GUI by manually performing the import-map-export operations, and launching Abaqus runs after each exchange of data. It is also possible to carry out the STAR-CCM+ steps automatically by using Java macros, which is demonstrated in this tutorial.

The case is an exhaust manifold subject to 10 m/s inlet gases at 500K. The objective is to find the steady-state solution for the flow and temperature in the exhaust gas, and the temperature in the solid manifold. Abaqus leads the process by providing the initial steady-state solution for the wall-temperature on the interior surface of the manifold, based on estimations for the thermal conditions in the manifold. From the initial Abaqus solution, STAR-CCM+ extracts the wall temperatures, map them onto the fluid model, and use them to set a boundary condition for the energy equation in the fluid region. From the STAR-CCM+ solution, the local heat transfer coefficients and reference temperatures are mapped and exported to Abaqus, and in the next iteration Abaqus uses this data to obtain a set of wall temperature data. These data exchanges are repeated until the wall temperatures, heat transfer coefficients, and reference temperatures have converged.



This type of coupling algorithm is referred to as a *loose coupling algorithm*, that is, where files on the hard disk are used to exchange data. STAR-CCM+ supports this method of coupling by allowing users to generate output files in a number of different formats that can be read by third-party software. In this tutorial, the heat transfer coefficients and reference temperatures are exported to a text file that can be read into Abaqus. This text file uses the keyword `*FILM` to specify the type of field data within the file. More information on the `*FILM` keyword can be found in the Abaqus Keywords Reference Manual.

An input file for the Abaqus analysis has been provided for this tutorial, which contains the manifold geometry and the settings required for the Abaqus job. A surface mesh for the fluid in the manifold has also been prepared for the tutorial; this is provided in `.dbs` format.

Note: You could also use the conjugate heat transfer capabilities within STAR-CCM+ to solve the solid manifold temperatures. The setup for such a case is demonstrated in the Thermal Strain: Exhaust Manifold tutorial.

Contents:

Prerequisites
Running the Initial Abaqus Job
Preparing the STAR-CCM+ Model
Importing the Abaqus Model
Importing Temperature Data
Mapping Imported Temperature Data
Preparing Scalar Scenes
Running the Simulation Using Imported Data

Reaching a Converged Solution
Summary

Prerequisites

The instructions in the Abaqus File-Based Coupling tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Techniques	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Running the Initial Abaqus Job

Run the Abaqus job to obtain the initial solution for the solid manifold.

First, copy the Abaqus input file required for this tutorial from the STAR-CCM+ installation directory into a separate directory. When the Abaqus job is run, a number of files are generated by Abaqus into this directory:

1. Create a directory for the tutorial called `fileBased`.
2. Navigate to `[INSTALL_DIR]/doc/startutorialsdata/couplingWithCaeCodes/data/fileBased`.
3. Copy the file `exhaust.inp` to the `fileBased` directory.

The Abaqus input file contains model and history data for the exhaust manifold. Run the Abaqus job first to obtain the initial solution for the solid manifold, based on the initial conditions and thermal loads specified in the input file. This allows you to use the initial Abaqus solution as a boundary condition in the fluid domain later on:

4. In a shell/command prompt, run the initial Abaqus job by typing:

   ```
   % abaqus interactive job=exhaust
   ```

   where `abaqus` is the command to run Abaqus.

   Monitor the output in the *shell/command prompt* window. When the job is finished, the line `Abaqus JOB exhaust COMPLETED` is displayed.

During the Abaqus analysis, a number of files are created, including: `exhaust.com`, `exhaust.dat`, `exhaust.msg`, `exhaust.odb`, `exhaust.sim`, and `exhaust.sta`. The output database file, `exhaust.odb`, contains the results of the analysis. This file can also be used in Abaqus/CAE to visualize the results of the job.

Preparing the STAR-CCM+ Model

Prepare the STAR-CCM+ simulation for the fluid inside the manifold.

For this tutorial, you are provided with a simulation file that contains pre-defined objects. To load the initial simulation:

1. Navigate to `[INSTALL_DIR]/doc/startutorialsdata/couplingWithCaeCodes/data` and copy the starting simulation file, `manifold-gas_start.sim`, into the `/fileBased` directory created for the tutorial.
2. Launch STAR-CCM+.
3. Select **File** > **Load Simulation...**
4. In the *Load Simulation* dialog, click **Browse...**
5. Select `manifold-gas_start.sim` then click **Open**.
6. In the *Load Simulation* dialog, click **OK**.
7. Save the simulation as `manifold-gas.sim`.

Contents:

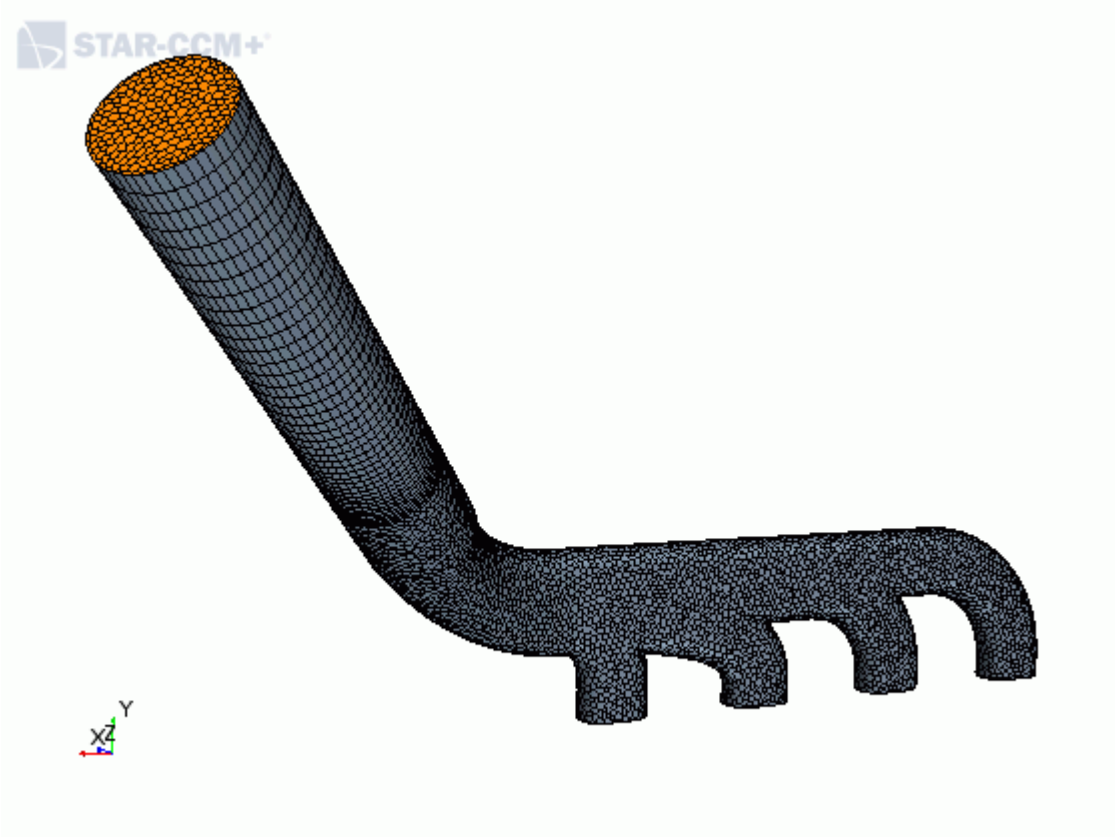
Generating the Volume Mesh
Selecting Physics Models
Defining Boundary Conditions
Setting Up a Monitor
Setting Stopping Criteria

Generating the Volume Mesh

The starting simulation file contains predefined mesh operations, which generate a polyhedral volume mesh extruded at the manifold outlet.

1. Click (**Generate Volume Mesh**).

   The output from the meshers is shown in the *Output* window and the mesh generation status is shown in the status bar.
2. To visualize the volume mesh, create a mesh scene.



3. Save the simulation.

Selecting Physics Models

Select physics models for the Fluid region.

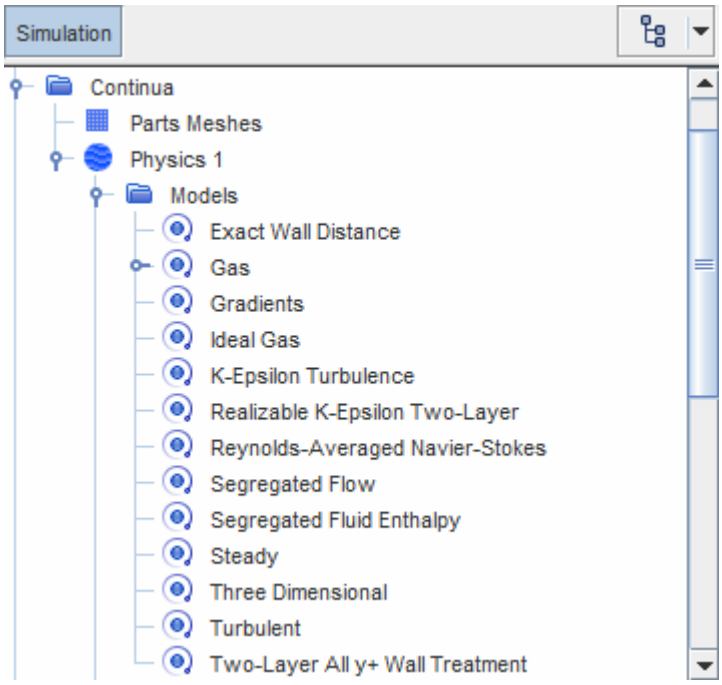
To select the physics models:

1. For the physics continuum, **Continua** > **Physics 1**, select the following physics models in order:

Group box	Model
Space	**Three Dimensional** (selected automatically)
Time	**Steady**
Material	**Gas**
Flow	**Segregated Flow**
Gradient Metrics	**Gradients** (selected automatically)
Equation of State	**Ideal Gas**
Energy	**Segregated Fluid Enthalpy**
Viscous Regime	**Turbulent**

Group box	Model
Turbulence	**Reynolds-Averaged Navier-Stokes** (selected automatically)
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**
K-Epsilon Turbulence Models	**Realizable K-Epsilon Two-Layer** (selected automatically)
K-Epsilon Two-Layer All y+ Wall Treatment	**Two-Layer All y+ Wall Treatment** (selected automatically)
Wall Distance	**Exact Wall Distance** (selected automatically)

2. To review the models, open the **Physics 1** > **Models** node.



Defining Boundary Conditions

The gas entering the manifold is traveling at 10 m/s with a temperature of 500 K. Specify these boundary conditions under the inlet boundary node, and also select a different turbulence specification.

To set the boundary conditions:

1. Edit the **Regions** > **Fluid** > **Boundaries** > **Inlets** node.
2. Set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Turbulence Specification**	*Method*	**Intensity + Length Scale**

Node	Property	Setting
Physics Values		
└ **Static Temperature**	*Value*	`500 K`
└ **Velocity Magnitude**	*Value*	`10 m/s`

3. Select the **Regions** > **Surface Extruder** > **Boundaries** > **Outlet_Top** node and set *Type* to **Outlet**.
4. Save the simulation.

Setting Up a Monitor

Create a report, monitor, and plot to monitor the temperature at a point inside the fluid domain, close to the solid-fluid interface.

By default, STAR-CCM+ produces a residuals plot which can be used to monitor the convergence of the fluid solution. You also need a way to assess convergence of the outer coupled solution between Abaqus and STAR-CCM+. Monitor the temperature at a point inside the fluid domain, close to the solid-fluid interface. When the outer coupled solution is converged, this temperature comes to a steady value.

1. Within the **Derived Parts** node, create a new point probe with the following settings:

Property	Setting
Input Parts	**Fluid**
Point	`[0.251, 0.107, -0.0405] m`
Display	**No Displayer**

   A new node, **Point**, appears within the **Derived Parts** node.

Use the point probe to monitor the temperature in the fluid:

2. Right-click the **Reports** node and select **New Report** > **Maximum**.
3. Rename the **Reports** > **Maximum 1** node to `Temperature Point`.
4. Edit the **Reports** > **Temperature Point** node and set the following properties:

Property	Setting
Field Function	**Temperature**
Parts	**Derived Parts** > **Point**

To monitor the convergence of the solution while the simulation is running:

5. Right-click the **Reports** > **Temperature Point** node and select **Create Monitor and Plot from Report**.

   Two nodes, **Temperature Point Monitor** and **Temperature Point Monitor Plot**, appear underneath the **Monitors** and **Plots** nodes, respectively.

Setting Stopping Criteria

Set the stopping criteria so that the simulation stops running when the variance of temperature across the outlet boundary falls below 0.001 K^2.

You want STAR-CCM+ to stop iterating when the variance of temperature across some part of the simulation falls to a sufficiently low value. Monitor the variance of temperature across the outlet boundary. Assume that the solution is sufficiently converged when this variance falls below 0.001 K^2.

1. Right-click the **Monitors** node and select **New Monitor** > **Field Variance**.
2. Rename the **Field Variance 1** node to `T Variance Monitor`.
3. Edit the **T Variance Monitor** node and set the following properties:

Property	Setting
Parts	**Regions** > **Surface Extruder** > **Outlet_Top**
Trigger	**Iteration**
Enable Sliding Sample Window	**Activated**
Field Function	**Temperature**
Publish Mean	**Activated**

   When *Field Function* is set to **Temperature**, a new field function, called **Variance of Temperature**, is created within the **Tools** > **Field Functions** node.

   When *Enable sliding sample window* is activated, a new node, called **Sliding Window**, appears within the **T Variance Monitor** node.
4. Select the **T Variance Monitor** > **Sliding Window** node and set *Sliding sample window size* to `10`.

To use the **Variance of Temperature** field function as a stopping criterion, you create:

- a maximum value report to return the function value on the outlet boundary
- a monitor to track the value of this report. Also, create a plot so that the value can be visualized
- a stopping criterion from the monitor

5. Right-click the **Reports** node and select **New Report** > **Maximum**.
6. Rename the **Maximum 1** node to `Outlet T Variance`.
7. Edit the **Reports** > **Outlet T Variance** node and set the following properties:

Property	Setting
Scalar Field Function	**Variance of Temperature**
Parts	**Regions** > **Surface Extruder** > **Outlet_Top**

8. Right-click the **Outlet T Variance** node and select **Create Monitor and Plot from Report**.

   Two new nodes are created: **Monitors** > **Outlet T Variance Monitor**, and **Plots** > **Outlet T Variance Monitor Plot**.

To create the stopping criterion based on the variance report monitor:

9. Right-click the **Monitors** > **Outlet T Variance Monitor** node and select **Create Stopping Criterion from Monitor**.
10. Edit the **Stopping Criteria** > **Outlet T Variance Monitor Criterion** node and set the following properties:

Node	Property	Setting
Outlet T Variance Monitor Criterion	*Logical Rule*	**And**
└ **Minimum Limit**	*Minimum Value*	`0.001 K^2`

Setting the logical rule to **And** means that the solution only stops when this criterion is satisfied with any other criterion whose logical rule is set to **And**.

To prevent the solution from stopping at the first iteration (when the variance of temperature across the outlet is zero), the Maximum Steps criterion is set to 10 iterations and its logical rule is also set to And:

11. Edit the **Stopping Criteria** > **Maximum Steps** node and set the following properties:

Property	Setting
Maximum Steps	`10`
Logical Rule	**And**

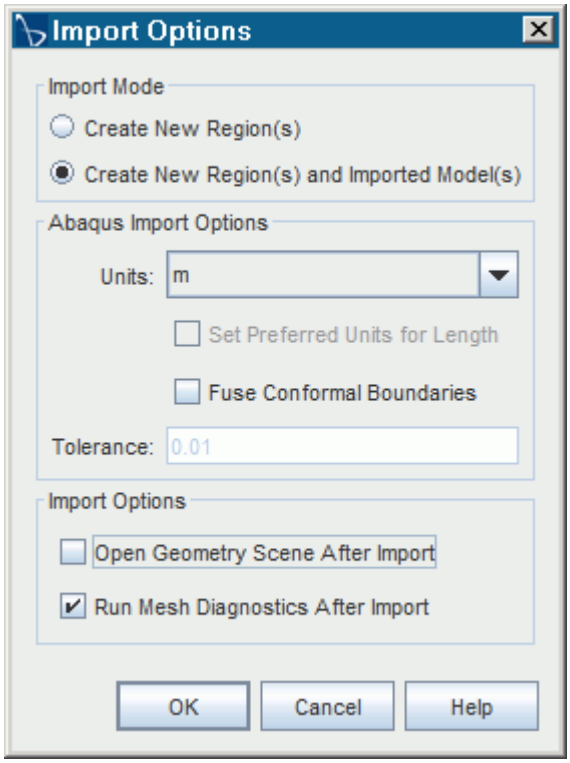
12. Save the simulation.

Importing the Abaqus Model

Import the solid model into the STAR-CCM+ simulation using the Abaqus `.inp` file.

When preparing a model in Abaqus for file-based coupling, include a surface definition (the *SURFACE keyword in the *.inp file). This command identifies the face of the 3D elements that lie at the fluid-structure interface. In this case, the interface surface has been called INNER, and two other surfaces have been defined to identify the faces for the flanges and the manifold exterior.

1. Select **File** > **Import** > **Import CAE Model...**.
2. In the *Open* dialog, select file `exhaust.inp` and click **Open**.



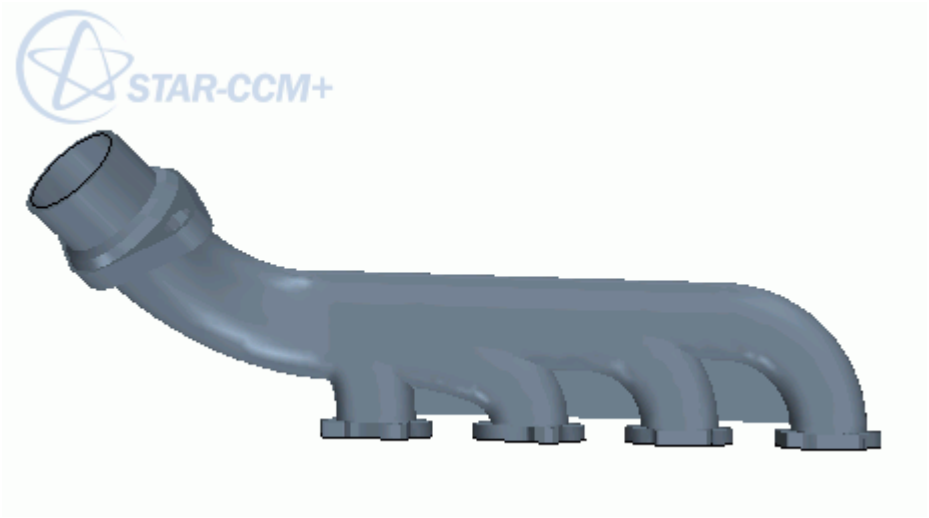
3. Click **OK**.

   A new node called **Imported Models** is added to the simulation tree. This node allows you to carry out file-based coupling operations.

   The imported Abaqus model is stored in the simulation under the **Regions** node. The model surfaces and elements (that is, shells and volumes), are also listed under the **Imported Models** node.

The solid model can be visualized in a geometry scene, as any other region in STAR-CCM+. To visualize the solid model:

4. Create a geometry scene.
5. Select the **Geometry Scene 2** > **Displayers** > **Geometry 1** > **Parts** node, and set *Parts* to **Regions** > **manifold-1**.
6. Select the **Geometry Scene 2** > **Displayers** > **Outline 1** > **Parts** node, and set *Parts* to **Regions** > **manifold-1**.
7. Position the geometry as shown below.



To highlight the imported surfaces in the scene, select their nodes under the **Imported Models** > **Imported Surfaces** node. Alternatively, select the corresponding boundaries within the **Regions** node.

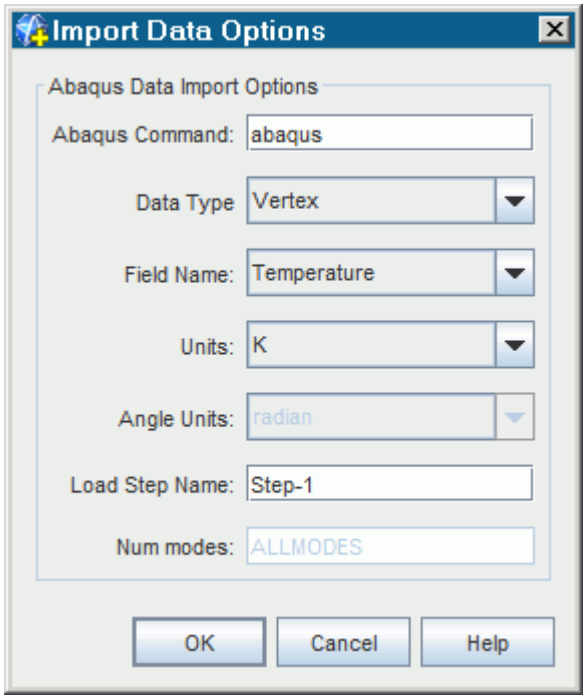
Importing Temperature Data

CAE solution data is imported in a separate step to the model. This tutorial requires that you import the temperature data from Abaqus.

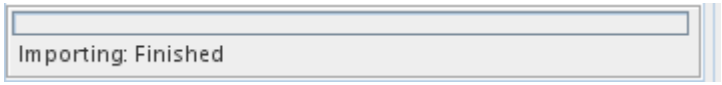
To import the Abaqus temperature data on the **INNER** surface of the Abaqus model:

1. Right-click the **Imported Models** > **Abaqus: exhaust** node and select **Import CAE Solution Data...**
2. In the *Open* dialog, navigate to the `/fileBased` directory, select the file `exhaust.odb` and click **Open**.
3. In the *Import Data Options* dialog that appears, do the following:
   a) In the *Abaqus Command* box enter the command to launch Abaqus, for example, `abaqus`. On Windows, enter `abaqus.bat`. If your environment is not set to include the command automatically from a command prompt, enter the full path.
   b) Accept the remaining default settings by clicking **OK**.

   Note: During this operation Abaqus is launched in the background to read the data, therefore you must have access to Abaqus.



The message `Importing: Finished` is displayed in the status bar when the operation is complete.



Contents:

Visualizing the Imported Temperature

Visualizing the Imported Temperature

Visualize the imported temperature data in a scalar scene.

During the process of importing the temperature data, STAR-CCM+ created a field function called **Imported Temperature**.

Use the **Imported Temperature** field function to visualize the temperature data:

1. Create a new scalar scene.
2. Rename the **Scalar Scene 1** node to `Imported Temperature`.
3. Edit the **Displayers** > **Outline 1** > **Parts** node.
4. In the *Parts* dialog, deselect the **Regions** node, then click **OK**.
5. Expand the **Displayers** > **Scalar 1** node and set the following properties:

Node	Property	Setting
Parts	*Parts*	**Imported Models** > **Abaqus: exhaust** > **Imported Surfaces** > **INNER**
Scalar Field	*Function*	**Imported Temperature**

Mapping Imported Temperature Data

Once the data has been imported into STAR-CCM+, map it onto the corresponding boundary in the Fluid region.

To map the imported temperature data:

1. Right-click the **Imported Models** > **Abaqus: exhaust** node and select **Map data** > **Surface Data...**

   The *Map Imported Model Data* dialog appears.

In this case, the **INNER** surface temperature data is mapped to the **Fluid-Wall** boundary.

2. In the *Map Imported Model Data* dialog:

   a) Set the following properties:

Group box	Property	Setting
Map Direction	*Imported Model to STAR-CCM+*	**Activated**
Mapping Inputs	*Imported Surfaces*	**Imported Surfaces** > **INNER**
	Boundaries	**Regions** > **Fluid** > **Boundaries** > **Fluid-Wall**
		Regions > **Surface Extruder** > **Boundaries** > **Fluid.Fluid-Wall**
	Data to map	**Imported Temperature** (default)

   b) Click **Apply** then **Close**.

   The *Output* window displays `Done` when the mapping process is complete. A node, **Surface Data Mapper 1**, is added under the **Tools** > **Data Mappers** node. This node contains all the settings that you entered in the mapping dialog in the previous steps.

   It would also have been possible to create an empty data mapper separately, specify the properties, then right-click the data mapper and select **Map Data**. The right-click menu options are a quicker way of setting up a data mapper.

Contents:

Visualizing the Mapped Imported Temperature

Visualizing the Mapped Imported Temperature

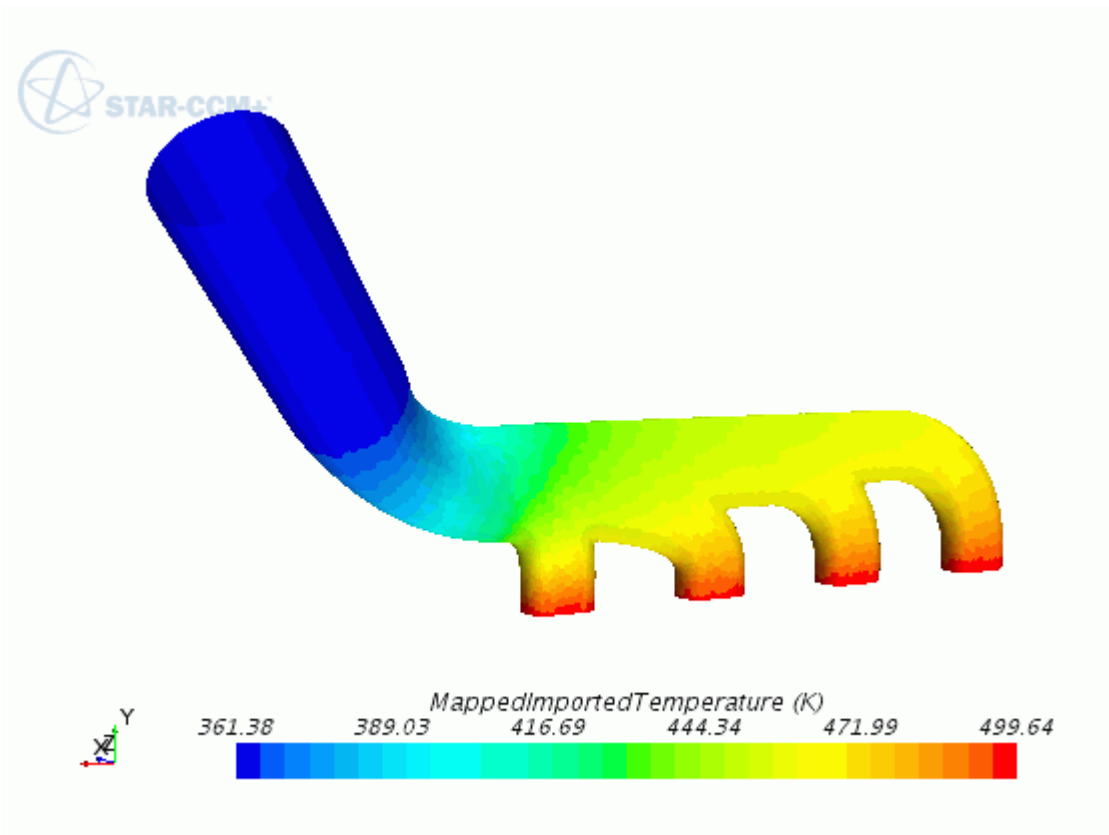
Visualize the mapped imported temperature in a scalar scene.

In the same way as for the imported temperature data, STAR-CCM+ created a mapped imported temperature field function during the mapping operation. Create a scalar scene and use this field function to visualize the mapped temperature data on the fluid wall boundary.

The new scalar scene shares a number of settings with the *Imported Temperature* scene, so create a duplicate of this scene to use as a starting point:

1. Copy the **Scenes** > **Imported Temperature** node and paste it onto the **Scenes** node.
2. Rename the **Copy of Imported Temperature** node to `Mapped Imported Temperature`.
3. Edit the **Displayers** > **Scalar 1** > **Parts** node.
4. In the *Edit* dialog:
   a) Deselect the **Imported Models** node.
   b) Select the **Regions** > **Fluid** > **Boundaries** > **Fluid-Wall** node and the **Regions** > **Surface Extruder** > **Boundaries** > **Fluid.Fluid-Wall** node.
   c) Click **OK**.
5. Select the **Scalar 1** > **Scalar Field** node, and set *Function* to **MappedImportedTemperature**.

A scene similar to the one shown below is displayed.



Comparing the imported temperature contour with the mapped imported temperature contour, it is clear that the imported temperature data from Abaqus has been mapped successfully onto the fluid wall boundary.

Preparing Scalar Scenes

Prepare two scalar scenes to visualize the contours of fluid temperature and velocity on a plane section as the simulation runs.

To create a plane section through the fluid region:

1. Activate *Mesh Scene 1* by clicking the corresponding tab at the top of the *Graphics* window.
2. Within the Derived Parts node, create a new plane section.
3. In the *Create Section* dialog, set the following properties:

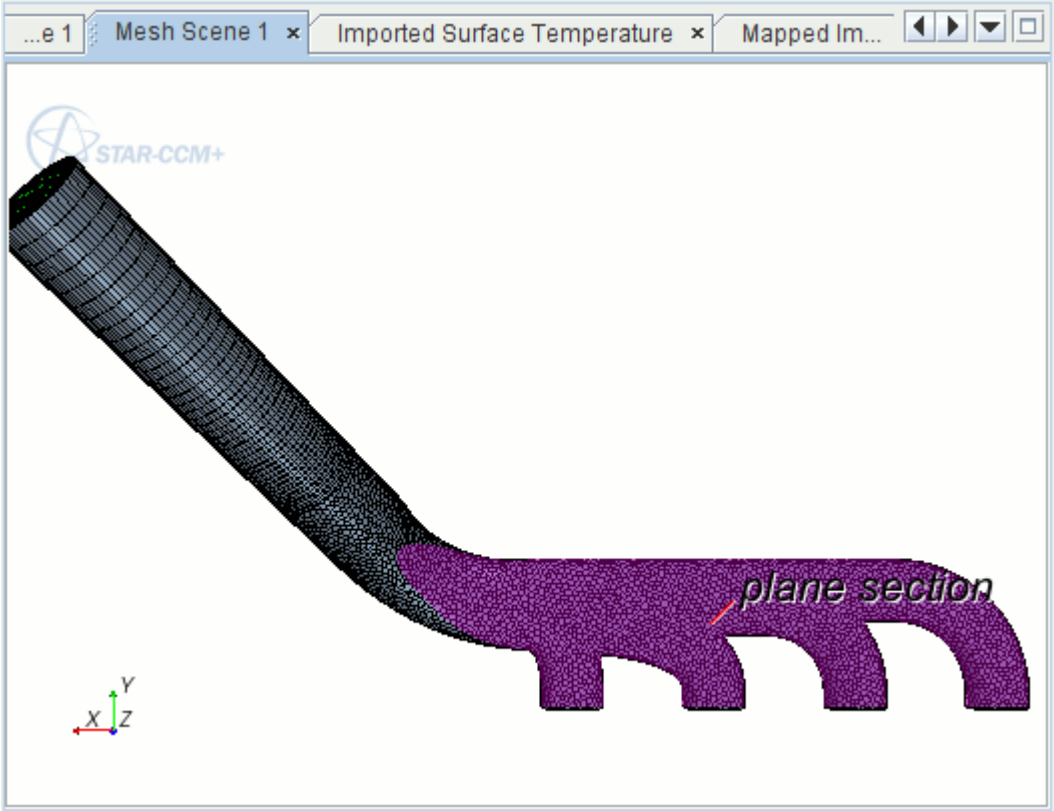
Property	Setting
Input Parts	**Fluid**
origin	`[0, 0, 0] m`
normal	`[0, 0, 1] m`
Display	**No Displayer**

4. Click **Create** and then **Close**.

   A node, called **Plane Section**, appears within the **Derived Parts** node.

To highlight this part in the active scene:

5. Expand the **Derived Parts** node and select **Plane Section**.



To visualize the velocity magnitude on the plane section:

6. Create a scalar scene.
7. Rename the **Scalar Scene 1** node to `Fluid Velocity`.
8. Select the **Fluid Velocity** > **Displayers** > **Outline 1** > **Parts** node and set *Parts* to **Derived Parts** > **Plane Section**.
9. Expand the **Fluid Velocity** > **Displayers** > **Scalar 1** node and set the following properties:

Node	Property	Setting
Parts	*Parts*	**Derived Parts** > **Plane Section**
Scalar Field	*Function*	**Velocity** > **Magnitude**

10. Click (**Save-Restore-Select views**) and select **Views** > **-Z** > **Up +Y**

To visualize the fluid temperature on the plane section:

11. Copy the **Scenes** > **Fluid Velocity** node and paste it onto the **Scenes** node.
12. Rename the **Copy of Fluid Velocity** node to `Fluid Temperature`.
13. Expand the **Scenes** > **Fluid Temperature** > **Displayers** > **Scalar 1** node.
14. Select the **Scalar Field** node and set *Function* to **Temperature**.
15. Save the simulation.

Running the Simulation Using Imported Data

The STAR-CCM+ simulation runs using the imported temperature data as a boundary condition.

The temperature data that has been imported from the Abaqus model and mapped onto the fluid boundary can be used to define a Temperature boundary condition. This fixes the temperature at the Fluid-Wall boundary, that is, the interface between the fluid and solid domains, to be the temperature that is calculated in the Abaqus analysis. It is expected that this initial temperature solution from Abaqus is a good starting point for the coupled simulation.

Contents:

Specifying a Temperature Boundary Condition
Running the Simulation
Visualizing Initial Results

Specifying a Temperature Boundary Condition

Set the temperature at the fluid-wall to the surface temperature imported from the Abaqus simulation.

To set the temperature at the fluid wall:

1. Multi-select the **Regions** > **Fluid** > **Boundaries** > **Fluid-Wall** node and the **Regions** > **Surface Extruder** > **Boundaries** > **Fluid.Fluid-Wall** node.
2. Right-click one of the selected nodes and select **Edit...**
3. Set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Thermal Specification**	*Condition*	**Temperature**
Physics Values		
└ **Static Temperature**	*Method*	**Field Function**
	Scalar Function	**MappedImportedTemperature**

4. Save the simulation.

Running the Simulation

The simulation is now fully set up and ready to be run using the imported data.

1. Run the simulation

   The iteration progress can be monitored in the *Output* window. Monitor the convergence by viewing the **Temperature Point Plot** and the **Residuals** using the visualization tabs in the **Graphics** window.

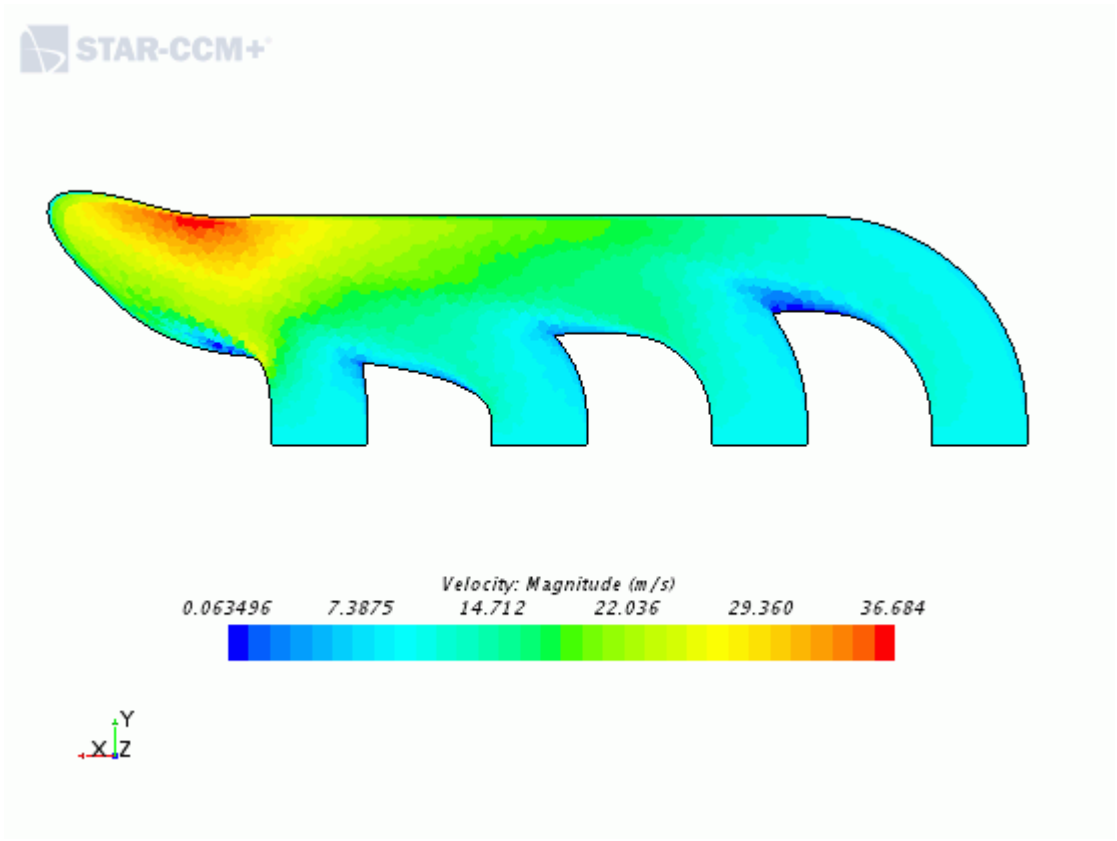
   When the convergence criterion is satisfied, the message `Stopping criteria Maximum Steps and Outlet T Variance Monitor Criterion satisfied` is displayed in the *Output* window.

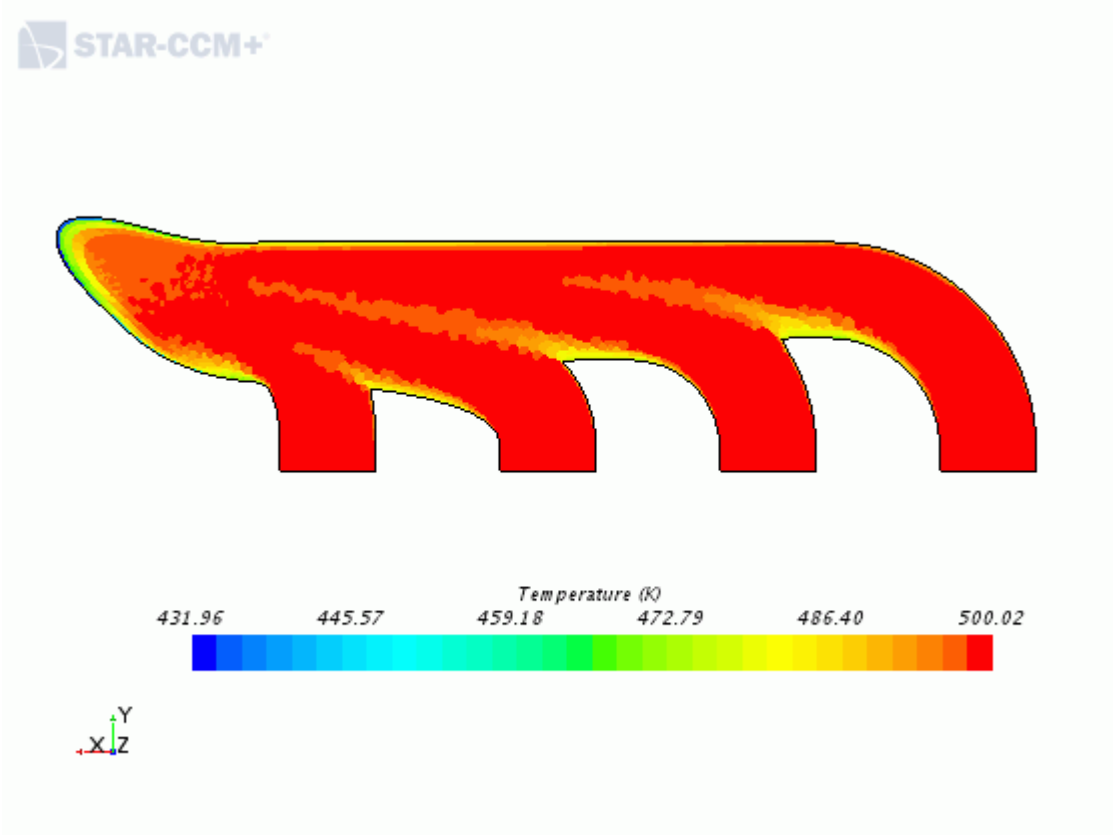
2. Save the simulation.

Visualizing Initial Results

Visualize the fluid velocity and fluid temperature scalar scenes that you prepared earlier.

1. Open the scenes to visualize the results from the initial run:





Reaching a Converged Solution

Perform multiple data exchange cycles between Abaqus and STAR-CCM+ to reach a converged solution.

Often, more iterations are required to obtain a fully converged solution. The number of iterations necessary between STAR-CCM+ and Abaqus depend on how close the initial conditions are to the final temperature solution. Perform the import and mapping operations each time data is passed from Abaqus to STAR-CCM+. When data is passed from STAR-CCM+ to Abaqus, map the data to the solid model, and export it.

To facilitate this process, record two macros for the following tasks:

- Mapping boundary data onto Abaqus surface and exporting it to Abaqus.
- Importing data from Abaqus and mapping it onto a boundary in STAR-CCM+.

These macros can be played for each required coupling iteration rather than going through the steps one at a time.

Contents:

Mapping Boundary Data onto the Abaqus Surface
Exporting Mapped Data
Modifying the Abaqus Input File and Running the Abaqus Job
Importing and Mapping Temperature Data
Running the Simulation

Running Abaqus and STAR-CCM+ Sequentially

Mapping Boundary Data onto the Abaqus Surface

Record a macro to map STAR-CCM+ boundary data to the Abaqus surface.

When the fluid solution has converged, the thermal data at the solid-fluid interface, that is, the Fluid-Wall boundary, can be mapped from STAR-CCM+ to the INNER surface on the imported model. This data can then be exported in a format that can be used to define a thermal load in the Abaqus model.

Before carrying out the mapping and export processes, start recording the first macro. Once the recording has started, each action you perform in STAR-CCM+ is included in the macro. Use this macro later on to repeat the map and export processes.

To start recording the first macro:

1. Click (**Start Recording**) to initiate the recording process.
2. In the *Save* dialog, enter `loopMapExport` for the name of the Java macro.
3. Click **Save**.

From this point onwards, any recordable action you make in STAR-CCM+ generates lines of Java code in this page. Clicking (**Pause Recording**) pauses generation of the Java code; clicking the same button resumes generation of the Java code. Clicking (**Stop Recording**) ends all generation of the Java code and write the macro to file. Do not click either of these buttons for now.

Proceed by mapping the STAR-CCM+ solution data onto the Abaqus model surface:

4. Right-click the **Imported Models** > **Abaqus: exhaust** node and select **Map data** > **Surface Data...**

   The *Map Imported Model Data* dialog appears, which allows you to set up the mapping operation.
5. In the *Map Imported Model Data* dialog:
   a) Set the following properties:

Group box	Property	Setting
Map Direction	*STAR-CCM+ to Imported Model*	**Activated**
Mapping Inputs	*Imported Surfaces*	**Imported Surfaces** > **INNER**
	Boundaries	**Regions** > **Fluid** > **Boundaries** > **Fluid-Wall**
		Regions > **Surface Extruder** > **Boundaries** > **Fluid.Fluid-Wall**
	Data to map	**Local Heat Transfer Coefficient** (default)
Reference Temperature Evaluation Option	*Use Local Heat Transfer Reference Temperature*	**Selected** (default)

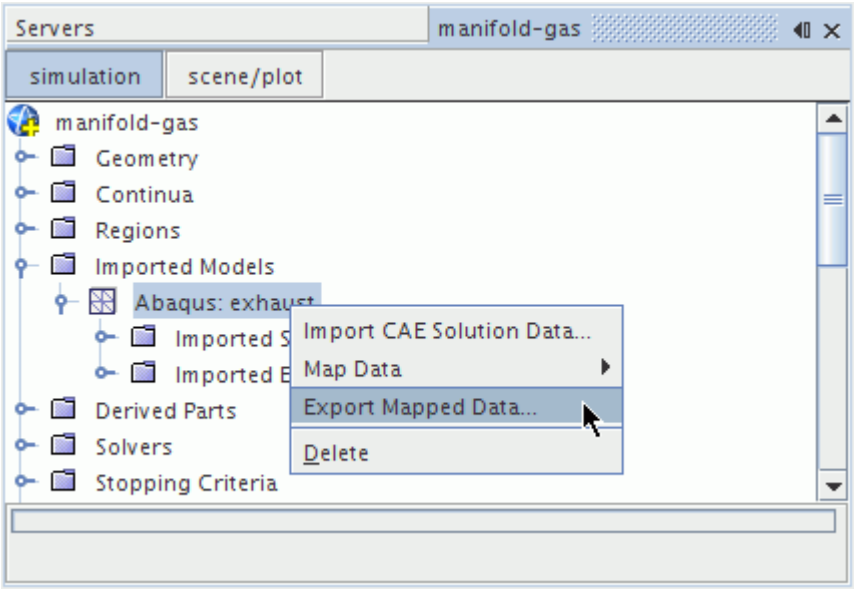
   b) Click **Apply** then **Close**.

The *Output* window displays the time to map the data, and `Done` when the mapping process is complete.

Exporting Mapped Data

Once the data has been mapped onto the imported Abaqus surface, it can be exported to an .inp file, which then can be read into the Abaqus job. Stop the macro recording when the export process is complete.

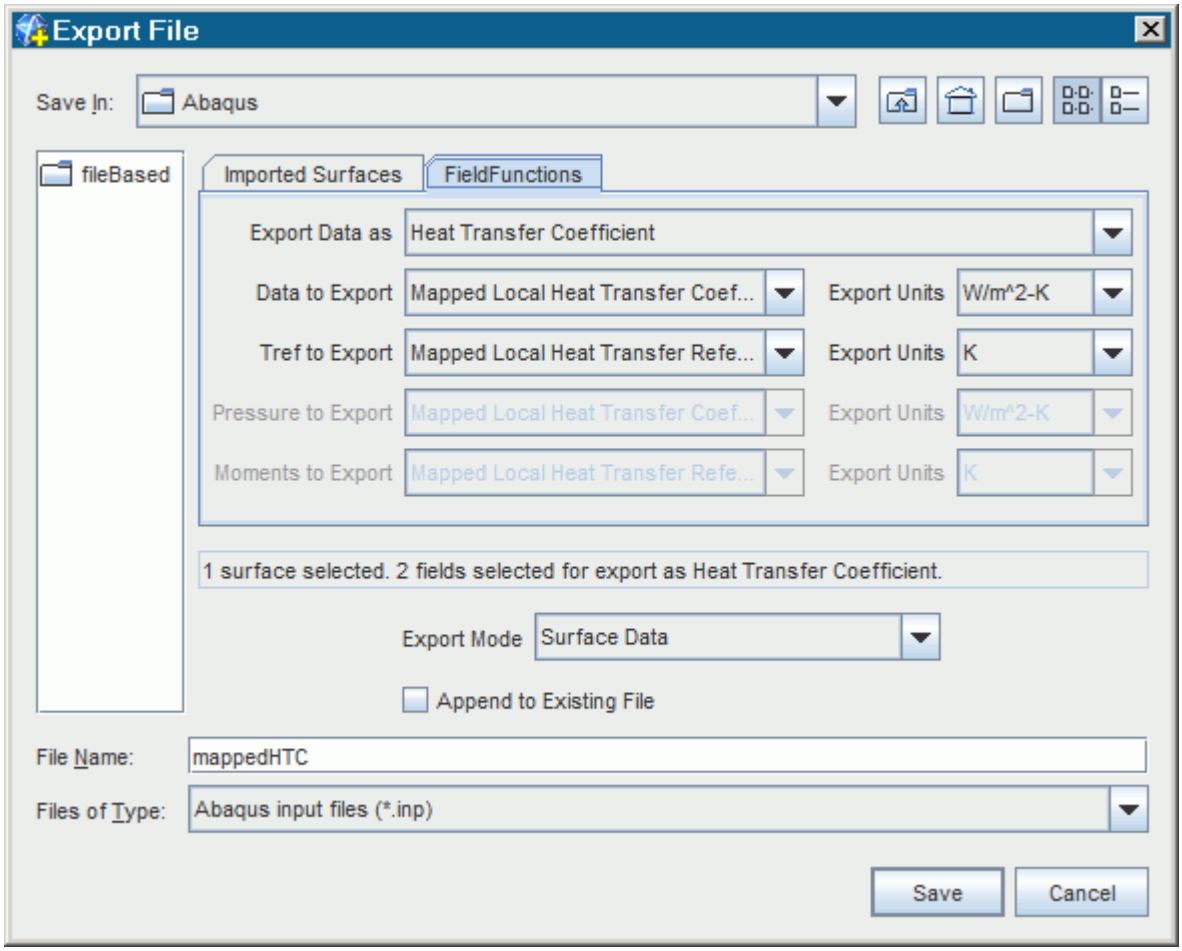
1. Right-click the **Imported Models** > **Abaqus: exhaust** node and select **Export Mapped Data...**



In the *Export File* dialog:

2. Click the *Imported Surfaces* tab and select the **INNER** node.
3. Click the *FieldFunctions* tab and enter the following settings:

Property	Setting
Export Data as	**Heat Transfer Coefficient**
Data to Export *Export Units*	**Mapped Local Heat Transfer Coefficient** **W/m^2-K**
Tref to Export *Export Units*	**Mapped Local Heat Transfer Reference Temperature** **K**
Export Mode	**Surface Data**
Append to Existing File	**Deactivated**
File Name	**mappedHTC**
Files of Type	**Abaqus input files (*.inp)**



4. Click **Save**.

   The export process is now complete, and the Java macro contains all the necessary steps for the map-export process.

5. Click (**Stop Recording**) to end the recording and write the macro to the file.

Modifying the Abaqus Input File and Running the Abaqus Job

Use the solution data from the fluid solution to define a thermal load in the Abaqus model.

Add a command in the Abaqus input file:

1. Open the `exhaust.inp` file in a script editor of your choice.
2. Scroll down to the section that begins with the following:

```
**
** STEP: Step-1
**
```

   This section of the file, up to `*End Step`, contains the history data for the Abaqus job, including solver parameters, boundary conditions, loads, and outputs.

3. To cause Abaqus to read in the heat transfer coefficients from the `mappedHTC.inp` file, add the following lines that are shown in bold text:

```
*Output, field, variable=PRESELECT
*Output, history, frequency=0
*Include, input=mappedHTC.inp
*End Step
```

4. Save the file under the name `exhaust-mod.inp` and exit the script editor.
5. In a shell/command prompt, run the Abaqus job using the modified input file by typing the following command:

```
> abaqus interactive job=exhaust-mod
```

where abaqus is the command you use to launch Abaqus.

The set of files created after running this job are named `exhaust-mod.*`

Importing and Mapping Temperature Data

The results from the Abaqus analysis, exhaust-mod, contain an updated set of temperature data. Import the set into STAR-CCM+ and map it onto the Fluid-Wall boundary using the same technique as described previously.

In a similar way as for the map-export process, record a macro for the process of importing the latest data from Abaqus, and mapping it onto the boundary. Include some additional steps in this macro, so that the simulation begins running once the data has been mapped.

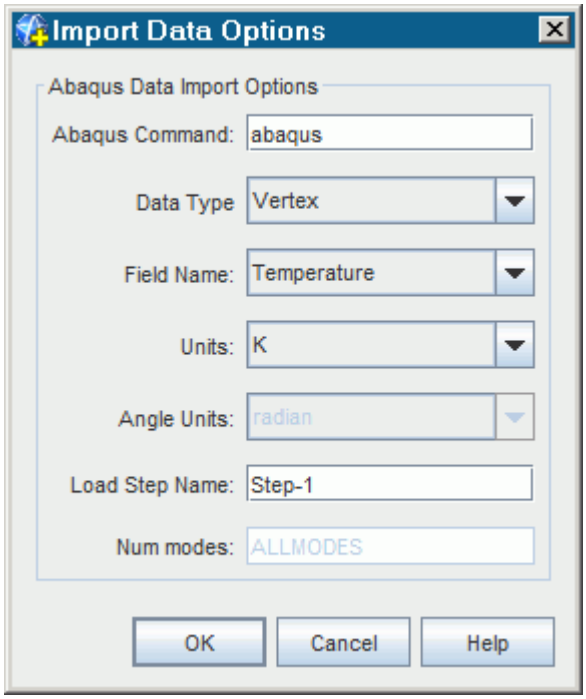
To start recording the second macro:

1. Click (**Start Recording**) to initiate the recording process.
2. In the *Save* dialog, enter `loopImportMap` for the name of the Java macro.
3. Click **Save**.

Proceed by importing the temperature data from the latest Abaqus analysis:

4. Right-click the **Imported Models** > **Abaqus: exhaust** node and select **Import CAE Solution Data**.
5. In the *Open* dialog, select the `exhaust-mod.odb` file and click **Open**.
6. In the *Import Data Options* dialog:
   a) In the *Abaqus Command* box enter the command to launch Abaqus, for example, `abaqus`. On Windows, enter `abaqus.bat`. If your environment is not set to include the command automatically from a command prompt, enter the full path.
   b) Accept the remaining default settings by clicking **OK**.

Note: During this operation Abaqus is launched in the background to read the data, therefore you must have access to Abaqus.



The message `Importing: Finished` is displayed in the status bar when the operation is complete. The new set of imported temperature data can be visualized in the *Imported Temperature* scene you created earlier.

7. Right-click the **Imported Models** > **Abaqus: exhaust** node and select **Map data** > **Surface Data...**

   The *Map Imported Model Data* dialog appears. This dialog lets you carry out mapping operations.

In this case, the **INNER** surface temperature data is mapped to the fluid wall boundary.

8. In the *Map Imported Model Data* dialog:

   a) Set the following properties:

Group box	Property	Setting
Map Direction	*Imported Model to STAR-CCM+*	**Activated**
Mapping Inputs	*Imported Surfaces*	**Imported Surfaces** > **INNER**
	Boundaries	**Regions** > **Fluid** > **Boundaries** > **Fluid-Wall**
		Regions > **Surface Extruder** > **Boundaries** > **Fluid.Fluid-Wall**
	Data to map	**Imported Temperature** (default)

   b) Click **Apply** then **Close**.

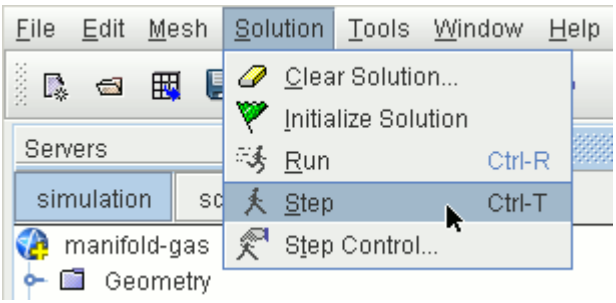
   The *Output* window displays `Done` when the mapping process is complete. The results of the mapping operation can be visualized in the *Mapped Imported Surface Temperature* scene.

Running the Simulation

Following the import and mapping operations in the previous section, you can proceed by running the simulation.

To make sure that the **Temperature Point Monitor** stopping criterion is reset, first perform a single iteration:

1. From the menu, select **Solution** > **Step**.



2. Run the simulation.
3. When the simulation has finished running, save the simulation.
4. Click (**Stop Recording**) to end recording and to write the macro to the file.

In the *Temperature Point Plot*, you can observe a “jump” in the temperature solution when the next set of temperature data from Abaqus was imported. This jump represents the exchange of solution data between the two codes. As the Abaqus solution converges with the STAR-CCM+ solution, these jumps become smaller, until you can say that the solution is fully converged.

Running Abaqus and STAR-CCM+ Sequentially

Following the completion of the second run in STAR-CCM+, you can now pass thermal data back to Abaqus and run the Abaqus job. You can then return to STAR-CCM+, import the latest set of temperature data, map the data, and run the simulation.

Overwrite the old data in the `mappedHTC.inp` file with the latest thermal data from STAR-CCM+. Recall that the first macro you recorded, `loopMapExport.java`, performs these actions automatically, so you can proceed by playing this macro.

1. Click (**Play Macro**).
2. In the *Open* dialog, select `loopMapExport.java` and click **Open**.

   When the macro has finished running, a message beginning `Export Mapped Data:MappedLocal` is displayed in the *Output* window. This signifies that the new set of thermal data is ready to be used in the Abaqus analysis.

Proceed by running the Abaqus job:

3. In a shell/command prompt, run the Abaqus job by typing:

```
> abaqus interactive job=exhaust-mod
```

where abaqus is the command you use to launch Abaqus. When prompted, confirm that it is OK to overwrite the old job files.

When the Abaqus job is complete, you can return to STAR-CCM+, import the latest set of temperature data, map the data, and run the simulation. These steps can be carried out automatically by using the macro `loopImportMap.java`:

4. In STAR-CCM+, click (**Play Macro**).
5. In the *Open* dialog, select `loopImportMap.java` and click **Open**.

   Recall that this macro has finished running when the stopping criterion has been reached and the simulation is saved.

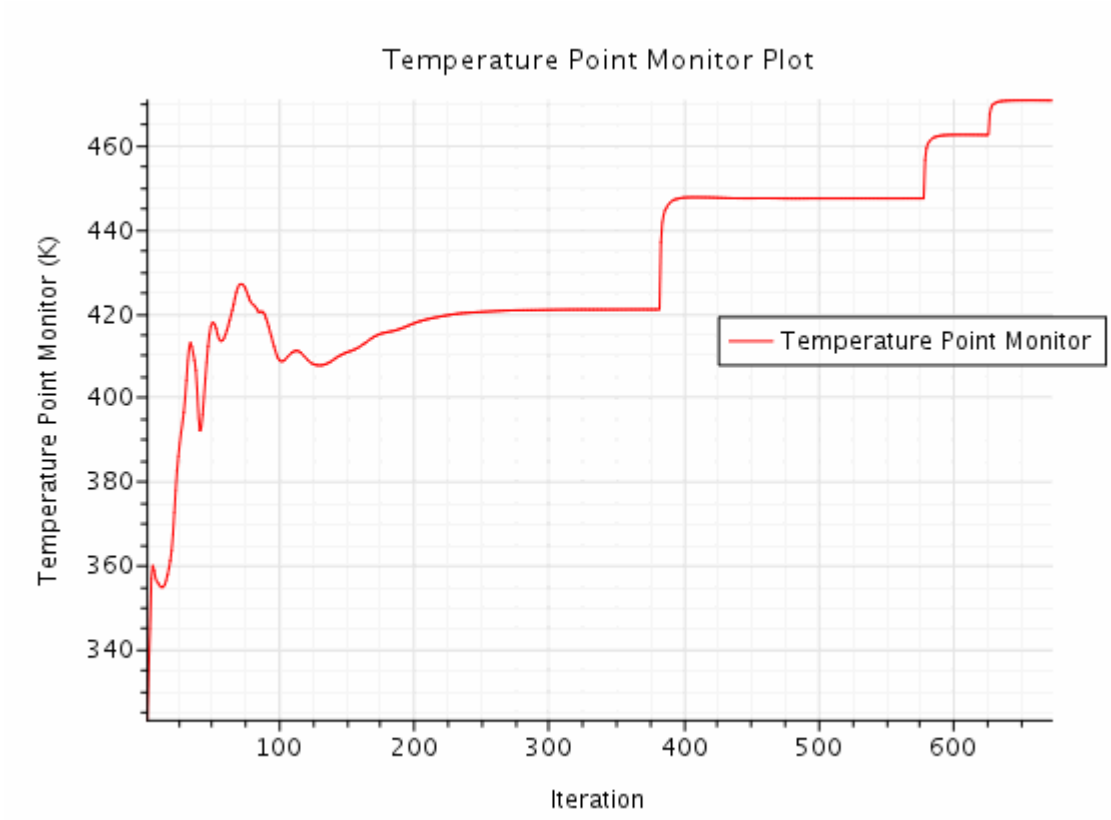
Each time the file-based coupling occurs, you move closer to reaching the final fluid-solid thermal solution. Continue to run Abaqus and STAR-CCM+ sequentially by repeating the following steps:

6. In STAR-CCM+, click (**Play Macro**) and run the `loopMapExport.java` macro.
7. In a shell/command prompt, run the Abaqus job by typing:

```
> abaqus interactive job=exhaust-mod
```

8. In STAR-CCM+, click (**Play Macro**) and run `loopImportMap.java` macro.

   The model converges after approximately four coupling iterations, and the *Temperature Point Monitor Plot* appears as shown below.



The thermal solution in the solid model can be post-processed in Abaqus/CAE using the `exhaust-mod.odb` file.

Summary

This tutorial has introduced the following STAR-CCM+ features:

- Creating an asymptotic stopping criterion from a monitor
- Importing an Abaqus model
- Importing and visualizing temperature data for the Abaqus model
- Mapping the imported temperature data
- Using the imported data to define a boundary condition
- Running a simulation using imported temperature data
- Mapping and exporting STAR-CCM+ solution data
- Using exported data in an Abaqus job
- Recording macros for import-map-export processes
- Reaching a converged solution between Abaqus and STAR-CCM+

Abaqus Co-Simulation: Thermal Coupling

This tutorial demonstrates how to run a thermal co-simulation case using STAR-CCM+ and Abaqus. Co-simulation involves a strong coupling between the two codes, in which data exchange occurs at frequent intervals, that are known as coupling steps. In this case, data is exchanged at the end of each time-step, and a relatively large time-step is used to approximate a steady-state solution.

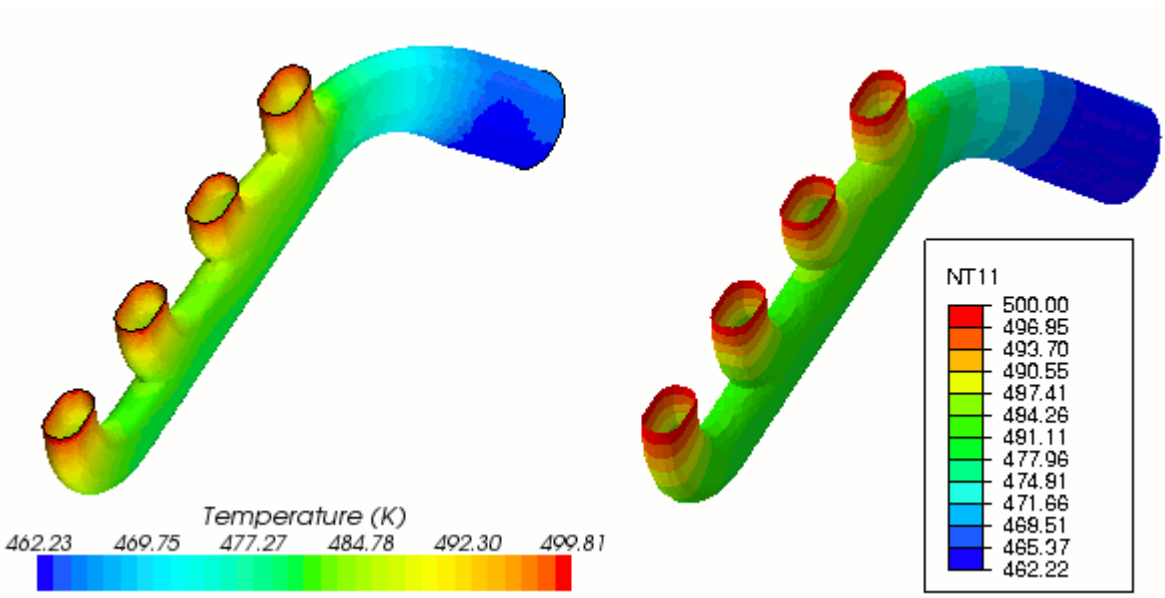
The deep level of communication between the solvers allows you to obtain a full solution across the solid-fluid interface. The solution includes the effects of the interaction between the solid and fluid domains. STAR-CCM+ passes heat flux data to Abaqus, and Abaqus passes temperatures back to STAR-CCM+. In each case, the thermal data defines a boundary condition in the model, to be used in the next set of iterations.

The coupling is carried out using the SIMULIA Co-Simulation Engine, which exchanges data between the two models automatically during the co-simulation. This technique differs from the file-based method in that the coupling between the two codes is much stronger; the exchange of data is carried out automatically, and STAR-CCM+ and Abaqus run simultaneously.

To prepare a co-simulation case, two models are required. The Abaqus model is for the solid domain, and is used to create an Abaqus input file. In STAR-CCM+ the fluid model is set up, with the appropriate boundary conditions defined. For each model, specify a set of co-simulation parameters which define the boundary/surface at the fluid-structure interface, the field functions that are imported and exported at this interface, the frequency of coupling steps, and so on. With the co-simulation parameters defined, Abaqus can be launched from the STAR-CCM+ GUI, and the co-simulation run.

Before running the co-simulation, it is good practice to run standalone cases for the fluid and solid models to make sure that they converge independently. In this example, the Abaqus model could be run standalone using a surface film condition, `*SFILM`, on the INNER surface with estimated constant values for the sink temperature and film coefficient. Similarly, in STAR-CCM+, the simulation could be run using an estimated temperature on the Fluid-Wall boundary.

The case is an exhaust manifold subject to 10 m/s inlet gases at 500 K. A solution for the flow and temperature in the exhaust gas, and the temperature in the solid manifold is required.



This tutorial is divided into three sections. In the first section, the Abaqus model and input file are created using Abaqus/CAE. In the second section, the fluid model is prepared in STAR-CCM+. The third section involves editing the Abaqus input file for co-simulation using a script editor, defining the co-simulation session in STAR-CCM+, and running the co-simulation. The third section also demonstrates how to start and close Abaqus via the STAR-CCM+ user interface.

Contents:

Prerequisites
Preparing the Abaqus Model
Preparing the STAR-CCM+ Model
Running the Co-Simulation
Visualizing the Results
Summary

Prerequisites

The instructions in the Abaqus Thermal Coupling tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Techniques	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

This tutorial demonstrates co-simulation with Abaqus 2016.

Preparing the Abaqus Model

In this section, you prepare the model for the exhaust manifold in Abaqus/CAE, including the corresponding Abaqus input file (`*.inp`).

It is recommended that you save the files for each application in separate subdirectories within an empty directory, to help you to organize the files that are created during the co-simulation.

1. In your working directory, create a directory `/thermalCoSimTutorial`.
2. In that directory, create two additional directories: `/abaqus` and `/starccm+`.

   The files for the Abaqus part of the co-simulation are saved in `/abaqus` and the files for STAR-CCM+ are saved in `/starccm+`.

   If you are unfamiliar with Abaqus/CAE, it is recommended that you work through the examples in the Getting Started with Abaqus: Interactive Edition manual before attempting this part of the tutorial, in particular “Abaqus Basics”, section 2.

   A copy of the input file that is created using Abaqus/CAE is provided in the STAR-CCM+ installation. If you do not wish to work through the steps involving the preparation of the Abaqus input file, you can jump directly to the tutorial section, Preparing the STAR-CCM+ Model.

   To work through the steps to set up the model in Abaqus/CAE, proceed to the next section.

Contents:

Starting Abaqus/CAE and Importing Geometry
Creating a Material
Defining and Assigning Section Properties
Creating a Partition
Creating an Assembly
Defining Surfaces and Sets
Defining the Analysis Steps
Defining Initial Conditions
Defining Boundary Conditions
Meshing the Model
Creating the Abaqus Input File
Specifying Co-Simulation Settings in Abaqus

Starting Abaqus/CAE and Importing Geometry

Import the manifold geometry into Abaqus/CAE as a part.

An exhaust manifold geometry has been prepared for this tutorial and saved in the STEP (Standard for the Exchange of Product model data) file format. Begin by copying this file from the STAR-CCM+ installation directory to the directory created for the Abaqus files.

1. Navigate to the `[INSTALL_DIR]/doc/startutorialsdata/couplingWithCaeCodes/data` directory and copy the file `manifold.x_t` into the `/thermalCoSimTutorial/abaqus/` directory.

To import the manifold geometry into Abaqus/CAE as a part:

2. Launch Abaqus/CAE and select the option to Create Model Database **With Standard/Explicit Model** in the *Start Session* dialog.
3. Select **File** > **Import** > **Part**.
4. In the *Import Part* dialog, change **File Filter** to **Parasolid (*.x_t*,*.x_b*,*.xmt*)** using the drop-down list.
5. Navigate to the `/thermalCoSimTutorial/abaqus/` directory and select the file `manifold.x_t`.
6. Click **OK**.
7. In the *Create Part from PARASOLID File* dialog, accept the default options by clicking **OK**.
8. If a warning message regarding imprecise geometry is displayed, click **Dismiss**.

   In this case, imprecise geometry does not affect the analysis, since the part is meshed using tetrahedral elements.

The geometry import process takes a few minutes, and when it is complete the manifold part is displayed in the viewport.

Creating a Material

Create a material definition for the manifold and specify the material properties that are required for a thermal analysis.

In general, given the type of analysis, you specify the following material properties:

- For steady-state heat transfer analyses, you specify:
  - Thermal Conductivity
- For transient heat transfer analyses, you also specify:
  - Density
  - Specific Heat
- For thermal-structural analyses, you also specify:
  - Young's Modulus
  - Poisson Ratio
  - Coefficient of Thermal Expansion

In your own cases, you can also specify other properties depending on the material and analysis requirements.

As this is a thermal analysis, you only specify the material thermal conductivity, density, and specific heat.

To set the material properties for this case, create a material definition and specify numerical values for density, thermal conductivity, and specific heat, without entering units. In this tutorial, the units used by Abaqus are automatically set to SI units.

1. In the *Model Tree*, double-click the **Materials** container to create a material definition.
2. In the material editor that appears:
   a) Name the material `GRAY CAST IRON`.
   b) Select **General** > **Density** and enter a value of `7817` (kg/m^3).

c) Select **Thermal** > **Conductivity** and enter a value of `55` (W/mK).

d) Select **Thermal** > **Specific Heat** and enter a value of `446` (J/kg K).

Note: In Abaqus, all numerical values are cast in the same system of units. When coupling STAR-CCM+ to Abaqus, you can control what units Abaqus uses from your STAR-CCM+ simulation, through the **Co-Simulations** > **Abaqus Co-Simulation 1** > **Values** > **External Code Units Manager** node. In this tutorial, you use SI units, that correspond to the default units specified under the **External Code Units Manager** node.

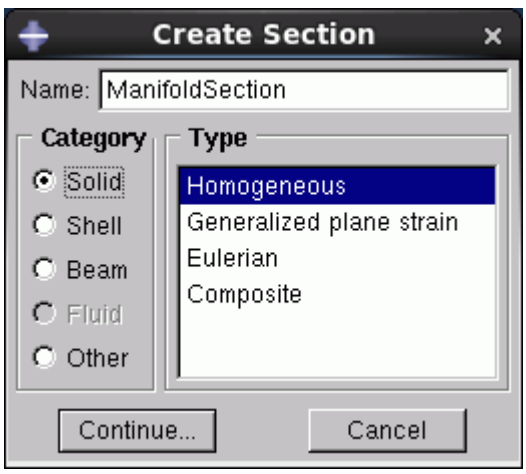
3. Click **OK** to close the dialog.

Defining and Assigning Section Properties

Create a section from the material and assign it to the manifold part.

To create a section definition:

1. In the Model Tree, double-click the **Sections** container.
2. In the *Create Section* dialog, change the name of the section to `ManifoldSection`.
3. Make sure that the category and type are set to **Solid** and **Homogeneous**, and click **Continue...**



4. In the *Edit Section* dialog, make sure that the **Material** is set to **GRAY CAST IRON** and click **OK**.

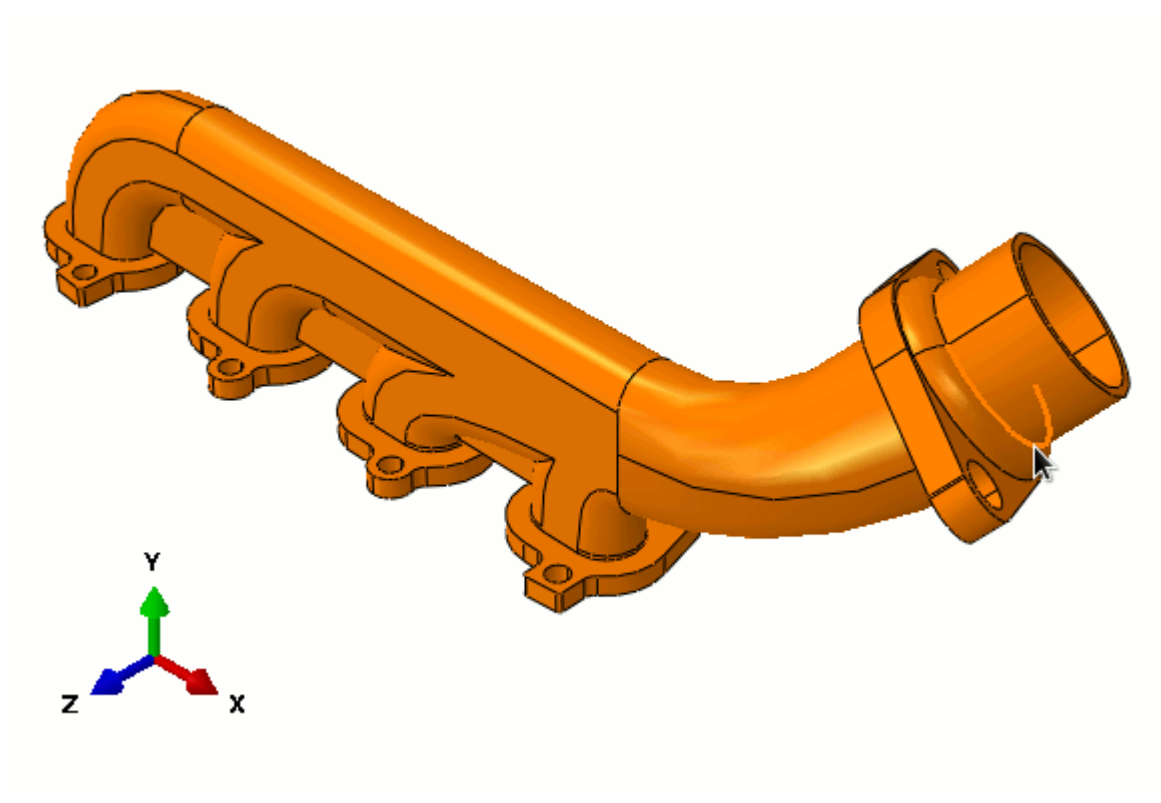
5. In the Model Tree, expand the **manifold** item underneath the **Parts** container and double-click **Section Assignments** in the list of part attributes.
6. To assign the section to the part, select the part in the viewport and click **Done** in the prompt area.
7. In the *Edit Section Assignment* dialog that appears, make sure that **Section** is set to **ManifoldSection** and click **OK**.

Creating a Partition

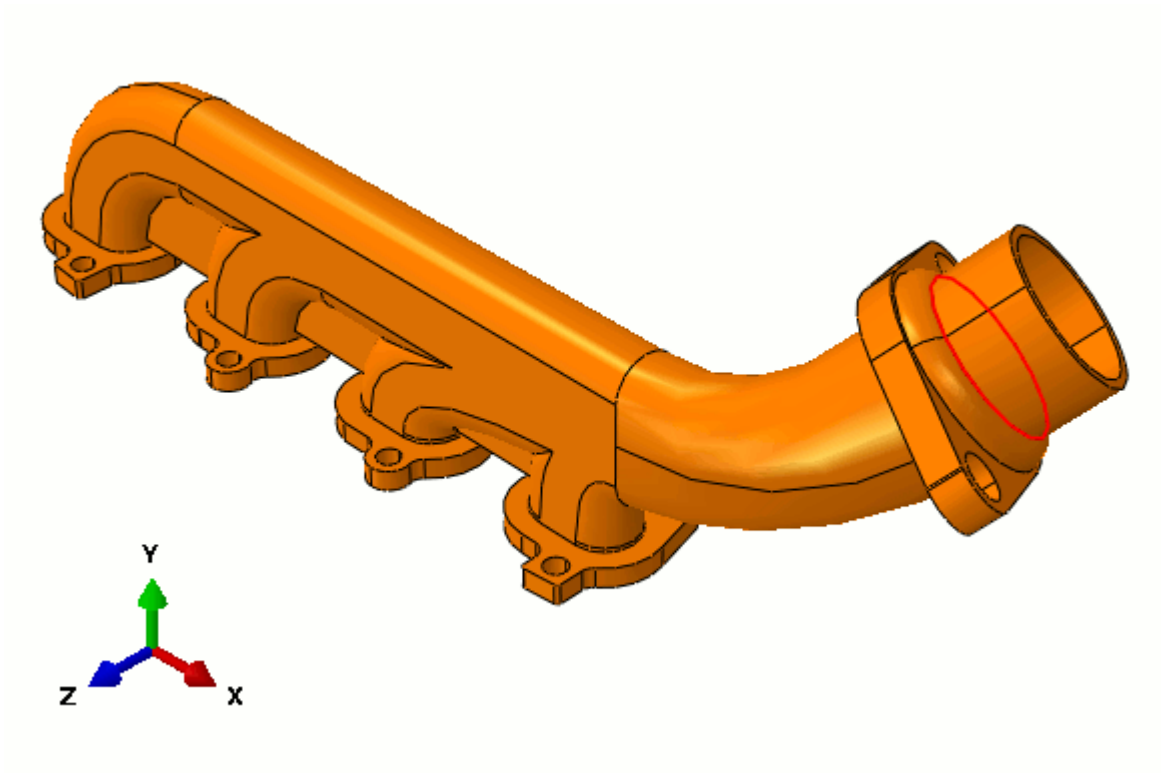
As the manifold geometry is complex, a partition is created to aid in the meshing process.

Note: Partition before defining surfaces and sets on the manifold to make sure that the correct faces are referenced when these features are defined.

1. Switch to the Mesh module by double-clicking **Mesh (Empty)** underneath the **Parts** > **manifold** container.
2. Select **Tools** > **Partition...**
3. In the *Create Partition* dialog, change *Type* to **Cell**, and select **Use n-sided patch** in the *Method* box.
4. At the bottom of the viewport, choose the option to **Select Edges**.
5. Hover the mouse over the edge that is shown in the following screenshot:



6. Left-click to select the edge; the whole loop is selected.



7. Click **Create Partition**, at the bottom of the viewport, to create the partition.

   When the outlet section has been partitioned, it is displayed in yellow in the viewport.
8. Close the *Create Partition* dialog and press <Esc> to exit the partitioning tool.

Creating an Assembly

An assembly contains all the geometry that forms the final finite element model. In this case, the assembly contains only the manifold geometry, which is added to the assembly by creating an instance of the part.

To create an instance:

1. In the Model Tree, expand the **Assembly** container and double-click **Instances**.
2. In the *Create Instance* dialog, make sure that **manifold** is selected, keep the default options and click **OK**.

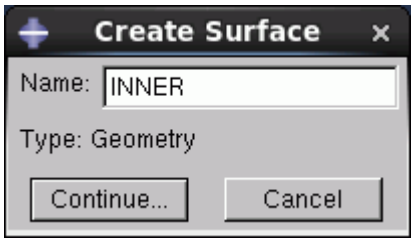
Defining Surfaces and Sets

Specify the FSI interface as the inner wetted surface of the manifold model, and define two additional surfaces to specify thermal conditions on the manifold exterior and the flanges.

Surfaces and sets identify particular regions of the model with a unique name. They can be used when defining loads and boundary conditions, and must be used in co-simulation cases to identify the interface between the fluid and solid domains. In this case, specify the FSI interface as the inner wetted surface of the manifold model, and define two additional surfaces to specify thermal conditions on the manifold exterior and the flanges. Create two sets to apply temperature boundary conditions.

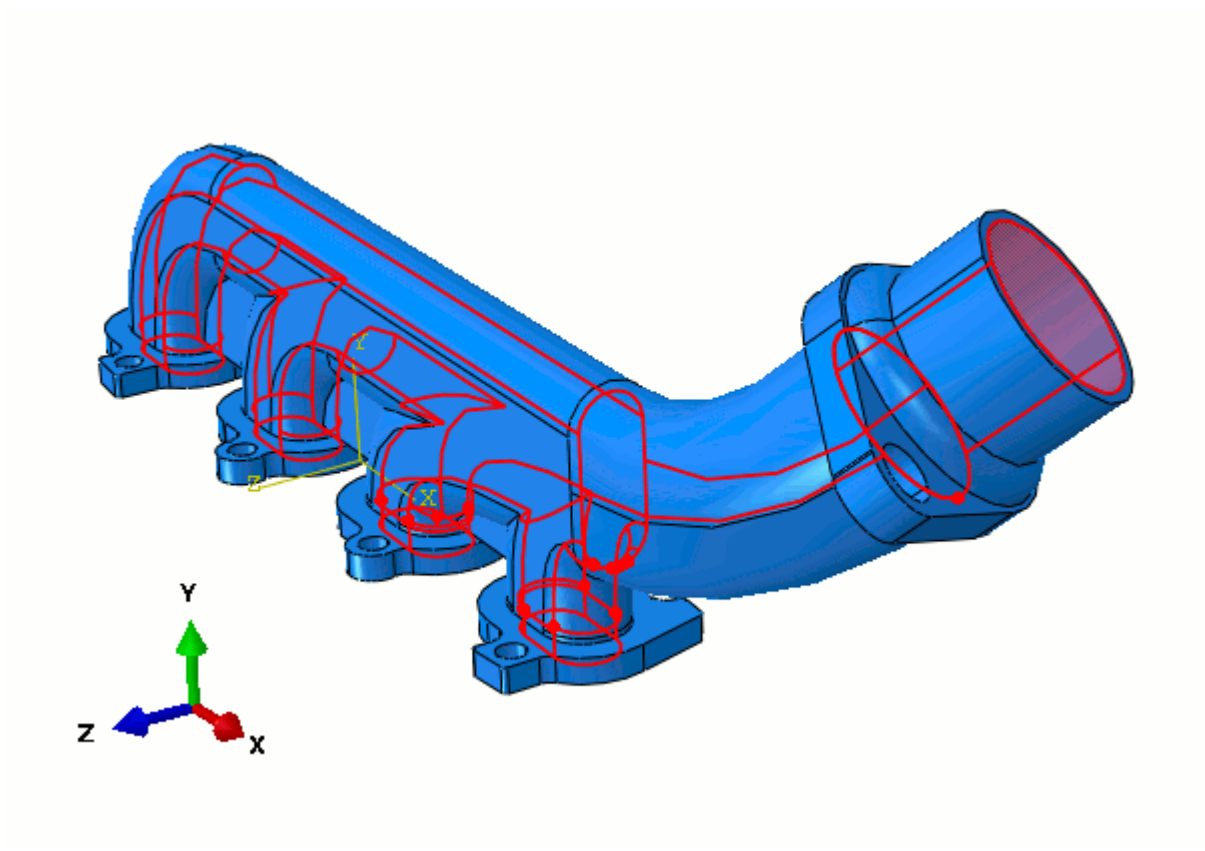
To create a surface:

1. Double-click **Surfaces** underneath the **Assembly** container in the Model Tree.
2. In the *Create Surface* dialog name the surface `INNER` and click **Continue...**



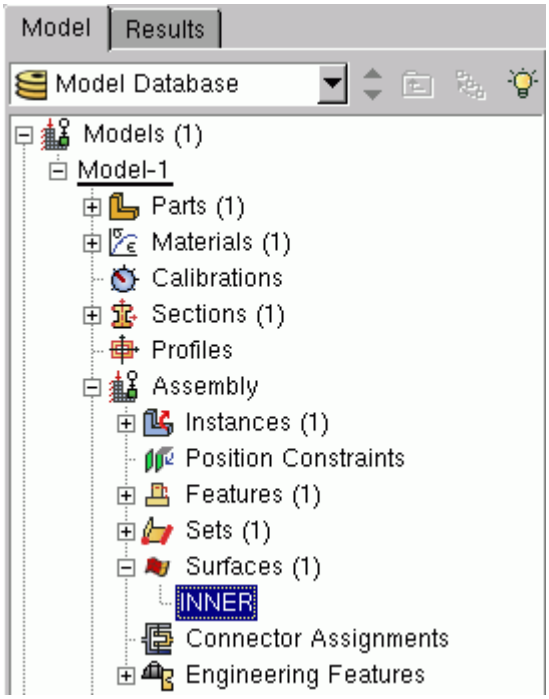
To select the inner faces of the model in the viewport:

3. At the bottom of the window, set *Select the regions for the surface* option to **by angle**.
4. Enter an angle of `89` and press `<Enter>`.
5. Hover the mouse pointer over one of the inner faces, and after a moment the inner surfaces are highlighted. Click to confirm this selection.

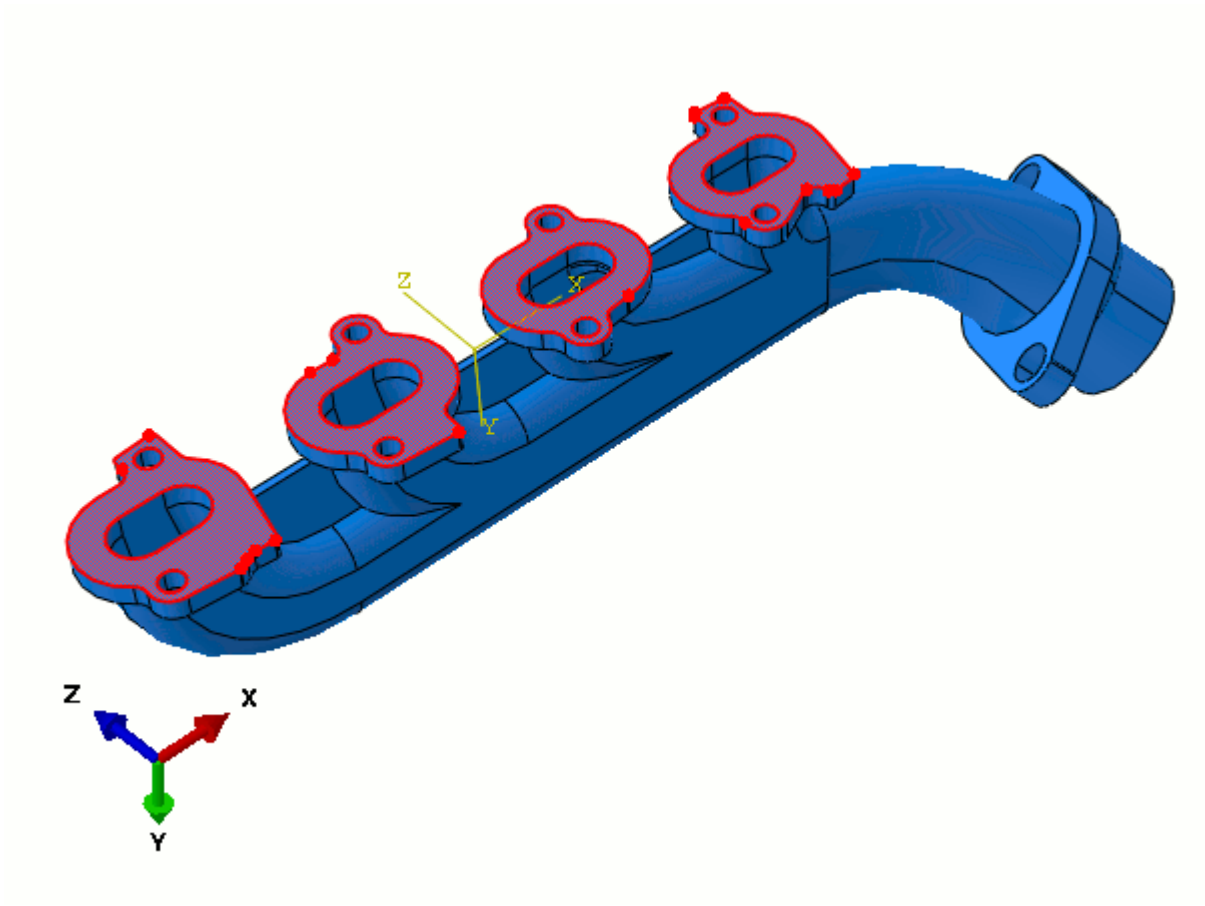


6. Once the inner faces have been highlighted, click **Done** at the bottom of the viewport.

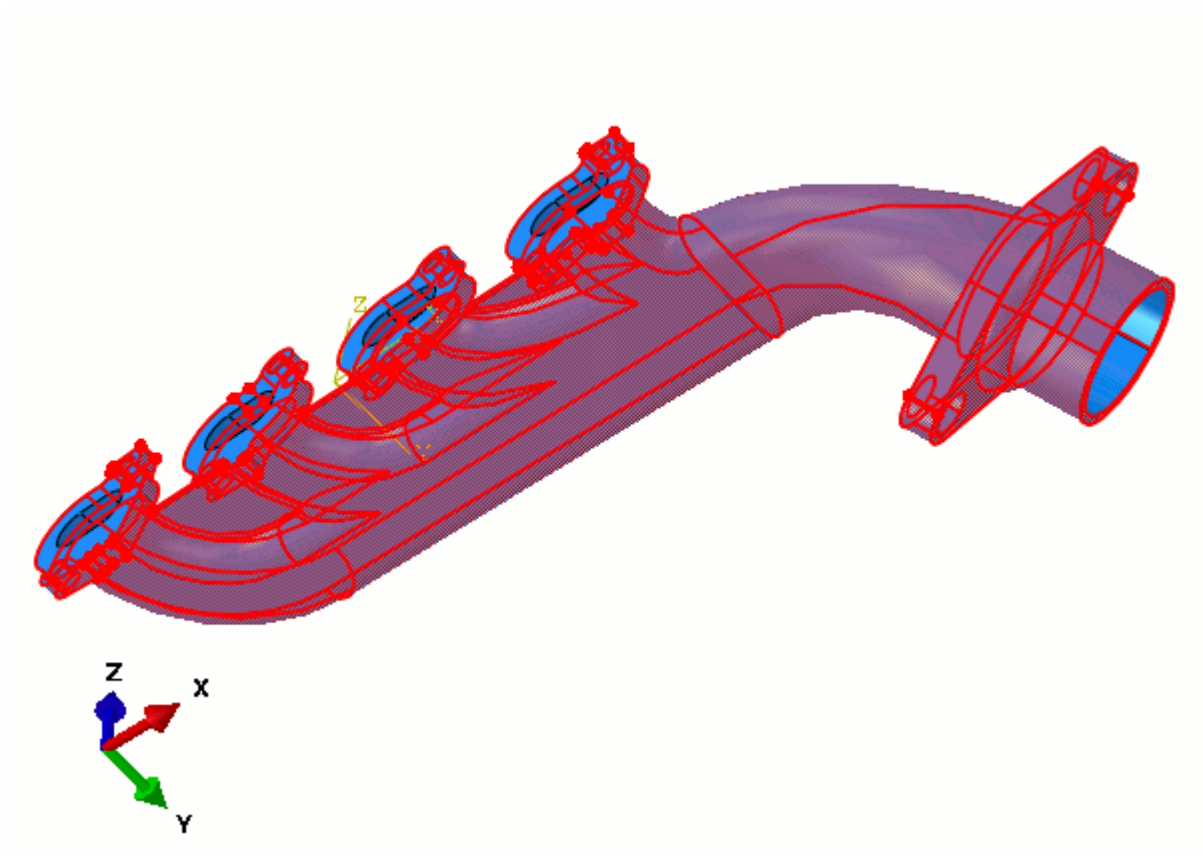
   The surface is shown under the **Surfaces** container.



7. Using the technique described above, create another surface called `FLANGES`.
8. In the viewport, rotate the manifold using `<Shift><Alt>` and left-click. Hold the `<Shift>` key and click the four flat faces on the bottom of the inlet flanges.



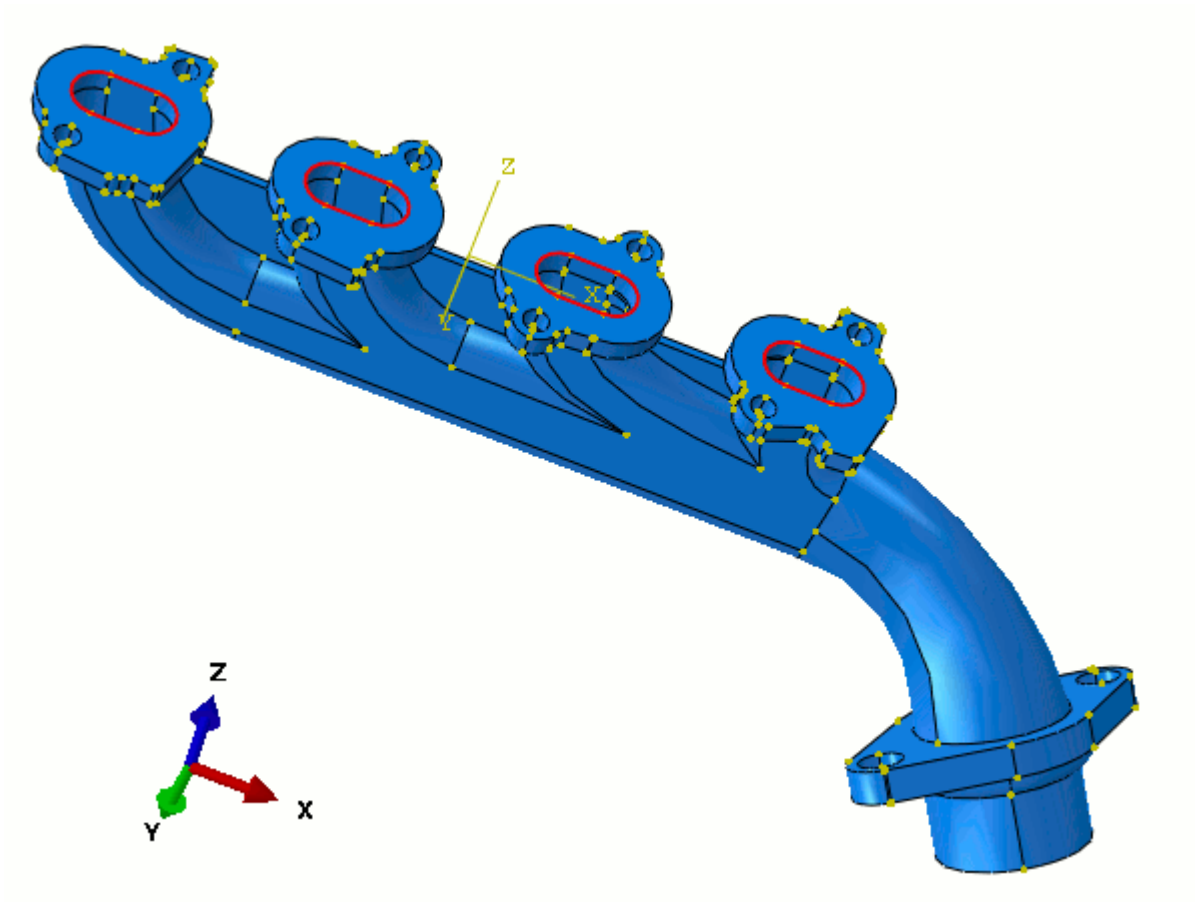
9. Click **Done** at the bottom of the viewport.
10. Finally, create a third surface called `EXTERIOR`.
11. In the viewport, use `<Shift>`-click to select all the exterior faces on the manifold part, including any face on the model that was not part of the FLANGES or INNER surfaces.



12. Click **Done** to create the surface.

Next, define two geometry sets:

13. Double-click **Sets** underneath the **Assembly** container.
14. In the *Create Set* dialog, name the set `INLET` and click **Continue**.
15. Select the continuous set of edges around each inlet as shown below.

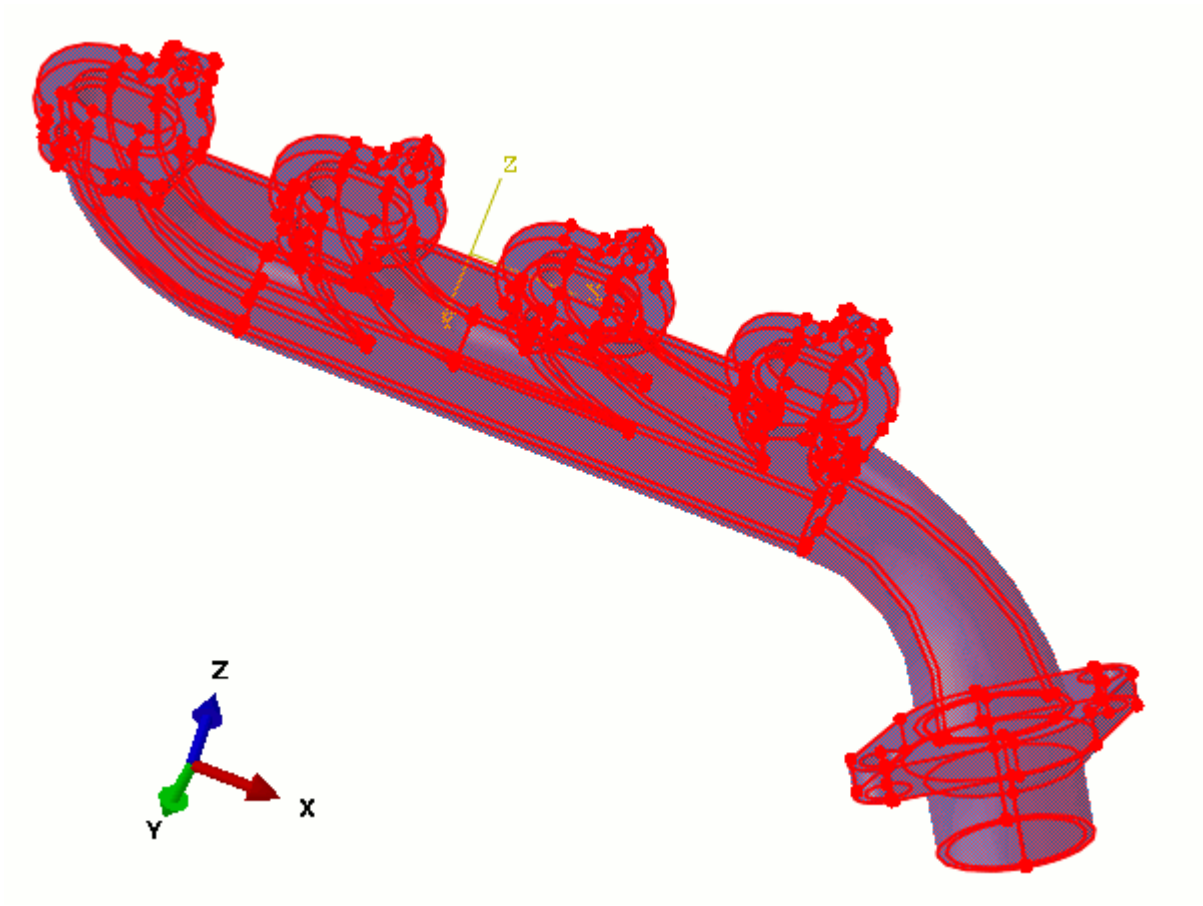


16. Click **Done** when you have selected these edges.

The second set includes the entire model and is used to define the initial temperature across the manifold:

17. Using the same technique as for the previous set, create another geometry set called `ALL`.
18. In the viewport, select the entire model by clicking in the space above and to the left of the model, and dragging the mouse down and to the right.

    A box is drawn around the model, and everything within the box is selected when you release the mouse button.



19. Click **Done** to confirm the set.

Defining the Analysis Steps

Define the heat transfer analysis step.

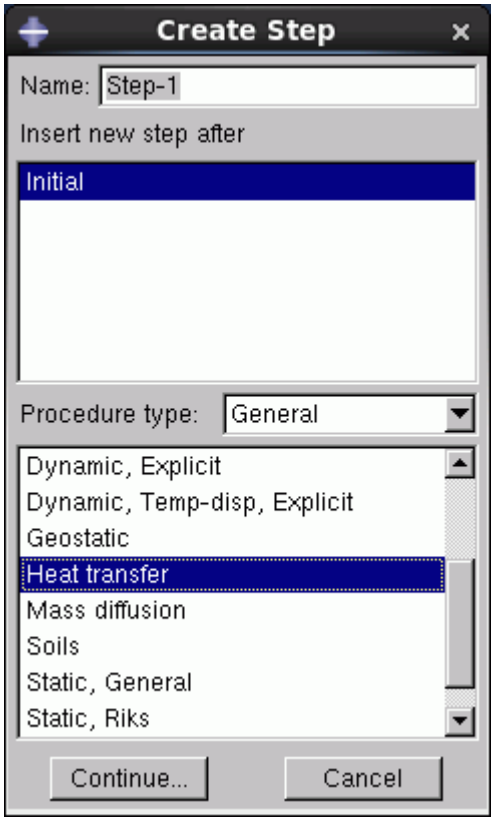
The analysis steps for the model include:

- An initial step, in which the initial temperature of the manifold is defined.
- A general heat transfer step, in which convection conditions are defined on the manifold exterior and flanges, and the temperature at the inlet is fixed at 500K. This step also includes the import of heat flux data from STAR-CCM+ via the SIMULIA Co-Simulation Engine.

By default, the Abaqus model already contains an initial step, which can be viewed by expanding the **Steps** container in the model tree.

To create the heat transfer step:

1. Double-click the **Steps** container in the model tree.
2. In the *Create Step* dialog, keep the default step name, `Step-1`, and select **Heat transfer** from the list.
3. Click **Continue...**



4. In the *Edit Step* dialog, set the following properties:

Tab	Property	Setting
Basic	*Description*	`Co-Simulation of Hot Exhaust Gases in the Manifold`
	Response	**Transient** (default)
	Time Period	`10000`
Incrementation	*Type*	**Fixed**
	Maximum number of increments	`2000`
	Increment size	`250`

5. Click **OK** to confirm these settings and create the step.

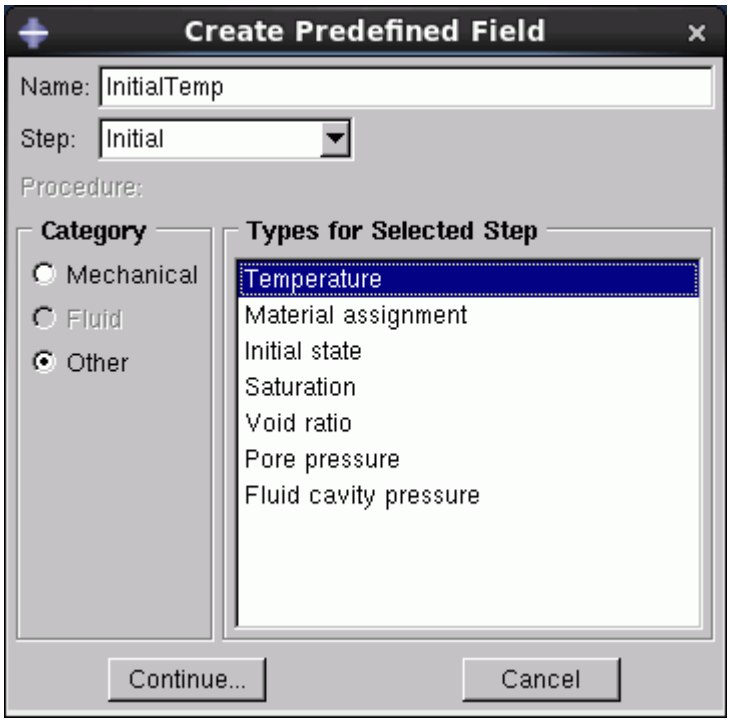
Defining Initial Conditions

Define the manifold initial temperature.

To specify the initial temperature through the manifold using a predefined field:

1. Double-click the **Predefined Fields** container in the Model Tree.
2. In the *Create Predefined Field* dialog, set the following properties:

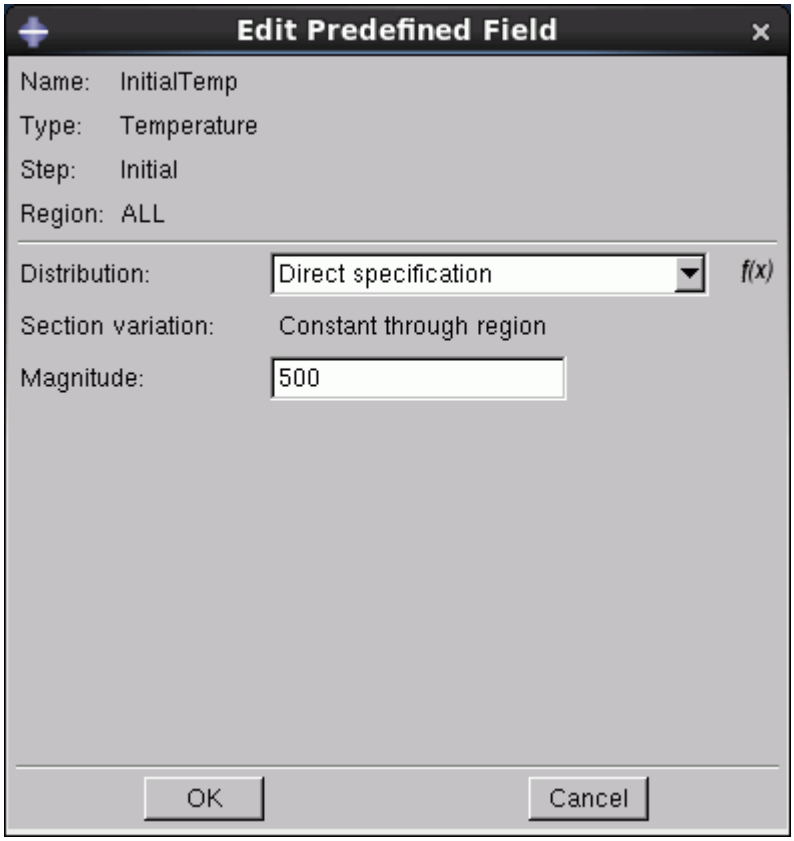
Property	Setting
Name	`InitialTemp`
Step	**Initial**
Category	**Other**
Types for Selected Step	**Temperature**



3. Click **Continue...**

This predefined field must now be applied to a region of the model, and in this case apply it to the whole model via the ALL geometry set:

4. Click **Sets** to the bottom right of the viewport.
5. In the *Region Selection* dialog, select **ALL** and click **Continue...**
6. In the *Edit Predefined Field* dialog, set **Magnitude** to `500` and click **OK**.



Defining Boundary Conditions

In the heat transfer analysis step, the convection conditions and fixed temperature at the inlet become active. Specify the convection properties as interactions, since the manifold transfers heat to the surroundings. Specify the temperature condition at the inlet as a boundary condition.

1. Double-click the **Interactions** container in the Model Tree.
2. In the *Create Interaction* dialog, set the following properties:

Property	Setting
Name	`SURFFILM`
Step	**Step-1**
Types for Selected Step	**Surface film condition**

3. Click **Continue...**
4. In the *Region Selection* dialog, select **EXTERIOR** and click **Continue...**

   (If the *Region Selection* dialog does not appear automatically, click **Surfaces...** at the bottom right of the *Viewport*).
5. In the *Edit Interaction* dialog, set the following properties:

Property	Setting
Film coefficient	`6.667`
Sink temperature	`300`

6. Click **OK**.
7. Repeat the steps above, this time using the following settings:

Property	Setting
Name	`SURFFILM-2`
Step	**Step-1** (default)
Type	**Surface film condition** (default)
Region	**FLANGES**
Film coefficient	`333.3`
Sink temperature	`500`

Specify the inlet temperature boundary condition next:

8. Double-click the **BCs** container in the Model Tree.
9. In the *Create Boundary Condition* dialog, set the following properties:

Property	Setting
Name	`Temp-BC`
Step	**Step-1** (default)
Types for Selected Steps	**Temperature** (default)

10. Click **Continue...**
11. In the *Region Selection* dialog, select **INLET** and click **Continue...**
12. In the *Edit Boundary Condition* dialog, set *Magnitude* to `500` and click **OK**.

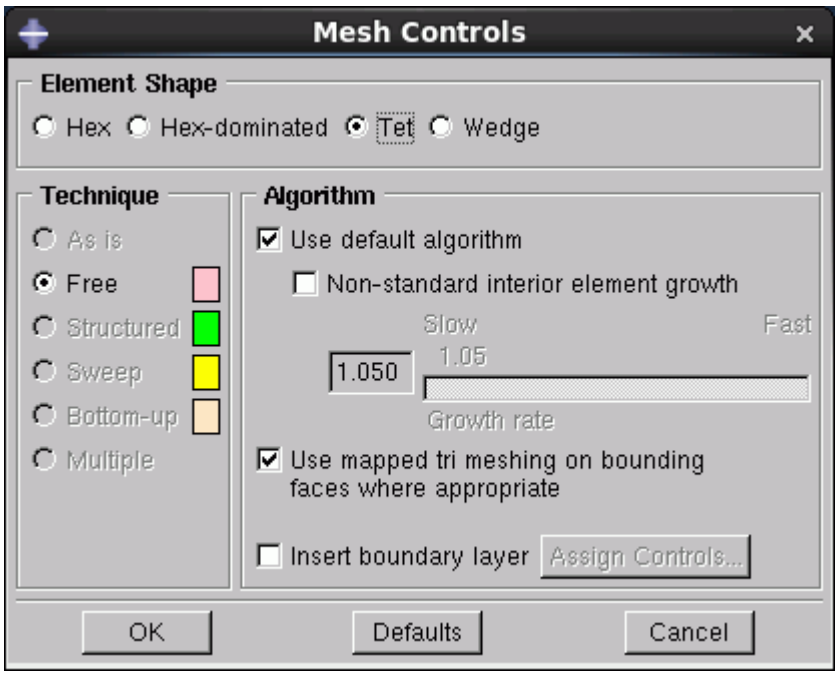
Meshing the Model

The next step involves defining and generating a mesh for the manifold part and is carried out within the Mesh module in Abaqus.

1. To switch to the Mesh module, expand the **Parts** > **manifold** container and double-click **Mesh (Empty)**.

The color of the manifold indicates which method Abaqus uses to mesh the part. Currently, most of the manifold is colored orange indicating that it cannot be meshed using the default element shape. Select a more appropriate element shape and use a free method to mesh the part:

2. Click (**Assign Mesh Controls**) in the mesh module toolbar or select **Mesh** > **Controls**.
3. In the viewport, hold the `<Shift>` key and select the main part of the manifold and the partitioned outlet section. When both parts are selected, click **Done**.
4. In the *Mesh Controls* dialog, change the *Element Shape* to **Tet**. Keep the default options and click **OK**.



5. Press `Esc` to exit the mesh controls tool.

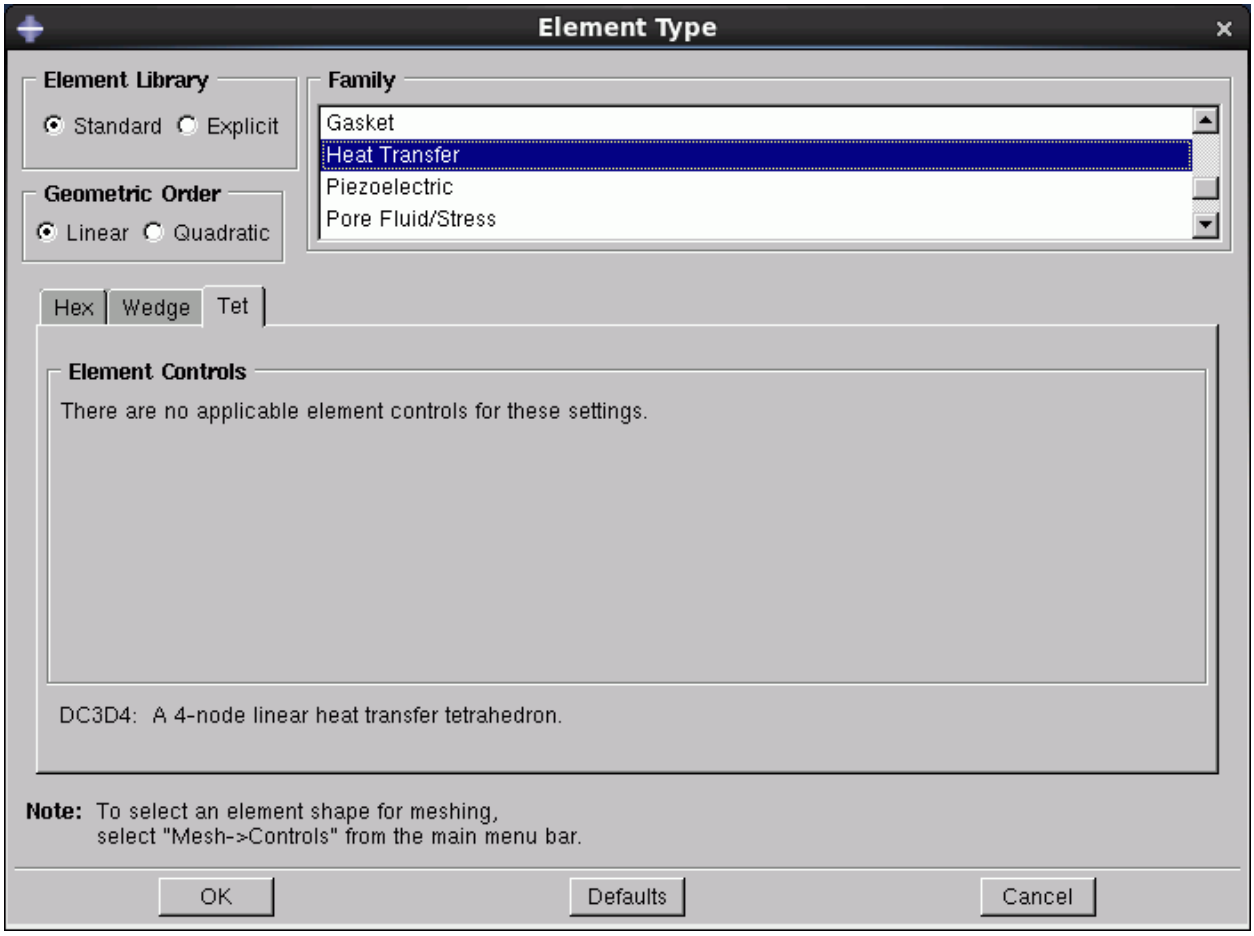
Choose the element type for the manifold part. In this case, 4-node linear tetrahedral elements are used:

6. Click (**Assign Element Type**) or select **Mesh** > **Element Type**.
7. In the same way as described previously, select both parts of the manifold in the viewport and click **Done**.
8. In the *Element Type* dialog, set the following properties:

Property	Setting
Element Library	**Standard** (default)
Geometric Order	**Linear**
Family	**Heat Transfer**

The *Tet* tab is selected by default. The element selection `DC3D4` is displayed at the bottom of the dialog.

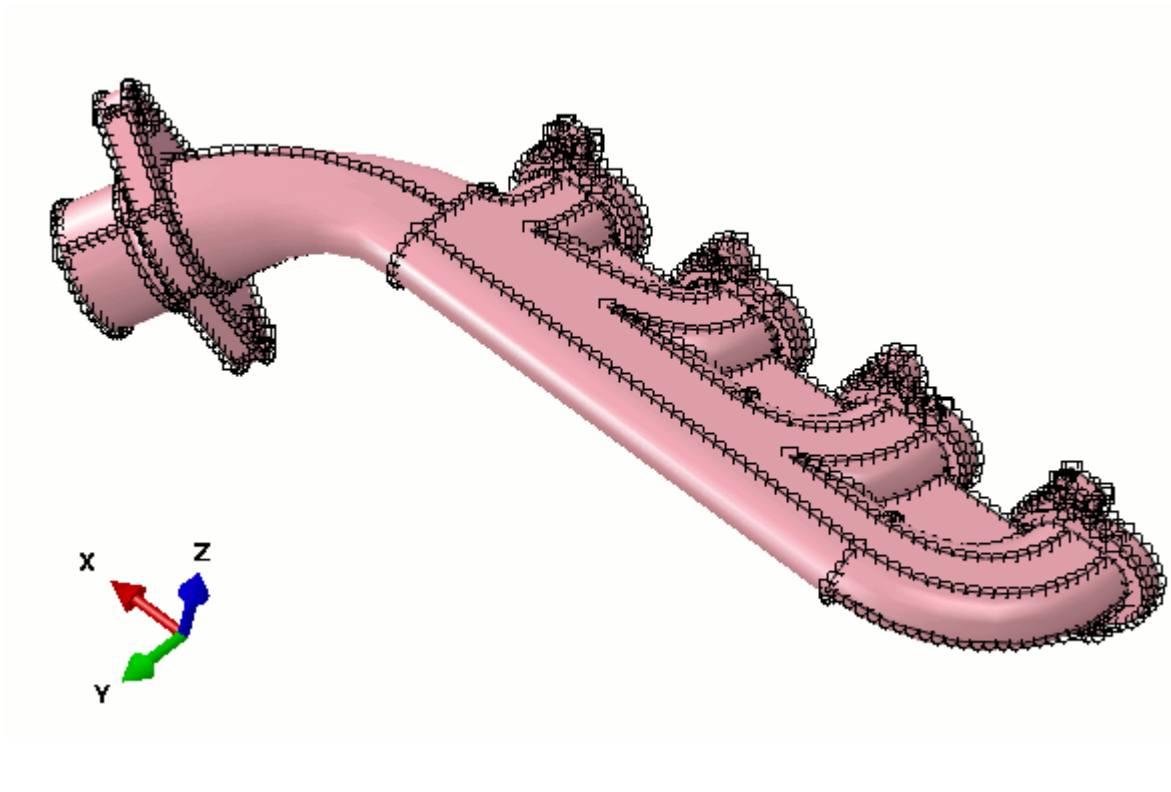
9. Click **OK**.



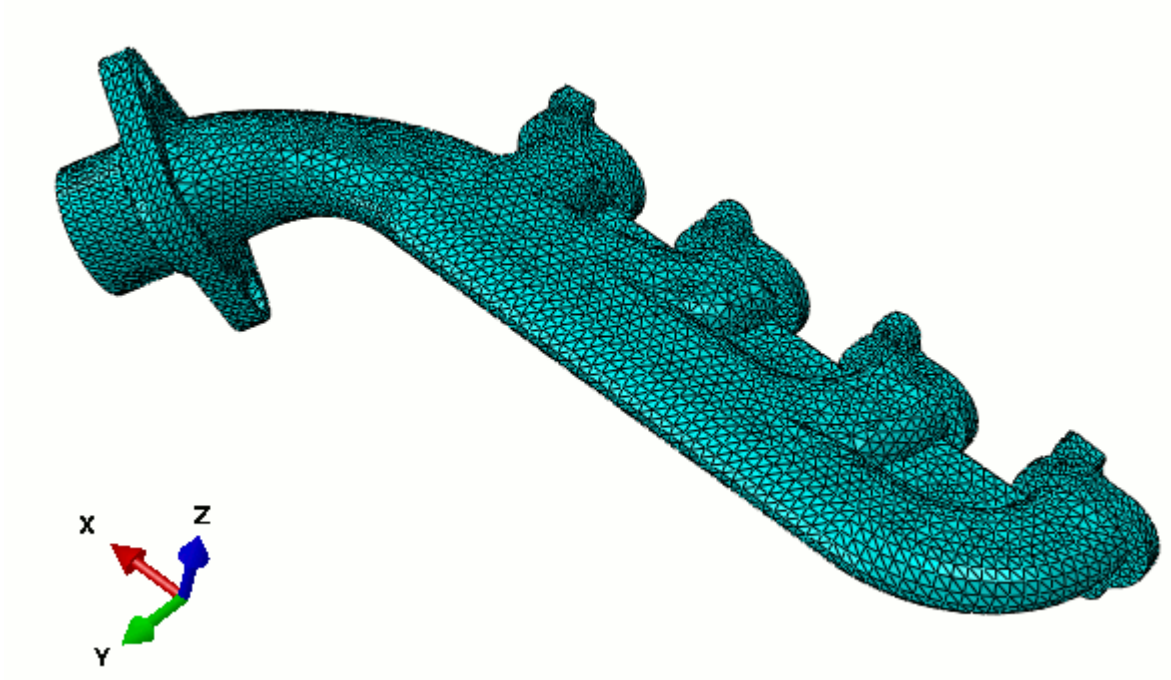
To determine the size of the mesh, specify a target global element size:

10. Click (**Seed Part**) or select **Seed** > **Part**.
11. In the *Global Seeds* dialog, set **Approximate global size** to `0.005`, then click **OK**.

    Seeds appear on each of the model edges.
12. Click **Done** at the bottom of the viewport to confirm the assignment of these seeds to the part.



13. Generate the mesh by clicking (**Mesh Part**) or by selecting **Mesh** > **Part**. Click **Yes** at the bottom of the viewport.



14. Save the model database by selecting **File** > **Save As**.

15. In the dialog that appears, navigate to the location to save the model, enter the name `manifold`, and click **OK**.

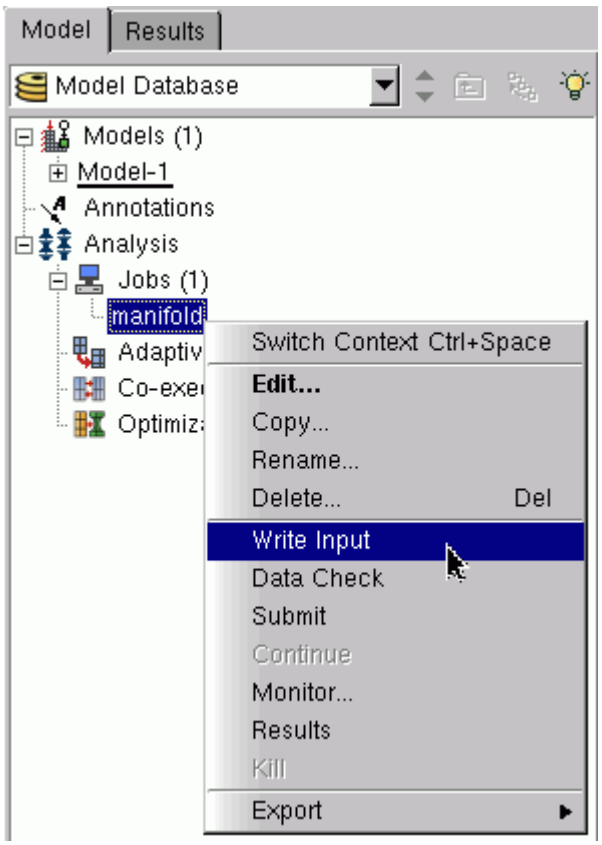
Creating the Abaqus Input File

The model preparation is now complete, and you can proceed to writing the input file.

Create an analysis job and the input file:

1. Double-click **Jobs** underneath the **Analysis** container.
2. In the *Create Job* dialog, set *Name* to `manifold`, then click **Continue...**
3. In the *Edit Job* dialog, set **Description** to `Exhaust manifold heat transfer`, then click **OK**.

   The job is added under the **Jobs** container.
4. Right-click the **manifold** job and select **Write Input**.



For the purposes of this tutorial, output history is not required. To generate the input file without requesting history output:

5. In the *warning* dialog, click **Yes**.

   The file `manifold.inp` is created in the working directory.
6. Click **Save** to save the model database.

Specifying Co-Simulation Settings in Abaqus

Add co-simulation commands to the Abaqus input file.

The final step in setting up the solid model in Abaqus/CAE was to create the input file, `manifold.inp`. This text-based file can be edited using a script editor. To prepare the Abaqus model for co-simulation, you add a set of commands to the input file to specify the Abaqus co-simulation step, the Abaqus coupled surface, and the fields that are imported and exported to/from this surface.

Note: For keyword lines containing many parameters, you can split the line at a comma. If the last character in a line is a comma, the next line is interpreted as a continuation of the line.

1. Open the `manifold.inp` file in a script editor of your choice.

The co-simulation is carried out during the heat transfer analysis step that you defined earlier.

2. Scroll down to the section that begins with the following:

```
**
** STEP: Step-1
**
```

This section of the file, up to `*End Step`, contains the history data for the Abaqus job, including solver parameters, boundary conditions, loads, and outputs.

3. To specify this step as the co-simulation step, add the following lines that are shown in bold text:

```
*Output, field, variable=PRESELECT
*Output, history, frequency=0
**
*CO-SIMULATION, NAME=FSI_Thermal, PROGRAM=MULTIPHYSICS
**
*End Step
```

This line specifies that co-simulation is carried out in this step, and defines the co-simulation name, `FSI_Thermal`.

4. To identify the interfacing surface and the fields that are imported and exported during the co-simulation, add the following lines:

```
*Output, field, variable=PRESELECT
*Output, history, frequency=0
**
*CO-SIMULATION, NAME=FSI_Thermal, PROGRAM=MULTIPHYSICS
*CO-SIMULATION REGION, TYPE=SURFACE, EXPORT
ASSEMBLY_INNER, NT
*CO-SIMULATION REGION, TYPE=SURFACE, IMPORT
ASSEMBLY_INNER, CFL
**
*End Step
```

The first line added here indicates that you are identifying the region in the model, in this case a surface, from which data are exported. In the next line, the surface is identified by entering the name of the assembly in which it is defined, `Assembly`, and then the name of the surface, `Inner`. The syntax of the surface identifier is `<ASSEMBLY NAME>_<SURFACE NAME>`. The next line defines the export field as the nodal temperature, `NT`. The next two lines use the same format to specify that heat flux, `CFL`, are imported into Abaqus.

5. Save the file as `manifold-mod.inp`.

Preparing the STAR-CCM+ Model

Set up the model in STAR-CCM+ for the fluid inside the manifold.

For this tutorial, you are provided with a simulation file that contains pre-defined objects. To load the initial simulation:

1. Navigate to `[INSTALL_DIR]/doc/startutorialsdata/couplingWithCaeCodes/data` and copy the starting simulation file, `manifold-gas_start.sim`, into your `/thermalCoSimTutorial/starccm+/` directory.
2. Launch STAR-CCM+.
3. Select **File** > **Load Simulation...**
4. In the *Load Simulation* dialog, click **Browse...**
5. In the *Open* dialog, navigate to `/thermalCoSimTutorial/starccm+/`.
6. Select `manifold-gas_start.sim`, then click **Open**.
7. In the *Load Simulation* dialog, click **OK**.
8. Save the simulation as `abaqus_ThermalCoSim.sim`.

Contents:

Generating the Volume Mesh
Selecting Physics Models
Defining Boundary Conditions
Setting Up a Monitor
Setting Solver Variables
Setting Stopping Criteria
Creating a Fluid Velocity Scalar Scene
Creating a Fluid Temperature Scalar Scene
Specifying Co-Simulation Settings

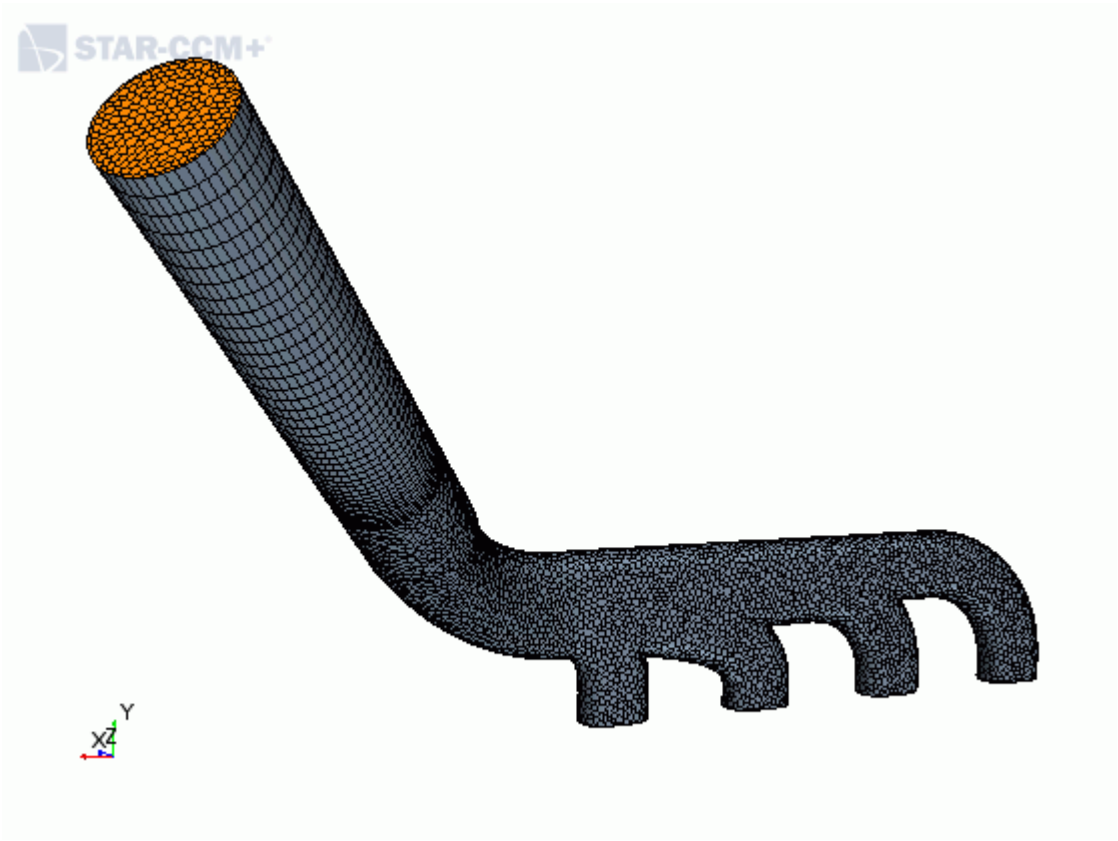
Generating the Volume Mesh

The starting simulation file contains predefined mesh operations, which generate a polyhedral volume mesh extruded at the manifold outlet.

1. Click (**Generate Volume Mesh**).

The output from the meshers is shown in the *Output* window and the mesh generation status is shown in the status bar.

2. To visualize the volume mesh, create a mesh scene.



3. Save the simulation.

Selecting Physics Models

A physics continuum was created automatically at the end of the mesh generation process. Select physics models for this continuum and make sure that it is assigned to the Fluid region.

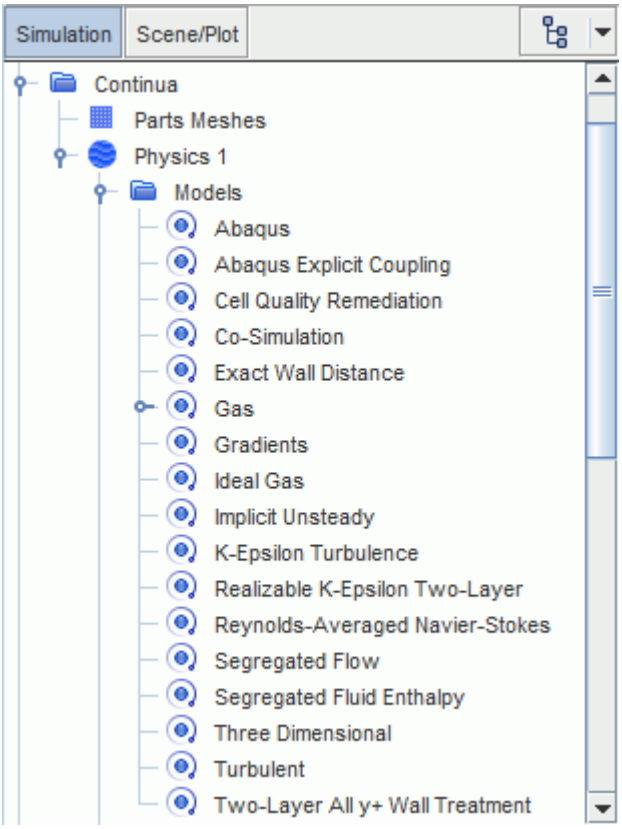
1. For the physics continuum, **Continua** > **Physics 1**, select the following physics models in order:

Group box	Model
Enabled Models	**Three Dimensional** (Selected automatically)
Time	**Implicit Unsteady**
Material	**Gas**
Flow	**Segregated Flow**
Enabled Models	**Gradients** (Selected automatically)
Equation of State	**Ideal Gas**

Group box	Model
Energy	**Segregated Fluid Enthalpy**
Viscous Regime	**Turbulent**
	Reynolds-Averaged Navier-Stokes (Selected automatically)
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**
	Realizable K-Epsilon Two-Layer (Selected automatically)
	Two-Layer All y+ Wall Treatment (Selected automatically)
	Exact Wall Distance (Selected automatically)
Optional Models	**Cell Quality Remediation**
Optional Models	**Co-Simulation**
Co-Simulation Models	**Abaqus**
Abaqus Coupling Models	**Abaqus Explicit Coupling**

Then close the *Physics 1 Model Selection* dialog.

2. To review the models, open the **Physics 1** > **Models** node.



Defining Boundary Conditions

Specify the fluid temperature, fluid velocity, and the turbulence settings at the manifold inlets.

The gas entering the manifold is traveling at 10 m/s with a temperature of 500 K. Specify these boundary conditions under the inlet boundary node, as well as selecting a different turbulence specification.

To set the boundary conditions:

1. Edit the **Regions** > **Fluid** > **Boundaries** > **Inlets** node.
2. Set the following properties:

Node	Property	Setting
Physics Conditions		
└ **Turbulence Specification**	*Method*	**Intensity + Length Scale**
Physics Values		
└ **Static Temperature**	*Value*	`500 K`
└ **Velocity Magnitude**	*Value*	`10 m/s`

3. Save the simulation.

Setting Up a Monitor

By default STAR-CCM+ produces a residuals plot which can be used to monitor the convergence of the solution. It is useful to create other monitors to help in assessing convergence of the solution. Monitor the temperature at a point inside the fluid domain, close to the solid-fluid interface.

1. Right-click the **Derived Parts** node and create a new point probe with the following settings:

Property	Setting
Input Parts	**[Fluid, Surface Extruder]** (default)
Point	`[0.251, 0.107, -0.0405] m`
Display	**No Displayer**

2. Click **Create** then **Close**.

Use this point to monitor the temperature in the fluid. Before you can create the monitor, first create a report. In this case use a Maximum report:

3. Right-click the **Reports** node and select **New Report** > **Maximum**.
4. Rename the **Reports** > **Maximum 1** node to `Temperature Point`.

Reports can be created for all derived parts and boundaries. In this case, select the point:

5. Edit the **Reports** > **Temperature Point** node and set the following properties:

Property	Setting
Field Function	**Temperature**
Parts	**Derived Parts** > **Point**
Smooth Values	**Activated**

The report settings are now complete and you can proceed by creating a monitor and plot from the report.

6. Right-click the **Temperature Point** node and select **Create Monitor and Plot from Report**.

   The monitor and plot are named **Temperature Point Monitor** and **Temperature Point Monitor Plot**, respectively.

Setting Solver Variables

Set the solver parameters to specify the simulation time-step and to display detailed output about data exchanges with Abaqus.

Response times to changes in the temperature in the solid and fluid domains are relatively slow, so specify a large time-step. In this co-simulation, data is exchanged between STAR-CCM+ and Abaqus at each time-step.

1. Edit the **Solvers** node and set the following properties:

Node	Property	Setting
Implicit Unsteady	*Time-Step*	`250 s`
Abaqus Co-Simulation	*Verbosity*	**High**

2. Save the simulation.

Setting Stopping Criteria

Set the maximum physical time and create an additional stopping criterion from the temperature point monitor.

In this tutorial, the co-simulation converges within 40 time steps, so use the maximum physical time to impose this limit:

1. Select the **Stopping Criteria** > **Maximum Physical Time** node and set *Maximum Physical Time* to `10000 s`.

Note: Note: the maximum physical time does not need to match the Time Period setting in Abaqus. During the co-simulation, you can terminate the run using either application, depending on your specified stopping criteria.

In the early stages of the simulation, you can expect the number of inner iterations STAR-CCM+ requires to be much higher, since STAR-CCM+ is effectively solving the initial steady-state solution in the fluid. However, as the simulation progresses, the number of inner iterations per time step reduces, as the solution converges much more quickly. To accelerate the progress of the co-simulation, either decrease the number of inner iterations manually (by observing the temperature monitor and adjusting the **Maximum Inner Iterations** stopping criterion) or use an additional stopping criterion. The latter technique is described.

Create a stopping criterion from a temperature monitor such that when the temperature solution has converged for each time step, STAR-CCM+ stops iterating. Increase the number of iterations for the Maximum Inner Iterations stopping criterion so that enough iterations are carried out in the initial steps:

2. Copy the **Monitors** > **Temperature Point Monitor** node and paste it onto the **Monitors** node.
3. Rename **Temperature Point Monitor 2** to `Temperature Point Monitor_Iteration`.
4. Select the **Temperature Point Monitor_Iteration** node and set *Trigger* to **Iteration**.
5. Right-click the **Temperature Point Monitor_Iteration** node and select **Create Stopping Criterion from Monitor**.

   A new node named **Temperature Point Monitor_Iteration Criterion** is created under the **Stopping Criteria** node.

The temperature solution can be said to have converged when the temperature variations per iteration have dropped below 0.05 K. To specify an asymptotic stopping criterion:

6. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Inner Iterations	*Maximum Inner Iterations*	`60`
Temperature Point Monitor_Iteration Criterion	*Criterion Option*	**Asymptotic**
└ **Asymptotic Limit**	*Normalized*	**Deactivated**
	|*Max-Min*|	`0.05 K`

7. Save the simulation.

Creating a Fluid Velocity Scalar Scene

Use a scalar scene to display the velocity magnitude on a plane section of the manifold.

1. Create a scalar scene.
2. Rename the **Scenes** > **Scalar Scene 1** node to `Fluid Velocity`.
3. Right-click the **Derived Parts** node and create a new plane section with the following settings:

Property	Setting
Input Parts	**[Fluid, Surface Extruder]** (default)
origin	`[0, 0, 0] m`
normal	`[0, 0, 1] m`
Display	**No Displayer**

4. Click **Create** then **Close**.
5. Edit the **Scenes** > **Fluid Velocity** > **Displayers** > **Outline 1** > **Parts** node.
6. In the *selection* dialog, deselect **Regions**, then select **Derived Parts** > **Plane Section**.
7. Click **OK**.
8. Edit the **Scenes** > **Fluid Velocity** > **Displayers** node.

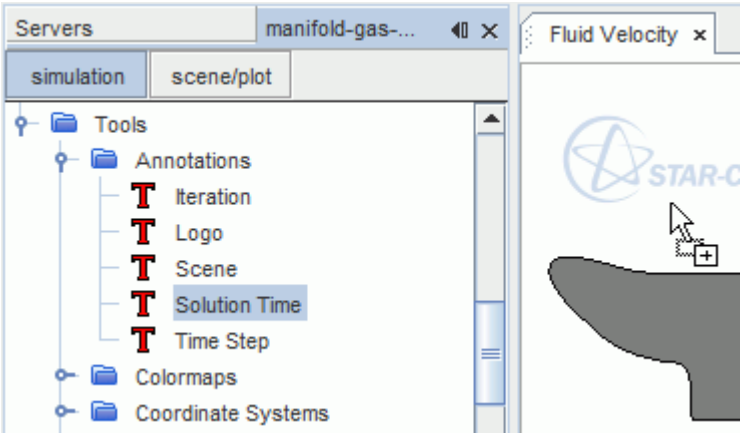
9. Set the following properties:

Node	Property	Setting
Scalar 1	*Contour Style*	**Smooth Filled**
└ **Parts**	*Parts*	**Derived Parts** > **Plane Section**
└ **Scalar Field**	*Function*	**Velocity** > **Magnitude**

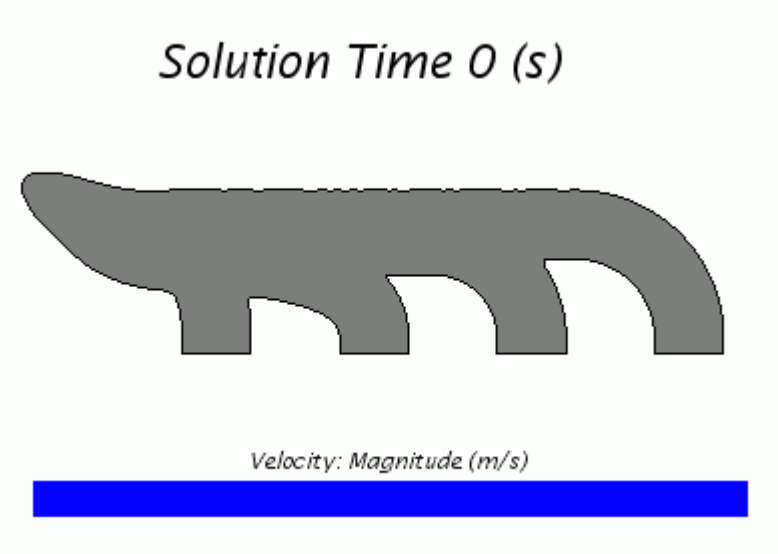
10. Click (**Save-Restore-Select views**) and select **Views** > **-Z** > **Up +Y**. Use the mouse to zoom in on the section and position it in the center of the scene.

To display the solution time in the scalar scene, add an annotation:

11. Expand the **Tools** > **Annotations** node. Select the **Solution Time** node and drag it into the scalar scene. Release the mouse button to add the annotation to the scene.



12. Click the annotation and drag it into a position as shown below.



13. Save the simulation.

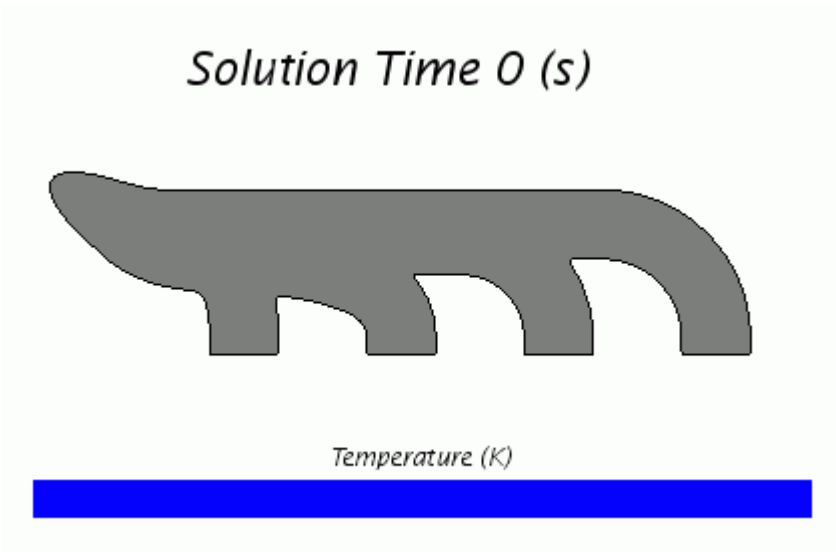
Creating a Fluid Temperature Scalar Scene

Create a second scalar scene to display the fluid temperature on the plane section.

Since many of the settings are the same as for the Fluid Velocity scene, use a duplicate of this scene as the starting point.

1. Copy the **Scenes** > **Fluid Velocity** node and paste it onto the **Scenes** node.
2. Rename the **Copy of Fluid Velocity** node to `Fluid Temperature`.
3. Select the **Scenes** > **Fluid Temperature** > **Displayers** > **Scalar 1** > **Scalar Field** node and set *Function* to **Temperature**.

   The scene appears as shown below.



4. Save the simulation.

Specifying Co-Simulation Settings

Prepare the simulation for co-simulation with Abaqus.

In STAR-CCM+, the properties of the co-simulation are defined within a co-simulation link. This includes details of the location of the Abaqus input file, the boundary defining the fluid-structure interface, and the fields that are exchanged at each coupling step.

STAR-CCM+ automatically adds a co-simulation link, **Co-Simulations** > **Link 1**, when you activate an Abaqus Co-Simulation model.

If you skipped the section of the tutorial involving the preparation of the Abaqus input file, you can find a copy of the `manifold-mod.inp` file in the STAR-CCM+ installation. Before proceeding:

- Navigate to `[INSTALL_DIR]/doc/startutorialsdata/couplingWithCaeCodes/data` and copy the `manifold-mod.inp` file into your `/thermalCoSimTutorial/abaqus/` directory.

1. Edit the **Co-Simulations** > **Link 1** > **Values** node, and set the following properties:

Node	Property	Setting
Abaqus Execution	*Current Job Name*	`co-simulation_manifold`
	Input File	`../abaqus/manifold-mod.inp`
	Executable Name	name you use to launch Abaqus, for example `abaqus`
Constant Coupling Time Interval	*Value*	`250 s`

   To specify the location of the input file, you type the path directly into the box. For example, if your working directory is `/thermalCoSimTutorial/starccm+/` and your input file is contained in the folder `/thermalCoSimTutorial/abaqus/`, the path to the input file would be `../abaqus/manifold-mod.inp`.

To specify the fluid wall as the coupled boundary:

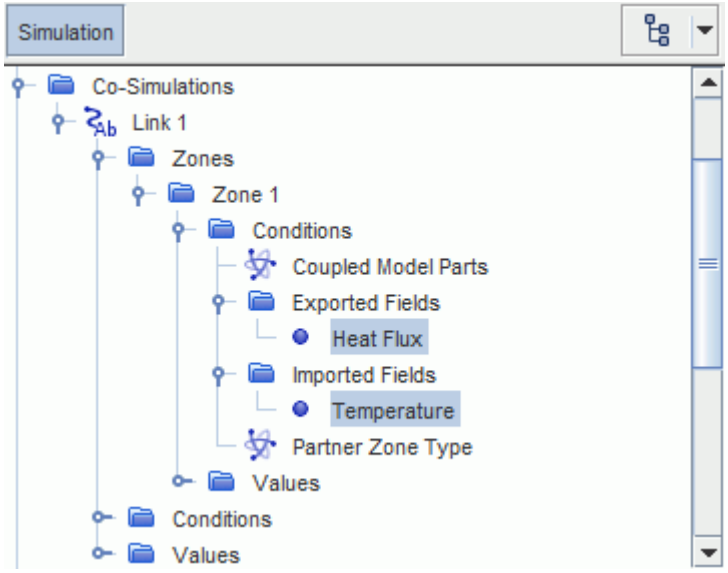
2. Expand the **Regions** > **Fluid** > **Boundaries** > **Fluid-Wall** node.
3. Edit the **Physics Conditions** > **External Code Coupling Specification** node and set *External Code Coupling Specification* to **Link 1: Zone 1**.
4. Expand the **Co-Simulations** > **Link 1** > **Zones** > **Zone 1** > **Conditions** node.

   STAR-CCM+ automatically sets the *Boundaries* property for the **Coupled Model Parts** node to the specified coupled boundaries.

STAR-CCM+ automatically defines import and export specifications based on the simulation settings. When the fluid continuum contains a flow model, STAR-CCM+ adds **Pressure** to the zone **Exported Fields**. When the fluid continuum contains an energy model, STAR-CCM+ adds boundary **Heat Flux** to the **Exported Fields** and **Temperature** to the **Imported Fields**.

Modify the default settings so that STAR-CCM+ only exports boundary heat flux data to Abaqus:

5. Right-click the **Zones** > **Zone 1** > **Conditions** > **Exported Fields** > **Pressure** node and select **Delete**.



By default, STAR-CCM+ automatically sets the *Thermal Specification* for the Fluid-Wall boundary to **Temperature** and applies the imported temperature data to the boundary.

6. Save the simulation.

Running the Co-Simulation

You are now ready to run the co-simulation.

During this process, STAR-CCM+ automatically launches Abaqus in the background and starts communicating with Abaqus via the SIMULIA Co-Simulation Engine.

1. Run the simulation.

   The co-simulation libraries are loaded and the co-simulation settings registered. The corresponding messages are displayed in the *Output* window.

   The iteration progress can be monitored in the *Output* window. Monitor the convergence by viewing the Temperature Point Plot and the Residuals in the *Graphics* window. You can also view the *Fluid Velocity* and *Fluid Temperature* scenes to visualize the solution while the co-simulation is running.

   At the end of each coupling step, data is exchanged between the two codes, and details are displayed in the *Output* window.

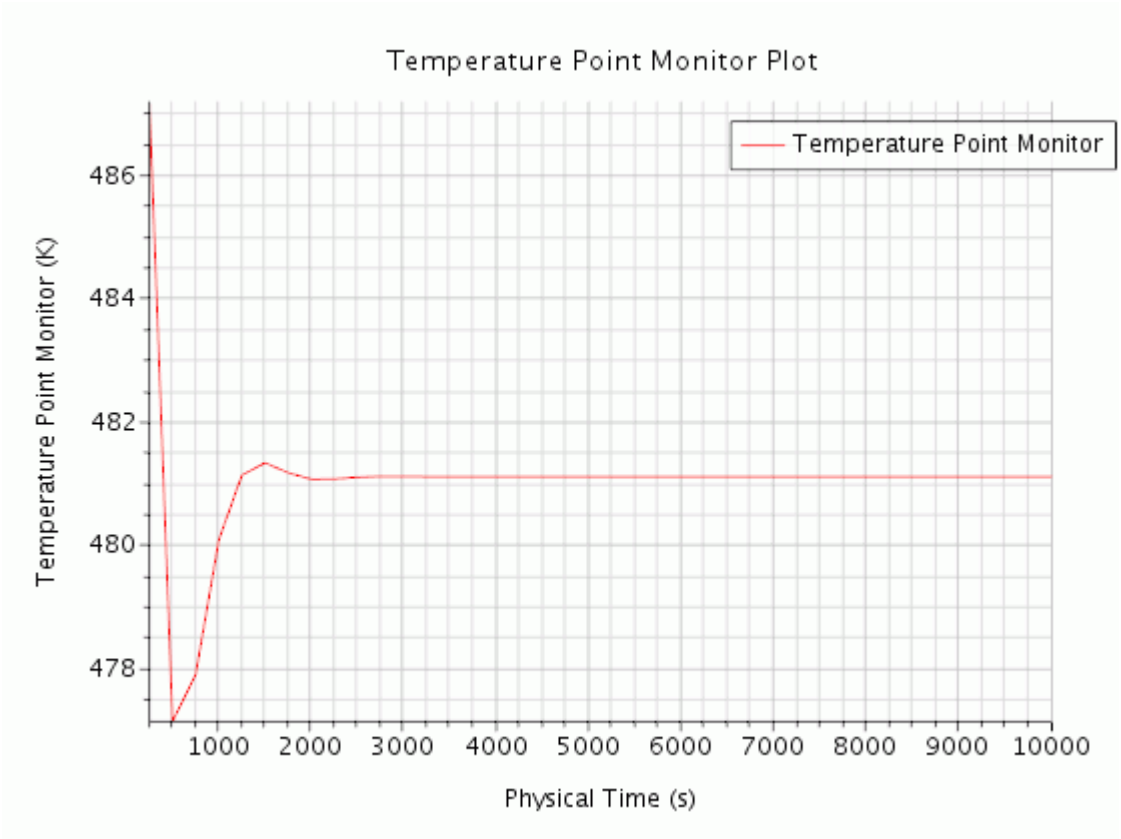
2. When the co-simulation is complete, save the simulation.

Visualizing the Results

Visualize the fluid velocity and temperature.

The results for the fluid solution can be observed by examining the solution that is displayed on the plane section in the *Fluid Velocity* and *Fluid Temperature* scalar scenes. To display each scene in the Graphics window, click the relevant tab at the top of the window.

You can see from the Temperature Point Plot that the fluid-solid solution has converged. Each time data was exchanged via the SIMULIA Co-Simulation Engine, that is, at each coupling step, the temperature at the point “jumped” in response to the imported temperature data. You can observe that as the temperature difference at the fluid-solid interface tended towards zero, the temperature at the monitor point close to the fluid wall converged.



The results for the solid solution can be post-processed by opening the file `/thermalCoSimTutorial/abaqus/co-simulation_manifold.odb` in Abaqus/CAE.

Summary

This tutorial demonstrated how to run a thermal co-simulation case using STAR-CCM+ and Abaqus.

This tutorial has introduced the following STAR-CCM+ features:

- Setting up a solid model in Abaqus/CAE, including:
  - importing the geometry
  - defining material properties
  - defining surfaces
  - specifying initial and boundary conditions
  - generating a mesh
  - creating the input file

- Setting up a fluid model in STAR-CCM+, including:
  - selecting physics models for thermal co-simulation
  - defining boundary conditions
  - defining a stopping criterion from a monitor
- Running a co-simulation, including:
  - editing the Abaqus input file for co-simulation
  - specifying co-simulation settings in STAR-CCM+
  - running the co-simulation

Abaqus Co-Simulation: Mechanical Coupling

This tutorial demonstrates how to carry out a mechanical co-simulation case using STAR-CCM+ and Abaqus. Co-simulation involves a strong coupling between the two codes. Data is exchanged at frequent intervals that are called coupling steps. This level of communication between the solvers allows you to obtain a full solution across the fluid-solid interface.

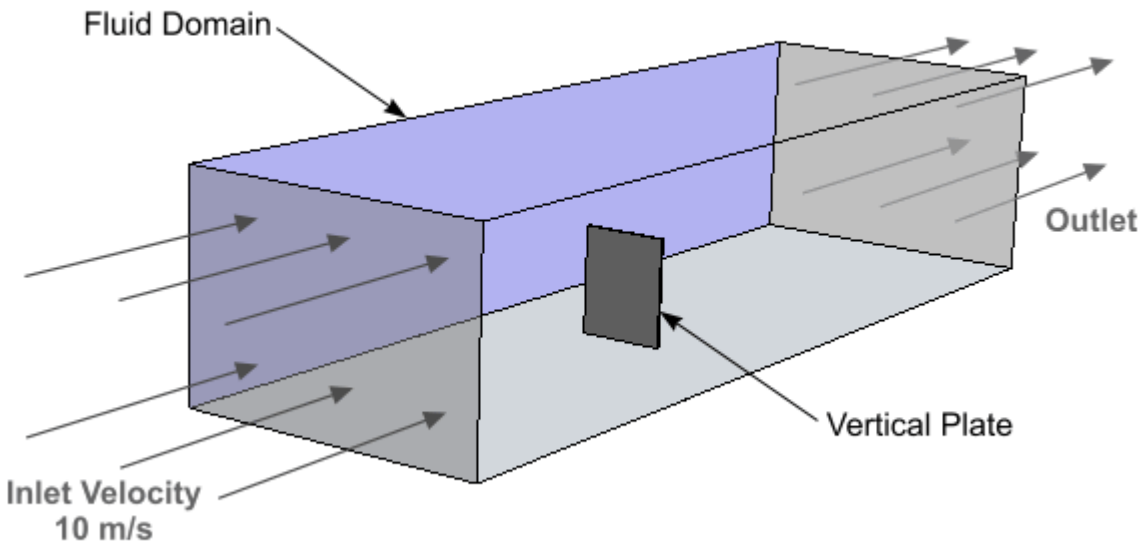
In a mechanical coupling, STAR-CCM+ passes traction loads to Abaqus (pressure + wall shear stress), and Abaqus passes displacements to STAR-CCM+. In Abaqus, the traction loads are applied to the surface of the solid structure. In STAR-CCM+, the displacements are used as an input to the mesh morpher.

During the co-simulation, data is automatically exchanged between STAR-CCM+ and Abaqus using the SIMULIA Co-Simulation Engine. This technique differs from the file-based method in that the coupling between the codes is much stronger, and STAR-CCM+ and Abaqus run simultaneously.

To set up a co-simulation case, separate models must be prepared for the fluid and the solid domains. The solid model is set up in Abaqus, and defined within an Abaqus input file. The fluid model is set up in a STAR-CCM+ simulation where the appropriate boundary conditions and physics are defined. For each model, a set of co-simulation parameters must be specified. The parameters identify the boundaries/surfaces that make up the fluid-structure interface, the field data that is imported/exported at the interface, the frequency of coupling steps, and so on. STAR-CCM+ is used to run the co-simulation when the setup is complete.

Before running the co-simulation, it is good practice to run standalone cases for the fluid and solid models to make sure that they converge independently. In this example, the Abaqus model could be run standalone using an applied pressure load, *DSLOAD, on the plate surface with an estimated constant value for the pressure. The STAR-CCM+ simulation could be run without mesh morphing, using a fixed plate.

This tutorial simulates an elastic flap that is pinned at the base, and then subjected to a 10 m/s flow of air. The mass and stiffness of the plate are such that the first bending mode has a frequency near 4 Hz and the first twisting mode has a frequency near 20 Hz. With a fixed plate, the vortex shedding at a 10 m/s inlet velocity is near 20 Hz, so you expect resonance between the vortex shedding and the first twisting mode.



In this tutorial, the fluid domain is initialized by running a quasi-steady simulation. The initial flow field is then used as the initial condition for the co-simulation. Once the initial flow field is established, the solver settings are changed to the values that are required for the co-simulation. When the co-simulation is run, the plate is released and bends in the wind. In the tutorial, the co-simulation is run for 0.125s, which is about half the fundamental period of the first bending mode and about 2.5 vortex shedding periods. Running the case for longer (around 4 s) shows that the first bending mode is damped, and the first twisting mode dominates plate motion.

Contents:

Prerequisites
Obtaining Input Files
Reviewing the Abaqus Model
Preparing the STAR-CCM+ Model
Obtaining an Initial Flow Field
Running the Co-Simulation
Summary

Prerequisites

The instructions in the Abaqus Mechanical Coupling tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Techniques	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

This tutorial demonstrates co-simulation with Abaqus 2016. For a list of all supported versions of Abaqus, refer to the section *External Packages Support* of the STAR-CCM+ Release Notes.

Obtaining Input Files

Copy the files that are required to complete the tutorial into your working directory.

When running a co-simulation case, it is good practice to create separate directories for the STAR-CCM+ and Abaqus files:

1. Create a directory called `/mechanicalCoSimTutorial`.
2. Within this directory, create two more directories: `/abaqus` and `/starccm+`.

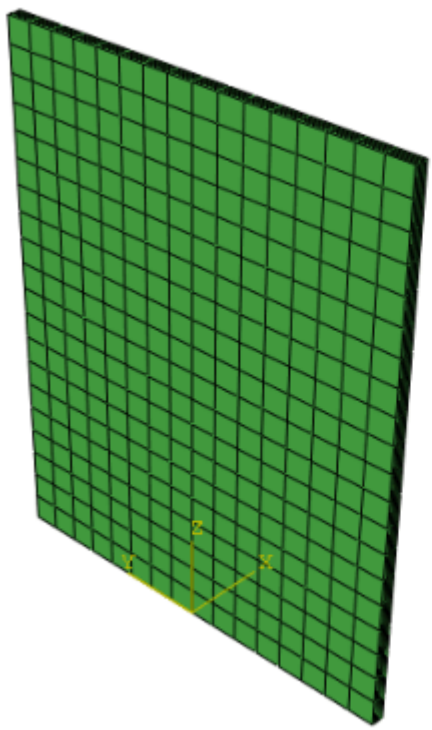
Two files have been prepared for this tutorial: an Abaqus input file, and a fluid STAR-CCM+ simulation. Copy these files into the `/abaqus` and `/starccm+` directories:

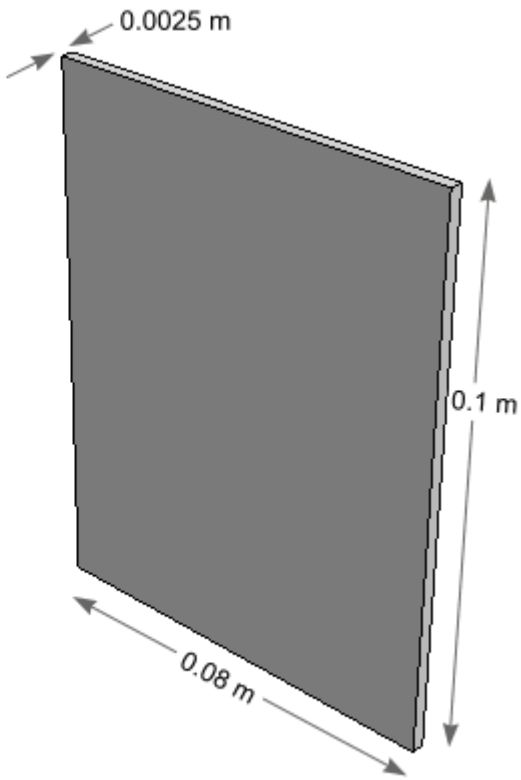
3. Navigate to `[INSTALL_DIR]/doc/startutorialsdata/couplingWithCaeCodes/data`.
4. Copy the file `plate.inp` into your `/mechanicalCoSimTutorial/abaqus/` directory.
5. Copy the file `plate-cosim.sim` into your `/mechanicalCoSimTutorial/starccm+/` directory.

Reviewing the Abaqus Model

Review the features of the Abaqus model and the co-simulation-settings.

The Abaqus model for this tutorial is a flat plate that is pinned at the base. The mesh and plate dimensions are shown in the diagram below.





Mesh	C3D8R elements
Material Properties	Young's modulus: 38.4 MPa
	Poisson's ratio: 0.3
	density: 4096 kg/m^3
Assumptions/Models	elastic material
	Rayleigh damping with beta coefficient of 1e-6
	Single dynamic load step
Maximum Time-Step	5.0e-4 s
Total Simulation Time	0.125 s

Co-Simulation Settings

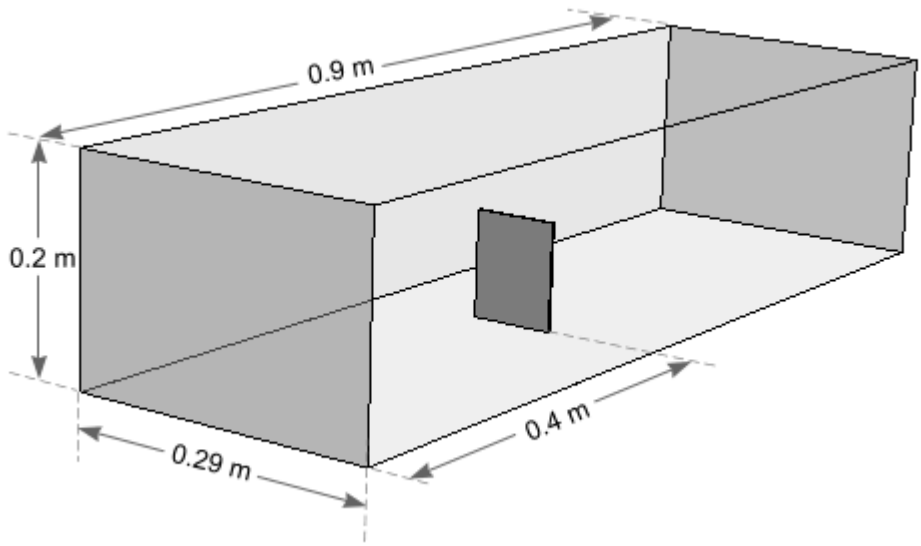
The Abaqus input file provided for this tutorial, `plate.inp`, already includes the necessary commands for co-simulation. The co-simulation is carried out during the load step. The co-simulation commands are included within the Step definition. You can review the co-simulation commands by opening the `plate.inp` file with a text editor of your choice.

```** ** STEP: Step-1 **```	This section of the file, up to `*End Step`, contains the history data for the Abaqus job, including solver parameters, boundary conditions, loads, and outputs.
```** *Output, history, variable=PRESELECT ** *CO-SIMULATION, NAME=Plate, PROGRAM=MULTIPHYSICS```	Within the Step definition, the line in bold specifies that the co-simulation takes place in this step, and defines an arbitrary name for the co-simulation, Plate.
```** *Output, history, variable=PRESELECT *CO-SIMULATION, NAME=Plate, PROGRAM=MULTIPHYSICS *CO-SIMULATION REGION, TYPE=SURFACE, EXPORT ASSEMBLY_FSI_INTERFACE, U *CO-SIMULATION REGION, TYPE=SURFACE, IMPORT ASSEMBLY_FSI_INTERFACE, CF```	The set of commands in bold identifies the interfacing surface and the fields that are imported and exported during the co-simulation. The first line identifies a region in the model, in this case a surface, from which data is exported. In the next line, the surface is identified by entering the name of the assembly in which it is defined, `Assembly`, and then the name of the surface, `FSI_Interface`. The syntax of the surface identifier is `<ASSEMBLY NAME>_<SURFACE NAME>`. The export field is defined as the nodal displacement, `U`. The next two lines follow the same pattern to specify that traction vectors, `CF`, are imported into Abaqus.

Preparing the STAR-CCM+ Model

Prepare the fluid model simulation in STAR-CCM+.

The STAR-CCM+ model defines a fluid domain surrounding the solid plate. In the simulation, air enters the fluid domain at 10 m/s. The outlet boundary is modeled as a Pressure Outlet, and the top and sides of the domain are specified as Symmetry boundaries. Flow is defined as incompressible, and turbulence is modeled using the K-Omega model.



The dimensions of the fluid domain are shown in the above diagram.

Contents:

Loading the Initial Simulation
Generating the Volume Mesh
Selecting Physics Models
Defining Boundary Conditions
Setting Up the Morpher
Creating a Plane Section
Preparing a Scalar Scene
Setting Up Automatic Scene Export
Specifying Co-Simulation Settings
Setting Up Monitors

Loading the Initial Simulation

For this tutorial, you are provided with a simulation file that contains pre-defined objects.

Load the initial simulation:

1. Launch STAR-CCM+.
2. Select **File** > **Load Simulation...**
3. In the *Load Simulation* dialog, click **Browse...**
4. In the *Open* dialog, navigate to the `/mechanicalCoSimTutorial/starccm+/` directory that you created for this tutorial.
5. Select `plate-cosim.sim`, then click **Open**.
6. In the *Load Simulation* dialog, click **OK**.

Generating the Volume Mesh

The simulation file provided for this tutorial is complete with the required mesh settings. In this tutorial, the volume mesh is generated using the Polyhedral Mesher and a surface control is used for mesh refinement at the FSI interface.

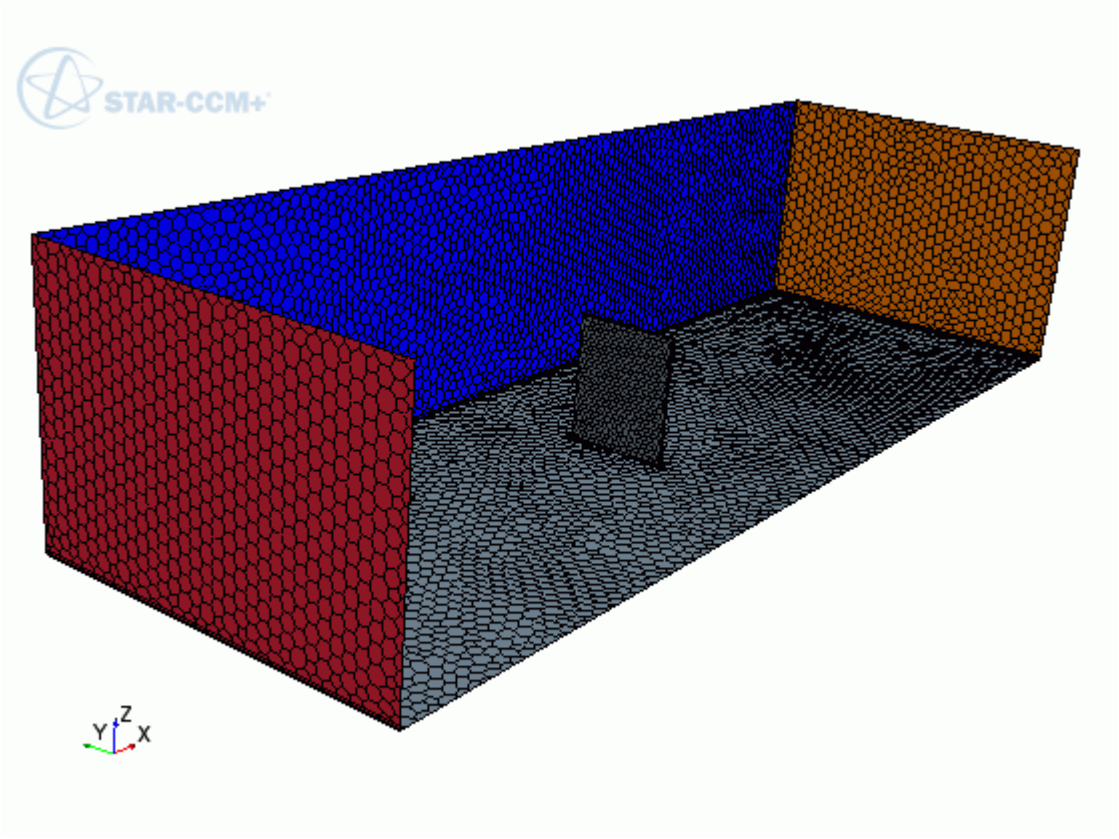
You can review the mesh set up under the **Geometry** > **Operations** > **Fluid Mesh** node.

To generate the volume mesh:

1. Click (**Generate Volume Mesh**).
   Output from the meshing process is shown in the *Output* window.
2. When the mesh generation process is complete, save the simulation.

To visualize the volume mesh using a mesh scene:

3. Create a new mesh scene.
   The volume mesh is displayed in **Mesh Scene 1**.
4. Remove some of the boundaries from the scene to see the plate inside the fluid domain:
   a) Edit the **Scenes** > **Mesh Scene 1** > **Displayers** > **Mesh 1** > **Parts** node.
   b) In the *Edit* dialog, deselect the **Top** and **Front** boundaries, then click **OK**.
5. Use the mouse to rotate the view and examine the mesh.
   The mesh appears as shown below.



Selecting Physics Models

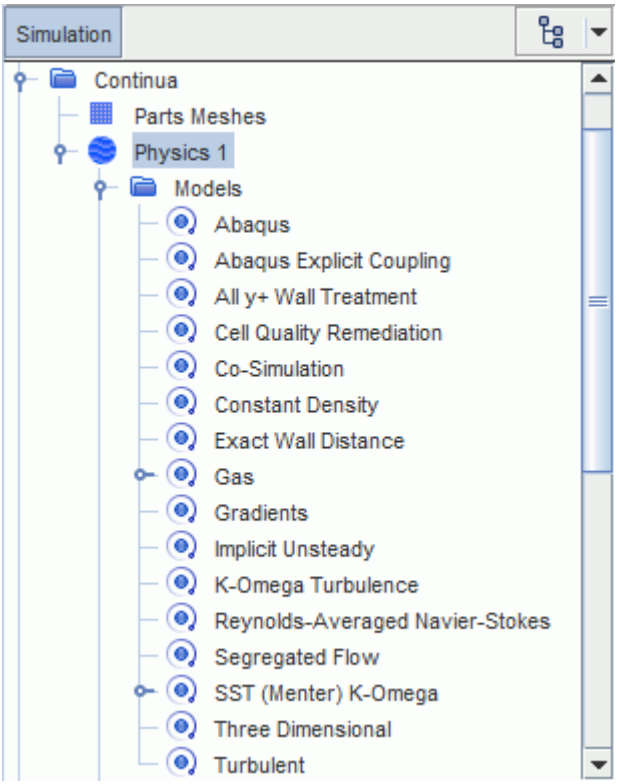
Following mesh generation, STAR-CCM+ automatically created a physics continuum. Use this continuum to define the physics models for the fluid domain.

Select the physics models and specify the fluid initial velocity:

1. For the physics continuum, **Continua** > **Physics 1**, select the following models in order:

Group Box	Model
Enabled Models	**Three-Dimensional** (Selected automatically)
Time	**Implicit Unsteady**
Material	**Gas**
Flow	**Segregated Flow**
	Gradients (Selected automatically)
Equation of State	**Constant Density**
Viscous Regime	**Turbulent**
	Reynolds-Averaged Navier-Stokes (Selected automatically)
Reynolds-Averaged Turbulence	**K-Omega Turbulence**
	SST (Menter) K-Omega (Selected automatically)
	Exact Wall Distance (Selected automatically)
	All Y+ Wall Treatment (Selected automatically)
Optional Models	**Co-Simulation**
Co-Simulation Models	**Abaqus**
Abaqus Coupling Models	**Abaqus Explicit Coupling**
Optional Models	**Cell Quality Remediation**

2. Click **Close**.
3. To review the models, open the **Physics 1** > **Models** node.



Initialize the air velocity in the domain at 10 m/s perpendicular to the plate:

4. Select the **Continua** > **Physics 1** > **Initial Conditions** > **Velocity** node and set *Value* to `[10, 0, 0] m/s`.

Defining Boundary Conditions

Specify the fluid velocity at the inlet.

1. Expand the **Regions** > **plateFluid** > **Boundaries** > **Inlet** > **Physics Values** node.
2. Select the **Velocity Magnitude** node and set *Value* to `10 m/s`.

Setting Up the Morpher

Use the Morpher to allow the fluid mesh to deform in response to the imported nodal displacements from Abaqus.

Create a morphing motion and assign it to the fluid region:

1. Right-click the **Tools** > **Motions** node and select **New** > **Morphing**.

   A new **Morphing** node is added under the **Motions** node. By default, the morpher is set to automatically remove vertices on boundaries while the simulation runs. The *Automatic Thin-out CI* factor determines the number of control vertices that are used in each iteration. As the level of deformation increases, the morpher automatically uses more control vertices.

Decrease the *Automatic Thin-out CI* factor to increase the sensitivity of the morpher to the mesh deformation:

2. Select the **Motions** > **Morphing** > **RBF Parameters** node and set *Automatic Thin-out CI* to `0.5`.

To apply this motion to the fluid region:

3. Select the **Regions** > **plateFluid** > **Physics Values** > **Motion Specification** node and set *Motion* to **Morphing**.

   The FSI boundary will be automatically set to move according to the displacements imported from Abaqus, once that a co-simulation link is set up. See Specifying Co-Simulation Settings.

Change the morpher method for the inlet and outlet boundaries to Fixed Plane. This morpher method is suitable for the selected boundaries and is more computationally efficient:

4. Expand the **Boundaries** node.
5. Multi-select the **Inlet** and **Outlet** nodes.
6. Right-click one of the selected boundaries and select **Edit**.
7. Select the **Physics Conditions** > **Morpher Specification** node and set *Specification* to **Constraint**.
8. Select the **Morpher Constraint Specification** node and set *Constraint* to **Fixed Boundary Plane**.
9. Close the *Multiple Objects* dialog.

To prevent inaccuracies from accumulating during the simulation (due to interpolation), change the mesh morpher settings:

10. Select the **Solvers** > **Mesh Morpher** node and activate *Morph From Zero*.

Creating a Plane Section

Create a plane section through the plate, to visualize the solution and mesh deformation.

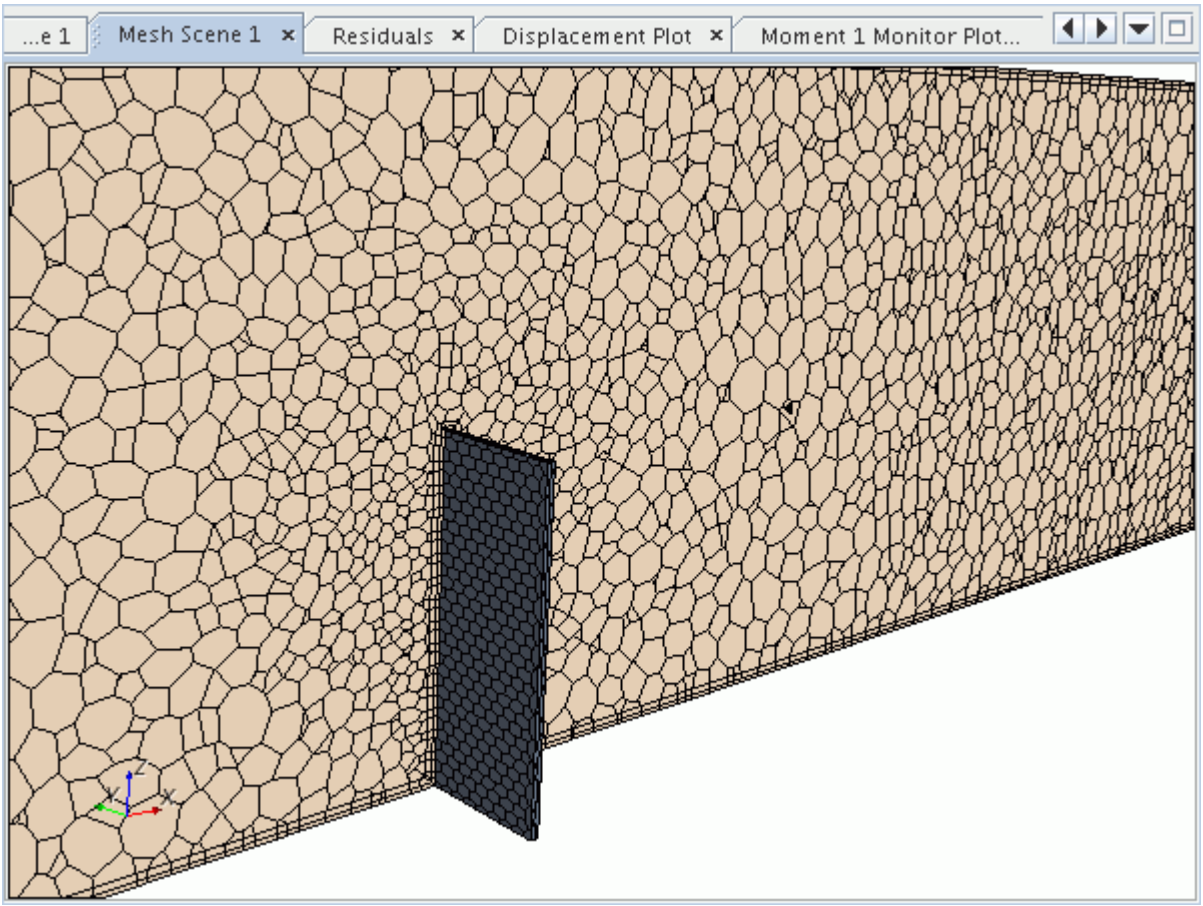
1. Click the **Mesh Scene 1** tab in the *Graphics* window.
2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane...**.
3. In the *Create Plane Section* in-place dialog:
   a) Set the following properties:

Property	Setting
origin	`[0.05, 0, 0.1] m`
normal	`[0, 1, 0] m`
Display	**Existing Displayer**

   b) Click **Create** then **Close**.

Modify the parts that are displayed in the scene to show the mesh on the plane section and the solid plate:

4. Edit the **Scenes** > **Mesh Scene 1** > **Displayers** > **Mesh 1** > **Parts** node.
5. In the *Parts* dialog, deselect all the boundaries in **plateFluid**, apart from **Plate_FSI**, then click **OK**.
6. Use the mouse to rotate the view so that it appears as shown below.



Preparing a Scalar Scene

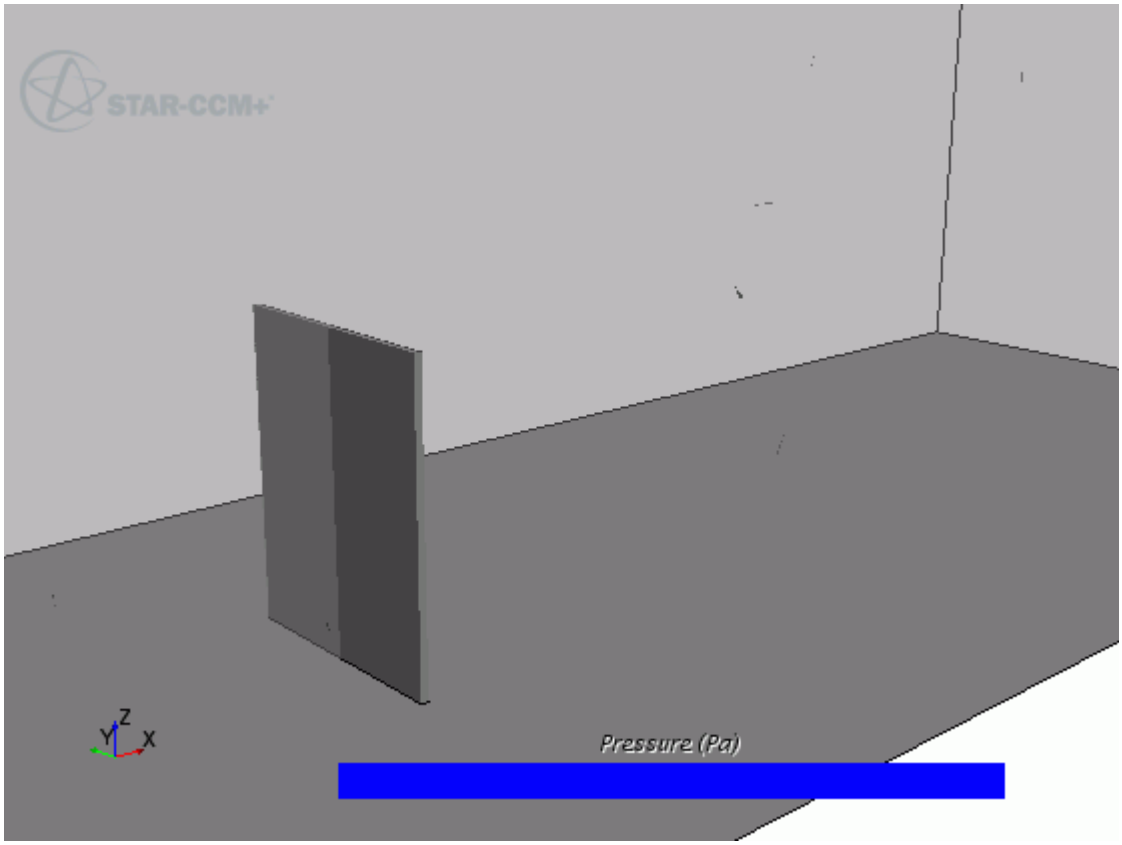
Prepare a scalar scene to display the pressure field around the plate. Include the plane section in a separate scalar displayer so that the opacity setting can be adjusted independently.

1. Create a scalar scene.
2. Right-click the **Scalar Scene 1** > **Displayers** node and select **New Displayer** > **Scalar**.
3. Edit the **Scalar Scene 1** > **Displayers** node and set the following properties:

Node	Property	Setting
Scalar 1	*Contour Style*	**Smooth Filled**
└ **Parts**	*Parts*	**Regions** > **plateFluid** > **Bottom**
		Regions > **plateFluid** > **Plate_FSI**
└ **Scalar Field**	*Function*	**Pressure**
Scalar 2	*Opacity*	`0.5`
	Contour Style	**Smooth Filled**

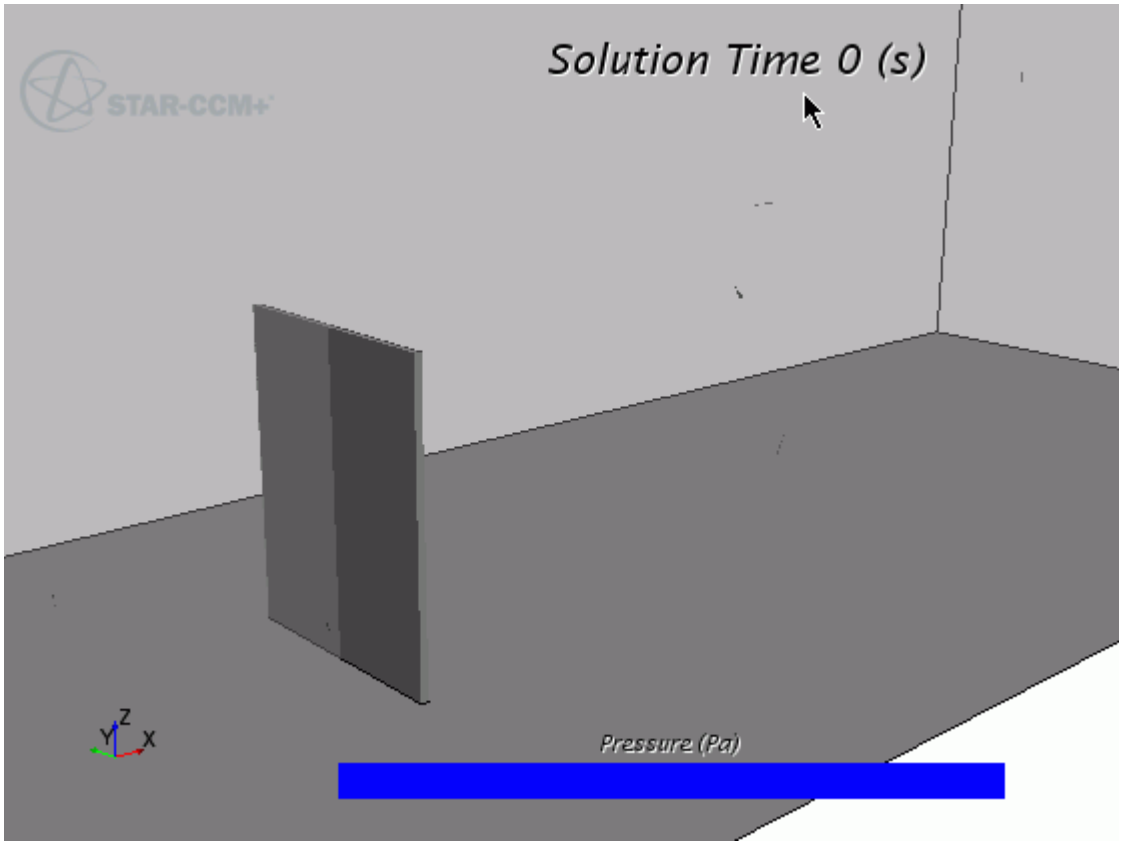
Node	Property	Setting
└ **Parts**	*Parts*	**Derived Parts** > **Plane Section**
└ **Scalar Field**	*Function*	**Pressure**
└ **Color Bar**	*Visible*	**Deactivated**

4. Position the scene as shown below.



Add an annotation to the scene to show the current solution time:

5. Expand the **Tools** > **Annotations** node.
6. Click the **Solution Time** node and drag it into the scene. Release the mouse button anywhere in the scene.
7. Move the annotation to the top-right of the scene using click-and-drag.



Add a solution time annotation to the mesh scene:

8. Display the mesh scene in the **Graphics** window by selecting the **Mesh Scene 1** tab.
9. Using the same technique as described above, add a Solution Time annotation to the scene.
10. Save the simulation.

Note: In transient cases, it is often useful to export copies of scalar scenes while the simulation is running. These image files can then be used to visualize the solution in an animation. The next page in this tutorial runs through the steps for setting up automatic scene export. If you do not wish to export scenes automatically during the co-simulation, you can jump directly to Specifying Co-Simulation Settings.

Setting Up Automatic Scene Export

In the following steps, set up automatic export of the mesh and pressure scenes.

The automatic export of scenes leads to the generation of a number of `.png` files on your hard drive while the simulation runs. These image files can then be used to create an animation.

1. Create an `/images` directory in your `/mechanicalCoSimTutorial/starccm+/` directory as a location to save the image files.

Many of the settings for the scenes are the same. You can apply the settings to both scenes simultaneously using the Multi-edit dialog.

2. Multi-select the **Scenes** > **Scalar Scene 1** and **Scenes** > **Mesh Scene 1** nodes.
3. Right-click one of the selected nodes and select **Edit...**
4. Set the following properties:

Node	Property	Setting
Mesh Scene 1, **Scalar Scene 1**		
└ **Attributes** > **Update**	*Save To File*	**Activated**
	Output Directory	`/mechanicalCoSimTutorial/starccm+/images`
	Trigger	**Time Step**
└ **Time-Step Frequency**	*Frequency*	`5`
└ **Hardcopy**	*Output Width*	`800`
	Output Height	`600`

Then close the dialog. A red * is shown next to the icon for each scene to indicate that copies are saved automatically when the co-simulation is run.

Finally, specify unique base filenames for STAR-CCM+ to use when saving the image files:

5. Select the **Scenes** > **Mesh Scene 1** > **Attributes** > **Update** node and set *Base Filename* to `mesh`.
6. Using the same technique, set the base filename for **Scalar Scene 1** to `pressure`.

Specifying Co-Simulation Settings

Specify co-simulation settings in STAR-CCM+, including the Abaqus input file location, the coupled boundary defining the fluid-structure interface, and the fields that are exchanged at each coupling step.

The Abaqus input file provided for this tutorial, `plate.inp`, already includes the necessary commands for co-simulation. For more information, see Reviewing the Abaqus Model.

In STAR-CCM+, you specify all co-simulation settings under the co-simulation link, **Co-Simulations** > **Abaqus Co-Simulation 1**, which appears in the simulation tree when you activate an Abaqus Co-Simulation model.

1. Select the **Co-Simulations** > **Link 1** > **Values** > **Abaqus Execution** node and set the following properties:

Property	Setting
Current Job Name	`co-simulation_plate`
Input File	`../abaqus/plate.inp`
Executable Name	command you use to launch Abaqus, for example: `abaqus`

To specify the location of the input file, you type the path directly into the box. For example, if your working directory is `/ mechanicalCoSimTutorial/starccm+/` and your input file is contained in the folder `/mechanicalCoSimTutorial/abaqus/`, the path to the input file would be `../abaqus/plate.inp`.

2. Select the **Values** > **Constant Coupling Time Interval** node and set *Value* to `5.0E-4`.

To specify the coupled boundaries:

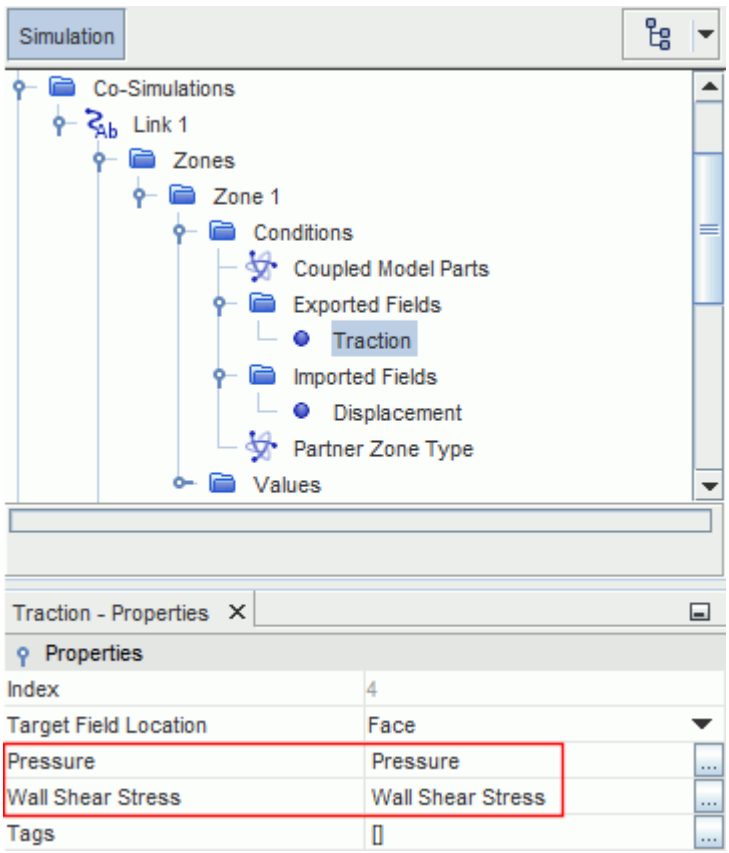
3. Expand the **Regions** > **plateFluid** > **Boundaries** > **Plate_FSI** node.
4. Select the **Physics Conditions** > **External Code Coupling Specification** node and set *External Code Coupling Specification* to **Link 1: Zone 1**.
5. Expand the **Co-Simulations** > **Link 1** > **Zones** > **Zone 1** > **Conditions** node.

   STAR-CCM+ automatically sets the *Boundaries* property for the **Coupled Model Parts** node to the specified coupled boundaries.

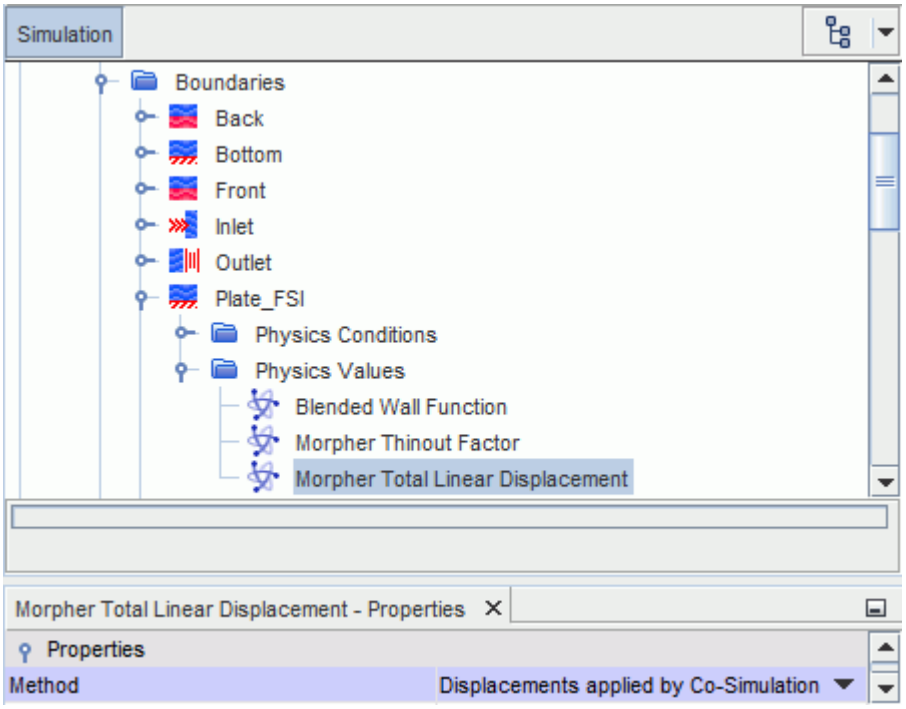
By default, STAR-CCM+ exports pressure and imports displacement. In this tutorial, you change the exported field to traction (fluid pressure and wall shear stress):

6. Right-click the **Zone 1** > **Conditions** > **Exported Fields** > **Pressure** node and select **Delete**. Click **OK** to confirm.
7. Right-click the **Exported Fields** node and select **Add Mechanical Field** > **Traction**.

   For each field specification, STAR-CCM+ automatically sets the field functions that are exchanged at the fluid-structure interface.



The default *Field Treatment* option for imported fields is **Managed**. This setting instructs STAR-CCM+ to automatically apply the imported data to the coupled model parts. In this case, STAR-CCM+ automatically updates the morpher settings for the FSI boundary, so that the boundary moves according to the imported displacements.



8. Save the simulation.

Setting Up Monitors

Create reports to monitor the displacement of the upper corners of the plate in the x-direction, the traction load, and the moment about the z-axis, during the co-simulation.

To monitor the displacements of the upper-left and upper-right corners of the plate, two point-probes must be created that follow the motion of the corners as the plate moves in the flow.

1. In the *Graphics* window, click the *Mesh Scene 1* tab.
2. Right-click the **Derived Parts** node and select **New Part** > **Probe** > **Point...**
3. In the *Create Point Probe* dialog, set the properties as follows:

Property	Setting
Input Parts	**Regions** > **plateFluid** > **Boundaries** > **Plate FSI**
X	`0.0 m`
Y	`0.04 m`
Z	`0.1 m`
Display	**No Displayer**

4. Click **Create** then **Close**.

   A node, **Point**, is added beneath the **Derived Parts** node.

To create another point probe for the other corner:

5. Copy the **Derived Parts** > **Point** node and paste it onto the **Derived Parts** node.
6. Rename the **Derived Parts** > **Point** node to `Upper +Y`, and the **Derived Parts** > **Copy of Point** to `Upper -Y`.
7. Edit the **Upper -Y** node and set *Point* to `[0.0, -0.04, 0.1] m,m,m`.

To make the points move with the plate when the morpher moves the mesh:

8. Multi-select the **Upper +Y** and **Upper -Y** nodes and activate *Follow Motion*.

Having created point probes on the upper corners of the plate, you can now create reports that report the displacement of these corners:

9. Right-click the **Reports** node and select **New Report** > **Maximum**.
10. Rename the **Reports** > **Maximum 1** node to `X Displacement Upper +Y`.
11. Set the following properties:

Property	Setting
Field Function	**Displacement Imported by Co-Simulation** > **Laboratory** > **i**
Parts	**Derived Parts** > **Upper +Y**
Smooth Values	**Activated**

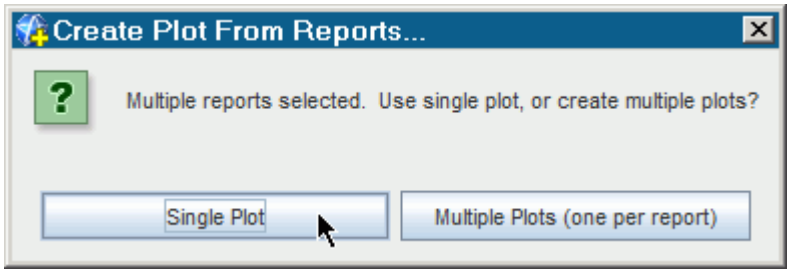
12. Create another **Maximum** report for the other corner.
13. Rename the report that you just created, **Reports** > **Maximum 1**, to `X Displacement Upper -Y`.
14. Set the following properties:

Property	Setting
Field Function	**Displacement Imported by Co-Simulation** > **Laboratory** > **i**
Parts	**Derived Parts** > **Upper -Y**
Smooth Values	**Activated**

Use the displacement reports to create monitors and a plot:

15. Multi-select the **X Displacement Upper +Y** and **X Displacement Upper -Y** reports.
16. Right-click one of the reports and select **Create Monitor and Plot from Report**.

    The *Create Plot From Reports* dialog appears.
17. Select **Single Plot**.



The report monitors are displayed under the **Monitors** node, and the plot is shown under the **Plots** node.

18. Rename the **Plots** > **Reports Plot** node to `Displacement Plot`.

Create two more reports to monitor the traction load and moment on the plate:

19. Create a **Force** report.
20. Select the **Reports** > **Force 1** node and set the following properties:

Property	Setting
Direction	`[1.0, 0.0, 0.0]` (default)
Force Option	**Pressure + Shear** (default)
Parts	**Regions** > **plateFluid** > **Plate_FSI**

21. Right-click the **Reports** node and select **New Report** > **Moment**.
22. Select the **Reports** > **Moment 1** node and set the following properties:

Property	Setting
Axis	`[0.0, 0.0, 1.0]` (default)
Force Option	**Pressure + Shear** (default)
Parts	**Regions** > **plateFluid** > **Plate_FSI**

Create two separate plots for the force and moment reports:

23. Multi-select the **Force 1** and **Moment 1** reports.
24. Right-click and select **Create Monitor and Plot from Reports**.
25. In the dialog that appears, select **Multiple Plots (one per report)**.

Finally, verify the trigger for monitoring the data is set to time-step:

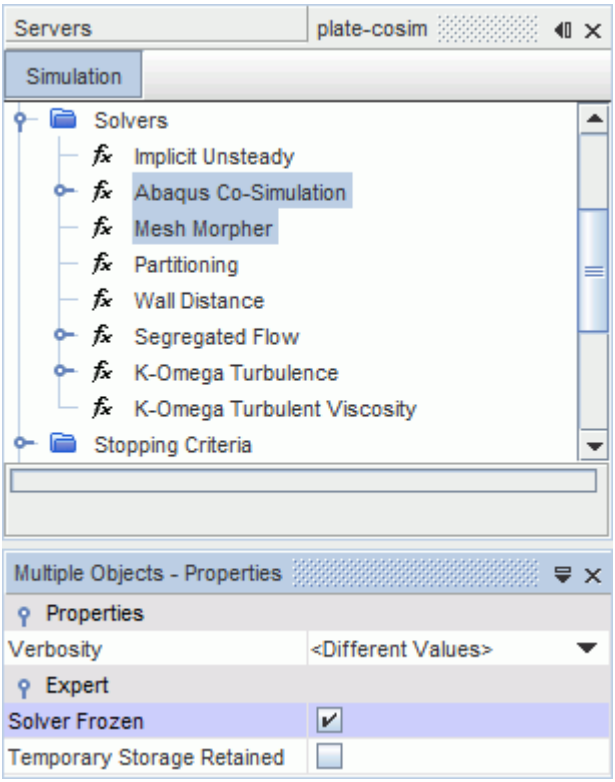
26. Expand the **Monitors** node and multi-select the four report monitors.
27. Make sure that *Trigger* is set to **Time Step**.
28. Open all plots.
29. Save the simulation.

Obtaining an Initial Flow Field

Run the simulation in a quasi-steady state to obtain an initial flow field. The solution from this initialization step is used as the initial condition for the co-simulation.

To solve for a steady-state flow about a rigid plate, freeze the co-simulation and mesh morpher solvers. The number of inner iterations per time-step are increased to allow a flow field to develop within a single time-step. Then, run the simulation for one time-step. As you are only searching for a steady-state solution, you do not need to be time-accurate. Therefore, use a large time-step of 1.5 s and a 1st-order temporal discretization setting.

1. Select the **Solvers** > **Implicit Unsteady** node and set *Time Step* to `1.5 s`.
2. Multi-select the **Solvers** > **Abaqus Co-Simulation** and **Solvers** > **Mesh Morpher** nodes.
3. Activate *Solver Frozen*.



4. Select the **Stopping Criteria** > **Maximum Inner Iterations** node and set *Maximum Inner Iterations* to `400`.

The quasi-steady simulation only runs for one step, so disable the maximum physical time and maximum steps stopping criteria. Before the co-simulation is run, re-enable the maximum physical time:

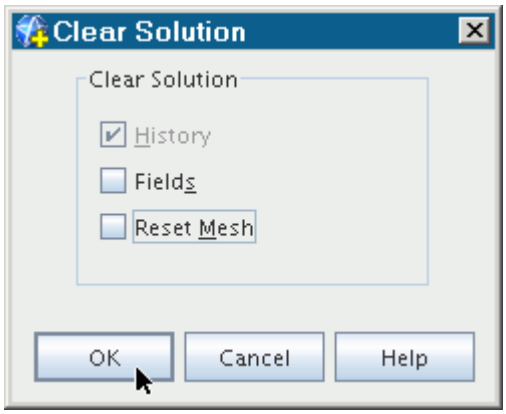
5. Multi-select the **Stopping Criteria** > **Maximum Physical Time** and **Stopping Criteria** > **Maximum Steps** nodes.
6. Deactivate *Enabled*.
7. From the menu, select **Solution** > **Step**.

When the initial simulation is run, the pressure field is not displayed in **Scalar Scene 1**. To display the solution:

8. Refresh **Scalar Scene 1**.
9. Save the simulation.
10. Also, save a copy of the simulation at this stage called `plate-cosim-initial.sim`.

The solution data is retained in the simulation, and provides the initial conditions for the co-simulation. Reset the time to zero by clearing the solution history. Clearing the solution fields would cause you to lose the initial solution that has been computed:

11. From the menu, select **Solution** > **Clear Solution**.
12. In the *Clear Solution* dialog, deactivate *Fields* and *Reset Mesh*, then click **OK**.



Running the Co-Simulation

Specify the solver settings, stopping criteria, and run the simulation.

The co-simulation settings are registered with the Co-Simulation Engine, and feedback is shown in the **Output** window.

To prepare for the co-simulation, specify solver settings and stopping criteria suitable for the unsteady simulation:

1. Select the **Solvers** > **Implicit Unsteady** node and set the following properties:

Property	Setting
Time Step	`5e-4 s`
Temporal Discretization	**2nd-order**

2. Multi-select the **Solvers** > **Abaqus Co-Simulation** and **Solvers** > **Mesh Morpher** nodes.
3. Deactivate *Solver Frozen*.
4. Select the **Solvers** > **Abaqus Co-Simulation** node and set *Verbosity* to **High**.
5. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Inner Iterations	*Maximum Inner Iterations*	`10`
Maximum Physical Time	*Maximum Physical Time*	`0.125 s`
	Enabled	**Activated**

You are now ready to run the co-simulation. During this process, Abaqus runs in the background, and STAR-CCM+ communicates with Abaqus using the SIMULIA Co-Simulation Engine:

6. Run the simulation.

   The co-simulation settings are registered with the Co-Simulation Engine, and feedback is shown in the *Output* window.

   While the co-simulation is running, you can monitor progress using the plots and scenes created earlier. At each coupling step solution data is exchanged between Abaqus and STAR-CCM+, and is reported in the *Output* window.

7. When the co-simulation is complete, save the simulation.

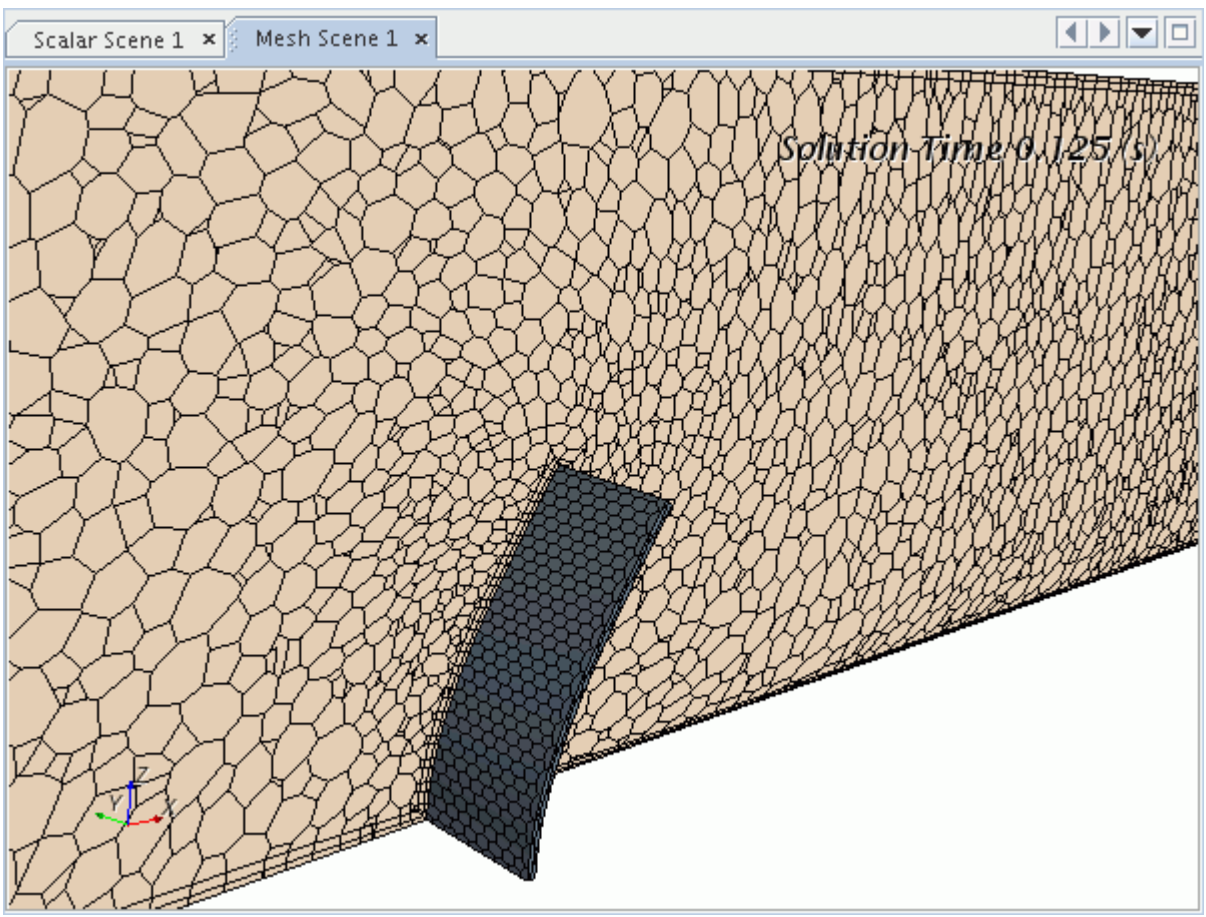
Contents:

Visualizing the Results

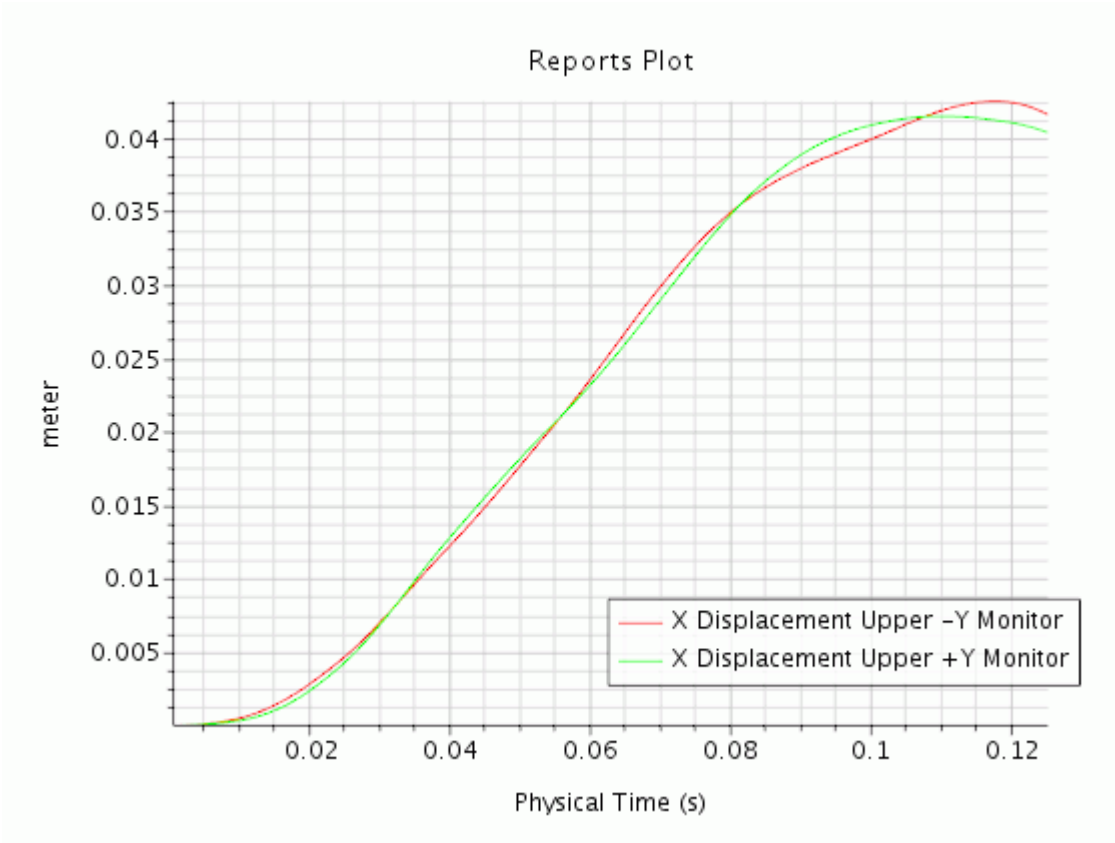
Visualizing the Results

Examine the mesh deformation and the Displacement Plot.

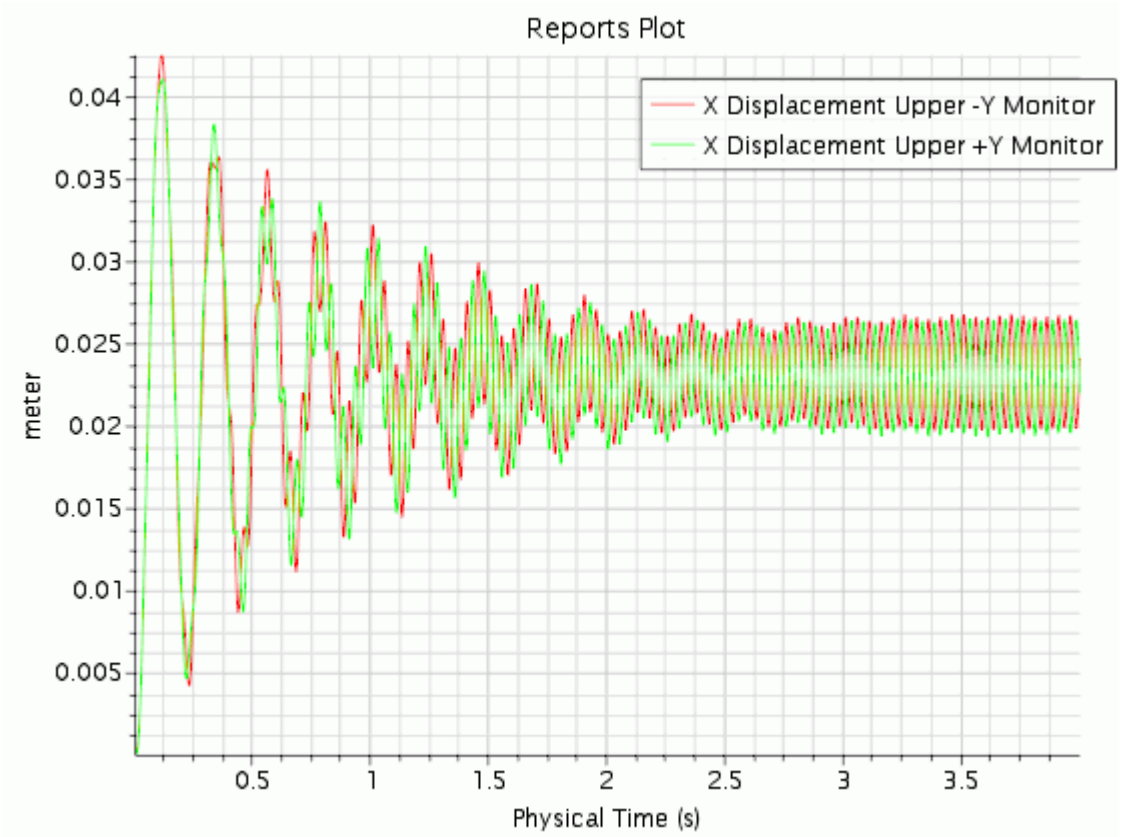
In *Mesh Scene 1*, you can see that the oncoming air flow has caused the plate to bend backwards. The morpher has deformed the mesh throughout the fluid region.



In the *Displacement Plot*, you can see that after 0.125s the plate has reached its maximum displacement in the x-direction, around 0.042m.



If you run the case for 4 s, the displacement plot shows that the first bending mode, where both corners of the plate would be in phase, is damped out after eight first bending mode periods, or 4 Hz. The remaining mode, the vortex shedding excites, is the first twisting mode at a frequency of about 20 Hz. In this mode, the corners of the plate are out of phase.



Summary

This tutorial demonstrated how to carry out a mechanical co-simulation case using STAR-CCM+ and Abaqus.

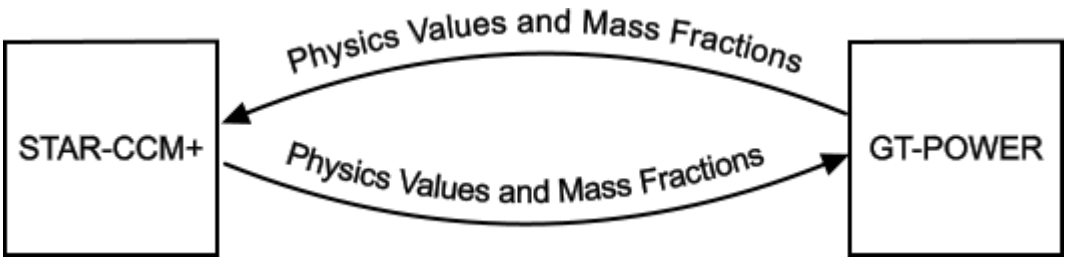
This tutorial has introduced the following STAR-CCM+ features:

- Setting up the STAR-CCM+ model:
  - Setting up a physics continuum for co-simulation.
  - Defining boundary conditions.
  - Setting up the morpher.
  - Setting up monitors and plots.
  - Setting up automatic scene export.
- Setting up the co-simulation.
- Obtaining an initial flow field.
- Running the co-simulation.

GT-SUITE Co-Simulation: 1D Coupling

This tutorial demonstrates the typical workflow for co-simulation with the engine simulation code GT-SUITE from Gamma Technologies.

In this tutorial, STAR-CCM+ simulates the three-dimensional flow within an intake manifold that forms part of an overall one-dimensional engine analysis modeled in GT-SUITE. The engine analysis simulate a four-cylinder, four-stroke engine running at 3000 rpm on Indolene fuel, including the effect of Exhaust Gas Recirculation (EGR).



During the co-simulation, STAR-CCM+ and GT-SUITE exchange the following data:

- species mass fractions
- density (Mass Flow inlet only)
- normal velocity (Mass Flow inlet only)
- pressure (Pressure outlet only)
- mass flow
- temperature

Coupling occurs once per time-step (explicit coupling scheme). STAR-CCM+ applies the data imported from GT-SUITE uniformly at the coupled boundary. Before sending data to GT-SUITE, STAR-CCM+ averages the three-dimensional fields across a sampling volume, to provide data transition from the three-dimensional to the one-dimensional simulation.

In this tutorial, you obtain initial conditions for the STAR-CCM+ simulation from the uncoupled GT-SUITE solution.

During the co-simulation, the GT-SUITE solver runs 20 pre-cycles before transferring data to STAR-CCM+, which allows the GT-SUITE model to converge to a periodic steady-state. Once coupling begins, the solvers exchange data every time-step for three more engine cycles.

Contents:

Prerequisites
Preparing the File Structure
Modifying the GT-SUITE Simulation for Co-Simulation
Setting Up the STAR-CCM+ Simulation
Coupling STAR-CCM+ to GT-SUITE
Running the Co-Simulation

Visualizing the Results
Summary

Prerequisites

The instructions in the GT-SUITE Co-Simulation - 1D Coupling tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Techniques	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

This tutorial also assumes that you are familiar with GT-SUITE, GT-ISE (GT-SUITE graphical interface), and GT-POST (GT-SUITE post-processing tool).

If you are not familiar with GT-SUITE, complete the standard tutorial. You can access this tutorial in GT-SUITE from **File** > **Tutorials** > **Modeling_Applications** > **Engine_Performance** > **Engine_Performace-tutorials.pdf**.

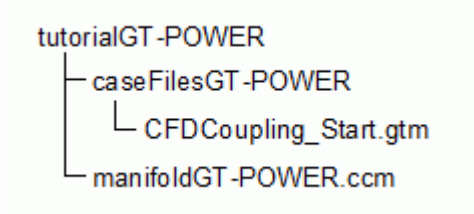
This tutorial describes the procedure using GT-SUITE v2017. For a list of all supported versions of GT-SUITE, refer to the section *External Packages Support* of the STAR-CCM+ Release Notes.

Preparing the File Structure

For this tutorial, you are provided with a starting GT-SUITE simulation file, containing the engine analysis, and the intake manifold STAR-CCM+ volume mesh. Copy these files to your working directory.

1. Navigate to `[INSTALL_DIR]/doc/startutorialsdata/couplingWithCaeCodes/data`.
2. Copy `manifoldGT-POWER.ccm` to your working directory.
3. In the working directory, create a subdirectory called `caseFilesGT-POWER`.
4. Navigate to [INSTALL_DIR]/doc/startutorialsdata/couplingWithCaeCodes/data/caseFilesGT-POWER_2017.
5. Copy `CFDCoupling_Start.gtm` to the new directory.

This structure makes it simpler when specifying the GT-SUITE simulation in STAR-CCM+.



Modifying the GT-SUITE Simulation for Co-Simulation

After running a standalone GT-SUITE analysis to convergence, you modify the GT-SUITE simulation for coupling with STAR-CCM+.

The required steps include:

- Adding CFD objects, to couple the one-dimensional GT-SUITE model with the three-dimensional simulation in STAR-CCM+.
- Creating and adding CFD Interfaces, which switch the flow solution from the one-dimensional components to STAR-CCM+ when coupling begins.
- Creating and adding CFD Connections, which represent the STAR-CCM+ boundaries in GT-SUITE.
- Creating and adding a CFD Component, which represents the STAR-CCM+ domain in GT-SUITE.
- Defining a species map that reduces the computational costs in STAR-CCM+ by solving fewer gas components.
- Making sure that the GT-SUITE simulation duration contains the coupled cycles. This practice allows the GT-SUITE solver to collect results for the coupled cycles for post-processing within GT-POST.
- Creating a `.dat` file that provides an input file for the GT-SUITE solver that defines the case setup.

Contents:

Running the GT-SUITE Simulation
Analyzing the Results in GT-POST
Adding CFD Objects to the Project Tree
Adding CFDInterfaces to the Model Map
Adding CFD Connections to the Model Map
Adding a CFD Component to the Model Map
Defining Species Mapping
Modifying the Run Setup
Creating the .dat file

Running the GT-SUITE Simulation

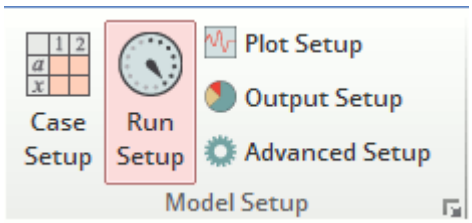
Before you couple STAR-CCM+ to GT-SUITE, you run a standalone GT-SUITE analysis and examine the results. This practice makes sure that the model is valid and determines how many cycles are required to converge to a steady state solution. It also reduces the risk of error and inaccuracies when running the coupled simulation.

To load the GT-SUITE case file:

1. Launch GT-SUITE.
2. Select **File** > **Open**.
3. In the *Open* dialog, navigate to your working directory, select `caseFilesGT-POWER/GT-CFDCoupling_Start.gtm`, and click **Open**.
4. Save the file as `CFDCoupling_Setup.gtm`.

Make sure that the solver terminates when the periodic, steady state criteria are met:

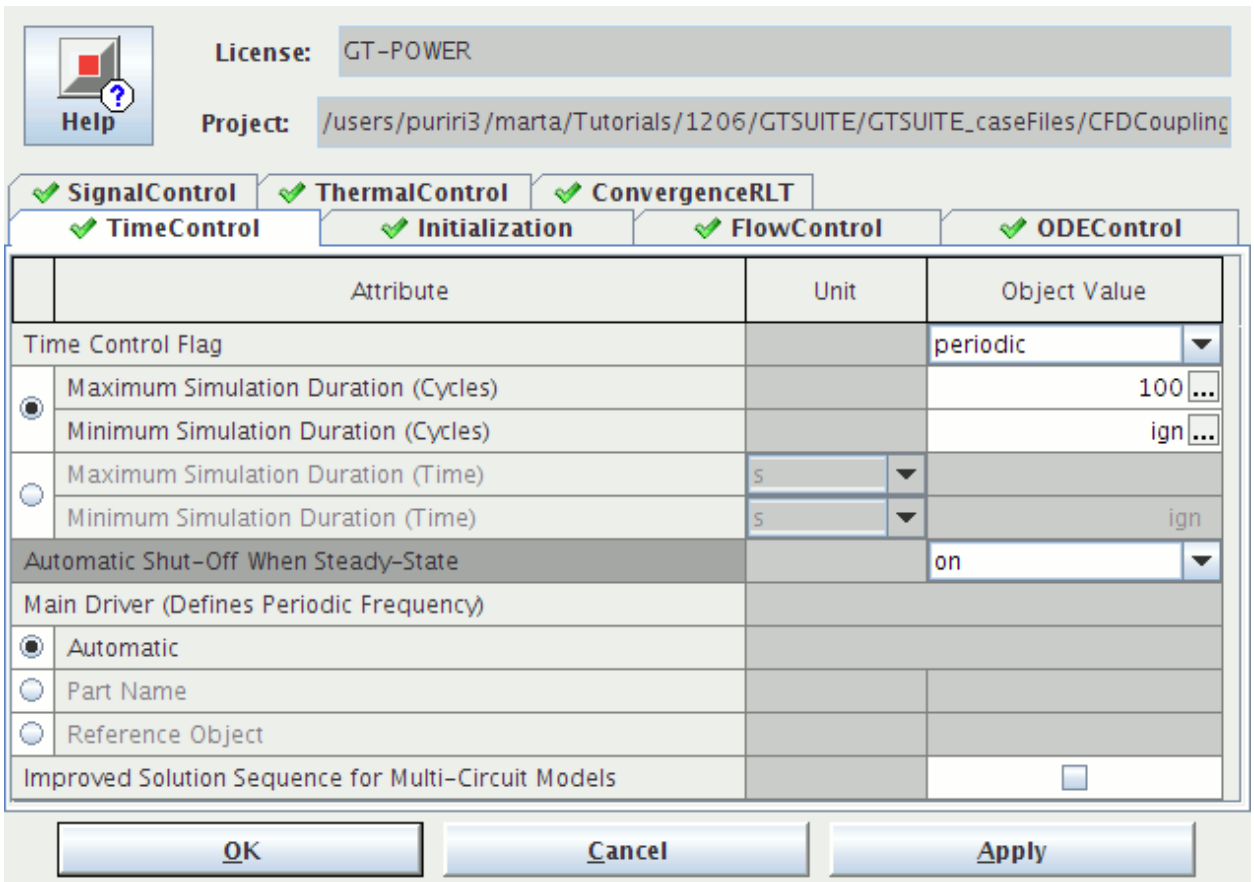
5. In the *Model Setup* ribbon, click **Run Setup**.



6. In the *Run Setup* dialog, select the *TimeControl* tab.

For this standalone analysis, you let the simulation run as many cycles as required to converge to steady state. To achieve this, you let GT-SUITE run for up to 100 cycles, and instruct it to stop when the convergence criteria are met.

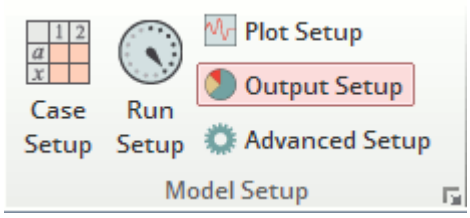
7. Make sure that *Maximum Simulation Duration (Cycles)* is set to `100` and *Automatic Shut-Off When Steady-State* is set to **on**.



8. Click **OK**.

Make sure that GT-SUITE writes Time RLT output data once per engine cycle:

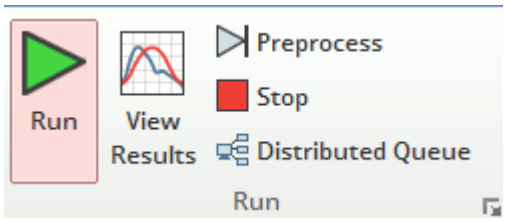
9. In the *Model Setup* ribbon, click **Output Setup**.



10. In the *Output Setup* dialog, select the *Data_Storage* tab.
11. Make sure that *Time RLT Storage Multiple (Periods)* is set to `1` and click **OK**.

To run the GT-SUITE solver:

12. In the *Run* ribbon, click **Run**.



13. In the *Simulation Wizard* dialog, click **Finish**.
14. When the solver is complete, close the solver window.

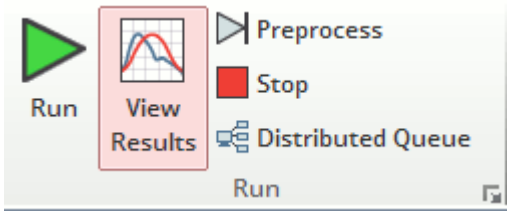
Analyzing the Results in GT-POST

Using GT-POST, you analyze the results of the GT-SUITE simulation. You can determine the number of cycles that are required to reach a steady state solution, and visualize the converged values for pressure, temperature, and mass fractions. These values can be used as initial conditions in STAR-CCM+, in order to speed up convergence of the co-simulation.

Note: The initial conditions in STAR-CCM+ do not need to match the GT-SUITE results exactly, as the influence of the initial conditions reduces during the transient solution. However, a close approximation speeds up convergence during the first few time-steps.

To launch GT-POST:

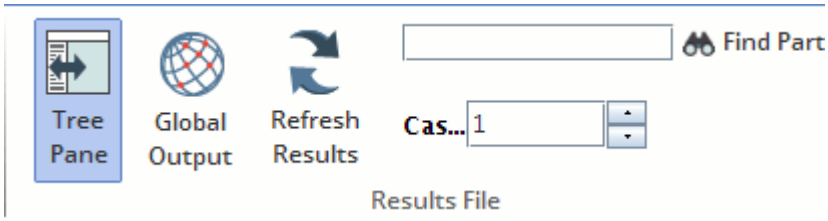
1. In the *Run* ribbon, click **View Results**.



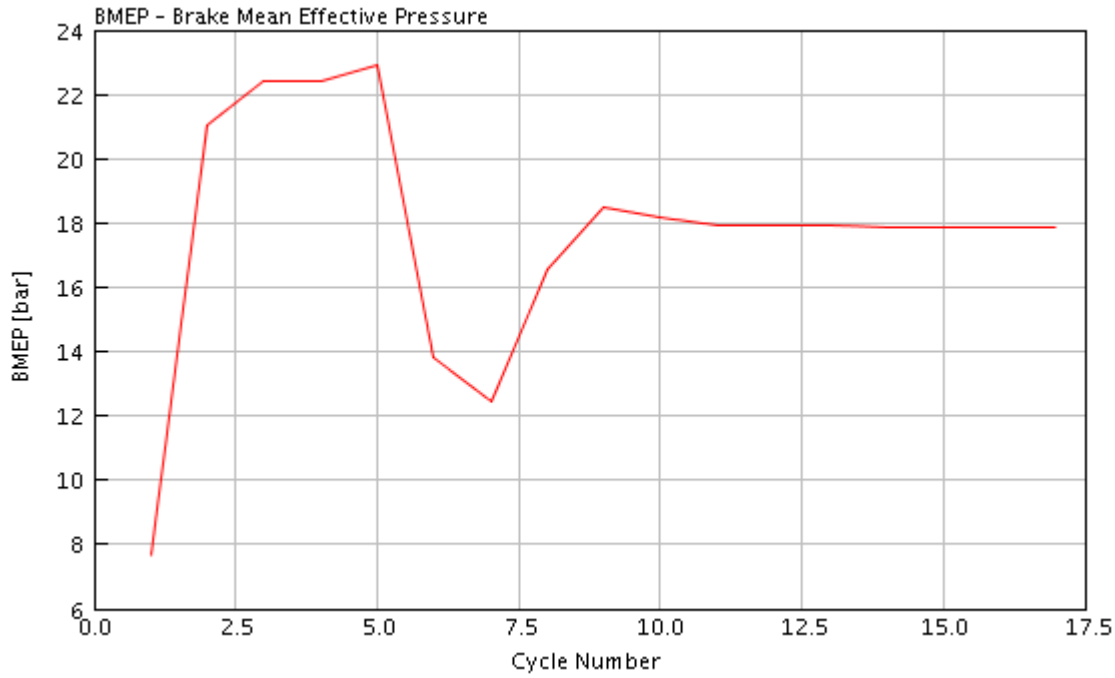
2. In the *Open* dialog, select `CFDCoupling_Setup.gdx` and click **Open**.

To visualize the number of cycles that are required to reach a steady state solution:

3. In the *Results File* ribbon, make sure that *Tree Pane* is selected.



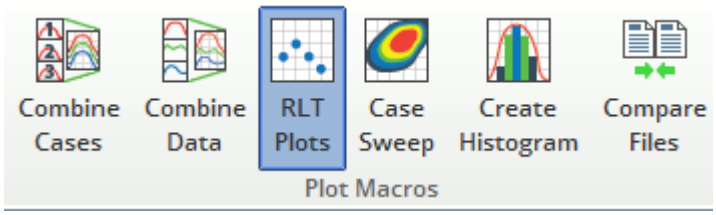
4. In the tree pane, select the *TimeRLT* tab.
5. Right-click the **EngineCrankTrain** > **engine** > **Performance** > **BMEP** plot and select **View**.



The simulation reaches a periodic steady state solution after 17 cycles. Therefore, when running the co-simulation, you allow GT-SUITE to run 20 pre-cycles before sending data to STAR-CCM+.

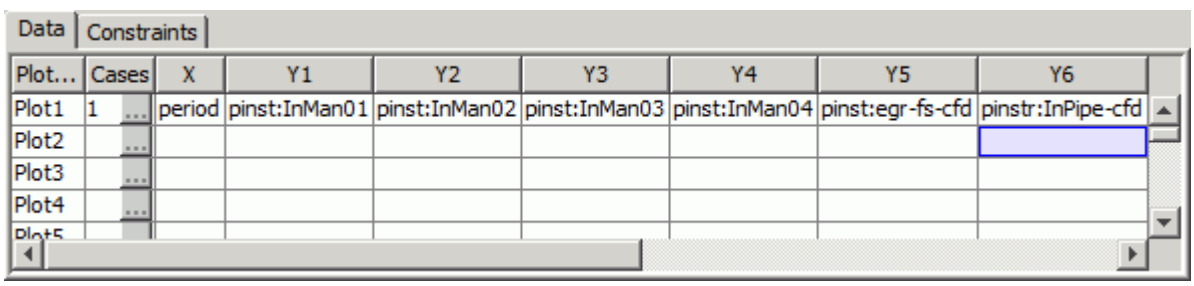
To visualize the pressure results:

6. In the menu bar, select the **Macros** tab.
7. In the *Plot Macros* ribbon, click **RLT Plots**.



8. In the *Run RLT Macro* dialog, select **Time RLT** and click **Next**.
9. In the *Data* tab, select the **Plot1**, *X* cell.
10. In the *RLT Selection* group box, select the **RLT tree** > **Special** node and select the **Cycle Number** data.
11. In the *Data* tab, select the **Plot1**, *Y1* cell.
12. In the *RLT Selection* group box, select the **RLT tree** > **main** > **Components** > **FlowSplitTRight** > **InMan01** > **Pressure** node, and select the **Pressure at End of Cycle** data.
13. Repeat the above steps and select the following data for the remaining components.

Cell	Node	Data
Y2	**FlowSplitTRight** > **InMan02** > **Pressure**	**Pressure at End of Cycle**
Y3	**FlowSplitTRight** > **InMan03** > **Pressure**	**Pressure at End of Cycle**
Y4	**FlowSplitTRight** > **InMan04** > **Pressure**	**Pressure at End of Cycle**
Y5	**FlowSplitGeneral** > **egr-fs-cfd** > **Pressure**	**Pressure at End of Cycle**
Y6	**PipeRound** > **InPipe-cfd** > **Pressure**	**Pressure at End of Cycle (Outlet)**



14. Click **Next**.
15. In the *Dataset Labeling* box, delete the existing text, enter [PARTNAME], and click **Finish**.

    When complete, GT-POST displays the following plot, which shows that pressure is approximately 2.13 bar.

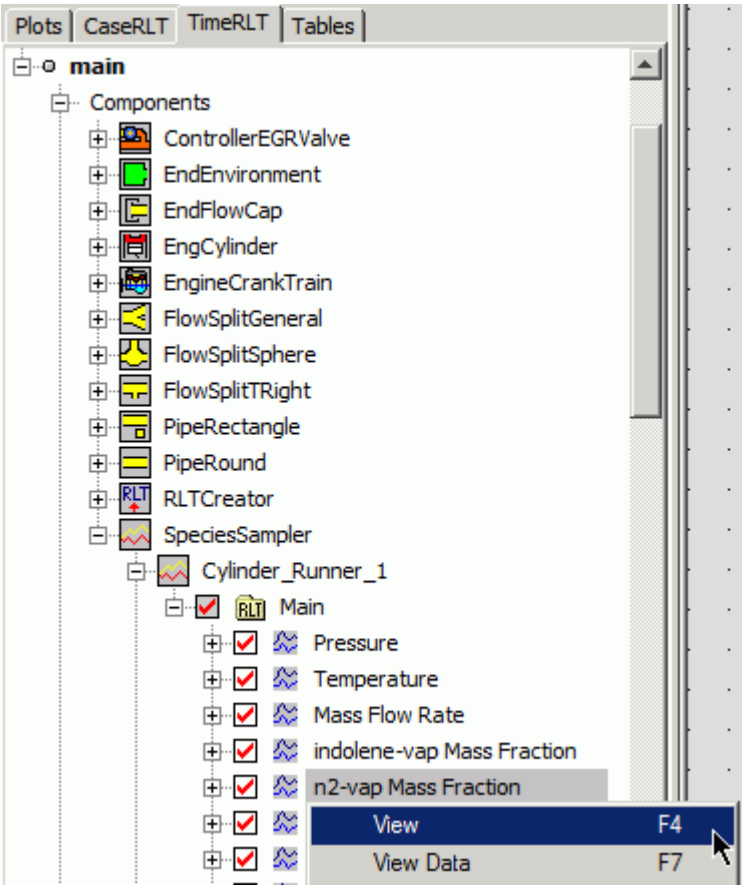
Use a similar procedure to visualize temperature:

16. Set the x- and y-axis data for the RLT plot as summarized below:

Cell	Node	Data
X	**RLT tree** > **Special**	**Cycle Number**
Y1	**FlowSplitTRight** > **InMan01** > **Thermal**	**Temperature at End of Cycle**
Y2	**FlowSplitTRight** > **InMan02** > **Thermal**	**Temperature at End of Cycle**
Y3	**FlowSplitTRight** > **InMan03** > **Thermal**	**Temperature at End of Cycle**
Y4	**FlowSplitTRight** > **InMan04** > **Thermal**	**Temperature at End of Cycle**
Y5	**FlowSplitGeneral** > **egr-fs-cfd** > **Thermal**	**Temperature at End of Cycle**
Y6	**PipeRound** > **InPipe-cfd** > **Thermal**	**Temperature at End of Cycle (Outlet)**

To visualize the species mass fractions:

17. Select the *TimeRLT* tab.
18. Right-click the **Components** > **SpeciesSampler** > **Cylinder_Runner_1** > **Main** > **n2-vap Mass Fraction** node and select **View**.



When complete, GT-POST shows that the initial n2-vap mass fraction is approximately 0.60.

19. Repeat the above procedure for each gas component, to visualize the remaining mass fractions.

    Trace components (<1%) are not included in the initial composition. All trace components are included in the n2-vap mass fraction, as it is the largest of the gas components and the total mass fraction must equal 1.

    The mass fractions are summarized in the table below:

Gas Component	Mass Fraction
n2-vap (Fresh Nitrogen)	0.601
o2-vap (Fresh Oxygen)	0.183
prod_n2 (Burnt Nitrogen)	0.154
prod_co2 (Carbon Dioxide Products)	0.043
prod_h2o (Water Products)	0.016
prod_o2 (Burnt Oxygen)	0.001

20. Close GT-POST.

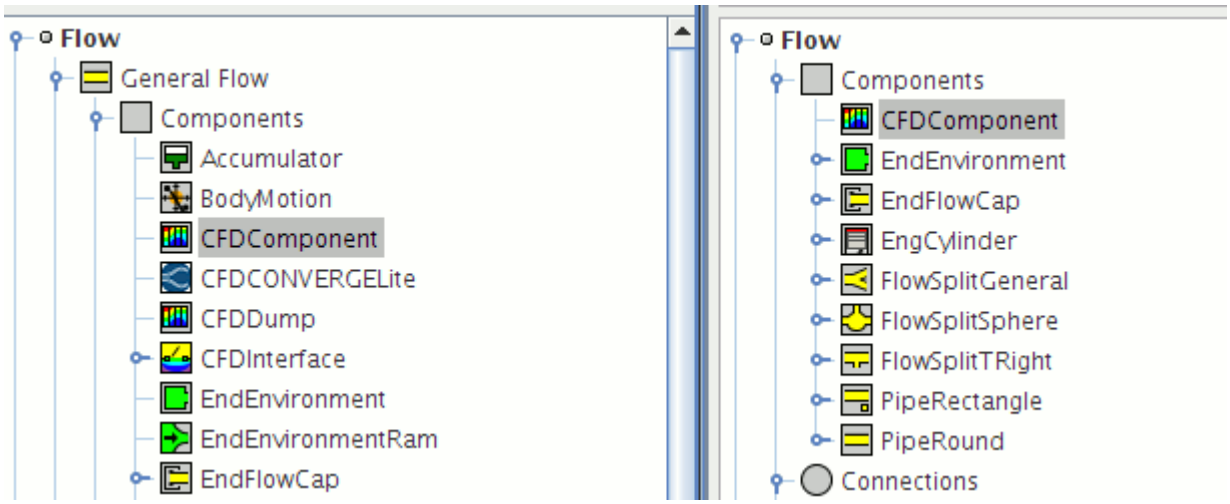
Adding CFD Objects to the Project Tree

In GT-SUITE, you define the co-simulation interface using three types of objects: CFD Interfaces, CFD Components, and CFD Connections.

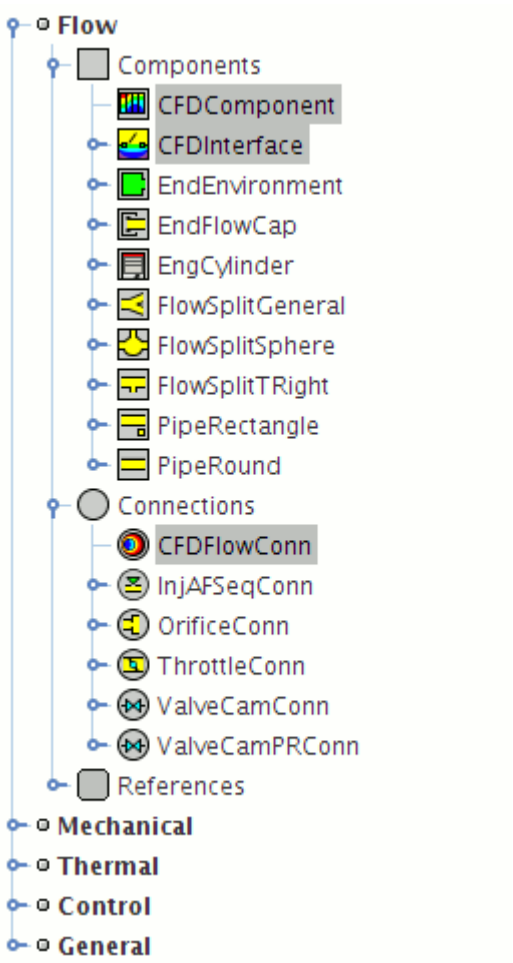
To add the required CFD objects to the project tree:

1. In the *Templates and Search* ribbon, click **Template Library**.

2. In the *Template Library* tree that appears on the left, expand the **Flow** > **General Flow** > **Components** node.
3. Drag and drop the **CFDComponent** node from the *Template Library* tree to the project tree.



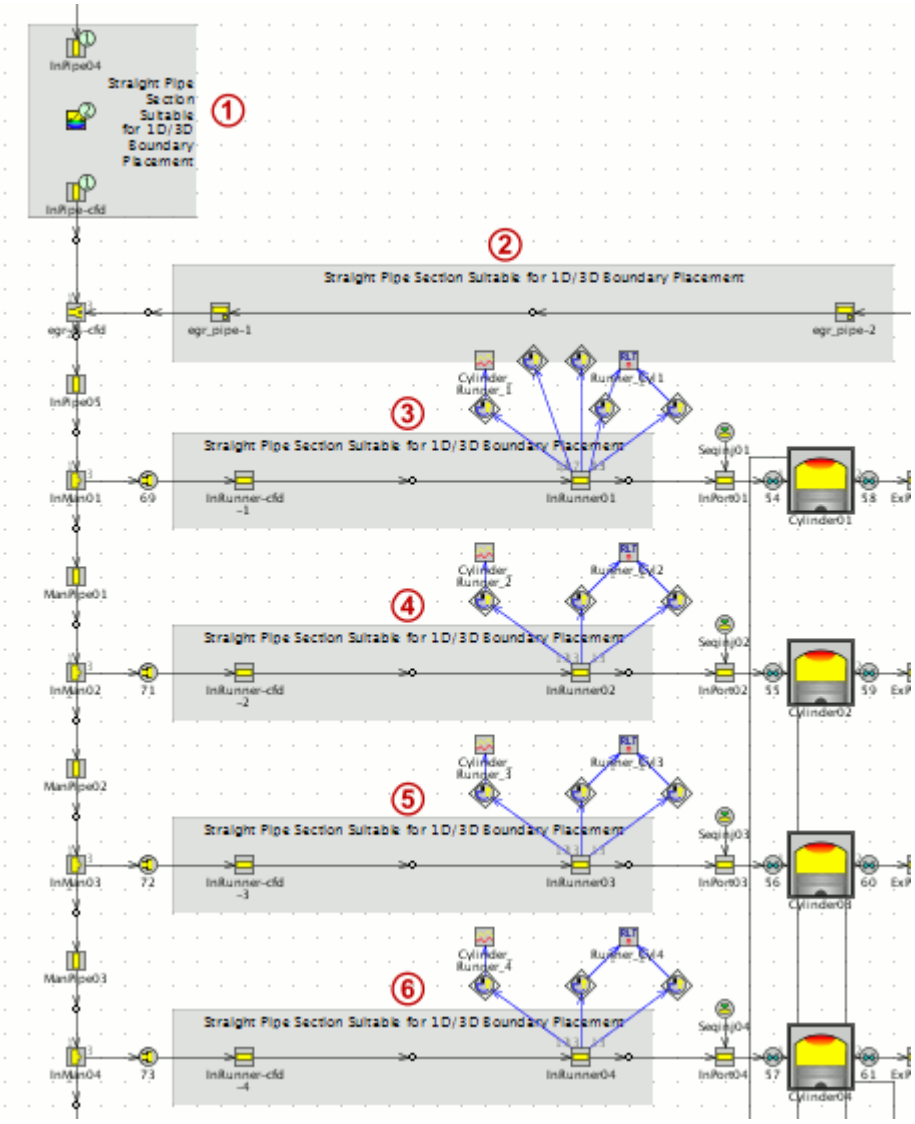
4. Similarly, drag and drop the **General Flow** > **Components** > **CFDInterface** node, and the **General Flow** > **Connections** > **CFDFlowConn** node.



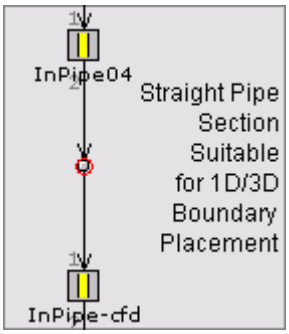
Adding CFDInterfaces to the Model Map

In the model map, you mark the 1D/3D interfaces between GT-SUITE and STAR-CCM+ using CFDInterface objects.

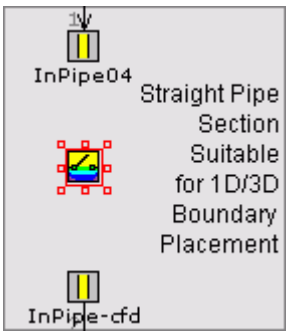
1. In the model map, locate the six pipe sections that require CFD Interfaces (in this case, they are marked as *Straight Pipe Section Suitable for 1D/3D Boundary Placement*).



2. For each of the six pipe sections, replace the default 1D orifices with 1D/3D **CFDInterface** objects using the following procedure:
   a) Delete the default 1D orifices.

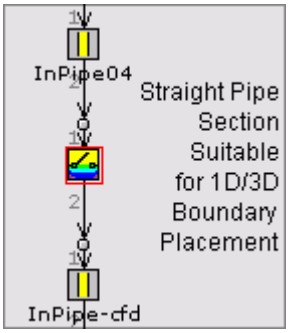


   b) In the project tree, expand the **Components** > **CFDInterface** node.
   c) Drag and drop the **def** node onto the model map and place it at the original orifice position.



d) Right-click the CFDInterface on the model map and select **Start New Link**.

e) Use the **Start New Link** tool to join the CFDInterface to the adjacent pipes, making sure that the arrows point in the direction of flow.



During the uncoupled cycles, GT-SUITE solves the flow in the one-dimensional components. When coupling commences, the GT-SUITE solver supplies boundary conditions to STAR-CCM+ and no longer solves the flow in the one-dimensional components.

3. Save the GT-SUITE model.

Adding CFD Connections to the Model Map

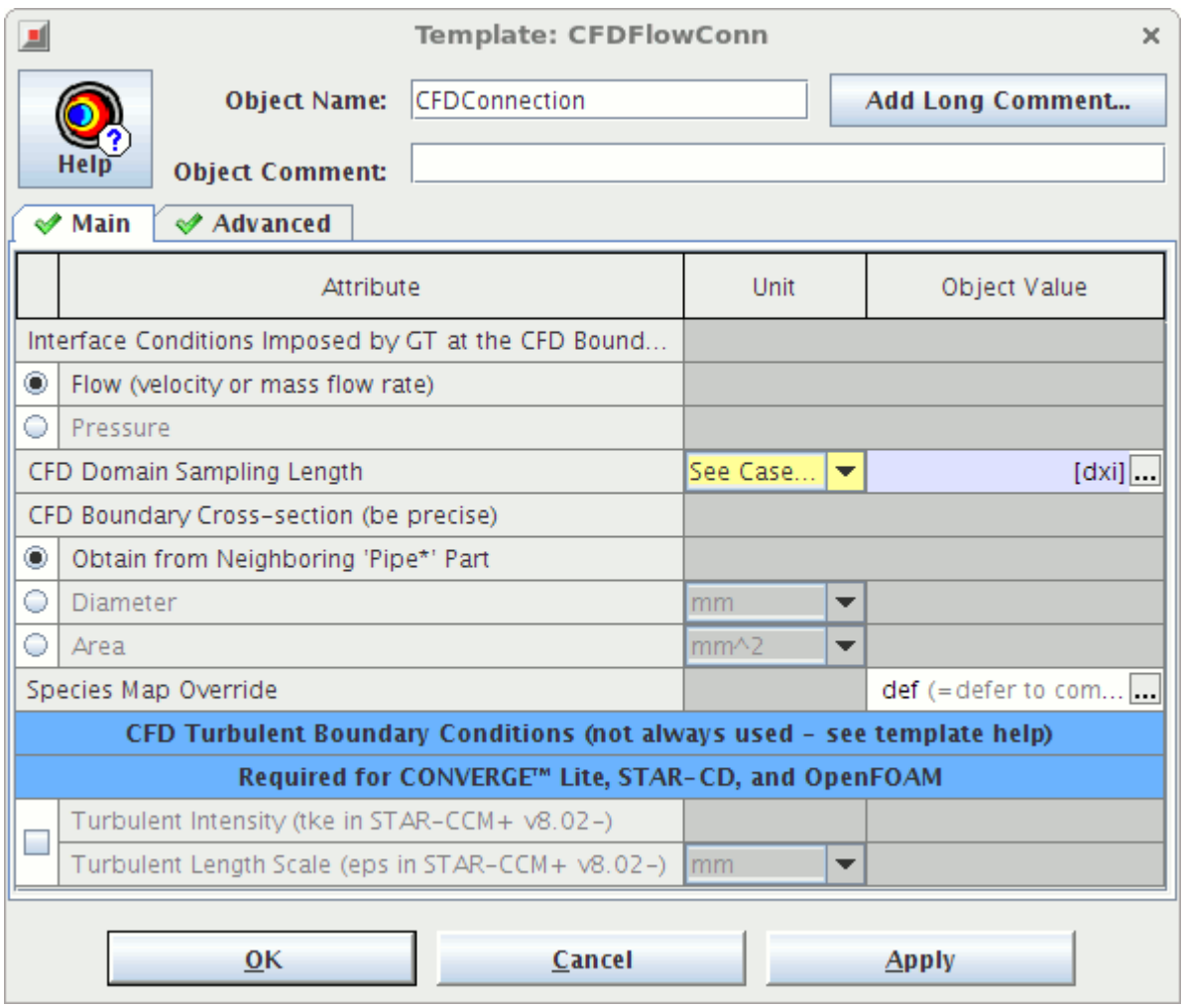
In GT-SUITE, you define the characteristics of the 1D/3D interfaces using CFD Connection objects. CFD Connections specify the exchanged data, the STAR-CCM+ volume-averaging zone length, and how to determine the cross-section of the coupled STAR-CCM+ boundary.

To add the required CFDFlowConn connections to the model map:

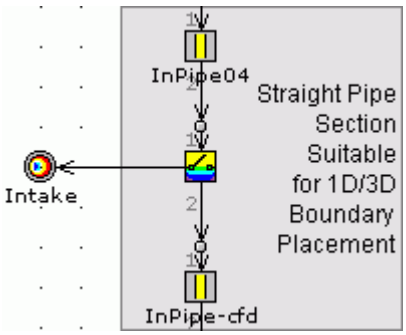
1. In the project tree, right-click the **Connections** > **CFDFlowConn** node and select **Add Object...**
2. In the *Template: CFDFlowConn* dialog, click the *Main* tab and set the following properties:

Property/Attribute	Setting
Object Name	`CFDConnection`
Interface Conditions Imposed by GT at the CFD Boundary	**Flow (velocity or mass flow rate)**
CFD Domain Sampling Length	*Object Value*: **[dxi]**

Property/Attribute	Setting
	This sets the STAR-CCM+ volume-averaging zone length to the discretization length in GT-SUITE. Generally, use a value greater than `25 mm` to prevent oscillations at the coupled boundary.
	Unit: **See Case Setup** (selected automatically)
CFD Boundary Cross-section (be precise)	**Obtain from Neighboring 'Pipe*' Part** In general, GT-SUITE permits the use of pipes with circular or rectangular cross-sections. For cross-sections that do not meet these requirements, calculate the effective diameter. You can find this effective diameter by using a **Sum** report of the *Area* field function at the coupled boundary in STAR-CCM+.
Species Map Override	**def**



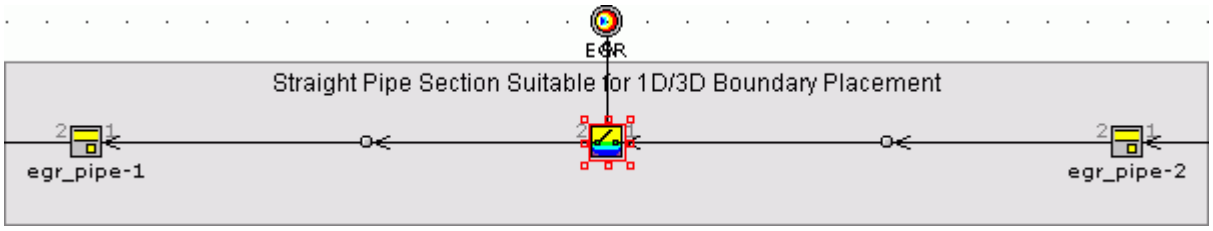
3. In the *Template: CFDFlowConn* dialog, click the *Advanced* tab and make sure that *Orifice Diameter* is set to **def**, which matches the value to the diameter of the adjacent pipe in the GT-SUITE model.
4. Click **OK**.
5. Drag and drop the **CFDFlowConn** > **CFDConnection** node from the project tree on to the model map and place it near the CFDInterface between **InPipe04** and **InPipe-cfd**.
6. Rename the CFDConnection to `Intake`.
7. Join the components from the **CFDInterface** to the **CFDConnection**.



8. Create another CFDConnection, with the following settings:

Property/Attribute	Setting
Object Name	`CFDConnEGR`
Interface Conditions Imposed by GT at the CFD Boundary	**Flow (velocity or mass flow rate)**
CFD Domain Sampling Length	*Object Value*: **[dxegr]**
	Unit: **See Case Setup** (selected automatically)
CFD Boundary Cross-section (be precise)	**Obtain from Neighboring 'Pipe*' Part**
Species Map Override	**def**

9. Place and join the **CFDConnEGR** connection between the **egr_pipe-1** and **egr_pipe-2**, and rename it to `EGR`.

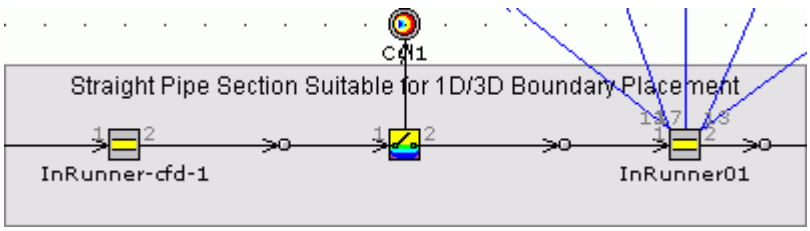


10. Create a final CFDConnection, with the following settings:

Property/Attribute	Setting
Object Name	`CFDConnCyl`

Property/Attribute	Setting
Interface Conditions Imposed by GT at the CFD Boundary	**Flow (velocity or mass flow rate)**
CFD Domain Sampling Length	*Object Value*: **[dxi]**
	Unit: **See Case Setup** (selected automatically)
CFD Boundary Cross-section (be precise)	**Obtain from Neighboring 'Pipe*' Part**
Species Map Override	**def**

11. Place and join four **CFDConnCyl** connections between the remaining 1D/3D sections (for example, between **InRunner-cfd-1** and **InRunner01**), and rename them to `Cyl1`, `Cyl2`, `Cyl3`, and `Cyl4`.



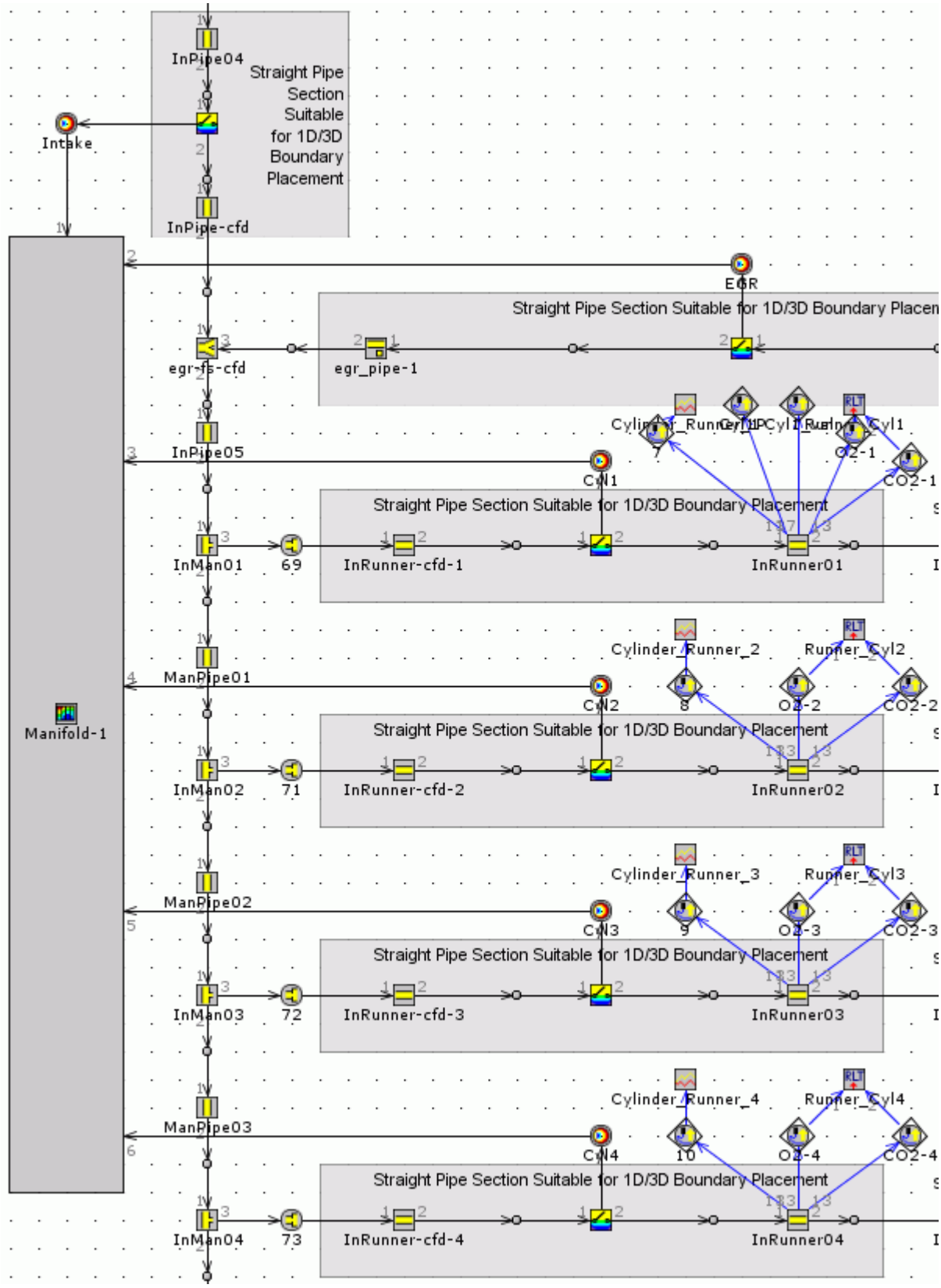
12. Save the GT-SUITE model.

Adding a CFD Component to the Model Map

In the model map, you link all the CFD Connections to a **CFDComponent** object. A CFDComponent defines several global settings, such as the number of pre-cycles to run before coupling with STAR-CCM+, and species mapping.

To add a CFDComponent to the model map:

1. In the project tree, right-click the **Components** > **CFDComponent** node and select **Add Object...**
2. In the *Create Object* dialog:
   a) Set *Object Name* to `Manifold`.
   b) In the *Simulation Phases* tab, set *Engine Cycles or Time Before CFD (or "cfdreload") Started* to `20`.
   This setting instructs GT-SUITE to run 20 pre-cycles before coupling with STAR-CCM+.
   c) Click **OK**.
3. Drag and drop the **CFDComponent** > **Manifold** node on to the model map and link all six CFDConnections to it. After resizing the **Manifold** component, the map appears as shown below.


InPipe04
Straight Pipe Section Suitable for 1D/3D Boundary Placement
Intake
InPipe-cfd
EGR
Straight Pipe Section Suitable for 1D/3D Boundary Placem
egr-fs-cfd
egr_pipe-1
InPipe05
Cyl1
CO2-1
Straight Pipe Section Suitable for 1D/3D Boundary Placement
InMan01
69
InRunner-cfd-1
InRunner01
Cylinder_Runner_2
Runner_Cyl2
ManPipe01
Cyl2
8
O2-2
CO2-2
Manifold-1
Straight Pipe Section Suitable for 1D/3D Boundary Placement
InMan02
71
InRunner-cfd-2
InRunner02
Cylinder_Runner_3
Runner_Cyl3
ManPipe02
Cyl3
9
O2-3
CO2-3
Straight Pipe Section Suitable for 1D/3D Boundary Placement
InMan03
72
InRunner-cfd-3
InRunner03
Cylinder_Runner_4
Runner_Cyl4
ManPipe03
Cyl4
10
O2-4
CO2-4
Straight Pipe Section Suitable for 1D/3D Boundary Placement
InMan04
73
InRunner-cfd-4
InRunner04

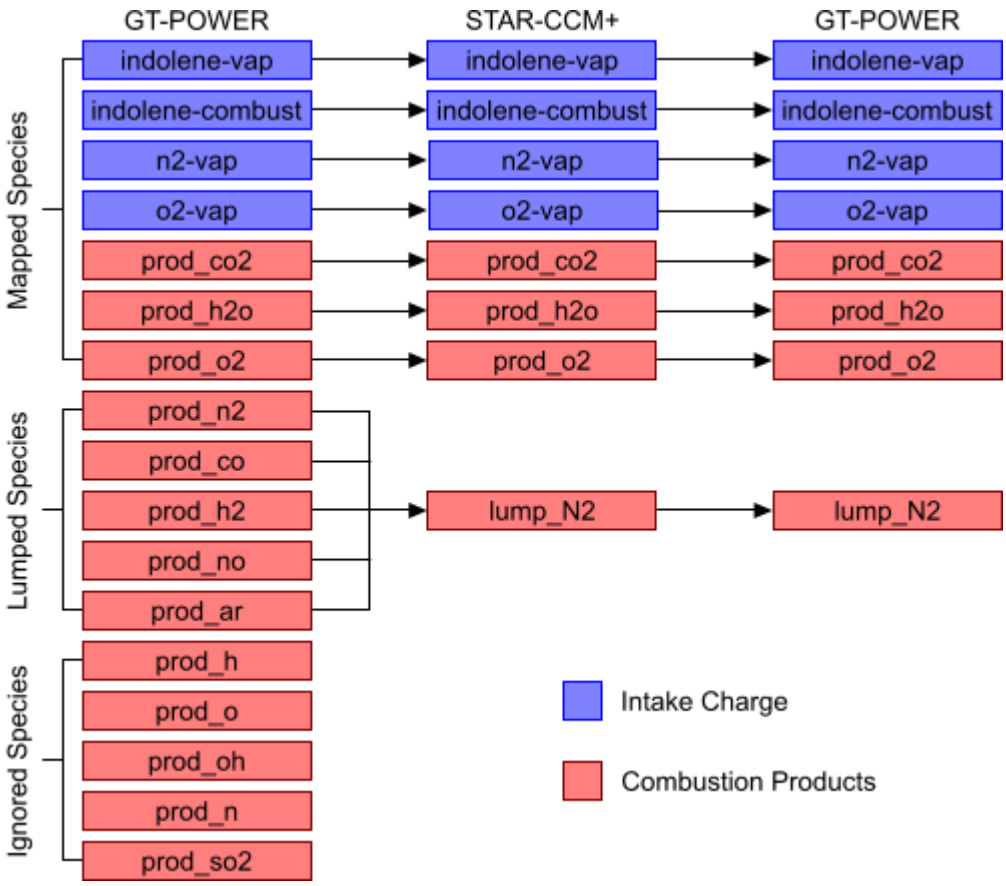
Defining Species Mapping

Define species mapping, so that GT-SUITE can map STAR-CCM+ gas components to its own species.

The advantage to using species mapping is that you do not have to represent every GT-SUITE species in the STAR-CCM+ simulation, which reduces the computational cost. Using species mapping also provides greater flexibility for you to choose which individual species are fully tracked in STAR-CCM+.

In this GT-SUITE model, there are four species that define the fresh air-fuel mixture (intake charge) entering the cylinder. In addition to these species, there are 13 products of combustion. Therefore, a total of 17 species are present in the GT-SUITE simulation. The intake charge species are always individually represented in STAR-CCM+. However, it is not necessary to track all burned species in STAR-CCM+, as some are only present in trace amounts. The lump_N2 scalar represents these negligible species in STAR-CCM+.

You map the species as illustrated in the diagram below:

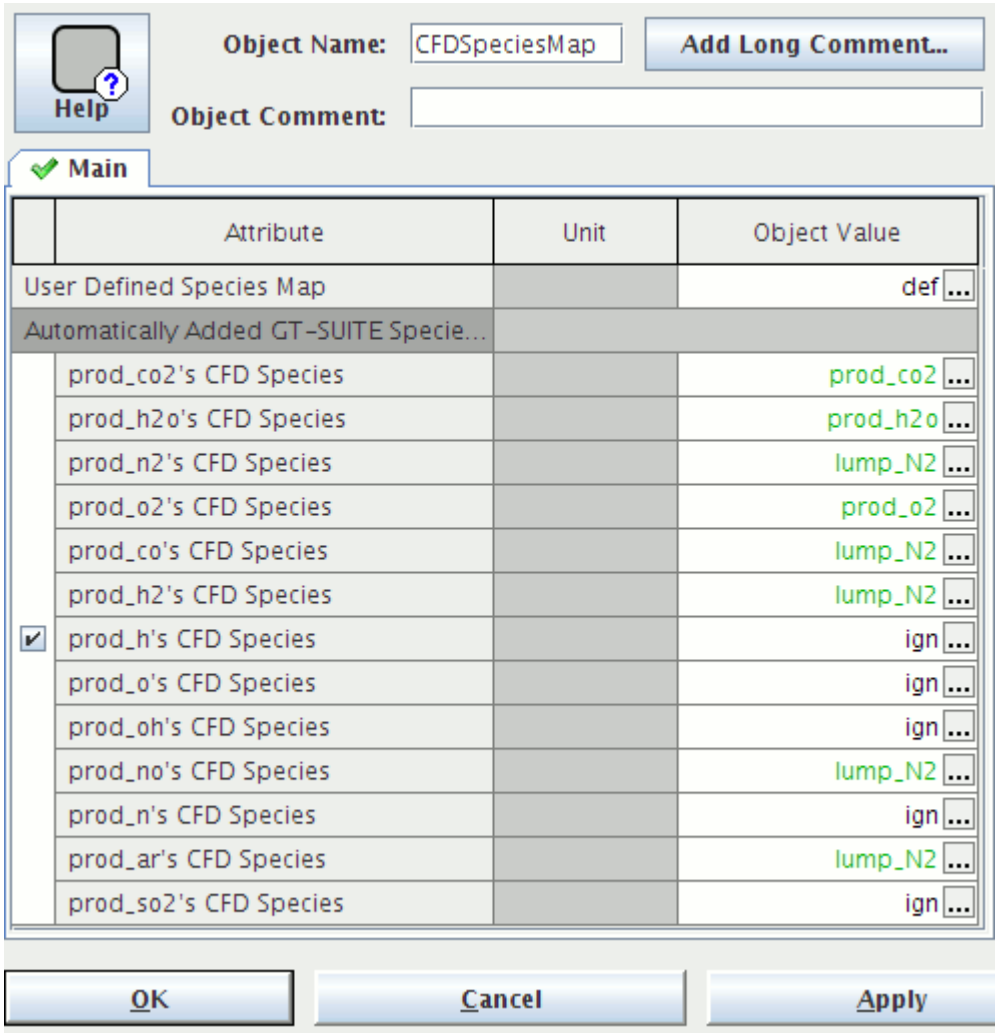


In this case, one species is always mapped to one cluster. This behavior is not a limitation of the mapping facility. For example, you could map all harmful gases to a cluster called Pollutants.

To implement the above species mapping:

1. Right-click the **Components** > **CFDComponent** > **Manifold** > **Manifold-1** node and select **Edit Properties...**
2. In the *Template: CFDComponent* dialog, select the *Coupled Cycles* tab.
3. Click the ellipsis (...) next to the *Species Mapping Object* attribute.

4. In the *Value Selector* dialog, select **CFDSpeciesMap** from the *Create New Object (Template Library)* group box and click **OK**.
5. In the *Template: CFDSpeciesMap* dialog:
   a) Set *Object Name* to **CFDSpeciesMap**.
   b) Set *User Defined Species Map* to `def`.
   c) Click the ellipsis (...) next to the *prod_co2's CFD Species* attribute.
   d) In the *Value Selector* dialog, select **CFDSpecies** > **prod_co2** from the *GT-SUITE Library* tab and click **OK**.
   e) Use a similar method to set the remaining attributes, as shown below:



6. Click **OK** to close the dialogs.
7. Save the GT-SUITE model.

Modifying the Run Setup

Modify the run setup to allow the GT-SUITE simulation to run 20 pre-cycles, and then 3 coupled cycles.

Make sure that the GT-SUITE simulation duration contains the coupled cycles. This practice allows the GT-SUITE solver to collect results for the coupled cycles for post-processing within GT-POST.

Previously, you ensured the solver would stop when the solution was steady state. However, you begin coupling to STAR-CCM+ after the GT-SUITE solution reaches a periodic steady state. Allow the solver to continue after the steady state criteria are met, and set the simulation to run 20 pre-cycles, plus 3 coupled cycles, which requires 23 cycles in total.

To modify the run setup:

1. In the *Model Setup* ribbon, click **Run Setup**.
2. In the *Run Setup* dialog, select the *TimeControl* tab.
3. To run the solver for the required number of engine cycles, set *Maximum Simulation Duration (Cycles)* to `23`.

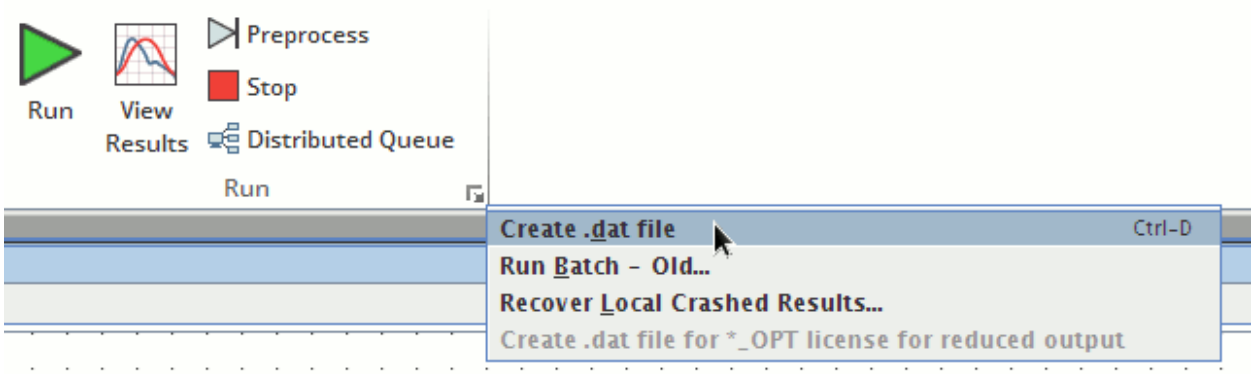
   This value gives enough cycles for the GT-SUITE simulation to converge (20), plus the required number of coupled cycles (3).
4. Set *Automatic Shut-Off When Steady-State* to **off**, so that the solver runs for the maximum number of cycles, without terminating prematurely.
5. Click **OK**.
6. Save the GT-SUITE model.

Creating the .dat file

After specifying the required settings, you create a .dat file with the simulation setup. This file provides an input file for the GT-SUITE solver.

To create the `.dat` file:

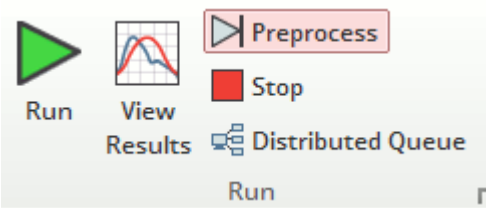
1. Click the arrow at the bottom-right of the *Run* ribbon and select **Create .dat file**.



2. In the *.dat File Warning* dialog, click **Yes** to overwrite the existing results.

To make sure that the model is valid, run the simulation preprocessing feature. This action generates a file containing materials properties of all species in the GT-SUITE model, which can be used in STAR-CCM+ to define identical materials.

3. In the *Run* ribbon, click **Preprocess**.



4. In the *.dat File Warning* dialog, click **Yes** to overwrite the existing results.

   If the model is valid, the solver preprocessor exits with no errors or warnings.
5. Make sure that the `CFDCoupling_Setup.chemkin_GT.dbs` file is present in the current working directory, as this file is required at a later stage of the tutorial.
6. Close GT-SUITE.

Setting Up the STAR-CCM+ Simulation

Prepare the intake manifold simulation in STAR-CCM+ and modify it for co-simulation with GT-SUITE.

Contents:

Importing the Volume Mesh
Selecting Physics Models
Importing the GT-SUITE Material Properties
Setting Up the Species Representation
Setting the Reference Pressure
Setting Initial Conditions
Defining Boundary Conditions
Specifying the Coupled Boundaries and Material Components
Setting the Time-Step
Setting Stopping Criteria
Plotting the Mass Flow for Each Cylinder
Plotting the Mass Fraction of Fresh Air at Cylinder 1
Visualizing the Velocity Field

Importing the Volume Mesh

To set up the STAR-CCM+ simulation, launch a simulation and import the supplied volume mesh.

1. Launch STAR-CCM+.
2. Start a simulation.
3. Import the `manifoldGT-POWER.ccm` volume mesh from your working directory.

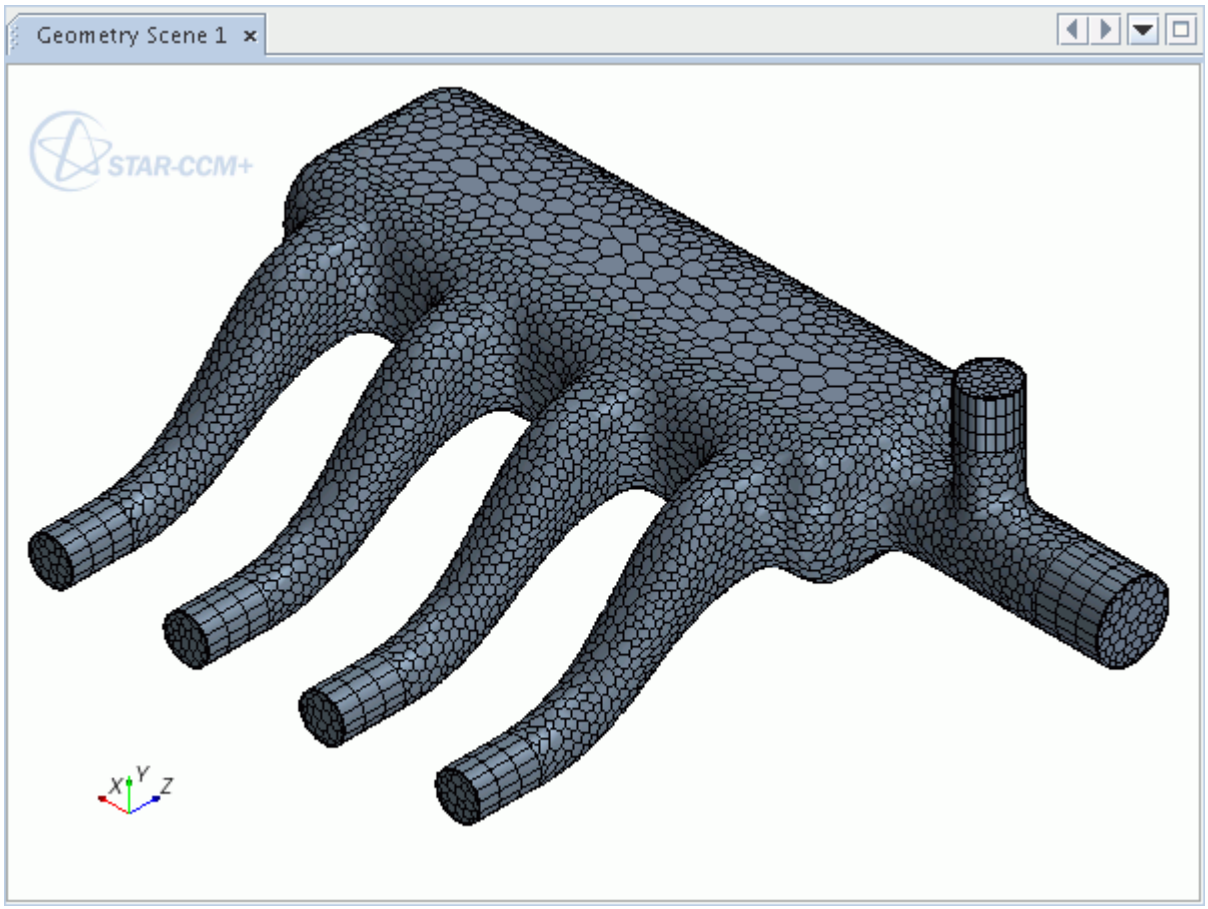
   Note: When creating this volume mesh, the Extruder was used to generate orthogonal cells in each averaging zone. This mesh helps to capture the one-dimensional nature of the flow in these zones.

STAR-CCM+ provides feedback on the import process in the *Output* window and a mesh region is created under the **Regions** node.

Display the imported volume mesh and store a convenient view:

4. Create a mesh scene.

5. Click (**Save-Restore-Select views**) and select **Views** > **-X+Y-Z** > **Up +Y**.



6. Click (**Save-Restore-Select views**) and select **Store Current View**, which saves this view for later use.
7. Save the simulation as `manifold-gt.sim`.

Selecting Physics Models

Select the physics models that solve for the fluid in STAR-CCM+. Also, activate a physics model that enables coupling between STAR-CCM+ and GT-SUITE.

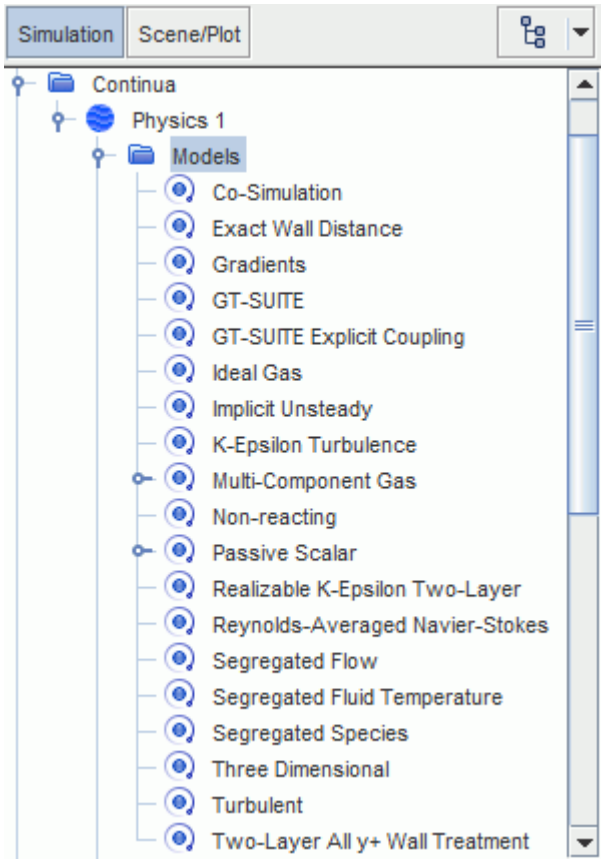
Physics models determine the behavior of the flow in the simulation. In this tutorial, activate the physics models for the fluid region and the GT-SUITE Co-Simulation model.

To select the physics models:

1. For the physics continuum, **Continua** > **Physics 1**, select the following models in order:

Group box	Model
Enabled Models	**Three-Dimensional** (selected automatically)
Time	**Implicit Unsteady**
Material	**Multi-Component Gas**
Reaction Regime	**Non-reacting**
Flow	**Segregated Flow**
Enabled Models	**Segregated Species** (selected automatically)
	Gradients (selected automatically)
Equation of State	**Ideal Gas**
Energy	**Segregated Fluid Temperature**
Viscous Regime	**Turbulent**
Enabled Models	**Reynolds-Averaged Navier-Stokes** (selected automatically)
Reynolds-Averaged Turbulence	**K-Epsilon Turbulence**
Enabled Models	**Realizable K-Epsilon Two-Layer** (selected automatically)
Enabled Models	**Two-Layer All y+ Wall Treatment** (selected automatically)
Optional Models	**Co-Simulation**
Co-Simulation Models	**GT-SUITE**
Enabled Models	**GT-SUITE Explicit Coupling** (selected automatically)
Optional Models	**Passive Scalar**

2. Click **Close**.
3. To review the models, open the **Physics 1** > **Models** node.



4. Save the simulation.

Importing the GT-SUITE Material Properties

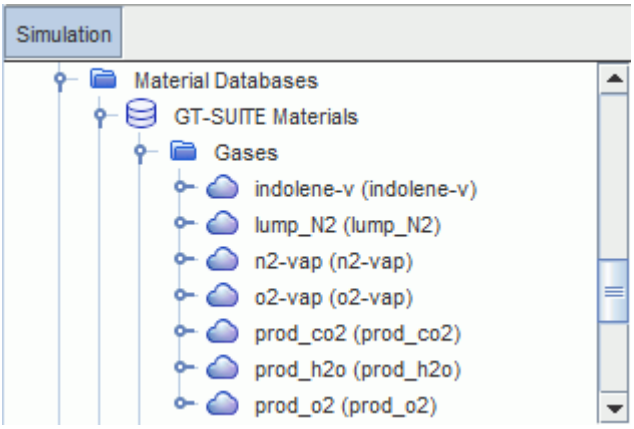
Create a material database and import the material properties from the GT-SUITE simulation.

When preprocessing the GT-SUITE solver, you created a file, `CFDCoupling_Setup.chemkin_GT.dbs`, that contains a database of material properties from the GT-SUITE simulation. You can import these properties into STAR-CCM+, so that you use identical materials in both codes.

To import the GT-SUITE material properties:

1. Right-click the **Tools** > **Material Databases** node and select **New Material Database...**
2. Rename the **matl** node as `GT-SUITE Materials`.
3. Right-click the **GT-SUITE Materials** node, and select **Import Properties...**
4. In the *Open* dialog, navigate to the `caseFilesGT-POWER` directory, select `CFDCoupling_Setup.chemkin_GT.dbs`, and click **Open**.
5. In the *Import Properties* dialog, select **Import properties for all materials** and click **OK**.
6. Open the **GT-SUITE Materials** > **Gases** node.

   The GT-SUITE material properties appear as shown below:



Setting Up the Species Representation

Define gas components and passive scalars under the relevant physics models.

To reduce computational costs, you use six mixture components and two passive scalars to represent the species that GT-SUITE transfers to STAR-CCM+.

The passive scalars represent species that are only found in small amounts (that is, less than 1% of the volume average). Therefore, it is safe to assume that they have little influence on the bulk fluid and can be modeled as passive.

The gas components and passive scalars are summarized below:

Gas Components	Passive Scalars
n2-vap (Fresh Nitrogen)	indolene-vap (Gaseous Fuel)
o2-vap (Fresh Oxygen)	indolene-combust (Liquid Fuel)
lump_N2 (Burnt Nitrogen and lumped, trace species)	
prod_o2 (Burnt Oxygen)	
prod_co2	
prod_h2o	

Contents:

Defining Gas Components
Defining Passive Scalars

Defining Gas Components

Define and reorder the gas components, and change the specific heat so that it uses the thermodynamic polynomial data from the GT-SUITE materials database file.

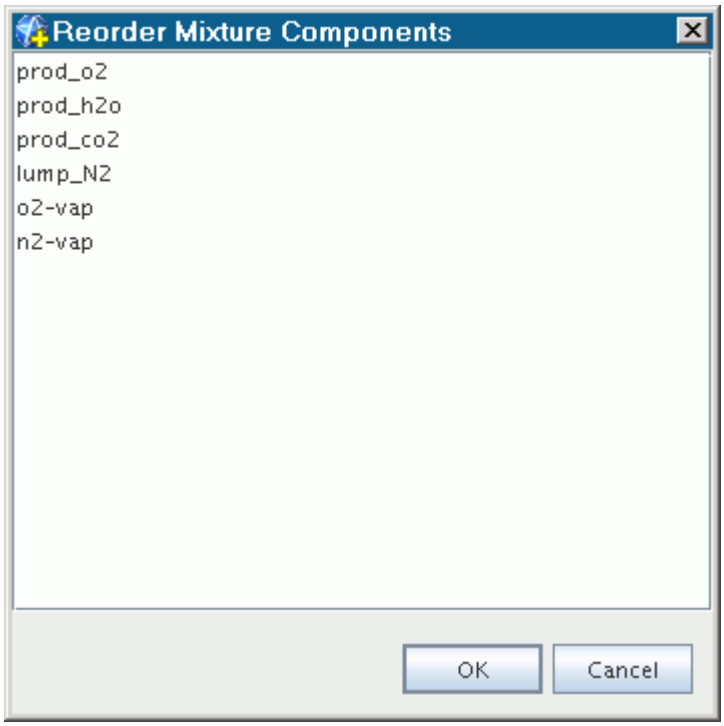
In the following steps, you define and reorder the gas components so that the material with the largest expected volume fraction, in this case n2-vap, is last in the list. This ordering improves numerical accuracy.

To define the gas components:

1. Expand the **Continua** > **Physics 1** > **Models** > **Multi-Component Gas** node.
2. Right-click **Gas Components** and select **Select Mixture Components...**
3. In the *Select Mixture Components* dialog, expand the **Material Databases** > **GT-SUITE Materials** > **Gases** node.
4. Activate the following components:
   - lump_N2
   - n2-vap
   - o2-vap
   - prod_co2
   - prod_h2o
   - prod_o2
5. Click **Apply** then **Close**.

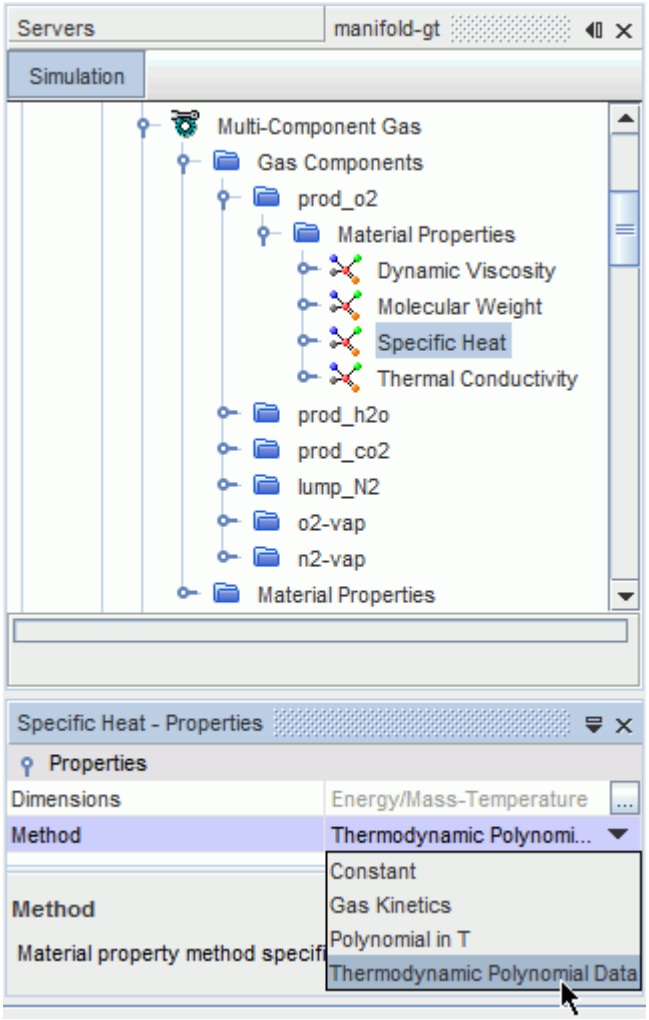
To reorder the gas components:

6. Right-click the **Gas Components** node and select **Reorder Mixture Components...**
7. Click and drag the components so that they are in the following order:



Change the specific heat so that it uses the thermodynamic polynomial data from the GT-SUITE materials database file, as specific heat is defined this way in GT-SUITE.

8. For each gas component, select the **Material Properties** > **Specific Heat** node, and set *Method* to **Thermodynamic Polynomial Data**.



9. Save the simulation.

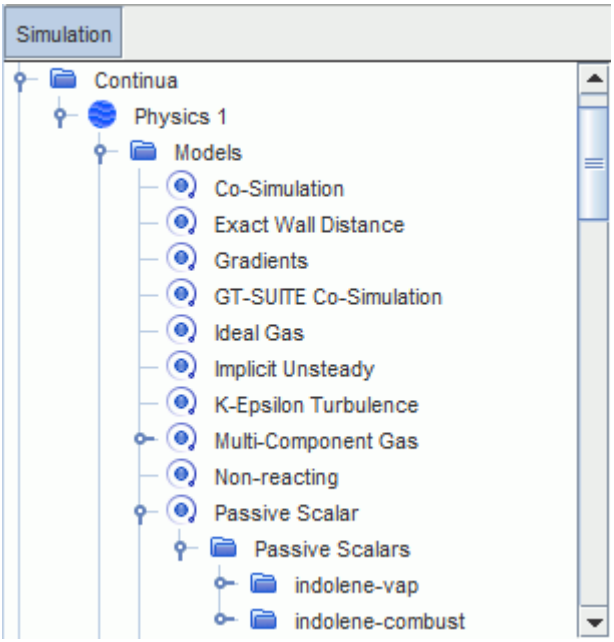
Defining Passive Scalars

Define passive scalars to represent species that are only found in small amounts (that is, less than 1% of the volume average).

To define passive scalars:

1. Right-click the **Continua** > **Physics 1** > **Models** > **Passive Scalar** > **Passive Scalars** node and select **New**.
2. Rename the **Passive Scalar** as `indolene-vap`.
3. Create another passive scalar called `indolene-combust`.

   The nodes corresponding to these passive scalars are as shown below.



4. Save the simulation.

Setting the Reference Pressure

Set the reference pressure to the value obtained from the independent GT-SUITE run.

STAR-CCM+ uses a reference pressure to reduce round-off error, with all other pressures in the simulation defined relative to its value. However, in GT-SUITE, all pressures are absolute.

To minimize the numerical round-off, set the reference pressure to the GT-SUITE pressure at the start of coupling. The reference pressure that you use in this tutorial comes from the independent GT-SUITE run at the start of this tutorial.

To set the reference pressure:

1. Select the **Continua** > **Physics 1** > **Reference Values** > **Reference Pressure** node and set *Value* to `2.13 bar`.

Setting Initial Conditions

Set the initial conditions to match the conditions in the GT-SUITE model at the start of coupling. This speeds up convergence at the start of the STAR-CCM+ simulation.

The initial conditions that you use in the following steps come from the independent GT-SUITE run at the start of this tutorial.

As stated in the previous section, pressure values in STAR-CCM+ are relative to the reference pressure. Therefore, you set the initial pressure to 0.0 bar, as you have already set the reference pressure accordingly.

To set the initial pressure, temperature, and turbulence settings:

1. Edit the **Continua** > **Physics 1** > **Initial Conditions** node, and set the following properties:

Node	Property	Setting
Pressure	*Value*	`0 bar`
Static Temperature	*Value*	`315 K`
Turbulence Specification	*Method*	**Intensity + Length Scale**
Turbulence Intensity	*Value*	`0.1`
Turbulent Length Scale	*Value*	`1 mm`

To set the initial mass fractions:

2. Select the **Initial Conditions** > **Species Mass Fraction** node. Click ... (**Custom Editor**) to set *Value* as follows:

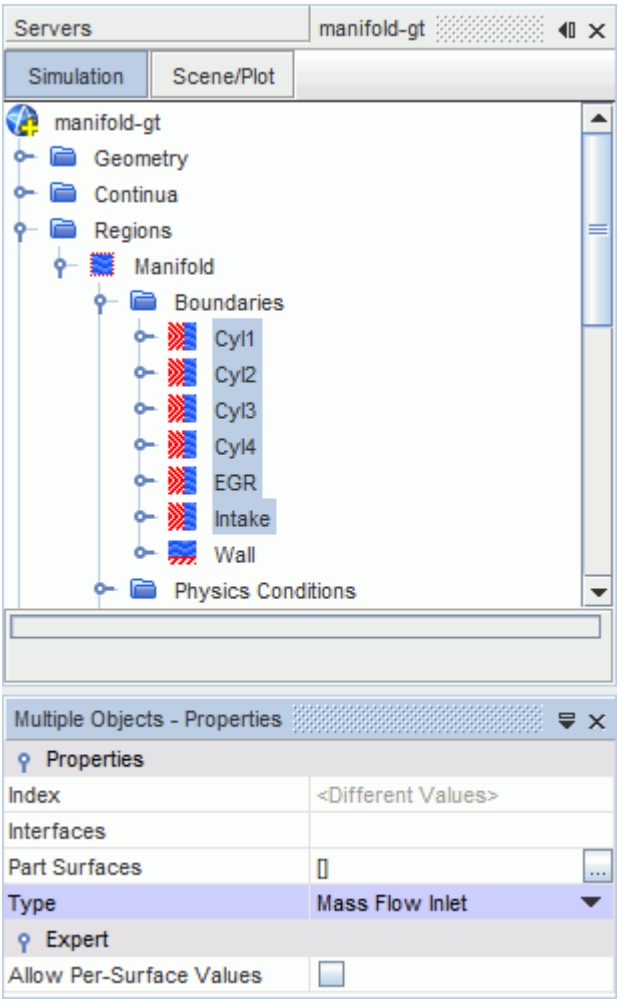
Gas Component	Mass Fraction
prod_o2	`0.01`
prod_h2o	`0.02`
prod_co2	`0.03`
lump_N2	`0.15`
o2-vap	`0.18`
n2-vap	`0.61`

3. Save the simulation.

Defining Boundary Conditions

Set the boundary types and specify turbulence settings at the flow boundaries. As GT-SUITE supplies all other boundary conditions, no other boundary specifications are required.

1. Open the **Regions** > **Manifold** > **Boundaries** node.
2. Multi-select the **Cyl1**, **Cyl2**, **Cyl3**, **Cyl4**, **EGR**, and **Intake** nodes, and set *Type* to **Mass Flow Inlet**.



3. Right-click any selected boundary and select **Edit**.
4. Set the following properties:

Node	Property	Setting
Physics Conditions > **Turbulence Specification**	*Method*	**Intensity + Length Scale**
Physics Values > **Turbulence Intensity**	*Value*	`0.05`

5. Select the **Intake** > **Physics Values** > **Turbulent Length Scale** node and set *Value* to `0.005 m`.
6. Select the **EGR** > **Physics Values** > **Turbulent Length Scale** node and set *Value* to `0.0032 m`.
7. Multi-select the **Cyl1**, **Cyl2**, **Cyl3**, and **Cyl4** nodes, right-click one of them and select **Edit**.
8. Select the **Physics Values** > **Turbulent Length Scale** node and set *Value* to `0.0035 m`.
9. Save the simulation.

Specifying the Coupled Boundaries and Material Components

Associate the STAR-CCM+ boundaries and material components with their GT-SUITE counterparts.

Start by creating a co-simulation zone for each of the CFD connections that you defined in GT-SUITE. Co-simulation zones allow you to couple STAR-CCM+ boundaries with the partner GT-SUITE CFD connections.

1. Expand the **Co-Simulations** > **Link 1** > **Zones** node.

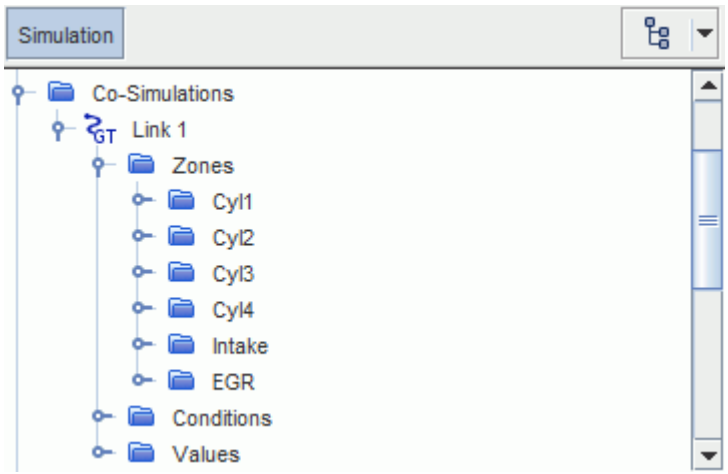
   STAR-CCM+ automatically added a co-simulation zone when you activated the **GT-SUITE** co-simulation model.

To successfully couple STAR-CCM+ and GT-SUITE, the names of the co-simulation zones in STAR-CCM+ must match the names of the CFD connections in GT-SUITE:

2. Rename **Zone 1** to `Cyl1`.

Create a co-simulation zone to couple the **Cyl2** boundary to the partner CFD connection, Cyl2:

3. Right-click the **Zones** node and select **New**. Rename the new zone to `Cyl2`.
4. Repeat the previous step to create 4 co-simulation zones named **Cyl3**, **Cyl4**, **Intake**, and **EGR**.



Assign the flow boundaries to the corresponding co-simulation zones, so they can exchange data with GT-SUITE:

5. Expand the **Regions** > **Manifold** > **Boundaries** node.
6. Select the **Cyl1** > **Physics Conditions** > **External Code Coupling Specification** node and set *External Code Coupling Specification* to **Link 1: Cyl1**.
7. Repeat the previous step to assign the remaining flow boundaries, **Cyl2**, **Cyl3**, **Cyl4**, **Intake**, and **EGR**, to the corresponding co-simulation zones.

GT-SUITE determines the species that are passed to STAR-CCM+ and handles the mapping of species from STAR-CCM+ back to GT-SUITE.

To couple the GT-SUITE species, *indolene-combust*, with the STAR-CCM+ passive scalar, **indolene-combust**:

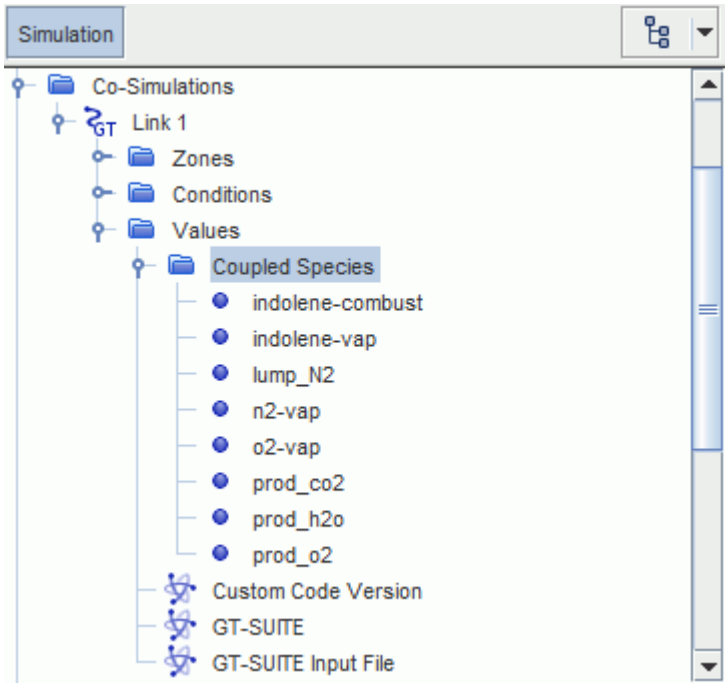
8. Expand the **Co-Simulations** > **Link 1** node.

Add a partner material component to represent the GT-SUITE species:

9. Right-click the **Values** > **Partner Material Components** node and select **New Partner Material Component**.

The name of the partner material component must match the name of the associated GT-SUITE species:

10. Rename **Material Component 1** to `indolene-combust`.
11. Select the **indolene-combust** node and set *Coupled STAR-CCM+ Mixture Component/Passive Scalar* to **indolene-combust**.
12. Repeat the previous steps to couple the remaining gas components and passive scalars to the corresponding GT-SUITE species.



13. Save the simulation.

Setting the Time-Step

Set the time step to a value of 5.55556E-5 seconds, which in this case is a one degree crank angle rotation.

The time step that GT-SUITE uses is not constant, but the STAR-CCM+ time step can be greater than the GT-SUITE time step during the simulation. This difference is possible because GT-SUITE can run several time steps while STAR-CCM+ is running one time step. However, for numerical reasons, it is recommended that the STAR-CCM+ time step is as close as possible to the GT-SUITE time step. As a general guideline, set the STAR-CCM+ time-step size to no more than three times the GT-SUITE time-step size.

It is recommended you round up the STAR-CCM+ time-step, so the elapsed duration of one cycle in STAR-CCM+ slightly exceeds the actual duration of the engine cycle. This rounding ensures the GT-SUITE solver completes the engine cycle before STAR-CCM+ terminates it.

To set the time step:

1. Select the **Solvers** > **Implicit Unsteady** node and set *Time-Step* to `5.55556E-5` s.

Setting Stopping Criteria

Set the stopping criteria to terminate the simulation after three engine cycles.

As the engine is running at 3000 rpm, each time step is a one degree crank angle rotation. Therefore, for three engine cycles, 2160 time steps are required. However, you run the simulation for a few extra time steps, as rounding errors in the time step duration can cause 720 time steps to be slightly less than a full cycle. When an incomplete cycle occurs, the GT-SUITE solver is stopped prematurely and output files are not generated correctly. Therefore, you use a total of 2162 time steps.

Also, the STAR-CCM+ solver runs iterations between time steps to solve the effect of transient boundary conditions. In this case, 5 inner iterations are adequate to guarantee convergence between time-steps.

To set the stopping criteria:

1. Edit the **Stopping Criteria** node and set the following properties:

Node	Property	Setting
Maximum Steps	*Maximum Steps*	`2162`
Maximum Inner Iterations	*Maximum Inner Iterations*	`5`

2. Save the simulation.

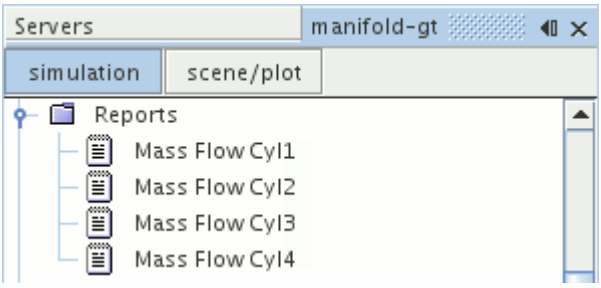
Plotting the Mass Flow for Each Cylinder

Create a mass flow report, monitor, and plot to analyze the fluid flow in and out of each cylinder.

To plot the mass flow for each cylinder:

1. Right-click the **Reports** node and select **New Report** > **Mass Flow**.
2. Rename the **Mass Flow 1** node to `Mass Flow Cyl1`.
3. Select the **Reports** > **Mass Flow Cyl1** node and set *Parts* to **Cyl1**.
4. Similarly, create three more reports to monitor mass flow in cylinders two, three and four, with the following names and boundaries:

Report	Boundary
Mass Flow Cyl2	**Cyl2**
Mass Flow Cyl3	**Cyl3**
Mass Flow Cyl4	**Cyl4**



5. Multi-select the **Mass Flow Cyl1**, **Mass Flow Cyl2**, **Mass Flow Cyl3**, and **Mass Flow Cyl4** nodes.
6. Right-click one of the selected nodes and select **Create Monitor and Plot from Report**.
7. In the *Create Plot From Reports* dialog, click **Single Plot**.
8. Save the simulation.

Plotting the Mass Fraction of Fresh Air at Cylinder 1

Create a surface average report, monitor, and plot for the mass fraction of fresh air at Cylinder 1.

First, define a field function to calculate the fresh air mass fraction:

1. Right-click the **Tools** > **Field Functions** node and select **New** > **Scalar**.
2. Rename the **User Field Function 1** node to `Fresh Air`.
3. Select the **Fresh Air** node and set the following properties:

Property	Setting
Function Name	`FreshAir`
Definition	`${MassFractionn2-vap}+${MassFractiono2-vap}`

The braces in the definition are required as the field function names use hyphens, which the code interprets as a minus operation.

To plot the mass fraction of fresh air at cylinder 1:

4. Right-click the **Reports** node and select **New Report** > **Surface Average**.
5. Rename the **Surface Average 1** node to `Fresh Air`.
6. Edit the **Fresh Air** node and set the following properties:

Property	Setting
Parts	**Cyl1**
Scalar Field Function	**Fresh Air**

7. Right-click the **Fresh Air** report and select **Create Monitor and Plot from Report**.
8. Save the simulation.

Visualizing the Velocity Field

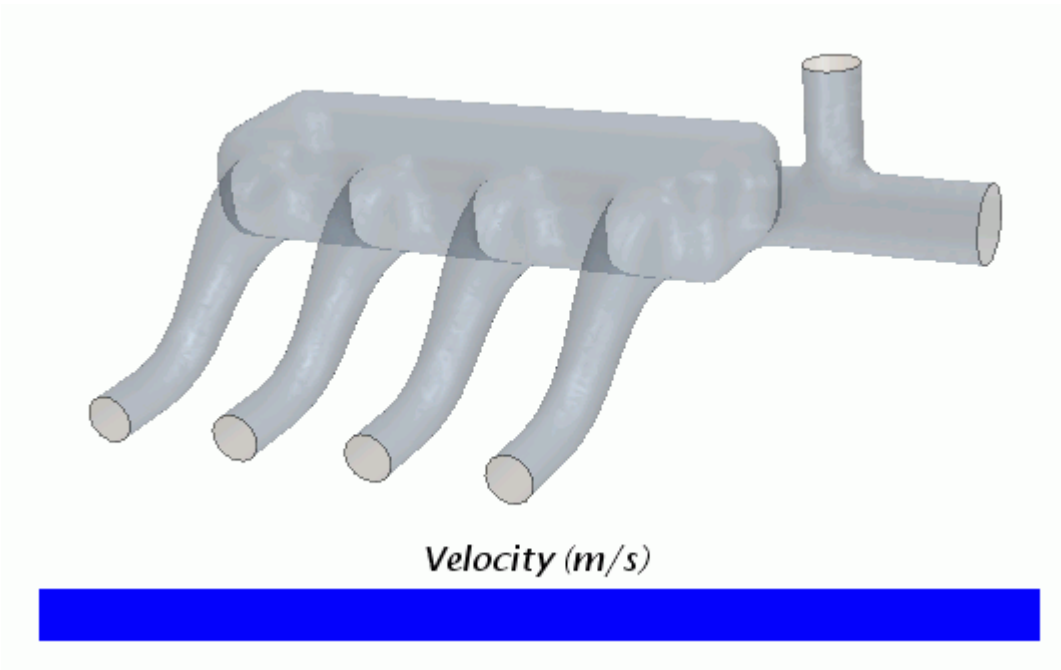
Create a vector scene that displays the velocity field at the mass flow boundaries.

1. Create a vector scene.
2. Edit the **Displayers** node and set the following properties:

Node	Property	Setting
Outline 1	*Surface*	**Activated**

Node	Property	Setting
	Opacity	0.3
Vector 1 > **Parts**	*Parts*	**Cyl1**
		Cyl2
		Cyl3
		Cyl4
		EGR
		Intake

3. Click (**Save-Restore-Select views**) and select **Restore View** > **View 1**.



4. Save the simulation.

Coupling STAR-CCM+ to GT-SUITE

Instruct STAR-CCM+ to launch GT-SUITE in server mode and connect to it. Use the average zone flag field function to visualize the areas where the flow is approximately one dimensional.

To set up the communication, specify the version of GT-SUITE (in this tutorial, v2017), the location of the GT-SUITE .dat file, and the method for launching and connecting to the GT-SUITE server:

1. Expand the **Co-Simulations** > **Link 1** > **Conditions** node and set the following properties:

Node	Property	Setting
GT-SUITE Version	*Version*	**v2017**

Node	Property	Setting
Launch Partner Option	*Option*	**Launch Default Application**

2. Expand the **Link 1** > **Values** node.
3. Select the **GT-SUITE Model File** node and set *GT-SUITE Input File* to the path to the GT-SUITE .dat file. If you set up your working directory as suggested, the path is: `GT-SUITE_caseFiles/CFDCoupling_Setup.dat`.

   You can type the file path and name manually, or you can click ... (**Custom Editor**) and use the dialog to navigate to the file.

In order to launch GT-SUITE in server mode, STAR-CCM+ requires you to specify the location of the GT-SUITE installation and license file using two environment variables, `$GTIHOME` and `$GTISOFT_LICENSE_FILE`, respectively.

- If you did not define the required global variables on your system, specify them locally by setting the following properties:

Node	Property	Setting
GTIHOME Environment Variable	*GTIHOME*	`[Path to GT-SUITE Installation]`
GTISOFT_LICENSE_FILE Environment Variable	*GTISOFT_LICENSE_FILE*	`[Path to GT-SUITE License]`

4. Right-click the **Link 1** node and select **Launch and Connect**.

   The GT-SUITE solver is started in the background, waiting for STAR-CCM+ to begin running its solver. Clearing the solution stops the GT-SUITE solver and breaks the connection to STAR-CCM+. If you clear the solution, restart the GT-SUITE solver as outlined above.

Visualize the volume where physics values and mass fractions are averaged and sent to GT-SUITE. Inspect the zone and determine whether the flow is relatively one dimensional.

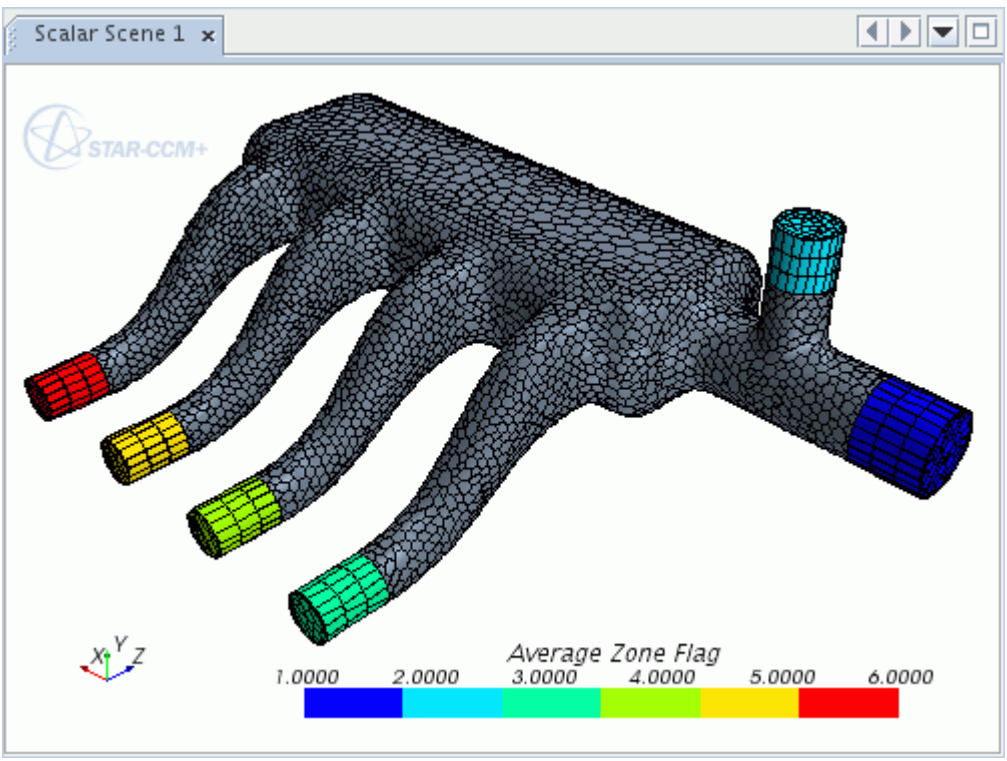
5. Create a scalar scene.
6. Edit the **Scalar Scene 1** > **Displayers** node and set the following properties:

Node	Property	Setting
Outline 1	*Surface*	**Activated**
	Mesh	**Activated**
Scalar 1		
└ **Parts**	*Parts*	**Cyl1**
		Cyl2
		Cyl3
		Cyl4
		EGR
		Intake

Node	Property	Setting
		Wall
└ **Scalar Field**	*Function*	**Average Zone Flag**
	Min	1
	Max	6
└ **Color Bar**	*Levels*	6

7. Click (**Save-Restore-Select views**) and restore **View 1**.
8. Initialize the Solution.

   A scene similar to the one shown below appears.



   The scalar values, which relate to which boundary was created first in STAR-CCM+, are of no significance and simply highlight the individual zones.

9. Save the simulation.

Running the Co-Simulation

Run the co-simulation and monitor the iterative progress output.

1. In STAR-CCM+, click (**Run**).

The iterative progress can be monitored in the *Output* window. The mass flow and mass fraction monitor plots show the state of the simulation up to the current time-step. The velocity vector scene shows the flow in and out of the boundaries at the current time-step.

There is a slight delay before any iterations are executed, as the GT-SUITE solver runs for 20 cycles before sending data to STAR-CCM+.

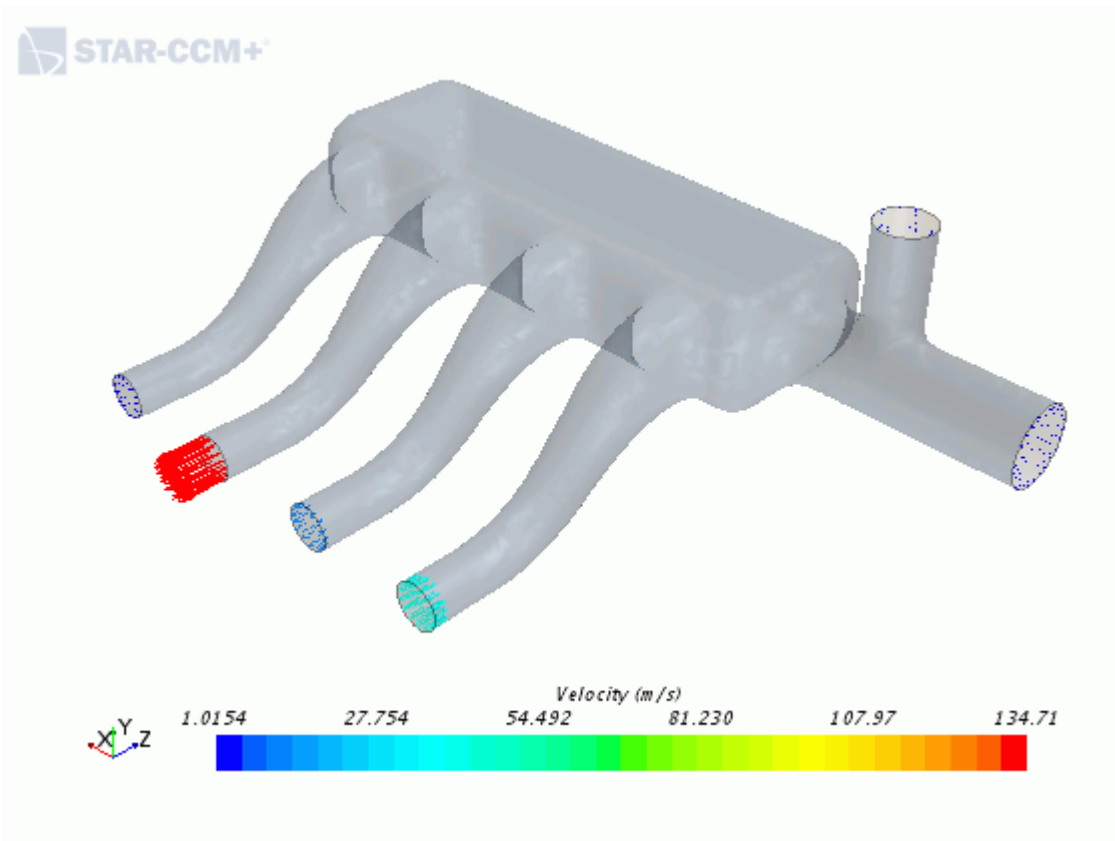
Note: In your own cases, activate the Autosave facility in STAR-CCM+ so you can easily perform a restart if the solver is unintentionally stopped or more cycles are required for convergence.

Visualizing the Results

Visualize the flow velocity field and the mass flow throughout the manifold.

The engine cycle in GT-SUITE starts when the Cylinder 1 Intake Valve closes, which the model defines as 110 degrees before Cylinder 1 Top Dead Center Firing (TDCF). Therefore, the results in STAR-CCM+ display the engine cycle from 110 degrees before Cylinder 1 TDCF, over a further 2160 degrees of crank rotation (three cycles).

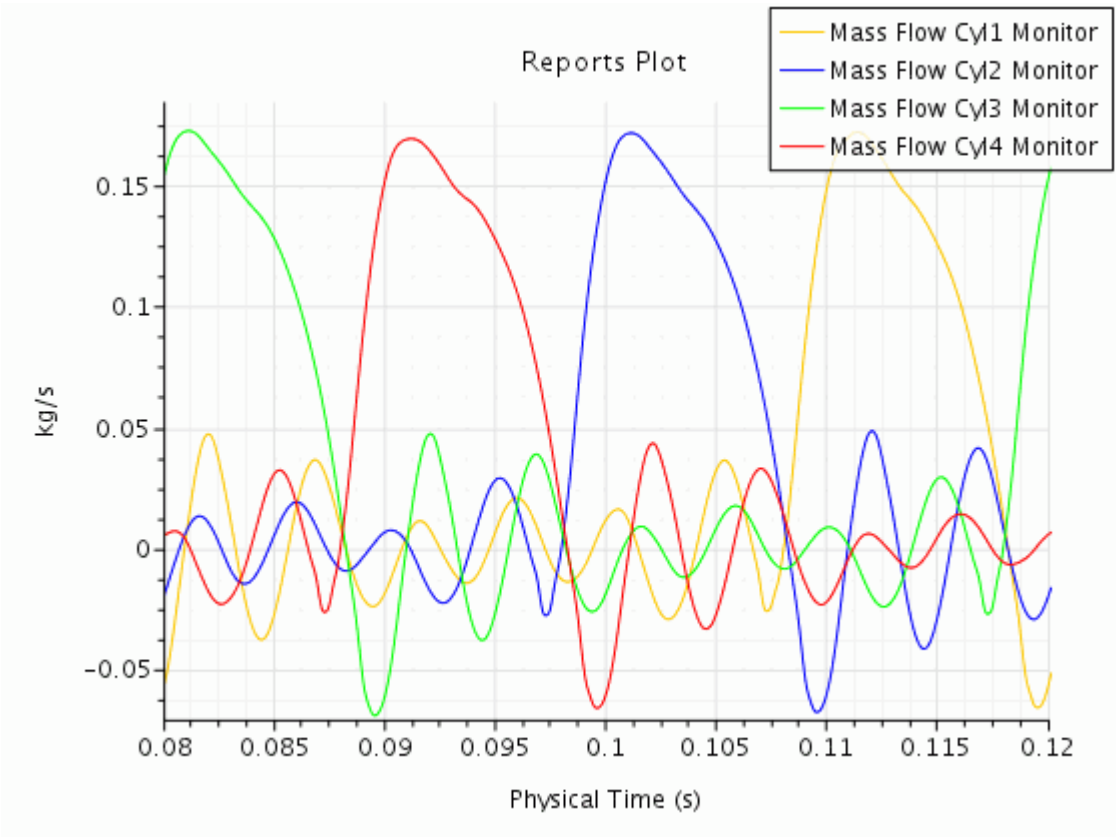
A velocity vector scene similar to the one shown below is displayed, which shows velocity vectors during the last time-step. It shows that GT-SUITE has opened the valve on Cylinder 3, with high velocity air flow.



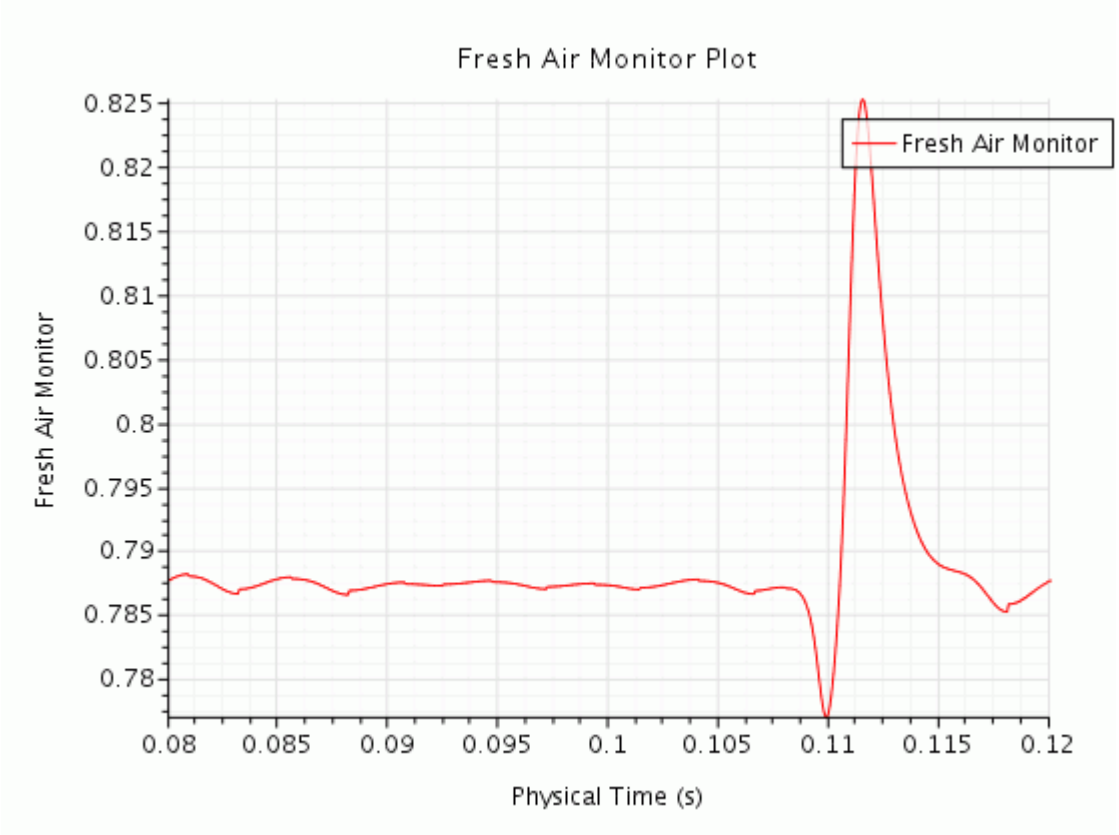
To view the mass flow plot over the last full engine cycle:

1. Right-click the **Plots** > **Reports Plot** and select **Open**.

2. Click **Scene/Plot**.
3. Select the **Axes** > **Bottom Axis** node and set *Minimum* to `0.08` and *Maximum* to `0.12`.



4. Similarly, view the fresh air mass fraction monitor over the last full engine cycle.



Summary

This tutorial has introduced the following STAR-CCM+ features:

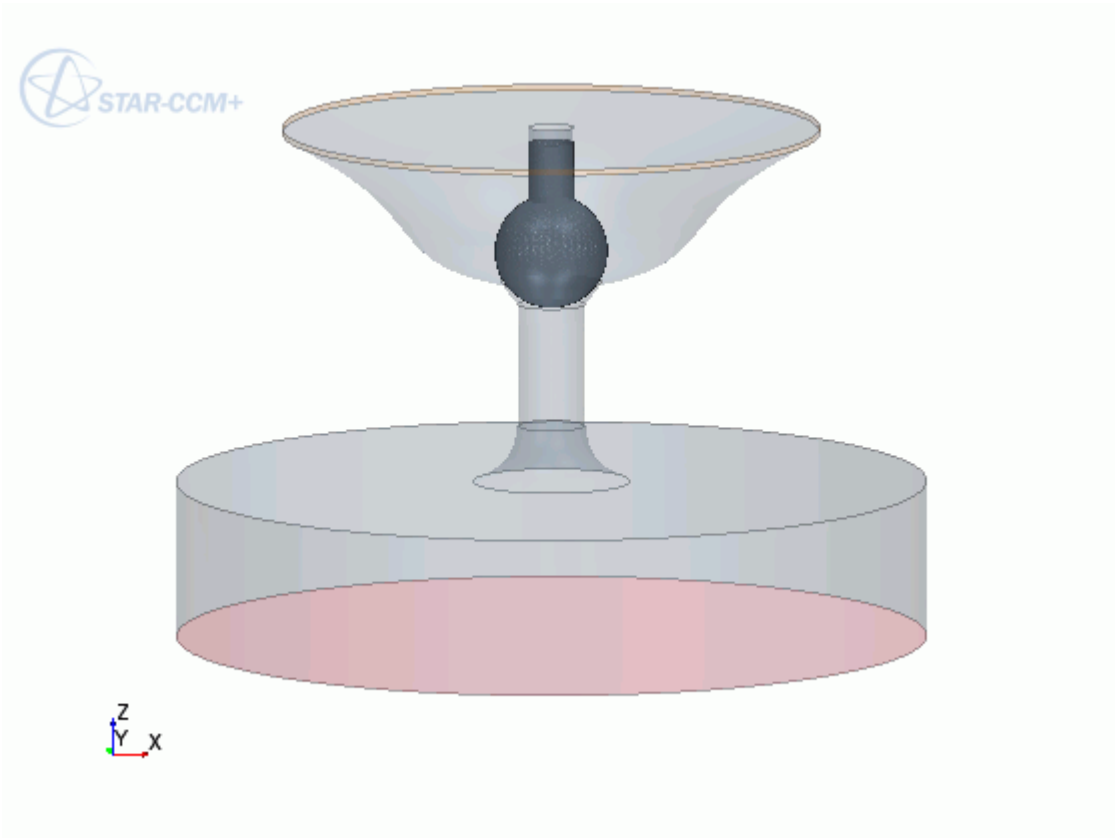
- Preparing the GT-SUITE model for coupling to STAR-CCM+.
- Obtaining initial conditions from an independent run of GT-SUITE.
- Setting up the STAR-CCM+ simulation for coupling:
  - Selecting physics models for coupling to GT-SUITE
  - Importing GT-SUITE material properties
  - Setting up the species representation
  - Defining coupled boundaries
  - Setting solver and stopping criteria
- Coupling STAR-CCM+ to GT-SUITE
- Visualizing the solution

Co-Simulation API: Spindle Valve

This tutorial demonstrates how to use the STAR-CCM+ Co-Simulation API (Application Programming Interface), in order to couple a partner program to a STAR-CCM+ simulation. The partner simulation models the behavior of a spindle ball valve in response to a multiphase flow.

To exchange data with a running STAR-CCM+ simulation, the partner code makes calls on the Co-Simulation API library that creates its own companion STAR-CCM+ simulation. Internally, the partner code exchanges data with its companion STAR-CCM+ API simulation that in turn exchanges data with the fluid simulation. The coupling between the companion STAR-CCM+ API simulation and the STAR-CCM+ fluid simulation behaves in the same way as a STAR-CCM+ to STAR-CCM+ co-simulation.

In this tutorial, you model the flow of water and gas through the opening between the ball and seat of a spindle valve. A numerical code is provided that models the position of the ball in response to the fluid forces. The fluid simulation uses the position data from the ball valve code to move the ball relative to the valve surface.



During the co-simulation, the ball valve code receives traction (pressure plus shear wall stress) from STAR-CCM+ and returns the ball displacement. On receiving the ball valve displacements, STAR-CCM+ applies the mesh morpher to move the surface of the ball valve in response.

Contents:

Prerequisites
The STAR-CCM+ Fluid Simulation
The Ball Valve Partner Code Simulation
Running the Co-Simulation
Visualizing the Results
Summary

Prerequisites

The instructions in the Co-Simulation API: Spindle Ball Valve tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Techniques	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+
STAR-CCM+ to STAR-CCM+ Co-Simulation	STAR-CCM+ to STAR-CCM+ Coupling: Heat Transfer in a Chimney

The STAR-CCM+ Fluid Simulation

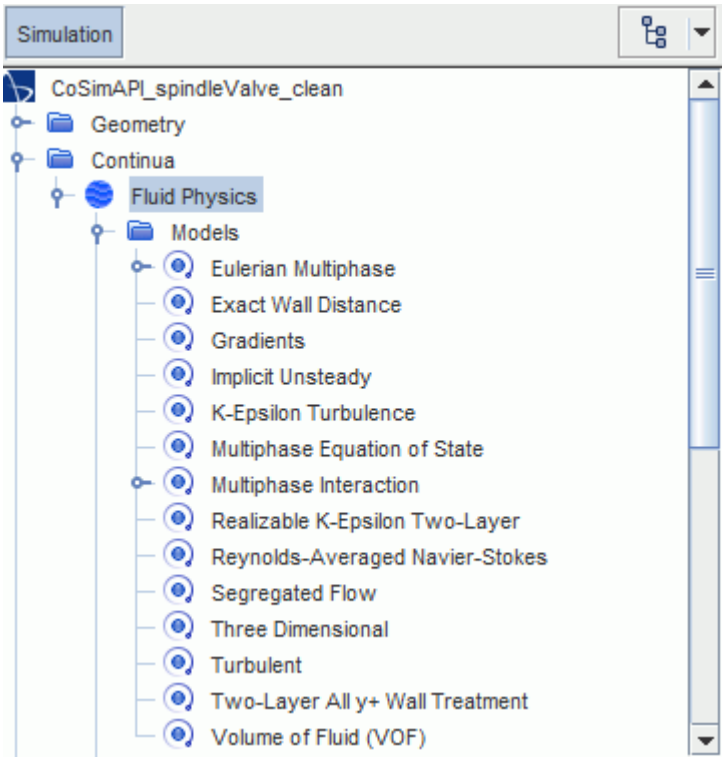
Prepare the STAR-CCM+ simulation that models the fluid flow in the spindle ball valve.

A pressure relief valve is designed to control the pressure in a system by relieving the pressurised fluid when the pressure exceeds preset limits. When the system pressure exceeds the design pressure limit, the valve is forced open and part of the fluid flows out of the system through an auxiliary outlet, lowering the system pressure.

In this simulation, STAR-CCM+ solves for the traction loads (pressure + shear wall stress) exerted by the fluid on the solid ball valve.

The turbulent multiphase flow of water and gas enters the valve inlet at a constant velocity of 0.2 m/s in the valve axial direction z.

The physics models and conditions for the fluid simulation are provided in the initial simulation file.



Contents:

Loading the Initial Simulation
Generating the Volume Mesh
Activating the Morpher
Specifying Co-Simulation Settings
Creating the Input Mesh File for the Partner Code
Monitoring the Solution

Loading the Initial Simulation

For this tutorial, you are provided with a simulation file that contains pre-defined objects.

Note: You are recommended to run this tutorial in parallel, using at least 4 cores.

To load the initial simulation:

1. Launch STAR-CCM+.
2. Select **File** > **Load Simulation...**
3. In the *Load Simulation* dialog, click **Browse...**
4. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/couplingWithCaeCodes/data`.
5. Select `CoSimApi_spindleValve.sim` then click **Open**.
6. In the *Load Simulation* dialog:

a) Within *Process Options*, select **Parallel on Local Host** and set the number of *Compute Processes* to at least 4.

b) Click **OK**.

7. Save the simulation as `spindleValve_fluid.sim`.

Generating the Volume Mesh

You are ready to generate the volume mesh for the fluid. All the mesh settings, including the meshing models and reference values, are provided in the initial simulation.

To generate the mesh:

1. Click (**Generate Volume Mesh**).
2. When the mesh generation is complete, save the simulation.
3. Visualize the volume mesh with a method of your choice, such as creating a mesh scene.

The mesh appears as shown below.

Activating the Morpher

Activate the mesh morpher to deform the fluid mesh according to the spindle ball displacements.

STAR-CCM+ exports traction loads (pressure + wall shear stress) to the partner code, and the partner code solves for the solid surface displacements. In STAR-CCM+, the displacements are used as inputs to the mesh morpher.

To activate the mesh morpher:

1. Expand the **Tools** node.
2. Right-click the **Motions** node and select **New** > **Morphing**.

The default morpher settings are appropriate to this case. Assign the morphing motion to the fluid region:

3. Expand the **Regions** > **Spindle Valve Fluid** > **Physics Values** node.
4. Select the **Motion Specification** node and set *Motion* to **Morphing**.

Select the appropriate morpher method for the relevant boundaries:

5. Expand the **Spindle Valve Fluid** > **Boundaries** > **slidingwall** > **Physics Conditions** node.
6. Select the **Morpher Specification** node and set *Specification* to **Constraint**.
7. Select the **Morpher Constraint Specification** node and set *Constraint* to **Initial Boundary**.

   This setting makes sure that the ball only moves in the direction of the spindle axis.

   When you specify the spindle ball boundary as the coupled boundary, STAR-CCM+ automatically sets the morpher method for this boundary, so that the morpher updates the mesh based on the displacements imported from the partner simulation.

As this co-simulation uses the implicit coupling scheme, instruct the morpher to update the fluid mesh at every inner iteration:

8. Select the **Solvers** > **Mesh Morpher** node and activate *Morph At Inner Iterations*.

Specifying Co-Simulation Settings

Prepare the fluid simulation for co-simulation. Setting up a co-simulation requires you to activate a co-simulation model and to specify the coupled boundaries and the fields exchanged with the partner code.

The fluid simulation couples with the STAR-CCM+ API server associated with the partner code. As the STAR-CCM+ API server has all the features of a STAR-CCM+ simulation, you specify the co-simulation settings in the fluid simulation as you would do for a standard STAR-CCM+ to STAR-CCM+ co-simulation.

To enable co-simulation, activate a STAR-CCM+ Co-Simulation model:

1. For the physics continuum, **Continua** > **Fluid Physics**, select the following models in order:

Group box	Model
Optional Models	**Co-Simulation**
Co-Simulation Models	**STAR-CCM+**
STAR-CCM+ Coupling Models	**STAR-CCM+ Implicit Coupling**

2. Click **Close**.

   STAR-CCM+ adds additional physics continua to represent the partner simulation.

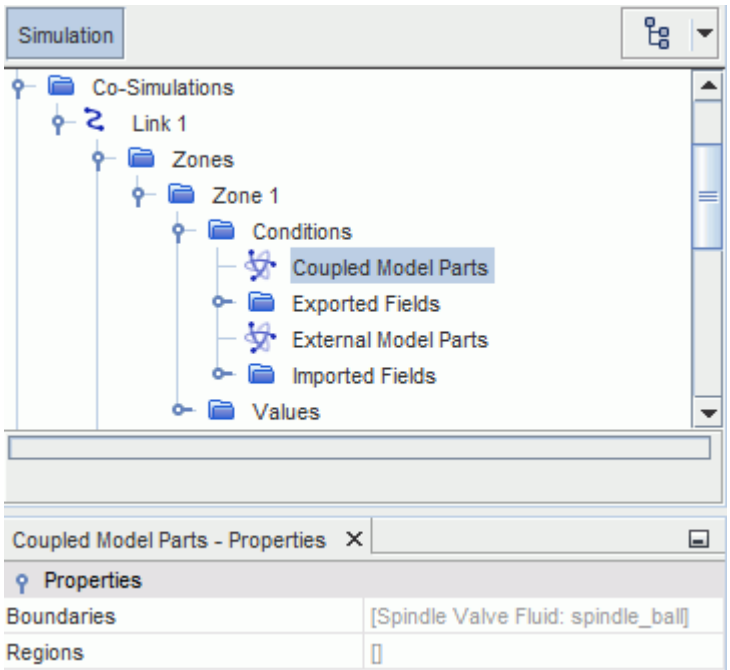
When you activate a co-simulation model, STAR-CCM+ adds a **Co-Simulations** node to the simulation tree. This node allows you to specify global co-simulation settings, such as the *Concurrency Mode*, as well as to define and set up the co-simulation coupling zones.

In this simulation you set up a single coupling zone, as you have a single solid-fluid interface across which the two simulations exchange data. By default, the STAR-CCM+ server is the leading simulation.

To specify the external surface of the spindle ball as the coupled boundary:

3. Expand the **Regions** > **Spindle Valve Fluid** > **Boundaries** > **spindle_ball** > **Physics Conditions** node.
4. Select the **External Code Coupling Specification** node and set *External Code Coupling Specification* to **Link 1: Zone 1**.

   Once you associate a boundary to a coupling zone, STAR-CCM+ updates the **Coupled Model Parts** condition for that zone.



When you activate the morpher with a STAR-CCM+ Co-Simulation model, STAR-CCM+ automatically sets up the simulation for mechanical co-simulation. By default, STAR-CCM+ sets the fluid simulation to import displacements and to export pressure data. To export traction (pressure and wall shear stress):

5. Expand the **Co-Simulations** > **Link 1** > **Zones** > **Zone 1** > **Conditions** node.
6. Right-click the **Exported Fields** > **Pressure** node and select **Delete**.
7. Right-click the **Exported Fields** node and select **Add Mechanical Field** > **Traction**.

Specify the number of inner iterations between exchanges:

8. Select the **Link 1** > **Conditions** > **Field Exchange Controls** node and set *Iterations Per Exchange* to `5`.

Creating the Input Mesh File for the Partner Code

Extract the spindle ball boundary surface to use as the input surface mesh in the partner code.

In this tutorial, the partner numerical code that solves for the ball position reads input mesh files with .stl coded format. You create this surface mesh by exporting the spindle ball boundary from the fluid simulation.

1. In your working directory, create a new folder named `CoSimAPI_Mesh`.
2. In your simulation, right-click the **Representations** > **Volume Mesh** node and select **Extract Boundary Surface...**
3. In the *Extract Boundary Surface* dialog, click **OK**.
4. Expand the **Representations** > **Extracted Surface** > **Regions** > **Spindle Valve Fluid** > **Boundaries** node.
5. Right-click the **spindle-ball** node and select **Export Boundaries...**
6. In the *Save* dialog:
   a) Navigate to `CoSimAPI_Mesh`.
   b) Set *Files of Type* to **Stereolithography File (*.stl)**.
   c) Click **Save**.
7. In the *STL Export Options* dialog, set *File Type* to **Coded** and click **OK**.
8. Save the simulation.

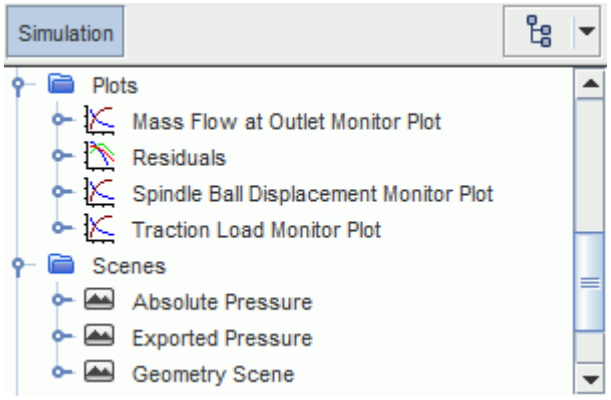
Monitoring the Solution

Prepare scenes and plots to monitor the solution progress while the co-simulation is running.

In the fluid simulation, you are provided with scenes and plots of some of the quantities that are relevant to this analysis. You can use these tools to monitor the solution progress during the co-simulation with the partner code.

Complete the set up of the plots and scenes that are provided in the simulation:

1. Expand the **Reports** node.
2. Select the **Spindle Ball Displacement** node and set *Field Function* to **Co-Simulations** > **Co-Simulation: Imported Displacement** > **Laboratory** > **k**.
3. Right-click the **Spindle Ball Displacement** report and select **Create Monitor and Plot from Report**.



4. Multi-select all the reports and set *Representation* to **Volume Mesh**.
5. Expand the **Scenes** > **Exported Pressure** > **Displayers** > **Scalar 1** node.

6. Select **Scalar Field** and set *Function* to **Co-Simulations** > **Co-Simulation: Exported Pressure**.
7. Multi-select the **Absolute Pressure** and **Exported Pressure** scenes. Right-click one of the selected scenes and select **Edit**.
8. In the *Multiple Objects* dialog, expand the **Displayers** > **Scalar 1** node and set *Representation* to **Volume Mesh**.
9. Save the simulation.

The Ball Valve Partner Code Simulation

In this tutorial, you are provided with a sample partner code that solves the equation of motion for the ball valve. Compile the code and launch the ball valve solid simulation.

A spring connected to the ball valve allows the valve motion in response to the fluid forces. The ball valve is constrained to move along its axial direction, that in this tutorial corresponds to the z direction. The partner code calculates the position of the valve treating the model as a 1-DoF spring-mass-damping system, with equation of motion:

$$m\ddot{z}(t) + c\dot{z}(t) + kz(t) = F_z(t) \tag{4215}$$

where m is the ball valve mass, c is the damping coefficient, k is the spring stiffness, and F_z is the z component of the traction field imported from the STAR-CCM+ fluid simulation, integrated over surface. In this tutorial damping effects are neglected, so $c = 0$. The equation of motion, Eqn. (4215), is solved using the Newmark-beta numerical integration method.

The code that is provided in this tutorial consists of several .cpp and .h files. The main file, `SpindleValve.cpp`, outlines the recommended workflow when using the Co-Simulation API, from loading the Co-Simulation API library to setting up the co-simulation loop. You can modify the ball valve simulation time-step and maximum physical time values in this file. You can modify the spring-mass-damping system parameters in the file, `SpringMassSolver.cpp`.

Compiling the code generates an executable file, that you use to launch the ball valve solid simulation.

Contents:

Compiling the Partner Code
Launching the Ball Valve Simulation

Compiling the Partner Code

Compile the sample partner code using the compiler helper script that is included in the STAR-CCM+ installation.

Start by copying the sample partner code to your working directory:

1. Navigate to `[INSTALL_DIR]/doc/startutorialsdata/couplingWithCaeCodes/data`.
2. Copy the entire `CoSimAPI_Code` folder to your working directory.

   The files that are contained in this folder make up the partner code that solves for the ball valve motion.

Based on the platform that you are using, follow the instructions for running the `buildapi` helper script for Linux or the `buildapi.bat` helper script for Windows. On Linux, you require g++ version 4.3.4 or greater.

Linux

To use the `buildapi` script to compile the sample partner code:

1. Open a terminal.
2. Navigate to the `CoSimAPI_Code` folder that contains the sample code files:

   ```
   % cd [PATH]/CoSimAPI_Code
   ```

   where `[PATH]` is the path to your working directory.
3. Enter the following command:

   ```
   % [INSTALLDIR]/star/bin/buildapi spindleValve_solid
   ```

   where `[INSTALLDIR]` is the path to your STAR-CCM+ installation directory.

   This command compiles all the `.cpp` files that are located in the `CoSimAPI_Code` folder and creates an executable output file, `spindleValve_solid`, that simulates the ball valve motion. Unless you specify a different file path in the command, the `spindleValve_solid` executable is saved in the `CoSimAPI_Code` folder.
4. Close the terminal.

Windows

To use the `buildapi.bat` script to compile the sample partner code:

1. Open a *Command Prompt*.
2. Navigate to the `CoSimAPI_Code` folder that contains the sample code files:

   ```
   > cd [PATH]\CoSimAPI_Code
   ```

   where `[PATH]` is the path to your working directory.
3. Call your Intel C++ Compiler by entering the following command:

   ```
   > call "[INSTALLDIR]\bin\compilervars.bat" intel64
   ```

   where `[INSTALLDIR]` is the path to the compiler installation directory.
4. To run the `buildapi.bat` helper script, enter the following command:

   ```
   > [INSTALLDIR]\star\bin\buildapi.bat spindleValve_solid
   ```

   where `[INSTALLDIR]` is the path to your STAR-CCM+ installation directory.

   This command compiles all the `.cpp` files that are located in the `CoSimAPI_Code` folder and creates an executable output file, `spindleValve_solid.exe`, that simulates the ball valve motion. Unless you specify a different file path in the command, the `spindleValve_solid.exe` executable is saved in the `CoSimAPI_Code` folder.
5. Close the *Command Prompt*.

Launching the Ball Valve Simulation

Launch the ball valve simulation using the `starapi` helper script for Linux or the `starapi.bat` helper script for Windows. These scripts are included in the STAR-CCM+ installation.

The `starapi` helper script launches the partner program, and allows the partner code to load the Co-Simulation API library and spawn the companion STAR-CCM+ API Server.

You can run the script from a command console or from the STAR-CCM+ fluid simulation. In this tutorial you run the script from a command console. This method requires two STAR-CCM+ licenses, one for the STAR-CCM+ fluid simulation and one for the STAR-CCM+ API Server. However, if you do not wish to monitor the output from the ball valve simulation, you can use the second method for sharing a single STAR-CCM+ license. For more information, see Running a Co-Simulation between Partner Code and STAR-CCM+.

Based on the platform that you are using, follow the instructions for running the `starapi` script for Linux or the `starapi.bat` script for Windows.

Linux

To launch the partner program:

1. Open a terminal.
2. Navigate to your working directory:

   ```
   % cd [PATH]/[WORKDIR_NAME]
   ```

   where `[PATH]` is the path to your working directory and `[WORKDIR_NAME]` is the name of your working directory.
3. Enter the following command, specifying the relative path to the `spindleValve_solid` executable and to the input mesh file that you created in the previous sections of this tutorial:

   ```
   % [INSTALLDIR]/star/bin/starapi [FILEPATH]/spindleValve_solid
   [FILEPATH]/spindle_ball.stl
   ```

   where `[INSTALLDIR]` is the path to your STAR-CCM+ installation directory.

   For example, if you saved all the files as instructed in this tutorial, the command complete with the path to the files is:

   ```
   % [INSTALLDIR]/star/bin/starapi CoSimAPI_Code/spindleValve_solid
   CoSimAPI_Mesh/spindle_ball.stl
   ```

Information on the launching process is displayed in the terminal.

The message `spindleValve_solid: Waiting for incoming fields...` tells you that the STAR-CCM+ API Server has been launched and the partner code, `spindleValve_solid`, is ready to import fields from the STAR-CCM+ `spindleValve_fluid` simulation.

Note: Leave the terminal open. Closing the terminal terminates the partner program process.

Windows

To launch the partner program:

1. Open a *Command Prompt*.
2. Navigate to your working directory:

   ```
   > cd [PATH]\[WORKDIR_NAME]
   ```

   where `[PATH]` is the path to your working directory and `[WORKDIR_NAME]` is the name of your working directory.
3. Enter the following command, specifying the relative path to the `spindleValve_solid.exe` executable and to the input mesh file that you created in the previous sections of this tutorial:

   ```
   > [INSTALLDIR]\star\bin\starapi.bat [FILEPATH]
   \spindleValve_solid.exe [FILEPATH]\spindle_ball.stl
   ```

   where `[INSTALLDIR]` is the path to your STAR-CCM+ installation directory.

   For example, if you saved all the files as instructed in this tutorial, the command complete with the path to the files is:

   ```
   > [INSTALLDIR]\star\bin\starapi.bat CoSimAPI_Code
   \spindleValve_solid.exe CoSimAPI_Mesh\spindle_ball.stl
   ```

Information on the launching process is displayed in the *Command Prompt*.

The message `spindleValve_solid: Waiting for incoming fields...` tells you that the STAR-CCM+ API Server has been launched and the partner code, `spindleValve_solid`, is ready to import fields from the STAR-CCM+ `spindleValve_fluid` simulation.

Note: Leave the *Command Prompt* open. Closing the *Command Prompt* terminates the partner program process.

Running the Co-Simulation

Couple the STAR-CCM+ fluid simulation to the ball valve simulation and run the co-simulation.

1. Select the **Co-Simulations** > **Link 1** > **Values** > **STAR-CCM+ Partner Host:Port** node.
2. In the *Properties* window, click ... (**Custom Editor**).
3. In the *STAR-CCM+ Partner Host:Port* dialog, select the STAR-CCM+ server named `spindleValve_solid`, then click **OK**.

   The fluid simulation is now connected to the partner code through the STAR-CCM+ API Server.
4. In the fluid simulation, click (**Run**).

   Information on the co-simulation process is displayed in both the fluid simulation *Output* window and the terminal where you launched the partner code.

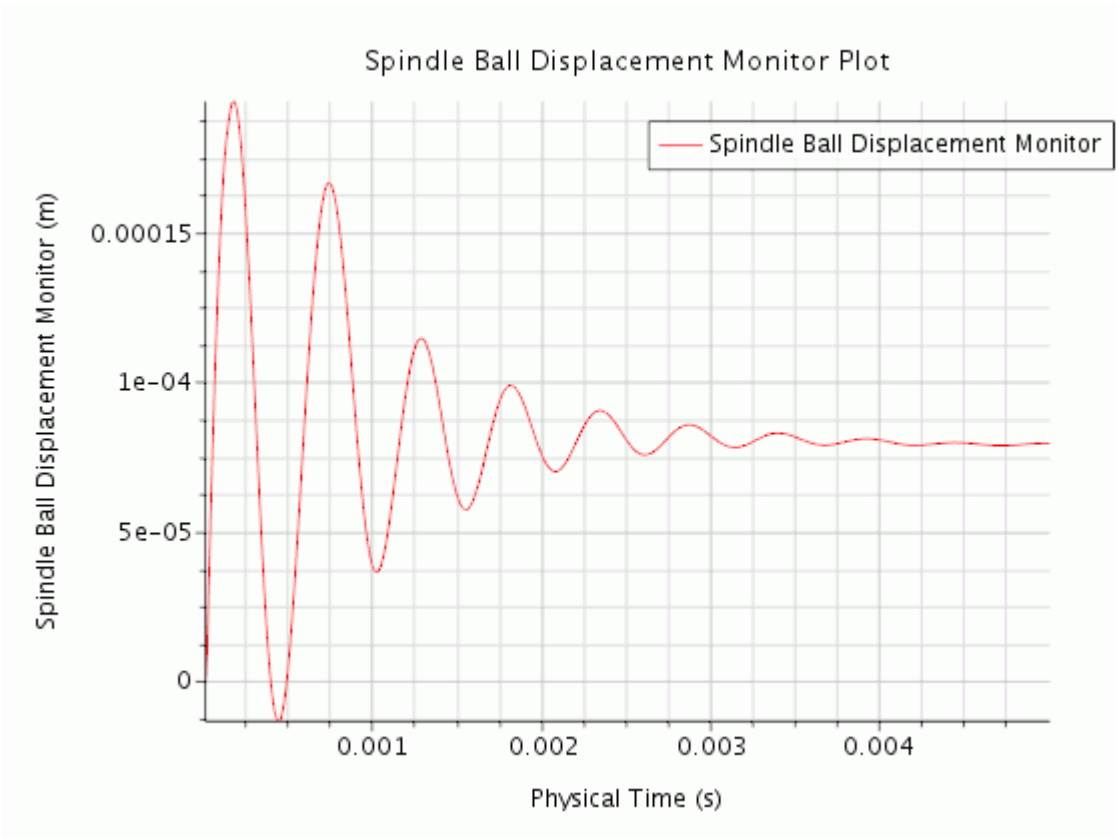
To monitor the solution progress during the co-simulation:

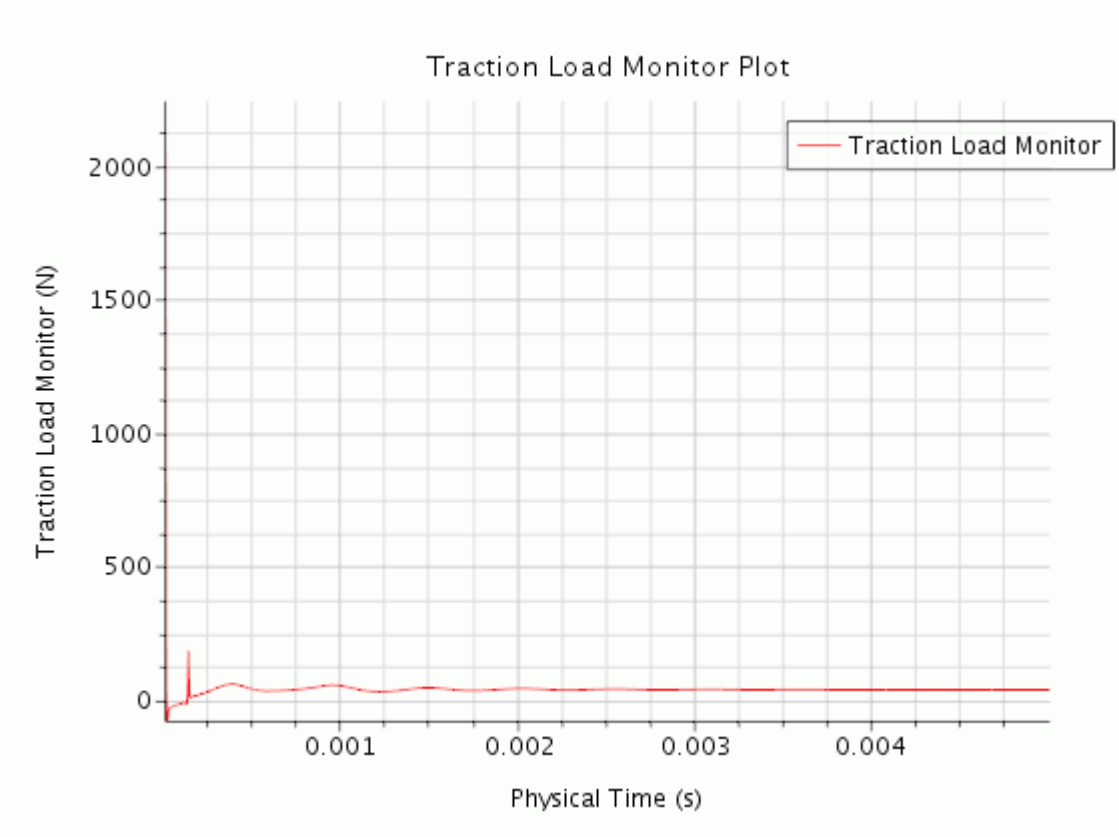
5. Right-click the **Plots** node and select **Open All Plots**.

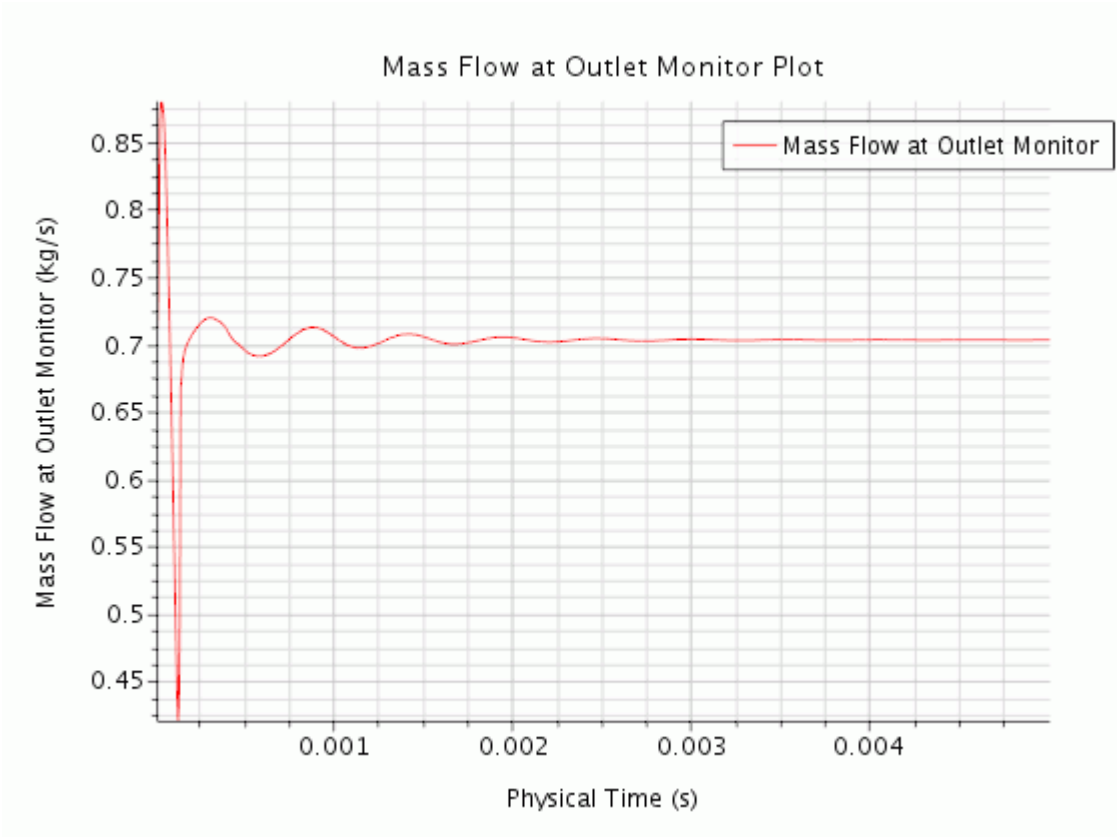
Visualizing the Results

Visualize the scenes and plots that are relevant to this mechanical co-simulation analysis.

When the co-simulation is complete, the plots for the spindle ball displacement, the traction loads, and the mass flow at the outlet as functions of time appear as shown below:



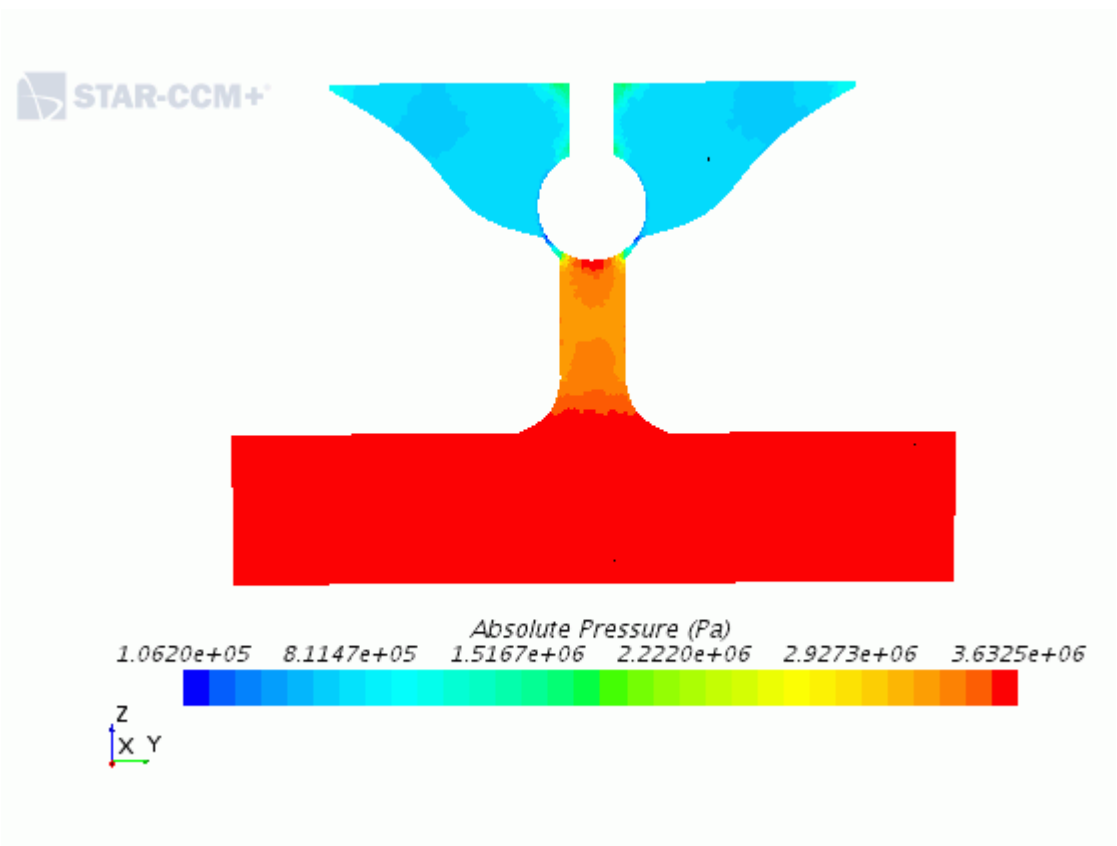

Traction Load Monitor Plot
Traction Load Monitor
Traction Load Monitor (N)
2000
1500
1000
500
0
0.001
0.002
0.003
0.004
Physical Time (s)


Mass Flow at Outlet Monitor Plot
Mass Flow at Outlet Monitor
Mass Flow at Outlet Monitor (kg/s)
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.001
0.002
0.003
0.004
Physical Time (s)

Visualize the scalar scenes displaying the pressure in the fluid domain and the pressure exported to the ball valve surface.

1. Open the **Absolute Pressure** and **Exported Pressure** scenes.

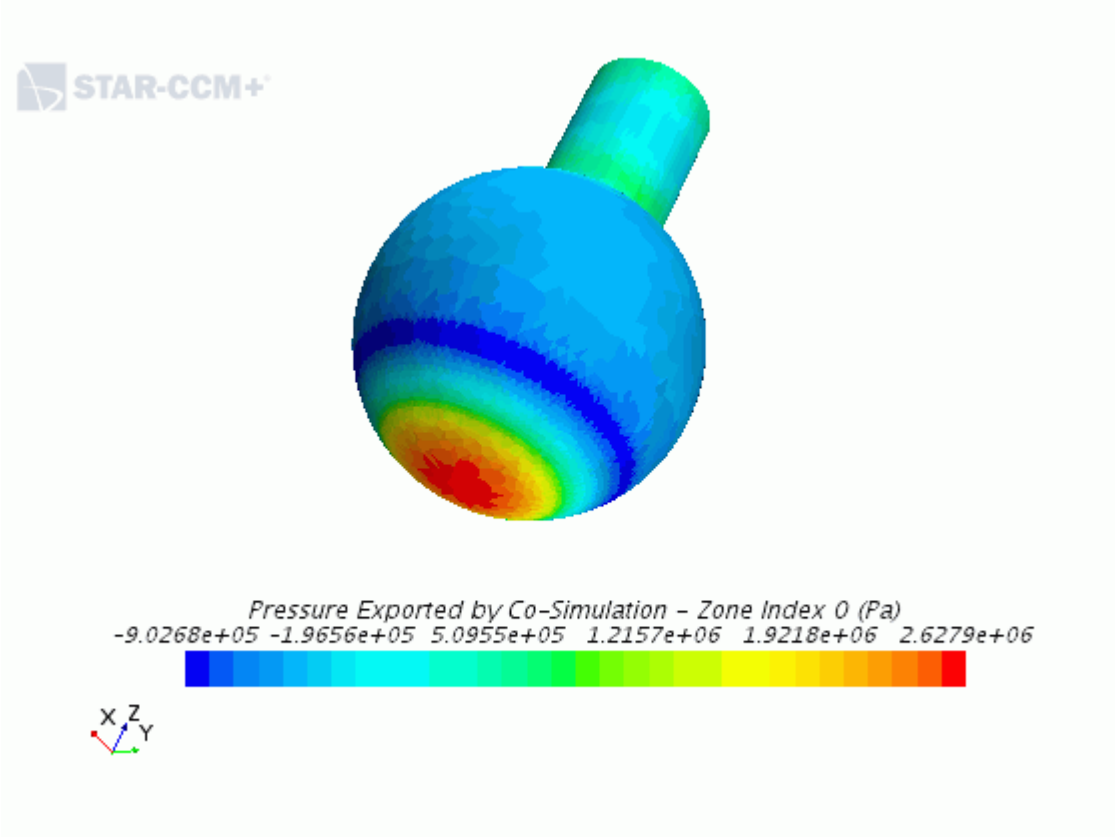
   The **Absolute Pressure** scene appears as shown below.



Visualize the exported pressure data on the ball valve surface, that is imported under the **Regions** node when you run the co-simulation.

2. Edit the **Exported Pressure** > **Displayers** > **Scalar 1** > **Parts** node.
3. In the *Parts* dialog, remove the selected boundary, select the **Spindle Ball Surface** > **Boundaries** > **Spindle Ball Surface** node then click **OK**.

   The **Exported Pressure** scene appears as shown below.



4. Save the simulation.

Summary

This tutorial demonstrated how to use the STAR-CCM+ Co-Simulation API to couple partner code with a running STAR-CCM+ simulation.

This tutorial has demonstrated the following features:

- Activating the morpher for a mechanical co-simulation analysis.
- Specifying co-simulation settings for coupling with partner code.
- Creating the input mesh for the partner code.
- Compiling the partner code using the `buildapi` helper script.
- Launching the STAR-CCM+ API Server using the `starapi` helper script.
- Coupling a STAR-CCM+ simulation with the partner code.
- Running the co-simulation.

Analysis Methods

The tutorials in this set illustrate various STAR-CCM+ features for analysing and visualizing simulation data.

Contents:

Volume Rendering: Steckler Room
Data Focus: Thermal Comfort
Simulation History: Lagrangian DEM on an Auger

Volume Rendering: Steckler Room

This tutorial demonstrates the advanced post-processing capabilities of STAR-CCM+ using the simulation file that was created in the tutorial, Fire and Smoke Wizard: Steckler Room. If your graphics card does not support volume rendering, activate advanced rendering for the relevant scene.

The procedure includes adjusting scalar values, creating a resampled volume derived part, setting up a custom colormap, adjusting lighting, rerunning the transient solution, and capturing transient results using the solution recording module.

Contents:

Prerequisites
Loading an Existing Simulation
Adjusting the Scalar
Setting Up the Resampled Volume
Setting Up a Colormap
Viewing the Volume-Rendered Soot Density
Modifying the Colormap Settings
Adjusting the Lighting
Setting Solver Parameters and Stopping Criteria
Setting Up the Solution History File
Visualizing the Solution
Running the Simulation
Visualizing the Results
Creating a Recorded Solution View
Creating an Animation from the Solution View
Summary

Prerequisites

The instructions in the Volume Rendering tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Loading an Existing Simulation

For this tutorial, use an existing simulation file that contains objects and solution data from the tutorial, Fire and Smoke Wizard: Steckler Room.

To start this tutorial, you can either:

- use your own version of the simulation file from the Steckler Room tutorial, if you have already worked through it; or,
- use the version from the Tutorials Files package located on the Steve Support Portal. To obtain the package:
  1. Follow the instructions in the section, Downloading the Tutorial Files.
  2. After extracting the package to disk, copy the file, `tutorials/reactingFlow/FireAndSmokeWizardStecklerRoom_final.sim`, into the working directory.
  3. Rename the file to `roomfire.sim`.

To open the simulation file:

1. Launch STAR-CCM+.
2. Select **File** > **Load Simulation...**
3. In the *Load Simulation* dialog, click **Browse...**
4. In the *Open* dialog, navigate to the directory that contains `roomfire.sim`.
5. Select `roomfire.sim` then click **Open**.
6. In the *Load Simulation* dialog, click **OK**.
7. Save the simulation as `FireAndSmokeResampled.sim`.

Adjusting the Scalar

One application of volume rendering is for visualizing the distribution of scalar quantities that represent combustion products.

In the reference tutorial, the Soot Density function provides an estimate of the amount of soot that a room fire generates. The function is defined as:

```
${Passive Scalar}*$Density*1000*1000
```

Prepare the Soot Density function:

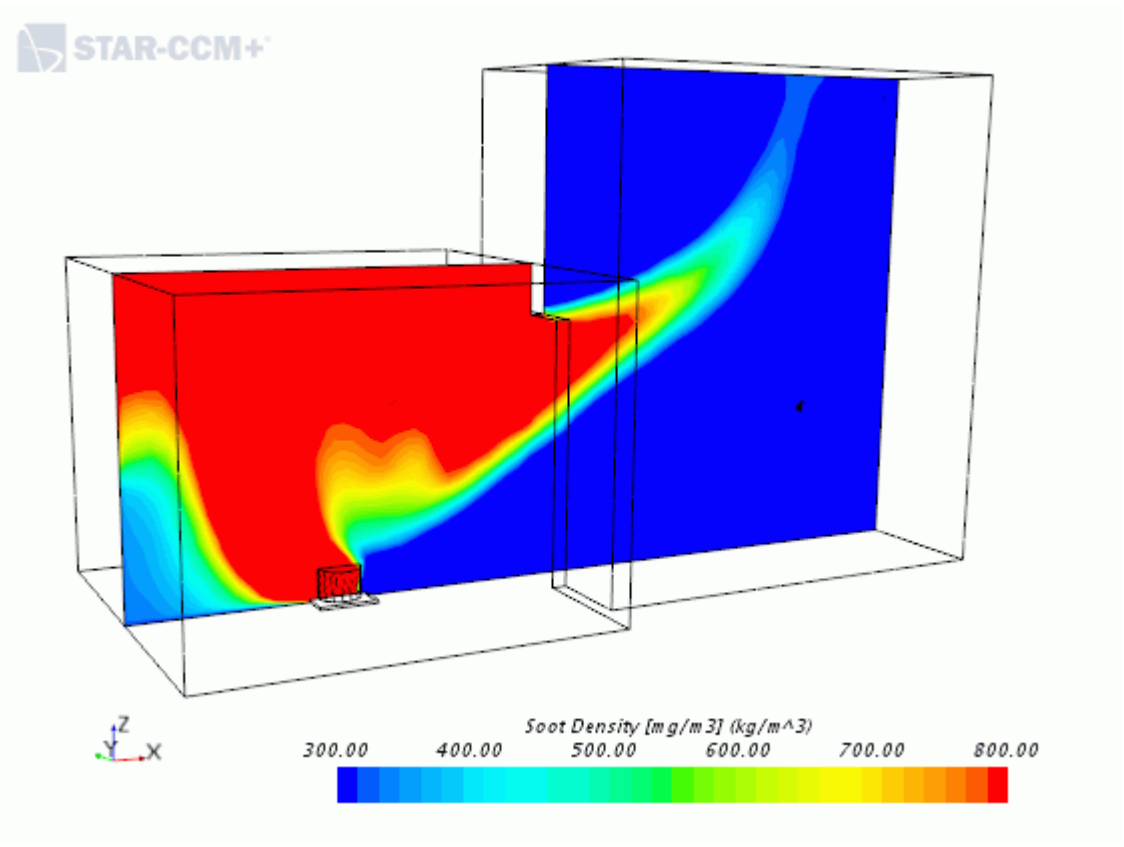
1. Select the **Tools** > **Field Functions** > **Soot Density** node.
2. Rename the node `Soot Density [mg/m3]`.
3. Set the *Function Name* property to `Soot Density [mg/m3]`.

To get a sense of the range of the scalar, use the scalar displayer.

4. Open **Scenes** > **Scalar Scene 1**.
5. Edit the **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Scalar Field** node and set the following properties:

Property	Setting
Function	**Soot Density [mg/m3]**
Clip	**Off**
Min	300.0
Max	800.0

The scalar scene now shows an approximation of the scalar variation in the volume.



Setting Up the Resampled Volume

To activate volume rendering, create a resampled volume derived part.

Volume resampling is a technique for rendering volume at a relatively low computational cost. A key element of this technique is the voxel, a pixel that represents volume in three-dimensional space.

In the dialog that you use for setting up a resampled volume derived part, the Voxel Count preview lets you decide how to balance quality and computing costs. A larger number of voxels produces better volume rendering, but takes more time. A smaller number saves time but reduces quality. The recommended starting voxel count is between 1 million and 10 million.

1. To create a resampled volume, right-click the **Derived Parts** node and select **New Part** > **Resampled Volume...**

2. In the dialog, set the following properties:

Property	Setting
Input Parts	**Room**, **Fire**
Cell Size to Voxel Ratio	`1.0`
Display	**New Scalar Displayer**

   Accept the other default settings.

3. To see what the resampled volume looks like, and to get the voxel count, click **Preview**.

   Beneath the **Preview** button, the voxel count appears: 3.5 million.

4. Click **Create**, then click **Close**.

   A new derived part named **Resampled Volume** is created in the **Derived Parts** manager node.

Setting Up a Colormap

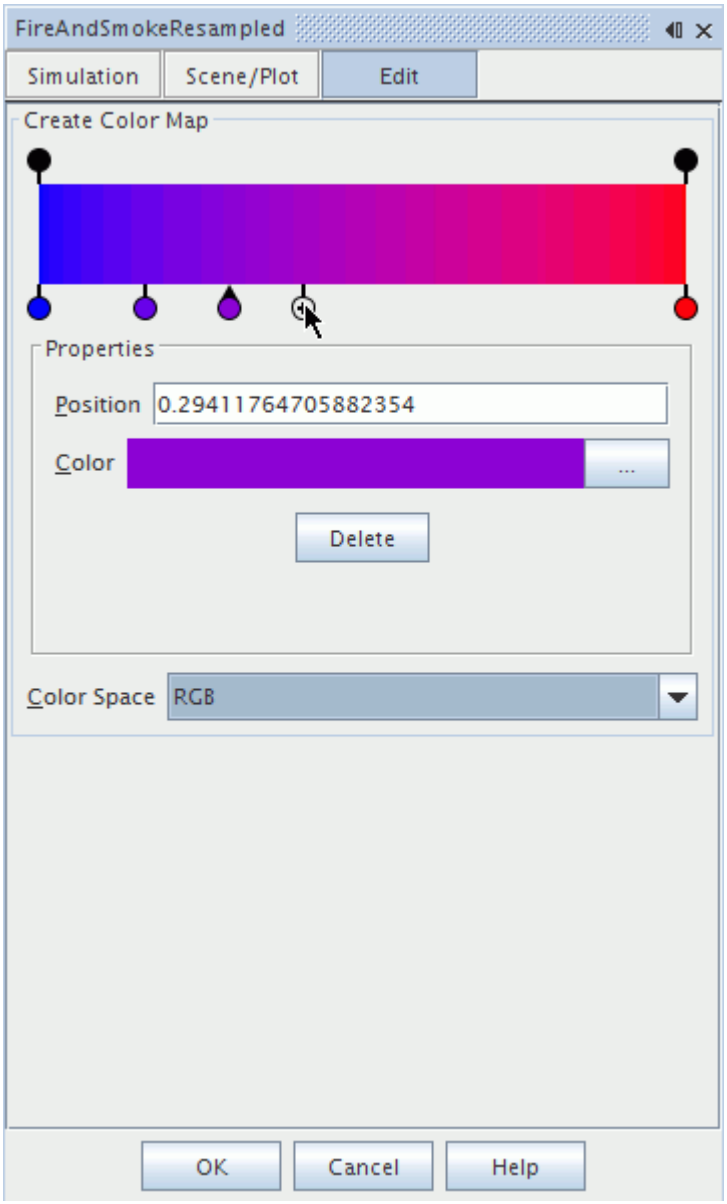
Prepare a custom colormap for the resampled volume.

The colormap maps scalar values to a set of colors for interpolating the results. This example includes a colormap to make volumes of higher soot density more obvious in two ways:

- Highlighting the scalars of interest by customizing the color choices -- colors that are isoluminant, or nearly so, make it easier to perceive change in the scalar.
- Using transparency to emphasize some changes in the scalar range

Color settings for control points are specified as RGB (Red, Green, Blue) in this example. RGB specification is the most common.

1. Expand the **Tools** node and right-click the **Color Maps** node. Select **New** from the pop-up menu.
2. In the *Create Color Map* dialog:
   a) Set **Color Space** to **RGB**.
   b) Add three color control points by clicking beneath the color spectrum for each one.



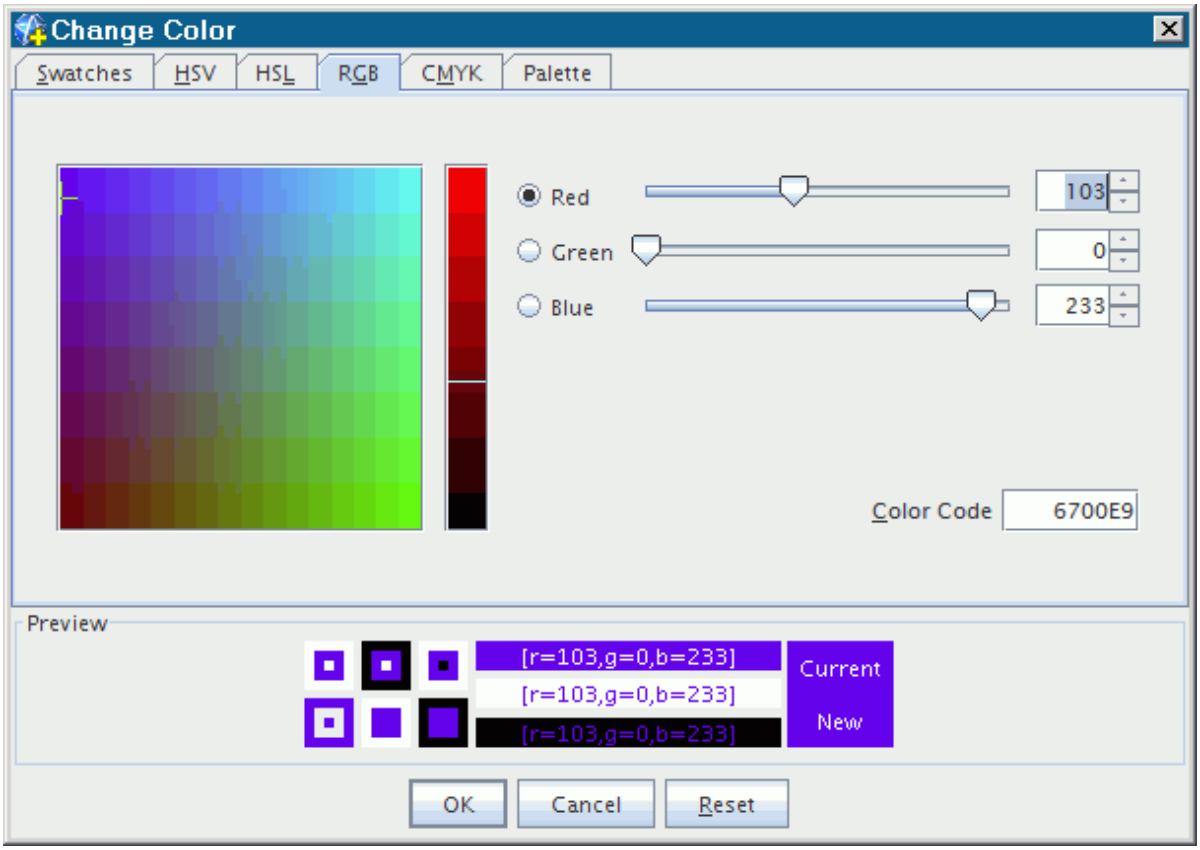
A total of five color control points are beneath the spectrum.

c) Select the new control points one at a time and set the *Position* property to `0.25`, `0.5`, and `0.75`, respectively.

You are ready to edit the color of each control point.

d) To edit a color control point, double-click it.

e) In the *Change Color* dialog, select the *RGB* tab and enter values for *Red*, *Green*, and *Blue* for the respective control point as listed in the table below. After you finish the color settings for each control point, click **OK**.

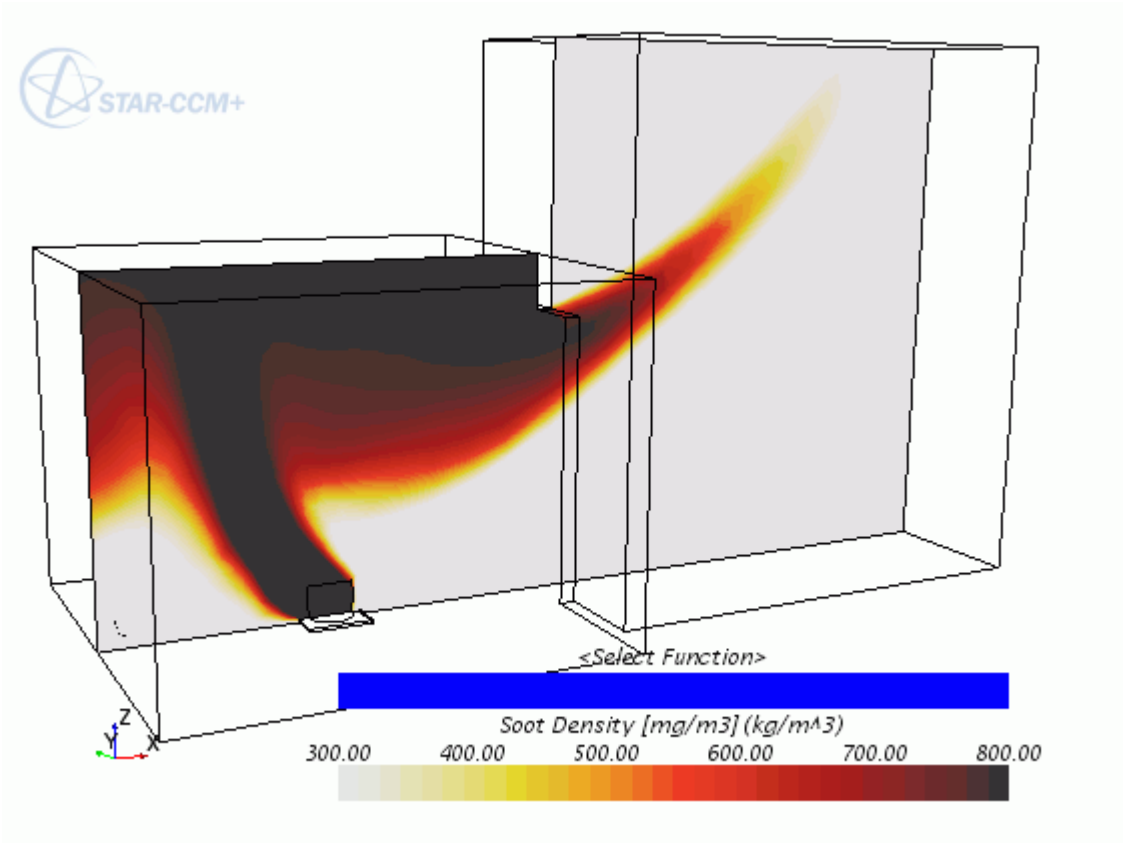


Position	Red	Green	Blue
Left Limit	231	231	231
0.25	229	216	41
0.5	236	61	37
0.75	167	26	25
Right Limit	51	51	51

f) Click **OK**.

3. Rename the new **user 1** colormap node, `soot_ramp`.
4. To preview the soot density colormap in the plane section, select the **Scalar Scene 1** > **Displayers** > **Scalar 1** > **Color Bar** node and set *Color Map* to **soot_ramp**.

   The soot density colors appear in the scalar scene.



5. Save the simulation.

Viewing the Volume-Rendered Soot Density

Use the resampled volume derived part for rendering the volume of the soot density.

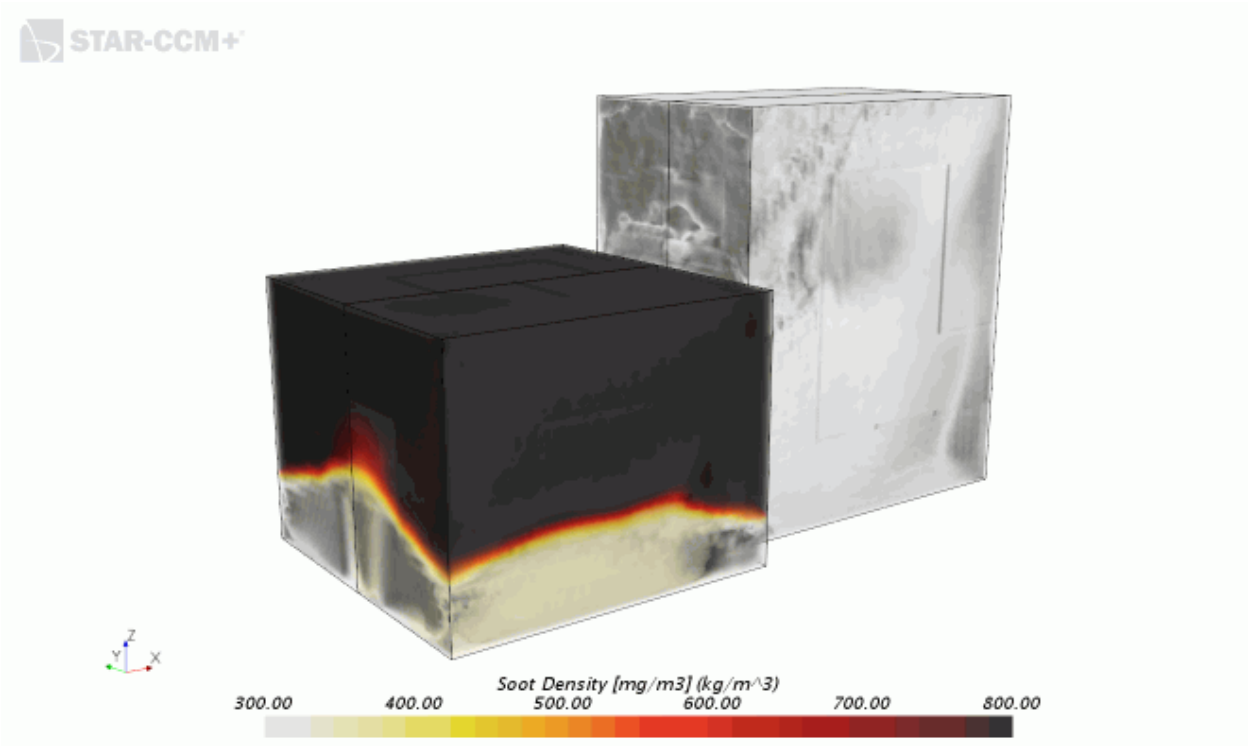
Switch scalar displayers to begin working with the resampled volume.

1. Right-click the **Scalar Scene 1** > **Displayers** > **Scalar 1** node and select **Toggle Visibility**.
2. Edit the **Displayers** > **Resampled Volume Scalar 1** > **Scalar Field** node and set the following properties:

Property	Setting
Function	**Soot Density [mg/m3]**
Clip	**Off**
Min	`300.0`
Max	`800.0`

3. Select the **Color Bar** node and set *Color Map* to **soot_ramp**.

   At this point, the room is entirely filled with the volume rendered resampled part.

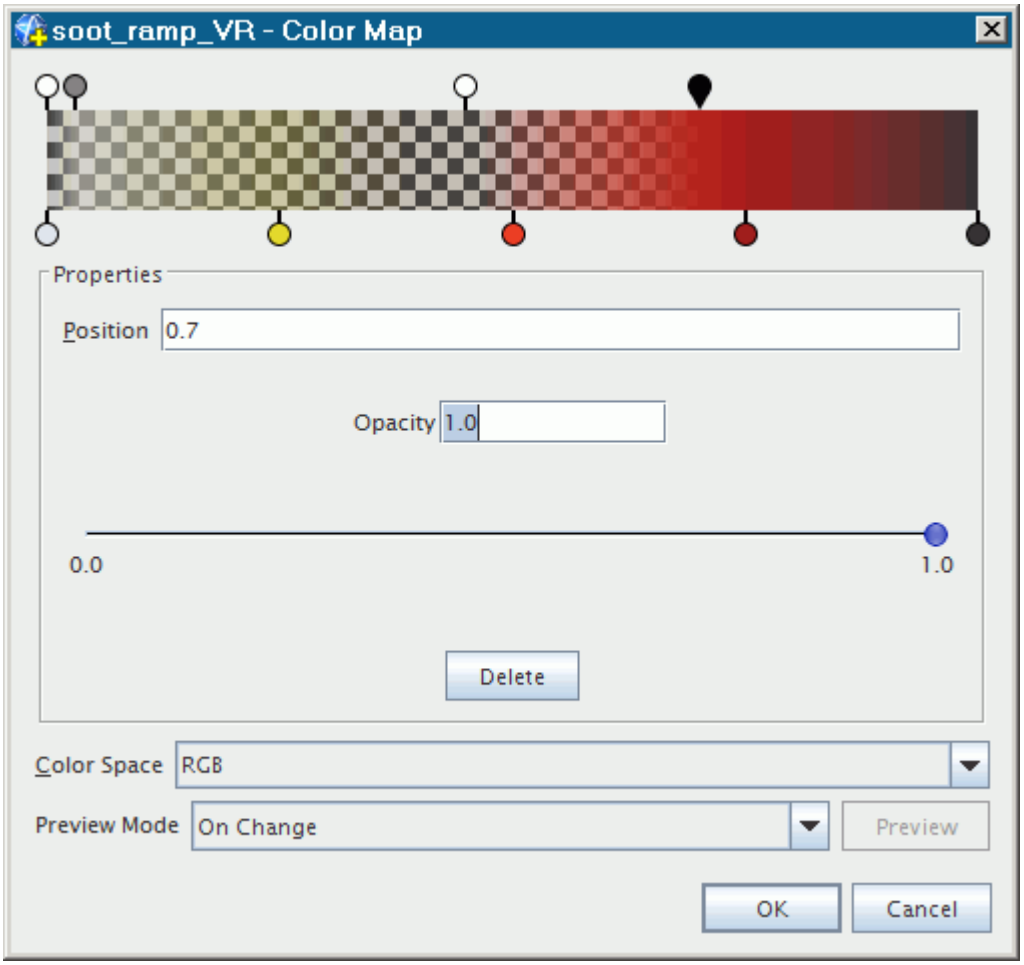


Modifying the Colormap Settings

To determine exactly where in the room the soot density is highest, change the opacity of the colormap.

1. Create a copy of the **soot_ramp** colormap node.
2. Rename the new colormap `soot_ramp_VR`.
3. To edit the **soot_ramp_VR** node, click ... (**Custom Editor**) for the *Color Map* property.
4. In the *soot_ramp_VR - Color Map* dialog:
   a) Add two opacity control points by clicking above the color spectrum for each one.
   A total of four opacity control points are above the spectrum.
   b) Select each control point, beginning with the one at the far left, and set its *Position* and *Opacity*:

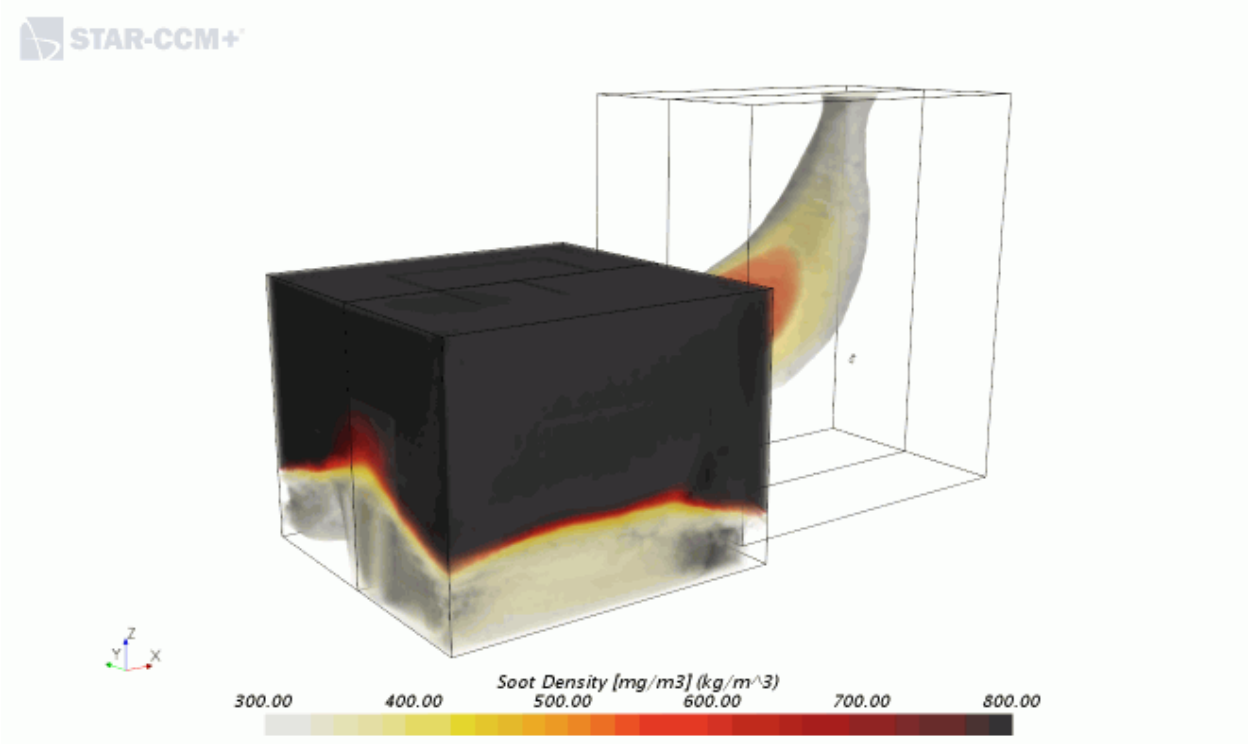
Opacity Control Point	Position	Opacity
First	0.0	0.0
Second	0.03	0.5
Third	0.45	0.0
Fourth	0.7	1.0



c) Click **OK**.

5. To preview the soot density colormap in the plane section, select the **Scalar Scene 1** > **Displayers** > **Resampled Volume Scalar 1** > **Color Bar** node and set *Color Map* to **soot_ramp_VR**.

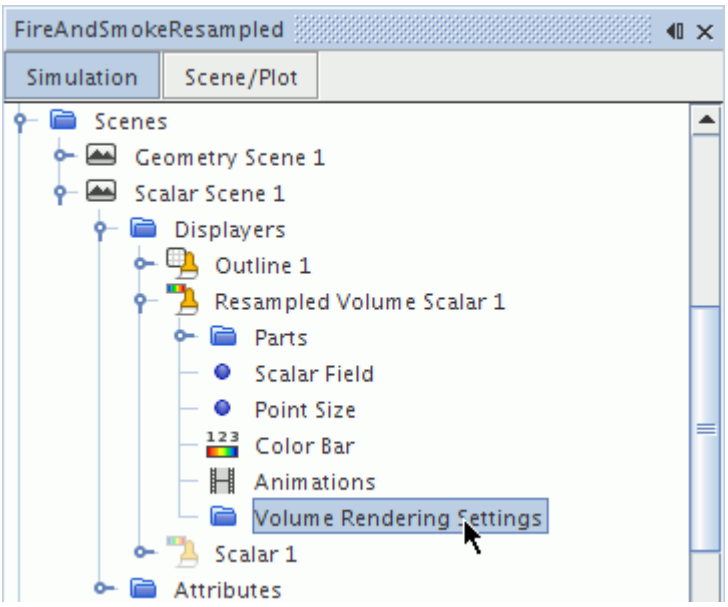
   The scene uses the new color set.



Adjusting the Lighting

To improve the appearance of the volume-rendered object, change lighting in the volume rendering settings and the displayer.

1. Select the **Resampled Volume Scalar 1** > **Volume Rendering Settings** node.



2. Set the following properties:

Property	Setting
Lighting Mode	**Local Lighting**
Quality	`0.8`

3. Select the **Resampled Volume Scalar 1** node and click ... (**Custom Editor**) for the *Lighting* property.
4. In the *Resampled Volume Scalar 1 - Lighting* dialog, set the following:

Property	Setting
specular	`0.7`
specular power	`25.0`

5. Click **OK**.

Setting Solver Parameters and Stopping Criteria

Prepare the simulation for a new analysis run in order to see how the soot cloud changes over time.

Change the time-step and run time as follows:

1. Select the **Solvers** > **Implicit Unsteady** node and set *Time-Step* to `0.1` s.
2. Select the **Stopping Criteria** > **Maximum Physical Time** node and set *Maximum Physical Time* to `25.0` s.

Setting Up the Solution History File

From the new transient analysis, you want to obtain an animation of the soot cloud developing over time. In order to generate the input frames for the animation, create a solution history file and use it to store selected solution data at specified time intervals.

1. Right-click the **Solution Histories** node and select **New...**.
2. In the *Save* dialog, enter `FireAndSmokeResampledData.simh` as the name of the solution history file and click **Save**.

   Within the **Solution Histories** node, STAR-CCM+ adds a new node containing the name of the solution history file.

   The red asterisk next to this node means that data is actively written to the file when the simulation runs.

Choose what data to save to the solution history file. As this is a fire and smoke case it is appropriate to store results for soot density:

3. Select the **Solution Histories** > **FireAndSmokeResampledData** node and set *Scalar Functions* to **Density**, **Passive Scalar**, and **Soot Density [mg/m3]**.

Set the frequency with which the selected data is written to the solution history file:

4. Select the **Solution Histories** > **FireAndSmokeResampledData** > **Update** node and set *Trigger* to **Time Step**.

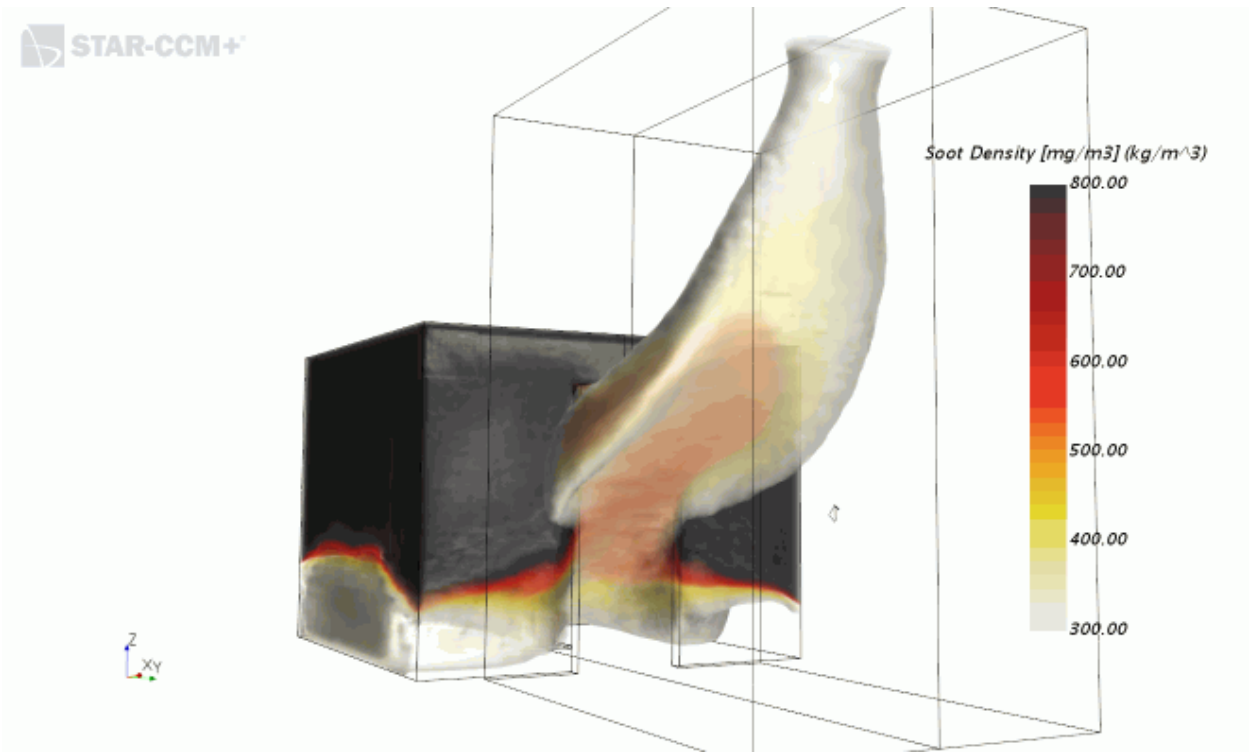
5. Select the **Update** > **Time-Step Frequency** node and make sure that *Frequency* is set to `1`.
6. Select the **Solution Histories** > **FireAndSmokeResampledData** node.

   The *Auto-record* property records the data to the solution history file at the required intervals. The *States* property displays the number of saved states stored in the selected solution history file; currently this is displaying 0 as the solver has not yet run.
7. Save the simulation.

Visualizing the Solution

Rearrange the scalar scene for an effective view angle and add a time annotation.

1. Rearrange the color bar and display to provide the following view.



Add an annotation displaying solution time to the scene:

2. Expand the **Tools** > **Annotations** node and drag the **Solution Time** node into the scene.

   The Solution Time annotation appears at the bottom left of the scene.
3. Drag the annotation to a more visible position in the scene.

Running the Simulation

Clear the results of the previous simulation and run the simulation again.

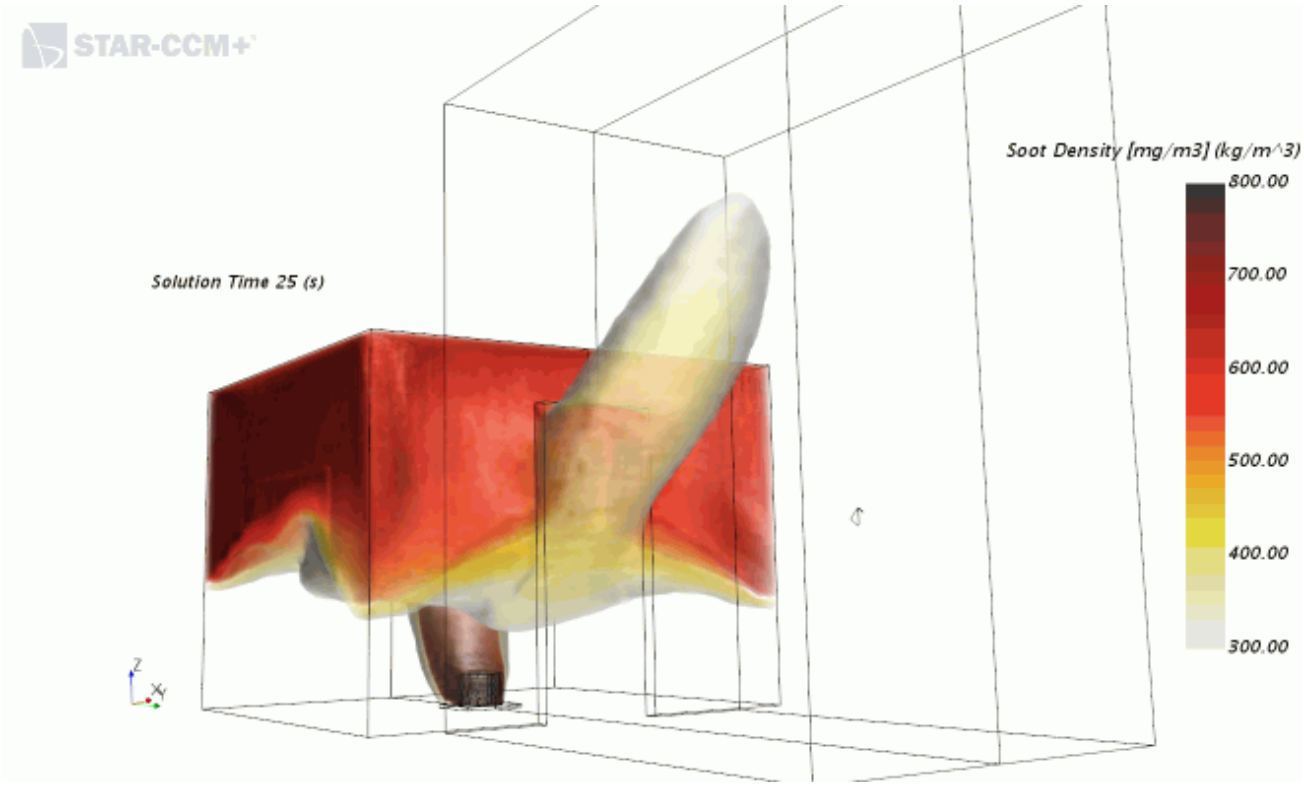
To clear the results and run the simulation again:

1. In the menu, select **Solution** > **Clear Solution**.
2. In the *Clear Solution* dialog, click **OK**.
3. Click (**Run**) in the *Solution* toolbar.
4. When the simulation has finished running, save it.

Visualizing the Results

When the simulation run is finished, the scalar scene shows the complete volume rendering of the soot density.

Take a look at the scalar scene.



Creating a Recorded Solution View

Solution views are used to interrogate the solution history data and make it available for post-processing.

Properties of the solution view set the point in the solution history at which data is read. Data is read into a separate representation linked to the solution view. The solution history file contains all the data that was specified in the previous part of the tutorial.

1. Right-click the **Solution Histories** > **FireAndSmokeResampledData** node and select **Create Recorded Solution View**.
2. Click **Yes** in the confirmation dialog.

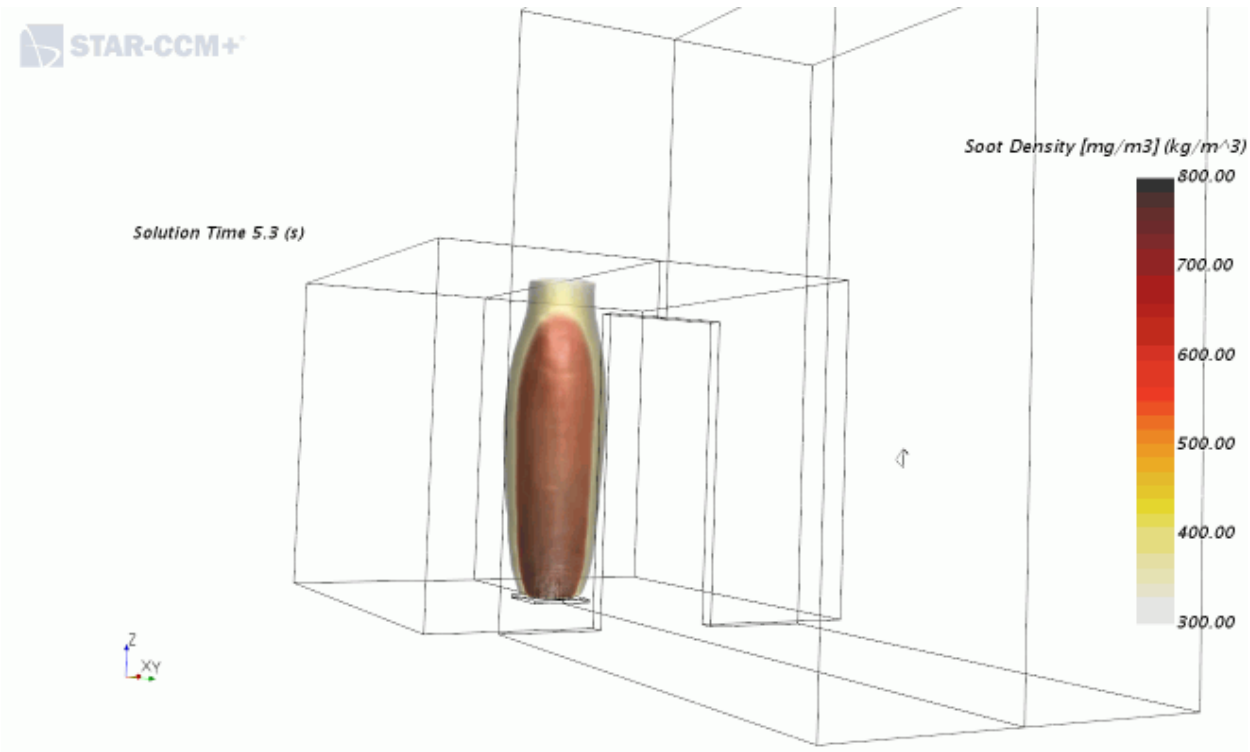
A new node, **FireAndSmokeResampledData**, is added to the **Solution Views** node.

3. Click the **Scalar Scene 1** tab in the **Graphics** window.
4. Drag and drop the **Solution Views** > **FireAndSmokeResampledData** node onto a blank area in the scene window.

   This action sets the *Representation* property for all displayers (**Outline 1**, **Resampled Volume Scalar 1**, and **Scalar 1**) and solution history annotation (**Solution Time**) in the scene to **FireAndSmokeResampledData**.

   This scene corresponds to the data stored in the solution history file for the first time-step. Adjust the solution time to display the solution data at 5.3 seconds.
5. Select the **Solution Views** > **FireAndSmokeResampledData** node and set the *Solution Time* property to approximately 5.3 seconds.

   The scene updates and the annotation indicates the time that you selected.



6. Save the simulation.

Creating an Animation from the Solution View

Create an animation that shows the spread of soot in the room.

1. Select the **Solution Views** > **FireAndSmokeResampledData** > **Animation** node and set the *Animation Mode* to **Solution Time**.

   A new node, **Solution Time Animation**, appears below the **Animation** node.

You have 250 states and a physical time of 25 seconds. Knowing this, you can calculate that 10 frames make up one second of simulation time.

Adjust the framerate for the animation.

2. Click the **Scene/Plot** tab in the explorer pane.
3. Select the **Scalar Scene 1** > **Attributes** > **Scene animation** node and set the *Target frame rate (fps)* to `10`.

Record the animation to a video file.

4. Click (**Write Movie**)
5. In the *Write animation* dialog, set the following properties:

Property	Value
Animation Length	`25`
Size	A resolution of your choice
File Name	`sootAnimation.avi`

6. Click **Save** to write the animation to disk.
7. Play back the animation using a player of your choice.

Summary

This tutorial demonstrated volume rendering and other advanced post-processing features in STAR-CCM+.

The features that were introduced include:

- Refining the settings of a scalar in a scene.
- Setting up a resampled volume derived part.
- Preparing a custom colormap.
- Viewing volume-rendered data.
- Adjusting display lighting.
- Creating a Solution History file.
- Creating a Recorded Solution View.
- Viewing recorded solution data in a scene.
- Recording an animation from the Solution History file.

Data Focus: Thermal Comfort

Data focus delivers a way to expose critical engineering results and improve decision making. It lets you select areas of interest within one or more plots and immediately visualize the data in those areas within connected scenes.

With this live and interactive link between numerical and visual data, you can examine unexpected or non-intuitive results, for example:

- Locate unburned fuel in areas where local temperatures are high.
- Find zones where residence time is high in areas where flow speed is also high.
- Determine where bubbles can promote better mixing through enhanced local turbulence.

In this tutorial, you explore relationships between temperature and velocity fields for a thermal comfort scenario. The tutorial begins by reviewing face sizes on the surface of the passenger. In an industrial case, this study could lead to further refinement of particular areas on a mesh.

There are multiple methods for applying data focus in a simulation; this tutorial presents two alternatives. One is to edit a data focus object directly; the other is to edit relevant properties within scenes and plots.

Perhaps the greatest benefit of data focus comes when you link resampled volume data with scatter plots that cover entire regions. You can survey large data sets interactively without creating any additional derived parts. In this tutorial, although the data set is much smaller than an industrial simulation, the workflow is the same.

Contents:

Prerequisites
Preparing the Simulation
Examining the Mesh
Exploring the Correlation of Velocity Magnitude and Temperature
Exploring Results in 3D
Summary

Prerequisites

The instructions in the Data Focus tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Preparing the Simulation

For this tutorial, you use an existing simulation file that contains objects and solution data from the tutorial, Thermal Comfort Wizard: Single Occupant in a Cabin.

To start this tutorial, you can either:

- use your own version of the simulation file from the Thermal Comfort tutorial, `personCabin.sim`, if you have already worked through it; or,
- use the version from the Tutorials Files package that you can download from the Steve Portal. To obtain the package, follow the instructions in the section, Downloading the Tutorial Files. The file in the package is `tutorials/heatTransferAndRadiation/ThermalComfortWizardSingleOccupantInACabin_final.sim`.

To prepare the simulation file for this tutorial:

1. Launch STAR-CCM+.
2. Select **File** > **Load...**
3. In the *Load a File* dialog, click **Browse...**
4. In the *Open* dialog, navigate to the directory that contains the simulation file, select the file, and click **Open**.
5. In the *Load a File* dialog, click **OK**.
6. Save the simulation as `thermalComfortDataFocus.sim`.
7. Expand **Scenes** and delete the existing scene objects.

For when you want to select groups of objects multiple times, STAR-CCM+ allows you to create filters that subsequently appear in object selection dialogs. Here, you create filters for selecting groups of region boundary surfaces:

8. Right-click the **Tools** > **Filters** node and select **New**.

   The filter is added to the **Filters** node.
9. In the *Filter* window, select **Objects (Static)** from the drop-down menu.

   A second drop-down menu appears with, **are**, already set.
10. Click ... (**Custom Editor**).
11. In the *Select Objects* dialog, select the following objects in the **Regions** > **Fluid** > **Boundaries** node:
    - **1 - Head**
    - **2 - Torso**
    - **3 - LeftUpperArm**
    - **4 - RightUpperArm**
    - **5 - LeftLowerArm**
    - **6 - RightLowerArm**
    - **7 - LeftHand**
    - **8 - RightHand**
    - **9 - LeftUpperLeg**
    - **10 - RightUpperLeg**

- **11 - LeftLowerLeg**
- **12 - RightLowerLeg**
- **13 - LeftFoot**
- **14 - RightFoot**

12. Click **OK** to close the *Select Objects* dialog.
13. Click (**Commit changes**).
14. Close the *Filter* window.
15. Rename the **Filter** node to `Body Boundary Surfaces`.

    The first filter is now complete.
16. Create a second filter using the same technique, except:
    a) Select the following objects in the **Regions** > **Fluid** > **Boundaries** node:
    - **Cabin**
    - **Console**
    - **Doors**
    - **Roof**
    - **Seats**
    - **Windows**

    b) Rename the second **Filter** node to `Car Boundary Surfaces`.
17. Save the simulation.

Examining the Mesh

For meshing, data focus tools give you a way to browse across the surface of a mesh and examine an appropriate metric. In this instance, you view the area of cell faces.

To look at the distribution of face sizes, plot position against area magnitude.

First, prepare the geometry scene for viewing the data:

1. Create a geometry scene.
2. Rename the scene to `Area Magnitude`.

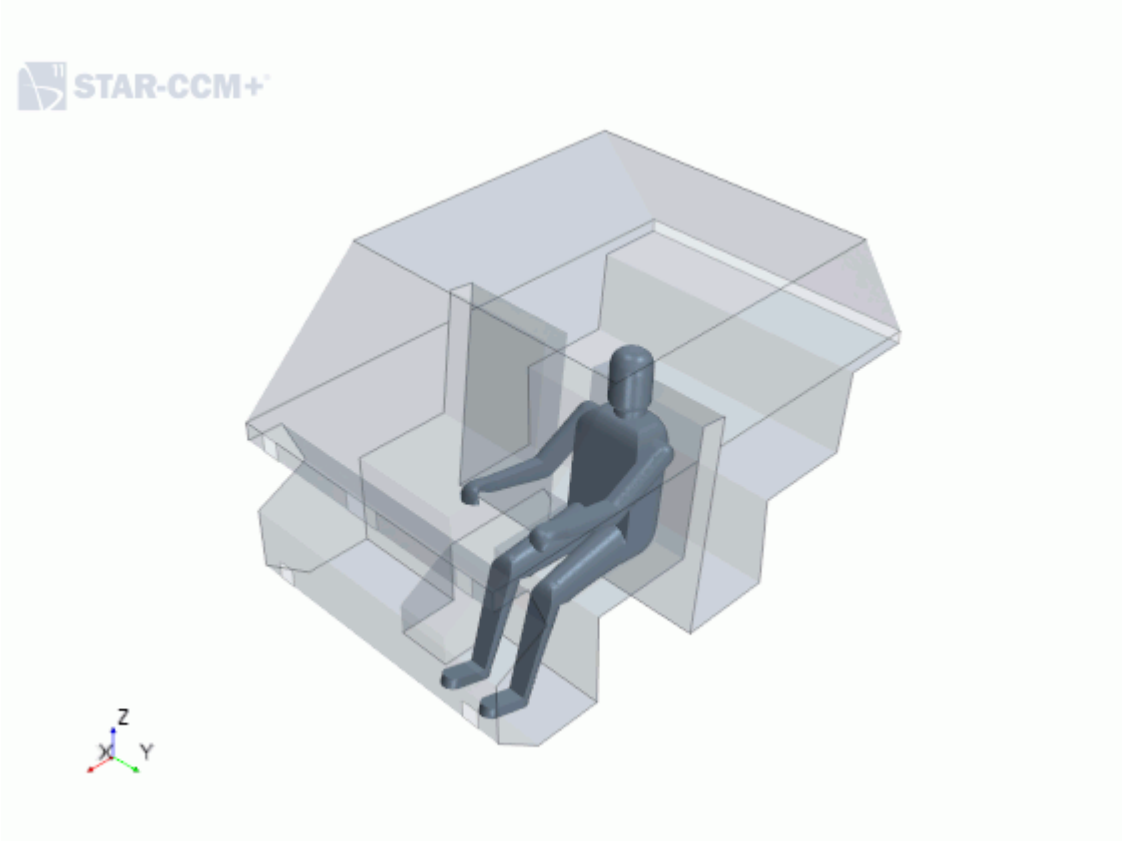
Assign parts to the scene using the filters that were created previously:

3. Within the **Scene/Plot** panel, select the **Area Magnitude** > **Displayers** > **Geometry 1** > **Parts** node.
4. In the *Parts* property, click (**Custom Editor**).
5. Deselect all parts in the object selector.
6. At the top left of the object selector, click (**Show/Hide Filters**).
7. Activate **Body Boundary Surfaces**.

   The contents of the *Tree View* window shows only the parts of the driver (wall boundaries).
8. Select all the boundary parts showing and click **OK**.
9. Select the **Area Magnitude** > **Displayers** > **Outline 1** node, activate *Surface* and set *Opacity* to `0.18`.

10. Similarly, select the **Outline 1** > **Parts** node and set *Parts* to the contents of the **Car Boundary Surfaces** filter.

11. Click (**Save-Restore-Select views**) and select **Views** > **+X +Y +Z** > **Up +Z**.



12. Click (**Save-Restore-Select views**) and select **Store Current View**.

Set up the plot of position against area magnitude:

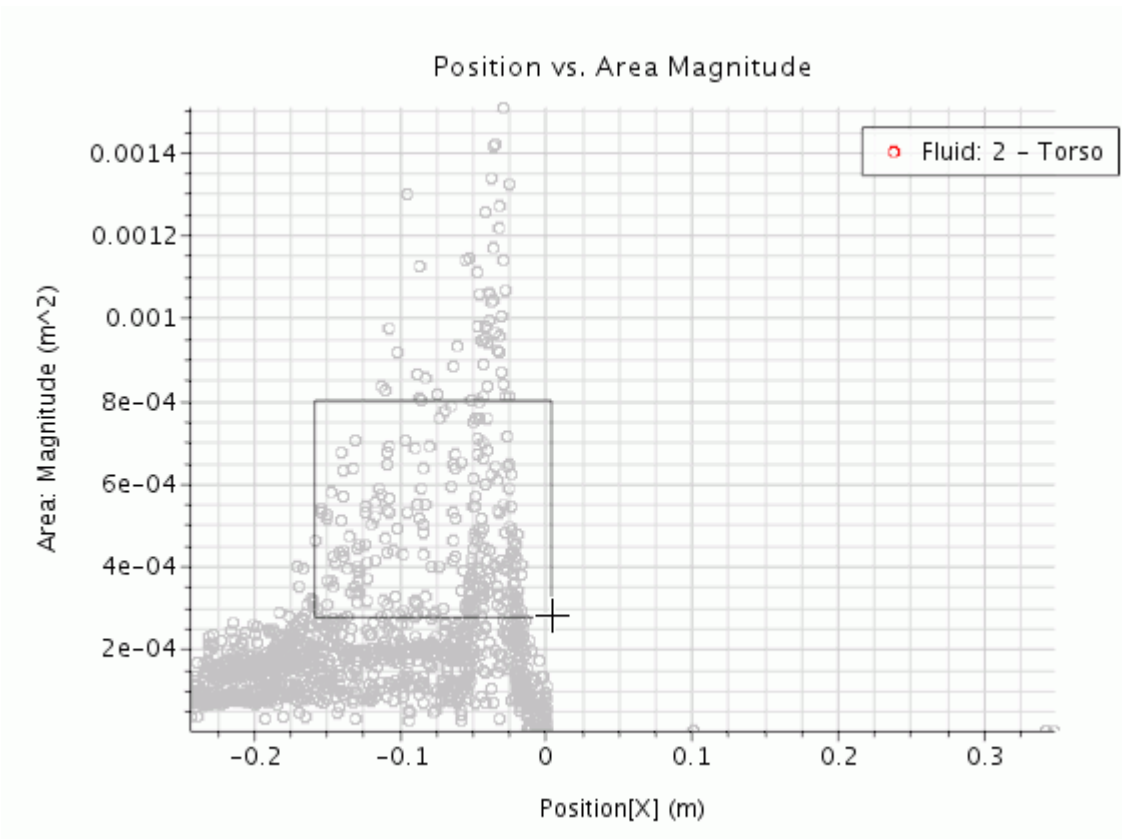
13. Create an XY plot.
14. Rename the plot to `Position vs. Area Magnitude`.
15. Edit the **Position vs. Area Magnitude** plot node and set the following properties:

Node	Property	Setting
Position vs. Area Magnitude	*Title*	`Position vs. Area Magnitude`
	Parts	In **Regions** > **Fluid** > **Boundaries**: **2 - Torso**
└ **X Type**	*Data Type*	**Scalar**
└ **Scalar Function**	*Field Function*	**Position(X)**
└ **Y Types**		

Node	Property	Setting
└ **Y Type 1** > **Scalar Function**	*Field Function*	**Area: Magnitude**

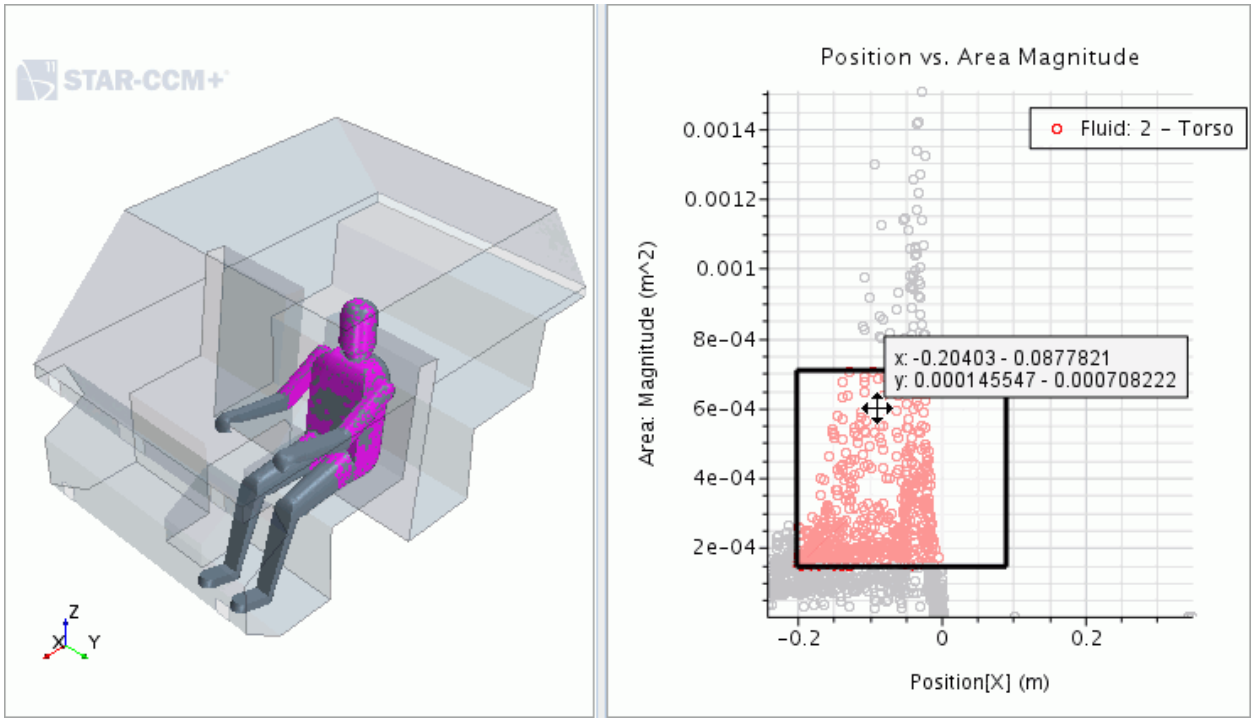
Create the data focus. There are two methods for applying data focus objects. The following steps demonstrate one method, and the other method will be shown later.

16. Right-click the **Tools** > **Data Focus** node and select **New**.
17. Rename the **Data Focus 1** node to `Position vs. Area Magnitude`.
18. Right-click the **Position vs. Area Magnitude** node and select **Apply To**.
19. In the *Apply Data Focus* object selector, select the following nodes:
    - **Scenes** > **Area Magnitude** > **Displayers** > **Geometry 1**
    - **Plots** > **Position vs. Area Magnitude** > **Y Types** > **Y Type 1**
20. Click **OK**.
21. Select the **Plots** > **Position vs. Area Magnitude** > **Y Types** > **Y Type 1** > **Data Focus** node and set *Coloring Mode* to **Mask**.
22. In the **Plot** toolbar, click (**Create Data Focus Selection**), select **Add Selection to Data Focus**, and draw a box in the plot with the mouse.



23. Place the scene and the plot side by side, and use the mouse to move, or "brush", the data focus box around the plot. Observe how the highlighted area changes in the scene.

Use this technique to find out which faces have the largest area. In a real scenario, these faces could require further refinement in order to achieve a uniform distribution of face sizes.



24. Save the simulation.

Exploring the Correlation of Velocity Magnitude and Temperature

Use a data focus to explore the relationship between velocity magnitude and temperature. A combination of high velocity and low temperature could be uncomfortable for the driver.

Set up the plot of velocity magnitude against temperature:

1. Create an XY plot.
2. Rename the plot to `Velocity Magnitude vs. Temperature`.
3. Edit the **Velocity Magnitude vs. Temperature** plot node and set the following properties:

Node	Property	Setting
Velocity Magnitude vs. Temperature	*Title*	`Velocity Magnitude vs. Temperature`
	Parts	**Regions** > **Fluid**
└ **X Type**	*Data Type*	**Scalar**
└ **Scalar Function**	*Field Function*	**Velocity: Magnitude**

Node	Property	Setting
Y Types		
Y Type 1 > **Scalar Function**	*Field Function*	**Temperature**
Y Type 1 > **Fluid** > **Heatmap Style**	*Bins*	`[128, 128]`
	Use Color Lookup Table	**Activated**

Prepare the vector scene:

4. Create a vector scene and rename it, `Velocity`.
5. Click (**Save-Restore-Select views**) and select **Restore View** > **View 1**, which was created in the tutorial, Thermal Comfort Wizard: Single Occupant in a Cabin.
6. Right-click **Velocity** > **Displayers** and select **New Displayer** > **Geometry**.
7. Select **Displayers** > **Geometry 1** and activate *Surface*.
8. Select the **Geometry 1** > **Parts** node and set *Parts* to the contents of filter **Body Boundary Surfaces**.
9. Select the **Displayers** > **Outline 1** node, activate *Surface*, and set *Opacity* to `0.18`.
10. Select the **Displayers** > **Outline 1** > **Parts** node and set *Parts* to the contents of filter **Car Boundary Surfaces**.
11. Select the **Displayers** > **Vector 1** node and set *Display Mode* to **Line Integral Convolution**.

Add derived parts to the vector scene.

12. Create a plane section derived part whose *Normal* is `[0, 1, 0]`.
13. Add the plane section to the vector displayer of the vector scene.

    STAR-CCM+ displays the fluid velocity on the plane.
14. Create a streamline displayer in the **Velocity** scene.
15. Select the **Streamlines 1** > **Parts** node and set *Parts* to **Streamline**.
16. Select the **Streamlines 1** > **Scalar Field** node and set *Function* to **Temperature**.
17. Rearrange the **Velocity** color bar so that the values appear clearly.

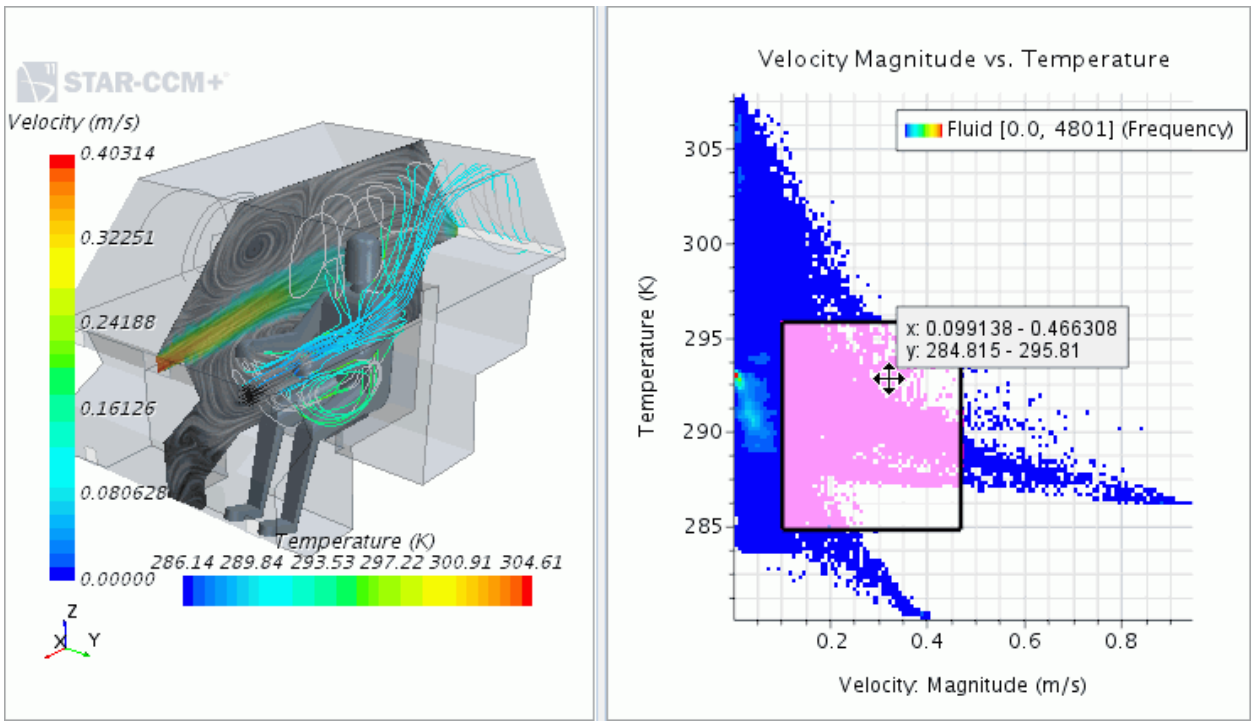
Prepare the data focus using the second method, that is, applying the data focus using the related properties of the plot and scene directly:

18. Within **Tools** > **Data Focus**, create a data focus and rename it, `Velocity Magnitude vs. Temperature`.
19. Select the **Plots** > **Velocity Magnitude vs. Temperature** > **Y Types** > **Y Type 1** node and set *Data Focus* to **Velocity Magnitude vs. Temperature**.
20. Edit the **Scenes** > **Velocity** > **Displayers** node and set the following properties:

Node	Property	Setting
Displayers		

Node	Property	Setting
Streamlines 1	*Data Focus*	**Velocity Magnitude vs. Temperature**
Vector 1	*Data Focus*	**Velocity Magnitude vs. Temperature**

21. In the *Graphics* window, arrange the *Velocity* scene and the *Velocity Magnitude vs. Temperature* plot so that they are both beside each other.
22. Select the *Velocity Magnitude vs. Temperature* plot. In the **Plot** toolbar, click (**Create Data Focus Selection**), select **Add Selection to Data Focus**, and draw a box across the plot.
23. Move the box to examine how the combined velocity and temperature affects driver comfort.



When you look at temperature together with velocity, you can locate the least comfortable zones in the vehicle cabin based on the highest velocity and lowest temperatures.

Exploring Results in 3D

Explore similar data, but in a much larger quantity, and discover how to find the essential results using data focus.

If your graphics card does not support volume rendering, activate advanced rendering for the relevant scene.

In the previous exercise, you worked with streamlines and vectors (with line integral convolution) to examine velocity magnitude and temperature. However, the amount of data available was relatively limited. In this

exercise, volume rendering makes a vast amount of data available in a 3D space. At the same time, this larger amount of data is manageable with data focus, which lets you explore results and visualize interdependence between data fields without creating numerous derived parts.

Prepare the scene:

1. Create a geometry scene and rename it, `Velocity 3D`.
2. Click (**Save-Restore-Select views**) and select **Restore View** > **View 1**.
3. Select the **Displayers** > **Geometry 1** > **Parts** node and set *Parts* to the contents of filter **Body Boundary Surfaces**.
4. Select the **Displayers** > **Outline 1** node, activate *Surface*, and set *Opacity* to `0.18`.
5. Select the **Displayers** > **Outline 1** > **Parts** node and set *Parts* to the contents of filter **Car Boundary Surfaces**.

Add a resampled volume of the fluid velocity:

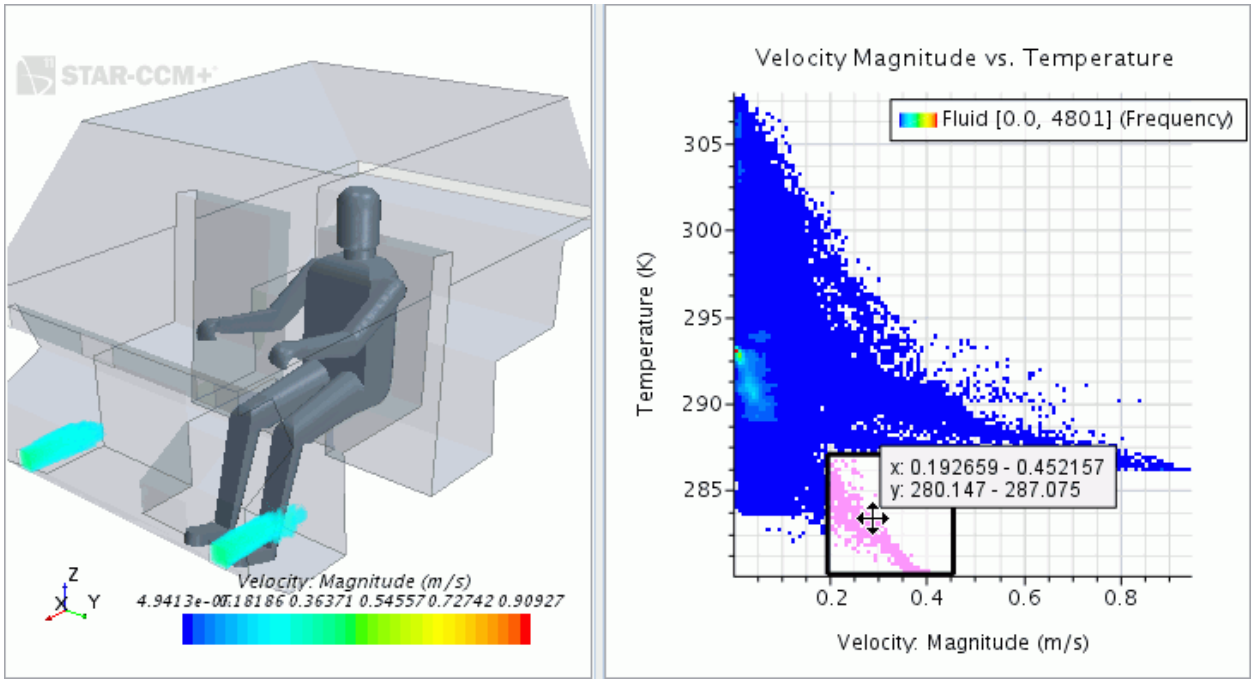
6. To create a resampled volume, right-click the **Derived Parts** node and select **New Part** > **Resampled Volume...**
7. In the dialog, set the following properties:

Property	Setting
Input Parts	**Fluid**
Cell Size to Voxel Ratio	`0.01`
Display	**New Scalar Displayer**

8. Click **Create** followed by **Close**.
9. Within **Velocity 3D**, select **Displayers** > **Resampled Volume Scalar 1** and set *Data Focus* to **Velocity Magnitude vs. Temperature**.
10. Edit the **Resampled Volume Scalar 1** node and set the following properties:

Node	Property	Setting
└ **Data Focus**	*Coloring Mode*	**Clip**
└ **Scalar Field**	*Function*	**Velocity: Magnitude**

11. Use the data focus to examine velocity in 3D. Brush the area of the plot with the coolest temperatures and medium velocity.



Summary

This tutorial demonstrated data focus and other advanced post-processing features in STAR-CCM+.

The features that were introduced include:

- Examining the mesh with data focus.
- Using line integral convolution in a vector scene.
- Correlating velocity (on a plane section) with temperature (on streamlines).
- Setting up a resampled volume derived part.
- Viewing volume-rendered data.

Simulation History: Lagrangian DEM on an Auger

This tutorial demonstrates STAR-CCM+ simulation histories with Lagrangian DEM, using an auger that moves biomass pellets of different sizes.

With simulation history and animation you examine the movement of the pellets throughout the simulation. You also use data focus to assess the pellet movement further.

Simulation histories provide two major benefits for working with solution data. You can save considerable disk space by choosing to export solution data only from a selected source, such as boundary surfaces. In post-processing, you can quickly navigate between stored time-steps.

Data focus tools help you locate portions of data that require closer examination. In this tutorial, you use data focus to locate pellets of certain sizes.

Simulating this movement answers some useful engineering questions: How fast do the pellets move through this auger? Equally important, do the different sizes stay mixed? If not, and one size emerges first, then that size is packaged separately from the other size, resulting in discrepancies in quality.

The objects in this simulation consist of the following specifications:

- Blade length (approx.): 2 m
- Blade diameter (approx.): 0.2 m
- Blade rotation rate: 400 rpm
- Pellet sizes (cylinder height and radius): Variable calculations, using global parameters and injector values

Contents:

Prerequisites
Loading an Existing Simulation
Setting Up Solution History Files and Running the Simulation
Post-Processing the Results
Examining the Cylinder Sizes with Data Focus
Summary

Prerequisites

The instructions in this tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Loading an Existing Simulation

For this tutorial, you use an existing simulation file that contains some of the required objects already prepared.

You are advised to run this tutorial in parallel, using at least 4 cores. This tutorial case has a large number of cells and can take a long time to run if using only one core.

Load the provided simulation file:

1. Start up STAR-CCM+ in a manner that is appropriate to your working environment. Remember to specify multiple cores.
2. Load `auger.sim` from the `[INSTALL_DIR]/doc/startutorialsdata/analysisMethods/data` directory.

   STAR-CCM+ provides feedback on the loading process in the *Output* window.
3. Save the simulation as `auger1.sim`.

Setting Up Solution History Files and Running the Simulation

In this exercise, you set up two solution histories: one for the auger blade movement, and the other for the Lagrangian DEM pellets.

The simulation history for the auger blade records movement of the blade, which is specified through the selection of **Rotation** as the motion of the **rotor** region. With the simulation history file, you are only recording data for the relevant boundary surfaces rather than for the entire region. This approach saves considerable disk space.

The simulation history for the pellets records their movement using the Discrete Element Method (DEM) of modeling particles. The pellets enter the auger using Lagrangian injectors; these injectors include specification of the height and radius of the cylindrical pellets on entry in the auger casing. For calculating the height and radius generally, global parameters are used. For more information about this helpful feature, see Using Tools > Global Parameters in the User Guide.

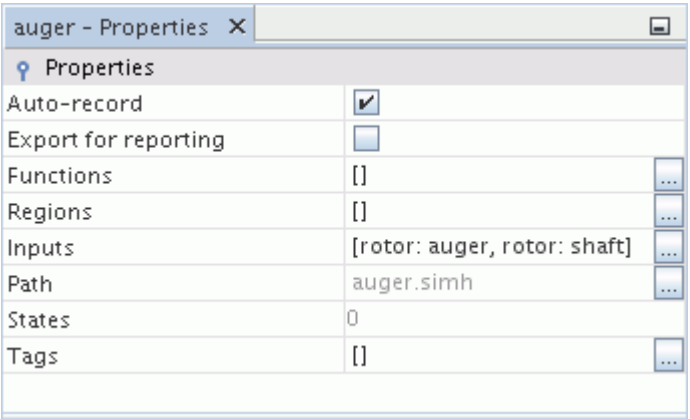
1. Right-click the **Solution Histories** node and select **New**.
2. Save the solution history file as `auger`.

   Next to the new **auger** node in the object tree, the red asterisk means that data is actively written to the file when the simulation runs. The *Auto-record* property controls this process.
3. To choose what data to write to the solution history file, and the frequency of writing it, edit the **Solution Histories** > **auger** node and set the following properties:

Node	Property	Setting
auger	*Auto-record*	Activated
	Regions	Remove the regions **rotor** and **casing**.
	Inputs	In **Regions** > **rotor** > **Boundaries**: **auger**

Node	Property	Setting
		shaft
└ **Update**	*Trigger*	**Time Step**
└ **Time-Step Frequency**	*Frequency*	`1`

The *Properties* window should resemble the following screenshot when you finish:



The following properties help you make the best use of the simulation history feature:

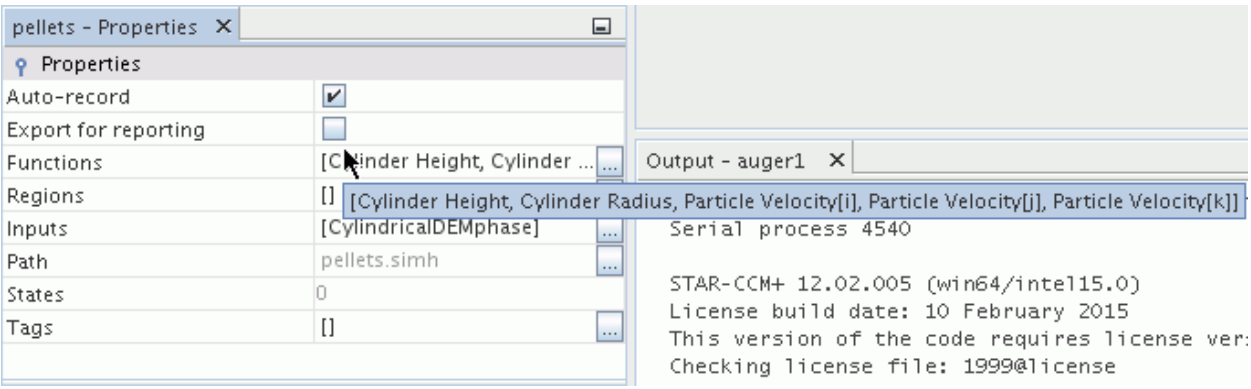
- *Auto-record* - When activated, the selected solution data is written to the simulation history file at the specified time intervals. When this option is activated, an asterisk appears besides the simulation history file node in the object tree. This property is activated by default.
- *Export for reporting* - When activated, the export includes any auxiliary field functions that are required for all standard scalar and vector reports. This property is deactivated by default. See Simulation History Phase Data in the User Guide.
- *Inputs* - This property lets you record only surface data to the `.simh` file, whereas selecting regions (using the *Regions* property) causes mesh data to be included. Since you are plotting the geometry at every time-step, the inclusion of mesh data would result in a much larger amount of recorded data.

4. Create another solution history for the Lagrangian DEM for the pellets. Save the `.simh` file as `pellets`.
5. To choose what data to save to the solution history file, along with the frequency, edit the **pellets** node and set properties as indicated in the table below. As this is a Lagrangian DEM case it is appropriate to store results for cylinder height, cylinder radius, and particle velocity.

Node	Property	Setting
pellets	*Auto-record*	Activated
	Functions	Within **Particle** > **[Angular Momentum ... Volume Overlap]**: **Cylinder Height** **Cylinder Radius** **Velocity** > **Laboratory** > **i** **Velocity** > **Laboratory** > **j**

Node	Property	Setting
		Velocity > **Laboratory** > **k**
	Regions	Remove the regions **rotor** and **casing**.
	Inputs	**Lagrangian Phases** > **CylindricalDEMphase**
└ **Update**	*Trigger*	**Time Step**
└ **Time-Step Frequency**	*Frequency*	`1`

The *Properties* window should resemble the following screenshot when you finish:



The hover text indicates the functions that are selected for this solution history.

6. Click (**Run**) in the *Solution* toolbar.
   The *Residuals* display is not pertinent to this tutorial because only DEM particle behavior is of interest.
7. After the run is finished, save the simulation.

Post-Processing the Results

You set up a scalar scene to view the solution. However, with solution views, you can also look at the results at every step. This feature lets you answer the question about how fast the pellets move through the auger.

You use clip planes in the scene to get a cutaway view of the geometry, and display the pellets inside it. Lighting settings add a touch of realism to the objects in the scene.

To navigate between stored time-steps, you create solution views from the solution history objects. At the same time, STAR-CCM+ generates a representation for each solution view. You can only create solution views if STAR-CCM+ finds the `.simh` files and recognizes them as being compatible with the simulation file. Once you have representations of the `.simh` content, you can update your scenes to use those representations. In addition to viewing, you set properties of the solution views to control whether to view part or all of the solution.

Take a look at how the solution history is divided into *states*. Select the **Solution Histories** > **auger** node, and then the **pellets** node. The number of states stored in the solution history file is displayed next to the *States* property. There are 1000 states stored in each solution history file.

Create two solution views: one for the auger, one for the pellets:

1. Right-click the **Solution Histories** > **auger** node and select **Create Recorded Solution View**.
2. Click **Yes** in the confirmation dialog.
3. Edit the **Solution Views** > **auger** node and set the following properties:

Node	Property	Setting
Animation	*Animation Mode*	**Time Step**
└ **Time Step Animation**	*Iteration Mode*	**Cycle Time**
	Auto Range	Deactivated
	Start Time Step	`1`
	End Time Step	`1000`
└ **Cycle Time**	*Value*	`1000.1`

4. Repeat the same procedure to set up a solution view for the **pellets** solution history node.

Set up a scalar scene for viewing the solution history:

5. Create a scalar scene and rename it `simh pellets`.
6. Switch to the **Scene/Plot** tree, select the **Displayers** > **Outline** node, and rename it `auger`.
7. Rename the **Displayers** > **Scalar 1** node `pellets`.
8. Right-click **Displayers** and select **New Displayer** > **Geometry**. Rename the new node `casing`.
9. Within **Displayers**, edit each of the following nodes and set their properties as listed:

auger

Node	Property	Setting
auger	*Surface*	Activated
	Color Mode	**Constant**
	Specular Highlight Mode	**Light**
	Lighting	`0,1,1,12` Each of these four numeric values specifies lighting differently, as follows: Ambient - light reflected or scattered from other surfaces in the scene Diffuse - the incidence of light onto a surface

Node	Property	Setting
		Specular - the direct reflection of the light source off a shiny object Specular power - the intensity of specular lighting
	Ignore Clip	Activated
	Representation	**auger**
└ **Parts**	*Parts*	In **Regions** > **rotor** > **Boundaries**: **auger** **shaft**
└ **Color**	*Color*	Click (**Custom Editor**) for the *Color* property. In the dialog that appears: 1. Select the **RGB** tab. 2. Set *Red* to `43`, *Green* to `60`, and *Blue* to `81`.

casing

Node	Property	Setting
casing	*Surface*	Activated
	Color Mode	**Constant**
	Specular Highlight Mode	**Light**
└ **Parts**	*Parts*	In **Regions** > **casing** > **Boundaries**: **casing_tube** **hopper**
└ **Color**	*Color*	**Burlywood R: 222, G: 184; B: 135**

pellets

Node	Property	Setting
pellets	*Point Size*	`1`
	Ignore Clip	Activated
	Point Scale Mode	**Particle Model**
	Representation	**pellets**

Node	Property	Setting
└ **Parts**	*Parts*	**Lagrangian Phases** > **CylindricalDEMphase**
└ **Scalar Field**	*Function*	**Particle Velocity [j]**
	Auto Range	**Off**
	Clip	**Off**
	Min	`-2.0`
	Max	`2.0`
└ **Color Bar**	*Color Map*	**purple-red basic (large difference)**

To view the auger mixing the pellets inside the casing, create a cutaway of the casing with a clip plane.

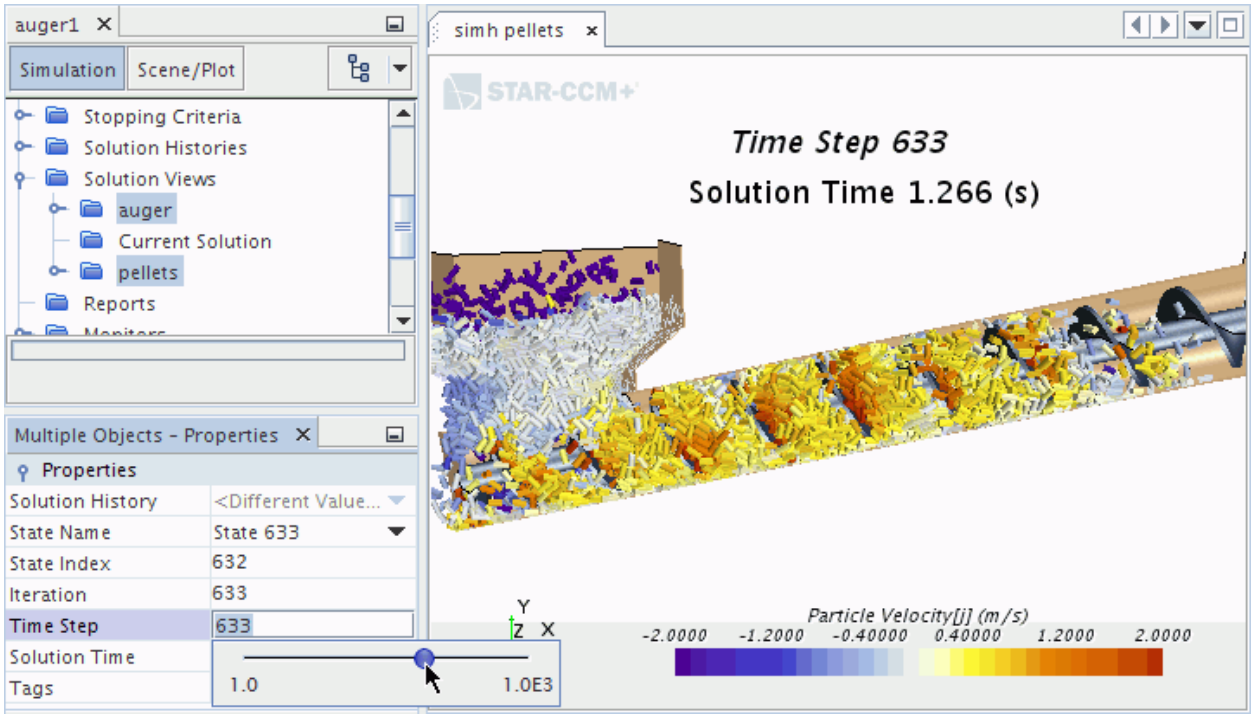
10. Select the **simh pellets** > **Attributes** > **Clip Planes** > **Plane 6** node and activate the *Enabled* property.

Set up annotations.

11. Switch back to the **Simulation** tree and expand the **Tools** > **Annotations** node.
12. Drag and drop two annotation nodes in the scalar scene: **Time Step** and **Solution Time**.
13. Move the annotations into appropriate positions in the scene.

Work with the solution views.

14. To see both solution states at the exact same point in time, switch to the **Simulation** tree and select both solution view nodes.
15. In the *Properties* window, click to the right of *Time Step*.
16. Use the slider to adjust time-steps and examine the solution in the scene.



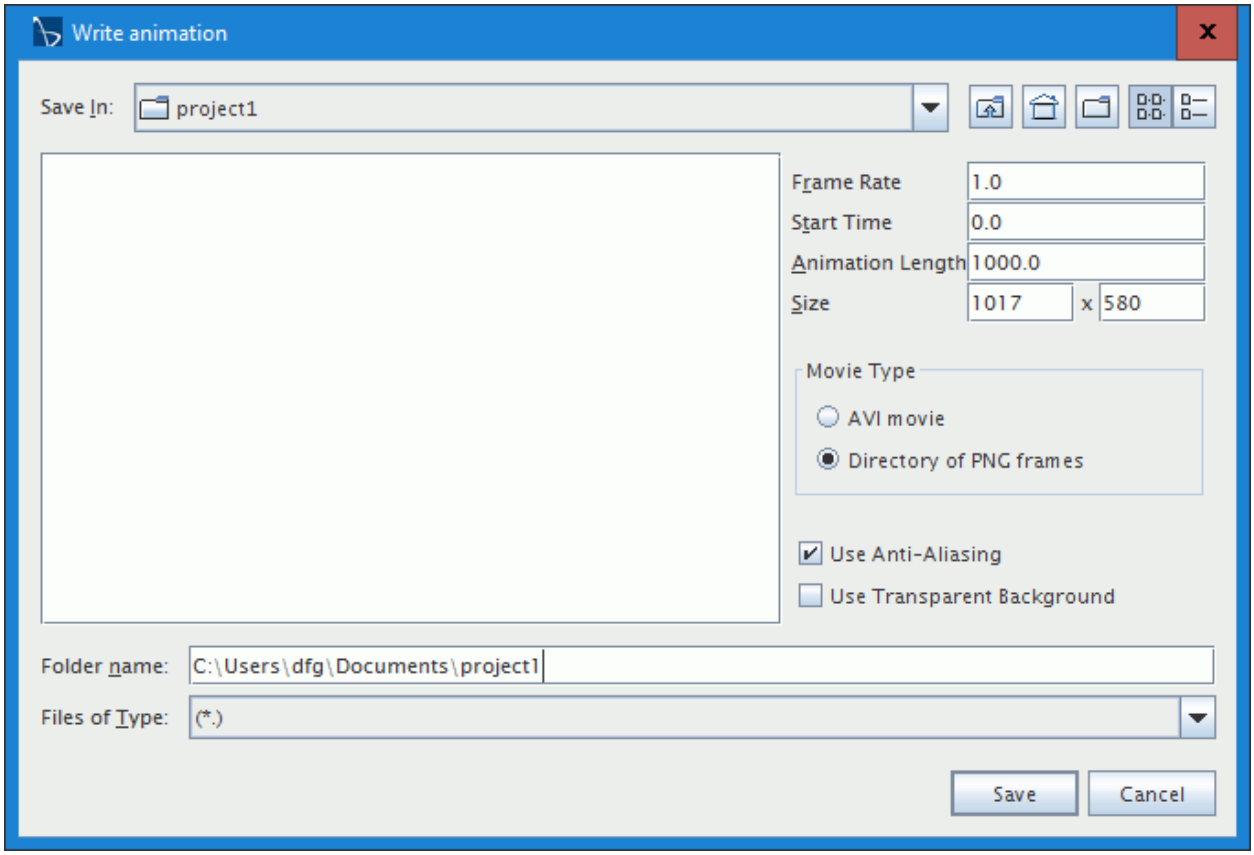
The `[j]` component is the vertical component of velocity. Using the specified colormap, you can see which pellets are moving up and down. Blue pellets are moving down, red ones are moving up, and pellets that are not moving up or down appear white.

17. To find out how fast pellets move through the auger:
    a) Move the time-steps back to the beginning. The first pellets are entering the auger.
    b) Adjust the time-steps forward until you see the first pellets reach the outlet of the auger.
    c) Look at the Solution Time annotation in the scene. It shows approximately 1.4 seconds.

    The pellets move through the auger in approximately 1.4 seconds.

Create an animation from a series of images that shows the movement of pellets in the auger. You are advised to set the size of the animation to match the size of your scene window.

18. To obtain the size of your scene window, select the **simh pellets** node and note the values for *Width* and *Height*.
19. In the *Animation* toolbar, click (**Write Movie**).
20. In the *Write animation* dialog, specify the creation of a stack of images along with other properties:



Property	Value
Frame Rate	`1.0`
Start Time (sec)	`0.0`
Animation Length	`1000.0`
Size	Enter the width and height in the first and second text boxes, respectively. You are advised to use the width and height of your scene window.
Movie Type	**Directory of PNG frames**
Folder Name	Enter the path for the images.

21. Click **Save** to generate the images.
22. Use the images to create an animation in the graphics program of your choice.

Examining the Cylinder Sizes with Data Focus

Use data focus tools to highlight cylinder data of interest. This feature lets you answer the question about whether the different sizes of pellets stay mixed.

With a data focus filter, you examine the locations of thin, long pellets and thick, short ones. In this exercise, you set up a plot with a heatmap, and link a data focus selection in the plot to the scalar scene. Then you move the data focus selection to highlight pellets with particular dimensions.

Set up the plot of cylinder radius against cylinder height.

1. Create an XY plot.
2. Rename the plot to `Cylinder Radius vs Height`.
3. Edit the **Cylinder Radius vs Height** plot node and set the following properties:

Node	Property	Setting
Cylinder Radius vs Height	*Title*	`Cylinder Radius vs Height`
	Parts	**Lagrangian Phases** > **CylindricalDEMphase**
	Representation	**pellets**
└ **X Type**	*Data Type*	**Scalar**
└ **Scalar Function**	*Field Function*	**Particle** > **Cylinder Height**
	Units	**mm**
└ **Y Types**		
└ **Y Type 1** > **Scalar Function**	*Field Function*	**Particle** > **Cylinder Radius**
	Units	**mm**
└ **Y Type 1** > **CylindricalDEMphase**	*Styles*	**Heatmap**
└ **Y Type 1** > **CylindricalDEMphase** > **Heatmap Style**	*Bins*	`128,64`

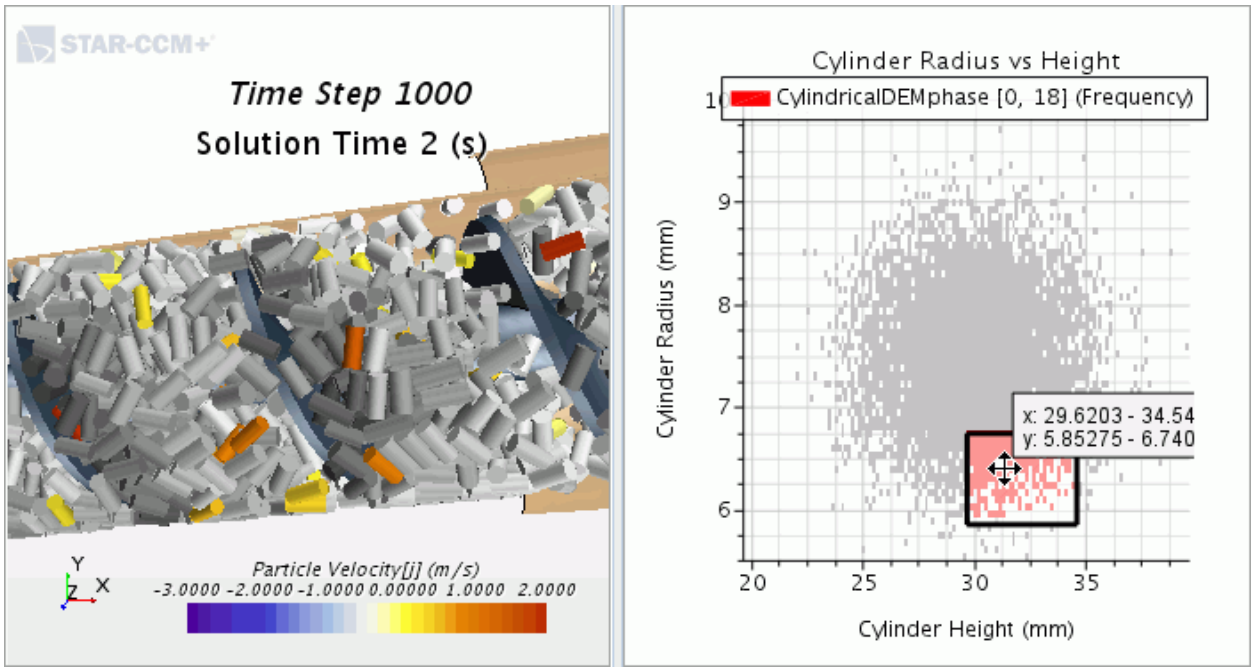
The *Bins* property sets the numbers of intervals of data along the x- and y-axes, respectively. The values for this exercise were chosen to render an optimum amount of data for visual examination.

Set up the data focus.

4. In the **Plot** toolbar, click (**Create Data Focus Selection**) and select **Add Selection to Data Focus**. Create a box in the plot display.

   This step also creates a **Data Focus 1** node in the **Tools** > **Data Focus** node.
5. In the graphics window, activate the *simh pellets* scene display.
6. Drag and drop the **Data Focus 1** node into the *simh pellets* scene display. In the context menu that appears, select **Apply to pellets**.
7. Select the **Plots** > **Cylinder Radius vs Height** > **Y Types** > **Y Type 1** > **Data Focus** node and set *Coloring Mode* to **Mask**.

8. To show the solution at the final time step, within the **Solution Views** node, multi-select the **auger** and **pellets** nodes and set *Time Step* to `1000`.
9. Place the scene and the plot side by side, and use the mouse to move, or "brush", the data focus box around the plot. You can also adjust the height and width of the box. In the scene, observe the locations of different sizes of pellets.



10. To find out whether the different sizes of pellets stay mixed:
    a) In the plot display, adjust the size and position of the data focus box so that it covers the top left quarter of the data in the heatmap. This action colors the thicker, shorter pellets in the scene.
    b) Look at the colored pellets in the scene. They are generally distributed evenly among the other (white and gray) pellets, so they are well mixed.
    c) Repeat this process to check the thinner, longer pellets. Move the data focus box so that it covers the bottom right quarter of the heatmap data.
    d) Look at the pellets that are colored in the scene now. They are also evenly distributed.

    The different sizes of pellets generally stay mixed as they move through the auger.

Summary

This tutorial demonstrated simulation history with Lagrangian DEM, as well as other advanced post-processing features in STAR-CCM+.

The features that were demonstrated include:

- Creating a solution history file.
- Creating a recorded solution View.
- Viewing recorded solution data in a scene and recording an animation.
- Examining the mixture of DEM particles with data focus.

You now have answers to the questions that were posed at the beginning of this tutorial:

- How fast do the pellets move through this auger? They pass through the auger in approximately 1.4 seconds.
- Do the different sizes stay mixed? Generally, yes.

Design Manager

Enter a short description of your concept here (optional).

This is the start of your concept.

Contents:

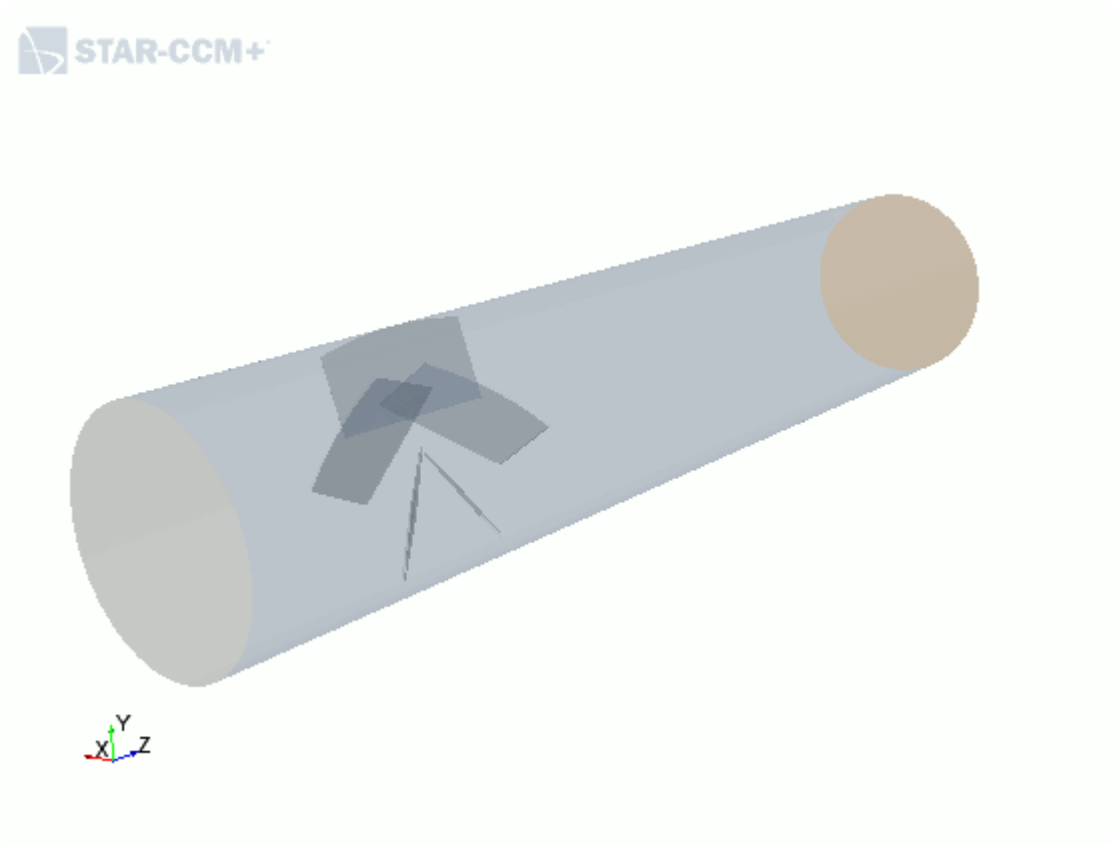
Optimizing a Static Mixer
Design Sweep of a Static Mixer

Optimizing a Static Mixer

One of the study types that Design Manager supports is the multiple-objective tradeoff optimization. In this type of study, Design Manager searches for the combination of inputs that gives the best overall result among competing objectives.

This tutorial requires a STAR-Innovate license in order to run the optimization study. You do not require a STAR-Innovate license to set up or post-process a Design Manager project.

For this tutorial, a mixture of Oxygen and Nitrogen is passing through a static mixer. The aim of the optimization is to make the mixture as homogeneous as possible at the outlet, as well as minimizing the pressure drop across the pipe.



Design Manager can change the following parameters on the geometry:

- The number of fins
- The thickness of the fins
- The width of the fins
- The depth of the fins
- The angle of the fins with respect to the centerline through the pipe

Contents:

Loading the Reference Simulation and Starting Design Manager
Setting Up the Inputs for the Study
Setting Up the Outputs for the Study
Running the Study
Post-Processing

Loading the Reference Simulation and Starting Design Manager

Create a Design Manager project using the reference simulation.

Before doing anything in Design Manager, you are advised to run the reference simulation to convergence. Doing so, makes sure that Design Manager has a good foundation to work with when making changes. In this tutorial, the simulation file provided is already set up and converges to an acceptable level. In order to reduce the size of the simulation file, and because Design Manager requires the reference simulation to have no existing solution, the solution data is cleared in the simulation file.

Load the reference simulation in STAR-CCM+ and create a Design Manager project:

1. In your workspace, create a folder named `Static Mixer`.
2. Navigate to `[INSTALL_DIR]/doc/startutorialsdata/designManager/data`.
3. Copy `staticMixer_reference.sim` to your workspace.
4. Launch STAR-CCM+ and select **File** > **Load...**.
5. In the *Load Simulation* dialog, click **Browse...** and open `staticMixer_reference.sim` from your workspace.
6. Right-click the root node of the simulation tree, **staticMixer_reference**, and select **Create Design Manager Project**.

   The contents of the simulation file are read into the Design Manager study and are ready for use. During this process, Design Manager scans the simulation file for objects that map to inputs and outputs. Design parameters can be used as inputs, while reports, scenes, and plots are outputs.
7. Click **File** > **Save**.
8. In the *Save* dialog, save the project as `staticMixer_optimization.dmprj`

Setting Up the Inputs for the Study

In Design Manager, inputs are parameters that you define in the reference simulation that Design Manager can vary for design exploration studies.

For this tutorial, you allow Design Manager to modify the values of 3D-CAD design parameters. These design parameters are inputs for Design Manager and have a range of acceptable values, where you specify the maximum and minimum values that can be explored.

1. Right-click the **Design Studies** node and select **New**.
2. Rename the **Design Study** node to `Optimization`.
3. Select the **Optimization** study node.

Information about the study, including the type of study and the location of the reference simulation, are displayed in the *Properties* window.

4. In the *Properties* window, set *Study Type* to **Optimization**.

   The icon of the study changes to to reflect the type of study performed.

5. Set the optimization settings:
   a) Select the **Optimization** > **Optimization Settings** node.
   b) Set *Optimization Type* to **Multiple Objective Tradeoff (Pareto)**.

      This optimization type allows for multiple competing objectives. In this tutorial, you want to make the mixture as homogeneous as possible while minimizing the pressure drop across the pipe.

   c) Set *Designs to Run* to `48`.

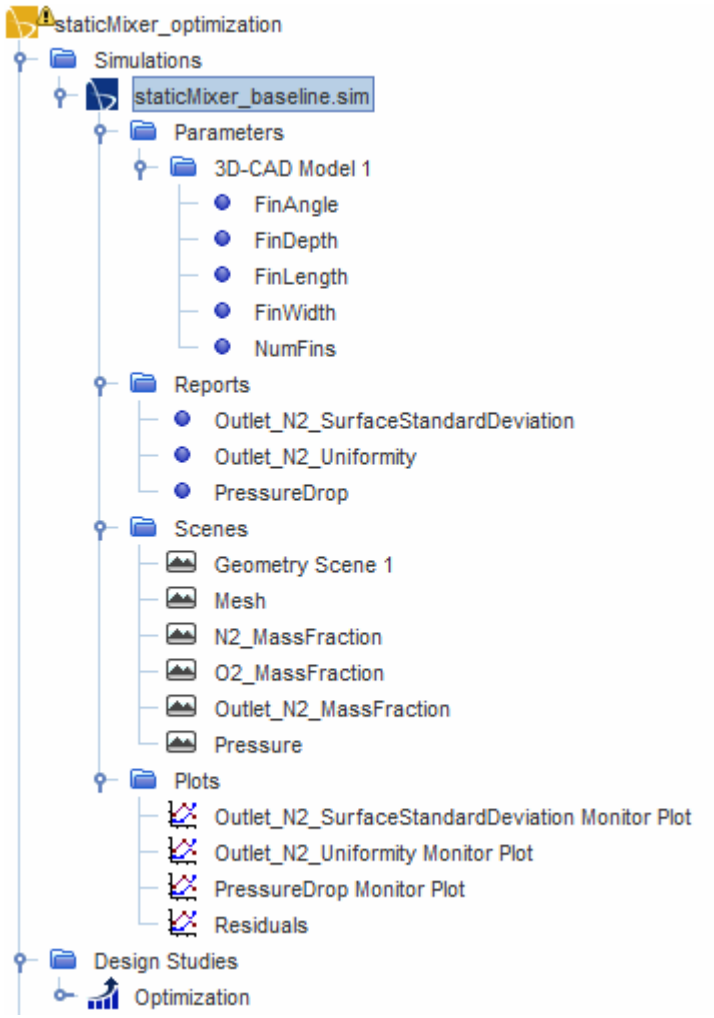
      The recommended number of runs is 8 to 12 times the *Archive Size* (the number of designs that run in a single cycle of an optimization study). For this tutorial, the *Archive Size* is 12 , making the recommended number of runs somewhere between of 96 and 144. To reduce the time it takes to complete this tutorial, you set this value to half of the recommended number of runs.

   d) Select the **Optimization** > **Optimization Settings** > **Archive Size** node and make sure *Number of Designs* is set to `12`.

      This number must be divisible by four and greater than or equal to 12, and greater than or equal to the number of variables in the study.

6. Expand the **Simulations** > **staticMixer_reference.sim** node.

   All the objects that are nested under this node can be used as inputs and outputs by Design Manager.



Design parameters are used as inputs, while reports, scenes, and plots, are outputs.

7. Define the input parameters for the study:
   a) Select the **Design Studies** > **Optimization** > **Input Parameters** node and click the property customizer (...) for *Parameters*.
   b) In the *Input Parameters - Parameters* dialog, select all parameters and click **OK**.

      All design parameters from 3D-CAD are added to the study.
   c) Right-click the **Optimization** > **Input Parameters** node, select **Edit...**, and set the following parameters:

Name	Type	Baseline Value	Range	Resolution
FinAngle	**Continuous**	`45 deg`	`[25.0, 65.0] deg`	`101`
FinDepth	**Continuous**	`0.5 m`	`[0.2, 0.75] m`	`101`
FinLength	**Continuous**	`1.0 m`	`[0.5, 1.5] m`	`101`

Name	Type	Baseline Value	Range	Resolution
FinWidth	**Continuous**	`0.02 m`	`[0.02, 0.2] m`	`101`

For each input parameters, the **Baseline Value** defines the values that Design Manager uses for the first design. The **Range** defines the upper and lower limit that Design Manager can use during the optimization. The **Resolution** defines the maximum number of designs to explore.

d) In the *Optimization - Input Parameters* dialog, set **Type** for the *NumFins* parameter to **Discrete**.

As the number of fins can only be integers, you specify the exact values that Design Manager can use while modifying this parameter.

e) Enter `[3.0, 4.0, 5.0, 6.0]` under **List**.

You want Design Manager to look at designs containing 3 to 6 fins.

f) Make sure that **Baseline Value** for *NumFins* is set to **4**.

g) Click **Close** to close the *Optimization - Input Parameters* dialog.

Setting Up the Outputs for the Study

In Design Manager, responses represent the outputs of your study. In the reference simulation, you monitor the quantities of interest using reports. In Design Manager, you monitor the response of these reports to different designs, using responses.

You can use responses to define study objectives and to set constraints in the optimization study. In this tutorial, you set the pressure drop and mixture homogeneity as competing goals, where the goal for each is to minimize the values. The mixture homogeneity is also added as a constraint, where values above 0.15 are not considered valid.

Scenes and plot data are in a study are also considered as outputs. In this tutorial, you want Design Manager to save the scenes that show the mass fraction of each mixture inside the pipe, as well as the mass fraction at the outlet. In addition to the scenes, you want Design Manager to save the plots containing the residuals and the pressure drop, which are useful to assess solution convergence.

1. Define the responses for the study:

a) Select the **Design Studies** > **Optimization** > **Responses** node and click the property customizer (...) for *Responses*.

b) In the *Responses - Responses* dialog, select **Outlet_N2_SurfaceStandardDeviation** and **PressureDrop**, and click **OK**.

The corresponding reports from the reference simulation are added as responses to the study.

c) Expand the **Responses** node.

d) Edit the **PressureDrop** node and set the following properties:

Parameter	Setting
Is Objective	**Activated**
Objective Properties	
└ **Goal**	**Minimum** (default)

Parameter	Setting
└ **Normalization Type**	**Baseline Normalization** (default)

e) Click **Close**.

f) Edit the **Outlet_N2_SurfaceStandardDeviation** node and set the following properties:

Parameter	Setting
Is Objective	**Activated**
Is Constraint	**Activated**
Objective Properties	
└ **Goal**	**Minimum** (default)
└ **Normalization Type**	**Baseline Normalization** (default)
└ **Weight**	`5.0`
Constraint Properties	
└ **Type**	**Maximum**
└ **Maximum Constraint** > **Maximum**	`0.15`

In optimization studies with multiple objectives, Design Manager prioritizes objectives based on the specified *Weight*. In this study, the weight assigned to reducing the Nitrogen standard deviation (making the mixture more homogeneous) is 5 times the weight assigned to the reducing the pressure drop objective. As a result, Design Manager prioritizes on making the mixture more homogeneous while optimizing the design of the static mixer.

The **Maximum Constraint** property tells Design Manager that designs that return values of **Outlet_N2_SurfaceStandardDeviation** above 0.15 are not valid.

g) Click **Close**.

The **Objective Properties** and **Constraint Properties** nodes are added to the **Outlet_N2_SurfaceStandardDeviation** node. These nodes control the specific objective and constraint properties for the **Outlet_N2_SurfaceStandardDeviation** response.

2. Select the data that you want to post-process:

a) Select the **Design Studies** > **Optimization** > **Scenes** node and click the property customizer (...) for *Scenes*.

b) In the *Scenes* dialog, select the following scenes and click **OK**:

Scene	Description
O2_MassFraction	Displays the mass fraction of Oxygen in the pipe. The isosurface in the scene is colored to show the pressure values in the pipe. The fins in the pipe are also displayed.
N2_MassFraction	Displays the mass fraction of Nitrogen in the pipe.

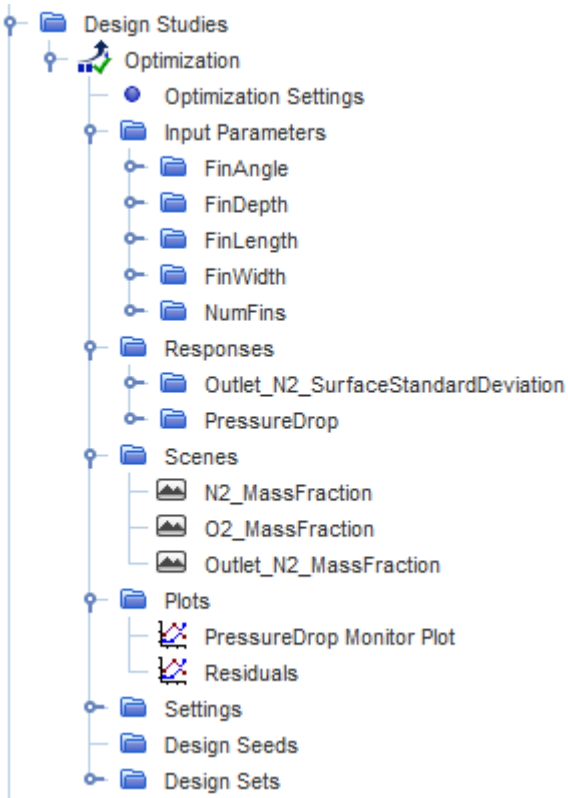
Scene	Description
Outlet_N2_MassFraction	Displays the mass fraction of Nitrogen at the outlet.

The selected scenes from the reference simulation are added as scenes to the study.

c) Select the **Design Studies** > **Optimization** > **Plots** node and click the property customizer (...) for *Plots*.

d) Select the **PressureDrop Monitor Plot** and **Residuals** plots.

The inputs and outputs for the optimization study are shown below:



3. Save the Design Manager project.

Running the Study

Specify the run settings for the study.

To reduce the time it takes to complete the study, run multiple jobs at the same time.

1. Expand the **Design Studies** > **Settings** node.
2. Select the **Run Settings** node and set the following settings:

Parameter	Setting
Save Simulation Files	**None**

Parameter	Setting
Simultaneous Jobs	`6`
Compute Processes	`1` (default)

3. Set up a history plot to show the path that Design Manager follows to reach the best design in the study:
   a) Right-click the **Design Studies** > **Optimization** node and select **Create Plot** > **History Plot**.
      The history plot opens and an **XY Plot** is added to the **Plots** node of the project.
   b) Rename the **XY Plot** to `History Plot`.
4. Right-click the **Design Studies** > **Optimization** node and select **Run Study**.

All necessary files for the study are created automatically.

When building completes, the jobs are submitted in the background. The STAR-CCM+ servers for each job are visible in the server list.

The *Output Table* window appears and displays a list of designs being investigated and the specific settings of the input parameters. In this window you can keep track of the progress of each design in real time.

Output Table - All

Design#	Name	State	Performance	FinAngle	FinDepth	FinLength	FinWidth	NumFins	Outlet_N2_SurfaceStandardDeviation	PressureDrop
1	Design 1			45.0 deg	0.5 m	1.0 m	0.02 m	4.0		
2	Design 2			32.2 deg	0.299000000...	1.319999999...	0.1676 m	4.0		
3	Design 3			43.2 deg	0.700500000...	0.865 m	0.183800000...	2.0		
4	Design 4			36.0 deg	0.49975 m	0.59 m	0.0695 m	1.0		
5	Design 5			61.4 deg	0.2495 m	0.775 m	0.0524 m	5.0		
6	Design 6			50.4 deg	0.75 m	1.225 m	0.0362 m	4.0		
7	Design 7			57.8 deg	0.54925 m	1.045 m	0.2 m	6.0		
8	Design 8			54.0 deg	0.400750000...	0.5 m	0.1505 m	3.0		
9	Design 9			46.8 deg	0.2 m	1.135 m	0.1343 m	1.0		
10	Design 10			65.0 deg	0.59875 m	1.410000000...	0.101900000...	2.0		
11	Design 11			28.6 deg	0.651 m	0.679999999...	0.118100000...	5.0		
12	Design 12			25.0 deg	0.351250000...	0.955000000...	0.02 m	3.0		

For more information on the various states of the designs, see Monitoring a Design Study.

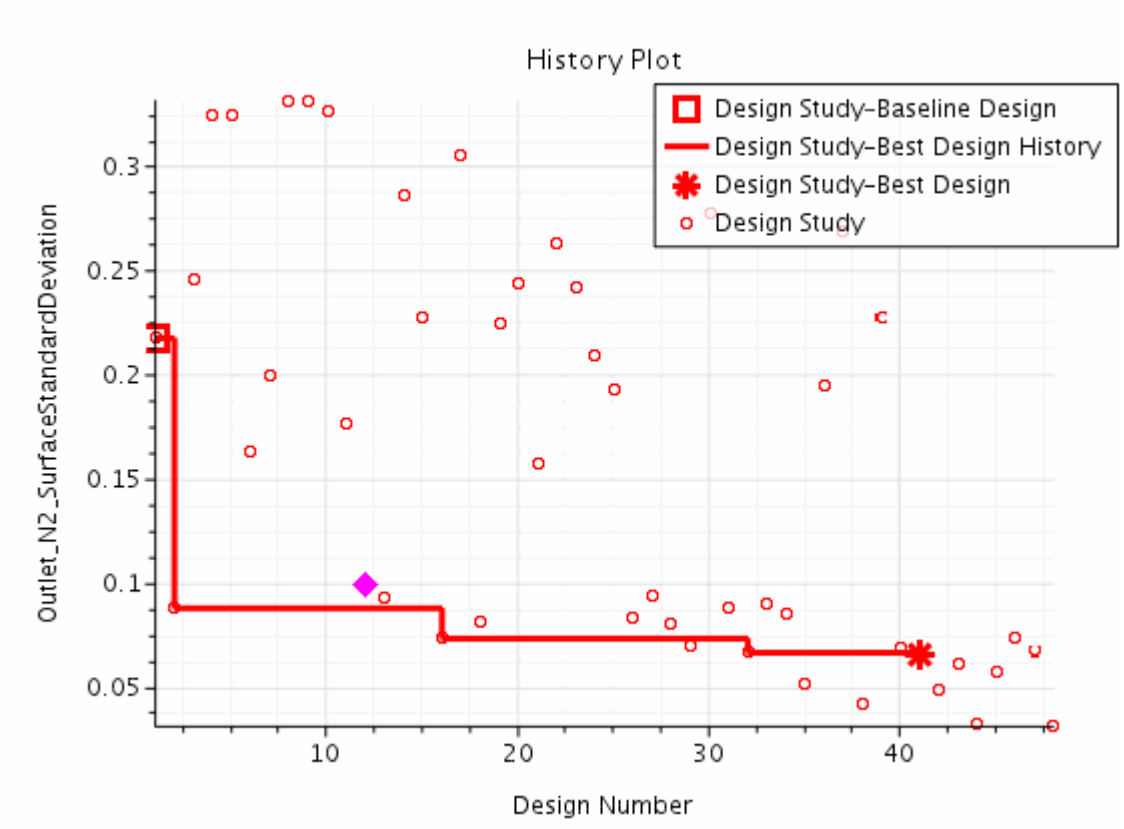
The study is divided into cycles, in this case 4 cycles with 12 designs in each cycle. Design Manager runs a single cycle at a time, and waits until all designs in that particular cycle are done before moving on to the next cycle. The reason for this is that Design Manager uses the SHERPA algorithm when running an optimization study, and the SHERPA algorithm performs better the more cycles that you run. For optimization studies with multiple competing objectives, the MO-SHERPA algorithm is used.

When the study completes, the best design is identified and mentioned in the *Output* window.

```
Design Study : The best design in cycle #4 is Design 42
Design Study : No more designs to execute..

 ---> The best design in Design Study is Design 42 <---
```

```
************************************************************

                Study Name : Design Study

              --- EXECUTION COMPLETED ---

************************************************************
```

The history plot shows the best design for each cycle chosen by Design Manager.



Post-Processing

Plot the Pareto front and assess how the input parameters affect the responses.

The results that you obtain in your optimization study can differ to the result displayed below. When performing optimization studies, Design Manager uses a random seed point and assesses each design in a random manner, steering the solution towards the optimized design. Results that are displayed below are for illustrative purposes.

1. Open the scenes in the Design Manager environment:
   a) In the *Output Table* window, right-click **Design 1** and select **Open Scenes** > **O2_MassFraction**.

      The *O2_MassFraction* scalar scene is loaded into the *Graphics* window. The *O2_MassFraction* scene is added to the **Snapshots** node. All open scenes and plots appear under this node.

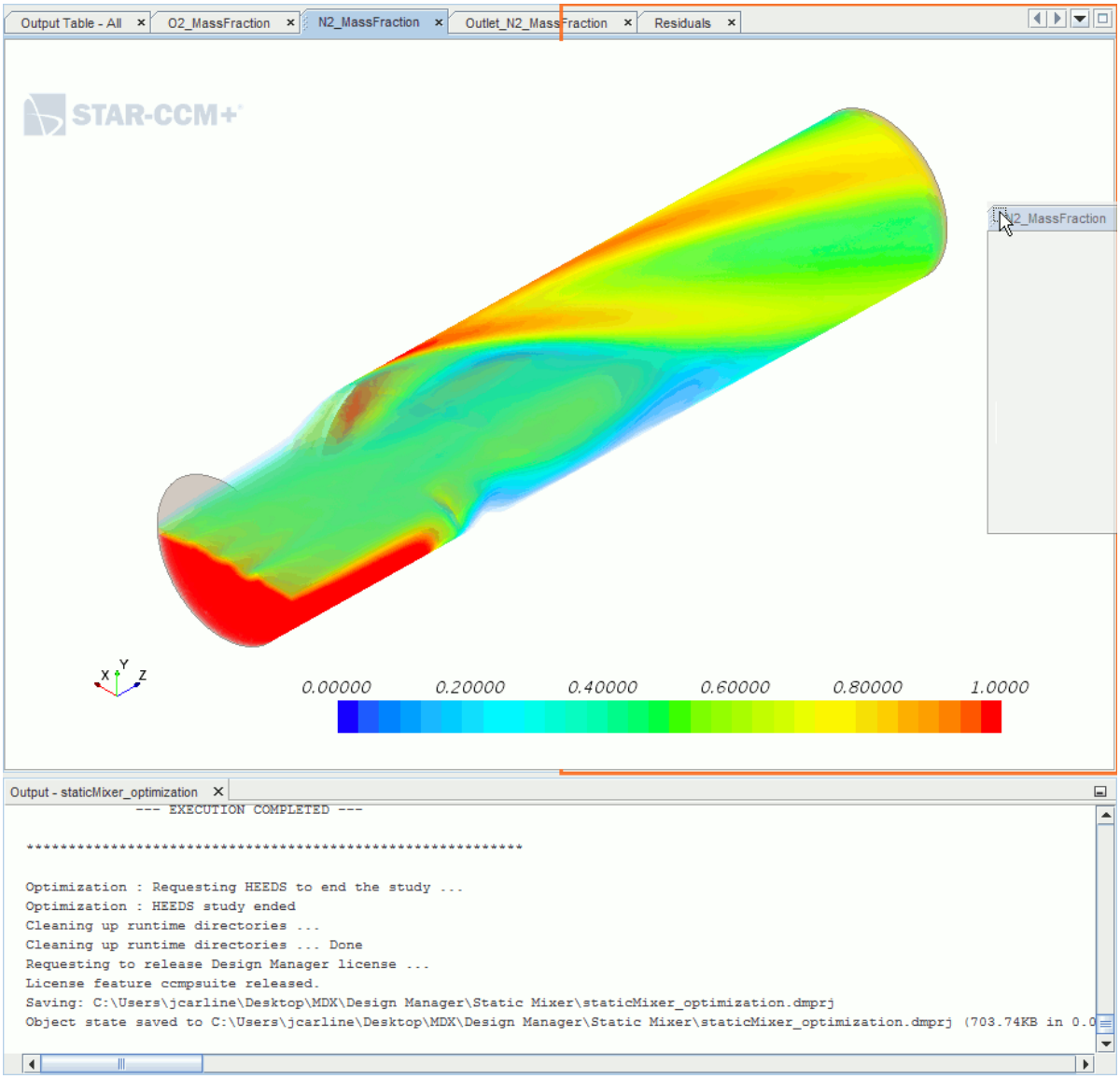
b) In the same way, open the **Outlet_N2_MassFraction** and **N2_MassFraction** scenes.

c) In the *Output Table* window, right-click **Design 1** and select **Open Plots** > **Residuals**.

2. Rearrange the Design Manager workspace to display multiple scenes and plots:

a) Click and drag the tab of the *N2_MassFraction* scene to the right until the scene pops out and splits the workspace into two, then release the mouse button.

A red box shows a preview of what section of the workspace will be split.



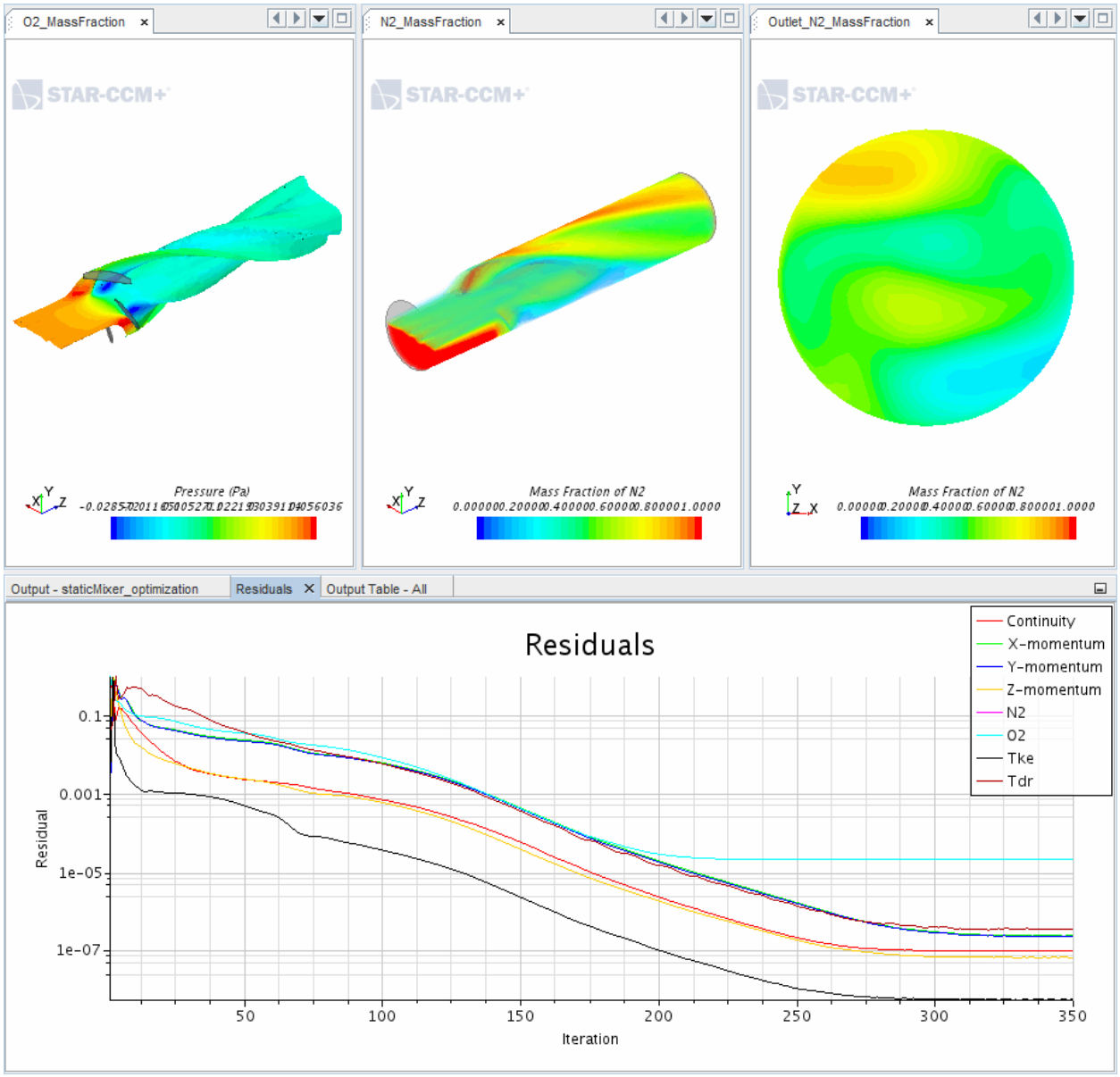
b) Repeat the above step for the *Outlet_N2_MassFraction* scene and drag it so that it appears to the right of the *N2_MassFraction* scene.

c) In the **Snapshots** node, multi-select the **Outlet_N2_MassFraction** and **N2_MassFraction** scenes, right-click a selection, and activate the **Linked View** option.

This option links the two scenes together, so that the geometry inside each scene is orientated the same way. Rotating the geometry in one scene automatically causes the geometry in the linked scene to do the same. The zoom levels in the linked scenes are also matched.

d) Drag the *Residuals* plot to the bottom half of the Design Manager workspace.

The final workspace is shown below.



3. Some designs can fail and are marked with a icon. If you have failed designs, you can filter the output table to only display the successful designs:
   a) Expand the **Design Studies** > **Optimization** > **Design Sets** node.
   b) Double-click the **Successful** node.

      An *Output Table* opens and only displays the successful designs.
   c) Close the *Output Table - Successful* window.
4. Use the *Design Manager Snapshot* toolbar, , to cycle through the different designs.
5. In this optimization study, you have two competing objectives. For such studies, a Pareto front is useful for visualizing how each design performs compared to the other designs.

a) Right-click the **Design Studies** > **Optimization** node and select **Create Plot** > **Response vs Response Plot**.

A plot, **XY Plot**, is added to the **Plots** node.

b) Rename **XY Plot** to `Pareto Front`.

c) Select the **Plots** > **Pareto Front** node and set *Title* to `Pareto Front`.

d) Select the **Data Series** > **Optimization** node and set *Input Design Sets* to **Successful**.

This option only shows the successful designs in the plot.

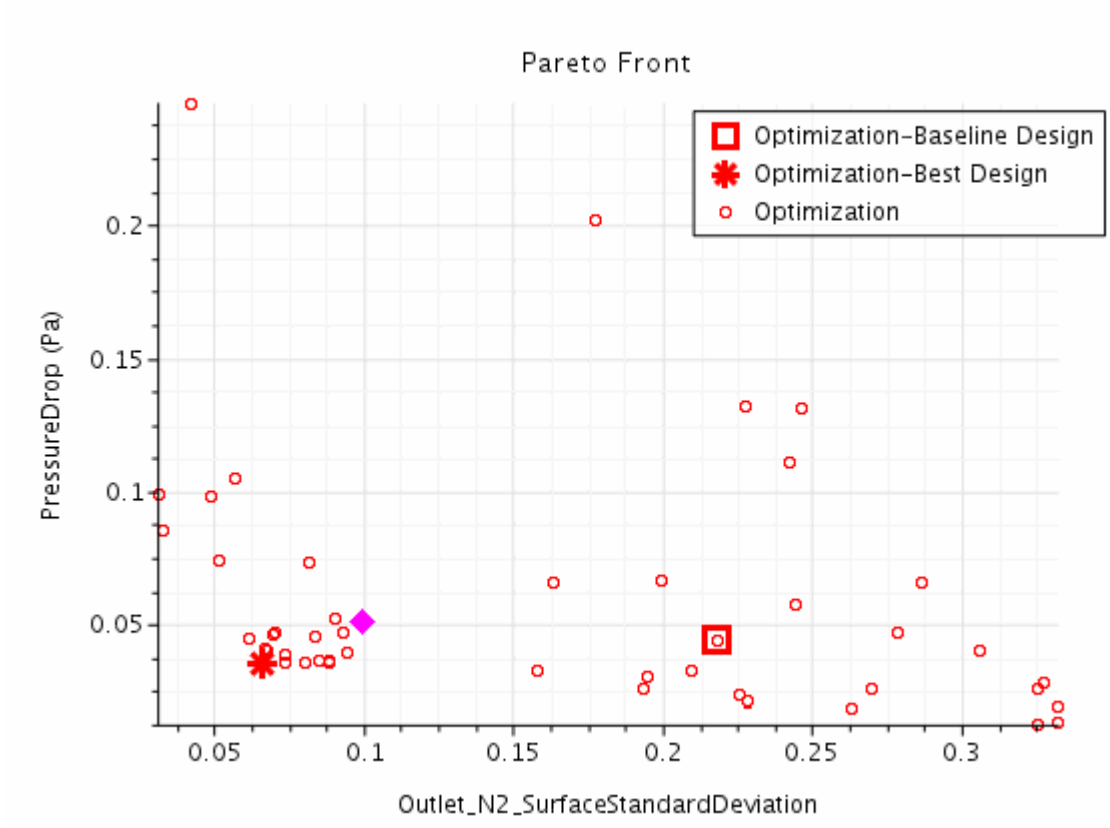
e) Select the **Data Series** > **Optimization** > **Best Design** node and activate the **Visible** option.

The best design is marked on the plot with an asterisk symbol.

f) Select the **Data Series** > **Optimization** > **Baseline Design** node and activate the **Visible** option.

The baseline design is marked on the plot with an empty square symbol.

The final Pareto front plot appears as follows. Note that your results may be different due to the random approach to the solution by Design Manager.



6. You can interact with the plot to explore the designs:

a) Move the mouse pointer around the plot.

The corresponding X and Y-axis values are displayed for each data point on the plot as you hover over them.

b) Click on a data point in the plot.

The scenes that you set up change to display the selected design.

7. Save the Design Manager project.

Design Sweep of a Static Mixer

Design Manager provides an automated method for evaluating the performance of a single product design.

In this tutorial, you obtain the performance data for an optimized design of a static mixer. You set up a global parameter in STAR-CCM+ that controls the mass flow rate at the inlet of the static mixer. Design Manager runs the solver for the same design using a range of mass flow rates.

The optimized design for the static mixer is obtained as part of the tutorial, Optimizing a Static Mixer. If you have a STAR-Innovate license, you are advised to complete that tutorial first. See Optimization of a Static Mixer. If you do not have a STAR-Innovate license, the optimized design is provided for you in a simulation file.

Contents:

Configuring the Reference Simulation and Starting Design Manager
Setting Up the Study
Running the Study and Reviewing the Performance Data

Configuring the Reference Simulation and Starting Design Manager

After configuring the chosen design from the Optimization tutorial, you create a Design Manager project.

Before you open Design Manager, you must set up the reference simulation so that Design Manager can modify the mass flow rate. Design Manager does not set properties on simulation objects directly. Rather, you create global parameters that are reference by object properties. In this tutorial, you create a global parameter that drives the mass flow rate at the inlet.

In order to reduce the size of the simulation file, and because Design Manager requires the reference simulation to have no existing solution, the solution data is cleared in the simulation file provided.

To configure the reference simulation in STAR-CCM+:

1. In your workspace, create a folder named `Static Mixer`.
2. Navigate to `[INSTALL_DIR]/doc/startutorialsdata/designManager/data` and copy `staticMixer_optimized.sim` to your workspace.
3. Launch STAR-CCM+.
4. Select **File** > **Load...**.
5. In the *Load Simulation* dialog, click **Browse...** and open `staticMixer_optimized.sim` from your workspace.
6. Create a global parameter that controls the mass flow rate on the inlet boundary:
   a) In the simulation tree, right-click the **Tools** > **Parameters** node and select **New** > **Scalar**.
   b) Rename the **Scalar** node to `Mass Flow Rate`.
   c) In the node properties for the **Mass Flow Rate**, click the property customizer (...) for *Dimensions*.
   d) In the *Mass Flow Rate - Dimensions* dialog, enter the following:

Parameter	Setting
Mass	`1`
Time	`-1`

These settings return the units of mass flow rate, `Kg/s`.

e) Click **OK**.

f) For the same node, set *Value* to `1.0 Kg/s`.

7. Use the global parameter to define the inlet mass flow rate:

   a) Expand the **Regions** > **Mixer** > **Boundaries** > **Inlet** node.

   b) Select the **Physics Values** > **Mass Flow Rate**, click the customizer (...) for *Value*.

   c) In the *Mass Flow Rate - Value* dialog, set **Definition** to `${Mass Flow Rate}` and click **OK**.

8. Save the simulation.

Launch Design Manager from the reference simulation:

9. Right-click the root node of the simulation tree and select **Create Design Manager Project**.

   The contents of the simulation file are read into a Design Manager study and are ready for use. During this process, Design Manager scans the simulation file for objects that map to inputs and outputs. Design parameters can be used as inputs, while reports, scenes, and plots are outputs.

10. Click **File** > **Save**.

11. In the *Save* dialog, save the project as `staticMixer_sweep.dmprj`

Setting Up the Study

After loading the reference simulation into Design Manager, you proceed to set input and output parameters.

For this tutorial, you configure Design Manager to modify the values of the mass flow inlet in order to obtain a performance curve.

For the outputs, you configure Design Manager to save the scenes that show the mass fraction of each mixture inside the pipe, as well as the mass fraction at the outlet. In addition to the scenes, you configure Design Manager to save the plots containing the residuals and the pressure drop, as these plots are useful for assessing solution convergence.

1. Right-click the **Design Studies** node and select **New**.

2. Rename the **Design Study** node to `Sweep`.

3. Select the **Sweep** study node.

   Information about the study, including the type of study and the location of the baseline simulation, is displayed in the *Properties* window.

4. In the *Properties* window, set *Study Type* to **Sweep**.

   The icon next to the study reflects the type of study performed.

5. Expand the **Simulations** > **staticMixer_baseline.sim** node.

   All the objects that are nested under this node can be used as inputs and outputs by Design Manager.

6. Define the input parameters for the study:
   a) Select the **Design Studies** > **Sweep** > **Input Parameters** node and click the property customizer (...) for *Parameters*.
   b) In the *Input Parameters* dialog, select **Simulation Parameters** > **Mass Flow Rate** and click **OK**.

      The **Mass Flow Rate** simulation parameter is added to the study.
   c) Select the **Sweep** > **Input Parameters** > **Mass Flow Rate** > **Continuous** node and set the following parameters:

Parameter	Setting
Range	`[0.1, 3.5] kg/s`
Resolution	`10`

      The **Range** defines the upper and lower limit for each design parameter that Design Manager can control during the optimization. The **Resolution** defines the number of different values to which mass flow rate is set. When you set **Resolution**, the value of **Increment** is calculated automatically.
7. Define the responses for the study:
   a) Select the **Design Studies** > **Sweep** > **Responses** node and click the property customizer (...) for *Response*.
   b) In the *Responses* dialog, select the **PressureDrop** response, and click **OK**.

      The **PressureDrop** report from the reference simulation is added as a response to the study.
8. Select the data that you want to post-process:
   a) Select the **Design Studies** > **Sweep** > **Scenes** node and click the property customizer (...) for *Scenes*.
   b) In the *Scenes* dialog, select the following scenes and click **OK**:

Scene	Description
O2_MassFraction	Displays the mass fraction of Oxygen in the pipe. The isosurface in the scene is colored to show the pressure values in the pipe. The fins in the pipe are also displayed.
N2_MassFraction	Displays the mass fraction of Nitrogen along the length of the pipe.
Outlet_N2_MassFraction	Displays the mass fraction of Nitrogen at the outlet.

      The selected scenes from the baseline simulation are added as scenes to the study.
   c) Select the **Design Studies** > **Sweep** > **Plots** node and click the property customizer (...) for *Plots*.
   d) Select the **PressureDrop Monitor Plot** and **Residuals** plots.
9. Set up a plot to show pressure drop vs. mass flow rate:
   a) Right-click the **Design Studies** > **Sweep** node and select **Create Plot** > **Parameter vs Response Plot**.

A plot scene opens and an **XY Plot** is added to the **Plots** node of the project.

b) Rename the **XY Plot** to `Performance Plot`.

c) Select the **Performance Plot** node and set *Title* to `Performance`.

d) Select the **Performance Plot** > **Data Series** > **Line Style** node and set *Style* to **Solid**.

10. Save the Design Manager project.

Running the Study and Reviewing the Performance Data

Specify the run settings for the study.

To reduce the time it takes to complete the study, run multiple jobs at the same time.

1. Expand the **Design Studies** > **Settings** node.
2. Select the **Run Settings** node and set the following settings:

Parameter	Setting
Save Simulation Files	**None**
Simultaneous Jobs	`5`
Compute Processes	`1`

3. Right-click the **Design Studies** > **Sweep** node and select **Run Study**.

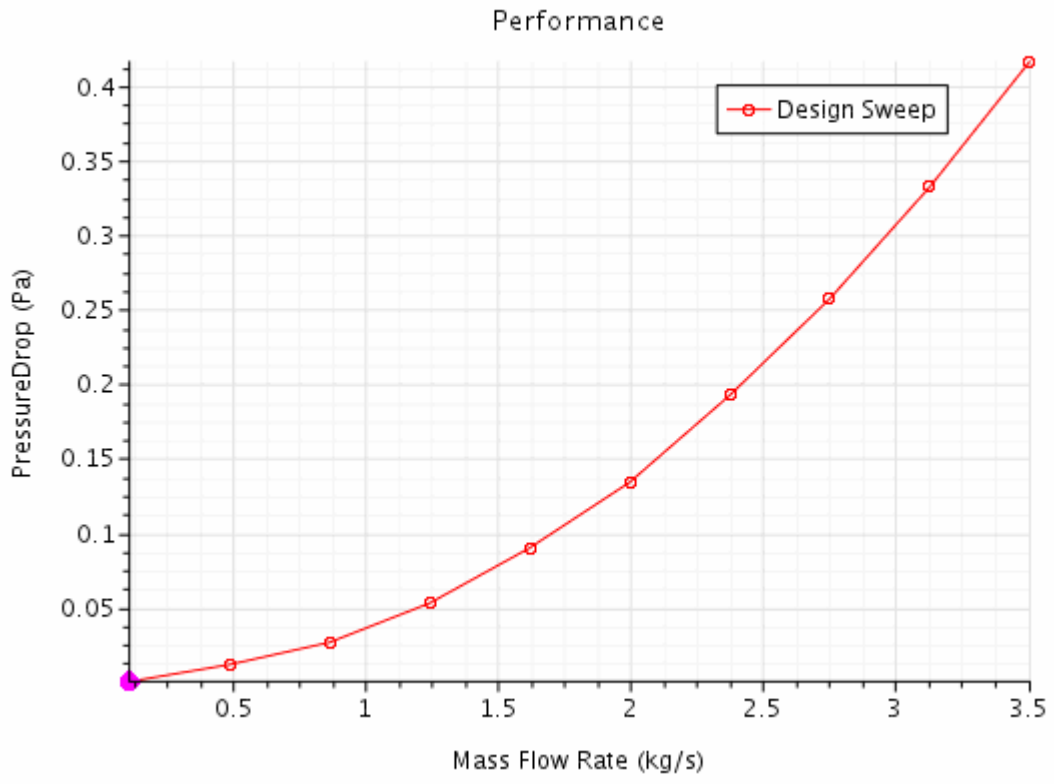
Design Manager creates all files that are necessary for running the performance sweep.

When building completes, Design Manager schedules each simulation for running. Depending on the number of simultaneous jobs that you set in the **Run Settings**, multiple jobs run at the same time. The STAR-CCM+ servers for each job are visible in the server list.

The *Output Table* window appears and displays a list of designs being investigated and the specific settings of the input parameters. In this window, you can monitor the progress of each design in real time.

For more information on the various states of the designs, see Monitoring a Design Study.

The performance plot updates as the optimized design is evaluated with different mass flows at the inlet. The final plot is shown as follows:


Performance
Design Sweep
PressureDrop (Pa)
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0.5
1
1.5
2
2.5
3
3.5
Mass Flow Rate (kg/s)

STAR-ICE

The tutorials in this set illustrate features for simulating internal combustion engines in STAR-CCM+ using the dedicated add-on STAR-ICE.

Contents:

Gasoline Engine

Gasoline Engine

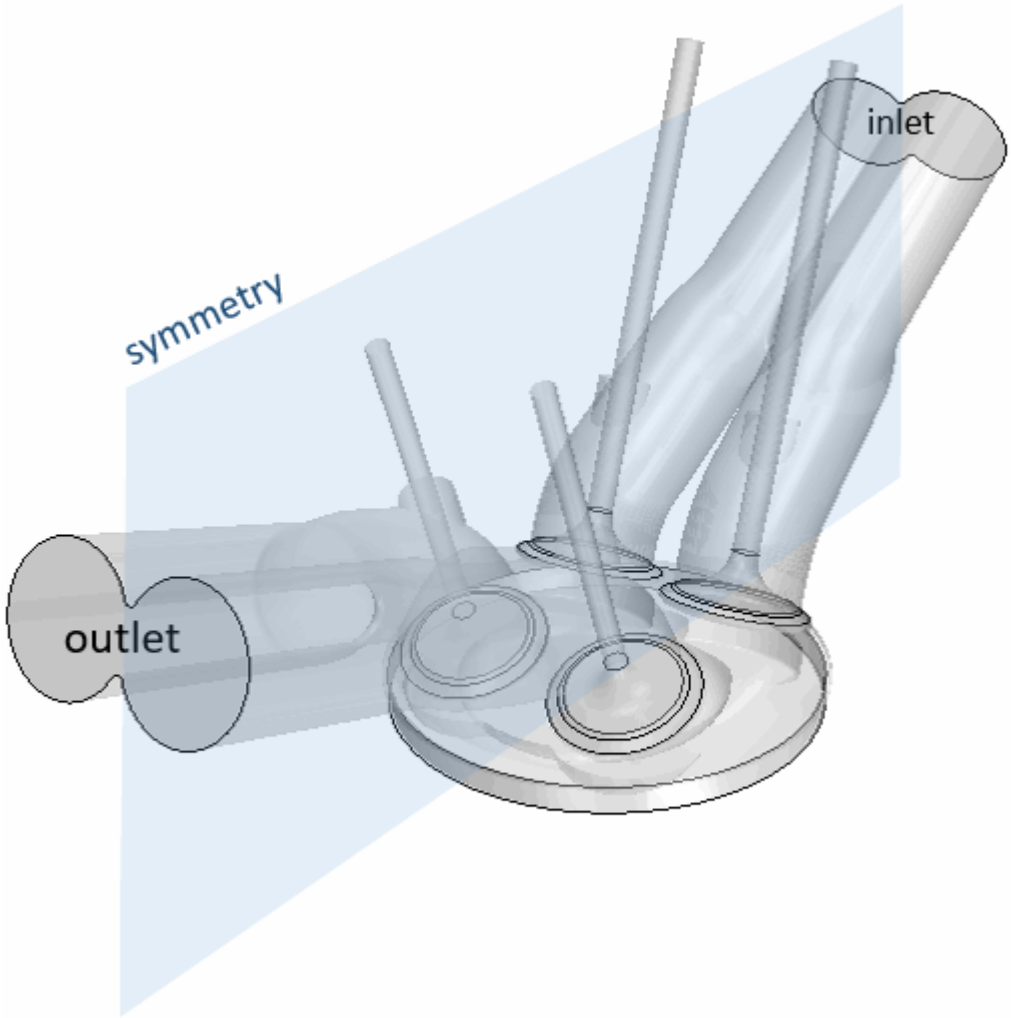
STAR-ICE provides an automated method to perform transient moving-mesh analyses of internal combustion engines.

The purpose of this tutorial is to predict the progress of tumble and turbulence kinetic energy in a single cylinder until the ignition point. In general, a significant tumble inside the cylinder during the intake stroke generates high turbulence at the end of the compression stroke, which results in fast burning rates.

You simulate the intake and compression stroke in a gasoline engine that runs at 3600 RPM. The simulation starts at 300 deg crank angle. You initialize the cylinder and the ports using tabular data that describe the temperature and absolute pressure values as functions of crank angle. STAR-ICE automatically evaluates the initial values at the respective start angle.

Nitrogen enters through the intake port inlet and exits through the exhaust port outlet, where both boundaries are modeled as pressure outlets. Similar to initialization, you specify the boundary conditions in the form of tables that describe the absolute pressure and temperature values as functions of crank angle.

The geometry consists of a four-valve cylinder, where the piston is in the TDC position and the valves are closed. The setup allows you to use symmetry conditions and include only half the geometry, thus reducing the computational cost of the simulation.



You model the motion of the piston and the valves:

- The piston moves in z-direction. To specify the piston motion, you set the connecting rod length to 145 mm. For the bore, you specify a value of 80 mm.
- For the motion of the valves, you import tables that describe the valve lifts as functions of crank angle. To reflect the play resulting from loose connections between the rocker arm and the valve stem, you define a global parameter for the valve lash and set its value to 0.5 mm.

The simulation stops at 720 deg crank angle. You investigate the simulation results using the post-processing capabilities that are available in STAR-ICE.

Contents:

Prerequisites
Importing the Geometry
Defining Engine Solver Settings and Operating Conditions
Controlling the Valve Lash
Creating a Half Engine Model
Setting the Initial Conditions

Setting up the Piston
Setting the Inlet and Outlet Boundary Conditions
Setting Up the Valve Motions
Preparing the Plots and Scenes for Visualization
Running the Simulation
Analyzing the Results
Summary

Prerequisites

This tutorial guides you through the setup of the gasoline engine simulation using the dedicated STAR-ICE add-on in STAR-CCM+.

You require a double-precision version of STAR-CCM+ to complete this tutorial.

Make sure that the (**Open STAR-ICE**) button is visible in the STAR-CCM+ toolbar. For information on how to include the button in the toolbar, see Accessing STAR-ICE.

Importing the Geometry

You start the simulation setup from a CAD geometry in Parasolid format.

Due to the large number of cells, and the time it takes the solver to run, launch STAR-ICE in parallel on at least 8 cores.

To launch STAR-ICE:

1. Start STAR-CCM+ and create a new simulation by clicking **File** > **New...**.
2. In the *Create a File* dialog, select an appropriate parallel option and set **Compute Processes** to at least `8`.

   For information on the parallel options, see Process Options.
3. Click **OK**.
4. From the STAR-CCM+ toolbar, click (**Open STAR-ICE**).

   A STAR-ICE tab opens within STAR-CCM+.
5. Select **File** > **Save As...** and save the simulation under the name `STAR-ICE_GasolineEngineTutorial.sim`.

To import the geometry:

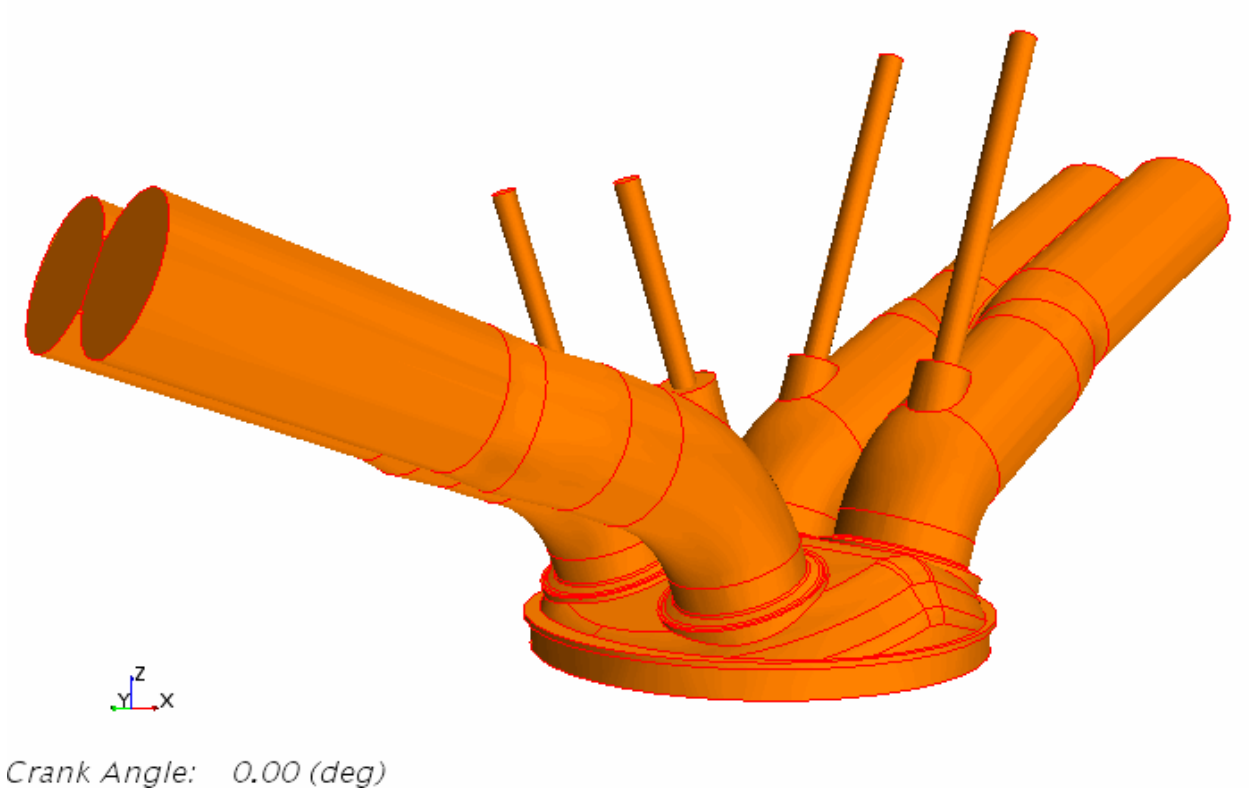
6. Right-click the **Geometry** node and select **Import CAD Model**.
7. In the *Open* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/STAR-ICE/data`.
8. Select `STAR-ICE_GasolineEngineTutorial_Geometry.x_b` and click **Open**.

   A total of five parts are imported:

   - **engine-fluid**
   - **exhaust-valve 1**

- **exhaust-valve 2**
- **intake-valve 1**
- **intake-valve 2**

9. In the *Graphics* window, position the parts to obtain a view similar to the one shown below.



For information on how to change the view in the *Graphics* window, see Mouse Configurations and Changing the View.

10. Save the simulation .

Defining Engine Solver Settings and Operating Conditions

Here, you set the time-step size, the crank angle at which the simulation begins, and the duration of the analysis.

To set the time-step size:

1. Right-click the **Setup** > **Solvers** node and select **Edit**.

   An *Edit* dialog opens that displays the default engine solver settings, auto-save settings, operating conditions, and valve motion.

2. In the *Engine Solver Settings* group-box, activate **Crank Angle Step** and set the value to `0.2 deg`.

To set the crank angle at which the simulation begins and the duration of the analysis:

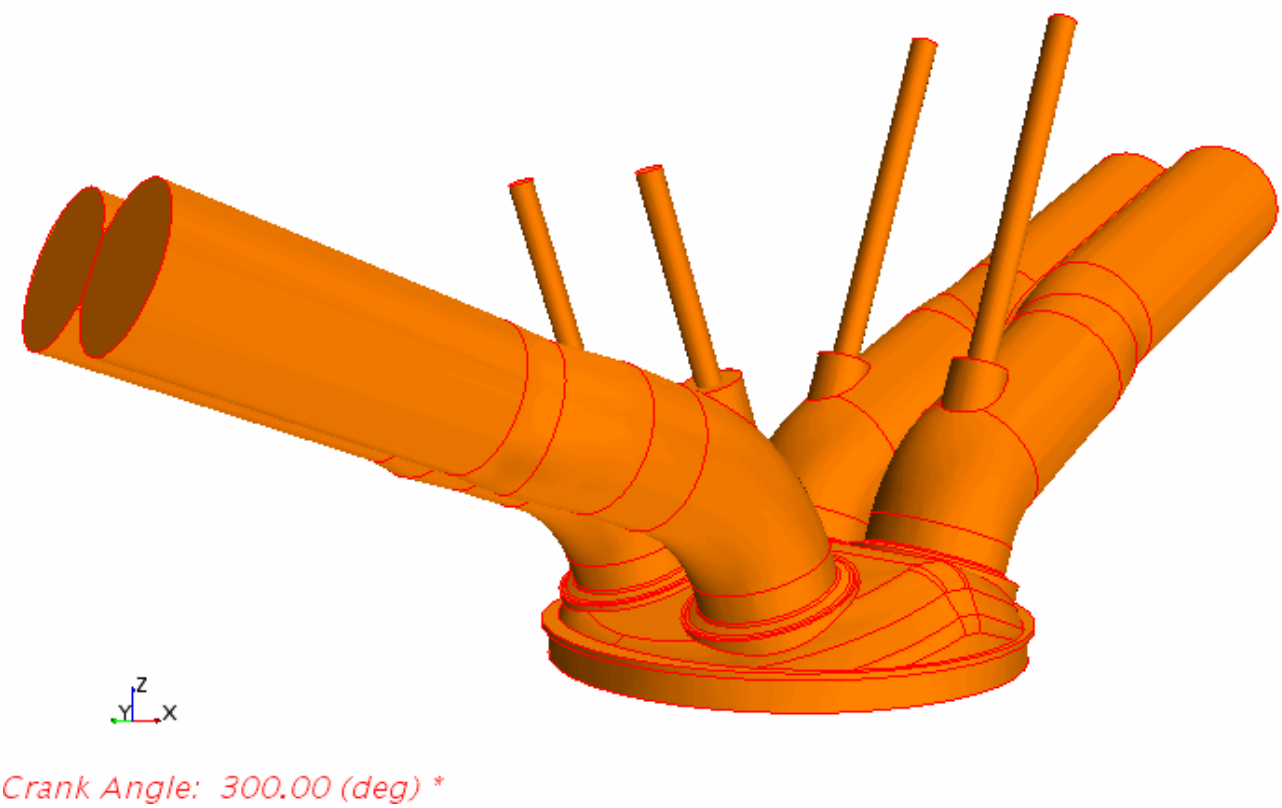
3. In the *Operating Conditions* group-box, set the following properties:

Property	Setting
Start Angle	`300.0 deg`
Duration	`420.0 deg`

In general, you run an analysis over multiple engine cycles to reduce the effect of initial conditions on the final results. If your processing time is limited, you are advised to start the analysis some crank angles before the intake valve opens—for example at 300 deg crank angle. At this stage, you can assume that the flow in the cylinder is relatively steady compared to the intake and combustion phases.

4. Click **Apply**, then **Close**.

The *Graphics* window still displays the engine geometry at TDC position. However, in the bottom left corner, the red-colored starred text indicates that the displayed imported geometry no longer represents your settings of 300 deg crank angle.



5. Save the simulation .

Controlling the Valve Lash

The valve lash adjusts the raw lift data in order to match the real engine conditions. To control the valve lash, you define a global parameter. You define this parameter once and apply it to all valve lift curves.

To add a parameter:

1. Right-click the **Parameters** node and select **Edit**.
2. In the *Edit* dialog, click **Add a new parameter** and set the following properties:

Property	Setting
Parameter	`ValveLash`
Value	`0.5 mm`

3. Click **Add**.

   The new parameter is added to the parameters list.
4. Click **Close**.
5. Save the simulation .

Creating a Half Engine Model

To reduce cell count and computational costs, create a half model.

To create a half model of the engine:

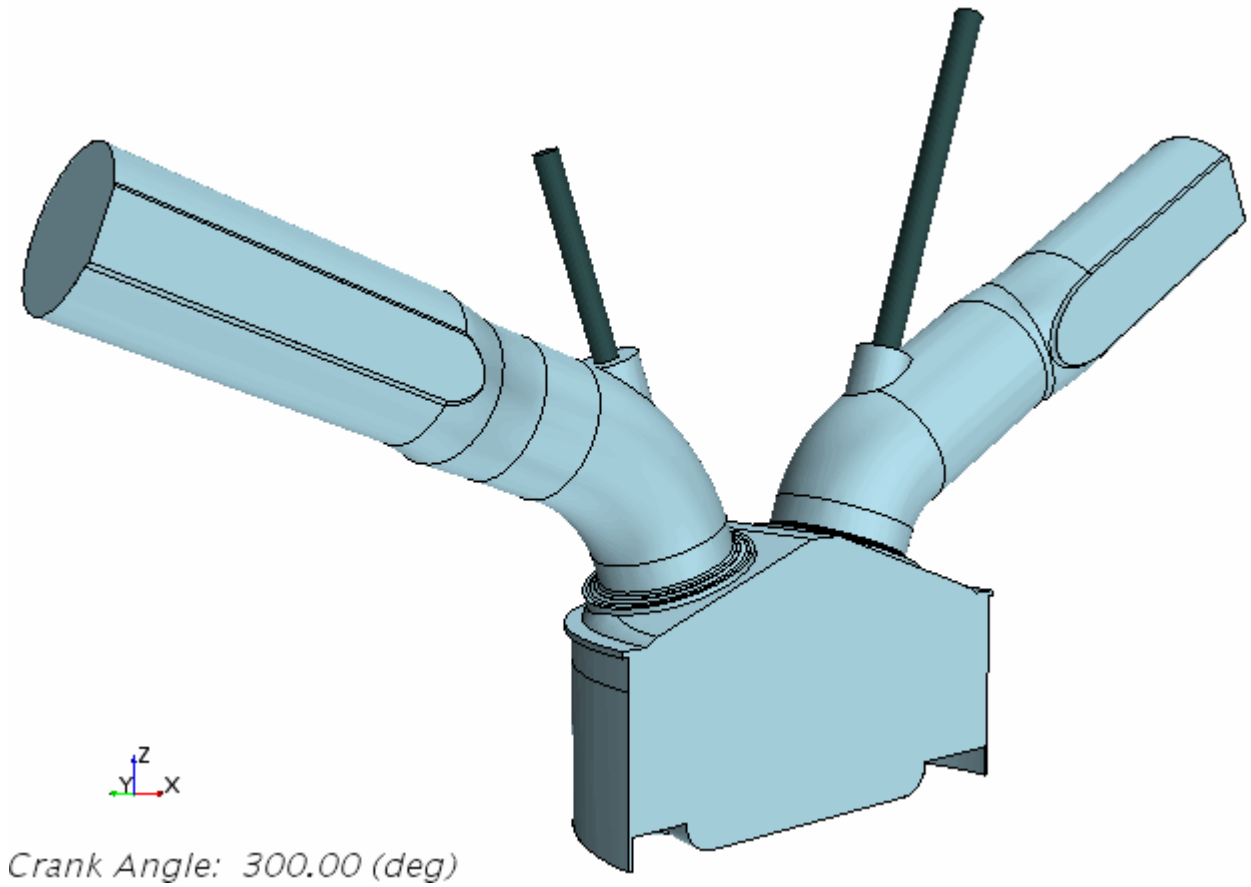
1. Right-click the **Engine** node and select **Create Cylinder and Valves**.
2. In the *Engine Symmetry* group-box, activate **Half Model** and click **OK**.

   STAR-ICE creates three Engine Parts on the +side of the symmetry plane:

   - **Cylinder**
   - **Exhaust Valve 1**
   - **Intake Valve 1**

   These Engine Parts allow you to set the physics for the engine components, such as initial and boundary conditions or motions.

   The *Graphics* window updates to display the half model at 300 deg crank angle:



STAR-ICE automatically applies symmetry conditions at the symmetry plane.

3. Save the simulation .

Setting the Initial Conditions

You specify the initial conditions in the cylinder and ports using tables. A single table describes both absolute pressure and temperature as a function of crank angle. STAR-ICE evaluates the initial values at the specified start angle (300 deg crank angle) using linear interpolation between the data points.

For the cylinder, you import a table that is derived from experimental results. For the ports, as no experimental results are available, you approximate the initial conditions using the inlet and outlet boundary conditions, respectively.

To set initial conditions for the cylinder:

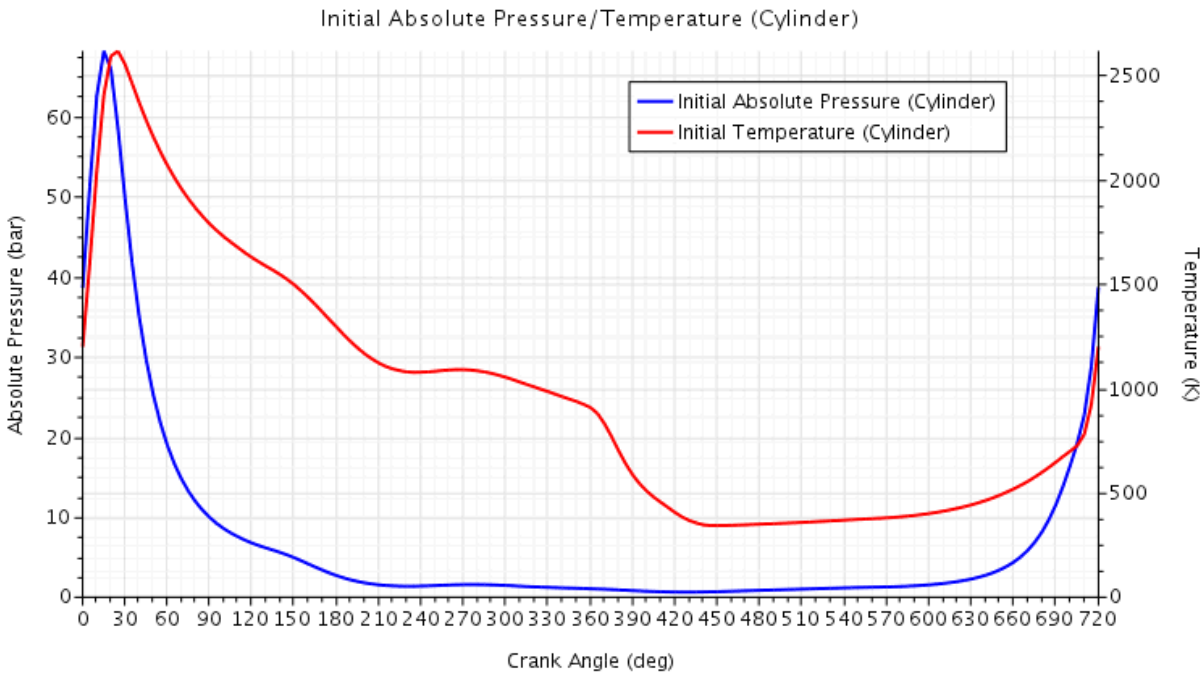
1. Right-click the **Engine** > **Cylinder** node and select *Edit*.
2. Within the *Edit* dialog, in the *Initial Conditions* tab, click **Cylinder**.
3. In the *Cylinder* group box, set the initial **Pressure** using tabular data:
   a) Next to the **File** drop-down menu, click **Import**.
   b) In the *Import Table* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/STAR-ICE/data`.
   c) Select `cylinder.csv` and click **Open**.

In the *Graphics* window, a plot opens that displays the absolute pressure in the cylinder as a function of crank angle. The correct table columns and units for the pressure are assigned automatically.

4. Set the initial **Temperature**:
   a) From the **File** drop-down menu, select the previously imported file **cylinder**.
   b) Set the following table properties:

Property	Setting
Temperature Column	**T_K**
Temperature Units	**K**

The initial conditions for the cylinder are now complete. The plot in the *Graphics* window displays the absolute pressure and temperature curves for the cylinder:



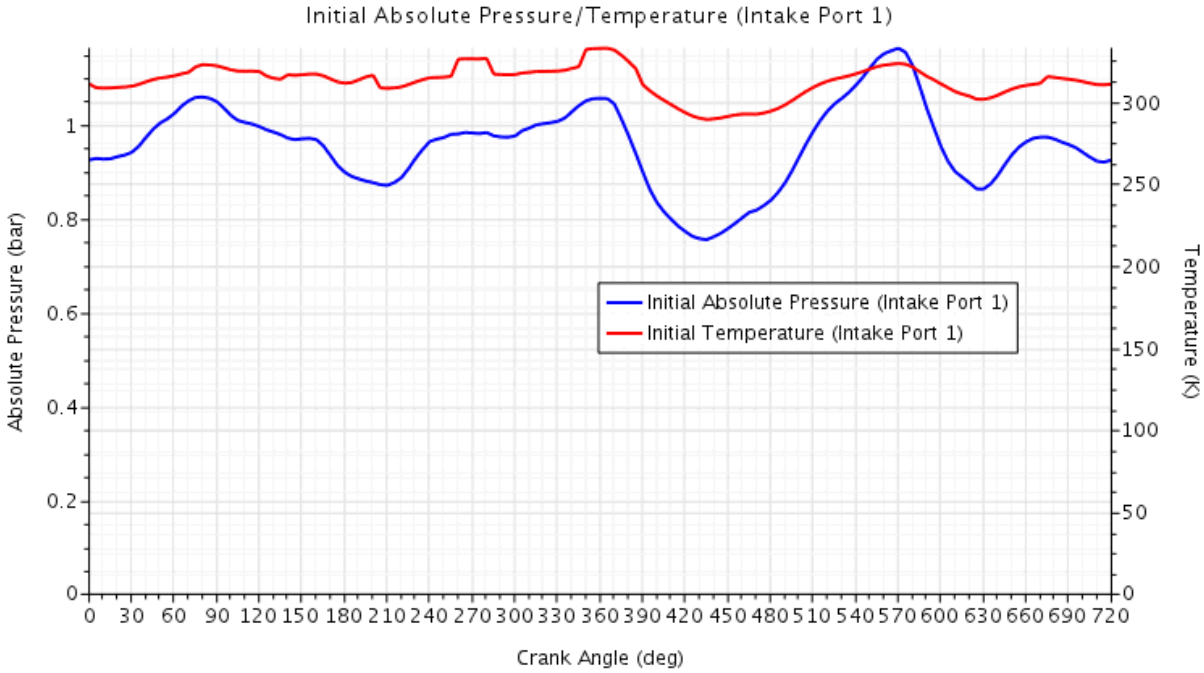
5. To determine the absolute pressure and temperature values that STAR-ICE uses for initialization, hover the mouse over the respective curve and stop at 300 deg crank angle.

   The coordinates are displayed next to the pointer. The initial absolute pressure value is 1.46769 bar and the initial temperature value is 1052.57 K.
6. Click **Cylinder** to minimize the group box.

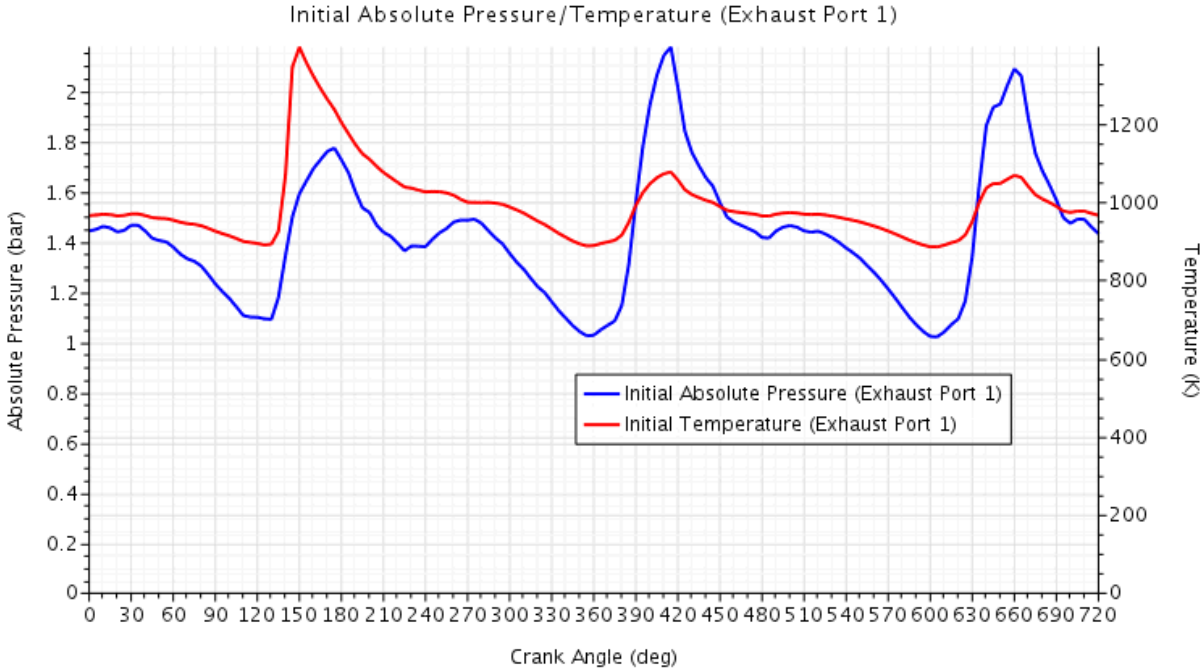
To set initial conditions for the ports:

7. Using the same procedure as described for the cylinder (Steps 2 - 5), set initial conditions for:
   - **Intake Port 1** using the file `intakePort1_Inlet.csv`
   - **Exhaust Port 1** using `exhaustPort1_Outlet.csv`

   The plots below display the absolute pressure and temperature curves:



At 300 deg crank angle, the intake port pressure value is 0.97650 bar. The temperature value is 316.772 K.



For the exhaust port, the initial absolute pressure value is 1.35256 bar and the initial temperature value is 986.679 K.

8. Click **Apply**.
9. Save the simulation .

Setting up the Piston

Here, you set the bore and the length of the connecting rod.

To set the bore and the connecting rod length:

1. In the *Edit* dialog for the cylinder, click the *Motion* tab.
2. Set the following properties:

Property	Setting
Bore	`80.0 mm`
Connecting Rod Length	`145.0 mm`

3. Click **Apply**, then **Close**.
4. Save the simulation .

Setting the Inlet and Outlet Boundary Conditions

You set the boundary conditions for the inlet and the outlet using the previously imported files `intakePort1_Inlet.csv` and `exhaustPort1_Outlet.csv`.

To set the boundary conditions for the inlet:

1. Right-click the **Engine** > **Cylinder** > **Inlet** node and select **Edit**.
2. Within the *Edit* dialog, in the **Static Temperature** group-box, set the following properties:

Property	Setting
File	**intakePort1_Inlet**
Temperature Column	**T_K**
Temperature Units	**K**

3. In the **Pressure** group-box, set **File** to **intaketPort1_Inlet**.
4. Click **Apply** and **Close**.

To set the boundary conditions for the outlet:

5. Right-click the **Engine** > **Cylinder** > **Outlet** node and select **Edit**.
6. Using the same procedure as described for the inlet (Steps 2 - 4), set the boundary conditions using the file **exhaustPort1_Outlet**.
7. Save the simulation .

Setting Up the Valve Motions

For the motion of the valves, you import tables that describe the valve lifts as a function of crank angle. You correct the lash of the valves using the ValveLash parameter.

To define the motion for the exhaust valve:

1. Right-click the **Engine** > **Exhaust Valve 1** node and select *Edit*.
2. In the *Edit* dialog, next to the **Valve Lift File** drop-down menu, click **Import**.
3. In the *Import Valve Lift Table* dialog, navigate to `[INSTALL_DIR]/doc/startutorialsdata/STAR-ICE/data`.
4. Select `exhaustValve1_vlift.csv` and click **Open**.

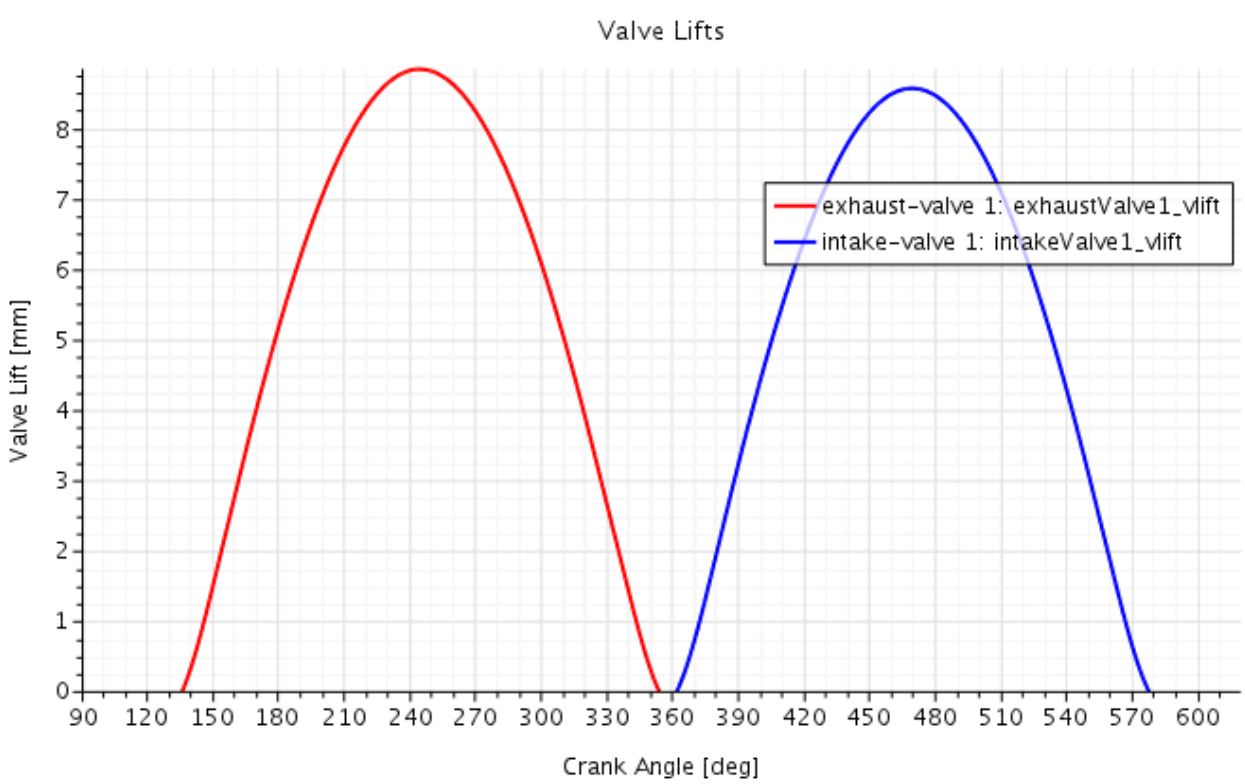
   In the *Graphics* window, a *Valve Lifts* plot opens, which displays the valve lift as a function of crank angle.
5. To correct the lash of the valve, set **Lash** to `$ValveLash`.

   The valve lift curve moves down by 0.5 mm.
6. Click **Apply**, then **Close**.

To define the motion for the intake valve:

7. Right-click the **Engine** > **Intake Valve 1** node and select *Edit*.
8. Using the same procedure as for the exhaust valve (Steps 2 - 6), define the valve lift using the file `intakeValve1_vlift.csv`.

   The *Valve Lifts* plot displays the lift curves for the exhaust and intake valves:



9. Save the simulation .

Preparing the Plots and Scenes for Visualization

STAR-ICE provides several plots and scenes by default. To watch the solution develop, open the plots for cylinder turbulence kinetic energy and tumble-y. Also open the scenes for velocity magnitude and absolute pressure. Write the scenes to image files at regular intervals.

To open the plots:

1. Right-click the **Results** > **Plots** > **Cylinder Turbulence Kinetic Energy** node and select **Show**.

   In the *Graphics* window, the *Cylinder Turbulence Kinetic Energy* plot opens and displays the mass-averaged turbulence kinetic energy in the cylinder as a function of crank angle.
2. Repeat Step 1 for the **Results** > **Plots** > **Tumble-Y** node.

   This plot displays the tumble about the y-axis as a function of crank angle.

To open the scenes and write them to image files:

3. Using <Ctrl>, multi-select the **Results** > **Scenes** > **Absolute Pressure 1** and **Velocity Magnitude 1** nodes.
4. Right-click and select **Open**.

   The scenes open in the *Graphics* window.
5. Right-click the selected nodes again and select **Edit**.
6. In the *Multiple Objects* dialog, set the following properties:

Property	Setting
Attributes	
⌊ **Update**	
⌊ **Save To File**	✔
⌊ **Trigger**	**Time Step**
⌊ **Time-Step Frequency**	
⌊ **Frequency**	5

7. Click **Close**.
8. Save the simulation .

Running the Simulation

The simulation is now set up and ready to run.

You are advised to run this tutorial on a multi-core machine with 8 cores or more. Depending on the specifications of the machine, this tutorial takes approximately 8 hours to complete.

To run the simulation

1. Right-click the **Engine** node and select **Run Solution**.

STAR-ICE updates the geometry to represent your settings and starts the meshing process. The *Residuals* display opens and shows the progress of the solver. While the simulation is running, STAR-ICE saves the image files for the velocity and pressure scenes to your working directory. The base name of each image file is combined with the number of the update to ensure that each file is unique.

2. After the run is finished, save the simulation .

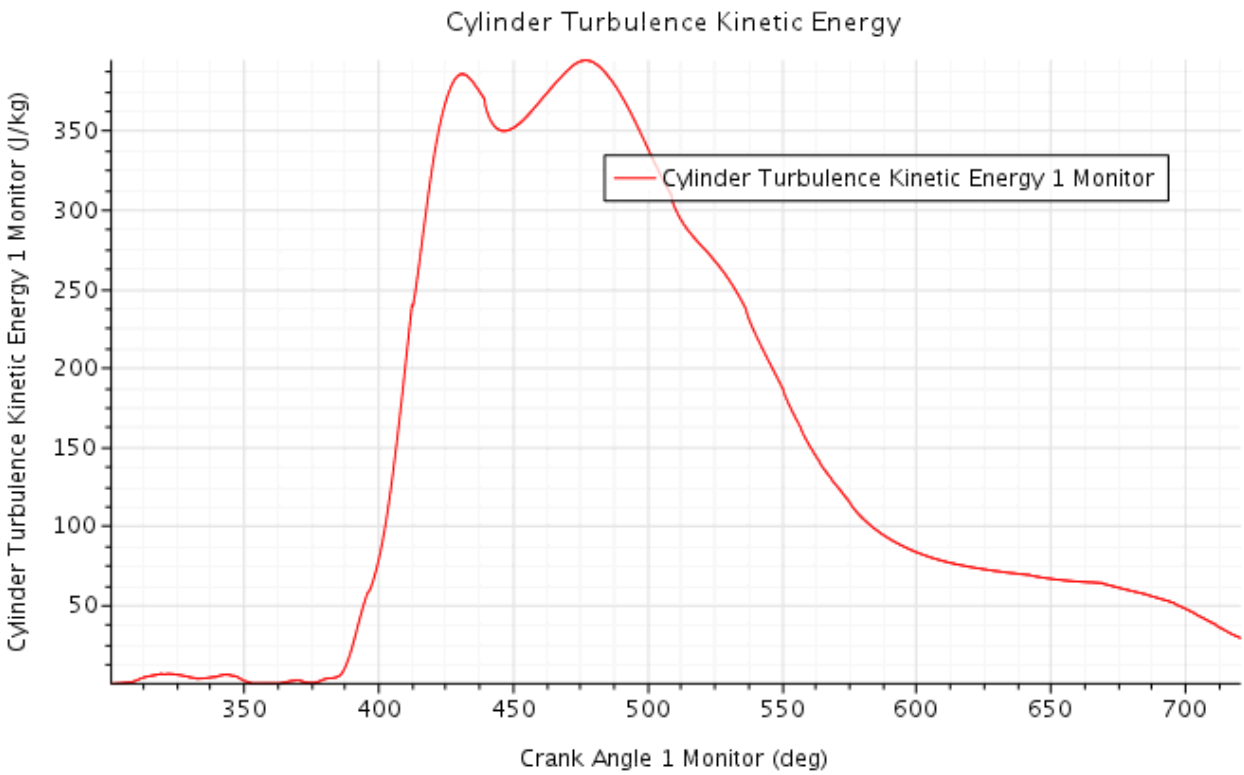
Analyzing the Results

After the simulation is complete, you visualize the results.

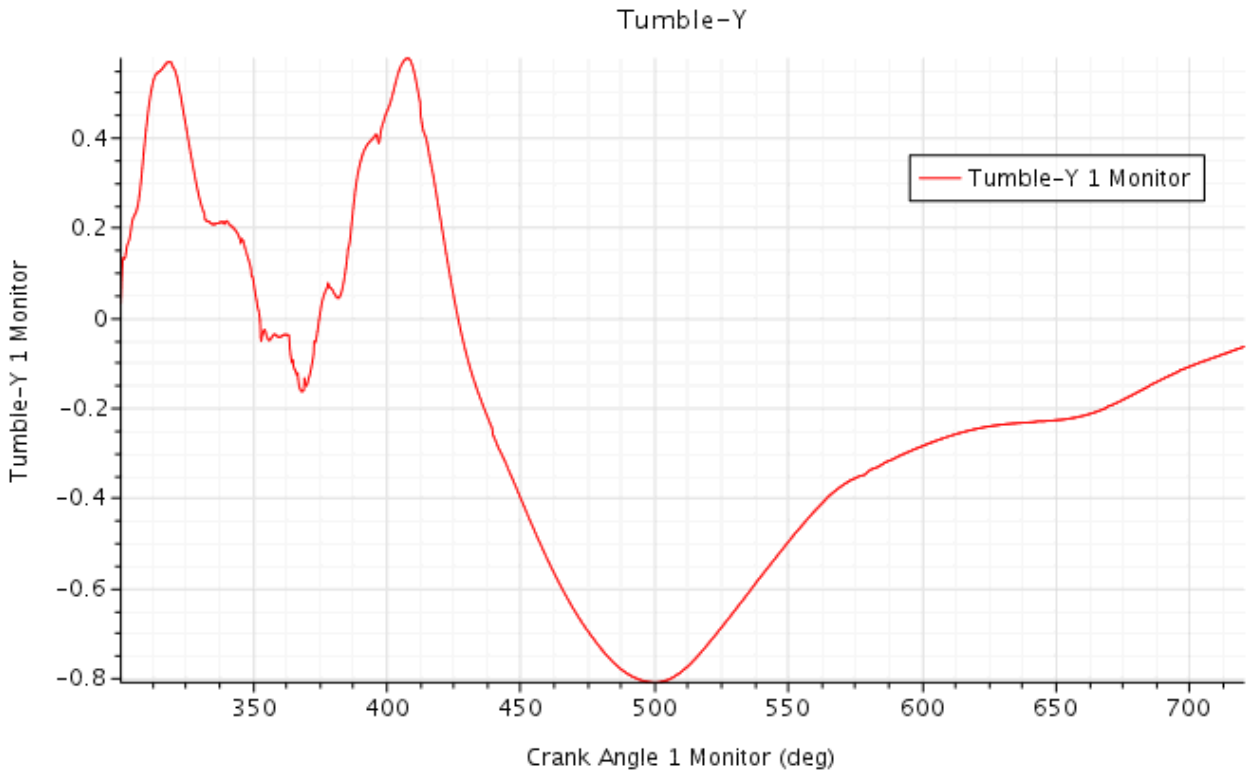
Examine the plots for turbulence kinetic energy and tumble-y:

1. In the *Graphics* window, click through the *Cylinder Turbulence Kinetic Energy* and *Tumble-Y* tabs.

   The following plots display the results after the run:



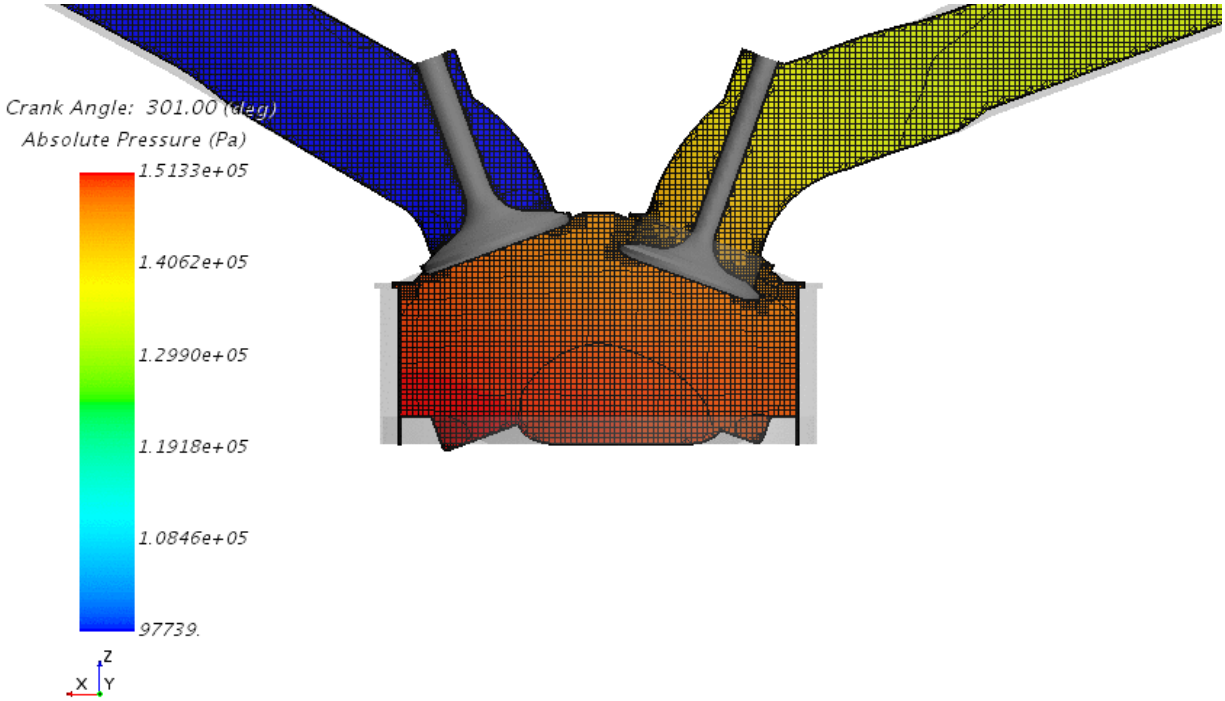
At the ignition point of 720 deg crank angle, the turbulence kinetic energy value is 28.88627 J/kg.

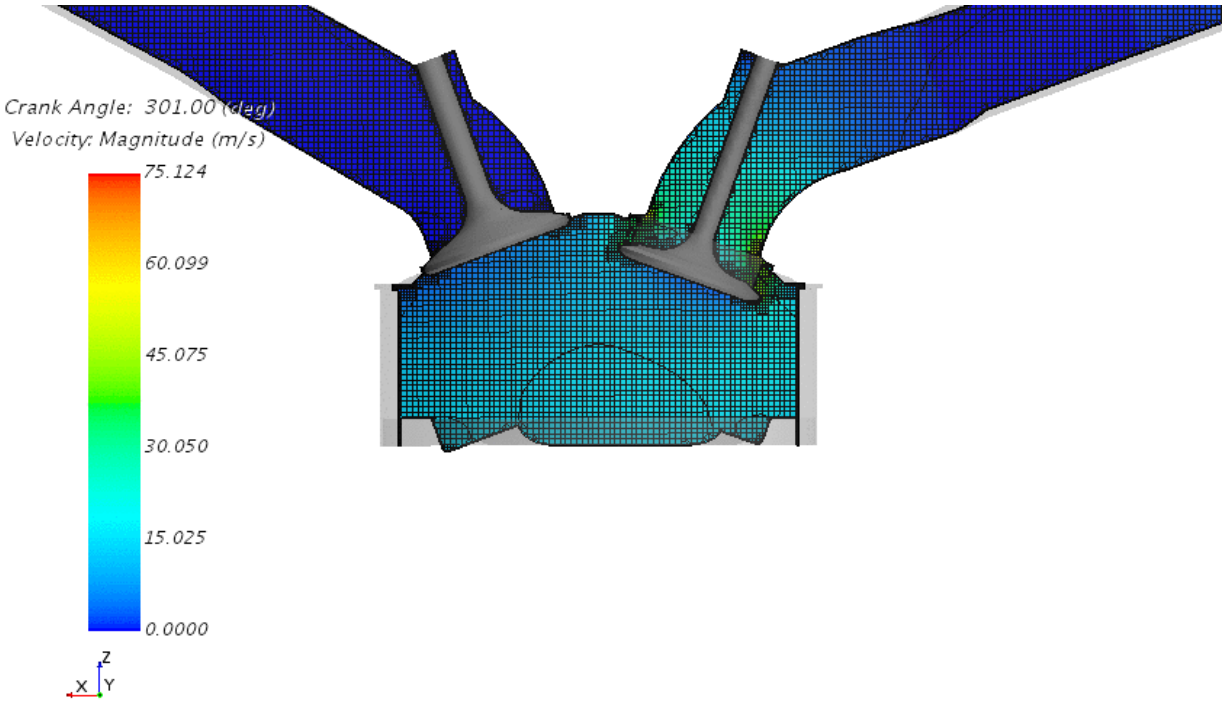


The tumble-y value at ignition point is -0.06428687.

Examine the scenes for absolute pressure and velocity magnitude:

2. In your working directory, click through the saved image files for absolute pressure and velocity magnitude. The saved scene files can be animated with a media program of your choice:





You can see in both animations that STAR-ICE automatically remeshes the geometry at regular intervals to ensure a good cell quality for all crank angles.

Summary

The Gasoline Engine tutorial uses a four-stroke gasoline engine to introduce various features of the STAR-ICE add-on:

The features that were introduced include:

- Launching the STAR-ICE add-on.
- Importing a CAD model.
- Defining the engine solver settings and operation conditions.
- Setting up a half model for the cylinder and valves.
- Defining initial conditions for the cylinder and ports using tabular data.
- Setting up the piston motion.
- Importing valve lift curves.
- Defining a parameter for the valve lash.
- Preparing plots and scenes for postprocessing.
- Running the simulation
- Plotting turbulence kinetic energy and tumble-y in the cylinder as a function of crank angle.
- Analyzing the absolute pressure and velocity magnitude fields.

Siemens PLM Software

Headquarters
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asia-Pacific
Suites 4301-4302, 43/F
AIA Kowloon Tower,
Landmark East
100 How Ming Street
Kwun Tong, Kowloon
Hong Kong
+852 2230 3308

About Siemens PLM Software

Siemens PLM Software, a business unit of the Siemens Digital Factory Division, is a leading global provider of product lifecycle management (PLM) and manufacturing operations management (MOM) software, systems and services with over 15 million licensed seats and more than 140,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software works collaboratively with its customers to provide industry software solutions that help companies everywhere achieve a sustainable competitive advantage by making real the innovations that matter. For more information on Siemens PLM Software products and services, visit www.siemens.com/plm.

© 2017 Siemens Product Lifecycle Management Software Inc. Siemens and the Siemens logo are registered trademarks of Siemens AG. D-Cubed, Femap, Fibersim, Geolus, GO PLM, I-deas, JT, NX, Parasolid, Polarion, Solid Edge, Syncrofit, Teamcenter and Tecnomatix are trademarks or reg istered trademarks of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries. All other logos, trademarks, registered trademarks or service marks belong to their respective holders.

SIEMENS
Ingenuity for life

Siemens PLM Software

STAR-View+
User Guide

Version
12.06

www.siemens.com/mdx

© 2017 Siemens Product Lifecycle Management Software Inc.

Any and all Siemens Product Lifecycle Management Software Inc. brand, product, service and feature names, logos and slogans are registered trademarks or trademarks of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries. All other brand, product, service and feature names or trademarks are the property of their respective owners.

Contents

Using STAR-View+ 1

What Is STAR-View+? 2

Installing STAR-View+ 3

Installing the Downloaded Package on Windows 4

Installing the Downloaded Package on Linux 5

Exporting a Scene to a STAR-View+ File 6

Creating Framesets 8

Working with STAR-View+ 10

Scene Operations 13

Saving STAR-View+ Scene Data 16

Accessing a STAR-View+ File from Microsoft PowerPoint 18

Closing a File and Exiting STAR-View+ 19

Introductory STAR-View+ Tutorial 20

Prerequisites 21

Loading Existing Simulations 22

Preparing Scenes 23

Creating a STAR-View+ Frameset 25

Creating a STAR-View+ Frameset Gallery 27

Exploring the Frameset 28

Comparing Cases in a Frameset Gallery 30

Exporting Scenes from STAR-View+ 32

Recording Scene Movements on Video 33

Accessing STAR-View+ Scenes from PowerPoint 34

Summary 36

Using STAR-View+

This part of the documentation guides you on the use of a viewer for three-dimensional STAR-CCM+ scenes, STAR-View+.

This viewer is only supported on Linux and Windows.

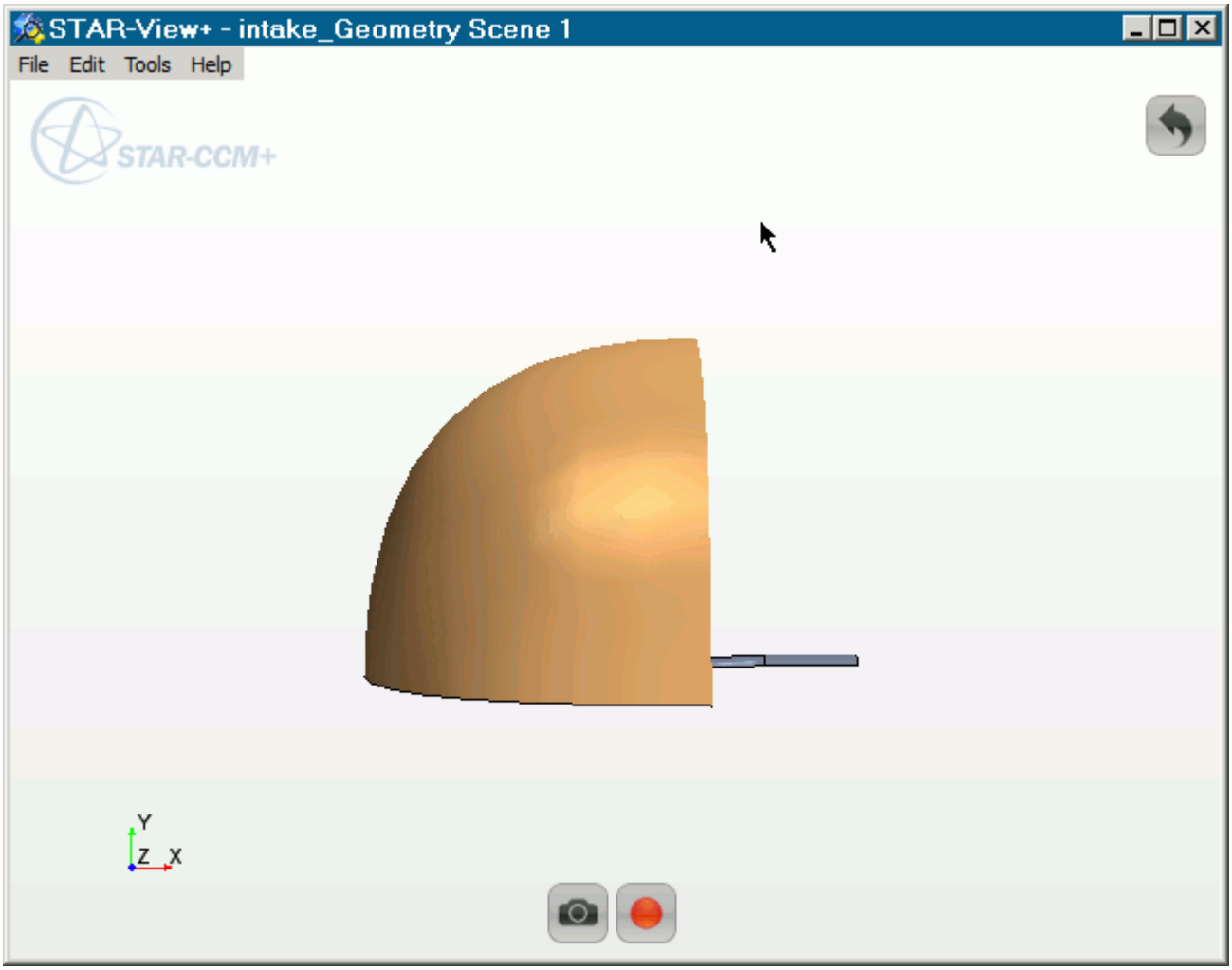
Contents:

What Is STAR-View+?
Installing STAR-View+
Exporting a Scene to a STAR-View+ File
Creating Framesets
Working with STAR-View+

What Is STAR-View+?

STAR-View+ is a light-weight, license-free standalone scene viewer for STAR-CCM+.

You can use it to examine a scene as you would in STAR-CCM+, when that scene is exported to a three-dimensional `.sce` graphics file.



As an alternative to running the full STAR-CCM+ client, this file format is strictly for interactive viewing. You cannot make scene modifications as you would in STAR-CCM+, such as adding parts to be displayed, changing the coloring or using a different scalar.

Installing STAR-View+

STAR-View+ is included as part of the default STAR-CCM+ installation on Windows and Linux 64-bit platforms.

It is also available to download as a separate installable package from the Siemens PLM Software support site.

As no licensing is required for STAR-View+, it is possible to use it for viewing the results of STAR-CCM+ simulations without needing to purchase the software.

Instructions are provided on how to install the downloaded package on Windows and Linux. Note that STAR-View+ requires a Java runtime environment. If necessary, you can provide a custom location for the Java installation using the `STARCCM+_JAVA_HOME` environment variable.

Contents:

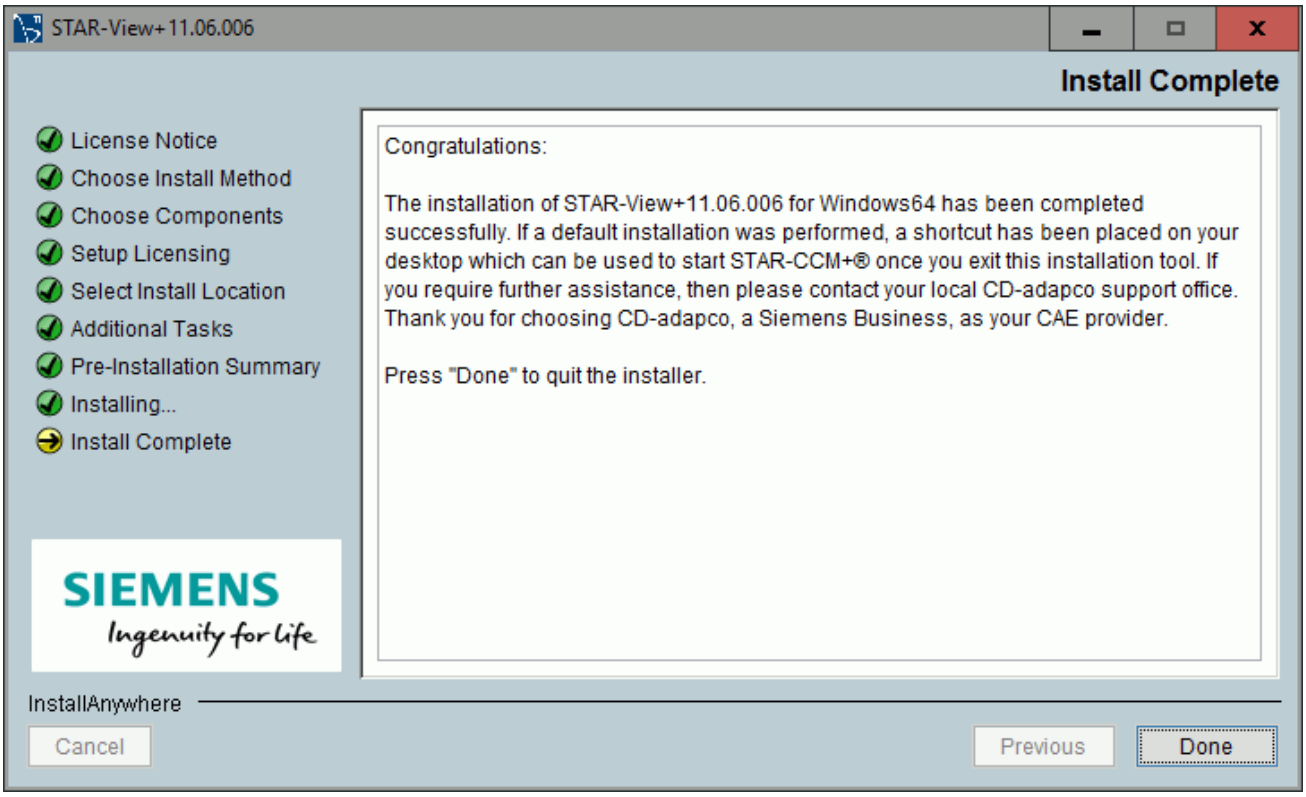
Installing the Downloaded Package on Windows
Installing the Downloaded Package on Linux

Installing the Downloaded Package on Windows

The steps that are required to install STAR-View+ on Windows 64-bit platforms are:

1. Download the appropriate package from the Siemens PLM Software website.
2. Navigate to the download directory using an appropriate file explorer and run the `.exe` file to launch the installer.
3. If a *User Account Control* dialog appears; it can be safely ignored. Click **Yes** to proceed to the installation.
4. The welcome screen of the installer appears.
5. Click **Next** to proceed with the installation. Subsequent pages are presented to ensure that the installation is installed in the correct location, and that appropriate options are provided for launching the tool.

The final screen indicates the end of the STAR-View+ installation process.



When the installation is complete, you can launch STAR-View+ as described in the section Starting STAR-View+.

Installing the Downloaded Package on Linux

Note: For additional information on the requirements for installing on Linux, see the Installation Guide and the Release Notes.

STAR-View+ requires GNU C library (glibc) version 2.12 or greater. The certified and supported platforms that do not run STAR-View+ are:

Certified Platforms that Do Not Run STAR-View+	
Platform	Version of glibc
Red Hat Enterprise Linux Server 5.3	2.5
Red Hat Enterprise Linux Server 5.5	2.5
Red Hat Enterprise Linux Server 5.8	2.5

Supported Platforms that Do Not Run STAR-View+	
Platform	Version of glibc
SUSE 9/10	2.5
SLES 11	2.9
openSUSE 11.1/11.2	2.9

The steps required to install STAR-View+ on Linux 64-bit platforms are:

1. Download the required package from the Siemens PLM Software website.
2. Run the installer script (a `StarView+*.bin` file).
3. In the installer dialog, accept the License Agreement.

   Subsequent pages are presented to ensure the installation is installed in the correct location.

   The final screen indicates the end of the STAR-View+ installation process.

When the installation is complete, STAR-View+ may be launched as described in the section Starting STAR-View+ .

Exporting a Scene to a STAR-View+ File

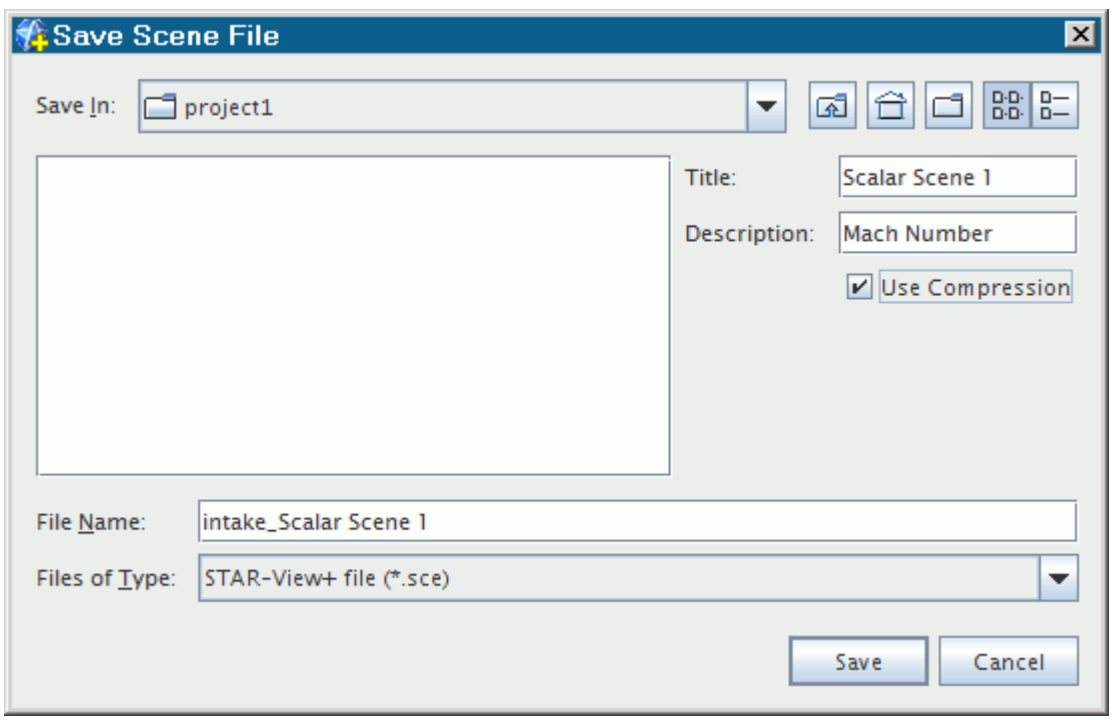
You can export the contents of a scene to a scene file (`.sce`) from STAR-CCM+. You can open the `.sce` file in STAR-View+ and work with the same three-dimensional image that you had in STAR-CCM+.

Be sure to use the same version of STAR-View+ as the version of STAR-CCM+ that you used to create the scene file. You can interact with the image to rotate, pan, and zoom. If you are exporting a scene that includes sweep animation, you must activate the *Precompute Steps* option on the relevant animation control beforehand.

1. To save the scene contents to these formats, within the STAR-CCM+ simulation, right-click on the scene node or the scene display, and select **Export 3D Visualization File...**.

   Note: This option is deactivated if more than one scene node is selected.

   This action activates the *Save Scene File* dialog, in which the **Files of Type:** filter is used to select the format of the saved file. The file type is set to the **STAR-View+ file (*.sce)** format by default.



2. To customize the information that appears in the thumbnail preview of the scene file in STAR-View+, enter a title or a description, or both, in the corresponding text boxes.
3. To compress the data that is saved in the scene file, activate the **Use Compression** option.

4. Navigate to the location where you want to save the file, enter the file name in the **File Name:** text box, and click **Save**.

You can auto-export a scene into a scene file while the simulation is running. This capability is especially useful for transient simulations as it generates framesets within the scene file. To auto-export the scene:

5. Open the node of the scene and select the **Attributes** > **Update** node.
6. Set *File Format* to **sce**.

See Also:

Creating Framesets

Creating Framesets

You can create a frameset, or "book", of scenes in a STAR-View+ file.

A frameset lets you flip through multiple scenes in a single scene file. This capability is useful for assembling a set of related 3D scene images. Examples include a scalar scene with different values of the function, different scalar functions, or even different scenes (such as mesh and scalar) of the object that you are simulating.

To create a frameset:

1. In STAR-CCM+, when saving to a scene file using the *Save Scene File* dialog, select the file of the same scene as the one that you are exporting.

   Note: It is not possible to append to .sce files that were exported from a version of STAR-CCM+ prior to 10.02.

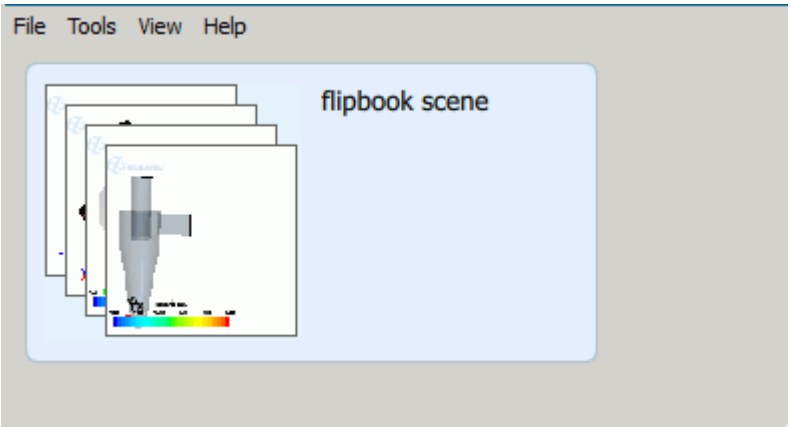
2. In the dialog that appears asking you if you want to overwrite the file, click **Append**.
3. To append a different type of scene, for example to append a scalar scene to the file of a geometry scene, edit the Title text box so that the same title is used.

   Note: You can append scenes to a scene file without editing the title, but the resulting file does not work as a frameset. Rather, it is simply a group of scenes that appear together in a STAR-View+ preview.

To view a frameset:

4. Open the scene file as you would any other.

   The preview shows a thumbnail preview of the frameset with the title and description that you specified.



5. To view the frameset, double-click its preview as you would for any other scene file.

6. To scroll through the frameset, click (**Step Forwards**) or (**Step Backwards**), or use the slider bar, near the bottom of the display.

   These buttons are available only in a scene file that has scenes appended using the same title.

Animating framesets

7. To animate a frameset, click (**Play Frameset Animation**). To stop the frameset animation, click (**Pause Frameset Animation**).
8. To have the frameset animation repeat in a loop, activate the checkbox to the right of the slider bar.
9. To adjust the frameset animation speed:
   a) Select **Tools** > **Settings**.
   b) In the *STAR-View+ Settings* dialog, you can enter values for **Target FPS for frameset animation** or **Target frameset animation length**.
   c) Click **OK**.

Note: These settings are named "Target" as the actual rates may vary, due to complexity of the scene files or limited computing power.

See Also:

Exporting a Scene to a STAR-View+ File

Working with STAR-View+

This section provides instructions on launching the standalone scene viewer STAR-View+, and viewing scene files.

Starting STAR-View+

Assuming you have followed the instructions for the installation, start STAR-View+ one of the following ways, depending on your platform:

- On Windows, in the **Start** menu, launch STAR-View+ by selecting its menu item in the **CD-adapco** submenu.

  Note: STAR-View+ does not work with Windows Remote Desktop.

- On Linux, point to the proper directory for starting STAR-CCM+, and enter the following command line:

  ```
  starview+
  ```

  Alternatively, you can include the name (and path if necessary) of the specific `.sce` file to open:

  ```
  starview+ file.sce
  ```

  Note: STAR-View+ does not work with Mesa.

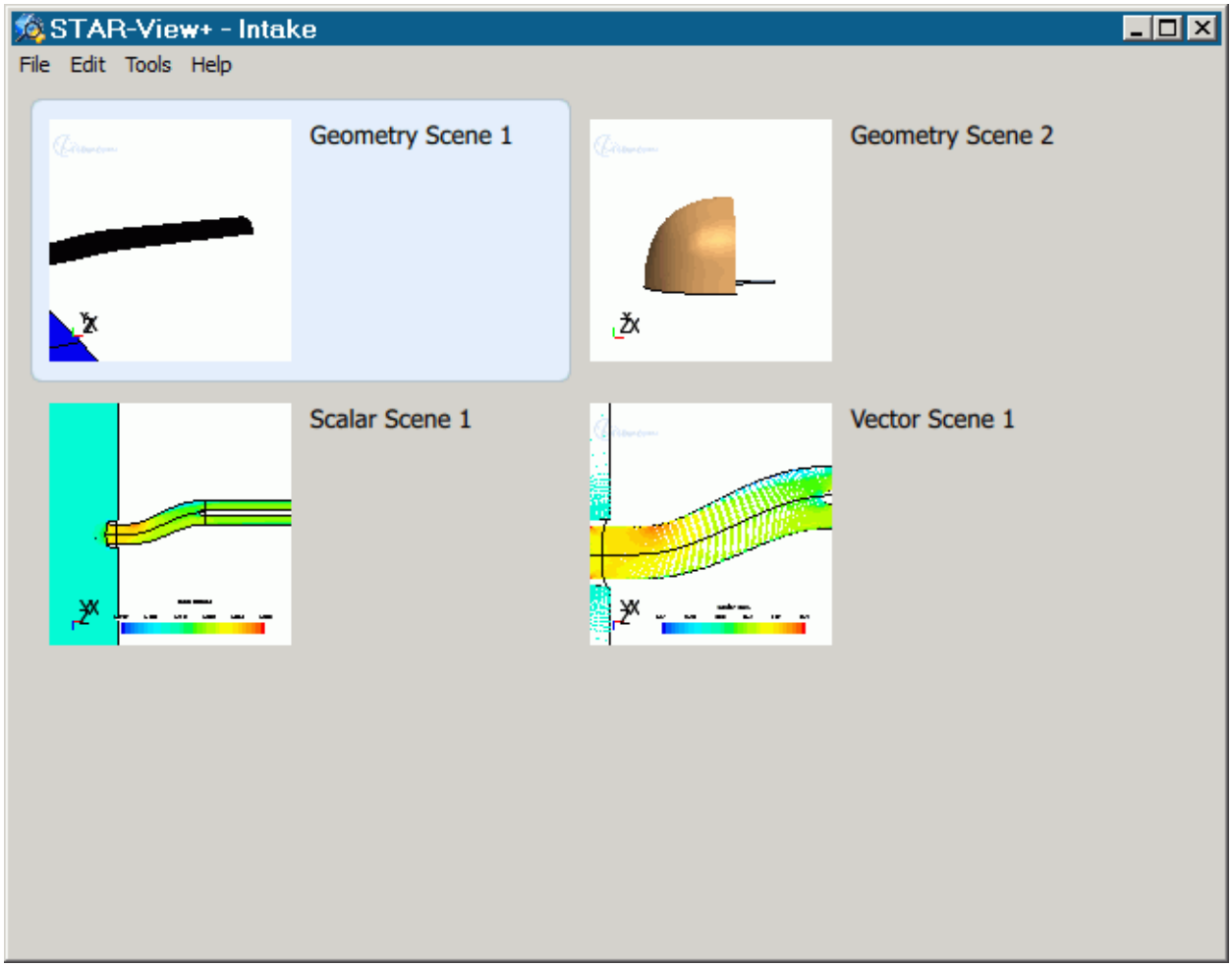
Opening a File

To open a `.sce` file in STAR-View+:

1. Start the viewer.
2. Select **File** > **Open**.
3. In the *Open Scene file* dialog, navigate to the `.sce` file. Click **Open**.

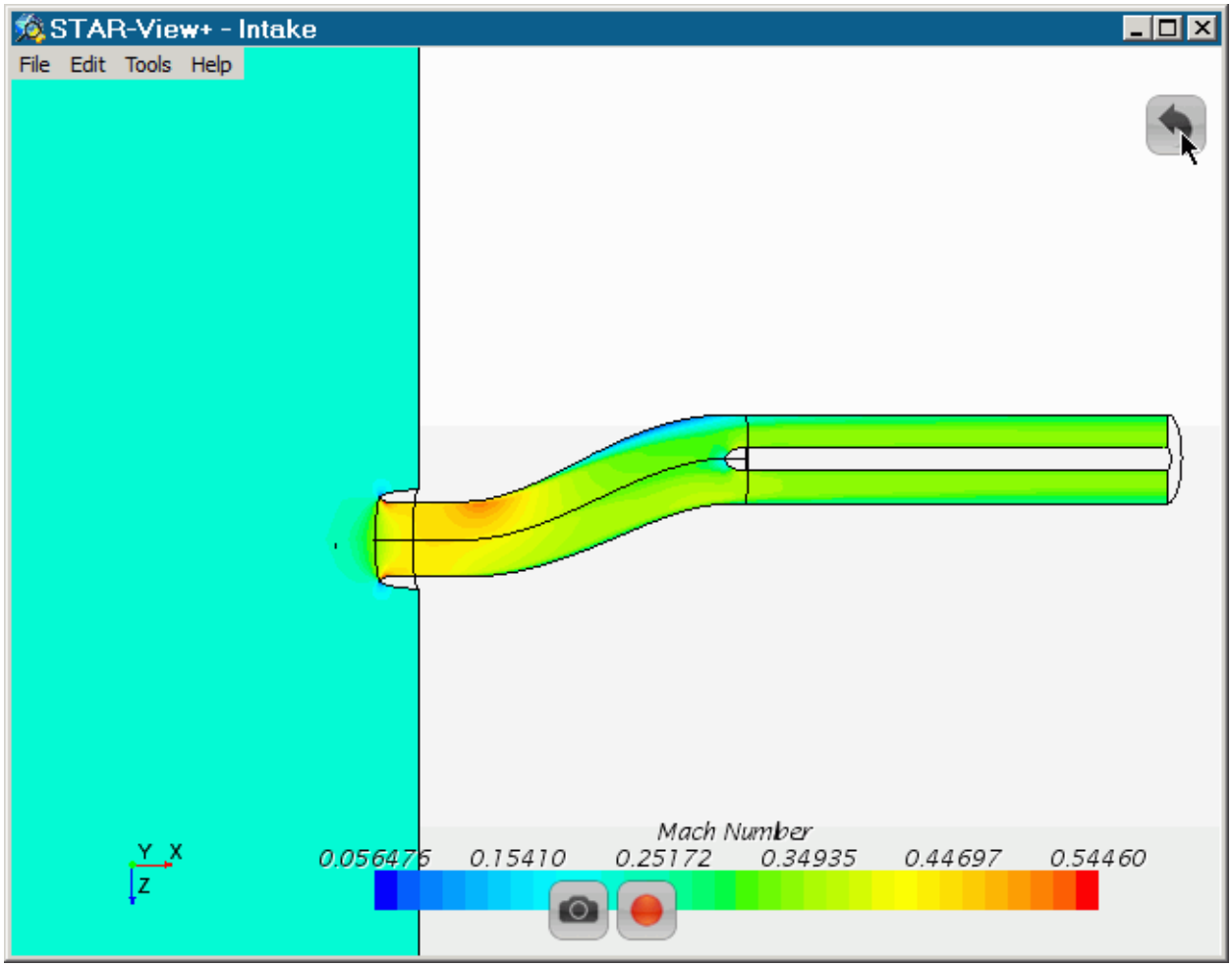
   STAR-View+ automatically opens the scene.

   For a scene file that has additional scenes appended to it, all those scenes are previewed.



If scenes were appended using the same title, a frameset is present, and will appear accordingly in the preview.

4. To view a scene, double-click on its image in the preview.
5. To return to the preview, click the arrow in the upper right corner of the scene.



6. To close the preview, select **File** > **Close**.

Note: STAR-View+ opens one file per session. To view multiple files, run multiple sessions.

Contents:

Scene Operations
Saving STAR-View+ Scene Data
Accessing a STAR-View+ File from Microsoft PowerPoint
Closing a File and Exiting STAR-View+

Scene Operations

Once you have the scene open you can move around a scene, hide and show scene objects, and restore a scene. Use the mouse for manipulating scenes within STAR-View+. If your exported scene contains animations, STAR-View+ can replay them. Such animations include streamlines, vectors, and sweeps.

Moving Around a Scene

The techniques for examining a scene in STAR-View+ are similar to interacting with the scene display in the STAR-CCM+ *Graphics* window.

Panning	To pan the image in the visualization display, hold down the right mouse button and drag the mouse in the direction where you want to move the object. The image appears to change its view angle as you pan. This effect is because the view is in the `Perspective` projection mode, which depicts three-dimensional space. In the `Parallel` projection mode, the view angle stays the same.
Rolling	To roll the scene, hold down the `<Ctrl>` key, and click and drag the mouse upward or downward. Using the `<Ctrl>` key rotates the object along an axis perpendicular to your screen.
Rotating	STAR-CCM+ uses a point on a part to rotate. If you select a point somewhere other than on a part, STAR-CCM+ uses the depth to that last point; the depth is normal to the view plane. The scene can be rotated around various axes, vertical, horizontal, or arbitrary, depending on the direction you move the mouse. To rotate, choose a point on or near the object in the visualization display, and click and drag (holding down the left mouse button) from that point. This technique rotates the object around the axis perpendicular to the direction of your drag. You set the location of the axis when you choose a point to begin clicking and dragging.
Zooming	There are two ways to zoom with the mouse: • Choosing a focal point: Choose a point inside the display, then click and drag downward using the middle mouse button. This action brings your view closer to that point. Moving the mouse straight up has the opposite effect. • Using the mouse wheel: Turning the mouse wheel toward you brings your view closer to the visualization display; the opposite wheel movement takes your view farther away. The mouse pointer must be within the visualization display for this operation to work. Its particular position within the display determines the direction of this type of zooming.

	Note: Vectors in STAR-View+ look as they did when you exported the scene; the vector lengths do not update when you zoom in. This is due to the fact that exported vector scenes do not retain all vector data.

Setting Predefined Views

The following options are available for predefined views in the **View** menu:

- **Side View**, also <S>
- **Top View**, also <T>
- **Front View**, also <F>
- **Flip vertical**, also <C>

Changing the STAR-View+ Projection Mode

Each scene can be viewed in either of two projection modes: parallel or perspective. The perspective projection provides a realistic view of the scene whereas the parallel mode looks less realistic and is similar in style to technical diagrams.

By default the scene is set to **Perspective Projection**. To change to parallel mode in the current view, select **View** > **Parallel Projection**. The view of the scene updates automatically.

Hiding and Showing Scene Objects

- To hide an object in a scene, right-click on that object and select **Hide object**.
- To restore hidden objects, right-click in the scene and select **Show All**.

Restoring a Scene

- To restore the default settings of the view, right-click on the display and select **Reset View**. Alternatively:
  - Select **View** > **Reset View**.
  - Press <R>.

Playing Scene Animations

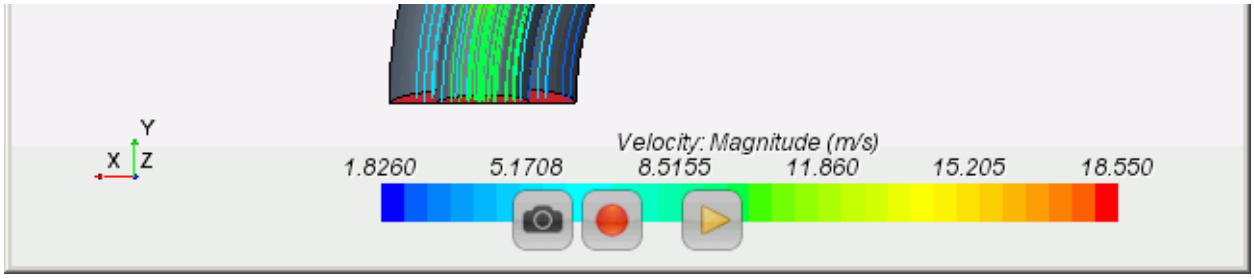
The controls for the scene animation become available only when the active scene contains an animation. To replay sweep animation, you must have activated the *Precompute Steps* option on the animation control before exporting the scene from STAR-CCM+. See "Sweeping Parts Animation" in the STAR-CCM+ User Guide.

Note:

Note: For a frameset, a file with multiple scene images, you can set STAR-View+ to flip from one scene to the next in an animation. See Animating framesets.

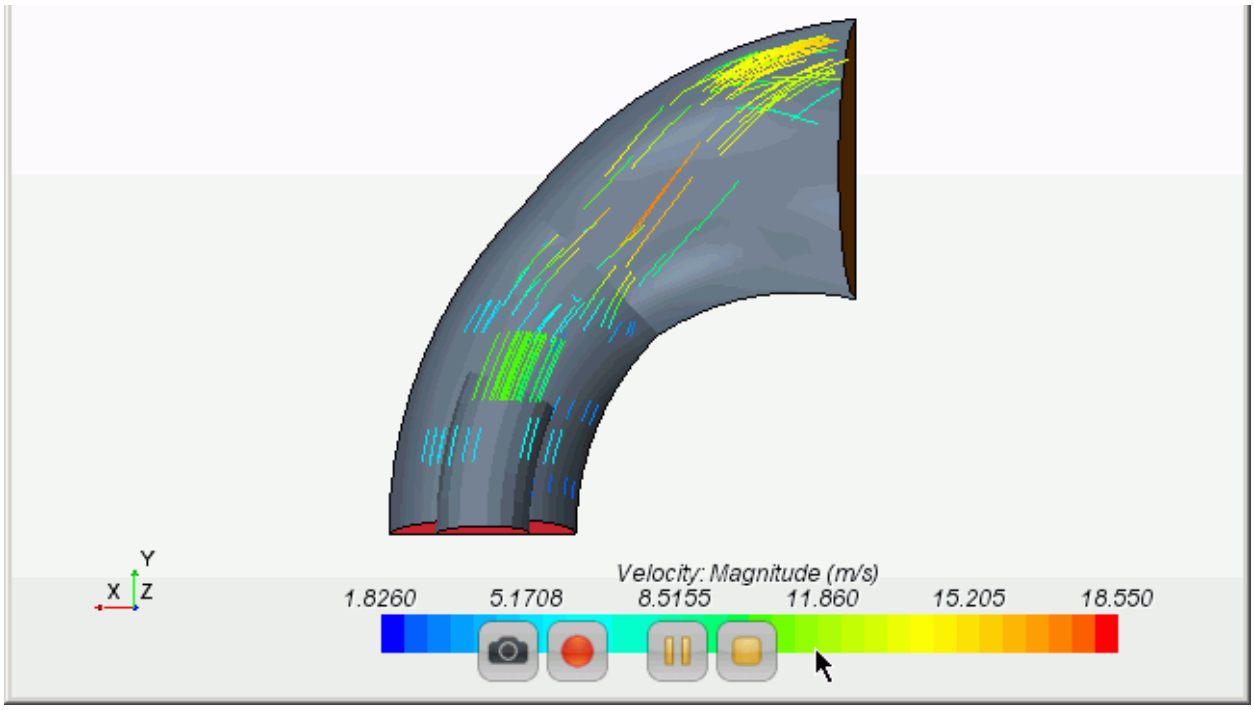
1. To play a scene (or inframe) animation, open the scene file.

   At the bottom of the scene, the controls are available.



2. To start the animation, click (**Play Inframe Animation**).

   This animation includes the same types of basic controls as the animations in STAR-CCM+: (**Pause Inframe Animation**) as well as (**Stop Inframe Animation**).



Saving STAR-View+ Scene Data

In STAR-View+ you can produce output of the contents of a scene file in a variety of ways.

Saving Scene Image Files

You can save a scene to an image file. To do this:

1. Click (**Camera**).

   Alternatively, select **File** > **Hardcopy**.
2. In the *Export Hardcopy* dialog:
   a. Navigate to the location for the graphic file.
   b. Enter a name for the file in the **File name:** text box.
   c. Select the graphics format from the **Save as type:** drop-down list.
   d. Click **Save**.

Saving Scene Files

You can export the contents of a scene file to another scene file using STAR-View+.

1. With a scene open, select **File** > **Export**.
2. In the *Export Hardcopy* dialog:
   a. Navigate to the location for the scene file.
   b. Enter a name for the file in the **File name:** text box.
   c. Click **Save**.

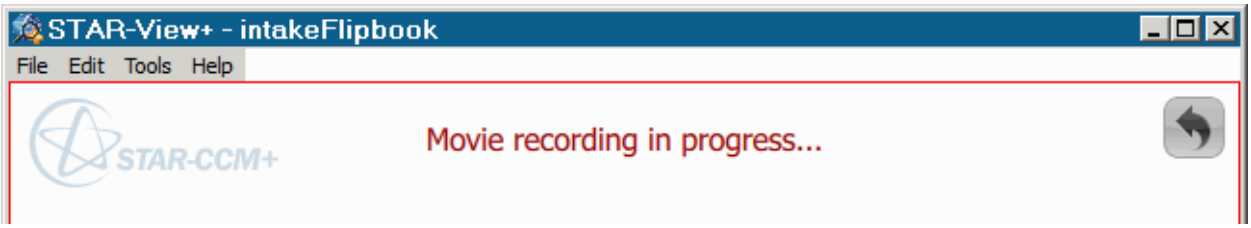
Recording Interaction with the Scene

You can record your movements of the scene view in an `.mpeg` video file. This feature lets you create a visual tour of the scene.

1. To begin the recording, click (**Record**).

   Alternatively, select **Tools** > **Record Movie**.
2. In the *Save Movie File* dialog:
   a. Navigate to the location for the video file.
   b. Enter a name for the file in the **File name:** text box.
   c. Click **Save**.

   Once you finish saving the file, the recording starts. During the recording, a message appears in the scene display.



3. Carry out the actions in the scene.
4. When done, click (**Stop Recording**).

   Alternatively, select **Tools** > **Stop Recording**.

Accessing a STAR-View+ File from Microsoft PowerPoint

You can open your scene file by using a link from a Microsoft PowerPoint 2010 and 2013 (32- or 64-bit) presentation.

Previous versions of Microsoft PowerPoint are not officially supported.

To link to a STAR-View+ scene (`.sce`) file in Microsoft PowerPoint 2010 or 2013, do the following:

1. Launch Microsoft PowerPoint 2010 or 2013.
2. Open the presentation file or create a new one.
3. Select the slide in which you wish to insert the link.
4. Click the **Insert** tab on the ribbon.
5. Click the **Hyperlink** button.
6. In the *Insert Hyperlink* dialog, find and select the `.sce` file, and click **OK**.

   The link appears in the slide.
7. You can edit the text of the link, for example to indicate the scalar field that you are displaying.
8. To open the scene file, do one of the following:
   - In the PowerPoint edit mode, right-click the link and select **Open Hyperlink**.
   - In the PowerPoint presentation mode, click the link.

   STAR-View+ launches with a preview of the scene file.
9. Work with the scene file in STAR-View+.

Closing a File and Exiting STAR-View+

To close a file, you can open another file or exit STAR-View+.

To exit STAR-View+, do either of the following:

- Select **File** > **Quit** from the menu.
- Press the `<Alt><F4>` keyboard shortcut.

Introductory STAR-View+ Tutorial

This tutorial introduces all of the basic functionality of STAR-View+.

Using scenes from a steady case, the simulation that was created in the tutorial, 3D-CAD: Cyclone Separator, the procedure includes setting up and exploring different types of scene (.sce) files for different uses. Additionally, various presentation techniques are explored, including video files and PowerPoint presentations.

Contents:

Prerequisites
Loading Existing Simulations
Preparing Scenes
Creating a STAR-View+ Frameset
Creating a STAR-View+ Frameset Gallery
Exploring the Frameset
Comparing Cases in a Frameset Gallery
Exporting Scenes from STAR-View+
Recording Scene Movements on Video
Accessing STAR-View+ Scenes from PowerPoint
Summary

Prerequisites

The instructions in the STAR-View+ tutorial assume that you are already familiar with certain techniques in STAR-CCM+.

Before attempting this tutorial, make sure that you are familiar with the following techniques:

Technique	Tutorial
The STAR-CCM+ workflow	Introduction to STAR-CCM+
Using visualization tools, scenes, and plots	Introduction to STAR-CCM+

Loading Existing Simulations

For this tutorial, use an existing simulation file that contains objects and solution data from the tutorial, 3D-CAD: Cyclone Separator.

To start this tutorial, you can either:

- use your own versions of the simulation files from the Cyclone Separator tutorial, if you have already worked through it; or,
- use the versions from the Tutorials Files package on the Steve Portal.

To use your own versions of the simulation files, copy the files `cyclone.sim` and `cycloneModified.sim` to the working directory.

To obtain the Tutorials Files package:

1. Follow the instructions in the section, Downloading the Tutorial Files from the Steve Portal.
2. After extracting the package to disk, copy the files `ThreeDCADCycloneSeparator1CreatingModelGeometry_final.sim` and `ThreeDCADCycloneSeparator2ModifyingGeometryAndReRunning_final.sim` from `tutorials/geometry` into the working directory.
3. In the working directory:
   a) Rename `ThreeDCADCycloneSeparator1CreatingModelGeometry_final.sim` to `cyclone.sim`.
   b) Rename `ThreeDCADCycloneSeparator2ModifyingGeometryAndReRunning_final.sim` to `cycloneModified.sim`.

To open the simulation files:

4. Launch STAR-CCM+.
5. Select **File** > **Load Simulation...**
6. In the *Load Simulation* dialog, click **Browse...**
7. In the *Open* dialog, navigate to the working directory.
8. Select `cyclone.sim` then click **Open**.
9. In the *Load Simulation* dialog, click **OK**.
10. To open `cycloneModified.sim`, select **File** > **Load Simulation...** and repeat the subsequent steps.

Preparing Scenes

Modify the scenes of the cyclone separator simulation files in STAR-CCM+ so that they are better suited for presentation.

Choose colormaps, add annotations to the mesh, scalar, and streamlines scenes, and prepare the scalar scenes for sweep animation.

Choose colormaps that highlight the large differences in scalar values.

1. In `cyclone.sim`, open all scenes except the geometry scene.
2. Expand the **Scenes** > **Scalar Scene 1** > **Displayers** > **Scalar 1** node.
3. Set the following properties:

Node	Property	Setting
└ **Scalar 1**	*Contour Style*	**Smooth Filled**
└ **Color Bar**	*Color Map*	**purple-red**
	Levels	`128`

4. Expand the **Scenes** > **Streamlines** > **Displayers** > **Streamlines Stream 1** node.
5. Set the following properties:

Node	Property	Setting
Streamlines Stream 1		
└ **Color Bar**	*Color Map*	**purple-red basic (large difference)**
	Levels	`128`

6. In `cycloneModified.sim`, repeat the above procedure.

Add annotations that identify each scene and its case:

7. In `cyclone.sim`, right-click the **Tools** > **Annotations** node and select **New** > **Simple Text**.

   The simple text annotation node appears in the object tree with the default name **User**.
8. Create two more simple text annotations.
9. Select the **Annotations** > **User** node.
10. For the *Text* property, enter `Mesh Scene Initial`.
11. Rename the **User** node with the same text, `Mesh Scene Initial`.
12. Repeat the above procedure with the two other annotations:

a) Use the text `Scalar Scene Initial` for the **User 2** node.

b) Use the text `Streamlines Initial` for the **User 3** node.

13. Drag each annotation node into the respective scene.

    Each annotation appears at the bottom left of the scene.

14. Drag the annotation to a more visible position in the scene.

    - If you prefer, edit the *Font* property of each annotation to adjust the appearance of the text.

15. Close the scenes and save the simulation.

16. In `cycloneModified.sim`, to set up annotations, repeat the above procedure except, in the annotation text, replace the word "Initial" with "Modified". For example, specify the annotation text of the mesh scene as `Mesh Scene Modified`.

Set up sweep animation in the simulations.

17. In `cyclone.sim`, edit the **Scenes** > **Scalar Scene 1** > **Displayers** node.

18. Set the following properties:

Node	Property	Setting
Displayers		
└ **Outline 1**	*Opacity*	`0.15`
	Surface	**Activated**
└ **Scalar 1**		
└ **Animations**	*Animation Mode*	**Continuous Sweep**
└ **Plane Section**	*Precompute Steps*	**Activated**
└ **Plane Section** > **Precompute Options**	*Number of Steps*	`50`

Note: While increasing *Number of Steps* generates a smoother motion in the animation, it also increases memory consumption.

19. Save the simulation.

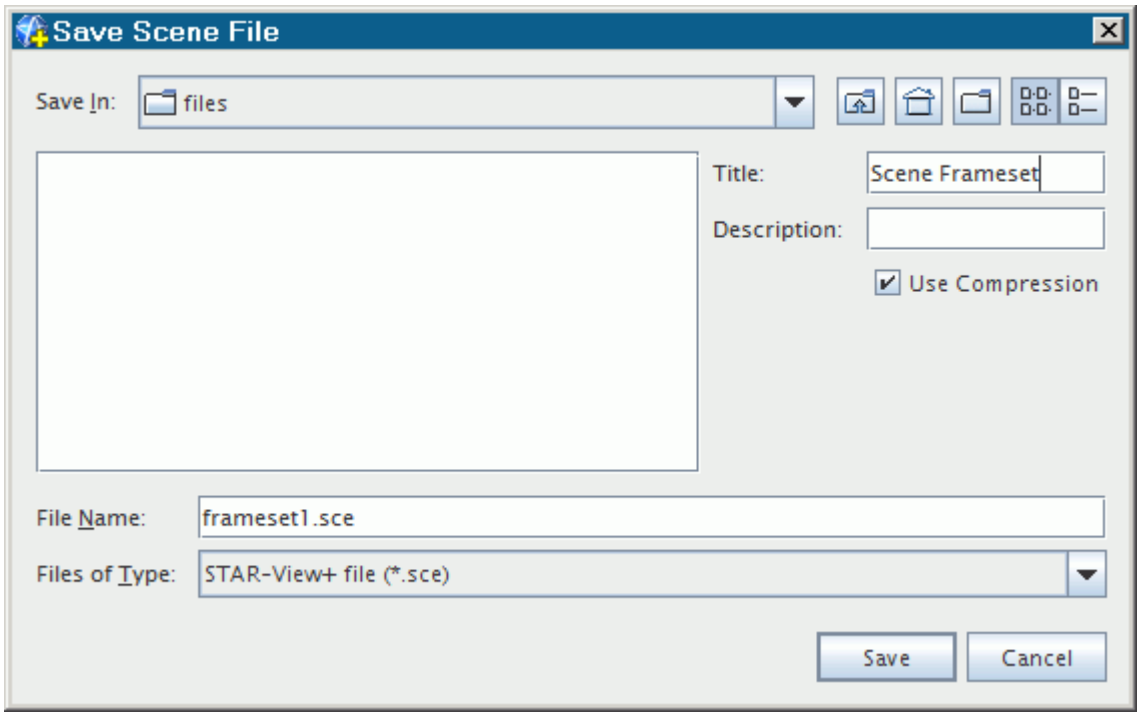
20. In `cycloneModified.sim`, to set up sweep animation, repeat the above procedure.

Creating a STAR-View+ Frameset

Begin exporting scenes to STAR-View+ `.sce` files by creating a frameset, which allows you to navigate back and forth among a set of STAR-CCM+ scenes.

Create the scene file.

1. In `cyclone.sim`, open all scenes except the geometry scene.
2. Make the mesh scene active.
3. Right-click in the scene display and select **Export 3D Visualization File...**.
4. In the *Save Scene File* dialog:
   a) Navigate to the working directory.
   b) In the **File Name:** text box, enter `frameset1.sce`.
   c) Activate the **Use Compression** option.
   d) For the title, enter `Scene Frameset`.



5. Click **Save**.

Append other scenes to the existing scene file.

6. Make the scalar scene active.
7. Right-click in the scene display and select **Export 3D Visualization File...**.
8. In the *Save Scene File* dialog, repeat the settings that you specified for the mesh scene.
9. Click **Save**.
10. In the dialog that appears, click **Append**.
11. Make the streamlines scene active.
12. Repeat the above procedure.

    You have finished setting up your frameset file.

Creating a STAR-View+ Frameset Gallery

The frameset gallery is a STAR-View+ `.sce` file that provides a set of thumbnail previews of the framesets.

Create the scene file.

1. In `cyclone.sim`, make the mesh scene active.
2. Right-click in the scene display and select **Export 3D Visualization File...**.
3. In the *Save Scene File* dialog:
   a) Navigate to the working directory.
   b) In the **File Name:** text box, enter `framesetGallery1.sce`.
   c) Activate the **Use Compression** option.
   d) Leave the title at its default.
4. Click **Save**.

Append other scenes to the existing scene file.

5. Make the scalar scene active.
6. Right-click in the scene display and select **Export 3D Visualization File...**.
7. In the *Save Scene File* dialog, repeat the settings that you specified for the mesh scene.
8. Click **Save**.
9. In the dialog that appears, click **Append**.
10. Make the streamlines scene active.
11. Repeat the above procedure.
12. Close the scenes.
13. Open `cycloneModified.sim`.
14. Open all scenes except the geometry scene.
15. To export the same types of scenes to the same scene file, `framesetGallery1.sce`, repeat the above procedures for `cycloneModified.sim`.

    You have finished setting up your frameset gallery for comparing simulations.

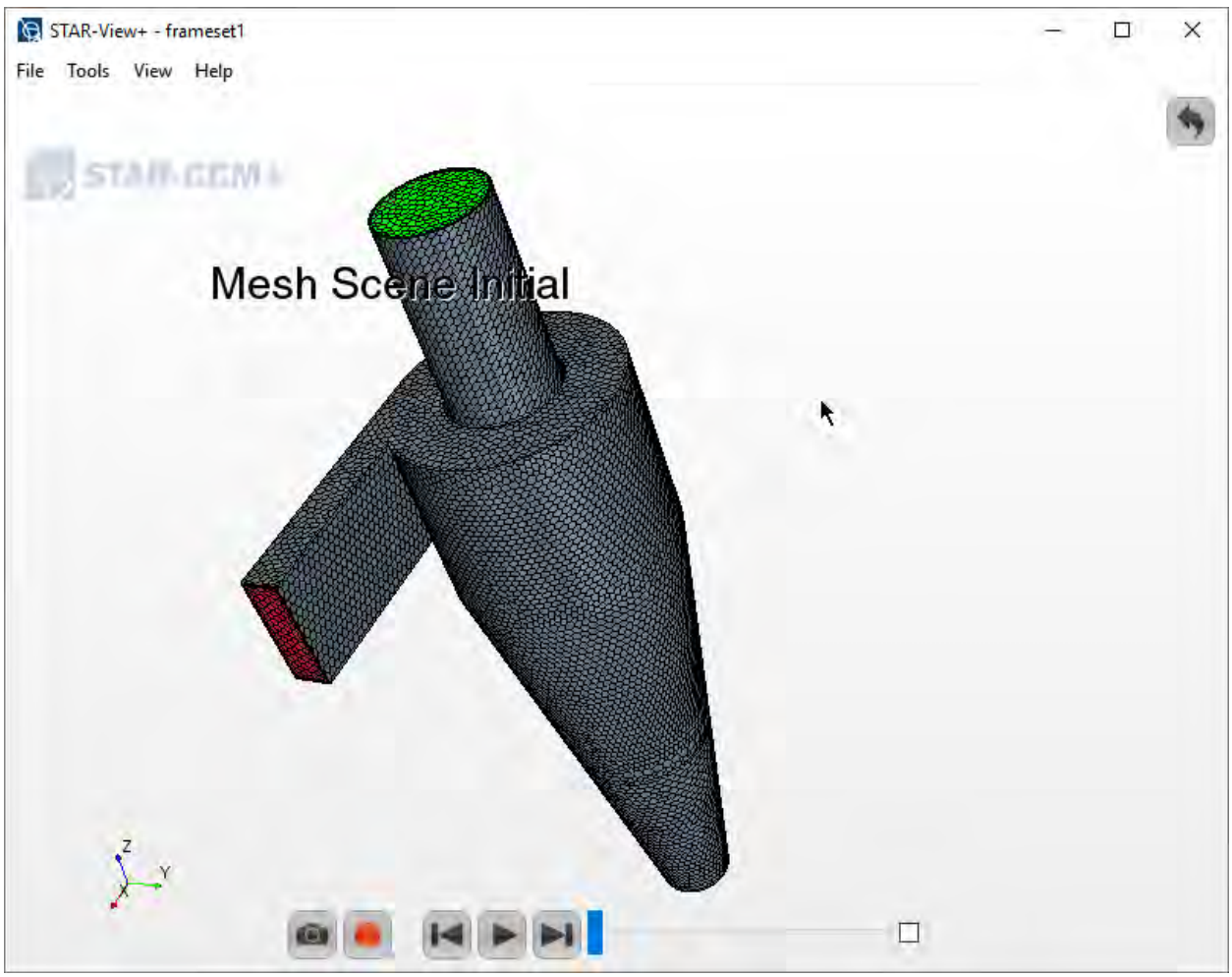
Exploring the Frameset

The STAR-View+ frameset lets you take a "tour" of the scenes, while keeping all of the scene interaction capabilities of STAR-CCM+.

1. Launch STAR-View+ using the appropriate instructions for your operating system.
2. Select **File** > **Open**.
3. In the *Open Scene File* dialog, navigate to the file `frameset1.sce` and click **Open**.

   The display shows the first scene that you exported, in this case the mesh scene.
4. Hover with the mouse in the scene display.

   As you do so, the available controls appear.



5. Use the mouse to move the image in the scene just as you would in STAR-CCM+.
6. Click (**Step Forwards**) and (**Step Backwards**), or use the slider bar, to move among the scenes. To animate the frameset, that is, to have STAR-View+ switch the display from one scene image to the next, click (**Play Frameset Animation**). To stop the frameset animation, click (**Pause Frameset Animation**).

   Because the scalar and streamlines scenes include the animations from STAR-CCM+, the STAR-View+ displays have additional controls.

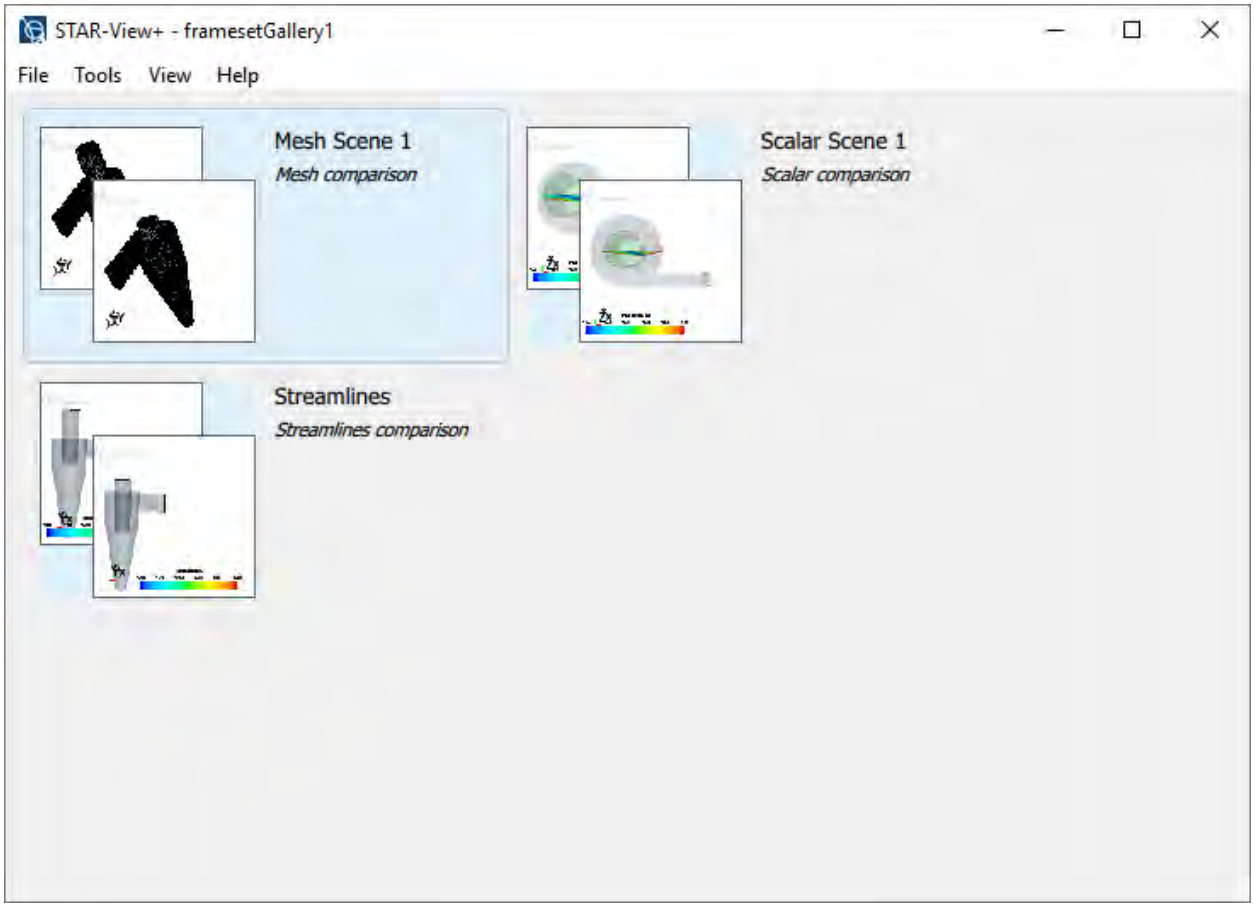
   - To start the animation, click (**Play Inframe Animation**).
   - To stop the animation, do one of the following:
     - Click (**Stop Inframe Animation**).
     - Click (**Pause Inframe Animation**).
     - Navigate to one of the other scenes.
7. To return to the preview, click (**Return**) in the upper right corner of the scene.
8. To close the preview, select **File** > **Close**.

Comparing Cases in a Frameset Gallery

A frameset gallery is a STAR-View+ scene file with multiple framesets.

1. Open the file `framesetGallery1.sce`.

   STAR-View+ shows you thumbnail previews of the available scenes, along with the descriptions.



2. Double-click on the thumbnail preview of the mesh scene frameset.
3. To compare the mesh scenes, use the (**Step Forwards**) and (**Step Backwards**) buttons.
4. Click (**Return**) to return to the previews.
5. Open the scalar scene frameset.
6. Click (**Play**).
7. Move to the next scene in the frameset and click (**Play**) again.

8. Return to the previews.
9. To compare the streamlines and animate them, open the streamlines scene frameset and work with it as with the scalar scene.
10. Return to the previews.
11. To close the preview, select **File** > **Close**.

Exporting Scenes from STAR-View+

You can export scenes from STAR-View+, either as entire scene files, or as image files.

Export the scalar scenes from the frameset gallery to separate scene files.

1. Open the file `framesetGallery1.sce`.
2. Open the scalar scene frameset.
3. Select **File** > **Export**.

   Note: When exporting from STAR-View+, you cannot append scenes to a .sce file. Additionally, animation is not retained.

4. In the *Export Scene File* dialog, save the file as `ScalarSceneInitial.sce`.
5. In the *starview+* dialog:
   a) Activate the **Compress** option.
   b) For the title, enter `Scalar Scene Initial`.
   c) Leave the **Description** text box blank.
6. Click **OK**.
7. In `framesetGallery1.sce`, navigate to the other scalar scene.
8. Repeat the above procedure, except:
   a) Save the file as `ScalarSceneModified.sce`.
   b) For the title, enter `Scalar Scene Modified`.

Export an image of the initial scalar scene.

9. In `framesetGallery1.sce`, return to the initial scalar scene.
10. Click (**Camera**).
11. Save the file as `scalarSceneInitial.png`.

    In the *Hardcopy Options* dialog, accept the default settings by clicking **OK**.

Recording Scene Movements on Video

In STAR-View+, you can examine and animate scenes while recording to a video (`.mpeg`) file.

1. Open `frameset1.sce`.
2. Click (**Record**).
3. In the *Save Movie File* dialog:
   a) Navigate to the location for the video file.
   b) Enter `frameset1` in the **File name:** text box.
   c) Click **Save**.
4. Using the (**Step Forwards**) and (**Step Backwards**) buttons, navigate among the scenes:
   - Move the objects in the scenes to examine significant features.
   - Activate the animations of the scalar scene and streamlines, using controls such as (**Play Inframe Animation**).
5. Click (**Stop Recording**).
6. Play the file `frameset1.mpeg`.

Accessing STAR-View+ Scenes from PowerPoint

You can open your scene file by using a link from PowerPoint, either in edit mode or presentation mode.

1. Launch Microsoft PowerPoint version 2010 or later.
2. In edit mode in PowerPoint, create a presentation file.
3. Make the **Insert** tab active.
4. Click the **Hyperlink** button.
5. In the *Insert Hyperlink* dialog, select the file `framesetGallery1.sce`.
6. Click **Open**.

   The link appears in the slide.

7. Edit the text of the link to indicate the scalar field that you are displaying.

8. Start the PowerPoint slide show (presentation mode).
9. Click the link. Click **OK** in the confirmation dialog.

   Note: If prompted, select STAR-View+ as the default program for opening this type of file.

   STAR-View+ launches with a preview of the scene file.

   Note: To open the scene file while in the PowerPoint edit mode, right-click the link and select **Open Hyperlink**.

10. Examine the scene file in STAR-View+.
11. In PowerPoint, return to edit mode.
12. Save and close the presentation file.

Summary

This tutorial demonstrated the visualization features in STAR-View+.

The features that were introduced include:

- Creating a STAR-View+ frameset.
- Creating a STAR-View+ frameset gallery from two different simulation files.
- Browsing scenes in a frameset.
- Playing a scene animation in STAR-View+.
- Navigating in a frameset gallery.
- Comparing scenes from two different simulations in a frameset gallery.
- Exporting a scene file from a scene file.
- Exporting an image file from a scene file.
- Browsing and examining scenes while recording to a video file.
- Accessing scenes from a Microsoft PowerPoint presentation.

Siemens PLM Software

Headquarters
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

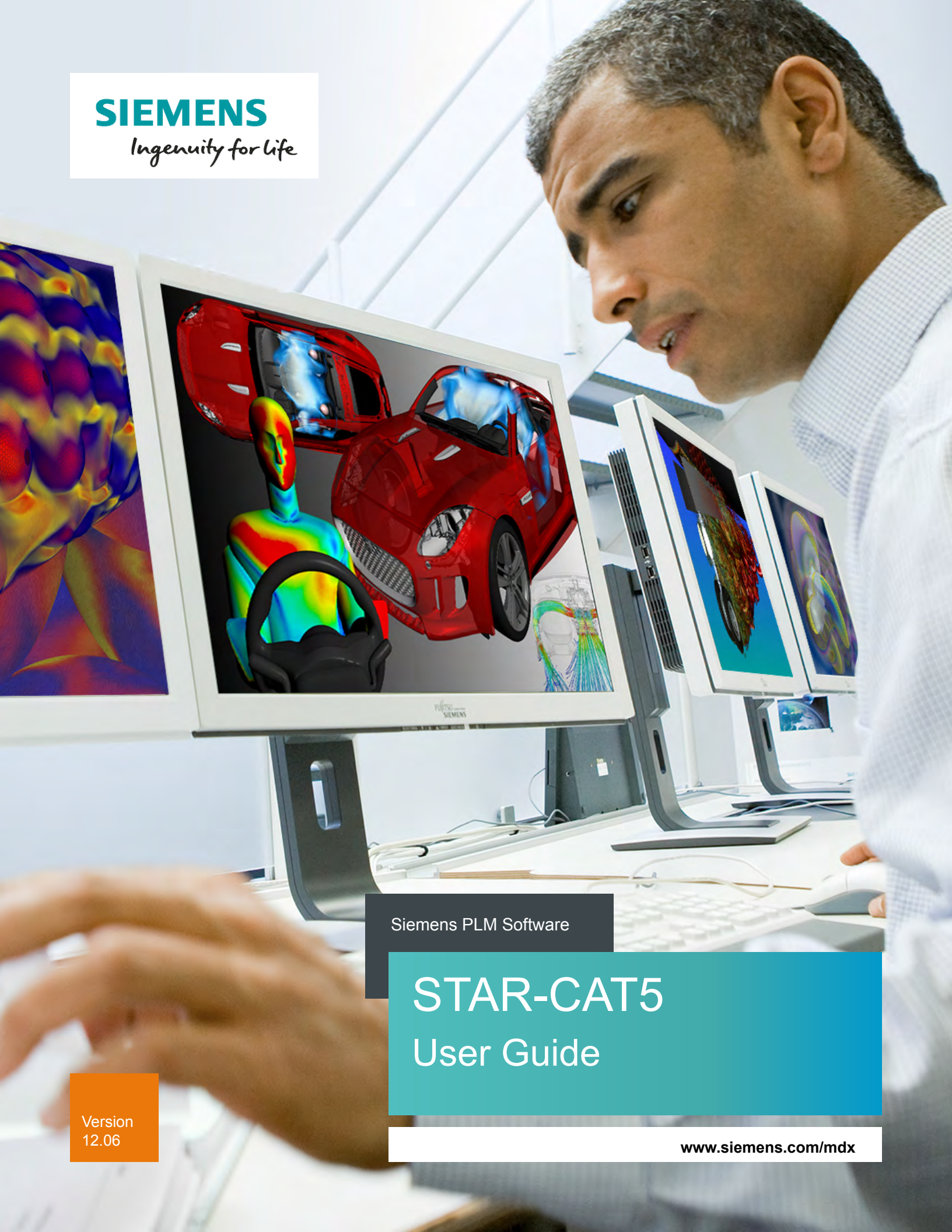
Asia-Pacific
Suites 4301-4302, 43/F
AIA Kowloon Tower,
Landmark East
100 How Ming Street
Kwun Tong, Kowloon
Hong Kong
+852 2230 3308

About Siemens PLM Software

Siemens PLM Software, a business unit of the Siemens Digital Factory Division, is a leading global provider of product lifecycle management (PLM) and manufacturing operations management (MOM) software, systems and services with over 15 million licensed seats and more than 140,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software works collaboratively with its customers to provide industry software solutions that help companies everywhere achieve a sustainable competitive advantage by making real the innovations that matter. For more information on Siemens PLM Software products and services, visit www.siemens.com/plm.

© 2017 Siemens Product Lifecycle Management Software Inc. Siemens and the Siemens logo are registered trademarks of Siemens AG. D-Cubed, Femap, Fibersim, Geolus, GO PLM, I-deas, JT, NX, Parasolid, Polarion, Solid Edge, Syncrofit, Teamcenter and Tecnomatix are trademarks or reg istered trademarks of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries. All other logos, trademarks, registered trademarks or service marks belong to their respective holders.

SIEMENS
Ingenuity for life



Siemens PLM Software

STAR-CAT5

User Guide

Version
12.06

www.siemens.com/mdx

© 2017 Siemens Product Lifecycle Management Software Inc.

Any and all Siemens Product Lifecycle Management Software Inc. brand, product, service and feature names, logos and slogans are registered trademarks or trademarks of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries. All other brand, product, service and feature names or trademarks are the property of their respective owners.

Contents

STAR-CAT5 1
STAR-CAT5 User Guide 2
Introducing STAR-CAT5 3
Transferring Geometry to STAR-CCM+ 32
Understanding the STAR-CAT5 Process 41
Connecting to Remote Machines 45
Defining Materials and Physics 49
Setting Boundary Conditions 73
Meshing the Model 81
Obtaining a Solution 106
Exporting Files and Data 125
Resetting STAR-CAT5 Settings 130
Guidelines 131
STAR-CAT5 Tutorials 186
Cold Flow Through a Motorcycle Exhaust Manifold 187
Porous Media Flow 213
Aircraft Wing Deicing System 238
Rotating Flow Inside a Graphics Card 277
External Flow Over a Car 311
Geometry Transfer: CAD to Simulation 343

STAR-CAT5

The STAR-CAT5 CAD client provides you with straightforward access to CD-adapco's STAR-CCM+ solver from within the CATIA environment.

Standard physics models, including turbulence, ideal gas, heat transfer, and gravity are supported for single body cases. Multiple body simulations involving conjugate heat transfer, rotating reference frames, and porous media are also supported. Advanced physics models can be added to the simulation by either launching STAR-CCM+ and selecting them directly, or by implementing additional Java macro commands.

This section contains the user guide and tutorials for the STAR-CAT5 CAD client.

Contents:

STAR-CAT5 User Guide
STAR-CAT5 Tutorials

STAR-CAT5 User Guide

This part of the documentation details how to create, set up, and run a Computational Fluid Dynamics (CFD) simulation in STAR-CAT5.

This section assumes that you have previous experience in using CATIA, and only describes the additional features available as part of STAR-CAT5.

You can use STAR-CAT5 to transfer your CAD geometry directly into STAR-CCM+ without transferring the CAE settings. This method is an alternate way importing geometry to using the STAR-CCM+ CAD translators. See Transferring Geometry to STAR-CCM+.

Contents:

Introducing STAR-CAT5
Transferring Geometry to STAR-CCM+
Understanding the STAR-CAT5 Process
Connecting to Remote Machines
Defining Materials and Physics
Setting Boundary Conditions
Meshing the Model
Obtaining a Solution
Exporting Files and Data
Resetting STAR-CAT5 Settings
Guidelines

Introducing STAR-CAT5

STAR-CAT5 provides you with straightforward access to CD-adapco's STAR-CCM+ solver from within the CATIA environment.

Standard physics models, including turbulence, ideal gas, heat transfer, and gravity are supported for single body cases. Multiple body simulations involving conjugate heat transfer, rotating reference frames, and porous media are also supported. Advanced physics models can be added to the simulation by either launching STAR-CCM+ and selecting them directly, or by implementing additional Java macro commands.

Contents:

What are the Capabilities?
Getting Started
Common Functions
Running STAR-CAT5 and STAR-CCM+ on Different Machines

See Also:

Using Java Macros in STAR-CAT5

What are the Capabilities?

This section highlights the capabilities of STAR-CAT5.

STAR-CAT5 enables you to:

- Define flow characteristics for the simulation
- Specify mesh parameters and controls
- Execute the STAR-CCM+ solver within STAR-CAT5, with specified solution controls
- Post-process results in STAR-CCM+
- Use parallel processing

Problem Definition

STAR-CAT5 provides an interface for the specification of continua, regions, and boundaries in order to define the CFD simulation. This information is then used by STAR-CCM+ to calculate a solution and present results.

The particular solver options that you can currently select through the STAR-CAT5 interface include:

- Turbulence
  - k-e Model
  - Laminar
- Temperature
  - Thermal Modelling
  - Conjugate Heat Transfer

- Equation of State
  - Constant Density
  - Ideal Gas
- Porous Media
  - Isotropic
  - Uni-Directional
- Moving Reference Frames
- Energy Source
- Gravity
- Passive Scalar

Meshing

STAR-CAT5 provides you with a transparent interface to CD-adapco's surface and volume meshing tools. This allows you to generate high-quality meshes using a solid model that was built within the CATIA environment. The specific meshing functionality includes:

- Surface meshing using the surface remesher for optimum mesh quality
- Prismatic subsurface meshing in the volume mesh for capturing near-wall boundary layers
- Volume meshing of parts with either:
  - Tetrahedral core mesh generation
  - Polyhedral core mesh generation
  - Trimmed hexahedral core mesh generation
- Automatic handling of region-to-region interfaces

Running the Solver

STAR-CAT5 allows the generation of solutions with varying accuracy and speed by providing the following solver controls:

- Number of iterations
- Stopping criterion at which to assume convergence
- Discretization scheme (1st or 2nd order)
- Solver under-relaxation factors
- Additional commands using Java macros

Post-processing

As post-processing is carried out in STAR-CCM+, a wide range of tools are available for analysis. These include:

- Scalar scenes
- Vector scenes
- Streamlines
- Isosurfaces
- Plane sections

- Reports
- Plots

Parallel Processing

STAR-CAT5 supports parallel processing for mesh generation and solver calculations. You can set up parallel processing in the Preferences dialog. The different STAR-CCM+ licensing options can also be selected here. See Setting Up Parallel Processing and Licensing.

Getting Started

The following section provides information on familiarization with the STAR-CAT5 interface, and basic concepts.

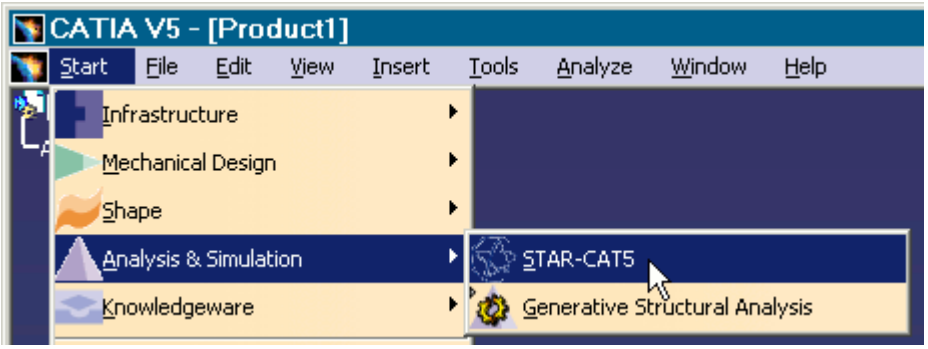
Contents:

Launching STAR-CAT5
The STAR-CAT5 Info App Window
Modifying the config.xml File
Understanding the STAR-CAT5 Interface
The STAR-CAT5 Object Tree
Setting Up Parallel Processing and Licensing
Changing the Units System
Transferring the CAE Model to STAR-CCM+
Editing STAR-CAT5 Settings While the STAR-CCM+ Client is Running

Launching STAR-CAT5

To launch STAR-CAT5:

1. Start CATIA using the STAR-CAT5 desktop icon that was created by the STAR-CCM+ installer.
   For version 12.06.###, this will be `STAR-CAT5 Rxx 12.06.###`. In the unlikely event that a desktop shortcut was not created during the installation process, you can create a new shortcut manually. See Creating a STAR-CAT5 Shortcut Manually.
2. Select **File** > **Open** and open the CATPart or CATProduct that you want to analyze.
3. Select **Start** > **Analysis & Simulation** > **STAR-CAT5**.



The STAR-CAT5 tree is located on the top-left of the window, and the STAR-CAT5 *Info App* window launches.

Contents:

Manually Creating a STAR-CAT5 Shortcut

Manually Creating a STAR-CAT5 Shortcut

If a STAR-CAT5 shortcut is not created by the installer, you can manually generate one yourself by following the steps outlined below.

1. Open a Command Prompt:
   a) Click **Start**.
   b) Type `cmd` in the search field at the bottom of the **Start** menu and select `cmd.exe` from the search results.
2. Enter the following line in the Command Prompt, taking care to preserve the correct spacing:

```
<setcatenv> -e "<shortcutname>" -p
"<catiainstalldir>;<starcadinstalldir>" -a global -icon no -menu
yes -cs CATIA
```

where:

- `<setcatenv>` is the full path to `setcatenv.exe`. The default location for a 64-bit installation is: `C:\Program Files\Dassault Systemes\Bxx\win_b64\code\bin\setcatenv.exe`
- `<shortcutname>` is the name of the new shortcut that will be generated.
- `<catiainstalldir>` is the full path to the CATIA Bxx directory. The default path is: `C:\Program Files\Dassault Systemes\Bxx`
- `<starcadinstalldir>` is the full path to the STAR-CAT5 Bxx directory. The default path for a 64-bit installation is: `C:\Program Files\CD-adapco\STAR-CAD[VERSION]\STAR-CAT5\win64\bin\Bxx` where `x` is the relevant version of your installation.

An example is shown below. In this case a new shortcut for a 64-bit installation of STAR-CAT5 12.06.### is shown. In this case the shortcut name is `STAR-CAT5_R21_TEST`.

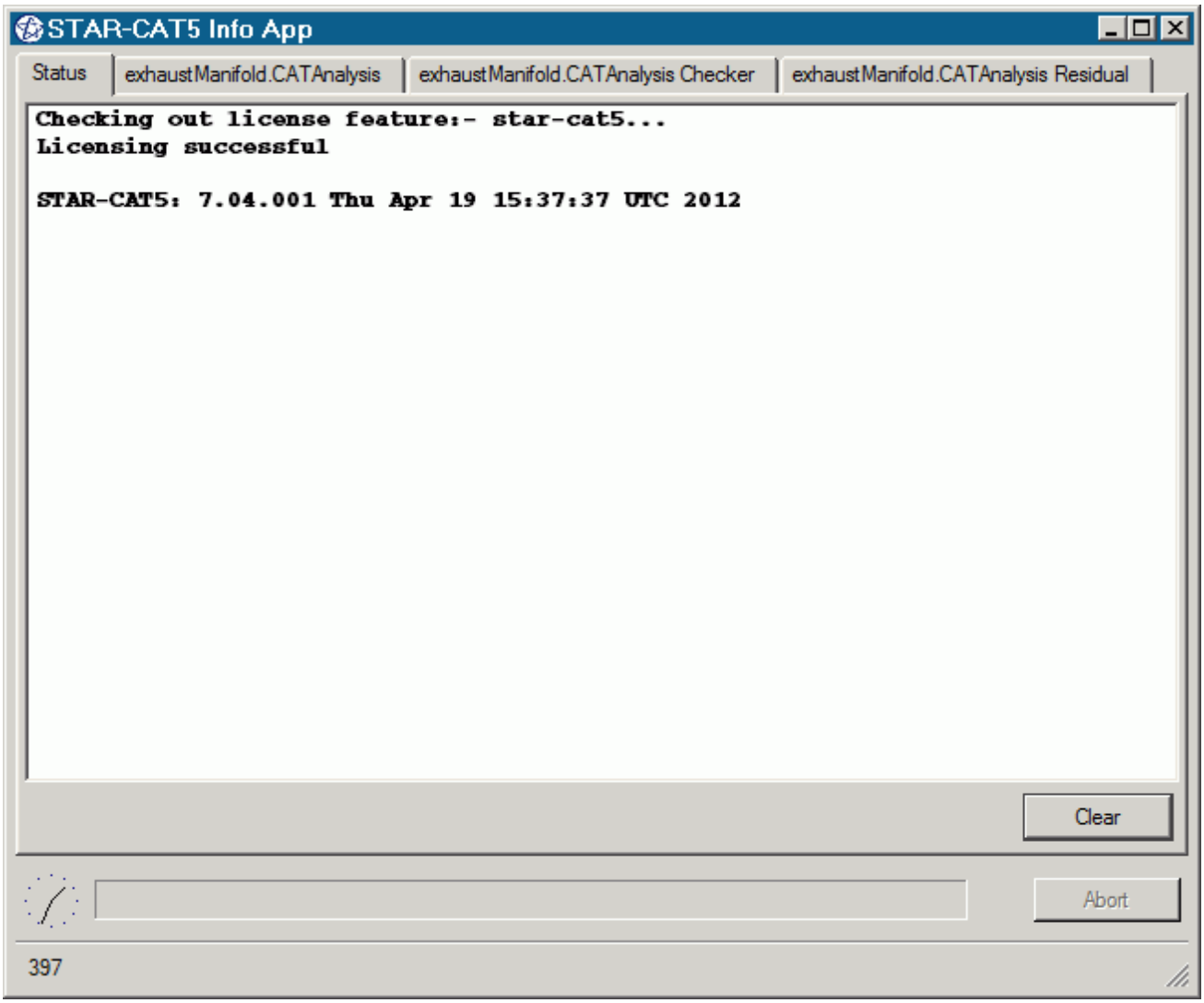
```
"C:\Program Files\Dassault Systemes\B21\win_b64\code\bin
\setcatenv.exe" -e "STAR-CAT5_R21_TEST" -p "C:\Program Files\Dassault
Systemes\B21;C:\Program Files\CD-adapco\STAR-CAD12.06.###\STAR-
CAT5\win64\bin\B21" -a global -icon no -menu yes -cs CATIA
```

A new shortcut will be generated and placed in the **Start** menu of CATIA. The default location is **Start** > **CATIA**.

The STAR-CAT5 Info App Window

The STAR-Works Info App window automatically appears when you start STAR-CAT5.

Initially, this window only contains the *Status* tab which displays the version and licensing information for STAR-CAT5. More tabs appear as you create or load geometry files in STAR-CAT5, allowing you to perform more actions.



You cannot close the STAR-CAT5 Info App window while CATIA or STAR-CAT5 are running. Attempting to close the window by clicking the **X** button returns a message in the *Status* tab informing you of this. You can maximize and minimize the window by clicking the **-** and **_** buttons respectively.

Use the **Clear** button to clear all the information that is displayed in a particular tab.

Use the **Abort** button to stop active processes, such as surface/volume meshing, or the solver. This button only appears when you can stop a process — it is grayed out at all other times.

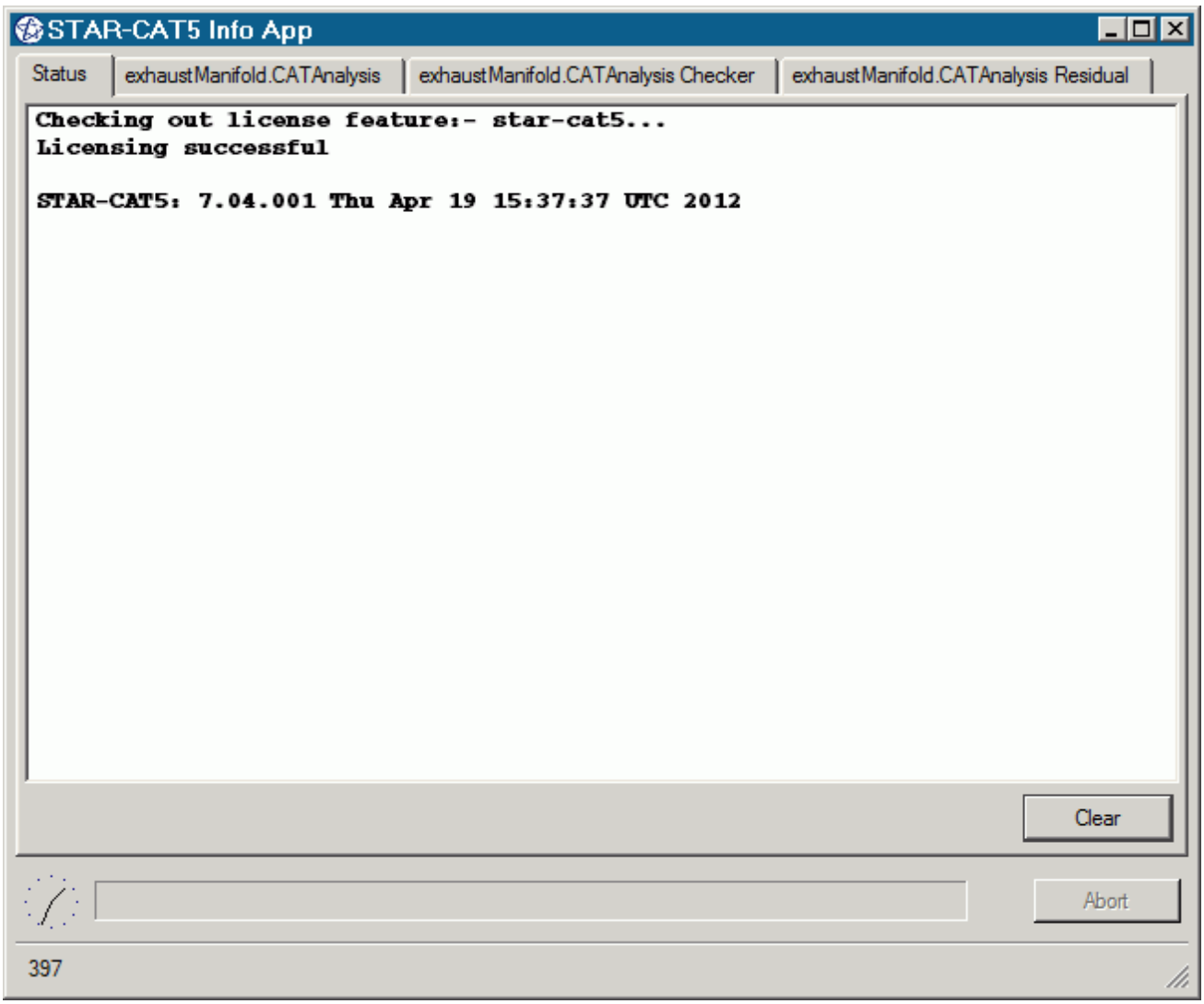
A clock is displayed in the bottom-left of the window. This clock displays the current time on your machine. A progress bar is located next to the clock and reports the progress of various processes, such as mesh generation and the solver.

More information on the various tabs is provided in the following sections:

- The Status Tab
- The Output Tab
- The Model Checker Tab
- The Residuals Tab

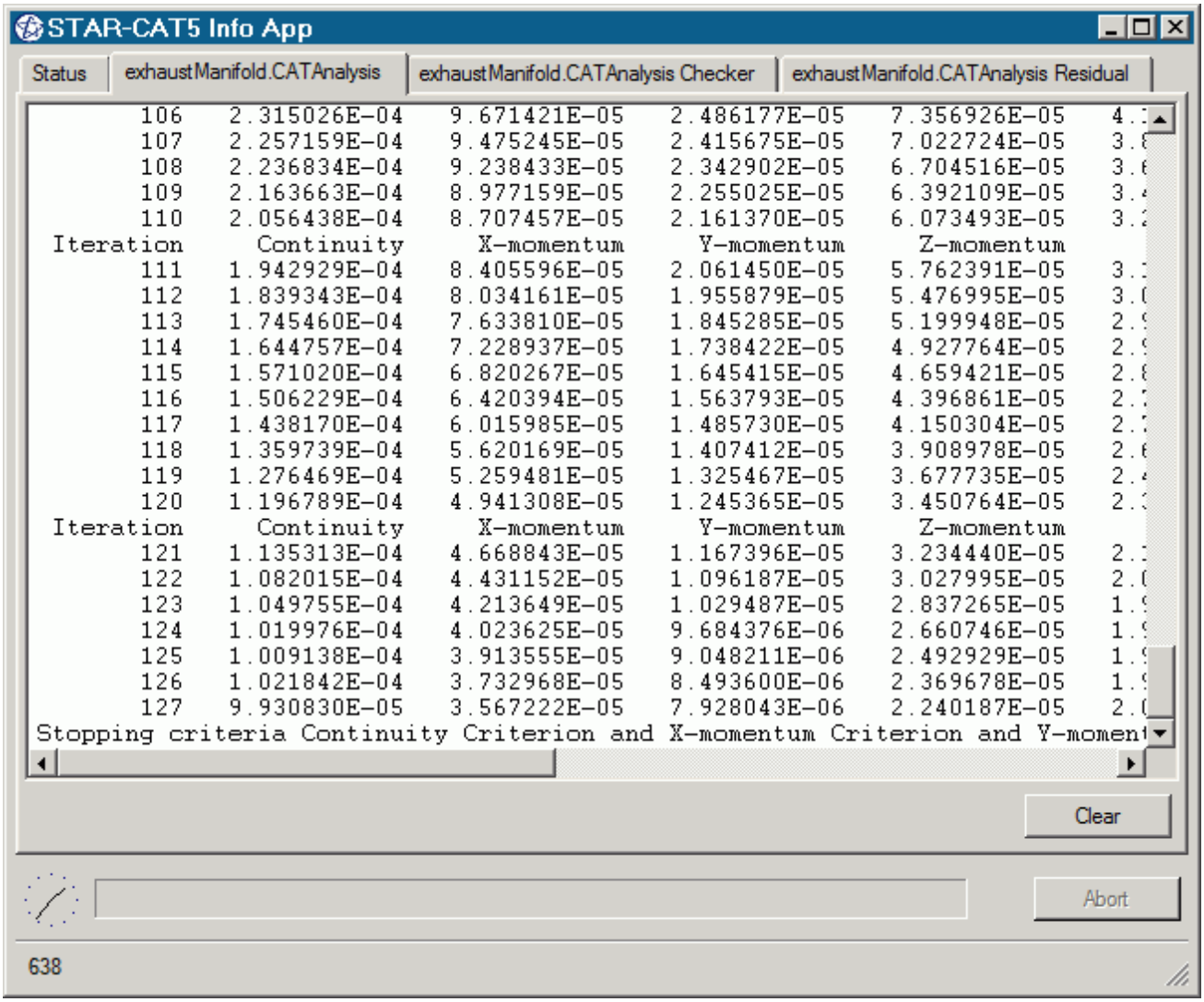
The Status Tab

The *Status* tab displays the licensing information for your current session, informing you whether the STAR-CAT5 license checked out successfully. The version of STAR-CAT5 is also displayed in this tab. This tab appears by default when you launch STAR-CAT5 without a geometry file.



The Output Tab

The *Output* tab appears once you create or open a geometry file in STAR-CAT5.



During the meshing process, this tab displays status information along with a brief report on the generated mesh. If errors were encountered during the meshing process, they are reported here with suggestions as to what can be causing them.

When the solver is running, the *Output* tab behaves like the output window in STAR-CCM+, displaying the data for each residual per iteration. When the solver finishes running, the same information is also displayed in the STAR-CCM+ *Output* window.

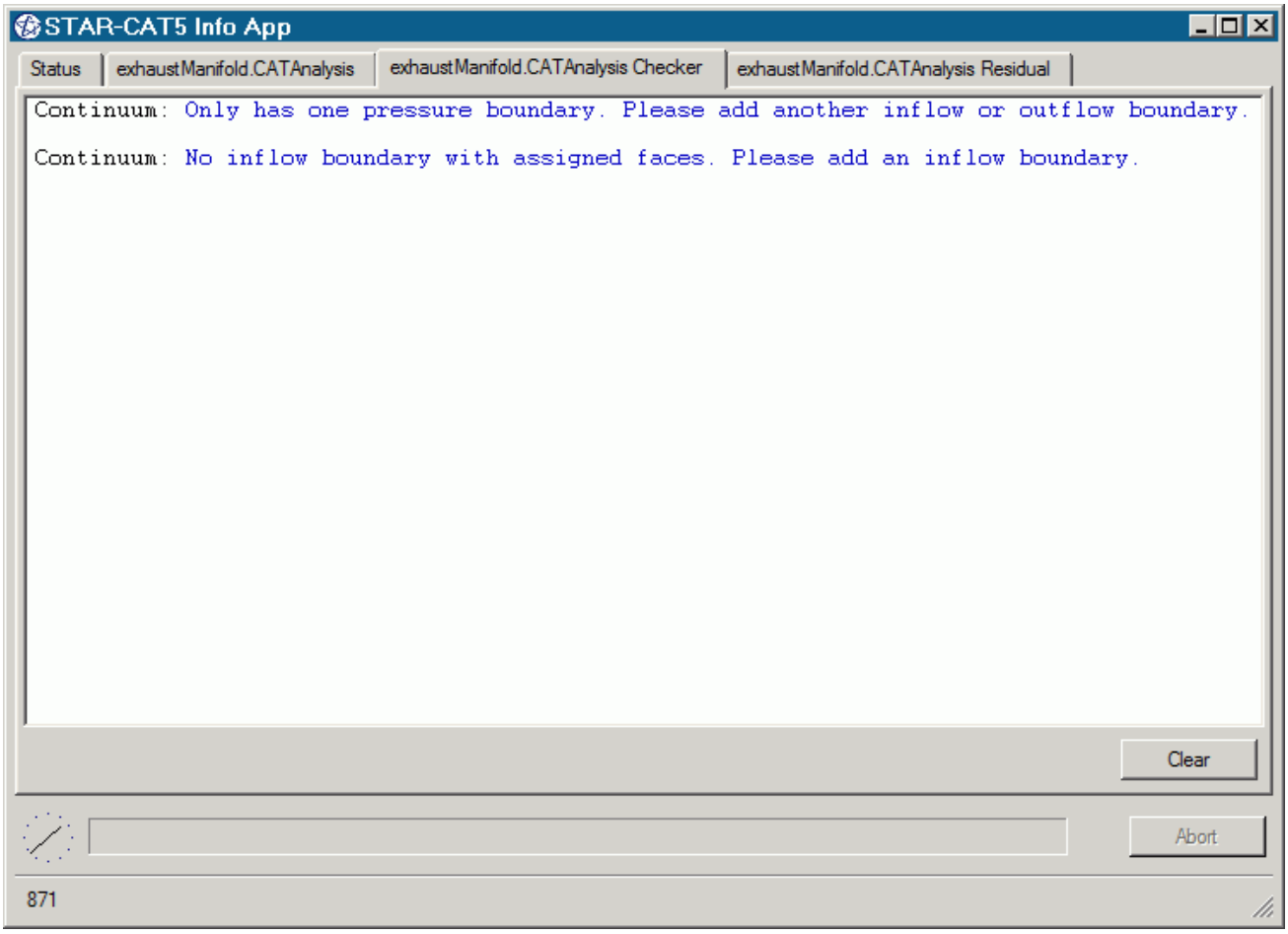
You can control the amount of information that is displayed in the *Output* tab by editing the verbosity in the `config.xml` file. See Modifying the Config.xml File.

To abort the mesher or solver, click the **Abort** button.

The Model Checker Tab

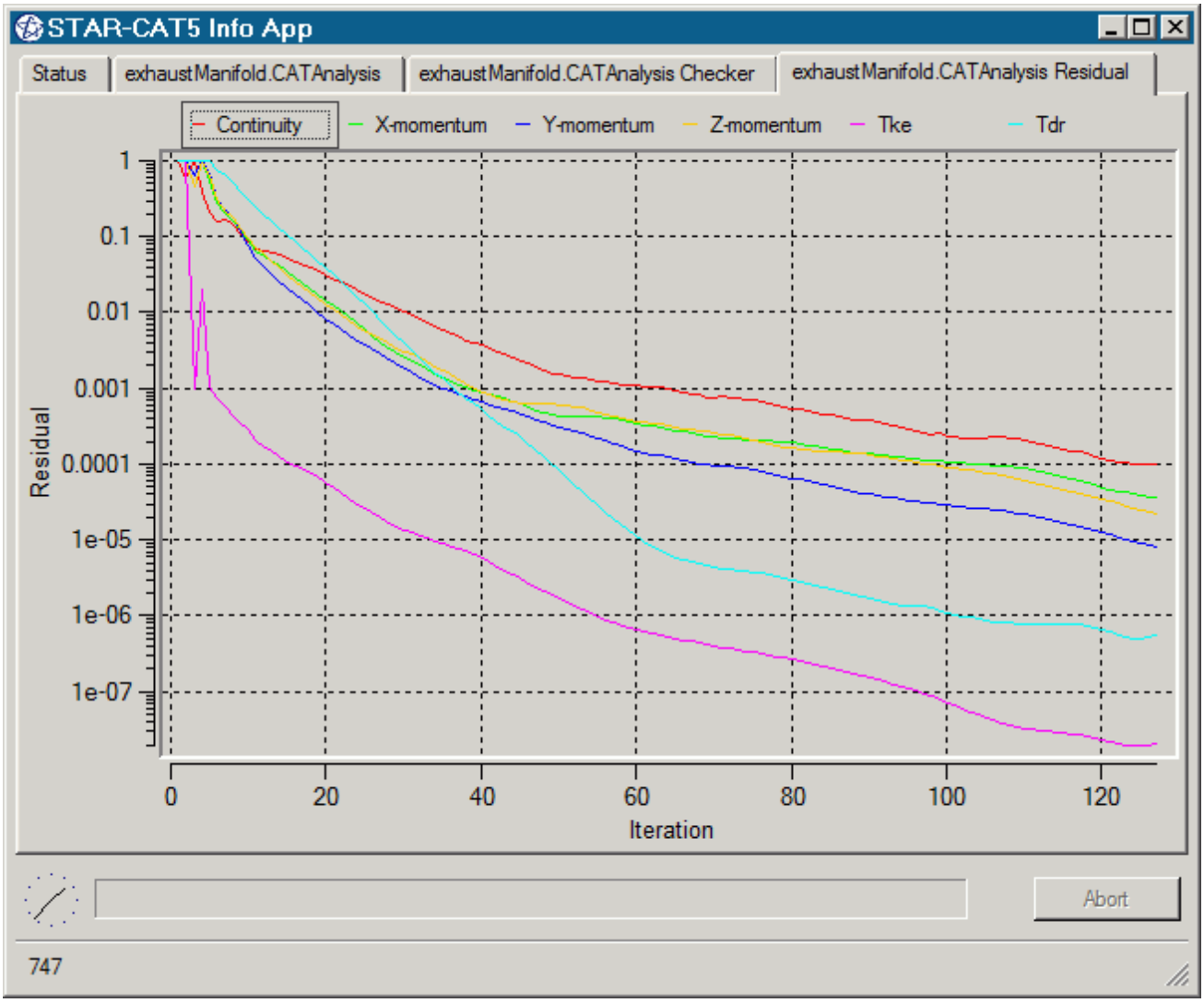
The *Model Checker* tab displays information from the live model checker, where the case setup is actively monitored for any problems. See Checking the Model for Errors.

The following example reports that the case has no inlet boundary defined.



The Residuals Tab

The *Residuals* tab appears when you launch the solver. Use the residuals to monitor the convergence of the solution being calculated.



Typically, a drop in three to five orders of magnitude in the residuals indicates that the solution has converged to an acceptable level. The residuals that are displayed in this tab also appear in STAR-CCM+ under the **Plots** node.

You can zoom in to the residuals by clicking and holding the left mouse button in the graph area and drawing a box around the area you want to view. To zoom out, right-click in the graph area. You can hide individual residuals from the *Residuals* tab by clicking their label at the top of the plot window.

Modifying the config.xml File

STAR-CCM+ requires a configuration file to enable it to find the appropriate CD-adapco components to run correctly. This configuration file is created during installation. An explanation of the file and settings within it is given here.

The configuration file is called `config.xml` and STAR-CAT5 searches for it in the location given by the `STARCADSERIES` environment variable.

Typical `config.xml` content is shown below. In this example STAR-CCM+ version 12.06.### is shown.

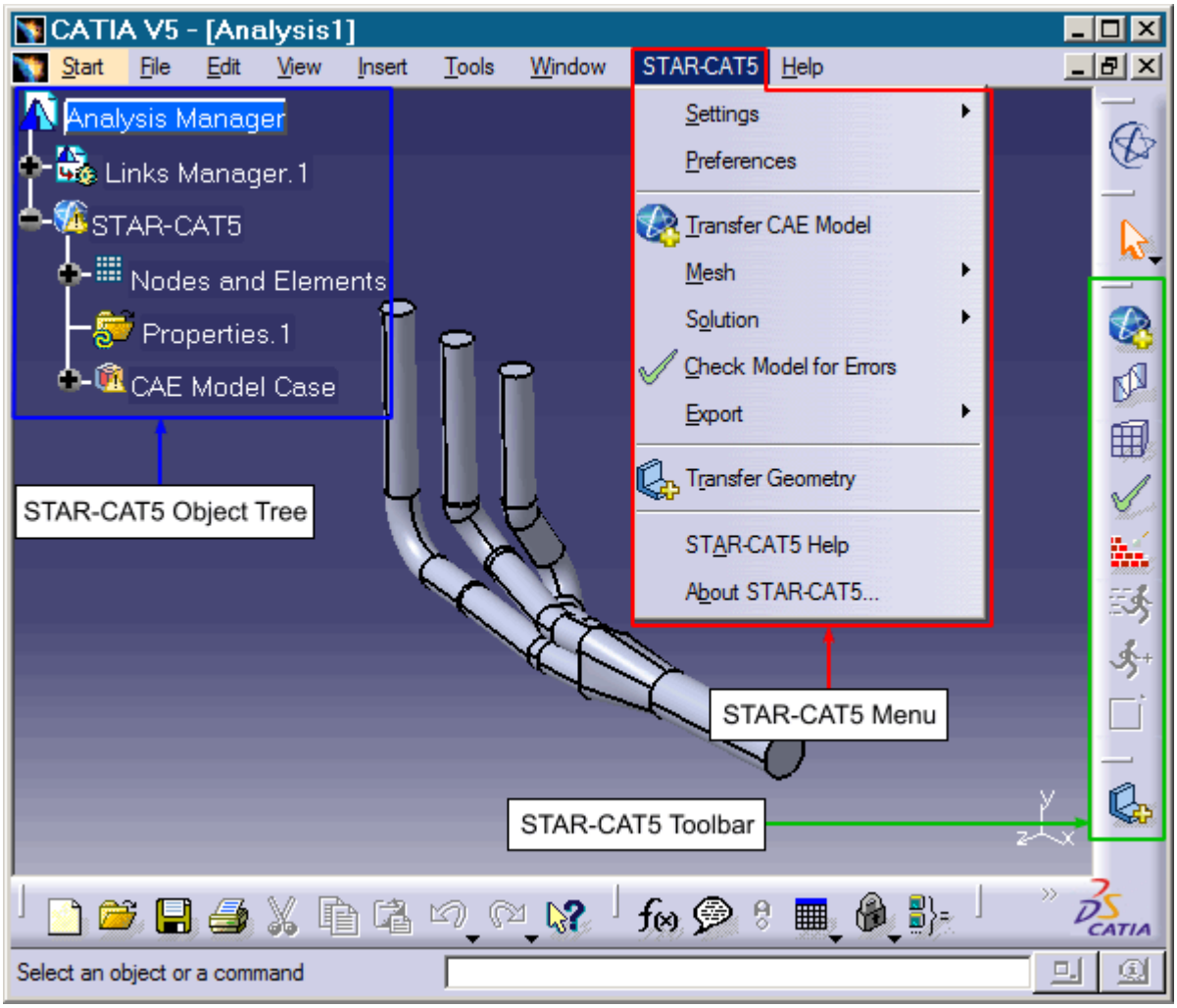
```
<CAD>
  <STAR_HOME>C:\Program Files\CD-adapco\12.06.###\STAR-CCM+12.06.###
\star</STAR_HOME>
  <SOLVER>C:\Program Files\CD-adapco\12.06.###\STAR-CCM+12.06.###\star
\bin\starccm+.exe</SOLVER>
  <LOGFILE>starcad.log</LOGFILE>
  <WWHELP>C:\Program Files\CD-adapco\12.06.###\STAR-CCM+12.06.###
\win64\Intel11.1\wwhelp.exe</WWHELP>
  <MANDIR>C:\Program Files\CD-adapco\12.06.###\STAR-CCM+12.06.###\doc
\en\online</MANDIR>
  <VERBOSITY>Normal</VERBOSITY> <!-- Can be Verbose, Log or Normal.
Default Normal -->
</CAD>
```

You can edit the following lines:

- `<SOLVER>` provides the location of the STAR-CCM+ solver executable. STAR-CCM+ licensing options are set in the *Preferences* dialog in the STAR-CAT5 user interface. See Setting Up Parallel Processing and Licensing.
- `<LOGFILE>` is the log file that will automatically be created and saved when STAR-CAT5 is run. A path of your choice can be used.
- `<VERBOSITY>` is the verbosity of the output window in the STAR-CAT5 Info App. This can be set to one of the following:
  - `Normal`—This is the default setting and will display the least amount of information in the *Output* panel.
  - `Verbose`—This will display a moderate amount of information in the *Output* panel.
  - `Log`—This will display all available information in the *Output* panel.

Understanding the STAR-CAT5 Interface

The STAR-CAT5 user interface is embedded within the CATIA environment, with the addition of the STAR-CAT5 object tree, menu, and toolbar.



The majority of the STAR-CAT5 operations are executed within the object tree and toolbar. Certain actions can also be carried out using multiple methods. A summary of the features available in each part of the STAR-CAT5 interface are detailed below.

Object Tree

The operations provided in the object tree include:

- Solution Controls
- Definition of Units System
- Continuum settings
- Region settings
- Add new boundary
- Boundary conditions
- Mesh Pipeline settings
- Add new mesh controls

Toolbar

The operations provided in the toolbar include:

- Transfer CAE Model
- Generate Surface Mesh
- Generate Volume Mesh
- Check Model for Errors
- Build and Run
- Run Solver
- Run For More Iterations
- Show/hide Mesh
- Design Parameters
- Transfer Geometry

Menu

The operations provided in the menu include:

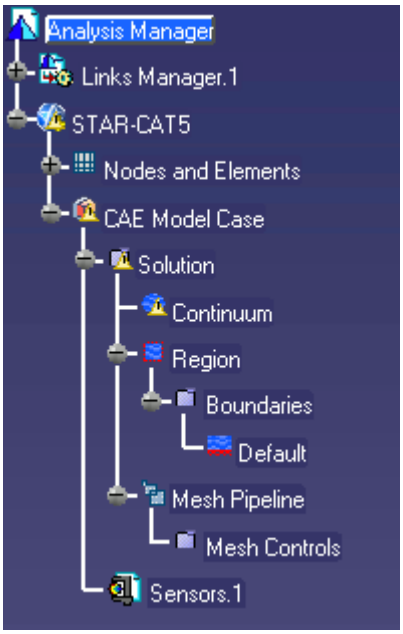
- Settings
  - Definition of Units System
  - Solution Controls
  - Clear All
- Preferences
- Transfer CAE Model
- Mesh
  - Generate Surface Mesh
  - Generate Volume Mesh
  - Clear Generated Meshes
  - Show/hide Mesh
- Solution
  - Build and Run
  - Run Solver
  - Run For More Iterations
  - Clear Solution
- Check Model for Errors
- Export
  - Surface Mesh (.dbs)
  - Volume Mesh (.ccm)
  - Engineering Report (.html, .csv)

- Simulation (.sim)
- Design Parameters
- Transfer Geometry
- STAR-Works Help
- About STAR-Works

The STAR-CAT5 Object Tree

STAR-CAT5 uses an object tree as the primary method for setting parameters.

The object tree consists of nodes, each one of which represents a particular component of the simulation. This approach to the user interface complements the method that is used by STAR-CCM+, allowing for an easy transition between the two software packages.



The object tree appears slightly different in assembly mode as it must accommodate multiple parts, but the principles of operation remain the same.

Nodes are arranged in an order that follows the recommended workflow for setting up a simulation in STAR-CAT5. At each stage, you typically right-click on a particular node in the tree and choose the appropriate action from the pop-up menu.

Certain functionality within STAR-CAT5 applies to the simulation as a whole, rather than to individual objects. These operations are provided in either the menu or toolbar.

A complete list of all functions and their respective location is included in Understanding the STAR-CAT5 Interface.

Setting Up Parallel Processing and Licensing

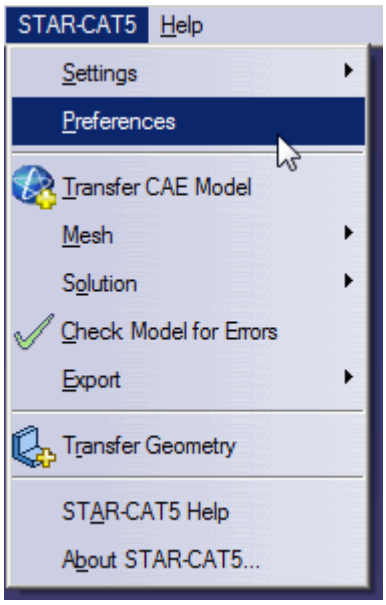
The licensing options and the options for setting up parallel processing are located in the *Preferences* dialog.

This dialog provides two sets of options: server preferences and server settings.

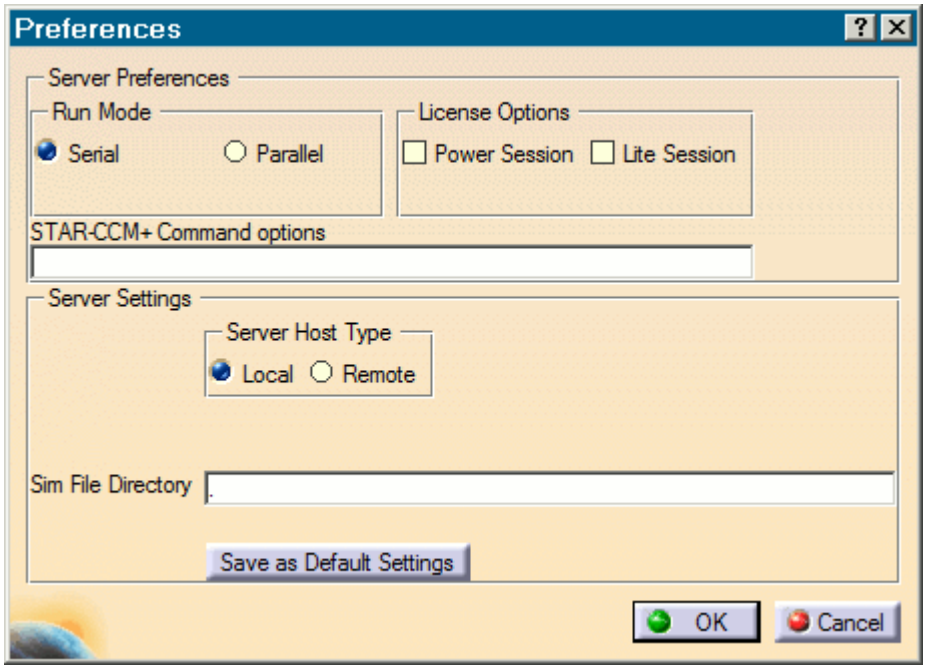
- Server preferences are applied whenever STAR-CAT5 starts a server.
- Server settings are saved in the CAD file and are applied only to that CAD file. Click **Save as Default Server Settings** to save the current server settings as the default settings to be applied to all CAD files.

To modify the run mode and licensing options:

1. Select **STAR-CAT5** > **Preferences** from the menu.



   The *Preferences* dialog appears.



2. Select the run mode:
   - For serial, set *Run Mode* to **Serial** (the default setting).
   - For parallel, set *Run Mode* to **Parallel** and *Compute Processes* to the desired number of cores.

     You can use the **up/down** buttons or you can type in the number of cores that you want to use.
   - For advanced licensing options, such as Power-on-Demand, enter the appropriate argument in the *Command* field.

     For Power-on-Demand, enter:

     ```
     -licpath 1999@flex.cd-adapco.com -podkey <pod_key>
     ```

     Where:
     - `1999@flex.cd-adapco.com` is the default CD-adapco Power-on-Demand server.
     - `<pod_key>` is the Power-on-Demand license key.

     For more information on Power-on-Demand licensing, see the section **User Guide** > **Using STAR-CCM+** > **STAR-CCM+ Licensing** > **Power-on-Demand Licensing**.

     A full list of command-line arguments is available in the STAR-CCM+ User Guide at **User Guide** > **Using STAR-CCM+** > **Command-Line Reference**.

   The *Command* field updates automatically to reflect the selections you make. Manual entries are not affected by these selections.
3. Select a licensing option:
   - To use a Power Session license, select **Power Session**.

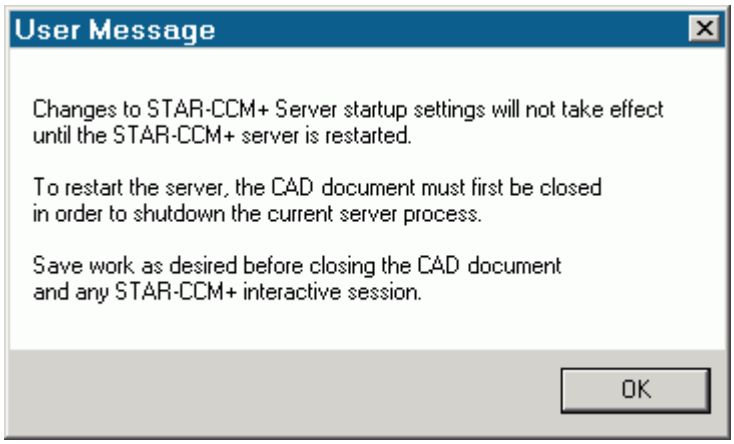
     This license is design for parallel simulations, where you can use an unlimited number of cores in this session.
   - To use a STAR-CCM+ Lite license, select **Lite Session**.

A limited set of STAR-CCM+ features are provided when you activate this option.

If you select no licensing option, the default CAD client license is used.

4. Click **OK**.

   If the STAR-CCM+ server is running when you modify the preferences, a message box appears informing you that both STAR-Works and STAR-CCM+ must be closed for the changes to take effect.



5. Click **OK** to proceed.
6. Close the active STAR-CAT5 and STAR-CCM+ sessions.
7. Restart STAR-CAT5.

Note: The STAR-CCM+ server only starts when it is required, such as during surface meshing or volume meshing. It does not start when you launch STAR-CAT5.

If an option in the *Preferences* dialog conflicts with an option in the `config.xml` file, the option that is set in the *Preferences* dialog takes precedence. An example of such a situation is if the *Compute Processes* property is set to `4` in the *Preferences* panel, while it is set to `2` in the `config.xml` file; the simulation runs using four processors. See Modifying the config.xml File

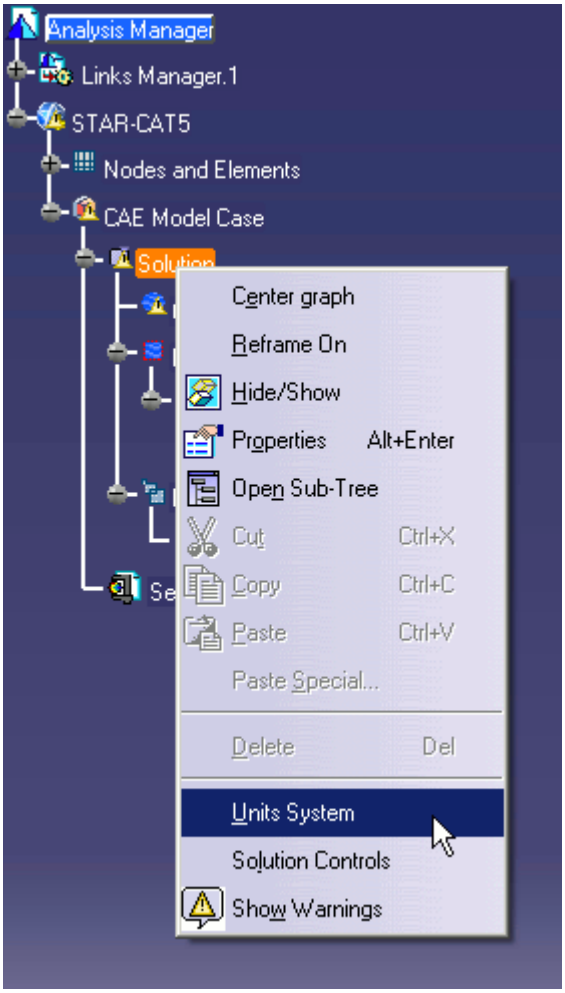
Note: IBM Platform MPI must be installed on your local machine for you to run in parallel. This program is included with the STAR-CCM+ package installer.

Changing the Units System

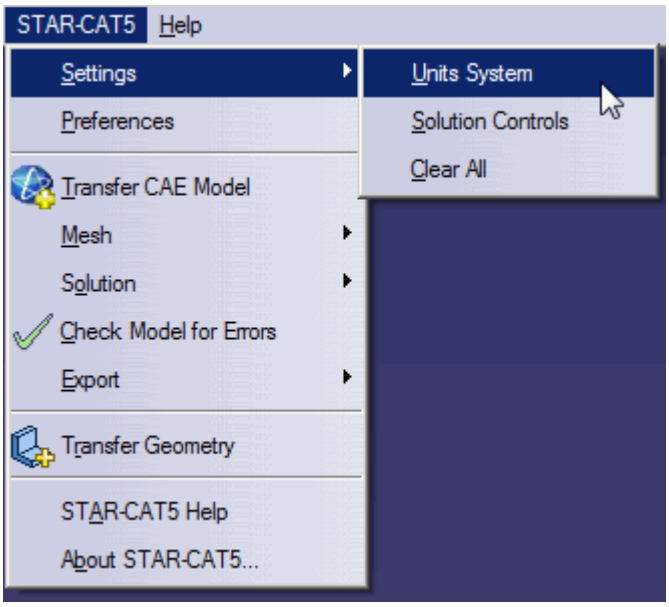
You can use either the SI (International Standard) or USCS (United States Customary System) units system in STAR-CAT5.

To change the units system:

1. Bring up the *Choose Units System* dialog:
   - Right-click the **STAR-CAT5** > **CAE Model Case** > **Solution** node within the object tree and select **Units System**.

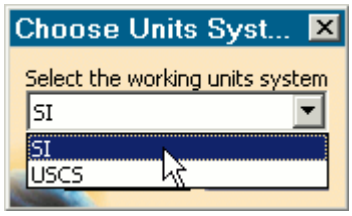


- Alternatively, from the menu, select STAR-Works**Settings** > **Units System**.



The *Choose Units System* dialog appears.

2. Select the desired units system from the drop-down menu.



3. Click **OK**.

Automatic Unit Conversion in Dialog Panels

Changing the default units system converts existing values to the new units system. Ensure that you select the desired units system before you enter the value, otherwise you will need to re-enter it.

Note: Specific values that you have already modified the units on are not affected by a change of units system.

Transferring Units to STAR-CCM+

Units that are used in the CAD geometry file are also used in the STAR-CCM+ simulation. Less common units, such as those listed below, are not supported in STAR-CCM+:

- Micrometers
- Angstroms
- Nanometers
- Mils
- Microinches
- Footinches (this unit may be converted to inches in STAR-CCM+)
- Miles

Note: This list may not be exhaustive and is subject to change.

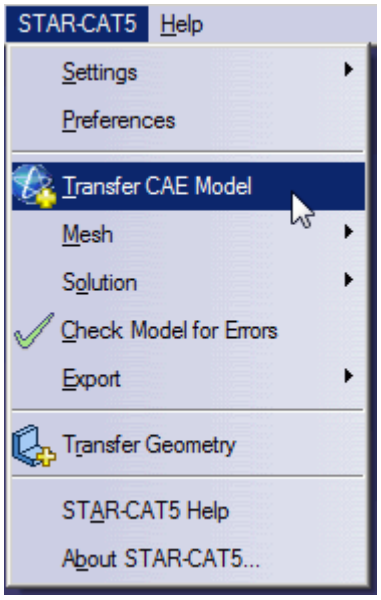
These units are converted to meters when the CAD geometry is imported into STAR-CCM+.

Transferring the CAE Model to STAR-CCM+

You can transfer your geometry and the simulation setup to STAR-CCM+ at any time from within the STAR-CAT5 environment. If you want to transfer only the geometry to STAR-CCM+, refer to Transferring Geometry to STAR-CCM+.

To transfer the geometry with the simulation set up:

1. From the STAR-CAT5 menu, select **Transfer CAE Model**.



The version of STAR-CCM+ that is affiliated with STAR-CAT5 launches.

Note: A surface mesh is required in order to visualize the geometry in STAR-CCM+. If you have not generated a mesh before clicking **Transfer CAE Model**, the geometry is tessellated with a medium density.

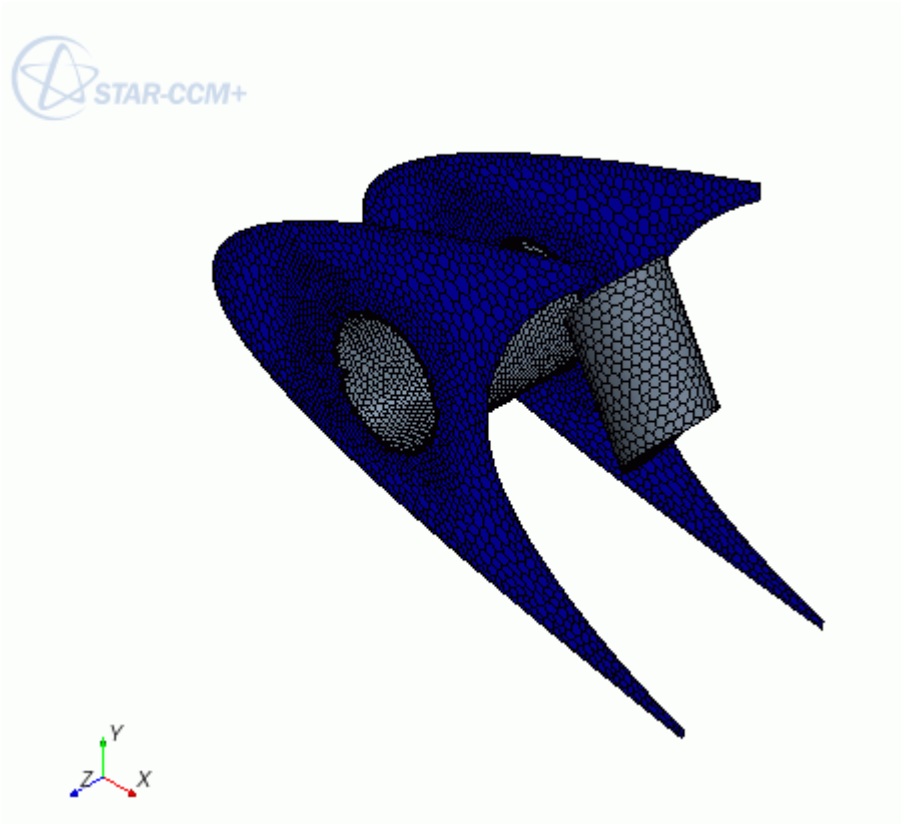
Editing STAR-CAT5 Settings While the STAR-CCM+ Client is Running

If you make a change in STAR-CAT5 while the STAR-CCM+ client is running, such as adding or deleting a boundary, you may need to update the volume mesh.

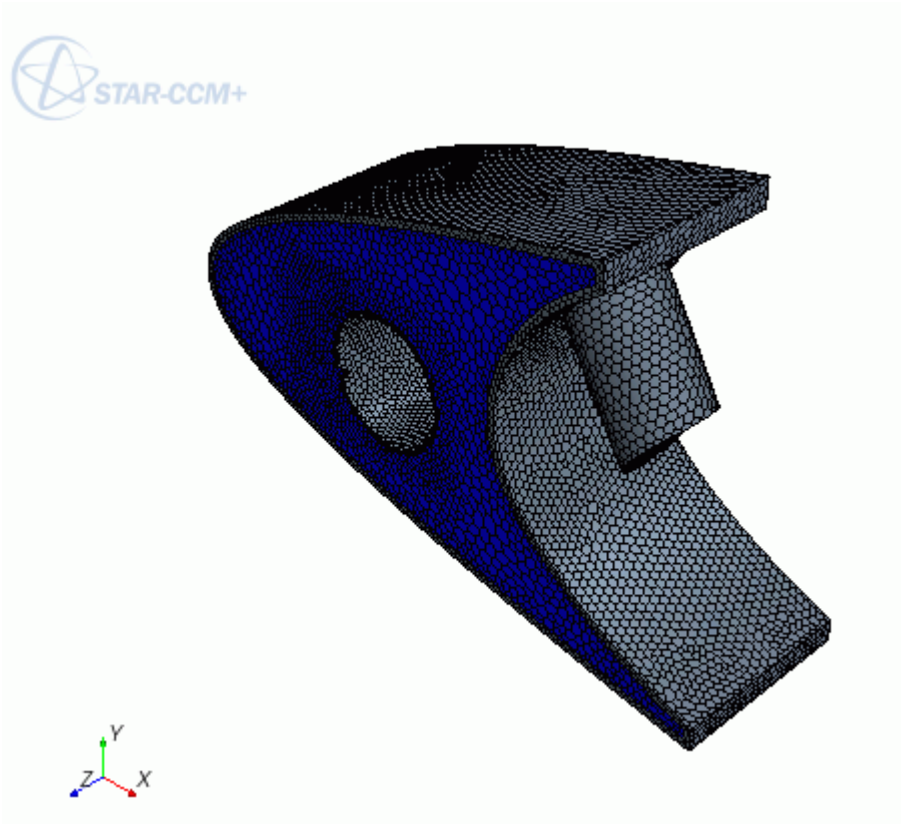
Scenes in the STAR-CCM+ session will not display the model correctly unless you keep the volume mesh up to date. A list of changes that require you to re-mesh the geometry is provided below:

- For boundaries and regions: changing the type, adding or deleting
- For mesh pipelines and mesh controls: editing, adding or deleting

When adding or removing regions and boundaries in STAR-CAT5, the scenes in STAR-CCM+ may be missing some parts, as shown below, even after re-meshing the geometry.



To update the scene, you need to manually add the modified regions and boundaries to the scene in the STAR-CCM+ client. You can do this by dragging and dropping the region or part from the STAR-CCM+ object tree, or by adding the region or part using the Part Selector. An alternative method would be to create a new scene, as this will automatically pick up the new regions and boundaries added to the simulation.



Common Functions

The following section details common operations used within the STAR-CAT5 environment.

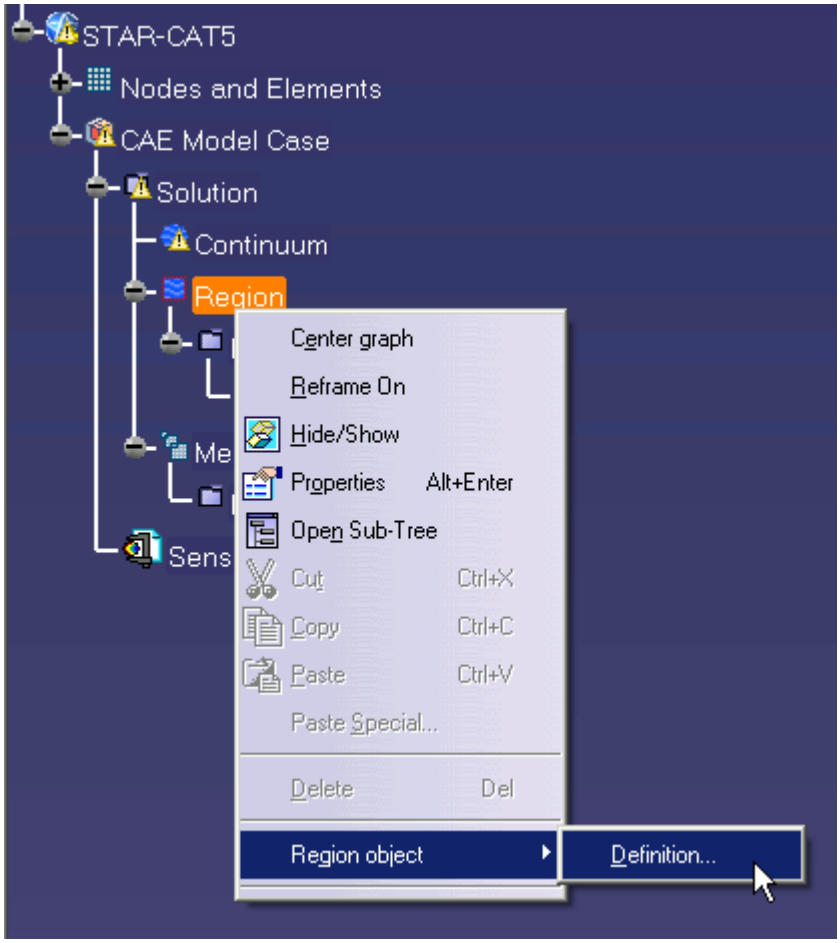
Contents:

Renaming an Object
Deleting an Object
Editing An Object
Assigning Surfaces and Parts to Objects in STAR-CAT5
Removing Surfaces and Parts from Objects in STAR-CAT5

Renaming an Object

To rename an object:

1. Right-click the object node in the object tree, and select **[entity]** > **[object]** > **Definition...**



The settings dialog for the object appears.

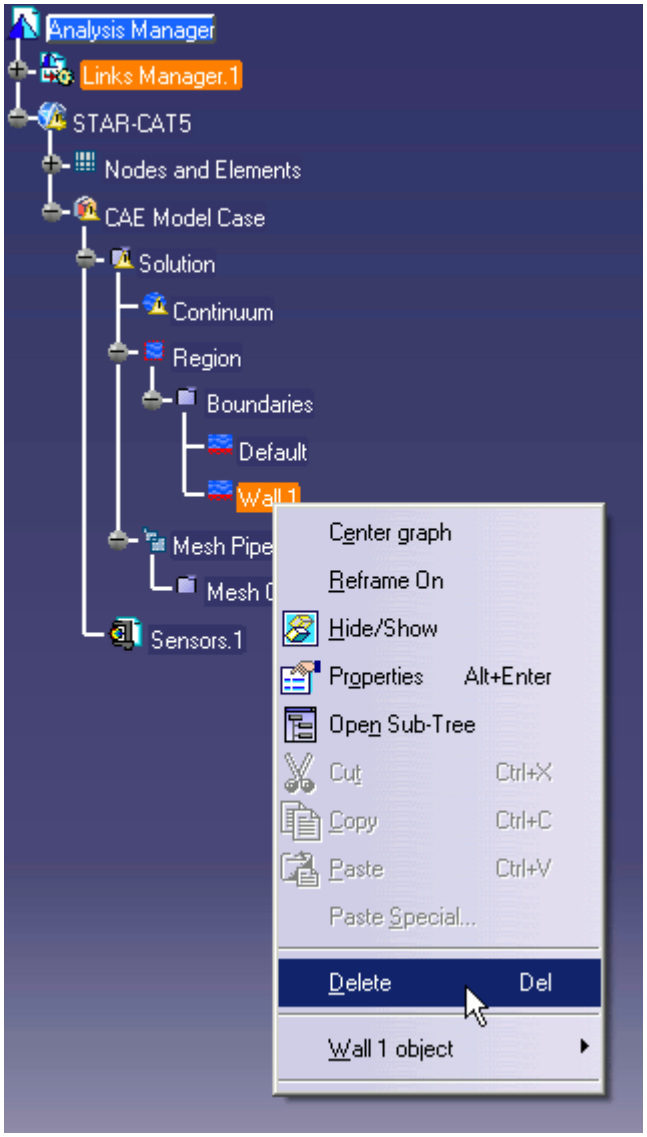
2. Enter the name in the **Name** box.
3. Click **OK** to close the dialog.

Note: You can also assign entity names at the point where you create them, for example, when adding boundaries, continua, regions, mesh pipelines and mesh controls. When adding a new object, locate the **Name** text box in the settings dialog and enter the required name.

Deleting an Object

To delete an object:

1. Right-click the object node within the object tree and select **Delete**.



To delete multiple entities:

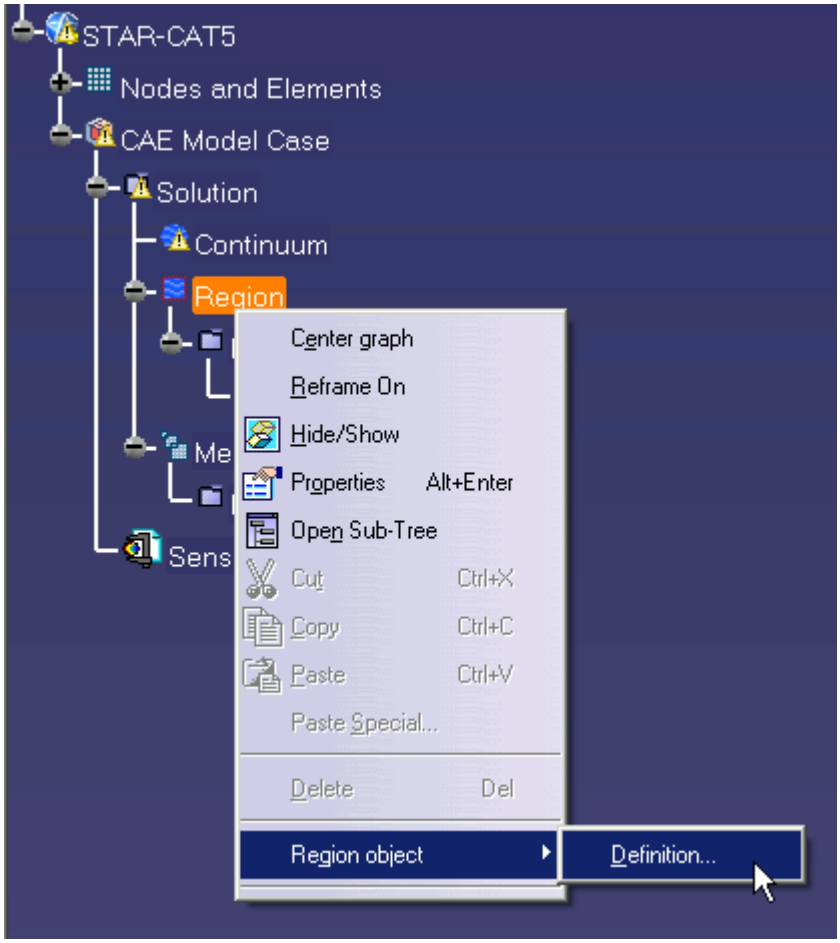
2. Select the desired nodes within the object tree (by holding down the `<Ctrl>` key), and right-click on any one of the selected nodes, then select **Delete** from the pop-up menu.

Note: Some combinations of selections cannot be deleted together using this method. In these cases, the **Delete** option is disabled in the pop-up menu.

Editing An Object

To edit an object:

1. Right-click on the desired node in the object tree and select **entity** > **object** > **Definition...**.



The settings dialog for the object appears.

2. Make the changes that you want.
3. Click **OK**.

Assigning Surfaces and Parts to Objects in STAR-CAT5

Regions, boundaries, mesh pipelines, and mesh controls are all represented as objects on the STAR-CAT5 object tree. You can assign parts and surfaces to these objects when their property dialog is open.

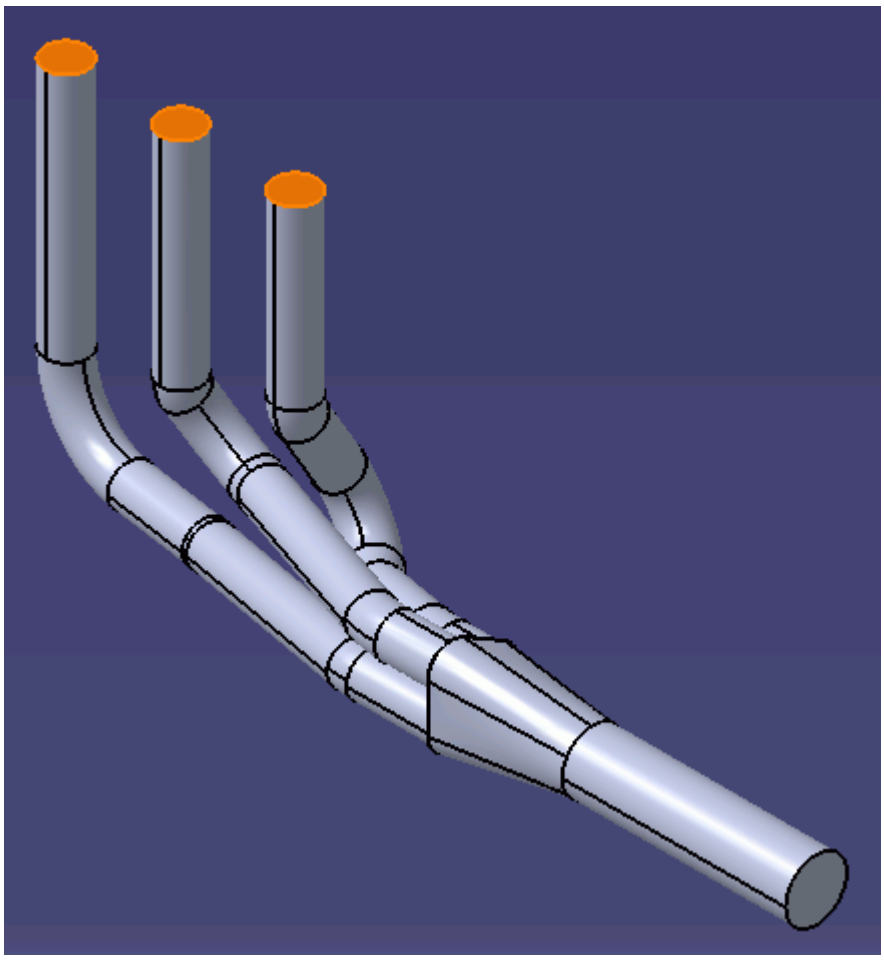
Assigning Surfaces to Objects

To assign a surface to an object (when its property dialog is open):

1. On the CAD geometry, click the surfaces that you want to assign to the object. You can add multiple surfaces by clicking more surfaces on the CAD geometry.

   Note: If you set the render style to perspective, **View** > **Render Style** > **Perspective**, you can zoom inside solid geometries and select hidden/internal surfaces.

   The selected surfaces are highlighted on the CAD geometry. By default, CATIA only highlights the edges of selected surfaces. This behavior can be changed so that the surface is highlighted as well, as is shown in the following screenshot.



The number of selected surfaces are displayed in the *Supports* box of the active dialog.

Parts that are assigned to other regions are automatically hidden on the CAD geometry.

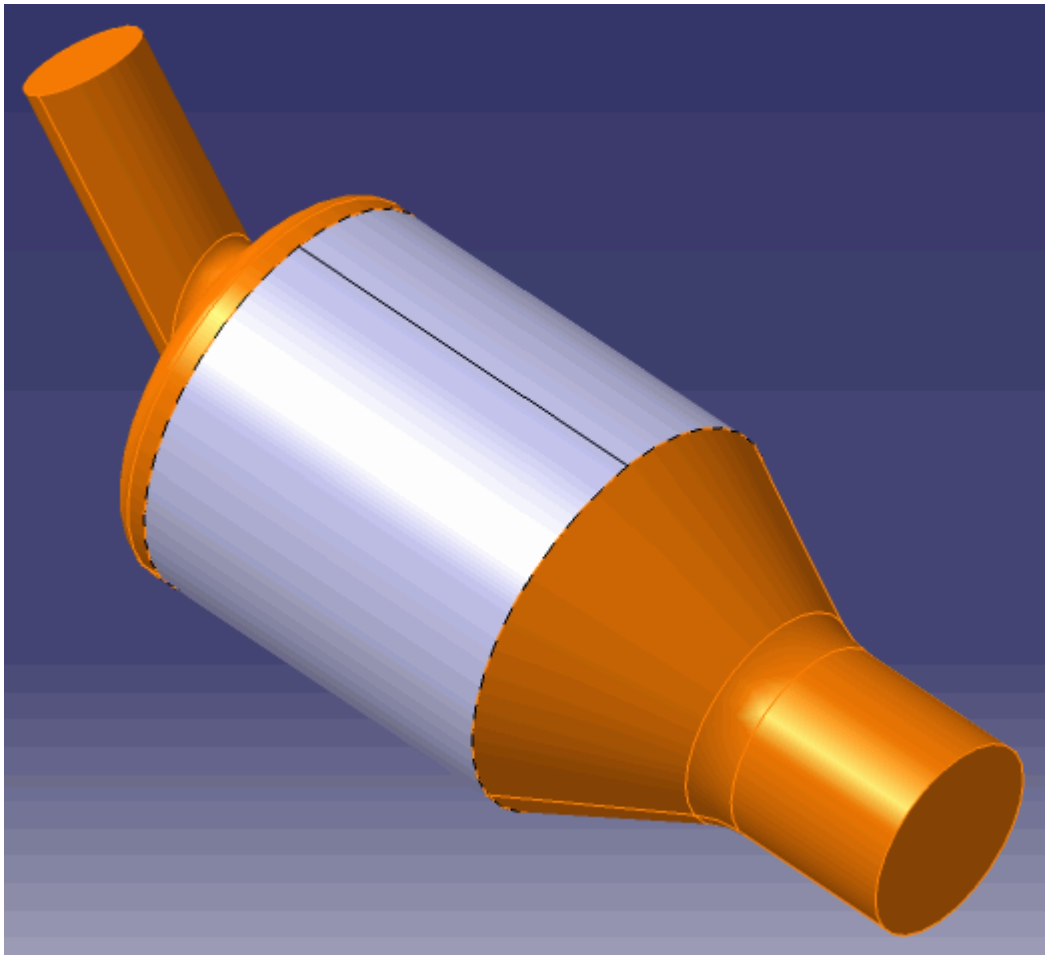
Assigning Parts to Objects

To assign a part to an object (when its property dialog is open):

2. On the CAD geometry, click the part that you want to assign to the object. You can add multiple parts by clicking more parts on the CAD geometry.

   Note: If you set the render style to perspective, **View** > **Render Style** > **Perspective**, you can zoom inside solid geometries and select hidden/internal parts.

   The selected parts are highlighted on the CAD geometry. By default, CATIA only highlights the edges of selected parts. This behavior can be changed so that the surfaces are highlighted as well, as is shown in the following screenshot.



The number of selected parts are displayed in the *Supports* box of the active dialog.

Parts that are assigned to other regions are automatically hidden on the CAD geometry.

Removing Surfaces and Parts from Objects in STAR-CAT5

Regions, boundaries, mesh pipelines, and mesh controls are all represented as objects on the STAR-CAT5 object tree. You can remove parts and surfaces from these objects when their property dialog is open.

A surface that is removed from a boundary is returned to the default boundary. Similarly, removing a part from a region returns it to the default region.

Removing Surfaces from Objects

To remove a surface from an object (when its property dialog is open):

1. On the CAD geometry, click the face that you want to remove.

Removed surfaces are no longer highlighted on the CAD geometry. Additionally, you can use the *Supports* box to monitor how many surfaces are currently assigned to a particular object.

Removing Parts from Objects

To remove a part from an object (when its property dialog is open):

2. On the CAD geometry, click the part that you want to remove.

Removed parts are no longer highlighted on the CAD geometry. Additionally, you can use the *Supports* box to monitor how many parts are currently assigned to a particular object.

Running STAR-CAT5 and STAR-CCM+ on Different Machines

You can run STAR-CAT5 and STAR-CCM+ on two different machines.

STAR-CCM+ operates on a client-server architecture, where STAR-CCM+ is the server and STAR-CAT5 is the client. A benefit of client-server architectures is that the client and the server do not have to be running on the same machine. Typically, CAD packages run on Windows operating systems. You can choose to run STAR-CCM+ on any remote machine with a supported or certified platform, including Linux platforms.

Contents:

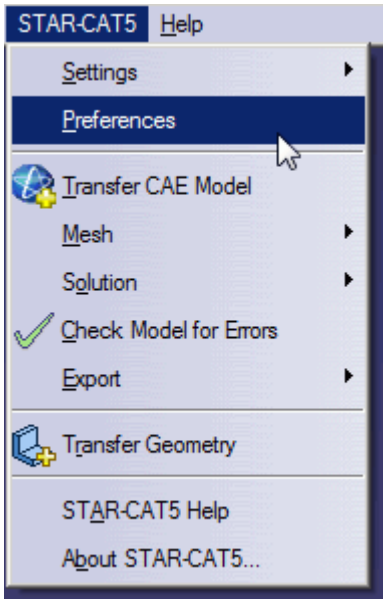
Setting Up STAR-CCM+ to Run on a Remote Machine with STAR-CAT5 on the Local Machine

Setting Up STAR-CCM+ to Run on a Remote Machine with STAR-CAT5 on the Local Machine

STAR-CAT5 and STAR-CCM+ can run on two different machines.

To set up STAR-CCM+ to run on a remote machine:

1. From the STAR-CAT5 menu, select **Preferences**.



2. In the *Preferences* dialog, set *Server Host Type* to **Remote**.

   The options for the remote server appear.
3. Set the remaining server settings to reflect your situation. See Server Settings Panel Reference.
4. If you want to store these settings for the next time you launch STAR-CAT5, click **Save as Default Settings**.
5. Click **OK**.

Contents:

Server Settings Panel Reference

Server Settings Panel Reference

The Server Settings panel that is located in the Preferences dialog allow you to run STAR-CAT5 on a remote machine.

Property	Description	
Server Host Type	Specifies whether to run STAR-CCM+ on the local machine or on a remote machine.	
	Local	STAR-CCM+ runs on the local machine.
	Remote	STAR-CCM+ runs on a remote machine
SSH Command	Specifies the absolute path to the SSH installation directory.	
Server Host	Specifies the name of the remote host where STAR-CCM+ is installed. This option can contain the IP address, name, or full name of the host.	
Sim File Directory	Specifies the absolute path to the directory where you want to store the simulation files on the remote machine. When in **Local** mode, the simulation	

Property	Description
	files are saved in the same directory as the geometry file. You can specify a custom directory.
Server Executable	Specifies the absolute path to the STAR-CCM+ executable on the remote machine.

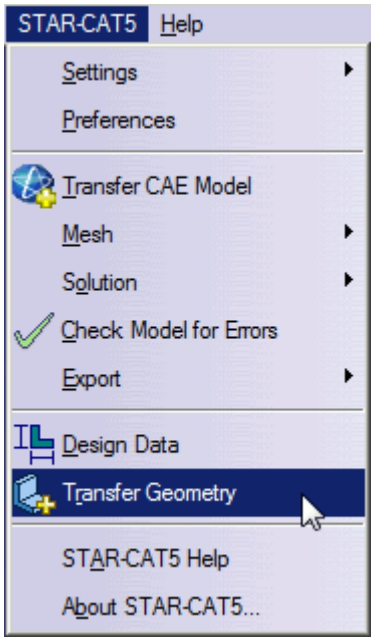
Transferring Geometry to STAR-CCM+

With STAR-CAT5, you can transfer your CAD geometry directly to STAR-CCM+ from within the CATIA environment.

The Transfer Geometry feature imports your geometry into STAR-CCM+. This function provides an alternative method to the CAD translator add-ons for importing geometry. This method of transferring geometry can provide a faster workflow when performing a design analysis.

To transfer geometry to STAR-CCM+:

1. From the STAR-CAT5 menu, select **Transfer Geometry**.



   STAR-CCM+ launches and imports the CAD geometry. You are now at the same stage in the workflow as if you imported the surface mesh directly into STAR-CCM+ using the CAD translators (**File** > **Import** > **Import Surface Mesh...**).

2. Set up your case in STAR-CCM+.

Caution: Do not click **Transfer CAE Model** if you have previously used the Transfer Geometry method and set up your case in STAR-CCM+. This action results in the loss of your settings in STAR-CCM+.

Contents:

Saving Simulation Files
Transferring Design Data to STAR-CCM+
Modifying Geometry from Within STAR-CCM+
Transferring CATIA Publications to STAR-CCM+
Controlling How the Geometry is Transferred to STAR-CCM+

Geometry Transfer Settings Dialog

Saving Simulation Files

After transferring the geometry to STAR-CCM+, you can save the simulation file and retain the associativity with STAR-CAT5.

To save your simulation files and keep the associativity:

- Save your CAD geometry in STAR-CAT5.

  Note: You can also safely rename and move the file using this method.

or

- Save your simulation file in STAR-CCM+.

  Note: Do not rename the simulation file when using this method, as it will lose the associativity with the CATIA file.

Resuming a Simulation

To resume a simulation after closing STAR-CAT5 and STAR-CCM+:

1. Launch STAR-CAT5.
2. Open the CAD geometry in STAR-CAT5.
3. Click **Transfer Geometry**.

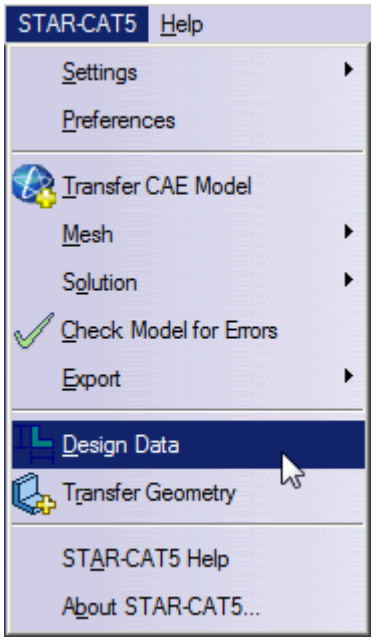
Transferring Design Data to STAR-CCM+

Design data consists of design parameters, coordinate systems, and reference planes that you can transfer to STAR-CCM+ along with the geometry.

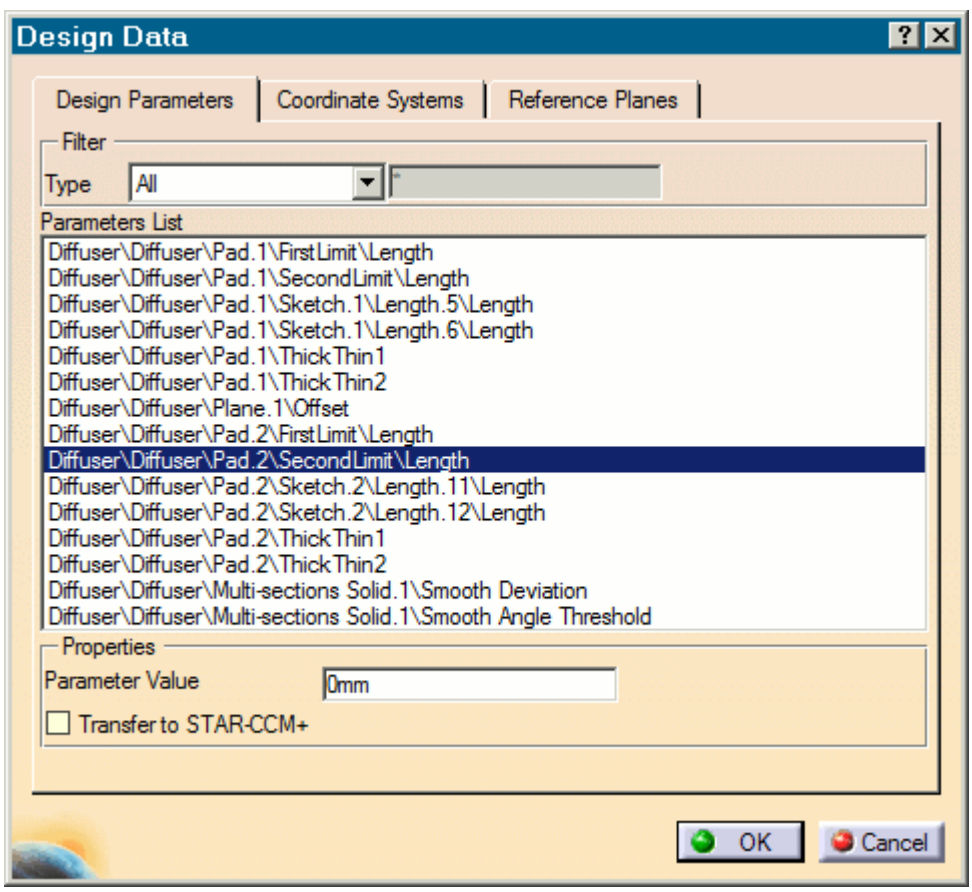
Design parameters control the feature dimensions, for example extrusion depth, and are useful for modifying the geometry in STAR-CCM+ after running the analysis. You can use reference planes in scalar scenes and reports in STAR-CCM+. You can use coordinate systems to define motion in the STAR-CCM+ simulation.

To transfer design data to STAR-CCM+:

1. From the STAR-CAT5 menu, click **Design Data**.



The *Design Data* dialog appears.



2. Select the design data to transfer:

- **Design Parameters**—Click the *Design Parameters* tab and select the design parameters that you want to transfer from the list. You can select multiple items at once.

  Note: STAR-CAT5 only recognizes native CAD features as design parameters. Only recognized design parameters are displayed in the list.

- **Coordinate Systems**—Click the *Coordinate Systems* tab and select the coordinate systems that you want to transfer from the list. You can select multiple items at once.
- **Reference Planes**—Click the *Reference Planes* tab and select the planes that you want to transfer from the list. You can select multiple items at once.

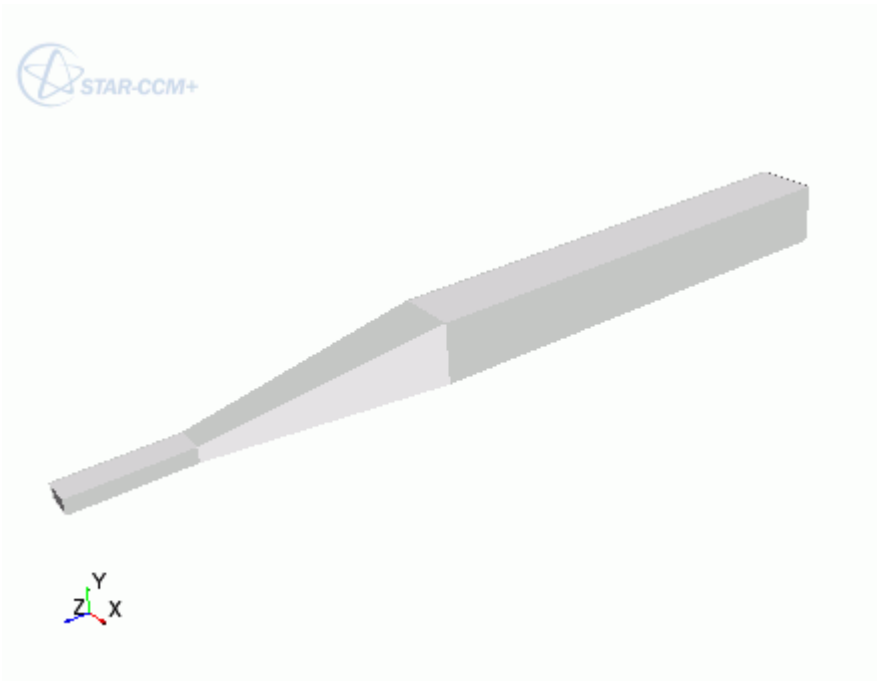
3. In each tab, activate the *Transfer to STAR-CCM+* option.
4. Click **OK**.
5. Transfer the geometry to STAR-CCM+. See Transferring Geometry to STAR-CCM+.

The geometry, along with the specified design data, is transferred to STAR-CCM+. Transferred design parameters are placed in the **Geometry** > **CAD Client Models** > **[geometry_name]** > **Design Parameters** node. Transferred planes are placed in the **Derived Parts** node in STAR-CCM+. Transferred coordinates systems are placed in the **Tools** > **Coordinate Systems** > **Laboratory** > **Local Coordinate Systems** node in STAR-CCM+.

Modifying Geometry from Within STAR-CCM+

Once you transfer a design parameter to STAR-CCM+, you can modify the geometry from within STAR-CCM+ with respect to that parameter. This feature is useful for quickly modifying a geometry after performing an analysis in STAR-CCM+ without needing to return to the original CAD software.

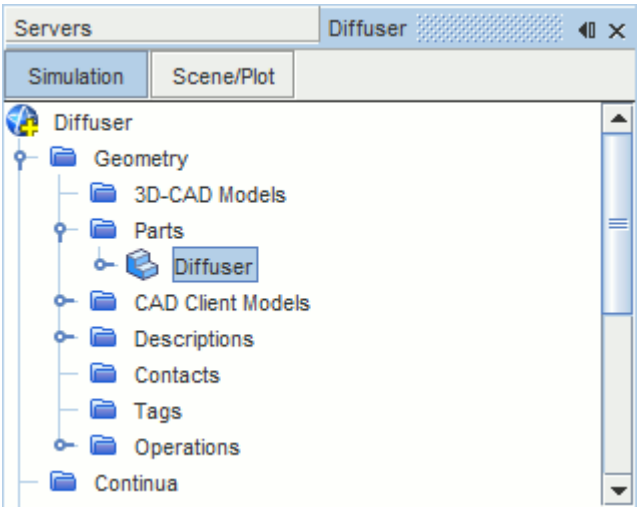
In the following example, the geometry of a diffuser is modified.



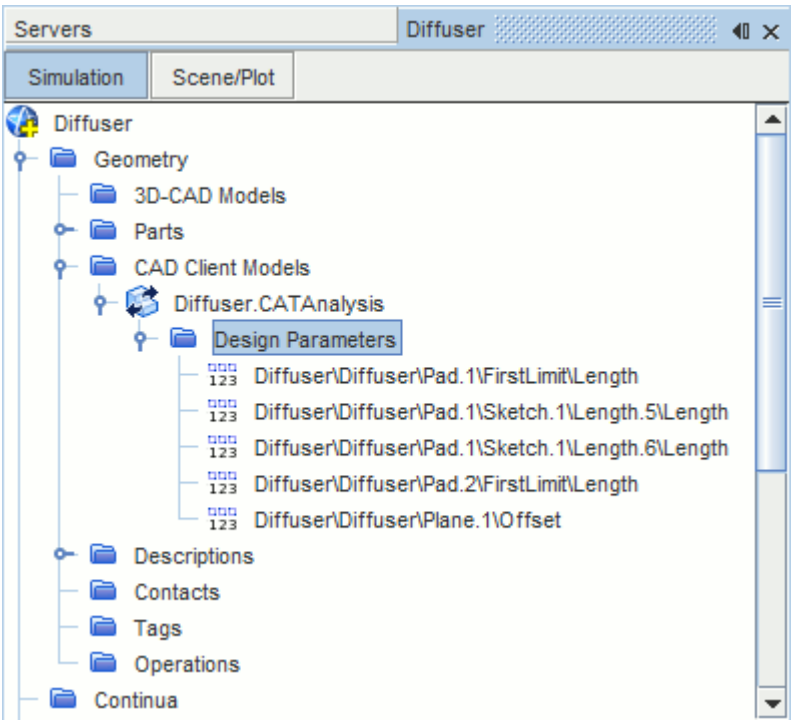
To modify a geometry in STAR-CCM+:

1. In STAR-CAT5, select the design parameters that you want to transfer to STAR-CCM+.
2. Transfer the geometry to STAR-CCM+. See Transferring Geometry to STAR-CCM+.

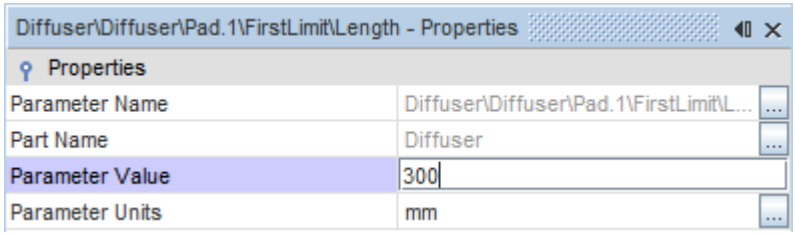
   Transferred geometries are located under the **Geometry** > **Parts** node.



   Transferred design parameters are located under the **Geometry** > **CAD Client Models** > **[part_name]** > **Design Parameters** node.

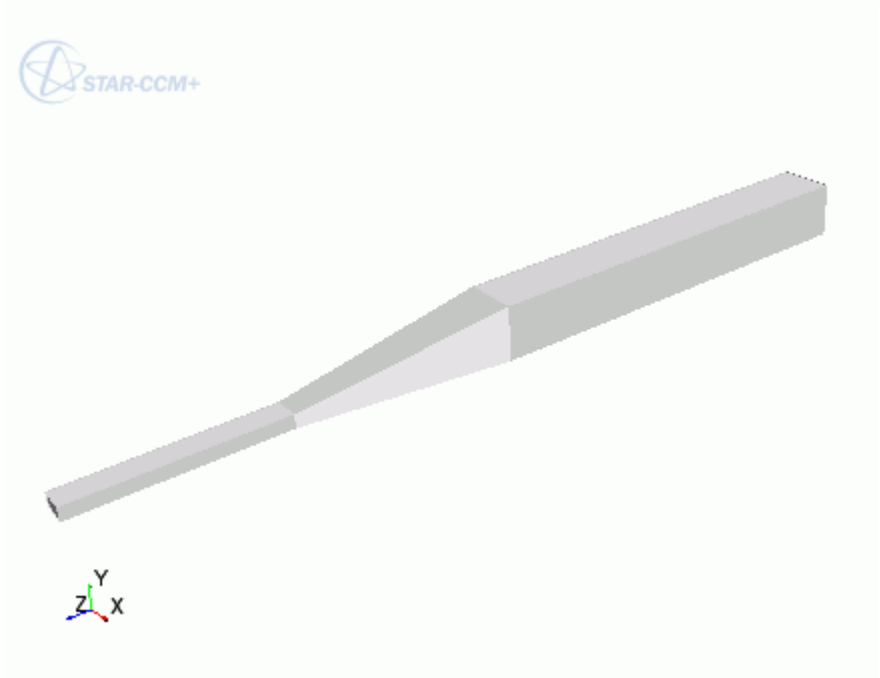


3. In STAR-CAT5, save the geometry file. The simulation file is also saved.
4. In STAR-CCM+, select the design parameter that you want to modify and set *Parameter Value* to the desire value.



5. Update the part in STAR-CCM+:
   - To update the part and save the updated parameters in the CAD file, right-click **Geometry** > **CAD Client Models** > **[part_name]** and select **Update Model and Save the CAD File**.
   - To update the part without updating the parameters in the CAD file, right-click **Geometry** > **CAD Client Models** > **[part_name]** and select **Update Model**.

   The geometry updates according to the design parameter changes.



6. Return to CATIA. The changes to the geometry are transferred back to the native CAD geometry.

This workflow maintains an active link between STAR-CAT5 and STAR-CCM+. When working with Geometry Transfer, you can modify the geometry in either package and the geometry in the relevant package is updated accordingly.

Updating the Geometry in STAR-CCM+ with CATIA Closed

You can close CATIA and continue working with the simulation in STAR-CCM+. Doing so, however, breaks the link between the two software packages. Any further updates that you make to the geometry in STAR-CCM+ are not transferred back to the geometry in CATIA. Instead, these changes are stored in memory, which allows you to include the design parameters in your optimization scripts.

This workflow also allows you to release the CATIA and STAR-CAT5 licenses when performing tasks in STAR-CCM+. In this case, the CATIA and STAR-CAT5 licenses are temporarily checked out only when the updates to the geometry are stored in memory.

Transferring the geometry from STAR-CAT5 will overwrite the geometry in STAR-CCM+.

Transferring CATIA Publications to STAR-CCM+

You can transfer published entity names from CATIA to STAR-CCM+ using the Transfer Geometry feature of STAR-CAT5.

When you transfer geometry that contains CATIA publication data, the published names are transferred instead of the default entity name.

You can transfer the published names for the following entities:

- *Surfaces*—by default, surface names are extracted from the last feature that created them, for example `Pad.1`. Transferring the published name allows you to control the names of the surfaces.
- *Design Parameters*—by default, design parameters are transferred with a generic name that describes what they are, for example `Length.19`. If the design parameter is published, then the name of the published design parameter is transferred to STAR-CCM+. In STAR-CCM+, design parameter names are concatenated with the part or assembly name that they come from, for example `Part\Design Parameter`.
- *Features*—by default, the feature name is transferred to STAR-CCM+. If the feature contains published data, then this name is transferred instead.
- *Bodies*—by default, body names are not used in STAR-CCM+. When a part contains more than one body, the bodies are transferred to STAR-CCM+ as parts inside a composite part. The published name of these parts will be transferred to STAR-CCM+.

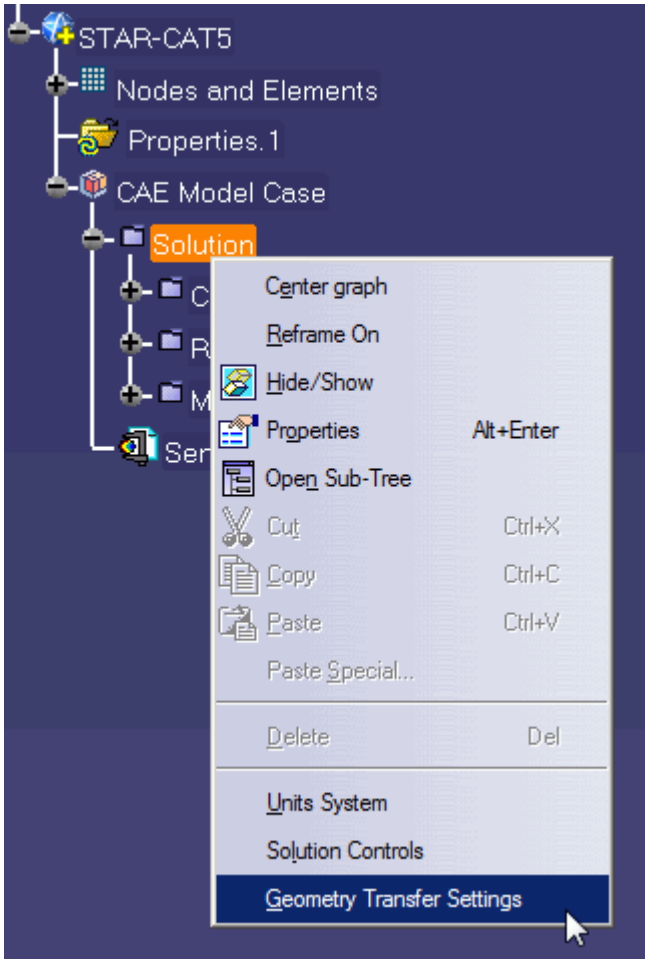
If the same entity, feature, or body is published more than once in the assembly tree, then the top-most publication is used for that entity, feature, or body respectively.

Controlling How the Geometry is Transferred to STAR-CCM+

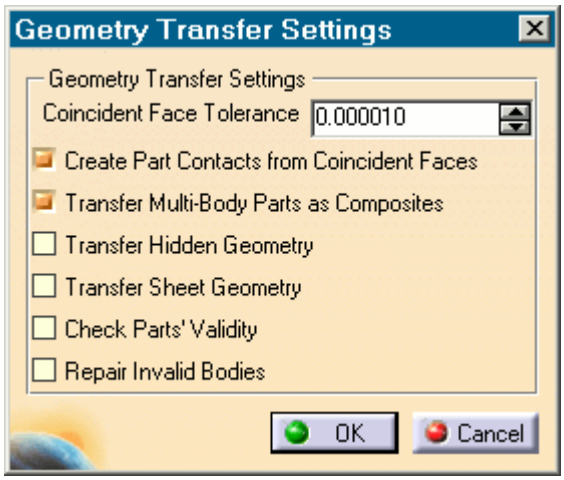
You can control how the geometry is transferred from CATIA to STAR-CCM+ through the *Geometry Transfer Settings* dialog.

To access the *Geometry Transfer Settings* dialog:

1. Right-click the **STAR-CAT5** > **CAE Model Case** > **Solution** and select **Geometry Transfer Settings**.



The *Geometry Transfer Settings* dialog appears.



2. Set the transfer parameters. See Geometry Transfer Settings Dialog.
3. Click **OK**.

Geometry Transfer Settings Dialog

Use the following properties to control how the geometry is transferred to STAR-CCM+.

Coincident Face Tolerance	Specifies the tolerance, in meters, for geometry in adjacent faces to be considered coincident. The default value is `1.0E-3`.
Create Part Contacts from Coincident Faces	Controls whether the geometry transfer feature searches for pairs of coincident faces on the CAD faces. For each pair found, part contact data is created and stored in the *Contacts* property (in STAR-CCM+). See the Creating Part Contacts from Coincident Entities section in the STAR-CCM+ user guide.
Transfer Multi-Body Parts as Composites	Specifies whether to transfer a single part containing multiple solid bodies as a composite part. Part contacts are created for each pair of coincident faces. Leaving this option deactivated merges all solid bodies in the part.
Transfer Hidden Geometry	Specifies whether to transfer hidden parts. By default, hidden parts in CATIA are not transferred. For files saved in STAR-CAT5 v10.06 and earlier, this option is activated by default.
Transfer Sheet Geometry	Specifies whether to transfer sheet bodies as part of the transfer process. This option is turned off by default.
Check Parts' Validity	Provides you with information on whether the bodies are geometrically and topologically valid, from the Parasolid geometry kernel point of view. A simple message is displayed in the output window which states if the surface mesh is valid. For example: `Body 2 is not valid`.
Repair Invalid Bodies	Attempts to repair invalid bodies. The number of invalid bodies that are successfully repaired is displayed in the output window.

Understanding the STAR-CAT5 Process

In STAR-CAT5 the typical workflow for running a simulation is as follows:

1. Ensure that the CAD model is suitable for analysis
2. Extract the fluid volume
3. Define the material properties for each part of the model, for example, air, water, aluminum
4. Define model physics, for example, turbulence, temperature, gravity
5. Specify boundary conditions, for example, inlets, outlets, symmetry planes
6. Specify mesh settings
7. Define solution controls
8. Run the case
9. Post-process the results in STAR-CCM+

Once you have obtained an initial solution you may wish to re-run the case with some different settings, or with some changes to the geometry. To do this you can:

1. Return to STAR-CAT5 and change settings or modify the geometry
2. Re-run the case

Using Suitable CAD Models in STAR-CAT5

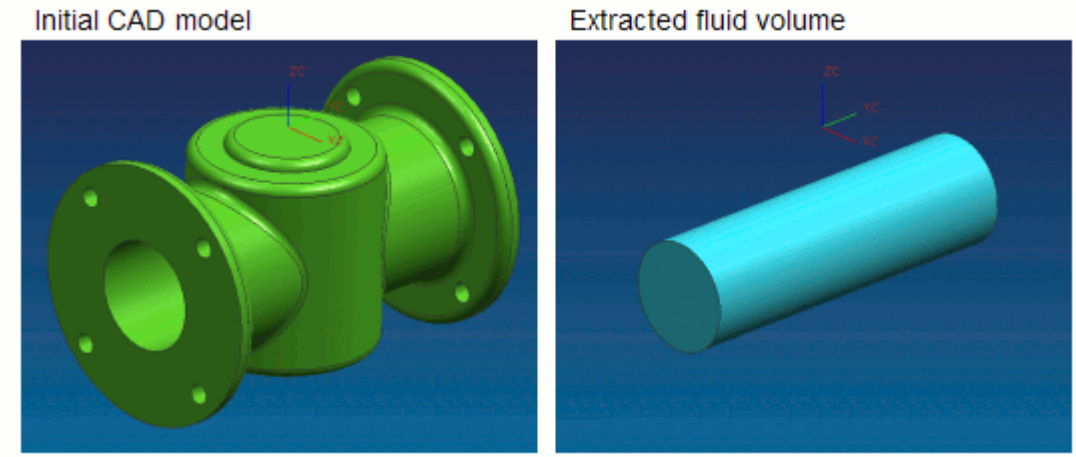
STAR-CAT5 only supports the use of solid bodies. If a geometry is made up entirely of surfaces then it will not be possible to define any boundary conditions, and the meshing options will be greyed out.

It is good practice to create copies of the original geometry file(s) before importing them into STAR-CAT5.

Suppressed or deactivated parts and features are not supported by STAR-CAT5 and will have no effect on the analysis. Hidden parts, however, are used in the analysis. Ensure that all desired parts and features are active before using them with STAR-CAT5.

Extracting the Fluid Volume

A CAD model usually represents a component made from solid material. However, in the case of flow simulation, a model representing the fluid volume is needed, requiring a different CAD model to be created. This is illustrated below:



In some instances, a simulation will require multiple regions, such as:

- Moving Reference Frames

  For rotating flow, such as turbines.
- Porous Media

  For representing regions with fine geometric structures
- Energy Source

  For simulating thermal specifications, such as conjugate heat transfer

These types of simulation require each region to be represented by individual contiguous parts within an assembly.

Defining Material Properties

Continua determines the characteristics of the fluid and solid entities used within the simulation. The behavior of a single continuum is primarily defined by the chosen material type and physics models.

Specification of the material type used in the simulation provides properties such as density, viscosity and conductivity, which are all used by the solver to calculate a solution. STAR-CAT5 includes a database of common materials and allows you to modify material properties where necessary.

Defining Model Physics

Physics models determine the nature of the fluid, and can be described as either:

- Turbulent or Laminar
- Constant Density or Ideal Gas

Additional models can also be activated to meet the needs of various simulation requirements:

- Passive scalar
- Temperature
- Gravity

A region is defined as a volume domain in a simulation, surrounded by boundaries. Each region in the simulation must be represented by a part in the native CAD package. When multiple regions are required, multiple parts must be used within an assembly.

All regions must be assigned to a continuum, as they represent a volume of material in the simulation. Region settings can be used to define porous media, moving reference frames, and an energy source.

Specifying Boundary Conditions

To define boundary conditions you first need to select the surface or surfaces on the model to assign to a separate boundary.

A boundary is defined as a surface, which along with other boundaries, surround and characterize a region. Boundary conditions determine how the boundary will affect the material within a region, and consequently, the simulation as a whole.

Inlet boundaries determine how much fluid enters the region, using a velocity, mass flow rate or stagnation inlet. Outlet boundaries define the process for fluid leaving the region, which are defined with either a pressure or flow-split ratio. Wall boundaries contain the fluid, and can also be used to define a thermal specification that will influence the transfer of energy to or from the region. Symmetry boundaries can be used to avoid modeling the entire region where appropriate, and so reduce calculation time.

As boundary types can affect the way in which the volume mesh is constructed, it is important to set them before generating the mesh.

Specifying Mesh Settings

The STAR-CCM+ solver uses the finite volume method to obtain an approximate solution to the underlying equations of fluid flow. This requires a mesh to be created, which represents the fluid volume as a collection of much smaller volumes known as cells.

A surface mesh is first created based on the boundaries defining the regions in the simulation. Thereafter a volume mesh is created to discretize the entire model.

The accuracy of the results is directly affected by mesh quality. If it is necessary to improve the quality of the mesh, some of the ways in which this can be done are to:

- Increase the mesh density, that is, the number of cells
- Add further sub-surface prism layers to capture near wall flow
- Refine the mesh in areas where flow characteristics are expected to change rapidly

As volume mesh quality is largely governed by the surface mesh quality, it is common practice to create and inspect the surface mesh before creating a volume mesh.

Defining Solution Controls

Solution controls determine the level of accuracy the solver will work towards. As a high accuracy solution will take longer, it may be necessary to compromise on this in order to obtain a result in a shorter time.

STAR-CAT5 includes five predefined solver settings ranging from fast (with low accuracy) to slow (with high accuracy). Advanced options are also available to provide further control.

Running the Case

When the simulation has been fully defined, the volume mesh is sent to STAR-CCM+ along with all the necessary data for running the simulation. The time to complete a simulation will vary depending on the mesh density and solution controls.

As the solver is running, residuals showing the rate at which individual equations are converging are shown in a plot. A scalar scene displaying absolute pressure is displayed on completion of the simulation.

Post-Processing the Results in STAR-CCM+

Post-processing of simulations is carried out using the STAR-CCM+ interface. This allows the results to be taken from the solver and displayed in a variety of scenes and plots.

Making Modifications to the Settings or Geometry

As STAR-CAT5 is completely integrated within the CATIA environment, it is very simple to perform parametric studies that examine the response of the fluid to changes in geometry. Once the first analysis is complete, the boundary conditions, run settings and all other properties of the CFD model are stored within the CAD file. Any changes in the flow field due to design changes can be recalculated by generating a new mesh and running the solver.

Connecting to Remote Machines

STAR-CCM+ uses a client-server architecture, and as such the CAD client (STAR-CAT5) and the server (STAR-CCM+) are not required to run on the same machine.

Using an SSH (secure shell) connection, you can connect the client and server on two different machines, even when these are on different platforms. Due to how the CAD client and server interact in the background, the SSH connection must not ask for a username or a password.

There are two ways that you can connect STAR-CAT5 and STAR-CCM+, depending on what your starting point is:

- You can start from the CAD client and launch the server on a remote machine. This method is useful for transferring CAD geometry from STAR-CAT5 running on Windows to STAR-CCM+ running on Linux.
- You can start from the server and connect to the CAD client on a remote machine. This method is useful for connecting to STAR-CAT5 to update the CAD geometry. With this method, you can perform an Optimate analysis on a cluster while the geometry is updated in STAR-CAT5.

Contents:

Running STAR-CAT5 with Remote STAR-CCM+
Running STAR-CCM+ with Remote STAR-CAT5

Running STAR-CAT5 with Remote STAR-CCM+

Starting from STAR-CAT5, you can choose to run the STAR-CCM+ server on a remote machine.

Using this method, you can set up the full CAE analysis in STAR-CAT5 on your local machine, and then run the STAR-CCM+ server on the remote machine. You can also choose to transfer your geometry from the CAD client to the remote STAR-CCM+ server. This feature is useful if you want to import geometry from STAR-CAT5 running on Windows to STAR-CCM+ running on Linux.

To configure STAR-CAT5 to launch STAR-CCM+ on a remote machine:

1. Make sure that you have a working SSH connection between the two machines.

   For the SSH connection, you require an SSH server on the machine running STAR-CCM+ and an SSH client on the machine running STAR-CAT5. The SSH connection to the remote machine must not ask for a username or a password. To avoid possible issues with a connection that still requires a password, add `-o NumberOfPasswordPrompts=0` to the SSH command. Setting this argument to zero stops the connection prompting for a password when one is needed — something that could cause the process to hang. See SSH Command.
2. In the STAR-CAT5 ribbon, click **Preferences**.
3. In the *Preferences* panel, set *Server Settings* to **Remote**.
4. Set the remaining settings for your connection to the remote STAR-CCM+ server. See Server Settings Panel.

   This information is used to build a command-line in the background when STAR-CAT5 invokes STAR-CCM+.

5. If you are going to connect to the same remote host in future sessions, click **Save as Default Settings**.

The next time STAR-CAT5 invokes STAR-CCM+, the STAR-CCM+ server launches on the specified remote machine. A STAR-CCM+ client is also launched on the local machine to allow you to see the active session on the remote machine.

Contents:

Server Settings Panel

Server Settings Panel

Specify the connection settings to the remote STAR-CCM+ server using the Server Settings pane.

Parameter	Description	
Server Host Type	Specifies whether to run the STAR-CCM+ server on the local host, or on a remote host.	
	Local	Runs the STAR-CCM+ server on the same host as STAR-CAT5.
	Remote	Runs the STAR-CCM+ server on the remote host that you specify.
SSH Command	Specifies the absolute path to the SSH program. If the path to the SSH program exists in your `Path` environment variable, you can leave this parameter as the default entry. You can include additional arguments in the SSH Command. The connection must not ask for a password or a username. To avoid possible issues with a connection that requires a password, add `-o NumberOfPasswordPrompts=0` to the SSH command. Setting this argument to zero stops the connection prompting for a password when one is needed, something that could cause the process to hang.	
Server Host	Specifies the remote host. You can enter the name, fully qualified domain name, IP address, or path to the host, depending on how your network is set up.	
Sim File Directory	Specifies the absolute path to the directory where you want to save the simulation file on the remote host. You must have read and write access to this directory. If a simulation file is present in this directory, STAR-CCM+ will use this file. If no file is found in this directory, a new one is created. An example directory on a Linux machine is shown below: `/users/linuxbox/james/documents/cadClientCases`	
Server Executable	Specifies the full path to the STAR-CCM+ executable on the remote host. An example path is shown below: `[STAR-CCM+_INSTALL_DIR]/star/bin/starccm+`	
Save as Default Settings	Saves the server settings that you specify in this panel as the default settings. Save these changes if you want to connect to the same remote server in future sessions.	

Running STAR-CCM+ with Remote STAR-CAT5

Starting from STAR-CCM+, you can connect to STAR-CAT5 on a remote machine.

This feature is useful if you have previously transferred your geometry from STAR-CAT5 on Windows to STAR-CCM+ on Linux, and you want to reconnect the two.

To connect STAR-CCM+ running on the local machine to STAR-CAT5 running on a remote machine:

1. Make sure that you have a working SSH connection between the two machines.

   For the SSH connection, you require an SSH server on the machine running STAR-CAT5 and an SSH client on the machine running STAR-CCM+. The SSH connection to the remote machine must not ask for a username or a password.
2. In STAR-CCM+, expand the **Geometry** > **CAD Client Models** > **[Model_Name]** and set *CAD-Client Update Mode* to **Remote**.

   The **Remote CAD Client Settings** node appears.
3. Set the remaining parameters for the CAD geometry. See CAD Geometry Settings.
4. Select the **Remote CAD Client Settings** node and set the connection setting for the remote CAD client. See Remote CAD Client Settings.
5. If you are connecting from a STAR-CCM+ server that is running on a cluster, activate the *Use Proxy Server* option and set up the proxy server. See Proxy Server Settings.

When you make changes to the imported CAD geometry in STAR-CCM+, the changes are passed back to STAR-CAT5 on the remote machine and the geometry is transferred back into STAR-CCM+.

Contents:

Remote CAD Client Settings Reference

Remote CAD Client Settings Reference

Specify the connection settings to the remote CAD client in the Remote CAD Client Settings panel.

CAD Geometry Settings

Parameter	Description	
CAD File	Specifies the full name of the CAD geometry file, including the extension.	
CAD File Location	Specifies the location of the CAD geometry file on the remote machine. An example path is shown below: `C:\CAD Client Cases`	
CAD-Client Update Mode	Specifies whether STAR-CAT5 is located on the local machine where STAR-CCM+ is installed, or whether STAR-CAT5 is located on a remote machine.	
	Local	STAR-CAT5 is located on the same local machine as STAR-CCM+.
	Remote	STAR-CAT5 is located on a remote machine.

Remote CAD Client Settings

Parameter	Description	
Remote Host	Specifies the remote host where STAR-CAT5 is installed. You can enter the name, full name, IP address, or path to the host, depending on how your network is set up.	
Remote CAD-Client Install Location	Specifies the absolute path to the root directory of the CAD client installation on the remote host. An example path to the default installation directory for version 12.06.### is shown below: `C:\Program Files\CD-adapco\12.06.###\STAR-CAD12.06.###`	
SSH Command	Specifies the command to use to run the SSH program. If you require a proxy server to connect to a cluster, you must use the following SSH command: `ssh -A -t <Cluster_Head_Node> ssh -A`	
Use Proxy Server	Specifies whether you require a proxy server to connect to a cluster.	
	Activated	Uses a proxy server as part of the connection to a cluster.
	Deactivated	Does not use a proxy server.

Proxy Server Settings

Parameter	Description
Proxy Host	Specifies the host name of the private network of the cluster. Enter the name, fully qualified domain name, or IP address. An example is shown below: `clusterhead.cm.cluster`
Proxy Public Host	Specifies the host name of the public network of the cluster. Enter the name, fully qualified domain name, or IP address. An example is shown below: `clusterhead.london.company.com`

Defining Materials and Physics

Defining the physical properties of a STAR-CAT5 model involves specifying the materials, flow characteristics, operating conditions, and physics models.

These definitions and settings are accomplished through the use of continua and regions, which are described in the following sections.

Contents:

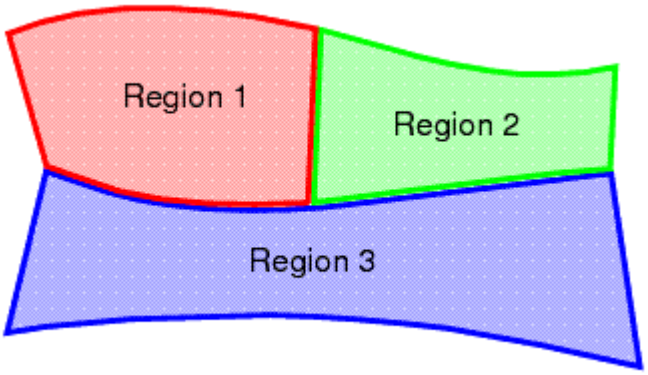
Working with Continua and Regions
Importing Geometry into STAR-CCM+ for Visualization

Working with Continua and Regions

Continua and regions are used together to capture the physical and material properties of a model.

A continuum represents a material in combination with a set of physics models that will be used to simulate the response of the material to the conditions imposed upon it. A continuum also provides the facility to define reference and initial values.

A region represents one or more volumes within the overall model, not necessarily contiguous, which will share the same continuum and set of boundary conditions. Regions are defined using CAD parts, and will be discretized by the volume mesher.



As well as holding a set of boundary conditions, regions can also be used to model specific physical behavior, including porous resistance, moving reference frames and energy sources.

Parts can be imported into STAR-CCM+ for visualization purposes without affecting the analysis.

Continua are assigned to single or multiple regions, which can contain single or multiple parts. In part mode, only one continuum and one region are available, as there is only one part. In assembly mode, new continua can be added to represent different materials present in the simulation, and new regions can be added to

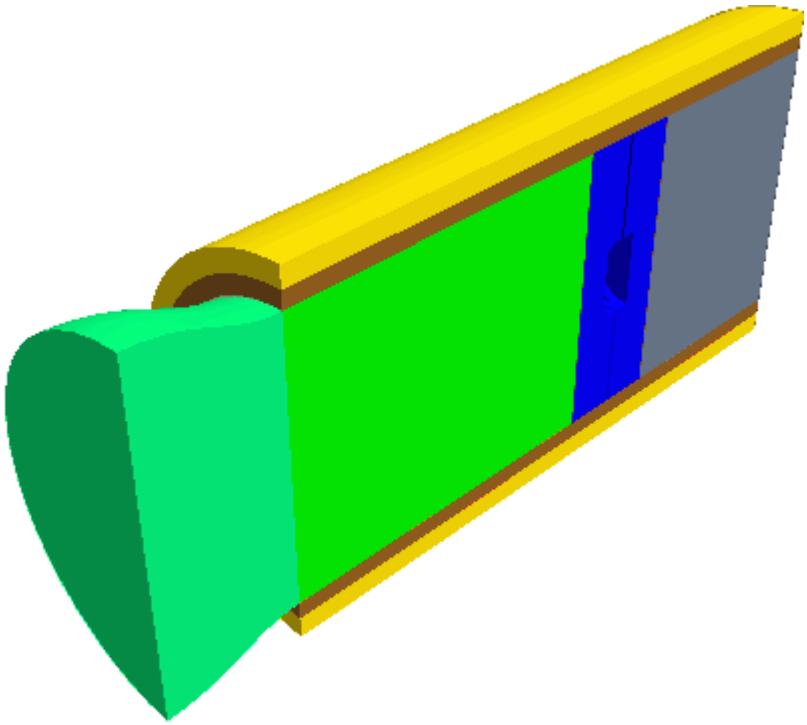
represent the parts composed of these materials. Continua and regions can be deleted if they are not required, although the simulation must always contain at least one of each.

There is a well defined relationship between continua, regions, boundary conditions and CAD parts.

- Every region must be associated with a continuum.
- Several regions can be associated with the same continuum.
- Every CAD part must belong to a region.
- Several CAD parts can be associated with the same region.
- Boundary conditions belong to regions, and cannot be shared between regions.
- Every CAD surface must have a boundary condition associated with it. A default wall boundary is applied to CAD surfaces that have not been specifically associated with a user-defined boundary condition.

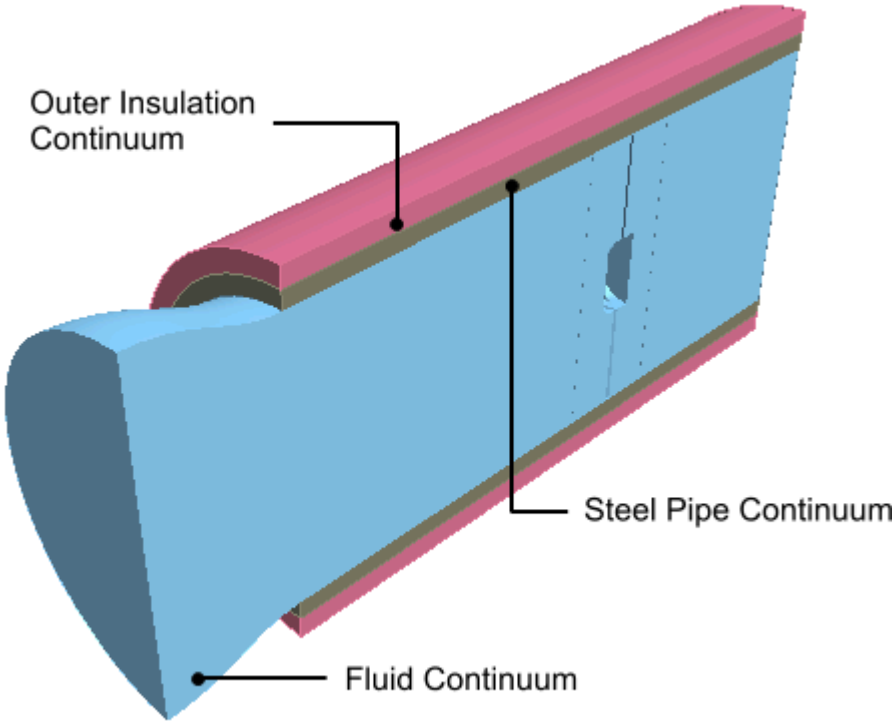
The following series of illustrations depict the relationship between CAD parts, continua, and regions, for a circular steel duct containing a rotating fan, and surrounded by a layer of insulation.

- Initially we have a collection of CAD parts in an assembly. These must be imprinted where appropriate to ensure that interfaces between parts can be created correctly.



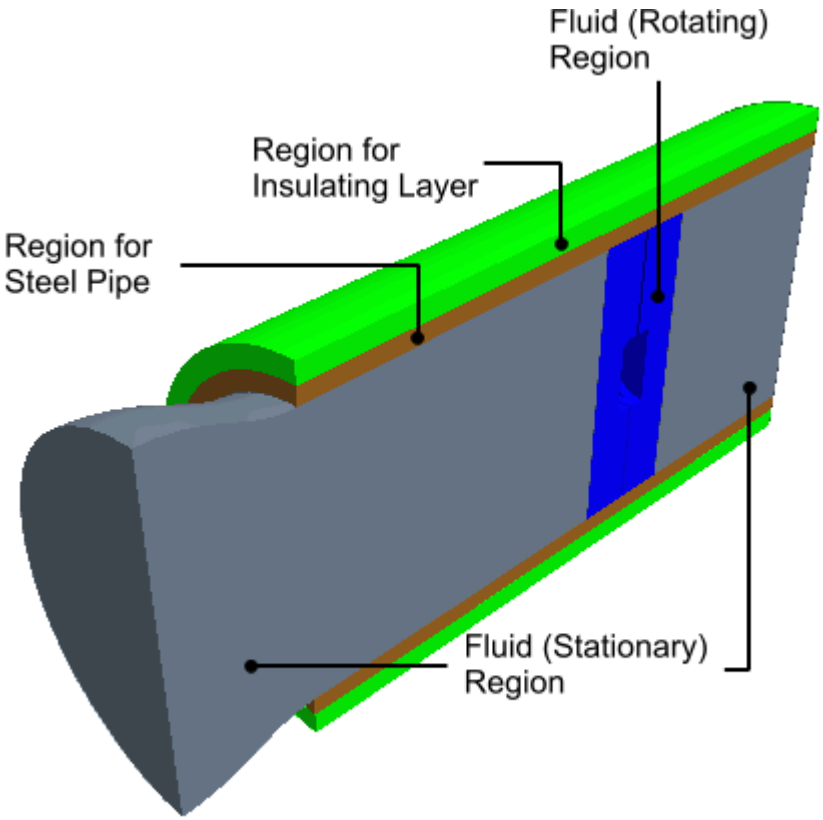
Assembly of CAD Parts (imprinted where necessary)

- Three continua are required to represent the three different materials present in the simulation: the fluid (for example, air), the steel pipe, and the insulation.



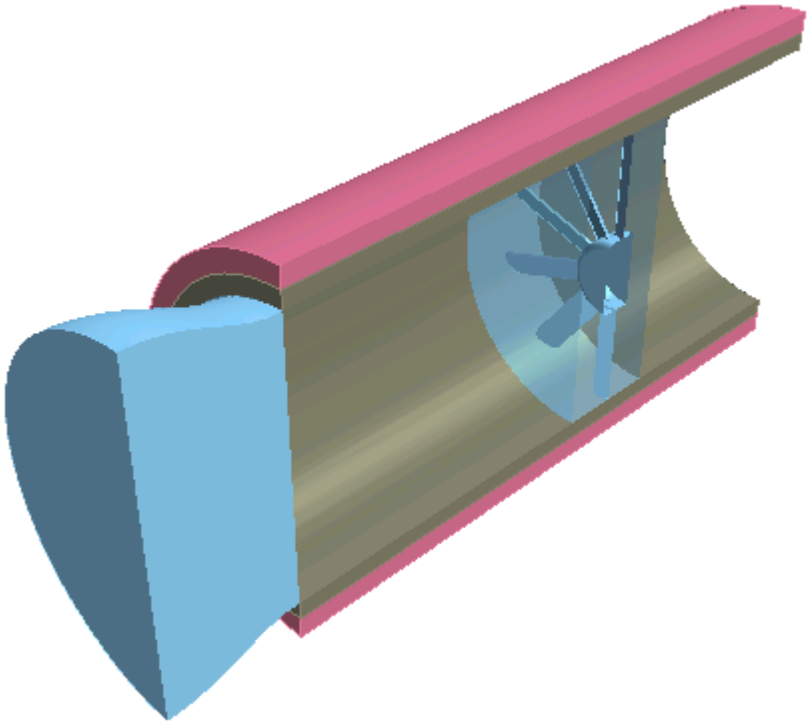
View of Continua Required for Simulation

- Two regions are required for the fluid portion of the simulation, as a rotating part is defined by applying a rotating frame of reference to a separate region surrounding the fan blades. Further regions are required for each of the solid parts (the steel pipe and insulating layer), which means a total of four separate regions overall.



View of Regions Required for Simulation

- A clearer picture can be given of the fan blade itself.



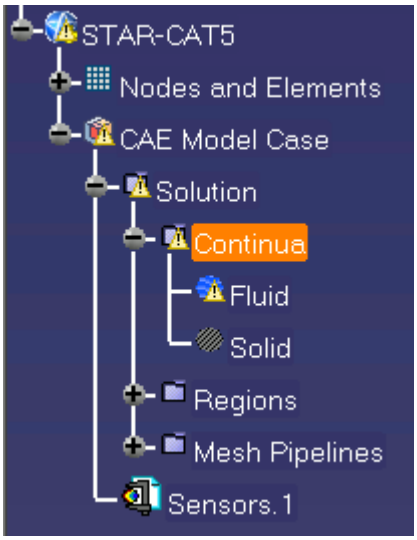
For an analysis containing a single part, for example a fluid-only case involving fluid flow through a converging-diverging pipe, the model will only require one continuum and one region in order to fully define the physical and material properties of the model. In this case, STAR-CAT5 automatically assigns the default continuum to the default region, and no further assignments are necessary.

Contents:

Setting Up Continua
Setting Up Regions

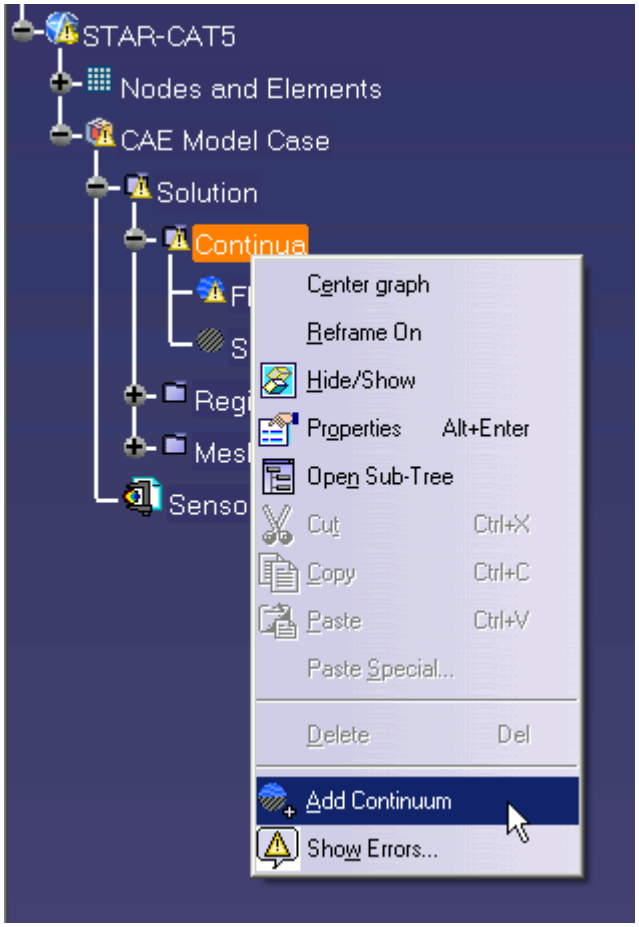
Setting Up Continua

To define different materials and associated physics for use in an assembly, you require additional continua. In each continuum you can specify a unique set of properties, such as a different material type or temperature. In assembly mode, continua are stored under a **Continua** node, as shown below.



To create an additional continuum:

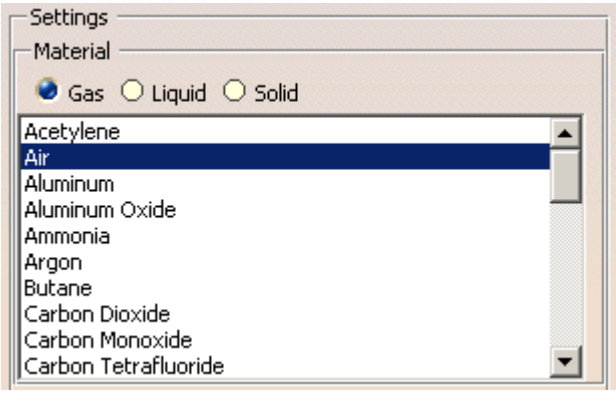
1. Right-click the **Continua** node in the object tree and select **Add Continuum**.



The *Continuum Settings* dialog appears.

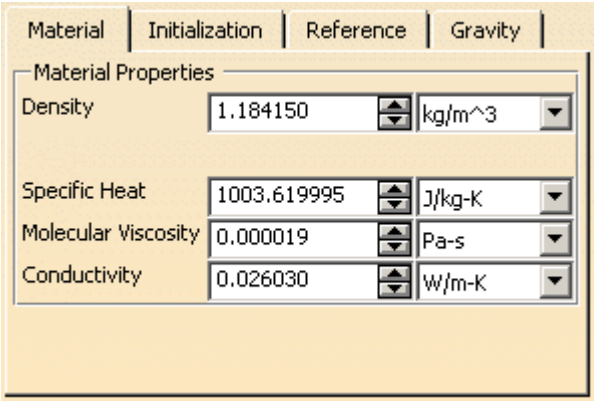
2. In the *Name* field, enter a name for the continuum.
3. In the *Materials* group, select the type of material that you want to define:
   - **Gas**
   - **Liquid**
   - **Solid**
4. Select a material from the list.

   You can modify the properties of any material that you select.



Note: You can select only one material per continuum. If your simulation contains a number of materials, you must create additional continua. This action is only possible for assemblies.

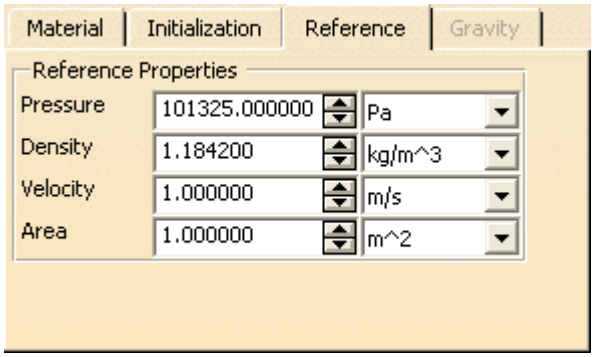
5. In the *Flow Characteristics* group, select the type of flow that you want to model: **Laminar** or **Turbulent**. This group only appears for fluids.
6. In the *Equation of State* group, select the equation of state : **Constant Density** or **Ideal Gas**. This group only appears for gases.
7. In the *Optional Physics Model* group, activate any optional physics models: **Passive Scalar** and **Temperature**.
8. Define the material properties in the *Material Properties* group.



See STAR-CAT5 Material Properties Panel.

Note: You can change units of properties from the drop-down menu.

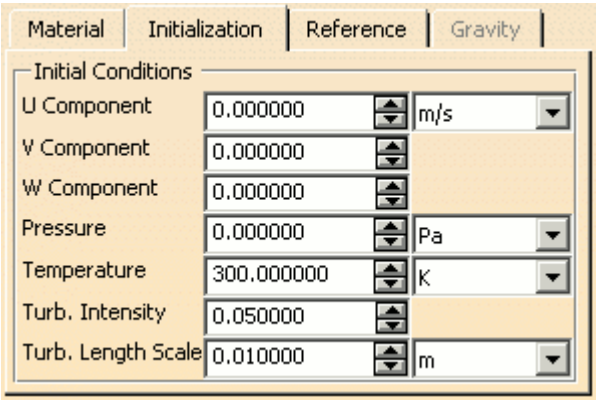
9. In the *Reference Properties* group, define the reference properties.



See STAR-CAT5 Reference Properties Panel.

10. If you want to include the body force due to gravity when calculating the momentum equations in STAR-CCM+, activate the gravity model in the *Gravity* group box.

    See Activating the Gravity Model in STAR-CAT5.
11. Set the initial conditions in the *Initialization* group box.



See STAR-CAT5 Initialization Panel.

Specifying initial conditions that are close to the expected solution can reduce the number of iterations required to reach a solution. You can set the initial conditions similar to the inlet conditions of the fluid domain.

Initial conditions are specified at a continuum level; if no initial conditions are specified, STAR-CAT5 will use default values to initialize the solution.

12. Click **OK**.

    The continuum is added to the object tree.

Once you have created a continuum, you can assign it to a region.

Contents:

STAR-CAT5 Continuum Properties
Activating the Gravity Model in STAR-CAT5

STAR-CAT5 Continuum Properties

The continuum dialog has a number of properties that you can define. This section provides a description for each property.

STAR-CAT5 Material Properties Panel

You can define the following material properties in STAR-CAT5:

Property	Description
Density	Available for all types of materials. For gases, this property is available when you use the Constant Density equation of state.
Specific Heat	Available for all material types when you activate the Temperature physics model.
Molecular Viscosity	Only available for fluids.

Property	Description
Conductivity	Available for all materials when you activate the temperature physics model.
Molecular Weight	Only applies to gases. This property is available when you use the Ideal Gas equation of state.

STAR-CAT5 Reference Properties Panel

You can define the following continuum reference properties:

Property	Description
Reference Pressure	The pressure values that are defined in the simulation are all relative to the reference pressure. For more information, see the What is the Reference Pressure? section.
Reference Temperature	The reference temperature is used to define the operating temperature of the fluid in the buoyancy source term of the Boussinesq model. This parameter is used by STAR-CCM+ when the temperature, constant density, and gravity physics models are enabled.
Reference Density	These quantities are not used as part of the CFD calculation, but is necessary to calculate the force coefficients when post-processing.
Reference Velocity	
Reference Area	

STAR-CAT5 Initialization Panel

Specifying initial conditions that are close to the expected solution can reduce the number of iterations required to reach a solution. You can also set the initial conditions to match inlet conditions of the fluid domain. You can define the following continuum reference properties:

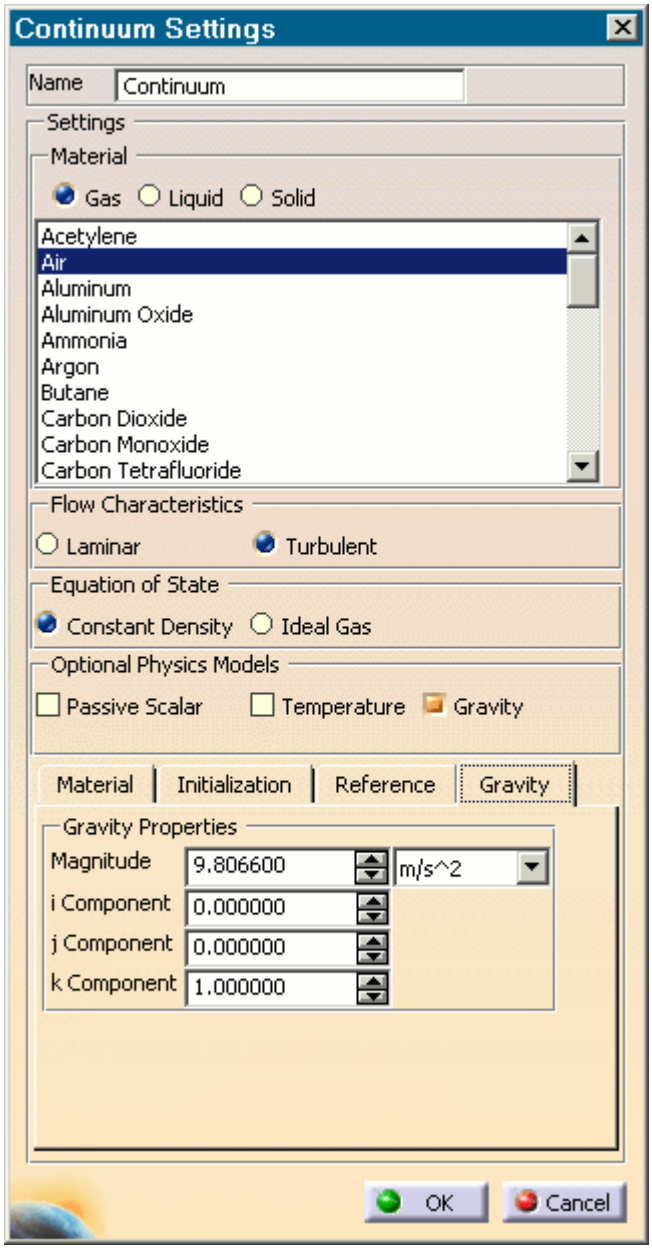
Property	Description
U Component	The U direction vector for the initial velocity.
V Component	The V direction vectors for the initial velocity.
W Component	The W direction vectors for the initial velocity.
Initial Pressure	The initial pressure. The physical value of the initial pressure defines the starting pressure in the regions to which the continuum is assigned. The initial pressure is specified with respect to the reference pressure. The default initial pressure is `0 Pa`.
Initial Temperature	The initial temperature. This physical value of the initial temperature defines the starting temperature

Property	Description
	throughout any region to which the continuum is assigned. The default initial temperature is `300 K`.
Turb. Intensity	The initial turbulence intensity. The default value is `0.05` (`5%`)
Turb. Length Scale	The initial turbulent length scale. The default value is `0.01 m`.

Activating the Gravity Model in STAR-CAT5

If you require the gravity model in STAR-CAT5, you can activate it in the *Continuum Settings* dialog:

1. In the *Continuum Settings* dialog, activate the **Gravity** checkbox in the *Optional Physics Models* group.



2. Set the acceleration due to gravity in the *Magnitude*box.

   You can change the units from the drop-down list.

3. Specify the direction vector for the gravity in the `[i, j, k]` *Components of Gravity* box.

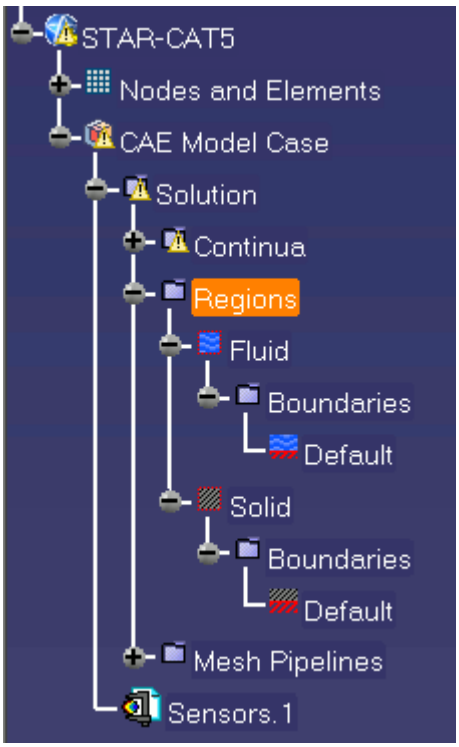
4. Click **OK**.

   The continuum is added to the object tree.

Setting Up Regions

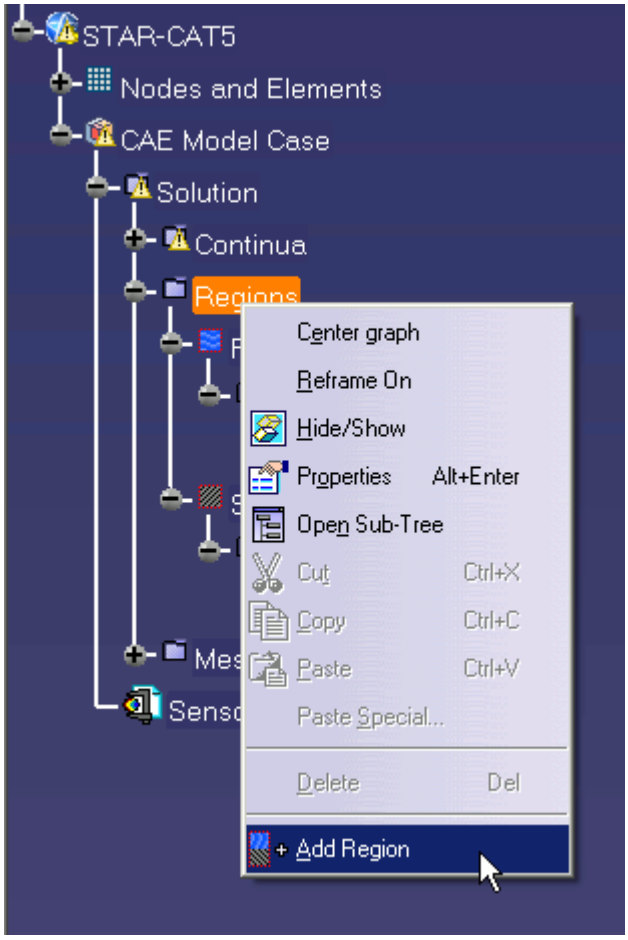
Additional regions are required when modelling multiple region physics, or when specific behavior such as porous resistance is required for a particular part within the same continuum. Assign parts from an assembly to regions. You can control the properties of these regions through the assigned continuum.

In assembly mode, regions are stored under the **Regions** node in the object tree, as shown below.

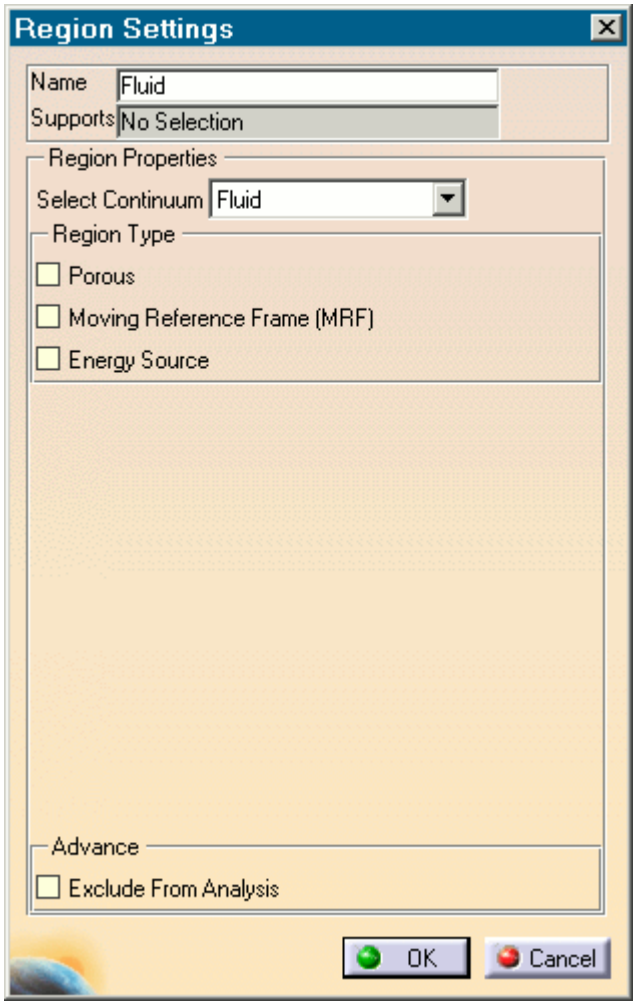


To create an additional region:

1. Right-click the **Regions** manager node in the object tree and select **Add Region**.



The *Region Settings* dialog appears.

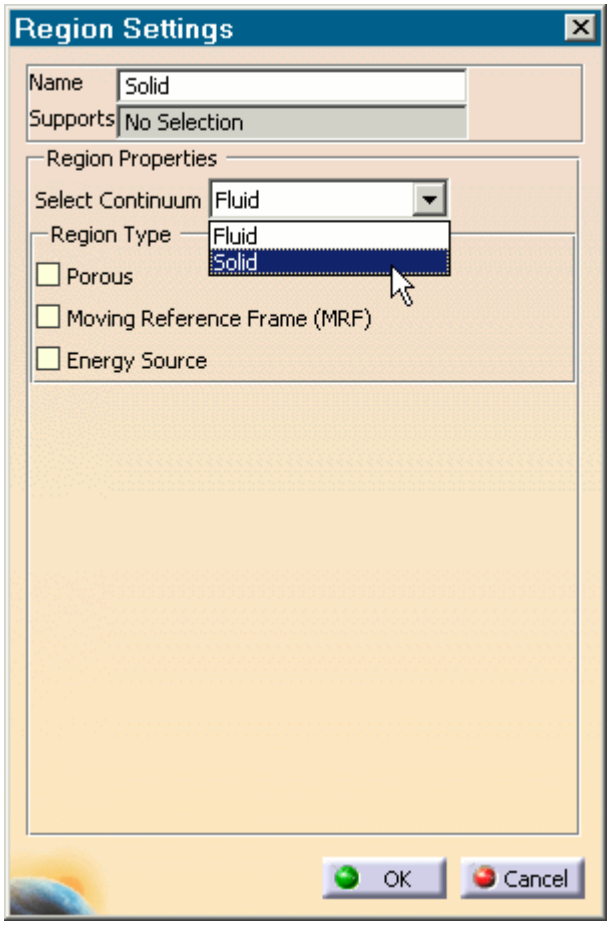


2. In the *Name* property, enter a name for the region.
3. Select the parts that you want to include in the region.

   You can select multiple parts from the graphics window or the CATIA feature tree.

   Selected parts appear in the *Supports* field.
4. Select a continuum to apply to the region.

   All available continua are listed here.



Note: Assigning a continuum to a region causes the region to inherit the physical and material properties defined in the continuum

5. Activate any additional region physics for the region:
   - Porous. See Defining a Porous Region.
   - Moving Reference Frame. See Defining a Moving Reference Frame.
   - Energy Source. See Defining an Energy Source.
6. If you want to import parts into STAR-CCM+ for visualization purposes, but do not want the region to be included in the CFD calculations, activate the *Exclude from Analysis* option. See Importing Geometry into STAR-CCM+ for Visualization.
7. Click **OK**.

A new region is added to the STAR-CAT5 object tree under the **Regions** node.

The parts that you assign to a region define the geometric extent of the region. Within this geometry, properties and behavior of the material is governed by the physics models and material properties defined in the region and continuum settings.

Note: Each region must be assigned at least one part, otherwise it will be ignored during the CFD analysis.

By default, every STAR-CAT5 model contains one continuum, and this is automatically assigned to the default region and each additional region that you create. If multiple continua are defined in the model, you can choose a different continuum to assign to each region. This enables you to control the physical properties of each region, including varying material types.

All the parts in an assembly are assigned to the default region (this is the initial region created by STAR-CAT5). When you assign a part to another region, it is automatically removed from the default region. If you remove a part from a region, it is reassigned back to the default region.

Contents:

Defining a Porous Region
Defining a Moving Reference Frame
Defining an Energy Source

Defining a Porous Region

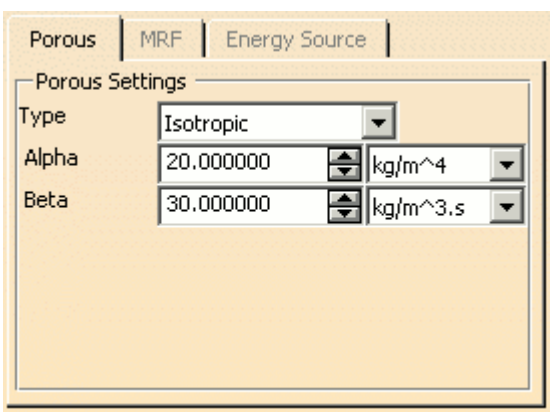
Porous resistance affects fluid flow through a region in a particular direction.

To define a porous region:

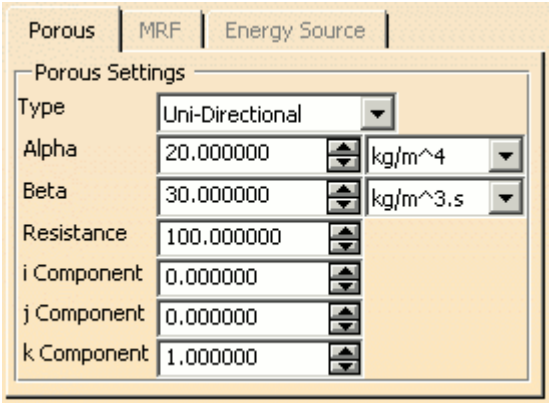
1. In the *Region Settings* dialog, activate the *Porous* tab.

   The Porous settings appear.

2. Select the type of resistance to model:
   - **Isotropic**



   - **Uni-Directional**



3. Specify the porosity settings. See Porous Region Panel Settings.

Contents:

Porous Panel Settings

Porous Panel Settings

Porous resistance affects the fluid flow through a region in a particular direction. There are two types of porous resistances available: Isotropic and Uni-Directional.

Parameter	Description
Isotropic	
Alpha	Specifies inertial resistance. You can change the units using the drop-down list.
Beta	Specifies viscous resistance. You can change the units using the drop-down list.
Uni-Directional	
Alpha	Specifies inertial resistance. You can change the units using the drop-down list.
Beta	Specifies viscous resistance. You can change the units using the drop-down list.
Resistance	Specifies the cross-flow resistance factor.
i Component	Specify the components that define the direction of the flow in the porous region. These values are required to determine the coefficients of inertial and viscous resistance in a direction perpendicular to the specified vector.
j Component	
k Component	

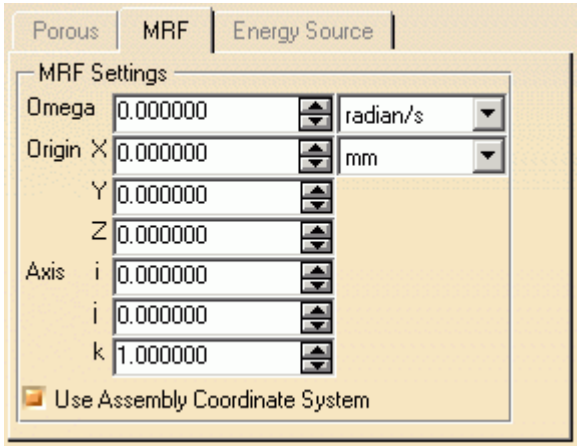
Defining a Moving Reference Frame

You can use moving reference frames to model region motion.

To define a moving reference frame:

1. In the *Region Settings* dialog, activate the *Moving Reference Frame* tab.

The Moving Reference Frame settings appear.



2. Specify the Moving Reference Frame parameters. See Moving Reference Frame Panel Settings.

Contents:

Moving Reference Frame Panel Settings

Moving Reference Frame Panel Settings

You can use Moving Reference Frames to model region motion, such as rotation.

Parameter	Description
Omega	Specifies the angular velocity. You can change the units from the drop-down list.
X	Specify the coordinates for the origin of the axis of rotation. The units that are used here are the same as the file defaults.
Y	
Z	
i	Specify the direction of the axis of revolution.
j	
k	

Parameter	Description
Use Assembly Coordinate System	Controls whether the properties of the axis of rotation are defined using the coordinate system of the assembly or the coordinate system of the selected part. To use the coordinate system of the assembly, activate the *Use Assembly Coordinate System* this option. If you select multiple parts for the new region, the *Use Assembly Coordinate System* option is automatically activated and grayed out

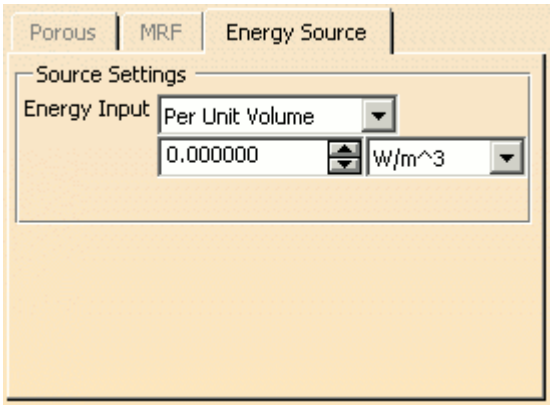
Defining an Energy Source

You can define a direct source or sink of energy to or from a region using the energy source model. This model is only available if you activated the Temperature physics model in the continuum.

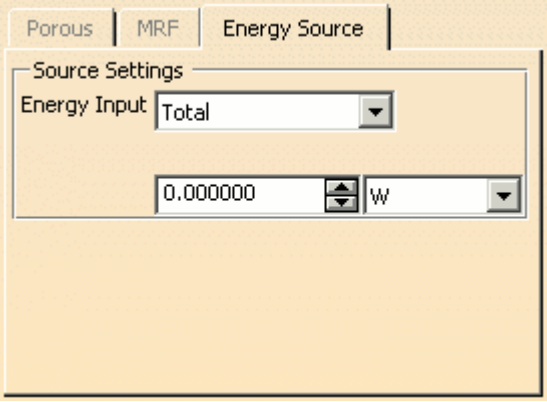
To specify that a region is an energy source:

1. Make sure that the Temperature model is activated in the continuum that is applied to the region. See Setting Up Continua.
2. In the *Region Settings* dialog, activate the *Energy Source* tab.

   The Energy Source settings appear.
3. Select the type of energy source to apply:
   - **Energy Input per Unit Volume**



   - **Total Energy Input**



4. Specify the energy source settings.

   You can change the units using the drop-down menu.

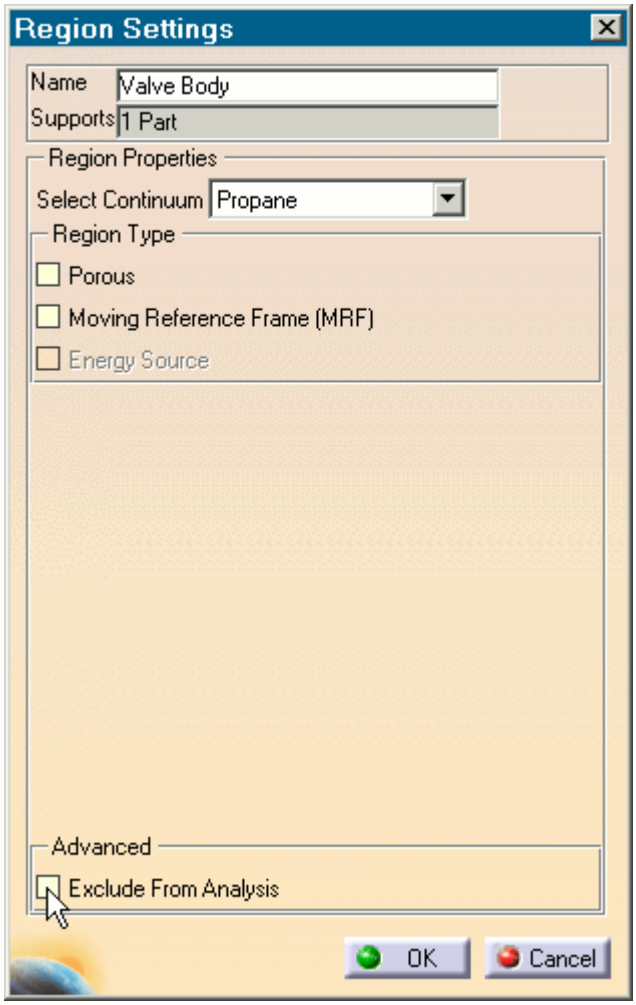
Importing Geometry into STAR-CCM+ for Visualization

You can import CAD parts, that do not form part of the analysis, into STAR-CCM+ for visualization purposes.

Add these parts to a separate region—this region is excluded from the analysis. This feature is useful for showing the original geometry in a post-processing scene in STAR-CCM+. For example, for an internal fluid volume of a pipe, the original geometry is the solid pipe.

To exclude CAD parts from the analysis:

1. In the *Region Settings* dialog, add the CAD parts that you want to use for visualization purposes.
2. Activate the *Exclude From Analysis* checkbox in the *Advanced* panel.



3. Click **OK**.

When you run the case, a surface mesh is generated for this region using the settings specified in the Mesh Pipeline.

Note: As is the case for all assemblies, you must imprint all parts that are used for visualization purposes onto any surface that they cut, otherwise you will not be able to generate the surface mesh.

An example of this feature is shown in the following section, where a valve and valve body geometry are imported into STAR-CCM+ for visualization.

Contents:

Setting Up a STAR-CCM+ Scene to Display Imported Geometry from STAR-CAT5

Setting Up a STAR-CCM+ Scene to Display Imported Geometry from STAR-CAT5

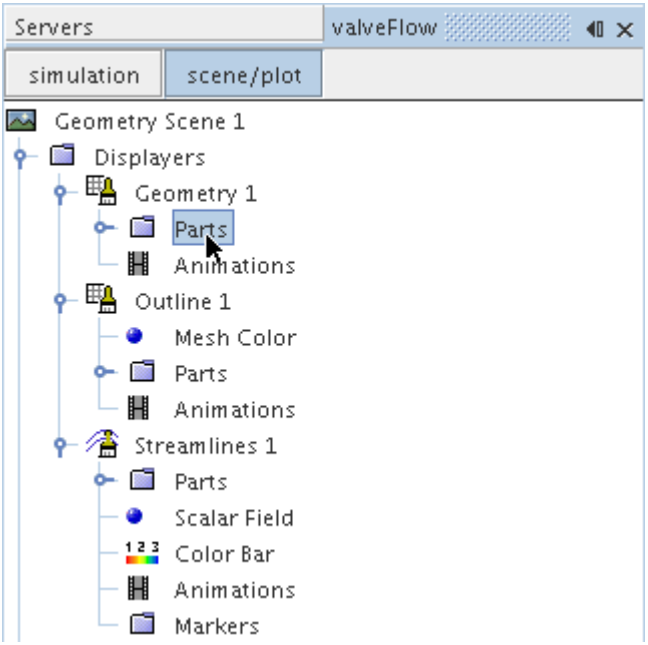
You can visualize geometry that is not included in the analysis. For example, you can add the valve and valve housing geometry to a Geometry Scene in STAR-CAT5, even though this does not have any effect to the flow.

To use parts that are excluded from the analysis in a scene:

1. In STAR-CCM+, create a new scene.

   In the following example, a Geometry Scene is used.
2. Create a streamlines derived part.
3. Click on the **scene/plot** button.

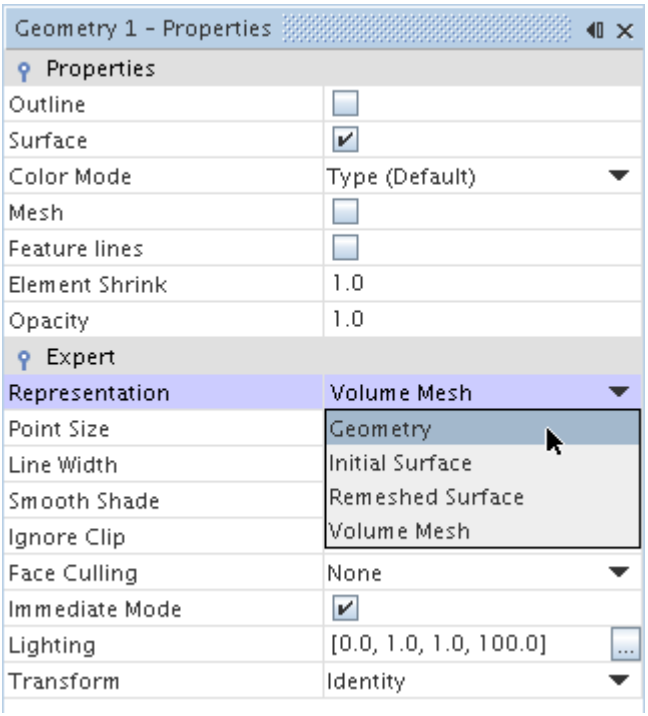
   The object tree switches to the scene nodes.
4. Double-click on the **Geometry Scene** > **Displayers** > **Geometry** > **Parts** node.



5. In the *Edit* dialog:
   a) Deactivate the **Regions** node.
   b) Expand the **Parts** node and activate the part that corresponds to the valve body.

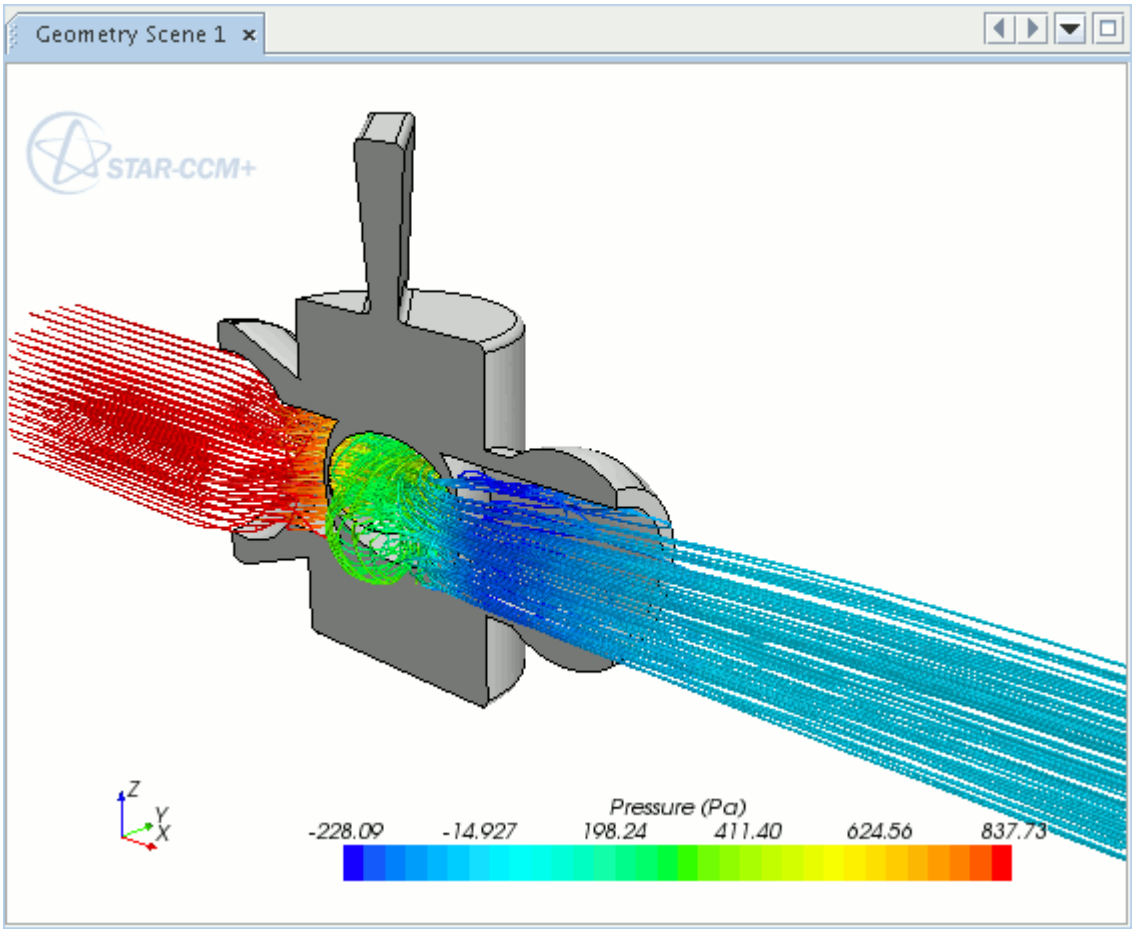
      This is the region that you chose to exclude from the analysis in STAR-CAT5.
6. Click **OK**.
7. Select the **Geometry** node and set *Representation* to **Geometry**.

   The imported geometry only contains a surface mesh and does not appear by default (the default *Representation* is set to **Volume**). This is why you must change the *Representation* to **Geometry**.



8. Activate the *Outline* option to change the color of the outlines from red to black.

The finished scene is shown below.



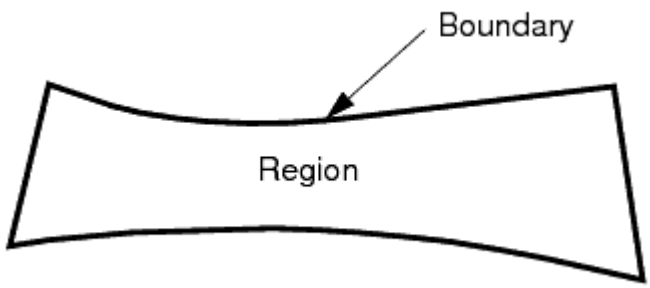
Displaying Imported Geometry in Scalar and Vector Scenes in STAR-CCM+

To display an imported geometry in a Scalar or Vector scene, you need to add a geometry displayer to the existing scene. You can then display the geometry by following steps described in Setting Up a STAR-CCM+ Scene to Display Imported Geometry from STAR-CAT5.

Setting Boundary Conditions

Boundary conditions tell the STAR-CCM+ solvers how the substance within the region interacts with its environment. A velocity inlet boundary, for example, tells the solvers that an amount of substance is being added to the fluid domain at a certain rate.

A STAR-CAT5 boundary is made up of one or more CAD surfaces within the simulation which, along with other boundaries, surround and define a region.



The boundary is discretized by the surface mesher, which is then used by the volume mesher to mesh the entire region.

It is important to set the boundary conditions before creating the volume mesh, as some parts of the volume mesh, such as prism layers, depend on the type of boundary.

By default, each region contains only one boundary. To define more boundaries you need to add boundaries to the region. You can modify the properties of existing boundaries by editing its boundary node.

Some boundaries may require the specification of turbulence quantities, temperature, and passive scalar concentration, depending on whether these modeling options have been activated in Continuum Settings.

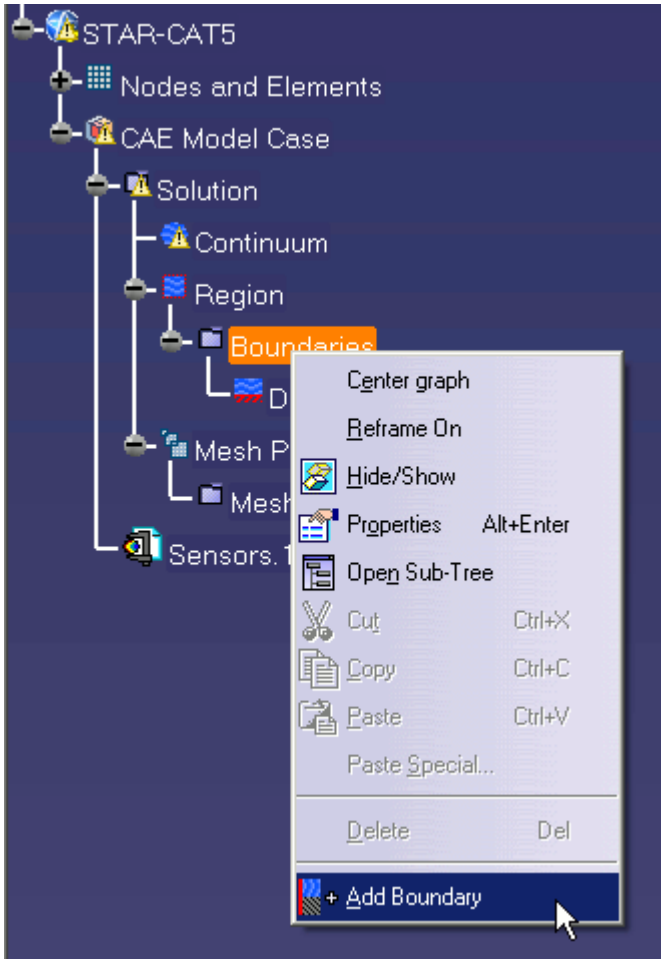
Contents:

Creating a New Boundary

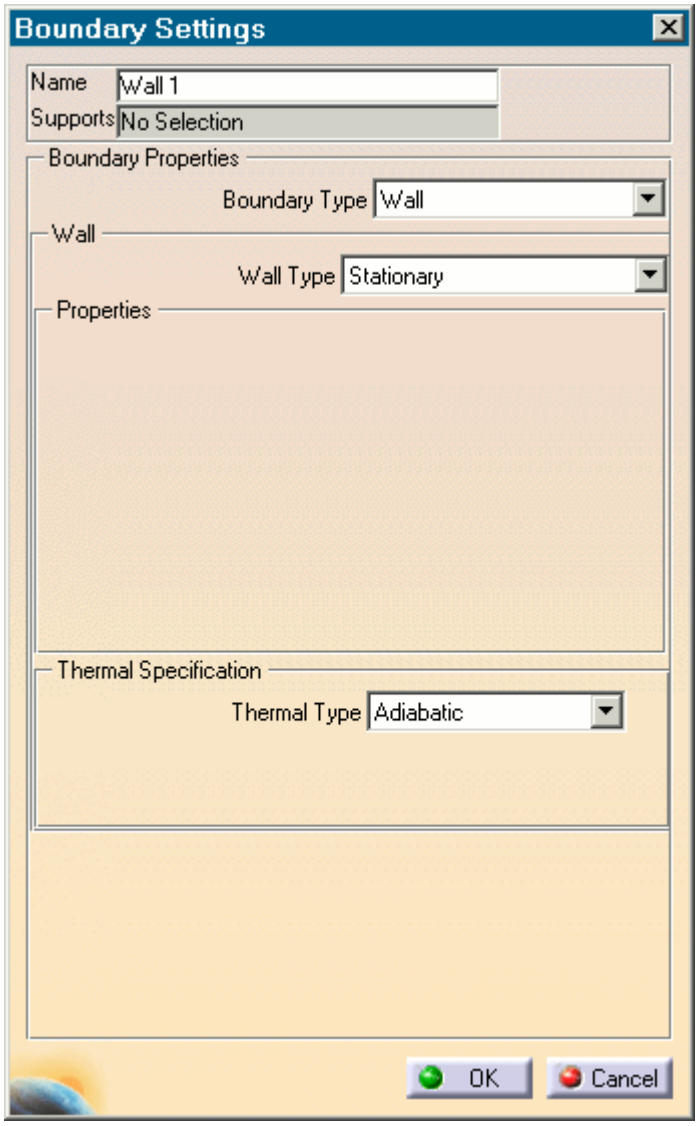
Creating a New Boundary

To create a new boundary:

1. Right-click the **Boundaries** node within the object tree and select **Add Boundary**.



The *Boundary Settings* dialog appears.



2. On the CAD model, select the surfaces that you want to include in the boundary. See Assigning Surfaces to Objects.

   These selections are reported in the *Supports* box in the Boundary Settings dialog.
3. In the *Name* field, enter a name for the boundary.
4. Select the boundary type and set the boundary conditions:
   - **Velocity Inlet**. See Velocity Inlet Boundary.
   - **Mass Flow Inlet**. See Mass Flow Inlet Boundary.
   - **Stagnation Inlet**. See Stagnation Inlet Boundary.
   - **Outlet**. See Outlet Boundary.
   - **Pressure Outlet**. See Pressure Outlet Boundary.
   - **Symmetry Plane**. See Symmetry Plane Boundary.
   - **Wall**. See Wall Boundary.
   - **Custom**. See Custom Boundary.
5. Click **OK**.

Contents:

Boundary Type Reference
Common Boundary Conditions

Boundary Type Reference

This section contains the various boundary types that are available in STAR-CAT5.

Velocity Inlet Boundary

A velocity inlet boundary represents the inlet of a duct at which you know the flow velocity. The velocity inlet settings are displayed below.

Property	Description
Normal Velocity	
Normal Velocity	Specifies the velocity normal to the surface. You can change the units using the drop-down list.
Cartesian Velocity Components	
U Component	Specify the components of the velocity vector. You can change the units using the drop-down list.
V Component	
W Component	
Cylindrical Velocity Components	
R Component	Specify the cylindrical components of the velocity vector. You can change the units using the drop-down list.
Theta Component	
Z Component	
X	Specify the coordinates of the origin for the velocity vector.
Y	
Z	
i	Specify the components for the direction vector.
j	
k	

Mass Flow Inlet Boundary

A mass flow inlet boundary represents the inlet of a duct at which you know the mass flow rate. The mass flow settings are displayed below.

Property	Description
Mass Flow Normal to Surface	

Property	Description
Mass Flow Rate	Specifies the mass flow rate, normal to the boundary surface. You can change the using the drop-down list.
Mass Flow in Specified Direction	
Mass Flow Rate	Specifies the mass flow rate. You can change the using the drop-down list.
i Component	Specify the direction components of the mass flow.
j Component	
k Component	

Stagnation Inlet Boundary

The stagnation inlet boundary is an inlet condition that is well-posed for compressible flows, although it is equally valid for incompressible flows. The stagnation conditions refer to the conditions in an imaginary plenum, far upstream, in which the flow is completely at rest. The stagnation inlet settings are displayed below.

Property	Description
Stagnation Flow Normal to Surface	
Pressure	Specifies the stagnation pressure. You can change the units using the drop-down list.
Stagnation Flow in Specified Direction	
Pressure	Specifies the stagnation pressure. You can change the units using the drop-down list.
i	Specify the direction components for the flow.
j	
U Component	

Outlet Boundary

An outlet boundary splits the flow between multiple outlets using a fraction of the total mass flow through each of the outlet boundaries. The outlet panel settings are provided below.

Property	Description
Split Ratio	Specifies the ratio of the split as a value between 0 and 1. The sum of all the flow-split ratio values in a region must equal 1. If you only have one flow-split boundary in the simulation, all the flow is assumed to exit through this boundary.

Pressure Outlet Boundary

A pressure boundary is a flow outlet boundary at which you specify the pressure. The pressure outlet panel settings are provided below.

Property	Description
Pressure	Specifies the pressure on the selected boundary. You can change units using the drop-down list.

Symmetry Plane Boundary

A symmetry plane boundary represents an imaginary plane of symmetry in the simulation. The solution that is obtained with a symmetry plane boundary is identical to the solution that would be obtained by mirroring the mesh about the symmetry plane (in half the resulting domain). There are no settings which are specific to symmetry plane boundaries.

Wall Boundary

A wall boundary represents an impermeable surface. The wall panel settings are provided below.

Property	Description	
Wall Type	Specifies the type of conditions to apply to the wall boundary.	
	Stationary	Specifies the wall as a stationary impenetrable surface.
	Translating	Specifies the wall as a moving impenetrable surface. You specify the velocity for the translation.
	Rotating	Specifies the wall as a rotating impenetrable surface.
	Slip	Specifies the wall as a stationary, impenetrable, but friction-free surface.
Translating		
U Component	Specify the velocity vector. You can change the units using the drop-down list.	
V Component		
W Component		
Rotating		
Rotational Velocity	Specifies the rotational velocity. You can change the units using the drop-down list.	
X	Specify the coordinates of the origin for the axis of rotation.	
Y		
Z		
i	Specify the components of the axis of rotation	
j		
k Component		

Custom Boundary

In STAR-CCM+, there are additional boundary types to those available in STAR-CAT5. The custom boundary in STAR-CAT5 acts as a placeholder for a boundary that you will customize in STAR-CCM+.

Common Boundary Conditions

Some boundary conditions are not specific to any boundary type and appear in multiple boundaries, so long as the physics model that is associated with the condition is activated.

Common Boundary Conditions on Inlet and Pressure Outlet Boundaries

Temperature, turbulence, and passive scalar boundary conditions are common to all inlet boundaries as well as the Pressure Outlet boundary.

Property	Description
Temperature	Specifies the temperature on the boundary surface. You can change the units using the drop-down menu.
Turbulence Intensity	Specifies the turbulence intensity on the boundary surface. You can change the units using the drop-down menu.
Turbulent Length Scale	Specifies the turbulent length scale on the boundary surface. You can change the units using the drop-down menu.
Passive Scalar	Specifies the amount of passive scalar crossing the boundary. You can change the units using the drop-down menu.

Boundary Conditions on Wall Boundaries

When the energy equation is being solved, you can specify the thermal specification on wall boundaries.

Property	Description	
Adiabatic	Specifies that no transfer of energy occurs between the boundary surface and the surrounding environment.	
Convection	Specifies the transfer of energy between the boundary surface and the fluid moving over the boundary surface.	
	Heat Transfer Coef.	Specifies the heat transfer coefficient. You can change the units using the drop-down menu.
	Ambient Temperature	Specifies the ambient temperature. You can change the units using the drop-down menu.
Heat Flux	Specifies the rate of energy transfer through the boundary surface.	

Property	Description	
	Heat Flux	Specifies the heat flux. You can change the units using the drop-down menu.
Temperature	Specifies the temperature on the boundary surface.	
	Temperature	Specifies the temperature on the boundary surface.

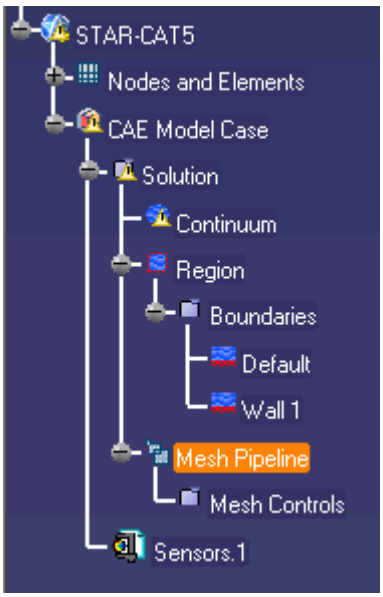
Meshing the Model

STAR-CAT5 provides options for generating polyhedral or trimmed volume meshes, with or without near-wall prism layers.

By default, prism layers are generated on all wall surfaces. The provision of a prism layer depends on the type of boundary that is applied to a face. As a consequence, it is recommended that you only generate the mesh after setting up the boundaries and regions.

The meshing process first creates a triangulated surface (the surface mesh) and thereafter converts it to the desired volume mesh (polyhedral or trimmer). As the quality of the volume mesh depends to a large extent on the quality of the surface mesh, it is prudent to check the surface mesh before proceeding with the volume mesh.

Although the entire meshing process can be launched with a single command (the Build And Run command from the STAR-CAT5 menu), it is possible to execute only single stages from the STAR-CAT5 **Mesh** menu. Settings that are made in the **Mesh Pipeline** node govern the mesh generation process.



A mesh pipeline contains all the settings that are applied to the surface and volume meshes created on the CATIA parts to which it is applied. In single part mode, only one mesh pipeline is available. In assembly mode, multiple mesh pipelines can be created if necessary.

When mesh generation commences, each mesh pipeline will:

1. Extract geometry from CATIA.
2. Transfer the model to STAR-CCM+.
3. Execute the surface remesher.
4. Execute the polyhedral or trimmer volume mesher with the prism layer mesher (if selected).
5. Create a mesh representation in the CATIA *Graphics* window.

Controls on the top-level mesh pipeline node allow you to set the base sizes for the surface mesh, and control the prism layer. Further refinement to specific faces, parts, or volume shapes can be made by adding additional controls to the mesh pipeline.

Contents:

Creation of Additional Surfaces in STAR-CCM+
Working with Mesh Pipelines
Working with Mesh Controls
Generating the Mesh STAR-CAT5
Common Causes for Meshing Failures

Creation of Additional Surfaces in STAR-CCM+

STAR-CCM+ creates geometry part surfaces during the geometry transfer stage.

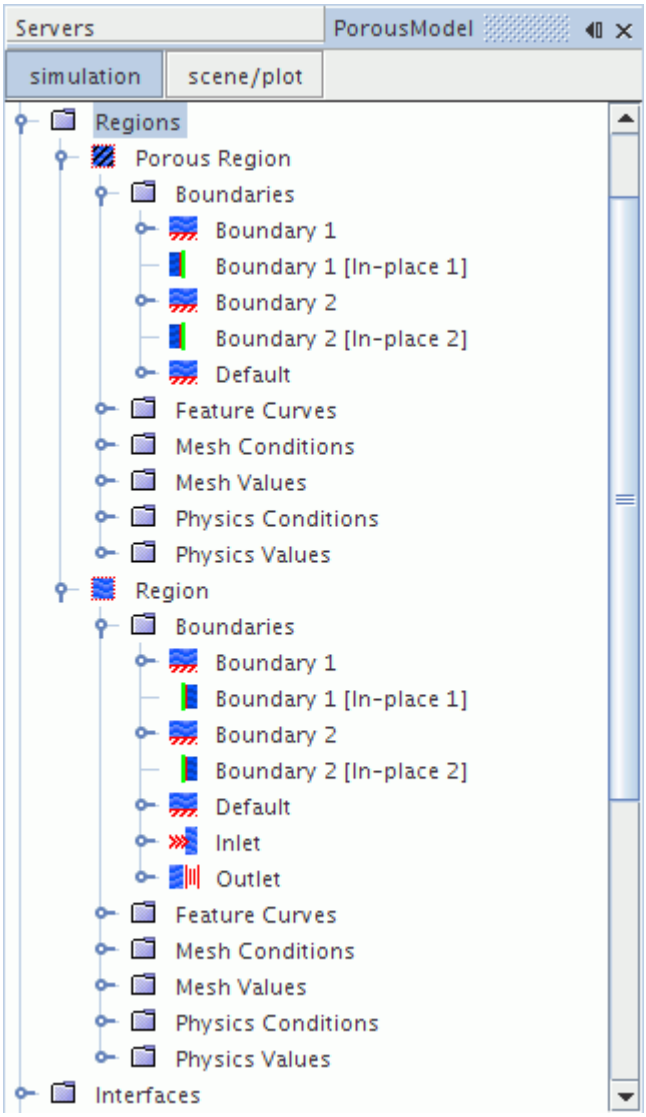
Initially, STAR-CAT5 creates one part surface for each face in the CAD model. These part surfaces are then combined so that one surface per boundary remains. This outcome, however, is not possible for boundaries that contain CAD surfaces from more than one part. The final part surfaces appear in the **Geometry** > **Parts** node in STAR-CCM+.

The following items affect the number of part surfaces that are created in STAR-CCM+:

- **The number of boundaries present in the model, including the default boundary.** Each boundary in STAR-CAT5 appears as a separate part surface in STAR-CCM+. If, however, a boundary contains CAD surfaces from multiple parts, the part surfaces in STAR-CCM+ cannot be combined.
- **The number of mesh pipeline controls used (if any).** Each CAD surface that is used to define a face mesh control in STAR-CAT5 appears as a separate part surface in STAR-CCM+.
- **The number of interfaces that are created during the meshing process (for assemblies only).** All the CAD surfaces that define one interface between two parts in an assembly are grouped together under one part surface in STAR-CCM+. One part surface exists for each interface in STAR-CCM+.

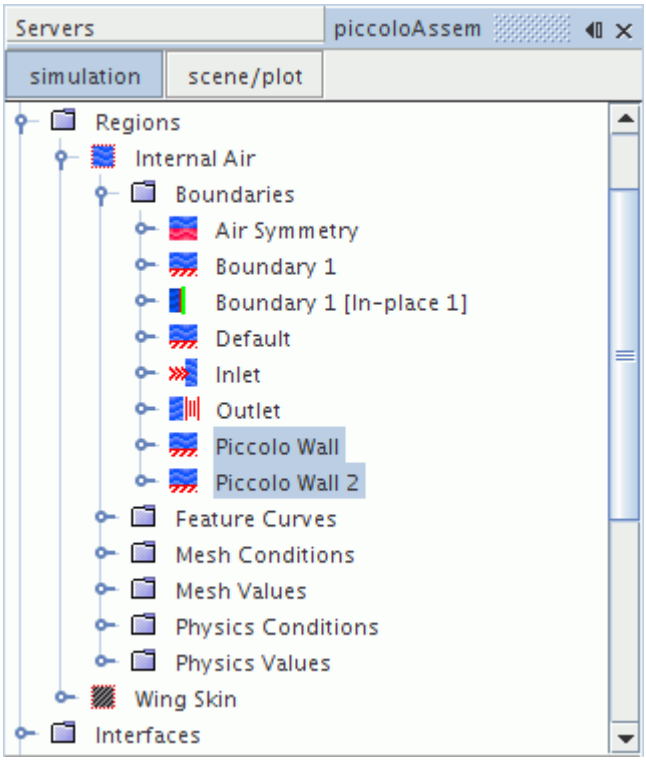
Boundary Splitting due to Interface Generation

Interfaces are automatically created between two parts in an assembly during the mesh generation. A new boundary is created for each interface on the surfaces that share the interface.



Boundary Splitting due to Face Mesh Controls and Part Mesh Controls

If you use a face mesh control or a part mesh control in your simulation, some boundaries may be split, depending on what faces are included in the mesh control. If a boundary and mesh control share the same faces, no splitting occurs. If a mesh control contains faces from multiple boundaries, splitting occurs and a new boundary is created. The name of the new boundary is derived from the original boundary.



Working with Mesh Pipelines

When working with a single part, you will only have one mesh pipeline defined in the model tree. This will specify the mesh settings for the single part available. In assembly mode, multiple mesh pipelines can be created to give specific controls for different parts, and parts must be specifically associated with a mesh pipeline.

Contents:

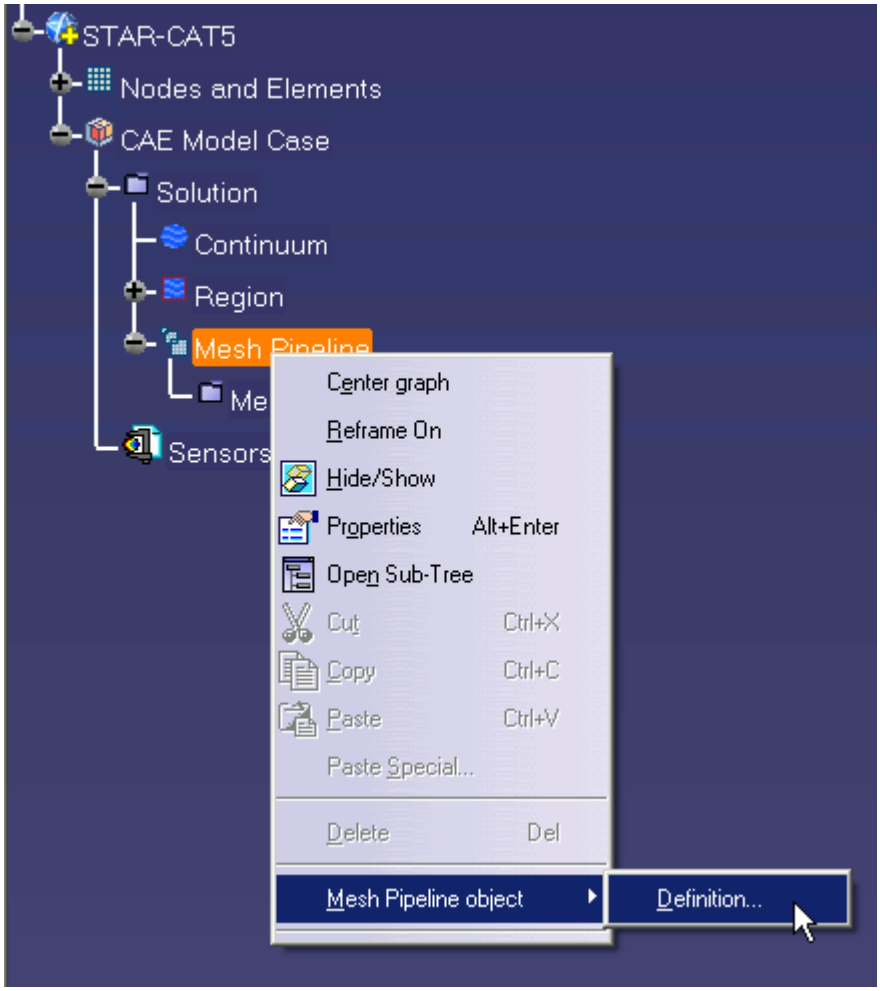
Setting Up the Mesh Pipeline

Setting Up the Mesh Pipeline

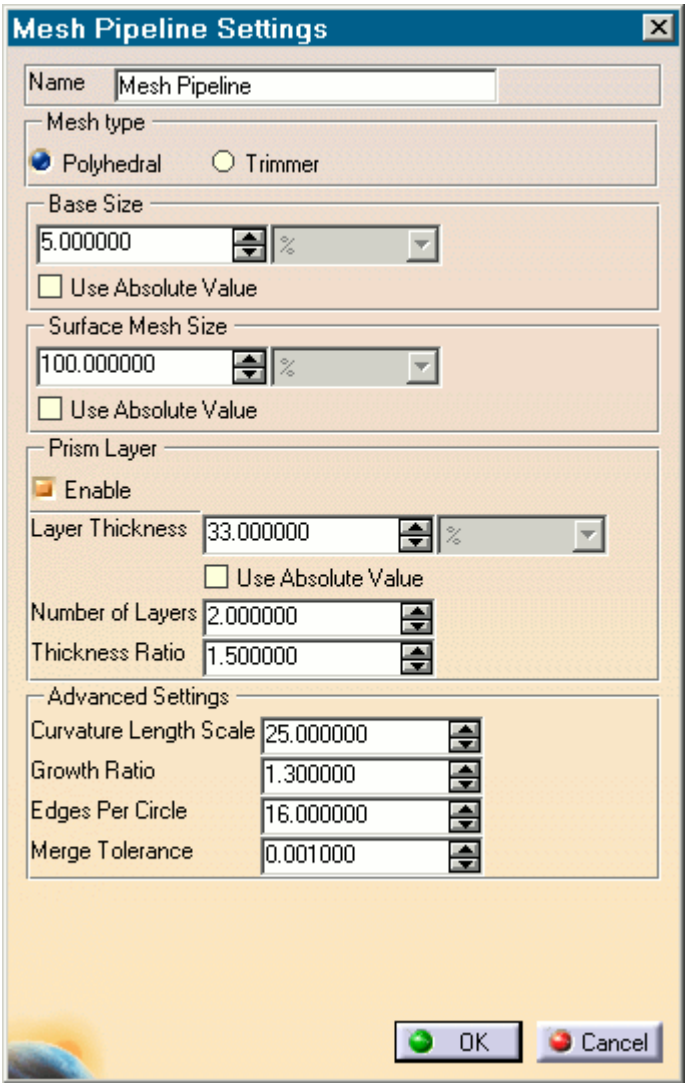
The mesh pipeline node contains the general mesh settings that will apply to all STAR-CAT5 parts associated within the pipeline in the absence of any further controls.

To set up the mesh pipeline:

1. Right-click the **Mesh Pipeline** node in the STAR-CAT5 tree and select **Mesh Pipeline object** > **Definition**.



The *Mesh Pipeline Settings* dialog appears.



2. In the *Name* field, enter the name of the mesh pipeline if required.
3. In the *Part Selection* box, specify the desired parts to use in the mesh pipeline.

   This property is only available when working with assemblies.
4. From the *Mesh Type* tab, select the type of mesh that you want to generate:
   - **Polyhedral**
   - **Trimmer**
5. Set *Base Size* to the desired value.

   The base size is a reference length used in other meshing parameters. It may be set to equal the length of the largest triangle you expect to have on your surface mesh, or to some other characteristic dimension.

   The base size can be set as either a relative or absolute value. When it is set as a relative value, the reference size will be the diagonal of the bounding box enclosing the entire geometry. In the case of multiple parts in an assembly, the reference size will be calculated based on the diagonal length of the bounding box enclosing the entire assembly. This includes all parts from all mesh pipelines. The unit specification menu will become active when setting an absolute value.

6. Set *Surface Mesh Size* to the desired value.

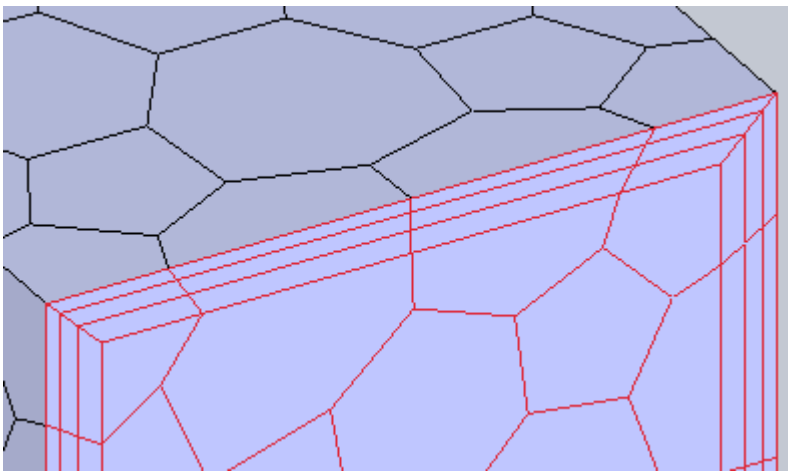
   The surface mesh size is the base edge length of triangles on the surface mesh. Note that the ultimate length of some (or all) triangles on the surface mesh will depend on other factors such as surface curvature and edge proximity.
7. Set the prism layer settings. See Prism Layer Panel.
8. Set the advanced settings. See Mesh Pipeline Advanced Settings Panel.
9. If you selected to generate a trimmed mesh, set the Trimmer settings. See Trimmer Panel.
10. Click **OK**.

Contents:

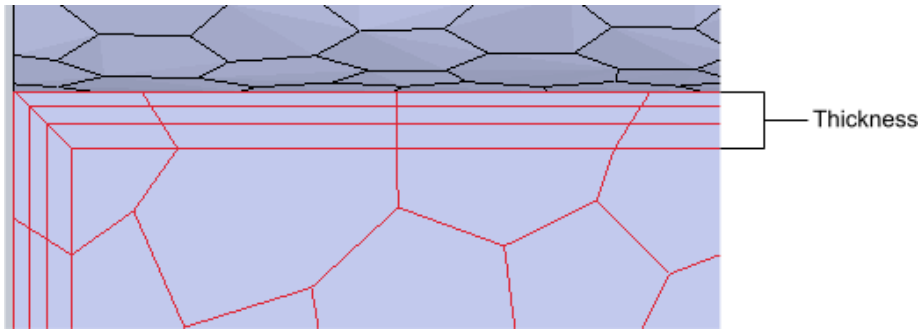
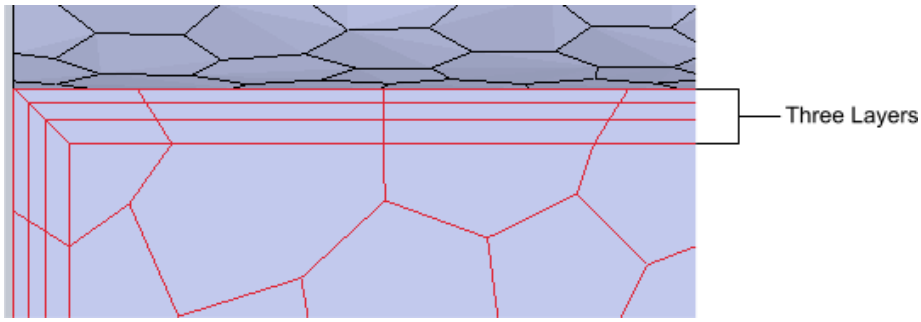
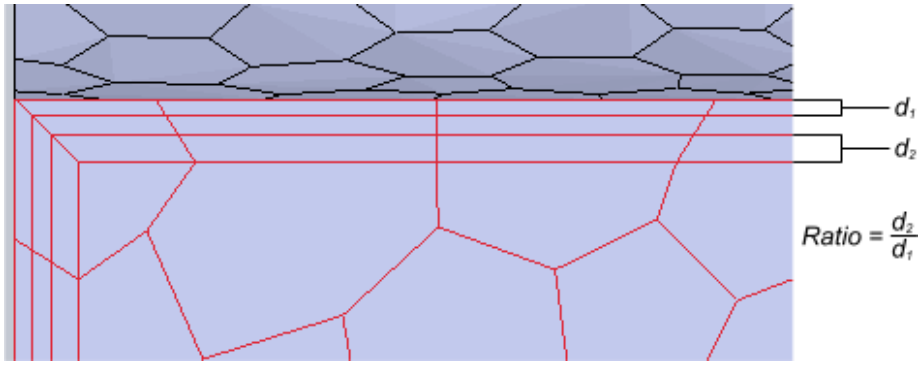
Prism Layer Panel
Mesh Pipeline Advanced Settings Panel
Trimmer Panel

Prism Layer Panel

Prims layers improve the simulation near non-slip walls, particularly when a turbulence model is active.



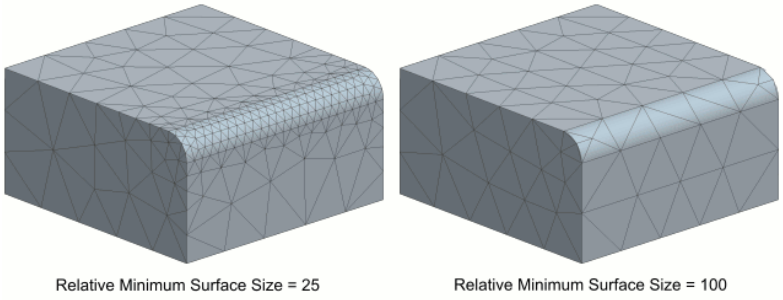
Property	Description
Layer Thickness	Defines the overall depth of the prism layer, which includes all of the individual layers, as illustrated below.

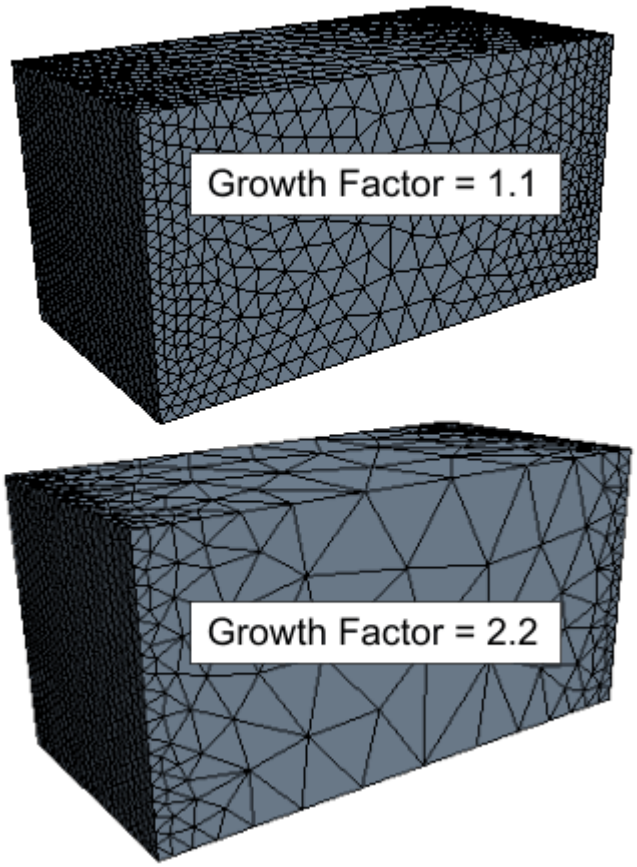
Property	Description
	
Number of Layers	Defines how many individual cell layers will be included in the overall prism layer, as illustrated below. 
Thickness Ratio	Specifies the size of the first and last layers within the prism layer. 

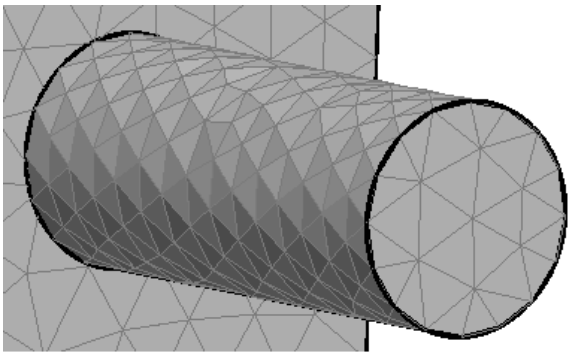
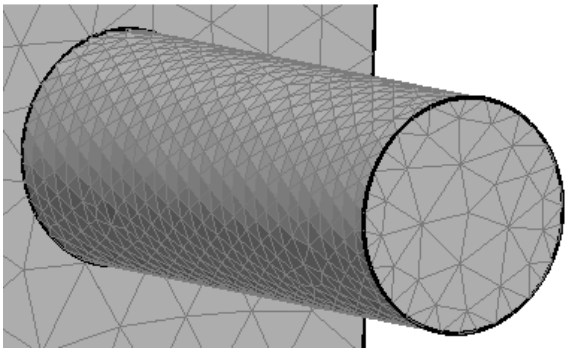
Mesh Pipeline Advanced Settings Panel

The advanced settings panel provides further customization of meshing parameters.

Property	Description
Relative Minimum Surface Size	Defines the minimum triangle size permitted on a feature curve. The minimum triangle size is set as a percentage of the surface

Property	Description
	mesh size. The default value is set to 25%. You must set the relative minimum surface size to a value greater than zero. If you set the relative minimum surface size to 100%, STAR-CCM+ will not modify the triangle size when it detects curvature. The figure below illustrates this behavior. Note that the base size was set to 20% in this example to emphasize the refinement on the curve.  Having a large value for the number of points on a circle property increases the mesh resolution on curved surfaces. The transition between the fine mesh and coarse mesh areas is controlled by the growth ratio.
Growth Ratio	Controls the rate at which triangle sizes change when moving from one triangle size to another. In the example below, the end faces of the block were set to have a smaller triangle size than the four side faces of the block. When the growth ratio is set to a low value, the transition between triangle size is very gradual. When a higher value is used, the transition is far more rapid.

Property	Description
	 The growth factor must be set to a value greater than 1.0. There is no upper limit for the growth factor.
Number of Points per Circle	Controls the number of triangle edges that appear on a curved surface, as shown below.

Property	Description
	 16 Points/Circle  32 Points/Circle Regardless of the number of points per circle defined, STAR-CAT5 will not permit the triangle edge length to shrink below 25% of the base size. The curvature length scale may also affect the number of edges on a curved surface.

Trimmer Panel

The trimmer mesh is controlled using the Trimmer Mesh Settings panel.

Property	Description
Default Growth Rate	Controls the stepping of the refinement from one cell size to the next. Five settings are available. By default, the *Default Growth Rate* is set to **Fast**, meaning that the fastest transition from the smallest cell size to the largest will be used. This usually equates to having only one layer of each cell size for each transition, which will result in the least number of cells being created for the supplied input values. The **Medium**, **Slow**, and **Very Slow** options further slow the transition from one cell size to the next in the mesh interior and therefore increase the overall cell count. In general, the medium option results in a minimum of two equal sized cell layers per transition, slow in four layers and the very slow option in eight layers.
Boundary Growth Rate	Controls the number of cell layers to grow next to surfaces in the template. Five settings are available. By default, the *Boundary Growth Rate* is set to **None**, meaning that the setting used for the Default Growth Rate is used

Property	Description
	for controlling all transitions. If the **Very Slow**, **Slow**, **Medium**, or **Fast** options are selected then it will result in eight, four, two or one layers of cells respectively of the same size being created next to the geometry surfaces in the template.

Working with Mesh Controls

Mesh controls allow you to define custom mesh settings for a particular section of the model.

Multiple controls can be assigned to the same item to control different mesh parameters. If, for example, a face has two mesh controls assigned to it--one that controls the mesh size and another that controls the prism layer thickness, the final mesh will respect both the custom mesh size and the custom prism layer thickness. If two mesh controls set the same mesh parameter, the control created last will take precedence over the first.

Contents:

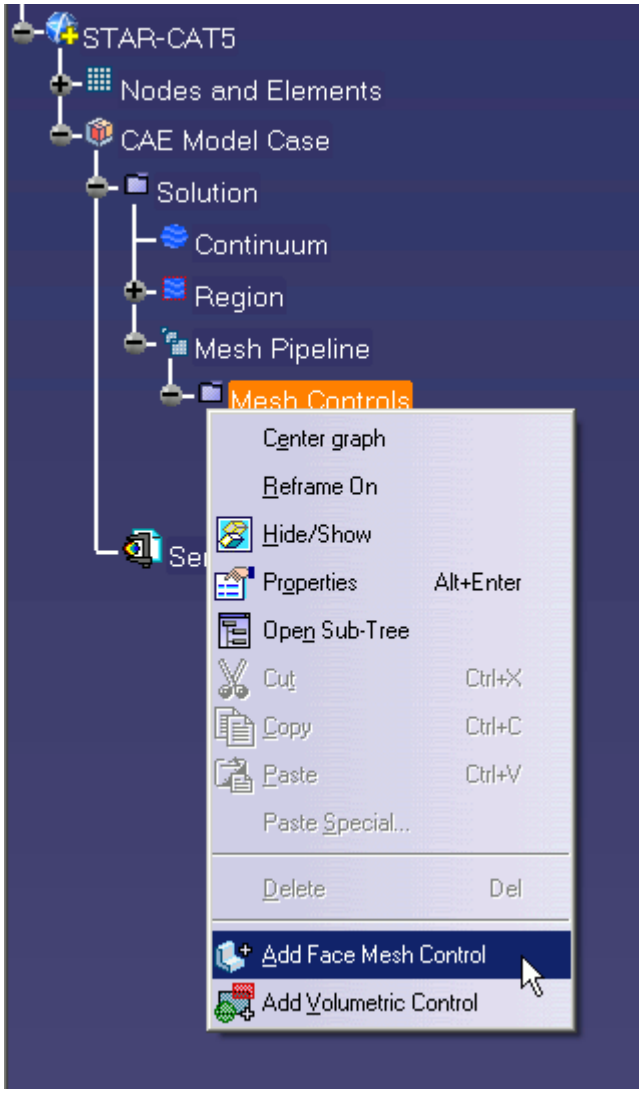
Using a Face Mesh Control
Using a Volumetric Mesh Control
Using a Part Mesh Control

Using a Face Mesh Control

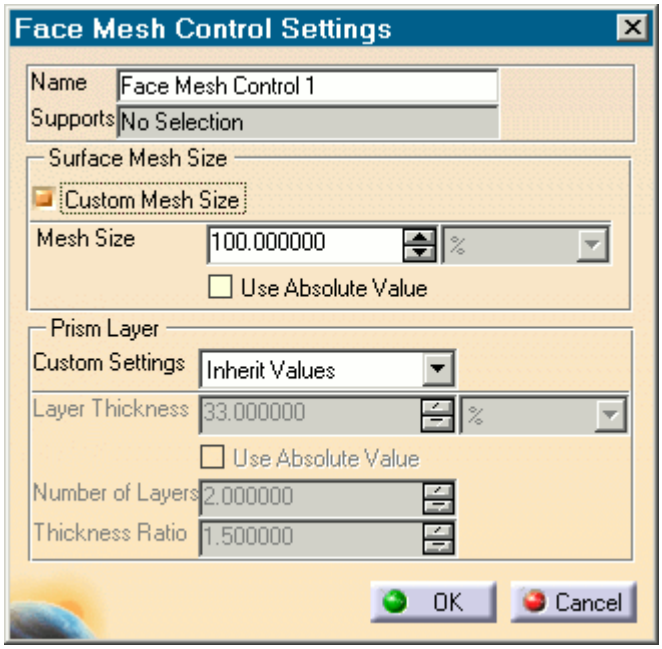
A face mesh control allows you to set the surface mesh size and the prism layer parameters on a face. Custom settings made on a face control will take priority over settings in the parent mesh pipeline, and over settings made in any part mesh controls.

To add a face mesh control:

1. Right-click the **Mesh Pipeline** > **Mesh Controls** node in the STAR-CAT5 tree and select **Add Face Mesh Control**.



The *Face Mesh Control Settings* dialog appears.



2. Set *Name* to an appropriate value.
3. On the CAD geometry, select the desired faces to include in this mesh control.
4. If you want to specify a custom surface mesh size on the selected surfaces, activate the **Custom Mesh Size** option.
   a) Set *Mesh Size* to the desired value.

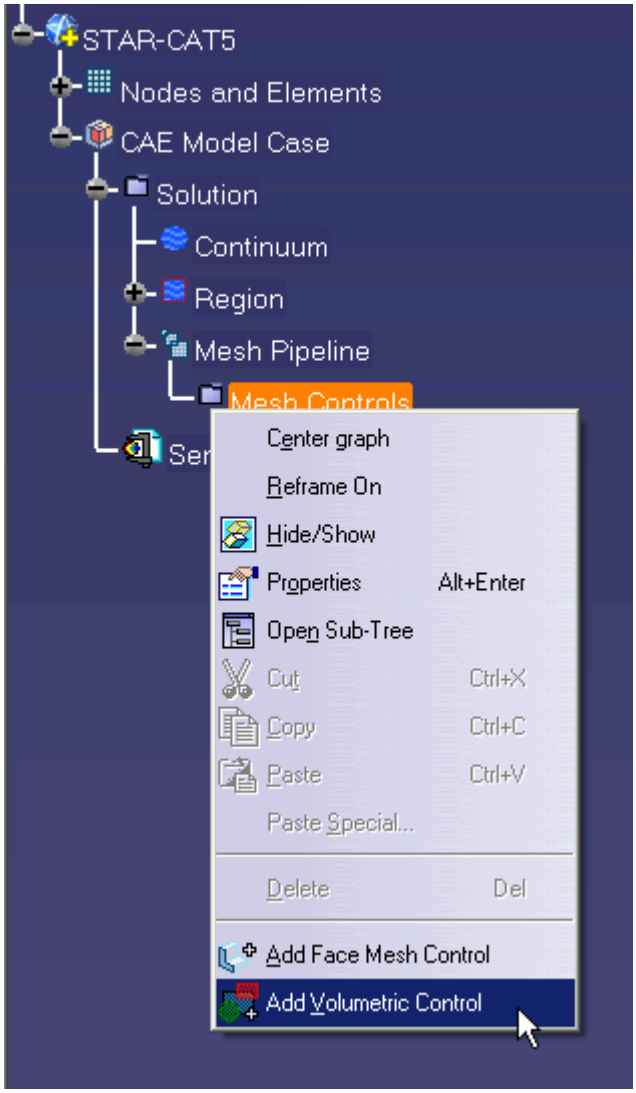
      You can enter this value as a percentage of the base size, or as an absolute value. If you select to enter the absolute value, you can select the units from the drop-down menu.
5. Set *Prism Layer Custom Settings* to the desired value:
   - **Inherit Values**: in the first instance, the face(s) will be assigned values from any part mesh control applied to the part to which the face(s) belong. If this does not apply, values will be taken from any applicable volumetric mesh control, and in the absence of this values will be taken from the parent mesh pipeline.
   - **Use Custom Values**: allows the prism layer parameters to be set directly on the face(s). See Prism Layer Panel.
   - **Disable**: no prism layer will be generated on the face(s).
6. Click **OK**.

Using a Volumetric Mesh Control

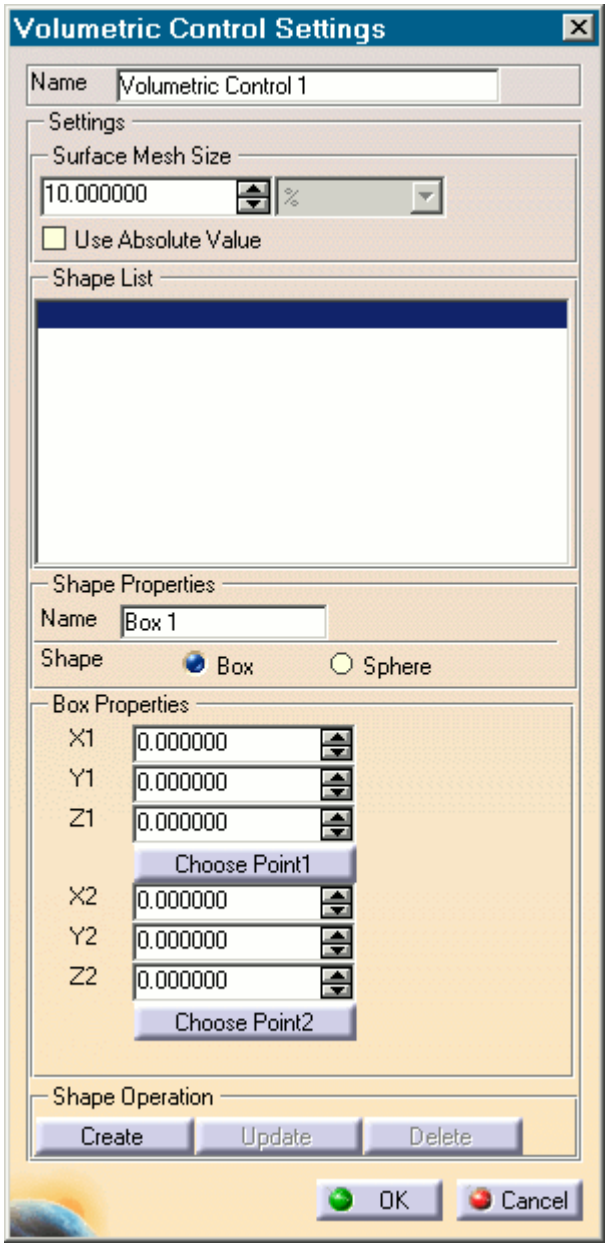
A volumetric control permits mesh refinement within a user-defined box or sphere. Only the mesh size can be modified with a volumetric control; prism layer parameters are not affected.

To add a volumetric mesh control:

1. Right-click the **Mesh Pipeline** > **Mesh Controls** node in the STAR-CAT5 tree and select **Add Volumetric Control**.



The *Volumetric Control Settings* dialog appears.



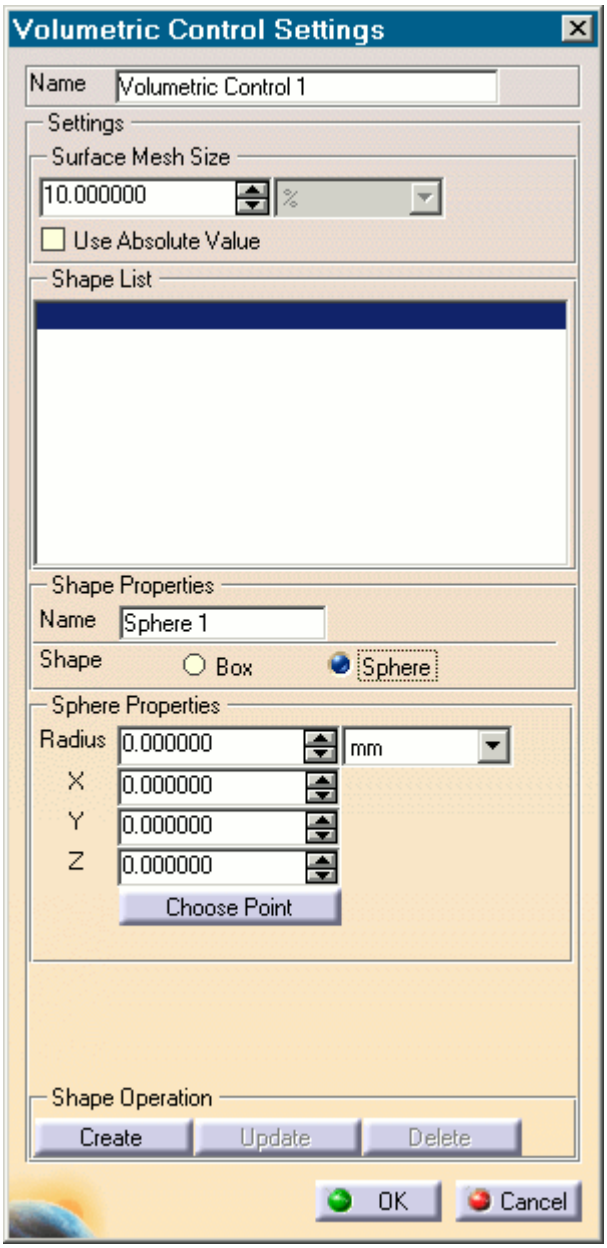
2. Set *Name* to an appropriate value.
3. Set *Surface Mesh Size* to the desired value.

   The mesh size defines the triangle edge length for surface mesh within the volume shape, and the characteristic cell size for volume mesh. The custom mesh size will still affect the volume mesh in cases where the volume shape does not intersect any surfaces of the Pro/E geometry.

   The mesh size can either be defined relative to the base size of the parent mesh pipeline, or as an absolute dimension with length units.
4. Select the type of volumetric control to create:
   - **Box**
   - **Sphere**
5. Enter the parameters for the volumetric shape:

- If you are creating a box, enter the coordinates for Point 1 and Point 2.

  X, Y, and Z coordinates of the two opposite corners of the box are required by the panel. These coordinates can either be entered manually, or taken from existing points within the Pro/E window. To use an existing point, click **Choose Point 1** or **Choose Point 2** and use the CATIA point selection operation to select the appropriate point.
- If you are creating a sphere, enter the coordinates for the center point and the radius of the sphere.



The sphere shape is defined by a center point and a radius. The center point can be set either by entering the coordinates directly, or by clicking the **Choose Point** button and using the Pro/E point selection procedure to choose an existing point in the Pro/E window. The coordinates of the existing point will be used to populate the panel.

6. Click **Create**.

   The shape is added to the *Shape List* list.
7. Click **OK**.

Editing a Volume Shape

To edit a volume shape

1. Select the volume shape from the list within the **Shapes** panel.
2. Edit the parameters as required and click **Update**.

Deleting a Volume Shape

To delete a volume shape

- Select the volume shape from the list in the *Shape* panel and click **Delete** .

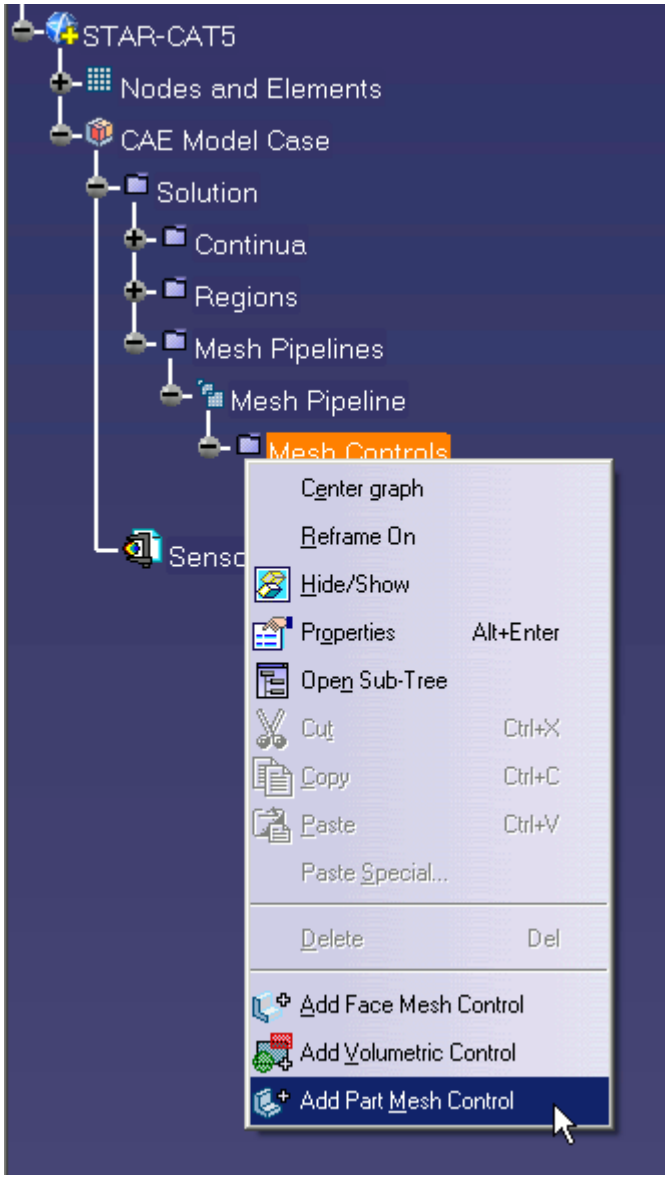
Using a Part Mesh Control

Custom mesh settings (such as surface mesh size and prism layer parameters) can be applied to one or more parts using a part mesh control.

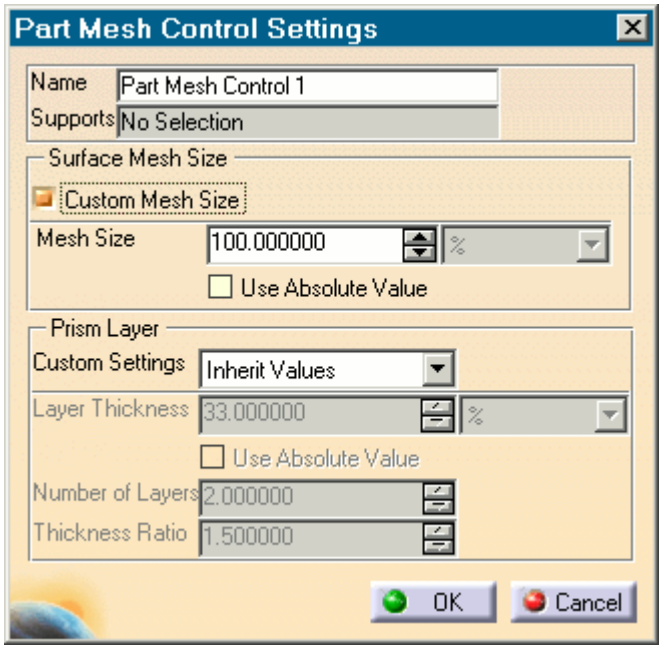
Settings made on a part mesh control will take priority over settings made on the parent mesh pipeline, but will not have priority over settings made on a face mesh control.

To add a part mesh control:

1. Right-click on **Mesh Pipelines** > **Mesh Pipeline** > **Mesh Controls** in the STAR-CAT5 tree and select **Add Part Mesh Control**.



The *Part Mesh Control Settings* dialog appears.



2. Set *Name* to an appropriate value.
3. On the CAD geometry, select the desired parts to include in this mesh control.
4. If you want to specify a custom surface mesh size on the selected surfaces, activate the **Custom Mesh Size** option.
   a) Set *Mesh Size* to the desired value.

      You can enter this value as a percentage of the base size, or as an absolute value. If you select to enter the absolute value, you can select the units from the drop-down menu.
5. Set *Prism Layer Custom Settings* to the desired value:
   - **Inherit Values**: for this option, prism layers on the surfaces of the part(s) will be generated based on settings taken from the parent mesh pipeline.
   - **Use Custom Values**: allows the prism layer parameters to be set directly on the part(s). See Prism Layer Panel.
   - **Disable**: no prism layer will be generated from the surfaces of the selected parts.
6. Click **OK**.

Generating the Mesh STAR-CAT5

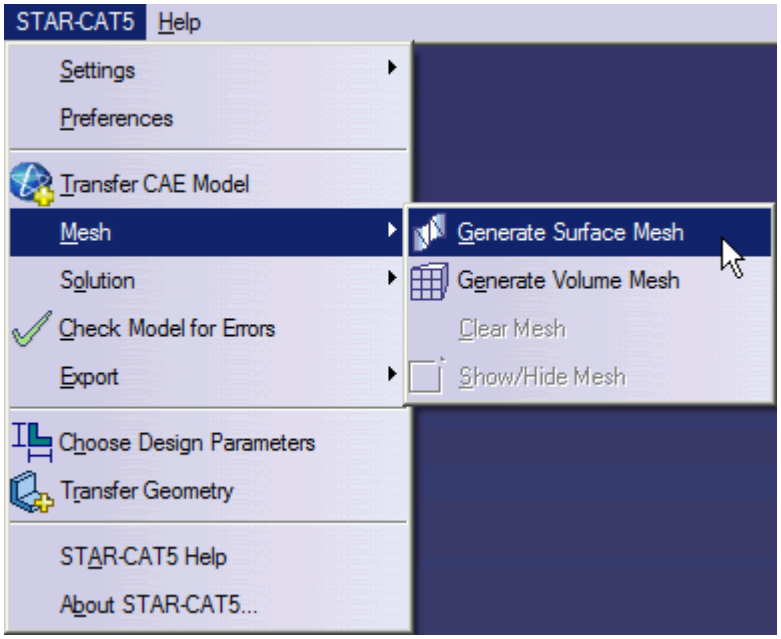
It is recommended that you first generate the surface mesh and inspect its quality before generating the volume mesh. Problems in the surface mesh will have a definite impact on the volume mesh, and may result in the volume mesher failing to complete.

Mesh generation will be carried out automatically when you issue the Build and Run command.

Generating the Surface Mesh in STAR-CAT5

To generate the surface mesh:

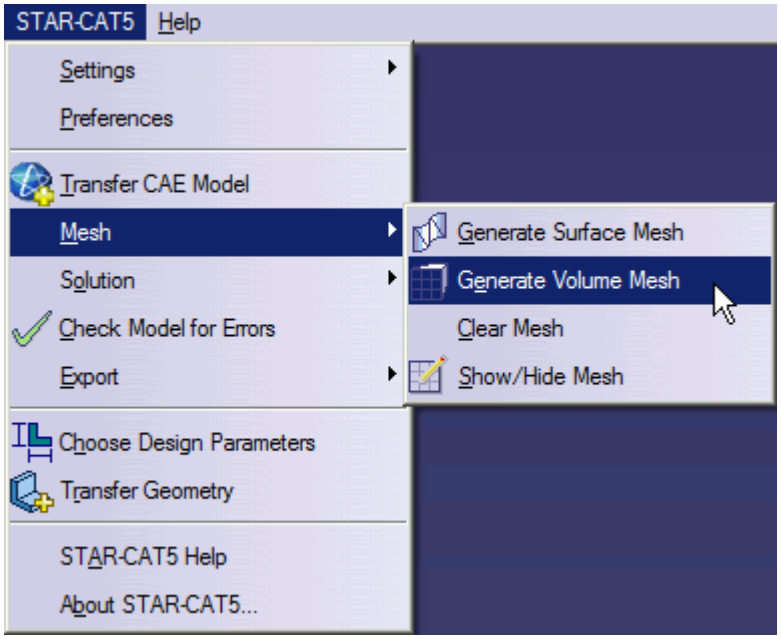
- From the STAR-CAT5 menu, select **Mesh** > **Generate Surface Mesh**.



Generating the Volume Mesh in STAR-CAT5

To generate the volume mesh:

- From the STAR-CAT5 menu, select **MeshGenerate Volume Mesh**.

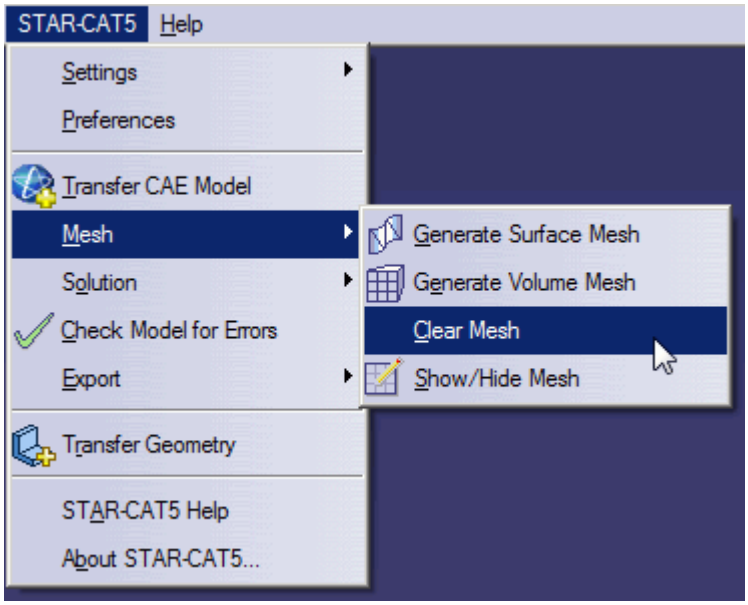


Clearing Generated Meshes

The **Clear Generated Meshes** action deletes all generated meshes without affecting the settings.

To clear the mesh:

- From the STAR-CAT5 menu, select **Mesh** > **Generate Volume Mesh**.



Contents:

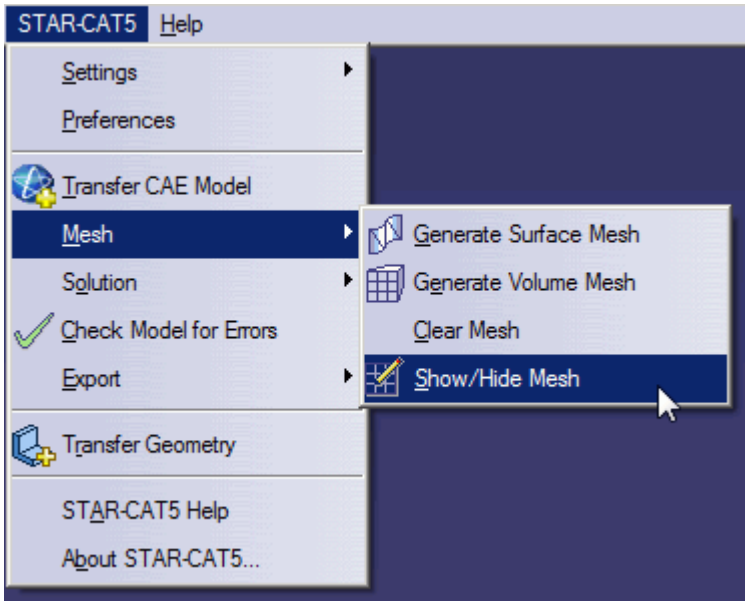
Visualizing the Mesh
Meshing Behavior at Interfaces

Visualizing the Mesh

Surface and volume meshes will automatically be displayed on first creation. If required, they can be hidden from view.

To hide a generated mesh in the geometry view:

1. From the STAR-CAT5 menu, select **Mesh** > **Show/Hide Mesh**.



Meshing Behavior at Interfaces

The volume mesher will treat interfaces between two regions differently depending on what type of region (fluid or solid) these are.

If the interface lies between two fluid regions, no prism layers will be grown from both sides of the interface. If an interface lies between a solid region and a fluid region, prism layers will be grown from the interface. The default behavior of the prism layer generation can be overridden using a mesh control, so prism layers can be disabled on a solid-fluid interface. See Working With Mesh Controls.

See Also:

Working with Mesh Controls

Common Causes for Meshing Failures

There may be instances when the mesh generation fails. In most cases this will occur during the surface meshing phase. STAR-CAT5 will try and identify the cause and suggest ways of rectifying this. This information will be printed in the *Output* tab of the STAR-CAT5 Info App window.

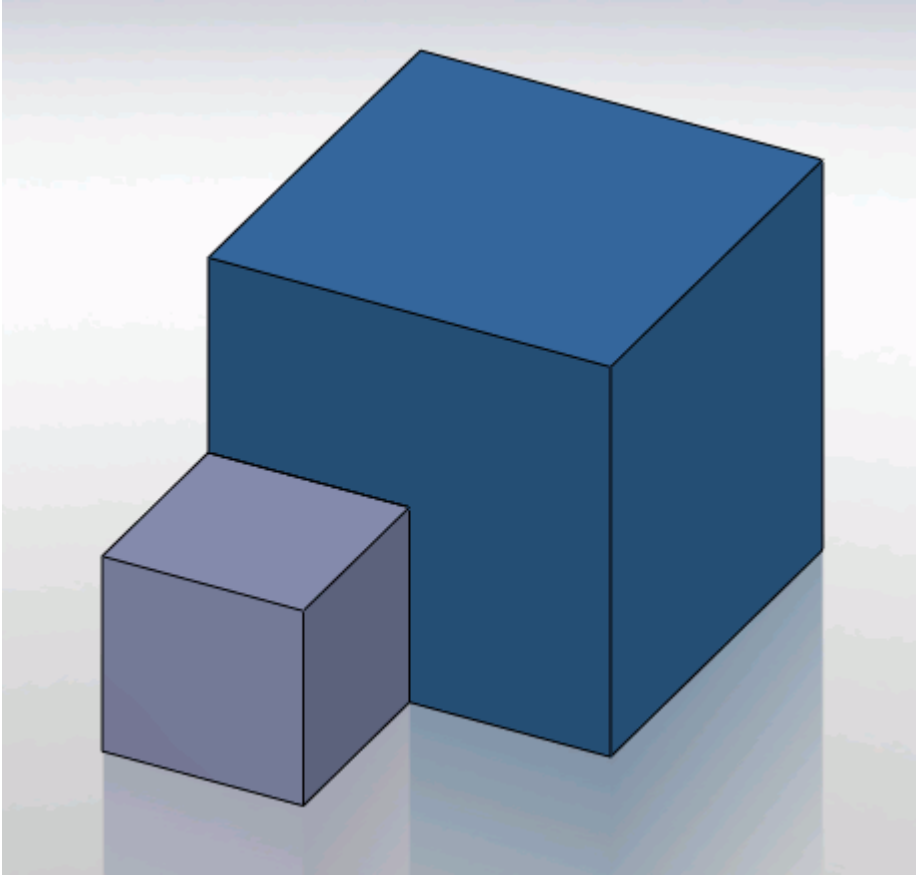
Contents:

The Parts in an Assembly are not Imprinted

The Parts in an Assembly are not Imprinted

The parts in an assembly should be imprinted together so that there is a common contact area between them.

A simple example of when imprinting is required is shown below.

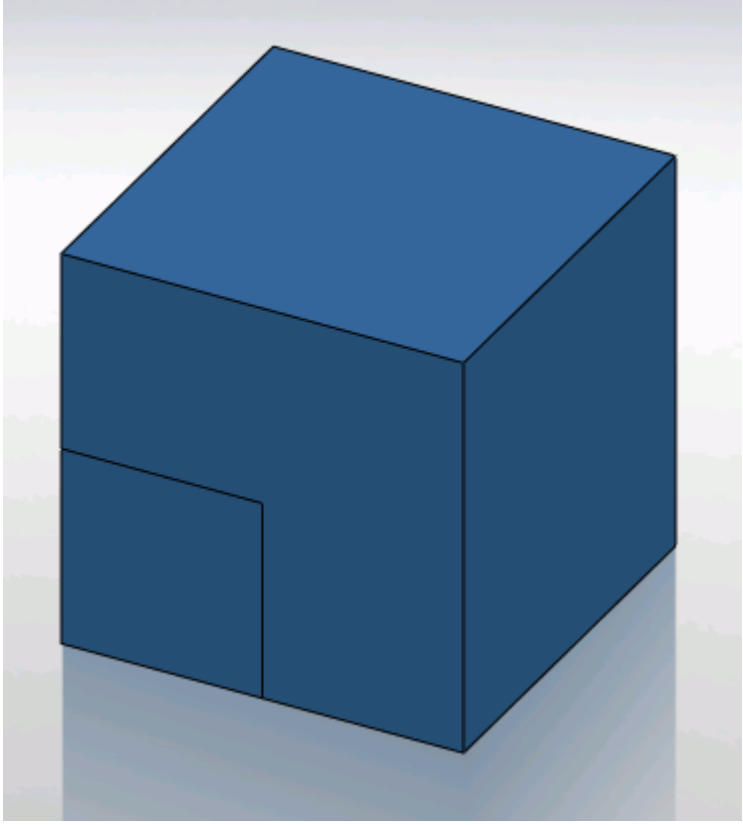


In this case the face of Part A (grey box) is splitting the face of Part B (blue box) causing this assembly to fail to mesh.

The output will display an error message.

```
Initializing meshing pipeline...
  All Geometry up to date.
Executing surface meshers...
Preparing surface for surface meshing in regions: Region 1, Region
Error during surface remeshing, deleting resulting mesh
An error occurred while attempting to orient the surface.
This is usually due to self-intersections in the surface. Check for problem(s) at :
  Geometric Range : (-0.0225836 0.00152109 0.0632873) and (0.0501103 0.0703362 0.0647127)
  Centroid of face : (0.0256414 0.0471603 0.064)
  Face type of face : 6
  Boundary = Default.
  Region = Region 1
Command: GenerateSurfaceMesh
   CompletedCommand: GenerateSurfaceMesh
   In: [Machine::main]
   Slave Message:
      Slave Name: Proc[0]
      Slave Rank: 0
   error: Server Error
```

The surface of Part B has to be imprinted with the contact face of Part A as shown below, after which meshing will complete successfully.



Obtaining a Solution

The following section details the execution of the STAR-CCM+ solvers in order to obtain a solution.

Contents:

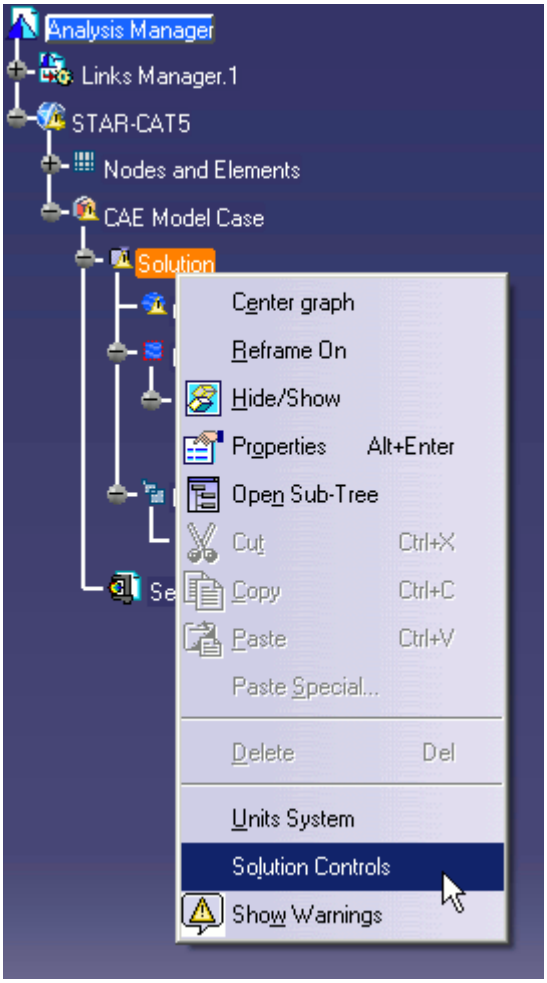
Setting Solution Controls
Checking the Model for Errors
Building and Running the Simulation in STAR-CAT5
Clearing the Solution
Visualizing the Solution
Setting Up Sensors
Working with Parameters

Setting Solution Controls

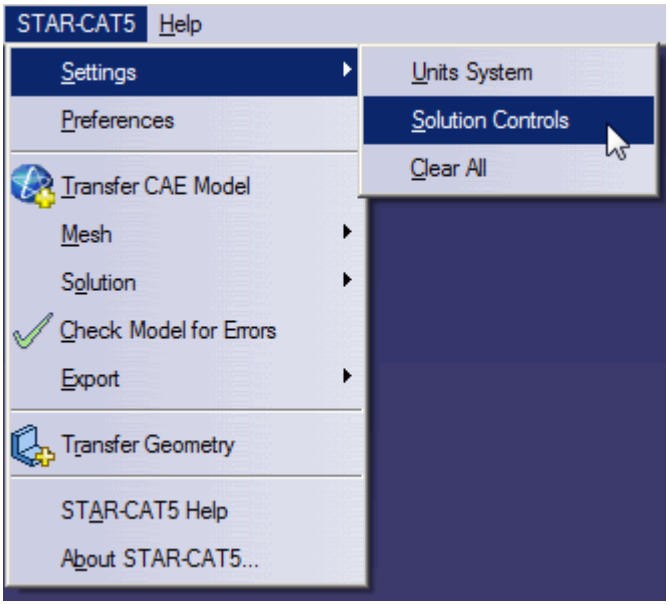
The solution you obtain from the solvers is a compromise between accuracy and processing time. You can manage the speed and accuracy of the solver from the *Solution Controls* dialog. The solution accuracy is also dependant on the mesh density.

To set up the solution controls in STAR-CAT5:

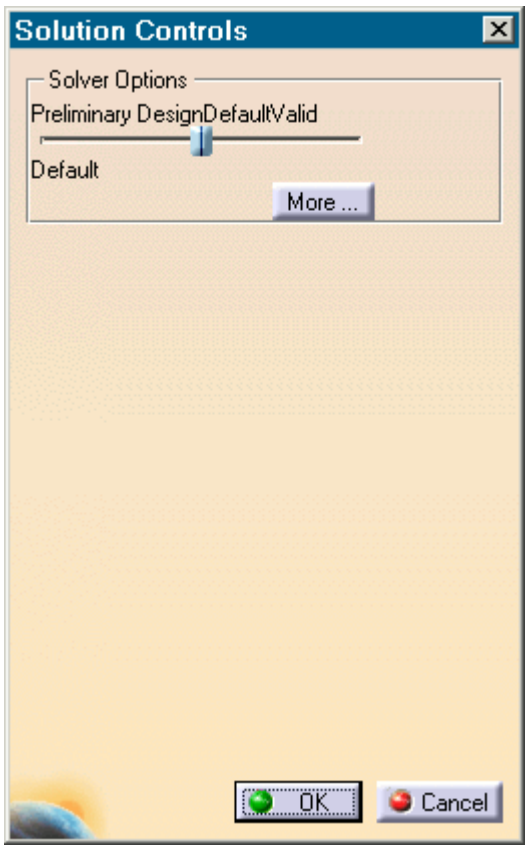
1. Either:
    - Right-click on the **Solution** node within the object tree and select **Solution Controls**.



- From the STAR-CAT5 menu, select **Settings** > **Solution Controls**.



The *Solution Controls* dialog appears.



2. Set the solution control parameters:
   - Using the slider, select one of the five pre-defined solver options. These options range from high accuracy and low speed, to low accuracy and high speed.
   - Click **More...** and set the solution control parameters manually. See Solution Controls Advanced Settings Panel.
3. If you want to run a Java macro with the simulation, activate the **Use Java File** option. See Using Java Macros with STAR-CAT5.
4. Click **OK**.

Contents:

Solution Controls Advanced Settings Panel
Using Java Macros in STAR-CAT5

Solution Controls Advanced Settings Panel

The advanced settings in the Solution Control dialog allow you to manually enter the values for the solution control parameters.

Parameter	Description
Stopping Criteria	
Iterations	Specifies the maximum number of iterations for which the solver can run. When the STAR-CCM+ solver reaches the specified number of iterations, it will stop. You can then post-process the results in STAR-CCM+. See Convergence Levels.
Convergence	Specifies how many orders of magnitude the residuals should drop before the solver stops. See Convergence Levels.
Solution Accuracy	
First Order	This option is generally the quicker, but less accurate solver option. The first order scheme introduces a dissipative error that is stabilizing and helps the solver achieve robust convergence. However, it can also lead to the neglect and merging of discontinuities, meaning the solution is not fully captured.
Second Order	This option is generally the more accurate, but slower solver option. The second order scheme attempts to minimize both dispersive and dissipative errors, providing a better representation of the simulation. However, in some situations, the lack of numerical dissipation can result in poorer convergence properties. It is common practice to use the first order scheme initially to ensure the simulation is running as expected and then switch to second order.
Under-Relaxation Factors	
Momentum	Specifies the under-relaxation factor for the momentum equation. See Solution Controls Advanced Settings Panel.
Pressure	Specifies the under-relaxation factor for the pressure-correction equation. See Solution Controls Advanced Settings Panel.
Turbulence	Specifies the under-relaxation factor for the turbulence equation. See Solution Controls Advanced Settings Panel.
Fluid Temperature	Specifies the fluid under-relaxation factor for the segregated energy solver. See Solution Controls Advanced Settings Panel.

Parameter	Description
Solid Temperature	Specifies the solid under-relaxation factor for the segregated energy solver. See Solution Controls Advanced Settings Panel.

Convergence Level

The numerical method used by the solver requires an iterative process in order to obtain a solution. After each iteration, residuals are produced that indicate how well the governing equations for each solver quantity are being satisfied numerically. In general, the residual magnitude should decrease as the solution converges. When the magnitude of the residuals for all the quantities falls below the convergence level, the solver will stop iterating, and results will be exported for post-processing.

Note: In most cases, a drop of 3-5 orders of magnitude in the residuals is sufficient to obtain an accurate solution.

You can adjust the convergence level parameter to meet accuracy or process time requirements. However, if the residuals are no longer decreasing from one iteration to the next, it is likely that machine accuracy has been reached for that case. For a case that does not exhibit decreasing residuals, it would be necessary to reconsider the setup in case the volume mesh is too coarse, or boundary conditions have been set incorrectly.

Under-Relaxation Parameters

Under-relaxation parameters determine how quickly the solver reaches a solution for a particular equation, on a scale from 0 to 1, with 1 being the fastest.

Ideally, all under-relaxation parameters would be set to 1, however, doing so would likely cause solver instability due to the complex nature of most simulations. Hence the under-relaxation parameters should be set to values that compromise speed with stability.

Note: Adjust the under-relaxation parameters if changes to other solution controls such as *Number of Iterations*, *Convergence Levels*, and *Solution Accuracy* have failed to improve the behavior of the solver.

Using Java Macros in STAR-CAT5

You can use Java macros with STAR-CCM+ to automate your simulation or to change a setting in STAR-CCM+, such as selecting an alternative turbulence model, before the solver starts running.

An example of how to change a setting in STAR-CCM+ using a Java macro is outlined below:

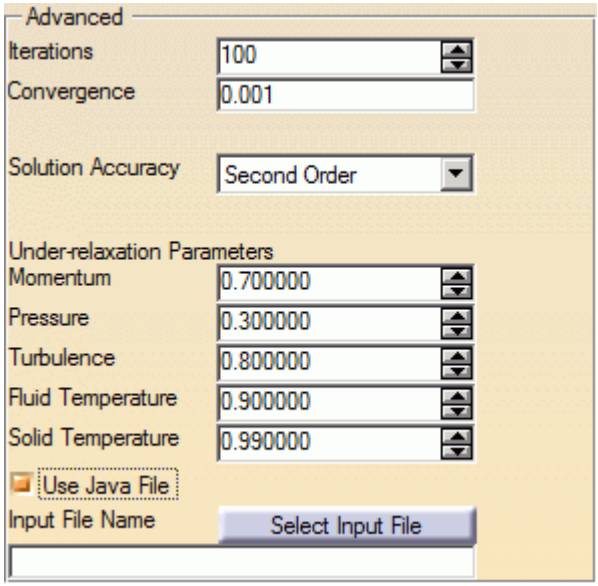
1. Launch STAR-CAT5 and set up the case.

   Define the continuum settings, boundary conditions, mesh settings, and solution control settings.
2. Launch STAR-CCM+.
3. In STAR-CCM+, start recording a macro and perform the desired operations, such as changing the physics models.
4. Stop recording the macro.

   The macro saves at this point.

5. Close STAR-CCM+ and do not save the simulation.
6. Return to STAR-CAT5, open the *Solution Controls* dialog and activate the *Use Java File* checkbox.

   A text entry field appears.



7. Click **Select Input File**.

   The *Open* dialog appears.
8. Navigate to your Java macro file and select it.

   Selected files appear in the text entry field. The path in the entry field is displayed relative to the current CAD geometry file.
9. Click **OK**.

When you run the simulation, STAR-CCM+ will launch and play the macro before starting the solver. This sequence allows you to set up and use additional STAR-CCM+ physics models that are not available in STAR-CAT5.

See Also:

Setting Up Continua

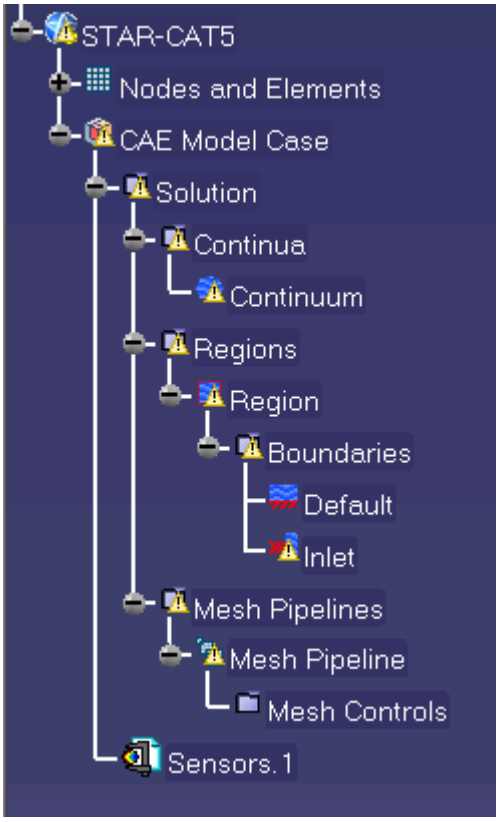
Setting Up Regions

Setting Boundary Conditions

Setting Up the Mesh Pipeline

Checking the Model for Errors

STAR-CAT5 monitors the active case for any errors in real-time and displays a warning triangle ⚠ next to any nodes for which there is a problem.

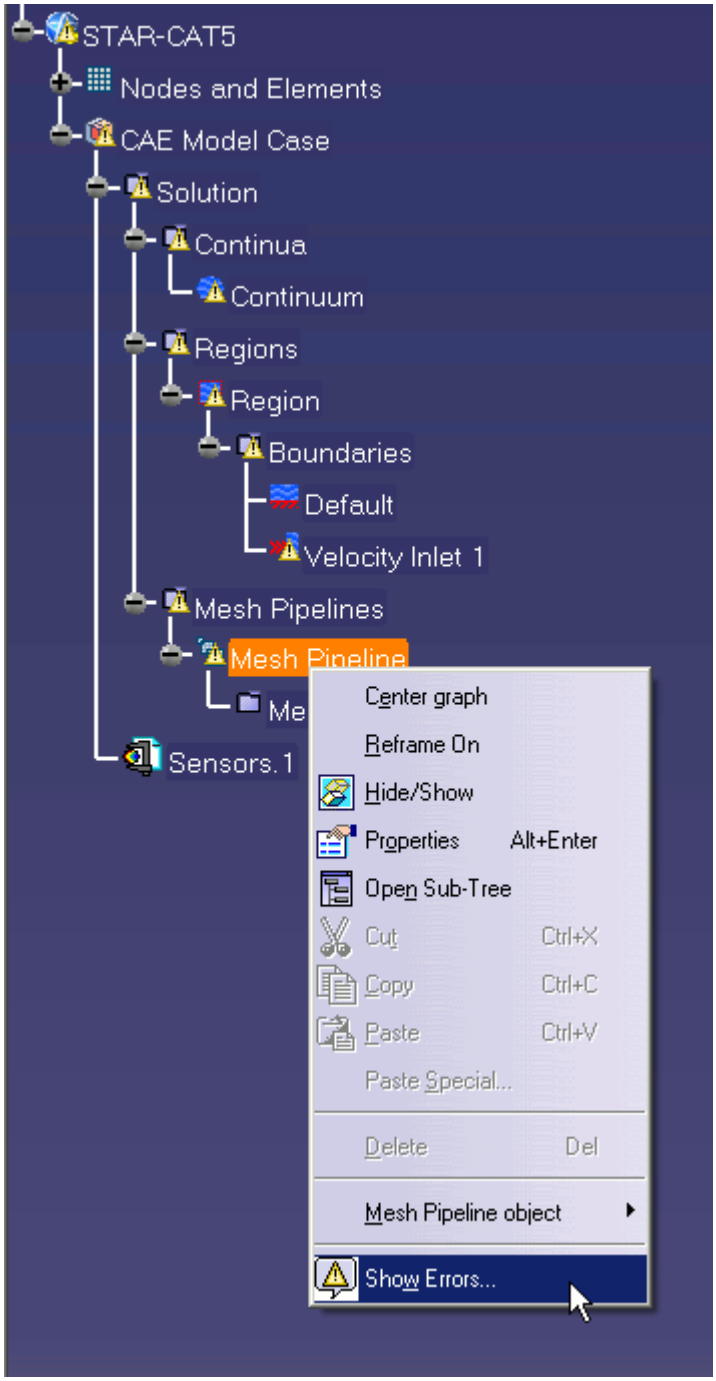


Three areas are monitored by the live model checker:

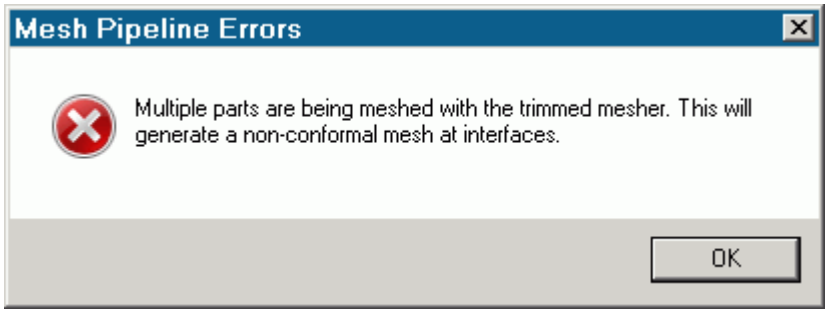
- **Continua**. The continua checker makes sure that all necessary boundaries are present in the case, such as the presence of at least one inlet and one outlet.
- **Regions and boundaries**. The regions and boundaries checker covers issues specific to boundaries, such as a velocity inlet boundary having a zero value.
- **Mesh pipelines**. The mesh pipeline checker covers issues that are related to the mesh, such as the presence of multiple parts in a trimmed mesh pipeline.

To see the cause of an error:

1. Right-click on an node displaying the warning triangle, and select **Show Errors...**



A dialog appears describing the error, as shown below.



2. Click **OK** to close the dialog.

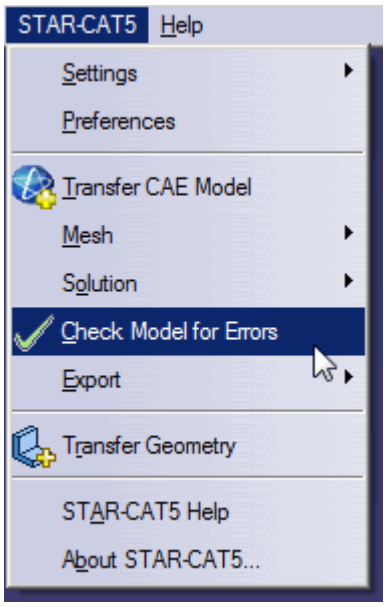
The **Show Errors...** right-click menu is available for any node in the STAR-CAT5 object tree that contains a warning triangle. Selecting it will show errors for that node and its sub-nodes.

To see a complete list of issues for the case, right-click on the STAR-CAT5 root node and select **Show Errors...**

You can also manually check for errors.

To perform a manual check:

- From the STAR-CAT5 menu, select **Check Model for Errors**.



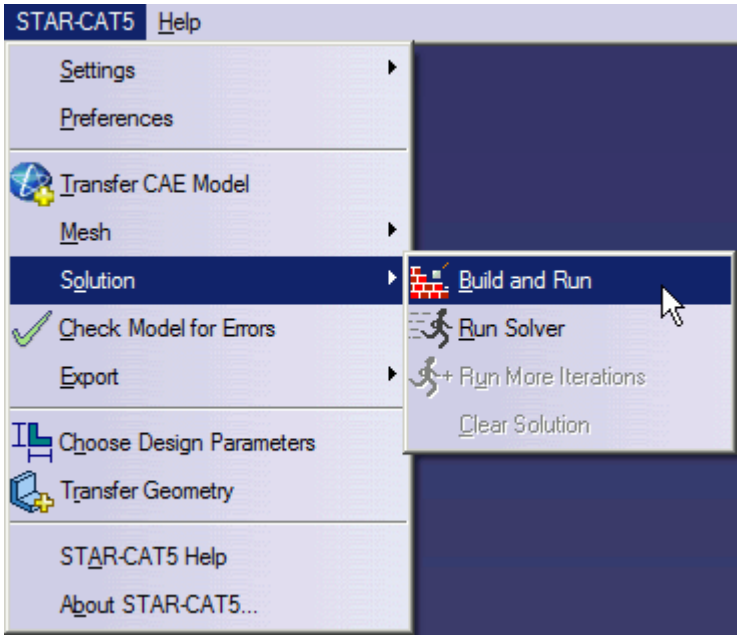
If errors are found in the case set up, these are displayed in the *Model Checker* tab of the *Info App* dialog. See The Model Checker Tab.

Building and Running the Simulation in STAR-CAT5

Once you have defined the model physics, boundary conditions, mesh settings, and solver settings in STAR-CAT5, you can build and run the case.

To build and run the case:

- From the STAR-CAT5 menu, select **Solution** > **Build and Run**.



This action automatically generates the mesh, and runs the solver. When you run the solver, a new *Residuals* tab is added to the *Info App* dialog. You can use this residuals plot to monitor solution convergence. When the solver finishes, STAR-CCM+ launches and a scalar scene is automatically created for you.

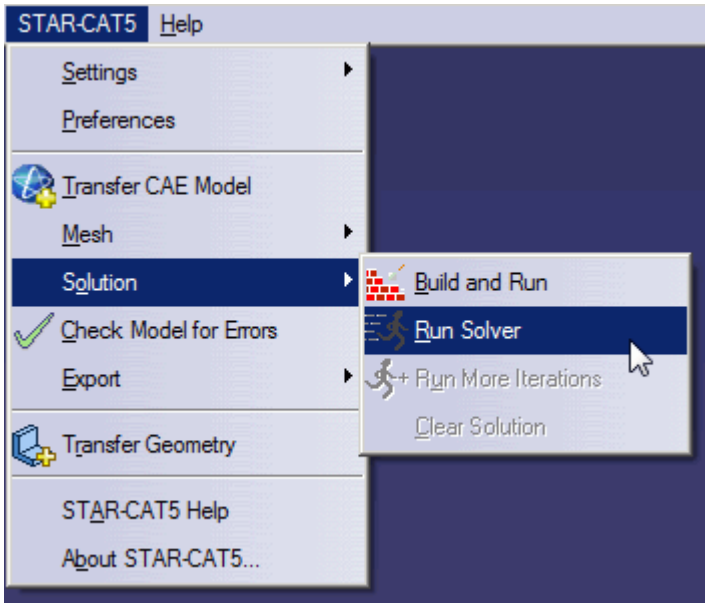
Note: If a solution has already been computed in STAR-CCM+, and the stopping criteria has been reached, clicking **Build and Run** clears the solution and starts running from the first iteration. If you want to run for more iterations, change the stopping criteria and click **Run Solver**, or click **Run More Iterations**.

Running the Simulation in STAR-CAT5

If you have already generated a volume mesh for the model, you can run the case directly.

To run the solver:

- From the STAR-CAT5 menu, select **Solution** > **Run Solver**.

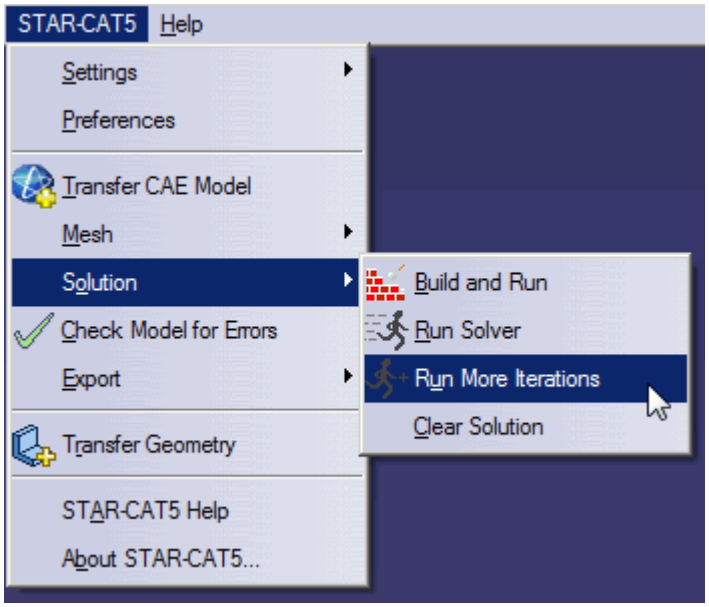


Running for More Iterations

You may need to run the STAR-CCM+ solvers for more iterations than you originally set in the Number of Iterations stopping criterion. The Run More Iteration function runs the solver for an additional number of iterations as specified in the stopping criteria. Take, for example, a case where you have already run the simulation for 300 iterations. You specify 500 iterations in the *Number of Iterations* stopping criterion. When you next click **Run for More Iterations**, the simulation will run for 500 extra iterations, making 800 iterations in total. The solver runs until one of the stopping criteria is satisfied.

To run for more iterations:

- From the STAR-CAT5 menu, select **Solution** > **Run for More Iterations** in the STAR-CAT5 menu.



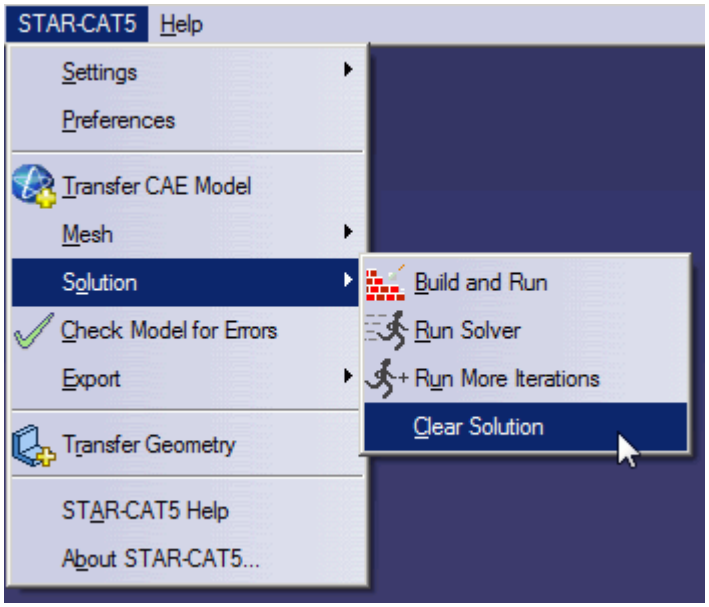
Clearing the Solution

You can clear all results that were calculated by the STAR-CCM+ solvers using the **Clear Solution** function. All other STAR-CAT5 simulation settings remain intact.

Caution: This action is irreversible.

To clear the solution:

1. From the STAR-CAT5 menu, select **Solution** > **Clear Solution**.



The solution that was calculated by the STAR-CCM+ solvers is deleted.

Visualizing the Solution

When STAR-CCM+ launches after the solution is calculated, it automatically displays a scalar scene of the model that shows the `Absolute Pressure` field function.

To visualize the results, you can use this scene to display different scalar quantities, or use any of the other visualization tools in STAR-CCM+, such as streamlines, plane sections and vector scenes.

You can import CAD parts into STAR-CCM+ for visualization purposes. These parts will not be taken into account during the analysis. This feature is useful for displaying the original geometry in a post-processing scene in STAR-CCM+. See Importing Geometry Into STAR-CCM+ for Visualization.

Setting Up Sensors

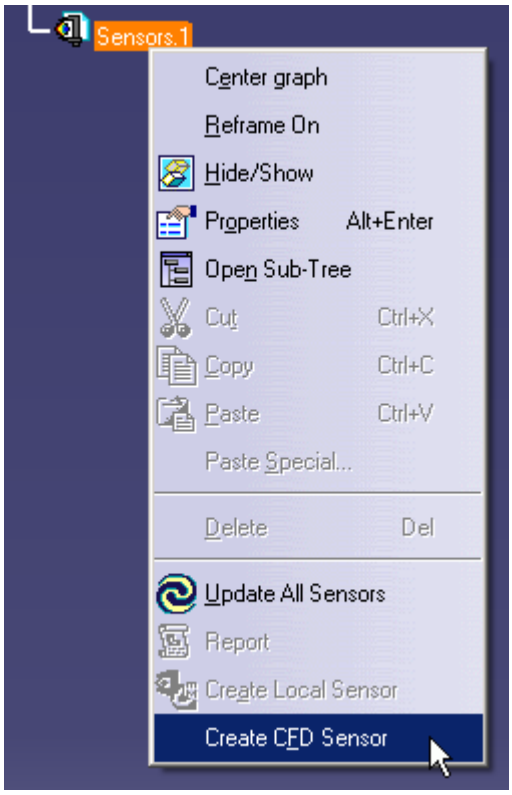
You can define sensors on boundaries in STAR-CAT5. A sensor is an object that can be used to monitor solution data that is created by STAR-CCM+. The sensor creates a CATIA parameter which can then be used within optimization loops.

Defining Sensors

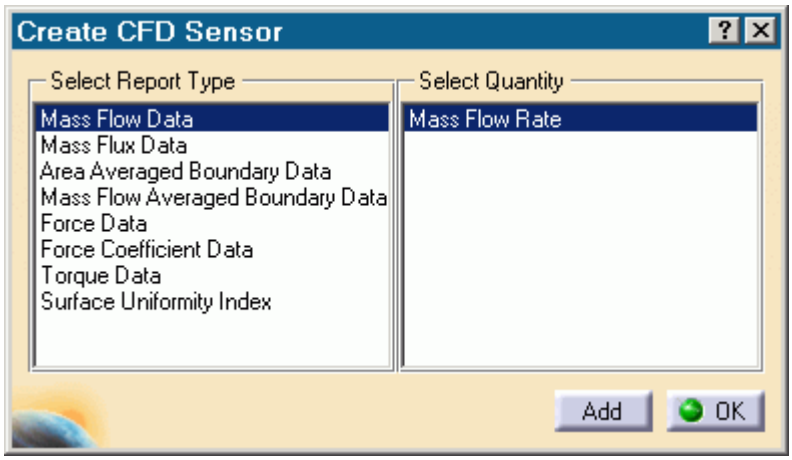
You can create a sensor either before or after running a case. If you create it before running the case, the sensor will be evaluated when the case is run. If you create it after running the case, you will need to update the sensor.

To create a sensor:

1. Right-click **Sensors.1** in the object tree and select **Create CFD Sensor**.



The *Create CFD Sensor* dialog appears.



2. Select the type of report and the quantity to monitor.

3. Click **Add**.

   A sensor object is created and added to the object tree.

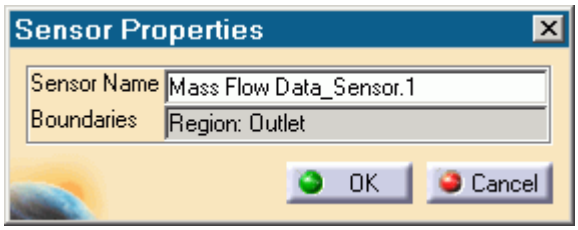
   Note: You can create multiple objects by clicking **Add** multiple times.

4. Click **OK** to close the dialog.

   The sensor object contains an ! icon. This icon indicates that the sensor definition is incomplete.

5. Double click the sensor to edit it.

   The *Sensor Properties* dialog appears.



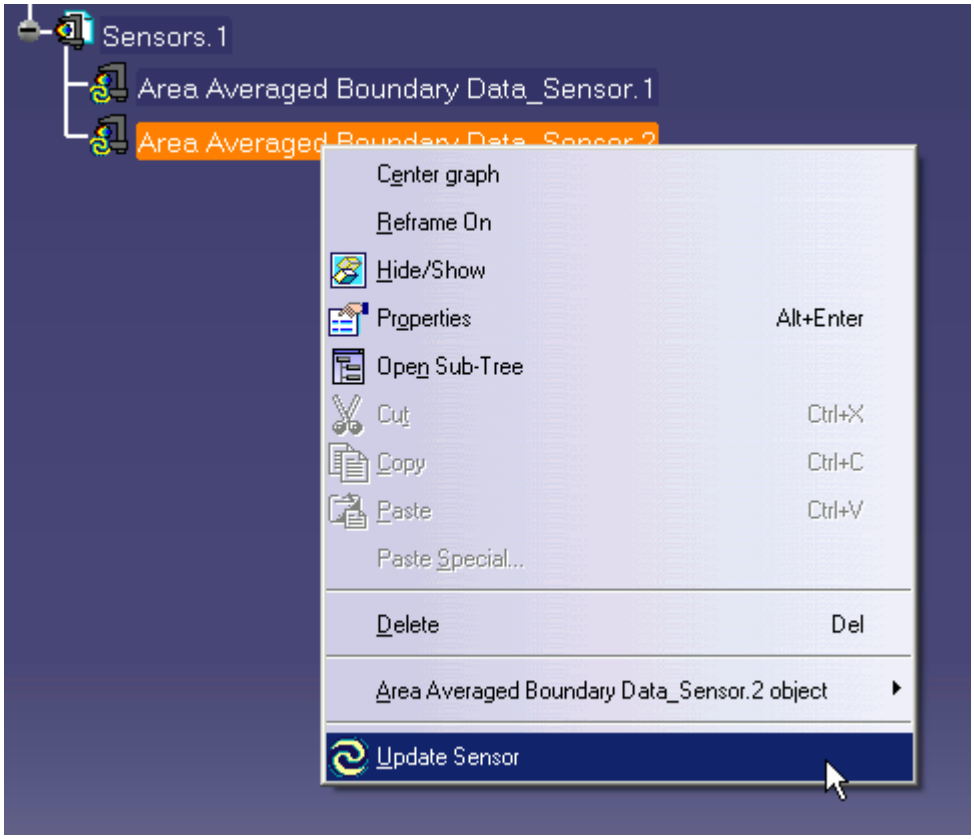
6. Rename the sensor to something more appropriate in the *Sensor Name* property.
7. Select the boundaries that you want to associate with the sensor from the STAR-CAT5 object tree.
8. Click **OK**.

The sensor icon in the object tree will show if it requires an update. The sensor will be evaluated when the case is run or when an update is requested.

Updating Sensors

To update the sensor:

1. Right-click on the sensor in the object tree and select **Update Sensor.**



2. Click **OK** to dismiss the warning dialog.

The data for the sensor is calculated and a new report sub-node is added to the sensor.

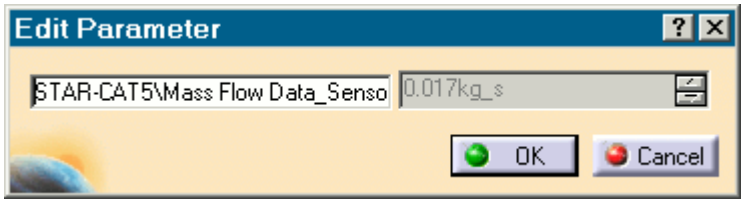
Note: The units that are used in the sensor are CATIA working units, not STAR-CAT5 units.

Viewing the Sensor Data

To view the sensor data after the solver has finished running:

1. Double-click on the new sensor report object in the feature tree.

The *Edit Parameter* dialog appears, displaying the report value.



2. Rename the report.
3. Click **OK** to close the report.

Working with Parameters

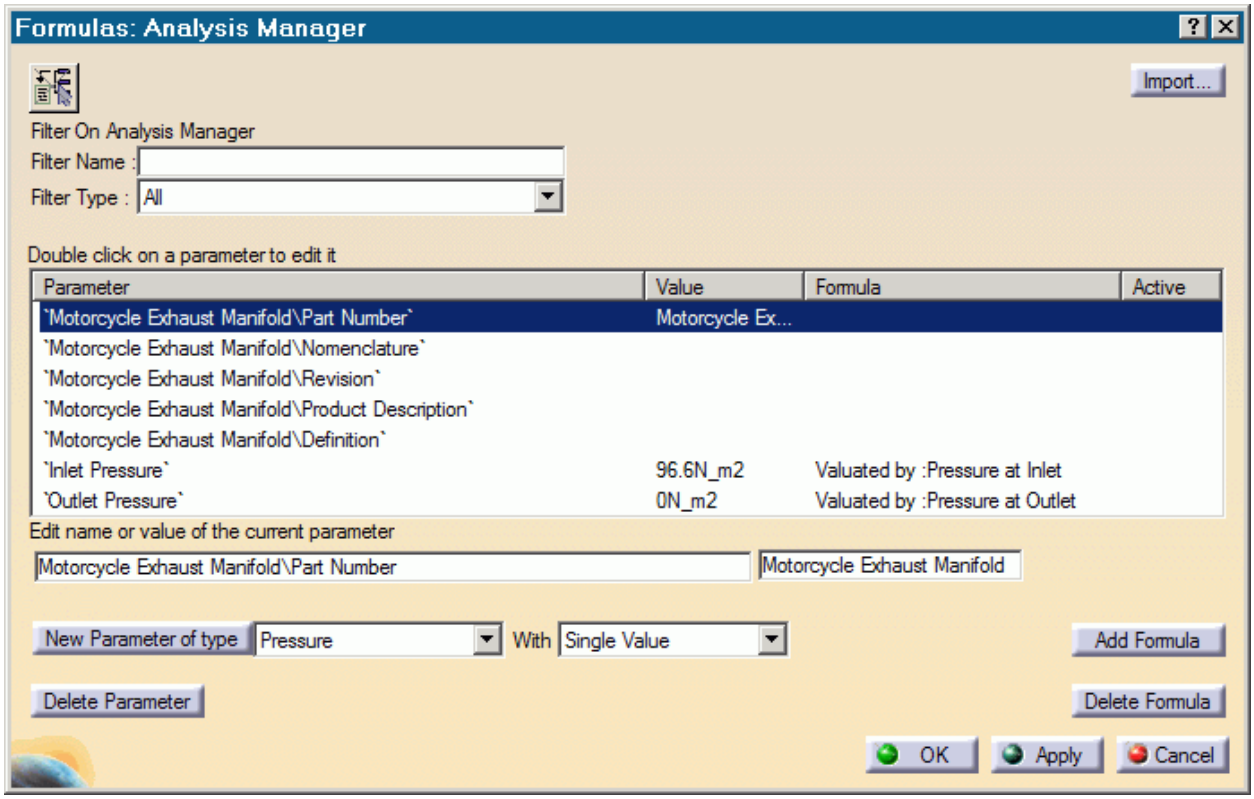
You can use CATIA's formula editor to add numerical data together. You can find this feature in CATIA's *Knowledge* toolbar.

To create a formula:

1. Click the **Formula** button.

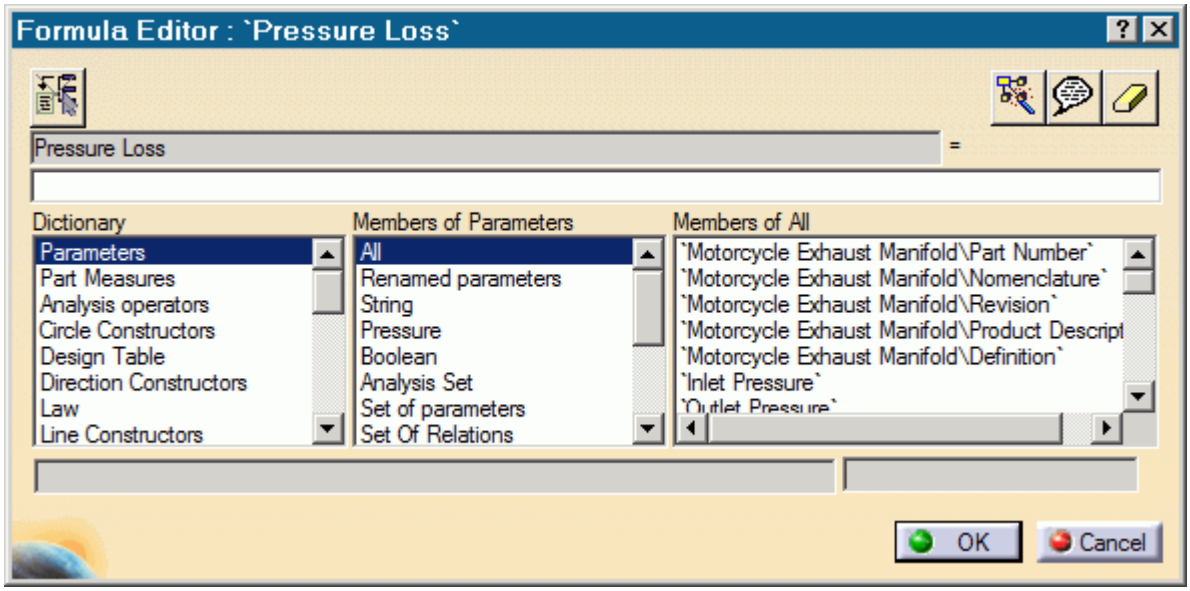
The *Formulas* dialog appears.

In the following example, pressure loss is calculated using the `Area Averaged Pressure` at the Inlet and Outlet.

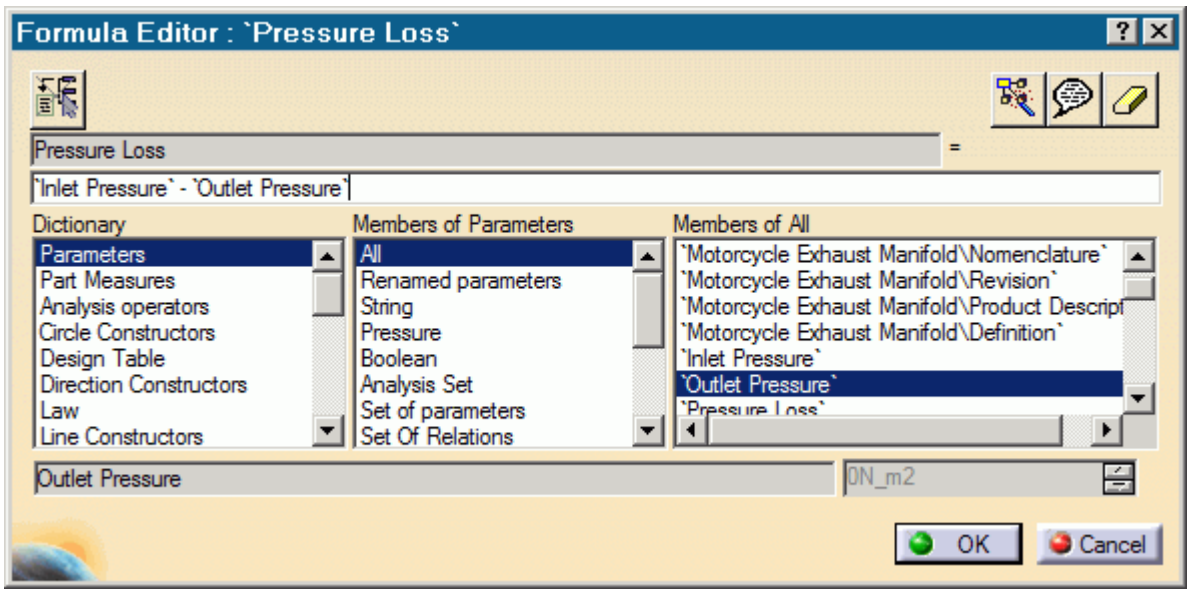


2. Next to **New Parameter of type**, select **Pressure** from the drop-down list.
3. Click on **New Parameter of type** to add a new pressure parameter to the list.
4. Give this new parameter a suitable name, such as `Pressure Loss`.
5. Click **Add Formula**.

   The *Formula Editor* dialog appears.



6. Enter the desired formula, in this case the pressure at the inlet minus the pressure at the outlet.



Note: You can add sensor report objects to the formula by double-clicking on them in the *Members of All* list.

7. Click **OK** to close the *Formula Editor* dialog.
8. Click **OK** to close the *Formulas* dialog.

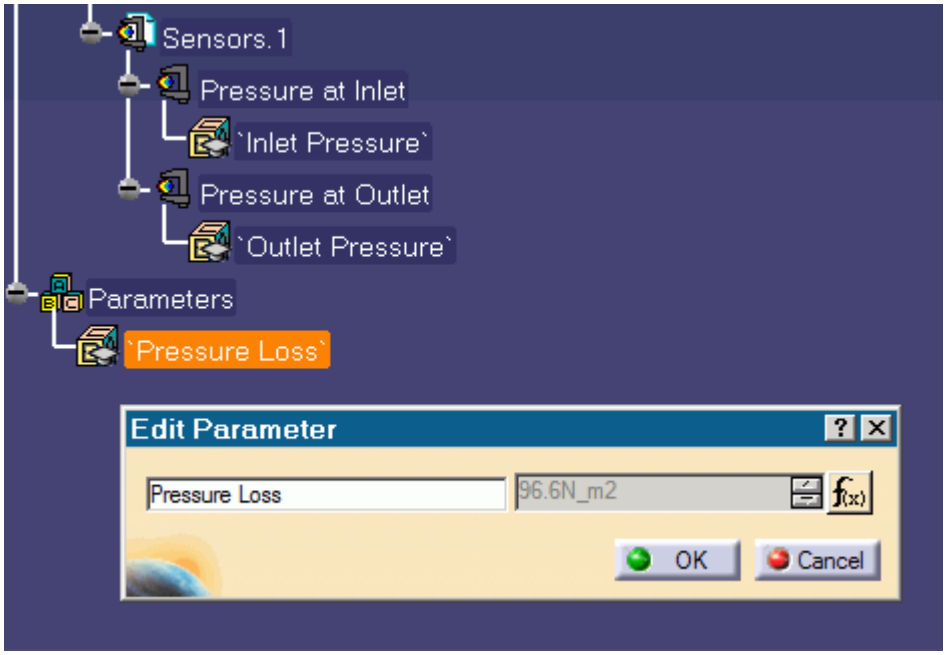
Displaying Parameters in the Analysis Feature Tree

You can display parameters that you created in the analysis feature tree:

1. Select **Tools** > **Options.**

   The *Options* dialog appears.
2. Navigate to the **Analysis & Simulation** > **General** tab.
3. Activate the *Show Parameters* option.
4. Click **OK** to close the *Options* dialog.

   A new node, **Parameters**, is added to the feature tree.
5. Double-click the parameter to view it.


Sensors.1
Pressure at Inlet
`Inlet Pressure`
Pressure at Outlet
`Outlet Pressure`
Parameters
`Pressure Loss`
Edit Parameter
Pressure Loss
96.6N_m2
OK
Cancel

6. Click **OK**.

Exporting Files and Data

This section details the process for exporting files and data from STAR-CAT5.

Contents:

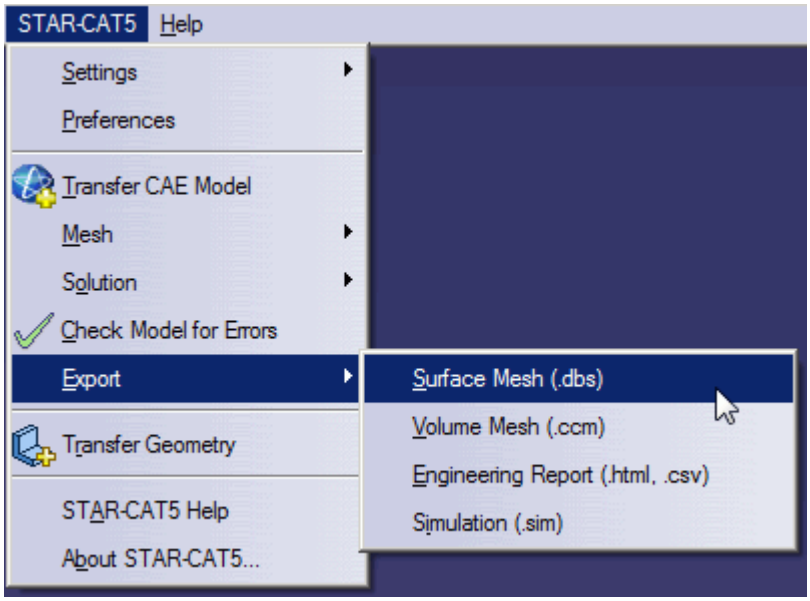
Exporting the Surface Mesh
Exporting the Volume Mesh
Exporting an Engineering Report
Exporting a STAR-CCM+ Simulation

Exporting the Surface Mesh

You can export the surface mesh generated in STAR-CAT5 as a native CD-adapco surface mesh (`.dbs`) file. The `.dbs` file format is the recommended format to use when transferring surface data between CD-adapco programs. The boundary names and boundary types are retained by the file, as well as the interfaces that were automatically created between regions.

To export the surface mesh:

1. From the STAR-CAT5 menu, select **Export** > **Surface Mesh (.dbs)**.



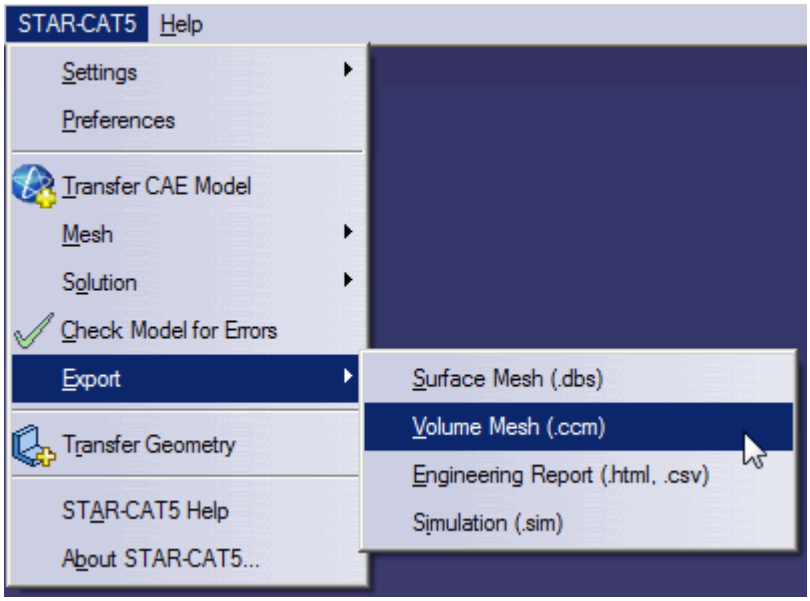
Exporting the Volume Mesh

You can export the volume mesh that you generated in STAR-CAT5 as a native CD-adapco volume mesh (`.ccm`) file. The `.ccm` file format is the recommended format to use when transferring volume meshes between CD-adapco programs. The region and boundary names, and region and boundary types are retained

in this file. The interfaces that were automatically created interfaces between regions are also retained in this file.

To export the volume mesh:

1. From the STAR-CAT5 menu, select **Export** > **Volume Mesh (.ccm)**.



Exporting an Engineering Report

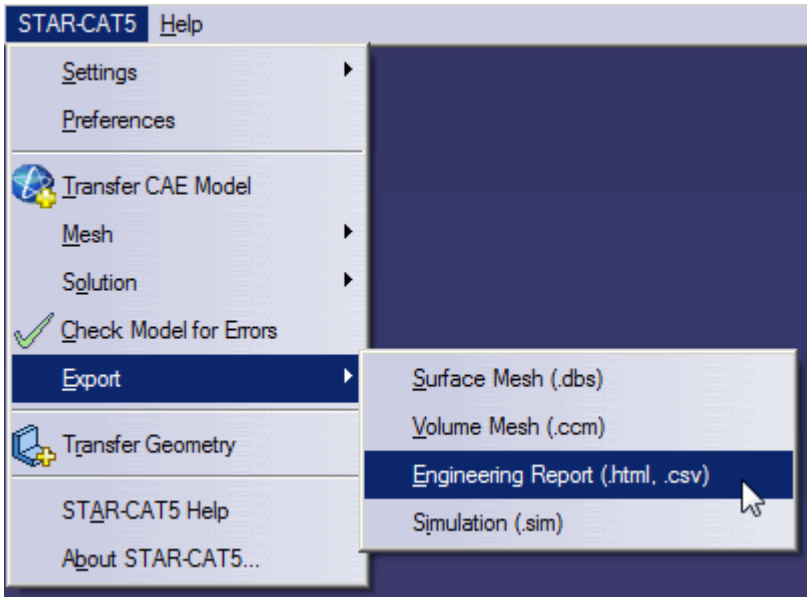
You can export a summary of the simulation settings and results in HTML format to two individual files:

- The first will be given a user-specified file name, and contains results from the simulation.
- The second will be given a user-specified file name appended with `_caseinfo`, and contains details on the simulation set-up.

Note: You can only export an engineering report after you obtain a solution.

To export an engineering report:

1. From the STAR-CAT5 menu, select **Export** > **Engineering Report (.html, .csv)**.



Results Report

The following is a summary of the information included in the results report (`<filename>.htm`). This data is provided in the report with reference to each material region in the simulation:

- Case name and time-stamp
- Solution information: CPU time, Elapsed Time and Number of Iterations
- Mass flow data, at each boundary
- Mass flux data, at each boundary
- Area averaged boundary data, for the following scalars:
  - Pressure (Pa)
  - Total Pressure (Pa)
  - Pressure Coefficient
  - Total Pressure Coefficient
  - Turbulent Kinetic Energy (J/kg)
  - Turbulent Dissipation Rate (m2/s3)
  - Turbulent Viscosity (Pa-s)
  - Density (kg/m3)
- Mass flow averaged boundary data, for the following scalars:
  - Pressure (Pa)
  - Total Pressure (Pa)
  - Pressure Coefficient
  - Total Pressure Coefficient
  - Turbulent Kinetic Energy (J/kg)
  - Turbulent Dissipation Rate (m2/s3)
  - Turbulent Viscosity (Pa-s)
  - Density (kg/m3)

- Force data, at each boundary:
  - Net Force [Fx, Fy, Fz]
  - Pressure Force [Fx, Fy, Fz]
  - Shear Force [Fx, Fy, Fz]
- Force coefficient data, at each boundary:
  - Coefficient of Net Force [Cfx, Cfy, Cfz]
  - Coefficient of Pressure Force [Cfx, Cfy, Cfz]
  - Coefficient of Shear Force [Cfx, Cfy, Cfz]

Setup Report

The following is a summary of the information found in the simulation setup report (`<filename>_caseinfo.htm`):

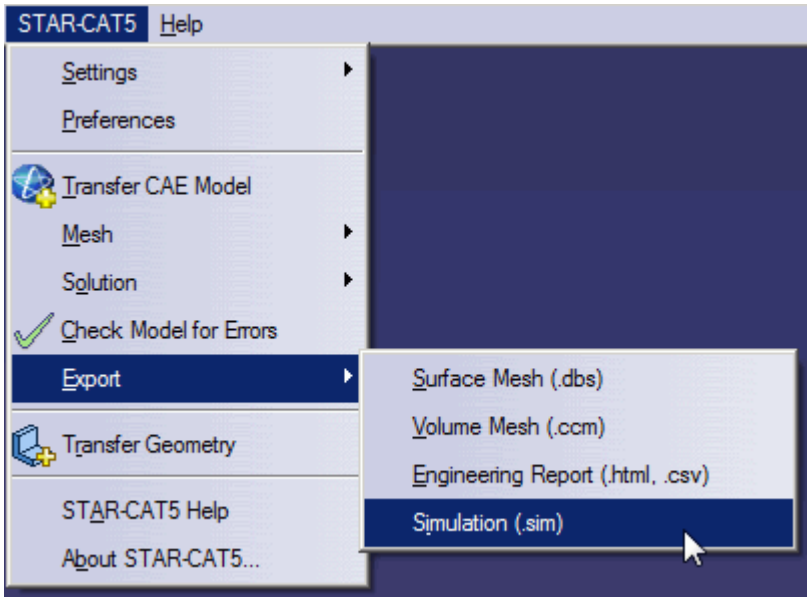
- The parts used in the simulation, and their associated regions
- The surfaces for each part, and their associated boundaries
- The continua used in the simulation, with the following information:
  - Physics models applied to the continuum
  - Materials properties
  - Turbulence properties
  - Segregated flow properties
  - Reference values
  - Initial conditions
- The mesh pipelines used in the simulation, with the following information:
  - Polyhedral mesher properties
  - Prism layer mesher properties (if activated)
  - Surface remesher properties
  - Reference values
  - Volumetric controls
- The regions used in the simulation, and their associated continua and mesh pipeline
- The boundaries for each continua, with the applied conditions and values
- Solver parameters
- Stopping criteria
- Units

Exporting a STAR-CCM+ Simulation

You can export the current simulation in STAR-CAT5 to the native STAR-CCM+ simulation (`.sim`) file format. The `.sim` file format retains all simulation settings defined in STAR-CAT5, as well as the surface and volume meshes.

To export a simulation file:

1. From the STAR-CAT5 menu, select **Export** > **Simulation (.sim)**.


STAR-CAT5
Help
Settings
Preferences
Transfer CAE Model
Mesh
Solution
Check Model for Errors
Export
Transfer Geometry
STAR-CAT5 Help
About STAR-CAT5...
Surface Mesh (.dbs)
Volume Mesh (.ccm)
Engineering Report (.html, .csv)
Simulation (.sim)

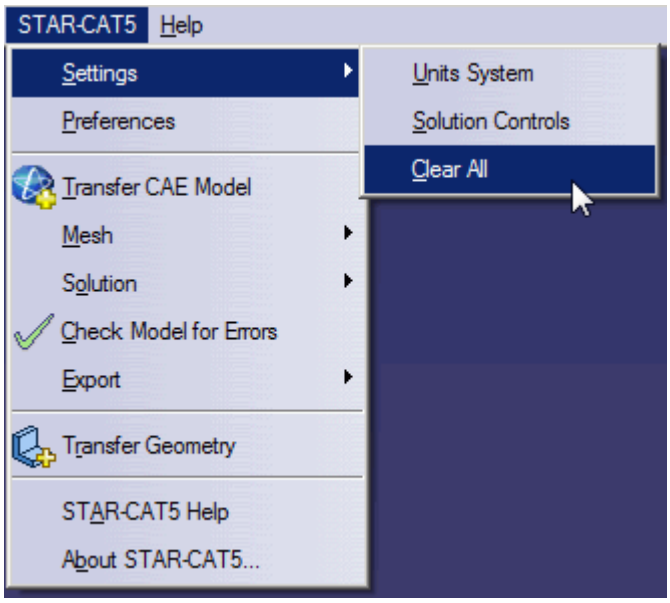
Resetting STAR-CAT5 Settings

STAR-CAT5 saves all case setup information to the CAD file. This means that all continuum settings, boundary settings, solution control settings, and mesh settings, are stored within the CAD geometry file. You can remove this information from the CAD geometry file.

Caution: This action is irreversible.

To clear all STAR-CAT5 settings:

1. From the STAR-CAT5 menu, select **Settings** > **Clear All**.



Guidelines

This section aims to provide guidance and assistance on making a geometry part suitable for CFD analysis using STAR-CAT5.

Contents:

Creating the Fluid Domain
Imprinting Parts

Creating the Fluid Domain

In the majority of CAD models, the fluid domain for analysis must be extracted from the original part or assembly. The topics that are containing within this section demonstrate how to prepare the fluid domain.

Contents:

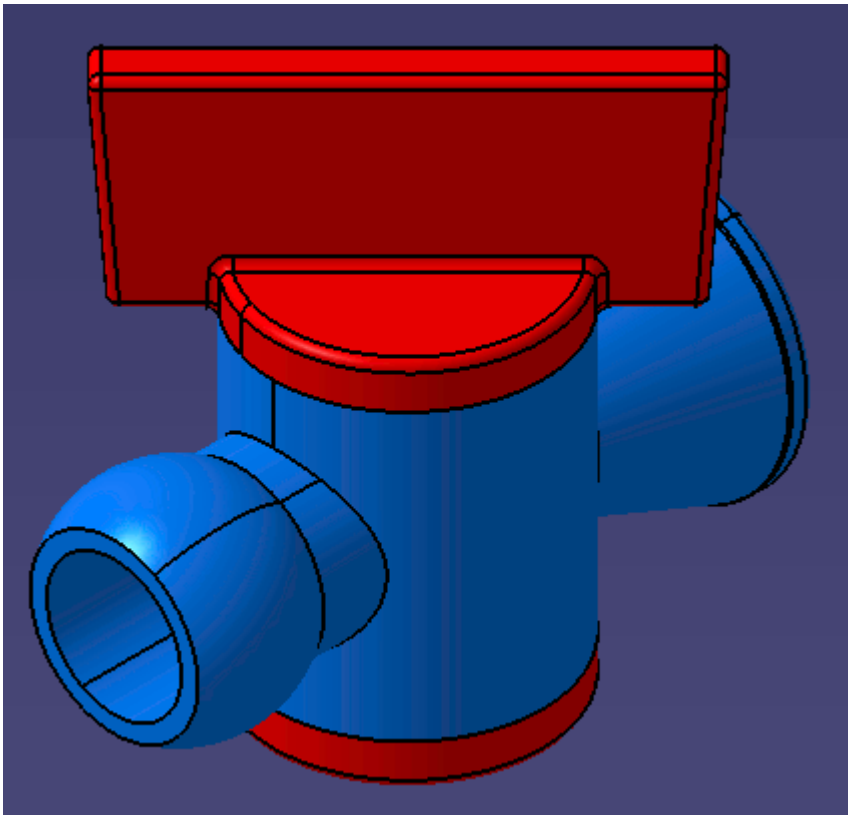
Internal Flow
External Flow
Multi Region Flow

Internal Flow

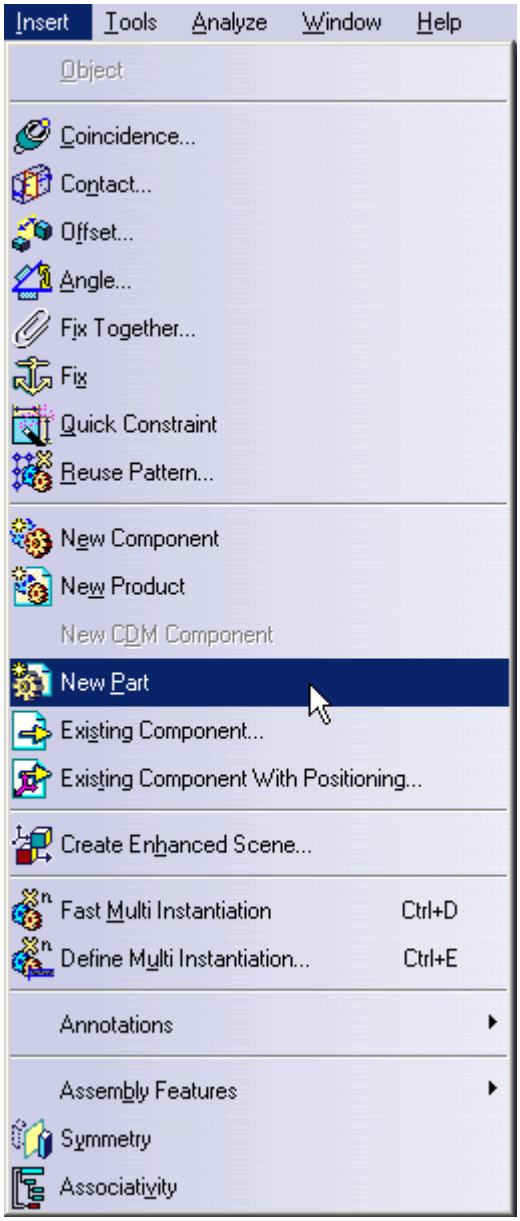
In general, the steps to follow in creating the fluid domain for an internal flow are:

- Work in a product. If the original geometry contains only a single part, create a new product that contains the part.
- Create additional parts that close off any openings in the original geometry, we will call these parts the “caps”.
- Create an overall block part that encloses all the other parts, the “enclosure”. This will form the internal fluid volume.
- Subtract all parts from the enclosure. This will result in a solid block that contains a cavity.
- Delete the outer solid body--the remaining geometry is the fluid domain.

A worked example is provided below. In this case the assembly contains two parts: the valve (shown in red), and the valve housing (shown in blue).



1. Load the part or product into CATIA. If the geometry is a part, add it to a product.
2. Insert a new part into the assembly by selecting **Insert** > **New Part**.



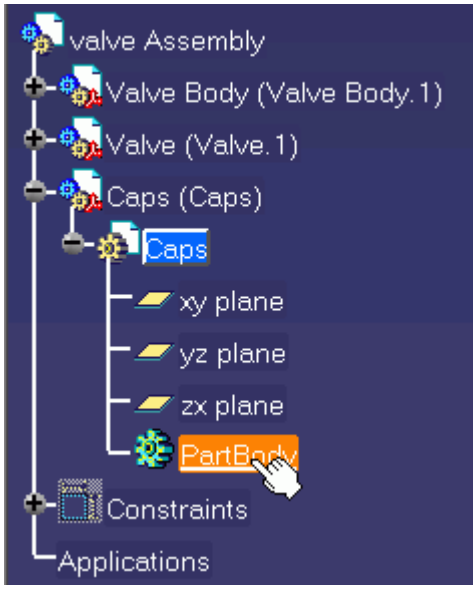
3. Click on the root node in the assembly's specification tree to insert the part into that assembly.

   A dialog will appear asking whether a new origin for this part should be defined or not. Choose either option.

4. Give the new part an appropriate name, in this example the part is called `Caps`.

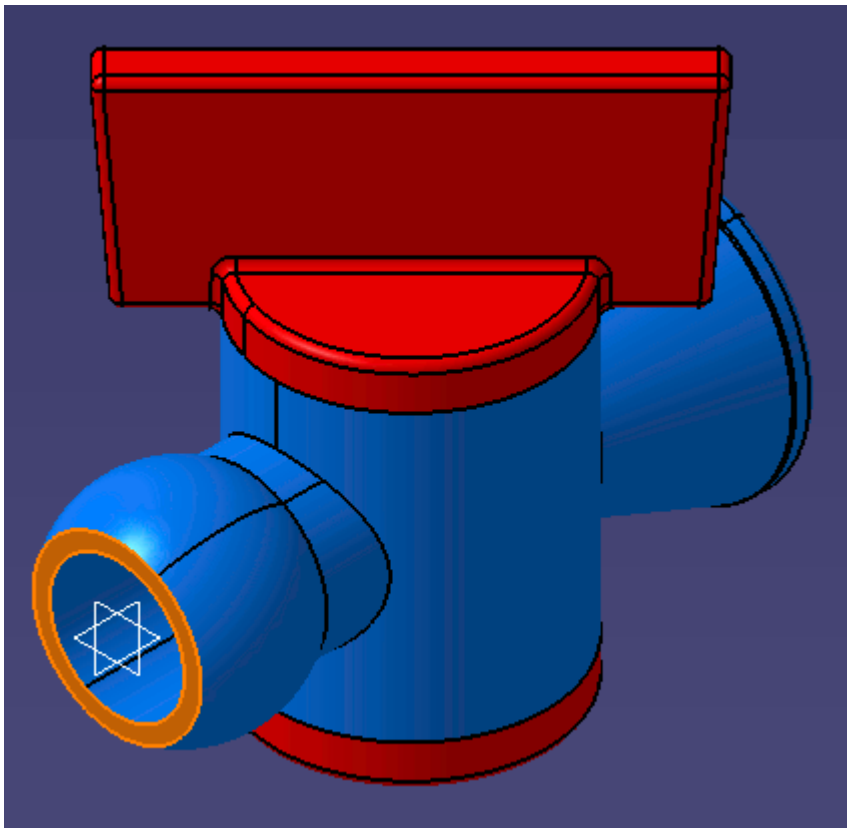
We will now close all the openings on the geometry--in this example two caps are needed as there are two openings.

5. Make the Caps part the active part by double-clicking on its **Part Body** node in the Specification Tree.

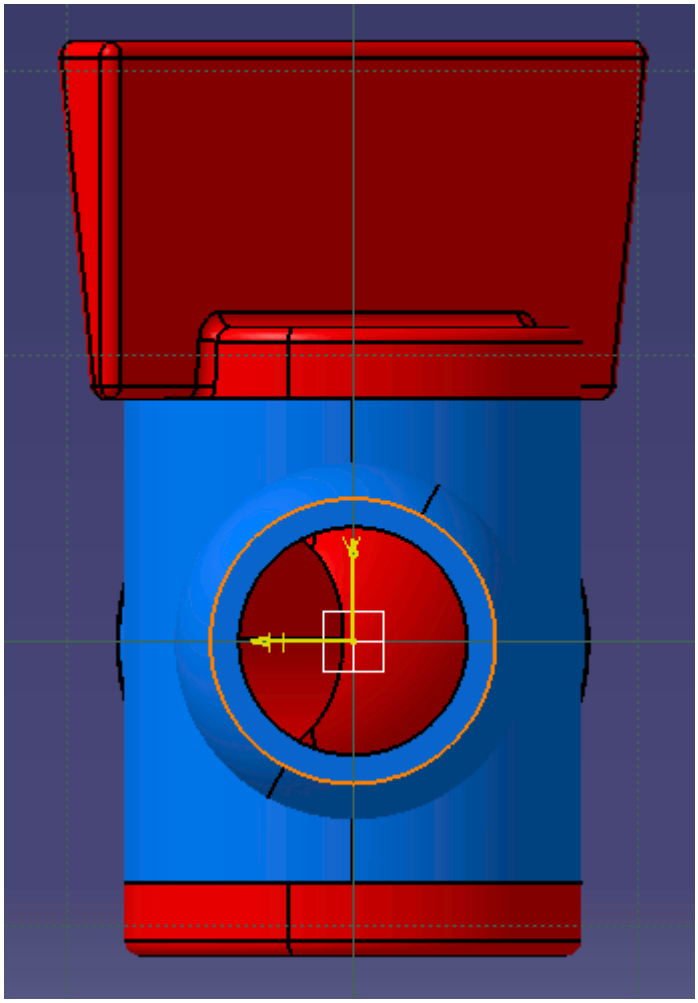


The part will become active and the **Part Design** workbench appears. When parts are active these will be highlighted with a blue background in the specification tree.

6. Create a new sketch on a surface that contains an opening, as shown below.



7. In the **Sketcher** workbench, select the outermost edge of the face

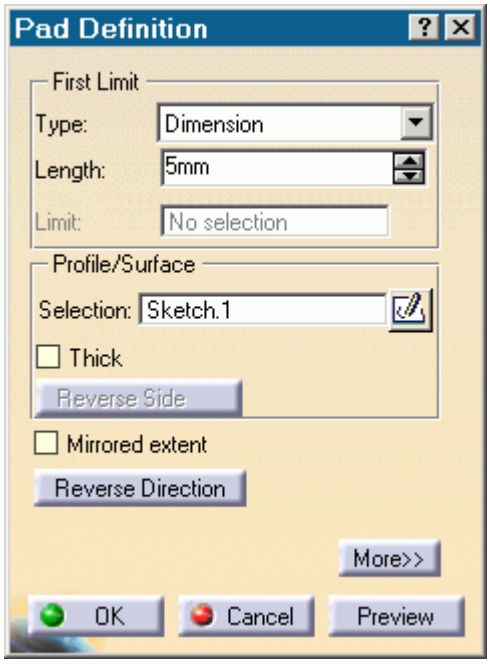


8. Use the **Project 3D Elements** tool.

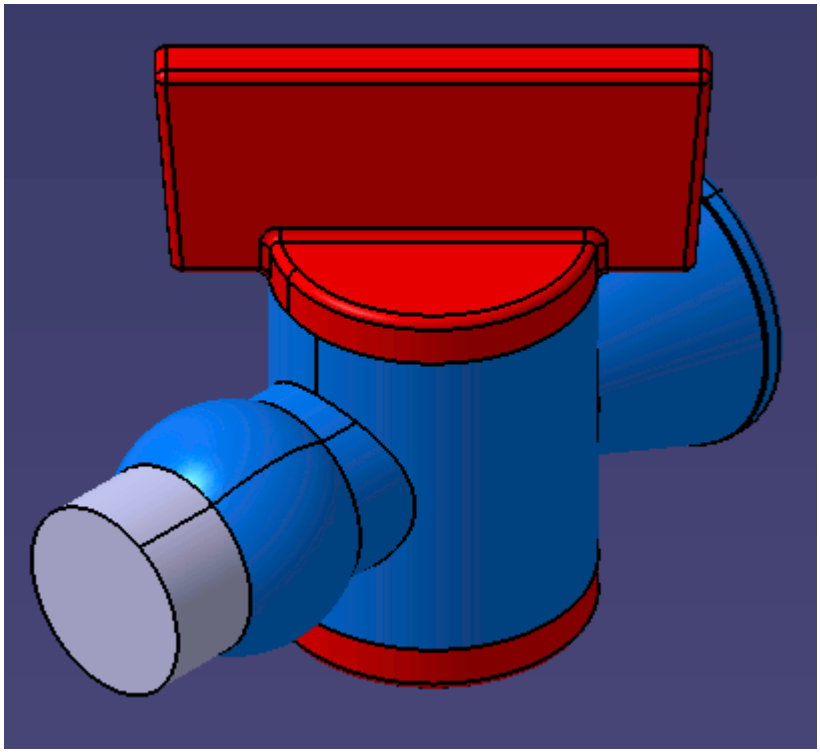
This will project the edge of the existing body onto the current sketch. The yellow color of the sketch means that it is associative, so if the diameter of the part changes, for example, the cap will automatically update to match the change.

9. Exit the Sketcher workbench and create a **Pad** from the sketch.

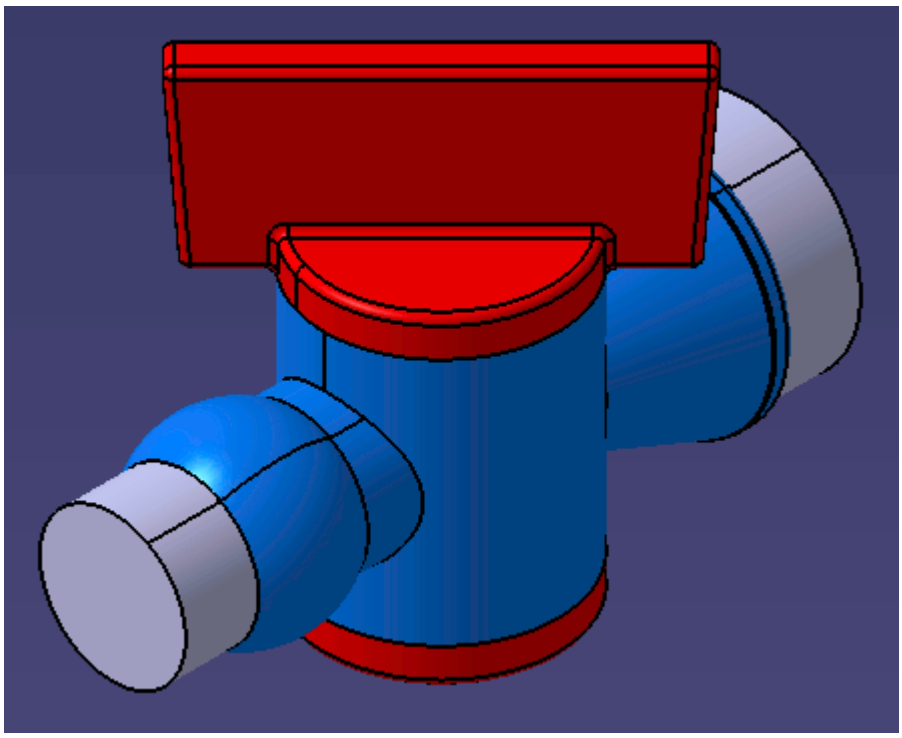
The length of the pad is not important, however ensure that the pad extends away from the geometry. If the pad is going inwards, use the **Reverse Direction** button in the *Pad Definition* dialog. In this example a *Length* of 5 mm is sufficient.



The result of this operation is shown below.



10. Repeat the same steps for the other opening(s).

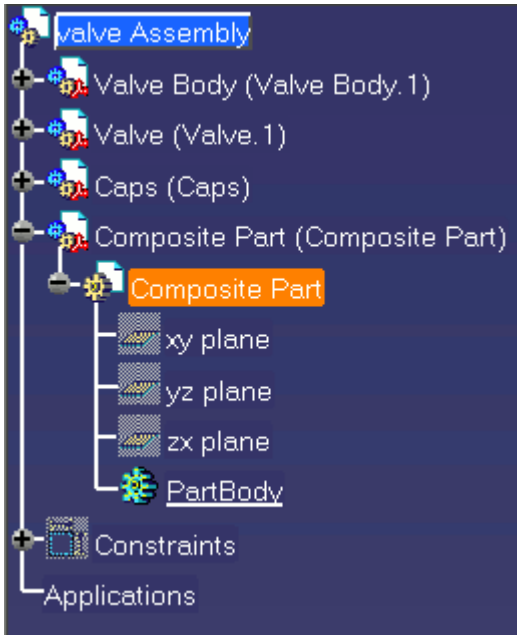


Note that all the caps should be in the same part.

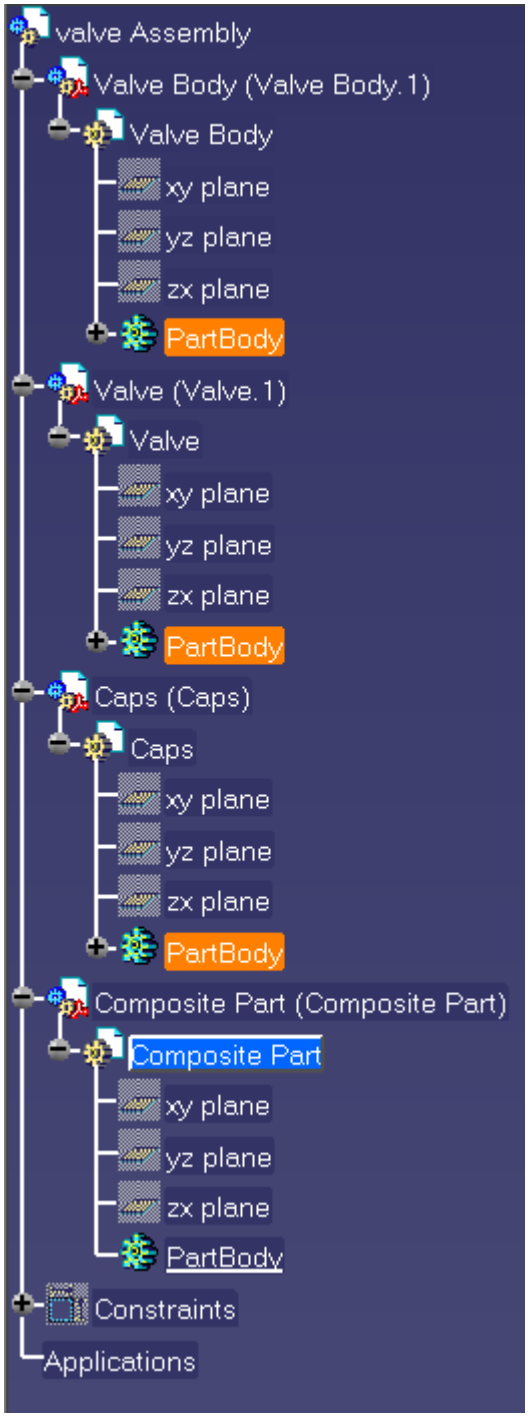
11. Double-click the root node in the specification tree to return to the **Assembly Design** workbench.

We will now insert a new part into the product. This part will become a composite of all the current parts, in this example the valve, the valve body, and the caps.

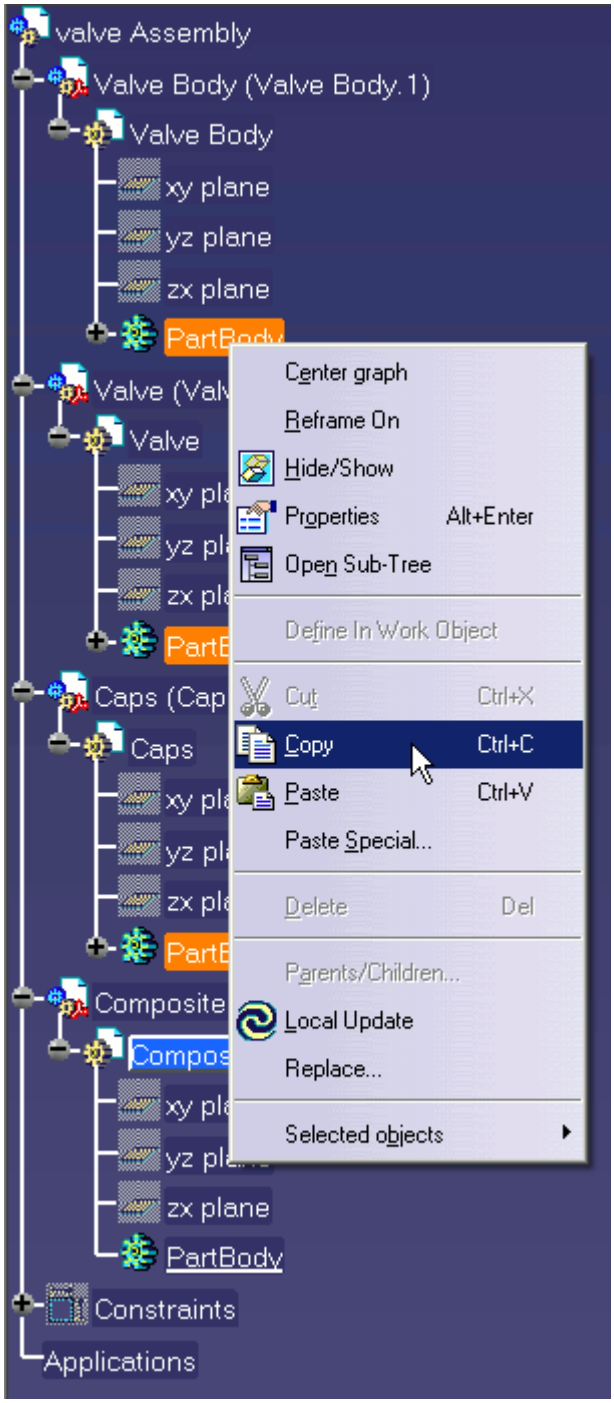
12. Insert a new part into the product as described previously. Give this new part a suitable name. In this example, the part was named `Composite Part`.
13. Double-click on the part file to activate it, as shown below.



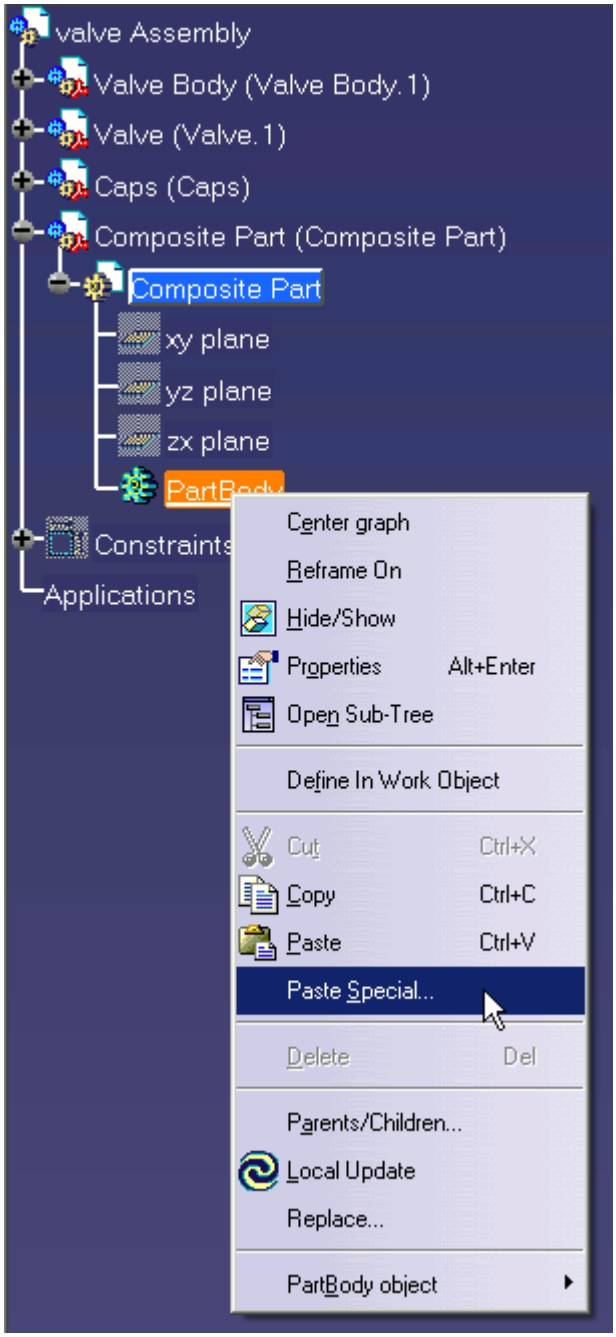
14. Expand the nodes for all the parts to reveal the **PartBody** nodes and select them as shown below. Hold down the <`ctrl`> key to select multiple items.



15. Right-click on one of the highlighted **PartBody** nodes and select **Copy**.

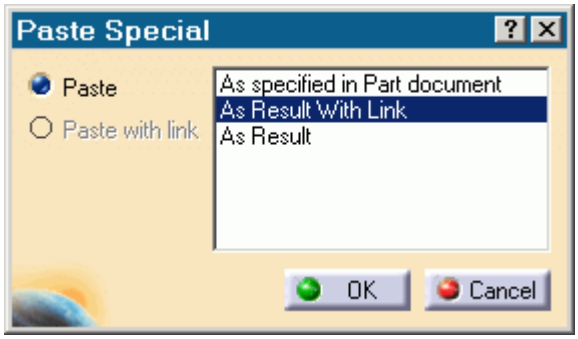


16. Paste these into the active **Composite Part** by right-clicking on it's **PartBody** node as shown below, and selecting **Paste Special...**



The *Paste Special* dialog appears.

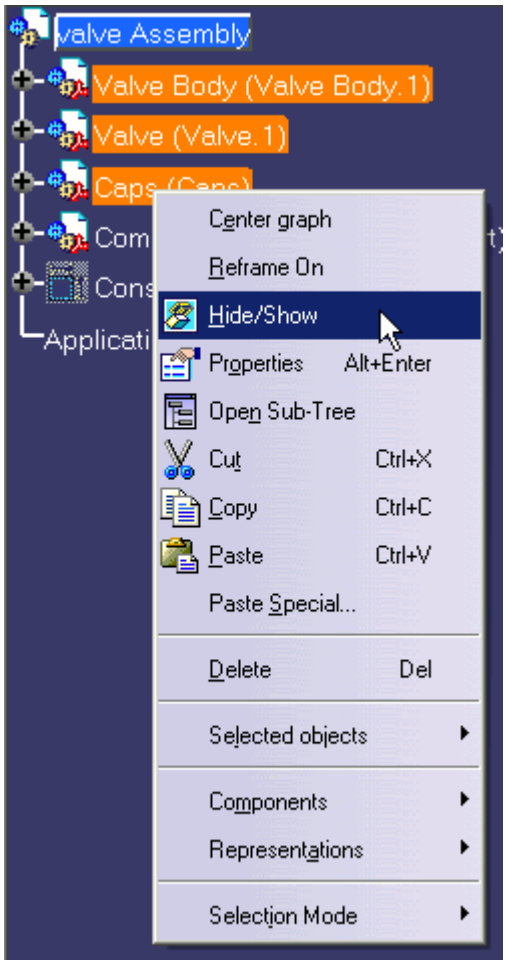
17. In the *Paste Special* dialog, select **As Result With Link** and click **OK**.



This will paste all the selected parts into the active part without breaking the link to the original parts, meaning that if the original parts change, the pasted parts will also change--this is useful when analyzing different configurations in the product such as valve angle in this example.

The original parts are no longer required and will be hidden.

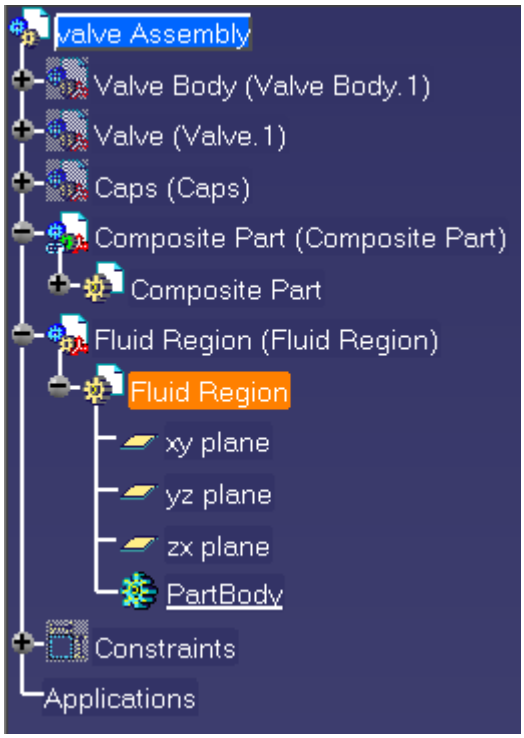
18. Return to the Assembly Design workbench by double-clicking on the root node in the specification tree.
19. Select the valve, valve body, and caps parts in the specification tree. Right-click on one of the nodes and select **Hide/Show**.



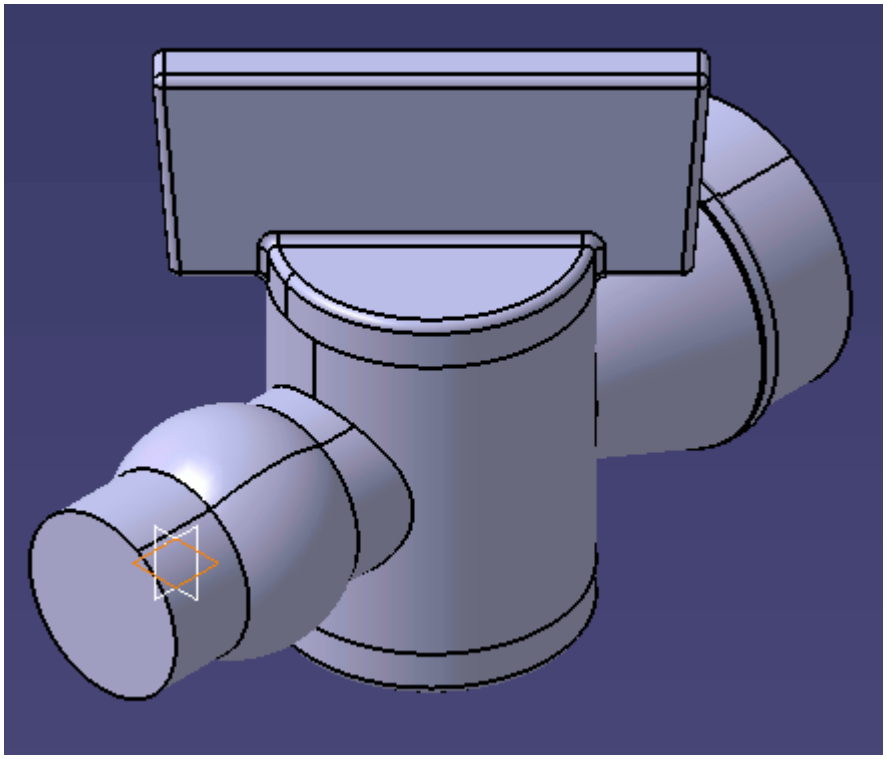
The parts' nodes will become greyed out indicating they are hidden.

We will now insert a new part into the product. This will ultimately become the internal fluid volume. To begin with, we will create an enclosure that completely encloses the existing parts. The caps and valve assembly will then be subtracted from the enclosure which will result in two solid bodies, the outer enclosure and the internal fluid volume. The outer enclosure will then be deleted, leaving only the desired internal fluid volume.

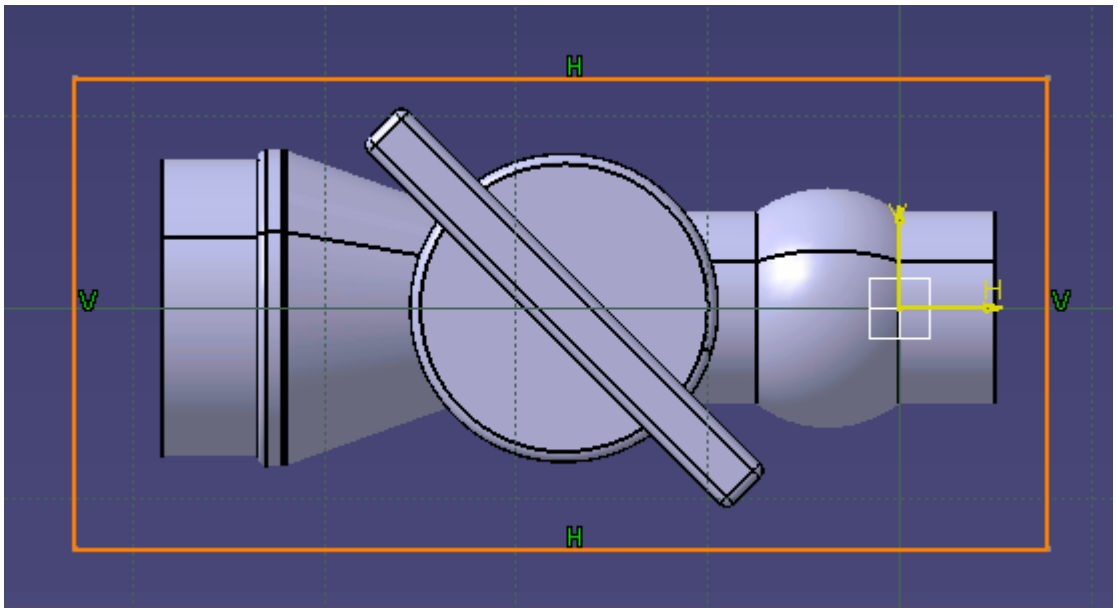
20. Insert a new part into the assembly as described previously and rename it to `Fluid Region`.
21. Activate the **Fluid Region** part by double-clicking on it.



22. Create a sketch on the top view plane.

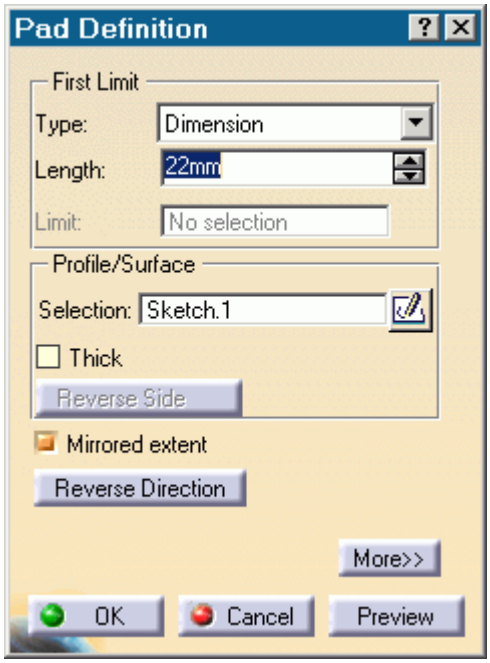


23. Draw a rough sketch that encloses the entire assembly as shown below.

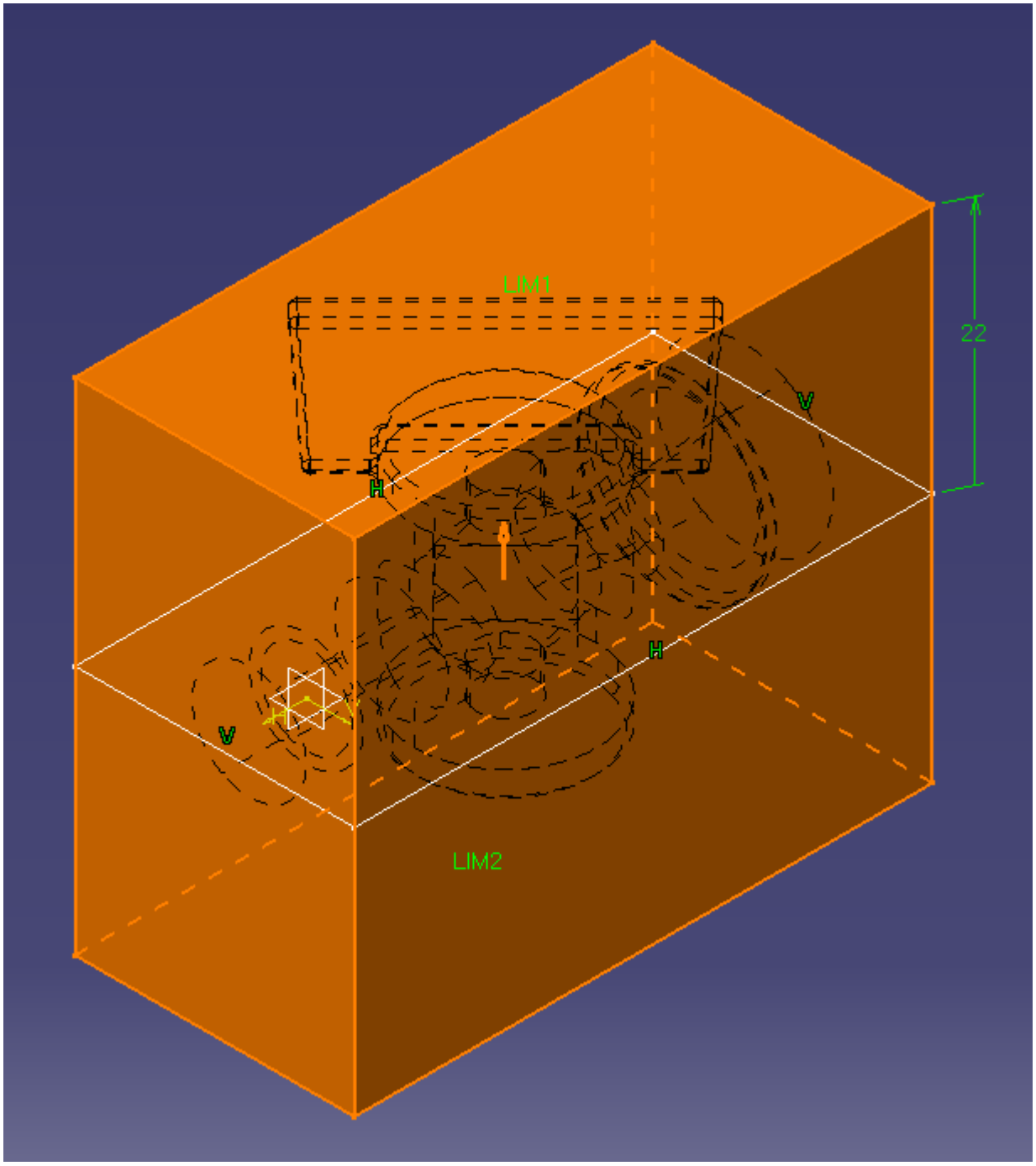


24. Exit the Sketcher workbench.
25. Create a pad from the sketch, ensuring that it encloses the geometry completely.

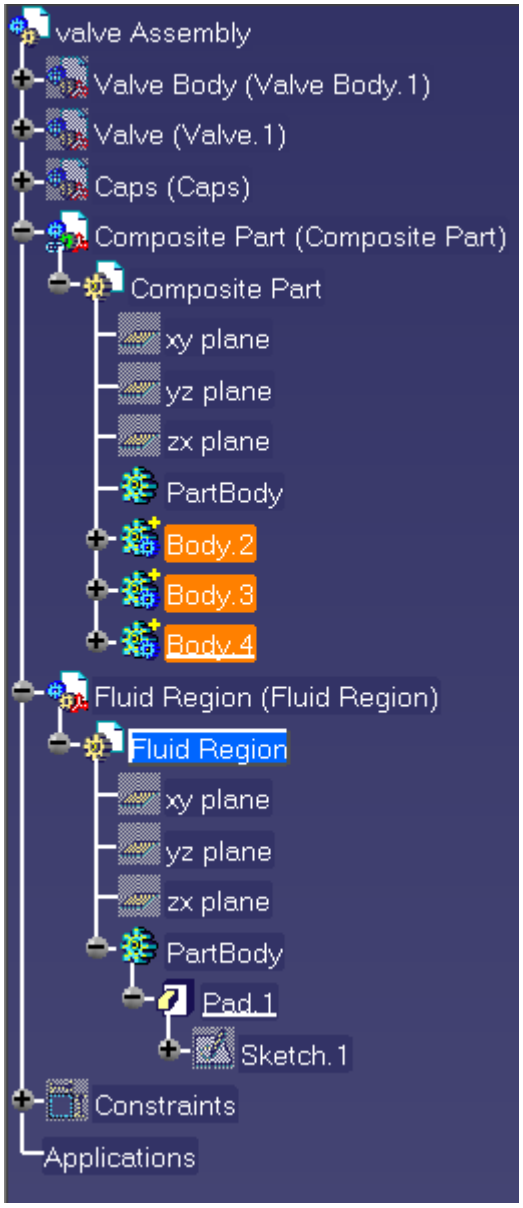
    The pad can be extruded in both directions by ticking the *Mirrored extent* option in the *Pad Definition* dialog.



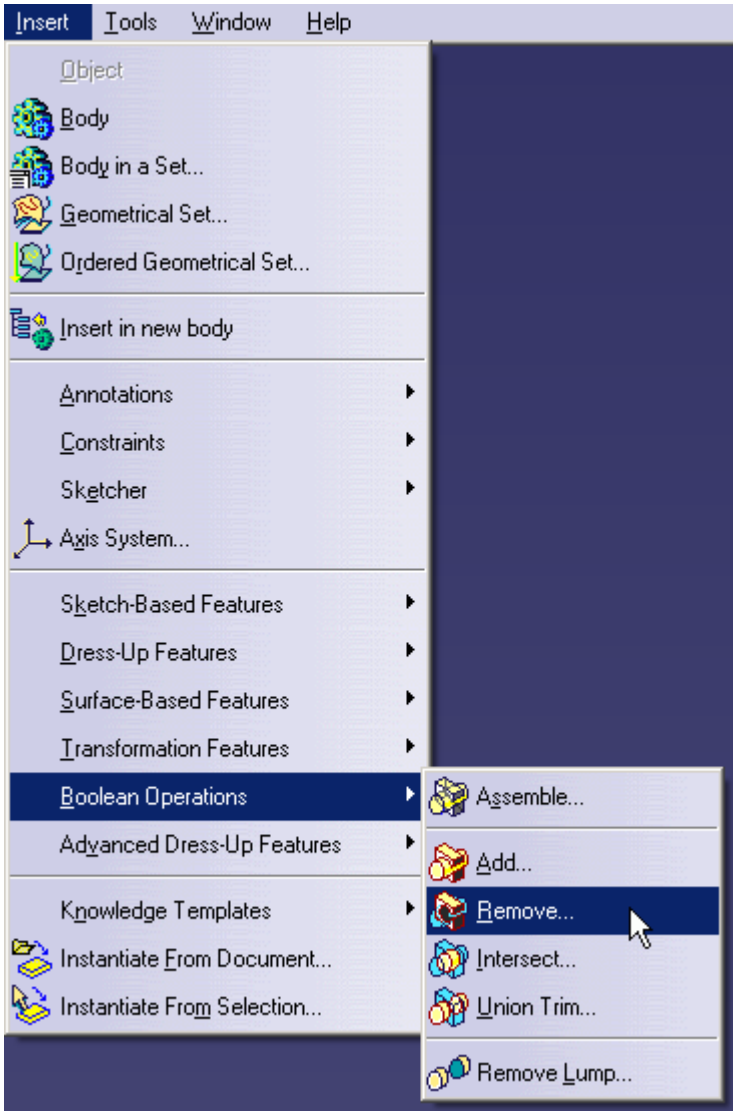
The view can be changed to **Shading with Edges and Hidden Edges** to show the hidden part inside the new pad.



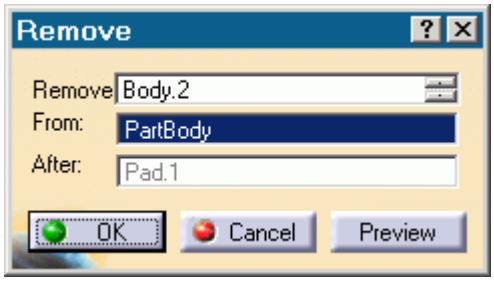
26. Select the copied **Body** parts (in this case **Body.2**, **Body.3**, and **Body.4**) from the **Composite Part** specification tree.



27. Select **Insert** > **Boolean Operations** > **Remove...**



This will remove the **Composite Part** from the **Fluid Region** using a boolean operation. The result will be a cavity. In the *Remove* dialog, ensure that the three parts from the **Composite Part** are listed in the *Remove* field and that the *From* field contains the **Fluid Region**'s **PartBody** as shown below.



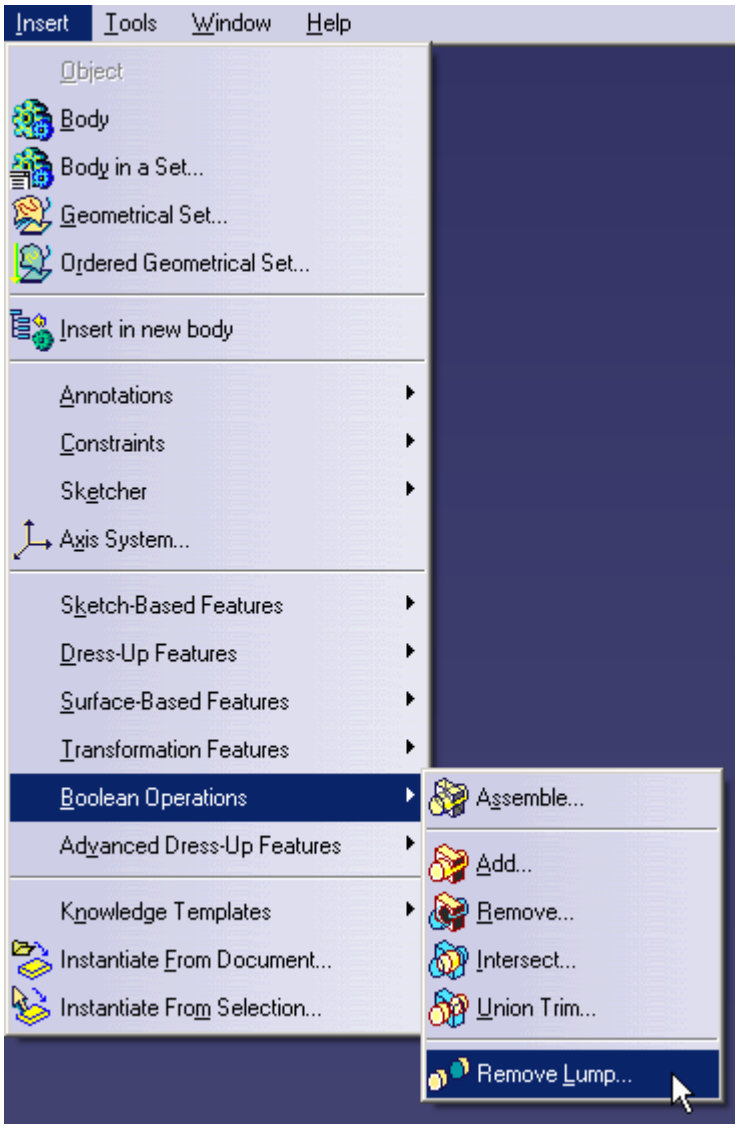
28. Click **OK** to close the dialog.

    The **Fluid Region** part now contains a cavity separating the internal fluid volume from the bulk solid.

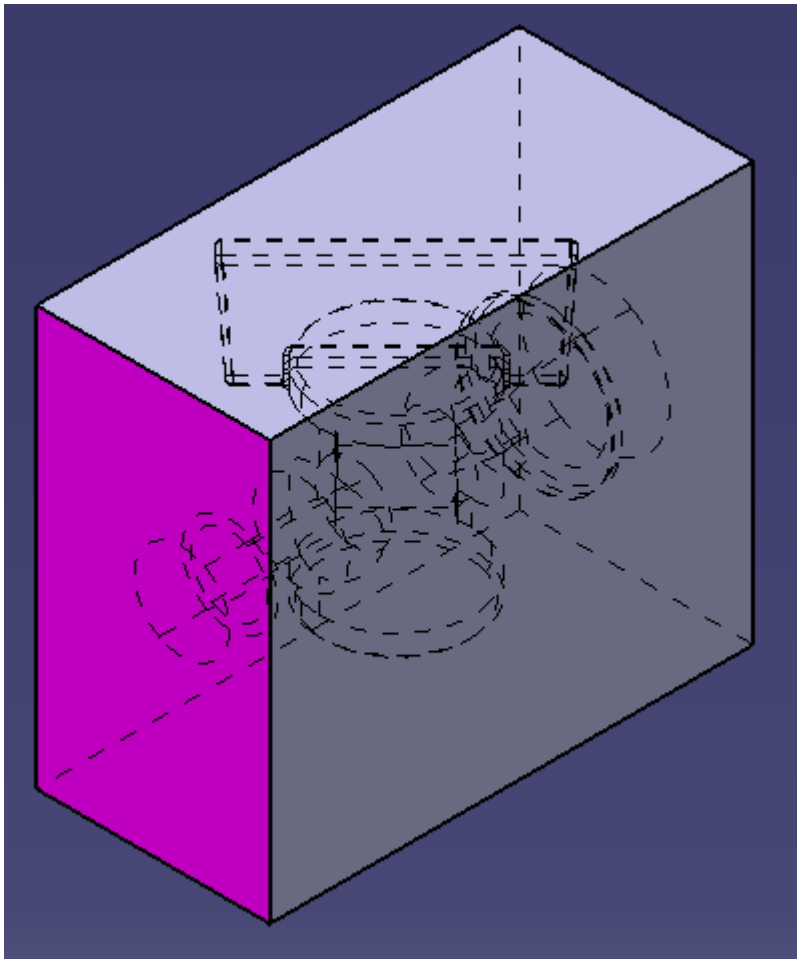
29. **Hide** the **Composite Part**.

The outer bulk solid will be removed using the **Remove Lump** operation, resulting in only the internal fluid volume.

30. Select **Insert** > **Boolean Operations** > **Remove Lump**.



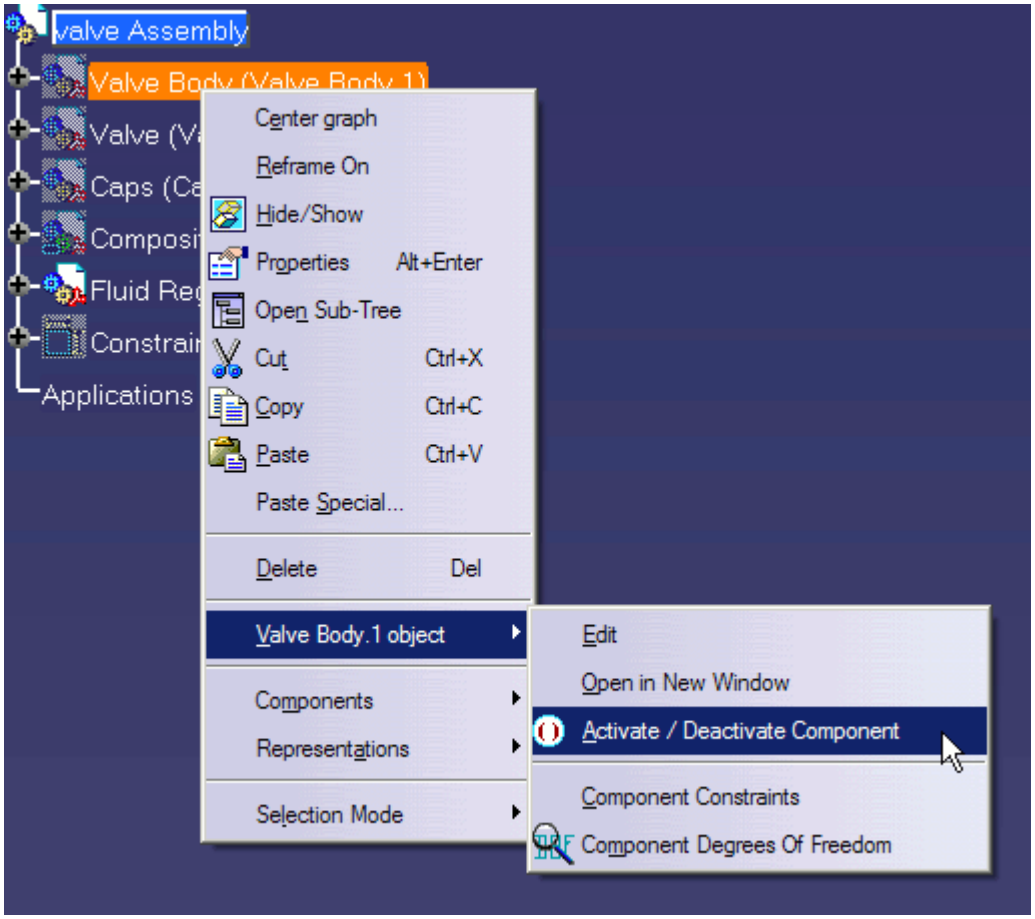
31. Click on any surface on the outer solid to bring up the *Remove Lump Definition* dialog.
32. Click in the *Faces to Remove* field and select any surface belonging to the outer body.



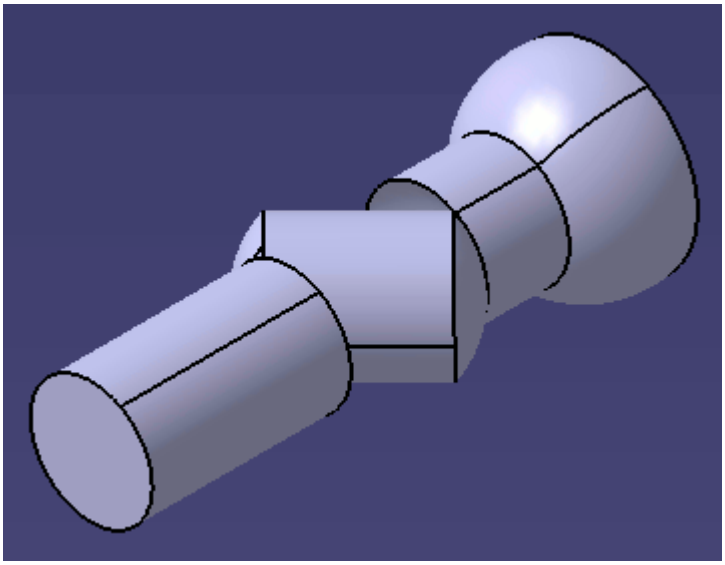
33. Click in the *Faces to Keep* field and select an internal face on the geometry.

    The face has to be a wetted face, such as the inside surface of the valve housing. To select hidden surfaces use `<Ctrl> + <F11>`.

34. Click **OK** to close the dialog.
35. Deactivate all parts that you do not want to take part in the analysis, such as the original parts and the composite part, by right-clicking on each node and selecting **[instance_name]** > **object** > **Activate/ Deactivate Component**.



The internal fluid volume will appear as shown below.



The geometry is now ready for the CAD client analysis. Deactivated parts in an assembly will not be included in the analysis. Suppressed parts in an assembly will not be included in the analysis. In the above example, it

is recommended that you extend the inlet and outlet regions to appropriate lengths so that a fully developed flow can be achieved before reaching the valve and to help reduce recirculation at the outlet.

External Flow

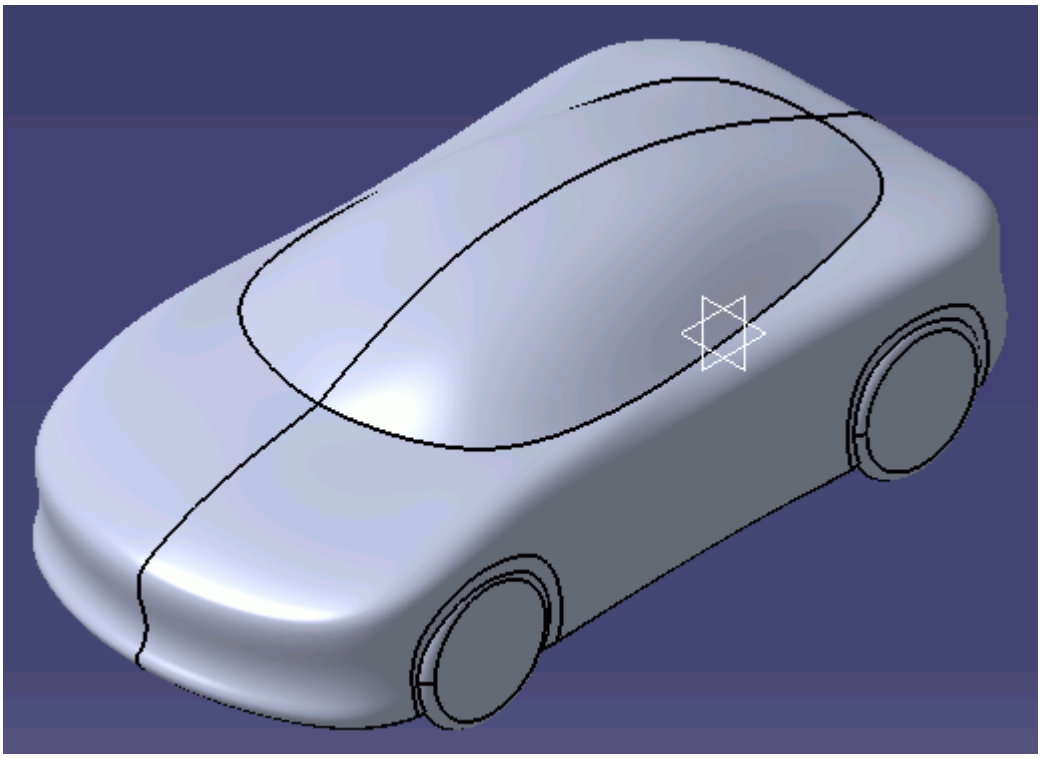
The generation of a domain for external aerodynamics is generally easier than for internal flow cases but the sizing of the domain is more critical. As a guideline (assuming a specific situation is not being modelled i.e. flow in a wind tunnel) the domain should be sized as follows:

- Inlet: 2 x part length from the front of the part
- Outlet: 4 x part length from the rear of the part
- Sides: 3 x part width from the side of the part
- Ceiling: 4 x part height from the top of the part

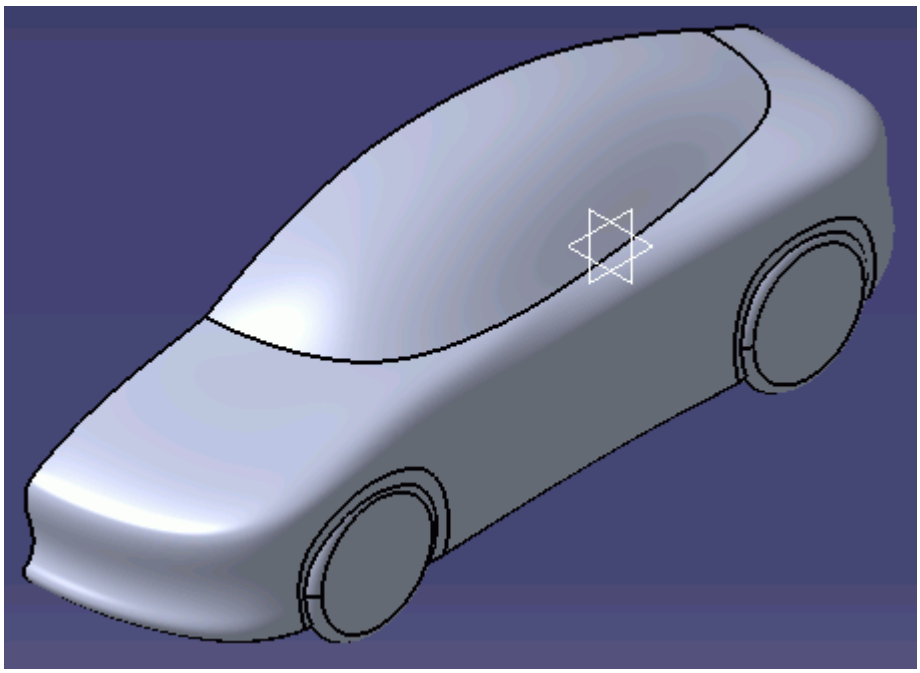
In general, the steps to follow in creating the fluid domain for an external flow are:

- Work in an assembly. If the original geometry contains only a single part, create a new assembly that contains the part.
- Create an overall block part that encloses all the other parts. The shape and size should correspond to the desired fluid domain.
- Subtract all parts from the overall block part - the remaining geometry is the fluid domain.

An example case is shown below. In this example a geometry of a car will be used.

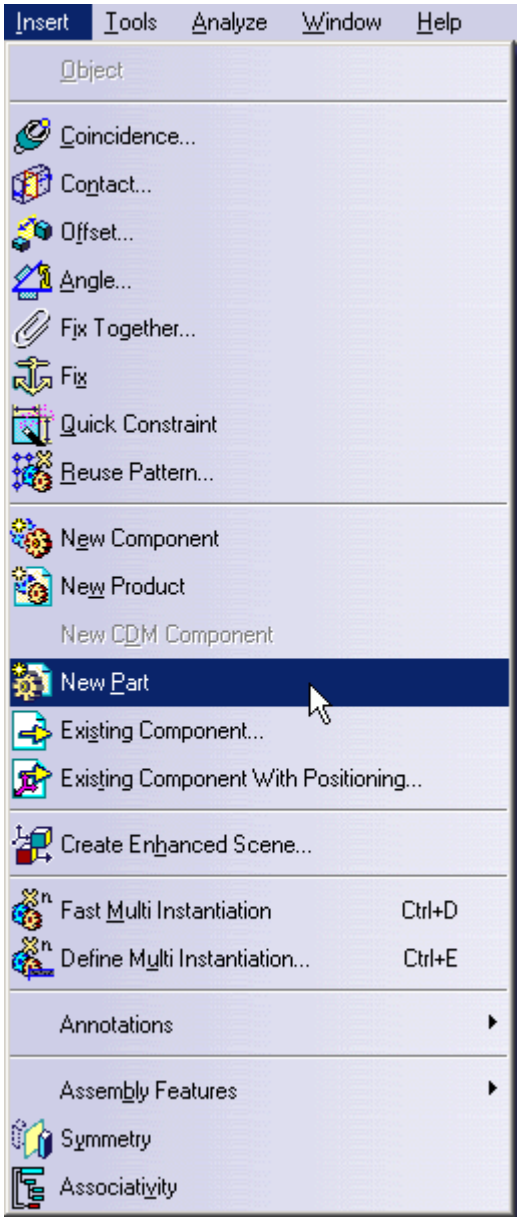


1. Load the product file in CATIA. If the geometry is a part, insert it into a product.
2. If the part is symmetrical, this should be cut in half, as shown below, to reduce computational costs during the analysis.

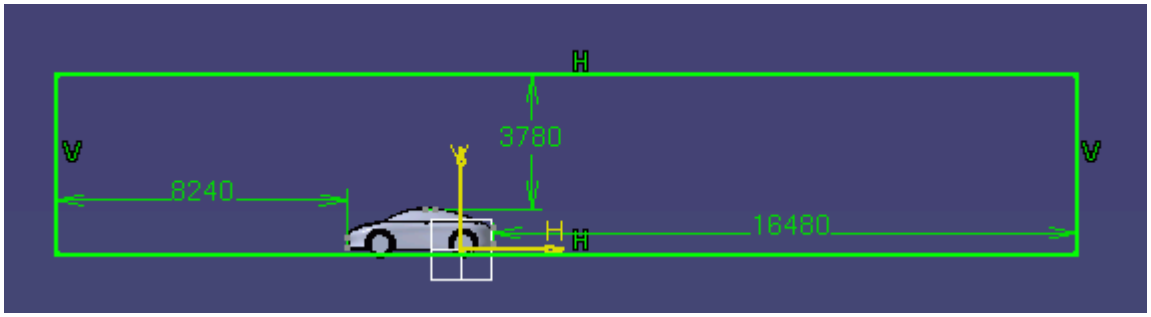


We will now insert a new part into the product. This will become the fluid domain.

3. Insert a new part into the product, **Insert** > **New Part**.

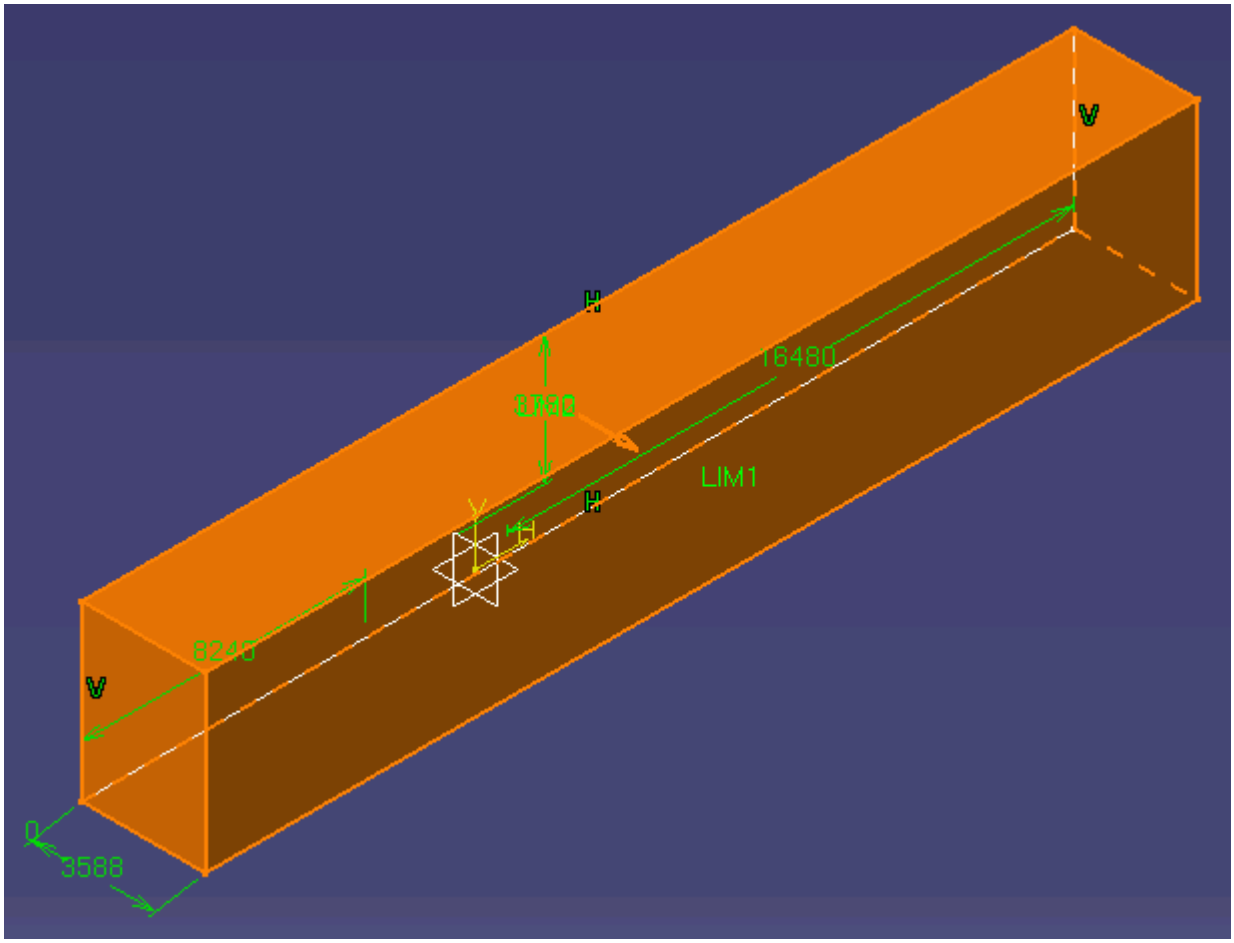


4. Click on the **Assembly** root node in the specification tree to insert the part into the product. A dialog will appear asking whether a new origin should be defined for the part or not--you may choose either option.
5. Create a new sketch on any plane. In this example the sketch was created on the symmetry plane of the car.
6. Draw a profile with suitable dimensions--this sketch represents the profile of the fluid domain.



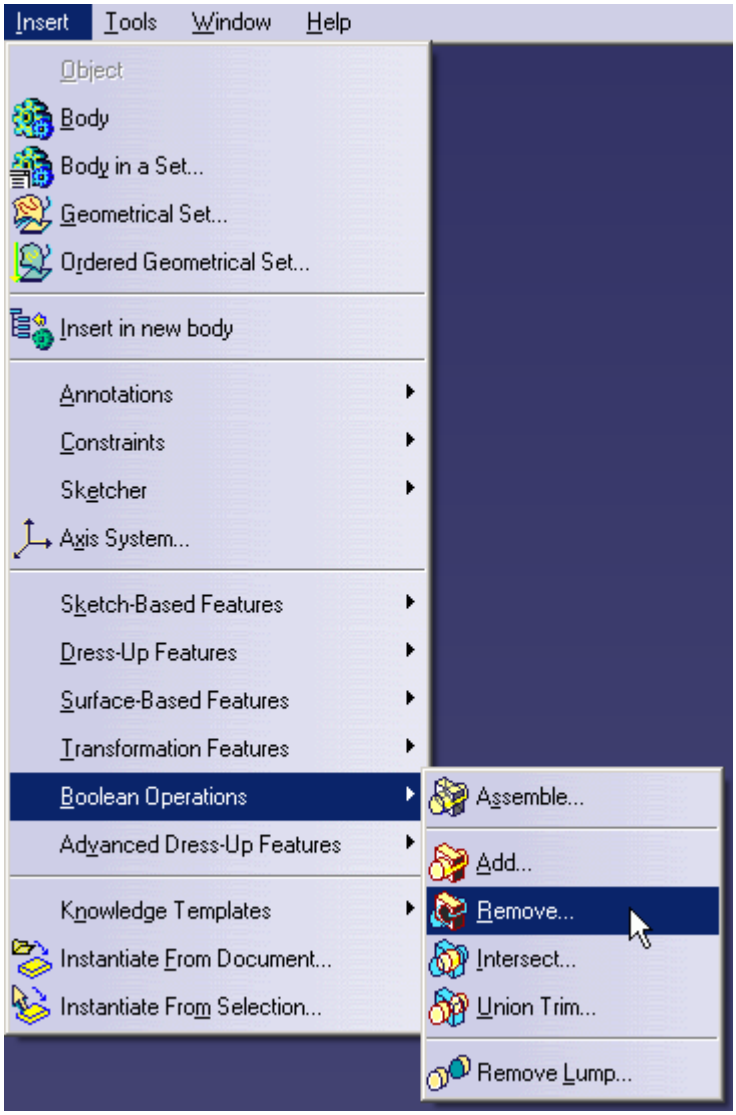
7. Exit the Sketcher workbench and create a **Pad** from the sketch.

   The length of the pad will correspond to the width of the fluid domain, as shown below.



We will now remove the car from the fluid domain using a boolean operation.

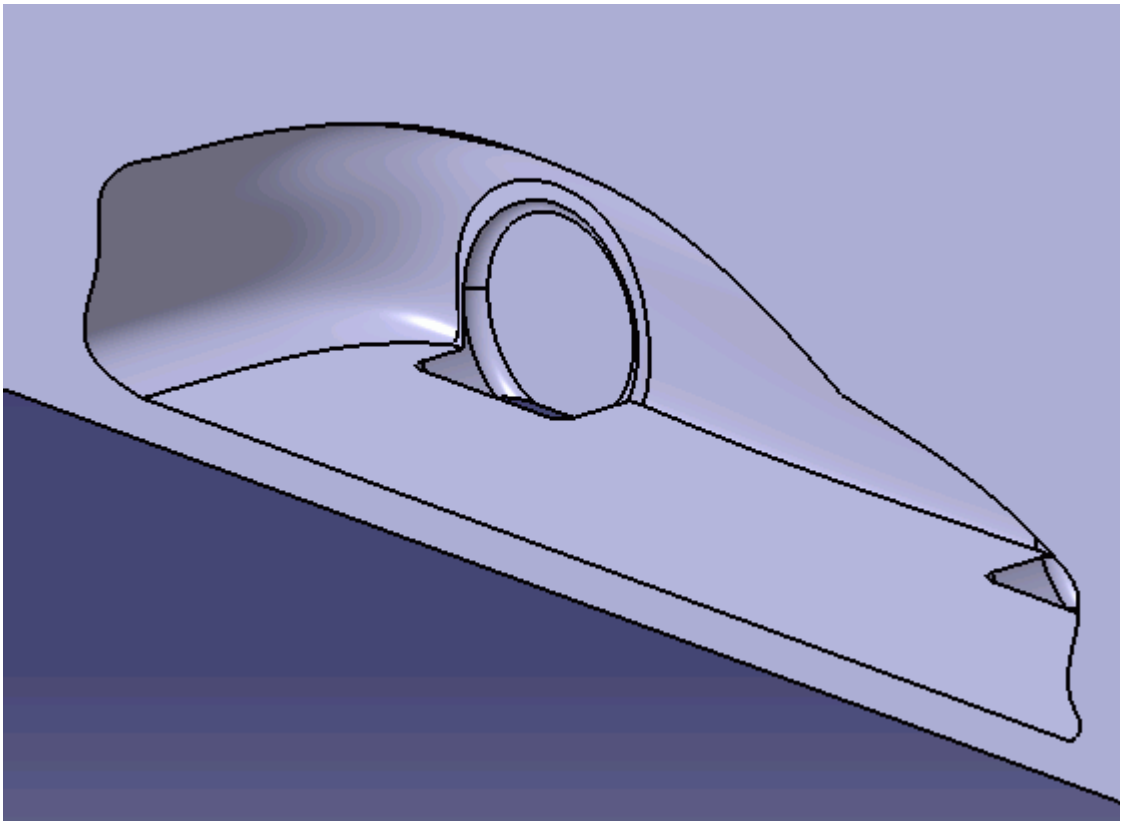
8. Select **Insert** > **Boolean Operations** > **Remove...**



9. In the *Remove* dialog that appears, select the car's part body for the *Remove* field and the fluid domain for the *From* field.

10. Return to the assembly after the operation is complete and hide the car part, right-click the car node in the specification tree and select **Show/Hide**.

The area around the car in the final domain is shown below.



The geometry is now ready for the cad client analysis. Deactivated parts are not included in the analysis. If you make any changes to the car geometry, these will automatically be passed on to the fluid region, giving you an up-to-date fluid domain.

Multi Region Flow

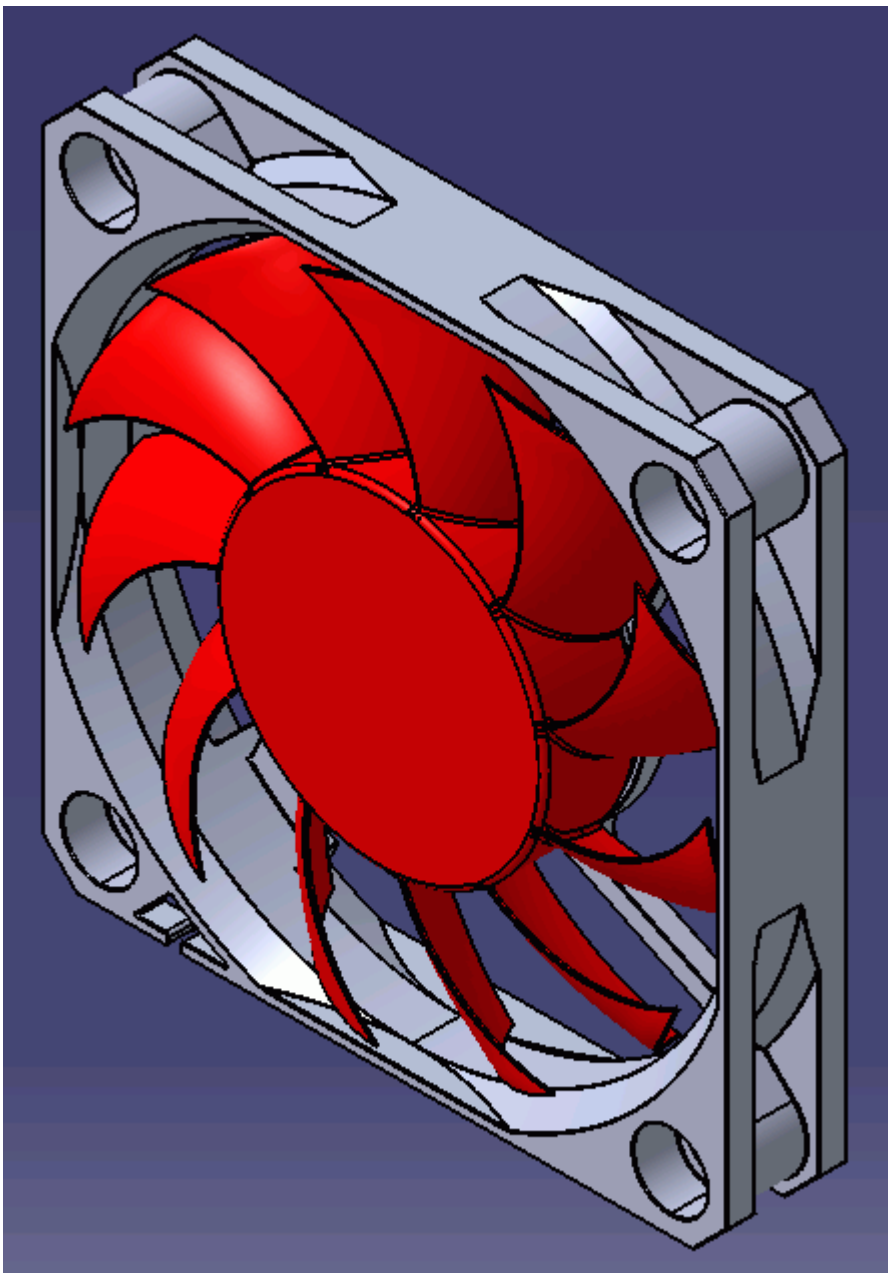
This section shows how to create the fluid domain for multiple regions such as rotating machinery and porous media flow.

For the numerical assumptions made using the Moving Reference Frame (MRF) technique to hold true, a minimum of two cells need to be present between the rotating-static interface and any wall boundaries (for example the fan blades), and the rotating region must be circular in section. Where there is a large clearance between the rotating object and any static objects, it is advisable to avoid placing the interface too close to either hence facilitating a small cell size on the interface. Where there is a small clearance between the rotating and static parts, there is little choice but to place the interface approximately midway between the two.

In general, the steps to follow in creating the fluid domain for multiple regions are:

- Work in an assembly. If the original geometry contains only a single part, create a new assembly that contains the part.
- Create an additional part that will enclose the rotating region (for example a fan) and subtract it from the part. In the case of a porous region, a separate part will be required for this as well.
- Create an additional part for the main static domain (the inlet and outlet regions) and subtract any obstructions/features that may be present there (for example the fan motor support arms on a fan casing).

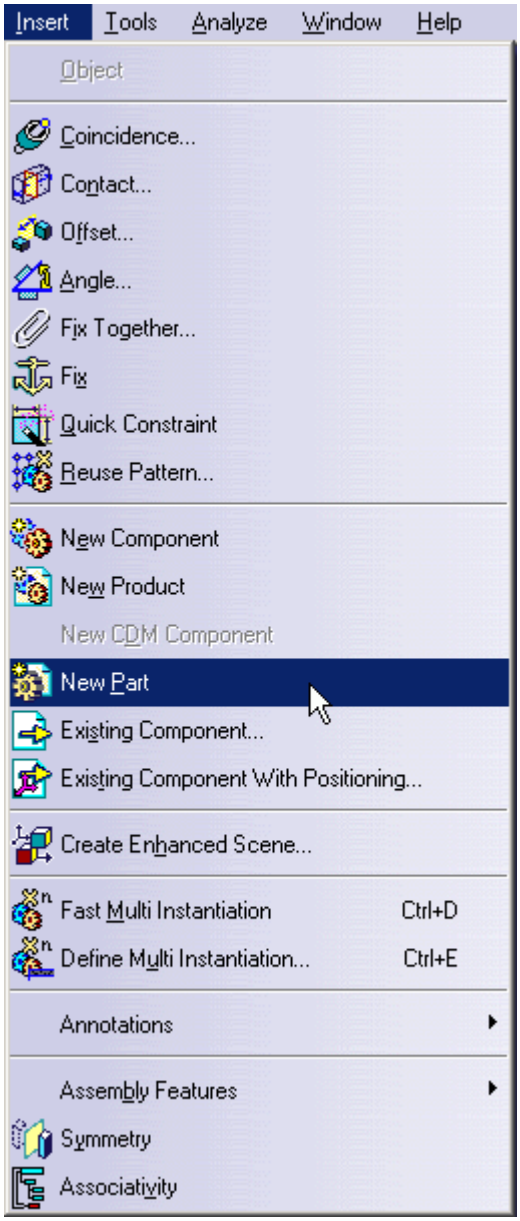
In this example, a fan and fan case are used.



1. Load the assembly in CATIA.

We will now insert a new part into the product. This will form the rotating region of the fluid domain.

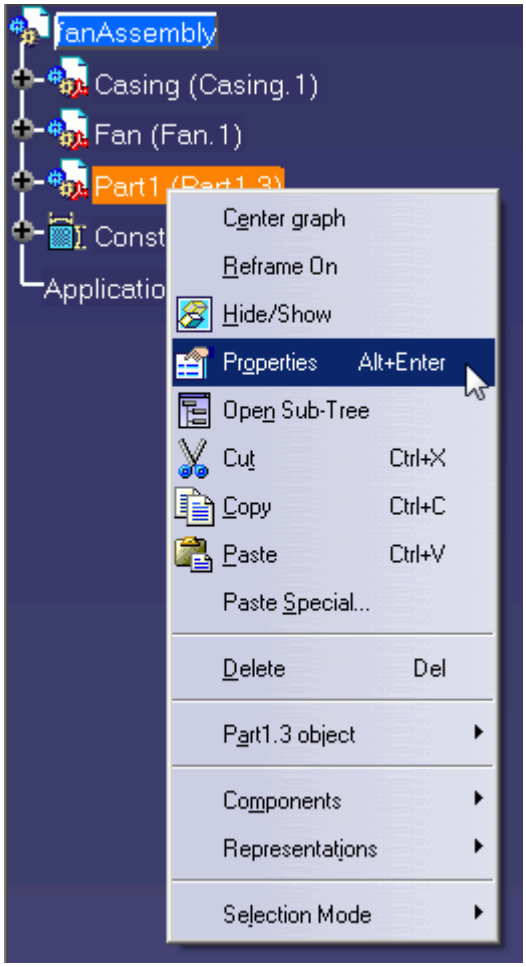
2. Insert a new part into the assembly, **Insert** > **New Part**.



3. Click on the **Assembly** node in the feature tree to insert the part into the assembly.

   A dialog will pop up asking whether a new origin should be defined or not. Choose either option.

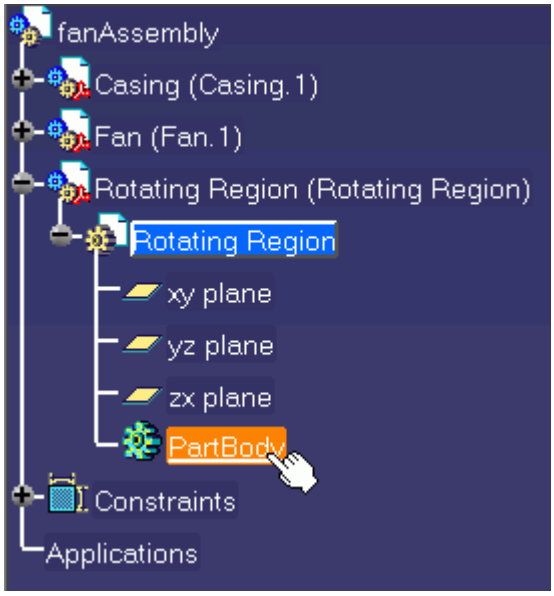
4. Rename the new part by right-clicking on it in the feature tree and selecting **Properties**.



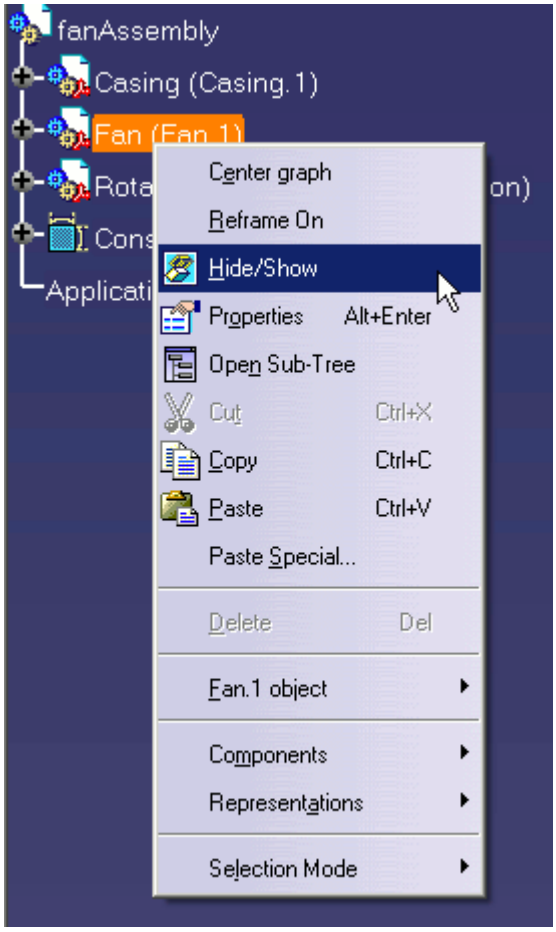
In the *Properties* dialog enter an appropriate name for the new part (in this case it will be called `Rotating Region`) and click **OK** to close the dialog.

5. Double-click the **Rotating Region** part body.

   The part will become active and the Part Design workbench appears. When parts are active the part node will be highlighted in blue.

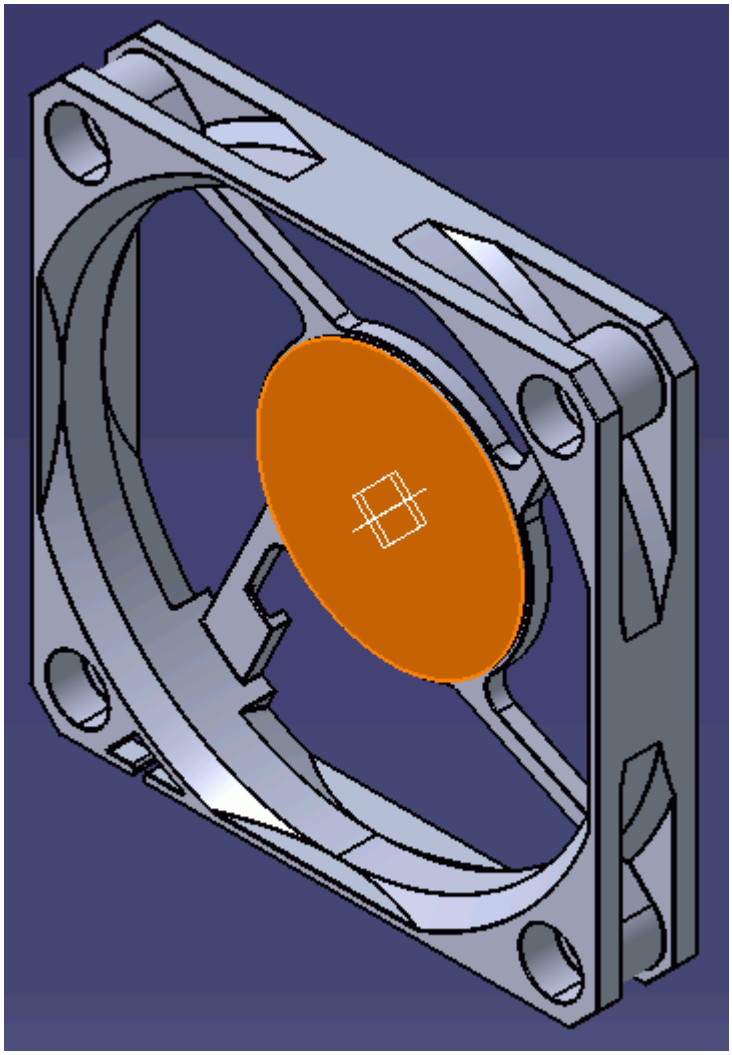


6. Hide the fan casing, right-click on the feature tree or on the geometry and select **Hide/Show**.



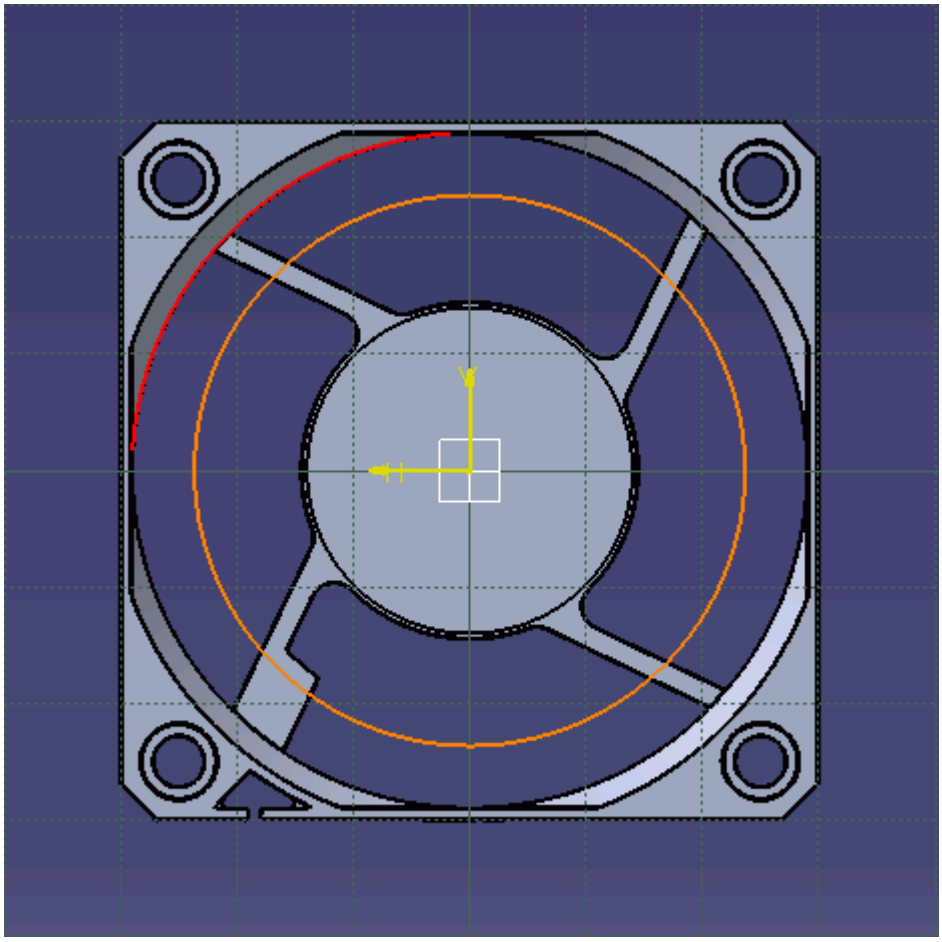
7. Create a sketch on the contact face of the fan casing (where the fan and the fan casing are coincident).

If the part is a native CATIA geometry then this will be possible by simply clicking on the desired face, as shown below.



If the part is an imported geometry, a new plane will need to be created on the desired face before a sketch can be created.

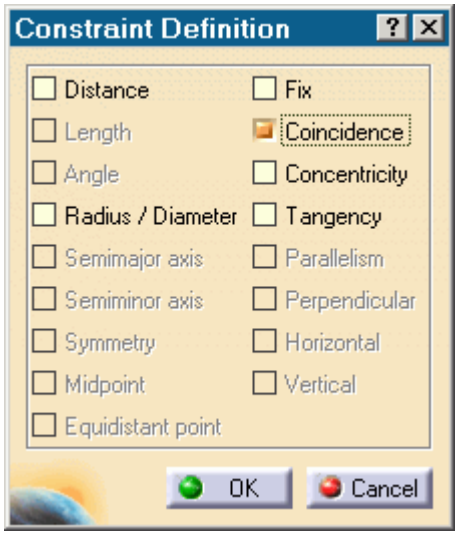
8. Create a circle with the same center point as the fan casing - the diameter of the circle is not important at this stage as it will be made coincident with the fan casing in the next step.
9. Select the circle (shown in orange) and the inner-most edge on the fan casing (shown in red).



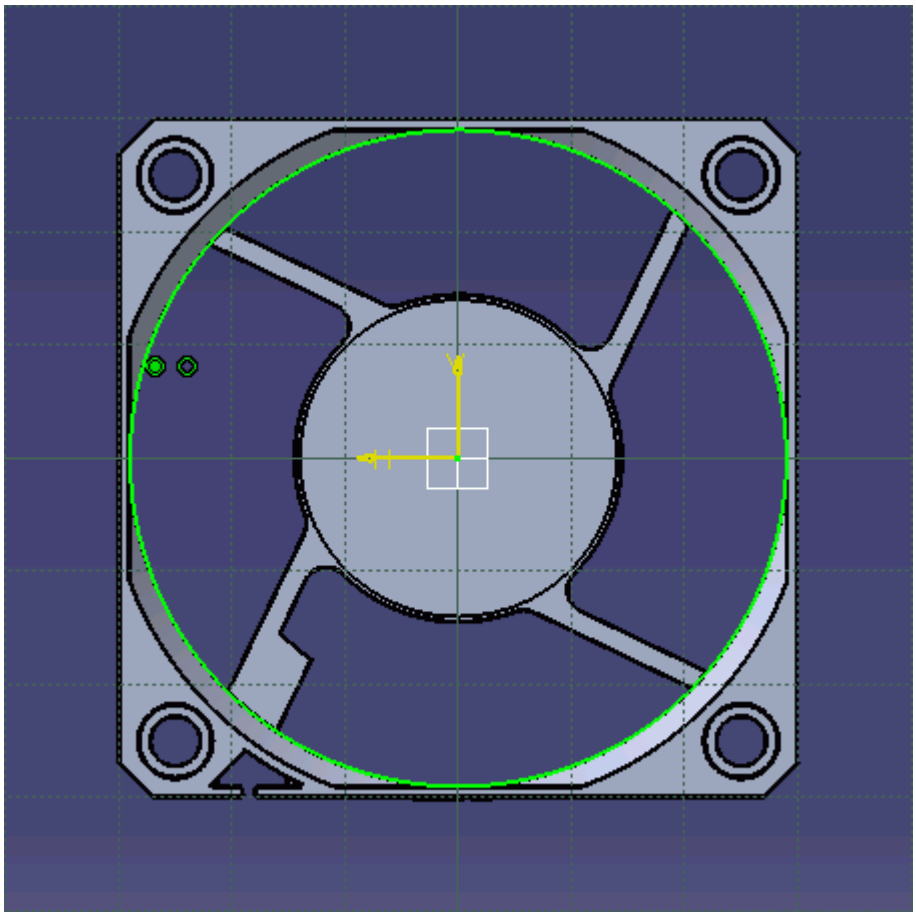
10. Click on the **Constraints Defined in Dialog Box** button.

The *Constraint Definition* dialog will appear.

11. In the *Constraint Definition* dialog, tick the *Coincidence* checkbox.

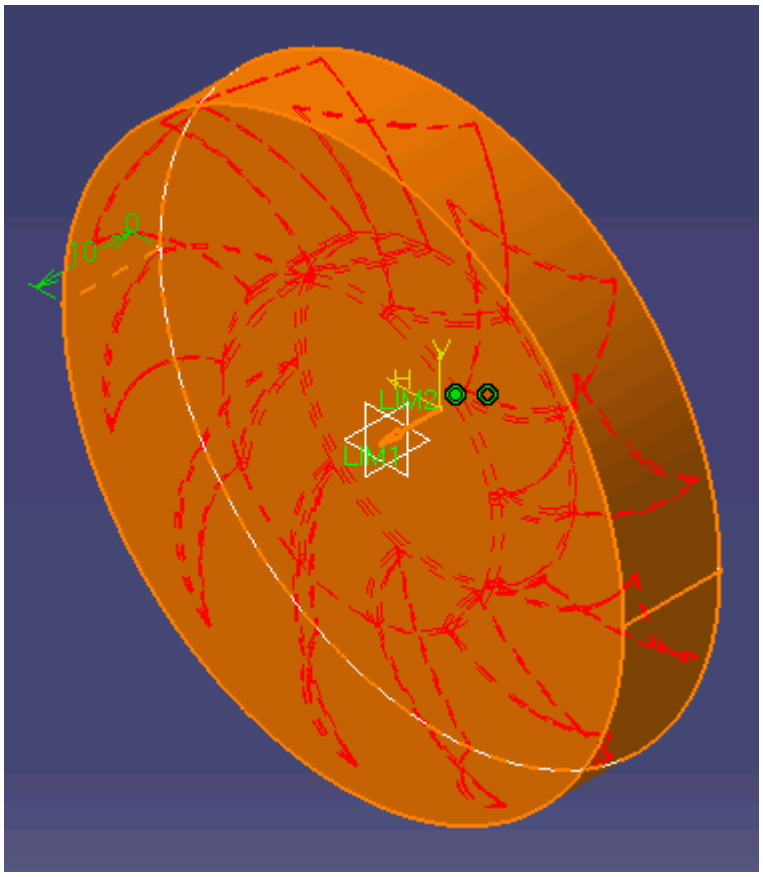


Notice that the sketch automatically snaps to the selected edge on the fan casing. The green color of the sketch entity indicates that it is fully defined.



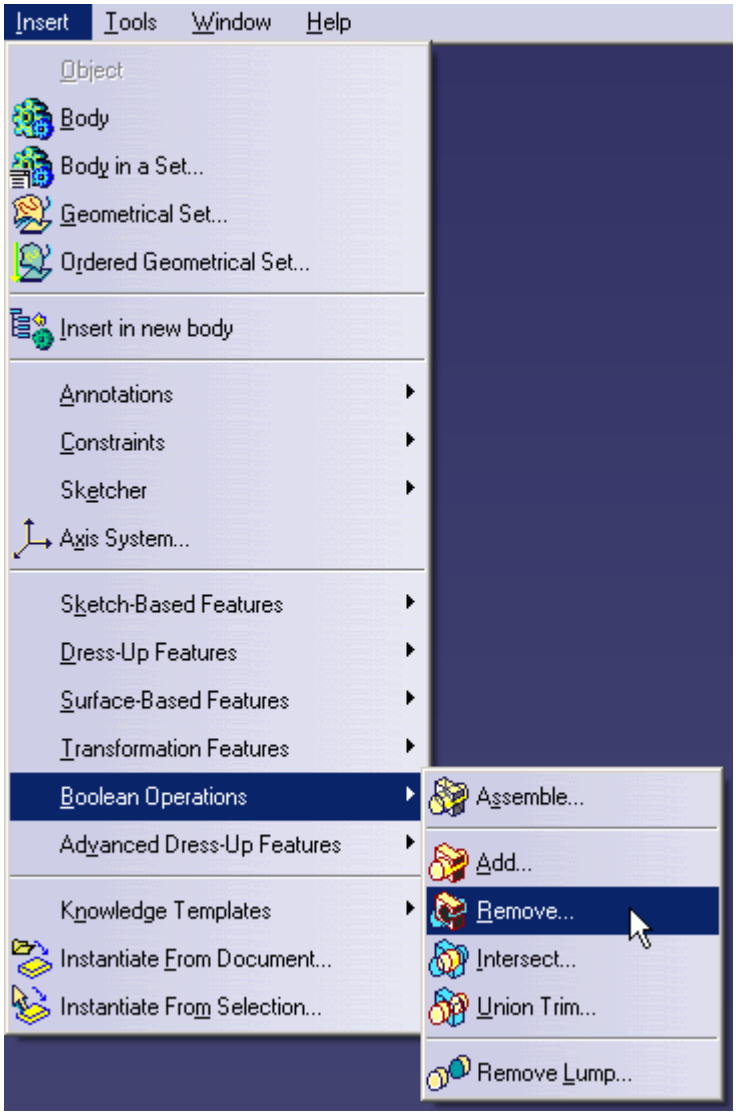
12. Exit the sketch.
13. Hide the fan casing part and show the fan part (using the **Show/Hide** option).
14. Create a pad from the sketch so that it encloses the fan, as shown below.

The fan can be visualized by changing the view to **Shading with Edges and Hidden Edges**

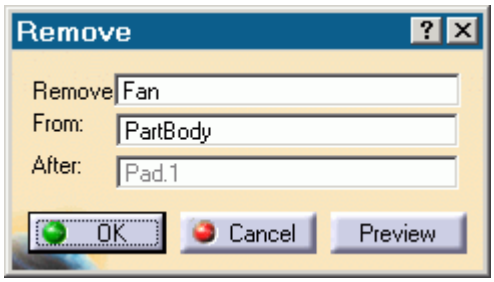


We will now remove the fan from this new pad using a boolean operation.

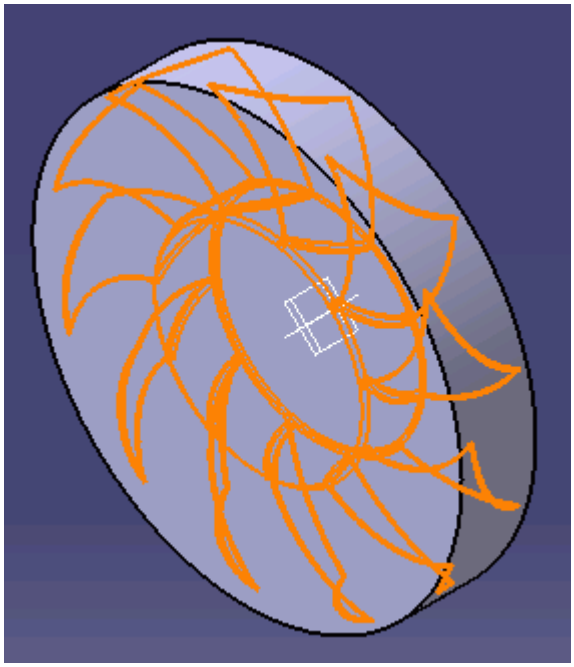
15. Change the view to **Shaded with Edges**.
16. Select **Insert** > **Boolean Operations** > **Remove...**.



17. In the *Remove* dialog, select the fan part (from the feature tree) for the *Remove* field and the pad for the *From* field as shown below.



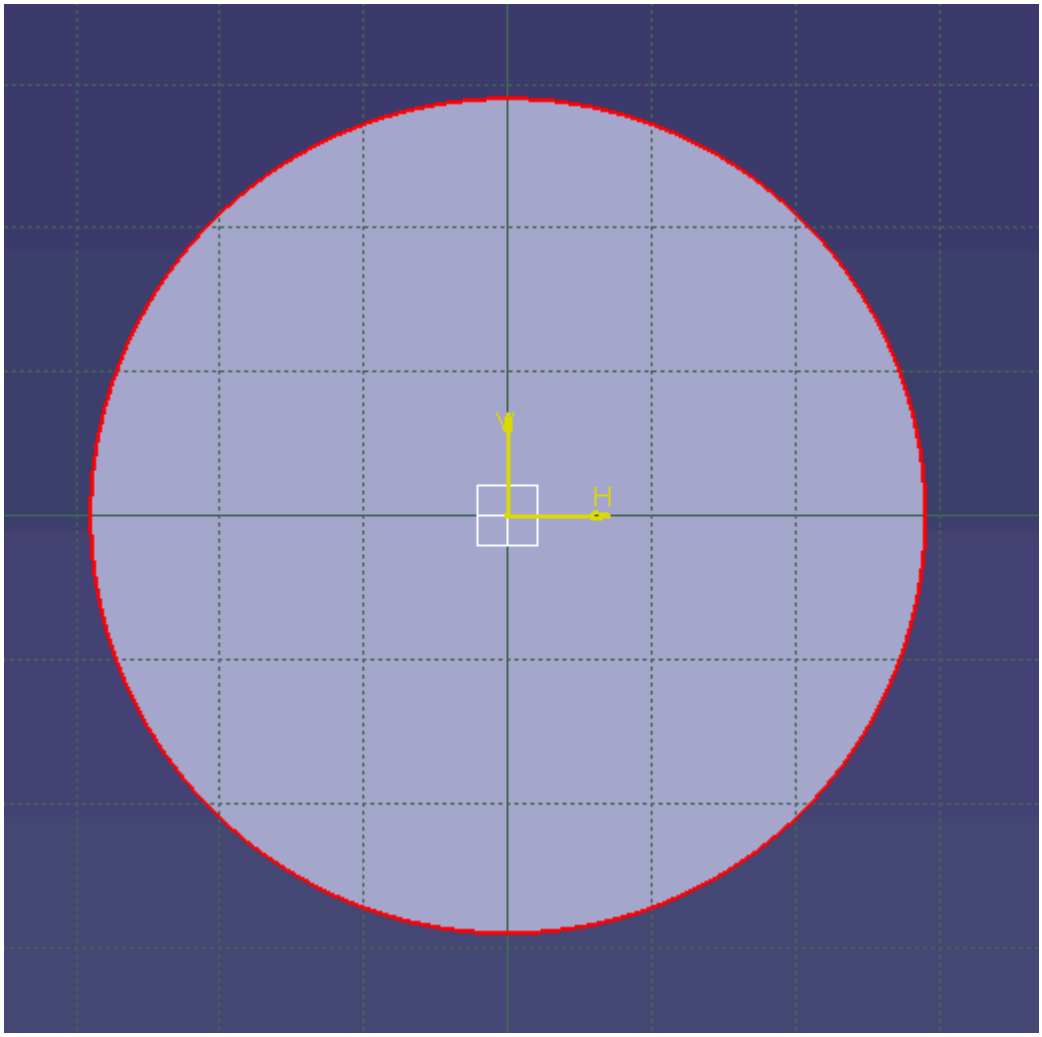
Click **Preview** to show a preview of the operation.



18. Click **OK** to perform the operation and close the dialog.
19. Hide the fan part as it is no longer required.

We will now insert a new part into the product. This will become the inlet region.

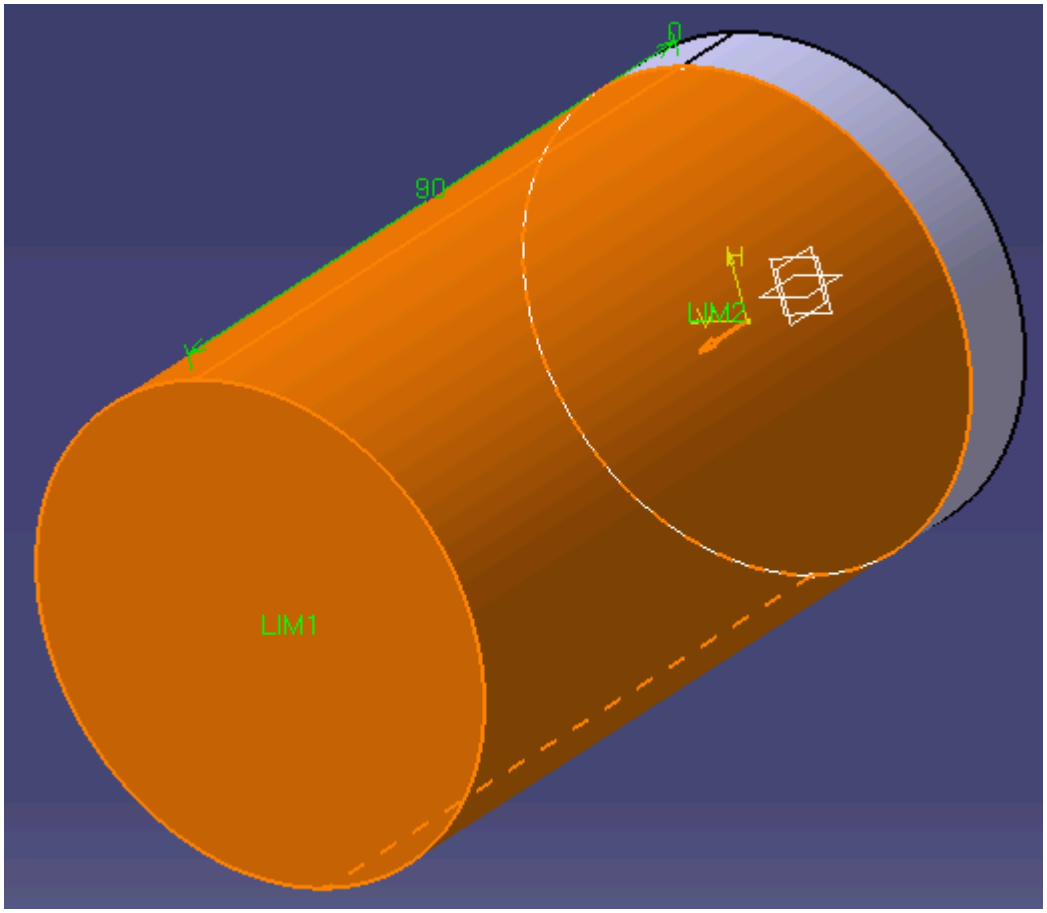
20. Return to the assembly by double-clicking on the root node in the specification tree.
21. Insert another part into the assembly and make it the active part as previously described.
22. Enter the sketch workbench and select the front face of the Rotating Region.
23. Select the outermost edge of the face



24. Use the **Project 3D Elements** tool.

This will project the selected edge from the existing body onto the current sketch. The yellow color of the sketch means that it is associative, so if the diameter of the part changes (for example), the sketch will automatically update to match the change.

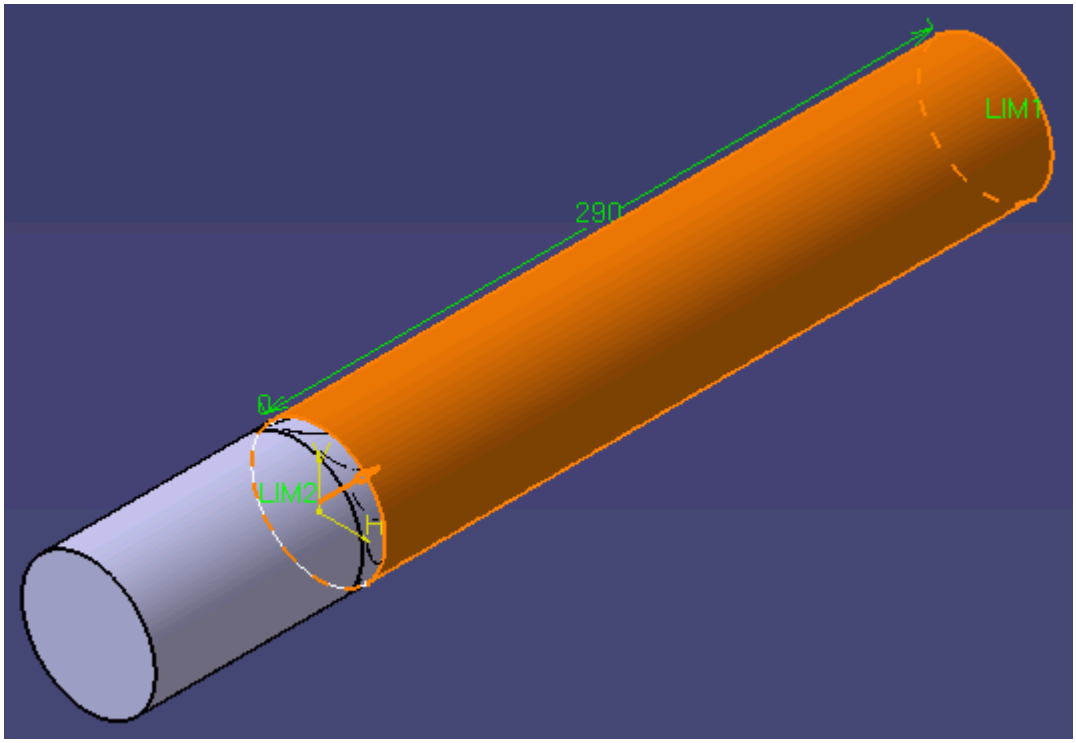
25. Exit the Sketcher workbench and create a **Pad** of appropriate length, from the sketch.



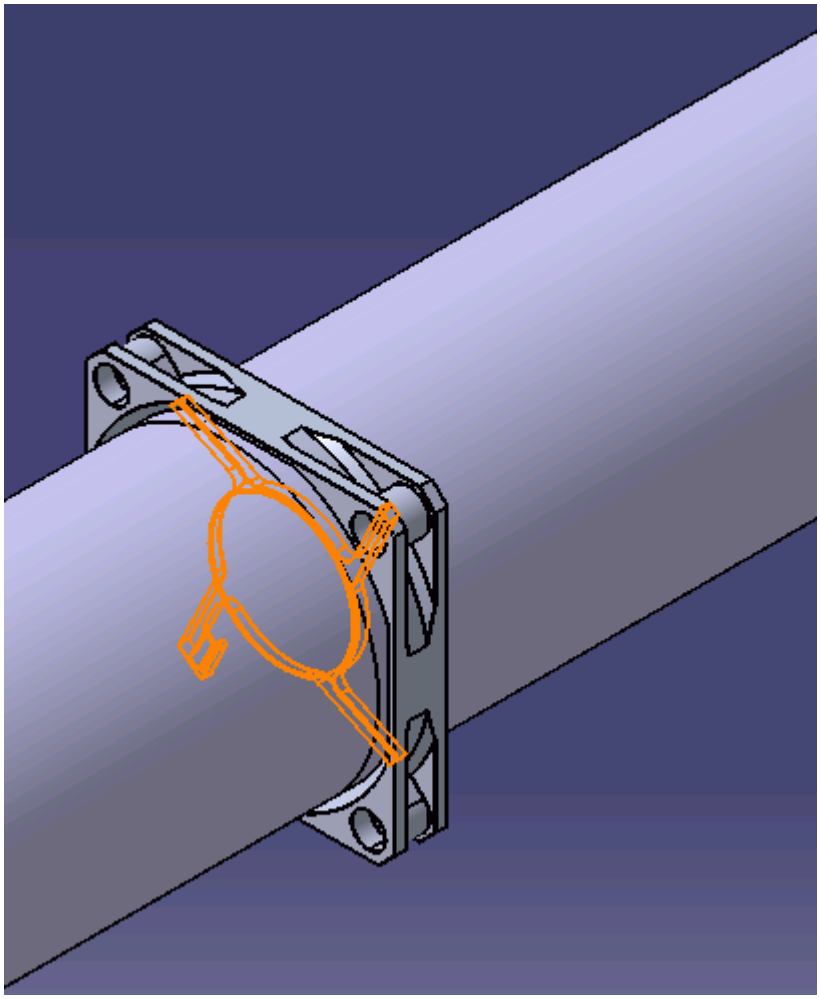
We will now insert a new part into the product. This will become the outlet region. The fan casing contains the fan motor's support arms--these will need to the removed from outlet region.

26. Return to the assembly.
27. Insert another part in the assembly and make it the active part.
28. Create a sketch on the back face of the Rotating Region part,.
29. Selecting the outermost edge of the rotating region and project it to the current sketch, as described previously.
30. Create a pad, of appropriate length, that extends downstream.

    If the direction of the pad is pointing the wrong way, use the **Reverse Direction** button in the *Pad Definition* dialog.

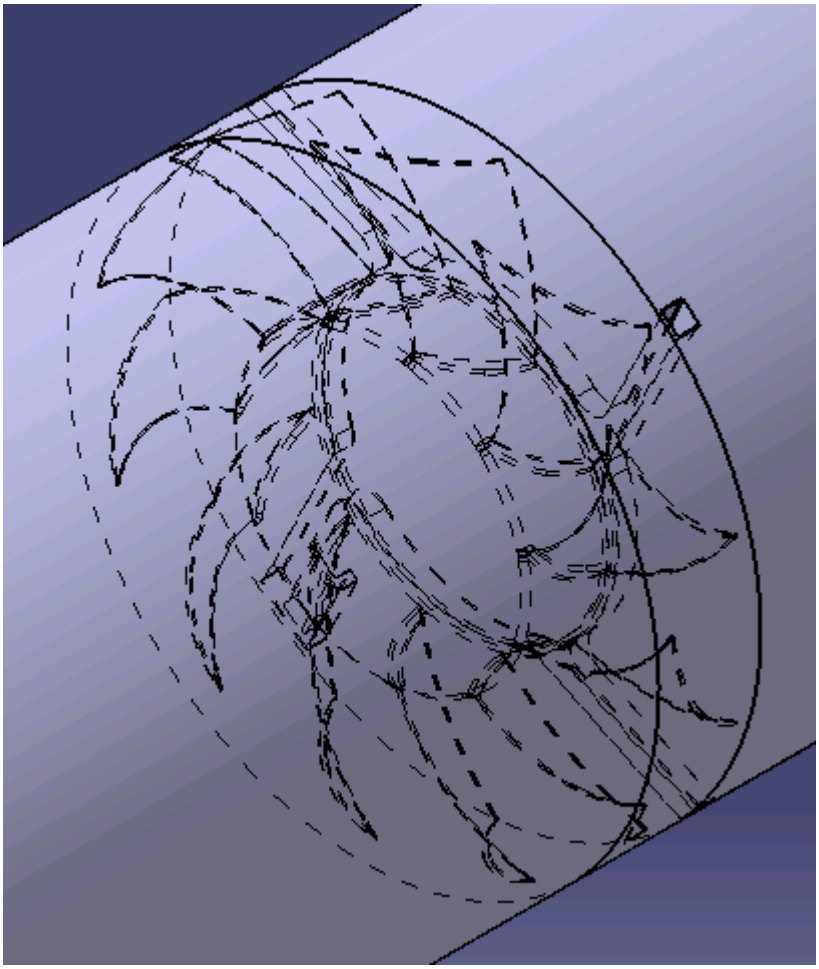


31. Show the fan casing part (right-click on the node in the specification tree and select **Show/Hide**).
32. Remove the fan casing from the Outlet Region, as described previously steps. A preview of this operation is shown below.

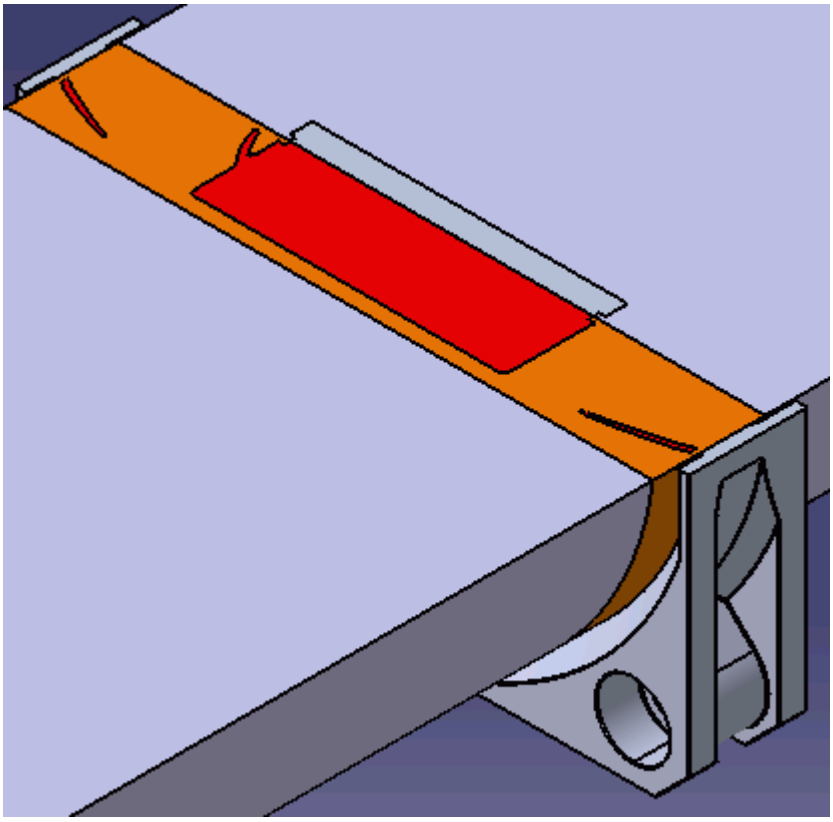


33. Return to the assembly and hide the fan casing.

The final fluid domain is shown below. The view was changing to **Shading with edges and hidden edges**.

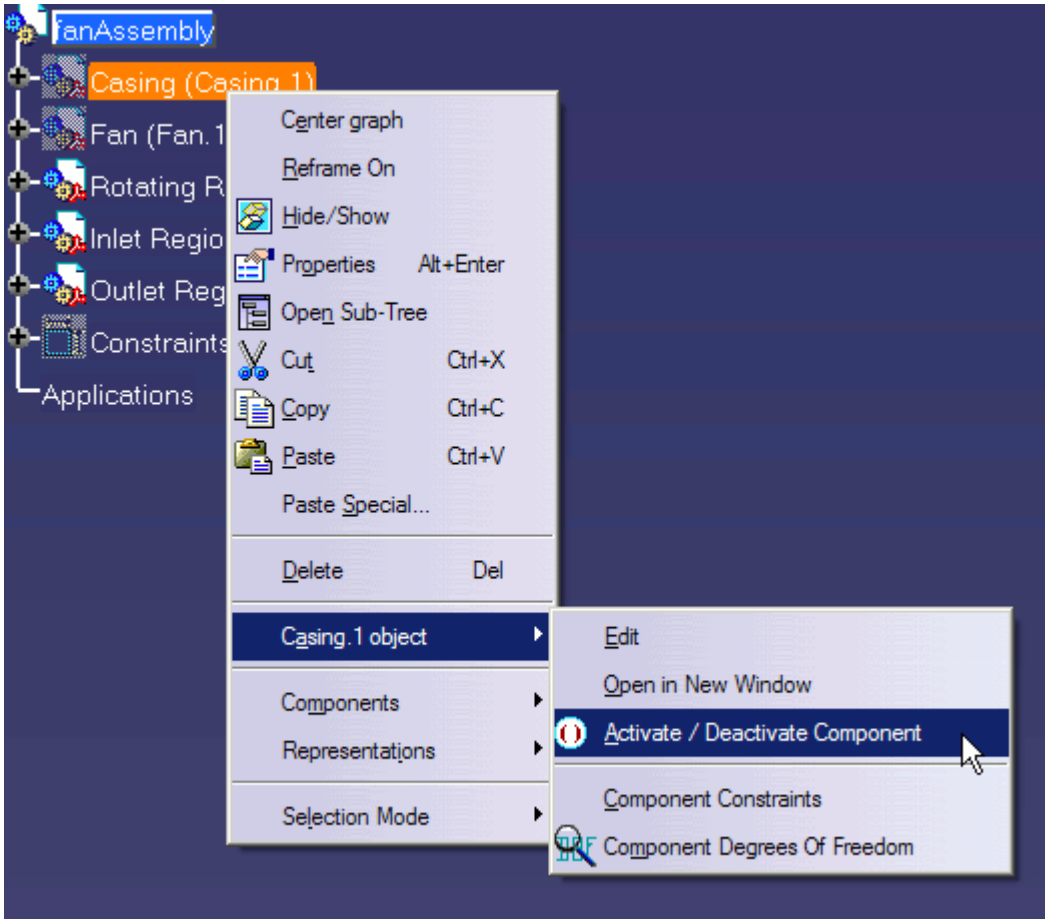


A cross-section of the fluid domain is shown below. Note that all components were made visible.



The fan is shown in red, the fan casing is shown in white, the Inlet and Outlet fluid regions are shown in grey, and the Rotating Region is shown highlighted in orange.

34. Deactivate all parts that you do not want to take part in the analysis, such as the fan and fan casing, by right-clicking on each node and selecting **[instance_name]** > **object** > **Activate/Deactivate Component**.



The geometry is now ready for cad client analysis. Deactivated parts will not be included in the analysis. The three regions that form the fluid domain all have a conformal match, so it will be possible to create the interfaces and obtain a topologically valid mesh in STAR-CCM+ without any further modifications to the geometry.

Imprinting Parts

Most CAD model assemblies are not suitable for meshing as the parts in the assembly are disjointed, where the coincident faces are overlapping. Attempting to generate a mesh with such an assembly will result in a meshing error. To rectify this problem, the parts in an assembly must be imprinted together.

Imprinting can also be used to split surfaces for boundary region applications. An example of such an application would be to model the area that a fan would occupy on the inside of computer case, such as in the third STAR-CAT5 tutorial, where a smaller section of the surface is needed to represent the fan, rather than the entire inner surface of the computer case.

Contents:

Imprinting Parts in an Assembly
Imprinting Surfaces on Parts for Boundary Region Applications

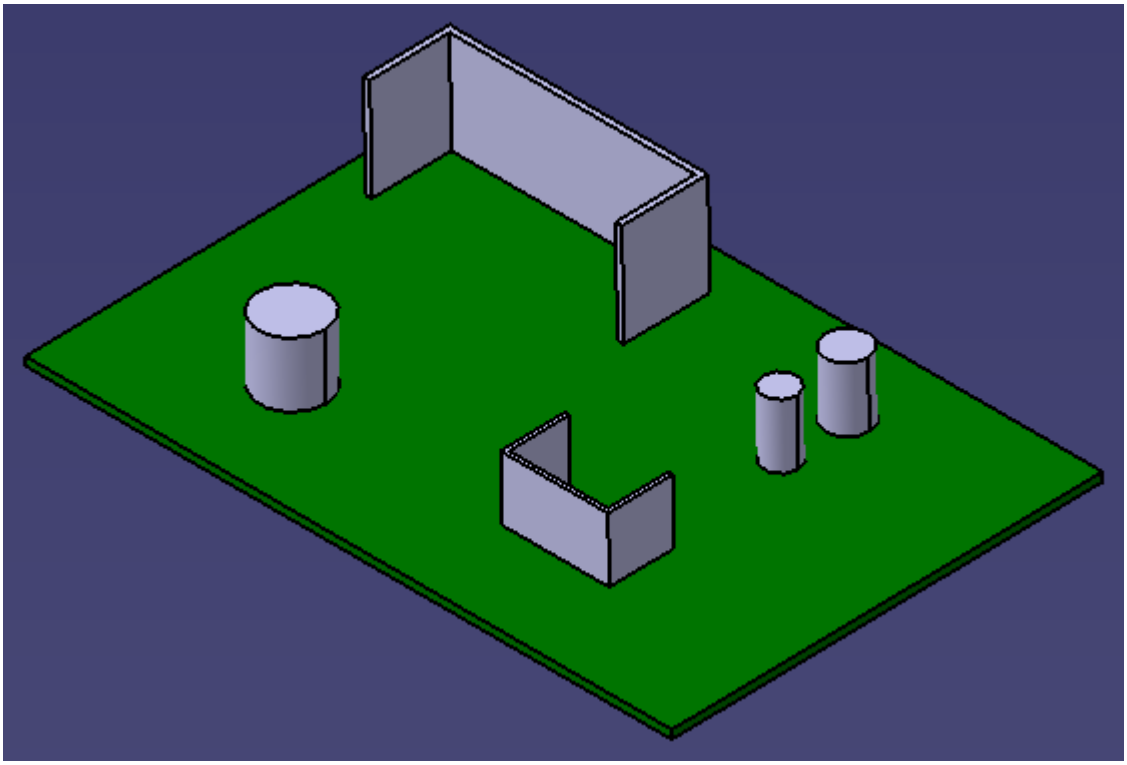
Imprinting Parts in an Assembly

In cases containing multiple adjacent parts, such as in Conjugate Heat Transfer cases, it is essential to imprint the profiles of the coincident faces of the smaller parts onto the face of the larger part. This will ensure that the volume mesh created by STAR-CCM+ contains interfaces between the adjacent parts.

In general, the steps to follow in creating the fluid domain for multiple regions are:

- Work within the product, making the part that requires imprinting the work part.
- Extract the contact surfaces from the other components in the product.
- Sew the extracted surfaces onto the face you want to imprint.

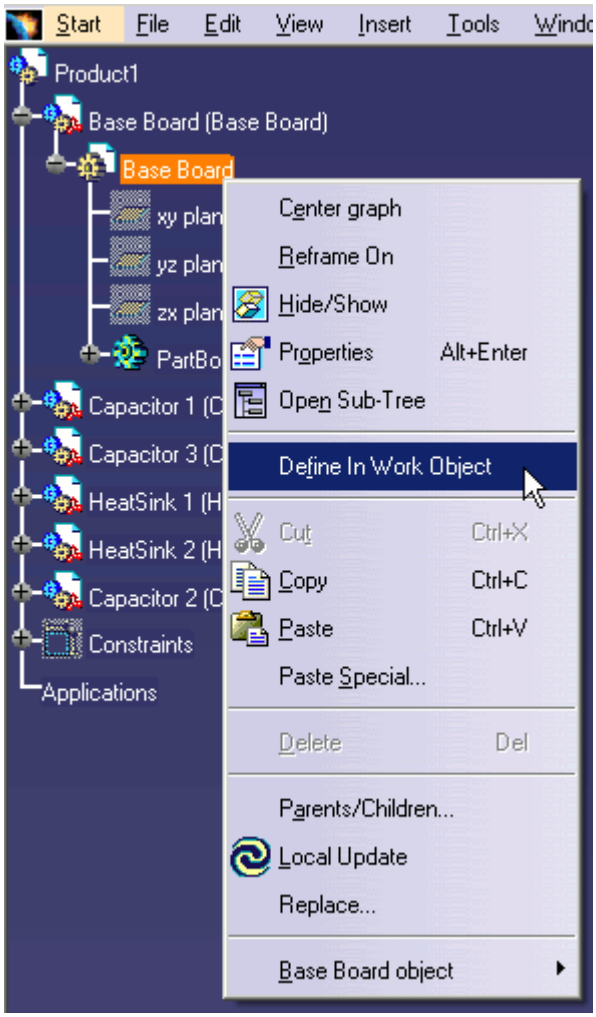
The simple circuit board shown below requires imprinting before the mesh can be generated.



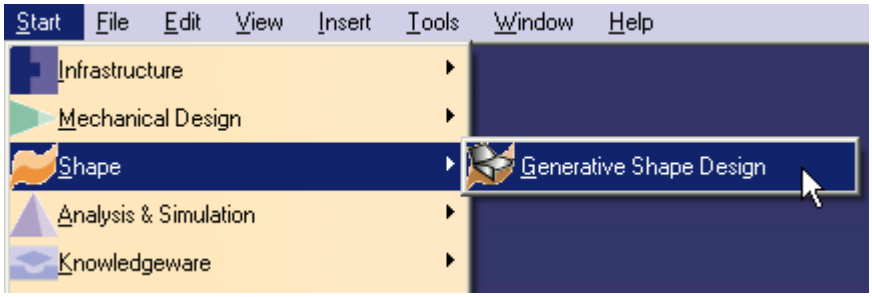
1. Load the product file in CATIA.

In this example, the green circuit board needs to be imprinted.

2. Right-click Base Board and select **Define in Work Object**.

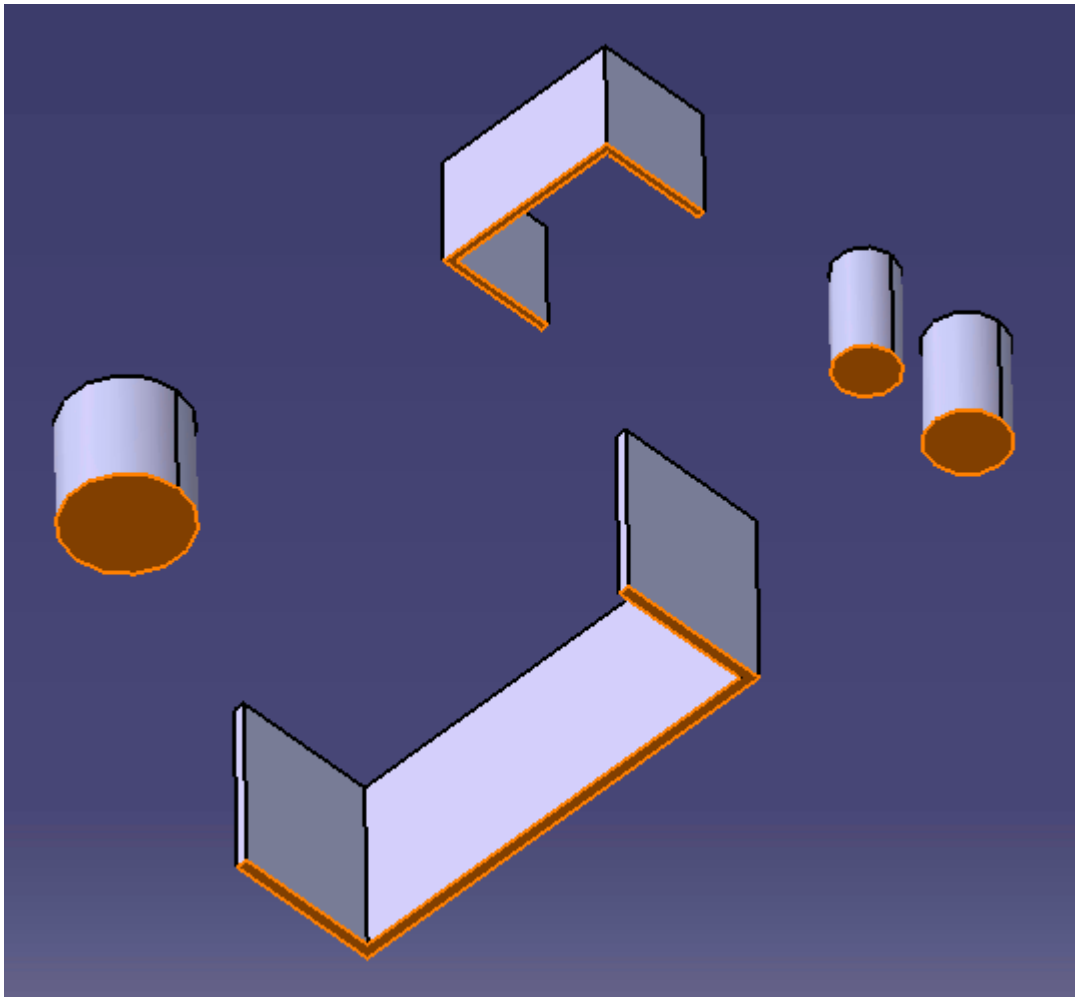


3. Switch to the **Generative Shape Design** workbench, by clicking on **Start** > **Shape** > **Generative Shape Design**.



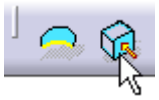
4. Hide the base board by right-clicking on it in the specification tree and select **Hide/Show**.
5. Select all the contact faces on the other parts in the product, holding down the <`Ctrl`> key.

   In this example the bottom surfaces of the capacitors and heat sinks are selected.



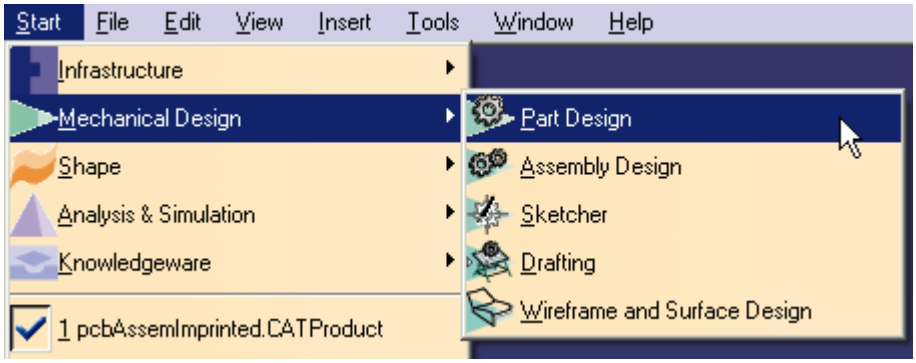
We will now extract the selected faces.

6. Click on the **Extract** button.



Highlighting multiple faces before using the **Extract** function will automatically perform the same operation for each selection. If the **Extract** function was used first this would only allow one face to be selected at a time.

7. Switch back to the Part Design workbench, **Mechanical Design** > **Part Design.**



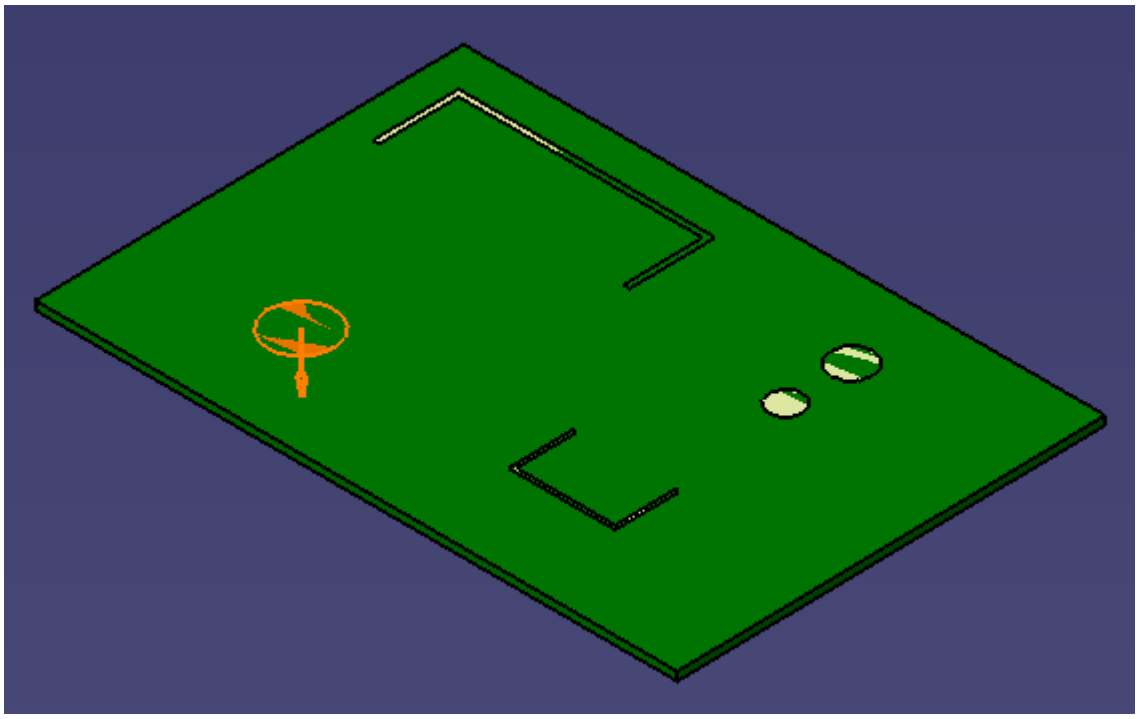
8. `Show` the circuit board and `hide` the capacitors and heat sinks. The extracted faces will appear in yellow and overlap the existing top face of the circuit board.

We will now sew the extracted surfaces onto the face that we want to imprint.

9. Use the **Sew** function to sew one of the extracted surfaces to the base board.

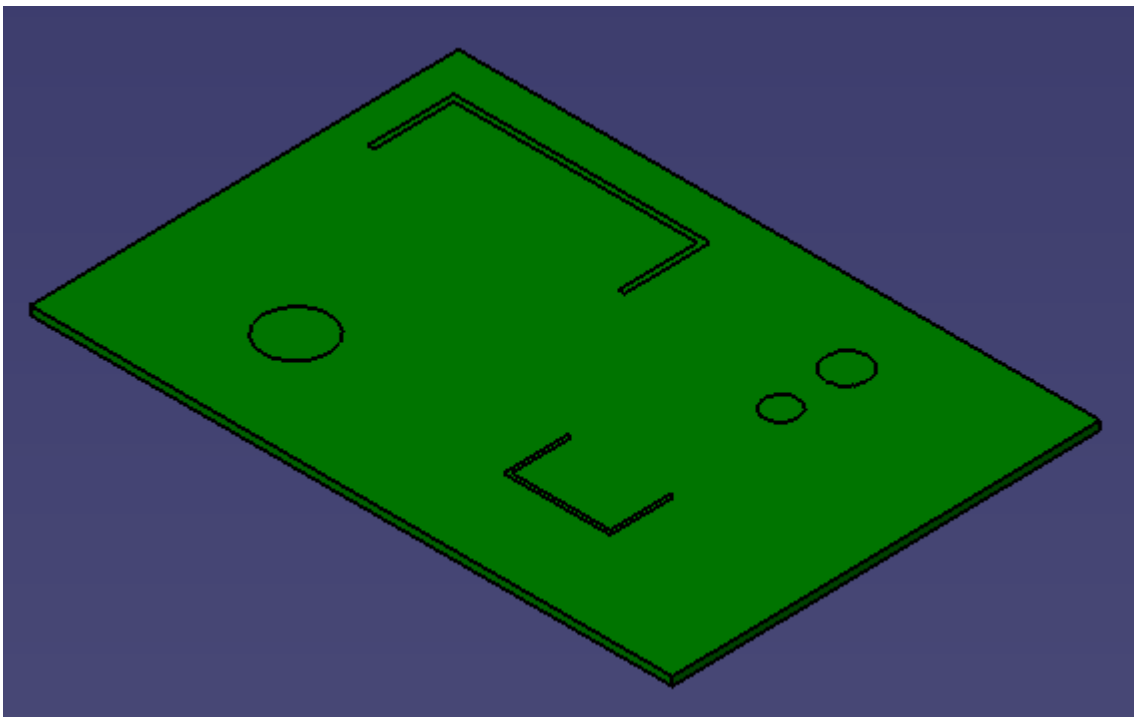
Ensure that the **Simplify Geometry** option in the *Sew Surface Definition* dialog is cleared.

Also note that the sew direction arrow should be pointing towards the surface of the body to be sewn.



10. Repeat the same steps for the other extracted surfaces.

The base board face is imprinted with the contact edges of the capacitors and heat sinks, as shown below.



A high quality mesh with fully defined interfaces can now be successfully generated for this assembly. Deactivated parts are not included in the STAR-CCM+ analysis, so for CHT cases, such as the above example, ensure that all relevant parts are active.

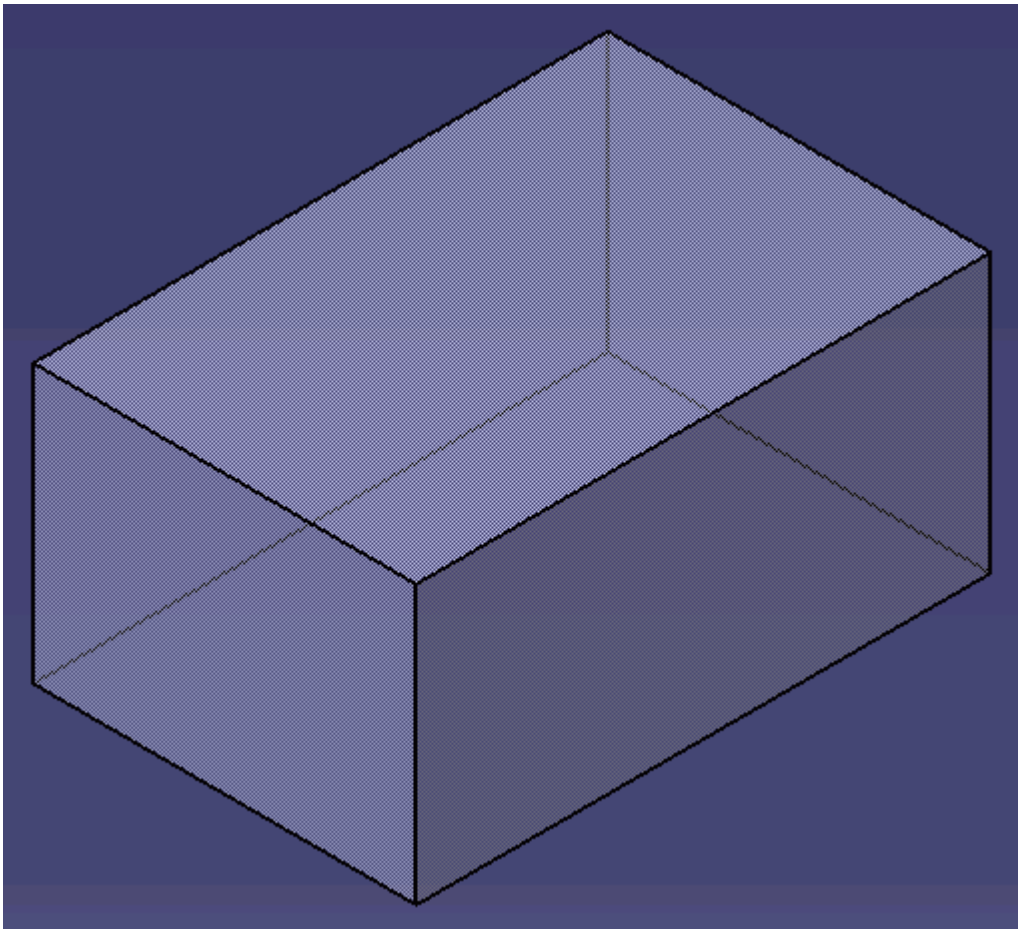
Imprinting Surfaces on Parts for Boundary Region Applications

Faces in a part body can be split into smaller faces for boundary application.

The general procedure to follow is:

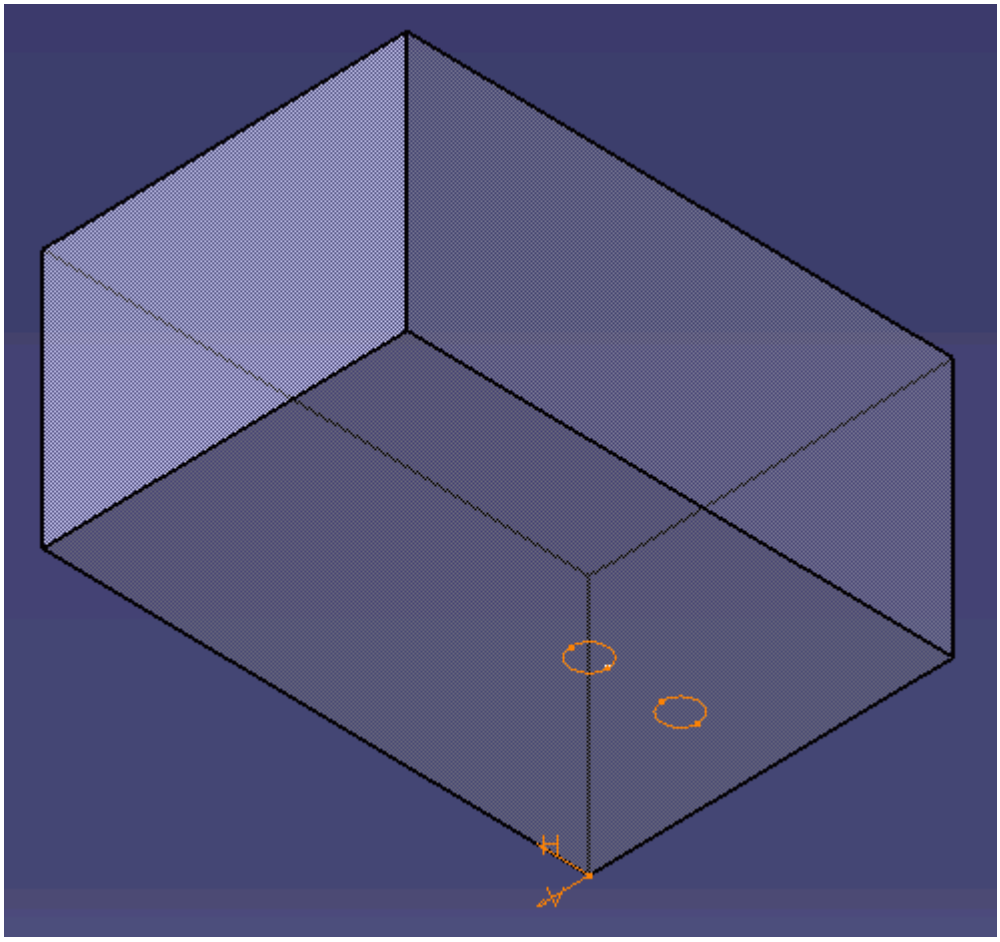
- Create a sketch on the surface you want to split. The shape of the sketch should correspond to the shape of the new face after it is split.
- Fill the sketch so that is becomes a surface.
- Sew the surface onto the face that requires imprinting.

A simple example of an aeration tank is shown below. The image shows the tank, the two air injectors need to be imprinted onto the base of the tank.

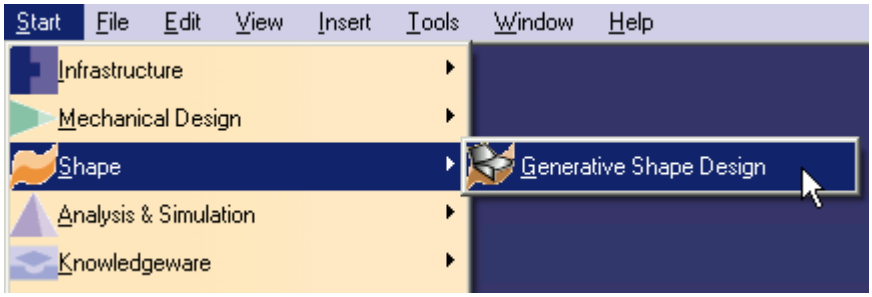


1. Load the geometry part into CATIA.
2. Create a sketch on the face that requires imprinting.

   In this case the air injectors are sketched onto the tank's bottom face.



3. Switch to the Generative Shape Design workbench, by clicking on Start**Shape** > **Generative Shape Design**.

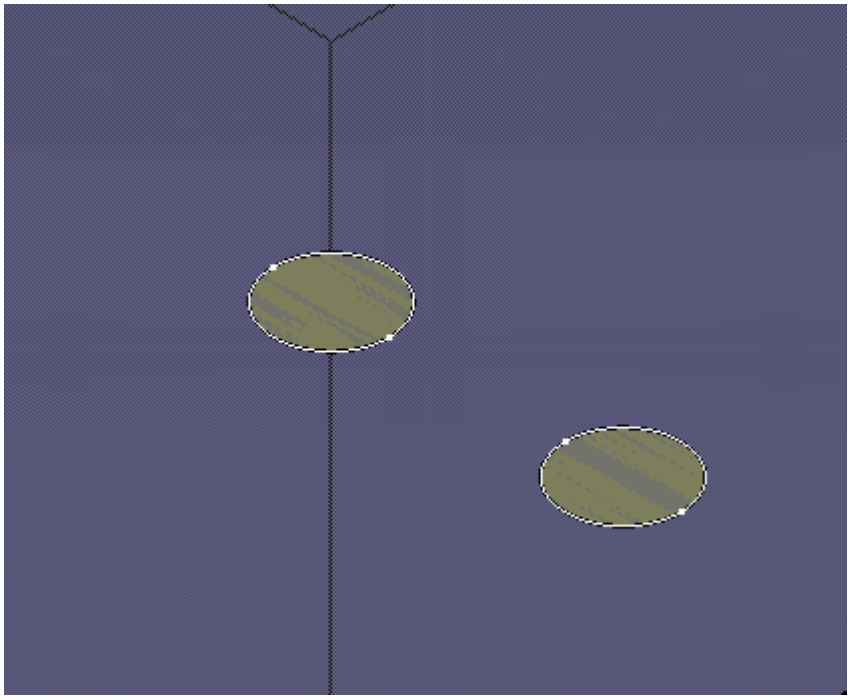


4. Use the Fill tool to create surfaces from the sketch profiles.

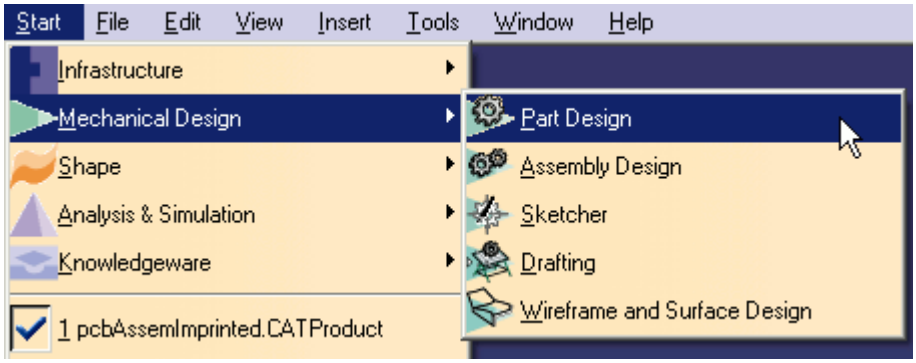
The *Fill Surface Definition* dialog appears.

5. Select the sketch that contains the profiles you want to imprint.
6. Click **OK**.
7. In the *Multi-Result Management* dialog, select the keep all the sub-elements radio button and click *OK*.

   The sketches are now filled in with a planar surface as shown below.

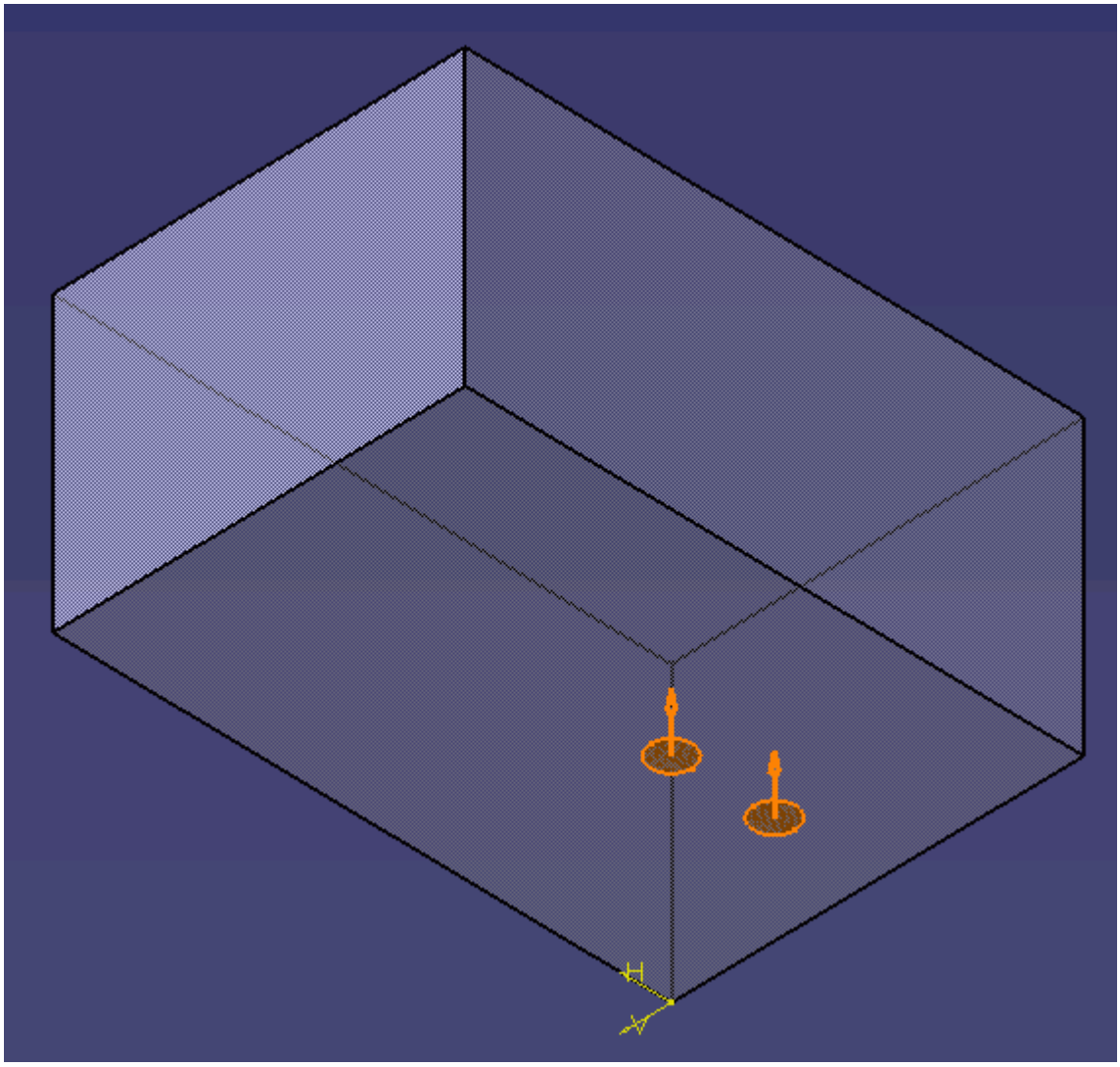


8. Switch back to the Part Design workbench, **Mechanical Design** > **Part Design**.



9. Use the **Sew** function to sew the surfaces to the bottom face of the tank.

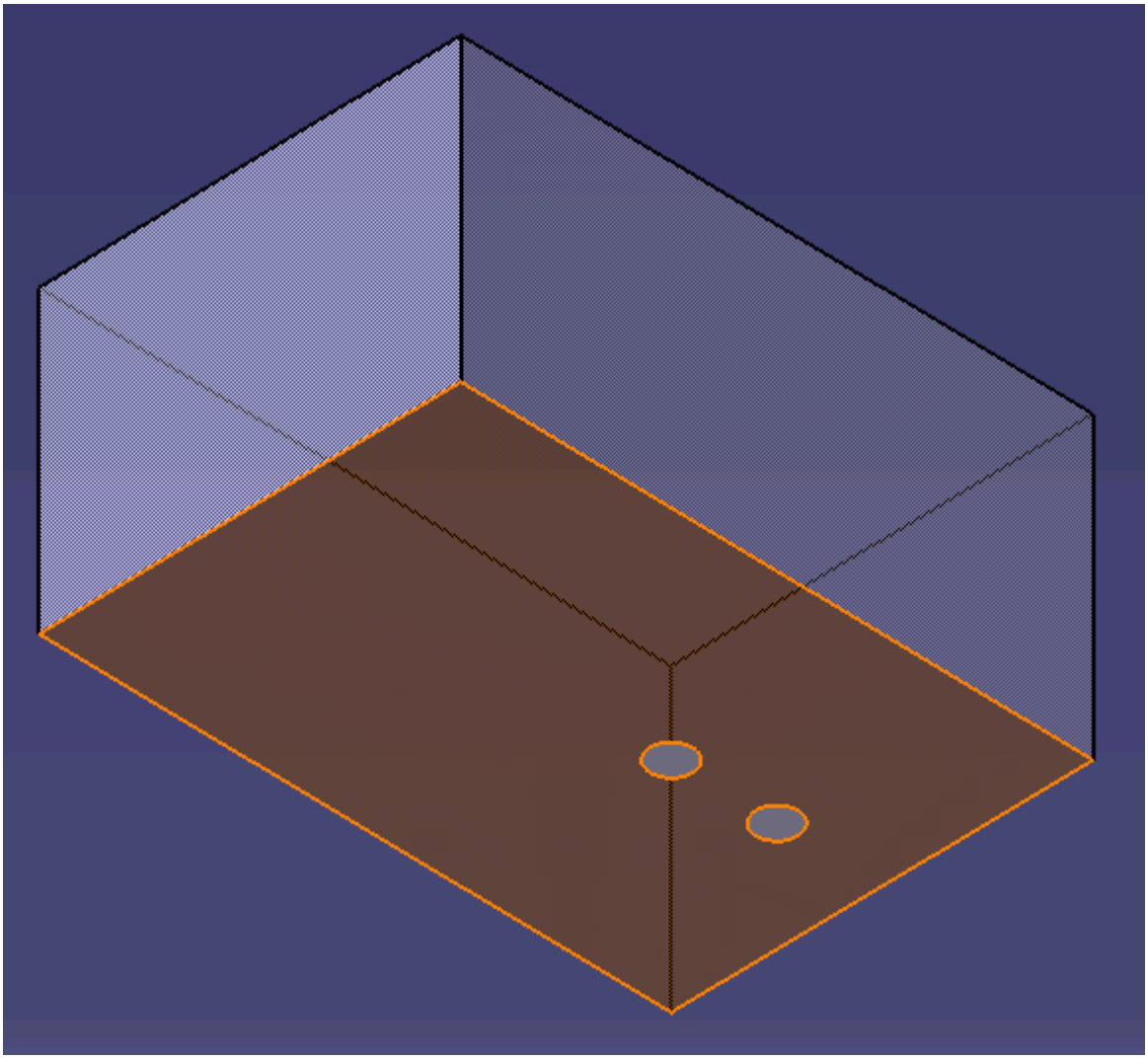
Ensure that the **Simplify Geometry** option in the *Sew Surface Definition* dialog is cleared.



Also note that the sew direction arrow should be pointing towards the surface of the body to be sewn.

10. Hide the fill feature and the sketch on the feature tree by right-clicking and selecting **Show/Hide**.

The base of the tank is now imprinted with the air injector profiles. Selecting the bottom face shows that the injectors are now separate faces that can be selected independently from each other.



The separate faces can now be defined as boundary regions.

STAR-CAT5 Tutorials

Several tutorials are provided for the STAR-CAT5 CAD Client.

These tutorials provide an overview of the major features available in the client and illustrate some of the applications where you can use STAR-CAT5.

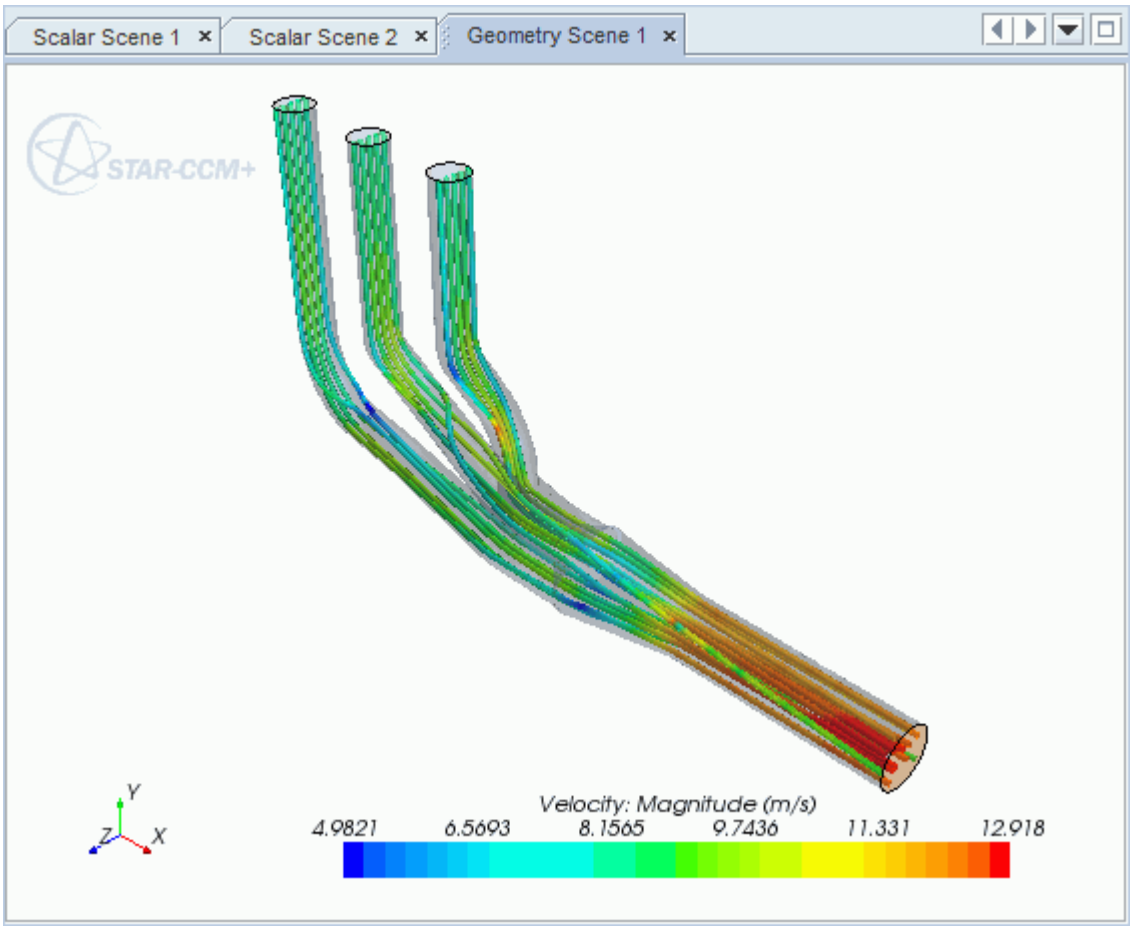
Contents:

Cold Flow Through a Motorcycle Exhaust Manifold
Porous Media Flow
Aircraft Wing Deicing System
Rotating Flow Inside a Graphics Card
External Flow Over a Car
Geometry Transfer: CAD to Simulation

Cold Flow Through a Motorcycle Exhaust Manifold

This tutorial demonstrates how to setup and run a CFD calculation in the STAR-CAT5 environment.

In this tutorial, the model is an exhaust manifold from a three-cylinder motorcycle engine.



Contents:

Opening the CAD Model
Selecting Fluid Physics Models
Defining Boundary Conditions
Setting the Mesh Pipeline
Modifying Solution Controls
Reviewing the Simulation Set Up
Generating the Surface Mesh
Generating the Volume Mesh
Building and Running
Post-Processing Results

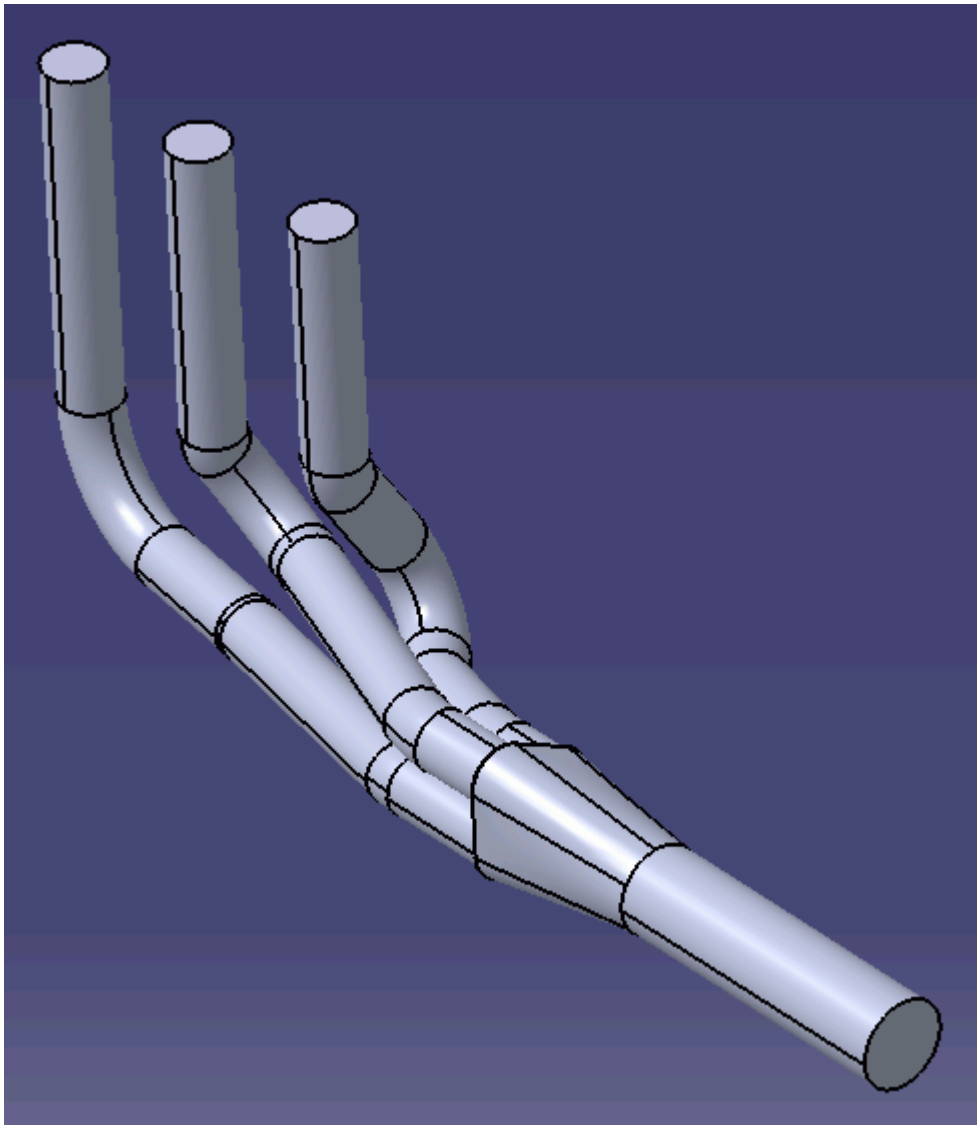
Opening the CAD Model

A directory containing the CAD geometry is provided with the STAR-CCM+ distribution.

This directory is located in `[STAR-CCM+_INSTALL_DIR]\doc\startutorialsdata\starcat5\data`.

1. Copy directory `data` to your own disk space and open the part file `exhaustManifold.CATPart` in STAR-CAT5.

   See Launching STAR-CAT5.
2. Position the geometry as shown below.



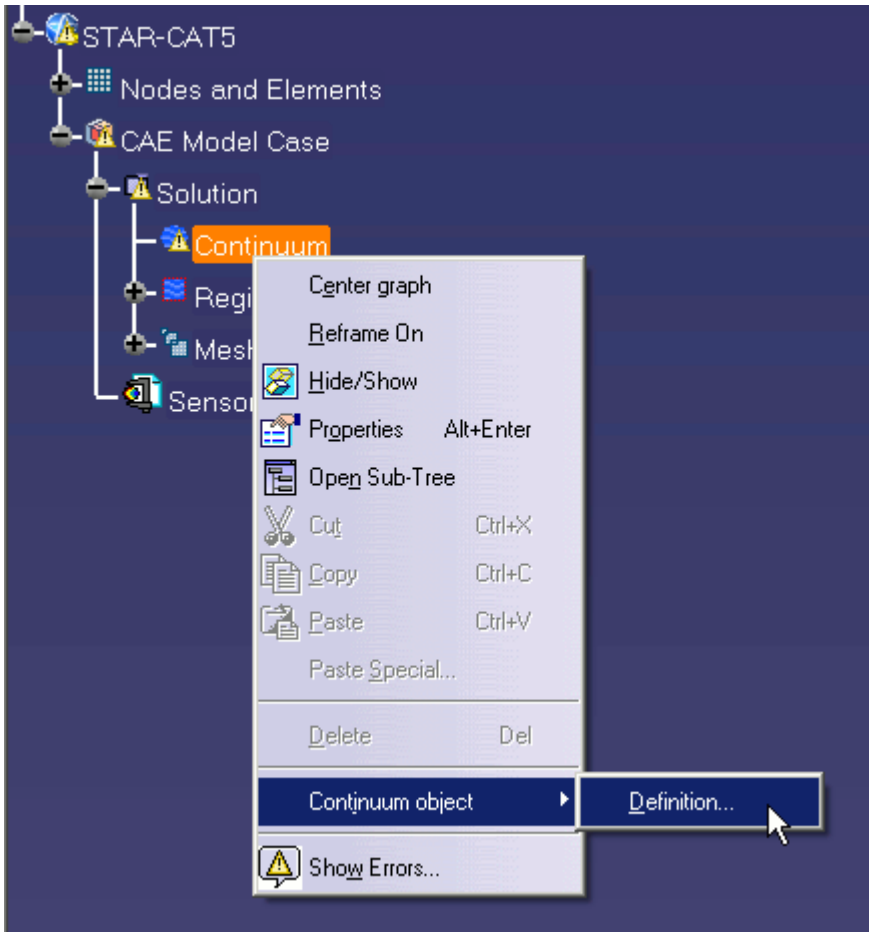
The STAR-CAT5 menu appears at the top of the screen.The STAR-CAT5 object tree is located on the left of the screen.

3. Save the analysis as `exhaustManifold.CATAnalysis`.

Selecting Fluid Physics Models

The first step in setting up this tutorial is to define the physics of the problem. For this case, the flow is turbulent and isothermal.

1. Right-click the **Continuum** node in the STAR-CAT5 tree and select **Continuum object** > **Definition...**.



   The *Continuum Settings* dialog appears.

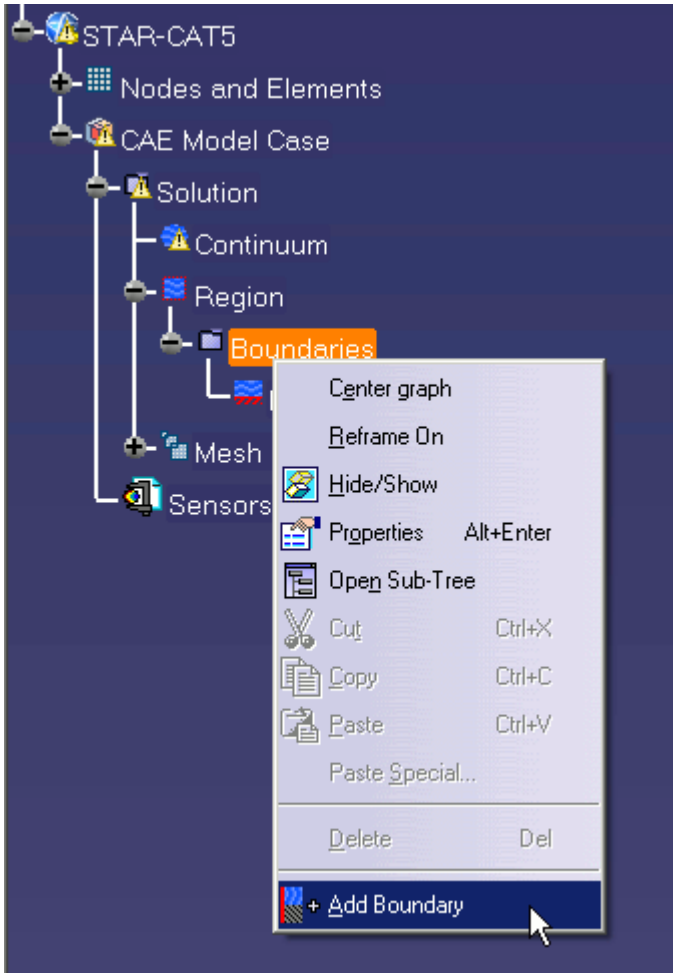
2. Deactivate the **Temperature** option in the *Optional Physics Models*.
3. Click **OK**.

Defining Boundary Conditions

You define the inlet and outlet boundary conditions for the simulation.

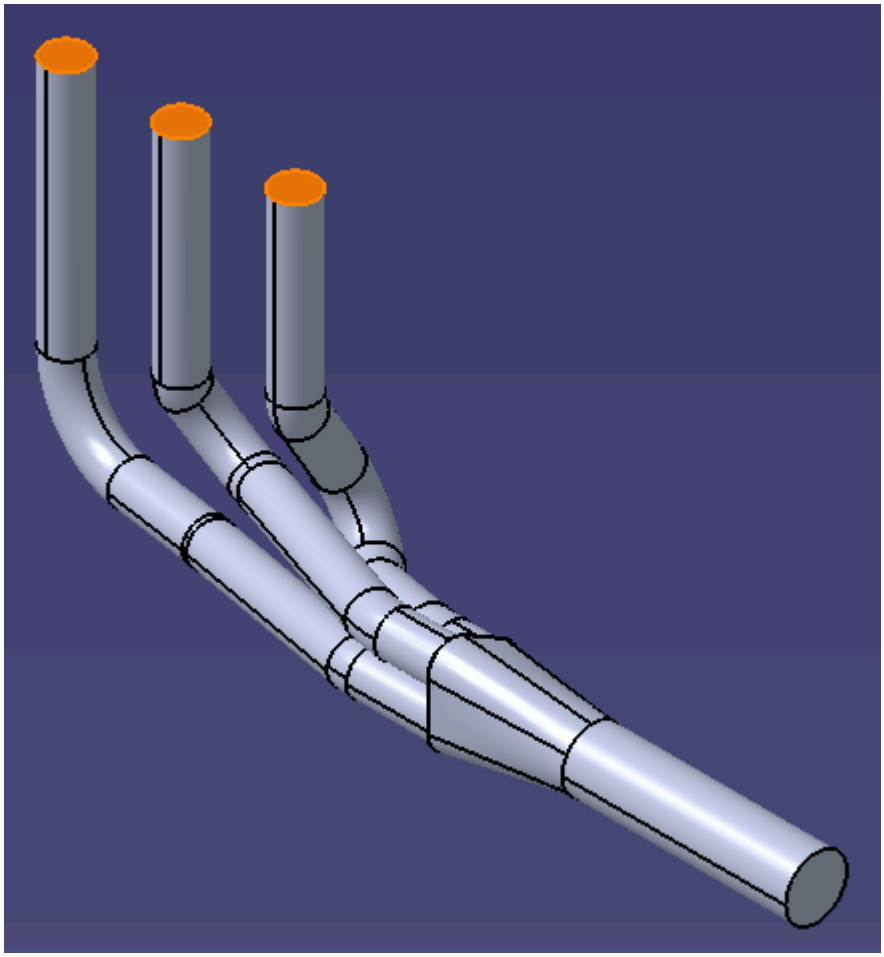
Create the inlet boundary first:

1. Right-click the **Region** > **Boundaries** node and select **Add Boundary**.



The *Boundary Settings* dialog appears.

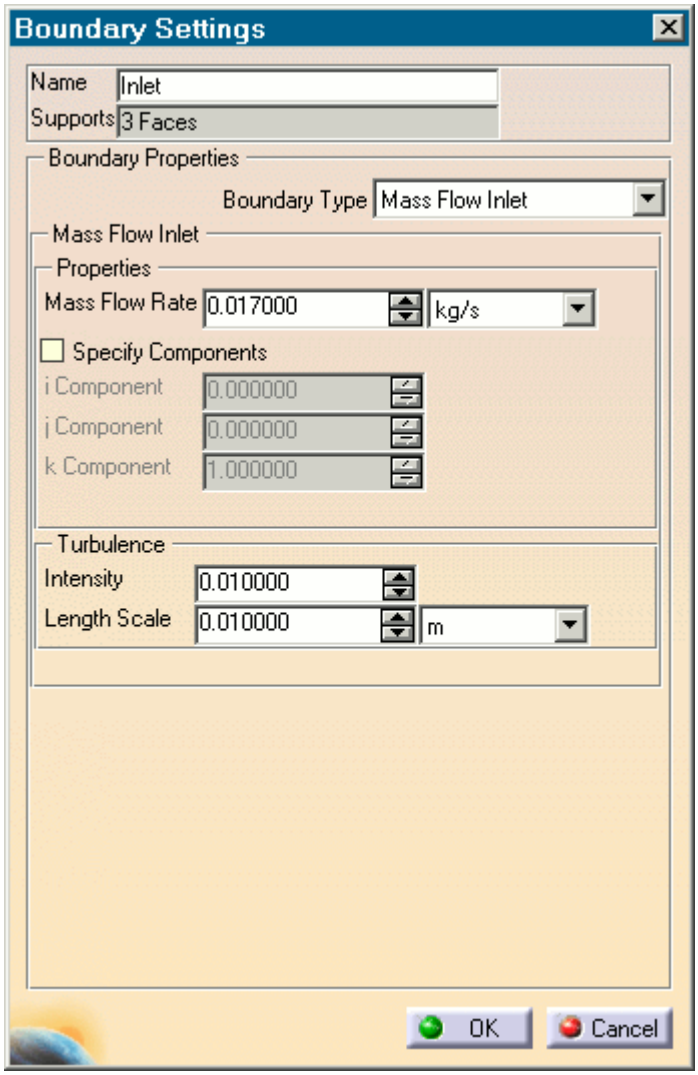
2. On the CAD model, select the three faces that are highlighted below.



These selections are reported in the *Supports box* in the *Boundary Settings* dialog.

3. Set *Boundary Type* to **Mass Flow Inlet**.
4. Set *Name* to `Inlet`.
5. Set *Mass Flow Rate* to `0.017` kg/s.

   The completed dialog is shown below.



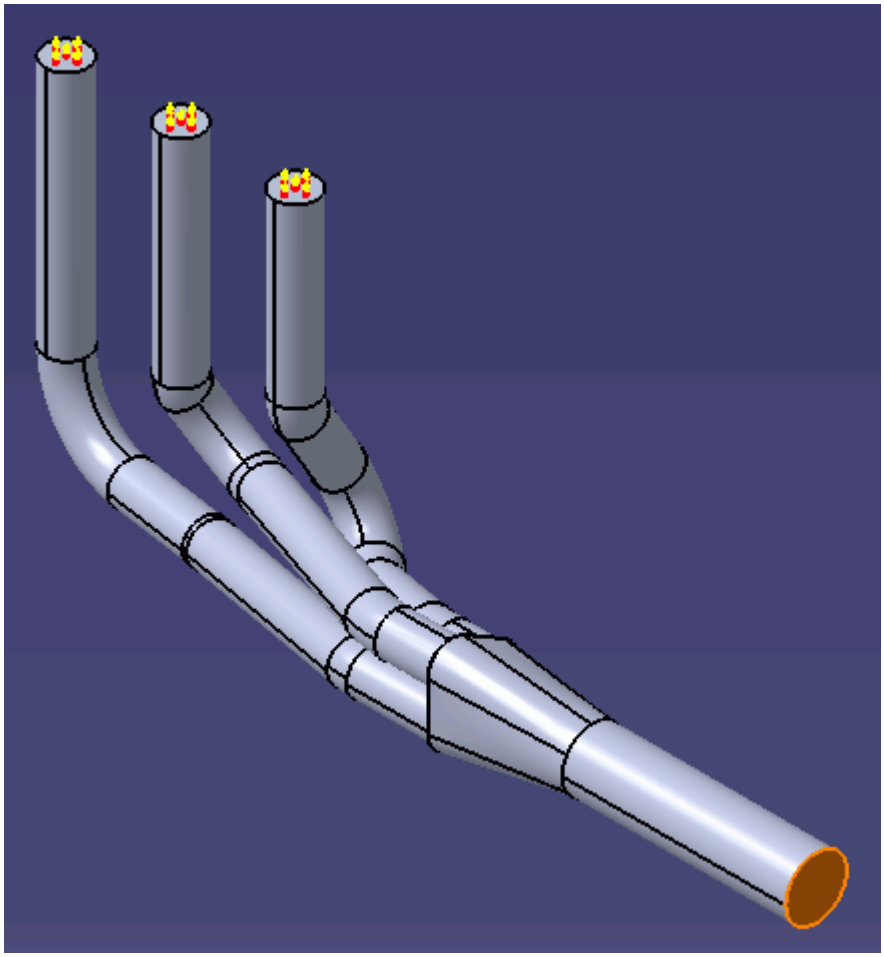
6. Click **OK**.

Create the outlet boundary:

7. Right-click the **Region** > **Boundaries** node and select **Add Boundary**.

   The *Boundary Settings* dialog appears.

8. Select the face that is highlighted below.



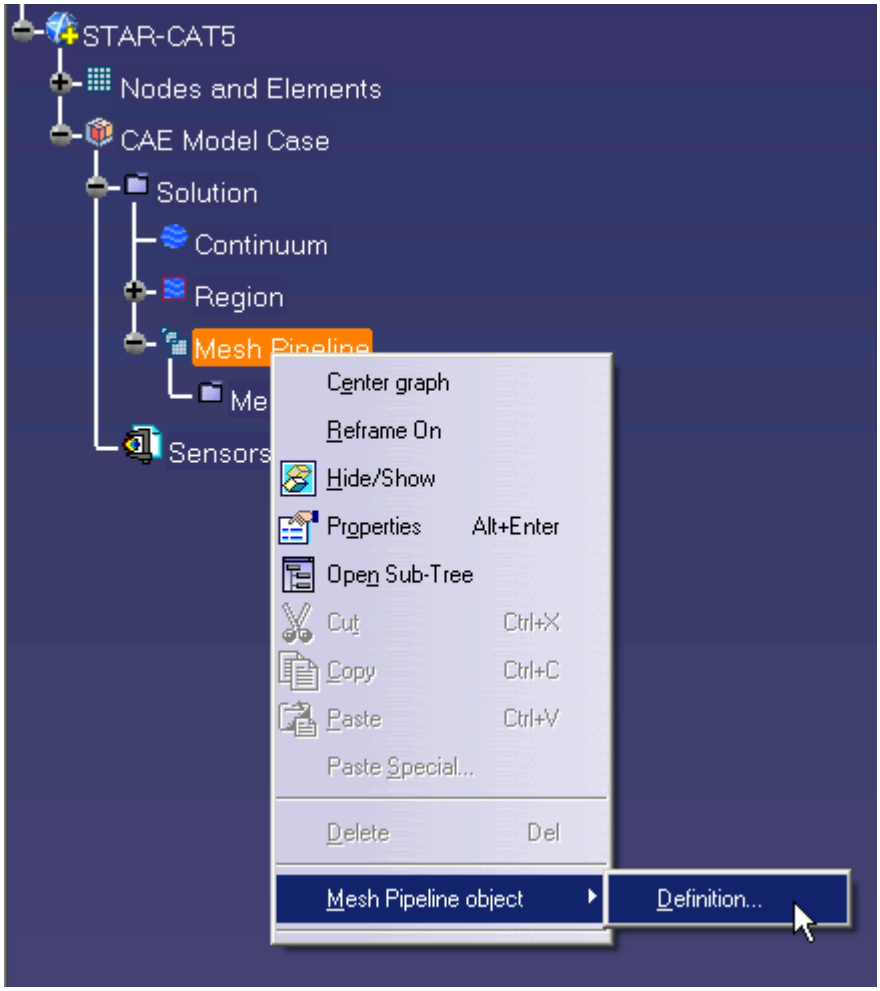
9. Set *Boundary Type* to **Pressure Outlet**.
10. Set *Name* to `Outlet`.

    The default settings are acceptable for this case.
11. Click **OK**.

Setting the Mesh Pipeline

To obtain more accurate results, modify the mesh pipeline:

1. Right-click the **Mesh Pipeline** node in the STAR-CAT5 tree and select **Mesh Pipeline object** > **Definition**.



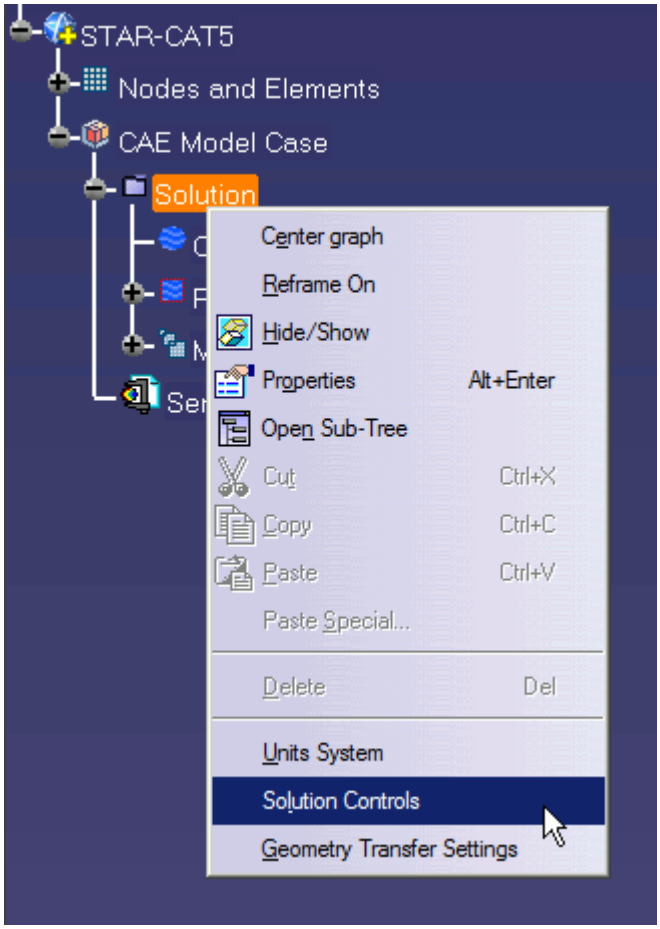
The *Mesh Pipeline Settings* dialog appears.

2. Set *Base Size* to `2.5`%.
3. Set *Prism Layer Thickness* to `15`%.
4. Click **OK**.

Modifying Solution Controls

Modify the maximum number of iterations and the solution accuracy:

1. Right-click the **Solution** node in the STAR-CAT5 object tree and select **Solution Controls**

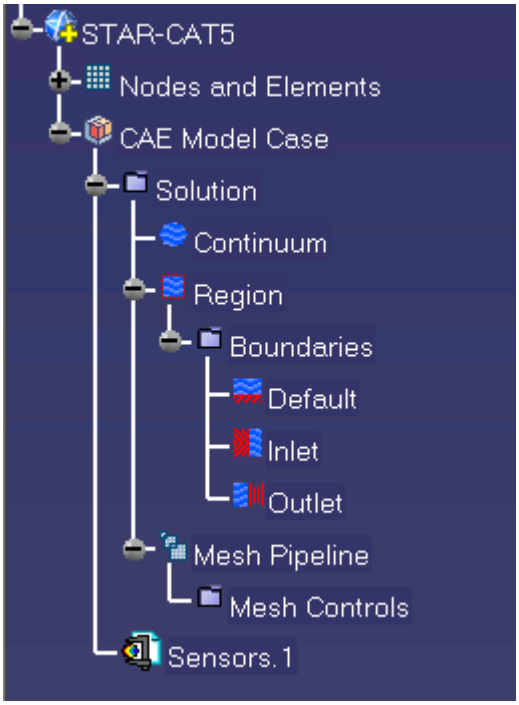


The *Solution Controls* dialog appears.

2. Click **More** to expand the *Advanced* tab.
3. Set *Iterations* to `200`.
4. Set *Convergence* to `0.0001`.
5. Click **OK**.
6. Save your progress.

Reviewing the Simulation Set Up

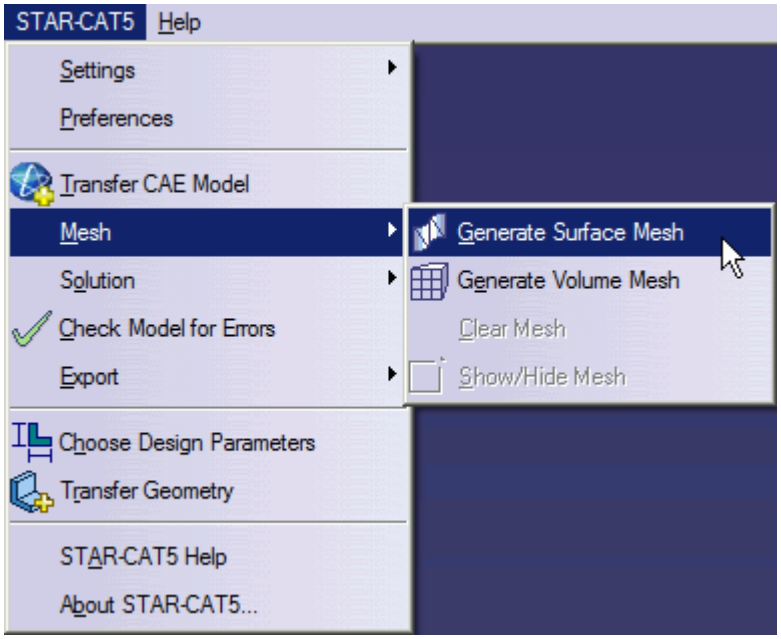
The final STAR-CAT5 object tree is shown below.



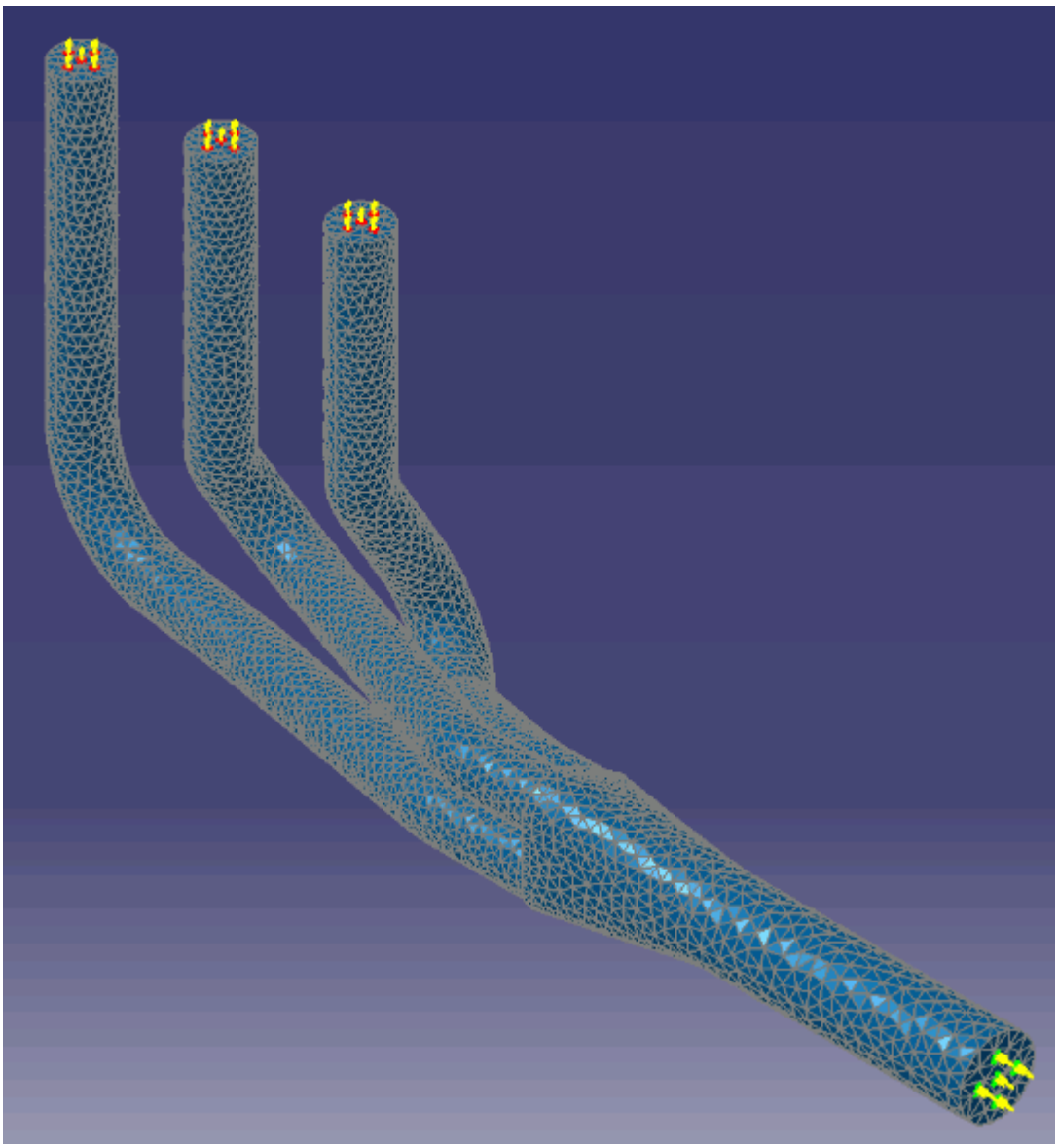
Generating the Surface Mesh

To generate the surface mesh:

1. Select **STAR-CAT5** > **Mesh** > **Generate Surface Mesh**.



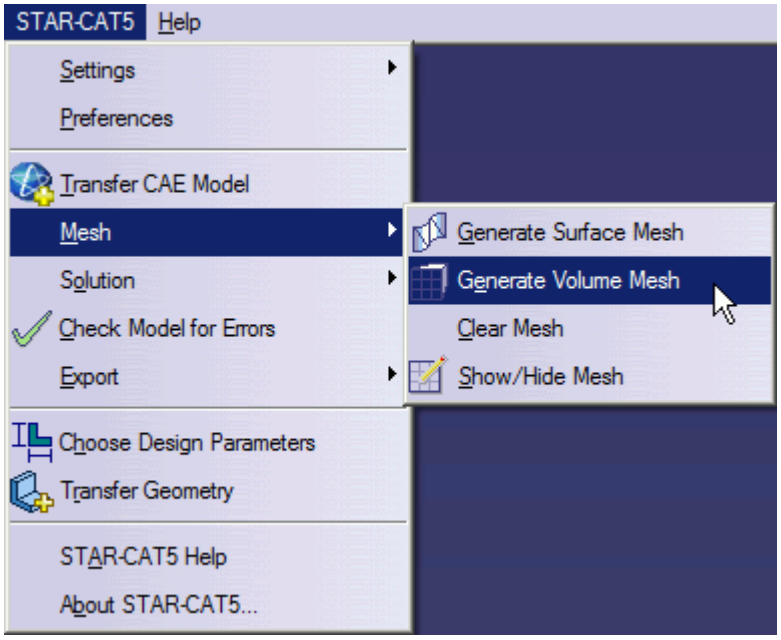
The surface mesh is shown below.



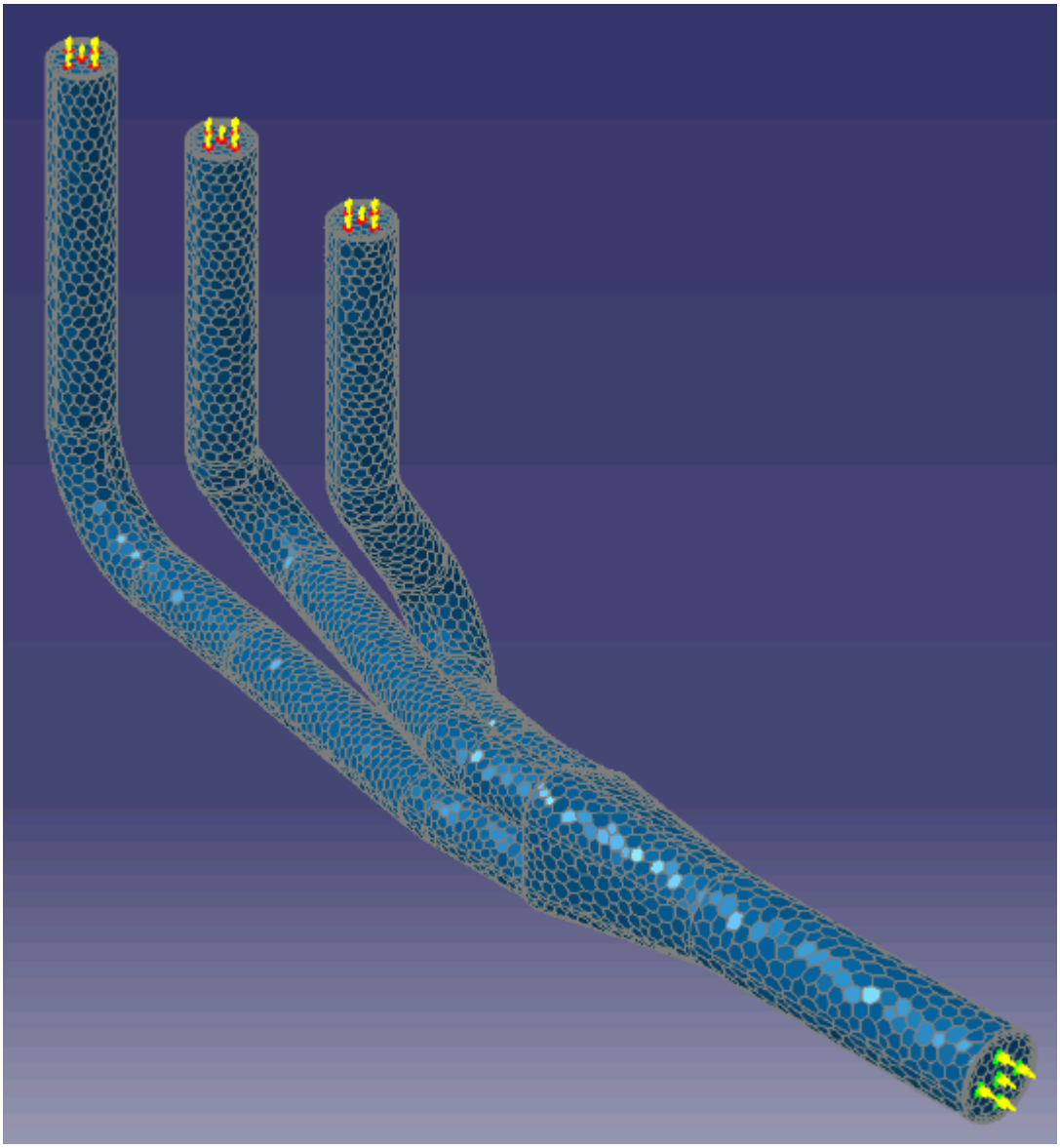
Generating the Volume Mesh

Generate the volume mesh:

1. Select **STAR-CAT5** > **Mesh** > **Generate Volume Mesh**.



The volume mesh is shown below.

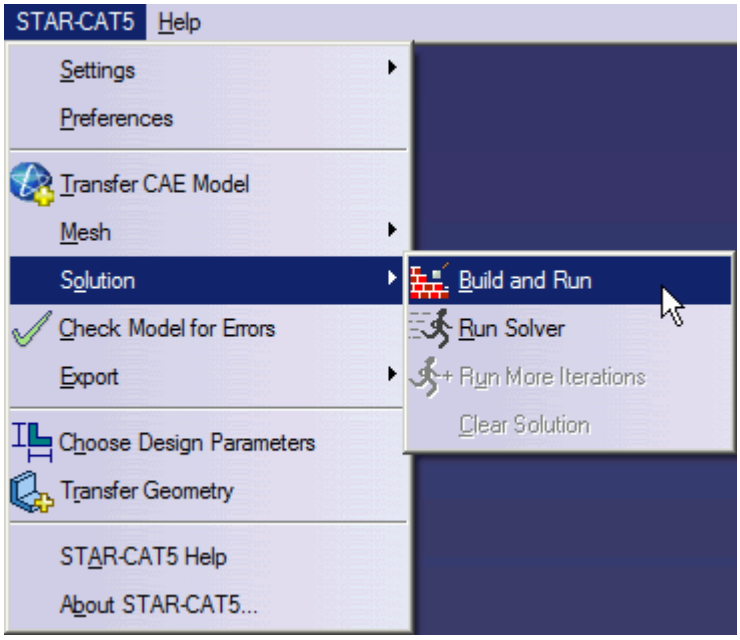


Building and Running

The case is now ready to run.

To run the simulation:

1. Select **STAR-CAT5** > **Solution** > **Build and Run**.



The *Residuals* plot appears in the STAR-CAT5 *Output* window. You can use this plot to monitor the convergence of the solution.

When the solver is done, STAR-CCM+ launches.

Post-Processing Results

You perform all post-processing in STAR-CCM+. By default, a scalar scene, *Scalar Scene 1*, automatically appears in the *Graphics* window when the solver finishes. This scene displays the absolute total pressure.

The results that are contained in the following sections are representative. You may see some variation in the results that you obtain, depending on what platform and package you are using.

1. In STAR-CCM+, click **File** > **Save** to save the simulation.

Contents:

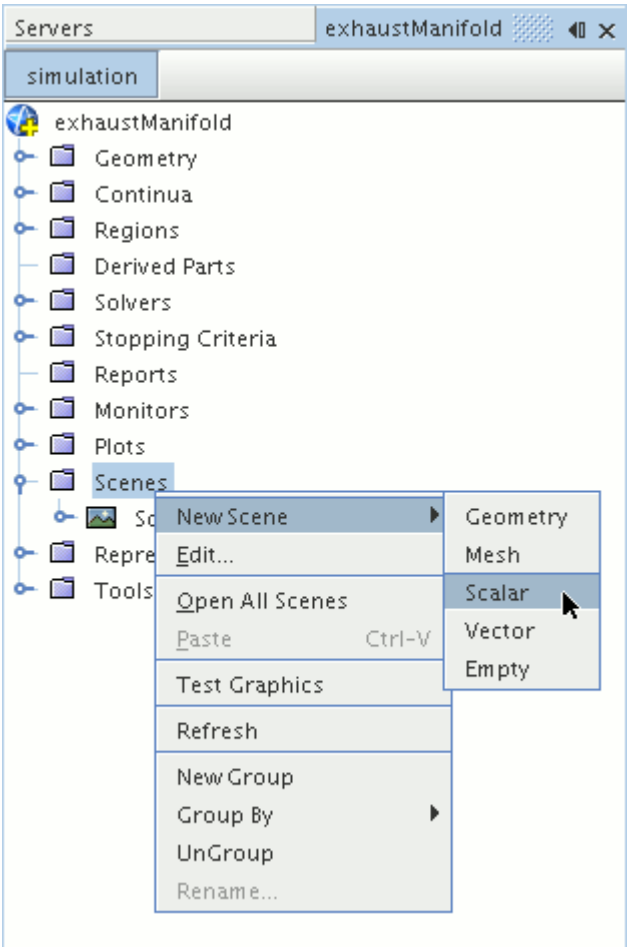
Creating a Scalar Scene
Creating A Streamlines Scene
Creating a Vector Scene
Creating a Streamlines Scene
Creating A Scalar Scene
Visualizing the Wing Skin Temperature
Creating a Streamlines Scene
Creating A Scalar Scene
Creating a Streamlines Scene
Creating a Scalar Scene
Modifying The Scalar Scene
Creating a Streamlines Scene

Creating a Scalar Scene

Create a scalar scene that displays the pressure on a cross-section of the exhaust manifold.

To create the scalar scene:

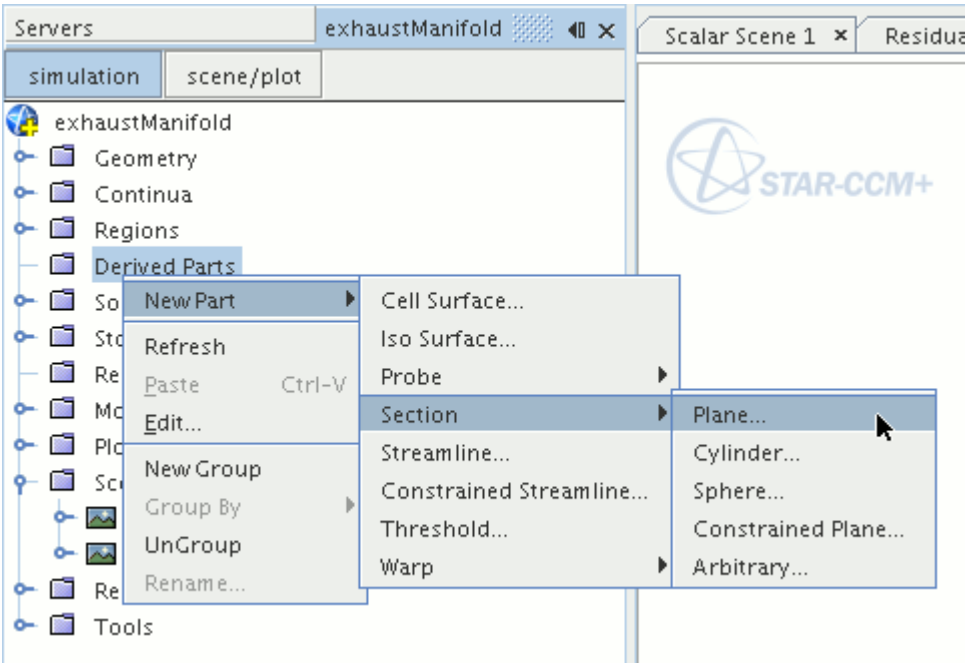
1. Right-click the **Scenes** node in the explorer pane and select **New Scene** > **Scalar**.



A new scalar scene, *Scalar Scene 2*, is created and selected in the *Graphics* window.

Create the section plane:

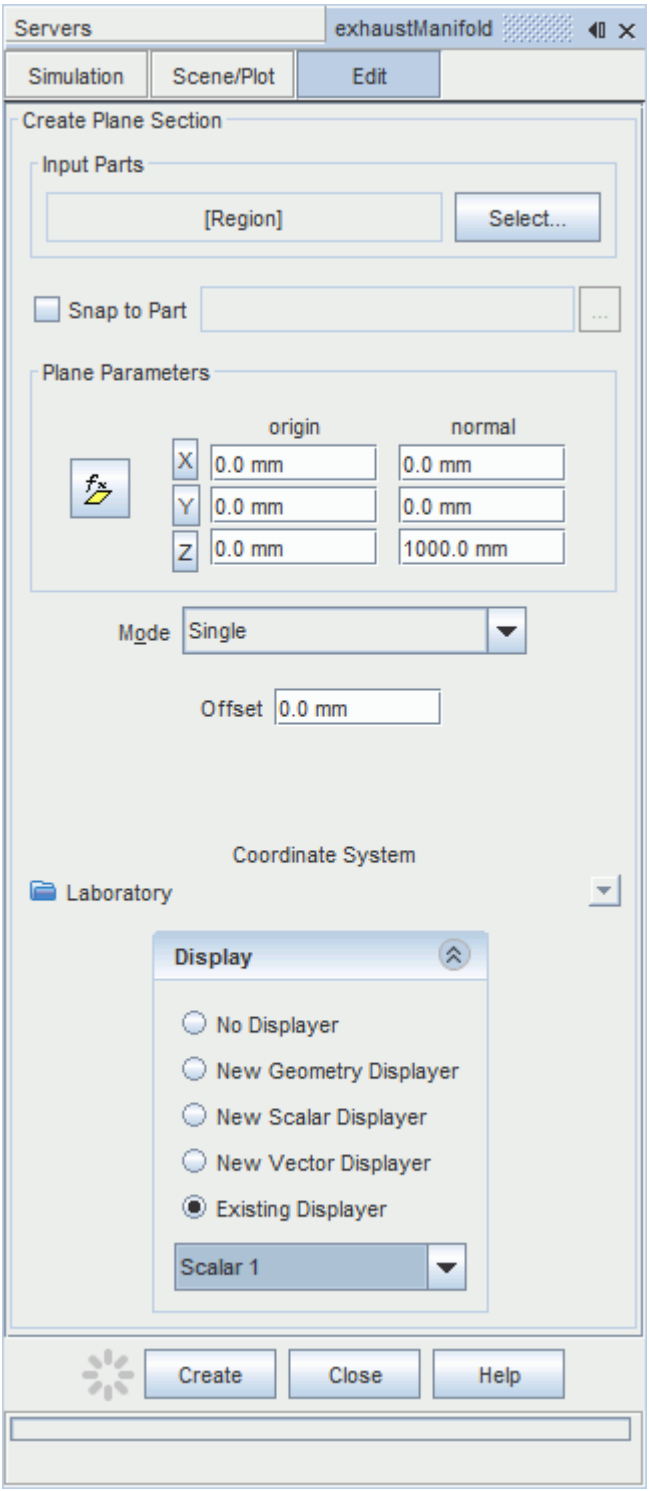
2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane...**



The *Create Plane Section* dialog appears.

3. In the *Plane Parameters* box, set the *origin* vector to `[0 mm, 0 mm, 0 mm]`
4. Set the *normal* vector to `[0 mm, 0 mm, 1000 mm]`
5. In the *Display* group box, select **Existing Displayer** and in the drop-down menu select **Scalar 1**.

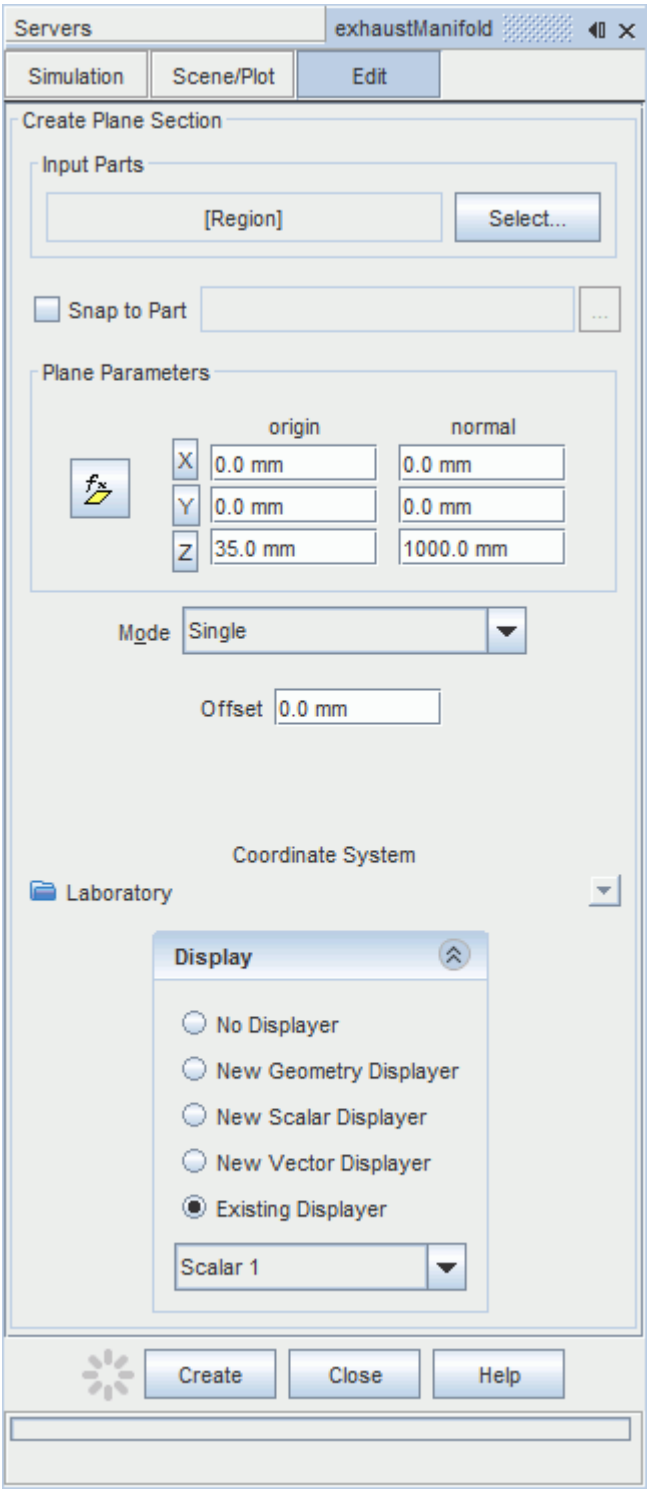
   The completed dialog is shown below.



6. Click **Create**.

Create the second section plane using the same dialog:

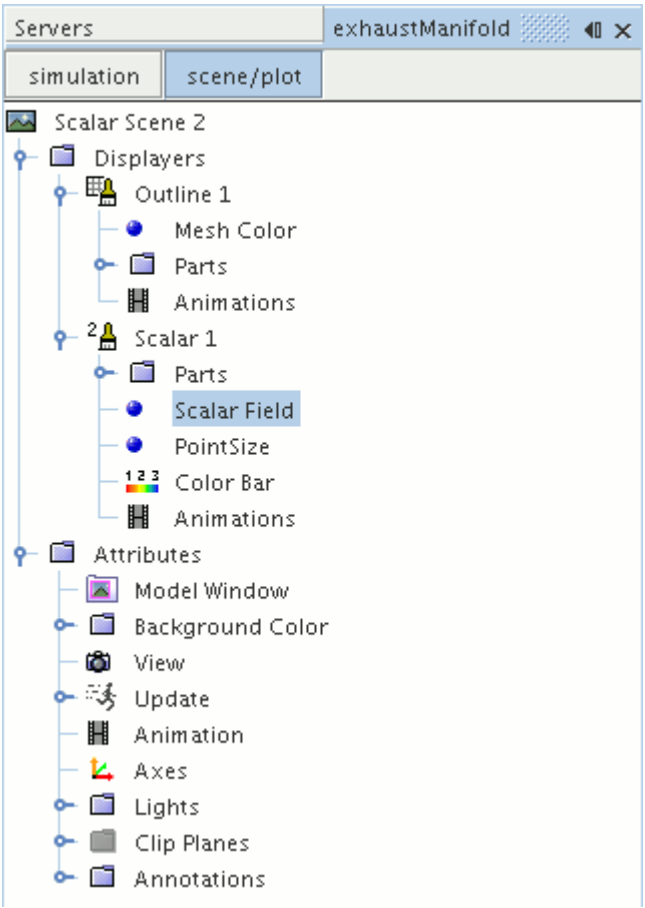
7. In the *Plane Parameters* box, set *origin* to `[0 mm, 0 mm, 35.0 mm]` and leave *normal* as `[0 mm, 0 mm, 1000 mm]`.



8. Click **Create** and then **Close**.

Define the scalar field to display the pressure contours:

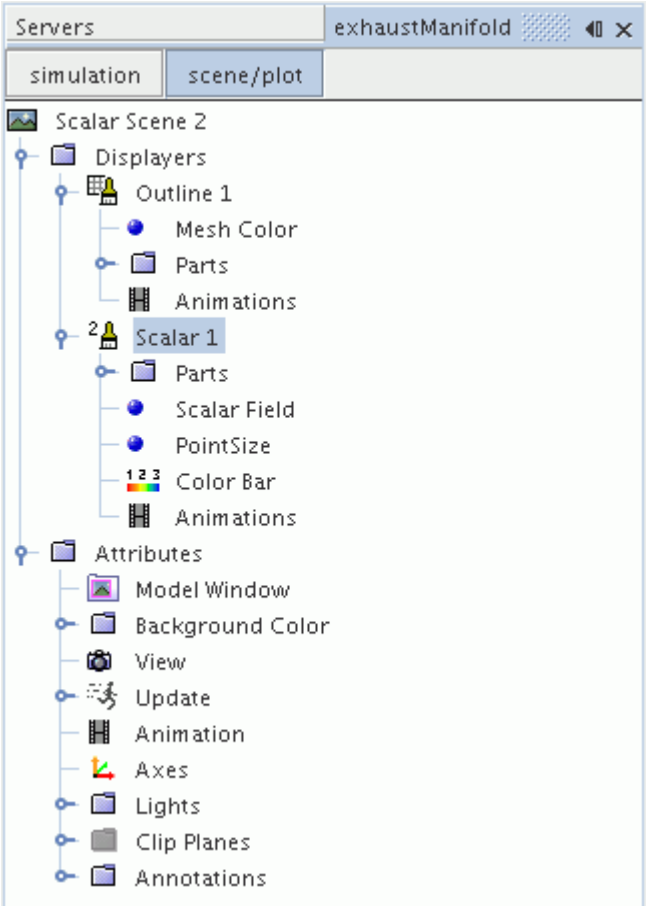
9. Click the **scene/plot** button in the explorer pane.
10. Select the **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Scalar Field** node.



11. Set *Function* to **Pressure**.

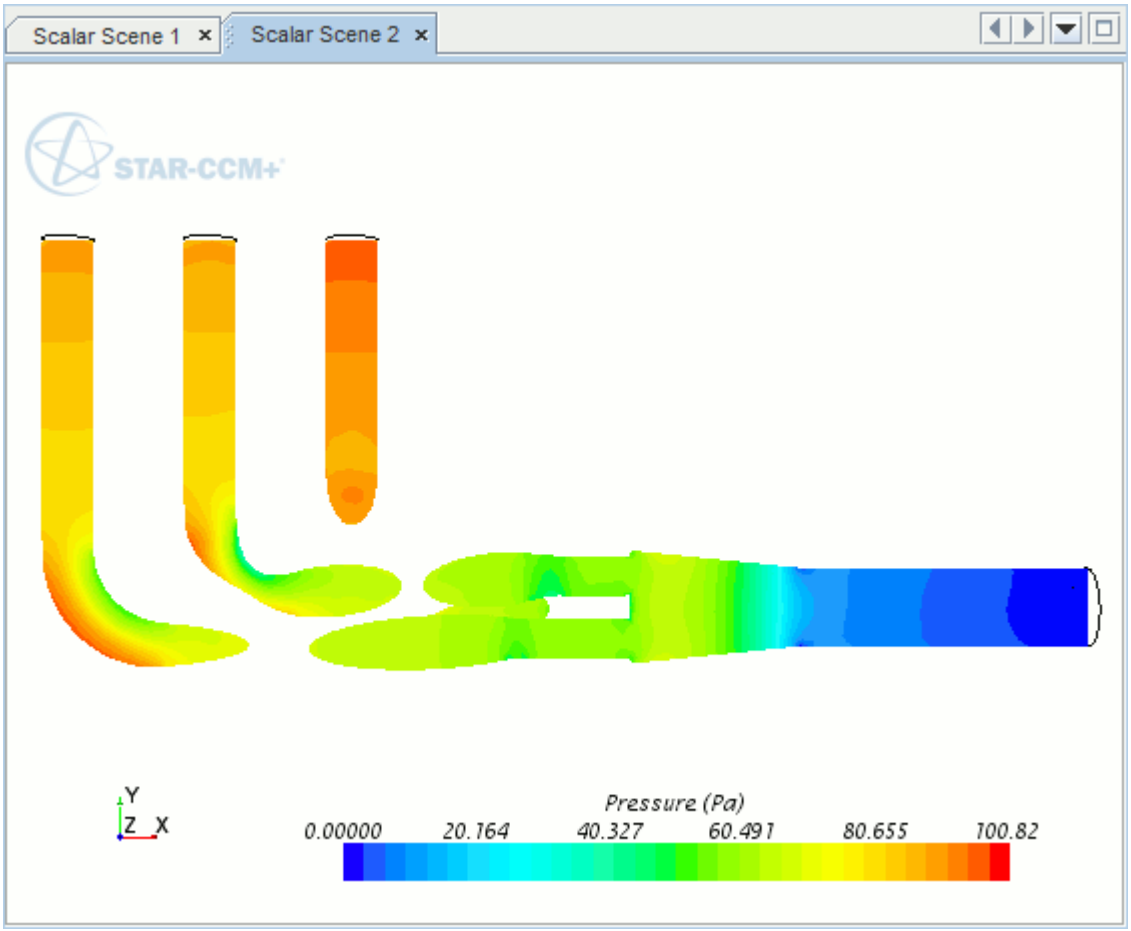
The pressure plot shows the pressure drop inside the exhaust manifold. In order to get a smooth transition between the different pressure values, set the contour style to smooth:

12. Select the **Scalar Scene 2** > **Displayers** > **Scalar 1** node.



13. Set *Contour Style* to **Smooth Filled**.
14. Make sure that the geometry is positioned on its side by pressing the <T> key while *Scalar Scene 2* is active.
15. Click the **simulation** button to return to the main STAR-CCM+ object tree.
16. Click (**Save**) to save the simulation .

The finished scene is shown below.

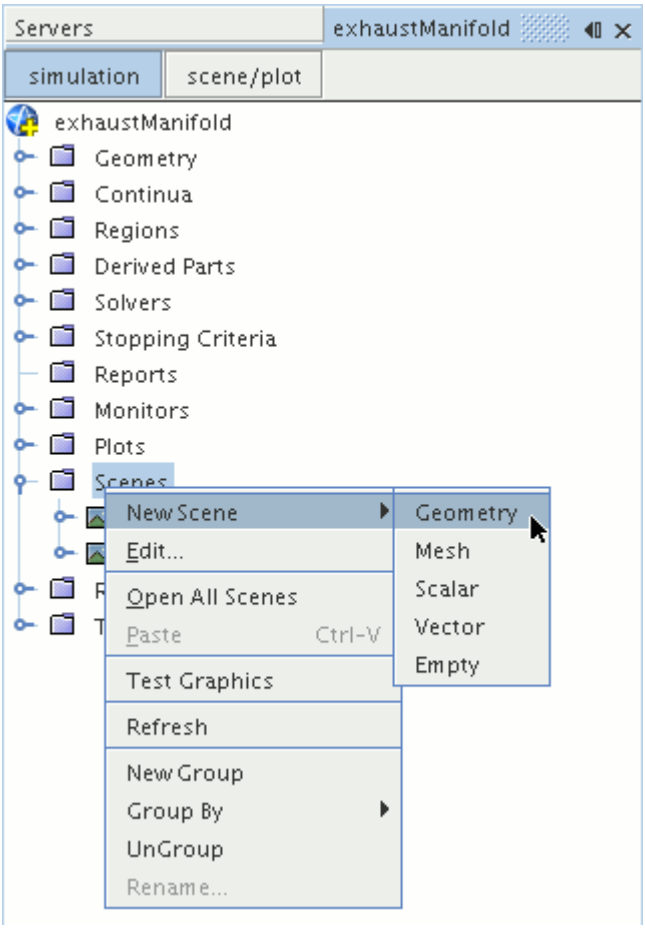


Creating A Streamlines Scene

Create a streamlines scene to get a better understanding of the flow inside the exhaust manifold.

First, create a geometry scene:

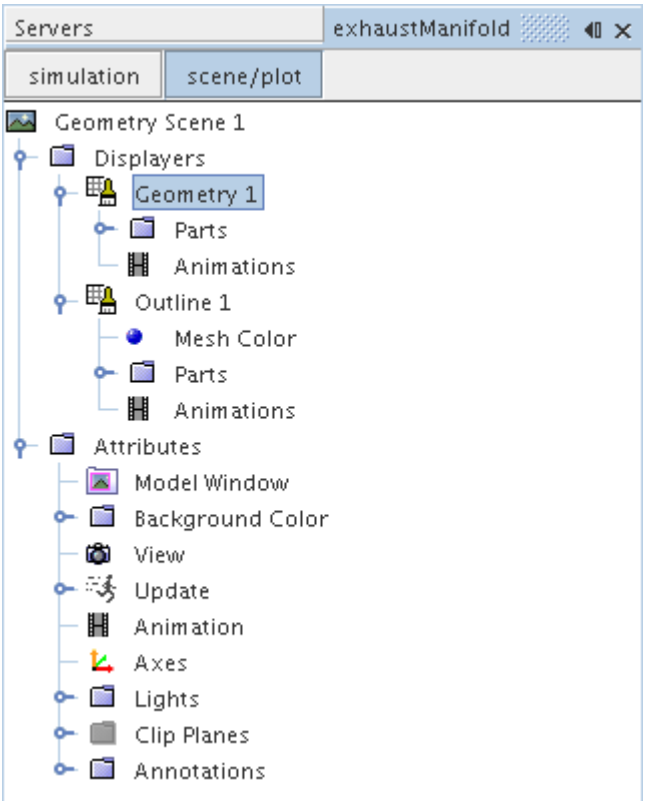
1. Right-click the **Scenes** node in the object tree and select **New Scene** > **Geometry**.



A new geometry scene is created and selected in the *Graphics* window.

Next, modify the opacity of the geometry model in the scene:

2. Click **scene/plot**.
3. Select the **Geometry Scene 1** > **Displayers** > **Geometry 1** node.



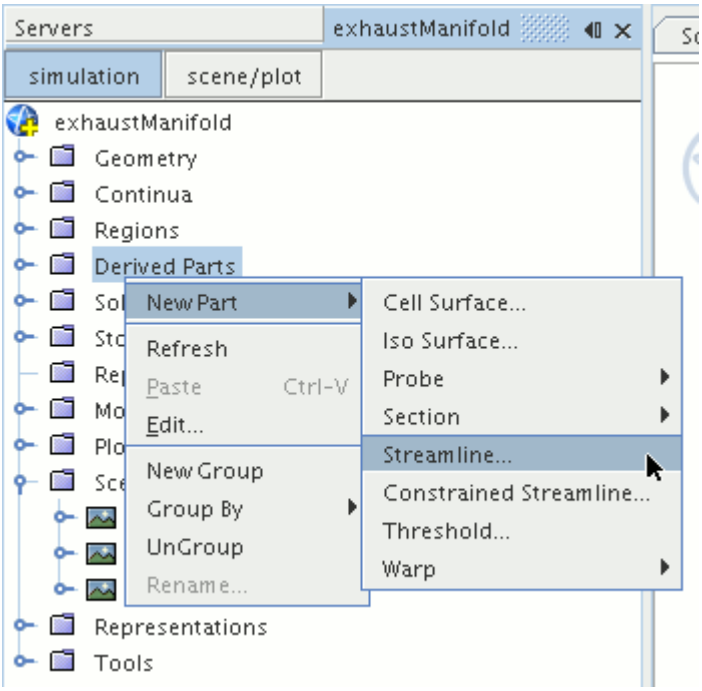
4. Set *Opacity* to `0.3`
5. Click **simulation**.

Change the view of the scene to something more suitable:

6. On the toolbar, click (**Save-Restore-Select views**) and select **Views** > **+X +Y +Z** > **Up +Y**.

Create the streamlines:

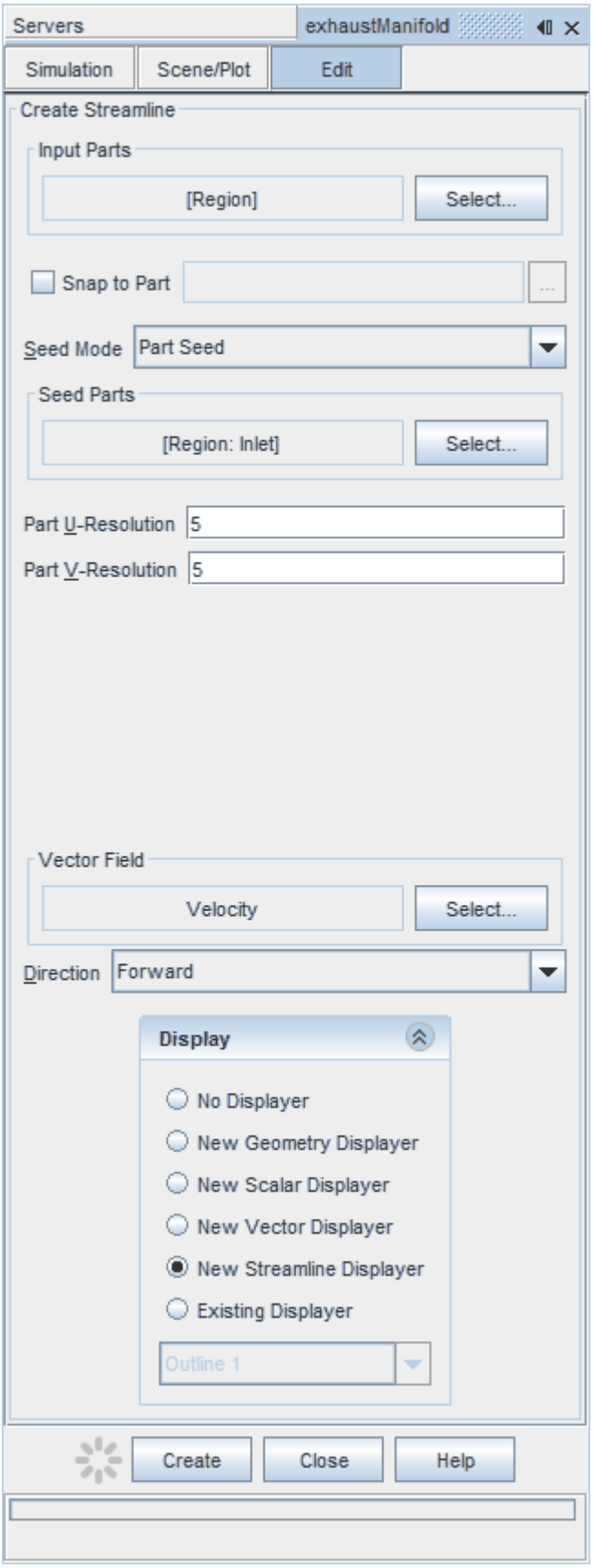
7. Right-click the **Derived Parts** node and select **New Part** > **Streamline...**.



The *Create Streamline* dialog appears.

8. Make sure that **Part Seed** is selected in the *Seed Mode* section.
9. Set *Seed Parts* to **Regions** > **Region** > **Boundaries** > **Inlet**.
10. Make sure that **New Streamline Displayer** is selected in the *Display* group box.

    The completed dialog is shown below.


Servers
exhaustManifold
Simulation
Scene/Plot
Edit
Create Streamline
Input Parts
[Region]
Select...
Snap to Part
Seed Mode
Part Seed
Seed Parts
[Region: Inlet]
Select...
Part U-Resolution
5
Part V-Resolution
5
Vector Field
Velocity
Select...
Direction
Forward
Display
No Displayer
New Geometry Displayer
New Scalar Displayer
New Vector Displayer
New Streamline Displayer
Existing Displayer
Outline 1
Create
Close
Help

11. Click **Create** to create the streamlines and add them to the geometry scene as part of a streamline displayer.
12. Click **Close** to close the *Create Streamline* dialog.

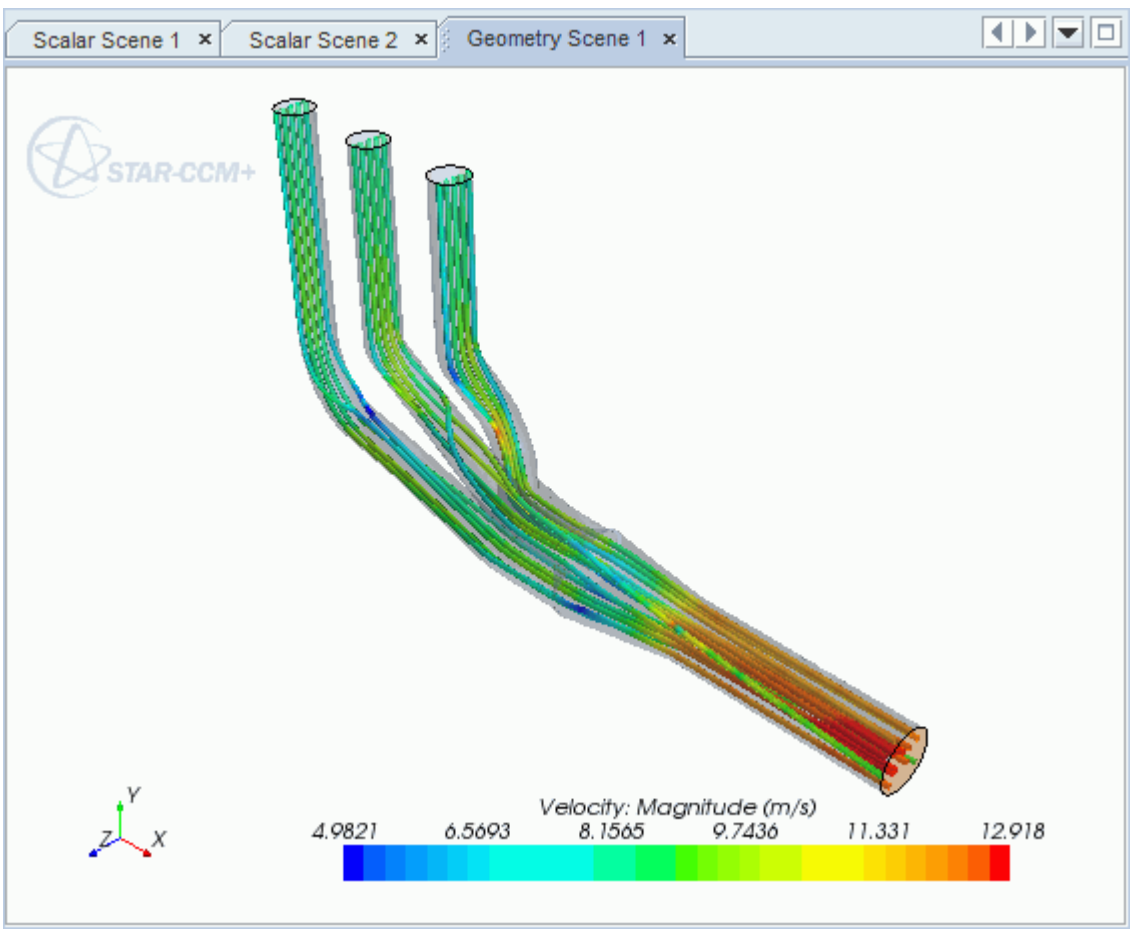
Change the display method of the streamlines to tubes instead of the default lines:

13. Click **scene/plot**.
14. Select the **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** node.
15. Set *Mode* to **Tubes** and *Width* to `0.0020`.

Change the color of the streamlines to represent velocity magnitude:

16. Select the **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** > **Scalar Field** node.
17. Set *Function* to **Velocity** > **Lab Reference Frame** > **Magnitude**.
18. Save the simulation.

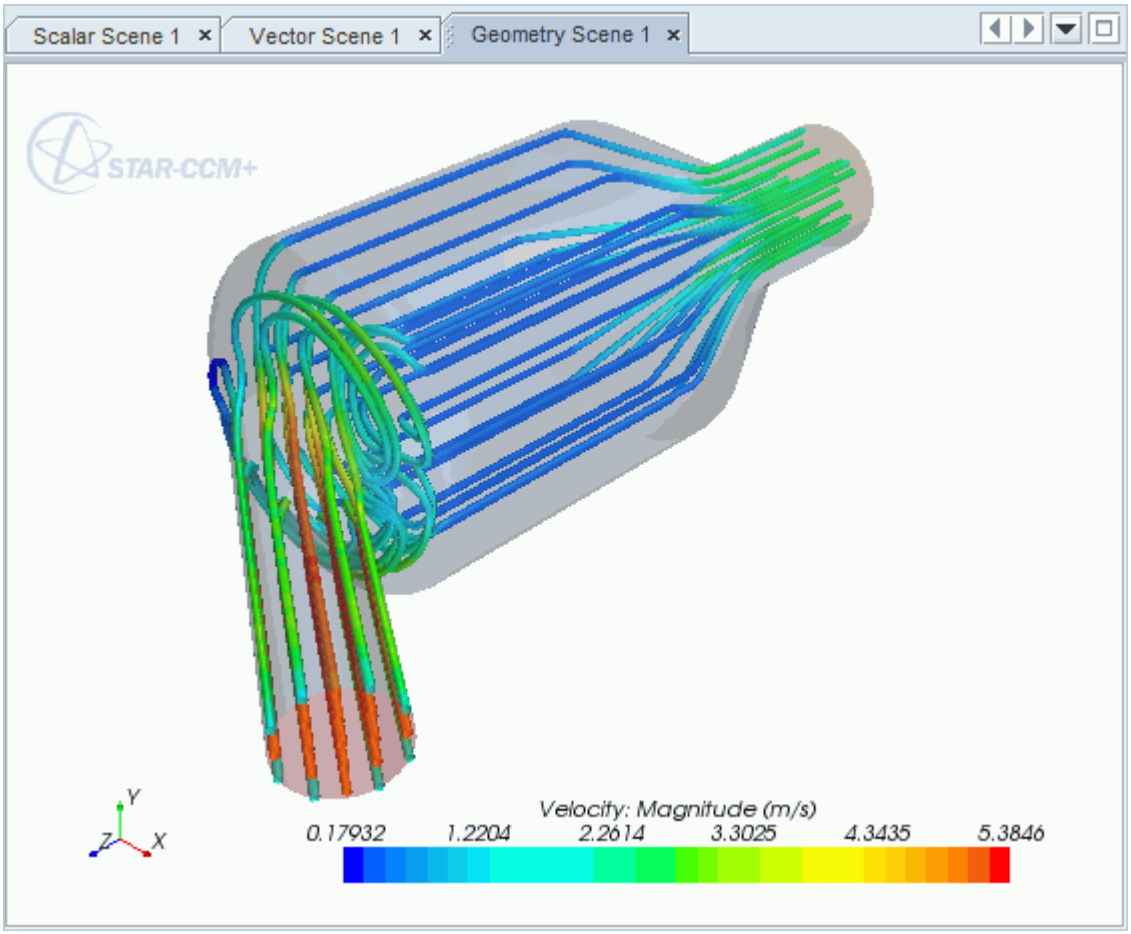
The finished scene is shown below.



Porous Media Flow

This tutorial takes you through the steps that are required to run a CFD calculation in the STAR-CAT5 environment.

In this tutorial, the model is an idealized catalyst.



Contents:

Opening the CAD Model
Selecting Fluid Physics Models
Setting the Porous Region
Defining Boundary Conditions
Setting the Mesh Pipeline
Modifying Solution Controls
Reviewing the Simulation Set Up
Generating the Surface Mesh
Generating the Volume Mesh

Building and Running
Post-Processing Results

Opening the CAD Model

A directory containing the CAD geometry is provided with the STAR-CCM+ distribution.

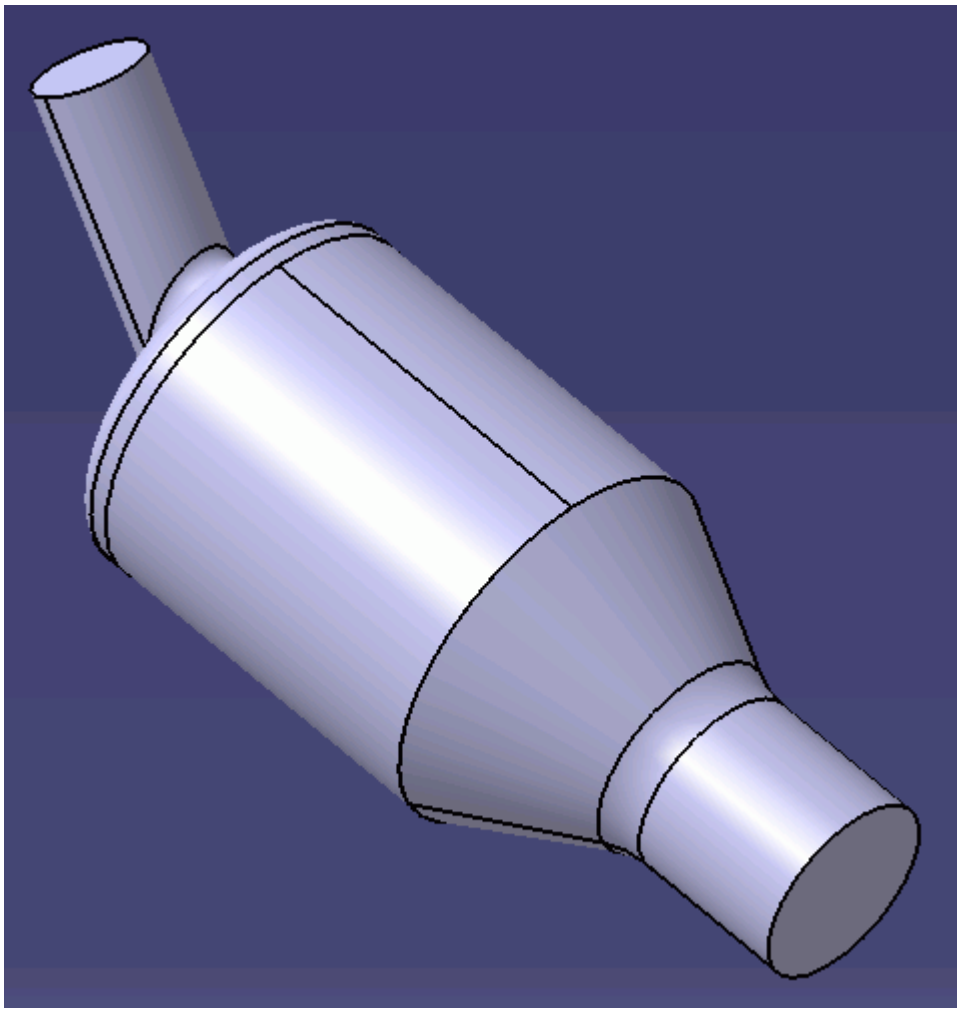
This directory is located in `[STAR-CCM+_INSTALL_DIR]\doc\startutorialsdata\starcat5\data`.

1. Copy directory `data` to your own disk space and open the part file `PorousModel.CATProduct` in STAR-CAT5.

   See Launching STAR-CAT5.

   The assembly is made up of three parts: the inlet region, the porous middle region, and the outlet region.

2. Position the geometry as shown below.



The STAR-CAT5 menu appears at the top of the screen.The STAR-CAT5 object tree is located on the left of the screen.

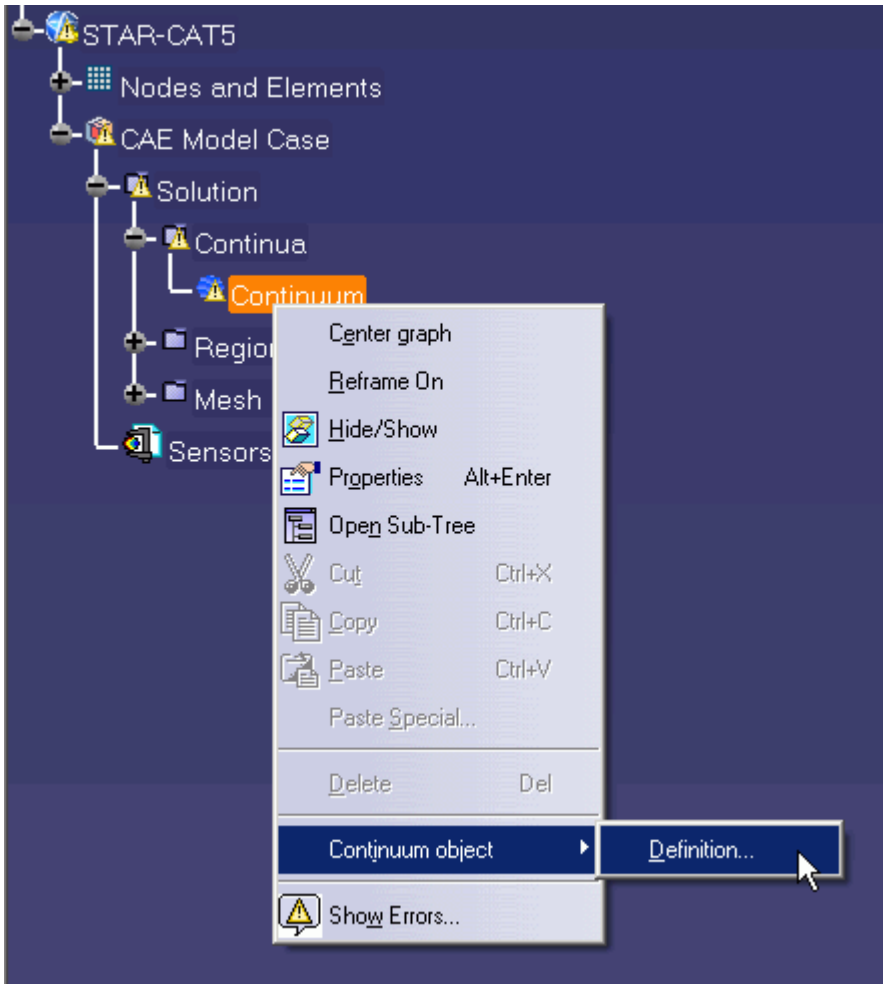
3. Save the analysis as `PorousMedia.CATAnalysis`.

Selecting Fluid Physics Models

The first step in setting up this tutorial is to define the physics of the problem. For this case, the flow is turbulent and isothermal. Temperature calculations are on by default in STAR-CAT5.

To turn temperature calculations off:

1. Right-click the **Continua** > **Continuum** node in the STAR-CAT5 tree and select **Continuum object** > **Definition...**.



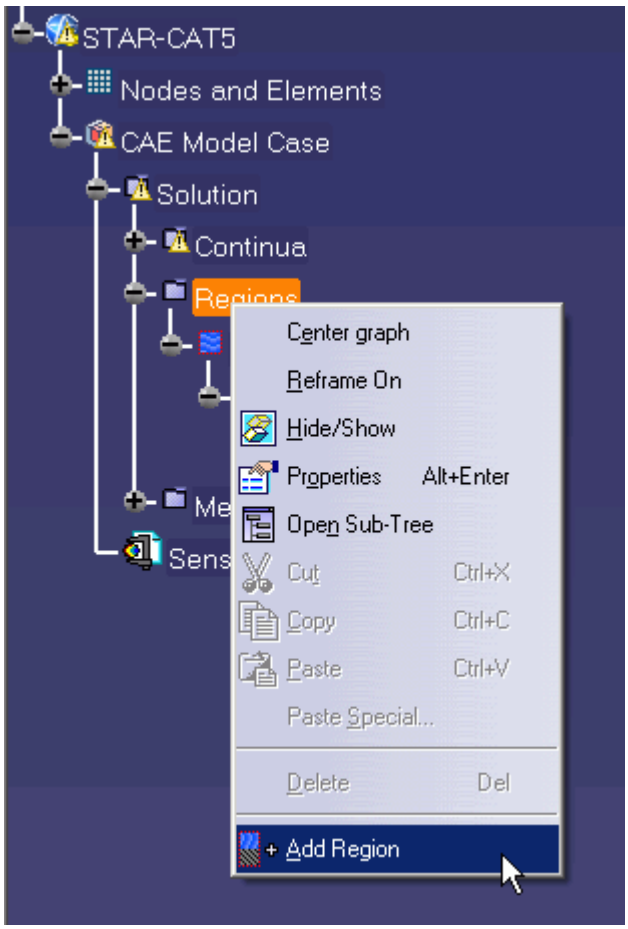
   The *Continuum Settings* dialog appears.

2. Deactivate the **Temperature** option in the *Optional Physics Models* box.
3. Click **OK**.

Setting the Porous Region

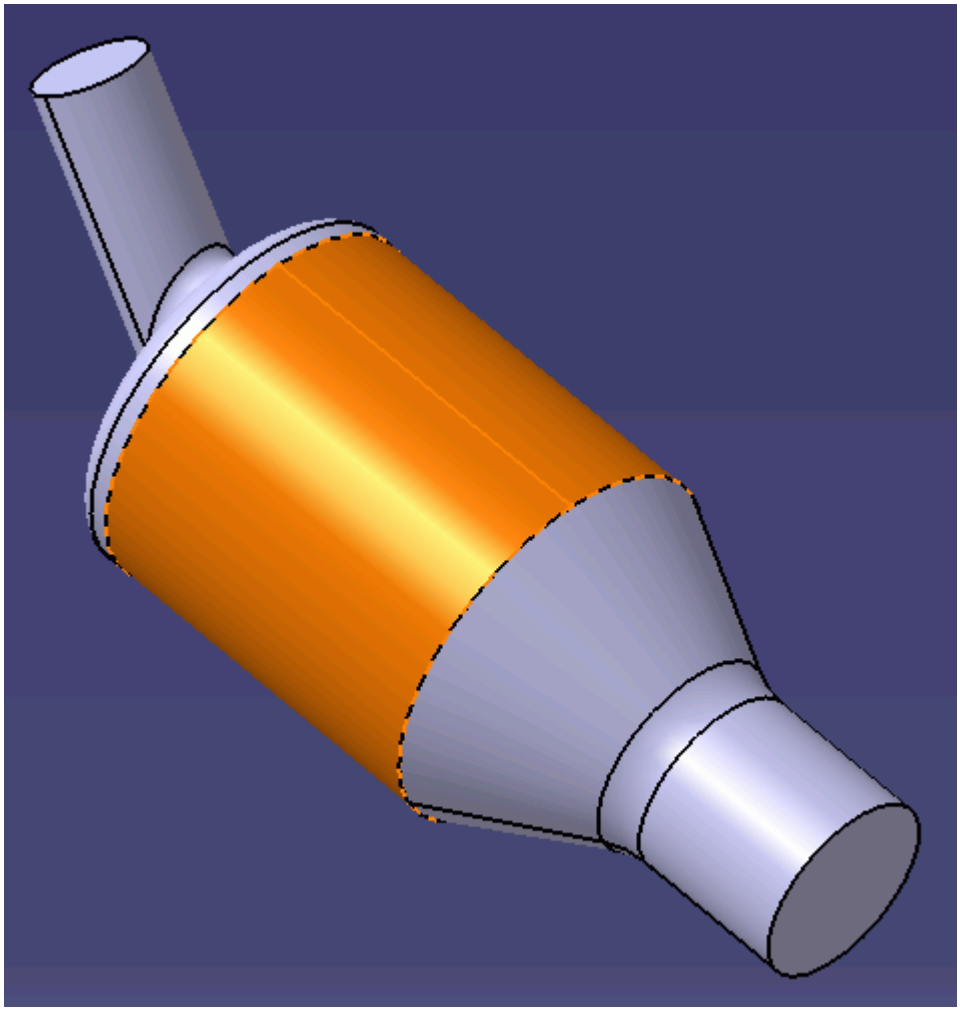
Create the porous region and define its properties:

1. Right-click the **Regions** node in the STAR-CAT5 object tree and select **Add Region**.



The *Region Settings* dialog appears.

2. Select the middle part of the assembly as shown below.



3. Set *Name* to `Porous Region`.
4. Activate the *Porous* model and select the **Uni-Directional** porous type.
5. Set the following properties for the *Porous* region:

Parameter	Value
Alpha	`25 kg/m^4`
Beta	`1500 kg/m^3.s`
Resistance	`250`

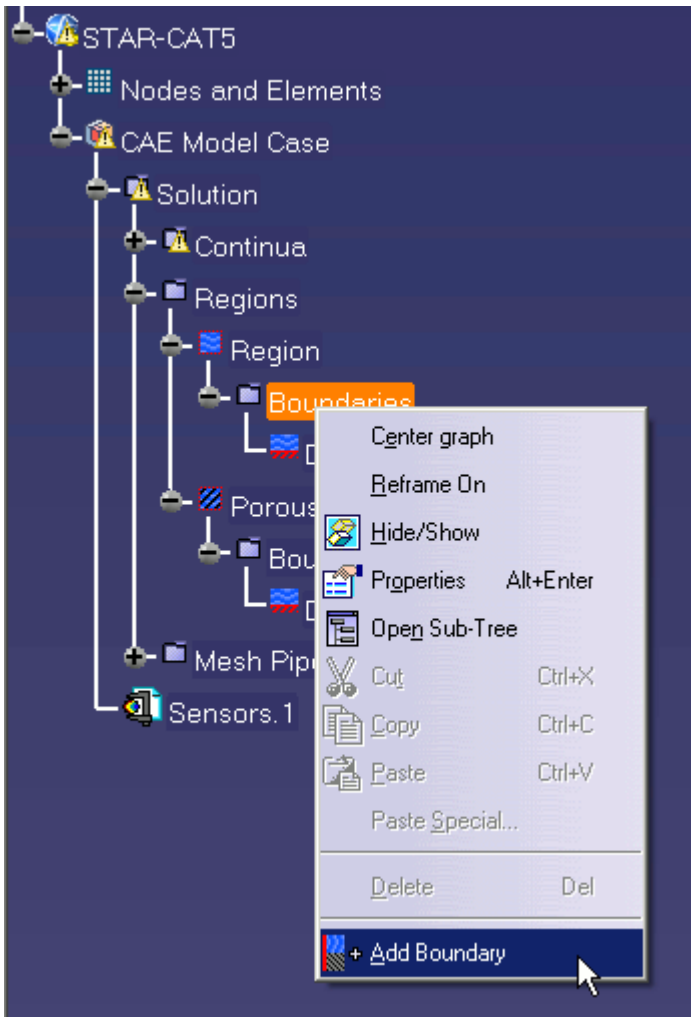
6. Click **OK**.

Defining Boundary Conditions

Create the inlet and outlet boundary conditions. STAR-CAT5 automatically adds interfaces between the different regions. These interfaces are visible in the final STAR-CCM+ simulation.

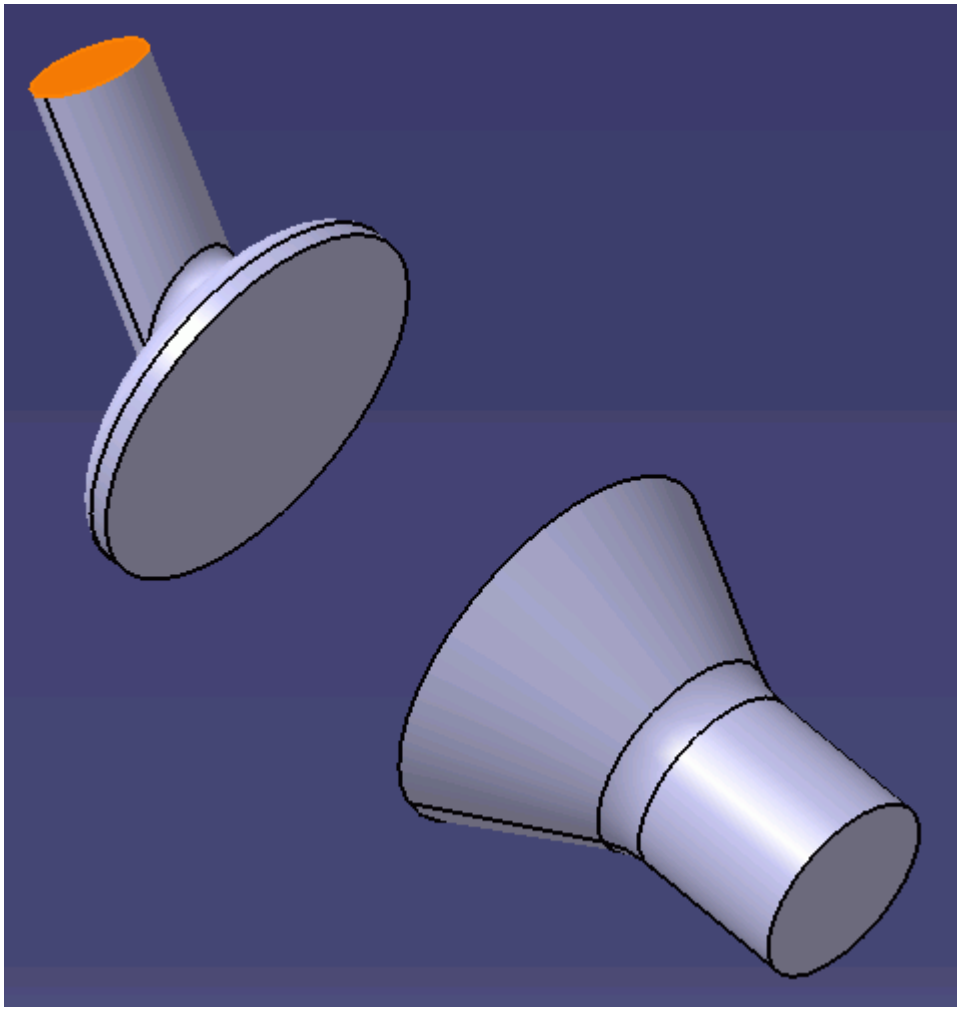
Create the inlet boundary first:

1. Right-click the **Regions** > **Region** > **Boundaries** node and select **Add Boundary**.



The *Boundary Settings* dialog appears.

2. Select the face that is highlighted below.



These selections are reported in the *Supports* box in the *Boundary Settings* dialog.

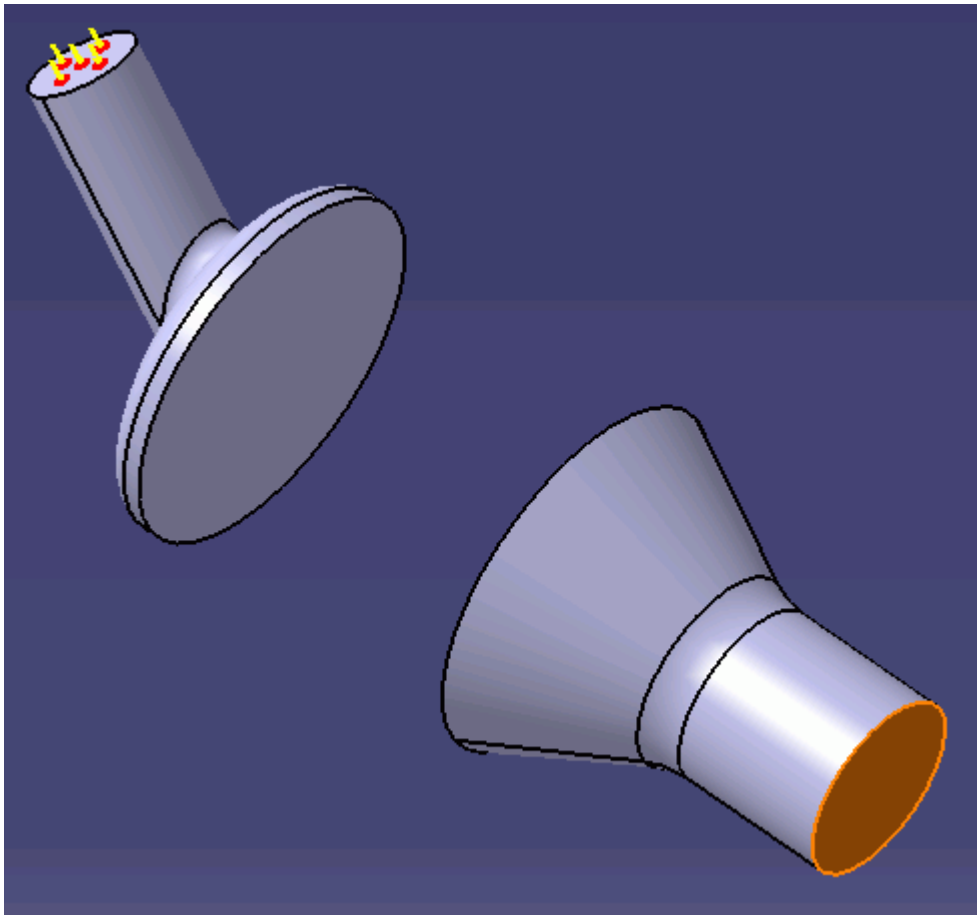
3. Set the following parameters:

Parameter	Value
Boundary Type	**Velocity Inlet**
Name	`Inlet`
Normal Velocity	`5 m/s`

4. Click **OK**.

Create the outlet boundary:

5. Using the same technique, create another boundary.
6. Select the face that is highlighted below.



7. Set the following parameters:

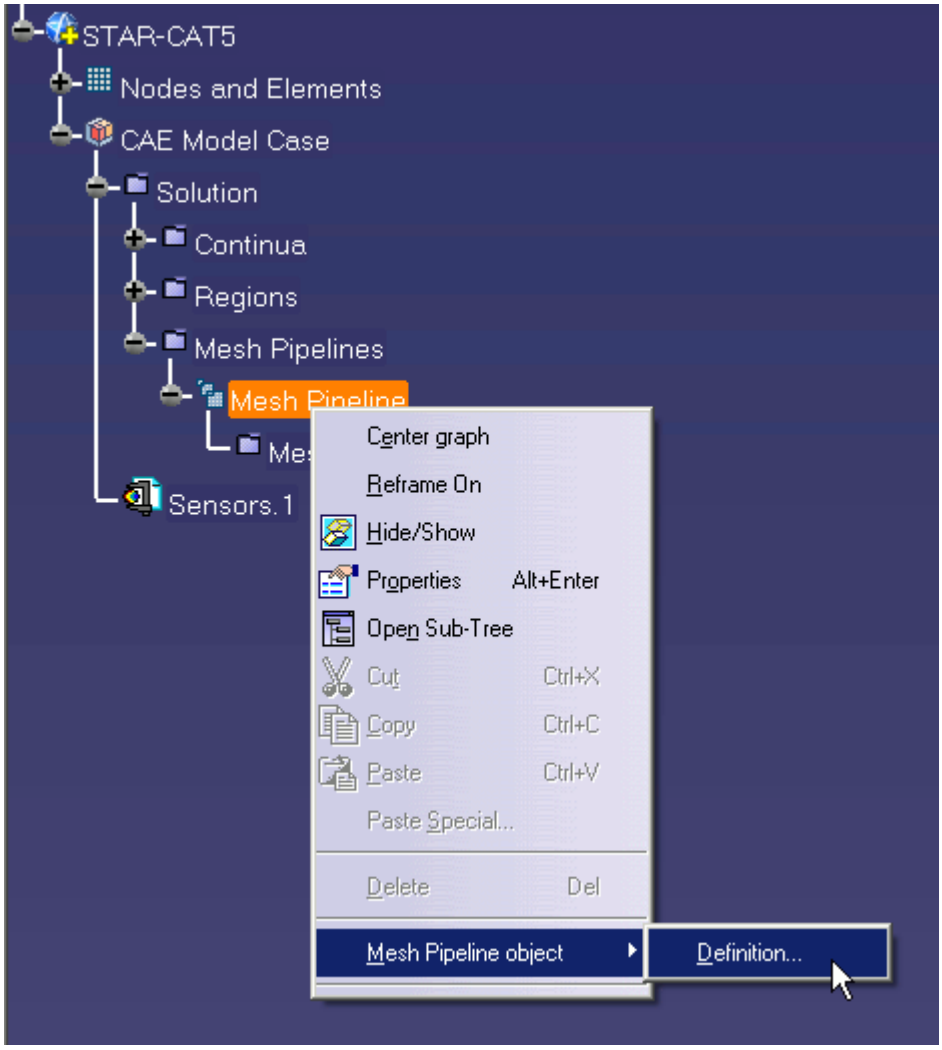
Parameter	Value
Boundary Type	**Pressure Outlet**
Name	`Outlet`

8. Click **OK**.

Setting the Mesh Pipeline

To obtain more accurate results, modify the mesh pipeline:

1. Right-click the **Mesh Pipelines** > **Mesh Pipeline** node in the STAR-CAT5 tree and select **Mesh Pipeline object** > **Definition**.



The *Mesh Pipeline Settings* dialog appears.

2. Set the following parameters:

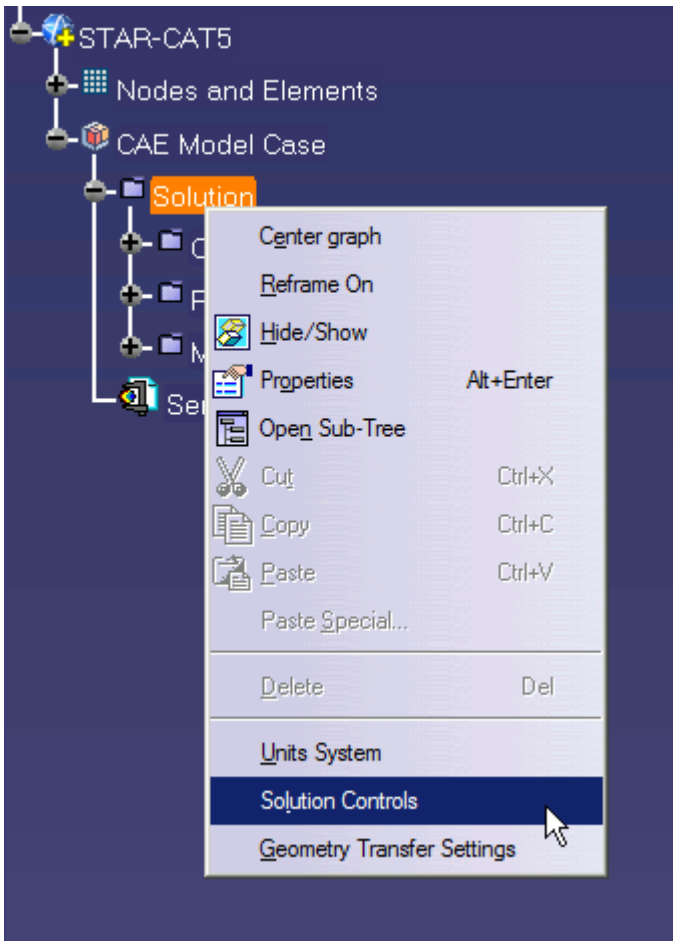
Parameter	Value
Base Size	
Use Absolute Value	**Activated**
Base Size	`0.0025 m`
Prism Layer	
User Absolute Value	**Activated**
Prism Layer Thickness	`0.001 m`

3. Click **OK**.

Modifying Solution Controls

Modify the maximum number of iterations and the solution accuracy settings.

1. Right-click the **Solution** node in the STAR-CAT5 object tree and select **Solution Controls**



The *Solution Controls* dialog appears.

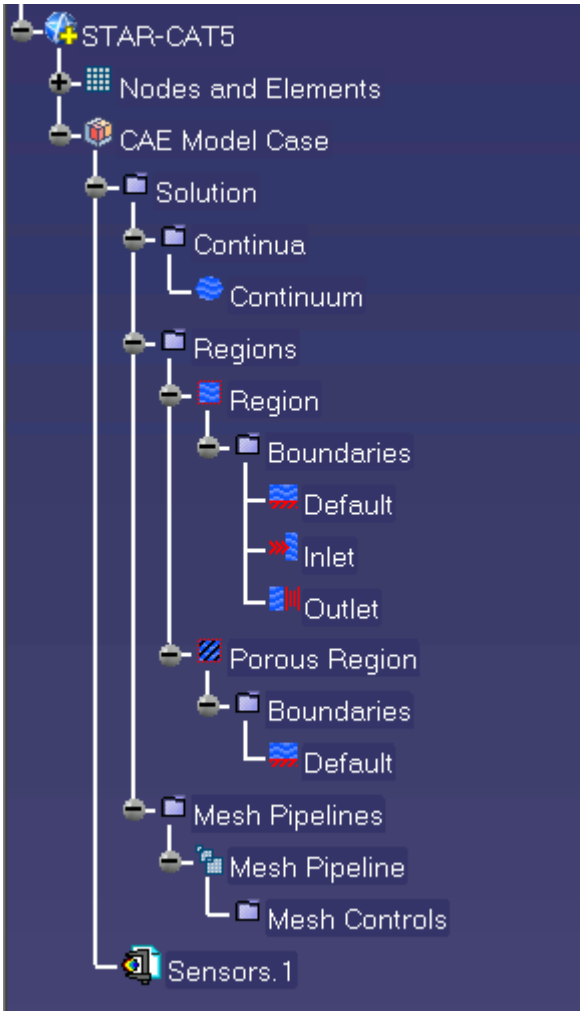
2. Click **More** to expand the *Solution Controls* menu.
3. Set the following parameters:

Parameter	Value
Iterations	400
Solution Accuracy	**First Order**

4. Click **OK**.
5. Save your progress.

Reviewing the Simulation Set Up

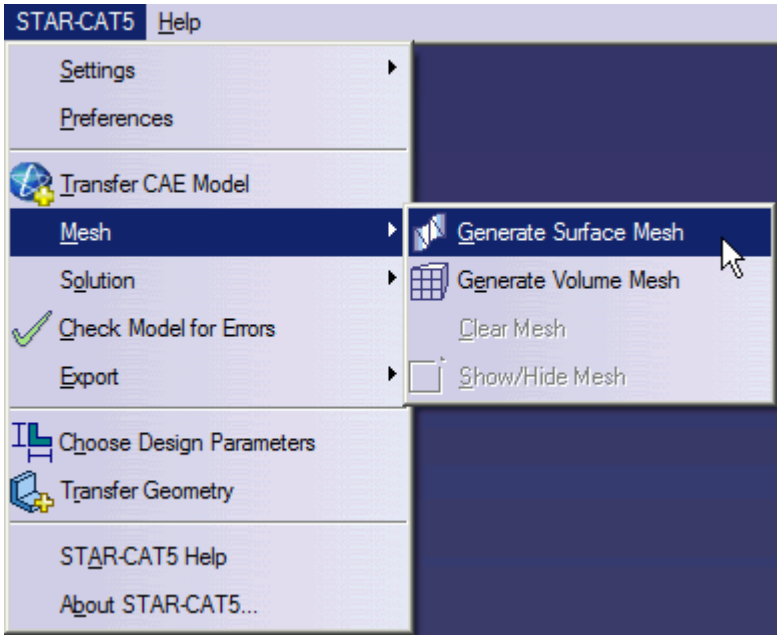
The final STAR-CAT5 object tree is shown below.



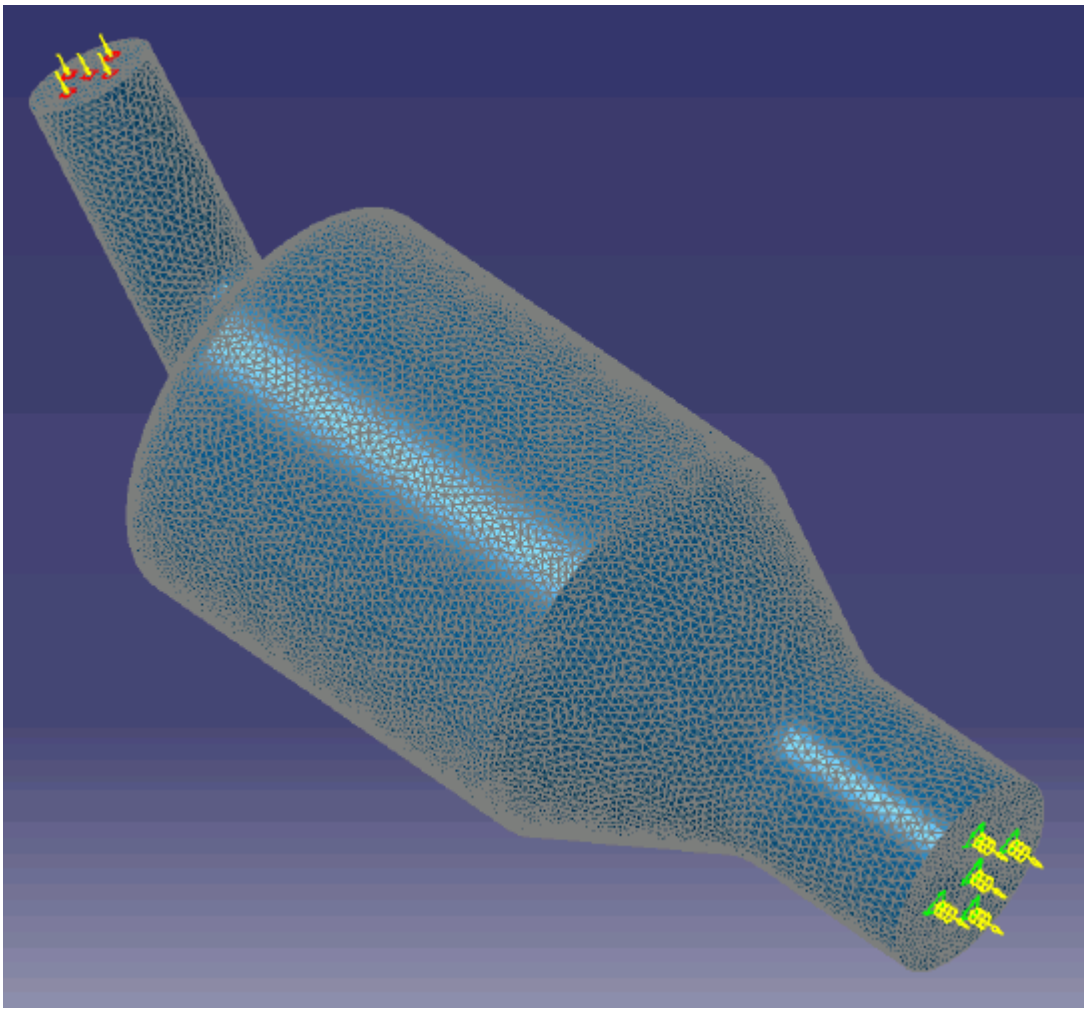
Generating the Surface Mesh

Generate the surface mesh:

1. Select **STAR-CAT5** > **Mesh** > **Generate Surface Mesh**.



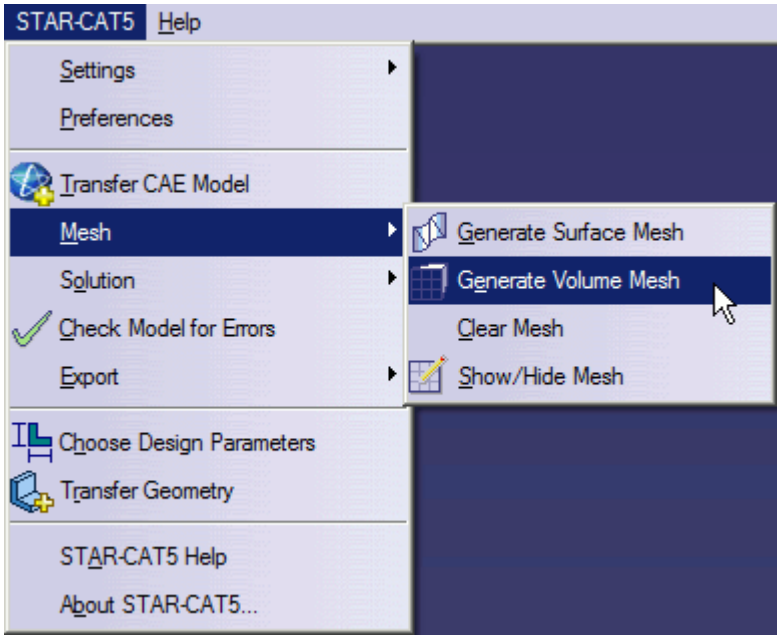
The surface mesh is shown below.



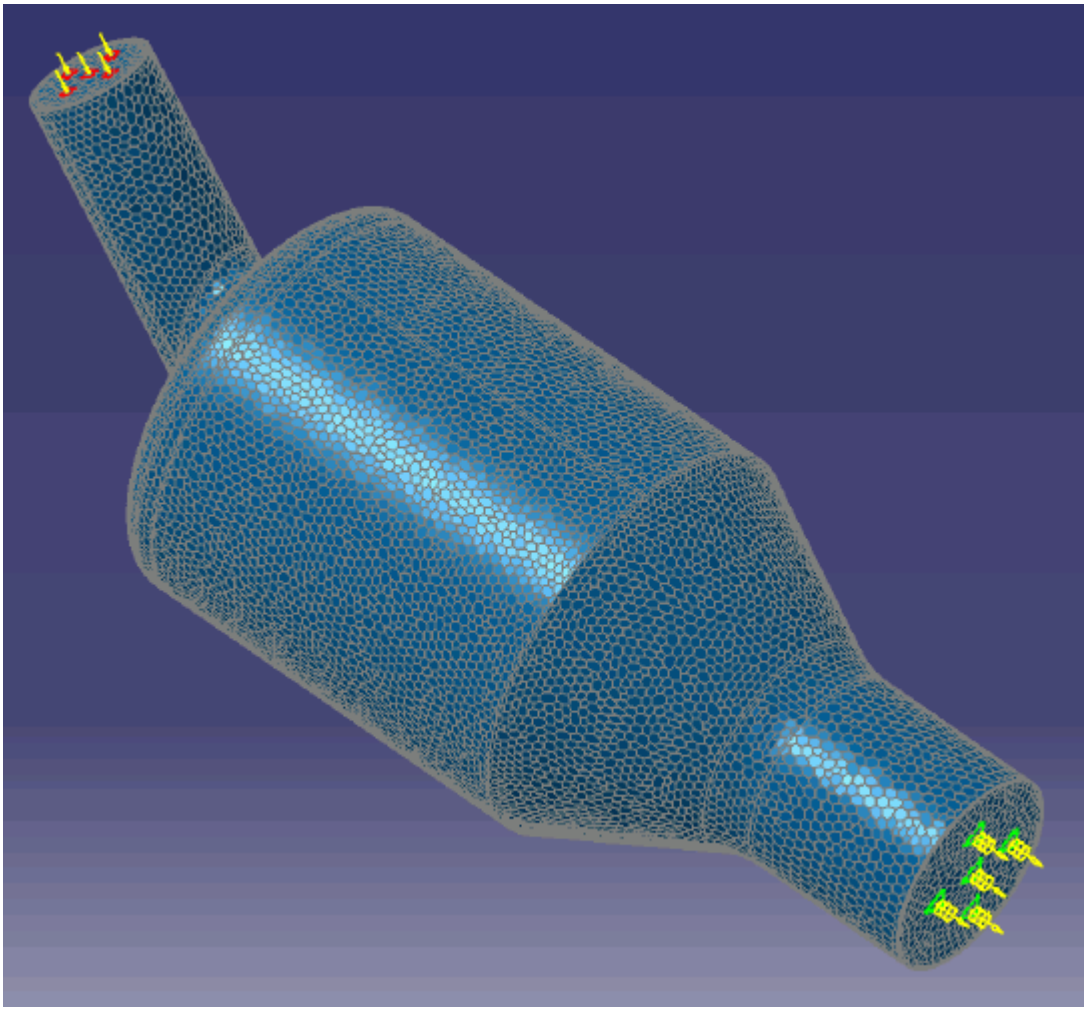
Generating the Volume Mesh

Generate the volume mesh:

1. Select **STAR-CAT5** > **Mesh** > **Generate Volume Mesh**.



The volume mesh is shown below.

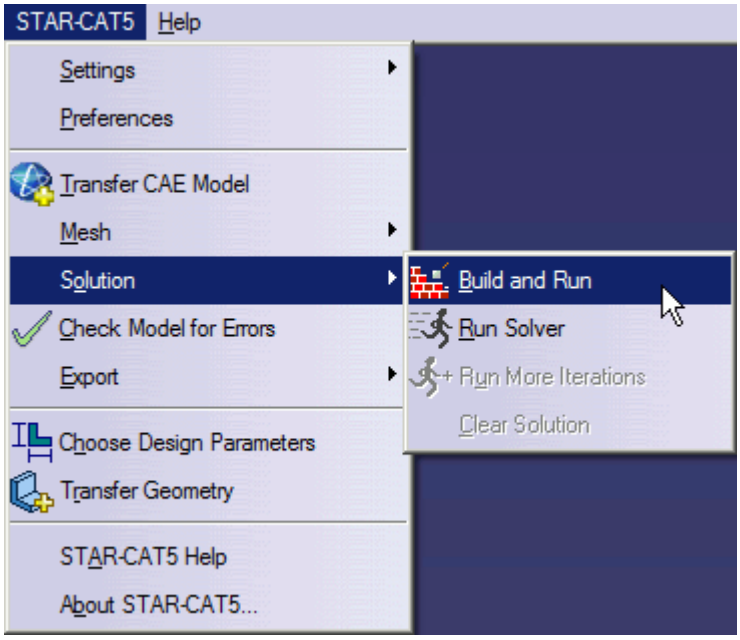


Building and Running

The case is now ready to run.

To run the simulation:

1. Select **STAR-CAT5** > **Solution** > **Build and Run**.



The *Residuals* plot appears in the STAR-CAT5 *Output* window. You can use this plot to monitor the convergence of the solution.

When the solver is done, STAR-CCM+ launches.

Post-Processing Results

You perform all post-processing in STAR-CCM+. By default, a scalar scene, *Scalar Scene 1*, automatically appears in the *Graphics* window when the solver finishes. This scene displays the absolute total pressure.

The results that are contained in the following sections are representative. You may see some variation in the results that you obtain, depending on what platform and package you are using.

1. In STAR-CCM+, click **File** > **Save** to save the simulation.

Contents:

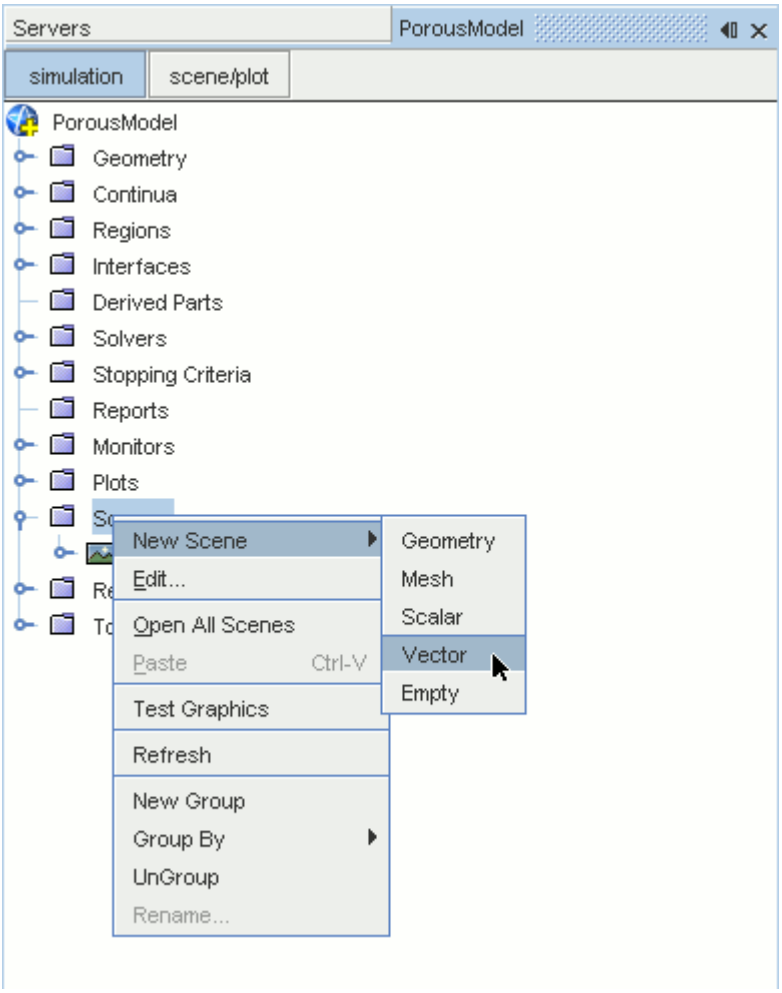
Creating a Scalar Scene
Creating A Streamlines Scene
Creating a Vector Scene
Creating a Streamlines Scene
Creating A Scalar Scene
Visualizing the Wing Skin Temperature
Creating a Streamlines Scene
Creating A Scalar Scene
Creating a Streamlines Scene
Creating a Scalar Scene
Modifying The Scalar Scene
Creating a Streamlines Scene

Creating a Vector Scene

Create a vector scene with a section plane bisecting the geometry. Velocity vectors are plotted on this plane.

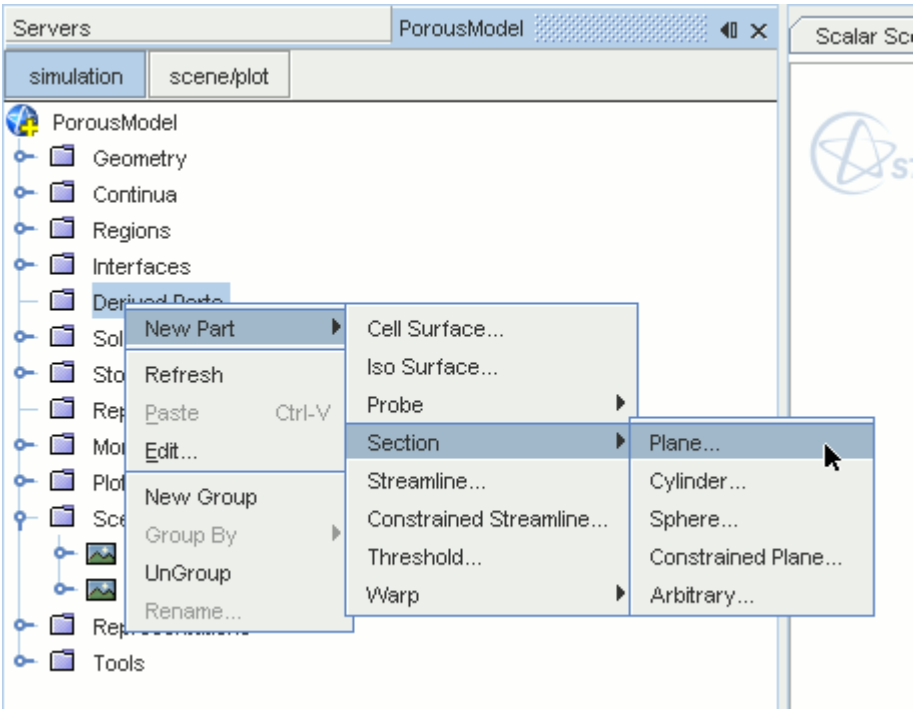
To create the vector scene:

1. Right-click the **Scenes** node in the explorer pane and select **New Scene** > **Vector**.



   A new vector scene, *Vector Scene 1*, is created and selected in the *Graphics* window.

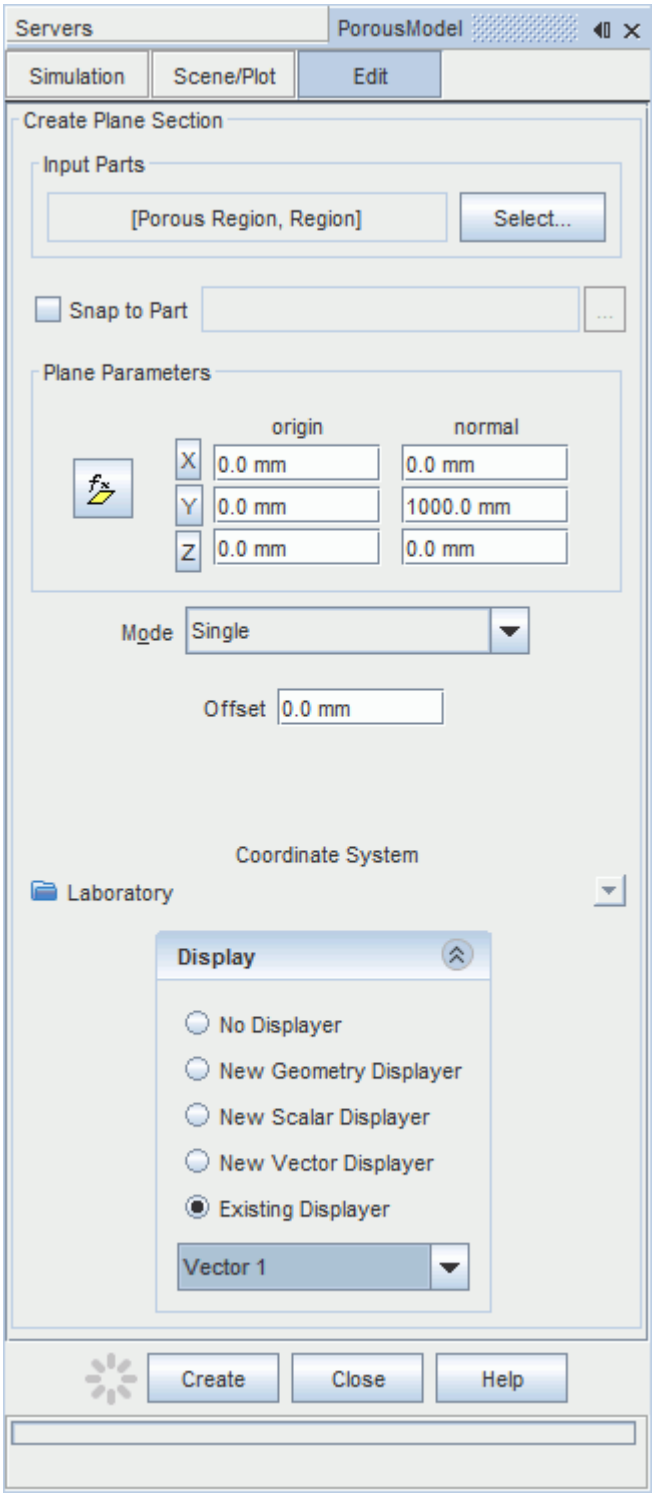
2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane...**.



The *Create Plane Section* dialog appears.

3. In the *Plane Parameters* box, set the *origin* vector to `[0 mm, 0 mm, 0 mm]`.
4. Set the *normal* vector to `[0 mm, 1000 mm, 0 mm]`.
5. In the *Display* group box, select **Existing Displayer** and in the drop-down menu select **Vector 1**.

The completed dialog is shown below.



6. Click **Create** followed by **Close**.

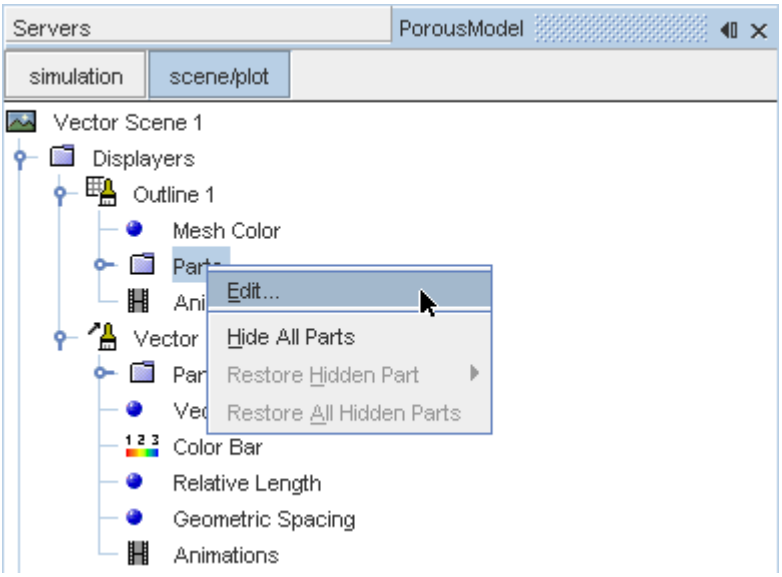
   Velocity is automatically set as the active function when creating a vector displayer.

Change the view of the scene to something more suitable:

7. On the toolbar, click (**Save-Restore-Select views**) and select **Views** > **+X +Y +Z** > **Up +Y**.

In order to remove the unwanted outlines from the scene do the following:

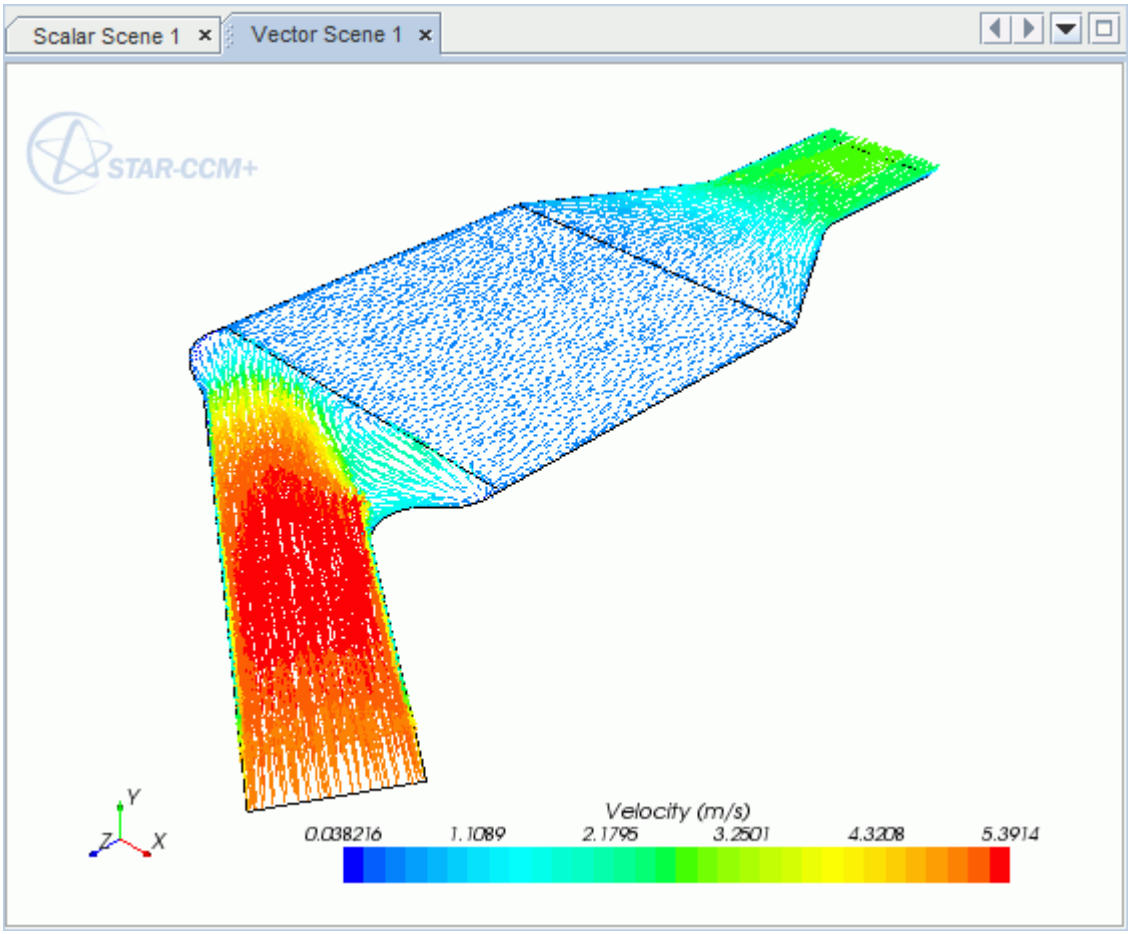
8. Click the **scene/plot** button in the explorer pane.
9. Navigate to **Vector Scene 1** > **Displayers** > **Outline 1** > **Parts**.
10. Right-click the **Parts** node and select **Edit...**.



The *Parts* dialog appears.

11. Right-click on a blank area and choose **Deselect All**.
12. Select the **Derived Parts** > **Plane Section** node.
13. Click **OK**.
14. Click the **simulation** button to return to the simulation object tree.
15. Click (**Save**) to save the simulation .

The finished scene is shown below.



Creating a Streamlines Scene

Create streamlines to get a better understanding of the flow inside the catalyst.

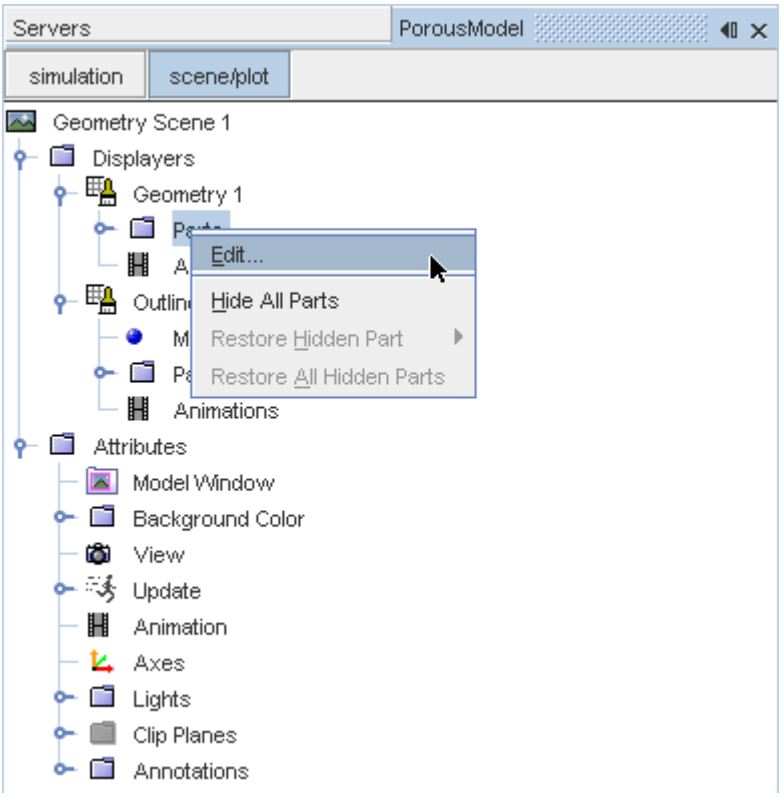
Begin by creating a geometry scene:

1. Right-click the **Scenes** node in the object tree and select **New Scene** > **Geometry**.

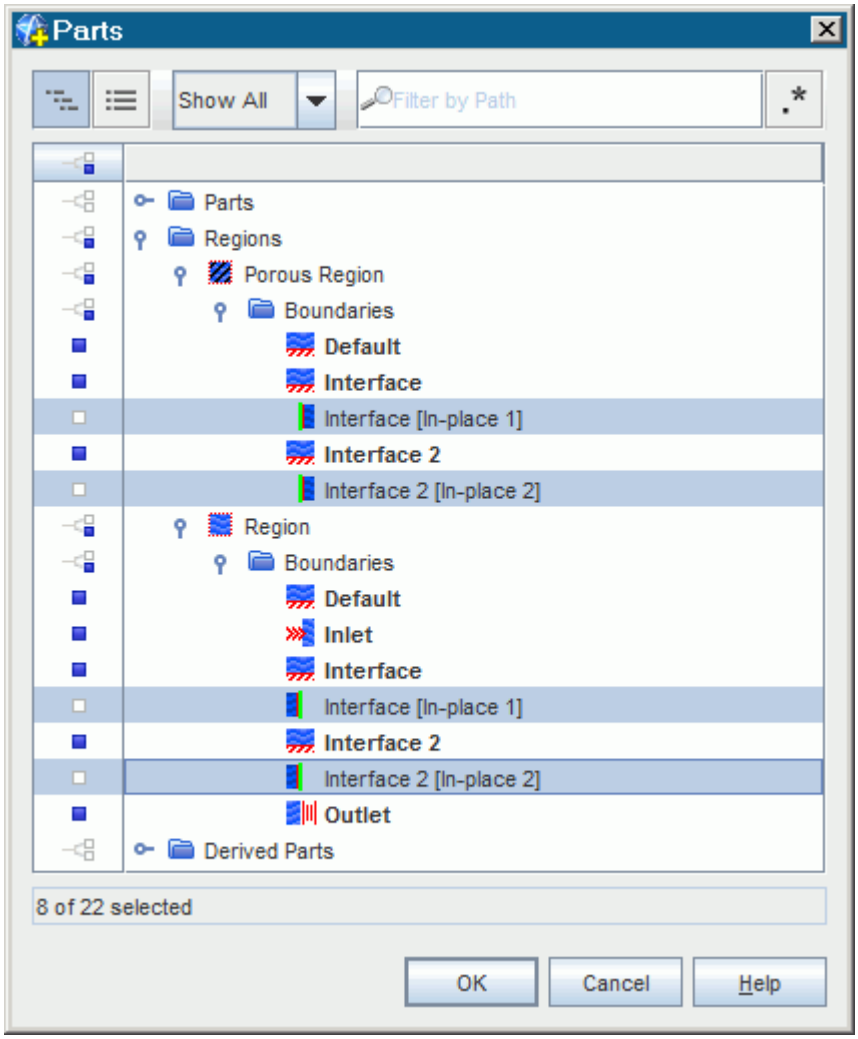
   A new geometry scene is added to the *Graphics* window.
2. Click **scene/plot**.
3. Navigate to **Geometry Scene 1** > **Displayers** > **Geometry 1**.
4. Set *Opacity* to `0.3`.

To remove the yellow interface boundaries from the scene:

5. Right-click **Geometry Scene 1** > **Displayers** > **Geometry 1** > **Parts** and select **Edit...**.



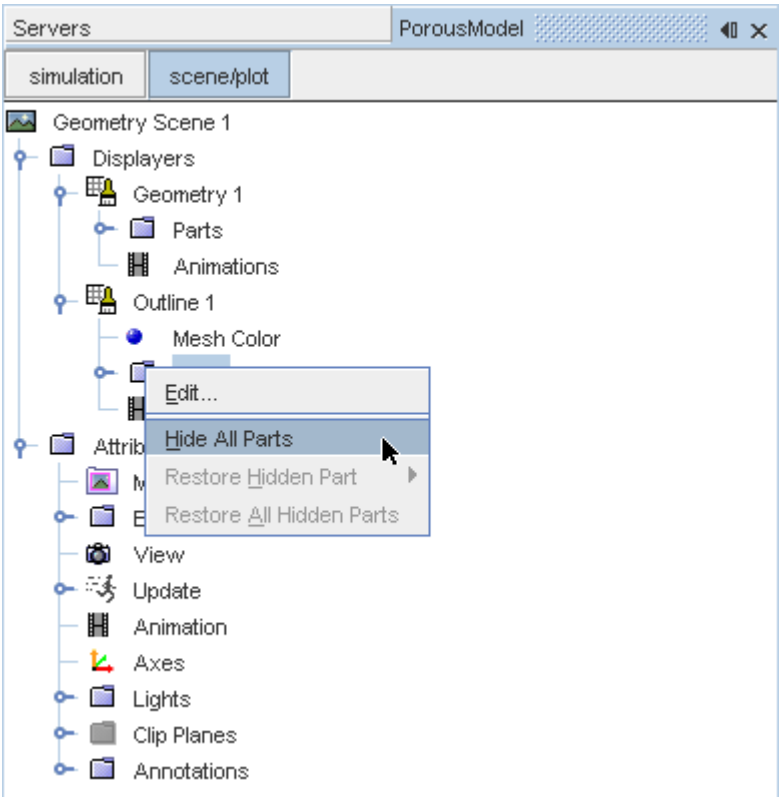
6. In the *Parts* dialog that appears, expand the **Regions** > **Porous Region** > **Boundaries** and **Regions** > **Region** > **Boundaries** nodes.
7. Deactivate the following items in each region:
   - **Interface [In-Place 1]**
   - **Interface 2 [In-Place 2]**



8. Click **OK**.

Hide the outlines on the geometry model:

9. Right-click the **Geometry Scene 1** > **Displayers** > **Outline 1** > **Parts** node in the object tree and select **Hide All Parts**.



10. Click **simulation**.

Change the view of the scene to something more suitable:

11. On the toolbar, click (**Save-Restore-Select views**) and select **Views** > **+X +Y +Z** > **Up +Y**.

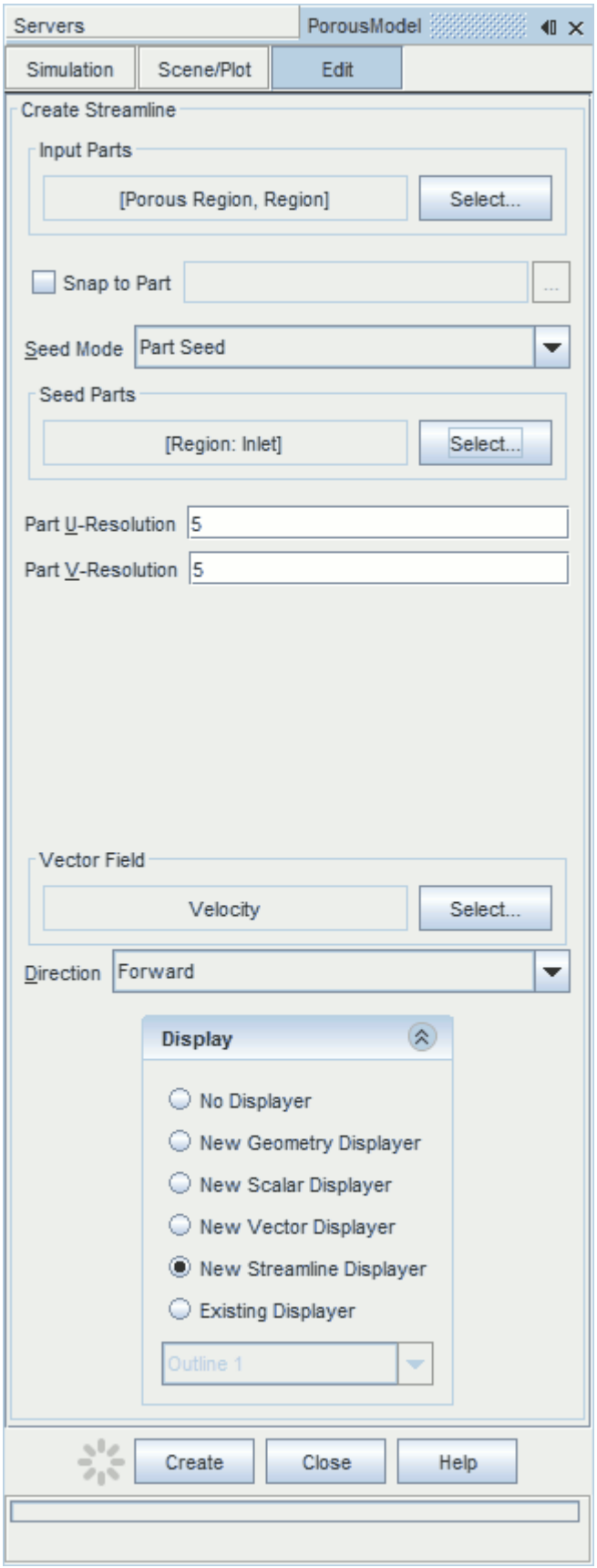
Create the streamlines:

12. Right-click on the **Derived Parts** node and select **New Part** > **Streamline...**.

The *Create Streamline* dialog appears.

13. Make sure that **Part Seed** is selected in the *Seed Mode* section.
14. Set *Seed Parts* to **Regions** > **Region** > **Boundaries** > **Inlet**.
15. Click **OK**.
16. Make sure that **New Streamline Displayer** is selected in the *Display* group box.

The completed dialog is shown below.


Servers
PorousModel
Simulation
Scene/Plot
Edit
Create Streamline
Input Parts
[Porous Region, Region]
Select...
Snap to Part
Seed Mode
Part Seed
Seed Parts
[Region: Inlet]
Select...
Part U-Resolution
5
Part V-Resolution
5
Vector Field
Velocity
Select...
Direction
Forward
Display
No Displayer
New Geometry Displayer
New Scalar Displayer
New Vector Displayer
New Streamline Displayer
Existing Displayer
Outline 1
Create
Close
Help

17. Click **Create** then **Close**.

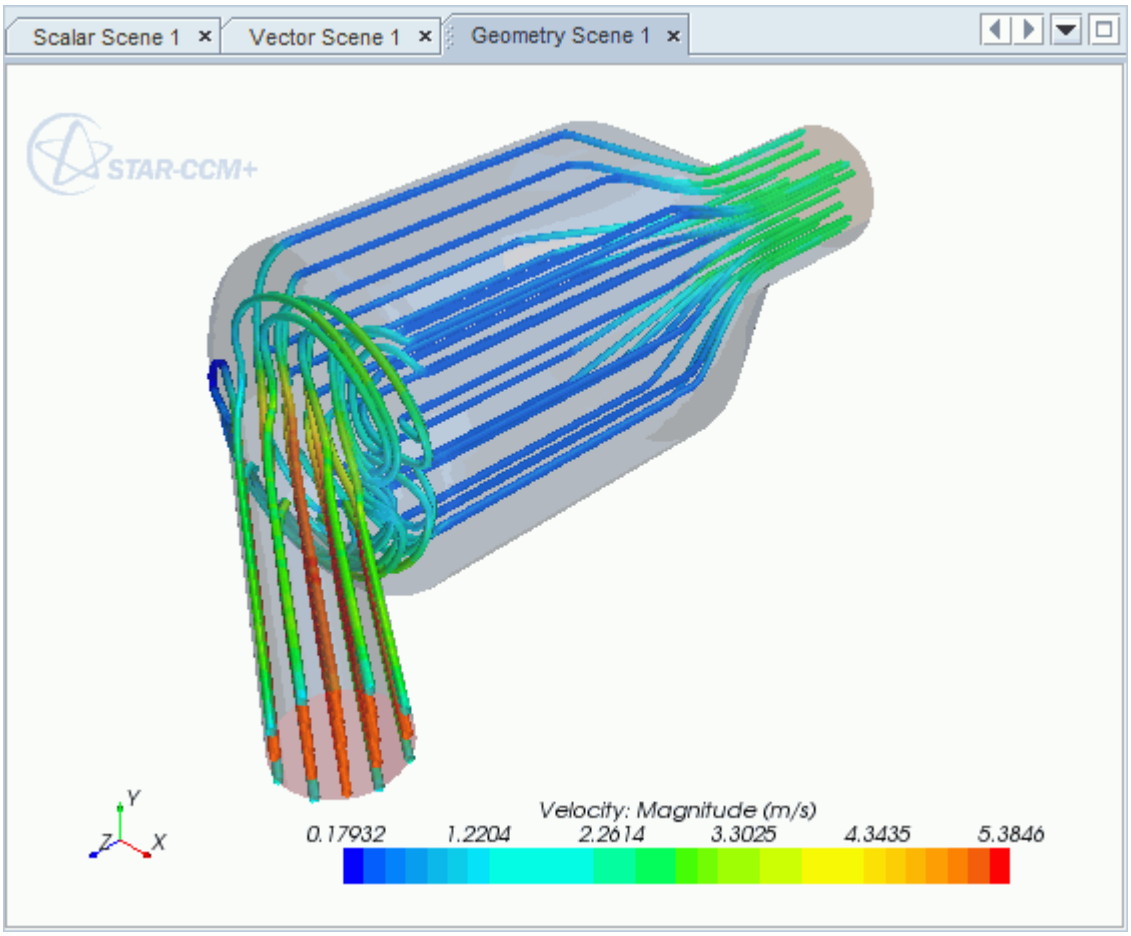
Change the display mode of the streamlines to use tubes instead of the default lines:

18. Click **scene/plot**.
19. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1**.
20. Set *Width* to `0.0012` and *Mode* to **Tubes**.

Change the color of the streamlines to represent velocity magnitude:

21. Navigate to the **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** > **Scalar Field**.
22. Set *Function* to **Velocity** > **Lab Reference Frame** > **Magnitude**.
23. Save the simulation.

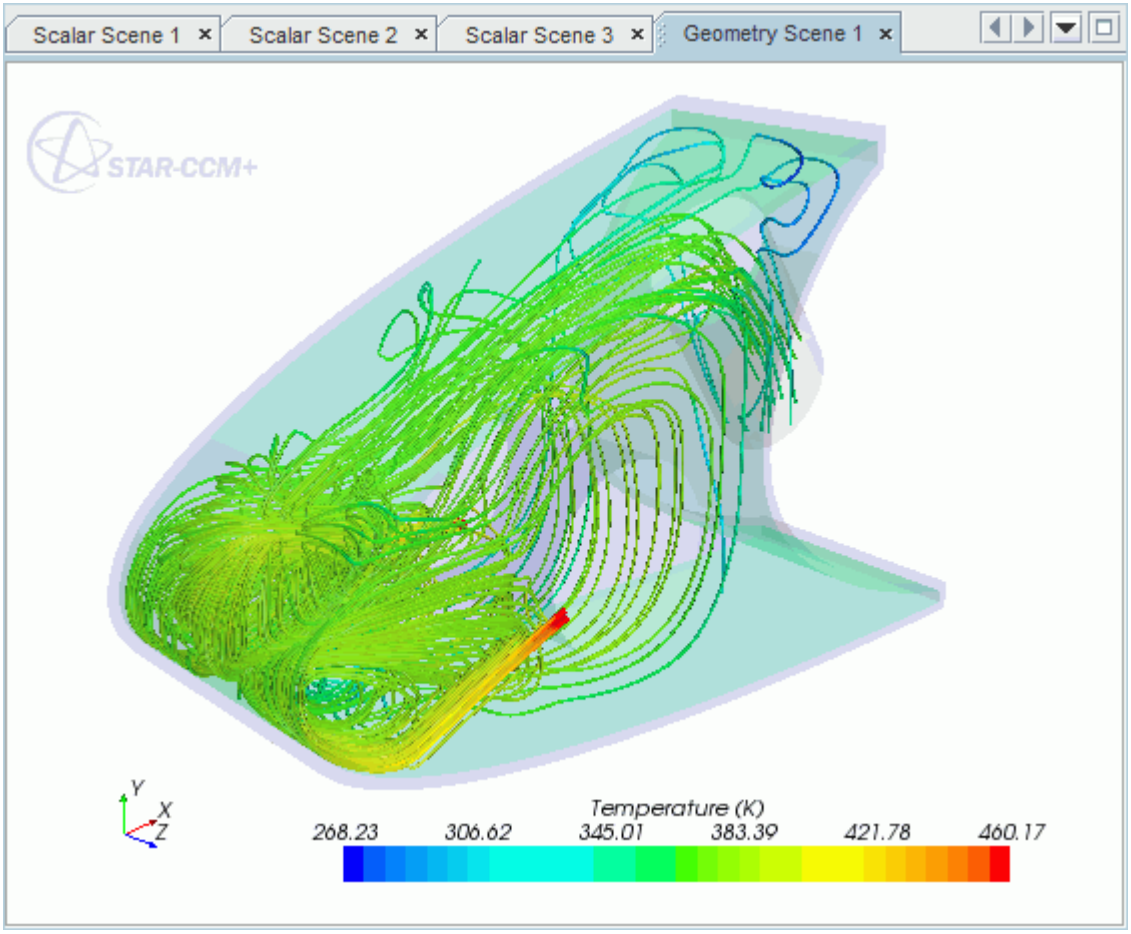
The finished scene is shown below.



Aircraft Wing Deicing System

This tutorial takes you through the steps that are required to run a CFD calculation in the STAR-CAT5 environment.

For this tutorial, the model is a modern deicing system that is used on aircraft wings. The hot air from the aircraft's jet engine is used to heat the leading edge of the wing.



Contents:

Opening the CAD Model
Selecting Fluid Physics Models
Adding Solid Physics Models
Setting the Solid Region
Defining Boundary Conditions for the Fluid Region
Defining Boundary Conditions for the Solid Region
Setting the Mesh Pipeline
Creating the Volumetric Mesh Controls

Creating a Part Mesh Control
Creating a Face Mesh Control
Modifying Solution Controls
Reviewing the Simulation Set Up
Generating the Surface Mesh
Generating the Volume Mesh
Building and Running
Post-Processing Results

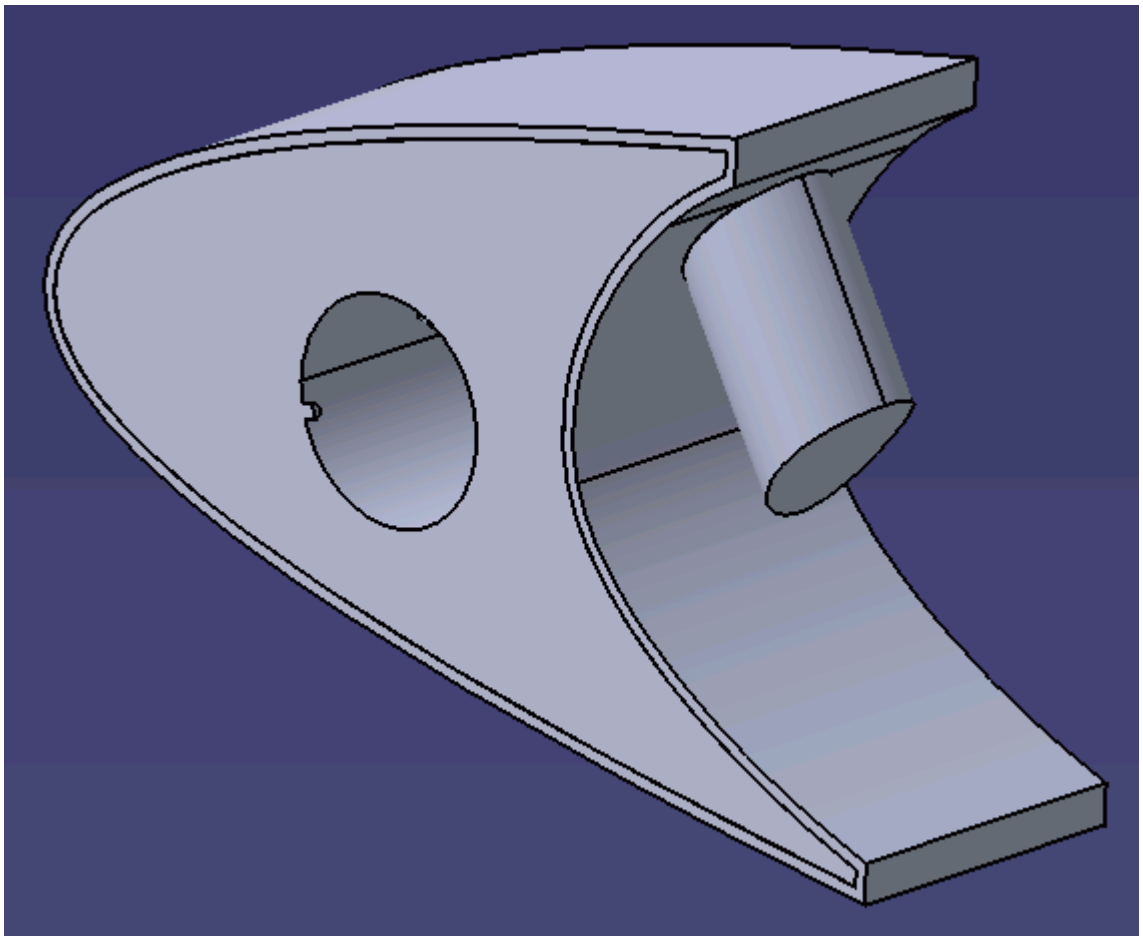
Opening the CAD Model

This directory is located in `[STAR-CCM+_INSTALL_DIR]\doc\startutorialsdata\starcat5\data`.

1. Copy directory `data` to your own disk space and open the part file `piccoloAssem.CATProduct` in STAR-CAT5.

   See Launching STAR-CAT5.

   The assembly is made up of two parts: the solid leading edge wing skin, and the internal air region.
2. Position the geometry as shown below.



The STAR-CAT5 menu appears at the top of the screen.The STAR-CAT5 object tree is located on the left of the screen.

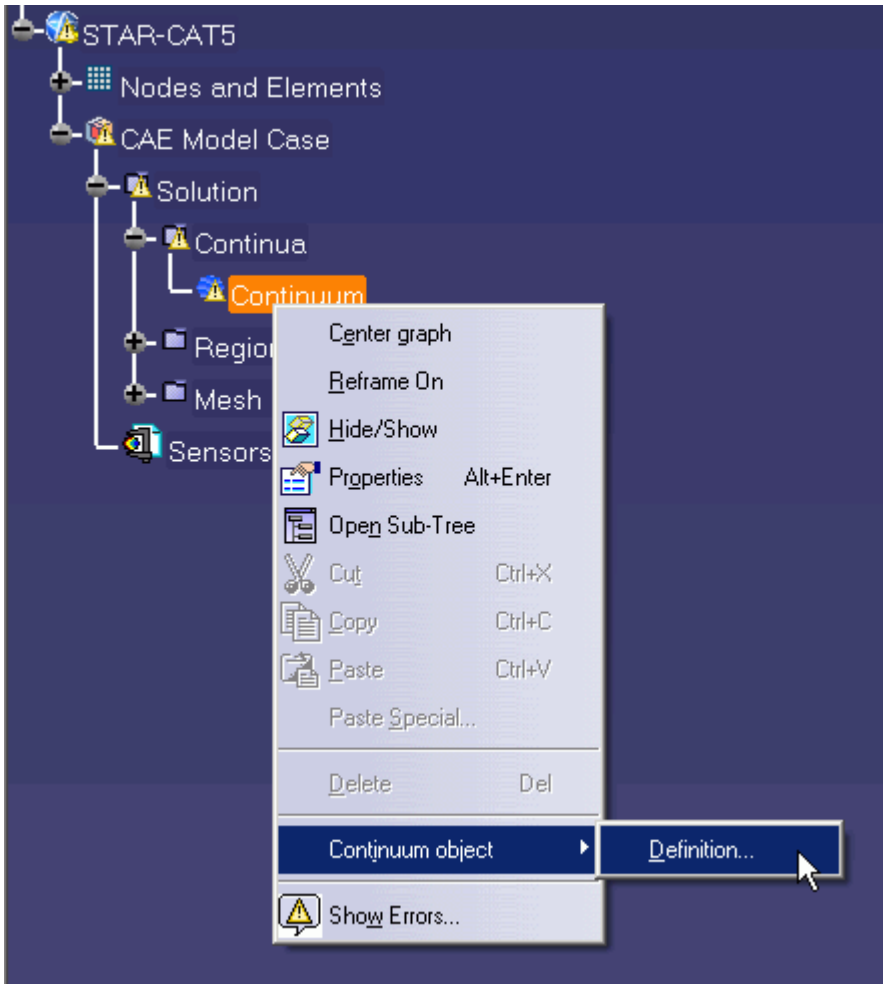
3. Save the analysis as `piccoloAssem.CATAnalysis`.

Selecting Fluid Physics Models

The first step in setting up this tutorial is to modify the physics continuum that is used to model air. For this case, the flow is turbulent and thermal. Temperature calculations are on by default in STAR-CAT5.

Modify the initial conditions:

1. Right-click the **Continua** > **Continuum** node in the STAR-CAT5 tree and select **Continuum object** > **Definition...**.



   The *Continuum Settings* dialog appears.

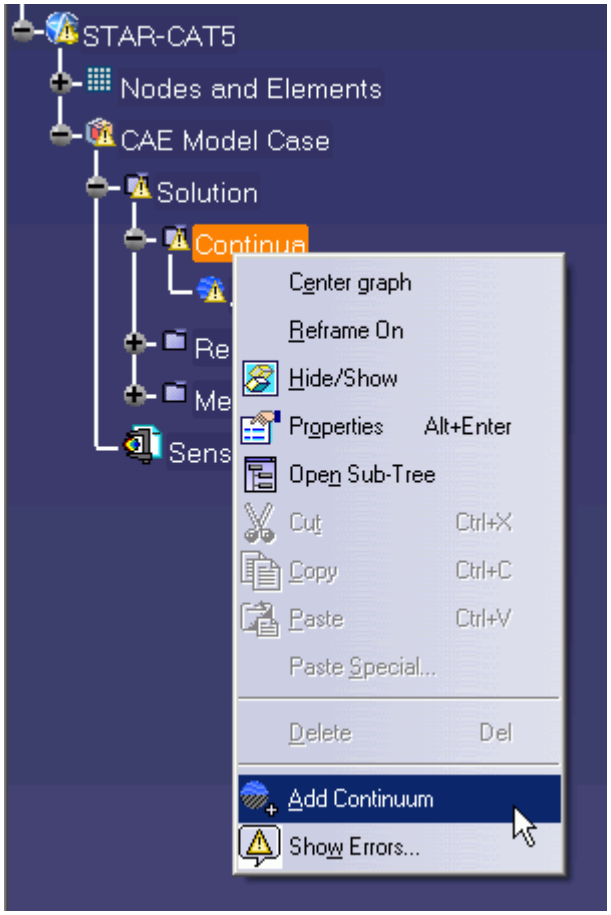
2. Set *Name* to `Air`.
3. Set the *Equation of State* to **Ideal Gas**.
4. Click the *Initialization* tab.

5. Set *Temperature* to `400` K.
6. Click **OK** to close the dialog.

Adding Solid Physics Models

The next step in setting up this tutorial is to define the physics continuum of the solid to be used.

1. Right-click the **Continua** node in the STAR-CAT5 object tree and select **Add Continuum**.

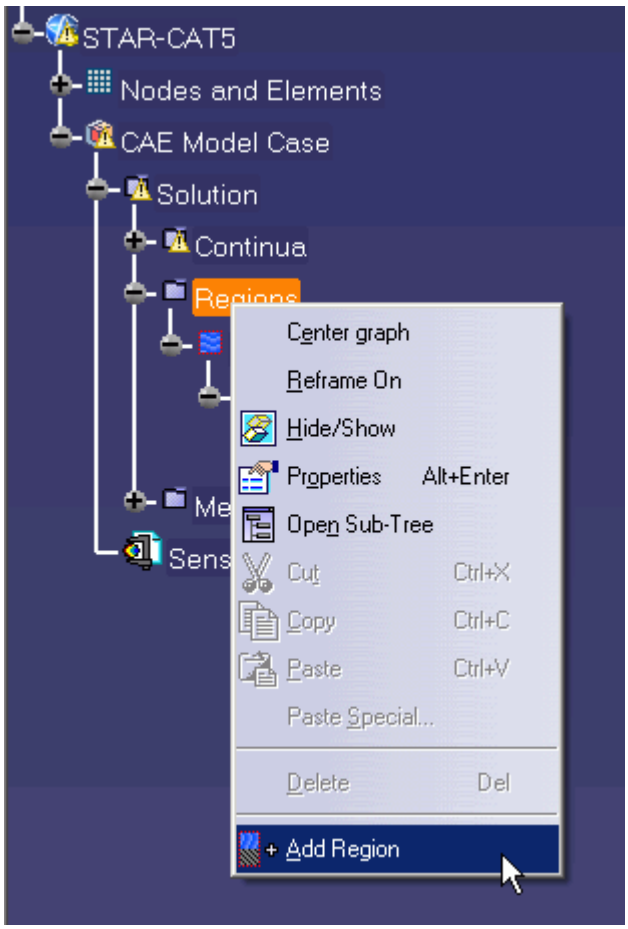


   The *Continuum Settings* dialog appears.
2. Set *Name* to `Al`.
3. Set *Material* to **Solid** and select **Aluminum Alloy** from the list.
4. Click the *Initialization* tab.
5. Set *Temperature* to `400 K`.
6. Click **OK**.

Setting the Solid Region

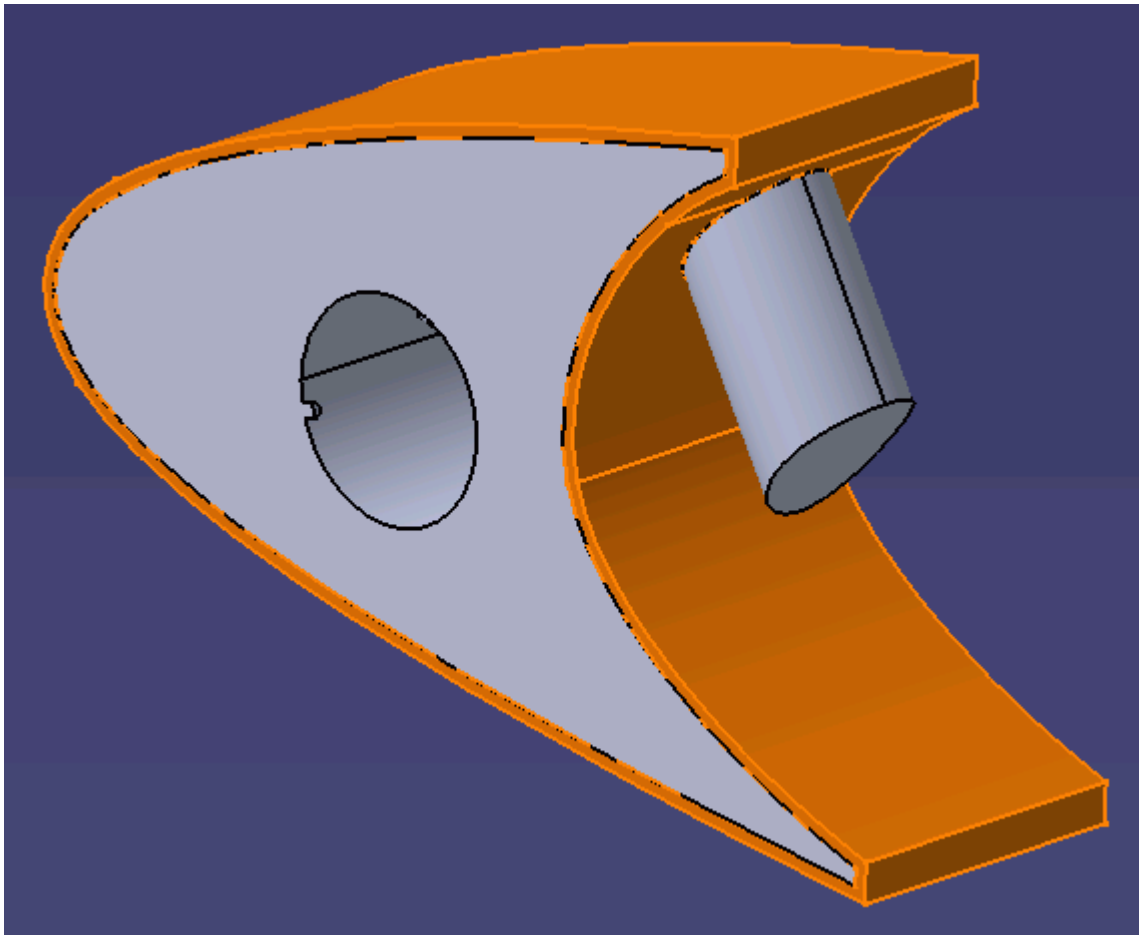
Create a new solid region for the wing skin:

1. Right-click the **Regions** node in the STAR-CAT5 object tree and select **Add Region**.



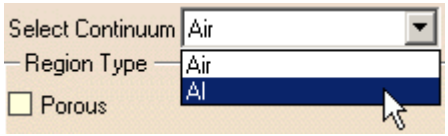
The *Region Settings* dialog appears.

2. Select the wing skin as shown below.



Once selected, the part becomes highlighted.

3. Set *Name* to `Leading Edge`.
4. Set *Select Continuum* to **Al**.

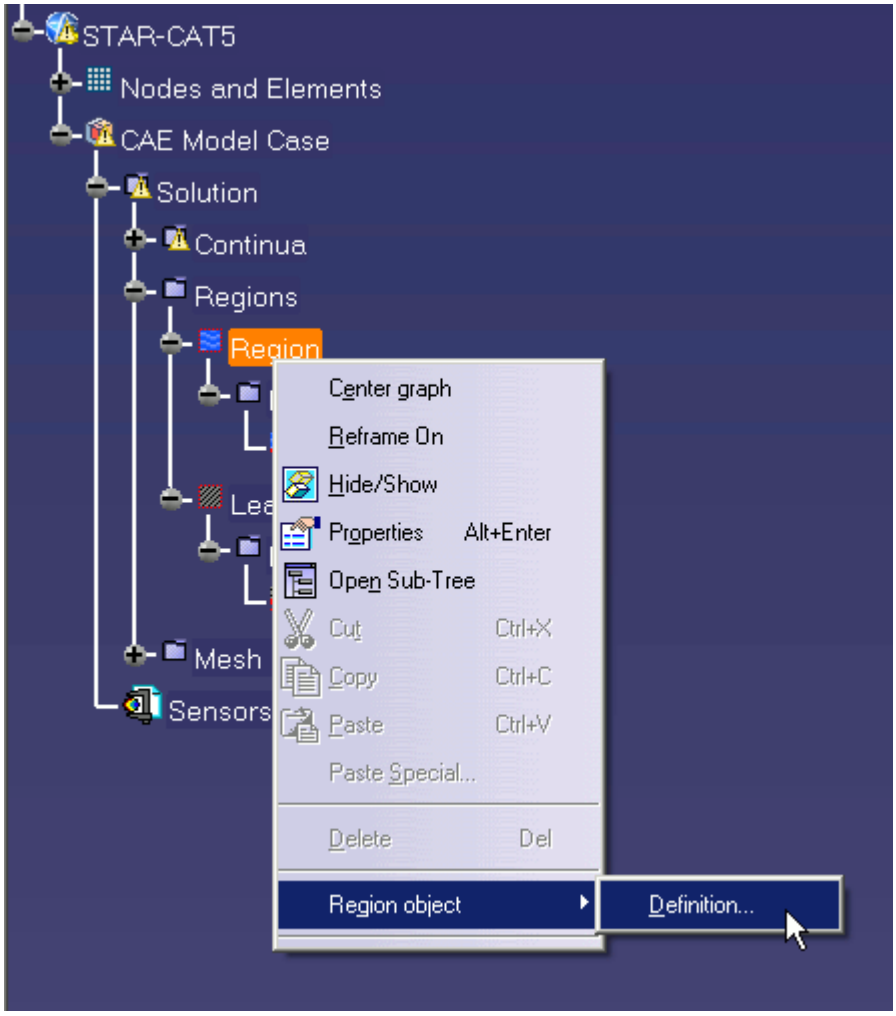


This action assigns the aluminum alloy that you created earlier as the material for this region.

5. Click **OK**.

Now that the solid region of the wing is defined, rename the default region to something more appropriate:

6. Right-click the **Regions** > **Region** node in the STAR-CAT5 tree and select **Region object** > **Definiton...**.



The *Region Settings* dialog appears.

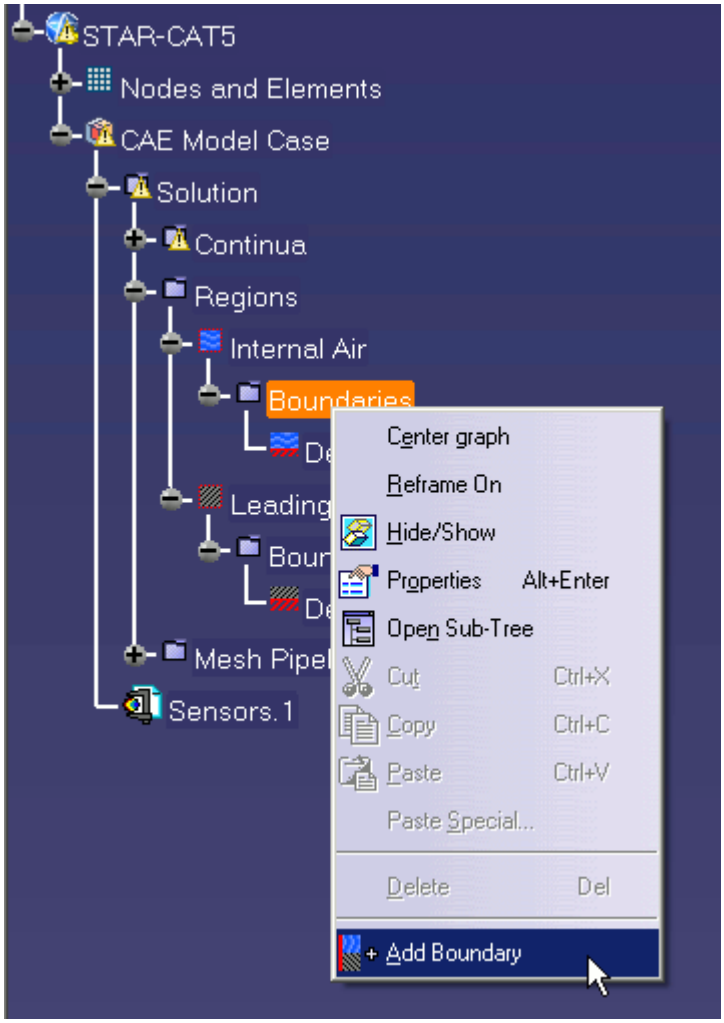
7. Enter `Internal Air` as the new name.
8. Click **OK** to close the dialog.

Defining Boundary Conditions for the Fluid Region

Create the inlet and outlet boundary conditions. Use symmetry boundaries on the side walls to extend the simulation domain. Use a wall boundary with a fixed temperature to model the piccolo walls.

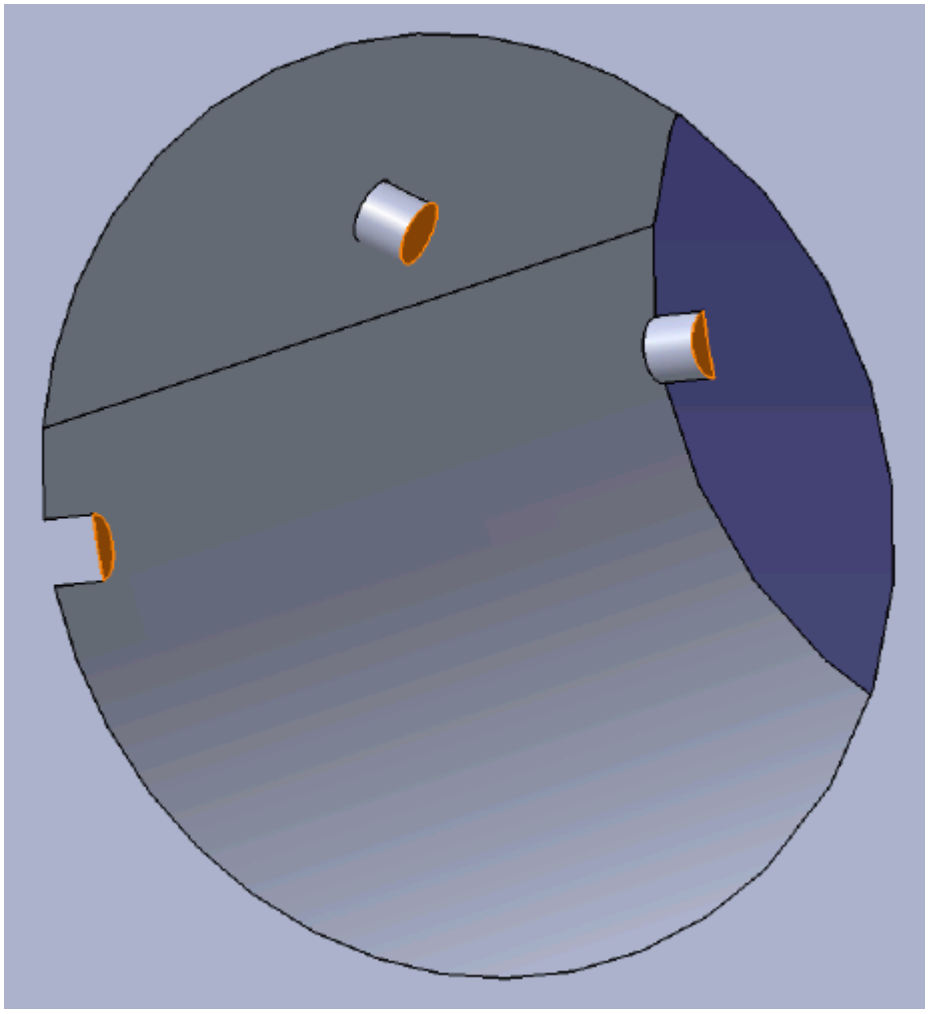
Create the inlet boundary first:

1. Right-click the **Regions** > **Internal Air** > **Boundaries** node and select **Add Boundary**.



The *Boundary Settings* dialog appears.

2. On the CAD model, select the three faces that are highlighted below.



These selections are reported in the *Supports* box in the *Boundary Settings* dialog.

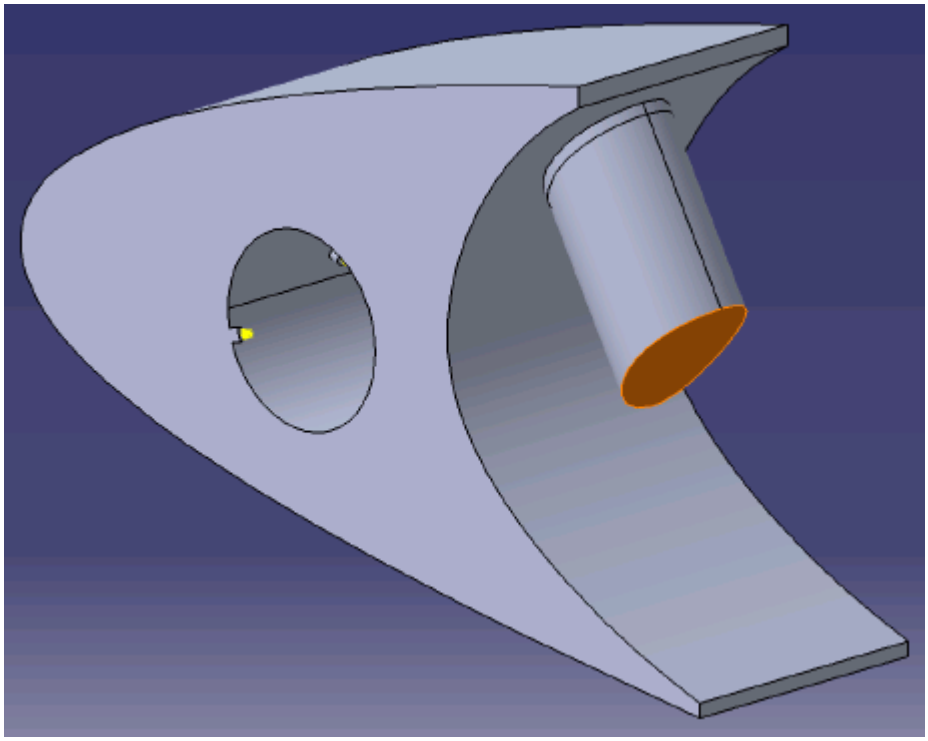
3. Set the following properties:

Parameter	Value
Boundary Type	**Velocity Inlet**
Name	`Inlet`
Normal Velocity	`90 m/s`
Temperature	`460 K`
Turbulence Intensity	`0.135`
Turbulent Length Scale	`2.2e-005 m`

4. Click **OK**.

Create the outlet boundary:

5. Using the same technique, create another boundary.
6. Select the face that is highlighted below.



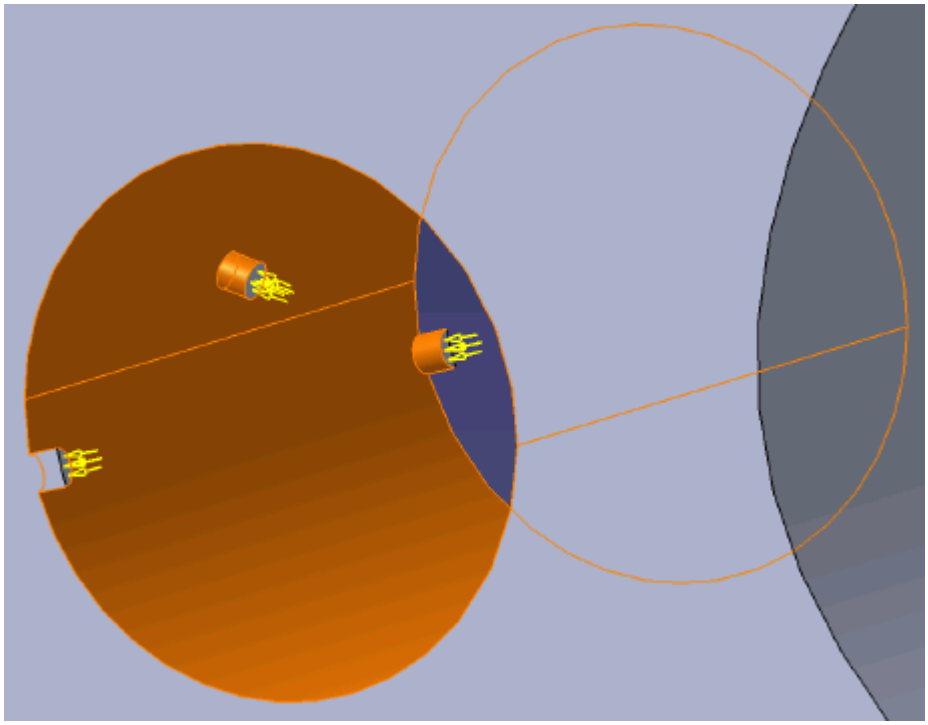
7. Set the following parameters:

Parameter	Value
Boundary Type	**Pressure Outlet**
Name	`Outlet`

8. Click **OK**.

Create the wall boundary that models the heated piccolo walls.

9. Create another boundary.
10. Select the six faces highlighted below, make sure to select the faces that are perpendicular to the inlet boundaries.



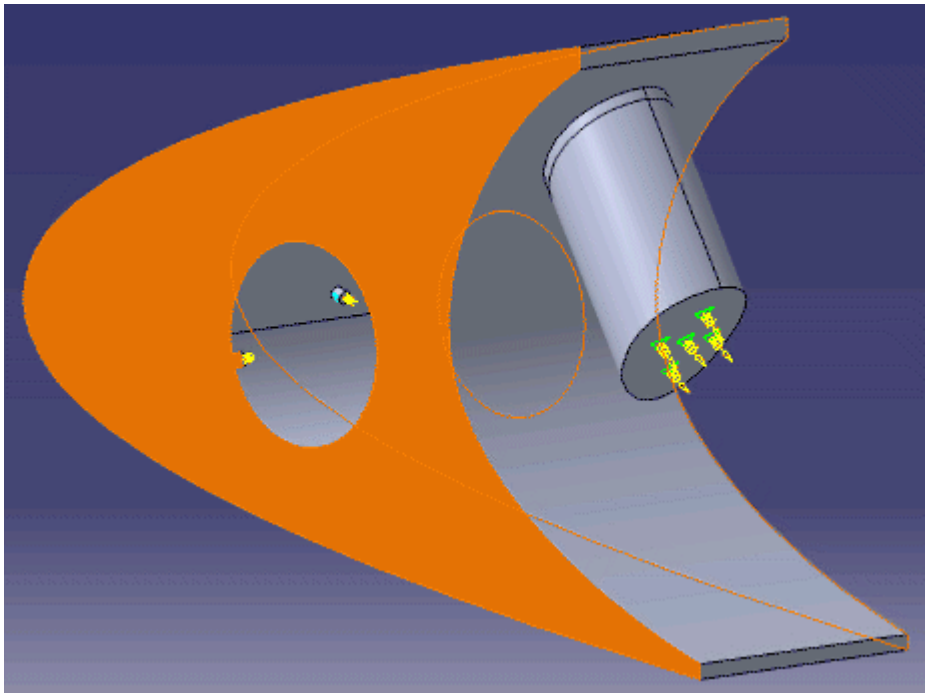
11. Set the following parameters:

Parameter	Value
Boundary Type	**Wall**
Name	`Piccolo Wall`
Thermal Type	**Temperature**
Temperature	`460 K`

12. Click **OK**.

Create the symmetry boundaries:

13. Create a new boundary.
14. Select the two faces that are highlighted below, one on either side of the internal air region.



15. Set the following parameters:

Parameter	Value
Boundary Type	**Symmetry Plane**
Name	`Air Symmetry`

16. Click **OK**.

Defining Boundary Conditions for the Solid Region

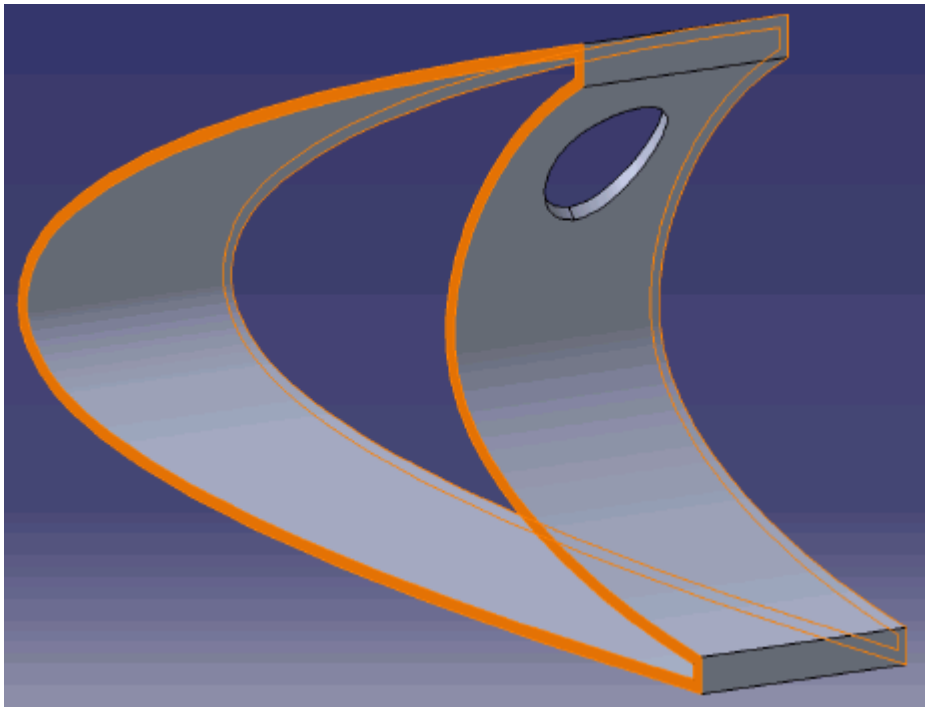
Use the convection boundary condition to model the heat propagation from the solid wing.

The convective boundary condition models ambient conditions on the outside of the domain. Use convection to model the heat propagation in the solid wing skin.

Create the symmetry boundary first:

1. Right-click the **Regions** > **Leading Edge** > **Boundaries** node and select **Add Boundary**.

   The *Boundary Settings* dialog appears.
2. Select the two faces highlighted below, one on either side of the geometry.



3. Set the following parameters:

Parameter	Value
Boundary Type	**Symmetry Plane**
Name	`Solid Symmetry`

4. Click **OK**.

Modify the default boundary so that it uses convection:

5. Right-click the **Regions** > **Leading Edge** > **Boundaries** > **Default** node and select **Default object** > **Definiton**.

   The *Boundary Settings* dialog appears.

6. Set the following properties:

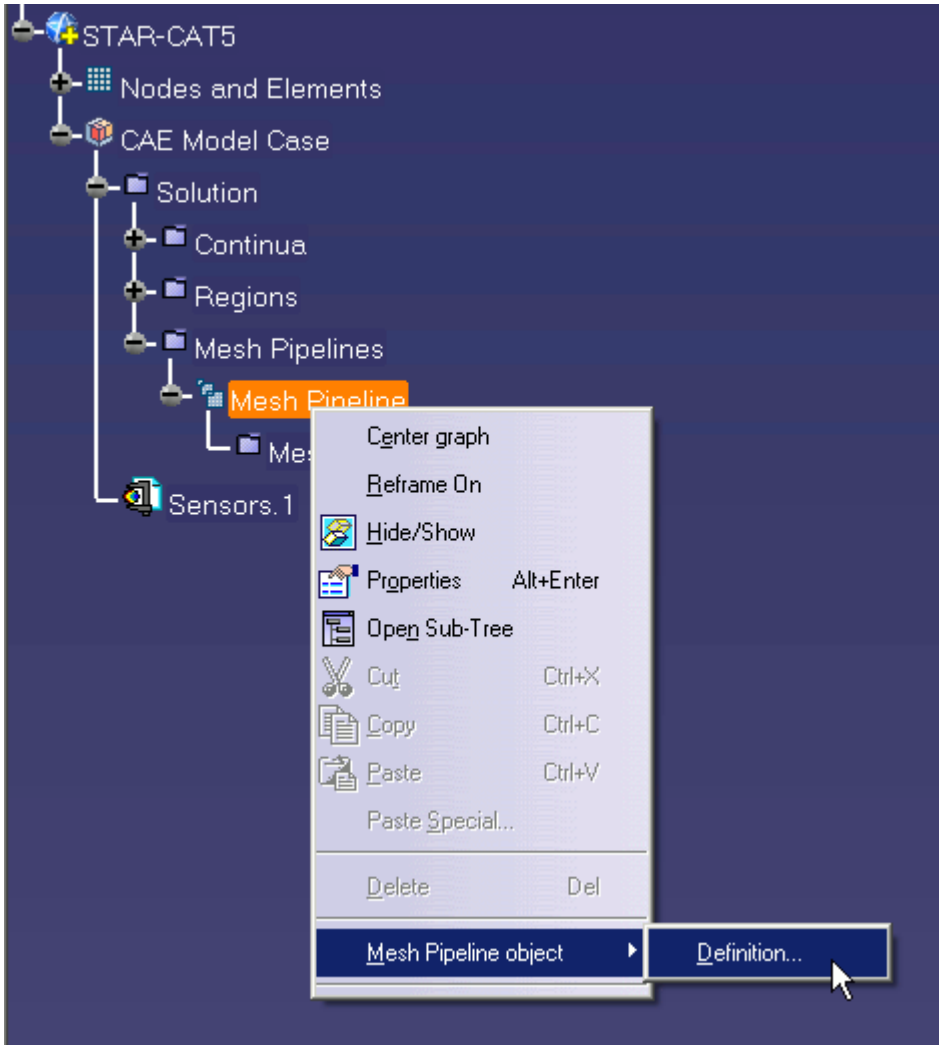
Parameter	Value
Thermal Type	**Convection**
Heat Transfer Coefficient	`100 W/m^2.K`
Temperature	`228 K`

7. Click **OK**.
8. Save your progress.

Setting the Mesh Pipeline

To obtain more accurate results, modify the mesh pipeline:

1. Right-click the **Mesh Pipelines** > **Mesh Pipeline** node in the STAR-CAT5 tree and select **Mesh Pipeline object** > **Definition**.



The *Mesh Pipeline Settings* dialog appears.

2. Set the following parameters:

Parameter	Value
Base Size	
Use Absolute Value	**Activated**
Base Size	`0.007 m`
Prism Layer	
Prism Layer Thickness	`10%`

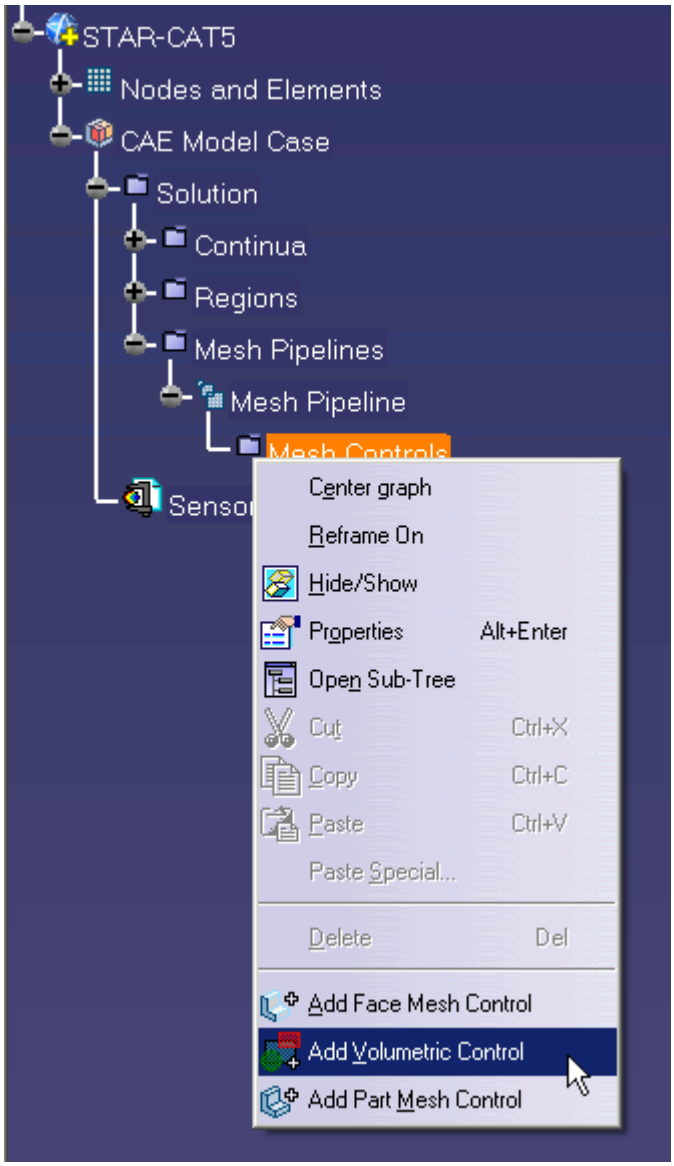
3. Click **OK**.

Creating the Volumetric Mesh Controls

Create volumetric mesh control regions to capture the finer details of the flow near the air jets that heat the leading edge of the wing.

Create the volumetric mesh control for the inlet:

1. Right-click the **Mesh Pipelines** > **Mesh Pipeline** > **Mesh Controls** node in the STAR-CAT5 tree and select **Add Volumetric Control**.



   The *Volumetric Control* dialog appears.

2. Set *Name* to `Inlet Box`.
3. Activate the *Use Absolute Value* option.
4. Set *Surface Mesh Size* to `0.0005 m`.
5. Enter the following coordinates for Point 1:

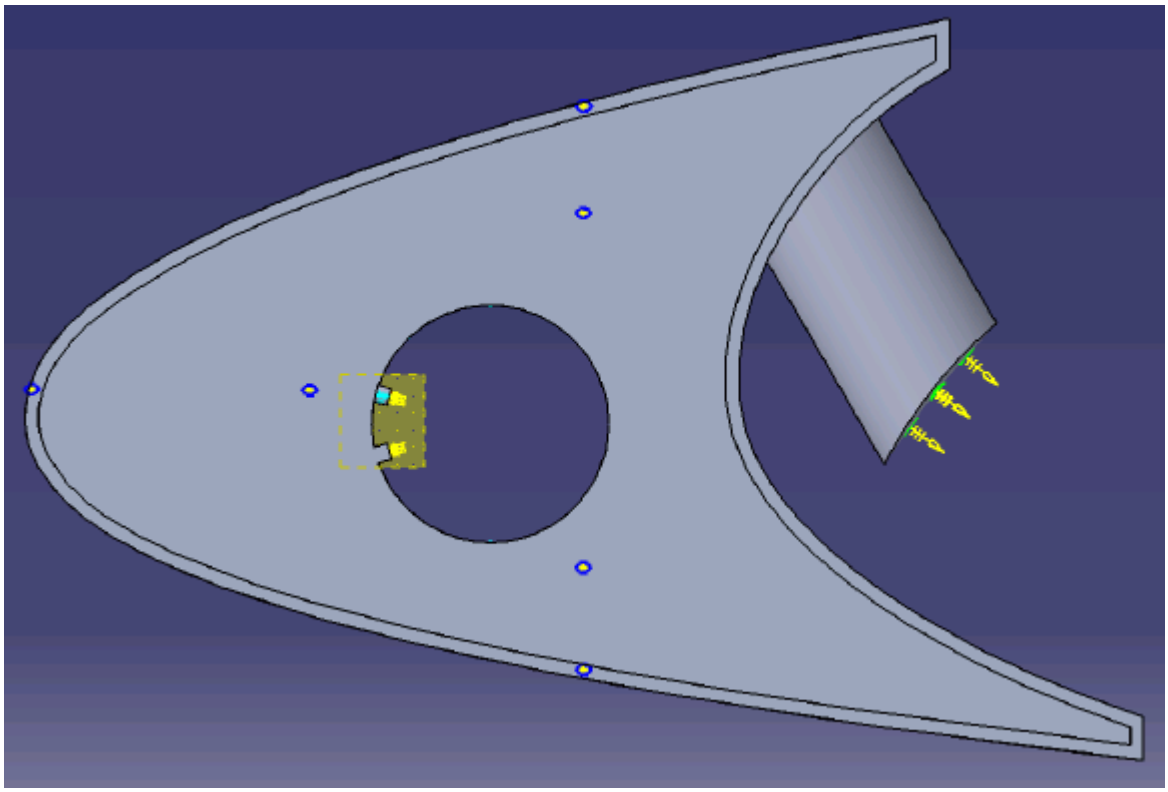
Coordinate	Value
X1	`-100.00 mm`
Y1	`21.00 mm`
Z1	`-20.00 mm`

6. Enter the following coordinates for Point 2:

Coordinate	Value
X2	`-90.00 mm`
Y2	`10.00 mm`
Z2	`60.00 mm`

7. Click **Create**.

   The volumetric control region is shown below.



8. Click **OK**.

Create the volumetric mesh control for the region near the piccolo tube and use the following settings:

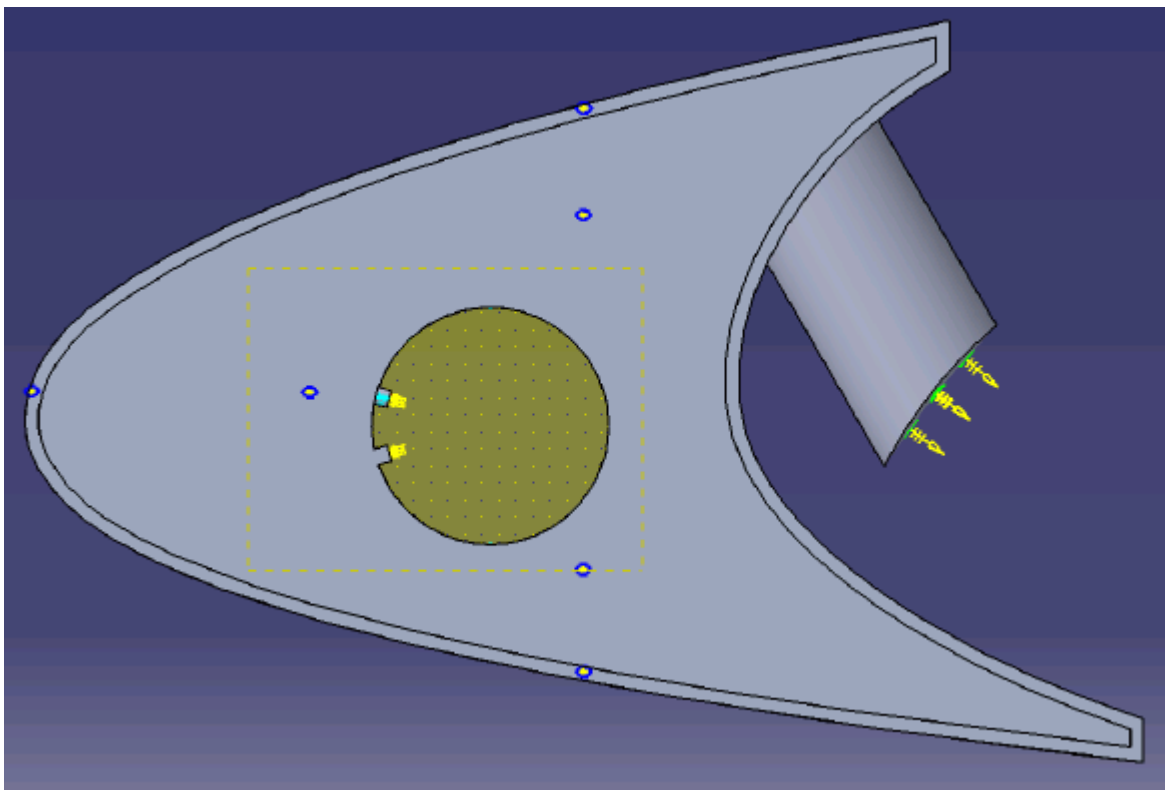
9. Using the same technique, create another volumetric mesh control with the following settings:

Parameter	Value
Name	`Mid Box`

Parameter	Value
Surface Mesh Size	
Use Absolute Value	**Activated**
Surface Mesh Size	`0.0015 m`
Shape	
X1	`-111.00 mm`
Y1	`34.00 mm`
Z1	`-20.00 mm`
X2	`-64.00 mm`
Y2	`-2.00 mm`
Z2	`60.00 mm`

10. Click **Create**.

    The volumetric control region is shown below.

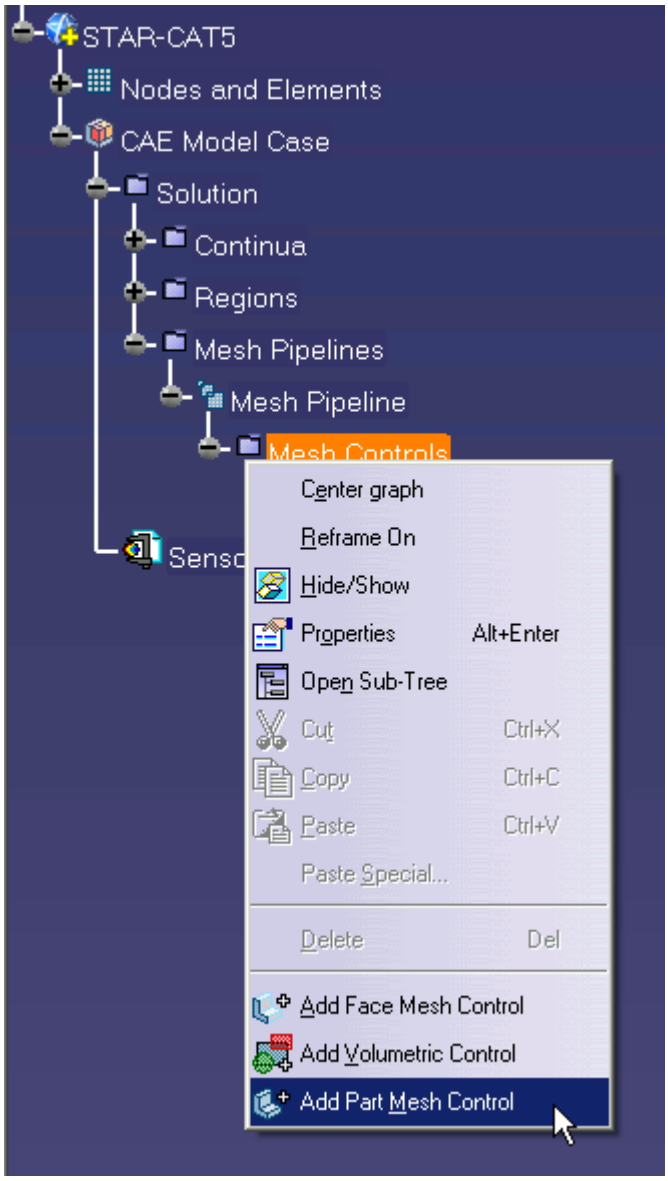


11. Click **OK**.
12. Save your progress.

Creating a Part Mesh Control

Create a part mesh control to disable the prism layer generation in the solid region. Solid regions do not require prism layers.

1. Right-click **Mesh Pipelines** > **Mesh Pipeline** > **Mesh Controls** in the STAR-CAT5 tree and select **Add Part Mesh Control**.

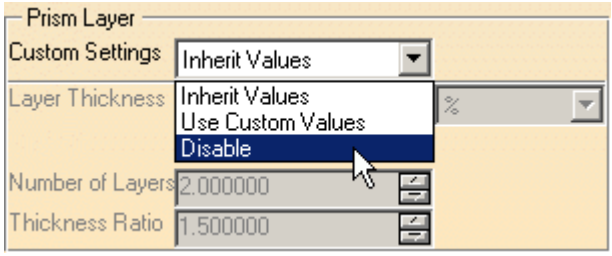


   The *Part Mesh Control Settings* dialog appears.

2. Select the solid wing region on the CAD model.

   The selection is added to the *Support* box.

3. Set *Name* to `Solid Region`.
4. In the *Prism Layer* box, set *Custom Settings* to **Disable** using the drop-down menu.

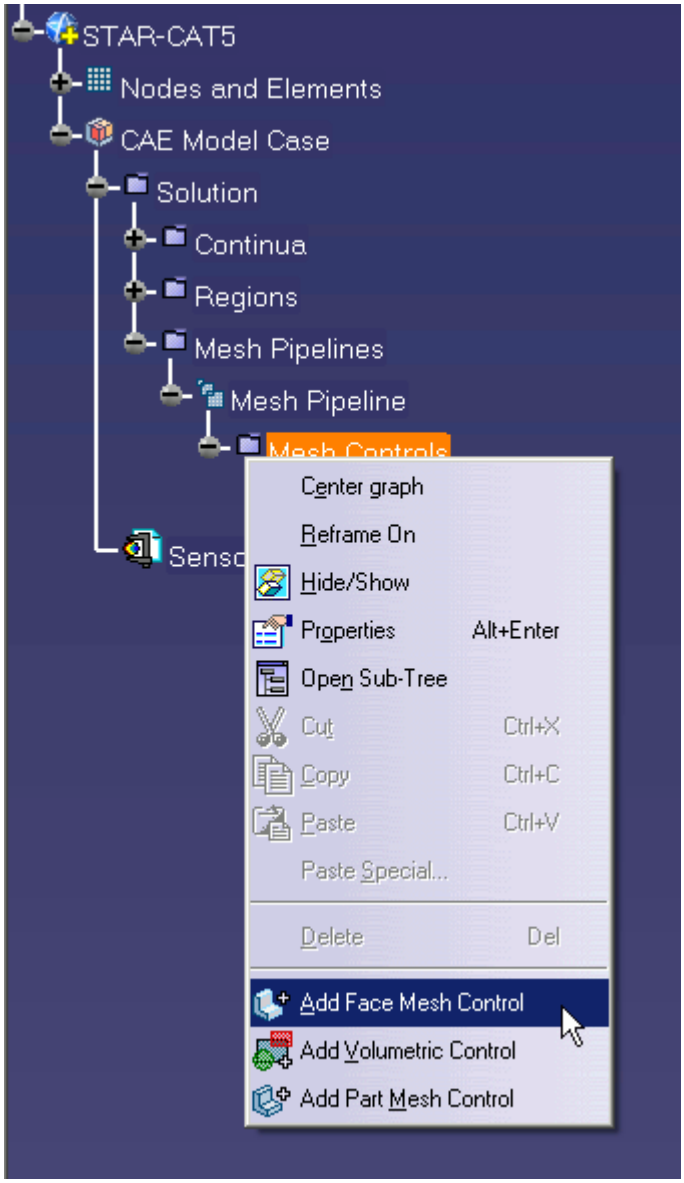


5. Click **OK**.

Creating a Face Mesh Control

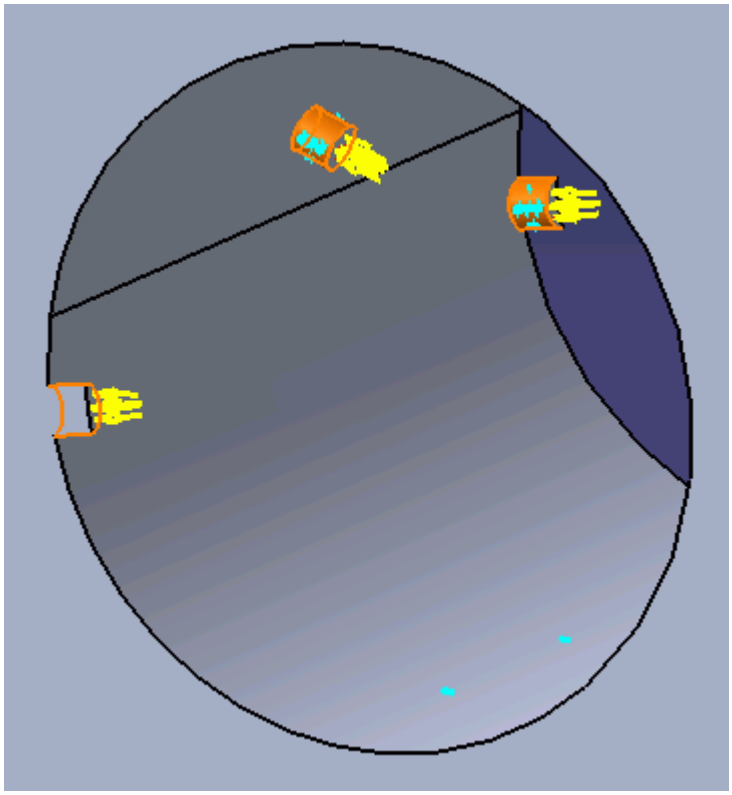
Create a face mesh control to reduce the size of the prism layer thickness at the inlet boundary.

1. Right-click **Mesh Pipelines** > **Mesh Pipeline** > **Mesh Controls** in the STAR-CAT5 tree and select **Add Face Mesh Control**.



The *Face Mesh Control Settings* dialog appears.

2. Select the cylindrical faces that are perpendicular to the inlet boundary, as shown below.

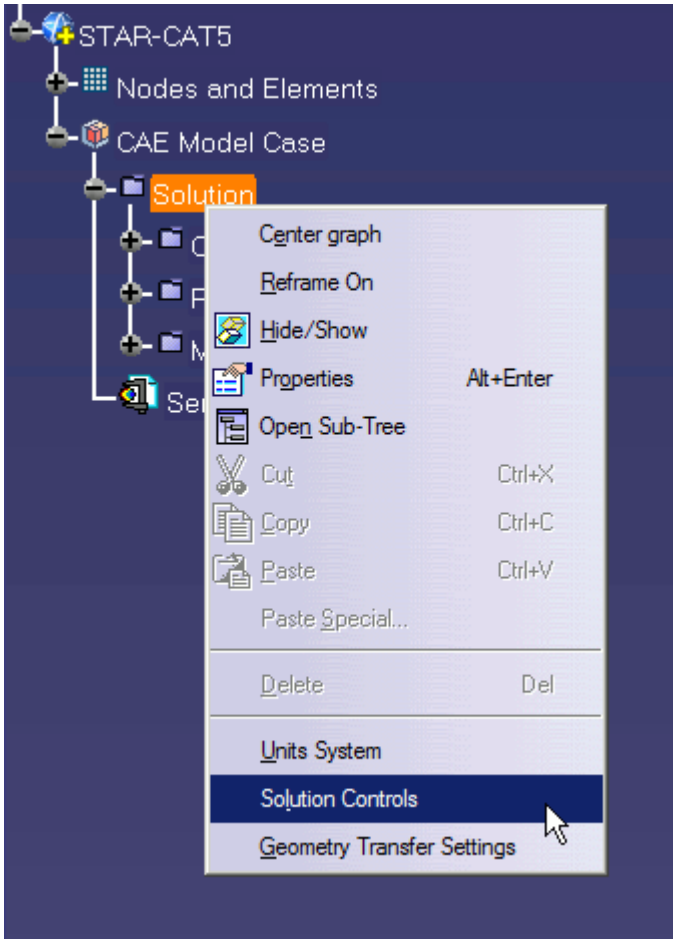


3. Set *Name* to `Inlet Refinement`.
4. Activate the *Custom Mesh Size* option.
5. Activate the *Use Absolute Value* option.
6. Set *Mesh Size* to `0.00025` m.
7. In the *Prism Layer* box, set *Custom Settings* to **Use Custom Values**.
8. Activate the *Use Absolute Value* option.
9. Set *Layer Thickness* to `0.0002` m.
10. Click **OK**.

Modifying Solution Controls

Modify the maximum number of iterations and the solution accuracy. To help accelerate convergence, edit the solid under-relaxation factor:

1. Right-click the **Solution** node in the STAR-CAT5 object tree and select **Solution Controls**



The *Solution Controls* dialog appears.

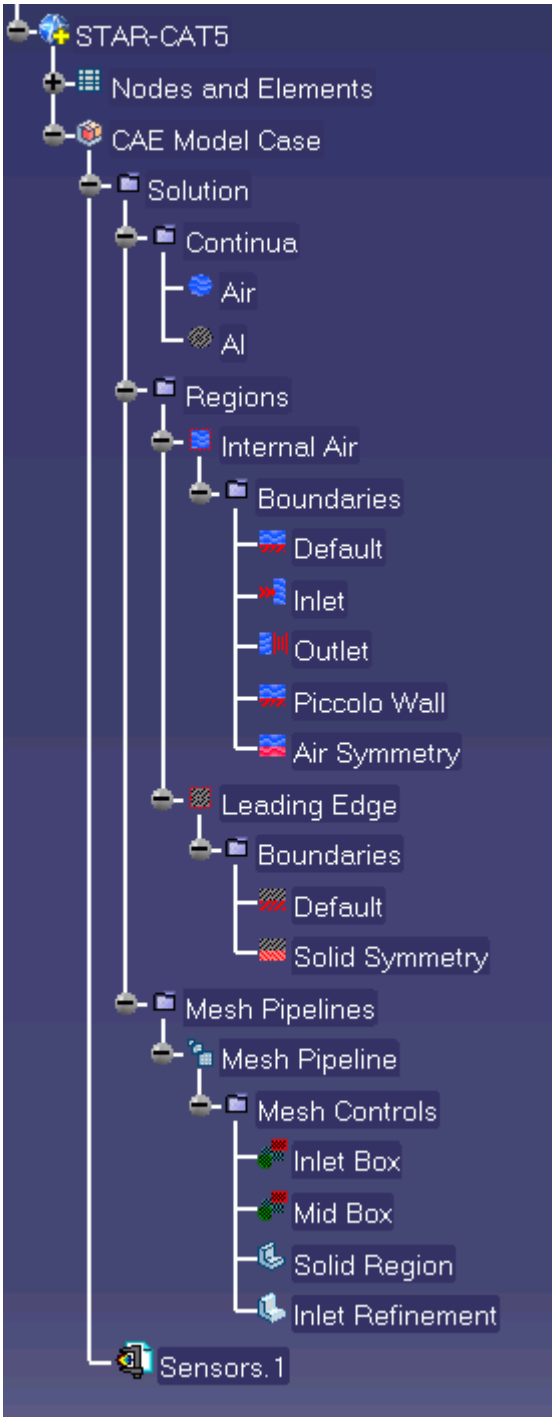
2. Click **More** to expand the *Solution Controls* menu.
3. Set the following parameters:

Parameter	Value
Iterations	500
Convergence	0.01
Solid Temperature Under-Relaxation Factor	0.9999

4. Click **OK**.
5. Save your progress.

Reviewing the Simulation Set Up

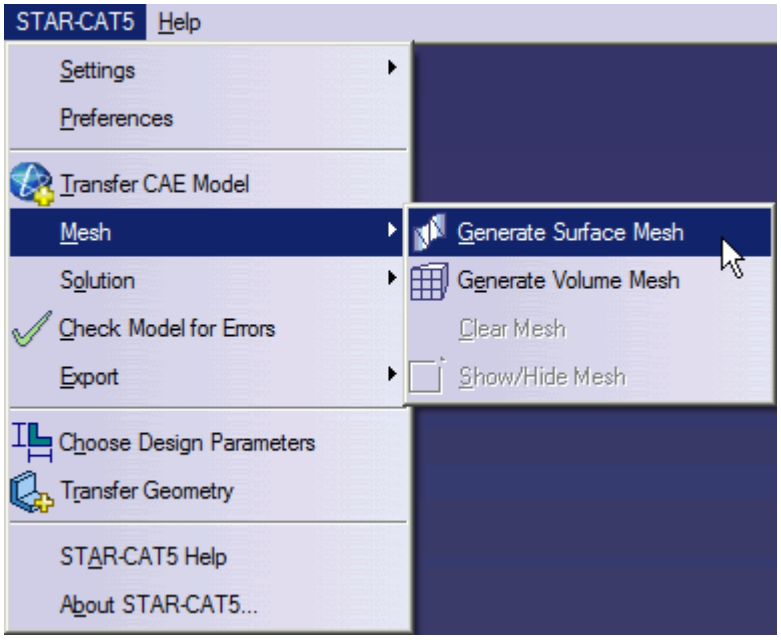
The final STAR-CAT5 object tree is shown below.



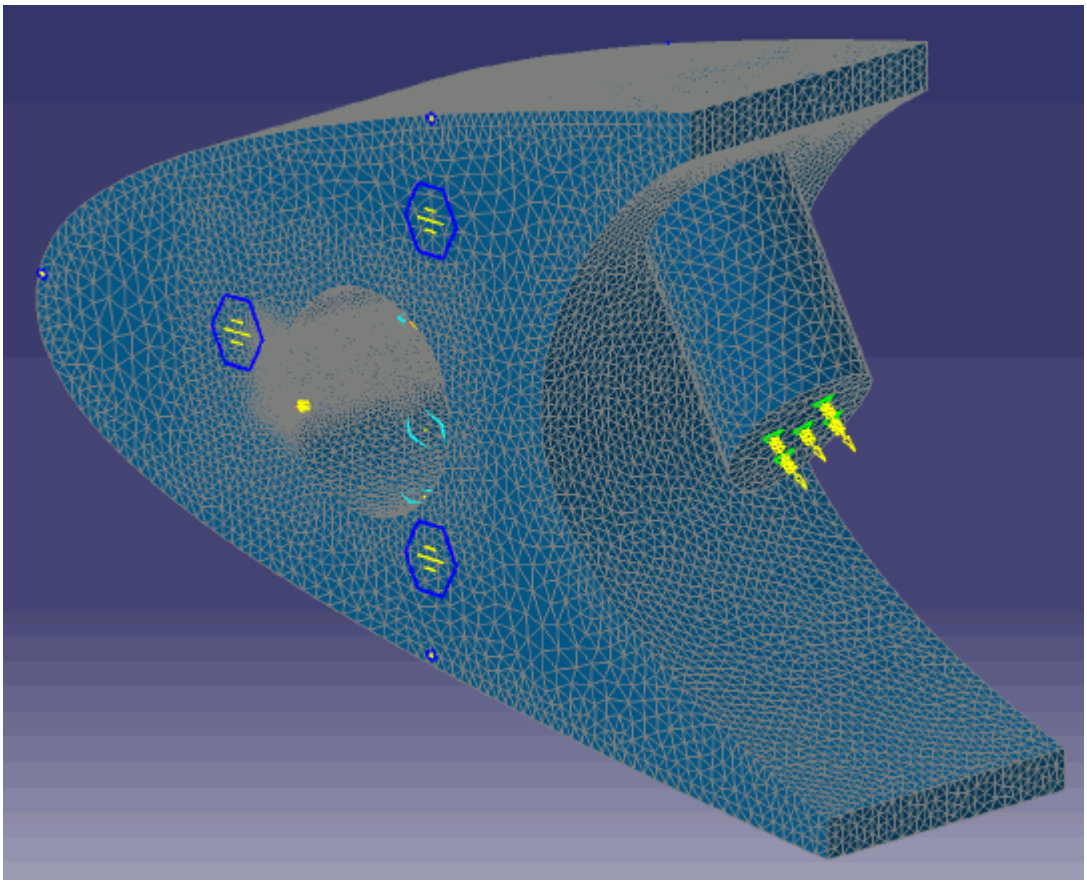
Generating the Surface Mesh

Generate the surface mesh:

1. Select **STAR-CAT5** > **Mesh** > **Generate Surface Mesh**.



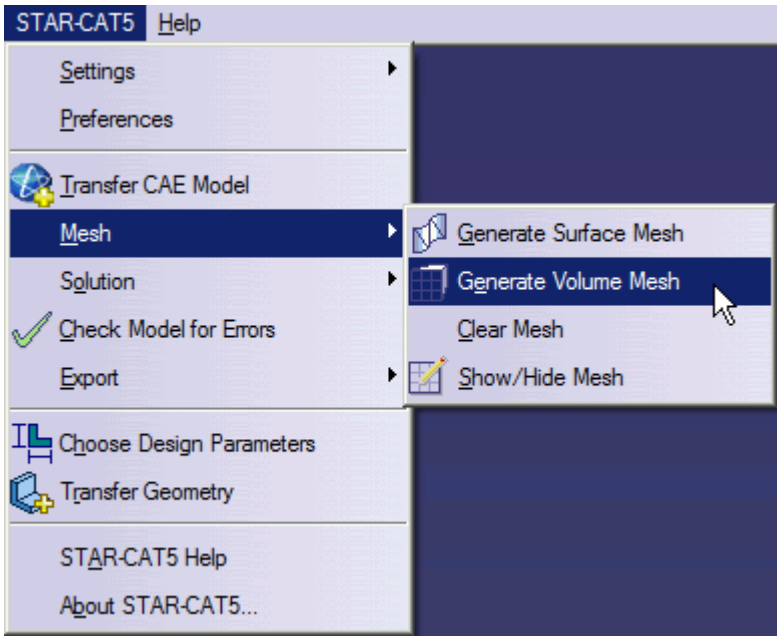
The surface mesh is shown below.



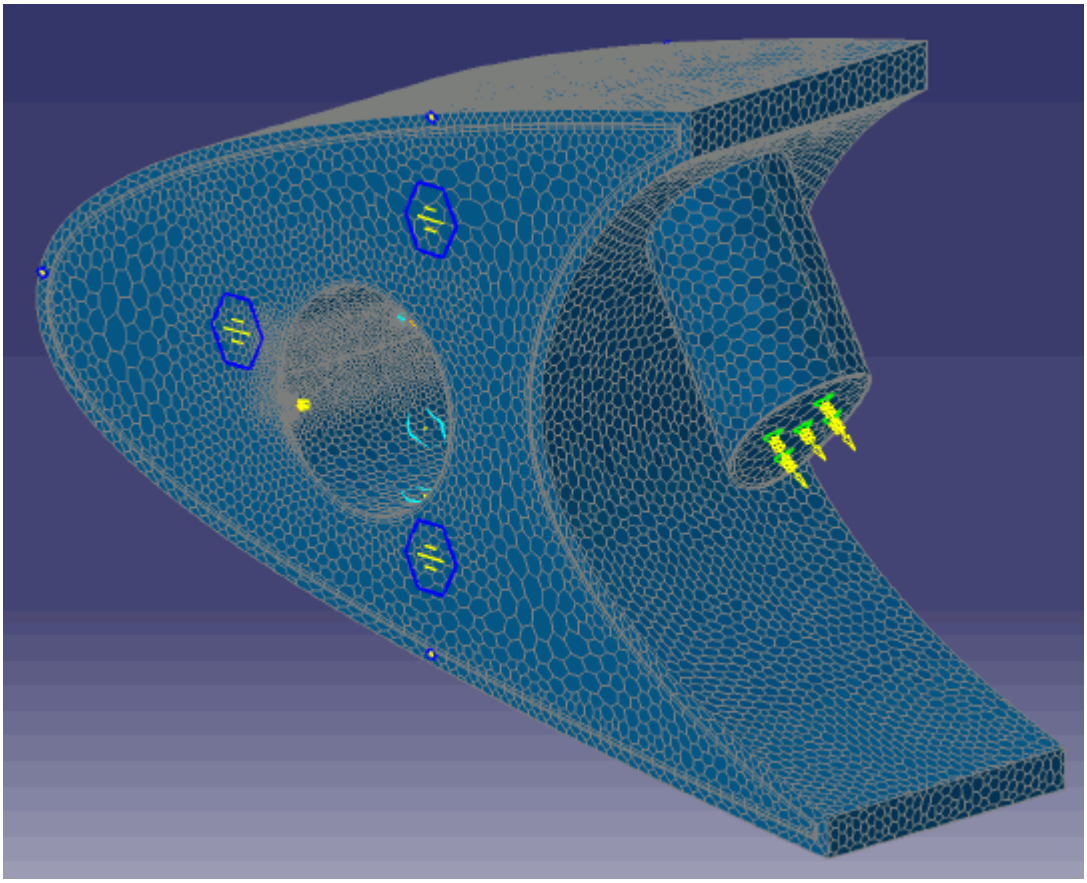
Generating the Volume Mesh

Generate the volume mesh:

1. Select **STAR-CAT5** > **Mesh** > **Generate Volume Mesh**.



The volume mesh is shown below.

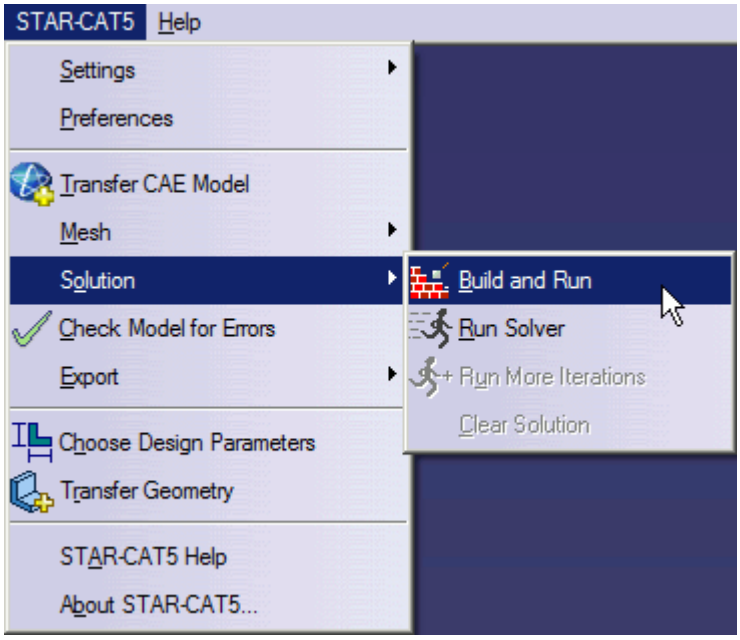


Building and Running

The case is now ready to run.

To run the simulation:

1. Select **STAR-CAT5** > **Solution** > **Build and Run**.



The *Residuals* plot appears in the STAR-CAT5 *Output* window. You can use this plot to monitor the convergence of the solution.

When the solver is done, STAR-CCM+ launches.

Post-Processing Results

You perform all post-processing in STAR-CCM+. By default, a scalar scene, *Scalar Scene 1*, automatically appears in the *Graphics* window when the solver finishes. This scene displays the absolute total pressure.

The results that are contained in the following sections are representative. You may see some variation in the results that you obtain, depending on what platform and package you are using.

1. In STAR-CCM+, click **File** > **Save** to save the simulation.

Contents:

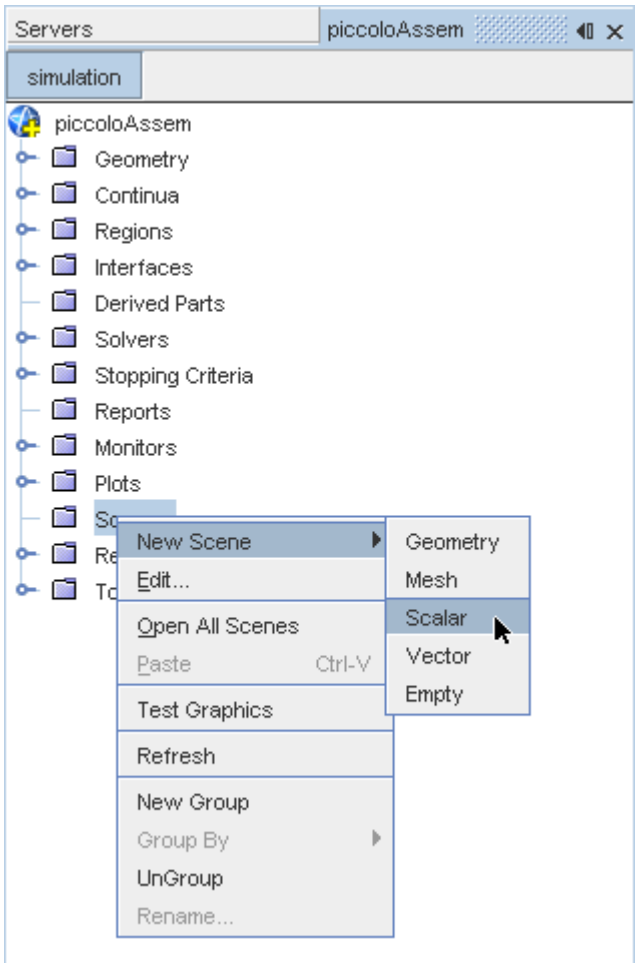
Creating a Scalar Scene
Creating A Streamlines Scene
Creating a Vector Scene
Creating a Streamlines Scene
Creating A Scalar Scene
Visualizing the Wing Skin Temperature
Creating a Streamlines Scene
Creating A Scalar Scene
Creating a Streamlines Scene
Creating a Scalar Scene
Modifying The Scalar Scene
Creating a Streamlines Scene

Creating A Scalar Scene

Create a scalar scene containing a cross-section of the model. Plot temperature on this section to demonstrate how the hot gas coming from the piccolo tube affects the wing skin and the air inside the wing.

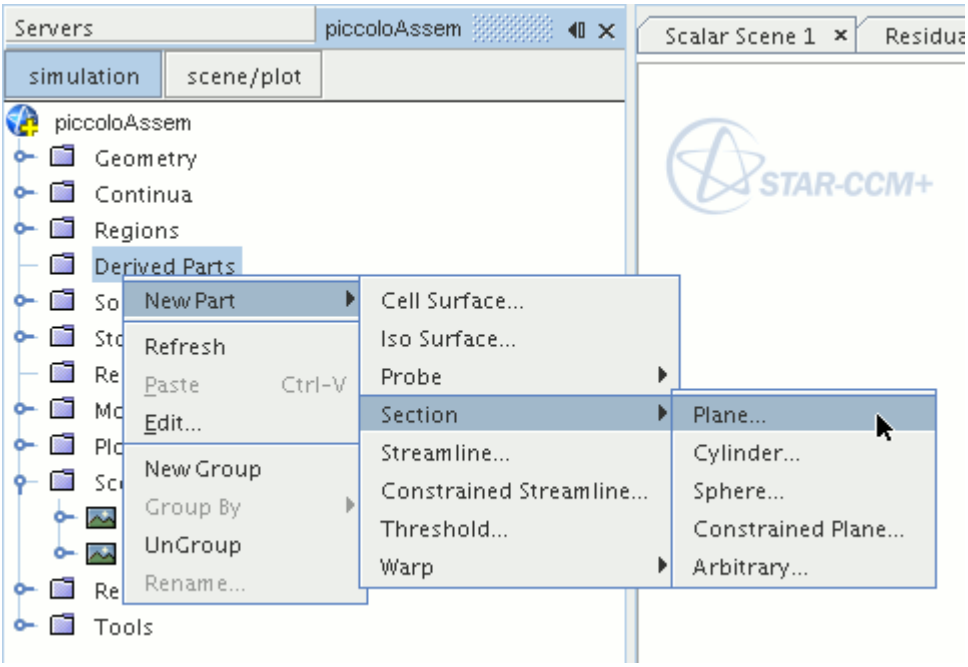
To create the scene:

1. Right-click the **Scenes** node in the explorer pane and select **New Scene** > **Scalar**.



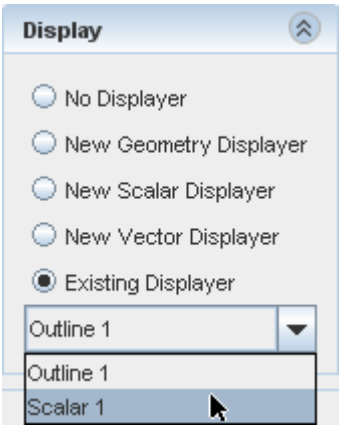
   A new scalar scene, *Scalar Scene 2*, is created and selected in the *Graphics* window.

2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane...**.



The *Create Plane Section* dialog appears.

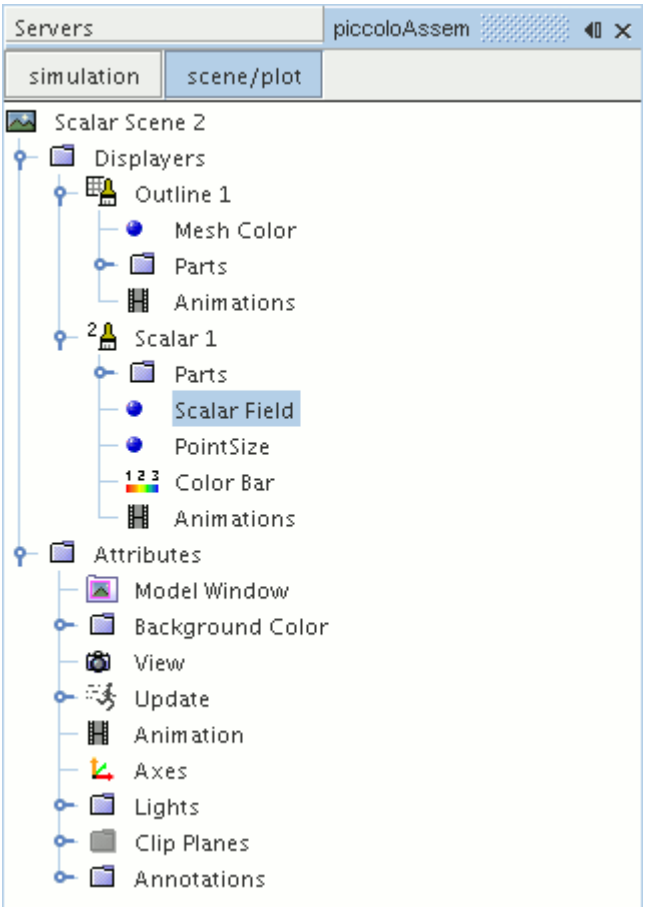
3. Set the origin coordinates to `[0.0 mm, 0.0 mm, 18.4 mm]`.
4. Set the *normal* vector to `[0.0 mm, 0.0 mm, 1.0 mm]`.
5. In the *Display* group box, select **Existing Displayer**, and select **Scalar 1** from the drop-down menu.



6. Click **Create** followed by **Close**.

Change the scalar field to display temperature contours:

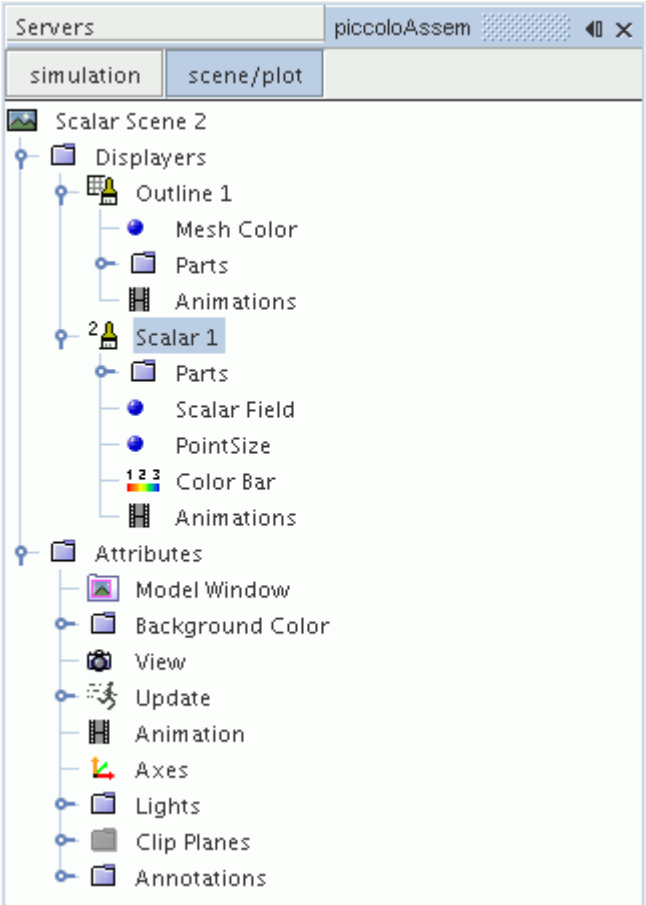
7. Click **scene/plot**.
8. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Scalar Field**.



9. Set *Function* to **Temperature**.

Modify the contour style to have a smooth transition between the different temperature values in the scene:

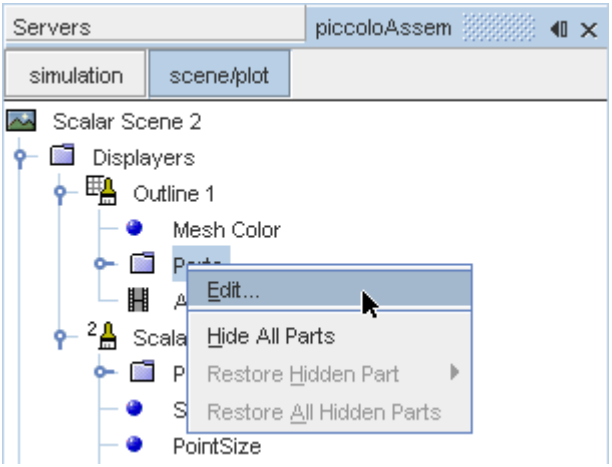
10. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1**.



11. Set *Contour Style* to **Smooth Filled**.

Adjust the outlines in the scene to show only the plane section:

12. Right-click **Scalar Scene 2** > **Displayers** > **Outline 1** > **Parts** and select **Edit...**.

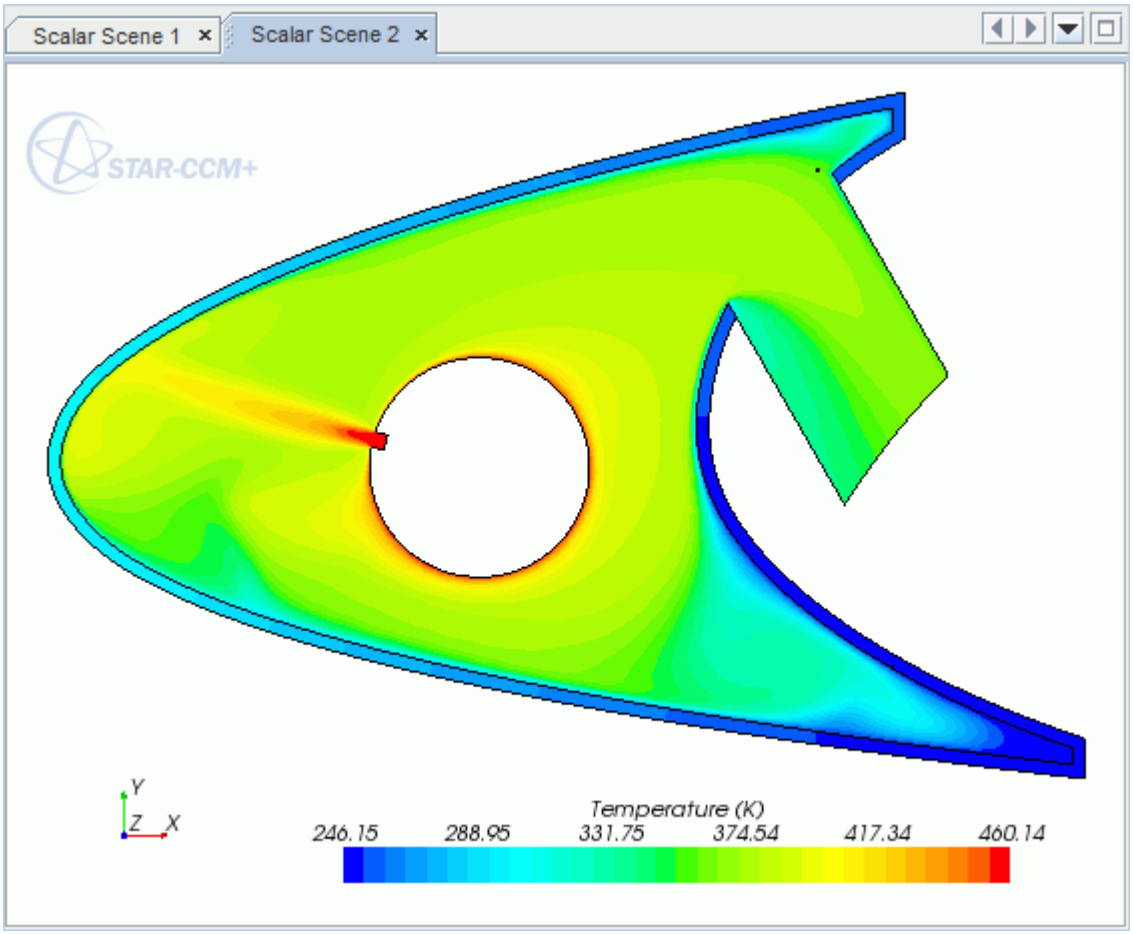


The *Parts* dialog appears.

13. Right-click on a blank area and choose **Deselect All**.

14. Select the **Derived Parts** > **plane section** node.
15. Click **OK**.
16. Position the scene to show the side view of the geometry.
17. Click **simulation**.
18. Click (**Save**) to save the simulation.

The finished scene is shown below.



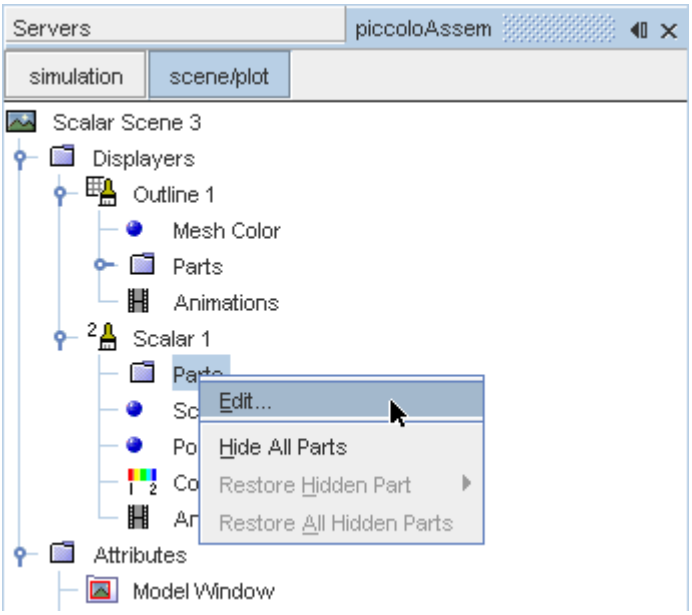
Visualizing the Wing Skin Temperature

Create another scalar scene to display the temperature on the wing skin geometry.

To create another scalar scene:

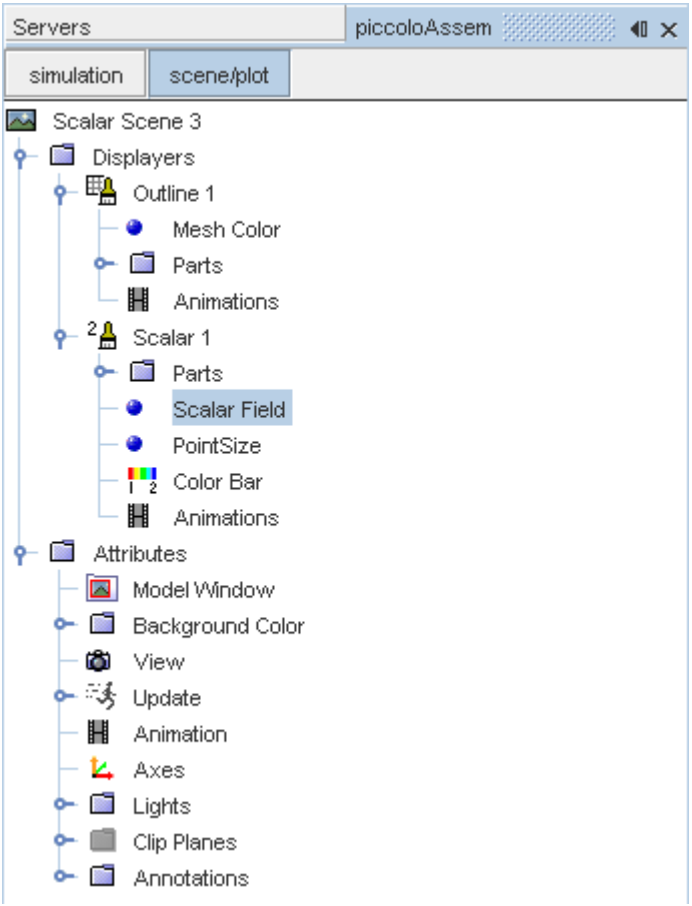
1. Right-click the **Scenes** node in the explorer pane and select **New Scene** > **Scalar**.

   A new scalar scene, *Scalar Scene 3*, is created and selected in the *Graphics* window.
2. Click **scene/plot**.
3. Right-click **Scalar Scene 3** > **Displayers** > **Scalar 1** > **Parts** and select **Edit...**.



The *Parts* dialog appears.

4. Expand the **Regions** node and select the **Leading Edge** node.
5. Click **OK**.
6. Navigate to **Scalar Scene 3** > **Displayers** > **Scalar 1** > **Scalar Field**.



7. Set *Function* to **Temperature.**

To have a smooth transition between the different temperature values, edit the contour style:

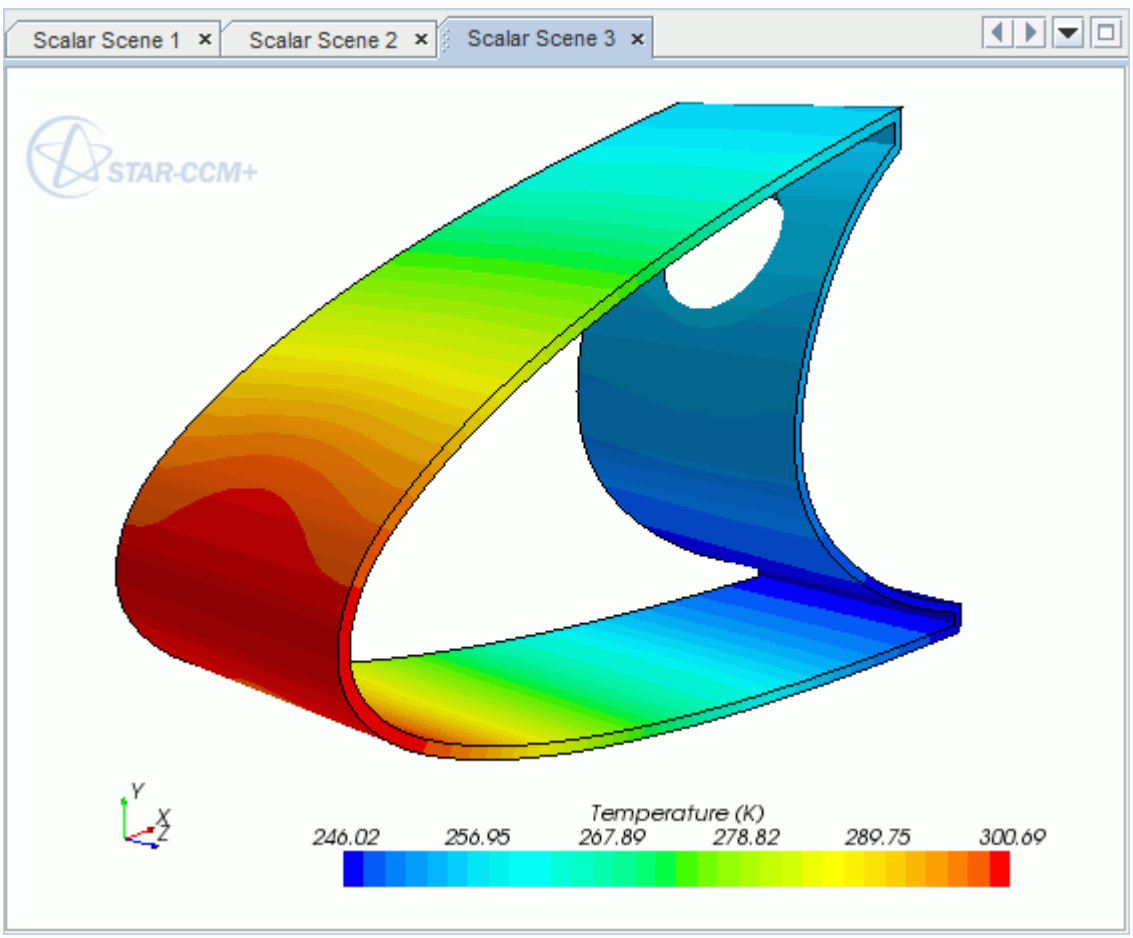
8. Navigate to **Scalar Scene 3** > **Displayers** > **Scalar 1**.
9. Set *Contour Style* to **Smooth Filled**.

Adjust the outlines to appear on the solid region only:

10. Right-click **Scalar Scene 3** > **Displayers** > **Outline 1** > **Parts** and select **Edit...**.

    The *Parts* dialog appears.
11. Expand the **Regions** node and deselect the **Internal Air** node.
12. Click **OK**.
13. Reorient the geometry to get a better view of the leading edge.
14. Click **simulation**.
15. Save the simulation.

The finished scene is shown below.

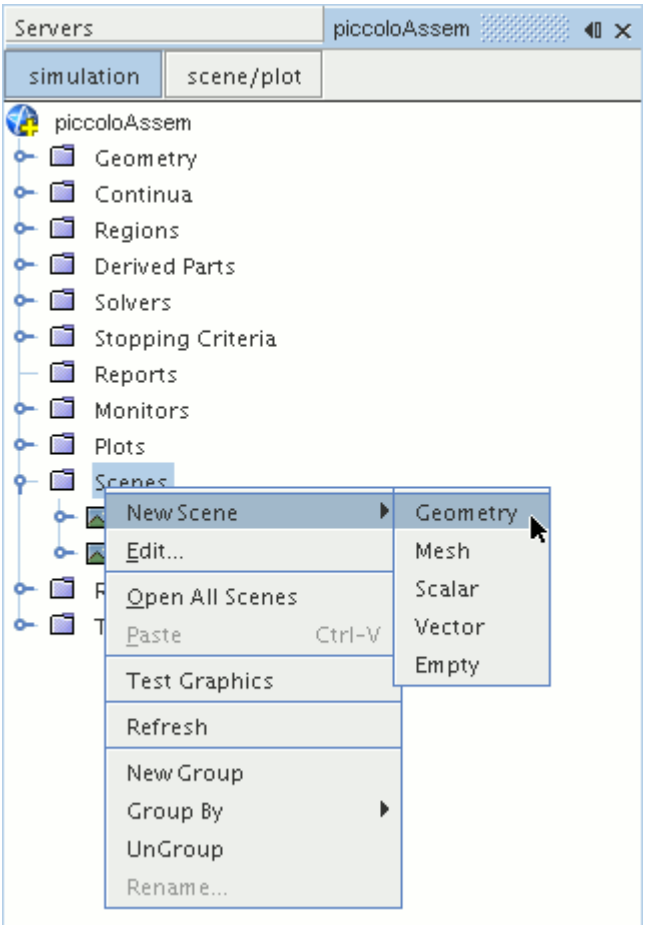


Creating a Streamlines Scene

Create streamlines to get a better understanding of how the hot air flows inside the leading edge of the wing.

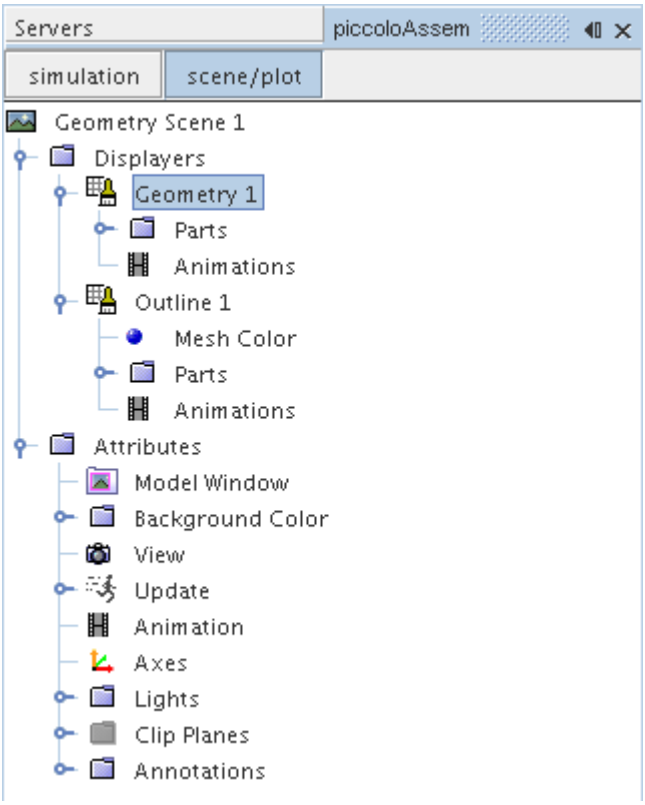
First, create a geometry scene:

1. Right-click the **Scenes** node in the object tree and select **New Scene** > **Geometry**.



   A new geometry scene, *Geometry Scene 1*, is created and selected in the *Graphics* window.

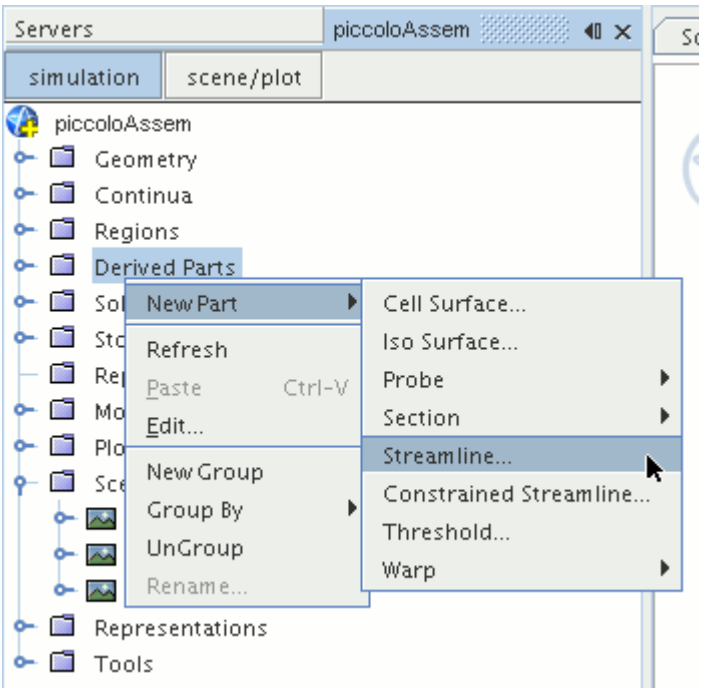
2. Click **scene/plot**.
3. Navigate to the **Geometry Scene 1** > **Displayers** > **Geometry 1** node.



4. Set *Opacity* to `0.1`.

Next, create the streamlines:

5. Click **simulation**.
6. Right-click the **Derived Parts** node and select **New Part** > **Streamline...**.

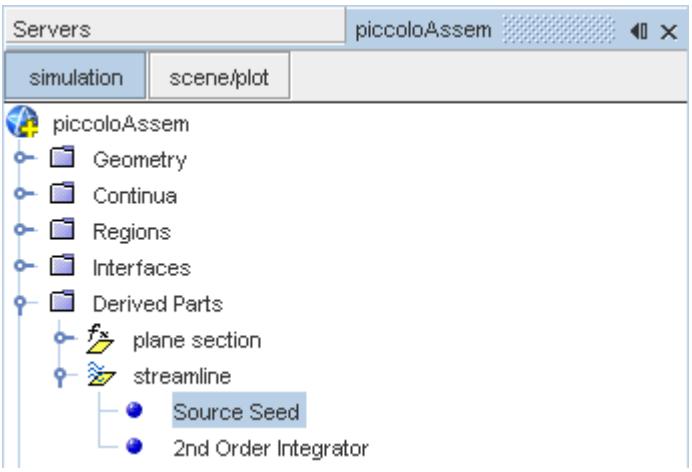


The *Create Streamline* dialog appears.

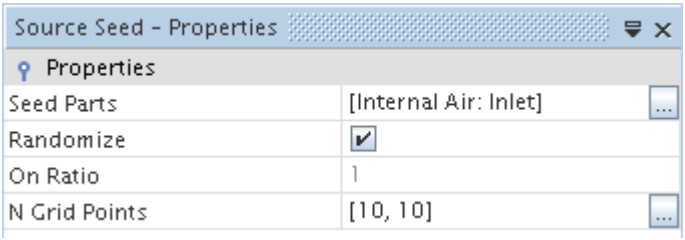
7. In the *Input Parts* box, click **Select...**.
8. In the *Select Objects* dialog, expand the **Regions** node and deselect **Leading Edge**.
9. Click **OK** to close the *Select Objects* dialog.
10. Make sure that **Part Seed** is selected in the *Seed Mode* section.
11. Set *Seed Parts* to **Regions** > **Internal Air** > **Boundaries** > **Inlet**.
12. Set *Part U-Resolution* to `10`.
13. Set *Part V-Resolution* to `10`.
14. Make sure that **New Streamline Displayer** is selected in the *Display* section.
15. Click **Create** followed by **Close**.

The streamlines do not yet appear in the geometry scene. Follow the steps that are outlined below to modify the streamlines:

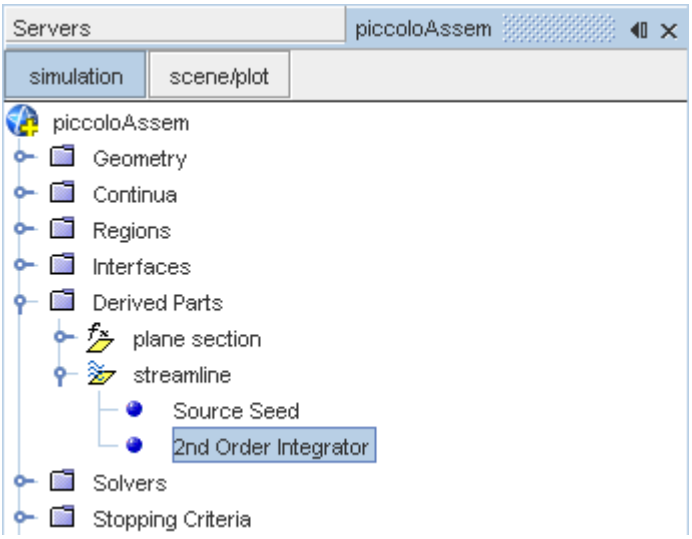
16. Navigate to **Derived Parts** > **Streamline** > **Source Seed**.



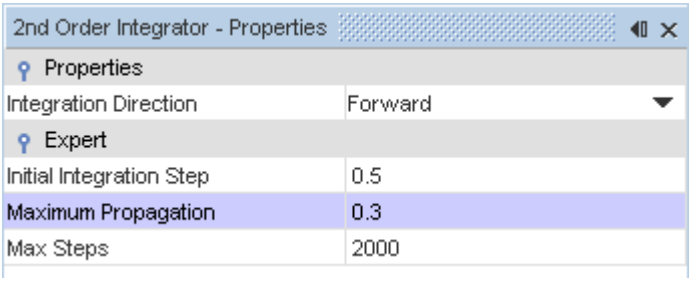
17. Activate the *Randomize* option.



18. Navigate to **Derived Parts** > **Streamline** > **2nd Order Integrator**.



19. Set *Maximum Propagation* to `0.3`.



Adjust the appearance of the streamlines to display tubes instead of the default lines:

20. Click **scene/plot**.
21. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1**.
22. Set *Mode* to **Tubes** and *Width* to `0.00025`.

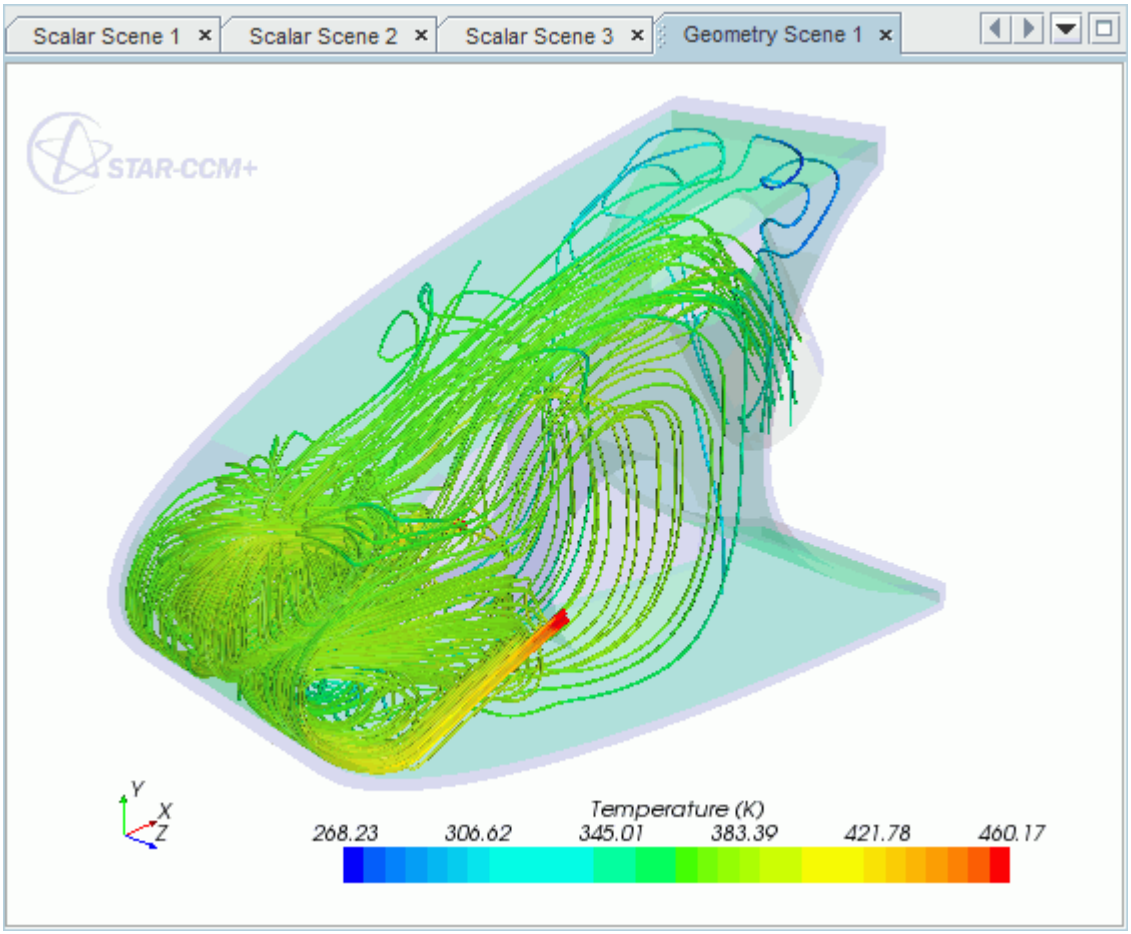
Adjust the scalar field of the streamlines to display temperature:

23. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** > **Scalar Field**.
24. Set *Function* to **Temperature**.

Turn off the outlines in the geometry scene to have a better view of the streamlines.

25. Right-click the **Geometry Scene 1** > **Displayers** > **Outline 1** and select **Toggle Visibility**.
26. Reorient the geometry to get a better view of the leading edge.
27. Save the simulation.

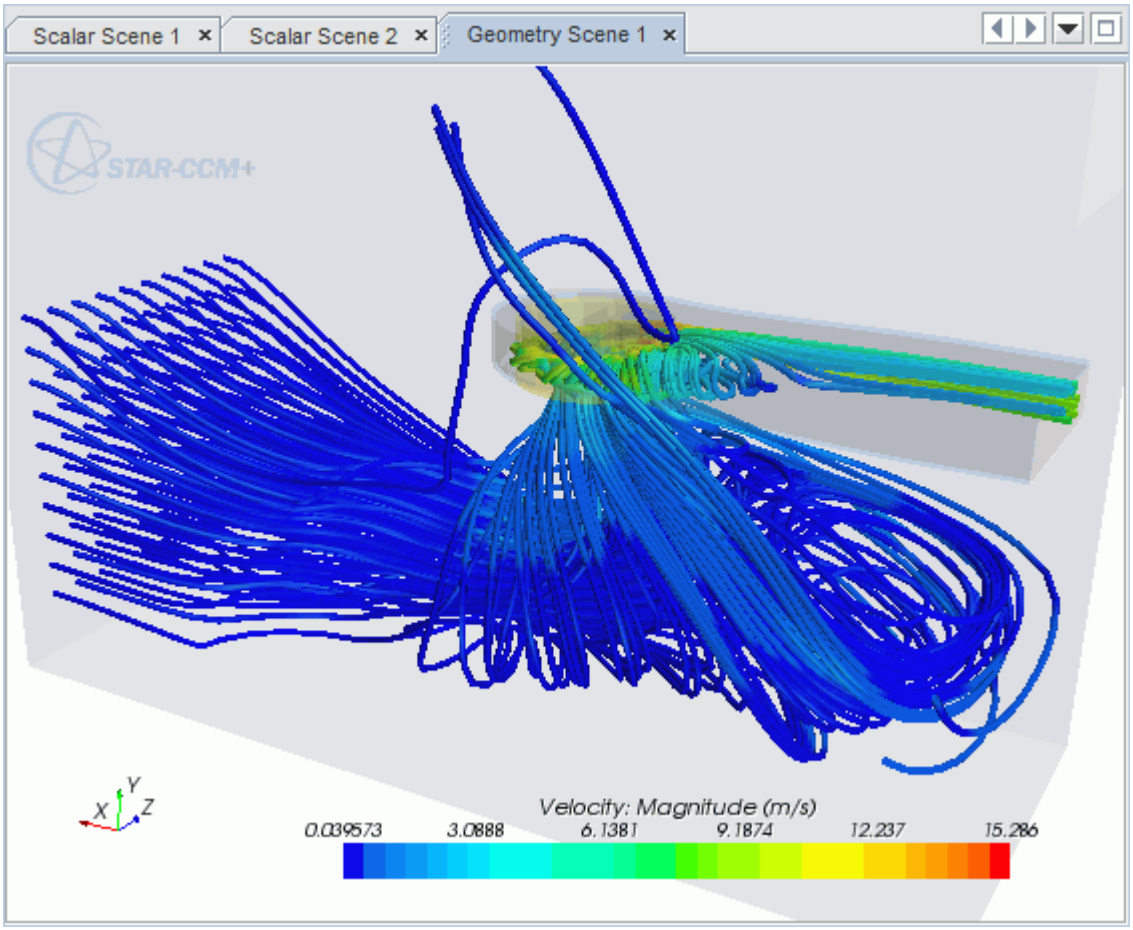
The finished scene is shown below.


Scalar Scene 1
Scalar Scene 2
Scalar Scene 3
Geometry Scene 1
STAR-CCM+
Y
X
Z
Temperature (K)
268.23
306.62
345.01
383.39
421.78
460.17

Rotating Flow Inside a Graphics Card

This tutorial takes you through the steps that are required to run a CFD calculation in the STAR-CAT5 environment.

For this tutorial, the model is a fully sealed graphics card that is located inside an empty, idealized PC tower case. The geometry that is used in this tutorial has been simplified on order to decrease the computational cost. As a consequence the mesh is relatively coarse at approximately 200,000 cells compared to the 3-4 million cells used for an industrial analysis.



Contents:

Opening the CAD Model
Selecting Fluid Physics Models
Setting the Rotating Region
Defining Boundary Conditions
Setting the Mesh Pipeline
Creating the Volumetric Mesh Controls
Modifying Solution Controls

Reviewing the Simulation Set Up
Generating The Surface Mesh
Generating the Volume Mesh
Building and Running
Post-Processing Results

Opening the CAD Model

A directory containing the CAD geometry is provided with the STAR-CCM+ distribution.

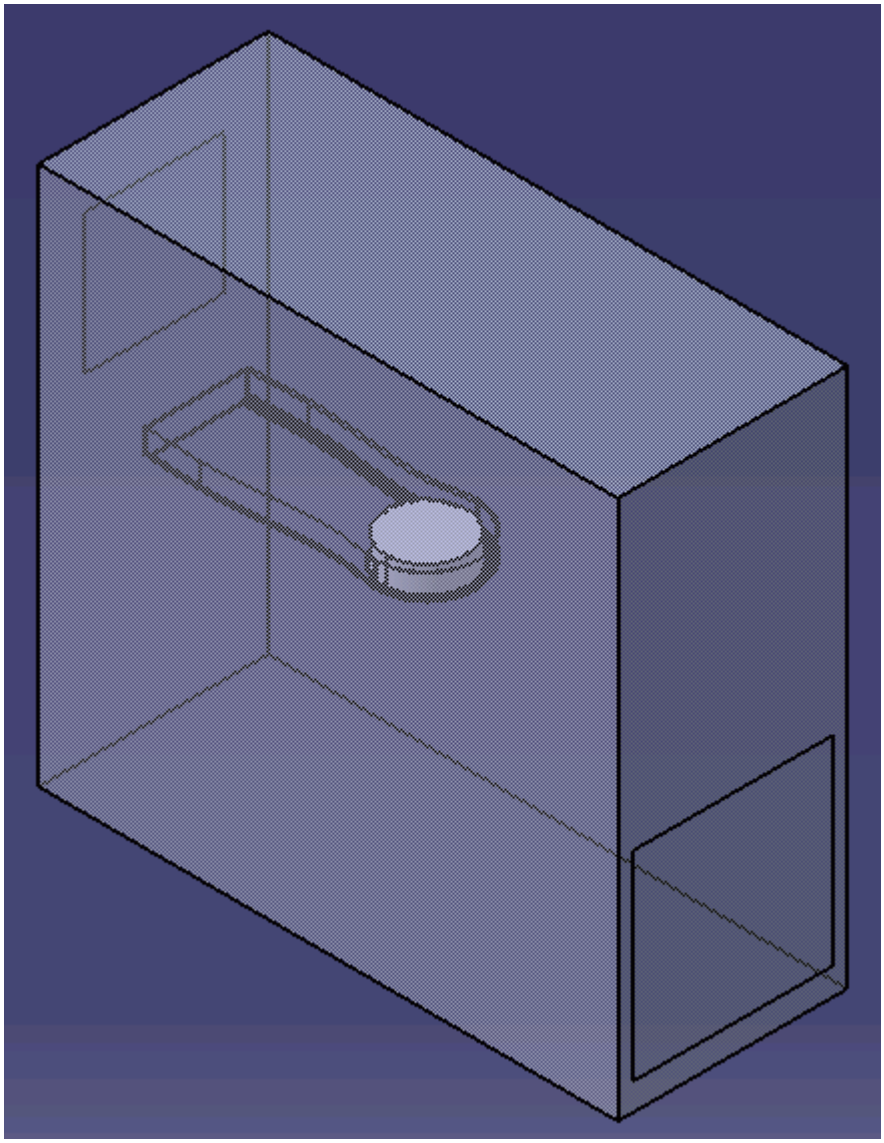
This directory is located in `[STAR-CCM+_INSTALL_DIR]\doc\startutorialsdata\starcat5\data`.

1. Copy directory `data` to your own disk space and open the part file `gpuAssem.CATProduct` in STAR-CAT5.

   See Launching STAR-CAT5.

   The assembly is made up of two parts: the PC tower case (including the GPU housing), and the rotating region that encloses the blower fan of the GPU.

2. Position the geometry as shown below.



The STAR-CAT5 menu appears at the top of the screen.The STAR-CAT5 object tree is located on the left of the screen.

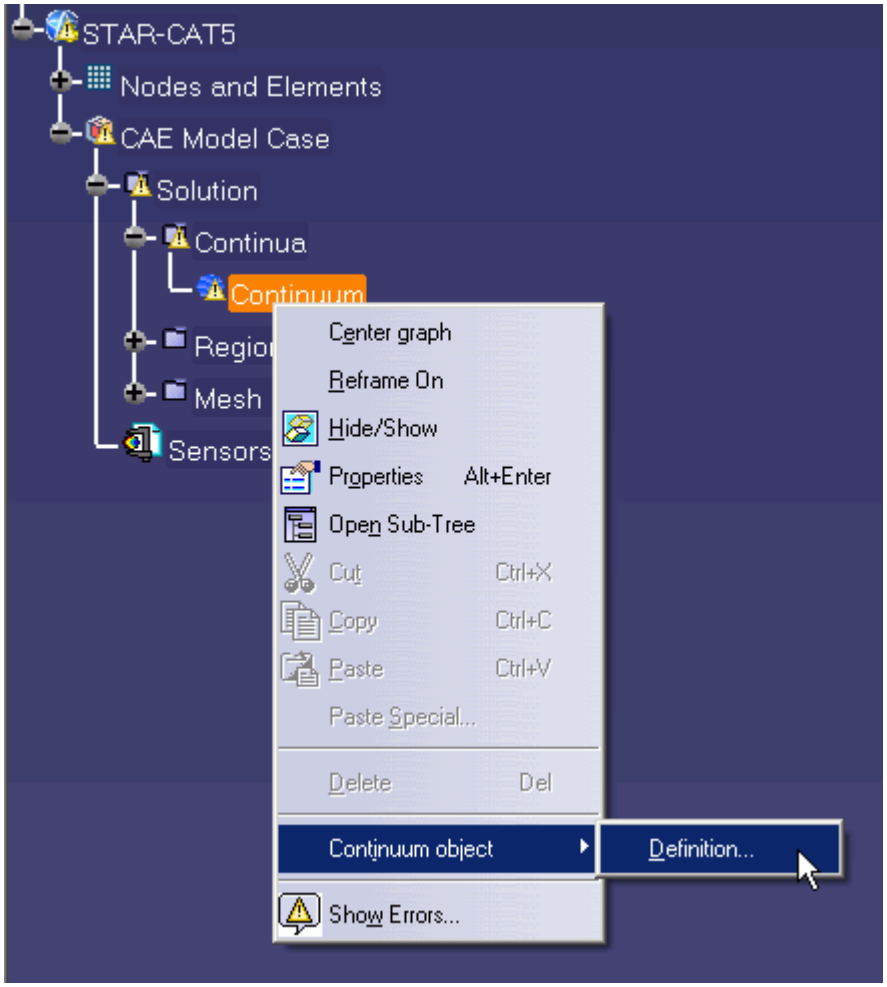
3. Save the analysis as `gpuAssem.CATAnalysis`.

Selecting Fluid Physics Models

The first step in setting up this tutorial is to define the physics of the problem. For this case, the flow is turbulent and isothermal. Temperature calculations are on by default in STAR-CAT5.

To turn temperature calculations off:

1. Right-click the **Continua** > **Continuum** node in the STAR-CAT5 tree and select **Continuum object** > **Definition...**.



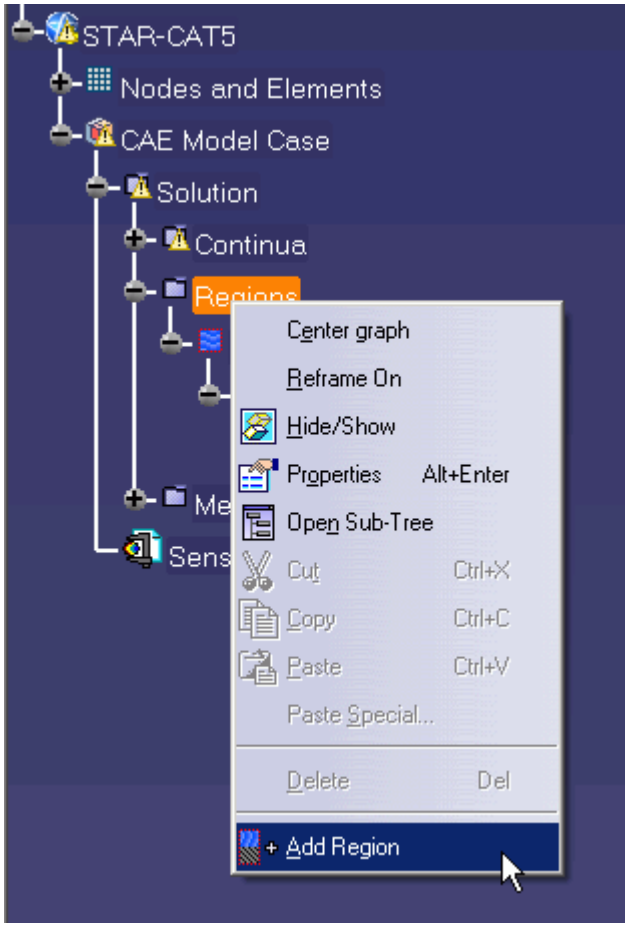
The *Continuum Settings* dialog appears.

2. Deactivate the **Temperature** option in the *Optional Physics Models* box.
3. Click **OK**.

Setting the Rotating Region

Create the rotating region that contains the GPU fan:

1. Right-click the **Regions** node in the STAR-CAT5 object tree and select **Add Region**.

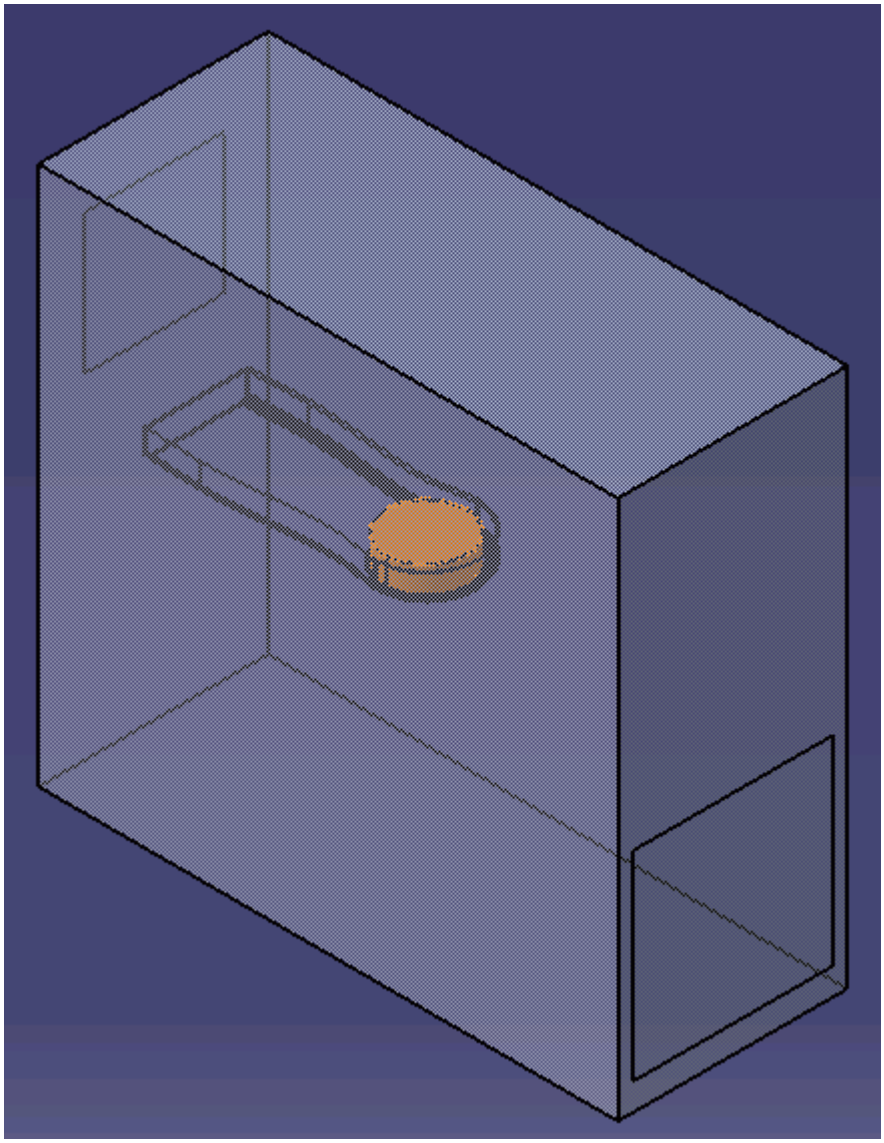


The *Region Settings* dialog appears.

2. Select the disc that is located inside the graphics card housing.

   You can select this part by expanding the **Links Manager.1** node in the Analysis Manager tree and selecting **Link.1** > **PC Tower and Graphics Card Assembly** > **gpuFan (Fan)**.

   Once selected, the part becomes highlighted as shown below.



3. Set *Name* to `Rotating Region`.
4. Activate the **Moving Reference Frame** option.
5. Set *Omega* to `-2800 rpm`.

Note: Make sure that you set the units to `rpm` using the drop-down menu before entering the value.

6. Set the origin of the axis of rotation to:

Property	Value
X	`-122.1761 mm`
Y	`-1.9434 mm`
Z	`-19.0609 mm`

7. Set the components of the axis of rotation to:

Property	Value
i (*i component*)	0
j (*j component*)	1
k (*k component*)	0

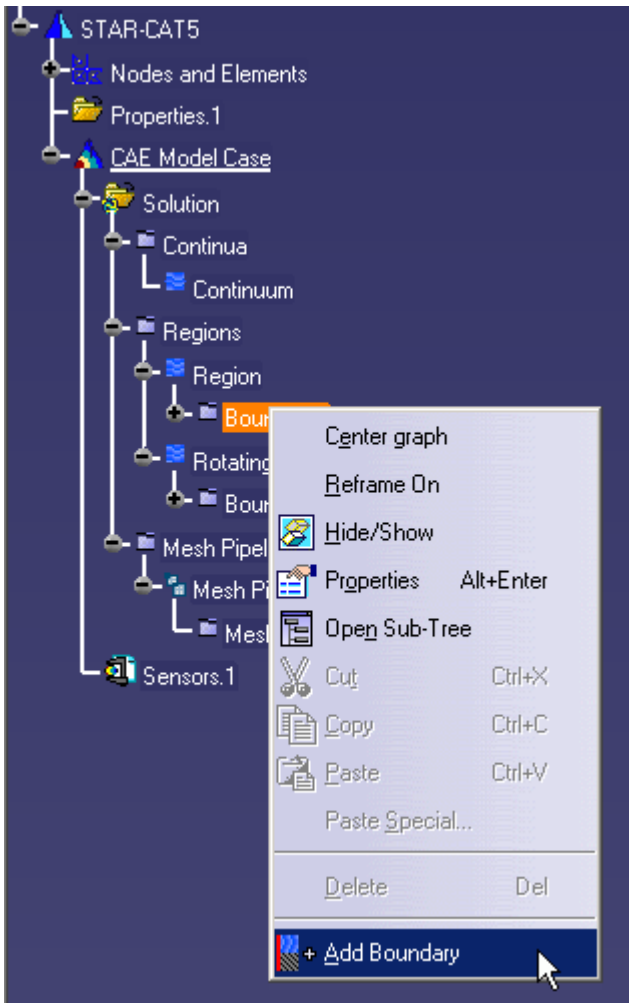
8. Make sure that the *Use Assembly Coordinate System* is activated.
9. Click **OK**.

Defining Boundary Conditions

Create the inlet and outlet boundary conditions for this simulation.

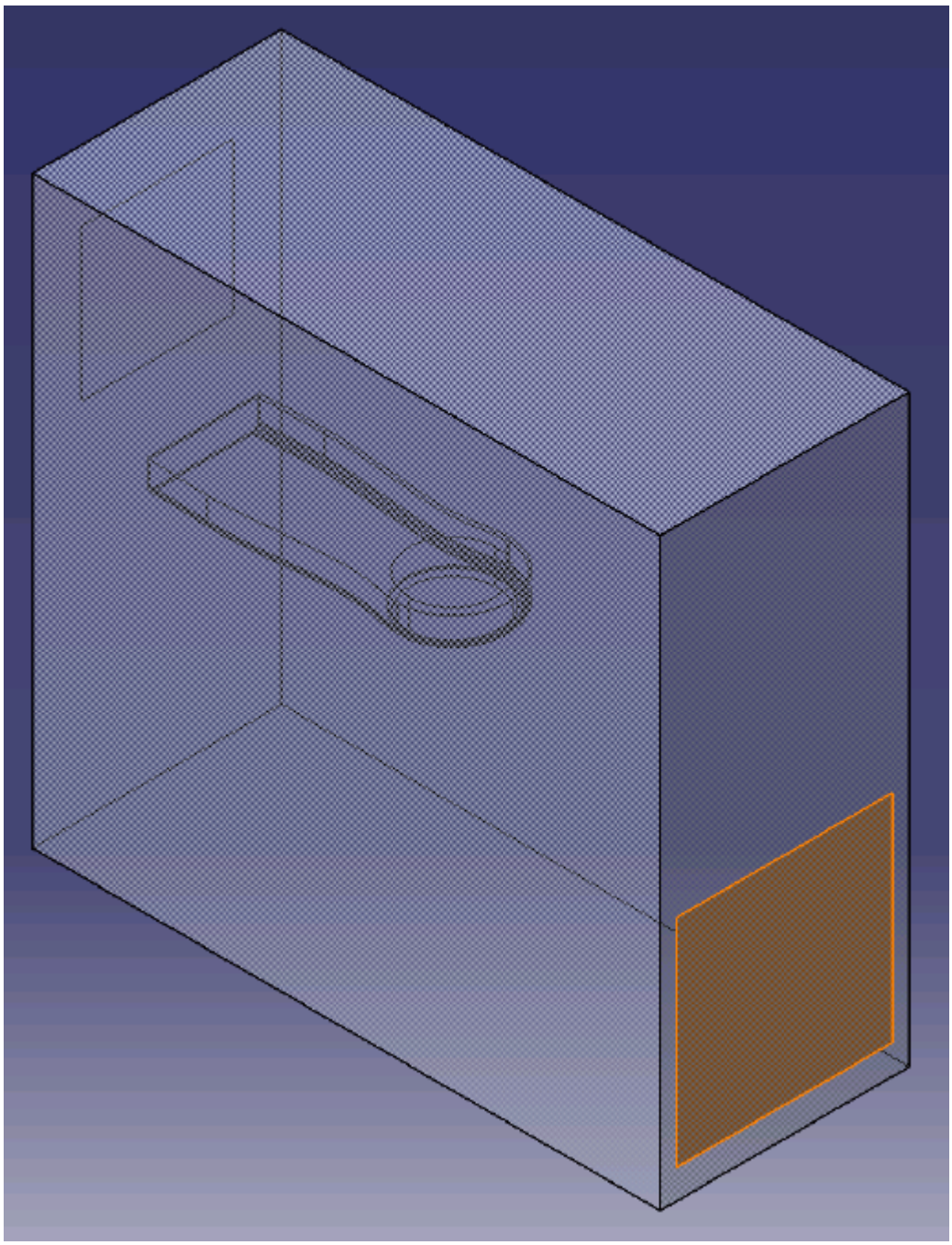
Create the inlet boundary first:

1. Right-click the **Regions** > **Region** > **Boundaries** node and select **Add Boundary**.



The *Boundary Settings* dialog appears.

2. Select the face that is highlighted below.



These selections are reported in the *Supports* box in the *Boundary Settings* dialog.

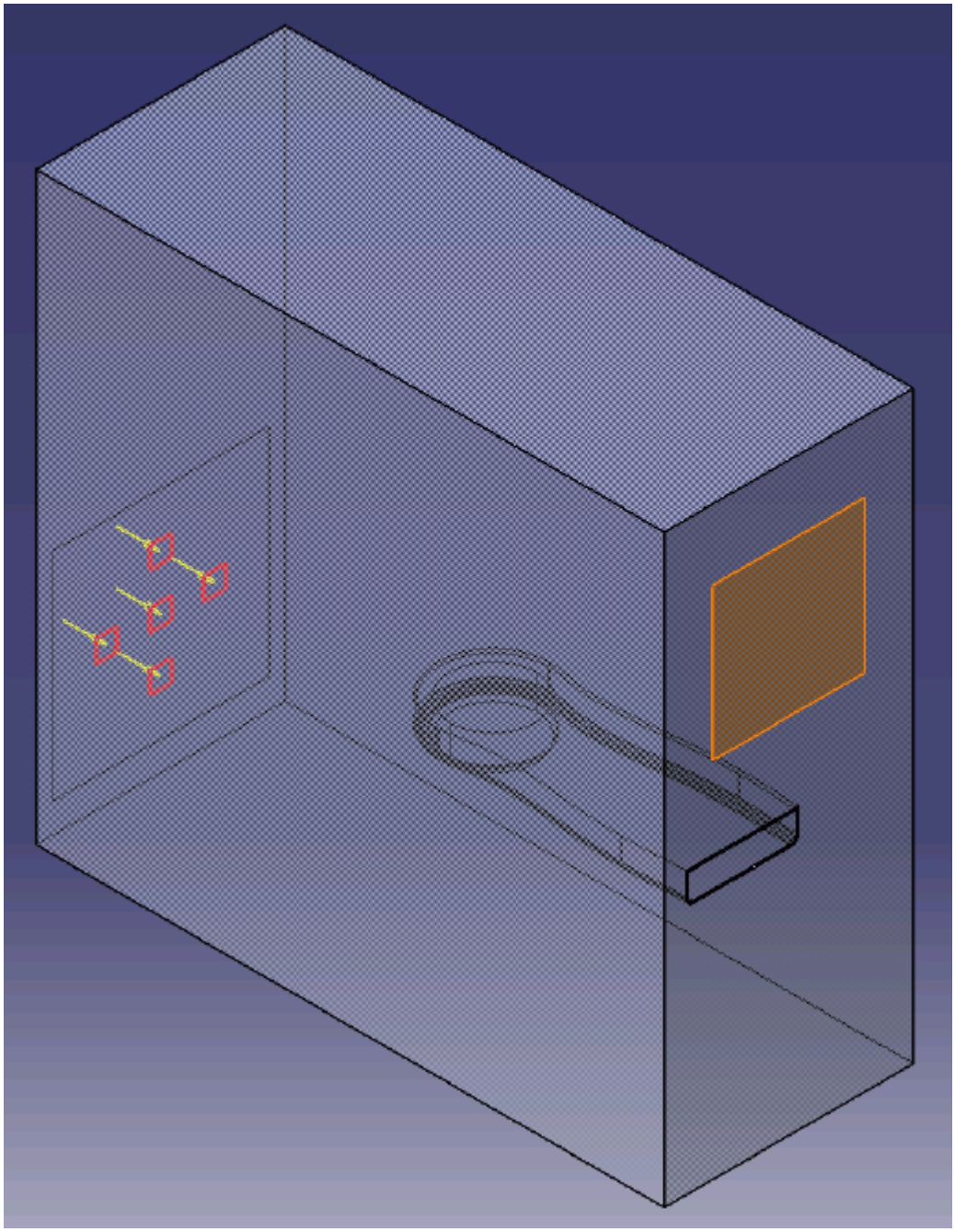
3. Set the following settings:

Parameter	Value
Boundary Type	**Velocity Inlet**
Name	`Case Inlet`
Normal Velocity	`0.1 m/s`

4. Click **OK**.

Create the case outlet boundary:

5. Using the same technique, create another boundary.
6. Select the face that is highlighted below.



7. Set the following parameters:

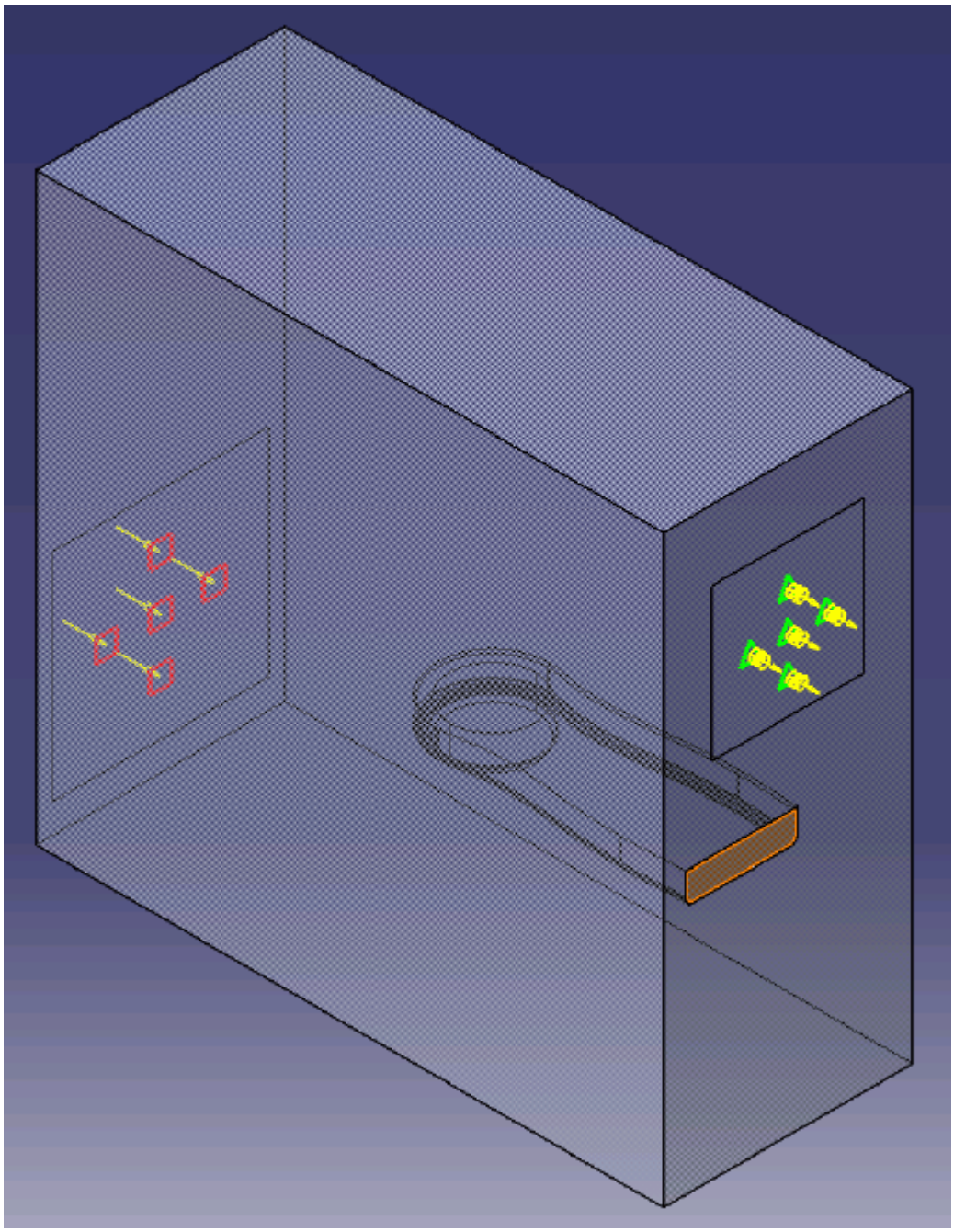
Parameter	Value
Boundary Type	**Pressure Outlet**

Parameter	Value
Name	`Case Outlet`

8. Click **OK**.

Create the GPU outlet boundary:

9. Create another boundary and select the following face.



10. Set the following parameters:

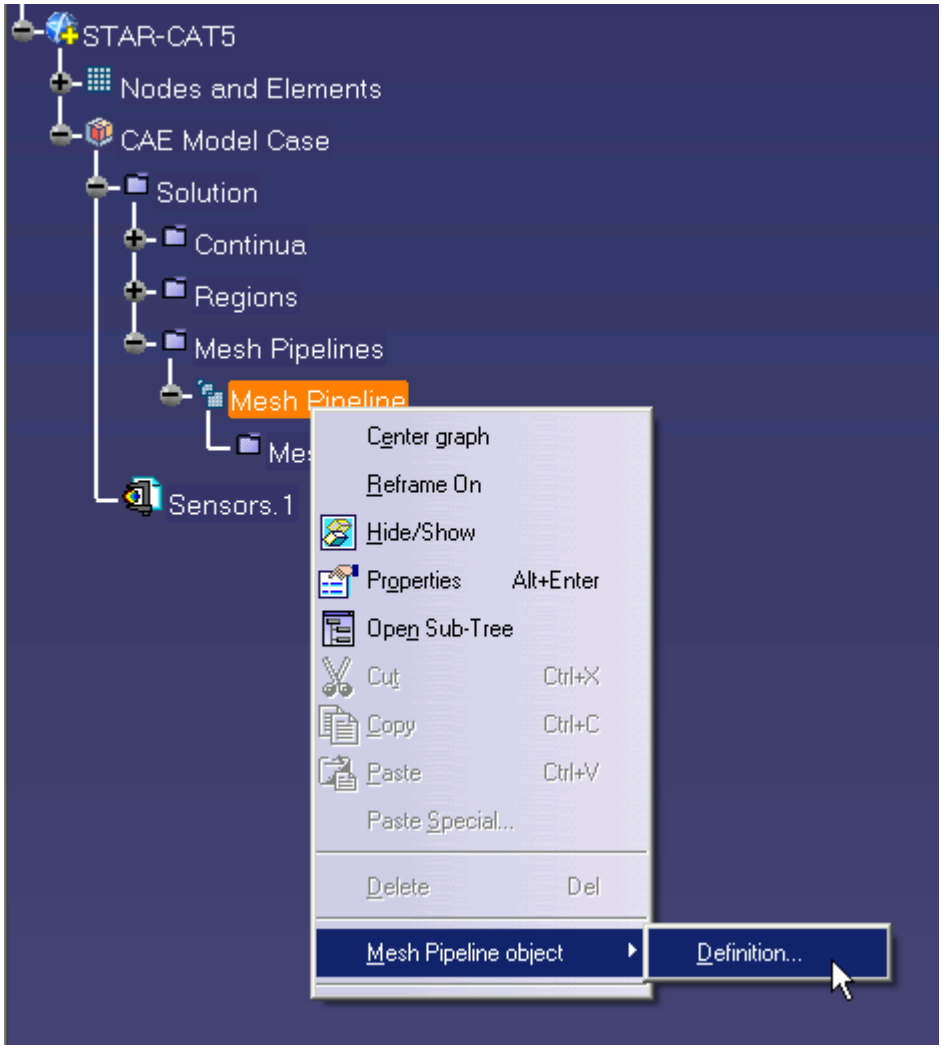
Parameter	Value
Boundary Type	**Pressure Outlet**
Name	`GPU Outlet`

11. Click **OK**.
12. Save your progress.

Setting the Mesh Pipeline

To obtain more accurate results, modify the mesh pipeline:

1. Right-click the **Mesh Pipelines** > **Mesh Pipeline** node in the STAR-CAT5 tree and select **Mesh Pipeline object** > **Definition**.



The *Mesh Pipeline Settings* dialog appears.

2. Set the following parameters:

Parameter	Value
Base Size	
Use Absolute Value	**Activated**
Base Size	`0.016 m`
Prism Layer	
Prism Layer Thickness	`20 %`

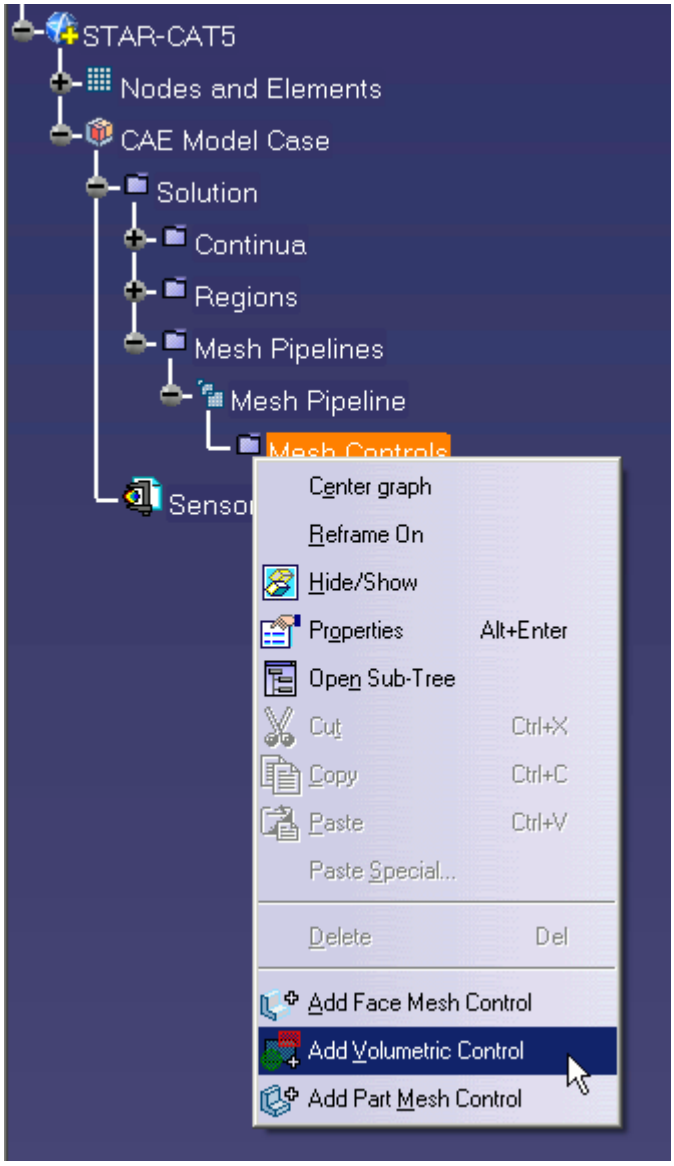
3. Click **OK**.

Creating the Volumetric Mesh Controls

Use volumetric mesh control regions to capture the finer details of the geometry model. Create three regions in total: a small region that closely surrounds the rotating region containing the fan, a region that encloses the GPU case, and a region that acts as an intermediary between the base size and the other volumetric control regions.

Create the volumetric mesh control for the rotating region:

1. Right-click the **Mesh Pipelines** > **Mesh Pipeline** > **Mesh Controls** node in the STAR-CAT5 tree and select **Add Volumetric Control**.



The *Volumetric Control* dialog appears.

2. Set *Name* to `Fan Box`.
3. Activate the *Use Absolute Value* option.
4. Set *Surface Mesh Size* to `0.002 m`.
5. Enter the following coordinates for Point 1:

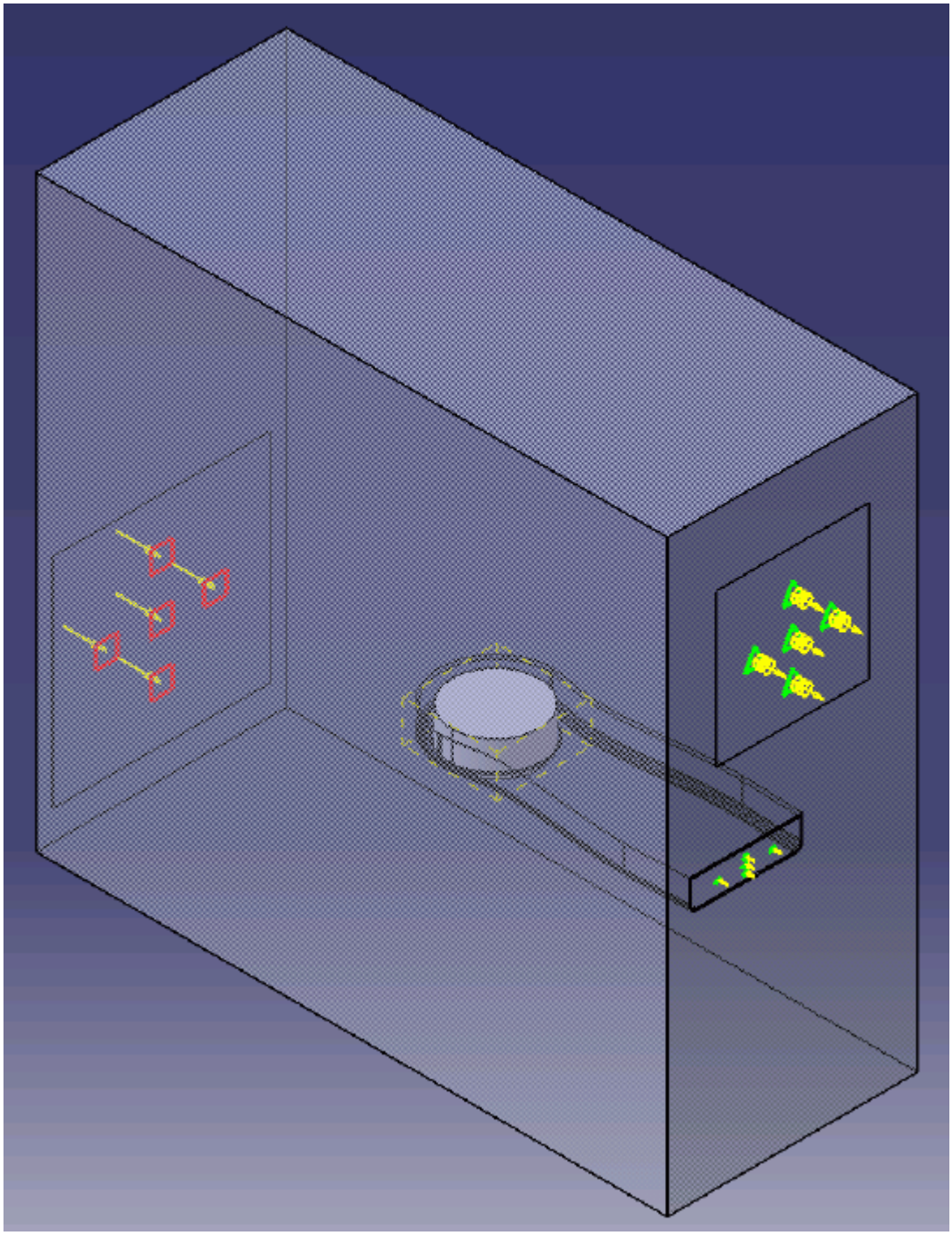
Coordinate	Value
X1	`-166.00 mm`
Y1	`4.00 mm`
Z1	`-61.00 mm`

6. Enter the following coordinates for Point 2:

Coordinate	Value
X2	-79.00 mm
Y2	-34.00 mm
Z2	25.00 mm

7. Click **Create**.

   The volumetric control region is shown below.



8. Click **OK**.

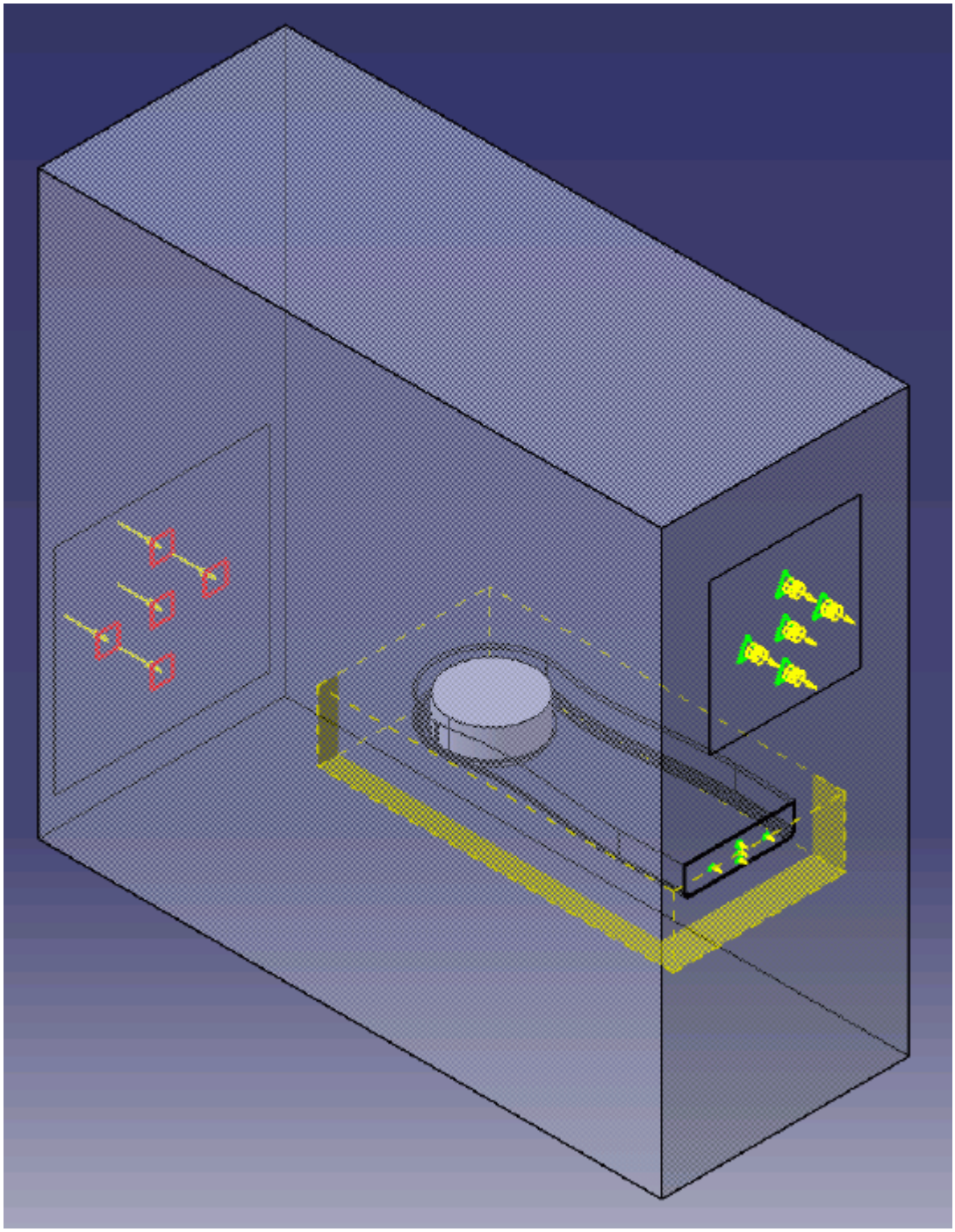
Create the volumetric mesh control for the GPU region:

9. Using the same technique, create another volumetric mesh control with the following settings:

Parameter	Value
Name	`GPU Box`
Surface Mesh Size	
Use Absolute Value	**Activated**
Surface Mesh Size	`0.005 m`
Shape	
X1	`-383.00 mm`
Y1	`16.00 mm`
Z1	`-111.00 mm`
X2	`-52.00 mm`
Y2	`-49.00 mm`
Z2	`49.00 mm`

10. Click **Create**.

    The volumetric control region is shown below.



11. Click **OK**.

Create the volumetric mesh control for the PC tower case:

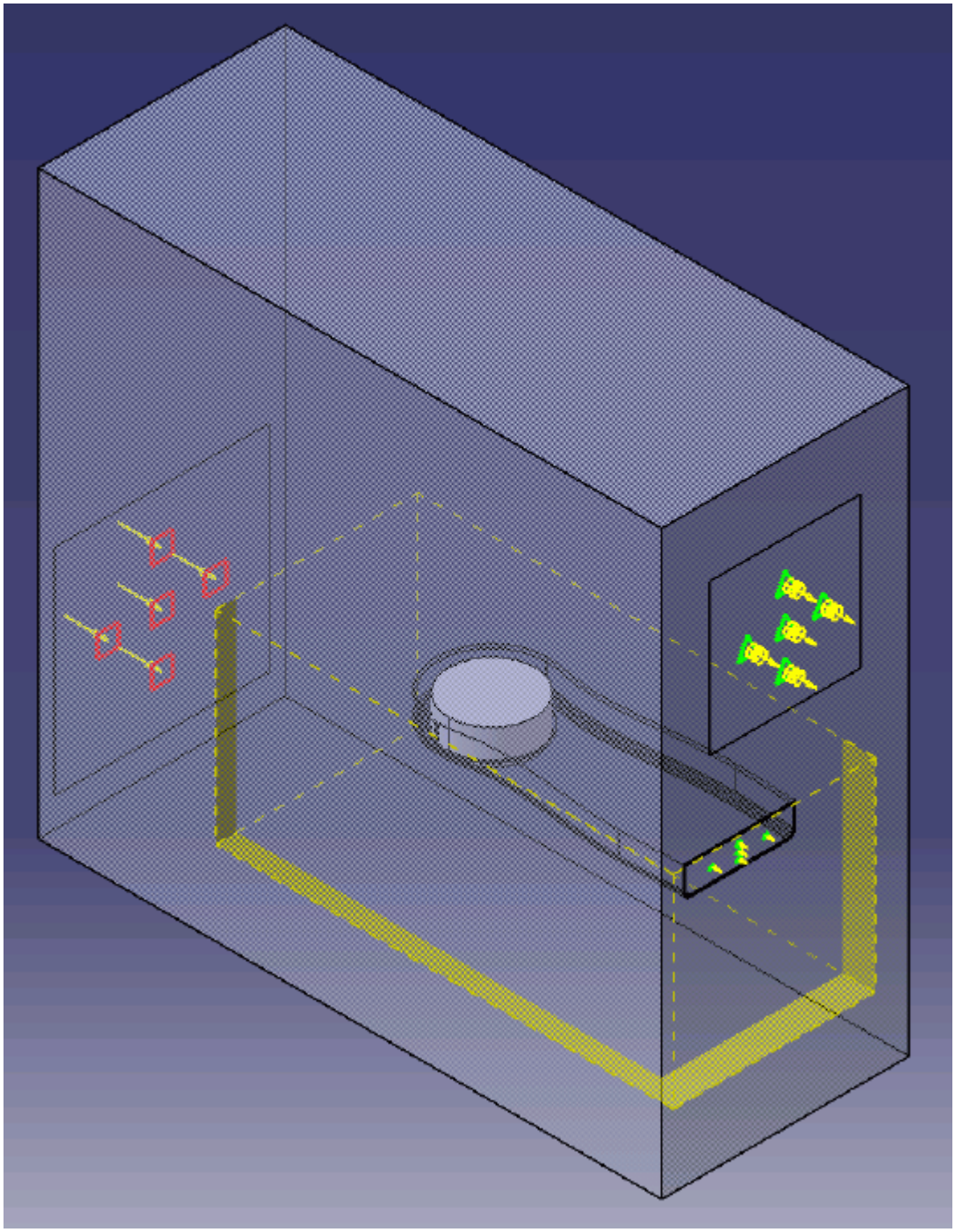
12. Using the same technique, create another volumetric mesh control with the following settings:

Parameter	Value
Name	`Case Box`
Surface Mesh Size	

Parameter	Value
Use Absolute Value	**Activated**
Surface Mesh Size	`0.009 m`
Shape	
X1	`-383.00 mm`
Y1	`30.00 mm`
Z1	`-111.00 mm`
X2	`43.00 mm`
Y2	`-159.00 mm`
Z2	`77.00 mm`

13. Click **Create**.

    The volumetric control region is shown below.

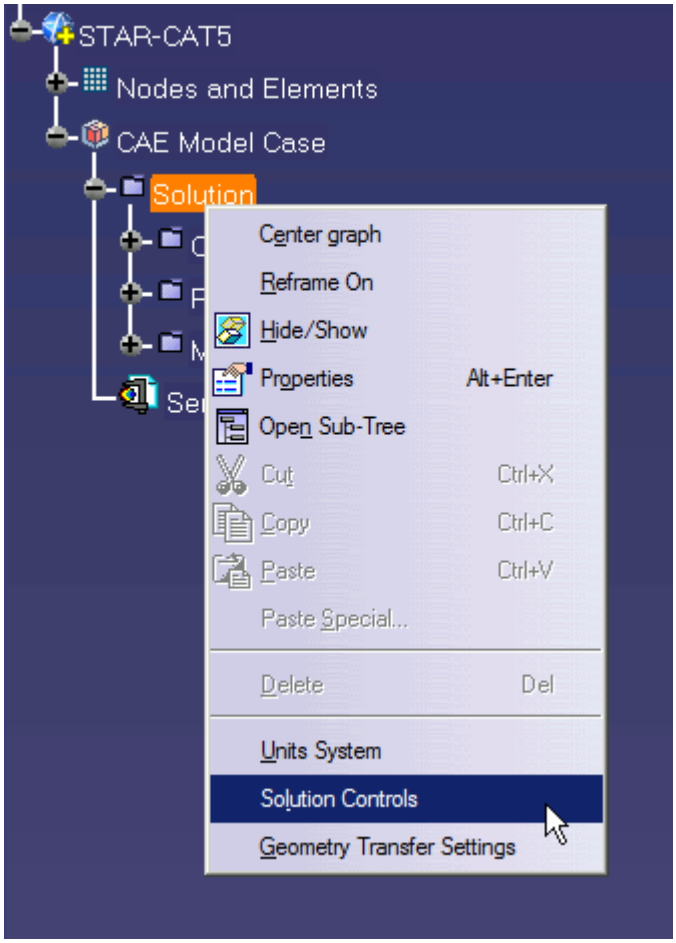


14. Click **OK**.

Modifying Solution Controls

Modify the maximum number of iterations and the solution accuracy:

1. Right-click the **Solution** node in the STAR-CAT5 object tree and select **Solution Controls**



The *Solution Controls* dialog appears.

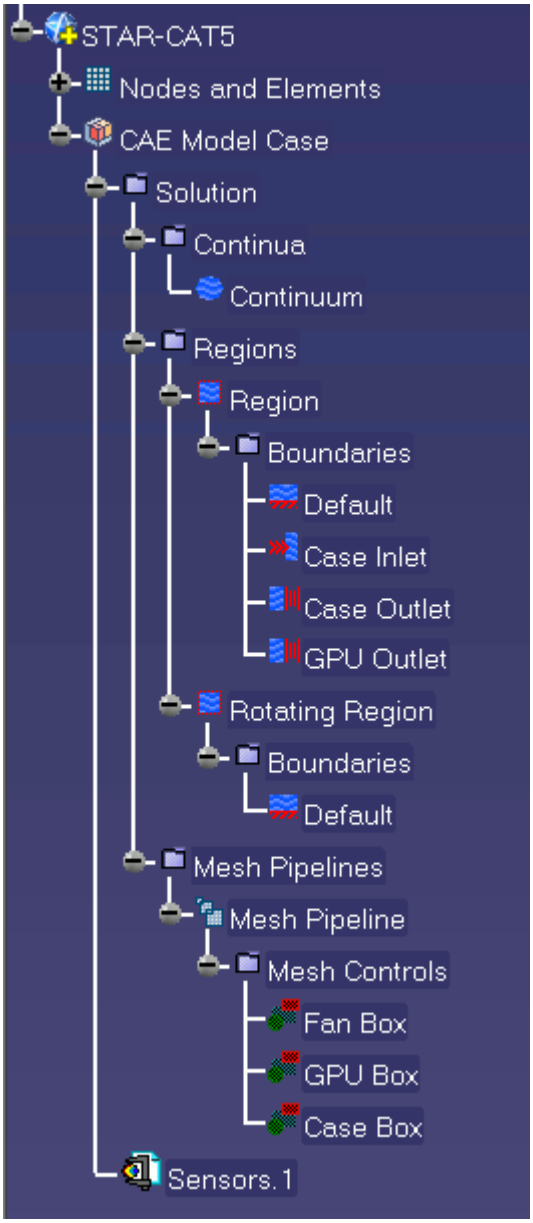
2. Click **More** to expand the *Solution Controls* menu.
3. Set the following parameters:

Parameter	Value
Iterations	300
Convergence	0.01
Momentum Under-Relaxation Factor	0.85

4. Click **OK**.
5. Save your progress.

Reviewing the Simulation Set Up

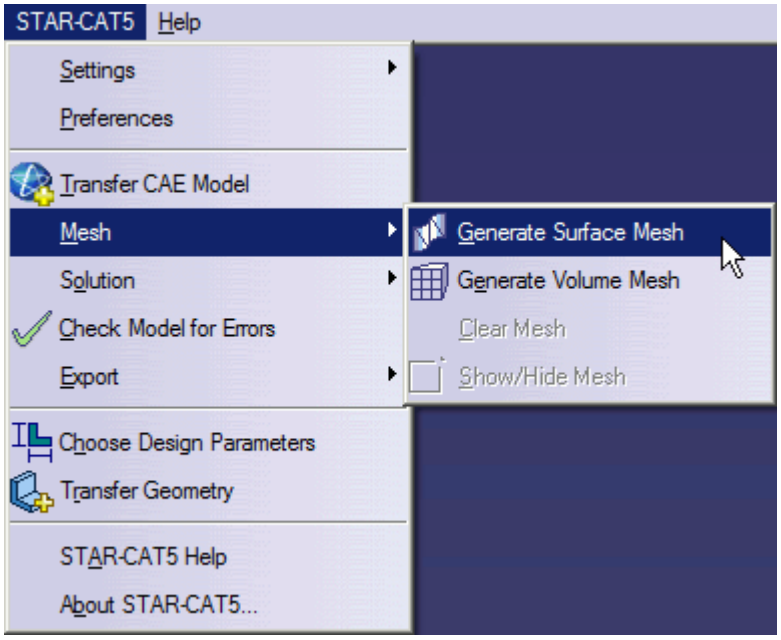
The final STAR-CAT5 object tree is shown below.



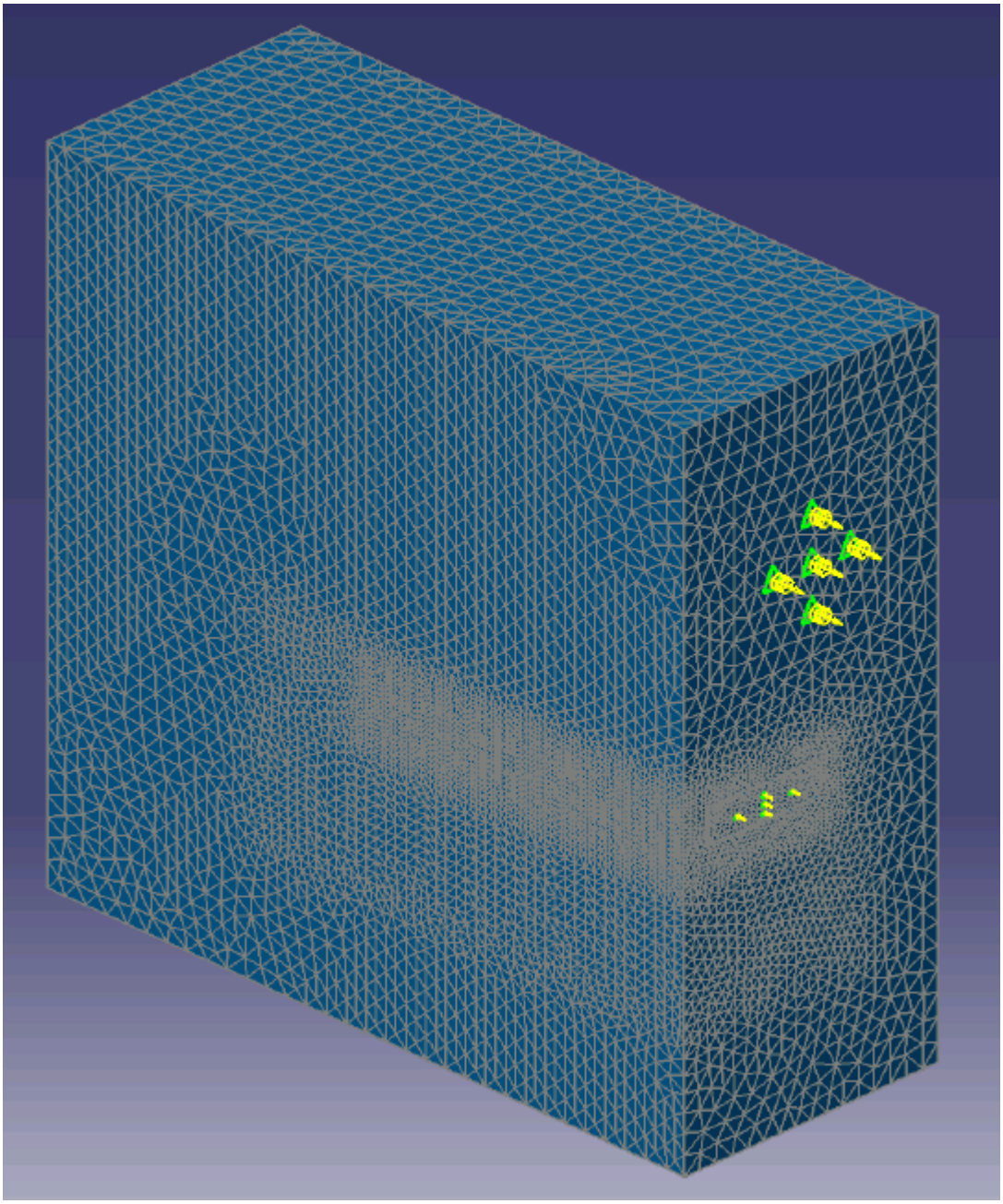
Generating The Surface Mesh

Generate the surface mesh:

1. Select **STAR-CAT5** > **Mesh** > **Generate Surface Mesh**.



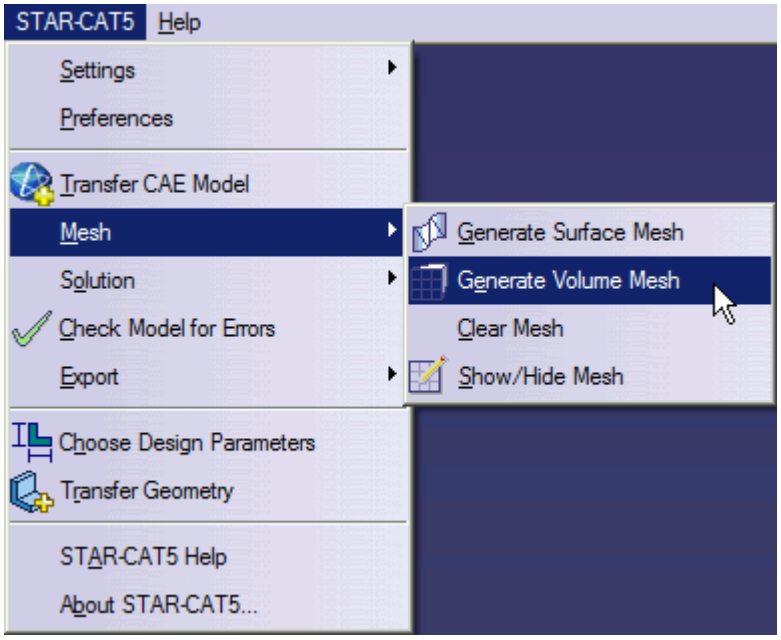
The surface mesh is shown below.



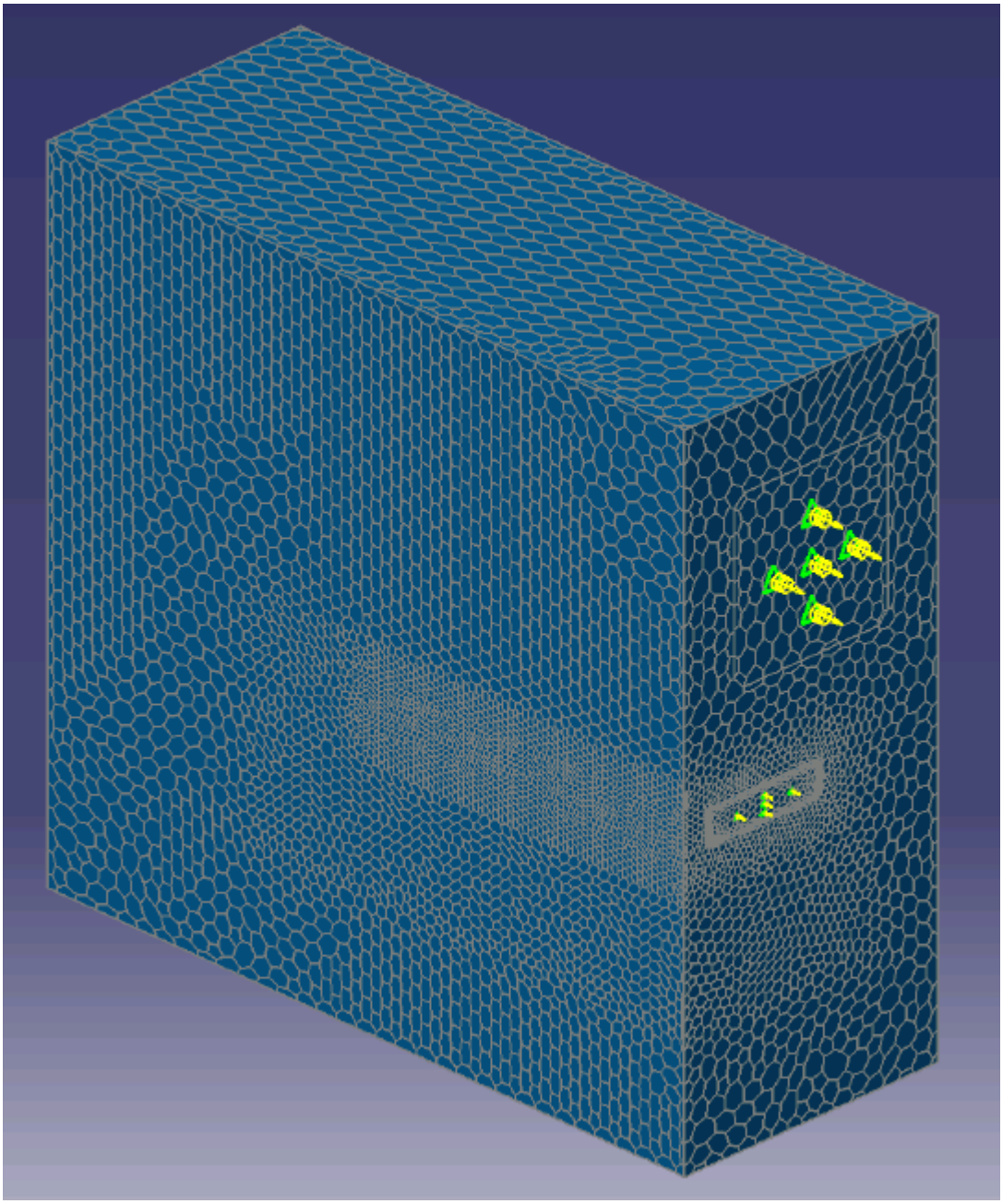
Generating the Volume Mesh

Generate the volume mesh:

1. Select **STAR-CAT5** > **Mesh** > **Generate Volume Mesh**.



The volume mesh is shown below.

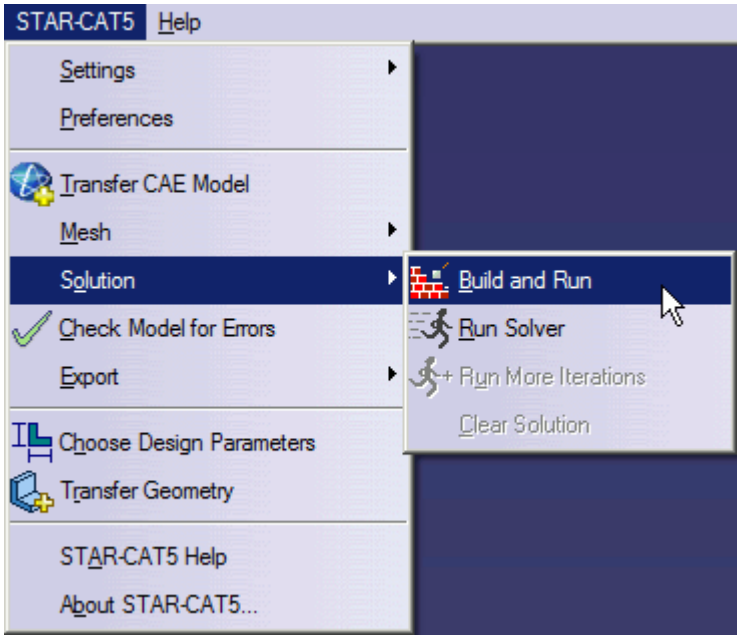


Building and Running

The case is now ready to run.

To run the simulation:

1. Select **STAR-CAT5** > **Solution** > **Build and Run**.



The *Residuals* plot appears in the STAR-CAT5 *Output* window. You can use this plot to monitor the convergence of the solution.

When the solver is done, STAR-CCM+ launches.

Post-Processing Results

You perform all post-processing in STAR-CCM+. By default, a scalar scene, *Scalar Scene 1*, automatically appears in the *Graphics* window when the solver finishes. This scene displays the absolute total pressure.

The results that are contained in the following sections are representative. You may see some variation in the results that you obtain, depending on what platform and package you are using.

1. In STAR-CCM+, click **File** > **Save** to save the simulation.

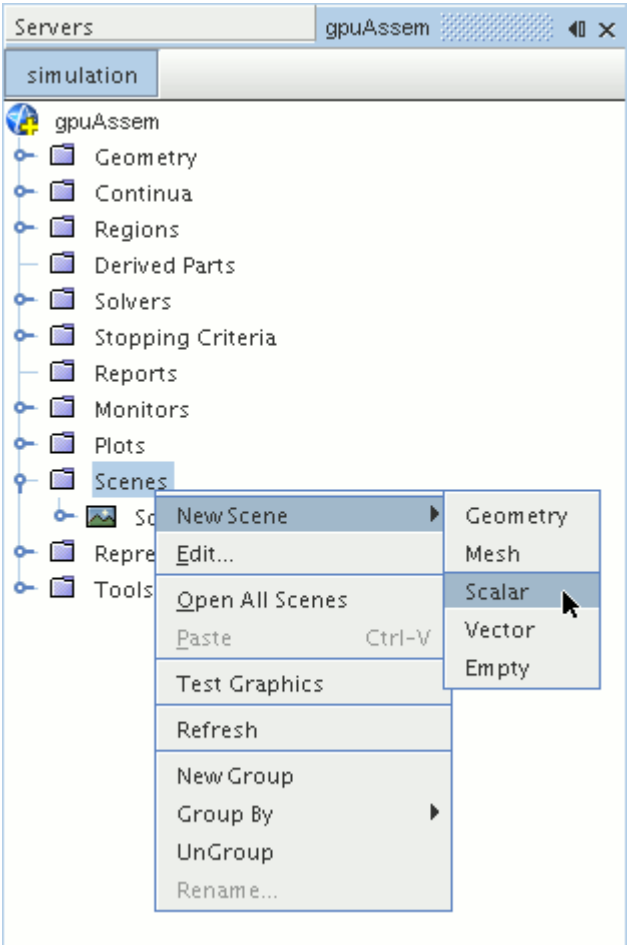
Contents:

Creating a Scalar Scene
Creating A Streamlines Scene
Creating a Vector Scene
Creating a Streamlines Scene
Creating A Scalar Scene
Visualizing the Wing Skin Temperature
Creating a Streamlines Scene
Creating A Scalar Scene
Creating a Streamlines Scene
Creating a Scalar Scene
Modifying The Scalar Scene
Creating a Streamlines Scene

Creating A Scalar Scene

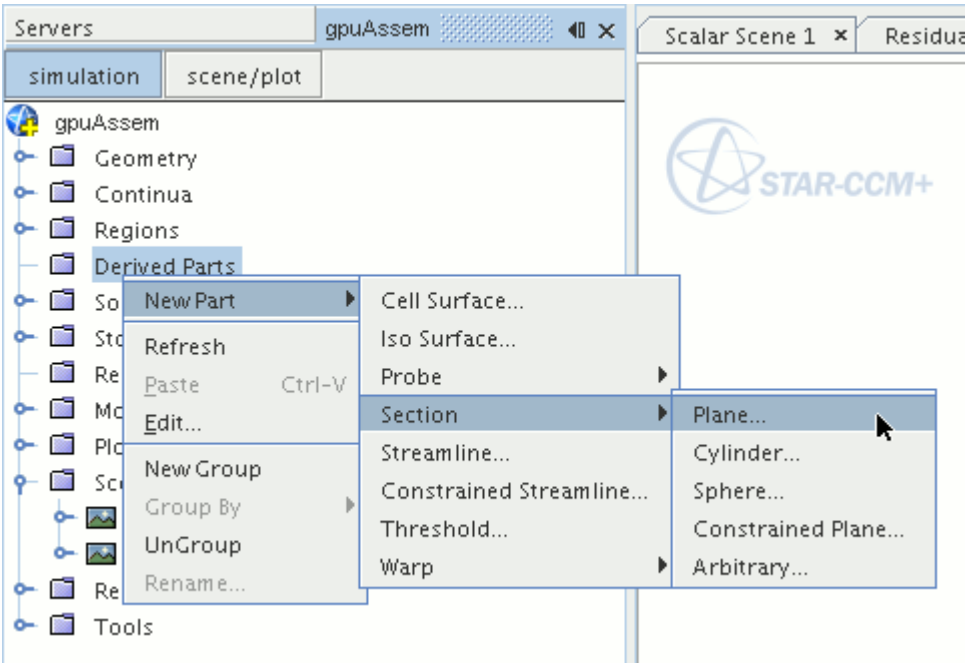
Create a scalar scene containing a cross-section of the GPU and plot velocity magnitude on this plane.

1. Right-click the **Scenes** node in the explorer pane and select **New Scene** > **Scalar**.



A new scalar scene, *Scalar Scene 2*, is created and selected in the *Graphics* window.

2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane...**.

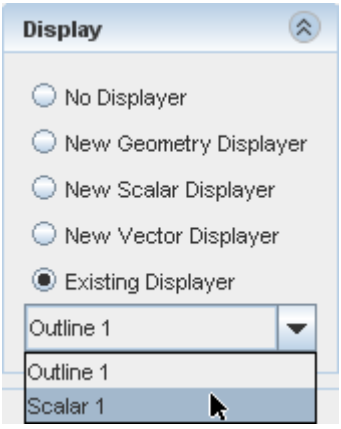


The *Create Plane Section* dialog appears.

3. In the *Plane Parameters* box, set the *origin* vector to `[0 m, -0.012 m, 0 m]` and the *normal* vector to `[0 m, 1 m, 0 m]`.

   Note that the units are in meters.

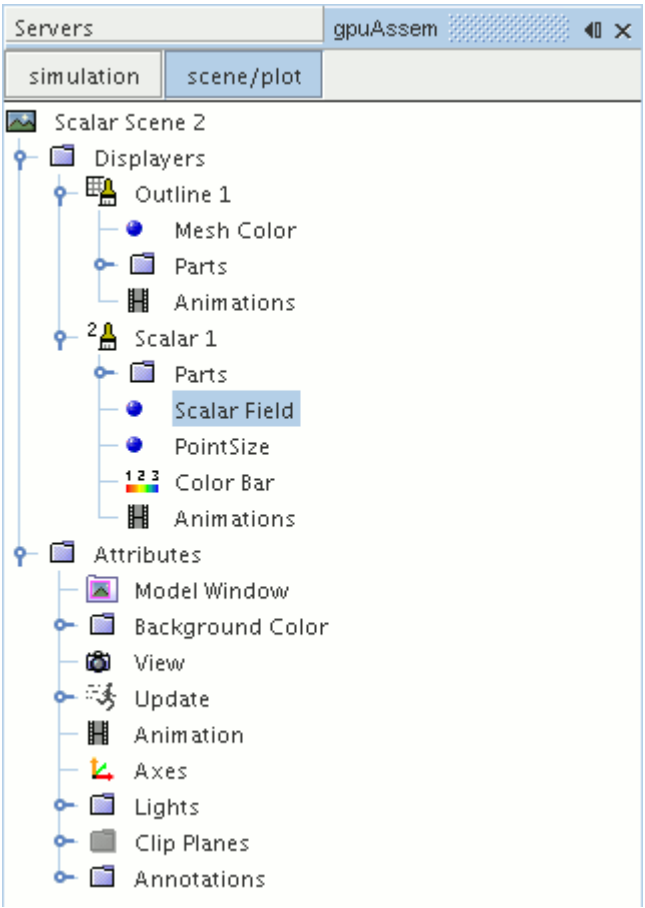
4. In the *Display* group box, select **Existing Displayer** and in the drop-down menu select **Scalar 1**.



5. Click **Create** and then **Close.**

Display velocity magnitude on the section plane:

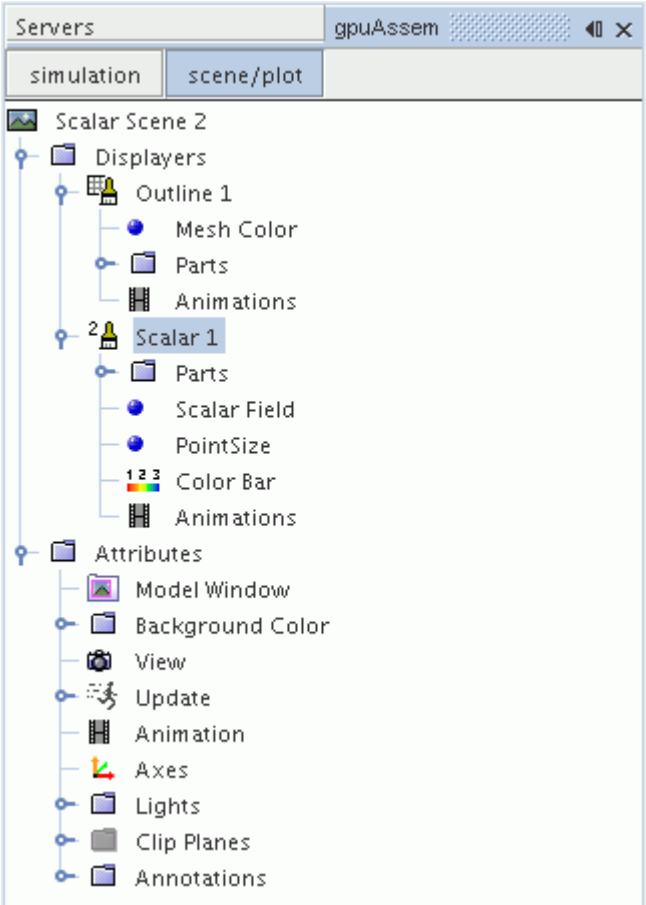
6. Click **scene/plot**.
7. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Scalar Field**.



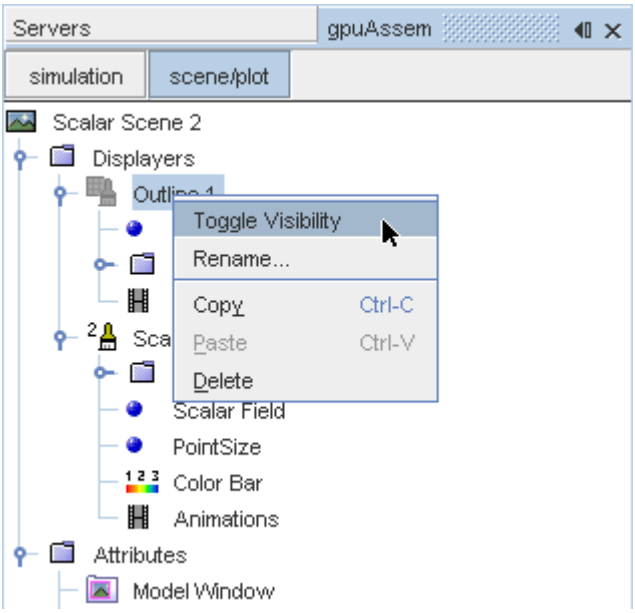
8. Set *Function* to **Velocity** > **Lab Reference Frame** > **Magnitude**.

To have a smooth transition between the different velocity values, set the contour style to smooth:

9. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1**.

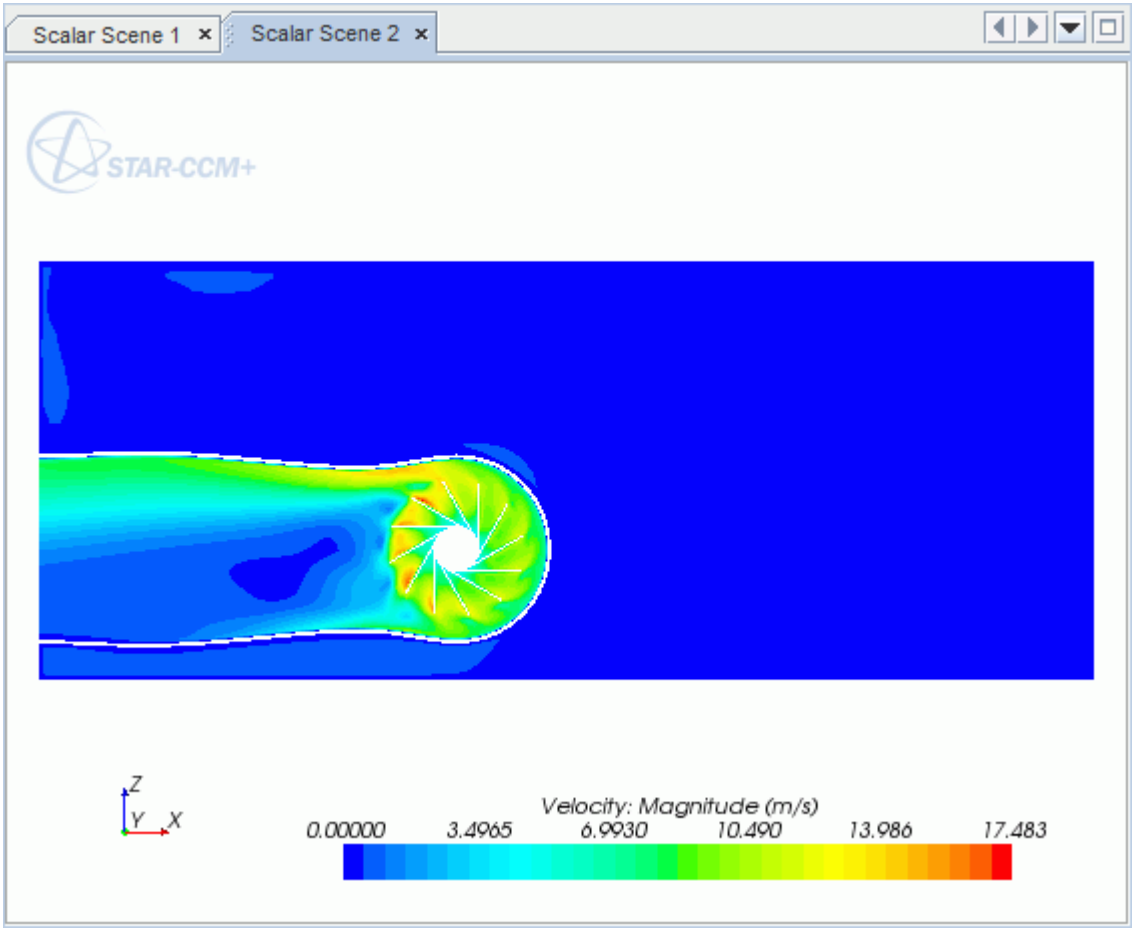


10. Set *Contour Style* to **Smooth Filled**.
11. To hide all outlines, right-click the **Scalar Scene 2** > **Displayers** > **Outline 1** node and select **Toggle Visibility**.



12. Click the *Scalar Scene 2* tab to activate the scene, and press the <S> key to change to the top view of this geometry.
13. Click **simulation**.
14. Click (**Save**).

The finished scene is shown below.

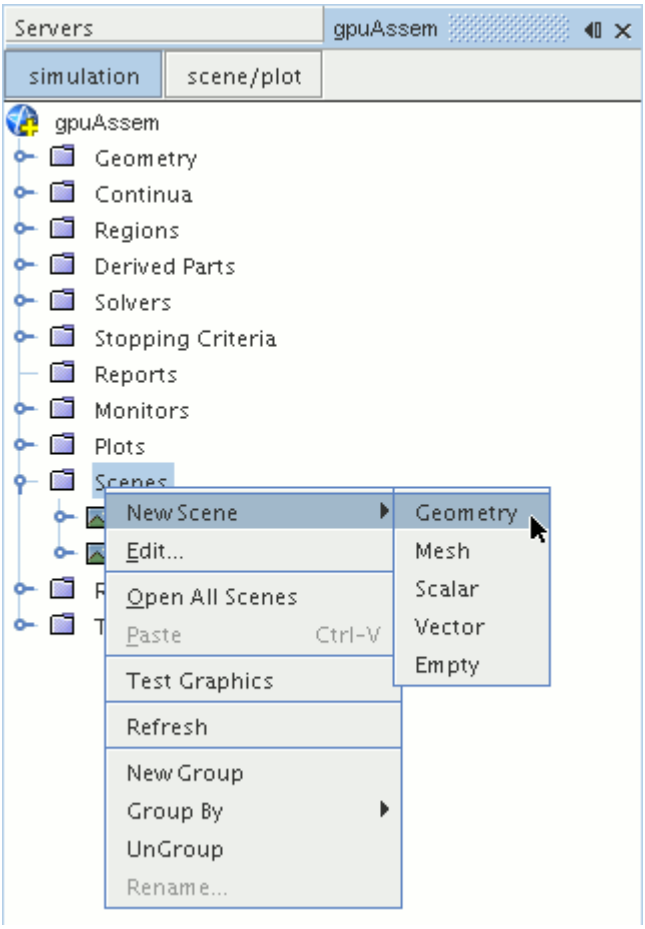


Creating a Streamlines Scene

Create streamlines to get a better understanding of the air flow inside the PC tower case and graphics card.

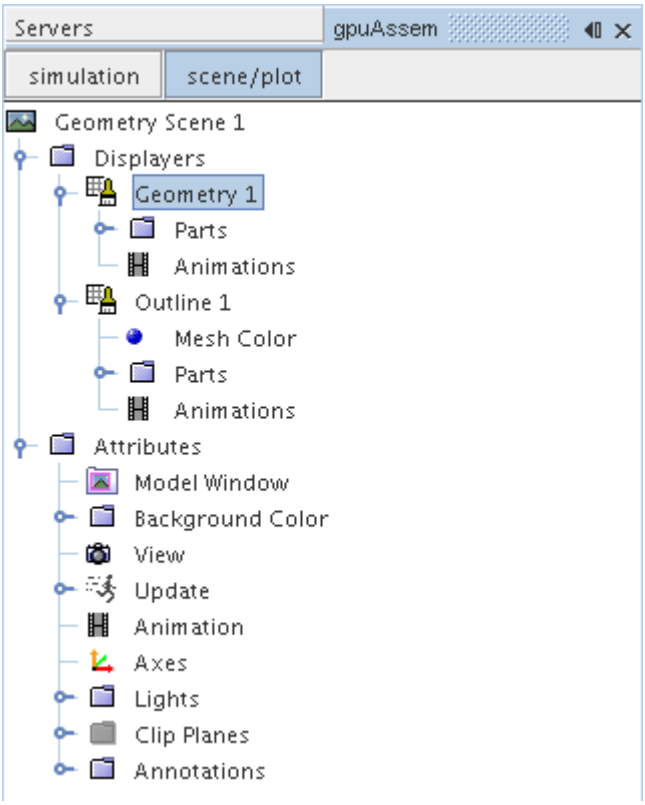
First, create a geometry scene:

1. Right-click the **Scenes** node in the object tree and select **New Scene** > **Geometry**.



A new geometry scene is created and selected in the *Graphics* window.

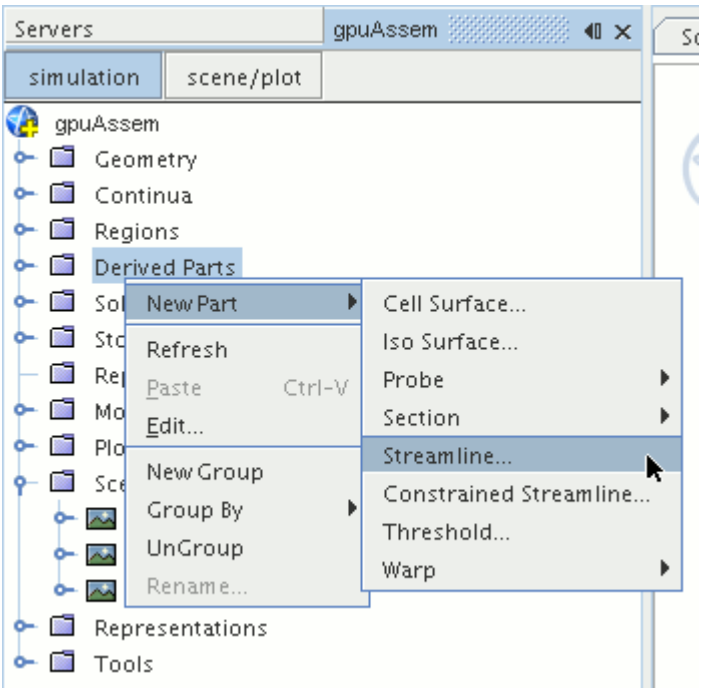
2. Click **scene/plot**.
3. Navigate to **Geometry Scene 1** > **Displayers** > **Geometry 1** node.



4. Set *Opacity* to `0.1`

Create the streamlines:

5. Click **simulation**.
6. Right-click the **Derived Parts** node and select **New Part** > **Streamline...**.



The *Create Streamline* dialog appears.

7. Make sure that **Part Seed** is selected in the *Seed Mode* section.
8. Set *Seed Parts* to **Regions** > **Region** > **Boundaries** > **Case Inlet**.
9. Set *Part U-Resolution* to `10`.
10. Set *Part V-Resolution* to `10`.
11. Make sure that **New Streamline Displayer** is selected in the *Display* section.
12. Click **Create** followed by **Close**.

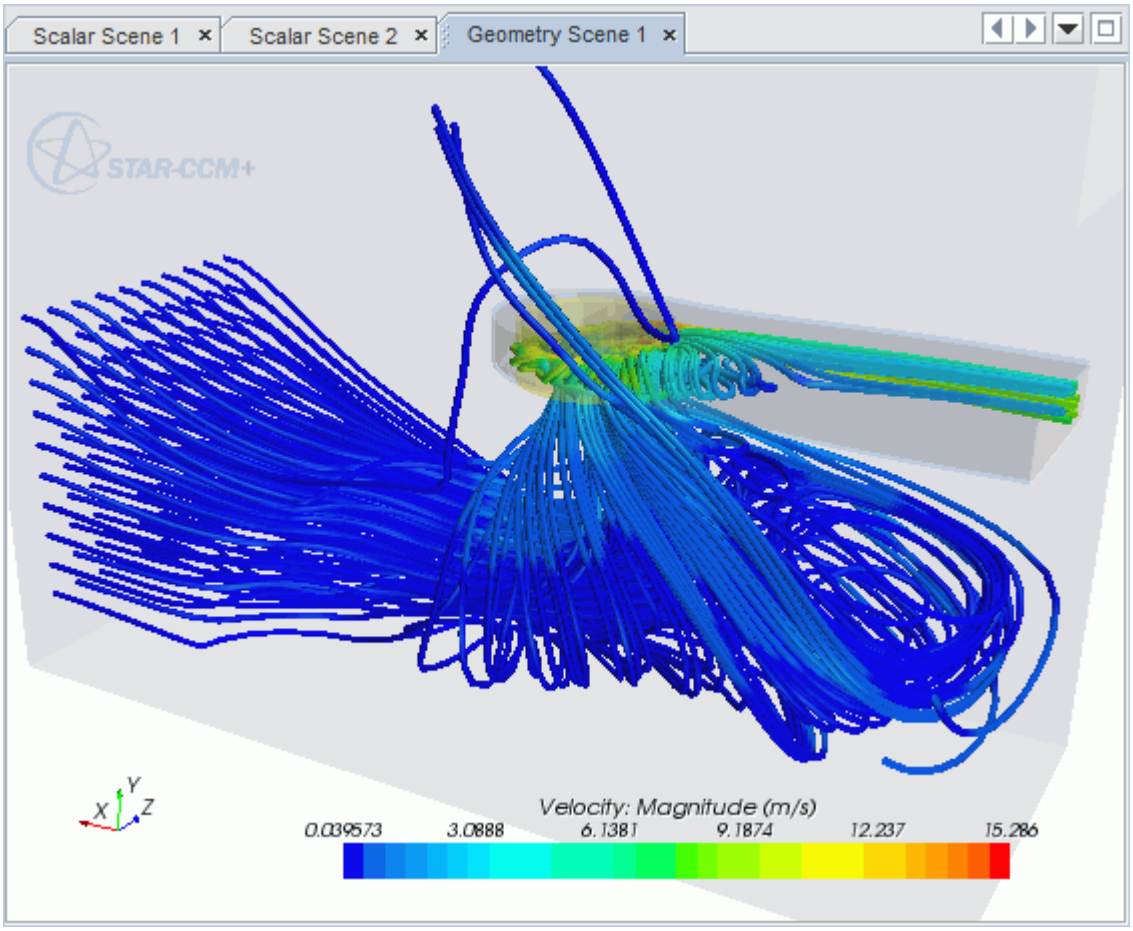
Adjust the appearance of streamlines to display tubes instead of the default lines:

13. Click **scene/plot**.
14. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** and set *Mode* to **Tubes** and *Width* to `0.0020`.
15. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** > **Scalar Field** and set *Function* to **Velocity** > **Lab Reference Frame** > **Magnitude**.

Hide all the outlines in the scene:

16. Right-click the **Geometry Scene 1** > **Displayers** > **Outline 1** node and select **Toggle Visibility**.
17. Reposition the geometry to get a better view of the flow inside the case.
18. Save the simulation.

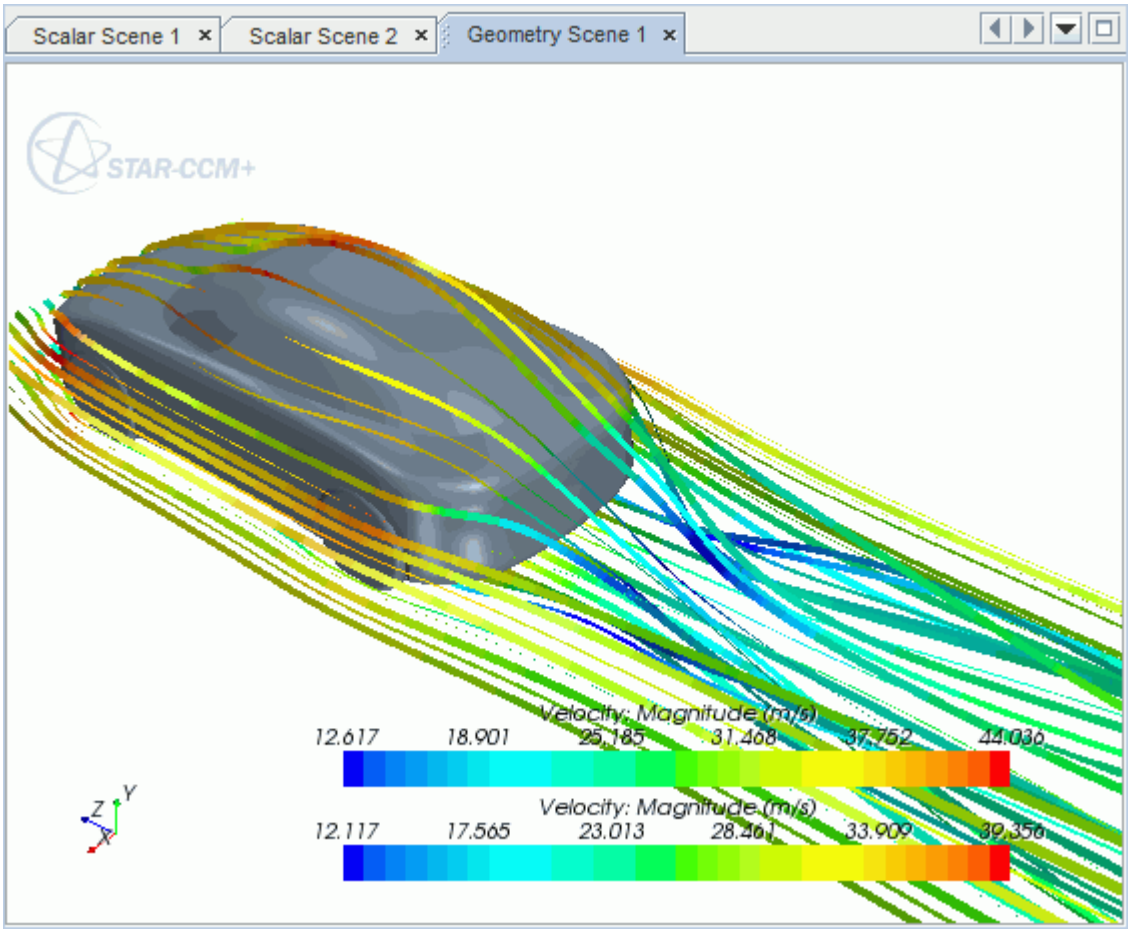
The finished scene is shown below.


Scalar Scene 1
Scalar Scene 2
Geometry Scene 1
STAR-CCM+
Velocity: Magnitude (m/s)
0.039573
3.0888
6.1381
9.1874
12.237
15.286
X
Y
Z

External Flow Over a Car

This tutorial takes you through the steps that are required to run a CFD calculation in the STAR-CAT5 environment.

For this tutorial, the model is a car in a fluid domain.



Contents:

Opening the CAD Model
Selecting Fluid Physics Models
Defining Boundary Conditions
Setting the Mesh Pipeline
Creating the Volumetric Mesh Controls
Creating the Face Mesh Control
Modifying Solution Controls
Reviewing the Simulation Set Up
Generating the Surface Mesh

Generating the Volume Mesh
Building and Running
Post-Processing Results

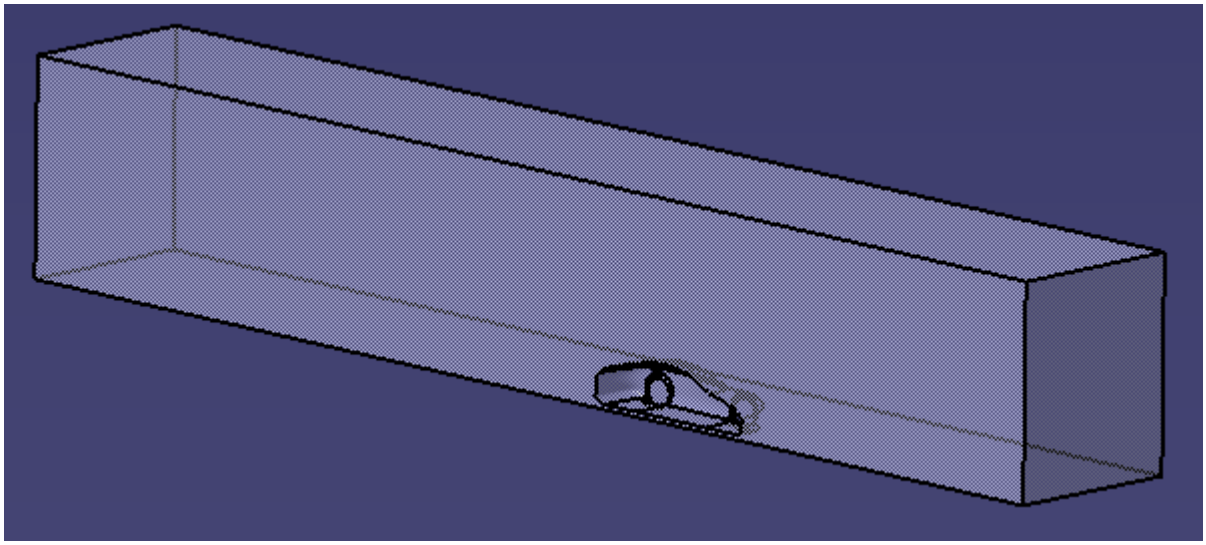
Opening the CAD Model

A directory containing the CAD geometry is provided with the STAR-CCM+ distribution.

This directory is located in `[STAR-CCM+_INSTALL_DIR]\doc\startutorialsdata\starcat5\data`.

1. Copy directory `data` to your own disk space and open the part file `car.CATPart` in STAR-CAT5.

   See Launching STAR-CAT5.
2. Position the geometry as shown below.



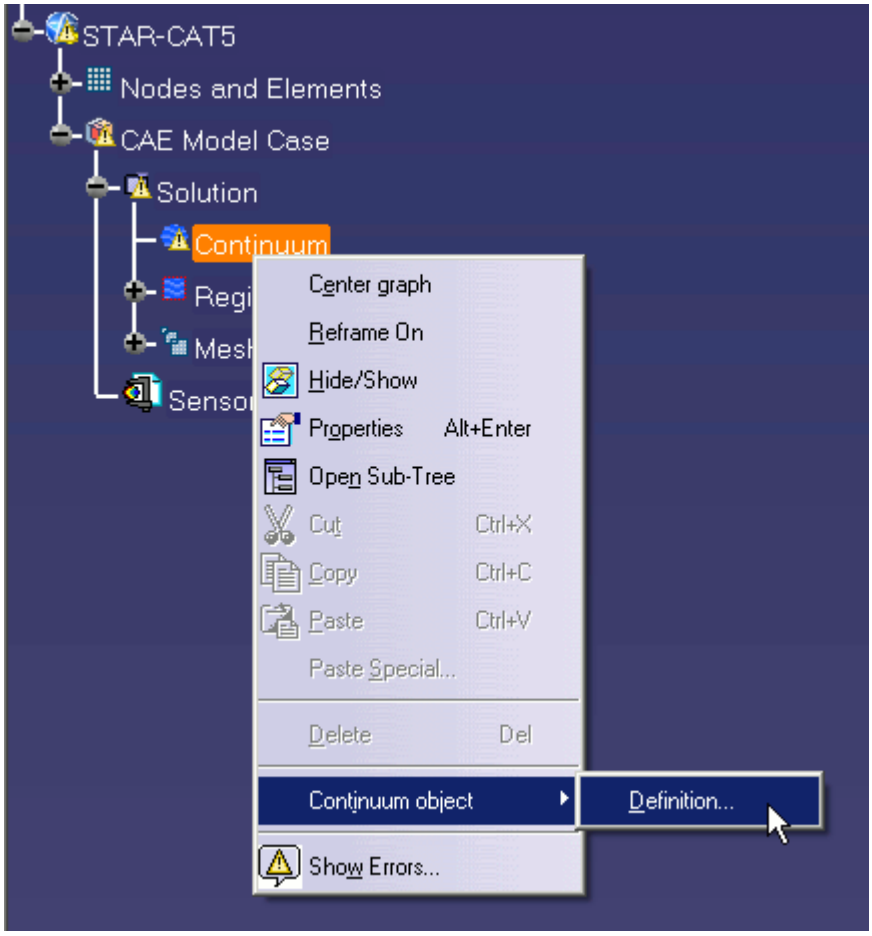
The STAR-CAT5 menu appears at the top of the screen.The STAR-CAT5 object tree is located on the left of the screen.

3. Save the analysis as `car.CATAnalysis`.

Selecting Fluid Physics Models

The first step in setting up this tutorial is to define the physics of the problem. For this case, the flow is turbulent and isothermal.

1. Right-click the **Continuum** node in the STAR-CAT5 tree and select **Continuum object** > **Definition...**.



The *Continuum Settings* dialog appears.

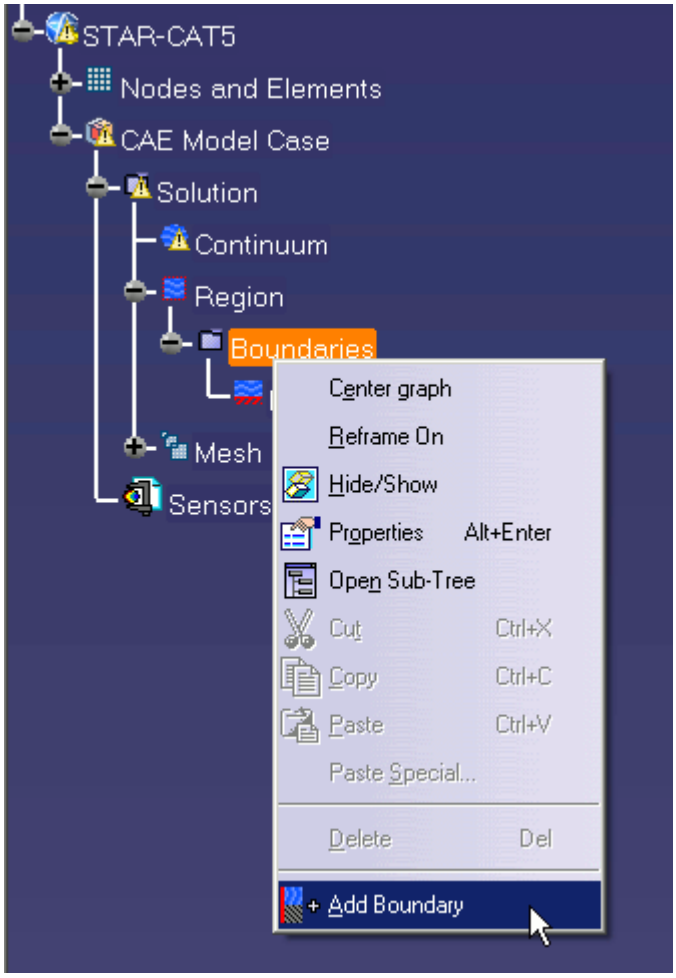
2. Deactivate the **Temperature** option in the *Optional Physics Models*.
3. Click **OK**.

Defining Boundary Conditions

Create the inlet and outlet boundary condition. Use a moving wall to model the relative movement of the road.

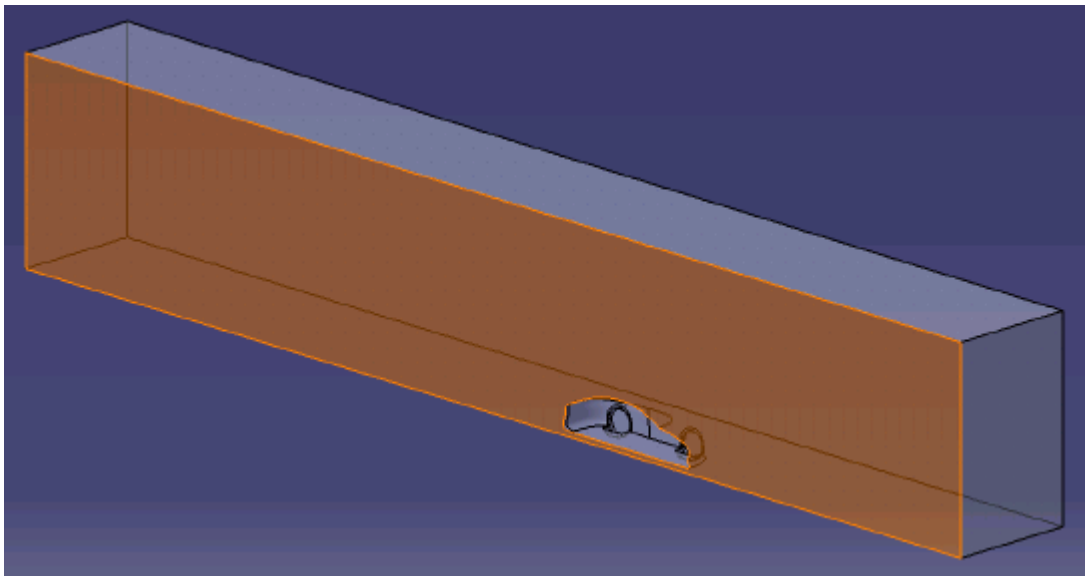
Create the symmetry plane first:

1. Right-click the **Region** > **Boundaries** node and select **Add Boundary**.



The *Boundary Settings* dialog appears.

2. On the CAD model, select the face that is highlighted below.



These selections are reported in the *Supports box* in the *Boundary Settings* dialog.

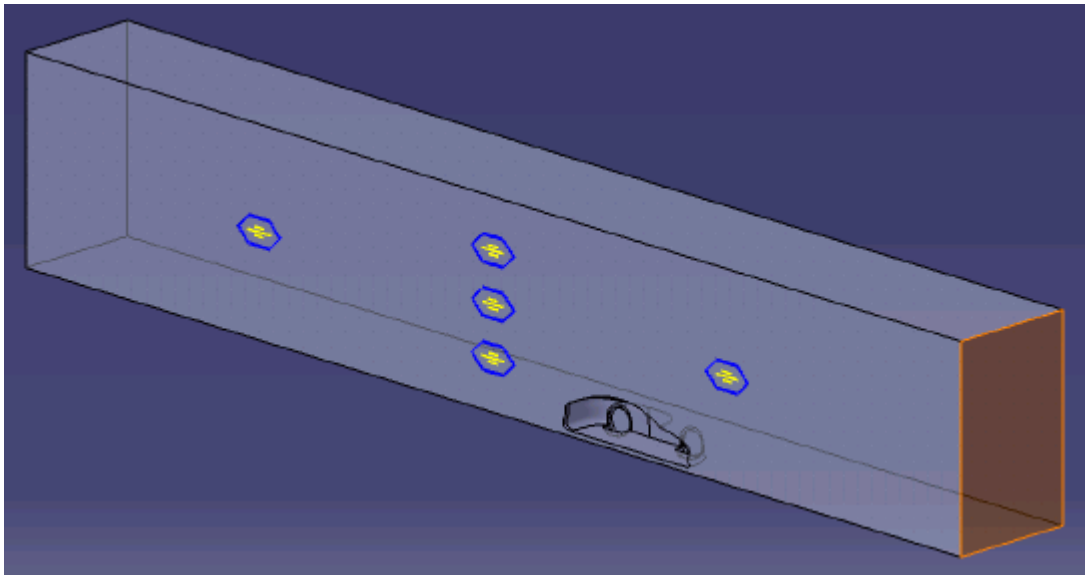
3. Set the following parameters:

Parameter	Value
Boundary Type	**Symmetry Plane**
Name	`Symmetry Plane`

4. Click **OK**.

Create the inlet boundary:

5. Create another boundary and select the following face.



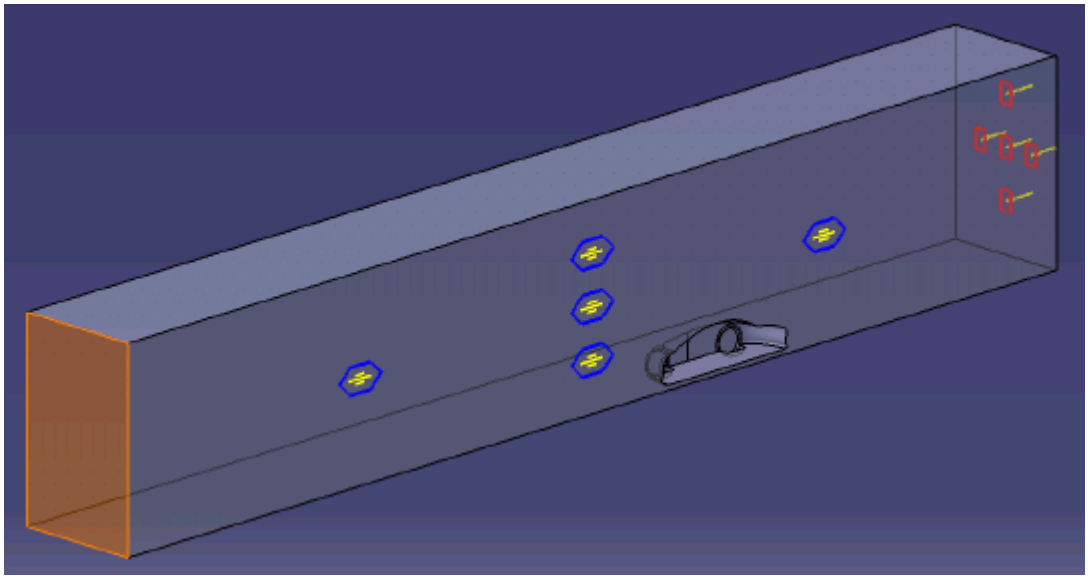
6. Set the following parameters:

Parameter	Value
Boundary Type	**Velocity Inlet**
Name	`Inlet`
Normal Velocity	`70 mph` **Note:** The units are in **mph**.

7. Click **OK**.

Create the outlet boundary:

8. Create another boundary.
9. Rotate the view to display the back face and select the face that is highlighted below.



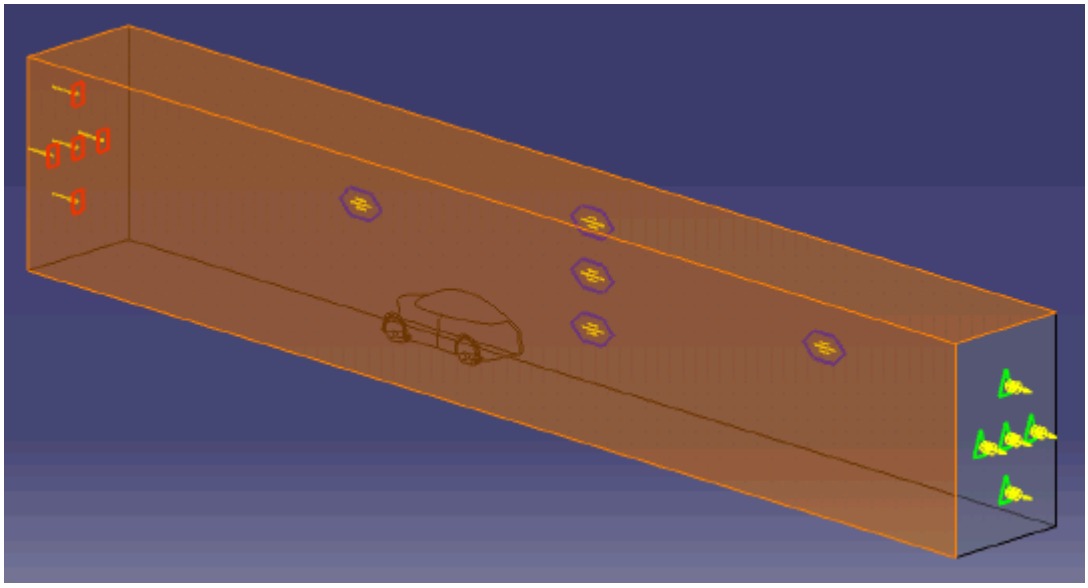
10. Set the following parameters:

Parameter	Value
Boundary Type	**Pressure Outlet**
Name	`Outlet`

11. Click **OK**.

Create the fluid domain symmetry boundaries:

12. Create another boundary.
13. Rotate the view to display the top and side walls, and select the faces that are highlighted below.



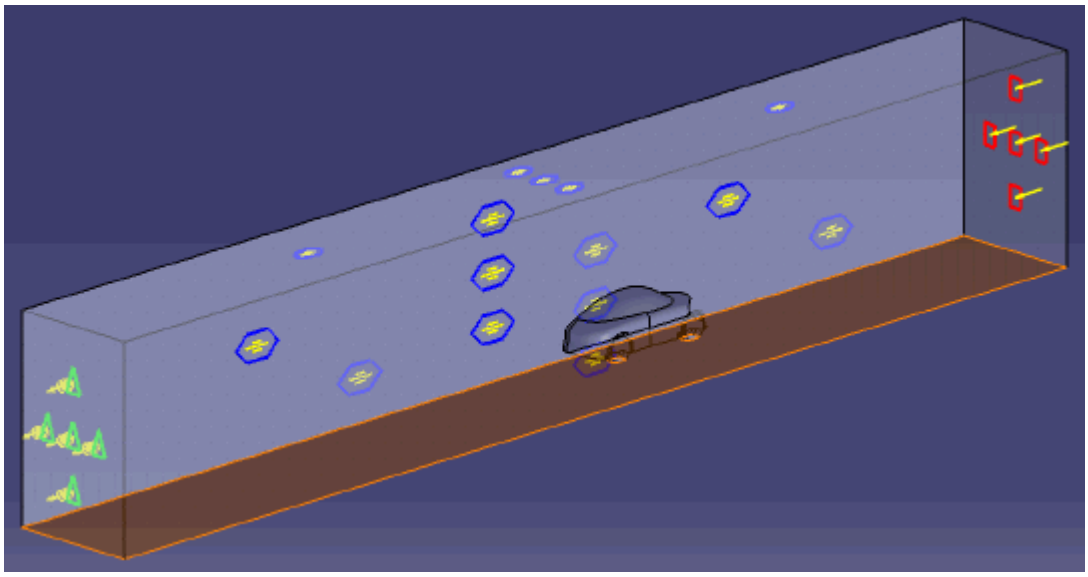
14. Set the following parameters:

Parameter	Value
Boundary Type	**Symmetry Plane**
Name	`Boundary Symmetry`

15. Click **OK**.

Create the boundary that will model the moving ground:

16. Create another boundary.
17. Rotate the view to display the bottom face, and select the face that is highlighted below.



18. Set the following parameters:

Parameter	Value
Boundary Type	**Wall**
Name	`Ground`
Wall Type	**Translating**
W Component of Wall Velocity Vector	`-70 mph` **Note:** The units are in **mph**.

19. Click **OK**.

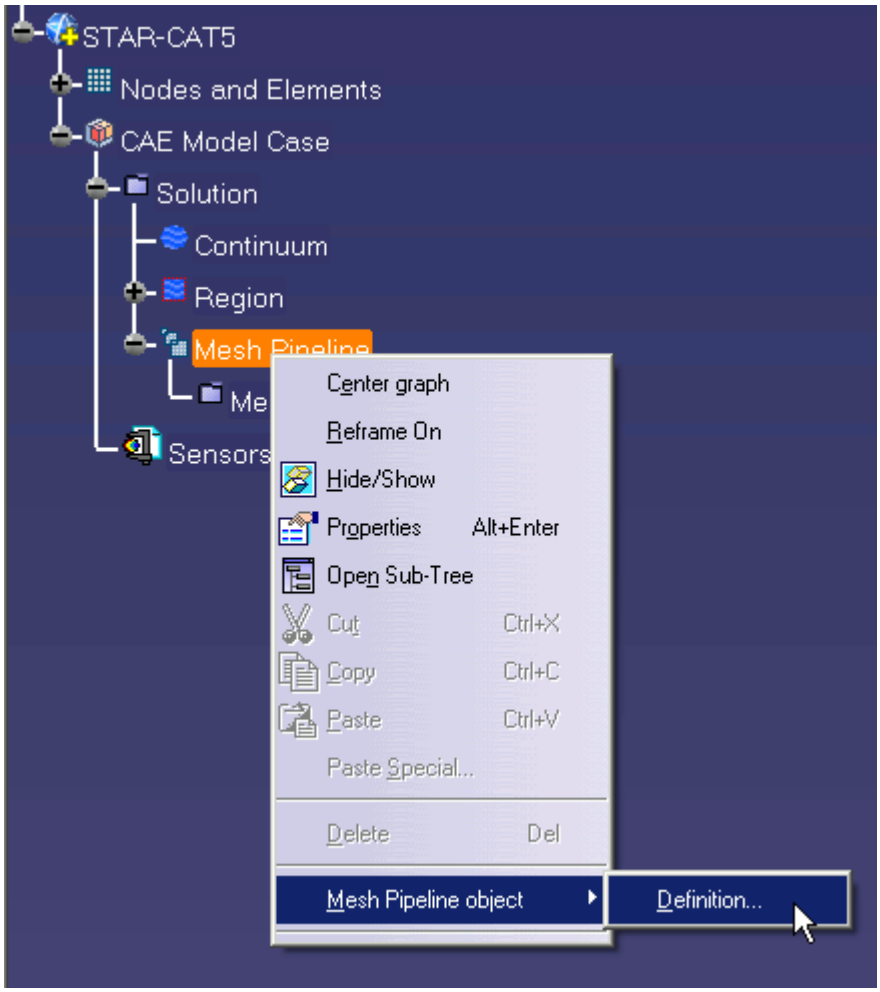
Rename the default boundary:

20. Right-click **Region** > **Boundaries** > **Default** and select **Default object** > **Definiton**.
21. In the *Boundary Settings* dialog, set *Name* to `Car`.
22. Click **OK** to close the dialog.
23. Save your progress.

Setting the Mesh Pipeline

To obtain more accurate results, modify the mesh pipeline:

1. Right-click the **Mesh Pipeline** node in the STAR-CAT5 tree and select **Mesh Pipeline object** > **Definition**.



The *Mesh Pipeline Settings* dialog appears.

2. Set the following parameters:

Parameter	Value
Base Size	
Use Absolute Value	**Activated**
Base Size	`0.03 m`
Prism Layer	
User Absolute Value	**Activated**
Prism Layer Thickness	`0.03 m`

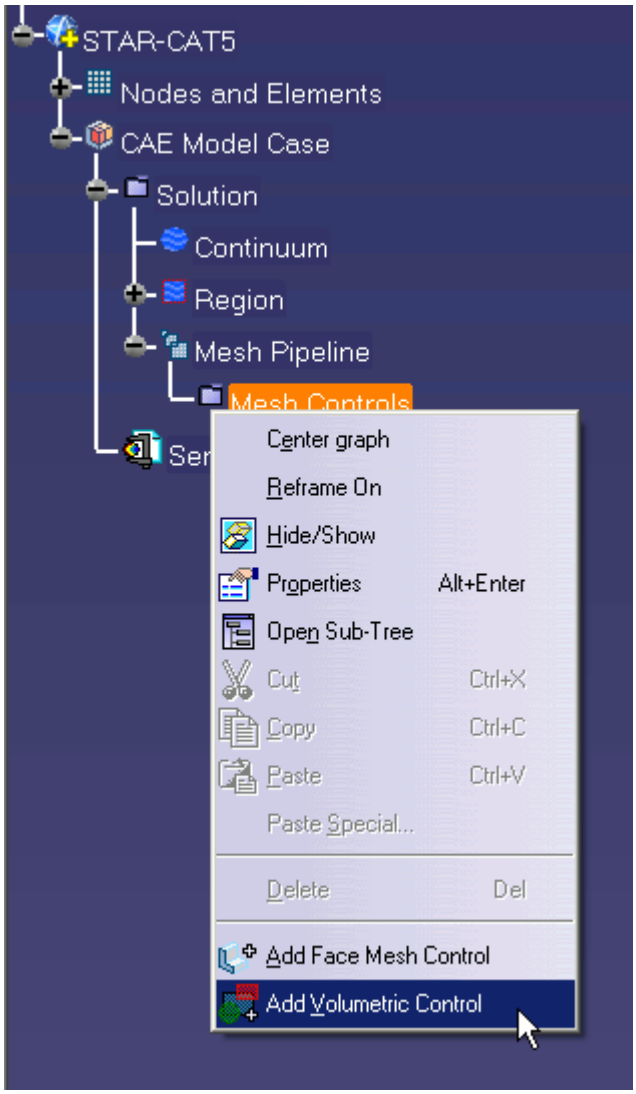
3. Click **OK**.

Creating the Volumetric Mesh Controls

Create two volumetric mesh control regions to capture the finer details of the flow: one around the car and one that extends into the wake of the car.

Create the volumetric mesh control for the car region:

1. Right-click the **Mesh Pipeline** > **Mesh Controls** node in the STAR-CAT5 tree and select **Add Volumetric Control**.



   The *Volumetric Control* dialog appears.

2. Set *Name* to `Car Box`.
3. Activate the *Use Absolute Value* option.
4. Set *Surface Mesh Size* to `0.09 m`.
5. Enter the following coordinates for Point 1:

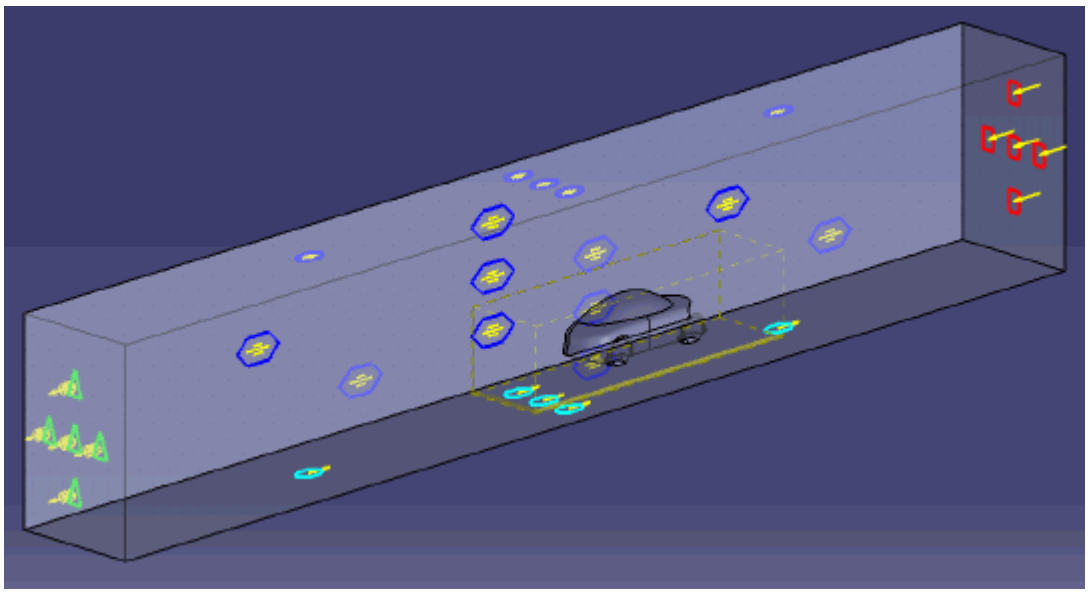
Coordinate	Value
X1	`-50.00 mm`
Y1	`-650.00 mm`
Z1	`3432.00 mm`

6. Enter the following coordinates for Point 2:

Coordinate	Value
X2	`2000.00 mm`
Y2	`1440.00 mm`
Z2	`-4567.00 mm`

7. Click **Create**.

   The volumetric control region is shown below.



8. Click **OK**.

Create the volumetric mesh control for the wake region:

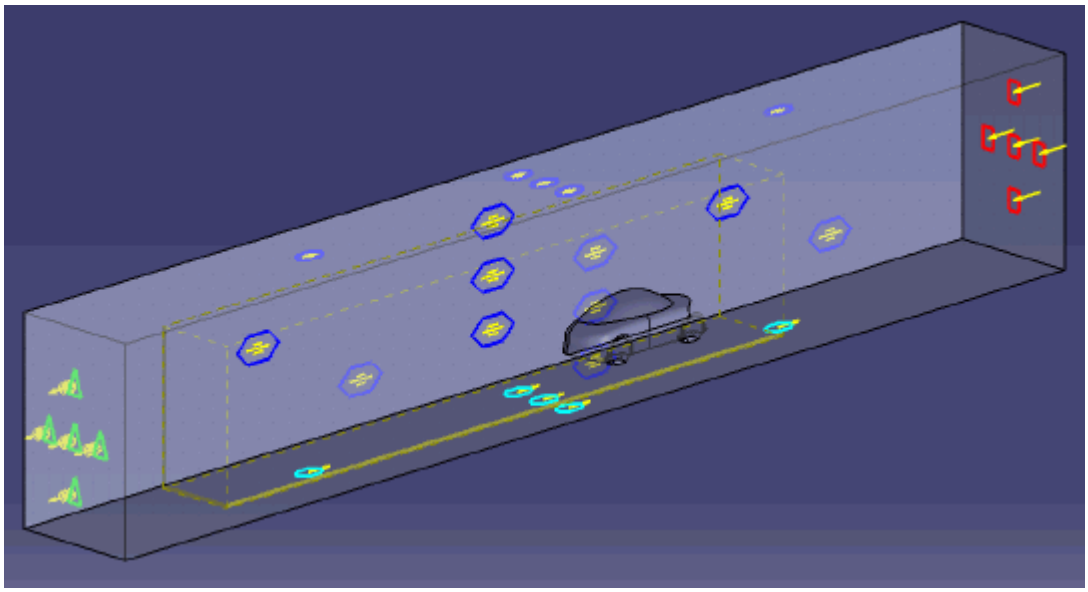
9. Using the same technique, create another volumetric mesh control with the following settings:

Parameter	Value
Name	`Wake Box`
Surface Mesh Size	
Use Absolute Value	**Activated**
Surface Mesh Size	`0.177 m`
Shape	

Parameter	Value
X1	-50.00 mm
Y1	-650.00 mm
Z1	3432.00 mm
X2	2000.00 mm
Y2	3240.00 mm
Z2	-14567.00 mm

10. Click **Create**.

    The volumetric control region is shown below.



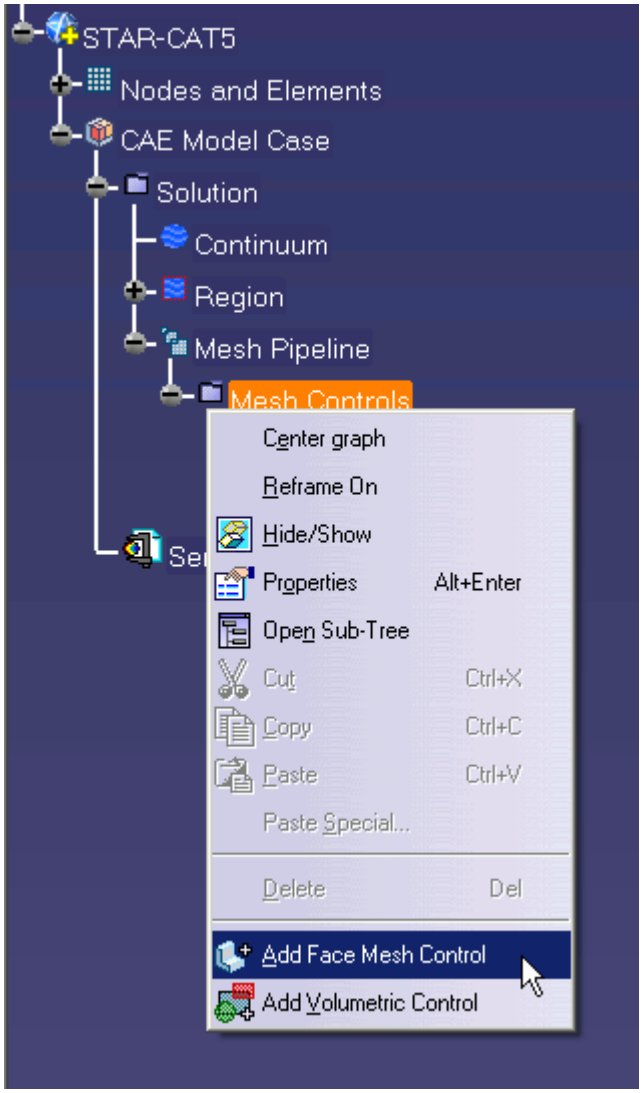
11. Click **OK**.

Creating the Face Mesh Control

Apply a face mesh control to the walls of the fluid domain to coarsen the mesh on these faces.

Create the face mesh control:

1. Right-click the **Mesh Pipeline** > **Mesh Controls** node in the STAR-CAT5 tree and select **Add Face Mesh Control**.



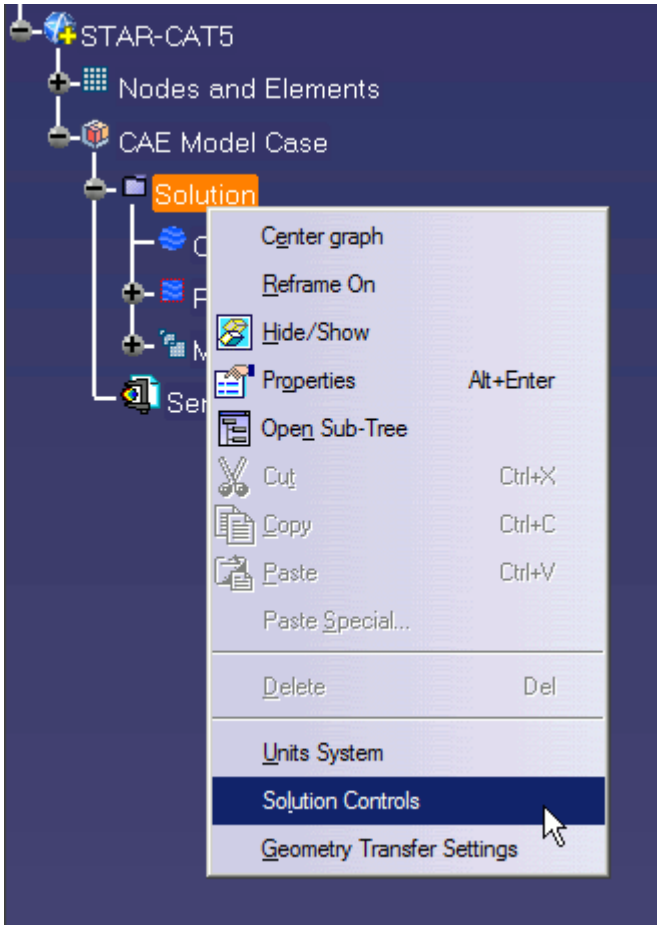
The *Face Mesh Control Settings* dialog appears.

2. Select all six boundary faces that form the fluid domain: `Inlet`, `Outlet`, `Boundary Symmetry`, `Symmetry Plane`, and `Ground`.
3. Set *Name* to `Fluid Domain Face Control`.
4. Activate the **Custom Mesh Size** option.
5. Activate the *Use Absolute Value* option.
6. Set *Mesh Size* to `0.35 m`.
7. Click **OK**.

Modifying Solution Controls

Modify the solution controls:

1. Right-click the **Solution** node in the STAR-CAT5 object tree and select **Solution Controls**



The *Solution Controls* dialog appears.

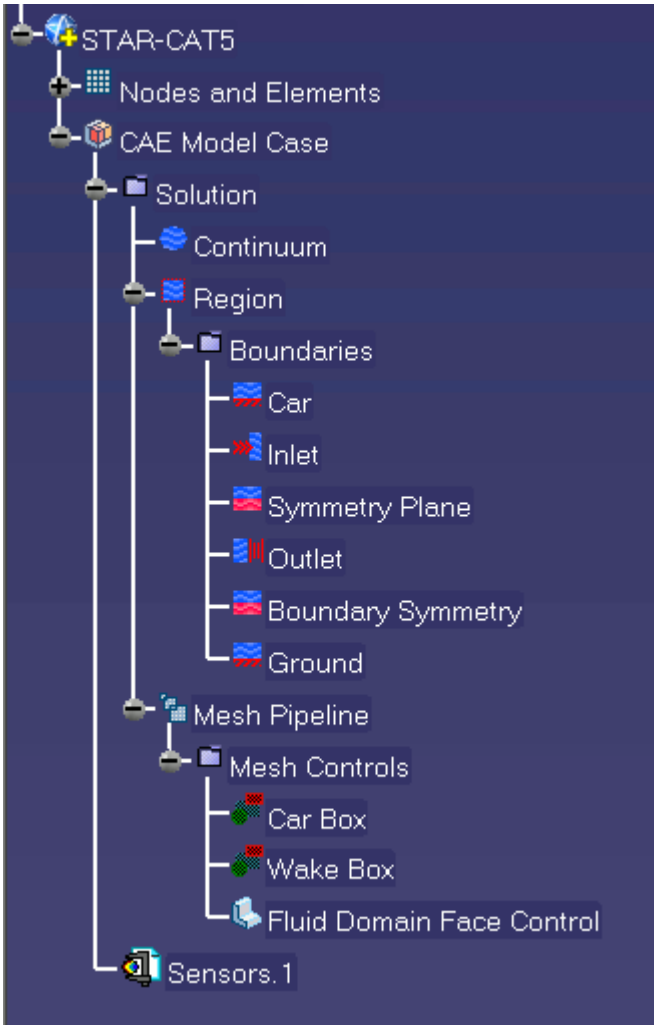
2. Click **More** to expand the *Solution Controls* menu.
3. Set the following parameters:

Parameter	Value
Iterations	`400`
Convergence	`0.0001`
Momentum Under-Relaxation Factor	`0.85`

4. Click **OK**.
5. Save your progress.

Reviewing the Simulation Set Up

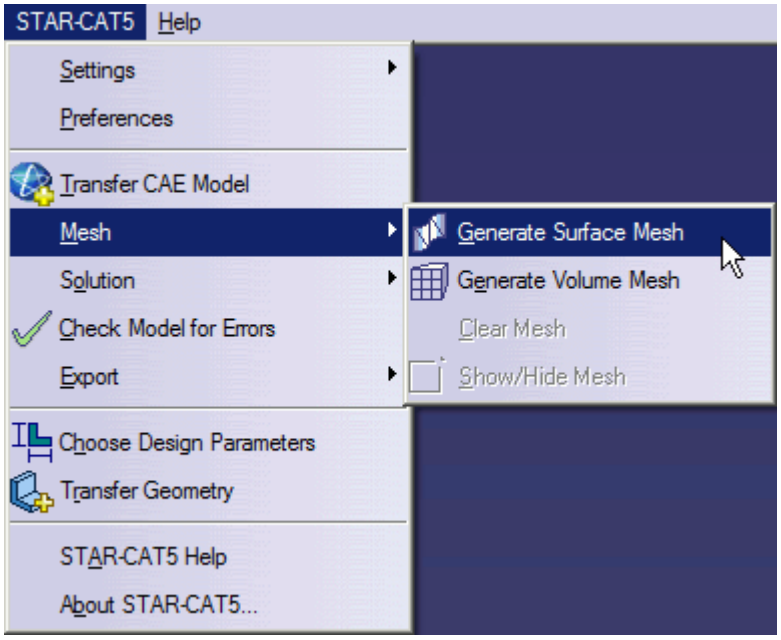
The final STAR-Works object tree is shown below.



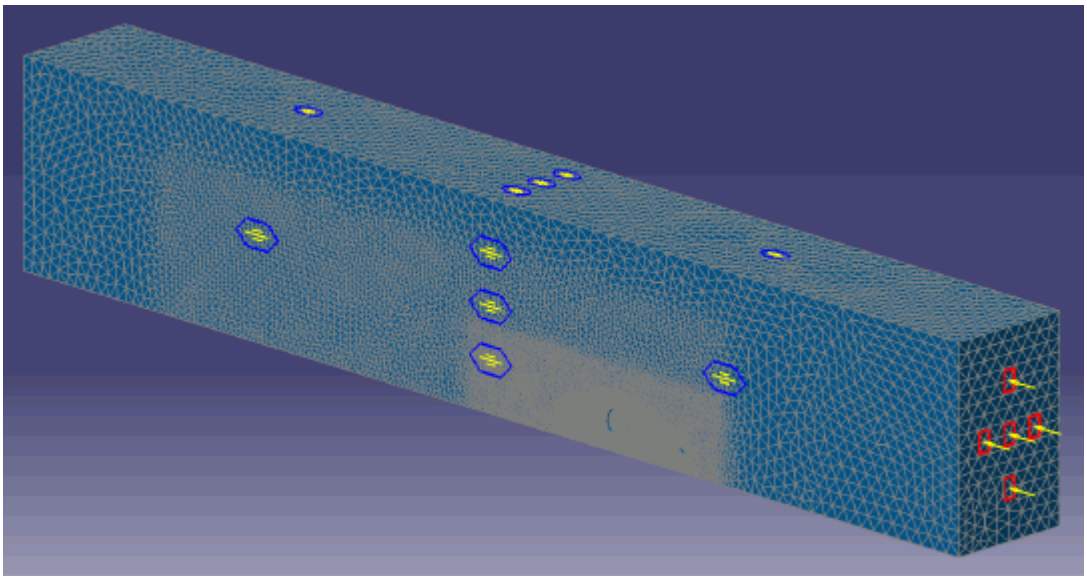
Generating the Surface Mesh

Generate the surface mesh:

1. Select **STAR-CAT5** > **Mesh** > **Generate Surface Mesh**.



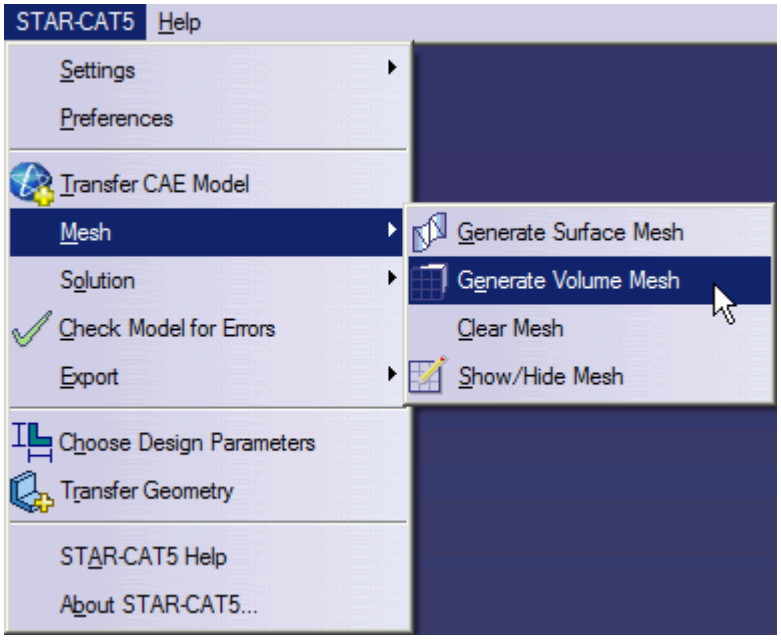
The surface mesh is shown below.



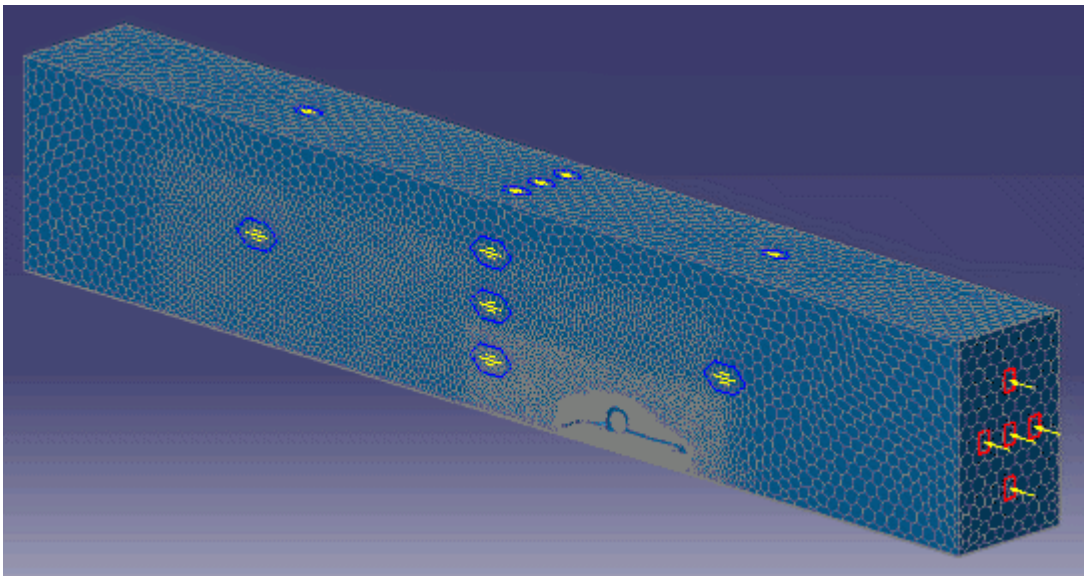
Generating the Volume Mesh

Generate the volume mesh:

1. Select **STAR-CAT5** > **Mesh** > **Generate Volume Mesh**.



The volume mesh is shown below.

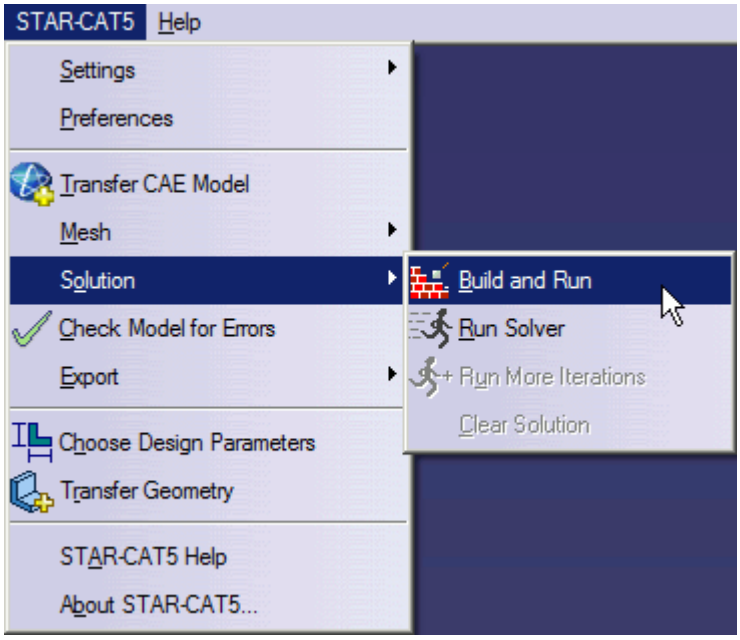


Building and Running

The case is now ready to run.

To run the simulation:

1. Select **STAR-CAT5** > **Solution** > **Build and Run**.



The *Residuals* plot appears in the STAR-CAT5 *Output* window. You can use this plot to monitor the convergence of the solution.

When the solver is done, STAR-CCM+ launches.

Post-Processing Results

You perform all post-processing in STAR-CCM+. By default, a scalar scene, *Scalar Scene 1*, automatically appears in the *Graphics* window when the solver finishes. This scene displays the absolute total pressure.

The results that are contained in the following sections are representative. You may see some variation in the results that you obtain, depending on what platform and package you are using.

1. In STAR-CCM+, click **File** > **Save** to save the simulation.

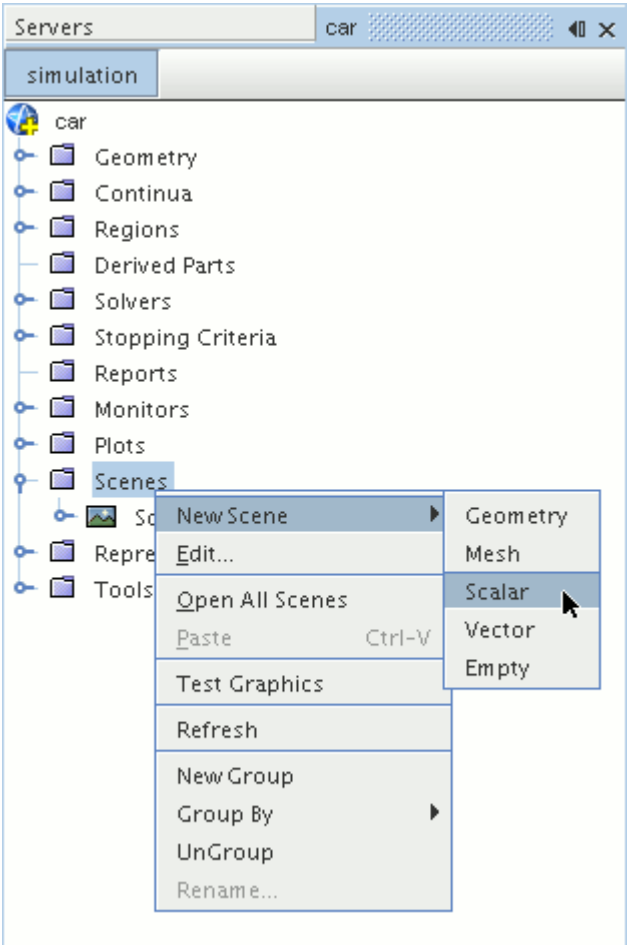
Contents:

Creating a Scalar Scene
Creating A Streamlines Scene
Creating a Vector Scene
Creating a Streamlines Scene
Creating A Scalar Scene
Visualizing the Wing Skin Temperature
Creating a Streamlines Scene
Creating A Scalar Scene
Creating a Streamlines Scene
Creating a Scalar Scene
Modifying The Scalar Scene
Creating a Streamlines Scene

Creating a Scalar Scene

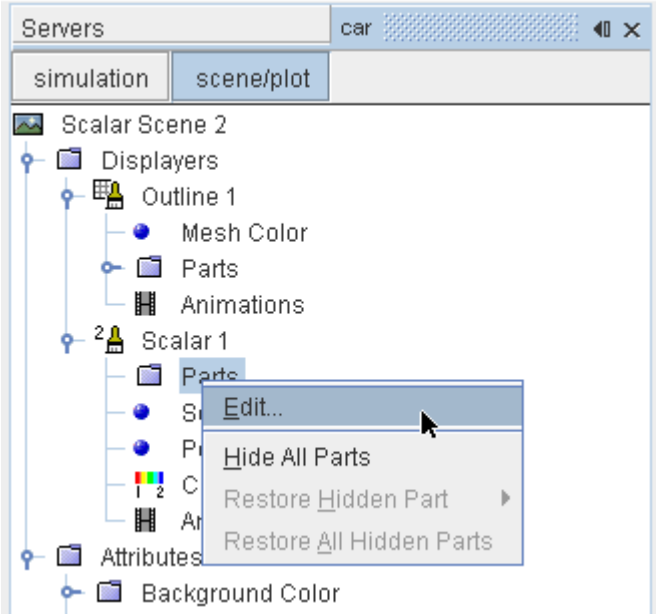
Create a scalar scene to display static pressure.

1. Right-click the **Scenes** node in the explorer pane and select **New Scene** > **Scalar**.



A new scalar scene, *Scalar Scene 2*, is created and selected in the *Graphics* window.

2. Click **scene/plot**.
3. Right-click the **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Parts** node and select **Edit...**.

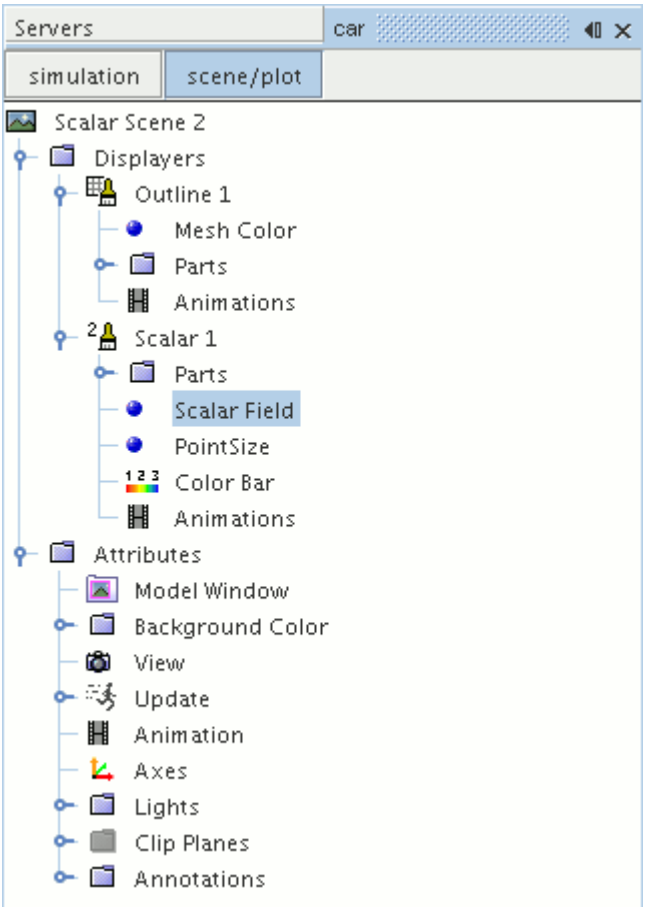


The *Parts* dialog appears.

4. Expand the **Regions** > **Region** > **Boundaries** node and select the following items:
   - **Car**
   - **Ground**
   - **Symmetry Plane**
5. Click **OK**.

Display the pressure contours in the scene:

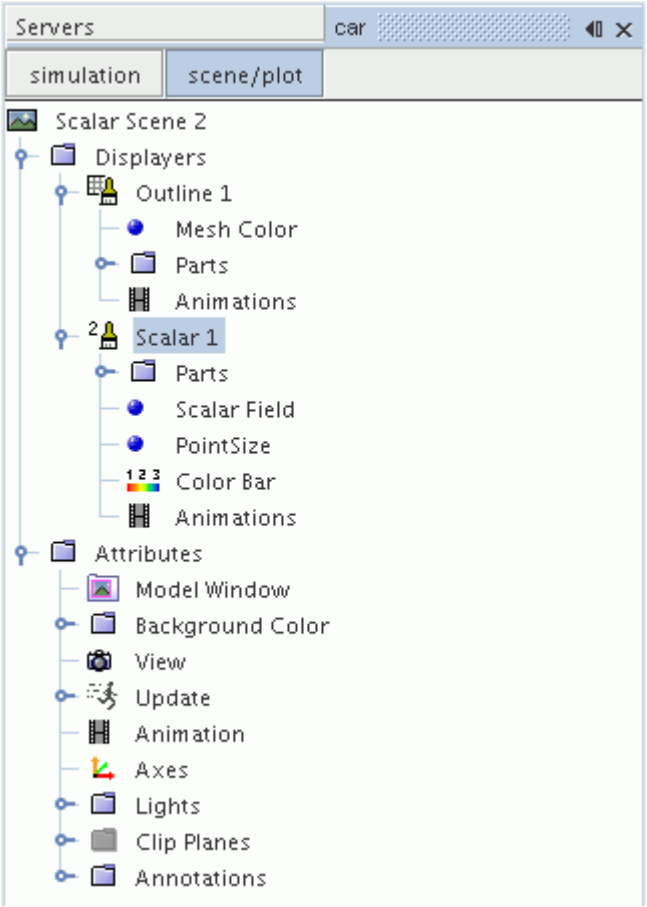
6. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Scalar Field**.



7. Set *Function* to **Static Pressure**.

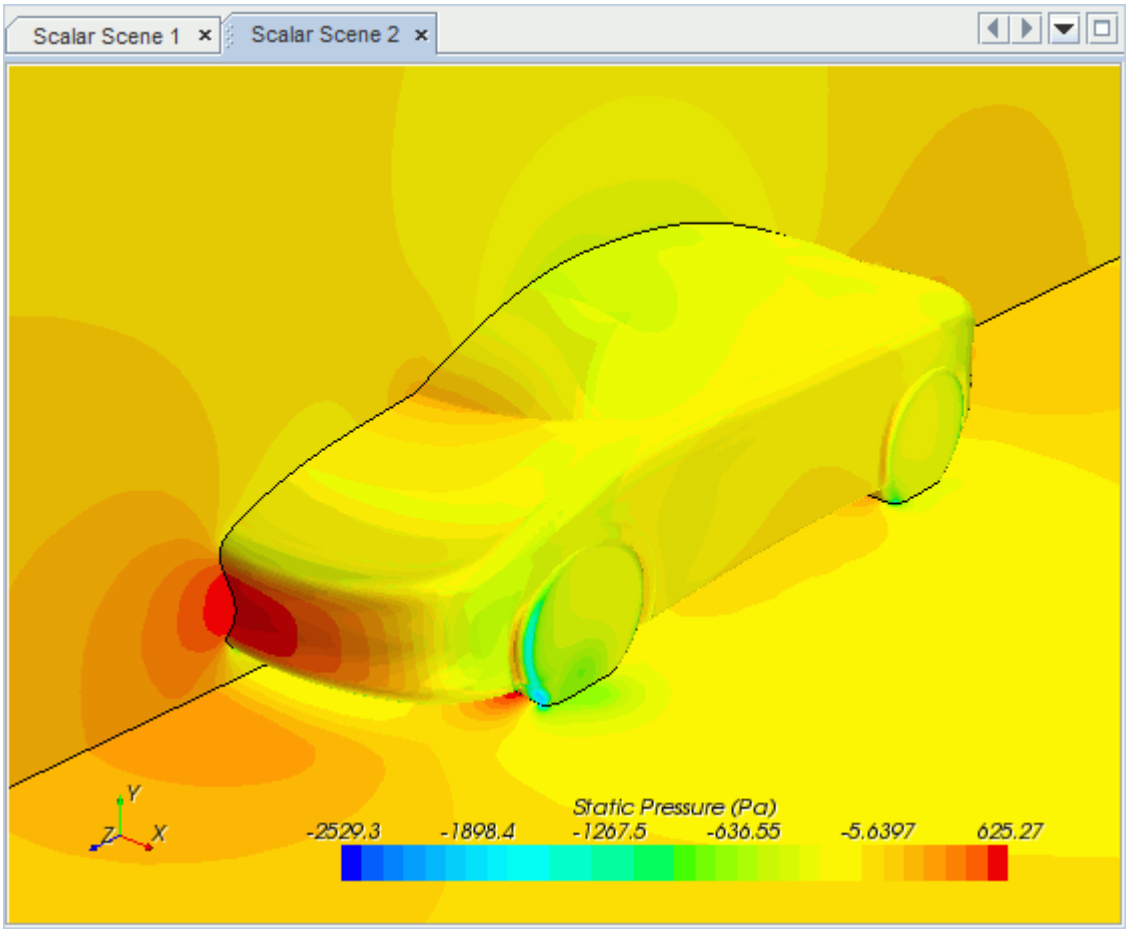
To have a smooth transition between the different pressure values:

8. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1**.



9. Set *Contour Style* to **Smooth Filled**.
10. Rotate the scene to get a better view of the front of the car.
11. Click (**Save**) to save the simulation.

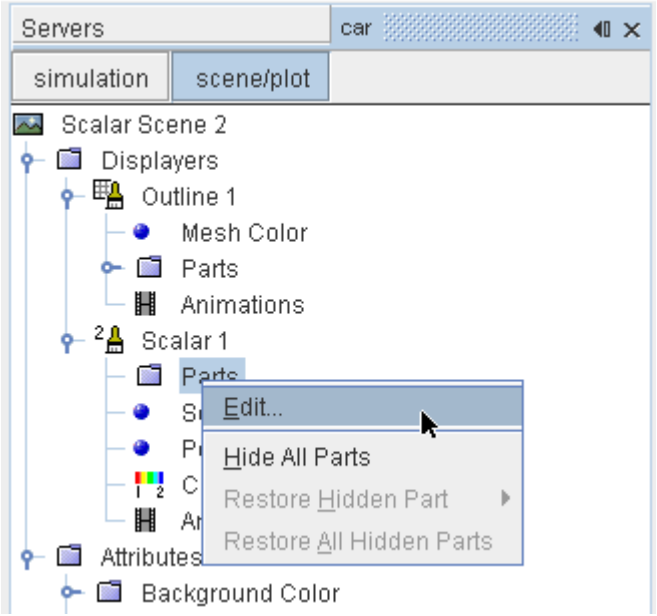
The finished scene is shown below.



Modifying The Scalar Scene

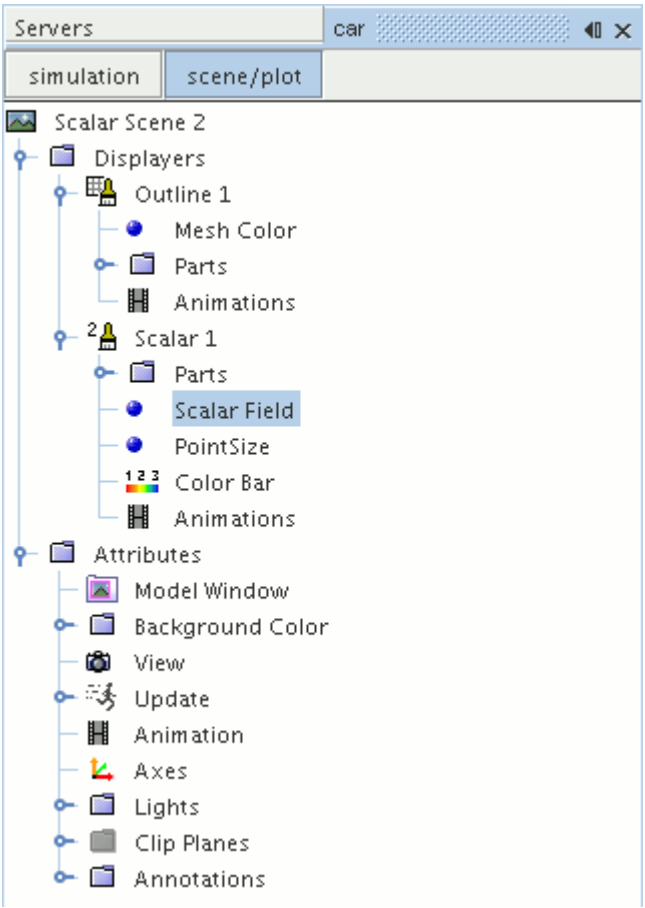
Modify Scalar Scene 2 to display Turbulent Kinetic Energy and remove the symmetry plane from the scene. You also mirror the car body:

1. Right-click the **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Parts** node and select **Edit...**.



The *Parts* dialog appears.

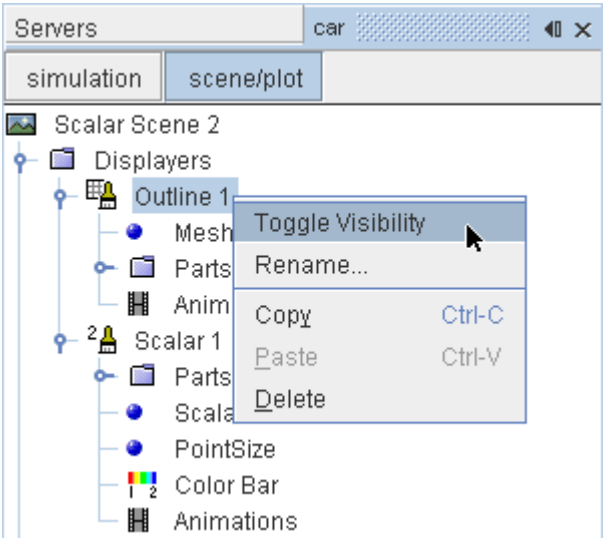
2. Expand the **Regions** > **Region** > **Boundaries** folder and deactivate **Symmetry Plane.**
3. Click **OK**.
4. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Scalar Field**.



5. Set *Function* to **Turbulent Kinetic Energy**.

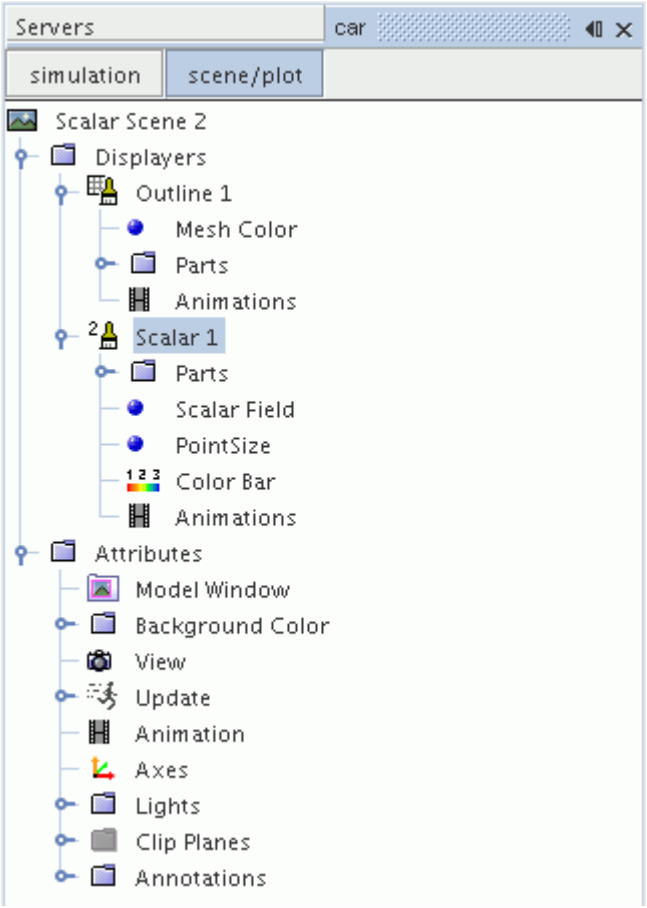
Hide the outlines in the scene:

6. Right-click the **Scalar Scene 2** > **Displayers** > **Outline 1** node and select **Toggle Visibility**.

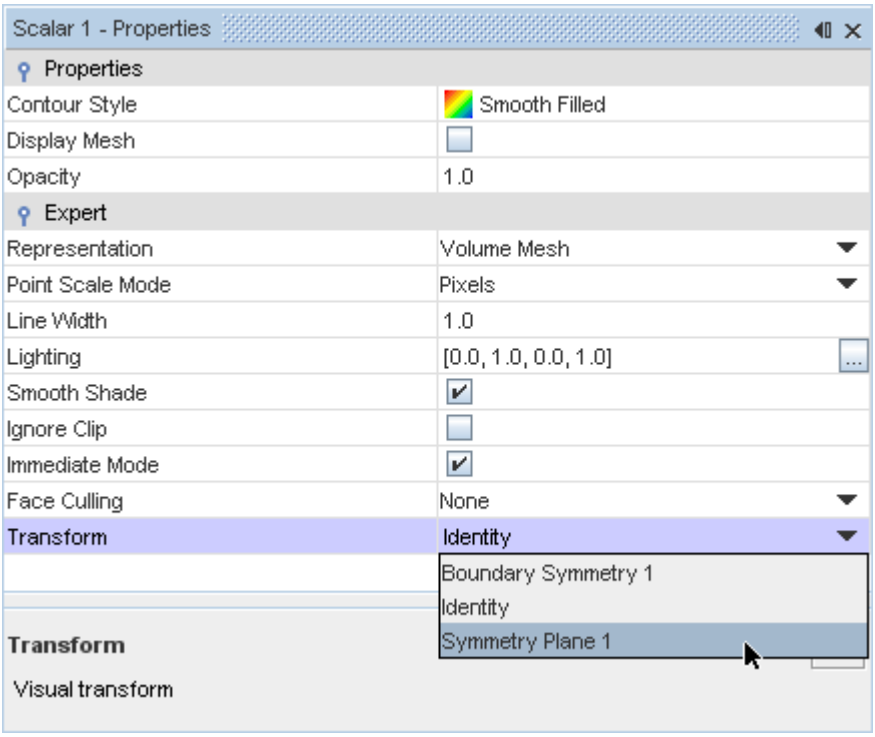


Mirror the car body:

7. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1**.

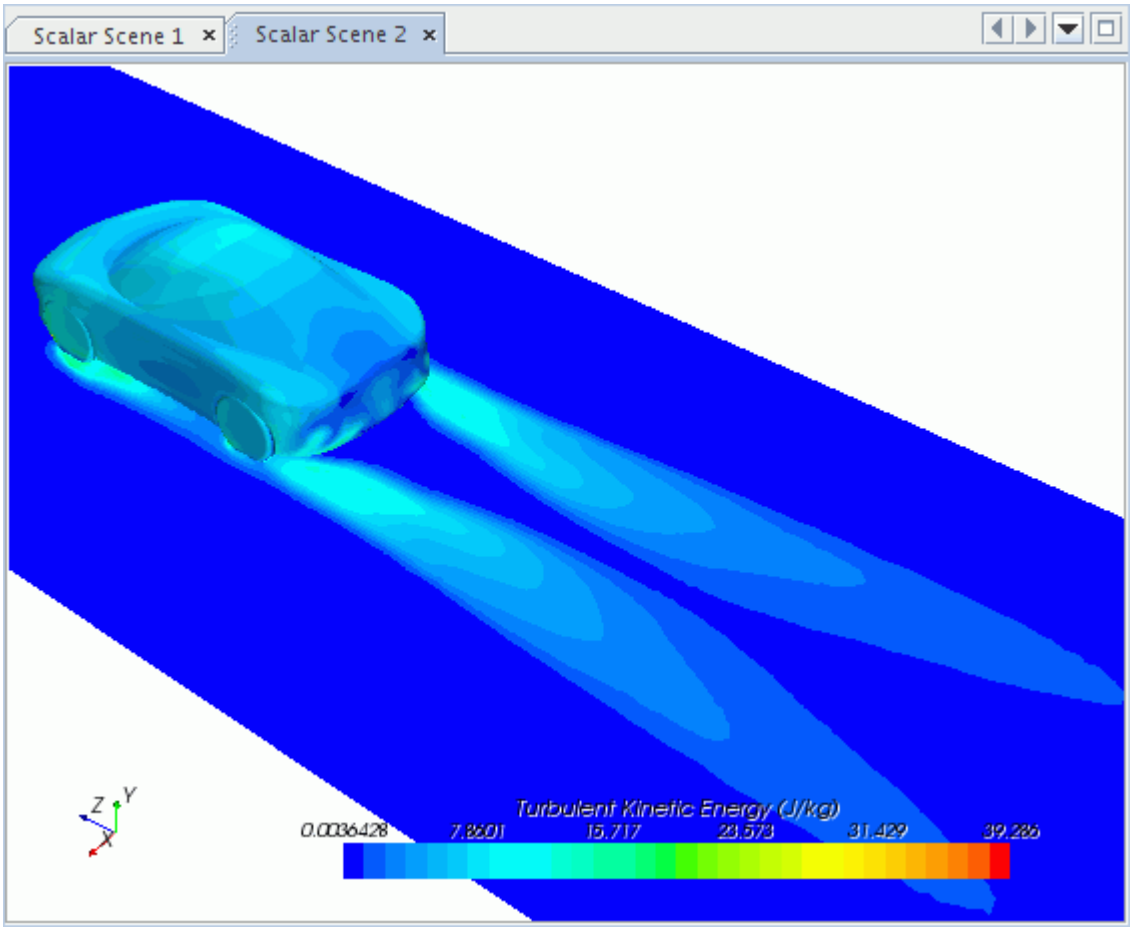


8. Set *Transform* to **Symmetry Plane 1**.



9. Rotate the scene to get a better view of the wake of the car.
10. Click **simulation**.
11. Save the simulation.

The finished scene is shown below.

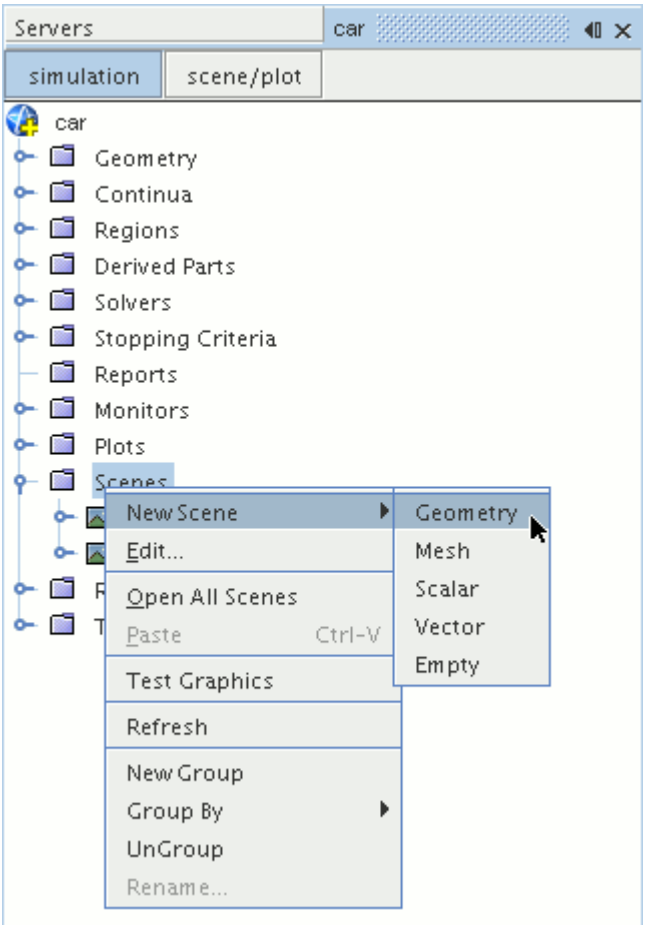


Creating a Streamlines Scene

Create streamlines to get a better understanding of the flow around the car and in the wake area.

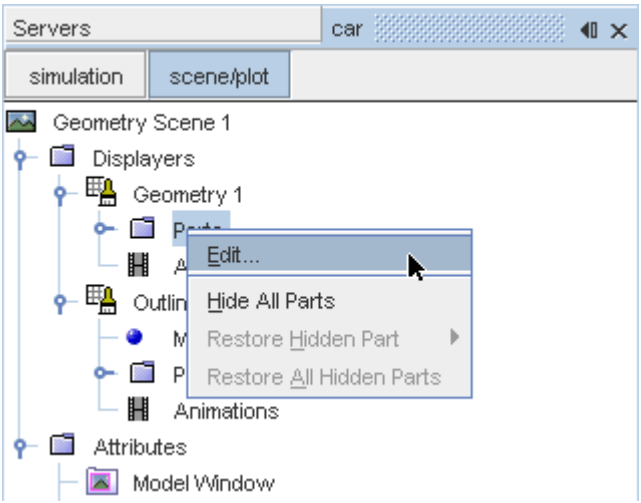
First, create a geometry scene:

1. Right-click the **Scenes** node in the object tree and select **New Scene** > **Geometry**.



A new geometry scene is created and selected in the *Graphics* window.

2. Click **scene/plot**.
3. Right-click the **Geometry Scene 1** > **Displayers** > **Geometry 1** > **Parts** node and select **Edit...**.



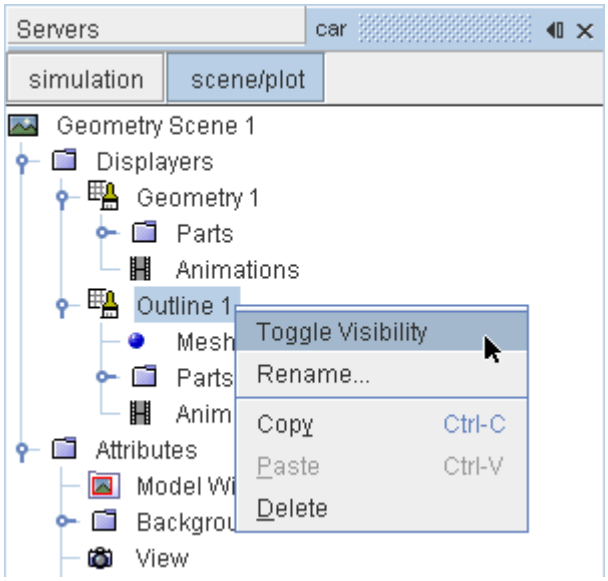
The *Parts* dialog appears.

4. Right-click on a blank area and choose **Deselect All**.

5. Expand the **Regions** > **Region** > **Boundaries** node and select **Car**.
6. Click **OK**.

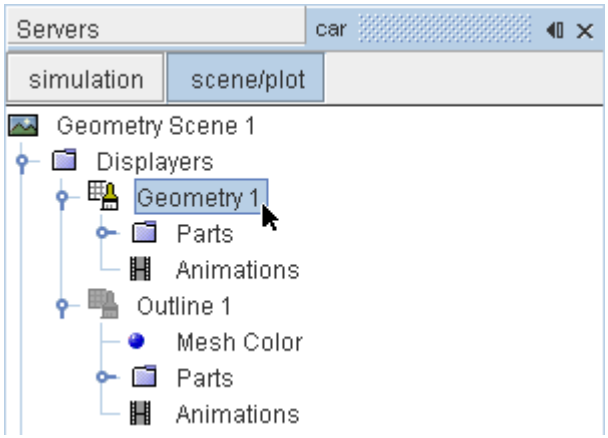
Hide the outlines in the scene:

7. Right-click the **Geometry Scene 1** > **Displayers** > **Outline 1** node and select **Toggle Visibility**.

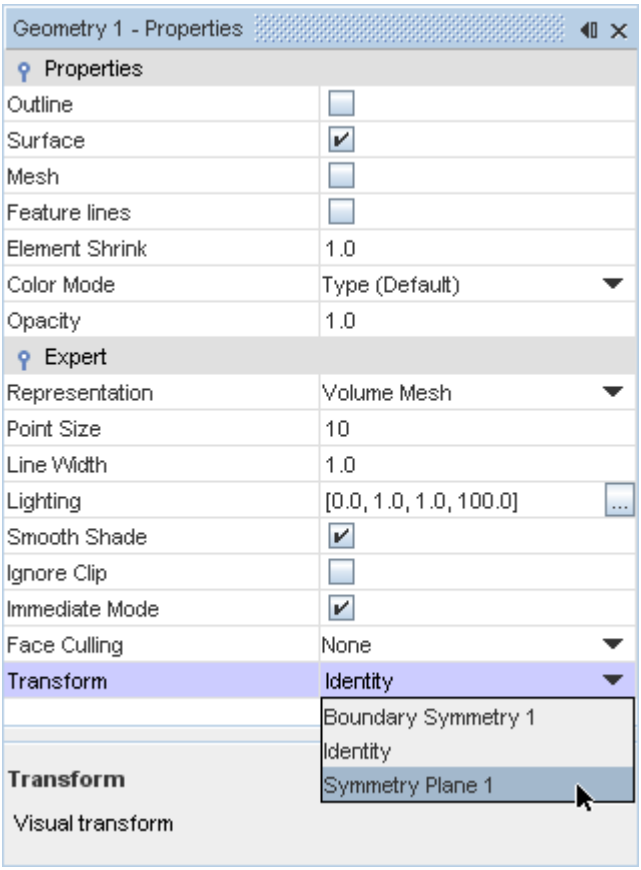


Mirror the car body:

8. Navigate to **Geometry Scene 1** > **Displayers** > **Geometry 1**.



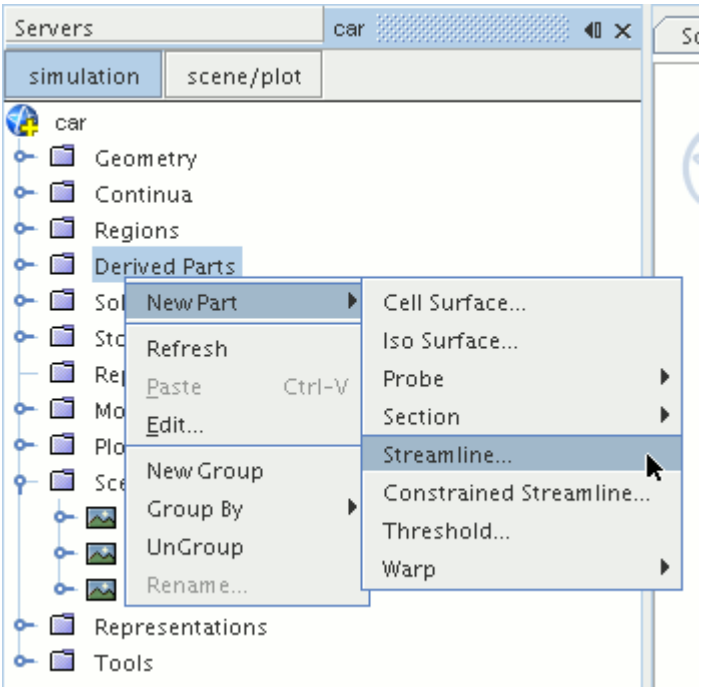
9. Set *Transform* to **Symmetry Plane 1**.



10. Click **simulation**.

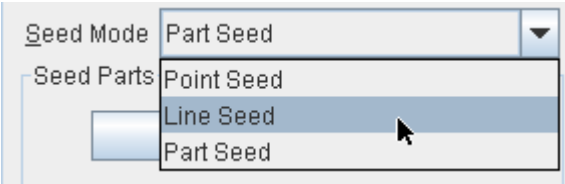
Create the streamlines:

11. Right-click the **Derived Parts** node and select **New Part** > **Streamline...**.



The *Create Streamline* dialog appears.

12. Set *Seed Mode* to **Line Seed**.



13. Enter these coordinates for the line:
    - *Point 1*:
        - X: `0 mm`
        - Y: `-123 mm`
        - Z: `2600 mm`
    - *Point 2*:
        - X: `1200 mm`
        - Y: `-123 mm`
        - Z: `2600 mm`
14. Set *Resolution* to `10`.
15. Make sure that **New Streamline Displayer** is selected in the *Display* section.
16. Click **Create**.

Create another set of streamlines using the existing dialog:

17. Enter these coordinates for the line:
    - *Point 1*:
        - X: `0 mm`
        - Y: `-473 mm`
        - Z: `2600 mm`
    - *Point 2*:
        - X: `1200 mm`
        - Y: `-473 mm`
        - Z: `2600 mm`
18. Click **Create** followed by **Close**.

Modify the appearance of the streamlines to display ribbons instead of the default lines.

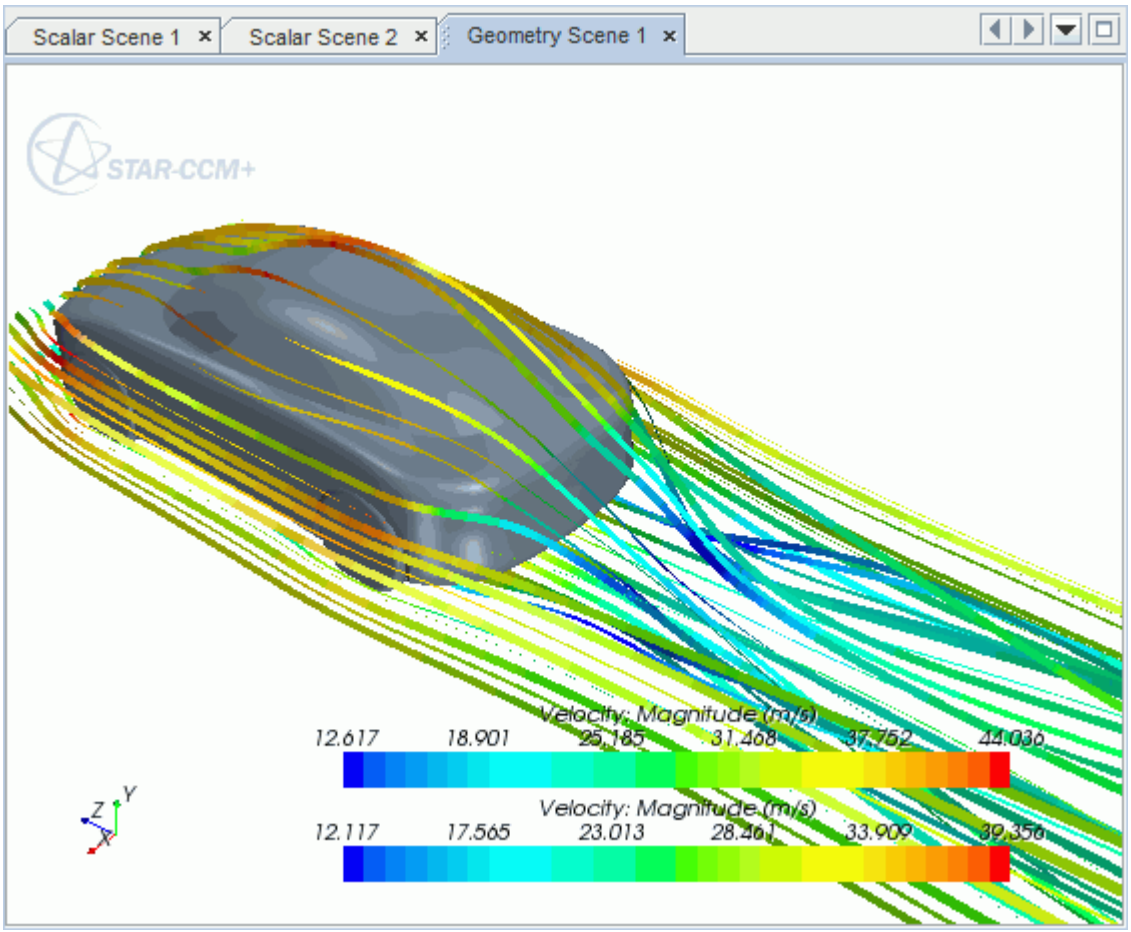
19. Click **scene/plot**.
20. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1**.
21. In the *Properties* window:
    - Set *Width* to `0.025`.
    - Set *Mode* to **Ribbons**.

- Set *Transform* to **Symmetry Plane 1**.

Modify the scalar field of the streamlines to display velocity magnitude:

22. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** > **Scalar Field** and set *Function* to **Velocity** > **Lab Reference Frame** > **Magnitude**.
23. Repeat the above steps for **Streamline Stream 2**.
24. Rotate the scene to get a better view of the wake of the car.
25. Save the simulation.

The finished scene is shown below.



Geometry Transfer: CAD to Simulation

STAR-CAT5 allows you to transfer geometry from CATIA to STAR-CCM+ through a direct link between the two packages.

This feature provides you with a workflow that is tailored to design optimization. You can apply changes to the geometry in either STAR-CAT5 or STAR-CCM+ and the changes are passed back to the other program. For example, you can make changes to the design after running an analysis in the STAR-CCM+ environment and these changes are passed back to the original CATIA file. This feature requires a CAD client license.

This tutorial demonstrates how to select the design parameters in your original geometry and transfer the geometry from CATIA to STAR-CCM+. In STAR-CCM+, you proceed to create a repeatable workflow (or pipeline) using mesh operations. After setting up this pipeline, you can make further changes to the geometry design and have STAR-CCM+ regenerate the corresponding volume mesh.

Contents:

Opening the CAD Model
Transferring the Geometry to STAR-CCM+
Creating the Shape Parts and Operations
Setting Up the Mesh Pipeline
Modifying the Design

Opening the CAD Model

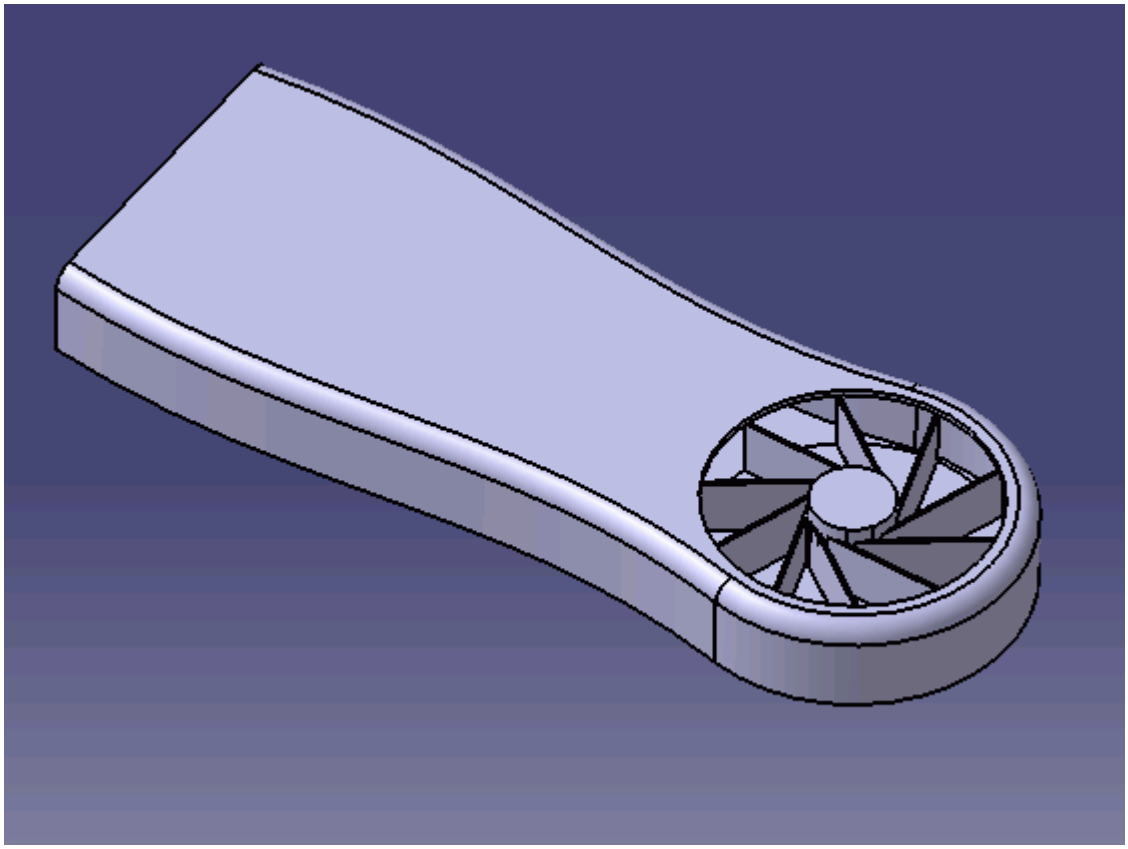
The STAR-CCM+ installation includes a directory containing the CAD geometry for this tutorial.

This directory is located in `[STAR-CCM+_INSTALL_DIR]\doc\startutorialsdata\starcat5\data`.

1. Copy directory `data` to your own disk space.
2. Launch STAR-CAT5 and open the product file `cardAssem.CATProduct`.

   See Launching STAR-CAT5.

   The assembly is made up of two parts: the card housing and the blower fan.
3. Position the geometry as shown below.



The STAR-CAT5 menu appears at the top of the screen. The STAR-CAT5 object tree is located on the left of the screen.

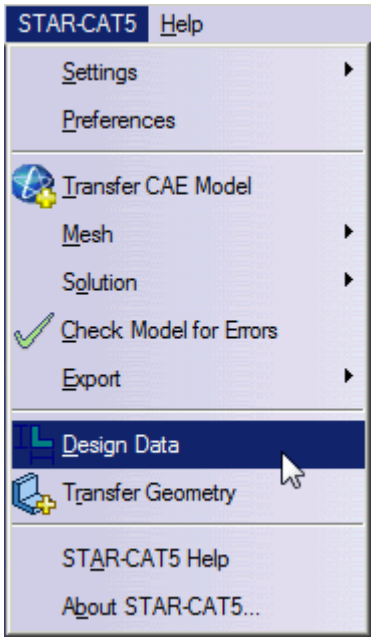
4. Save the analysis as `cardAssem.CATAnalysis`.

Transferring the Geometry to STAR-CCM+

In STAR-CAT5, choose the design parameters to expose to STAR-CCM+ and then transfer the geometry.

Specify the design parameters to transfer to STAR-CCM+:

1. From the STAR-CAT5 menu, click **Design Data**.



The *Design Data* dialog appears.

2. In the *Design Parameters* tab, select **blowerFan\PartBody\CircPattern.1\AngularNumber**.

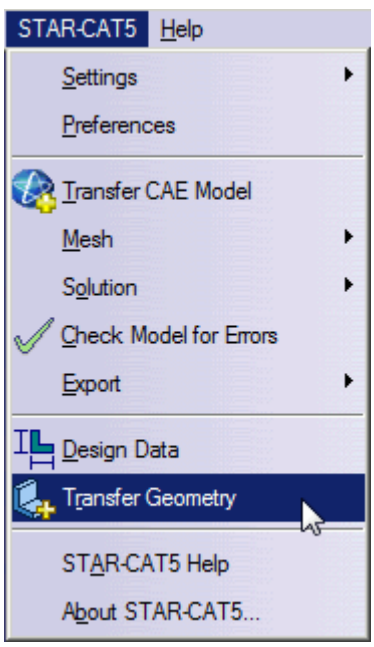
   STAR-CAT5 only recognizes native CAD features as design parameters.

   The value of this parameter is displayed in the *Parameter Value* property.

3. Activate the *Transfer to STAR-CCM+* option.
4. Click **OK**.

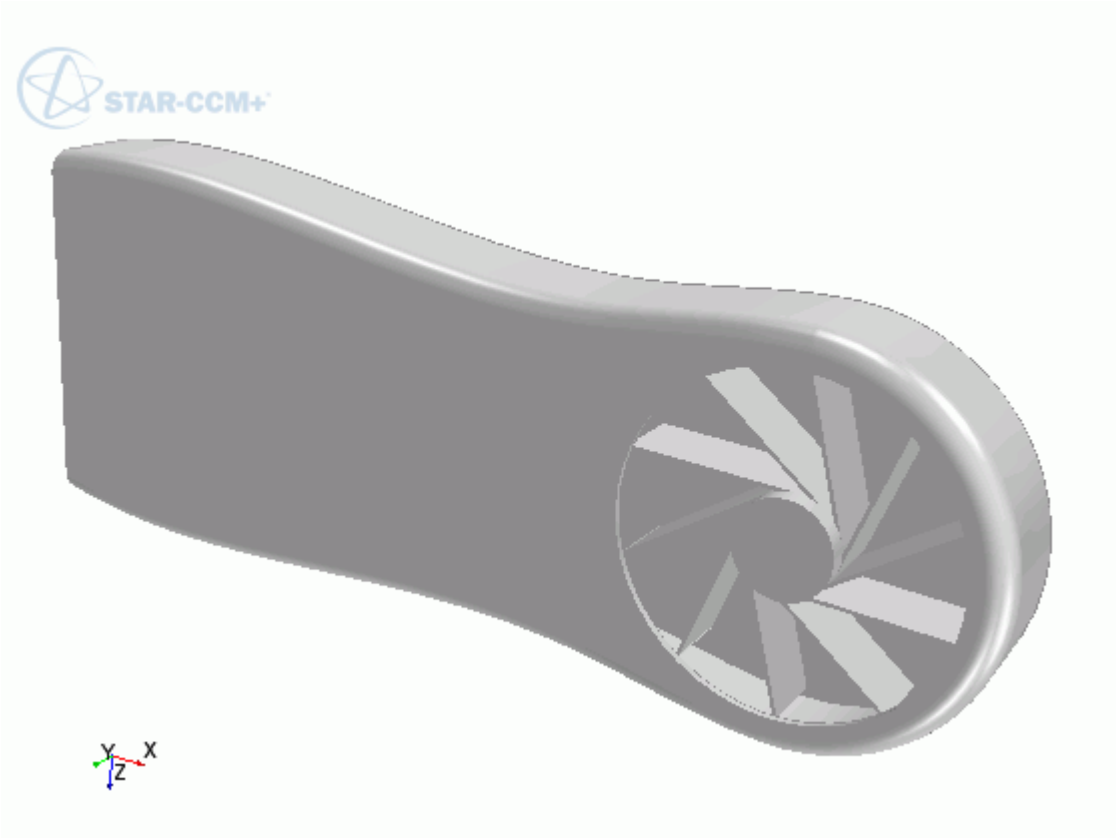
Transfer the geometry to STAR-CCM+:

5. From the STAR-CAT5 menu, select **Transfer Geometry**.



STAR-CCM+ launches and the CAD geometry is transferred over.

6. In STAR-CCM+, create a geometry scene and position the geometry as shown below.



7. In STAR-CCM+, expand the **Geometry** node.

   The transferred geometry appears under the **Parts** node. The properties that control the connection back to CATIA, including the design parameter value, appear under the **CAD Client Models** node.

When transferring geometry from CATIA to STAR-CCM+, a link exists between the two software packages. This link allows data to be transferred between the two packages. To maintain this link, you must first save the geometry file in CATIA:

8. Switch to CATIA and save the geometry file.

   In response, STAR-CCM+ saves the simulation file with the name of the CATIA geometry file. Using this technique, you can rename or relocate both the geometry and simulation files without breaking the link between the two packages. You can also save the simulation file in STAR-CCM+ so long as you do not rename it or relocate it. Doing so will break the link between the packages.

Creating the Shape Parts and Operations

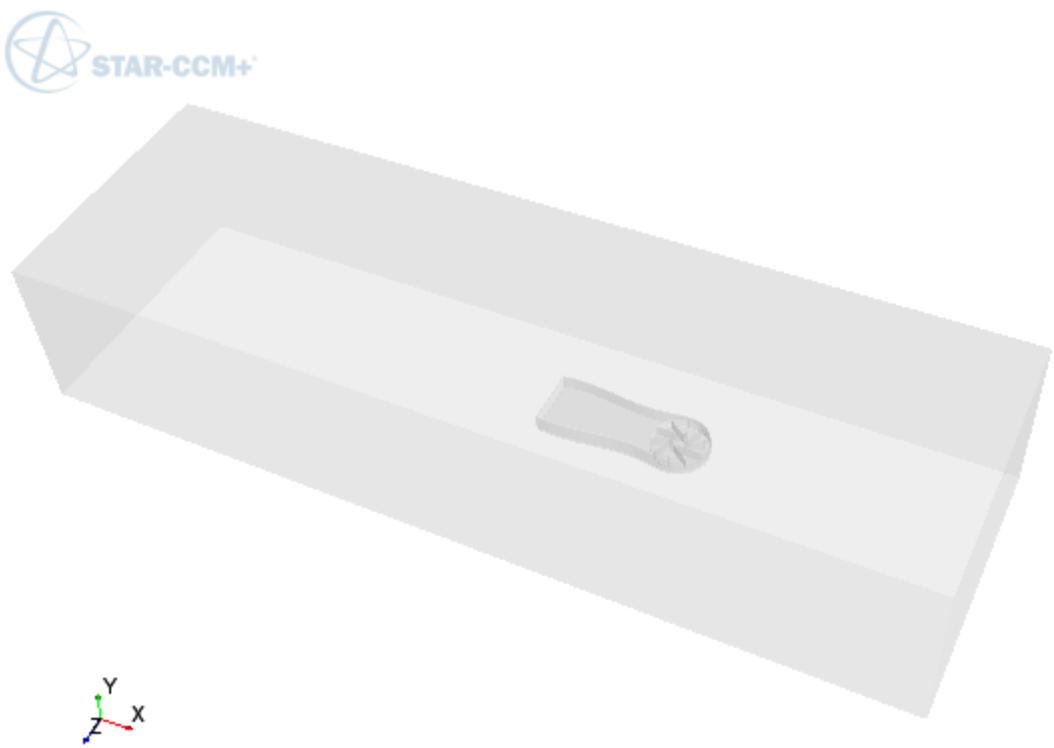
The transferred geometry consists of the card housing and the blower fan. To simulate this case, you require two regions: one static region that provides an enclosure for the simulation, and another for the rotating region around the fan.

The transferred geometry contains the solid parts. You require the fluid volume around these parts. The first stage in obtaining the fluid volume is to create part shapes that correspond to the two regions. To save time, you are provided with these part shapes, one for the enclosure, and another for the rotating region.

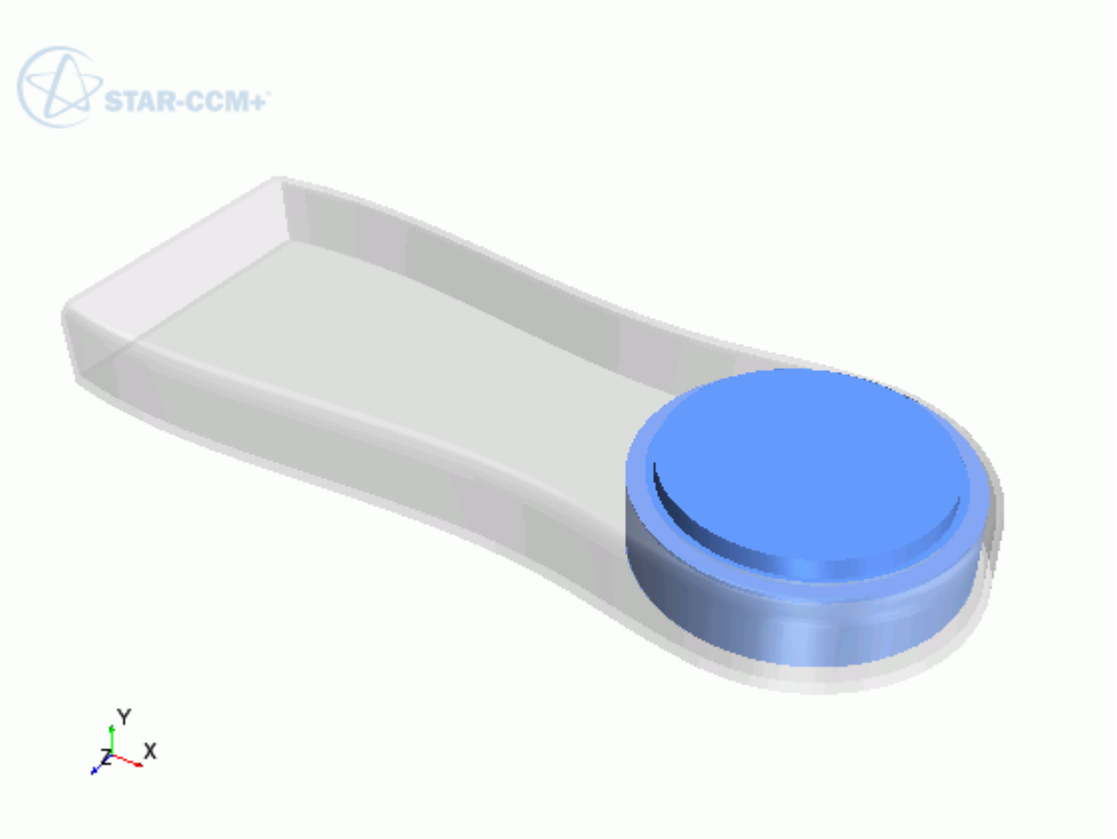
Import the shape parts:

1. Select **File** > **Import** > **Import Surface Mesh**.
2. Import the part shapes:
   a) In the *Open* dialog, navigate to the `data` directory and select **box.x_t** and **disc.x_t**.
   b) Click **Open**.
   c) In the *Import Surface Options* dialog, deactivate the **Open Geometry Scene After Import** option.
   d) Click **OK**.

   The static region, **box**, is shown below in relation to the graphics card:



The rotating region, **disc**, is show below in relation to the graphics card:



Now subtract the solid CAD geometry from these shape parts to obtain the final fluid volume. Begin by subtracting the blower fan geometry from the **disc** part:

3. Right-click the **Operations** node and select **New** > **Boolean** > **Subtract**.
4. In the *Create Subtract Operation* dialog, set the following properties:

Property	Setting
Input Parts	**disc**
	cardAssem > **blowerfan**
Target Parts	**disc**

5. Click **OK**.
6. Rename the **Operations** > **Subtract** node to `Fan Subtraction`.
7. Rename **Parts** > **Subtract** node to `Rotating Region`.

Subtract the card housing geometry and the disc part from the static region:

8. Create another subtract operation with the following settings:

Property	Setting
Input Parts	**box**
	disc
	cardAssem > **cardCase**

Property	Setting
Target Parts	**box**

9. Rename **Operations** > **Subtract** node to `Card Subtraction`.
10. Rename **Parts** > **Subtract** node to `Static Region`.

To get a conformal mesh between the two regions, imprint both fluid volumes together:

11. Right-click the **Operations** node and select **New** > **Boolean** > **Imprint**.
12. In the *Create Imprint Operation* dialog, select **Rotating Region** and **Static Region** and click **OK**.

The transferred geometry contains CAD data. To ensure best results, instruct all operations to use the CAD data where available:

13. Under the **Operations** node, multi-select the **Fan Subtraction** and **Card Subtraction** nodes and activate the *Perform CAD Boolean* option.
14. Select the **Operations** > **Imprint** node and set *Merge/Imprint Method* to **CAD Imprint**.

    The contacts between the two parts are created. These contacts are used while creating the regions to define the interfaces between the regions.

Execute all operations:

15. Right-click on the **Operations** node and select **Execute All**.

Setting Up the Mesh Pipeline

Assign the parts to regions and create the automated mesh operation.

Assign the fluid parts to regions:

1. Expand the **Geometry** > **Parts** node and multi-select the **Rotating Region** and **Static Region** nodes.
2. Right-click on one of the selected parts and choose **Assign Parts to Regions...**.
3. In the *Assign Parts to Regions* dialog:
   a) Select **Create a Region for Each Part**.
   b) Select **Create Boundary for Each Part Surface**.
   c) Click **Apply** and **Close**.

Create the automated mesh operation:

4. Right-click the **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
5. In the *Create Automated Mesh Operation* dialog:
   a) From the *Parts* list, select **Rotating Region** and **Static Region**.
   b) Select the following meshers, in order:

Group	Mesher
Surface Meshers	**Surface Remesher**
Volume Meshers	**Polyhedral Mesher**
Optional Boundary Layer Meshers	**Prism Layer Mesher**

c) Click **OK**.

An automated mesh operation is added to the **Operations** node. Refine the mesh settings for this case:

6. Expand the **Automated Mesh** > **Default Controls** node.
7. Select the **Base Size** node and set *Value* to `16 mm`.
8. Select the **Prism Layer Total Thickness** > **Relative Size** node and set *Percentage of Base* to `20`.

Generate the volume mesh:

9. Click (**Generate Volume Mesh**).
10. When the volume mesh is generated, create a mesh scene that displays the following parts:
    - **Rotating Region** > **Boundaries** > **cardAssem.blowerfan.Faces**
    - **Static Region** > **Boundaries** > **cardAssem.cardcase.Faces**

You can exit STAR-CCM+ and CATIA and maintain the link between the two packages. To demonstrate, in the following steps you save the simulation and exit both applications. In the next section, you will resume the simulation and continue where you left off.

11. Save the simulation.
12. Exit STAR-CCM+.
13. Exit CATIA

Modifying the Design

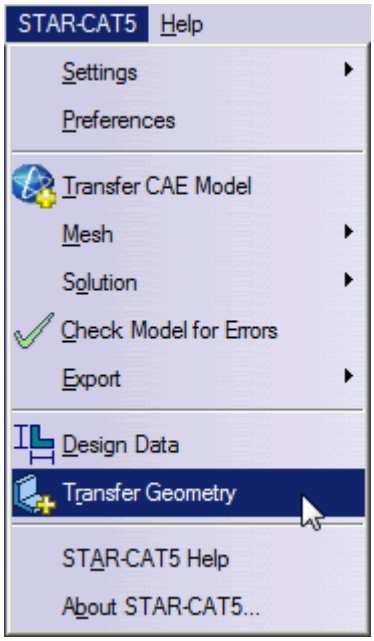
Modify the number of blades on the blower fan using the design parameter.

In this tutorial, you modify the number of blades on the graphics card blower fan from 10 to 15 blades.

To re-establish the link between CATIA and STAR-CCM+, you must load the geometry in STAR-CAT5 first and transfer the geometry to STAR-CCM+.

Resume the analysis and re-establish the link between STAR-CAT5 and STAR-CCM+:

1. Launch CATIA with STAR-CAT5 and load `cardAssem.CATAnalysis`.
2. From the STAR-CAT5 menu, select **Transfer Geometry**.



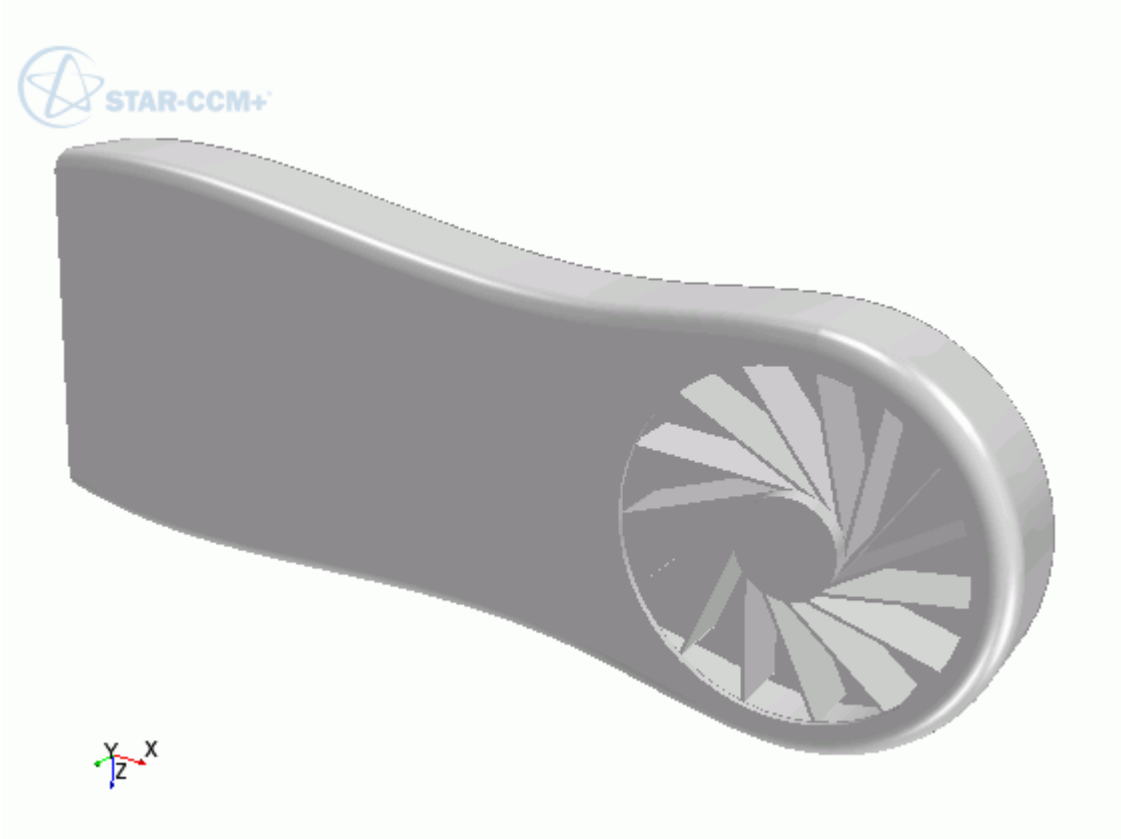
   STAR-CCM+ launches and the CAD geometry is transferred over.
3. Switch to STAR-CCM+.
4. Open the geometry scene to visualize the initial geometry representation.
5. Expand the **Geometry** > **CAD Client Models** > **cardAssem.CATAnalysis** > **Design Parameters** node.
6. Select the **blowerFan\PartBody\CircPattern.1\AngularNumber** node and set *Parameter Value* to `15`.

   A yellow warning icon appears on the design parameter node as well as the assembly node. This icon indicates that the CAD Client Model is out of date.

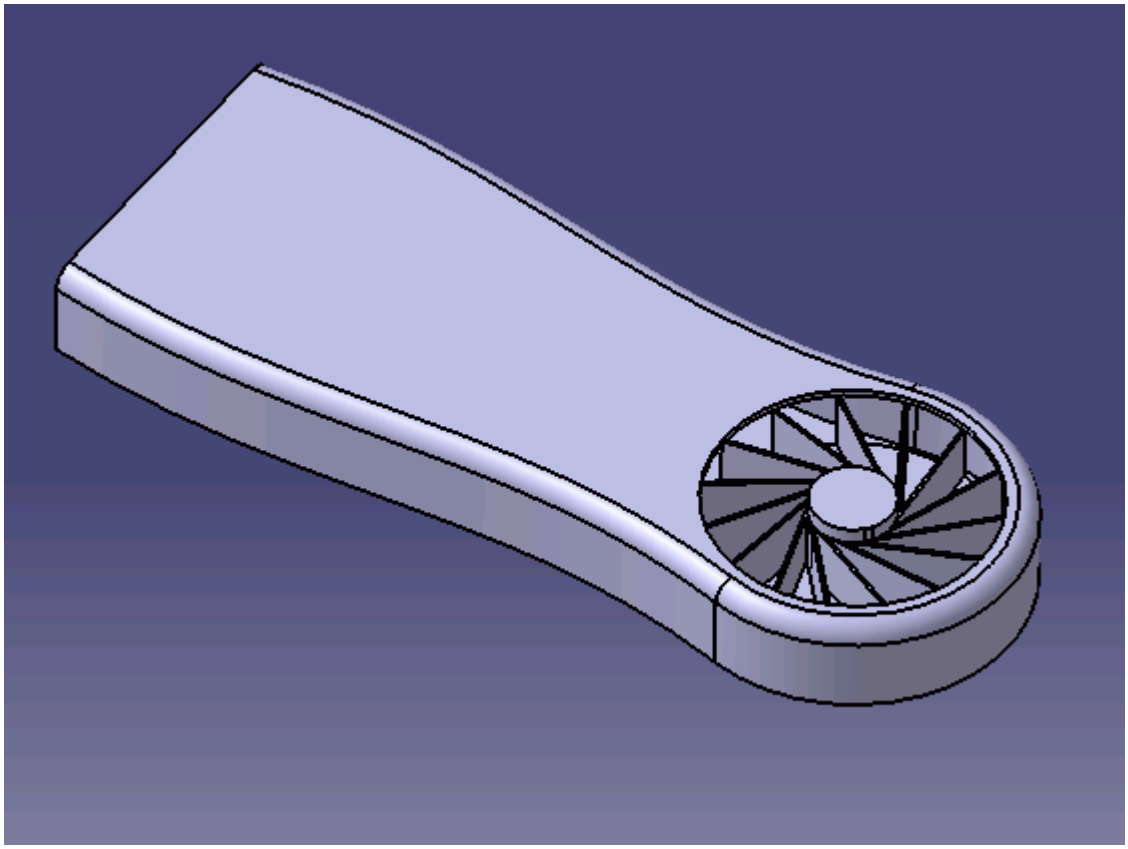
To update the model:

7. Right-click the **cardAssem.CATAnalysis** node and select **Update Model**.

   The geometry updates in the scene. STAR-CCM+ passes the change in the design back to CATIA.



8. Switch back to CATIA and view the updated geometry.



The pipeline you set up is fully repeatable. To update the volume mesh:

9. In STAR-CCM+, open the mesh scene.

10. Click (**Generate Volume Mesh**).
    The volume mesh updates to reflect the changes in the design.

All boundaries and boundary conditions that you set are maintained after this change. This feature reduces the amount of work that is required when analyzing multiple designs.

Siemens PLM Software

Headquarters
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

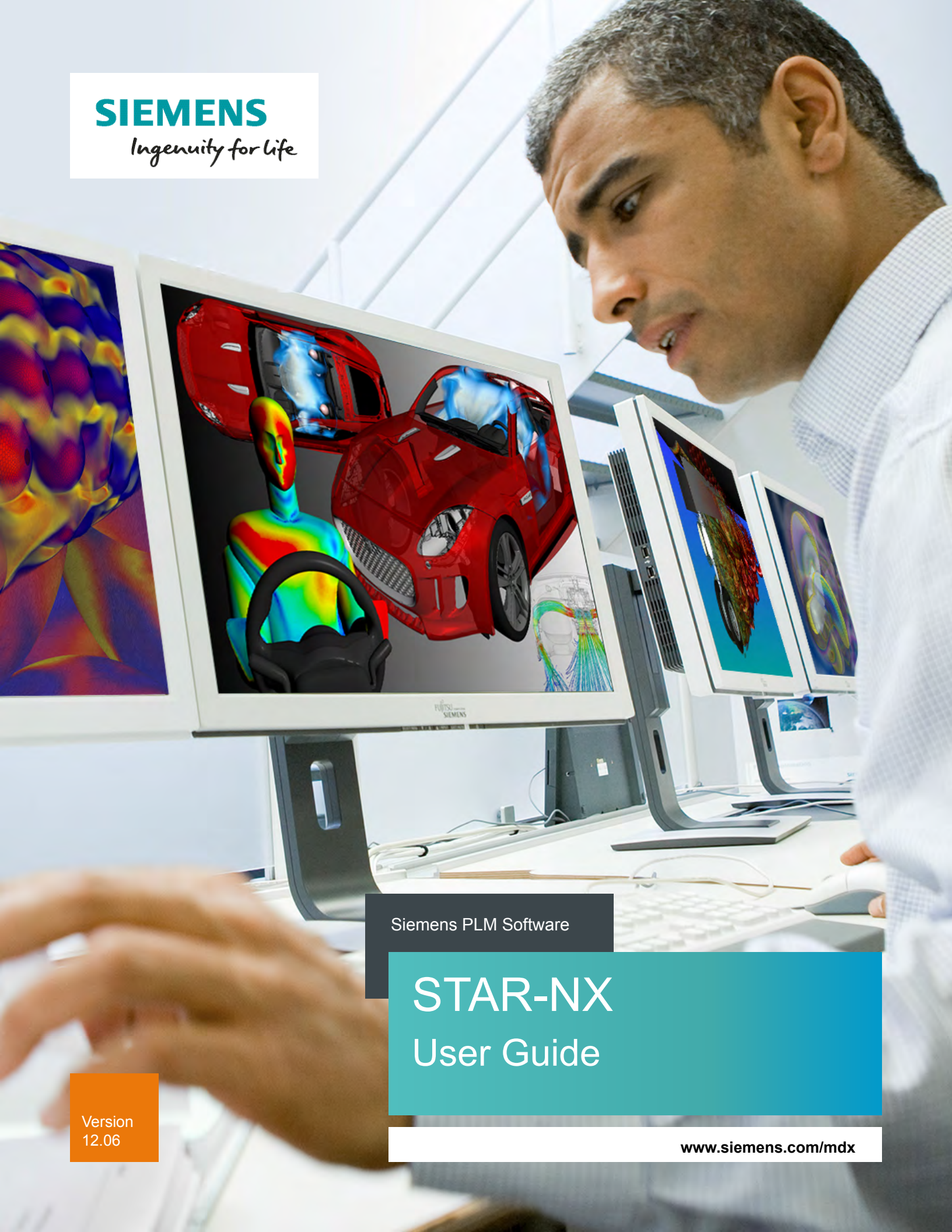
Asia-Pacific
Suites 4301-4302, 43/F
AIA Kowloon Tower,
Landmark East
100 How Ming Street
Kwun Tong, Kowloon
Hong Kong
+852 2230 3308

About Siemens PLM Software

Siemens PLM Software, a business unit of the Siemens Digital Factory Division, is a leading global provider of product lifecycle management (PLM) and manufacturing operations management (MOM) software, systems and services with over 15 million licensed seats and more than 140,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software works collaboratively with its customers to provide industry software solutions that help companies everywhere achieve a sustainable competitive advantage by making real the innovations that matter. For more information on Siemens PLM Software products and services, visit www.siemens.com/plm.

© 2017 Siemens Product Lifecycle Management Software Inc. Siemens and the Siemens logo are registered trademarks of Siemens AG. D-Cubed, Femap, Fibersim, Geolus, GO PLM, I-deas, JT, NX, Parasolid, Polarion, Solid Edge, Syncrofit, Teamcenter and Tecnomatix are trademarks or reg istered trademarks of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries. All other logos, trademarks, registered trademarks or service marks belong to their respective holders.

SIEMENS
Ingenuity for life



Siemens PLM Software

STAR-NX
User Guide

Version
12.06

www.siemens.com/mdx

© 2017 Siemens Product Lifecycle Management Software Inc.

Any and all Siemens Product Lifecycle Management Software Inc. brand, product, service and feature names, logos and slogans are registered trademarks or trademarks of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries. All other brand, product, service and feature names or trademarks are the property of their respective owners.

Contents

STAR-NX **1**

STAR-NX User Guide **2**

Introducing STAR-NX 3
Transferring Geometry to STAR-CCM+ 36
Understanding the STAR-NX Process 44
Connecting to Remote Machines 48
Defining Materials and Physics 52
Setting Boundary Conditions 74
Meshing the Model 81
Obtaining a Solution 102
Exporting Files and Data 111
Resetting STAR-NX Settings 115
Guidelines 116

STAR-NX Tutorials **150**

Cold Flow Through a Motorcycle Exhaust Manifold 151
Porous Media Flow 174
Aircraft Wing Deicing System 196
Rotating Flow Inside a Graphics Card 229
External Flow Over a Car 259
Geometry Transfer: CAD to Simulation 290

STAR-NX

The STAR-NX CAD client provides you with straightforward access to CD-adapco's STAR-CCM+ solver from within the NX environment.

Standard physics models, including turbulence, ideal gas, heat transfer, and gravity are supported for single body cases. Multiple body simulations involving conjugate heat transfer, rotating reference frames, and porous media are also supported. Advanced physics models can be added to the simulation by either launching STAR-CCM+ and selecting them directly, or by implementing additional Java macro commands.

This section contains the user guide and tutorials for the STAR-NX CAD client.

Contents:

STAR-NX User Guide
STAR-NX Tutorials

STAR-NX User Guide

This part of the documentation details how to create, set up, and run a Computational Fluid Dynamics (CFD) simulation in STAR-NX.

This section assumes that you have previous experience in using NX, and only describes the additional features available as part of STAR-NX.

You can use STAR-NX to transfer your CAD geometry directly into STAR-CCM+ without transferring the CAE settings. This method is an alternate way importing geometry to using the STAR-CCM+ CAD translators. See Transferring Geometry to STAR-CCM+.

Contents:

Introducing STAR-NX
Transferring Geometry to STAR-CCM+
Understanding the STAR-NX Process
Connecting to Remote Machines
Defining Materials and Physics
Setting Boundary Conditions
Meshing the Model
Obtaining a Solution
Exporting Files and Data
Resetting STAR-NX Settings
Guidelines

Introducing STAR-NX

STAR-NX provides you with straightforward access to CD-adapco's STAR-CCM+ solver from within the NX environment.

Standard physics models, including turbulence, ideal gas, heat transfer, and gravity are supported for single body cases. Multiple body simulations involving conjugate heat transfer, rotating reference frames, and porous media are also supported. Advanced physics models can be added to the simulation by either launching STAR-CCM+ and selecting them directly, or by implementing additional Java macro commands.

Contents:

What are the Capabilities?
Getting Started
Common Functions
Running STAR-NX and STAR-CCM+ on Different Machines

See Also:

Using Java Macros in STAR-NX

What are the Capabilities?

This section highlights the capabilities of STAR-NX.

STAR-NX enables you to:

- Define flow characteristics for the simulation
- Specify mesh parameters and controls
- Execute the STAR-CCM+ solver within STAR-NX, with specified solution controls
- Post-process results in STAR-CCM+
- Use parallel processing

Problem Definition

STAR-NX provides an interface for the specification of continua, regions, and boundaries in order to define the CFD simulation. This information is then used by STAR-CCM+ to calculate a solution and present results.

The particular solver options that you can currently select through the STAR-NX interface include:

- Turbulence
  - k-e Model
  - Laminar
- Temperature
  - Thermal Modelling
  - Conjugate Heat Transfer

- Equation of State
  - Constant Density
  - Ideal Gas
- Porous Media
  - Isotropic
  - Uni-Directional
- Moving Reference Frames
- Energy Source
- Gravity
- Passive Scalar

Meshing

STAR-NX provides you with a transparent interface to CD-adapco's surface and volume meshing tools. This allows you to generate high-quality meshes using a solid model that was built within the NX environment. The specific meshing functionality includes:

- Surface meshing using the surface remesher for optimum mesh quality
- Prismatic subsurface meshing in the volume mesh for capturing near-wall boundary layers
- Volume meshing of parts with either:
  - Tetrahedral core mesh generation
  - Polyhedral core mesh generation
  - Trimmed hexahedral core mesh generation
- Automatic handling of region-to-region interfaces

Running the Solver

STAR-NX allows the generation of solutions with varying accuracy and speed by providing the following solver controls:

- Number of iterations
- Stopping criterion at which to assume convergence
- Discretization scheme (1st or 2nd order)
- Solver under-relaxation factors
- Additional commands using Java macros

Post-processing

As post-processing is carried out in STAR-CCM+, a wide range of tools are available for analysis. These include:

- Scalar scenes
- Vector scenes
- Streamlines
- Isosurfaces
- Plane sections

- Reports
- Plots

Parallel Processing

STAR-NX supports parallel processing for mesh generation and solver calculations. You can set up parallel processing in the Preferences dialog. The different STAR-CCM+ licensing options can also be selected here. See Setting Up Parallel Processing and Licensing.

Getting Started

The following section provides information on familiarization with the STAR-NX interface, and basic concepts.

Contents:

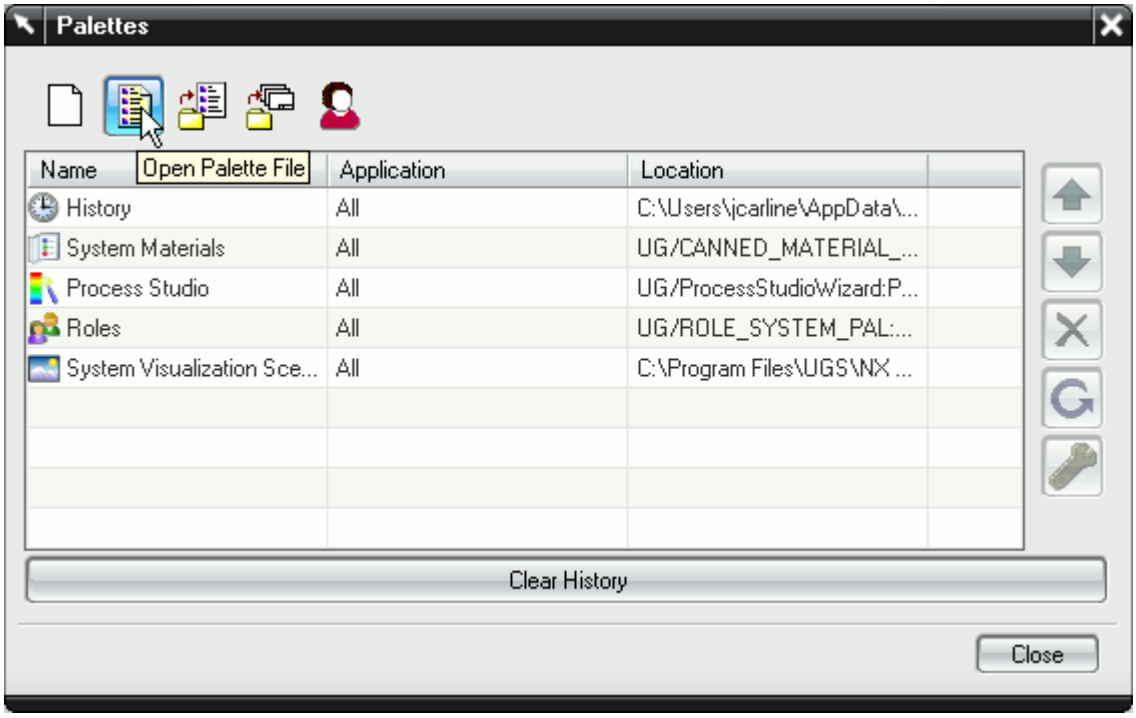
Launching STAR-NX Manually
The STAR-NX Info App Window
Modifying the config.xml File
Understanding the STAR-NX Interface
The STAR-NX Object Tree
Setting Up Parallel Processing and Licensing
Changing the Units System
Transferring the CAE Model to STAR-CCM+
Using NX Mouse Controls in STAR-CCM+
Editing STAR-NX Settings While the STAR-CCM+ Client is Running
Using STAR-NX with Multiple Installed Versions of NX
Disabling STAR-NX

Launching STAR-NX Manually

By default, STAR-NX starts automatically once you launch NX. The STAR-NX object tree icon appears on the left hand side of the screen when you open a part or assembly in NX.

If this icon does not appear, follow the steps outlined below to manually create it:

1. Start NX.
2. Select **Preferences** > **Palettes** from the menu bar.
3. In the *Palettes* dialog, click **Open Palette File**.

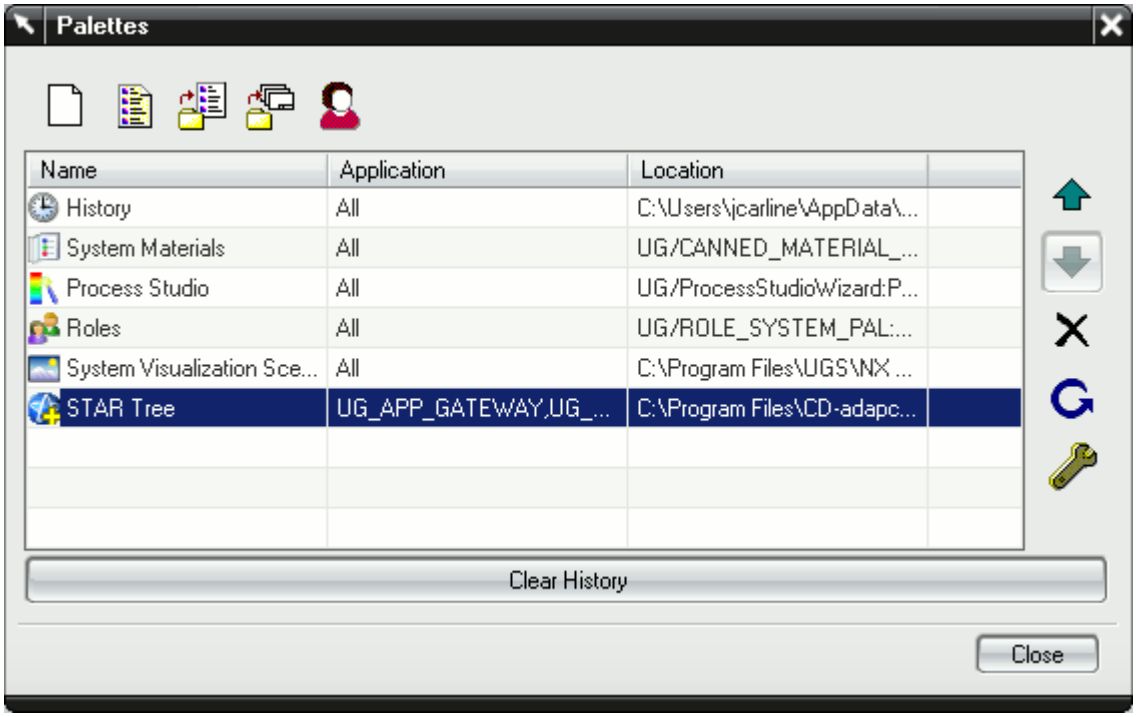


4. Browse to and open the `starnx.pax` palette file.

   Typically this file is located in `C:\Program Files\CD-adapco\STAR-CAD[VERSION]\STAR-NX\win64\bin\star-nx[VERSION]\startup`.

5. Click **OK** to load the palette.

   The palette file appears in the *Palettes* dialog.



6. Click **Close**.

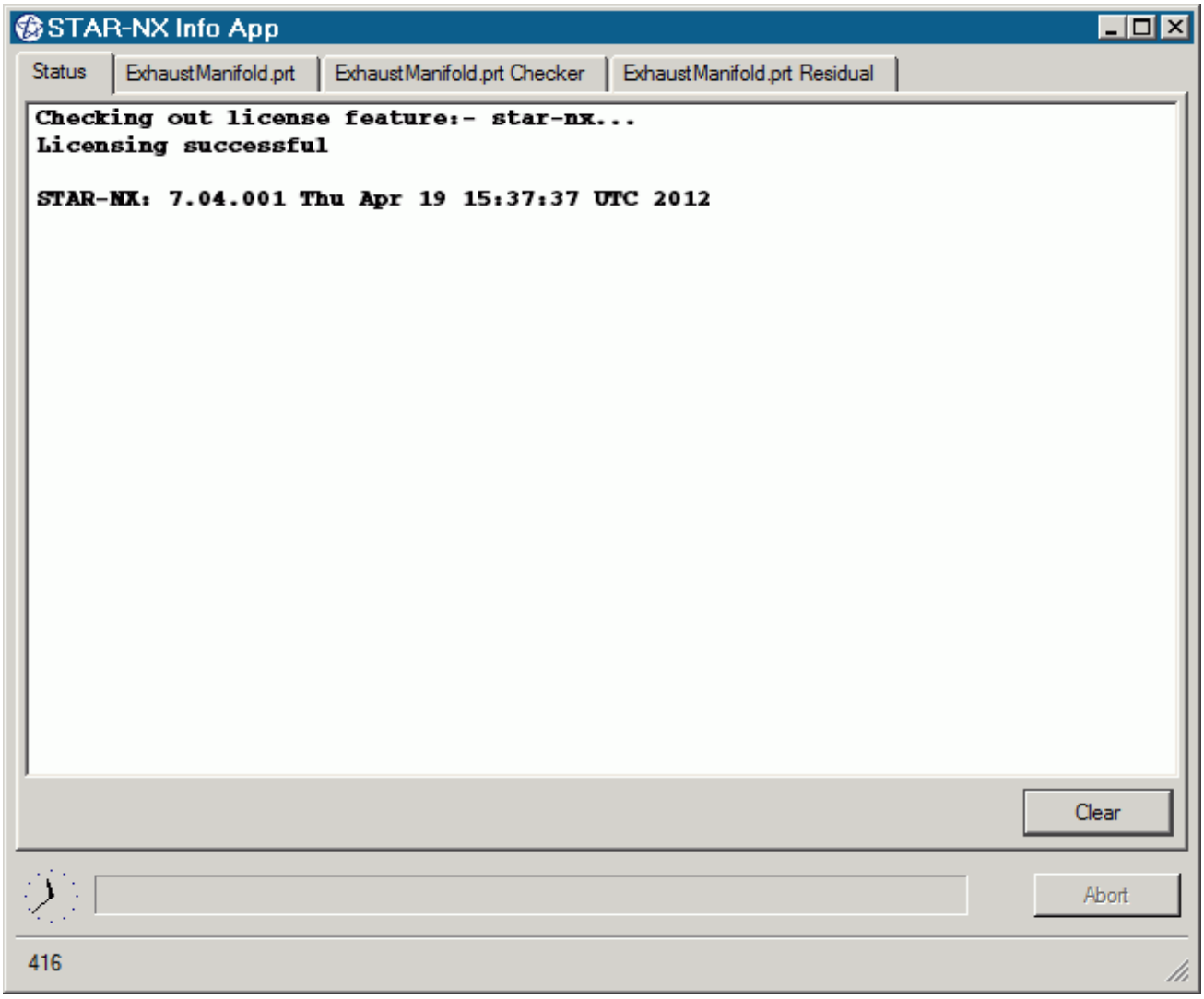
The (STAR-NX) icon appears on the left of the screen when you load or create a geometry file. Click on this icon to display the object tree. The STAR-NX ribbon also appears.

Note: You may need to click on the icon twice for the tree to appear if you just loaded the palette.

The STAR-NX Info App Window

The STAR-NX Info App window automatically appears when you start NX.

Initially, this window only contains the *Status* tab which displays the version and licensing information for STAR-NX. More tabs appear as you create or load geometry files in STAR-NX, allowing you to perform more actions.



You cannot close the STAR-NX Info App window while NX or STAR-NX are running. Attempting to close the window by clicking the **X** button returns a message in the *Status* tab informing you of this. You can maximize and minimize the window by clicking the **-** and **_** buttons respectively.

Use the **Clear** button to clear all the information that is displayed in a particular tab.

Use the **Abort** button to stop active processes, such as surface/volume meshing, or the solver. This button only appears when you can stop a process — it is grayed out at all other times.

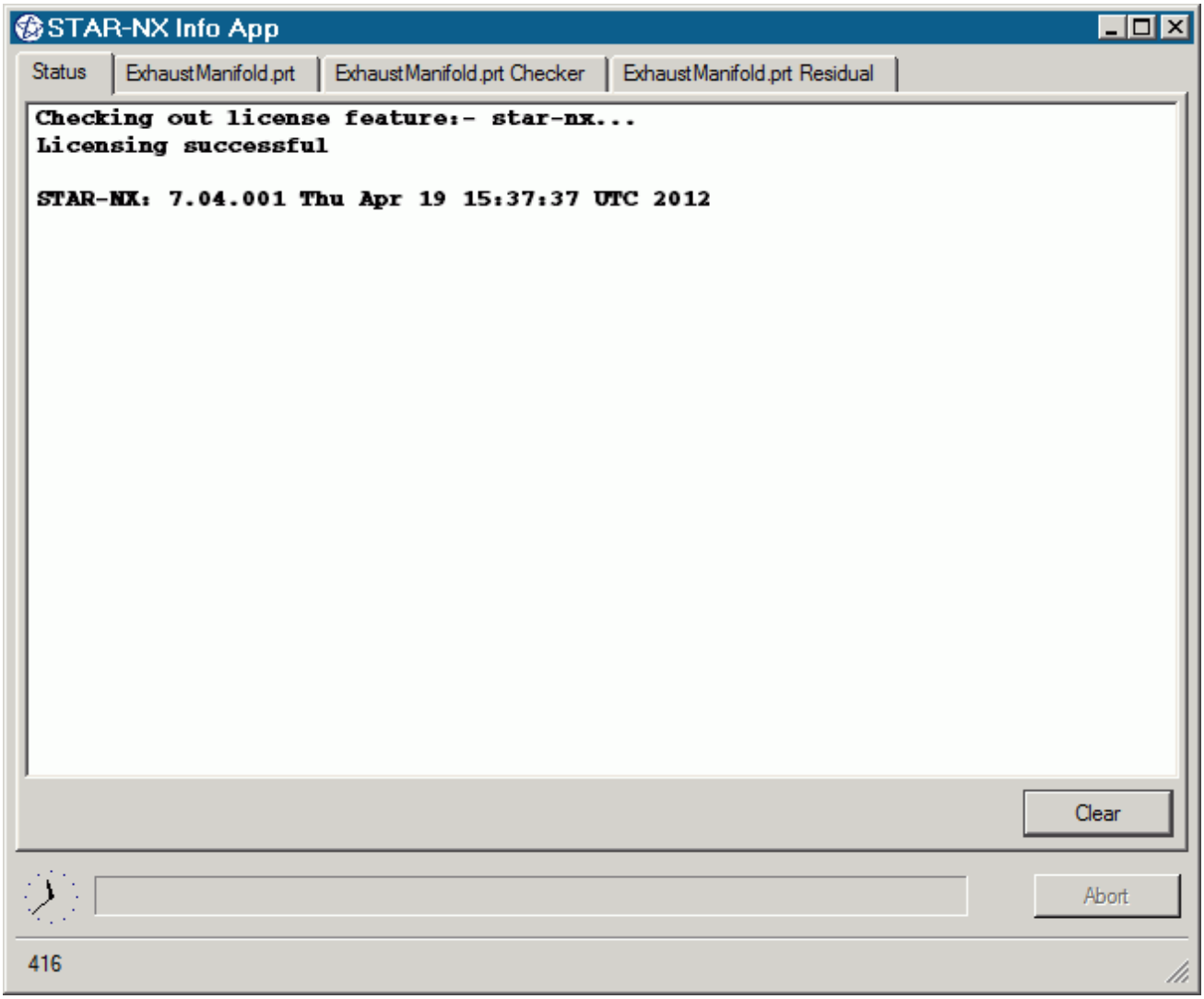
A clock is displayed in the bottom-left of the window. This clock displays the current time on your machine. A progress bar is located next to the clock and reports the progress of various processes, such as mesh generation and the solver.

More information on the various tabs is provided in the following sections:

- The Status Tab
- The Output Tab
- The Model Checker Tab
- The Residuals Tab

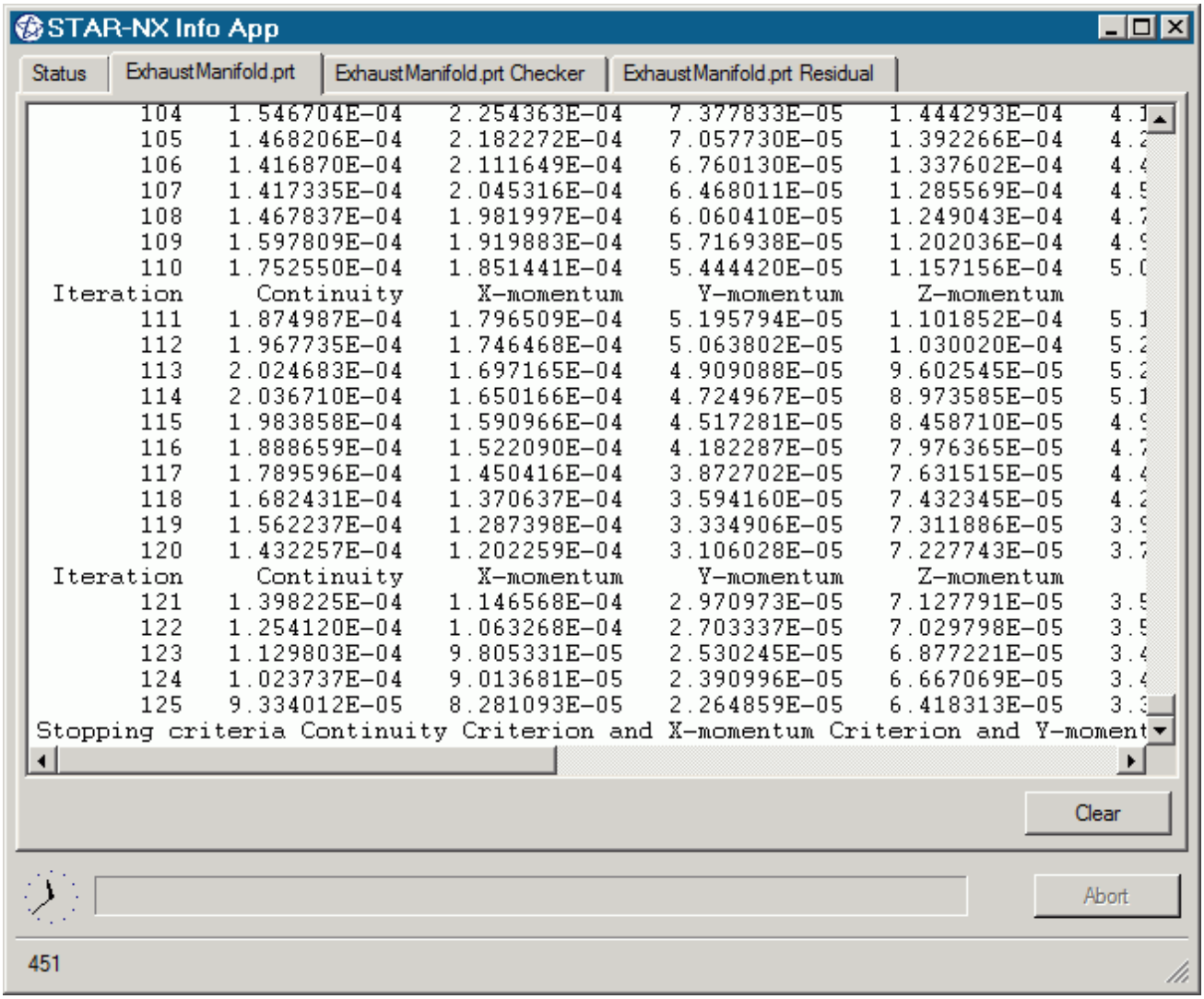
The Status Tab

The *Status* tab displays the licensing information for your current session, informing you whether the STAR-NX license checked out successfully. The version of STAR-NX is also displayed in this tab. This tab appears by default when you launch NX without a geometry file.



The Output Tab

The *Output* tab appears once you create or open a geometry file in STAR-NX.



During the meshing process, this tab displays status information along with a brief report on the generated mesh. If errors were encountered during the meshing process, they are reported here with suggestions as to what can be causing them.

When the solver is running, the *Output* tab behaves like the output window in STAR-CCM+, displaying the data for each residual per iteration. When the solver finishes running, the same information is also displayed in the STAR-CCM+ *Output* window.

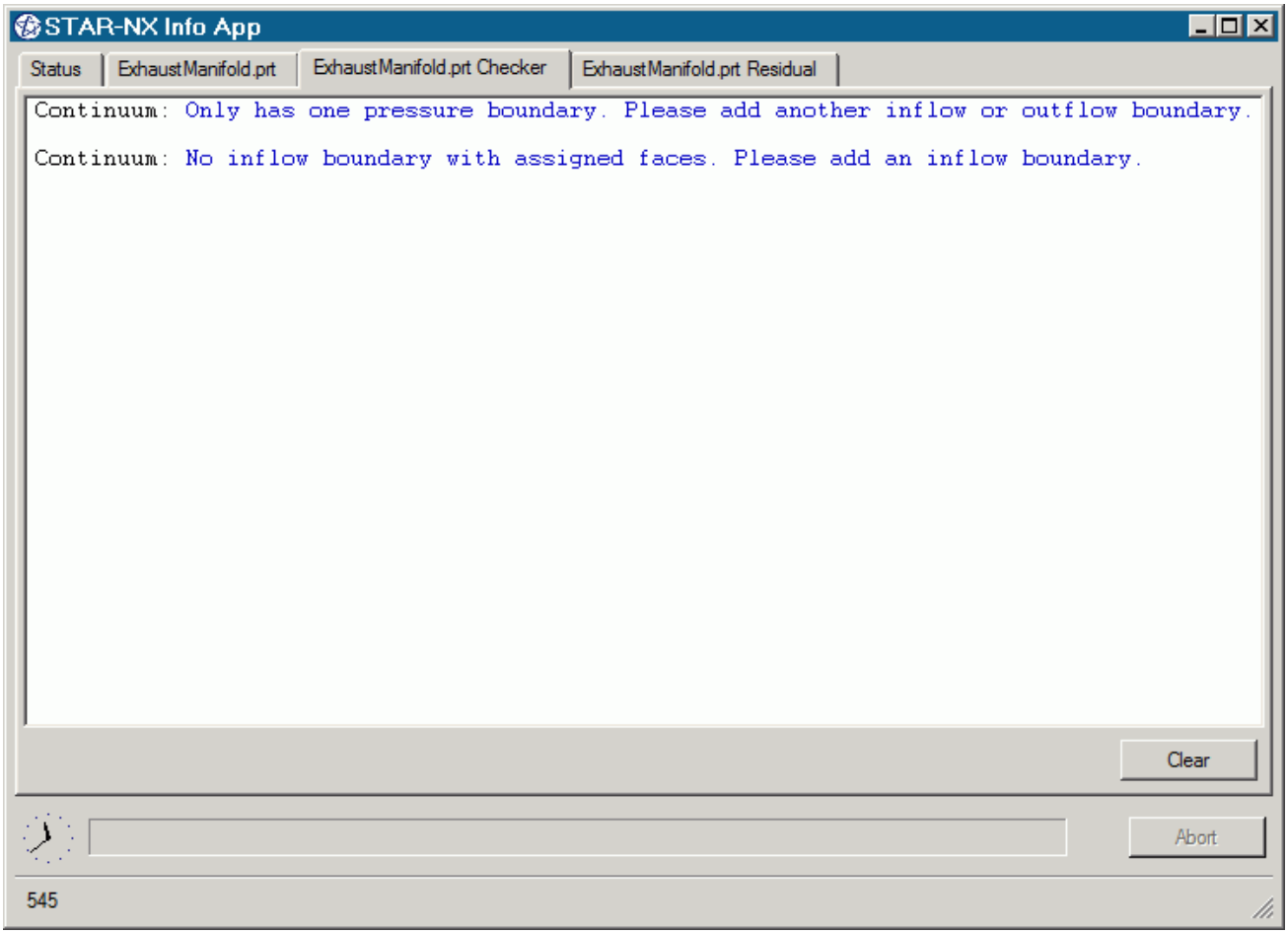
You can control the amount of information that is displayed in the *Output* tab by editing the verbosity in the `config.xml` file. See Modifying the Config.xml File.

To abort the mesher or solver, click the **Abort** button.

The Model Checker Tab

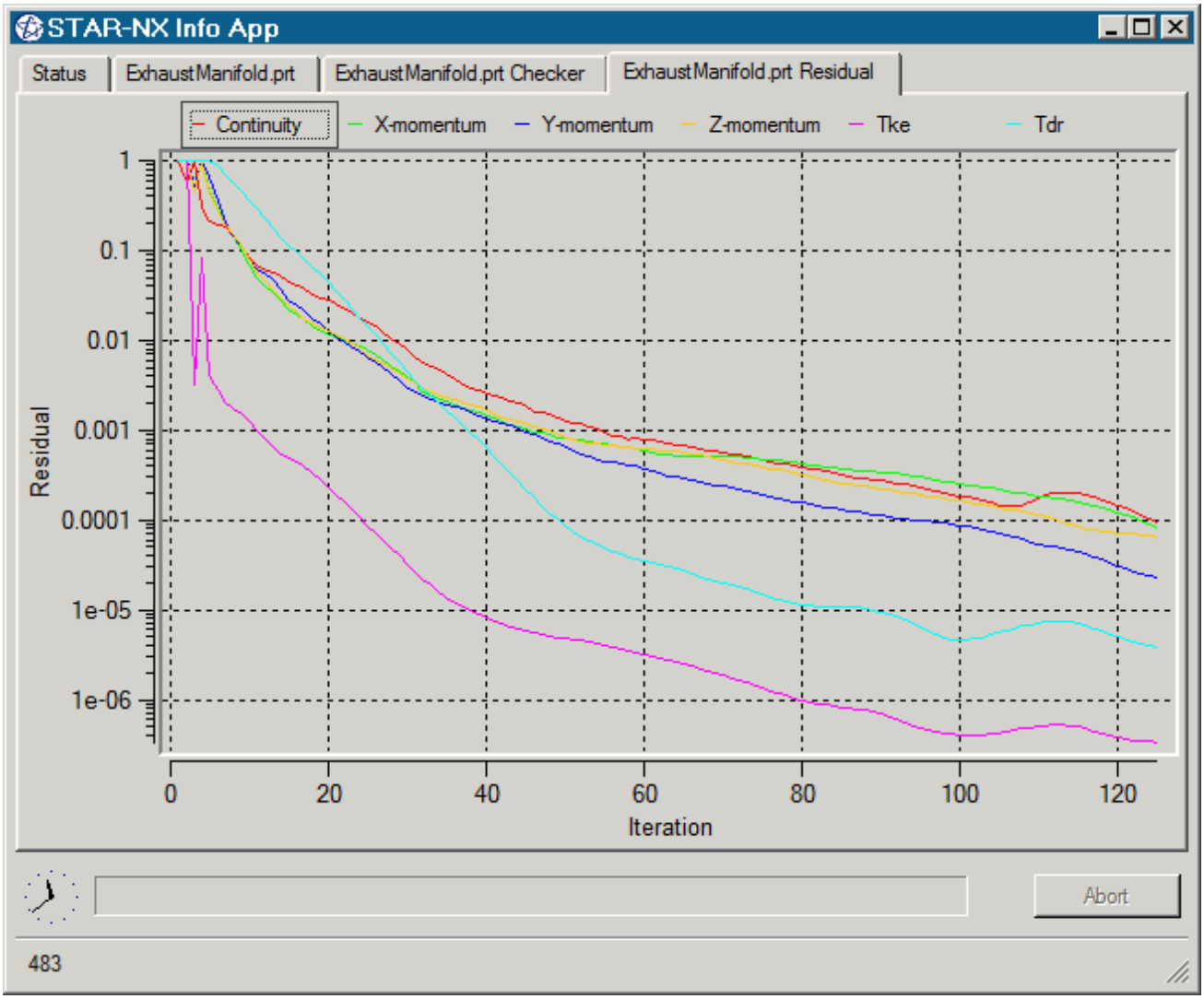
The *Model Checker* tab displays information from the live model checker, where the case setup is actively monitored for any problems. See Checking the Model for Errors.

The following example reports that the case has no inlet boundary defined.



The Residuals Tab

The *Residuals* tab appears when you launch the solver. Use the residuals to monitor the convergence of the solution being calculated.



Typically, a drop in three to five orders of magnitude in the residuals indicates that the solution has converged to an acceptable level. The residuals that are displayed in this tab also appear in STAR-CCM+ under the **Plots** node.

You can zoom in to the residuals by clicking and holding the left mouse button in the graph area and drawing a box around the area you want to view. To zoom out, right-click in the graph area. You can hide individual residuals from the *Residuals* tab by clicking their label at the top of the plot window.

Modifying the config.xml File

STAR-CCM+ requires a configuration file to enable it to find the appropriate CD-adapco components to run correctly. This configuration file is created during installation. An explanation of the file and settings within it is given here.

The configuration file is called `config.xml` and STAR-NX searches for it in the location given by the `STARCADSERIES` environment variable.

Typical `config.xml` content is shown below. In this example STAR-CCM+ version 12.06.### is shown.

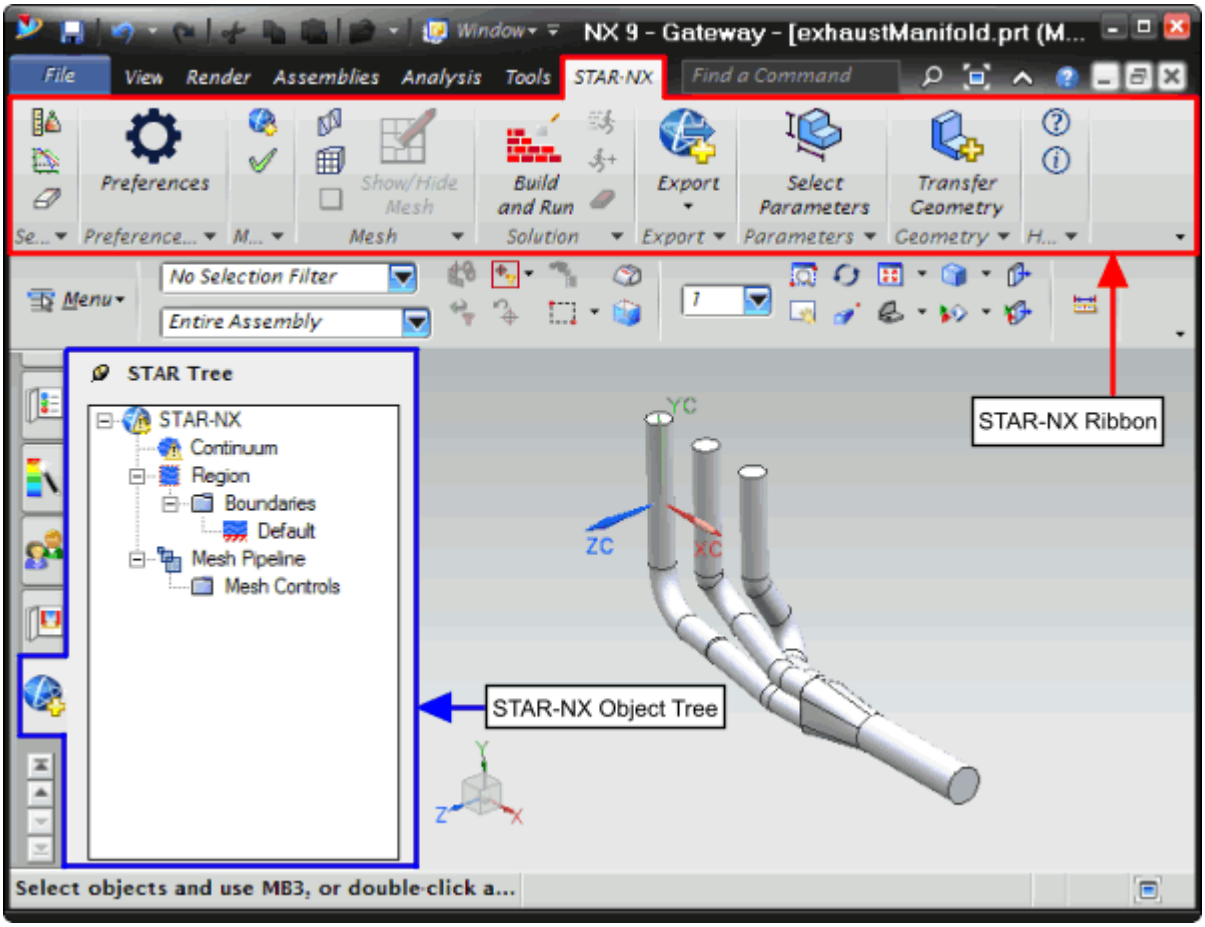
```
<CAD>
  <STAR_HOME>C:\Program Files\CD-adapco\12.06.###\STAR-CCM+12.06.###
\star</STAR_HOME>
  <SOLVER>C:\Program Files\CD-adapco\12.06.###\STAR-CCM+12.06.###\star
\bin\starccm+.exe</SOLVER>
  <LOGFILE>starcad.log</LOGFILE>
  <WWHELP>C:\Program Files\CD-adapco\12.06.###\STAR-CCM+12.06.###
\win64\Intel11.1\wwhelp.exe</WWHELP>
  <MANDIR>C:\Program Files\CD-adapco\12.06.###\STAR-CCM+12.06.###\doc
\en\online</MANDIR>
  <VERBOSITY>Normal</VERBOSITY> <!-- Can be Verbose, Log or Normal.
Default Normal -->
</CAD>
```

You can edit the following lines:

- `<SOLVER>` provides the location of the STAR-CCM+ solver executable. STAR-CCM+ licensing options are set in the *Preferences* dialog in the STAR-NX user interface. See Setting Up Parallel Processing and Licensing.
- `<LOGFILE>` is the log file that will automatically be created and saved when STAR-NX is run. A path of your choice can be used.
- `<VERBOSITY>` is the verbosity of the output window in the STAR-NX Info App. This can be set to one of the following:
  - `Normal`—This is the default setting and will display the least amount of information in the *Output* panel.
  - `Verbose`—This will display a moderate amount of information in the *Output* panel.
  - `Log`—This will display all available information in the *Output* panel.

Understanding the STAR-NX Interface

The STAR-NX user interface is embedded within the NX environment, with the addition of the STAR-NX object tree and ribbon.



The majority of the STAR-NX operations are executed within the object tree and ribbon. Certain actions can also be carried out using multiple methods. A summary of the features available in each part of the STAR-NX interface are detailed below.

Object Tree

The operations provided in the object tree include:

- Solution Controls
- Definition of Units System
- Continuum settings
- Region settings
- Add new boundary
- Boundary conditions
- Mesh Pipeline settings
- Add new mesh controls

Ribbon

- Settings
  - Definition of Units System

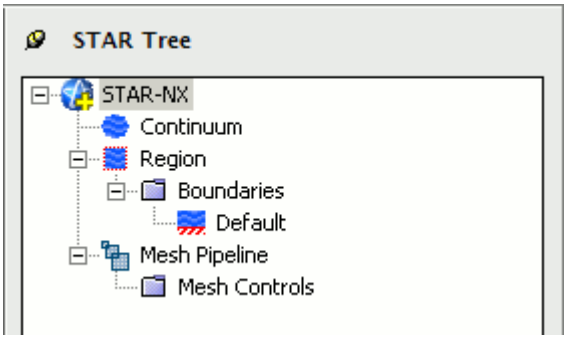
- Solution Controls
- Clear All

- Preferences
- Mesh
  - Generate Surface Mesh
  - Generate Volume Mesh
  - Clear Generated Meshes
  - Show/hide Mesh
- Solution
  - Build and Run
  - Run Solver
  - Run For More Iterations
  - Clear Solution
- Model
  - Check Model for Errors
  - Transfer CAE Model
- Export
  - Surface Mesh (.dbs)
  - Volume Mesh (.ccm)
  - Engineering Report (.html, .csv)
  - Simulation (.sim)
- Geometry
  - Transfer Geometry
  - Design Parameters
- STAR-Works Help
- About STAR-Works

The STAR-NX Object Tree

STAR-NX uses an object tree as the primary method for setting parameters.

The object tree consists of nodes, each one of which represents a particular component of the simulation. This approach to the user interface complements the method that is used by STAR-CCM+, allowing for an easy transition between the two software packages.



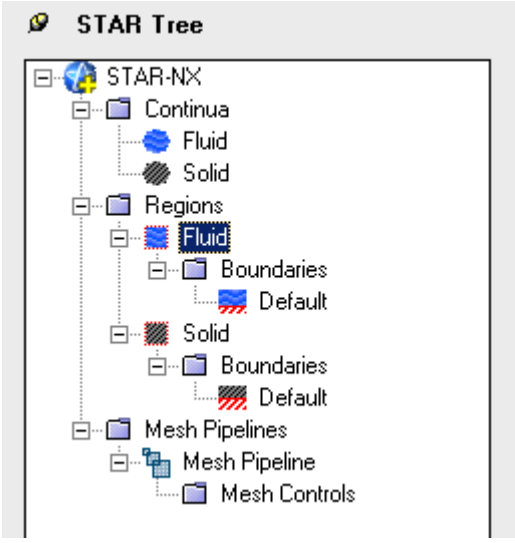
The object tree appears slightly different in assembly mode as it must accommodate multiple parts, but the principles of operation remain the same.

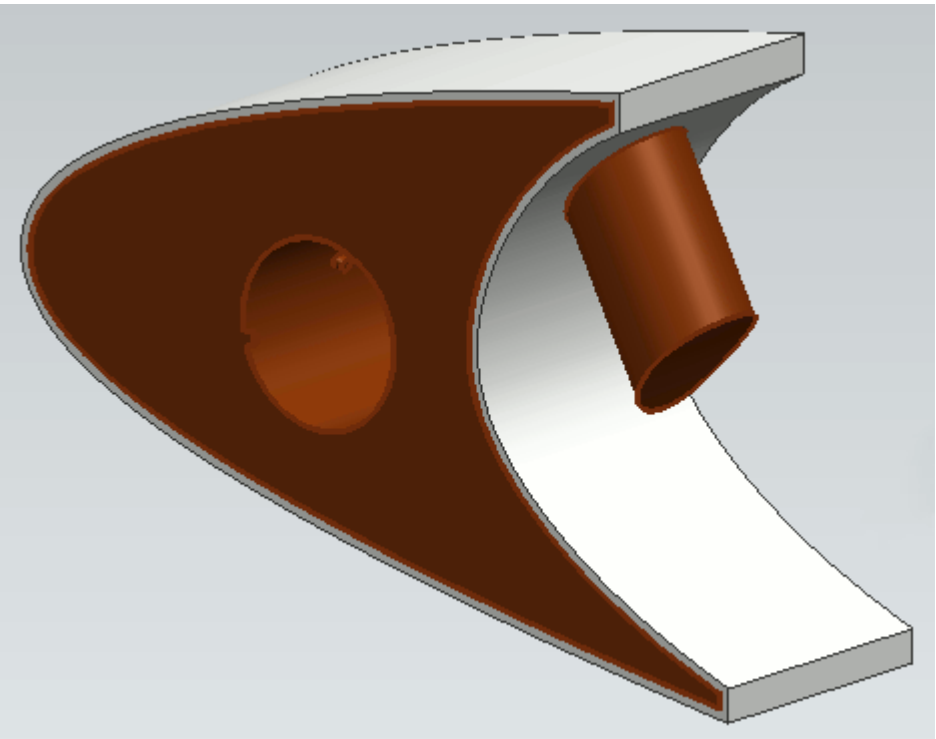
Nodes are arranged in an order that follows the recommended workflow for setting up a simulation in STAR-NX. At each stage, you typically right-click on a particular node in the tree and choose the appropriate action from the pop-up menu.

Certain functionality within STAR-NX applies to the simulation as a whole, rather than to individual objects. These operations are provided in the ribbon.

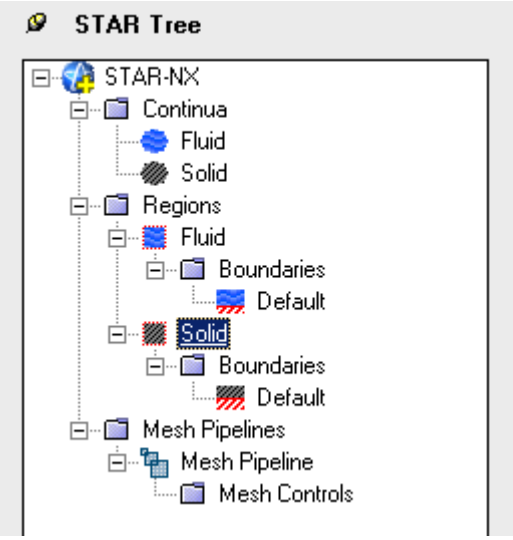
A complete list of all functions and their respective location is included in Understanding the STAR-NX Interface.

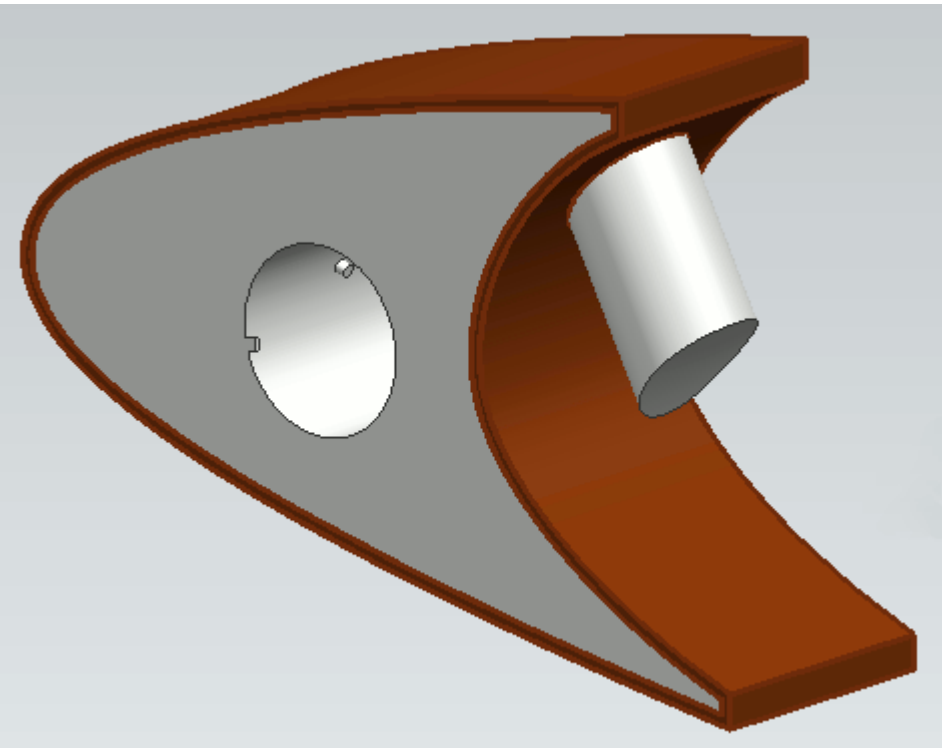
Clicking on the **Region** node within the object tree will highlight all the relevant faces on the geometry model for that particular region. If different parts are assigned to multiple regions, then clicking on a region will only highlight the part and faces associated with that specific region. For example, clicking on the **Fluid** node below will highlight all the faces associated with the Fluid region.





Similarly, clicking on the **Solid** node highlights all the faces associated with the Solid region.





Clicking on a boundary node in the STAR-NX object tree highlights all the faces on the CAD geometry model that are associated with it. Note that clicking on the *Default* boundary node will not highlight any faces on the CAD geometry model. To stop highlighting the faces, you can either click on a blank space in the geometry window, or click on the root node in the STAR-NX object tree.

Setting Up Parallel Processing and Licensing

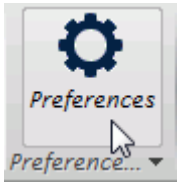
The licensing options and the options for setting up parallel processing are located in the *Preferences* dialog.

This dialog provides two sets of options: server preferences and server settings.

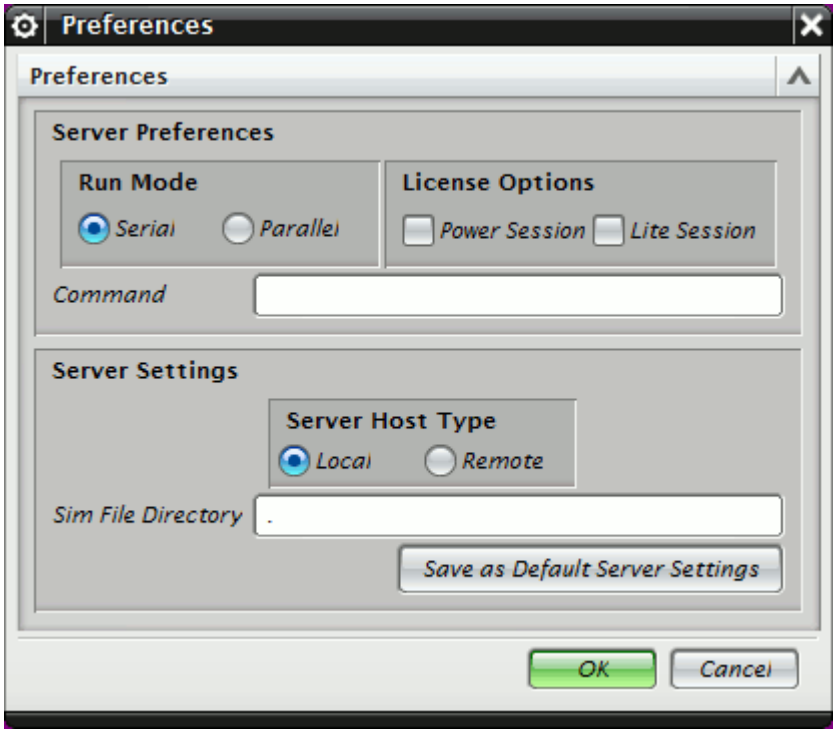
- Server preferences are applied whenever STAR-NX starts a server.
- Server settings are saved in the CAD file and are applied only to that CAD file. Click **Save as Default Server Settings** to save the current server settings as the default settings to be applied to all CAD files.

To modify the run mode and licensing options:

1. Select **Preferences** from the ribbon.



The *Preferences* dialog appears.



2. Select the run mode:
   - For serial, set *Run Mode* to **Serial** (the default setting).
   - For parallel, set *Run Mode* to **Parallel** and *Compute Processes* to the desired number of cores.

     You can use the **up/down** buttons or you can type in the number of cores that you want to use.
   - For advanced licensing options, such as Power-on-Demand, enter the appropriate argument in the *Command* field.

     For Power-on-Demand, enter:

     ```
     -licpath 1999@flex.cd-adapco.com -podkey <pod_key>
     ```

     Where:
     - `1999@flex.cd-adapco.com` is the default CD-adapco Power-on-Demand server.
     - `<pod_key>` is the Power-on-Demand license key.

     For more information on Power-on-Demand licensing, see the section **User Guide** > **Using STAR-CCM+** > **STAR-CCM+ Licensing** > **Power-on-Demand Licensing**.

     A full list of command-line arguments is available in the STAR-CCM+ User Guide at **User Guide** > **Using STAR-CCM+** > **Command-Line Reference**.

   The *Command* field updates automatically to reflect the selections you make. Manual entries are not affected by these selections.
3. Select a licensing option:
   - To use a Power Session license, select **Power Session**.

     This license is design for parallel simulations, where you can use an unlimited number of cores in this session.

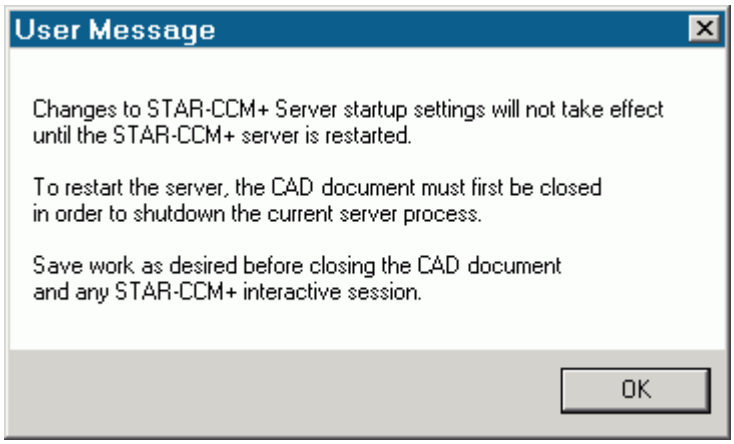
- To use a STAR-CCM+ Lite license, select **Lite Session**.

  A limited set of STAR-CCM+ features are provided when you activate this option.

If you select no licensing option, the default CAD client license is used.

4. Click **OK**.

   If the STAR-CCM+ server is running when you modify the preferences, a message box appears informing you that both STAR-Works and STAR-CCM+ must be closed for the changes to take effect.



5. Click **OK** to proceed.
6. Close the active STAR-NX and STAR-CCM+ sessions.
7. Restart STAR-NX.

Note: The STAR-CCM+ server only starts when it is required, such as during surface meshing or volume meshing. It does not start when you launch STAR-NX.

If an option in the *Preferences* dialog conflicts with an option in the `config.xml` file, the option that is set in the *Preferences* dialog takes precedence. An example of such a situation is if the *Compute Processes* property is set to `4` in the *Preferences* panel, while it is set to `2` in the `config.xml` file; the simulation runs using four processors. See Modifying the config.xml File

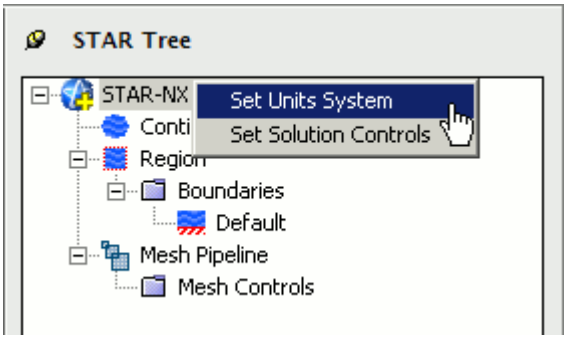
Note: IBM Platform MPI must be installed on your local machine for you to run in parallel. This program is included with the STAR-CCM+ package installer.

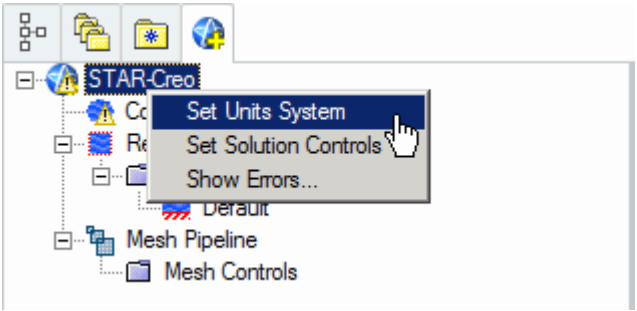
Changing the Units System

You can use either the SI (International Standard) or USCS (United States Customary System) units system in STAR-NX.

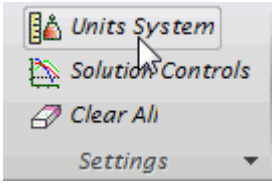
To change the units system:

1. Bring up the *Units System* dialog:
   - Right-click the STAR-NX node within the object tree and select **Set Units System**.



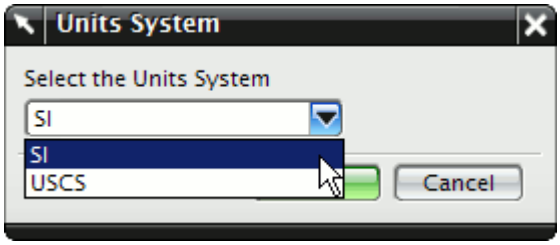


- Alternatively, from the ribbon, select **Units System**.



The *Units System* dialog appears.

2. Select the desired units system from the drop-down menu.



3. Click **OK**.

Automatic Unit Conversion in Dialog Panels

Changing the default units system converts existing values to the new units system. Ensure that you select the desired units system before you enter the value, otherwise you will need to re-enter it.

Note: Specific values that you have already modified the units on are not affected by a change of units system.

Transferring Units to STAR-CCM+

Units that are used in the CAD geometry file are also used in the STAR-CCM+ simulation. Less common units, such as those listed below, are not supported in STAR-CCM+:

- Micrometers
- Angstroms
- Nanometers
- Mils
- Microinches
- Footinches (this unit may be converted to inches in STAR-CCM+)
- Miles

Note: This list may not be exhaustive and is subject to change.

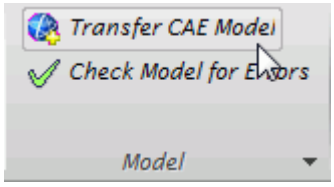
These units are converted to meters when the CAD geometry is imported into STAR-CCM+.

Transferring the CAE Model to STAR-CCM+

You can transfer your geometry and the simulation setup to STAR-CCM+ at any time from within the STAR-NX environment. If you want to transfer only the geometry to STAR-CCM+, refer to Transferring Geometry to STAR-CCM+.

To transfer the geometry with the simulation set up:

1. From the STAR-NX ribbon, click **Transfer CAE Model**.



The version of STAR-CCM+ that is affiliated with STAR-NX launches.

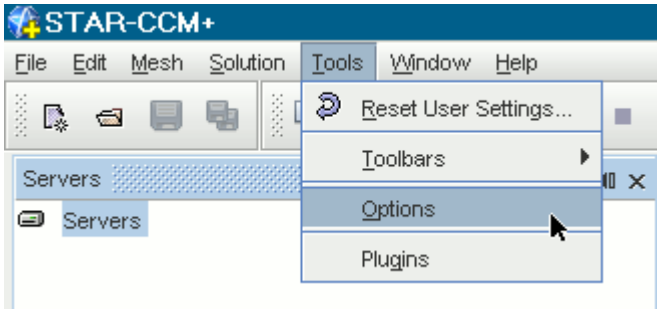
Note: A surface mesh is required in order to visualize the geometry in STAR-CCM+. If you have not generated a mesh before clicking **Transfer CAE Model**, the geometry is tessellated with a medium density.

Using NX Mouse Controls in STAR-CCM+

STAR-CCM+ offers the ability to set the mouse interactor style to match that of NX; this could make geometry manipulation in STAR-CCM+ more natural if you are used to working with NX.

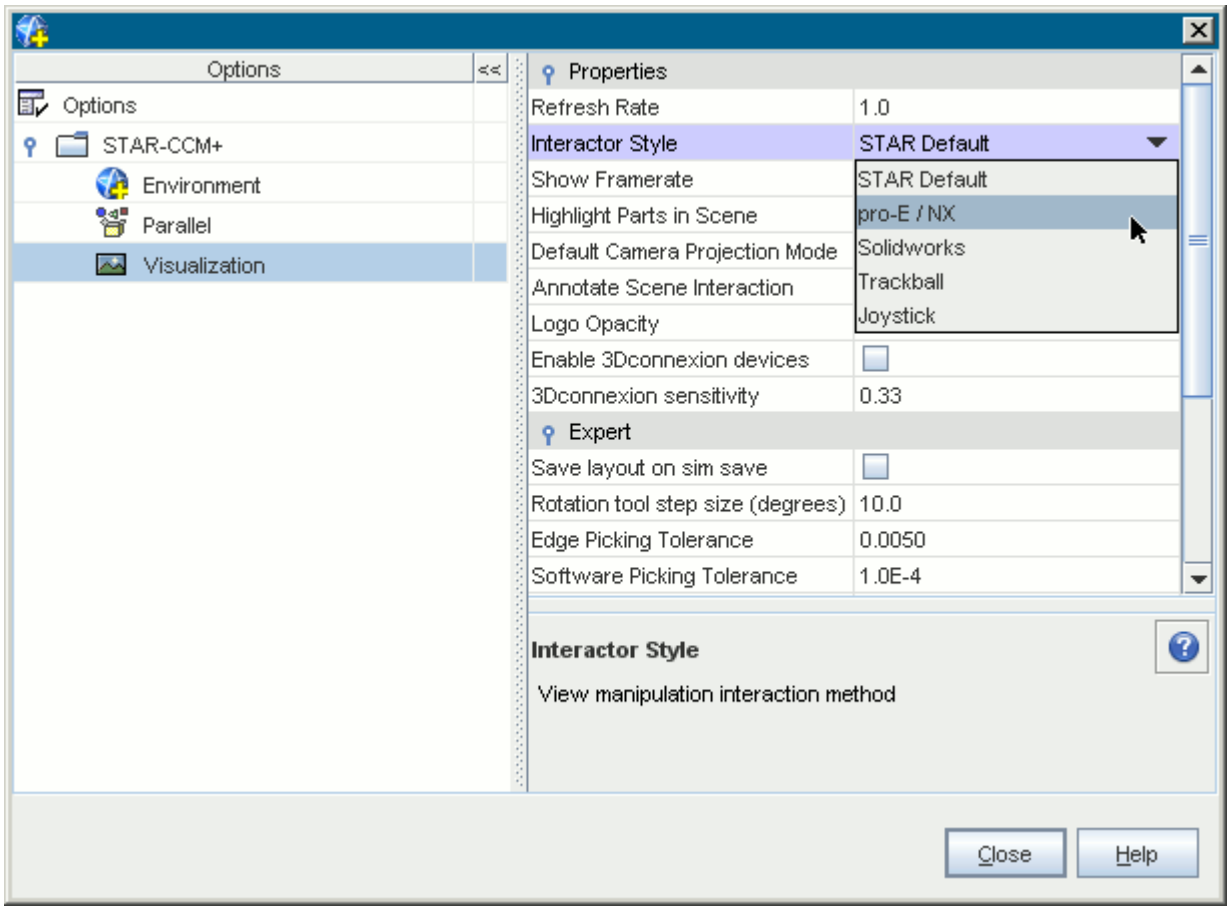
To change the mouse interactor style:

1. Launch STAR-CCM+.
2. Click **Tools** and select **Options** from the menu.



The *Options* dialog appears.

3. Expand the STAR-CCM+ node and select **Visualization**.
4. In the *Properties* panel, set *Interactor Style* to**pro-E / NX** using the drop-down menu.



5. Click **Close**.

STAR-CCM+ will now use this mouse interactor style each time it launches.

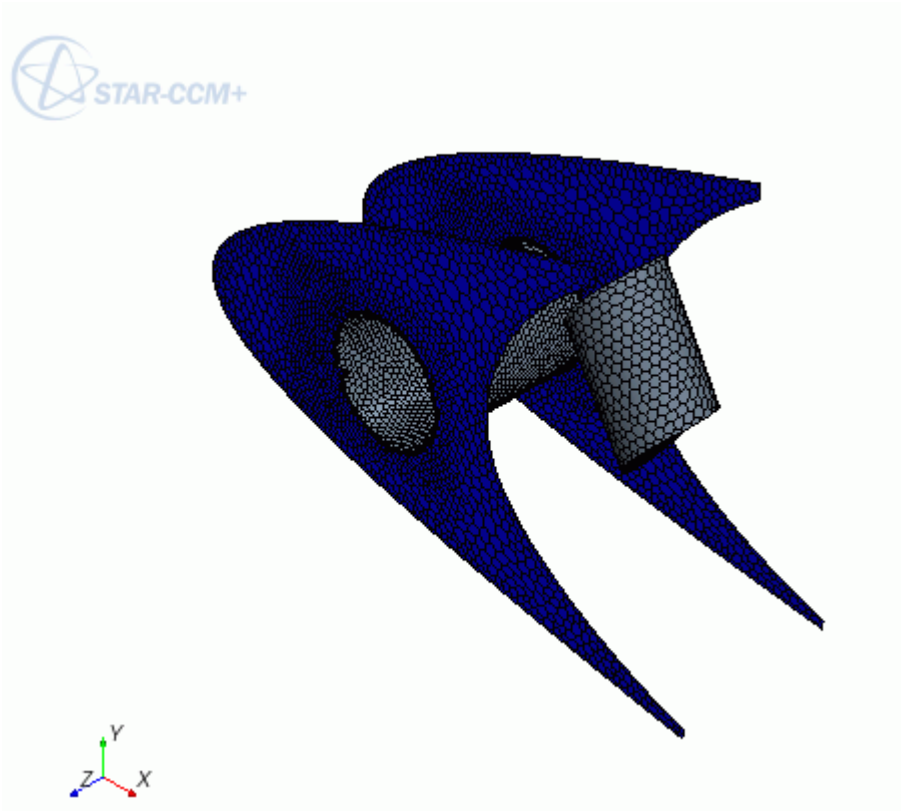
Editing STAR-NX Settings While the STAR-CCM+ Client is Running

If you make a change in STAR-NX while the STAR-CCM+ client is running, such as adding or deleting a boundary, you may need to update the volume mesh.

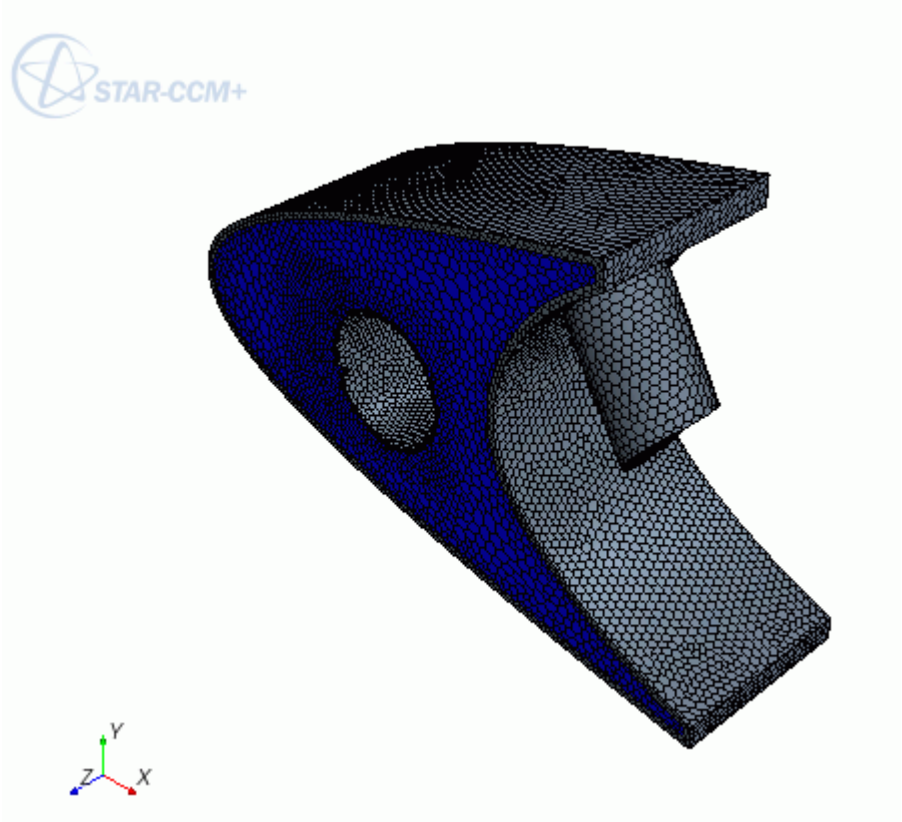
Scenes in the STAR-CCM+ session will not display the model correctly unless you keep the volume mesh up to date. A list of changes that require you to re-mesh the geometry is provided below:

- For boundaries and regions: changing the type, adding or deleting
- For mesh pipelines and mesh controls: editing, adding or deleting

When adding or removing regions and boundaries in STAR-NX, the scenes in STAR-CCM+ may be missing some parts, as shown below, even after re-meshing the geometry.



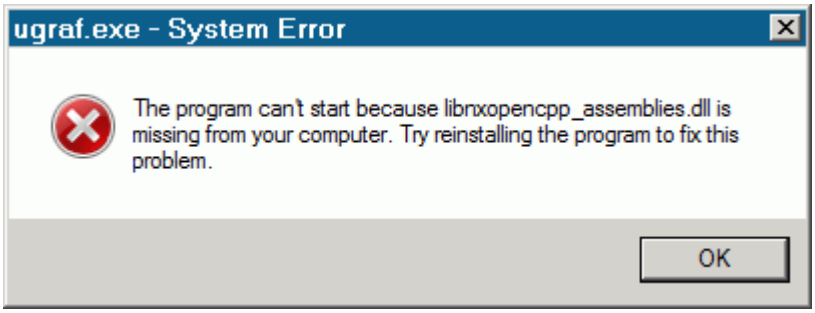
To update the scene, you need to manually add the modified regions and boundaries to the scene in the STAR-CCM+ client. You can do this by dragging and dropping the region or part from the STAR-CCM+ object tree, or by adding the region or part using the Part Selector. An alternative method would be to create a new scene, as this will automatically pick up the new regions and boundaries added to the simulation.



Using STAR-NX with Multiple Installed Versions of NX

If you have multiple versions of NX installed on your machine, STAR-NX is installed to work with the latest version.

Attempting to run a different version of NX results in an error message with the `ugraf.exe` executable.



Note: The error message can differ depending on what version of NX you are trying to run.

This error indicates that STAR-NX is not set to work with the version of NX that you are trying to run. Clicking **OK** dismisses the message and allows you to work in that version of NX without STAR-NX.

You can choose which version of NX STAR-NX works with from the `custom_dirs.dat` file.

You can find the `custom_dirs.dat` file by checking the `UGII_CUSTOM_DIRECTORY_FILE` environment variable.

To specify which version of NX to use with STAR-NX:

1. Navigate to the folder that contains the `custom_dirs.dat` file.

   The default location is:

   ```
   C:\Program Files\CD-adapco\12.06.###\STAR-CAD12.06.###\STAR-NX\
   ```

2. Locate and edit the `custom_dirs.dat` file using a text editor of your choice.

   Example `custom_dirs.dat` content is shown below. Depending on what versions of NX are installed on your machine, the contents of this file can differ:

   ```
   // If uncommented, the following line supports STAR-NX with NX 9.
   C:\Program Files\CD-adapco\12.06.###\STAR-CAD12.06.###\STAR-NX
   \win64\bin\STAR-NX9

   // If uncommented, the following line supports STAR-NX with NX 8.
   // C:\Program Files\CD-adapco\12.06.###\STAR-CAD12.06.###\STAR-NX
   \win64\bin\STAR-NX8
   ```

Uncomment the line that corresponds to the path of the version of NX that you want to use. In this example, NX8.

3. Comment out the lines that correspond to the path of any other versions of NX.

The final `custom_dirs.dat` content appears as follows:

```
// If uncommented, the following line supports STAR-NX with NX 9.
// C:\Program Files\CD-adapco\12.06.###\STAR-CAD12.06.###\STAR-NX
\win64\bin\STAR-NX9

// If uncommented, the following line supports STAR-NX with NX 8.
C:\Program Files\CD-adapco\12.06.###\STAR-CAD12.06.###\STAR-NX
\win64\bin\STAR-NX8
```

4. Save the file and exit the text editor.

Disabling STAR-NX

If for any reason you would like to temporarily disable STAR-NX without having to uninstall it, do the following:

1. Change the *variable name* of the `UGII_CUSTOM_DIRECTORY_FILE` environmental variable. It is recommended that the *variable value* is not changed.

or

2. Navigate to the STAR-Works installation folder and rename the `custom_dirs.dat` file.

   The typical path is `C:\CD-adapco\12.06.###\STAR-CAD12.06.###\STAR-NX`

NX will now launch without STAR-NX. To enable STAR-NX again, undo the changes listed above.

Common Functions

The following section details common operations used within the STAR-NX environment.

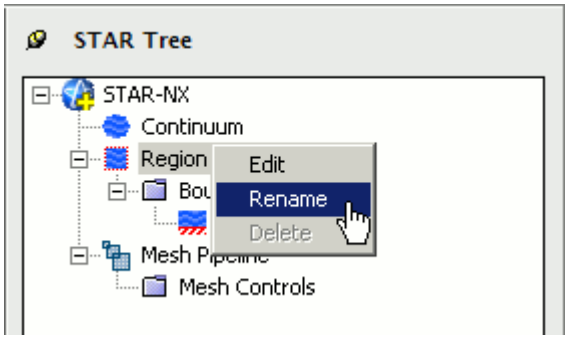
Contents:

Renaming an Object
Deleting an Object
Editing An Object
Assigning Surfaces and Parts to Objects in STAR-NX
Removing Surfaces and Parts from Objects in STAR-NX

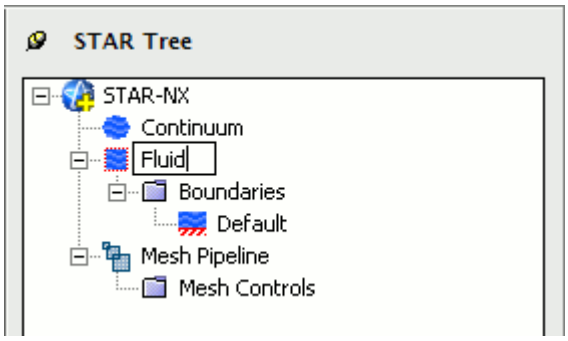
Renaming an Object

To rename an object:

1. Right-click the object node within the object tree and select **Rename**.



2. Enter the name.



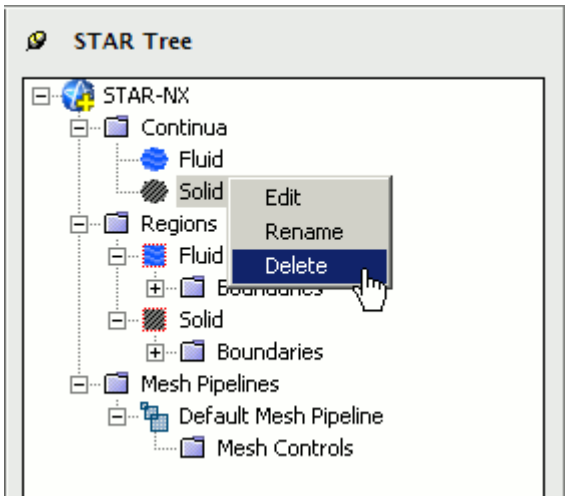
3. Click on a blank space in the object tree.

Note: You can also assign entity names at the point where you create them, for example, when adding boundaries, continua, regions, mesh pipelines and mesh controls. When adding a new object, locate the **Name** text box in the settings dialog and enter the required name.

Deleting an Object

To delete an object:

1. Right-click the object node within the object tree and select **Delete**.



To delete multiple entities:

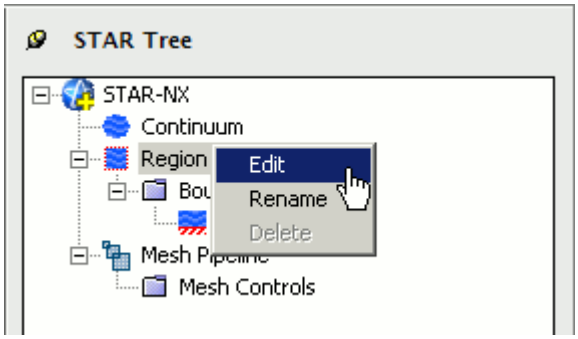
2. Select the desired nodes within the object tree (by holding down the `<Ctrl>` key), and right-click on any one of the selected nodes, then select **Delete** from the pop-up menu.

Note: Some combinations of selections cannot be deleted together using this method. In these cases, no pop-up menu appears.

Editing An Object

To edit an object:

1. Right-click on the desired node in the object tree and select **Edit...**.



The settings dialog for the object appears.

2. Make the changes that you want.
3. Click **OK**.

Assigning Surfaces and Parts to Objects in STAR-NX

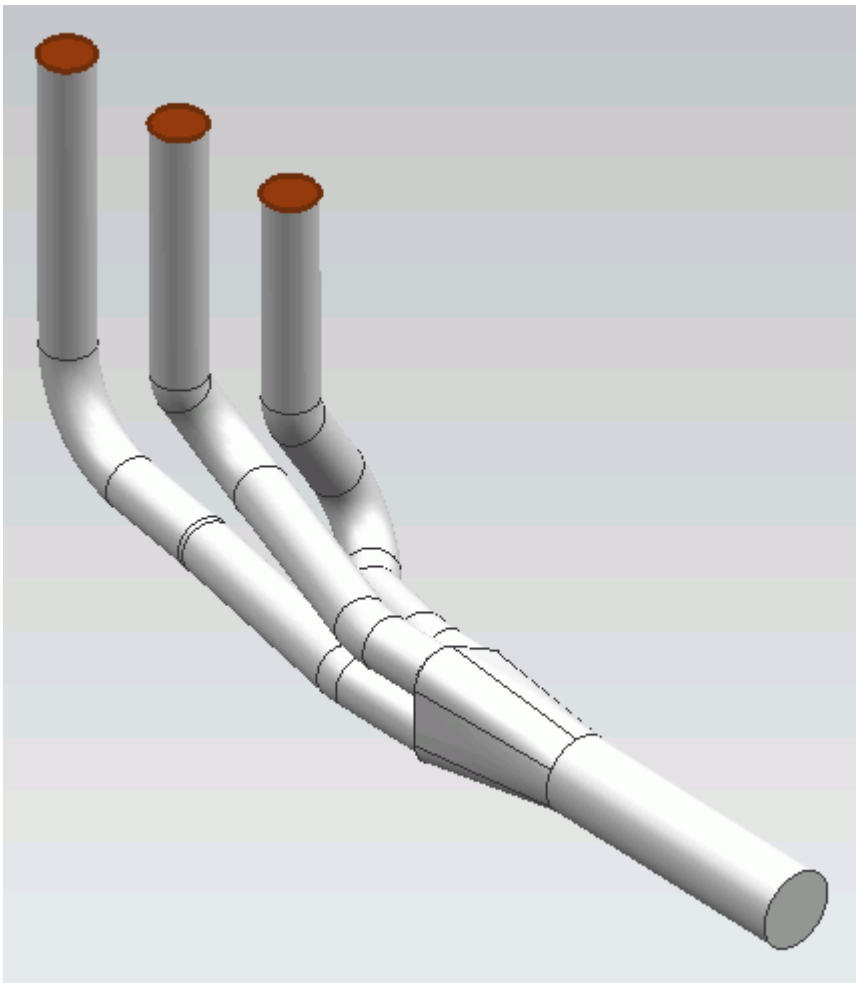
Regions, boundaries, mesh pipelines, and mesh controls are all represented as objects on the STAR-NX object tree. You can assign parts and surfaces to these objects when their property dialog is open.

Assigning Surfaces to Objects

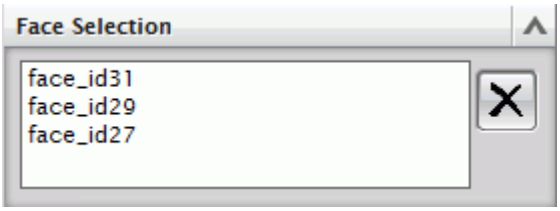
To assign a surface to an object (when its property dialog is open):

1. On the CAD geometry, click the surfaces that you want to assign to the object. You can add multiple surfaces by clicking more surfaces on the CAD geometry.

   The selected surfaces are highlighted on the CAD geometry.



The selected surfaces are also displayed in the *Face Selection* box of the active dialog.



Parts that are assigned to other regions are automatically hidden on the CAD geometry.

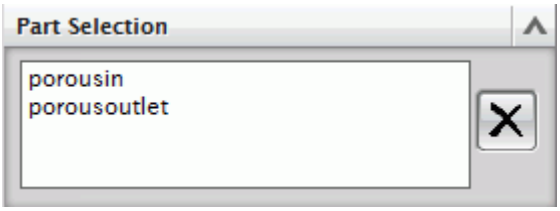
Assigning Parts to Objects

To assign a part to an object (when its property dialog is open):

2. On the CAD geometry, click the part that you want to assign to the object. You can add multiple parts by clicking more parts on the CAD geometry.

   The selected parts are highlighted on the CAD geometry.

The selected parts are displayed in the *Part Selection* box of the active dialog.



Parts that are assigned to other regions are automatically hidden on the CAD geometry.

Removing Surfaces and Parts from Objects in STAR-NX

Regions, boundaries, mesh pipelines, and mesh controls are all represented as objects on the STAR-NX object tree. You can remove parts and surfaces from these objects when their property dialog is open.

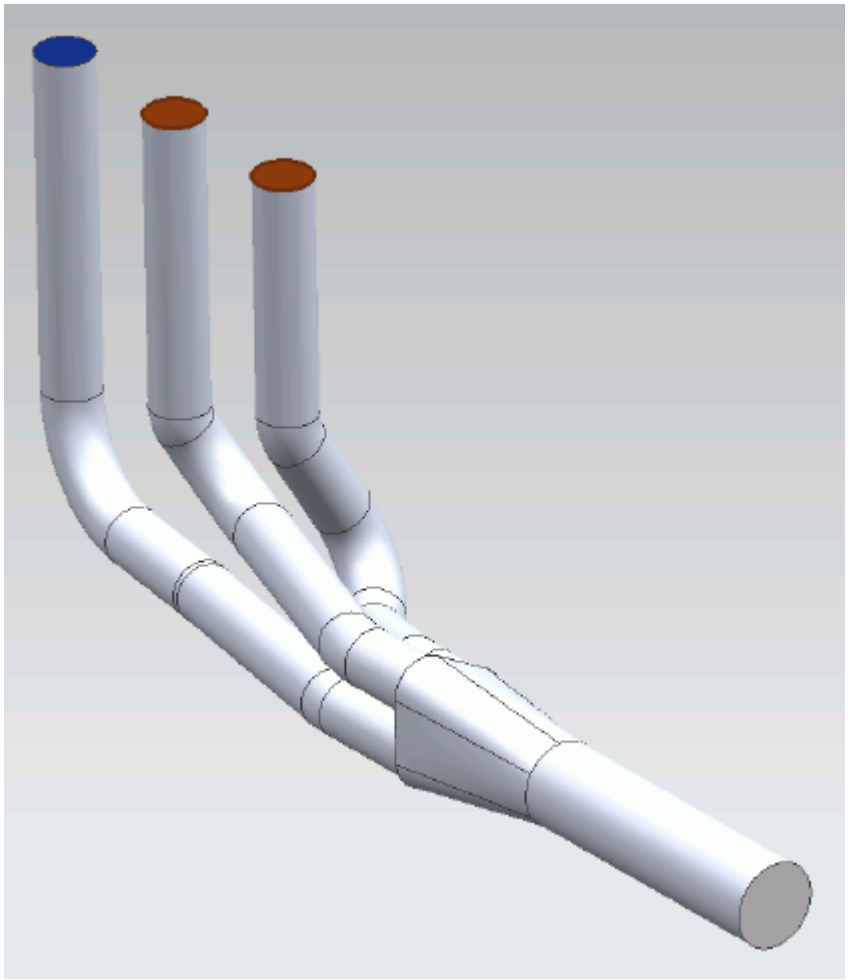
A surface that is removed from a boundary is returned to the default boundary. Similarly, removing a part from a region returns it to the default region.

Removing Surfaces from Objects

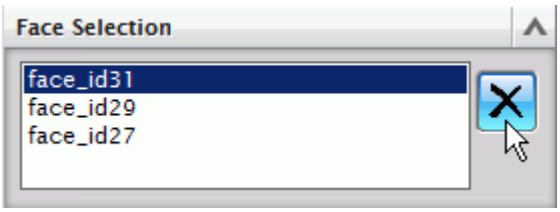
To remove a surface from an object (when its property dialog is open):

1. In the *Face Selection* box, select the surface you want to delete.

   The selected surface on the CAD geometry is highlighted in blue.



2. Click (**Remove Selected Face**).



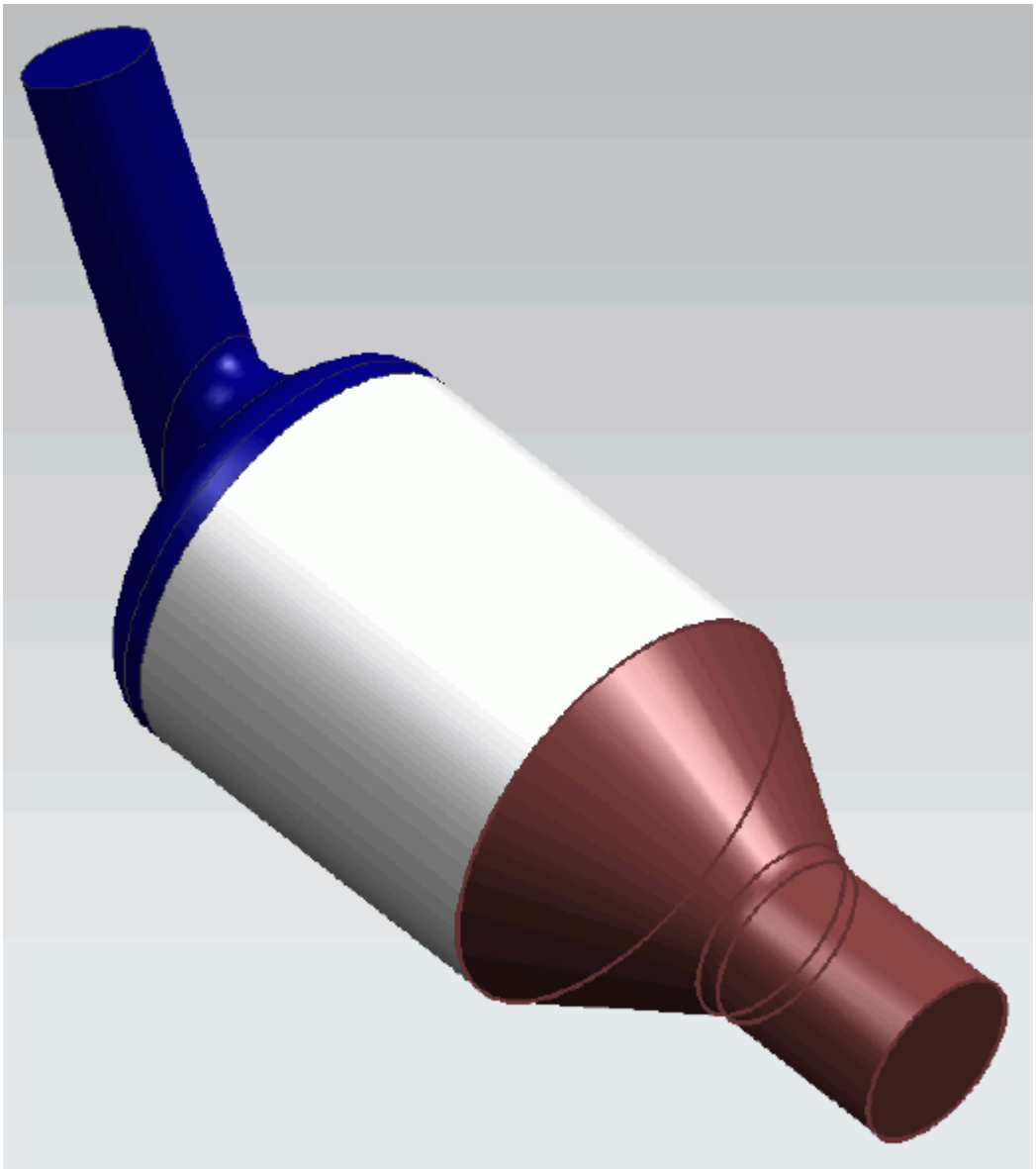
The surface is removed from the *Face Selection* box. You can select multiple surfaces for deletion by holding down the <`Ctrl`> key, and remove them in one operation.

Removing Parts from Objects

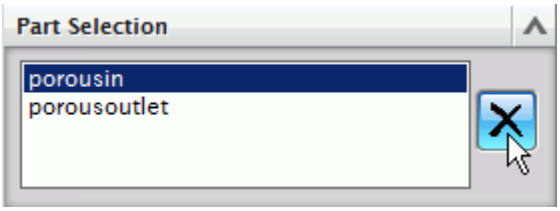
To remove a part from an object (when its property dialog is open):

3. In the *Part Selection* box, select the part you want to remove.

The selected part on the CAD geometry is highlighted in dark blue.



4. Click (**Remove Selected Part**).



The part is removed from the *Part Selection* box. You can select multiple parts for deletion by holding down the <`Ctrl`> key, and remove them in one operation.

Running STAR-NX and STAR-CCM+ on Different Machines

You can run STAR-NX and STAR-CCM+ on two different machines.

STAR-CCM+ operates on a client-server architecture, where STAR-CCM+ is the server and STAR-NX is the client. A benefit of client-server architectures is that the client and the server do not have to be running on the same machine. Typically, CAD packages run on Windows operating systems. You can choose to run STAR-CCM+ on any remote machine with a supported or certified platform, including Linux platforms.

Contents:

Setting Up STAR-CCM+ to Run on a Remote Machine with STAR-NX on the Local Machine

Setting Up STAR-CCM+ to Run on a Remote Machine with STAR-NX on the Local Machine

STAR-NX and STAR-CCM+ can run on two different machines.

To set up STAR-CCM+ to run on a remote machine:

1. From the STAR-NX ribbon, click **Preferences**.

2. In the *Preferences* dialog, set *Server Host Type* to **Remote**.
   The options for the remote server appear.
3. Set the remaining server settings to reflect your situation. See Server Settings Panel Reference.
4. If you want to store these settings for the next time you launch STAR-NX, click **Save as Default Settings**.
5. Click **OK**.

Contents:

Server Settings Panel Reference

Server Settings Panel Reference

The Server Settings panel that is located in the Preferences dialog allow you to run STAR-NX on a remote machine.

Property	Description	
Server Host Type	Specifies whether to run STAR-CCM+ on the local machine or on a remote machine.	
	Local	STAR-CCM+ runs on the local machine.

Property	Description	
	Remote	STAR-CCM+ runs on a remote machine
SSH Command	Specifies the absolute path to the SSH installation directory.	
Server Host	Specifies the name of the remote host where STAR-CCM+ is installed. This option can contain the IP address, name, or full name of the host.	
Sim File Directory	Specifies the absolute path to the directory where you want to store the simulation files on the remote machine. When in **Local** mode, the simulation files are saved in the same directory as the geometry file. You can specify a custom directory.	
Server Executable	Specifies the absolute path to the STAR-CCM+ executable on the remote machine.	

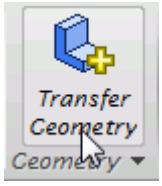
Transferring Geometry to STAR-CCM+

With STAR-NX, you can transfer your CAD geometry directly to STAR-CCM+ from within the NX environment.

The Transfer Geometry feature imports your geometry into STAR-CCM+. This function provides an alternative method to the CAD translator add-ons for importing geometry. This method of transferring geometry can provide a faster workflow when performing a design analysis.

To transfer geometry to STAR-CCM+:

1. From the STAR-NX ribbon, click **Transfer Geometry**.



   STAR-CCM+ launches and imports the CAD geometry. You are now at the same stage in the workflow as if you imported the surface mesh directly into STAR-CCM+ using the CAD translators (**File** > **Import** > **Import Surface Mesh...**).

2. Set up your case in STAR-CCM+.

Caution: Do not click **Transfer CAE Model** if you have previously used the Transfer Geometry method and set up your case in STAR-CCM+. This action results in the loss of your settings in STAR-CCM+.

Contents:

Saving Simulation Files
Transferring Design Data to STAR-CCM+
Modifying Geometry from Within STAR-CCM+
Controlling How the Geometry is Transferred to STAR-CCM+
Geometry Transfer Settings Dialog

Saving Simulation Files

After transferring the geometry to STAR-CCM+, you can save the simulation file and retain the associativity with STAR-NX.

To save your simulation files and keep the associativity:

- Save your CAD geometry in STAR-NX.

  Note: You can also safely rename and move the file using this method.

or

- Save your simulation file in STAR-CCM+.

Note: Do not rename the simulation file when using this method, as it will lose the associativity with the NX file.

Resuming a Simulation

To resume a simulation after closing STAR-NX and STAR-CCM+:

1. Launch STAR-NX.
2. Open the CAD geometry in STAR-NX.
3. Click **Transfer Geometry**.

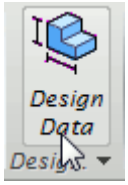
Transferring Design Data to STAR-CCM+

Design data consists of design parameters, coordinate systems, and reference planes that you can transfer to STAR-CCM+ along with the geometry.

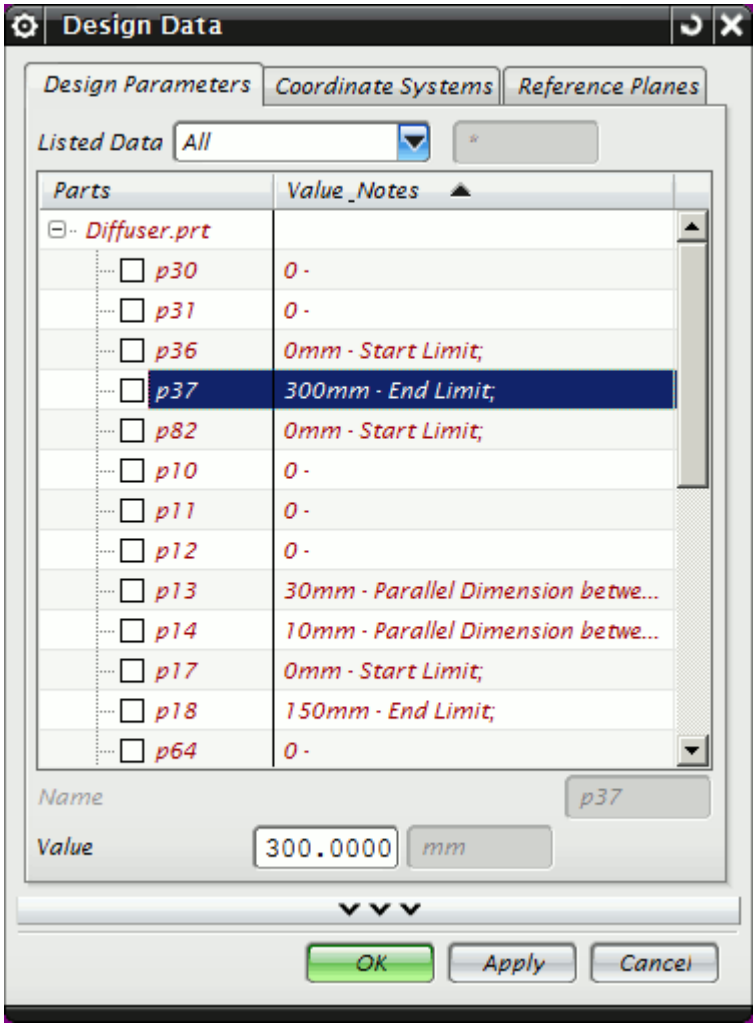
Design parameters control the feature dimensions, for example extrusion depth, and are useful for modifying the geometry in STAR-CCM+ after running the analysis. You can use reference planes in scalar scenes and reports in STAR-CCM+. You can use coordinate systems to define motion in the STAR-CCM+ simulation.

To transfer design data to STAR-CCM+:

1. From the STAR-NX ribbon, click **Design Data**.



The *Design Data* dialog appears.



2. Select the design data to transfer:
   - **Design Parameters**—Click the *Design Parameters* tab and select the design parameters that you want to transfer from the list. You can select multiple items at once.

     Note: STAR-NX only recognizes native CAD features as design parameters. Only recognized design parameters are displayed in the list.
   - **Coordinate Systems**—Click the *Coordinate Systems* tab and select the coordinate systems that you want to transfer from the list. You can select multiple items at once.
   - **Reference Planes**—Click the *Reference Planes* tab and select the planes that you want to transfer from the list. You can select multiple items at once.
3. In each tab, click the checkbox next to the selected item to activate it.
4. Click **OK**.
5. Transfer the geometry to STAR-CCM+. See Transferring Geometry to STAR-CCM+.

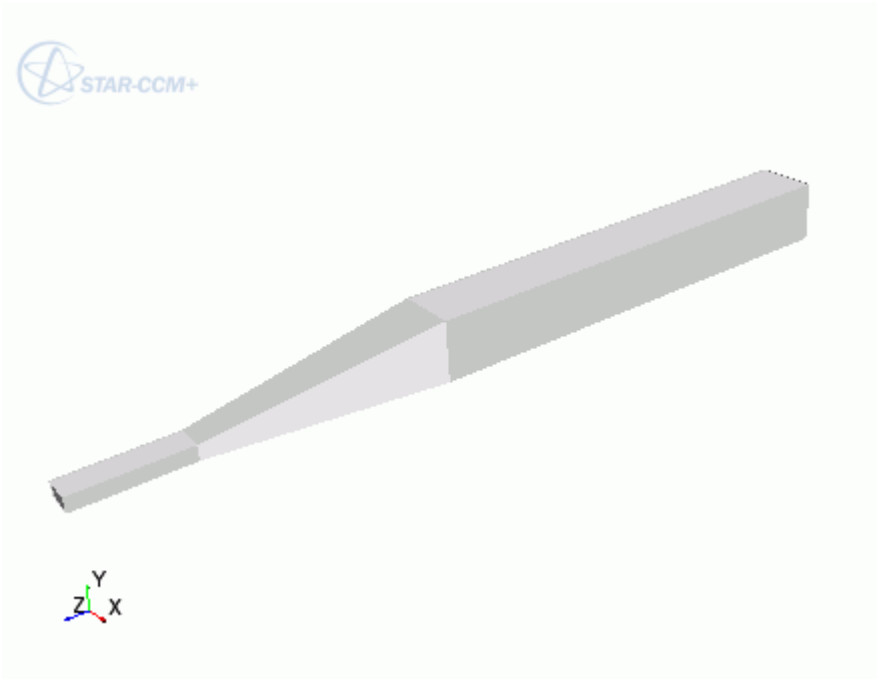
The geometry, along with the specified design data, is transferred to STAR-CCM+. Transferred design parameters are placed in the **Geometry** > **CAD Client Models** > **[geometry_name]** > **Design Parameters** node. Transferred planes are placed in the **Derived Parts** node in STAR-CCM+. Transferred coordinates

systems are placed in the **Tools** > **Coordinate Systems** > **Laboratory** > **Local Coordinate Systems** node in STAR-CCM+.

Modifying Geometry from Within STAR-CCM+

Once you transfer a design parameter to STAR-CCM+, you can modify the geometry from within STAR-CCM+ with respect to that parameter. This feature is useful for quickly modifying a geometry after performing an analysis in STAR-CCM+ without needing to return to the original CAD software.

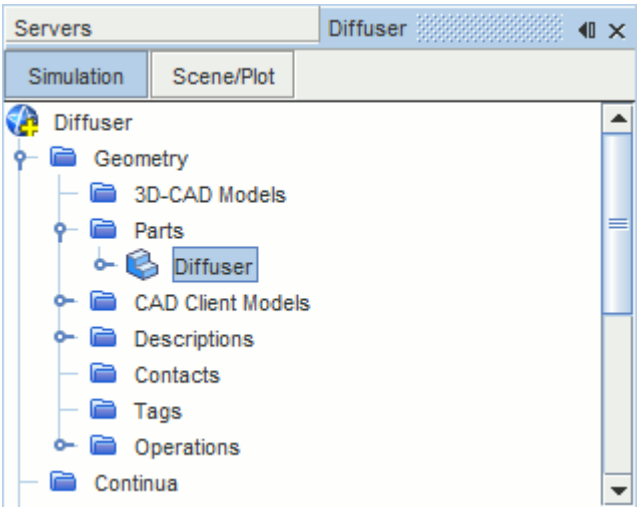
In the following example, the geometry of a diffuser is modified.



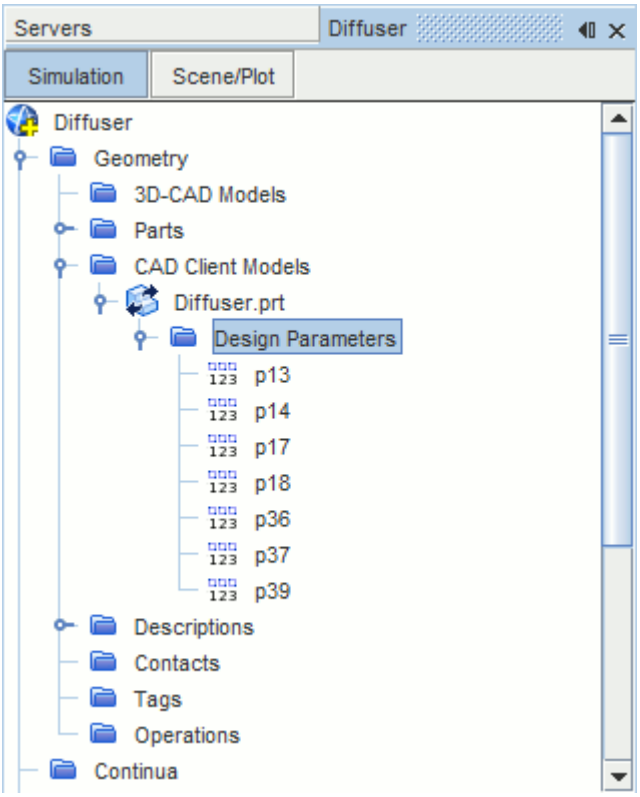
To modify a geometry in STAR-CCM+:

1. In STAR-NX, select the design parameters that you want to transfer to STAR-CCM+.
2. Transfer the geometry to STAR-CCM+. See Transferring Geometry to STAR-CCM+.

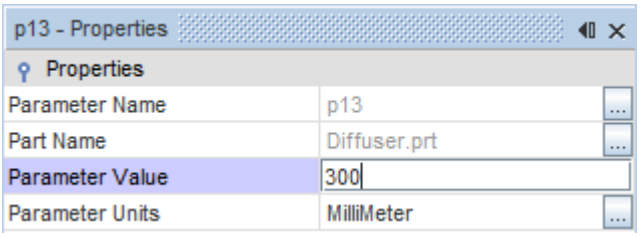
   Transferred geometries are located under the **Geometry** > **Parts** node.



Transferred design parameters are located under the **Geometry** > **CAD Client Models** > **[part_name]** > **Design Parameters** node.

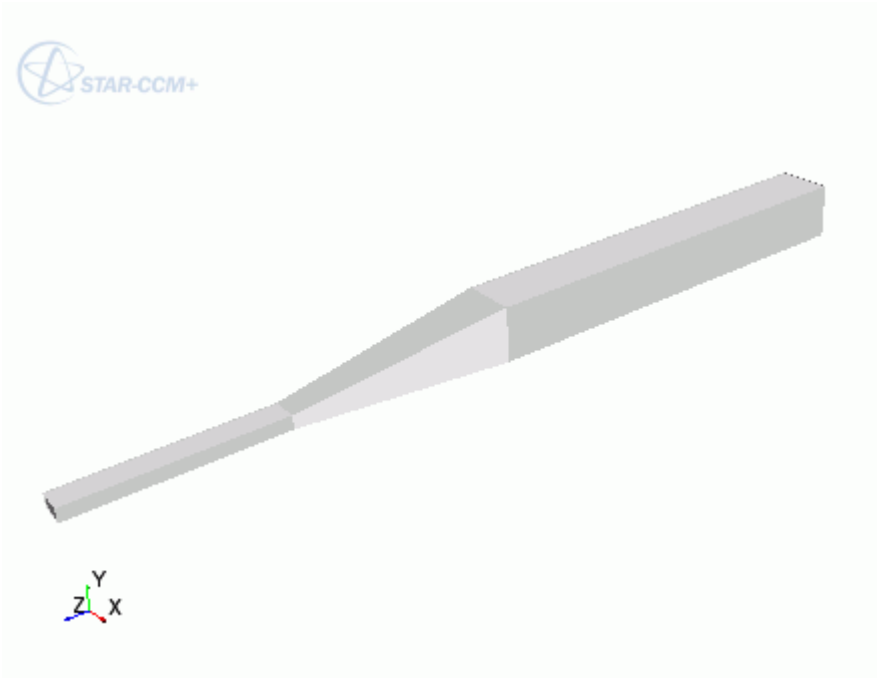


3. In STAR-NX, save the geometry file. The simulation file is also saved.
4. In STAR-CCM+, select the design parameter that you want to modify and set *Parameter Value* to the desire value.



5. Update the part in STAR-CCM+:
   - To update the part and save the updated parameters in the CAD file, right-click **Geometry** > **CAD Client Models** > **[part_name]** and select **Update Model and Save the CAD File**.
   - To update the part without updating the parameters in the CAD file, right-click **Geometry** > **CAD Client Models** > **[part_name]** and select **Update Model**.

   The geometry updates according to the design parameter changes.



6. Return to NX. The changes to the geometry are transferred back to the native CAD geometry.

This workflow maintains an active link between STAR-NX and STAR-CCM+. When working with Geometry Transfer, you can modify the geometry in either package and the geometry in the relevant package is updated accordingly.

Updating the Geometry in STAR-CCM+ with NX Closed

You can close NX and continue working with the simulation in STAR-CCM+. Doing so, however, breaks the link between the two software packages. Any further updates that you make to the geometry in STAR-CCM+ are not transferred back to the geometry in NX. Instead, these changes are stored in memory, which allows you to include the design parameters in your optimization scripts.

This workflow also allows you to release the NX and STAR-NX licenses when performing tasks in STAR-CCM+. In this case, the NX and STAR-NX licenses are temporarily checked out only when the updates to the geometry are stored in memory.

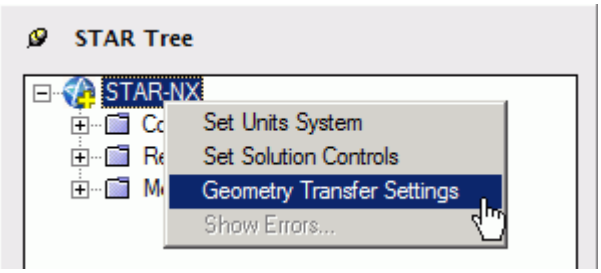
Transferring the geometry from STAR-NX will overwrite the geometry in STAR-CCM+.

Controlling How the Geometry is Transferred to STAR-CCM+

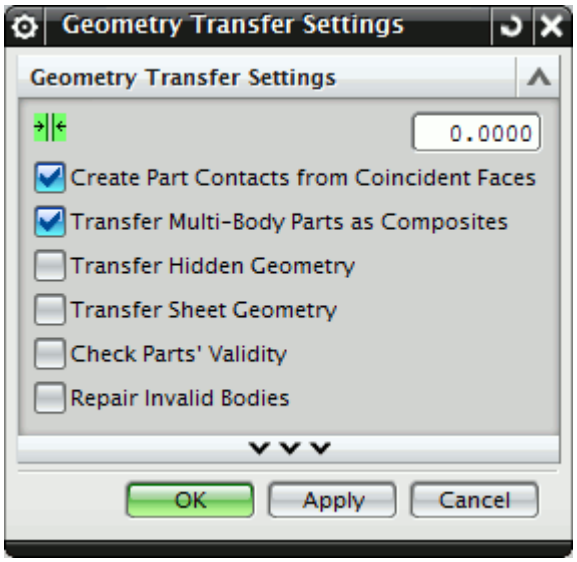
You can control how the geometry is transferred from NX to STAR-CCM+ through the *Geometry Transfer Settings* dialog.

To access the *Geometry Transfer Settings* dialog:

1. Right-click the STAR-NX root node and select **Geometry Transfer Settings**.



The *Geometry Transfer Settings* dialog appears.



2. Set the transfer parameters. See Geometry Transfer Settings Dialog.
3. Click **OK**.

Geometry Transfer Settings Dialog

Use the following properties to control how the geometry is transferred to STAR-CCM+.

Coincident Face Tolerance	Specifies the tolerance, in meters, for geometry in adjacent faces to be considered coincident. The default value is `1.0E-3`.
Create Part Contacts from Coincident Faces	Controls whether the geometry transfer feature searches for pairs of coincident faces on the CAD faces. For each pair found, part contact data is created and stored in the *Contacts* property (in STAR-CCM+). See the Creating Part Contacts from Coincident Entities section in the STAR-CCM+ user guide.
Transfer Multi-Body Parts as Composites	Specifies whether to transfer a single part containing multiple solid bodies as a composite part. Part contacts are created for each pair of coincident faces. Leaving this option deactivated merges all solid bodies in the part.
Transfer Hidden Geometry	Specifies whether to transfer hidden parts. By default, hidden parts in NX are not transferred. For files saved in STAR-NX v10.06 and earlier, this option is activated by default.
Transfer Sheet Geometry	Specifies whether to transfer sheet bodies as part of the transfer process. This option is turned off by default.
Check Parts' Validity	Provides you with information on whether the bodies are geometrically and topologically valid, from the Parasolid geometry kernel point of view. A simple message is displayed in the output window which states if the surface mesh is valid. For example: `Body 2 is not valid`.
Repair Invalid Bodies	Attempts to repair invalid bodies. The number of invalid bodies that are successfully repaired is displayed in the output window.

Understanding the STAR-NX Process

In STAR-NX the typical workflow for running a simulation is as follows:

1. Ensure that the CAD model is suitable for analysis
2. Extract the fluid volume
3. Define the material properties for each part of the model, for example, air, water, aluminum
4. Define model physics, for example, turbulence, temperature, gravity
5. Specify boundary conditions, for example, inlets, outlets, symmetry planes
6. Specify mesh settings
7. Define solution controls
8. Run the case
9. Post-process the results in STAR-CCM+

Once you have obtained an initial solution you may wish to re-run the case with some different settings, or with some changes to the geometry. To do this you can:

1. Return to STAR-NX and change settings or modify the geometry
2. Re-run the case

Using Suitable CAD Models in STAR-NX

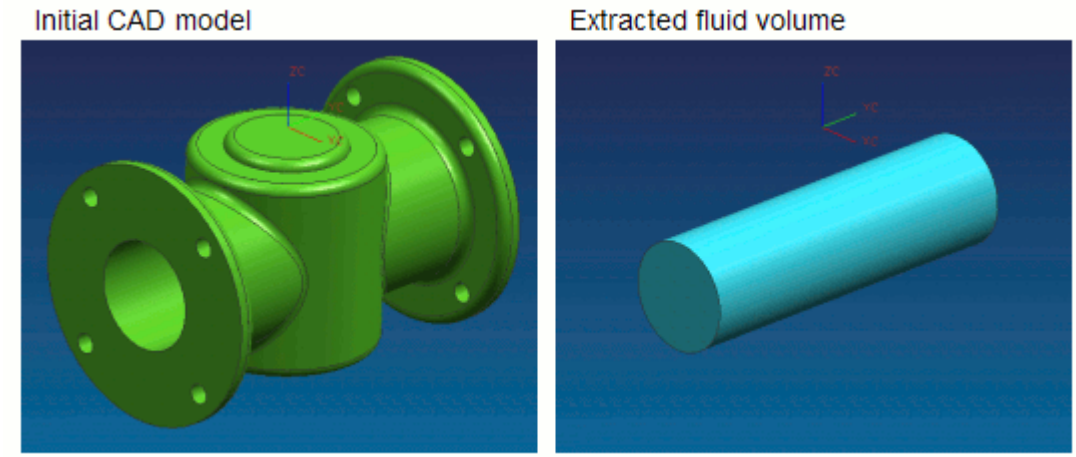
STAR-NX only supports the use of solid bodies. If a geometry is made up entirely of surfaces then it will not be possible to define any boundary conditions, and the meshing options will be greyed out.

It is good practice to create copies of the original geometry file(s) before importing them into STAR-NX.

Suppressed or deactivated parts and features are not supported by STAR-NX and will have no effect on the analysis. Hidden parts, however, are used in the analysis. Ensure that all desired parts and features are active before using them with STAR-NX.

Extracting the Fluid Volume

A CAD model usually represents a component made from solid material. However, in the case of flow simulation, a model representing the fluid volume is needed, requiring a different CAD model to be created. This is illustrated below:



In some instances, a simulation will require multiple regions, such as:

- Moving Reference Frames

  For rotating flow, such as turbines.
- Porous Media

  For representing regions with fine geometric structures
- Energy Source

  For simulating thermal specifications, such as conjugate heat transfer

These types of simulation require each region to be represented by individual contiguous parts within an assembly.

Defining Material Properties

Continua determines the characteristics of the fluid and solid entities used within the simulation. The behavior of a single continuum is primarily defined by the chosen material type and physics models.

Specification of the material type used in the simulation provides properties such as density, viscosity and conductivity, which are all used by the solver to calculate a solution. STAR-NX includes a database of common materials and allows you to modify material properties where necessary.

Defining Model Physics

Physics models determine the nature of the fluid, and can be described as either:

- Turbulent or Laminar
- Constant Density or Ideal Gas

Additional models can also be activated to meet the needs of various simulation requirements:

- Passive scalar
- Temperature
- Gravity

A region is defined as a volume domain in a simulation, surrounded by boundaries. Each region in the simulation must be represented by a part in the native CAD package. When multiple regions are required, multiple parts must be used within an assembly.

All regions must be assigned to a continuum, as they represent a volume of material in the simulation. Region settings can be used to define porous media, moving reference frames, and an energy source.

Specifying Boundary Conditions

To define boundary conditions you first need to select the surface or surfaces on the model to assign to a separate boundary.

A boundary is defined as a surface, which along with other boundaries, surround and characterize a region. Boundary conditions determine how the boundary will affect the material within a region, and consequently, the simulation as a whole.

Inlet boundaries determine how much fluid enters the region, using a velocity, mass flow rate or stagnation inlet. Outlet boundaries define the process for fluid leaving the region, which are defined with either a pressure or flow-split ratio. Wall boundaries contain the fluid, and can also be used to define a thermal specification that will influence the transfer of energy to or from the region. Symmetry boundaries can be used to avoid modeling the entire region where appropriate, and so reduce calculation time.

As boundary types can affect the way in which the volume mesh is constructed, it is important to set them before generating the mesh.

Specifying Mesh Settings

The STAR-CCM+ solver uses the finite volume method to obtain an approximate solution to the underlying equations of fluid flow. This requires a mesh to be created, which represents the fluid volume as a collection of much smaller volumes known as cells.

A surface mesh is first created based on the boundaries defining the regions in the simulation. Thereafter a volume mesh is created to discretize the entire model.

The accuracy of the results is directly affected by mesh quality. If it is necessary to improve the quality of the mesh, some of the ways in which this can be done are to:

- Increase the mesh density, that is, the number of cells
- Add further sub-surface prism layers to capture near wall flow
- Refine the mesh in areas where flow characteristics are expected to change rapidly

As volume mesh quality is largely governed by the surface mesh quality, it is common practice to create and inspect the surface mesh before creating a volume mesh.

Defining Solution Controls

Solution controls determine the level of accuracy the solver will work towards. As a high accuracy solution will take longer, it may be necessary to compromise on this in order to obtain a result in a shorter time.

STAR-NX includes five predefined solver settings ranging from fast (with low accuracy) to slow (with high accuracy). Advanced options are also available to provide further control.

Running the Case

When the simulation has been fully defined, the volume mesh is sent to STAR-CCM+ along with all the necessary data for running the simulation. The time to complete a simulation will vary depending on the mesh density and solution controls.

As the solver is running, residuals showing the rate at which individual equations are converging are shown in a plot. A scalar scene displaying absolute pressure is displayed on completion of the simulation.

Post-Processing the Results in STAR-CCM+

Post-processing of simulations is carried out using the STAR-CCM+ interface. This allows the results to be taken from the solver and displayed in a variety of scenes and plots.

Making Modifications to the Settings or Geometry

As STAR-NX is completely integrated within the NX environment, it is very simple to perform parametric studies that examine the response of the fluid to changes in geometry. Once the first analysis is complete, the boundary conditions, run settings and all other properties of the CFD model are stored within the CAD file. Any changes in the flow field due to design changes can be recalculated by generating a new mesh and running the solver.

Connecting to Remote Machines

STAR-CCM+ uses a client-server architecture, and as such the CAD client (STAR-NX) and the server (STAR-CCM+) are not required to run on the same machine.

Using an SSH (secure shell) connection, you can connect the client and server on two different machines, even when these are on different platforms. Due to how the CAD client and server interact in the background, the SSH connection must not ask for a username or a password.

There are two ways that you can connect STAR-NX and STAR-CCM+, depending on what your starting point is:

- You can start from the CAD client and launch the server on a remote machine. This method is useful for transferring CAD geometry from STAR-NX running on Windows to STAR-CCM+ running on Linux.
- You can start from the server and connect to the CAD client on a remote machine. This method is useful for connecting to STAR-NX to update the CAD geometry. With this method, you can perform an Optimate analysis on a cluster while the geometry is updated in STAR-NX.

Contents:

Running STAR-NX with Remote STAR-CCM+
Running STAR-CCM+ with Remote STAR-NX

Running STAR-NX with Remote STAR-CCM+

Starting from STAR-NX, you can choose to run the STAR-CCM+ server on a remote machine.

Using this method, you can set up the full CAE analysis in STAR-NX on your local machine, and then run the STAR-CCM+ server on the remote machine. You can also choose to transfer your geometry from the CAD client to the remote STAR-CCM+ server. This feature is useful if you want to import geometry from STAR-NX running on Windows to STAR-CCM+ running on Linux.

To configure STAR-NX to launch STAR-CCM+ on a remote machine:

1. Make sure that you have a working SSH connection between the two machines.

   For the SSH connection, you require an SSH server on the machine running STAR-CCM+ and an SSH client on the machine running STAR-NX. The SSH connection to the remote machine must not ask for a username or a password. To avoid possible issues with a connection that still requires a password, add `-o NumberOfPasswordPrompts=0` to the SSH command. Setting this argument to zero stops the connection prompting for a password when one is needed — something that could cause the process to hang. See SSH Command.
2. In the STAR-NX ribbon, click **Preferences**.
3. In the *Preferences* panel, set *Server Settings* to **Remote**.
4. Set the remaining settings for your connection to the remote STAR-CCM+ server. See Server Settings Panel.

   This information is used to build a command-line in the background when STAR-NX invokes STAR-CCM+.

5. If you are going to connect to the same remote host in future sessions, click **Save as Default Settings**.

The next time STAR-NX invokes STAR-CCM+, the STAR-CCM+ server launches on the specified remote machine. A STAR-CCM+ client is also launched on the local machine to allow you to see the active session on the remote machine.

Contents:

Server Settings Panel

Server Settings Panel

Specify the connection settings to the remote STAR-CCM+ server using the Server Settings pane.

Parameter	Description	
Server Host Type	Specifies whether to run the STAR-CCM+ server on the local host, or on a remote host.	
	Local	Runs the STAR-CCM+ server on the same host as STAR-NX.
	Remote	Runs the STAR-CCM+ server on the remote host that you specify.
SSH Command	Specifies the absolute path to the SSH program. If the path to the SSH program exists in your `Path` environment variable, you can leave this parameter as the default entry. You can include additional arguments in the SSH Command. The connection must not ask for a password or a username. To avoid possible issues with a connection that requires a password, add `-o NumberOfPasswordPrompts=0` to the SSH command. Setting this argument to zero stops the connection prompting for a password when one is needed, something that could cause the process to hang.	
Server Host	Specifies the remote host. You can enter the name, fully qualified domain name, IP address, or path to the host, depending on how your network is set up.	
Sim File Directory	Specifies the absolute path to the directory where you want to save the simulation file on the remote host. You must have read and write access to this directory. If a simulation file is present in this directory, STAR-CCM+ will use this file. If no file is found in this directory, a new one is created. An example directory on a Linux machine is shown below: `/users/linuxbox/james/documents/cadClientCases`	
Server Executable	Specifies the full path to the STAR-CCM+ executable on the remote host. An example path is shown below: `[STAR-CCM+_INSTALL_DIR]/star/bin/starccm+`	
Save as Default Settings	Saves the server settings that you specify in this panel as the default settings. Save these changes if you want to connect to the same remote server in future sessions.	

Running STAR-CCM+ with Remote STAR-NX

Starting from STAR-CCM+, you can connect to STAR-NX on a remote machine.

This feature is useful if you have previously transferred your geometry from STAR-NX on Windows to STAR-CCM+ on Linux, and you want to reconnect the two.

To connect STAR-CCM+ running on the local machine to STAR-NX running on a remote machine:

1. Make sure that you have a working SSH connection between the two machines.

   For the SSH connection, you require an SSH server on the machine running STAR-NX and an SSH client on the machine running STAR-CCM+. The SSH connection to the remote machine must not ask for a username or a password.
2. In STAR-CCM+, expand the **Geometry** > **CAD Client Models** > **[Model_Name]** and set *CAD-Client Update Mode* to **Remote**.

   The **Remote CAD Client Settings** node appears.
3. Set the remaining parameters for the CAD geometry. See CAD Geometry Settings.
4. Select the **Remote CAD Client Settings** node and set the connection setting for the remote CAD client. See Remote CAD Client Settings.
5. If you are connecting from a STAR-CCM+ server that is running on a cluster, activate the *Use Proxy Server* option and set up the proxy server. See Proxy Server Settings.

When you make changes to the imported CAD geometry in STAR-CCM+, the changes are passed back to STAR-NX on the remote machine and the geometry is transferred back into STAR-CCM+.

Contents:

Remote CAD Client Settings Reference

Remote CAD Client Settings Reference

Specify the connection settings to the remote CAD client in the Remote CAD Client Settings panel.

CAD Geometry Settings

Parameter	Description	
CAD File	Specifies the full name of the CAD geometry file, including the extension.	
CAD File Location	Specifies the location of the CAD geometry file on the remote machine. An example path is shown below: `C:\CAD Client Cases`	
CAD-Client Update Mode	Specifies whether STAR-NX is located on the local machine where STAR-CCM+ is installed, or whether STAR-NX is located on a remote machine.	
	Local	STAR-NX is located on the same local machine as STAR-CCM+.
	Remote	STAR-NX is located on a remote machine.

Remote CAD Client Settings

Parameter	Description	
Remote Host	Specifies the remote host where STAR-NX is installed. You can enter the name, full name, IP address, or path to the host, depending on how your network is set up.	
Remote CAD-Client Install Location	Specifies the absolute path to the root directory of the CAD client installation on the remote host. An example path to the default installation directory for version 12.06.### is shown below: `C:\Program Files\CD-adapco\12.06.###\STAR-CAD12.06.###`	
SSH Command	Specifies the command to use to run the SSH program. If you require a proxy server to connect to a cluster, you must use the following SSH command: `ssh -A -t <Cluster_Head_Node> ssh -A`	
Use Proxy Server	Specifies whether you require a proxy server to connect to a cluster.	
	Activated	Uses a proxy server as part of the connection to a cluster.
	Deactivated	Does not use a proxy server.

Proxy Server Settings

Parameter	Description
Proxy Host	Specifies the host name of the private network of the cluster. Enter the name, fully qualified domain name, or IP address. An example is shown below: `clusterhead.cm.cluster`
Proxy Public Host	Specifies the host name of the public network of the cluster. Enter the name, fully qualified domain name, or IP address. An example is shown below: `clusterhead.london.company.com`

Defining Materials and Physics

Defining the physical properties of a STAR-NX model involves specifying the materials, flow characteristics, operating conditions, and physics models.

These definitions and settings are accomplished through the use of continua and regions, which are described in the following sections.

Contents:

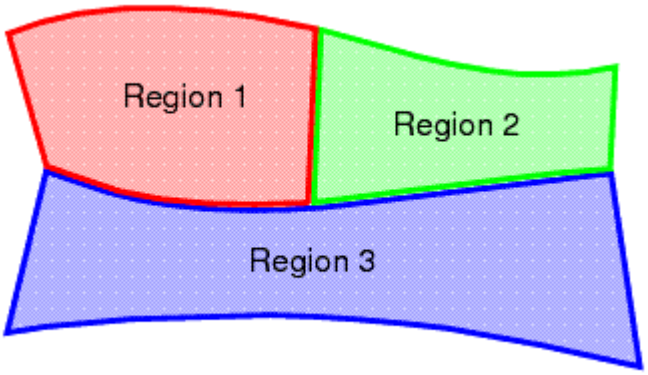
Working with Continua and Regions
Importing Geometry into STAR-CCM+ for Visualization

Working with Continua and Regions

Continua and regions are used together to capture the physical and material properties of a model.

A continuum represents a material in combination with a set of physics models that will be used to simulate the response of the material to the conditions imposed upon it. A continuum also provides the facility to define reference and initial values.

A region represents one or more volumes within the overall model, not necessarily contiguous, which will share the same continuum and set of boundary conditions. Regions are defined using CAD parts, and will be discretized by the volume mesher.



As well as holding a set of boundary conditions, regions can also be used to model specific physical behavior, including porous resistance, moving reference frames and energy sources.

Parts can be imported into STAR-CCM+ for visualization purposes without affecting the analysis.

Continua are assigned to single or multiple regions, which can contain single or multiple parts. In part mode, only one continuum and one region are available, as there is only one part. In assembly mode, new continua can be added to represent different materials present in the simulation, and new regions can be added to

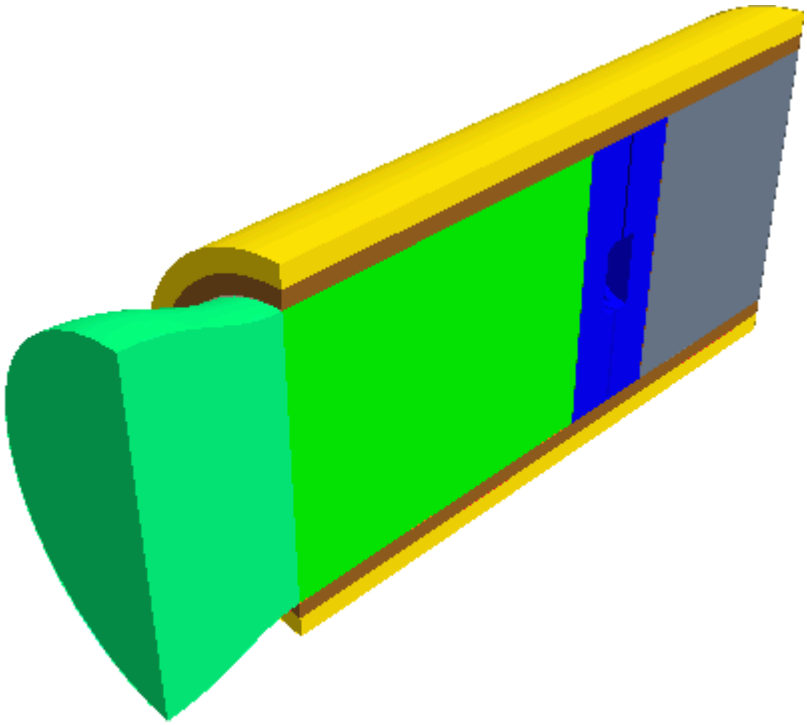
represent the parts composed of these materials. Continua and regions can be deleted if they are not required, although the simulation must always contain at least one of each.

There is a well defined relationship between continua, regions, boundary conditions and CAD parts.

- Every region must be associated with a continuum.
- Several regions can be associated with the same continuum.
- Every CAD part must belong to a region.
- Several CAD parts can be associated with the same region.
- Boundary conditions belong to regions, and cannot be shared between regions.
- Every CAD surface must have a boundary condition associated with it. A default wall boundary is applied to CAD surfaces that have not been specifically associated with a user-defined boundary condition.

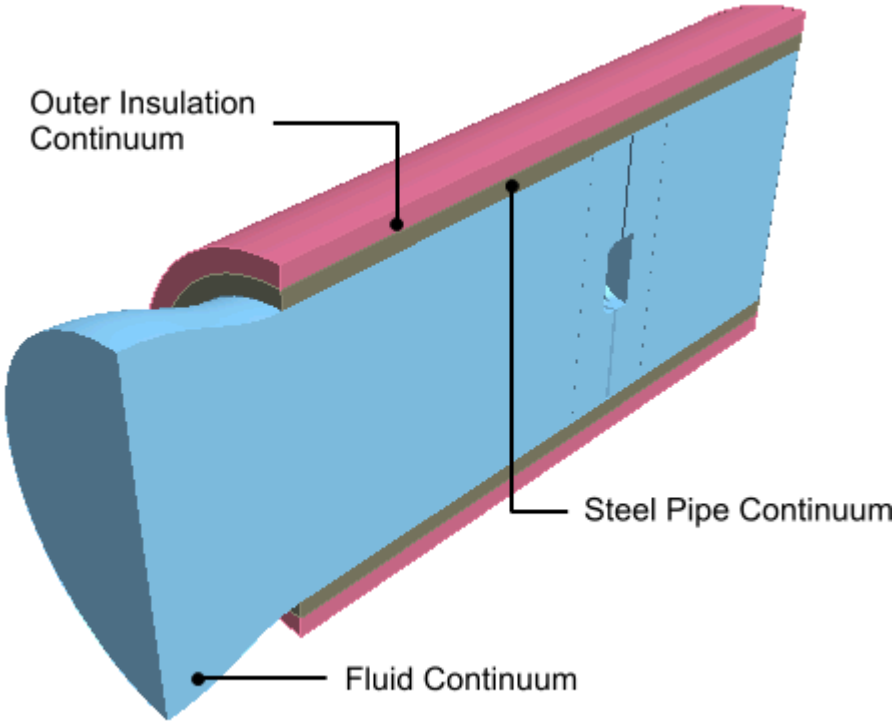
The following series of illustrations depict the relationship between CAD parts, continua, and regions, for a circular steel duct containing a rotating fan, and surrounded by a layer of insulation.

- Initially we have a collection of CAD parts in an assembly. These must be imprinted where appropriate to ensure that interfaces between parts can be created correctly.



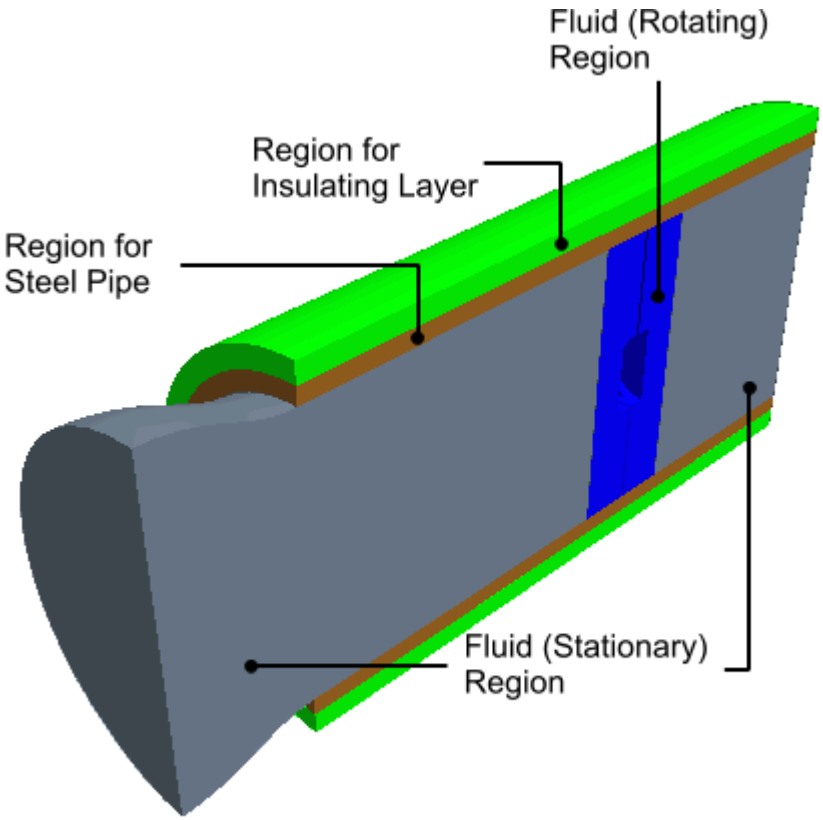
Assembly of CAD Parts (imprinted where necessary)

- Three continua are required to represent the three different materials present in the simulation: the fluid (for example, air), the steel pipe, and the insulation.



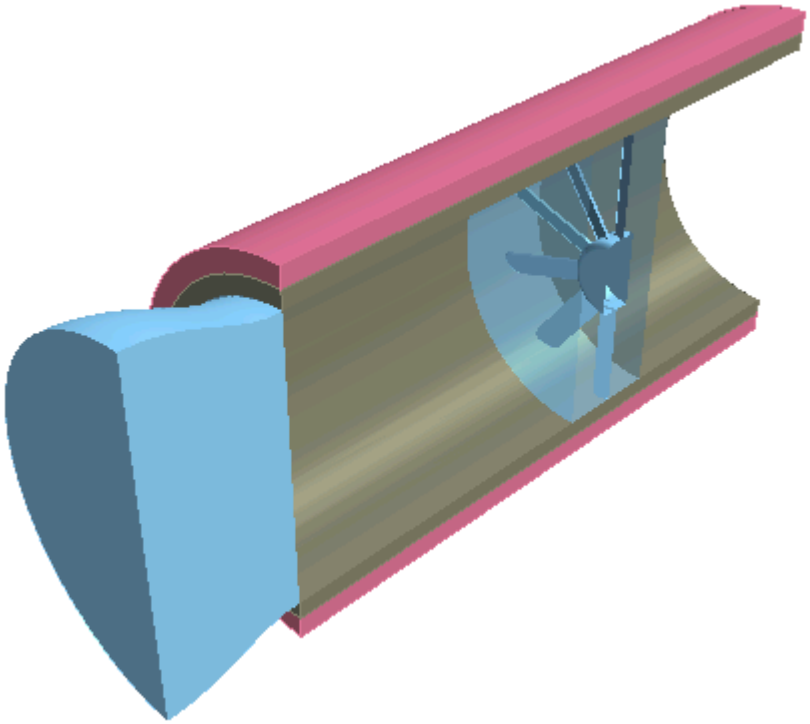
View of Continua Required for Simulation

- Two regions are required for the fluid portion of the simulation, as a rotating part is defined by applying a rotating frame of reference to a separate region surrounding the fan blades. Further regions are required for each of the solid parts (the steel pipe and insulating layer), which means a total of four separate regions overall.



View of Regions Required for Simulation

- A clearer picture can be given of the fan blade itself.



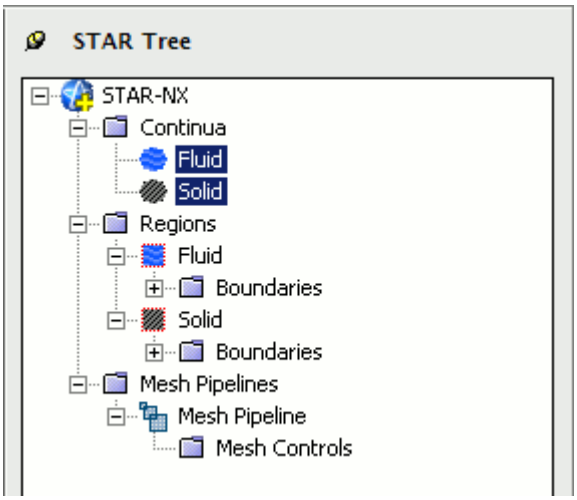
For an analysis containing a single part, for example a fluid-only case involving fluid flow through a converging-diverging pipe, the model will only require one continuum and one region in order to fully define the physical and material properties of the model. In this case, STAR-NX automatically assigns the default continuum to the default region, and no further assignments are necessary.

Contents:

Setting Up Continua
Setting Up Regions

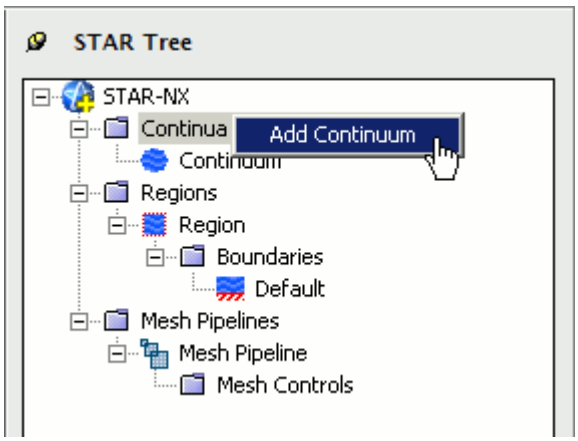
Setting Up Continua

To define different materials and associated physics for use in an assembly, you require additional continua. In each continuum you can specify a unique set of properties, such as a different material type or temperature. In assembly mode, continua are stored under a **Continua** node, as shown below.



To create an additional continuum:

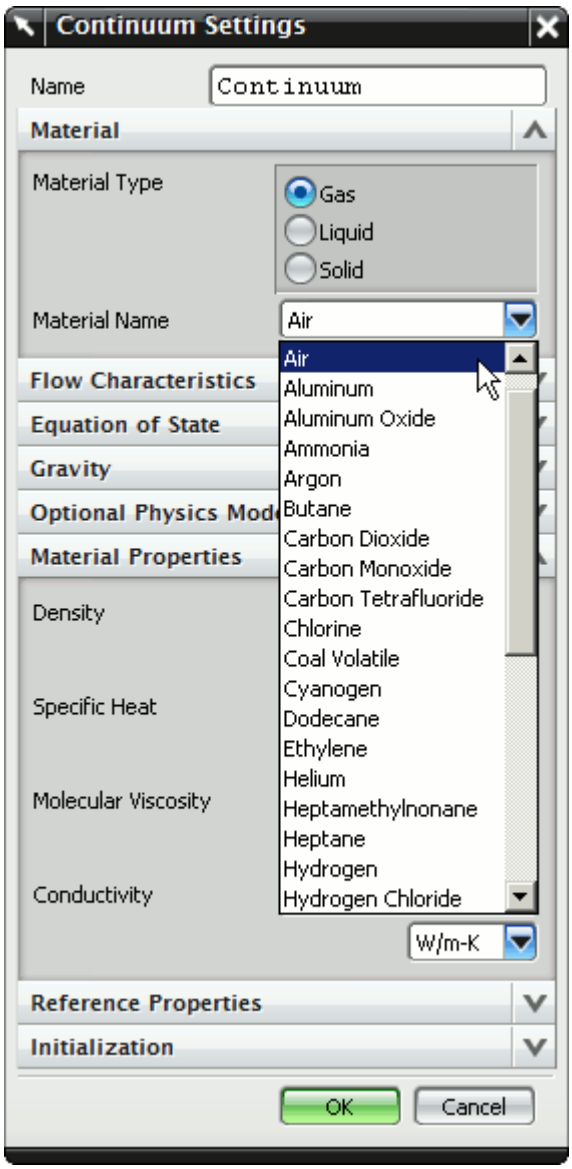
1. Right-click the **Continua** node in the object tree and select **Add Continuum**.



The *Continuum Settings* dialog appears.

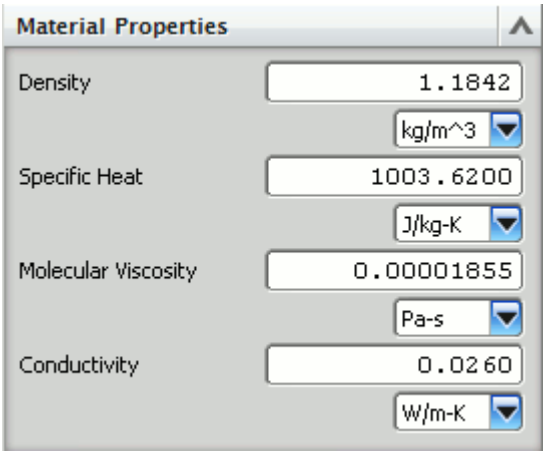
2. In the *Name* field, enter a name for the continuum.
3. In the *Materials* group, select the type of material that you want to define:
   - **Gas**
   - **Liquid**
   - **Solid**
4. Select a material from the list.

   You can modify the properties of any material that you select.



Note: You can select only one material per continuum. If your simulation contains a number of materials, you must create additional continua. This action is only possible for assemblies.

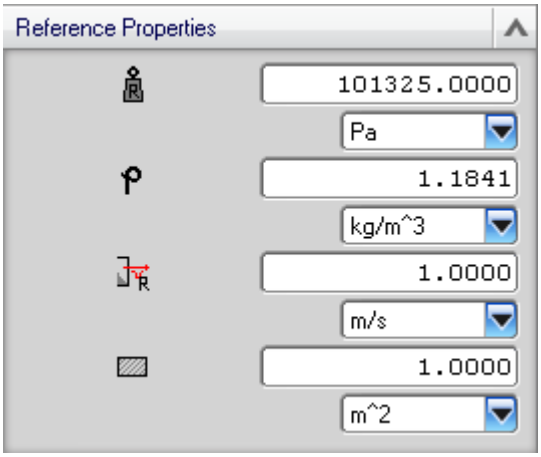
5. In the *Flow Characteristics* group, select the type of flow that you want to model: **Laminar** or **Turbulent**. This group only appears for fluids.
6. In the *Equation of State* group, select the equation of state : **Constant Density** or **Ideal Gas**. This group only appears for gases.
7. In the *Optional Physics Model* group, activate any optional physics models: **Passive Scalar** and **Temperature**.
8. Define the material properties in the *Material Properties* group.



See STAR-NX Material Properties Panel.

Note: You can change units of properties from the drop-down menu.

9. In the *Reference Properties* group, define the reference properties.

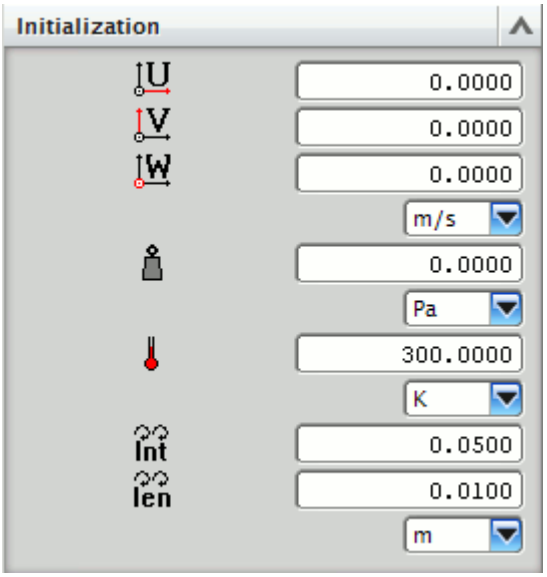


See STAR-NX Reference Properties Panel.

10. If you want to include the body force due to gravity when calculating the momentum equations in STAR-CCM+, activate the gravity model in the *Gravity* group box.

    See Activating the Gravity Model in STAR-NX.

11. Set the initial conditions in the *Initialization* group box.



See STAR-NX Initialization Panel.

Specifying initial conditions that are close to the expected solution can reduce the number of iterations required to reach a solution. You can set the initial conditions similar to the inlet conditions of the fluid domain.

Initial conditions are specified at a continuum level; if no initial conditions are specified, STAR-NX will use default values to initialize the solution.

12. Click **OK**.

    The continuum is added to the object tree.

Once you have created a continuum, you can assign it to a region.

Contents:

STAR-NX Continuum Properties
Activating the Gravity Model in STAR-NX

STAR-NX Continuum Properties

The continuum dialog has a number of properties that you can define. This section provides a description for each property.

STAR-NX Material Properties Panel

You can define the following material properties in STAR-NX:

Property	Description
Density	Available for all types of materials. For gases, this property is available when you use the Constant Density equation of state.

Property	Description
Specific Heat	Available for all material types when you activate the Temperature physics model.
Molecular Viscosity	Only available for fluids.
Conductivity	Available for all materials when you activate the temperature physics model.
Molecular Weight	Only applies to gases. This property is available when you use the Ideal Gas equation of state.

STAR-NX Reference Properties Panel

You can define the following continuum reference properties:

Property	Description
(*Reference Pressure*)	The pressure values that are defined in the simulation are all relative to the reference pressure. For more information, see the What is the Reference Pressure? section.
(*Reference Temperature*)	The reference temperature is used to define the operating temperature of the fluid in the buoyancy source term of the Boussinesq model. This parameter is used by STAR-CCM+ when the temperature, constant density, and gravity physics models are enabled.
(*Reference Density*)	These quantities are not used as part of the CFD calculation, but is necessary to calculate the force coefficients when post-processing.
(*Reference Velocity*)	
(*Reference Area*)	

STAR-NX Initialization Panel

Specifying initial conditions that are close to the expected solution can reduce the number of iterations required to reach a solution. You can also set the initial conditions to match inlet conditions of the fluid domain. You can define the following continuum reference properties:

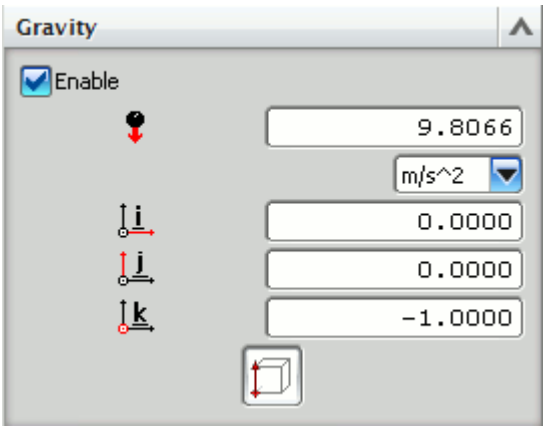
Property	Description
(*U Component of Velocity*)	The U direction vector for the initial velocity.
(*V Component of Velocity*)	The V direction vectors for the initial velocity.
(*W Component of Velocity*)	The W direction vectors for the initial velocity.
(*Initial Pressure*)	The initial pressure. The physical value of the initial pressure defines the starting pressure in the regions

Property	Description
	to which the continuum is assigned. The initial pressure is specified with respect to the reference pressure. The default initial pressure is `0 Pa`.
(*Initial Temperature*)	The initial temperature. This physical value of the initial temperature defines the starting temperature throughout any region to which the continuum is assigned. The default initial temperature is `300 K`.
(*Turbulence Intensity*)	The initial turbulence intensity. The default value is `0.05` (`5%`)
(*Turbulent Length Scale*)	The initial turbulent length scale. The default value is `0.01 m`.

Activating the Gravity Model in STAR-NX

If you require the gravity model in STAR-NX, you can activate it in the *Continuum Settings* dialog:

1. In the *Continuum Settings* dialog, activate the **Enable** checkbox in the *Gravity* group.

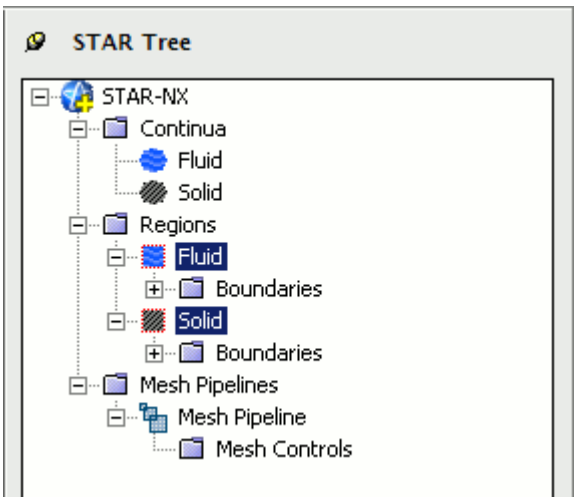


2. Set the acceleration due to gravity in the (*Acceleration due to gravity*) box.
   You can change the units from the drop-down list.
3. Specify the direction vector for the gravity in the `[i, j, k]` *Components of Gravity* box.
4. Click **OK**.
   The continuum is added to the object tree.

Setting Up Regions

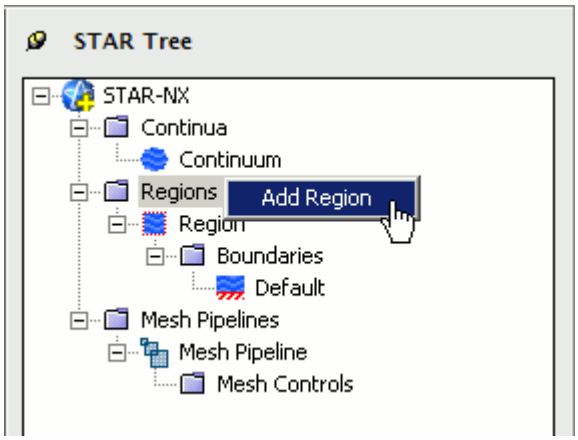
Additional regions are required when modelling multiple region physics, or when specific behavior such as porous resistance is required for a particular part within the same continuum. Assign parts from an assembly to regions. You can control the properties of these regions through the assigned continuum.

In assembly mode, regions are stored under the **Regions** node in the object tree, as shown below.

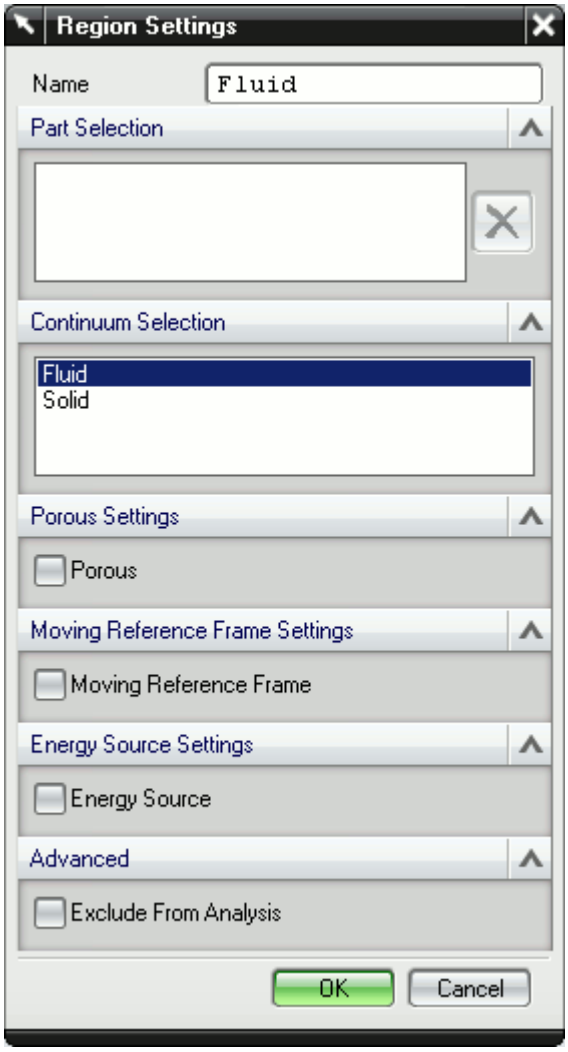


To create an additional region:

1. Right-click the **Regions** manager node in the object tree and select **Add Region**.



The *Region Settings* dialog appears.

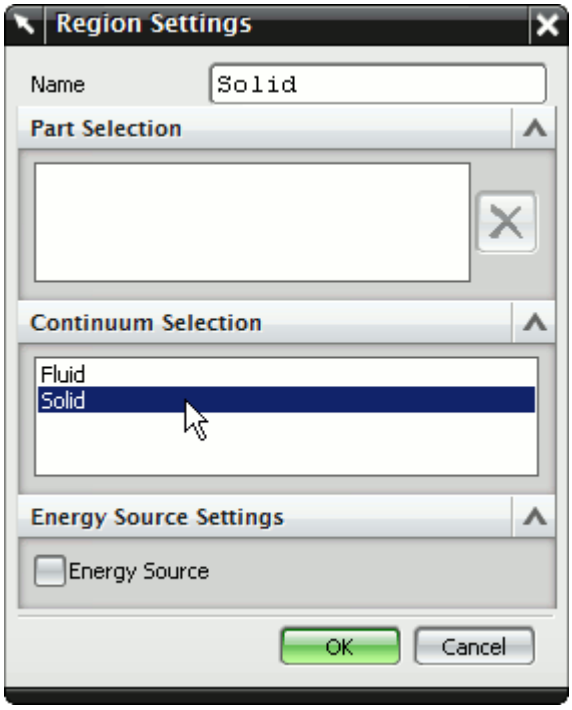


2. In the *Name* property, enter a name for the region.
3. Select the parts that you want to include in the region.

   You can select multiple parts from the graphics window or the NX feature tree.

   Selected parts appear in the *Parts Selection* group box field.
4. Select a continuum to apply to the region.

   All available continua are listed here.



Note: Assigning a continuum to a region causes the region to inherit the physical and material properties defined in the continuum

5. Activate any additional region physics for the region:
   - Porous. See Defining a Porous Region.
   - Moving Reference Frame. See Defining a Moving Reference Frame.
   - Energy Source. See Defining an Energy Source.
6. If you want to import parts into STAR-CCM+ for visualization purposes, but do not want the region to be included in the CFD calculations, activate the *Exclude from Analysis* option. See Importing Geometry into STAR-CCM+ for Visualization.
7. Click **OK**.

A new region is added to the STAR-NX object tree under the **Regions** node.

The parts that you assign to a region define the geometric extent of the region. Within this geometry, properties and behavior of the material is governed by the physics models and material properties defined in the region and continuum settings.

Note: Each region must be assigned at least one part, otherwise it will be ignored during the CFD analysis.

By default, every STAR-NX model contains one continuum, and this is automatically assigned to the default region and each additional region that you create. If multiple continua are defined in the model, you can choose a different continuum to assign to each region. This enables you to control the physical properties of each region, including varying material types.

All the parts in an assembly are assigned to the default region (this is the initial region created by STAR-NX). When you assign a part to another region, it is automatically removed from the default region. If you remove a part from a region, it is reassigned back to the default region.

Contents:

Defining a Porous Region
Defining a Moving Reference Frame
Defining an Energy Source

Defining a Porous Region

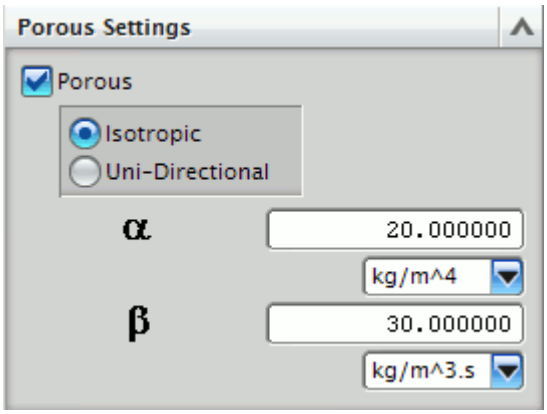
Porous resistance affects fluid flow through a region in a particular direction.

To define a porous region:

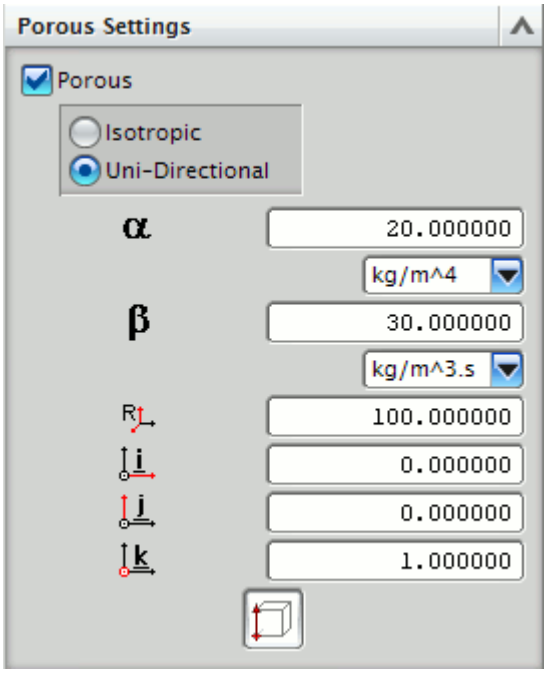
1. In the *Region Settings* dialog, activate the *Porous* tab.

   The Porous settings appear.

2. Select the type of resistance to model:
   - **Isotropic**



   - **Uni-Directional**



3. Specify the porosity settings. See Porous Region Panel Settings.

Contents:

Porous Panel Settings

Porous Panel Settings

Porous resistance affects the fluid flow through a region in a particular direction. There are two types of porous resistances available: Isotropic and Uni-Directional.

Parameter	Description
Isotropic	
α (*Alpha*)	Specifies inertial resistance. You can change the units using the drop-down list.
β (*Beta*)	Specifies viscous resistance. You can change the units using the drop-down list.
Uni-Directional	
α (*Alpha*)	Specifies inertial resistance. You can change the units using the drop-down list.
β (*Beta*)	Specifies viscous resistance. You can change the units using the drop-down list.
(*Cross-Flow Resistance Factor*)	Specifies the cross-flow resistance factor.
(*i Component of Flow Direction*)	Specify the components that define the direction of the flow in the porous region. These values are

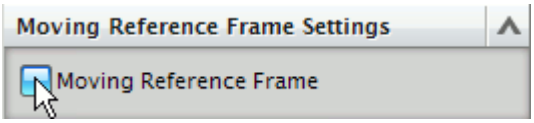
Parameter	Description
j (*j Component of Flow Direction*)	required to determine the coefficients of inertial and viscous resistance in a direction perpendicular to the specified vector.
k (*k Component of Flow Direction*)	

Defining a Moving Reference Frame

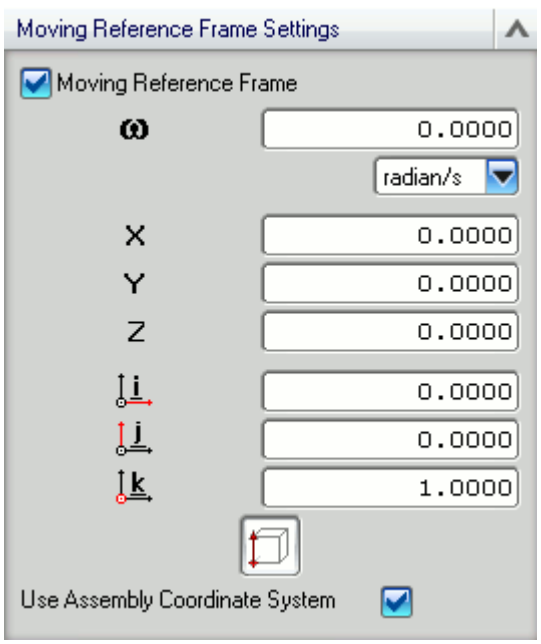
You can use moving reference frames to model region motion.

To define a moving reference frame:

1. In the *Region Settings* dialog, activate the *Moving Reference Frame* tab.



The Moving Reference Frame settings appear.



2. Specify the Moving Reference Frame parameters. See Moving Reference Frame Panel Settings.

Contents:

Moving Reference Frame Panel Settings

Moving Reference Frame Panel Settings

You can use Moving Reference Frames to model region motion, such as rotation.

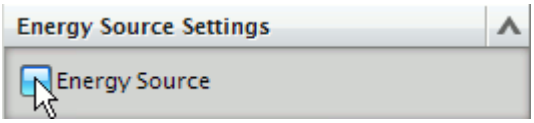
Parameter	Description
(*Rotational Velocity*)	Specifies the angular velocity. You can change the units from the drop-down list.
X (*X Coordinate of Axis Origin*)	Specify the coordinates for the origin of the axis of rotation. The units that are used here are the same as the file defaults.
Y (*Y Coordinate of Axis Origin*)	
Z (*Z Coordinate of Axis Origin*)	
(*i Component of Axis Vector*)	Specify the direction of the axis of revolution.
(*j Component of Axis Vector*)	
(*k Component of Axis Vector*)	
(**Set Axis Origin and Vector**)	Brings up the *Vector* dialog, where you can set the origin of the axis of rotation, and the vector direction using native NX functionality.
Use Assembly Coordinate System	Controls whether the properties of the axis of rotation are defined using the coordinate system of the assembly or the coordinate system of the selected part. To use the coordinate system of the assembly, activate the *Use Assembly Coordinate System* this option. If you select multiple parts for the new region, the *Use Assembly Coordinate System* option is automatically activated and grayed out

Defining an Energy Source

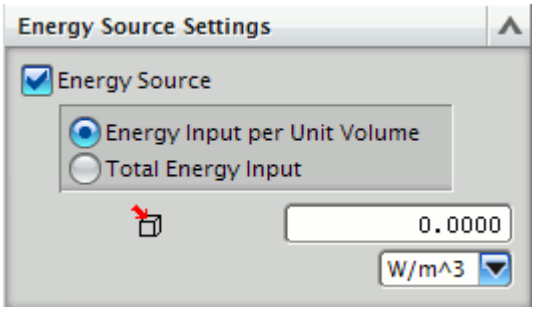
You can define a direct source or sink of energy to or from a region using the energy source model. This model is only available if you activated the Temperature physics model in the continuum.

To specify that a region is an energy source:

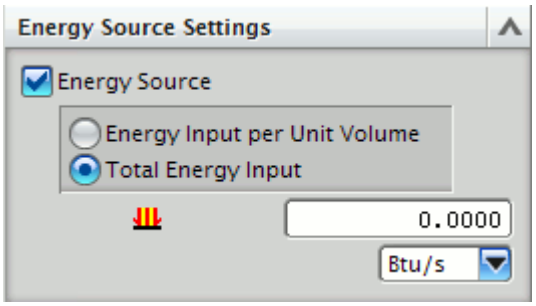
1. Make sure that the Temperature model is activated in the continuum that is applied to the region. See Setting Up Continua.
2. In the *Region Settings* dialog, activate the *Energy Source* tab.



   The Energy Source settings appear.
3. Select the type of energy source to apply:
   - **Energy Input per Unit Volume**



- **Total Energy Input**



4. Specify the energy source settings:
   - For energy input, set (*Energy Input per Unit Volume*) to the desired value.
   - For total heat, set (*Total Energy Input*) to the desired value.

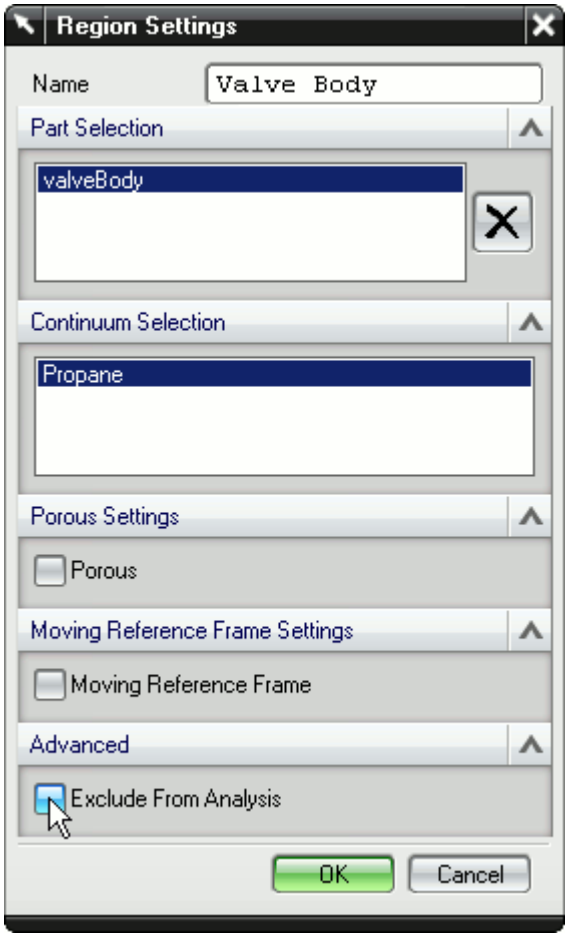
Importing Geometry into STAR-CCM+ for Visualization

You can import CAD parts, that do not form part of the analysis, into STAR-CCM+ for visualization purposes.

Add these parts to a separate region—this region is excluded from the analysis. This feature is useful for showing the original geometry in a post-processing scene in STAR-CCM+. For example, for an internal fluid volume of a pipe, the original geometry is the solid pipe.

To exclude CAD parts from the analysis:

1. In the *Region Settings* dialog, add the CAD parts that you want to use for visualization purposes.
2. Activate the *Exclude From Analysis* checkbox in the *Advanced* panel.



3. Click **OK**.

When you run the case, a surface mesh is generated for this region using the settings specified in the Mesh Pipeline.

Note: As is the case for all assemblies, you must imprint all parts that are used for visualization purposes onto any surface that they cut, otherwise you will not be able to generate the surface mesh.

An example of this feature is shown in the following section, where a valve and valve body geometry are imported into STAR-CCM+ for visualization.

Contents:

Setting Up a STAR-CCM+ Scene to Display Imported Geometry from STAR-NX

Setting Up a STAR-CCM+ Scene to Display Imported Geometry from STAR-NX

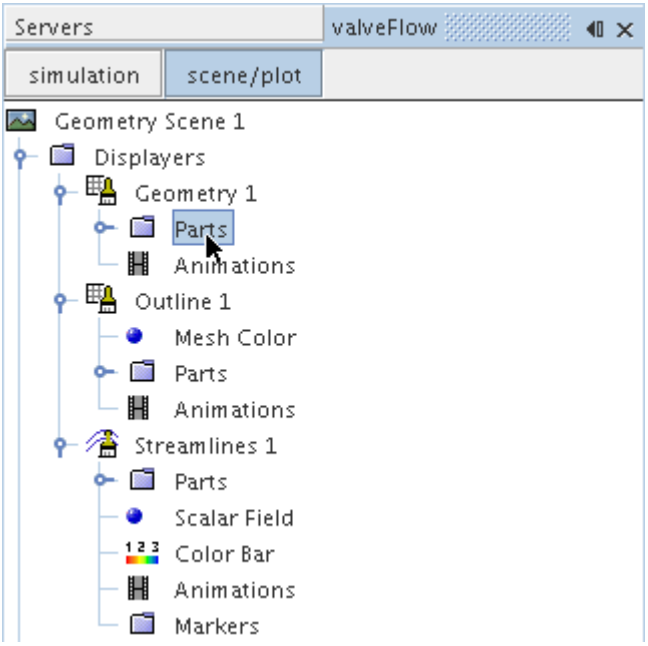
You can visualize geometry that is not included in the analysis. For example, you can add the valve and valve housing geometry to a Geometry Scene in STAR-NX, even though this does not have any effect to the flow.

To use parts that are excluded from the analysis in a scene:

1. In STAR-CCM+, create a new scene.

   In the following example, a Geometry Scene is used.
2. Create a streamlines derived part.
3. Click on the **scene/plot** button.

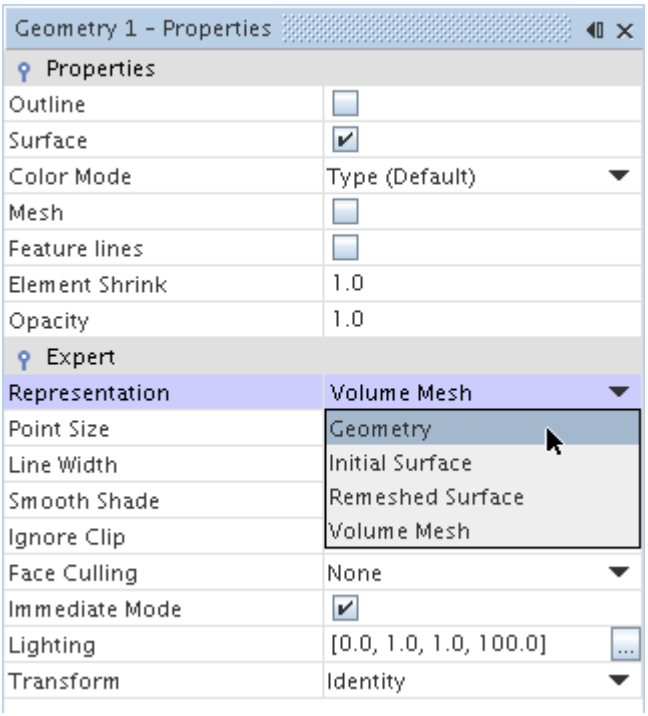
   The object tree switches to the scene nodes.
4. Double-click on the **Geometry Scene** > **Displayers** > **Geometry** > **Parts** node.



5. In the *Edit* dialog:
   a) Deactivate the **Regions** node.
   b) Expand the **Parts** node and activate the part that corresponds to the valve body.

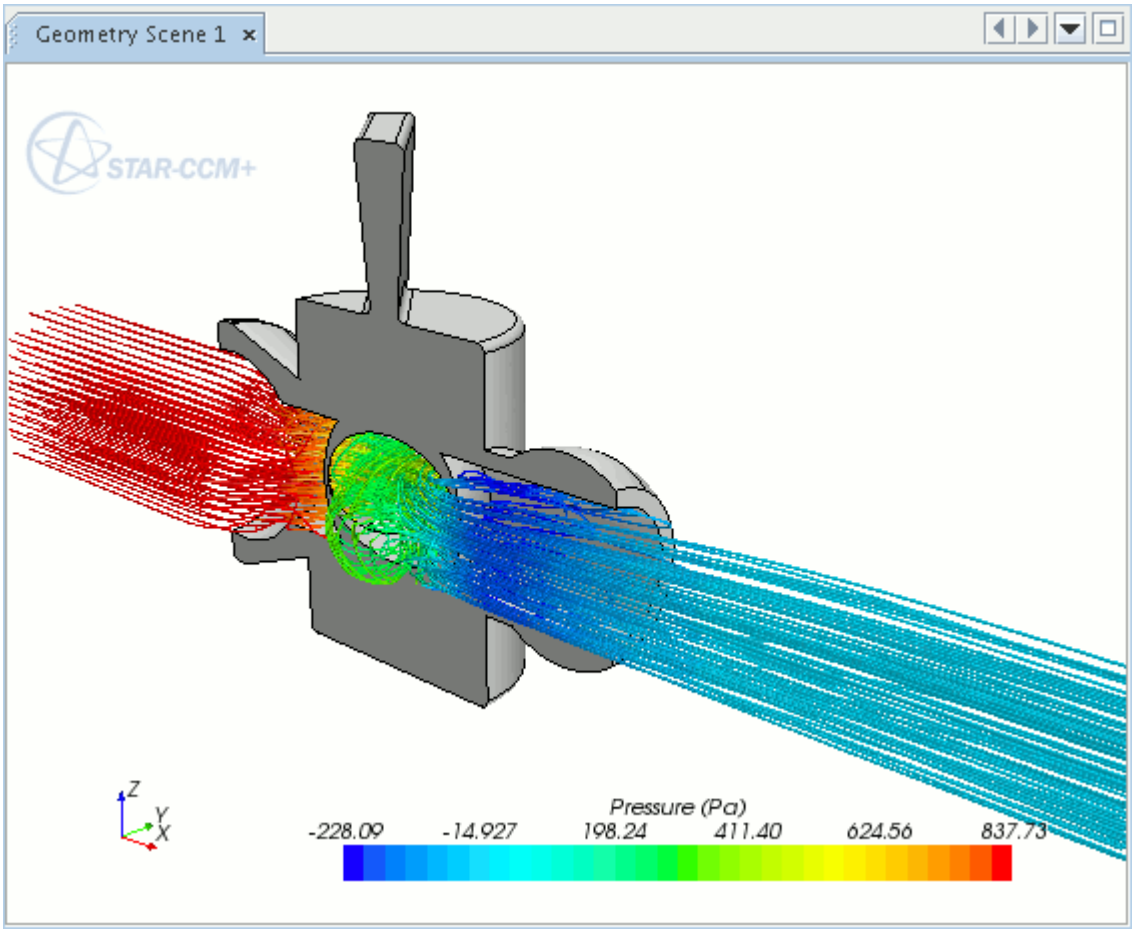
      This is the region that you chose to exclude from the analysis in STAR-NX.
6. Click **OK**.
7. Select the **Geometry** node and set *Representation* to **Geometry**.

   The imported geometry only contains a surface mesh and does not appear by default (the default *Representation* is set to **Volume**). This is why you must change the *Representation* to **Geometry**.



8. Activate the *Outline* option to change the color of the outlines from red to black.

The finished scene is shown below.



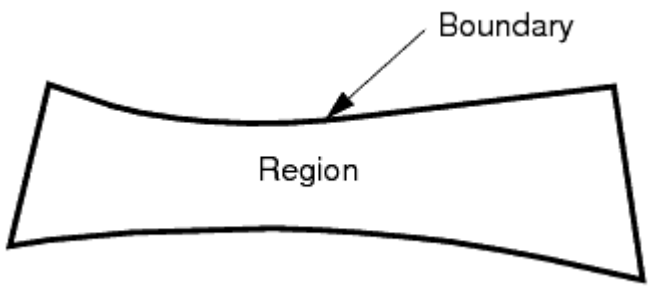
Displaying Imported Geometry in Scalar and Vector Scenes in STAR-CCM+

To display an imported geometry in a Scalar or Vector scene, you need to add a geometry displayer to the existing scene. You can then display the geometry by following steps described in Setting Up a STAR-CCM+ Scene to Display Imported Geometry from STAR-NX.

Setting Boundary Conditions

Boundary conditions tell the STAR-CCM+ solvers how the substance within the region interacts with its environment. A velocity inlet boundary, for example, tells the solvers that an amount of substance is being added to the fluid domain at a certain rate.

A STAR-NX boundary is made up of one or more CAD surfaces within the simulation which, along with other boundaries, surround and define a region.



The boundary is discretized by the surface mesher, which is then used by the volume mesher to mesh the entire region.

It is important to set the boundary conditions before creating the volume mesh, as some parts of the volume mesh, such as prism layers, depend on the type of boundary.

By default, each region contains only one boundary. To define more boundaries you need to add boundaries to the region. You can modify the properties of existing boundaries by editing its boundary node.

Some boundaries may require the specification of turbulence quantities, temperature, and passive scalar concentration, depending on whether these modeling options have been activated in Continuum Settings.

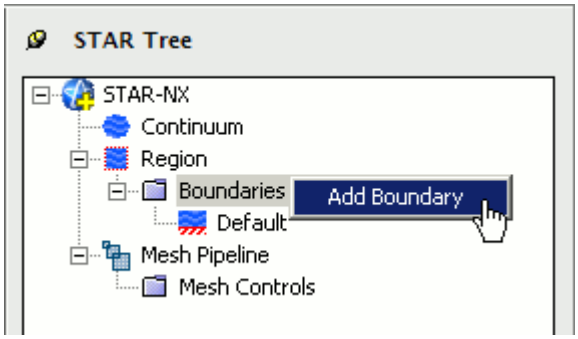
Contents:

Creating a New Boundary

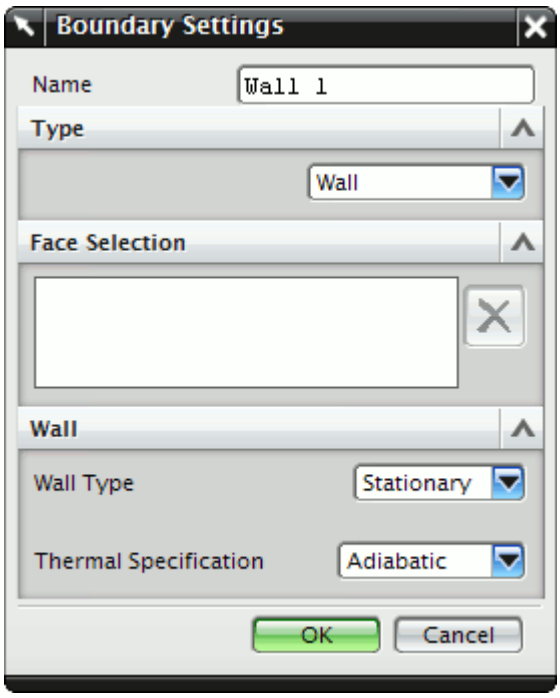
Creating a New Boundary

To create a new boundary:

1. Right-click the **Boundaries** node within the object tree and select **New**.



The *Boundary Settings* dialog appears.



2. On the CAD model, select the surfaces that you want to include in the boundary. See Assigning Surfaces to Objects.

   These selections are reported in the *Face Selection* box in the *Boundary Settings* dialog.
3. In the *Name* field, enter a name for the boundary.
4. Select the boundary type and set the boundary conditions:
   - **Velocity Inlet**. See Velocity Inlet Boundary.
   - **Mass Flow Inlet**. See Mass Flow Inlet Boundary.
   - **Stagnation Inlet**. See Stagnation Inlet Boundary.
   - **Outlet**. See Outlet Boundary.
   - **Pressure Outlet**. See Pressure Outlet Boundary.
   - **Symmetry Plane**. See Symmetry Plane Boundary.
   - **Wall**. See Wall Boundary.
   - **Custom**. See Custom Boundary.

5. Click **OK**.

Contents:

Boundary Type Reference
Common Boundary Conditions

Boundary Type Reference

This section contains the various boundary types that are available in STAR-NX.

Velocity Inlet Boundary

A velocity inlet boundary represents the inlet of a duct at which you know the flow velocity. The velocity inlet settings are displayed below.

Property	Description
Normal Velocity	
(*Normal Velocity*)	Specifies the velocity normal to the surface. You can change the units using the drop-down list.
Cartesian Velocity Components	
U (*U Component of Velocity Vector*)	Specify the components of the velocity vector. You can change the units using the drop-down list.
V (*V Component of Velocity Vector*)	
W (*W Component of Velocity Vector*)	
Cylindrical Velocity Components	
V_r (*r Component of Velocity Vector*)	Specify the cylindrical components of the velocity vector. You can change the units using the drop-down list.
V_θ (*theta Component of Velocity Vector*)	
V_z (*z Component of Velocity Vector*)	
X (*X Location of Axis of Origin*)	Specify the coordinates of the origin for the velocity vector.
Y (*Y Location of Axis of Origin*)	
Z (*Z Location of Axis of Origin*)	
i (*i Component of Direction Vector*)	Specify the components for the direction vector.
j (*j Component of Direction Vector*)	
k (*k Component of Direction Vector*)	
(**Set Axis Origin and Vector**)	Brings up the *Vector* dialog, where you can set the origin and direction of the velocity vector using native NX functionality.

Mass Flow Inlet Boundary

A mass flow inlet boundary represents the inlet of a duct at which you know the mass flow rate. The mass flow settings are displayed below.

Property	Description
Mass Flow Normal to Surface	
(*Mass Flow Rate*)	Specifies the mass flow rate, normal to the boundary surface. You can change the using the drop-down list.
Mass Flow in Specified Direction	
(*Mass Flow Rate*)	Specifies the mass flow rate. You can change the using the drop-down list.
(*i Component of Mass Flow*)	Specify the direction components of the mass flow.
(*j Component of Mass Flow*)	
(*k Component of Mass Flow*)	

Stagnation Inlet Boundary

The stagnation inlet boundary is an inlet condition that is well-posed for compressible flows, although it is equally valid for incompressible flows. The stagnation conditions refer to the conditions in an imaginary plenum, far upstream, in which the flow is completely at rest. The stagnation inlet settings are displayed below.

Property	Description
Stagnation Flow Normal to Surface	
(*Stagnation Pressure*)	Specifies the stagnation pressure. You can change the units using the drop-down list.
Stagnation Flow in Specified Direction	
(*Stagnation Pressure*)	Specifies the stagnation pressure. You can change the units using the drop-down list.
(*i Component of Direction Vector*)	Specify the direction components for the flow.
(*j Component of Direction Vector*)	
(*k Component of Direction Vector*)	

Outlet Boundary

An outlet boundary splits the flow between multiple outlets using a fraction of the total mass flow through each of the outlet boundaries. The outlet panel settings are provided below.

Property	Description
60 40 (*Flow-Split Ratio*)	Specifies the ratio of the split as a value between 0 and 1. The sum of all the flow-split ratio values in a region must equal 1. If you only have one flow-split boundary in the simulation, all the flow is assumed to exit through this boundary.

Pressure Outlet Boundary

A pressure boundary is a flow outlet boundary at which you specify the pressure. The pressure outlet panel settings are provided below.

Property	Description
(*Pressure*)	Specifies the pressure on the selected boundary. You can change units using the drop-down list.

Symmetry Plane Boundary

A symmetry plane boundary represents an imaginary plane of symmetry in the simulation. The solution that is obtained with a symmetry plane boundary is identical to the solution that would be obtained by mirroring the mesh about the symmetry plane (in half the resulting domain). There are no settings which are specific to symmetry plane boundaries.

Wall Boundary

A wall boundary represents an impermeable surface. The wall panel settings are provided below.

Property	Description	
Wall Type	Specifies the type of conditions to apply to the wall boundary.	
	Stationary	Specifies the wall as a stationary impenetrable surface.
	Translating	Specifies the wall as a moving impenetrable surface. You specify the velocity for the translation.
	Rotating	Specifies the wall as a rotating impenetrable surface.
	Slip	Specifies the wall as a stationary, impenetrable, but friction-free surface.
Translating		
U (*U Component of Velocity Vector*)	Specify the velocity vector. You can change the units using the drop-down list.	
V (*U Component of Velocity Vector*)		
W (*U Component of Velocity Vector*)		

Property	Description
Rotating	
(*Rotational Velocity*)	Specifies the rotational velocity. You can change the units using the drop-down list.
X (*X Location of Axis Origin*)	Specify the coordinates of the origin for the axis of rotation.
Y (*Y Location of Axis Origin*)	
Z (*Z Location of Axis Origin*)	
(*i Component of Direction Vector*)	Specify the components of the axis of rotation
(*j Component of Direction Vector*)	
(*k Component of Direction Vector*)	
(**Set Axis Origin and Vector**)	Brings up the *Vector* dialog, where you can set the origin and direction vector of the axis of rotation using native NX functionality.

Custom Boundary

In STAR-CCM+, there are additional boundary types to those available in STAR-NX. The custom boundary in STAR-NX acts as a placeholder for a boundary that you will customize in STAR-CCM+.

Common Boundary Conditions

Some boundary conditions are not specific to any boundary type and appear in multiple boundaries, so long as the physics model that is associated with the condition is activated.

Common Boundary Conditions on Inlet and Pressure Outlet Boundaries

Temperature, turbulence, and passive scalar boundary conditions are common to all inlet boundaries as well as the Pressure Outlet boundary.

Property	Description
(*Temperature*)	Specifies the temperature on the boundary surface. You can change the units using the drop-down menu.
(*Turbulence Intensity*)	Specifies the turbulence intensity on the boundary surface. You can change the units using the drop-down menu.
(*Turbulent Length Scale*)	Specifies the turbulent length scale on the boundary surface. You can change the units using the drop-down menu.

Property	Description
(*Passive Scalar*)	Specifies the amount of passive scalar crossing the boundary. You can change the units using the drop-down menu.

Boundary Conditions on Wall Boundaries

When the energy equation is being solved, you can specify the thermal specification on wall boundaries.

Property	Description	
Adiabatic	Specifies that no transfer of energy occurs between the boundary surface and the surrounding environment.	
Convection	Specifies the transfer of energy between the boundary surface and the fluid moving over the boundary surface.	
	HTC (*Heat Transfer Coefficient*)	Specifies the heat transfer coefficient. You can change the units using the drop-down menu.
	(*Ambient Temperature*)	Specifies the ambient temperature. You can change the units using the drop-down menu.
Heat Flux	Specifies the rate of energy transfer through the boundary surface.	
	(*Heat Flux*)	Specifies the heat flux. You can change the units using the drop-down menu.
Temperature	Specifies the temperature on the boundary surface.	
	(*Temperature*)	Specifies the temperature on the boundary surface.

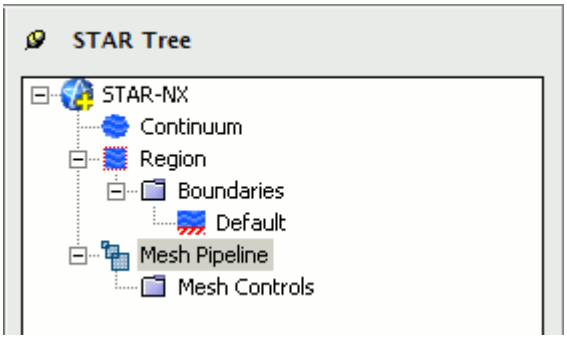
Meshing the Model

STAR-NX provides options for generating polyhedral or trimmed volume meshes, with or without near-wall prism layers.

By default, prism layers are generated on all wall surfaces. The provision of a prism layer depends on the type of boundary that is applied to a face. As a consequence, it is recommended that you only generate the mesh after setting up the boundaries and regions.

The meshing process first creates a triangulated surface (the surface mesh) and thereafter converts it to the desired volume mesh (polyhedral or trimmer). As the quality of the volume mesh depends to a large extent on the quality of the surface mesh, it is prudent to check the surface mesh before proceeding with the volume mesh.

Although the entire meshing process can be launched with a single command (the **Build And Run** command from the STAR-NX ribbon), it is possible to execute only single stages from the STAR-NXribbon. Settings that are made in the **Mesh Pipeline** node govern the mesh generation process.



A mesh pipeline contains all the settings that are applied to the surface and volume meshes created on the NX parts to which it is applied. In single part mode, only one mesh pipeline is available. In assembly mode, multiple mesh pipelines can be created if necessary.

When mesh generation commences, each mesh pipeline will:

1. Extract geometry from NX.
2. Transfer the model to STAR-CCM+.
3. Execute the surface remesher.
4. Execute the polyhedral or trimmer volume mesher with the prism layer mesher (if selected).
5. Create a mesh representation in the NX *Graphics* window.

Controls on the top-level mesh pipeline node allow you to set the base sizes for the surface mesh, and control the prism layer. Further refinement to specific faces, parts, or volume shapes can be made by adding additional controls to the mesh pipeline.

Contents:

Creation of Additional Surfaces in STAR-CCM+

Working with Mesh Pipelines
Working with Mesh Controls
Generating the Mesh STAR-NX
Common Causes for Meshing Failures

Creation of Additional Surfaces in STAR-CCM+

STAR-CCM+ creates geometry part surfaces during the geometry transfer stage.

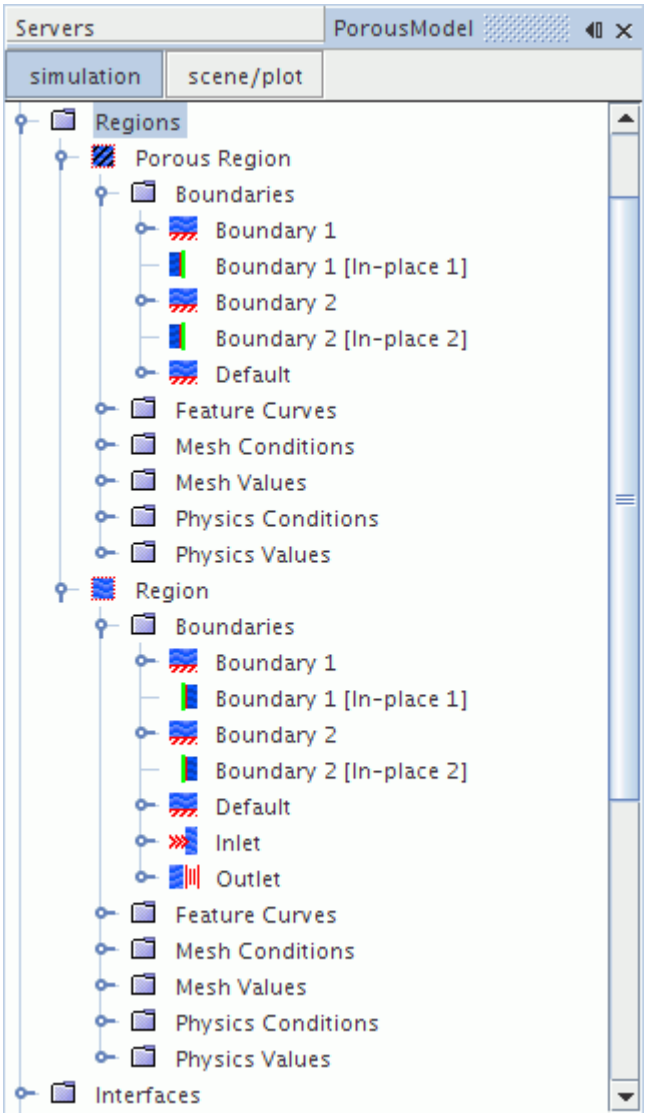
Initially, STAR-NX creates one part surface for each face in the CAD model. These part surfaces are then combined so that one surface per boundary remains. This outcome, however, is not possible for boundaries that contain CAD surfaces from more than one part. The final part surfaces appear in the **Geometry** > **Parts** node in STAR-CCM+.

The following items affect the number of part surfaces that are created in STAR-CCM+:

- **The number of boundaries present in the model, including the default boundary.** Each boundary in STAR-NX appears as a separate part surface in STAR-CCM+. If, however, a boundary contains CAD surfaces from multiple parts, the part surfaces in STAR-CCM+ cannot be combined.
- **The number of mesh pipeline controls used (if any).** Each CAD surface that is used to define a face mesh control in STAR-NX appears as a separate part surface in STAR-CCM+.
- **The number of interfaces that are created during the meshing process (for assemblies only).** All the CAD surfaces that define one interface between two parts in an assembly are grouped together under one part surface in STAR-CCM+. One part surface exists for each interface in STAR-CCM+.

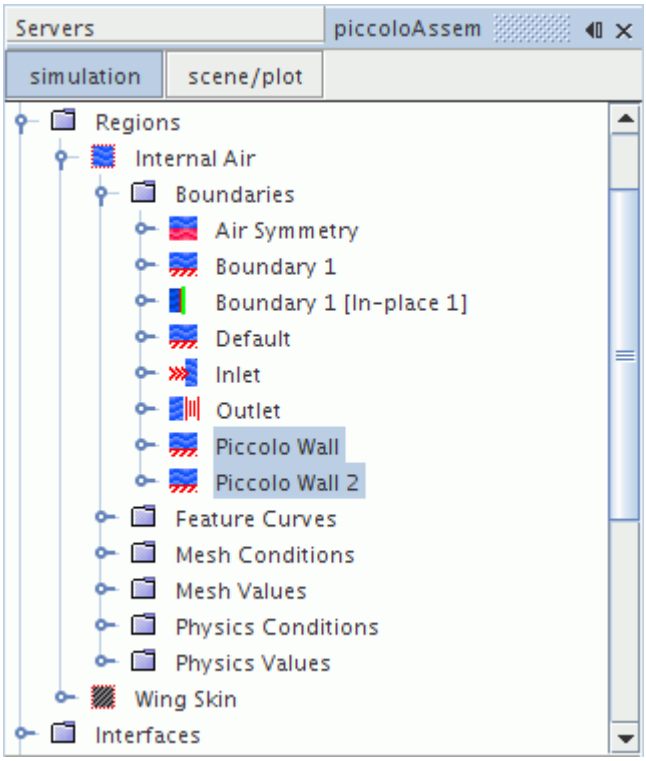
Boundary Splitting due to Interface Generation

Interfaces are automatically created between two parts in an assembly during the mesh generation. A new boundary is created for each interface on the surfaces that share the interface.



Boundary Splitting due to Face Mesh Controls and Part Mesh Controls

If you use a face mesh control or a part mesh control in your simulation, some boundaries may be split, depending on what faces are included in the mesh control. If a boundary and mesh control share the same faces, no splitting occurs. If a mesh control contains faces from multiple boundaries, splitting occurs and a new boundary is created. The name of the new boundary is derived from the original boundary.



Working with Mesh Pipelines

When working with a single part, you will only have one mesh pipeline defined in the model tree. This will specify the mesh settings for the single part available. In assembly mode, multiple mesh pipelines can be created to give specific controls for different parts, and parts must be specifically associated with a mesh pipeline.

Contents:

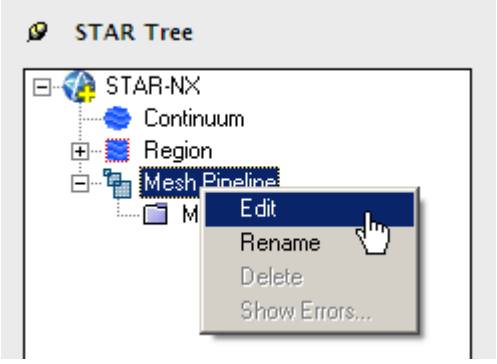
Setting Up the Mesh Pipeline

Setting Up the Mesh Pipeline

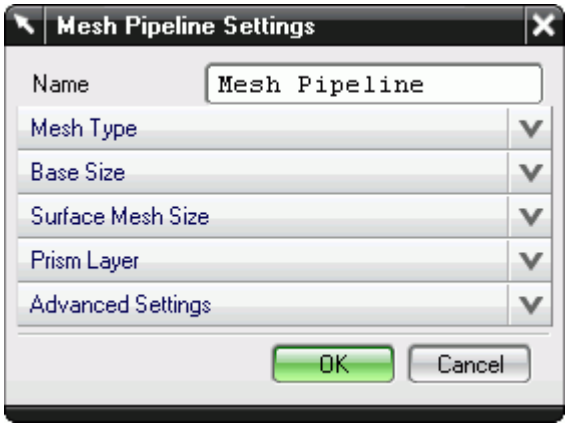
The mesh pipeline node contains the general mesh settings that will apply to all STAR-NX parts associated within the pipeline in the absence of any further controls.

To set up the mesh pipeline:

1. Right-click the **Mesh Pipeline** node in the STAR-NX tree and select **Edit**.



The *Mesh Pipeline Settings* dialog appears.



2. In the *Name* field, enter the name of the mesh pipeline if required.
3. In the *Part Selection* box, specify the desired parts to use in the mesh pipeline.

   This property is only available when working with assemblies.
4. From the *Mesh Type* tab, select the type of mesh that you want to generate:
   - **Polyhedral**
   - **Trimmer**
5. Set *Base Size* to the desired value.

   The base size is a reference length used in other meshing parameters. It may be set to equal the length of the largest triangle you expect to have on your surface mesh, or to some other characteristic dimension.

   The base size can be set as either a relative or absolute value. When it is set as a relative value, the reference size will be the diagonal of the bounding box enclosing the entire geometry. In the case of multiple parts in an assembly, the reference size will be calculated based on the diagonal length of the bounding box enclosing the entire assembly. This includes all parts from all mesh pipelines. The unit specification menu will become active when setting an absolute value.
6. Set *Surface Mesh Size* to the desired value.

   The surface mesh size is the base edge length of triangles on the surface mesh. Note that the ultimate length of some (or all) triangles on the surface mesh will depend on other factors such as surface curvature and edge proximity.

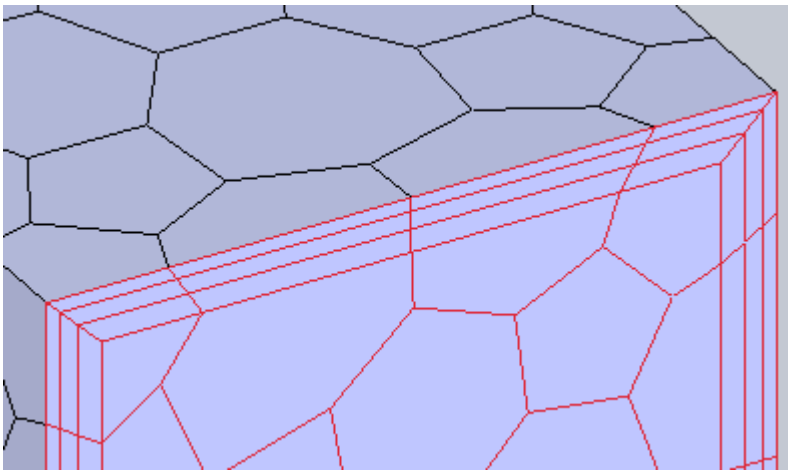
7. Set the prism layer settings. See Prism Layer Panel.
8. Set the advanced settings. See Mesh Pipeline Advanced Settings Panel.
9. If you selected to generate a trimmed mesh, set the Trimmer settings. See Trimmer Panel.
10. Click **OK**.

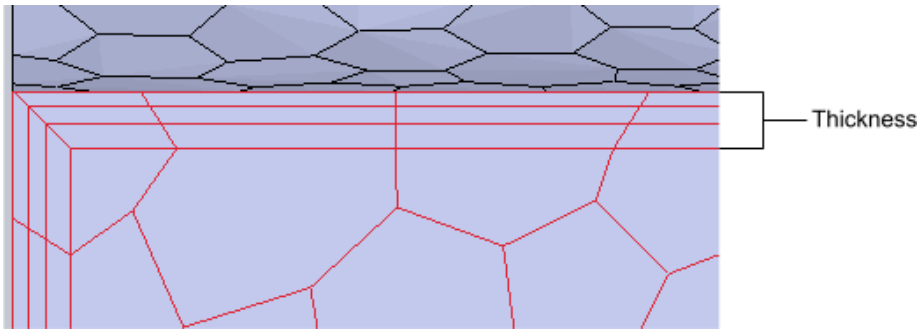
Contents:

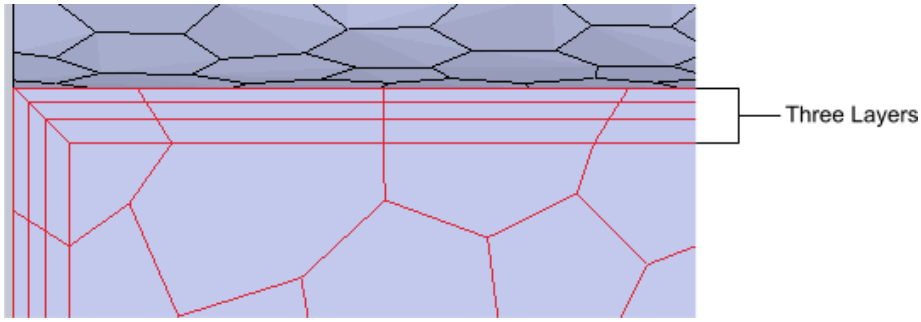
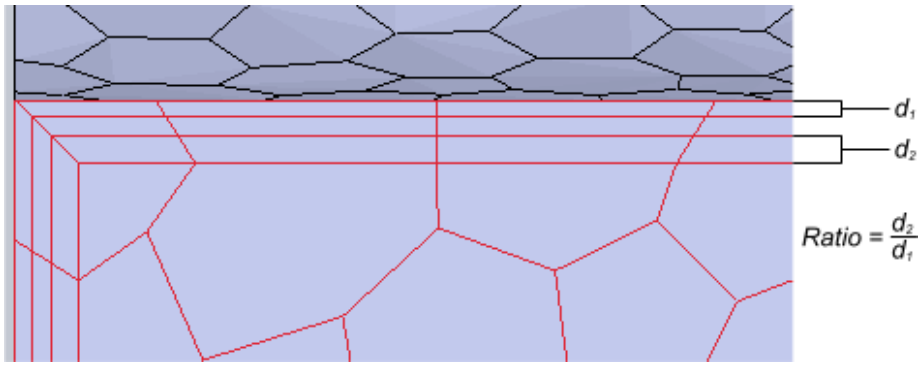
Prism Layer Panel
Mesh Pipeline Advanced Settings Panel
Trimmer Panel

Prism Layer Panel

Prims layers improve the simulation near non-slip walls, particularly when a turbulence model is active.



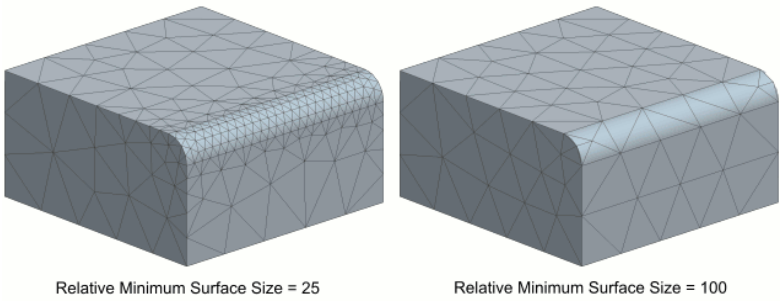
Property	Description
Layer Thickness	Defines the overall depth of the prism layer, which includes all of the individual layers, as illustrated below. 
Number of Layers	Defines how many individual cell layers will be included in the overall prism layer, as illustrated below.

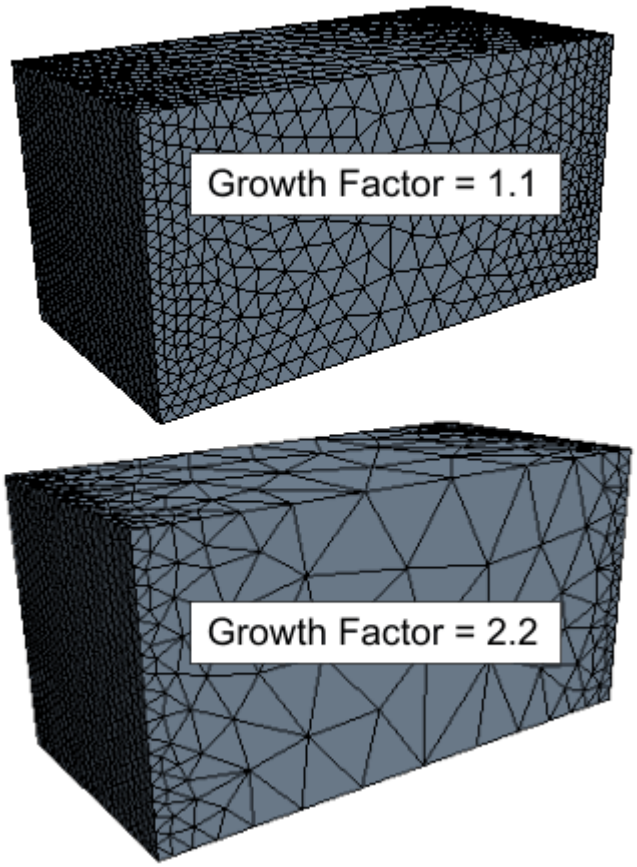
Property	Description
	
Thickness Ratio	Specifies the size of the first and last layers within the prism layer. 

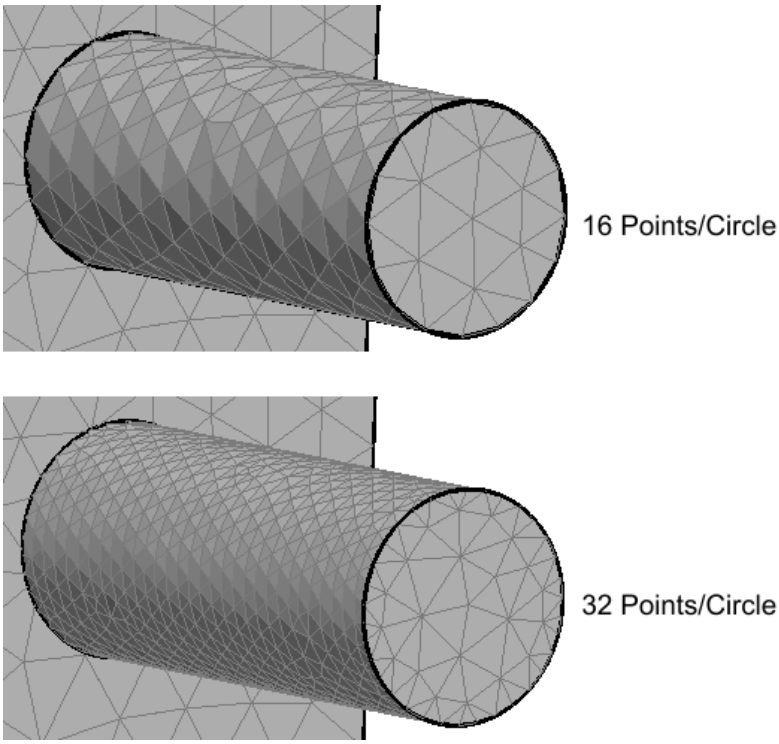
Mesh Pipeline Advanced Settings Panel

The advanced settings panel provides further customization of meshing parameters.

Property	Description
Relative Minimum Surface Size	Defines the minimum triangle size permitted on a feature curve. The minimum triangle size is set as a percentage of the surface mesh size. The default value is set to 25%. You must set the relative minimum surface size to a value greater than zero. If you set the relative minimum surface size to 100%, STAR-CCM+ will not modify the triangle size when it detects curvature. The figure below illustrates this behavior. Note that the base size was set to 20% in this example to emphasize the refinement on the curve.

Property	Description
	 Having a large value for the number of points on a circle property increases the mesh resolution on curved surfaces. The transition between the fine mesh and coarse mesh areas is controlled by the growth ratio.
Growth Ratio	Controls the rate at which triangle sizes change when moving from one triangle size to another. In the example below, the end faces of the block were set to have a smaller triangle size than the four side faces of the block. When the growth ratio is set to a low value, the transition between triangle size is very gradual. When a higher value is used, the transition is far more rapid.

Property	Description
	 The growth factor must be set to a value greater than 1.0. There is no upper limit for the growth factor.
Number of Points per Circle	Controls the number of triangle edges that appear on a curved surface, as shown below.

Property	Description
	 Regardless of the number of points per circle defined, STAR-NX will not permit the triangle edge length to shrink below 25% of the base size. The curvature length scale may also affect the number of edges on a curved surface.

Trimmer Panel

The trimmer mesh is controlled using the Trimmer panel.

Property	Description
Default Growth Rate	Controls the stepping of the refinement from one cell size to the next. Five settings are available. By default, the *Default Growth Rate* is set to **Fast**, meaning that the fastest transition from the smallest cell size to the largest will be used. This usually equates to having only one layer of each cell size for each transition, which will result in the least number of cells being created for the supplied input values. The **Medium**, **Slow**, and **Very Slow** options further slow the transition from one cell size to the next in the mesh interior and therefore increase the overall cell count. In general, the medium option results in a minimum of two equal sized cell layers per transition, slow in four layers and the very slow option in eight layers.
Boundary Growth Rate	Controls the number of cell layers to grow next to surfaces in the template. Five settings are available. By default, the *Boundary Growth Rate* is set to **None**, meaning that the setting used for the Default Growth Rate is used

Property	Description
	for controlling all transitions. If the **Very Slow**, **Slow**, **Medium**, or **Fast** options are selected then it will result in eight, four, two or one layers of cells respectively of the same size being created next to the geometry surfaces in the template.

Working with Mesh Controls

Mesh controls allow you to define custom mesh settings for a particular section of the model.

Multiple controls can be assigned to the same item to control different mesh parameters. If, for example, a face has two mesh controls assigned to it--one that controls the mesh size and another that controls the prism layer thickness, the final mesh will respect both the custom mesh size and the custom prism layer thickness. If two mesh controls set the same mesh parameter, the control created last will take precedence over the first.

Contents:

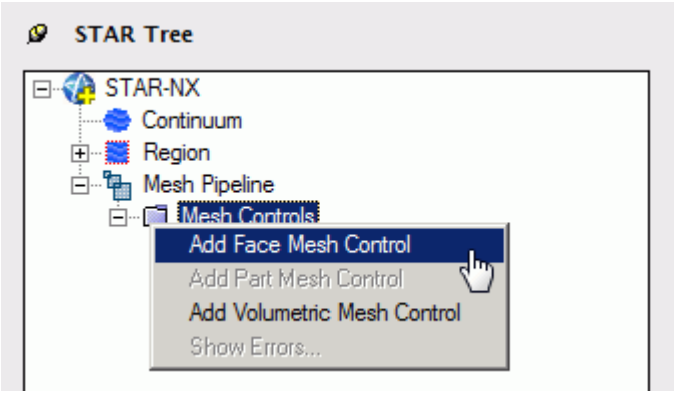
Using a Face Mesh Control
Using a Volumetric Mesh Control
Using a Part Mesh Control

Using a Face Mesh Control

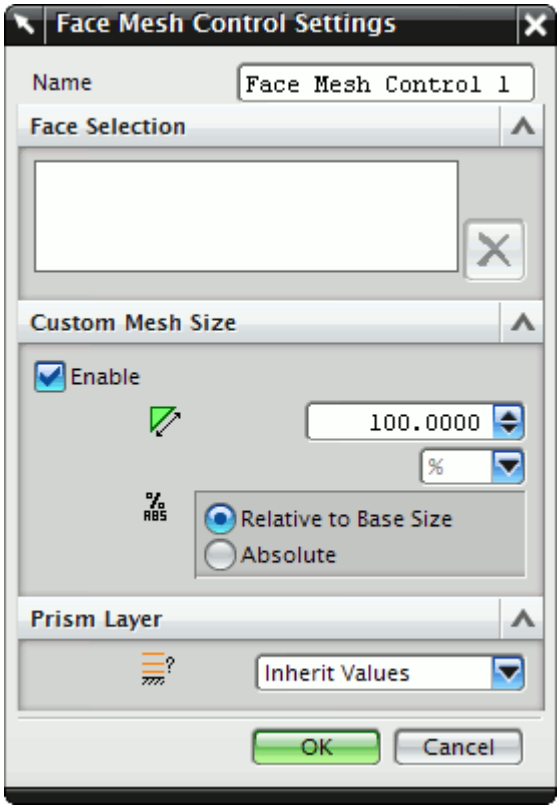
A face mesh control allows you to set the surface mesh size and the prism layer parameters on a face. Custom settings made on a face control will take priority over settings in the parent mesh pipeline, and over settings made in any part mesh controls.

To add a face mesh control:

1. Right-click the **Mesh Pipeline** > **Mesh Controls** node in the STAR-NX tree and select **Add Face Mesh Control**.



The *Face Mesh Control Settings* dialog appears.



2. Set *Name* to an appropriate value.
3. On the CAD geometry, select the desired faces to include in this mesh control.
4. If you want to specify a custom surface mesh size on the selected surfaces, activate the **Custom Mesh Size** option.

   a) Set (*Mesh Size*) to the desired value.

   You can enter this value as a percentage of the base size, or as an absolute value. If you select to enter the absolute value, you can select the units from the drop-down menu.

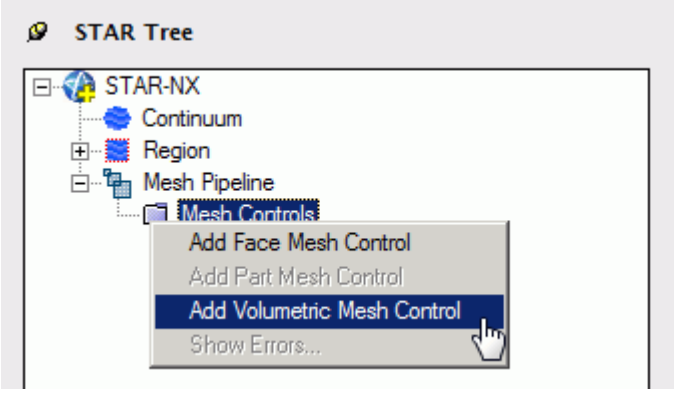
5. Set (*Prism Layer Option*) to the desired value:
   - **Inherit Values**: in the first instance, the face(s) will be assigned values from any part mesh control applied to the part to which the face(s) belong. If this does not apply, values will be taken from any applicable volumetric mesh control, and in the absence of this values will be taken from the parent mesh pipeline.
   - **Use Custom Values**: allows the prism layer parameters to be set directly on the face(s). See Prism Layer Panel.
   - **Disable**: no prism layer will be generated on the face(s).
6. Click **OK**.

Using a Volumetric Mesh Control

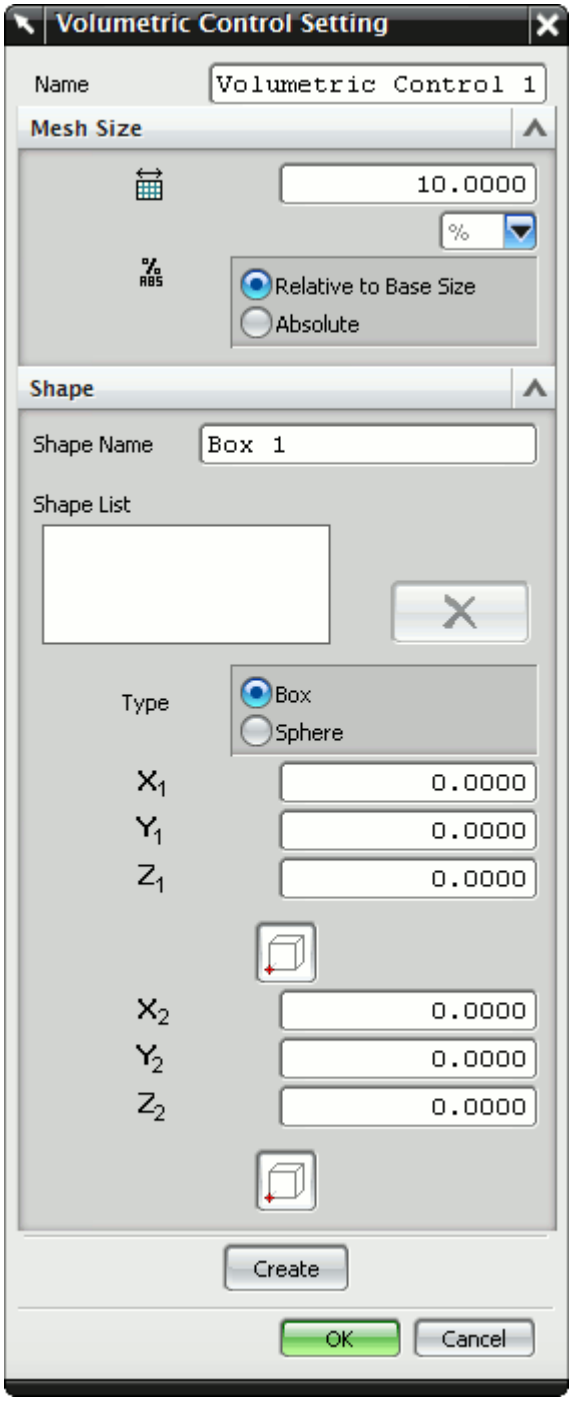
A volumetric control permits mesh refinement within a user-defined box or sphere. Only the mesh size can be modified with a volumetric control; prism layer parameters are not affected.

To add a volumetric mesh control:

1. Right-click the **Mesh Pipeline** > **Mesh Controls** node in the STAR-NX tree and select **New Volumetric Mesh Control**.



The *Volumetric Control Settings* dialog appears.



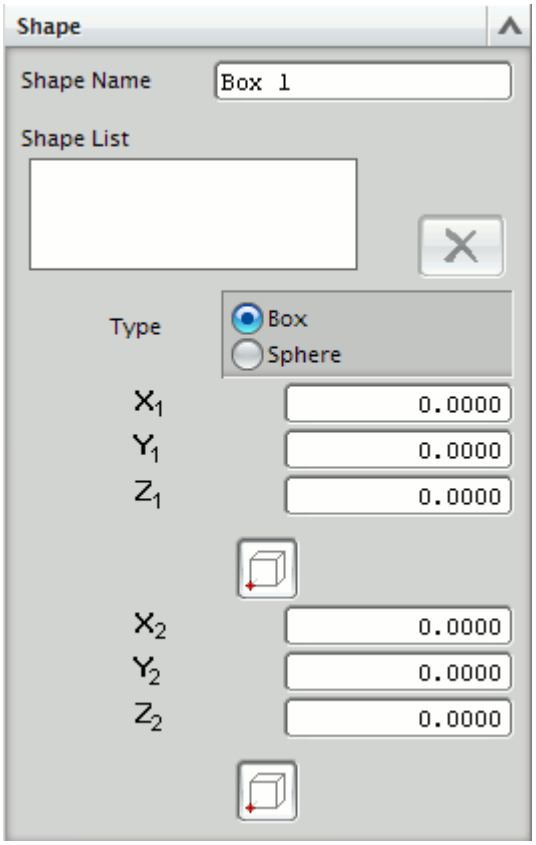
2. Set *Name* to an appropriate value.

3. Set (*Mesh Size*) to the desired value.

   The mesh size defines the triangle edge length for surface mesh within the volume shape, and the characteristic cell size for volume mesh. The custom mesh size will still affect the volume mesh in cases where the volume shape does not intersect any surfaces of the Pro/E geometry.

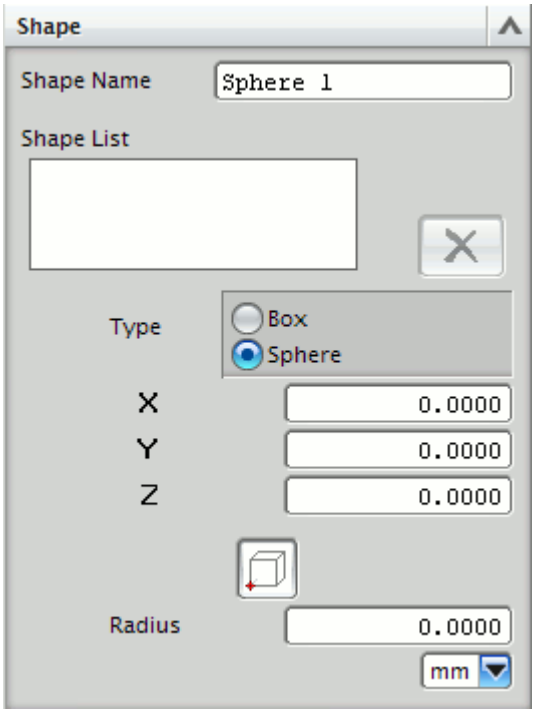
   The mesh size can either be defined relative to the base size of the parent mesh pipeline, or as an absolute dimension with length units.

4. Select the type of volumetric control to create:
   - **Box**
   - **Sphere**
5. Enter the parameters for the volumetric shape:
   - If you are creating a box, enter the coordinates for Point 1 and Point 2.



X, Y, and Z coordinates of the two opposite corners of the box are required by the panel. These coordinates can either be entered manually, or taken from existing points within the Pro/E window. To use an existing point, click (**Set Point**) and use the NX point selection operation to select the appropriate point.

   - If you are creating a sphere, enter the coordinates for the center point and the radius of the sphere.



The sphere shape is defined by a center point and a radius. The center point can be set either by entering the coordinates directly, or by clicking the (**Set Point**) button and using the Pro/E point selection procedure to choose an existing point in the Pro/E window. The coordinates of the existing point will be used to populate the panel.

6. Click **Create**.

   The shape is added to the *Shape List* list.

7. Click **OK**.

Editing a Volume Shape

To edit a volume shape

1. Select the volume shape from the list within the **Shapes** panel.
2. Edit the parameters as required and click **Update**.

Deleting a Volume Shape

To delete a volume shape

- Select it from the list in the *Shape* panel and click (**Delete selected shape**).

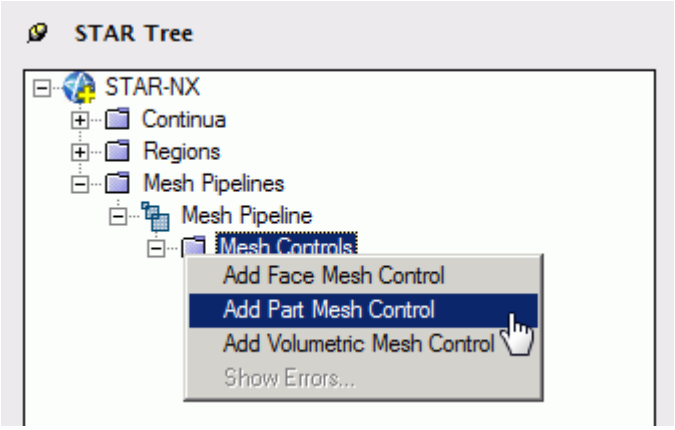
Using a Part Mesh Control

Custom mesh settings (such as surface mesh size and prism layer parameters) can be applied to one or more parts using a part mesh control.

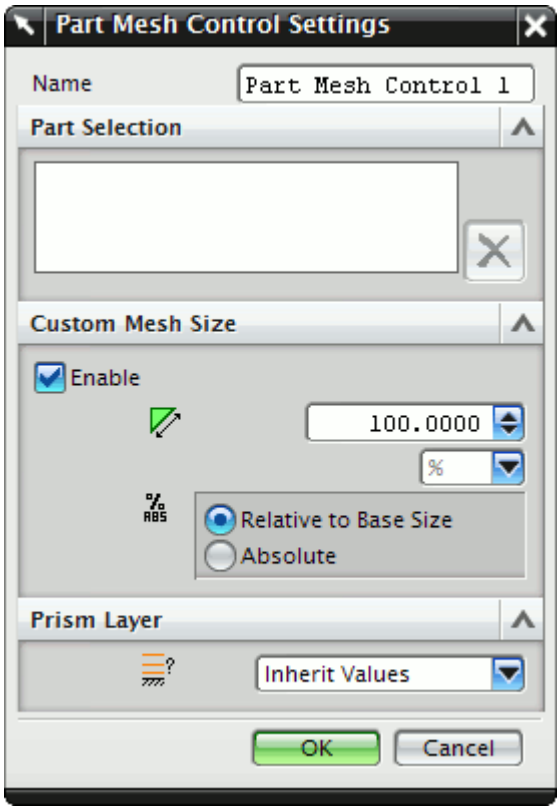
Settings made on a part mesh control will take priority over settings made on the parent mesh pipeline, but will not have priority over settings made on a face mesh control.

To add a part mesh control:

1. Right-click on **Mesh Pipelines** > **Mesh Pipeline** > **Mesh Controls** in the STAR-NX tree and select **Add Part Mesh Control**.



The *Part Mesh Control Settings* dialog appears.



2. Set *Name* to an appropriate value.
3. On the CAD geometry, select the desired parts to include in this mesh control.

4. If you want to specify a custom surface mesh size on the selected surfaces, activate the **Custom Mesh Size** option.

   a) Set (*Mesh Size*) to the desired value.

   You can enter this value as a percentage of the base size, or as an absolute value. If you select to enter the absolute value, you can select the units from the drop-down menu.

5. Set (*Prism Layer Option*) to the desired value:
   - **Inherit Values**: for this option, prism layers on the surfaces of the part(s) will be generated based on settings taken from the parent mesh pipeline.
   - **Use Custom Values**: allows the prism layer parameters to be set directly on the part(s). See Prism Layer Panel.
   - **Disable**: no prism layer will be generated from the surfaces of the selected parts.
6. Click **OK**.

Generating the Mesh STAR-NX

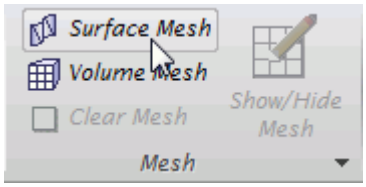
It is recommended that you first generate the surface mesh and inspect its quality before generating the volume mesh. Problems in the surface mesh will have a definite impact on the volume mesh, and may result in the volume mesher failing to complete.

Mesh generation will be carried out automatically when you issue the Build and Run command.

Generating the Surface Mesh in STAR-NX

To generate the surface mesh:

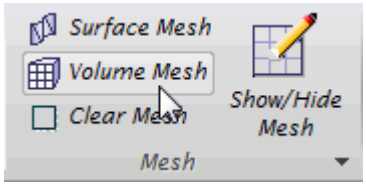
- From the STAR-NX ribbon, click **Surface Mesh**.



Generating the Volume Mesh in STAR-NX

To generate the volume mesh:

- From the STAR-NX ribbon, click **Volume Mesh**.

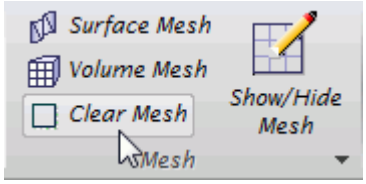


Clearing Generated Meshes

The **Clear Generated Meshes** action deletes all generated meshes without affecting the settings.

To clear the mesh:

- From the STAR-NX ribbon, click **Volume Mesh**.



Contents:

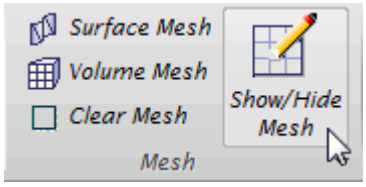
Visualizing the Mesh
Meshing Behavior at Interfaces

Visualizing the Mesh

Surface and volume meshes will automatically be displayed on first creation. If required, they can be hidden from view.

To hide a generated mesh in the geometry view:

1. From STAR-NX the ribbon, click **Show/Hide Mesh**.



Meshing Behavior at Interfaces

The volume mesher will treat interfaces between two regions differently depending on what type of region (fluid or solid) these are.

If the interface lies between two fluid regions, no prism layers will be grown from both sides of the interface. If an interface lies between a solid region and a fluid region, prism layers will be grown from the interface. The default behavior of the prism layer generation can be overridden using a mesh control, so prism layers can be disabled on a solid-fluid interface. See Working With Mesh Controls.

See Also:

Working with Mesh Controls

Common Causes for Meshing Failures

There may be instances when the mesh generation fails. In most cases this will occur during the surface meshing phase. STAR-NX will try and identify the cause and suggest ways of rectifying this. This information will be printed in the *Output* tab of the STAR-NX Info App window.

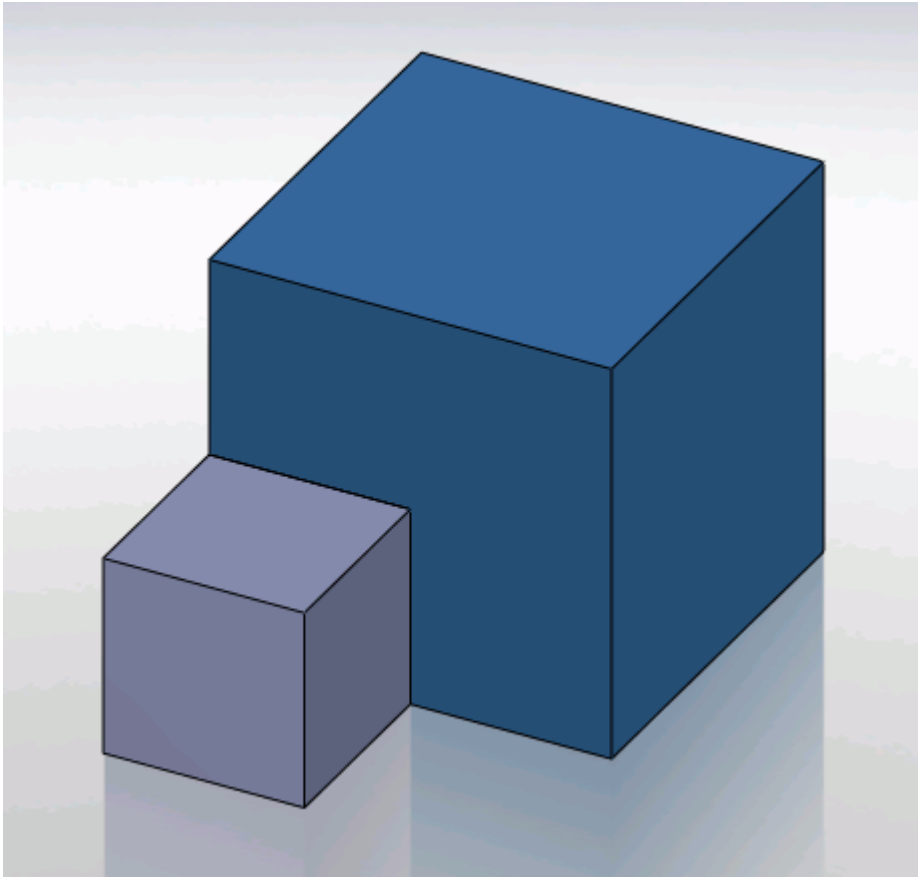
Contents:

The Parts in an Assembly are not Imprinted

The Parts in an Assembly are not Imprinted

The parts in an assembly should be imprinted together so that there is a common contact area between them.

A simple example of when imprinting is required is shown below.

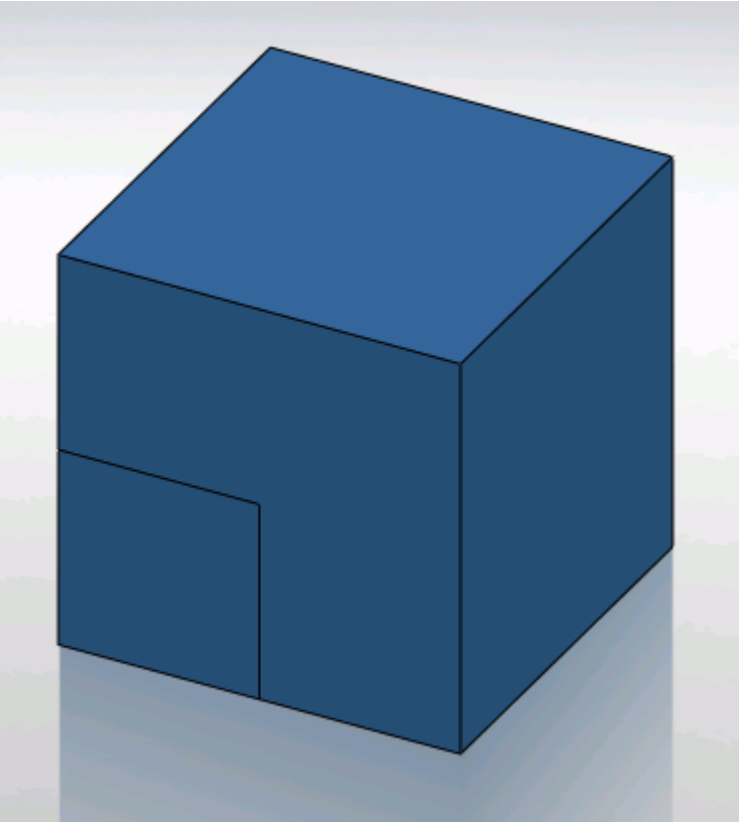


In this case the face of Part A (grey box) is splitting the face of Part B (blue box) causing this assembly to fail to mesh.

The output will display an error message.

```
Initializing meshing pipeline...
  All Geometry up to date.
Executing surface meshers...
Preparing surface for surface meshing in regions: Region 1, Region
Error during surface remeshing, deleting resulting mesh
An error occurred while attempting to orient the surface.
This is usually due to self-intersections in the surface. Check for problem(s) at :
  Geometric Range : (-0.0225836 0.00152109 0.0632873) and (0.0501103 0.0703362 0.0647127)
  Centroid of face : (0.0256414 0.0471603 0.064)
  Face type of face : 6
  Boundary = Default.
  Region = Region 1
Command: GenerateSurfaceMesh
   CompletedCommand: GenerateSurfaceMesh
   In: [Machine::main]
   Slave Message:
      Slave Name: Proc[0]
      Slave Rank: 0
   error: Server Error
```

The surface of Part B has to be imprinted with the contact face of Part A as shown below, after which meshing will complete successfully.



Obtaining a Solution

The following section details the execution of the STAR-CCM+ solvers in order to obtain a solution.

Contents:

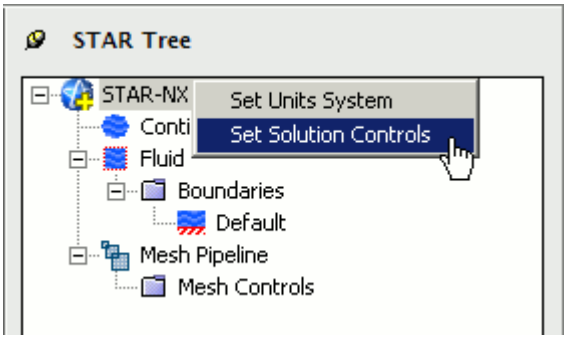
Setting Solution Controls
Checking the Model for Errors
Building and Running the Simulation in STAR-NX
Clearing the Solution
Visualizing the Solution

Setting Solution Controls

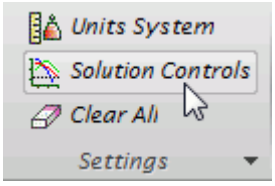
The solution you obtain from the solvers is a compromise between accuracy and processing time. You can manage the speed and accuracy of the solver from the *Solution Controls* dialog. The solution accuracy is also dependant on the mesh density.

To set up the solution controls in STAR-NX:

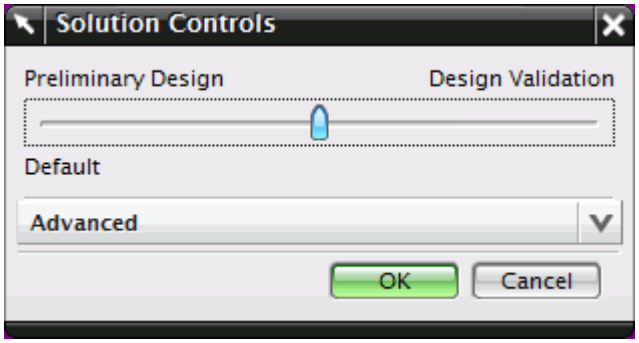
1. Either:
   - Right-click on the **STAR-NX** node within the object tree and select **Set Solution Controls**.



   - From the STAR-NX ribbon, select **Solution Controls**.



   The *Solution Controls* dialog appears.



2. Set the solution control parameters:
   - Using the slider, select one of the five pre-defined solver options. These options range from high accuracy and low speed, to low accuracy and high speed.
   - Click to expand the *Advanced Settings* tab and set the solution control parameters manually. See Solution Controls Advanced Settings Panel.
3. If you want to run a Java macro with the simulation, activate the **Use Java File** option. See Using Java Macros with STAR-NX.
4. Click **OK**.

Contents:

Solution Controls Advanced Settings Panel
Using Java Macros in STAR-NX

Solution Controls Advanced Settings Panel

The advanced settings in the Solution Control dialog allow you to manually enter the values for the solution control parameters.

Parameter	Description
Stopping Criteria	
(*Number of Iterations*)	Specifies the maximum number of iterations for which the solver can run. When the STAR-CCM+ solver reaches the specified number of iterations, it will stop. You can then post-process the results in STAR-CCM+. See Convergence Levels.
(*Convergence Level*)	Specifies how many orders of magnitude the residuals should drop before the solver stops. See Convergence Levels.
Solution Accuracy	
First Order	This option is generally the quicker, but less accurate solver option. The first order scheme introduces a dissipative error that is stabilizing and helps the

Parameter	Description
	solver achieve robust convergence. However, it can also lead to the neglect and merging of discontinuities, meaning the solution is not fully captured.
Second Order	This option is generally the more accurate, but slower solver option. The second order scheme attempts to minimize both dispersive and dissipative errors, providing a better representation of the simulation. However, in some situations, the lack of numerical dissipation can result in poorer convergence properties. It is common practice to use the first order scheme initially to ensure the simulation is running as expected and then switch to second order.
Under-Relaxation Factors	
(*Momentum Under-Relaxation Factor*)	Specifies the under-relaxation factor for the momentum equation. See Solution Controls Advanced Settings Panel.
(*Pressure Under-Relaxation Factor*)	Specifies the under-relaxation factor for the pressure-correction equation. See Solution Controls Advanced Settings Panel.
(*Turbulence Under-Relaxation Factor*)	Specifies the under-relaxation factor for the turbulence equation. See Solution Controls Advanced Settings Panel.
(*Fluid Temperature Under-Relaxation Factor*)	Specifies the fluid under-relaxation factor for the segregated energy solver. See Solution Controls Advanced Settings Panel.
(*Solid Temperature Under-Relaxation Factor*)	Specifies the solid under-relaxation factor for the segregated energy solver. See Solution Controls Advanced Settings Panel.

Convergence Level

The numerical method used by the solver requires an iterative process in order to obtain a solution. After each iteration, residuals are produced that indicate how well the governing equations for each solver quantity are being satisfied numerically. In general, the residual magnitude should decrease as the solution converges. When the magnitude of the residuals for all the quantities falls below the convergence level, the solver will stop iterating, and results will be exported for post-processing.

Note: In most cases, a drop of 3-5 orders of magnitude in the residuals is sufficient to obtain an accurate solution.

You can adjust the convergence level parameter to meet accuracy or process time requirements. However, if the residuals are no longer decreasing from one iteration to the next, it is likely that machine accuracy has

been reached for that case. For a case that does not exhibit decreasing residuals, it would be necessary to reconsider the setup in case the volume mesh is too coarse, or boundary conditions have been set incorrectly.

Under-Relaxation Parameters

Under-relaxation parameters determine how quickly the solver reaches a solution for a particular equation, on a scale from 0 to 1, with 1 being the fastest.

Ideally, all under-relaxation parameters would be set to 1, however, doing so would likely cause solver instability due to the complex nature of most simulations. Hence the under-relaxation parameters should be set to values that compromise speed with stability.

Note: Adjust the under-relaxation parameters if changes to other solution controls such as *Number of Iterations*, *Convergence Levels*, and *Solution Accuracy* have failed to improve the behavior of the solver.

Using Java Macros in STAR-NX

You can use Java macros with STAR-CCM+ to automate your simulation or to change a setting in STAR-CCM+, such as selecting an alternative turbulence model, before the solver starts running.

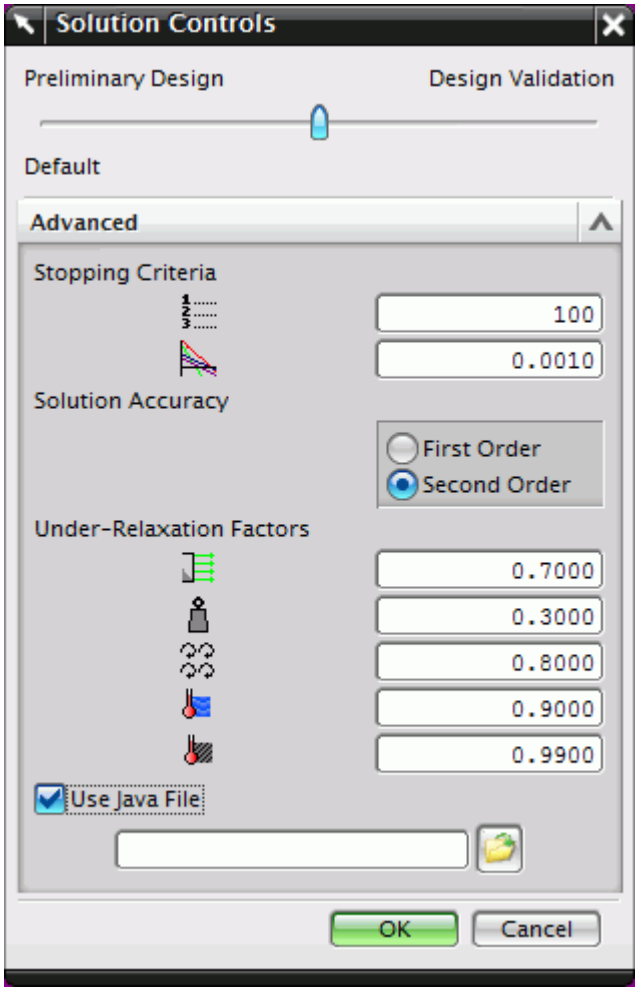
An example of how to change a setting in STAR-CCM+ using a Java macro is outlined below:

1. Launch STAR-NX and set up the case.

   Define the continuum settings, boundary conditions, mesh settings, and solution control settings.
2. Launch STAR-CCM+.
3. In STAR-CCM+, start recording a macro and perform the desired operations, such as changing the physics models.
4. Stop recording the macro.

   The macro saves at this point.
5. Close STAR-CCM+ and do not save the simulation.
6. Return to STAR-NX, open the *Solution Controls* dialog and activate the *Use Java File* checkbox.

   A text entry field appears.



7. Click (**Select Input File**).

   The *Open* dialog appears.

8. Navigate to your Java macro file and select it.

   Selected files appear in the text entry field. The path in the entry field is displayed relative to the current CAD geometry file.

9. Click **OK**.

When you run the simulation, STAR-CCM+ will launch and play the macro before starting the solver. This sequence allows you to set up and use additional STAR-CCM+ physics models that are not available in STAR-NX.

See Also:

Setting Up Continua

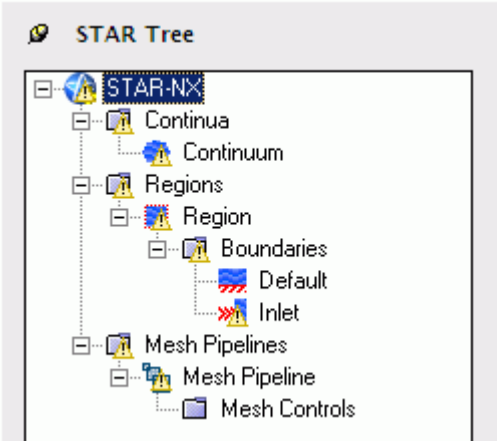
Setting Up Regions

Setting Boundary Conditions

Setting Up the Mesh Pipeline

Checking the Model for Errors

STAR-NX monitors the active case for any errors in real-time and displays a warning triangle ⚠ next to any nodes for which there is a problem.

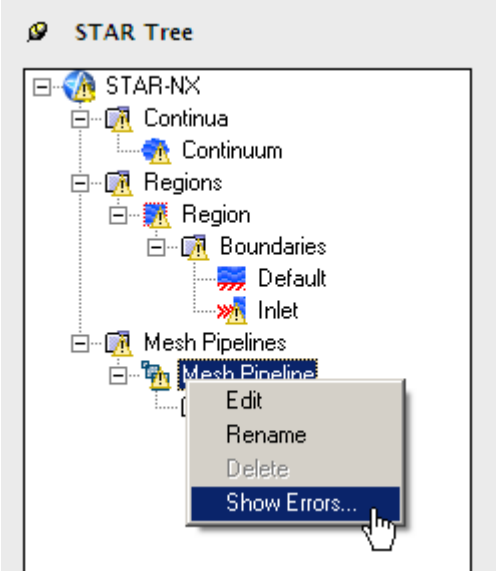


Three areas are monitored by the live model checker:

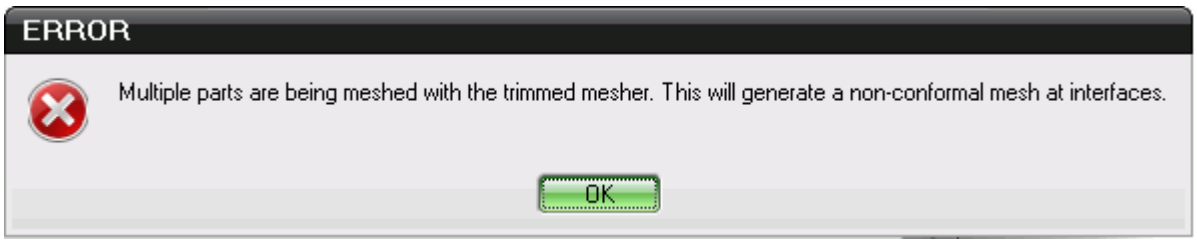
- **Continua**. The continua checker makes sure that all necessary boundaries are present in the case, such as the presence of at least one inlet and one outlet.
- **Regions and boundaries**. The regions and boundaries checker covers issues specific to boundaries, such as a velocity inlet boundary having a zero value.
- **Mesh pipelines**. The mesh pipeline checker covers issues that are related to the mesh, such as the presence of multiple parts in a trimmed mesh pipeline.

To see the cause of an error:

1. Right-click on an node displaying the warning triangle, and select **Show Errors...**



A dialog appears describing the error, as shown below.



2. Click **OK** to close the dialog.

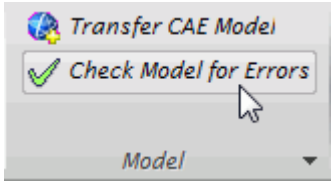
The **Show Errors...** right-click menu is available for any node in the STAR-NX object tree that contains a warning triangle. Selecting it will show errors for that node and its sub-nodes.

To see a complete list of issues for the case, right-click on the STAR-NX root node and select **Show Errors...**

You can also manually check for errors.

To perform a manual check:

- From the STAR-NX ribbon, click **Check Model for Errors**.



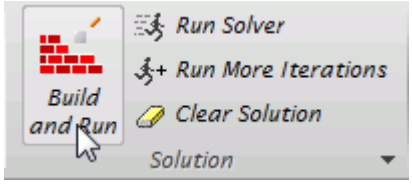
If errors are found in the case set up, these are displayed in the *Model Checker* tab of the *Info App* dialog. See The Model Checker Tab.

Building and Running the Simulation in STAR-NX

Once you have defined the model physics, boundary conditions, mesh settings, and solver settings in STAR-NX, you can build and run the case.

To build and run the case:

- From the STAR-NX ribbon, click **Build and Run**.



This action automatically generates the mesh, and runs the solver. When you run the solver, a new *Residuals* tab is added to the *Info App* dialog. You can use this residuals plot to monitor solution convergence. When the solver finishes, STAR-CCM+ launches and a scalar scene is automatically created for you.

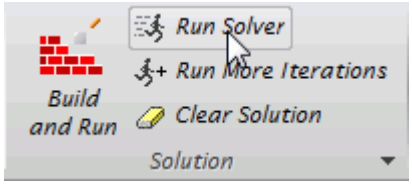
Note: If a solution has already been computed in STAR-CCM+, and the stopping criteria has been reached, clicking **Build and Run** clears the solution and starts running from the first iteration. If you want to run for more iterations, change the stopping criteria and click **Run Solver**, or click **Run More Iterations**.

Running the Simulation in STAR-NX

If you have already generated a volume mesh for the model, you can run the case directly.

To run the solver:

- From the STAR-NX ribbon, click **Run Solver**.

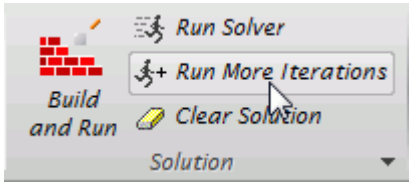


Running for More Iterations

You may need to run the STAR-CCM+ solvers for more iterations than you originally set in the Number of Iterations stopping criterion. The Run More Iteration function runs the solver for an additional number of iterations as specified in the stopping criteria. Take, for example, a case where you have already run the simulation for 300 iterations. You specify 500 iterations in the *Number of Iterations* stopping criterion. When you next click **Run for More Iterations**, the simulation will run for 500 extra iterations, making 800 iterations in total. The solver runs until one of the stopping criteria is satisfied.

To run for more iterations:

- From the STAR-NX ribbon, click **Run For More Iterations**.



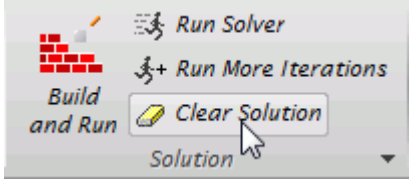
Clearing the Solution

You can clear all results that were calculated by the STAR-CCM+ solvers using the **Clear Solution** function. All other STAR-NX simulation settings remain intact.

Caution: This action is irreversible.

To clear the solution:

1. From the STAR-NX ribbon, select **Clear Solution**.



The solution that was calculated by the STAR-CCM+ solvers is deleted.

Visualizing the Solution

When STAR-CCM+ launches after the solution is calculated, it automatically displays a scalar scene of the model that shows the `Absolute Pressure` field function.

To visualize the results, you can use this scene to display different scalar quantities, or use any of the other visualization tools in STAR-CCM+, such as streamlines, plane sections and vector scenes.

You can import CAD parts into STAR-CCM+ for visualization purposes. These parts will not be taken into account during the analysis. This feature is useful for displaying the original geometry in a post-processing scene in STAR-CCM+. See Importing Geometry Into STAR-CCM+ for Visualization.

Exporting Files and Data

This section details the process for exporting files and data from STAR-NX.

Contents:

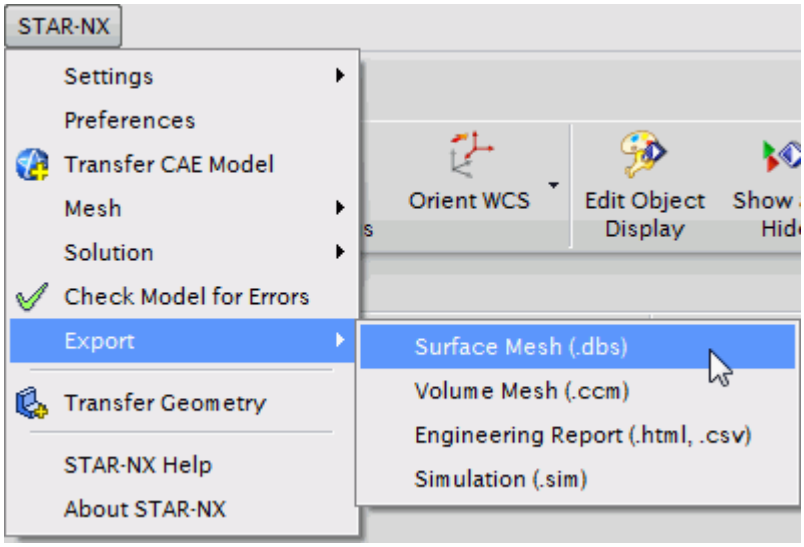
Exporting the Surface Mesh
Exporting the Volume Mesh
Exporting an Engineering Report
Exporting a STAR-CCM+ Simulation

Exporting the Surface Mesh

You can export the surface mesh generated in STAR-NX as a native CD-adapco surface mesh (`.dbs`) file. The `.dbs` file format is the recommended format to use when transferring surface data between CD-adapco programs. The boundary names and boundary types are retained by the file, as well as the interfaces that were automatically created between regions.

To export the surface mesh:

1. From the STAR-NX ribbon, click **Export > Surface Mesh (.dbs)**.

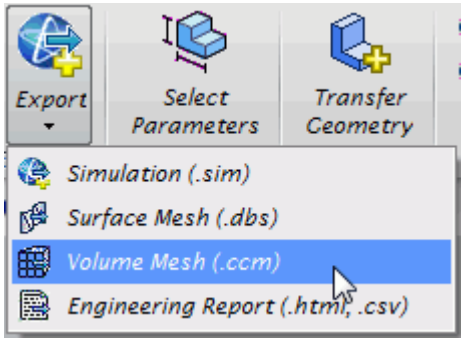


Exporting the Volume Mesh

You can export the volume mesh that you generated in STAR-NX as a native CD-adapco volume mesh (`.ccm`) file. The `.ccm` file format is the recommended format to use when transferring volume meshes between CD-adapco programs. The region and boundary names, and region and boundary types are retained in this file. The interfaces that were automatically created interfaces between regions are also retained in this file.

To export the volume mesh:

1. From the STAR-NX ribbon, click **Export** > **Volume Mesh (.ccm)**.



Exporting an Engineering Report

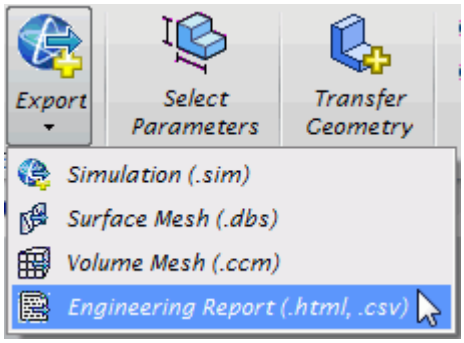
You can export a summary of the simulation settings and results in HTML format to two individual files:

- The first will be given a user-specified file name, and contains results from the simulation.
- The second will be given a user-specified file name appended with `_caseinfo`, and contains details on the simulation set-up.

Note: You can only export an engineering report after you obtain a solution.

To export an engineering report:

1. From the STAR-NX ribbon, click **Export** > **Engineering Report (.html,.csv)**.



Results Report

The following is a summary of the information included in the results report (`<filename>.htm`). This data is provided in the report with reference to each material region in the simulation:

- Case name and time-stamp
- Solution information: CPU time, Elapsed Time and Number of Iterations
- Mass flow data, at each boundary
- Mass flux data, at each boundary
- Area averaged boundary data, for the following scalars:

  - Pressure (Pa)
  - Total Pressure (Pa)
  - Pressure Coefficient
  - Total Pressure Coefficient
  - Turbulent Kinetic Energy (J/kg)
  - Turbulent Dissipation Rate (m2/s3)
  - Turbulent Viscosity (Pa-s)
  - Density (kg/m3)
- Mass flow averaged boundary data, for the following scalars:
  - Pressure (Pa)
  - Total Pressure (Pa)
  - Pressure Coefficient
  - Total Pressure Coefficient
  - Turbulent Kinetic Energy (J/kg)
  - Turbulent Dissipation Rate (m2/s3)
  - Turbulent Viscosity (Pa-s)
  - Density (kg/m3)
- Force data, at each boundary:
  - Net Force [Fx, Fy, Fz]
  - Pressure Force [Fx, Fy, Fz]
  - Shear Force [Fx, Fy, Fz]
- Force coefficient data, at each boundary:
  - Coefficient of Net Force [Cfx, Cfy, Cfz]
  - Coefficient of Pressure Force [Cfx, Cfy, Cfz]
  - Coefficient of Shear Force [Cfx, Cfy, Cfz]

Setup Report

The following is a summary of the information found in the simulation setup report (`<filename>_caseinfo.htm`):

- The parts used in the simulation, and their associated regions
- The surfaces for each part, and their associated boundaries
- The continua used in the simulation, with the following information:
  - Physics models applied to the continuum
  - Materials properties
  - Turbulence properties
  - Segregated flow properties
  - Reference values
  - Initial conditions
- The mesh pipelines used in the simulation, with the following information:
  - Polyhedral mesher properties

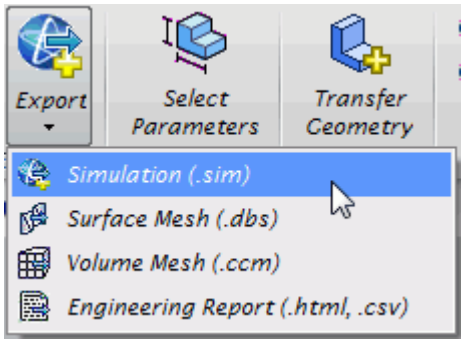
  - Prism layer mesher properties (if activated)
  - Surface remesher properties
  - Reference values
  - Volumetric controls
- The regions used in the simulation, and their associated continua and mesh pipeline
- The boundaries for each continua, with the applied conditions and values
- Solver parameters
- Stopping criteria
- Units

Exporting a STAR-CCM+ Simulation

You can export the current simulation in STAR-NX to the native STAR-CCM+ simulation (`.sim`) file format. The `.sim` file format retains all simulation settings defined in STAR-NX, as well as the surface and volume meshes.

To export a simulation file:

1. From the STAR-NX ribbon, click **Export** > **Simulation (.sim)**.



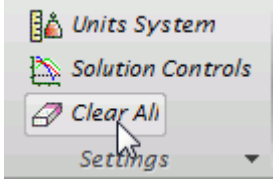
Resetting STAR-NX Settings

STAR-NX saves all case setup information to the CAD file. This means that all continuum settings, boundary settings, solution control settings, and mesh settings, are stored within the CAD geometry file. You can remove this information from the CAD geometry file.

Caution: This action is irreversible.

To clear all STAR-NX settings:

1. From the STAR-NX ribbon, click **Clear All**.



Guidelines

This section aims to provide guidance and assistance on making a geometry part suitable for CFD analysis using STAR-NX.

Contents:

Creating the Fluid Domain
Imprinting Parts

Creating the Fluid Domain

In the majority of CAD models, the fluid domain for analysis must be extracted from the original part or assembly. The topics that are containing within this section demonstrate how to prepare the fluid domain.

Contents:

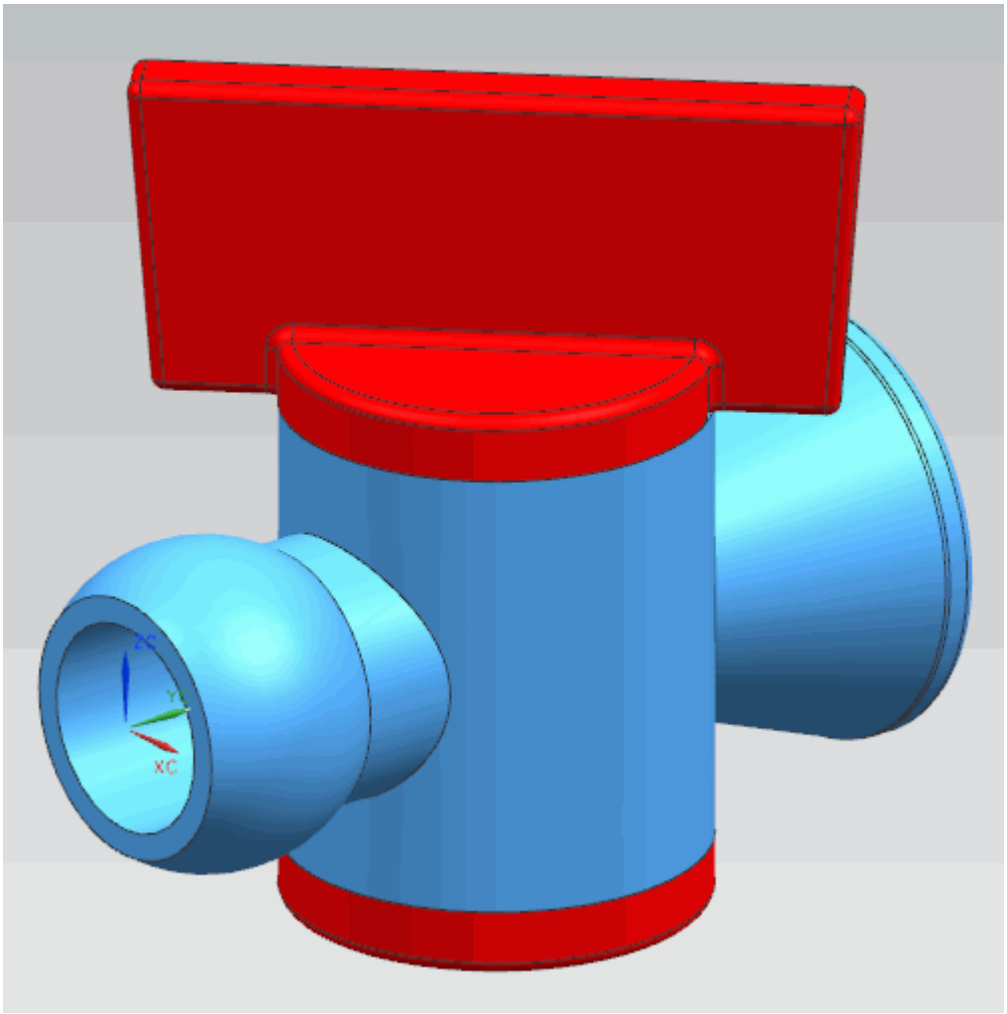
Internal Flow
External Flow
Multi Region Flow

Internal Flow

In general, the steps to follow in creating the fluid domain for an internal flow are:

- Work in an assembly. If the original geometry contains only a single part, create a new assembly that contains the part.
- Create an additional part that will close off any openings in the original geometry, these are called “caps”.
- Unite all parts with the caps to obtain one solid part.
- Create an overall block part that encloses all the other parts, this is the “enclosure”.
- Subtract all parts from the enclosure and delete the outer solid body--the remaining geometry is the fluid domain.

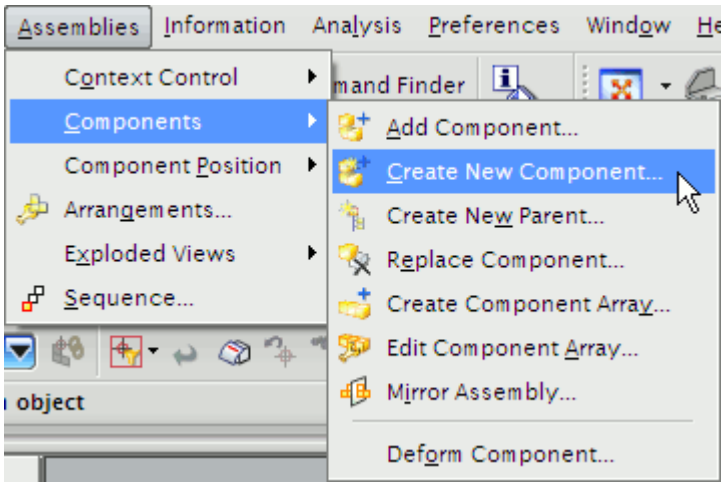
A worked example is provided below. In this case the assembly contains two parts: the valve (shown in red), and the valve housing (shown in blue).



1. Load the assembly into NX. If the geometry is a part, insert it into an assembly.

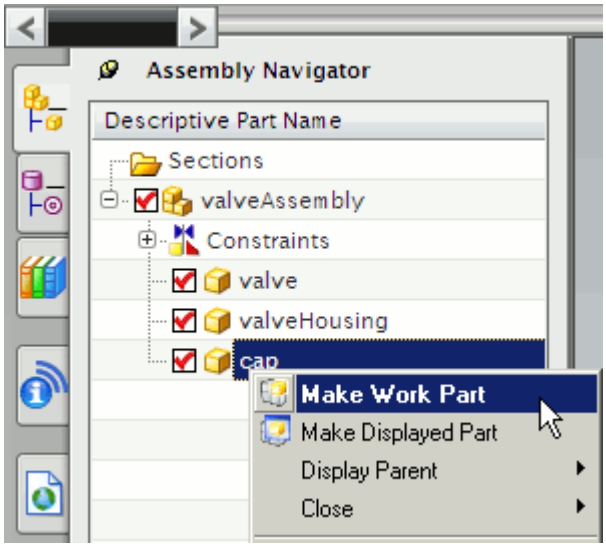
We will now insert a new component into the assembly. This part will close any openings in the assembly using caps.

2. Insert a new component into the assembly.

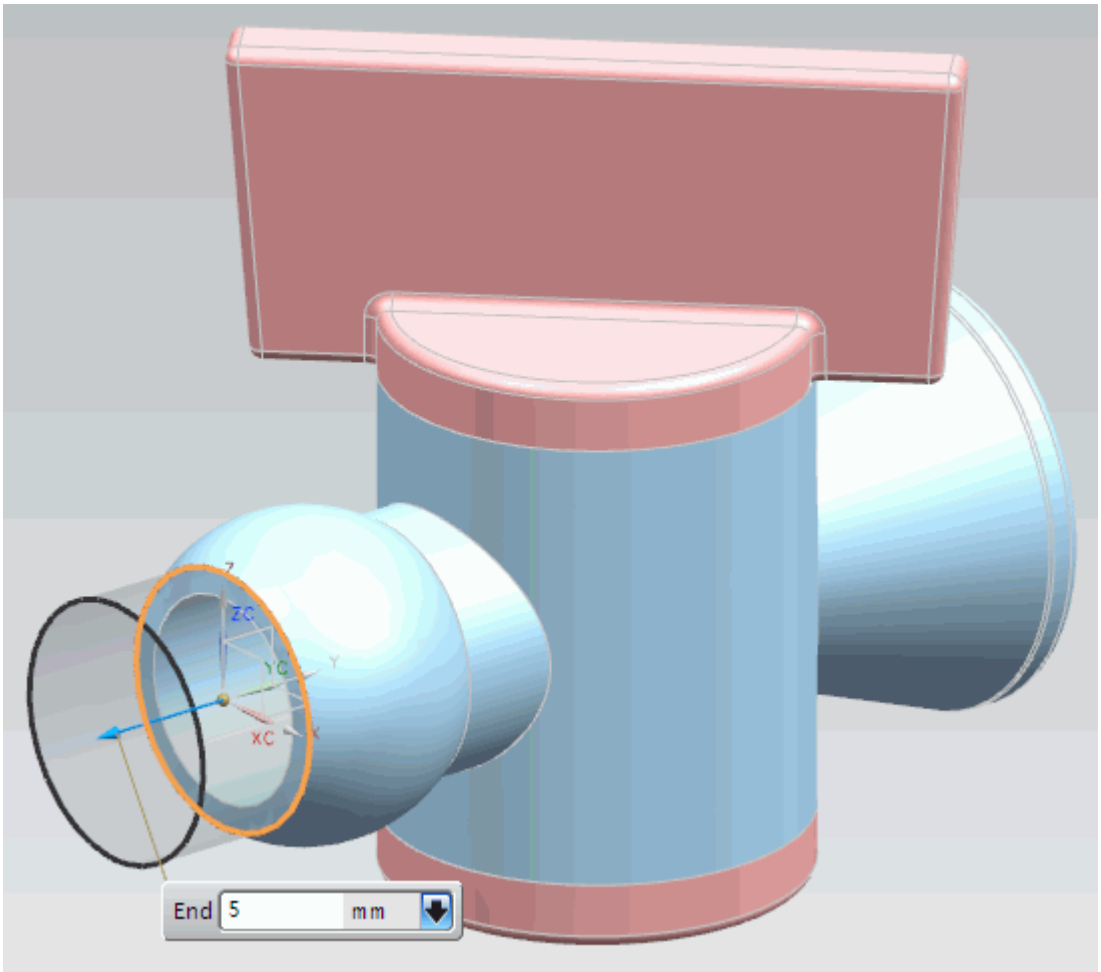


Give this component an appropriate name and then accept the default placement settings. In this case the new component was given the name `cap`.

3. Make the new component the work part.

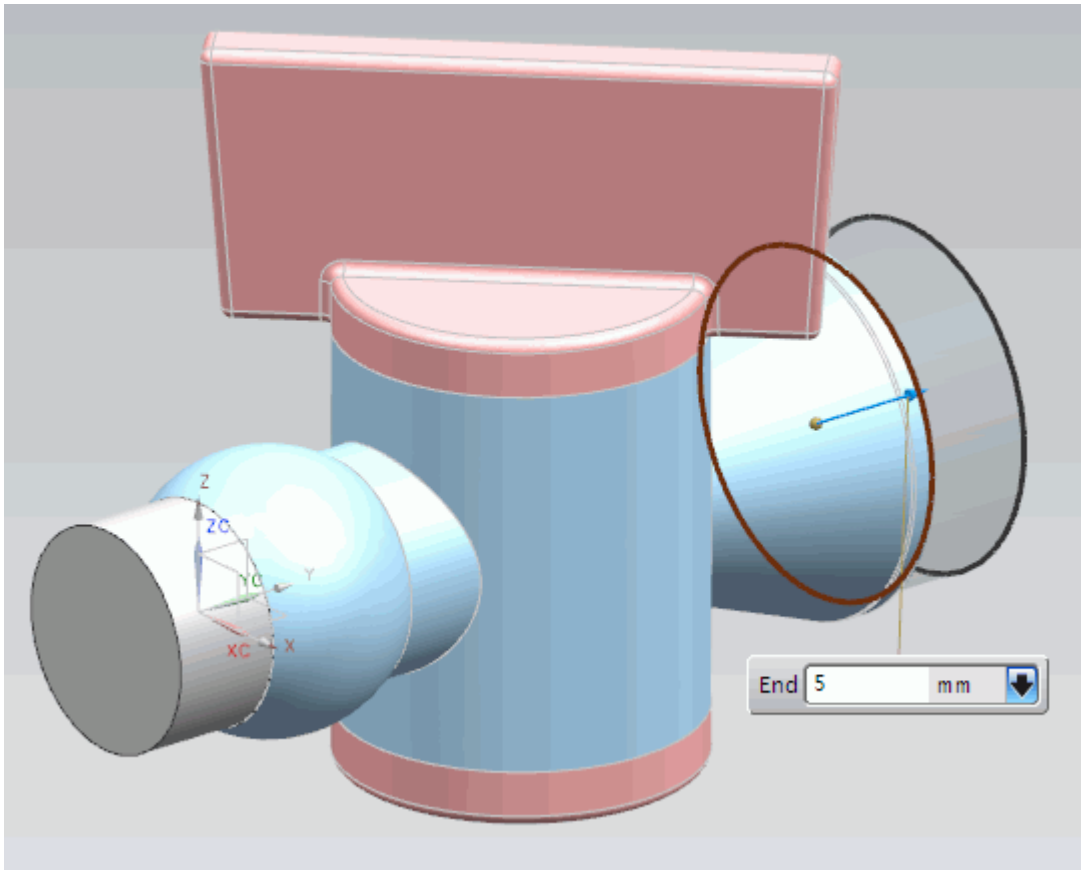


4. Select the **Extrusion** tool and select the outer edge of one of the openings on the geometry, as shown below.

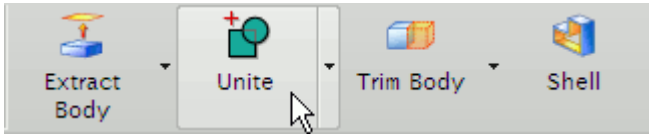


Ensure that the *Type Filter* is set to **No Selection Filter** and that the *Selection Scope* is set to **Entire Assembly**, as shown below, otherwise you will only be able to select features from the active part.

5. Perform the same actions for all other openings.

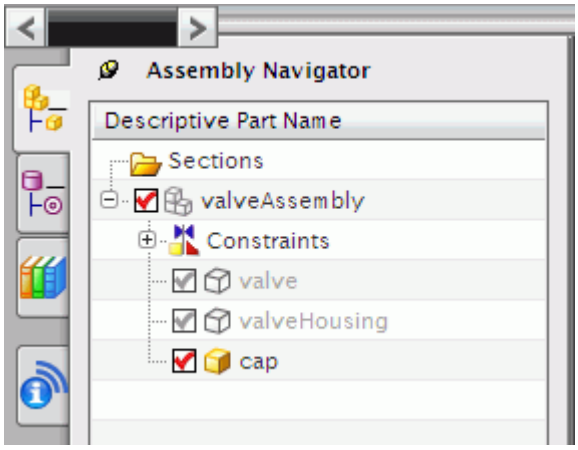


6. Use the **Unite** operation to join all the parts and form one solid.

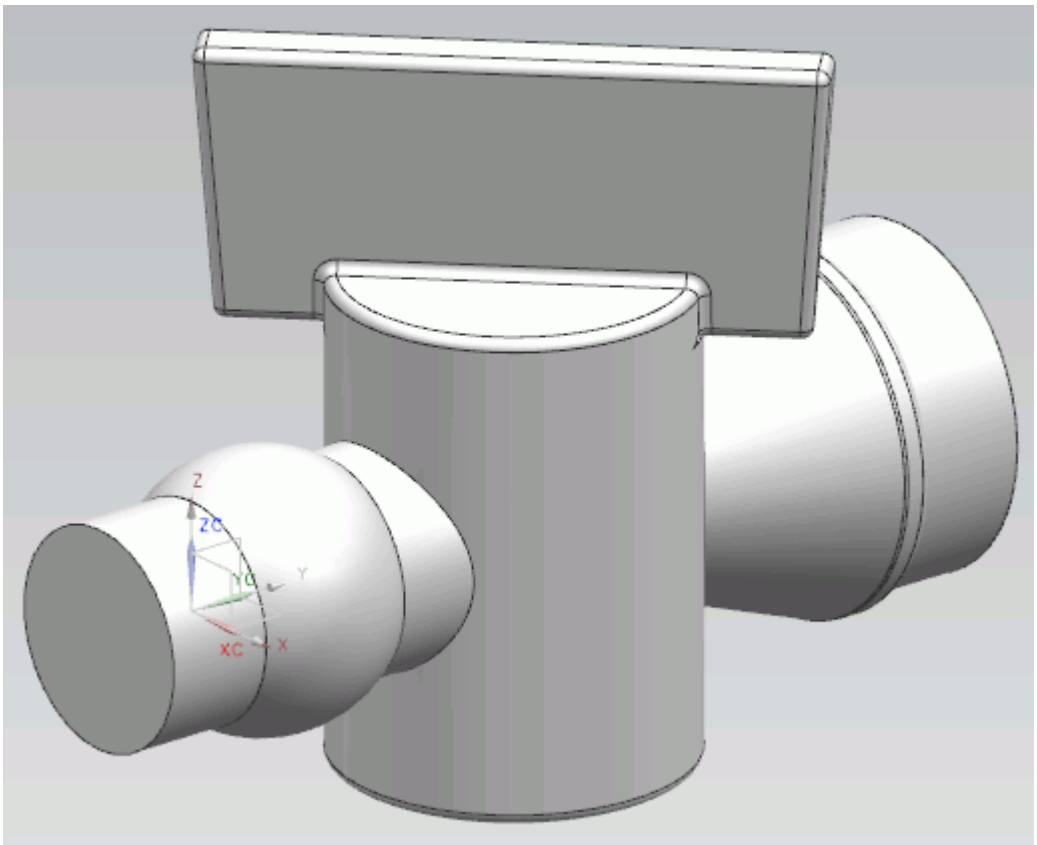


In the *Unite* dialog, select one of the caps you created in the previous step as the *Target*. Select the other parts (the other cap, valve and valve housing) as the *Tool*.

7. Hide the original valve and valve housing parts by going to the *Assembly Navigator* and clicking on the red tick besides each part. These should become greyed out as shown below.



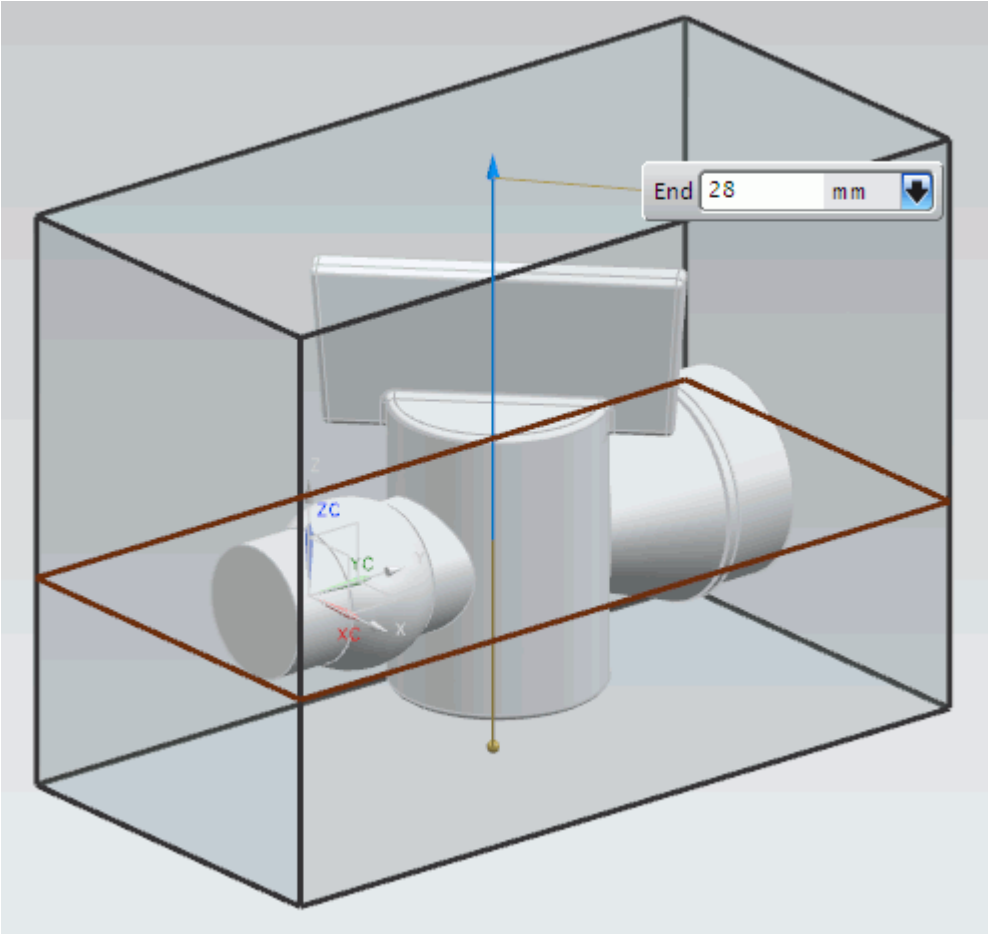
The part will appear as shown below.



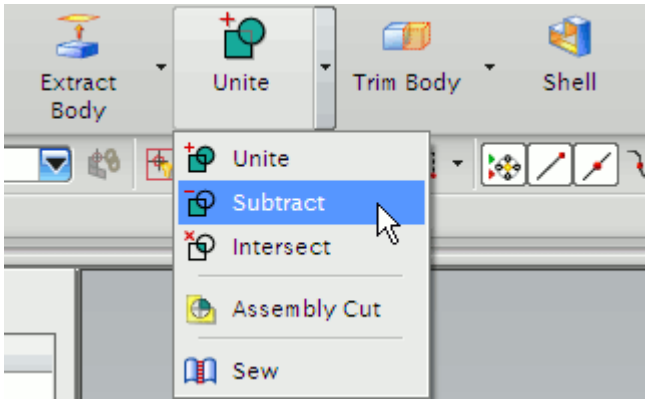
8. Return the to the assembly and make the assembly the work part.

A new component will be inserted into the assembly. A sketch that encloses the entire assembly will then be created and extruded. The valve, valve housing, and caps will then be subtracted from this, leaving two bodies. The outer body will be deleted, leaving only the internal fluid volume.

9. Insert another component into the assembly and make it the work part.
10. Create a sketch on any plane and sketch a bounding box that will, when extruded, enclose the entire assembly, as shown below.

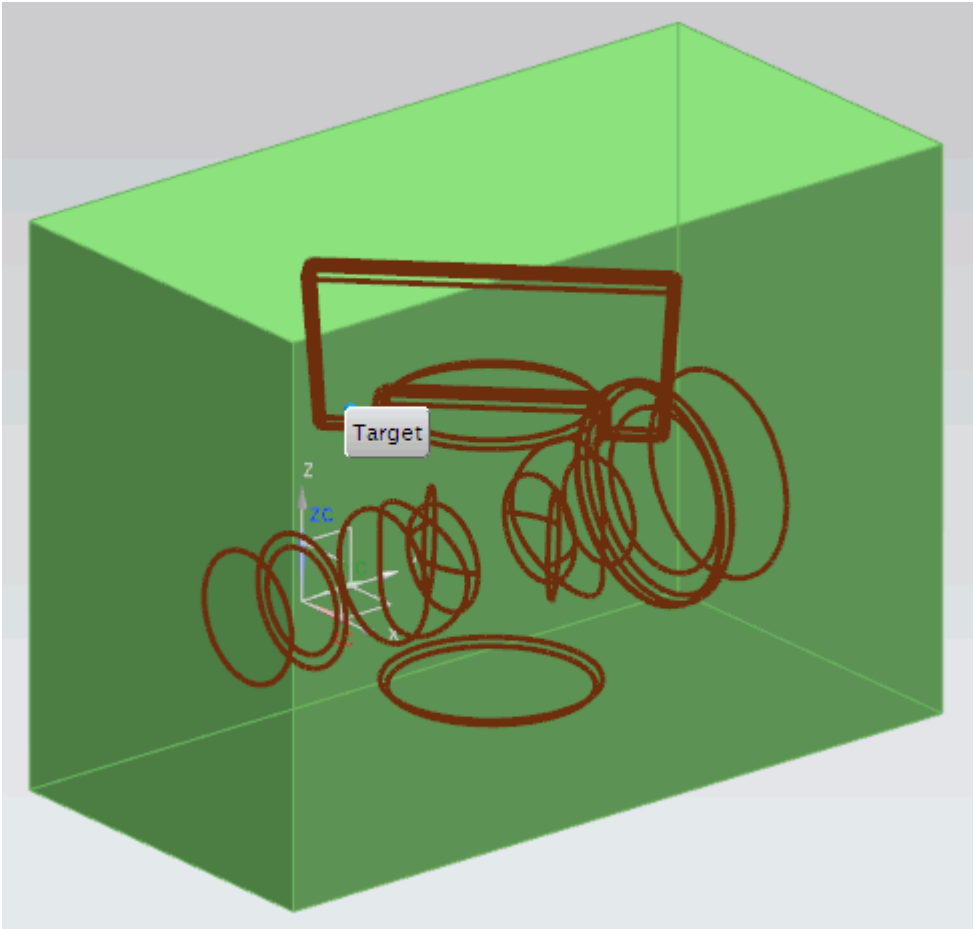


11. Subtract the cap part from the internal fluid part.

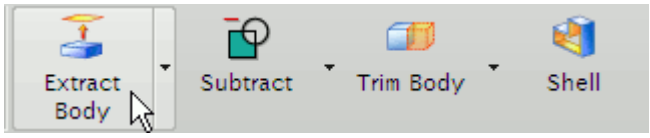


Select the fluid box as the *Target* and the caps as the *Tool*. To be able to select parts through solids, click on the

Highlight Hidden Edges button. Moving the mouse over the geometry will then highlight internal bodies in brown (the default color), as shown below.

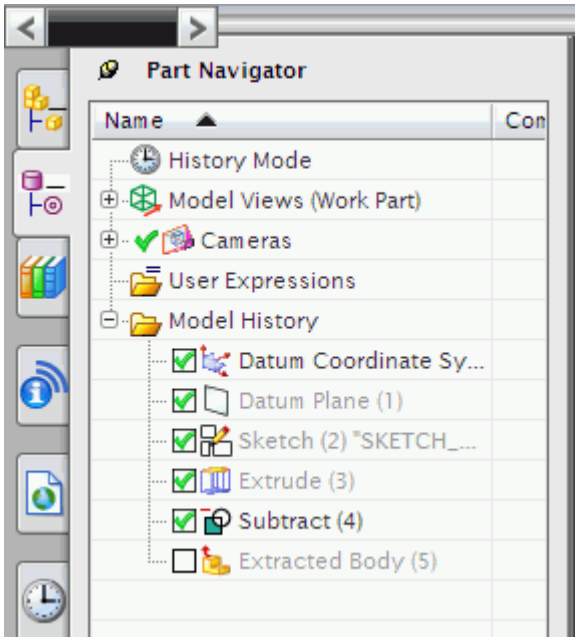


12. Use the **Extract Bodies** tool to extract the internal fluid region.

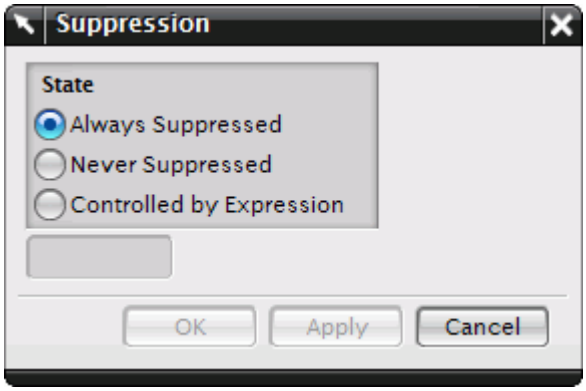


With the *Extract Body* dialog active, click on any face on the outer body to select it. Note that there is another free-floating solid body enclosed within this body, which will not get selected -- this is the internal fluid volume.

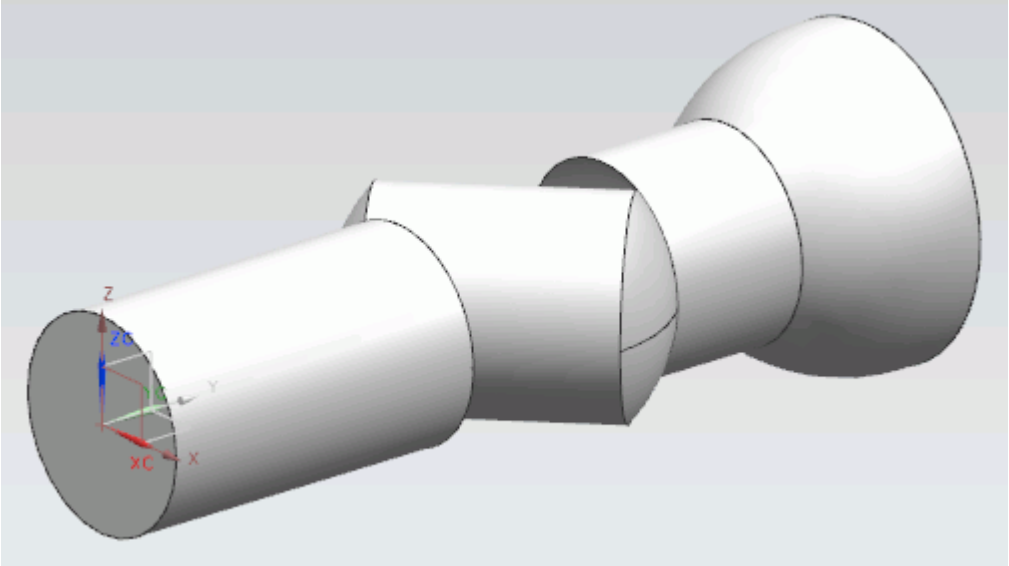
13. Hide the caps geometry.
14. Go to the **Part Navigator** and suppress the **Extracted Body** feature. This is done by clicking on the tick next to the feature--when suppressed the tick besides the node will disappear.



15. Suppress the components that you do not want to take part in the analysis, such as the valve, valve housing and caps, by right-clicking on these and selecting **Suppression...**
16. In the *Suppression* dialog that appears set the component to **Always Suppressed** and click on **OK**.



17. The extracted internal fluid volume is shown below.



18.
19.
20.
21.
22. Enter your first step here.
    Enter the result of your step here (optional).

Enter an example that illustrates the current task (optional).

External Flow

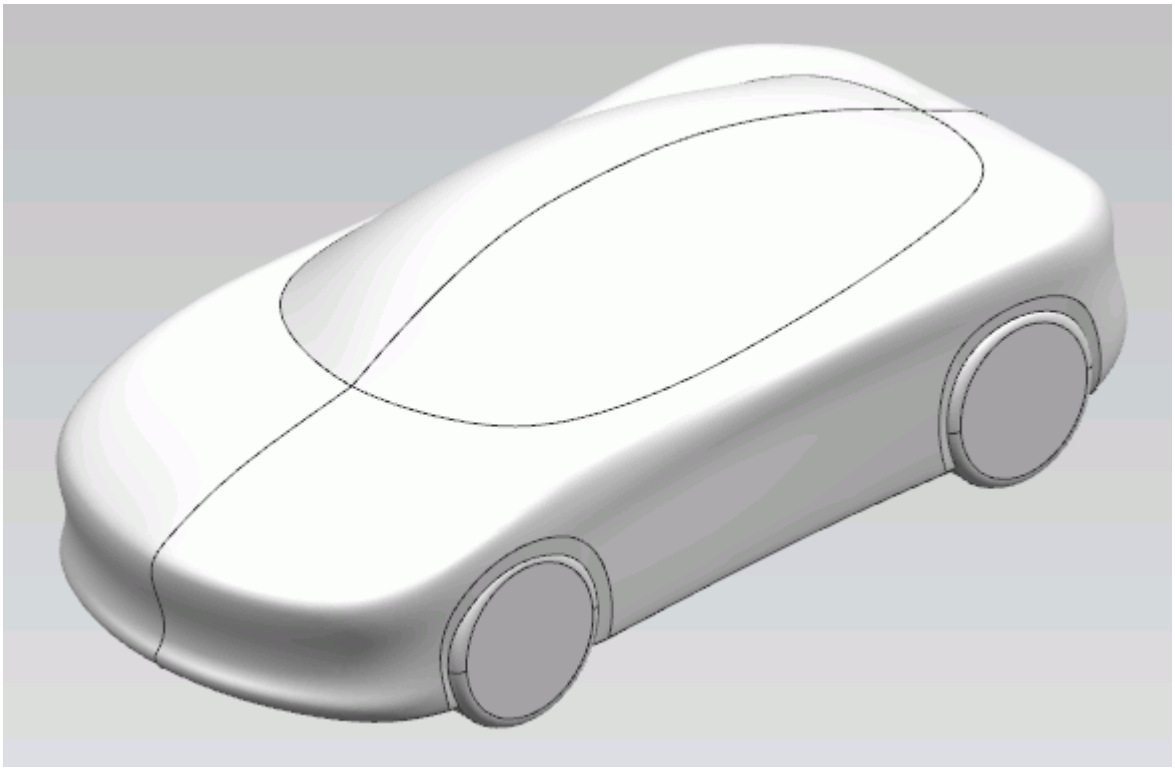
The generation of a domain for external aerodynamics is generally easier than for internal flow cases but the sizing of the domain is more critical. As a guideline (assuming a specific situation is not being modelled i.e. flow in a wind tunnel) the domain should be sized as follows:

- Inlet: 2 x part length from the front of the part
- Outlet: 4 x part length from the rear of the part
- Sides: 3 x part width from the side of the part
- Ceiling: 4 x part height from the top of the part

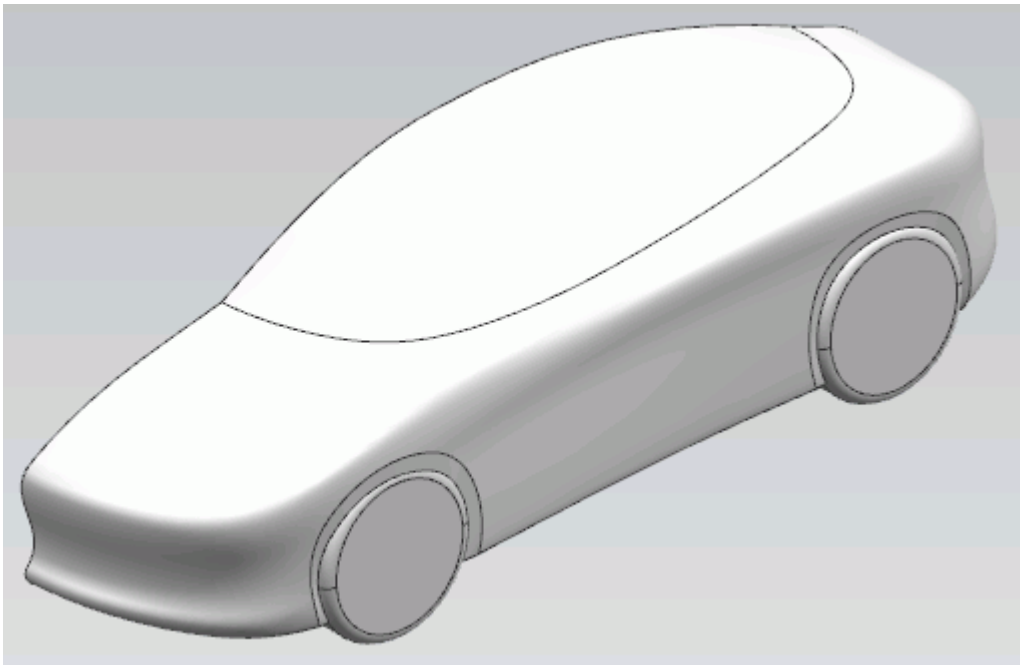
In general, the steps to follow in creating the fluid domain for an external flow are:

- Work in an assembly. If the original geometry contains only a single part, create a new assembly that contains the part.
- Create an overall block part that encloses all the other parts. The shape and size should correspond to the desired fluid domain.
- Subtract all parts from the overall block part - the remaining geometry is the fluid domain.

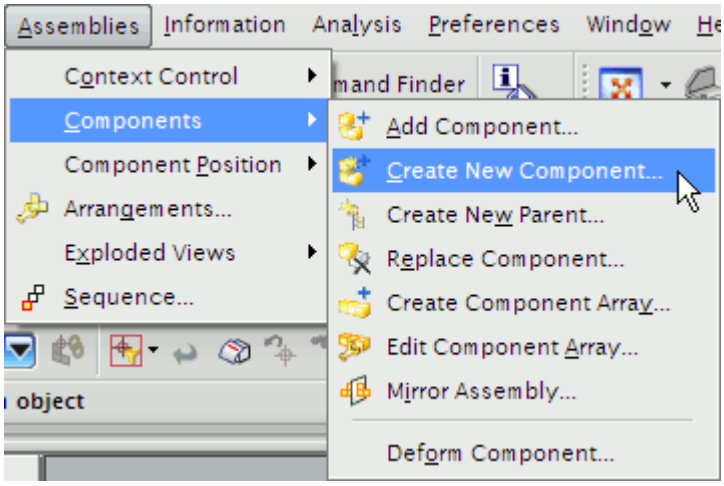
An example case is shown below. In this example a geometry of a car will be used.



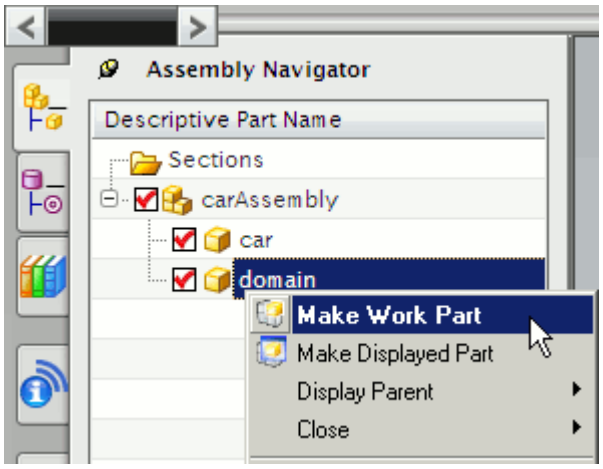
1. Load the assembly in NX. If the geometry is a part file, insert this into an assembly.
2. If the geometry is symmetrical, cut this in half to save on computation costs.
   The easiest way of doing this is to perform an extrusion to remove the unwanted half.



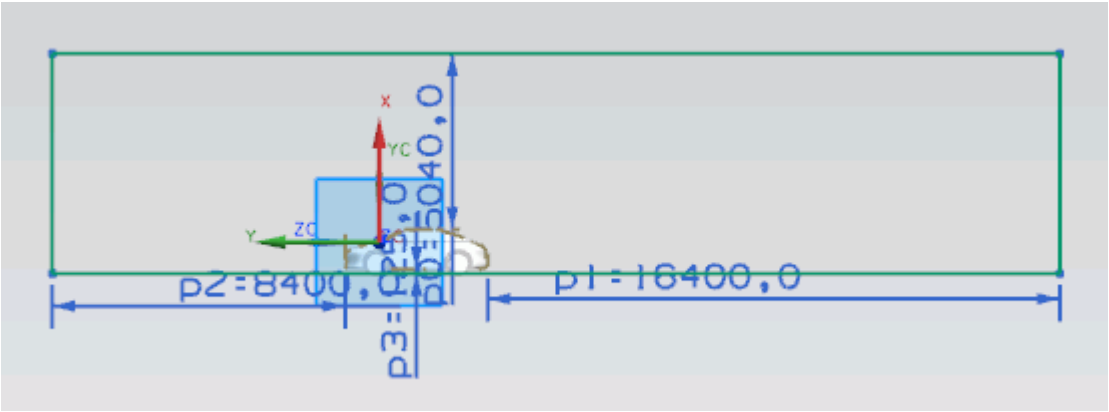
3. Insert a new component into the assembly.



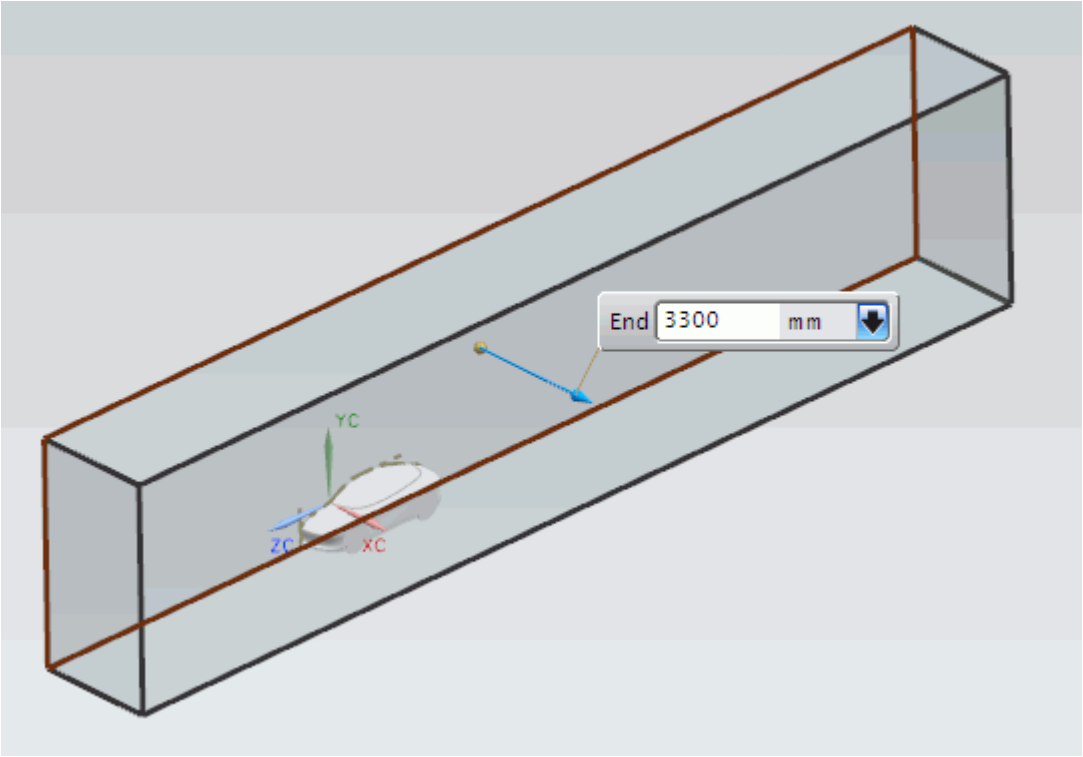
4. Give this component an appropriate name and then accept the default placement settings. In this example, the component was called `domain`.
5. Make the new component the active part, right click on it in the feature tree and select **Make Work Part**.



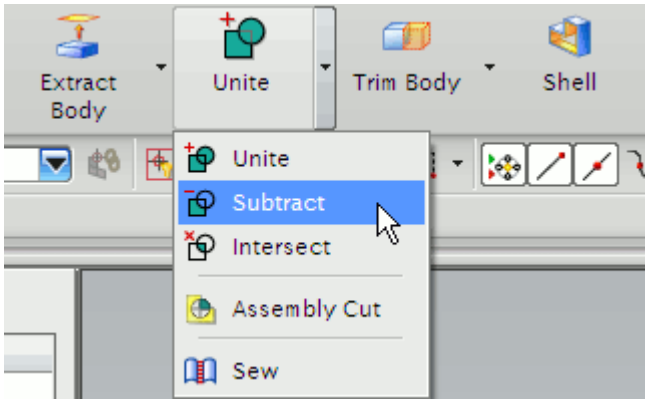
6. Create a sketch on the symmetry plane of the geometry. This sketch will represent the fluid domain profile.



7. Extrude the sketch to an appropriate length to complete the domain creation.



8. Subtract the car geometry from the domain geometry. This is done using the **Subtract** feature.

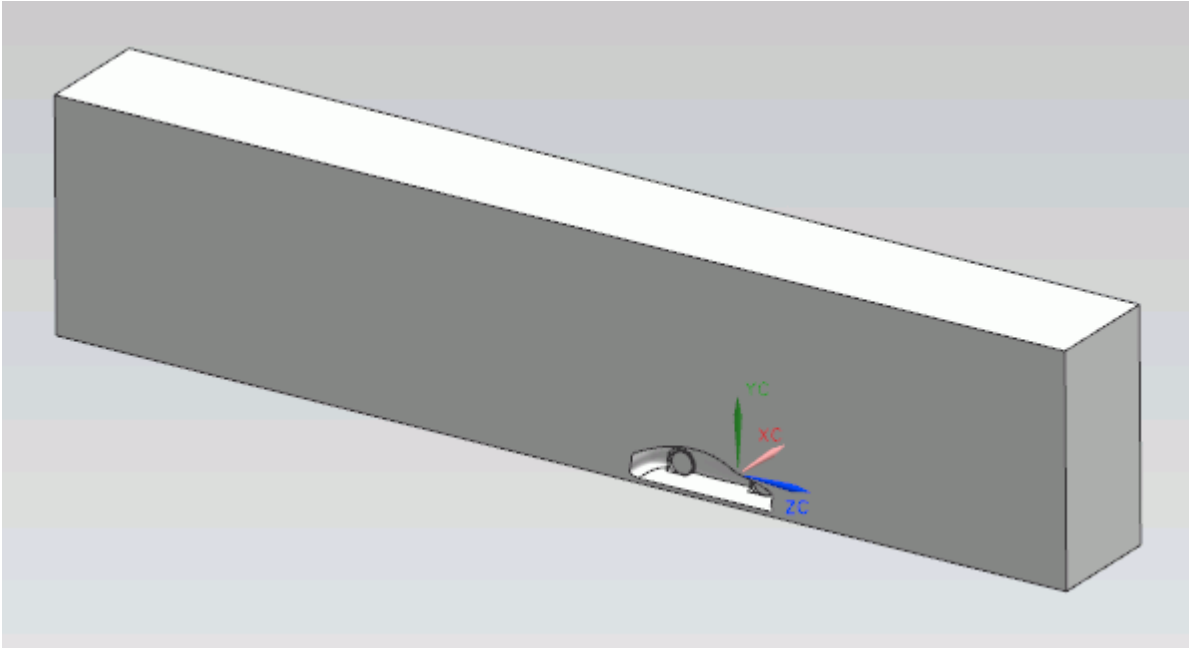


9. In the *Subtract* dialog, select the domain as the *Target* and the car as the *Tool*.

   Ensure that the *Type Filter* is set to **No Selection Filter** and that the *Selection Scope* is set to **Entire Assembly**, as shown below, otherwise you will only be able to select the active part.

10. Hide the car part, by clicking on the tick next to the part. Once done, the tick will be greyed out.

The final result is shown below.



Multi Region Flow

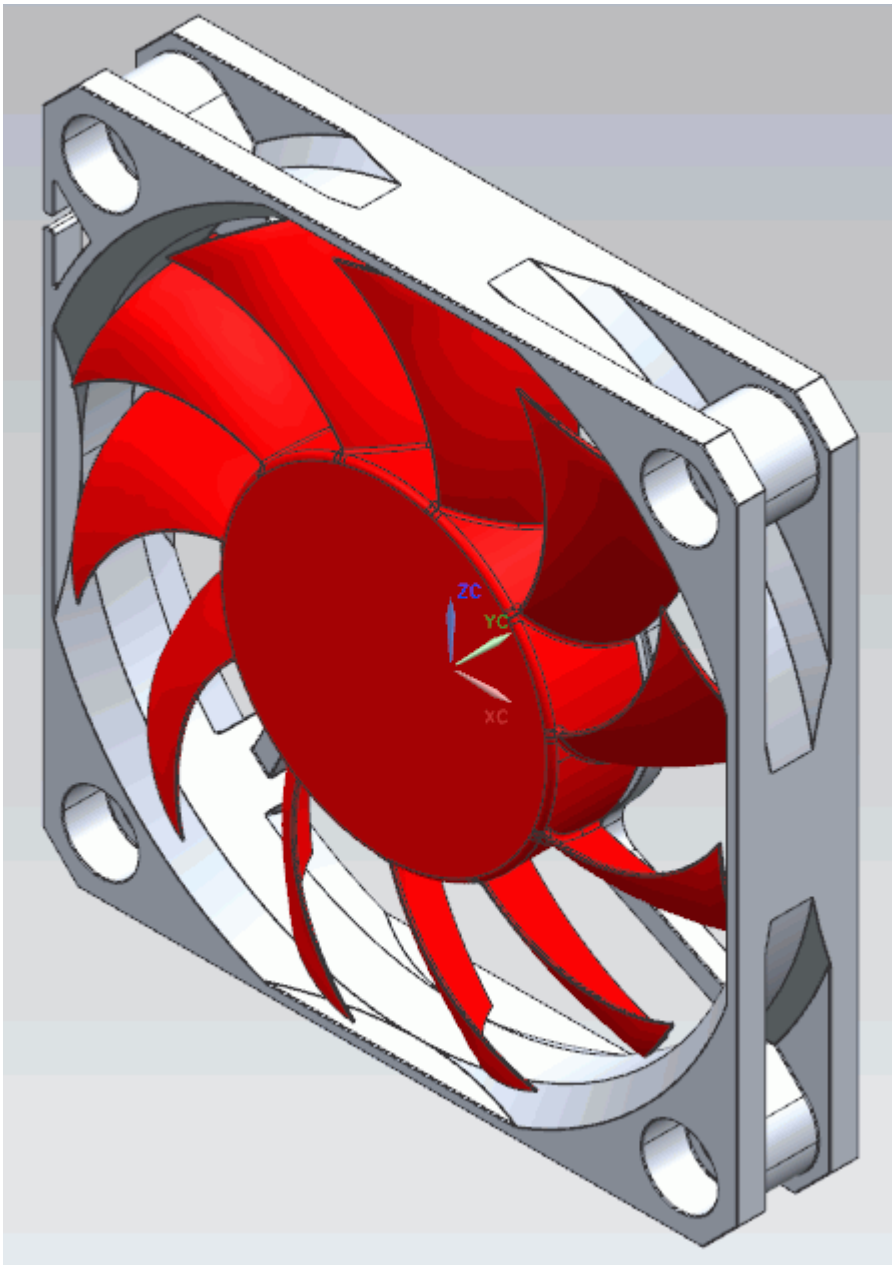
This section shows how to create the fluid domain for multiple regions such as rotating machinery and porous media flow.

For the numerical assumptions made using the Moving Reference Frame (MRF) technique to hold true, a minimum of two cells need to be present between the rotating-static interface and any wall boundaries (for example the fan blades), and the rotating region must be circular in section. Where there is a large clearance between the rotating object and any static objects, it is advisable to avoid placing the interface too close to either hence facilitating a small cell size on the interface. Where there is a small clearance between the rotating and static parts, there is little choice but to place the interface approximately midway between the two.

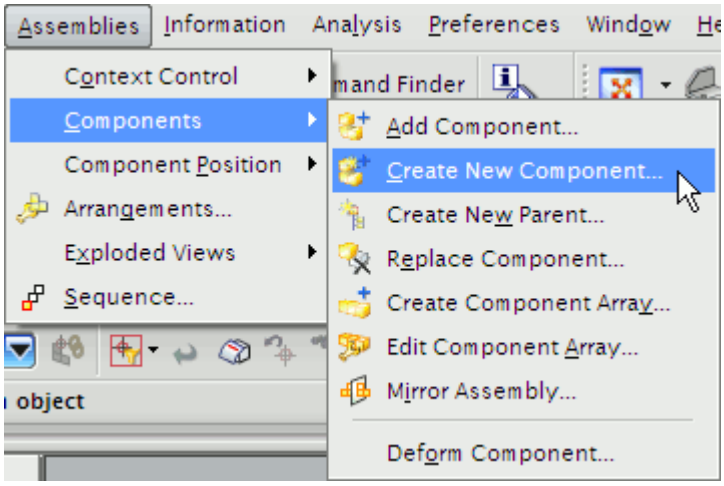
In general, the steps to follow in creating the fluid domain for multiple regions are:

- Work in an assembly. If the original geometry contains only a single part, create a new assembly that contains the part.
- Create an additional part that will enclose the rotating region (for example a fan) and subtract it from the part. In the case of a porous region, a separate part will be required for this as well.
- Create an additional part for the main static domain (the inlet and outlet regions) and subtract any obstructions/features that may be present there (for example the fan motor support arms on a fan casing).

In this example, a fan and fan case are used.

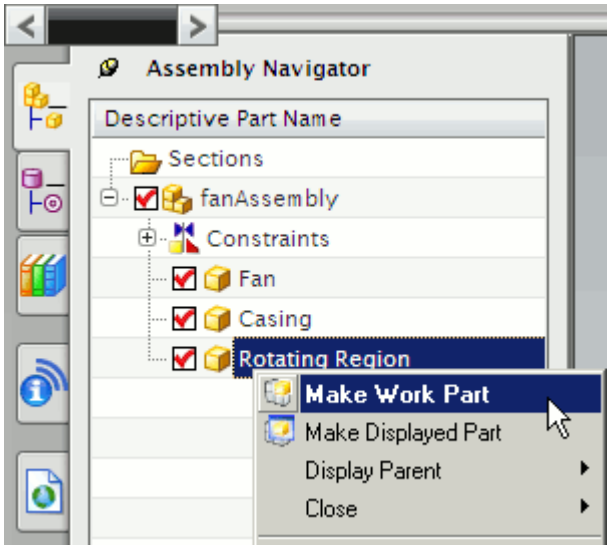


1. Load the assembly into NX.
2. Insert a new empty component into the assembly.

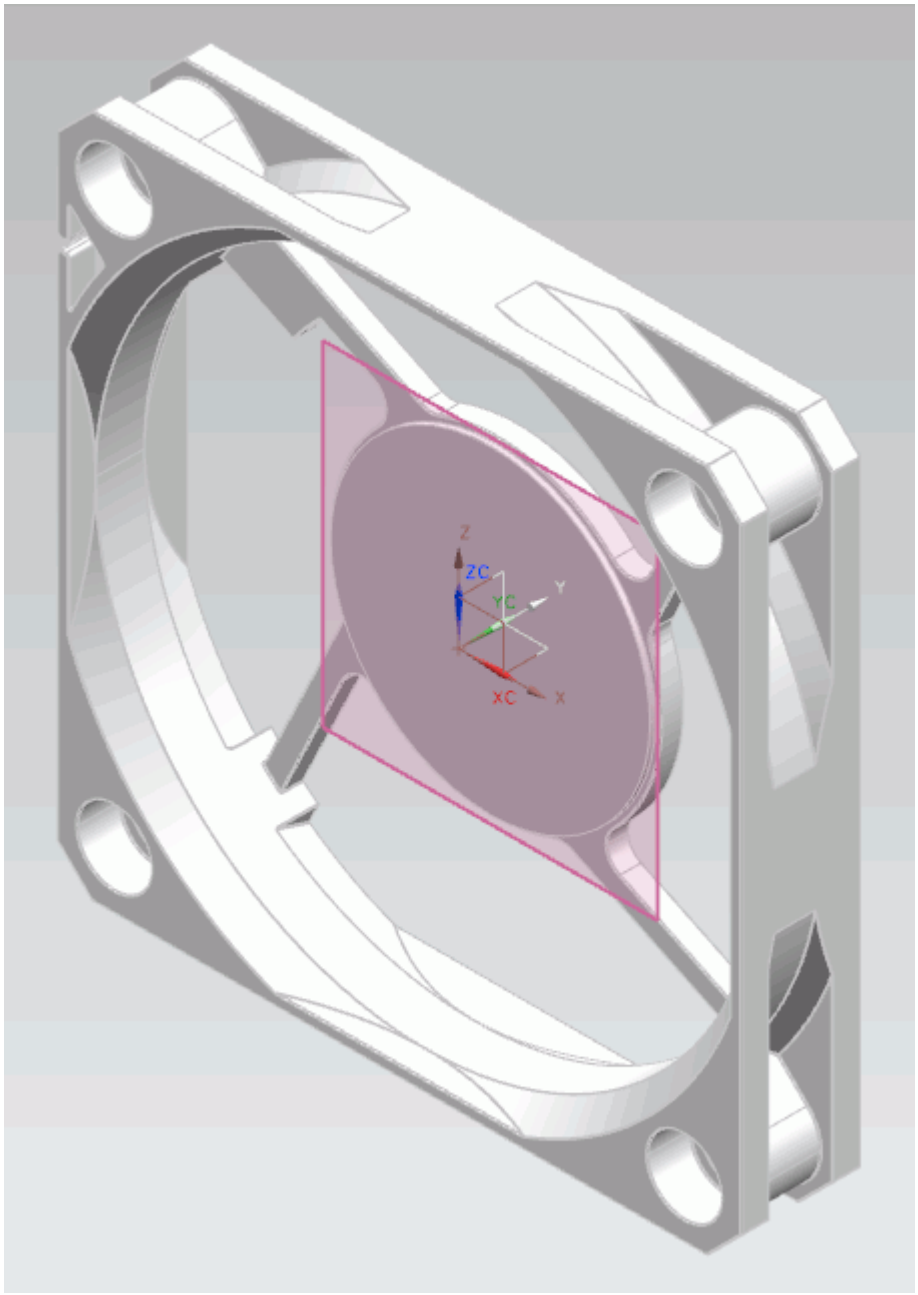


Give this component an appropriate name and then accept the default placement settings. In this example the new component was given the name `Rotating Region`.

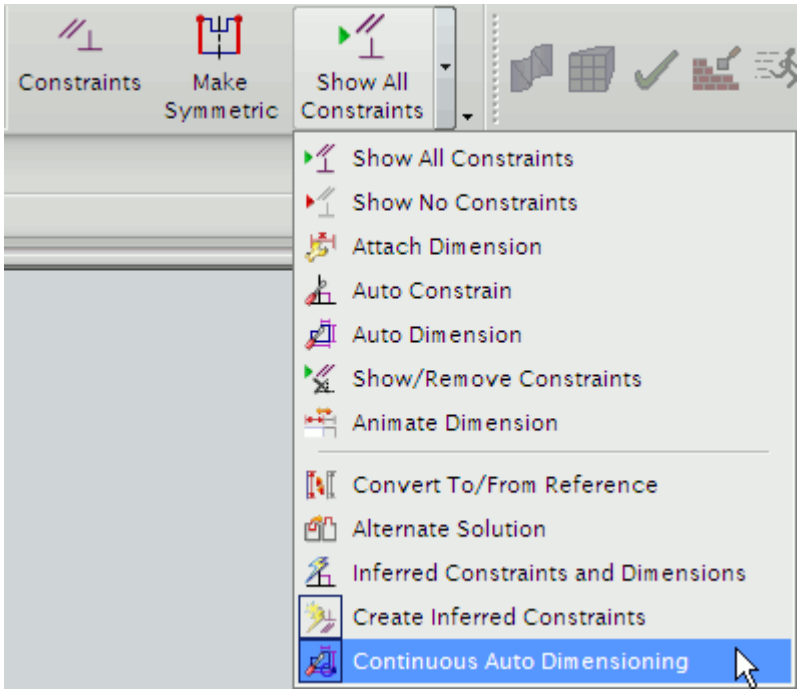
3. Make the new component the work part by right-clicking on its node in the assembly navigator and selecting **Make Work Part**.



4. Hide the Fan part by going to the assembly navigator and clicking on the red tick besides the part. The tick will turn grey and the part will not be visible in the geometry window.
5. Create a sketch on the contact face of the fan casing (where the fan and the fan casing are coincident).



6. If the continuous auto dimensioning feature of NX is turned on, disable it by clicking on the **Continuous Auto Dimensioning** button in the toolbar.

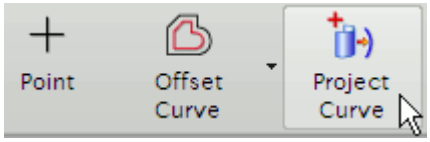


7. Draw a circle that shares the same center point as the sketch plane face.

   The size of the circle is not important at this stage as it will be made coincident with the innermost edge of the fan casing in a later step.

8. Ensure that the *Type Filter* is set to **No Selection Filter** and that the *Selection Scope* is set to **Entire Assembly**, as shown below, otherwise you will not be able to select edges from other parts.

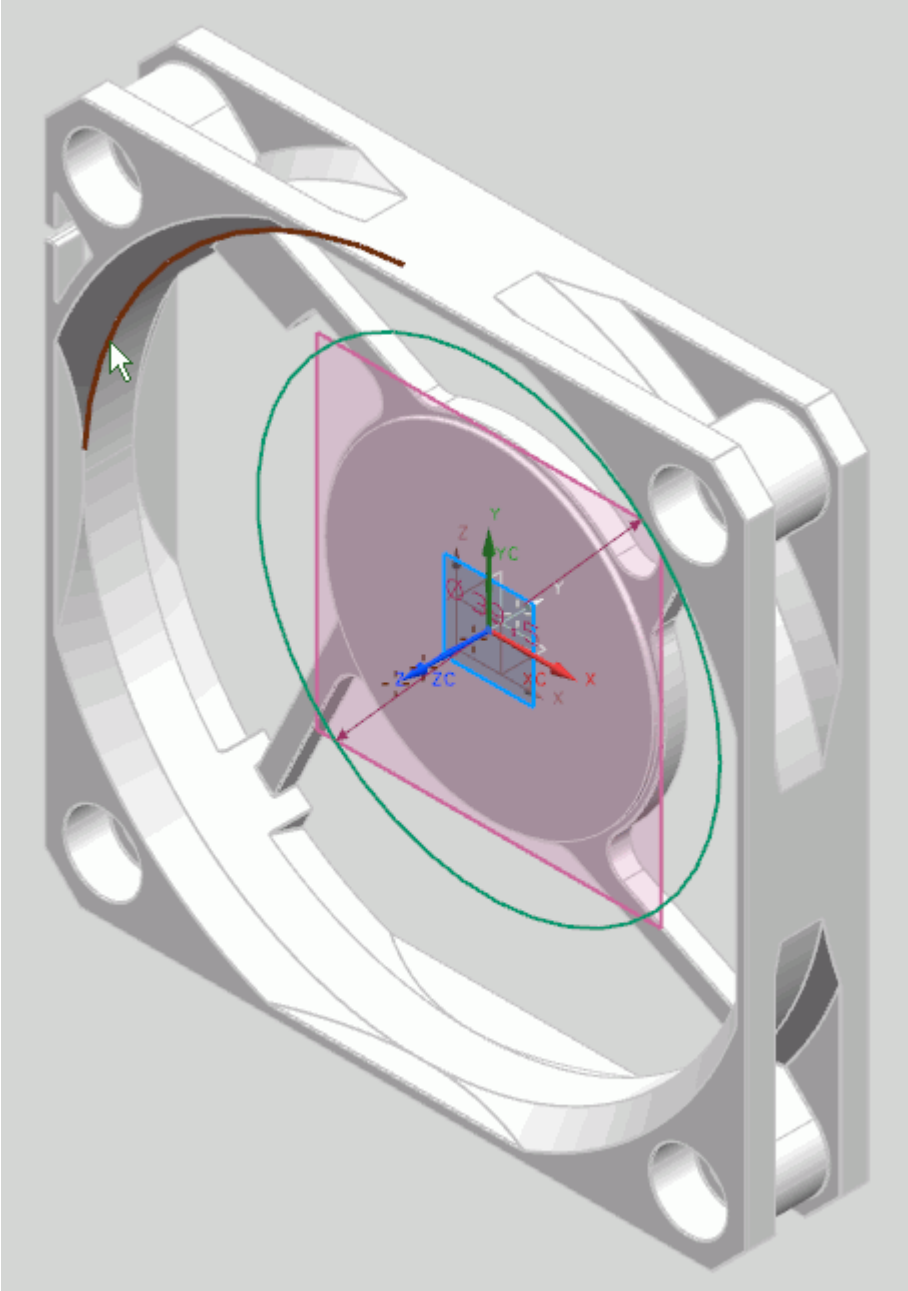
9. Click on the **Project Curve** button in the toolbar



   If this button is missing from the toolbar, you may need to click on the (**Open in Sketch Task Environment**) button to open the active sketch in the sketch task environment.

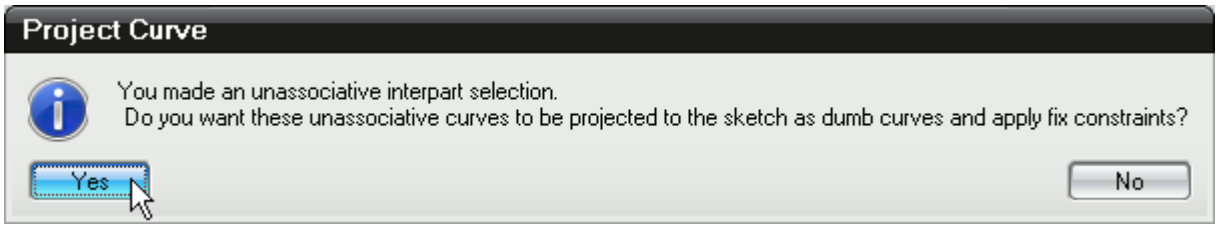
   The *Project Curve* dialog appears.

10. Select the innermost edge of the fan casing.

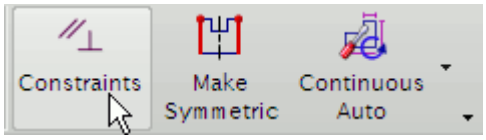


11. Exit the dialog.

    A message will appear warning you that the projection will not be associative. Click on **Yes** to accept and close the dialog.

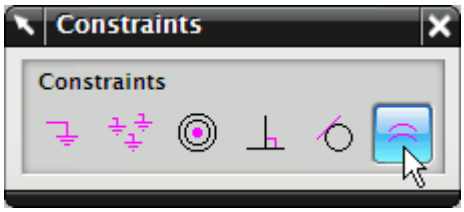


12. Right-click on the projected edge and select **Convert to Reference**.
13. Click on the **Constraints** button in the toolbar.

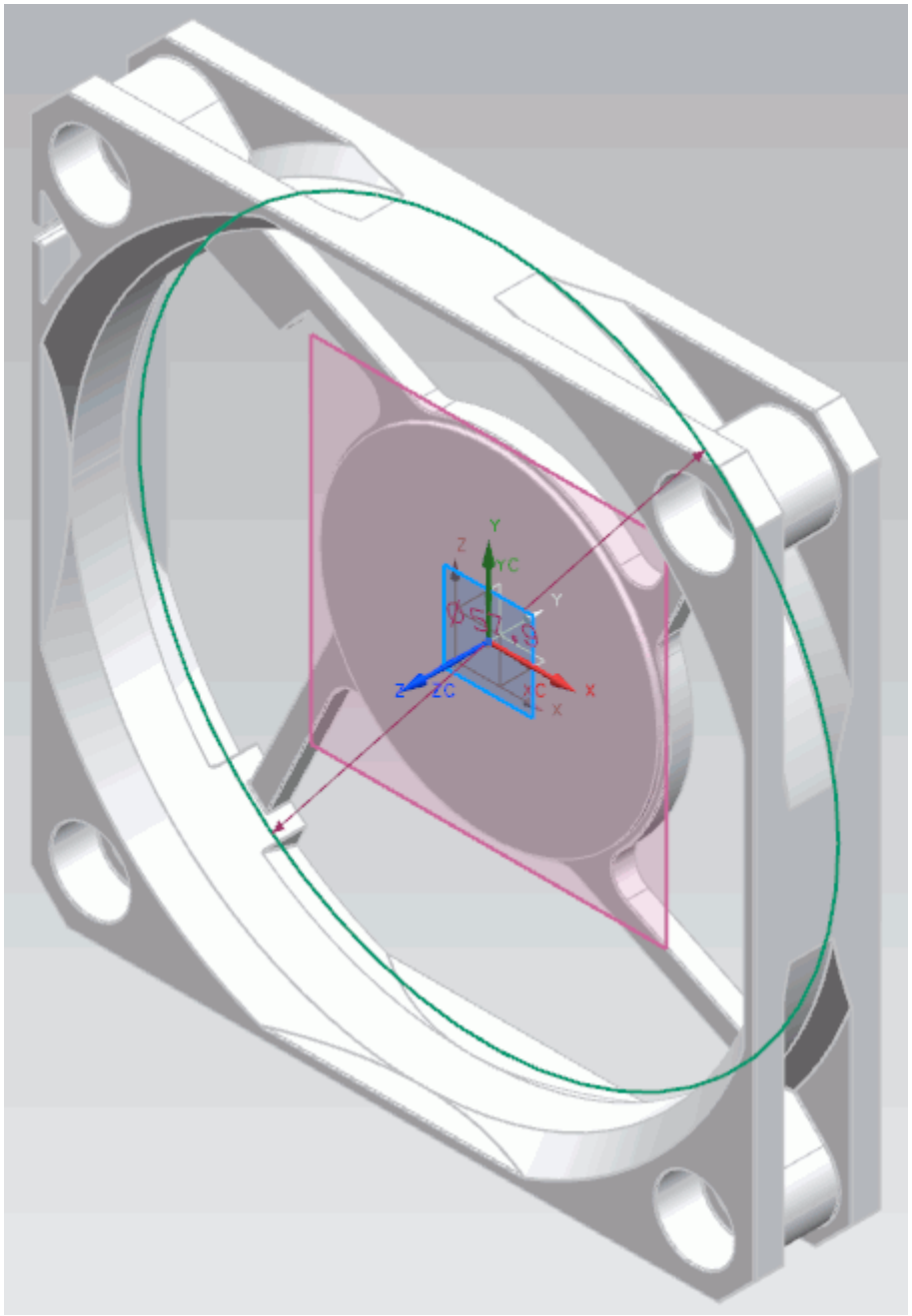


The *Constraints* dialog will appear.

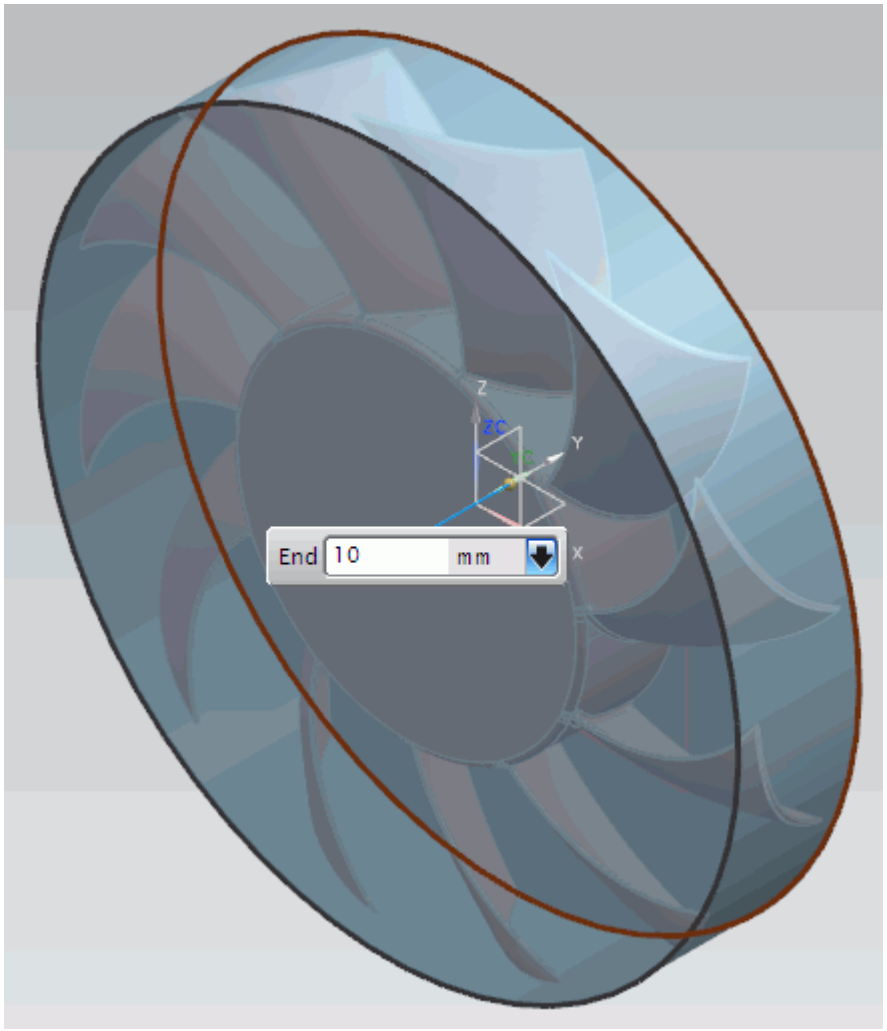
14. Select the circle and the innermost edge of the fan casing, as shown below.
    This will add new options in the *Constraints* dialog.
15. Click on the **Equal Radius** button to make the circle and the projected edge have the same radius.



The circle will snap to the projected edge, as shown below.

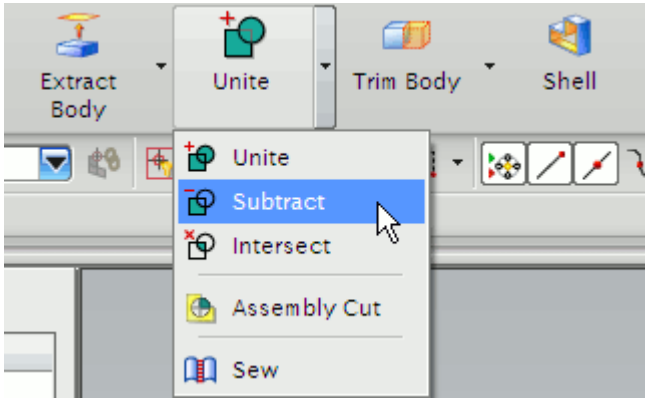


16. Hide the fan casing and show the fan.
17. Extrude the sketch so that it fully encloses the fan.



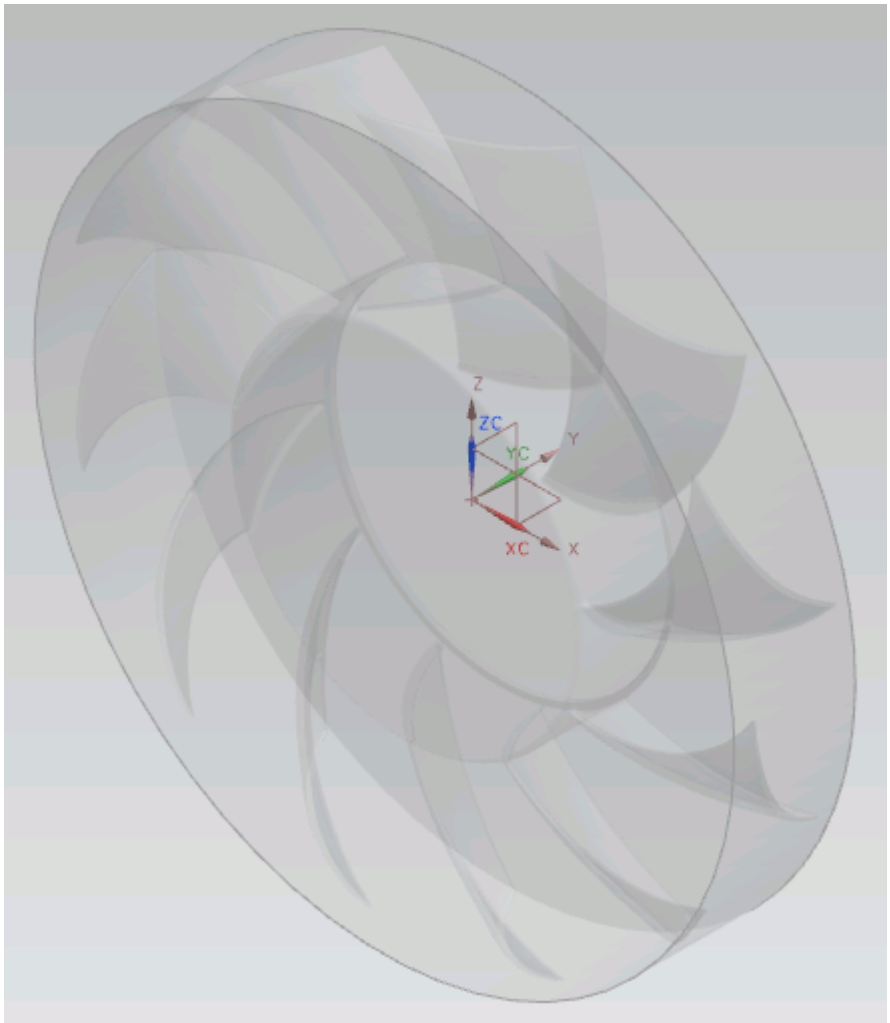
Ensure that no boolean operations are set to take place in the *Extrude* dialog -- these might result in an error during the extrusion.

18. Use the boolean subtract operation to remove the fan from the Rotating Region.



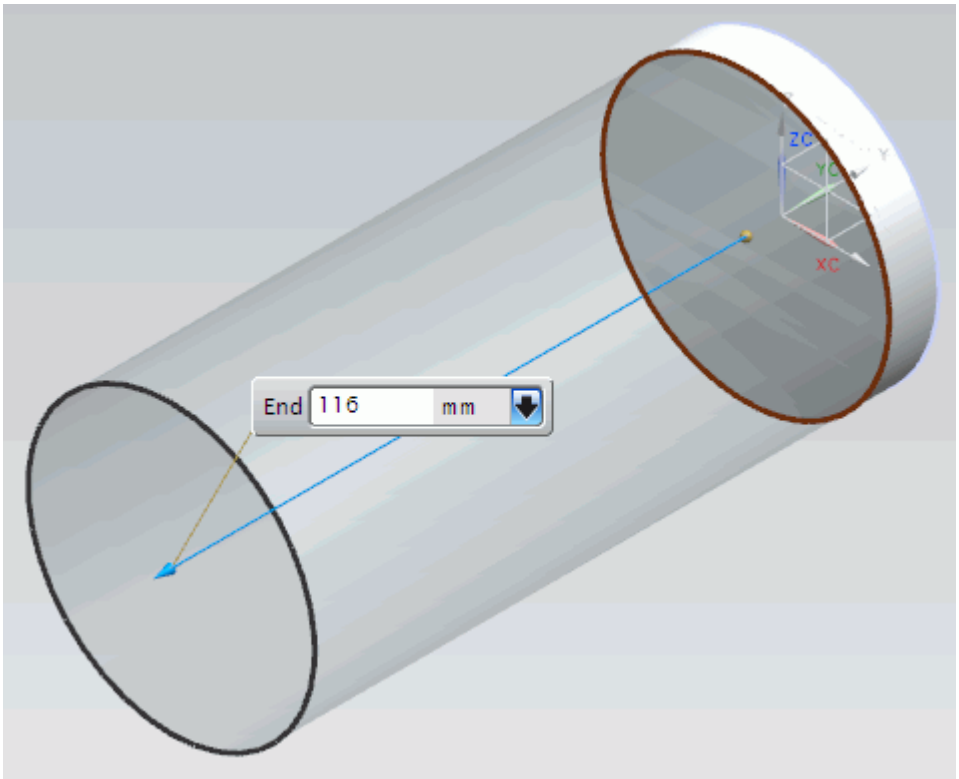
You can use the **See-Thru** view with the **Highlight Hidden Edges** button to see and select the fan inside the Rotating Region.

Hiding the fan and using the See-Thru view will show the result of this operation.



19. Return to the assembly and make the assembly the work part.
20. Hide the fan part; this will no longer be required.
21. Insert another component into the assembly as previously described and make that the work part.
22. Use the **Extrude** tool and select the existing outer edge on the front face of the Rotating Region. To make it easier to see which is the front face, use the see-thru tool.

The preview of the extrude is shown below.

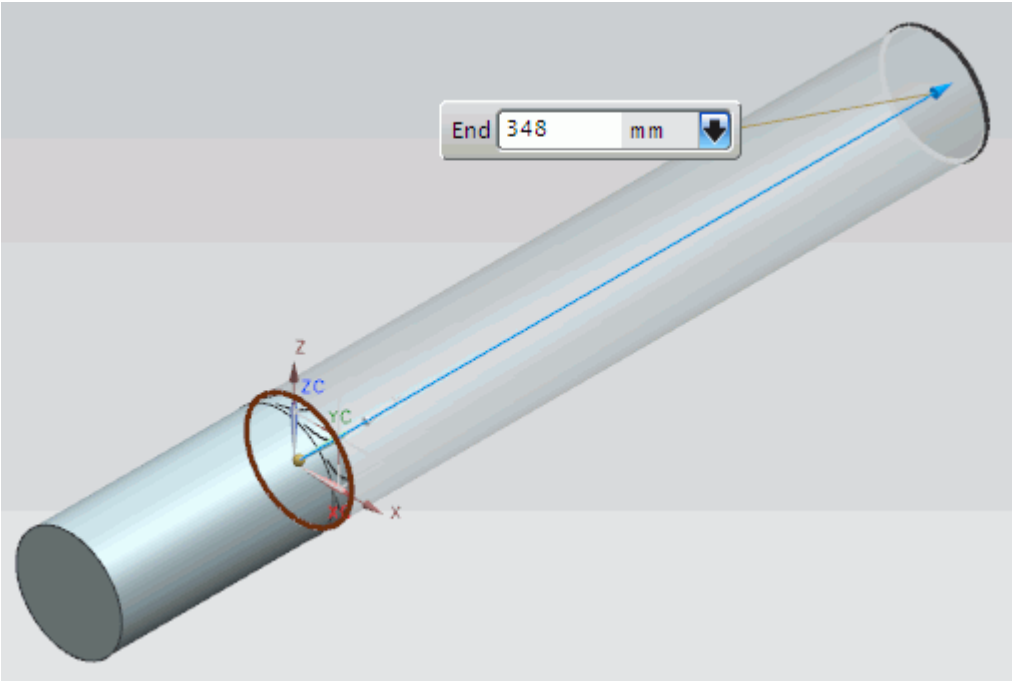


The extrusion depth should be suitable for the case you are investigating.

23. Return to the assembly and make this the work part.

The outlet region will be created using the same technique used for the inlet region.

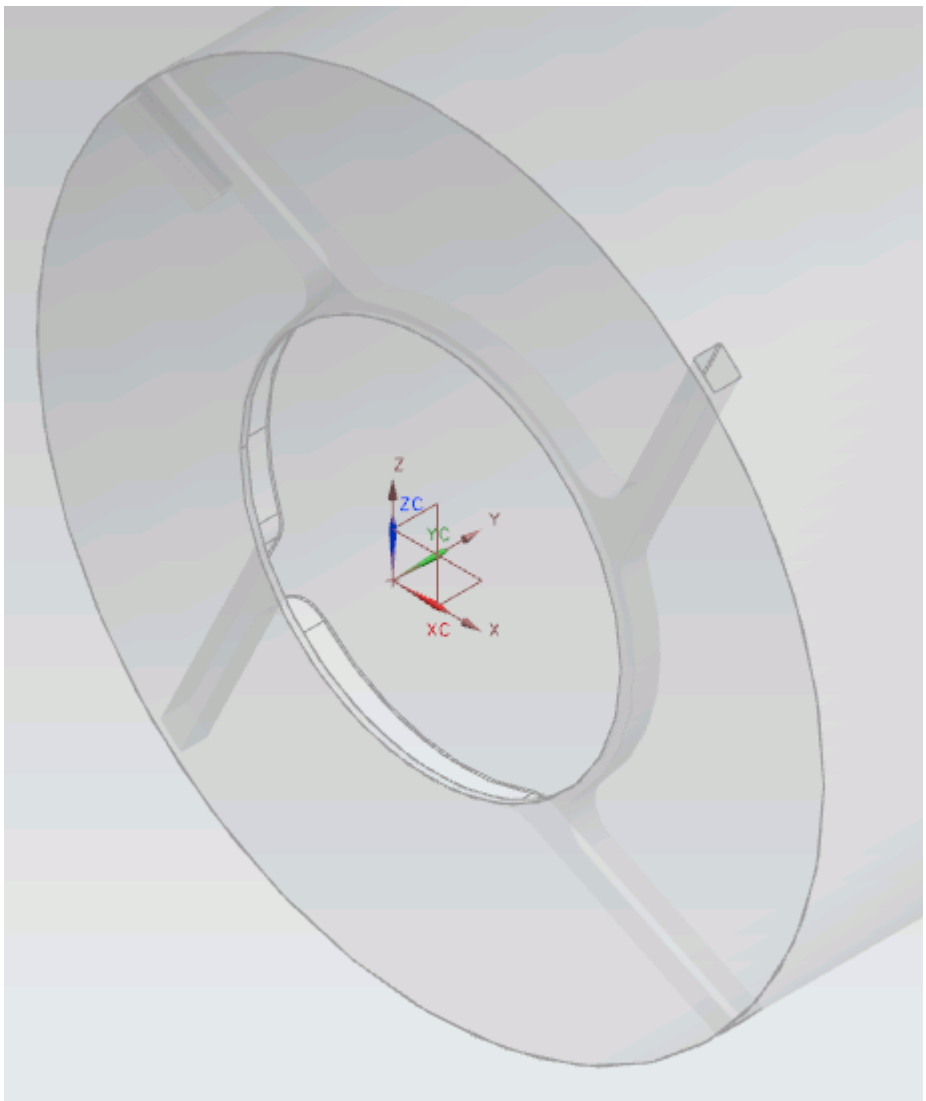
24. Insert another part into the assembly. Make this the working part.
25. Create an extrusion using the outer edge of the back face of the Rotating Region.



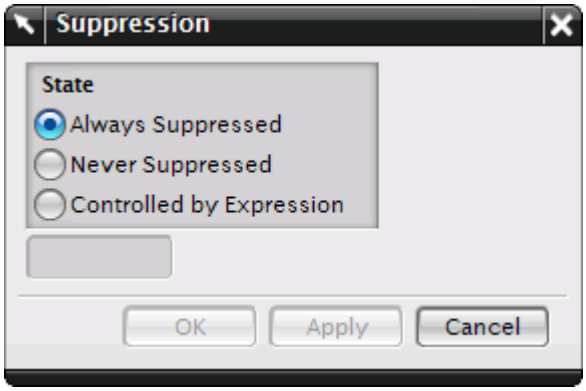
26. Hide the Inlet Region and the Rotating Region, leaving only the Outlet Region visible.
27. Show the fan casing part; this will be subtracted from the Outlet Region.
28. Subtract the fan casing from the outlet region.

    In the *Subtract* dialog select the outlet region as the *Target* and the fan casing as the *Tool*.

    The result of this operation is shown below (using the See-Thru view).

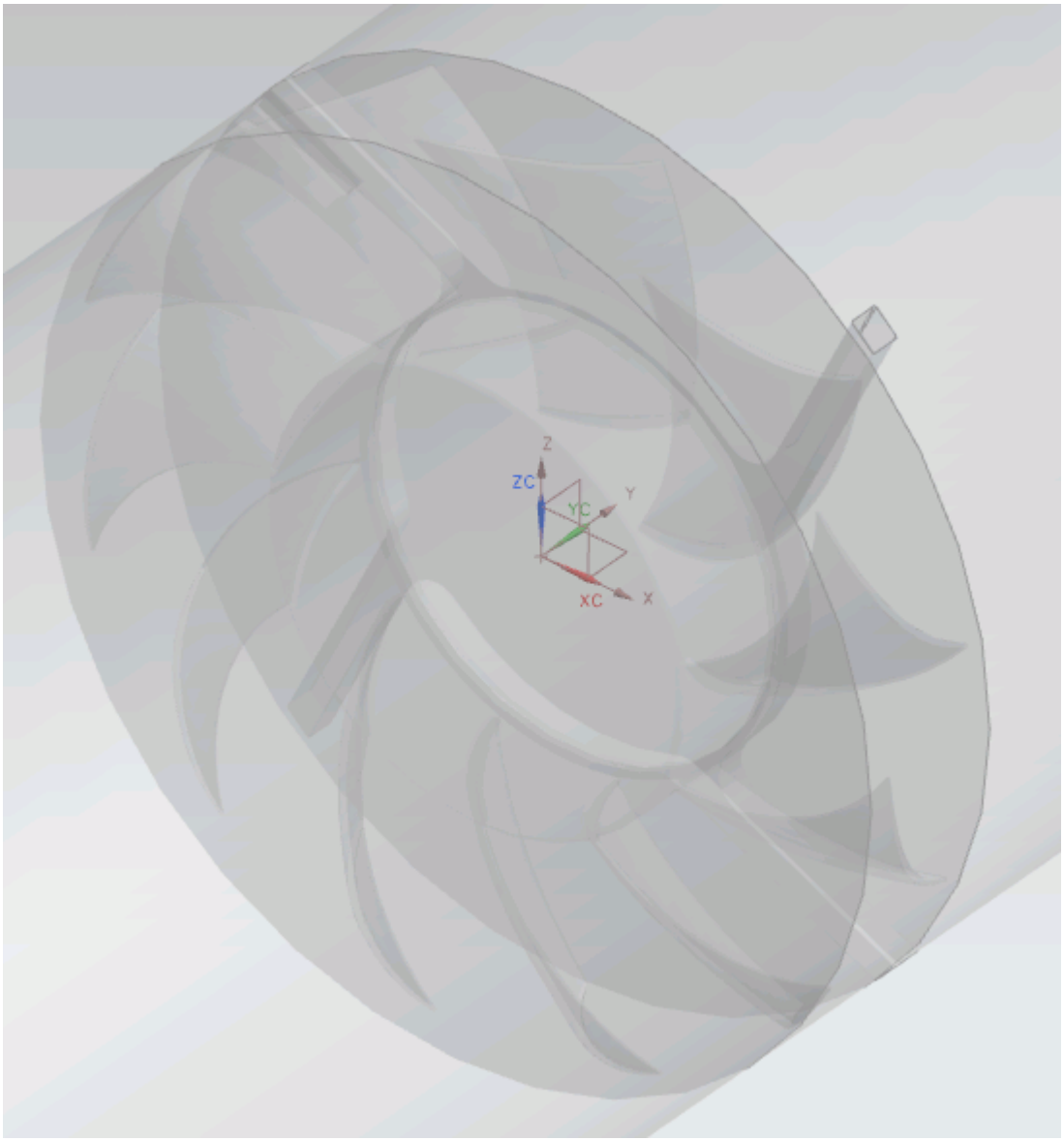


29. Show the Inlet Region and the Outlet Region.
30. Suppress the components that you do not want to take part in the analysis, such as the fan and casing, by right-clicking on these and selecting **Suppression...**.
31. In the *Suppression* dialog that appears set the component to **Always Suppressed**.



32. Click **OK**.

The final fluid domain is shown below.



The geometry is now ready for cad client analysis. Suppressed parts will not be included in the analysis. The three regions that form the fluid domain all have a conformal match, so it will be possible to create the interfaces and obtain a topologically valid mesh in STAR-CCM+ without any further modifications to the geometry.

Imprinting Parts

Most CAD model assemblies are not suitable for meshing as the parts in the assembly are disjointed, where the coincident faces are overlapping. Attempting to generate a mesh with such an assembly will result in a meshing error. To rectify this problem, the parts in an assembly must be imprinted together.

Imprinting can also be used to split surfaces for boundary region applications. An example of such an application would be to model the area that a fan would occupy on the inside of computer case, such as in the third STAR-NX tutorial, where a smaller section of the surface is needed to represent the fan, rather than the entire inner surface of the computer case.

Contents:

Imprinting Parts in an Assembly
Imprinting Surfaces on Parts for Boundary Region Applications

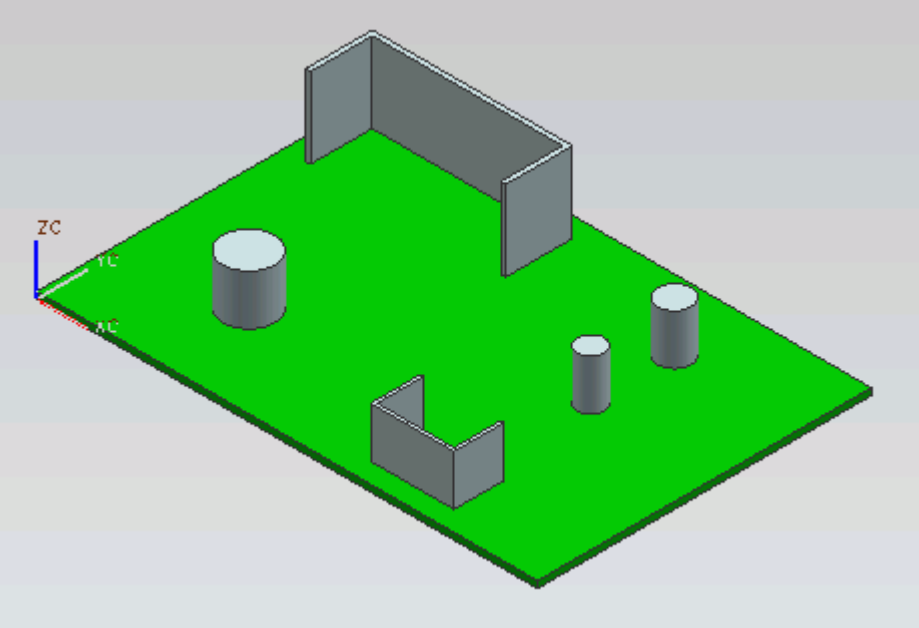
Imprinting Parts in an Assembly

In cases containing multiple adjacent parts, such as in Conjugate Heat Transfer cases, it is essential to imprint the profiles of the coincident faces of the smaller parts onto the face of the larger part. This will ensure that the volume mesh created by STAR-CCM+ contains interfaces between the adjacent parts.

In general, the steps to follow in creating the fluid domain for multiple regions are:

- Work within the assembly, making the part that requires imprinting the work part.
- Create a sketch on the face you want to imprint, and project the contact edges of the other components in the assembly onto this sketch.
- Split the face you want to imprint using the sketch created in the previous step.

The simple circuit board shown below requires imprinting before the mesh can be generated.

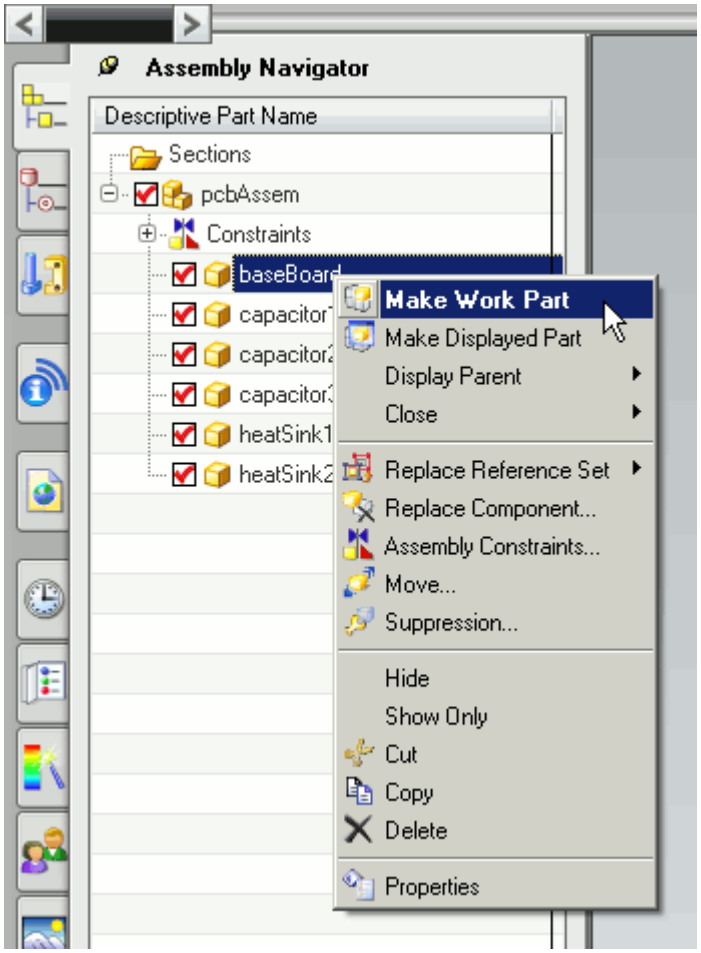


In this example, the capacitors and heat sinks are not imprinted to the circuit board. This would result in a failure when attempting to generate a mesh.

To imprint the capacitors and heat sinks onto the circuit board follow the steps outlined below:

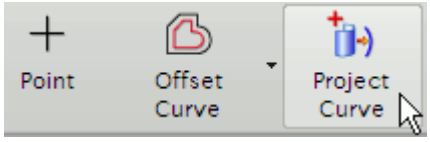
1. Load the assembly file in NX.
2. Make the part that will be imprinted the **Work Part**, right click on the part's node in the feature tree.

   In this example this is the base board.



3. Use the **See-Thru** button from the toolbar to make the other parts translucent.

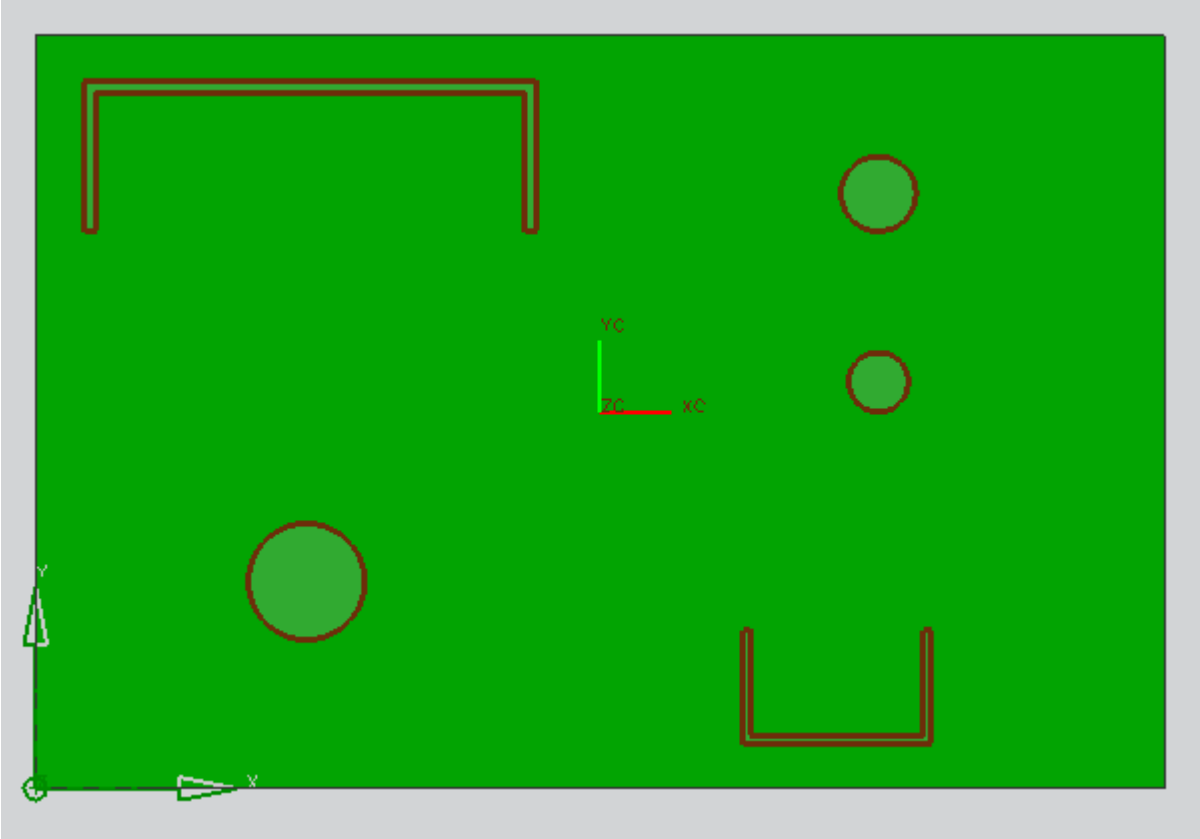
4. Create a new sketch on the surface that requires imprinting.
5. Click on the **Project Curve** button on the toolbar.



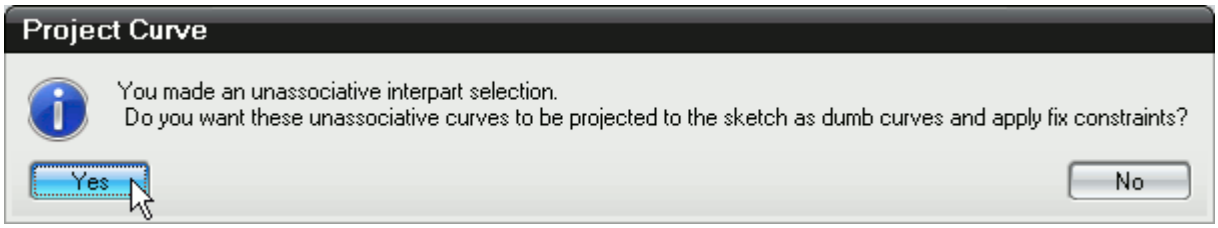
If this button is missing from the toolbar, you may need to click on the (**Open in Sketch Task Environment**) button to open the active sketch in the sketch task environment.

The *Project Curve* dialog will appear.

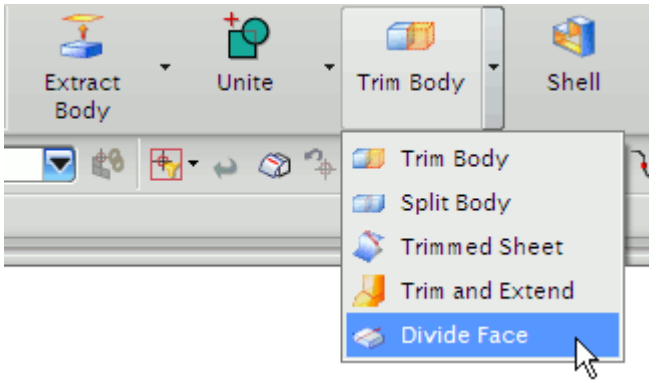
6. Click on the contact face of the other parts to select the contact edges, as shown below.



7. Clicking **OK** in the *Project Curve* dialog will bring up message informing you that the projected edges will not be associative. Click on **Yes** to accept and continue.



8. Click on the **Divide Face** button in the toolbar.

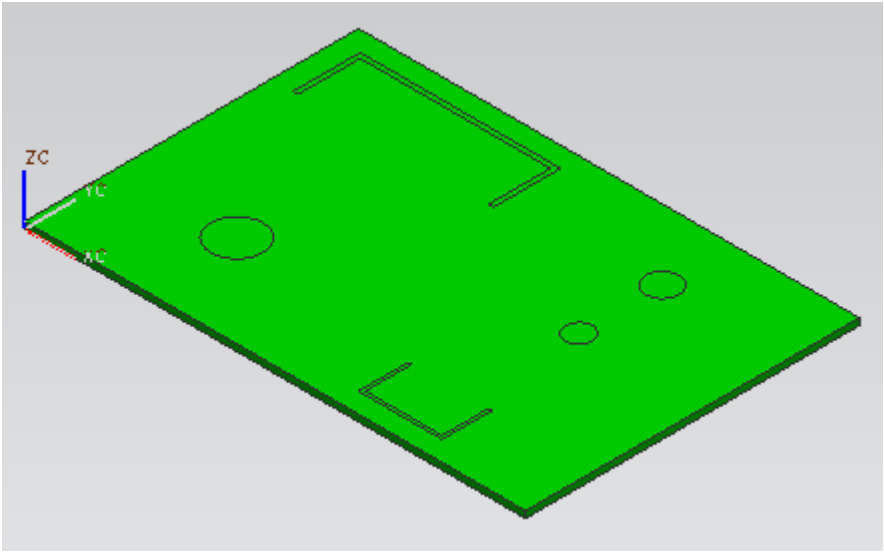


The *Divide Face* dialog will appear.

9. In the *Faces to Divide* box, select the face you want to imprint.
10. In the *Dividing Objects* box, select the sketch that contains the contact edges of the other parts.
11. Ensure that the projection direction (the arrow in the geometry window) is pointing towards the face to be imprinted.

    This can be reversed by clicking on the **Reverse Direction** button in the *Projection Direction* box.
12. Return to the assembly and make the assembly the work part.

The contact edges of the capacitors and heat sinks have now been imprinted onto the base board. This can be verified by the parts in the assembly.



A high quality mesh with fully defined interfaces can now be successfully generated for this assembly. Suppressed parts are not included in the STAR-CCM+ analysis, so for CHT cases, such as the above example, ensure that all relevant parts are active.

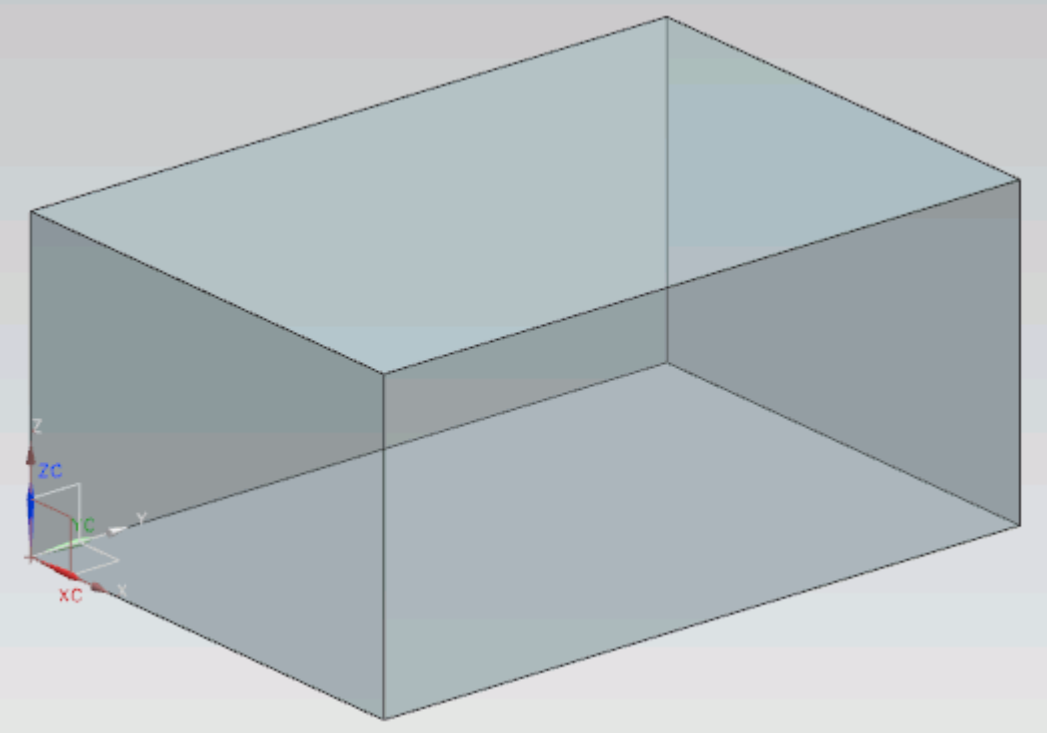
Imprinting Surfaces on Parts for Boundary Region Applications

Faces in a part body can be split into smaller faces for boundary application.

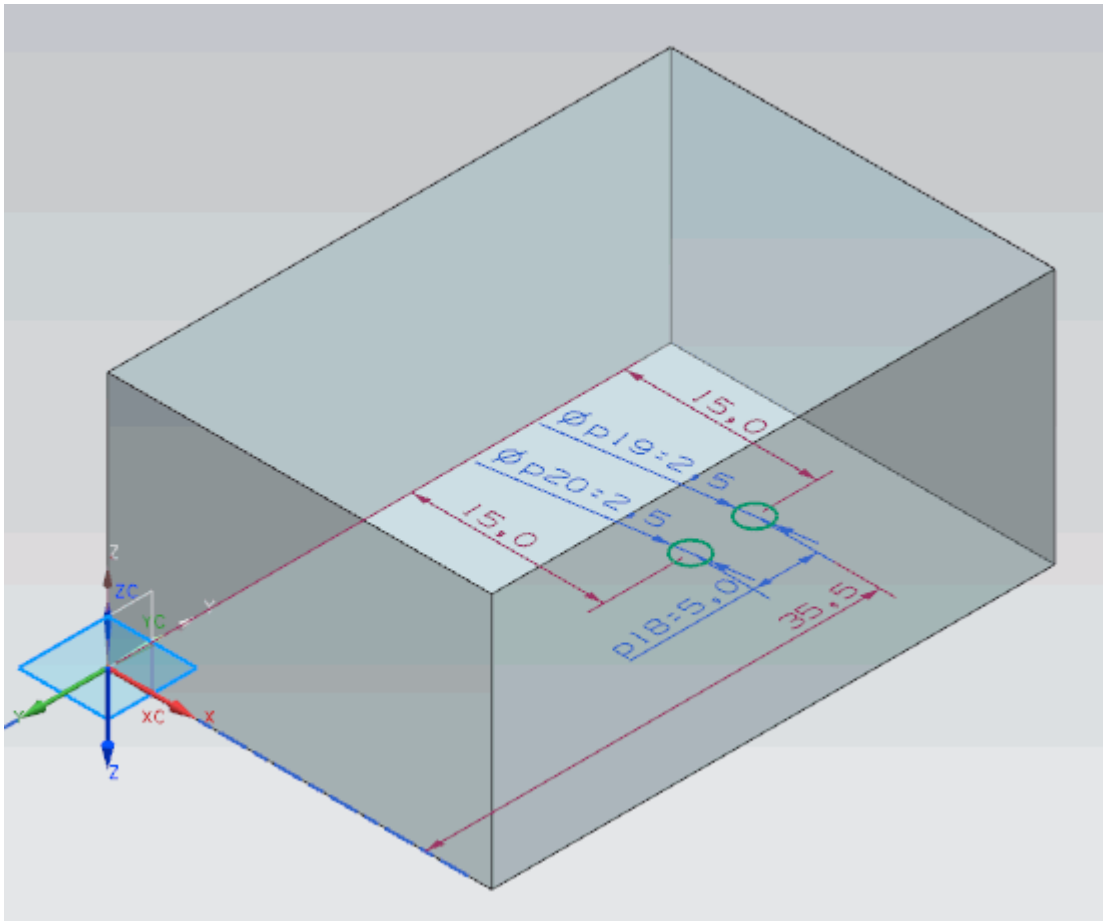
The general procedure to follow is:

- Create a sketch on the surface you want to split. The shape of the sketch should correspond to the shape of the new face after it is split.
- Split the face that requires imprinting with the sketch containing the profiles.

A simple example of an aeration tank is shown below. The image shows the tank, the two air injectors need to be imprinted onto the base of the tank.

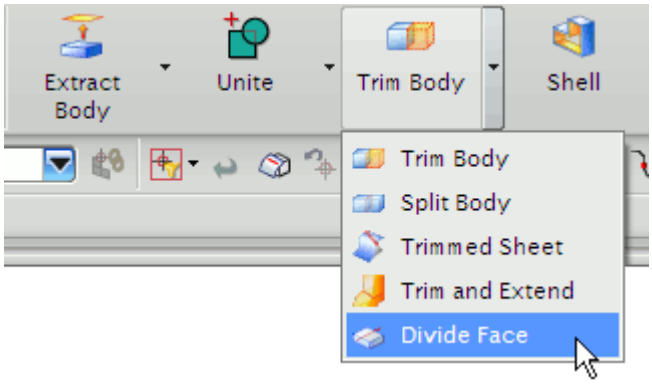


1. Load the part into NX.
2. Create a sketch on the face that requires imprinting.
   In this case the bottom face of the tank needs imprinting.
3. Draw the profile of the boundary you want imprinted.
   In this case, the air injectors are circular in profile.



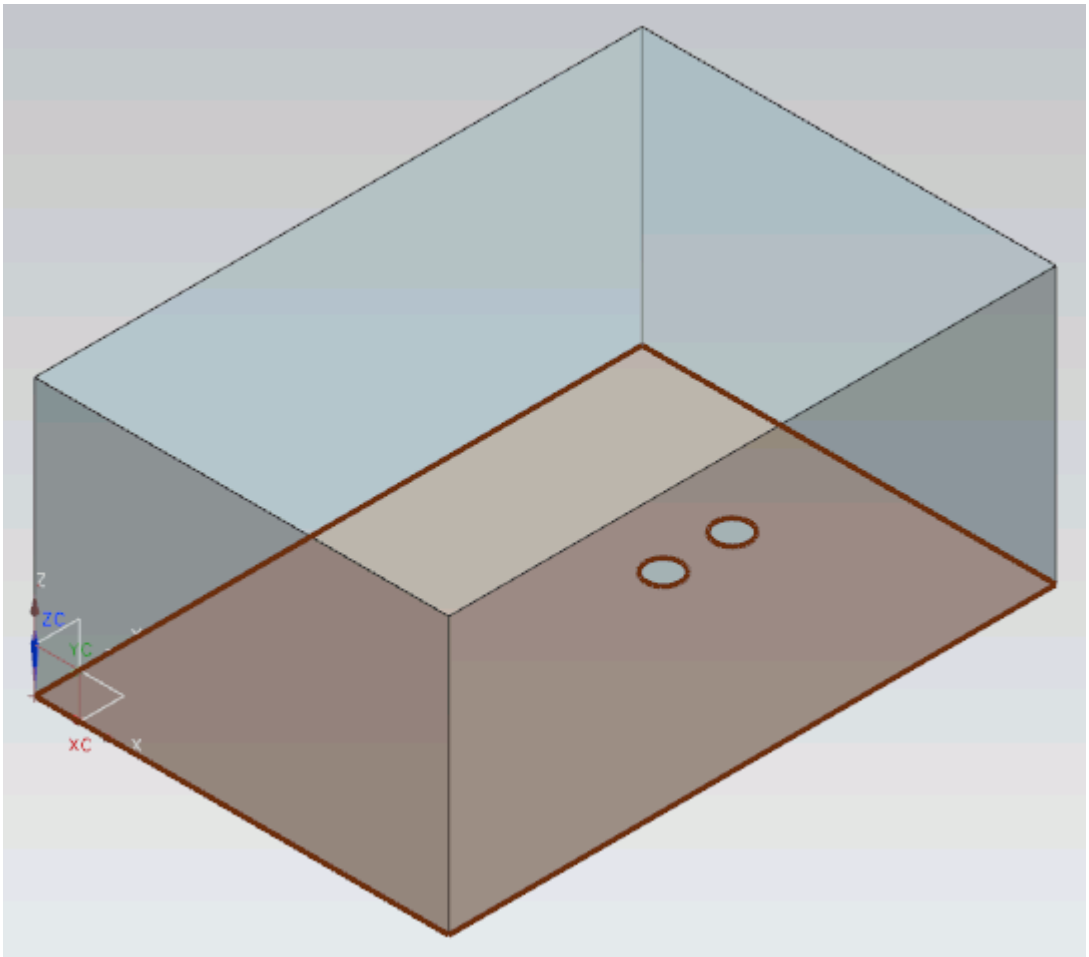
You can use <Shift> + <F8> to change the view so that it is normal to the sketch plane.

4. Use the **Divide Face** function to split the face.



Select the face you want to divide for the *Faces to Divide* field, and the sketch profile(s) you created in the previous step as the *Dividing Objects*.

The face is now imprinted; each face can be selected independently of each other, and can be used when defining boundaries.



The separate faces can now be defined as boundary regions.

STAR-NX Tutorials

Several tutorials are provided for the STAR-NX CAD Client.

These tutorials provide an overview of the major features available in the client and illustrate some of the applications where you can use STAR-NX.

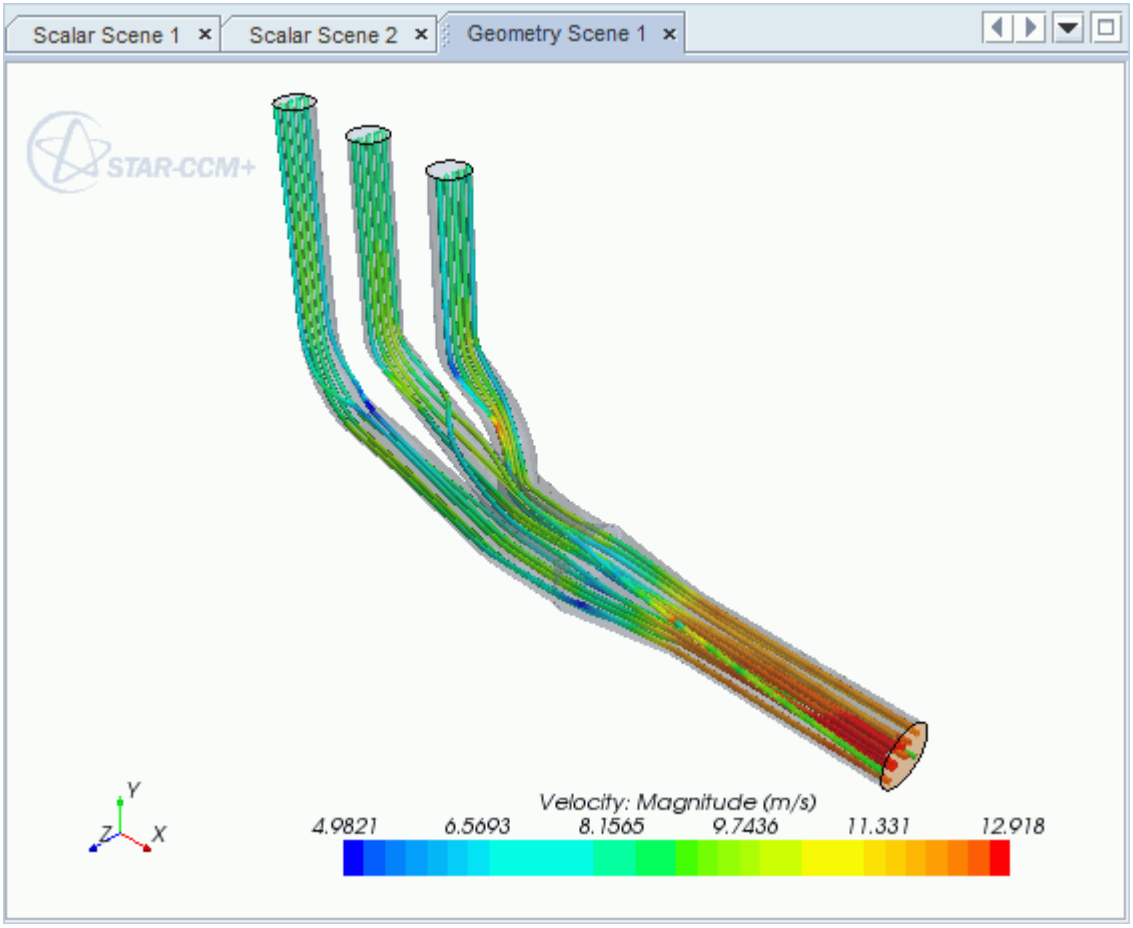
Contents:

Cold Flow Through a Motorcycle Exhaust Manifold
Porous Media Flow
Aircraft Wing Deicing System
Rotating Flow Inside a Graphics Card
External Flow Over a Car
Geometry Transfer: CAD to Simulation

Cold Flow Through a Motorcycle Exhaust Manifold

This tutorial demonstrates how to setup and run a CFD calculation in the STAR-NX environment.

In this tutorial, the model is an exhaust manifold from a three-cylinder motorcycle engine.



Contents:

Opening the CAD Model
Selecting Fluid Physics Models
Defining Boundary Conditions
Setting the Mesh Pipeline
Modifying Solution Controls
Reviewing the Simulation Set Up
Generating the Surface Mesh
Generating the Volume Mesh
Building and Running
Post-Processing Results

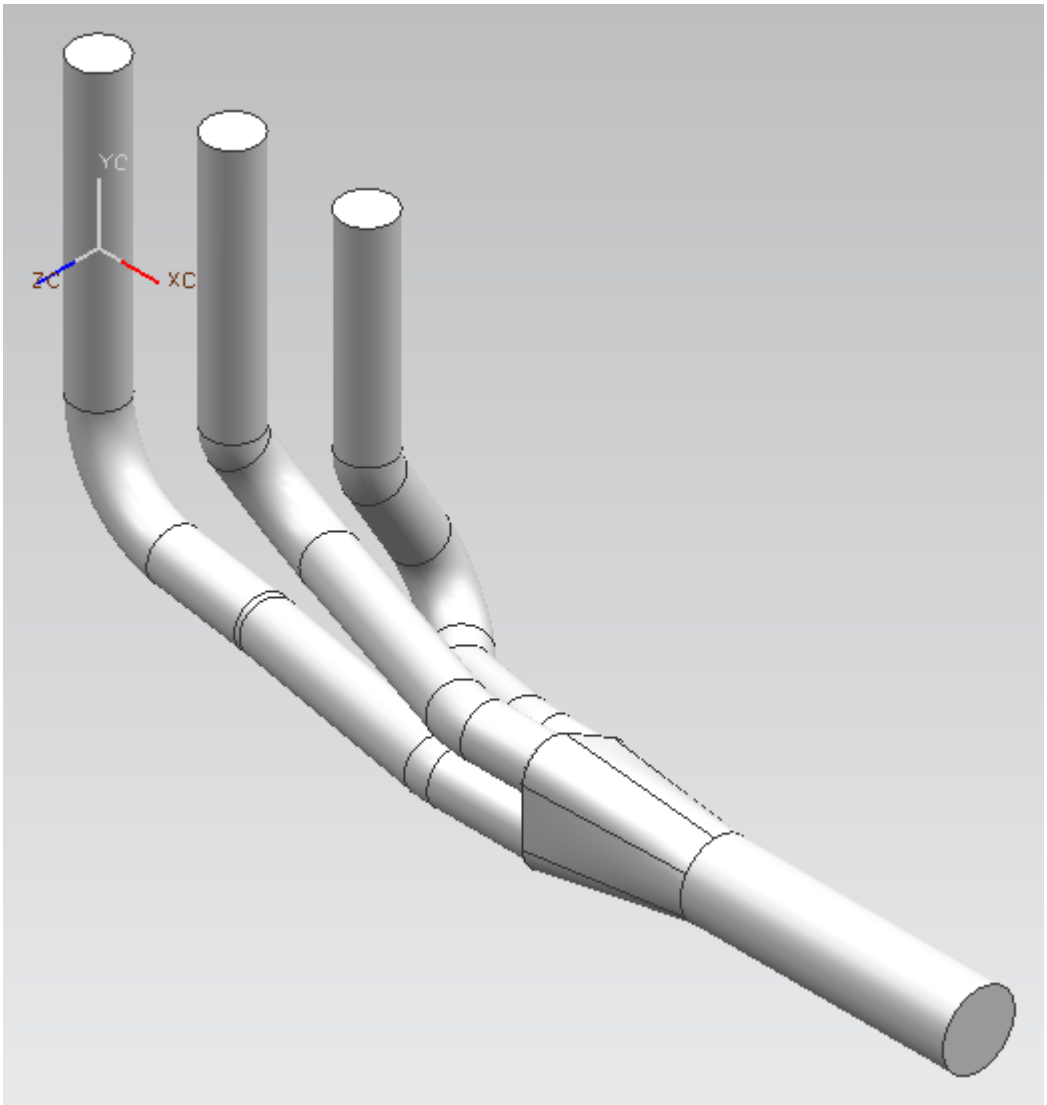
Opening the CAD Model

A directory containing the CAD geometry is provided with the STAR-CCM+ distribution.

This directory is located in `[STAR-CCM+_INSTALL_DIR]\doc\startutorialsdata\starnx\data`.

1. Copy directory `data` to your own disk space and open the part file `exhaustManifold.prt` in STAR-NX.

   The imported geometry is shown below.



The STAR-NX ribbon appears at the top of the screen.The STAR-NX object tree is located on the left of the screen.

To switch to this object tree:

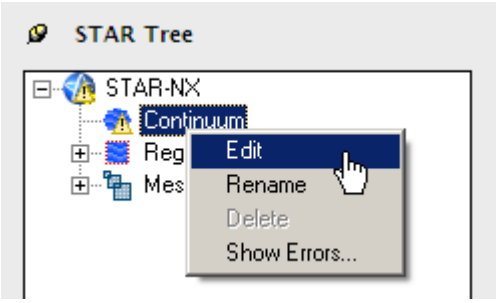
2. Click (**STAR Tree**).

If these items are not visible, refer to Launching STAR-NX Manually.

Selecting Fluid Physics Models

The first step in setting up this tutorial is to define the physics of the problem. For this case, the flow is turbulent and isothermal.

1. Right-click the **Continuum** node in the STAR-NX tree and select **Edit**.



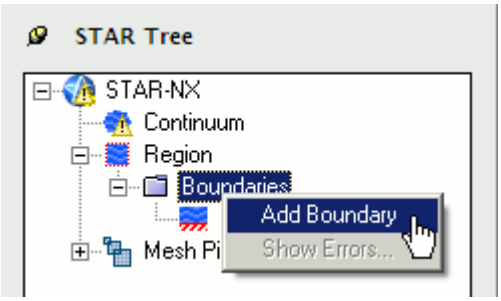
   The *Continuum Settings* dialog appears.
2. Deactivate the **Temperature** option in the *Optional Physics Models*.
3. Click **OK**.

Defining Boundary Conditions

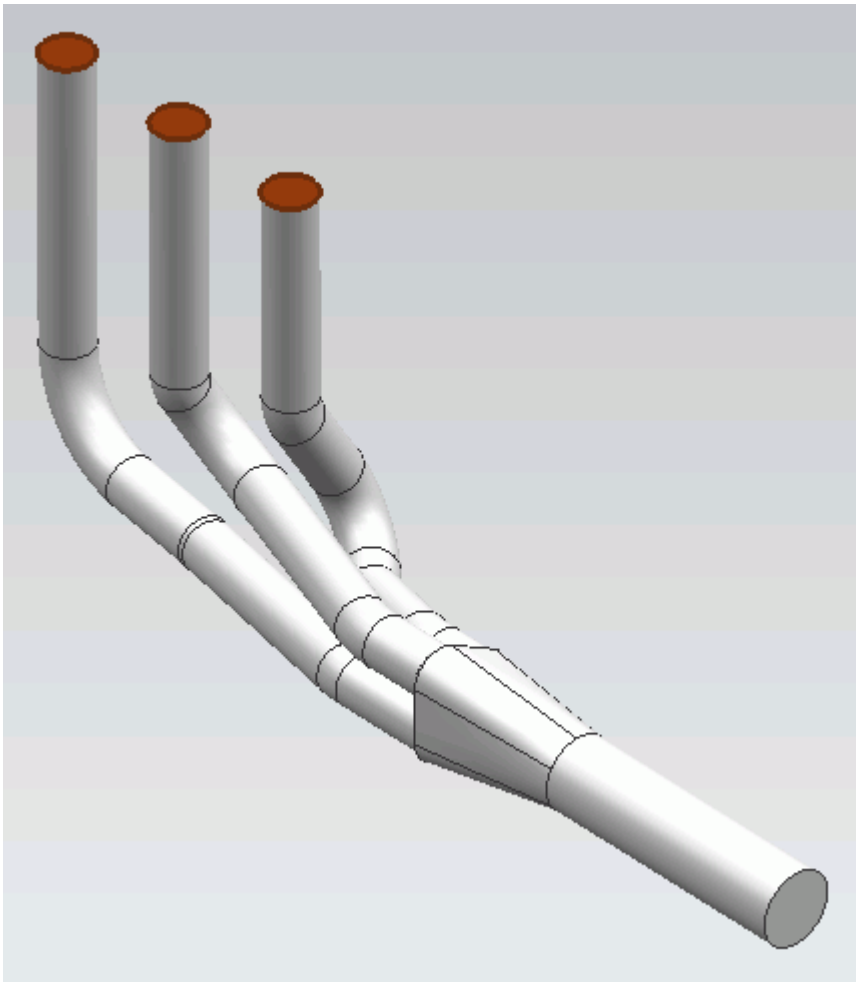
You define the inlet and outlet boundary conditions for the simulation.

Create the inlet boundary first:

1. Right-click the **Region** > **Boundaries** node and select **Add Boundary**.



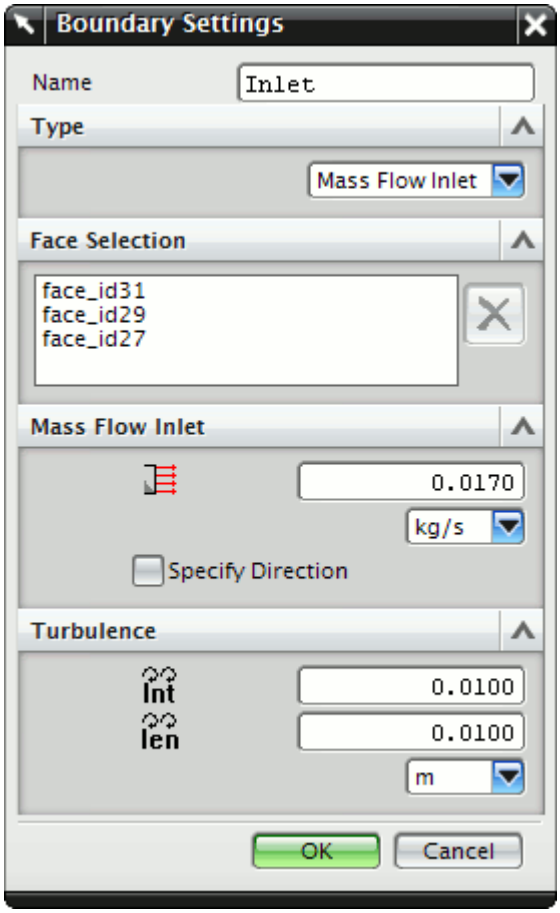
   The *Boundary Settings* dialog appears.
2. On the CAD model, select the three faces that are highlighted below.



These selections are reported in the *Face Selection* box in the *Boundary Settings* dialog.

3. Set *Type* to **Mass Flow Inlet**.
4. Set *Boundary Name* to `Inlet`.
5. Set (*Mass Flow Rate*) to `0.017` kg/s.

   The completed dialog is shown below.



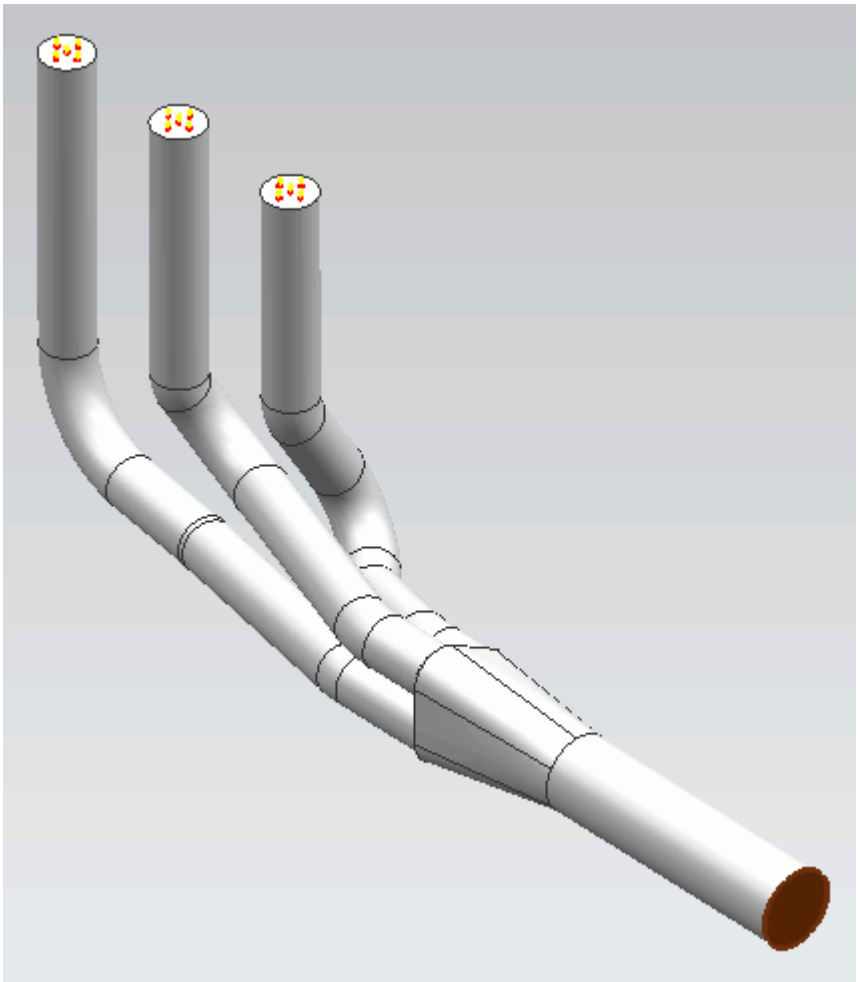
6. Click **OK**.

Create the outlet boundary:

7. Right-click the **Region** > **Boundaries** node and select **Add Boundary**.

   The *Boundary Settings* dialog appears.

8. Select the face that is highlighted below.



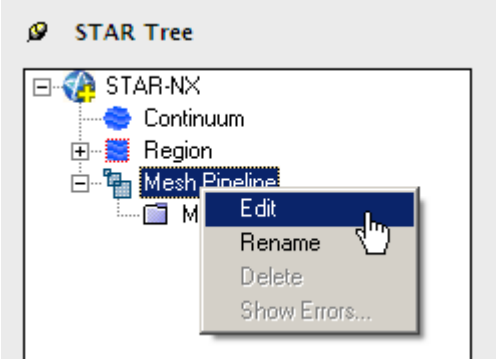
9. Set *Type* to **Pressure Outlet**.
10. Set *Boundary Name* to `Outlet`.

    The default settings are acceptable for this case.
11. Click **OK**.

Setting the Mesh Pipeline

To obtain more accurate results, modify the mesh pipeline:

1. Right-click the **Mesh Pipeline** node in the STAR-NX tree and select **Edit**.



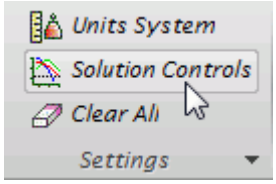
The *Mesh Pipeline Settings* dialog appears.

2. Set (*Base Size*) to `2.5`%.
3. Click to expand the *Prism Layer* tab.
4. Set (*Prism Layer Thickness*) to `15`%.
5. Click **OK**.

Modifying Solution Controls

Modify the maximum number of iterations and the solution accuracy:

1. Click **Solution Controls** on the STAR-NX ribbon.

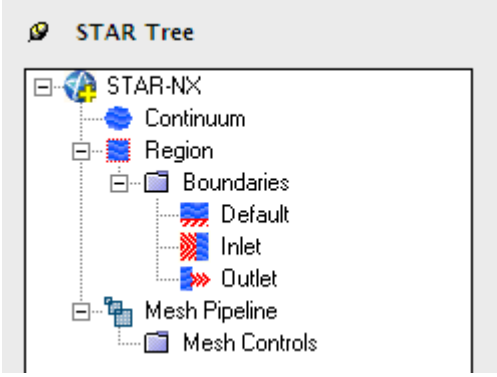


The *Solution Controls* dialog appears.

2. Click to expand the *Advanced* tab.
3. Set (*Number of Iterations*) to `200`.
4. Set (*Convergence Level*) to `0.0001`.
5. Click **OK**.
6. Save your progress.

Reviewing the Simulation Set Up

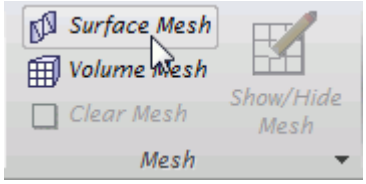
The final STAR-NX object tree is shown below.



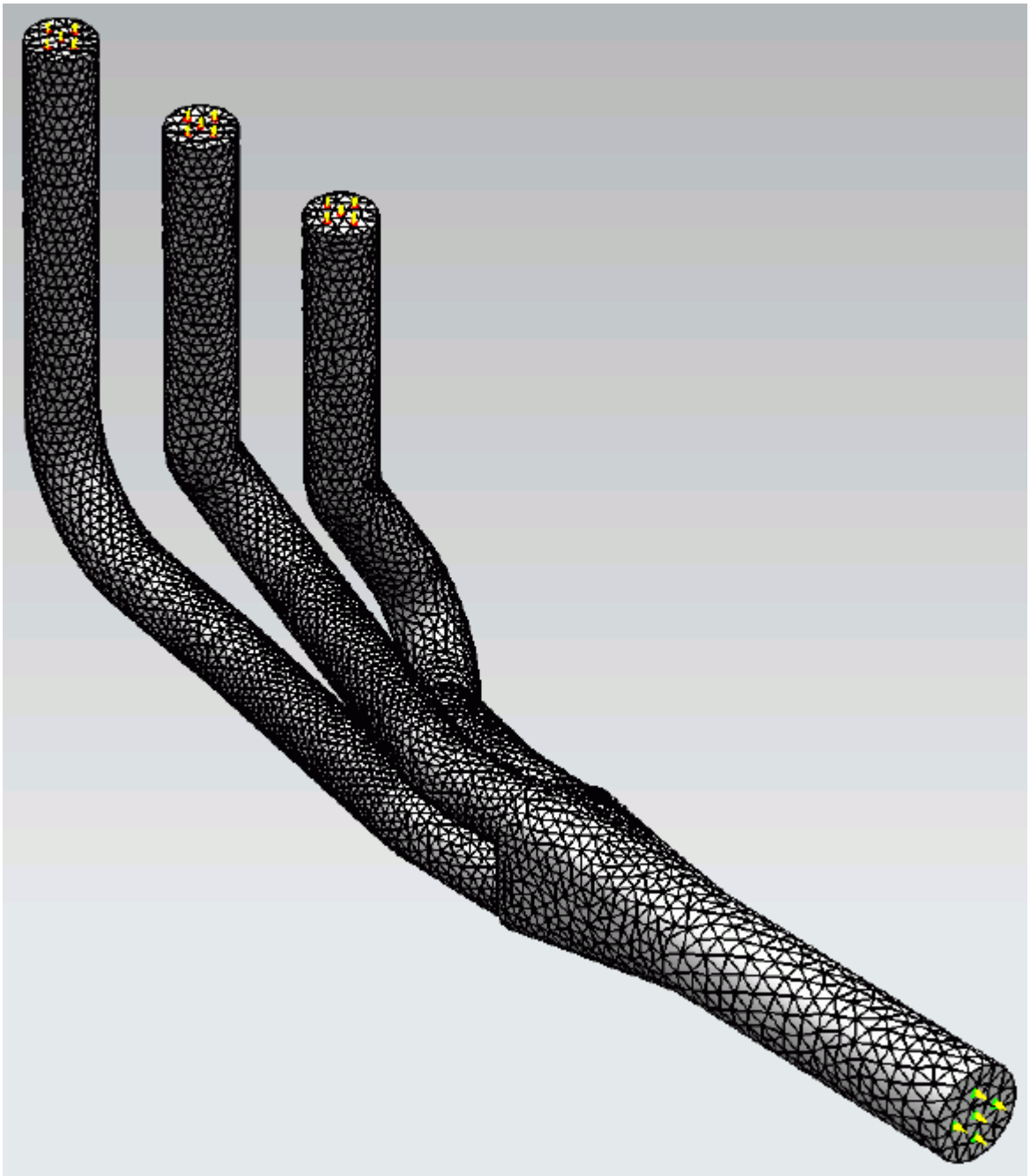
Generating the Surface Mesh

To generate the surface mesh:

1. From the STAR-NX ribbon, click **Surface Mesh**.



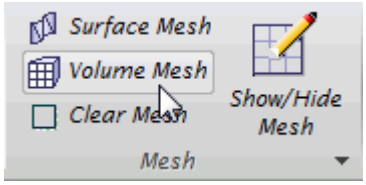
The surface mesh is shown below.



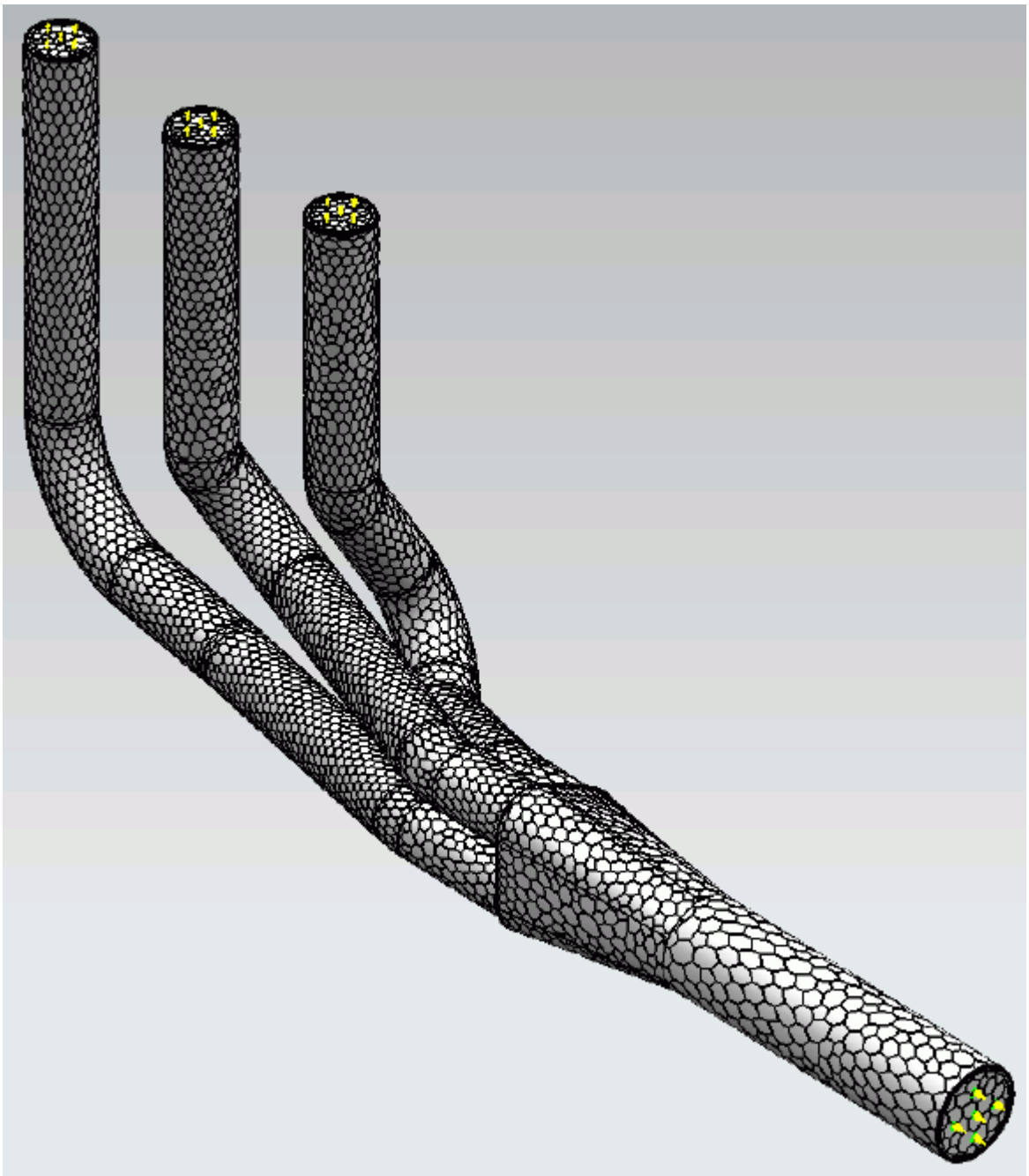
Generating the Volume Mesh

Generate the volume mesh:

1. From the STAR-NX ribbon, click **Volume Mesh**.



The volume mesh is shown below.

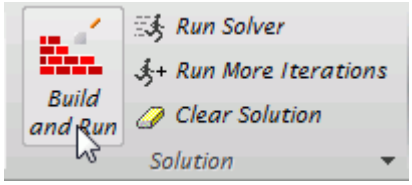


Building and Running

The case is now ready to run.

To run the simulation:

1. From the ribbon, click **Build and Run**.



The *Residuals* plot appears in the STAR-NX *Output* window. You can use this plot to monitor the convergence of the solution.

When the solver is done, STAR-CCM+ launches.

Post-Processing Results

You perform all post-processing in STAR-CCM+. By default, a scalar scene, *Scalar Scene 1*, automatically appears in the *Graphics* window when the solver finishes. This scene displays the absolute total pressure.

The results that are contained in the following sections are representative. You may see some variation in the results that you obtain, depending on what platform and package you are using.

1. In STAR-CCM+, click **File** > **Save** to save the simulation.

Contents:

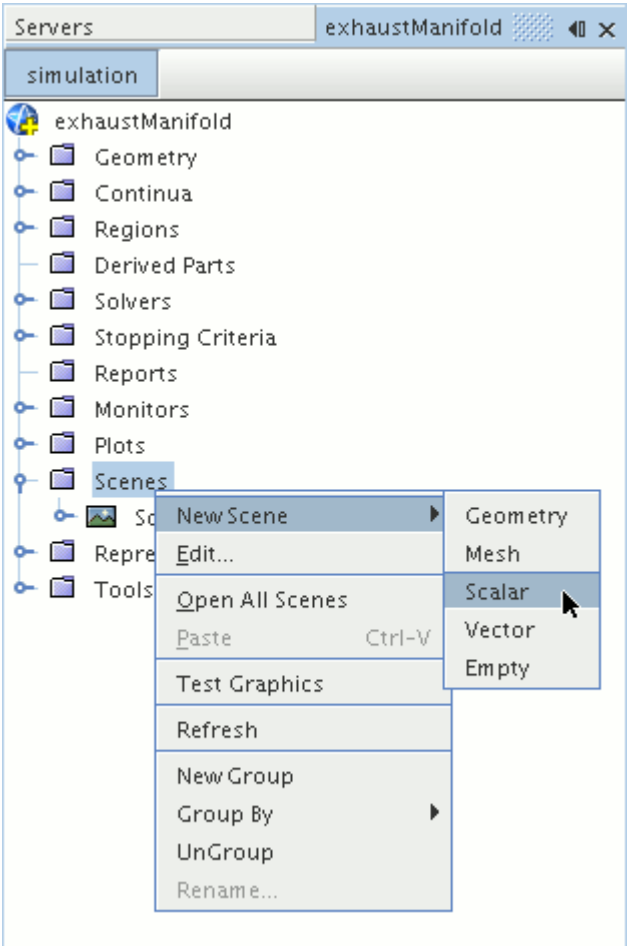
Creating a Scalar Scene
Creating A Streamlines Scene
Creating a Vector Scene
Creating a Streamlines Scene
Creating A Scalar Scene
Visualizing the Wing Skin Temperature
Creating a Streamlines Scene
Creating A Scalar Scene
Creating a Streamlines Scene
Creating a Scalar Scene
Modifying The Scalar Scene
Creating a Streamlines Scene

Creating a Scalar Scene

Create a scalar scene that displays the pressure on a cross-section of the exhaust manifold.

To create the scalar scene:

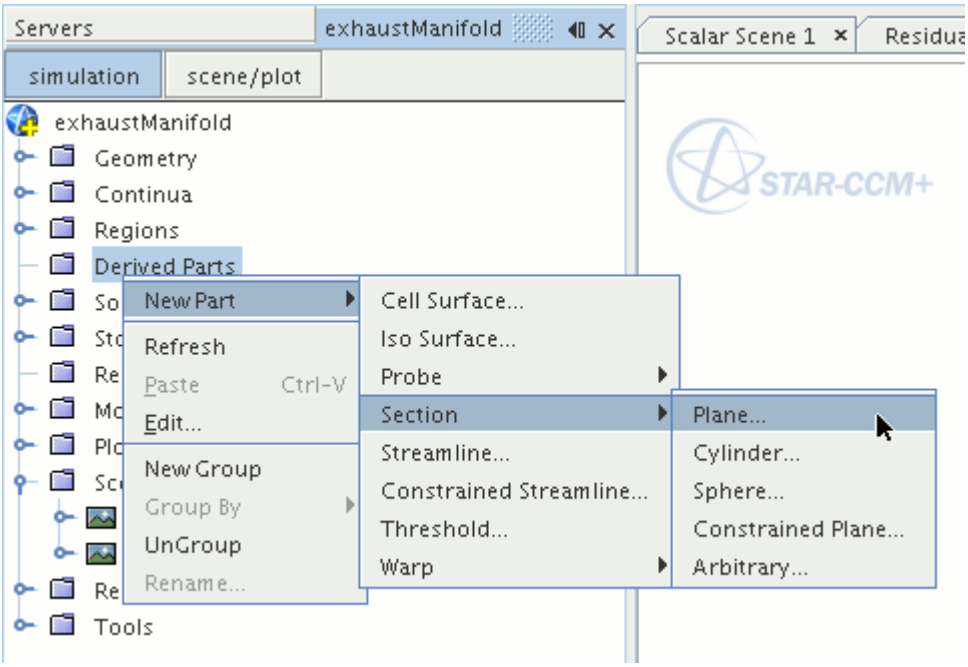
1. Right-click the **Scenes** node in the explorer pane and select **New Scene** > **Scalar**.



A new scalar scene, *Scalar Scene 2*, is created and selected in the *Graphics* window.

Create the section plane:

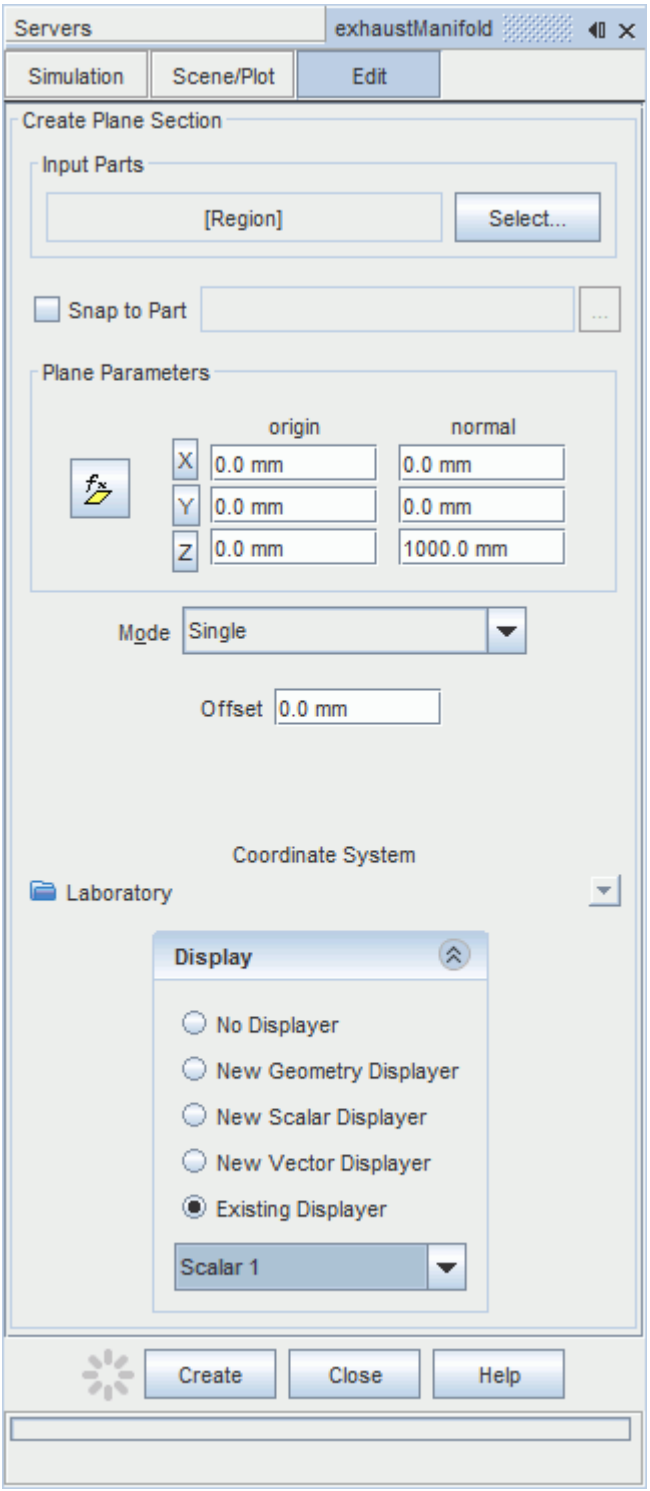
2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane...**



The *Create Plane Section* dialog appears.

3. In the *Plane Parameters* box, set the *origin* vector to `[0 mm, 0 mm, 0 mm]`
4. Set the *normal* vector to `[0 mm, 0 mm, 1000 mm]`
5. In the *Display* group box, select **Existing Displayer** and in the drop-down menu select **Scalar 1**.

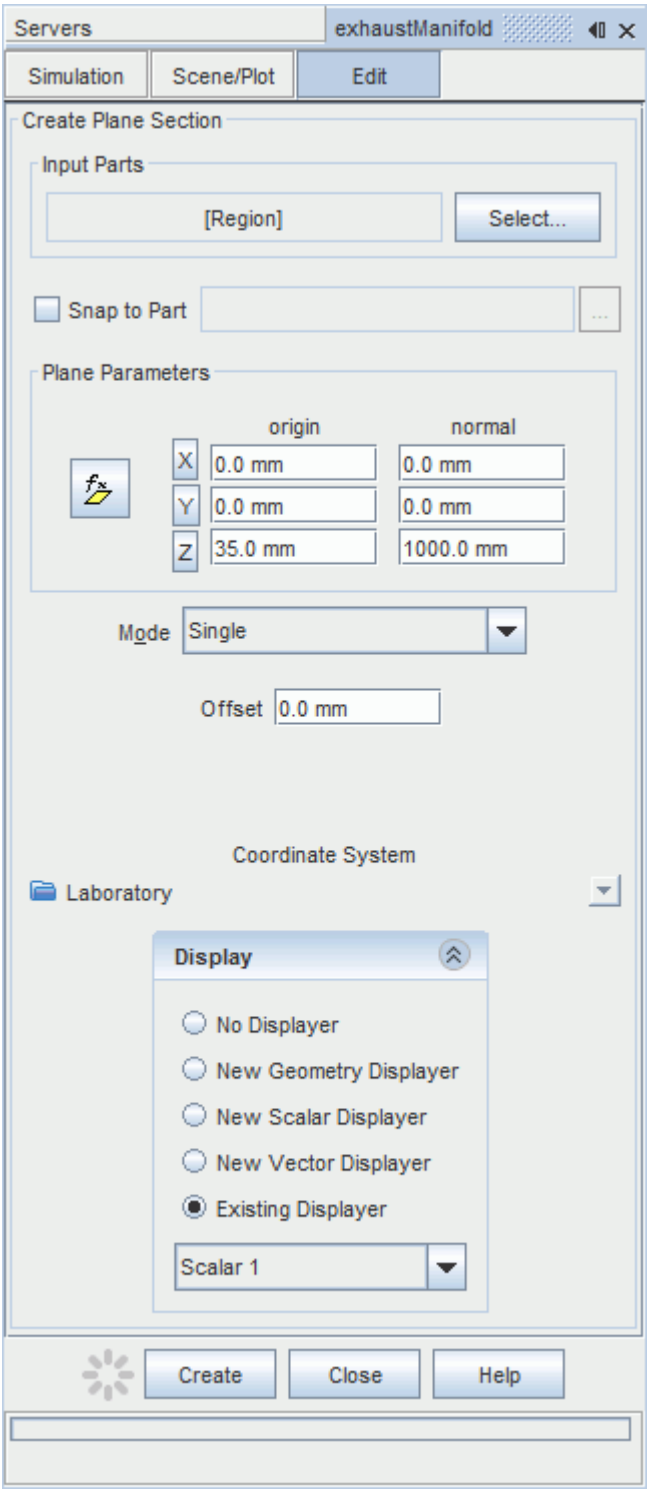
   The completed dialog is shown below.



6. Click **Create**.

Create the second section plane using the same dialog:

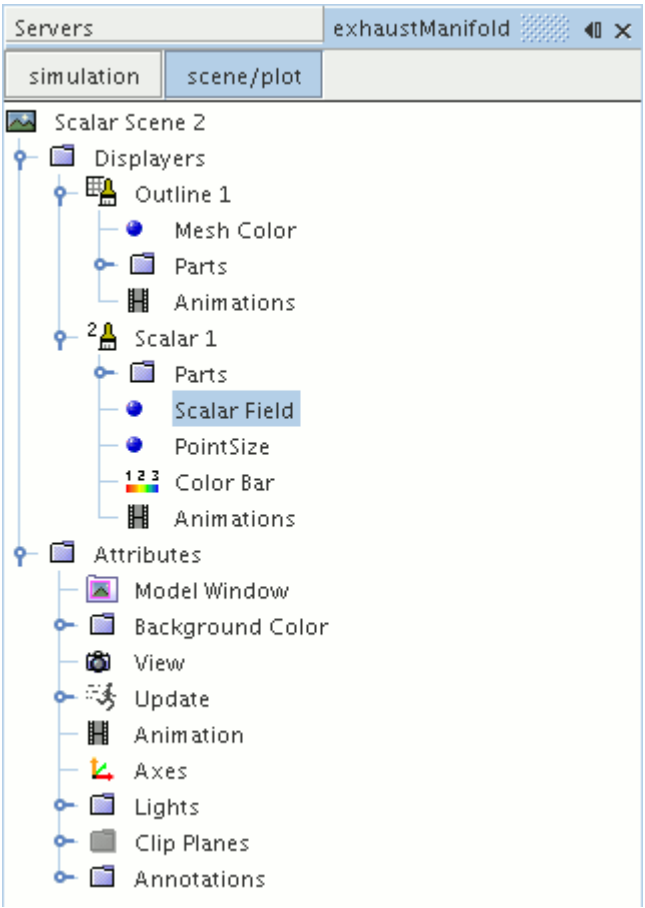
7. In the *Plane Parameters* box, set *origin* to `[0 mm, 0 mm, 35.0 mm]` and leave *normal* as `[0 mm, 0 mm, 1000 mm]`.



8. Click **Create** and then **Close**.

Define the scalar field to display the pressure contours:

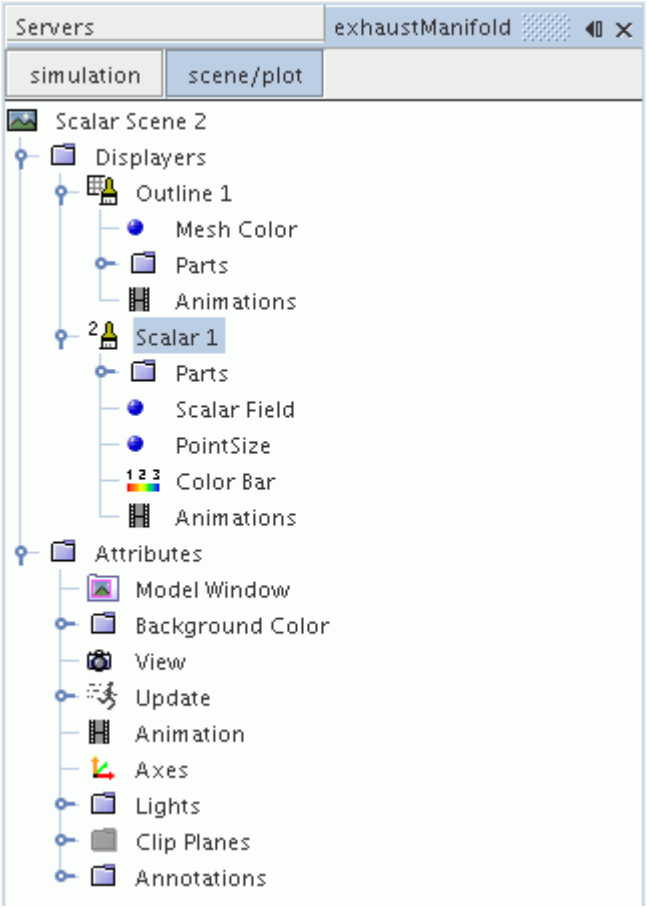
9. Click the **scene/plot** button in the explorer pane.
10. Select the **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Scalar Field** node.



11. Set *Function* to **Pressure**.

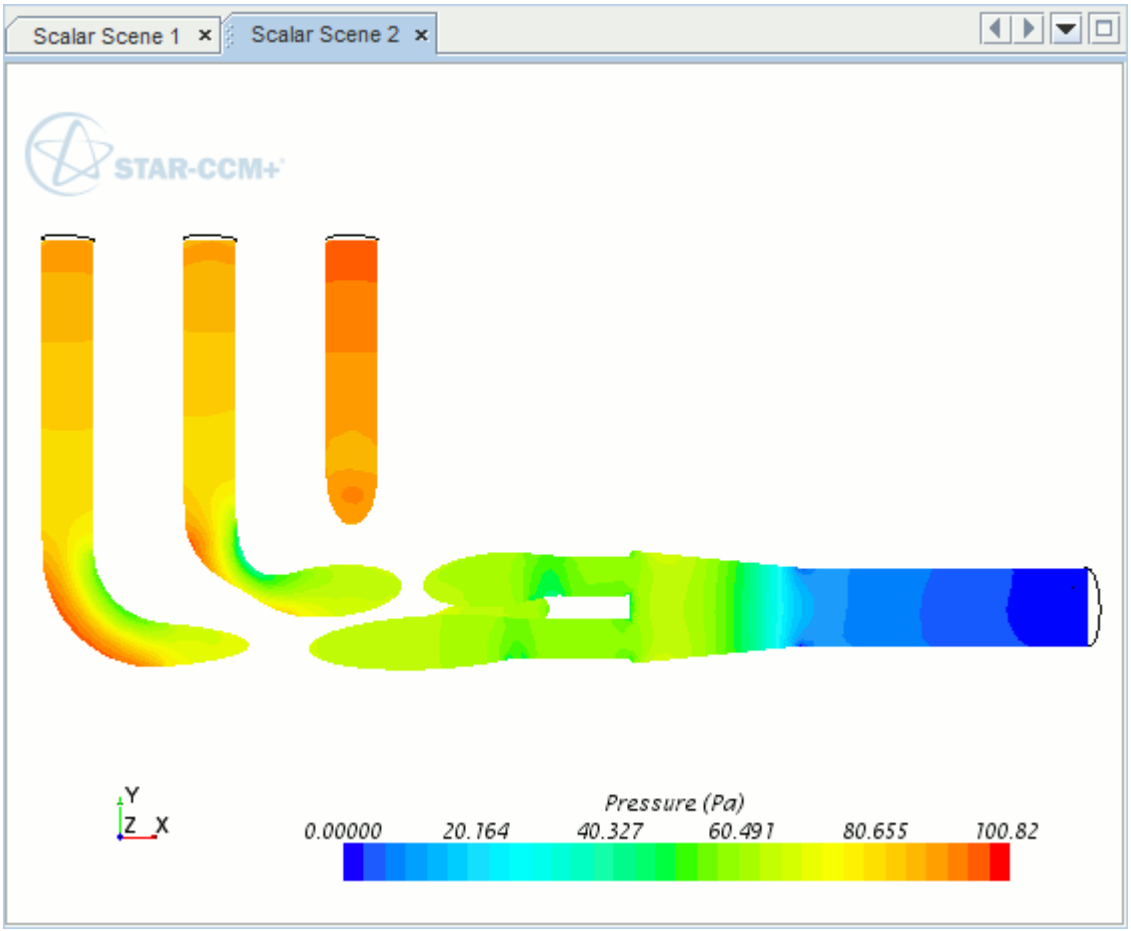
The pressure plot shows the pressure drop inside the exhaust manifold. In order to get a smooth transition between the different pressure values, set the contour style to smooth:

12. Select the **Scalar Scene 2** > **Displayers** > **Scalar 1** node.



13. Set *Contour Style* to **Smooth Filled**.
14. Make sure that the geometry is positioned on its side by pressing the <T> key while *Scalar Scene 2* is active.
15. Click the **simulation** button to return to the main STAR-CCM+ object tree.
16. Click (**Save**) to save the simulation .

The finished scene is shown below.

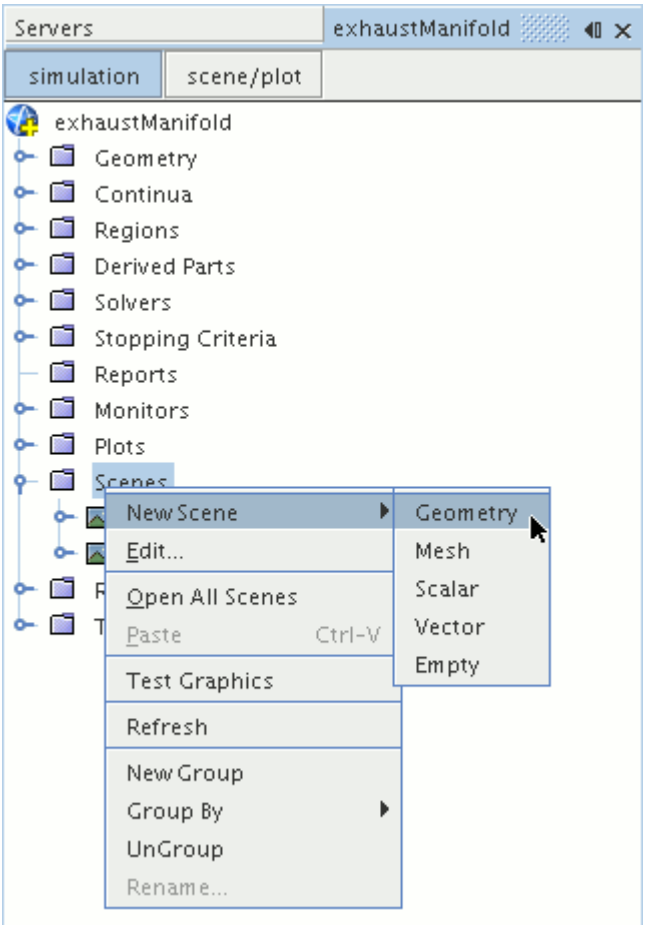


Creating A Streamlines Scene

Create a streamlines scene to get a better understanding of the flow inside the exhaust manifold.

First, create a geometry scene:

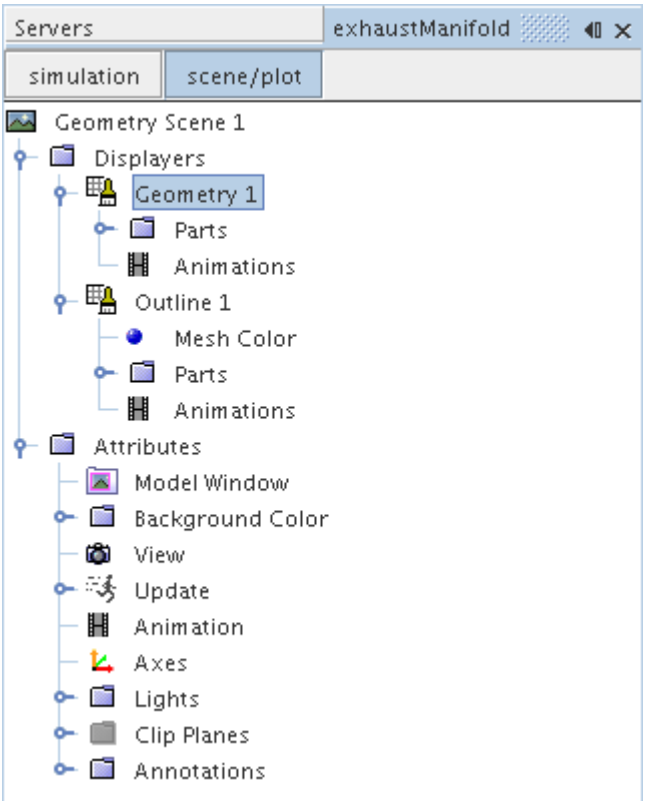
1. Right-click the **Scenes** node in the object tree and select **New Scene** > **Geometry**.



A new geometry scene is created and selected in the *Graphics* window.

Next, modify the opacity of the geometry model in the scene:

2. Click **scene/plot**.
3. Select the **Geometry Scene 1** > **Displayers** > **Geometry 1** node.



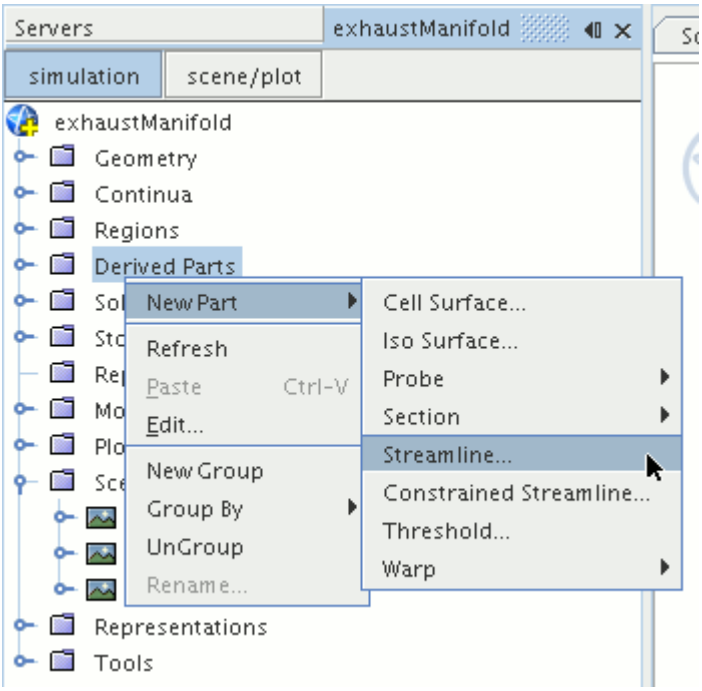
4. Set *Opacity* to `0.3`
5. Click **simulation**.

Change the view of the scene to something more suitable:

6. On the toolbar, click (**Save-Restore-Select views**) and select **Views** > **+X +Y +Z** > **Up +Y**.

Create the streamlines:

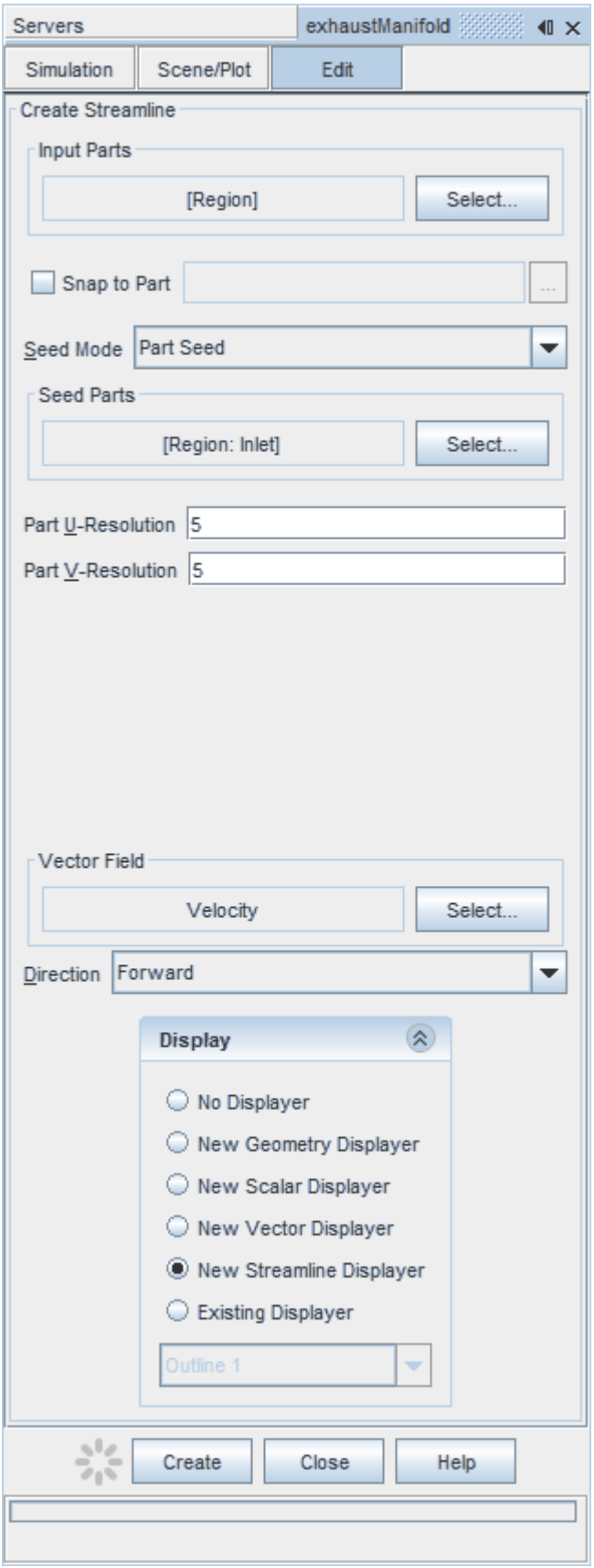
7. Right-click the **Derived Parts** node and select **New Part** > **Streamline...**.



The *Create Streamline* dialog appears.

8. Make sure that **Part Seed** is selected in the *Seed Mode* section.
9. Set *Seed Parts* to **Regions** > **Region** > **Boundaries** > **Inlet**.
10. Make sure that **New Streamline Displayer** is selected in the *Display* group box.

    The completed dialog is shown below.


Servers
exhaustManifold
Simulation
Scene/Plot
Edit
Create Streamline
Input Parts
[Region]
Select...
Snap to Part
Seed Mode
Part Seed
Seed Parts
[Region: Inlet]
Select...
Part U-Resolution
5
Part V-Resolution
5
Vector Field
Velocity
Select...
Direction
Forward
Display
No Displayer
New Geometry Displayer
New Scalar Displayer
New Vector Displayer
New Streamline Displayer
Existing Displayer
Outline 1
Create
Close
Help

11. Click **Create** to create the streamlines and add them to the geometry scene as part of a streamline displayer.
12. Click **Close** to close the *Create Streamline* dialog.

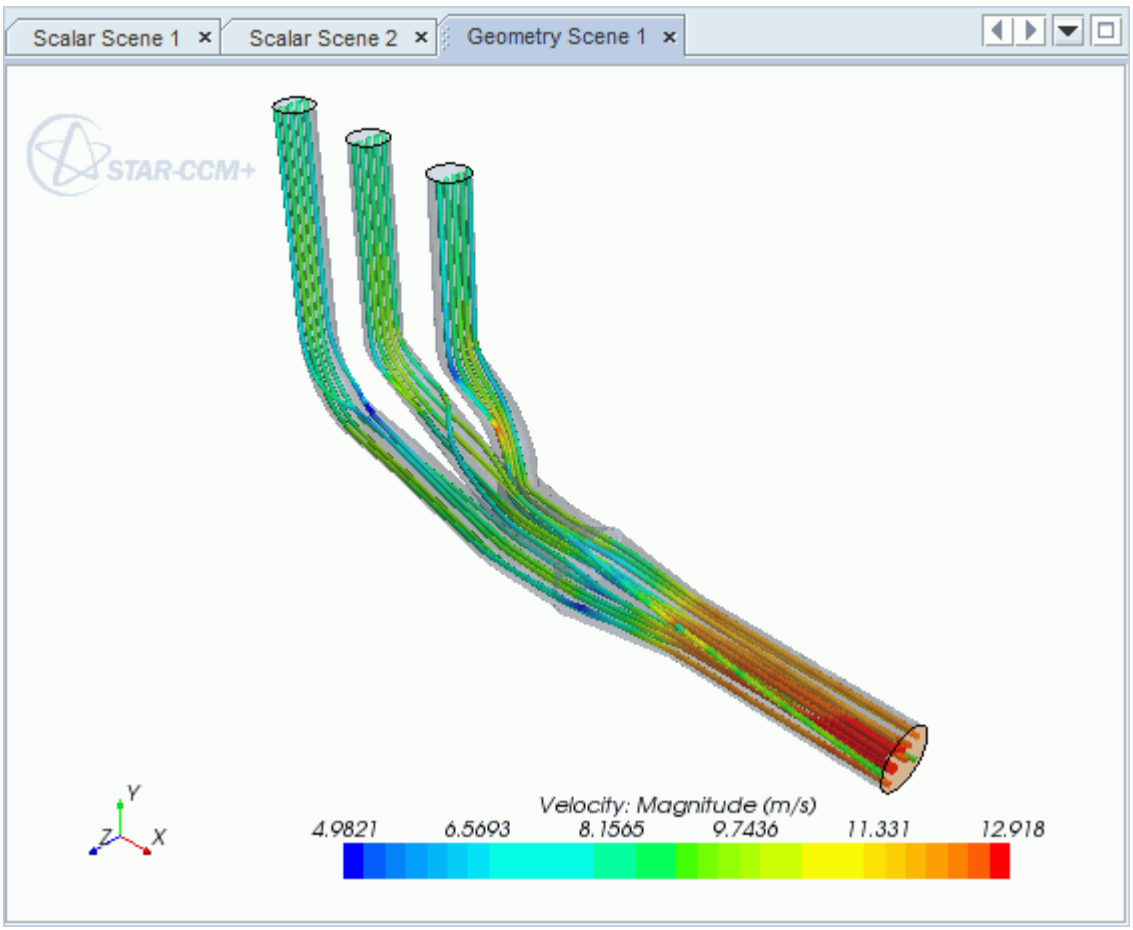
Change the display method of the streamlines to tubes instead of the default lines:

13. Click **scene/plot**.
14. Select the **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** node.
15. Set *Mode* to **Tubes** and *Width* to `0.0020`.

Change the color of the streamlines to represent velocity magnitude:

16. Select the **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** > **Scalar Field** node.
17. Set *Function* to **Velocity** > **Lab Reference Frame** > **Magnitude**.
18. Save the simulation.

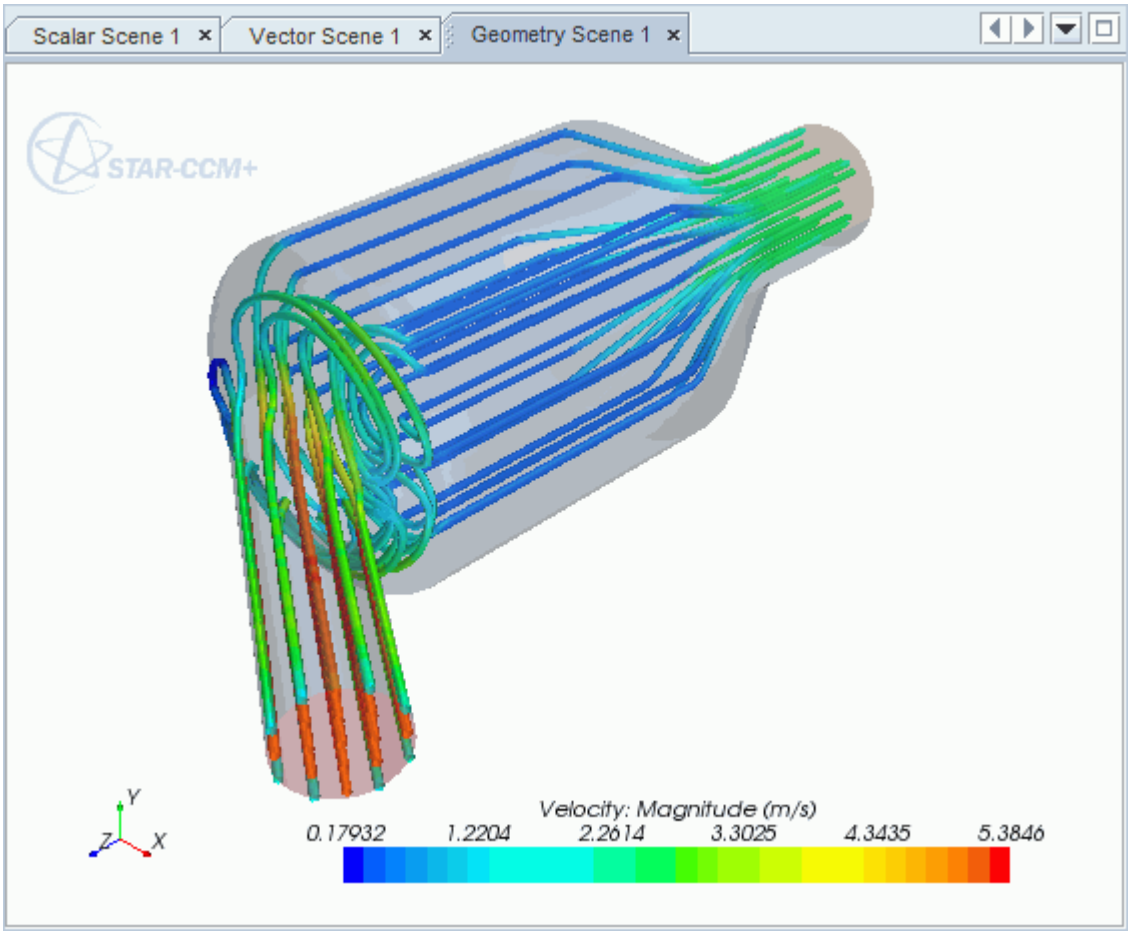
The finished scene is shown below.



Porous Media Flow

This tutorial takes you through the steps that are required to run a CFD calculation in the STAR-NX environment.

In this tutorial, the model is an idealized catalyst.



Contents:

Opening the CAD Model
Selecting Fluid Physics Models
Setting the Porous Region
Defining Boundary Conditions
Setting the Mesh Pipeline
Modifying Solution Controls
Reviewing the Simulation Set Up
Generating the Surface Mesh
Generating the Volume Mesh

Building and Running
Post-Processing Results

Opening the CAD Model

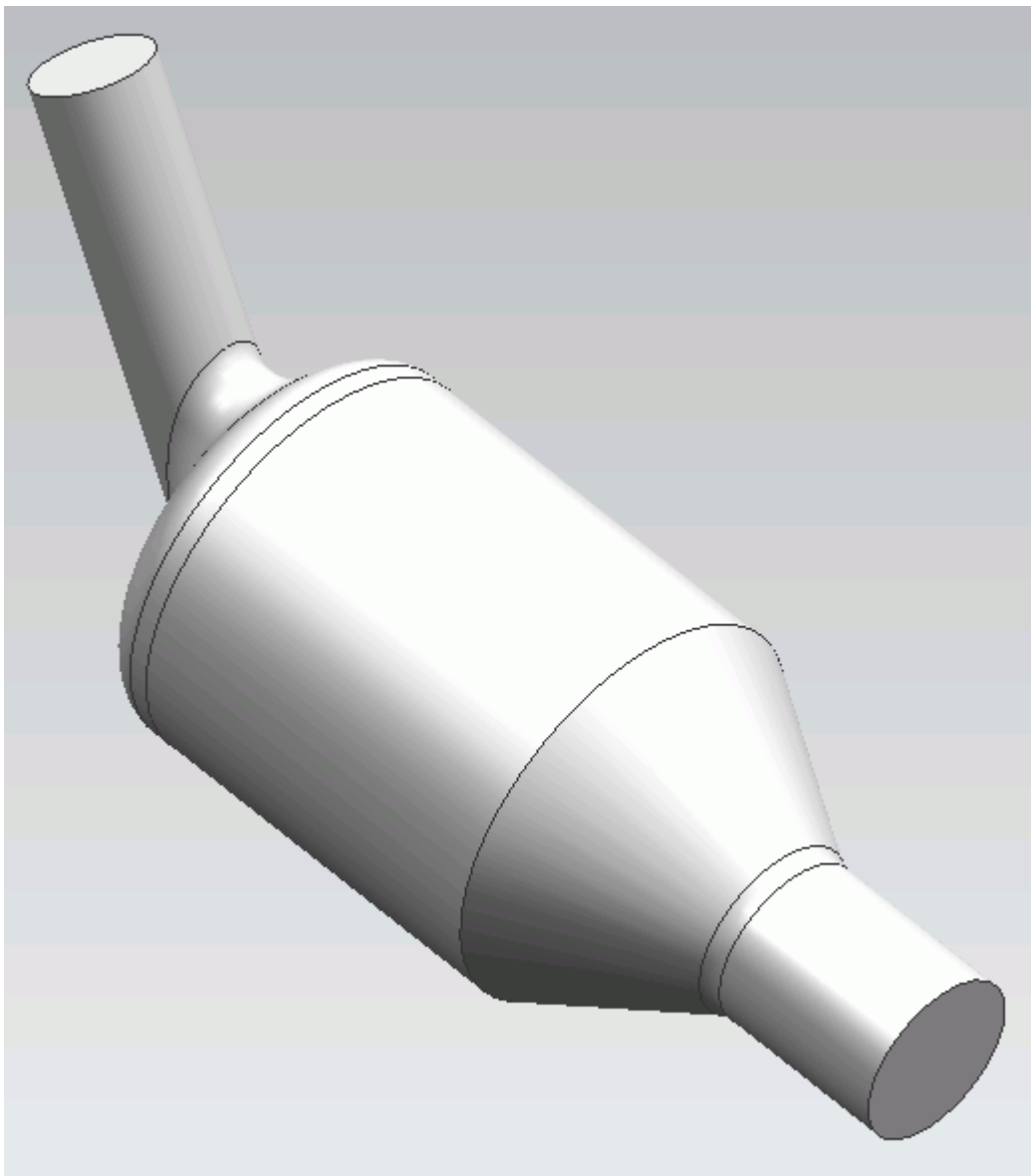
A directory containing the CAD geometry is provided with the STAR-CCM+ distribution.

This directory is located in `[STAR-CCM+_INSTALL_DIR]\doc\startutorialsdata\starnx\data`.

1. Copy directory `data` to your own disk space and open the part file `PorousModel.prt` in STAR-NX.

   The assembly is made up of three parts: the inlet region, the porous middle region, and the outlet region.

   The imported geometry is shown below.



The STAR-NX ribbon appears at the top of the screen.The STAR-NX object tree is located on the left of the screen.

To switch to this object tree:

2. Click (**STAR Tree**).

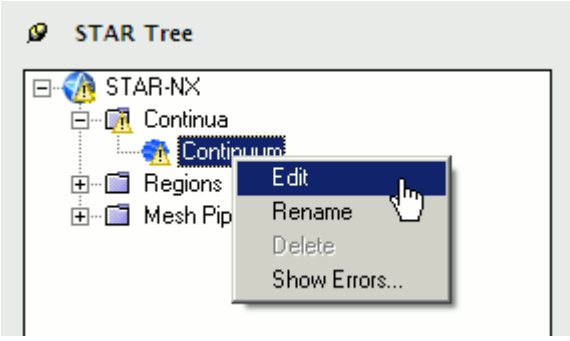
If these items are not visible, refer to Launching STAR-NX Manually.

Selecting Fluid Physics Models

The first step in setting up this tutorial is to define the physics of the problem. For this case, the flow is turbulent and isothermal. Temperature calculations are on by default in STAR-NX.

To turn temperature calculations off:

1. Right-click the **Continua** > **Continuum** node in the STAR-NX tree and select **Edit**.



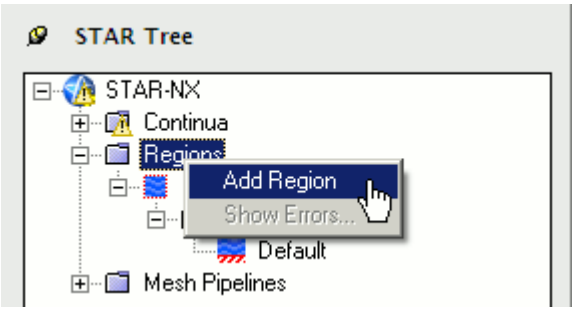
The *Continuum Settings* dialog appears.

2. Deactivate the **Temperature** option in the *Optional Physics Models* box.
3. Click **OK**.

Setting the Porous Region

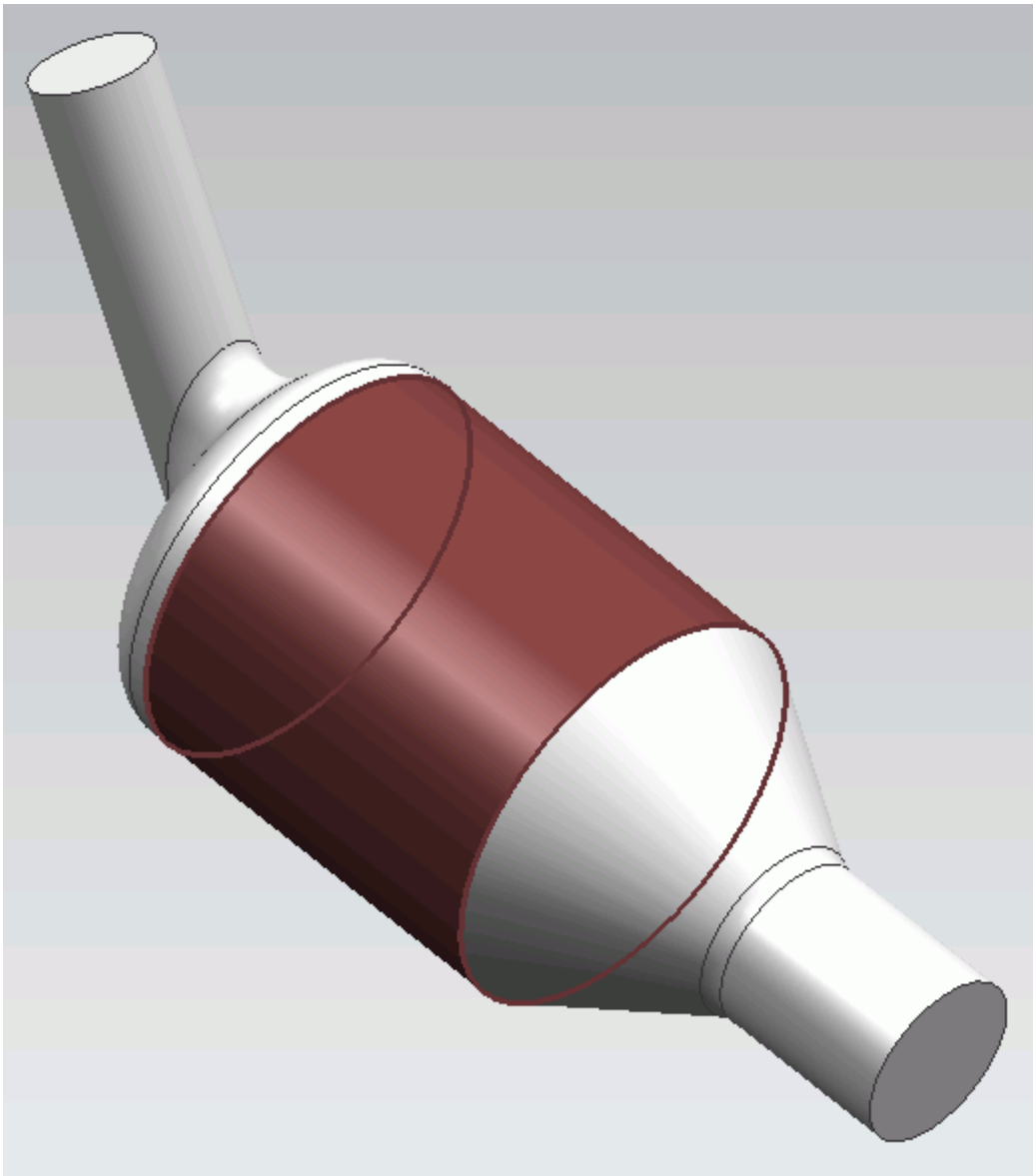
Create the porous region and define its properties:

1. Right-click the **Regions** node in the STAR-NX object tree and select **Add Region**.



The *Region Settings* dialog appears.

2. Select the middle part of the assembly as shown below.



3. Set *Name* to `Porous Region`.
4. Activate the *Porous* model and select the **Uni-Directional** porous type.
5. Set the following properties for the *Porous* region:

Parameter	Value
α (*Alpha coefficient*)	`25 kg/m^4`
β (*Beta coefficient*)	`1500 kg/m^3.s`
(*Cross-Flow Resistance Factor*)	`250`

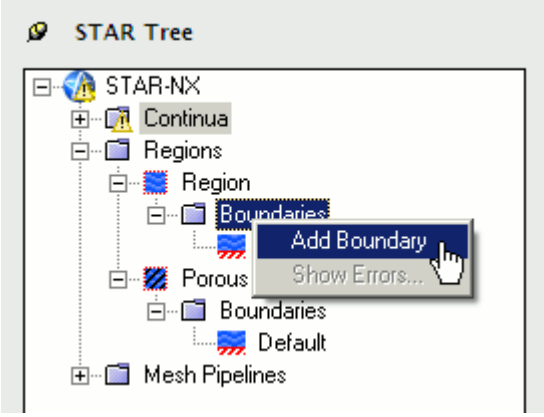
6. Click **OK**.

Defining Boundary Conditions

Create the inlet and outlet boundary conditions. STAR-NX automatically adds interfaces between the different regions. These interfaces are visible in the final STAR-CCM+ simulation.

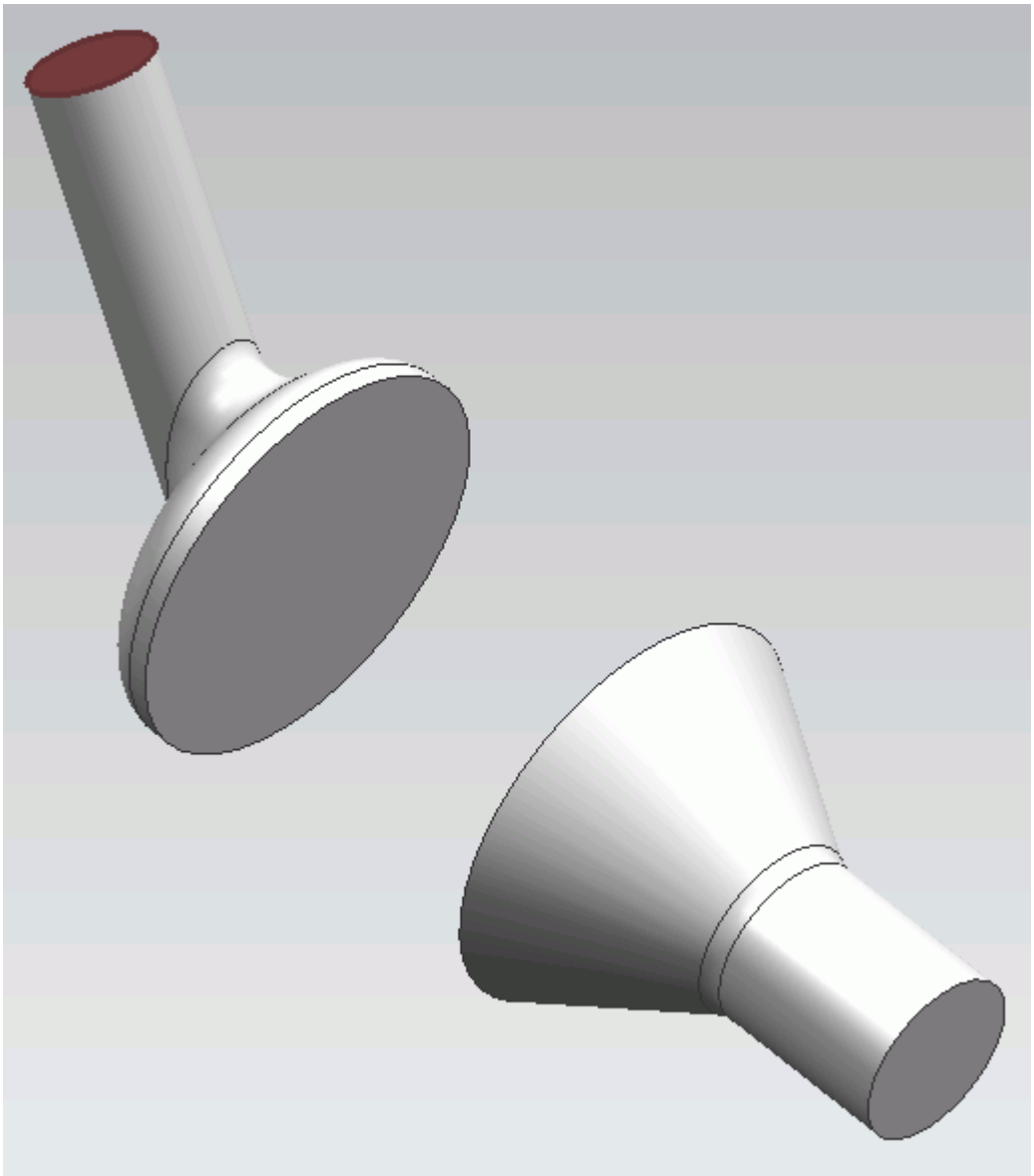
Create the inlet boundary first:

1. Right-click the **Regions** > **Region** > **Boundaries** node and select **Add Boundary**.



   The *Boundary Settings* dialog appears.

2. Select the face that is highlighted below.



These selections are reported in the *Face Selection* box in the *Boundary Settings* dialog.

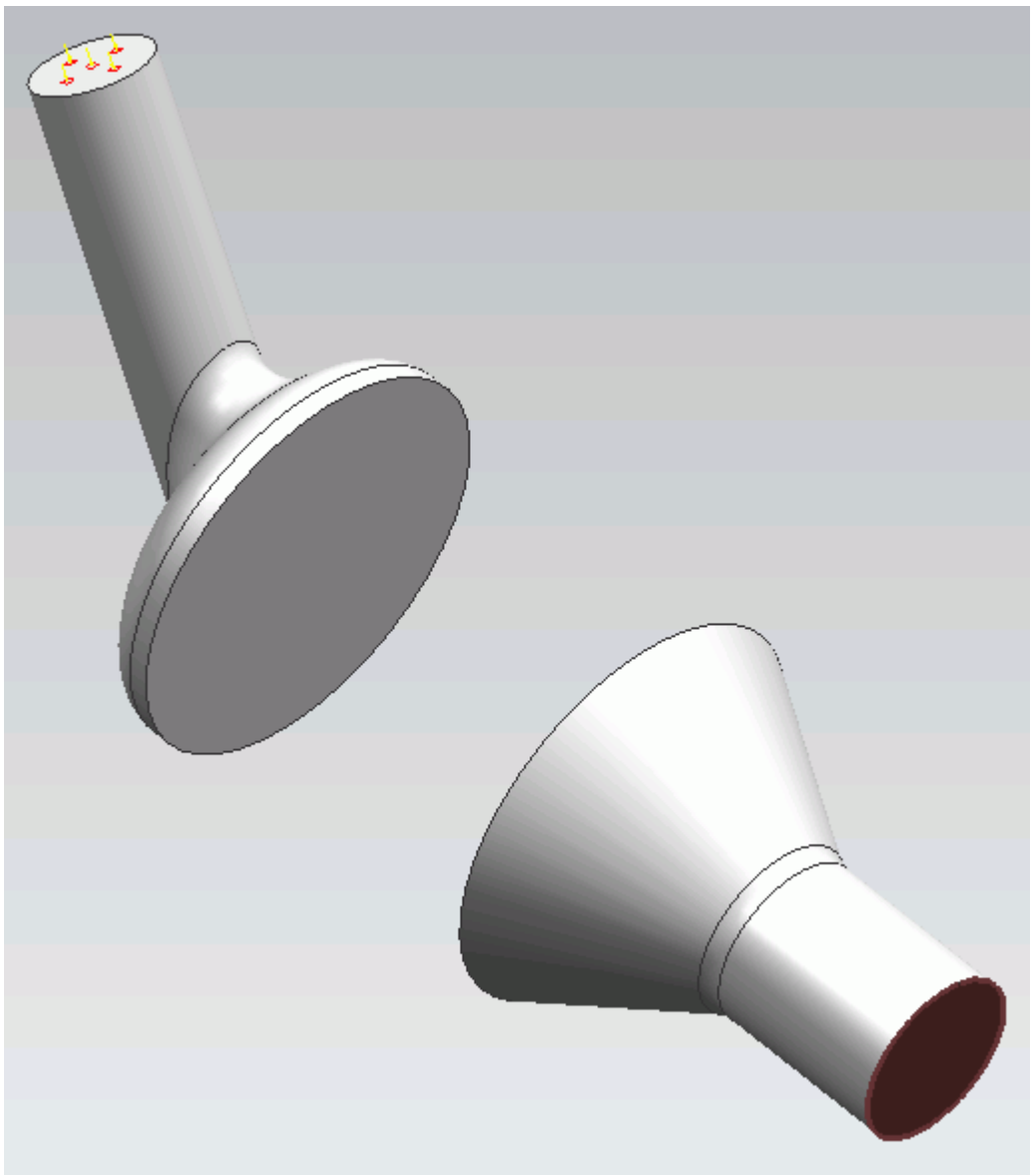
3. Set the following parameters:

Parameter	Value
Type	**Velocity Inlet**
Name	`Inlet`
(*Normal Velocity*)	`5 m/s`

4. Click **OK**.

Create the outlet boundary:

5. Using the same technique, create another boundary.
6. Select the face that is highlighted below.



7. Set the following parameters:

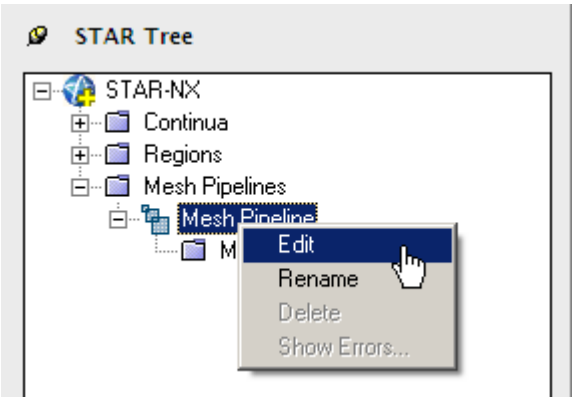
Parameter	Value
Type	**Pressure Outlet**
Name	`Outlet`

8. Click **OK**.

Setting the Mesh Pipeline

To obtain more accurate results, modify the mesh pipeline:

1. Right-click the **Mesh Pipelines** > **Mesh Pipeline** node in the STAR-NX tree and select **Edit**.



The *Mesh Pipeline Settings* dialog appears.

2. Set the following parameters:

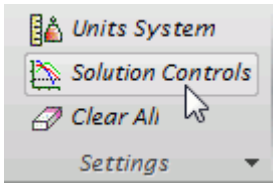
Parameter	Value
Base Size	
(**Base size option**)	**Absolute**
(*Base Size*)	`0.0025 m`
Prism Layer	
(**Layer thickness option**)	**Absolute**
(*Prism Layer Thickness*)	`0.001 m`

3. Click **OK**.

Modifying Solution Controls

Modify the maximum number of iterations and the solution accuracy settings.

1. Click **Solution Controls** on the STAR-NX ribbon.



The *Solution Controls* dialog appears.

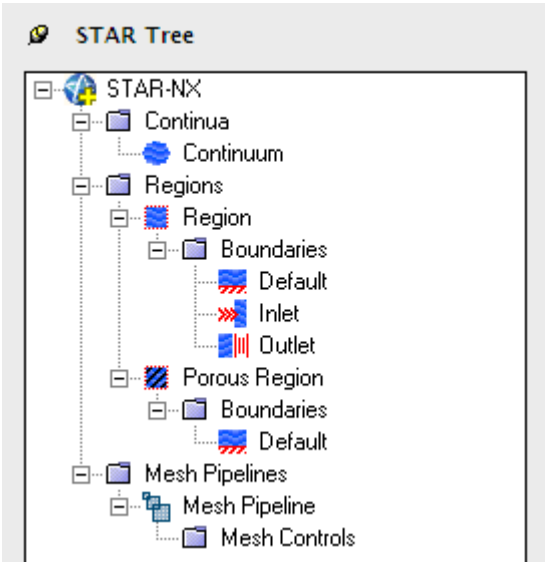
2. Expand the *Advanced* group box.
3. Set the following parameters:

Parameter	Value
(*Number of Iterations*)	400
Solution Accuracy	**First Order**

4. Click **OK**.
5. Save your progress.

Reviewing the Simulation Set Up

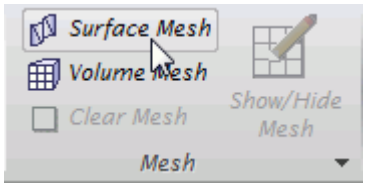
The final STAR-NX object tree is shown below.



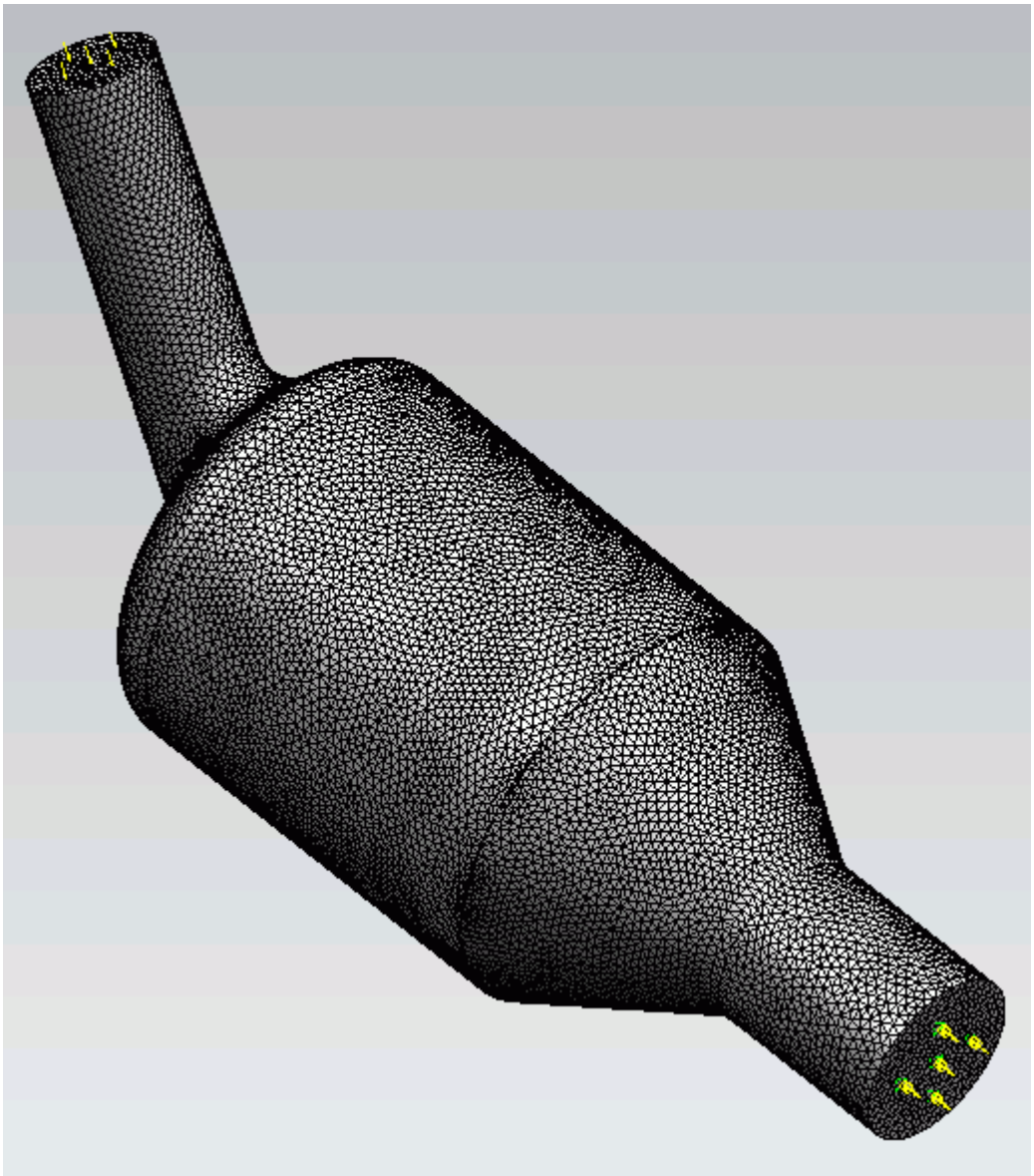
Generating the Surface Mesh

Generate the surface mesh:

1. From the STAR-NX ribbon, click **Surface Mesh**.



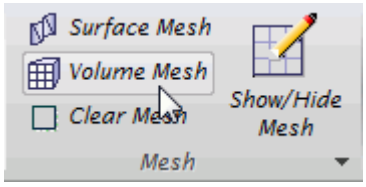
The surface mesh is shown below.



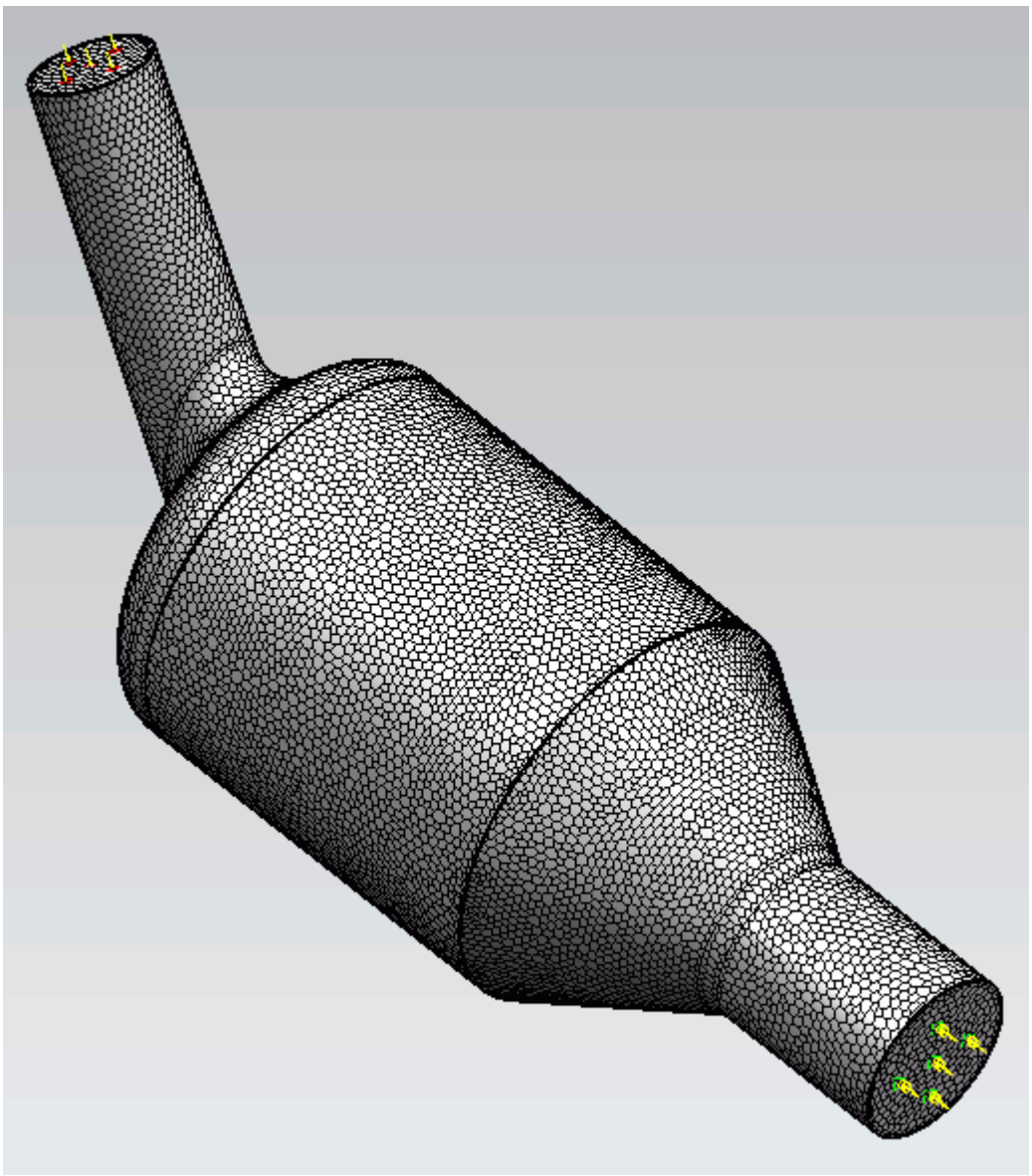
Generating the Volume Mesh

Generate the volume mesh:

1. From the STAR-NX ribbon, click **Volume Mesh**.



The volume mesh is shown below.

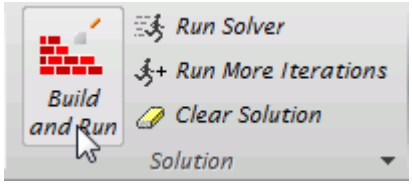


Building and Running

The case is now ready to run.

To run the simulation:

1. From the ribbon, click **Build and Run**.



The *Residuals* plot appears in the STAR-NX *Output* window. You can use this plot to monitor the convergence of the solution.

When the solver is done, STAR-CCM+ launches.

Post-Processing Results

You perform all post-processing in STAR-CCM+. By default, a scalar scene, *Scalar Scene 1*, automatically appears in the *Graphics* window when the solver finishes. This scene displays the absolute total pressure.

The results that are contained in the following sections are representative. You may see some variation in the results that you obtain, depending on what platform and package you are using.

1. In STAR-CCM+, click **File** > **Save** to save the simulation.

Contents:

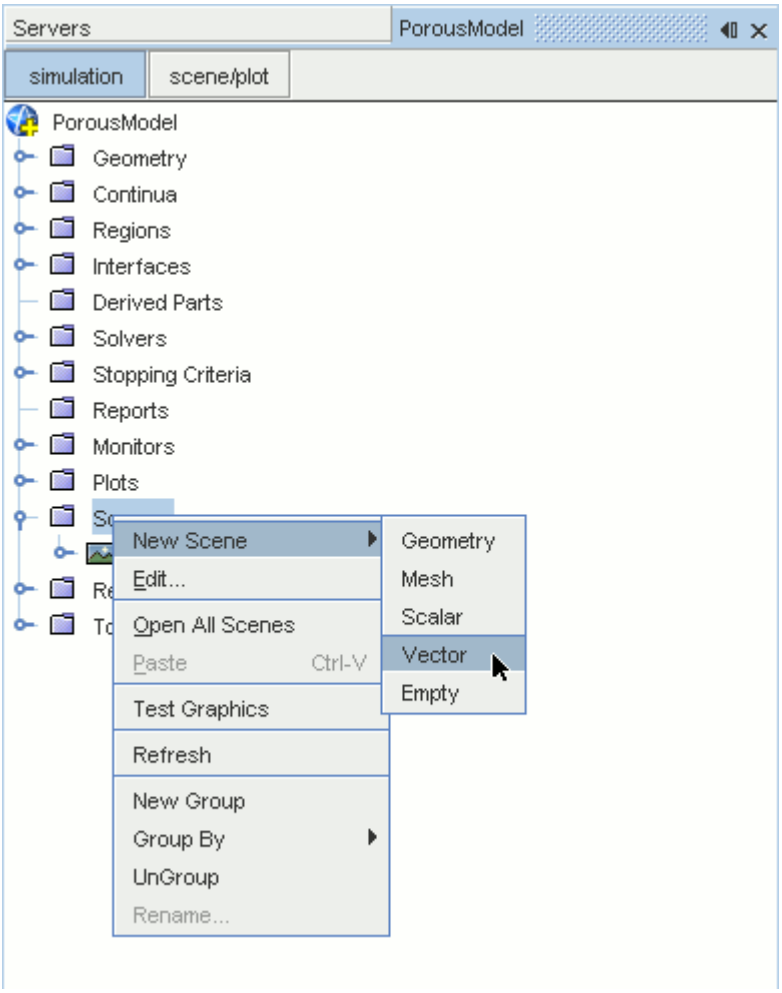
Creating a Scalar Scene
Creating A Streamlines Scene
Creating a Vector Scene
Creating a Streamlines Scene
Creating A Scalar Scene
Visualizing the Wing Skin Temperature
Creating a Streamlines Scene
Creating A Scalar Scene
Creating a Streamlines Scene
Creating a Scalar Scene
Modifying The Scalar Scene
Creating a Streamlines Scene

Creating a Vector Scene

Create a vector scene with a section plane bisecting the geometry. Velocity vectors are plotted on this plane.

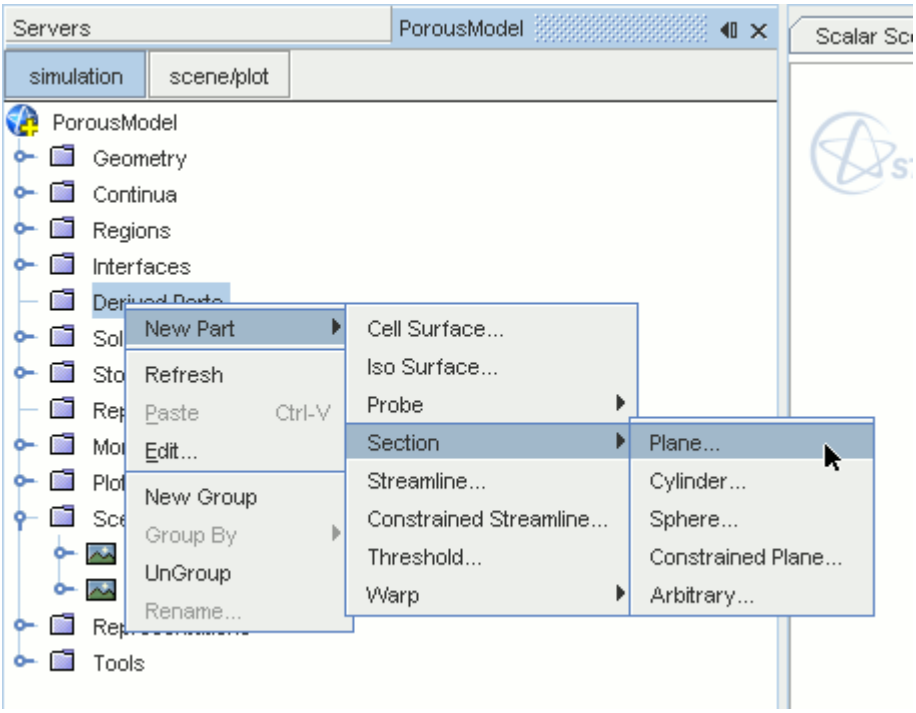
To create the vector scene:

1. Right-click the **Scenes** node in the explorer pane and select **New Scene** > **Vector**.



A new vector scene, *Vector Scene 1*, is created and selected in the *Graphics* window.

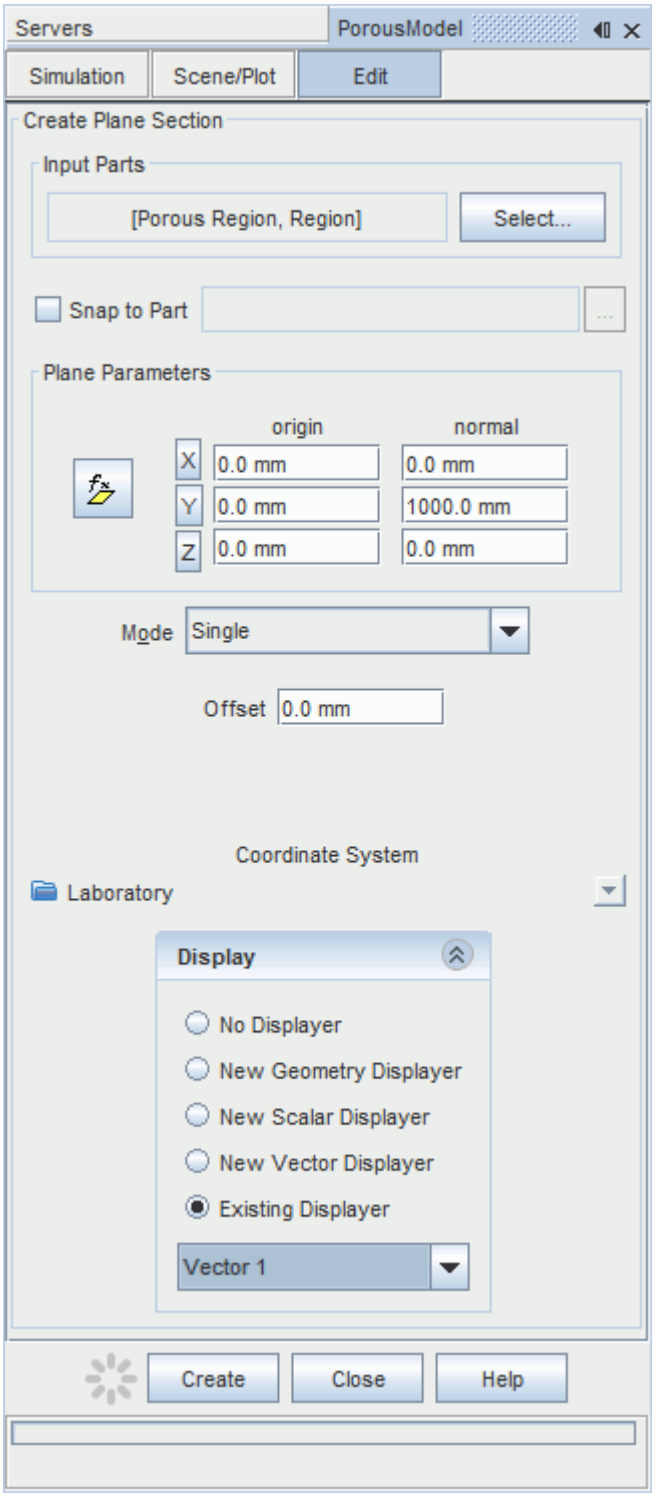
2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane...**.



The *Create Plane Section* dialog appears.

3. In the *Plane Parameters* box, set the *origin* vector to `[0 mm, 0 mm, 0 mm]`.
4. Set the *normal* vector to `[0 mm, 1000 mm, 0 mm]`.
5. In the *Display* group box, select **Existing Displayer** and in the drop-down menu select **Vector 1**.

The completed dialog is shown below.



6. Click **Create** followed by **Close**.

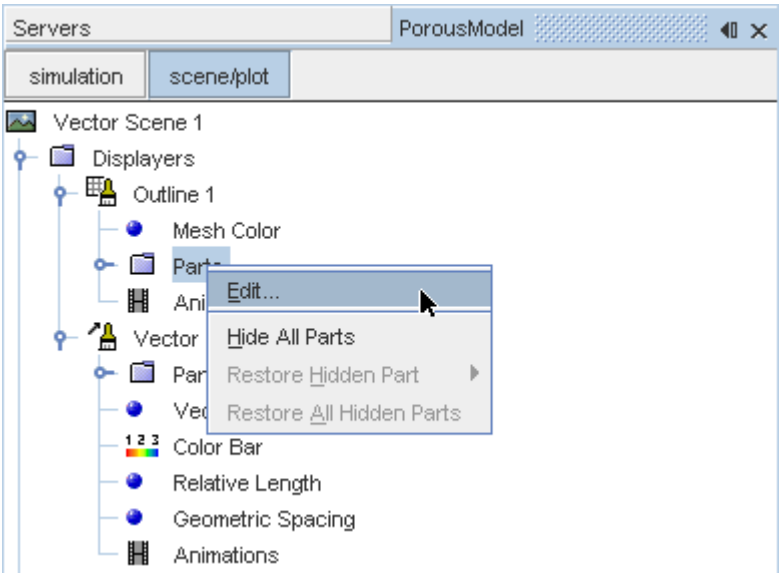
   Velocity is automatically set as the active function when creating a vector displayer.

Change the view of the scene to something more suitable:

7. On the toolbar, click (**Save-Restore-Select views**) and select **Views** > **+X +Y +Z** > **Up +Y**.

In order to remove the unwanted outlines from the scene do the following:

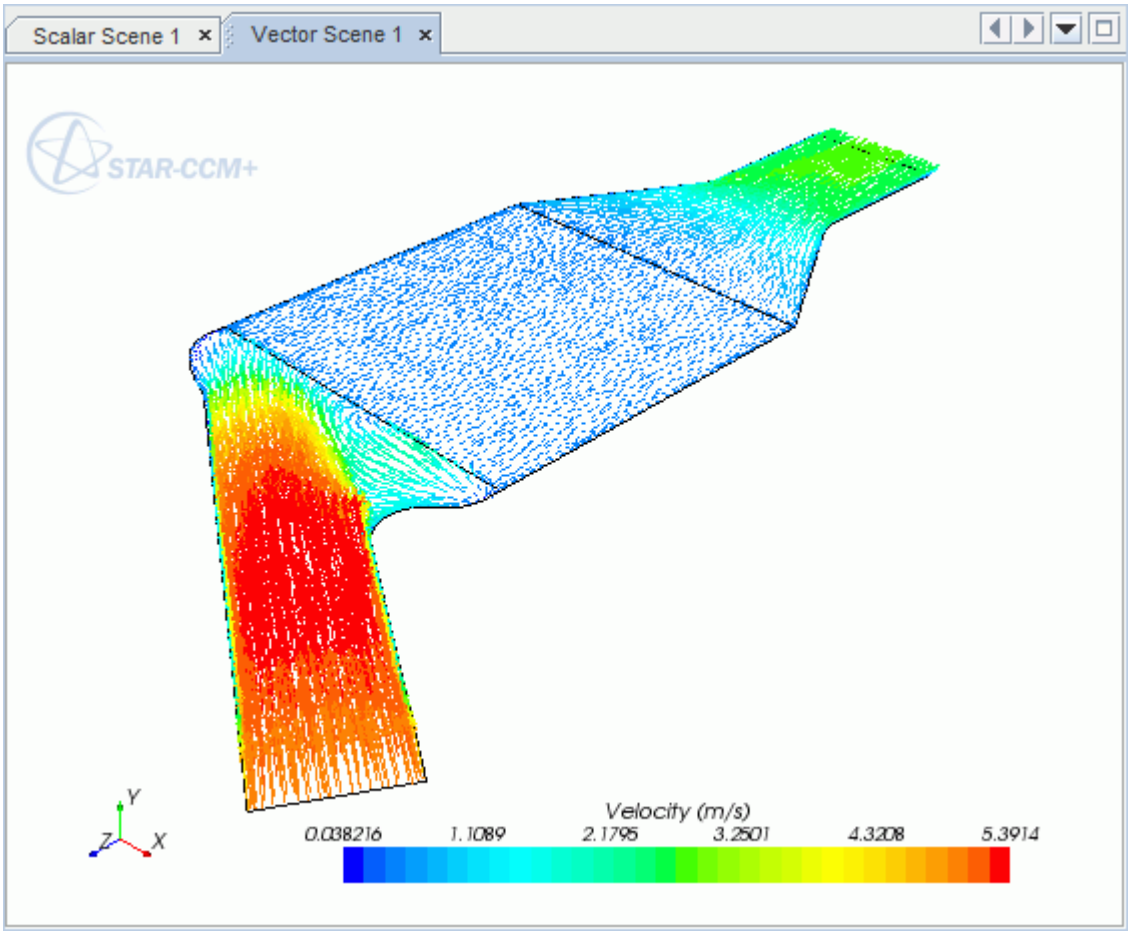
8. Click the **scene/plot** button in the explorer pane.
9. Navigate to **Vector Scene 1** > **Displayers** > **Outline 1** > **Parts**.
10. Right-click the **Parts** node and select **Edit...**.



The *Parts* dialog appears.

11. Right-click on a blank area and choose **Deselect All**.
12. Select the **Derived Parts** > **Plane Section** node.
13. Click **OK**.
14. Click the **simulation** button to return to the simulation object tree.
15. Click (**Save**) to save the simulation .

The finished scene is shown below.



Creating a Streamlines Scene

Create streamlines to get a better understanding of the flow inside the catalyst.

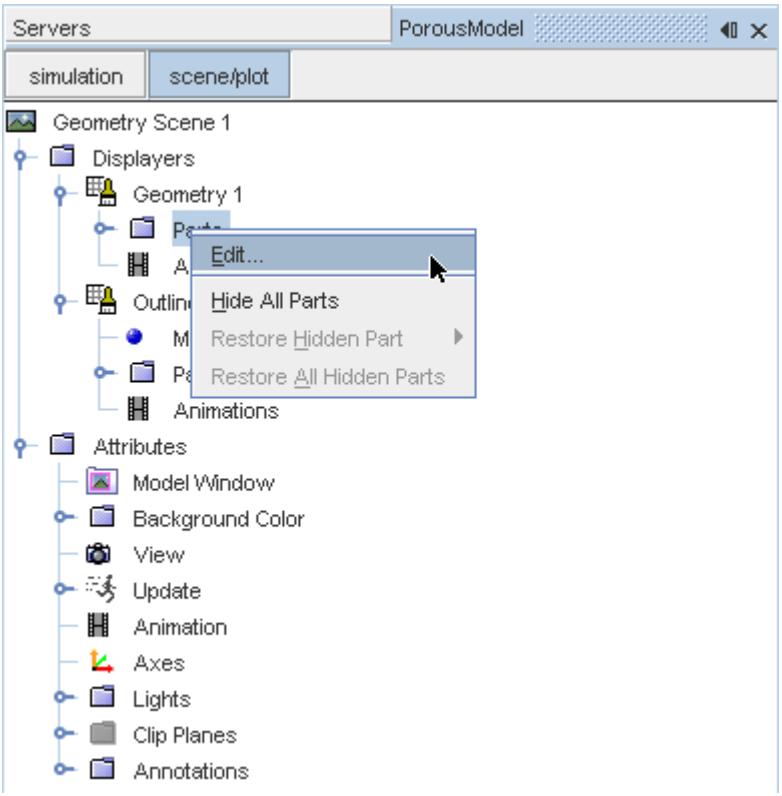
Begin by creating a geometry scene:

1. Right-click the **Scenes** node in the object tree and select **New Scene** > **Geometry**.

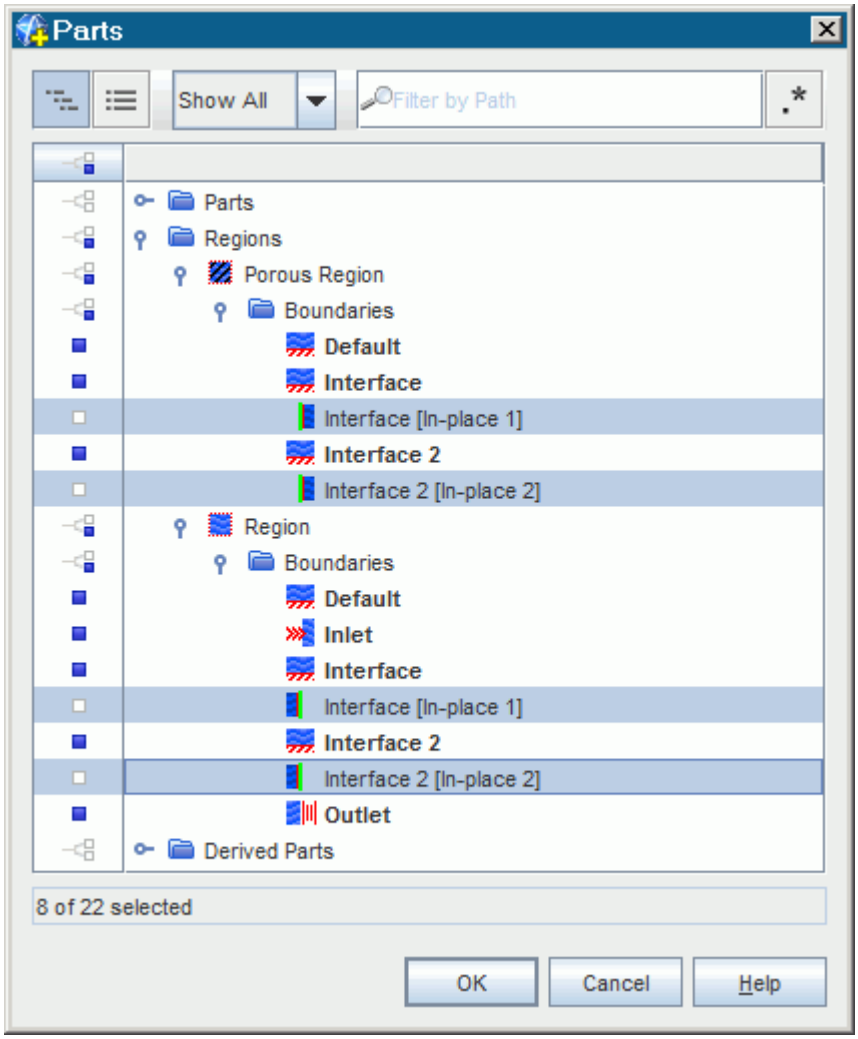
   A new geometry scene is added to the *Graphics* window.
2. Click **scene/plot**.
3. Navigate to **Geometry Scene 1** > **Displayers** > **Geometry 1**.
4. Set *Opacity* to `0.3`.

To remove the yellow interface boundaries from the scene:

5. Right-click **Geometry Scene 1** > **Displayers** > **Geometry 1** > **Parts** and select **Edit...**.



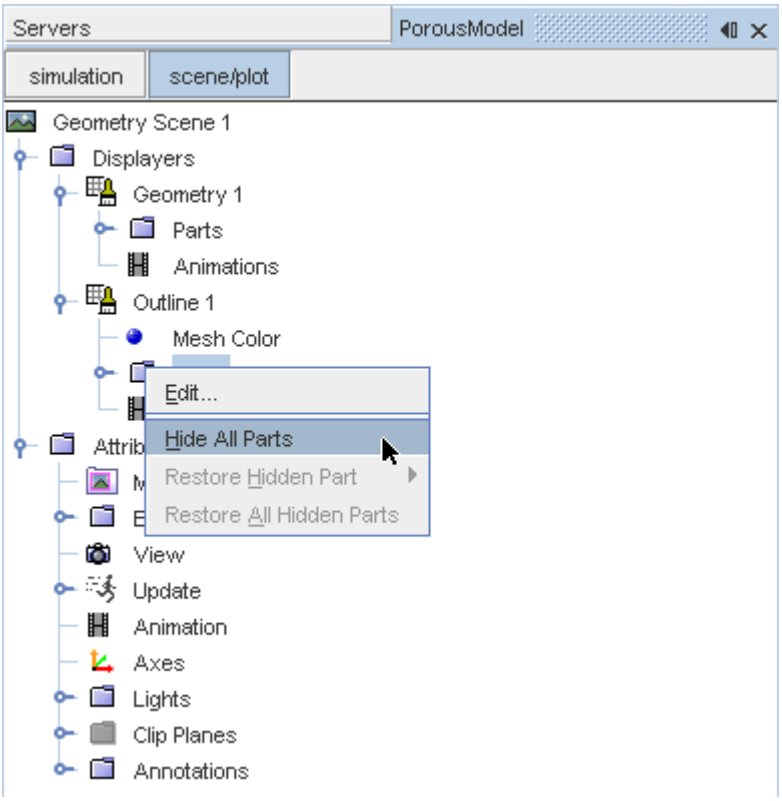
6. In the *Parts* dialog that appears, expand the **Regions** > **Porous Region** > **Boundaries** and **Regions** > **Region** > **Boundaries** nodes.
7. Deactivate the following items in each region:
   - **Interface [In-Place 1]**
   - **Interface 2 [In-Place 2]**



8. Click **OK**.

Hide the outlines on the geometry model:

9. Right-click the **Geometry Scene 1** > **Displayers** > **Outline 1** > **Parts** node in the object tree and select **Hide All Parts**.



10. Click **simulation**.

Change the view of the scene to something more suitable:

11. On the toolbar, click (**Save-Restore-Select views**) and select **Views** > **+X +Y +Z** > **Up +Y**.

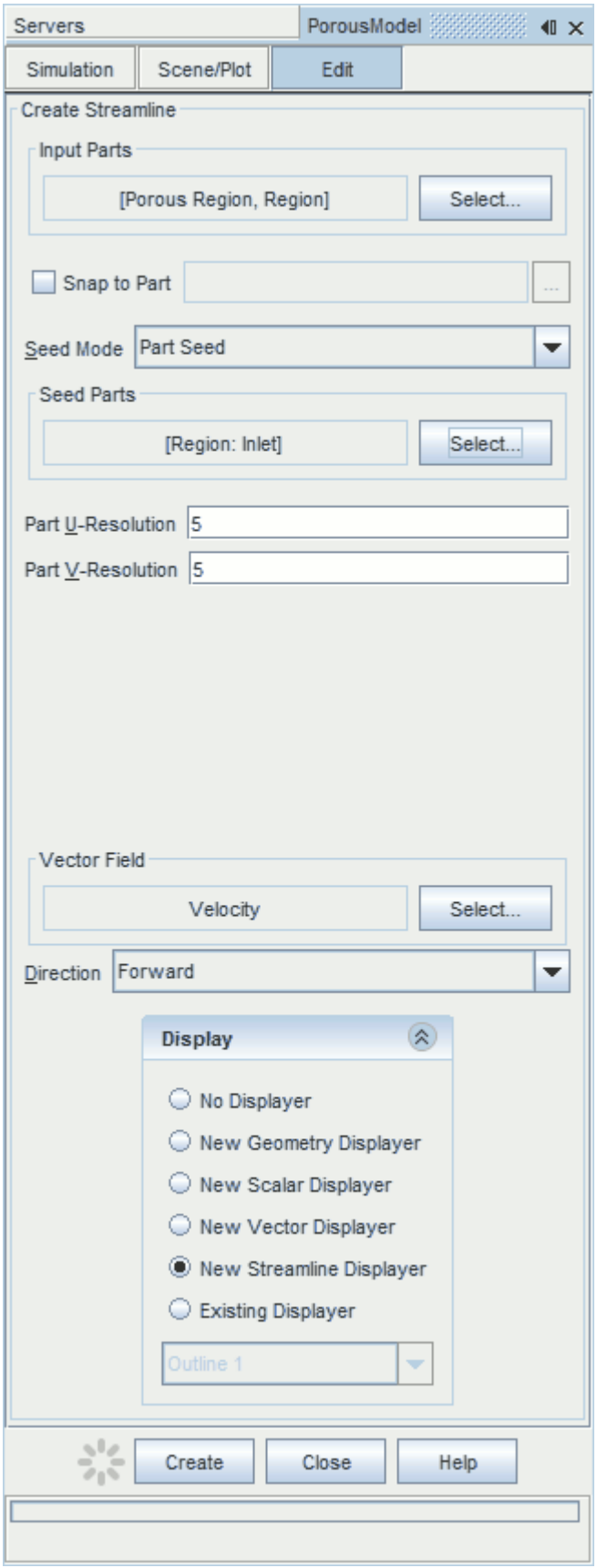
Create the streamlines:

12. Right-click on the **Derived Parts** node and select **New Part** > **Streamline...**.

The *Create Streamline* dialog appears.

13. Make sure that **Part Seed** is selected in the *Seed Mode* section.
14. Set *Seed Parts* to **Regions** > **Region** > **Boundaries** > **Inlet**.
15. Click **OK**.
16. Make sure that **New Streamline Displayer** is selected in the *Display* group box.

The completed dialog is shown below.


Servers
PorousModel
Simulation
Scene/Plot
Edit
Create Streamline
Input Parts
[Porous Region, Region]
Select...
Snap to Part
Seed Mode
Part Seed
Seed Parts
[Region: Inlet]
Select...
Part U-Resolution
5
Part V-Resolution
5
Vector Field
Velocity
Select...
Direction
Forward
Display
No Displayer
New Geometry Displayer
New Scalar Displayer
New Vector Displayer
New Streamline Displayer
Existing Displayer
Outline 1
Create
Close
Help

17. Click **Create** then **Close**.

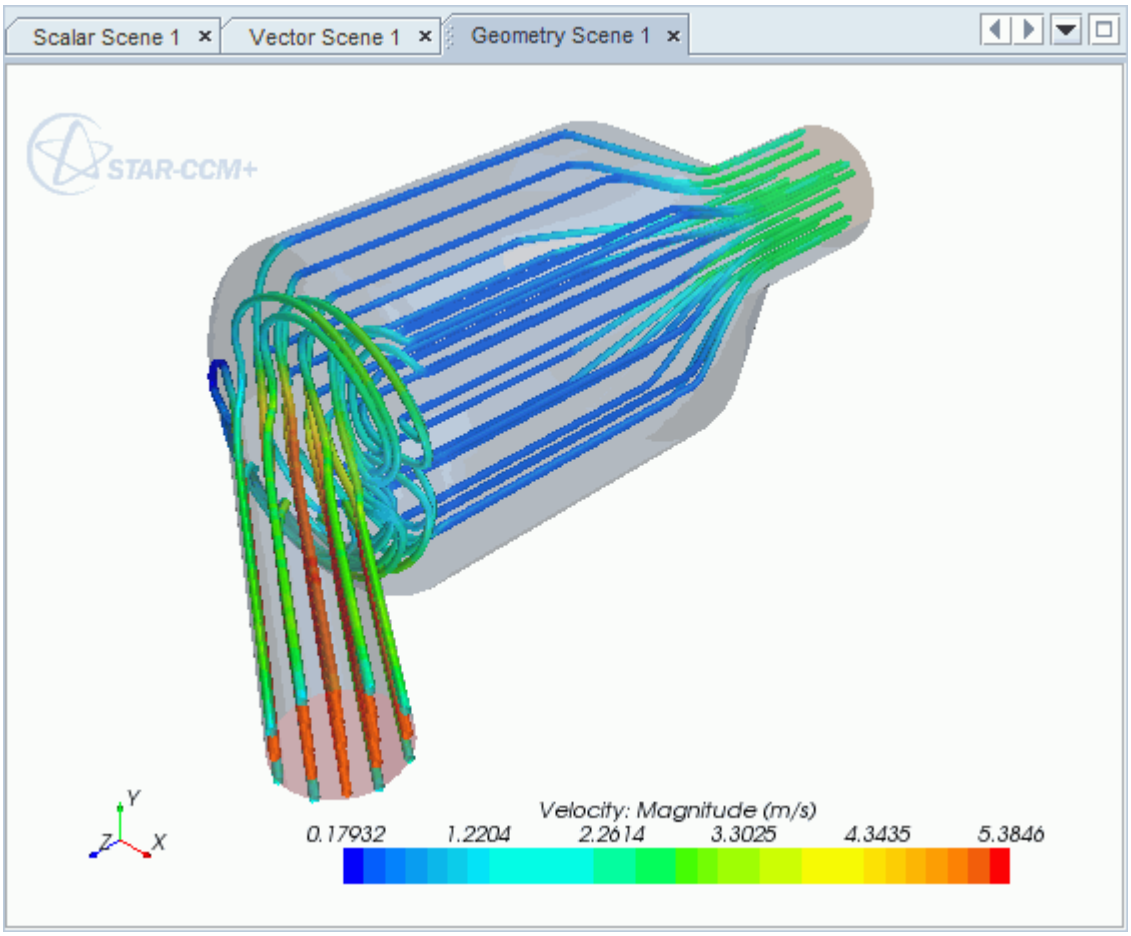
Change the display mode of the streamlines to use tubes instead of the default lines:

18. Click **scene/plot**.
19. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1**.
20. Set *Width* to `0.0012` and *Mode* to **Tubes**.

Change the color of the streamlines to represent velocity magnitude:

21. Navigate to the **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** > **Scalar Field**.
22. Set *Function* to **Velocity** > **Lab Reference Frame** > **Magnitude**.
23. Save the simulation.

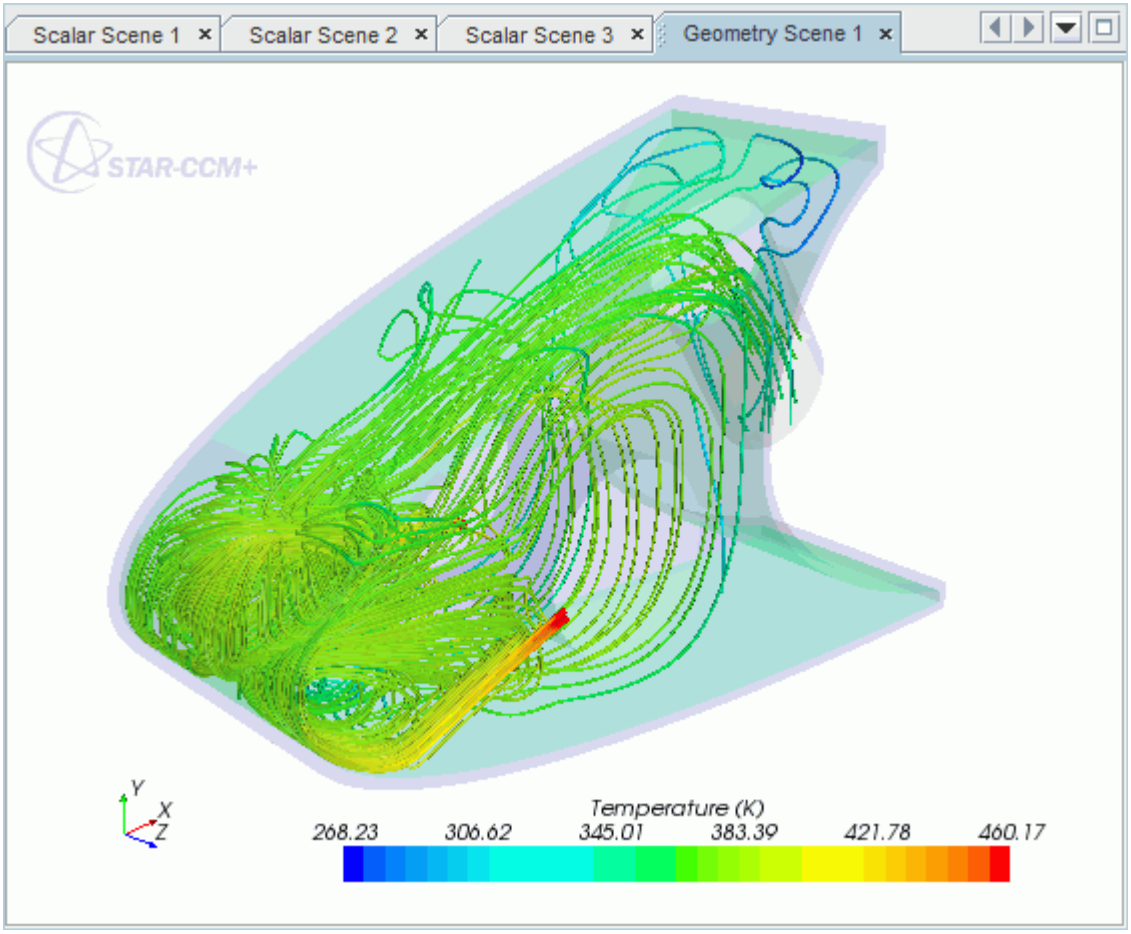
The finished scene is shown below.



Aircraft Wing Deicing System

This tutorial takes you through the steps that are required to run a CFD calculation in the STAR-NX environment.

For this tutorial, the model is a modern deicing system that is used on aircraft wings. The hot air from the aircraft's jet engine is used to heat the leading edge of the wing.



Contents:

Opening the CAD Model
Selecting Fluid Physics Models
Adding Solid Physics Models
Setting the Solid Region
Defining Boundary Conditions for the Fluid Region
Defining Boundary Conditions for the Solid Region
Setting the Mesh Pipeline
Setting the Coincident Faces Tolerance

Creating the Volumetric Mesh Controls
Creating a Part Mesh Control
Creating a Face Mesh Control
Modifying Solution Controls
Reviewing the Simulation Set Up
Generating the Surface Mesh
Generating the Volume Mesh
Building and Running
Post-Processing Results

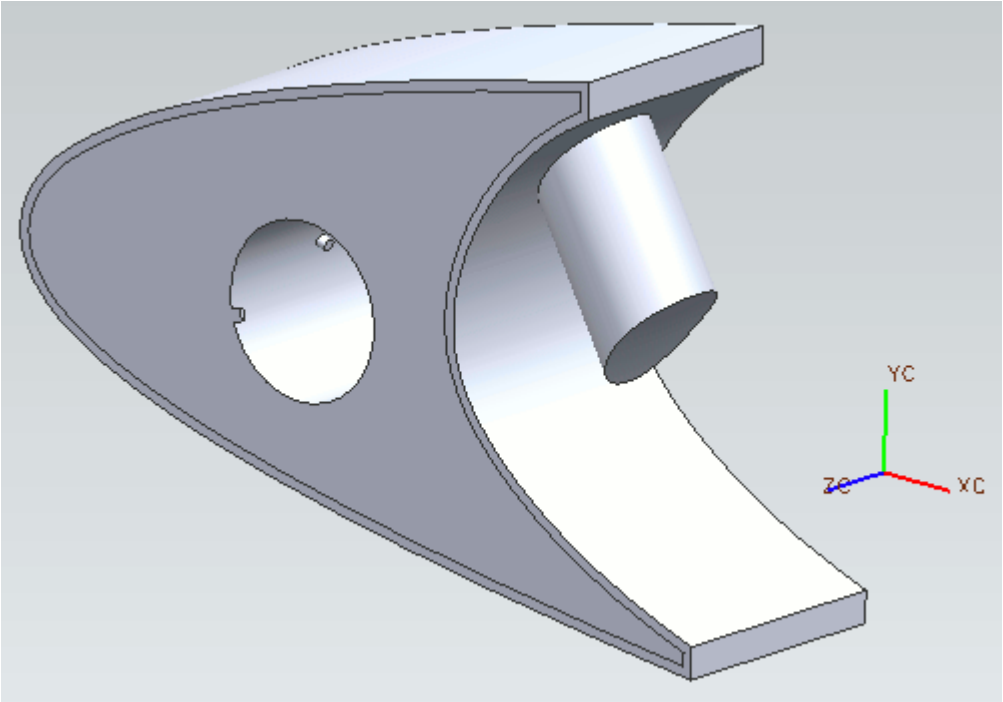
Opening the CAD Model

This directory is located in `[STAR-CCM+_INSTALL_DIR]\doc\startutorialsdata\starnx\data`.

1. Copy directory `data` to your own disk space and open the part file `piccoloAssem.prt` in STAR-NX.

   The assembly is made up of two parts: the solid leading edge wing skin, and the internal air region.

   The imported geometry is shown below.



The STAR-NX ribbon appears at the top of the screen.The STAR-NX object tree is located on the left of the screen.

To switch to this object tree:

2. Click (**STAR Tree**).

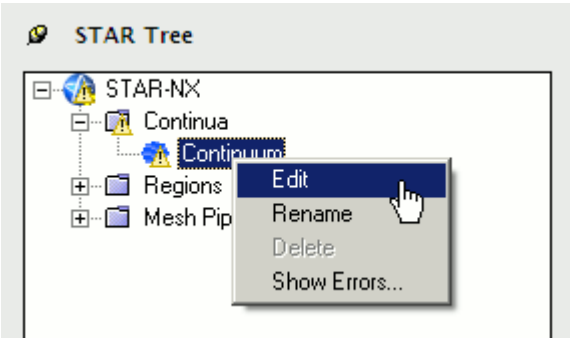
If these items are not visible, refer to Launching STAR-NX Manually.

Selecting Fluid Physics Models

The first step in setting up this tutorial is to modify the physics continuum that is used to model air. For this case, the flow is turbulent and thermal. Temperature calculations are on by default in STAR-NX.

Modify the initial conditions:

1. Right-click the **Continua** > **Continuum** node in the STAR-NX tree and select **Edit**.

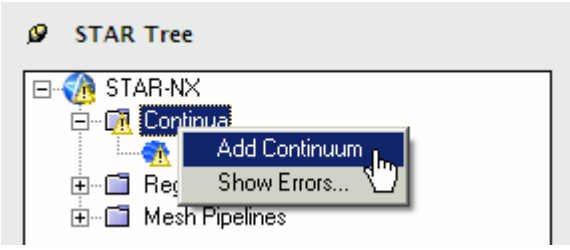


   The *Continuum Settings* dialog appears.
2. Set *Name* to `Air`.
3. Set the *Equation of State* to **Ideal Gas**.
4. Click to expand the *Initialization* tab.
5. Set (*Temperature*) to `400` K.
6. Click **OK** to close the dialog.

Adding Solid Physics Models

The next step in setting up this tutorial is to define the physics continuum of the solid to be used.

1. Right-click the **Continua** node in the STAR-NX object tree and select **Add Continuum**.



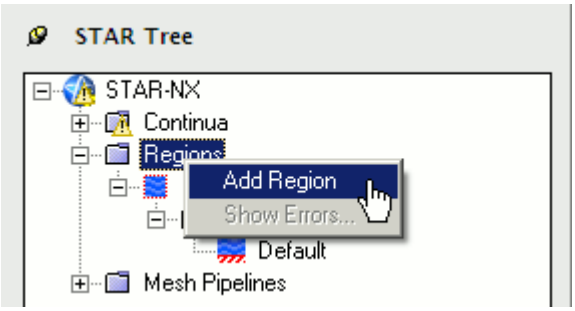
   The *Continuum Settings* dialog appears.
2. Set *Name* to `Al`.
3. Set *Material Type* to **Solid**.

4. Set *Material Name* to **Aluminum Alloy**
5. Click to expand the *Initialization* tab.
6. Set (*Temperature*) to `400 K`.
7. Click **OK**.

Setting the Solid Region

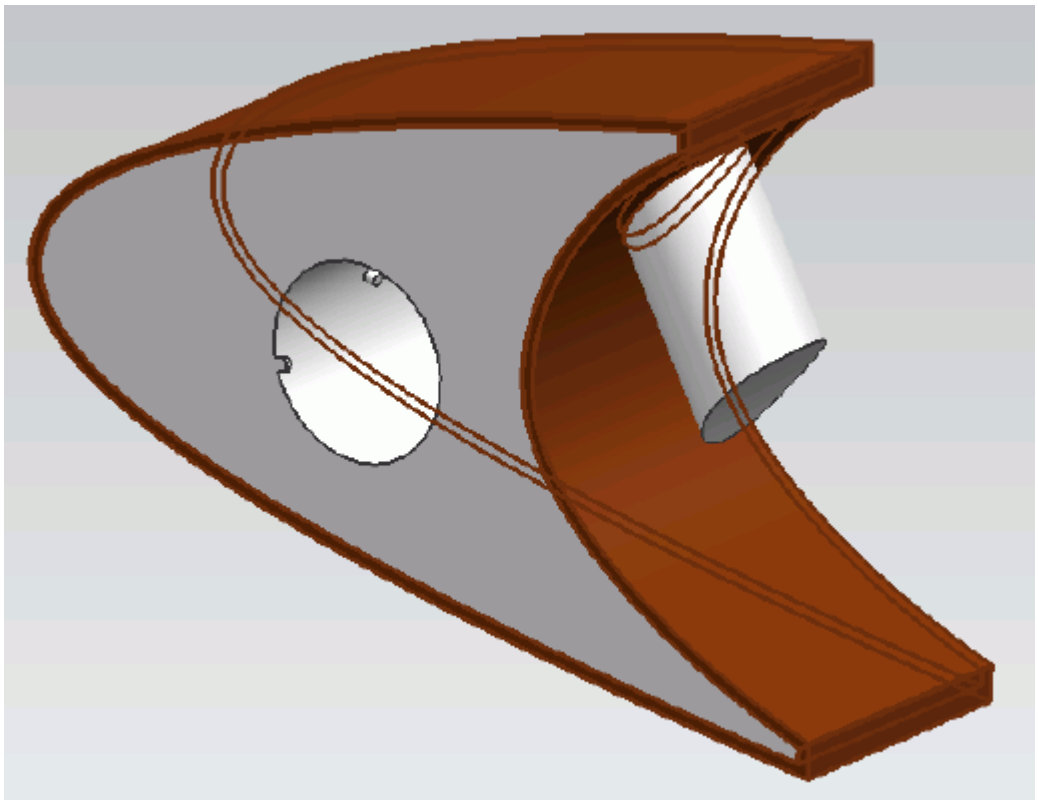
Create a new solid region for the wing skin:

1. Right-click the **Regions** node in the STAR-NX object tree and select **Add Region**.



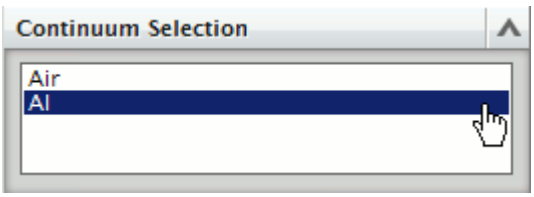
The *Region Settings* dialog appears.

2. Select the wing skin as shown below.



Once selected, the part becomes highlighted.

3. Set *Name* to `Leading Edge`.
4. Set *Continuum Selection* to **Al**.

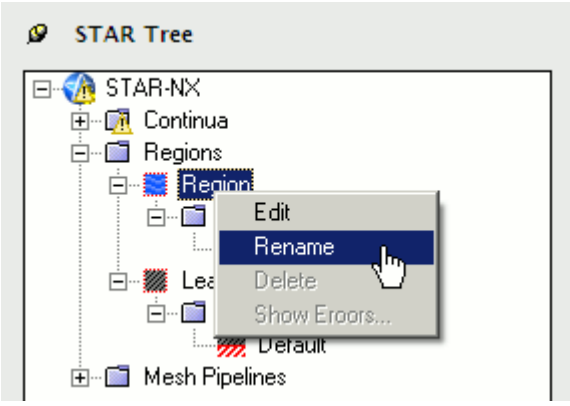


This action assigns the aluminum alloy that you created earlier as the material for this region.

5. Click **OK**.

Now that the solid region of the wing is defined, rename the default region to something more appropriate:

6. Right-click the **Regions** > **Region** node in the STAR-NX tree and select **Rename**.



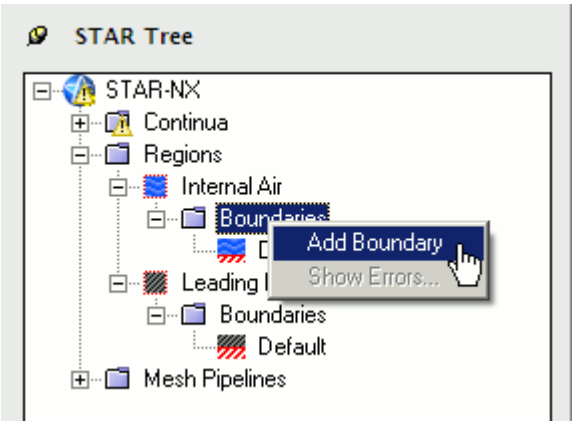
7. Enter `Internal Air` as the new name.
8. Click a blank area to accept the name change.

Defining Boundary Conditions for the Fluid Region

Create the inlet and outlet boundary conditions. Use symmetry boundaries on the side walls to extend the simulation domain. Use a wall boundary with a fixed temperature to model the piccolo walls.

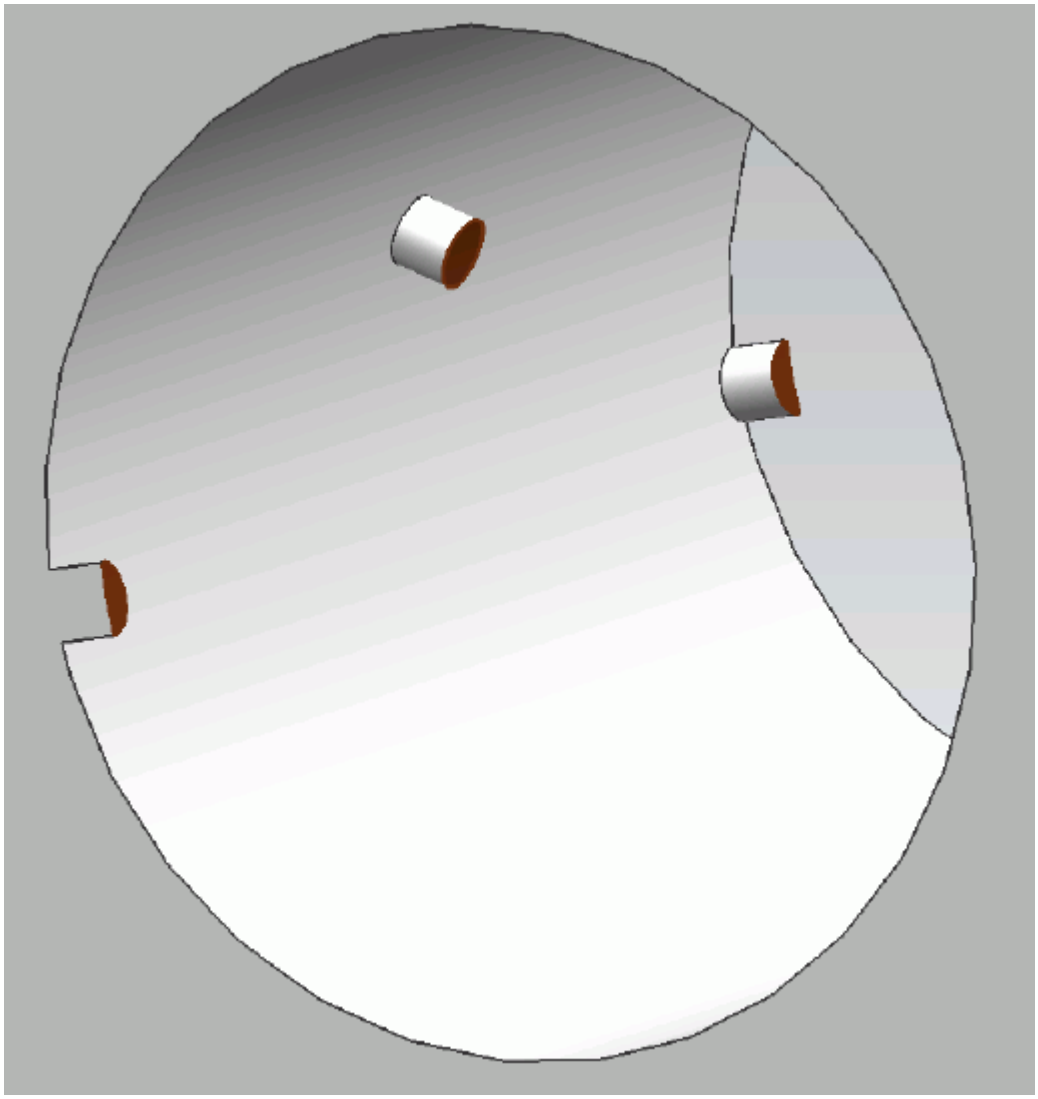
Create the inlet boundary first:

1. Right-click the **Regions** > **Internal Air** > **Boundaries** node and select **Add Boundary**.



The *Boundary Settings* dialog appears.

2. On the CAD model, select the three faces that are highlighted below.



These selections are reported in the *Face Selection* box in the *Boundary Settings* dialog.

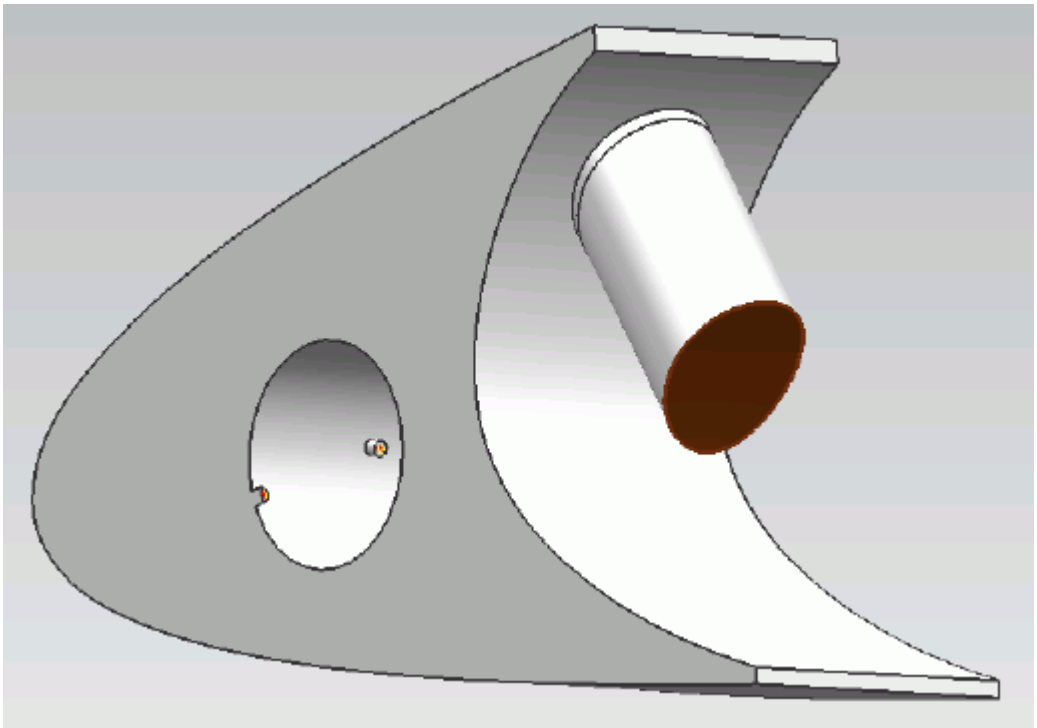
3. Set the following properties:

Parameter	Value
Type	**Velocity Inlet**
Boundary NameName	`Inlet`
(*Normal Velocity*)	`90 m/s`
(*Temperature*)	`460 K`
(*Turbulence Intensity*)	`0.135`
(*Turbulent Length Scale*)	`2.2e-005 m`

4. Click **OK**.

Create the outlet boundary:

5. Using the same technique, create another boundary.
6. Select the face that is highlighted below.



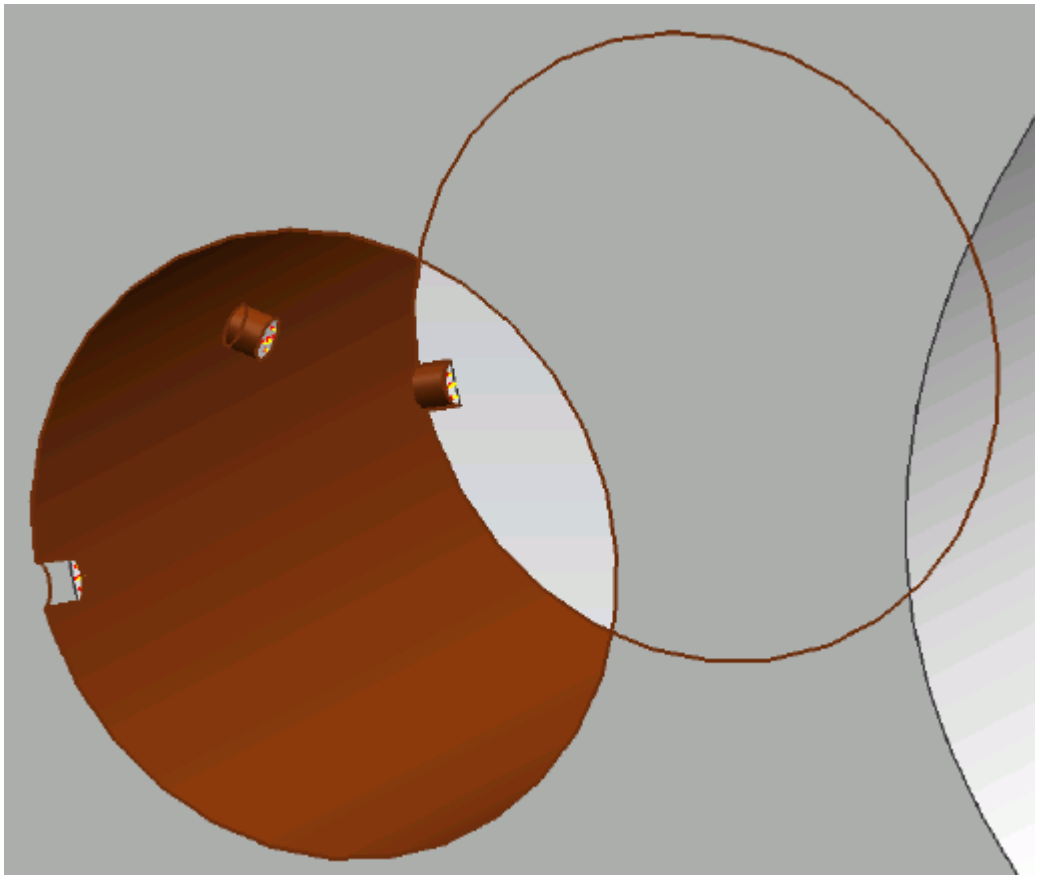
7. Set the following parameters:

Parameter	Value
Type	**Pressure Outlet**
Name	`Outlet`

8. Click **OK**.

Create the wall boundary that models the heated piccolo walls.

9. Create another boundary.
10. Select the four faces highlighted below, make sure to select the faces that are perpendicular to the inlet boundaries.



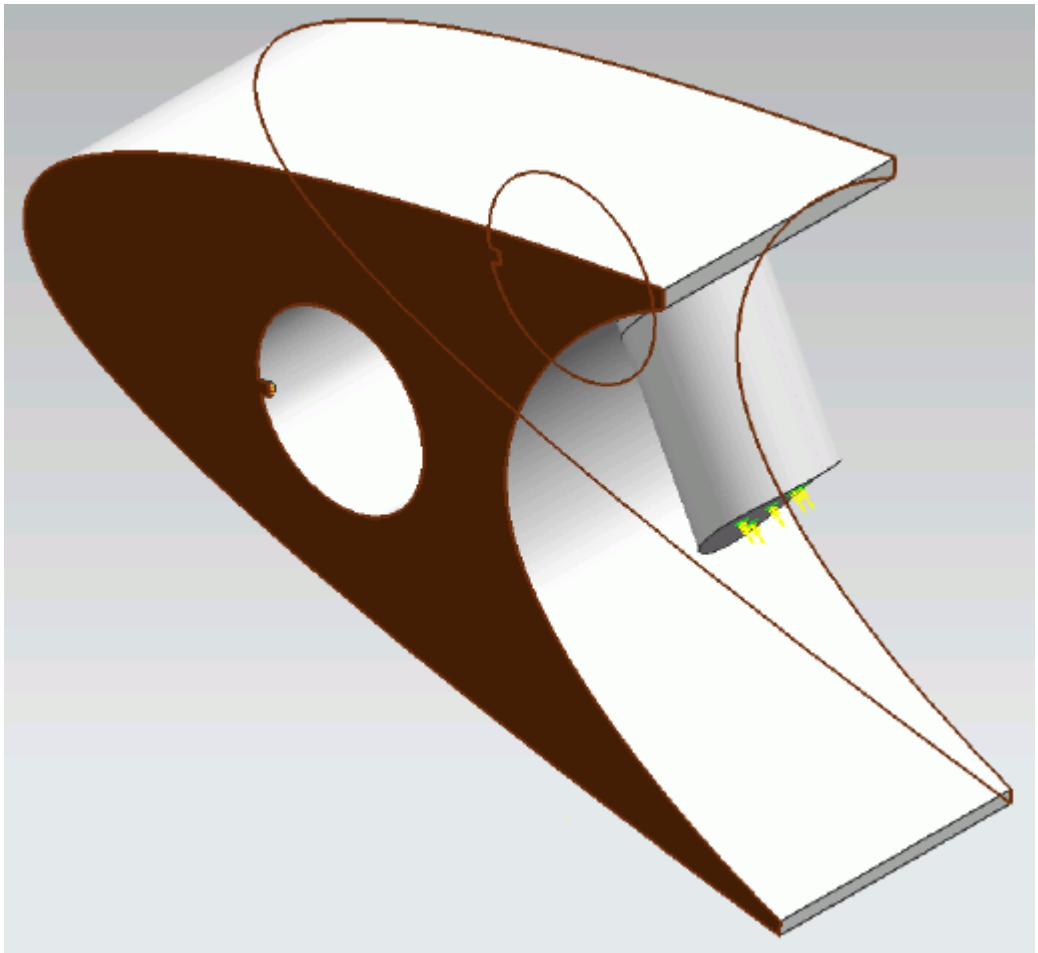
11. Set the following parameters:

Parameter	Value
Type	**Wall**
Name	`Piccolo Wall`
Thermal Specification	**Temperature**
(*Temperature*)	`460 K`

12. Click **OK**.

Create the symmetry boundaries:

13. Create a new boundary.
14. Select the two faces that are highlighted below, one on either side of the internal air region.



15. Set the following parameters:

Parameter	Value
Type	**Symmetry Plane**
Name	`Air Symmetry`

16. Click **OK**.

Defining Boundary Conditions for the Solid Region

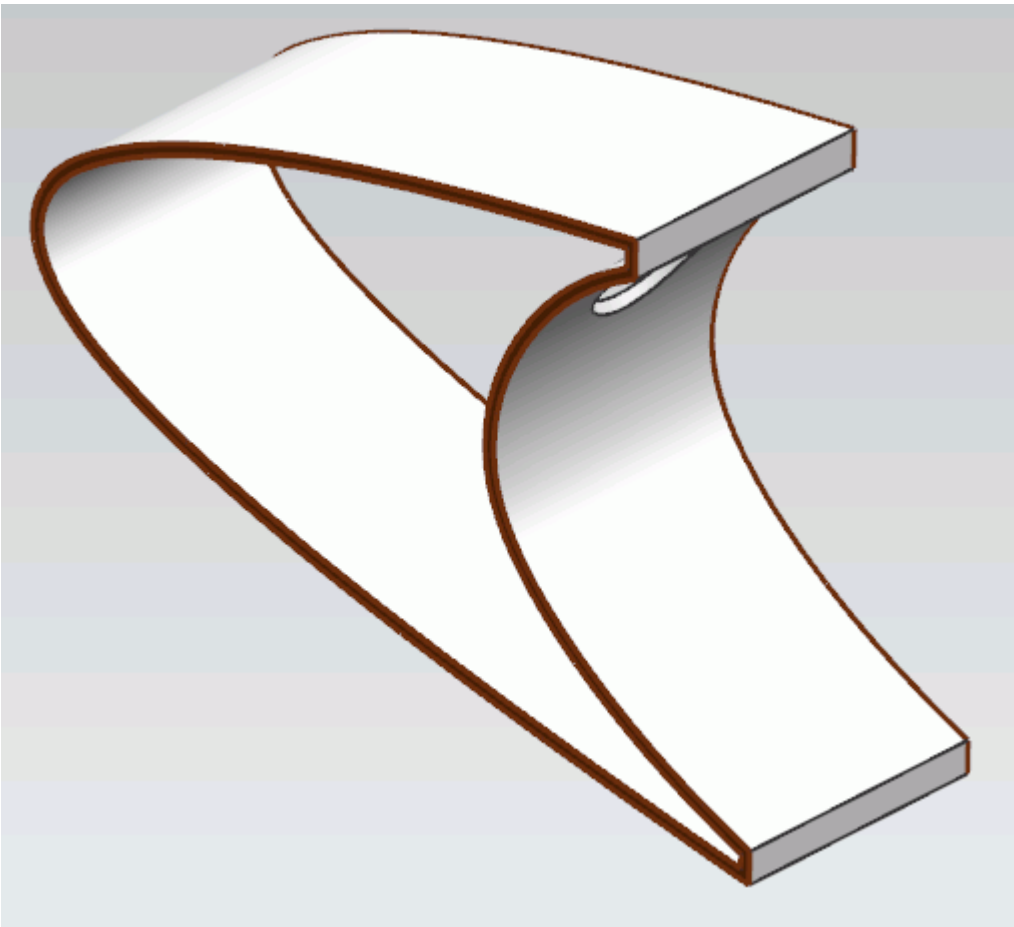
Use the convection boundary condition to model the heat propagation from the solid wing.

The convective boundary condition models ambient conditions on the outside of the domain. Use convection to model the heat propagation in the solid wing skin.

Create the symmetry boundary first:

1. Right-click the **Regions** > **Leading Edge** > **Boundaries** node and select **Add Boundary**.

   The *Boundary Settings* dialog appears.
2. Select the two faces highlighted below, one on either side of the geometry.



3. Set the following parameters:

Parameter	Value
Type	**Symmetry Plane**
Name	`Solid Symmetry`

4. Click **OK**.

Modify the default boundary so that it uses convection:

5. Right-click the **Regions** > **Leading Edge** > **Boundaries** > **Default** node and select **Edit**.

   The *Boundary Settings* dialog appears.

6. Set the following properties:

Parameter	Value
Thermal Specification	**Convection**
HTC (*Heat Transfer Coefficient*)	`100 W/m^2.K`
(*Temperature*)	`228 K`

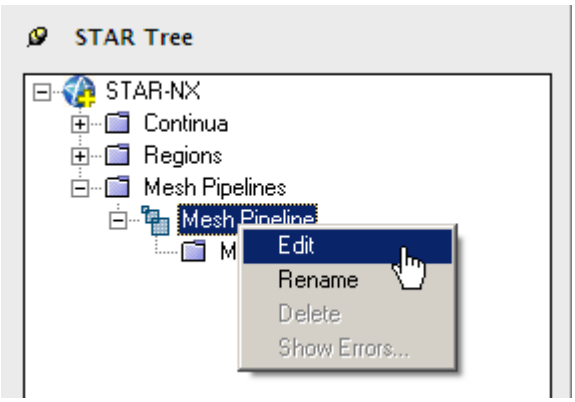
7. Click **OK**.

8. Save your progress.

Setting the Mesh Pipeline

To obtain more accurate results, modify the mesh pipeline:

1. Right-click the **Mesh Pipelines** > **Mesh Pipeline** node in the STAR-NX tree and select **Edit**.



The *Mesh Pipeline Settings* dialog appears.

2. Set the following parameters:

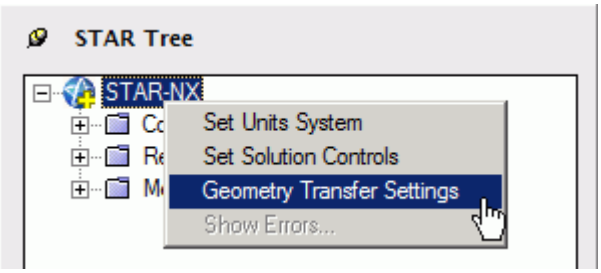
Parameter	Value
Base Size	
(**Base size option**)	**Absolute**
(*Base Size*)	`0.007 m`
Prism Layer	
(*Prism Layer Thickness*)	`10%`

3. Click **OK**.

Setting the Coincident Faces Tolerance

The geometry that is used in this tutorial has large changes in feature size and requires you to reduce the Coincident Faces Detection Tolerance.

1. Right-click STAR-NX and select **Geometry Transfer Settings**.



The *Geometry Transfer Settings* dialog appears.

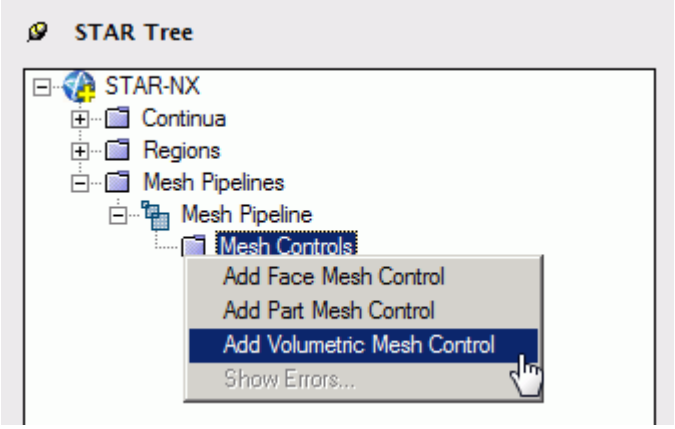
2. Set (*Coincident Faces Tolerance*) to `0.0001`.
3. Click **OK**.

Creating the Volumetric Mesh Controls

Create volumetric mesh control regions to capture the finer details of the flow near the air jets that heat the leading edge of the wing.

Create the volumetric mesh control for the inlet:

1. Right-click the **Mesh Pipelines** > **Mesh Pipeline** > **Mesh Controls** node in the STAR-NX tree and select **Add Volumetric Mesh Control**.



The *Volumetric Control* dialog appears.

2. Set *Name* to `Inlet Box`.
3. Activate the *Absolute* option.
4. Set (*Mesh Size*) to `0.0005 m`.
5. Enter the following coordinates for Point 1:

Coordinate	Value
X1	`-100.00`
Y1	`21.00`

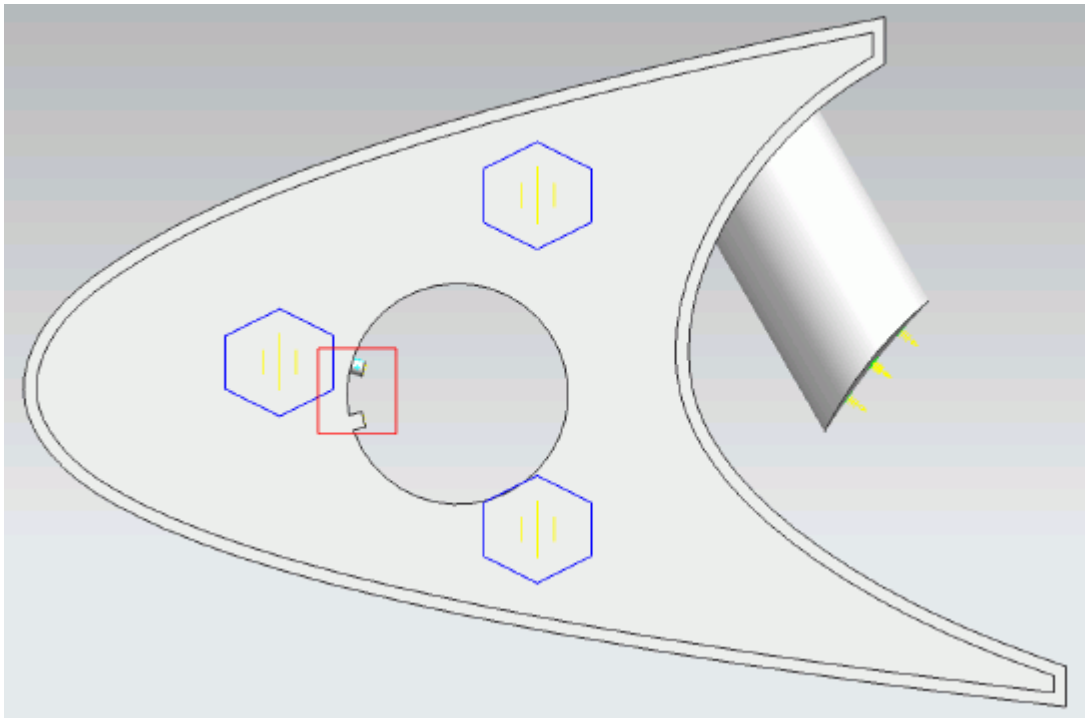
Coordinate	Value
Z1	`-20.00`

6. Enter the following coordinates for Point 2:

Coordinate	Value
X2	`-90.00`
Y2	`10.00`
Z2	`60.00`

7. Click **Create**.

   The volumetric control region is shown below.



8. Click **OK**.

Create the volumetric mesh control for the region near the piccolo tube and use the following settings:

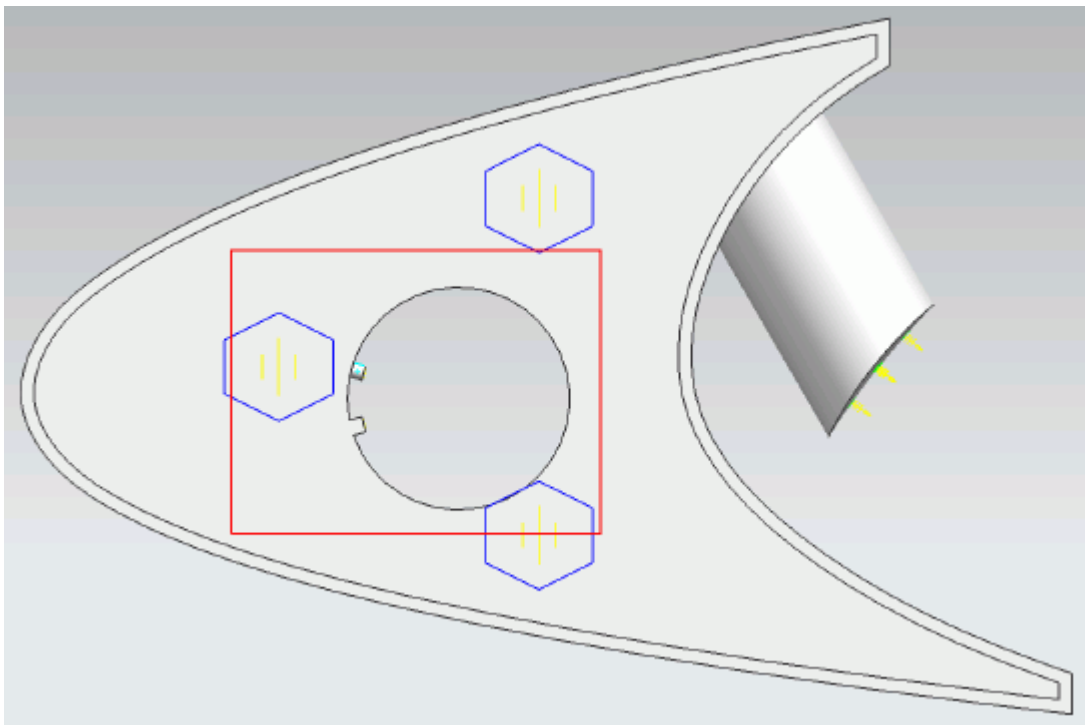
9. Using the same technique, create another volumetric mesh control with the following settings:

Parameter	Value
Name	`Mid Box`
Mesh Size	
(**Base size option**)	**Absolute**
(*Mesh Size*)	`0.0015 m`

Parameter	Value
Shape	
X1	-111.00
Y1	34.00
Z1	-20.00
X2	-64.00
Y2	-2.00
Z2	60.00

10. Click **Create**.

    The volumetric control region is shown below.

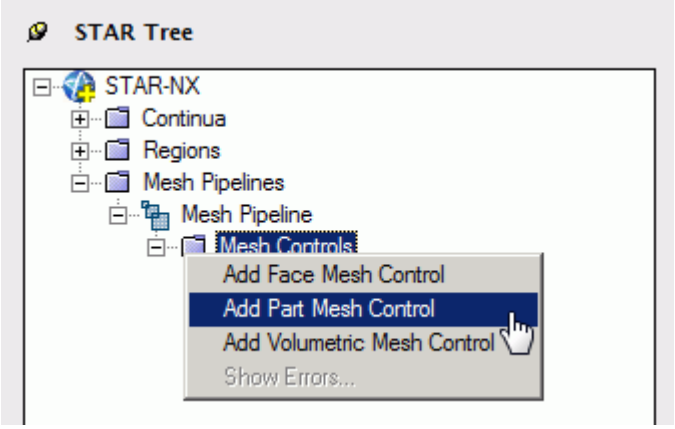


11. Click **OK**.
12. Save your progress.

Creating a Part Mesh Control

Create a part mesh control to disable the prism layer generation in the solid region. Solid regions do not require prism layers.

1. Right-click **Mesh Pipelines** > **Mesh Pipeline** > **Mesh Controls** in the STAR-NX tree and select **Add Part Mesh Control**.



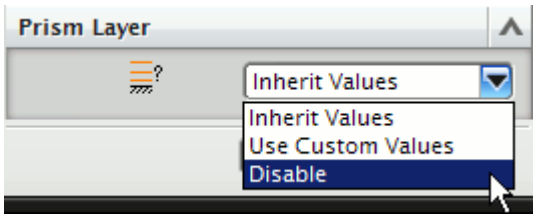
The *Part Mesh Control Settings* dialog appears.

2. Select the solid wing region on the CAD model.

   The selection is added to the *Part Selection* box.

3. Set *Name* to `Solid Region`.

4. In the *Prism Layer* box, set (*Prism layer option*) to **Disable** using the drop-down menu.

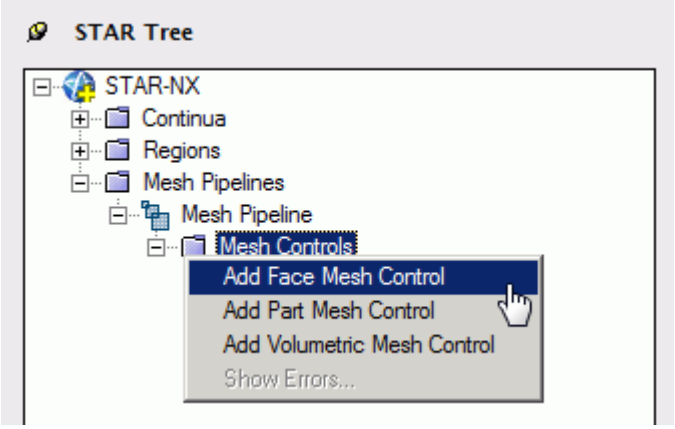


5. Click **OK**.

Creating a Face Mesh Control

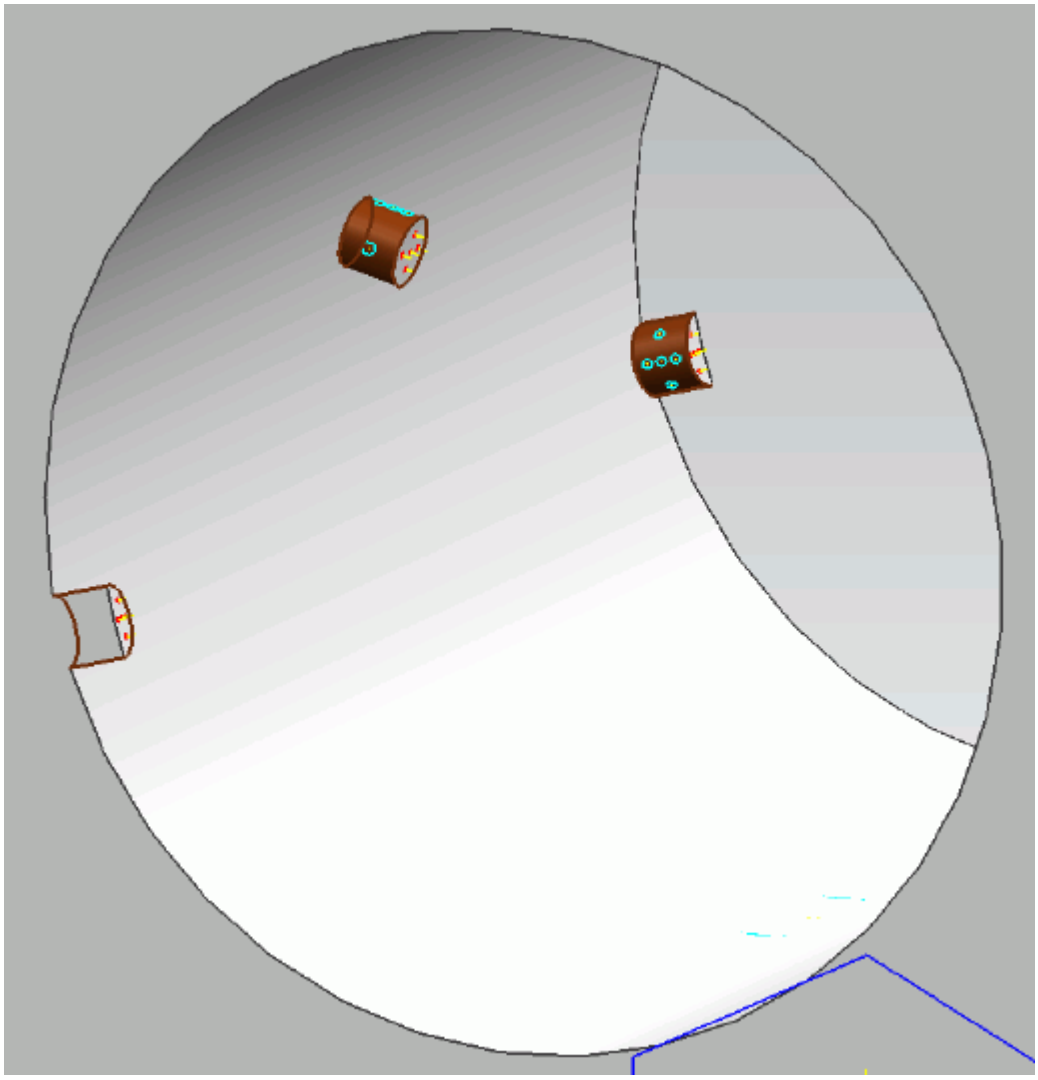
Create a face mesh control to reduce the size of the prism layer thickness at the inlet boundary.

1. Right-click **Mesh Pipelines** > **Mesh Pipeline** > **Mesh Controls** in the STAR-NX tree and select **Add Face Mesh Control**.



The *Face Mesh Control Settings* dialog appears.

2. Select the cylindrical faces that are perpendicular to the inlet boundary, as shown below.



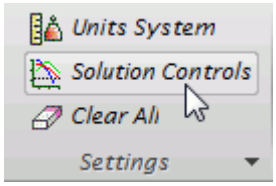
3. Set *Name* to `Inlet Refinement`.

4. Activate the *Custom Mesh Size* option.
5. Activate the *Absolute* option.
6. Set (*Mesh Size*) to `0.00025` m.
7. In the *Prism Layer* box, set (*Prism Layer Option*) to **Use Custom Values**.
8. Set *Layer Thickness Option* to **Absolute**.
9. Set (*Layer Thickness*) to `0.0002` m.
10. Click **OK**.

Modifying Solution Controls

Modify the maximum number of iterations and the solution accuracy. To help accelerate convergence, edit the solid under-relaxation factor:

1. Click **Solution Controls** on the STAR-NX ribbon.



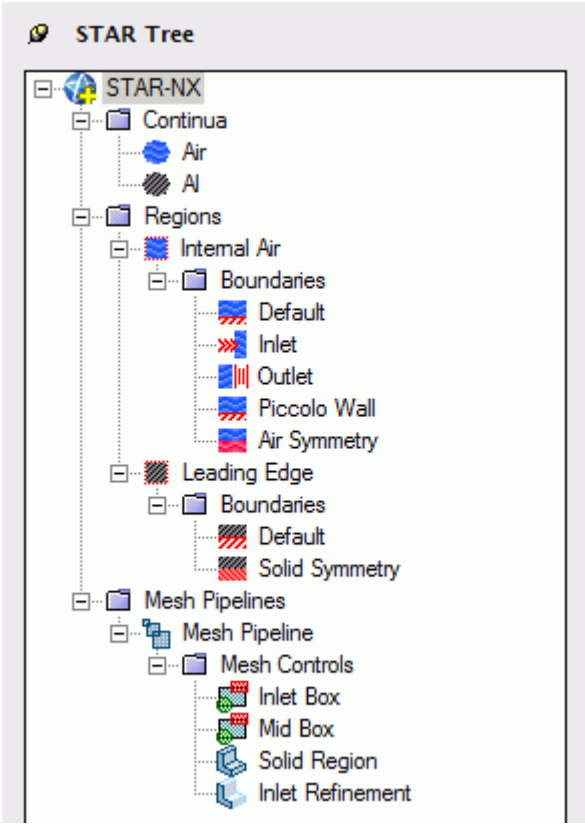
   The *Solution Controls* dialog appears.
2. Set the following parameters:

Parameter	Value
(*Number of Iterations*)	`500`
(*Convergence Level*)	`0.01`
(*Solid Temperature Under-Relaxation Factor*)	`0.9999`

3. Click **OK**.
4. Save your progress.

Reviewing the Simulation Set Up

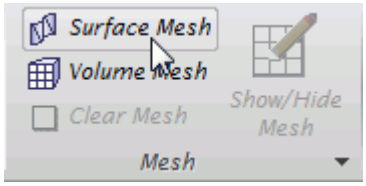
The final STAR-NX object tree is shown below.



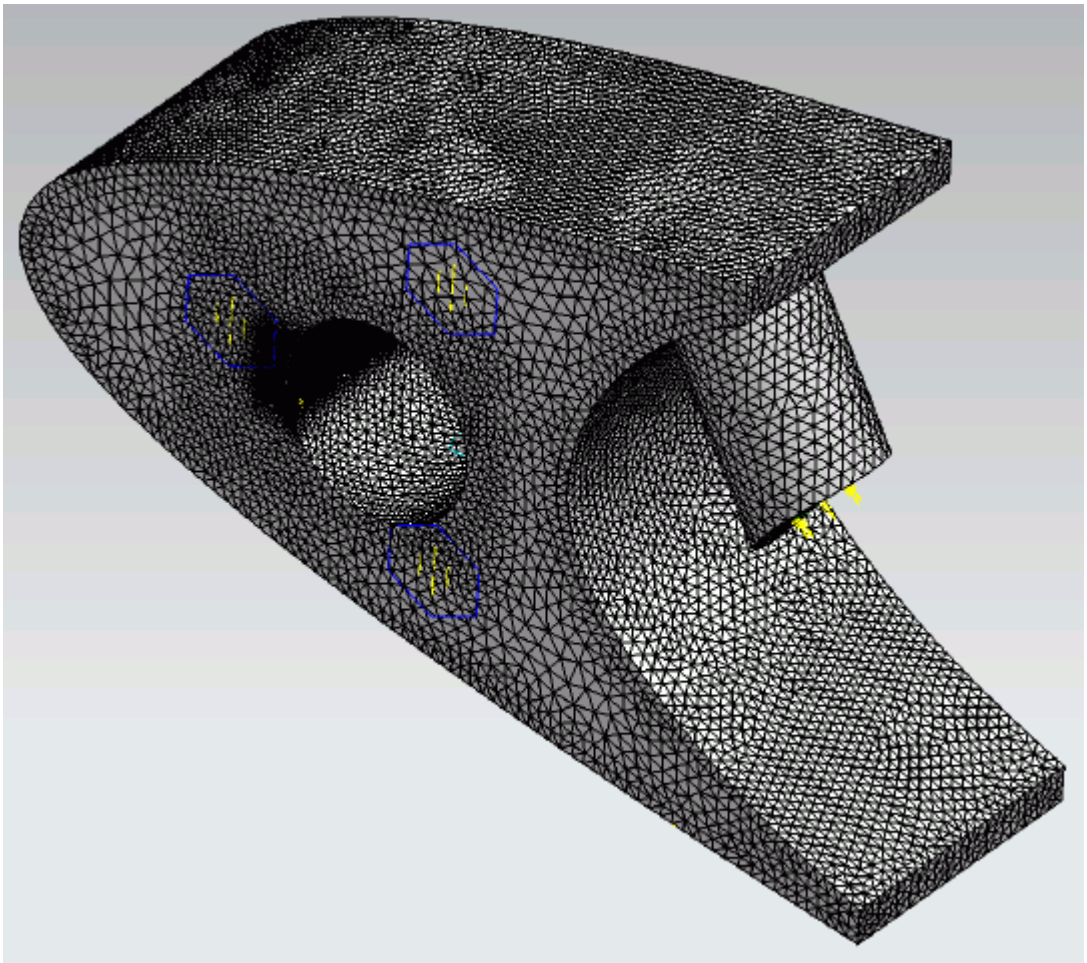
Generating the Surface Mesh

Generate the surface mesh:

1. From the STAR-NX ribbon, click **Surface Mesh**.



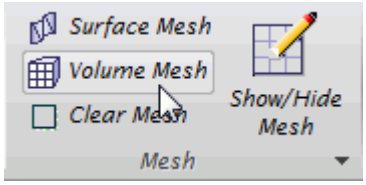
The surface mesh is shown below.



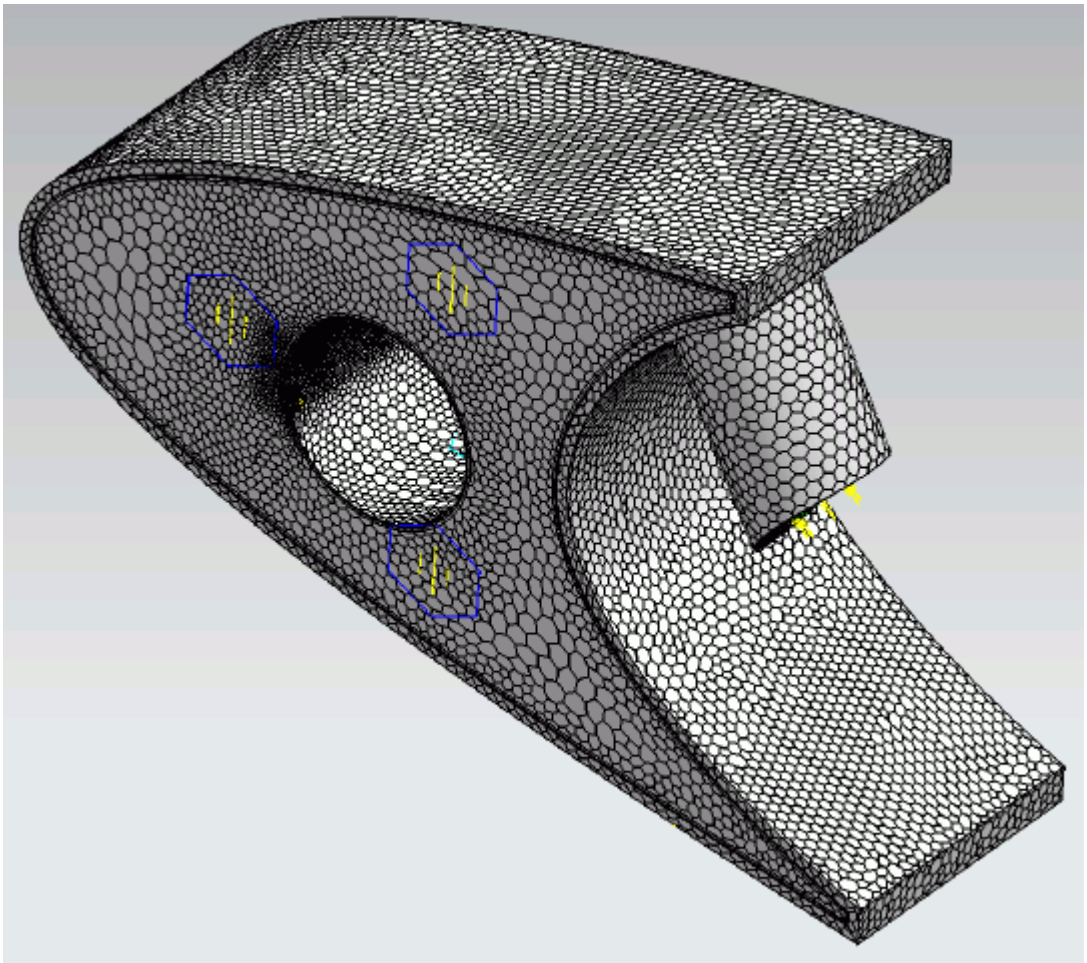
Generating the Volume Mesh

Generate the volume mesh:

1. From the STAR-NX ribbon, click **Volume Mesh**.



The volume mesh is shown below.

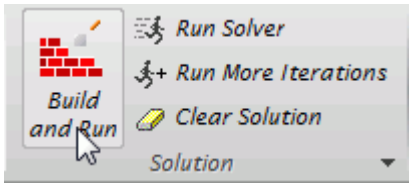


Building and Running

The case is now ready to run.

To run the simulation:

1. From the ribbon, click **Build and Run**.



The *Residuals* plot appears in the STAR-NX *Output* window. You can use this plot to monitor the convergence of the solution.

When the solver is done, STAR-CCM+ launches.

Post-Processing Results

You perform all post-processing in STAR-CCM+. By default, a scalar scene, *Scalar Scene 1*, automatically appears in the *Graphics* window when the solver finishes. This scene displays the absolute total pressure.

The results that are contained in the following sections are representative. You may see some variation in the results that you obtain, depending on what platform and package you are using.

1. In STAR-CCM+, click **File** > **Save** to save the simulation.

Contents:

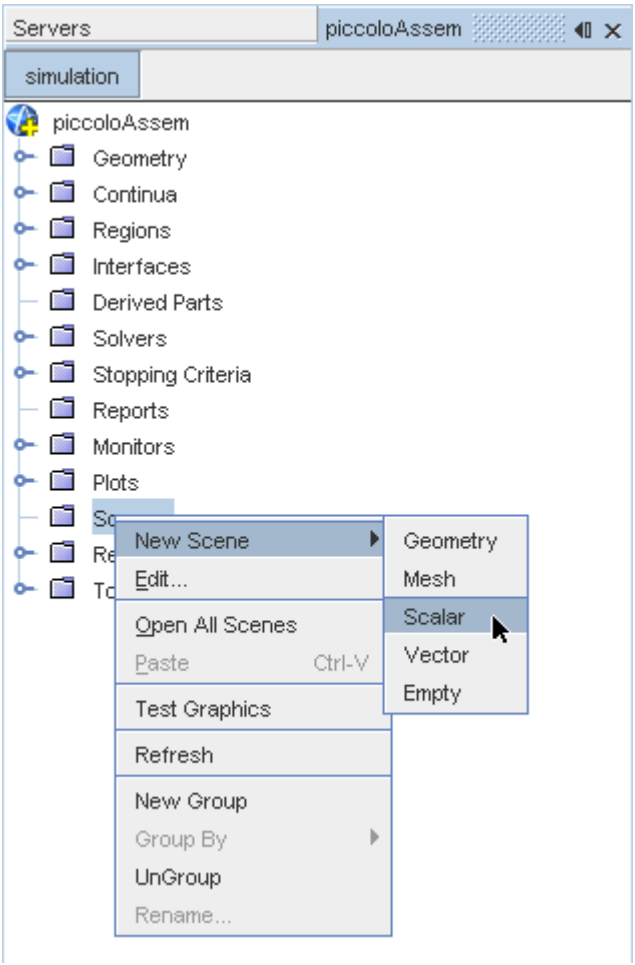
Creating a Scalar Scene
Creating A Streamlines Scene
Creating a Vector Scene
Creating a Streamlines Scene
Creating A Scalar Scene
Visualizing the Wing Skin Temperature
Creating a Streamlines Scene
Creating A Scalar Scene
Creating a Streamlines Scene
Creating a Scalar Scene
Modifying The Scalar Scene
Creating a Streamlines Scene

Creating A Scalar Scene

Create a scalar scene containing a cross-section of the model. Plot temperature on this section to demonstrate how the hot gas coming from the piccolo tube affects the wing skin and the air inside the wing.

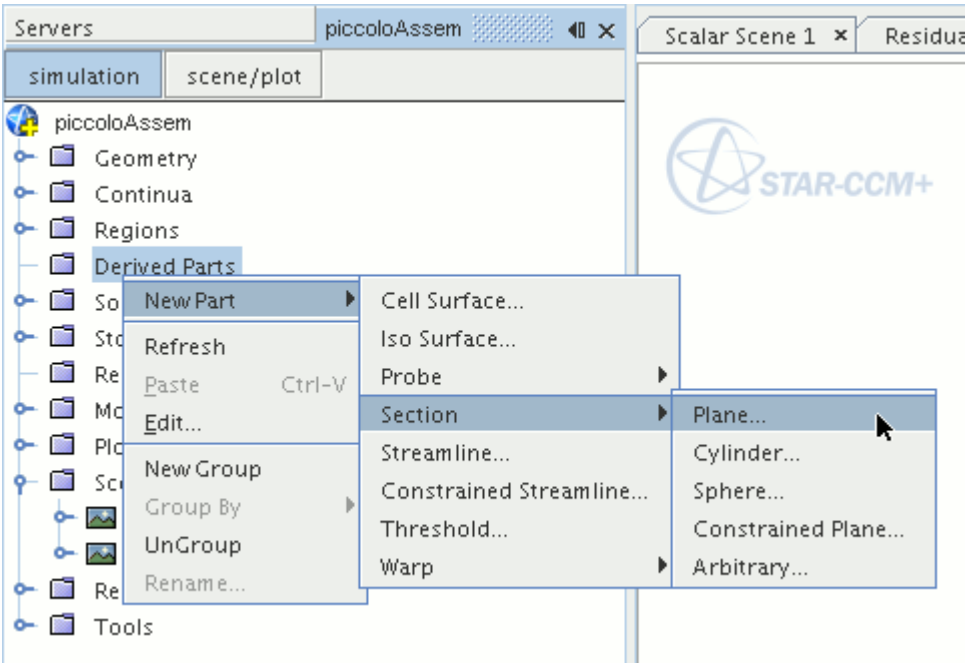
To create the scene:

1. Right-click the **Scenes** node in the explorer pane and select **New Scene** > **Scalar**.



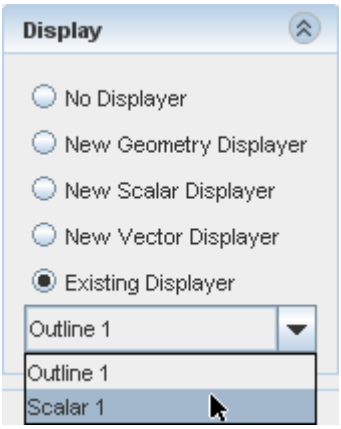
A new scalar scene, *Scalar Scene 2*, is created and selected in the *Graphics* window.

2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane...**.



The *Create Plane Section* dialog appears.

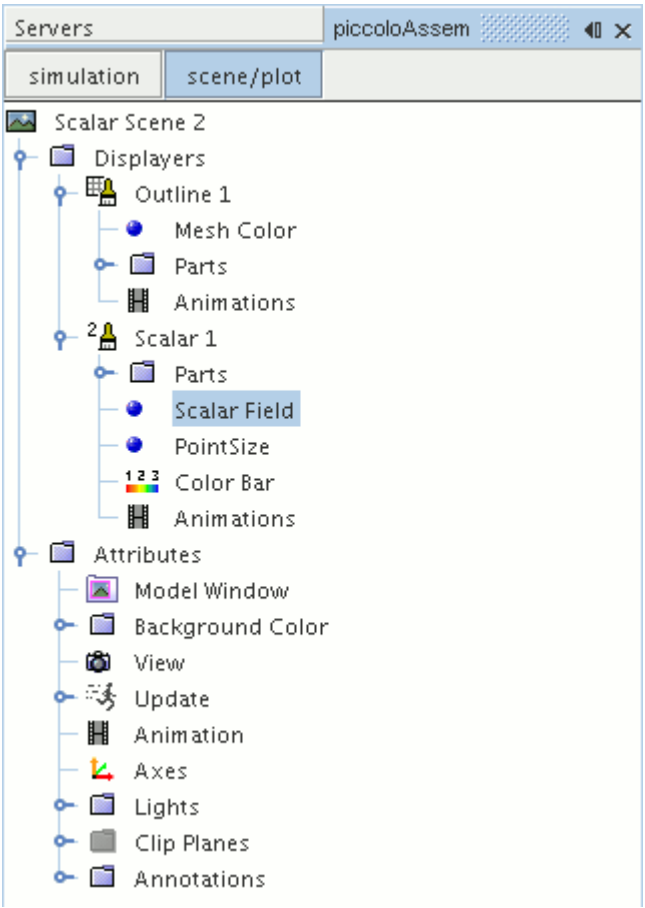
3. Set the origin coordinates to `[0.0 mm, 0.0 mm, 18.4 mm]`.
4. Set the *normal* vector to `[0.0 mm, 0.0 mm, 1.0 mm]`.
5. In the *Display* group box, select **Existing Displayer**, and select **Scalar 1** from the drop-down menu.



6. Click **Create** followed by **Close**.

Change the scalar field to display temperature contours:

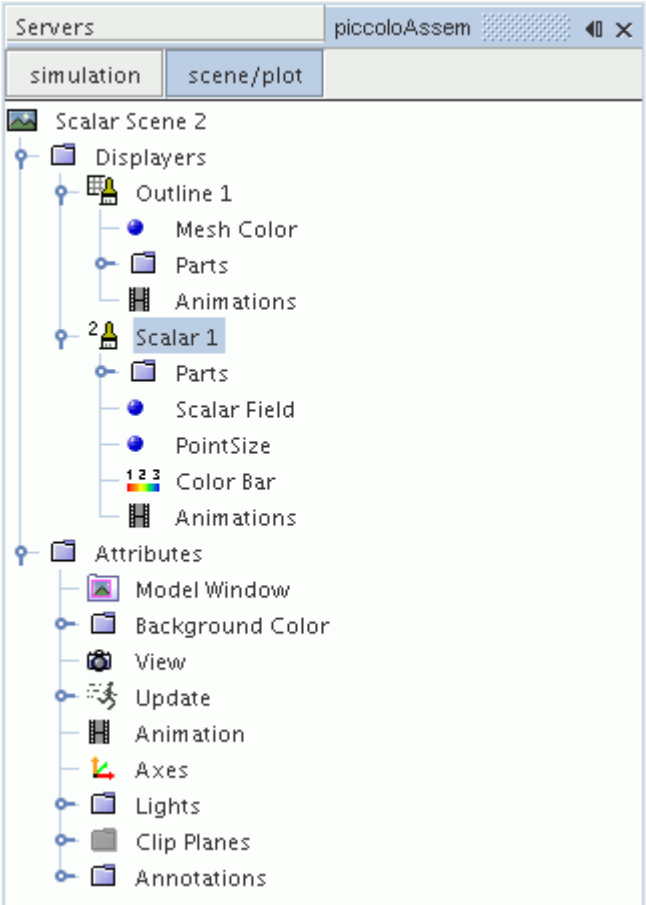
7. Click **scene/plot**.
8. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Scalar Field**.



9. Set *Function* to **Temperature**.

Modify the contour style to have a smooth transition between the different temperature values in the scene:

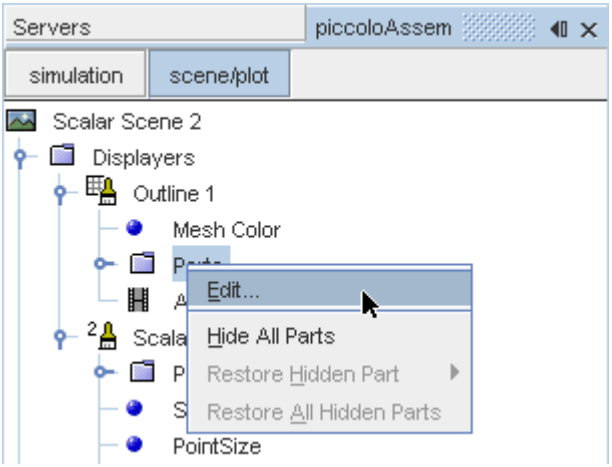
10. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1**.



11. Set *Contour Style* to **Smooth Filled**.

Adjust the outlines in the scene to show only the plane section:

12. Right-click **Scalar Scene 2** > **Displayers** > **Outline 1** > **Parts** and select **Edit...**.

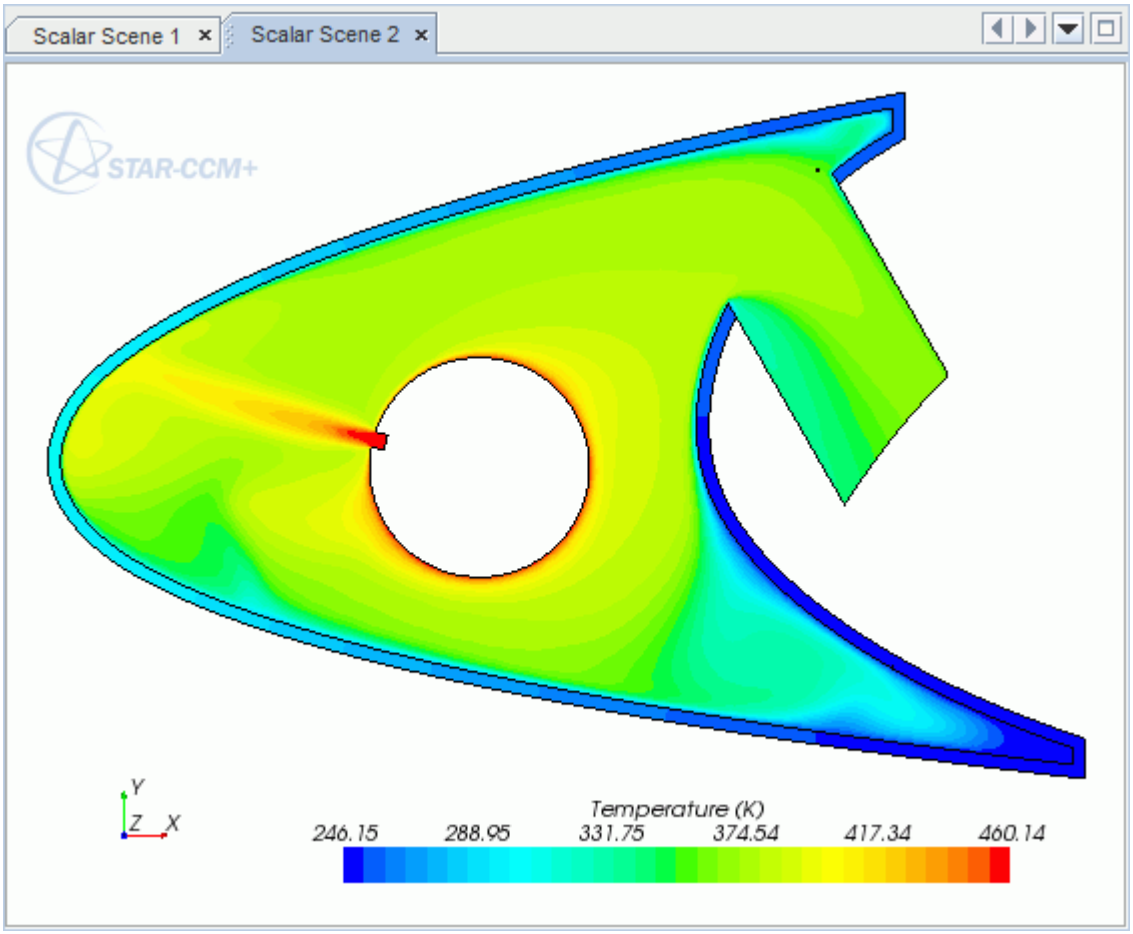


The *Parts* dialog appears.

13. Right-click on a blank area and choose **Deselect All**.

14. Select the **Derived Parts** > **plane section** node.
15. Click **OK**.
16. Position the scene to show the side view of the geometry.
17. Click **simulation**.
18. Click (**Save**) to save the simulation.

The finished scene is shown below.



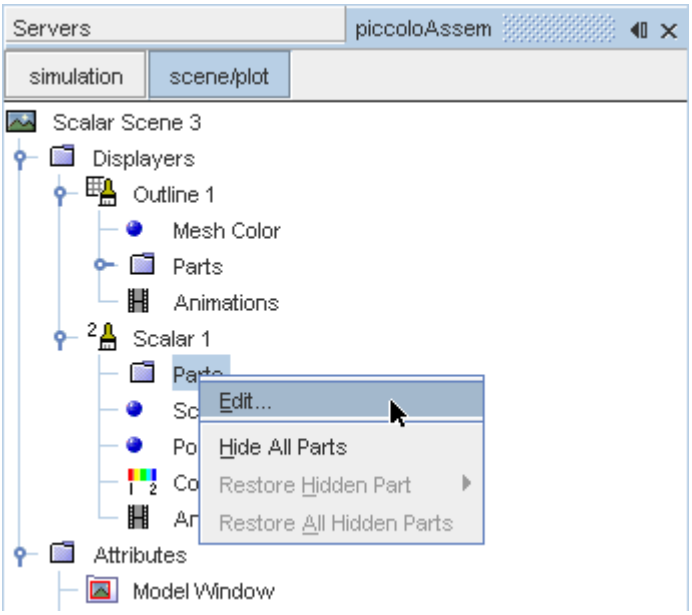
Visualizing the Wing Skin Temperature

Create another scalar scene to display the temperature on the wing skin geometry.

To create another scalar scene:

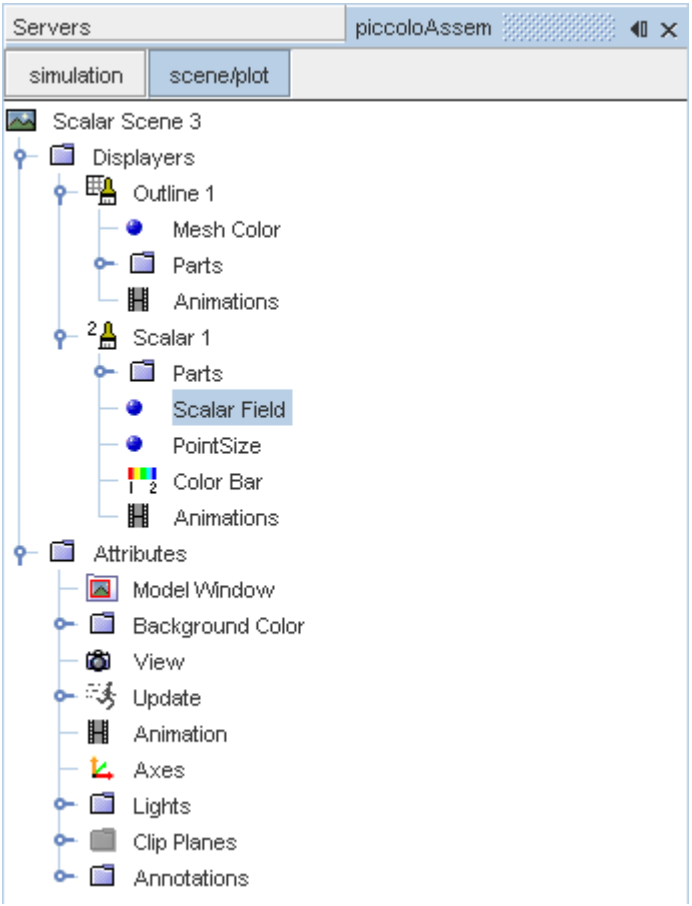
1. Right-click the **Scenes** node in the explorer pane and select **New Scene** > **Scalar**.

   A new scalar scene, *Scalar Scene 3*, is created and selected in the *Graphics* window.
2. Click **scene/plot**.
3. Right-click **Scalar Scene 3** > **Displayers** > **Scalar 1** > **Parts** and select **Edit...**.



The *Parts* dialog appears.

4. Expand the **Regions** node and select the **Leading Edge** node.
5. Click **OK**.
6. Navigate to **Scalar Scene 3** > **Displayers** > **Scalar 1** > **Scalar Field**.



7. Set *Function* to **Temperature.**

To have a smooth transition between the different temperature values, edit the contour style:

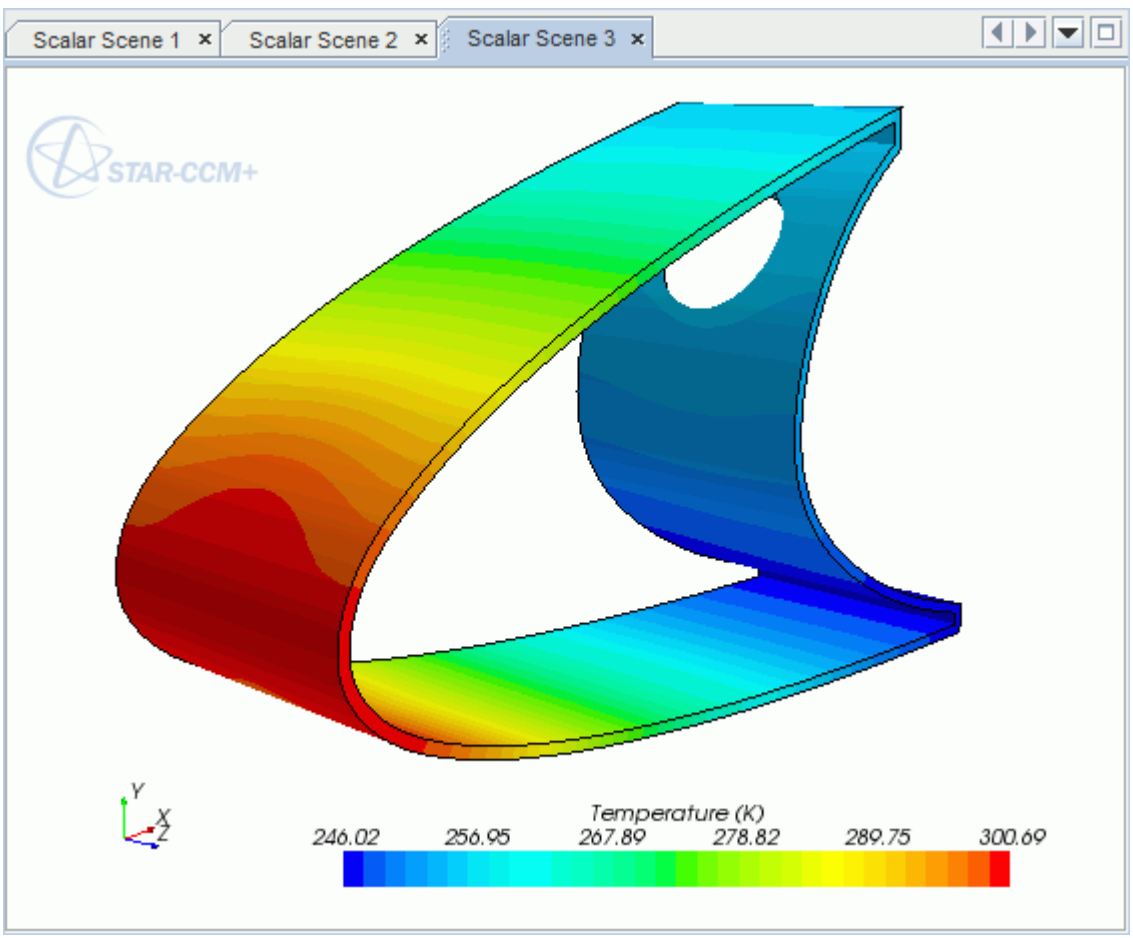
8. Navigate to **Scalar Scene 3** > **Displayers** > **Scalar 1**.
9. Set *Contour Style* to **Smooth Filled**.

Adjust the outlines to appear on the solid region only:

10. Right-click **Scalar Scene 3** > **Displayers** > **Outline 1** > **Parts** and select **Edit...**.

    The *Parts* dialog appears.
11. Expand the **Regions** node and deselect the **Internal Air** node.
12. Click **OK**.
13. Reorient the geometry to get a better view of the leading edge.
14. Click **simulation**.
15. Save the simulation.

The finished scene is shown below.

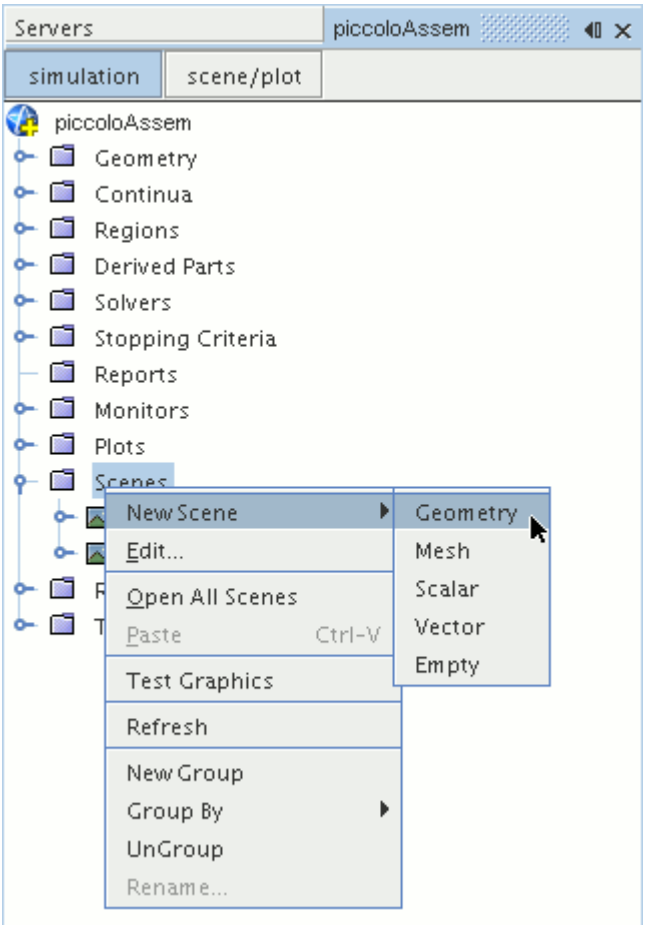


Creating a Streamlines Scene

Create streamlines to get a better understanding of how the hot air flows inside the leading edge of the wing.

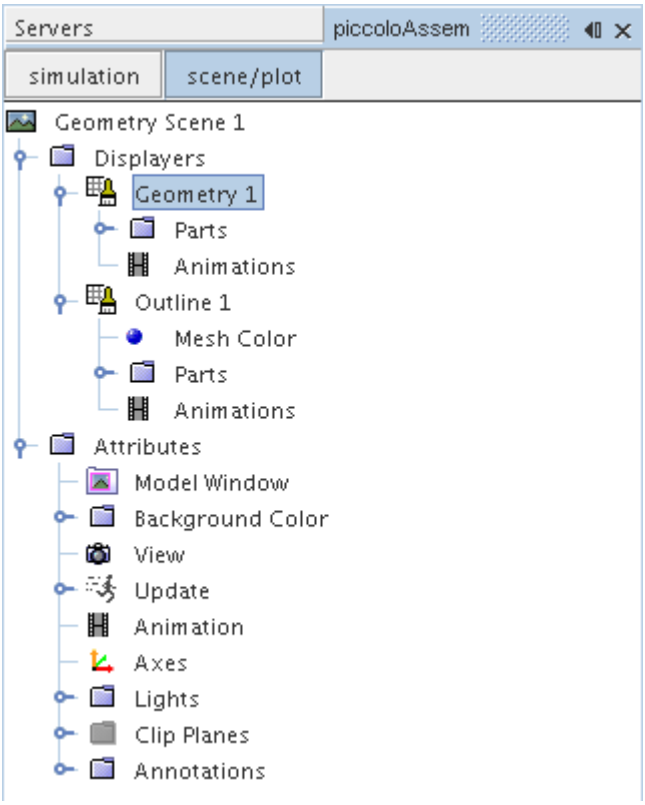
First, create a geometry scene:

1. Right-click the **Scenes** node in the object tree and select **New Scene** > **Geometry**.



   A new geometry scene, *Geometry Scene 1*, is created and selected in the *Graphics* window.

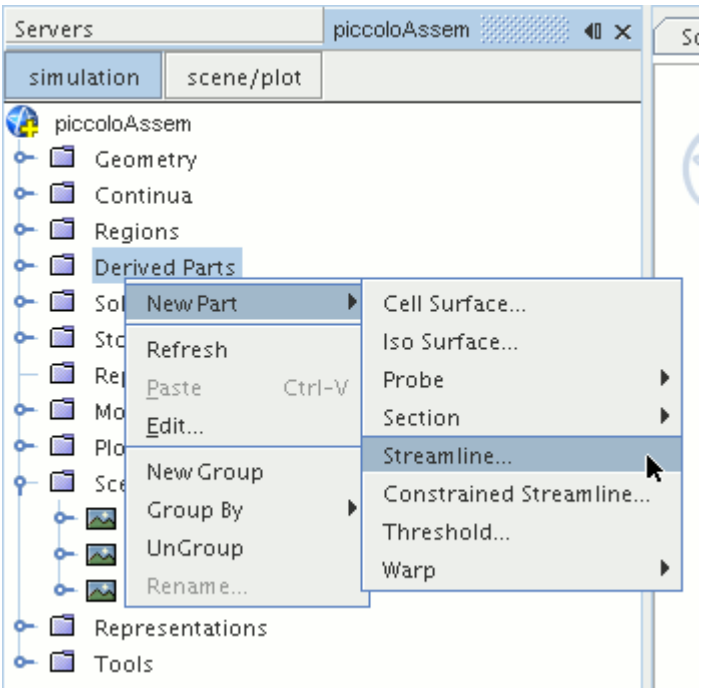
2. Click **scene/plot**.
3. Navigate to the **Geometry Scene 1** > **Displayers** > **Geometry 1** node.



4. Set *Opacity* to `0.1`.

Next, create the streamlines:

5. Click **simulation**.
6. Right-click the **Derived Parts** node and select **New Part** > **Streamline...**.

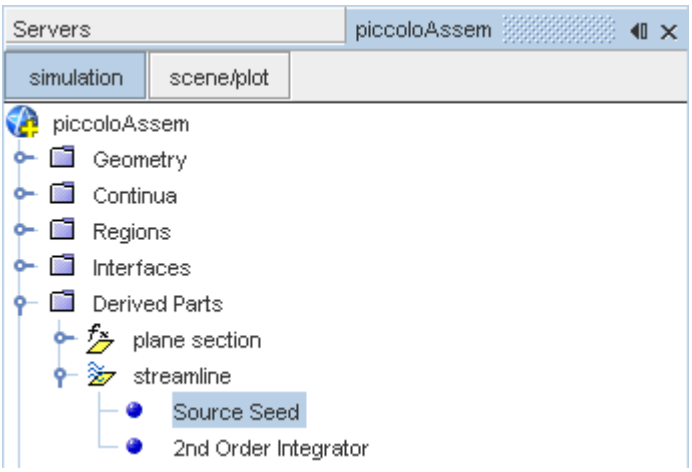


The *Create Streamline* dialog appears.

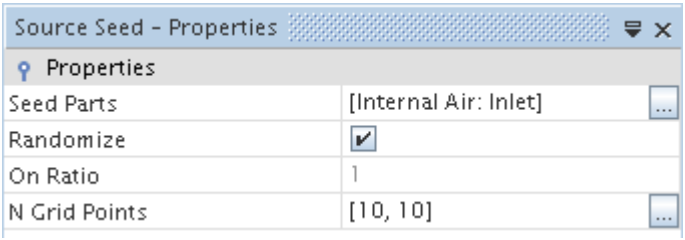
7. In the *Input Parts* box, click **Select...**.
8. In the *Select Objects* dialog, expand the **Regions** node and deselect **Leading Edge**.
9. Click **OK** to close the *Select Objects* dialog.
10. Make sure that **Part Seed** is selected in the *Seed Mode* section.
11. Set *Seed Parts* to **Regions** > **Internal Air** > **Boundaries** > **Inlet**.
12. Set *Part U-Resolution* to `10`.
13. Set *Part V-Resolution* to `10`.
14. Make sure that **New Streamline Displayer** is selected in the *Display* section.
15. Click **Create** followed by **Close**.

The streamlines do not yet appear in the geometry scene. Follow the steps that are outlined below to modify the streamlines:

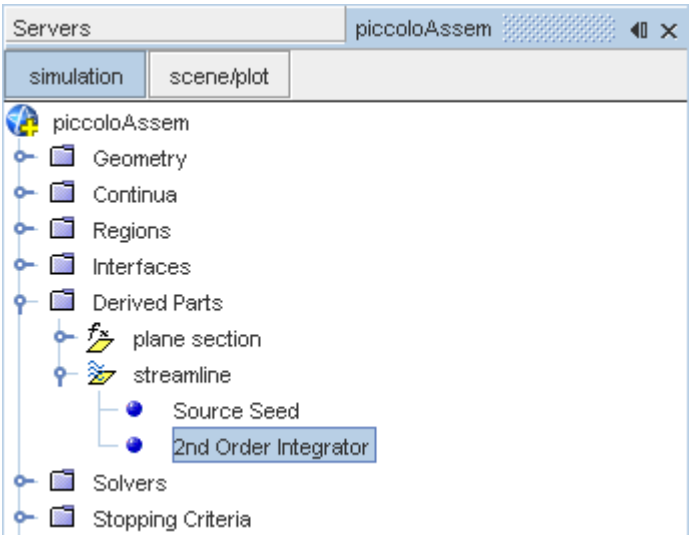
16. Navigate to **Derived Parts** > **Streamline** > **Source Seed**.



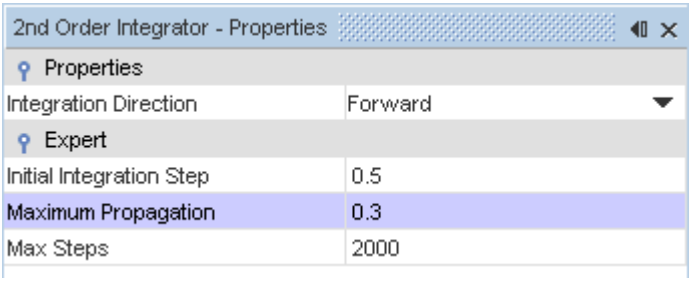
17. Activate the *Randomize* option.



18. Navigate to **Derived Parts** > **Streamline** > **2nd Order Integrator**.



19. Set *Maximum Propagation* to `0.3`.



Adjust the appearance of the streamlines to display tubes instead of the default lines:

20. Click **scene/plot**.
21. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1**.
22. Set *Mode* to **Tubes** and *Width* to `0.00025`.

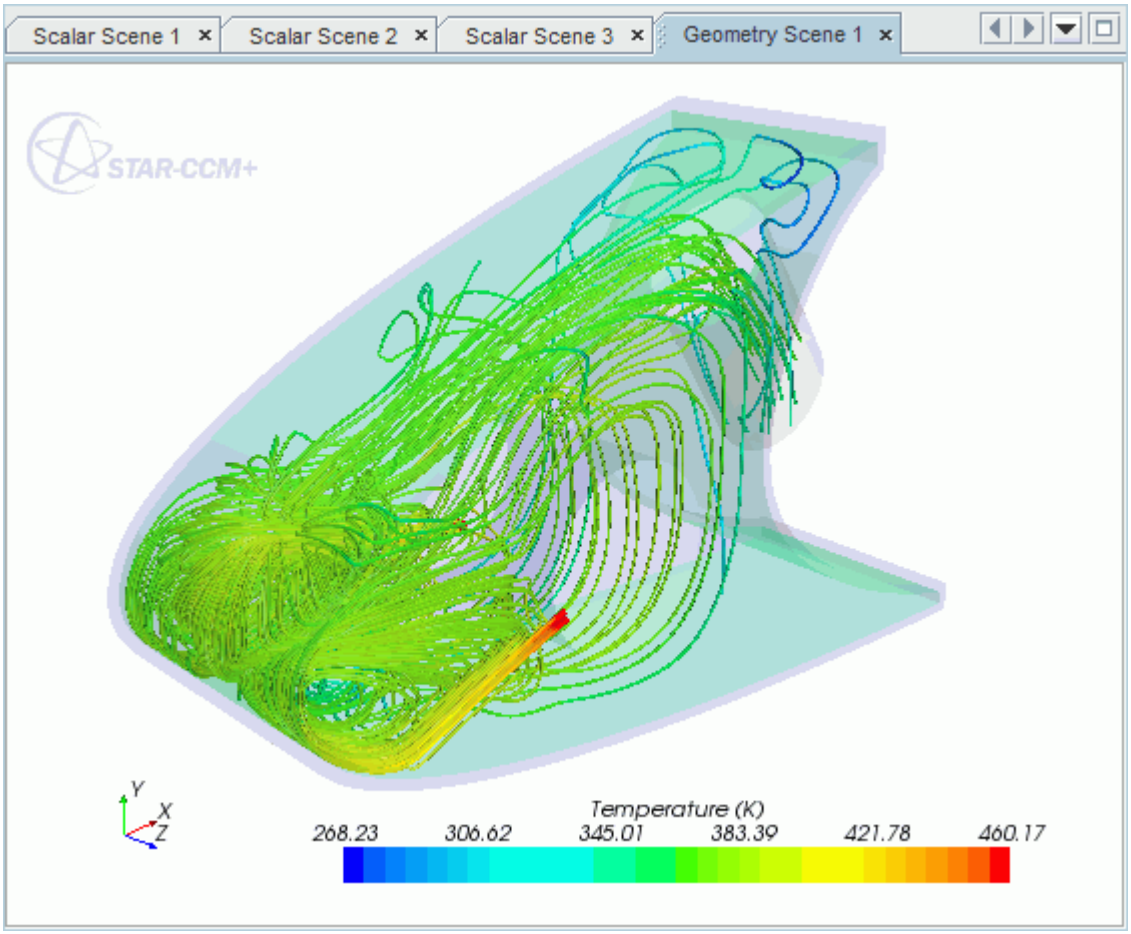
Adjust the scalar field of the streamlines to display temperature:

23. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** > **Scalar Field**.
24. Set *Function* to **Temperature**.

Turn off the outlines in the geometry scene to have a better view of the streamlines.

25. Right-click the **Geometry Scene 1** > **Displayers** > **Outline 1** and select **Toggle Visibility**.
26. Reorient the geometry to get a better view of the leading edge.
27. Save the simulation.

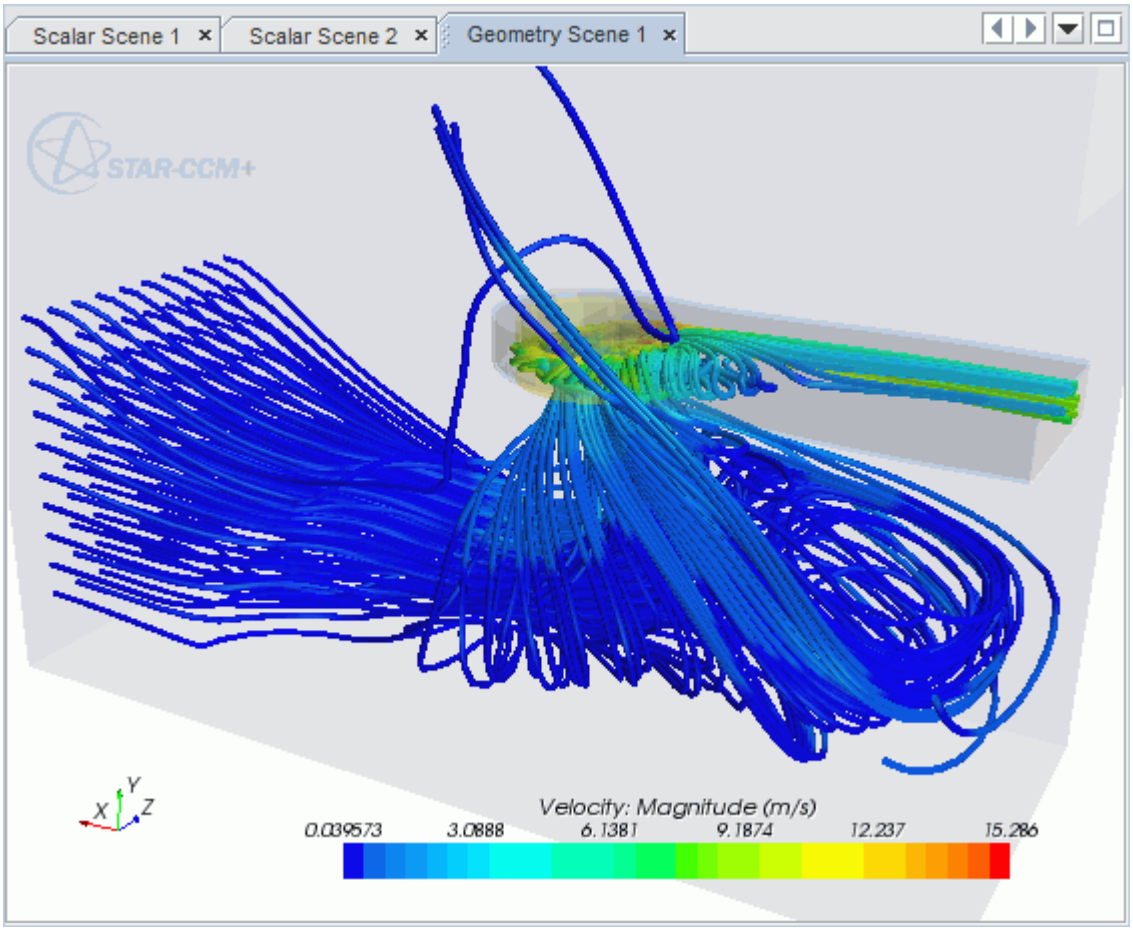
The finished scene is shown below.


Scalar Scene 1
Scalar Scene 2
Scalar Scene 3
Geometry Scene 1
STAR-CCM+
Y
X
Z
Temperature (K)
268.23
306.62
345.01
383.39
421.78
460.17

Rotating Flow Inside a Graphics Card

This tutorial takes you through the steps that are required to run a CFD calculation in the STAR-NX environment.

For this tutorial, the model is a fully sealed graphics card that is located inside an empty, idealized PC tower case. The geometry that is used in this tutorial has been simplified on order to decrease the computational cost. As a consequence the mesh is relatively coarse at approximately 200,000 cells compared to the 3-4 million cells used for an industrial analysis.



Contents:

Opening the CAD Model
Selecting Fluid Physics Models
Setting the Rotating Region
Defining Boundary Conditions
Setting the Mesh Pipeline
Creating the Volumetric Mesh Controls
Modifying Solution Controls

Reviewing the Simulation Set Up
Generating The Surface Mesh
Generating the Volume Mesh
Building and Running
Post-Processing Results

Opening the CAD Model

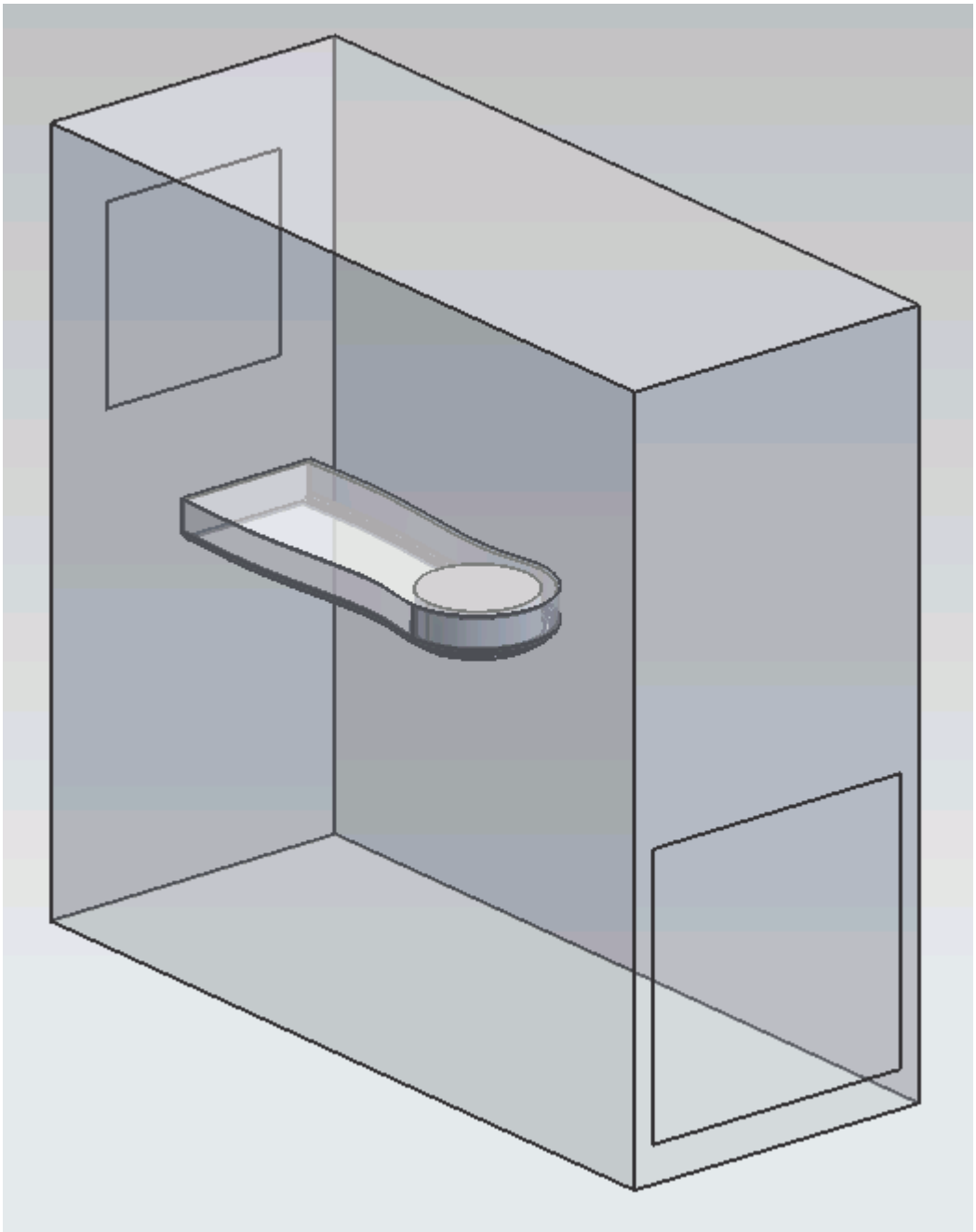
A directory containing the CAD geometry is provided with the STAR-CCM+ distribution.

This directory is located in `[STAR-CCM+_INSTALL_DIR]\doc\startutorialsdata\starnx\data`.

1. Copy directory `data` to your own disk space and open the part file `gpuAssem.prt` in STAR-NX.

   The assembly is made up of two parts: the PC tower case (including the GPU housing), and the rotating region that encloses the blower fan of the GPU.

   The imported geometry is shown below.



The STAR-NX menu appears at the top of the screen.The STAR-NX object tree is located on the left of the screen.

To switch to this object tree:

2. Click (**STAR Tree**).

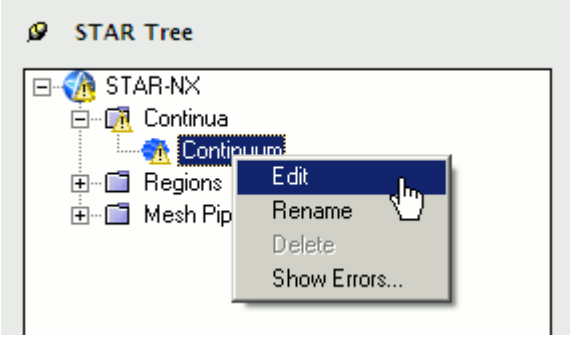
If these items are not visible, refer to Launching STAR-NX Manually.

Selecting Fluid Physics Models

The first step in setting up this tutorial is to define the physics of the problem. For this case, the flow is turbulent and isothermal. Temperature calculations are on by default in STAR-NX.

To turn temperature calculations off:

1. Right-click the **Continua** > **Continuum** node in the STAR-NX tree and select **Edit**.

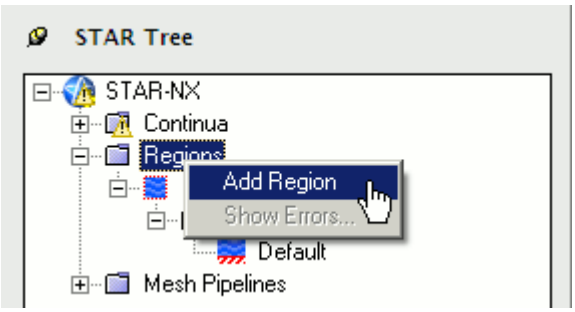


   The *Continuum Settings* dialog appears.
2. Deactivate the **Temperature** option in the *Optional Physics Models* box.
3. Click **OK**.

Setting the Rotating Region

Create the rotating region that contains the GPU fan:

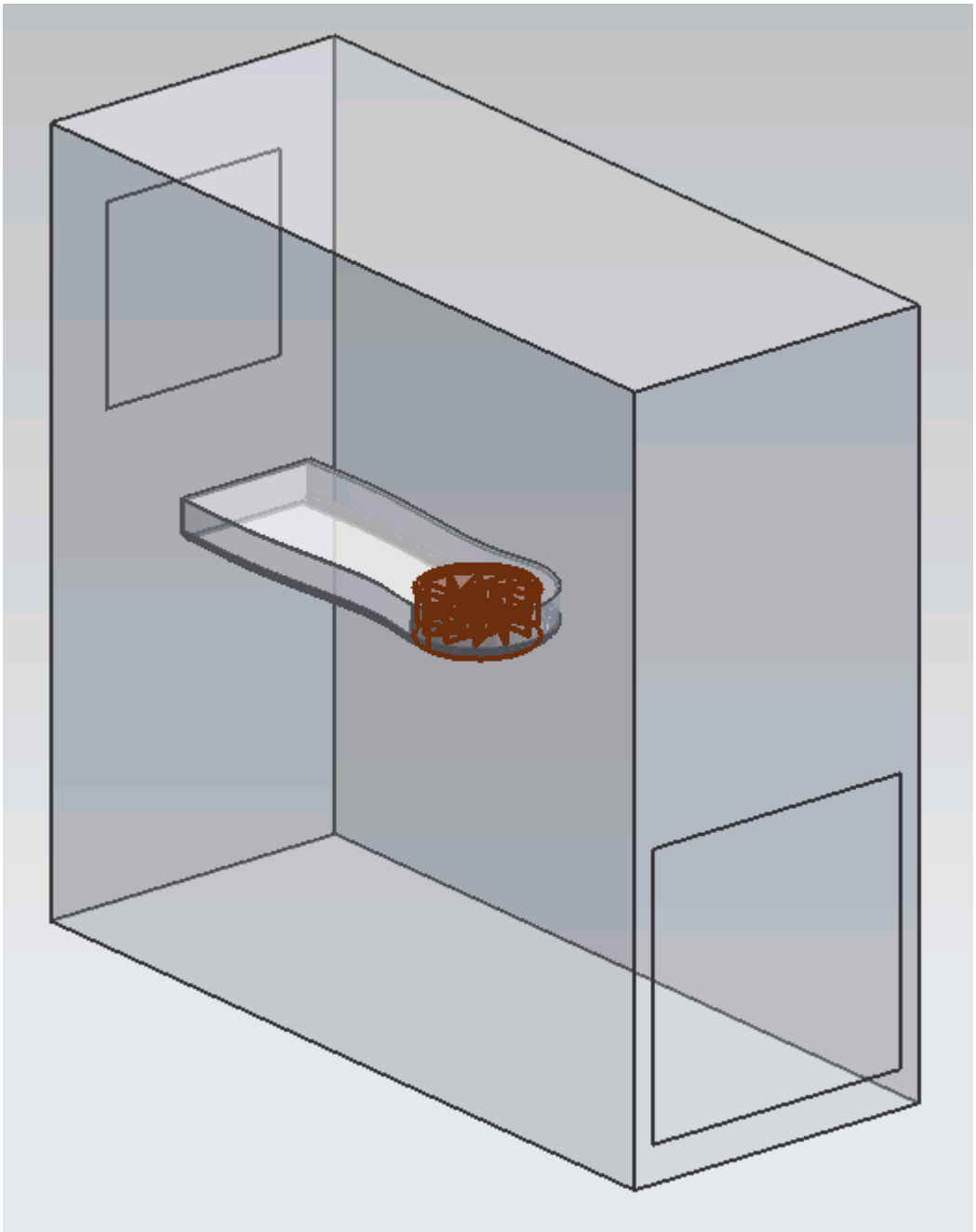
1. Right-click the **Regions** node in the STAR-NX object tree and select **Add Region**.



   The *Region Settings* dialog appears.
2. Select the disc that is located inside the graphics card housing.

   You can select this part from within the PC tower by right-clicking the disc and selecting **Select from List...** > **Solid Body in GPUBLOWERFAN** from the *Quick Pick* dialog.

   Once selected, the part becomes highlighted as shown below.



3. Set *Name* to `Rotating Region`.
4. Activate the **Moving Reference Frame** option.
5. Set (*Rotational Velocity*) to `-2800 rpm`.

   Note: Make sure that you set the units to `rpm` using the drop-down menu before entering the value.

6. Set the origin of the axis of rotation to:

Property	Value
X	`-122.1761`

Property	Value
Y	-1.9434
Z	-19.0609

7. Set the components of the axis of rotation to:

Property	Value
(*i component*)	0
(*j component*)	1
(*k component*)	0

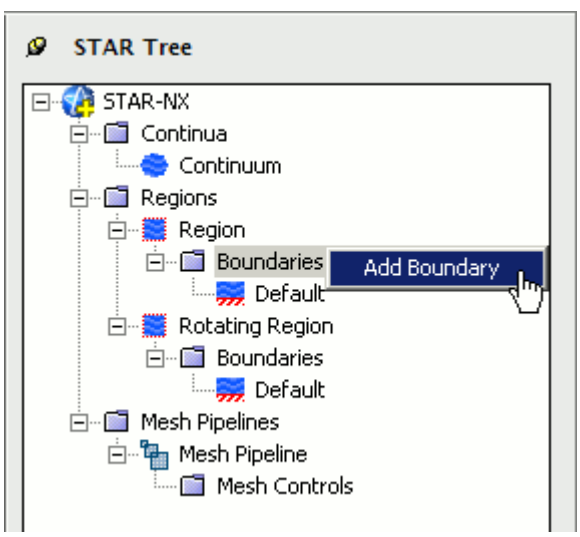
8. Make sure that the *Use Assembly Coordinate System* is activated.
9. Click **OK**.

Defining Boundary Conditions

Create the inlet and outlet boundary conditions for this simulation.

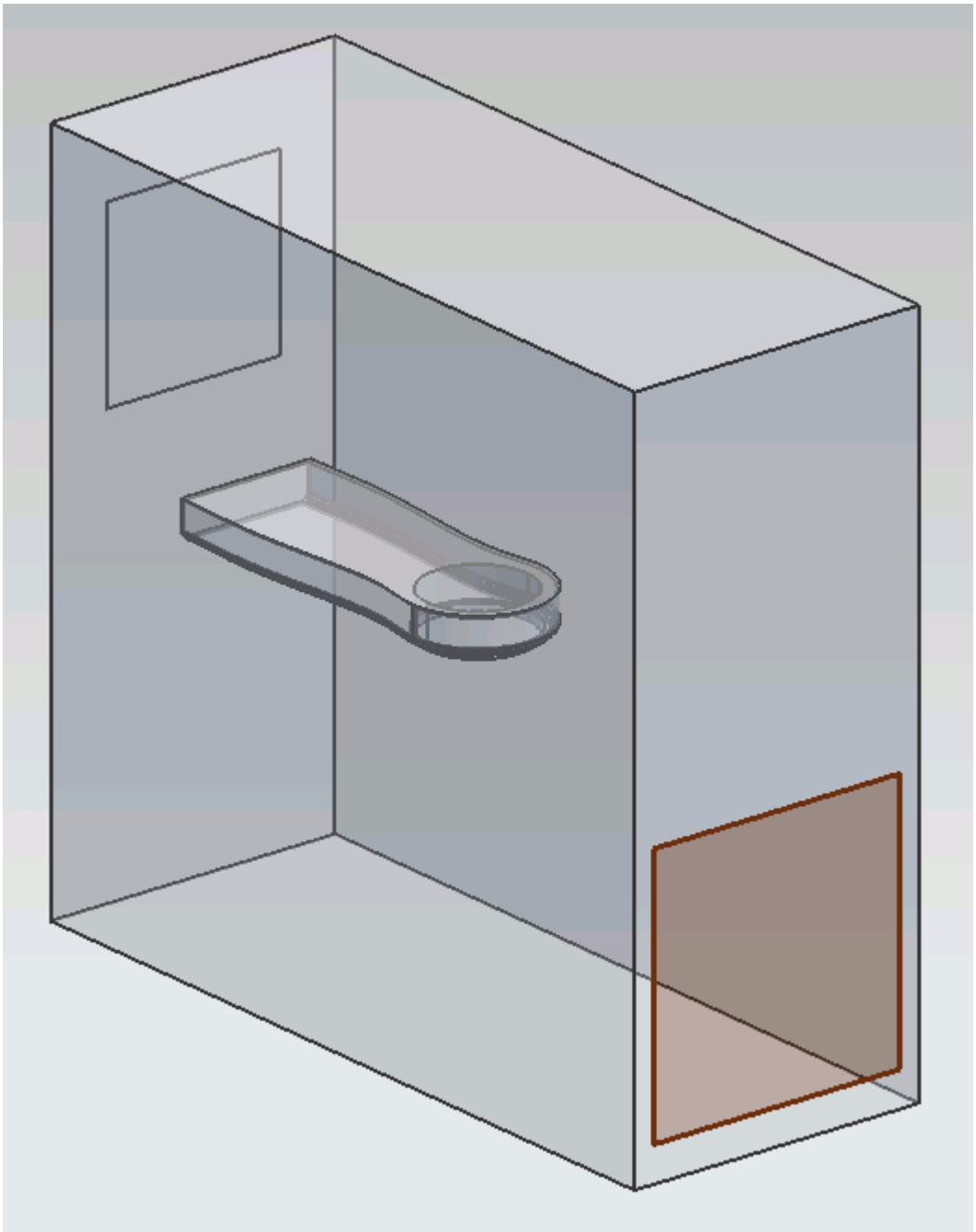
Create the inlet boundary first:

1. Right-click the **Regions** > **Region** > **Boundaries** node and select **Add Boundary**.



The *Boundary Settings* dialog appears.

2. Select the face that is highlighted below.



These selections are reported in the *Face Selection* box in the *Boundary Settings* dialog.

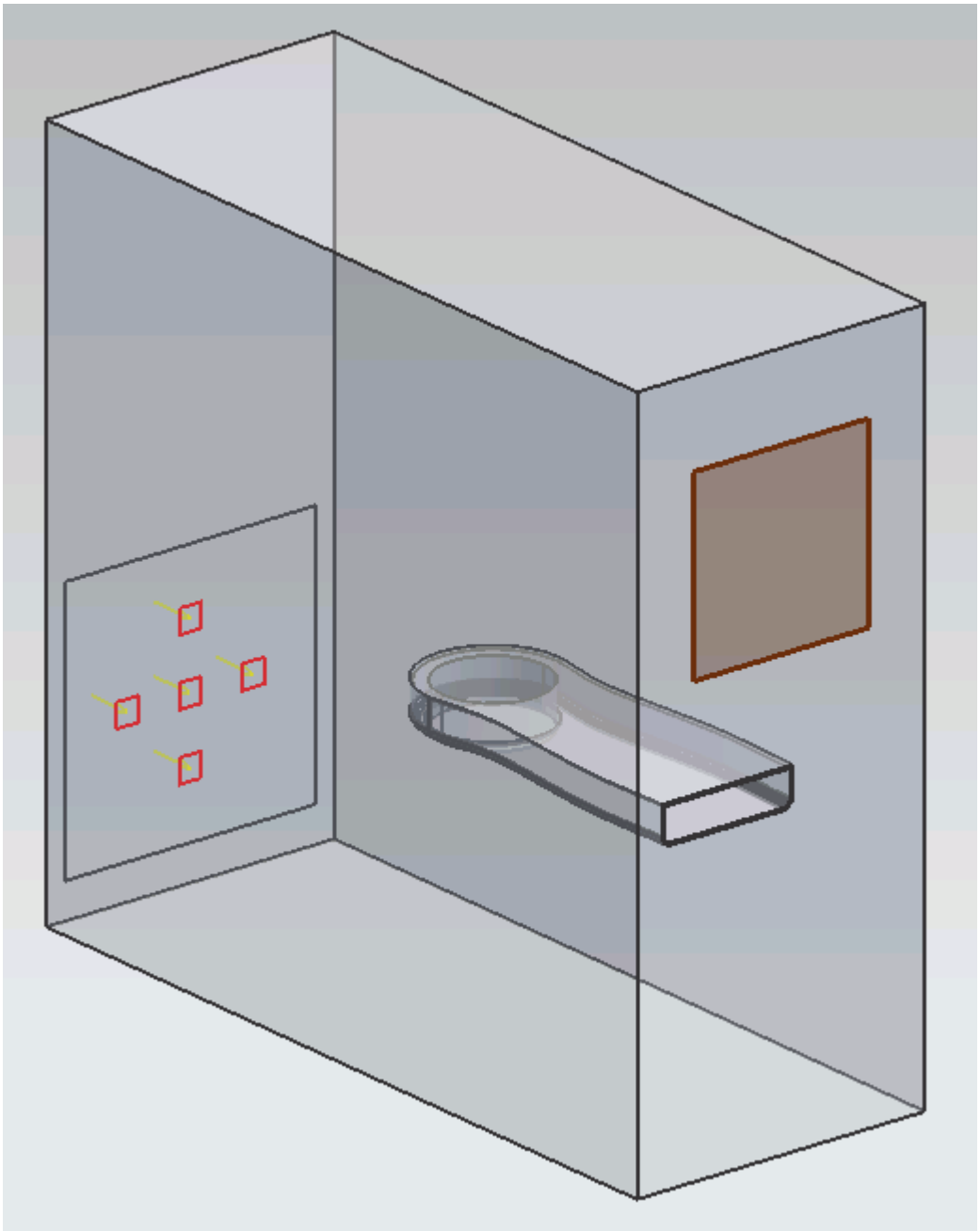
3. Set the following settings:

Parameter	Value
Type	**Velocity Inlet**
Name	`Case Inlet`
(*Normal Velocity*)	`0.1 m/s`

4. Click **OK**.

Create the case outlet boundary:

5. Using the same technique, create another boundary.
6. Select the face that is highlighted below.



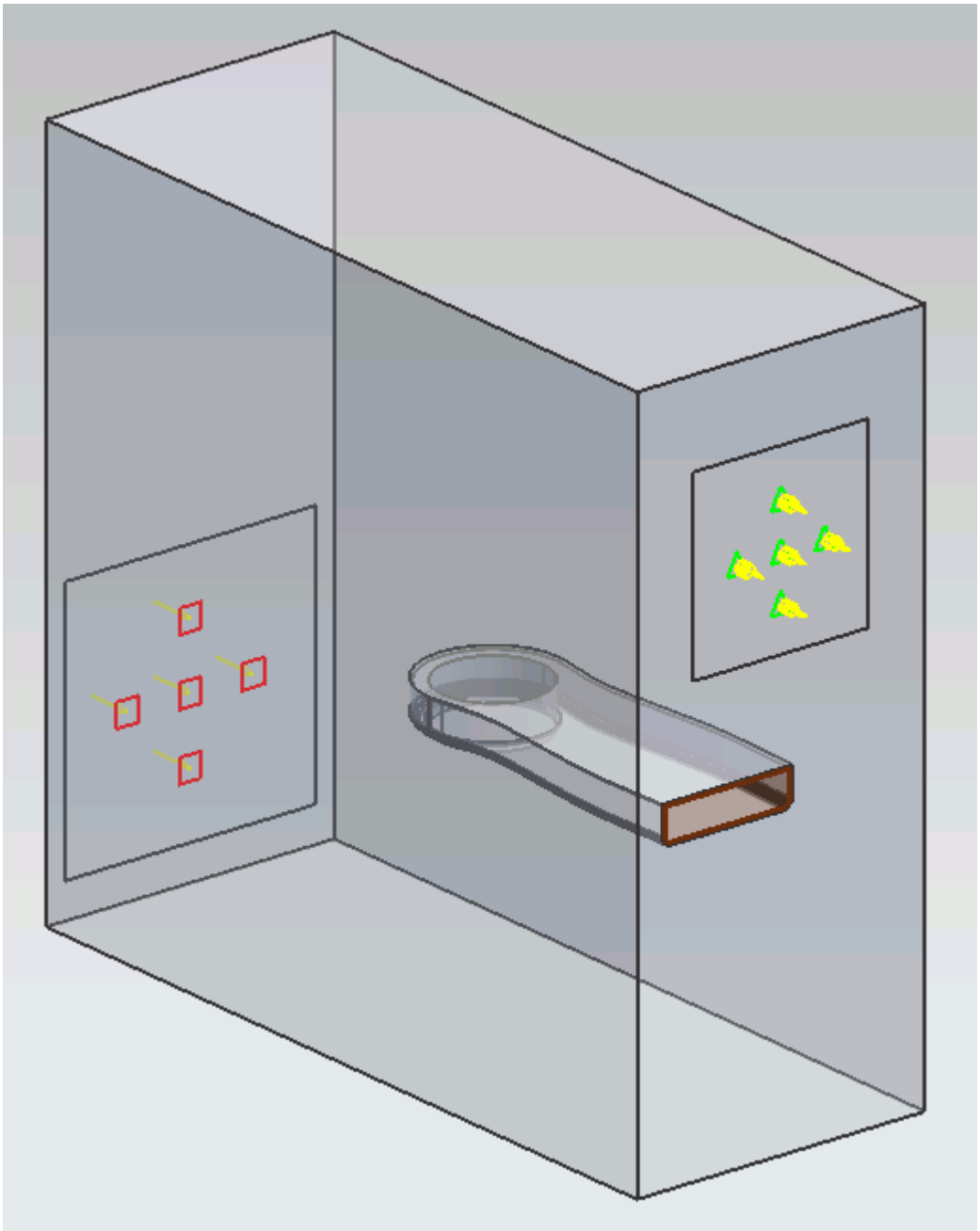
7. Set the following parameters:

Parameter	Value
Type	**Pressure Outlet**
Name	`Case Outlet`

8. Click **OK**.

Create the GPU outlet boundary:

9. Create another boundary and select the following face.



10. Set the following parameters:

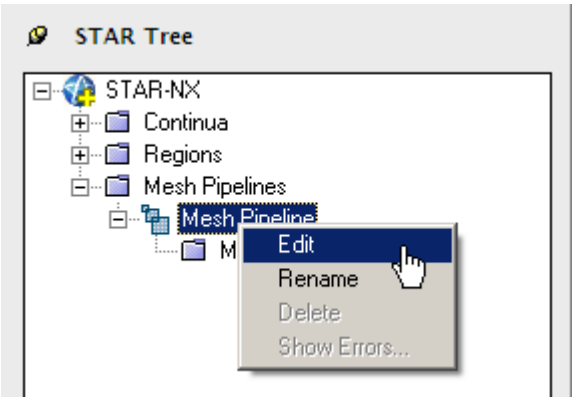
Parameter	Value
Type	**Pressure Outlet**
Name	`GPU Outlet`

11. Click **OK**.
12. Save your progress.

Setting the Mesh Pipeline

To obtain more accurate results, modify the mesh pipeline:

1. Right-click the **Mesh Pipelines** > **Mesh Pipeline** node in the STAR-NX tree and select **Edit**.



   The *Mesh Pipeline Settings* dialog appears.

2. Set the following parameters:

Parameter	Value
Base Size	
(**Base size option**)	**Absolute**
(*Base Size*)	`0.016 m`
Prism Layer	
(*Prism Layer Thickness*)	`20 %`

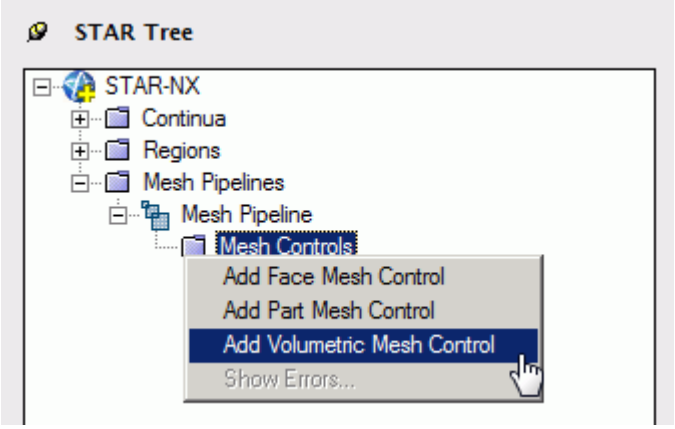
3. Click **OK**.

Creating the Volumetric Mesh Controls

Use volumetric mesh control regions to capture the finer details of the geometry model. Create three regions in total: a small region that closely surrounds the rotating region containing the fan, a region that encloses the GPU case, and a region that acts as an intermediary between the base size and the other volumetric control regions.

Create the volumetric mesh control for the rotating region:

1. Right-click the **Mesh Pipelines** > **Mesh Pipeline** > **Mesh Controls** node in the STAR-NX tree and select **Add Volumetric Mesh Control**.



The *Volumetric Control* dialog appears.

2. Set *Name* to `Fan Box`.
3. Activate the *Absolute* option.
4. Set (*Mesh Size*) to `0.002 m`.
5. Enter the following coordinates for Point 1:

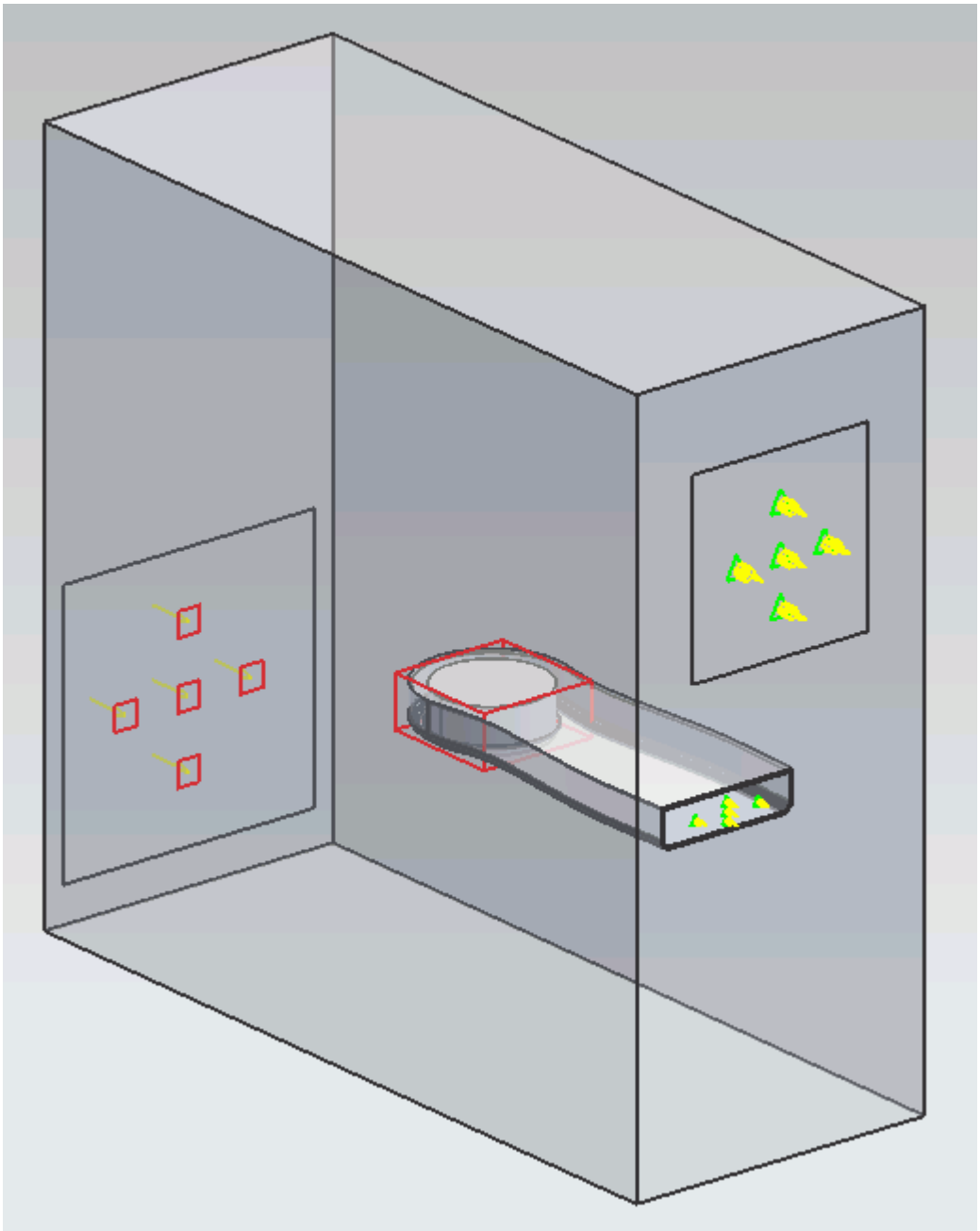
Coordinate	Value
X1	`-166.00`
Y1	`4.00`
Z1	`-61.00`

6. Enter the following coordinates for Point 2:

Coordinate	Value
X2	`-79.00`
Y2	`-34.00`
Z2	`25.00`

7. Click **Create**.

   The volumetric control region is shown below.



8. Click **OK**.

Create the volumetric mesh control for the GPU region:

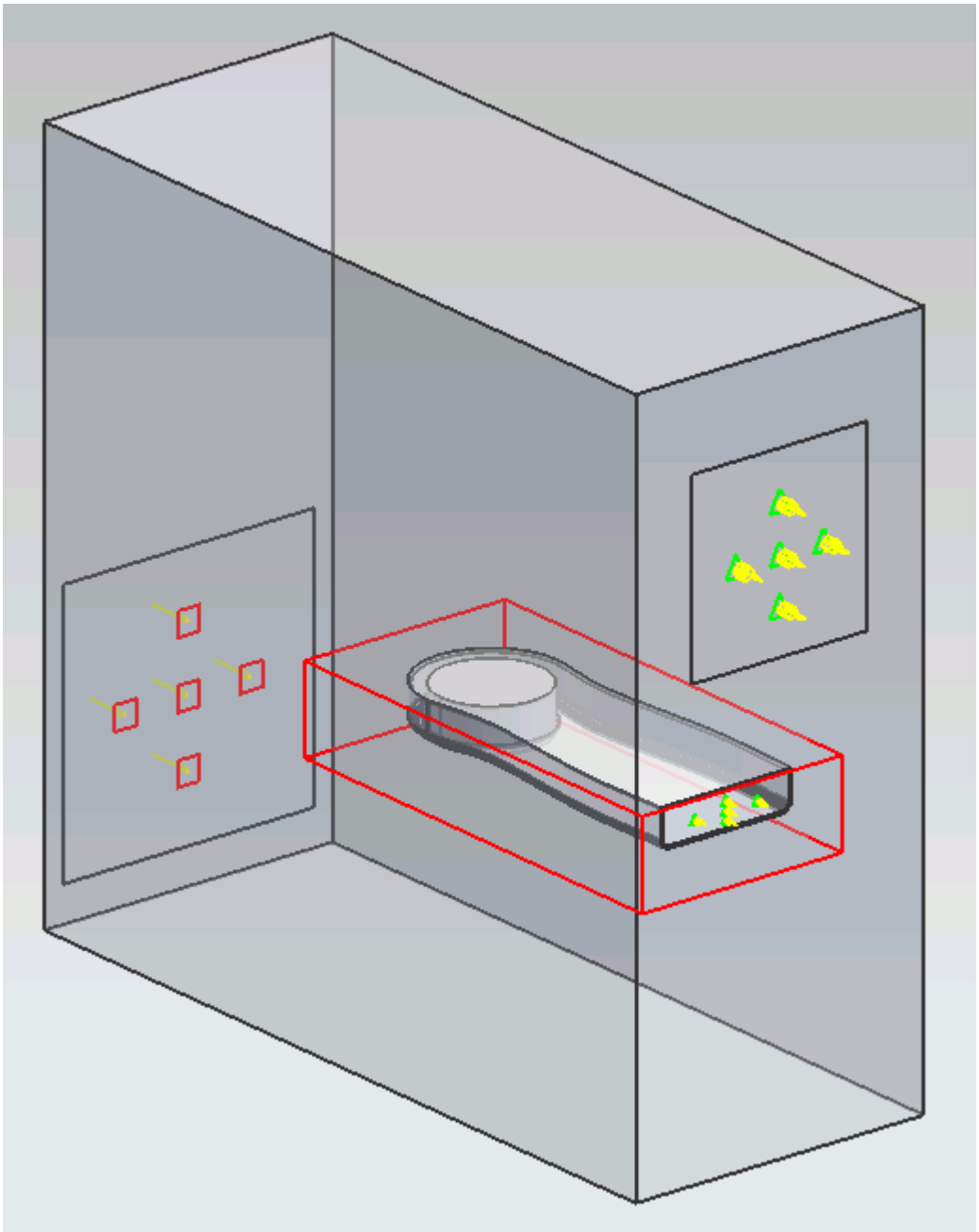
9. Using the same technique, create another volumetric mesh control with the following settings:

Parameter	Value
Name	`GPU Box`
Mesh Size	
(**Base size option**)	**Absolute**

Parameter	Value
(*Mesh Size*)	0.005 m
Shape	
X1	-383.00
Y1	16.00
Z1	-111.00
X2	-52.00
Y2	-49.00
Z2	49.00

10. Click **Create**.

    The volumetric control region is shown below.



11. Click **OK**.

Create the volumetric mesh control for the PC tower case:

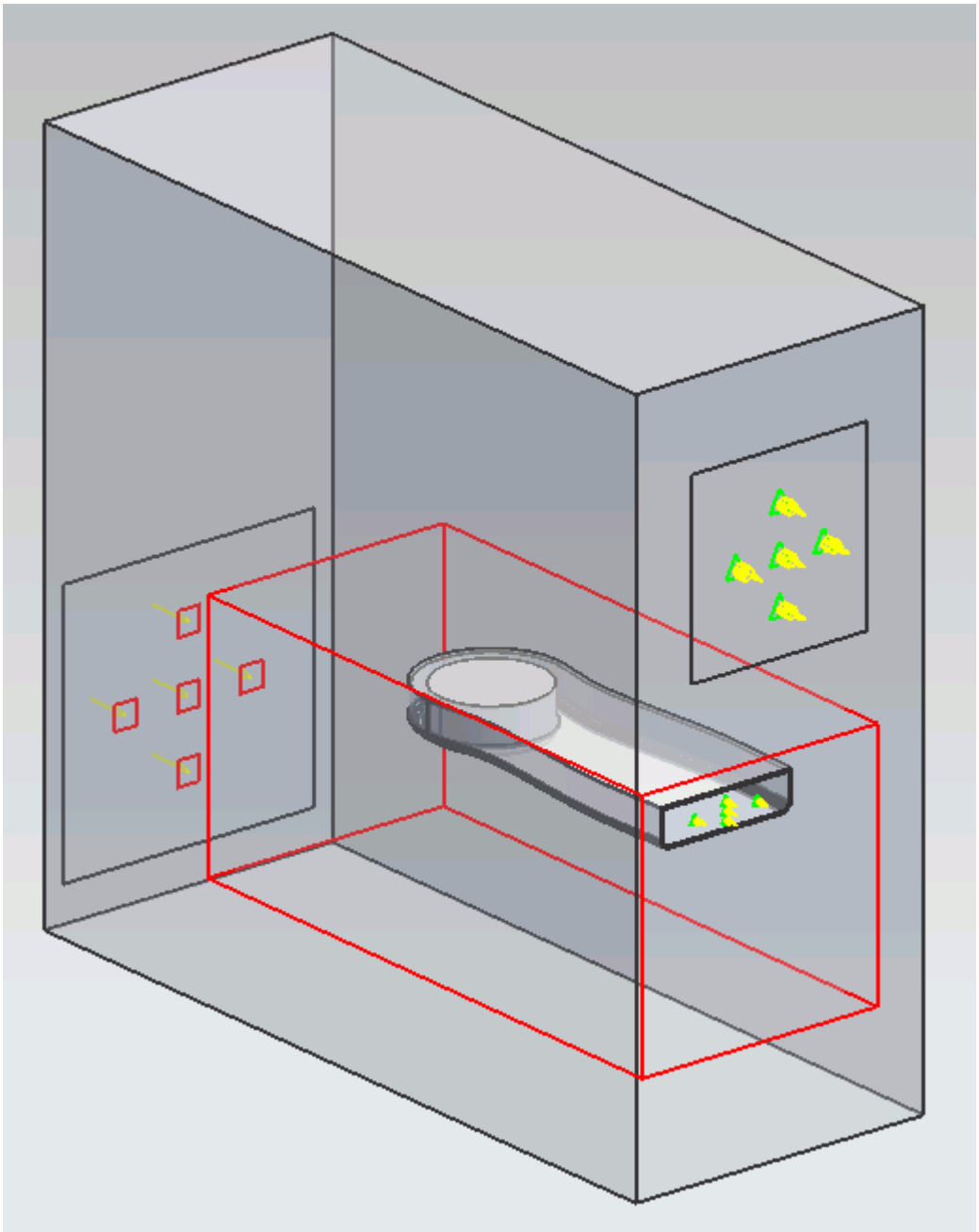
12. Using the same technique, create another volumetric mesh control with the following settings:

Parameter	Value
Name	`Case Box`
Mesh Size	
(**Base size option**)	**Absolute**

Parameter	Value
(*Mesh Size*)	0.009 m
Shape	
X1	-383.00
Y1	30.00
Z1	-111.00
X2	43.00
Y2	-159.00
Z2	77.00

13. Click **Create**.

    The volumetric control region is shown below.

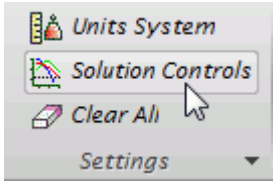


14. Click **OK**.

Modifying Solution Controls

Modify the maximum number of iterations and the solution accuracy:

1. Click **Solution Controls** on the STAR-NX ribbon.



The *Solution Controls* dialog appears.

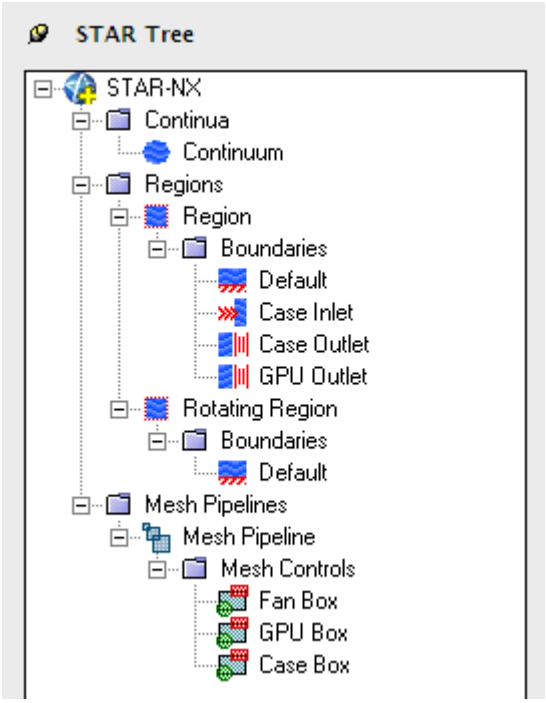
2. Set the following parameters:

Parameter	Value
(*Number of Iterations*)	300
(*Convergence Level*)	0.01
(*Momentum Under-Relaxation Factor*)	0.85

3. Click **OK**.
4. Save your progress.

Reviewing the Simulation Set Up

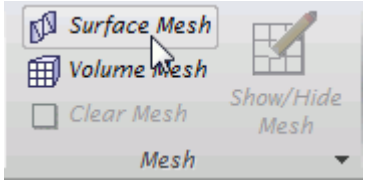
The final STAR-NX object tree is shown below.



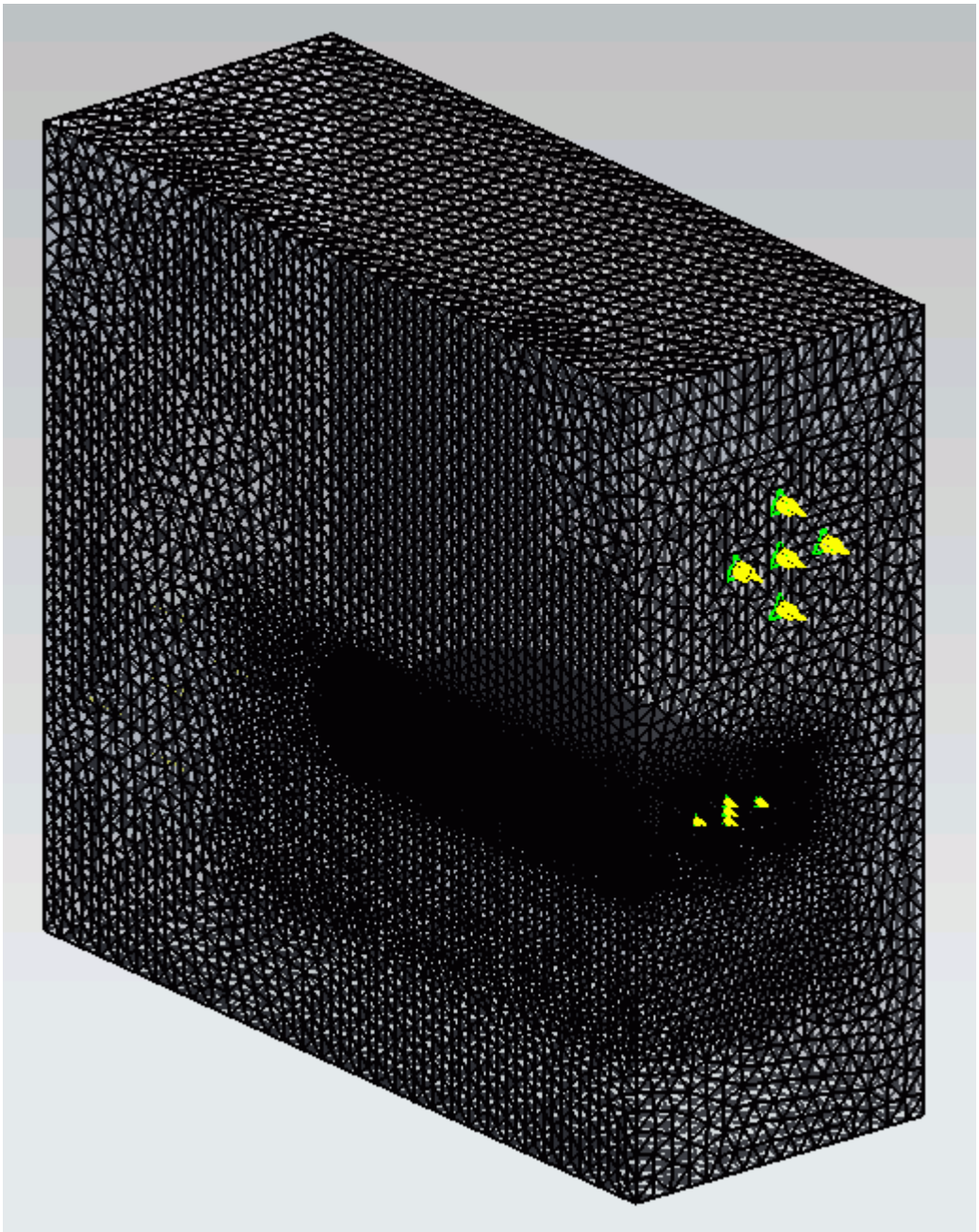
Generating The Surface Mesh

Generate the surface mesh:

1. From the STAR-NX ribbon, click **Surface Mesh**.



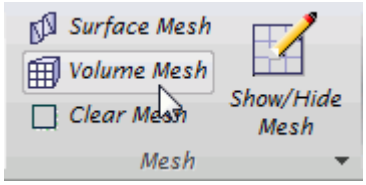
The surface mesh is shown below.



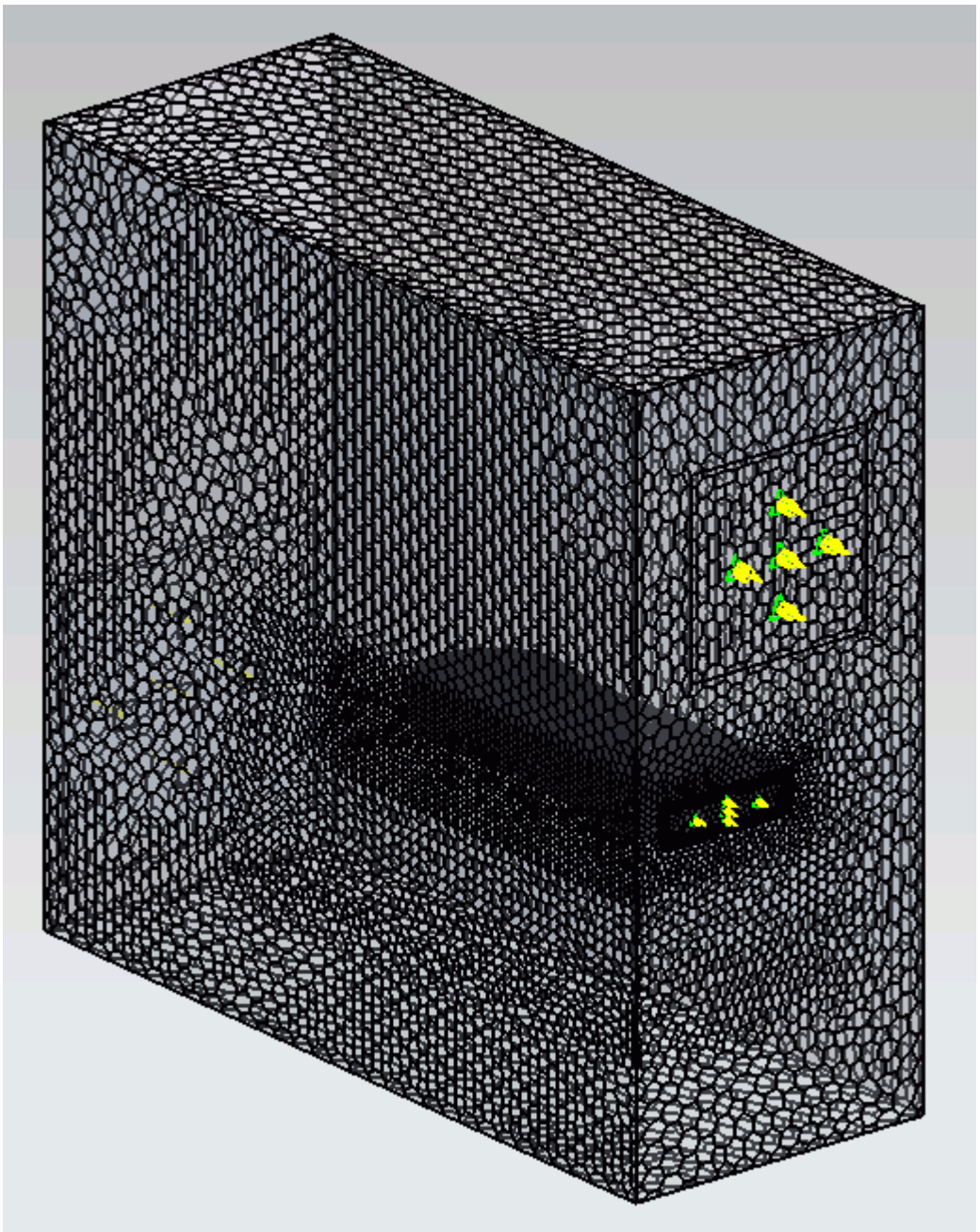
Generating the Volume Mesh

Generate the volume mesh:

1. From the STAR-NX ribbon, click **Volume Mesh**.



The volume mesh is shown below.

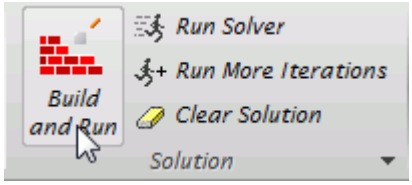


Building and Running

The case is now ready to run.

To run the simulation:

1. From the ribbon, click **Build and Run**.



The *Residuals* plot appears in the STAR-NX *Output* window. You can use this plot to monitor the convergence of the solution.

When the solver is done, STAR-CCM+ launches.

Post-Processing Results

You perform all post-processing in STAR-CCM+. By default, a scalar scene, *Scalar Scene 1*, automatically appears in the *Graphics* window when the solver finishes. This scene displays the absolute total pressure.

The results that are contained in the following sections are representative. You may see some variation in the results that you obtain, depending on what platform and package you are using.

1. In STAR-CCM+, click **File** > **Save** to save the simulation.

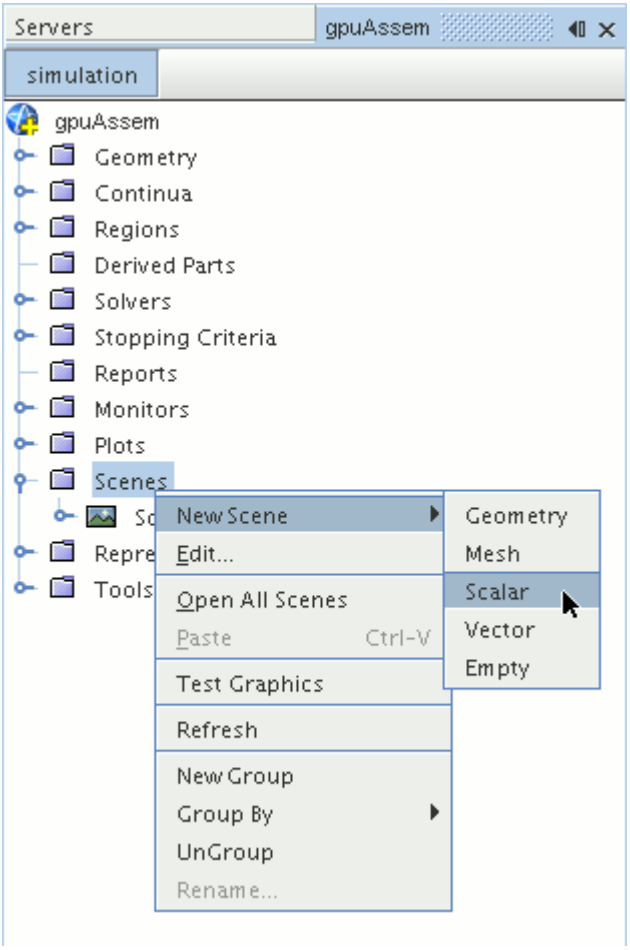
Contents:

Creating a Scalar Scene
Creating A Streamlines Scene
Creating a Vector Scene
Creating a Streamlines Scene
Creating A Scalar Scene
Visualizing the Wing Skin Temperature
Creating a Streamlines Scene
Creating A Scalar Scene
Creating a Streamlines Scene
Creating a Scalar Scene
Modifying The Scalar Scene
Creating a Streamlines Scene

Creating A Scalar Scene

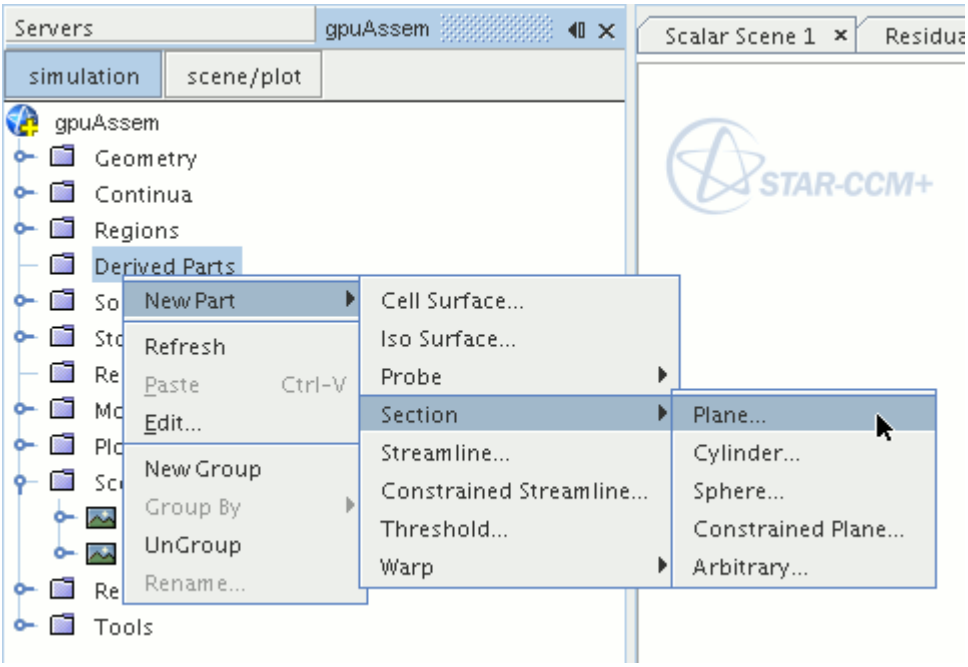
Create a scalar scene containing a cross-section of the GPU and plot velocity magnitude on this plane.

1. Right-click the **Scenes** node in the explorer pane and select **New Scene** > **Scalar**.



A new scalar scene, *Scalar Scene 2*, is created and selected in the *Graphics* window.

2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane...**.

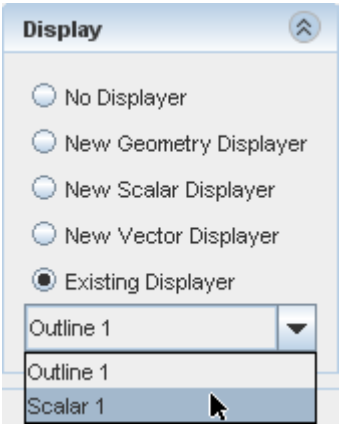


The *Create Plane Section* dialog appears.

3. In the *Plane Parameters* box, set the *origin* vector to `[0 m, -0.012 m, 0 m]` and the *normal* vector to `[0 m, 1 m, 0 m]`.

   Note that the units are in meters.

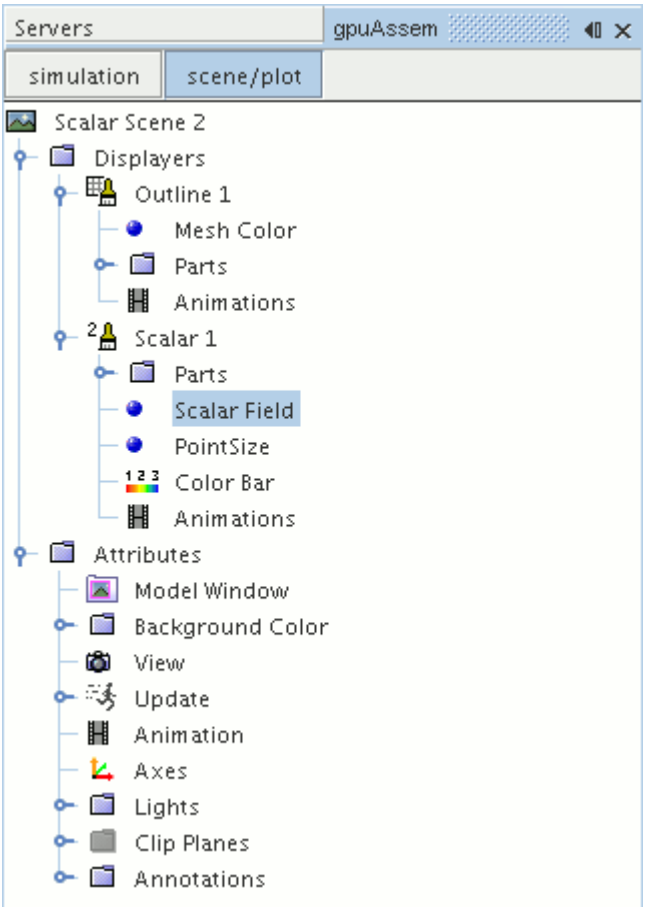
4. In the *Display* group box, select **Existing Displayer** and in the drop-down menu select **Scalar 1**.



5. Click **Create** and then **Close.**

Display velocity magnitude on the section plane:

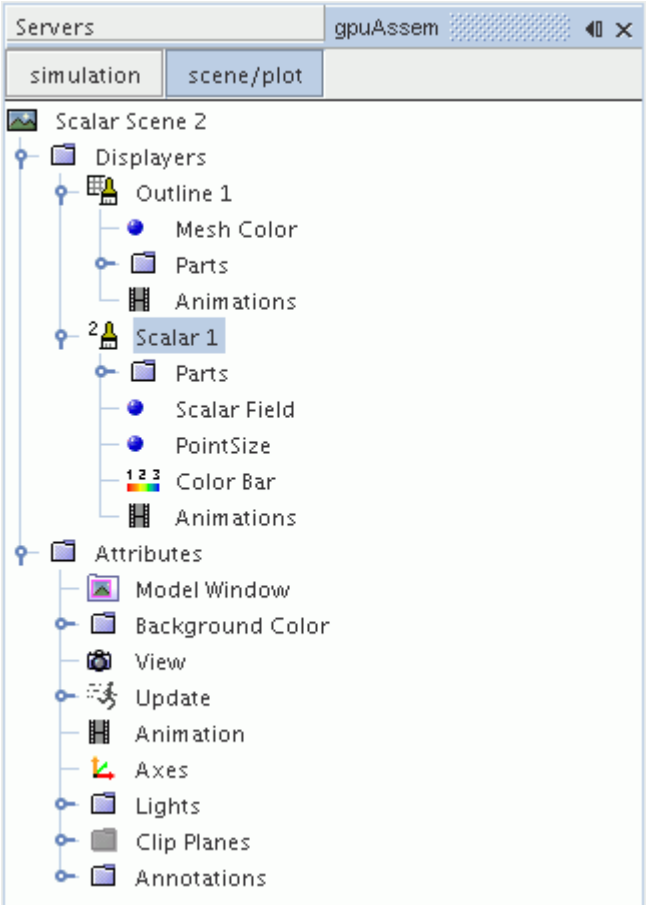
6. Click **scene/plot**.
7. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Scalar Field**.



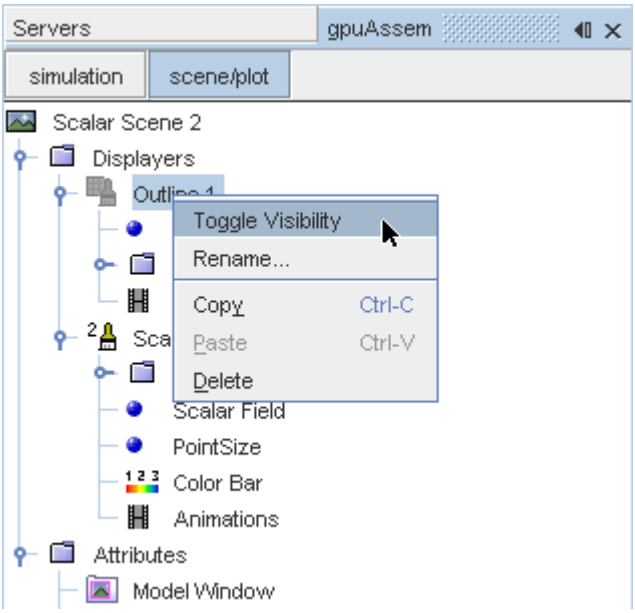
8. Set *Function* to **Velocity** > **Lab Reference Frame** > **Magnitude**.

To have a smooth transition between the different velocity values, set the contour style to smooth:

9. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1**.

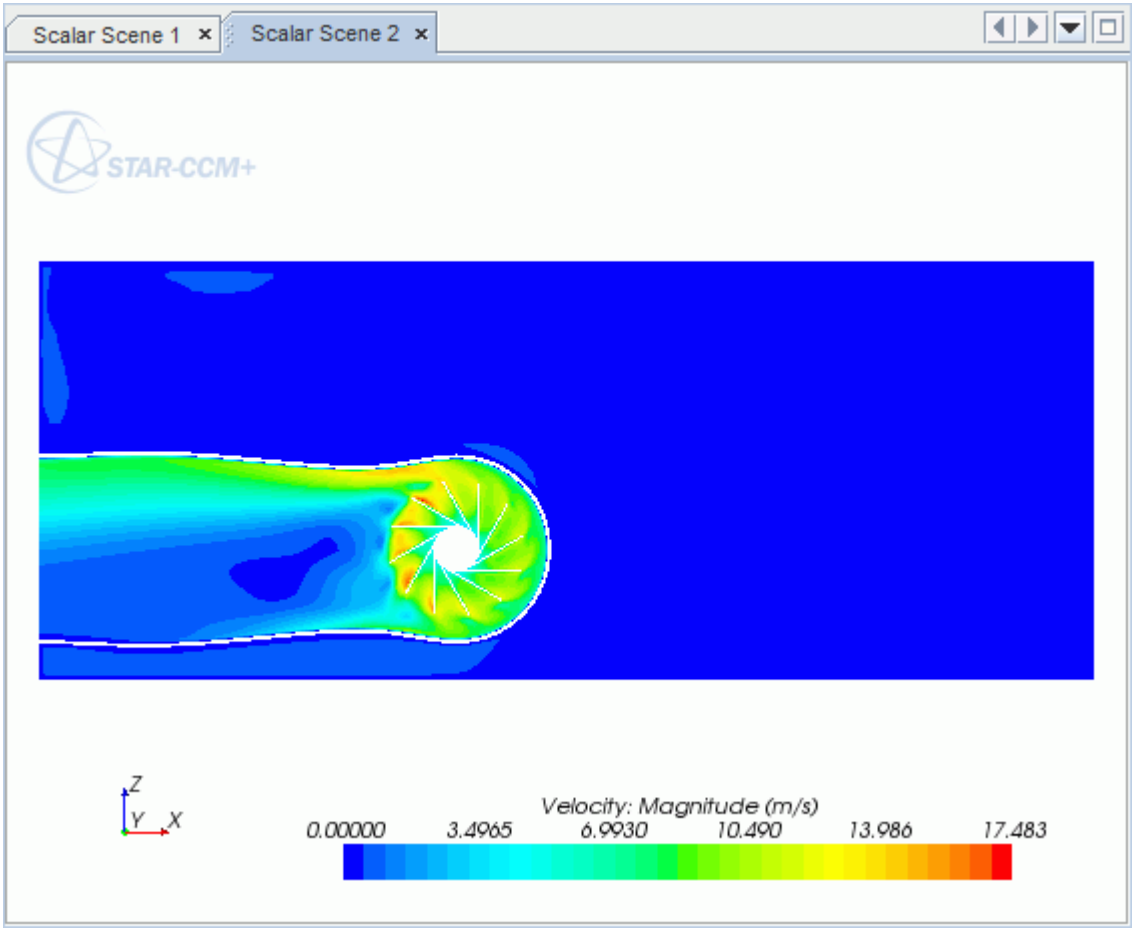


10. Set *Contour Style* to **Smooth Filled**.
11. To hide all outlines, right-click the **Scalar Scene 2** > **Displayers** > **Outline 1** node and select **Toggle Visibility**.



12. Click the *Scalar Scene 2* tab to activate the scene, and press the <S> key to change to the top view of this geometry.
13. Click **simulation**.
14. Click (**Save**).

The finished scene is shown below.

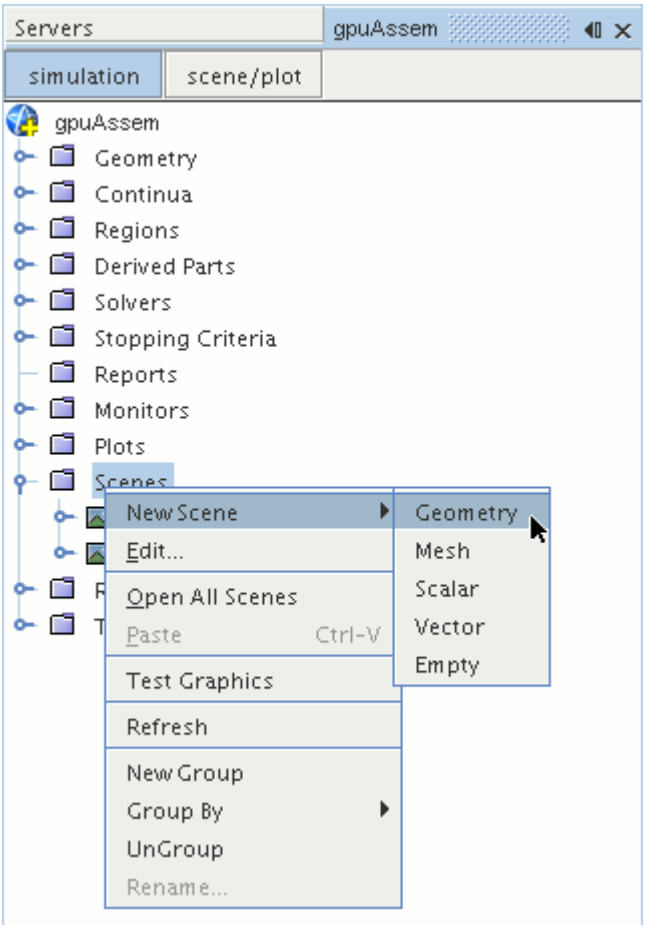


Creating a Streamlines Scene

Create streamlines to get a better understanding of the air flow inside the PC tower case and graphics card.

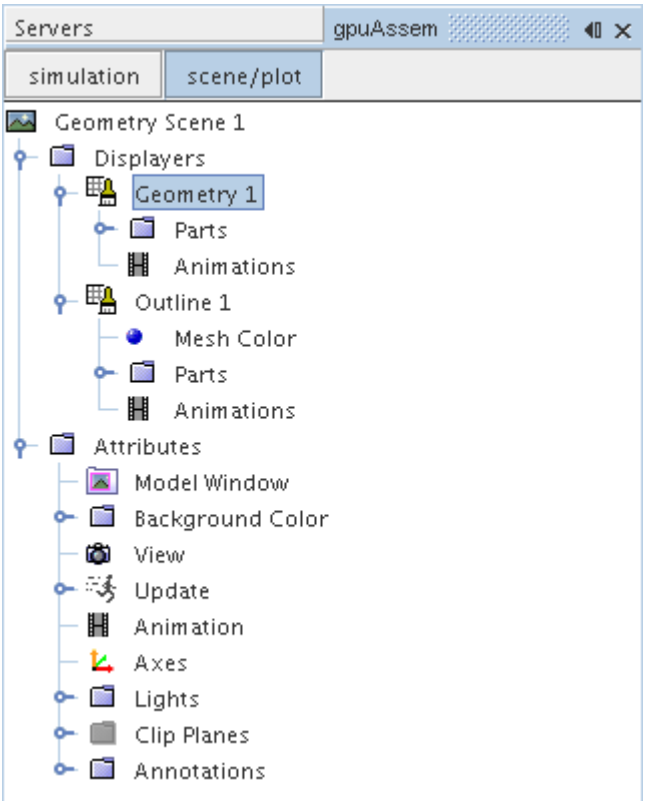
First, create a geometry scene:

1. Right-click the **Scenes** node in the object tree and select **New Scene** > **Geometry**.



A new geometry scene is created and selected in the *Graphics* window.

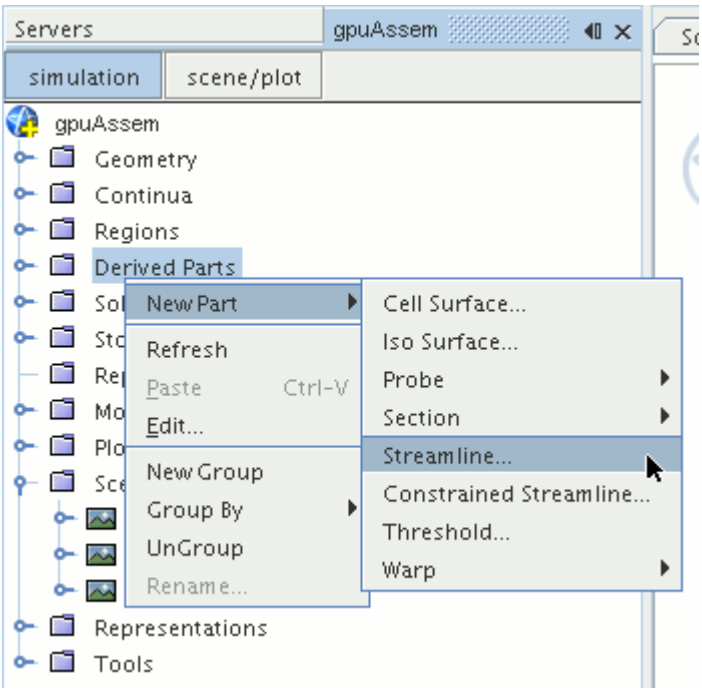
2. Click **scene/plot**.
3. Navigate to **Geometry Scene 1** > **Displayers** > **Geometry 1** node.



4. Set *Opacity* to `0.1`

Create the streamlines:

5. Click **simulation**.
6. Right-click the **Derived Parts** node and select **New Part** > **Streamline...**.



The *Create Streamline* dialog appears.

7. Make sure that **Part Seed** is selected in the *Seed Mode* section.
8. Set *Seed Parts* to **Regions** > **Region** > **Boundaries** > **Case Inlet**.
9. Set *Part U-Resolution* to `10`.
10. Set *Part V-Resolution* to `10`.
11. Make sure that **New Streamline Displayer** is selected in the *Display* section.
12. Click **Create** followed by **Close**.

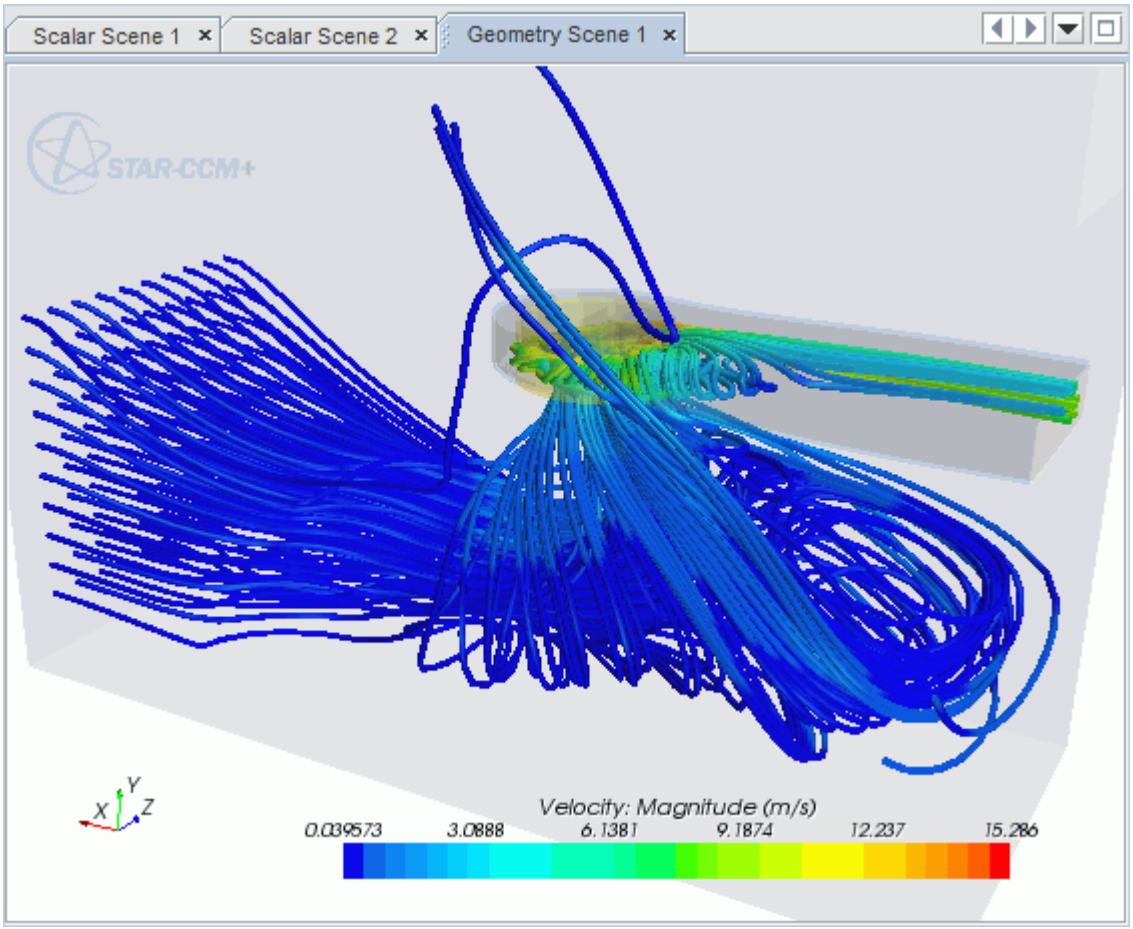
Adjust the appearance of streamlines to display tubes instead of the default lines:

13. Click **scene/plot**.
14. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** and set *Mode* to **Tubes** and *Width* to `0.0020`.
15. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** > **Scalar Field** and set *Function* to **Velocity** > **Lab Reference Frame** > **Magnitude**.

Hide all the outlines in the scene:

16. Right-click the **Geometry Scene 1** > **Displayers** > **Outline 1** node and select **Toggle Visibility**.
17. Reposition the geometry to get a better view of the flow inside the case.
18. Save the simulation.

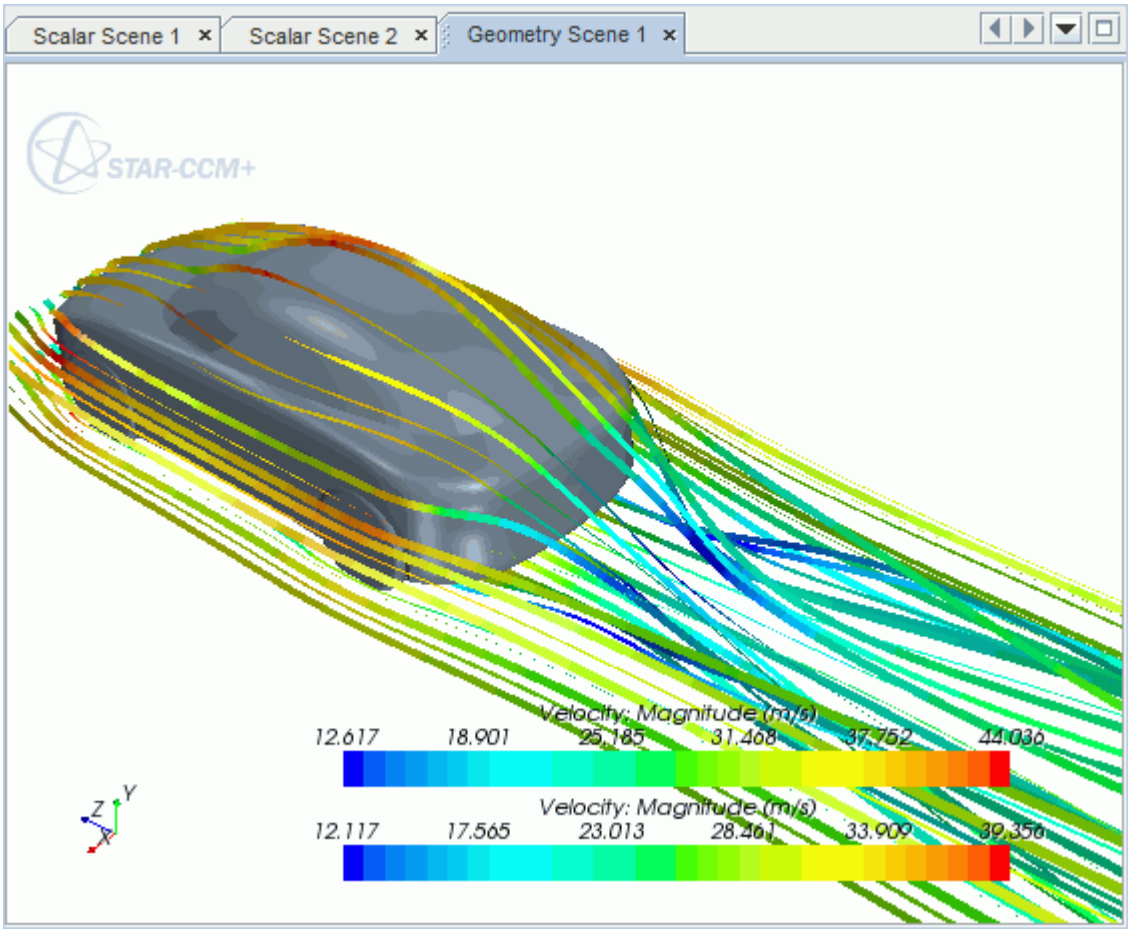
The finished scene is shown below.


Scalar Scene 1
Scalar Scene 2
Geometry Scene 1
STAR-CCM+
Velocity: Magnitude (m/s)
0.039573
3.0888
6.1381
9.1874
12.237
15.286
X
Y
Z

External Flow Over a Car

This tutorial takes you through the steps that are required to run a CFD calculation in the STAR-NX environment.

For this tutorial, the model is a car in a fluid domain.



Contents:

Opening the CAD Model
Selecting Fluid Physics Models
Defining Boundary Conditions
Setting the Mesh Pipeline
Creating the Volumetric Mesh Controls
Creating the Face Mesh Control
Modifying Solution Controls
Reviewing the Simulation Set Up
Generating the Surface Mesh

Generating the Volume Mesh
Building and Running
Post-Processing Results

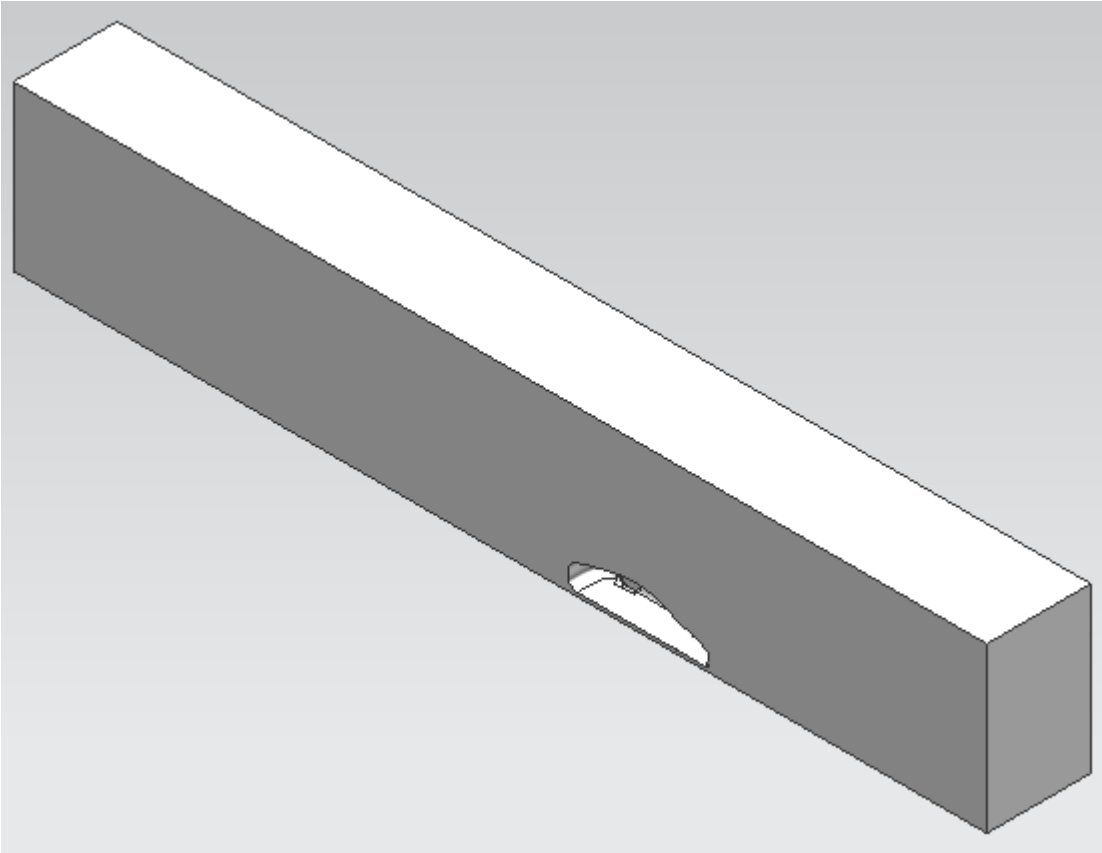
Opening the CAD Model

A directory containing the CAD geometry is provided with the STAR-CCM+ distribution.

This directory is located in `[STAR-CCM+_INSTALL_DIR]\doc\startutorialsdata\starnx\data`.

1. Copy directory `data` to your own disk space and open the part file `car.prt` in STAR-NX.

   The imported geometry is shown below.



The STAR-NX ribbon appears at the top of the screen.The STAR-NX object tree is located on the left of the screen.

To switch to this object tree:

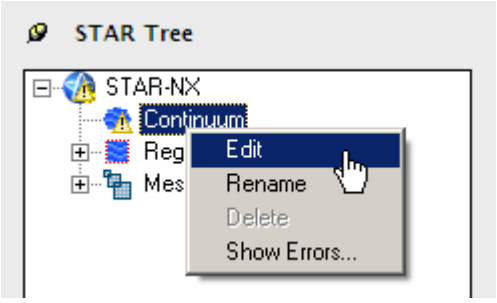
2. Click (**STAR Tree**).

If these items are not visible, refer to Launching STAR-NX Manually.

Selecting Fluid Physics Models

The first step in setting up this tutorial is to define the physics of the problem. For this case, the flow is turbulent and isothermal.

1. Right-click the **Continuum** node in the STAR-NX tree and select **Edit**.



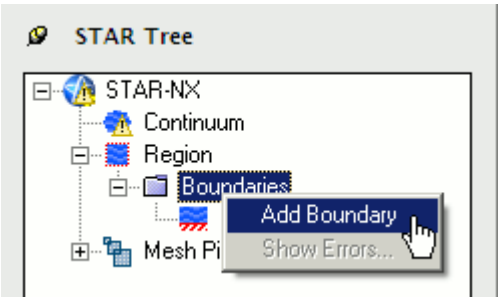
   The *Continuum Settings* dialog appears.
2. Deactivate the **Temperature** option in the *Optional Physics Models*.
3. Click **OK**.

Defining Boundary Conditions

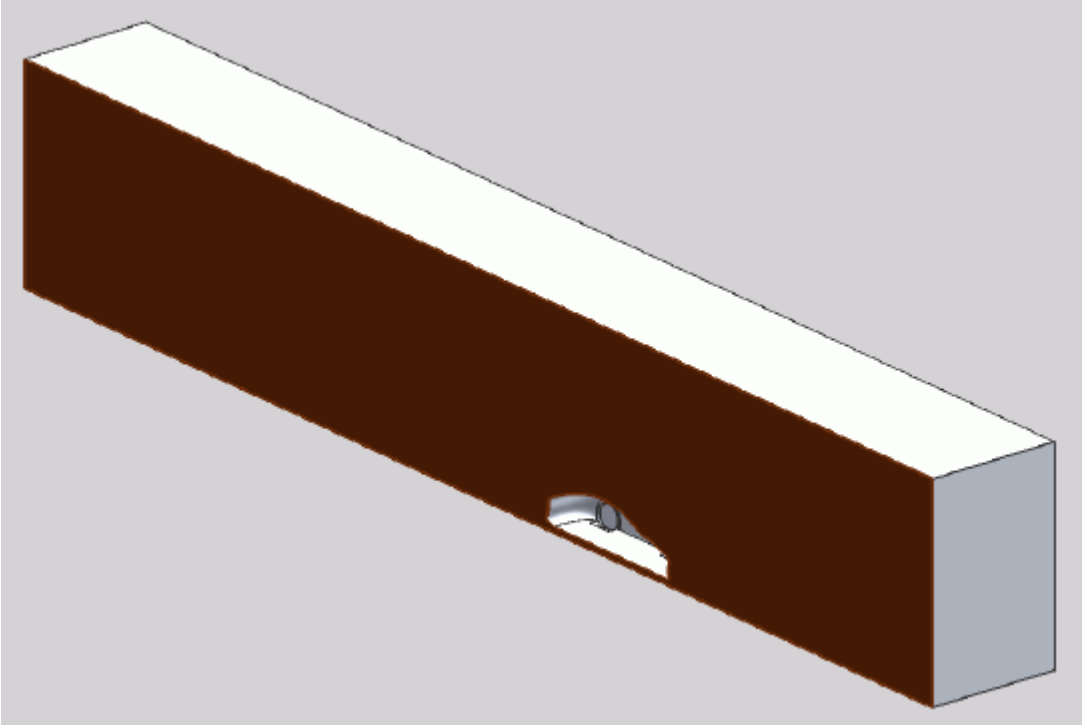
Create the inlet and outlet boundary condition. Use a moving wall to model the relative movement of the road.

Create the symmetry plane first:

1. Right-click the **Region** > **Boundaries** node and select **Add Boundary**.



   The *Boundary Settings* dialog appears.
2. On the CAD model, select the face that is highlighted below.



These selections are reported in the *Face Selection* box in the *Boundary Settings* dialog.

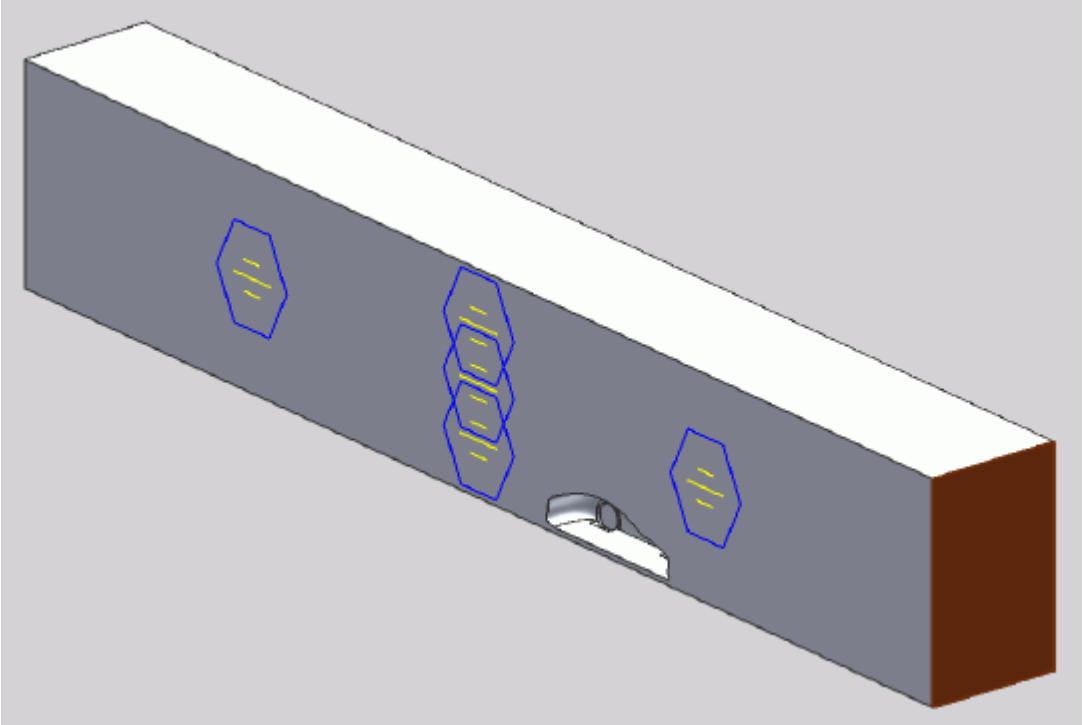
3. Set the following parameters:

Parameter	Value
Type	**Symmetry Plane**
Name	`Symmetry Plane`

4. Click **OK**.

Create the inlet boundary:

5. Create another boundary and select the following face.



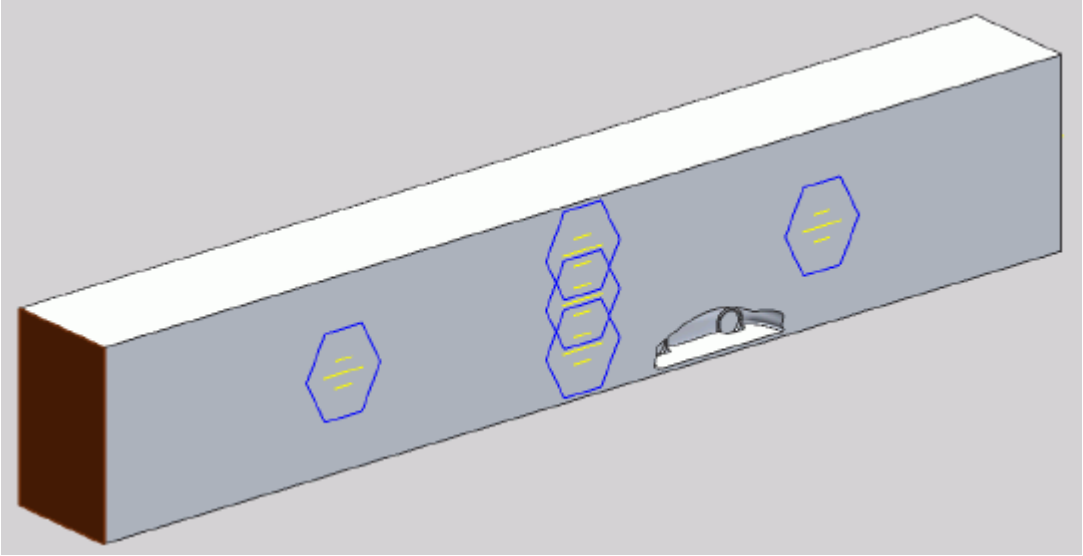
6. Set the following parameters:

Parameter	Value
Type	**Velocity Inlet**
Name	`Inlet`
(*Normal Velocity*)	`70 mph` **Note:** The units are in **mph**.

7. Click **OK**.

Create the outlet boundary:

8. Create another boundary.
9. Rotate the view to display the back face and select the face that is highlighted below.



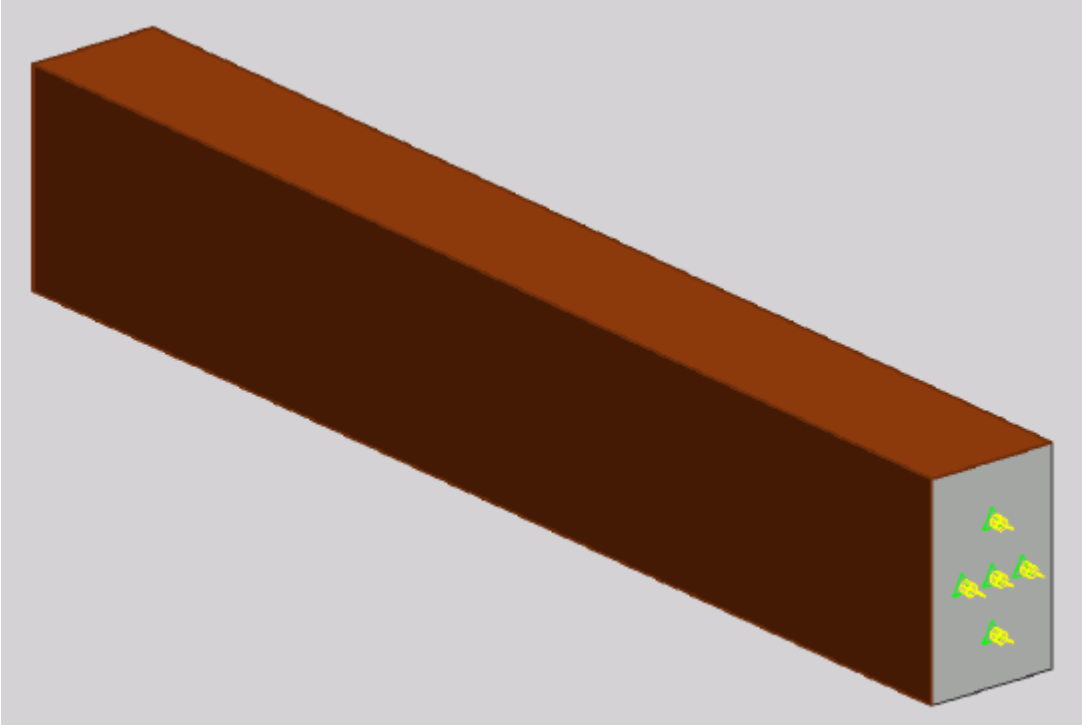
10. Set the following parameters:

Parameter	Value
Type	**Pressure Outlet**
Name	`Outlet`

11. Click **OK**.

Create the fluid domain symmetry boundaries:

12. Create another boundary.
13. Rotate the view to display the top and side walls, and select the faces that are highlighted below.



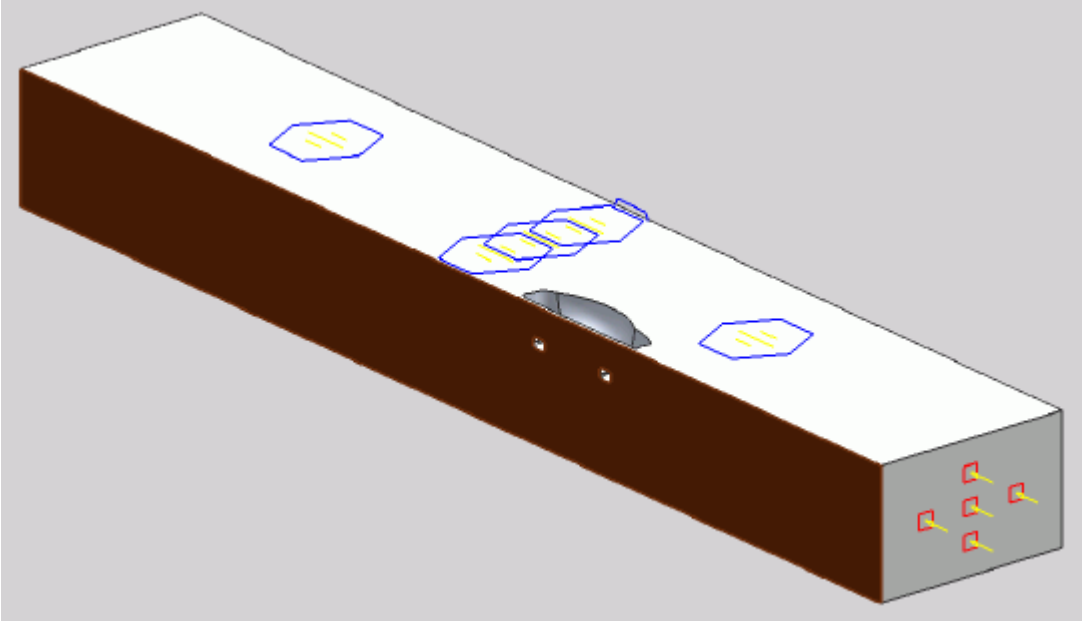
14. Set the following parameters:

Parameter	Value
Type	**Symmetry Plane**
Name	`Boundary Symmetry`

15. Click **OK**.

Create the boundary that will model the moving ground:

16. Create another boundary.
17. Rotate the view to display the bottom face, and select the face that is highlighted below.



18. Set the following parameters:

Parameter	Value
Type	**Wall**
Name	`Ground`
Wall Type	**Translating**
(*W Component of Wall Velocity Vector*)	`-70 mph` **Note:** The units are in **mph**.

19. Click **OK**.

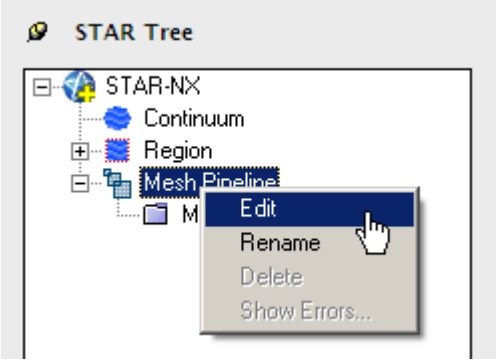
Rename the default boundary:

20. Right-click **Region** > **Boundaries** > **Default** and select **Rename**.
21. Enter `Car` as the new name.
22. You may need to click on a blank area of the workspace to accept the name change.
23. Save your progress.

Setting the Mesh Pipeline

To obtain more accurate results, modify the mesh pipeline:

1. Right-click the **Mesh Pipeline** node in the STAR-NX tree and select **Edit**.



The *Mesh Pipeline Settings* dialog appears.

2. Set the following parameters:

Parameter	Value
Base Size	
(**Base size option**)	**Absolute**
(*Base Size*)	`0.03 m`
Prism Layer	
(**Layer thickness option**)	**Absolute**
(*Prism Layer Thickness*)	`0.03 m`

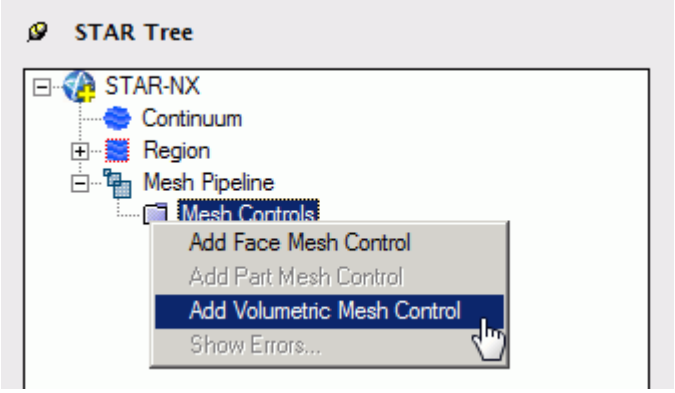
3. Click **OK**.

Creating the Volumetric Mesh Controls

Create two volumetric mesh control regions to capture the finer details of the flow: one around the car and one that extends into the wake of the car.

Create the volumetric mesh control for the car region:

1. Right-click the **Mesh Pipeline** > **Mesh Controls** node in the STAR-NX tree and select **Add Volumetric Mesh Control**.



The *Volumetric Control* dialog appears.

2. Set *Name* to `Car Box`.
3. Activate the *Absolute* option.
4. Set (*Mesh Size*) to `0.09 m`.
5. Enter the following coordinates for Point 1:

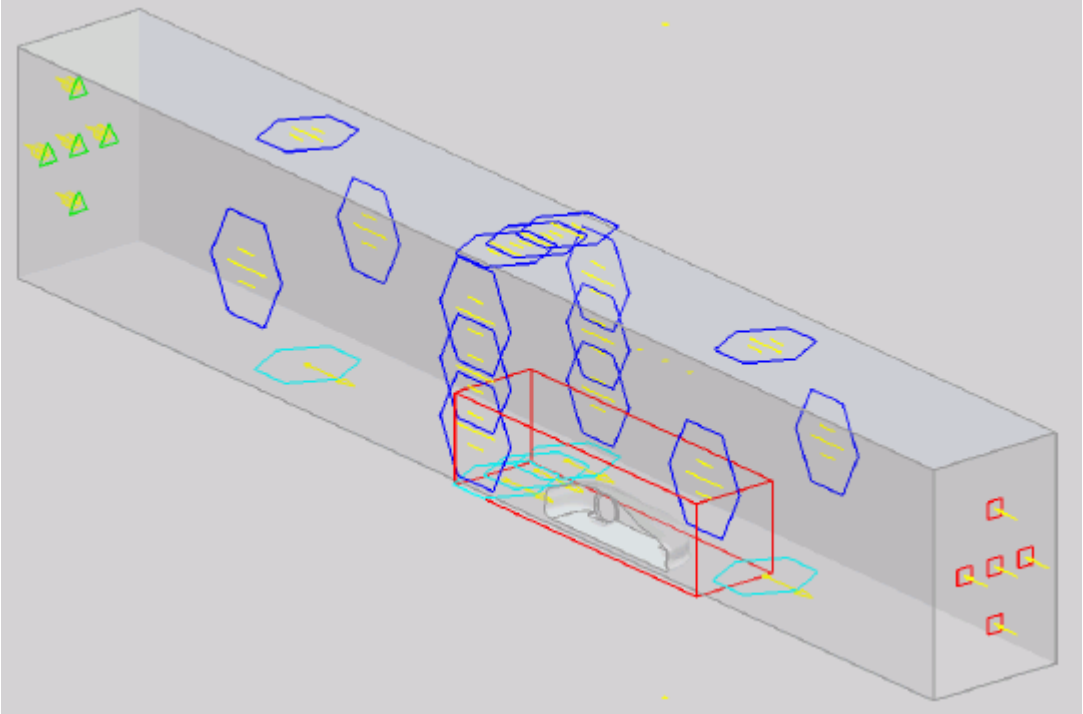
Coordinate	Value
X1	`-50.00`
Y1	`-650.00`
Z1	`3432.00`

6. Enter the following coordinates for Point 2:

Coordinate	Value
X2	`2000.00`
Y2	`1440.00`
Z2	`-4567.00`

7. Click **Create**.

   The volumetric control region is shown below.



8. Click **OK**.

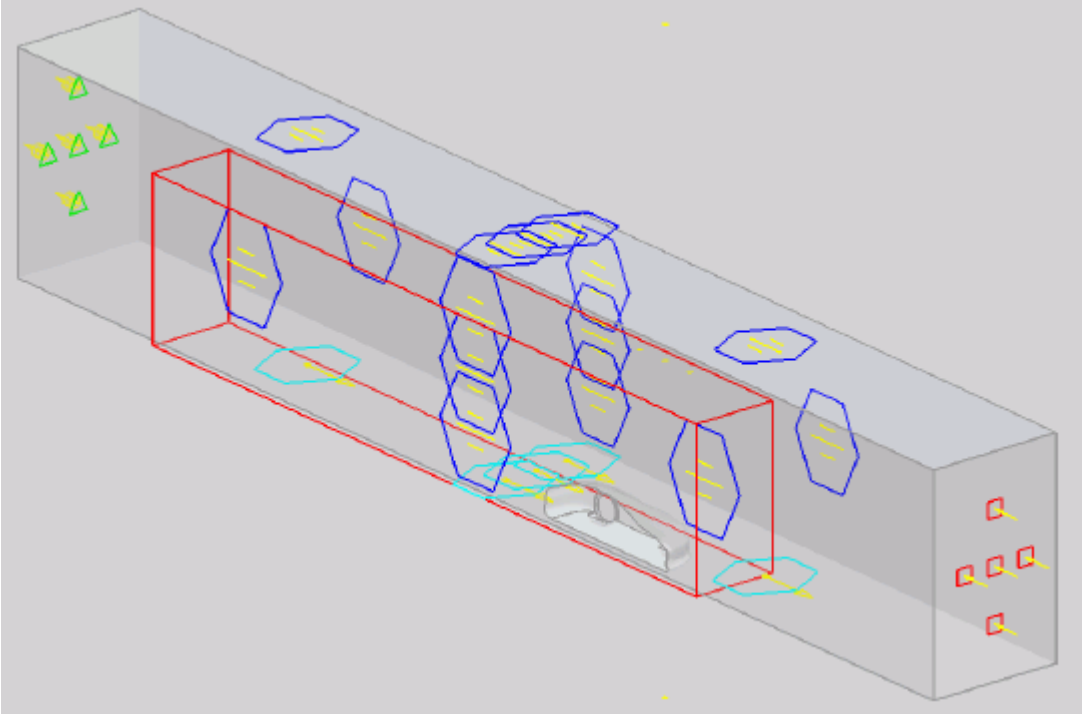
Create the volumetric mesh control for the wake region:

9. Using the same technique, create another volumetric mesh control with the following settings:

Parameter	Value
Name	`Wake Box`
Mesh Size	
(**Base size option**)	**Absolute**
(*Mesh Size*)	`0.177 m`
Shape	
X1	`-50.00`
Y1	`-650.00`
Z1	`3432.00`
X2	`2000.00`
Y2	`3240.00`
Z2	`-14567.00`

10. Click **Create**.

    The volumetric control region is shown below.



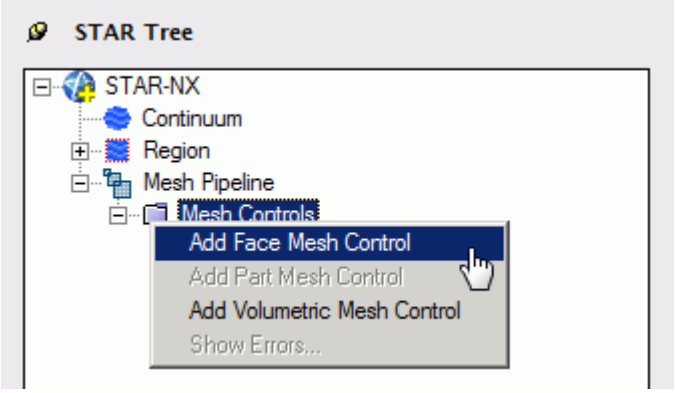
11. Click **OK**.

Creating the Face Mesh Control

Apply a face mesh control to the walls of the fluid domain to coarsen the mesh on these faces.

Create the face mesh control:

1. Right-click the **Mesh Pipeline** > **Mesh Controls** node in the STAR-NX tree and select **Add Face Mesh Control**.



   The *Face Mesh Control Settings* dialog appears.

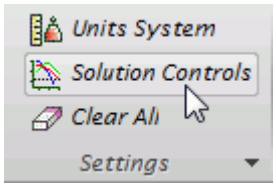
2. Select all six boundary faces that form the fluid domain: `Inlet`, `Outlet`, `Boundary Symmetry`, `Symmetry Plane`, and `Ground`.
3. Set *Name* to `Fluid Domain Face Control`.

4. Activate the **Enable** option in the *Custom Mesh Size* box.
5. Activate the *Absolute* option.
6. Set (*Mesh Size*) to `0.35 m`.
7. Click **OK**.

Modifying Solution Controls

Modify the solution controls:

1. Click **Solution Controls** on the STAR-NX ribbon.



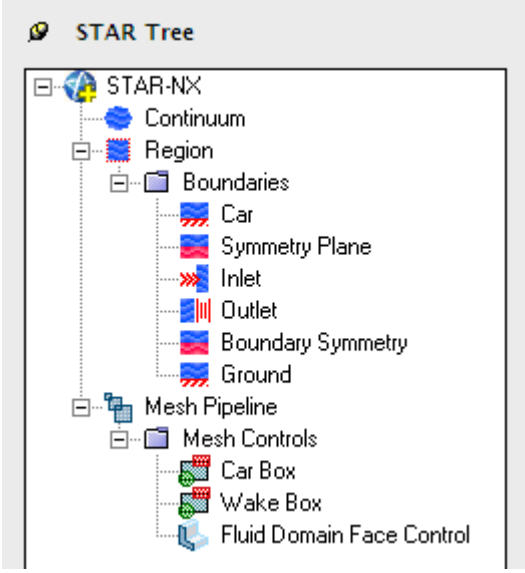
   The *Solution Controls* dialog appears.
2. Expand the *Advanced* group box.
3. Set the following parameters:

Parameter	Value
(*Number of Iterations*)	`400`
(*Convergence Level*)	`0.0001`
(*Momentum Under-Relaxation Factor*)	`0.85`

4. Click **OK**.
5. Save your progress.

Reviewing the Simulation Set Up

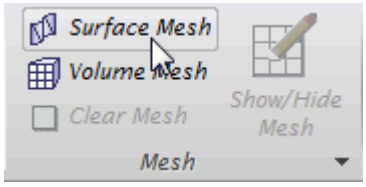
The final STAR-Works object tree is shown below.



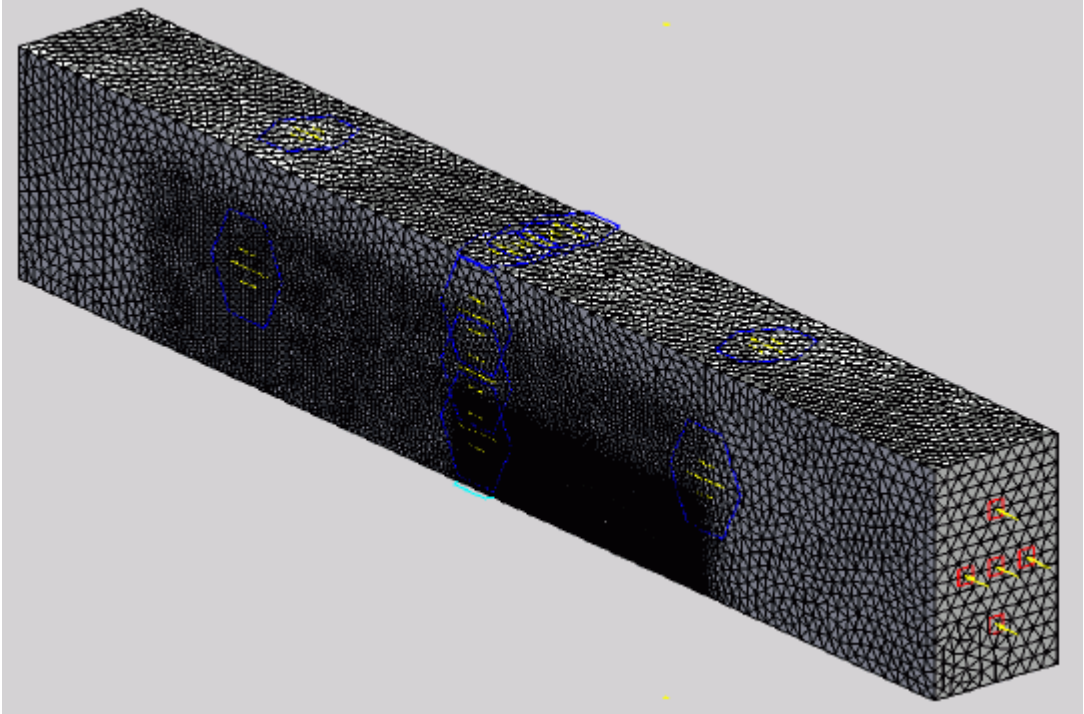
Generating the Surface Mesh

Generate the surface mesh:

1. From the STAR-NX ribbon, click **Surface Mesh**.



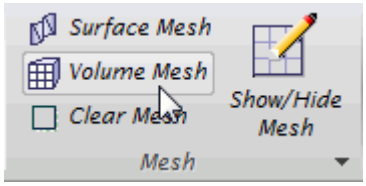
The surface mesh is shown below.



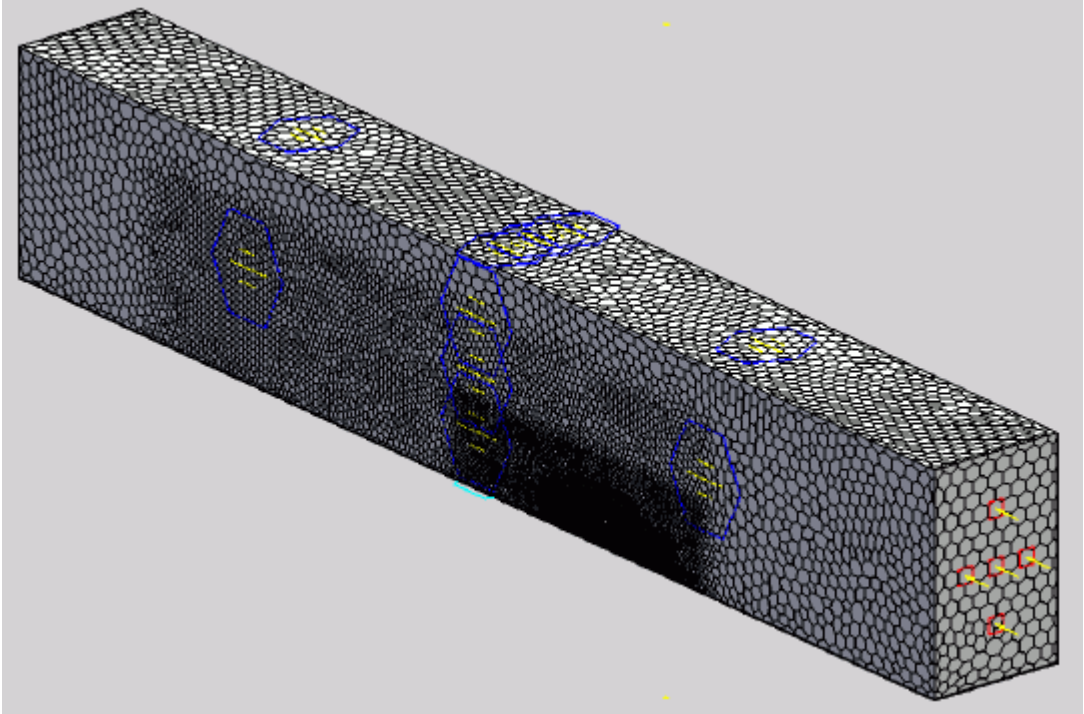
Generating the Volume Mesh

Generate the volume mesh:

1. From the STAR-NX ribbon, click **Volume Mesh**.



The volume mesh is shown below.

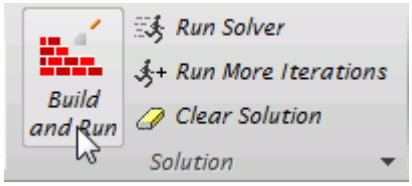


Building and Running

The case is now ready to run.

To run the simulation:

1. From the ribbon, click **Build and Run**.



The *Residuals* plot appears in the STAR-NX *Output* window. You can use this plot to monitor the convergence of the solution.

When the solver is done, STAR-CCM+ launches.

Post-Processing Results

You perform all post-processing in STAR-CCM+. By default, a scalar scene, *Scalar Scene 1*, automatically appears in the *Graphics* window when the solver finishes. This scene displays the absolute total pressure.

The results that are contained in the following sections are representative. You may see some variation in the results that you obtain, depending on what platform and package you are using.

1. In STAR-CCM+, click **File** > **Save** to save the simulation.

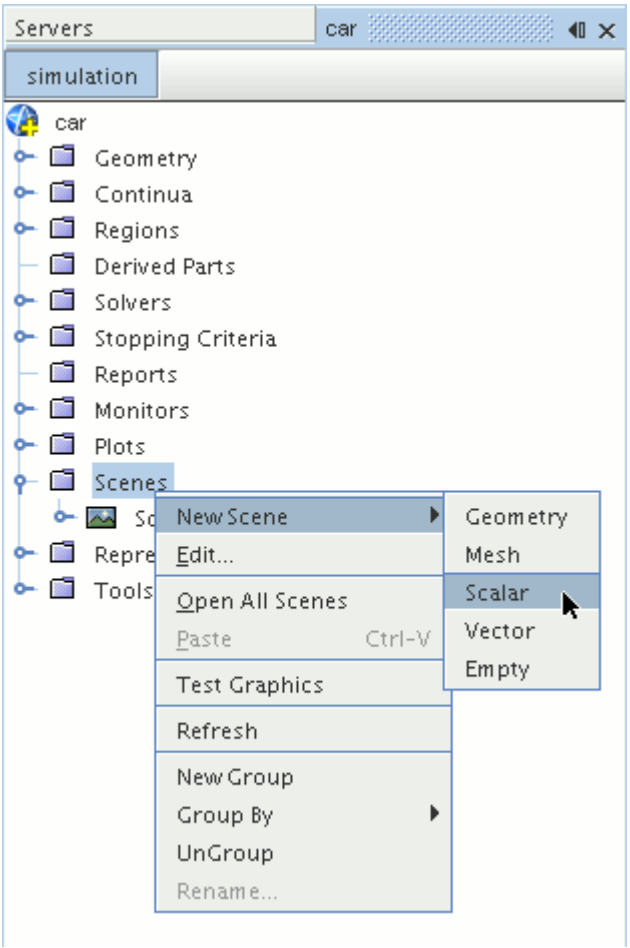
Contents:

Creating a Scalar Scene
Creating A Streamlines Scene
Creating a Vector Scene
Creating a Streamlines Scene
Creating A Scalar Scene
Visualizing the Wing Skin Temperature
Creating a Streamlines Scene
Creating A Scalar Scene
Creating a Streamlines Scene
Creating a Scalar Scene
Modifying The Scalar Scene
Creating a Streamlines Scene

Creating a Scalar Scene

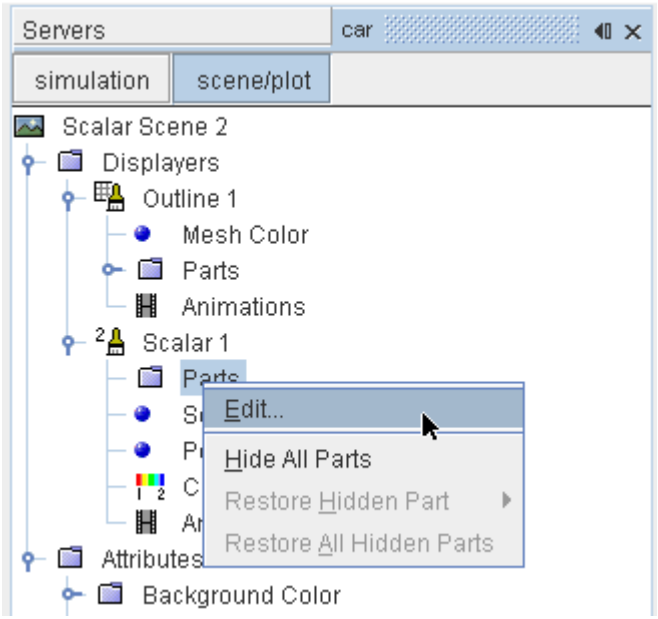
Create a scalar scene to display static pressure.

1. Right-click the **Scenes** node in the explorer pane and select **New Scene** > **Scalar**.



A new scalar scene, *Scalar Scene 2*, is created and selected in the *Graphics* window.

2. Click **scene/plot**.
3. Right-click the **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Parts** node and select **Edit...**.

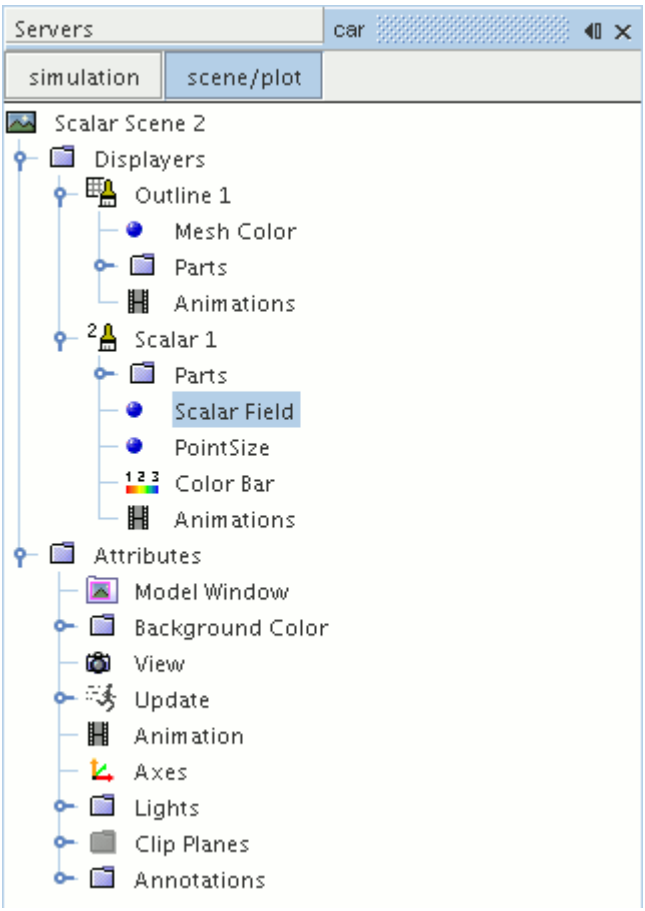


The *Parts* dialog appears.

4. Expand the **Regions** > **Region** > **Boundaries** node and select the following items:
   - **Car**
   - **Ground**
   - **Symmetry Plane**
5. Click **OK**.

Display the pressure contours in the scene:

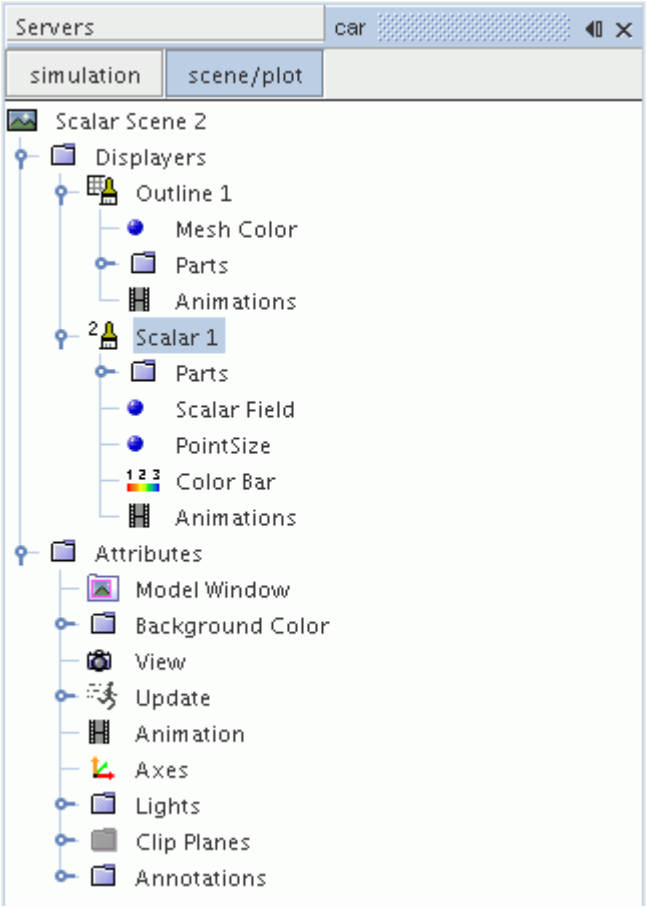
6. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Scalar Field**.



7. Set *Function* to **Static Pressure**.

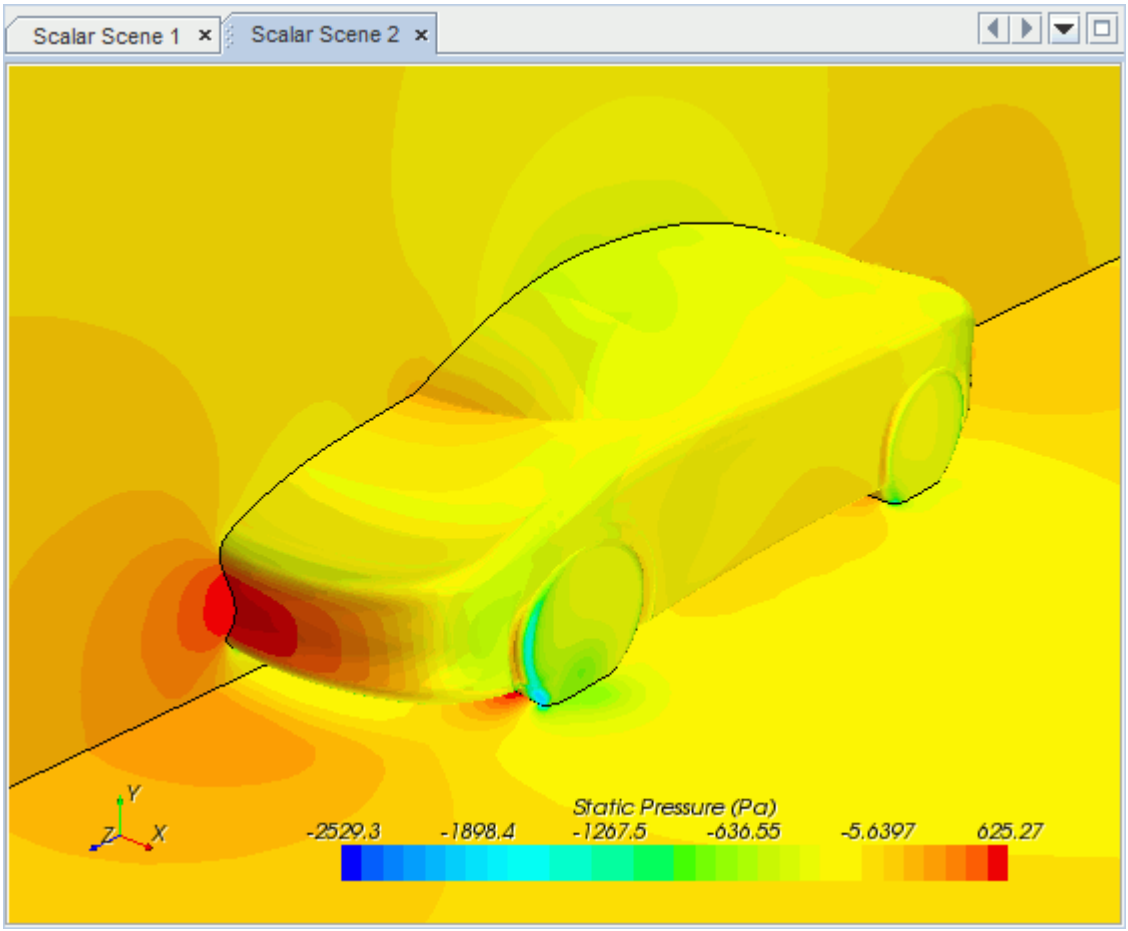
To have a smooth transition between the different pressure values:

8. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1**.



9. Set *Contour Style* to **Smooth Filled**.
10. Rotate the scene to get a better view of the front of the car.
11. Click (**Save**) to save the simulation.

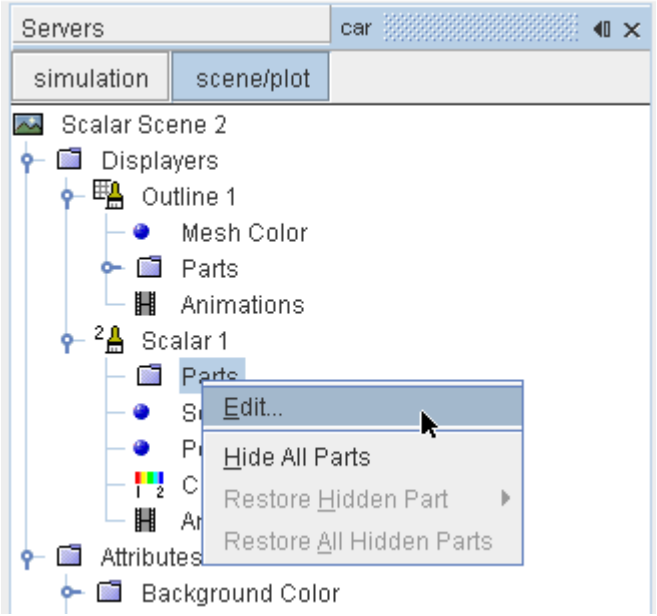
The finished scene is shown below.



Modifying The Scalar Scene

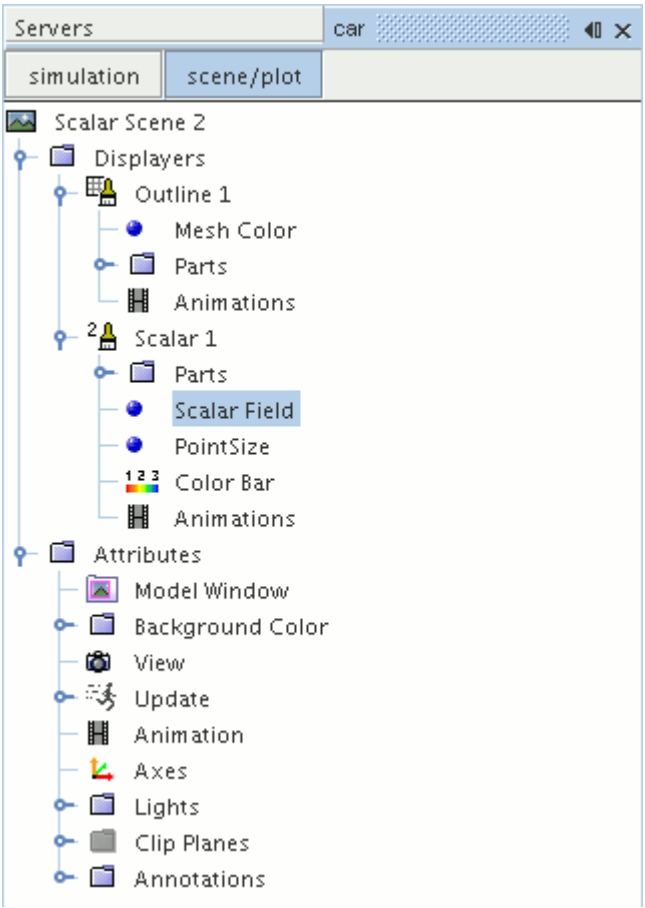
Modify Scalar Scene 2 to display Turbulent Kinetic Energy and remove the symmetry plane from the scene. You also mirror the car body:

1. Right-click the **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Parts** node and select **Edit...**.



The *Parts* dialog appears.

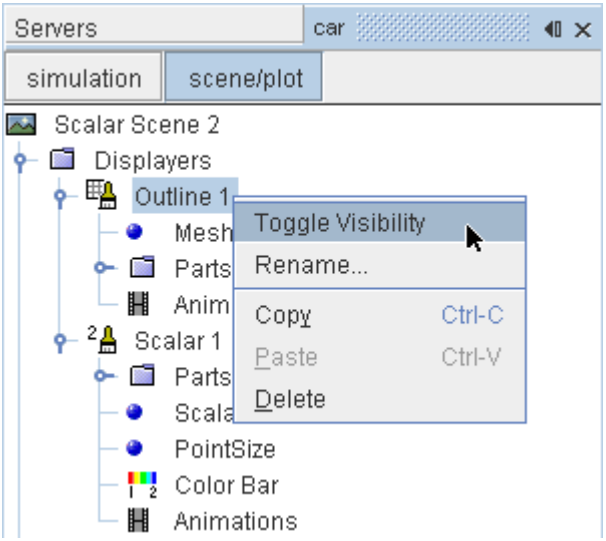
2. Expand the **Regions** > **Region** > **Boundaries** folder and deactivate **Symmetry Plane.**
3. Click **OK**.
4. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Scalar Field**.



5. Set *Function* to **Turbulent Kinetic Energy**.

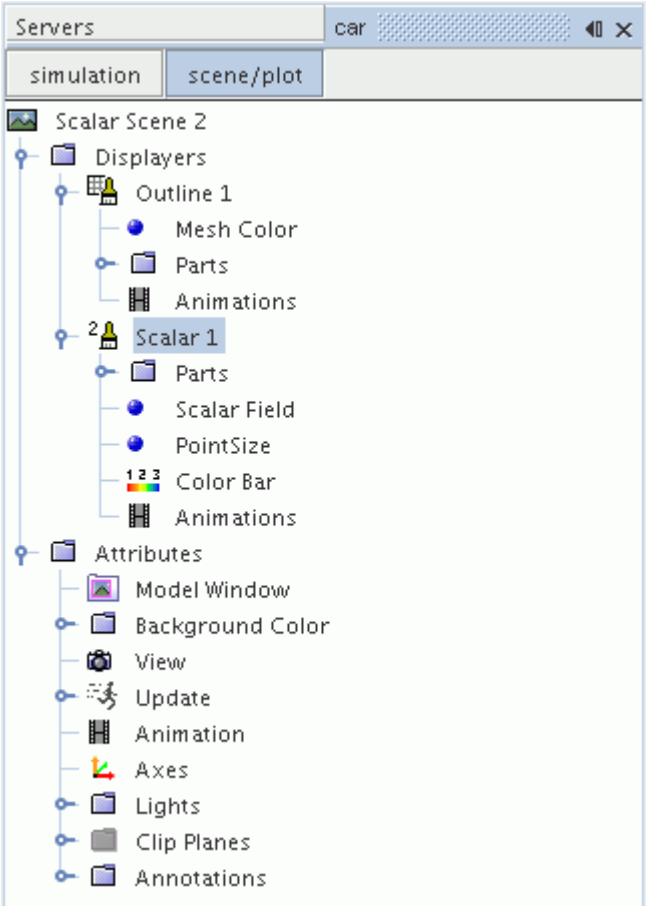
Hide the outlines in the scene:

6. Right-click the **Scalar Scene 2** > **Displayers** > **Outline 1** node and select **Toggle Visibility**.

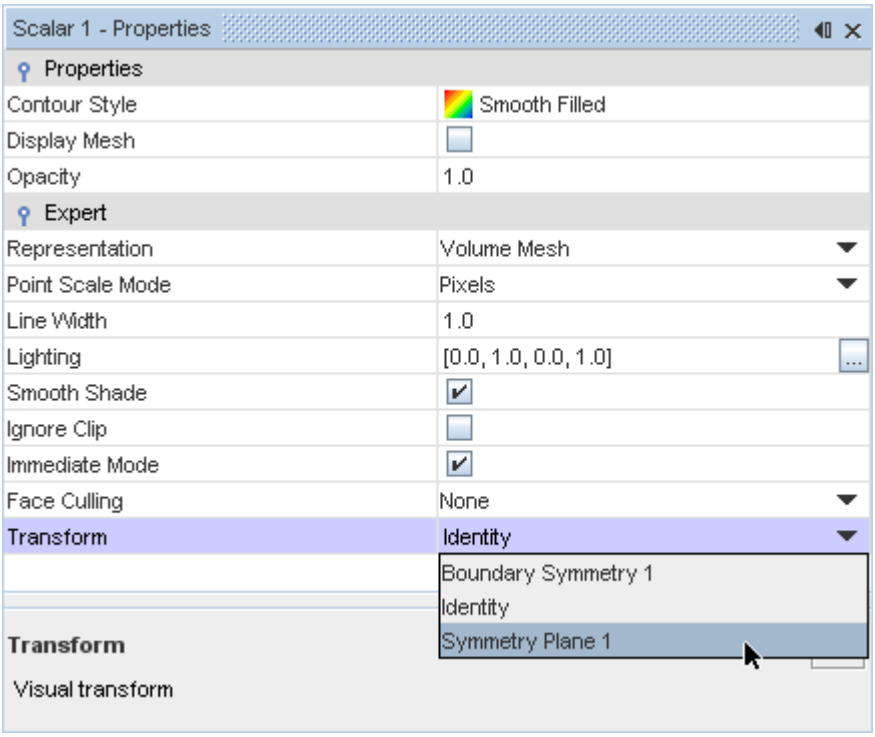


Mirror the car body:

7. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1**.

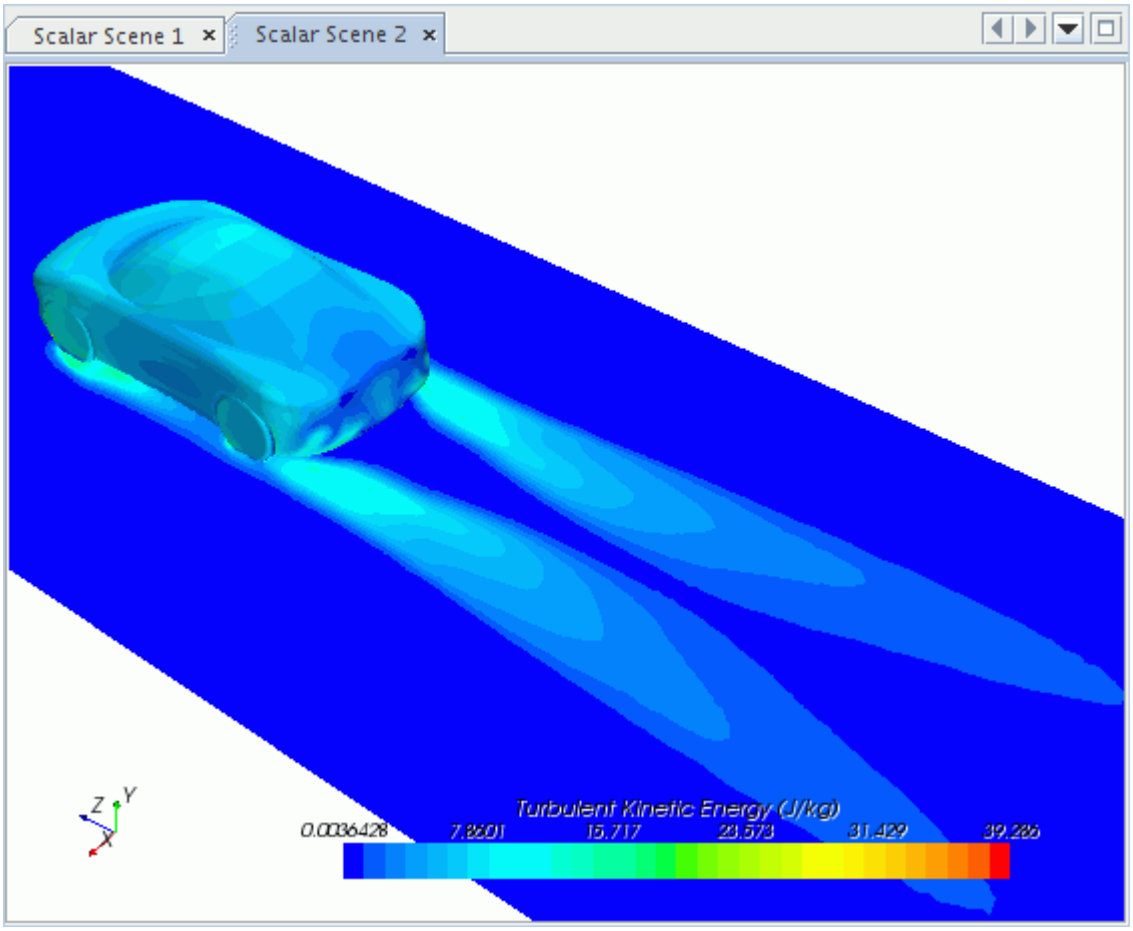


8. Set *Transform* to **Symmetry Plane 1**.



9. Rotate the scene to get a better view of the wake of the car.
10. Click **simulation**.
11. Save the simulation.

The finished scene is shown below.

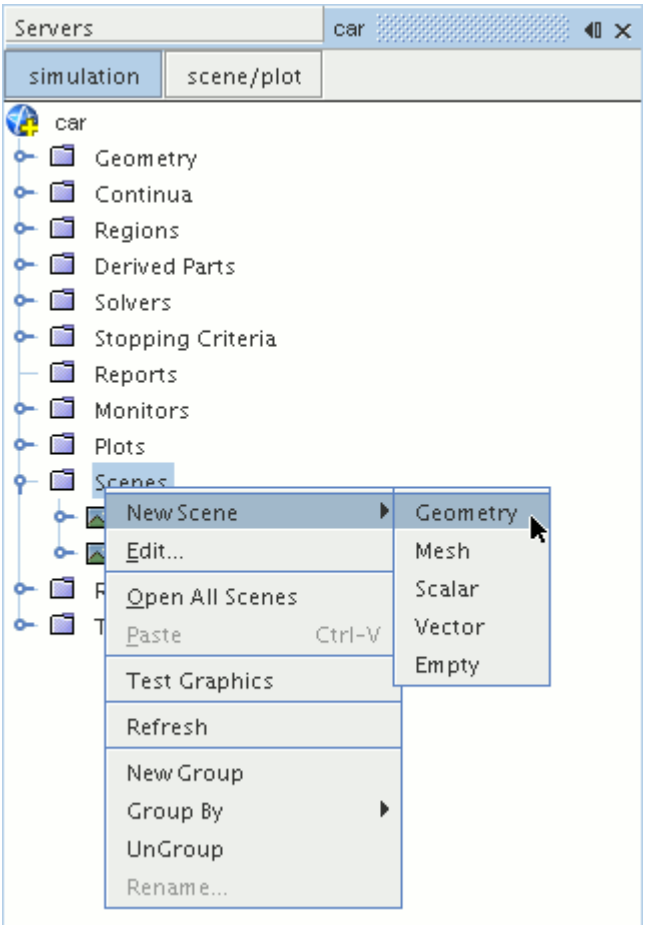


Creating a Streamlines Scene

Create streamlines to get a better understanding of the flow around the car and in the wake area.

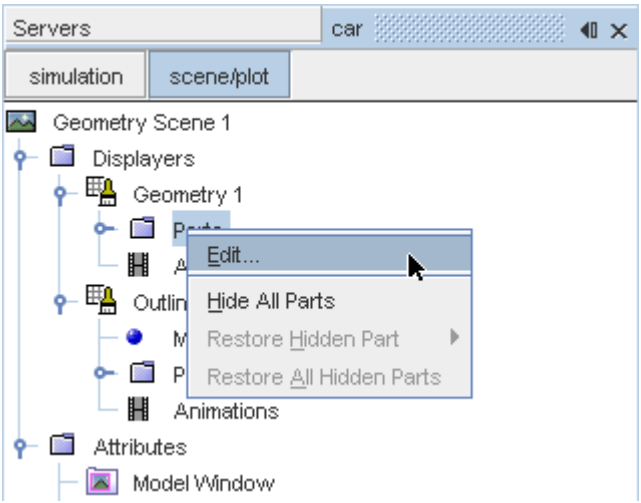
First, create a geometry scene:

1. Right-click the **Scenes** node in the object tree and select **New Scene** > **Geometry**.



A new geometry scene is created and selected in the *Graphics* window.

2. Click **scene/plot**.
3. Right-click the **Geometry Scene 1** > **Displayers** > **Geometry 1** > **Parts** node and select **Edit...**.



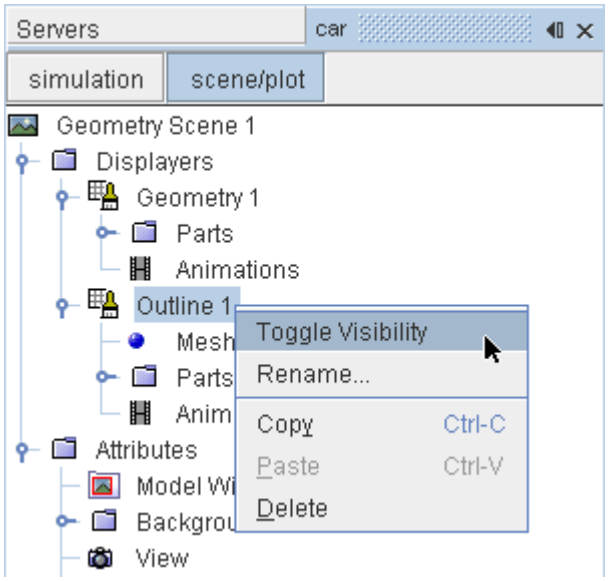
The *Parts* dialog appears.

4. Right-click on a blank area and choose **Deselect All**.

5. Expand the **Regions** > **Region** > **Boundaries** node and select **Car**.
6. Click **OK**.

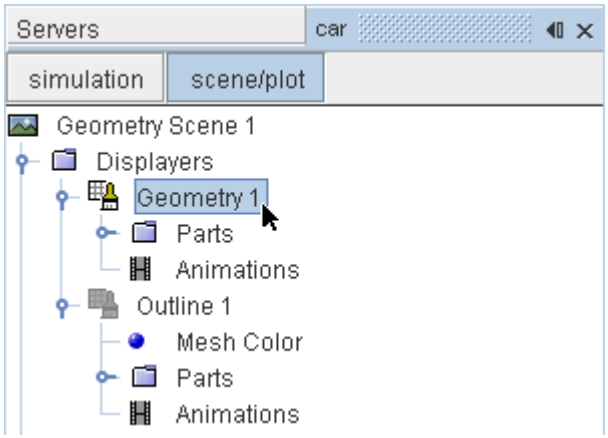
Hide the outlines in the scene:

7. Right-click the **Geometry Scene 1** > **Displayers** > **Outline 1** node and select **Toggle Visibility**.

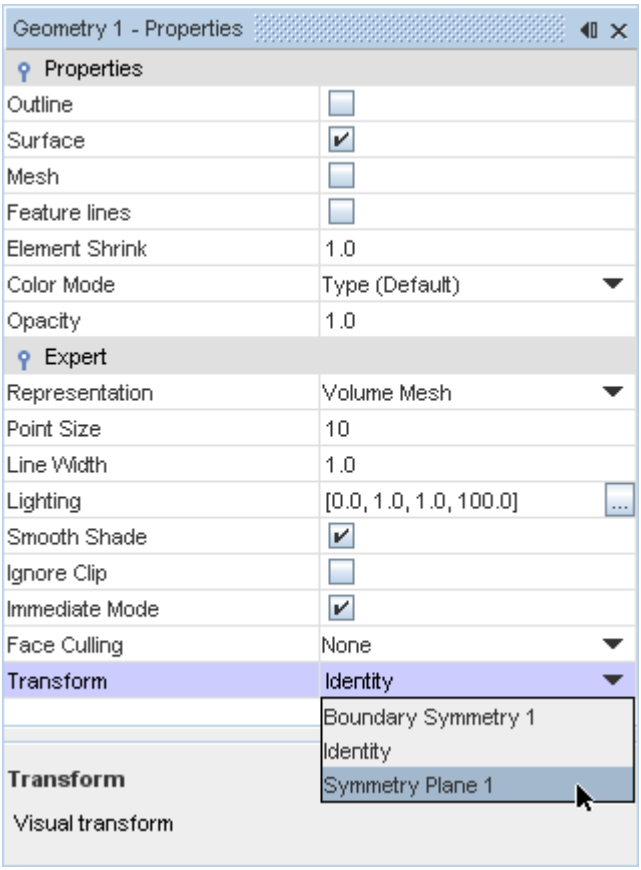


Mirror the car body:

8. Navigate to **Geometry Scene 1** > **Displayers** > **Geometry 1**.



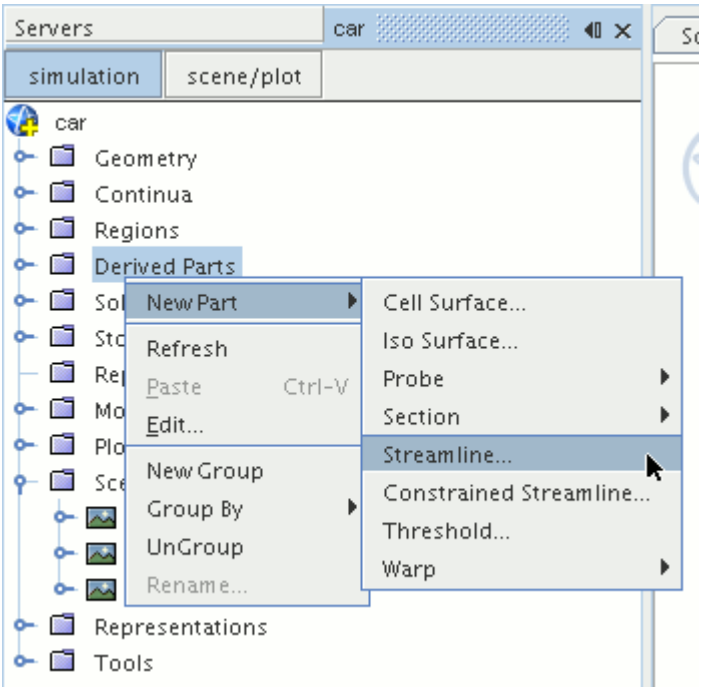
9. Set *Transform* to **Symmetry Plane 1**.



10. Click **simulation**.

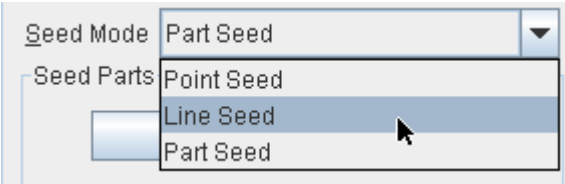
Create the streamlines:

11. Right-click the **Derived Parts** node and select **New Part** > **Streamline...**.



The *Create Streamline* dialog appears.

12. Set *Seed Mode* to **Line Seed**.



13. Enter these coordinates for the line:
    - *Point 1*:
        - X: `0 mm`
        - Y: `-123 mm`
        - Z: `2600 mm`
    - *Point 2*:
        - X: `1200 mm`
        - Y: `-123 mm`
        - Z: `2600 mm`
14. Set *Resolution* to `10`.
15. Make sure that **New Streamline Displayer** is selected in the *Display* section.
16. Click **Create**.

Create another set of streamlines using the existing dialog:

17. Enter these coordinates for the line:
    - *Point 1*:
        - X: `0 mm`
        - Y: `-473 mm`
        - Z: `2600 mm`
    - *Point 2*:
        - X: `1200 mm`
        - Y: `-473 mm`
        - Z: `2600 mm`
18. Click **Create** followed by **Close**.

Modify the appearance of the streamlines to display ribbons instead of the default lines.

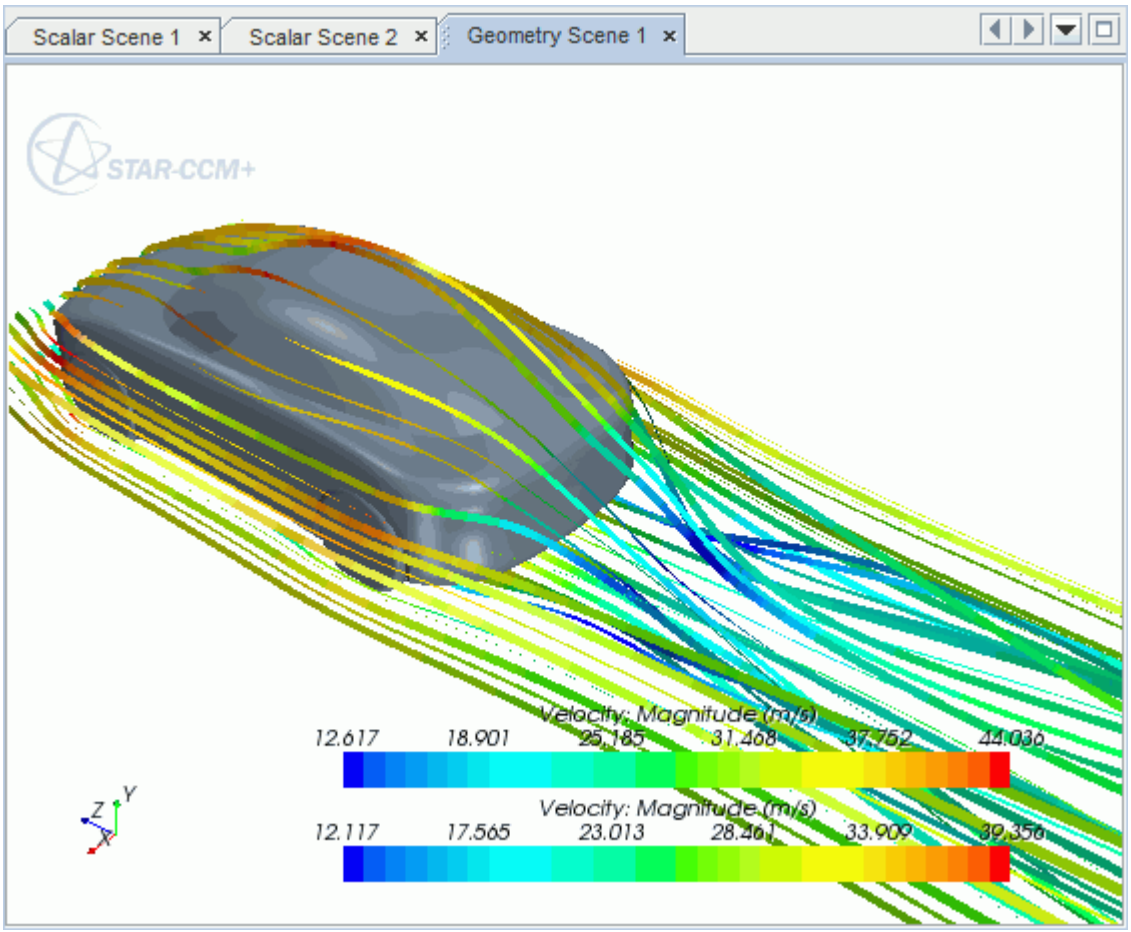
19. Click **scene/plot**.
20. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1**.
21. In the *Properties* window:
    - Set *Width* to `0.025`.
    - Set *Mode* to **Ribbons**.

- Set *Transform* to **Symmetry Plane 1**.

Modify the scalar field of the streamlines to display velocity magnitude:

22. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** > **Scalar Field** and set *Function* to **Velocity** > **Lab Reference Frame** > **Magnitude**.
23. Repeat the above steps for **Streamline Stream 2**.
24. Rotate the scene to get a better view of the wake of the car.
25. Save the simulation.

The finished scene is shown below.



Geometry Transfer: CAD to Simulation

STAR-NX allows you to transfer geometry from NX to STAR-CCM+ through a direct link between the two packages.

This feature provides you with a workflow that is tailored to design optimization. You can apply changes to the geometry in either STAR-NX or STAR-CCM+ and the changes are passed back to the other program. For example, you can make changes to the design after running an analysis in the STAR-CCM+ environment and these changes are passed back to the original NX file. This feature requires a CAD client license.

This tutorial demonstrates how to select the design parameters in your original geometry and transfer the geometry from NX to STAR-CCM+. In STAR-CCM+, you proceed to create a repeatable workflow (or pipeline) using mesh operations. After setting up this pipeline, you can make further changes to the geometry design and have STAR-CCM+ regenerate the corresponding volume mesh.

Contents:

Opening the CAD Model
Transferring the Geometry to STAR-CCM+
Creating the Shape Parts and Operations
Setting Up the Mesh Pipeline
Modifying the Design

Opening the CAD Model

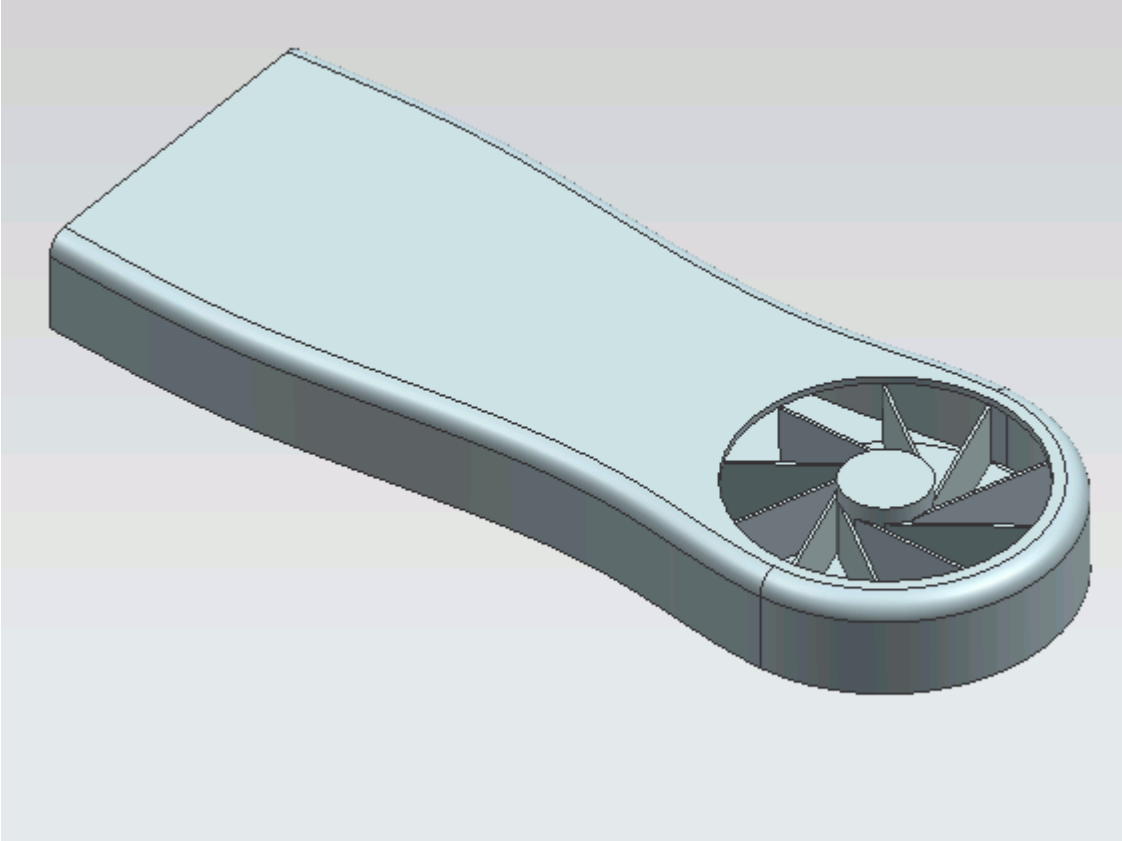
The STAR-CCM+ installation includes a directory containing the CAD geometry for this tutorial.

This directory is located in `[STAR-CCM+_INSTALL_DIR]\doc\startutorialsdata\starnx\data`.

1. Copy directory `data` to your own disk space.
2. Launch STAR-NX and open the assembly file `cardAssem.prt`.

   The assembly is made up of two parts: the card housing and the blower fan.

   The imported geometry is shown below.



The STAR-NX menu appears at the top of the screen. The STAR-NX object tree is located on the left of the screen.

To switch to the object tree:

3. Click (**STAR Tree**).

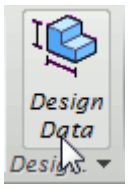
If these items are not visible, refer to Launching STAR-NX Manually.

Transferring the Geometry to STAR-CCM+

In STAR-NX, choose the design parameters to expose to STAR-CCM+ and then transfer the geometry.

Specify the design parameters to transfer to STAR-CCM+:

1. From the STAR-NX ribbon, click **Design Data**.



The *Design Data* dialog appears.

2. In the *Design Parameters* tab, select **blowerFan.prt** > **p60**.

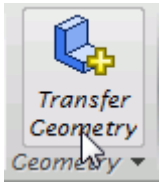
   STAR-NX only recognizes native CAD features as design parameters.

   The value of this parameter is displayed in the *Value* property.

3. Click the checkbox next to the selected parameter to activate it.
4. Click **OK**.

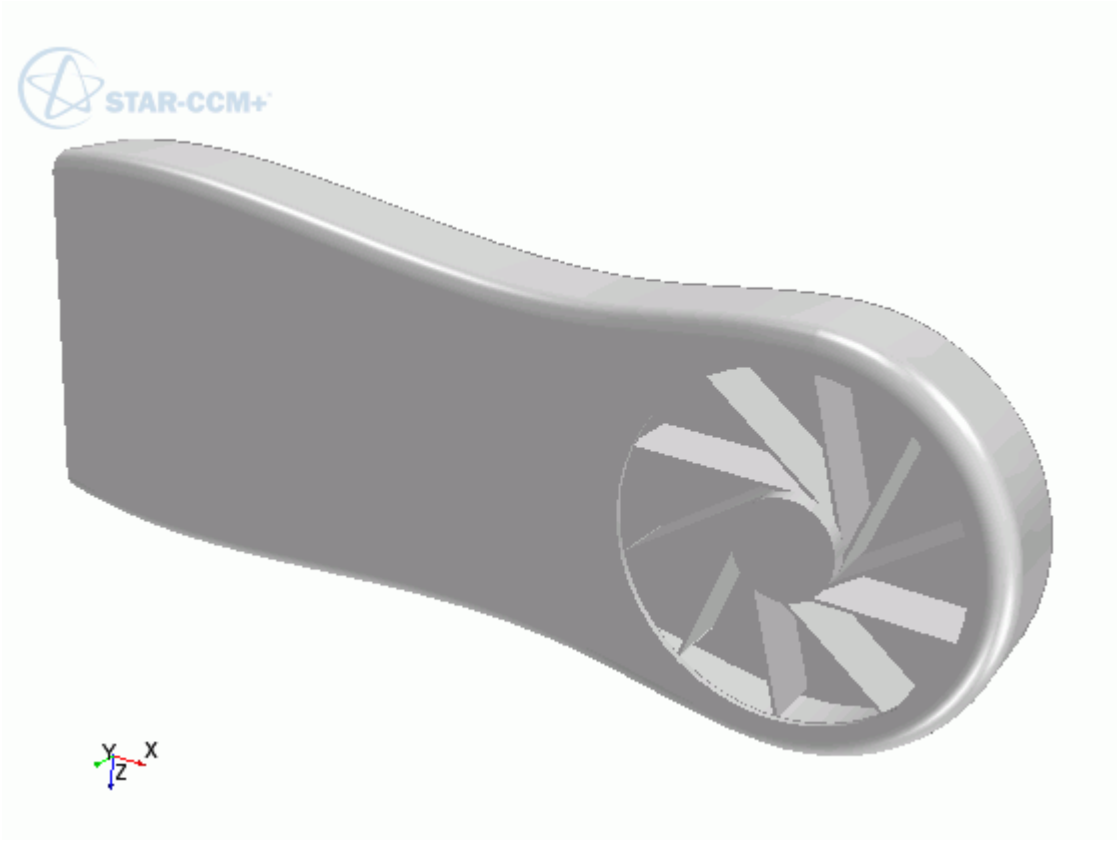
Transfer the geometry to STAR-CCM+:

5. From the STAR-NX ribbon, click **Transfer Geometry**.



   STAR-CCM+ launches and the CAD geometry is transferred over.

6. In STAR-CCM+, create a geometry scene and position the geometry as shown below.



7. In STAR-CCM+, expand the **Geometry** node.

   The transferred geometry appears under the **Parts** node. The properties that control the connection back to NX, including the design parameter value, appear under the **CAD Client Models** node.

When transferring geometry from NX to STAR-CCM+, a link exists between the two software packages. This link allows data to be transferred between the two packages. To maintain this link, you must first save the geometry file in NX:

8. Switch to NX and save the geometry file.

   In response, STAR-CCM+ saves the simulation file with the name of the NX geometry file. Using this technique, you can rename or relocate both the geometry and simulation files without breaking the link between the two packages. You can also save the simulation file in STAR-CCM+ so long as you do not rename it or relocate it. Doing so will break the link between the packages.

Creating the Shape Parts and Operations

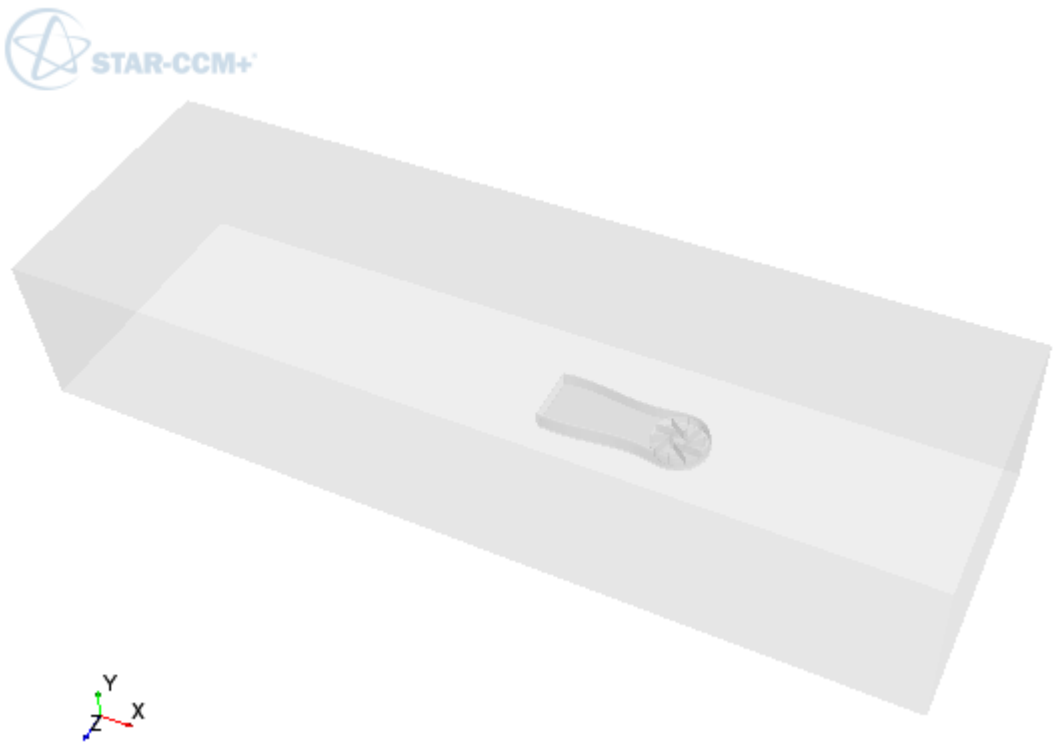
The transferred geometry consists of the card housing and the blower fan. To simulate this case, you require two regions: one static region that provides an enclosure for the simulation, and another for the rotating region around the fan.

The transferred geometry contains the solid parts. You require the fluid volume around these parts. The first stage in obtaining the fluid volume is to create part shapes that correspond to the two regions. To save time, you are provided with these part shapes, one for the enclosure, and another for the rotating region.

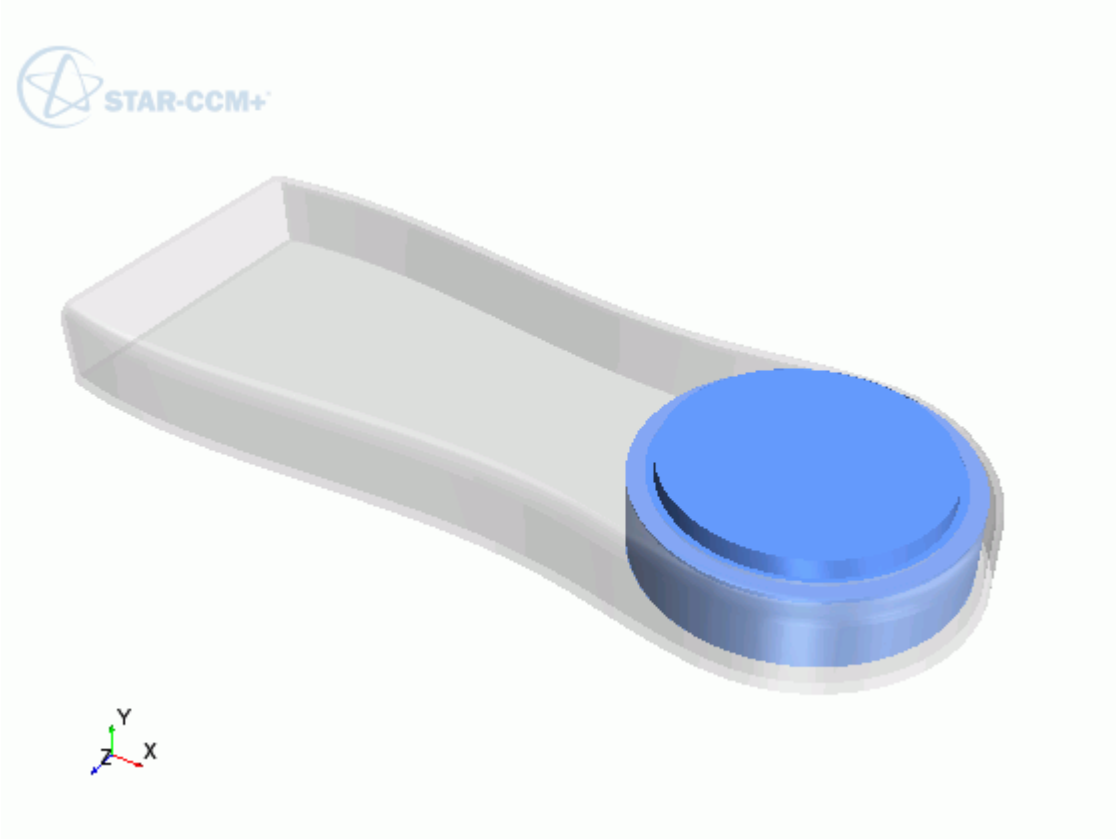
Import the shape parts:

1. Select **File** > **Import** > **Import Surface Mesh**.
2. Import the part shapes:
   a) In the *Open* dialog, navigate to the `data` directory and select **box.x_t** and **disc.x_t**.
   b) Click **Open**.
   c) In the *Import Surface Options* dialog, deactivate the **Open Geometry Scene After Import** option.
   d) Click **OK**.

   The static region, **box**, is shown below in relation to the graphics card:



The rotating region, **disc**, is show below in relation to the graphics card:



Now subtract the solid CAD geometry from these shape parts to obtain the final fluid volume. Begin by subtracting the blower fan geometry from the **disc** part:

3. Right-click the **Operations** node and select **New** > **Boolean** > **Subtract**.
4. In the *Create Subtract Operation* dialog, set the following properties:

Property	Setting
Input Parts	**disc**
	cardAssem > **blowerFan**
Target Parts	**disc**

5. Click **OK**.
6. Rename the **Operations** > **Subtract** node to `Fan Subtraction`.
7. Rename **Parts** > **Subtract** node to `Rotating Region`.

Subtract the card housing geometry and the disc part from the static region:

8. Create another subtract operation with the following settings:

Property	Setting
Input Parts	**box**
	disc
	cardAssem > **cardCase**

Property	Setting
Target Parts	**box**

9. Rename **Operations** > **Subtract** node to `Card Subtraction`.
10. Rename **Parts** > **Subtract** node to `Static Region`.

To get a conformal mesh between the two regions, imprint both fluid volumes together:

11. Right-click the **Operations** node and select **New** > **Boolean** > **Imprint**.
12. In the *Create Imprint Operation* dialog, select **Rotating Region** and **Static Region** and click **OK**.

The transferred geometry contains CAD data. To ensure best results, instruct all operations to use the CAD data where available:

13. Under the **Operations** node, multi-select the **Fan Subtraction** and **Card Subtraction** nodes and activate the *Perform CAD Boolean* option.
14. Select the **Operations** > **Imprint** node and set *Merge/Imprint Method* to **CAD Imprint**.

    The contacts between the two parts are created. These contacts are used while creating the regions to define the interfaces between the regions.

Execute all operations:

15. Right-click on the **Operations** node and select **Execute All**.

Setting Up the Mesh Pipeline

Assign the parts to regions and create the automated mesh operation.

Assign the fluid parts to regions:

1. Expand the **Geometry** > **Parts** node and multi-select the **Rotating Region** and **Static Region** nodes.
2. Right-click on one of the selected parts and choose **Assign Parts to Regions...**.
3. In the *Assign Parts to Regions* dialog:
   a) Select **Create a Region for Each Part**.
   b) Select **Create Boundary for Each Part Surface**.
   c) Click **Apply** and **Close**.

Create the automated mesh operation:

4. Right-click the **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
5. In the *Create Automated Mesh Operation* dialog:
   a) From the *Parts* list, select **Rotating Region** and **Static Region**.
   b) Select the following meshers, in order:

Group	Mesher
Surface Meshers	**Surface Remesher**
Volume Meshers	**Polyhedral Mesher**
Optional Boundary Layer Meshers	**Prism Layer Mesher**

c) Click **OK**.

An automated mesh operation is added to the **Operations** node. Refine the mesh settings for this case:

6. Expand the **Automated Mesh** > **Default Controls** node.
7. Select the **Base Size** node and set *Value* to `16 mm`.
8. Select the **Prism Layer Total Thickness** > **Relative Size** node and set *Percentage of Base* to `20`.

Generate the volume mesh:

9. Click (**Generate Volume Mesh**).
10. When the volume mesh is generated, create a mesh scene that displays the following parts:
    - **Rotating Region** > **Boundaries** > **cardAssem.blowerFan.Faces**
    - **Static Region** > **Boundaries** > **cardAssem.cardCase.Faces**

You can exit STAR-CCM+ and NX and maintain the link between the two packages. To demonstrate, in the following steps you save the simulation and exit both applications. In the next section, you will resume the simulation and continue where you left off.

11. Save the simulation.
12. Exit STAR-CCM+.
13. Exit NX

Modifying the Design

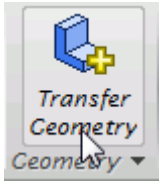
Modify the number of blades on the blower fan using the design parameter.

In this tutorial, you modify the number of blades on the graphics card blower fan from 10 to 15 blades.

To re-establish the link between NX and STAR-CCM+, you must load the geometry in STAR-NX first and transfer the geometry to STAR-CCM+.

Resume the analysis and re-establish the link between STAR-NX and STAR-CCM+:

1. Launch NX with STAR-NX and load `cardAssem.prt`.
2. From the STAR-NX ribbon, click **Transfer Geometry**.



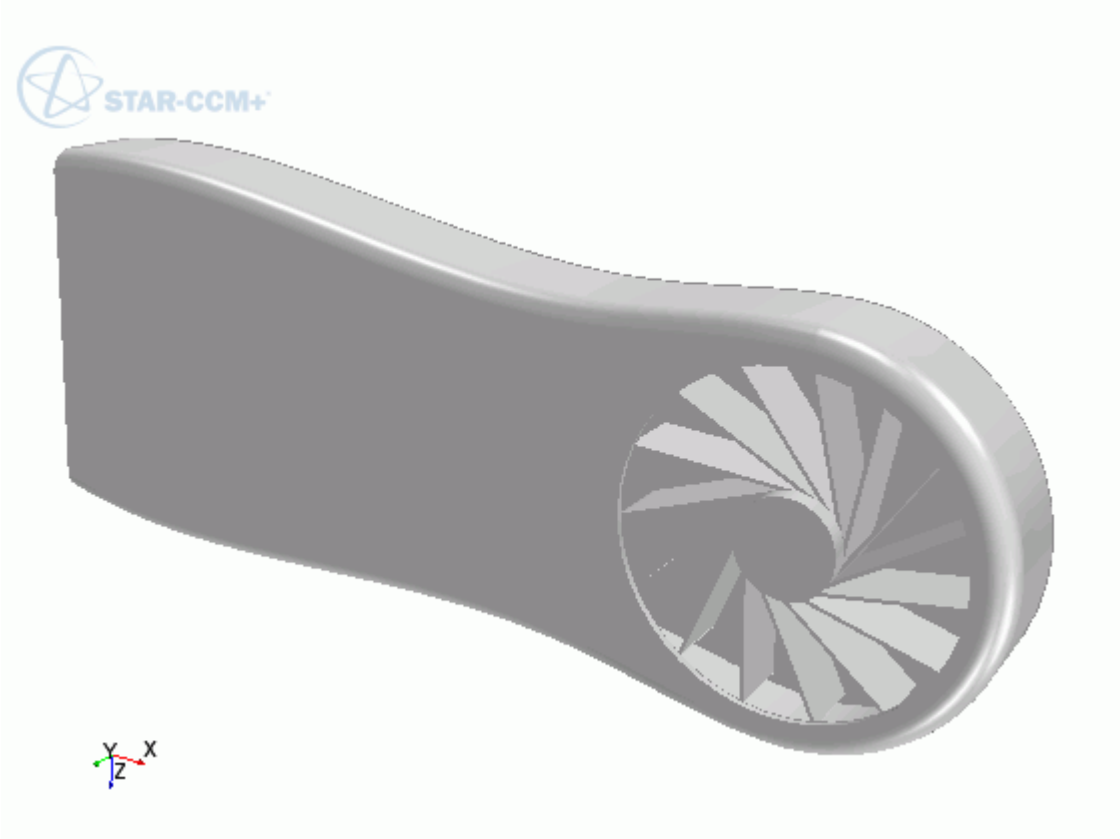
   STAR-CCM+ launches and the CAD geometry is transferred over.
3. Switch to STAR-CCM+.
4. Open the geometry scene to visualize the initial geometry representation.
5. Expand the **Geometry** > **CAD Client Models** > **cardAssem.prt** > **Design Parameters** node.
6. Select the **blowerFan.prt\p60** node and set *Parameter Value* to `15`.

   A yellow warning icon appears on the design parameter node as well as the assembly node. This icon indicates that the CAD Client Model is out of date.

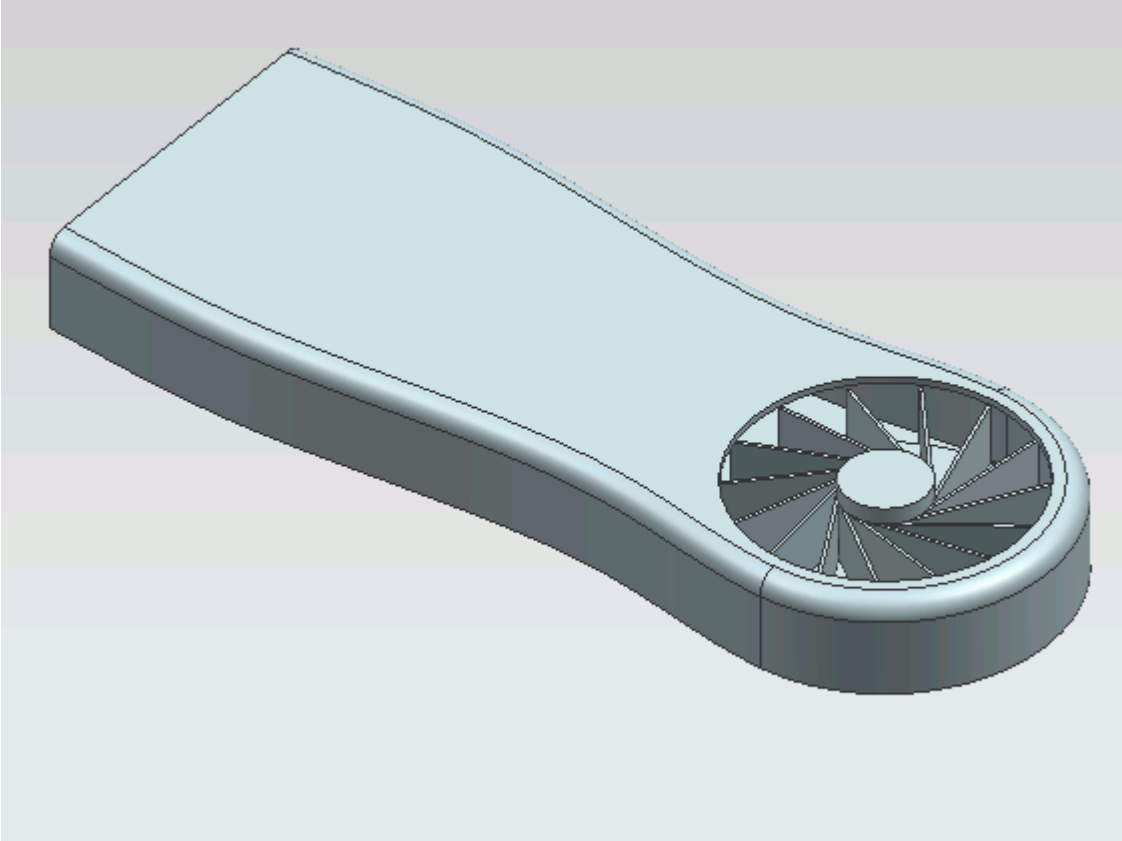
To update the model:

7. Right-click the **cardAssem.prt** node and select **Update Model**.

   The geometry updates in the scene. STAR-CCM+ passes the change in the design back to NX.



8. Switch back to NX and view the updated geometry.



The pipeline you set up is fully repeatable. To update the volume mesh:

9. In STAR-CCM+, open the mesh scene.

10. Click (**Generate Volume Mesh**).
    The volume mesh updates to reflect the changes in the design.

All boundaries and boundary conditions that you set are maintained after this change. This feature reduces the amount of work that is required when analyzing multiple designs.

Siemens PLM Software

Headquarters
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

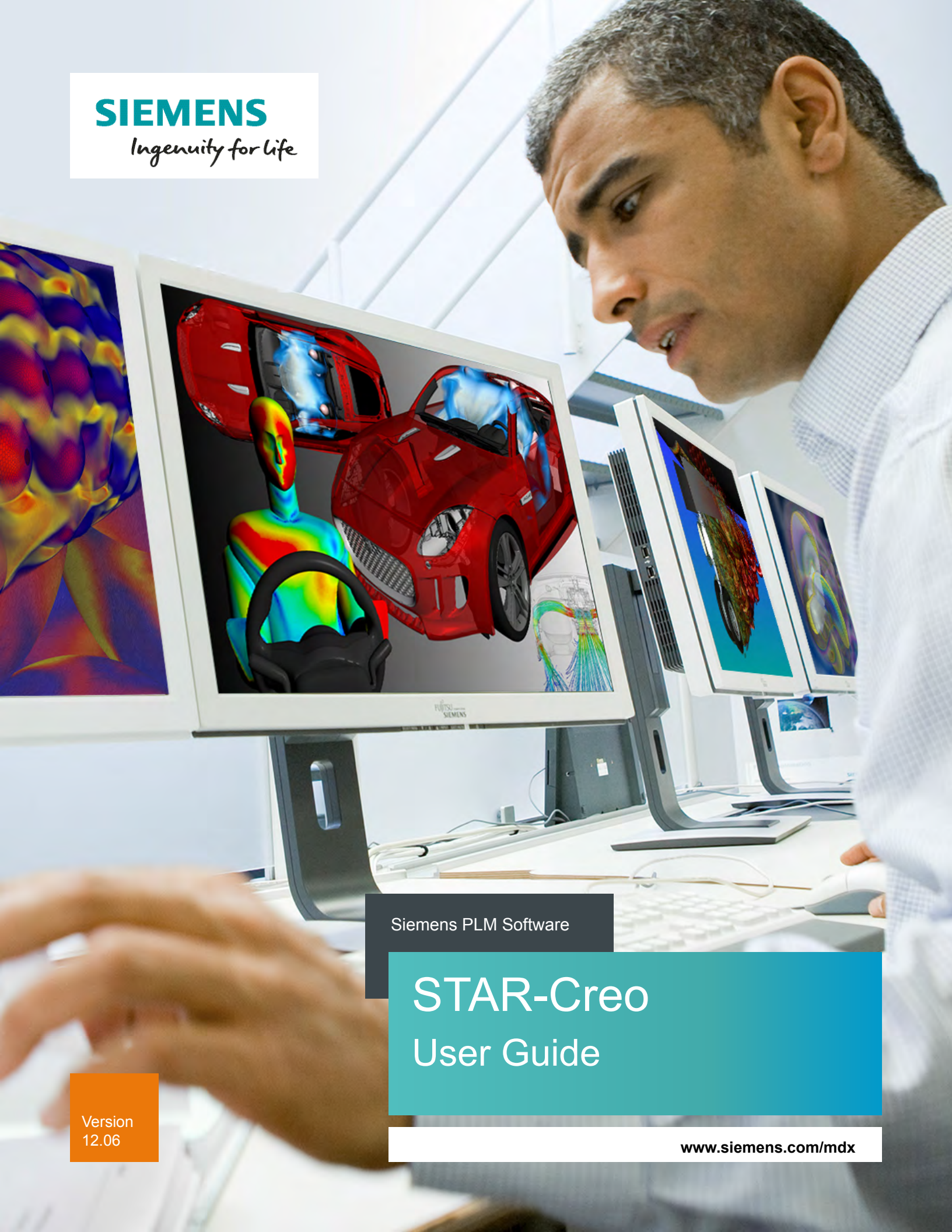
Asia-Pacific
Suites 4301-4302, 43/F
AIA Kowloon Tower,
Landmark East
100 How Ming Street
Kwun Tong, Kowloon
Hong Kong
+852 2230 3308

About Siemens PLM Software

Siemens PLM Software, a business unit of the Siemens Digital Factory Division, is a leading global provider of product lifecycle management (PLM) and manufacturing operations management (MOM) software, systems and services with over 15 million licensed seats and more than 140,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software works collaboratively with its customers to provide industry software solutions that help companies everywhere achieve a sustainable competitive advantage by making real the innovations that matter. For more information on Siemens PLM Software products and services, visit www.siemens.com/plm.

© 2017 Siemens Product Lifecycle Management Software Inc. Siemens and the Siemens logo are registered trademarks of Siemens AG. D-Cubed, Femap, Fibersim, Geolus, GO PLM, I-deas, JT, NX, Parasolid, Polarion, Solid Edge, Syncrofit, Teamcenter and Tecnomatix are trademarks or reg istered trademarks of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries. All other logos, trademarks, registered trademarks or service marks belong to their respective holders.

SIEMENS
Ingenuity for life



Siemens PLM Software

STAR-Creo
User Guide

Version
12.06

www.siemens.com/mdx

© 2017 Siemens Product Lifecycle Management Software Inc.

Any and all Siemens Product Lifecycle Management Software Inc. brand, product, service and feature names, logos and slogans are registered trademarks or trademarks of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries. All other brand, product, service and feature names or trademarks are the property of their respective owners.

Contents

STAR-Creo **1**
STAR-Creo User Guide **2**
Introducing STAR-Creo 3
Transferring Geometry to STAR-CCM+ 35
Understanding the STAR-Creo Process 43
Working with Simplified Representations in STAR-Creo 47
Connecting to Remote Machines 48
Defining Materials and Physics 52
Setting Boundary Conditions 75
Meshing the Model 82
Obtaining a Solution 103
Exporting Files and Data 112
Resetting STAR-Creo Settings 116
STAR-Creo Tutorials **117**
Cold Flow Through a Motorcycle Exhaust Manifold 118
Porous Media Flow 141
Aircraft Wing Deicing System 163
Rotating Flow Inside a Graphics Card 196
External Flow Over a Car 226
Geometry Transfer: CAD to Simulation 256

STAR-Creo

The STAR-Creo CAD client provides you with straightforward access to CD-adapco's STAR-CCM+ solver from within the Creo environment.

Standard physics models, including turbulence, ideal gas, heat transfer, and gravity are supported for single body cases. Multiple body simulations involving conjugate heat transfer, rotating reference frames, and porous media are also supported. Advanced physics models can be added to the simulation by either launching STAR-CCM+ and selecting them directly, or by implementing additional Java macro commands.

This section contains the user guide and tutorials for the STAR-Creo CAD client.

Contents:

STAR-Creo User Guide
STAR-Creo Tutorials

STAR-Creo User Guide

This part of the documentation details how to create, set up, and run a Computational Fluid Dynamics (CFD) simulation in STAR-Creo.

This section assumes that you have previous experience in using Creo, and only describes the additional features available as part of STAR-Creo.

You can use STAR-Creo to transfer your CAD geometry directly into STAR-CCM+ without transferring the CAE settings. This method is an alternate way importing geometry to using the STAR-CCM+ CAD translators. See Transferring Geometry to STAR-CCM+.

Contents:

Introducing STAR-Creo
Transferring Geometry to STAR-CCM+
Understanding the STAR-Creo Process
Working with Simplified Representations in STAR-Creo
Connecting to Remote Machines
Defining Materials and Physics
Setting Boundary Conditions
Meshing the Model
Obtaining a Solution
Exporting Files and Data
Resetting STAR-Creo Settings

Introducing STAR-Creo

STAR-Creo provides you with straightforward access to CD-adapco's STAR-CCM+ solver from within the Creo environment.

Standard physics models, including turbulence, ideal gas, heat transfer, and gravity are supported for single body cases. Multiple body simulations involving conjugate heat transfer, rotating reference frames, and porous media are also supported. Advanced physics models can be added to the simulation by either launching STAR-CCM+ and selecting them directly, or by implementing additional Java macro commands.

Contents:

What are the Capabilities?
Getting Started
Common Functions
Running STAR-Creo and STAR-CCM+ on Different Machines

See Also:

Using Java Macros in STAR-Creo

What are the Capabilities?

This section highlights the capabilities of STAR-Creo.

STAR-Creo enables you to:

- Define flow characteristics for the simulation
- Specify mesh parameters and controls
- Execute the STAR-CCM+ solver within STAR-Creo, with specified solution controls
- Post-process results in STAR-CCM+
- Use parallel processing

Problem Definition

STAR-Creo provides an interface for the specification of continua, regions, and boundaries in order to define the CFD simulation. This information is then used by STAR-CCM+ to calculate a solution and present results.

The particular solver options that you can currently select through the STAR-Creo interface include:

- Turbulence
  - k-e Model
  - Laminar
- Temperature
  - Thermal Modelling
  - Conjugate Heat Transfer

- Equation of State
  - Constant Density
  - Ideal Gas
- Porous Media
  - Isotropic
  - Uni-Directional
- Moving Reference Frames
- Energy Source
- Gravity
- Passive Scalar

Meshing

STAR-Creo provides you with a transparent interface to CD-adapco's surface and volume meshing tools. This allows you to generate high-quality meshes using a solid model that was built within the Creo environment. The specific meshing functionality includes:

- Surface meshing using the surface remesher for optimum mesh quality
- Prismatic subsurface meshing in the volume mesh for capturing near-wall boundary layers
- Volume meshing of parts with either:
  - Tetrahedral core mesh generation
  - Polyhedral core mesh generation
  - Trimmed hexahedral core mesh generation
- Automatic handling of region-to-region interfaces

Running the Solver

STAR-Creo allows the generation of solutions with varying accuracy and speed by providing the following solver controls:

- Number of iterations
- Stopping criterion at which to assume convergence
- Discretization scheme (1st or 2nd order)
- Solver under-relaxation factors
- Additional commands using Java macros

Post-processing

As post-processing is carried out in STAR-CCM+, a wide range of tools are available for analysis. These include:

- Scalar scenes
- Vector scenes
- Streamlines
- Isosurfaces
- Plane sections

- Reports
- Plots

Parallel Processing

STAR-Creo supports parallel processing for mesh generation and solver calculations. You can set up parallel processing in the Preferences dialog. The different STAR-CCM+ licensing options can also be selected here. See Setting Up Parallel Processing and Licensing.

Getting Started

The following section provides information on familiarization with the STAR-Creo interface, and basic concepts.

Contents:

Launching STAR-Creo Manually
The STAR-Creo Info App Window
Modifying the config.xml File
Understanding the STAR-Creo Interface
The STAR-Creo Object Tree
Setting Up Parallel Processing and Licensing
Changing the Units System
Transferring the CAE Model to STAR-CCM+
Using Creo Mouse Controls in STAR-CCM+
Editing STAR-Creo Settings While the STAR-CCM+ Client is Running
Using STAR-Creo with Multiple Installed Versions of Creo
Disabling STAR-Creo

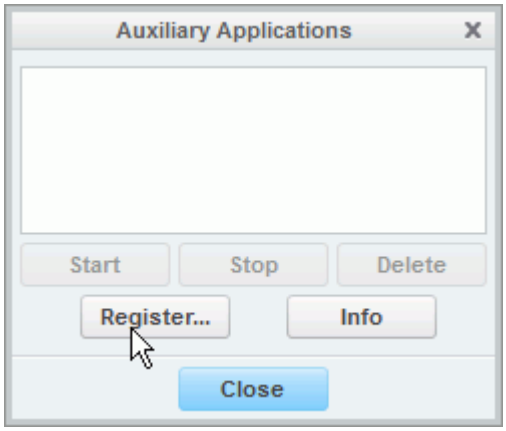
Launching STAR-Creo Manually

By default, STAR-Creo starts automatically when you launch Creo. The *STAR-Creo Info App* window appears, along with the STAR-Creo object tree and ribbon.

If this does not happen, you can manually launch STAR-Creo by following the steps outlined below:

1. Start Creo.
2. Create an empty part or load an existing part.
3. In the ribbon, select **Auxiliary Applications** from the *Tools* tab.

   The *Auxiliary Applications* dialog appears.
4. Click **Register...**.



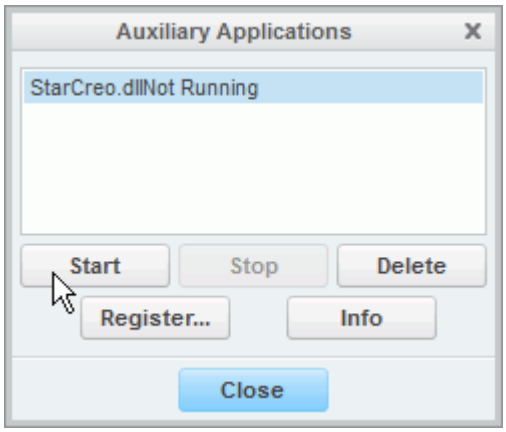
5. Browse to the `protk.dat` file in your STAR-Creo installation directory.

   Typically this file is located in `C:\Program Files\CD-adapco\STARCAD[VERSION]\STAR-Creo`.

6. Select this file and click **Open**.

   A entry stating that `starproe.dllNot Running` appears in the *Auxiliary Applications* dialog.

7. Select this entry and click **Start**.



   The *STAR-Creo Info App* window appears and the checks for licensing begin. If everything is in order, a `Licensing successful` message appears in the *Status* page of the STAR-Creo *Info App* window, and the `StarCreo.dll Not Running` message vanishes from the *Auxiliary Applications* dialog.

8. Click **Close** in the *Auxiliary Applications* dialog.
9. Close the open geometry file and reopen it again.

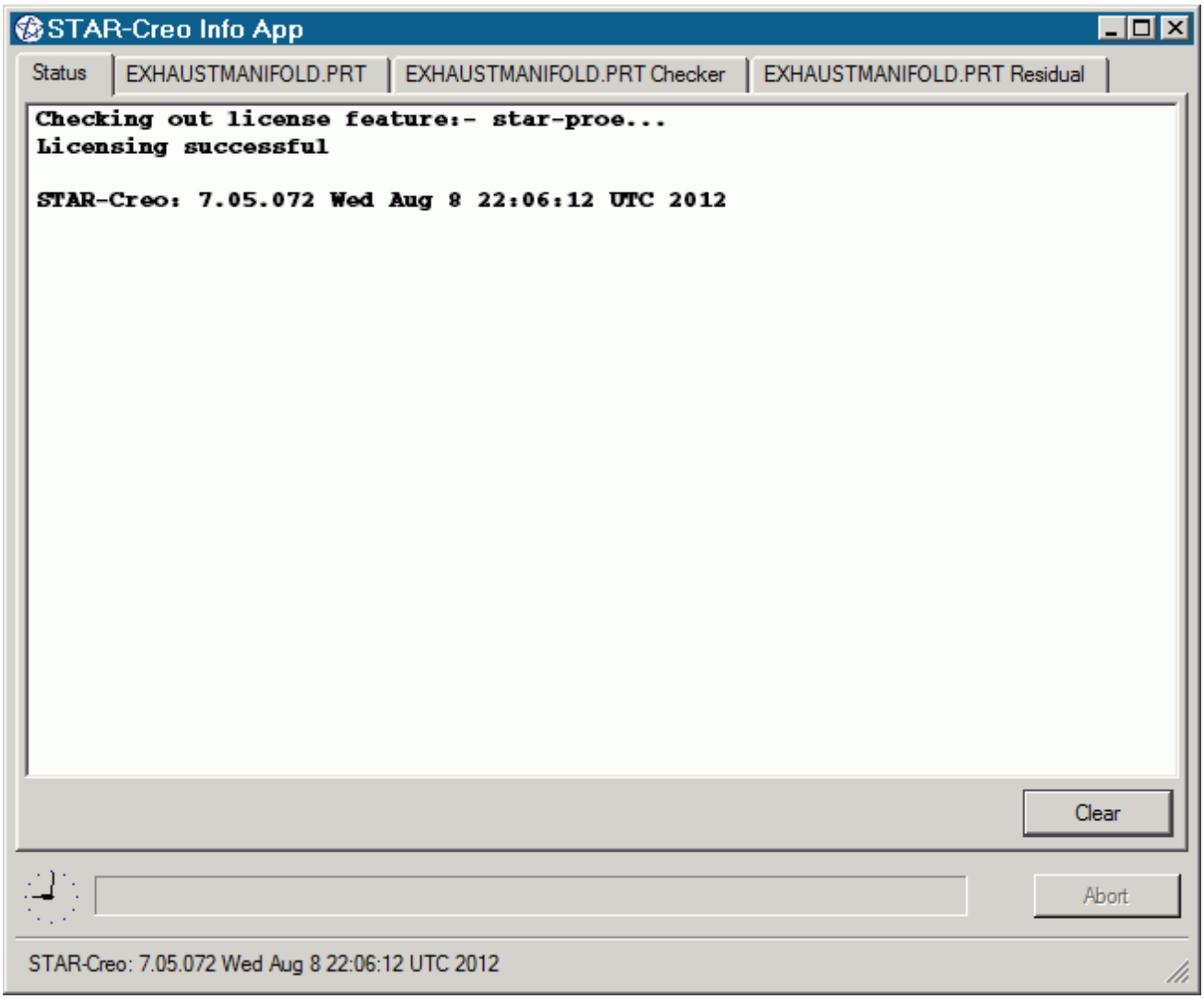
The (STAR-Creo) icon appears on the left of the screen when you load or create a geometry file. Click on this icon to display the object tree.

The STAR-Creo ribbon also appears.

The STAR-Creo Info App Window

The STAR-Creo Info App window automatically appears when you start Creo.

Initially, this window only contains the *Status* tab which displays the version and licensing information for STAR-Creo. More tabs appear as you create or load geometry files in STAR-Creo, allowing you to perform more actions.



You cannot close the STAR-Creo Info App window while Creo or STAR-Creo are running. Attempting to close the window by clicking the **X** button returns a message in the *Status* tab informing you of this. You can maximize and minimize the window by clicking the **-** and **_** buttons respectively.

Use the **Clear** button to clear all the information that is displayed in a particular tab.

Use the **Abort** button to stop active processes, such as surface/volume meshing, or the solver. This button only appears when you can stop a process — it is grayed out at all other times.

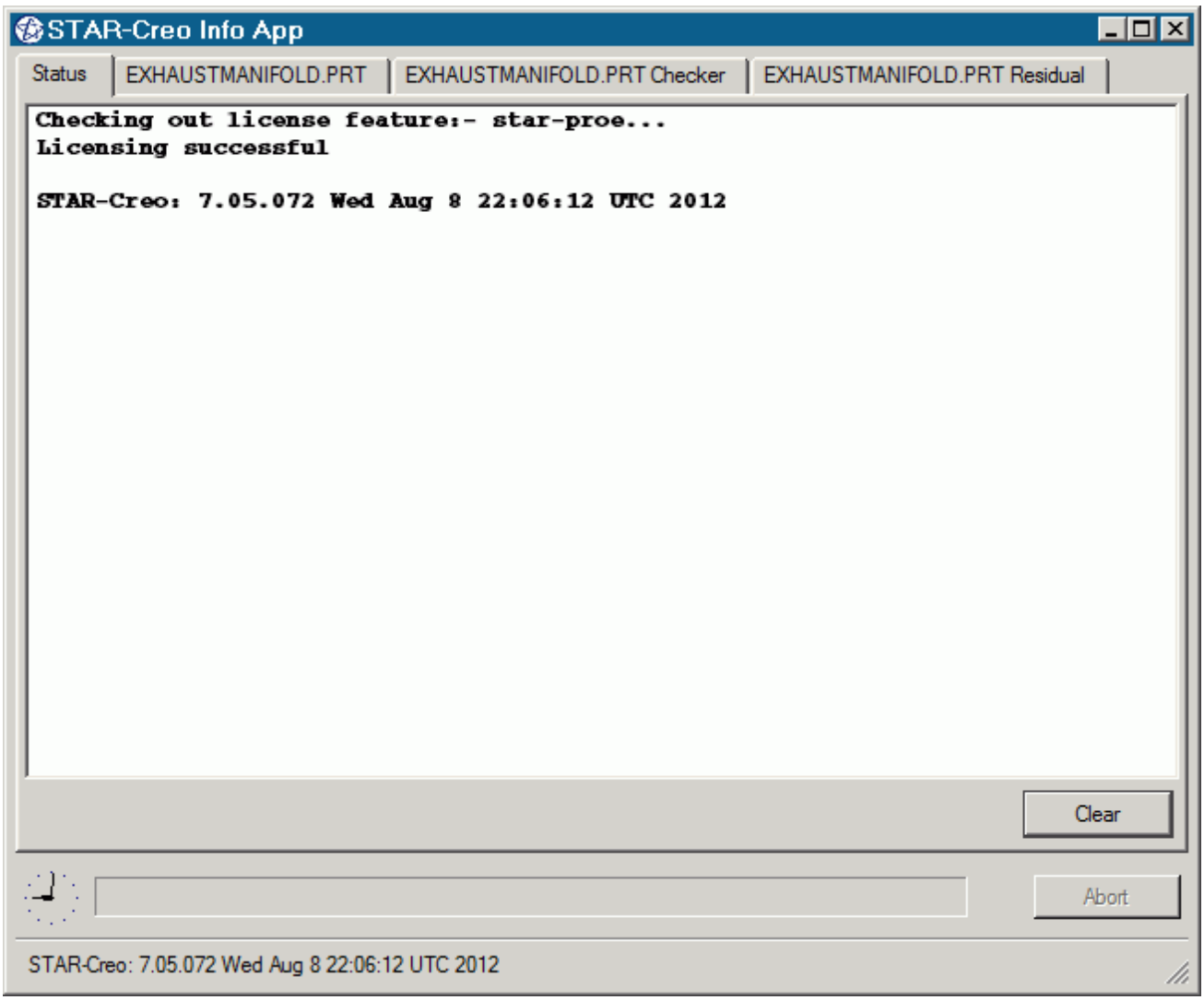
A clock is displayed in the bottom-left of the window. This clock displays the current time on your machine. A progress bar is located next to the clock and reports the progress of various processes, such as mesh generation and the solver.

More information on the various tabs is provided in the following sections:

- The Status Tab
- The Output Tab
- The Model Checker Tab
- The Residuals Tab

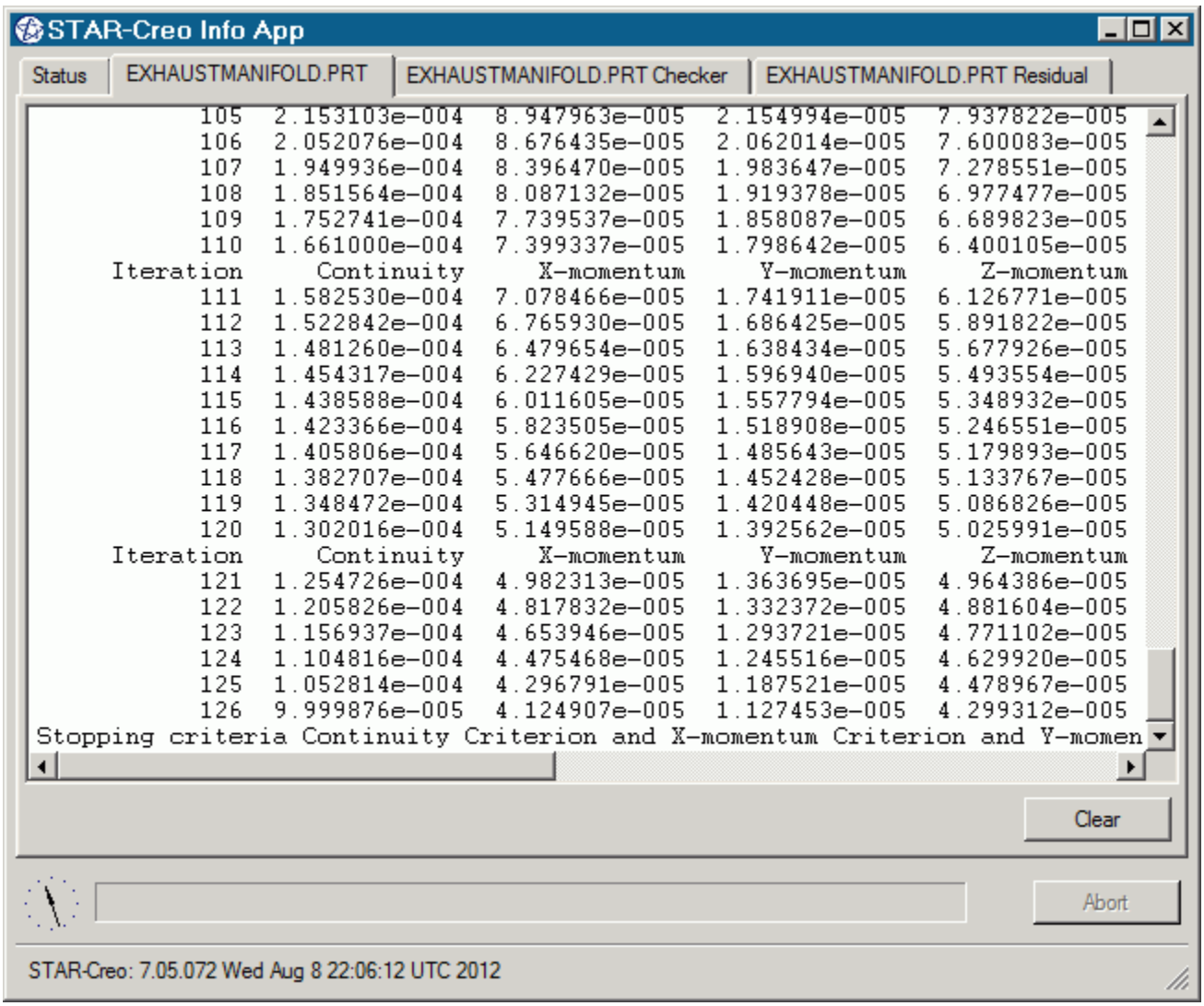
The Status Tab

The *Status* tab displays the licensing information for your current session, informing you whether the STAR-Creo license checked out successfully. The version of STAR-Creo is also displayed in this tab. This tab appears by default when you launch Creo without a geometry file.



The Output Tab

The *Output* tab appears once you create or open a geometry file in STAR-Creo.



During the meshing process, this tab displays status information along with a brief report on the generated mesh. If errors were encountered during the meshing process, they are reported here with suggestions as to what can be causing them.

When the solver is running, the *Output* tab behaves like the output window in STAR-CCM+, displaying the data for each residual per iteration. When the solver finishes running, the same information is also displayed in the STAR-CCM+ *Output* window.

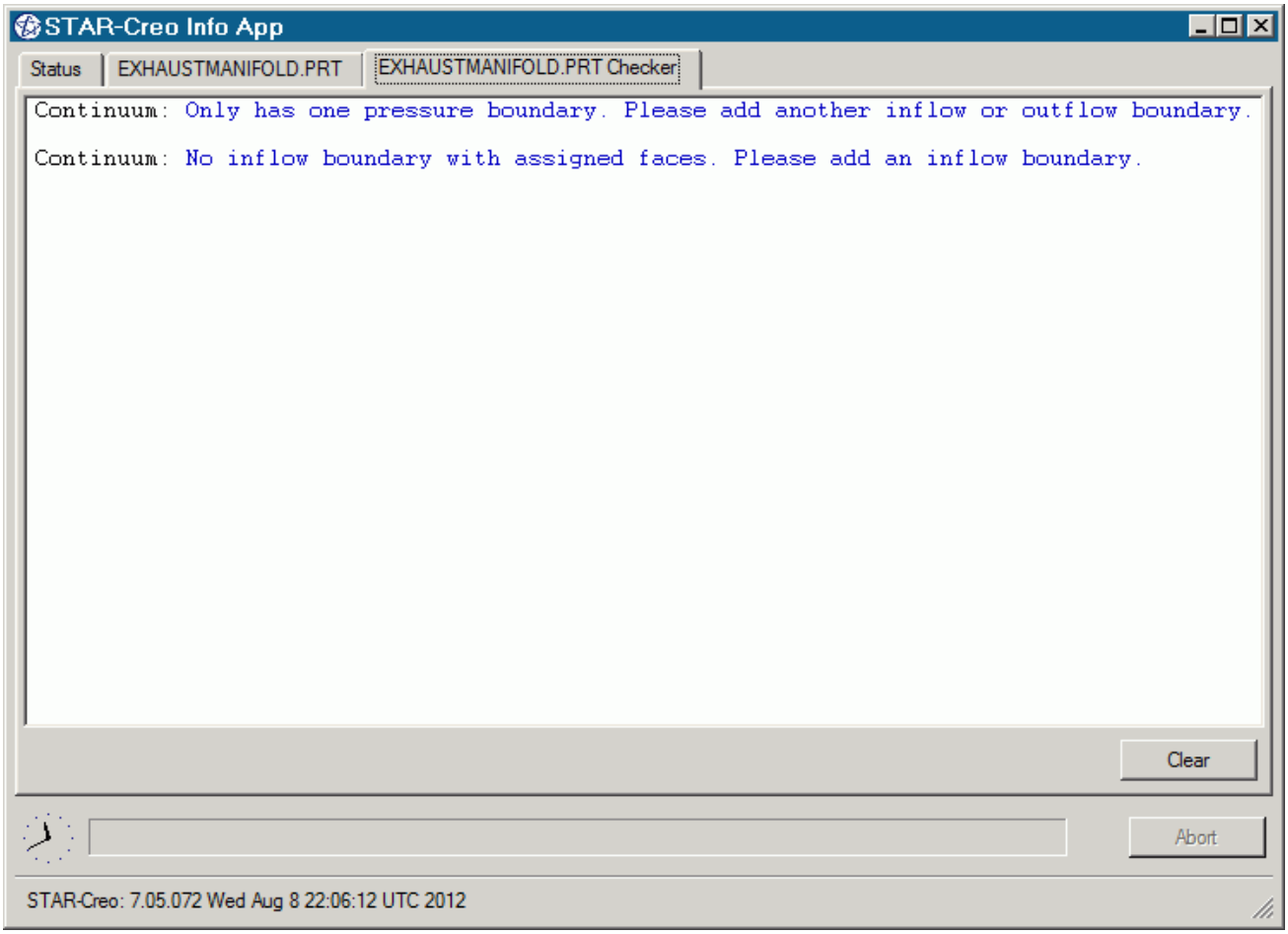
You can control the amount of information that is displayed in the *Output* tab by editing the verbosity in the `config.xml` file. See Modifying the Config.xml File.

To abort the mesher or solver, click the **Abort** button.

The Model Checker Tab

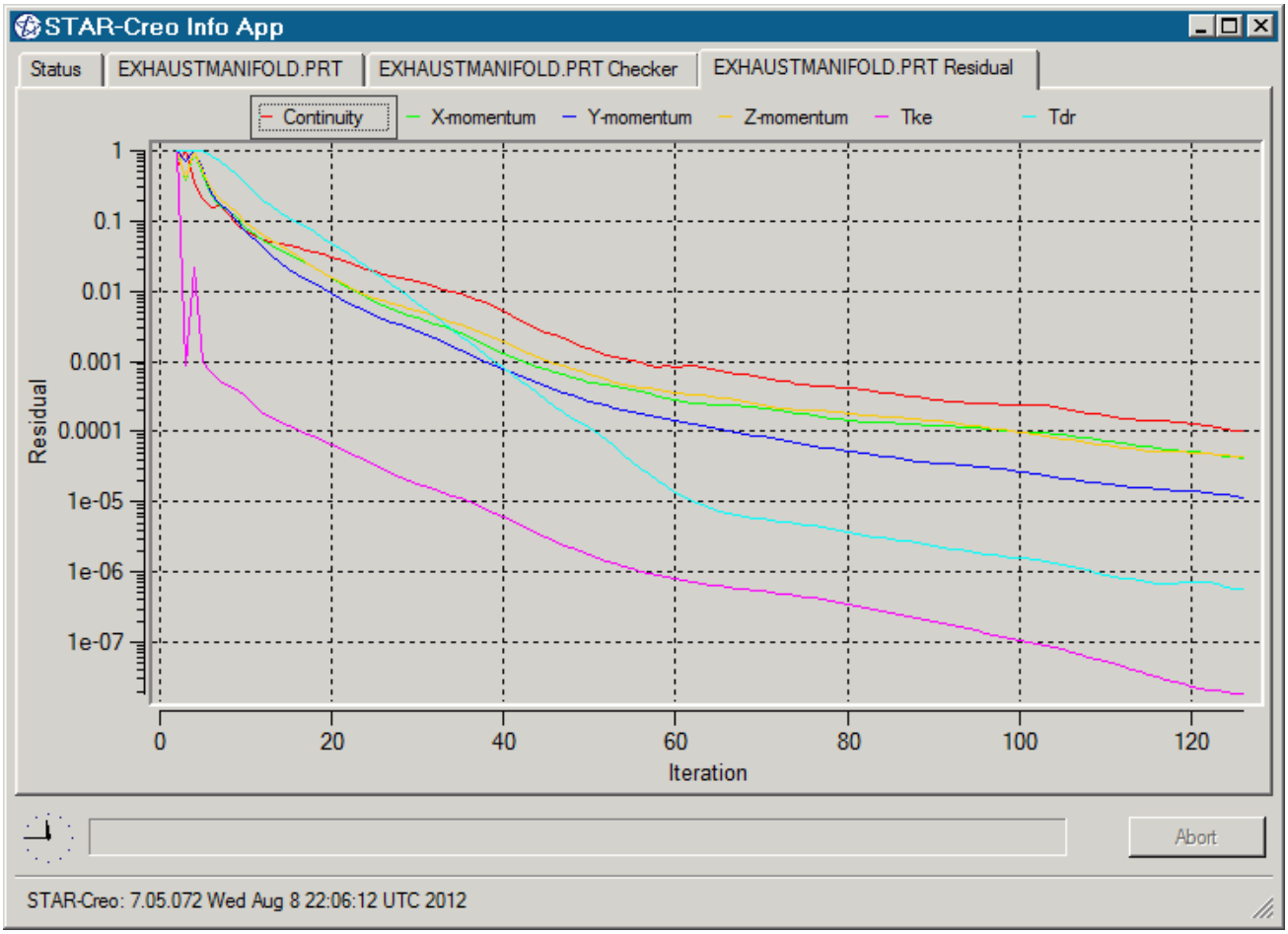
The *Model Checker* tab displays information from the live model checker, where the case setup is actively monitored for any problems. See Checking the Model for Errors.

The following example reports that the case has no inlet boundary defined.



The Residuals Tab

The *Residuals* tab appears when you launch the solver. Use the residuals to monitor the convergence of the solution being calculated.



Typically, a drop in three to five orders of magnitude in the residuals indicates that the solution has converged to an acceptable level. The residuals that are displayed in this tab also appear in STAR-CCM+ under the **Plots** node.

You can zoom in to the residuals by clicking and holding the left mouse button in the graph area and drawing a box around the area you want to view. To zoom out, right-click in the graph area. You can hide individual residuals from the *Residuals* tab by clicking their label at the top of the plot window.

Modifying the config.xml File

STAR-CCM+ requires a configuration file to enable it to find the appropriate CD-adapco components to run correctly. This configuration file is created during installation. An explanation of the file and settings within it is given here.

The configuration file is called `config.xml` and STAR-Creo searches for it in the location given by the `STARCADSERIES` environment variable.

Typical `config.xml` content is shown below. In this example STAR-CCM+ version 12.06.### is shown.

```
<CAD>
  <STAR_HOME>C:\Program Files\CD-adapco\12.06.###\STAR-CCM+12.06.###
\star</STAR_HOME>
```

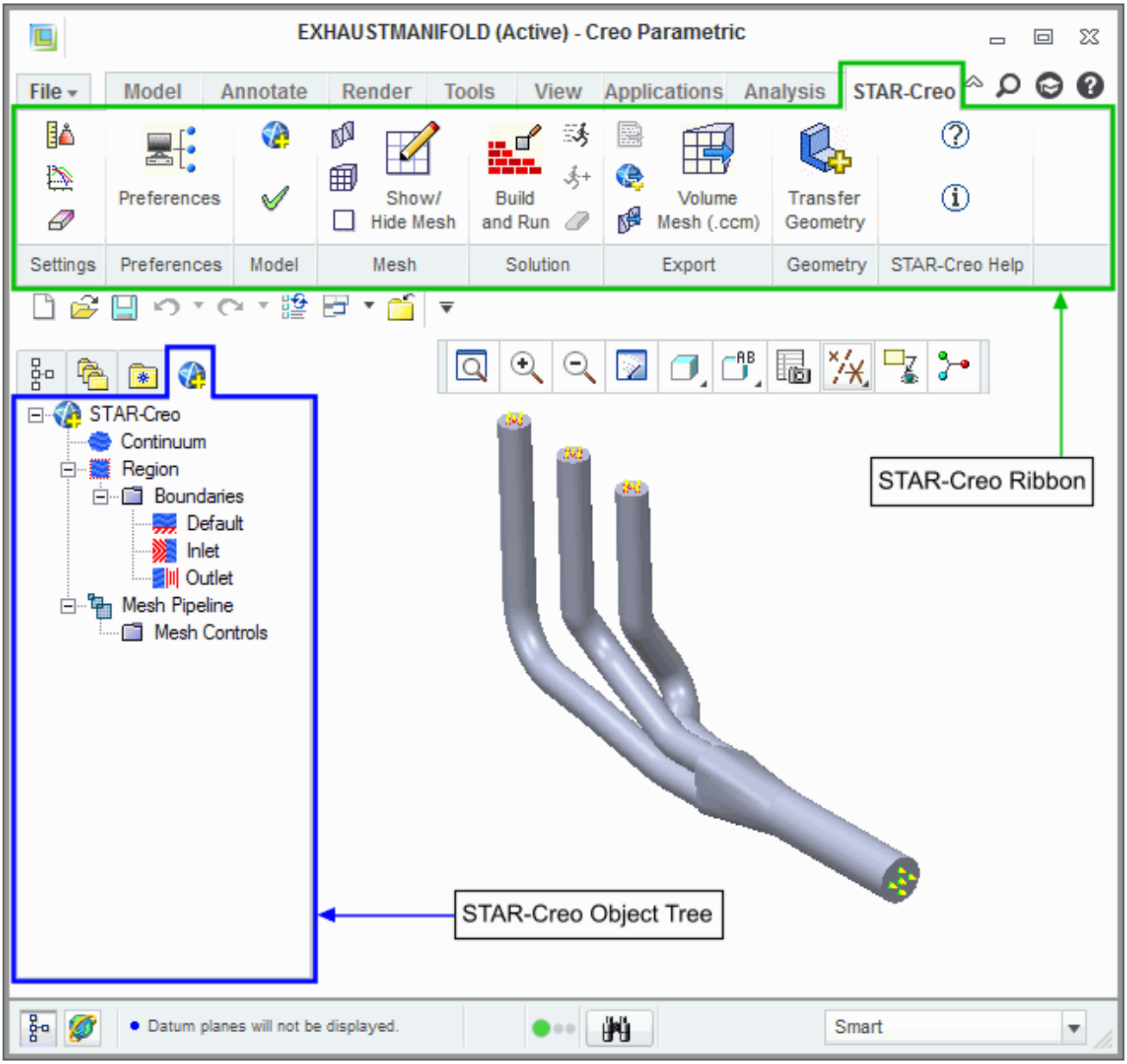
```
  <SOLVER>C:\Program Files\CD-adapco\12.06.###\STAR-CCM+12.06.###\star
\bin\starccm+.exe</SOLVER>
  <LOGFILE>starcad.log</LOGFILE>
  <WWHELP>C:\Program Files\CD-adapco\12.06.###\STAR-CCM+12.06.###
\win64\Intel11.1\wwhelp.exe</WWHELP>
  <MANDIR>C:\Program Files\CD-adapco\12.06.###\STAR-CCM+12.06.###\doc
\en\online</MANDIR>
  <VERBOSITY>Normal</VERBOSITY> <!-- Can be Verbose, Log or Normal.
Default Normal -->
</CAD>
```

You can edit the following lines:

- `<SOLVER>` provides the location of the STAR-CCM+ solver executable. STAR-CCM+ licensing options are set in the *Preferences* dialog in the STAR-Creo user interface. See Setting Up Parallel Processing and Licensing.
- `<LOGFILE>` is the log file that will automatically be created and saved when STAR-Creo is run. A path of your choice can be used.
- `<VERBOSITY>` is the verbosity of the output window in the STAR-Creo Info App. This can be set to one of the following:
  - `Normal`—This is the default setting and will display the least amount of information in the *Output* panel.
  - `Verbose`—This will display a moderate amount of information in the *Output* panel.
  - `Log`—This will display all available information in the *Output* panel.

Understanding the STAR-Creo Interface

The STAR-Creo user interface is embedded within the Creo environment, with the addition of the STAR-Creo object tree and ribbon.



The majority of the STAR-Creo operations are executed within the object tree and ribbon. Certain actions can also be carried out using multiple methods. A summary of the features available in each part of the STAR-Creo interface are detailed below.

Object Tree

The operations provided in the object tree include:

- Solution Controls
- Definition of Units System
- Continuum settings
- Region settings
- Add new boundary

- Boundary conditions
- Mesh Pipeline settings
- Add new mesh controls

Ribbon

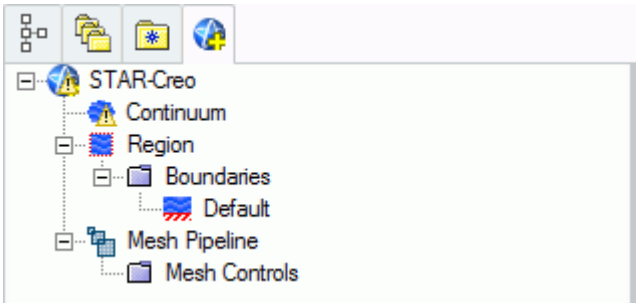
- Settings
  - Definition of Units System
  - Solution Controls
  - Clear All
- Preferences
- Mesh
  - Generate Surface Mesh
  - Generate Volume Mesh
  - Clear Generated Meshes
  - Show/hide Mesh
- Solution
  - Build and Run
  - Run Solver
  - Run For More Iterations
  - Clear Solution
- Model
  - Check Model for Errors
  - Transfer CAE Model
- Export
  - Surface Mesh (.dbs)
  - Volume Mesh (.ccm)
  - Engineering Report (.html, .csv)
  - Simulation (.sim)
- Geometry
  - Transfer Geometry

- Design Parameters
- STAR-Works Help
- About STAR-Works

The STAR-Creo Object Tree

STAR-Creo uses an object tree as the primary method for setting parameters.

The object tree consists of nodes, each one of which represents a particular component of the simulation. This approach to the user interface complements the method that is used by STAR-CCM+, allowing for an easy transition between the two software packages.



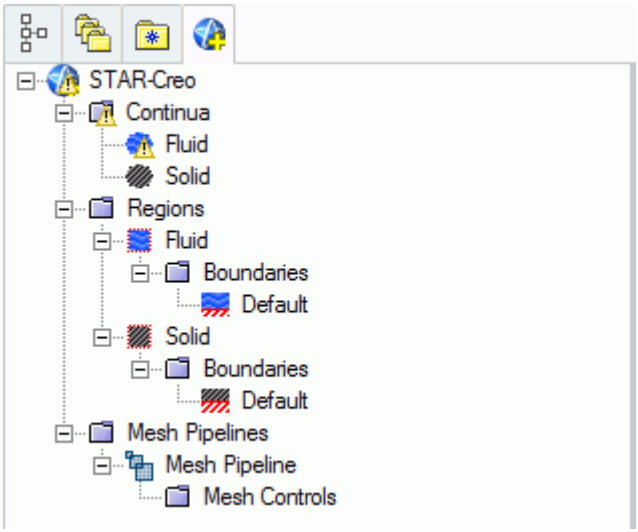
The object tree appears slightly different in assembly mode as it must accommodate multiple parts, but the principles of operation remain the same.

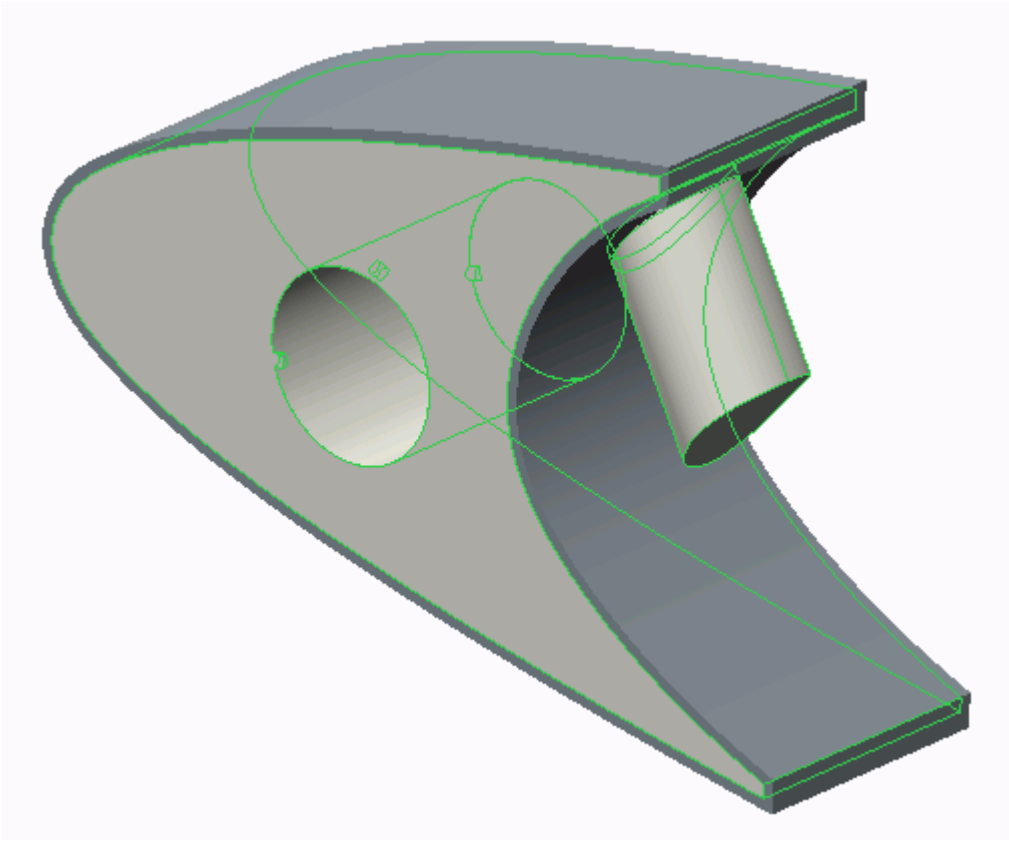
Nodes are arranged in an order that follows the recommended workflow for setting up a simulation in STAR-Creo. At each stage, you typically right-click on a particular node in the tree and choose the appropriate action from the pop-up menu.

Certain functionality within STAR-Creo applies to the simulation as a whole, rather than to individual objects. These operations are provided in the ribbon.

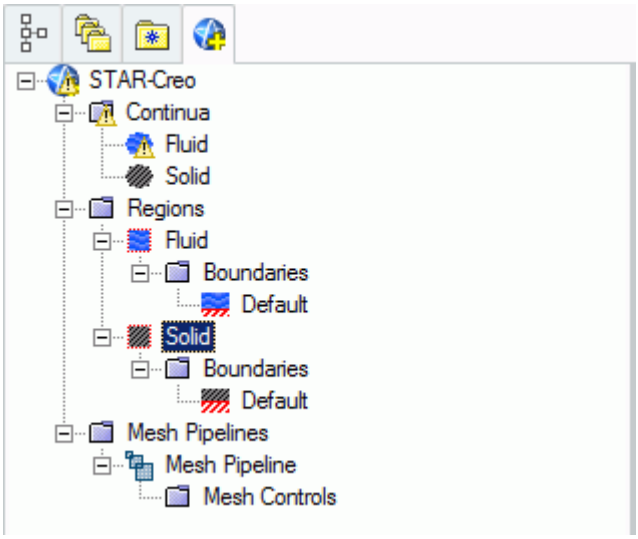
A complete list of all functions and their respective location is included in Understanding the STAR-Creo Interface.

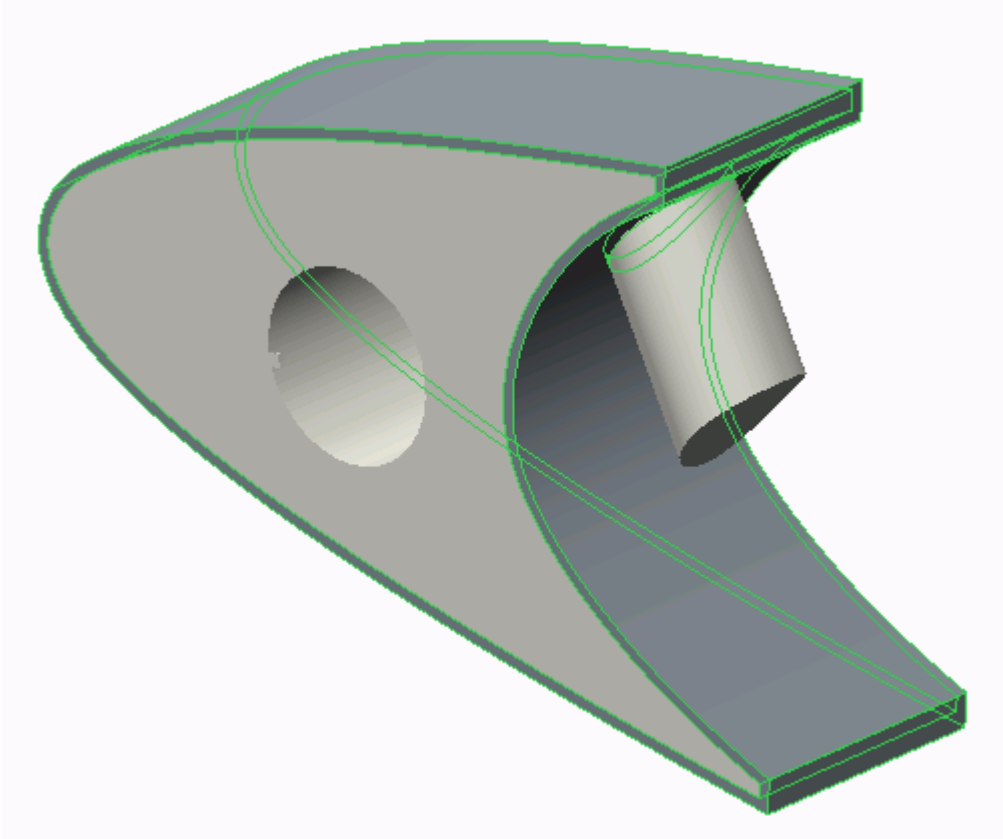
Clicking on the **Region** node within the object tree will highlight all the relevant faces on the geometry model for that particular region. If different parts are assigned to multiple regions, then clicking on a region will only highlight the part and faces associated with that specific region. For example, clicking on the **Fluid** node below will highlight all the faces associated with the Fluid region.





Similarly, clicking on the **Solid** node highlights all the faces associated with the Solid region.





Clicking on a boundary node in the STAR-Creo object tree highlights all the faces on the CAD geometry model that are associated with it. Note that clicking on the *Default* boundary node will not highlight any faces on the CAD geometry model. To stop highlighting the faces, you can either click on a blank space in the geometry window, or click on the root node in the STAR-Creo object tree.

Setting Up Parallel Processing and Licensing

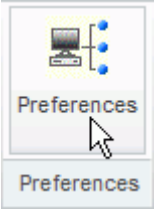
The licensing options and the options for setting up parallel processing are located in the *Preferences* dialog.

This dialog provides two sets of options: server preferences and server settings.

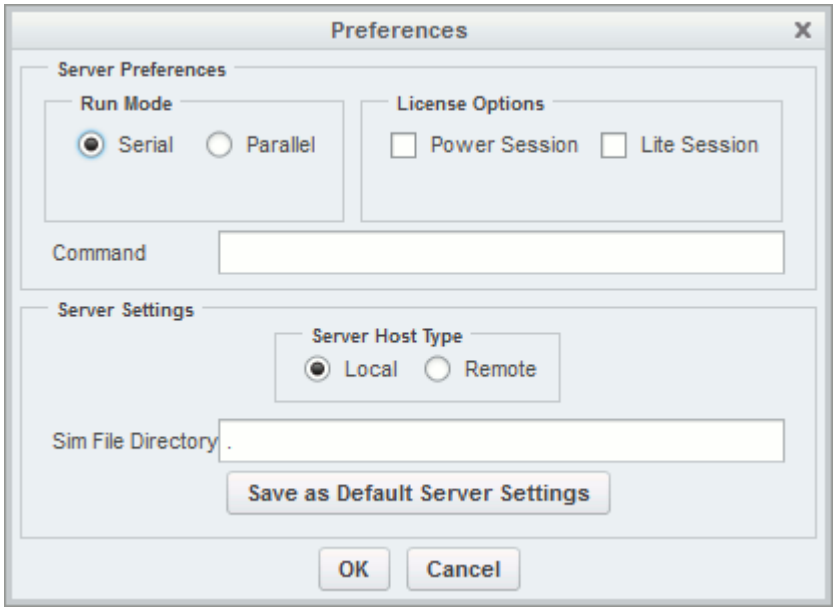
- Server preferences are applied whenever STAR-Creo starts a server.
- Server settings are saved in the CAD file and are applied only to that CAD file. Click **Save as Default Server Settings** to save the current server settings as the default settings to be applied to all CAD files.

To modify the run mode and licensing options:

1. Select **Preferences** from the ribbon.



The *Preferences* dialog appears.



2. Select the run mode:
   - For serial, set *Run Mode* to **Serial** (the default setting).
   - For parallel, set *Run Mode* to **Parallel** and *Compute Processes* to the desired number of cores.

     You can use the **up/down** buttons or you can type in the number of cores that you want to use.
   - For advanced licensing options, such as Power-on-Demand, enter the appropriate argument in the *Command* field.

For Power-on-Demand, enter:

```
-licpath 1999@flex.cd-adapco.com -podkey <pod_key>
```

Where:

- `1999@flex.cd-adapco.com` is the default CD-adapco Power-on-Demand server.
- `<pod_key>` is the Power-on-Demand license key.

For more information on Power-on-Demand licensing, see the section **User Guide** > **Using STAR-CCM+** > **STAR-CCM+ Licensing** > **Power-on-Demand Licensing**.

A full list of command-line arguments is available in the STAR-CCM+ User Guide at **User Guide** > **Using STAR-CCM+** > **Command-Line Reference**.

The *Command* field updates automatically to reflect the selections you make. Manual entries are not affected by these selections.

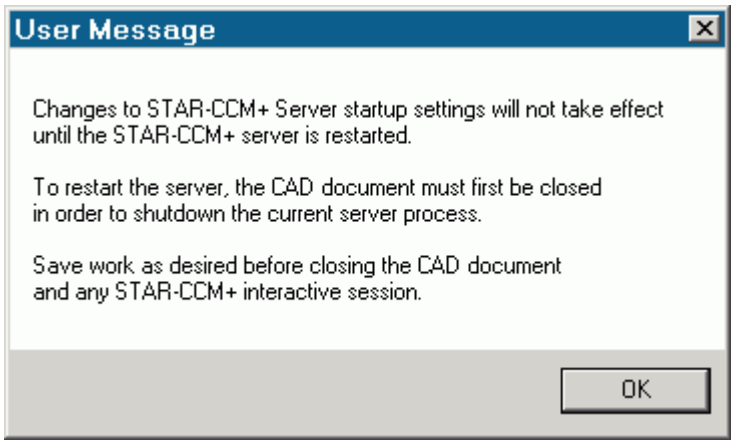
3. Select a licensing option:
   - To use a Power Session license, select **Power Session**.

     This license is design for parallel simulations, where you can use an unlimited number of cores in this session.
   - To use a STAR-CCM+ Lite license, select **Lite Session**.

     A limited set of STAR-CCM+ features are provided when you activate this option.

   If you select no licensing option, the default CAD client license is used.
4. Click **OK**.

   If the STAR-CCM+ server is running when you modify the preferences, a message box appears informing you that both STAR-Works and STAR-CCM+ must be closed for the changes to take effect.



5. Click **OK** to proceed.
6. Close the active STAR-Creo and STAR-CCM+ sessions.
7. Restart STAR-Creo.

Note: The STAR-CCM+ server only starts when it is required, such as during surface meshing or volume meshing. It does not start when you launch STAR-Creo.

If an option in the *Preferences* dialog conflicts with an option in the `config.xml` file, the option that is set in the *Preferences* dialog takes precedence. An example of such a situation is if the *Compute Processes* property is set to `4` in the *Preferences* panel, while it is set to `2` in the `config.xml` file; the simulation runs using four processors. See Modifying the config.xml File

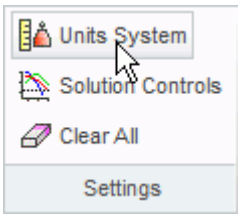
Note: IBM Platform MPI must be installed on your local machine for you to run in parallel. This program is included with the STAR-CCM+ package installer.

Changing the Units System

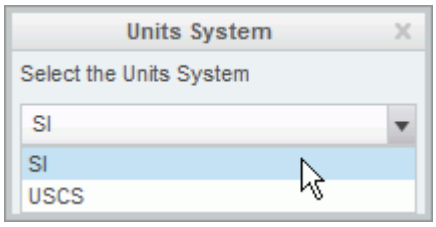
You can use either the SI (International Standard) or USCS (United States Customary System) units system in STAR-Creo.

To change the units system:

1. Bring up the *Units System* dialog:
   - Right-click the STAR-Creo node within the object tree and select **Set Units System**.
   - Alternatively, from the ribbon, select **Units System**.



   The *Units System* dialog appears.
2. Select the desired units system from the drop-down menu.



3. Click **OK**.

Automatic Unit Conversion in Dialog Panels

Changing the default units system converts existing values to the new units system. Ensure that you select the desired units system before you enter the value, otherwise you will need to re-enter it.

Note: Specific values that you have already modified the units on are not affected by a change of units system.

Transferring Units to STAR-CCM+

Units that are used in the CAD geometry file are also used in the STAR-CCM+ simulation. Less common units, such as those listed below, are not supported in STAR-CCM+:

- Micrometers
- Angstroms
- Nanometers
- Mils
- Microinches
- Footinches (this unit may be converted to inches in STAR-CCM+)
- Miles

Note: This list may not be exhaustive and is subject to change.

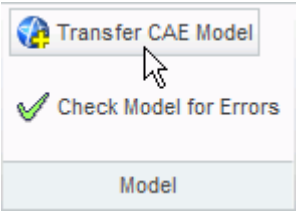
These units are converted to meters when the CAD geometry is imported into STAR-CCM+.

Transferring the CAE Model to STAR-CCM+

You can transfer your geometry and the simulation setup to STAR-CCM+ at any time from within the STAR-Creo environment. If you want to transfer only the geometry to STAR-CCM+, refer to Transferring Geometry to STAR-CCM+.

To transfer the geometry with the simulation set up:

1. From the STAR-Creo ribbon, click **Transfer CAE Model**.



The version of STAR-CCM+ that is affiliated with STAR-Creo launches.

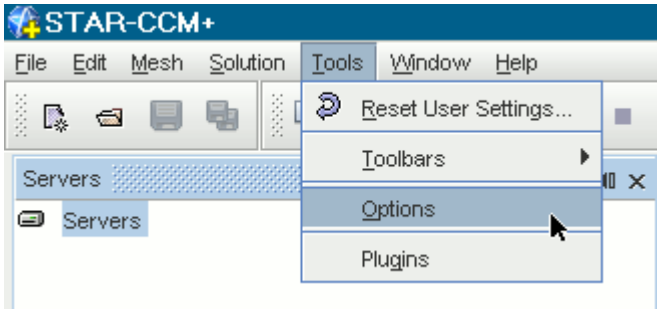
Note: A surface mesh is required in order to visualize the geometry in STAR-CCM+. If you have not generated a mesh before clicking **Transfer CAE Model**, the geometry is tessellated with a medium density.

Using Creo Mouse Controls in STAR-CCM+

STAR-CCM+ offers the ability to set the mouse interactor style to match that of Creo; this could make geometry manipulation in STAR-CCM+ more natural if you are used to working with Creo.

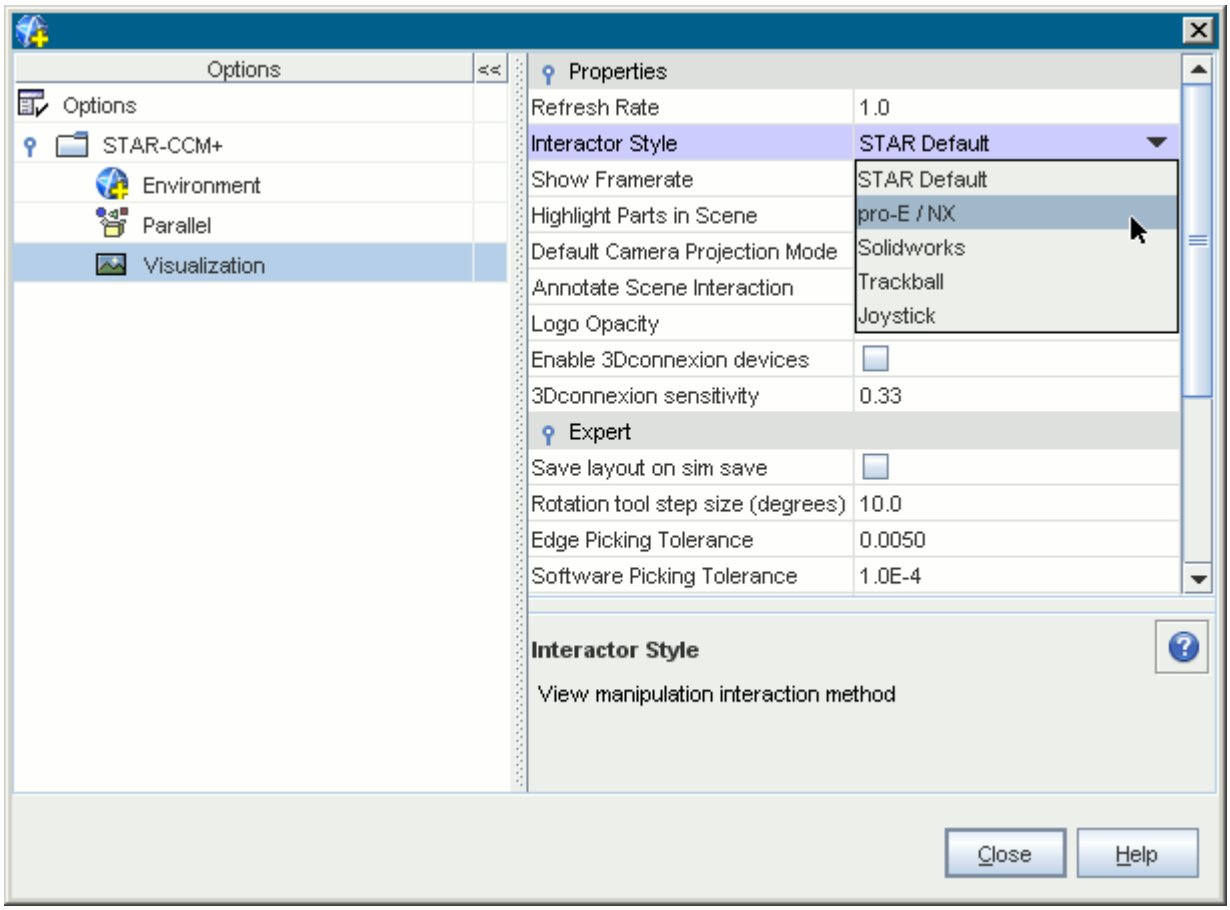
To change the mouse interactor style:

1. Launch STAR-CCM+.
2. Click **Tools** and select **Options** from the menu.



The *Options* dialog appears.

3. Expand the STAR-CCM+ node and select **Visualization**.
4. In the *Properties* panel, set *Interactor Style* to**pro-E / NX** using the drop-down menu.



5. Click **Close**.

STAR-CCM+ will now use this mouse interactor style each time it launches.

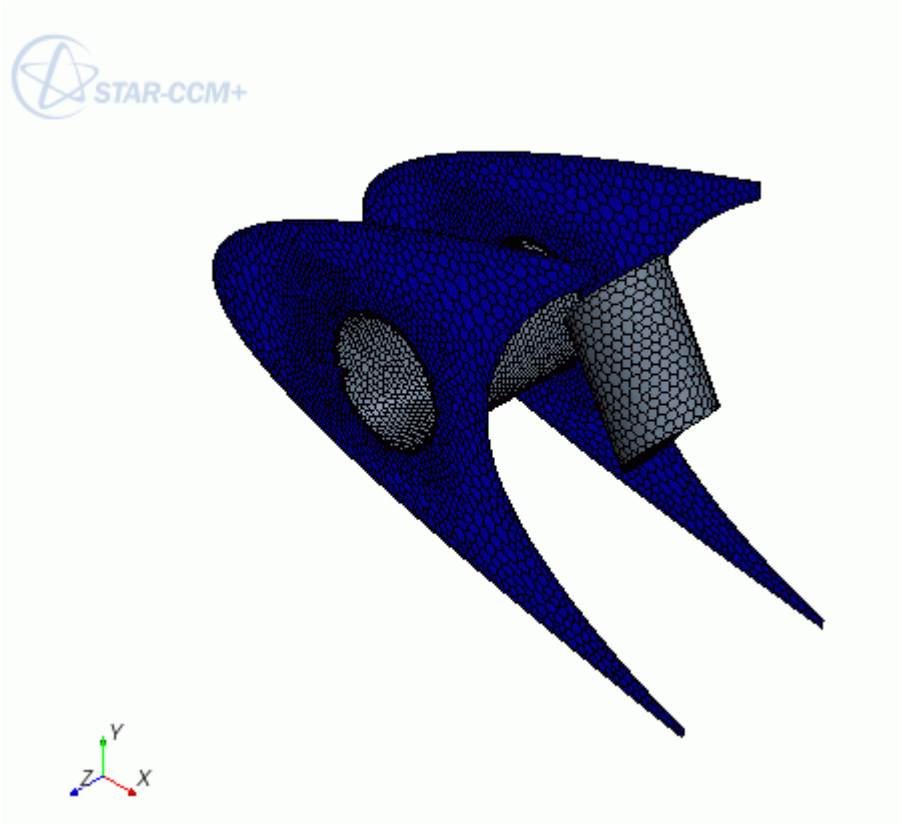
Editing STAR-Creo Settings While the STAR-CCM+ Client is Running

If you make a change in STAR-Creo while the STAR-CCM+ client is running, such as adding or deleting a boundary, you may need to update the volume mesh.

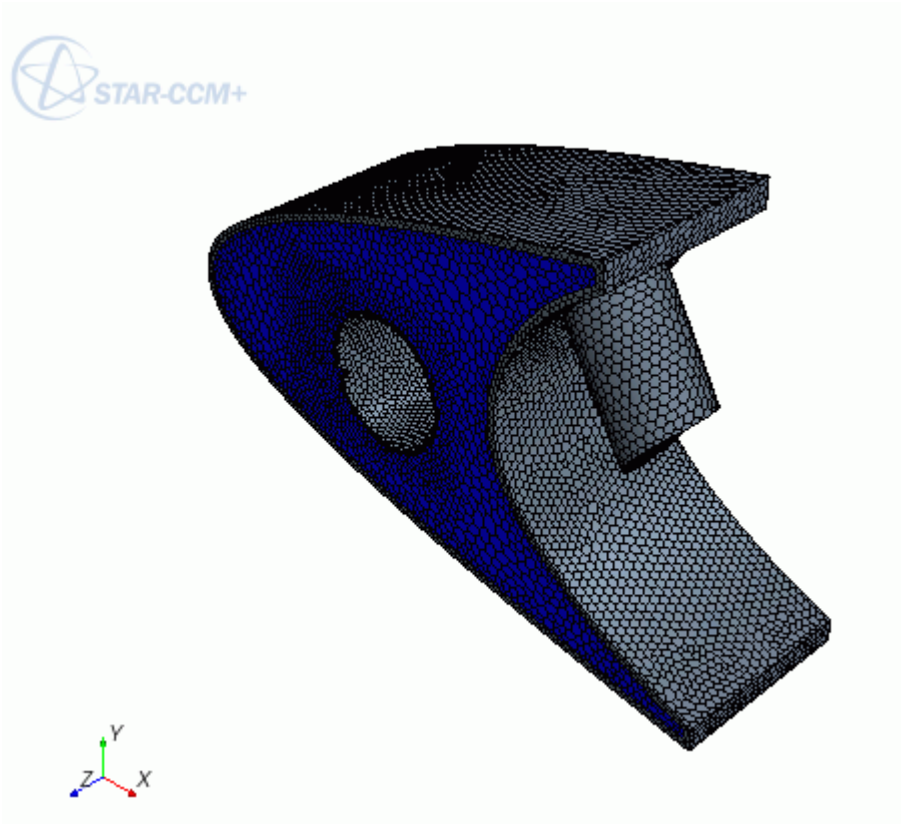
Scenes in the STAR-CCM+ session will not display the model correctly unless you keep the volume mesh up to date. A list of changes that require you to re-mesh the geometry is provided below:

- For boundaries and regions: changing the type, adding or deleting
- For mesh pipelines and mesh controls: editing, adding or deleting

When adding or removing regions and boundaries in STAR-Creo, the scenes in STAR-CCM+ may be missing some parts, as shown below, even after re-meshing the geometry.



To update the scene, you need to manually add the modified regions and boundaries to the scene in the STAR-CCM+ client. You can do this by dragging and dropping the region or part from the STAR-CCM+ object tree, or by adding the region or part using the Part Selector. An alternative method would be to create a new scene, as this will automatically pick up the new regions and boundaries added to the simulation.



Using STAR-Creo with Multiple Installed Versions of Creo

If you have multiple versions of Creo installed on your machine, STAR-Creo is installed to work with the latest version.

You can manually integrate STAR-Creo with an older supported version of Creo:

1. In the installation directory of the older version of Creo, navigate to the folder that contains the `config.pro` file.

   The default location is:

   ```
   C:\Program Files\PTC\Creo X.0\Common Files\XXXX\text
   ```

2. Locate and edit the `config.pro` file using a text editor of your choice.

   You require administrator privileges to edit the `config.pro` file.

3. Enter the following line at the end of the file:

   ```
   protkdat C:\Program Files\CD-adapco\12.06.###\STAR-CAD12.06.###
   \STAR-Creo\protk.dat
   ```

   This line contains the full path to the `protk.dat` file in the STAR-Creo installation directory. Make sure to enter the correct full version of STAR-Creo.

4. Save the file and exit the text editor.

Disabling STAR-Creo

If for any reason you would like to temporarily disable STAR-Creo without having to uninstall it, do the following:

1. Go to the text folder in the Creo installation directory.

   The typical path is `C:\Program Files\PTC\Creo X.0\Common Files\XXXX\text`.
2. Locate and edit the `config.pro` file using a text editor of your choice.
3. Comment out the following line with an "`!`":

   ```
   protkdat C:\Program Files\CD-adapco\12.06.###\STAR-CAD12.06.###\STAR-Creo\protk.dat
   ```

   so that it looks like this:

   ```
   !protkdat C:\Program Files\CD-adapco\12.06.###\STAR-CAD12.06.###\STAR-Creo\protk.dat12.06.###
   ```

Creo will now launch without STAR-Creo. To enable STAR-Creo again, undo the changes listed above.

Common Functions

The following section details common operations used within the STAR-Creo environment.

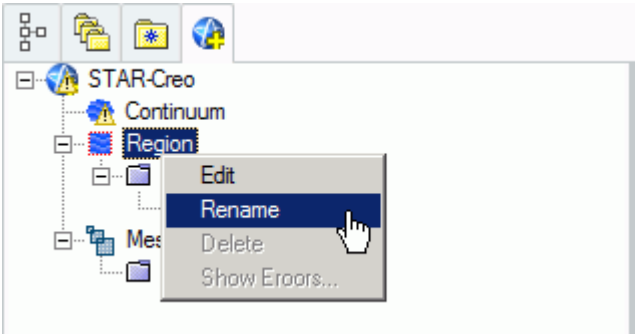
Contents:

Renaming an Object
Deleting an Object
Editing An Object
Assigning Surfaces and Parts to Objects in STAR-Creo
Removing Surfaces and Parts from Objects in STAR-Creo

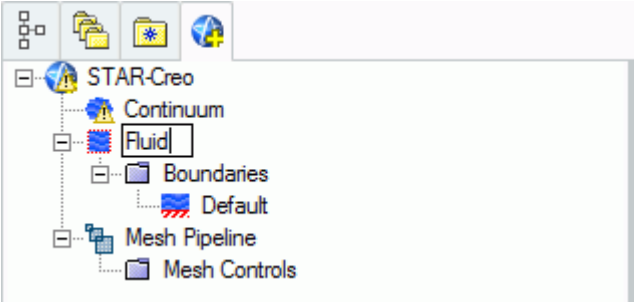
Renaming an Object

To rename an object:

1. Right-click the object node within the object tree and select **Rename**.



2. Enter the name.



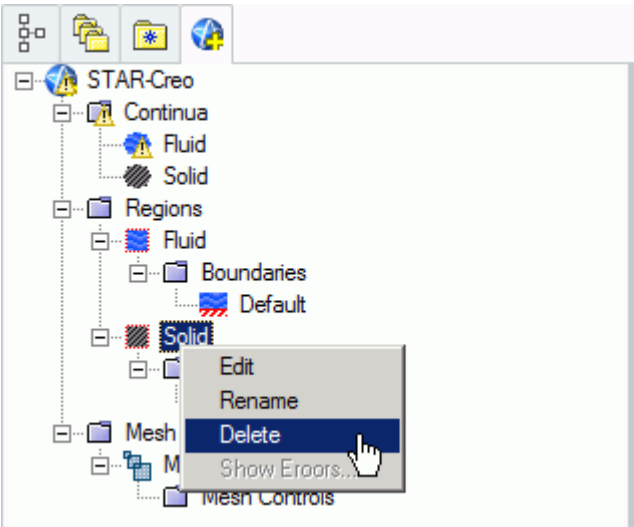
3. Click on a blank space in the object tree.

Note: You can also assign entity names at the point where you create them, for example, when adding boundaries, continua, regions, mesh pipelines and mesh controls. When adding a new object, locate the **Name** text box in the settings dialog and enter the required name.

Deleting an Object

To delete an object:

1. Right-click the object node within the object tree and select **Delete**.



To delete multiple entities:

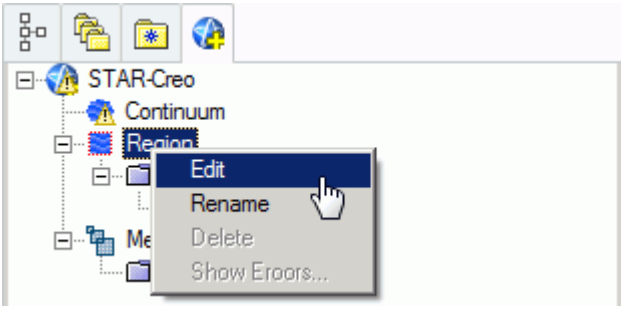
2. Select the desired nodes within the object tree (by holding down the `<Ctrl>` key), and right-click on any one of the selected nodes, then select **Delete** from the pop-up menu.

Note: Some combinations of selections cannot be deleted together using this method. In these cases, no pop-up menu appears.

Editing An Object

To edit an object:

1. Right-click on the desired node in the object tree and select **Edit...**.



   The settings dialog for the object appears.
2. Make the changes that you want.
3. Click **OK**.

Assigning Surfaces and Parts to Objects in STAR-Creo

Regions, boundaries, mesh pipelines, and mesh controls are all represented as objects on the STAR-Creo object tree. You can assign parts and surfaces to these objects when their property dialog is open.

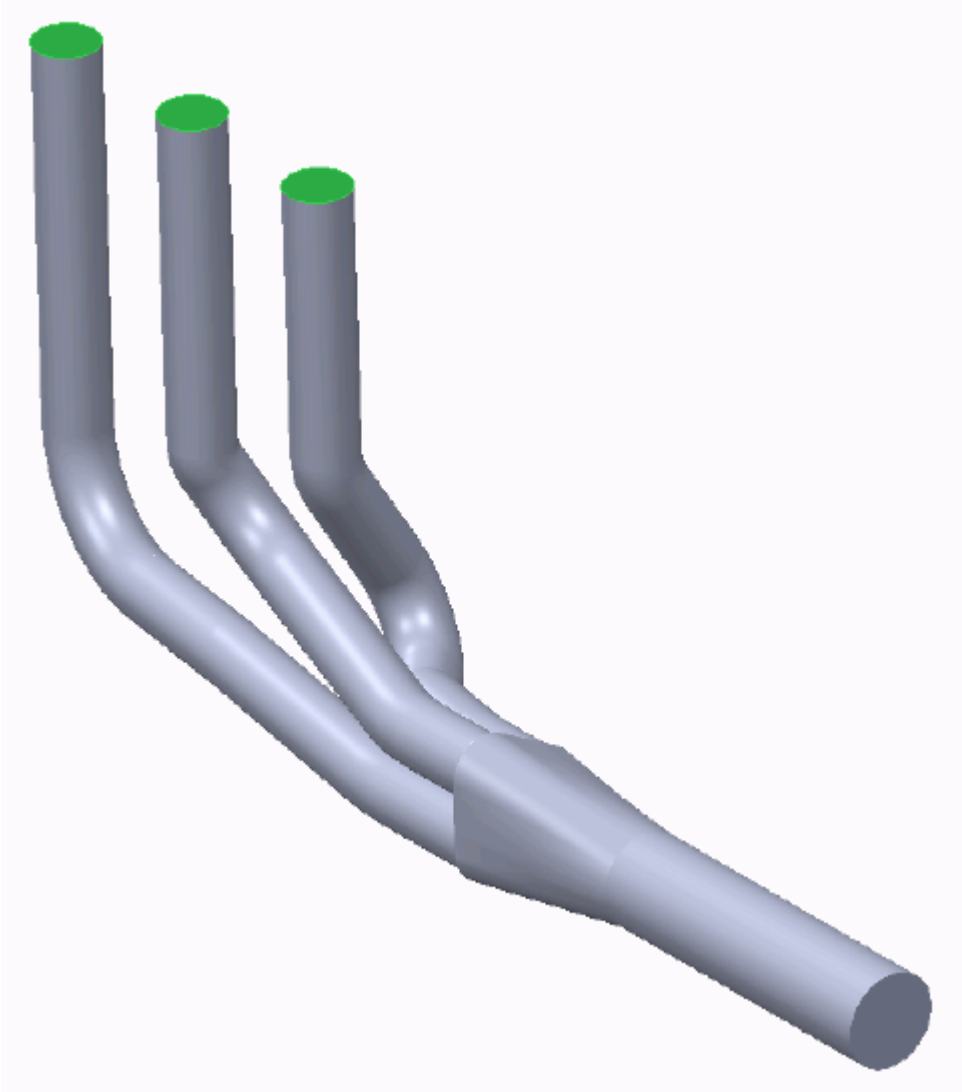
Assigning Surfaces to Objects

To assign a surface to an object (when its property dialog is open):

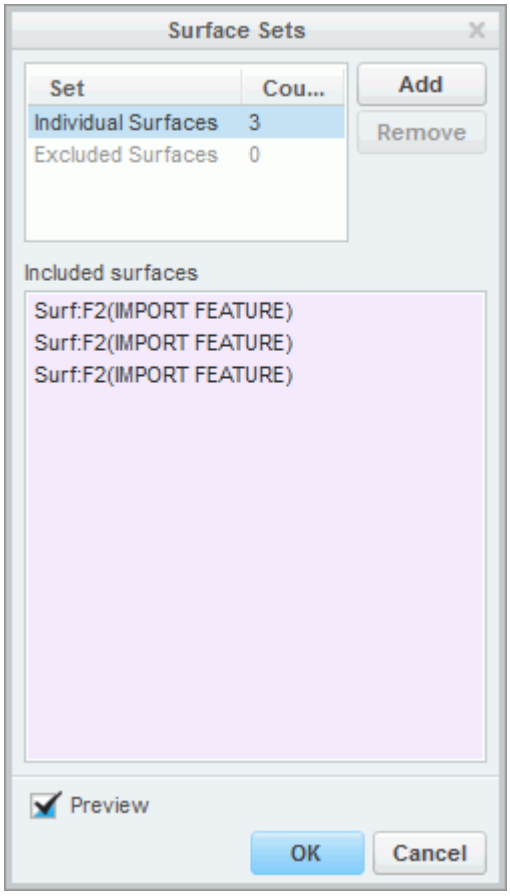
1. Click (**Select Face**).

   This button activates the Creo surface selection mechanism and allows you to select the faces on the CAD geometry.
2. On the CAD geometry, click the surfaces that you want to assign to the object. You can select multiple faces by holding down the `<Ctrl>` key whilst clicking the surfaces.

   The selected surfaces are highlighted on the CAD geometry.

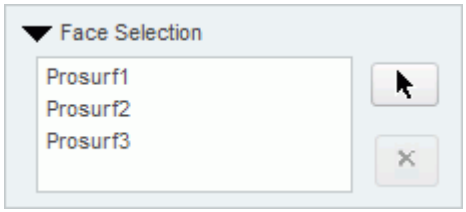


The selected surfaces are added to the *Included Surfaces* box in the *Surface Sets* dialog.



3. Click **OK** to close the *Surface Sets* dialog.

   The surfaces are then added to the *Face Selection* box of the active dialog.



Parts that are assigned to other regions are automatically hidden on the CAD geometry.

Assigning Parts to Objects

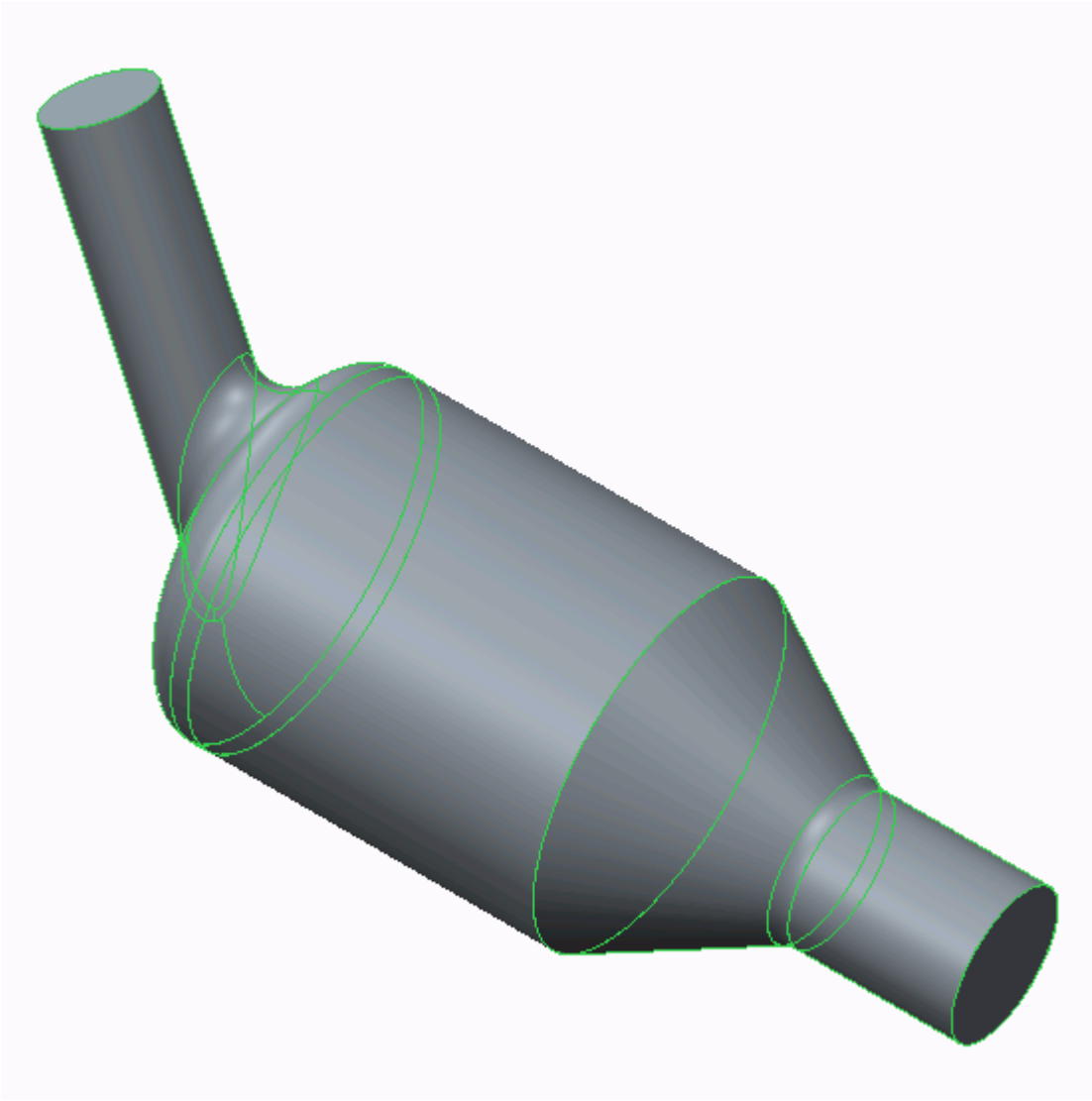
To assign a part to an object (when its property dialog is open):

4. Click (**Select part**).

   This button activates the Creo part selection mechanism and allows you to select parts on the CAD geometry.

5. On the CAD geometry, click the part that you want to assign to the object. You can select multiple parts by holding down the `<Ctrl>` key whilst clicking the parts.

   The selected parts are highlighted on the CAD geometry.



The selected parts are added to memory in the Creo part selection mechanism.

Parts that are assigned to other regions are automatically hidden on the CAD geometry.

Removing Surfaces and Parts from Objects in STAR-Creo

Regions, boundaries, mesh pipelines, and mesh controls are all represented as objects on the STAR-Creo object tree. You can remove parts and surfaces from these objects when their property dialog is open.

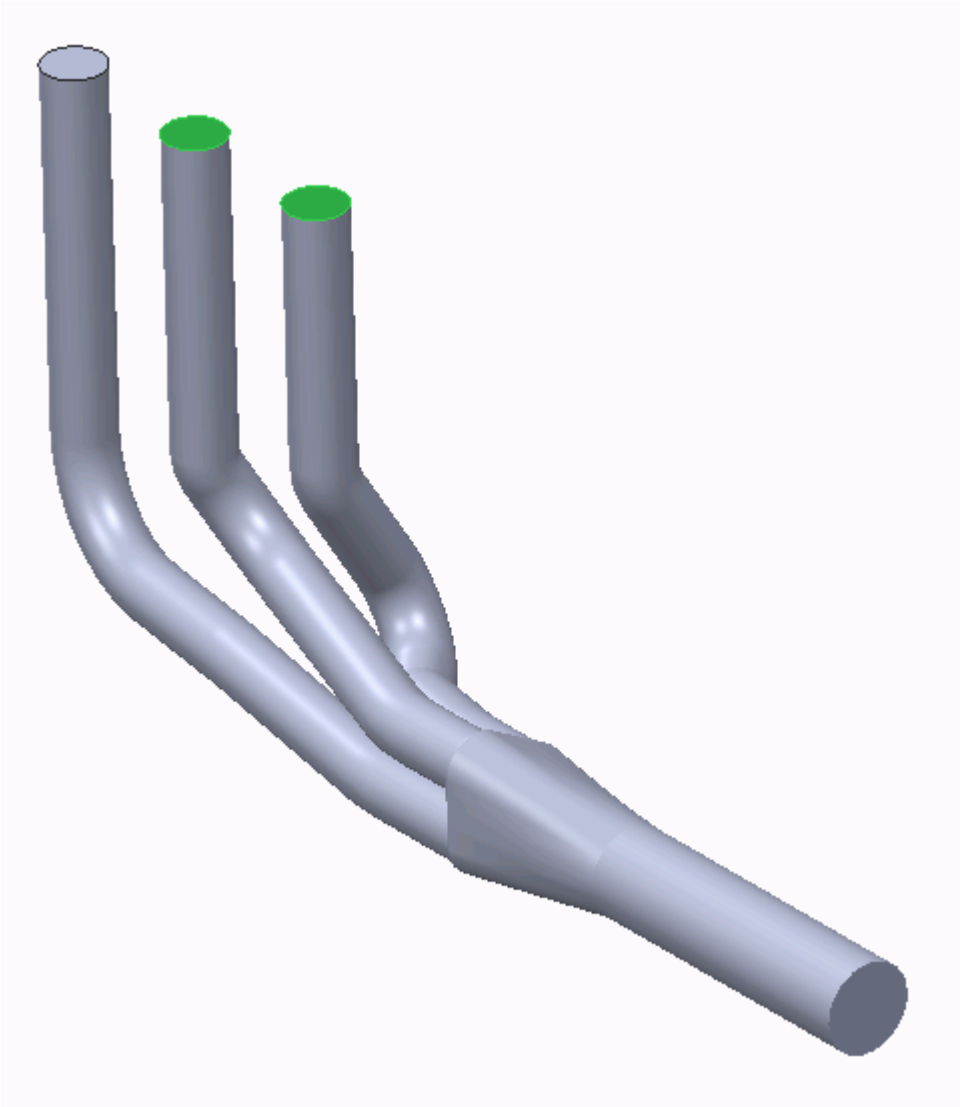
A surface that is removed from a boundary is returned to the default boundary. Similarly, removing a part from a region returns it to the default region.

Removing Surfaces from Objects

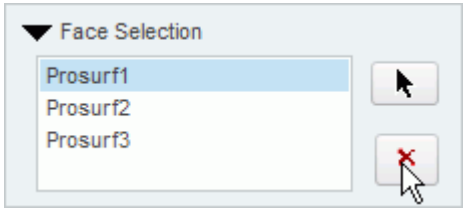
To remove a surface from an object (when its property dialog is open):

1. In the *Face Selection* box, select the surface you want to delete.

The selected surface on the CAD geometry is cleared.



2. Click (**Remove a face**).



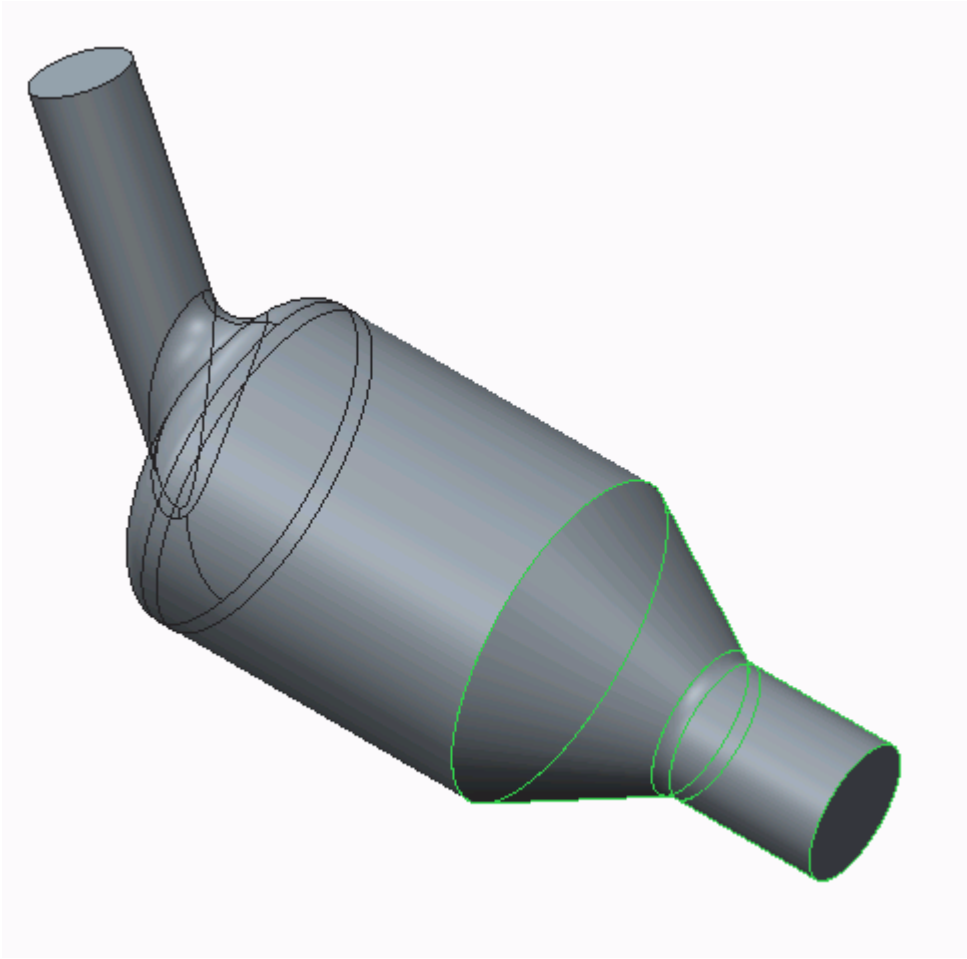
The surface is removed from the *Face Selection* box. You can select multiple surfaces for deletion by holding down the <`Ctrl`> key, and remove them in one operation.

Removing Parts from Objects

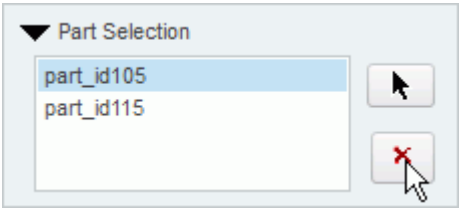
To remove a part from an object (when its property dialog is open):

3. In the *Part Selection* box, select the part you want to remove.

The selected part on the CAD geometry is highlighted in black.



4. Click (**Remove a part**).



The part is removed from the *Part Selection* box. You can select multiple parts for deletion by holding down the <`Ctrl`> key, and remove them in one operation.

Running STAR-Creo and STAR-CCM+ on Different Machines

You can run STAR-Creo and STAR-CCM+ on two different machines.

STAR-CCM+ operates on a client-server architecture, where STAR-CCM+ is the server and STAR-Creo is the client. A benefit of client-server architectures is that the client and the server do not have to be running on the same machine. Typically, CAD packages run on Windows operating systems. You can choose to run STAR-CCM+ on any remote machine with a supported or certified platform, including Linux platforms.

Contents:

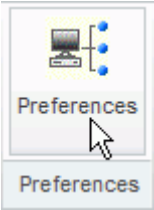
Setting Up STAR-CCM+ to Run on a Remote Machine with STAR-Creo on the Local Machine

Setting Up STAR-CCM+ to Run on a Remote Machine with STAR-Creo on the Local Machine

STAR-Creo and STAR-CCM+ can run on two different machines.

To set up STAR-CCM+ to run on a remote machine:

1. From the STAR-Creo ribbon, click **Preferences**.



2. In the *Preferences* dialog, set *Server Host Type* to **Remote**.

   The options for the remote server appear.
3. Set the remaining server settings to reflect your situation. See Server Settings Panel Reference.
4. If you want to store these settings for the next time you launch STAR-Creo, click **Save as Default Settings**.
5. Click **OK**.

Contents:

Server Settings Panel Reference

Server Settings Panel Reference

The Server Settings panel that is located in the Preferences dialog allow you to run STAR-Creo on a remote machine.

Property	Description
Server Host Type	Specifies whether to run STAR-CCM+ on the local machine or on a remote machine.

Property	Description	
	Local	STAR-CCM+ runs on the local machine.
	Remote	STAR-CCM+ runs on a remote machine
SSH Command	Specifies the absolute path to the SSH installation directory.	
Server Host	Specifies the name of the remote host where STAR-CCM+ is installed. This option can contain the IP address, name, or full name of the host.	
Sim File Directory	Specifies the absolute path to the directory where you want to store the simulation files on the remote machine. When in **Local** mode, the simulation files are saved in the same directory as the geometry file. You can specify a custom directory.	
Server Executable	Specifies the absolute path to the STAR-CCM+ executable on the remote machine.	

Transferring Geometry to STAR-CCM+

With STAR-Creo, you can transfer your CAD geometry directly to STAR-CCM+ from within the Creo environment.

The Transfer Geometry feature imports your geometry into STAR-CCM+. This function provides an alternative method to the CAD translator add-ons for importing geometry. This method of transferring geometry can provide a faster workflow when performing a design analysis.

To transfer geometry to STAR-CCM+:

1. From the STAR-Creo ribbon, click **Transfer Geometry**.

   STAR-CCM+ launches and imports the CAD geometry. You are now at the same stage in the workflow as if you imported the surface mesh directly into STAR-CCM+ using the CAD translators (**File** > **Import** > **Import Surface Mesh...**).

2. Set up your case in STAR-CCM+.

Caution: Do not click **Transfer CAE Model** if you have previously used the Transfer Geometry method and set up your case in STAR-CCM+. This action results in the loss of your settings in STAR-CCM+.

Contents:

Saving Simulation Files
Transferring Design Data to STAR-CCM+
Modifying Geometry from Within STAR-CCM+
Controlling How the Geometry is Transferred to STAR-CCM+
Geometry Transfer Settings Dialog

Saving Simulation Files

After transferring the geometry to STAR-CCM+, you can save the simulation file and retain the associativity with STAR-Creo.

To save your simulation files and keep the associativity:

- Save your CAD geometry in STAR-Creo.

  Note: You can also safely rename and move the file using this method.

or

- Save your simulation file in STAR-CCM+.

Note: Do not rename the simulation file when using this method, as it will lose the associativity with the Creo file.

Resuming a Simulation

To resume a simulation after closing STAR-Creo and STAR-CCM+:

1. Launch STAR-Creo.
2. Open the CAD geometry in STAR-Creo.
3. Click **Transfer Geometry**.

Transferring Design Data to STAR-CCM+

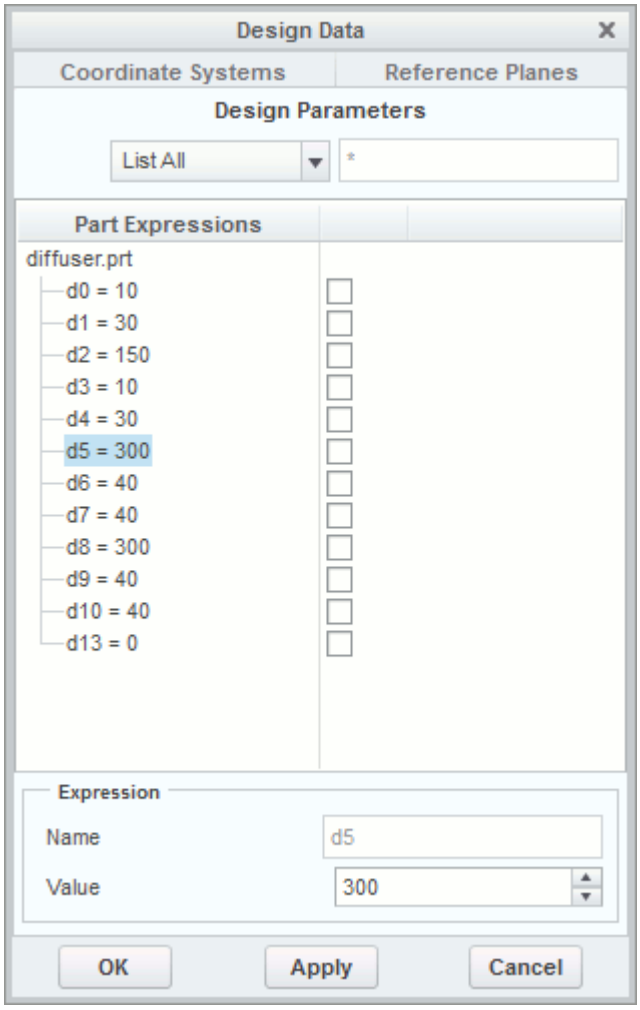
Design data consists of design parameters, coordinate systems, and reference planes that you can transfer to STAR-CCM+ along with the geometry.

Design parameters control the feature dimensions, for example extrusion depth, and are useful for modifying the geometry in STAR-CCM+ after running the analysis. You can use reference planes in scalar scenes and reports in STAR-CCM+. You can use coordinate systems to define motion in the STAR-CCM+ simulation.

To transfer design data to STAR-CCM+:

1. From the STAR-Creo ribbon, click **Design Data**.

The *Design Data* dialog appears.



2. Select the design data to transfer:
   - **Design Parameters**—Click the *Design Parameters* tab and select the design parameters that you want to transfer from the list. You can select multiple items at once.

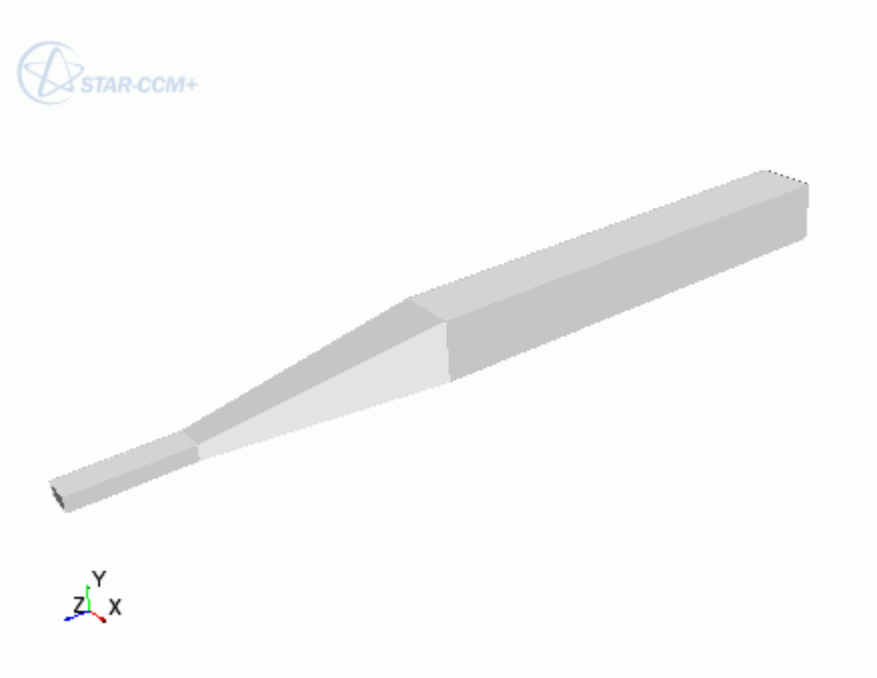
     Note: STAR-Creo only recognizes native CAD features as design parameters. Only recognized design parameters are displayed in the list.
   - **Coordinate Systems**—Click the *Coordinate Systems* tab and select the coordinate systems that you want to transfer from the list. You can select multiple items at once.
   - **Reference Planes**—Click the *Reference Planes* tab and select the planes that you want to transfer from the list. You can select multiple items at once.
3. In each tab, click the checkbox next to the selected item to activate it.
4. Click **OK**.
5. Transfer the geometry to STAR-CCM+. See Transferring Geometry to STAR-CCM+.

The geometry, along with the specified design data, is transferred to STAR-CCM+. Transferred design parameters are placed in the **Geometry** > **CAD Client Models** > **[geometry_name]** > **Design Parameters** node. Transferred planes are placed in the **Derived Parts** node in STAR-CCM+. Transferred coordinates systems are placed in the **Tools** > **Coordinate Systems** > **Laboratory** > **Local Coordinate Systems** node in STAR-CCM+.

Modifying Geometry from Within STAR-CCM+

Once you transfer a design parameter to STAR-CCM+, you can modify the geometry from within STAR-CCM+ with respect to that parameter. This feature is useful for quickly modifying a geometry after performing an analysis in STAR-CCM+ without needing to return to the original CAD software.

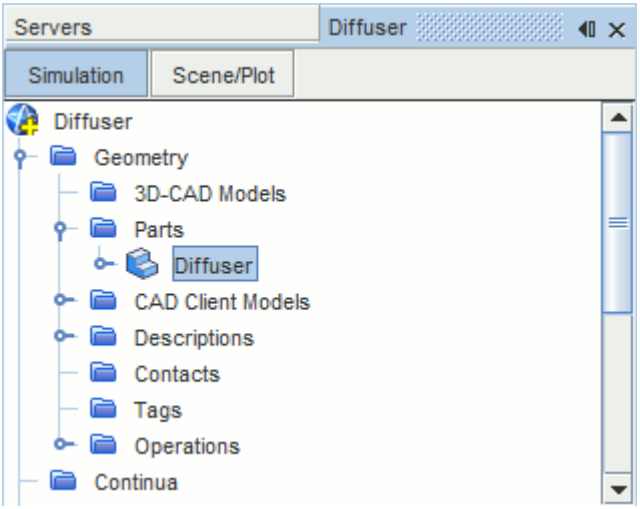
In the following example, the geometry of a diffuser is modified.



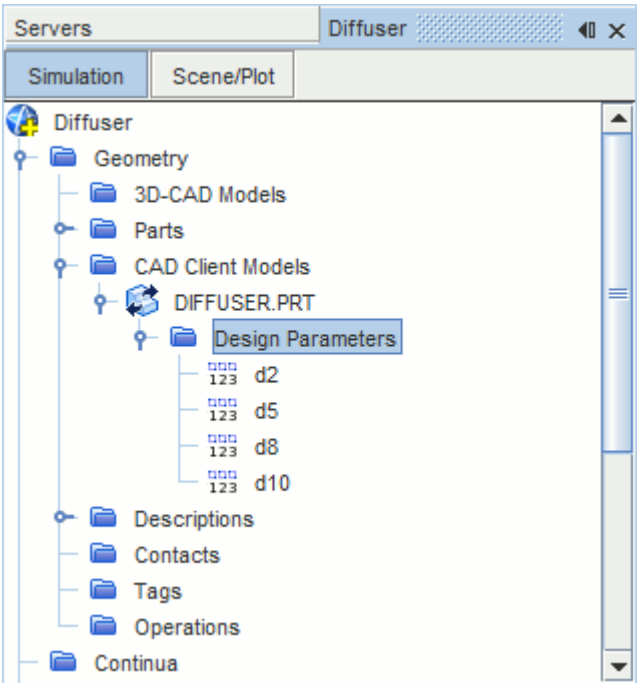
To modify a geometry in STAR-CCM+:

1. In STAR-Creo, select the design parameters that you want to transfer to STAR-CCM+.
2. Transfer the geometry to STAR-CCM+. See Transferring Geometry to STAR-CCM+.

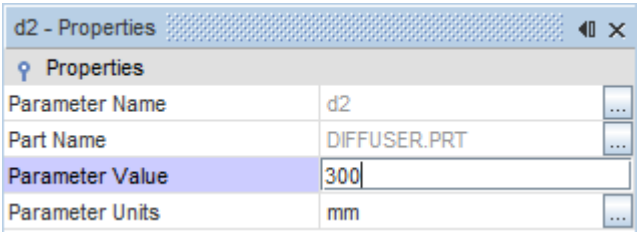
   Transferred geometries are located under the **Geometry** > **Parts** node.



Transferred design parameters are located under the **Geometry** > **CAD Client Models** > **[part_name]** > **Design Parameters** node.

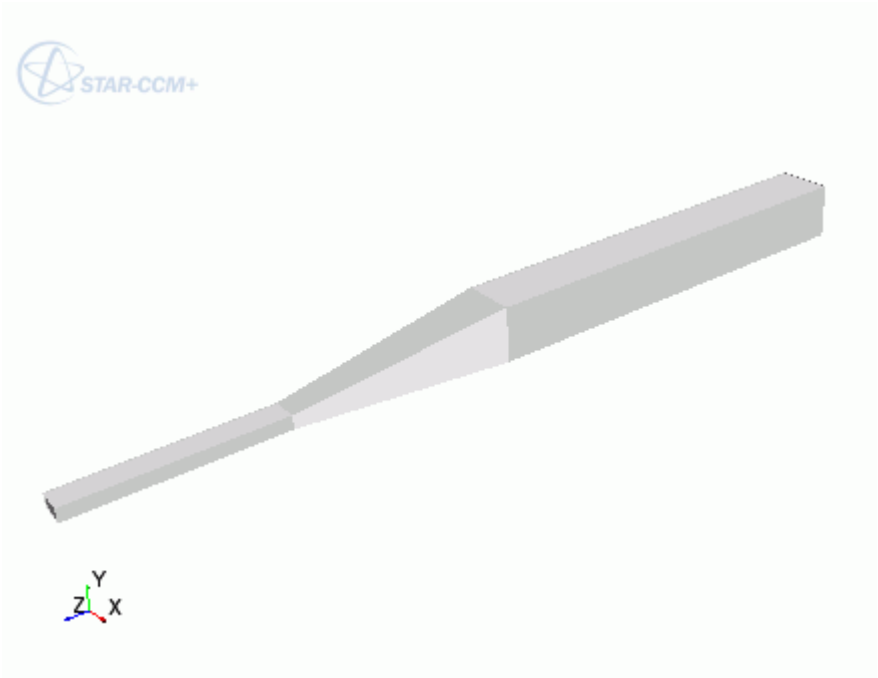


3. In STAR-Creo, save the geometry file. The simulation file is also saved.
4. In STAR-CCM+, select the design parameter that you want to modify and set *Parameter Value* to the desire value.



5. Update the part in STAR-CCM+:
   - To update the part and save the updated parameters in the CAD file, right-click **Geometry** > **CAD Client Models** > **[part_name]** and select **Update Model and Save the CAD File**.
   - To update the part without updating the parameters in the CAD file, right-click **Geometry** > **CAD Client Models** > **[part_name]** and select **Update Model**.

   The geometry updates according to the design parameter changes.



6. Return to Creo. The changes to the geometry are transferred back to the native CAD geometry.

This workflow maintains an active link between STAR-Creo and STAR-CCM+. When working with Geometry Transfer, you can modify the geometry in either package and the geometry in the relevant package is updated accordingly.

Updating the Geometry in STAR-CCM+ with Creo Closed

You can close Creo and continue working with the simulation in STAR-CCM+. Doing so, however, breaks the link between the two software packages. Any further updates that you make to the geometry in STAR-CCM+ are not transferred back to the geometry in Creo. Instead, these changes are stored in memory, which allows you to include the design parameters in your optimization scripts.

This workflow also allows you to release the Creo and STAR-Creo licenses when performing tasks in STAR-CCM+. In this case, the Creo and STAR-Creo licenses are temporarily checked out only when the updates to the geometry are stored in memory.

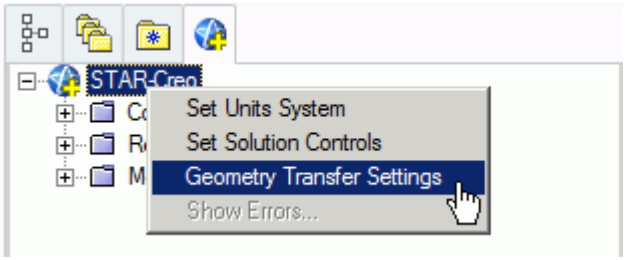
Transferring the geometry from STAR-Creo will overwrite the geometry in STAR-CCM+.

Controlling How the Geometry is Transferred to STAR-CCM+

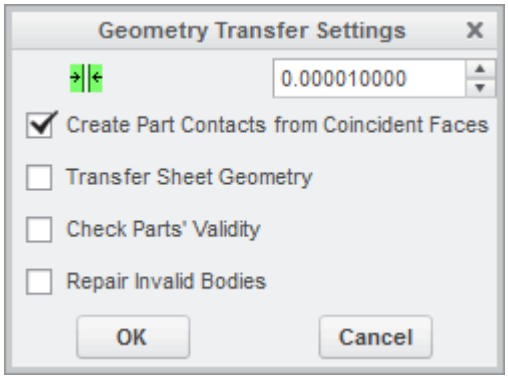
You can control how the geometry is transferred from Creo to STAR-CCM+ through the *Geometry Transfer Settings* dialog.

To access the *Geometry Transfer Settings* dialog:

1. Right-click the STAR-Creo root node and select **Geometry Transfer Settings**.



The *Geometry Transfer Settings* dialog appears.



2. Set the transfer parameters. See Geometry Transfer Settings Dialog.
3. Click **OK**.

Geometry Transfer Settings Dialog

Use the following properties to control how the geometry is transferred to STAR-CCM+.

Coincident Face Tolerance	Specifies the tolerance, in meters, for geometry in adjacent faces to be considered coincident. The default value is `1.0E-3`.
Create Part Contacts from Coincident Faces	Controls whether the geometry transfer feature searches for pairs of coincident faces on the CAD faces. For each pair found, part contact data is created and stored in the *Contacts* property (in STAR-CCM+). See the Creating Part Contacts from Coincident Entities section in the STAR-CCM+ user guide.
Transfer Sheet Geometry	Specifies whether to transfer sheet bodies as part of the transfer process. This option is turned off by default.
Check Parts' Validity	Provides you with information on whether the bodies are geometrically and topologically valid, from the Parasolid geometry kernel point of view.

	A simple message is displayed in the output window which states if the surface mesh is valid. For example: `Body 2 is not valid`.
Repair Invalid Bodies	Attempts to repair invalid bodies. The number of invalid bodies that are successfully repaired is displayed in the output window.

Understanding the STAR-Creo Process

In STAR-Creo the typical workflow for running a simulation is as follows:

1. Ensure that the CAD model is suitable for analysis
2. Extract the fluid volume
3. Define the material properties for each part of the model, for example, air, water, aluminum
4. Define model physics, for example, turbulence, temperature, gravity
5. Specify boundary conditions, for example, inlets, outlets, symmetry planes
6. Specify mesh settings
7. Define solution controls
8. Run the case
9. Post-process the results in STAR-CCM+

Once you have obtained an initial solution you may wish to re-run the case with some different settings, or with some changes to the geometry. To do this you can:

1. Return to STAR-Creo and change settings or modify the geometry
2. Re-run the case

Using Suitable CAD Models in STAR-Creo

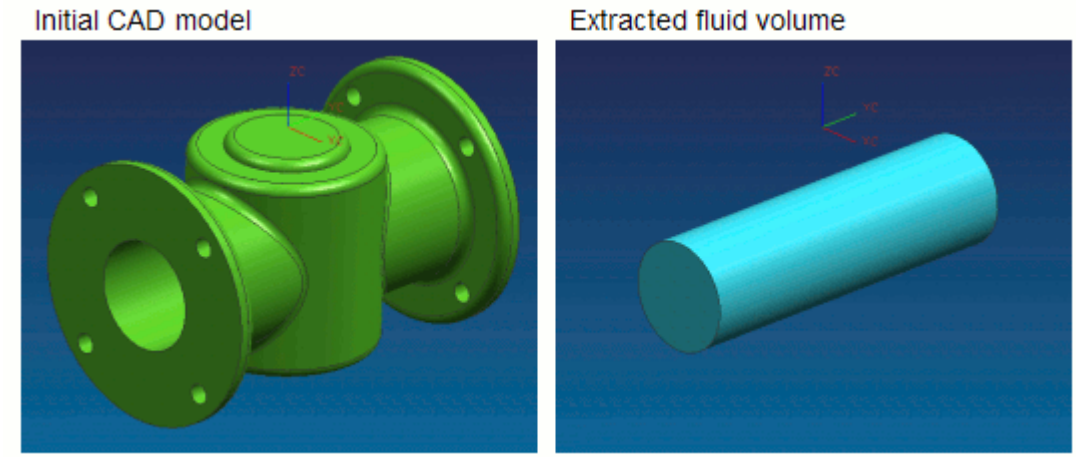
STAR-Creo only supports the use of solid bodies. If a geometry is made up entirely of surfaces then it will not be possible to define any boundary conditions, and the meshing options will be greyed out.

It is good practice to create copies of the original geometry file(s) before importing them into STAR-Creo.

Suppressed or deactivated parts and features are not supported by STAR-Creo and will have no effect on the analysis. Hidden parts, however, are used in the analysis. Ensure that all desired parts and features are active before using them with STAR-Creo.

Extracting the Fluid Volume

A CAD model usually represents a component made from solid material. However, in the case of flow simulation, a model representing the fluid volume is needed, requiring a different CAD model to be created. This is illustrated below:



In some instances, a simulation will require multiple regions, such as:

- Moving Reference Frames

  For rotating flow, such as turbines.
- Porous Media

  For representing regions with fine geometric structures
- Energy Source

  For simulating thermal specifications, such as conjugate heat transfer

These types of simulation require each region to be represented by individual contiguous parts within an assembly.

Defining Material Properties

Continua determines the characteristics of the fluid and solid entities used within the simulation. The behavior of a single continuum is primarily defined by the chosen material type and physics models.

Specification of the material type used in the simulation provides properties such as density, viscosity and conductivity, which are all used by the solver to calculate a solution. STAR-Creo includes a database of common materials and allows you to modify material properties where necessary.

Defining Model Physics

Physics models determine the nature of the fluid, and can be described as either:

- Turbulent or Laminar
- Constant Density or Ideal Gas

Additional models can also be activated to meet the needs of various simulation requirements:

- Passive scalar
- Temperature
- Gravity

A region is defined as a volume domain in a simulation, surrounded by boundaries. Each region in the simulation must be represented by a part in the native CAD package. When multiple regions are required, multiple parts must be used within an assembly.

All regions must be assigned to a continuum, as they represent a volume of material in the simulation. Region settings can be used to define porous media, moving reference frames, and an energy source.

Specifying Boundary Conditions

To define boundary conditions you first need to select the surface or surfaces on the model to assign to a separate boundary.

A boundary is defined as a surface, which along with other boundaries, surround and characterize a region. Boundary conditions determine how the boundary will affect the material within a region, and consequently, the simulation as a whole.

Inlet boundaries determine how much fluid enters the region, using a velocity, mass flow rate or stagnation inlet. Outlet boundaries define the process for fluid leaving the region, which are defined with either a pressure or flow-split ratio. Wall boundaries contain the fluid, and can also be used to define a thermal specification that will influence the transfer of energy to or from the region. Symmetry boundaries can be used to avoid modeling the entire region where appropriate, and so reduce calculation time.

As boundary types can affect the way in which the volume mesh is constructed, it is important to set them before generating the mesh.

Specifying Mesh Settings

The STAR-CCM+ solver uses the finite volume method to obtain an approximate solution to the underlying equations of fluid flow. This requires a mesh to be created, which represents the fluid volume as a collection of much smaller volumes known as cells.

A surface mesh is first created based on the boundaries defining the regions in the simulation. Thereafter a volume mesh is created to discretize the entire model.

The accuracy of the results is directly affected by mesh quality. If it is necessary to improve the quality of the mesh, some of the ways in which this can be done are to:

- Increase the mesh density, that is, the number of cells
- Add further sub-surface prism layers to capture near wall flow
- Refine the mesh in areas where flow characteristics are expected to change rapidly

As volume mesh quality is largely governed by the surface mesh quality, it is common practice to create and inspect the surface mesh before creating a volume mesh.

Defining Solution Controls

Solution controls determine the level of accuracy the solver will work towards. As a high accuracy solution will take longer, it may be necessary to compromise on this in order to obtain a result in a shorter time.

STAR-Creo includes five predefined solver settings ranging from fast (with low accuracy) to slow (with high accuracy). Advanced options are also available to provide further control.

Running the Case

When the simulation has been fully defined, the volume mesh is sent to STAR-CCM+ along with all the necessary data for running the simulation. The time to complete a simulation will vary depending on the mesh density and solution controls.

As the solver is running, residuals showing the rate at which individual equations are converging are shown in a plot. A scalar scene displaying absolute pressure is displayed on completion of the simulation.

Post-Processing the Results in STAR-CCM+

Post-processing of simulations is carried out using the STAR-CCM+ interface. This allows the results to be taken from the solver and displayed in a variety of scenes and plots.

Making Modifications to the Settings or Geometry

As STAR-Creo is completely integrated within the Creo environment, it is very simple to perform parametric studies that examine the response of the fluid to changes in geometry. Once the first analysis is complete, the boundary conditions, run settings and all other properties of the CFD model are stored within the CAD file. Any changes in the flow field due to design changes can be recalculated by generating a new mesh and running the solver.

Working with Simplified Representations in STAR-Creo

A simplified representation is a simplified form of a geometry model. The purpose of a simplified representation is to improve the performance of large models in Creo.

A complete model representation is known as the master representation. The procedure for transferring geometry with simplified representations to STAR-CCM+ is the same as the procedure for transferring geometry with a master representation. The difference is that features which are excluded in the simplified representation are not transferred to STAR-CCM+.

Assembly models with simplified representations can have components with the following status:

- Exclude Representation
- Geometry Representation
- Graphics Representation
- Light Graphics Representation
- Symbolic Representation
- Boundary Box Only Representation
- Default Envelope Only Representation
- Substitute Representation

If the status is *Substitute Representation*, STAR-Creo checks for its type. If the type is *Interchange*, then the substitute geometry is transferred to STAR-CCM+. All components in the simplified representations with the remaining status are not transferred to STAR-CCM+. If the status of a component is set to master representation, then the component is transferred to STAR-CCM+.

If you are setting up a CAE simulation within STAR-Creo, simplified representations can affect the way boundaries are displayed in the *Graphics* window. If you change between representations, you may require to click on a blank area of the STAR-Creo tree.

Connecting to Remote Machines

STAR-CCM+ uses a client-server architecture, and as such the CAD client (STAR-Creo) and the server (STAR-CCM+) are not required to run on the same machine.

Using an SSH (secure shell) connection, you can connect the client and server on two different machines, even when these are on different platforms. Due to how the CAD client and server interact in the background, the SSH connection must not ask for a username or a password.

There are two ways that you can connect STAR-Creo and STAR-CCM+, depending on what your starting point is:

- You can start from the CAD client and launch the server on a remote machine. This method is useful for transferring CAD geometry from STAR-Creo running on Windows to STAR-CCM+ running on Linux.
- You can start from the server and connect to the CAD client on a remote machine. This method is useful for connecting to STAR-Creo to update the CAD geometry. With this method, you can perform an Optimate analysis on a cluster while the geometry is updated in STAR-Creo.

Contents:

Running STAR-Creo with Remote STAR-CCM+
Running STAR-CCM+ with Remote STAR-Creo

Running STAR-Creo with Remote STAR-CCM+

Starting from STAR-Creo, you can choose to run the STAR-CCM+ server on a remote machine.

Using this method, you can set up the full CAE analysis in STAR-Creo on your local machine, and then run the STAR-CCM+ server on the remote machine. You can also choose to transfer your geometry from the CAD client to the remote STAR-CCM+ server. This feature is useful if you want to import geometry from STAR-Creo running on Windows to STAR-CCM+ running on Linux.

To configure STAR-Creo to launch STAR-CCM+ on a remote machine:

1. Make sure that you have a working SSH connection between the two machines.

   For the SSH connection, you require an SSH server on the machine running STAR-CCM+ and an SSH client on the machine running STAR-Creo. The SSH connection to the remote machine must not ask for a username or a password. To avoid possible issues with a connection that still requires a password, add `-o NumberOfPasswordPrompts=0` to the SSH command. Setting this argument to zero stops the connection prompting for a password when one is needed — something that could cause the process to hang. See SSH Command.
2. In the STAR-Creo ribbon, click **Preferences**.
3. In the *Preferences* panel, set *Server Settings* to **Remote**.
4. Set the remaining settings for your connection to the remote STAR-CCM+ server. See Server Settings Panel.

   This information is used to build a command-line in the background when STAR-Creo invokes STAR-CCM+.

5. If you are going to connect to the same remote host in future sessions, click **Save as Default Settings**.

The next time STAR-Creo invokes STAR-CCM+, the STAR-CCM+ server launches on the specified remote machine. A STAR-CCM+ client is also launched on the local machine to allow you to see the active session on the remote machine.

Contents:

Server Settings Panel

Server Settings Panel

Specify the connection settings to the remote STAR-CCM+ server using the Server Settings pane.

Parameter	Description	
Server Host Type	Specifies whether to run the STAR-CCM+ server on the local host, or on a remote host.	
	Local	Runs the STAR-CCM+ server on the same host as STAR-Creo.
	Remote	Runs the STAR-CCM+ server on the remote host that you specify.
SSH Command	Specifies the absolute path to the SSH program. If the path to the SSH program exists in your `Path` environment variable, you can leave this parameter as the default entry. You can include additional arguments in the SSH Command. The connection must not ask for a password or a username. To avoid possible issues with a connection that requires a password, add `-o NumberOfPasswordPrompts=0` to the SSH command. Setting this argument to zero stops the connection prompting for a password when one is needed, something that could cause the process to hang.	
Server Host	Specifies the remote host. You can enter the name, fully qualified domain name, IP address, or path to the host, depending on how your network is set up.	
Sim File Directory	Specifies the absolute path to the directory where you want to save the simulation file on the remote host. You must have read and write access to this directory. If a simulation file is present in this directory, STAR-CCM+ will use this file. If no file is found in this directory, a new one is created. An example directory on a Linux machine is shown below: `/users/linuxbox/james/documents/cadClientCases`	
Server Executable	Specifies the full path to the STAR-CCM+ executable on the remote host. An example path is shown below: `[STAR-CCM+_INSTALL_DIR]/star/bin/starccm+`	
Save as Default Settings	Saves the server settings that you specify in this panel as the default settings. Save these changes if you want to connect to the same remote server in future sessions.	

Running STAR-CCM+ with Remote STAR-Creo

Starting from STAR-CCM+, you can connect to STAR-Creo on a remote machine.

This feature is useful if you have previously transferred your geometry from STAR-Creo on Windows to STAR-CCM+ on Linux, and you want to reconnect the two.

To connect STAR-CCM+ running on the local machine to STAR-Creo running on a remote machine:

1. Make sure that you have a working SSH connection between the two machines.

   For the SSH connection, you require an SSH server on the machine running STAR-Creo and an SSH client on the machine running STAR-CCM+. The SSH connection to the remote machine must not ask for a username or a password.
2. In STAR-CCM+, expand the **Geometry** > **CAD Client Models** > **[Model_Name]** and set *CAD-Client Update Mode* to **Remote**.

   The **Remote CAD Client Settings** node appears.
3. Set the remaining parameters for the CAD geometry. See CAD Geometry Settings.
4. Select the **Remote CAD Client Settings** node and set the connection setting for the remote CAD client. See Remote CAD Client Settings.
5. If you are connecting from a STAR-CCM+ server that is running on a cluster, activate the *Use Proxy Server* option and set up the proxy server. See Proxy Server Settings.

When you make changes to the imported CAD geometry in STAR-CCM+, the changes are passed back to STAR-Creo on the remote machine and the geometry is transferred back into STAR-CCM+.

Contents:

Remote CAD Client Settings Reference

Remote CAD Client Settings Reference

Specify the connection settings to the remote CAD client in the Remote CAD Client Settings panel.

CAD Geometry Settings

Parameter	Description	
CAD File	Specifies the full name of the CAD geometry file, including the extension.	
CAD File Location	Specifies the location of the CAD geometry file on the remote machine. An example path is shown below: `C:\CAD Client Cases`	
CAD-Client Update Mode	Specifies whether STAR-Creo is located on the local machine where STAR-CCM+ is installed, or whether STAR-Creo is located on a remote machine.	
	Local	STAR-Creo is located on the same local machine as STAR-CCM+.
	Remote	STAR-Creo is located on a remote machine.

Remote CAD Client Settings

Parameter	Description	
Remote Host	Specifies the remote host where STAR-Creo is installed. You can enter the name, full name, IP address, or path to the host, depending on how your network is set up.	
Remote CAD-Client Install Location	Specifies the absolute path to the root directory of the CAD client installation on the remote host. An example path to the default installation directory for version 12.06.### is shown below: `C:\Program Files\CD-adapco\12.06.###\STAR-CAD12.06.###`	
SSH Command	Specifies the command to use to run the SSH program. If you require a proxy server to connect to a cluster, you must use the following SSH command: `ssh -A -t <Cluster_Head_Node> ssh -A`	
Use Proxy Server	Specifies whether you require a proxy server to connect to a cluster.	
	Activated	Uses a proxy server as part of the connection to a cluster.
	Deactivated	Does not use a proxy server.

Proxy Server Settings

Parameter	Description
Proxy Host	Specifies the host name of the private network of the cluster. Enter the name, fully qualified domain name, or IP address. An example is shown below: `clusterhead.cm.cluster`
Proxy Public Host	Specifies the host name of the public network of the cluster. Enter the name, fully qualified domain name, or IP address. An example is shown below: `clusterhead.london.company.com`

Defining Materials and Physics

Defining the physical properties of a STAR-Creo model involves specifying the materials, flow characteristics, operating conditions, and physics models.

These definitions and settings are accomplished through the use of continua and regions, which are described in the following sections.

Contents:

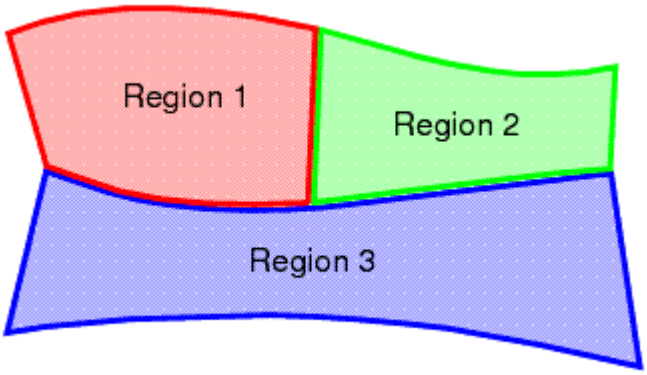
Working with Continua and Regions
Importing Geometry into STAR-CCM+ for Visualization

Working with Continua and Regions

Continua and regions are used together to capture the physical and material properties of a model.

A continuum represents a material in combination with a set of physics models that will be used to simulate the response of the material to the conditions imposed upon it. A continuum also provides the facility to define reference and initial values.

A region represents one or more volumes within the overall model, not necessarily contiguous, which will share the same continuum and set of boundary conditions. Regions are defined using CAD parts, and will be discretized by the volume mesher.



As well as holding a set of boundary conditions, regions can also be used to model specific physical behavior, including porous resistance, moving reference frames and energy sources.

Parts can be imported into STAR-CCM+ for visualization purposes without affecting the analysis.

Continua are assigned to single or multiple regions, which can contain single or multiple parts. In part mode, only one continuum and one region are available, as there is only one part. In assembly mode, new continua can be added to represent different materials present in the simulation, and new regions can be added to

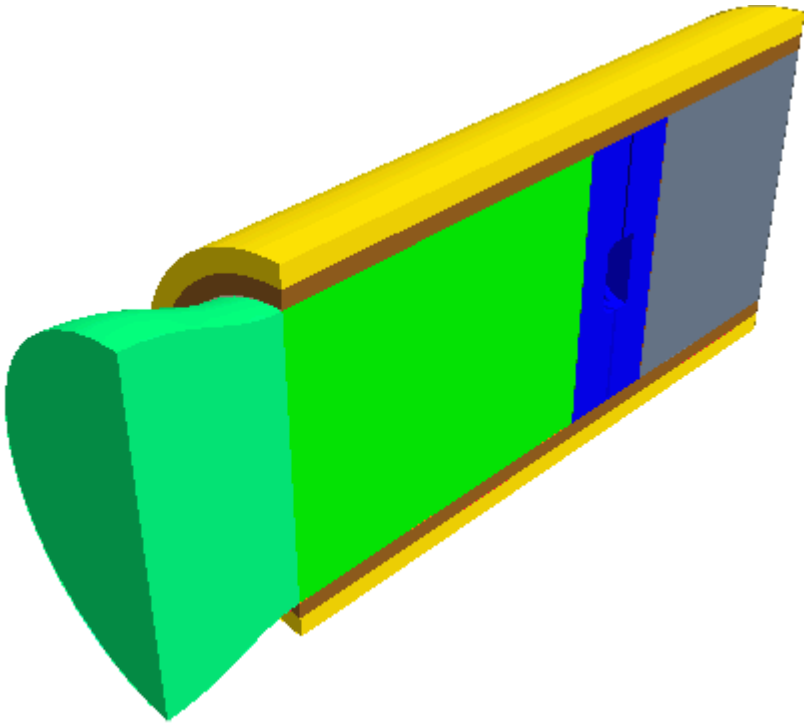
represent the parts composed of these materials. Continua and regions can be deleted if they are not required, although the simulation must always contain at least one of each.

There is a well defined relationship between continua, regions, boundary conditions and CAD parts.

- Every region must be associated with a continuum.
- Several regions can be associated with the same continuum.
- Every CAD part must belong to a region.
- Several CAD parts can be associated with the same region.
- Boundary conditions belong to regions, and cannot be shared between regions.
- Every CAD surface must have a boundary condition associated with it. A default wall boundary is applied to CAD surfaces that have not been specifically associated with a user-defined boundary condition.

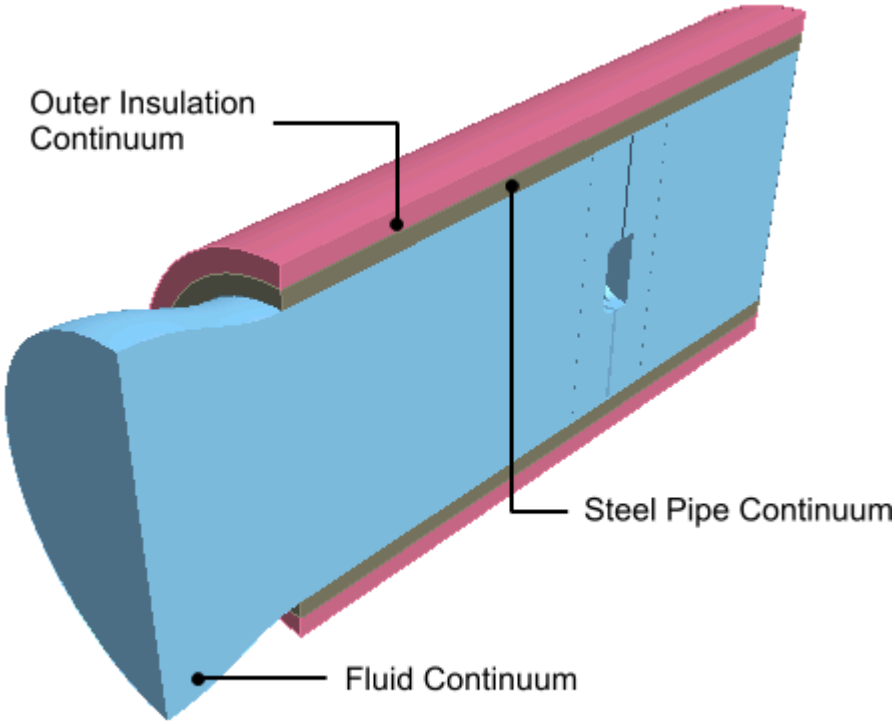
The following series of illustrations depict the relationship between CAD parts, continua, and regions, for a circular steel duct containing a rotating fan, and surrounded by a layer of insulation.

- Initially we have a collection of CAD parts in an assembly. These must be imprinted where appropriate to ensure that interfaces between parts can be created correctly.



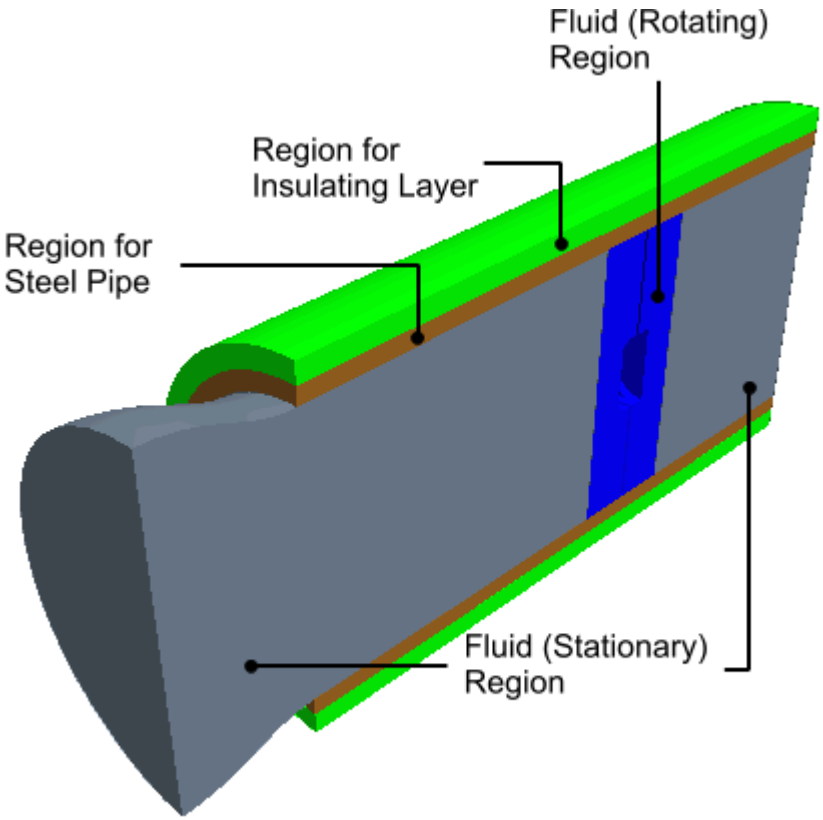
Assembly of CAD Parts (imprinted where necessary)

- Three continua are required to represent the three different materials present in the simulation: the fluid (for example, air), the steel pipe, and the insulation.



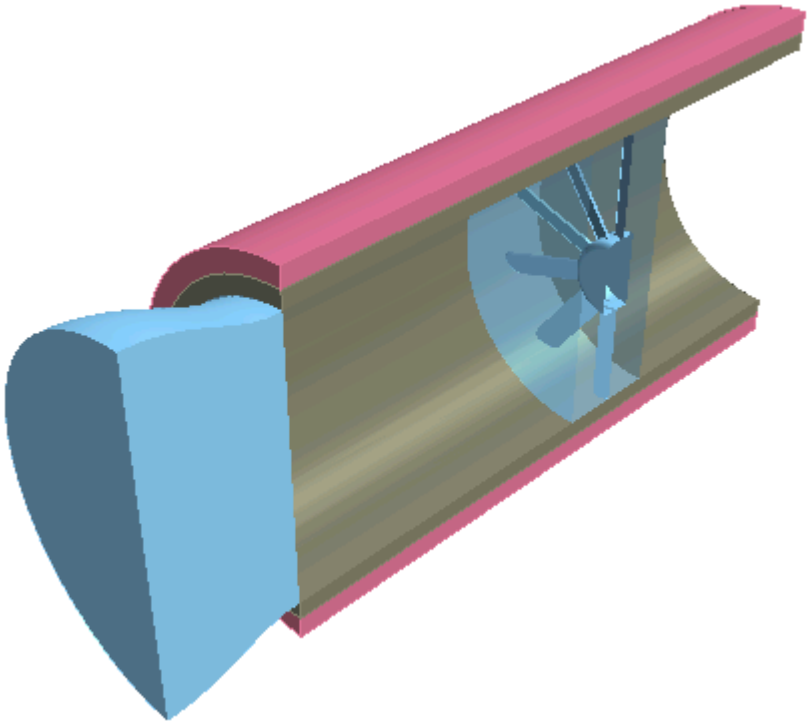
View of Continua Required for Simulation

- Two regions are required for the fluid portion of the simulation, as a rotating part is defined by applying a rotating frame of reference to a separate region surrounding the fan blades. Further regions are required for each of the solid parts (the steel pipe and insulating layer), which means a total of four separate regions overall.



View of Regions Required for Simulation

- A clearer picture can be given of the fan blade itself.



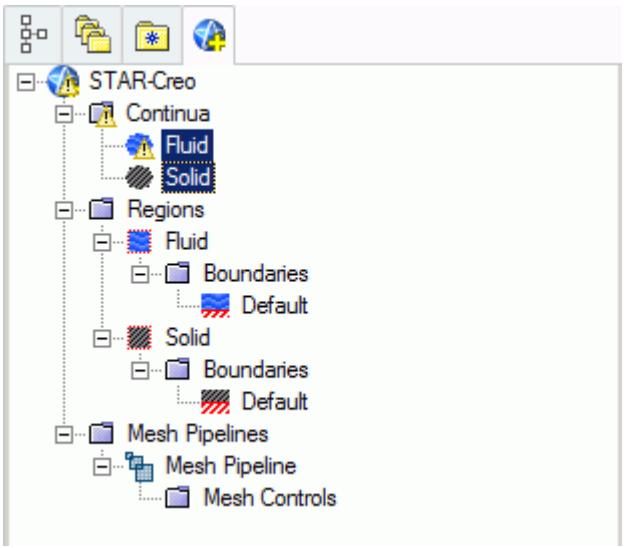
For an analysis containing a single part, for example a fluid-only case involving fluid flow through a converging-diverging pipe, the model will only require one continuum and one region in order to fully define the physical and material properties of the model. In this case, STAR-Creo automatically assigns the default continuum to the default region, and no further assignments are necessary.

Contents:

Setting Up Continua
Setting Up Regions

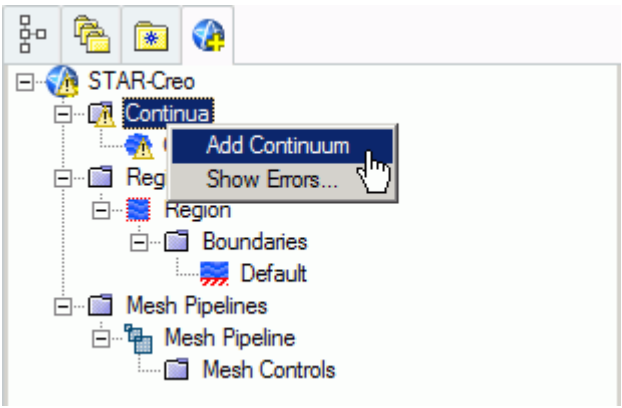
Setting Up Continua

To define different materials and associated physics for use in an assembly, you require additional continua. In each continuum you can specify a unique set of properties, such as a different material type or temperature. In assembly mode, continua are stored under a **Continua** node, as shown below.



To create an additional continuum:

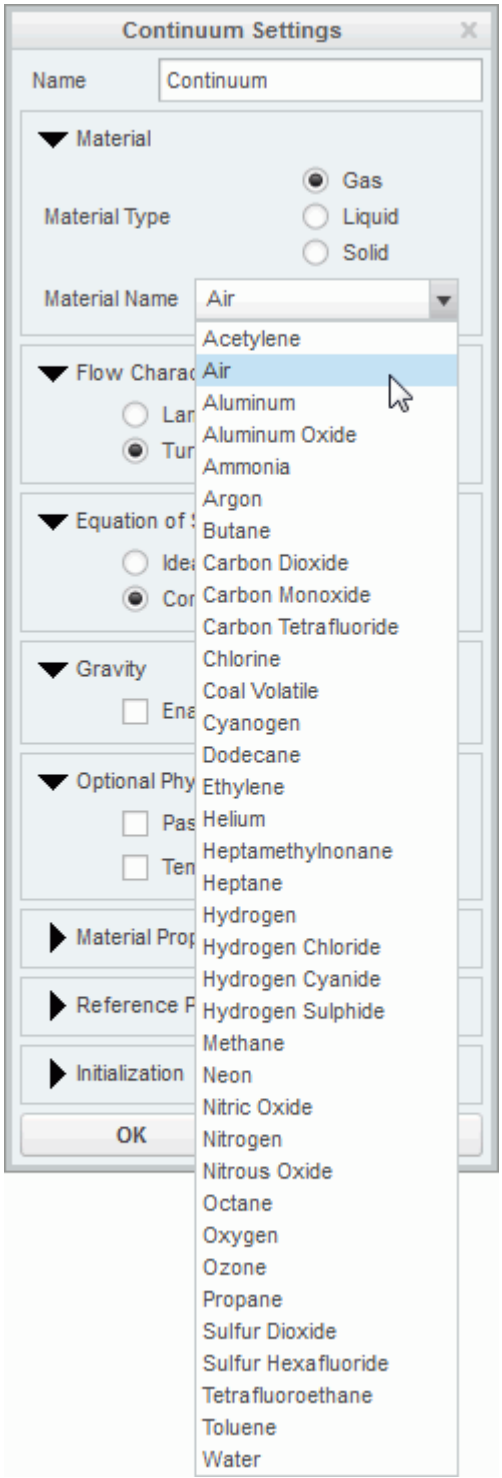
1. Right-click the **Continua** node in the object tree and select **Add Continuum**.



The *Continuum Settings* dialog appears.

2. In the *Name* field, enter a name for the continuum.
3. In the *Materials* group, select the type of material that you want to define:
   - **Gas**
   - **Liquid**
   - **Solid**
4. Select a material from the list.

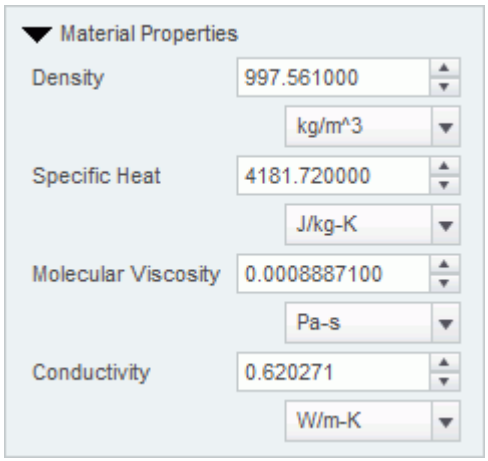
   You can modify the properties of any material that you select.



Note: You can select only one material per continuum. If your simulation contains a number of materials, you must create additional continua. This action is only possible for assemblies.

5. In the *Flow Characteristics* group, select the type of flow that you want to model: **Laminar** or **Turbulent**. This group only appears for fluids.
6. In the *Equation of State* group, select the equation of state : **Constant Density** or **Ideal Gas**. This group only appears for gases.

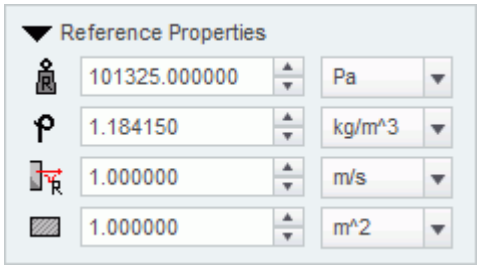
7. In the *Optional Physics Model* group, activate any optional physics models: **Passive Scalar** and **Temperature**.
8. Define the material properties in the *Material Properties* group.



See STAR-Creo Material Properties Panel.

Note: You can change units of properties from the drop-down menu.

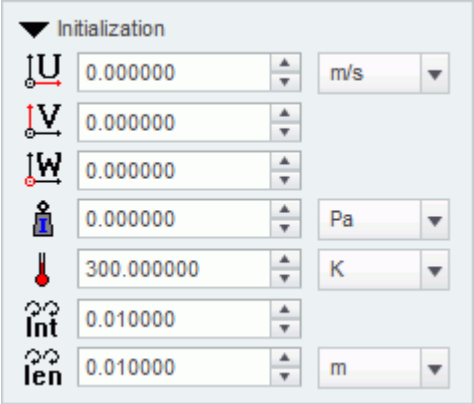
9. In the *Reference Properties* group, define the reference properties.



See STAR-Creo Reference Properties Panel.

10. If you want to include the body force due to gravity when calculating the momentum equations in STAR-CCM+, activate the gravity model in the *Gravity* group box.

    See Activating the Gravity Model in STAR-Creo.
11. Set the initial conditions in the *Initialization* group box.



See STAR-Creo Initialization Panel.

Specifying initial conditions that are close to the expected solution can reduce the number of iterations required to reach a solution. You can set the initial conditions similar to the inlet conditions of the fluid domain.

Initial conditions are specified at a continuum level; if no initial conditions are specified, STAR-Creo will use default values to initialize the solution.

12. Click **OK**.

    The continuum is added to the object tree.

Once you have created a continuum, you can assign it to a region.

Contents:

STAR-Creo Continuum Properties
Activating the Gravity Model in STAR-Creo

STAR-Creo Continuum Properties

The continuum dialog has a number of properties that you can define. This section provides a description for each property.

STAR-Creo Material Properties Panel

You can define the following material properties in STAR-Creo:

Property	Description
Density	Available for all types of materials. For gases, this property is available when you use the Constant Density equation of state.
Specific Heat	Available for all material types when you activate the Temperature physics model.
Molecular Viscosity	Only available for fluids.

Property	Description
Conductivity	Available for all materials when you activate the temperature physics model.
Molecular Weight	Only applies to gases. This property is available when you use the Ideal Gas equation of state.

STAR-Creo Reference Properties Panel

You can define the following continuum reference properties:

Property	Description
(*Reference Pressure*)	The pressure values that are defined in the simulation are all relative to the reference pressure. For more information, see the What is the Reference Pressure? section.
(*Reference Temperature*)	The reference temperature is used to define the operating temperature of the fluid in the buoyancy source term of the Boussinesq model. This parameter is used by STAR-CCM+ when the temperature, constant density, and gravity physics models are enabled.
(*Reference Density*)	These quantities are not used as part of the CFD calculation, but is necessary to calculate the force coefficients when post-processing.
(*Reference Velocity*)	
(*Reference Area*)	

STAR-Creo Initialization Panel

Specifying initial conditions that are close to the expected solution can reduce the number of iterations required to reach a solution. You can also set the initial conditions to match inlet conditions of the fluid domain. You can define the following continuum reference properties:

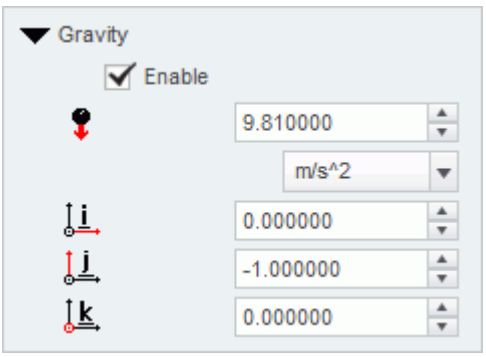
Property	Description
(*U Component of Velocity*)	The U direction vector for the initial velocity.
(*V Component of Velocity*)	The V direction vectors for the initial velocity.
(*W Component of Velocity*)	The W direction vectors for the initial velocity.
(*Initial Pressure*)	The initial pressure. The physical value of the initial pressure defines the starting pressure in the regions to which the continuum is assigned. The initial pressure is specified with respect to the reference pressure. The default initial pressure is `0 Pa`.

Property	Description
(*Initial Temperature*)	The initial temperature. This physical value of the initial temperature defines the starting temperature throughout any region to which the continuum is assigned. The default initial temperature is `300 K`.
(*Turbulence Intensity*)	The initial turbulence intensity. The default value is `0.05` (`5%`)
(*Turbulent Length Scale*)	The initial turbulent length scale. The default value is `0.01 m`.

Activating the Gravity Model in STAR-Creo

If you require the gravity model in STAR-Creo, you can activate it in the *Continuum Settings* dialog:

1. In the *Continuum Settings* dialog, activate the **Enable** checkbox in the *Gravity* group.



2. Set the acceleration due to gravity in the (*Acceleration due to gravity*) box.

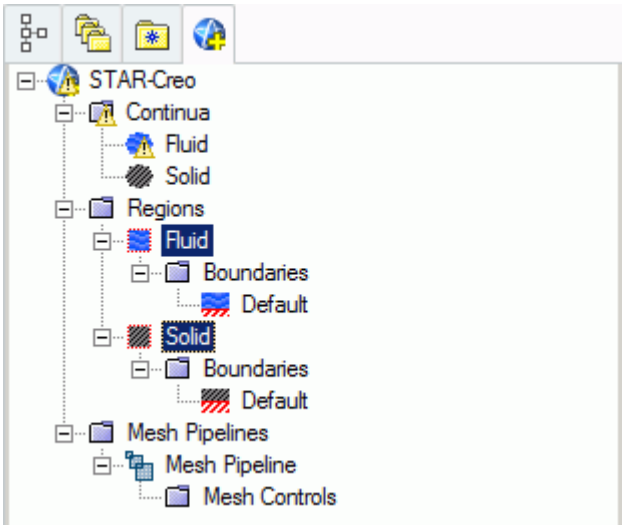
   You can change the units from the drop-down list.
3. Specify the direction vector for the gravity in the `[i, j, k]` *Components of Gravity* box.
4. Click **OK**.

   The continuum is added to the object tree.

Setting Up Regions

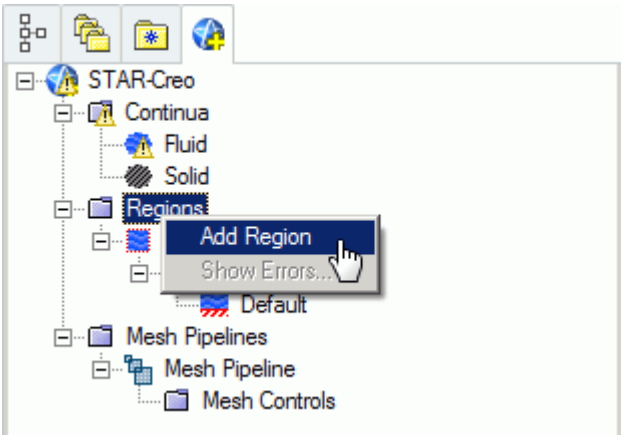
Additional regions are required when modelling multiple region physics, or when specific behavior such as porous resistance is required for a particular part within the same continuum. Assign parts from an assembly to regions. You can control the properties of these regions through the assigned continuum.

In assembly mode, regions are stored under the **Regions** node in the object tree, as shown below.

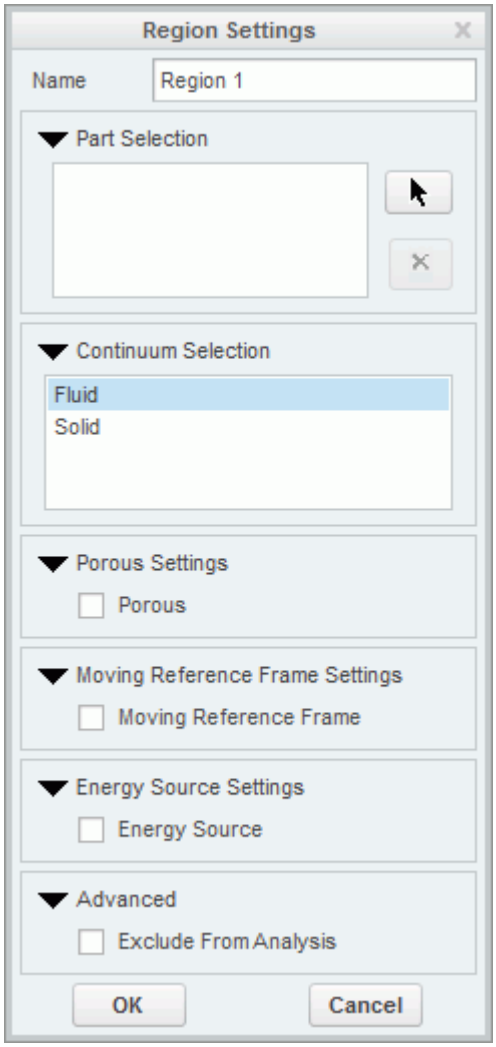


To create an additional region:

1. Right-click the **Regions** manager node in the object tree and select **Add Region**.



The *Region Settings* dialog appears.

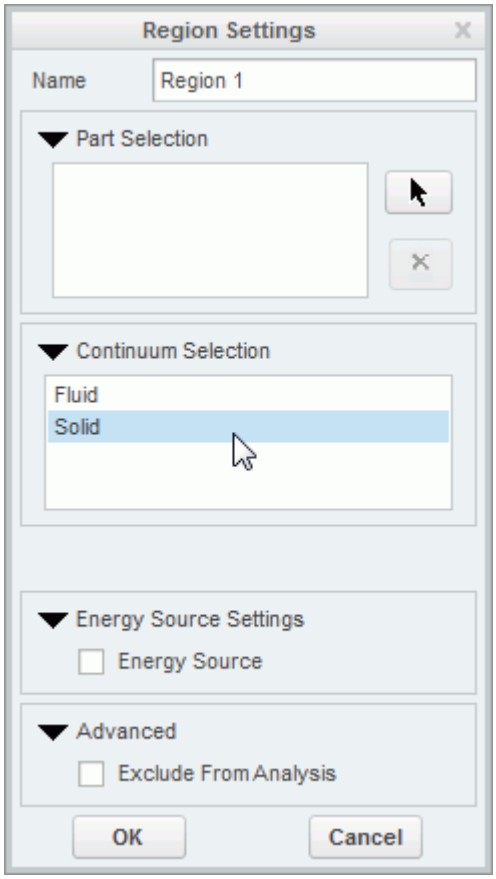


2. In the *Name* property, enter a name for the region.
3. Select the parts that you want to include in the region.

   You can select multiple parts from the graphics window or the Creo feature tree.

   Selected parts appear in the *Parts Selection* group box field.
4. Select a continuum to apply to the region.

   All available continua are listed here.



Note: Assigning a continuum to a region causes the region to inherit the physical and material properties defined in the continuum

5. Activate any additional region physics for the region:
   - Porous. See Defining a Porous Region.
   - Moving Reference Frame. See Defining a Moving Reference Frame.
   - Energy Source. See Defining an Energy Source.
6. If you want to import parts into STAR-CCM+ for visualization purposes, but do not want the region to be included in the CFD calculations, activate the *Exclude from Analysis* option. See Importing Geometry into STAR-CCM+ for Visualization.
7. Click **OK**.

A new region is added to the STAR-Creo object tree under the **Regions** node.

The parts that you assign to a region define the geometric extent of the region. Within this geometry, properties and behavior of the material is governed by the physics models and material properties defined in the region and continuum settings.

Note: Each region must be assigned at least one part, otherwise it will be ignored during the CFD analysis.

By default, every STAR-Creo model contains one continuum, and this is automatically assigned to the default region and each additional region that you create. If multiple continua are defined in the model, you can

choose a different continuum to assign to each region. This enables you to control the physical properties of each region, including varying material types.

All the parts in an assembly are assigned to the default region (this is the initial region created by STAR-Creo). When you assign a part to another region, it is automatically removed from the default region. If you remove a part from a region, it is reassigned back to the default region.

Contents:

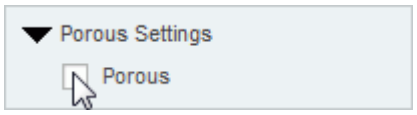
Defining a Porous Region
Defining a Moving Reference Frame
Defining an Energy Source

Defining a Porous Region

Porous resistance affects fluid flow through a region in a particular direction.

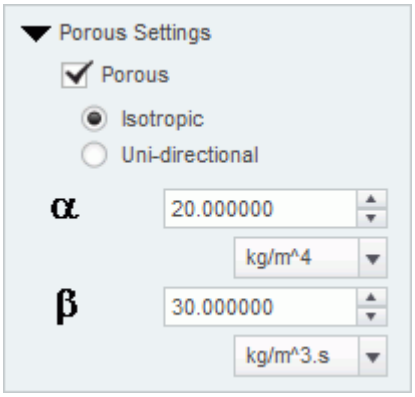
To define a porous region:

1. In the *Region Settings* dialog, activate the *Porous* tab.

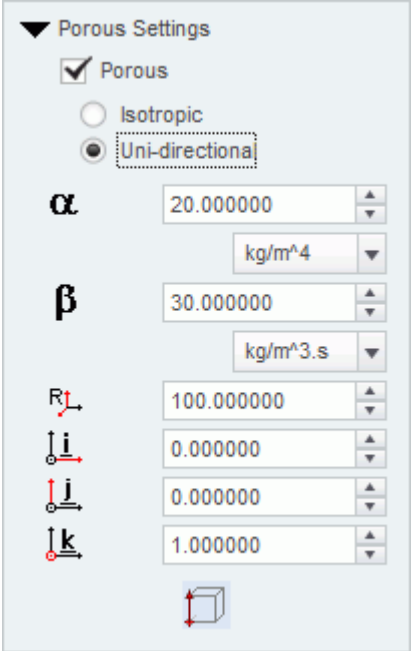


   The Porous settings appear.

2. Select the type of resistance to model:
   - **Isotropic**



   - **Uni-Directional**



3. Specify the porosity settings. See Porous Region Panel Settings.

Contents:

Porous Panel Settings

Porous Panel Settings

Porous resistance affects the fluid flow through a region in a particular direction. There are two types of porous resistances available: Isotropic and Uni-Directional.

Parameter	Description
Isotropic	
α (*Alpha*)	Specifies inertial resistance. You can change the units using the drop-down list.
β (*Beta*)	Specifies viscous resistance. You can change the units using the drop-down list.
Uni-Directional	
α (*Alpha*)	Specifies inertial resistance. You can change the units using the drop-down list.
β (*Beta*)	Specifies viscous resistance. You can change the units using the drop-down list.
(*Cross-Flow Resistance Factor*)	Specifies the cross-flow resistance factor.
(*i Component of Flow Direction*)	Specify the components that define the direction of the flow in the porous region. These values are

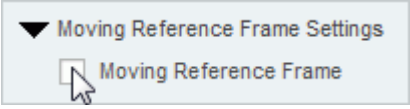
Parameter	Description
j (*j Component of Flow Direction*)	required to determine the coefficients of inertial and viscous resistance in a direction perpendicular to the specified vector.
k (*k Component of Flow Direction*)	

Defining a Moving Reference Frame

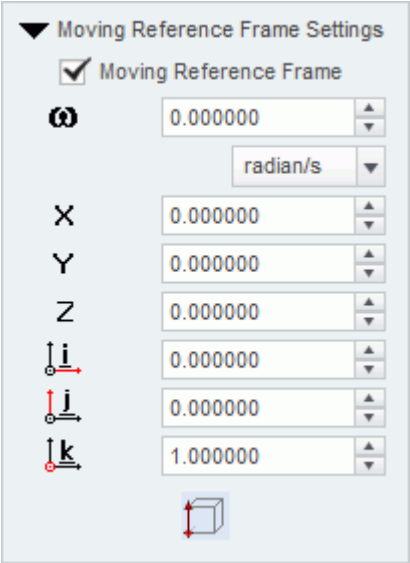
You can use moving reference frames to model region motion.

To define a moving reference frame:

1. In the *Region Settings* dialog, activate the *Moving Reference Frame* tab.



The Moving Reference Frame settings appear.



2. Specify the Moving Reference Frame parameters. See Moving Reference Frame Panel Settings.

Contents:

Moving Reference Frame Panel Settings

Moving Reference Frame Panel Settings

You can use Moving Reference Frames to model region motion, such as rotation.

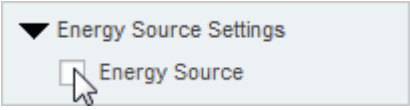
Parameter	Description
(*Rotational Velocity*)	Specifies the angular velocity. You can change the units from the drop-down list.
X (*X Coordinate of Axis Origin*)	Specify the coordinates for the origin of the axis of rotation. The units that are used here are the same as the file defaults.
Y (*Y Coordinate of Axis Origin*)	
Z (*Z Coordinate of Axis Origin*)	
(*i Component of Axis Vector*)	Specify the direction of the axis of revolution.
(*j Component of Axis Vector*)	
(*k Component of Axis Vector*)	
(**Set Axis Origin and Vector**)	Enables you to select two vertices on the geometry to define the axis of rotation. While holding down `<Ctrl>`, select two vertices on the geometry: the first selection acts as the origin of the axis of rotation, and the second selection define the direction vector of the axis.
Use Assembly Coordinate System	Controls whether the properties of the axis of rotation are defined using the coordinate system of the assembly or the coordinate system of the selected part. To use the coordinate system of the assembly, activate the *Use Assembly Coordinate System* this option. If you select multiple parts for the new region, the *Use Assembly Coordinate System* option is automatically activated and grayed out

Defining an Energy Source

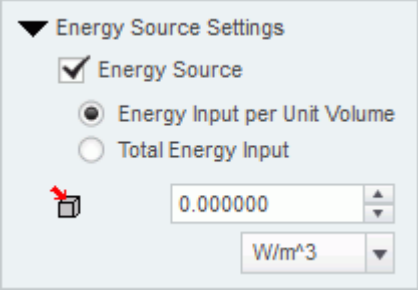
You can define a direct source or sink of energy to or from a region using the energy source model. This model is only available if you activated the Temperature physics model in the continuum.

To specify that a region is an energy source:

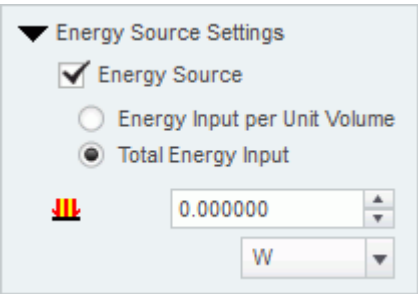
1. Make sure that the Temperature model is activated in the continuum that is applied to the region. See Setting Up Continua.
2. In the *Region Settings* dialog, activate the *Energy Source* tab.



   The Energy Source settings appear.
3. Select the type of energy source to apply:
   - **Energy Input per Unit Volume**



- **Total Energy Input**



4. Specify the energy source settings:
   - For energy input, set (*Energy Input per Unit Volume*) to the desired value.
   - For total heat, set (*Total Energy Input*) to the desired value.

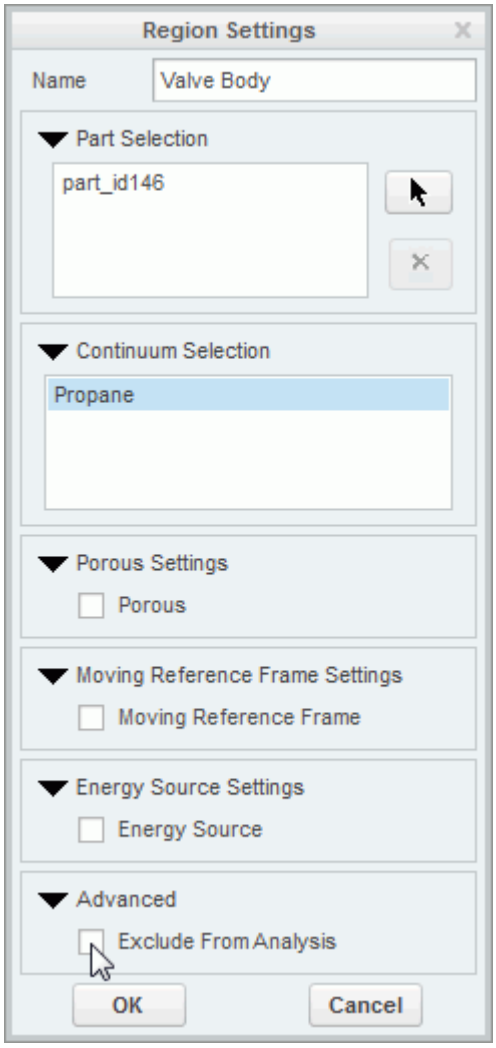
Importing Geometry into STAR-CCM+ for Visualization

You can import CAD parts, that do not form part of the analysis, into STAR-CCM+ for visualization purposes.

Add these parts to a separate region—this region is excluded from the analysis. This feature is useful for showing the original geometry in a post-processing scene in STAR-CCM+. For example, for an internal fluid volume of a pipe, the original geometry is the solid pipe.

To exclude CAD parts from the analysis:

1. In the *Region Settings* dialog, add the CAD parts that you want to use for visualization purposes.
2. Activate the *Exclude From Analysis* checkbox in the *Advanced* panel.



3. Click **OK**.

When you run the case, a surface mesh is generated for this region using the settings specified in the Mesh Pipeline.

Note: As is the case for all assemblies, you must imprint all parts that are used for visualization purposes onto any surface that they cut, otherwise you will not be able to generate the surface mesh.

An example of this feature is shown in the following section, where a valve and valve body geometry are imported into STAR-CCM+ for visualization.

Contents:

Setting Up a STAR-CCM+ Scene to Display Imported Geometry from STAR-Creo

Setting Up a STAR-CCM+ Scene to Display Imported Geometry from STAR-Creo

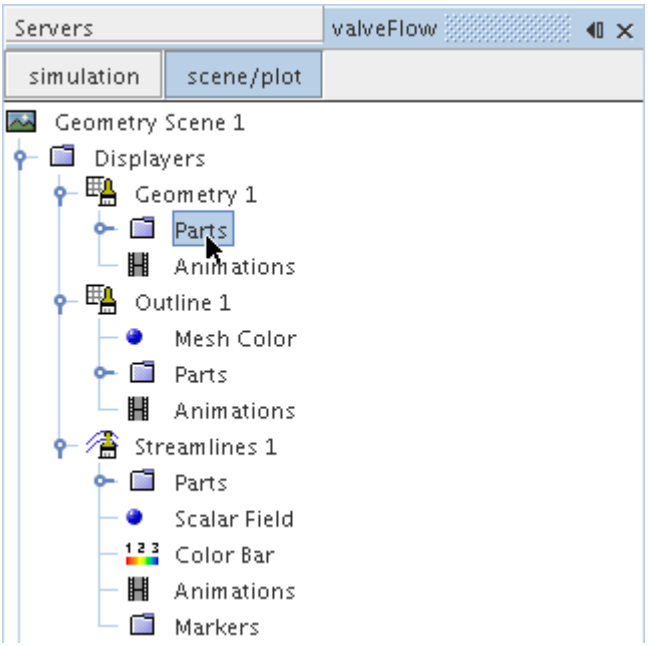
You can visualize geometry that is not included in the analysis. For example, you can add the valve and valve housing geometry to a Geometry Scene in STAR-Creo, even though this does not have any effect to the flow.

To use parts that are excluded from the analysis in a scene:

1. In STAR-CCM+, create a new scene.

   In the following example, a Geometry Scene is used.
2. Create a streamlines derived part.
3. Click on the **scene/plot** button.

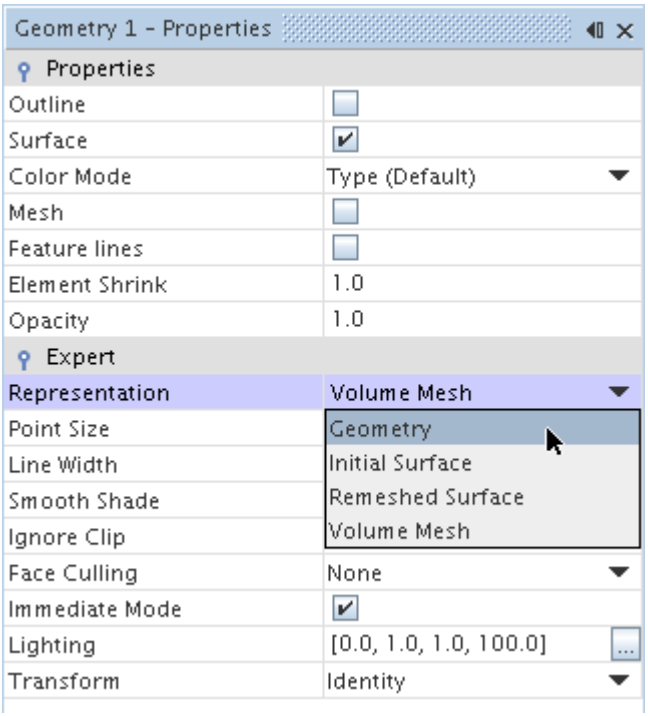
   The object tree switches to the scene nodes.
4. Double-click on the **Geometry Scene** > **Displayers** > **Geometry** > **Parts** node.



5. In the *Edit* dialog:
   a) Deactivate the **Regions** node.
   b) Expand the **Parts** node and activate the part that corresponds to the valve body.

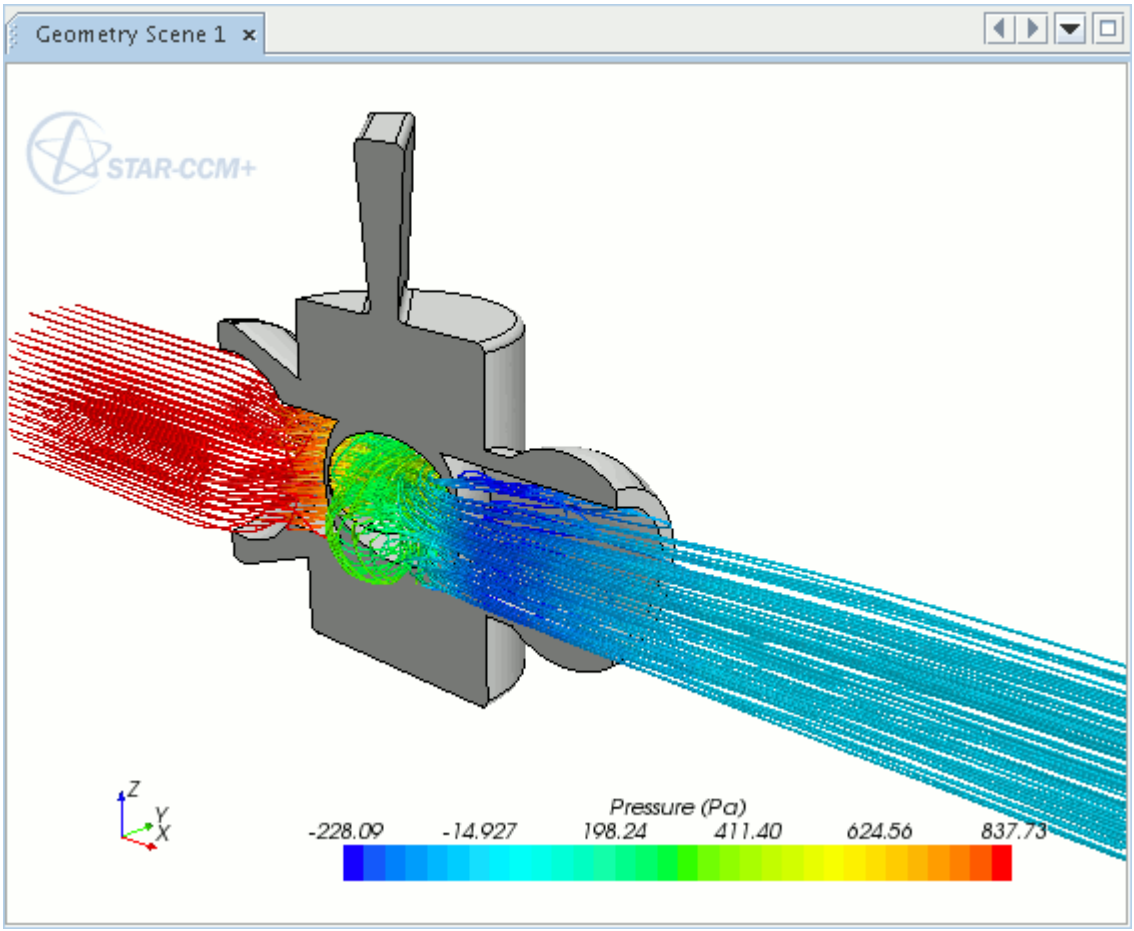
      This is the region that you chose to exclude from the analysis in STAR-Creo.
6. Click **OK**.
7. Select the **Geometry** node and set *Representation* to **Geometry**.

   The imported geometry only contains a surface mesh and does not appear by default (the default *Representation* is set to **Volume**). This is why you must change the *Representation* to **Geometry**.



8. Activate the *Outline* option to change the color of the outlines from red to black.

The finished scene is shown below.



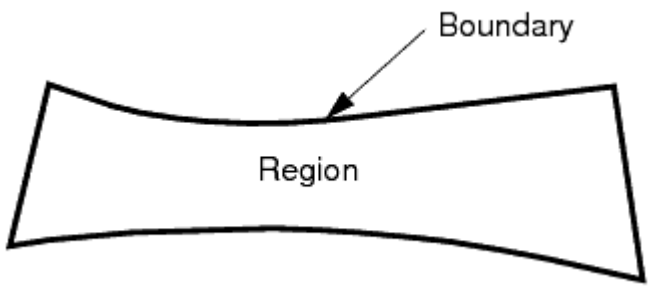
Displaying Imported Geometry in Scalar and Vector Scenes in STAR-CCM+

To display an imported geometry in a Scalar or Vector scene, you need to add a geometry displayer to the existing scene. You can then display the geometry by following steps described in Setting Up a STAR-CCM+ Scene to Display Imported Geometry from STAR-Creo.

Setting Boundary Conditions

Boundary conditions tell the STAR-CCM+ solvers how the substance within the region interacts with its environment. A velocity inlet boundary, for example, tells the solvers that an amount of substance is being added to the fluid domain at a certain rate.

A STAR-Creo boundary is made up of one or more CAD surfaces within the simulation which, along with other boundaries, surround and define a region.



The boundary is discretized by the surface mesher, which is then used by the volume mesher to mesh the entire region.

It is important to set the boundary conditions before creating the volume mesh, as some parts of the volume mesh, such as prism layers, depend on the type of boundary.

By default, each region contains only one boundary. To define more boundaries you need to add boundaries to the region. You can modify the properties of existing boundaries by editing its boundary node.

Some boundaries may require the specification of turbulence quantities, temperature, and passive scalar concentration, depending on whether these modeling options have been activated in Continuum Settings.

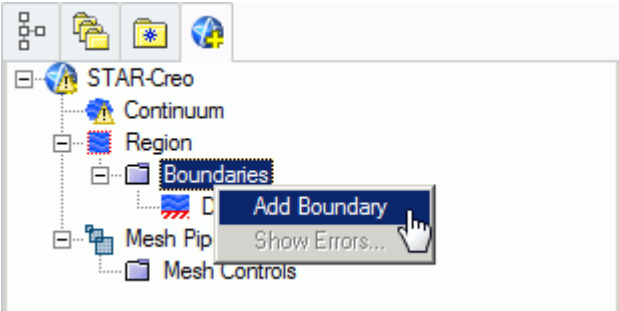
Contents:

Creating a New Boundary

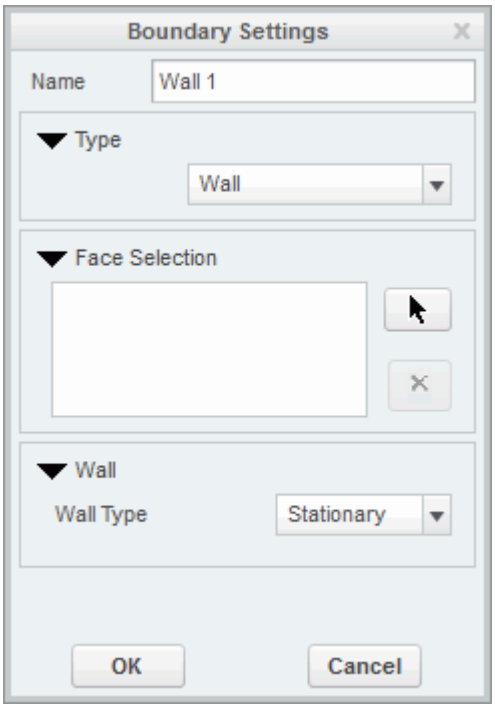
Creating a New Boundary

To create a new boundary:

1. Right-click the **Boundaries** node within the object tree and select **New**.



The *Boundary Settings* dialog appears.



2. On the CAD model, select the surfaces that you want to include in the boundary. See Assigning Surfaces to Objects.

   These selections are reported in the Included Surfaces box in the Surface Sets dialog.
3. In the *Name* field, enter a name for the boundary.
4. Select the boundary type and set the boundary conditions:
   - **Velocity Inlet**. See Velocity Inlet Boundary.
   - **Mass Flow Inlet**. See Mass Flow Inlet Boundary.
   - **Stagnation Inlet**. See Stagnation Inlet Boundary.
   - **Outlet**. See Outlet Boundary.
   - **Pressure Outlet**. See Pressure Outlet Boundary.
   - **Symmetry Plane**. See Symmetry Plane Boundary.
   - **Wall**. See Wall Boundary.
   - **Custom**. See Custom Boundary.

5. Click **OK**.

Contents:

Boundary Type Reference
Common Boundary Conditions

Boundary Type Reference

This section contains the various boundary types that are available in STAR-Creo.

Velocity Inlet Boundary

A velocity inlet boundary represents the inlet of a duct at which you know the flow velocity. The velocity inlet settings are displayed below.

Property	Description
Normal Velocity	
(*Normal Velocity*)	Specifies the velocity normal to the surface. You can change the units using the drop-down list.
Cartesian Velocity Components	
U (*U Component of Velocity Vector*)	Specify the components of the velocity vector. You can change the units using the drop-down list.
V (*V Component of Velocity Vector*)	
W (*W Component of Velocity Vector*)	
Cylindrical Velocity Components	
V_r (*r Component of Velocity Vector*)	Specify the cylindrical components of the velocity vector. You can change the units using the drop-down list.
V_θ (*theta Component of Velocity Vector*)	
V_z (*z Component of Velocity Vector*)	
X (*X Location of Axis of Origin*)	Specify the coordinates of the origin for the velocity vector.
Y (*Y Location of Axis of Origin*)	
Z (*Z Location of Axis of Origin*)	
i (*i Component of Direction Vector*)	Specify the components for the direction vector.
j (*j Component of Direction Vector*)	
k (*k Component of Direction Vector*)	

Mass Flow Inlet Boundary

A mass flow inlet boundary represents the inlet of a duct at which you know the mass flow rate. The mass flow settings are displayed below.

Property	Description
Mass Flow Normal to Surface	
(*Mass Flow Rate*)	Specifies the mass flow rate, normal to the boundary surface. You can change the using the drop-down list.
Mass Flow in Specified Direction	
(*Mass Flow Rate*)	Specifies the mass flow rate. You can change the using the drop-down list.
(*i Component of Mass Flow*)	Specify the direction components of the mass flow.
(*j Component of Mass Flow*)	
(*k Component of Mass Flow*)	

Stagnation Inlet Boundary

The stagnation inlet boundary is an inlet condition that is well-posed for compressible flows, although it is equally valid for incompressible flows. The stagnation conditions refer to the conditions in an imaginary plenum, far upstream, in which the flow is completely at rest. The stagnation inlet settings are displayed below.

Property	Description
Stagnation Flow Normal to Surface	
(*Stagnation Pressure*)	Specifies the stagnation pressure. You can change the units using the drop-down list.
Stagnation Flow in Specified Direction	
(*Stagnation Pressure*)	Specifies the stagnation pressure. You can change the units using the drop-down list.
(*i Component of Direction Vector*)	Specify the direction components for the flow.
(*j Component of Direction Vector*)	
(*k Component of Direction Vector*)	

Outlet Boundary

An outlet boundary splits the flow between multiple outlets using a fraction of the total mass flow through each of the outlet boundaries. The outlet panel settings are provided below.

Property	Description
60 40 (*Flow-Split Ratio*)	Specifies the ratio of the split as a value between 0 and 1. The sum of all the flow-split ratio values in a region must equal 1. If you only have one flow-split boundary in the simulation, all the flow is assumed to exit through this boundary.

Pressure Outlet Boundary

A pressure boundary is a flow outlet boundary at which you specify the pressure. The pressure outlet panel settings are provided below.

Property	Description
(*Pressure*)	Specifies the pressure on the selected boundary. You can change units using the drop-down list.

Symmetry Plane Boundary

A symmetry plane boundary represents an imaginary plane of symmetry in the simulation. The solution that is obtained with a symmetry plane boundary is identical to the solution that would be obtained by mirroring the mesh about the symmetry plane (in half the resulting domain). There are no settings which are specific to symmetry plane boundaries.

Wall Boundary

A wall boundary represents an impermeable surface. The wall panel settings are provided below.

Property	Description	
Wall Type	Specifies the type of conditions to apply to the wall boundary.	
	Stationary	Specifies the wall as a stationary impenetrable surface.
	Translating	Specifies the wall as a moving impenetrable surface. You specify the velocity for the translation.
	Rotating	Specifies the wall as a rotating impenetrable surface.
	Slip	Specifies the wall as a stationary, impenetrable, but friction-free surface.
Translating		
U (*U Component of Velocity Vector*)	Specify the velocity vector. You can change the units using the drop-down list.	
V (*U Component of Velocity Vector*)		
W (*U Component of Velocity Vector*)		

Property	Description
Rotating	
(*Rotational Velocity*)	Specifies the rotational velocity. You can change the units using the drop-down list.
X (*X Location of Axis Origin*)	Specify the coordinates of the origin for the axis of rotation.
Y (*Y Location of Axis Origin*)	
Z (*Z Location of Axis Origin*)	
(*i Component of Direction Vector*)	Specify the components of the axis of rotation
(*j Component of Direction Vector*)	
(*k Component of Direction Vector*)	

Custom Boundary

In STAR-CCM+, there are additional boundary types to those available in STAR-Creo. The custom boundary in STAR-Creo acts as a placeholder for a boundary that you will customize in STAR-CCM+.

Common Boundary Conditions

Some boundary conditions are not specific to any boundary type and appear in multiple boundaries, so long as the physics model that is associated with the condition is activated.

Common Boundary Conditions on Inlet and Pressure Outlet Boundaries

Temperature, turbulence, and passive scalar boundary conditions are common to all inlet boundaries as well as the Pressure Outlet boundary.

Property	Description
(*Temperature*)	Specifies the temperature on the boundary surface. You can change the units using the drop-down menu.
(*Turbulence Intensity*)	Specifies the turbulence intensity on the boundary surface. You can change the units using the drop-down menu.
(*Turbulent Length Scale*)	Specifies the turbulent length scale on the boundary surface. You can change the units using the drop-down menu.
(*Passive Scalar*)	Specifies the amount of passive scalar crossing the boundary. You can change the units using the drop-down menu.

Boundary Conditions on Wall Boundaries

When the energy equation is being solved, you can specify the thermal specification on wall boundaries.

Property	Description	
Adiabatic	Specifies that no transfer of energy occurs between the boundary surface and the surrounding environment.	
Convection	Specifies the transfer of energy between the boundary surface and the fluid moving over the boundary surface.	
	HTC (*Heat Transfer Coefficient*)	Specifies the heat transfer coefficient. You can change the units using the drop-down menu.
	(*Ambient Temperature*)	Specifies the ambient temperature. You can change the units using the drop-down menu.
Heat Flux	Specifies the rate of energy transfer through the boundary surface.	
	(*Heat Flux*)	Specifies the heat flux. You can change the units using the drop-down menu.
Temperature	Specifies the temperature on the boundary surface.	
	(*Temperature*)	Specifies the temperature on the boundary surface.

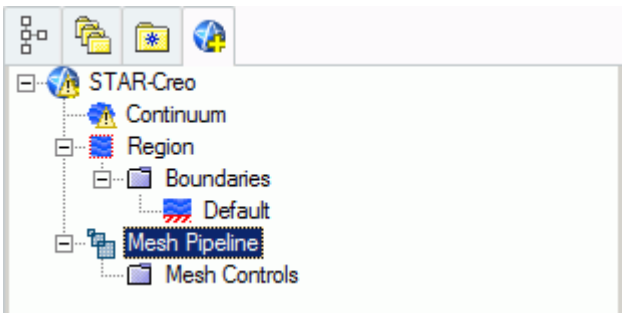
Meshing the Model

STAR-Creo provides options for generating polyhedral or trimmed volume meshes, with or without near-wall prism layers.

By default, prism layers are generated on all wall surfaces. The provision of a prism layer depends on the type of boundary that is applied to a face. As a consequence, it is recommended that you only generate the mesh after setting up the boundaries and regions.

The meshing process first creates a triangulated surface (the surface mesh) and thereafter converts it to the desired volume mesh (polyhedral or trimmer). As the quality of the volume mesh depends to a large extent on the quality of the surface mesh, it is prudent to check the surface mesh before proceeding with the volume mesh.

Although the entire meshing process can be launched with a single command (the **Build And Run** command from the STAR-Creo ribbon), it is possible to execute only single stages from the STAR-Creoribbon. Settings that are made in the **Mesh Pipeline** node govern the mesh generation process.



A mesh pipeline contains all the settings that are applied to the surface and volume meshes created on the Creo parts to which it is applied. In single part mode, only one mesh pipeline is available. In assembly mode, multiple mesh pipelines can be created if necessary.

When mesh generation commences, each mesh pipeline will:

1. Extract geometry from Creo.
2. Transfer the model to STAR-CCM+.
3. Execute the surface remesher.
4. Execute the polyhedral or trimmer volume mesher with the prism layer mesher (if selected).
5. Create a mesh representation in the Creo *Graphics* window.

Controls on the top-level mesh pipeline node allow you to set the base sizes for the surface mesh, and control the prism layer. Further refinement to specific faces, parts, or volume shapes can be made by adding additional controls to the mesh pipeline.

Contents:

Creation of Additional Surfaces in STAR-CCM+
Working with Mesh Pipelines

Working with Mesh Controls
Generating the Mesh STAR-Creo
Common Causes for Meshing Failures

Creation of Additional Surfaces in STAR-CCM+

STAR-CCM+ creates geometry part surfaces during the geometry transfer stage.

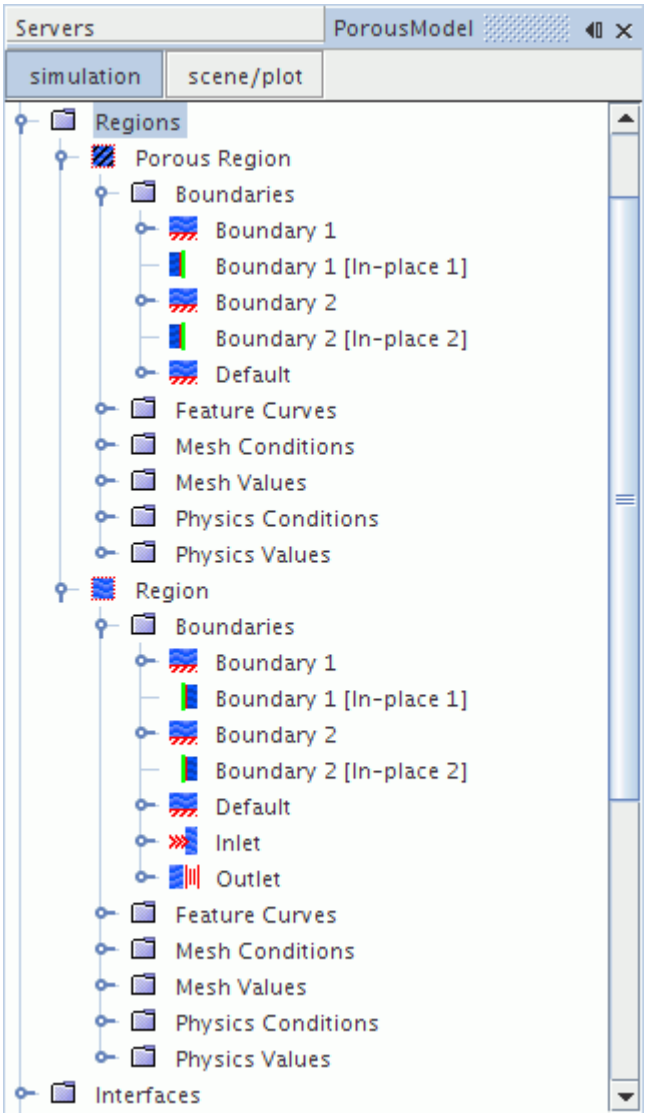
Initially, STAR-Creo creates one part surface for each face in the CAD model. These part surfaces are then combined so that one surface per boundary remains. This outcome, however, is not possible for boundaries that contain CAD surfaces from more than one part. The final part surfaces appear in the **Geometry** > **Parts** node in STAR-CCM+.

The following items affect the number of part surfaces that are created in STAR-CCM+:

- **The number of boundaries present in the model, including the default boundary.** Each boundary in STAR-Creo appears as a separate part surface in STAR-CCM+. If, however, a boundary contains CAD surfaces from multiple parts, the part surfaces in STAR-CCM+ cannot be combined.
- **The number of mesh pipeline controls used (if any).** Each CAD surface that is used to define a face mesh control in STAR-Creo appears as a separate part surface in STAR-CCM+.
- **The number of interfaces that are created during the meshing process (for assemblies only).** All the CAD surfaces that define one interface between two parts in an assembly are grouped together under one part surface in STAR-CCM+. One part surface exists for each interface in STAR-CCM+.

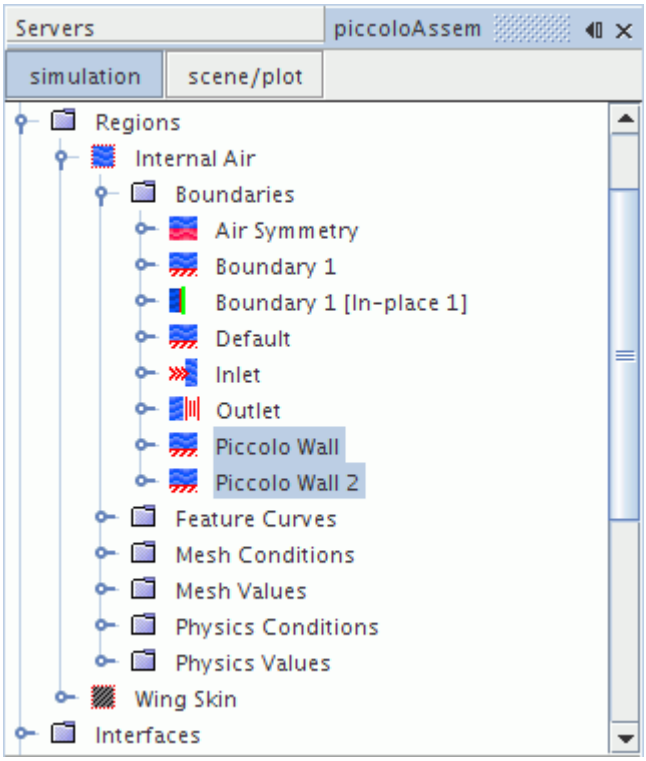
Boundary Splitting due to Interface Generation

Interfaces are automatically created between two parts in an assembly during the mesh generation. A new boundary is created for each interface on the surfaces that share the interface.



Boundary Splitting due to Face Mesh Controls and Part Mesh Controls

If you use a face mesh control or a part mesh control in your simulation, some boundaries may be split, depending on what faces are included in the mesh control. If a boundary and mesh control share the same faces, no splitting occurs. If a mesh control contains faces from multiple boundaries, splitting occurs and a new boundary is created. The name of the new boundary is derived from the original boundary.



Working with Mesh Pipelines

When working with a single part, you will only have one mesh pipeline defined in the model tree. This will specify the mesh settings for the single part available. In assembly mode, multiple mesh pipelines can be created to give specific controls for different parts, and parts must be specifically associated with a mesh pipeline.

Contents:

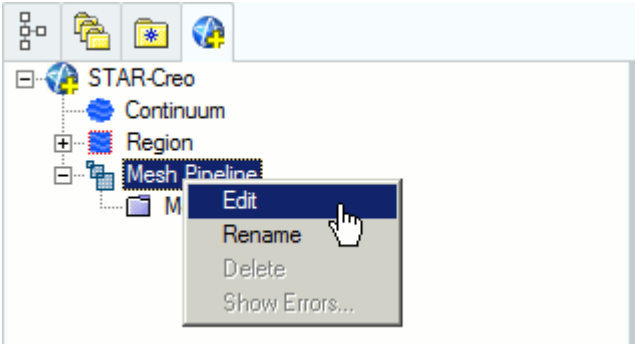
Setting Up the Mesh Pipeline

Setting Up the Mesh Pipeline

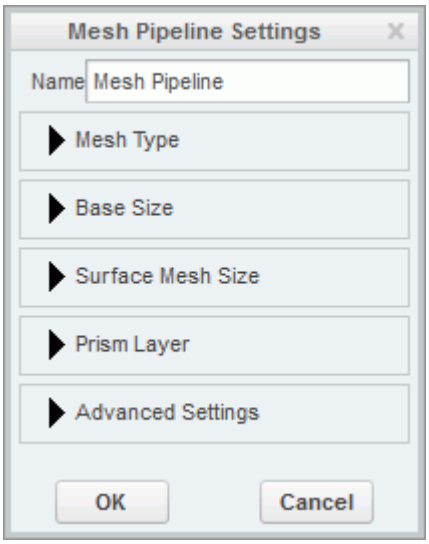
The mesh pipeline node contains the general mesh settings that will apply to all STAR-Creo parts associated within the pipeline in the absence of any further controls.

To set up the mesh pipeline:

1. Right-click the **Mesh Pipeline** node in the STAR-Creo tree and select **Edit**.



The *Mesh Pipeline Settings* dialog appears.



2. In the *Name* field, enter the name of the mesh pipeline if required.
3. In the *Part Selection* box, specify the desired parts to use in the mesh pipeline.

   This property is only available when working with assemblies.
4. From the *Mesh Type* tab, select the type of mesh that you want to generate:
   - **Polyhedral**
   - **Trimmer**
5. Set *Base Size* to the desired value.

   The base size is a reference length used in other meshing parameters. It may be set to equal the length of the largest triangle you expect to have on your surface mesh, or to some other characteristic dimension.

   The base size can be set as either a relative or absolute value. When it is set as a relative value, the reference size will be the diagonal of the bounding box enclosing the entire geometry. In the case of multiple parts in an assembly, the reference size will be calculated based on the diagonal length of the bounding box enclosing the entire assembly. This includes all parts from all mesh pipelines. The unit specification menu will become active when setting an absolute value.
6. Set *Surface Mesh Size* to the desired value.

The surface mesh size is the base edge length of triangles on the surface mesh. Note that the ultimate length of some (or all) triangles on the surface mesh will depend on other factors such as surface curvature and edge proximity.

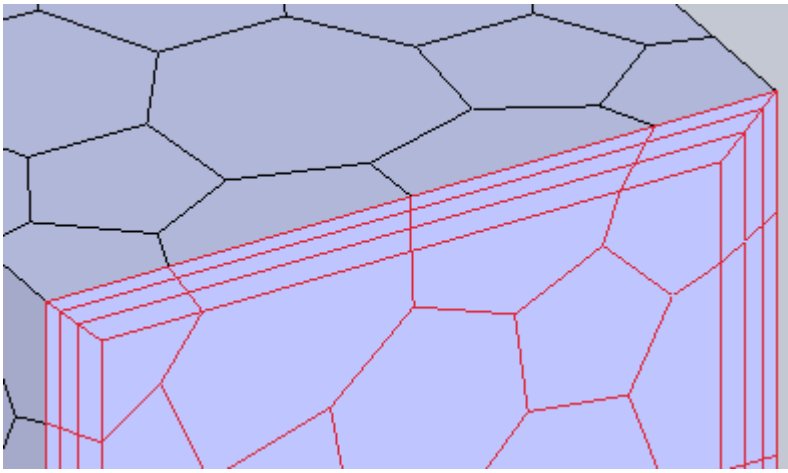
7. Set the prism layer settings. See Prism Layer Panel.
8. Set the advanced settings. See Mesh Pipeline Advanced Settings Panel.
9. If you selected to generate a trimmed mesh, set the Trimmer settings. See Trimmer Panel.
10. Click **OK**.

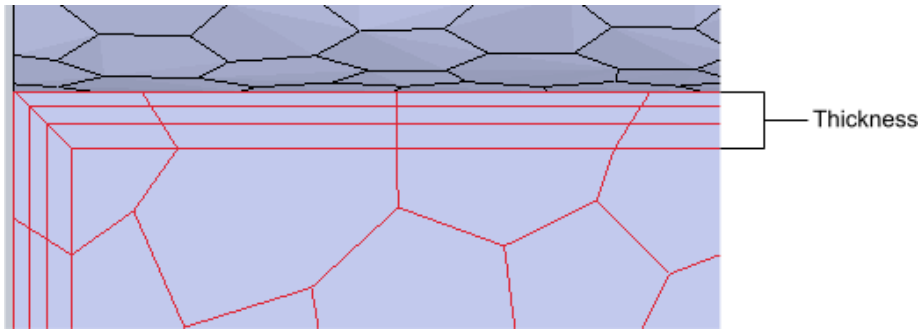
Contents:

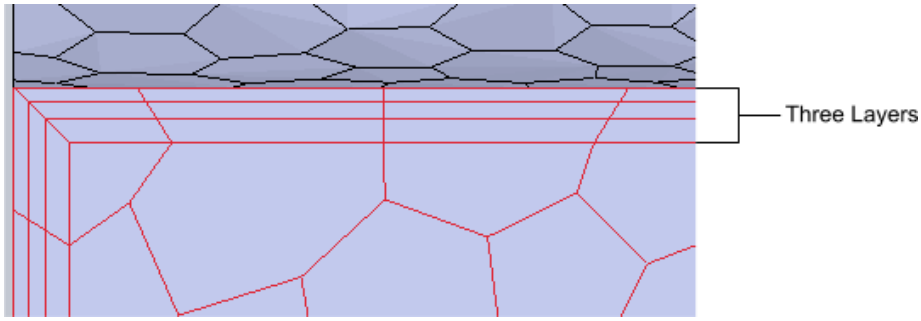
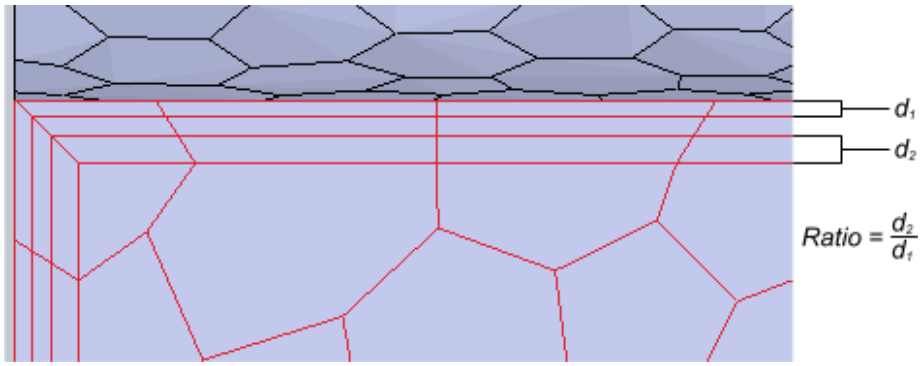
Prism Layer Panel
Mesh Pipeline Advanced Settings Panel
Trimmer Panel

Prism Layer Panel

Prims layers improve the simulation near non-slip walls, particularly when a turbulence model is active.



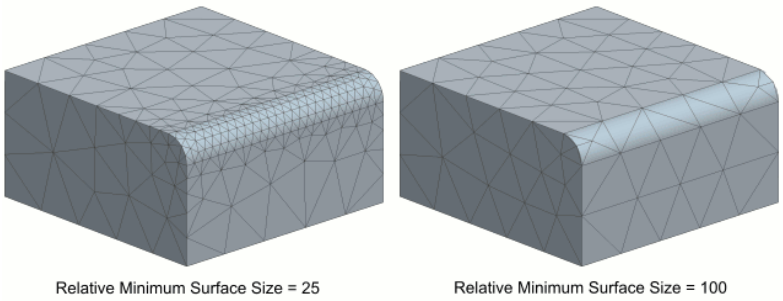
Property	Description
Layer Thickness	Defines the overall depth of the prism layer, which includes all of the individual layers, as illustrated below. 

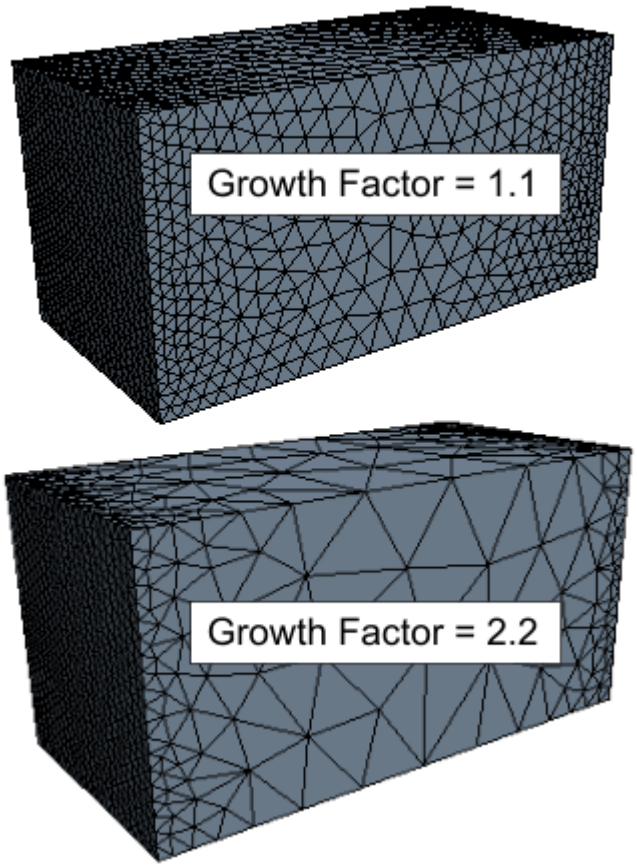
Property	Description
Number of Layers	Defines how many individual cell layers will be included in the overall prism layer, as illustrated below. 
Thickness Ratio	Specifies the size of the first and last layers within the prism layer. 

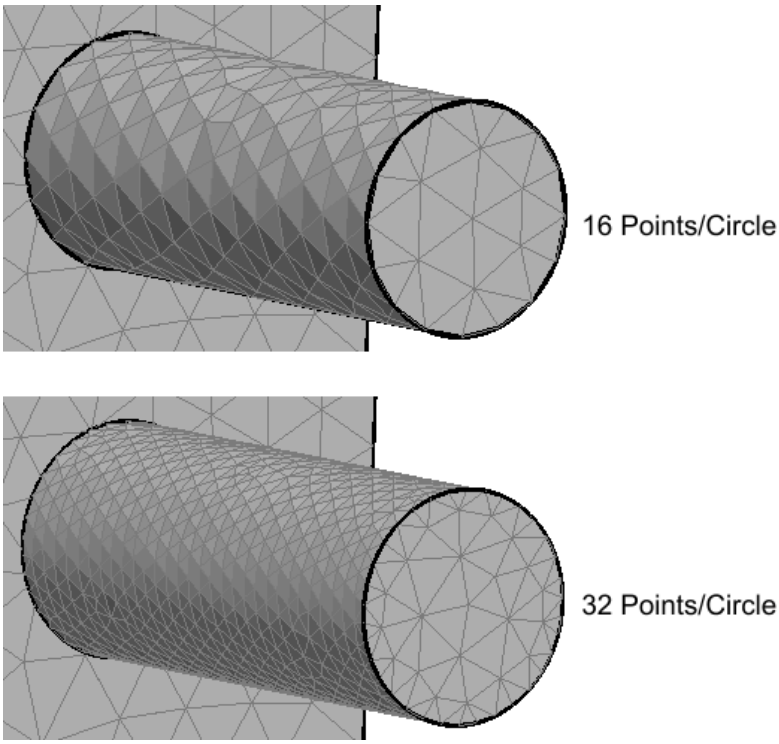
Mesh Pipeline Advanced Settings Panel

The advanced settings panel provides further customization of meshing parameters.

Property	Description
Relative Minimum Surface Size	Defines the minimum triangle size permitted on a feature curve. The minimum triangle size is set as a percentage of the surface mesh size. The default value is set to 25%. You must set the relative minimum surface size to a value greater than zero. If you set the relative minimum surface size to 100%, STAR-CCM+ will not modify the triangle size when it detects curvature. The figure below illustrates this behavior. Note that the base size was set to 20% in this example to emphasize the refinement on the curve.

Property	Description
	 Having a large value for the number of points on a circle property increases the mesh resolution on curved surfaces. The transition between the fine mesh and coarse mesh areas is controlled by the growth ratio.
Growth Ratio	Controls the rate at which triangle sizes change when moving from one triangle size to another. In the example below, the end faces of the block were set to have a smaller triangle size than the four side faces of the block. When the growth ratio is set to a low value, the transition between triangle size is very gradual. When a higher value is used, the transition is far more rapid.

Property	Description
	 The growth factor must be set to a value greater than 1.0. There is no upper limit for the growth factor.
Number of Points per Circle	Controls the number of triangle edges that appear on a curved surface, as shown below.

Property	Description
	 Regardless of the number of points per circle defined, STAR-Creo will not permit the triangle edge length to shrink below 25% of the base size. The curvature length scale may also affect the number of edges on a curved surface.

Trimmer Panel

The trimmer mesh is controlled using the Trimmer panel.

Property	Description
Default Growth Rate	Controls the stepping of the refinement from one cell size to the next. Five settings are available. By default, the *Default Growth Rate* is set to **Fast**, meaning that the fastest transition from the smallest cell size to the largest will be used. This usually equates to having only one layer of each cell size for each transition, which will result in the least number of cells being created for the supplied input values. The **Medium**, **Slow**, and **Very Slow** options further slow the transition from one cell size to the next in the mesh interior and therefore increase the overall cell count. In general, the medium option results in a minimum of two equal sized cell layers per transition, slow in four layers and the very slow option in eight layers.
Boundary Growth Rate	Controls the number of cell layers to grow next to surfaces in the template. Five settings are available. By default, the *Boundary Growth Rate* is set to **None**, meaning that the setting used for the Default Growth Rate is used

Property	Description
	for controlling all transitions. If the **Very Slow**, **Slow**, **Medium**, or **Fast** options are selected then it will result in eight, four, two or one layers of cells respectively of the same size being created next to the geometry surfaces in the template.

Working with Mesh Controls

Mesh controls allow you to define custom mesh settings for a particular section of the model.

Multiple controls can be assigned to the same item to control different mesh parameters. If, for example, a face has two mesh controls assigned to it--one that controls the mesh size and another that controls the prism layer thickness, the final mesh will respect both the custom mesh size and the custom prism layer thickness. If two mesh controls set the same mesh parameter, the control created last will take precedence over the first.

Contents:

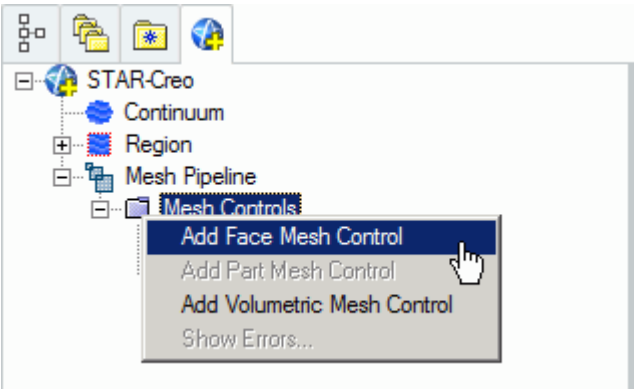
Using a Face Mesh Control
Using a Volumetric Mesh Control
Using a Part Mesh Control

Using a Face Mesh Control

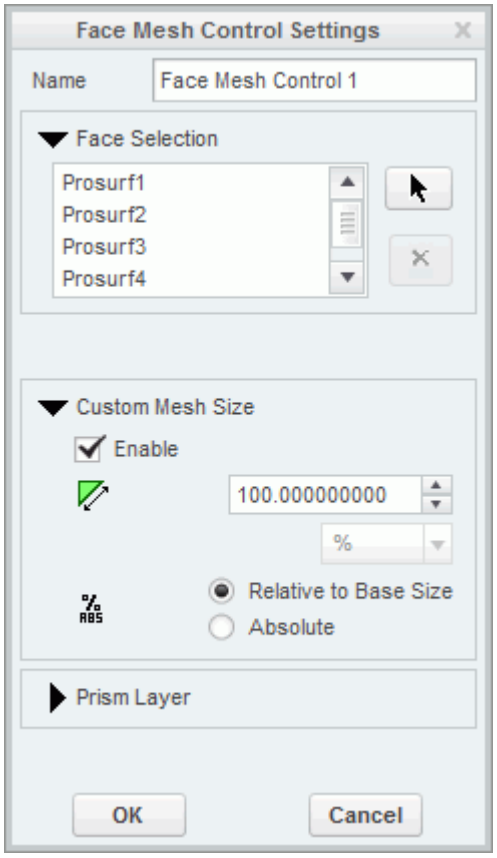
A face mesh control allows you to set the surface mesh size and the prism layer parameters on a face. Custom settings made on a face control will take priority over settings in the parent mesh pipeline, and over settings made in any part mesh controls.

To add a face mesh control:

1. Right-click the **Mesh Pipeline** > **Mesh Controls** node in the STAR-Creo tree and select **Add Face Mesh Control**.



The *Face Mesh Control Settings* dialog appears.



2. Set *Name* to an appropriate value.
3. On the CAD geometry, select the desired faces to include in this mesh control.
4. If you want to specify a custom surface mesh size on the selected surfaces, activate the **Custom Mesh Size** option.

   a) Set (*Custom Mesh Size*) to the desired value.Set (*Mesh Size*) to the desired value.

   You can enter this value as a percentage of the base size, or as an absolute value. If you select to enter the absolute value, you can select the units from the drop-down menu.

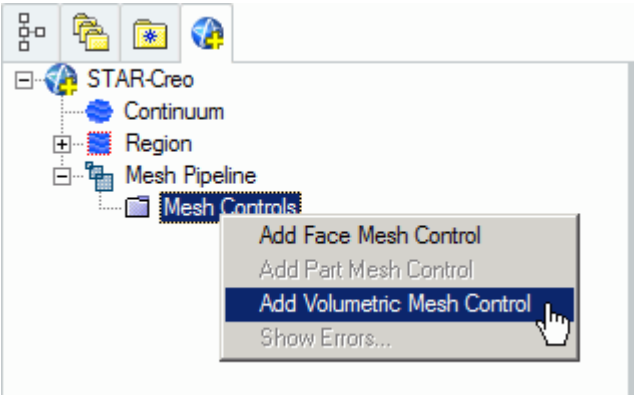
5. Set (*Prism Layer Option*) to the desired value:
   - **Inherit Values**: in the first instance, the face(s) will be assigned values from any part mesh control applied to the part to which the face(s) belong. If this does not apply, values will be taken from any applicable volumetric mesh control, and in the absence of this values will be taken from the parent mesh pipeline.
   - **Use Custom Values**: allows the prism layer parameters to be set directly on the face(s). See Prism Layer Panel.
   - **Disable**: no prism layer will be generated on the face(s).
6. Click **OK**.

Using a Volumetric Mesh Control

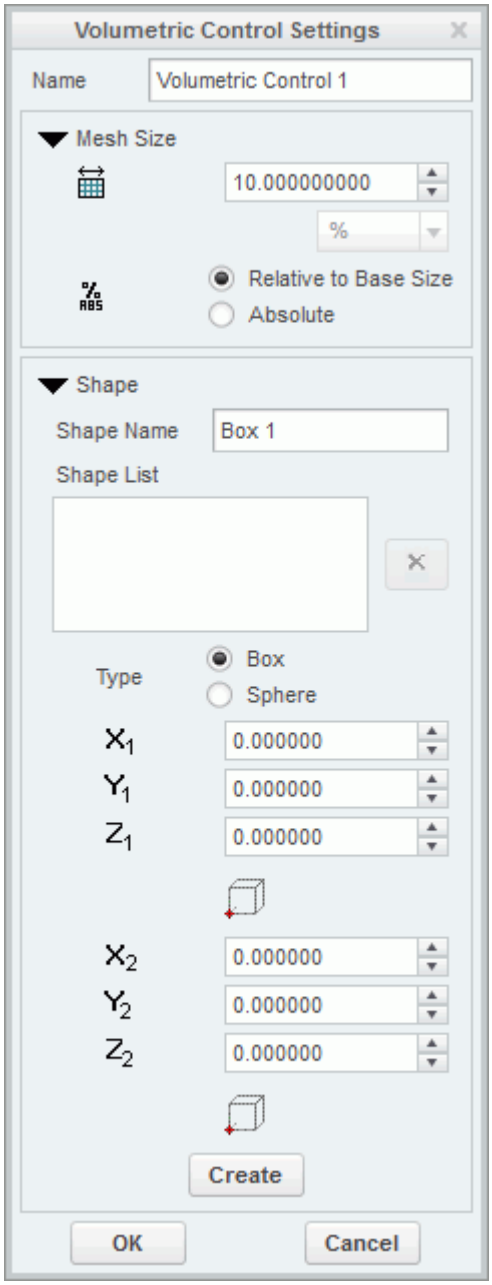
A volumetric control permits mesh refinement within a user-defined box or sphere. Only the mesh size can be modified with a volumetric control; prism layer parameters are not affected.

To add a volumetric mesh control:

1. Right-click the **Mesh Pipeline** > **Mesh Controls** node in the STAR-Creo tree and select **New Volumetric Mesh Control**.



The *Volumetric Control Settings* dialog appears.



2. Set *Name* to an appropriate value.

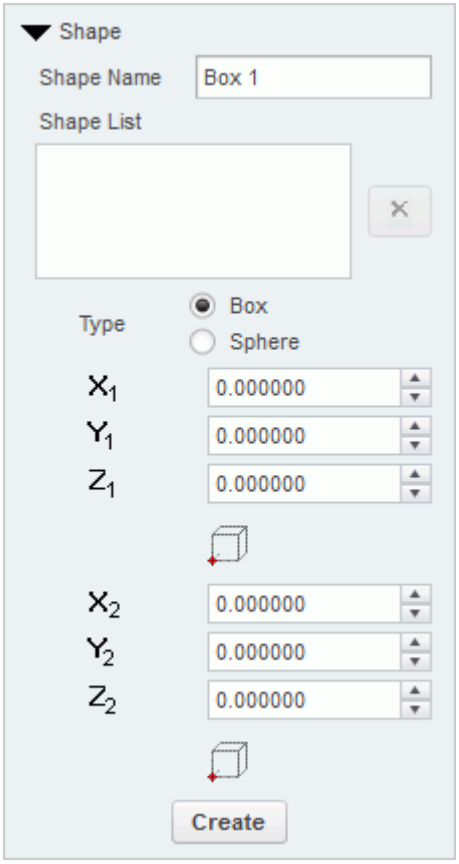
3. Set (*Mesh Size*) to the desired value.

   The mesh size defines the triangle edge length for surface mesh within the volume shape, and the characteristic cell size for volume mesh. The custom mesh size will still affect the volume mesh in cases where the volume shape does not intersect any surfaces of the Pro/E geometry.

   The mesh size can either be defined relative to the base size of the parent mesh pipeline, or as an absolute dimension with length units.

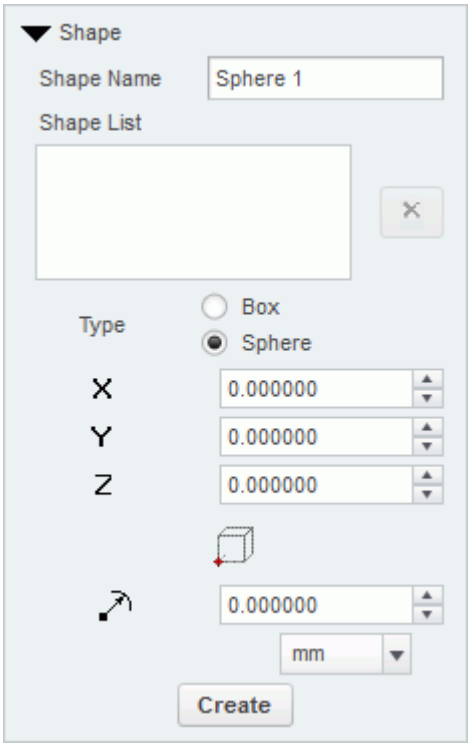
4. Select the type of volumetric control to create:
   - **Box**
   - **Sphere**

5. Enter the parameters for the volumetric shape:
   - If you are creating a box, enter the coordinates for Point 1 and Point 2.



X, Y, and Z coordinates of the two opposite corners of the box are required by the panel. These coordinates can either be entered manually, or taken from existing points within the Pro/E window. To use an existing point, click (**Set Point**) and use the Creo point selection operation to select the appropriate point.

   - If you are creating a sphere, enter the coordinates for the center point and the radius of the sphere.



The sphere shape is defined by a center point and a radius. The center point can be set either by entering the coordinates directly, or by clicking the (**Set Point**) button and using the Pro/E point selection procedure to choose an existing point in the Pro/E window. The coordinates of the existing point will be used to populate the panel.

6. Click **Create**.

   The shape is added to the *Shape List* list.

7. Click **OK**.

Editing a Volume Shape

To edit a volume shape

1. Select the volume shape from the list within the **Shapes** panel.
2. Edit the parameters as required and click **Update**.

Deleting a Volume Shape

To delete a volume shape

- Select it from the list in the *Shape* panel and click the (**Delete**).

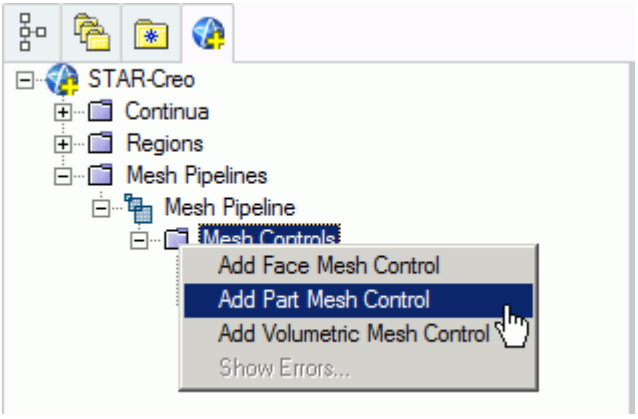
Using a Part Mesh Control

Custom mesh settings (such as surface mesh size and prism layer parameters) can be applied to one or more parts using a part mesh control.

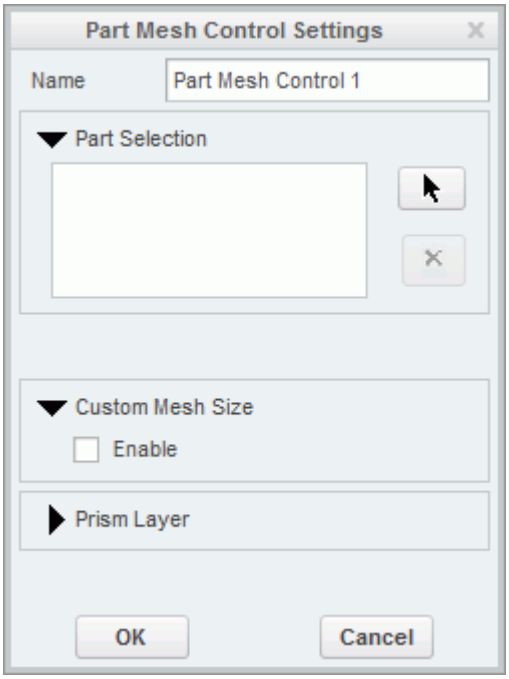
Settings made on a part mesh control will take priority over settings made on the parent mesh pipeline, but will not have priority over settings made on a face mesh control.

To add a part mesh control:

1. Right-click on **Mesh Pipelines** > **Mesh Pipeline** > **Mesh Controls** in the STAR-Creo tree and select **Add Part Mesh Control**.



The *Part Mesh Control Settings* dialog appears.



2. Set *Name* to an appropriate value.
3. On the CAD geometry, select the desired parts to include in this mesh control.
4. If you want to specify a custom surface mesh size on the selected surfaces, activate the **Custom Mesh Size** option.

   a) Set (*Custom Mesh Size*) to the desired value.Set (*Mesh Size*) to the desired value.

You can enter this value as a percentage of the base size, or as an absolute value. If you select to enter the absolute value, you can select the units from the drop-down menu.

5. Set (*Prism Layer Option*) to the desired value:
   - **Inherit Values**: for this option, prism layers on the surfaces of the part(s) will be generated based on settings taken from the parent mesh pipeline.
   - **Use Custom Values**: allows the prism layer parameters to be set directly on the part(s). See Prism Layer Panel.
   - **Disable**: no prism layer will be generated from the surfaces of the selected parts.
6. Click **OK**.

Generating the Mesh STAR-Creo

It is recommended that you first generate the surface mesh and inspect its quality before generating the volume mesh. Problems in the surface mesh will have a definite impact on the volume mesh, and may result in the volume mesher failing to complete.

Mesh generation will be carried out automatically when you issue the Build and Run command.

Generating the Surface Mesh in STAR-Creo

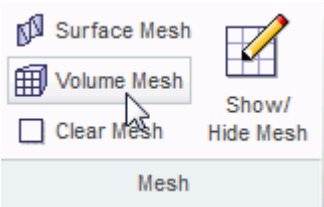
To generate the surface mesh:

- From the STAR-Creo ribbon, click **Surface Mesh**.

Generating the Volume Mesh in STAR-Creo

To generate the volume mesh:

- From the STAR-Creo ribbon, click **Volume Mesh**.

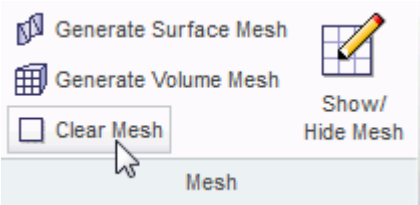


Clearing Generated Meshes

The **Clear Generated Meshes** action deletes all generated meshes without affecting the settings.

To clear the mesh:

- From the STAR-Creo ribbon, click **Volume Mesh**.



Contents:

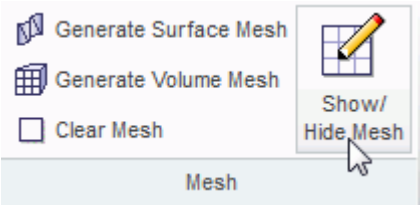
Visualizing the Mesh
Meshing Behavior at Interfaces

Visualizing the Mesh

Surface and volume meshes will automatically be displayed on first creation. If required, they can be hidden from view.

To hide a generated mesh in the geometry view:

1. From STAR-Creo the ribbon, click **Show/Hide Mesh**.



Meshing Behavior at Interfaces

The volume mesher will treat interfaces between two regions differently depending on what type of region (fluid or solid) these are.

If the interface lies between two fluid regions, no prism layers will be grown from both sides of the interface. If an interface lies between a solid region and a fluid region, prism layers will be grown from the interface. The default behavior of the prism layer generation can be overridden using a mesh control, so prism layers can be disabled on a solid-fluid interface. See Working With Mesh Controls.

See Also:

Working with Mesh Controls

Common Causes for Meshing Failures

There may be instances when the mesh generation fails. In most cases this will occur during the surface meshing phase. STAR-Creo will try and identify the cause and suggest ways of rectifying this. This information will be printed in the *Output* tab of the STAR-Creo Info App window.

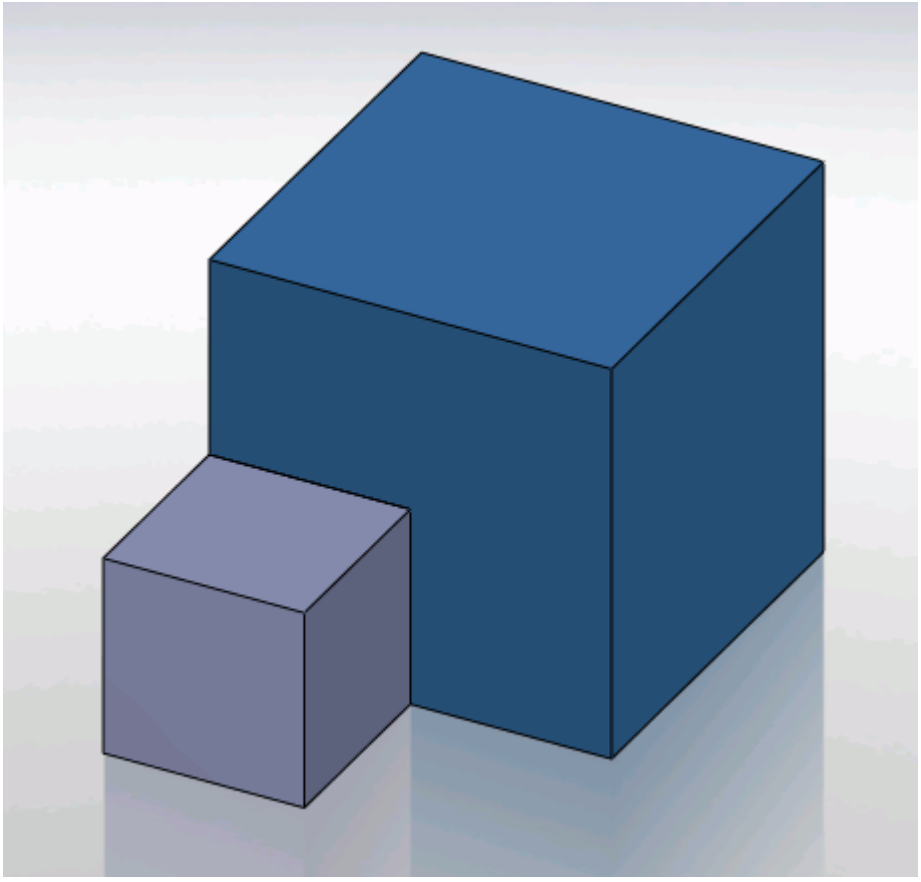
Contents:

The Parts in an Assembly are not Imprinted

The Parts in an Assembly are not Imprinted

The parts in an assembly should be imprinted together so that there is a common contact area between them.

A simple example of when imprinting is required is shown below.

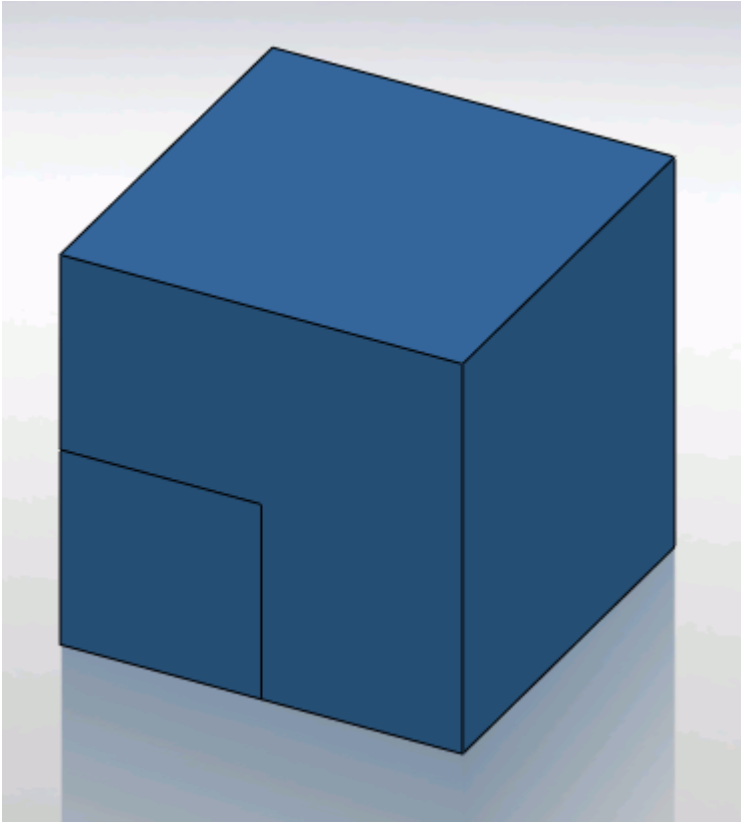


In this case the face of Part A (grey box) is splitting the face of Part B (blue box) causing this assembly to fail to mesh.

The output will display an error message.

```
Initializing meshing pipeline...
  All Geometry up to date.
Executing surface meshers...
Preparing surface for surface meshing in regions: Region 1, Region
Error during surface remeshing, deleting resulting mesh
An error occurred while attempting to orient the surface.
This is usually due to self-intersections in the surface. Check for problem(s) at :
  Geometric Range : (-0.0225836 0.00152109 0.0632873) and (0.0501103 0.0703362 0.0647127)
  Centroid of face : (0.0256414 0.0471603 0.064)
  Face type of face : 6
  Boundary = Default.
  Region = Region 1
Command: GenerateSurfaceMesh
   CompletedCommand: GenerateSurfaceMesh
   In: [Machine::main]
   Slave Message:
      Slave Name: Proc[0]
      Slave Rank: 0
   error: Server Error
```

The surface of Part B has to be imprinted with the contact face of Part A as shown below, after which meshing will complete successfully.



Obtaining a Solution

The following section details the execution of the STAR-CCM+ solvers in order to obtain a solution.

Contents:

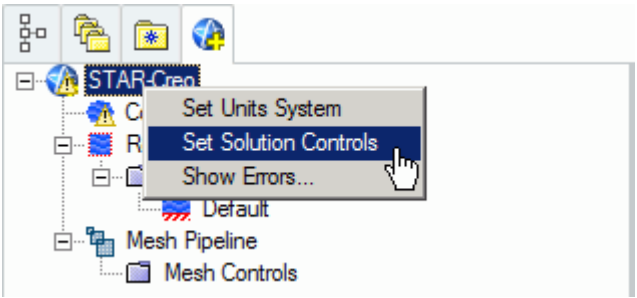
Setting Solution Controls
Checking the Model for Errors
Building and Running the Simulation in STAR-Creo
Clearing the Solution
Visualizing the Solution

Setting Solution Controls

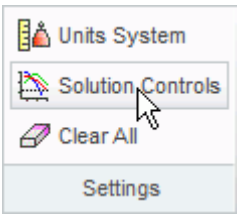
The solution you obtain from the solvers is a compromise between accuracy and processing time. You can manage the speed and accuracy of the solver from the *Solution Controls* dialog. The solution accuracy is also dependant on the mesh density.

To set up the solution controls in STAR-Creo:

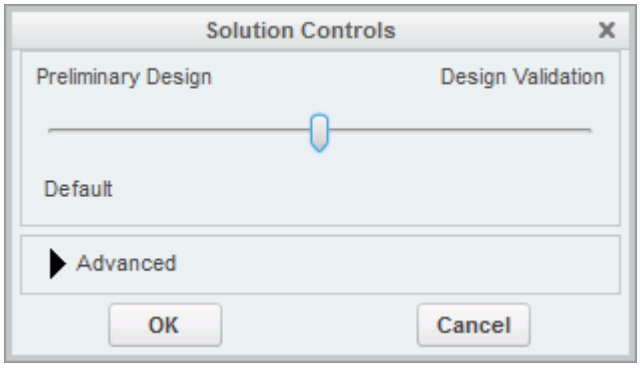
1. Either:
   - Right-click on the **STAR-Creo** node within the object tree and select **Set Solution Controls**.



   - From the STAR-Creo ribbon, select **Solution Controls**.



   The *Solution Controls* dialog appears.



2. Set the solution control parameters:
   - Using the slider, select one of the five pre-defined solver options. These options range from high accuracy and low speed, to low accuracy and high speed.
   - Click ▶ to expand the *Advanced Settings* tab and set the solution control parameters manually. See Solution Controls Advanced Settings Panel.
3. If you want to run a Java macro with the simulation, activate the **Use Java File** option. See Using Java Macros with STAR-Creo.
4. Click **OK**.

Contents:

Solution Controls Advanced Settings Panel
Using Java Macros in STAR-Creo

Solution Controls Advanced Settings Panel

The advanced settings in the Solution Control dialog allow you to manually enter the values for the solution control parameters.

Parameter	Description
Stopping Criteria	
(*Number of Iterations*)	Specifies the maximum number of iterations for which the solver can run. When the STAR-CCM+ solver reaches the specified number of iterations, it will stop. You can then post-process the results in STAR-CCM+. See Convergence Levels.
(*Convergence Level*)	Specifies how many orders of magnitude the residuals should drop before the solver stops. See Convergence Levels.
Solution Accuracy	
First Order	This option is generally the quicker, but less accurate solver option. The first order scheme introduces a dissipative error that is stabilizing and helps the

Parameter	Description
	solver achieve robust convergence. However, it can also lead to the neglect and merging of discontinuities, meaning the solution is not fully captured.
Second Order	This option is generally the more accurate, but slower solver option. The second order scheme attempts to minimize both dispersive and dissipative errors, providing a better representation of the simulation. However, in some situations, the lack of numerical dissipation can result in poorer convergence properties. It is common practice to use the first order scheme initially to ensure the simulation is running as expected and then switch to second order.
Under-Relaxation Factors	
(*Momentum Under-Relaxation Factor*)	Specifies the under-relaxation factor for the momentum equation. See Solution Controls Advanced Settings Panel.
(*Pressure Under-Relaxation Factor*)	Specifies the under-relaxation factor for the pressure-correction equation. See Solution Controls Advanced Settings Panel.
(*Turbulence Under-Relaxation Factor*)	Specifies the under-relaxation factor for the turbulence equation. See Solution Controls Advanced Settings Panel.
(*Fluid Temperature Under-Relaxation Factor*)	Specifies the fluid under-relaxation factor for the segregated energy solver. See Solution Controls Advanced Settings Panel.
(*Solid Temperature Under-Relaxation Factor*)	Specifies the solid under-relaxation factor for the segregated energy solver. See Solution Controls Advanced Settings Panel.

Convergence Level

The numerical method used by the solver requires an iterative process in order to obtain a solution. After each iteration, residuals are produced that indicate how well the governing equations for each solver quantity are being satisfied numerically. In general, the residual magnitude should decrease as the solution converges. When the magnitude of the residuals for all the quantities falls below the convergence level, the solver will stop iterating, and results will be exported for post-processing.

Note: In most cases, a drop of 3-5 orders of magnitude in the residuals is sufficient to obtain an accurate solution.

You can adjust the convergence level parameter to meet accuracy or process time requirements. However, if the residuals are no longer decreasing from one iteration to the next, it is likely that machine accuracy has

been reached for that case. For a case that does not exhibit decreasing residuals, it would be necessary to reconsider the setup in case the volume mesh is too coarse, or boundary conditions have been set incorrectly.

Under-Relaxation Parameters

Under-relaxation parameters determine how quickly the solver reaches a solution for a particular equation, on a scale from 0 to 1, with 1 being the fastest.

Ideally, all under-relaxation parameters would be set to 1, however, doing so would likely cause solver instability due to the complex nature of most simulations. Hence the under-relaxation parameters should be set to values that compromise speed with stability.

Note: Adjust the under-relaxation parameters if changes to other solution controls such as *Number of Iterations*, *Convergence Levels*, and *Solution Accuracy* have failed to improve the behavior of the solver.

Using Java Macros in STAR-Creo

You can use Java macros with STAR-CCM+ to automate your simulation or to change a setting in STAR-CCM+, such as selecting an alternative turbulence model, before the solver starts running.

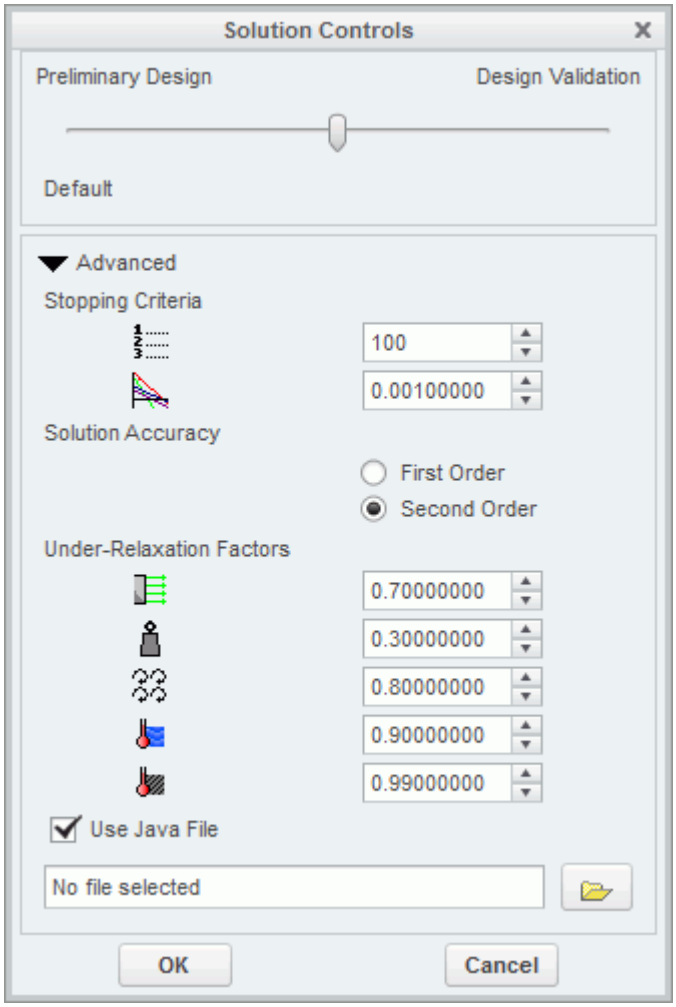
An example of how to change a setting in STAR-CCM+ using a Java macro is outlined below:

1. Launch STAR-Creo and set up the case.

   Define the continuum settings, boundary conditions, mesh settings, and solution control settings.
2. Launch STAR-CCM+.
3. In STAR-CCM+, start recording a macro and perform the desired operations, such as changing the physics models.
4. Stop recording the macro.

   The macro saves at this point.
5. Close STAR-CCM+ and do not save the simulation.
6. Return to STAR-Creo, open the *Solution Controls* dialog and activate the *Use Java File* checkbox.

   A text entry field appears.



7. Click (**Select Input File**).

   The *Open* dialog appears.

8. Navigate to your Java macro file and select it.

   Selected files appear in the text entry field. The path in the entry field is displayed relative to the current CAD geometry file.

9. Click **OK**.

When you run the simulation, STAR-CCM+ will launch and play the macro before starting the solver. This sequence allows you to set up and use additional STAR-CCM+ physics models that are not available in STAR-Creo.

See Also:

Setting Up Continua

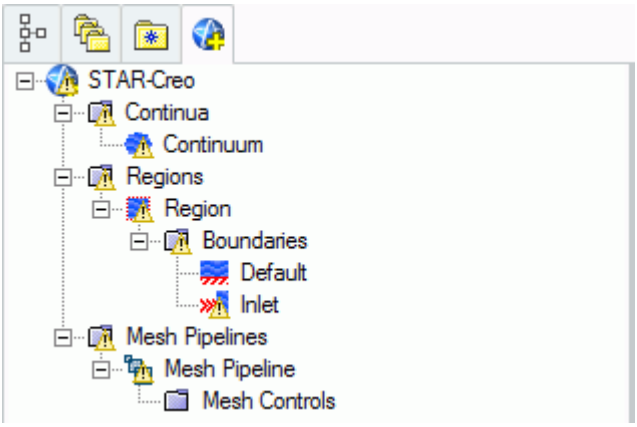
Setting Up Regions

Setting Boundary Conditions

Setting Up the Mesh Pipeline

Checking the Model for Errors

STAR-Creo monitors the active case for any errors in real-time and displays a warning triangle ⚠ next to any nodes for which there is a problem.

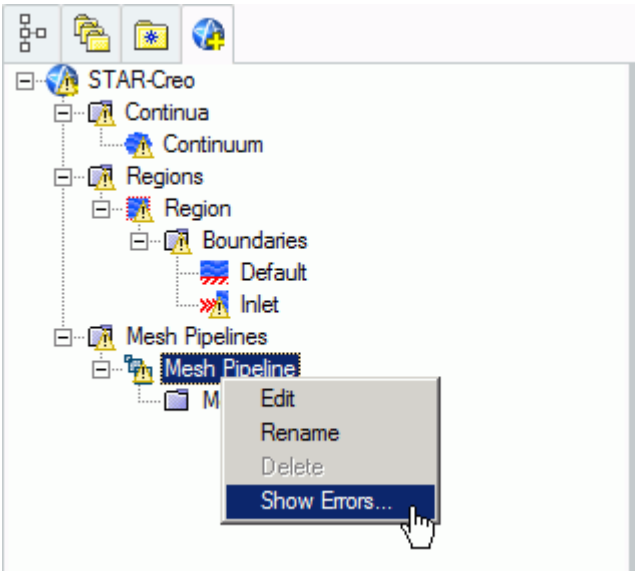


Three areas are monitored by the live model checker:

- **Continua**. The continua checker makes sure that all necessary boundaries are present in the case, such as the presence of at least one inlet and one outlet.
- **Regions and boundaries**. The regions and boundaries checker covers issues specific to boundaries, such as a velocity inlet boundary having a zero value.
- **Mesh pipelines**. The mesh pipeline checker covers issues that are related to the mesh, such as the presence of multiple parts in a trimmed mesh pipeline.

To see the cause of an error:

1. Right-click on an node displaying the warning triangle, and select **Show Errors...**



A dialog appears describing the error, as shown below.

2. Click **OK** to close the dialog.

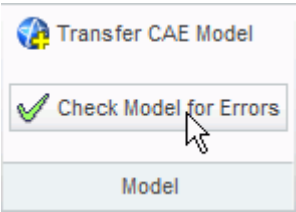
The **Show Errors...** right-click menu is available for any node in the STAR-Creo object tree that contains a warning triangle. Selecting it will show errors for that node and its sub-nodes.

To see a complete list of issues for the case, right-click on the STAR-Creo root node and select **Show Errors...**

You can also manually check for errors.

To perform a manual check:

- From the STAR-Creo ribbon, click **Check Model for Errors**.



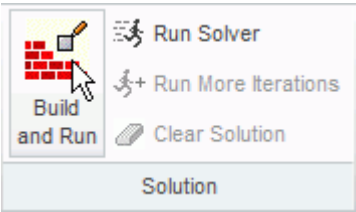
If errors are found in the case set up, these are displayed in the *Model Checker* tab of the *Info App* dialog. See The Model Checker Tab.

Building and Running the Simulation in STAR-Creo

Once you have defined the model physics, boundary conditions, mesh settings, and solver settings in STAR-Creo, you can build and run the case.

To build and run the case:

- From the STAR-Creo ribbon, click **Build and Run**.



This action automatically generates the mesh, and runs the solver. When you run the solver, a new *Residuals* tab is added to the *Info App* dialog. You can use this residuals plot to monitor solution convergence. When the solver finishes, STAR-CCM+ launches and a scalar scene is automatically created for you.

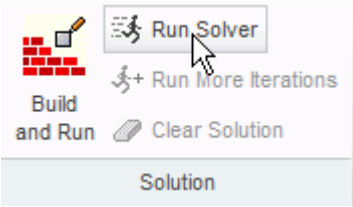
Note: If a solution has already been computed in STAR-CCM+, and the stopping criteria has been reached, clicking **Build and Run** clears the solution and starts running from the first iteration. If you want to run for more iterations, change the stopping criteria and click **Run Solver**, or click **Run More Iterations**.

Running the Simulation in STAR-Creo

If you have already generated a volume mesh for the model, you can run the case directly.

To run the solver:

- From the STAR-Creo ribbon, click **Run Solver**.

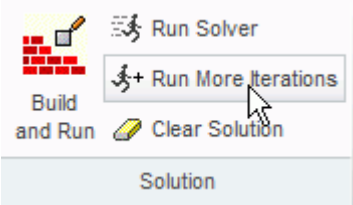


Running for More Iterations

You may need to run the STAR-CCM+ solvers for more iterations than you originally set in the Number of Iterations stopping criterion. The Run More Iteration function runs the solver for an additional number of iterations as specified in the stopping criteria. Take, for example, a case where you have already run the simulation for 300 iterations. You specify 500 iterations in the *Number of Iterations* stopping criterion. When you next click **Run for More Iterations**, the simulation will run for 500 extra iterations, making 800 iterations in total. The solver runs until one of the stopping criteria is satisfied.

To run for more iterations:

- From the STAR-Creo ribbon, click **Run For More Iterations**.



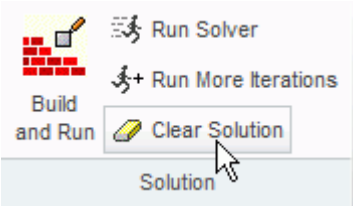
Clearing the Solution

You can clear all results that were calculated by the STAR-CCM+ solvers using the **Clear Solution** function. All other STAR-Creo simulation settings remain intact.

Caution: This action is irreversible.

To clear the solution:

1. From the STAR-Creo ribbon, select **Clear Solution**.



The solution that was calculated by the STAR-CCM+ solvers is deleted.

Visualizing the Solution

When STAR-CCM+ launches after the solution is calculated, it automatically displays a scalar scene of the model that shows the `Absolute Pressure` field function.

To visualize the results, you can use this scene to display different scalar quantities, or use any of the other visualization tools in STAR-CCM+, such as streamlines, plane sections and vector scenes.

You can import CAD parts into STAR-CCM+ for visualization purposes. These parts will not be taken into account during the analysis. This feature is useful for displaying the original geometry in a post-processing scene in STAR-CCM+. See Importing Geometry Into STAR-CCM+ for Visualization.

Exporting Files and Data

This section details the process for exporting files and data from STAR-Creo.

Contents:

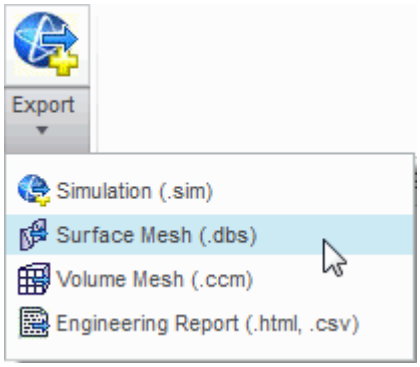
Exporting the Surface Mesh
Exporting the Volume Mesh
Exporting an Engineering Report
Exporting a STAR-CCM+ Simulation

Exporting the Surface Mesh

You can export the surface mesh generated in STAR-Creo as a native CD-adapco surface mesh (`.dbs`) file. The `.dbs` file format is the recommended format to use when transferring surface data between CD-adapco programs. The boundary names and boundary types are retained by the file, as well as the interfaces that were automatically created between regions.

To export the surface mesh:

1. From the STAR-Creo ribbon, click **Export > Surface Mesh (.dbs)**.

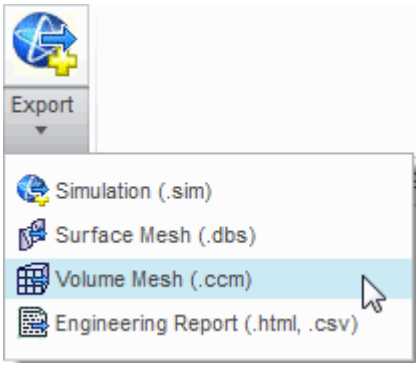


Exporting the Volume Mesh

You can export the volume mesh that you generated in STAR-Creo as a native CD-adapco volume mesh (`.ccm`) file. The `.ccm` file format is the recommended format to use when transferring volume meshes between CD-adapco programs. The region and boundary names, and region and boundary types are retained in this file. The interfaces that were automatically created interfaces between regions are also retained in this file.

To export the volume mesh:

1. From the STAR-Creo ribbon, click **Export** > **Volume Mesh (.ccm)**.



Exporting an Engineering Report

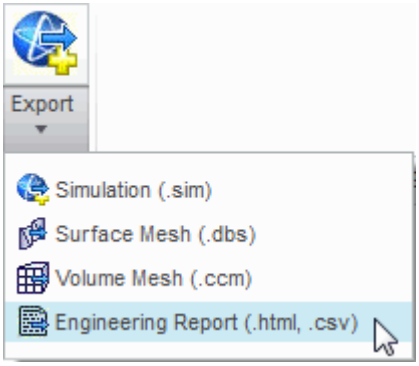
You can export a summary of the simulation settings and results in HTML format to two individual files:

- The first will be given a user-specified file name, and contains results from the simulation.
- The second will be given a user-specified file name appended with `_caseinfo`, and contains details on the simulation set-up.

Note: You can only export an engineering report after you obtain a solution.

To export an engineering report:

1. From the STAR-Creo ribbon, click **Export** > **Engineering Report (.html,.csv)**.



Results Report

The following is a summary of the information included in the results report (`<filename>.htm`). This data is provided in the report with reference to each material region in the simulation:

- Case name and time-stamp
- Solution information: CPU time, Elapsed Time and Number of Iterations
- Mass flow data, at each boundary
- Mass flux data, at each boundary
- Area averaged boundary data, for the following scalars:
  - Pressure (Pa)

  - Total Pressure (Pa)
  - Pressure Coefficient
  - Total Pressure Coefficient
  - Turbulent Kinetic Energy (J/kg)
  - Turbulent Dissipation Rate (m2/s3)
  - Turbulent Viscosity (Pa-s)
  - Density (kg/m3)
- Mass flow averaged boundary data, for the following scalars:
  - Pressure (Pa)
  - Total Pressure (Pa)
  - Pressure Coefficient
  - Total Pressure Coefficient
  - Turbulent Kinetic Energy (J/kg)
  - Turbulent Dissipation Rate (m2/s3)
  - Turbulent Viscosity (Pa-s)
  - Density (kg/m3)
- Force data, at each boundary:
  - Net Force [Fx, Fy, Fz]
  - Pressure Force [Fx, Fy, Fz]
  - Shear Force [Fx, Fy, Fz]
- Force coefficient data, at each boundary:
  - Coefficient of Net Force [Cfx, Cfy, Cfz]
  - Coefficient of Pressure Force [Cfx, Cfy, Cfz]
  - Coefficient of Shear Force [Cfx, Cfy, Cfz]

Setup Report

The following is a summary of the information found in the simulation setup report (`<filename>_caseinfo.htm`):

- The parts used in the simulation, and their associated regions
- The surfaces for each part, and their associated boundaries
- The continua used in the simulation, with the following information:
  - Physics models applied to the continuum
  - Materials properties
  - Turbulence properties
  - Segregated flow properties
  - Reference values
  - Initial conditions
- The mesh pipelines used in the simulation, with the following information:
  - Polyhedral mesher properties
  - Prism layer mesher properties (if activated)

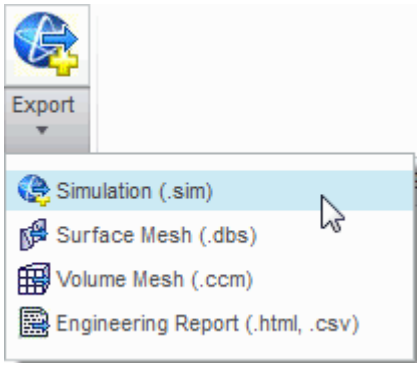
  - Surface remesher properties
  - Reference values
  - Volumetric controls
- The regions used in the simulation, and their associated continua and mesh pipeline
- The boundaries for each continua, with the applied conditions and values
- Solver parameters
- Stopping criteria
- Units

Exporting a STAR-CCM+ Simulation

You can export the current simulation in STAR-Creo to the native STAR-CCM+ simulation (`.sim`) file format. The `.sim` file format retains all simulation settings defined in STAR-Creo, as well as the surface and volume meshes.

To export a simulation file:

1. From the STAR-Creo ribbon, click **Export** > **Simulation (.sim)**.



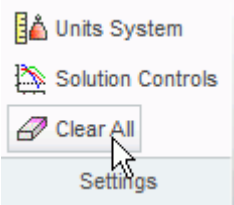
Resetting STAR-Creo Settings

STAR-Creo saves all case setup information to the CAD file. This means that all continuum settings, boundary settings, solution control settings, and mesh settings, are stored within the CAD geometry file. You can remove this information from the CAD geometry file.

Caution: This action is irreversible.

To clear all STAR-Creo settings:

1. From the STAR-Creo ribbon, click **Clear All**.



STAR-Creo Tutorials

Several tutorials are provided for the STAR-Creo CAD Client.

These tutorials provide an overview of the major features available in the client and illustrate some of the applications where you can use STAR-Creo.

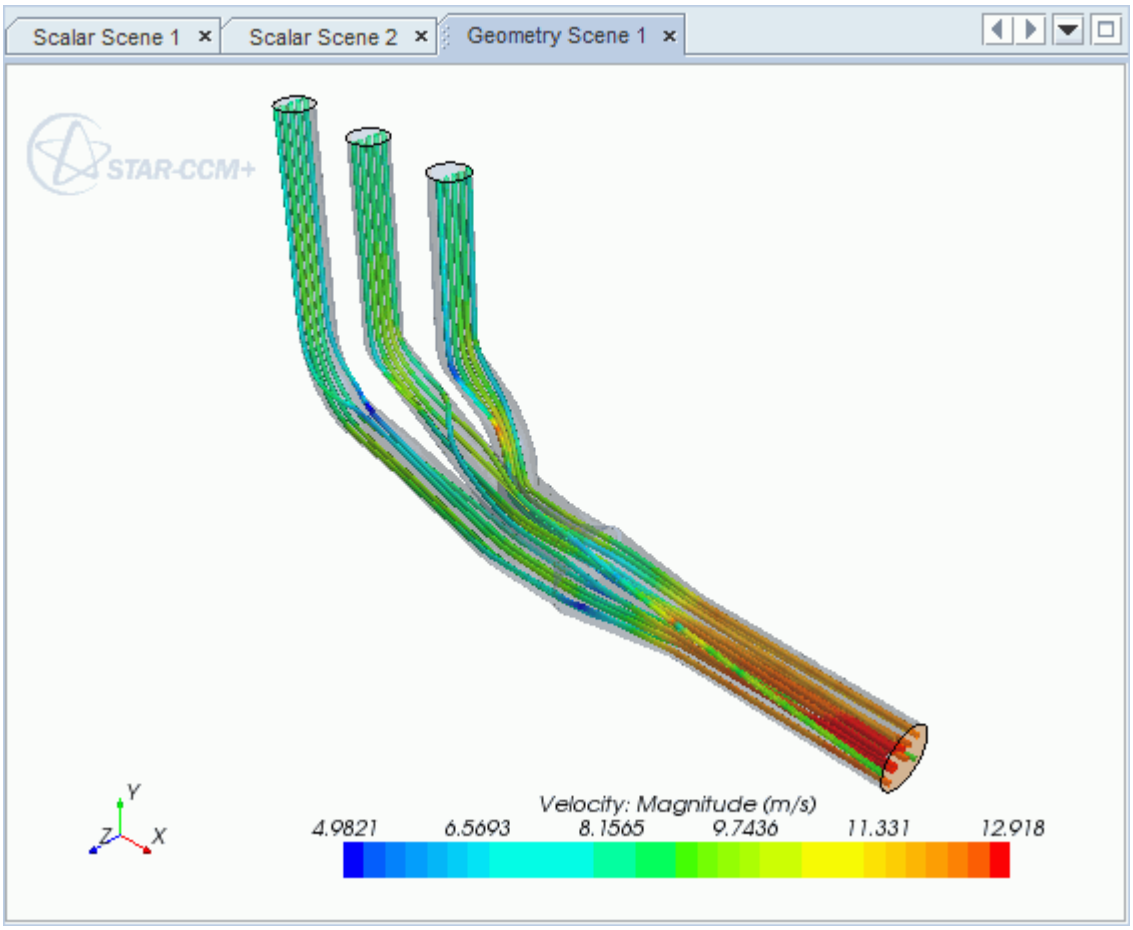
Contents:

Cold Flow Through a Motorcycle Exhaust Manifold
Porous Media Flow
Aircraft Wing Deicing System
Rotating Flow Inside a Graphics Card
External Flow Over a Car
Geometry Transfer: CAD to Simulation

Cold Flow Through a Motorcycle Exhaust Manifold

This tutorial demonstrates how to setup and run a CFD calculation in the STAR-Creo environment.

In this tutorial, the model is an exhaust manifold from a three-cylinder motorcycle engine.



Contents:

Opening the CAD Model
Selecting Fluid Physics Models
Defining Boundary Conditions
Setting the Mesh Pipeline
Modifying Solution Controls
Reviewing the Simulation Set Up
Generating the Surface Mesh
Generating the Volume Mesh
Building and Running
Post-Processing Results

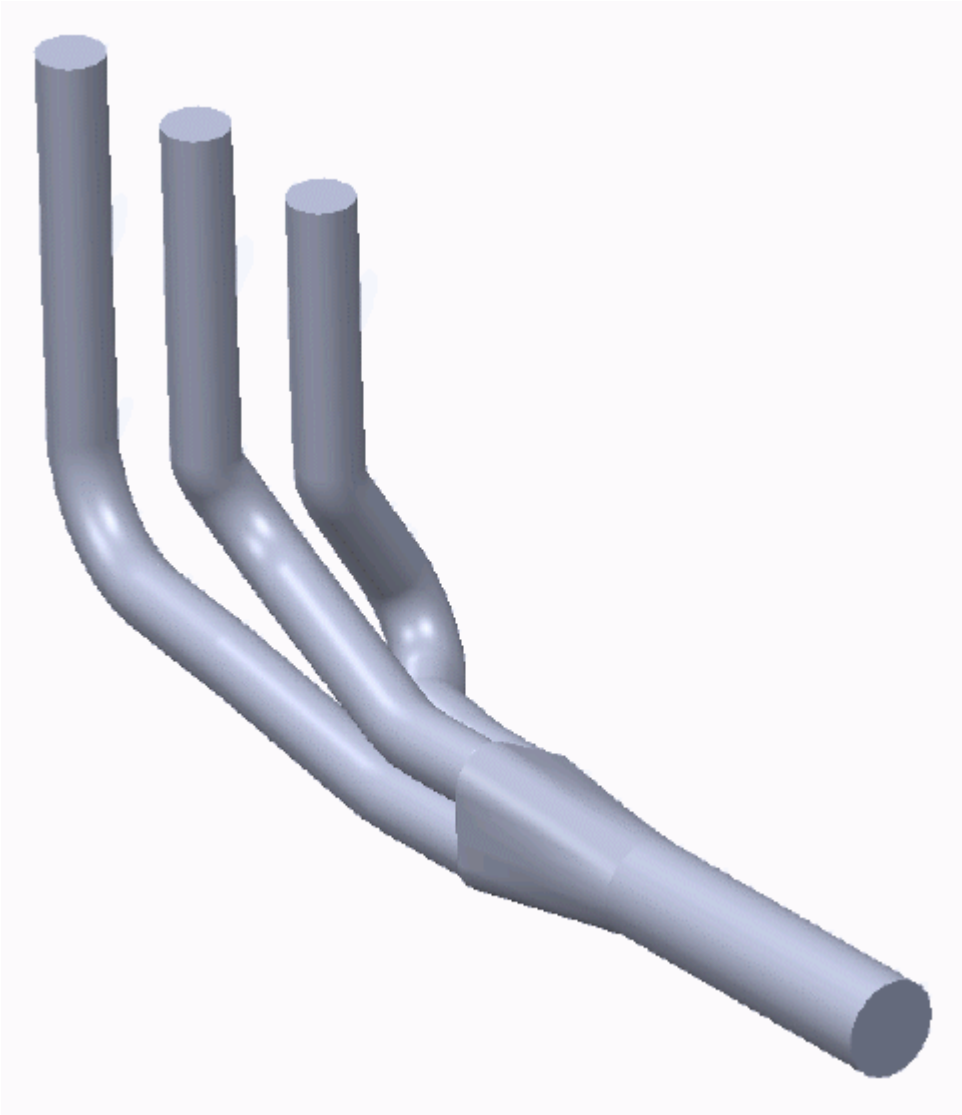
Opening the CAD Model

A directory containing the CAD geometry is provided with the STAR-CCM+ distribution.

This directory is located in `[STAR-CCM+_INSTALL_DIR]\doc\startutorialsdata\starcreo\data`.

1. Copy directory `data` to your own disk space and open the geometry file `exhaustmanifold.prt` in STAR-Creo.

   The imported geometry is shown below.



The STAR-Creo ribbon appears at the top of the screen.The STAR-Creo object tree is located on the left of the screen.

To switch to this object tree:

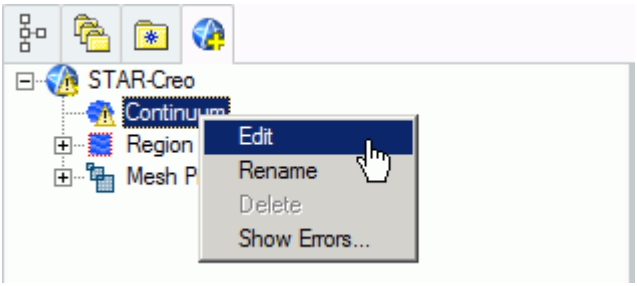
2. Click (**STAR Tree**).

If these items are not visible, refer to Launching STAR-Creo Manually.

Selecting Fluid Physics Models

The first step in setting up this tutorial is to define the physics of the problem. For this case, the flow is turbulent and isothermal.

1. Right-click the **Continuum** node in the STAR-Creo tree and select **Edit**.



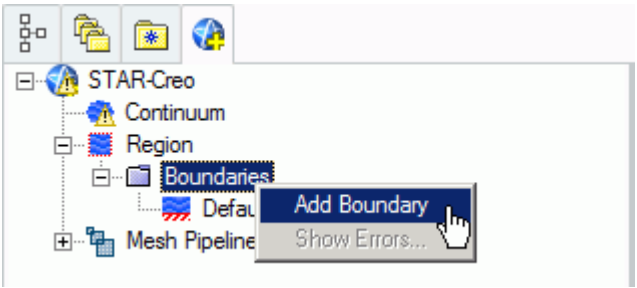
   The *Continuum Settings* dialog appears.
2. Deactivate the **Temperature** option in the *Optional Physics Models*.
3. Click **OK**.

Defining Boundary Conditions

You define the inlet and outlet boundary conditions for the simulation.

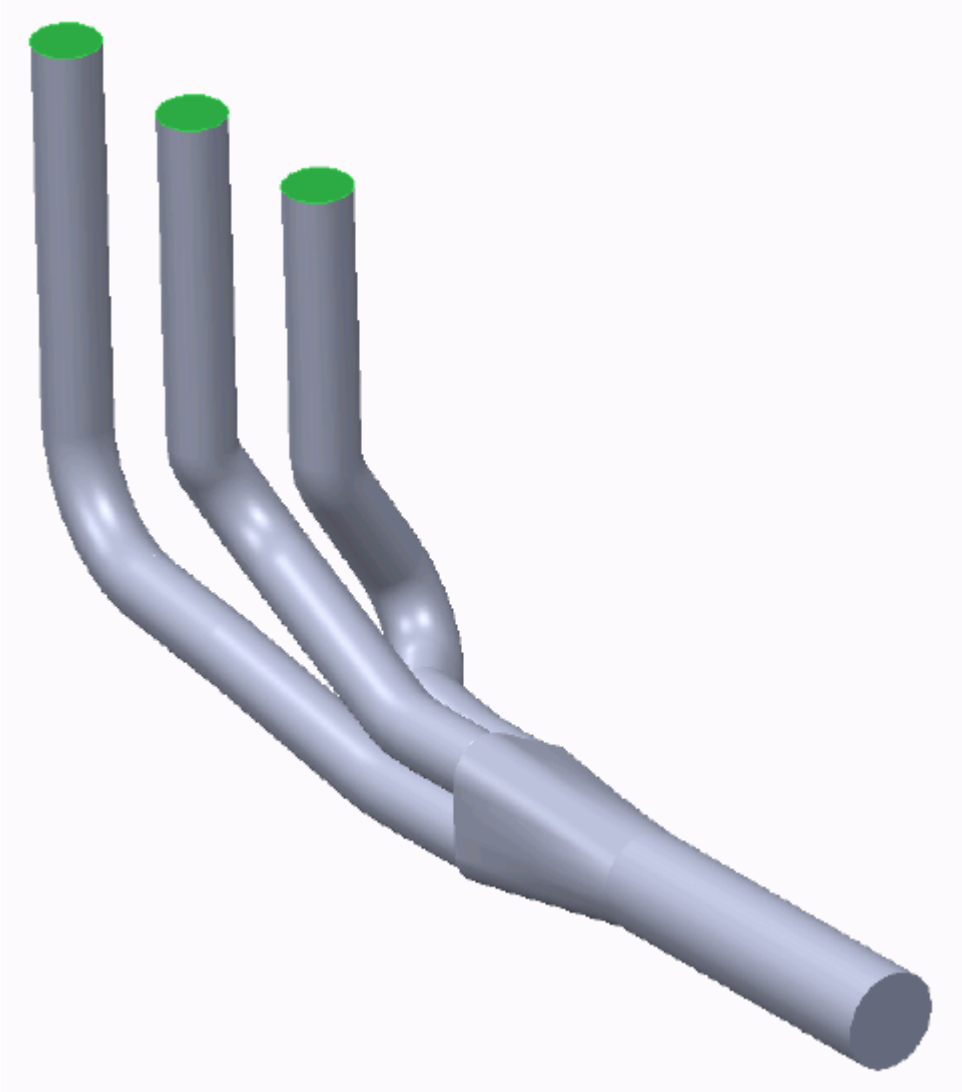
Create the inlet boundary first:

1. Right-click the **Region** > **Boundaries** node and select **Add Boundary**.



   The *Boundary Settings* dialog appears.

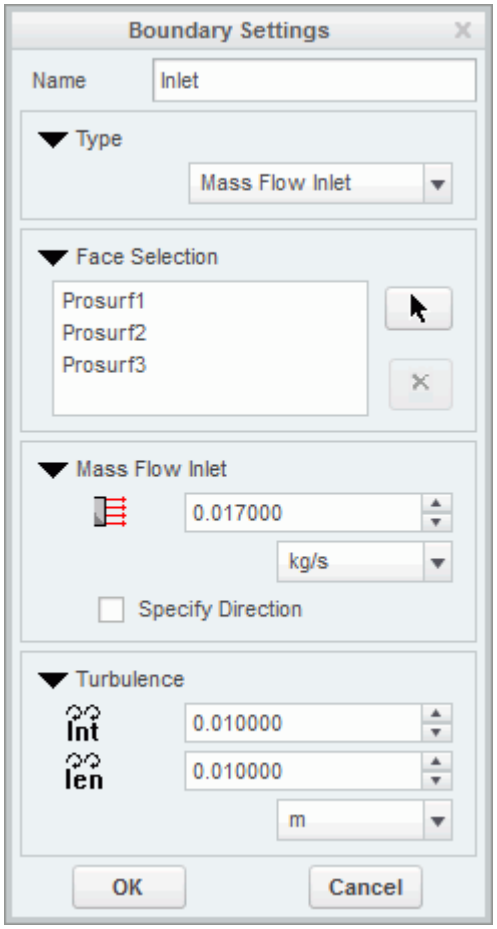
2. Click (**Select Face**) to activate the Creo face selection mechanism.
3. On the CAD model, select the three faces that are highlighted below.
   To select multiple faces, hold down the <`Ctrl`> key.



These selections are reported in the Included *Surfaces* box in the *Surface Sets* dialog.

4. Click **OK** to close the *Surface Sets* dialog.
5. Set *Type* to **Mass Flow Inlet**.
6. Set *Boundary Name* to `Inlet`.
7. Set (*Mass Flow Rate*) to `0.017` kg/s.

   The completed dialog is shown below.



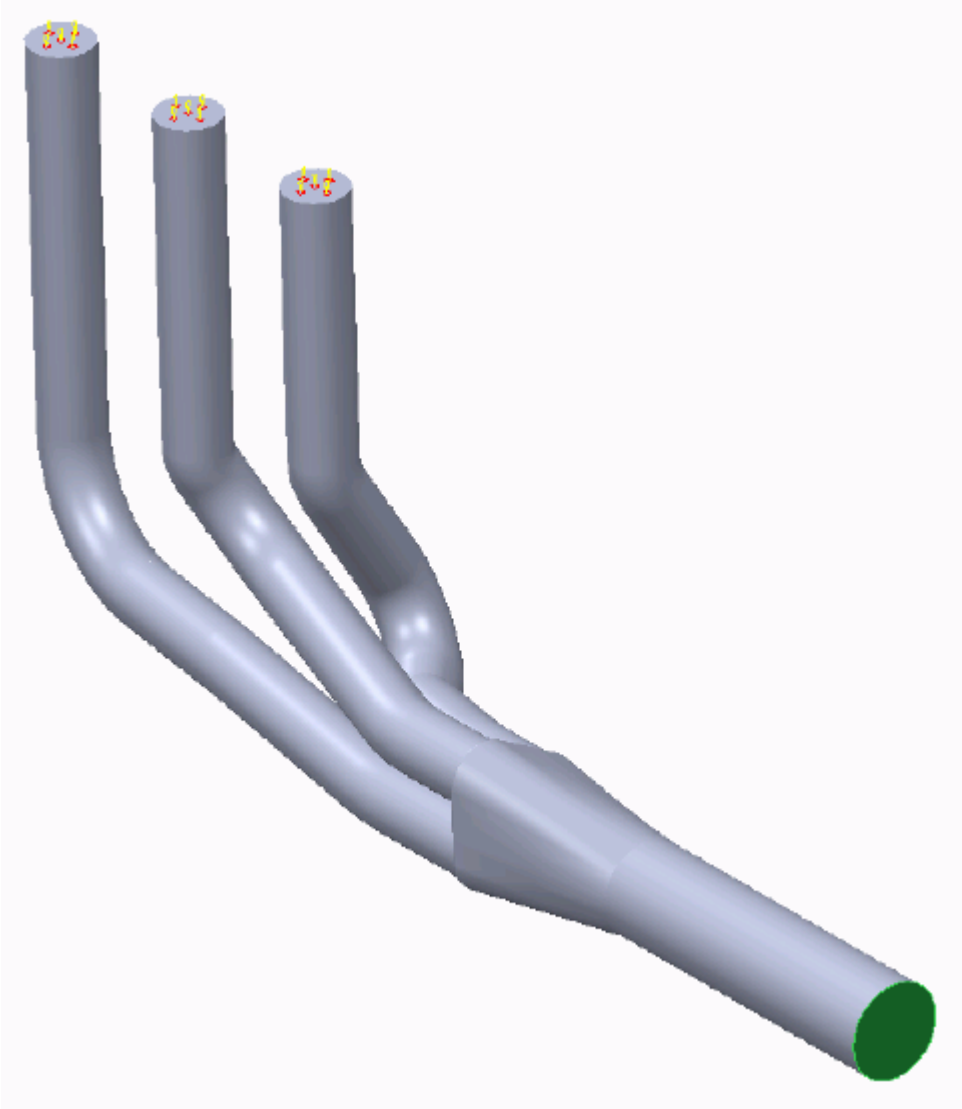
8. Click **OK**.

Create the outlet boundary:

9. Right-click the **Region** > **Boundaries** node and select **Add Boundary**.

   The *Boundary Settings* dialog appears.

10. Use the Creo face selection mechanism to select the face that is highlighted below.



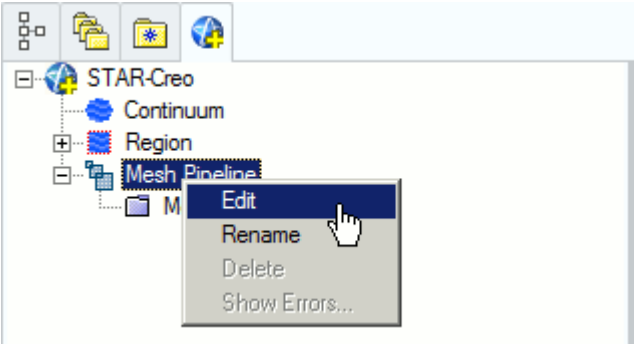
11. Set *Type* to **Pressure Outlet**.
12. Set *Boundary Name* to `Outlet`.

    The default settings are acceptable for this case.
13. Click **OK**.

Setting the Mesh Pipeline

To obtain more accurate results, modify the mesh pipeline:

1. Right-click the **Mesh Pipeline** node in the STAR-Creo tree and select **Edit**.



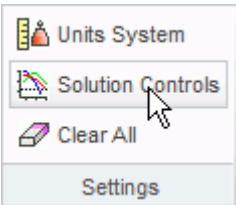
The *Mesh Pipeline Settings* dialog appears.

2. Set (*Base Size*) to `2.5`%.
3. Click to expand the *Prism Layer* tab.
4. Set (*Prism Layer Thickness*) to `15`%.
5. Click **OK**.

Modifying Solution Controls

Modify the maximum number of iterations and the solution accuracy:

1. Click **Solution Controls** on the STAR-Creo ribbon.

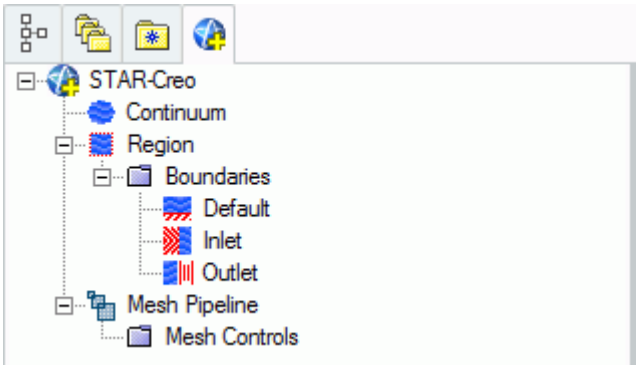


The *Solution Controls* dialog appears.

2. Click to expand the *Advanced* tab.
3. Set (*Number of Iterations*) to `200`.
4. Set (*Convergence Level*) to `0.0001`.
5. Click **OK**.
6. Save your progress.

Reviewing the Simulation Set Up

The final STAR-Creo object tree is shown below.

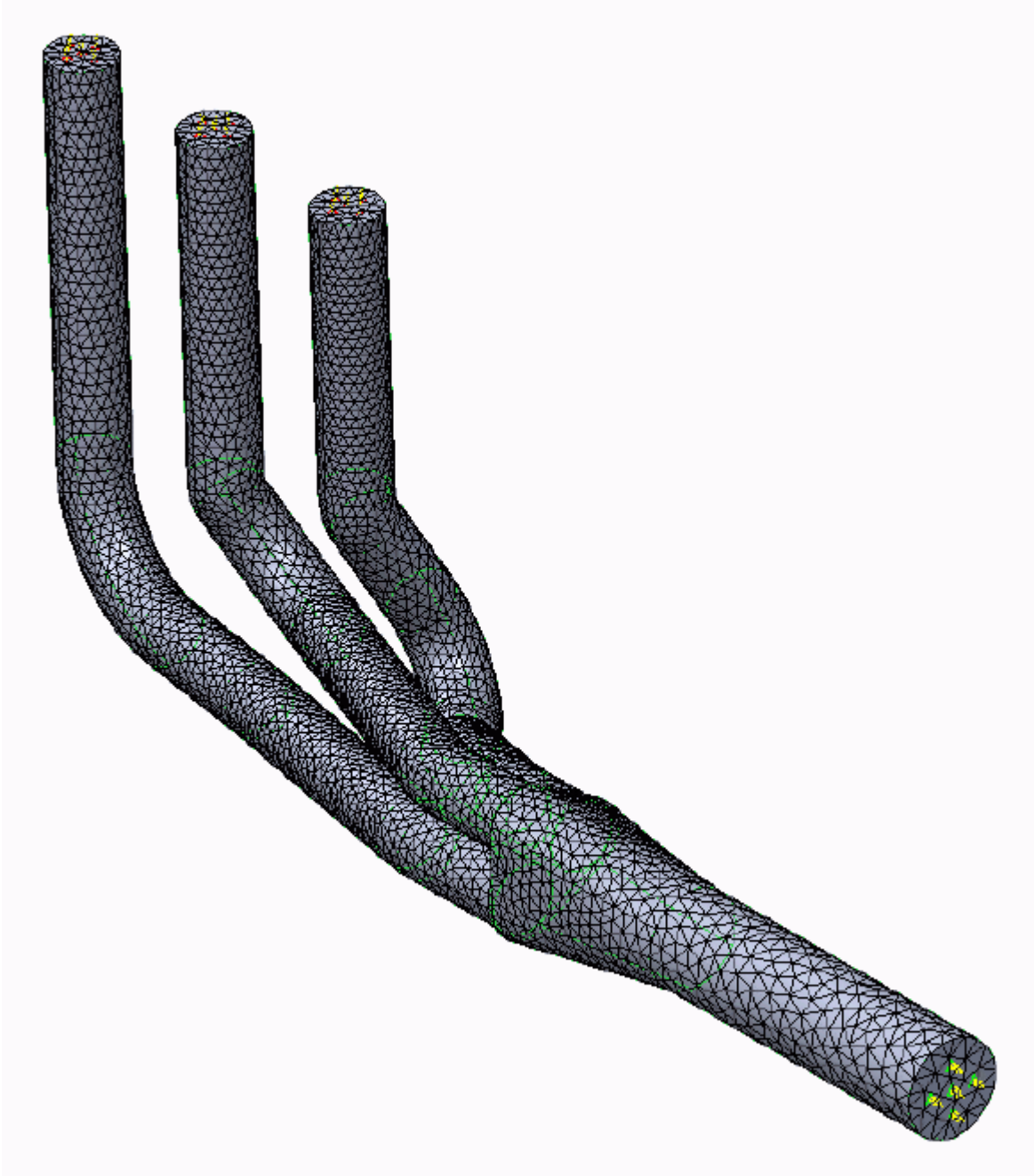


Generating the Surface Mesh

To generate the surface mesh:

1. From the STAR-Creo ribbon, click **Surface Mesh**.

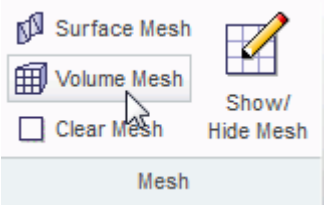
The surface mesh is shown below.



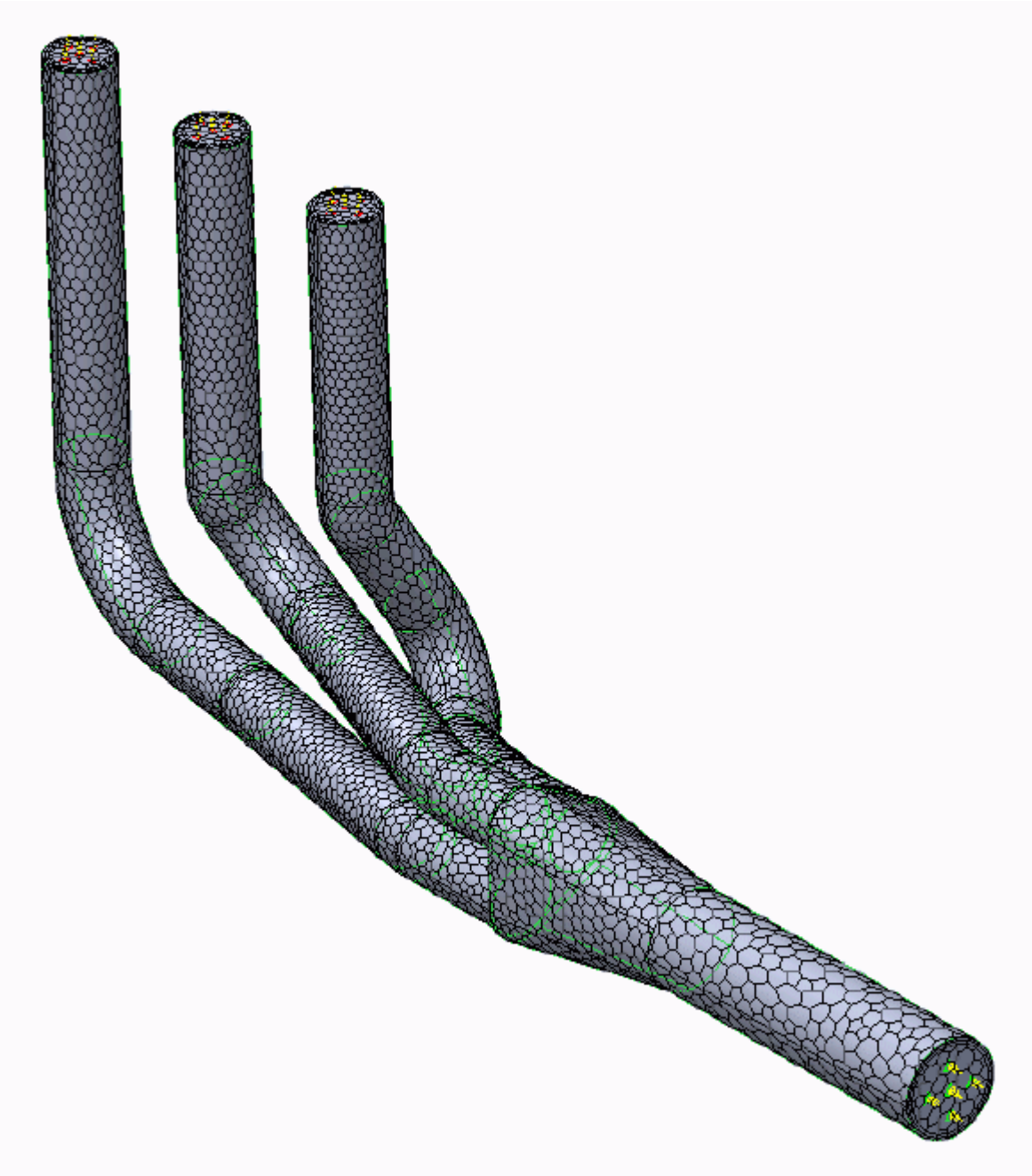
Generating the Volume Mesh

Generate the volume mesh:

1. From the STAR-Creo ribbon, click **Volume Mesh**.



The volume mesh is shown below.

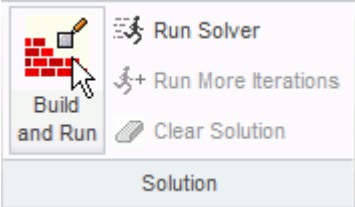


Building and Running

The case is now ready to run.

To run the simulation:

1. From the ribbon, click **Build and Run**.



The *Residuals* plot appears in the STAR-Creo *Output* window. You can use this plot to monitor the convergence of the solution.

When the solver is done, STAR-CCM+ launches.

Post-Processing Results

You perform all post-processing in STAR-CCM+. By default, a scalar scene, *Scalar Scene 1*, automatically appears in the *Graphics* window when the solver finishes. This scene displays the absolute total pressure.

The results that are contained in the following sections are representative. You may see some variation in the results that you obtain, depending on what platform and package you are using.

1. In STAR-CCM+, click **File** > **Save** to save the simulation.

Contents:

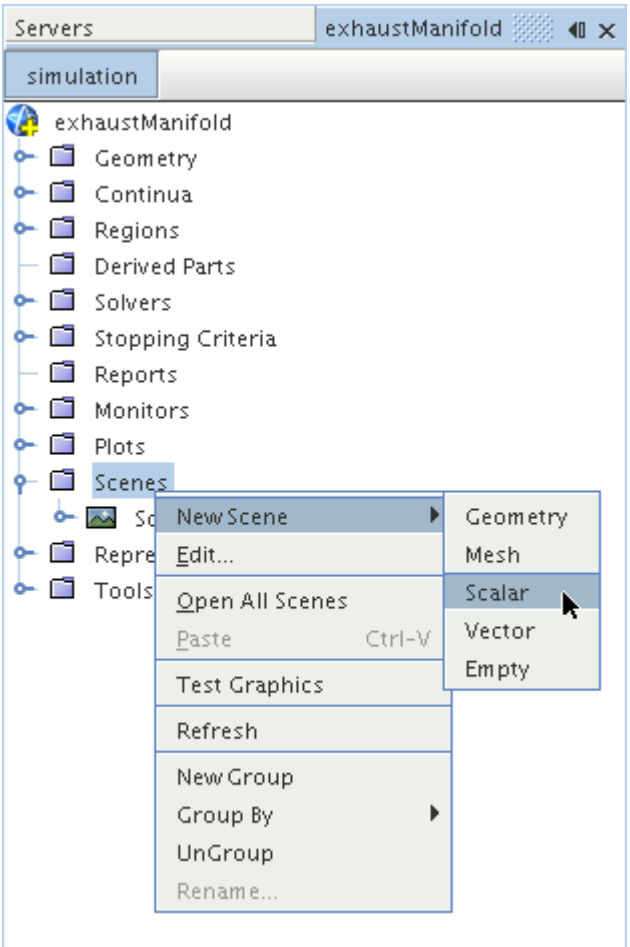
Creating a Scalar Scene
Creating A Streamlines Scene
Creating a Vector Scene
Creating a Streamlines Scene
Creating A Scalar Scene
Visualizing the Wing Skin Temperature
Creating a Streamlines Scene
Creating A Scalar Scene
Creating a Streamlines Scene
Creating a Scalar Scene
Modifying The Scalar Scene
Creating a Streamlines Scene

Creating a Scalar Scene

Create a scalar scene that displays the pressure on a cross-section of the exhaust manifold.

To create the scalar scene:

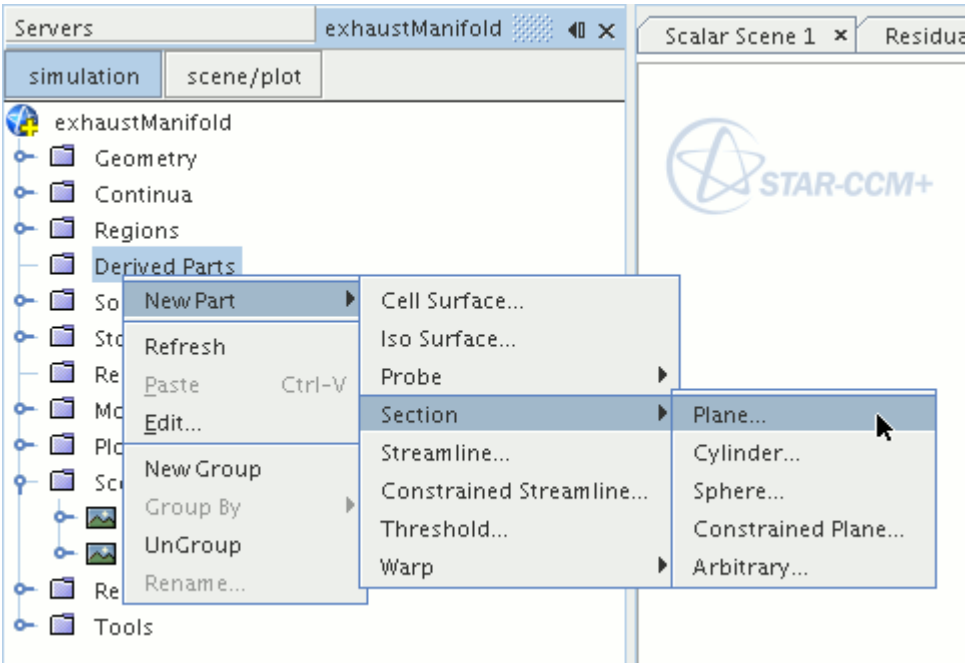
1. Right-click the **Scenes** node in the explorer pane and select **New Scene** > **Scalar**.



A new scalar scene, *Scalar Scene 2*, is created and selected in the *Graphics* window.

Create the section plane:

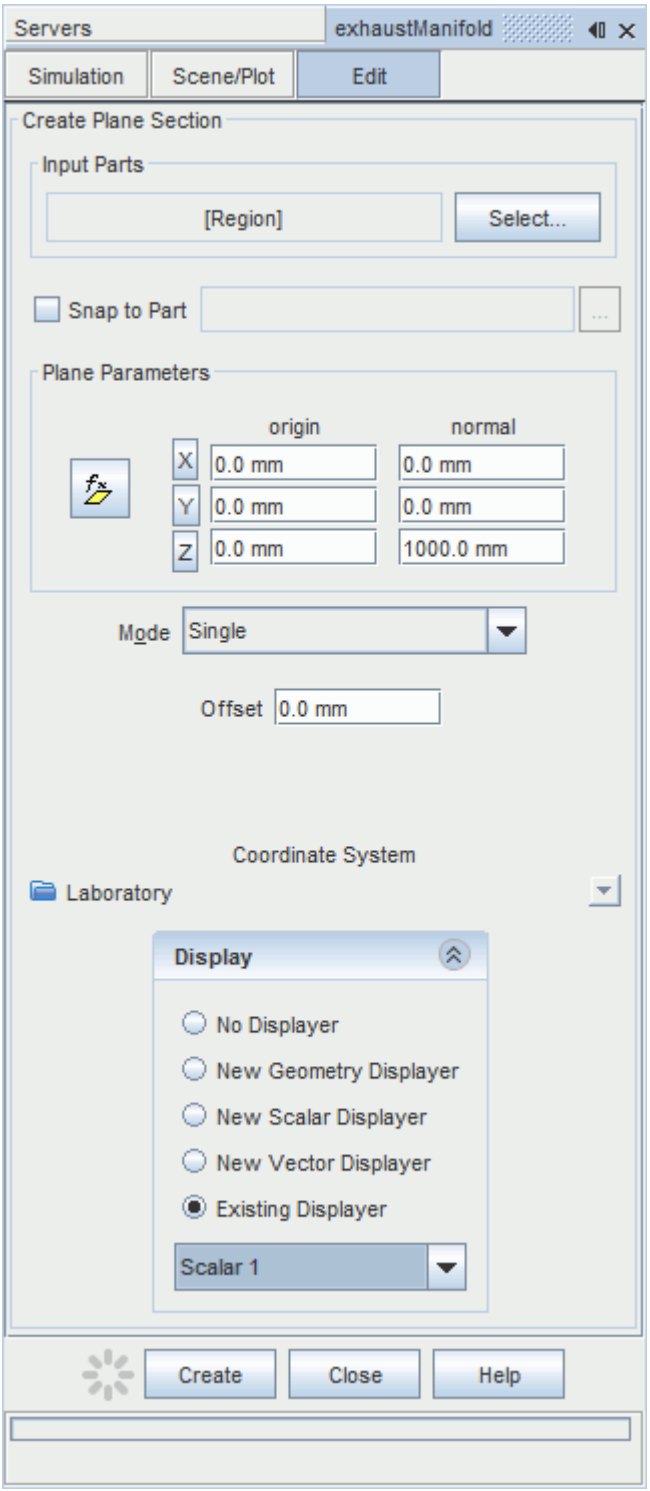
2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane...**



The *Create Plane Section* dialog appears.

3. In the *Plane Parameters* box, set the *origin* vector to `[0 mm, 0 mm, 0 mm]`
4. Set the *normal* vector to `[0 mm, 0 mm, 1000 mm]`
5. In the *Display* group box, select **Existing Displayer** and in the drop-down menu select **Scalar 1**.

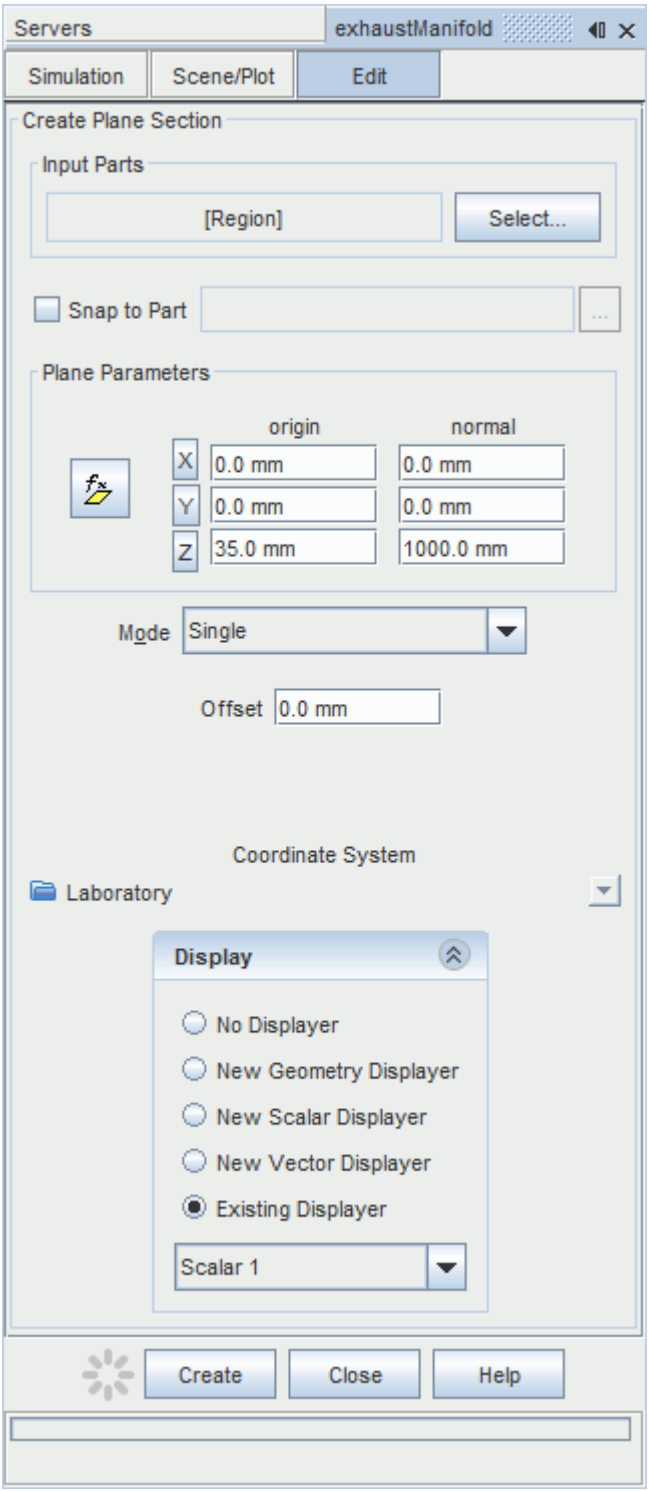
   The completed dialog is shown below.



6. Click **Create**.

Create the second section plane using the same dialog:

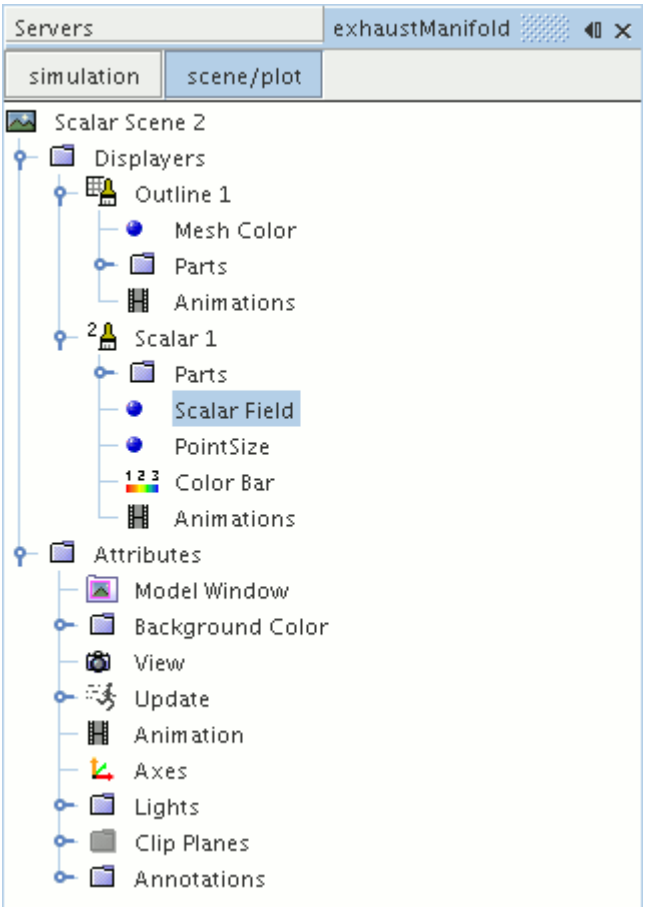
7. In the *Plane Parameters* box, set *origin* to `[0 mm, 0 mm, 35.0 mm]` and leave *normal* as `[0 mm, 0 mm, 1000 mm]`.



8. Click **Create** and then **Close**.

Define the scalar field to display the pressure contours:

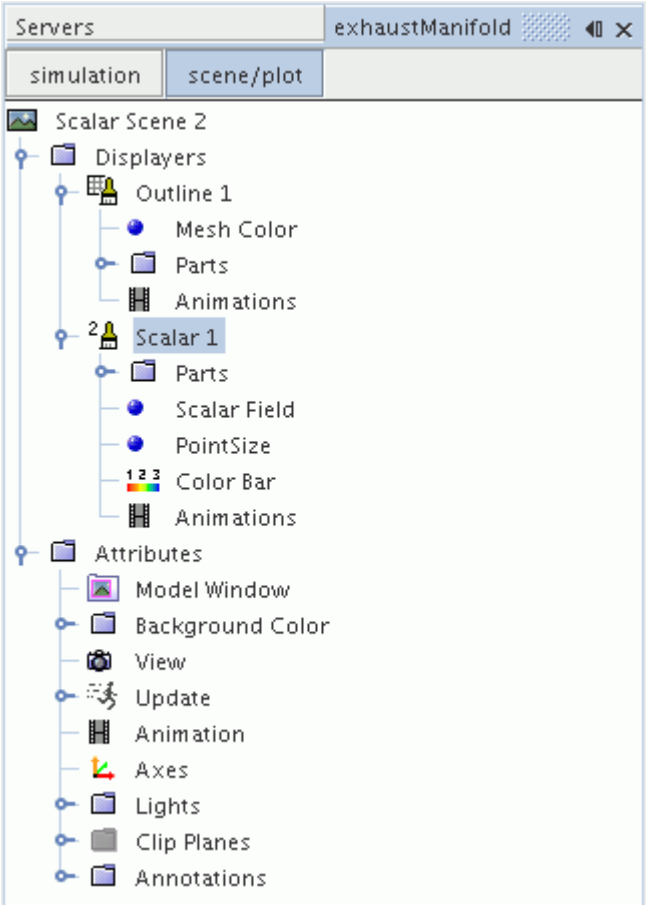
9. Click the **scene/plot** button in the explorer pane.
10. Select the **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Scalar Field** node.



11. Set *Function* to **Pressure**.

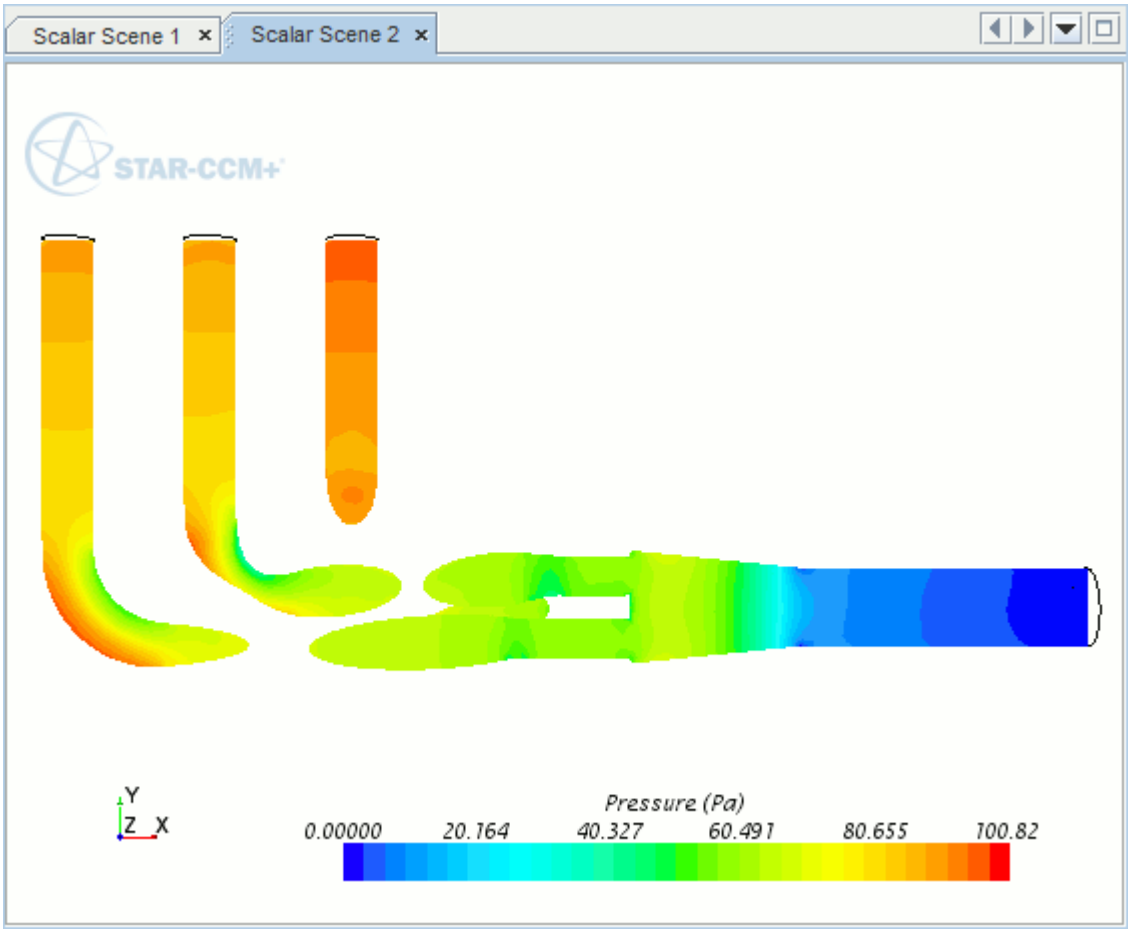
The pressure plot shows the pressure drop inside the exhaust manifold. In order to get a smooth transition between the different pressure values, set the contour style to smooth:

12. Select the **Scalar Scene 2** > **Displayers** > **Scalar 1** node.



13. Set *Contour Style* to **Smooth Filled**.
14. Make sure that the geometry is positioned on its side by pressing the <T> key while *Scalar Scene 2* is active.
15. Click the **simulation** button to return to the main STAR-CCM+ object tree.
16. Click (**Save**) to save the simulation .

The finished scene is shown below.

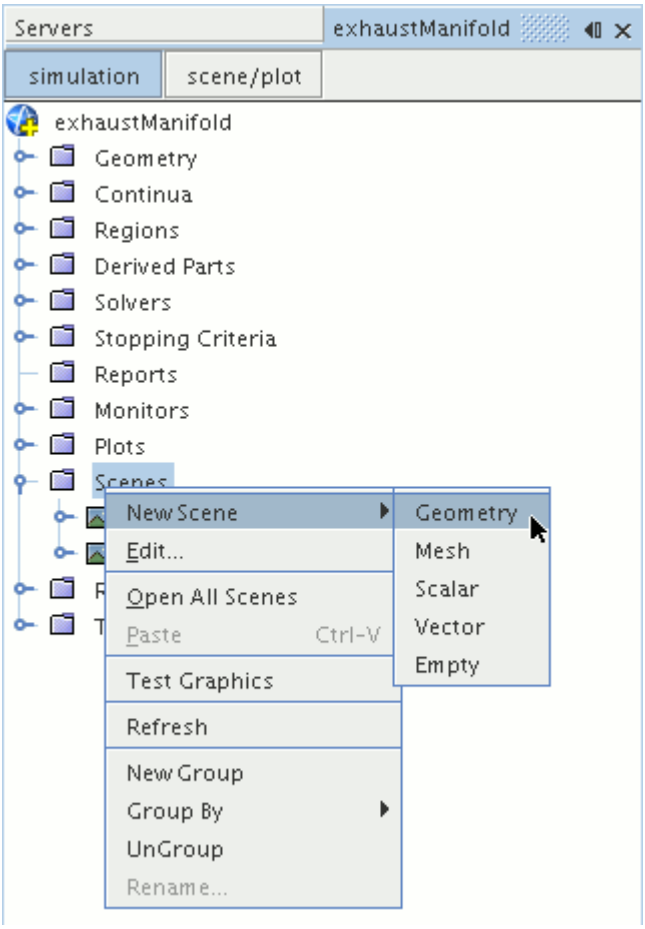


Creating A Streamlines Scene

Create a streamlines scene to get a better understanding of the flow inside the exhaust manifold.

First, create a geometry scene:

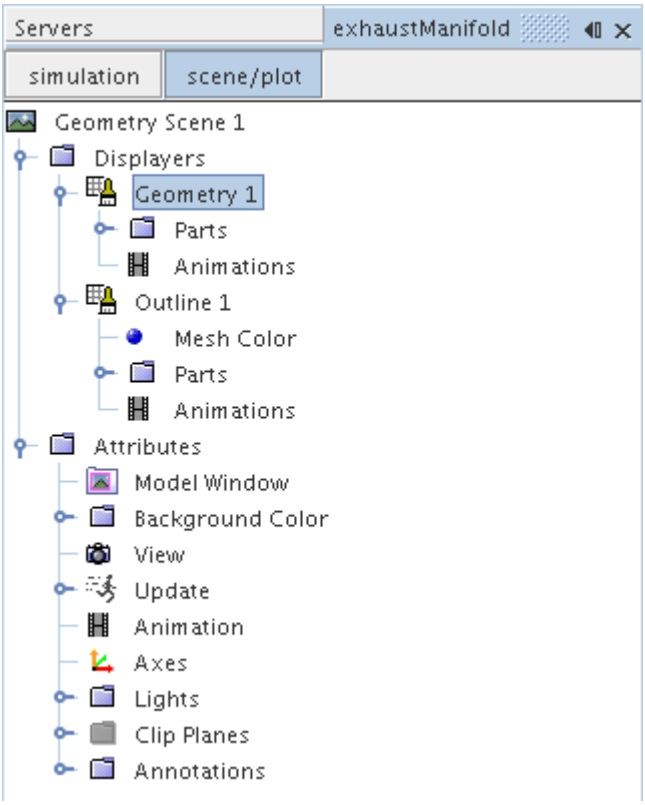
1. Right-click the **Scenes** node in the object tree and select **New Scene** > **Geometry**.



A new geometry scene is created and selected in the *Graphics* window.

Next, modify the opacity of the geometry model in the scene:

2. Click **scene/plot**.
3. Select the **Geometry Scene 1** > **Displayers** > **Geometry 1** node.



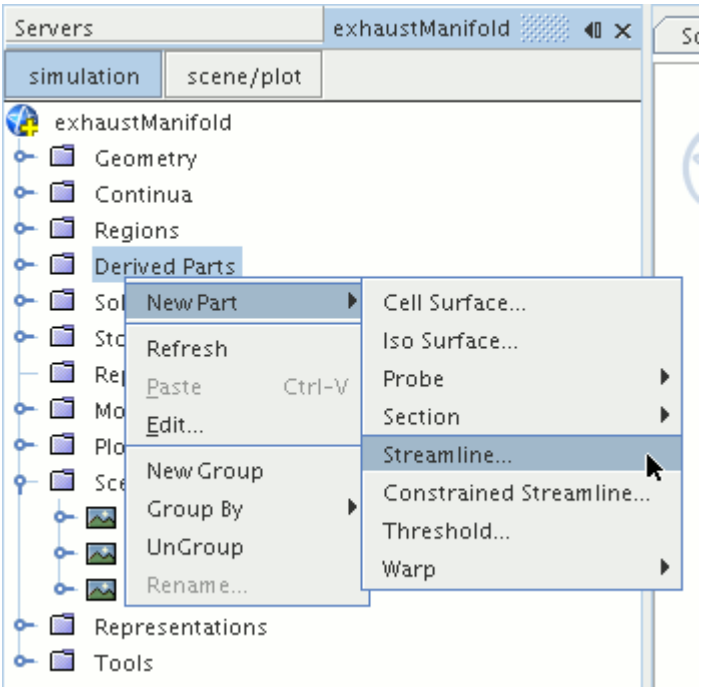
4. Set *Opacity* to `0.3`
5. Click **simulation**.

Change the view of the scene to something more suitable:

6. On the toolbar, click (**Save-Restore-Select views**) and select **Views** > **+X +Y +Z** > **Up +Y**.

Create the streamlines:

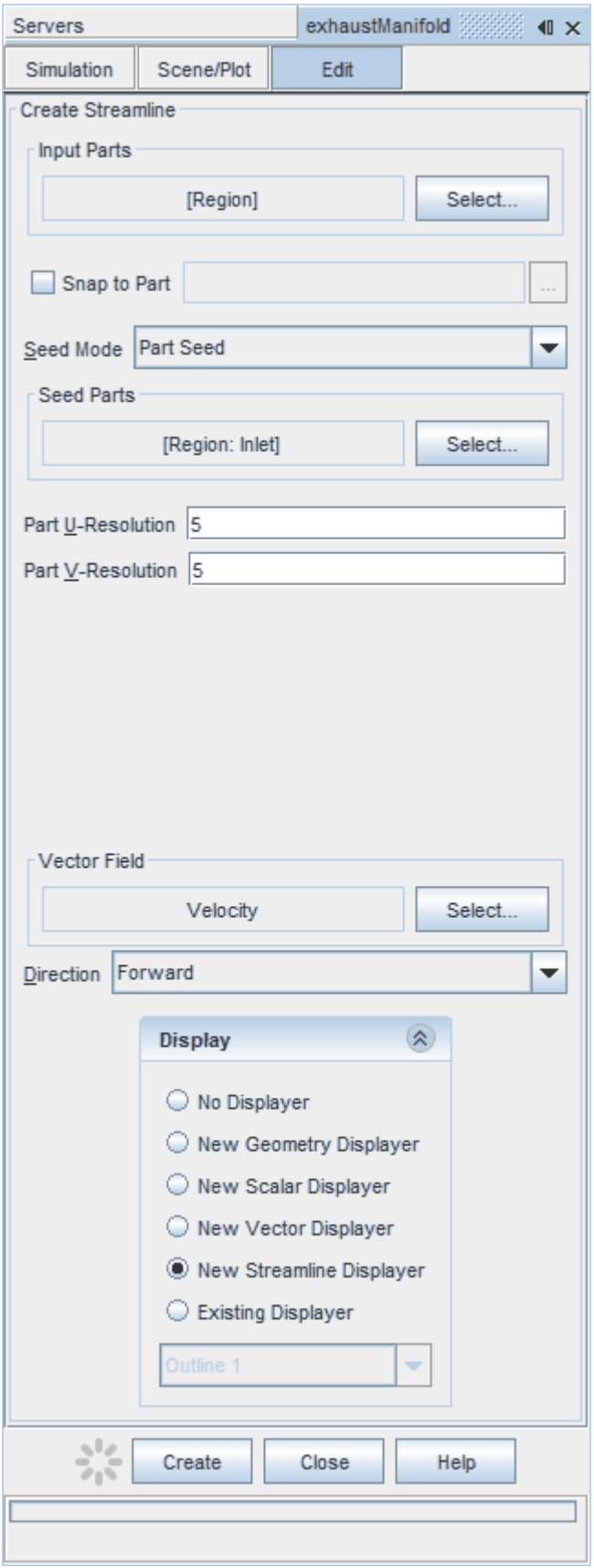
7. Right-click the **Derived Parts** node and select **New Part** > **Streamline...**.



The *Create Streamline* dialog appears.

8. Make sure that **Part Seed** is selected in the *Seed Mode* section.
9. Set *Seed Parts* to **Regions** > **Region** > **Boundaries** > **Inlet**.
10. Make sure that **New Streamline Displayer** is selected in the *Display* group box.

    The completed dialog is shown below.


Servers
exhaustManifold
Simulation
Scene/Plot
Edit
Create Streamline
Input Parts
[Region]
Select...
Snap to Part
Seed Mode
Part Seed
Seed Parts
[Region: Inlet]
Select...
Part U-Resolution
5
Part V-Resolution
5
Vector Field
Velocity
Select...
Direction
Forward
Display
No Displayer
New Geometry Displayer
New Scalar Displayer
New Vector Displayer
New Streamline Displayer
Existing Displayer
Outline 1
Create
Close
Help

11. Click **Create** to create the streamlines and add them to the geometry scene as part of a streamline displayer.
12. Click **Close** to close the *Create Streamline* dialog.

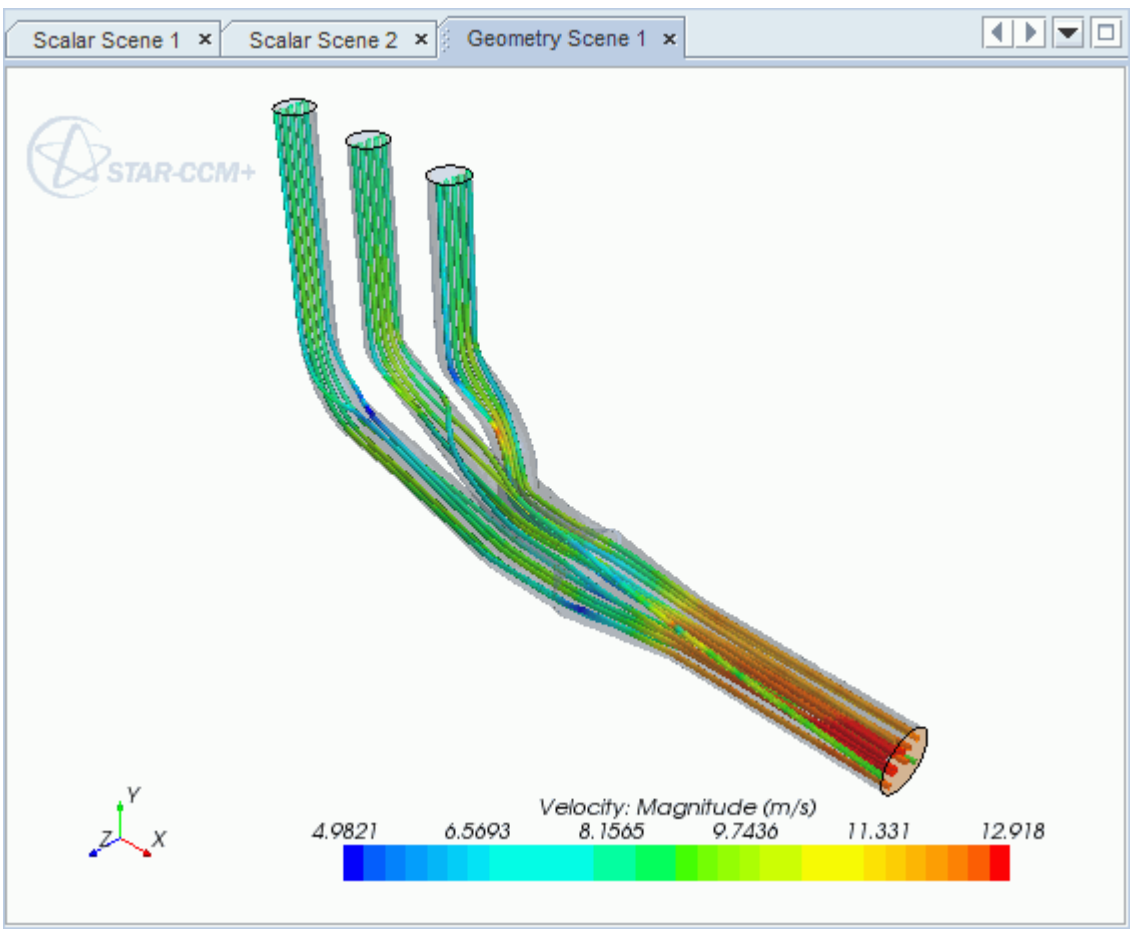
Change the display method of the streamlines to tubes instead of the default lines:

13. Click **scene/plot**.
14. Select the **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** node.
15. Set *Mode* to **Tubes** and *Width* to `0.0020`.

Change the color of the streamlines to represent velocity magnitude:

16. Select the **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** > **Scalar Field** node.
17. Set *Function* to **Velocity** > **Lab Reference Frame** > **Magnitude**.
18. Save the simulation.

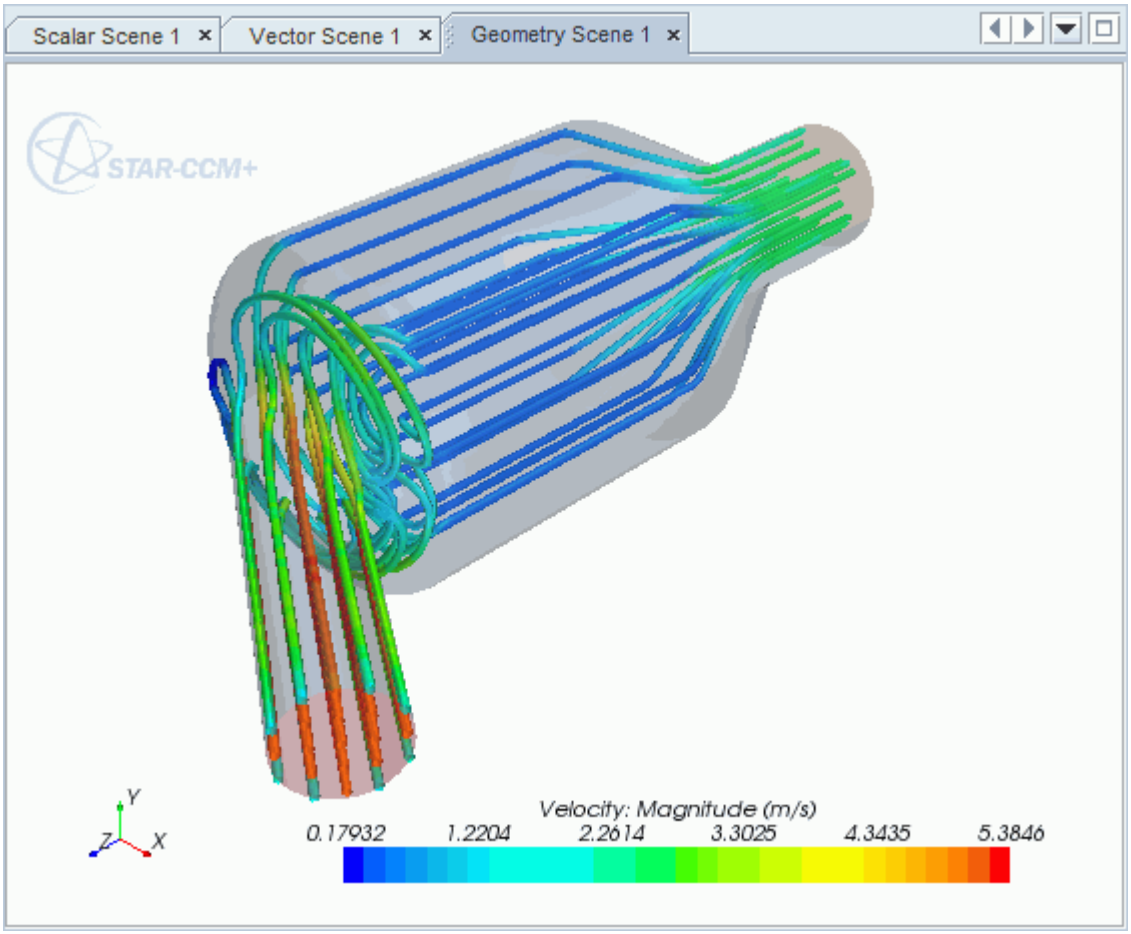
The finished scene is shown below.



Porous Media Flow

This tutorial takes you through the steps that are required to run a CFD calculation in the STAR-Creo environment.

In this tutorial, the model is an idealized catalyst.



Contents:

Opening the CAD Model
Selecting Fluid Physics Models
Setting the Porous Region
Defining Boundary Conditions
Setting the Mesh Pipeline
Modifying Solution Controls
Reviewing the Simulation Set Up
Generating the Surface Mesh
Generating the Volume Mesh

Building and Running
Post-Processing Results

Opening the CAD Model

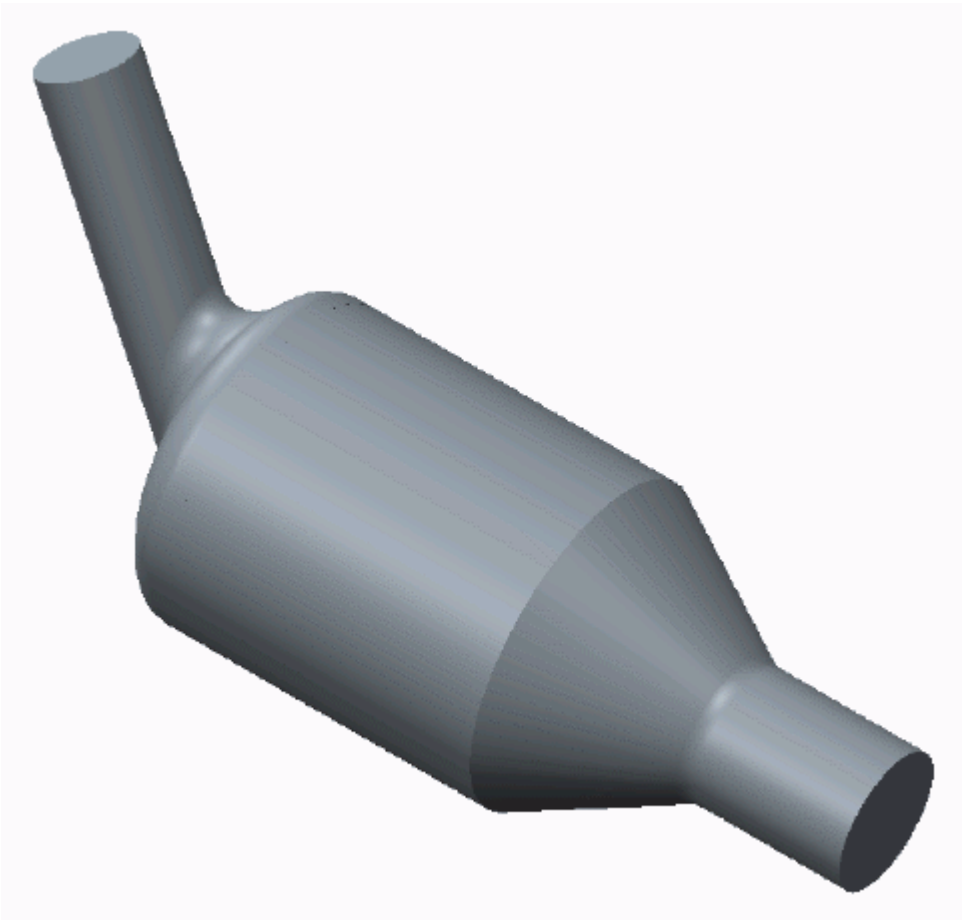
A directory containing the CAD geometry is provided with the STAR-CCM+ distribution.

This directory is located in `[STAR-CCM+_INSTALL_DIR]\doc\startutorialsdata\starcreo\data`.

1. Copy directory `data` to your own disk space and open the geometry file `porousmodel.asm` in STAR-Creo.

   The assembly is made up of three parts: the inlet region, the porous middle region, and the outlet region.

   The imported geometry is shown below.



The STAR-Creo ribbon appears at the top of the screen.The STAR-Creo object tree is located on the left of the screen.

To switch to this object tree:

2. Click (**STAR Tree**).

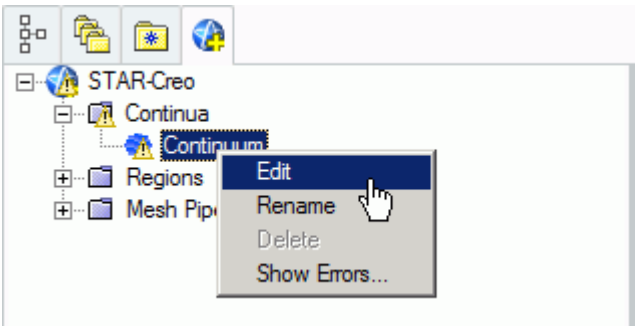
If these items are not visible, refer to Launching STAR-Creo Manually.

Selecting Fluid Physics Models

The first step in setting up this tutorial is to define the physics of the problem. For this case, the flow is turbulent and isothermal. Temperature calculations are on by default in STAR-Creo.

To turn temperature calculations off:

1. Right-click the **Continua** > **Continuum** node in the STAR-Creo tree and select **Edit**.



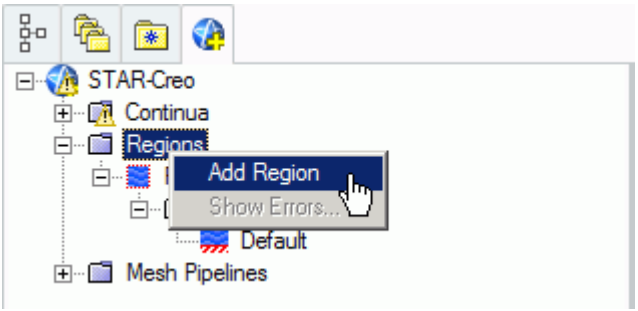
   The *Continuum Settings* dialog appears.

2. Deactivate the **Temperature** option in the *Optional Physics Models* box.
3. Click **OK**.

Setting the Porous Region

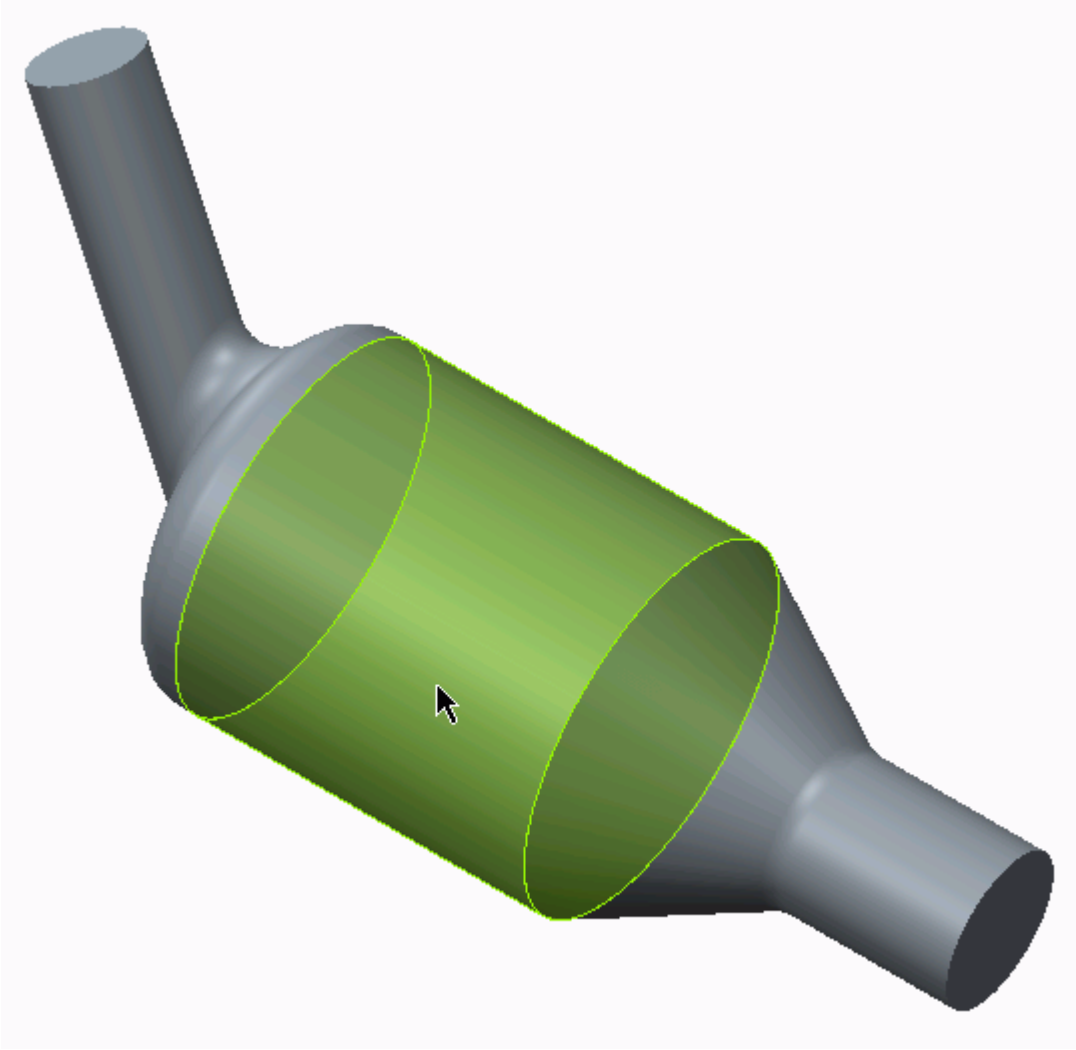
Create the porous region and define its properties:

1. Right-click the **Regions** node in the STAR-Creo object tree and select **Add Region**.



   The *Region Settings* dialog appears.

2. Click (**Select Part**) to activate the Creo part selection mechanism.
3. Select the middle part of the assembly as shown below.



4. Click **OK** to close the *Select* dialog.
5. Set *Name* to `Porous Region`.
6. Activate the *Porous* model and select the **Uni-Directional** porous type.
7. Set the following properties for the *Porous* region:

Parameter	Value
α (*Alpha*)	`25 kg/m^4`
β (*Beta*)	`1500 kg/m^3.s`
(*Cross-Flow Resistance Factor*)	`250`

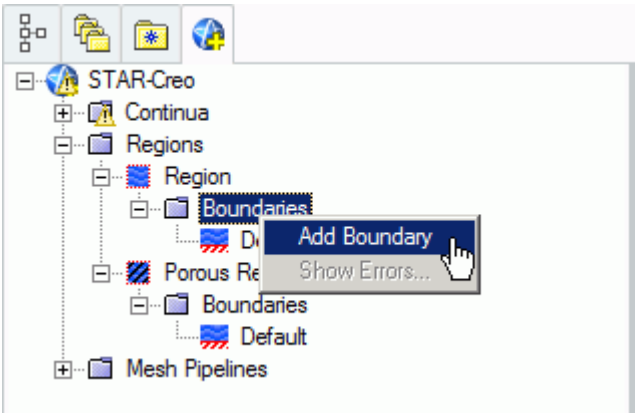
8. Click **OK**.

Defining Boundary Conditions

Create the inlet and outlet boundary conditions. STAR-Creo automatically adds interfaces between the different regions. These interfaces are visible in the final STAR-CCM+ simulation.

Create the inlet boundary first:

1. Right-click the **Regions** > **Region** > **Boundaries** node and select **Add Boundary**.



   The *Boundary Settings* dialog appears.

2. Click (**Select Face**) to activate the Creo face selection mechanism.
3. Select the face that is highlighted below.

These selections are reported in the *Included Surfaces* box in the *Surface Sets* dialog.

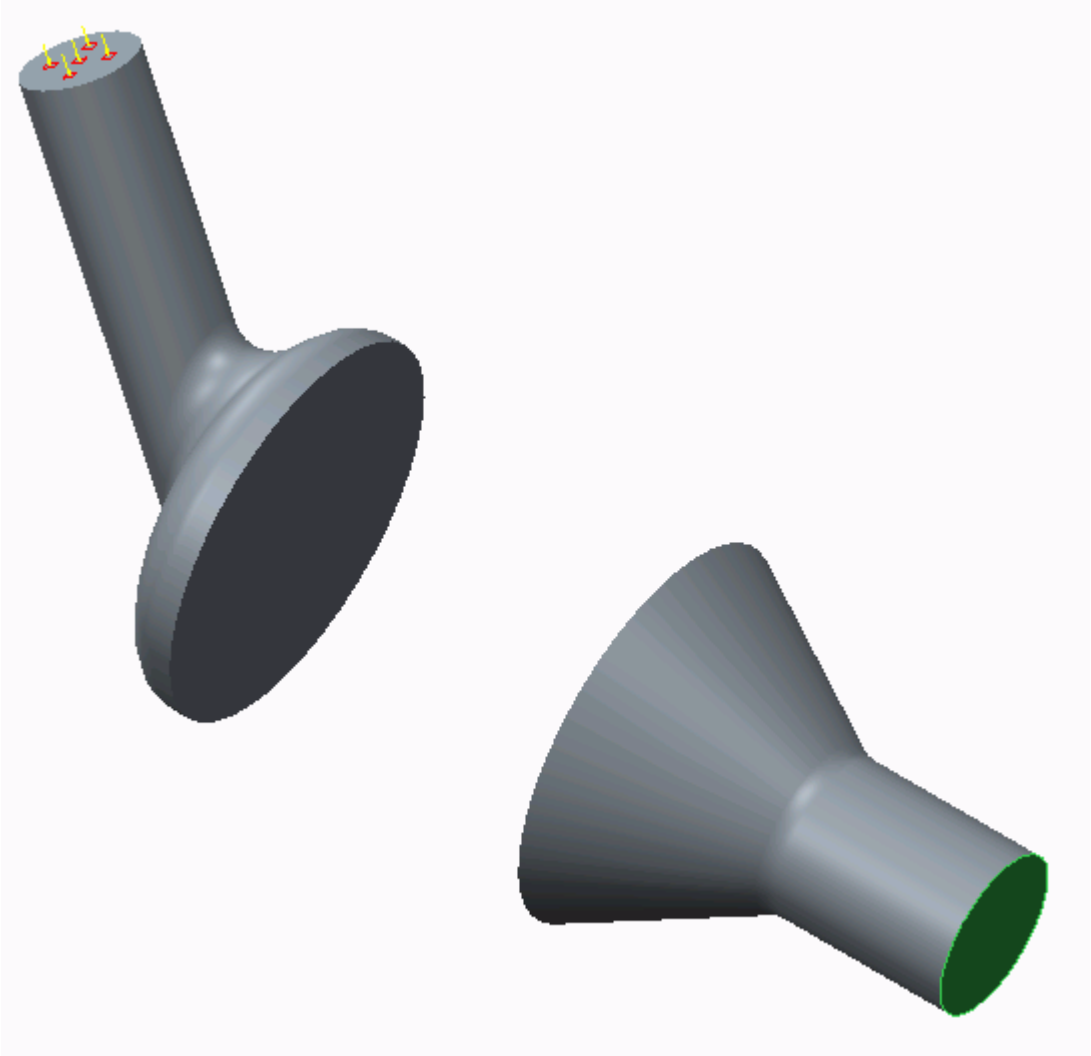
4. Click **OK** to close the *Surface Sets* dialog.
5. Set the following parameters:

Parameter	Value
Type	**Velocity Inlet**
Name	`Inlet`
(*Normal Velocity*)	`5 m/s`

6. Click **OK**.

Create the outlet boundary:

7. Using the same technique, create another boundary.
8. Select the face that is highlighted below.



9. Set the following parameters:

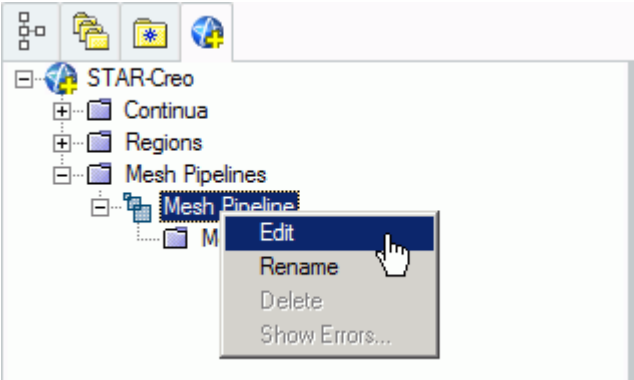
Parameter	Value
Type	**Pressure Outlet**
Name	`Outlet`

10. Click **OK**.

Setting the Mesh Pipeline

To obtain more accurate results, modify the mesh pipeline:

1. Right-click the **Mesh Pipelines** > **Mesh Pipeline** node in the STAR-Creo tree and select **Edit**.



The *Mesh Pipeline Settings* dialog appears.

2. Set the following parameters:

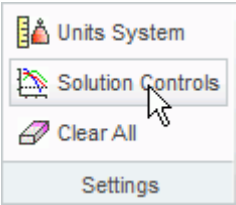
Parameter	Value
Base Size	
(**Base size option**)	**Absolute**
(*Base Size*)	`0.0025 m`
Prism Layer	
(**Layer thickness option**)	**Absolute**
(*Prism Layer Thickness*)	`0.001 m`

3. Click **OK**.

Modifying Solution Controls

Modify the maximum number of iterations and the solution accuracy settings.

1. Click **Solution Controls** on the STAR-Creo ribbon.



The *Solution Controls* dialog appears.

2. Set the following parameters:

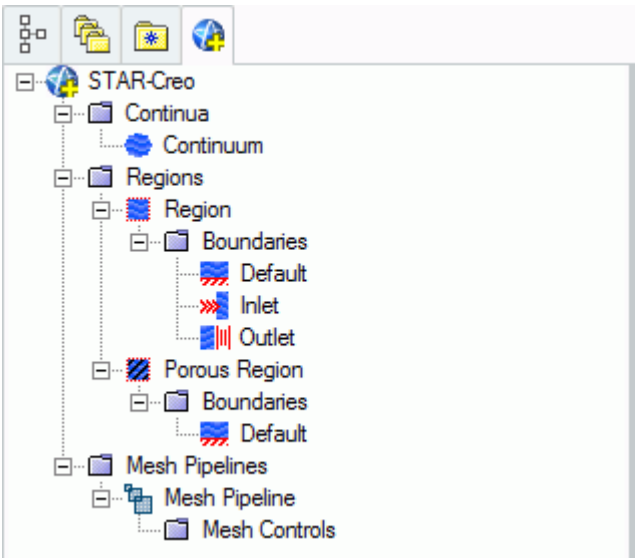
Parameter	Value
(*Number of Iterations*)	`400`

Parameter	Value
Solution Accuracy	**First Order**

3. Click **OK**.
4. Save your progress.

Reviewing the Simulation Set Up

The final STAR-Creo object tree is shown below.



Generating the Surface Mesh

Generate the surface mesh:

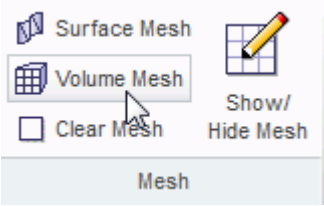
1. From the STAR-Creo ribbon, click **Surface Mesh**.

The surface mesh is shown below.

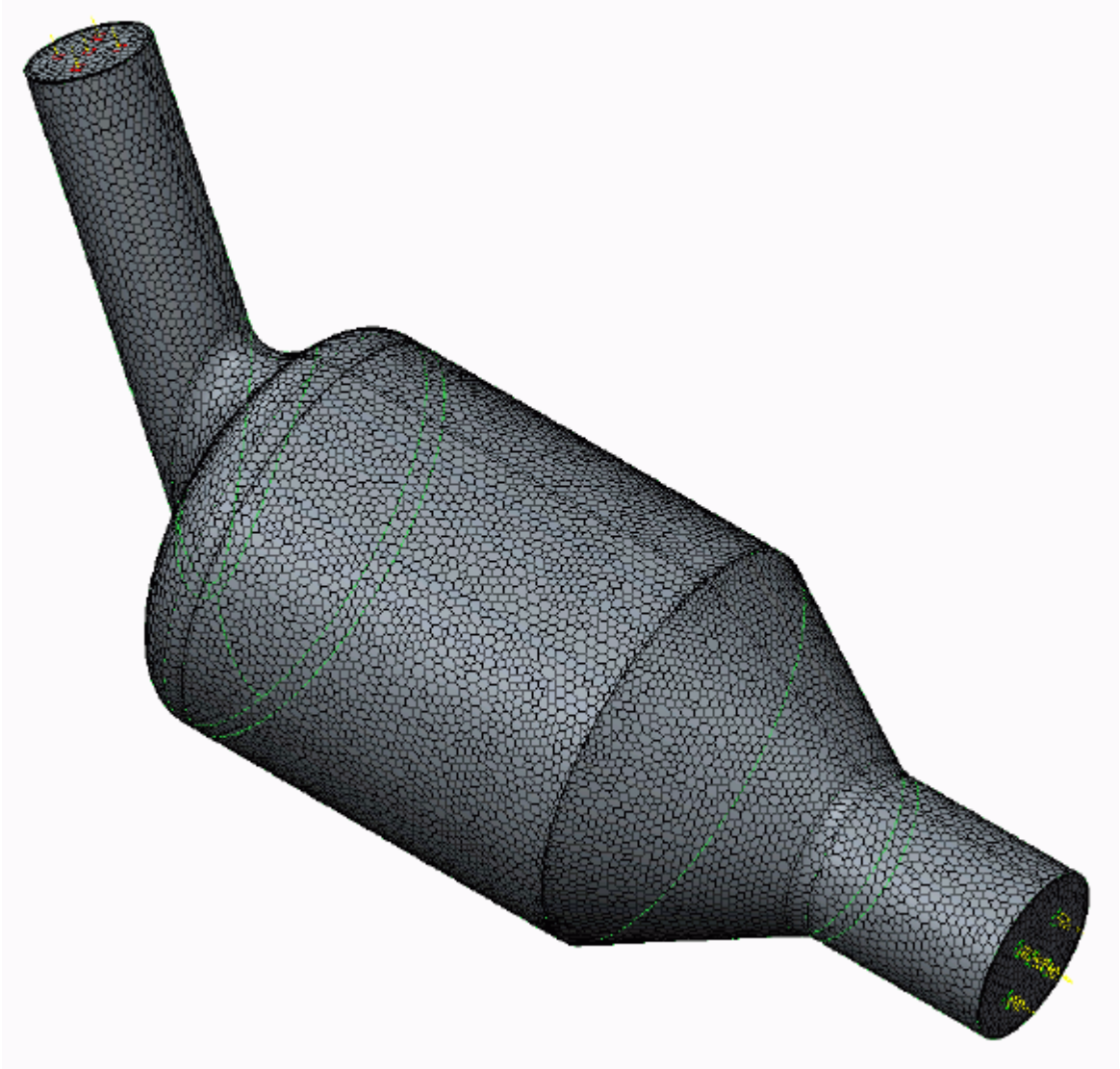
Generating the Volume Mesh

Generate the volume mesh:

1. From the STAR-Creo ribbon, click **Volume Mesh**.



The volume mesh is shown below.

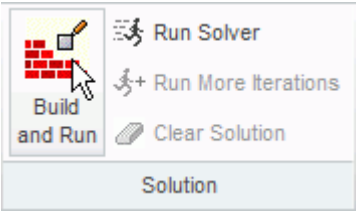


Building and Running

The case is now ready to run.

To run the simulation:

1. From the ribbon, click **Build and Run**.



The *Residuals* plot appears in the STAR-Creo *Output* window. You can use this plot to monitor the convergence of the solution.

When the solver is done, STAR-CCM+ launches.

Post-Processing Results

You perform all post-processing in STAR-CCM+. By default, a scalar scene, *Scalar Scene 1*, automatically appears in the *Graphics* window when the solver finishes. This scene displays the absolute total pressure.

The results that are contained in the following sections are representative. You may see some variation in the results that you obtain, depending on what platform and package you are using.

1. In STAR-CCM+, click **File** > **Save** to save the simulation.

Contents:

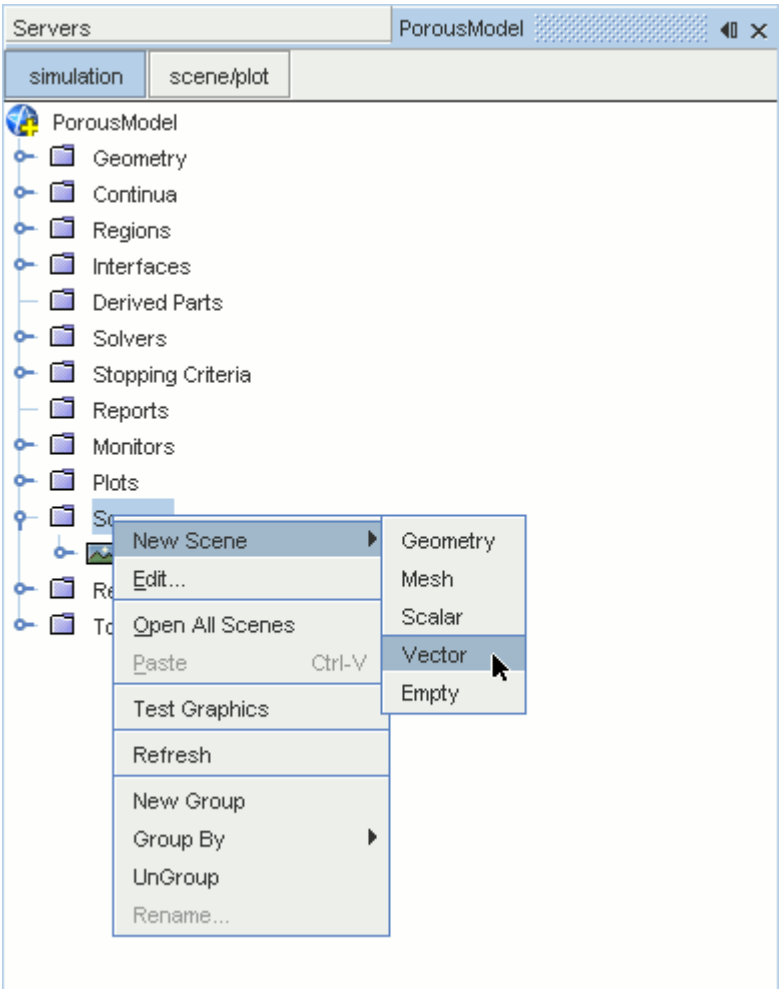
Creating a Scalar Scene
Creating A Streamlines Scene
Creating a Vector Scene
Creating a Streamlines Scene
Creating A Scalar Scene
Visualizing the Wing Skin Temperature
Creating a Streamlines Scene
Creating A Scalar Scene
Creating a Streamlines Scene
Creating a Scalar Scene
Modifying The Scalar Scene
Creating a Streamlines Scene

Creating a Vector Scene

Create a vector scene with a section plane bisecting the geometry. Velocity vectors are plotted on this plane.

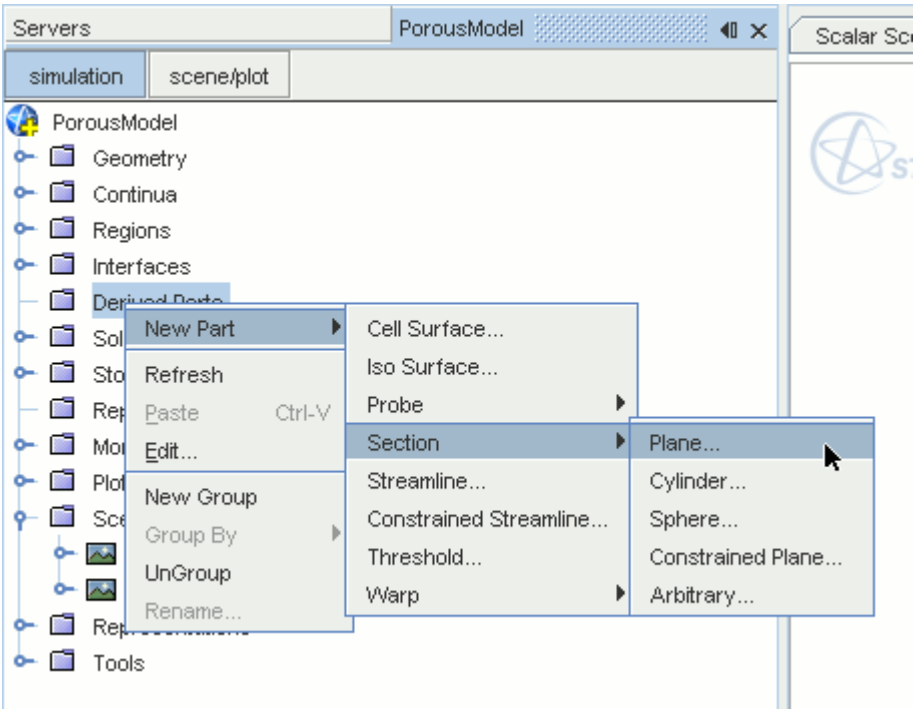
To create the vector scene:

1. Right-click the **Scenes** node in the explorer pane and select **New Scene** > **Vector**.



A new vector scene, *Vector Scene 1*, is created and selected in the *Graphics* window.

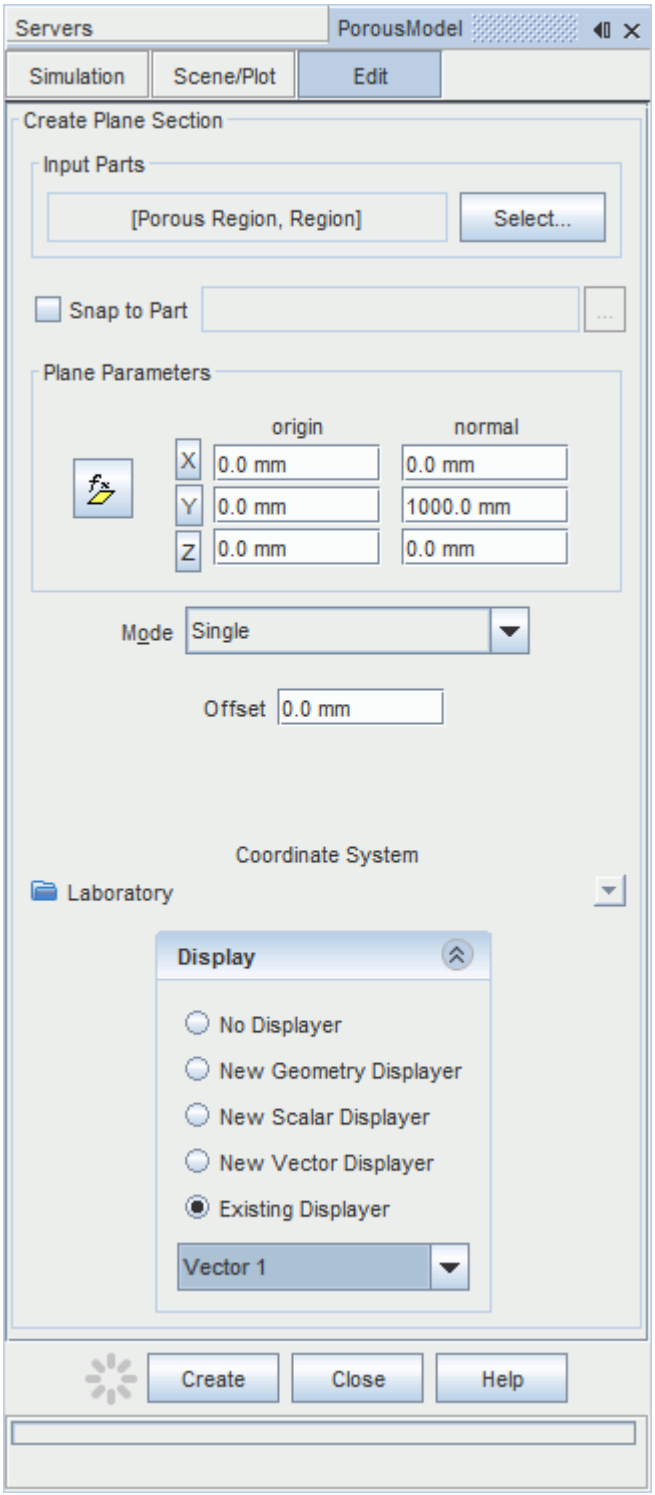
2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane...**.



The *Create Plane Section* dialog appears.

3. In the *Plane Parameters* box, set the *origin* vector to `[0 mm, 0 mm, 0 mm]`.
4. Set the *normal* vector to `[0 mm, 1000 mm, 0 mm]`.
5. In the *Display* group box, select **Existing Displayer** and in the drop-down menu select **Vector 1**.

The completed dialog is shown below.



6. Click **Create** followed by **Close**.

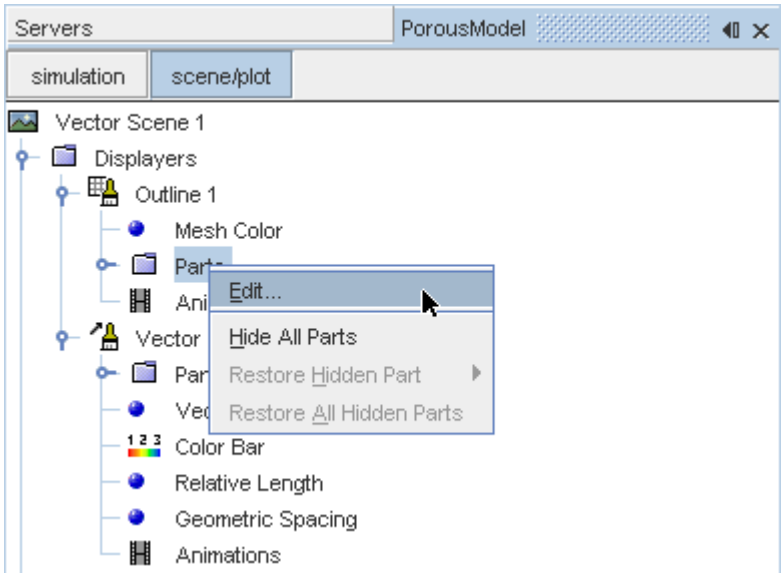
   Velocity is automatically set as the active function when creating a vector displayer.

Change the view of the scene to something more suitable:

7. On the toolbar, click (**Save-Restore-Select views**) and select **Views** > **+X +Y +Z** > **Up +Y**.

In order to remove the unwanted outlines from the scene do the following:

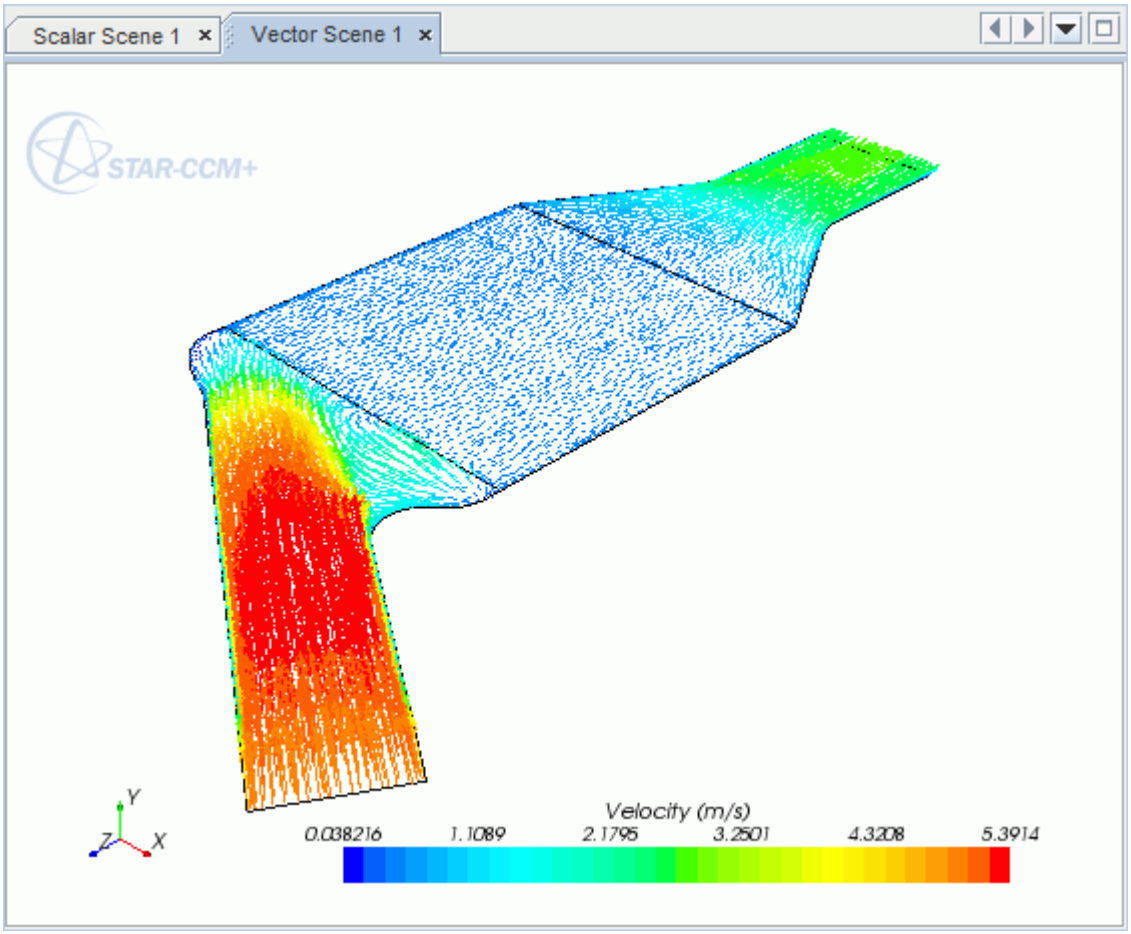
8. Click the **scene/plot** button in the explorer pane.
9. Navigate to **Vector Scene 1** > **Displayers** > **Outline 1** > **Parts**.
10. Right-click the **Parts** node and select **Edit...**.



The *Parts* dialog appears.

11. Right-click on a blank area and choose **Deselect All**.
12. Select the **Derived Parts** > **Plane Section** node.
13. Click **OK**.
14. Click the **simulation** button to return to the simulation object tree.
15. Click (**Save**) to save the simulation .

The finished scene is shown below.



Creating a Streamlines Scene

Create streamlines to get a better understanding of the flow inside the catalyst.

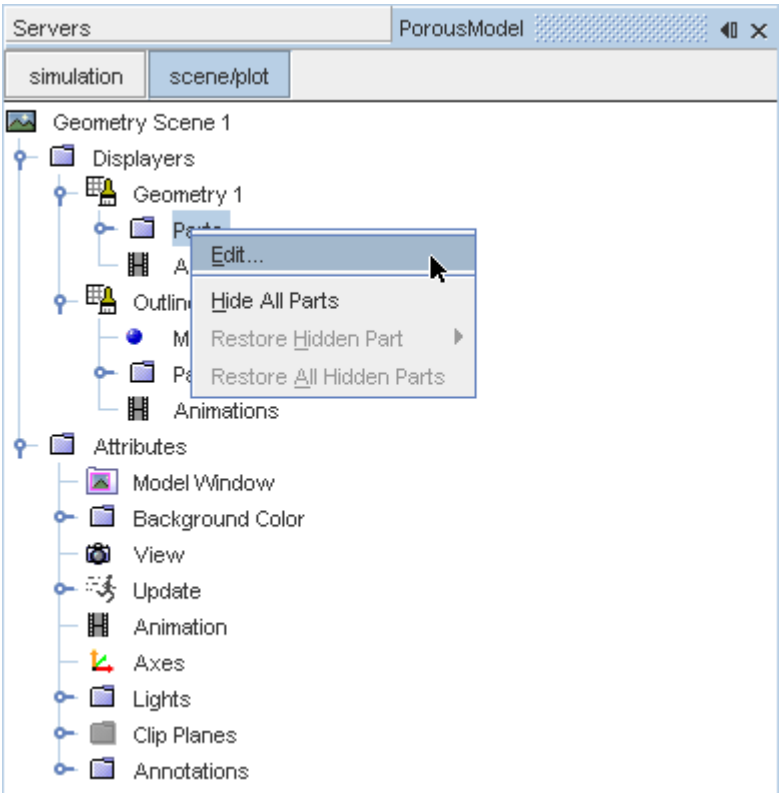
Begin by creating a geometry scene:

1. Right-click the **Scenes** node in the object tree and select **New Scene** > **Geometry**.

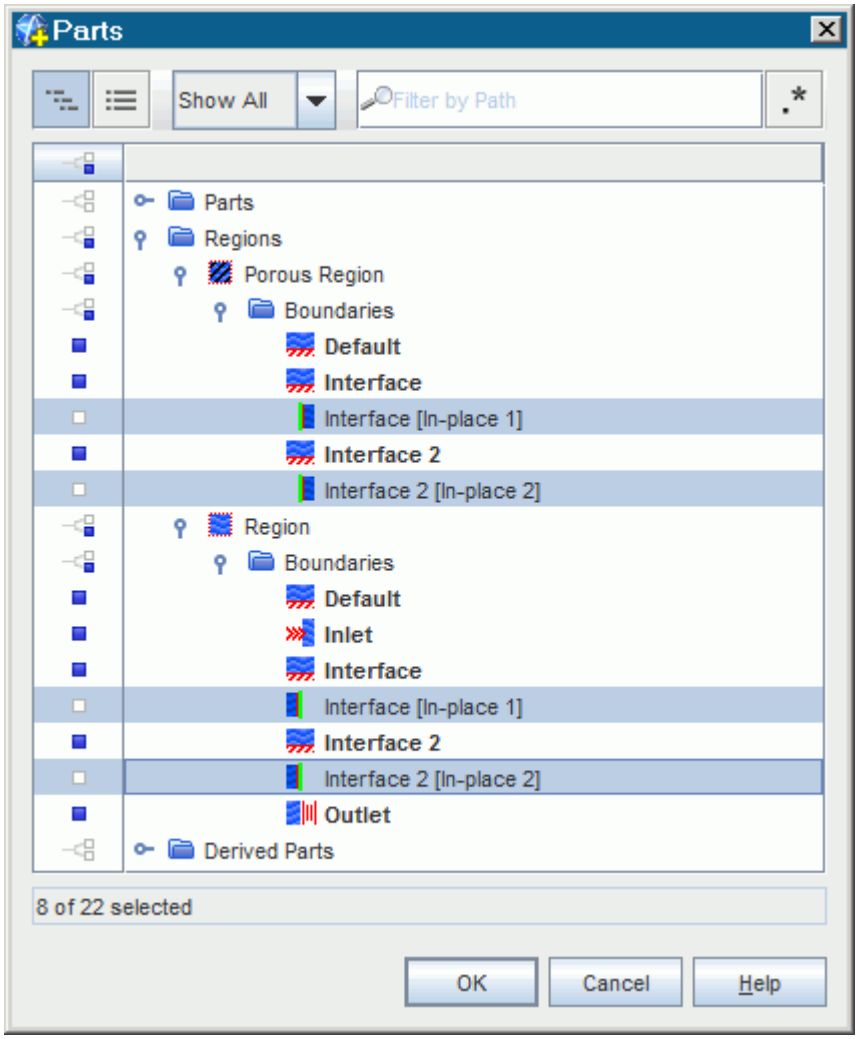
   A new geometry scene is added to the *Graphics* window.
2. Click **scene/plot**.
3. Navigate to **Geometry Scene 1** > **Displayers** > **Geometry 1**.
4. Set *Opacity* to `0.3`.

To remove the yellow interface boundaries from the scene:

5. Right-click **Geometry Scene 1** > **Displayers** > **Geometry 1** > **Parts** and select **Edit...**.



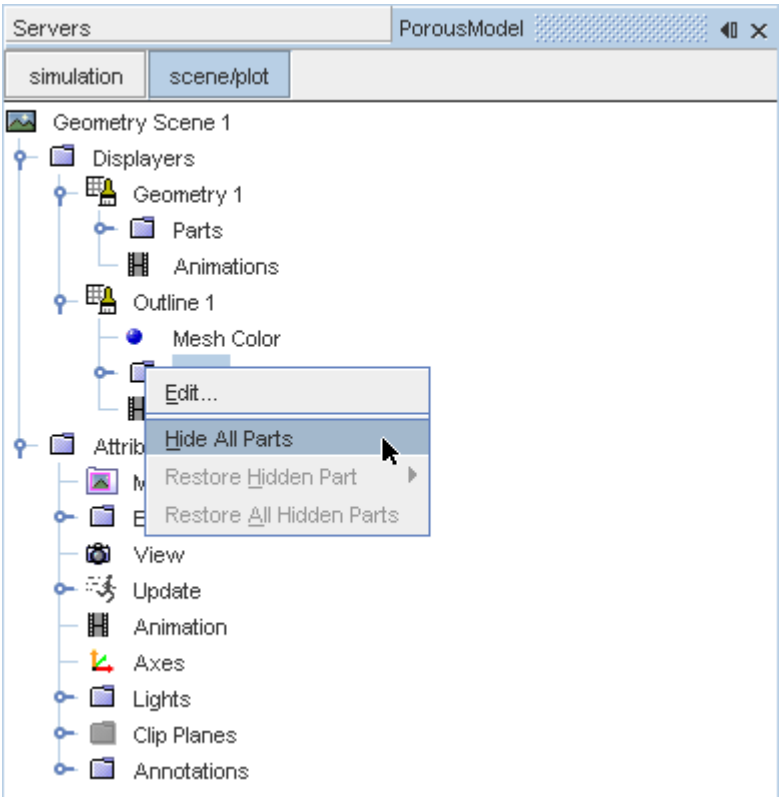
6. In the *Parts* dialog that appears, expand the **Regions** > **Porous Region** > **Boundaries** and **Regions** > **Region** > **Boundaries** nodes.
7. Deactivate the following items in each region:
   - **Interface [In-Place 1]**
   - **Interface 2 [In-Place 2]**



8. Click **OK**.

Hide the outlines on the geometry model:

9. Right-click the **Geometry Scene 1** > **Displayers** > **Outline 1** > **Parts** node in the object tree and select **Hide All Parts**.



10. Click **simulation**.

Change the view of the scene to something more suitable:

11. On the toolbar, click (**Save-Restore-Select views**) and select **Views** > **+X +Y +Z** > **Up +Y**.

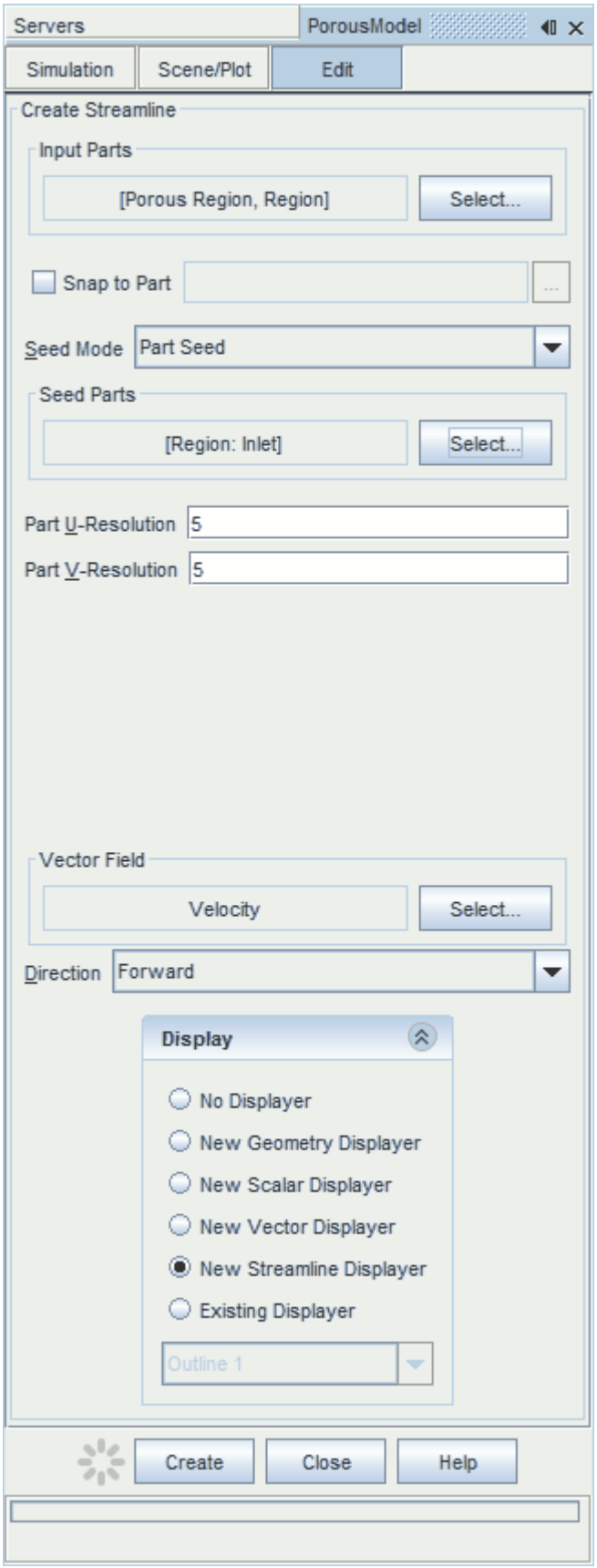
Create the streamlines:

12. Right-click on the **Derived Parts** node and select **New Part** > **Streamline...**.

The *Create Streamline* dialog appears.

13. Make sure that **Part Seed** is selected in the *Seed Mode* section.
14. Set *Seed Parts* to **Regions** > **Region** > **Boundaries** > **Inlet**.
15. Click **OK**.
16. Make sure that **New Streamline Displayer** is selected in the *Display* group box.

The completed dialog is shown below.


Servers
PorousModel
Simulation
Scene/Plot
Edit
Create Streamline
Input Parts
[Porous Region, Region]
Select...
Snap to Part
Seed Mode
Part Seed
Seed Parts
[Region: Inlet]
Select...
Part U-Resolution
5
Part V-Resolution
5
Vector Field
Velocity
Select...
Direction
Forward
Display
No Displayer
New Geometry Displayer
New Scalar Displayer
New Vector Displayer
New Streamline Displayer
Existing Displayer
Outline 1
Create
Close
Help

17. Click **Create** then **Close**.

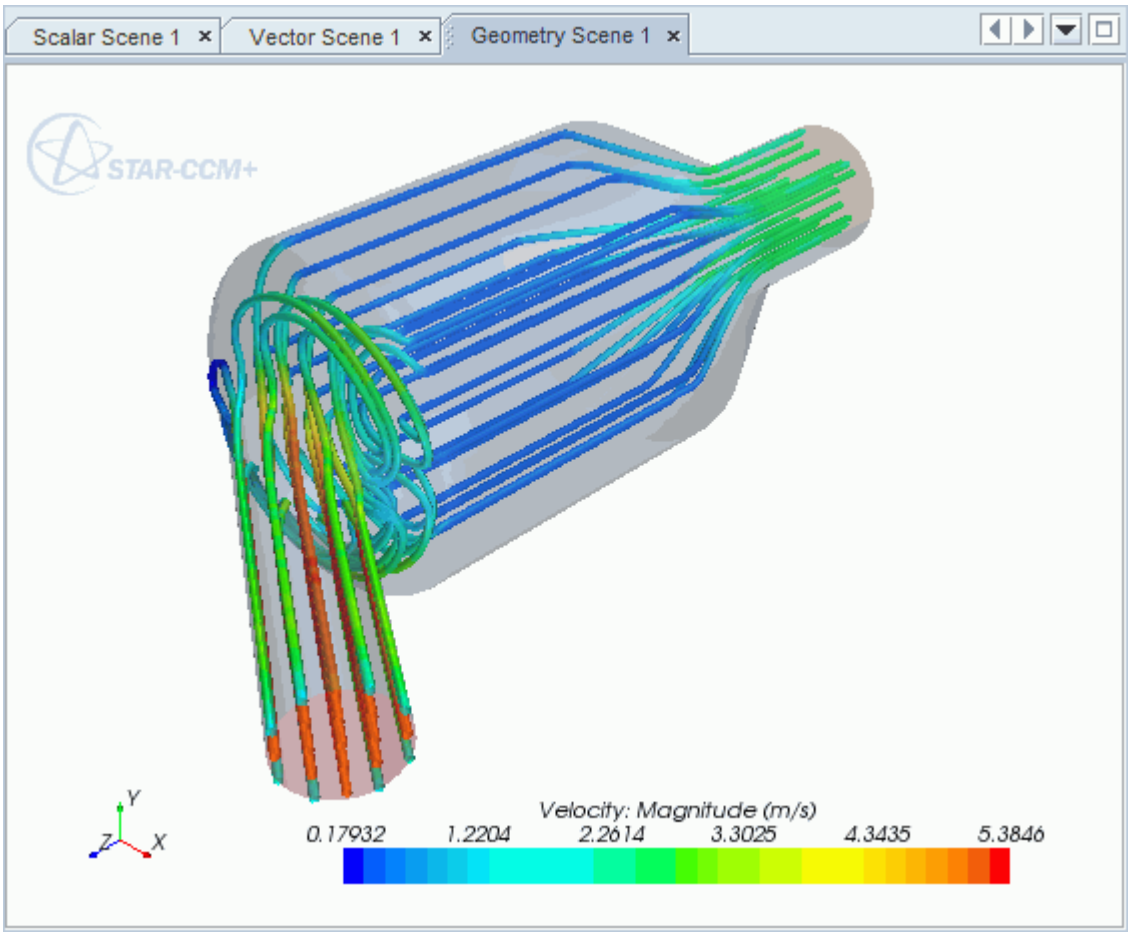
Change the display mode of the streamlines to use tubes instead of the default lines:

18. Click **scene/plot**.
19. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1**.
20. Set *Width* to `0.0012` and *Mode* to **Tubes**.

Change the color of the streamlines to represent velocity magnitude:

21. Navigate to the **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** > **Scalar Field**.
22. Set *Function* to **Velocity** > **Lab Reference Frame** > **Magnitude**.
23. Save the simulation.

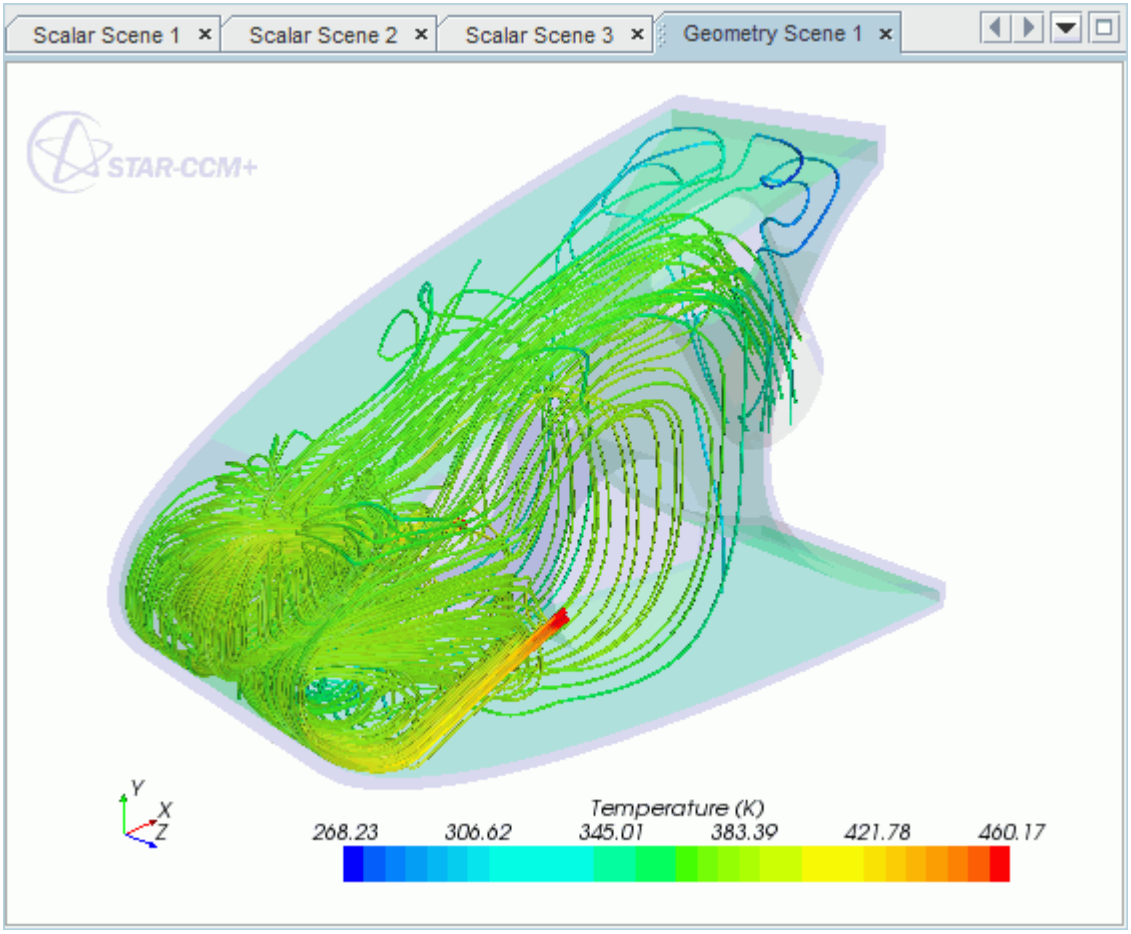
The finished scene is shown below.



Aircraft Wing Deicing System

This tutorial takes you through the steps that are required to run a CFD calculation in the STAR-Creo environment.

For this tutorial, the model is a modern deicing system that is used on aircraft wings. The hot air from the aircraft's jet engine is used to heat the leading edge of the wing.



Contents:

Opening the CAD Model
Selecting Fluid Physics Models
Adding Solid Physics Models
Setting the Solid Region
Defining Boundary Conditions for the Fluid Region
Defining Boundary Conditions for the Solid Region
Setting the Mesh Pipeline
Creating the Volumetric Mesh Controls

Creating a Part Mesh Control
Creating a Face Mesh Control
Modifying Solution Controls
Reviewing the Simulation Set Up
Generating the Surface Mesh
Generating the Volume Mesh
Building and Running
Post-Processing Results

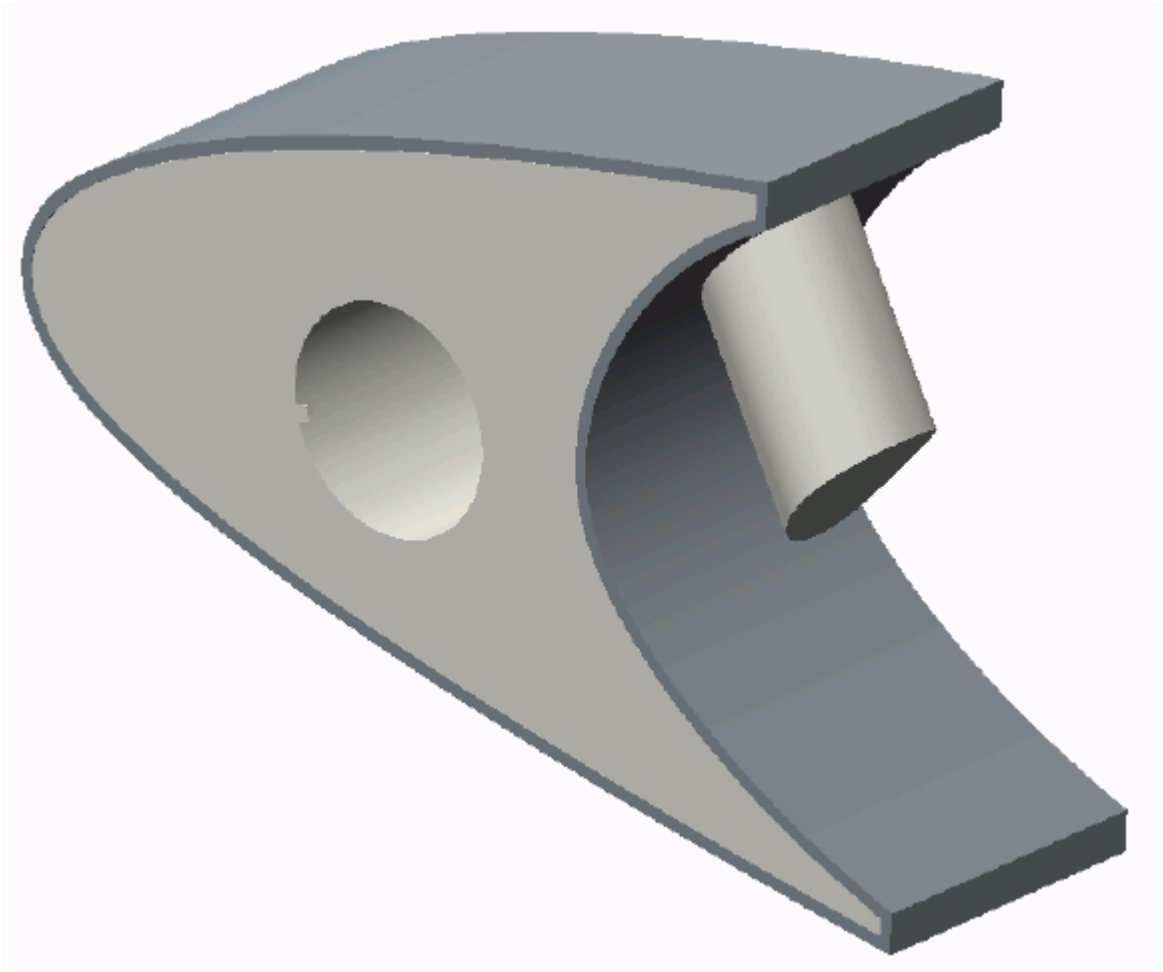
Opening the CAD Model

This directory is located in `[STAR-CCM+_INSTALL_DIR]\doc\startutorialsdata\starcreo\data`.

1. Copy directory `data` to your own disk space and open the part file `piccoloassem.asm` in STAR-Creo.

   The assembly is made up of two parts: the solid leading edge wing skin, and the internal air region.

   The imported geometry is shown below.



The STAR-Creo ribbon appears at the top of the screen.The STAR-Creo object tree is located on the left of the screen.

To switch to this object tree:

2. Click (**STAR Tree**).

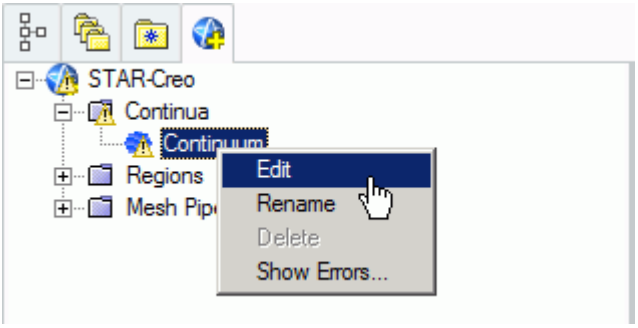
If these items are not visible, refer to Launching STAR-Creo Manually.

Selecting Fluid Physics Models

The first step in setting up this tutorial is to modify the physics continuum that is used to model air. For this case, the flow is turbulent and thermal. Temperature calculations are on by default in STAR-Creo.

Modify the initial conditions:

1. Right-click the **Continua** > **Continuum** node in the STAR-Creo tree and select **Edit**.

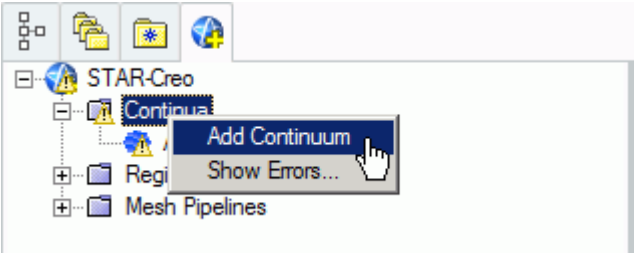


   The *Continuum Settings* dialog appears.
2. Set *Name* to `Air`.
3. Set the *Equation of State* to **Ideal Gas**.
4. Click to expand the *Initialization* tab.
5. Set (*Temperature*) to `400` K.
6. Click **OK** to close the dialog.

Adding Solid Physics Models

The next step in setting up this tutorial is to define the physics continuum of the solid to be used.

1. Right-click the **Continua** node in the STAR-Creo object tree and select **Add Continuum**.



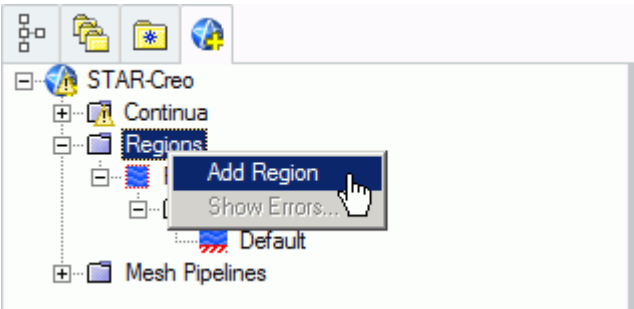
The *Continuum Settings* dialog appears.

2. Set *Name* to `Al`.
3. Set *Material Type* to **Solid**.
4. Set *Material Name* to **Aluminum Alloy**
5. Click ▶ to expand the *Initialization* tab.
6. Set (*Temperature*) to `400 K`.
7. Click **OK**.

Setting the Solid Region

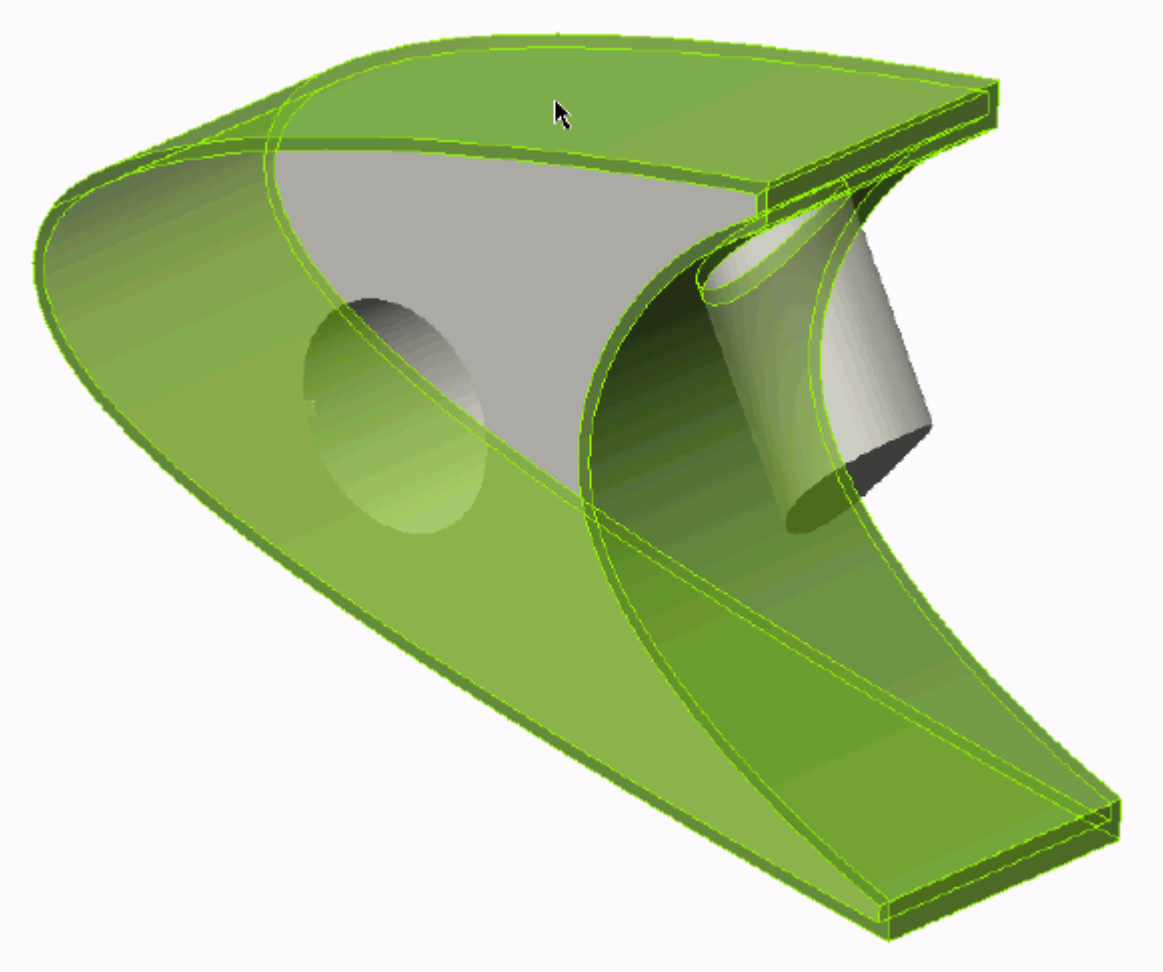
Create a new solid region for the wing skin:

1. Right-click the **Regions** node in the STAR-Creo object tree and select **Add Region**.



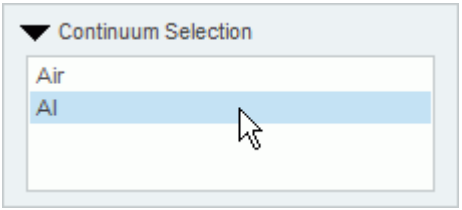
The *Region Settings* dialog appears.

2. Click (**Select Part**) to activate the Creo part selection mechanism.
3. Select the wing skin as shown below.



Once selected, the part becomes highlighted.

4. Set *Name* to `Leading Edge`.
5. Set *Continuum Selection* to **Al**.

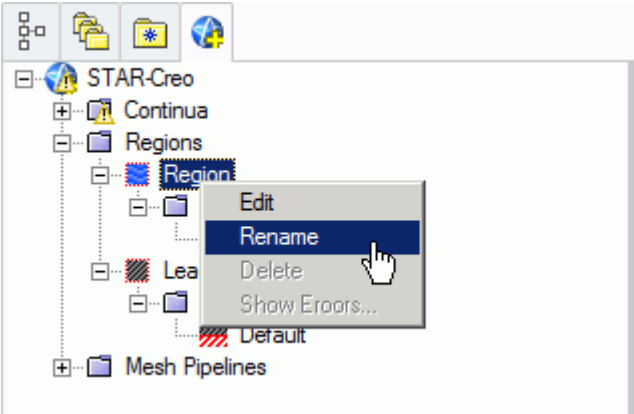


This action assigns the aluminum alloy that you created earlier as the material for this region.

6. Click **OK**.

Now that the solid region of the wing is defined, rename the default region to something more appropriate:

7. Right-click the **Regions** > **Region** node in the STAR-Creo tree and select **Rename**.



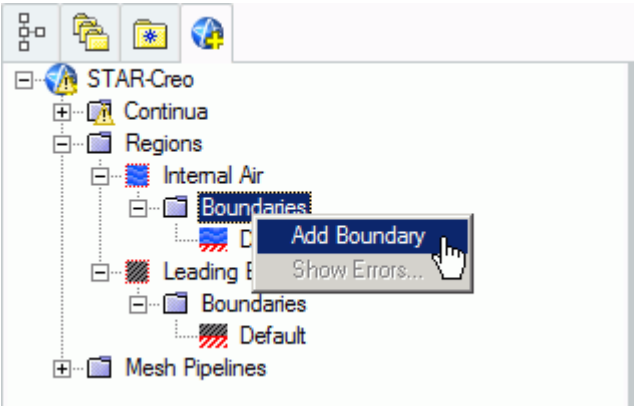
8. Enter `Internal Air` as the new name.
9. Click a blank area to accept the name change.

Defining Boundary Conditions for the Fluid Region

Create the inlet and outlet boundary conditions. Use symmetry boundaries on the side walls to extend the simulation domain. Use a wall boundary with a fixed temperature to model the piccolo walls.

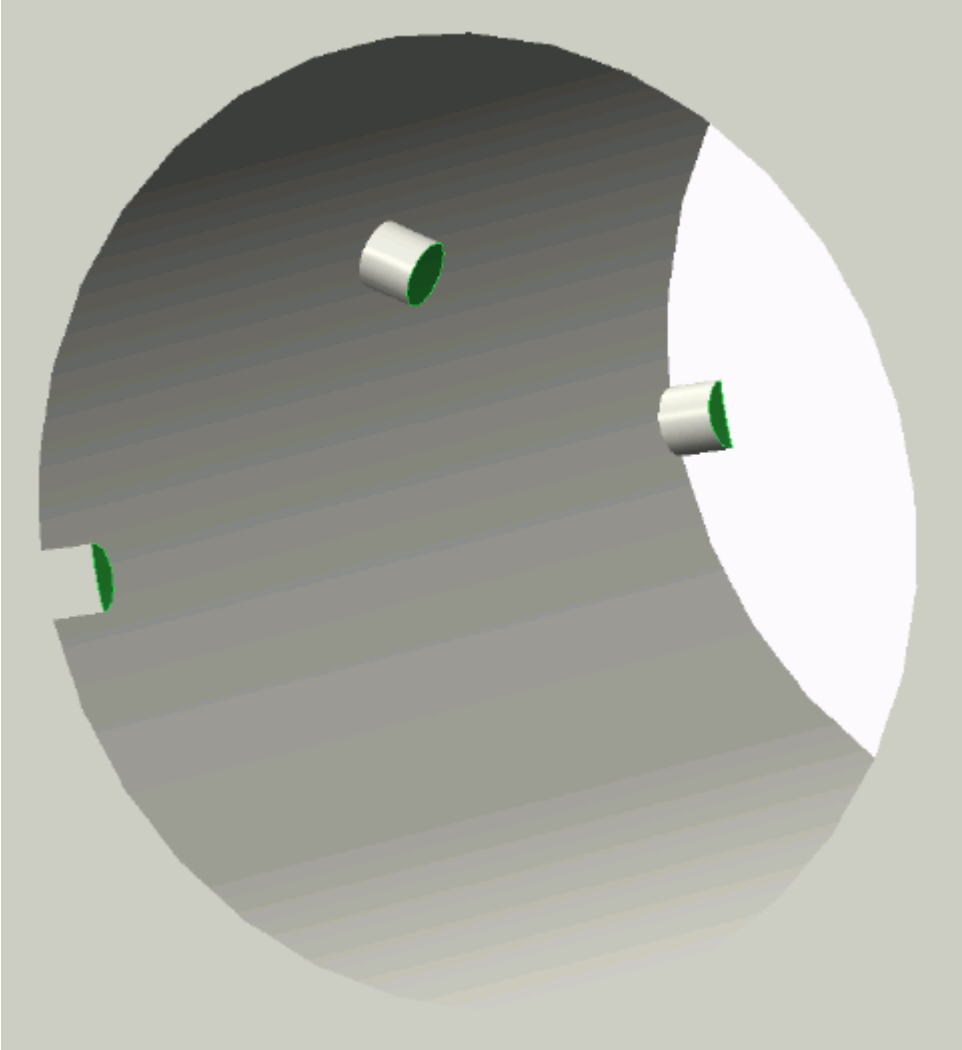
Create the inlet boundary first:

1. Right-click the **Regions** > **Internal Air** > **Boundaries** node and select **Add Boundary**.



The *Boundary Settings* dialog appears.

2. Click (**Select Face**) to activate the Creo face selection mechanism.
3. On the CAD model, select the three faces that are highlighted below.



These selections are reported in the *Included Surfaces* box in the *Surface Sets* dialog.

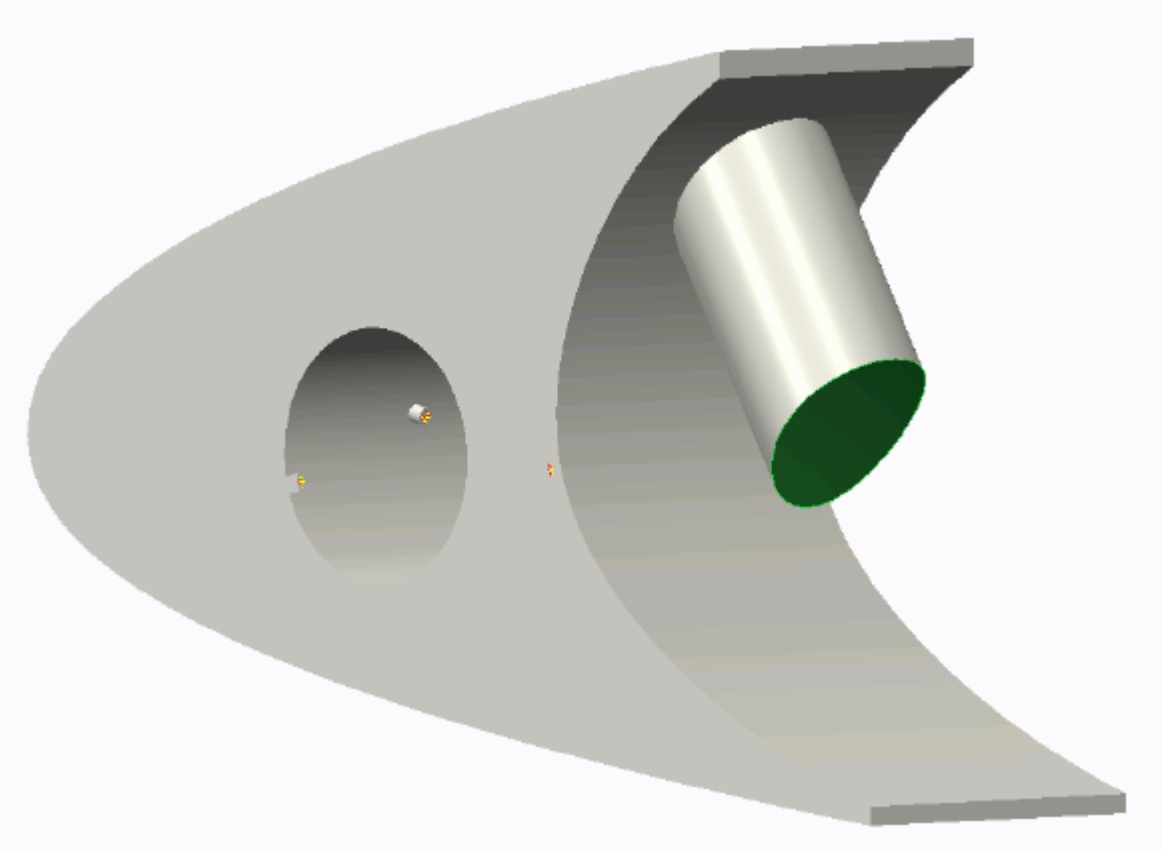
4. Set the following properties:

Parameter	Value
Type	**Velocity Inlet**
Boundary NameName	`Inlet`
(*Normal Velocity*)	`90 m/s`
(*Temperature*)	`460 K`
(*Turbulence Intensity*)	`0.135`
(*Turbulent Length Scale*)	`2.2e-005 m`

5. Click **OK**.

Create the outlet boundary:

6. Using the same technique, create another boundary.
7. Select the face that is highlighted below.



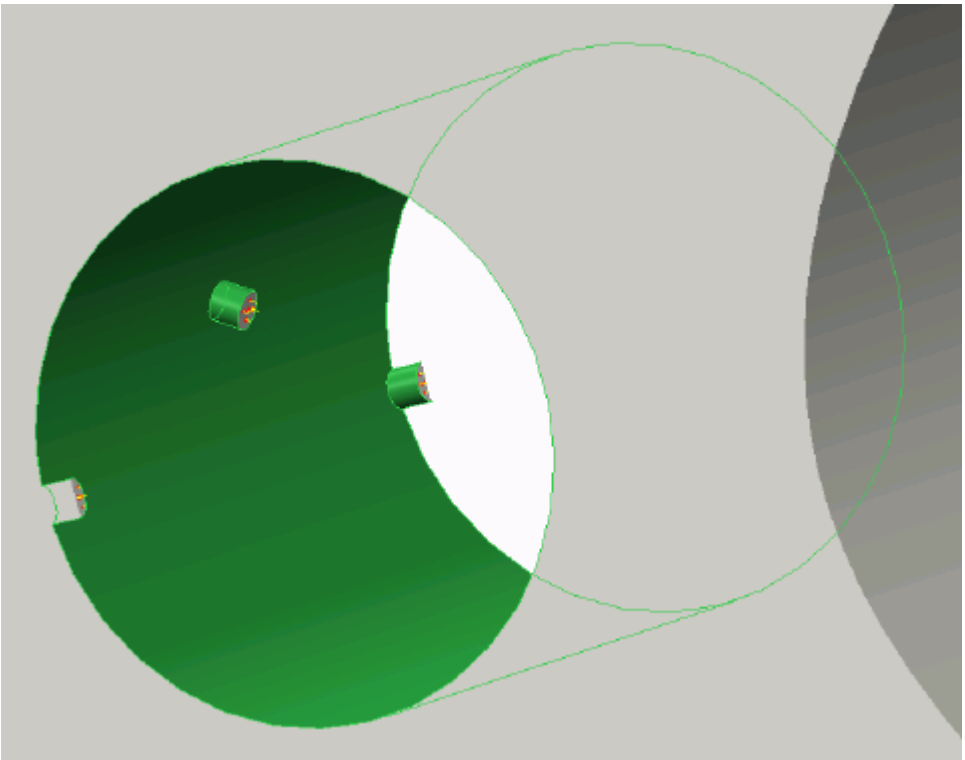
8. Set the following parameters:

Parameter	Value
Type	**Pressure Outlet**
Name	`Outlet`

9. Click **OK**.

Create the wall boundary that models the heated piccolo walls.

10. Create another boundary.
11. Select the six faces highlighted below, make sure to select the faces that are perpendicular to the inlet boundaries. To select multiple faces hold down the <Ctrl> key.



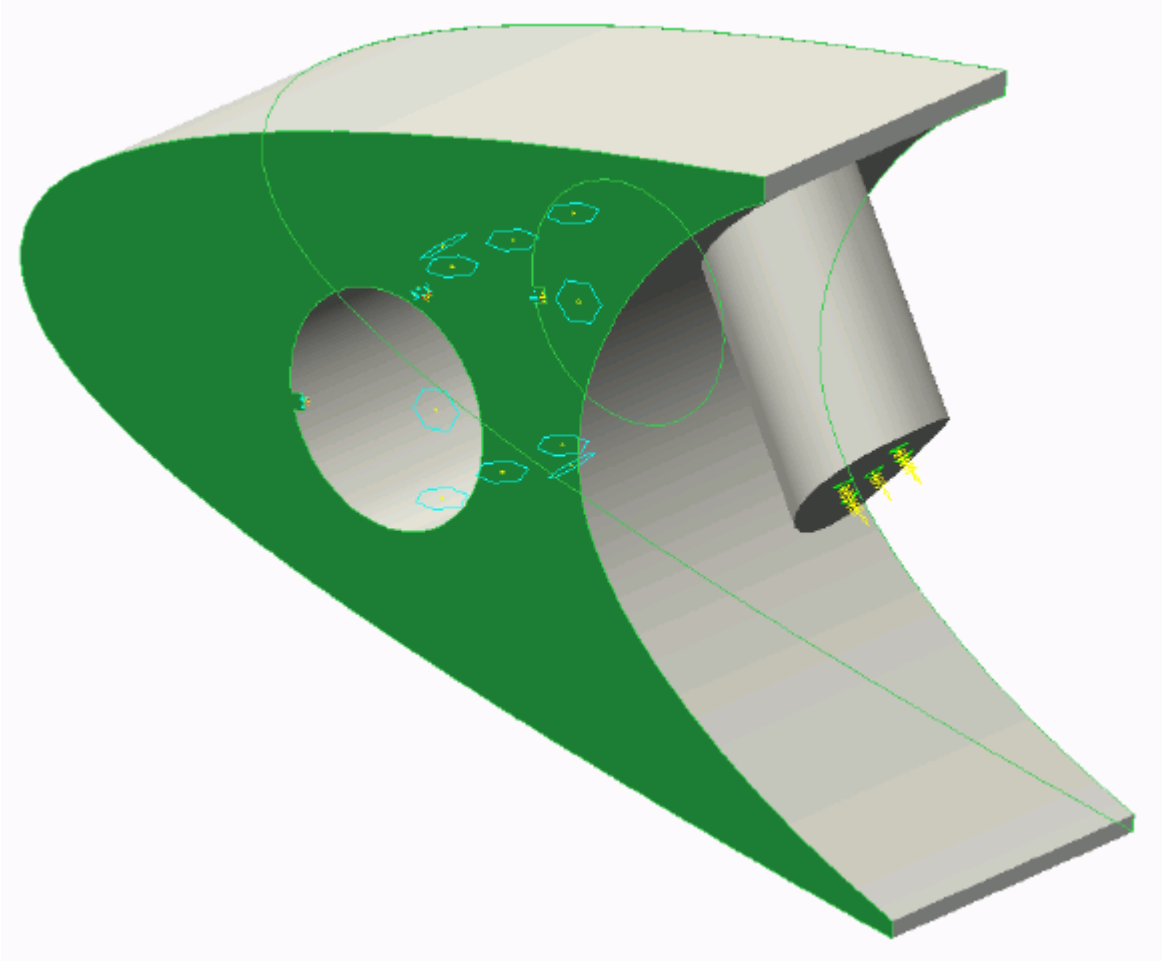
12. Set the following parameters:

Parameter	Value
Type	**Wall**
Name	`Piccolo Wall`
Thermal Specification	**Temperature**
(*Temperature*)	`460 K`

13. Click **OK**.

Create the symmetry boundaries:

14. Create a new boundary.
15. Select the two faces that are highlighted below, one on either side of the internal air region.



16. Set the following parameters:

Parameter	Value
Type	**Symmetry Plane**
Name	`Air Symmetry`

17. Click **OK**.

Defining Boundary Conditions for the Solid Region

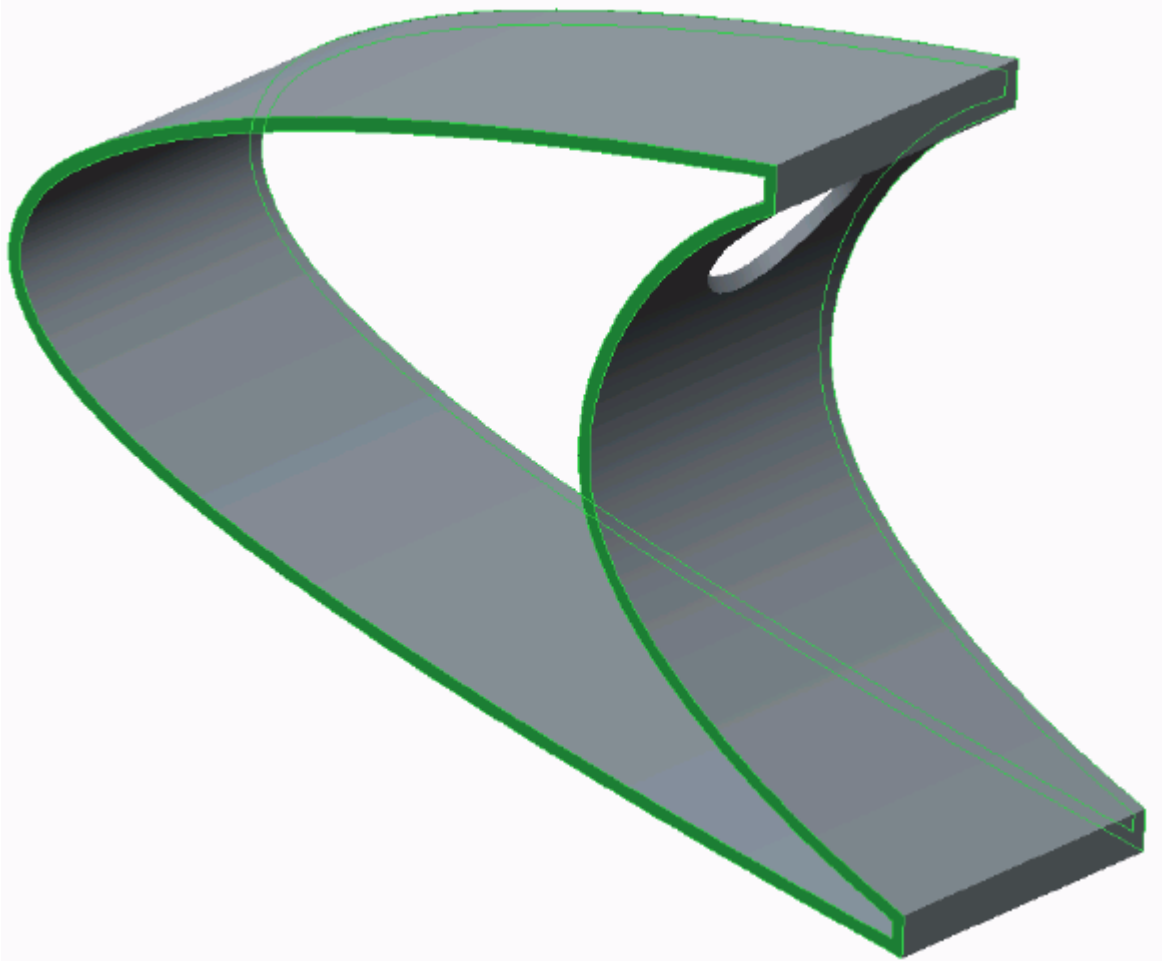
Use the convection boundary condition to model the heat propagation from the solid wing.

The convective boundary condition models ambient conditions on the outside of the domain. Use convection to model the heat propagation in the solid wing skin.

Create the symmetry boundary first:

1. Right-click the **Regions** > **Leading Edge** > **Boundaries** node and select **Add Boundary**.

   The *Boundary Settings* dialog appears.
2. Select the two faces highlighted below, one on either side of the geometry.



3. Set the following parameters:

Parameter	Value
Type	**Symmetry Plane**
Name	`Solid Symmetry`

4. Click **OK**.

Modify the default boundary so that it uses convection:

5. Right-click the **Regions** > **Leading Edge** > **Boundaries** > **Default** node and select **Edit**.

   The *Boundary Settings* dialog appears.

6. Set the following properties:

Parameter	Value
Thermal Specification	**Convection**
HTC (*Heat Transfer Coefficient*)	`100 W/m^2.K`
(*Temperature*)	`228 K`

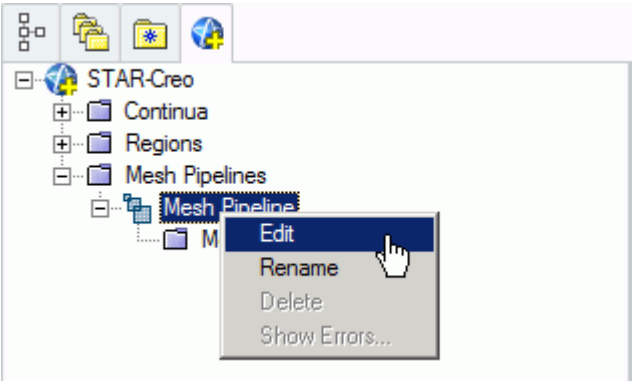
7. Click **OK**.

8. Save your progress.

Setting the Mesh Pipeline

To obtain more accurate results, modify the mesh pipeline:

1. Right-click the **Mesh Pipelines** > **Mesh Pipeline** node in the STAR-Creo tree and select **Edit**.



The *Mesh Pipeline Settings* dialog appears.

2. Set the following parameters:

Parameter	Value
Base Size	
(**Base size option**)	**Absolute**
(*Base Size*)	`0.007 m`
Prism Layer	
(*Prism Layer Thickness*)	`10%`

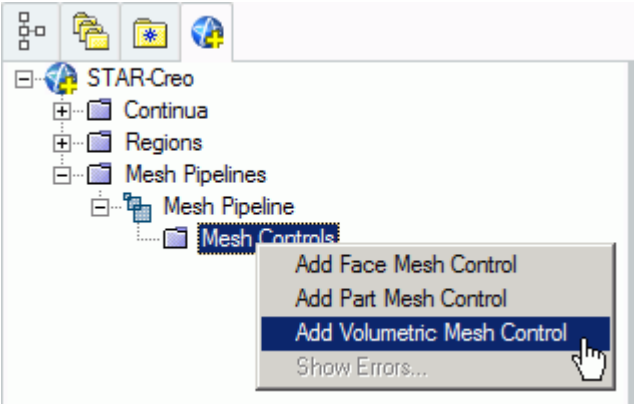
3. Click **OK**.

Creating the Volumetric Mesh Controls

Create volumetric mesh control regions to capture the finer details of the flow near the air jets that heat the leading edge of the wing.

Create the volumetric mesh control for the inlet:

1. Right-click the **Mesh Pipelines** > **Mesh Pipeline** > **Mesh Controls** node in the STAR-Creo tree and select **Add Volumetric Mesh Control**.



The *Volumetric Control* dialog appears.

2. Set *Name* to `Inlet Box`.
3. Activate the *Absolute* option.
4. Set (*Mesh Size*) to `0.0005 m`.
5. Enter the following coordinates for Point 1:

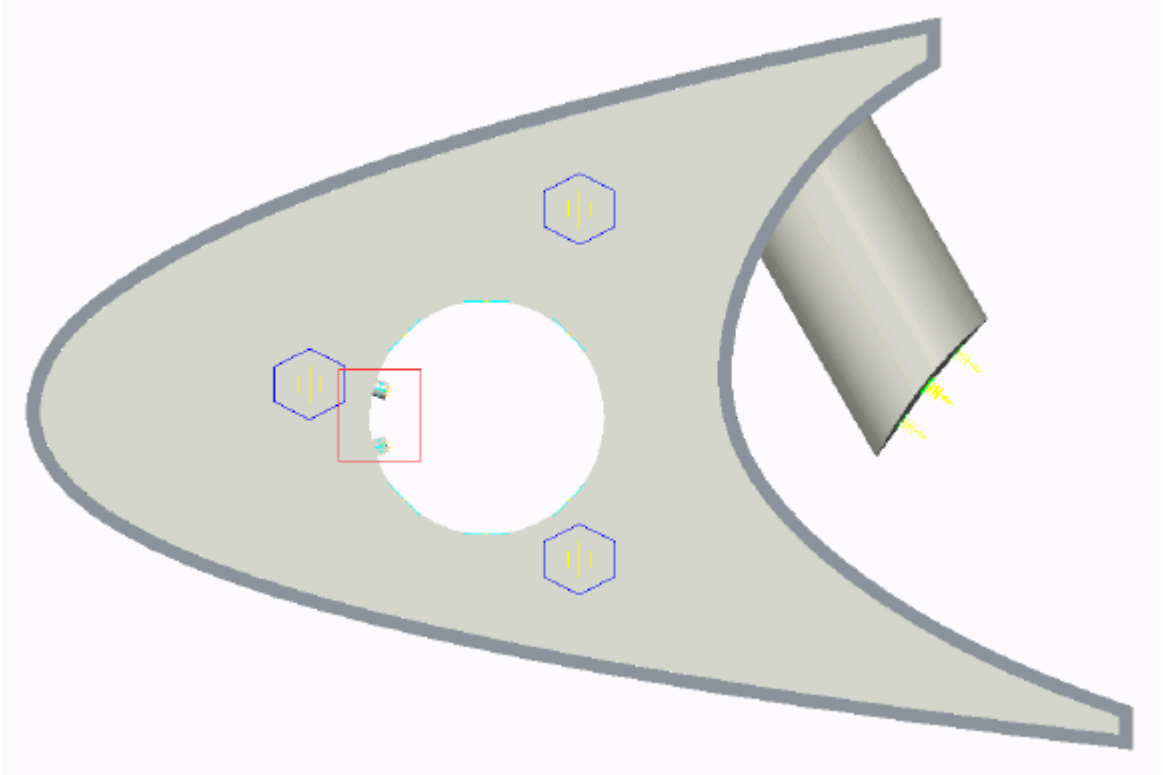
Coordinate	Value
X1	`-100.00`
Y1	`21.00`
Z1	`-20.00`

6. Enter the following coordinates for Point 2:

Coordinate	Value
X2	`-90.00`
Y2	`10.00`
Z2	`60.00`

7. Click **Create**.

   The volumetric control region is shown below.



8. Click **OK**.

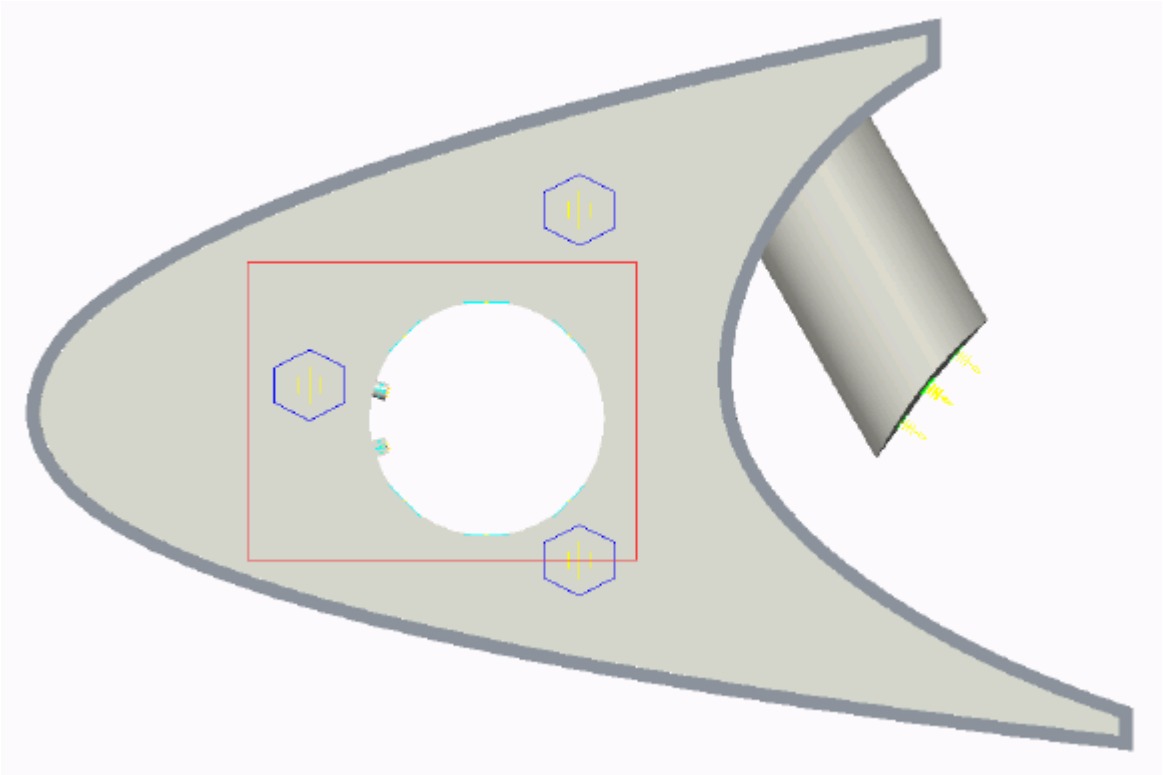
Create the volumetric mesh control for the region near the piccolo tube and use the following settings:

9. Using the same technique, create another volumetric mesh control with the following settings:

Parameter	Value
Name	`Mid Box`
Mesh Size	
(**Base size option**)	**Absolute**
(*Mesh Size*)	`0.0015 m`
Shape	
X1	`-111.00`
Y1	`34.00`
Z1	`-20.00`
X2	`-64.00`
Y2	`-2.00`
Z2	`60.00`

10. Click **Create**.

    The volumetric control region is shown below.

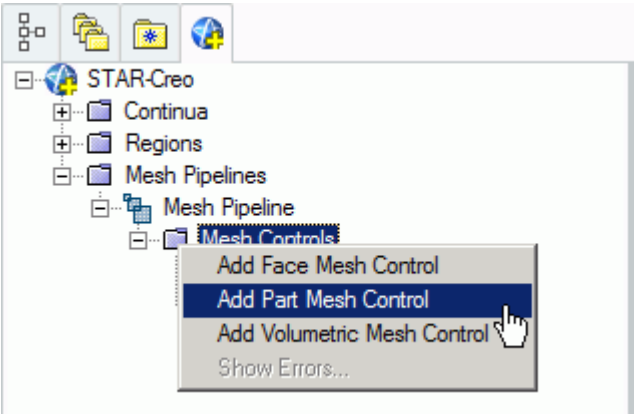


11. Click **OK**.
12. Save your progress.

Creating a Part Mesh Control

Create a part mesh control to disable the prism layer generation in the solid region. Solid regions do not require prism layers.

1. Right-click **Mesh Pipelines** > **Mesh Pipeline** > **Mesh Controls** in the STAR-Creo tree and select **Add Part Mesh Control**.



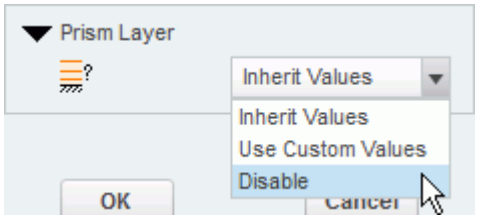
The *Part Mesh Control Settings* dialog appears.

2. Select the solid wing region on the CAD model.

   The selection is added to the *Part Selection* box.

3. Set *Name* to `Solid Region`.

4. In the *Prism Layer* box, set (*Prism layer option*) to **Disable** using the drop-down menu.

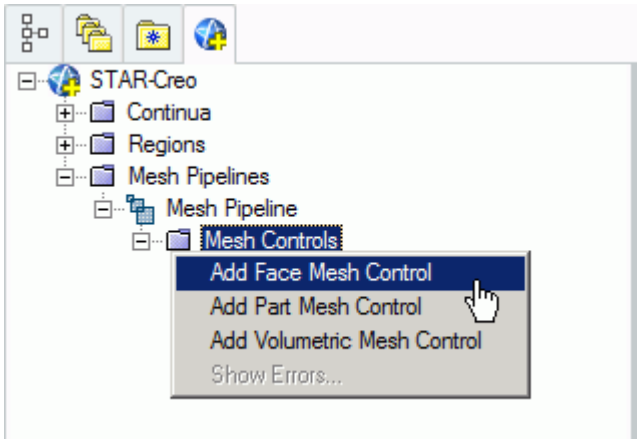


5. Click **OK**.

Creating a Face Mesh Control

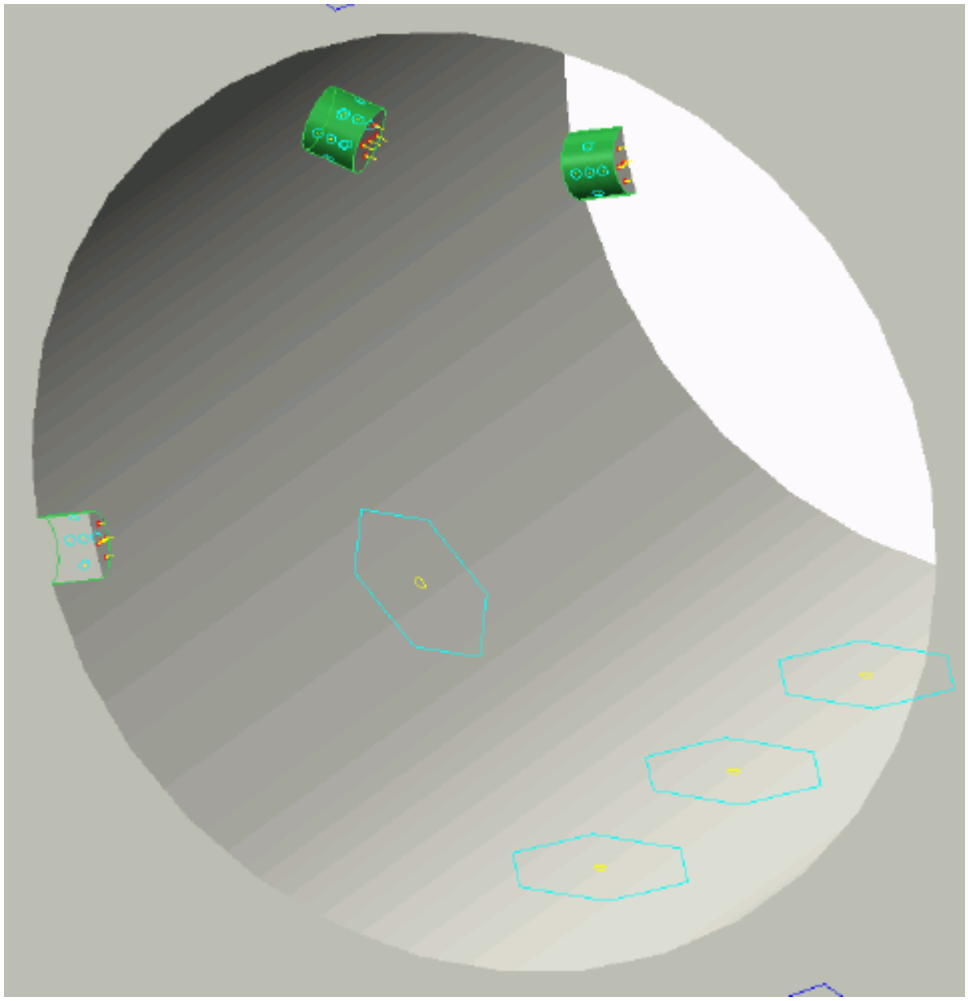
Create a face mesh control to reduce the size of the prism layer thickness at the inlet boundary.

1. Right-click **Mesh Pipelines** > **Mesh Pipeline** > **Mesh Controls** in the STAR-Creo tree and select **Add Face Mesh Control**.



   The *Face Mesh Control Settings* dialog appears.

2. Select the cylindrical faces that are perpendicular to the inlet boundary, as shown below.

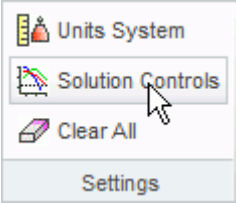


3. Set *Name* to `Inlet Refinement`.
4. Activate the *Custom Mesh Size* option.
5. Activate the *Absolute* option.
6. Set (*Custom Mesh Size*) to `0.00025` m.
7. In the *Prism Layer* box, set (*Custom Prism Layer Option*) to **Use Custom Values**.
8. Set *Layer Thickness Option* to **Absolute**.
9. Set (*Layer Thickness*) to `0.0002` m.
10. Click **OK**.

Modifying Solution Controls

Modify the maximum number of iterations and the solution accuracy. To help accelerate convergence, edit the solid under-relaxation factor:

1. Click **Solution Controls** on the STAR-Creo ribbon.



The *Solution Controls* dialog appears.

2. Set the following parameters:

Parameter	Value
(*Number of Iterations*)	500
(*Convergence Level*)	0.01
(*Solid Temperature Under-Relaxation Factor*)	0.9999

3. Click **OK**.
4. Save your progress.

Reviewing the Simulation Set Up

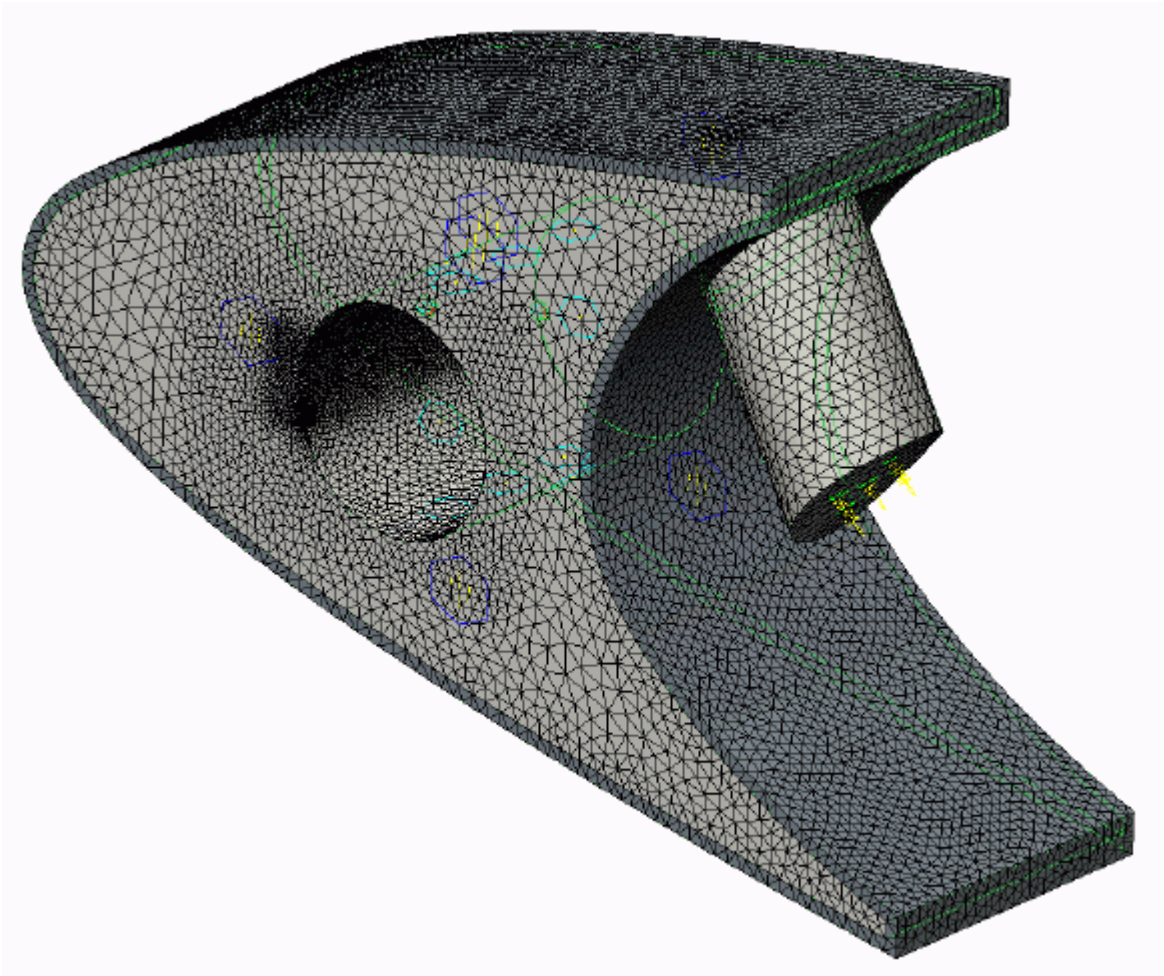
The final STAR-Creo object tree is shown below.

Generating the Surface Mesh

Generate the surface mesh:

1. From the STAR-Creo ribbon, click **Surface Mesh**.

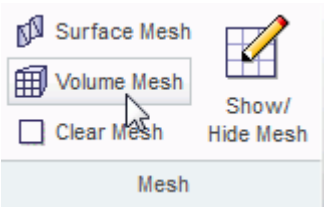
The surface mesh is shown below.



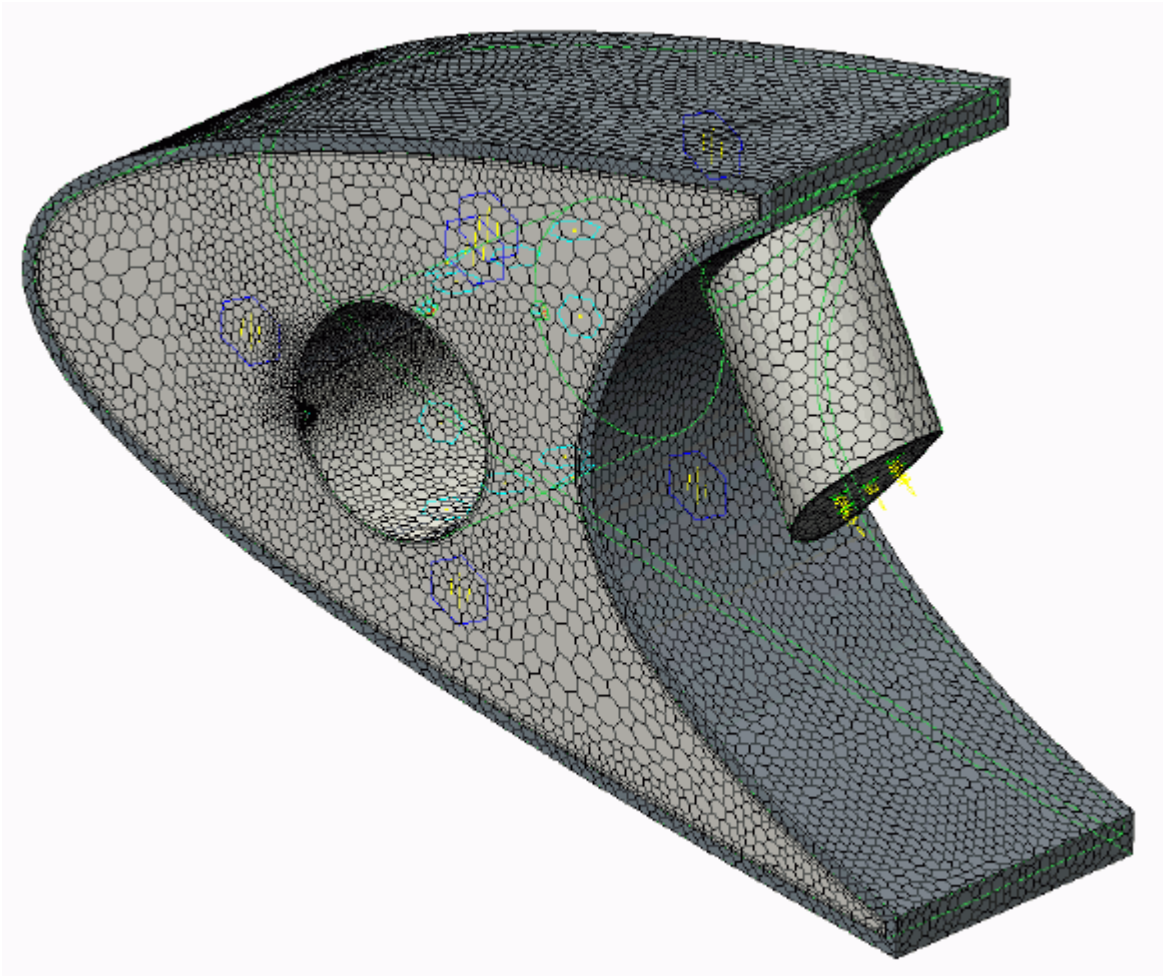
Generating the Volume Mesh

Generate the volume mesh:

1. From the STAR-Creo ribbon, click **Volume Mesh**.



The volume mesh is shown below.

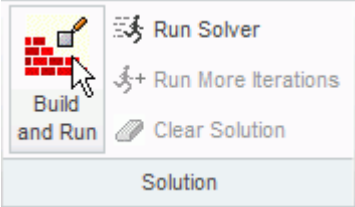


Building and Running

The case is now ready to run.

To run the simulation:

1. From the ribbon, click **Build and Run**.



The *Residuals* plot appears in the STAR-Creo *Output* window. You can use this plot to monitor the convergence of the solution.

When the solver is done, STAR-CCM+ launches.

Post-Processing Results

You perform all post-processing in STAR-CCM+. By default, a scalar scene, *Scalar Scene 1*, automatically appears in the *Graphics* window when the solver finishes. This scene displays the absolute total pressure.

The results that are contained in the following sections are representative. You may see some variation in the results that you obtain, depending on what platform and package you are using.

1. In STAR-CCM+, click **File** > **Save** to save the simulation.

Contents:

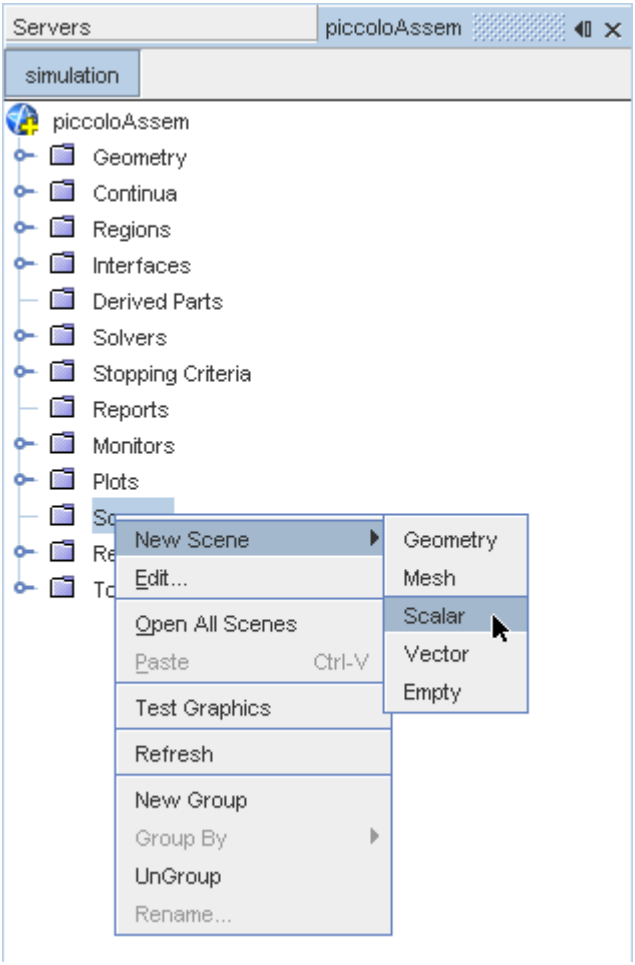
Creating a Scalar Scene
Creating A Streamlines Scene
Creating a Vector Scene
Creating a Streamlines Scene
Creating A Scalar Scene
Visualizing the Wing Skin Temperature
Creating a Streamlines Scene
Creating A Scalar Scene
Creating a Streamlines Scene
Creating a Scalar Scene
Modifying The Scalar Scene
Creating a Streamlines Scene

Creating A Scalar Scene

Create a scalar scene containing a cross-section of the model. Plot temperature on this section to demonstrate how the hot gas coming from the piccolo tube affects the wing skin and the air inside the wing.

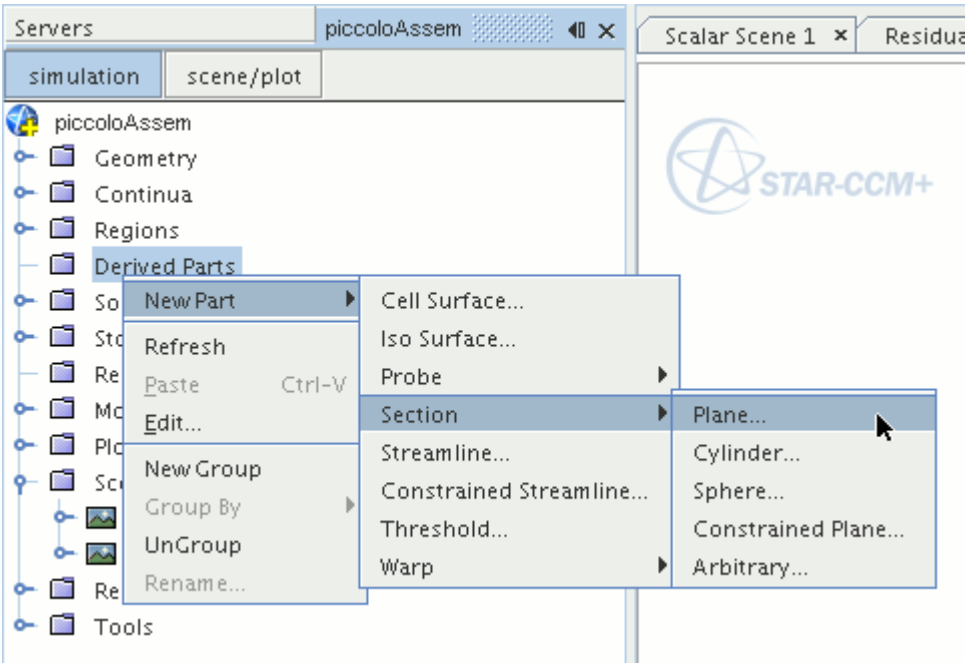
To create the scene:

1. Right-click the **Scenes** node in the explorer pane and select **New Scene** > **Scalar**.



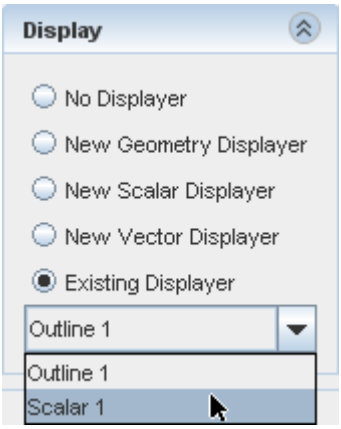
A new scalar scene, *Scalar Scene 2*, is created and selected in the *Graphics* window.

2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane...**.



The *Create Plane Section* dialog appears.

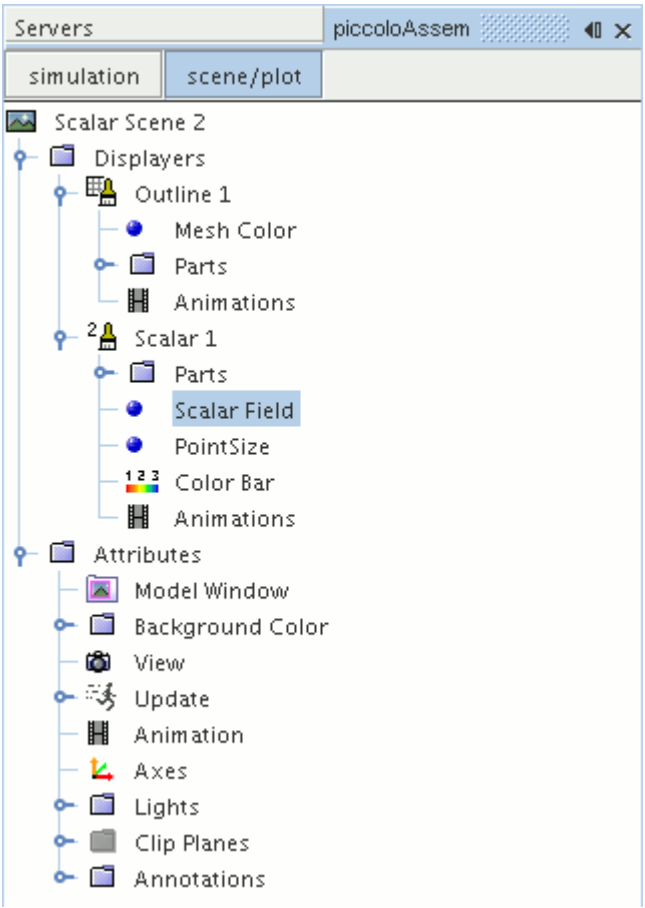
3. Set the origin coordinates to `[0.0 mm, 0.0 mm, 18.4 mm]`.
4. Set the *normal* vector to `[0.0 mm, 0.0 mm, 1.0 mm]`.
5. In the *Display* group box, select **Existing Displayer**, and select **Scalar 1** from the drop-down menu.



6. Click **Create** followed by **Close**.

Change the scalar field to display temperature contours:

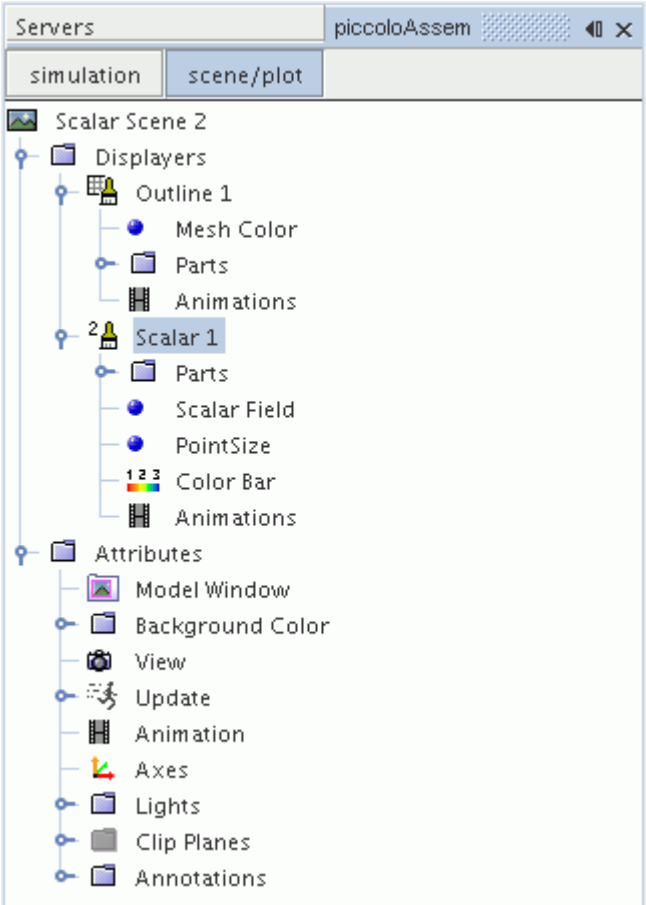
7. Click **scene/plot**.
8. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Scalar Field**.



9. Set *Function* to **Temperature**.

Modify the contour style to have a smooth transition between the different temperature values in the scene:

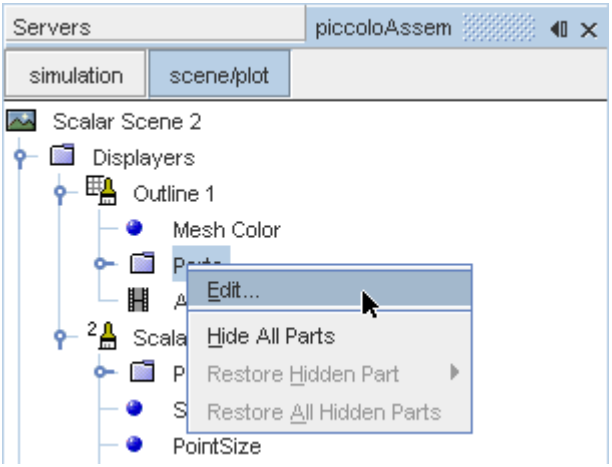
10. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1**.



11. Set *Contour Style* to **Smooth Filled**.

Adjust the outlines in the scene to show only the plane section:

12. Right-click **Scalar Scene 2** > **Displayers** > **Outline 1** > **Parts** and select **Edit...**.

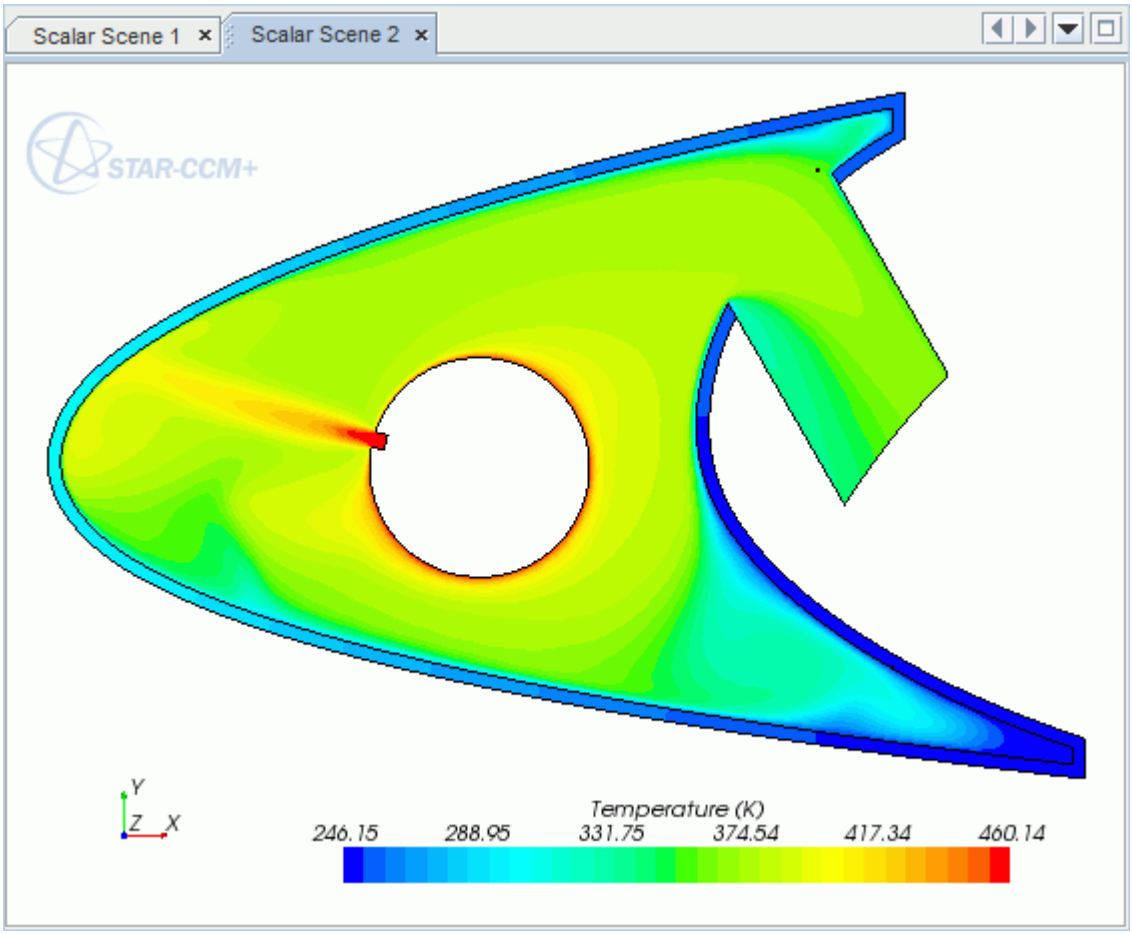


The *Parts* dialog appears.

13. Right-click on a blank area and choose **Deselect All**.

14. Select the **Derived Parts** > **plane section** node.
15. Click **OK**.
16. Position the scene to show the side view of the geometry.
17. Click **simulation**.
18. Click (**Save**) to save the simulation.

The finished scene is shown below.



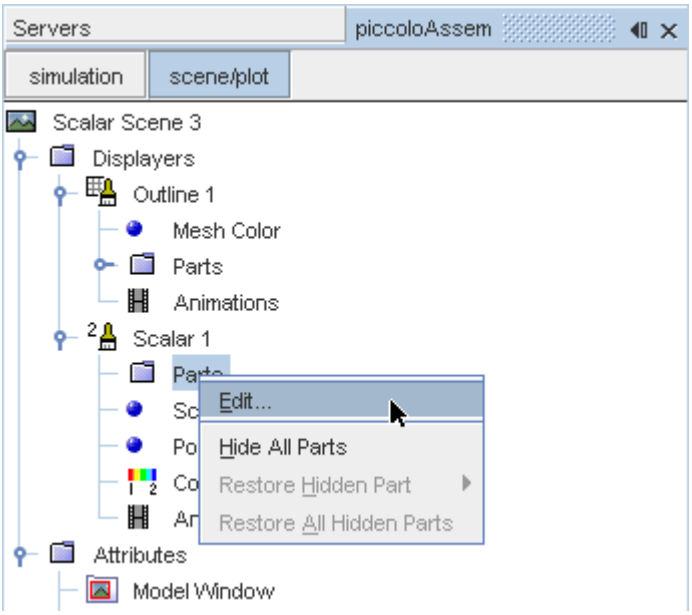
Visualizing the Wing Skin Temperature

Create another scalar scene to display the temperature on the wing skin geometry.

To create another scalar scene:

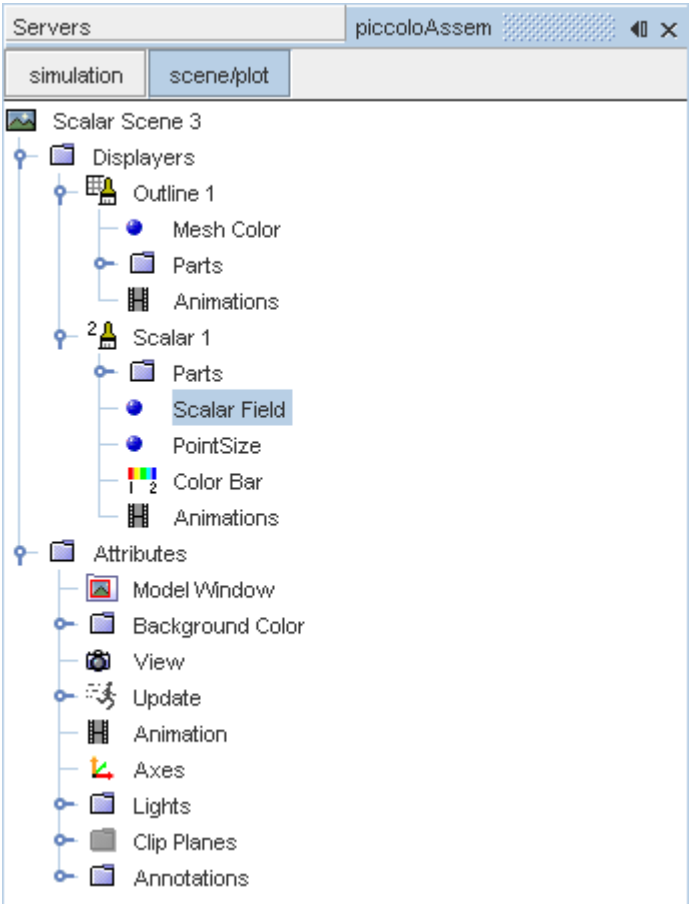
1. Right-click the **Scenes** node in the explorer pane and select **New Scene** > **Scalar**.

   A new scalar scene, *Scalar Scene 3*, is created and selected in the *Graphics* window.
2. Click **scene/plot**.
3. Right-click **Scalar Scene 3** > **Displayers** > **Scalar 1** > **Parts** and select **Edit...**.



The *Parts* dialog appears.

4. Expand the **Regions** node and select the **Leading Edge** node.
5. Click **OK**.
6. Navigate to **Scalar Scene 3** > **Displayers** > **Scalar 1** > **Scalar Field**.



7. Set *Function* to **Temperature.**

To have a smooth transition between the different temperature values, edit the contour style:

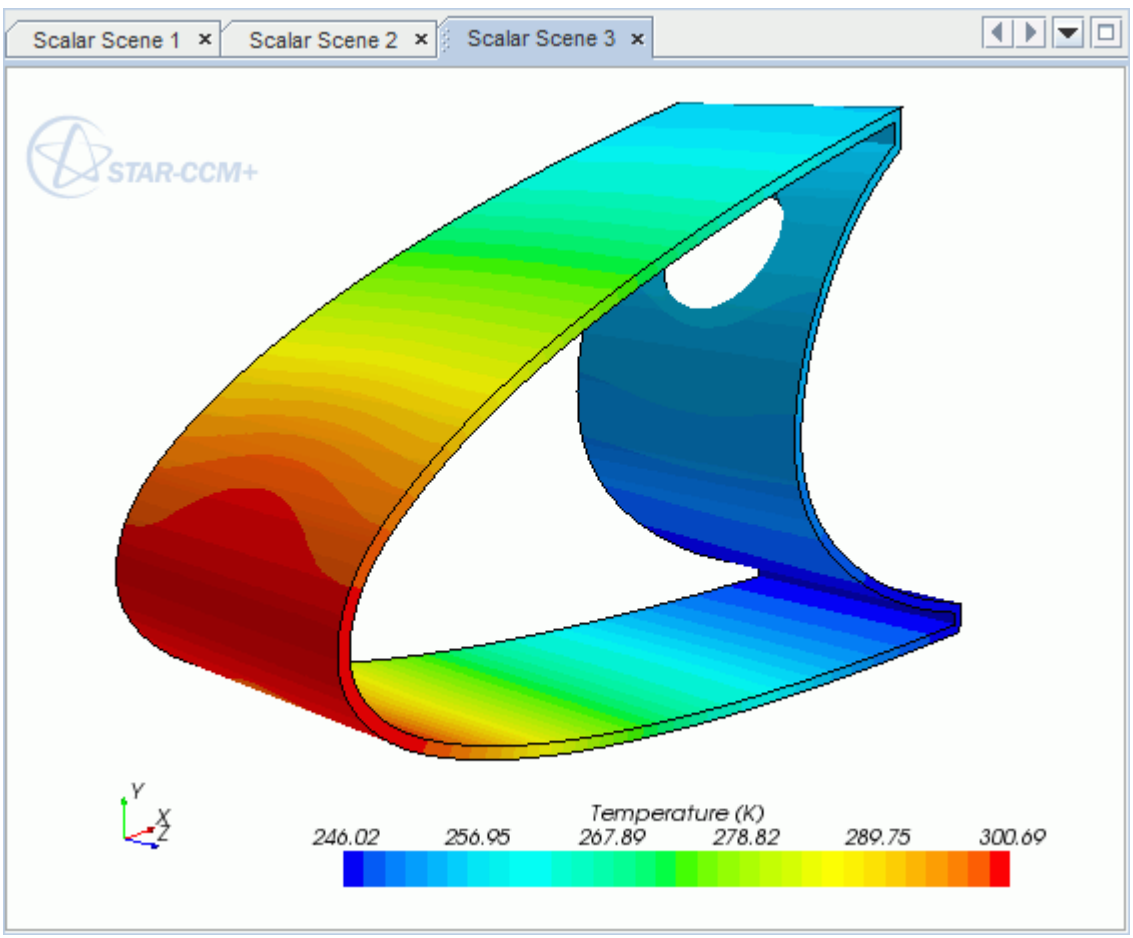
8. Navigate to **Scalar Scene 3** > **Displayers** > **Scalar 1**.
9. Set *Contour Style* to **Smooth Filled**.

Adjust the outlines to appear on the solid region only:

10. Right-click **Scalar Scene 3** > **Displayers** > **Outline 1** > **Parts** and select **Edit...**.

    The *Parts* dialog appears.
11. Expand the **Regions** node and deselect the **Internal Air** node.
12. Click **OK**.
13. Reorient the geometry to get a better view of the leading edge.
14. Click **simulation**.
15. Save the simulation.

The finished scene is shown below.

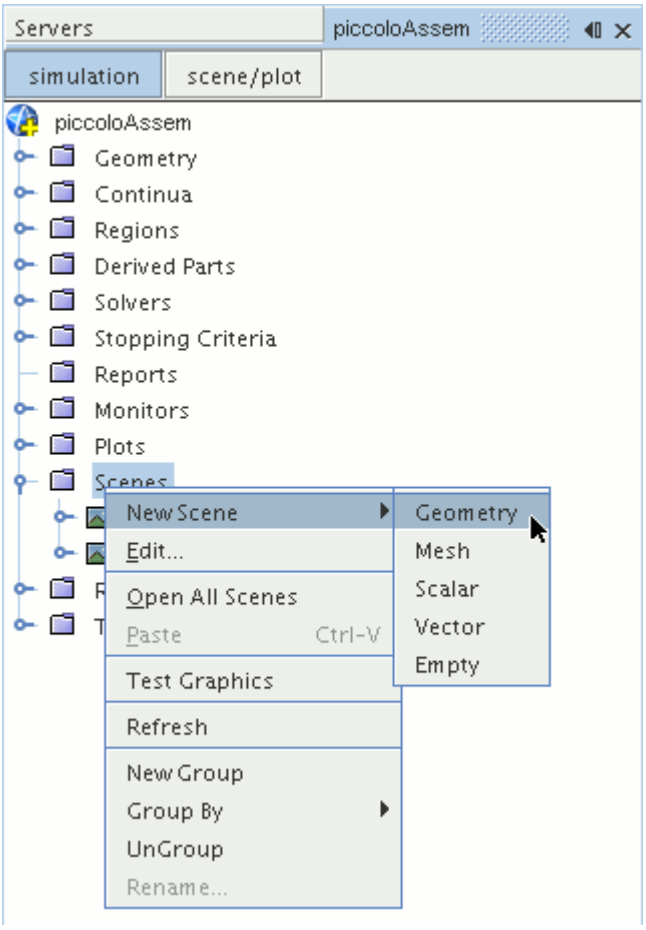


Creating a Streamlines Scene

Create streamlines to get a better understanding of how the hot air flows inside the leading edge of the wing.

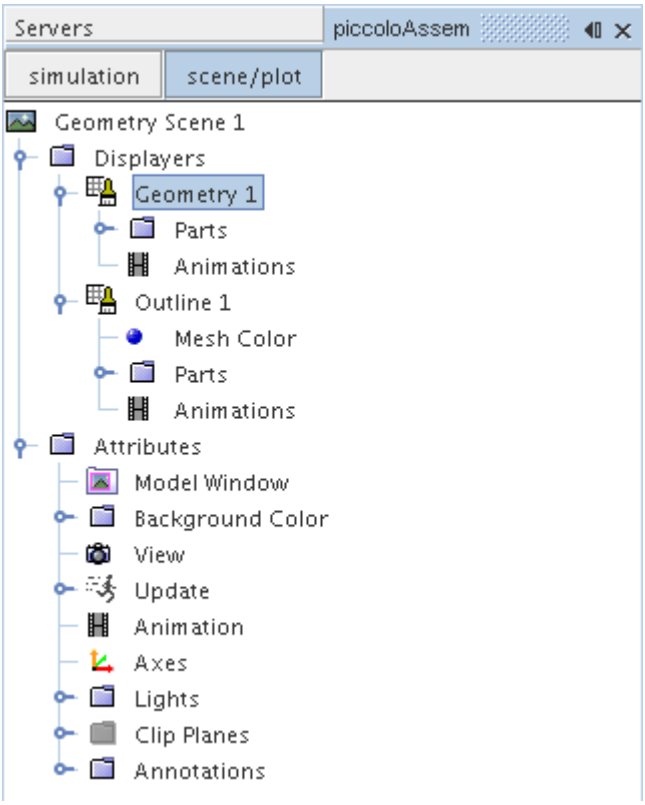
First, create a geometry scene:

1. Right-click the **Scenes** node in the object tree and select **New Scene** > **Geometry**.



   A new geometry scene, *Geometry Scene 1*, is created and selected in the *Graphics* window.

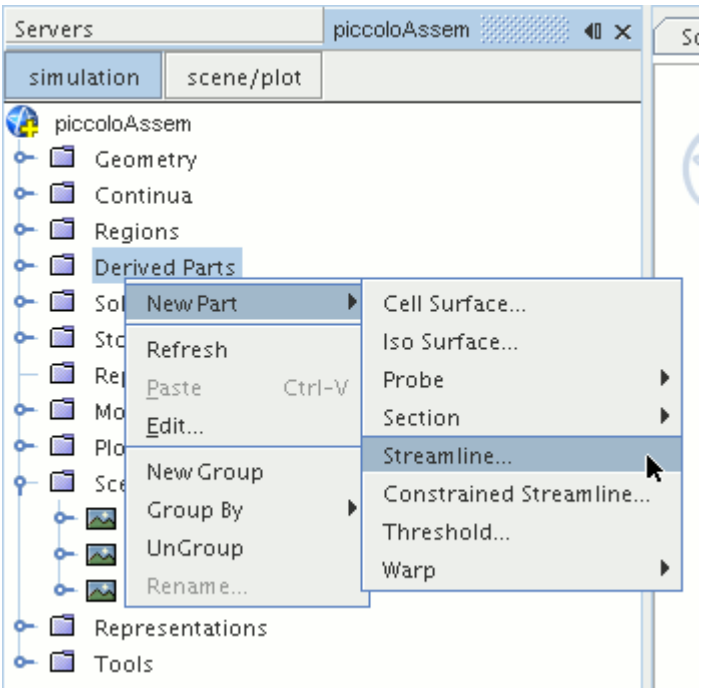
2. Click **scene/plot**.
3. Navigate to the **Geometry Scene 1** > **Displayers** > **Geometry 1** node.



4. Set *Opacity* to `0.1`.

Next, create the streamlines:

5. Click **simulation**.
6. Right-click the **Derived Parts** node and select **New Part** > **Streamline...**.

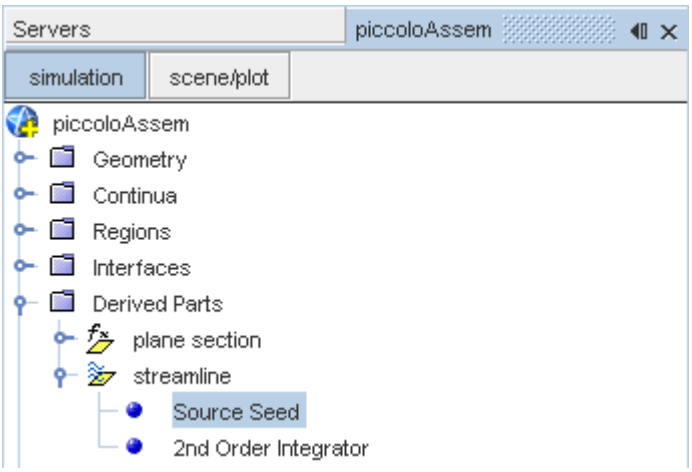


The *Create Streamline* dialog appears.

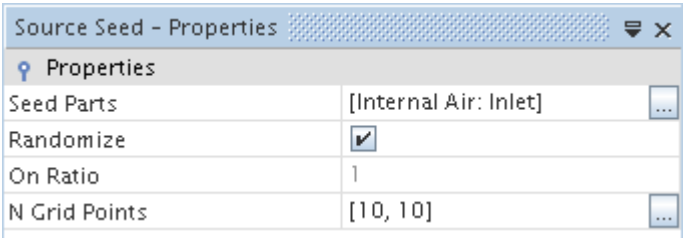
7. In the *Input Parts* box, click **Select...**.
8. In the *Select Objects* dialog, expand the **Regions** node and deselect **Leading Edge**.
9. Click **OK** to close the *Select Objects* dialog.
10. Make sure that **Part Seed** is selected in the *Seed Mode* section.
11. Set *Seed Parts* to **Regions** > **Internal Air** > **Boundaries** > **Inlet**.
12. Set *Part U-Resolution* to `10`.
13. Set *Part V-Resolution* to `10`.
14. Make sure that **New Streamline Displayer** is selected in the *Display* section.
15. Click **Create** followed by **Close**.

The streamlines do not yet appear in the geometry scene. Follow the steps that are outlined below to modify the streamlines:

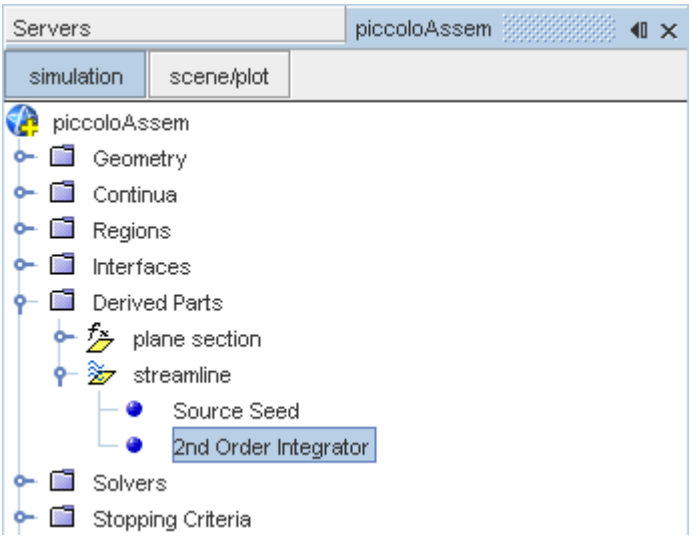
16. Navigate to **Derived Parts** > **Streamline** > **Source Seed**.



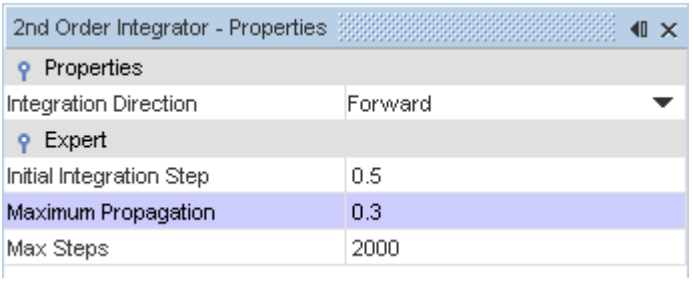
17. Activate the *Randomize* option.



18. Navigate to **Derived Parts** > **Streamline** > **2nd Order Integrator**.



19. Set *Maximum Propagation* to `0.3`.



Adjust the appearance of the streamlines to display tubes instead of the default lines:

20. Click **scene/plot**.
21. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1**.
22. Set *Mode* to **Tubes** and *Width* to `0.00025`.

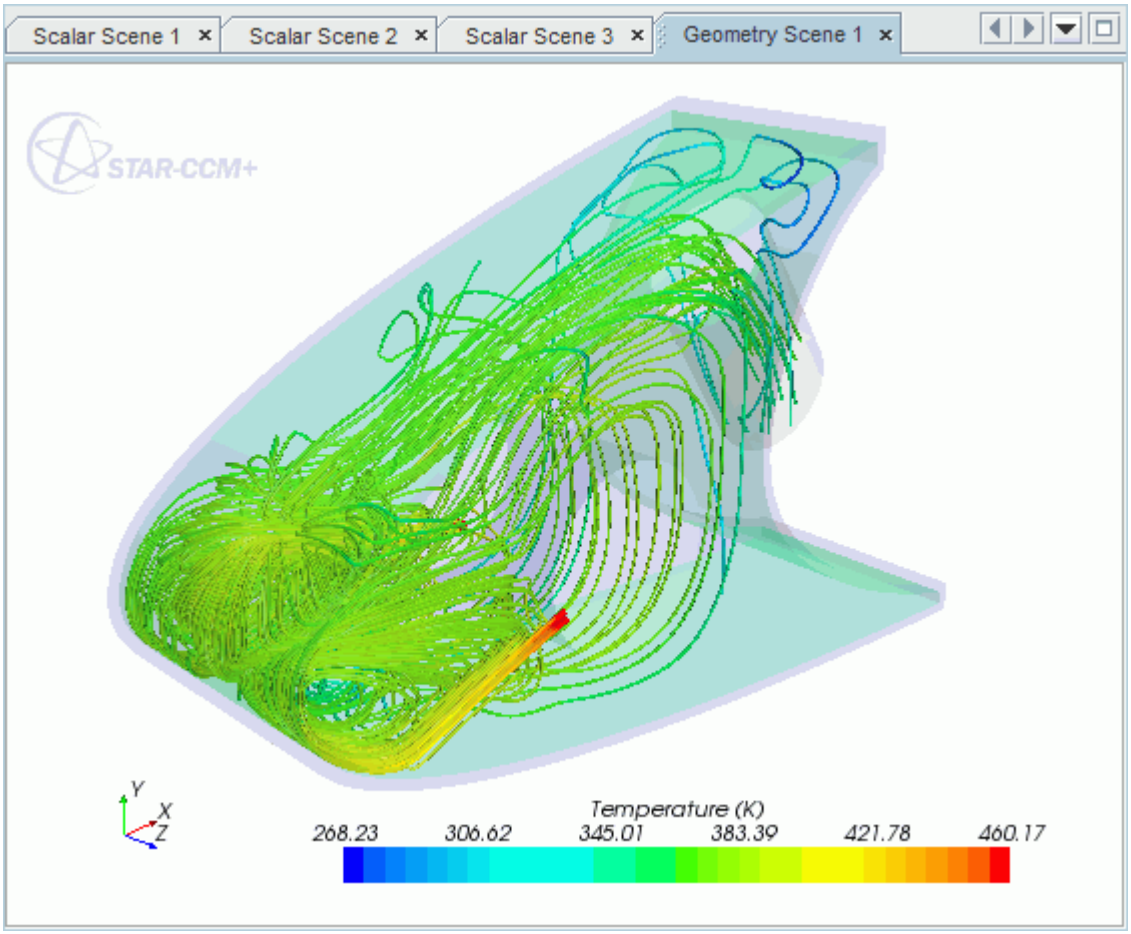
Adjust the scalar field of the streamlines to display temperature:

23. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** > **Scalar Field**.
24. Set *Function* to **Temperature**.

Turn off the outlines in the geometry scene to have a better view of the streamlines.

25. Right-click the **Geometry Scene 1** > **Displayers** > **Outline 1** and select **Toggle Visibility**.
26. Reorient the geometry to get a better view of the leading edge.
27. Save the simulation.

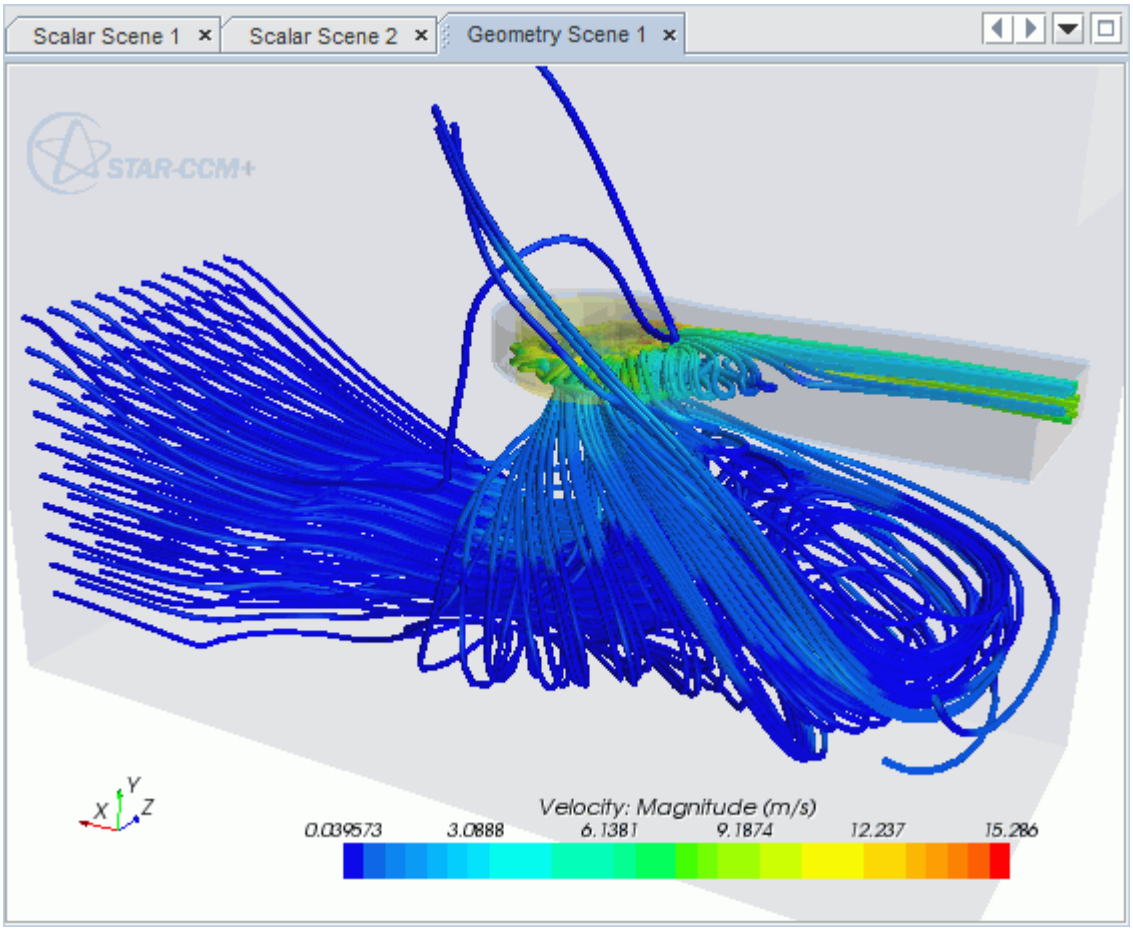
The finished scene is shown below.


Scalar Scene 1
Scalar Scene 2
Scalar Scene 3
Geometry Scene 1
STAR-CCM+
Y
X
Z
Temperature (K)
268.23
306.62
345.01
383.39
421.78
460.17

Rotating Flow Inside a Graphics Card

This tutorial takes you through the steps that are required to run a CFD calculation in the STAR-Creo environment.

For this tutorial, the model is a fully sealed graphics card that is located inside an empty, idealized PC tower case. The geometry that is used in this tutorial has been simplified on order to decrease the computational cost. As a consequence the mesh is relatively coarse at approximately 200,000 cells compared to the 3-4 million cells used for an industrial analysis.



Contents:

Opening the CAD Model
Selecting Fluid Physics Models
Setting the Rotating Region
Defining Boundary Conditions
Setting the Mesh Pipeline
Creating the Volumetric Mesh Controls
Modifying Solution Controls

Reviewing the Simulation Set Up
Generating The Surface Mesh
Generating the Volume Mesh
Building and Running
Post-Processing Results

Opening the CAD Model

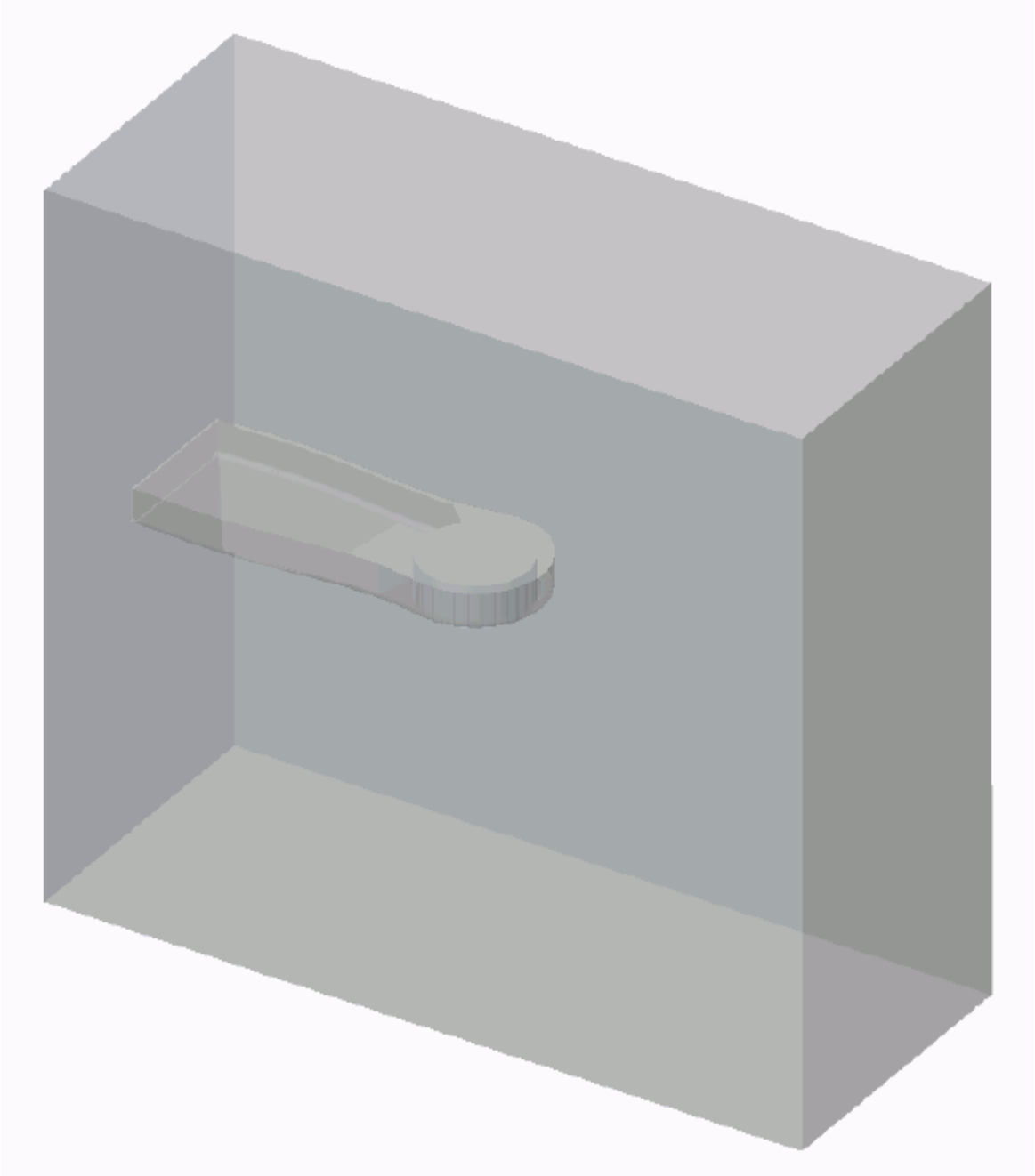
A directory containing the CAD geometry is provided with the STAR-CCM+ distribution.

This directory is located in `[STAR-CCM+_INSTALL_DIR]\doc\startutorialsdata\starcreo\data`.

1. Copy directory `data` to your own disk space and open the part file `gpuassem.asm` in STAR-Creo.

   The assembly is made up of two parts: the PC tower case (including the GPU housing), and the rotating region that encloses the blower fan of the GPU.

   The imported geometry is shown below.



The STAR-Creo ribbon appears at the top of the screen.The STAR-Creo object tree is located on the left of the screen.

To switch to this object tree:

2. Click (**STAR Tree**).

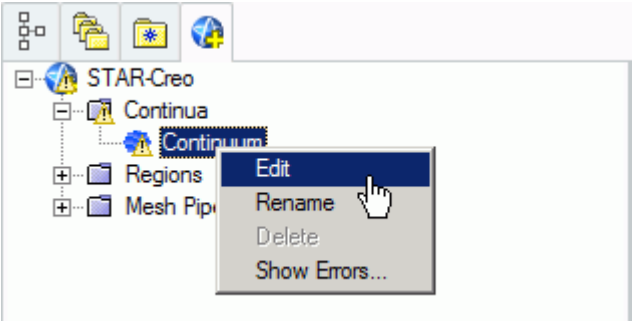
If these items are not visible, refer to Launching STAR-Creo Manually.

Selecting Fluid Physics Models

The first step in setting up this tutorial is to define the physics of the problem. For this case, the flow is turbulent and isothermal. Temperature calculations are on by default in STAR-Creo.

To turn temperature calculations off:

1. Right-click the **Continua** > **Continuum** node in the STAR-Creo tree and select **Edit**.

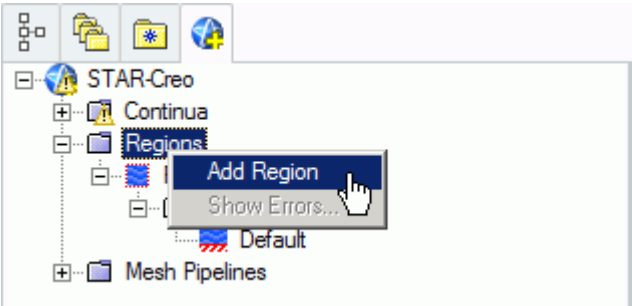


   The *Continuum Settings* dialog appears.
2. Deactivate the **Temperature** option in the *Optional Physics Models* box.
3. Click **OK**.

Setting the Rotating Region

Create the rotating region that contains the GPU fan:

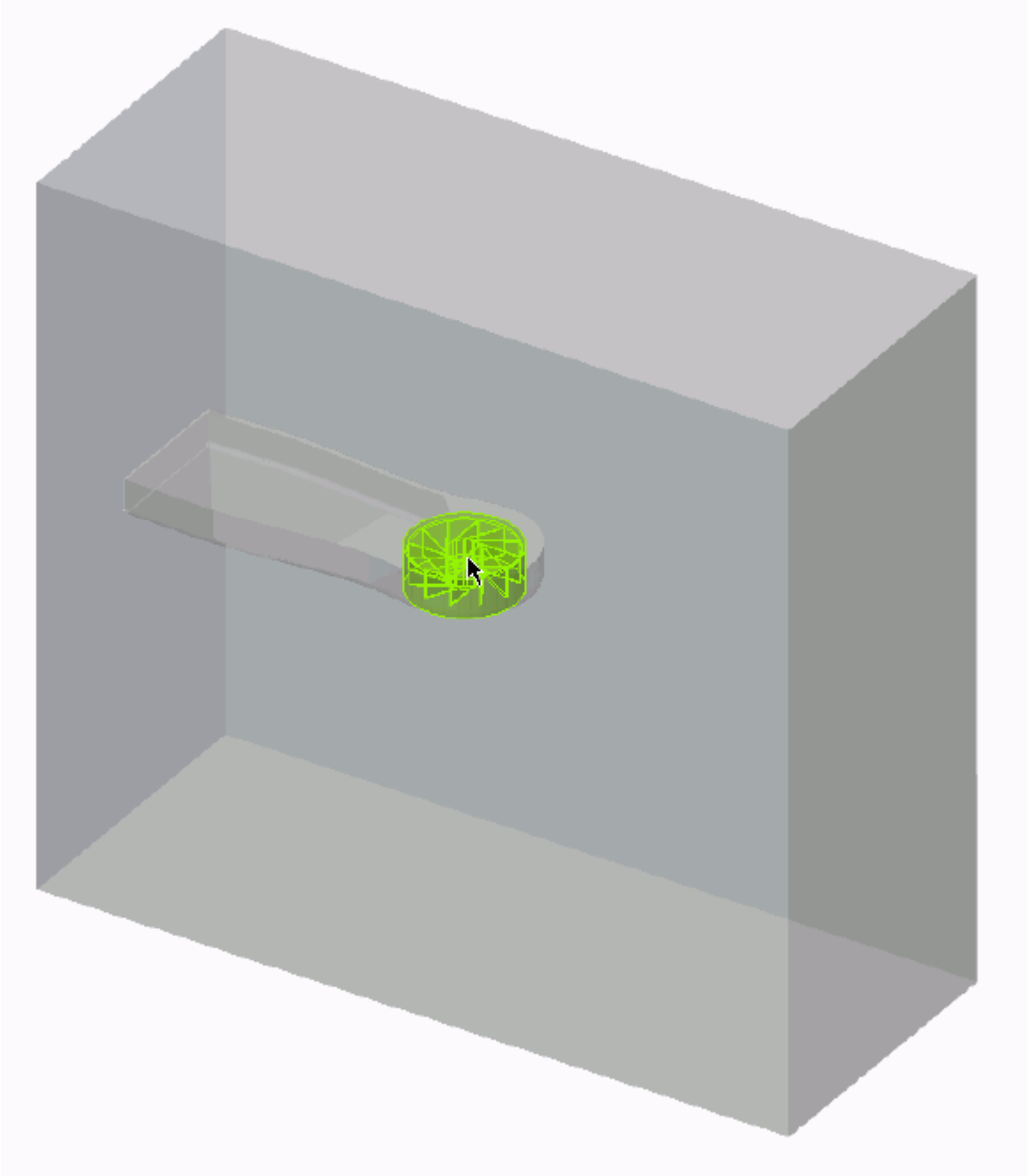
1. Right-click the **Regions** node in the STAR-Creo object tree and select **Add Region**.



   The *Region Settings* dialog appears.
2. Click (**Select Part**) to activate the Creo part selection mechanism.
3. Select the disc that is located inside the graphics card housing.

   You can select this part by moving the mouse over the fan area and right-clicking once. This action highlights the fan. Left-click to select the fan.

   Once selected, the part becomes highlighted as shown below.



4. Click **OK** to close the *Select* dialog.
5. Set *Name* to `Rotating Region`.
6. Activate the **Moving Reference Frame** option.
7. Set (*Rotational Velocity*) to `-2800 rpm`.

   Note: Make sure that you set the units to `rpm` using the drop-down menu before entering the value.

8. Set the origin of the axis of rotation to:

Property	Value
X	-122.1761
Y	-1.9434
Z	-19.0609

9. Set the components of the axis of rotation to:

Property	Value
(*i component*)	0
(*j component*)	1
(*k component*)	0

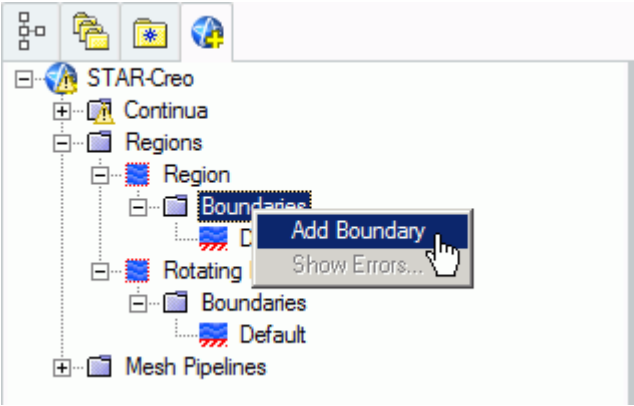
10. Make sure that the *Use Assembly Coordinate System* is activated.
11. Click **OK**.

Defining Boundary Conditions

Create the inlet and outlet boundary conditions for this simulation.

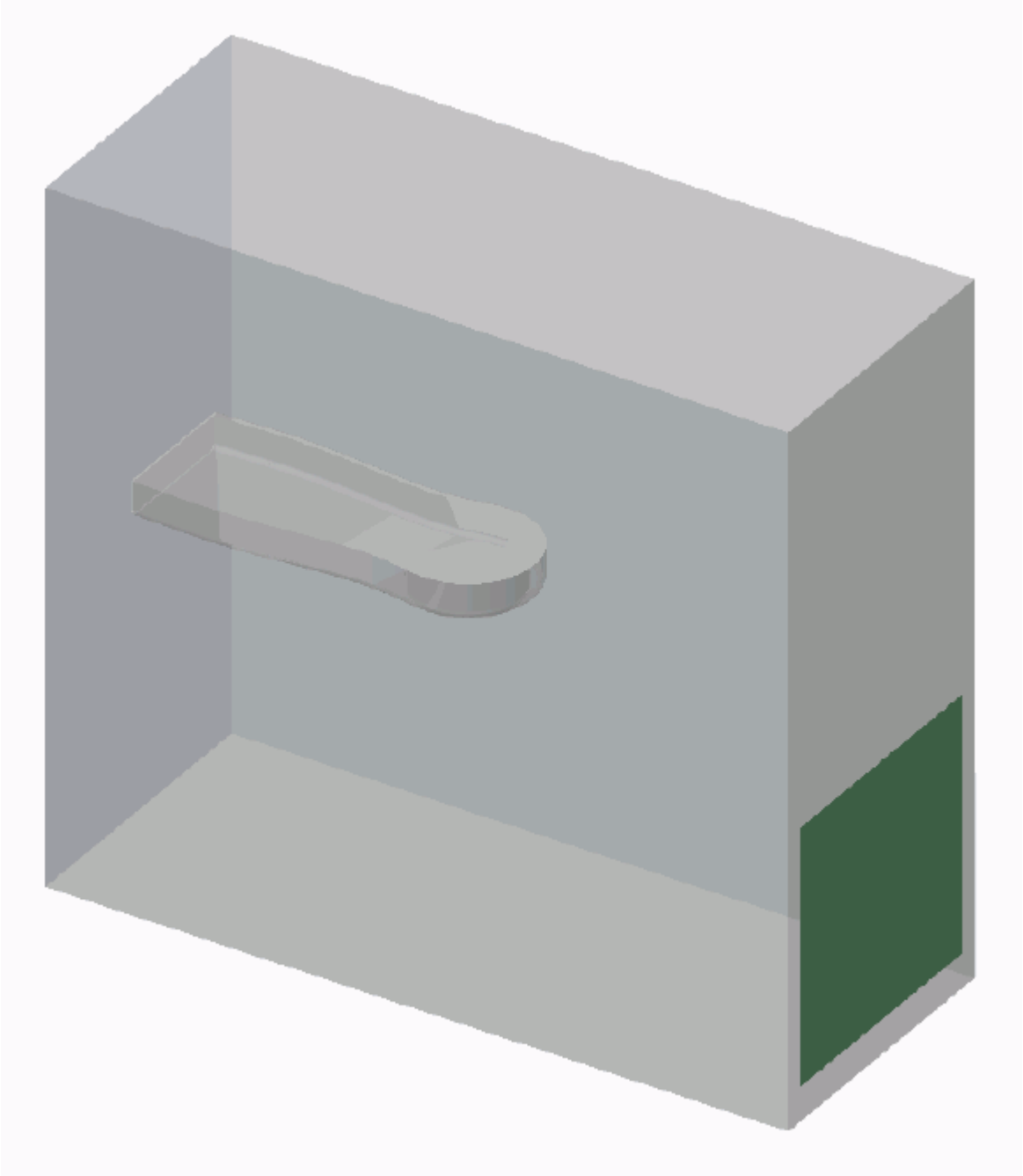
Create the inlet boundary first:

1. Right-click the **Regions** > **Region** > **Boundaries** node and select **Add Boundary**.



The *Boundary Settings* dialog appears.

2. Click (**Select Face**) to activate the Creo face selection mechanism.
3. Select the face that is highlighted below.



These selections are reported in the *Included Surfaces* box in the *Surface Sets* dialog.

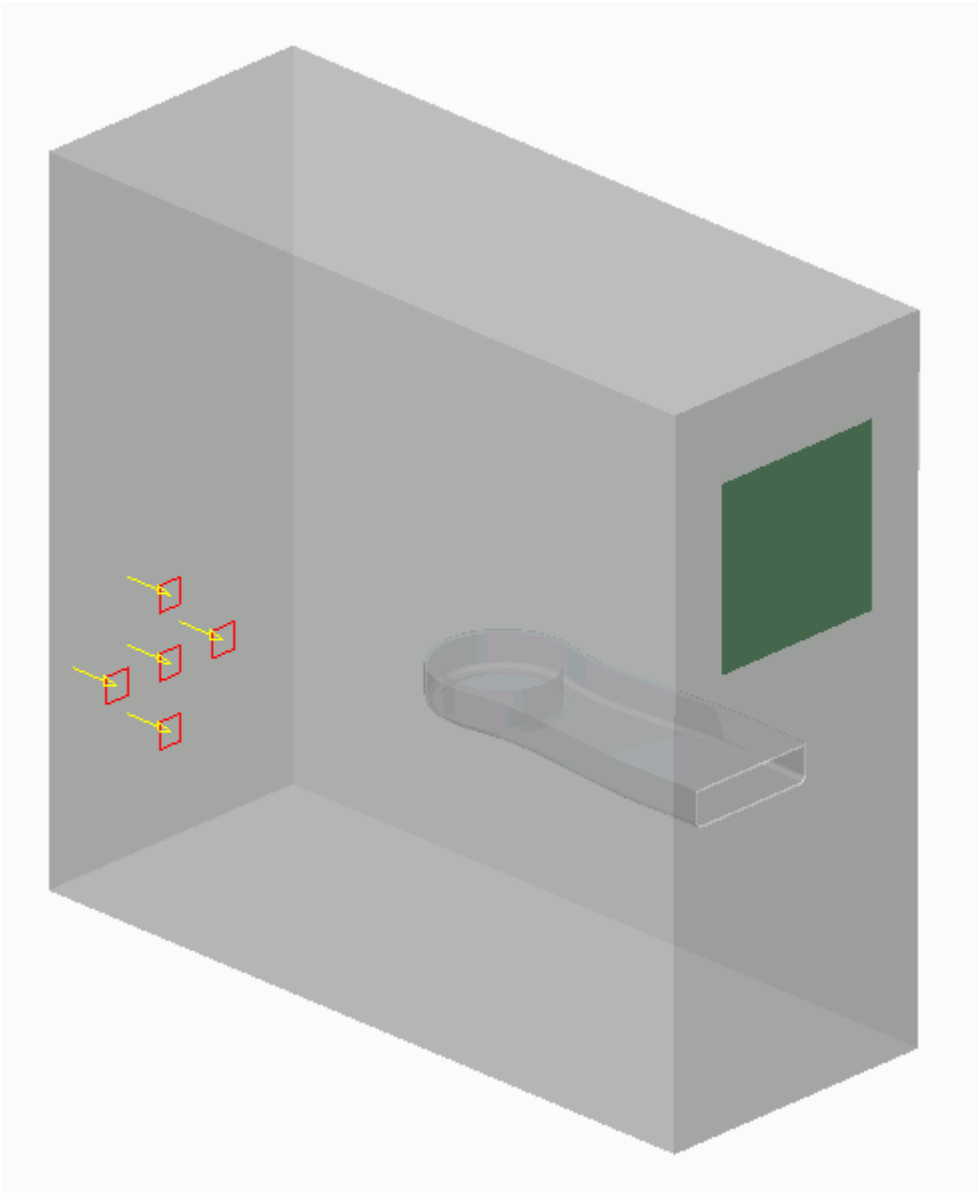
4. Click **OK** to close the *Surface Sets* dialog.
5. Set the following settings:

Parameter	Value
Type	**Velocity Inlet**
Name	`Case Inlet`
(*Normal Velocity*)	`0.1 m/s`

6. Click **OK**.

Create the case outlet boundary:

7. Using the same technique, create another boundary.
8. Select the face that is highlighted below.



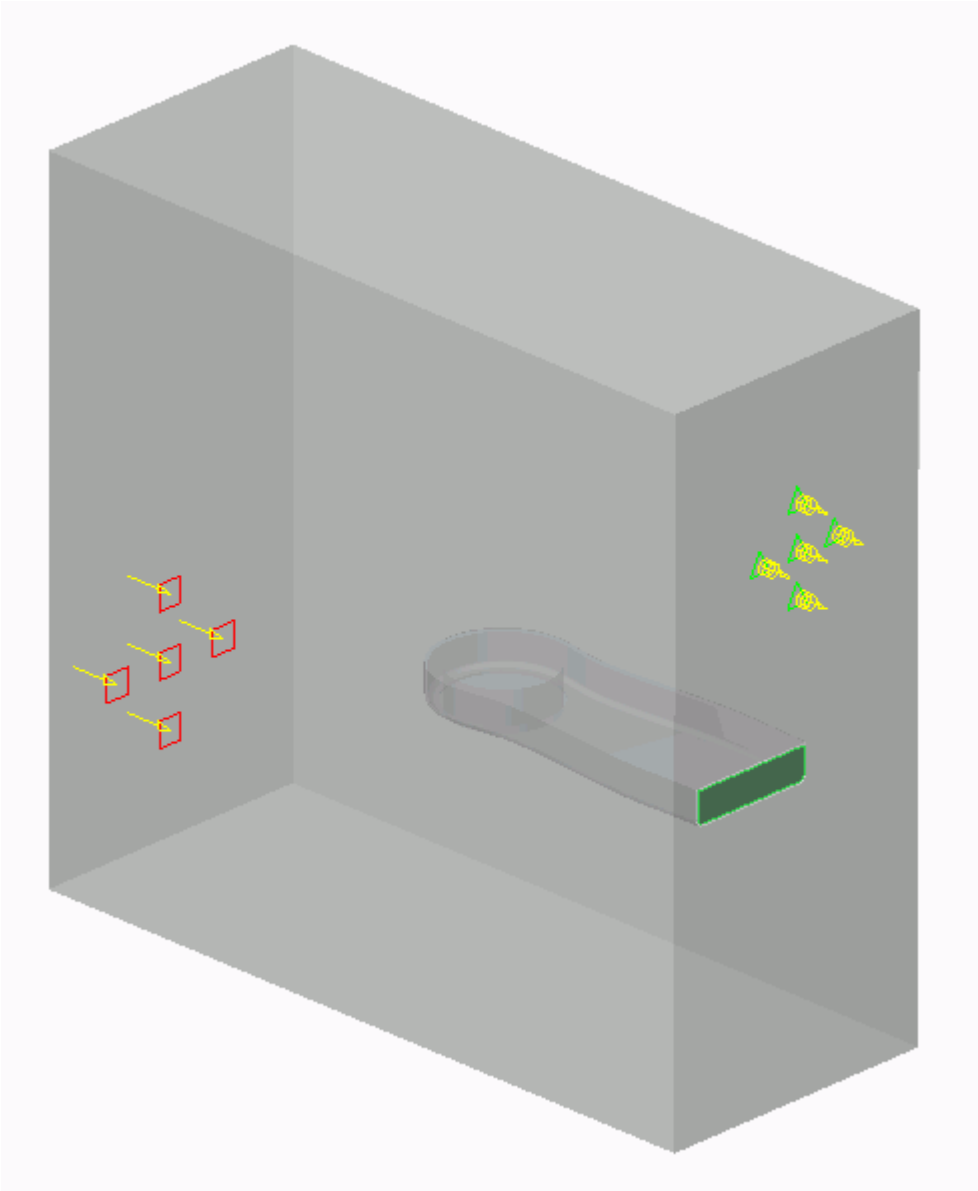
9. Set the following parameters:

Parameter	Value
Type	**Pressure Outlet**
Name	`Case Outlet`

10. Click **OK**.

Create the GPU outlet boundary:

11. Create another boundary and select the following face.



12. Set the following parameters:

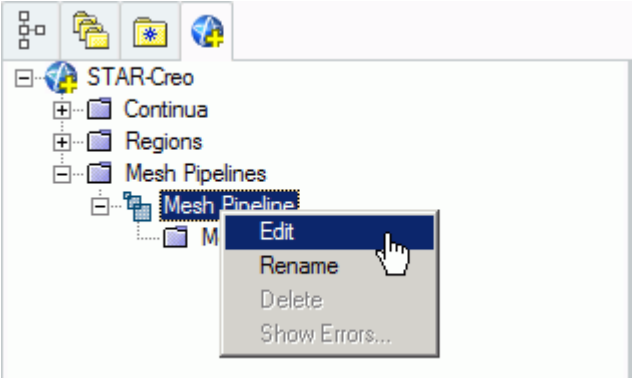
Parameter	Value
Type	**Pressure Outlet**
Name	`GPU Outlet`

13. Click **OK**.
14. Save your progress.

Setting the Mesh Pipeline

To obtain more accurate results, modify the mesh pipeline:

1. Right-click the **Mesh Pipelines** > **Mesh Pipeline** node in the STAR-Creo tree and select **Edit**.



The *Mesh Pipeline Settings* dialog appears.

2. Set the following parameters:

Parameter	Value
Base Size	
(**Base size option**)	**Absolute**
(*Base Size*)	`0.016 m`
Prism Layer	
(*Prism Layer Thickness*)	`20 %`

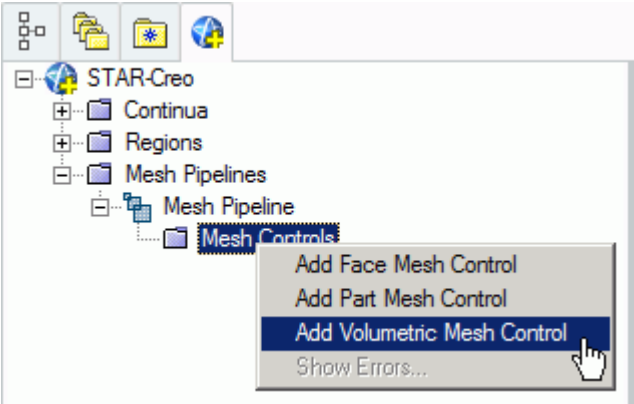
3. Click **OK**.

Creating the Volumetric Mesh Controls

Use volumetric mesh control regions to capture the finer details of the geometry model. Create three regions in total: a small region that closely surrounds the rotating region containing the fan, a region that encloses the GPU case, and a region that acts as an intermediary between the base size and the other volumetric control regions.

Create the volumetric mesh control for the rotating region:

1. Right-click the **Mesh Pipelines** > **Mesh Pipeline** > **Mesh Controls** node in the STAR-Creo tree and select **Add Volumetric Mesh Control**.



The *Volumetric Control* dialog appears.

2. Set *Name* to `Fan Box`.
3. Activate the *Absolute* option.
4. Set (*Mesh Size*) to `0.002 m`.
5. Enter the following coordinates for Point 1:

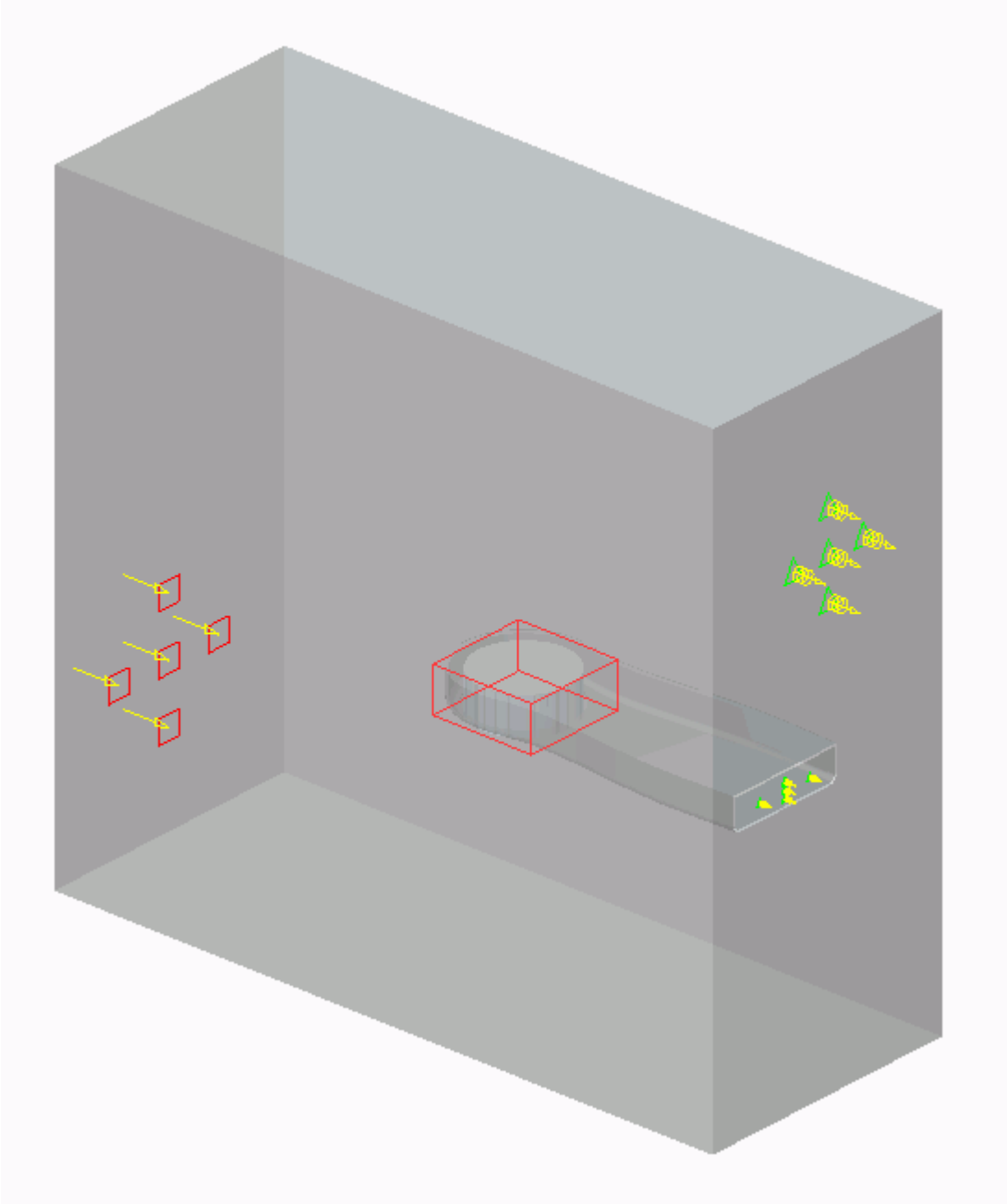
Coordinate	Value
X1	`-166.00`
Y1	`4.00`
Z1	`-61.00`

6. Enter the following coordinates for Point 2:

Coordinate	Value
X2	`-79.00`
Y2	`-34.00`
Z2	`25.00`

7. Click **Create**.

   The volumetric control region is shown below.



8. Click **OK**.

Create the volumetric mesh control for the GPU region:

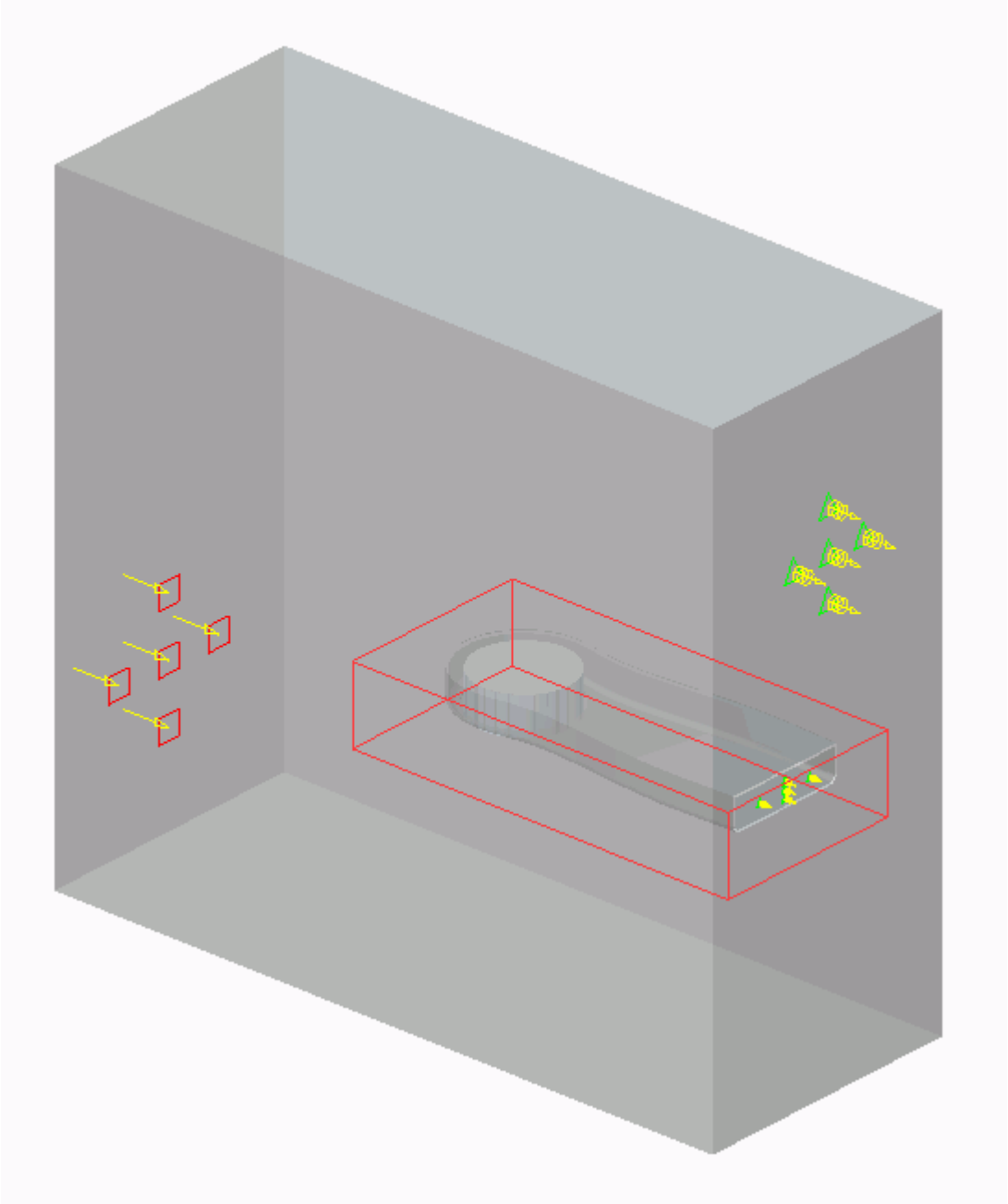
9. Using the same technique, create another volumetric mesh control with the following settings:

Parameter	Value
Name	`GPU Box`
Mesh Size	
(**Base size option**)	**Absolute**

Parameter	Value
(*Mesh Size*)	0.005 m
Shape	
X1	-383.00
Y1	16.00
Z1	-111.00
X2	-52.00
Y2	-49.00
Z2	49.00

10. Click **Create**.

    The volumetric control region is shown below.



11. Click **OK**.

Create the volumetric mesh control for the PC tower case:

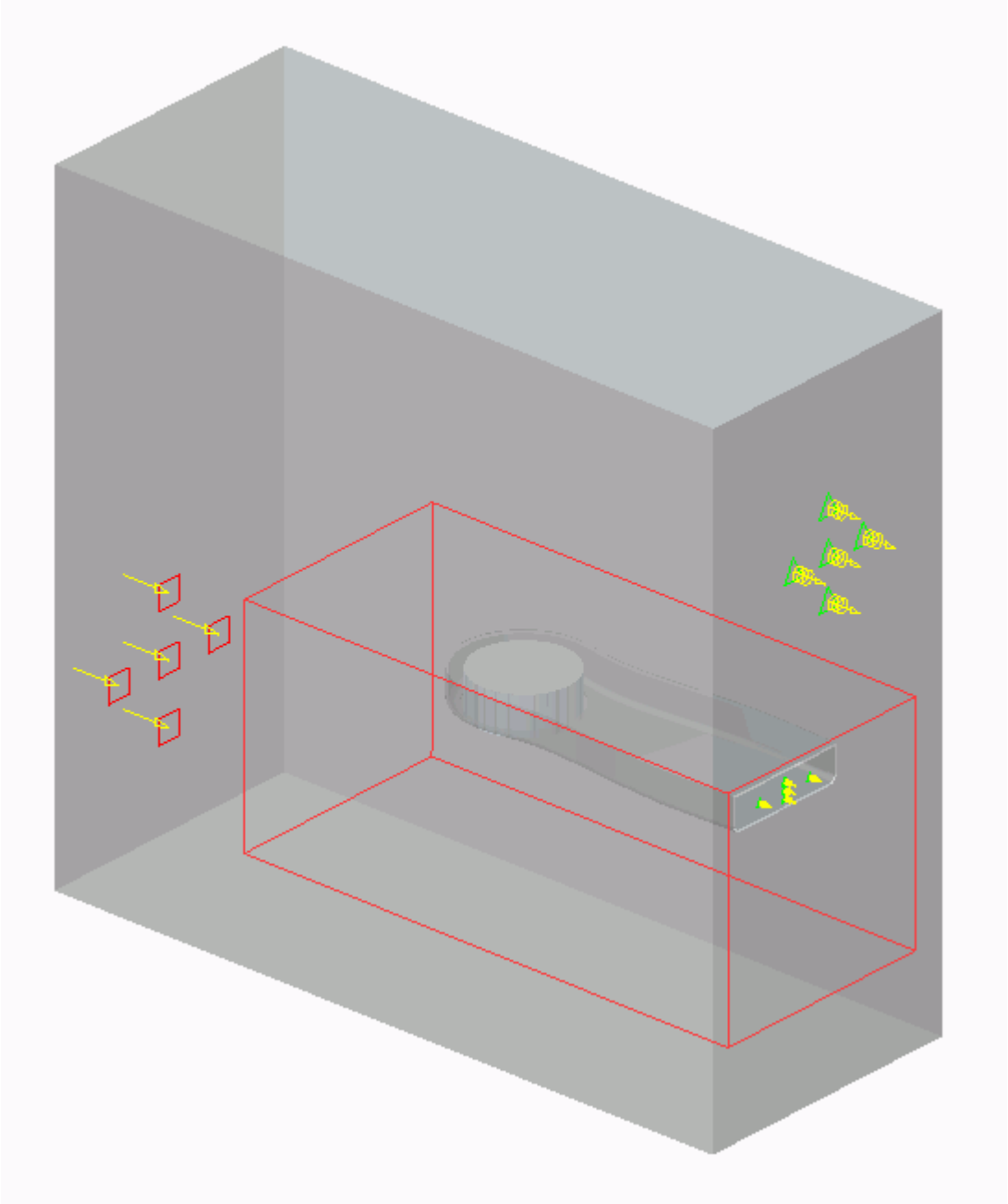
12. Using the same technique, create another volumetric mesh control with the following settings:

Parameter	Value
Name	`Case Box`
Mesh Size	
(**Base size option**)	**Absolute**

Parameter	Value
(*Mesh Size*)	0.009 m
Shape	
X1	-383.00
Y1	30.00
Z1	-111.00
X2	43.00
Y2	-159.00
Z2	77.00

13. Click **Create**.

    The volumetric control region is shown below.

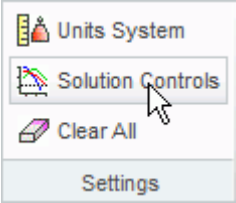


14. Click **OK**.

Modifying Solution Controls

Modify the maximum number of iterations and the solution accuracy:

1. Click **Solution Controls** on the STAR-Creo ribbon.



The *Solution Controls* dialog appears.

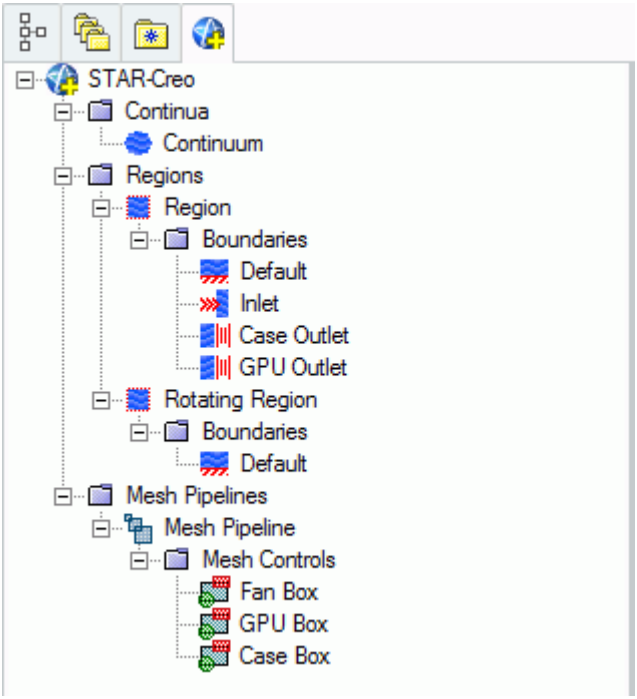
2. Set the following parameters:

Parameter	Value
(*Number of Iterations*)	300
(*Convergence Level*)	0.01
(*Momentum Under-Relaxation Factor*)	0.85

3. Click **OK**.
4. Save your progress.

Reviewing the Simulation Set Up

The final STAR-Creo object tree is shown below.

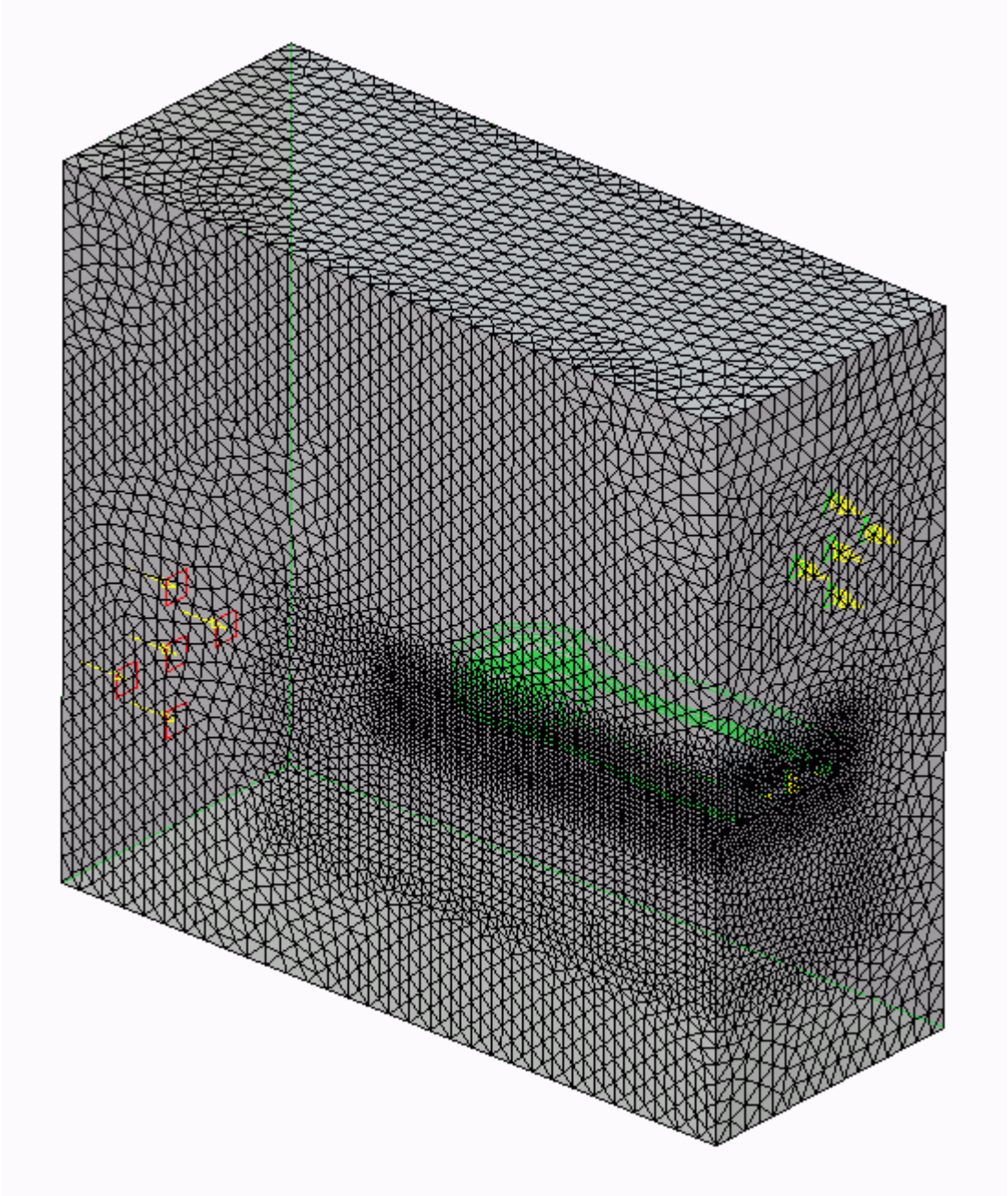


Generating The Surface Mesh

Generate the surface mesh:

1. From the STAR-Creo ribbon, click **Surface Mesh**.

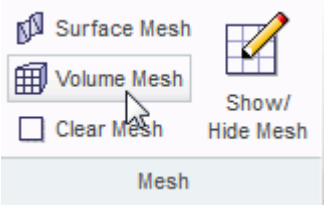
The surface mesh is shown below.



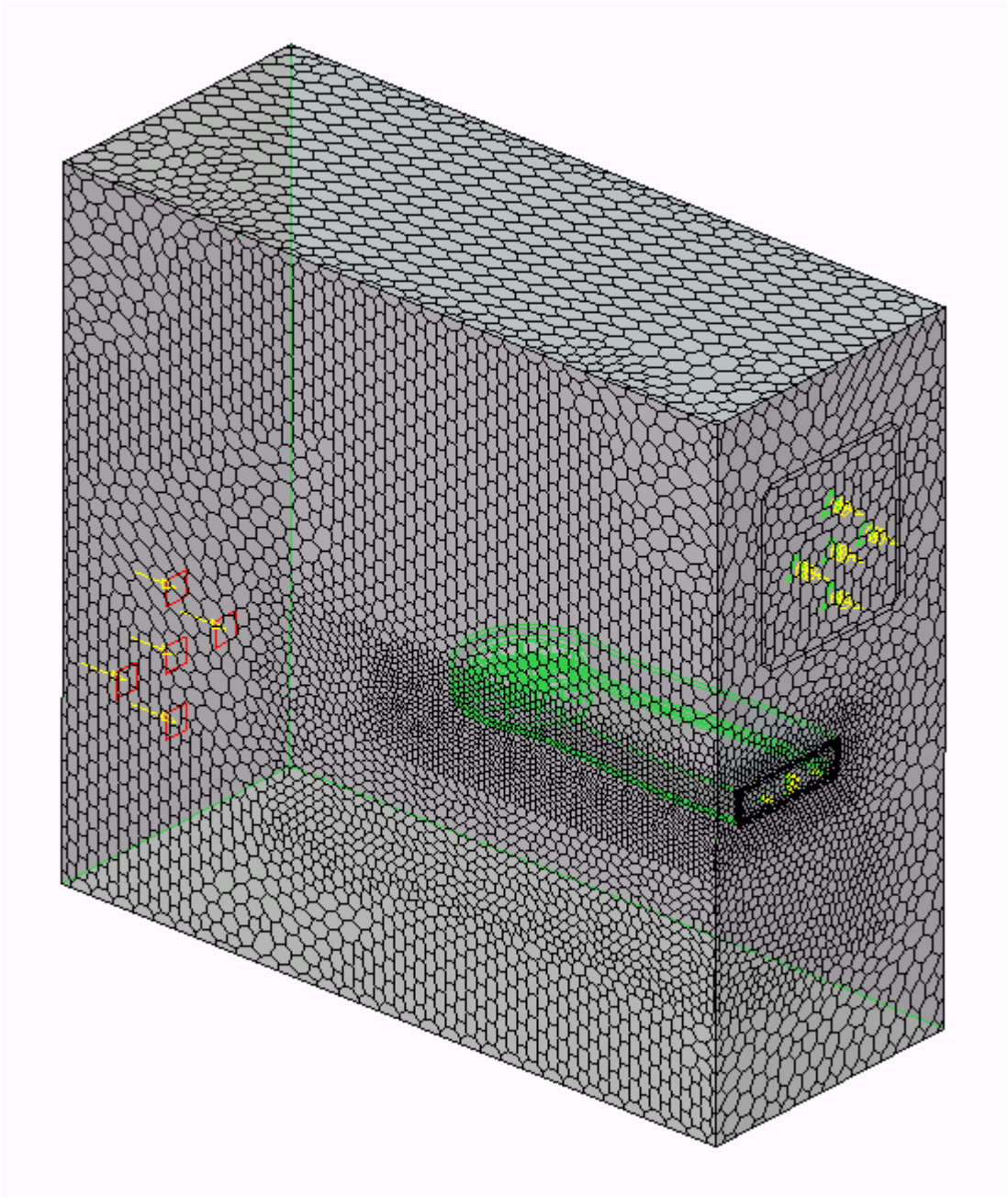
Generating the Volume Mesh

Generate the volume mesh:

1. From the STAR-Creo ribbon, click **Volume Mesh**.



The volume mesh is shown below.

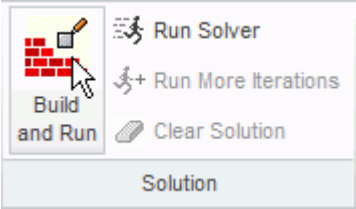


Building and Running

The case is now ready to run.

To run the simulation:

1. From the ribbon, click **Build and Run**.



The *Residuals* plot appears in the STAR-Creo *Output* window. You can use this plot to monitor the convergence of the solution.

When the solver is done, STAR-CCM+ launches.

Post-Processing Results

You perform all post-processing in STAR-CCM+. By default, a scalar scene, *Scalar Scene 1*, automatically appears in the *Graphics* window when the solver finishes. This scene displays the absolute total pressure.

The results that are contained in the following sections are representative. You may see some variation in the results that you obtain, depending on what platform and package you are using.

1. In STAR-CCM+, click **File** > **Save** to save the simulation.

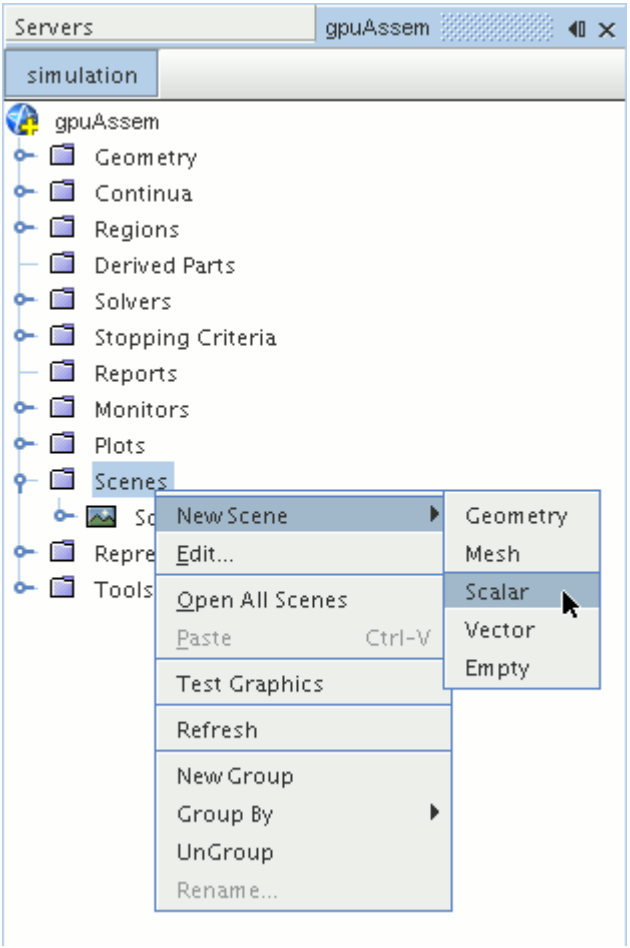
Contents:

Creating a Scalar Scene
Creating A Streamlines Scene
Creating a Vector Scene
Creating a Streamlines Scene
Creating A Scalar Scene
Visualizing the Wing Skin Temperature
Creating a Streamlines Scene
Creating A Scalar Scene
Creating a Streamlines Scene
Creating a Scalar Scene
Modifying The Scalar Scene
Creating a Streamlines Scene

Creating A Scalar Scene

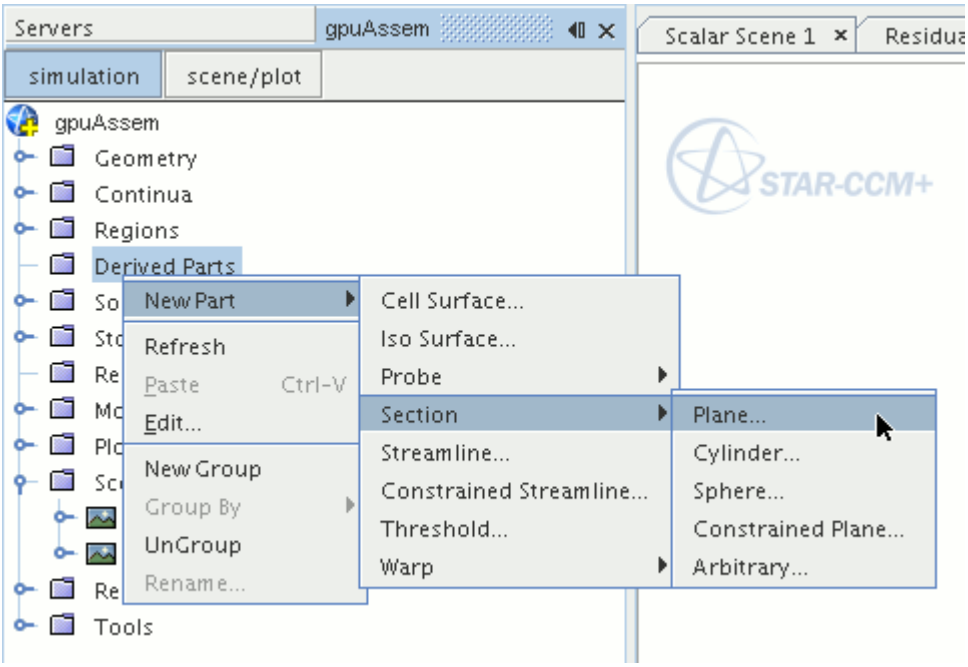
Create a scalar scene containing a cross-section of the GPU and plot velocity magnitude on this plane.

1. Right-click the **Scenes** node in the explorer pane and select **New Scene** > **Scalar**.



A new scalar scene, *Scalar Scene 2*, is created and selected in the *Graphics* window.

2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane...**.

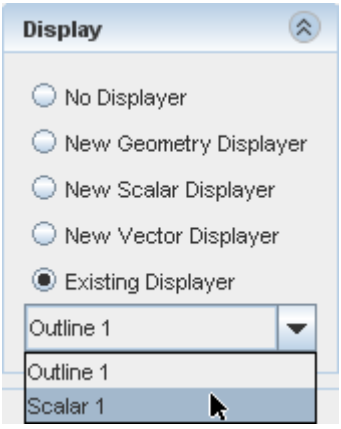


The *Create Plane Section* dialog appears.

3. In the *Plane Parameters* box, set the *origin* vector to `[0 m, -0.012 m, 0 m]` and the *normal* vector to `[0 m, 1 m, 0 m]`.

   Note that the units are in meters.

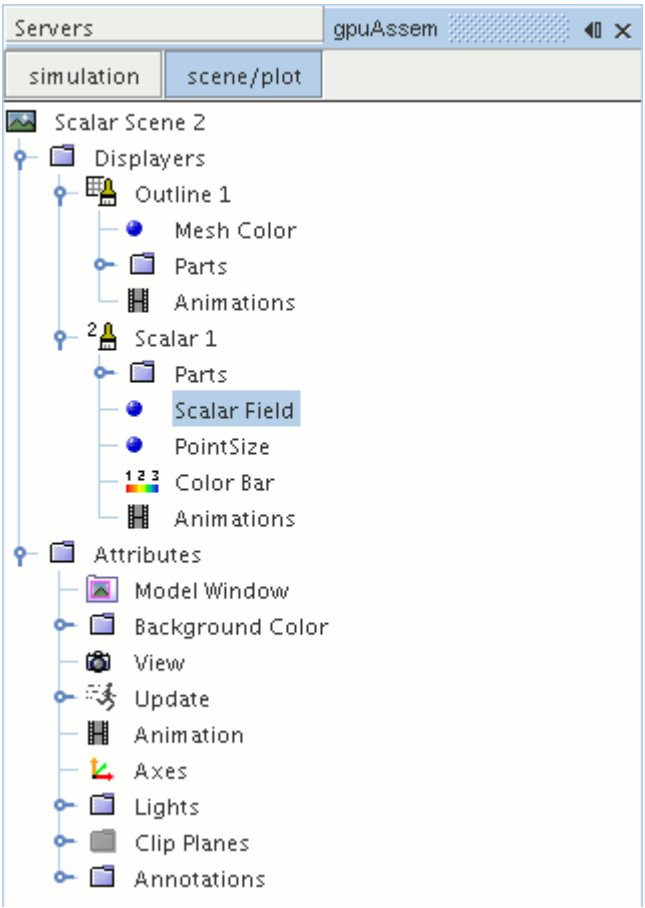
4. In the *Display* group box, select **Existing Displayer** and in the drop-down menu select **Scalar 1**.



5. Click **Create** and then **Close.**

Display velocity magnitude on the section plane:

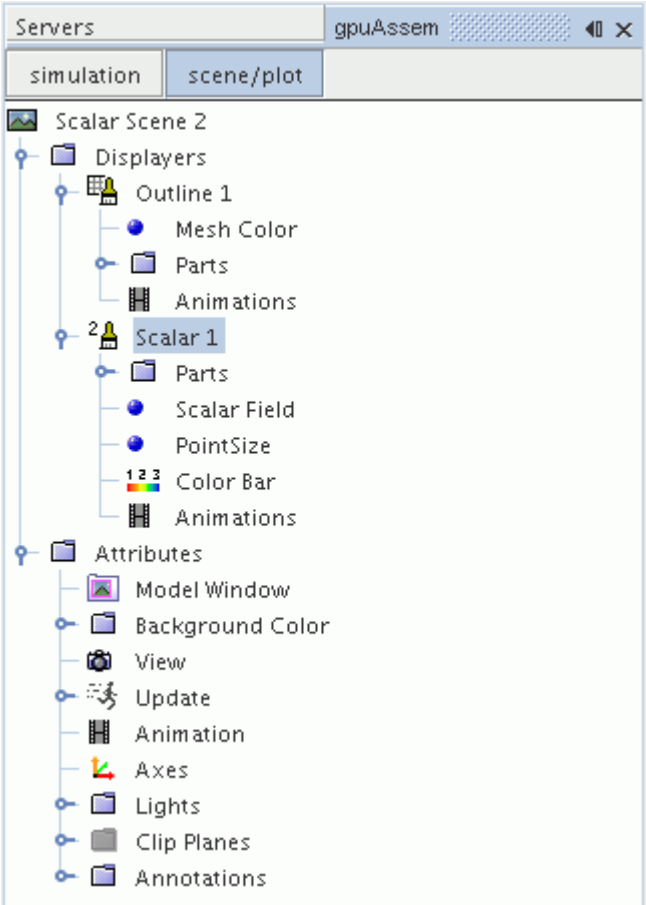
6. Click **scene/plot**.
7. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Scalar Field**.



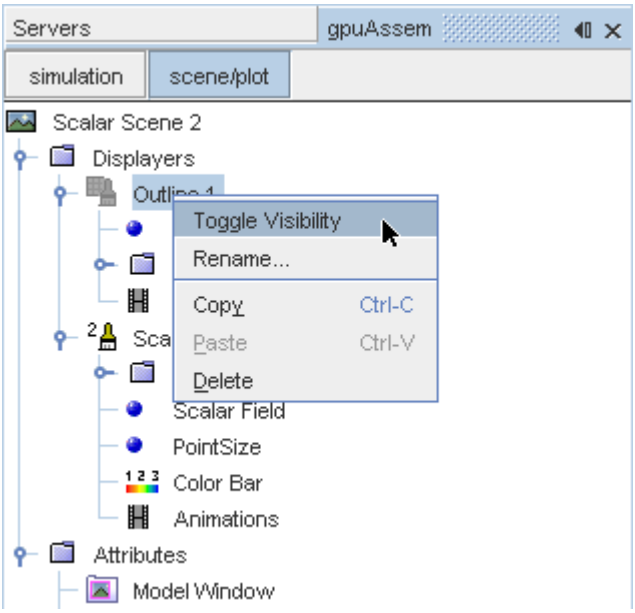
8. Set *Function* to **Velocity** > **Lab Reference Frame** > **Magnitude**.

To have a smooth transition between the different velocity values, set the contour style to smooth:

9. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1**.

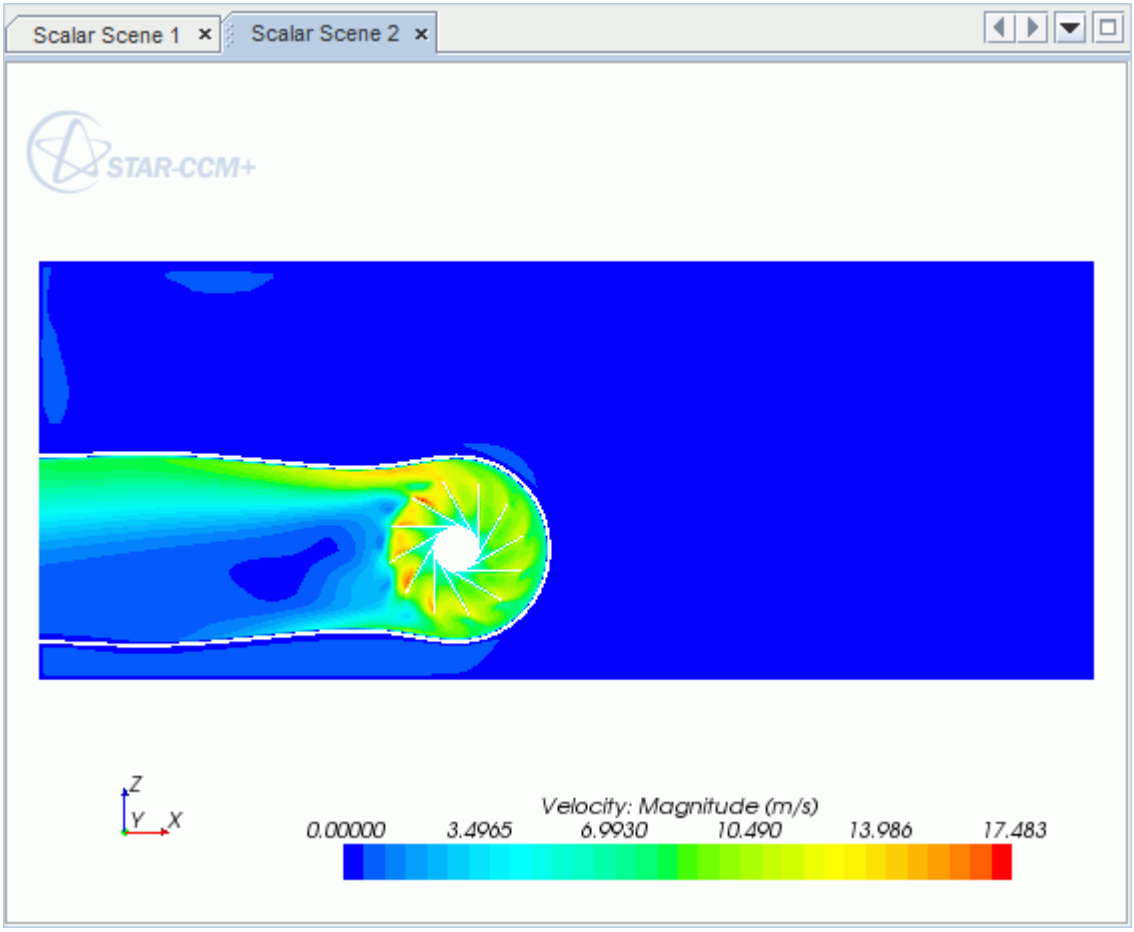


10. Set *Contour Style* to **Smooth Filled**.
11. To hide all outlines, right-click the **Scalar Scene 2** > **Displayers** > **Outline 1** node and select **Toggle Visibility**.



12. Click the *Scalar Scene 2* tab to activate the scene, and press the <S> key to change to the top view of this geometry.
13. Click **simulation**.
14. Click (**Save**).

The finished scene is shown below.

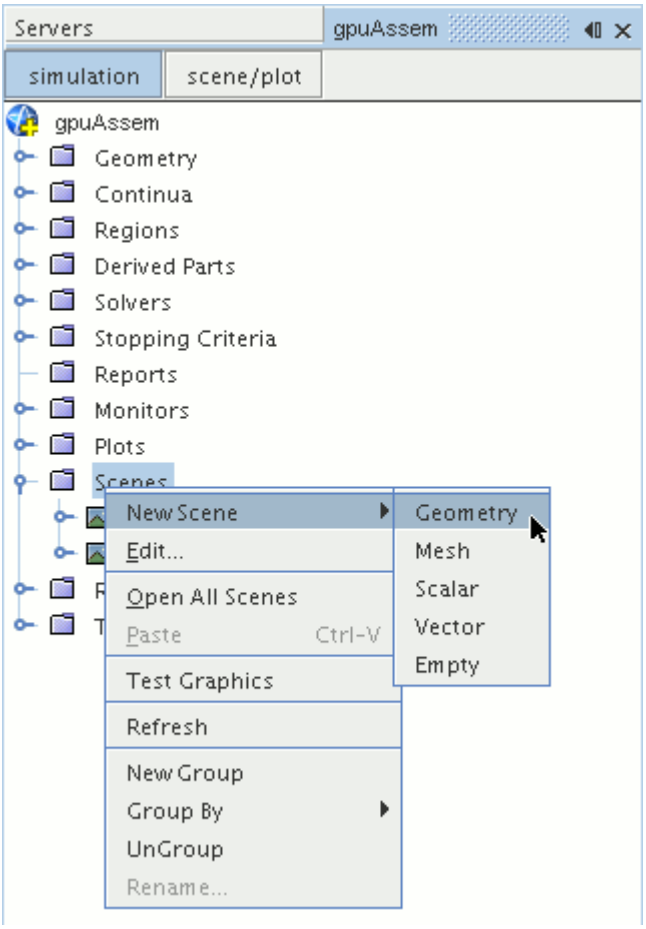


Creating a Streamlines Scene

Create streamlines to get a better understanding of the air flow inside the PC tower case and graphics card.

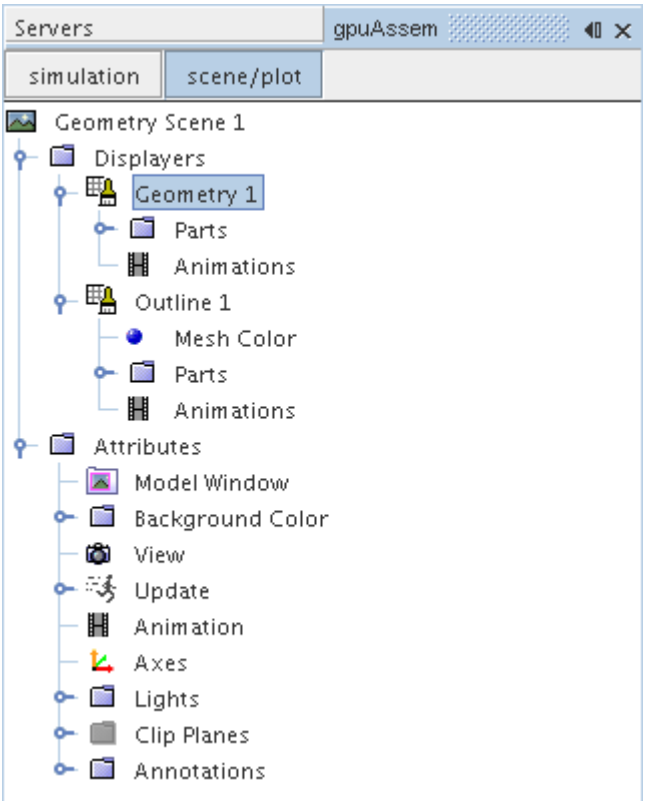
First, create a geometry scene:

1. Right-click the **Scenes** node in the object tree and select **New Scene** > **Geometry**.



A new geometry scene is created and selected in the *Graphics* window.

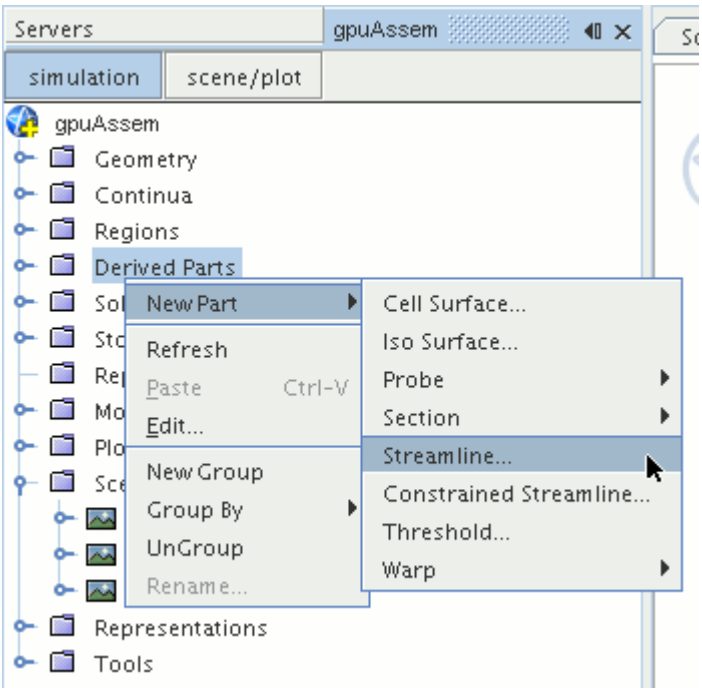
2. Click **scene/plot**.
3. Navigate to **Geometry Scene 1** > **Displayers** > **Geometry 1** node.



4. Set *Opacity* to `0.1`

Create the streamlines:

5. Click **simulation**.
6. Right-click the **Derived Parts** node and select **New Part** > **Streamline...**.



The *Create Streamline* dialog appears.

7. Make sure that **Part Seed** is selected in the *Seed Mode* section.
8. Set *Seed Parts* to **Regions** > **Region** > **Boundaries** > **Case Inlet**.
9. Set *Part U-Resolution* to `10`.
10. Set *Part V-Resolution* to `10`.
11. Make sure that **New Streamline Displayer** is selected in the *Display* section.
12. Click **Create** followed by **Close**.

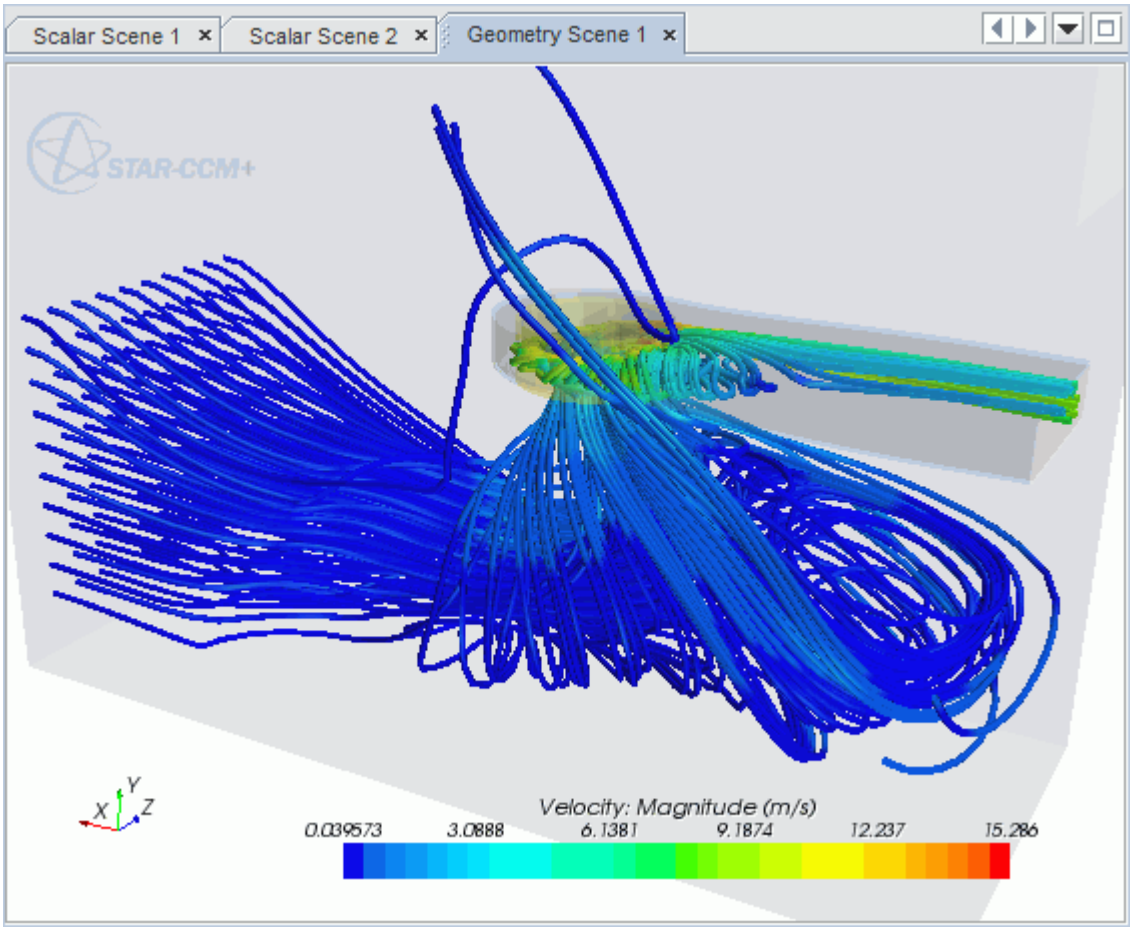
Adjust the appearance of streamlines to display tubes instead of the default lines:

13. Click **scene/plot**.
14. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** and set *Mode* to **Tubes** and *Width* to `0.0020`.
15. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** > **Scalar Field** and set *Function* to **Velocity** > **Lab Reference Frame** > **Magnitude**.

Hide all the outlines in the scene:

16. Right-click the **Geometry Scene 1** > **Displayers** > **Outline 1** node and select **Toggle Visibility**.
17. Reposition the geometry to get a better view of the flow inside the case.
18. Save the simulation.

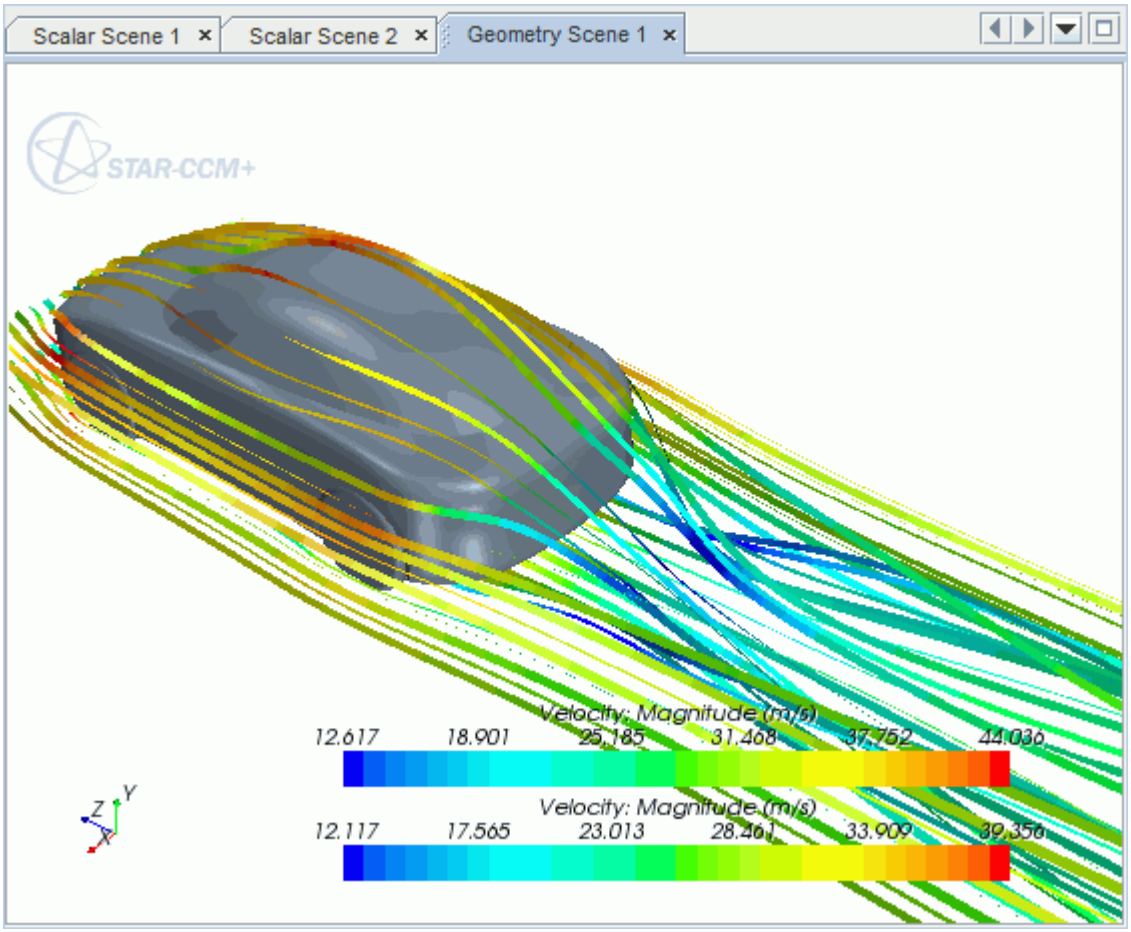
The finished scene is shown below.


Scalar Scene 1
Scalar Scene 2
Geometry Scene 1
STAR-CCM+
Velocity: Magnitude (m/s)
0.039573
3.0888
6.1381
9.1874
12.237
15.286
X
Y
Z

External Flow Over a Car

This tutorial takes you through the steps that are required to run a CFD calculation in the STAR-Creo environment.

For this tutorial, the model is a car in a fluid domain.



Contents:

Opening the CAD Model
Selecting Fluid Physics Models
Defining Boundary Conditions
Setting the Mesh Pipeline
Creating the Volumetric Mesh Controls
Creating the Face Mesh Control
Modifying Solution Controls
Reviewing the Simulation Set Up
Generating the Surface Mesh

Generating the Volume Mesh
Building and Running
Post-Processing Results

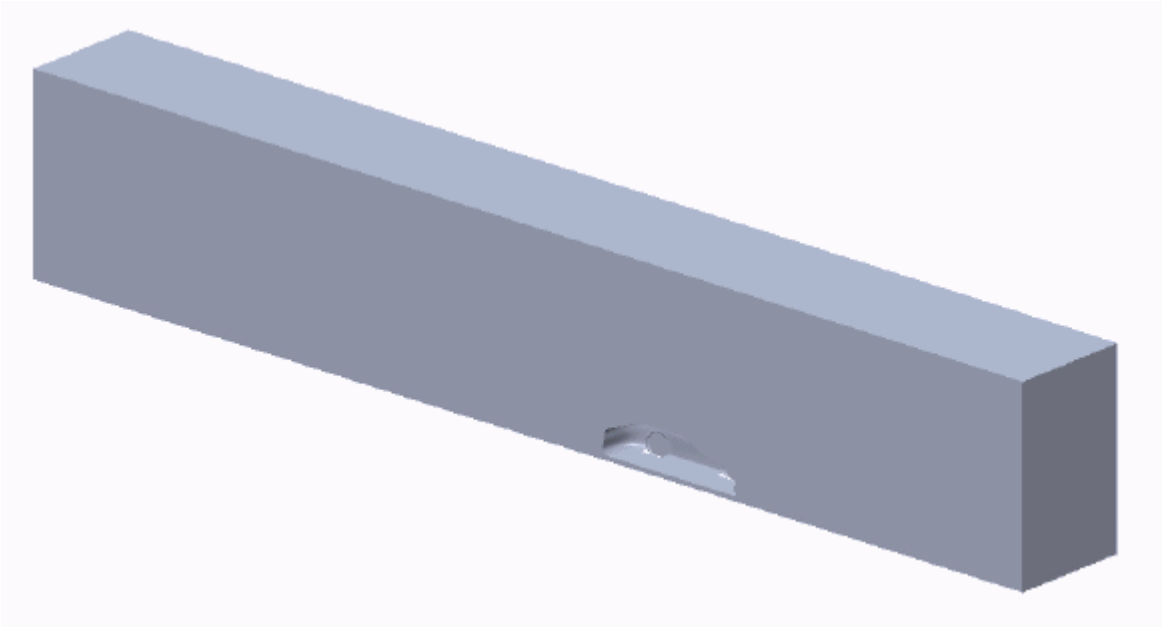
Opening the CAD Model

A directory containing the CAD geometry is provided with the STAR-CCM+ distribution.

This directory is located in `[STAR-CCM+_INSTALL_DIR]\doc\startutorialsdata\starcreo\data`.

1. Copy directory `data` to your own disk space and open the part file `car.prt` in STAR-Creo.

   The imported geometry is shown below.



The STAR-Creo ribbon appears at the top of the screen.The STAR-Creo object tree is located on the left of the screen.

To switch to this object tree:

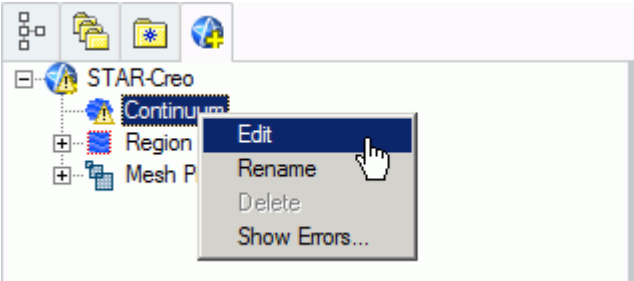
2. Click (**STAR Tree**).

If these items are not visible, refer to Launching STAR-Creo Manually.

Selecting Fluid Physics Models

The first step in setting up this tutorial is to define the physics of the problem. For this case, the flow is turbulent and isothermal.

1. Right-click the **Continuum** node in the STAR-Creo tree and select **Edit**.



The *Continuum Settings* dialog appears.

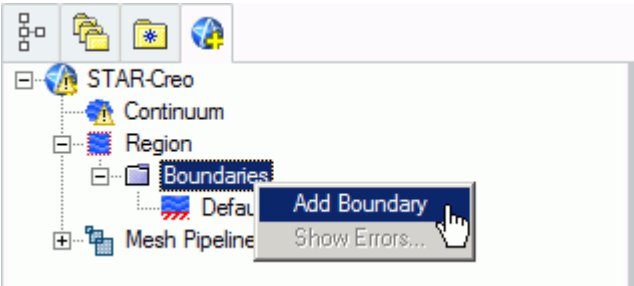
2. Deactivate the **Temperature** option in the *Optional Physics Models*.
3. Click **OK**.

Defining Boundary Conditions

Create the inlet and outlet boundary condition. Use a moving wall to model the relative movement of the road.

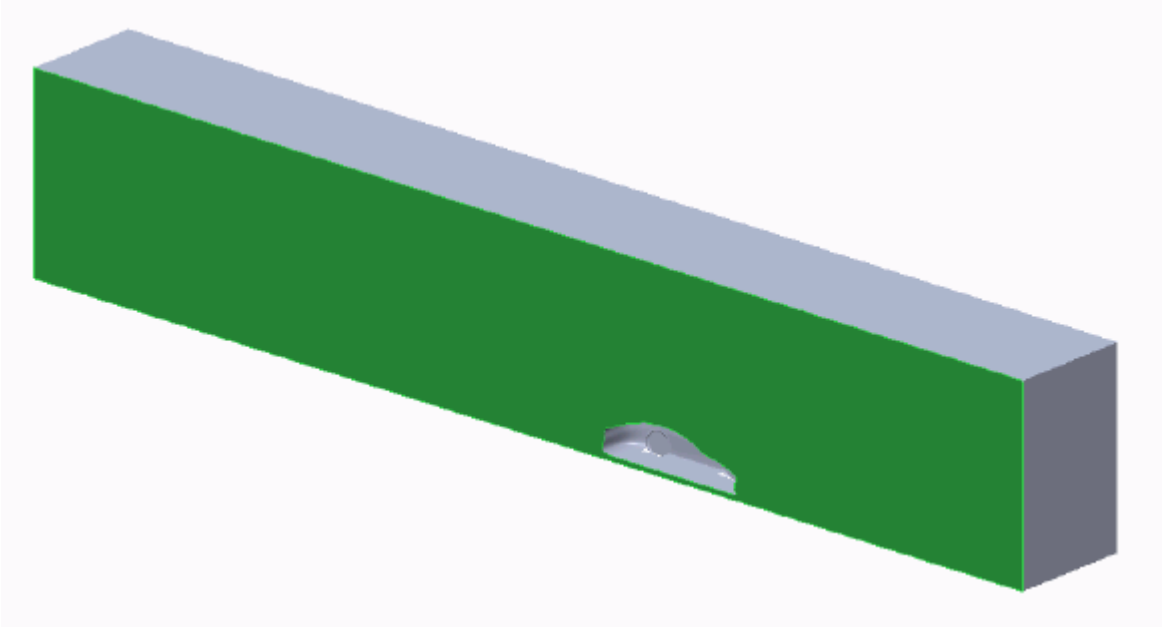
Create the symmetry plane first:

1. Right-click the **Region** > **Boundaries** node and select **Add Boundary**.



The *Boundary Settings* dialog appears.

2. Click (**Select Face**) to activate the Creo face selection mechanism.
3. On the CAD model, select the face that is highlighted below.



These selections are reported in the Included *Surfaces* box in the *Surface Sets* dialog.

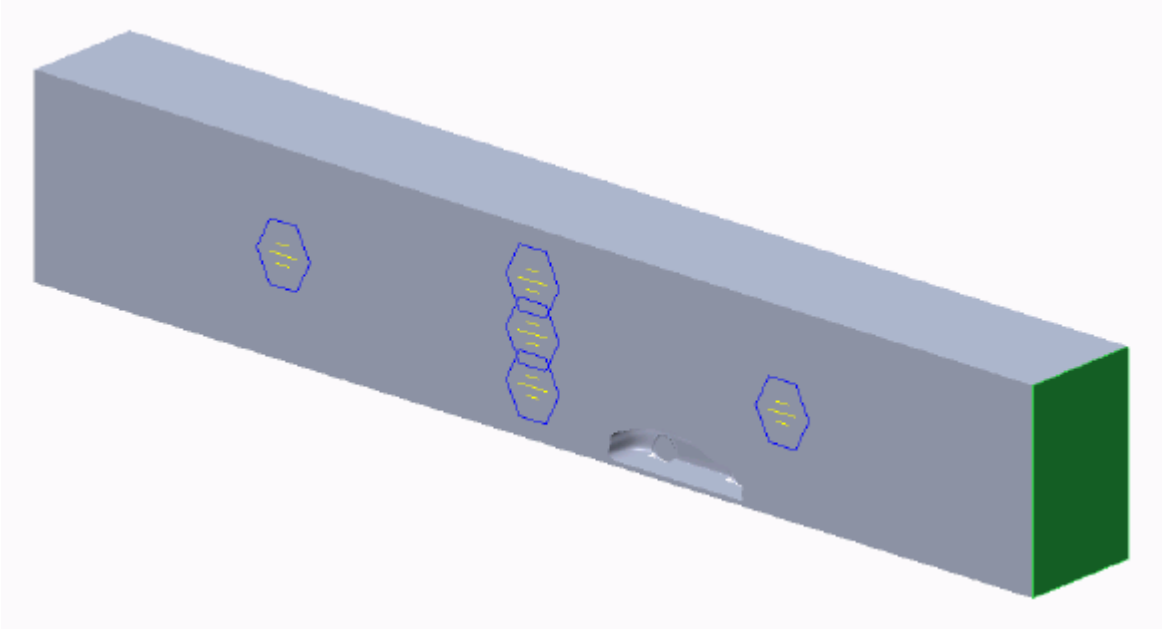
4. Click **OK** to close the *Surface Sets* dialog.
5. Set the following parameters:

Parameter	Value
Type	**Symmetry Plane**
Name	`Symmetry Plane`

6. Click **OK**.

Create the inlet boundary:

7. Create another boundary and select the following face.



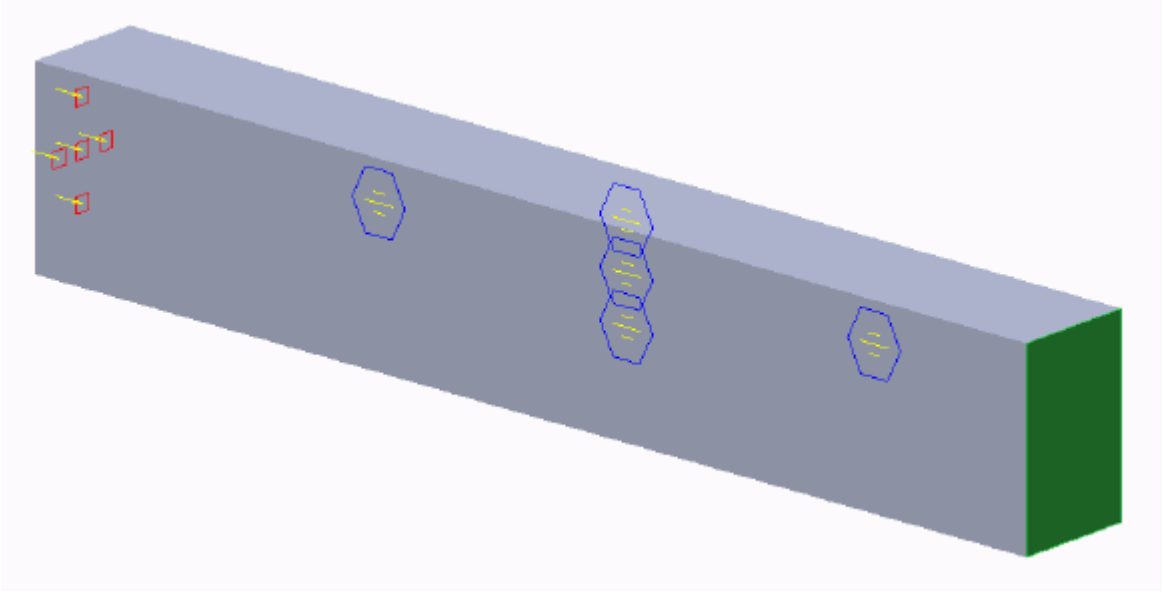
8. Set the following parameters:

Parameter	Value
Type	**Velocity Inlet**
Name	`Inlet`
(*Normal Velocity*)	`70 mph` **Note:** The units are in **mph**.

9. Click **OK**.

Create the outlet boundary:

10. Create another boundary.
11. Rotate the view to display the back face and select the face that is highlighted below.



12. Set the following parameters:

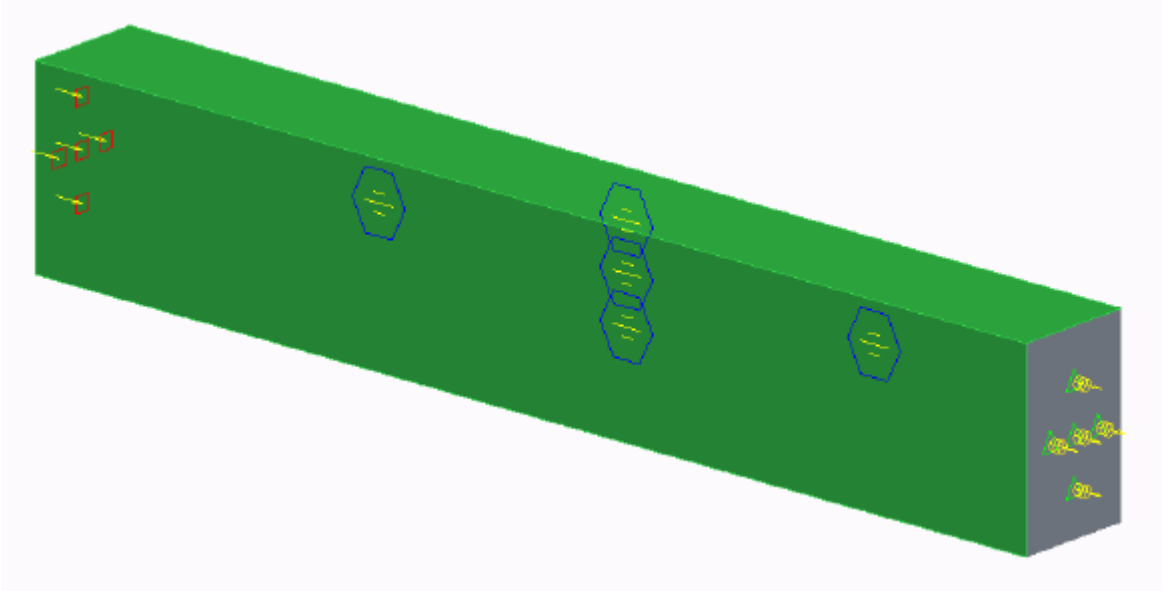
Parameter	Value
Type	**Pressure Outlet**
Name	`Outlet`

13. Click **OK**.

Create the fluid domain symmetry boundaries:

14. Create another boundary.
15. Rotate the view to display the top and side walls, and select the faces that are highlighted below.

    To select multiple faces, hold down the <`Ctrl`>.



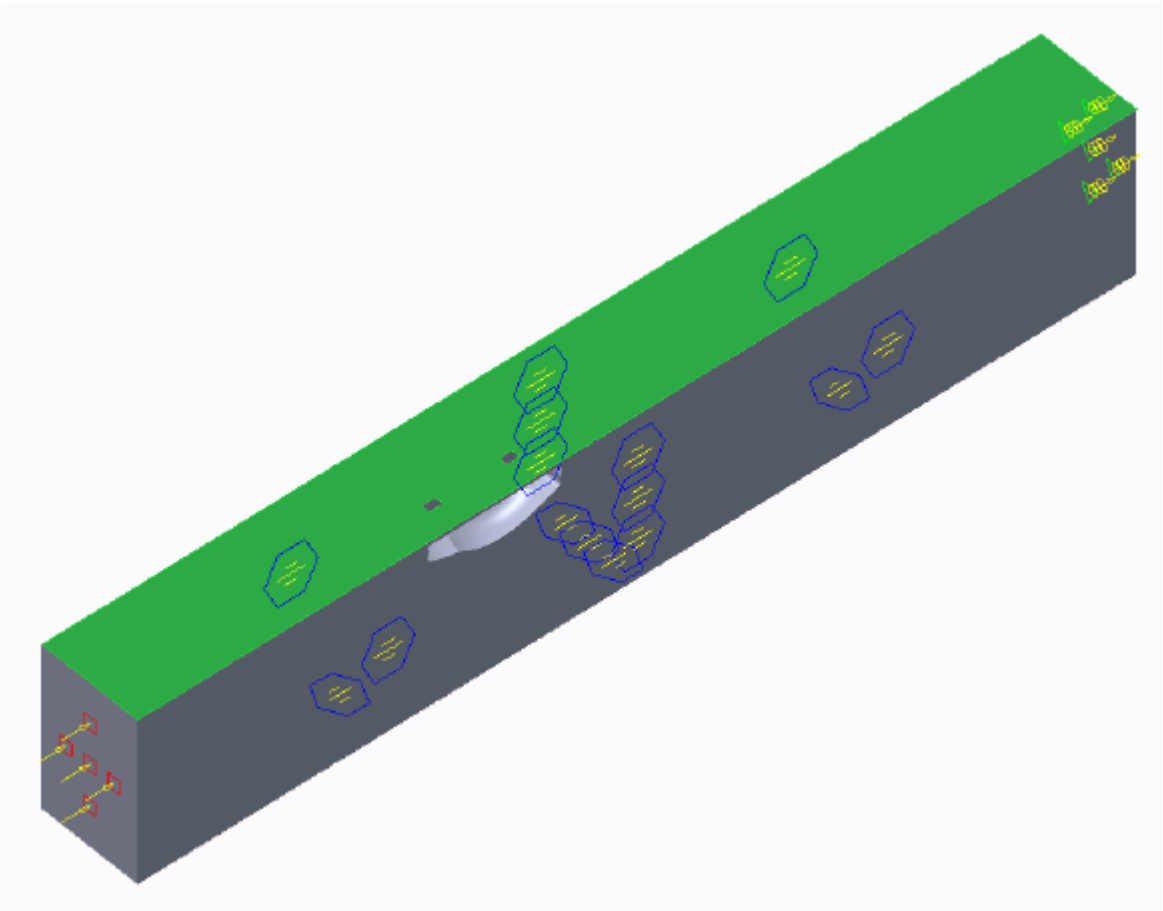
16. Set the following parameters:

Parameter	Value
Type	**Symmetry Plane**
Name	`Boundary Symmetry`

17. Click **OK**.

Create the boundary that will model the moving ground:

18. Create another boundary.
19. Rotate the view to display the bottom face, and select the face that is highlighted below.



20. Set the following parameters:

Parameter	Value
Type	**Wall**
Name	`Ground`
Wall Type	**Translating**
(*W Component of Wall Velocity Vector*)	`-70 mph`

Parameter	Value
	Note: The units are in **mph**.

21. Click **OK**.

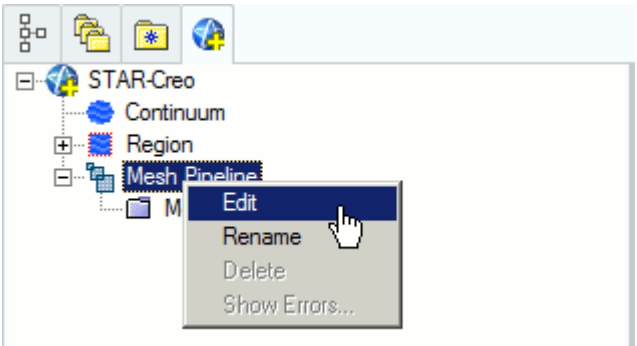
Rename the default boundary:

22. Right-click **Region** > **Boundaries** > **Default** and select **Rename**.
23. Enter `Car` as the new name.
24. You may need to click on a blank area of the workspace to accept the name change.
25. Save your progress.

Setting the Mesh Pipeline

To obtain more accurate results, modify the mesh pipeline:

1. Right-click the **Mesh Pipeline** node in the STAR-Creo tree and select **Edit**.



The *Mesh Pipeline Settings* dialog appears.

2. Set the following parameters:

Parameter	Value
Base Size	
(**Base size option**)	**Absolute**
(*Base Size*)	`0.03 m`
Prism Layer	
(**Layer thickness option**)	**Absolute**
(*Prism Layer Thickness*)	`0.03 m`

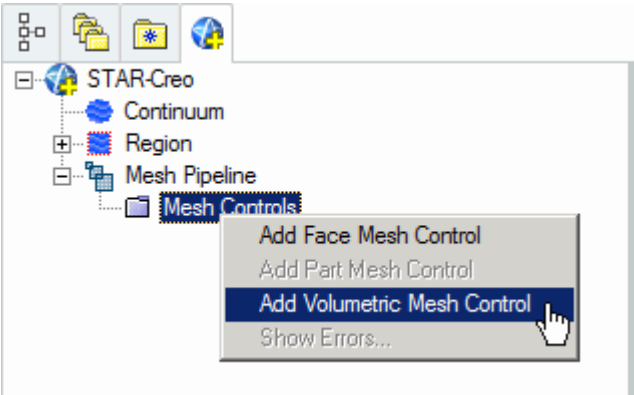
3. Click **OK**.

Creating the Volumetric Mesh Controls

Create two volumetric mesh control regions to capture the finer details of the flow: one around the car and one that extends into the wake of the car.

Create the volumetric mesh control for the car region:

1. Right-click the **Mesh Pipeline** > **Mesh Controls** node in the STAR-Creo tree and select **Add Volumetric Mesh Control**.



   The *Volumetric Control* dialog appears.
2. Set *Name* to `Car Box`.
3. Activate the *Absolute* option.
4. Set (*Mesh Size*) to `0.09 m`.
5. Enter the following coordinates for Point 1:

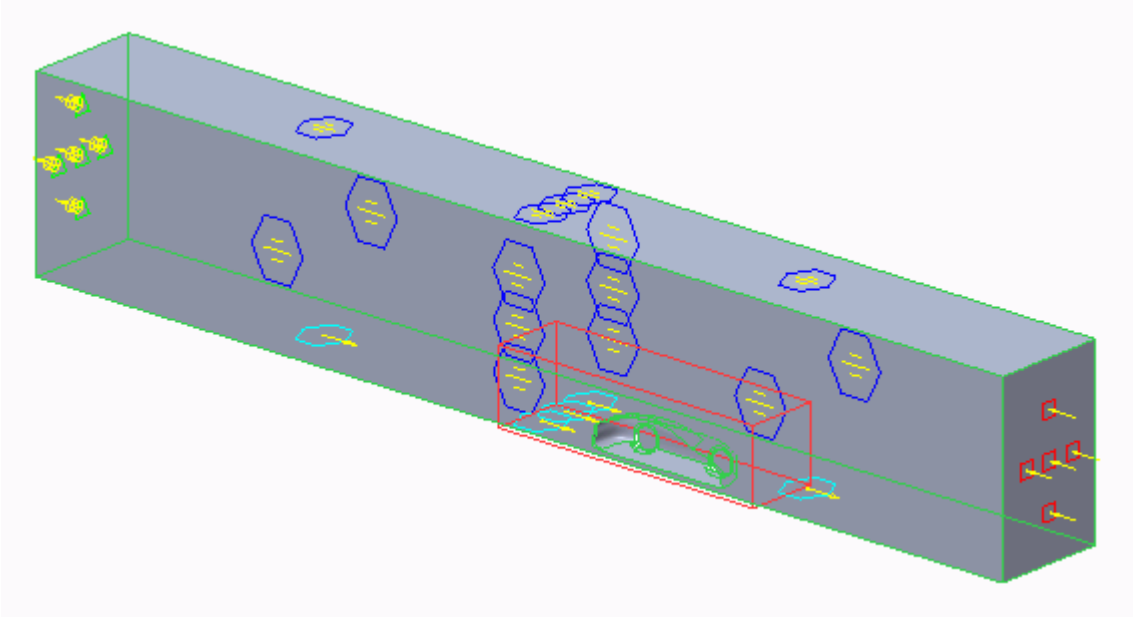
Coordinate	Value
X1	`-50.00`
Y1	`-650.00`
Z1	`3432.00`

6. Enter the following coordinates for Point 2:

Coordinate	Value
X2	`2000.00`
Y2	`1440.00`
Z2	`-4567.00`

7. Click **Create**.

   The volumetric control region is shown below.



8. Click **OK**.

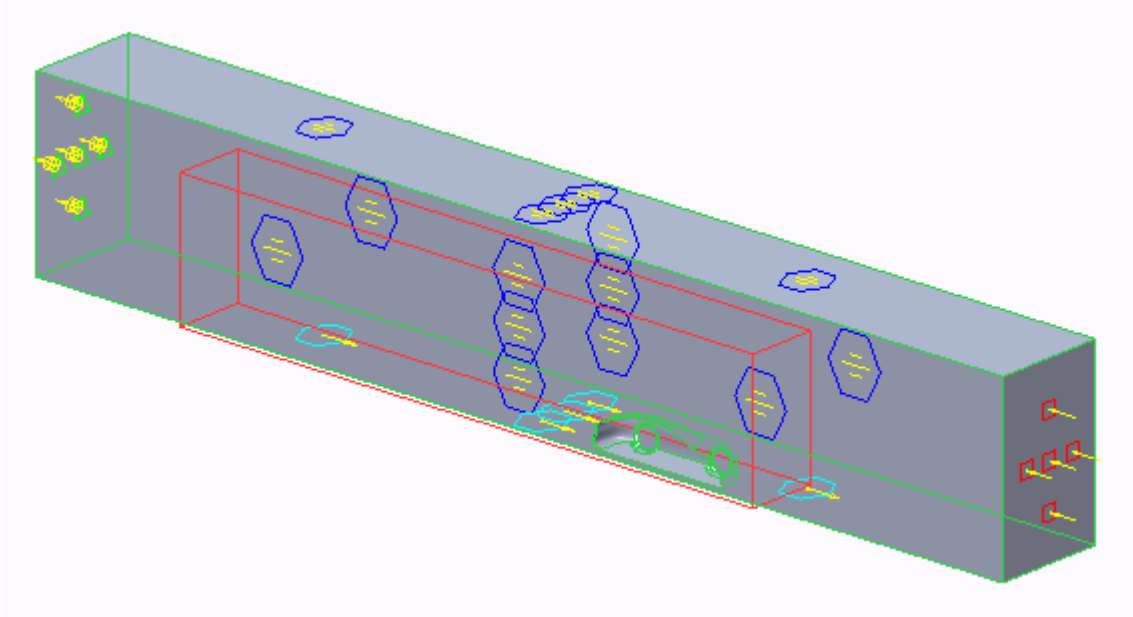
Create the volumetric mesh control for the wake region:

9. Using the same technique, create another volumetric mesh control with the following settings:

Parameter	Value
Name	`Wake Box`
Mesh Size	
(**Base size option**)	**Absolute**
(*Mesh Size*)	`0.177 m`
Shape	
X1	`-50.00`
Y1	`-650.00`
Z1	`3432.00`
X2	`2000.00`
Y2	`3240.00`
Z2	`-14567.00`

10. Click **Create**.

    The volumetric control region is shown below.



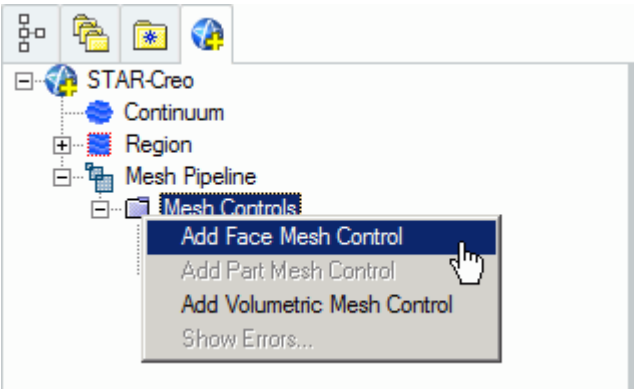
11. Click **OK**.

Creating the Face Mesh Control

Apply a face mesh control to the walls of the fluid domain to coarsen the mesh on these faces.

Create the face mesh control:

1. Right-click the **Mesh Pipeline** > **Mesh Controls** node in the STAR-Creo tree and select **Add Face Mesh Control**.



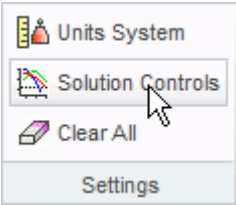
   The *Face Mesh Control Settings* dialog appears.
2. Select all six boundary faces that form the fluid domain: `Inlet`, `Outlet`, `Boundary Symmetry`, `Symmetry Plane`, and `Ground`.
3. Set *Name* to `Fluid Domain Face Control`.
4. Activate the **Enable** option in the *Custom Mesh Size* box.
5. Activate the *Absolute* option.

6. Set (*Mesh Size*) to `0.35 m`.
7. Click **OK**.

Modifying Solution Controls

Modify the solution controls:

1. Click **Solution Controls** on the STAR-Creo ribbon.



   The *Solution Controls* dialog appears.

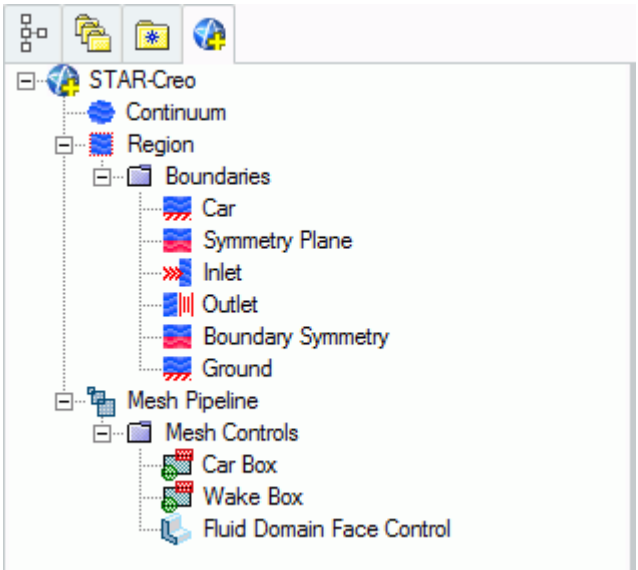
2. Set the following parameters:

Parameter	Value
(*Number of Iterations*)	`400`
(*Convergence Level*)	`0.0001`
(*Momentum Under-Relaxation Factor*)	`0.85`

3. Click **OK**.
4. Save your progress.

Reviewing the Simulation Set Up

The final STAR-Works object tree is shown below.

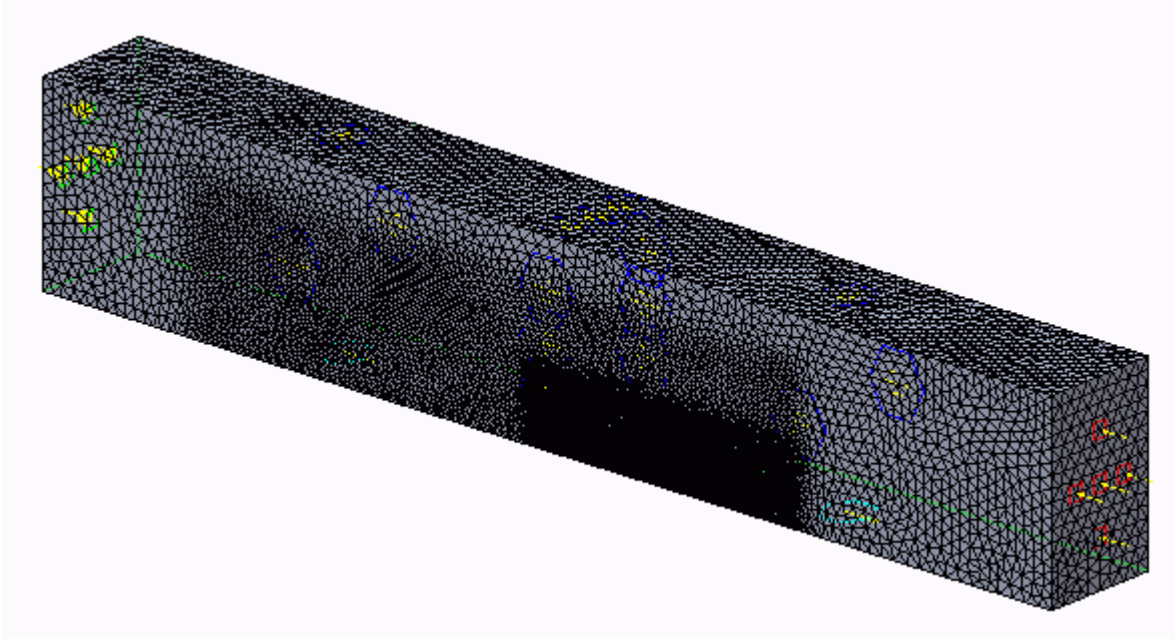


Generating the Surface Mesh

Generate the surface mesh:

1. From the STAR-Creo ribbon, click **Surface Mesh**.

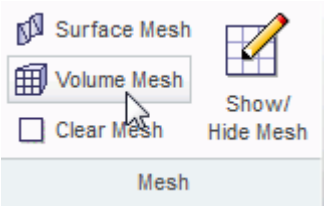
The surface mesh is shown below.



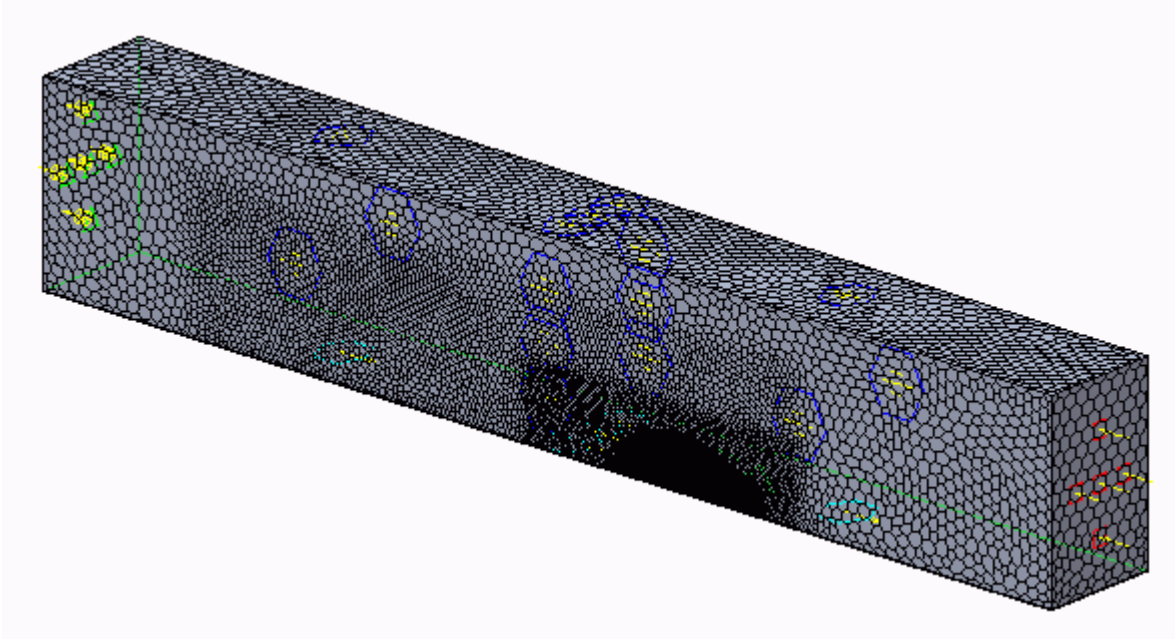
Generating the Volume Mesh

Generate the volume mesh:

1. From the STAR-Creo ribbon, click **Volume Mesh**.



The volume mesh is shown below.

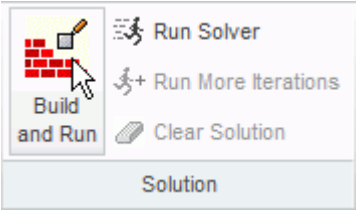


Building and Running

The case is now ready to run.

To run the simulation:

1. From the ribbon, click **Build and Run**.



The *Residuals* plot appears in the STAR-Creo *Output* window. You can use this plot to monitor the convergence of the solution.

When the solver is done, STAR-CCM+ launches.

Post-Processing Results

You perform all post-processing in STAR-CCM+. By default, a scalar scene, *Scalar Scene 1*, automatically appears in the *Graphics* window when the solver finishes. This scene displays the absolute total pressure.

The results that are contained in the following sections are representative. You may see some variation in the results that you obtain, depending on what platform and package you are using.

1. In STAR-CCM+, click **File** > **Save** to save the simulation.

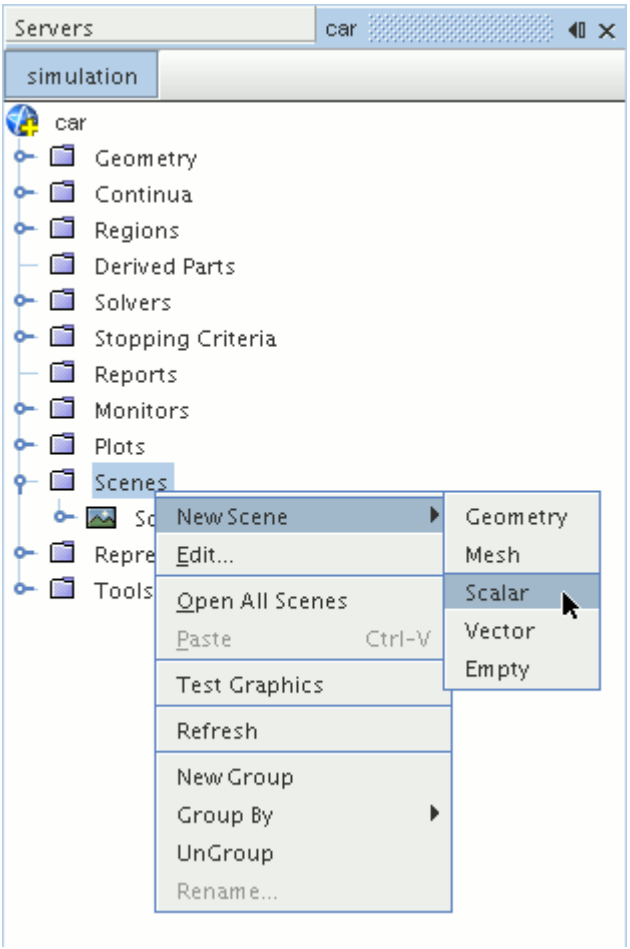
Contents:

Creating a Scalar Scene
Creating A Streamlines Scene
Creating a Vector Scene
Creating a Streamlines Scene
Creating A Scalar Scene
Visualizing the Wing Skin Temperature
Creating a Streamlines Scene
Creating A Scalar Scene
Creating a Streamlines Scene
Creating a Scalar Scene
Modifying The Scalar Scene
Creating a Streamlines Scene

Creating a Scalar Scene

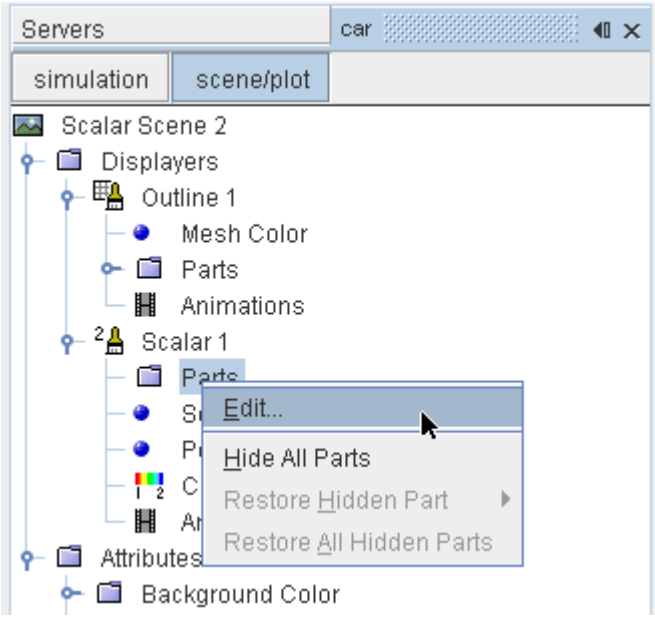
Create a scalar scene to display static pressure.

1. Right-click the **Scenes** node in the explorer pane and select **New Scene** > **Scalar**.



A new scalar scene, *Scalar Scene 2*, is created and selected in the *Graphics* window.

2. Click **scene/plot**.
3. Right-click the **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Parts** node and select **Edit...**.

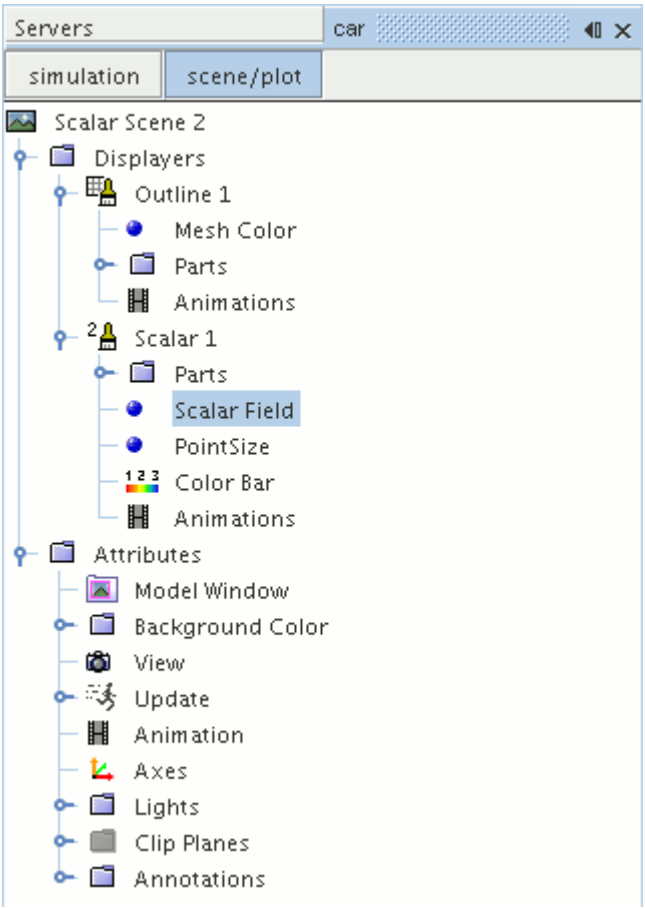


The *Parts* dialog appears.

4. Expand the **Regions** > **Region** > **Boundaries** node and select the following items:
   - **Car**
   - **Ground**
   - **Symmetry Plane**
5. Click **OK**.

Display the pressure contours in the scene:

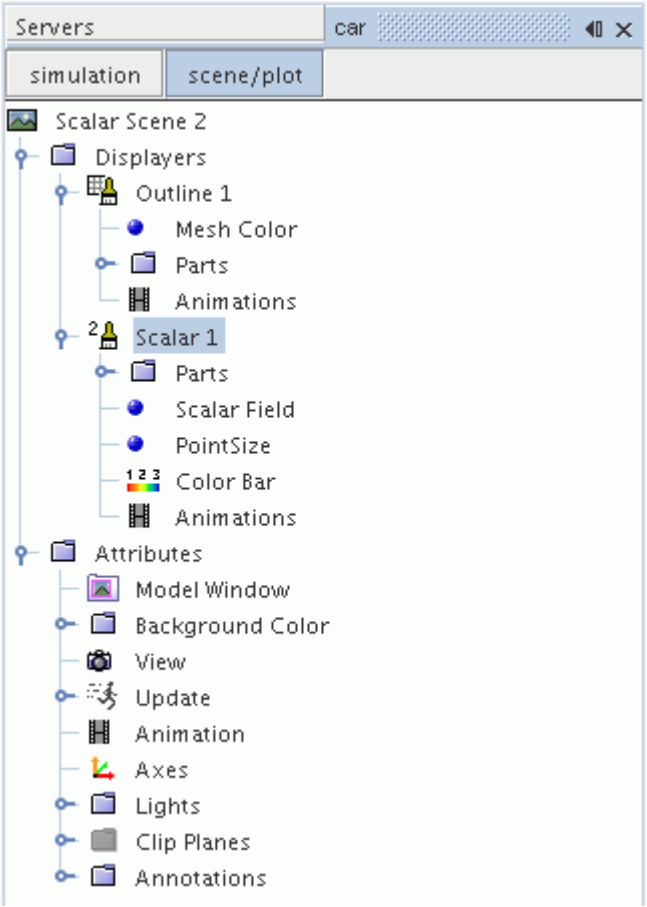
6. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Scalar Field**.



7. Set *Function* to **Static Pressure**.

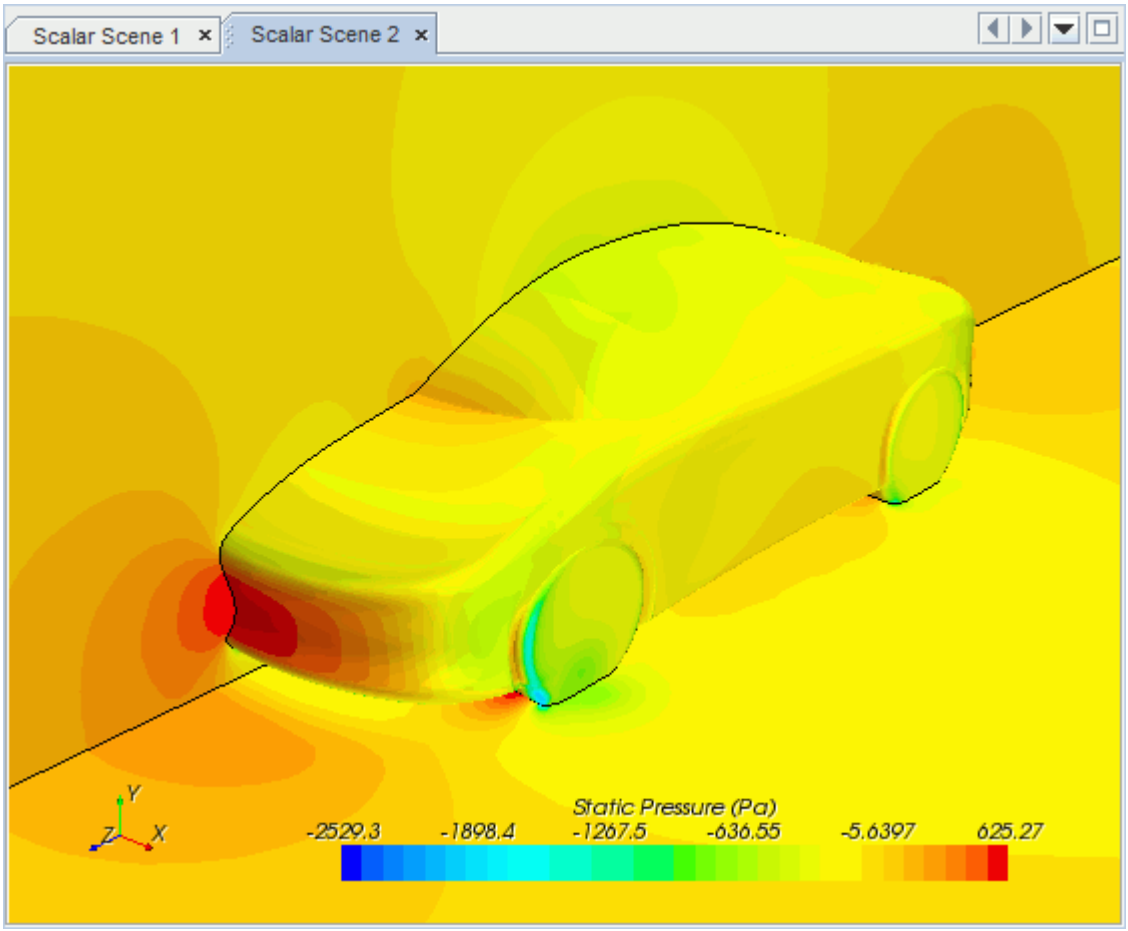
To have a smooth transition between the different pressure values:

8. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1**.



9. Set *Contour Style* to **Smooth Filled**.
10. Rotate the scene to get a better view of the front of the car.
11. Click (**Save**) to save the simulation.

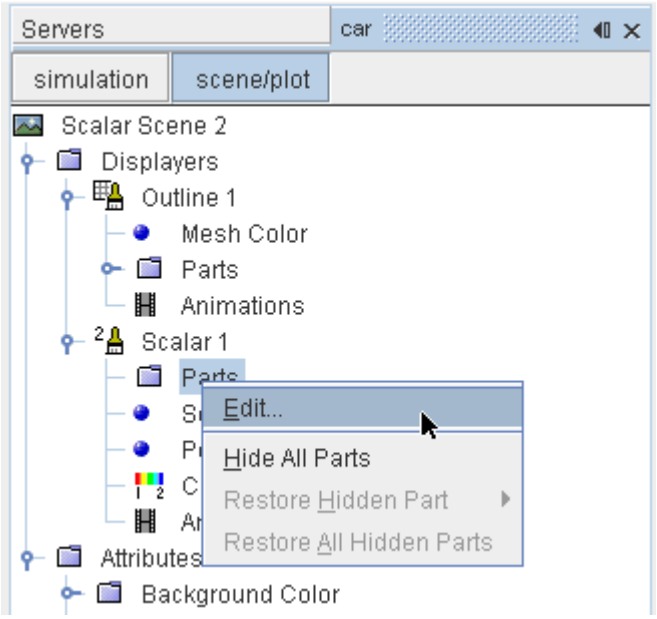
The finished scene is shown below.



Modifying The Scalar Scene

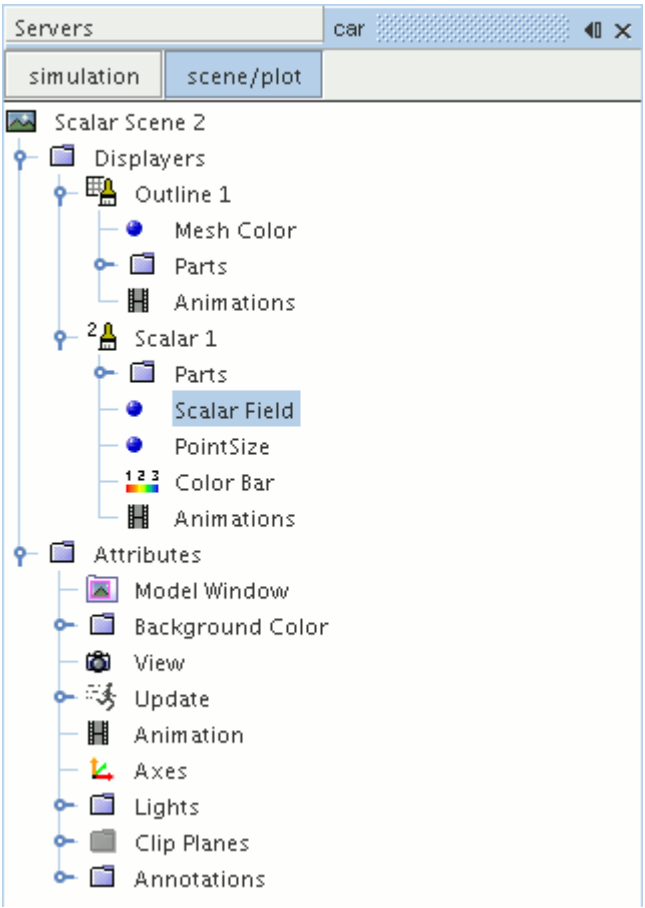
Modify Scalar Scene 2 to display Turbulent Kinetic Energy and remove the symmetry plane from the scene. You also mirror the car body:

1. Right-click the **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Parts** node and select **Edit...**.



The *Parts* dialog appears.

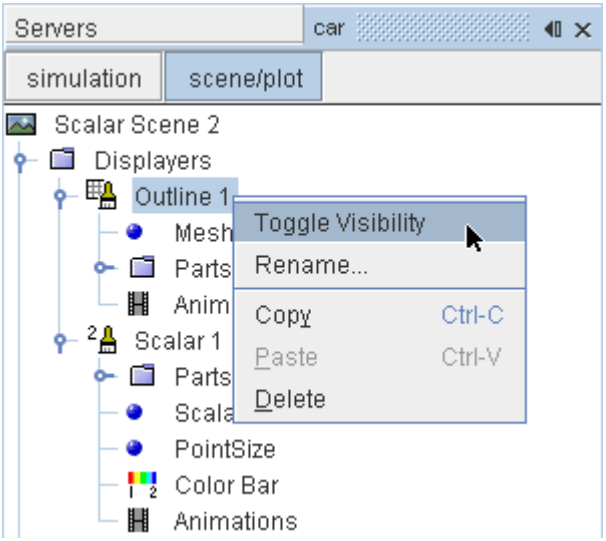
2. Expand the **Regions** > **Region** > **Boundaries** folder and deactivate **Symmetry Plane.**
3. Click **OK**.
4. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Scalar Field**.



5. Set *Function* to **Turbulent Kinetic Energy**.

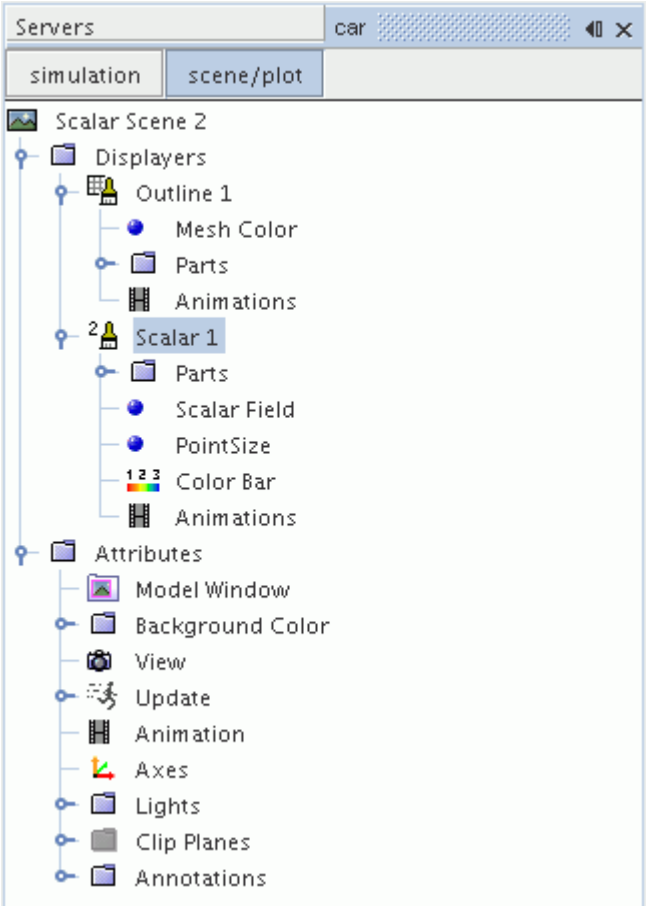
Hide the outlines in the scene:

6. Right-click the **Scalar Scene 2** > **Displayers** > **Outline 1** node and select **Toggle Visibility**.

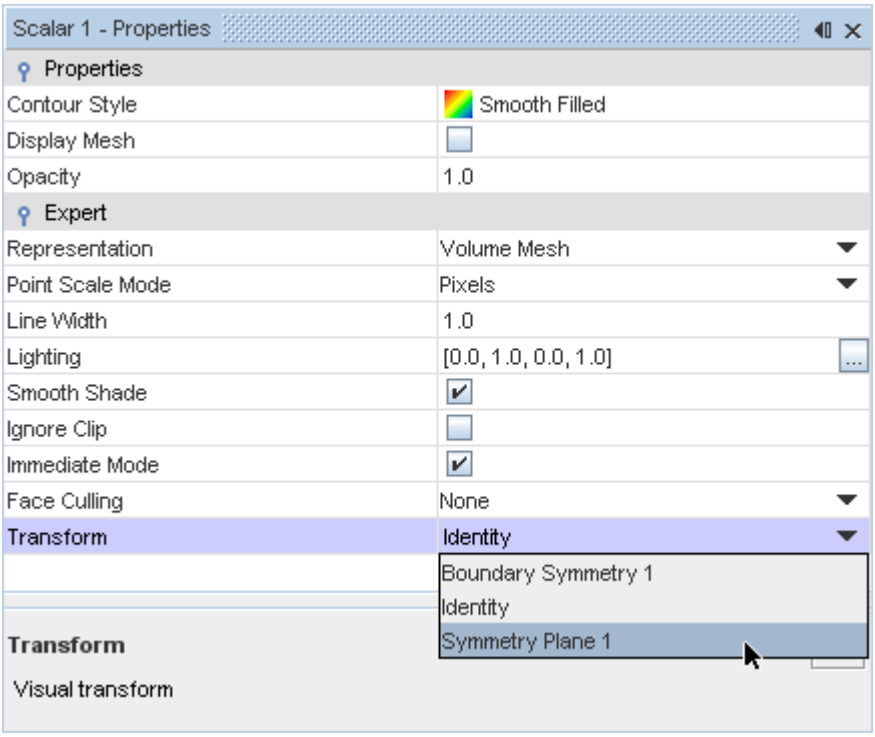


Mirror the car body:

7. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1**.

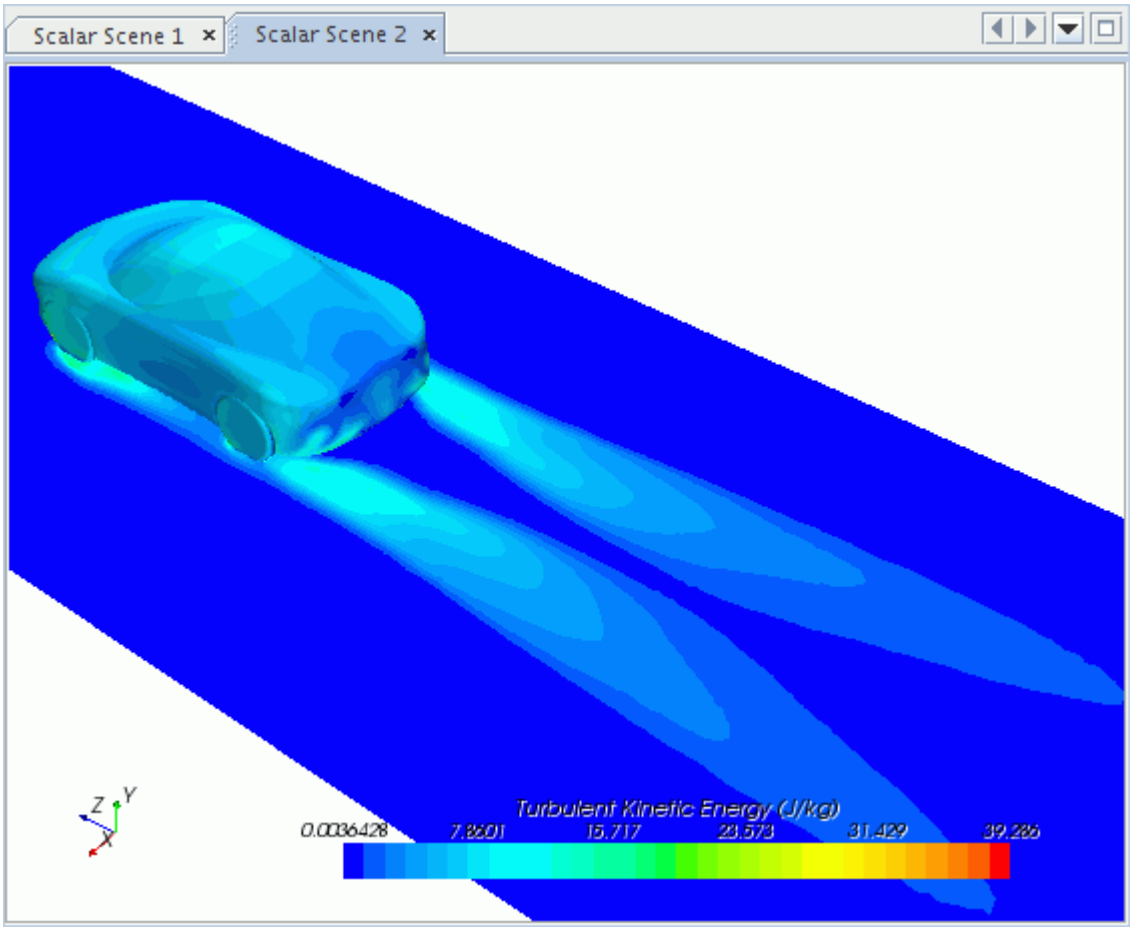


8. Set *Transform* to **Symmetry Plane 1**.



9. Rotate the scene to get a better view of the wake of the car.
10. Click **simulation**.
11. Save the simulation.

The finished scene is shown below.

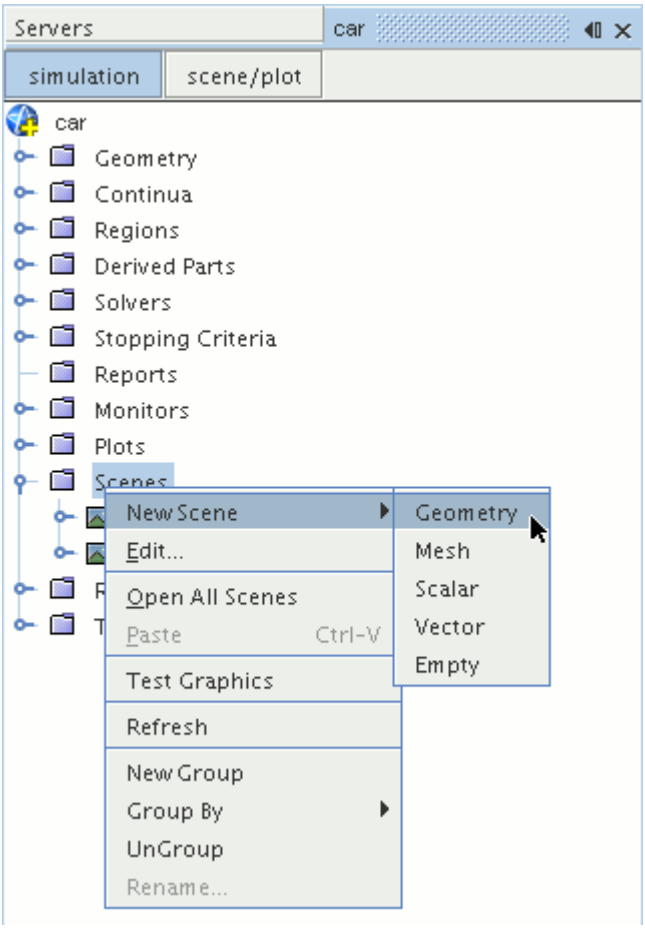


Creating a Streamlines Scene

Create streamlines to get a better understanding of the flow around the car and in the wake area.

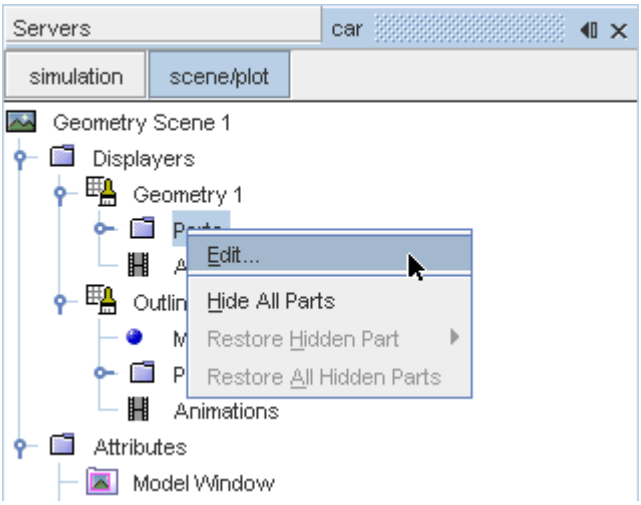
First, create a geometry scene:

1. Right-click the **Scenes** node in the object tree and select **New Scene** > **Geometry**.



A new geometry scene is created and selected in the *Graphics* window.

2. Click **scene/plot**.
3. Right-click the **Geometry Scene 1** > **Displayers** > **Geometry 1** > **Parts** node and select **Edit...**.



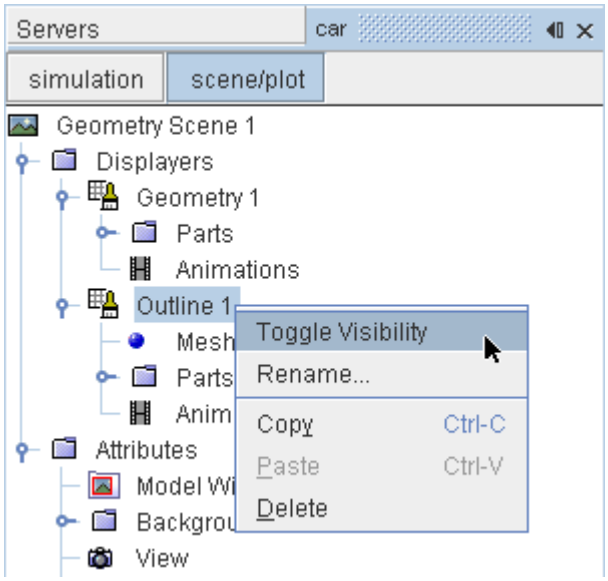
The *Parts* dialog appears.

4. Right-click on a blank area and choose **Deselect All**.

5. Expand the **Regions** > **Region** > **Boundaries** node and select **Car**.
6. Click **OK**.

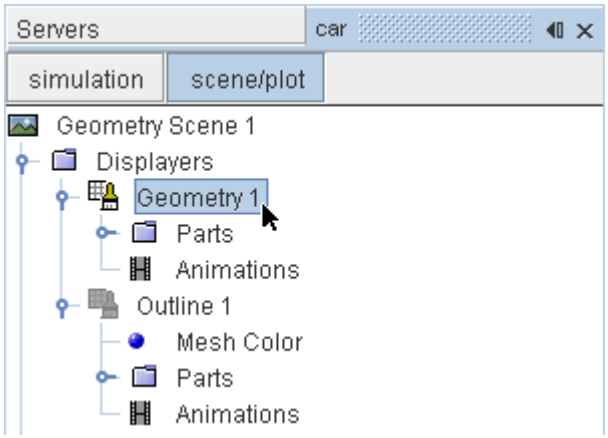
Hide the outlines in the scene:

7. Right-click the **Geometry Scene 1** > **Displayers** > **Outline 1** node and select **Toggle Visibility**.

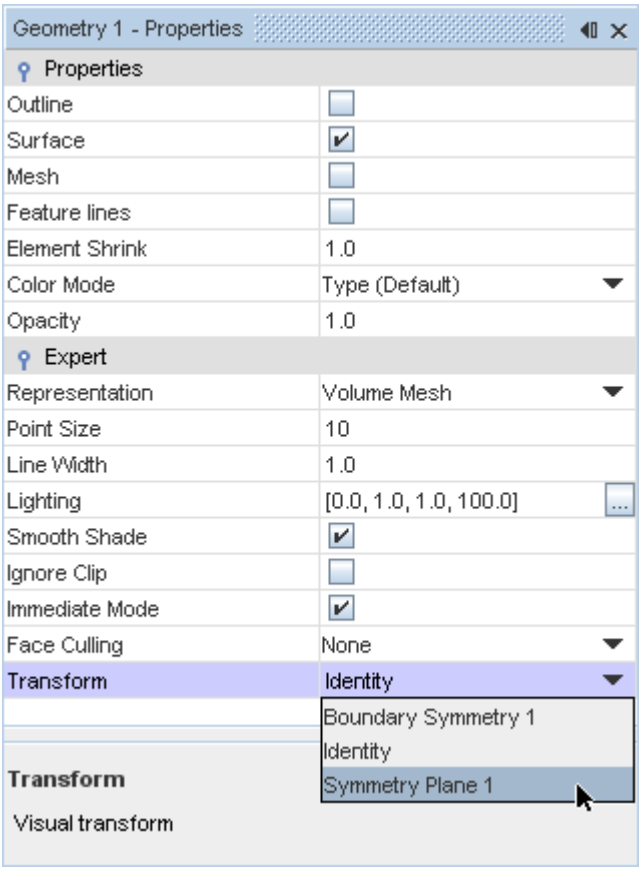


Mirror the car body:

8. Navigate to **Geometry Scene 1** > **Displayers** > **Geometry 1**.



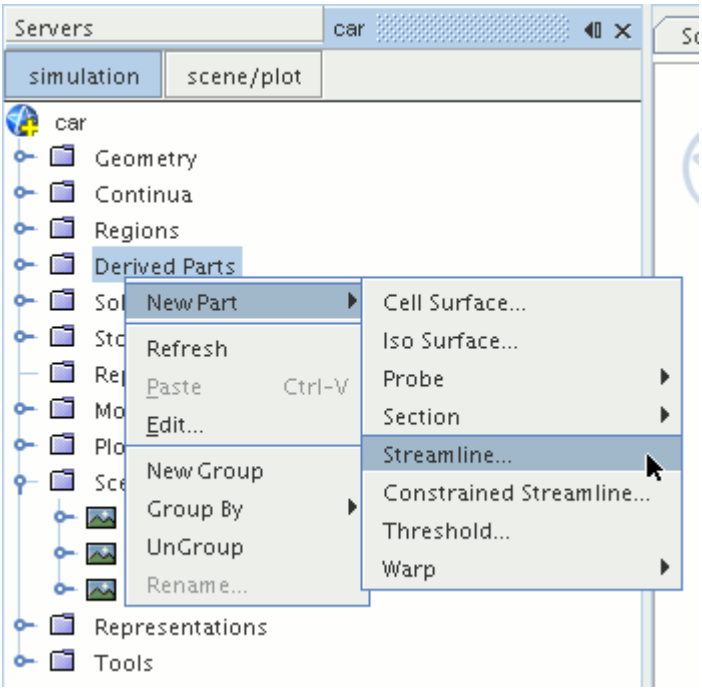
9. Set *Transform* to **Symmetry Plane 1**.



10. Click **simulation**.

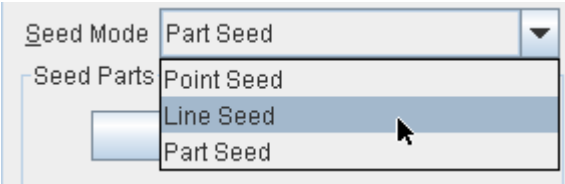
Create the streamlines:

11. Right-click the **Derived Parts** node and select **New Part** > **Streamline...**.



The *Create Streamline* dialog appears.

12. Set *Seed Mode* to **Line Seed**.



13. Enter these coordinates for the line:
    - *Point 1*:
        - X: `0 mm`
        - Y: `-123 mm`
        - Z: `2600 mm`
    - *Point 2*:
        - X: `1200 mm`
        - Y: `-123 mm`
        - Z: `2600 mm`
14. Set *Resolution* to `10`.
15. Make sure that **New Streamline Displayer** is selected in the *Display* section.
16. Click **Create**.

Create another set of streamlines using the existing dialog:

17. Enter these coordinates for the line:
    - *Point 1*:
        - X: `0 mm`
        - Y: `-473 mm`
        - Z: `2600 mm`
    - *Point 2*:
        - X: `1200 mm`
        - Y: `-473 mm`
        - Z: `2600 mm`
18. Click **Create** followed by **Close**.

Modify the appearance of the streamlines to display ribbons instead of the default lines.

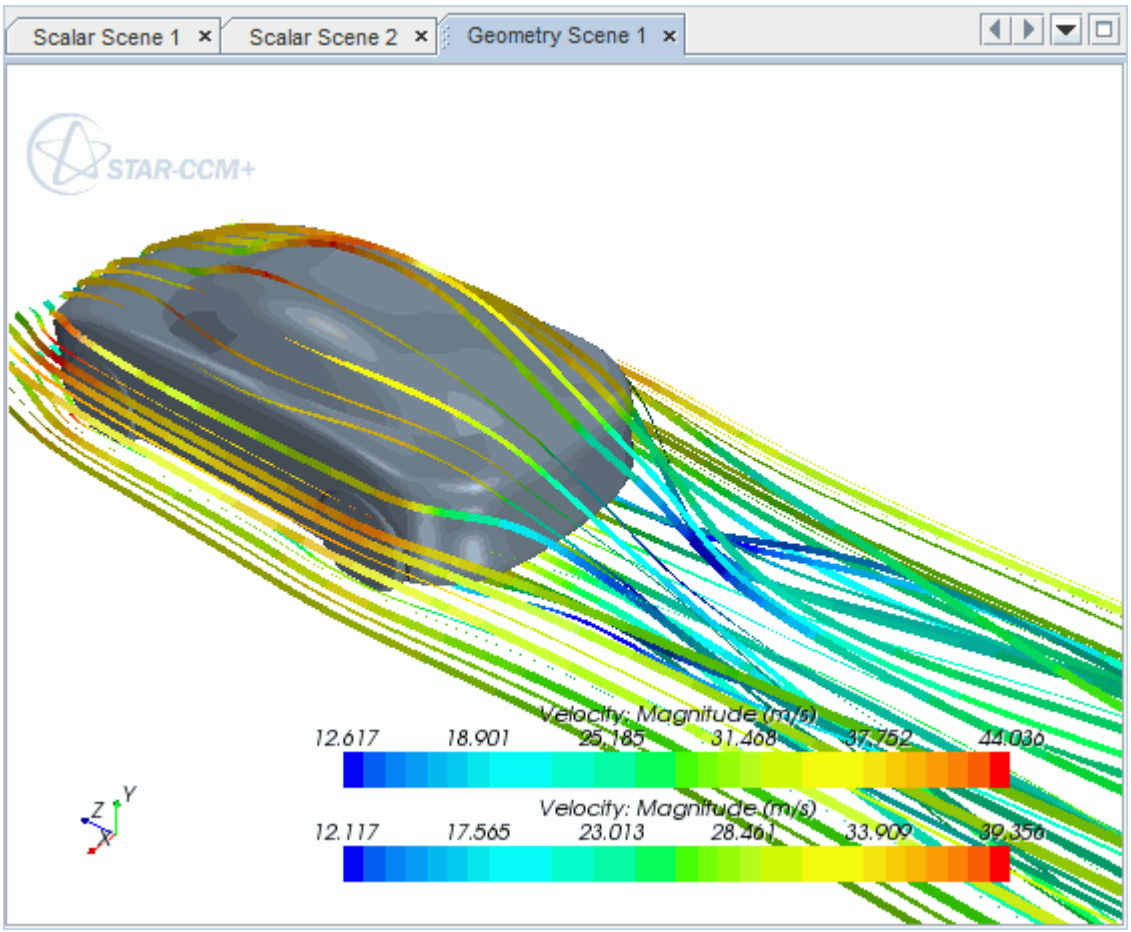
19. Click **scene/plot**.
20. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1**.
21. In the *Properties* window:
    - Set *Width* to `0.025`.
    - Set *Mode* to **Ribbons**.

- Set *Transform* to **Symmetry Plane 1**.

Modify the scalar field of the streamlines to display velocity magnitude:

22. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** > **Scalar Field** and set *Function* to **Velocity** > **Lab Reference Frame** > **Magnitude**.
23. Repeat the above steps for **Streamline Stream 2**.
24. Rotate the scene to get a better view of the wake of the car.
25. Save the simulation.

The finished scene is shown below.



Geometry Transfer: CAD to Simulation

STAR-Creo allows you to transfer geometry from Creo to STAR-CCM+ through a direct link between the two packages.

This feature provides you with a workflow that is tailored to design optimization. You can apply changes to the geometry in either STAR-Creo or STAR-CCM+ and the changes are passed back to the other program. For example, you can make changes to the design after running an analysis in the STAR-CCM+ environment and these changes are passed back to the original Creo file. This feature requires a CAD client license.

This tutorial demonstrates how to select the design parameters in your original geometry and transfer the geometry from Creo to STAR-CCM+. In STAR-CCM+, you proceed to create a repeatable workflow (or pipeline) using mesh operations. After setting up this pipeline, you can make further changes to the geometry design and have STAR-CCM+ regenerate the corresponding volume mesh.

Contents:

Opening the CAD Model
Transferring the Geometry to STAR-CCM+
Creating the Shape Parts and Operations
Setting Up the Mesh Pipeline
Modifying the Design

Opening the CAD Model

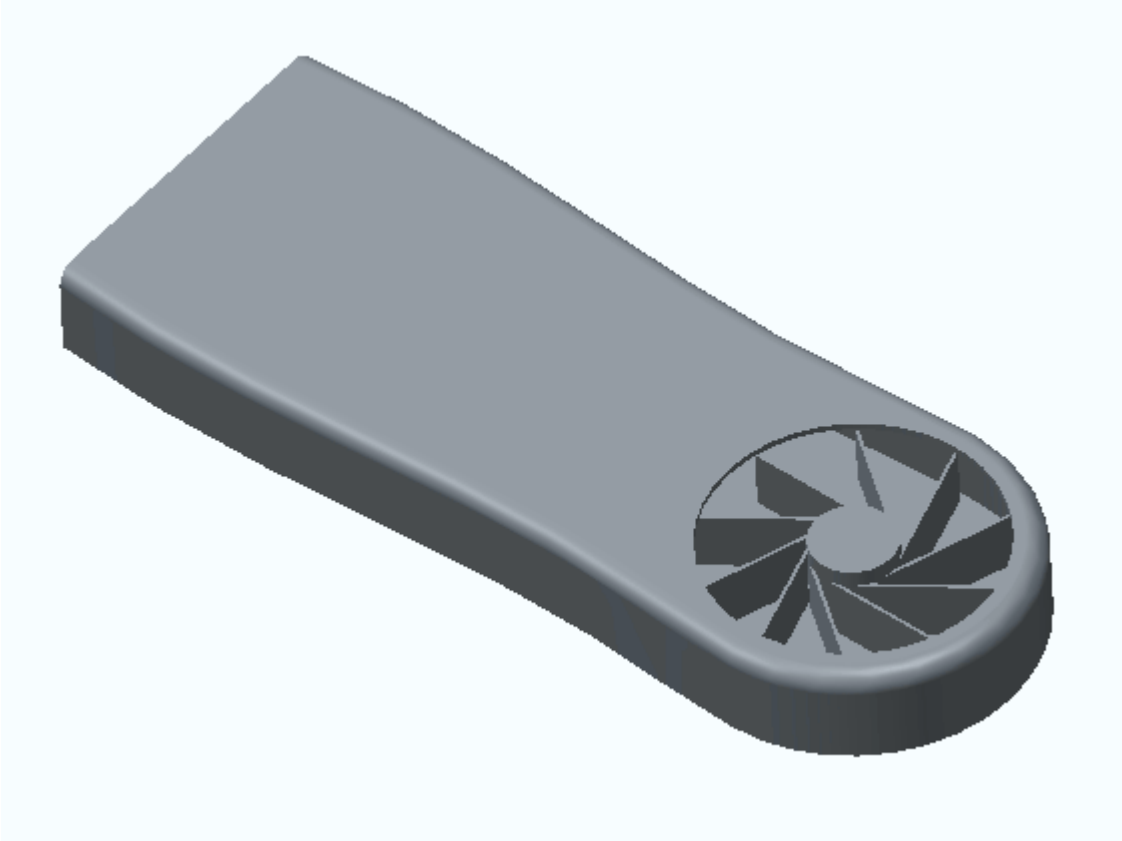
The STAR-CCM+ installation includes a directory containing the CAD geometry for this tutorial.

This directory is located in `[STAR-CCM+_INSTALL_DIR]\doc\startutorialsdata\starcreo\data`.

1. Copy directory `data` to your own disk space.
2. Launch STAR-Creo and open the assembly file `cardassem.asm` in STAR-Creo.

   The assembly is made up of two parts: the card housing and the blower fan.

   The imported geometry is shown below.



The STAR-Creo ribbon appears at the top of the screen. The STAR-Creo object tree is located on the left of the screen.

To switch to the object tree:

3. Click (**STAR Tree**).

If these items are not visible, refer to Launching STAR-Creo Manually.

Transferring the Geometry to STAR-CCM+

In STAR-Creo, choose the design parameters to expose to STAR-CCM+ and then transfer the geometry.

Specify the design parameters to transfer to STAR-CCM+:

1. From the STAR-Creo ribbon, click **Design Data**.

The *Design Data* dialog appears.

2. In the *Design Parameters* tab, select **blowerfan.prt** > **p14 = 10**.

   STAR-Creo only recognizes native CAD features as design parameters.

   The value of this parameter is displayed in the *Value* property.

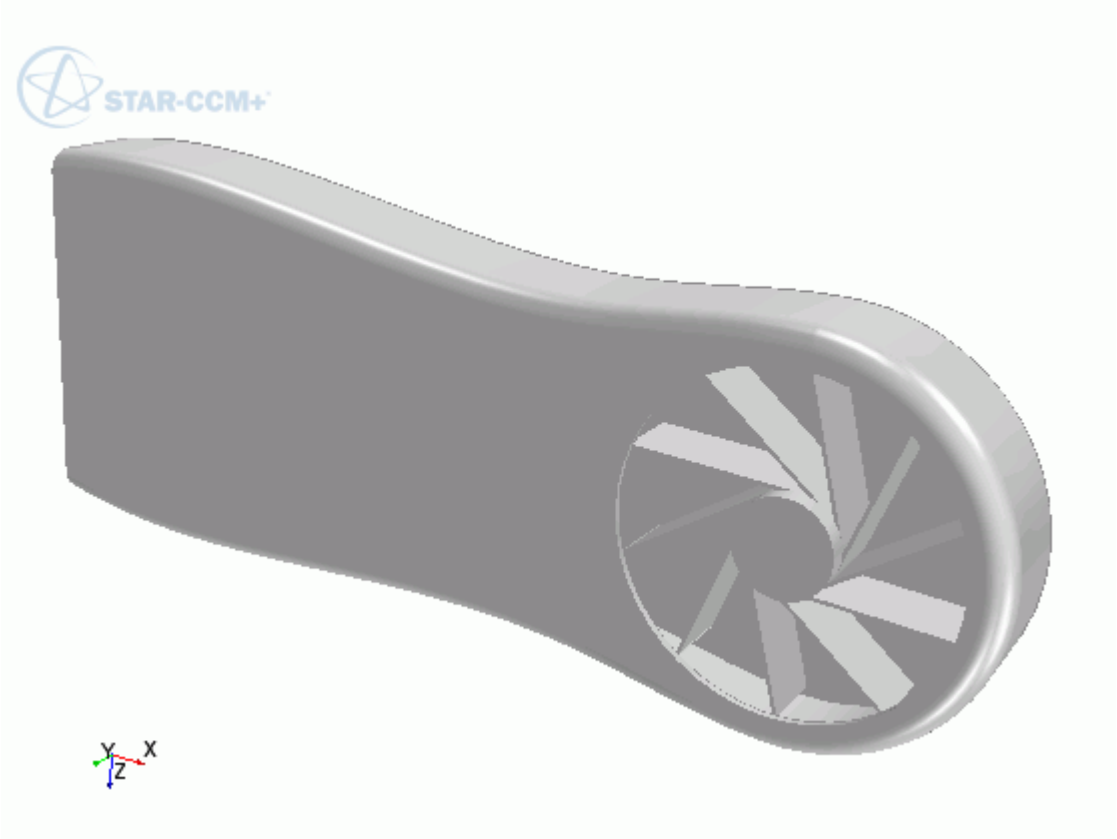
3. Click the checkbox next to the selected parameter to activate it.
4. Click **OK**.

Transfer the geometry to STAR-CCM+:

5. From the STAR-Creo ribbon, click **Transfer Geometry**.

STAR-CCM+ launches and the CAD geometry is transferred over.

6. In STAR-CCM+, create a geometry scene and position the geometry as shown below.



7. In STAR-CCM+, expand the **Geometry** node.

The transferred geometry appears under the **Parts** node. The properties that control the connection back to Creo, including the design parameter value, appear under the **CAD Client Models** node.

When transferring geometry from Creo to STAR-CCM+, a link exists between the two software packages. This link allows data to be transferred between the two packages. To maintain this link, you must first save the geometry file in Creo:

8. Switch to Creo and save the geometry file.

   In response, STAR-CCM+ saves the simulation file with the name of the Creo geometry file. Using this technique, you can rename or relocate both the geometry and simulation files without breaking the link between the two packages. You can also save the simulation file in STAR-CCM+ so long as you do not rename it or relocate it. Doing so will break the link between the packages.

Creating the Shape Parts and Operations

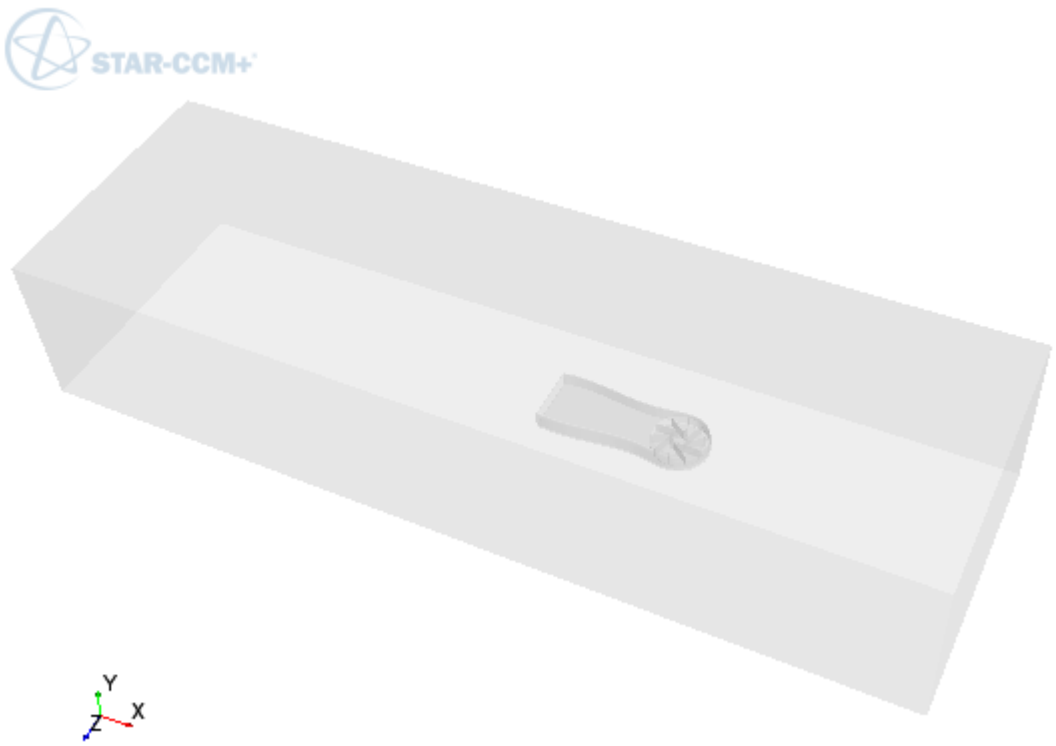
The transferred geometry consists of the card housing and the blower fan. To simulate this case, you require two regions: one static region that provides an enclosure for the simulation, and another for the rotating region around the fan.

The transferred geometry contains the solid parts. You require the fluid volume around these parts. The first stage in obtaining the fluid volume is to create part shapes that correspond to the two regions. To save time, you are provided with these part shapes, one for the enclosure, and another for the rotating region.

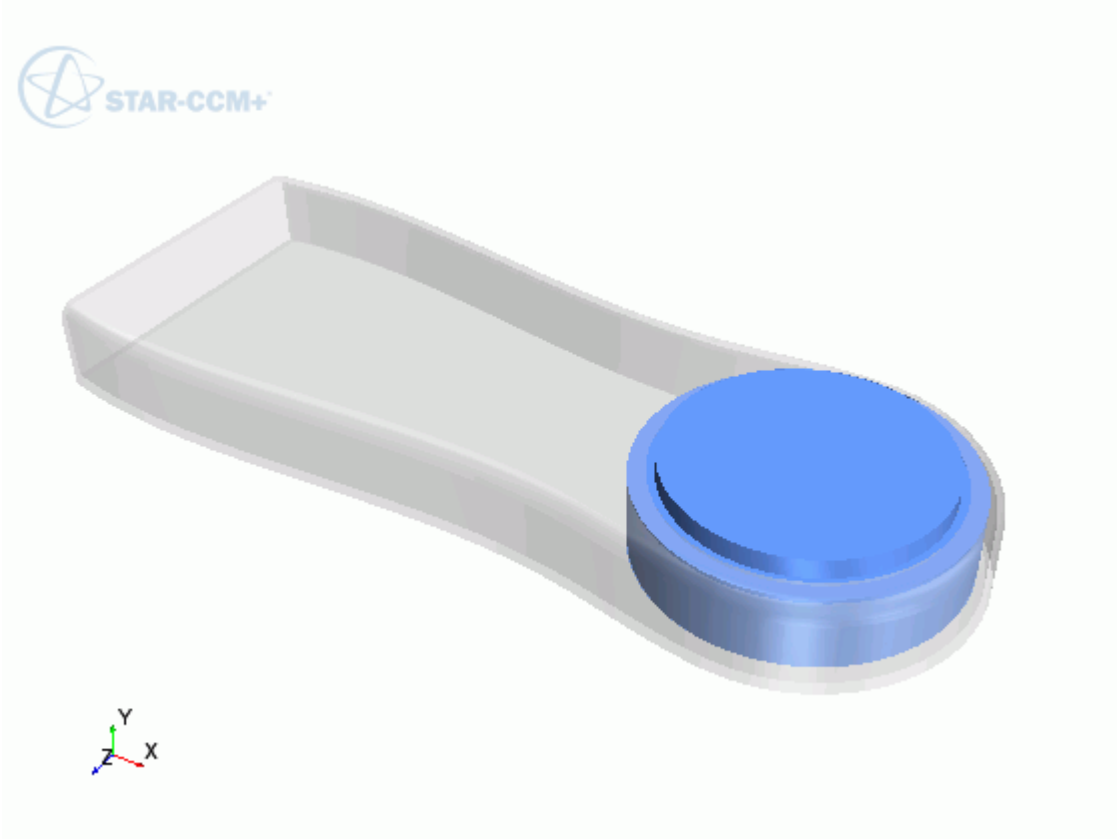
Import the shape parts:

1. Select **File** > **Import** > **Import Surface Mesh**.
2. Import the part shapes:
   a) In the *Open* dialog, navigate to the `data` directory and select **box.x_t** and **disc.x_t**.
   b) Click **Open**.
   c) In the *Import Surface Options* dialog, deactivate the **Open Geometry Scene After Import** option.
   d) Click **OK**.

   The static region, **box**, is shown below in relation to the graphics card:



The rotating region, **disc**, is show below in relation to the graphics card:



Now subtract the solid CAD geometry from these shape parts to obtain the final fluid volume. Begin by subtracting the blower fan geometry from the **disc** part:

3. Right-click the **Operations** node and select **New** > **Boolean** > **Subtract**.
4. In the *Create Subtract Operation* dialog, set the following properties:

Property	Setting
Input Parts	**disc**
	CARDASSEM > **BLOWERFAN**
Target Parts	**disc**

5. Click **OK**.
6. Rename the **Operations** > **Subtract** node to `Fan Subtraction`.
7. Rename **Parts** > **Subtract** node to `Rotating Region`.

Subtract the card housing geometry and the disc part from the static region:

8. Create another subtract operation with the following settings:

Property	Setting
Input Parts	**box**
	disc
	CARDASSEM > **CARDCASE**

Property	Setting
Target Parts	**box**

9. Rename **Operations** > **Subtract** node to `Card Subtraction`.
10. Rename **Parts** > **Subtract** node to `Static Region`.

To get a conformal mesh between the two regions, imprint both fluid volumes together:

11. Right-click the **Operations** node and select **New** > **Boolean** > **Imprint**.
12. In the *Create Imprint Operation* dialog, select **Rotating Region** and **Static Region** and click **OK**.

The transferred geometry contains CAD data. To ensure best results, instruct all operations to use the CAD data where available:

13. Under the **Operations** node, multi-select the **Fan Subtraction** and **Card Subtraction** nodes and activate the *Perform CAD Boolean* option.
14. Select the **Operations** > **Imprint** node and set *Merge/Imprint Method* to **CAD Imprint**.

    The contacts between the two parts are created. These contacts are used while creating the regions to define the interfaces between the regions.

Execute all operations:

15. Right-click on the **Operations** node and select **Execute All**.

Setting Up the Mesh Pipeline

Assign the parts to regions and create the automated mesh operation.

Assign the fluid parts to regions:

1. Expand the **Geometry** > **Parts** node and multi-select the **Rotating Region** and **Static Region** nodes.
2. Right-click on one of the selected parts and choose **Assign Parts to Regions...**.
3. In the *Assign Parts to Regions* dialog:
   a) Select **Create a Region for Each Part**.
   b) Select **Create Boundary for Each Part Surface**.
   c) Click **Apply** and **Close**.

Create the automated mesh operation:

4. Right-click the **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
5. In the *Create Automated Mesh Operation* dialog:
   a) From the *Parts* list, select **Rotating Region** and **Static Region**.
   b) Select the following meshers, in order:

Group	Mesher
Surface Meshers	**Surface Remesher**
Volume Meshers	**Polyhedral Mesher**
Optional Boundary Layer Meshers	**Prism Layer Mesher**

c) Click **OK**.

An automated mesh operation is added to the **Operations** node. Refine the mesh settings for this case:

6. Expand the **Automated Mesh** > **Default Controls** node.
7. Select the **Base Size** node and set *Value* to `16 mm`.
8. Select the **Prism Layer Total Thickness** > **Relative Size** node and set *Percentage of Base* to `20`.

Generate the volume mesh:

9. Click (**Generate Volume Mesh**).
10. When the volume mesh is generated, create a mesh scene that displays the following parts:
    - **Rotating Region** > **Boundaries** > **CARDASSEM.BLOWERFAN.Faces**
    - **Static Region** > **Boundaries** > **CARDASSEM.CARDCASE.Faces**

You can exit STAR-CCM+ and Creo and maintain the link between the two packages. To demonstrate, in the following steps you save the simulation and exit both applications. In the next section, you will resume the simulation and continue where you left off.

11. Save the simulation.
12. Exit STAR-CCM+.
13. Exit Creo

Modifying the Design

Modify the number of blades on the blower fan using the design parameter.

In this tutorial, you modify the number of blades on the graphics card blower fan from 10 to 15 blades.

To re-establish the link between Creo and STAR-CCM+, you must load the geometry in STAR-Creo first and transfer the geometry to STAR-CCM+.

Resume the analysis and re-establish the link between STAR-Creo and STAR-CCM+:

1. Launch Creo with STAR-Creo and load `cardassem.asm`.
2. From the STAR-Creo ribbon, click **Transfer Geometry**.

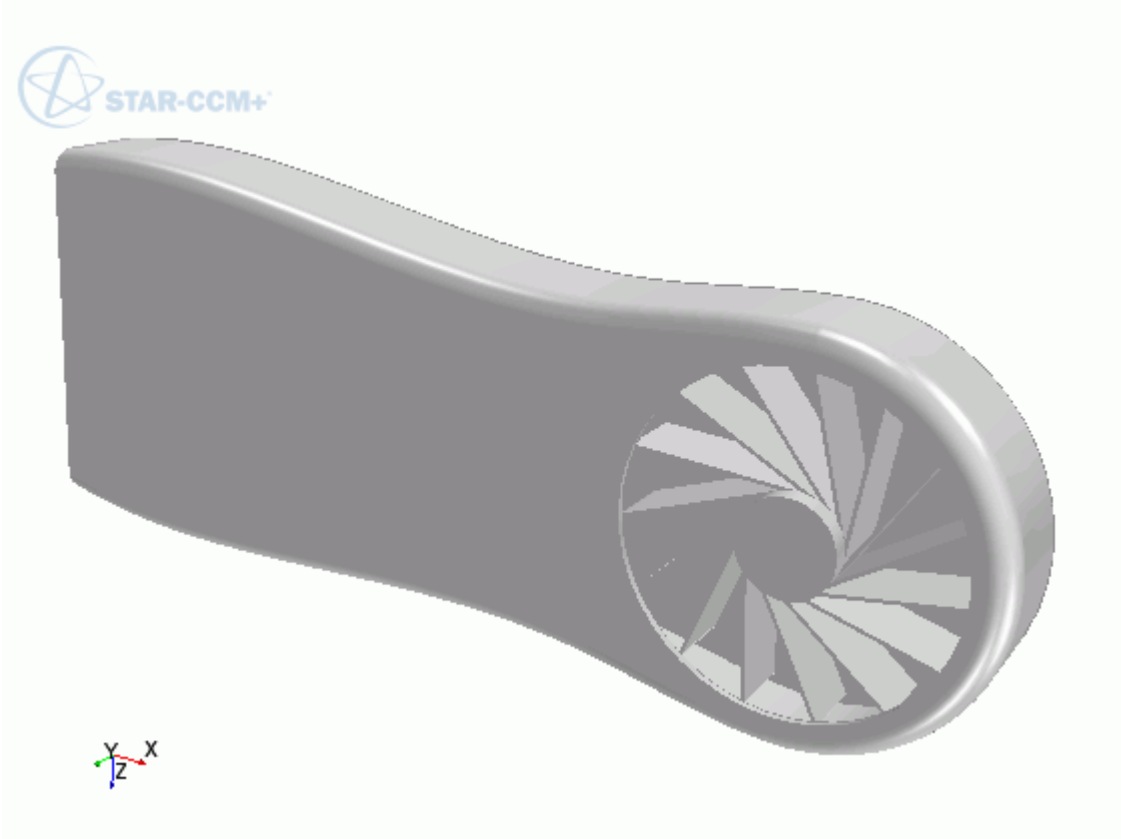
   STAR-CCM+ launches and the CAD geometry is transferred over.
3. Switch to STAR-CCM+.
4. Open the geometry scene to visualize the initial geometry representation.
5. Expand the **Geometry** > **CAD Client Models** > **cardassem.asm** > **Design Parameters** node.
6. Select the **blowerfan.prt\p14** node and set *Parameter Value* to `15`.

   A yellow warning icon appears on the design parameter node as well as the assembly node. This icon indicates that the CAD Client Model is out of date.

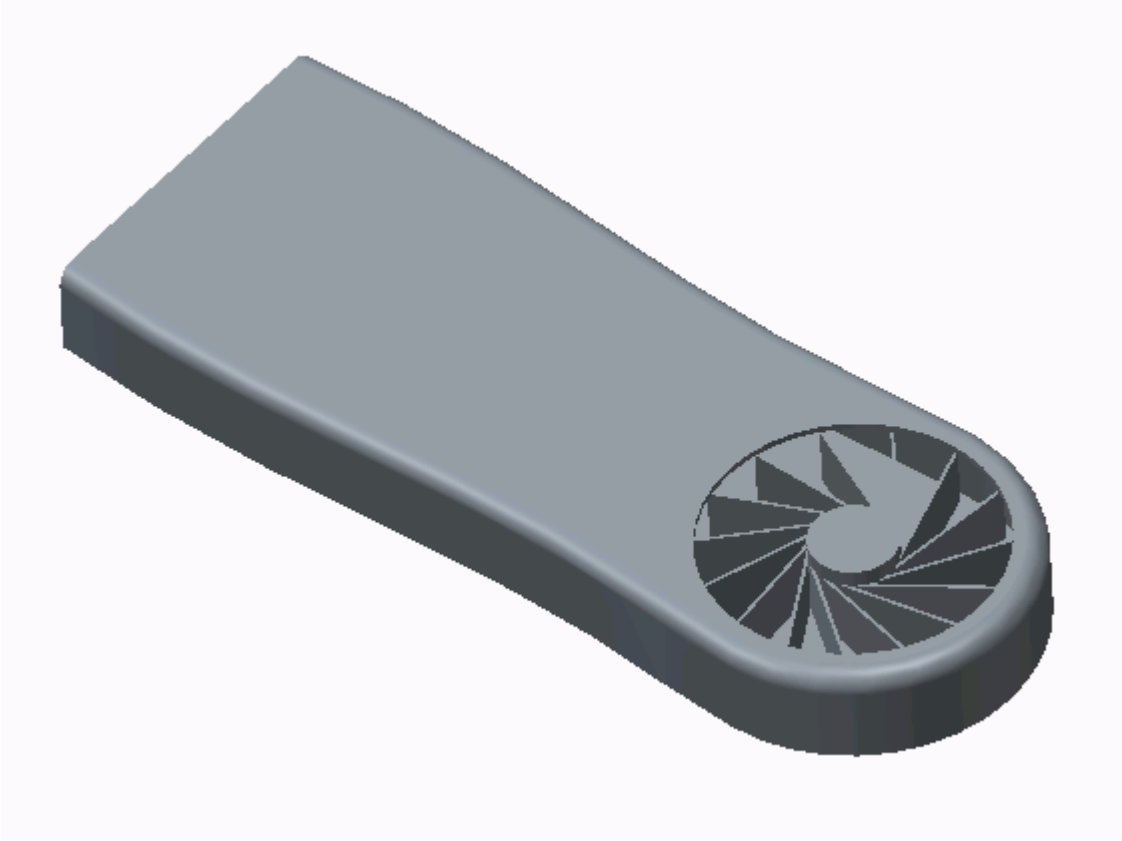
To update the model:

7. Right-click the **cardassem.prt** node and select **Update Model**.

   The geometry updates in the scene. STAR-CCM+ passes the change in the design back to Creo.



8. Switch back to Creo and view the updated geometry.



The pipeline you set up is fully repeatable. To update the volume mesh:

9. In STAR-CCM+, open the mesh scene.

10. Click (**Generate Volume Mesh**).
    The volume mesh updates to reflect the changes in the design.

All boundaries and boundary conditions that you set are maintained after this change. This feature reduces the amount of work that is required when analyzing multiple designs.

Siemens PLM Software

Headquarters
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asia-Pacific
Suites 4301-4302, 43/F
AIA Kowloon Tower,
Landmark East
100 How Ming Street
Kwun Tong, Kowloon
Hong Kong
+852 2230 3308

About Siemens PLM Software

Siemens PLM Software, a business unit of the Siemens Digital Factory Division, is a leading global provider of product lifecycle management (PLM) and manufacturing operations management (MOM) software, systems and services with over 15 million licensed seats and more than 140,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software works collaboratively with its customers to provide industry software solutions that help companies everywhere achieve a sustainable competitive advantage by making real the innovations that matter. For more information on Siemens PLM Software products and services, visit www.siemens.com/plm.

© 2017 Siemens Product Lifecycle Management Software Inc. Siemens and the Siemens logo are registered trademarks of Siemens AG. D-Cubed, Femap, Fibersim, Geolus, GO PLM, I-deas, JT, NX, Parasolid, Polarion, Solid Edge, Syncrofit, Teamcenter and Tecnomatix are trademarks or reg istered trademarks of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries. All other logos, trademarks, registered trademarks or service marks belong to their respective holders.

SIEMENS
Ingenuity for life

Siemens PLM Software

STAR-Inventor
User Guide

Version
12.06

www.siemens.com/mdx

© 2017 Siemens Product Lifecycle Management Software Inc.

Any and all Siemens Product Lifecycle Management Software Inc. brand, product, service and feature names, logos and slogans are registered trademarks or trademarks of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries. All other brand, product, service and feature names or trademarks are the property of their respective owners.

Contents

STAR-Inventor **1**
STAR-Inventor User Guide **2**
Introducing STAR-Inventor 3
Transferring Geometry to STAR-CCM+ 36
Understanding the STAR-Inventor Process 44
Connecting to Remote Machines 48
Defining Materials and Physics 52
Setting Boundary Conditions 76
Meshing the Model 83
Obtaining a Solution 108
Exporting Files and Data 117
Resetting STAR-Inventor Settings 121
Guidelines 122
STAR-Inventor Tutorials **162**
Cold Flow Through a Motorcycle Exhaust Manifold 163
Porous Media Flow 186
Aircraft Wing Deicing System 208
Rotating Flow Inside a Graphics Card 243
External Flow Over a Car 275
Geometry Transfer: CAD to Simulation 306

STAR-Inventor

The STAR-Inventor CAD client provides you with straightforward access to CD-adapco's STAR-CCM+ solver from within the Inventor environment.

Standard physics models, including turbulence, ideal gas, heat transfer, and gravity are supported for single body cases. Multiple body simulations involving conjugate heat transfer, rotating reference frames, and porous media are also supported. Advanced physics models can be added to the simulation by either launching STAR-CCM+ and selecting them directly, or by implementing additional Java macro commands.

This section contains the user guide and tutorials for the STAR-Inventor CAD client.

Contents:

STAR-Inventor User Guide
STAR-Inventor Tutorials

STAR-Inventor User Guide

This part of the documentation details how to create, set up, and run a Computational Fluid Dynamics (CFD) simulation in STAR-Inventor.

This section assumes that you have previous experience in using Inventor, and only describes the additional features available as part of STAR-Inventor.

You can use STAR-Inventor to transfer your CAD geometry directly into STAR-CCM+ without transferring the CAE settings. This method is an alternate way importing geometry to using the STAR-CCM+ CAD translators. See Transferring Geometry to STAR-CCM+.

Contents:

Introducing STAR-Inventor
Transferring Geometry to STAR-CCM+
Understanding the STAR-Inventor Process
Connecting to Remote Machines
Defining Materials and Physics
Setting Boundary Conditions
Meshing the Model
Obtaining a Solution
Exporting Files and Data
Resetting STAR-Inventor Settings
Guidelines

Introducing STAR-Inventor

STAR-Inventor provides you with straightforward access to CD-adapco's STAR-CCM+ solver from within the Inventor environment.

Standard physics models, including turbulence, ideal gas, heat transfer, and gravity are supported for single body cases. Multiple body simulations involving conjugate heat transfer, rotating reference frames, and porous media are also supported. Advanced physics models can be added to the simulation by either launching STAR-CCM+ and selecting them directly, or by implementing additional Java macro commands.

Contents:

What are the Capabilities?
Getting Started
Common Functions
Running STAR-Inventor and STAR-CCM+ on Different Machines

See Also:

Using Java Macros in STAR-Inventor

What are the Capabilities?

This section highlights the capabilities of STAR-Inventor.

STAR-Inventor enables you to:

- Define flow characteristics for the simulation
- Specify mesh parameters and controls
- Execute the STAR-CCM+ solver within STAR-Inventor, with specified solution controls
- Post-process results in STAR-CCM+
- Use parallel processing

Problem Definition

STAR-Inventor provides an interface for the specification of continua, regions, and boundaries in order to define the CFD simulation. This information is then used by STAR-CCM+ to calculate a solution and present results.

The particular solver options that you can currently select through the STAR-Inventor interface include:

- Turbulence
  - k-e Model
  - Laminar
- Temperature
  - Thermal Modelling

  - Conjugate Heat Transfer
- Equation of State
  - Constant Density
  - Ideal Gas
- Porous Media
  - Isotropic
  - Uni-Directional
- Moving Reference Frames
- Energy Source
- Gravity
- Passive Scalar

Meshing

STAR-Inventor provides you with a transparent interface to CD-adapco's surface and volume meshing tools. This allows you to generate high-quality meshes using a solid model that was built within the Inventor environment. The specific meshing functionality includes:

- Surface meshing using the surface remesher for optimum mesh quality
- Prismatic subsurface meshing in the volume mesh for capturing near-wall boundary layers
- Volume meshing of parts with either:
  - Tetrahedral core mesh generation
  - Polyhedral core mesh generation
  - Trimmed hexahedral core mesh generation
- Automatic handling of region-to-region interfaces

Running the Solver

STAR-Inventor allows the generation of solutions with varying accuracy and speed by providing the following solver controls:

- Number of iterations
- Stopping criterion at which to assume convergence
- Discretization scheme (1st or 2nd order)
- Solver under-relaxation factors
- Additional commands using Java macros

Post-processing

As post-processing is carried out in STAR-CCM+, a wide range of tools are available for analysis. These include:

- Scalar scenes
- Vector scenes
- Streamlines
- Isosurfaces

- Plane sections
- Reports
- Plots

Parallel Processing

STAR-Inventor supports parallel processing for mesh generation and solver calculations. You can set up parallel processing in the Preferences dialog. The different STAR-CCM+ licensing options can also be selected here. See Setting Up Parallel Processing and Licensing.

Getting Started

The following section provides information on familiarization with the STAR-Inventor interface, and basic concepts.

Contents:

Launching STAR-Inventor Manually
The STAR-Inventor Info App Window
Modifying the config.xml File
Understanding the STAR-Inventor Interface
The STAR-Inventor Object Tree
Setting Up Parallel Processing and Licensing
Changing the Units System
Saving Files in Inventor
Transferring the CAE Model to STAR-CCM+
Editing STAR-Inventor Settings While the STAR-CCM+ Client is Running

Launching STAR-Inventor Manually

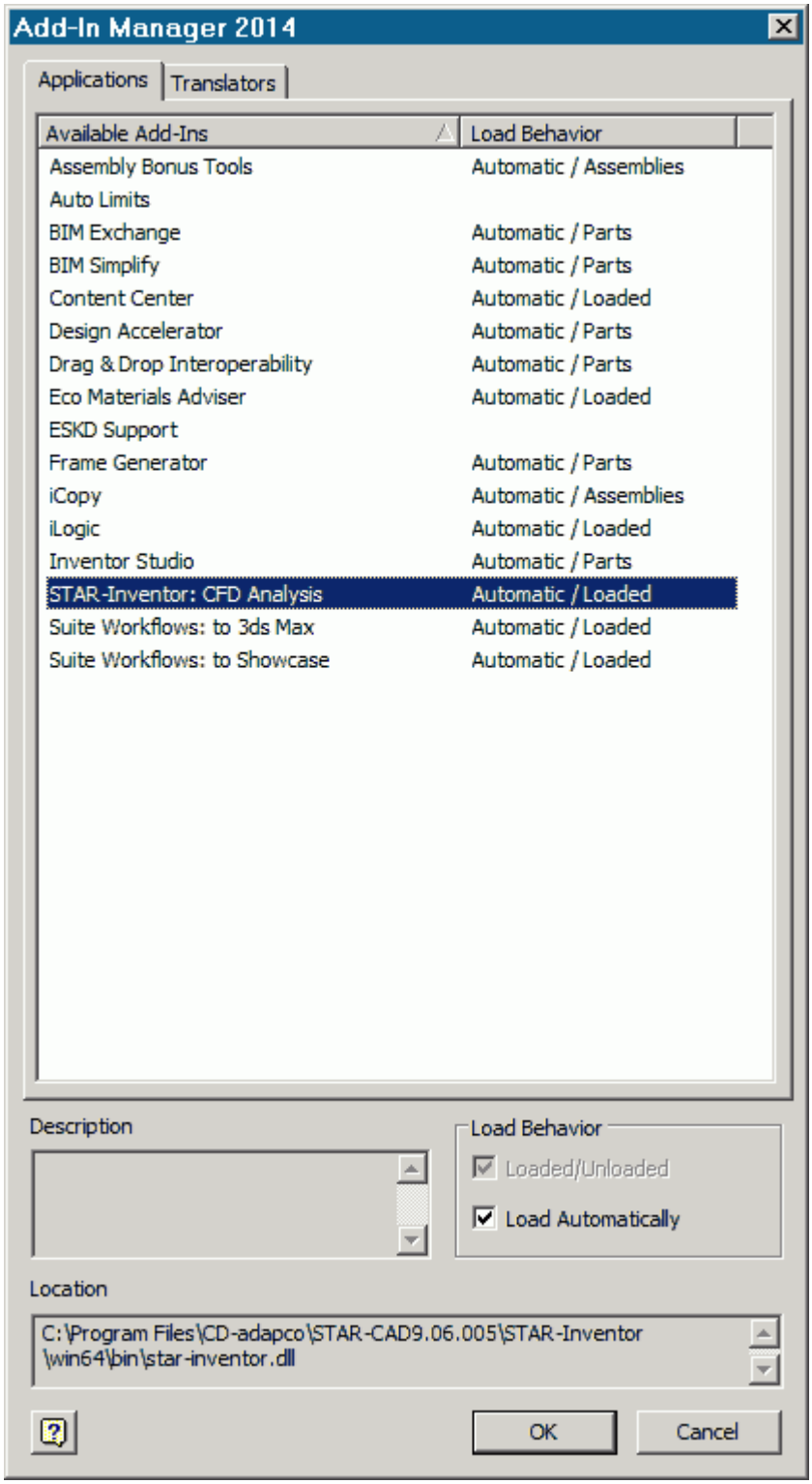
By default STAR-Inventor starts automatically once Inventor launches.

If this does not happen, you can launch STAR-Inventor manually by following the steps outlined below:

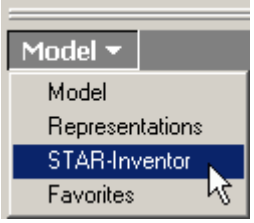
1. Start Inventor.
2. Click on the **Tools** tab to reveal the *Tools* ribbon.
3. Click the **Add-Ins** button on the *Tools* ribbon.

   The *Add-In Manager 20xx* dialog appears.
4. Select **STAR-Inventor: CFD Analysis** from the list and activate the **Loaded/Unloaded** option.

   This option activates STAR-Inventor for the current session only.
5. Activate the **Load on Startup** option to automatically load STAR-Inventor every time you launch Inventor.



6. Click **OK** to apply the settings.
7. Click **Model** and select **STAR-Inventor**.

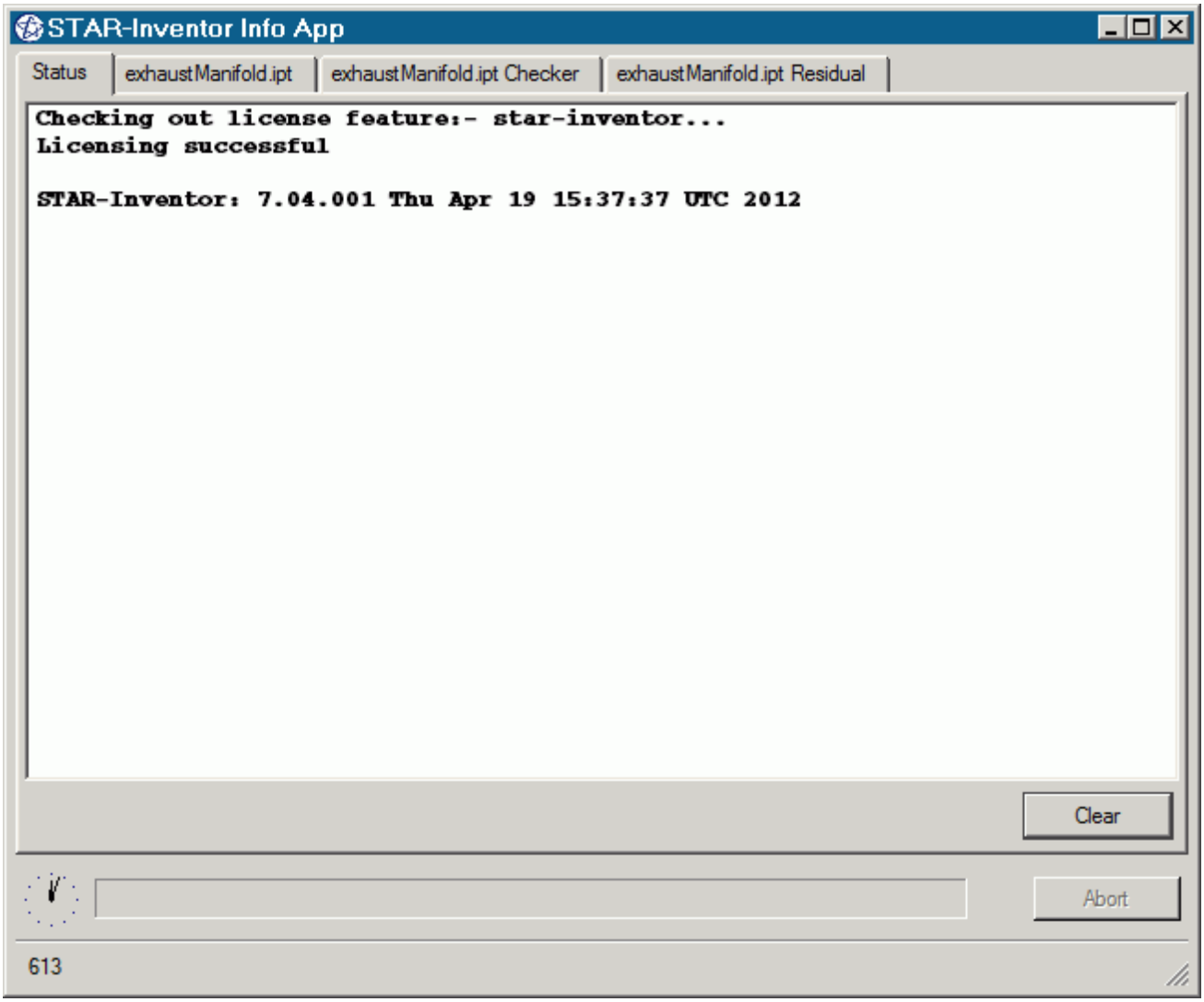


The STAR-Inventor object tree appears to the left of the screen. The STAR-Inventor ribbon also appears.

The STAR-Inventor Info App Window

The STAR-Inventor Info App window automatically appears when you start Inventor.

Initially, this window only contains the *Status* tab which displays the version and licensing information for STAR-Inventor. More tabs appear as you create or load geometry files in STAR-Inventor, allowing you to perform more actions.



You cannot close the STAR-Inventor Info App window while Inventor or STAR-Inventor are running. Attempting to close the window by clicking the **X** button returns a message in the *Status* tab informing you of this. You can maximize and minimize the window by clicking the **-** and **_** buttons respectively.

Use the **Clear** button to clear all the information that is displayed in a particular tab.

Use the **Abort** button to stop active processes, such as surface/volume meshing, or the solver. This button only appears when you can stop a process — it is grayed out at all other times.

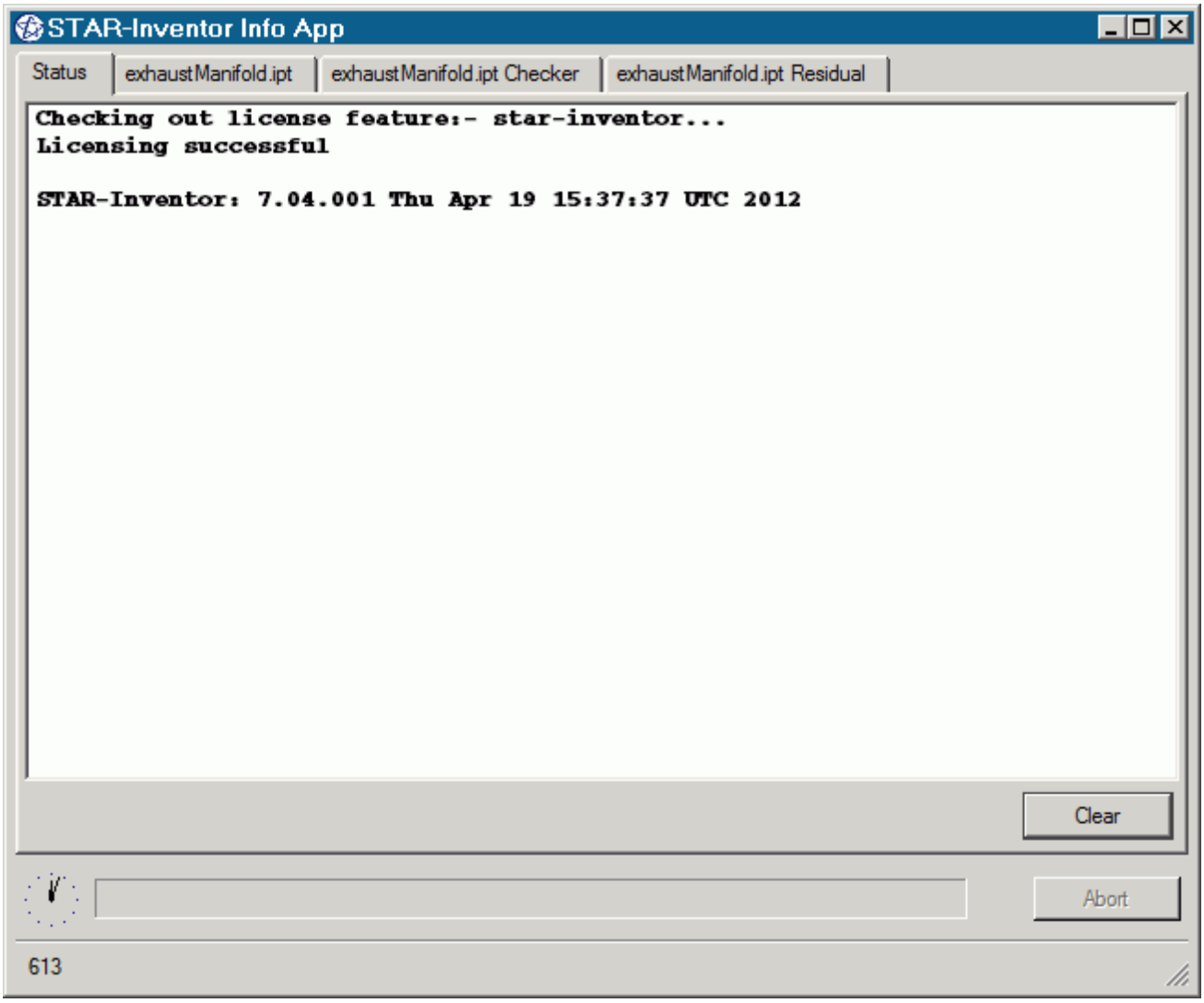
A clock is displayed in the bottom-left of the window. This clock displays the current time on your machine. A progress bar is located next to the clock and reports the progress of various processes, such as mesh generation and the solver.

More information on the various tabs is provided in the following sections:

- The Status Tab
- The Output Tab
- The Model Checker Tab
- The Residuals Tab

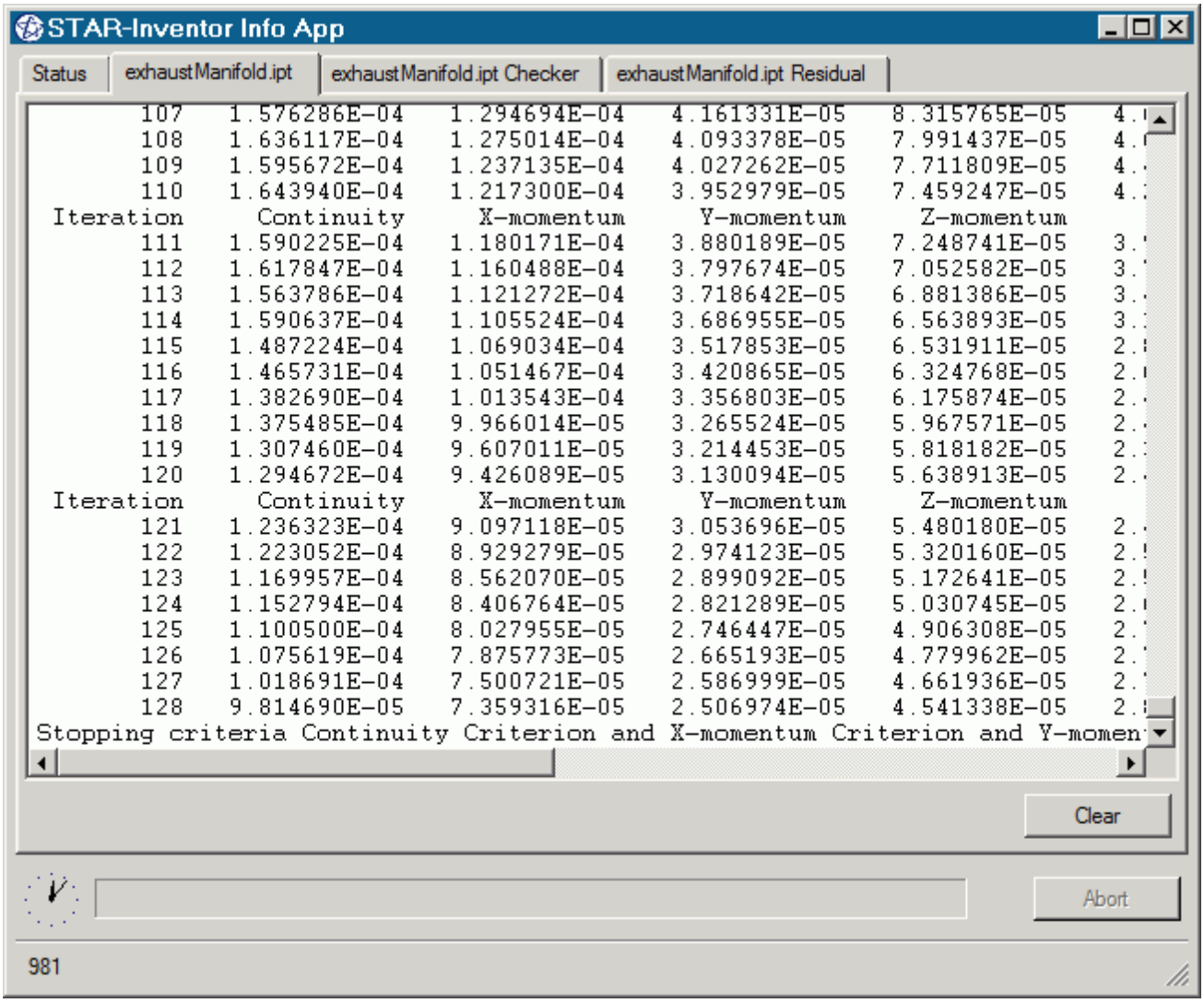
The Status Tab

The *Status* tab displays the licensing information for your current session, informing you whether the STAR-Inventor license checked out successfully. The version of STAR-Inventor is also displayed in this tab. This tab appears by default when you launch Inventor without a geometry file.



The Output Tab

The *Output* tab appears once you create or open a geometry file in STAR-Inventor.



During the meshing process, this tab displays status information along with a brief report on the generated mesh. If errors were encountered during the meshing process, they are reported here with suggestions as to what can be causing them.

When the solver is running, the *Output* tab behaves like the output window in STAR-CCM+, displaying the data for each residual per iteration. When the solver finishes running, the same information is also displayed in the STAR-CCM+ *Output* window.

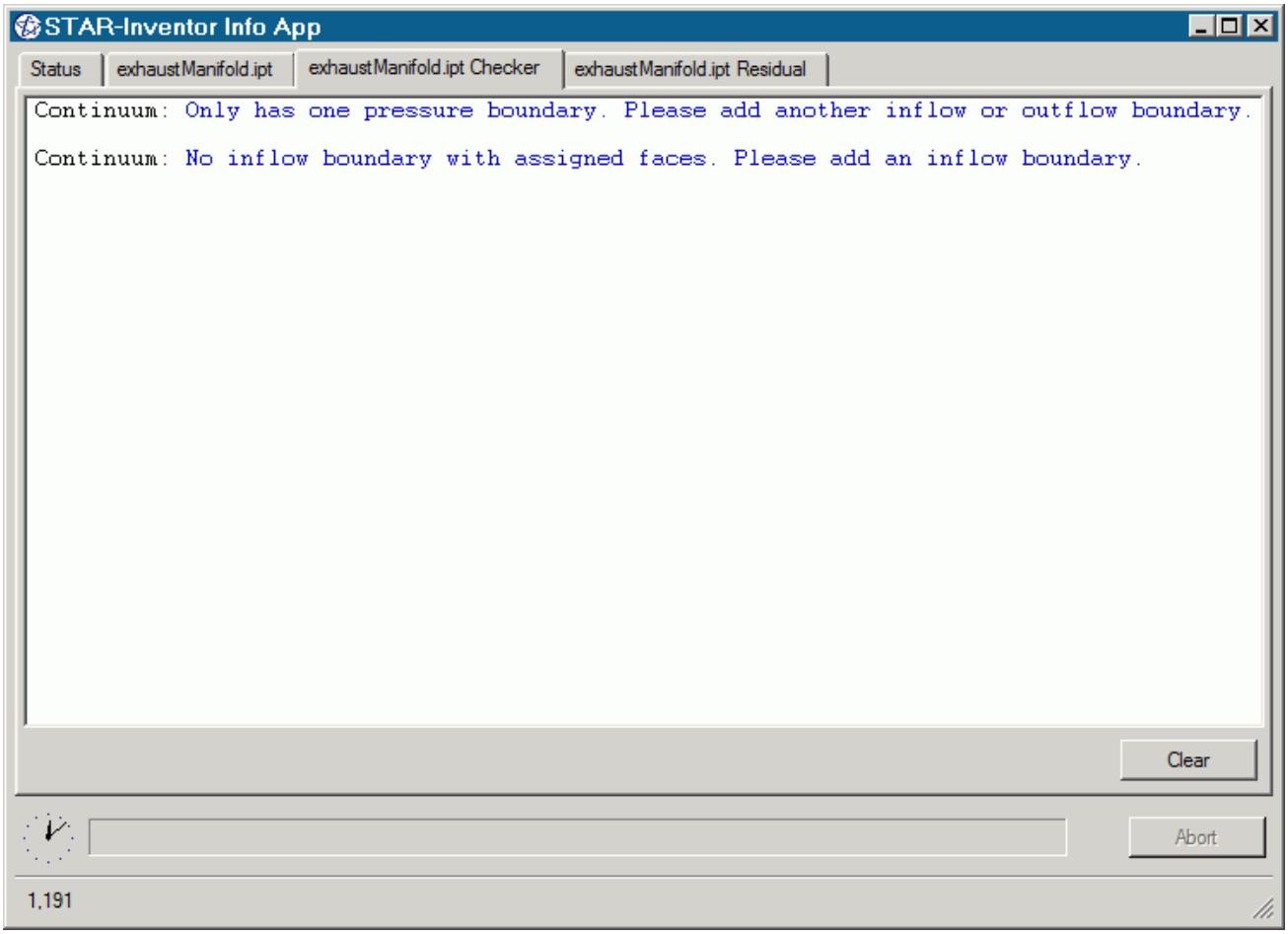
You can control the amount of information that is displayed in the *Output* tab by editing the verbosity in the `config.xml` file. See Modifying the Config.xml File.

To abort the mesher or solver, click the **Abort** button.

The Model Checker Tab

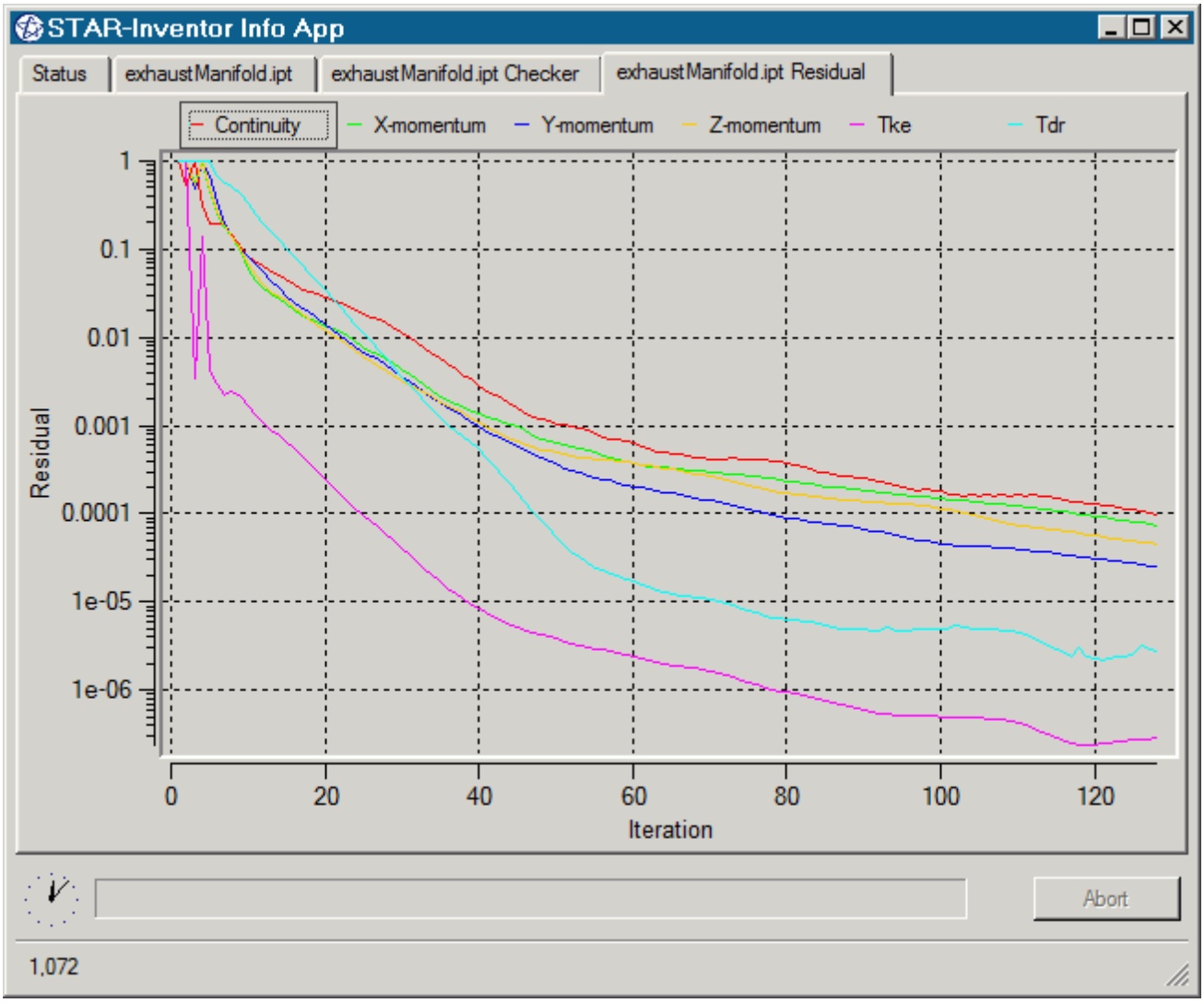
The *Model Checker* tab displays information from the live model checker, where the case setup is actively monitored for any problems. See Checking the Model for Errors.

The following example reports that the case has no inlet boundary defined.



The Residuals Tab

The *Residuals* tab appears when you launch the solver. Use the residuals to monitor the convergence of the solution being calculated.



Typically, a drop in three to five orders of magnitude in the residuals indicates that the solution has converged to an acceptable level. The residuals that are displayed in this tab also appear in STAR-CCM+ under the **Plots** node.

You can zoom in to the residuals by clicking and holding the left mouse button in the graph area and drawing a box around the area you want to view. To zoom out, right-click in the graph area. You can hide individual residuals from the *Residuals* tab by clicking their label at the top of the plot window.

Modifying the config.xml File

STAR-CCM+ requires a configuration file to enable it to find the appropriate CD-adapco components to run correctly. This configuration file is created during installation. An explanation of the file and settings within it is given here.

The configuration file is called `config.xml` and STAR-Inventor searches for it in the location given by the `STARCADSERIES` environment variable.

Typical `config.xml` content is shown below. In this example STAR-CCM+ version 12.06.### is shown.

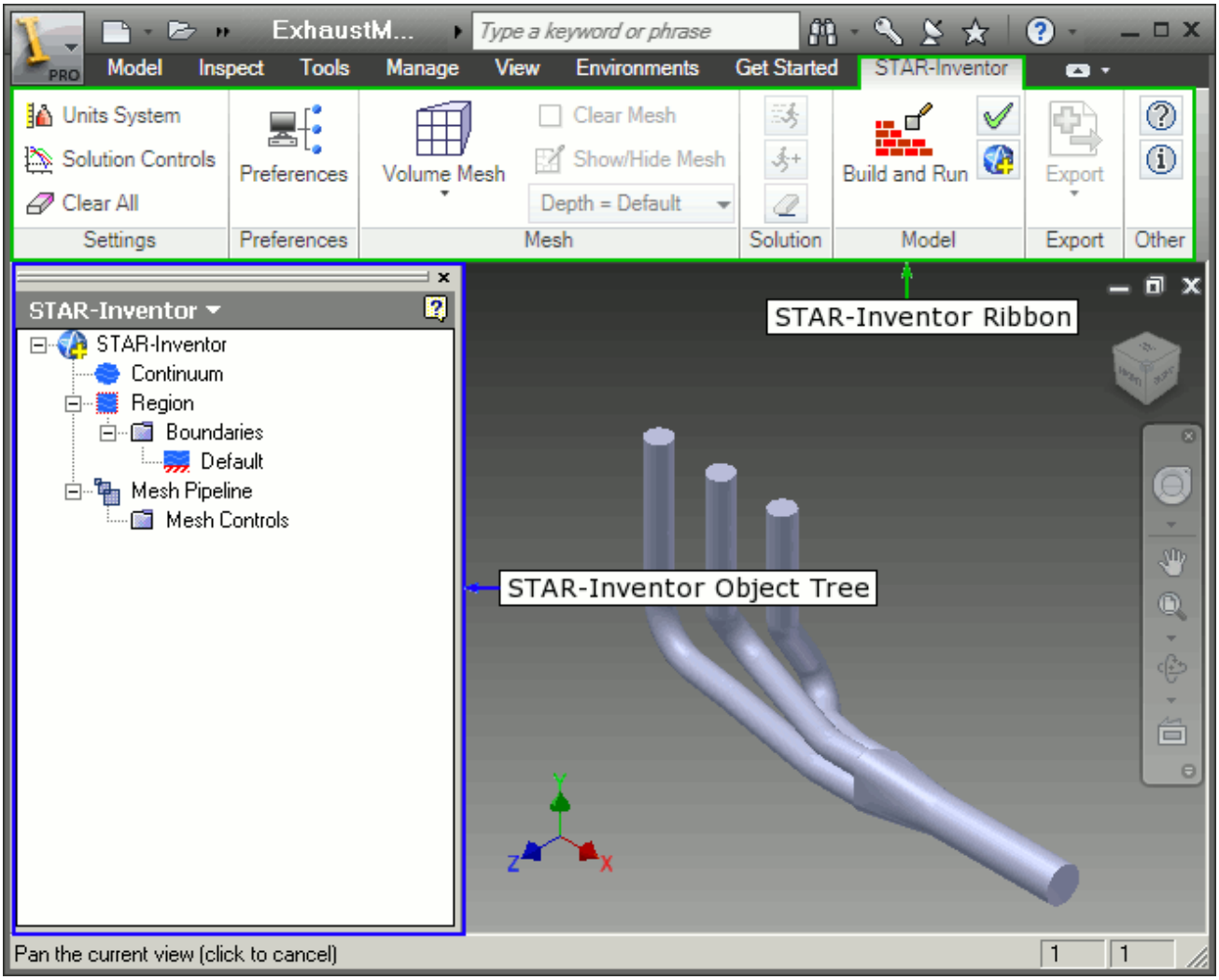
```
<CAD>
  <STAR_HOME>C:\Program Files\CD-adapco\12.06.###\STAR-CCM+12.06.###
\star</STAR_HOME>
  <SOLVER>C:\Program Files\CD-adapco\12.06.###\STAR-CCM+12.06.###\star
\bin\starccm+.exe</SOLVER>
  <LOGFILE>starcad.log</LOGFILE>
  <WWHELP>C:\Program Files\CD-adapco\12.06.###\STAR-CCM+12.06.###
\win64\Intel11.1\wwhelp.exe</WWHELP>
  <MANDIR>C:\Program Files\CD-adapco\12.06.###\STAR-CCM+12.06.###\doc
\en\online</MANDIR>
  <VERBOSITY>Normal</VERBOSITY> <!-- Can be Verbose, Log or Normal.
Default Normal -->
</CAD>
```

You can edit the following lines:

- `<SOLVER>` provides the location of the STAR-CCM+ solver executable. STAR-CCM+ licensing options are set in the *Preferences* dialog in the STAR-Inventor user interface. See Setting Up Parallel Processing and Licensing.
- `<LOGFILE>` is the log file that will automatically be created and saved when STAR-Inventor is run. A path of your choice can be used.
- `<VERBOSITY>` is the verbosity of the output window in the STAR-Inventor Info App. This can be set to one of the following:
  - `Normal`—This is the default setting and will display the least amount of information in the *Output* panel.
  - `Verbose`—This will display a moderate amount of information in the *Output* panel.
  - `Log`—This will display all available information in the *Output* panel.

Understanding the STAR-Inventor Interface

The STAR-Inventor user interface is embedded within the Inventor environment, with the addition of the STAR-Inventor object tree and ribbon.



The majority of the STAR-Inventor operations are executed within the object tree and ribbon. Certain actions can also be carried out using multiple methods. A summary of the features available in each part of the STAR-Inventor interface are detailed below.

Object Tree

The operations provided in the object tree include:

- Solution Controls
- Definition of Units System
- Continuum settings
- Region settings
- Add new boundary
- Boundary conditions
- Mesh Pipeline settings
- Add new mesh controls

Ribbon

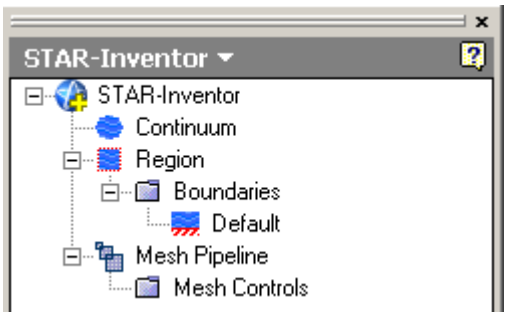
- Settings
  - Definition of Units System
  - Solution Controls
  - Clear All
- Preferences
- Mesh
  - Generate Surface Mesh
  - Generate Volume Mesh
  - Clear Generated Meshes
  - Show/hide Mesh
  - Depth
- Solution
  - Build and Run
  - Run Solver
  - Run For More Iterations
  - Clear Solution
- Model
  - Check Model for Errors
  - Transfer CAE Model
- Export
  - Surface Mesh (.dbs)
  - Volume Mesh (.ccm)
  - Engineering Report (.html, .csv)
  - Simulation (.sim)
- Geometry
  - Transfer Geometry
  - Design Parameters
- STAR-Works Help

- About STAR-Works

The STAR-Inventor Object Tree

STAR-Inventor uses an object tree as the primary method for setting parameters.

The object tree consists of nodes, each one of which represents a particular component of the simulation. This approach to the user interface complements the method that is used by STAR-CCM+, allowing for an easy transition between the two software packages.



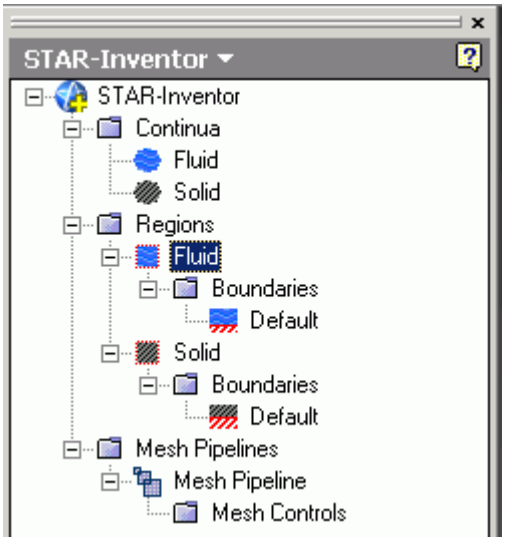
The object tree appears slightly different in assembly mode as it must accommodate multiple parts, but the principles of operation remain the same.

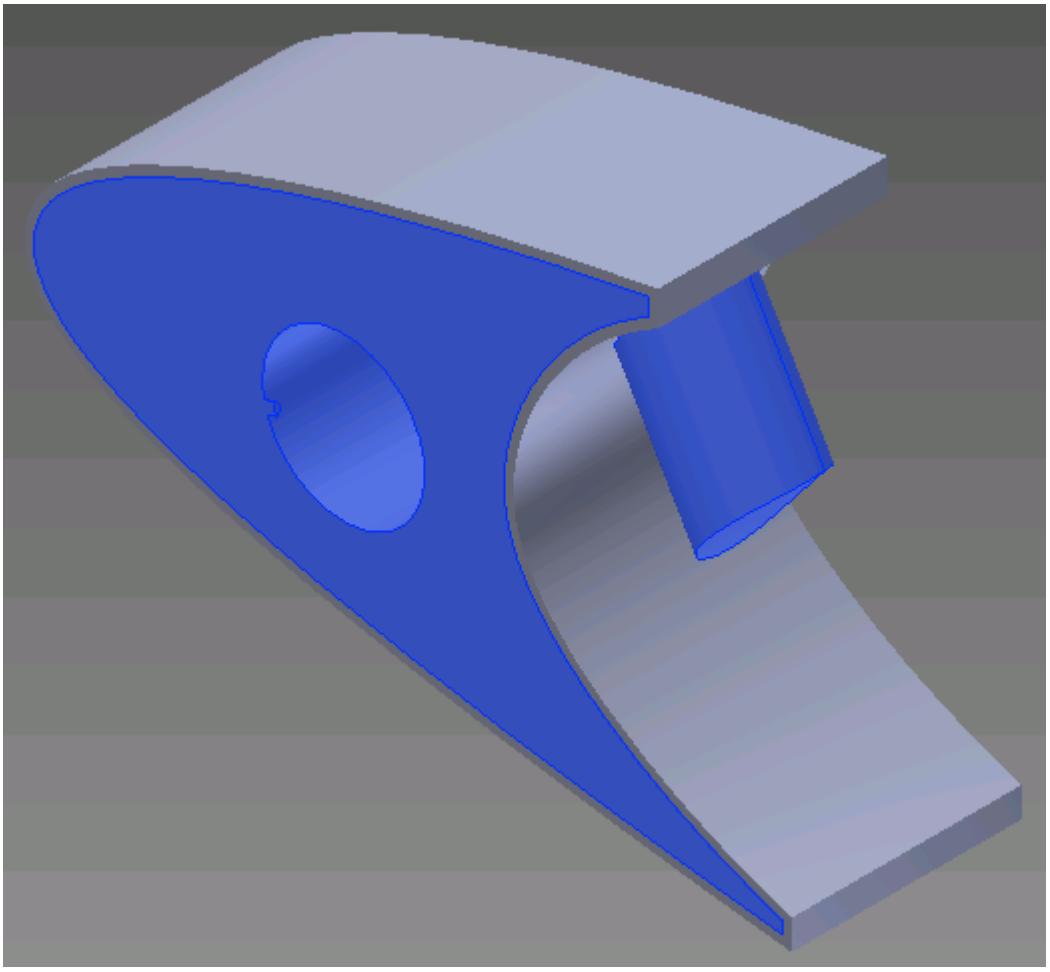
Nodes are arranged in an order that follows the recommended workflow for setting up a simulation in STAR-Inventor. At each stage, you typically right-click on a particular node in the tree and choose the appropriate action from the pop-up menu.

Certain functionality within STAR-Inventor applies to the simulation as a whole, rather than to individual objects. These operations are provided in the ribbon.

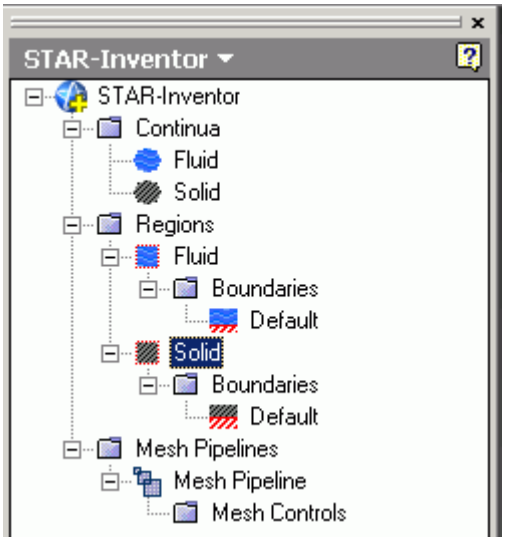
A complete list of all functions and their respective location is included in Understanding the STAR-Inventor Interface.

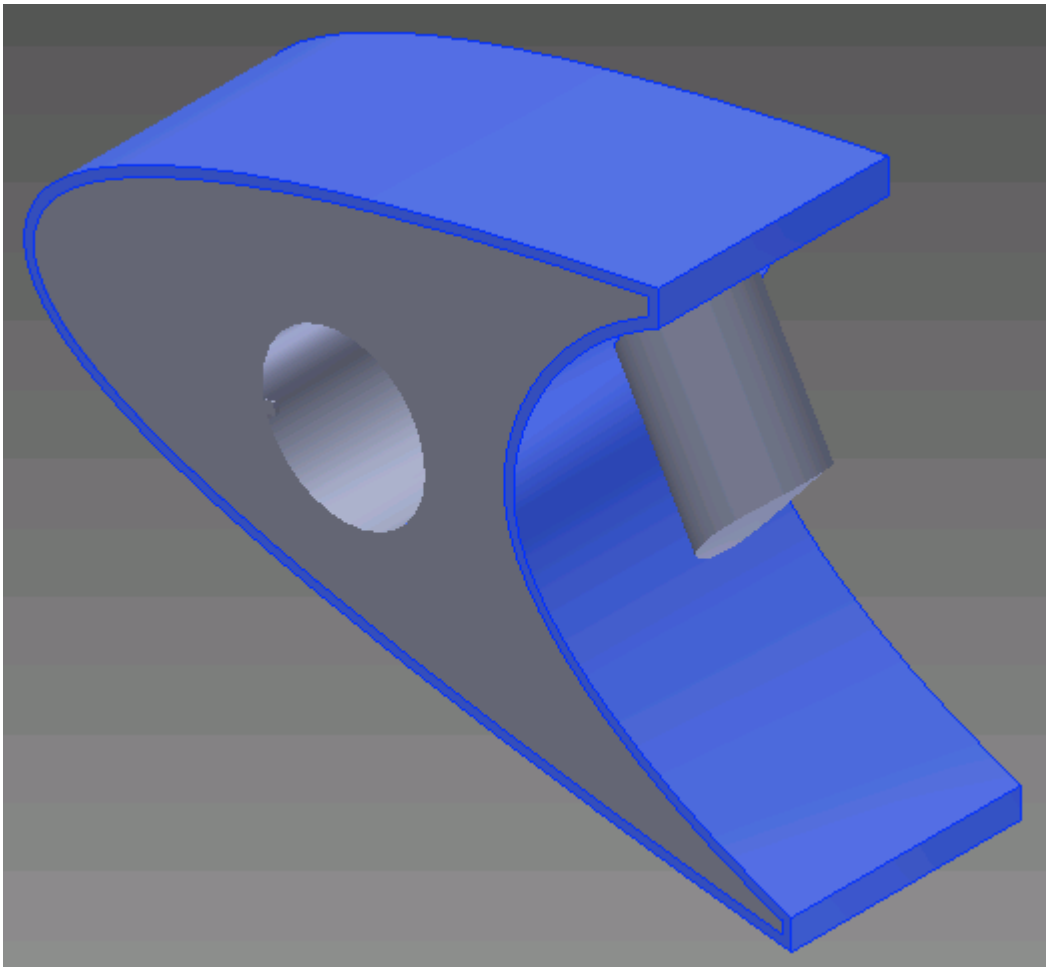
Clicking on the **Region** node within the object tree will highlight all the relevant faces on the geometry model for that particular region. If different parts are assigned to multiple regions, then clicking on a region will only highlight the part and faces associated with that specific region. For example, clicking on the **Fluid** node below will highlight all the faces associated with the Fluid region.





Similarly, clicking on the **Solid** node highlights all the faces associated with the Solid region.





Clicking on a boundary node in the STAR-Inventor object tree highlights all the faces on the CAD geometry model that are associated with it. Note that clicking on the *Default* boundary node will not highlight any faces on the CAD geometry model. To stop highlighting the faces, you can either click on a blank space in the geometry window, or click on the root node in the STAR-Inventor object tree.

Setting Up Parallel Processing and Licensing

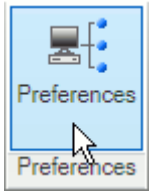
The licensing options and the options for setting up parallel processing are located in the *Preferences* dialog.

This dialog provides two sets of options: server preferences and server settings.

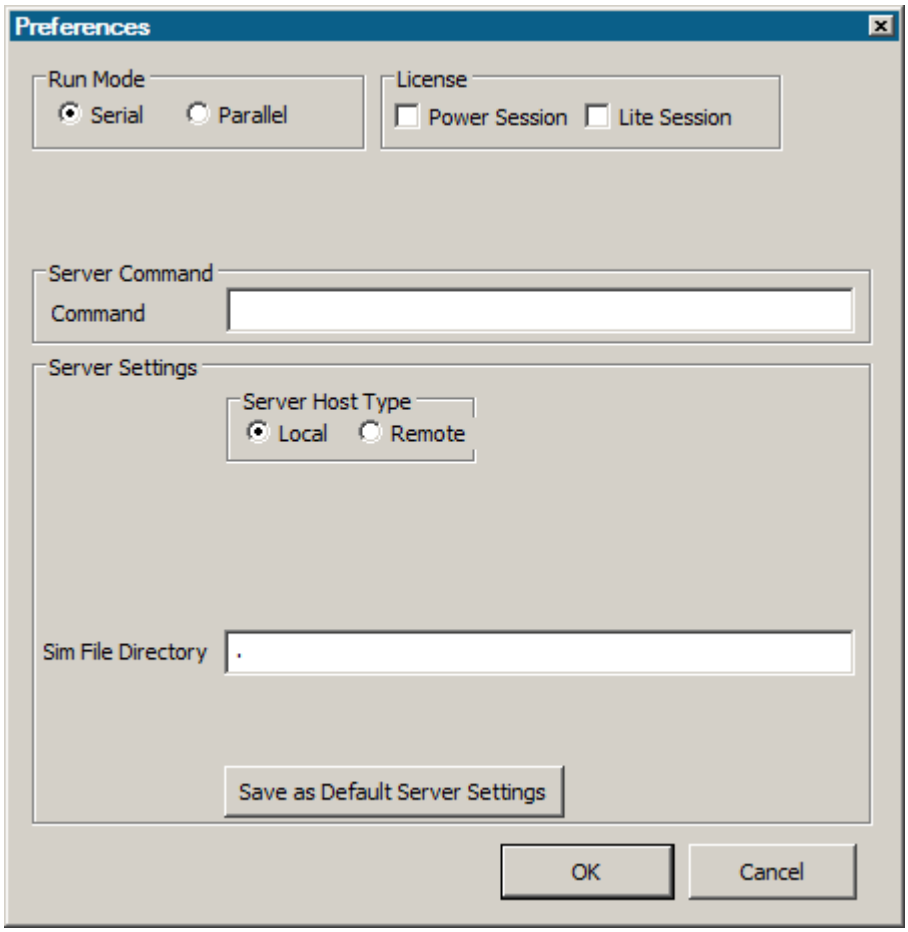
- Server preferences are applied whenever STAR-Inventor starts a server.
- Server settings are saved in the CAD file and are applied only to that CAD file. Click **Save as Default Server Settings** to save the current server settings as the default settings to be applied to all CAD files.

To modify the run mode and licensing options:

1. Select **Preferences** from the ribbon.



The *Preferences* dialog appears.



2. Select the run mode:

- For serial, set *Run Mode* to **Serial** (the default setting).
- For parallel, set *Run Mode* to **Parallel** and *Compute Processes* to the desired number of cores.

  You can use the **up/down** buttons or you can type in the number of cores that you want to use.
- For advanced licensing options, such as Power-on-Demand, enter the appropriate argument in the *Command* field.

  For Power-on-Demand, enter:

  ```
  -licpath 1999@flex.cd-adapco.com -podkey <pod_key>
  ```

  Where:

  - `1999@flex.cd-adapco.com` is the default CD-adapco Power-on-Demand server.
  - `<pod_key>` is the Power-on-Demand license key.

  For more information on Power-on-Demand licensing, see the section **User Guide** > **Using STAR-CCM+** > **STAR-CCM+ Licensing** > **Power-on-Demand Licensing**.

  A full list of command-line arguments is available in the STAR-CCM+ User Guide at **User Guide** > **Using STAR-CCM+** > **Command-Line Reference**.

The *Command* field updates automatically to reflect the selections you make. Manual entries are not affected by these selections.

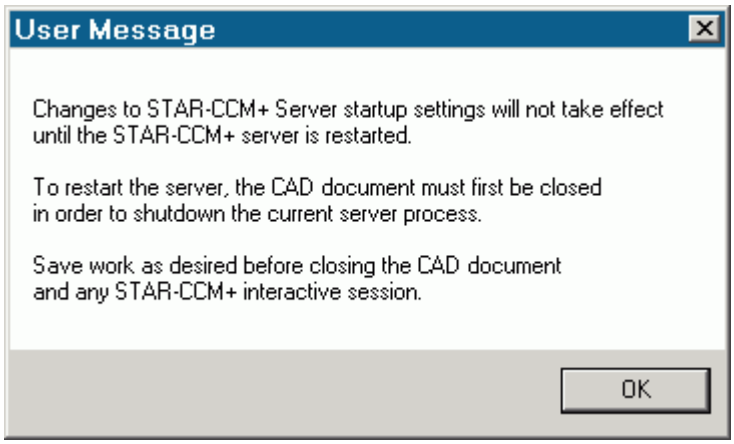
3. Select a licensing option:
   - To use a Power Session license, select **Power Session**.

     This license is design for parallel simulations, where you can use an unlimited number of cores in this session.
   - To use a STAR-CCM+ Lite license, select **Lite Session**.

     A limited set of STAR-CCM+ features are provided when you activate this option.

   If you select no licensing option, the default CAD client license is used.
4. Click **OK**.

   If the STAR-CCM+ server is running when you modify the preferences, a message box appears informing you that both STAR-Works and STAR-CCM+ must be closed for the changes to take effect.



5. Click **OK** to proceed.

6. Close the active STAR-Inventor and STAR-CCM+ sessions.
7. Restart STAR-Inventor.

Note: The STAR-CCM+ server only starts when it is required, such as during surface meshing or volume meshing. It does not start when you launch STAR-Inventor.

If an option in the *Preferences* dialog conflicts with an option in the `config.xml` file, the option that is set in the *Preferences* dialog takes precedence. An example of such a situation is if the *Compute Processes* property is set to `4` in the *Preferences* panel, while it is set to `2` in the `config.xml` file; the simulation runs using four processors. See Modifying the config.xml File

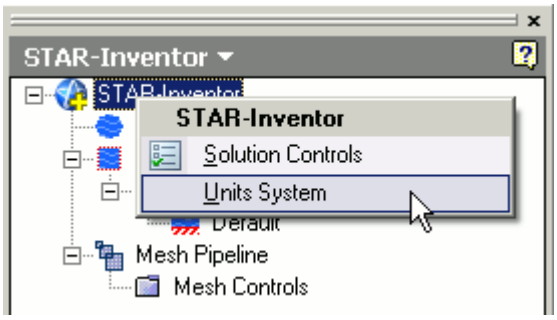
Note: IBM Platform MPI must be installed on your local machine for you to run in parallel. This program is included with the STAR-CCM+ package installer.

Changing the Units System

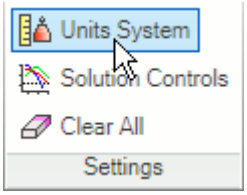
You can use either the SI (International Standard) or USCS (United States Customary System) units system in STAR-Inventor.

To change the units system:

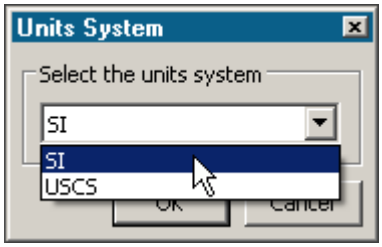
1. Bring up the *Units System* dialog:
   - Right-click the STAR-Works node within the object tree and select **Units System**.



   - Alternatively, from the ribbon, select **Units System**.



   The *Units System* dialog appears.
2. Select the desired units system from the drop-down menu.



3. Click **OK**.

Automatic Unit Conversion in Dialog Panels

Changing the default units system converts existing values to the new units system. Ensure that you select the desired units system before you enter the value, otherwise you will need to re-enter it.

Note: Specific values that you have already modified the units on are not affected by a change of units system.

Transferring Units to STAR-CCM+

Units that are used in the CAD geometry file are also used in the STAR-CCM+ simulation. Less common units, such as those listed below, are not supported in STAR-CCM+:

- Micrometers
- Angstroms
- Nanometers
- Mils
- Microinches
- Footinches (this unit may be converted to inches in STAR-CCM+)
- Miles

Note: This list may not be exhaustive and is subject to change.

These units are converted to meters when the CAD geometry is imported into STAR-CCM+.

Saving Files in Inventor

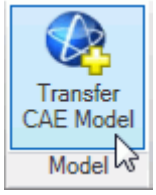
Using the **Save As** option to save a geometry file in Inventor results in any mesh displayed on the geometry model, and information in the output window, to be cleared. This occurs due to Inventor closing the old document file after saving the new document. All saved STAR-Inventor settings, such as boundary settings, are not affected by this operation and are retained. The mesh data is not lost. To display the mesh on the geometry model again, click on the **Generate Volume Mesh** button in the ribbon.

Transferring the CAE Model to STAR-CCM+

You can transfer your geometry and the simulation setup to STAR-CCM+ at any time from within the STAR-Inventor environment. If you want to transfer only the geometry to STAR-CCM+, refer to Transferring Geometry to STAR-CCM+.

To transfer the geometry with the simulation set up:

1. From the STAR-Inventor ribbon, click **Transfer CAE Model**.



The version of STAR-CCM+ that is affiliated with STAR-Inventor launches.

Note: A surface mesh is required in order to visualize the geometry in STAR-CCM+. If you have not generated a mesh before clicking **Transfer CAE Model**, the geometry is tessellated with a medium density.

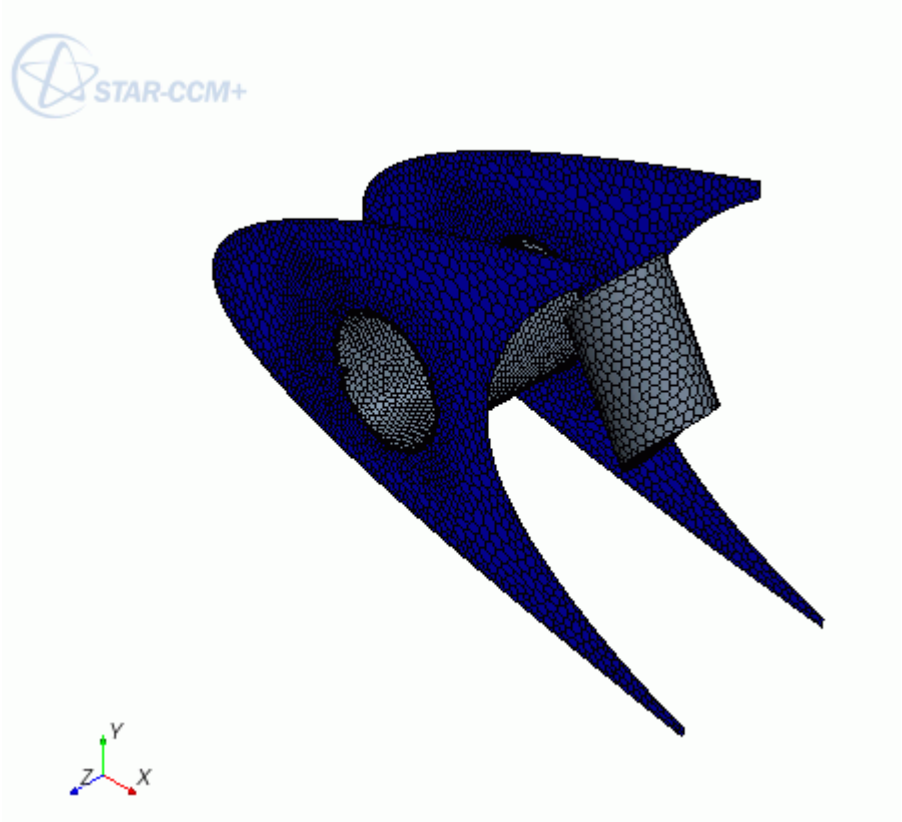
Editing STAR-Inventor Settings While the STAR-CCM+ Client is Running

If you make a change in STAR-Inventor while the STAR-CCM+ client is running, such as adding or deleting a boundary, you may need to update the volume mesh.

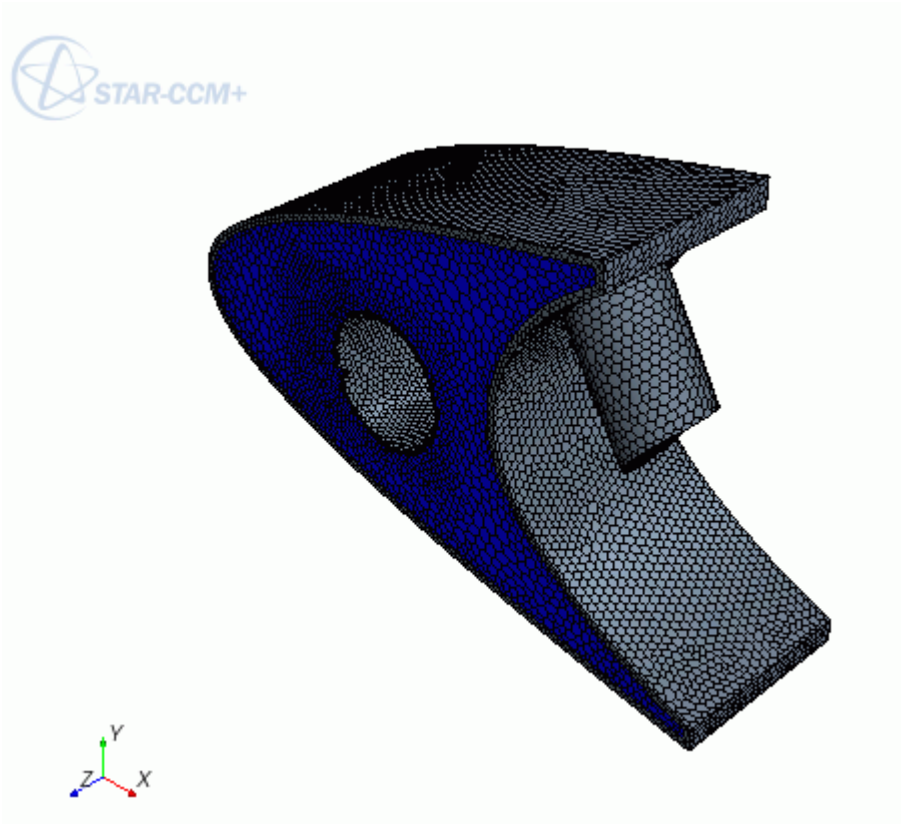
Scenes in the STAR-CCM+ session will not display the model correctly unless you keep the volume mesh up to date. A list of changes that require you to re-mesh the geometry is provided below:

- For boundaries and regions: changing the type, adding or deleting
- For mesh pipelines and mesh controls: editing, adding or deleting

When adding or removing regions and boundaries in STAR-Inventor, the scenes in STAR-CCM+ may be missing some parts, as shown below, even after re-meshing the geometry.



To update the scene, you need to manually add the modified regions and boundaries to the scene in the STAR-CCM+ client. You can do this by dragging and dropping the region or part from the STAR-CCM+ object tree, or by adding the region or part using the Part Selector. An alternative method would be to create a new scene, as this will automatically pick up the new regions and boundaries added to the simulation.



Common Functions

The following section details common operations used within the STAR-Inventor environment.

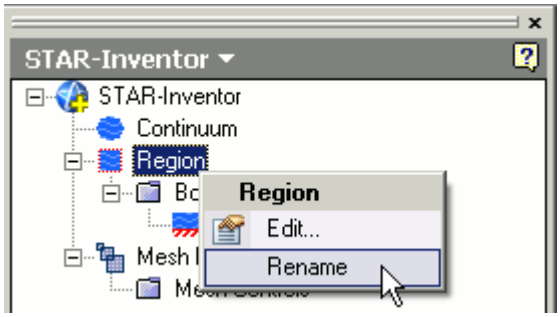
Contents:

Renaming an Object
Deleting an Object
Editing An Object
Assigning Surfaces and Parts to Objects in STAR-Inventor
Removing Surfaces and Parts from Objects in STAR-Inventor

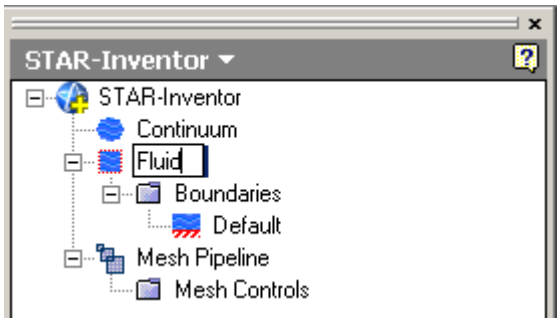
Renaming an Object

To rename an object:

1. Right-click the object node within the object tree and select **Rename**.



2. Enter the name.



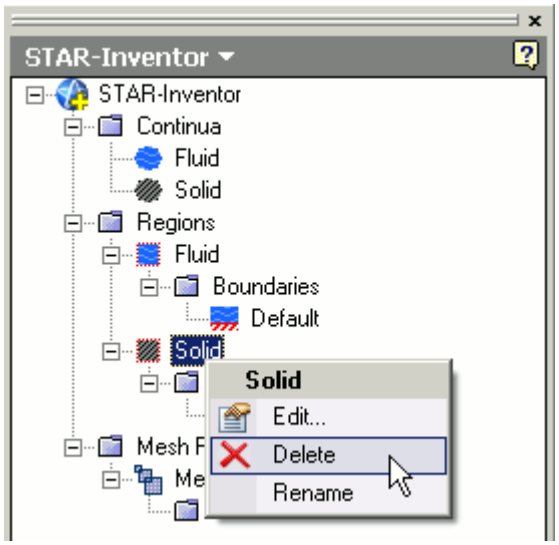
3. Click on a blank space in the object tree.

Note: You can also assign entity names at the point where you create them, for example, when adding boundaries, continua, regions, mesh pipelines and mesh controls. When adding a new object, locate the **Name** text box in the settings dialog and enter the required name.

Deleting an Object

To delete an object:

1. Right-click the object node within the object tree and select **Delete**.



To delete multiple entities:

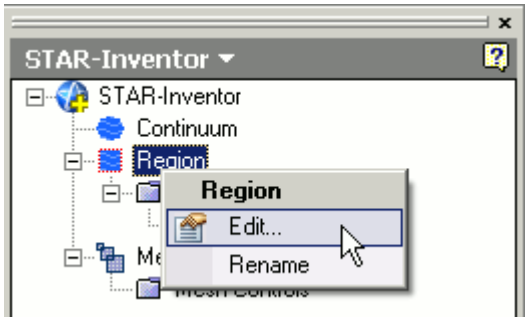
2. Select the desired nodes within the object tree (by holding down the `<Ctrl>` key), and right-click on any one of the selected nodes, then select **Delete** from the pop-up menu.

Note: Some combinations of selections cannot be deleted together using this method. In these cases, no pop-up menu appears.

Editing An Object

To edit an object:

1. Right-click on the desired node in the object tree and select **Edit...**.



   The settings dialog for the object appears.

2. Make the changes that you want.
3. Click **OK**.

Assigning Surfaces and Parts to Objects in STAR-Inventor

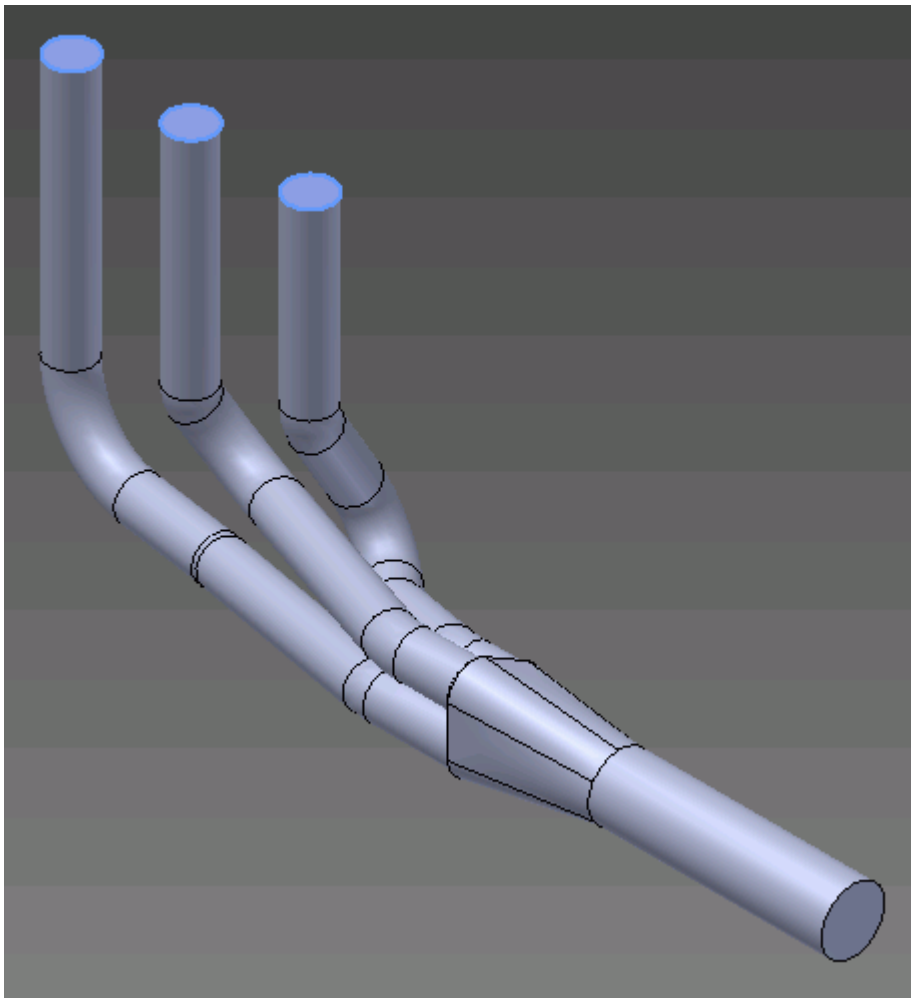
Regions, boundaries, mesh pipelines, and mesh controls are all represented as objects on the STAR-Inventor object tree. You can assign parts and surfaces to these objects when their property dialog is open.

Assigning Surfaces to Objects

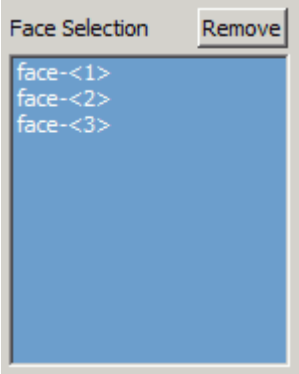
To assign a surface to an object (when its property dialog is open):

1. On the CAD geometry, click the surfaces that you want to assign to the object. You can add multiple surfaces by clicking more surfaces on the CAD geometry.

   The selected surfaces are highlighted on the CAD geometry.



   The selected surfaces are also displayed in the *Face Selection* box of the active dialog.



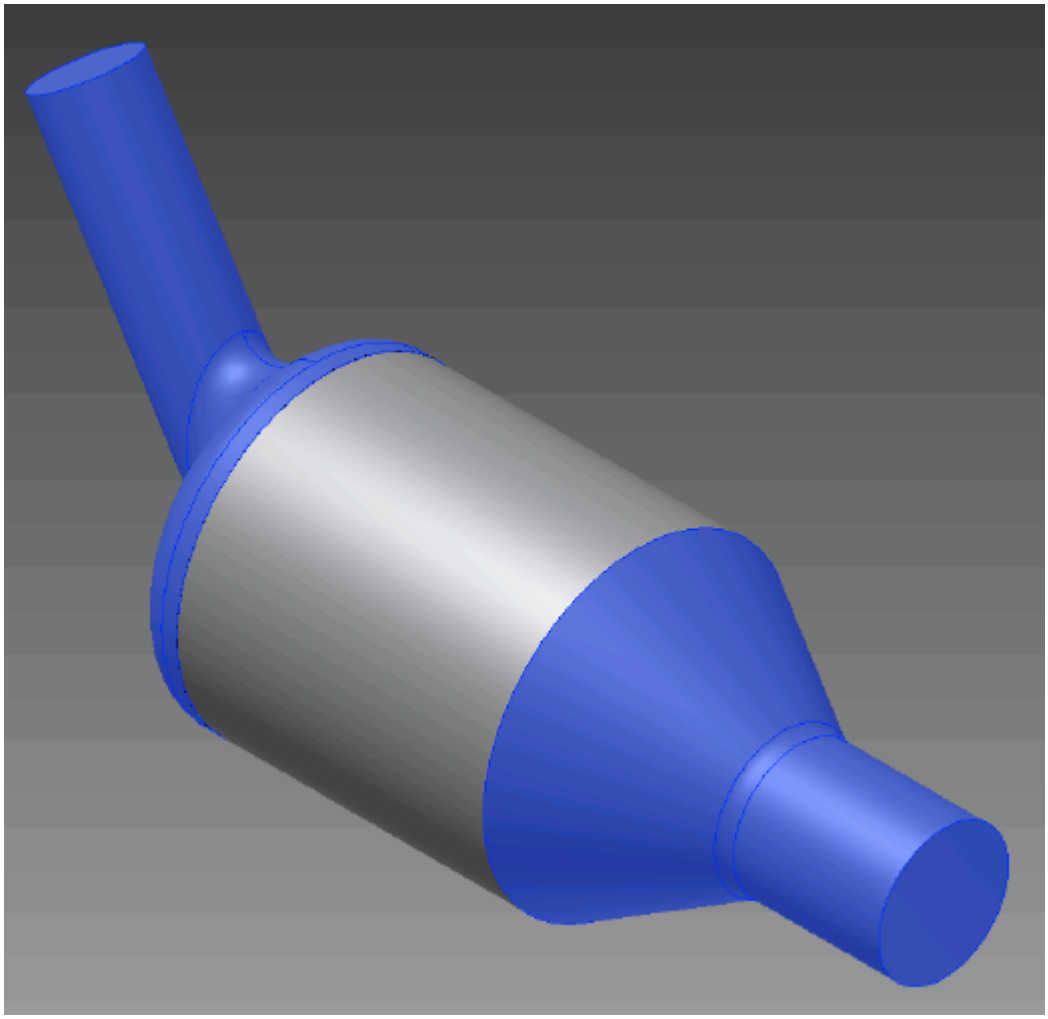
Parts that are assigned to other regions are automatically hidden on the CAD geometry.

Assigning Parts to Objects

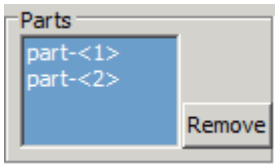
To assign a part to an object (when its property dialog is open):

2. On the CAD geometry, click the part that you want to assign to the object. You can add multiple parts by clicking more parts on the CAD geometry.

   The selected parts are highlighted on the CAD geometry.



The selected parts are displayed in the *Part Selection* box of the active dialog.



Parts that are assigned to other regions are automatically hidden on the CAD geometry.

Removing Surfaces and Parts from Objects in STAR-Inventor

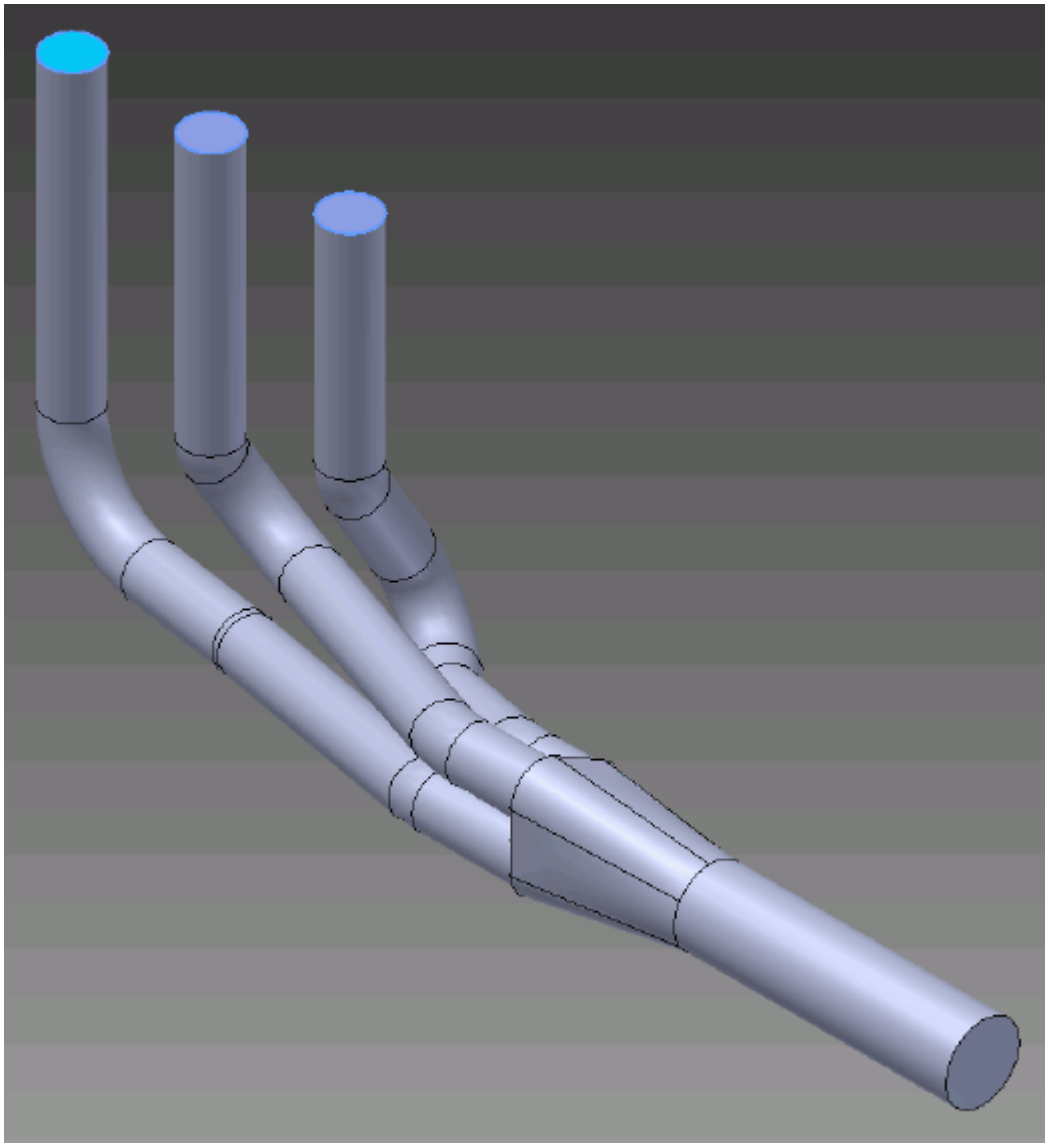
Regions, boundaries, mesh pipelines, and mesh controls are all represented as objects on the STAR-Inventor object tree. You can remove parts and surfaces from these objects when their property dialog is open.

A surface that is removed from a boundary is returned to the default boundary. Similarly, removing a part from a region returns it to the default region.

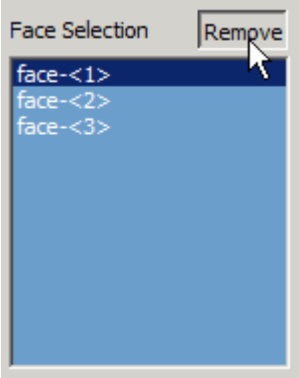
Removing Surfaces from Objects

To remove a surface from an object (when its property dialog is open):

1. In the *Face Selection* box, select the surface you want to delete.
   The selected surface on the CAD geometry is highlighted in light blue.



2. Click **Remove**.



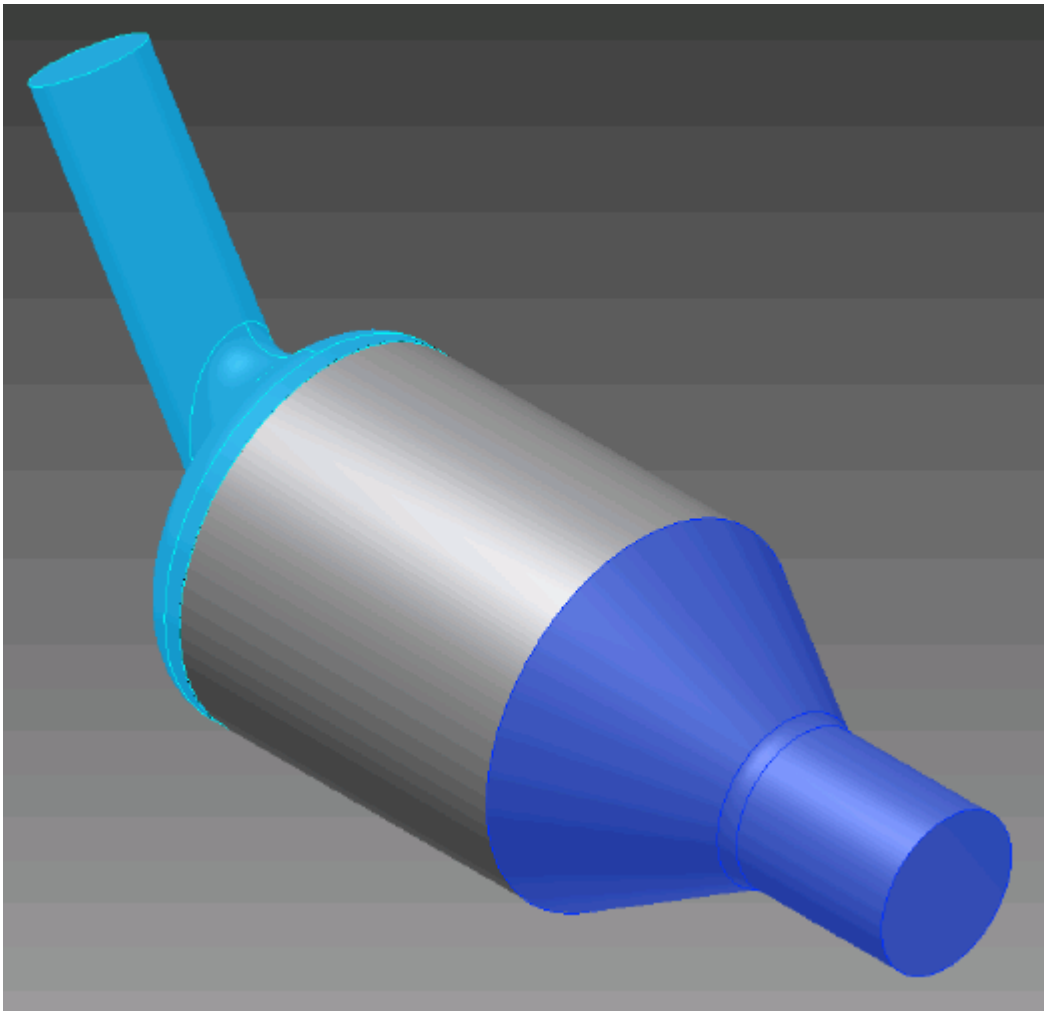
The surface is removed from the *Face Selection* box. You can select multiple surfaces for deletion by holding down the <`Ctrl`> key, and remove them in one operation.

Removing Parts from Objects

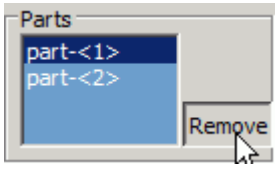
To remove a part from an object (when its property dialog is open):

3. In the *Part Selection* box, select the part you want to remove.

The selected part on the CAD geometry is highlighted in light blue.



4. Click **Remove**.



The part is removed from the *Part Selection* box. You can select multiple parts for deletion by holding down the <`Ctrl`> key, and remove them in one operation.

Running STAR-Inventor and STAR-CCM+ on Different Machines

You can run STAR-Inventor and STAR-CCM+ on two different machines.

STAR-CCM+ operates on a client-server architecture, where STAR-CCM+ is the server and STAR-Inventor is the client. A benefit of client-server architectures is that the client and the server do not have to be running on

the same machine. Typically, CAD packages run on Windows operating systems. You can choose to run STAR-CCM+ on any remote machine with a supported or certified platform, including Linux platforms.

Contents:

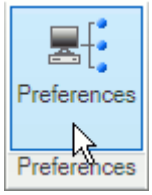
Setting Up STAR-CCM+ to Run on a Remote Machine with STAR-Inventor on the Local Machine

Setting Up STAR-CCM+ to Run on a Remote Machine with STAR-Inventor on the Local Machine

STAR-Inventor and STAR-CCM+ can run on two different machines.

To set up STAR-CCM+ to run on a remote machine:

1. From the STAR-Inventor ribbon, click **Preferences**.



2. In the *Preferences* dialog, set *Server Host Type* to **Remote**.
   The options for the remote server appear.
3. Set the remaining server settings to reflect your situation. See Server Settings Panel Reference.
4. If you want to store these settings for the next time you launch STAR-Inventor, click **Save as Default Settings**.
5. Click **OK**.

Contents:

Server Settings Panel Reference

Server Settings Panel Reference

The Server Settings panel that is located in the Preferences dialog allow you to run STAR-Inventor on a remote machine.

Property	Description	
Server Host Type	Specifies whether to run STAR-CCM+ on the local machine or on a remote machine.	
	Local	STAR-CCM+ runs on the local machine.
	Remote	STAR-CCM+ runs on a remote machine
SSH Command	Specifies the absolute path to the SSH installation directory.	

Property	Description
Server Host	Specifies the name of the remote host where STAR-CCM+ is installed. This option can contain the IP address, name, or full name of the host.
Sim File Directory	Specifies the absolute path to the directory where you want to store the simulation files on the remote machine. When in **Local** mode, the simulation files are saved in the same directory as the geometry file. You can specify a custom directory.
Server Executable	Specifies the absolute path to the STAR-CCM+ executable on the remote machine.

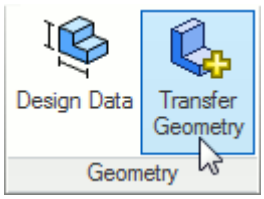
Transferring Geometry to STAR-CCM+

With STAR-Inventor, you can transfer your CAD geometry directly to STAR-CCM+ from within the Inventor environment.

The Transfer Geometry feature imports your geometry into STAR-CCM+. This function provides an alternative method to the CAD translator add-ons for importing geometry. This method of transferring geometry can provide a faster workflow when performing a design analysis.

To transfer geometry to STAR-CCM+:

1. From the STAR-Inventor ribbon, click **Transfer Geometry**.



   STAR-CCM+ launches and imports the CAD geometry. You are now at the same stage in the workflow as if you imported the surface mesh directly into STAR-CCM+ using the CAD translators (**File** > **Import** > **Import Surface Mesh...**).

2. Set up your case in STAR-CCM+.

Caution: Do not click **Transfer CAE Model** if you have previously used the Transfer Geometry method and set up your case in STAR-CCM+. This action results in the loss of your settings in STAR-CCM+.

Contents:

Saving Simulation Files
Transferring Design Data to STAR-CCM+
Modifying Geometry from Within STAR-CCM+
Controlling How the Geometry is Transferred to STAR-CCM+
Geometry Transfer Settings Dialog

Saving Simulation Files

After transferring the geometry to STAR-CCM+, you can save the simulation file and retain the associativity with STAR-Inventor.

To save your simulation files and keep the associativity:

- Save your CAD geometry in STAR-Inventor.

  Note: You can also safely rename and move the file using this method.

or

- Save your simulation file in STAR-CCM+.

Note: Do not rename the simulation file when using this method, as it will lose the associativity with the Inventor file.

Resuming a Simulation

To resume a simulation after closing STAR-Inventor and STAR-CCM+:

1. Launch STAR-Inventor.
2. Open the CAD geometry in STAR-Inventor.
3. Click **Transfer Geometry**.

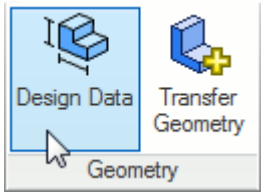
Transferring Design Data to STAR-CCM+

Design data consists of design parameters, coordinate systems, and reference planes that you can transfer to STAR-CCM+ along with the geometry.

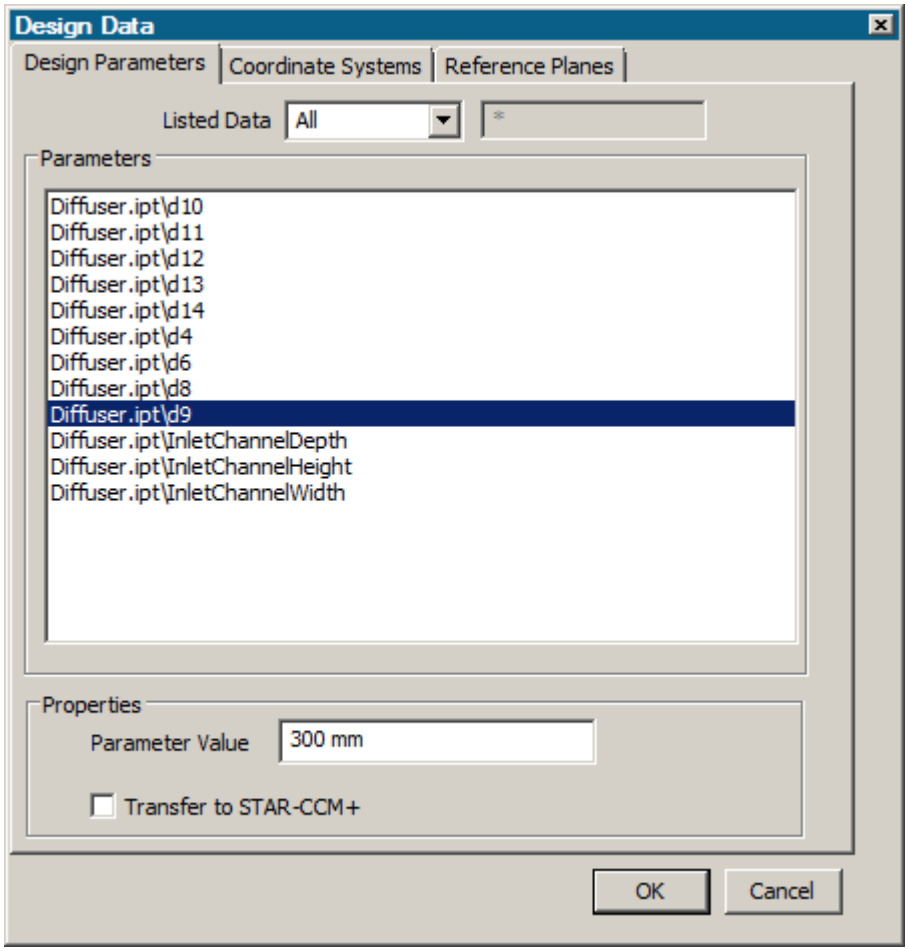
Design parameters control the feature dimensions, for example extrusion depth, and are useful for modifying the geometry in STAR-CCM+ after running the analysis. You can use reference planes in scalar scenes and reports in STAR-CCM+. You can use coordinate systems to define motion in the STAR-CCM+ simulation.

To transfer design data to STAR-CCM+:

1. From the STAR-Inventor ribbon, click **Design Data**.



The *Design Data* dialog appears.



2. Select the design data to transfer:
   - **Design Parameters**—Click the *Design Parameters* tab and select the design parameters that you want to transfer from the list. You can select multiple items at once.

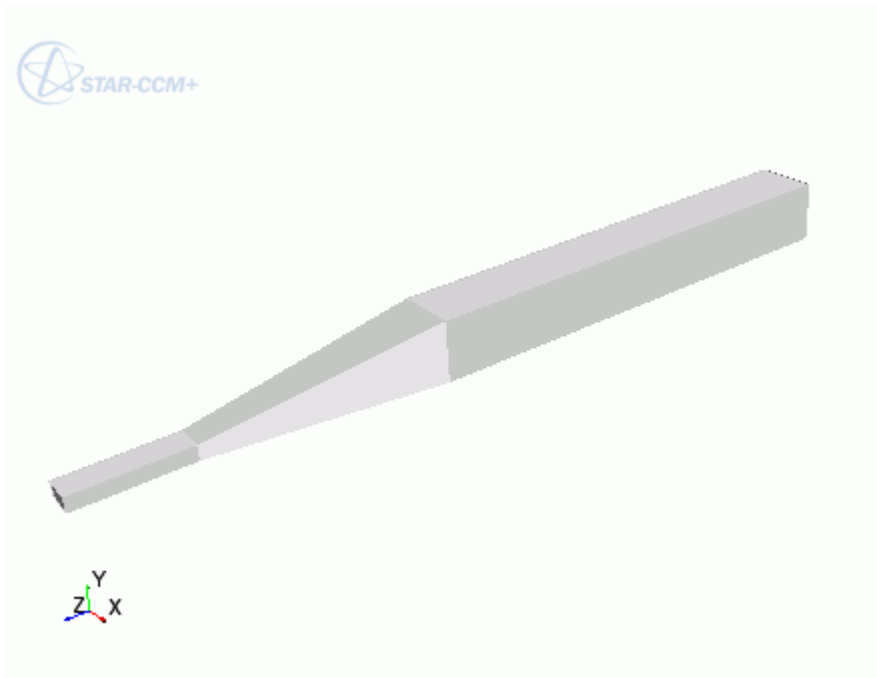
     Note: STAR-Inventor only recognizes native CAD features as design parameters. Only recognized design parameters are displayed in the list.
   - **Coordinate Systems**—Click the *Coordinate Systems* tab and select the coordinate systems that you want to transfer from the list. You can select multiple items at once.
   - **Reference Planes**—Click the *Reference Planes* tab and select the planes that you want to transfer from the list. You can select multiple items at once.
3. In each tab, activate the *Transfer to STAR-CCM+* option.
4. Click **OK**.
5. Transfer the geometry to STAR-CCM+. See Transferring Geometry to STAR-CCM+.

The geometry, along with the specified design data, is transferred to STAR-CCM+. Transferred design parameters are placed in the **Geometry** > **CAD Client Models** > **[geometry_name]** > **Design Parameters** node. Transferred planes are placed in the **Derived Parts** node in STAR-CCM+. Transferred coordinates systems are placed in the **Tools** > **Coordinate Systems** > **Laboratory** > **Local Coordinate Systems** node in STAR-CCM+.

Modifying Geometry from Within STAR-CCM+

Once you transfer a design parameter to STAR-CCM+, you can modify the geometry from within STAR-CCM+ with respect to that parameter. This feature is useful for quickly modifying a geometry after performing an analysis in STAR-CCM+ without needing to return to the original CAD software.

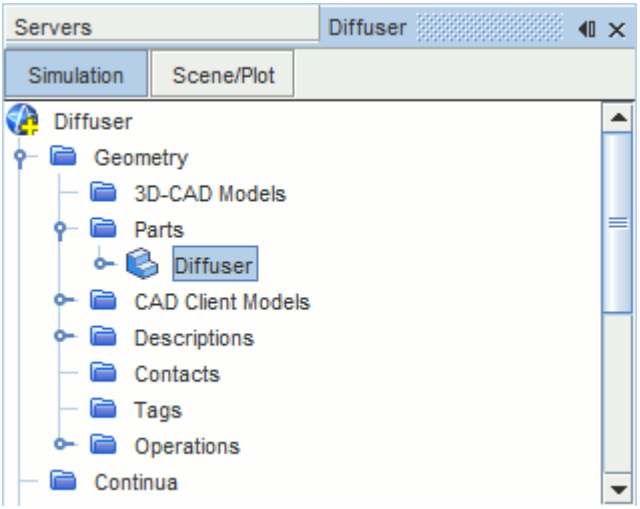
In the following example, the geometry of a diffuser is modified.



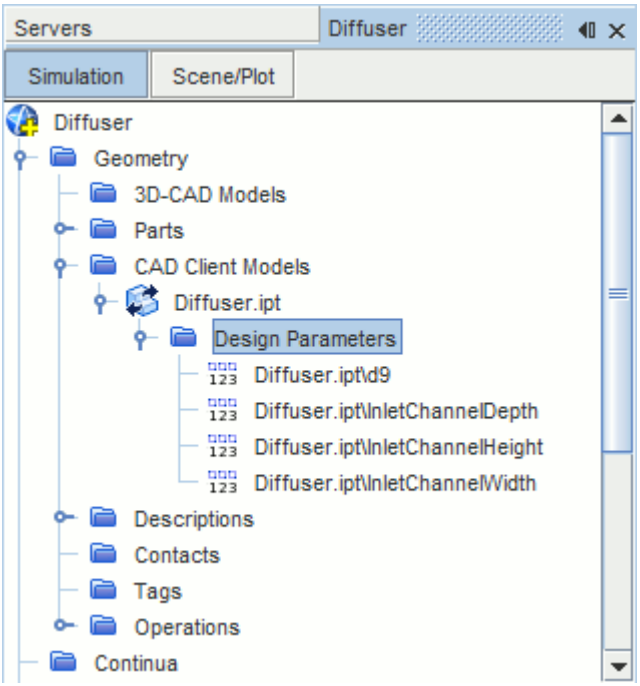
To modify a geometry in STAR-CCM+:

1. In STAR-Inventor, select the design parameters that you want to transfer to STAR-CCM+.
2. Transfer the geometry to STAR-CCM+. See Transferring Geometry to STAR-CCM+.

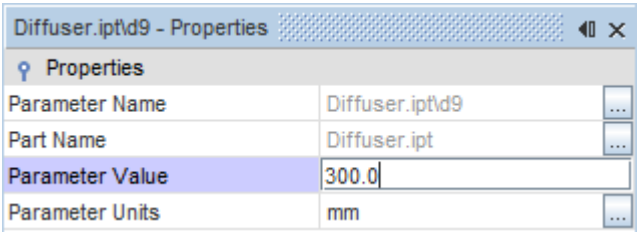
   Transferred geometries are located under the **Geometry** > **Parts** node.



Transferred design parameters are located under the **Geometry** > **CAD Client Models** > **[part_name]** > **Design Parameters** node.

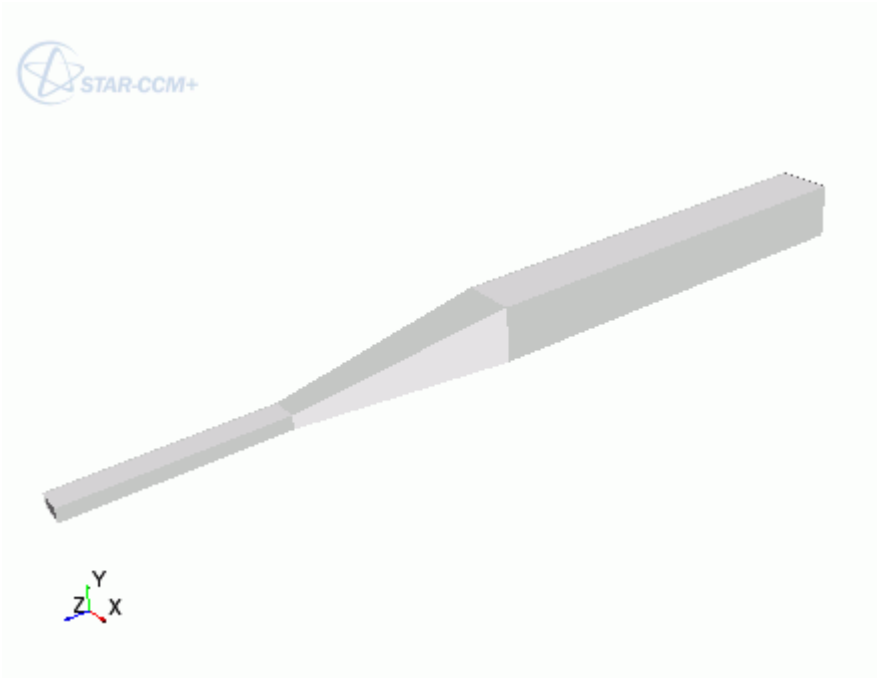


3. In STAR-Inventor, save the geometry file. The simulation file is also saved.
4. In STAR-CCM+, select the design parameter that you want to modify and set *Parameter Value* to the desire value.



5. Update the part in STAR-CCM+:
   - To update the part and save the updated parameters in the CAD file, right-click **Geometry** > **CAD Client Models** > **[part_name]** and select **Update Model and Save the CAD File**.
   - To update the part without updating the parameters in the CAD file, right-click **Geometry** > **CAD Client Models** > **[part_name]** and select **Update Model**.

   The geometry updates according to the design parameter changes.



6. Return to Inventor. The changes to the geometry are transferred back to the native CAD geometry.

This workflow maintains an active link between STAR-Inventor and STAR-CCM+. When working with Geometry Transfer, you can modify the geometry in either package and the geometry in the relevant package is updated accordingly.

Updating the Geometry in STAR-CCM+ with Inventor Closed

You can close Inventor and continue working with the simulation in STAR-CCM+. Doing so, however, breaks the link between the two software packages. Any further updates that you make to the geometry in STAR-CCM+ are not transferred back to the geometry in Inventor. Instead, these changes are stored in memory, which allows you to include the design parameters in your optimization scripts.

This workflow also allows you to release the Inventor and STAR-Inventor licenses when performing tasks in STAR-CCM+. In this case, the Inventor and STAR-Inventor licenses are temporarily checked out only when the updates to the geometry are stored in memory.

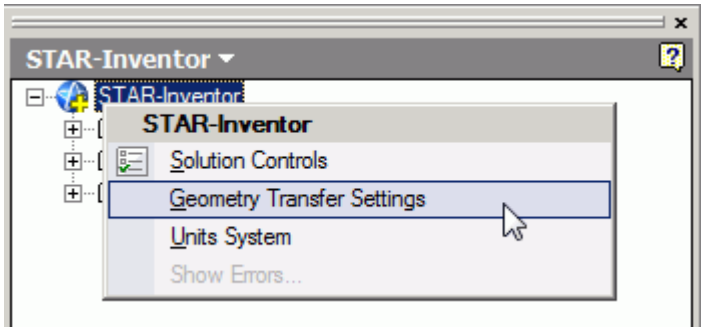
Transferring the geometry from STAR-Inventor will overwrite the geometry in STAR-CCM+.

Controlling How the Geometry is Transferred to STAR-CCM+

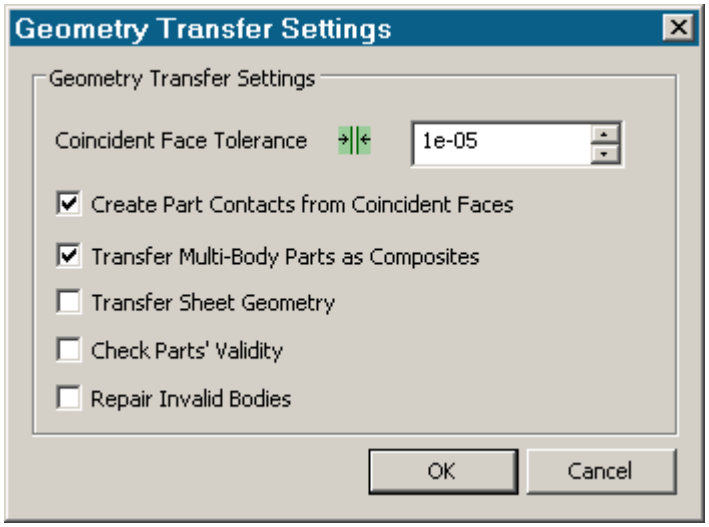
You can control how the geometry is transferred from Inventor to STAR-CCM+ through the *Geometry Transfer Settings* dialog.

To access the *Geometry Transfer Settings* dialog:

1. Right-click the STAR-Inventor root node and select **Geometry Transfer Settings**.



The *Geometry Transfer Settings* dialog appears.



2. Set the transfer parameters. See Geometry Transfer Settings Dialog.
3. Click **OK**.

Geometry Transfer Settings Dialog

Use the following properties to control how the geometry is transferred to STAR-CCM+.

Coincident Face Tolerance	Specifies the tolerance, in meters, for geometry in adjacent faces to be considered coincident. The default value is `1.0E-3`.
Create Part Contacts from Coincident Faces	Controls whether the geometry transfer feature searches for pairs of coincident faces on the CAD faces. For each pair found, part contact data is created and stored in the *Contacts* property (in STAR-CCM+). See the Creating Part Contacts from Coincident Entities section in the STAR-CCM+ user guide.

Transfer Multi-Body Parts as Composites	Specifies whether to transfer a single part containing multiple solid bodies as a composite part. Part contacts are created for each pair of coincident faces. Leaving this option deactivated merges all solid bodies in the part.
Transfer Sheet Geometry	Specifies whether to transfer sheet bodies as part of the transfer process. This option is turned off by default.
Check Parts' Validity	Provides you with information on whether the bodies are geometrically and topologically valid, from the Parasolid geometry kernel point of view. A simple message is displayed in the output window which states if the surface mesh is valid. For example: `Body 2 is not valid`.
Repair Invalid Bodies	Attempts to repair invalid bodies. The number of invalid bodies that are successfully repaired is displayed in the output window.

Understanding the STAR-Inventor Process

In STAR-Inventor the typical workflow for running a simulation is as follows:

1. Ensure that the CAD model is suitable for analysis
2. Extract the fluid volume
3. Define the material properties for each part of the model, for example, air, water, aluminum
4. Define model physics, for example, turbulence, temperature, gravity
5. Specify boundary conditions, for example, inlets, outlets, symmetry planes
6. Specify mesh settings
7. Define solution controls
8. Run the case
9. Post-process the results in STAR-CCM+

Once you have obtained an initial solution you may wish to re-run the case with some different settings, or with some changes to the geometry. To do this you can:

1. Return to STAR-Inventor and change settings or modify the geometry
2. Re-run the case

Using Suitable CAD Models in STAR-Inventor

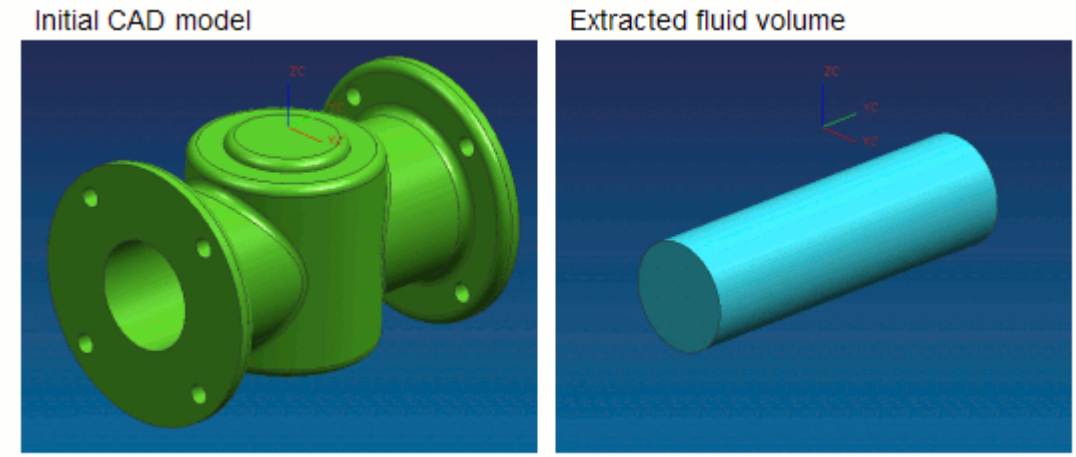
STAR-Inventor only supports the use of solid bodies. If a geometry is made up entirely of surfaces then it will not be possible to define any boundary conditions, and the meshing options will be greyed out.

It is good practice to create copies of the original geometry file(s) before importing them into STAR-Inventor.

Suppressed or deactivated parts and features are not supported by STAR-Inventor and will have no effect on the analysis. Hidden parts, however, are used in the analysis. Ensure that all desired parts and features are active before using them with STAR-Inventor.

Extracting the Fluid Volume

A CAD model usually represents a component made from solid material. However, in the case of flow simulation, a model representing the fluid volume is needed, requiring a different CAD model to be created. This is illustrated below:



In some instances, a simulation will require multiple regions, such as:

- Moving Reference Frames

  For rotating flow, such as turbines.
- Porous Media

  For representing regions with fine geometric structures
- Energy Source

  For simulating thermal specifications, such as conjugate heat transfer

These types of simulation require each region to be represented by individual contiguous parts within an assembly.

Defining Material Properties

Continua determines the characteristics of the fluid and solid entities used within the simulation. The behavior of a single continuum is primarily defined by the chosen material type and physics models.

Specification of the material type used in the simulation provides properties such as density, viscosity and conductivity, which are all used by the solver to calculate a solution. STAR-Inventor includes a database of common materials and allows you to modify material properties where necessary.

Defining Model Physics

Physics models determine the nature of the fluid, and can be described as either:

- Turbulent or Laminar
- Constant Density or Ideal Gas

Additional models can also be activated to meet the needs of various simulation requirements:

- Passive scalar
- Temperature
- Gravity

A region is defined as a volume domain in a simulation, surrounded by boundaries. Each region in the simulation must be represented by a part in the native CAD package. When multiple regions are required, multiple parts must be used within an assembly.

All regions must be assigned to a continuum, as they represent a volume of material in the simulation. Region settings can be used to define porous media, moving reference frames, and an energy source.

Specifying Boundary Conditions

To define boundary conditions you first need to select the surface or surfaces on the model to assign to a separate boundary.

A boundary is defined as a surface, which along with other boundaries, surround and characterize a region. Boundary conditions determine how the boundary will affect the material within a region, and consequently, the simulation as a whole.

Inlet boundaries determine how much fluid enters the region, using a velocity, mass flow rate or stagnation inlet. Outlet boundaries define the process for fluid leaving the region, which are defined with either a pressure or flow-split ratio. Wall boundaries contain the fluid, and can also be used to define a thermal specification that will influence the transfer of energy to or from the region. Symmetry boundaries can be used to avoid modeling the entire region where appropriate, and so reduce calculation time.

As boundary types can affect the way in which the volume mesh is constructed, it is important to set them before generating the mesh.

Specifying Mesh Settings

The STAR-CCM+ solver uses the finite volume method to obtain an approximate solution to the underlying equations of fluid flow. This requires a mesh to be created, which represents the fluid volume as a collection of much smaller volumes known as cells.

A surface mesh is first created based on the boundaries defining the regions in the simulation. Thereafter a volume mesh is created to discretize the entire model.

The accuracy of the results is directly affected by mesh quality. If it is necessary to improve the quality of the mesh, some of the ways in which this can be done are to:

- Increase the mesh density, that is, the number of cells
- Add further sub-surface prism layers to capture near wall flow
- Refine the mesh in areas where flow characteristics are expected to change rapidly

As volume mesh quality is largely governed by the surface mesh quality, it is common practice to create and inspect the surface mesh before creating a volume mesh.

Defining Solution Controls

Solution controls determine the level of accuracy the solver will work towards. As a high accuracy solution will take longer, it may be necessary to compromise on this in order to obtain a result in a shorter time.

STAR-Inventor includes five predefined solver settings ranging from fast (with low accuracy) to slow (with high accuracy). Advanced options are also available to provide further control.

Running the Case

When the simulation has been fully defined, the volume mesh is sent to STAR-CCM+ along with all the necessary data for running the simulation. The time to complete a simulation will vary depending on the mesh density and solution controls.

As the solver is running, residuals showing the rate at which individual equations are converging are shown in a plot. A scalar scene displaying absolute pressure is displayed on completion of the simulation.

Post-Processing the Results in STAR-CCM+

Post-processing of simulations is carried out using the STAR-CCM+ interface. This allows the results to be taken from the solver and displayed in a variety of scenes and plots.

Making Modifications to the Settings or Geometry

As STAR-Inventor is completely integrated within the Inventor environment, it is very simple to perform parametric studies that examine the response of the fluid to changes in geometry. Once the first analysis is complete, the boundary conditions, run settings and all other properties of the CFD model are stored within the CAD file. Any changes in the flow field due to design changes can be recalculated by generating a new mesh and running the solver.

Connecting to Remote Machines

STAR-CCM+ uses a client-server architecture, and as such the CAD client (STAR-Inventor) and the server (STAR-CCM+) are not required to run on the same machine.

Using an SSH (secure shell) connection, you can connect the client and server on two different machines, even when these are on different platforms. Due to how the CAD client and server interact in the background, the SSH connection must not ask for a username or a password.

There are two ways that you can connect STAR-Inventor and STAR-CCM+, depending on what your starting point is:

- You can start from the CAD client and launch the server on a remote machine. This method is useful for transferring CAD geometry from STAR-Inventor running on Windows to STAR-CCM+ running on Linux.
- You can start from the server and connect to the CAD client on a remote machine. This method is useful for connecting to STAR-Inventor to update the CAD geometry. With this method, you can perform an Optimate analysis on a cluster while the geometry is updated in STAR-Inventor.

Contents:

Running STAR-Inventor with Remote STAR-CCM+
Running STAR-CCM+ with Remote STAR-Inventor

Running STAR-Inventor with Remote STAR-CCM+

Starting from STAR-Inventor, you can choose to run the STAR-CCM+ server on a remote machine.

Using this method, you can set up the full CAE analysis in STAR-Inventor on your local machine, and then run the STAR-CCM+ server on the remote machine. You can also choose to transfer your geometry from the CAD client to the remote STAR-CCM+ server. This feature is useful if you want to import geometry from STAR-Inventor running on Windows to STAR-CCM+ running on Linux.

To configure STAR-Inventor to launch STAR-CCM+ on a remote machine:

1. Make sure that you have a working SSH connection between the two machines.

   For the SSH connection, you require an SSH server on the machine running STAR-CCM+ and an SSH client on the machine running STAR-Inventor. The SSH connection to the remote machine must not ask for a username or a password. To avoid possible issues with a connection that still requires a password, add `-o NumberOfPasswordPrompts=0` to the SSH command. Setting this argument to zero stops the connection prompting for a password when one is needed — something that could cause the process to hang. See SSH Command.
2. In the STAR-Inventor ribbon, click **Preferences**.
3. In the *Preferences* panel, set *Server Settings* to **Remote**.
4. Set the remaining settings for your connection to the remote STAR-CCM+ server. See Server Settings Panel.

   This information is used to build a command-line in the background when STAR-Inventor invokes STAR-CCM+.

5. If you are going to connect to the same remote host in future sessions, click **Save as Default Settings**.

The next time STAR-Inventor invokes STAR-CCM+, the STAR-CCM+ server launches on the specified remote machine. A STAR-CCM+ client is also launched on the local machine to allow you to see the active session on the remote machine.

Contents:

Server Settings Panel

Server Settings Panel

Specify the connection settings to the remote STAR-CCM+ server using the Server Settings pane.

Parameter	Description	
Server Host Type	Specifies whether to run the STAR-CCM+ server on the local host, or on a remote host.	
	Local	Runs the STAR-CCM+ server on the same host as STAR-Inventor.
	Remote	Runs the STAR-CCM+ server on the remote host that you specify.
SSH Command	Specifies the absolute path to the SSH program. If the path to the SSH program exists in your `Path` environment variable, you can leave this parameter as the default entry. You can include additional arguments in the SSH Command. The connection must not ask for a password or a username. To avoid possible issues with a connection that requires a password, add `-o NumberOfPasswordPrompts=0` to the SSH command. Setting this argument to zero stops the connection prompting for a password when one is needed, something that could cause the process to hang.	
Server Host	Specifies the remote host. You can enter the name, fully qualified domain name, IP address, or path to the host, depending on how your network is set up.	
Sim File Directory	Specifies the absolute path to the directory where you want to save the simulation file on the remote host. You must have read and write access to this directory. If a simulation file is present in this directory, STAR-CCM+ will use this file. If no file is found in this directory, a new one is created. An example directory on a Linux machine is shown below: `/users/linuxbox/james/documents/cadClientCases`	
Server Executable	Specifies the full path to the STAR-CCM+ executable on the remote host. An example path is shown below: `[STAR-CCM+_INSTALL_DIR]/star/bin/starccm+`	
Save as Default Settings	Saves the server settings that you specify in this panel as the default settings. Save these changes if you want to connect to the same remote server in future sessions.	

Running STAR-CCM+ with Remote STAR-Inventor

Starting from STAR-CCM+, you can connect to STAR-Inventor on a remote machine.

This feature is useful if you have previously transferred your geometry from STAR-Inventor on Windows to STAR-CCM+ on Linux, and you want to reconnect the two.

To connect STAR-CCM+ running on the local machine to STAR-Inventor running on a remote machine:

1. Make sure that you have a working SSH connection between the two machines.

   For the SSH connection, you require an SSH server on the machine running STAR-Inventor and an SSH client on the machine running STAR-CCM+. The SSH connection to the remote machine must not ask for a username or a password.
2. In STAR-CCM+, expand the **Geometry** > **CAD Client Models** > **[Model_Name]** and set *CAD-Client Update Mode* to **Remote**.

   The **Remote CAD Client Settings** node appears.
3. Set the remaining parameters for the CAD geometry. See CAD Geometry Settings.
4. Select the **Remote CAD Client Settings** node and set the connection setting for the remote CAD client. See Remote CAD Client Settings.
5. If you are connecting from a STAR-CCM+ server that is running on a cluster, activate the *Use Proxy Server* option and set up the proxy server. See Proxy Server Settings.

When you make changes to the imported CAD geometry in STAR-CCM+, the changes are passed back to STAR-Inventor on the remote machine and the geometry is transferred back into STAR-CCM+.

Contents:

Remote CAD Client Settings Reference

Remote CAD Client Settings Reference

Specify the connection settings to the remote CAD client in the Remote CAD Client Settings panel.

CAD Geometry Settings

Parameter	Description	
CAD File	Specifies the full name of the CAD geometry file, including the extension.	
CAD File Location	Specifies the location of the CAD geometry file on the remote machine. An example path is shown below: `C:\CAD Client Cases`	
CAD-Client Update Mode	Specifies whether STAR-Inventor is located on the local machine where STAR-CCM+ is installed, or whether STAR-Inventor is located on a remote machine.	
	Local	STAR-Inventor is located on the same local machine as STAR-CCM+.

Parameter	Description	
	Remote	STAR-Inventor is located on a remote machine.

Remote CAD Client Settings

Parameter	Description	
Remote Host	Specifies the remote host where STAR-Inventor is installed. You can enter the name, full name, IP address, or path to the host, depending on how your network is set up.	
Remote CAD-Client Install Location	Specifies the absolute path to the root directory of the CAD client installation on the remote host. An example path to the default installation directory for version 12.06.### is shown below: `C:\Program Files\CD-adapco\12.06.###\STAR-CAD12.06.###`	
SSH Command	Specifies the command to use to run the SSH program. If you require a proxy server to connect to a cluster, you must use the following SSH command: `ssh -A -t <Cluster_Head_Node> ssh -A`	
Use Proxy Server	Specifies whether you require a proxy server to connect to a cluster.	
	Activated	Uses a proxy server as part of the connection to a cluster.
	Deactivated	Does not use a proxy server.

Proxy Server Settings

Parameter	Description
Proxy Host	Specifies the host name of the private network of the cluster. Enter the name, fully qualified domain name, or IP address. An example is shown below: `clusterhead.cm.cluster`
Proxy Public Host	Specifies the host name of the public network of the cluster. Enter the name, fully qualified domain name, or IP address. An example is shown below: `clusterhead.london.company.com`

Defining Materials and Physics

Defining the physical properties of a STAR-Inventor model involves specifying the materials, flow characteristics, operating conditions, and physics models.

These definitions and settings are accomplished through the use of continua and regions, which are described in the following sections.

Contents:

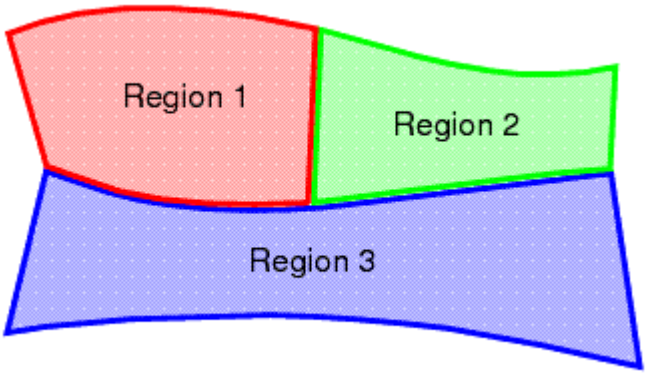
Working with Continua and Regions
Importing Geometry into STAR-CCM+ for Visualization

Working with Continua and Regions

Continua and regions are used together to capture the physical and material properties of a model.

A continuum represents a material in combination with a set of physics models that will be used to simulate the response of the material to the conditions imposed upon it. A continuum also provides the facility to define reference and initial values.

A region represents one or more volumes within the overall model, not necessarily contiguous, which will share the same continuum and set of boundary conditions. Regions are defined using CAD parts, and will be discretized by the volume mesher.



As well as holding a set of boundary conditions, regions can also be used to model specific physical behavior, including porous resistance, moving reference frames and energy sources.

Parts can be imported into STAR-CCM+ for visualization purposes without affecting the analysis.

Continua are assigned to single or multiple regions, which can contain single or multiple parts. In part mode, only one continuum and one region are available, as there is only one part. In assembly mode, new continua can be added to represent different materials present in the simulation, and new regions can be added to

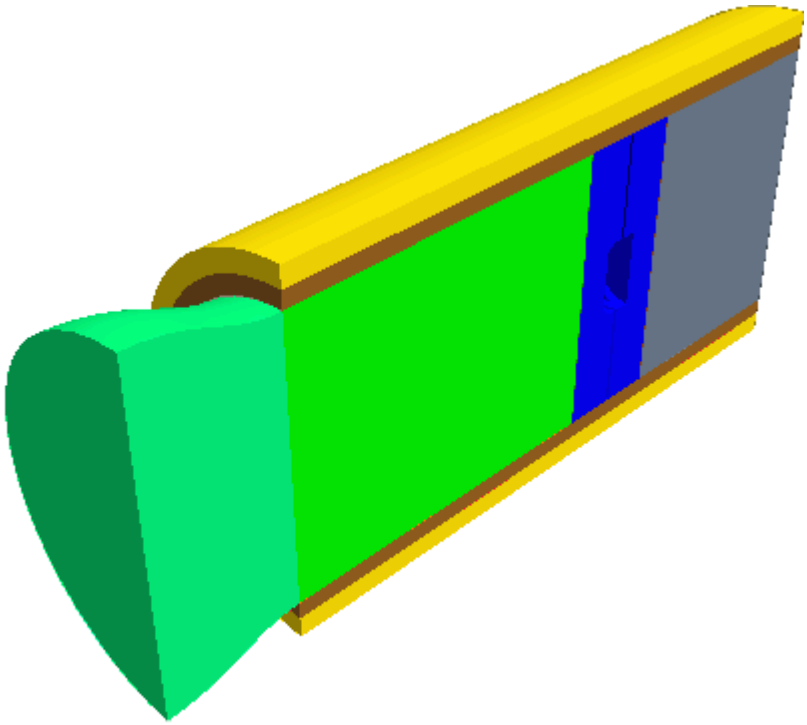
represent the parts composed of these materials. Continua and regions can be deleted if they are not required, although the simulation must always contain at least one of each.

There is a well defined relationship between continua, regions, boundary conditions and CAD parts.

- Every region must be associated with a continuum.
- Several regions can be associated with the same continuum.
- Every CAD part must belong to a region.
- Several CAD parts can be associated with the same region.
- Boundary conditions belong to regions, and cannot be shared between regions.
- Every CAD surface must have a boundary condition associated with it. A default wall boundary is applied to CAD surfaces that have not been specifically associated with a user-defined boundary condition.

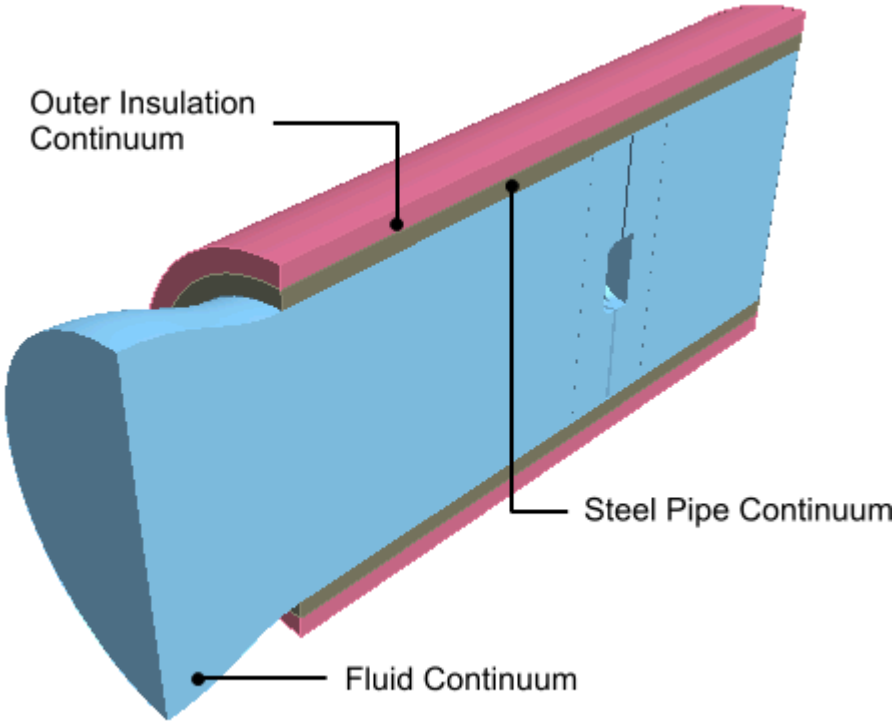
The following series of illustrations depict the relationship between CAD parts, continua, and regions, for a circular steel duct containing a rotating fan, and surrounded by a layer of insulation.

- Initially we have a collection of CAD parts in an assembly. These must be imprinted where appropriate to ensure that interfaces between parts can be created correctly.



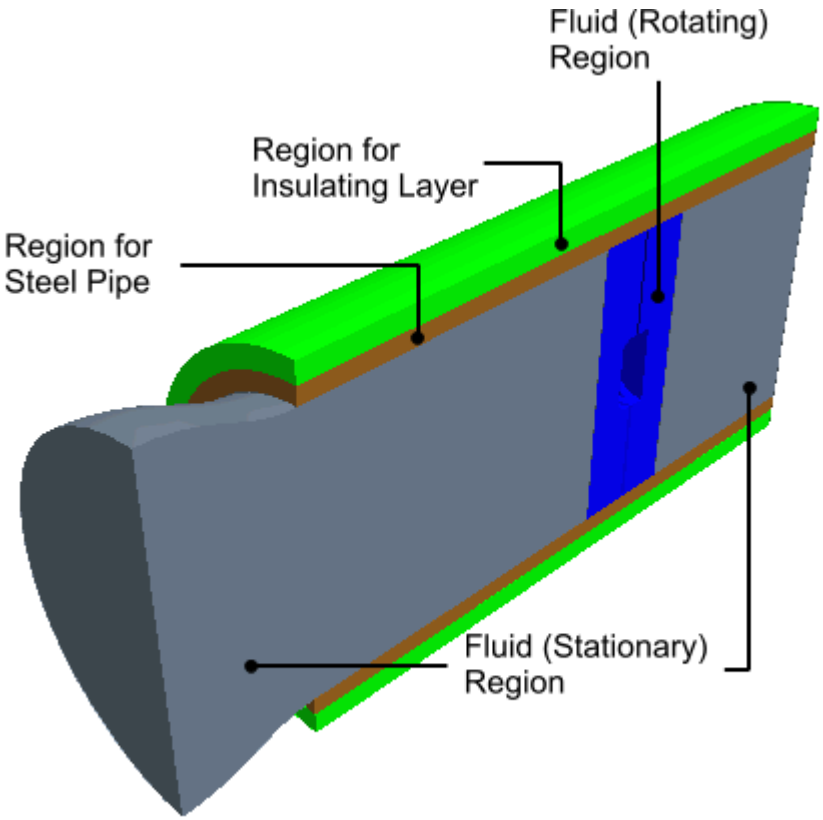
Assembly of CAD Parts (imprinted where necessary)

- Three continua are required to represent the three different materials present in the simulation: the fluid (for example, air), the steel pipe, and the insulation.



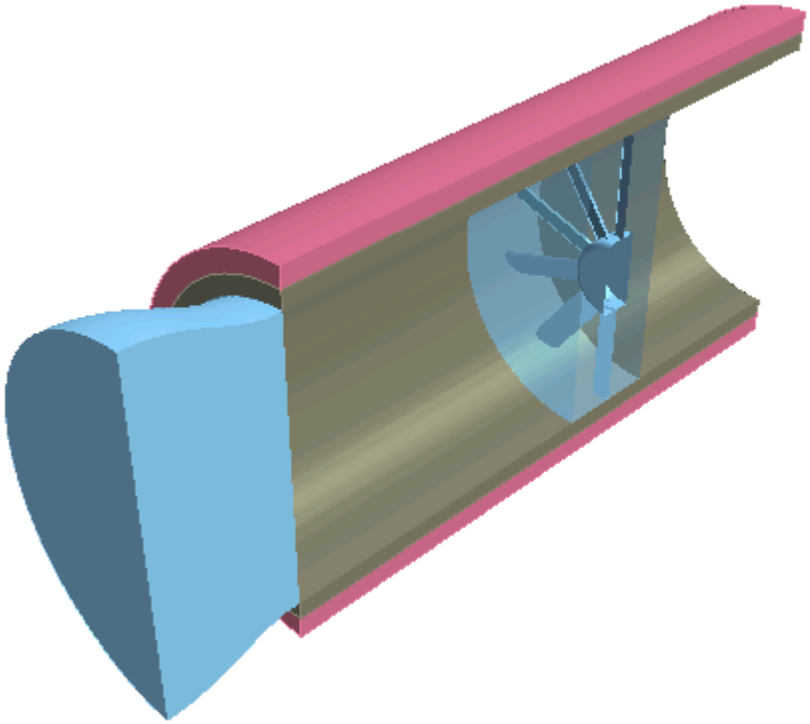
View of Continua Required for Simulation

- Two regions are required for the fluid portion of the simulation, as a rotating part is defined by applying a rotating frame of reference to a separate region surrounding the fan blades. Further regions are required for each of the solid parts (the steel pipe and insulating layer), which means a total of four separate regions overall.



View of Regions Required for Simulation

- A clearer picture can be given of the fan blade itself.



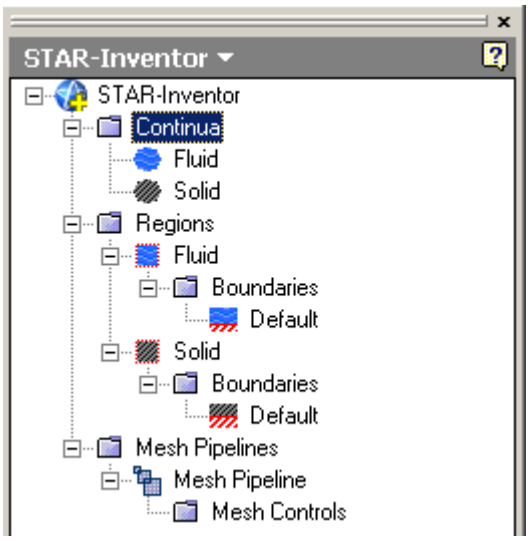
For an analysis containing a single part, for example a fluid-only case involving fluid flow through a converging-diverging pipe, the model will only require one continuum and one region in order to fully define the physical and material properties of the model. In this case, STAR-Inventor automatically assigns the default continuum to the default region, and no further assignments are necessary.

Contents:

Setting Up Continua
Setting Up Regions

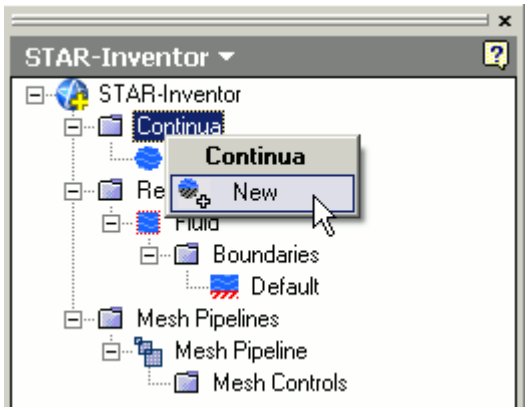
Setting Up Continua

To define different materials and associated physics for use in an assembly, you require additional continua. In each continuum you can specify a unique set of properties, such as a different material type or temperature. In assembly mode, continua are stored under a **Continua** node, as shown below.



To create an additional continuum:

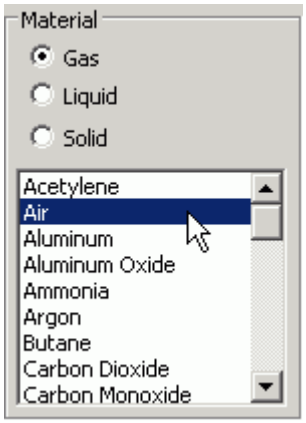
1. Right-click the **Continua** node in the object tree and select **New**.



The *Continuum Settings* dialog appears.

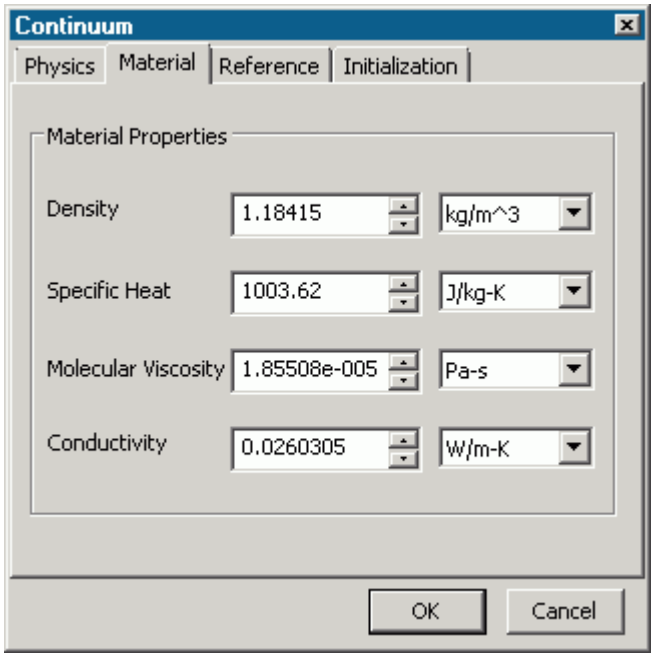
2. In the *Name* field, enter a name for the continuum.
3. In the *Materials* group, select the type of material that you want to define:
   - **Gas**
   - **Liquid**
   - **Solid**
4. Select a material from the list.

   You can modify the properties of any material that you select.



Note: You can select only one material per continuum. If your simulation contains a number of materials, you must create additional continua. This action is only possible for assemblies.

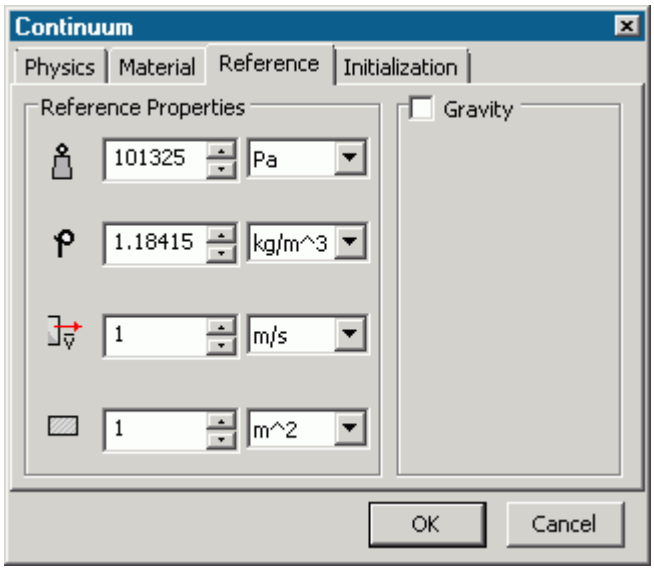
5. In the *Flow Characteristics* group, select the type of flow that you want to model: **Laminar** or **Turbulent**. This group only appears for fluids.
6. In the *Equation of State* group, select the equation of state : **Constant Density** or **Ideal Gas**. This group only appears for gases.
7. In the *Optional Physics Model* group, activate any optional physics models: **Passive Scalar** and **Temperature**.
8. Define the material properties in the *Material Properties* group.



See STAR-Inventor Material Properties Panel.

Note: You can change units of properties from the drop-down menu.

9. In the *Reference Properties* group, define the reference properties.

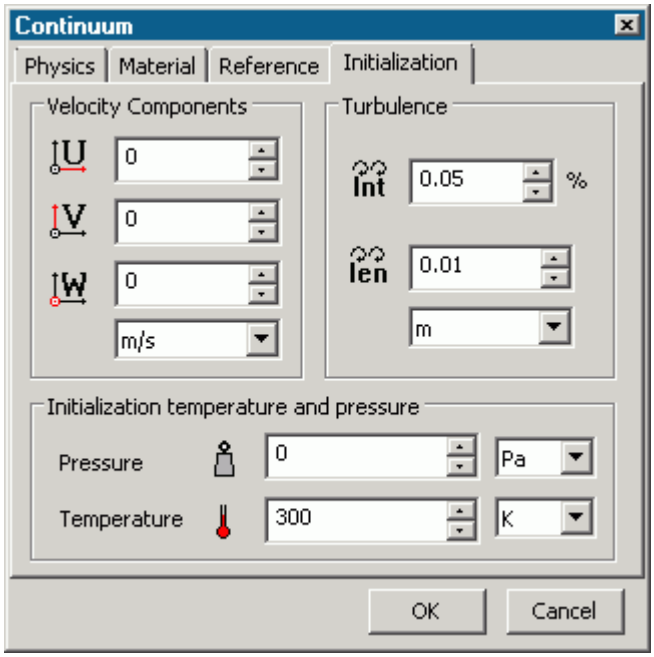


See STAR-Inventor Reference Properties Panel.

10. If you want to include the body force due to gravity when calculating the momentum equations in STAR-CCM+, activate the gravity model in the *Gravity* group box.

    See Activating the Gravity Model in STAR-Inventor.

11. Set the initial conditions in the *Initialization* group box.



See STAR-Inventor Initialization Panel.

Specifying initial conditions that are close to the expected solution can reduce the number of iterations required to reach a solution. You can set the initial conditions similar to the inlet conditions of the fluid domain.

Initial conditions are specified at a continuum level; if no initial conditions are specified, STAR-Inventor will use default values to initialize the solution.

12. Click **OK**.

    The continuum is added to the object tree.

Once you have created a continuum, you can assign it to a region.

Contents:

STAR-Inventor Continuum Properties
Activating the Gravity Model in STAR-Inventor

STAR-Inventor Continuum Properties

The continuum dialog has a number of properties that you can define. This section provides a description for each property.

STAR-Inventor Material Properties Panel

You can define the following material properties in STAR-Inventor:

Property	Description
Density	Available for all types of materials. For gases, this property is available when you use the Constant Density equation of state.
Specific Heat	Available for all material types when you activate the Temperature physics model.
Molecular Viscosity	Only available for fluids.
Conductivity	Available for all materials when you activate the temperature physics model.
Molecular Weight	Only applies to gases. This property is available when you use the Ideal Gas equation of state.

STAR-Inventor Reference Properties Panel

You can define the following continuum reference properties:

Property	Description
(*Reference Pressure*)	The pressure values that are defined in the simulation are all relative to the reference pressure. For more information, see the What is the Reference Pressure? section.
(*Reference Temperature*)	The reference temperature is used to define the operating temperature of the fluid in the buoyancy source term of the Boussinesq model. This parameter is used by STAR-CCM+ when the temperature, constant density, and gravity physics models are enabled.
(*Reference Density*)	These quantities are not used as part of the CFD calculation, but is necessary to calculate the force coefficients when post-processing.
(*Reference Velocity*)	
(*Reference Area*)	

STAR-Inventor Initialization Panel

Specifying initial conditions that are close to the expected solution can reduce the number of iterations required to reach a solution. You can also set the initial conditions to match inlet conditions of the fluid domain. You can define the following continuum reference properties:

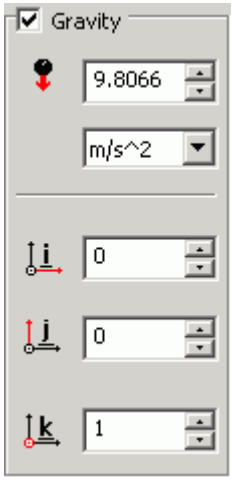
Property	Description
(*U Component of Velocity*)	The U direction vector for the initial velocity.
(*V Component of Velocity*)	The V direction vectors for the initial velocity.

Property	Description
(*W Component of Velocity*)	The W direction vectors for the initial velocity.
(*Initial Pressure*)	The initial pressure. The physical value of the initial pressure defines the starting pressure in the regions to which the continuum is assigned. The initial pressure is specified with respect to the reference pressure. The default initial pressure is `0 Pa`.
(*Initial Temperature*)	The initial temperature. This physical value of the initial temperature defines the starting temperature throughout any region to which the continuum is assigned. The default initial temperature is `300 K`.
(*Turbulence Intensity*)	The initial turbulence intensity. The default value is `0.05` (`5%`)
(*Turbulent Length Scale*)	The initial turbulent length scale. The default value is `0.01 m`.

Activating the Gravity Model in STAR-Inventor

If you require the gravity model in STAR-Inventor, you can activate it in the *Continuum Settings* dialog:

1. In the *Continuum Settings* dialog, navigate to the *Reference* tab.
2. Activate the **Gravity** checkbox.

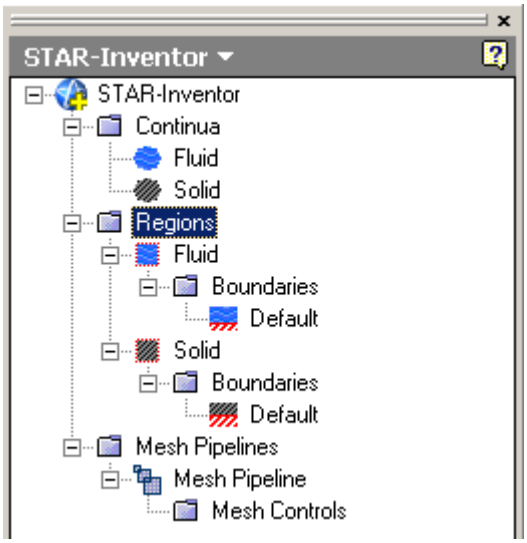


3. Set the acceleration due to gravity in the (*Acceleration due to gravity*) box.
   You can change the units from the drop-down list.
4. Specify the direction vector for the gravity in the `[i, j, k]` *Components of Gravity* box.
5. Click **OK**.
   The continuum is added to the object tree.

Setting Up Regions

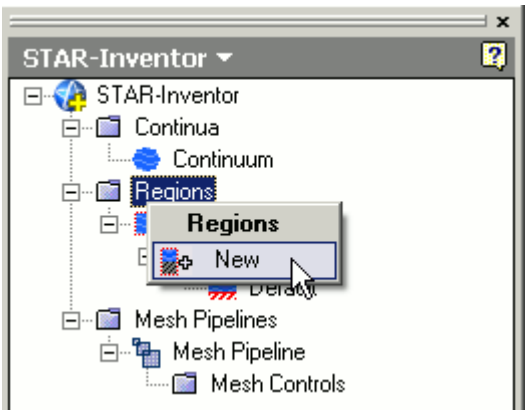
Additional regions are required when modelling multiple region physics, or when specific behavior such as porous resistance is required for a particular part within the same continuum. Assign parts from an assembly to regions. You can control the properties of these regions through the assigned continuum.

In assembly mode, regions are stored under the **Regions** node in the object tree, as shown below.

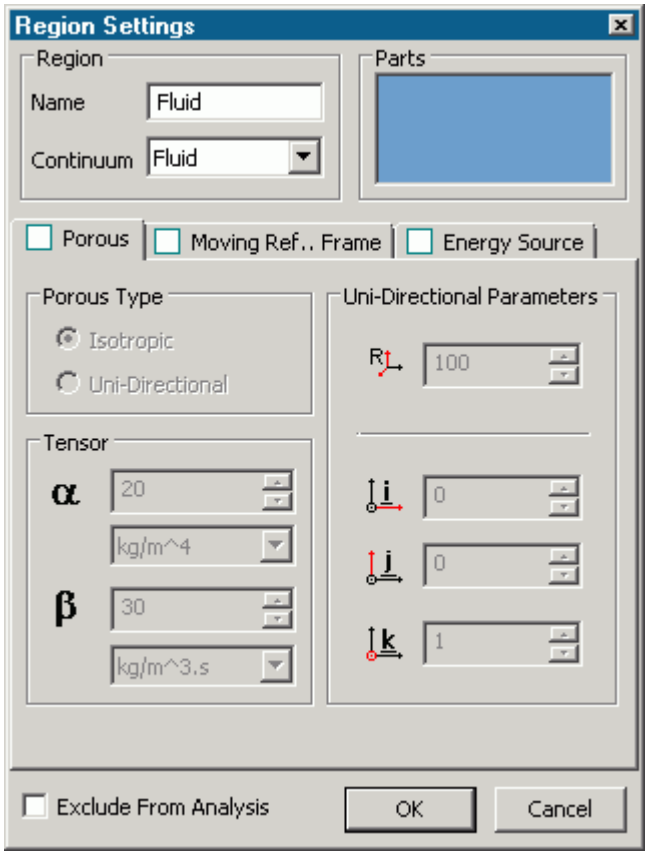


To create an additional region:

1. Right-click the **Regions** node in the object tree and select **New**.



The *Region Settings* dialog appears.

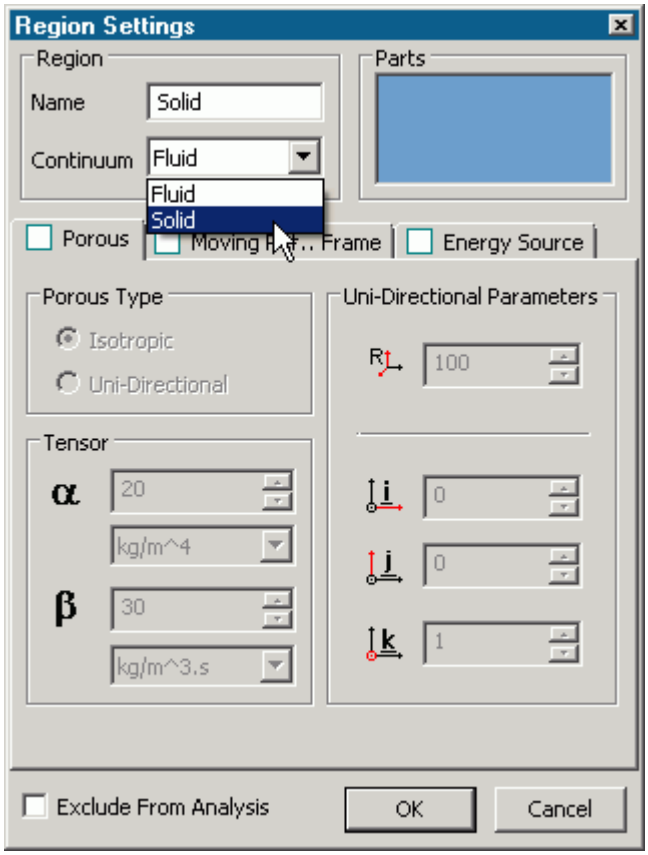


2. In the *Name* property, enter a name for the region.
3. Select the parts that you want to include in the region.

   You can select multiple parts from the graphics window or the Inventor feature tree.

   Selected parts appear in the *Parts Selection* group box field.
4. Select a continuum to apply to the region.

   All available continua are listed here.



Note: Assigning a continuum to a region causes the region to inherit the physical and material properties defined in the continuum

5. Activate any additional region physics for the region:
   - Porous. See Defining a Porous Region.
   - Moving Reference Frame. See Defining a Moving Reference Frame.
   - Energy Source. See Defining an Energy Source.
6. If you want to import parts into STAR-CCM+ for visualization purposes, but do not want the region to be included in the CFD calculations, activate the *Exclude from Analysis* option. See Importing Geometry into STAR-CCM+ for Visualization.
7. Click **OK**.

A new region is added to the STAR-Inventor object tree under the **Regions** node.

The parts that you assign to a region define the geometric extent of the region. Within this geometry, properties and behavior of the material is governed by the physics models and material properties defined in the region and continuum settings.

Note: Each region must be assigned at least one part, otherwise it will be ignored during the CFD analysis.

By default, every STAR-Inventor model contains one continuum, and this is automatically assigned to the default region and each additional region that you create. If multiple continua are defined in the model, you

can choose a different continuum to assign to each region. This enables you to control the physical properties of each region, including varying material types.

All the parts in an assembly are assigned to the default region (this is the initial region created by STAR-Inventor). When you assign a part to another region, it is automatically removed from the default region. If you remove a part from a region, it is reassigned back to the default region.

Contents:

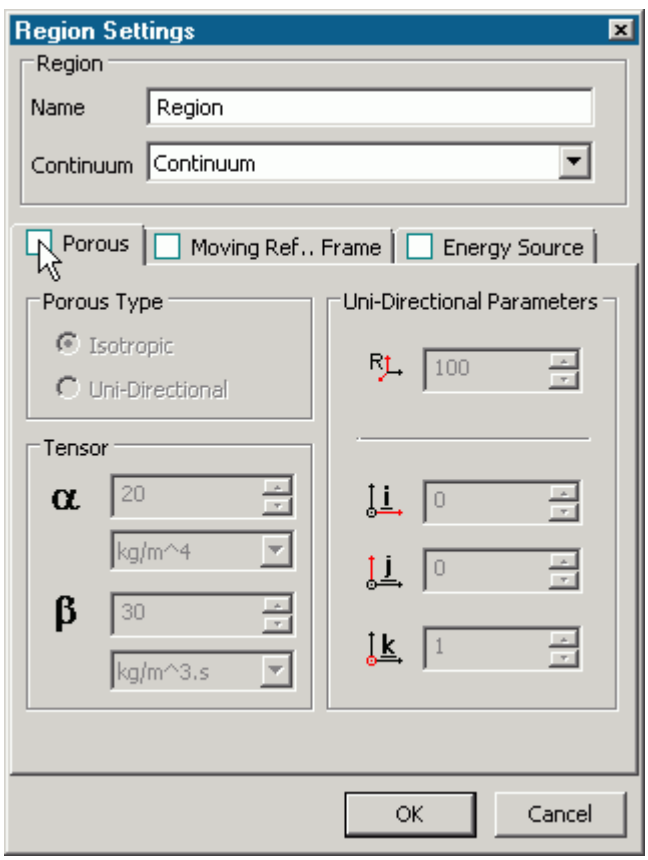
Defining a Porous Region
Defining a Moving Reference Frame
Defining an Energy Source

Defining a Porous Region

Porous resistance affects fluid flow through a region in a particular direction.

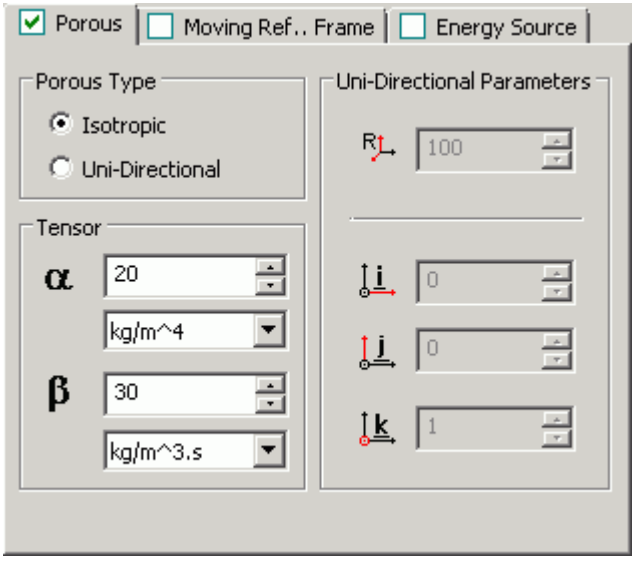
To define a porous region:

1. In the *Region Settings* dialog, activate the *Porous* tab.

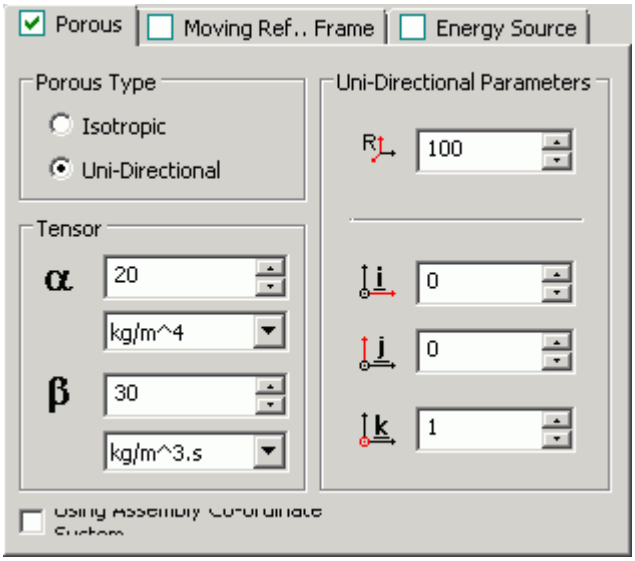


The Porous settings appear.

2. Select the type of resistance to model:
   - **Isotropic**



- **Uni-Directional**



3. Specify the porosity settings. See Porous Region Panel Settings.

Contents:

Porous Panel Settings

Porous Panel Settings

Porous resistance affects the fluid flow through a region in a particular direction. There are two types of porous resistances available: Isotropic and Uni-Directional.

Parameter	Description
Isotropic	

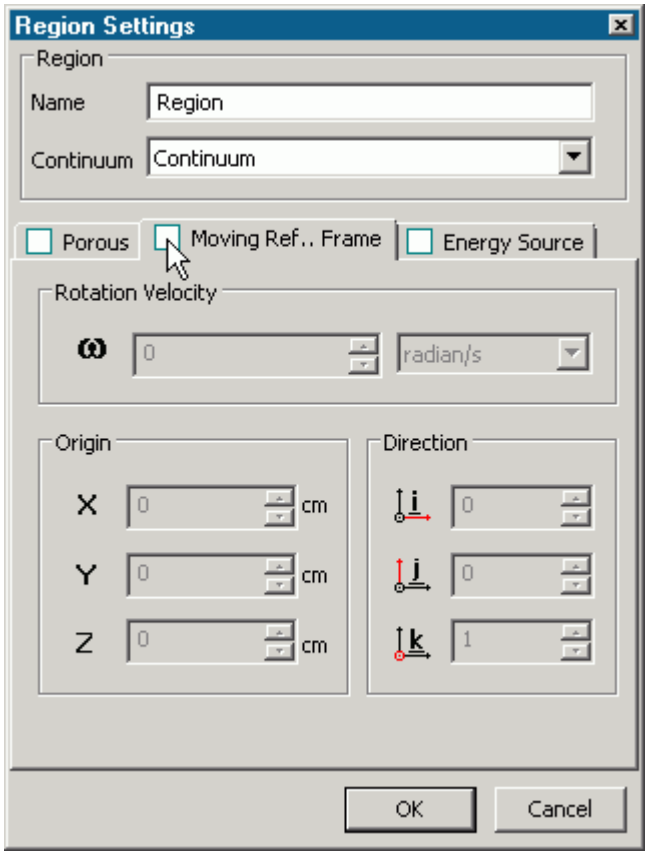
Parameter	Description
α (*Alpha*)	Specifies inertial resistance. You can change the units using the drop-down list.
β (*Beta*)	Specifies viscous resistance. You can change the units using the drop-down list.
Uni-Directional	
α (*Alpha*)	Specifies inertial resistance. You can change the units using the drop-down list.
β (*Beta*)	Specifies viscous resistance. You can change the units using the drop-down list.
(*Cross-Flow Resistance Factor*)	Specifies the cross-flow resistance factor.
(*i Component of Flow Direction*)	Specify the components that define the direction of the flow in the porous region. These values are required to determine the coefficients of inertial and viscous resistance in a direction perpendicular to the specified vector.
(*j Component of Flow Direction*)	
(*k Component of Flow Direction*)	

Defining a Moving Reference Frame

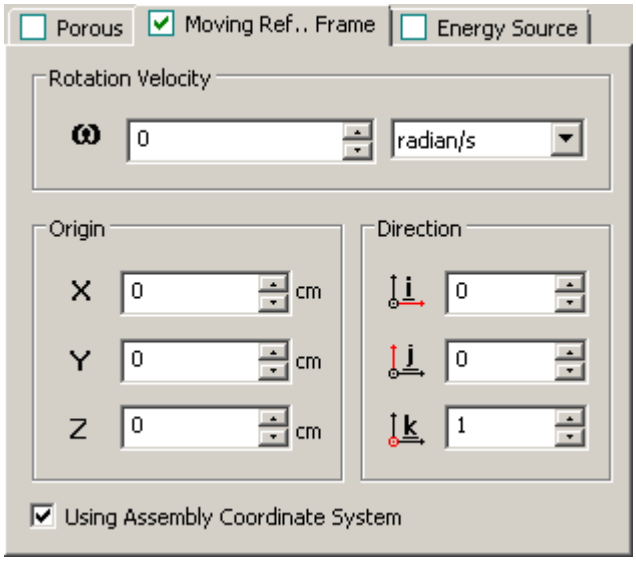
You can use moving reference frames to model region motion.

To define a moving reference frame:

1. In the *Region Settings* dialog, activate the *Moving Reference Frame* tab.



The Moving Reference Frame settings appear.



2. Specify the Moving Reference Frame parameters. See Moving Reference Frame Panel Settings.

Contents:

Moving Reference Frame Panel Settings

Moving Reference Frame Panel Settings

You can use Moving Reference Frames to model region motion, such as rotation.

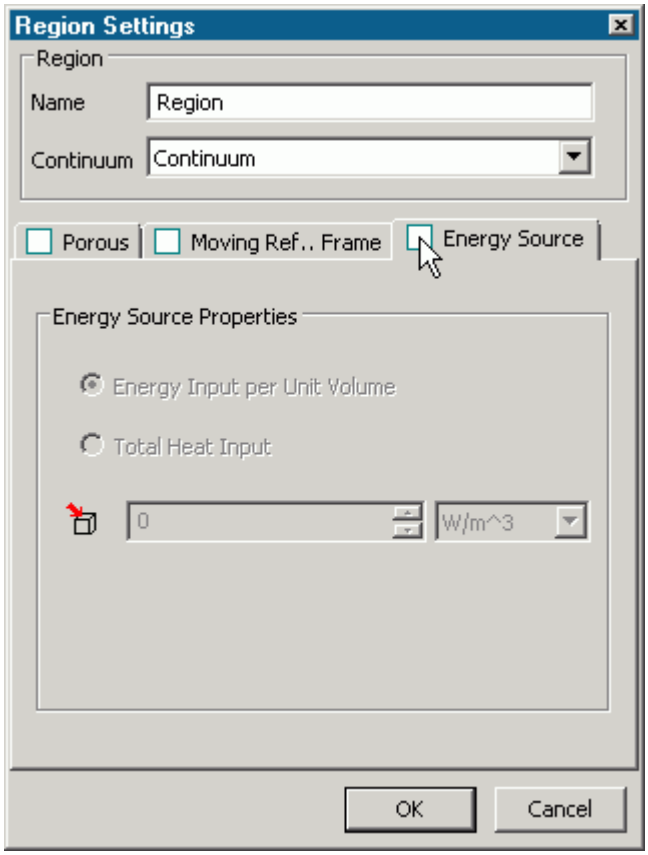
Parameter	Description
(*Rotational Velocity*)	Specifies the angular velocity. You can change the units from the drop-down list.
X (*X Coordinate of Axis Origin*)	Specify the coordinates for the origin of the axis of rotation. The units that are used here are the same as the file defaults.
Y (*Y Coordinate of Axis Origin*)	
Z (*Z Coordinate of Axis Origin*)	
(*i Component of Axis Vector*)	Specify the direction of the axis of revolution.
(*j Component of Axis Vector*)	
(*k Component of Axis Vector*)	
Use Assembly Coordinate System	Controls whether the properties of the axis of rotation are defined using the coordinate system of the assembly or the coordinate system of the selected part. To use the coordinate system of the assembly, activate the *Use Assembly Coordinate System* this option. If you select multiple parts for the new region, the *Use Assembly Coordinate System* option is automatically activated and grayed out

Defining an Energy Source

You can define a direct source or sink of energy to or from a region using the energy source model. This model is only available if you activated the Temperature physics model in the continuum.

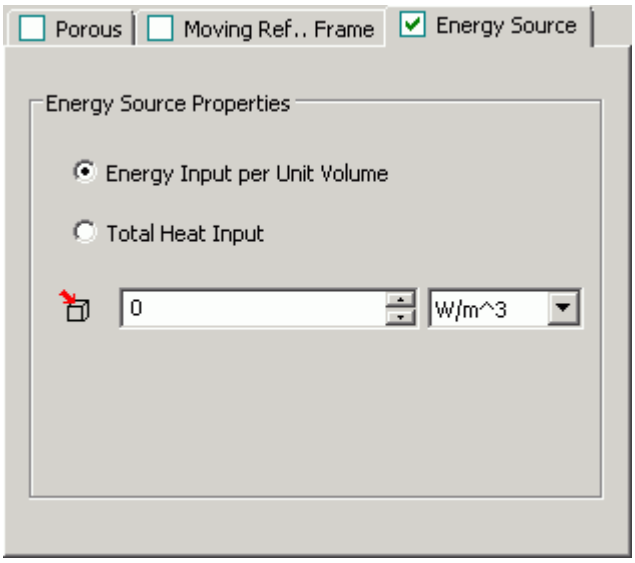
To specify that a region is an energy source:

1. Make sure that the Temperature model is activated in the continuum that is applied to the region. See Setting Up Continua.
2. In the *Region Settings* dialog, activate the *Energy Source* tab.

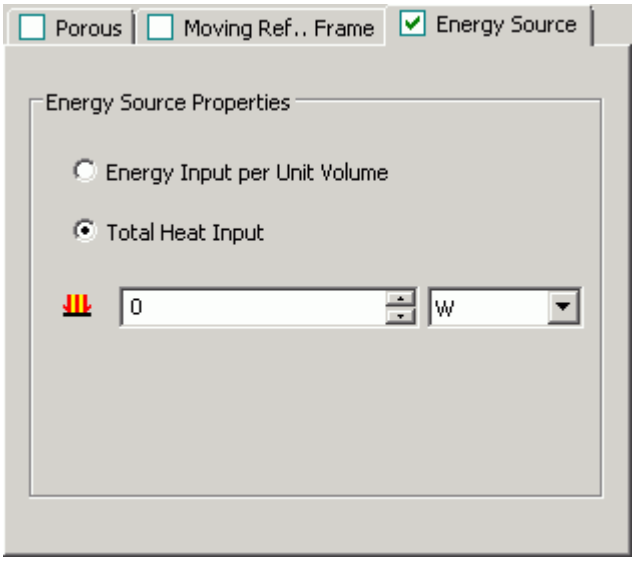


The Energy Source settings appear.

3. Select the type of energy source to apply:
   - **Energy Input per Unit Volume**



   - **Total Energy Input**



4. Specify the energy source settings:
   - For energy input, set (*Energy Input per Unit Volume*) to the desired value.
   - For total heat, set (*Total Energy Input*) to the desired value.

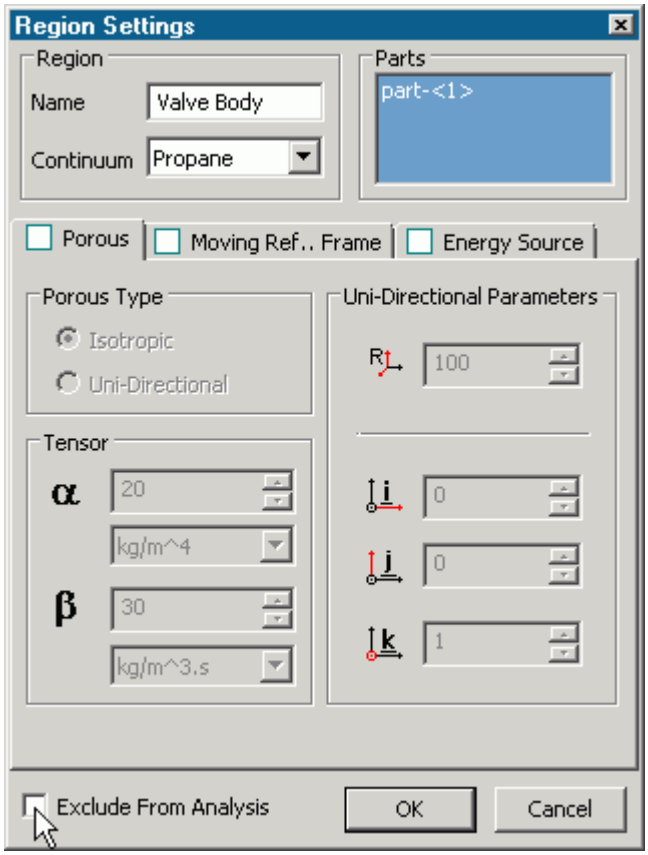
Importing Geometry into STAR-CCM+ for Visualization

You can import CAD parts, that do not form part of the analysis, into STAR-CCM+ for visualization purposes.

Add these parts to a separate region—this region is excluded from the analysis. This feature is useful for showing the original geometry in a post-processing scene in STAR-CCM+. For example, for an internal fluid volume of a pipe, the original geometry is the solid pipe.

To exclude CAD parts from the analysis:

1. In the *Region Settings* dialog, add the CAD parts that you want to use for visualization purposes.
2. Activate the *Exclude From Analysis* checkbox in the *Advanced* panel.



3. Click **OK**.

When you run the case, a surface mesh is generated for this region using the settings specified in the Mesh Pipeline.

Note: As is the case for all assemblies, you must imprint all parts that are used for visualization purposes onto any surface that they cut, otherwise you will not be able to generate the surface mesh.

An example of this feature is shown in the following section, where a valve and valve body geometry are imported into STAR-CCM+ for visualization.

Contents:

Setting Up a STAR-CCM+ Scene to Display Imported Geometry from STAR-Inventor

Setting Up a STAR-CCM+ Scene to Display Imported Geometry from STAR-Inventor

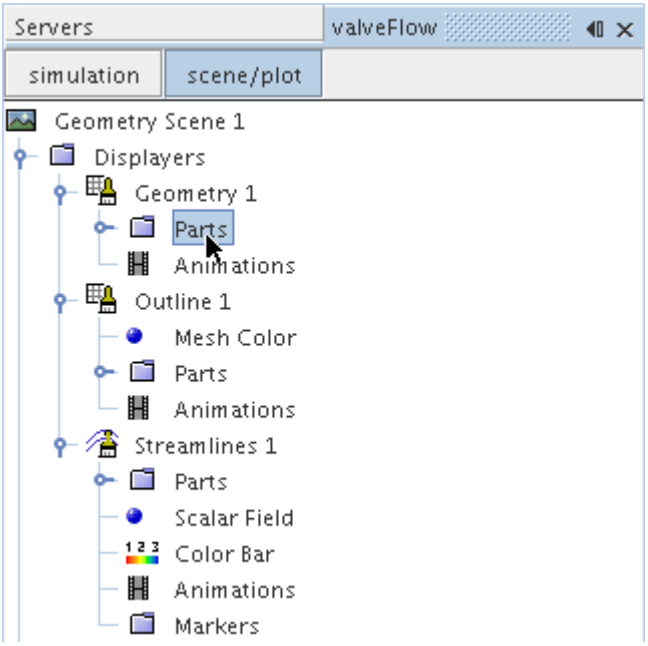
You can visualize geometry that is not included in the analysis. For example, you can add the valve and valve housing geometry to a Geometry Scene in STAR-Inventor, even though this does not have any effect to the flow.

To use parts that are excluded from the analysis in a scene:

1. In STAR-CCM+, create a new scene.

   In the following example, a Geometry Scene is used.
2. Create a streamlines derived part.
3. Click on the **scene/plot** button.

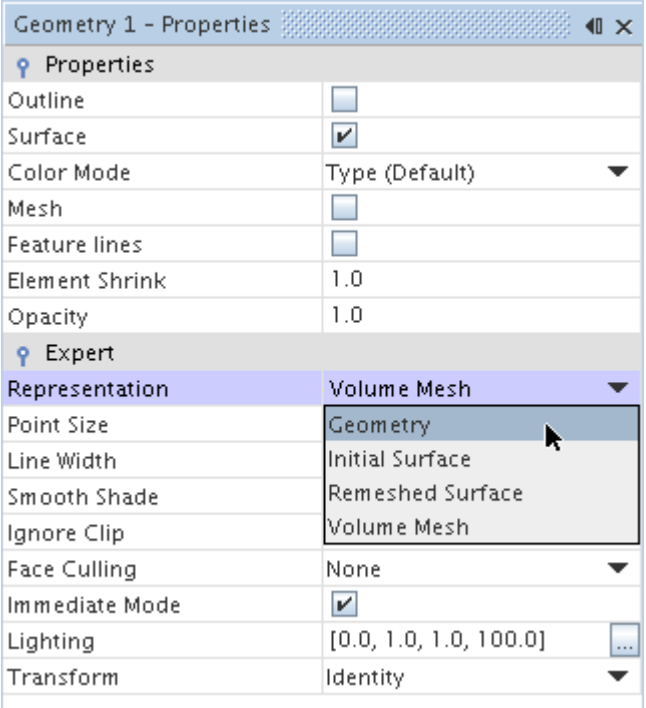
   The object tree switches to the scene nodes.
4. Double-click on the **Geometry Scene** > **Displayers** > **Geometry** > **Parts** node.



5. In the *Edit* dialog:
   a) Deactivate the **Regions** node.
   b) Expand the **Parts** node and activate the part that corresponds to the valve body.

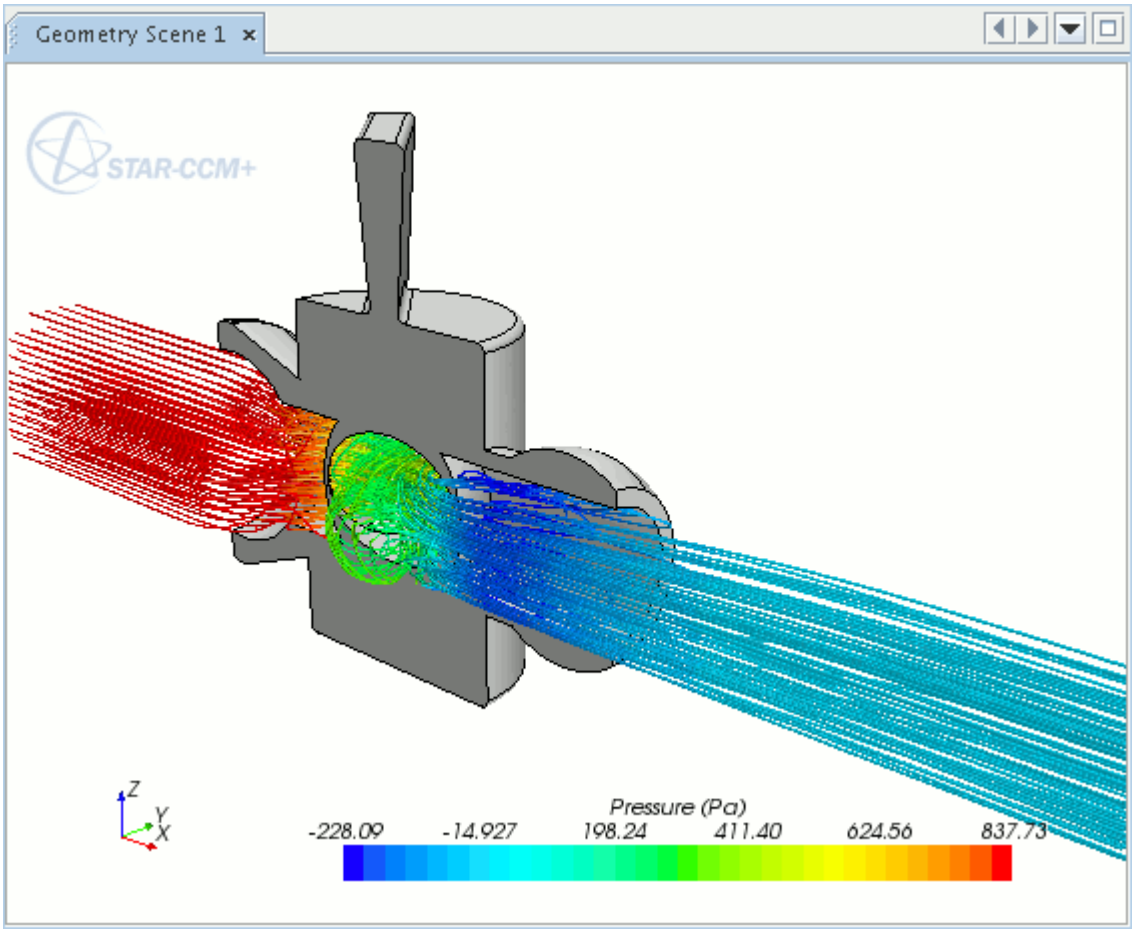
      This is the region that you chose to exclude from the analysis in STAR-Inventor.
6. Click **OK**.
7. Select the **Geometry** node and set *Representation* to **Geometry**.

   The imported geometry only contains a surface mesh and does not appear by default (the default *Representation* is set to **Volume**). This is why you must change the *Representation* to **Geometry**.



8. Activate the *Outline* option to change the color of the outlines from red to black.

The finished scene is shown below.



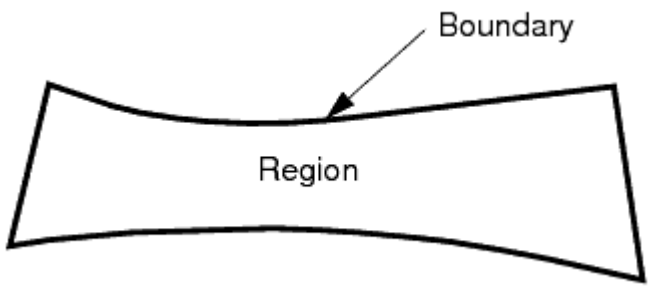
Displaying Imported Geometry in Scalar and Vector Scenes in STAR-CCM+

To display an imported geometry in a Scalar or Vector scene, you need to add a geometry displayer to the existing scene. You can then display the geometry by following steps described in Setting Up a STAR-CCM+ Scene to Display Imported Geometry from STAR-Inventor.

Setting Boundary Conditions

Boundary conditions tell the STAR-CCM+ solvers how the substance within the region interacts with its environment. A velocity inlet boundary, for example, tells the solvers that an amount of substance is being added to the fluid domain at a certain rate.

A STAR-Inventor boundary is made up of one or more CAD surfaces within the simulation which, along with other boundaries, surround and define a region.



The boundary is discretized by the surface mesher, which is then used by the volume mesher to mesh the entire region.

It is important to set the boundary conditions before creating the volume mesh, as some parts of the volume mesh, such as prism layers, depend on the type of boundary.

By default, each region contains only one boundary. To define more boundaries you need to add boundaries to the region. You can modify the properties of existing boundaries by editing its boundary node.

Some boundaries may require the specification of turbulence quantities, temperature, and passive scalar concentration, depending on whether these modeling options have been activated in Continuum Settings.

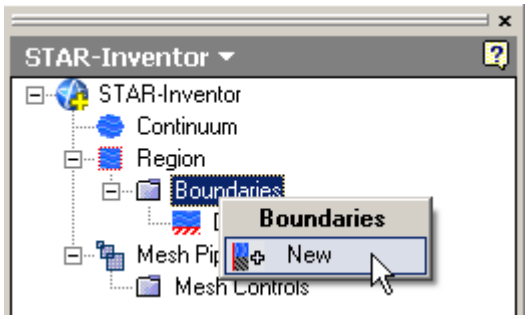
Contents:

Creating a New Boundary

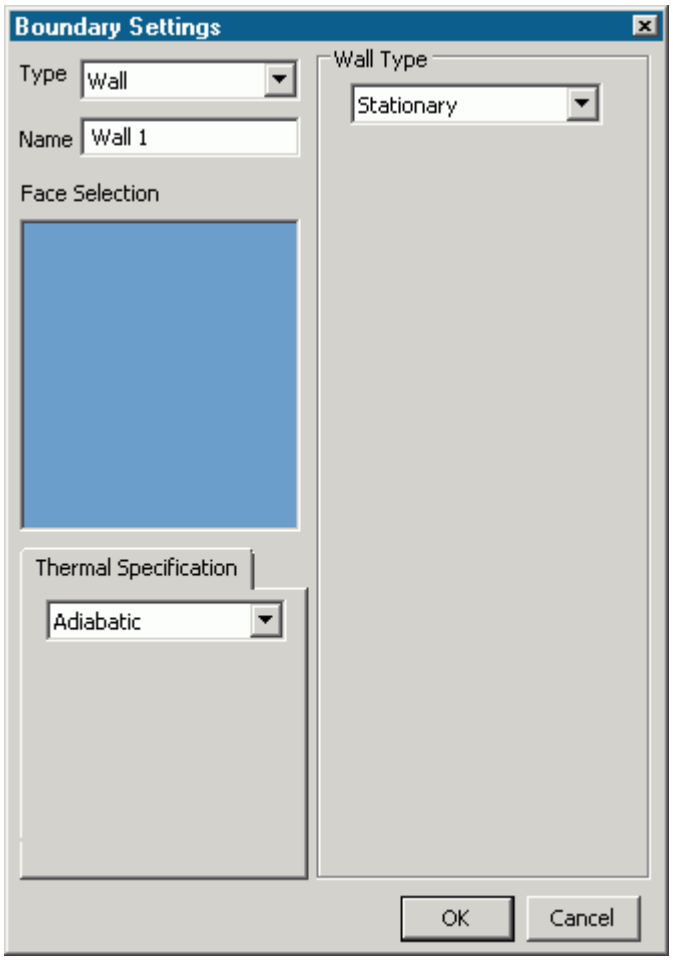
Creating a New Boundary

To create a new boundary:

1. Right-click the **Boundaries** node within the object tree and select **New**.



The *Boundary Settings* dialog appears.



2. On the CAD model, select the surfaces that you want to include in the boundary. See Assigning Surfaces to Objects.

   These selections are reported in the *Face Selection* box in the *Boundary Settings* dialog.
3. In the *Name* field, enter a name for the boundary.
4. Select the boundary type and set the boundary conditions:
   - **Velocity Inlet**. See Velocity Inlet Boundary.
   - **Mass Flow Inlet**. See Mass Flow Inlet Boundary.
   - **Stagnation Inlet**. See Stagnation Inlet Boundary.

- **Outlet**. See Outlet Boundary.
- **Pressure Outlet**. See Pressure Outlet Boundary.
- **Symmetry Plane**. See Symmetry Plane Boundary.
- **Wall**. See Wall Boundary.
- **Custom**. See Custom Boundary.

5. Click **OK**.

Contents:

Boundary Type Reference
Common Boundary Conditions

Boundary Type Reference

This section contains the various boundary types that are available in STAR-Inventor.

Velocity Inlet Boundary

A velocity inlet boundary represents the inlet of a duct at which you know the flow velocity. The velocity inlet settings are displayed below.

Property	Description
Normal Velocity	
(*Normal Velocity*)	Specifies the velocity normal to the surface. You can change the units using the drop-down list.
Cartesian Velocity Components	
U (*U Component of Velocity Vector*)	Specify the components of the velocity vector. You can change the units using the drop-down list.
V (*V Component of Velocity Vector*)	
W (*W Component of Velocity Vector*)	
Cylindrical Velocity Components	
V_r (*r Component of Velocity Vector*)	Specify the cylindrical components of the velocity vector. You can change the units using the drop-down list.
V_θ (*theta Component of Velocity Vector*)	
V_z (*z Component of Velocity Vector*)	
X (*X Location of Axis of Origin*)	Specify the coordinates of the origin for the velocity vector.
Y (*Y Location of Axis of Origin*)	
Z (*Z Location of Axis of Origin*)	
i (*i Component of Direction Vector*)	Specify the components for the direction vector.

Property	Description
(*j Component of Direction Vector*)	
(*k Component of Direction Vector*)	

Mass Flow Inlet Boundary

A mass flow inlet boundary represents the inlet of a duct at which you know the mass flow rate. The mass flow settings are displayed below.

Property	Description
Mass Flow Normal to Surface	
(*Mass Flow Rate*)	Specifies the mass flow rate, normal to the boundary surface. You can change the using the drop-down list.
Mass Flow in Specified Direction	
(*Mass Flow Rate*)	Specifies the mass flow rate. You can change the using the drop-down list.
(*i Component of Mass Flow*)	Specify the direction components of the mass flow.
(*j Component of Mass Flow*)	
(*k Component of Mass Flow*)	

Stagnation Inlet Boundary

The stagnation inlet boundary is an inlet condition that is well-posed for compressible flows, although it is equally valid for incompressible flows. The stagnation conditions refer to the conditions in an imaginary plenum, far upstream, in which the flow is completely at rest. The stagnation inlet settings are displayed below.

Property	Description
Stagnation Flow Normal to Surface	
(*Stagnation Pressure*)	Specifies the stagnation pressure. You can change the units using the drop-down list.
Stagnation Flow in Specified Direction	
(*Stagnation Pressure*)	Specifies the stagnation pressure. You can change the units using the drop-down list.
(*i Component of Direction Vector*)	Specify the direction components for the flow.
(*j Component of Direction Vector*)	
(*k Component of Direction Vector*)	

Outlet Boundary

An outlet boundary splits the flow between multiple outlets using a fraction of the total mass flow through each of the outlet boundaries. The outlet panel settings are provided below.

Property	Description
60 40 (*Flow-Split Ratio*)	Specifies the ratio of the split as a value between 0 and 1. The sum of all the flow-split ratio values in a region must equal 1. If you only have one flow-split boundary in the simulation, all the flow is assumed to exit through this boundary.

Pressure Outlet Boundary

A pressure boundary is a flow outlet boundary at which you specify the pressure. The pressure outlet panel settings are provided below.

Property	Description
(*Pressure*)	Specifies the pressure on the selected boundary. You can change units using the drop-down list.

Symmetry Plane Boundary

A symmetry plane boundary represents an imaginary plane of symmetry in the simulation. The solution that is obtained with a symmetry plane boundary is identical to the solution that would be obtained by mirroring the mesh about the symmetry plane (in half the resulting domain). There are no settings which are specific to symmetry plane boundaries.

Wall Boundary

A wall boundary represents an impermeable surface. The wall panel settings are provided below.

Property	Description	
Wall Type	Specifies the type of conditions to apply to the wall boundary.	
	Stationary	Specifies the wall as a stationary impenetrable surface.
	Translating	Specifies the wall as a moving impenetrable surface. You specify the velocity for the translation.
	Rotating	Specifies the wall as a rotating impenetrable surface.
	Slip	Specifies the wall as a stationary, impenetrable, but friction-free surface.
Translating		
U (*U Component of Velocity Vector*)	Specify the velocity vector. You can change the units using the drop-down list.	
V (*U Component of Velocity Vector*)		

Property	Description
(*U Component of Velocity Vector*)	
Rotating	
(*Rotational Velocity*)	Specifies the rotational velocity. You can change the units using the drop-down list.
X (*X Location of Axis Origin*)	Specify the coordinates of the origin for the axis of rotation.
Y (*Y Location of Axis Origin*)	
Z (*Z Location of Axis Origin*)	
(*i Component of Direction Vector*)	Specify the components of the axis of rotation
(*j Component of Direction Vector*)	
(*k Component of Direction Vector*)	

Custom Boundary

In STAR-CCM+, there are additional boundary types to those available in STAR-Inventor. The custom boundary in STAR-Inventor acts as a placeholder for a boundary that you will customize in STAR-CCM+.

Common Boundary Conditions

Some boundary conditions are not specific to any boundary type and appear in multiple boundaries, so long as the physics model that is associated with the condition is activated.

Common Boundary Conditions on Inlet and Pressure Outlet Boundaries

Temperature, turbulence, and passive scalar boundary conditions are common to all inlet boundaries as well as the Pressure Outlet boundary.

Property	Description
(*Temperature*)	Specifies the temperature on the boundary surface. You can change the units using the drop-down menu.
(*Turbulence Intensity*)	Specifies the turbulence intensity on the boundary surface. You can change the units using the drop-down menu.
(*Turbulent Length Scale*)	Specifies the turbulent length scale on the boundary surface. You can change the units using the drop-down menu.

Property	Description
(*Passive Scalar*)	Specifies the amount of passive scalar crossing the boundary. You can change the units using the drop-down menu.

Boundary Conditions on Wall Boundaries

When the energy equation is being solved, you can specify the thermal specification on wall boundaries.

Property	Description	
Adiabatic	Specifies that no transfer of energy occurs between the boundary surface and the surrounding environment.	
Convection	Specifies the transfer of energy between the boundary surface and the fluid moving over the boundary surface.	
	HTC (*Heat Transfer Coefficient*)	Specifies the heat transfer coefficient. You can change the units using the drop-down menu.
	(*Ambient Temperature*)	Specifies the ambient temperature. You can change the units using the drop-down menu.
Heat Flux	Specifies the rate of energy transfer through the boundary surface.	
	(*Heat Flux*)	Specifies the heat flux. You can change the units using the drop-down menu.
Temperature	Specifies the temperature on the boundary surface.	
	(*Temperature*)	Specifies the temperature on the boundary surface.

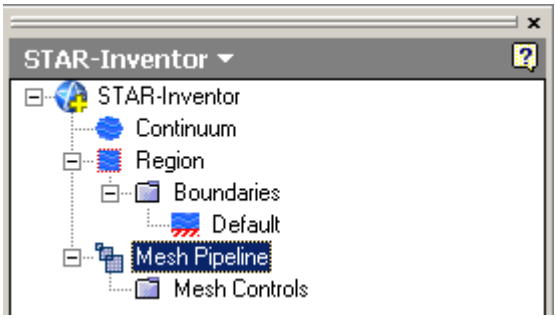
Meshing the Model

STAR-Inventor provides options for generating polyhedral or trimmed volume meshes, with or without near-wall prism layers.

By default, prism layers are generated on all wall surfaces. The provision of a prism layer depends on the type of boundary that is applied to a face. As a consequence, it is recommended that you only generate the mesh after setting up the boundaries and regions.

The meshing process first creates a triangulated surface (the surface mesh) and thereafter converts it to the desired volume mesh (polyhedral or trimmer). As the quality of the volume mesh depends to a large extent on the quality of the surface mesh, it is prudent to check the surface mesh before proceeding with the volume mesh.

Although the entire meshing process can be launched with a single command (the **Build And Run** command from the STAR-Inventor ribbon), it is possible to execute only single stages from the STAR-Inventorribbon. Settings that are made in the **Mesh Pipeline** node govern the mesh generation process.



A mesh pipeline contains all the settings that are applied to the surface and volume meshes created on the Inventor parts to which it is applied. In single part mode, only one mesh pipeline is available. In assembly mode, multiple mesh pipelines can be created if necessary.

When mesh generation commences, each mesh pipeline will:

1. Extract geometry from Inventor.
2. Transfer the model to STAR-CCM+.
3. Execute the surface remesher.
4. Execute the polyhedral or trimmer volume mesher with the prism layer mesher (if selected).
5. Create a mesh representation in the Inventor *Graphics* window.

Controls on the top-level mesh pipeline node allow you to set the base sizes for the surface mesh, and control the prism layer. Further refinement to specific faces, parts, or volume shapes can be made by adding additional controls to the mesh pipeline.

Contents:

Creation of Additional Surfaces in STAR-CCM+
Working with Mesh Pipelines

Working with Mesh Controls
Generating the Mesh STAR-Inventor
Common Causes for Meshing Failures

Creation of Additional Surfaces in STAR-CCM+

STAR-CCM+ creates geometry part surfaces during the geometry transfer stage.

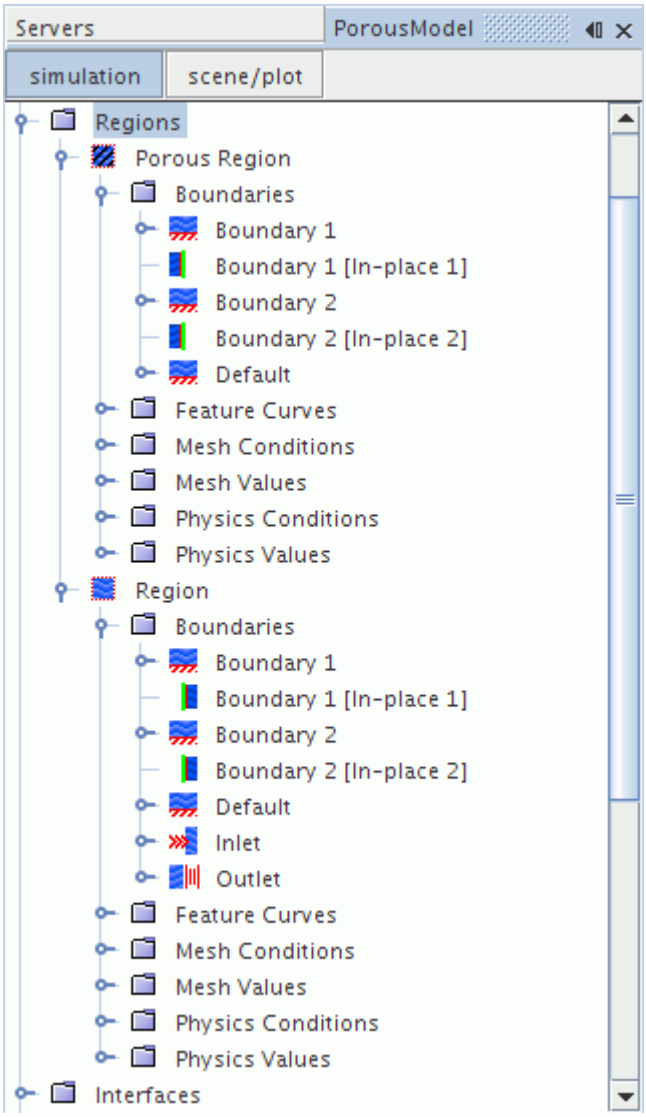
Initially, STAR-Inventor creates one part surface for each face in the CAD model. These part surfaces are then combined so that one surface per boundary remains. This outcome, however, is not possible for boundaries that contain CAD surfaces from more than one part. The final part surfaces appear in the **Geometry** > **Parts** node in STAR-CCM+.

The following items affect the number of part surfaces that are created in STAR-CCM+:

- **The number of boundaries present in the model, including the default boundary.** Each boundary in STAR-Inventor appears as a separate part surface in STAR-CCM+. If, however, a boundary contains CAD surfaces from multiple parts, the part surfaces in STAR-CCM+ cannot be combined.
- **The number of mesh pipeline controls used (if any).** Each CAD surface that is used to define a face mesh control in STAR-Inventor appears as a separate part surface in STAR-CCM+.
- **The number of interfaces that are created during the meshing process (for assemblies only).** All the CAD surfaces that define one interface between two parts in an assembly are grouped together under one part surface in STAR-CCM+. One part surface exists for each interface in STAR-CCM+.

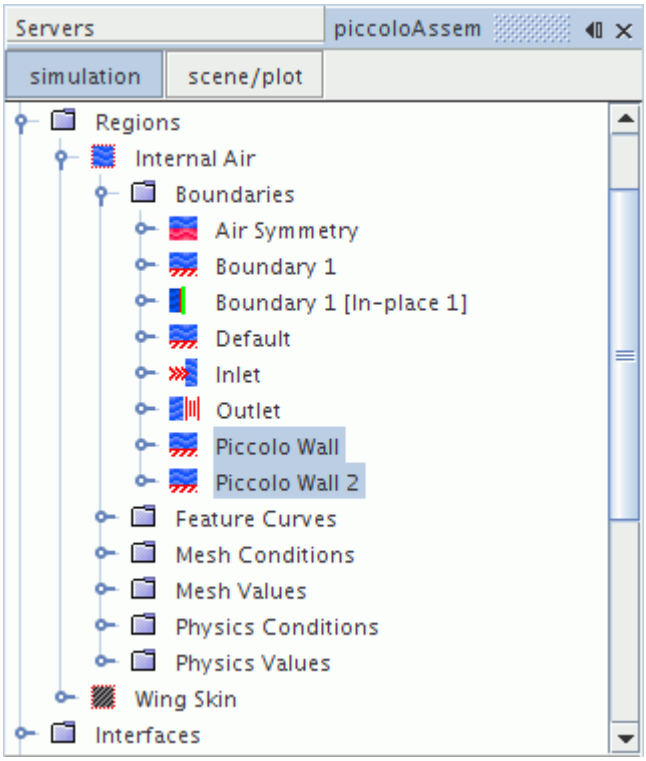
Boundary Splitting due to Interface Generation

Interfaces are automatically created between two parts in an assembly during the mesh generation. A new boundary is created for each interface on the surfaces that share the interface.



Boundary Splitting due to Face Mesh Controls and Part Mesh Controls

If you use a face mesh control or a part mesh control in your simulation, some boundaries may be split, depending on what faces are included in the mesh control. If a boundary and mesh control share the same faces, no splitting occurs. If a mesh control contains faces from multiple boundaries, splitting occurs and a new boundary is created. The name of the new boundary is derived from the original boundary.



Working with Mesh Pipelines

When working with a single part, you will only have one mesh pipeline defined in the model tree. This will specify the mesh settings for the single part available. In assembly mode, multiple mesh pipelines can be created to give specific controls for different parts, and parts must be specifically associated with a mesh pipeline.

Contents:

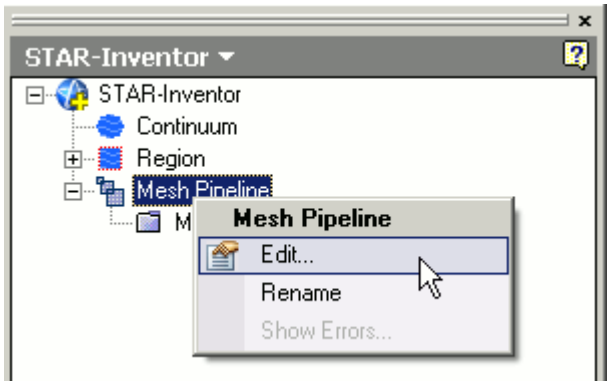
Setting Up the Mesh Pipeline

Setting Up the Mesh Pipeline

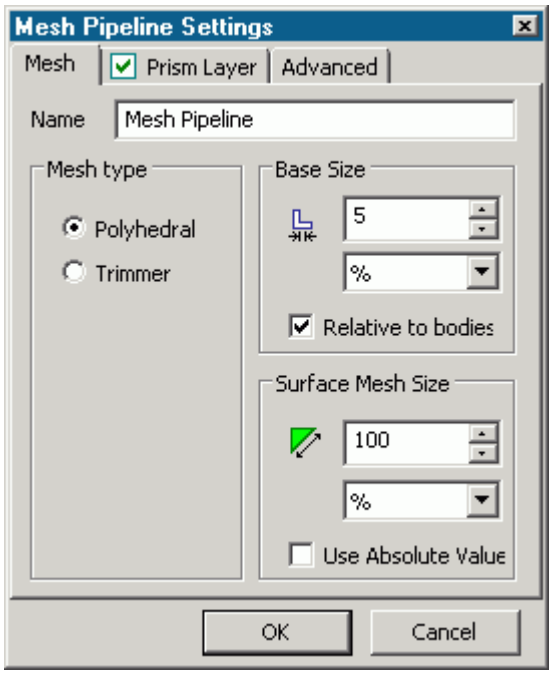
The mesh pipeline node contains the general mesh settings that will apply to all STAR-Inventor parts associated within the pipeline in the absence of any further controls.

To set up the mesh pipeline:

1. Right-click the **Mesh Pipeline** node in the STAR-Inventor tree and select **Edit**.



The *Mesh Pipeline Settings* dialog appears.



2. In the *Name* field, enter the name of the mesh pipeline if required.
3. In the *Part Selection* box, specify the desired parts to use in the mesh pipeline.

   This property is only available when working with assemblies.
4. From the *Mesh Type* tab, select the type of mesh that you want to generate:
   - **Polyhedral**
   - **Trimmer**
5. Set *Base Size* to the desired value.

   The base size is a reference length used in other meshing parameters. It may be set to equal the length of the largest triangle you expect to have on your surface mesh, or to some other characteristic dimension.

   The base size can be set as either a relative or absolute value. When it is set as a relative value, the reference size will be the diagonal of the bounding box enclosing the entire geometry. In the case of multiple parts in an assembly, the reference size will be calculated based on the diagonal length of the

bounding box enclosing the entire assembly. This includes all parts from all mesh pipelines. The unit specification menu will become active when setting an absolute value.

6. Set *Surface Mesh Size* to the desired value.

   The surface mesh size is the base edge length of triangles on the surface mesh. Note that the ultimate length of some (or all) triangles on the surface mesh will depend on other factors such as surface curvature and edge proximity.

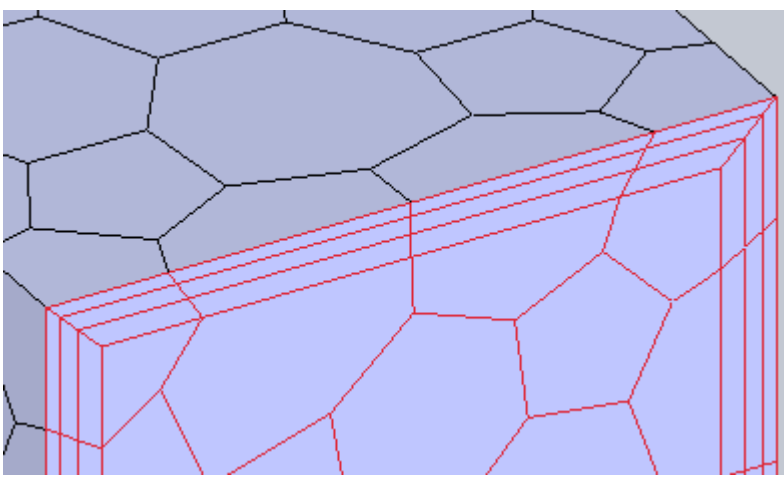
7. Set the prism layer settings. See Prism Layer Panel.
8. Set the advanced settings. See Mesh Pipeline Advanced Settings Panel.
9. If you selected to generate a trimmed mesh, set the Trimmer settings. See Trimmer Panel.
10. Click **OK**.

Contents:

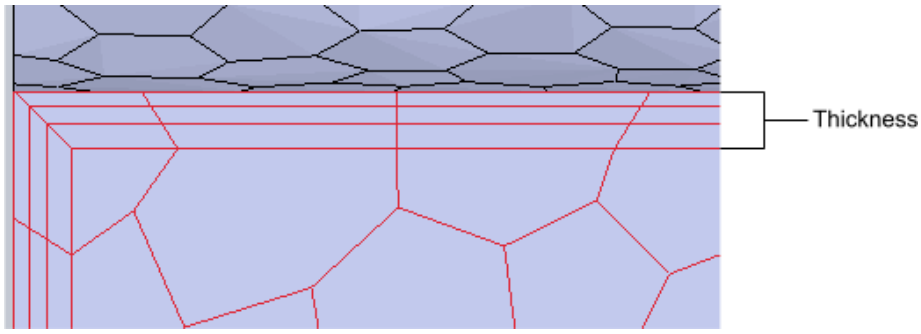
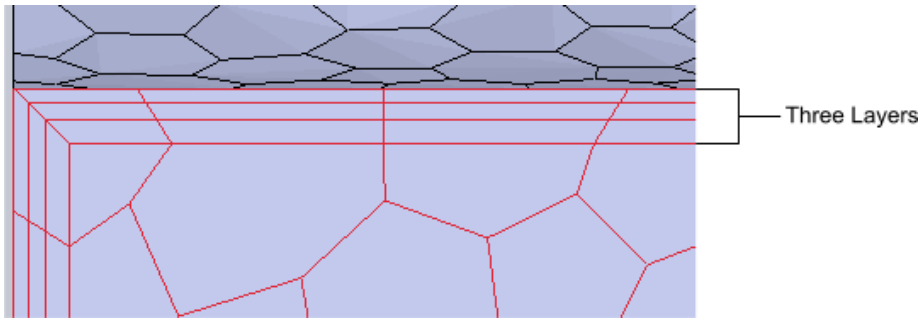
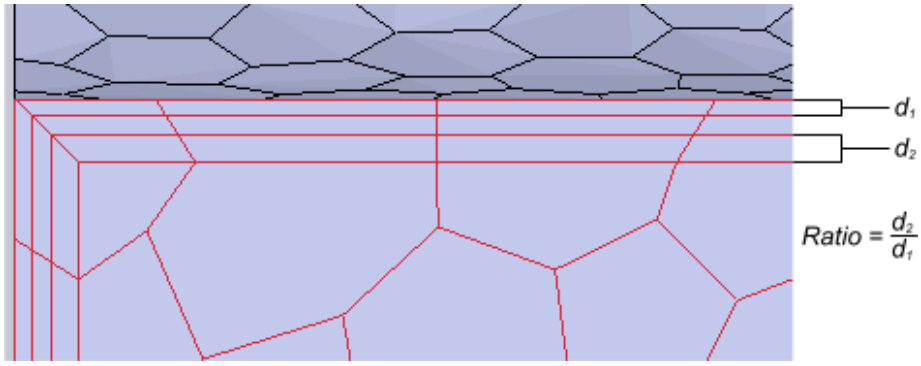
Prism Layer Panel
Mesh Pipeline Advanced Settings Panel
Trimmer Panel

Prism Layer Panel

Prims layers improve the simulation near non-slip walls, particularly when a turbulence model is active.



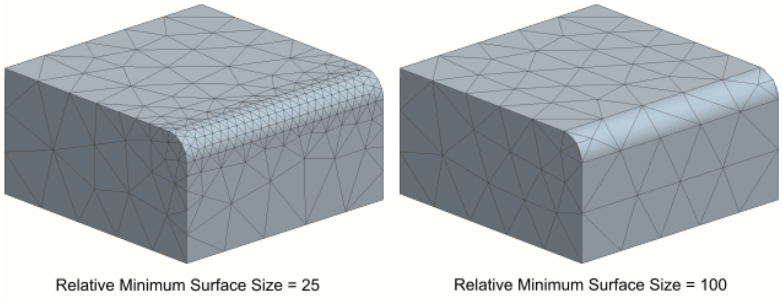
Property	Description
Layer Thickness	Defines the overall depth of the prism layer, which includes all of the individual layers, as illustrated below.

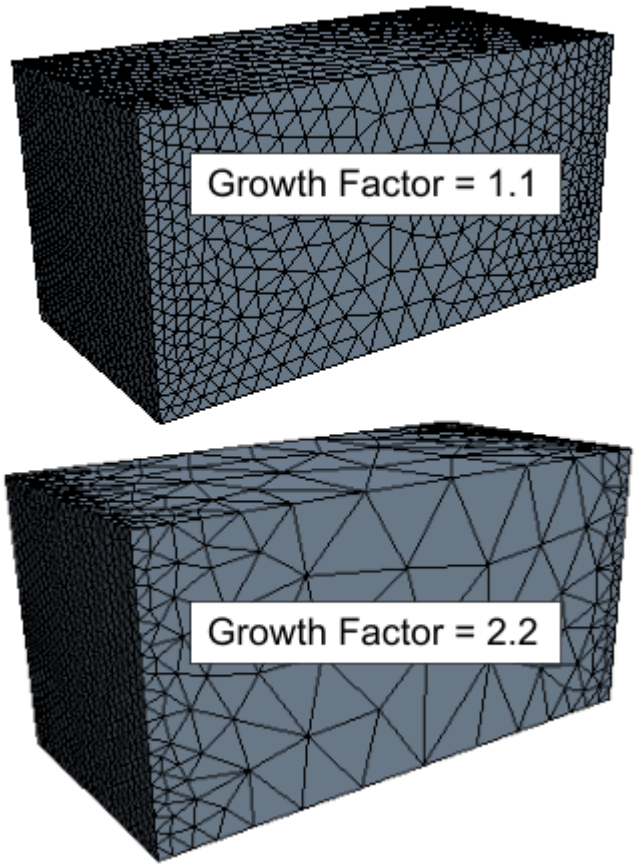
Property	Description
	
Number of Layers	Defines how many individual cell layers will be included in the overall prism layer, as illustrated below. 
Thickness Ratio	Specifies the size of the first and last layers within the prism layer. 

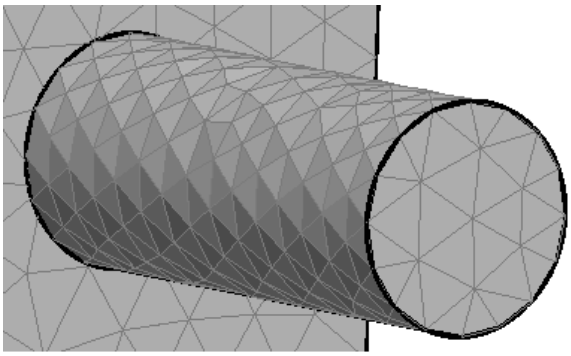
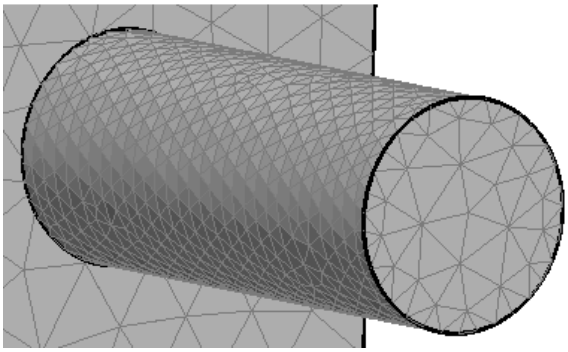
Mesh Pipeline Advanced Settings Panel

The advanced settings panel provides further customization of meshing parameters.

Property	Description
Relative Minimum Surface Size	Defines the minimum triangle size permitted on a feature curve. The minimum triangle size is set as a percentage of the surface

Property	Description
	mesh size. The default value is set to 25%. You must set the relative minimum surface size to a value greater than zero. If you set the relative minimum surface size to 100%, STAR-CCM+ will not modify the triangle size when it detects curvature. The figure below illustrates this behavior. Note that the base size was set to 20% in this example to emphasize the refinement on the curve.  Relative Minimum Surface Size = 25 Relative Minimum Surface Size = 100 Having a large value for the number of points on a circle property increases the mesh resolution on curved surfaces. The transition between the fine mesh and coarse mesh areas is controlled by the growth ratio.
Growth Ratio	Controls the rate at which triangle sizes change when moving from one triangle size to another. In the example below, the end faces of the block were set to have a smaller triangle size than the four side faces of the block. When the growth ratio is set to a low value, the transition between triangle size is very gradual. When a higher value is used, the transition is far more rapid.

Property	Description
	 The growth factor must be set to a value greater than 1.0. There is no upper limit for the growth factor.
Number of Points per Circle	Controls the number of triangle edges that appear on a curved surface, as shown below.

Property	Description
	 16 Points/Circle  32 Points/Circle Regardless of the number of points per circle defined, STAR-Inventor will not permit the triangle edge length to shrink below 25% of the base size. The curvature length scale may also affect the number of edges on a curved surface.

Trimmer Panel

The trimmer mesh is controlled using the Trimmer panel.

Property	Description
Default Growth Rate	Controls the stepping of the refinement from one cell size to the next. Five settings are available. By default, the *Default Growth Rate* is set to **Fast**, meaning that the fastest transition from the smallest cell size to the largest will be used. This usually equates to having only one layer of each cell size for each transition, which will result in the least number of cells being created for the supplied input values. The **Medium**, **Slow**, and **Very Slow** options further slow the transition from one cell size to the next in the mesh interior and therefore increase the overall cell count. In general, the medium option results in a minimum of two equal sized cell layers per transition, slow in four layers and the very slow option in eight layers.
Boundary Growth Rate	Controls the number of cell layers to grow next to surfaces in the template. Five settings are available. By default, the *Boundary Growth Rate* is set to **None**, meaning that the setting used for the Default Growth Rate is used

Property	Description
	for controlling all transitions. If the **Very Slow**, **Slow**, **Medium**, or **Fast** options are selected then it will result in eight, four, two or one layers of cells respectively of the same size being created next to the geometry surfaces in the template.

Working with Mesh Controls

Mesh controls allow you to define custom mesh settings for a particular section of the model.

Multiple controls can be assigned to the same item to control different mesh parameters. If, for example, a face has two mesh controls assigned to it--one that controls the mesh size and another that controls the prism layer thickness, the final mesh will respect both the custom mesh size and the custom prism layer thickness. If two mesh controls set the same mesh parameter, the control created last will take precedence over the first.

Contents:

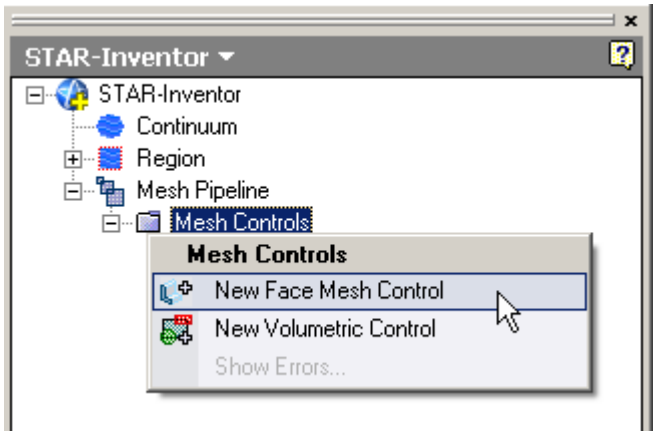
Using a Face Mesh Control
Using a Volumetric Mesh Control
Using a Part Mesh Control

Using a Face Mesh Control

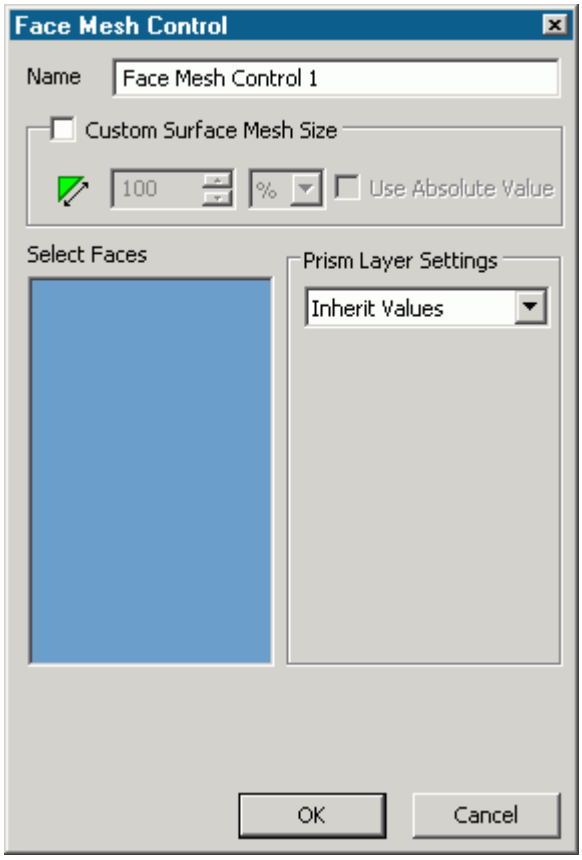
A face mesh control allows you to set the surface mesh size and the prism layer parameters on a face. Custom settings made on a face control will take priority over settings in the parent mesh pipeline, and over settings made in any part mesh controls.

To add a face mesh control:

1. Right-click the **Mesh Pipeline** > **Mesh Controls** node in the STAR-Inventor tree and select **New Face Mesh Control**.



The *Face Mesh Control* dialog appears.



2. Set *Name* to an appropriate value.
3. On the CAD geometry, select the desired faces to include in this mesh control.
4. If you want to specify a custom surface mesh size on the selected surfaces, activate the **Custom Surface Mesh Size** option.

   a) Set (*Custom Surface Mesh Size*) to the desired value.

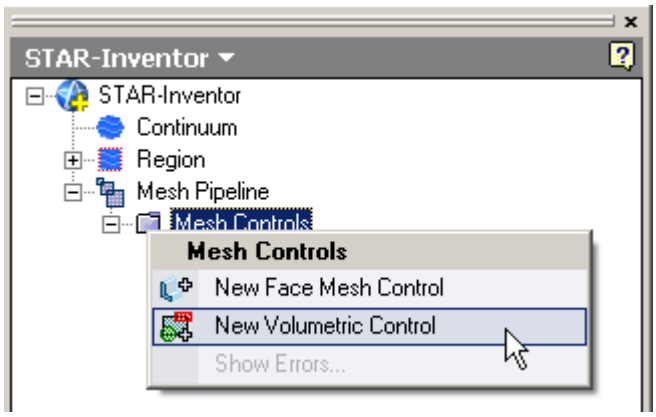
   You can enter this value as a percentage of the base size, or as an absolute value. If you select to enter the absolute value, you can select the units from the drop-down menu.
5. Set *Prism Layer Settings* to the desired value:
   - **Inherit Values**: in the first instance, the face(s) will be assigned values from any part mesh control applied to the part to which the face(s) belong. If this does not apply, values will be taken from any applicable volumetric mesh control, and in the absence of this values will be taken from the parent mesh pipeline.
   - **Use Custom Values**: allows the prism layer parameters to be set directly on the face(s). See Prism Layer Panel.
   - **Disable**: no prism layer will be generated on the face(s).
6. Click **OK**.

Using a Volumetric Mesh Control

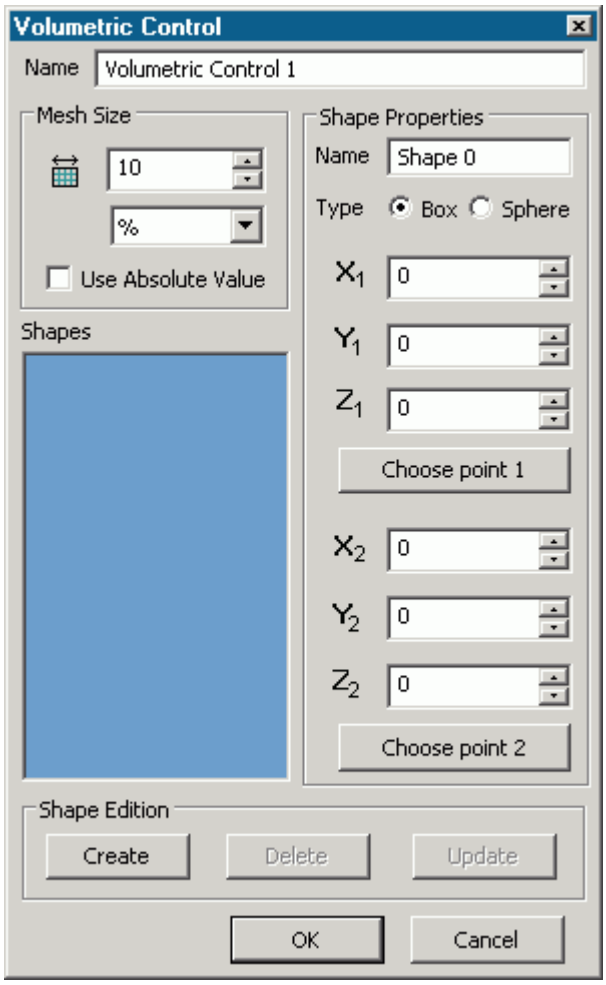
A volumetric control permits mesh refinement within a user-defined box or sphere. Only the mesh size can be modified with a volumetric control; prism layer parameters are not affected.

To add a volumetric mesh control:

1. Right-click the **Mesh Pipeline** > **Mesh Controls** node in the STAR-Inventor tree and select **New Volumetric Control**.



The *Volumetric Control* dialog appears.



2. Set *Name* to an appropriate value.

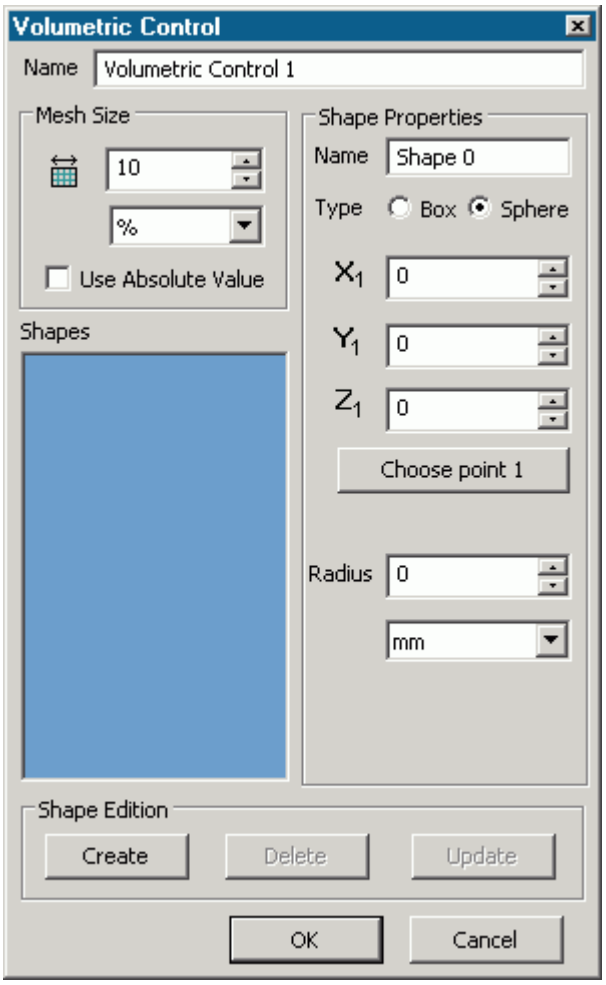
3. Set *Custom Surface Mesh Size* to the desired value.

   The mesh size defines the triangle edge length for surface mesh within the volume shape, and the characteristic cell size for volume mesh. The custom mesh size will still affect the volume mesh in cases where the volume shape does not intersect any surfaces of the Pro/E geometry.

   The mesh size can either be defined relative to the base size of the parent mesh pipeline, or as an absolute dimension with length units.

4. Select the type of volumetric control to create:
   - **Box**
   - **Sphere**
5. Enter the parameters for the volumetric shape:
   - If you are creating a box, enter the coordinates for Point 1 and Point 2.

     X, Y, and Z coordinates of the two opposite corners of the box are required by the panel. These coordinates can either be entered manually, or taken from existing points within the Pro/E window. To use an existing point, click **Choose Point 1** or **Choose Point 2** and use the Inventor point selection operation to select the appropriate point.
   - If you are creating a sphere, enter the coordinates for the center point and the radius of the sphere.



The sphere shape is defined by a center point and a radius. The center point can be set either by entering the coordinates directly, or by clicking the **Choose Point** button and using the Pro/E point selection procedure to choose an existing point in the Pro/E window. The coordinates of the existing point will be used to populate the panel.

6. Click **Create**.

   The shape is added to the *Shapes* list.

7. Click **OK**.

Editing a Volume Shape

To edit a volume shape

1. Select the volume shape from the list within the **Shapes** panel.
2. Edit the parameters as required and click **Update**.

Deleting a Volume Shape

To delete a volume shape

- Select the volume shape from the list in the *Shape* panel and click **Delete** .

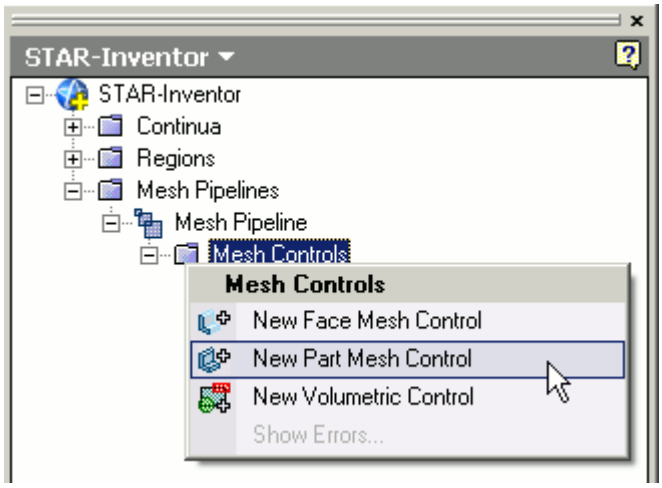
Using a Part Mesh Control

Custom mesh settings (such as surface mesh size and prism layer parameters) can be applied to one or more parts using a part mesh control.

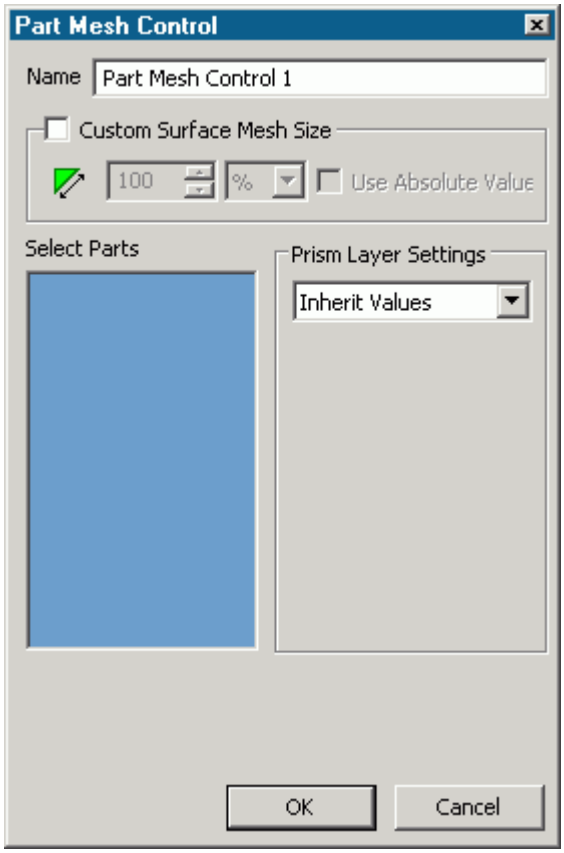
Settings made on a part mesh control will take priority over settings made on the parent mesh pipeline, but will not have priority over settings made on a face mesh control.

To add a part mesh control:

1. Right-click on **Mesh Pipelines** > **Mesh Pipeline** > **Mesh Controls** in the STAR-Inventor tree and select **New Part Mesh Control**.



   The *Part Mesh Control* dialog appears.



2. Set *Name* to an appropriate value.
3. On the CAD geometry, select the desired parts to include in this mesh control.
4. If you want to specify a custom surface mesh size on the selected surfaces, activate the **Custom Surface Mesh Size** option.

   a) Set (*Custom Surface Mesh Size*) to the desired value.

   You can enter this value as a percentage of the base size, or as an absolute value. If you select to enter the absolute value, you can select the units from the drop-down menu.
5. Set *Prism Layer Settings* to the desired value:
   - **Inherit Values**: for this option, prism layers on the surfaces of the part(s) will be generated based on settings taken from the parent mesh pipeline.
   - **Use Custom Values**: allows the prism layer parameters to be set directly on the part(s). See Prism Layer Panel.
   - **Disable**: no prism layer will be generated from the surfaces of the selected parts.
6. Click **OK**.

Generating the Mesh STAR-Inventor

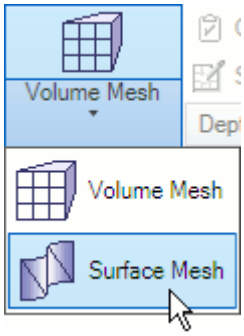
It is recommended that you first generate the surface mesh and inspect its quality before generating the volume mesh. Problems in the surface mesh will have a definite impact on the volume mesh, and may result in the volume mesher failing to complete.

Mesh generation will be carried out automatically when you issue the Build and Run command.

Generating the Surface Mesh in STAR-Inventor

To generate the surface mesh:

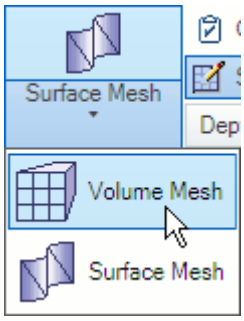
- From the STAR-Inventor ribbon, click ▾ next to the **Volume Mesh** icon and select **Surface Mesh**.



Generating the Volume Mesh in STAR-Inventor

To generate the volume mesh:

- From the STAR-Inventor ribbon, click ▾ next to the **Surface Mesh** icon and select **Volume Mesh**.

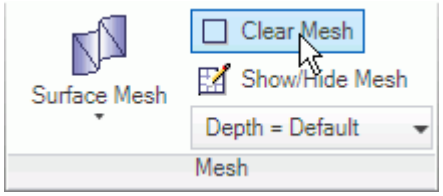


Clearing Generated Meshes

The **Clear Generated Meshes** action deletes all generated meshes without affecting the settings.

To clear the mesh:

- From the STAR-Inventor ribbon, click **Volume Mesh**.



Contents:

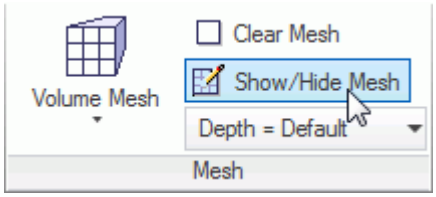
Visualizing the Mesh
Controlling the Mesh Depth
Meshing Behavior at Interfaces

Visualizing the Mesh

Surface and volume meshes will automatically be displayed on first creation. If required, they can be hidden from view.

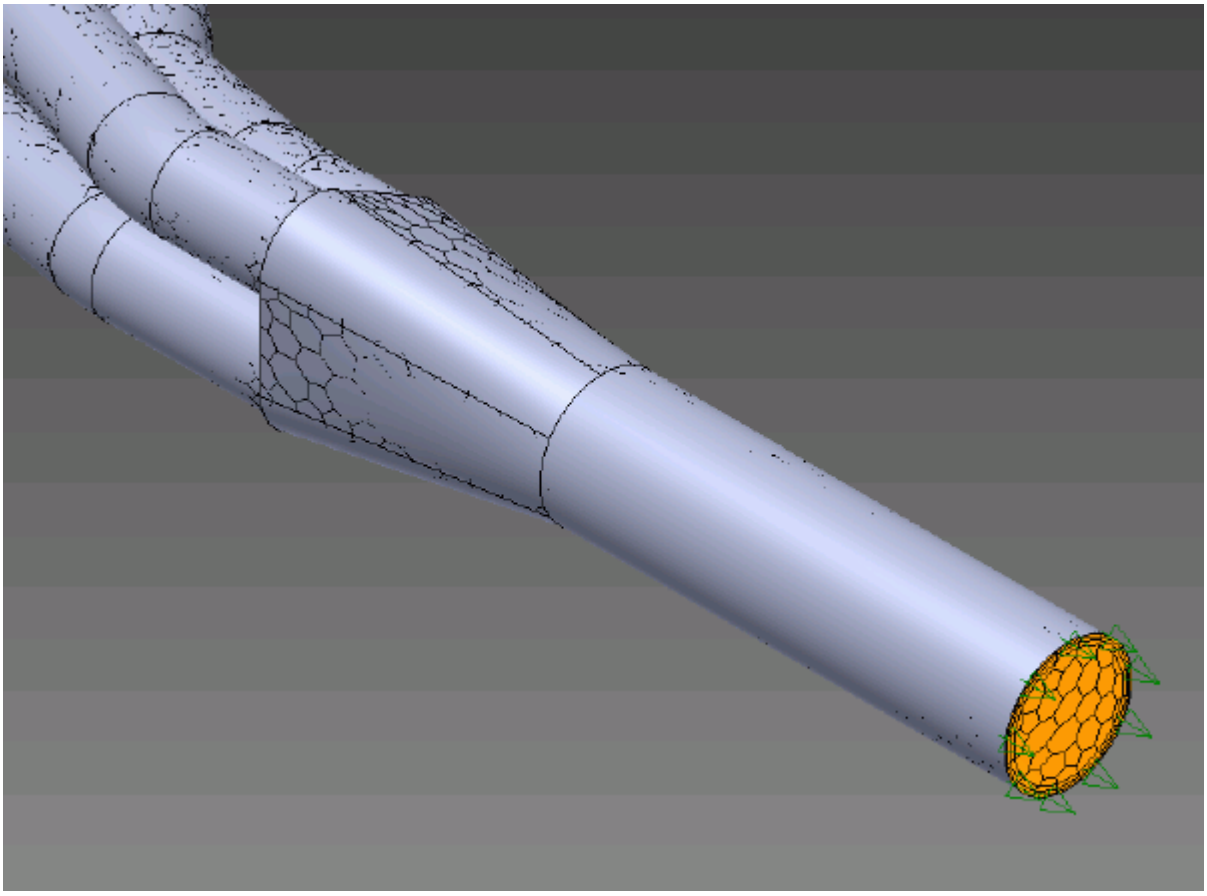
To hide a generated mesh in the geometry view:

1.

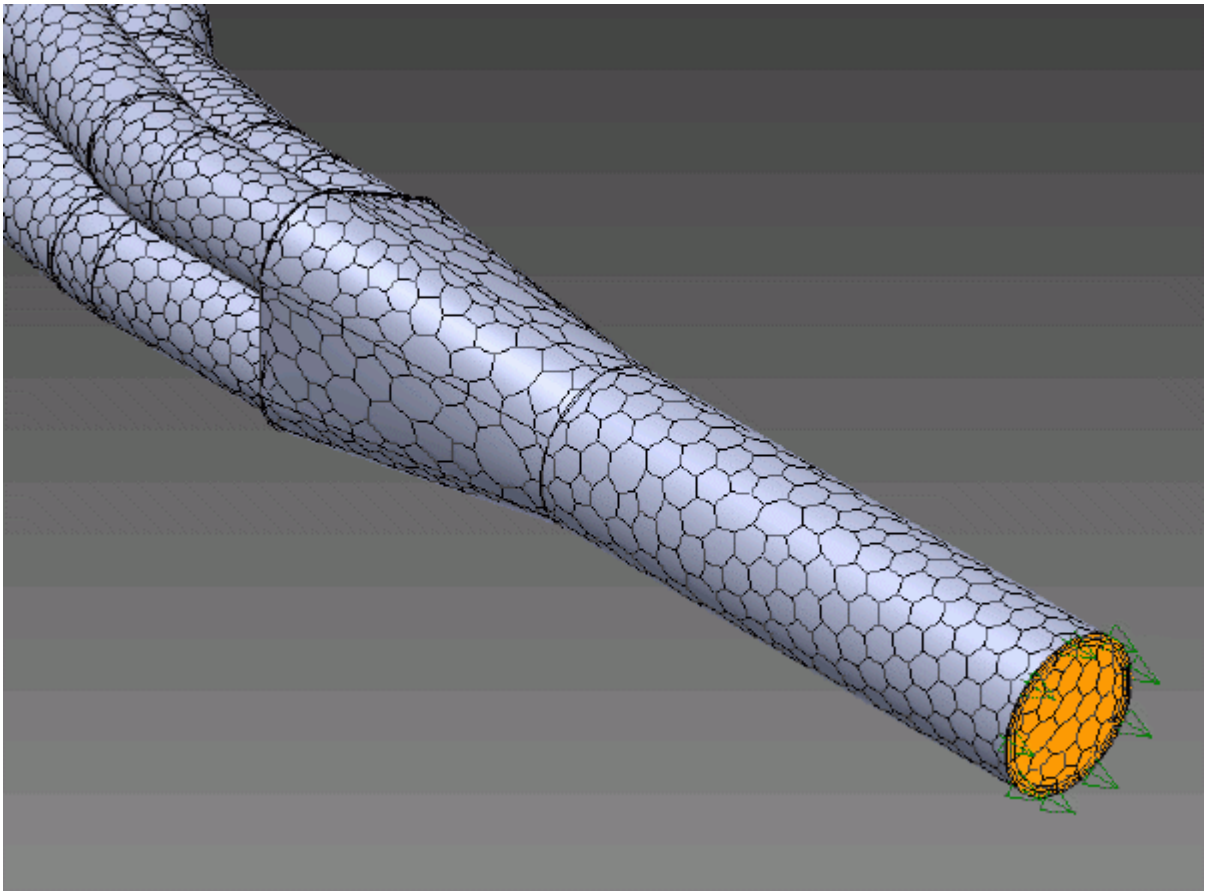


Controlling the Mesh Depth

The Depth option controls the mesh visualization (mesh depth). If the mesh appears to be hidden behind the geometry surface and is hard to see (particularly when zooming in closer, as shown in the screenshot below), changing the mesh Depth option may help improve mesh visualization.

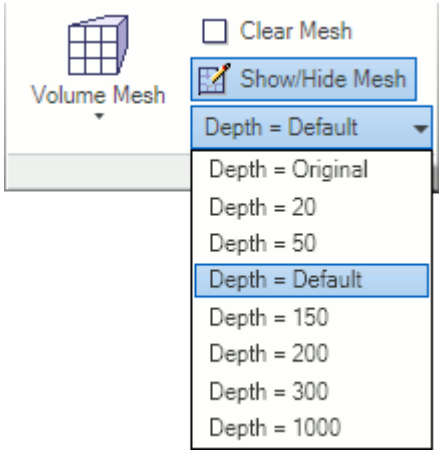


Increasing the Depth value will improve the mesh definition.



To change the depth of the mesh:

1. Click **Depth = Default** in the STAR-Inventor ribbon and select the desired depth from the drop-down list.



The higher the number the more pronounced the mesh will be, however increasing this value too much will cause the mesh from behind the geometry to start appearing (particularly near the edges). For thin geometries, a high depth number should be avoided as the entire mesh will be visible, including the mesh from behind the geometry.

The default setting (**Depth = Default**) should work well with most geometries. The original Pro/E value is stored in the **Depth = Original** setting--this tends to hide the mesh when zooming in and is not recommended for most cases.

Meshing Behavior at Interfaces

The volume mesher will treat interfaces between two regions differently depending on what type of region (fluid or solid) these are.

If the interface lies between two fluid regions, no prism layers will be grown from both sides of the interface. If an interface lies between a solid region and a fluid region, prism layers will be grown from the interface. The default behavior of the prism layer generation can be overridden using a mesh control, so prism layers can be disabled on a solid-fluid interface. See Working With Mesh Controls.

See Also:

Working with Mesh Controls

Common Causes for Meshing Failures

There may be instances when the mesh generation fails. In most cases this will occur during the surface meshing phase. STAR-Inventor will try and identify the cause and suggest ways of rectifying this. This information will be printed in the *Output* tab of the STAR-Inventor Info App window.

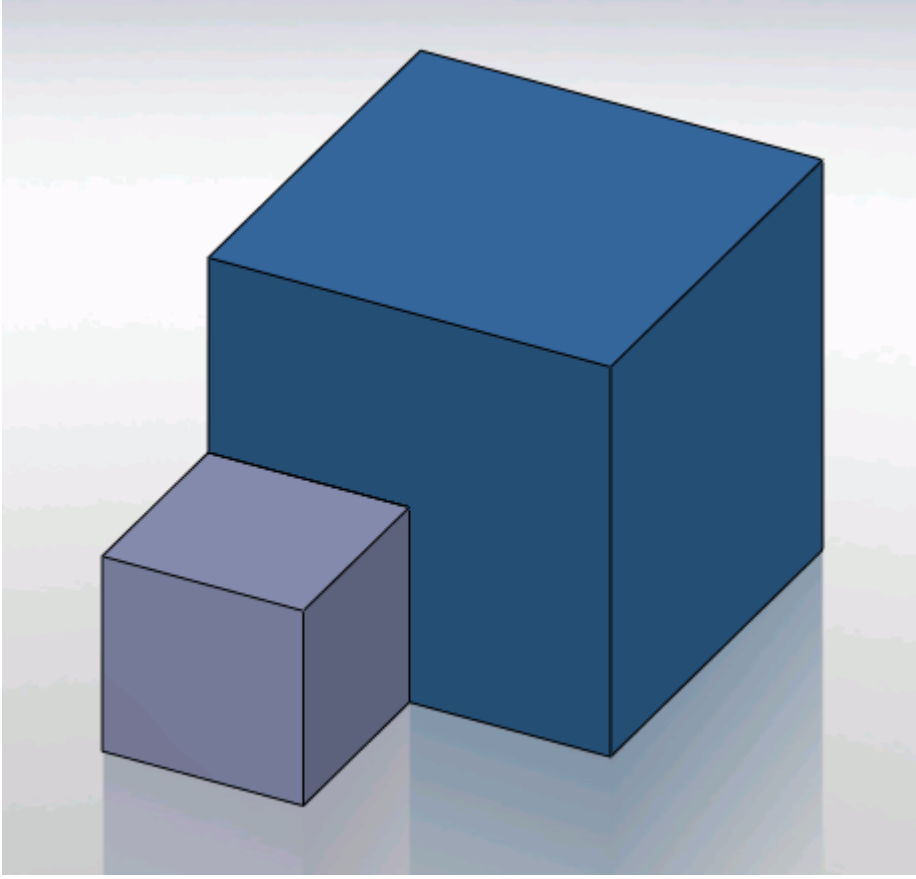
Contents:

The Parts in an Assembly are not Imprinted
Faces May Require Splitting

The Parts in an Assembly are not Imprinted

The parts in an assembly should be imprinted together so that there is a common contact area between them.

A simple example of when imprinting is required is shown below.

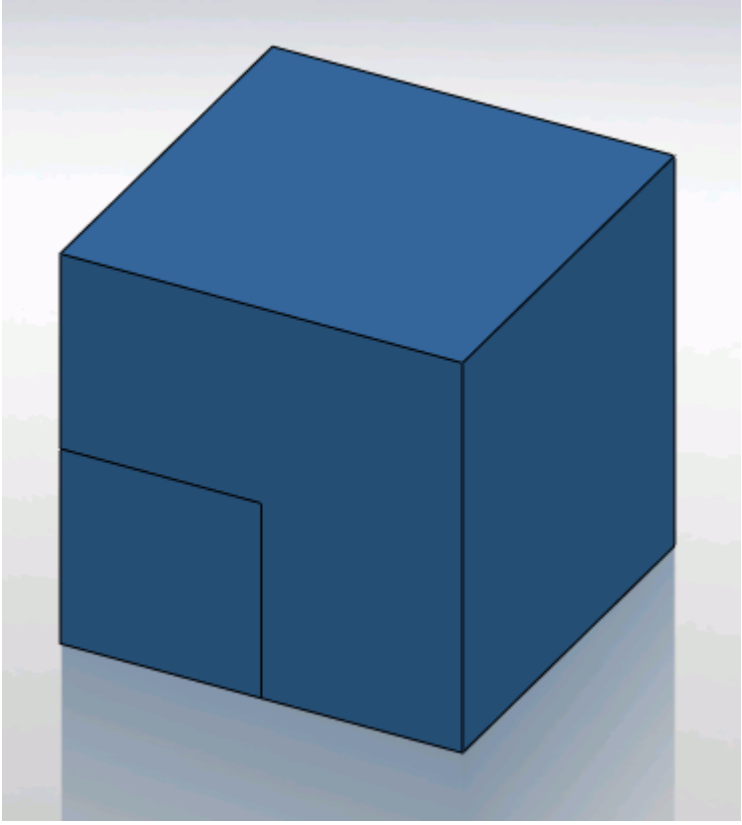


In this case the face of Part A (grey box) is splitting the face of Part B (blue box) causing this assembly to fail to mesh.

The output will display an error message.

```
Initializing meshing pipeline...
  All Geometry up to date.
Executing surface meshers...
Preparing surface for surface meshing in regions: Region 1, Region
Error during surface remeshing, deleting resulting mesh
An error occurred while attempting to orient the surface.
This is usually due to self-intersections in the surface. Check for problem(s) at :
  Geometric Range : (-0.0225836 0.00152109 0.0632873) and (0.0501103 0.0703362 0.0647127)
  Centroid of face : (0.0256414 0.0471603 0.064)
  Face type of face : 6
  Boundary = Default.
  Region = Region 1
Command: GenerateSurfaceMesh
   CompletedCommand: GenerateSurfaceMesh
   In: [Machine::main]
   Slave Message:
      Slave Name: Proc[0]
      Slave Rank: 0
   error: Server Error
```

The surface of Part B has to be imprinted with the contact face of Part A as shown below, after which meshing will complete successfully.



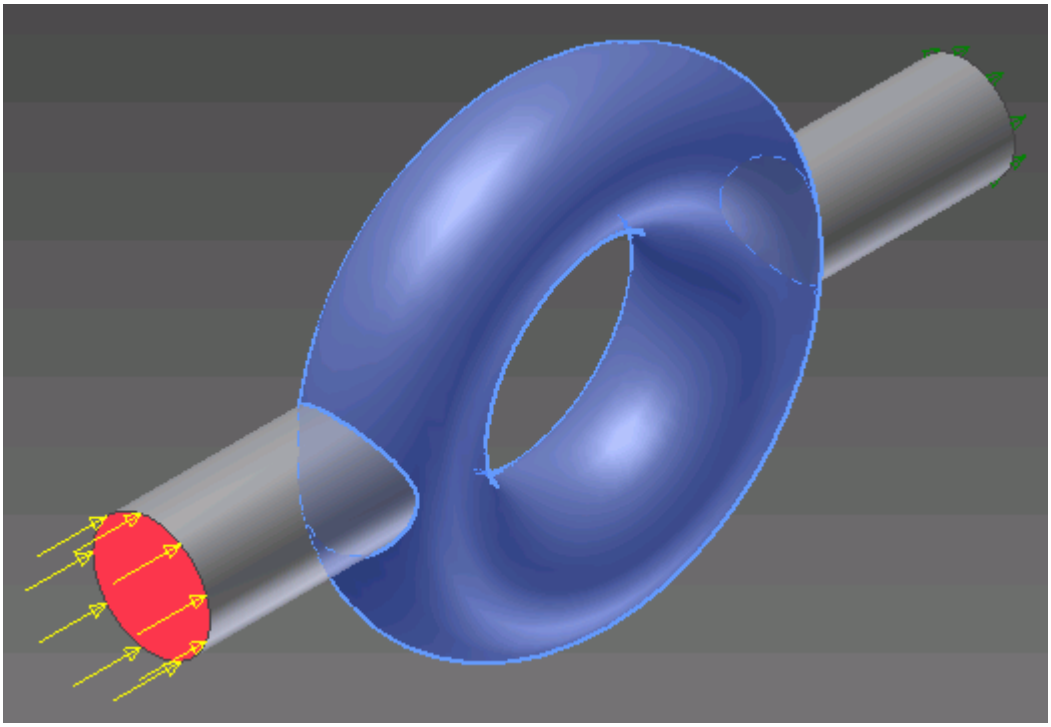
Faces May Require Splitting

Enter a short description of your concept here.

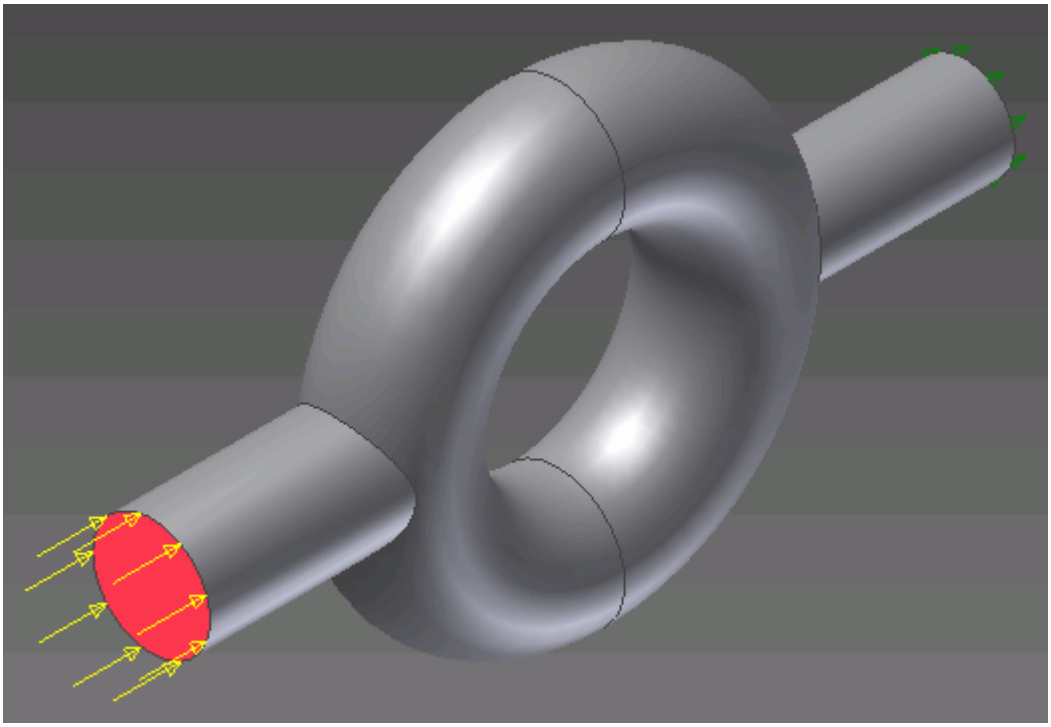
In some cases specific geometric shapes may not be able to mesh in STAR-Pro/E, and an error message such as the one shown below will be displayed in the output window.

```
Creating Surface Mesh
Estimated Cell count = 795 Global scale fact = 1
Surfaces repaired successfully
Preparing surfaces
Meshing surfaces
Could not orient the edges of inner trimming loop 1 of surface 5.
     Winding angle should be 360 degrees, calculated 2.66759e-006 degrees
1 surfaces did not close properly

The faulty faces are highlighted.
Surface Mesh Error: -
49 bad edges detected. Surface Meshing Error
The surfaces 1 3 4 containing bad edges are highlighted.
```

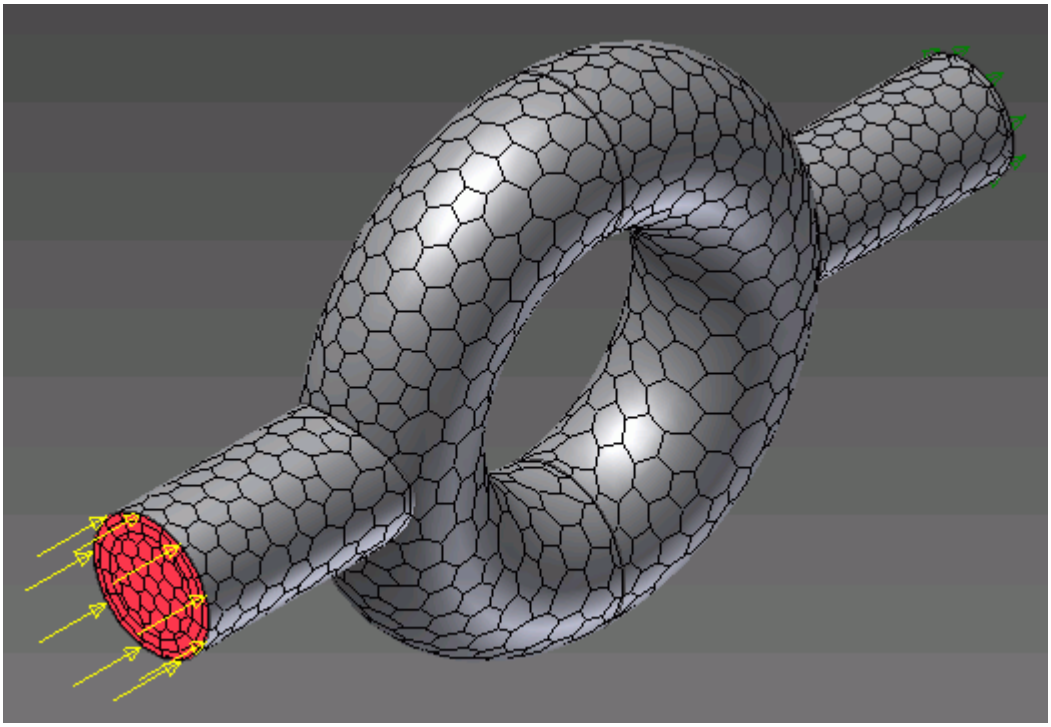
The face that is causing the problem will be highlighted in the geometry window.



This issue can be overcome by splitting the highlighted face in Pro/E - the simplest way of achieving this is by creating a plane that dissects the aforementioned face and use it as the splitting tool. The result is shown below.



A mesh can now be generated.



Obtaining a Solution

The following section details the execution of the STAR-CCM+ solvers in order to obtain a solution.

Contents:

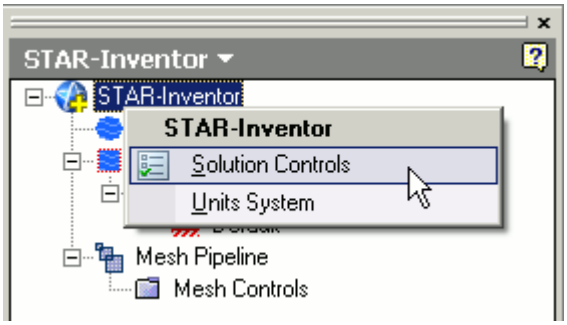
Setting Solution Controls
Checking the Model for Errors
Building and Running the Simulation in STAR-Inventor
Clearing the Solution
Visualizing the Solution

Setting Solution Controls

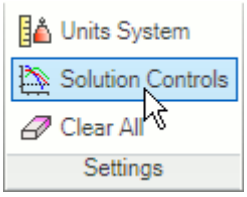
The solution you obtain from the solvers is a compromise between accuracy and processing time. You can manage the speed and accuracy of the solver from the *Solution Controls* dialog. The solution accuracy is also dependant on the mesh density.

To set up the solution controls in STAR-Inventor:

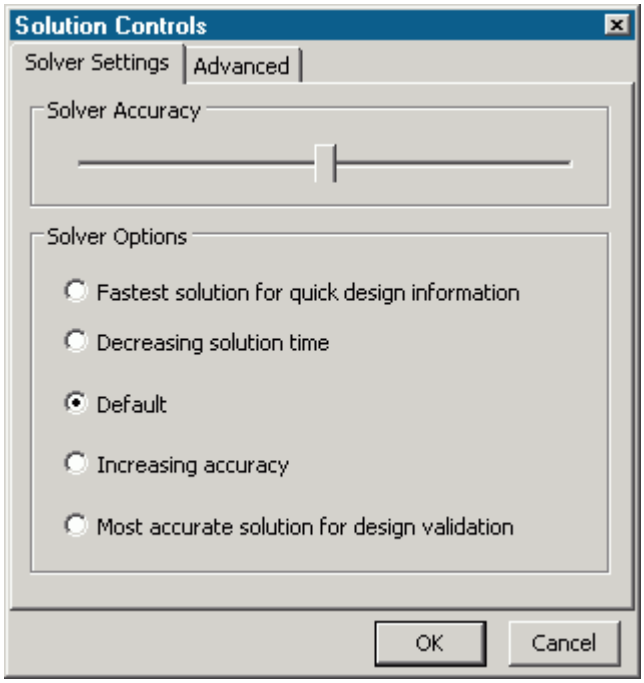
1. Either:
   - Right-click on the **STAR-Inventor** node within the object tree and select **Solution Controls**.



   - From the STAR-Inventor ribbon, select **Solution Controls**.



The *Solution Controls* dialog appears.



2. Set the solution control parameters:
   - Using the slider, select one of the five pre-defined solver options. These options range from high accuracy and low speed, to low accuracy and high speed.
   - Click the **Advanced** tab and set the solution control parameters manually. See Solution Controls Advanced Settings Panel.
3. If you want to run a Java macro with the simulation, activate the **Use Java File** option. See Using Java Macros with STAR-Inventor.
4. Click **OK**.

Contents:

Solution Controls Advanced Settings Panel
Using Java Macros in STAR-Inventor

Solution Controls Advanced Settings Panel

The advanced settings in the Solution Control dialog allow you to manually enter the values for the solution control parameters.

Parameter	Description
Stopping Criteria	
(*Number of Iterations*)	Specifies the maximum number of iterations for which the solver can run. When the STAR-CCM+ solver reaches the specified number of iterations, it will stop. You can then post-process the results in STAR-CCM+. See Convergence Levels.

Parameter	Description
(*Convergence Level*)	Specifies how many orders of magnitude the residuals should drop before the solver stops. See Convergence Levels.
Solution Accuracy	
First Order	This option is generally the quicker, but less accurate solver option. The first order scheme introduces a dissipative error that is stabilizing and helps the solver achieve robust convergence. However, it can also lead to the neglect and merging of discontinuities, meaning the solution is not fully captured.
Second Order	This option is generally the more accurate, but slower solver option. The second order scheme attempts to minimize both dispersive and dissipative errors, providing a better representation of the simulation. However, in some situations, the lack of numerical dissipation can result in poorer convergence properties. It is common practice to use the first order scheme initially to ensure the simulation is running as expected and then switch to second order.
Under-Relaxation Factors	
(*Momentum Under-Relaxation Factor*)	Specifies the under-relaxation factor for the momentum equation. See Solution Controls Advanced Settings Panel.
(*Pressure Under-Relaxation Factor*)	Specifies the under-relaxation factor for the pressure-correction equation. See Solution Controls Advanced Settings Panel.
(*Turbulence Under-Relaxation Factor*)	Specifies the under-relaxation factor for the turbulence equation. See Solution Controls Advanced Settings Panel.
(*Fluid Temperature Under-Relaxation Factor*)	Specifies the fluid under-relaxation factor for the segregated energy solver. See Solution Controls Advanced Settings Panel.
(*Solid Temperature Under-Relaxation Factor*)	Specifies the solid under-relaxation factor for the segregated energy solver. See Solution Controls Advanced Settings Panel.

Convergence Level

The numerical method used by the solver requires an iterative process in order to obtain a solution. After each iteration, residuals are produced that indicate how well the governing equations for each solver quantity are

being satisfied numerically. In general, the residual magnitude should decrease as the solution converges. When the magnitude of the residuals for all the quantities falls below the convergence level, the solver will stop iterating, and results will be exported for post-processing.

Note: In most cases, a drop of 3-5 orders of magnitude in the residuals is sufficient to obtain an accurate solution.

You can adjust the convergence level parameter to meet accuracy or process time requirements. However, if the residuals are no longer decreasing from one iteration to the next, it is likely that machine accuracy has been reached for that case. For a case that does not exhibit decreasing residuals, it would be necessary to reconsider the setup in case the volume mesh is too coarse, or boundary conditions have been set incorrectly.

Under-Relaxation Parameters

Under-relaxation parameters determine how quickly the solver reaches a solution for a particular equation, on a scale from 0 to 1, with 1 being the fastest.

Ideally, all under-relaxation parameters would be set to 1, however, doing so would likely cause solver instability due to the complex nature of most simulations. Hence the under-relaxation parameters should be set to values that compromise speed with stability.

Note: Adjust the under-relaxation parameters if changes to other solution controls such as *Number of Iterations*, *Convergence Levels*, and *Solution Accuracy* have failed to improve the behavior of the solver.

Using Java Macros in STAR-Inventor

You can use Java macros with STAR-CCM+ to automate your simulation or to change a setting in STAR-CCM+, such as selecting an alternative turbulence model, before the solver starts running.

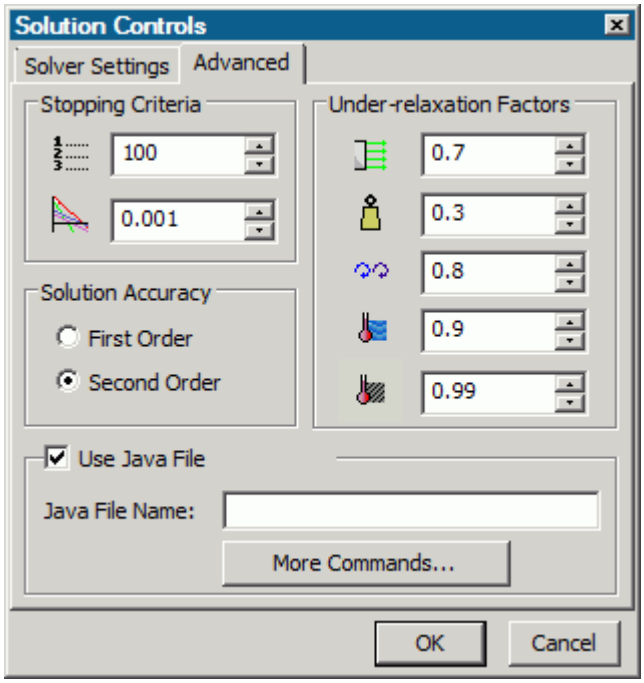
An example of how to change a setting in STAR-CCM+ using a Java macro is outlined below:

1. Launch STAR-Inventor and set up the case.

   Define the continuum settings, boundary conditions, mesh settings, and solution control settings.
2. Launch STAR-CCM+.
3. In STAR-CCM+, start recording a macro and perform the desired operations, such as changing the physics models.
4. Stop recording the macro.

   The macro saves at this point.
5. Close STAR-CCM+ and do not save the simulation.
6. Return to STAR-Inventor, open the *Solution Controls* dialog and activate the *Use Java File* checkbox.

   A text entry field appears.



7. Click **More Commands...**.

   The *Open* dialog appears.

8. Navigate to your Java macro file and select it.

   Selected files appear in the text entry field. The path in the entry field is displayed relative to the current CAD geometry file.

9. Click **OK**.

When you run the simulation, STAR-CCM+ will launch and play the macro before starting the solver. This sequence allows you to set up and use additional STAR-CCM+ physics models that are not available in STAR-Inventor.

See Also:

Setting Up Continua

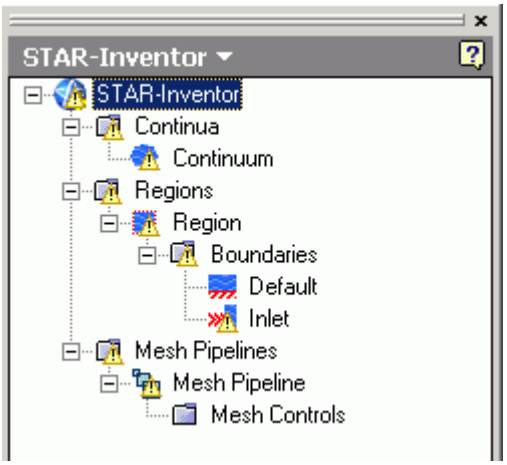
Setting Up Regions

Setting Boundary Conditions

Setting Up the Mesh Pipeline

Checking the Model for Errors

STAR-Inventor monitors the active case for any errors in real-time and displays a warning triangle ⚠ next to any nodes for which there is a problem.

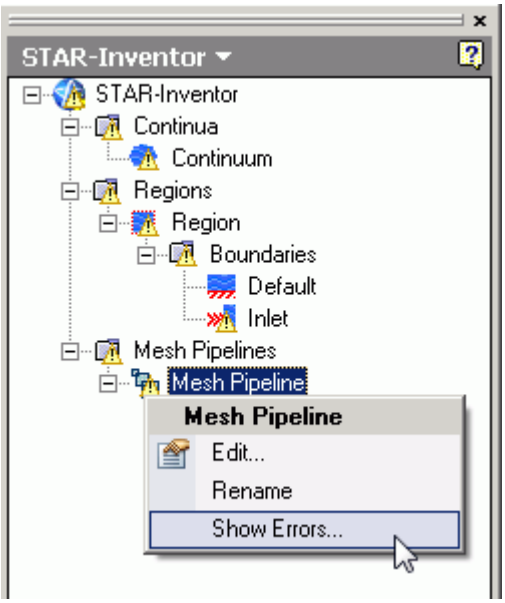


Three areas are monitored by the live model checker:

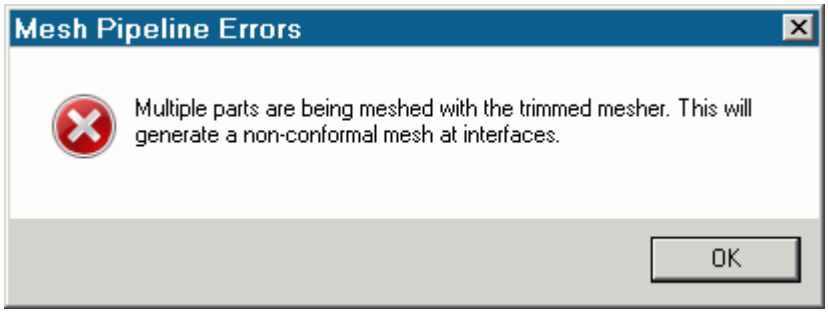
- **Continua**. The continua checker makes sure that all necessary boundaries are present in the case, such as the presence of at least one inlet and one outlet.
- **Regions and boundaries**. The regions and boundaries checker covers issues specific to boundaries, such as a velocity inlet boundary having a zero value.
- **Mesh pipelines**. The mesh pipeline checker covers issues that are related to the mesh, such as the presence of multiple parts in a trimmed mesh pipeline.

To see the cause of an error:

1. Right-click on an node displaying the warning triangle, and select **Show Errors...**



A dialog appears describing the error, as shown below.



2. Click **OK** to close the dialog.

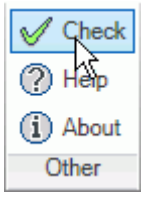
The **Show Errors...** right-click menu is available for any node in the STAR-Inventor object tree that contains a warning triangle. Selecting it will show errors for that node and its sub-nodes.

To see a complete list of issues for the case, right-click on the STAR-Inventor root node and select **Show Errors...**

You can also manually check for errors.

To perform a manual check:

- From the STAR-Inventor ribbon, click t **Check**.



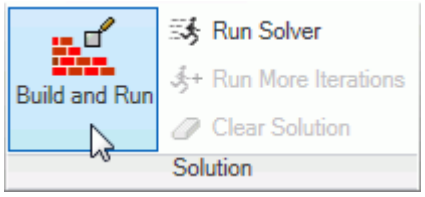
If errors are found in the case set up, these are displayed in the *Model Checker* tab of the *Info App* dialog. See The Model Checker Tab.

Building and Running the Simulation in STAR-Inventor

Once you have defined the model physics, boundary conditions, mesh settings, and solver settings in STAR-Inventor, you can build and run the case.

To build and run the case:

- From the STAR-Inventor ribbon, click **Build and Run**.



This action automatically generates the mesh, and runs the solver. When you run the solver, a new *Residuals* tab is added to the *Info App* dialog. You can use this residuals plot to monitor solution convergence. When the solver finishes, STAR-CCM+ launches and a scalar scene is automatically created for you.

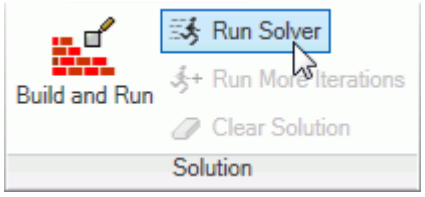
Note: If a solution has already been computed in STAR-CCM+, and the stopping criteria has been reached, clicking **Build and Run** clears the solution and starts running from the first iteration. If you want to run for more iterations, change the stopping criteria and click **Run Solver**, or click **Run More Iterations**.

Running the Simulation in STAR-Inventor

If you have already generated a volume mesh for the model, you can run the case directly.

To run the solver:

- From the STAR-Inventor ribbon, click **Run Solver**.

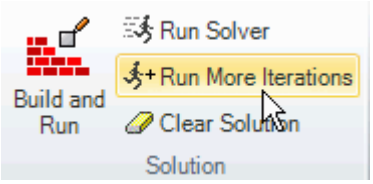


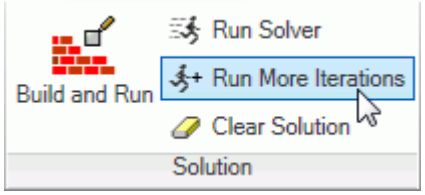
Running for More Iterations

You may need to run the STAR-CCM+ solvers for more iterations than you originally set in the Number of Iterations stopping criterion. The Run More Iteration function runs the solver for an additional number of iterations as specified in the stopping criteria. Take, for example, a case where you have already run the simulation for 300 iterations. You specify 500 iterations in the *Number of Iterations* stopping criterion. When you next click **Run for More Iterations**, the simulation will run for 500 extra iterations, making 800 iterations in total. The solver runs until one of the stopping criteria is satisfied.

To run for more iterations:

- From the STAR-Inventor ribbon, click **Run For More Iterations**.





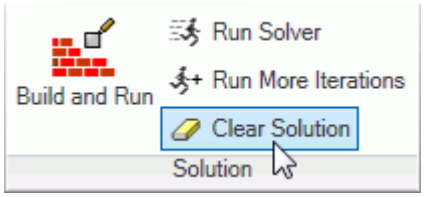
Clearing the Solution

You can clear all results that were calculated by the STAR-CCM+ solvers using the **Clear Solution** function. All other STAR-Inventor simulation settings remain intact.

Caution: This action is irreversible.

To clear the solution:

1. From the STAR-Inventor ribbon, select **Clear Solution**.



The solution that was calculated by the STAR-CCM+ solvers is deleted.

Visualizing the Solution

When STAR-CCM+ launches after the solution is calculated, it automatically displays a scalar scene of the model that shows the `Absolute Pressure` field function.

To visualize the results, you can use this scene to display different scalar quantities, or use any of the other visualization tools in STAR-CCM+, such as streamlines, plane sections and vector scenes.

You can import CAD parts into STAR-CCM+ for visualization purposes. These parts will not be taken into account during the analysis. This feature is useful for displaying the original geometry in a post-processing scene in STAR-CCM+. See Importing Geometry Into STAR-CCM+ for Visualization.

Exporting Files and Data

This section details the process for exporting files and data from STAR-Inventor.

Contents:

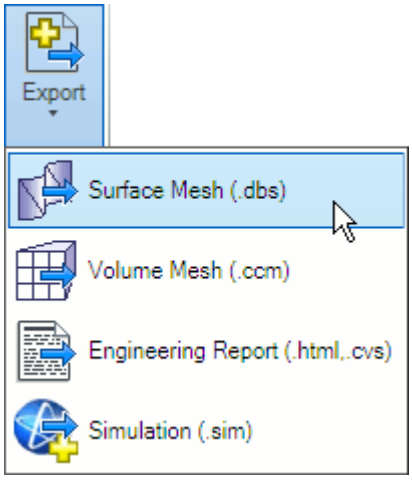
Exporting the Surface Mesh
Exporting the Volume Mesh
Exporting an Engineering Report
Exporting a STAR-CCM+ Simulation

Exporting the Surface Mesh

You can export the surface mesh generated in STAR-Inventor as a native CD-adapco surface mesh (`.dbs`) file. The `.dbs` file format is the recommended format to use when transferring surface data between CD-adapco programs. The boundary names and boundary types are retained by the file, as well as the interfaces that were automatically created between regions.

To export the surface mesh:

1. From the STAR-Inventor ribbon, click **Export > Surface Mesh (.dbs)**.

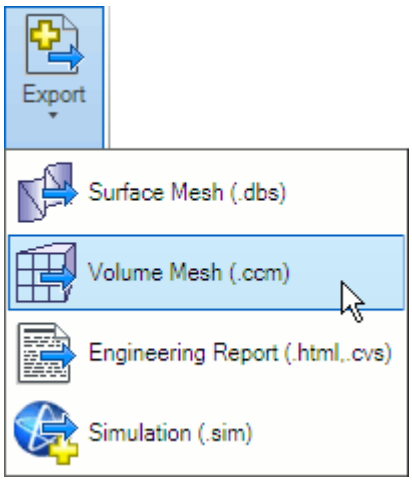


Exporting the Volume Mesh

You can export the volume mesh that you generated in STAR-Inventor as a native CD-adapco volume mesh (`.ccm`) file. The `.ccm` file format is the recommended format to use when transferring volume meshes between CD-adapco programs. The region and boundary names, and region and boundary types are retained in this file. The interfaces that were automatically created interfaces between regions are also retained in this file.

To export the volume mesh:

1. From the STAR-Inventor ribbon, click **Export** > **Volume Mesh (.ccm)**.



Exporting an Engineering Report

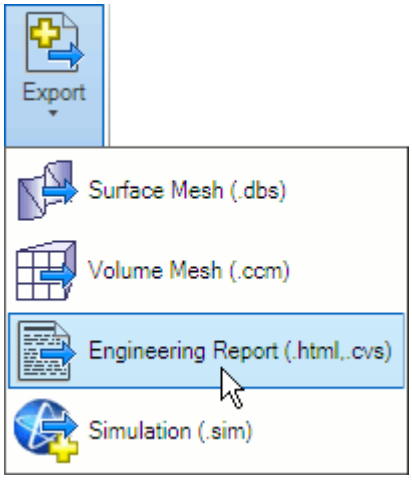
You can export a summary of the simulation settings and results in HTML format to two individual files:

- The first will be given a user-specified file name, and contains results from the simulation.
- The second will be given a user-specified file name appended with `_caseinfo`, and contains details on the simulation set-up.

Note: You can only export an engineering report after you obtain a solution.

To export an engineering report:

1. From the STAR-Inventor ribbon, click **Export** > **Engineering Report (.html,.csv)**.



Results Report

The following is a summary of the information included in the results report (`<filename>.htm`). This data is provided in the report with reference to each material region in the simulation:

- Case name and time-stamp
- Solution information: CPU time, Elapsed Time and Number of Iterations
- Mass flow data, at each boundary
- Mass flux data, at each boundary
- Area averaged boundary data, for the following scalars:
  - Pressure (Pa)
  - Total Pressure (Pa)
  - Pressure Coefficient
  - Total Pressure Coefficient
  - Turbulent Kinetic Energy (J/kg)
  - Turbulent Dissipation Rate (m2/s3)
  - Turbulent Viscosity (Pa-s)
  - Density (kg/m3)
- Mass flow averaged boundary data, for the following scalars:
  - Pressure (Pa)
  - Total Pressure (Pa)
  - Pressure Coefficient
  - Total Pressure Coefficient
  - Turbulent Kinetic Energy (J/kg)
  - Turbulent Dissipation Rate (m2/s3)
  - Turbulent Viscosity (Pa-s)
  - Density (kg/m3)
- Force data, at each boundary:
  - Net Force [Fx, Fy, Fz]
  - Pressure Force [Fx, Fy, Fz]
  - Shear Force [Fx, Fy, Fz]
- Force coefficient data, at each boundary:
  - Coefficient of Net Force [Cfx, Cfy, Cfz]
  - Coefficient of Pressure Force [Cfx, Cfy, Cfz]
  - Coefficient of Shear Force [Cfx, Cfy, Cfz]

Setup Report

The following is a summary of the information found in the simulation setup report (`<filename>_caseinfo.htm`):

- The parts used in the simulation, and their associated regions
- The surfaces for each part, and their associated boundaries
- The continua used in the simulation, with the following information:
  - Physics models applied to the continuum
  - Materials properties
  - Turbulence properties

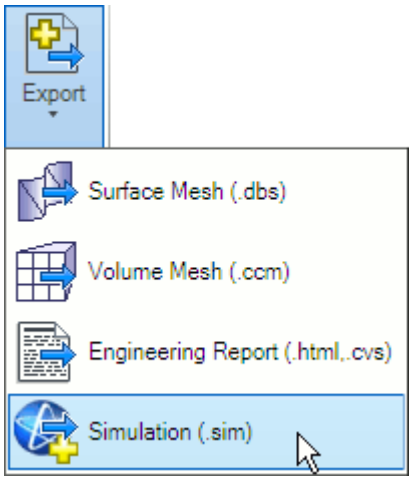
  - Segregated flow properties
  - Reference values
  - Initial conditions
- The mesh pipelines used in the simulation, with the following information:
  - Polyhedral mesher properties
  - Prism layer mesher properties (if activated)
  - Surface remesher properties
  - Reference values
  - Volumetric controls
- The regions used in the simulation, and their associated continua and mesh pipeline
- The boundaries for each continua, with the applied conditions and values
- Solver parameters
- Stopping criteria
- Units

Exporting a STAR-CCM+ Simulation

You can export the current simulation in STAR-Inventor to the native STAR-CCM+ simulation (`.sim`) file format. The `.sim` file format retains all simulation settings defined in STAR-Inventor, as well as the surface and volume meshes.

To export a simulation file:

1. From the STAR-Inventor ribbon, click **Export** > **Simulation (.sim)**.



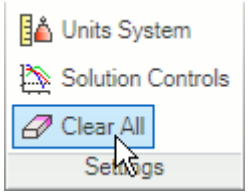
Resetting STAR-Inventor Settings

STAR-Inventor saves all case setup information to the CAD file. This means that all continuum settings, boundary settings, solution control settings, and mesh settings, are stored within the CAD geometry file. You can remove this information from the CAD geometry file.

Caution: This action is irreversible.

To clear all STAR-Inventor settings:

1. From the STAR-Inventor ribbon, click **Clear All**.



Guidelines

This section aims to provide guidance and assistance on making a geometry part suitable for CFD analysis using STAR-Inventor.

Contents:

Creating the Fluid Domain
Imprinting Parts

Creating the Fluid Domain

In the majority of CAD models, the fluid domain for analysis must be extracted from the original part or assembly. The topics that are containing within this section demonstrate how to prepare the fluid domain.

Contents:

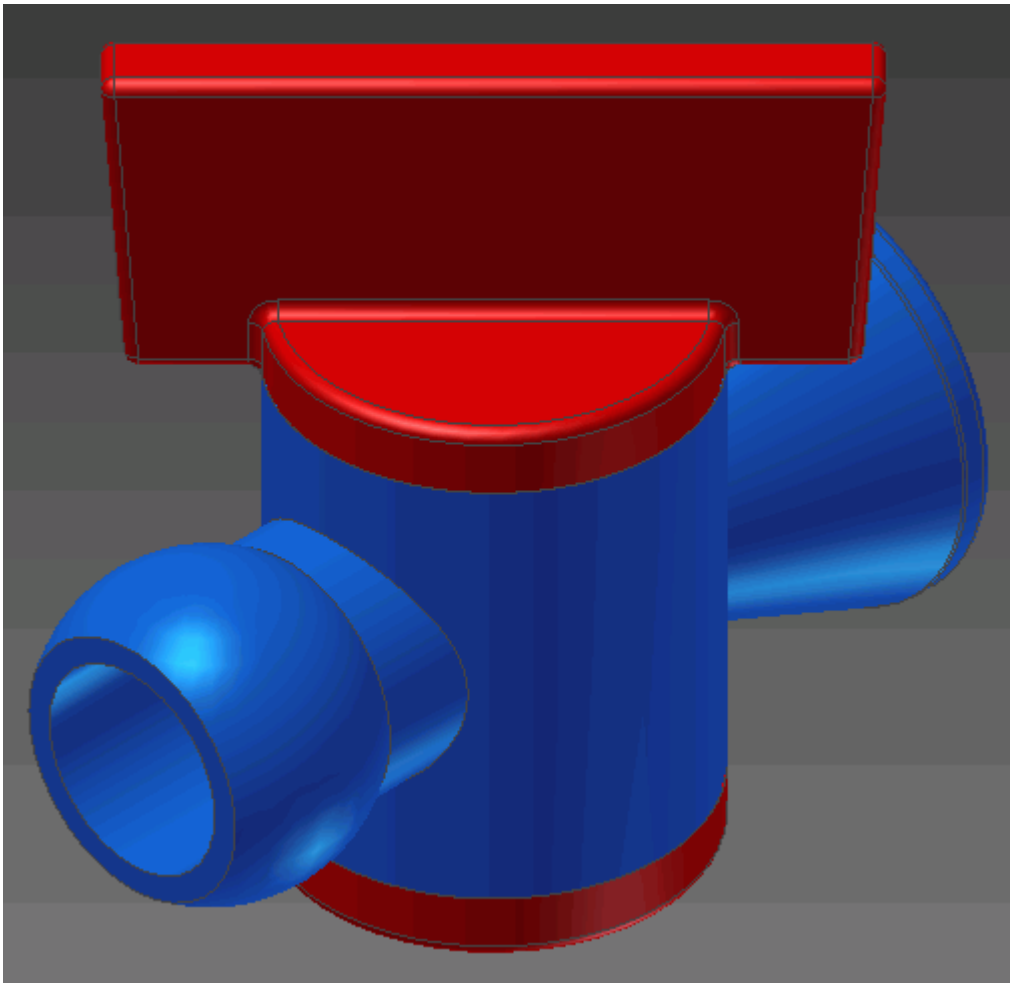
Internal Flow
External Flow
Multi Region Flow

Internal Flow

In general, the steps to follow in creating the fluid domain for an internal flow are:

- Work in a part. Import the original geometry into the new part (by making the new part derived), ensuring that all parts are maintained as separate solid bodies.
- Close any openings by creating “caps”.
- Create an overall block that encloses all the other geometry parts, the “enclosure”.
- Subtract all parts from the enclosure and delete the outer solid body--the remaining geometry is the fluid domain.
- Create a new assembly from the multi body derived part.

A worked example is provided below. In this case the assembly contains two parts: the valve (shown in red), and the valve housing (shown in blue).



1. Create a new part file in CATIA and name it `Internal Fluid Volume`.

   By default CATIA will open a sketch--close this sketch.

We will make this part file derived and import the assembly we want to extract the internal fluid volume from. During the import operation we can choose to import each part of the assembly as separate bodies. This is useful if you want to visualize the original components or perform a conjugate heat transfer analysis.

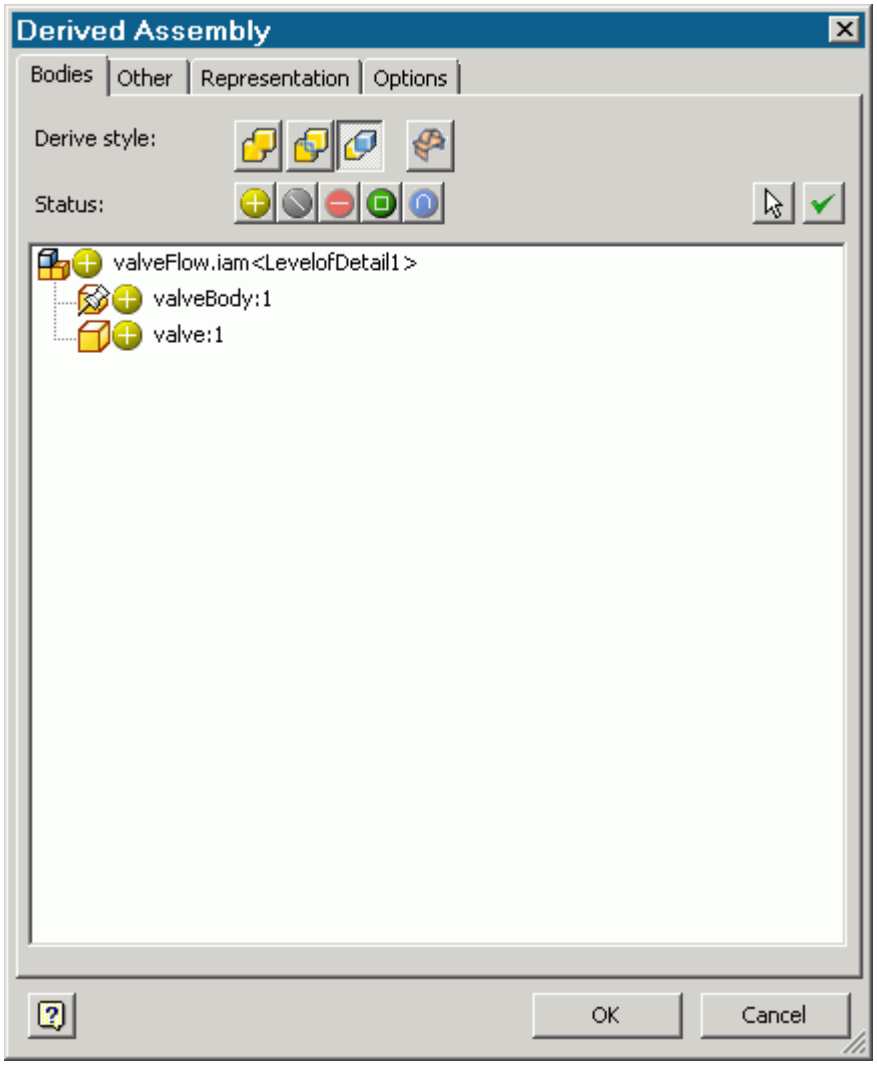
2. Click on the **Derive** button in the **Create** panel of the ribbon.

3. Locate and select the assembly or part file where you want to create the internal volume.

   The *Derived Assembly* dialog appears.

4. In the *Derived Assembly* dialog, click (**Maintain each body as a solid body**).

This will keep the solids in the assembly as separate bodies. The default option is to merge all solids into one solid body.



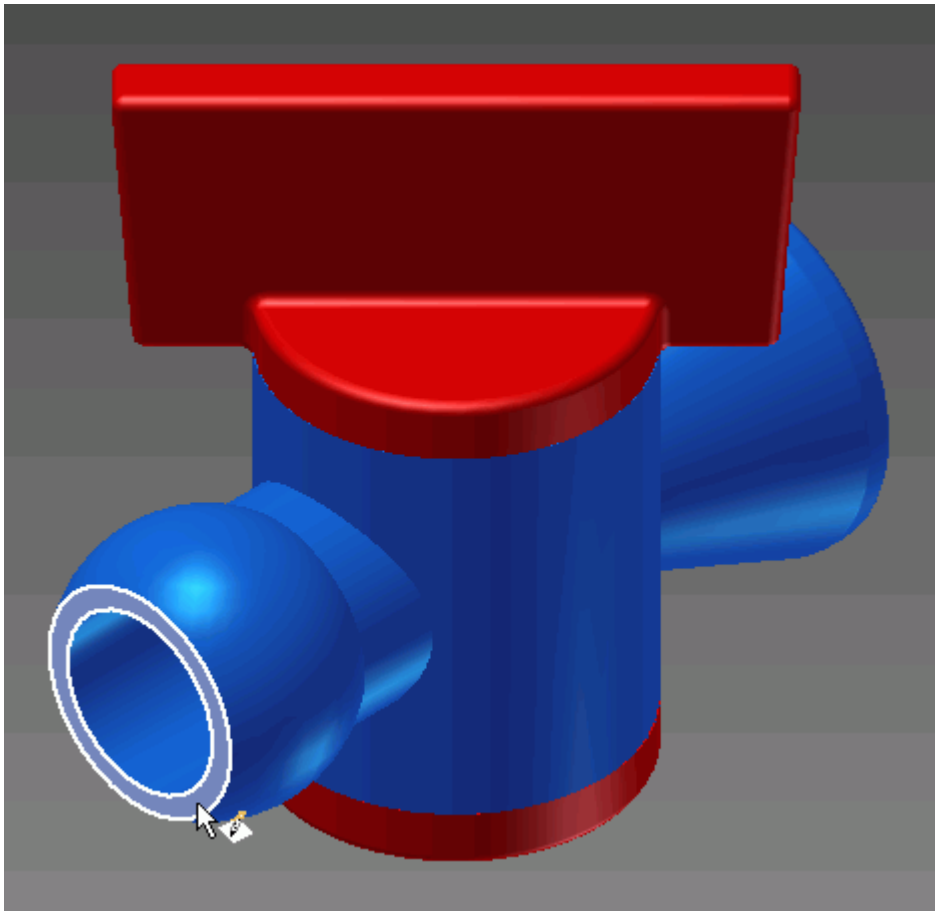
Notice that two bodies were imported into the part file under the **Solid Bodies** node in the feature tree.

5. Rename the imported bodies to something more representative--in this example the two bodies were renamed to `Valve Housing` and `Valve` respectively.

   Hovering the mouse pointer over the body nodes will highlights the corresponding body in the geometry window, this makes it easier to identify the bodies when renaming them.

We will now cap any openings.

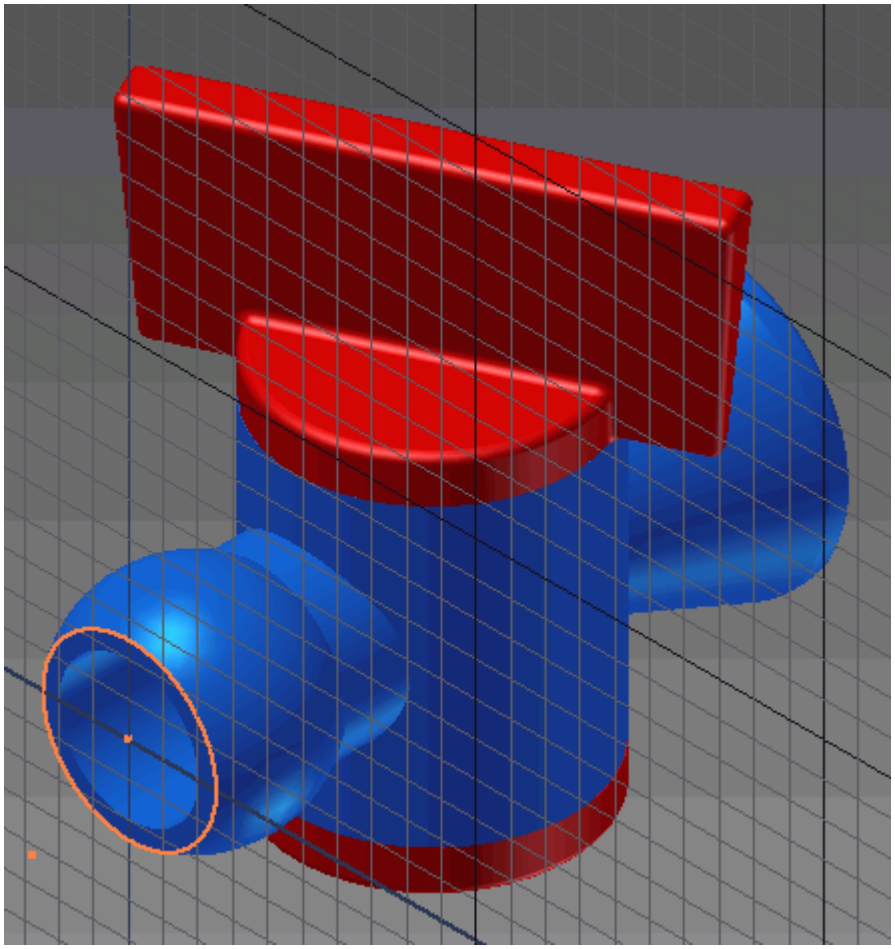
6. Create a sketch on one of the faces containing an opening, as shown below.



By default, CATIA automatically converts any edges that lie on the sketch plane into sketch entities. These will appear in orange.

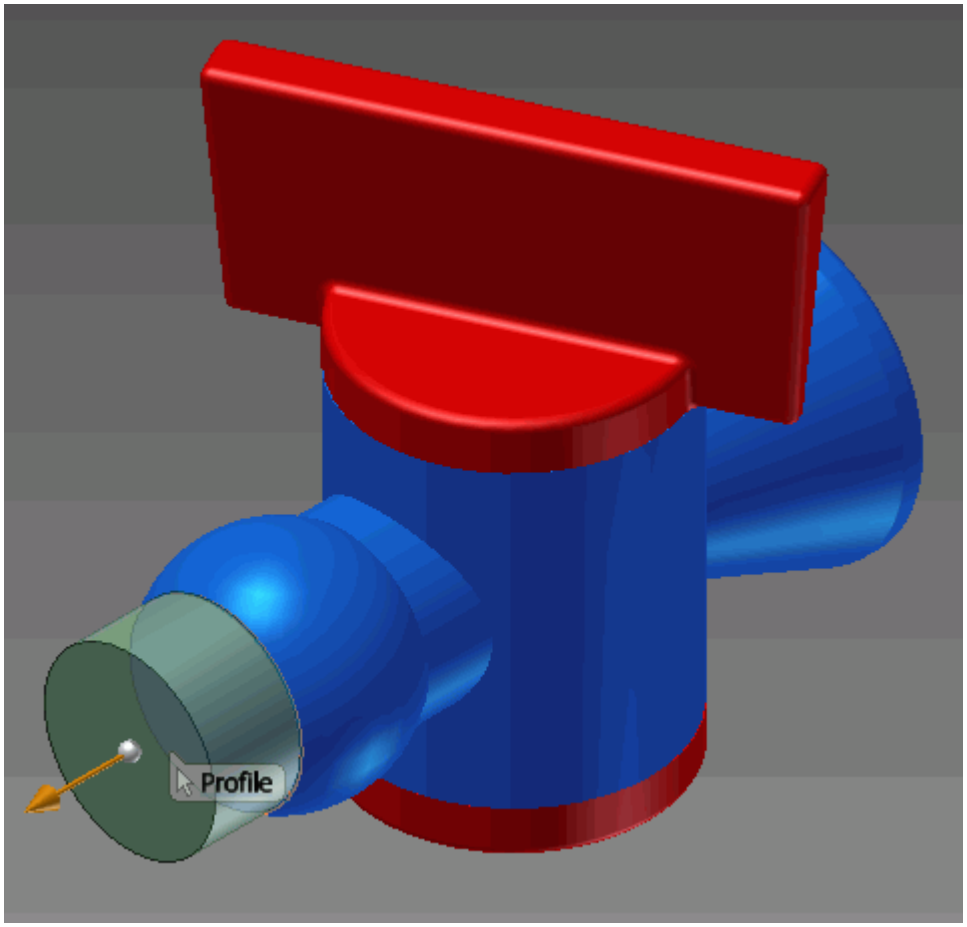
7. In this case we only require the outer edge, so click on the inner edge and delete it by pressing `<Delete>`.

   If the automatic projection option was turned off, you will need to manually project the outer edge of the face onto the sketch.

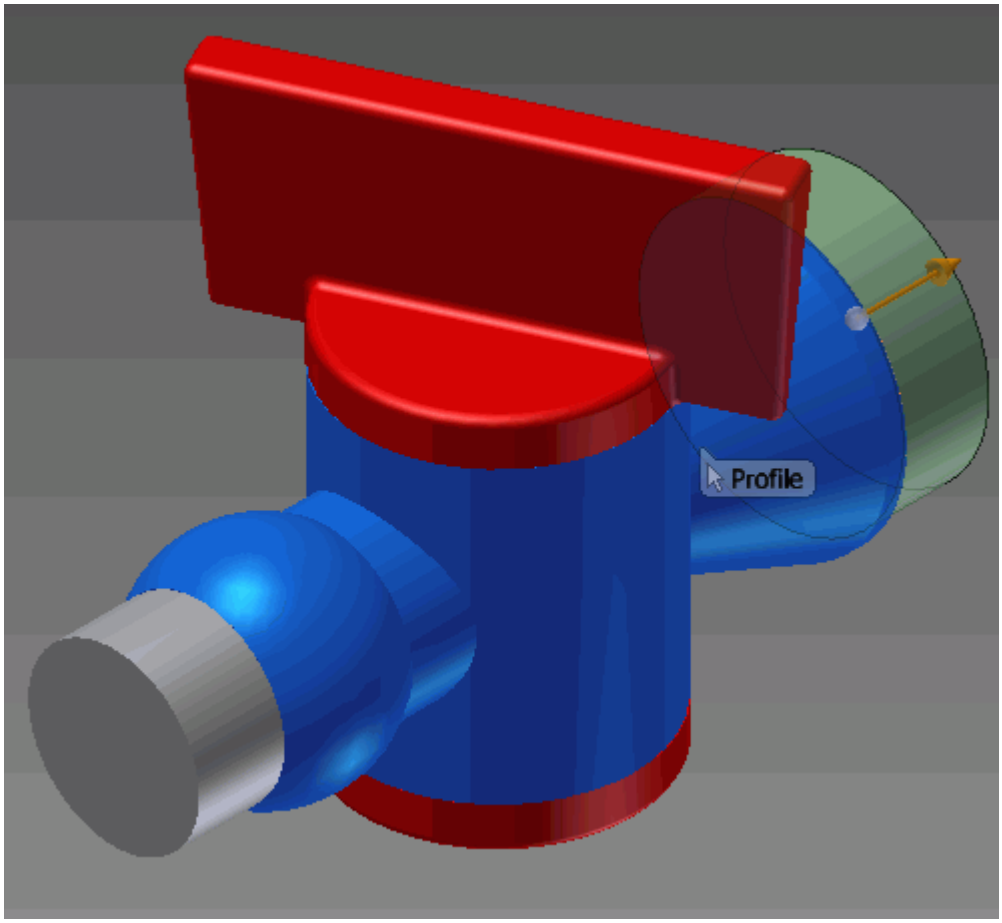


8. Create an extrusion from the sketch to cap the opening. The length of the extrusion is not important.

   In the Extrude dialog, click on the (**New solid**) button to make the cap a new body rather than merge it with the existing body it is in contact with.

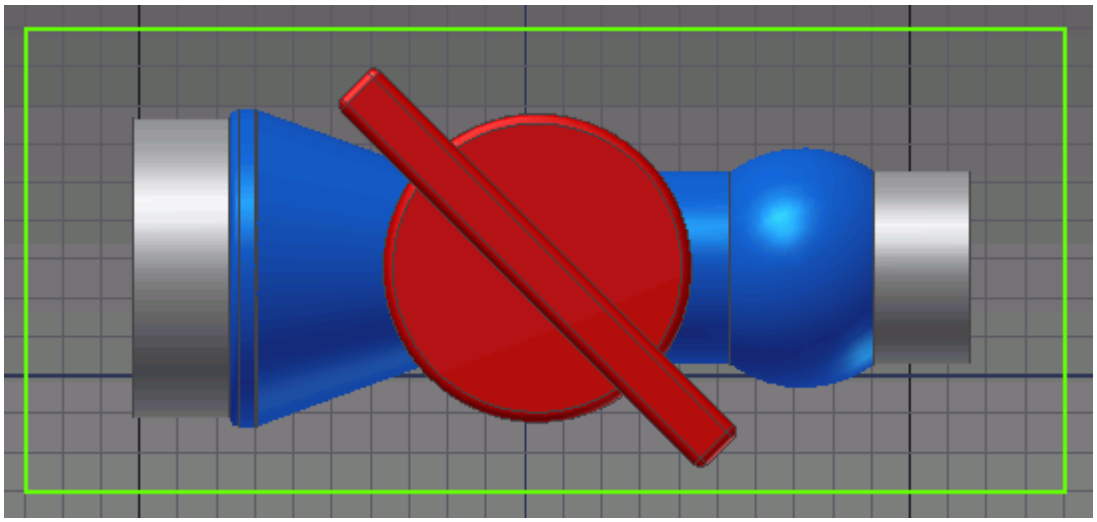


9. A new body will be added to the **Solid Bodies** node in the feature tree; rename this to `Cap1`.
10. Using the same technique, cap any other openings on the geometry, and rename the new bodies in the **Solid Bodies** node in the feature tree.



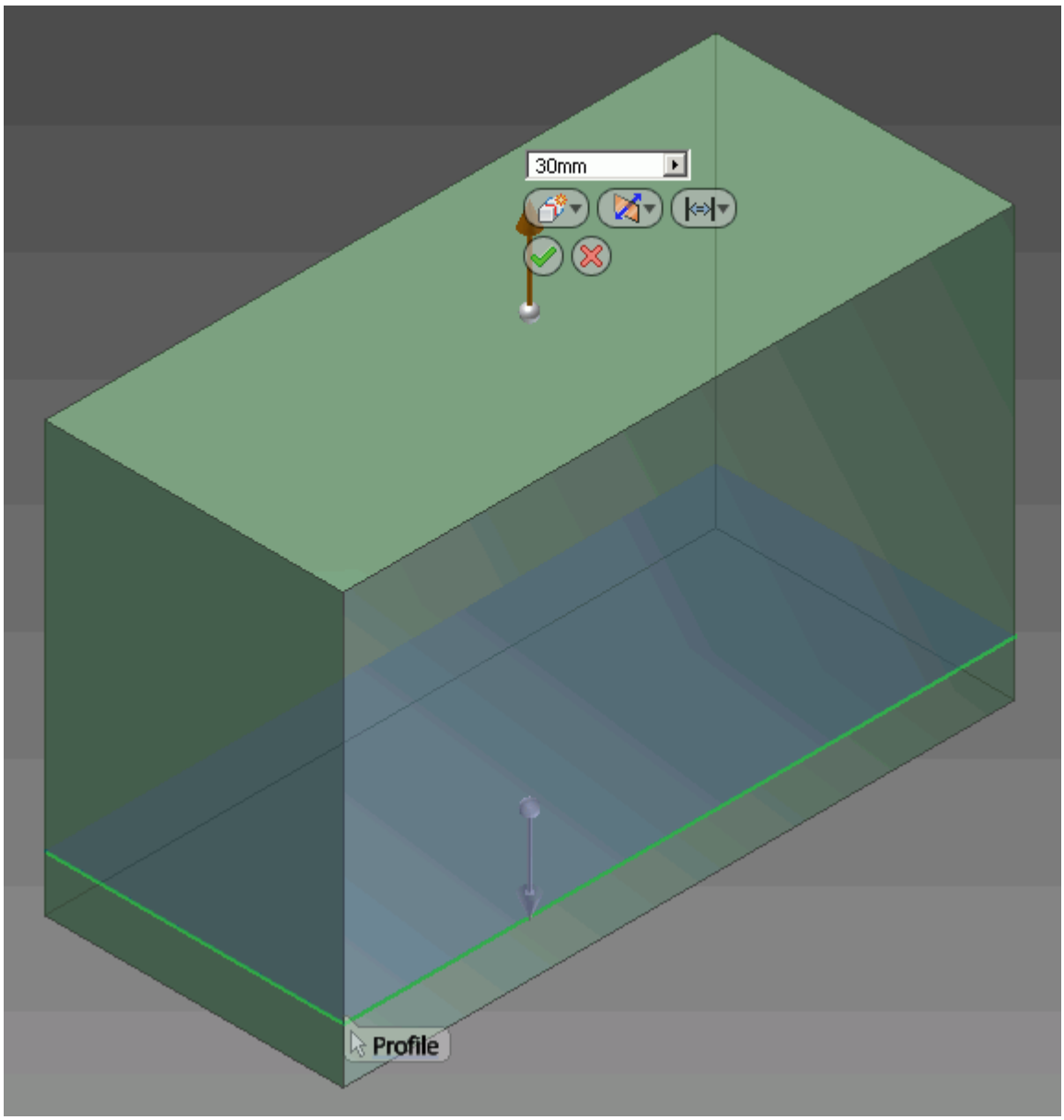
We will now create a solid block that completely encloses the other parts. This block will be added as a separate solid body in the part file.

11. Create a sketch that completely encloses the assembly, as shown below.



12. Extrude the sketch so that it completely encloses the geometry.

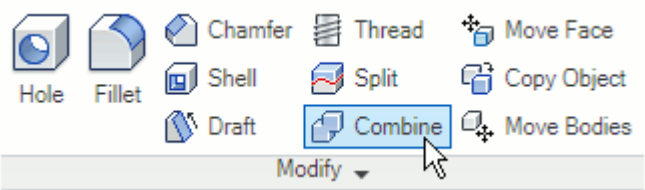
In the *Extrusion* dialog, click on the (**New Solid**) button, otherwise the geometry contained within the extrusion would be merged with the extrusion.



13. Note that a new solid is added to the **Solid Bodies** node in the feature tree, rename this to `Internal Fluid Volume`.

We will now subtract the original assembly and caps from the solid block. The result will be a solid block that contains a cavity separating the outer solid from the inner solid (the internal fluid volume).

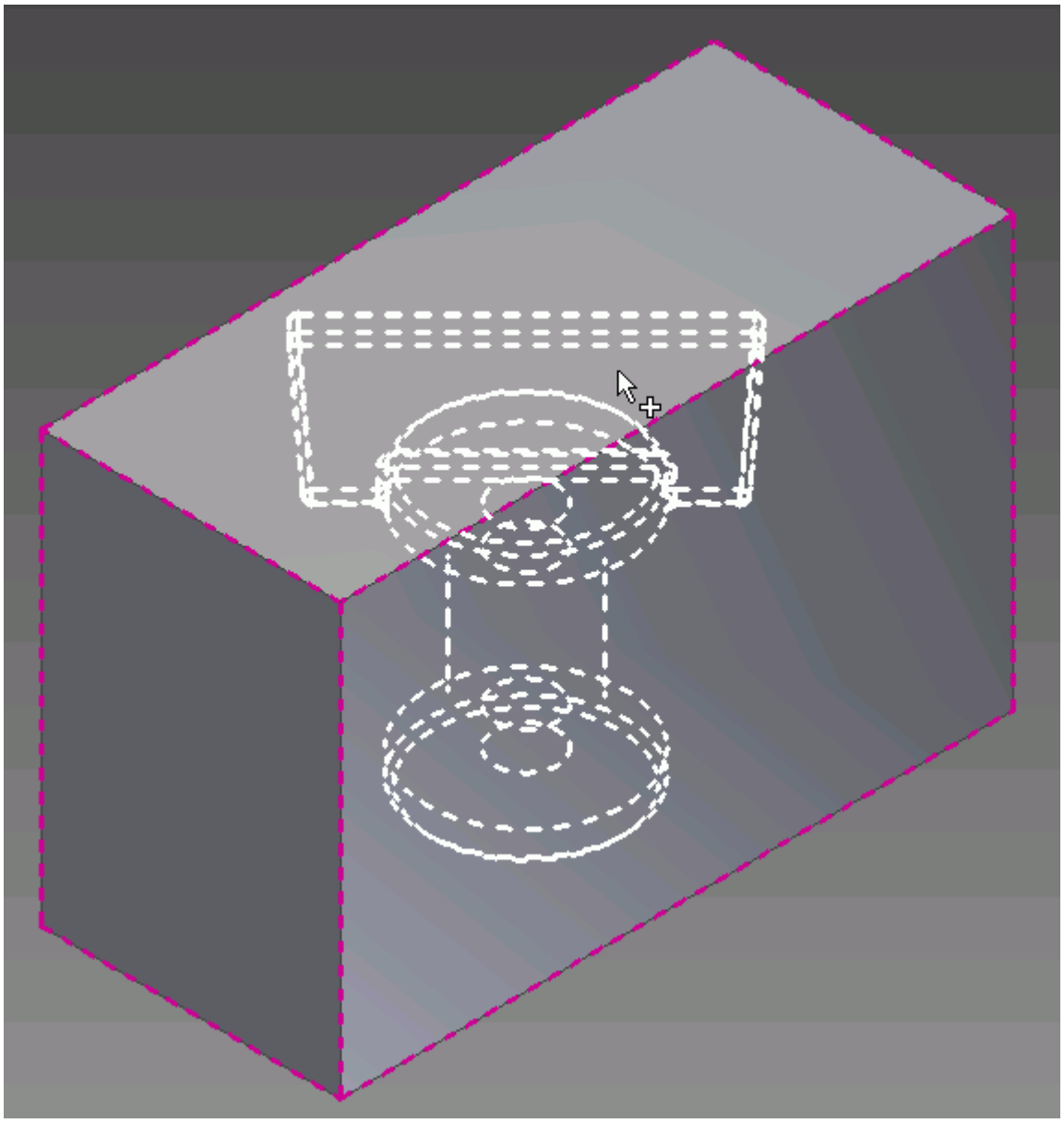
14. Click on **Combine** in the **Modify** panel of the ribbon.



The *Combine* dialog will appear.

15. Click on the (**Cut**) button, and select the enclosure as the *Base*.
16. Now select the other bodies (the original assembly parts and the caps) as the *Toolbody*.

    A dashed outline will appear when the mouse pointer hovers over hidden parts, as shown below. Click to select these parts.

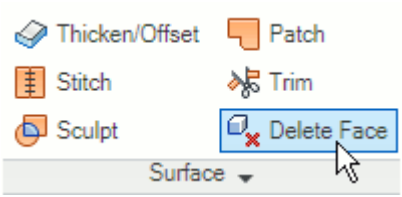


17. In the *Combine* dialog, activate the *Keep Toolbody* option.

    By default the toolbody parts are consumed by this operation, however we want to keep them.

We will now delete the outer solid. The remaining solid is the internal fluid volume.

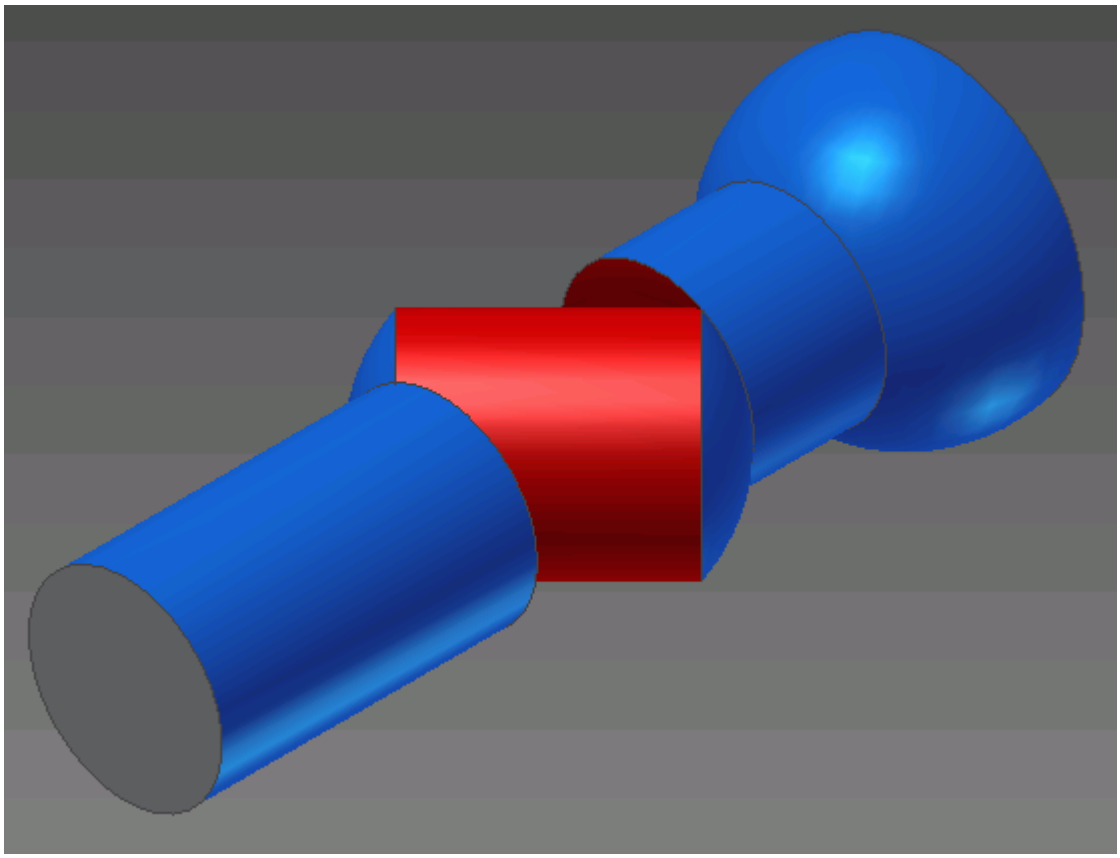
18. Click on the **Delete Face** button in the **Surface** panel of the ribbon.



    The *Delete Face* dialog will appear.

19. Click on the (**Select lump or void**) button, and select the solid body in the geometry window--this will delete the outer solid body and leave the internal fluid volume.

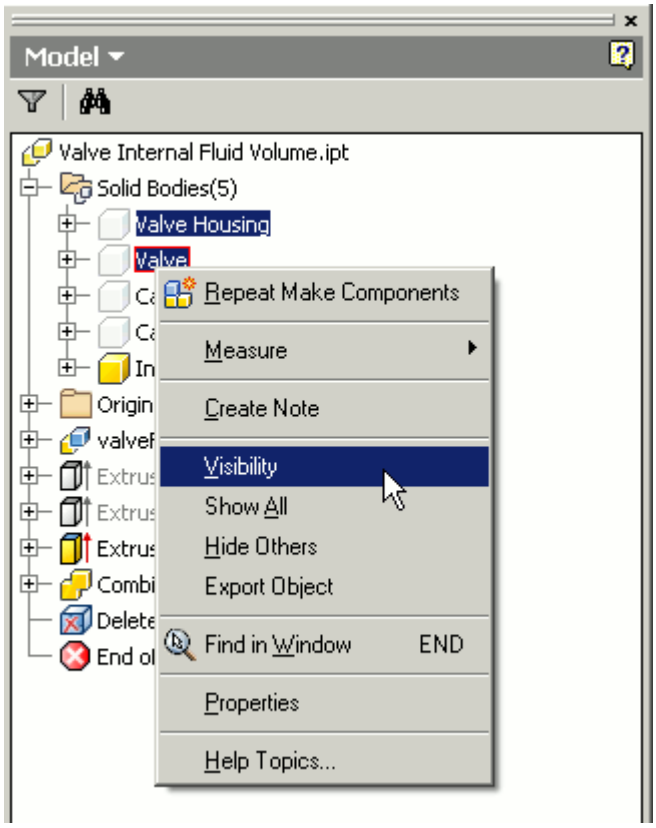
    The internal fluid volume of the geometry is shown below.



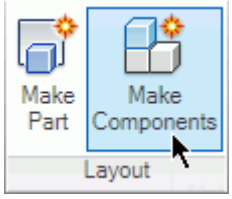
The last step is to export the bodies from this multi body part into an assembly, where they can be selected for use in STAR-CAT5.

20. Save the current part file using a suitable name.

21. Make the original assembly components visible by right-clicking on these in the **Solid Bodies** node in the feature tree and selecting **Visibility**.



22. Click on the *Manage* tab of the ribbon.
23. Click on the **Make Components** button in the **Layout** panel of the ribbon.



The *Make Components : Selection* dialog appears.

24. Select the internal fluid volume and the original components from the geometry window. These will be added to the dialog.
25. Click **Next**.

    The *Make Components : Bodies* dialog appears.
26. Give the new assembly (and parts) a suitable name and click **OK**.

    The new assembly will be opened automatically.
27. Save the new assembly.

The derived part is associative, meaning that if the original part/assembly is modified, the derived part can be updated to reflect this. The up-to-date derived part can then be exported to an assembly, as described above, where it can be used with STAR-CAT5.

The geometry is now ready for the CAD client analysis. Suppressed parts in an assembly will not be included in the analysis. In the above example, it is recommended that you extend the inlet and outlet regions to appropriate lengths so that a fully developed flow can be achieved before reaching the valve and to help reduce recirculation at the outlet.

External Flow

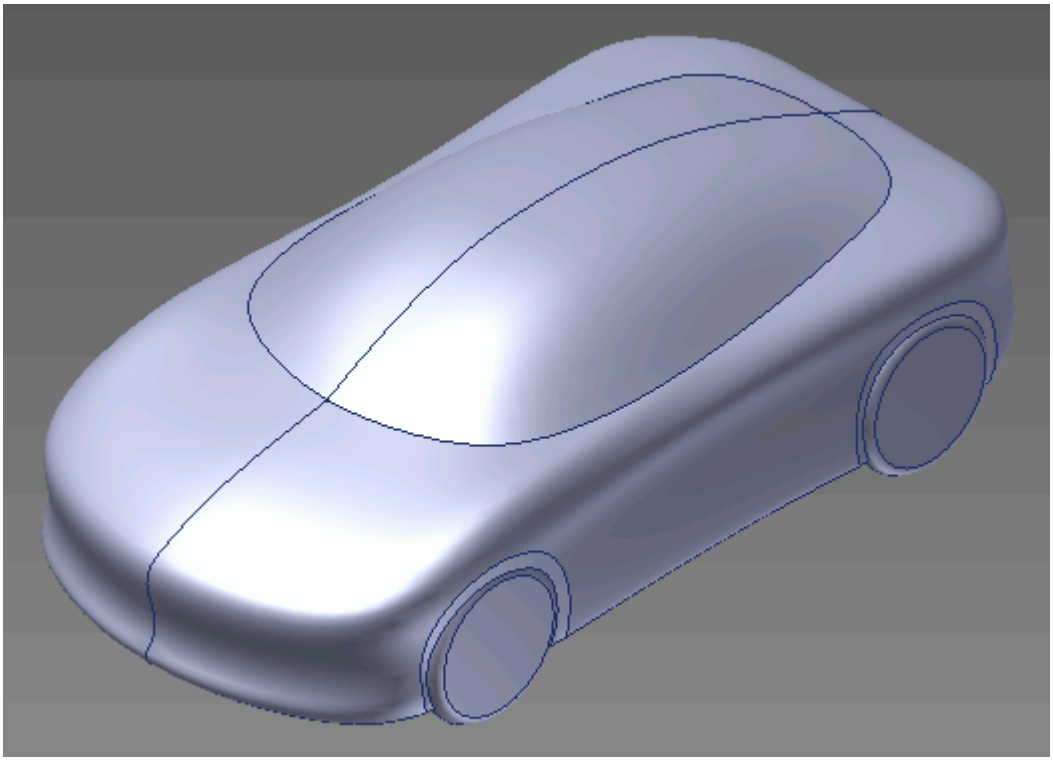
The generation of a domain for external aerodynamics is generally easier than for internal flow cases but the sizing of the domain is more critical. As a guideline (assuming a specific situation is not being modelled i.e. flow in a wind tunnel) the domain should be sized as follows:

- Inlet: 2 x part length from the front of the part
- Outlet: 4 x part length from the rear of the part
- Sides: 3 x part width from the side of the part
- Ceiling: 4 x part height from the top of the part

In general, the steps to follow in creating the fluid domain for an external flow are:

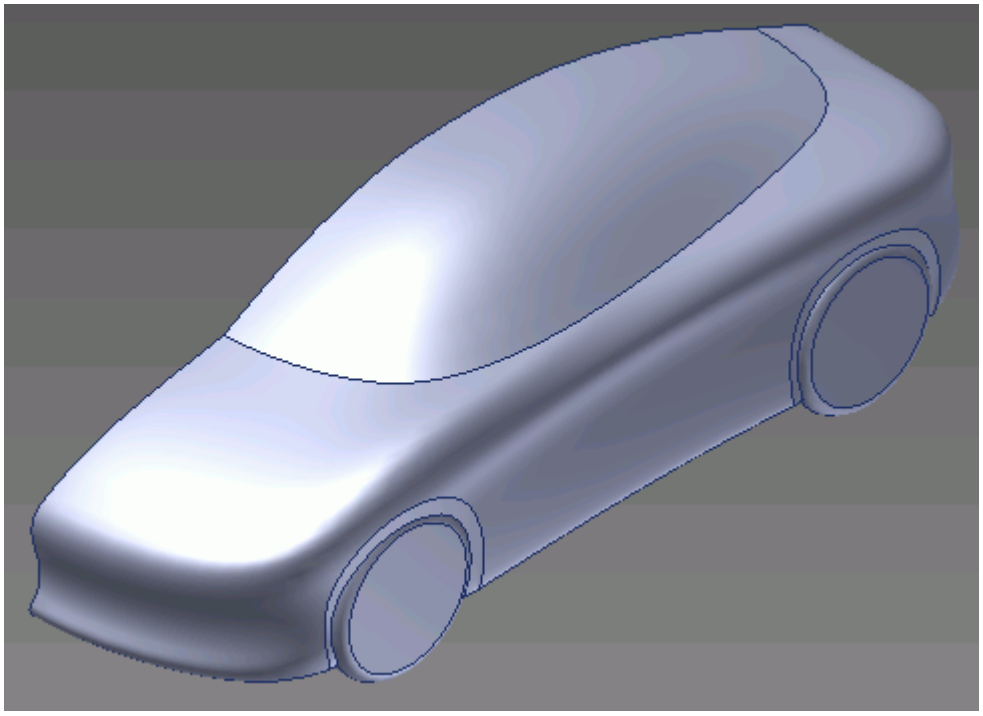
- Work in a part file. If the original geometry contains an assembly then create a new part and make it derived, loading the assembly into this part.
- Create an overall block that encloses all the other parts. Ensure that this block is a separate solid body in the part. The shape and size should correspond to the desired fluid domain.
- Subtract all parts from the overall block part - the remaining geometry is the fluid domain.

An example case is shown below. In this example a geometry of a car will be used.



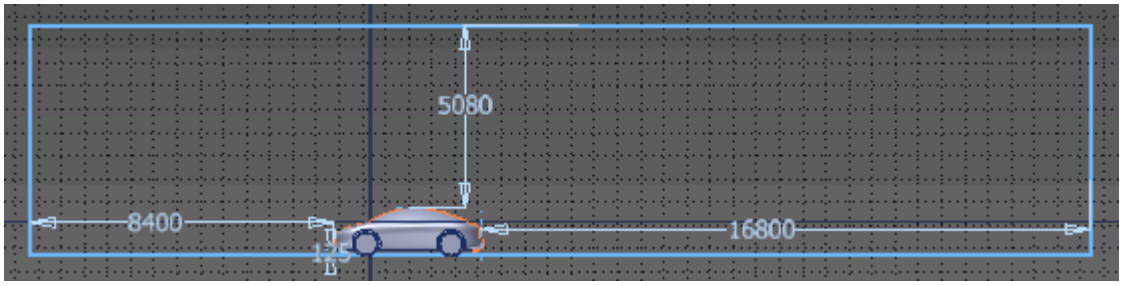
1. Create a new part file and make this a derived part.

2. Load the assembly into the part file, using the default import options. The assembly parts will be merged into one solid body.
3. If the geometry is symmetrical, as is the case with the car in this example, cut it in half to save on computational costs.

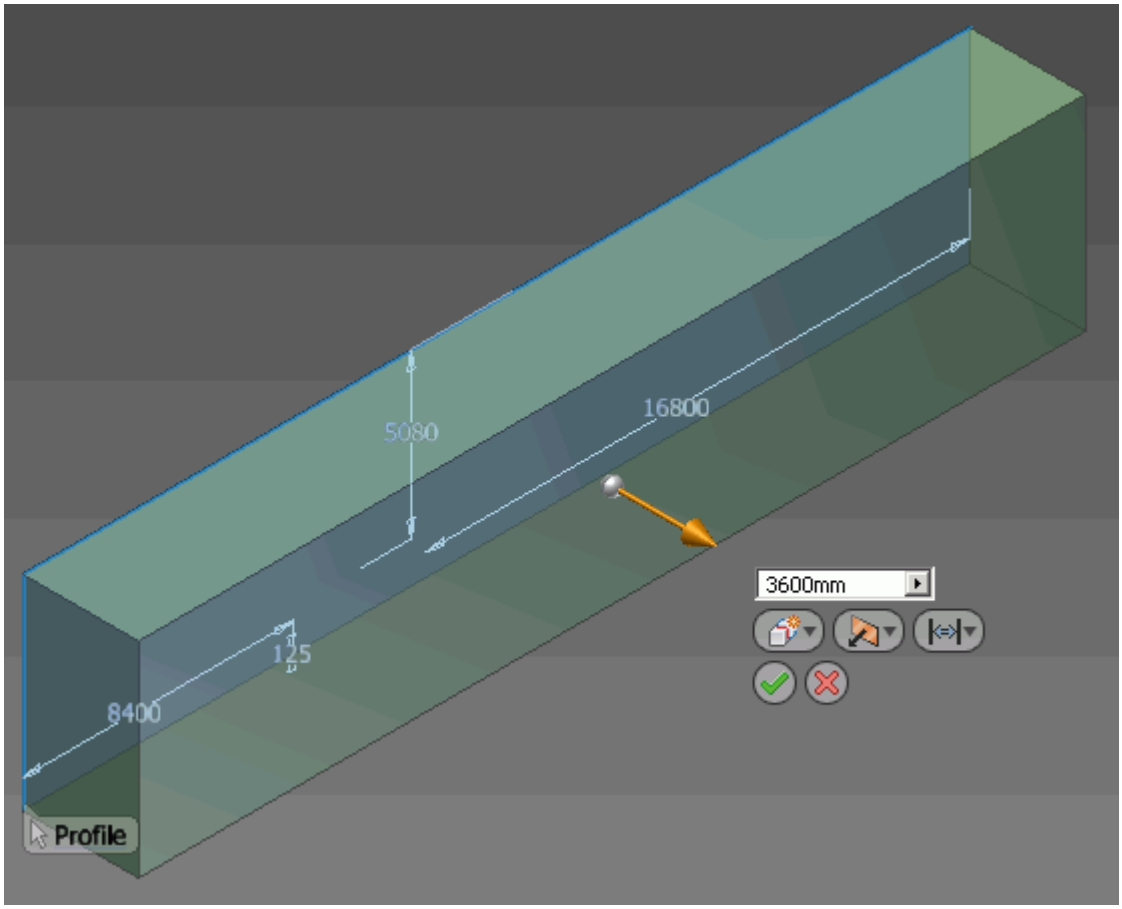


We will now create the fluid domain for the case.

4. Create a sketch on a suitable plane, in this example the symmetry plane of the car will do.
5. Draw the bounding box of the fluid domain using appropriate dimensions. In this case the sketch will represent the profile of the fluid domain, as shown below.

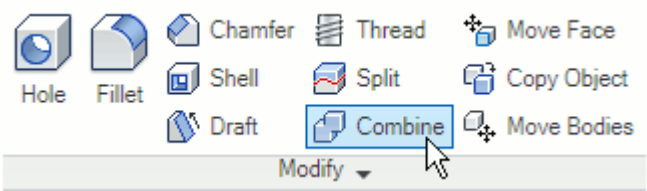


6. Extrude the sketch to the appropriate thickness for the domain.



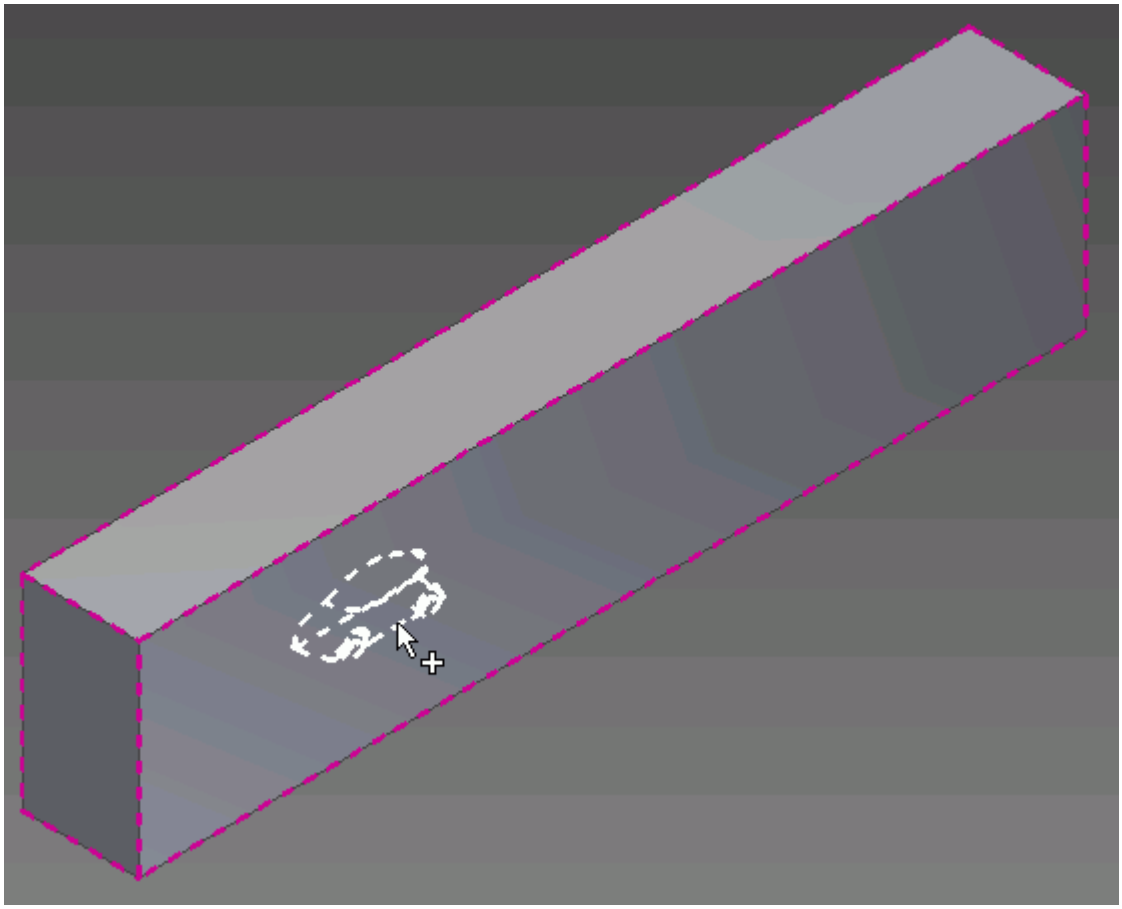
The *Extrude* dialog appears.

7. Click on the (**New Body**) button to create a new solid body from this extrusion--the default option would merge all solids together. Note that a new solid body is added to the **Solid Bodies** node in the feature tree.
8. Click on **Combine** in the **Modify** panel of the ribbon.

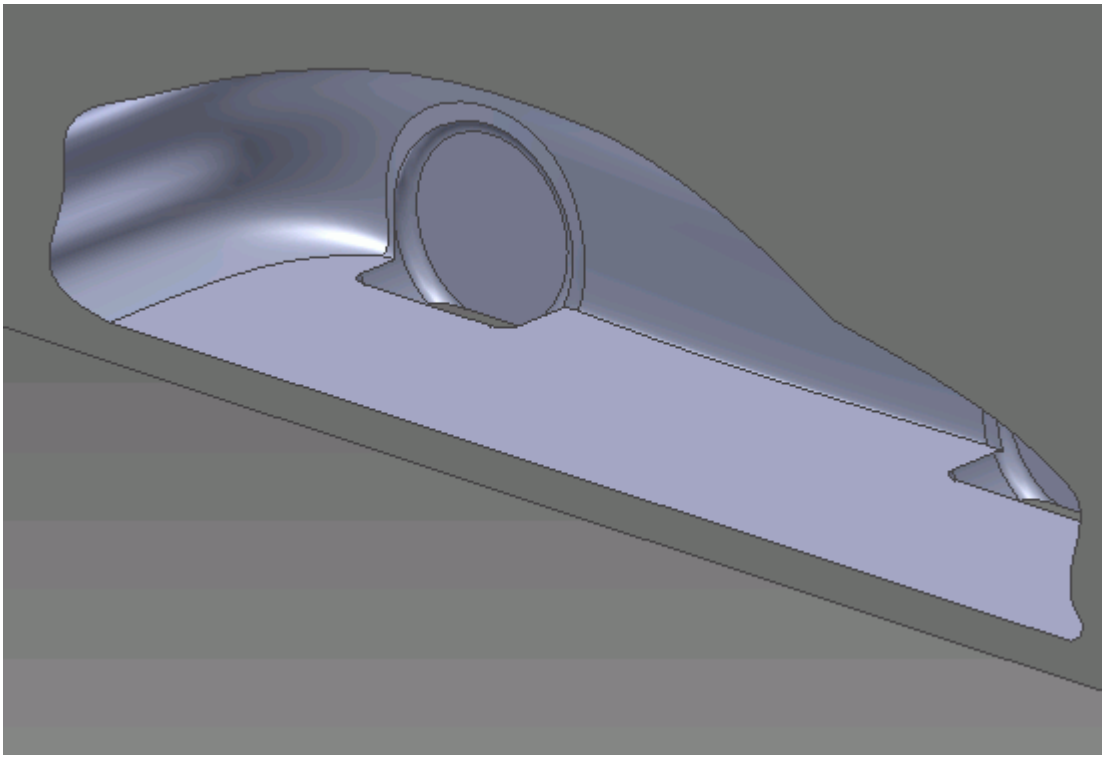


The *Combine* dialog appears.

9. Click on the (**Cut**) button, and select the domain as the *Base*, and the car as the *Toolbody*.
   A dashed outline will appear when the mouse pointer hovers over hidden parts, as shown below; click on this to select the hidden body.



The fluid domain is now complete. A close-up of the car in the fluid domain is shown below.



The geometry is now ready for the cad client analysis. Deactivated parts are not included in the analysis.Suppressed parts are not included in the analysis. If you make any changes to the car geometry, these will automatically be passed on to the fluid region, giving you an up-to-date fluid domain.

Multi Region Flow

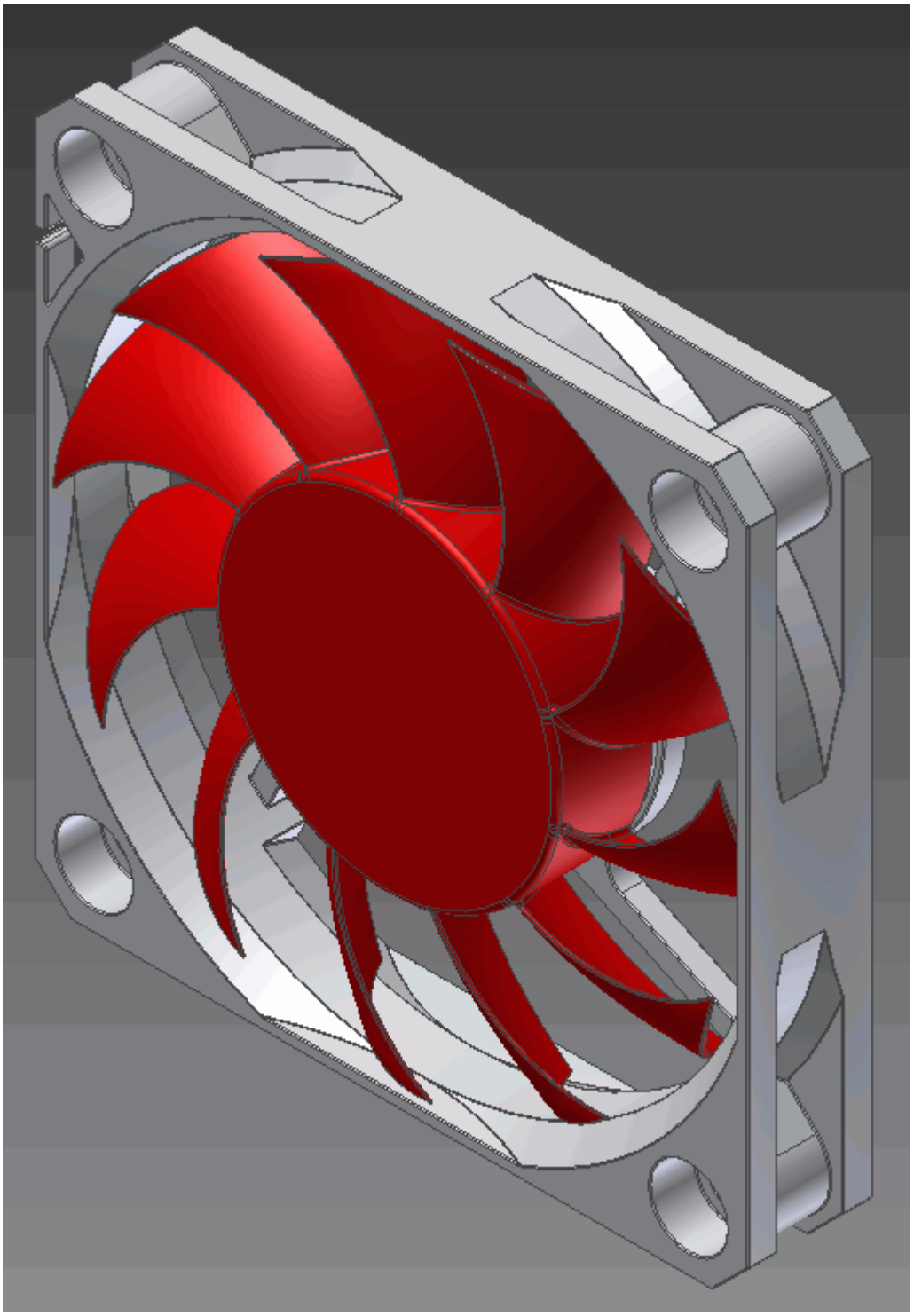
This section shows how to create the fluid domain for multiple regions such as rotating machinery and porous media flow.

For the numerical assumptions made using the Moving Reference Frame (MRF) technique to hold true, a minimum of two cells need to be present between the rotating-static interface and any wall boundaries (for example the fan blades), and the rotating region must be circular in section. Where there is a large clearance between the rotating object and any static objects, it is advisable to avoid placing the interface too close to either hence facilitating a small cell size on the interface. Where there is a small clearance between the rotating and static parts, there is little choice but to place the interface approximately midway between the two.

In general, the steps to follow in creating the fluid domain for multiple regions are:

- Create a new derived part, importing the original geometry into this part. Ensure that all parts are maintained as separate solid bodies.
- Create a new body that encloses the rotating component, and subtract the rotating component from the solid. This is now the rotating region.
- Create additional bodies that will form the inlet and outlet regions, and subtract any obstructions/features that may be present. An example would be the support arms of the fan motor in a fan casing.
- Create a new assembly from the multi body derived part.

In this example, a fan and fan case are used.



1. Create a new part file in Inventor. Name this new part `Fluid Domain`.

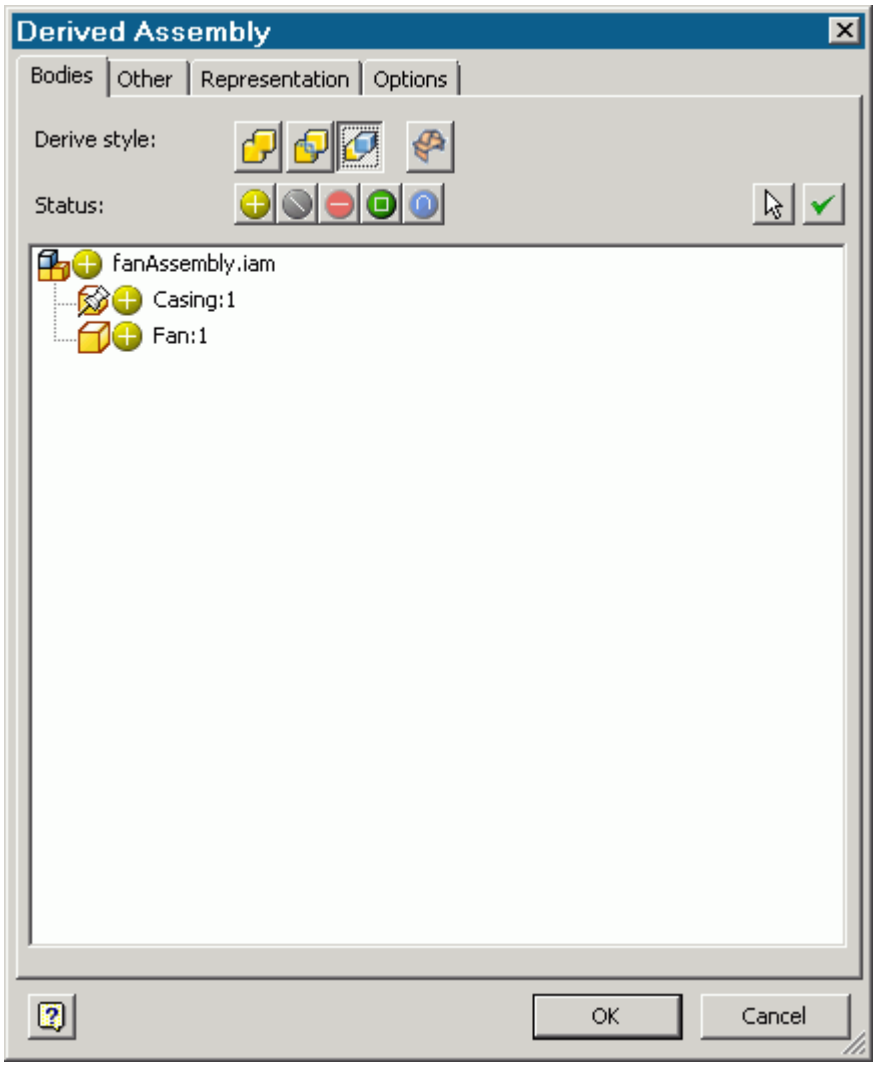
Make this part derived and import the assembly for which you want to create the fluid domain for.

2. Click on the **Derive** button in the **Create** panel of the ribbon.

3. Locate and select the assembly or part file you want to create the fluid domain for.

4. In the *Derived Assembly* dialog, click on the (**Maintain each body as a solid body**) button.

   This will keep the solids in the assembly as separate bodies. The default option is to merge all solids into one solid body.



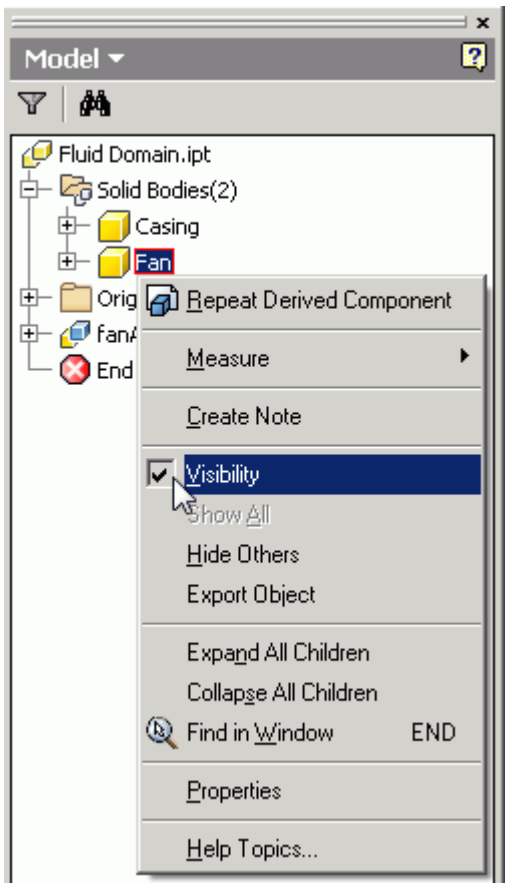
   Notice that two bodies were imported into the part file under the **=** > **Solid Bodies** node in the feature tree.

5. Rename the imported bodies to something more representative--in this example the two bodies were renamed to `Casing` and `Fan` respectively.

Hovering the mouse pointer over the body nodes highlights the corresponding body in the geometry window, this makes it easier to identify the bodies when renaming them.

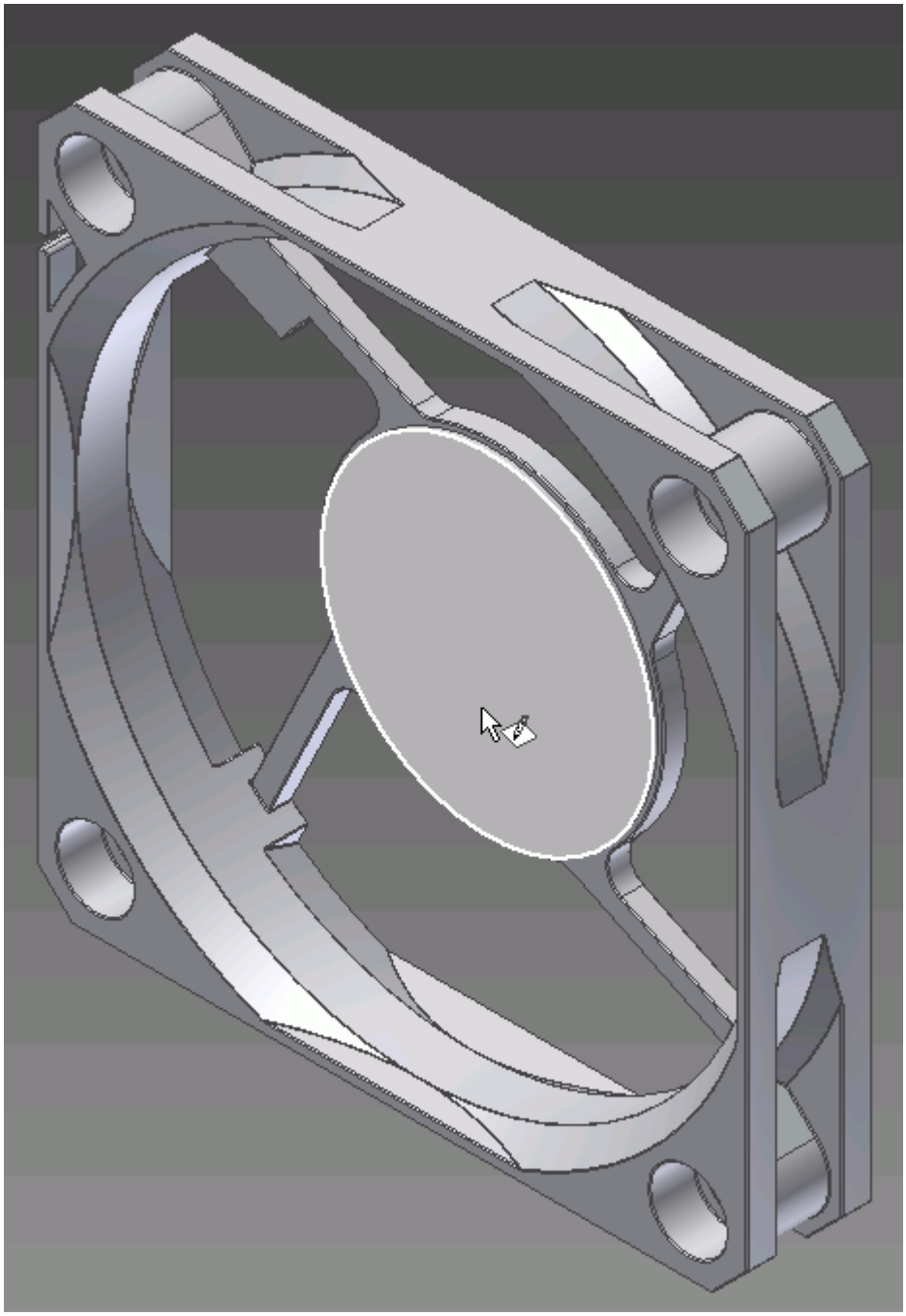
We will now create a solid body that will enclose the rotating component, this body will become the rotating region.

6. Hide the fan by right-clicking on its corresponding body sub-node in the **Solid Bodies** node and clear the checkbox next to **Visibility**.



To make a body visible again, right-click on it in the feature tree and tick the **Visibility** option.

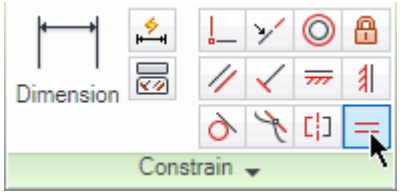
7. Create a sketch on the circular face of the fan casing.



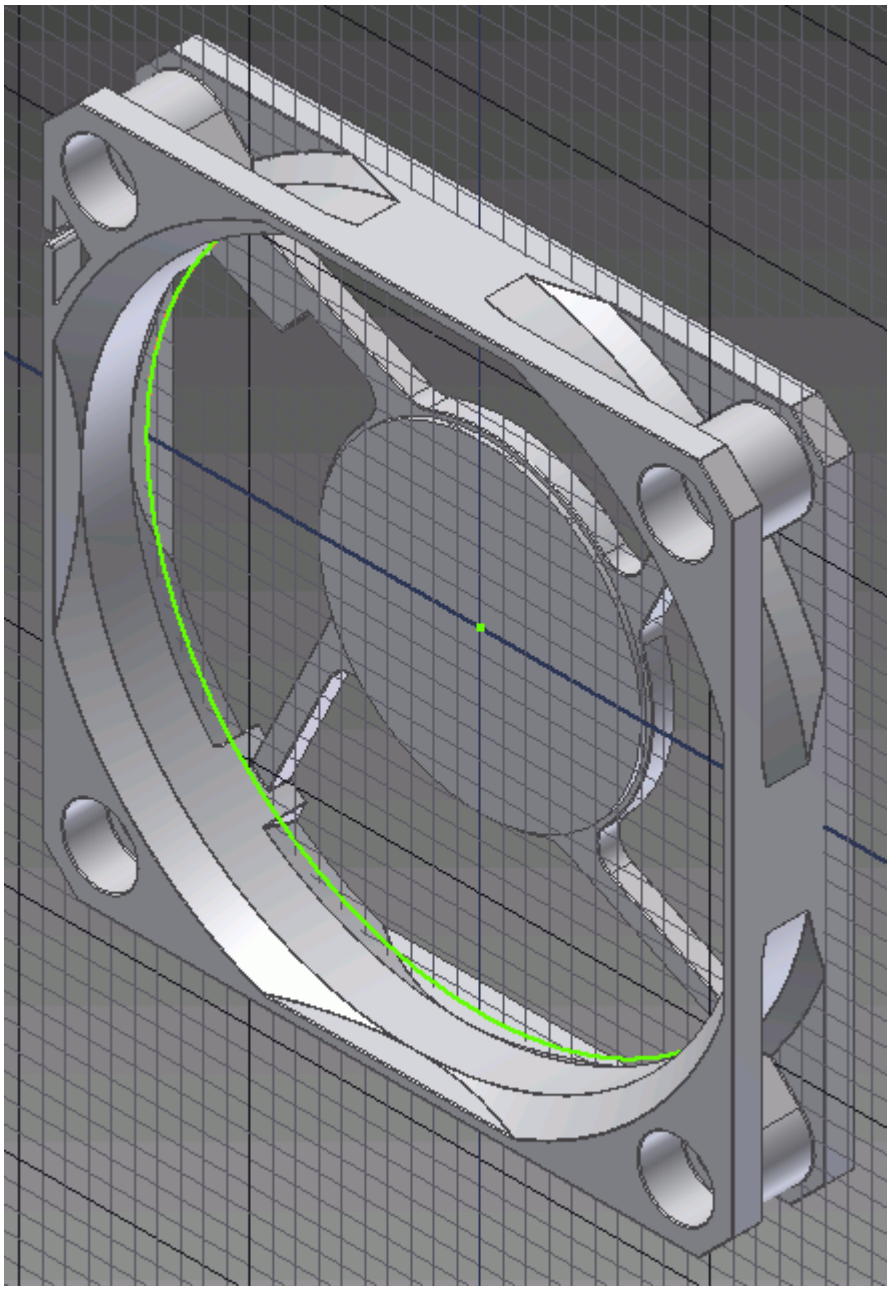
By default Inventor automatically adds any edges that lie on the sketch plane to the current sketch--these appear in orange. Delete this edge.

8. Draw a circle that shares the same center point as the fan casing. The edge of the circle should coincide with the innermost edge of the fan casing.

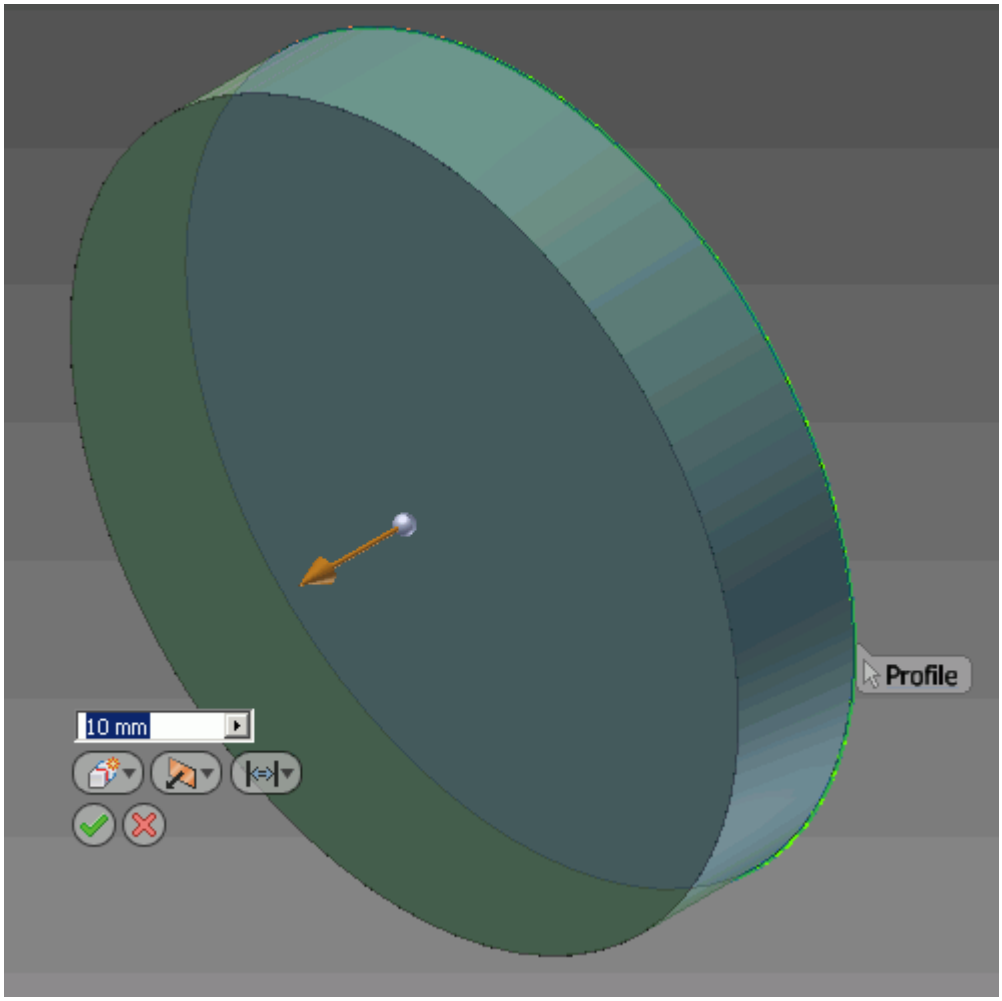
   To achieve this, select both the circle sketch and the innermost edge on the fan casing and click on the **Equal** button in the **Constrain** panel of the ribbon.



The result of this operation is shown below--the circle will snap to the selected edge.



9. Hide the fan casing and show the fan again.
10. Extrude the sketch until it completely encloses the fan.

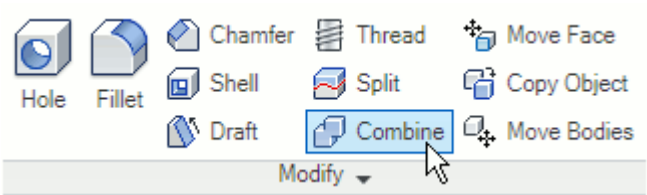


The *Extrude* dialog appears.

11. Click on the (**New Body**) button to create a new solid body from this extrusion--the default option would merge all solids together.
12. Upon completing the operation, a new solid body will be added to the **Solid Bodies** node in the feature tree. Rename this body to `Rotating Region`.

We will now subtract the fan from the rotating region.

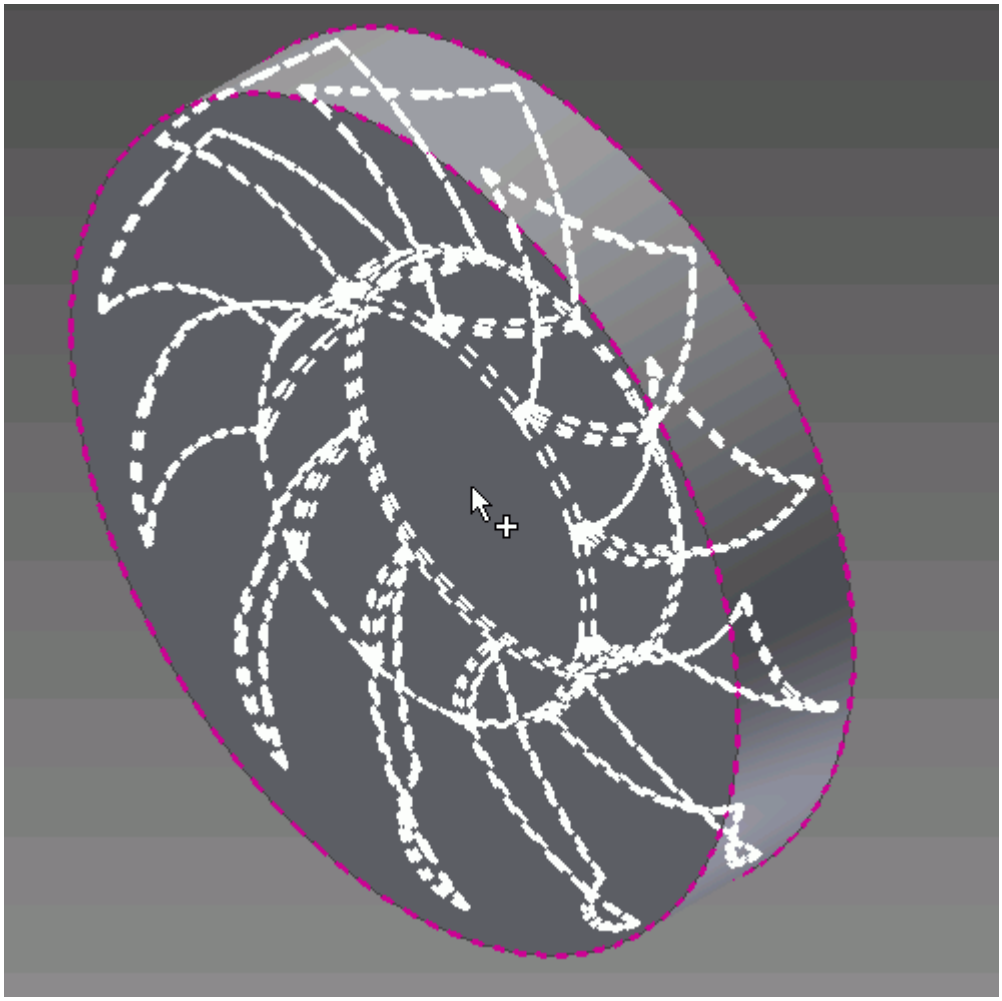
13. Click on Combine in the Modify panel of the ribbon.



The *Combine* dialog will appear.

14. Click on the (**Cut**) button, and select the rotating region as the *Base*, and the fan as the *Toolbody*.

A dashed outline appears when the mouse pointer hovers over hidden parts, as shown below; click on this to select the hidden body.

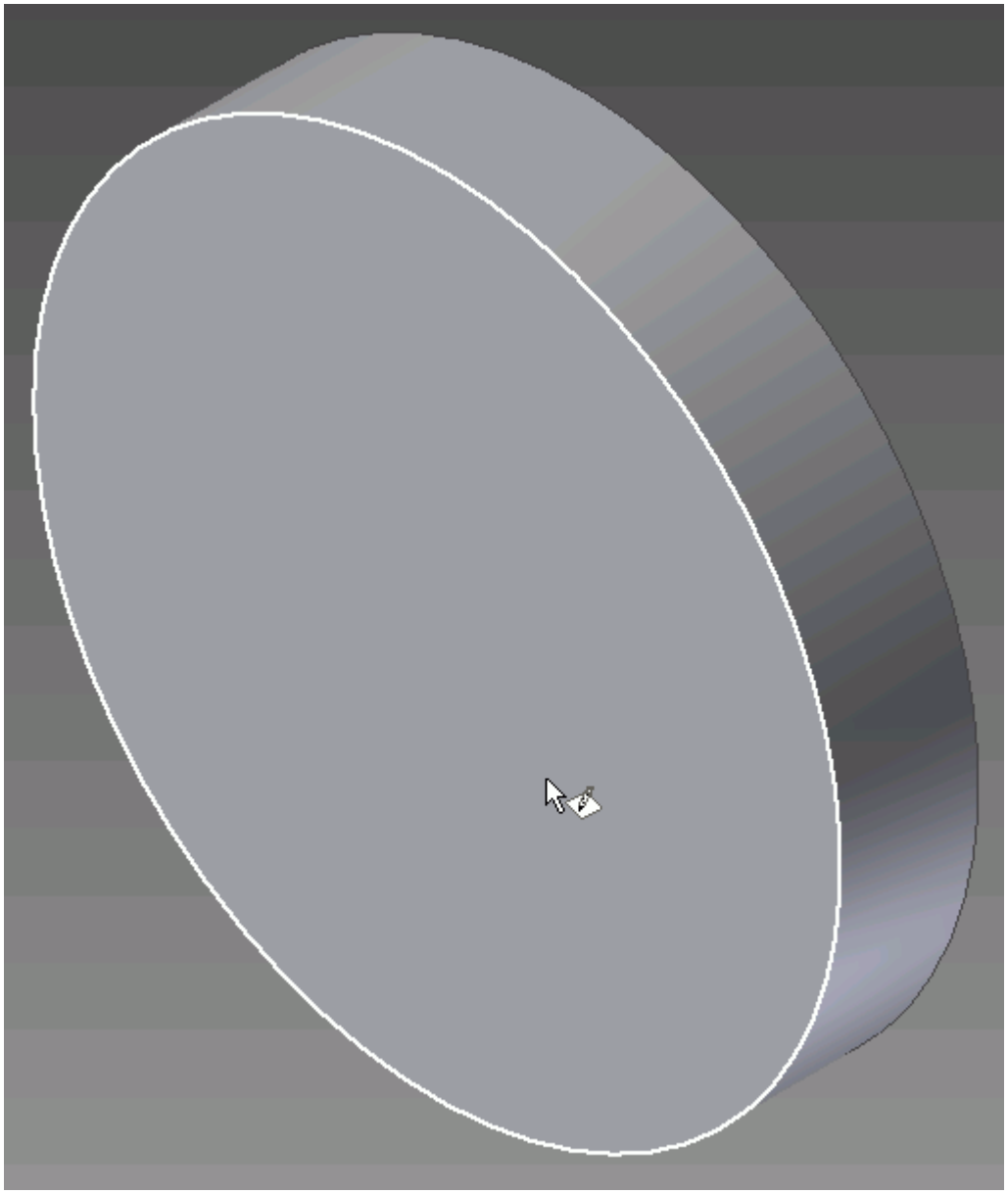


15. In the *Combine* dialog, tick the *Keep Toolbody* checkbox to keep the toolbody.

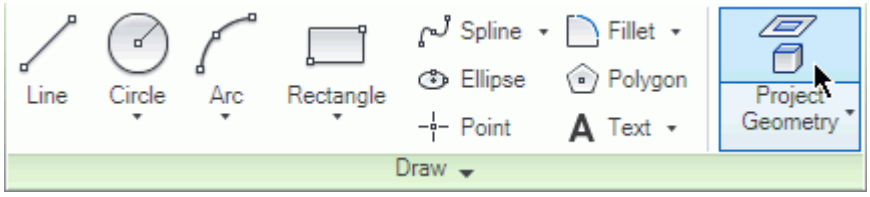
    If this is not selected, the toolbody will be consumed by this operation. Keeping the bodies is useful if you want to import these into STAR-CCM+ for visualization purposes.

We will now create a new body that will extend upstream of the rotating region. This will be the inlet region.

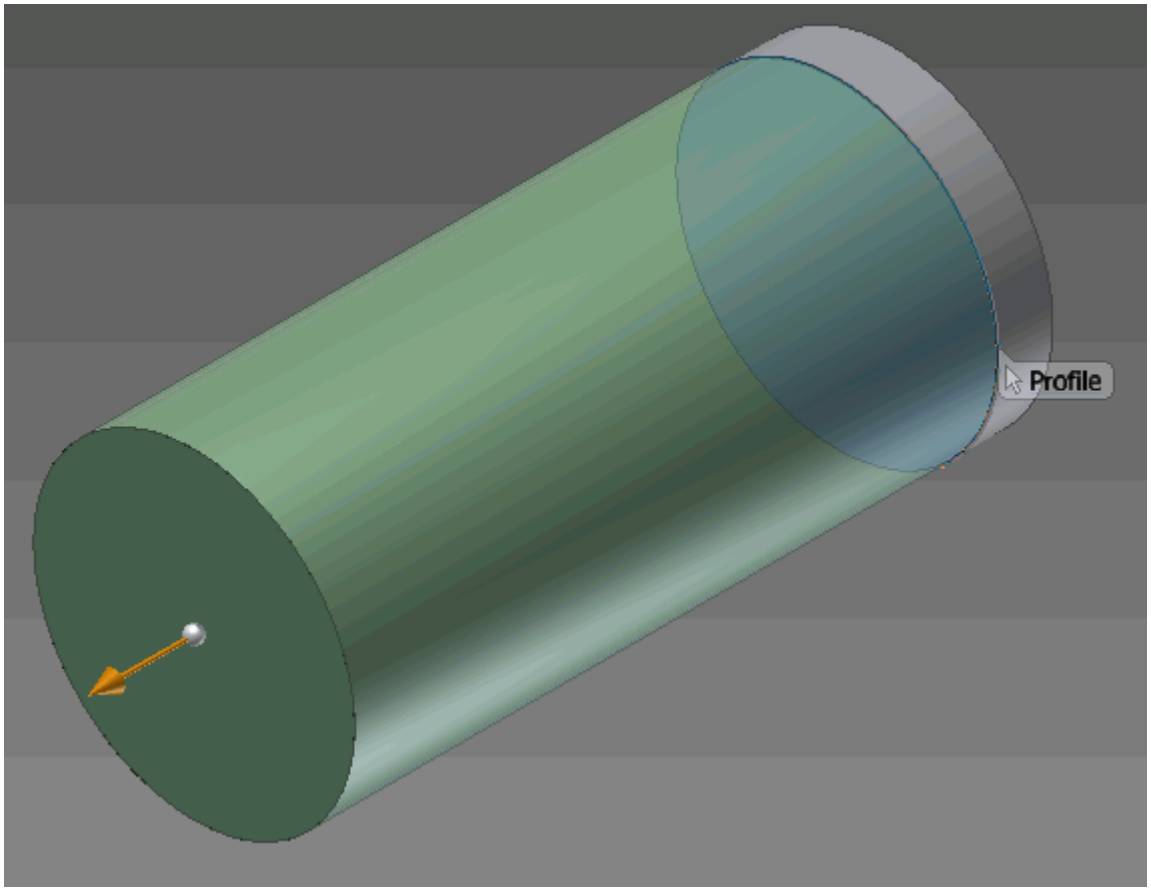
16. Create a sketch on the front face of the rotating region.



By default Inventor automatically adds any edges that lie on the sketch plane to the current sketch; these appear in orange. Keep this sketch and exit the sketch. If this feature of Inventor was turned off, you can select the outermost edge on the body and click on the **Project Geometry** button on the **Draw** panel of the ribbon.



17. Extrude the sketch to a suitable length for the case.

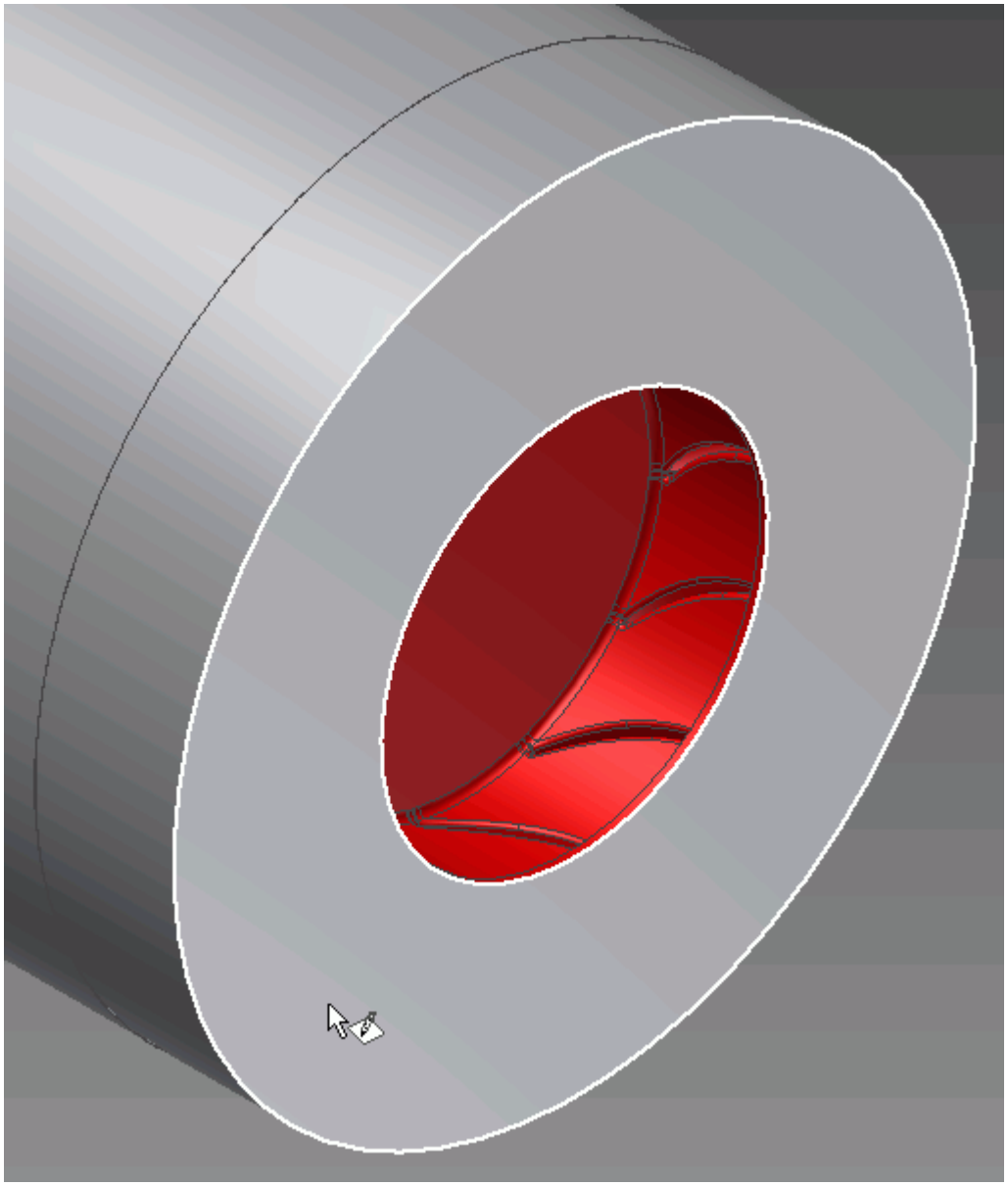


In the *Extrude* dialog, click on the (**New Body**) button to create a new solid body from this extrusion--the default option would merge all solids together.

Note that a new solid body was added to the **Solid Bodies** node in the feature tree,> Rename this to `Inlet Region`.

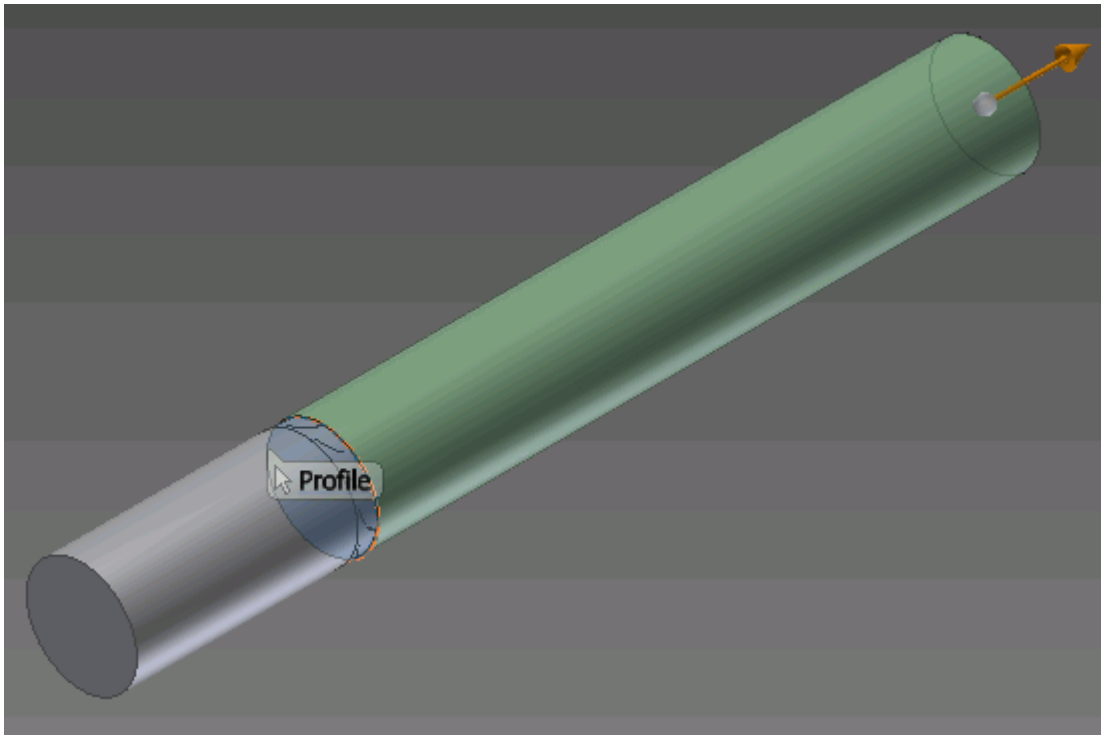
We will now create a new body that will extend downstream of the rotating region. This will be the outlet region.

18. Create a sketch on the back face of the rotating region.



By default Inventor automatically adds any edges that lie on the sketch plane to the current sketch; these appear in orange. We only need the outer edge, so the inner edge can be deleted. If this option was turned off, you can manually project the outermost edge as previously described.

19. Extrude the sketch to a suitable length for the case.



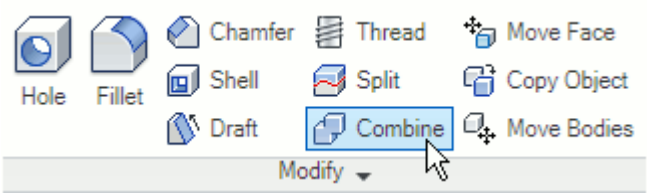
The *Extrude* dialog appears.

20. Click on the (**New solid**) button to create a new solid body from this extrusion--the default option would merge all solids together.

    Note that a new solid body was added to the **Solid Bodies** node in the feature tree, rename this to `Outlet Region`.

In this example, the fan casing contains support arms that hold the base of the motor. We need to subtracted these from the outlet region.

21. Hide the rotating region and the inlet region, leaving only the outlet region visible.
22. Make the fan casing body visible.
23. Click on **Combine** in the **Modify** panel of the ribbon.

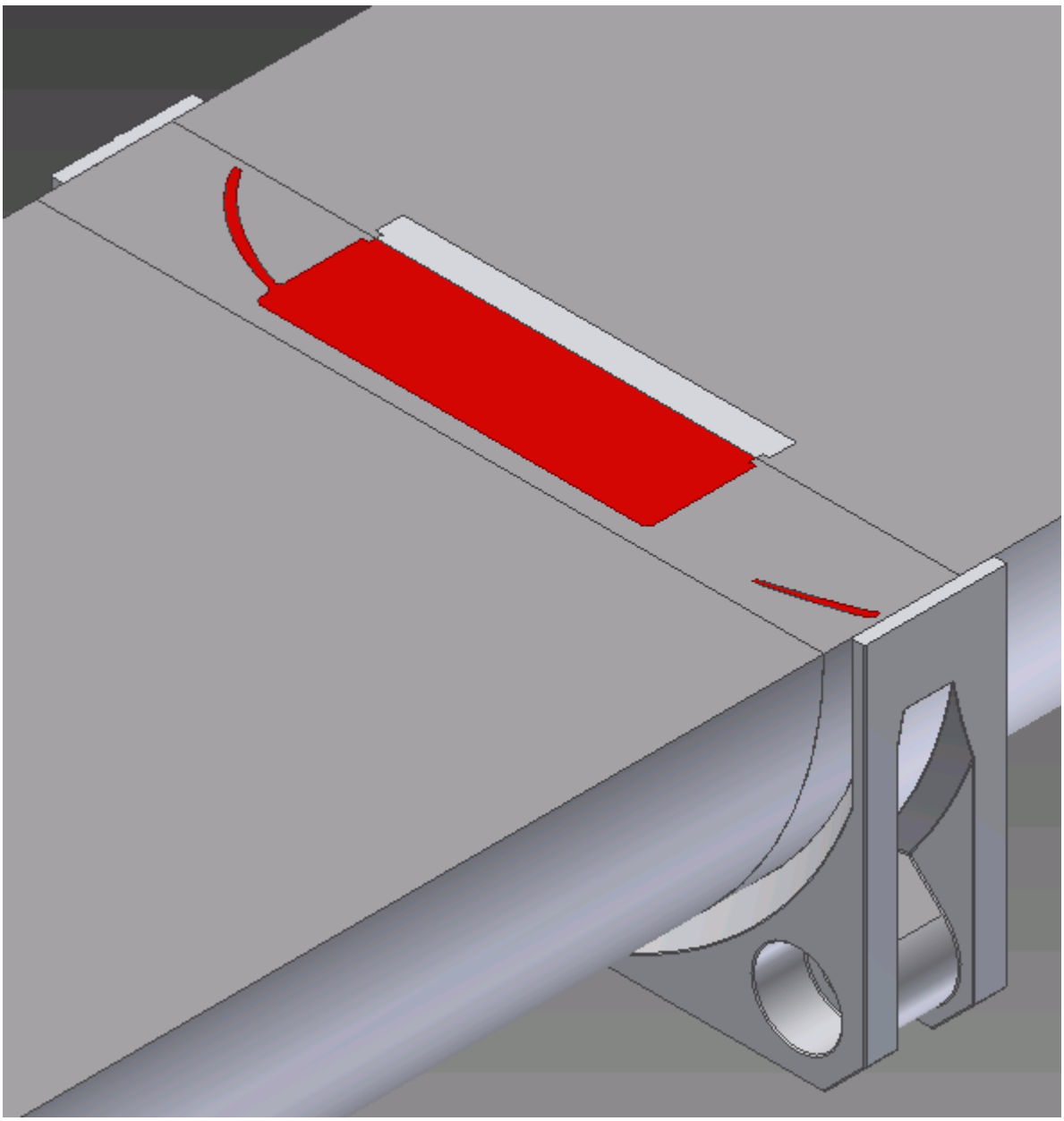


The *Combine* dialog appears.

24. Click on the (**Cut**) button, select the outlet region as the *Base*, and the fan casing as the *Toolbody*.
25. In the *Combine* dialog, tick the *Keep Toolbody* checkbox to keep the toolbody.

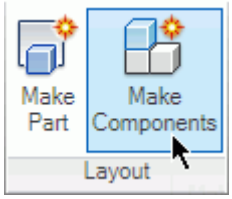
    The toolbody will be hidden automatically after the operation completes.
26. Show the rotating region and the inlet region.

The multi-region fluid domain is complete. A cross-section of the fluid domain is shown below (note that the fan and fan casing were set to visible for this screenshot), where the fan is shown in red, the fan casing is shown in white, and the multi-region fluid domain is shown in grey.



The last step is to export the bodies from this multi body part into an assembly, where they can be selected for use in STAR-Inventor.

27. Save the current part file using a suitable name.
28. Make the original assembly components visible by right-clicking on these in the **Solid Bodies** node in the feature tree and selecting **Visibility**.
29. Click on the *Manage* tab of the ribbon.
30. Click on the **Make Components** button in the **Layout** panel of the ribbon.



The *Make Components : Selection* dialog appears.

31. Select the internal fluid volume and the original components from the geometry window.

    These will be added to the dialog.

32. Click **Next**.

    The *Make Components : Bodies* dialog appears.

33. Give the new assembly (and parts) a suitable name and click **OK**.

    The new assembly will be opened automatically.

34. Save the new assembly.

The derived part is associative, meaning that if the original part/assembly is modified, the derived part can be updated to reflect this. The up-to-date derived part can then be exported to an assembly, as described above, where it can be used with STAR-Inventor.

The geometry is now ready for cad client analysis. Suppressed parts will not be included in the analysis. The three regions that form the fluid domain all have a conformal match, so it will be possible to create the interfaces and obtain a topologically valid mesh in STAR-CCM+ without any further modifications to the geometry.

Imprinting Parts

Most CAD model assemblies are not suitable for meshing as the parts in the assembly are disjointed, where the coincident faces are overlapping. Attempting to generate a mesh with such an assembly will result in a meshing error. To rectify this problem, the parts in an assembly must be imprinted together.

Imprinting can also be used to split surfaces for boundary region applications. An example of such an application would be to model the area that a fan would occupy on the inside of computer case, such as in the third STAR-Inventor tutorial, where a smaller section of the surface is needed to represent the fan, rather than the entire inner surface of the computer case.

Contents:

Imprinting Parts in an Assembly
Imprinting Surfaces on Parts for Boundary Region Applications

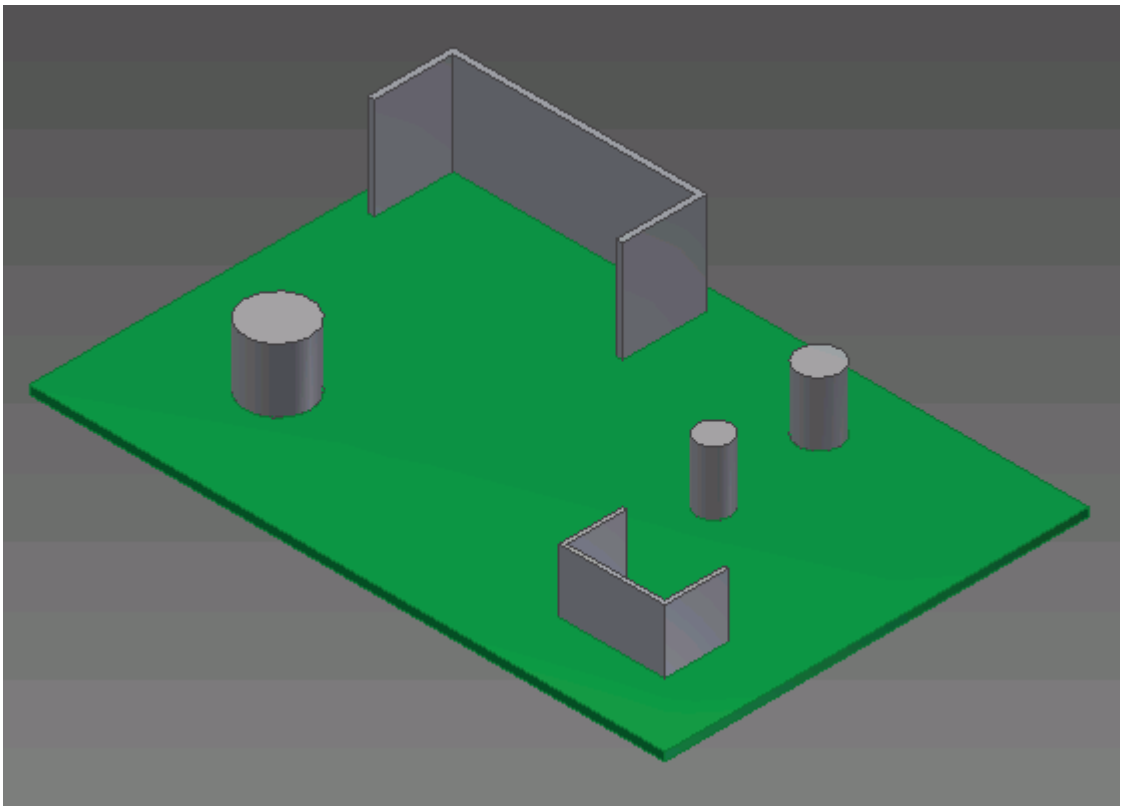
Imprinting Parts in an Assembly

In cases containing multiple adjacent parts, such as in Conjugate Heat Transfer cases, it is essential to imprint the profiles of the coincident faces of the smaller parts onto the face of the larger part. This will ensure that the volume mesh created by STAR-CCM+ contains interfaces between the adjacent parts.

In general, the steps to follow in creating the fluid domain for multiple regions are:

- Work within the assembly, making the part that requires imprinting the work part.
- Create a sketch on the face you want to imprint, and project the contact edges of the other components in the assembly onto this sketch.
- Split the face you want to imprint using the sketch created in the previous step.

The simple circuit board shown below requires imprinting before the mesh can be generated.

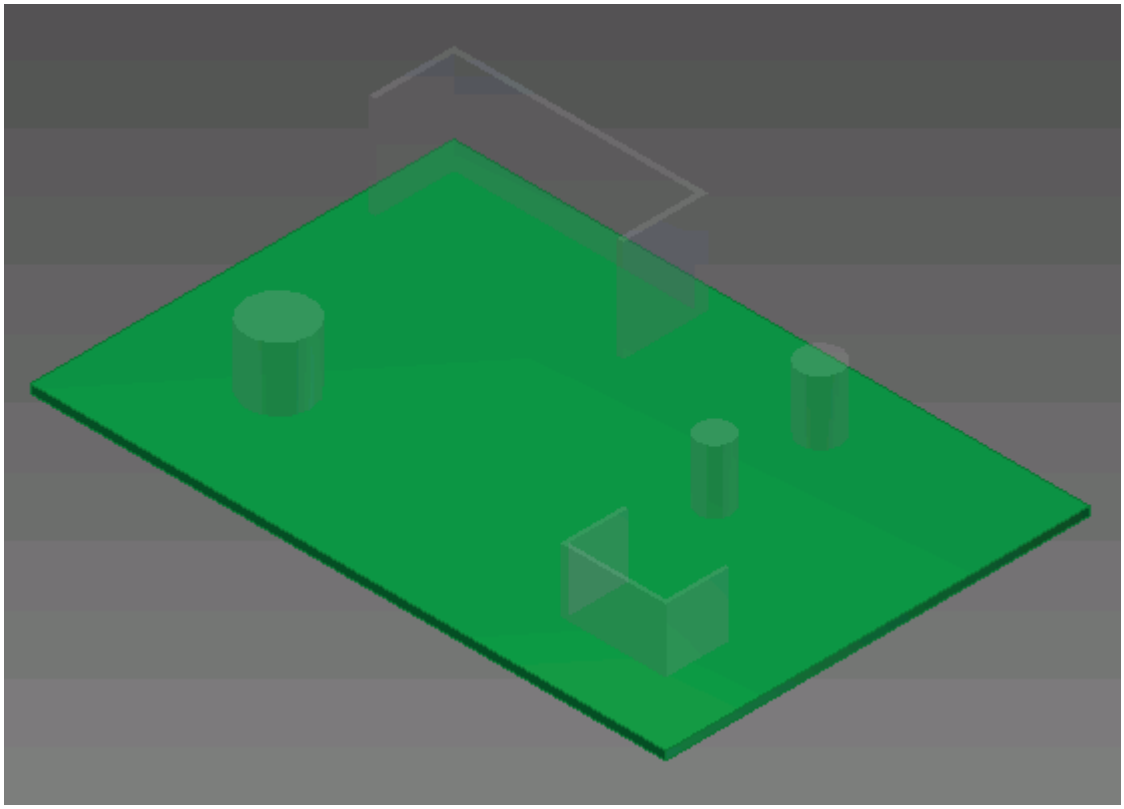


In this example, the capacitors and heat sinks are not imprinted to the circuit board. This would result in a failure when attempting to generate a mesh.

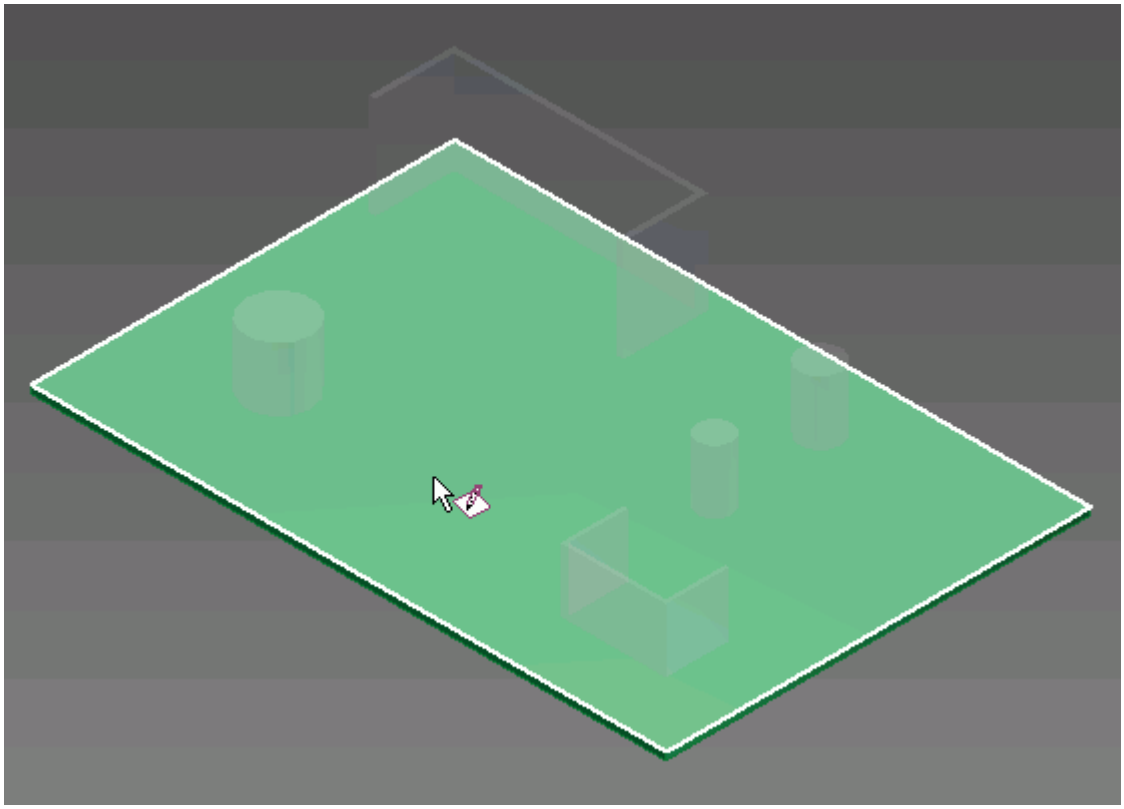
To imprint the capacitors and heat sinks onto the circuit board follow the steps outlined below:

1. Load the assembly into Inventor.
2. Double-click on the node of the part that you want to imprint.

   This will make it the active part; all other parts in the assembly will become transparent.

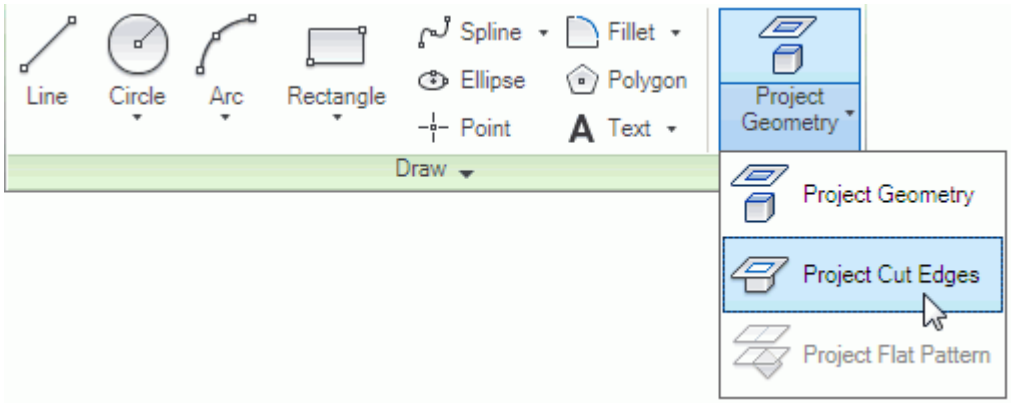


3. Create a sketch on the face that requires imprinting.

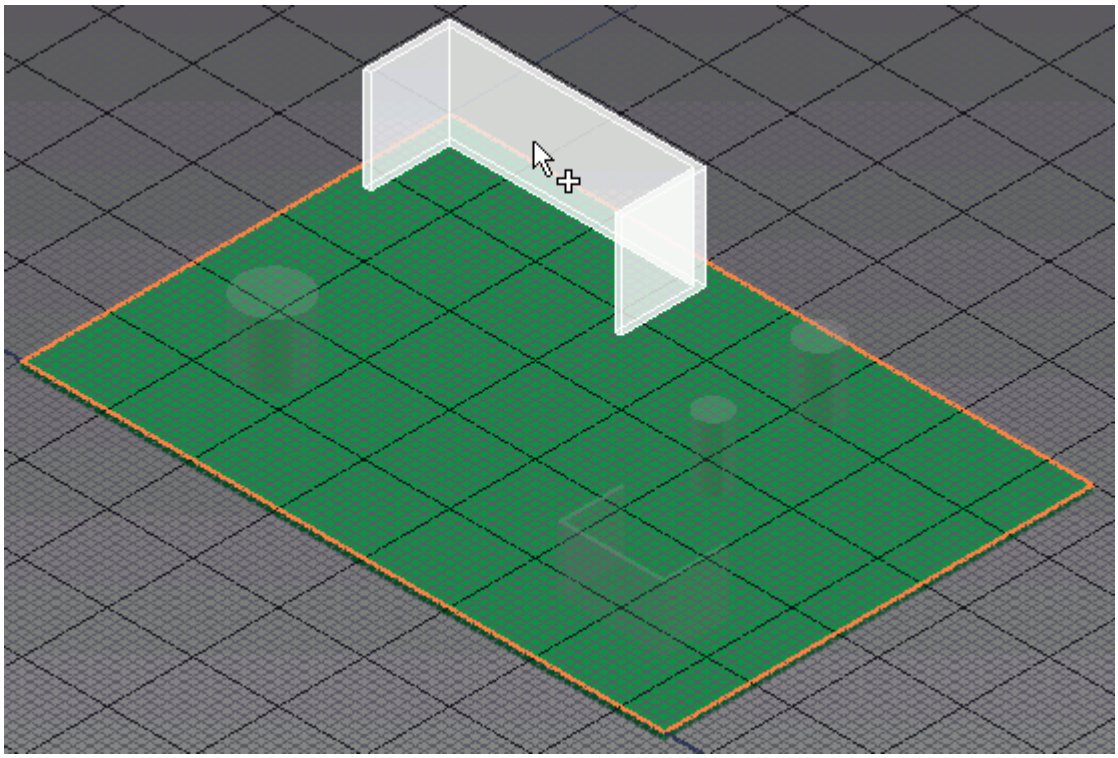


We will now project the intersecting edges of the other components onto the active sketch.

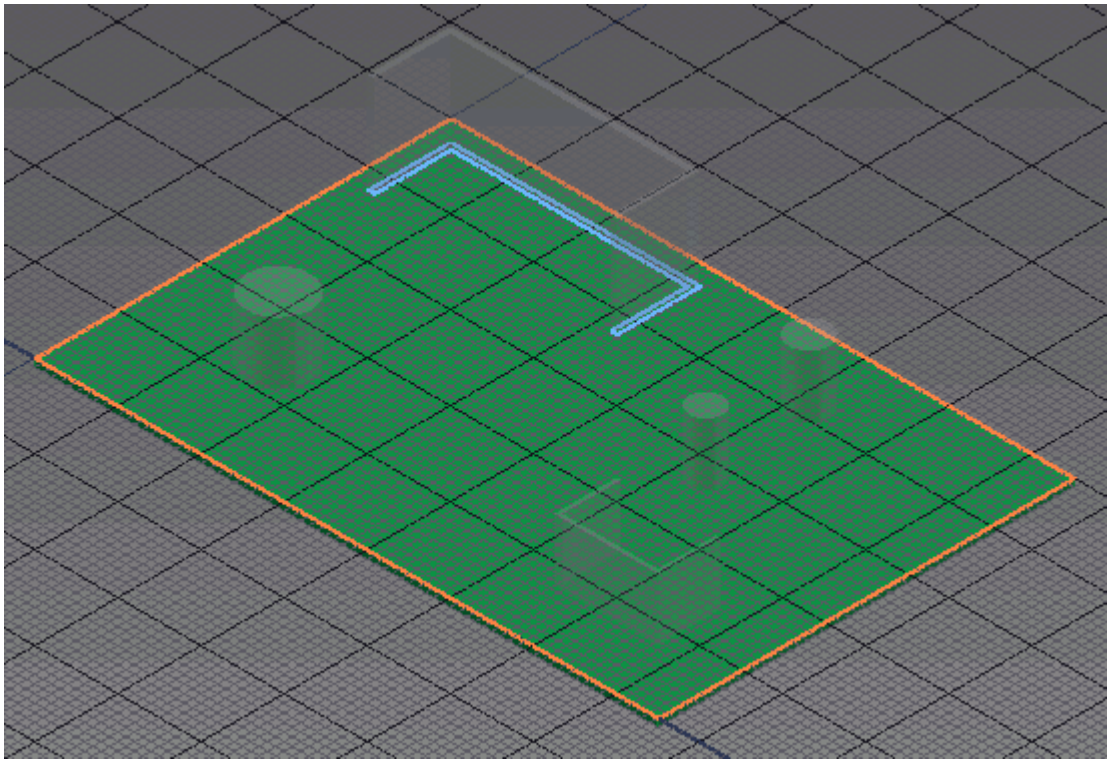
4. Click on the **Project Cut Edges** button in the **Draw** panel of the ribbon.



5. Hover the mouse pointer over one of the transparent parts, this will become highlighted in white--click to select it.



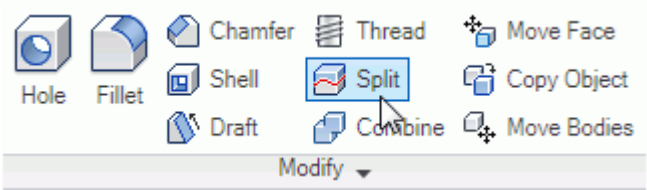
The contact edges of the part are added to the sketch; these appear colored in blue.



6. Exit the sketch when done.

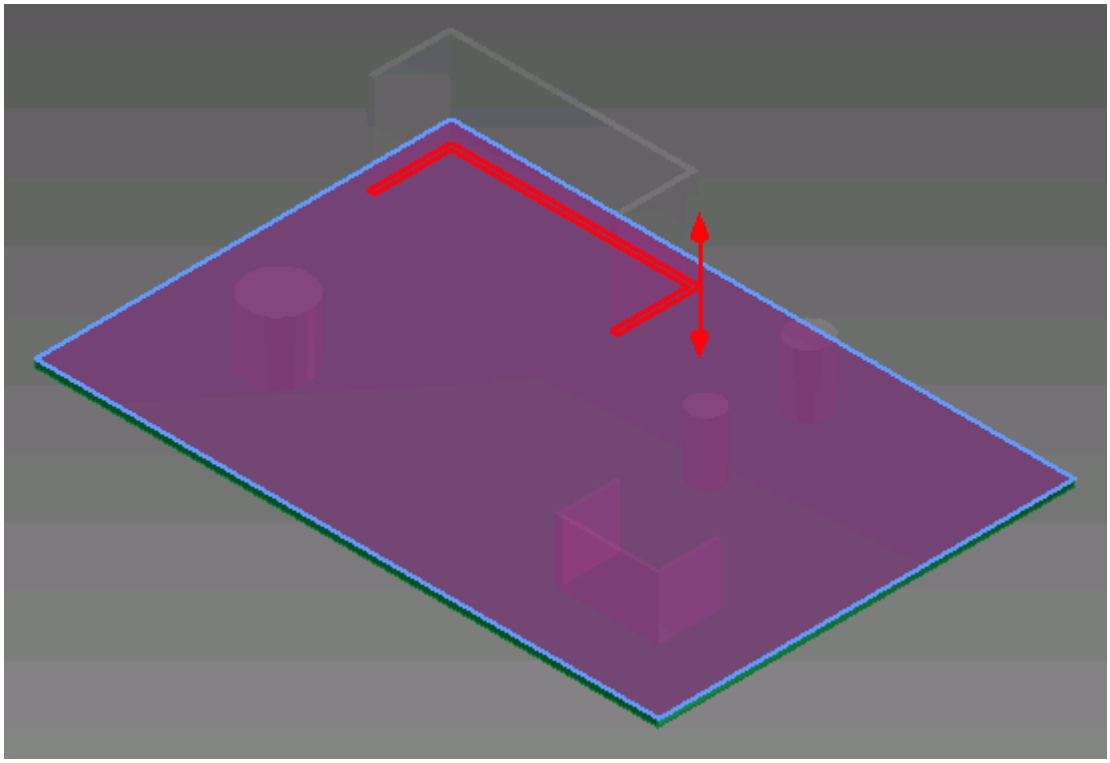
We will imprint the face with the sketch containing the contact edges of the other components.

7. Click on the **Split** button in the **Modify** panel of the ribbon.



The *Split* dialog appears.

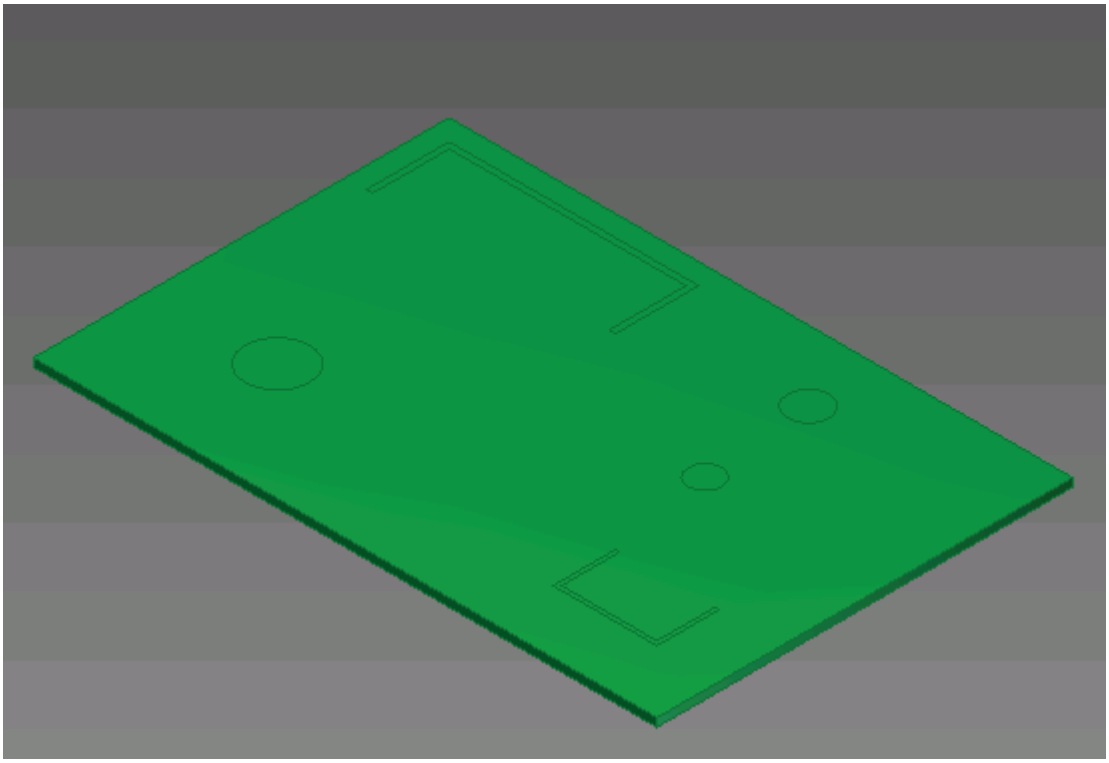
8. Select the sketch containing the contact edges to act as the *Split Tool*.



Note that the sketch containing the profile will be consumed by this operation.

9. Select the face you want to imprint using the **Faces to be Split** tool.
10. Repeat the above steps (obtaining the profiles and splitting the face) for all the other components in the assembly.
11. Return to the assembly by double-clicking on the root node in the feature tree.

To verify that the face was correctly imprinted, we can hide the other parts by right-clicking on the imprinted part and selecting **Isolate**.



A high quality mesh with fully defined interfaces can now be successfully generated for this assembly. Suppressed parts are not included in the STAR-CCM+ analysis, so for CHT cases, such as the above example, ensure that all relevant parts are active.

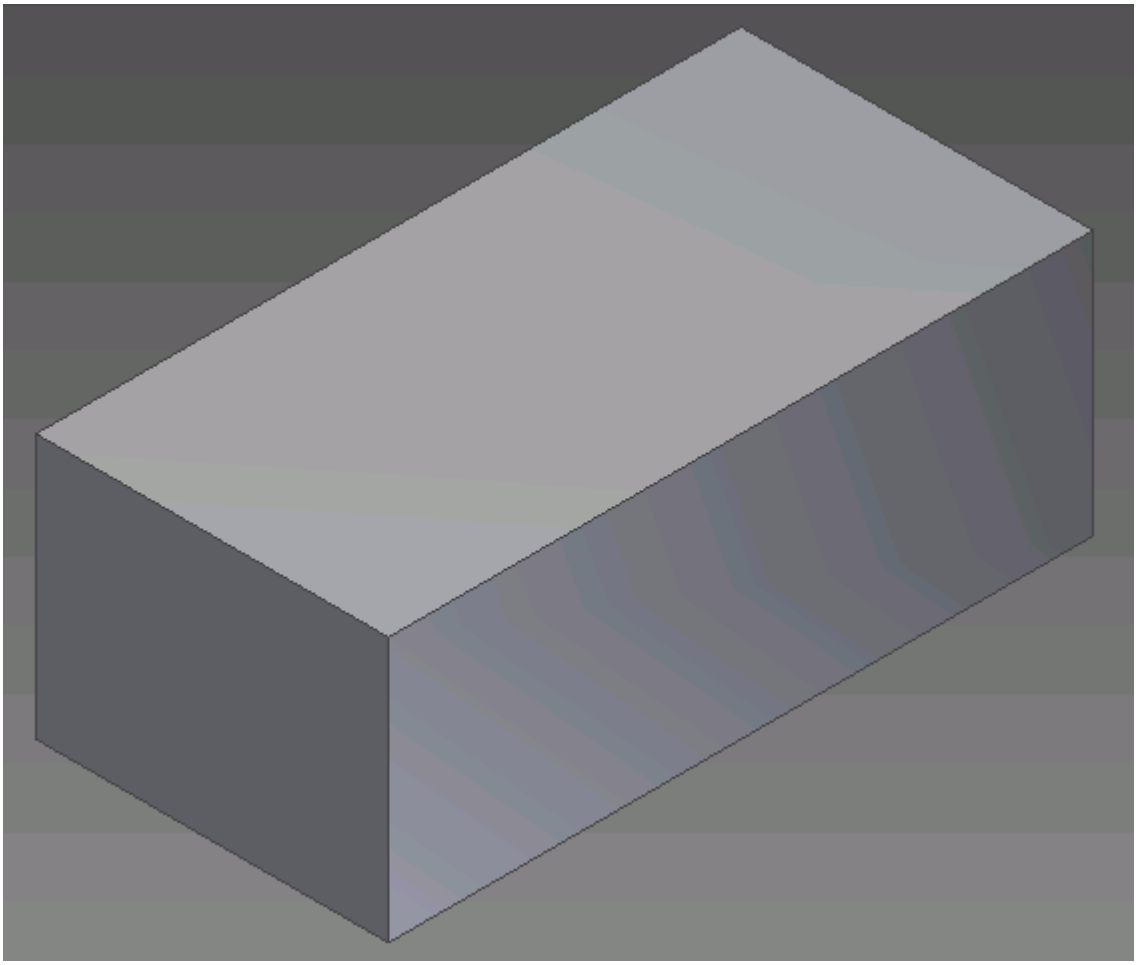
Imprinting Surfaces on Parts for Boundary Region Applications

Faces in a part body can be split into smaller faces for boundary application.

The general procedure to follow is:

- Create a sketch on the surface you want to split. The shape of the sketch should correspond to the shape of the new face after it is split.
- Split the face that requires imprinting with the sketch containing the profiles.

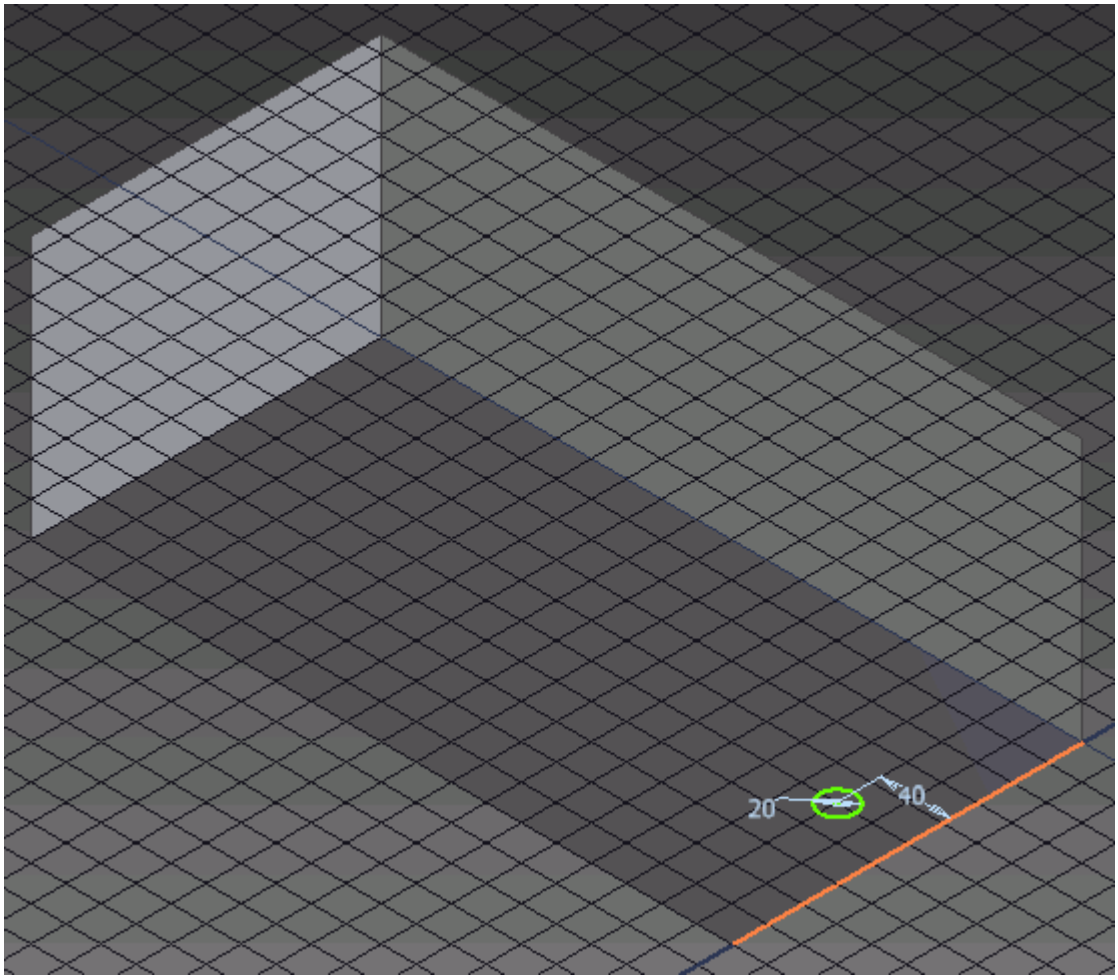
A simple example of an aeration tank is shown below. The image shows the tank, the two air injectors need to be imprinted onto the base of the tank.



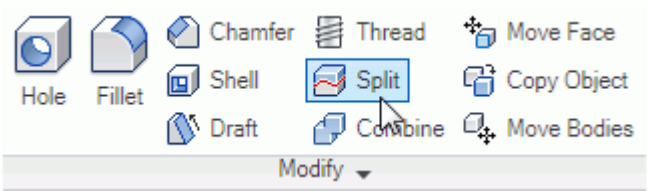
1. Load the part into Inventor.
2. Create a sketch on the face that requires imprinting.

   By default Inventor will automatically project edges that lie on the sketch plane--these will appear in orange. Delete these sketch entities.
3. Draw the profile of the faces you want to imprint.

   In this example, the air injector profiles are circular in shape.

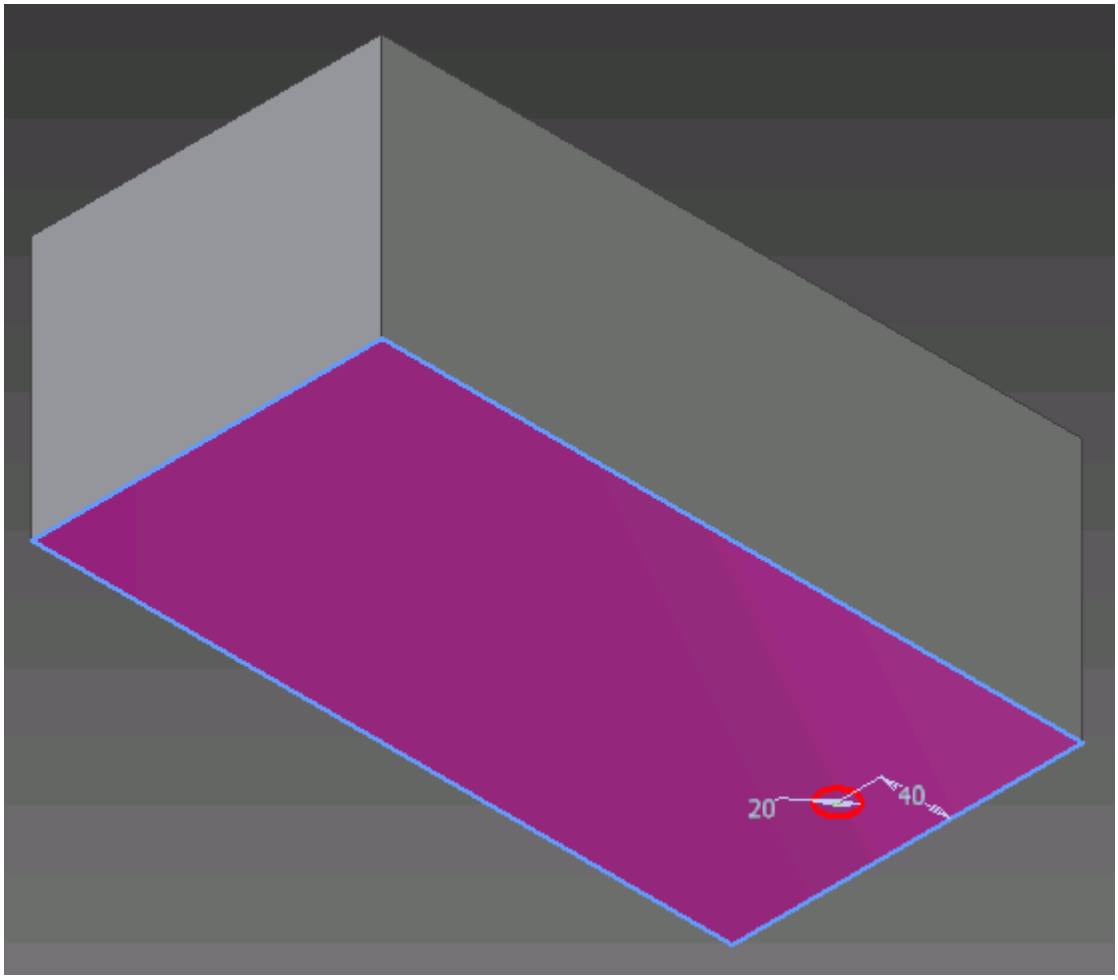


4. Click on the **Split** button in the **Modify** panel of the ribbon.



The *Split* dialog appears.

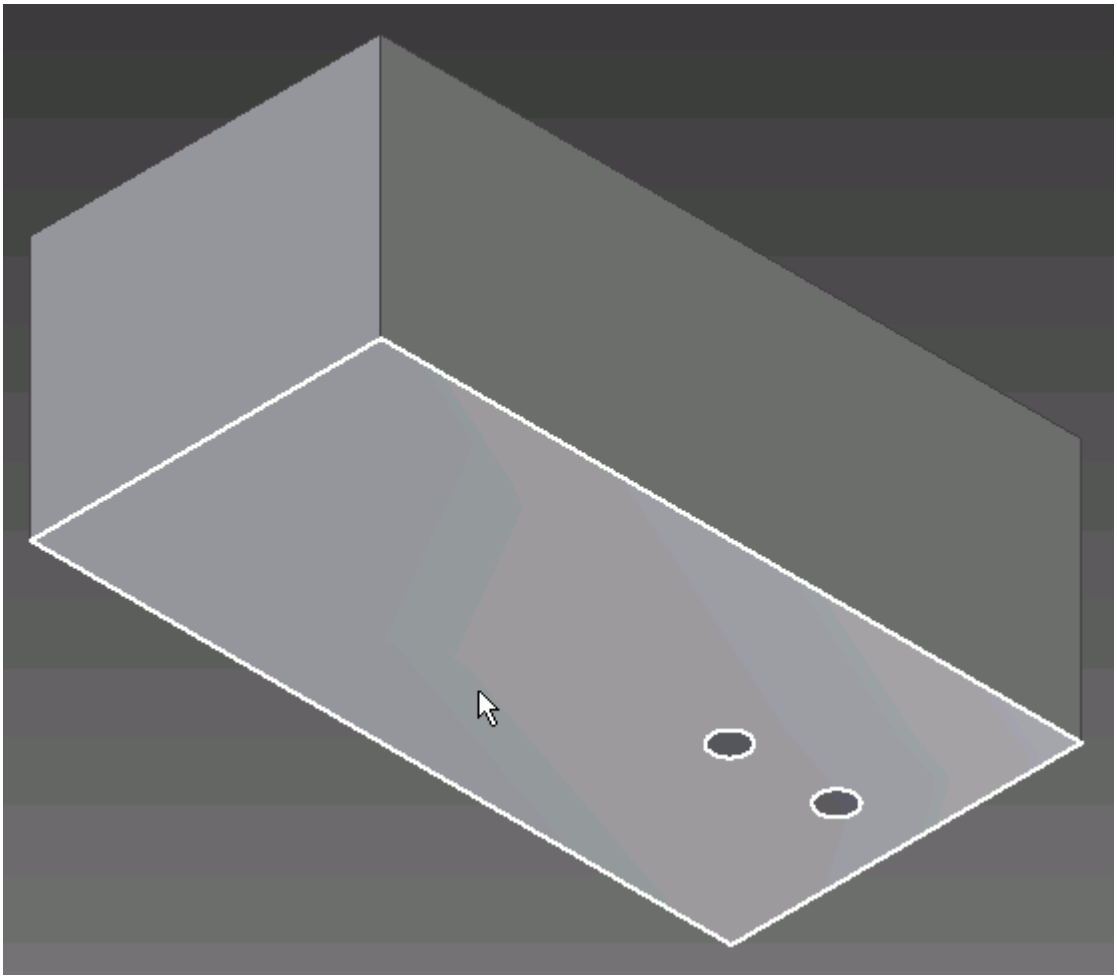
5. Select one of the profiles you created in the previous step to act as the *Split Tool*.



Note that the sketch containing the profile will be consumed by this operation.

6. Select the face you want to imprint using the **Faces to be Split** tool.
7. Repeat the above steps (creating the profiles and completing the split) for all the new boundaries you want to add.

The face is now imprinted; each face can be selected independently of each other, and can be used when defining boundaries.



The separate faces can now be defined as boundary regions.

STAR-Inventor Tutorials

Several tutorials are provided for the STAR-Inventor CAD Client.

These tutorials provide an overview of the major features available in the client and illustrate some of the applications where you can use STAR-Inventor.

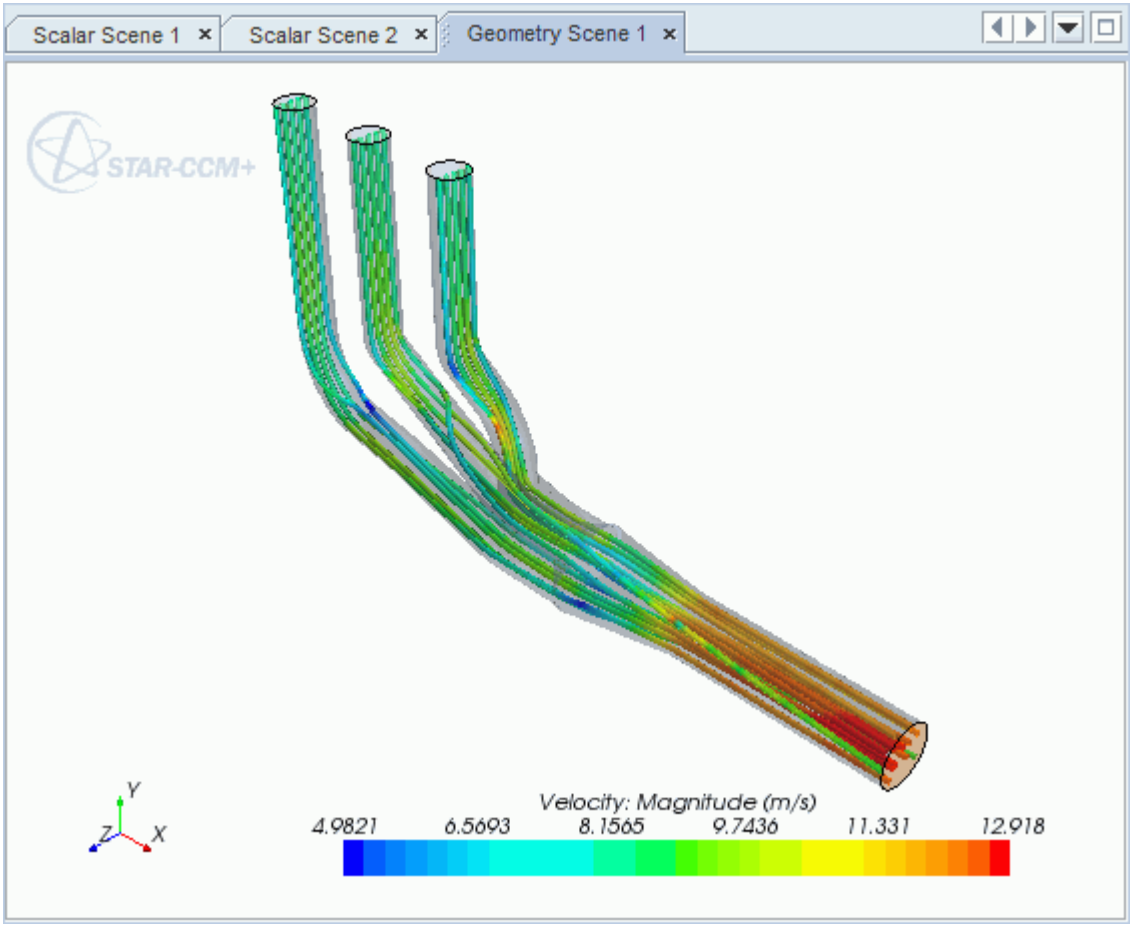
Contents:

Cold Flow Through a Motorcycle Exhaust Manifold
Porous Media Flow
Aircraft Wing Deicing System
Rotating Flow Inside a Graphics Card
External Flow Over a Car
Geometry Transfer: CAD to Simulation

Cold Flow Through a Motorcycle Exhaust Manifold

This tutorial demonstrates how to setup and run a CFD calculation in the STAR-Inventor environment.

In this tutorial, the model is an exhaust manifold from a three-cylinder motorcycle engine.



Contents:

Opening the CAD Model
Selecting Fluid Physics Models
Defining Boundary Conditions
Setting the Mesh Pipeline
Modifying Solution Controls
Reviewing the Simulation Set Up
Generating the Surface Mesh
Generating the Volume Mesh
Building and Running
Post-Processing Results

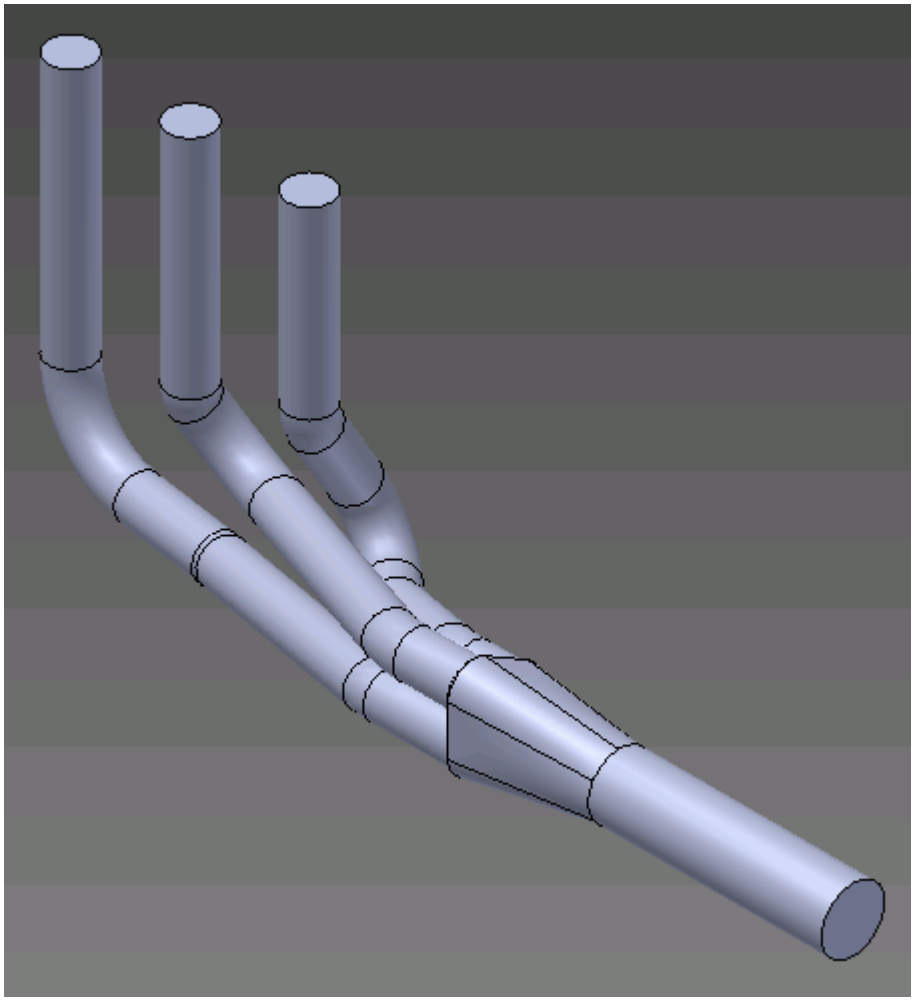
Opening the CAD Model

A directory containing the CAD geometry is provided with the STAR-CCM+ distribution.

This directory is located in `[STAR-CCM+_INSTALL_DIR]\doc\startutorialsdata\starinventor\data`.

1. Copy directory `data` to your own disk space and open the part file `exhaustManifold.ipt` in STAR-Inventor.

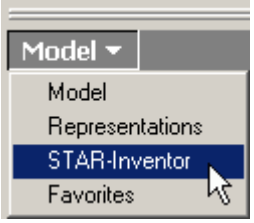
   The imported geometry is shown below.



The STAR-Inventor ribbon appears at the top of the screen.The STAR-Inventor object tree is located on the left of the screen.

To switch to this object tree:

2. Click **Model** and select **STAR-Inventor**.

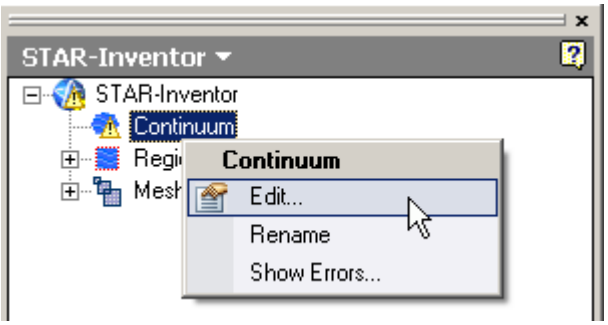


If these items are not visible, refer to Launching STAR-Inventor Manually.

Selecting Fluid Physics Models

The first step in setting up this tutorial is to define the physics of the problem. For this case, the flow is turbulent and isothermal.

1. Right-click the **Continuum** node in the STAR-Inventor tree and select **Edit**.



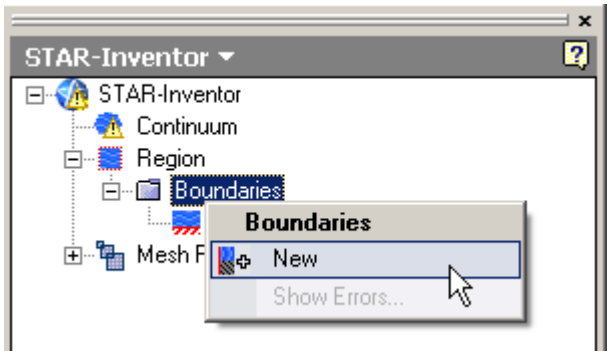
   The *Continuum Settings* dialog appears.
2. Deactivate the **Temperature** option in the *Optional Physics Models*.
3. Click **OK**.

Defining Boundary Conditions

You define the inlet and outlet boundary conditions for the simulation.

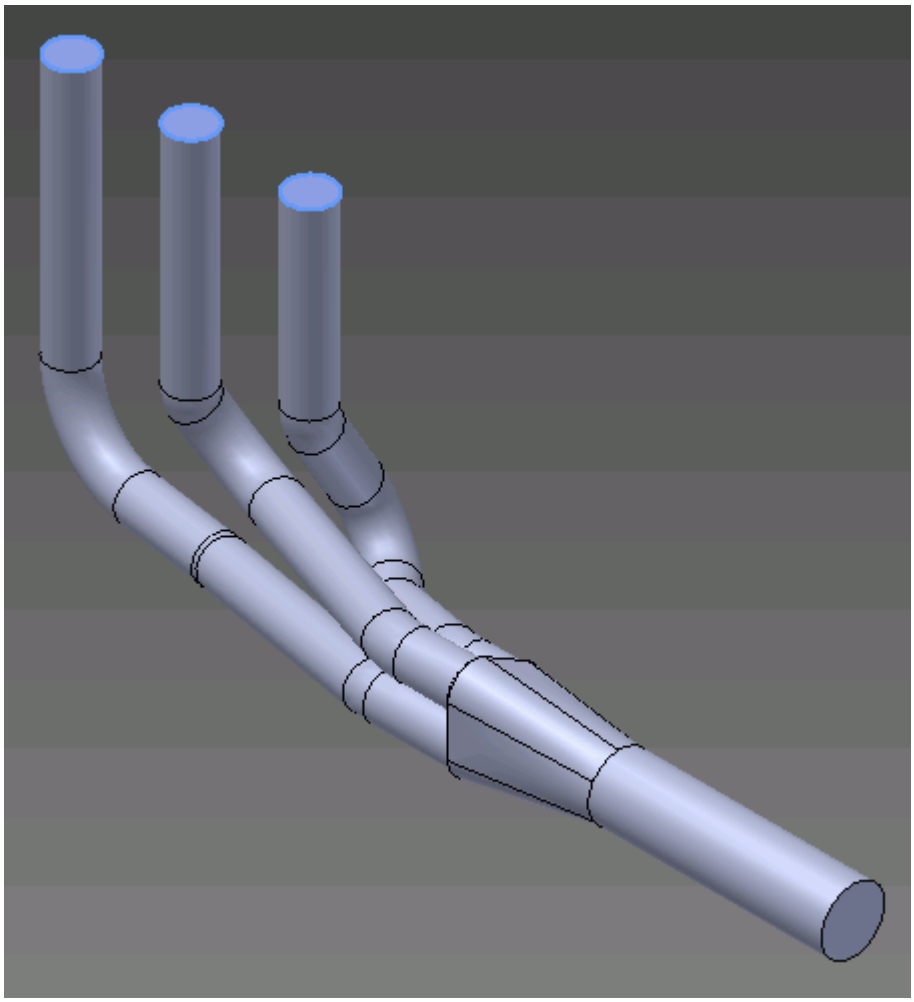
Create the inlet boundary first:

1. Right-click the **Region** > **Boundaries** node and select **New**.



The *Boundary Settings* dialog appears.

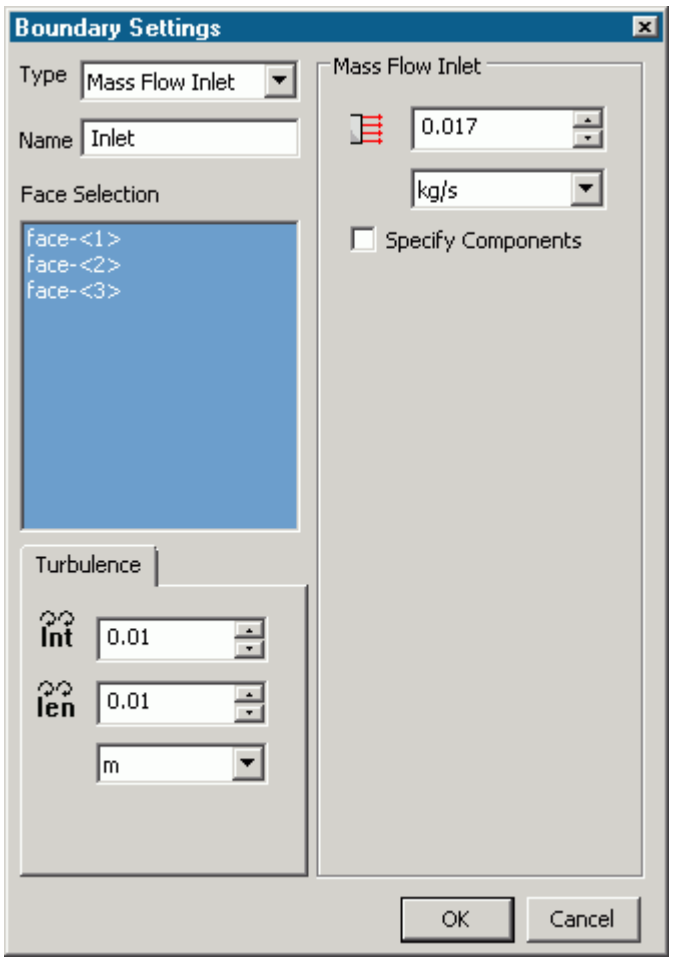
2. On the CAD model, select the three faces that are highlighted below.



These selections are reported in the *Face Selection* box in the *Boundary Settings* dialog.

3. Set *Type* to **Mass Flow Inlet**.
4. Set *Boundary Name* to `Inlet`.
5. Set (*Mass Flow Rate*) to `0.017` kg/s.

The completed dialog is shown below.



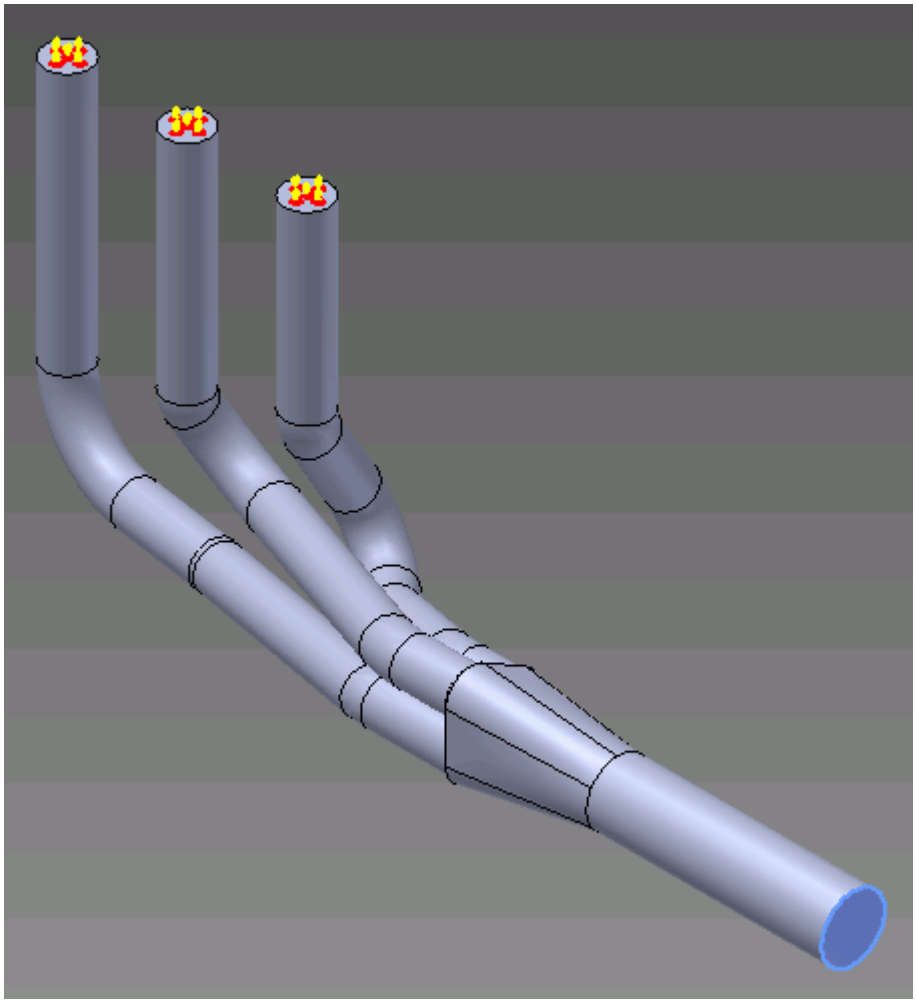
6. Click **OK**.

Create the outlet boundary:

7. Right-click the **Region** > **Boundaries** node and select **New**.

   The *Boundary Settings* dialog appears.

8. Select the face that is highlighted below.



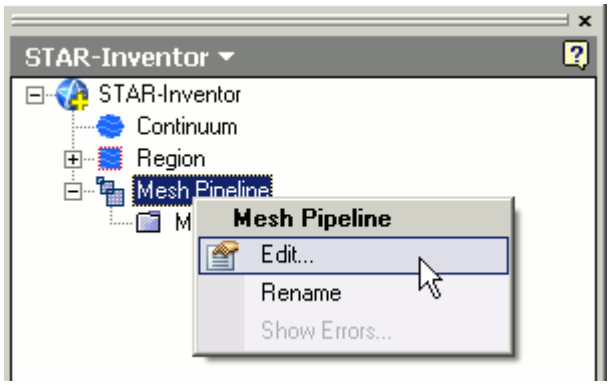
9. Set *Type* to **Pressure Outlet**.
10. Set *Boundary Name* to `Outlet`.

    The default settings are acceptable for this case.
11. Click **OK**.

Setting the Mesh Pipeline

To obtain more accurate results, modify the mesh pipeline:

1. Right-click the **Mesh Pipeline** node in the STAR-Inventor tree and select **Edit**.



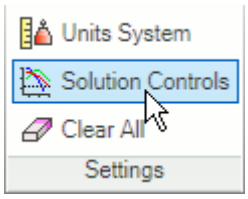
The *Mesh Pipeline Settings* dialog appears.

2. Set (*Base Size*) to `2.5`%.
3. Set (*Prism Layer Thickness*) to `15`%.
4. Click **OK**.

Modifying Solution Controls

Modify the maximum number of iterations and the solution accuracy:

1. Click **Solution Controls** on the STAR-Inventor ribbon.

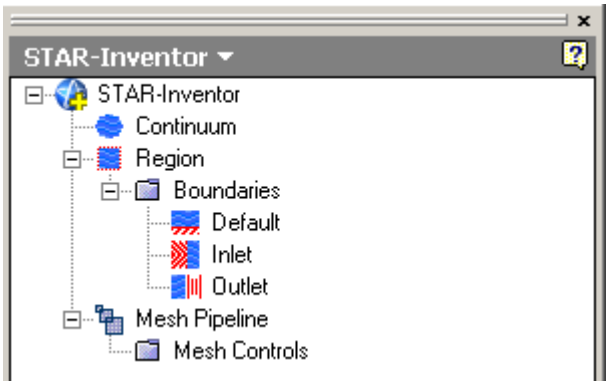


The *Solution Controls* dialog appears.

2. Click the *Advanced* tab.
3. Set (*Number of Iterations*) to `200`.
4. Set (*Convergence Level*) to `0.0001`.
5. Click **OK**.
6. Save your progress.

Reviewing the Simulation Set Up

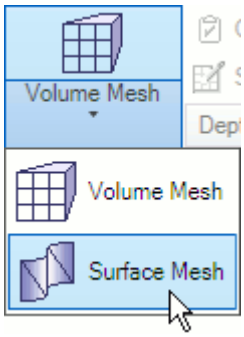
The final STAR-Inventor object tree is shown below.



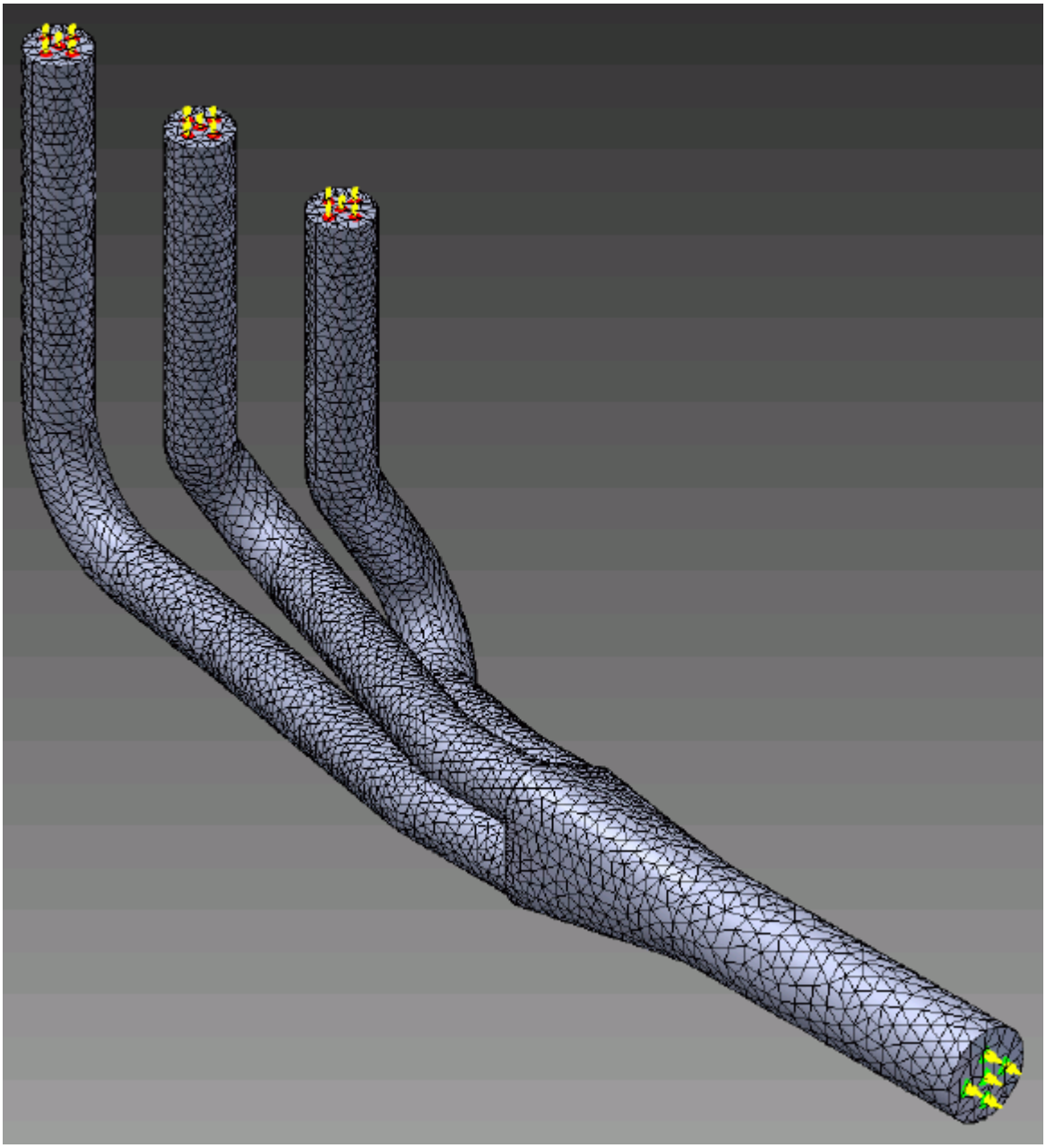
Generating the Surface Mesh

To generate the surface mesh:

1. From the STAR-Inventor ribbon, click **Surface Mesh**.From the STAR-Inventor ribbon, click ▾ next to the **Volume Mesh** icon and select **Surface Mesh**.



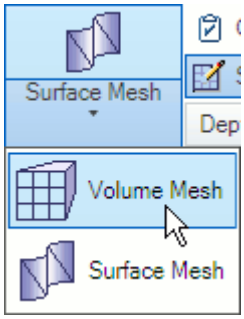
The surface mesh is shown below.



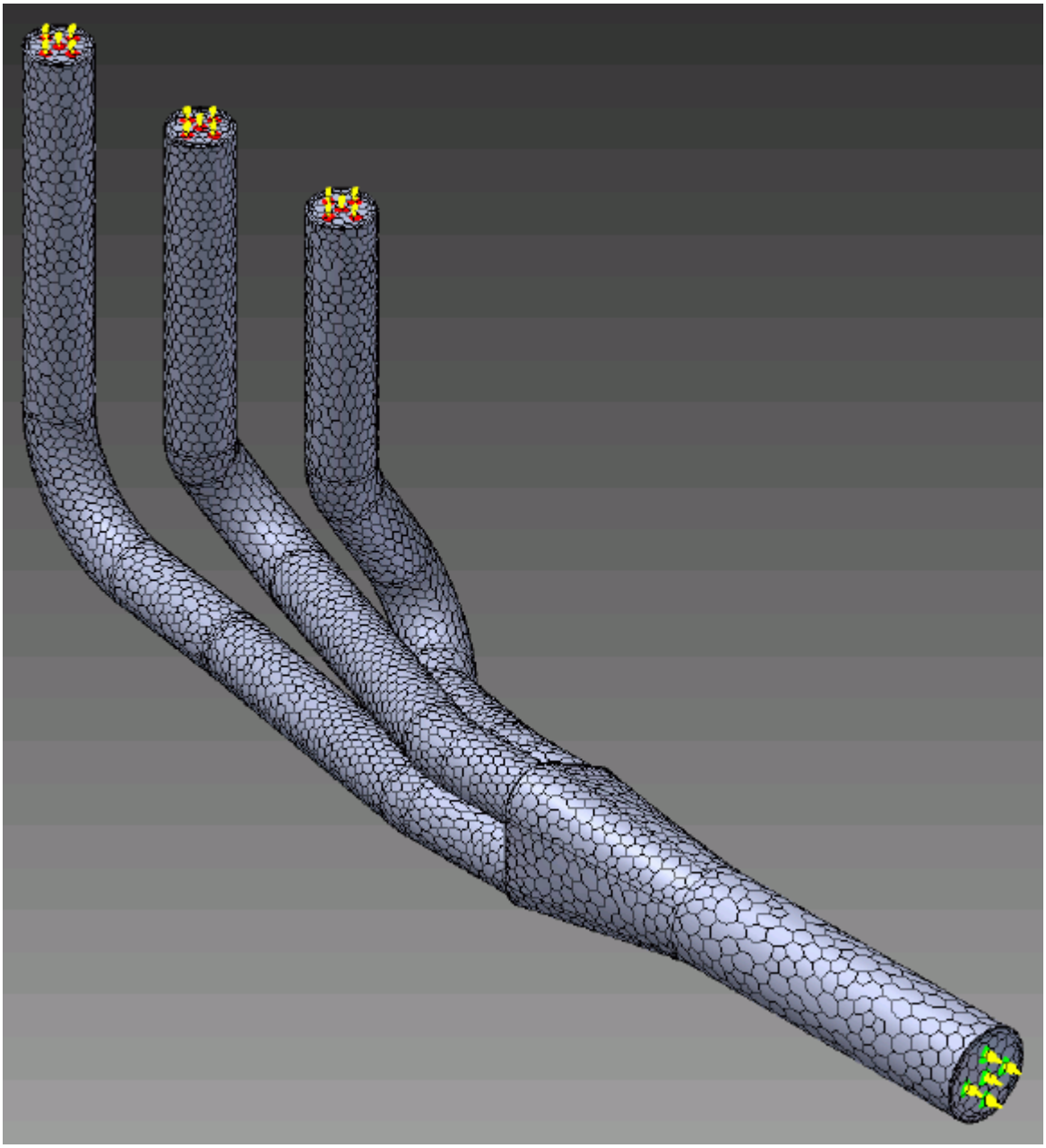
Generating the Volume Mesh

Generate the volume mesh:

1. From the STAR-Inventor ribbon, click **Volume Mesh**.From the STAR-Inventor ribbon, click ▾ next to the **Surface Mesh** icon and select **Volume Mesh**.



The volume mesh is shown below.

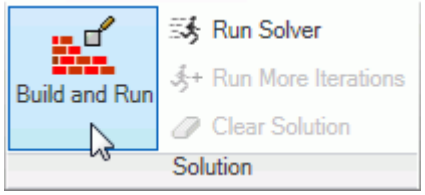


Building and Running

The case is now ready to run.

To run the simulation:

1. From the ribbon, click **Build and Run**.



The *Residuals* plot appears in the STAR-Inventor *Output* window. You can use this plot to monitor the convergence of the solution.

When the solver is done, STAR-CCM+ launches.

Post-Processing Results

You perform all post-processing in STAR-CCM+. By default, a scalar scene, *Scalar Scene 1*, automatically appears in the *Graphics* window when the solver finishes. This scene displays the absolute total pressure.

The results that are contained in the following sections are representative. You may see some variation in the results that you obtain, depending on what platform and package you are using.

1. In STAR-CCM+, click **File** > **Save** to save the simulation.

Contents:

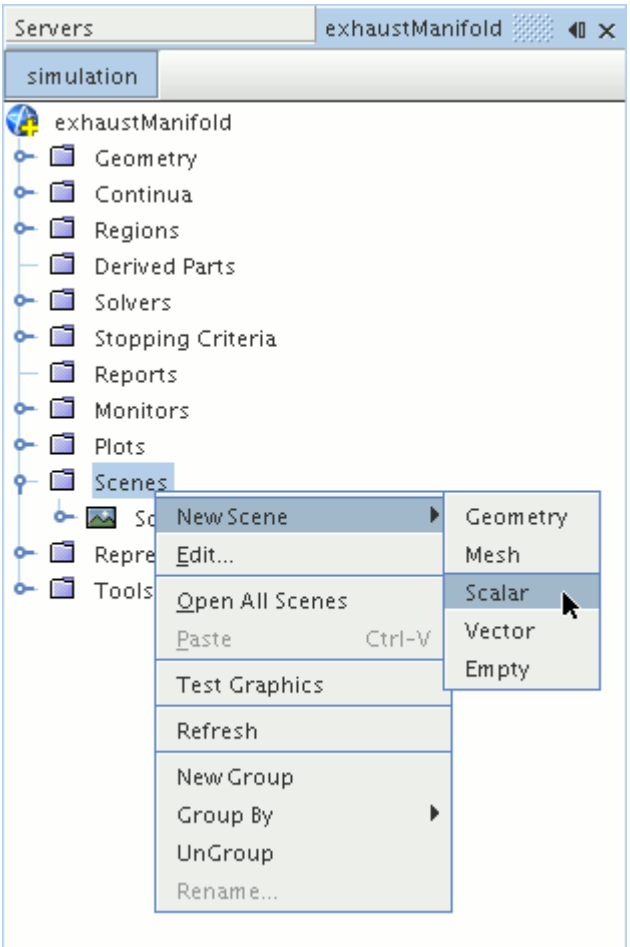
Creating a Scalar Scene
Creating A Streamlines Scene
Creating a Vector Scene
Creating a Streamlines Scene
Creating A Scalar Scene
Visualizing the Wing Skin Temperature
Creating a Streamlines Scene
Creating A Scalar Scene
Creating a Streamlines Scene
Creating a Scalar Scene
Modifying The Scalar Scene
Creating a Streamlines Scene

Creating a Scalar Scene

Create a scalar scene that displays the pressure on a cross-section of the exhaust manifold.

To create the scalar scene:

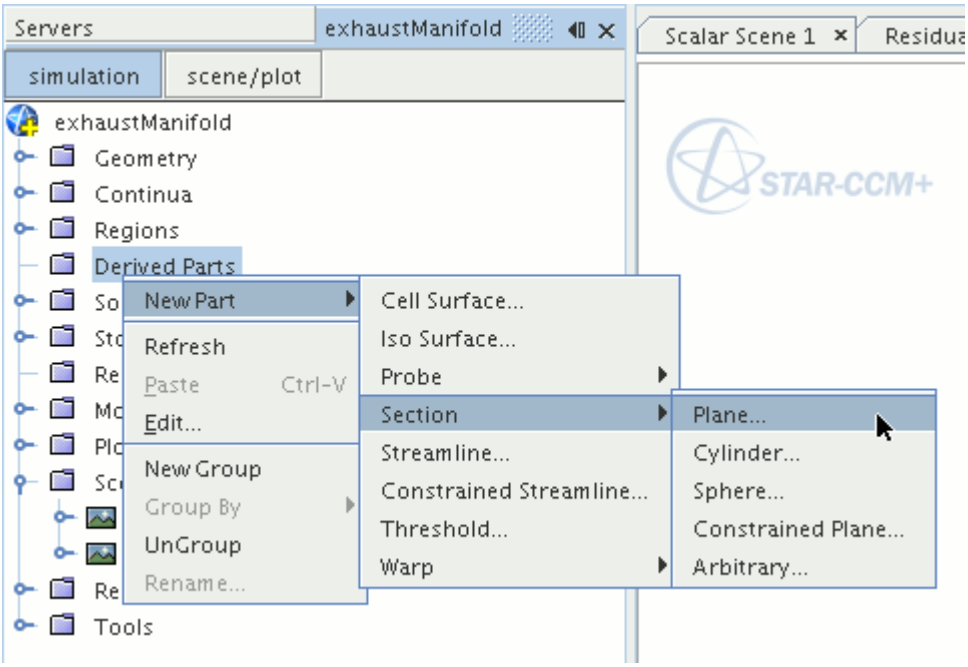
1. Right-click the **Scenes** node in the explorer pane and select **New Scene** > **Scalar**.



A new scalar scene, *Scalar Scene 2*, is created and selected in the *Graphics* window.

Create the section plane:

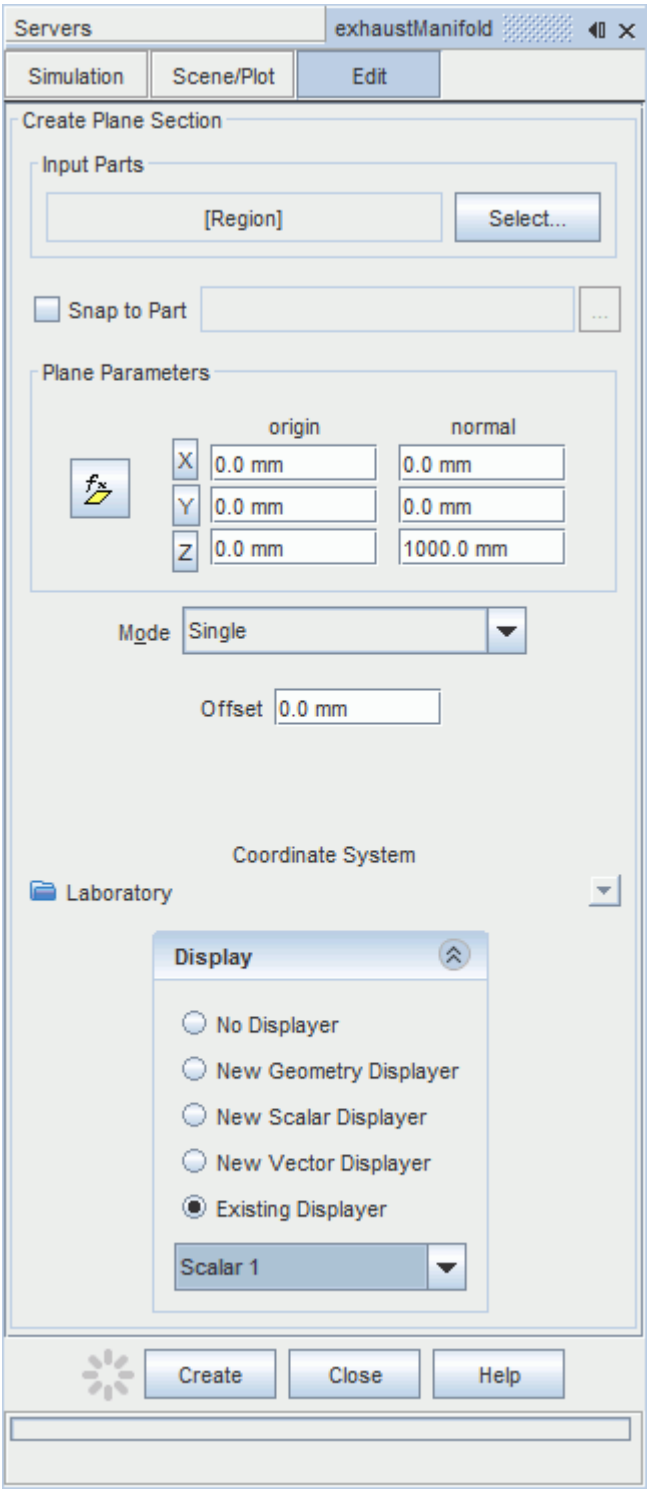
2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane...**



The *Create Plane Section* dialog appears.

3. In the *Plane Parameters* box, set the *origin* vector to `[0 mm, 0 mm, 0 mm]`
4. Set the *normal* vector to `[0 mm, 0 mm, 1000 mm]`
5. In the *Display* group box, select **Existing Displayer** and in the drop-down menu select **Scalar 1**.

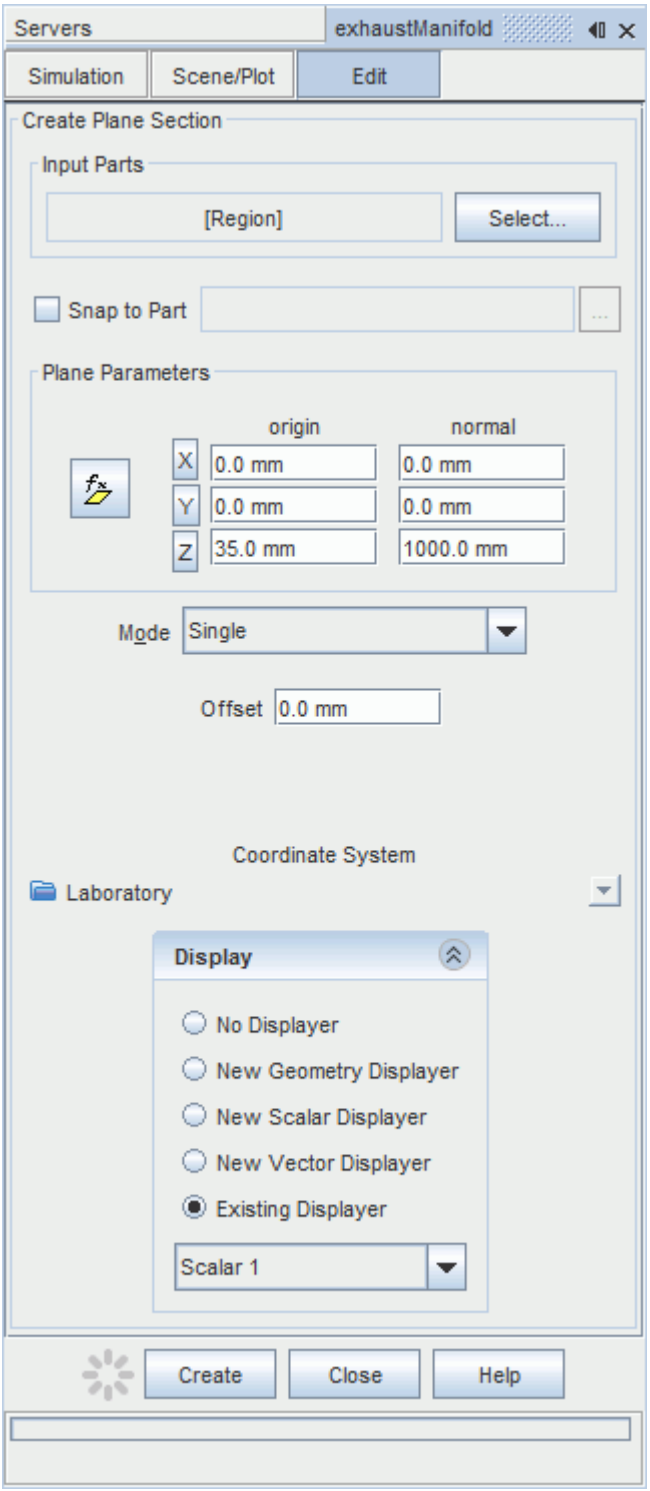
   The completed dialog is shown below.



6. Click **Create**.

Create the second section plane using the same dialog:

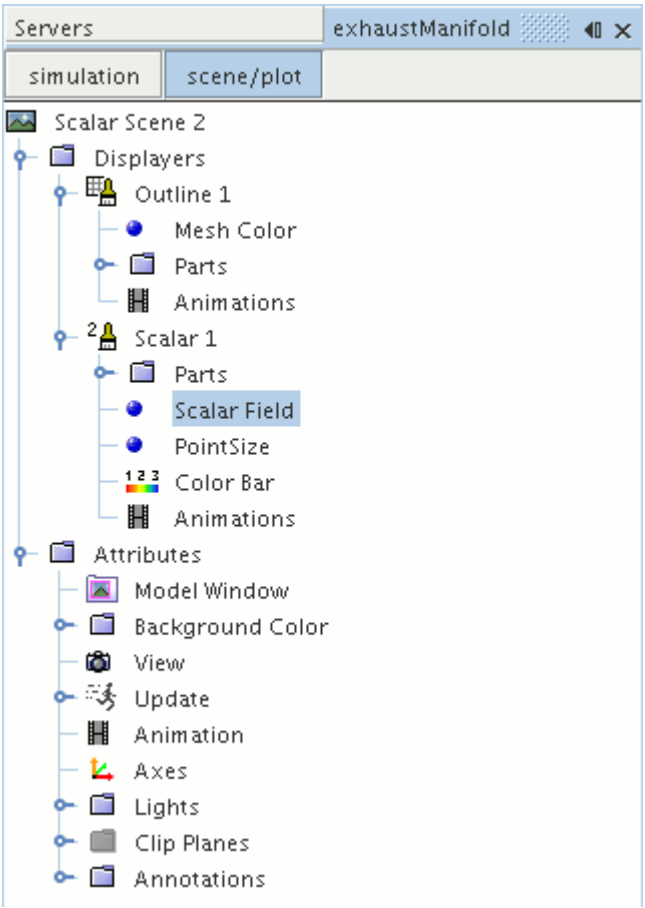
7. In the *Plane Parameters* box, set *origin* to `[0 mm, 0 mm, 35.0 mm]` and leave *normal* as `[0 mm, 0 mm, 1000 mm]`.



8. Click **Create** and then **Close**.

Define the scalar field to display the pressure contours:

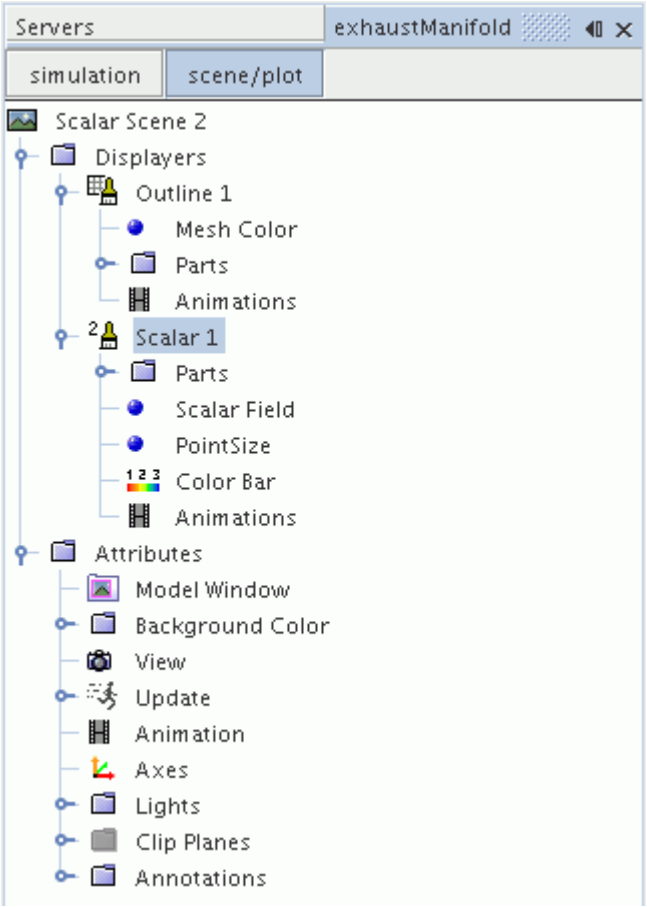
9. Click the **scene/plot** button in the explorer pane.
10. Select the **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Scalar Field** node.



11. Set *Function* to **Pressure**.

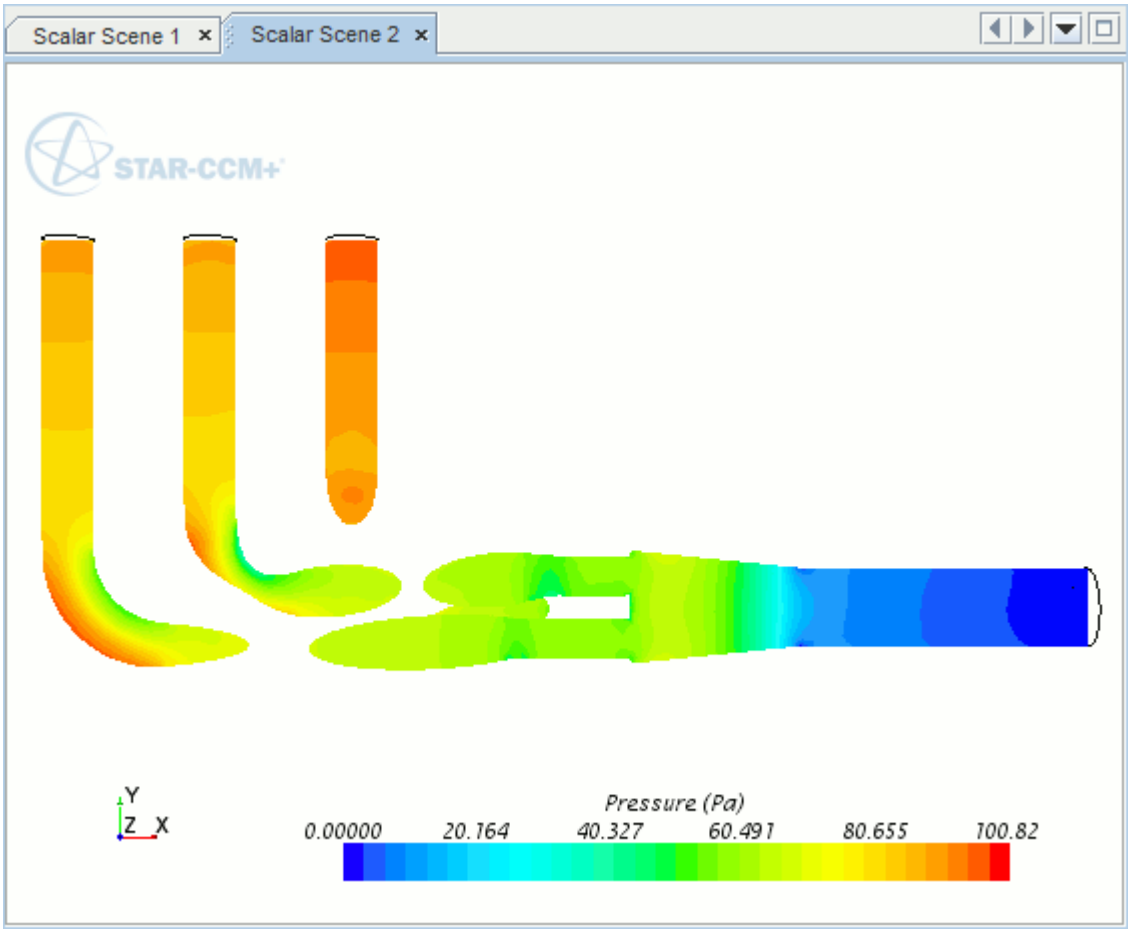
The pressure plot shows the pressure drop inside the exhaust manifold. In order to get a smooth transition between the different pressure values, set the contour style to smooth:

12. Select the **Scalar Scene 2** > **Displayers** > **Scalar 1** node.



13. Set *Contour Style* to **Smooth Filled**.
14. Make sure that the geometry is positioned on its side by pressing the <T> key while *Scalar Scene 2* is active.
15. Click the **simulation** button to return to the main STAR-CCM+ object tree.
16. Click (**Save**) to save the simulation .

The finished scene is shown below.

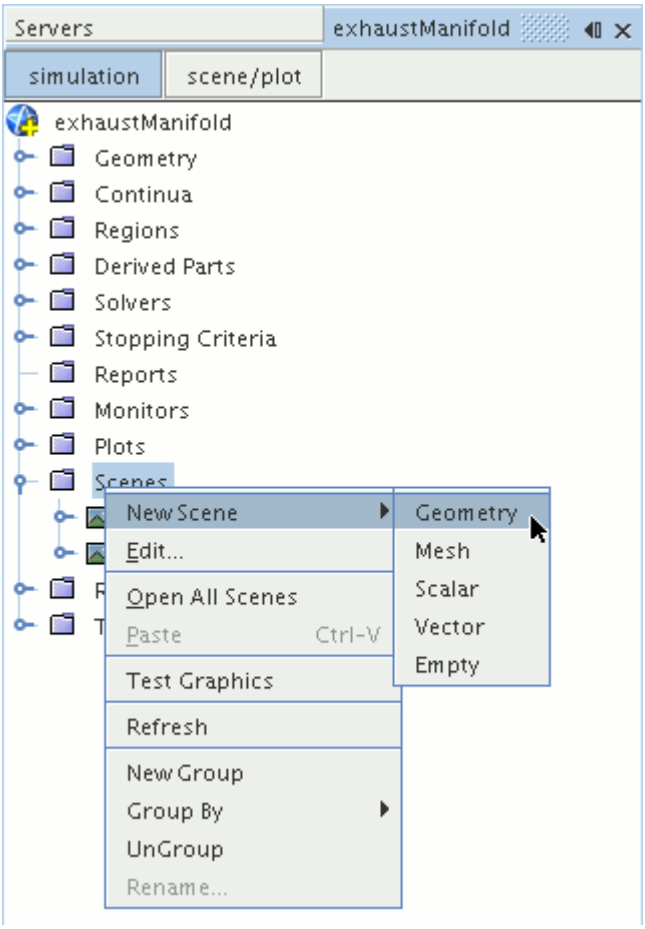


Creating A Streamlines Scene

Create a streamlines scene to get a better understanding of the flow inside the exhaust manifold.

First, create a geometry scene:

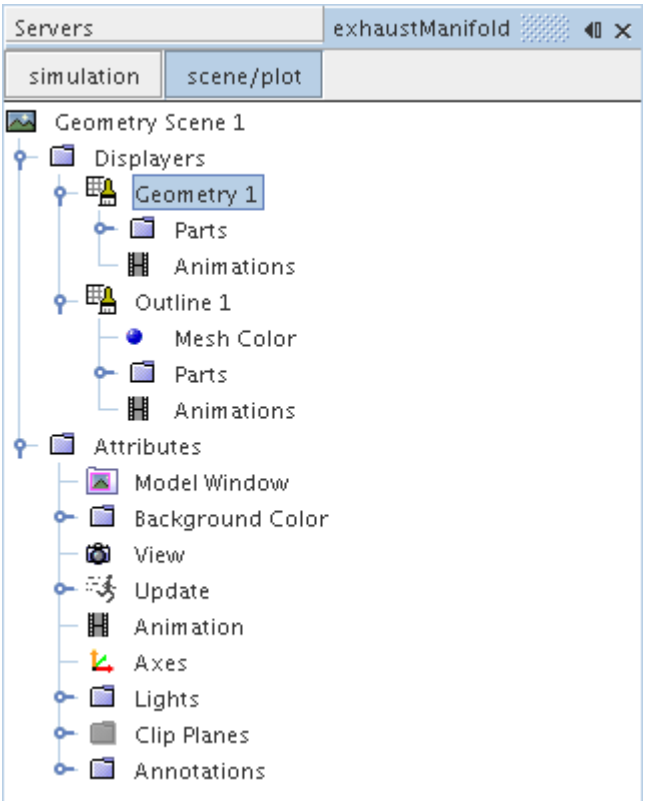
1. Right-click the **Scenes** node in the object tree and select **New Scene** > **Geometry**.



A new geometry scene is created and selected in the *Graphics* window.

Next, modify the opacity of the geometry model in the scene:

2. Click **scene/plot**.
3. Select the **Geometry Scene 1** > **Displayers** > **Geometry 1** node.



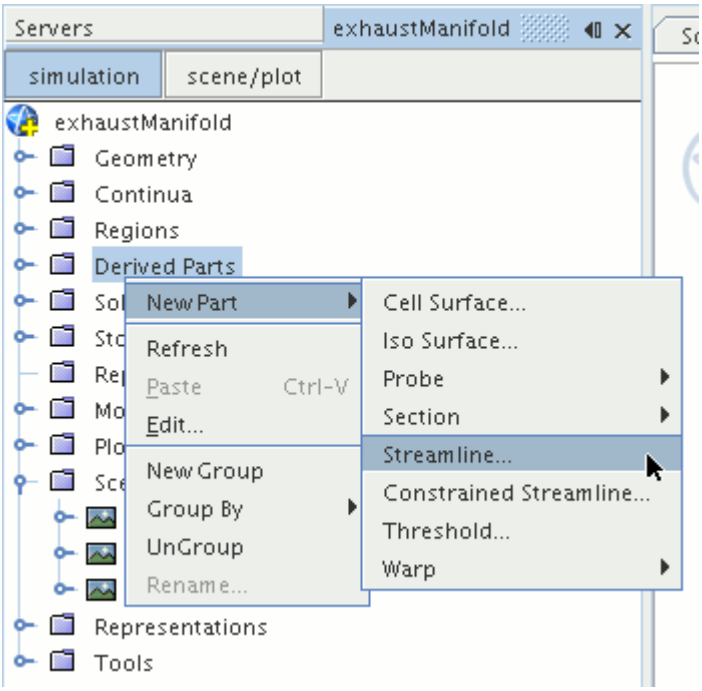
4. Set *Opacity* to `0.3`
5. Click **simulation**.

Change the view of the scene to something more suitable:

6. On the toolbar, click (**Save-Restore-Select views**) and select **Views** > **+X +Y +Z** > **Up +Y**.

Create the streamlines:

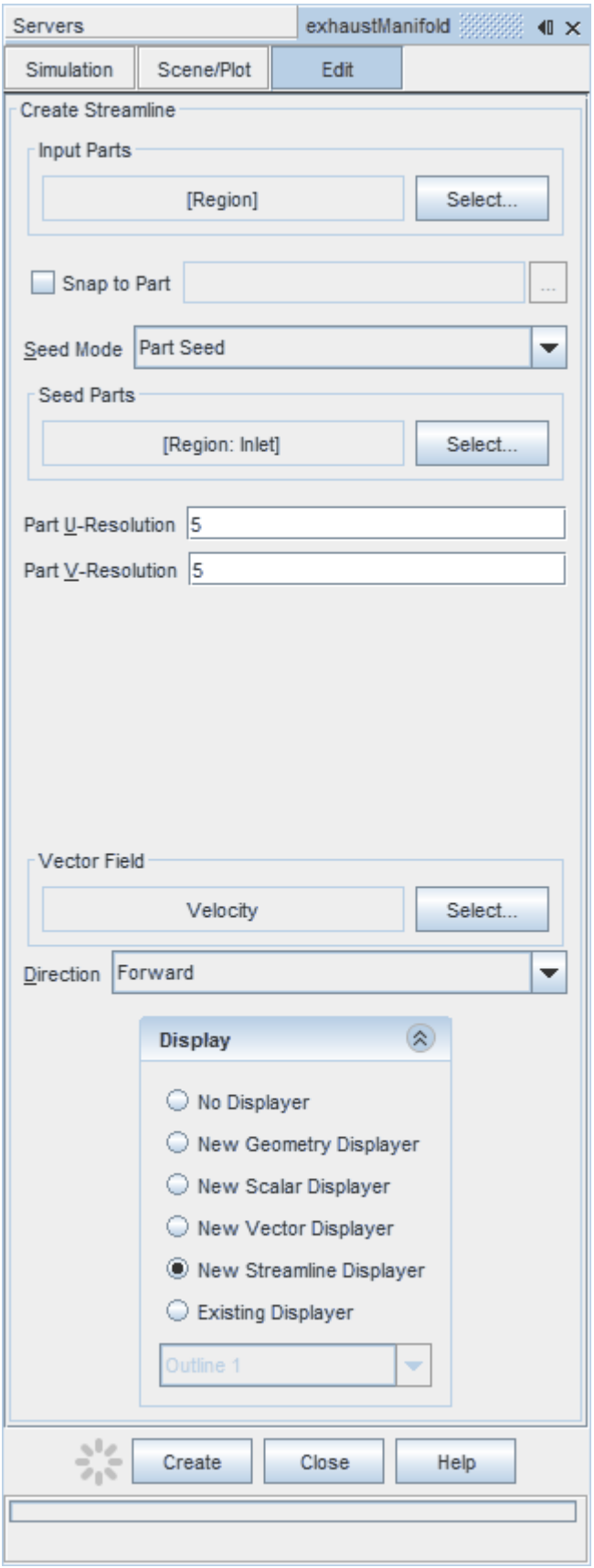
7. Right-click the **Derived Parts** node and select **New Part** > **Streamline...**.



The *Create Streamline* dialog appears.

8. Make sure that **Part Seed** is selected in the *Seed Mode* section.
9. Set *Seed Parts* to **Regions** > **Region** > **Boundaries** > **Inlet**.
10. Make sure that **New Streamline Displayer** is selected in the *Display* group box.

    The completed dialog is shown below.


Servers
exhaustManifold
Simulation
Scene/Plot
Edit
Create Streamline
Input Parts
[Region]
Select...
Snap to Part
Seed Mode
Part Seed
Seed Parts
[Region: Inlet]
Select...
Part U-Resolution
5
Part V-Resolution
5
Vector Field
Velocity
Select...
Direction
Forward
Display
No Displayer
New Geometry Displayer
New Scalar Displayer
New Vector Displayer
New Streamline Displayer
Existing Displayer
Outline 1
Create
Close
Help

11. Click **Create** to create the streamlines and add them to the geometry scene as part of a streamline displayer.
12. Click **Close** to close the *Create Streamline* dialog.

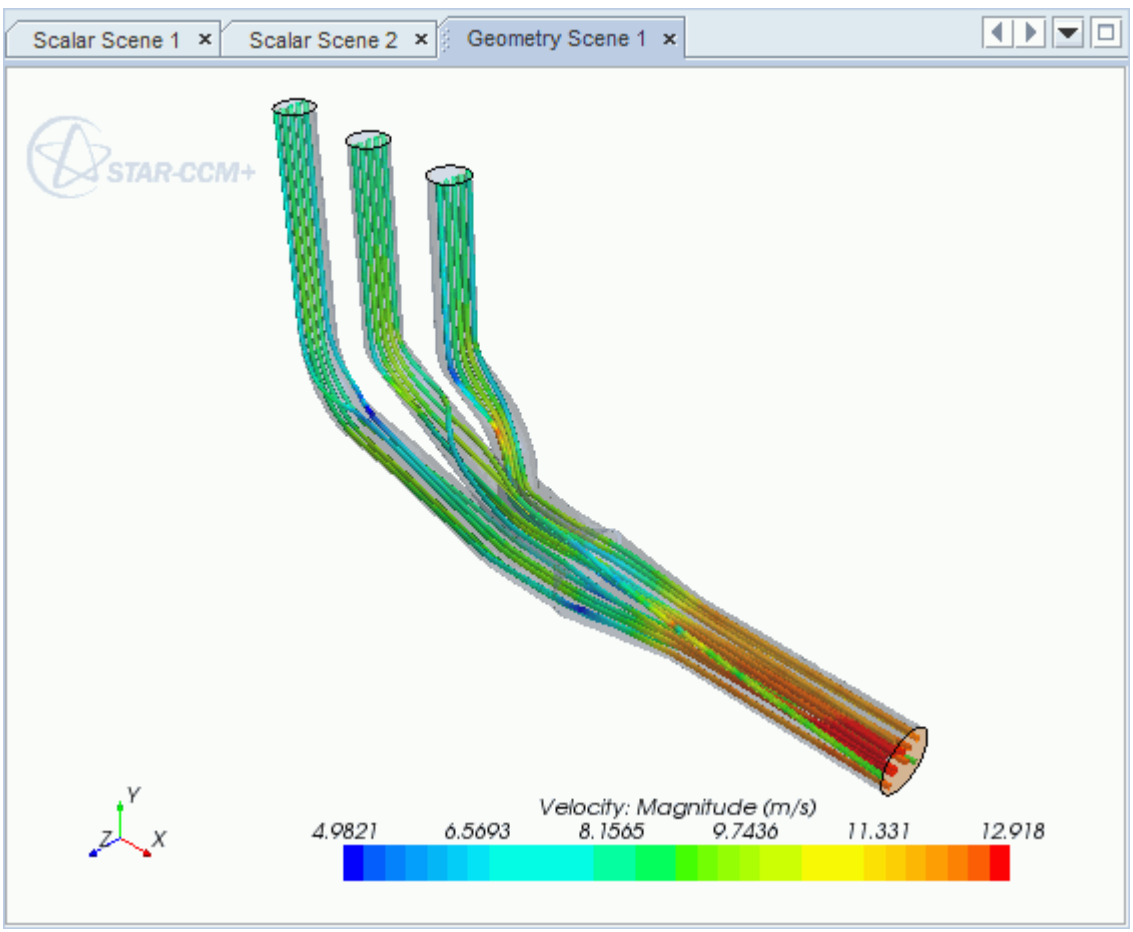
Change the display method of the streamlines to tubes instead of the default lines:

13. Click **scene/plot**.
14. Select the **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** node.
15. Set *Mode* to **Tubes** and *Width* to `0.0020`.

Change the color of the streamlines to represent velocity magnitude:

16. Select the **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** > **Scalar Field** node.
17. Set *Function* to **Velocity** > **Lab Reference Frame** > **Magnitude**.
18. Save the simulation.

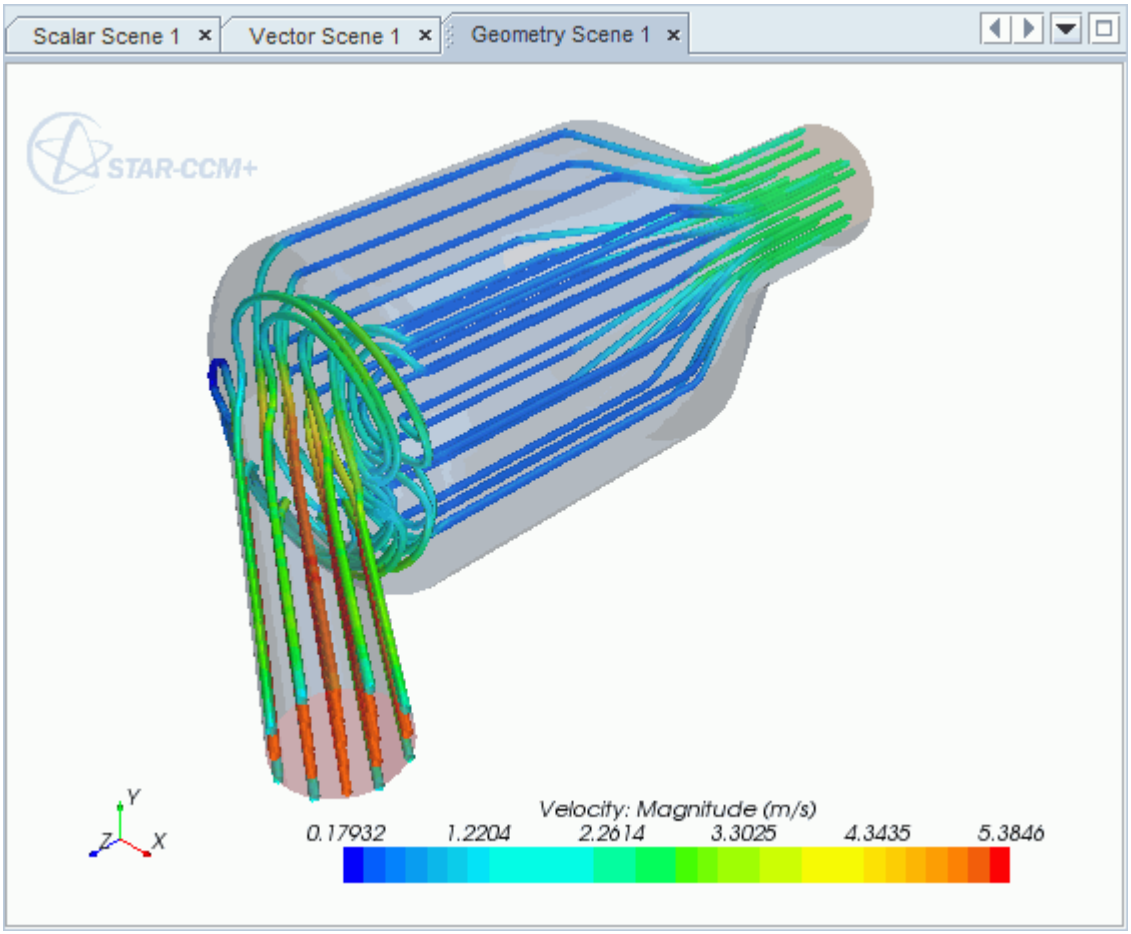
The finished scene is shown below.



Porous Media Flow

This tutorial takes you through the steps that are required to run a CFD calculation in the STAR-Inventor environment.

In this tutorial, the model is an idealized catalyst.



Contents:

Opening the CAD Model
Selecting Fluid Physics Models
Setting the Porous Region
Defining Boundary Conditions
Setting the Mesh Pipeline
Modifying Solution Controls
Reviewing the Simulation Set Up
Generating the Surface Mesh
Generating the Volume Mesh

Building and Running
Post-Processing Results

Opening the CAD Model

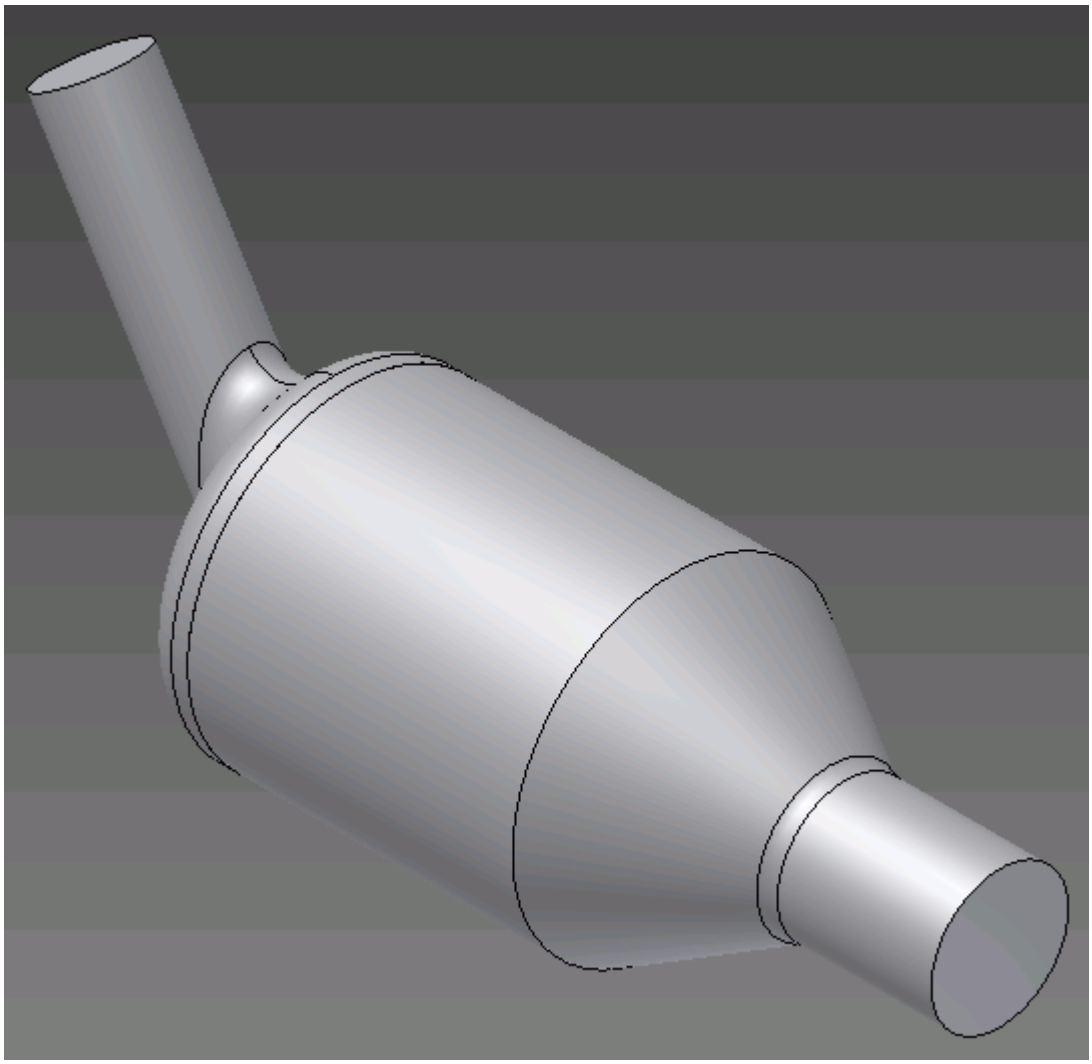
A directory containing the CAD geometry is provided with the STAR-CCM+ distribution.

This directory is located in `[STAR-CCM+_INSTALL_DIR]\doc\startutorialsdata\starinventor\data`.

1. Copy directory `data` to your own disk space and open the part file `PorousModel.iam` in STAR-Inventor.

   The assembly is made up of three parts: the inlet region, the porous middle region, and the outlet region.

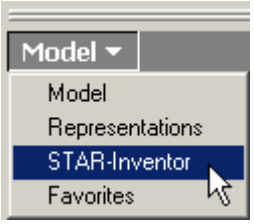
   The imported geometry is shown below.



The STAR-Inventor ribbon appears at the top of the screen.The STAR-Inventor object tree is located on the left of the screen.

To switch to this object tree:

2. Click **Model** and select **STAR-Inventor**.



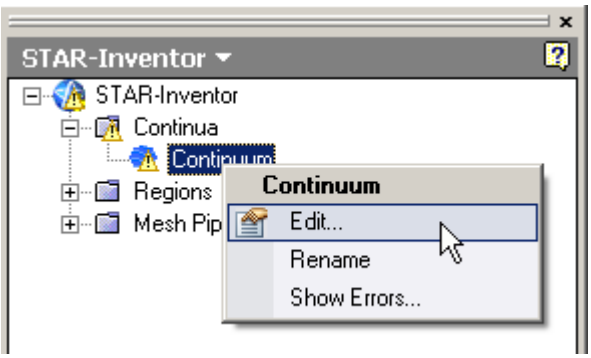
If these items are not visible, refer to Launching STAR-Inventor Manually.

Selecting Fluid Physics Models

The first step in setting up this tutorial is to define the physics of the problem. For this case, the flow is turbulent and isothermal. Temperature calculations are on by default in STAR-Inventor.

To turn temperature calculations off:

1. Right-click the **Continua** > **Continuum** node in the STAR-Inventor tree and select **Edit**.



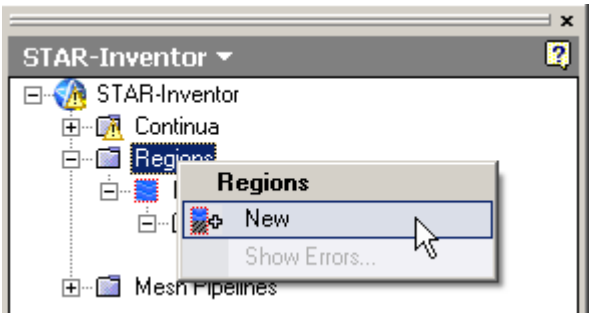
   The *Continuum Settings* dialog appears.

2. Deactivate the **Temperature** option in the *Optional Physics Models* box.
3. Click **OK**.

Setting the Porous Region

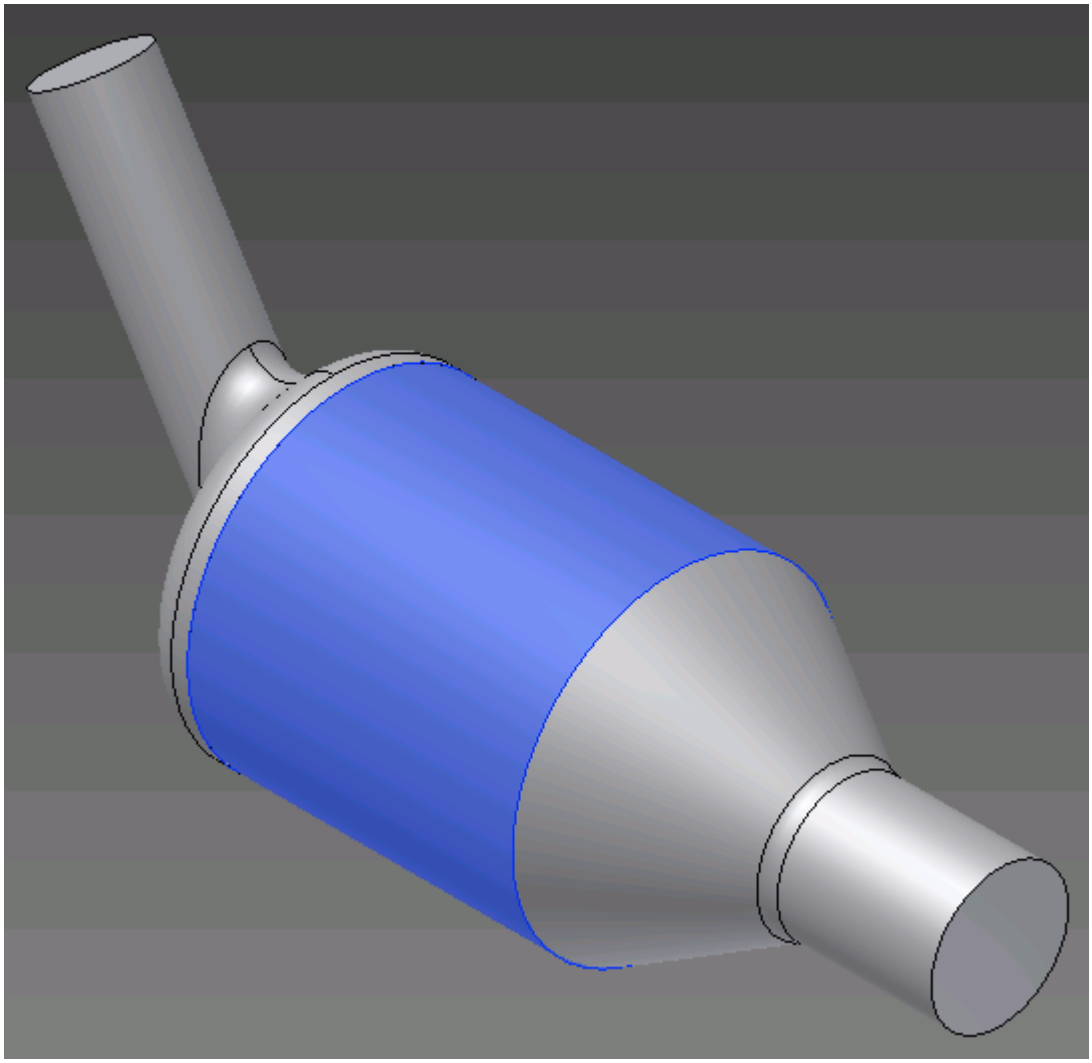
Create the porous region and define its properties:

1. Right-click the **Regions** node in the STAR-Inventor object tree and select **New**.



The *Region Settings* dialog appears.

2. Select the middle part of the assembly as shown below.



3. Set *Name* to `Porous Region`.
4. Activate the *Porous* model and select the **Uni-Directional** porous type.
5. Set the following properties for the *Porous* region:

Parameter	Value
α (*Alpha*)	`25 kg/m^4`
β (*Beta*)	`1500 kg/m^3.s`
(*Cross-Flow Resistance Factor*)	`250`

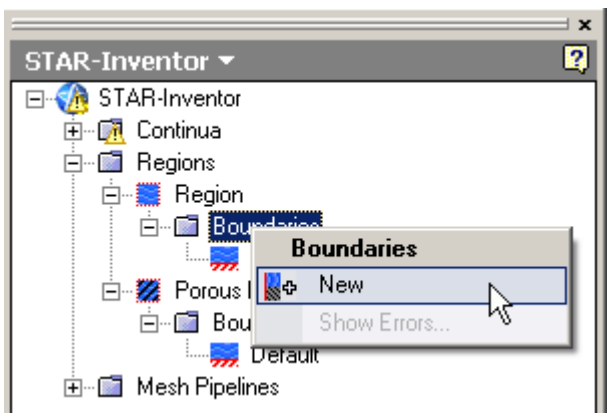
6. Click **OK**.

Defining Boundary Conditions

Create the inlet and outlet boundary conditions. STAR-Inventor automatically adds interfaces between the different regions. These interfaces are visible in the final STAR-CCM+ simulation.

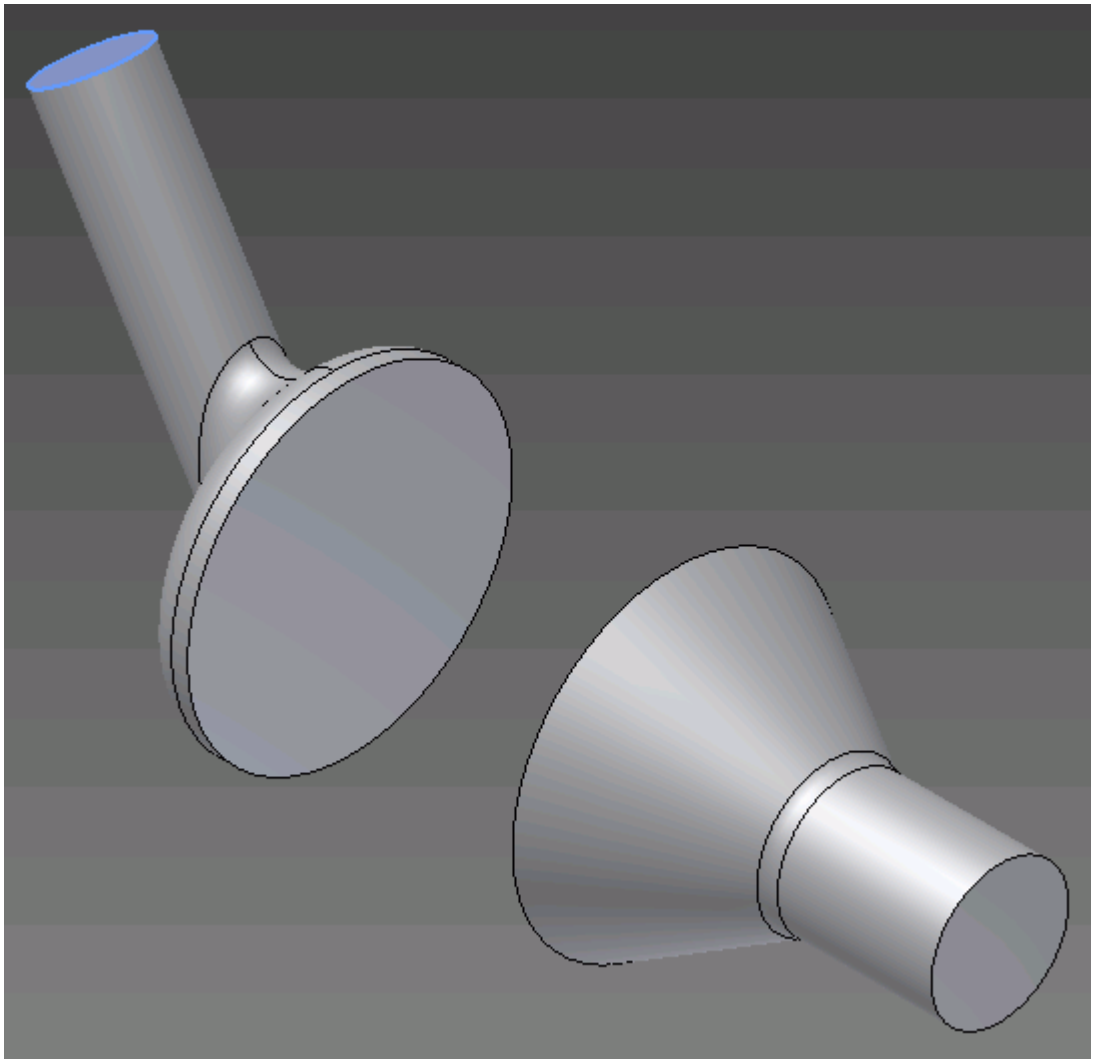
Create the inlet boundary first:

1. Right-click the **Regions** > **Region** > **Boundaries** node and select **New**.



The *Boundary Settings* dialog appears.

2. Select the face that is highlighted below.



These selections are reported in the *Face Selection* box in the *Boundary Settings* dialog.

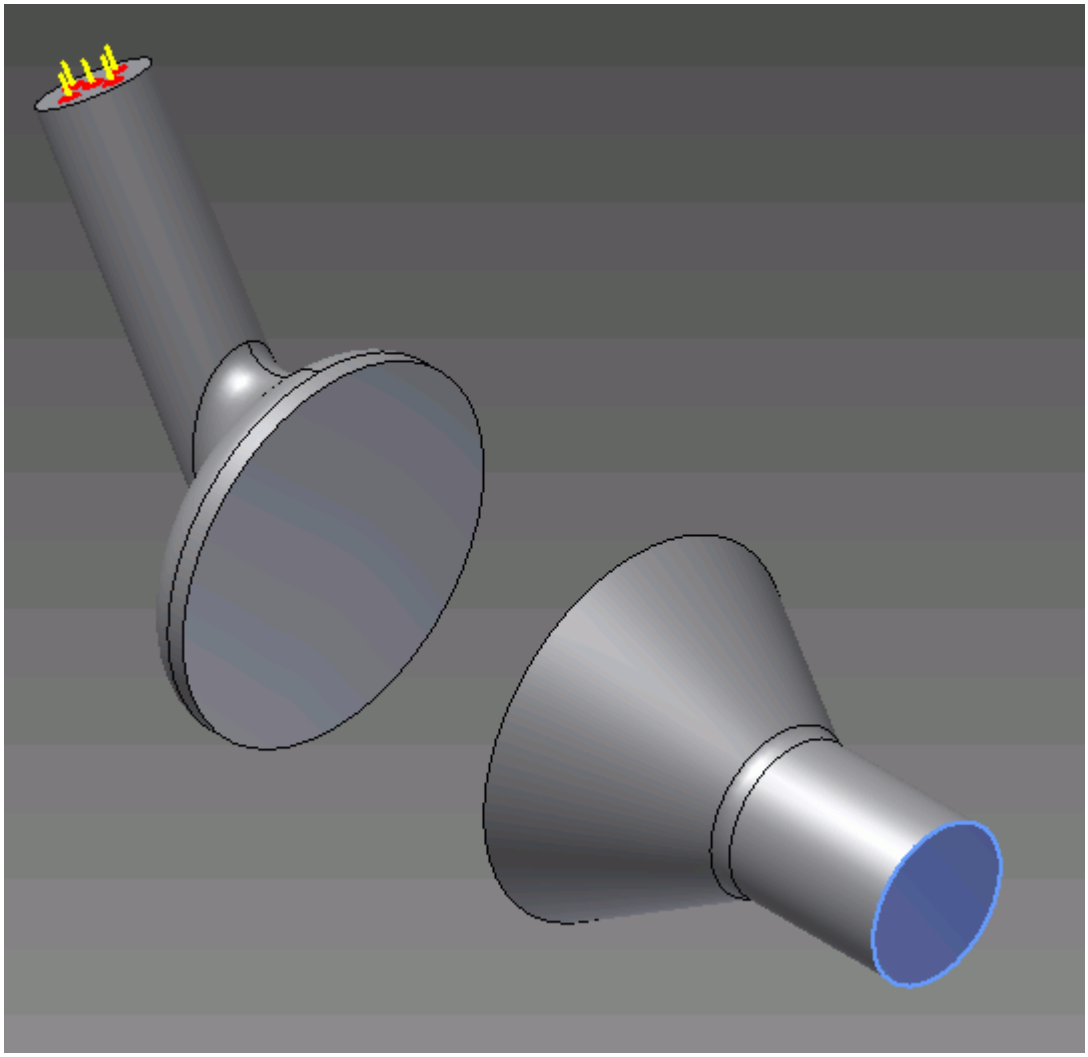
3. Set the following parameters:

Parameter	Value
Type	**Velocity Inlet**
Name	`Inlet`
(*Normal Velocity*)	`5 m/s`

4. Click **OK**.

Create the outlet boundary:

5. Using the same technique, create another boundary.
6. Select the face that is highlighted below.



7. Set the following parameters:

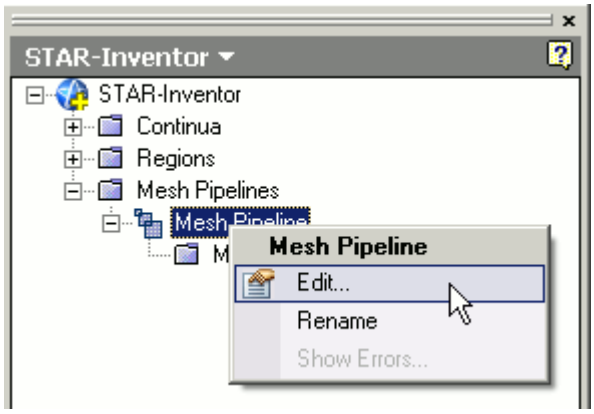
Parameter	Value
Type	**Pressure Outlet**
Name	`Outlet`

8. Click **OK**.

Setting the Mesh Pipeline

To obtain more accurate results, modify the mesh pipeline:

1. Right-click the **Mesh Pipelines** > **Mesh Pipeline** node in the STAR-Inventor tree and select **Edit**.



The *Mesh Pipeline Settings* dialog appears.

2. Set the following parameters:

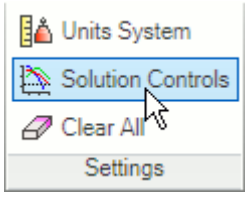
Parameter	Value
Base Size	
Relative to bodies	**Deactivated**
(*Base Size*)	`0.0025 m`
Prism Layer	
User Absolute Value	**Activated**
(*Prism Layer Thickness*)	`0.001 m`

3. Click **OK**.

Modifying Solution Controls

Modify the maximum number of iterations and the solution accuracy settings.

1. Click **Solution Controls** on the STAR-Inventor ribbon.



The *Solution Controls* dialog appears.

2. Set the following parameters:

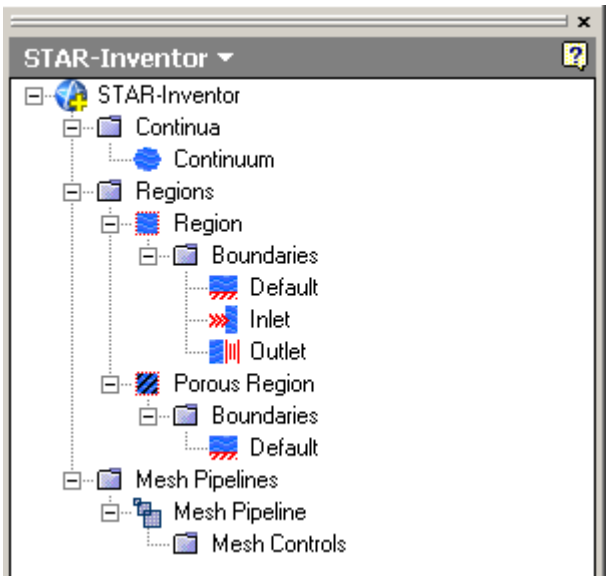
Parameter	Value
(*Number of Iterations*)	`400`

Parameter	Value
Solution Accuracy	**First Order**

3. Click **OK**.
4. Save your progress.

Reviewing the Simulation Set Up

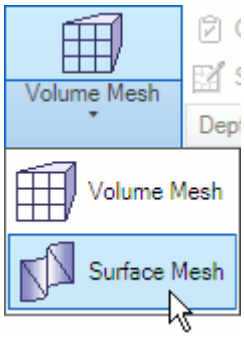
The final STAR-Inventor object tree is shown below.



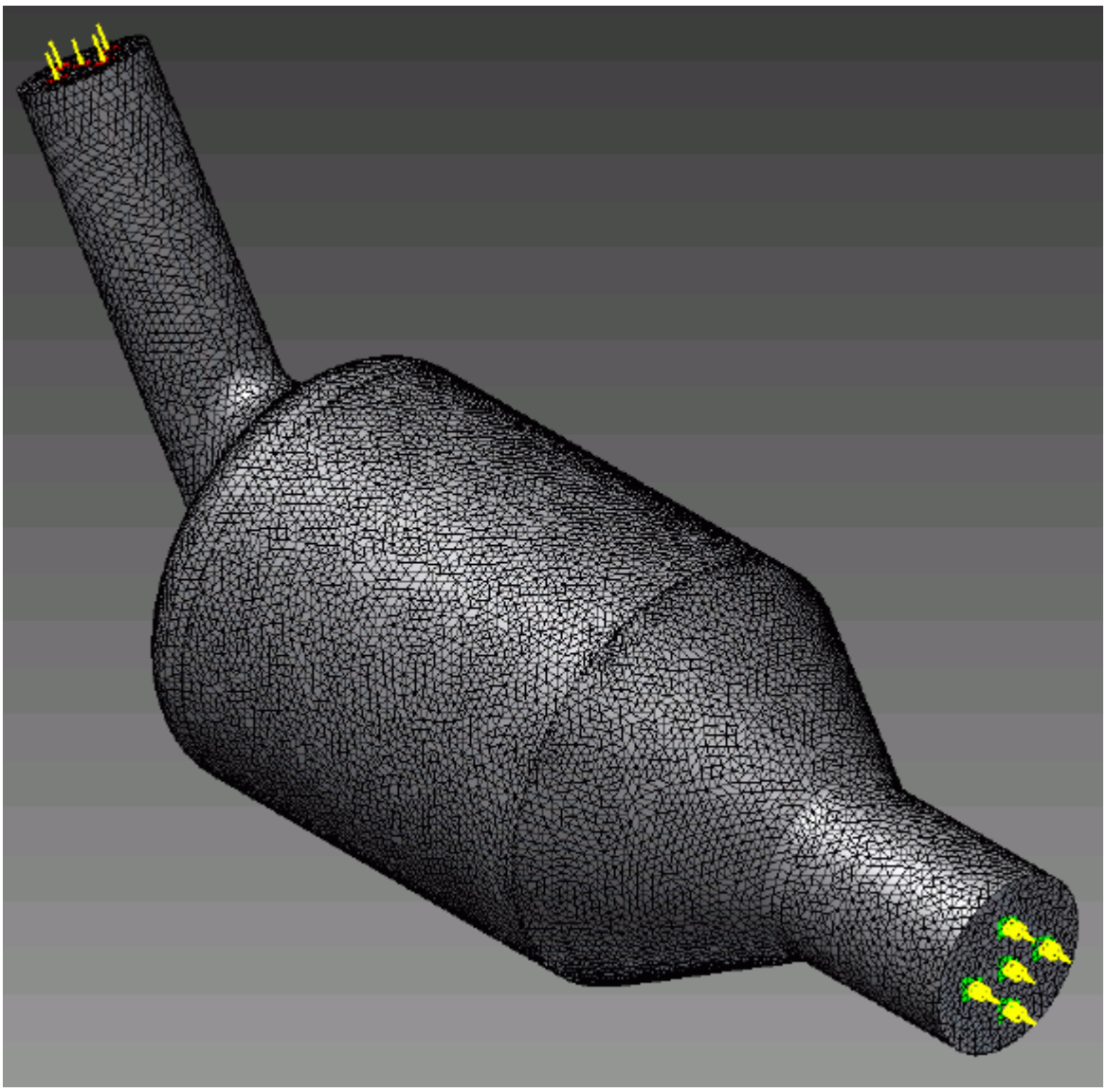
Generating the Surface Mesh

Generate the surface mesh:

1. From the STAR-Inventor ribbon, click **Surface Mesh**.From the STAR-Inventor ribbon, click ▾ next to the **Volume Mesh** icon and select **Surface Mesh**.



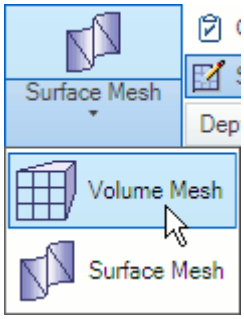
The surface mesh is shown below.



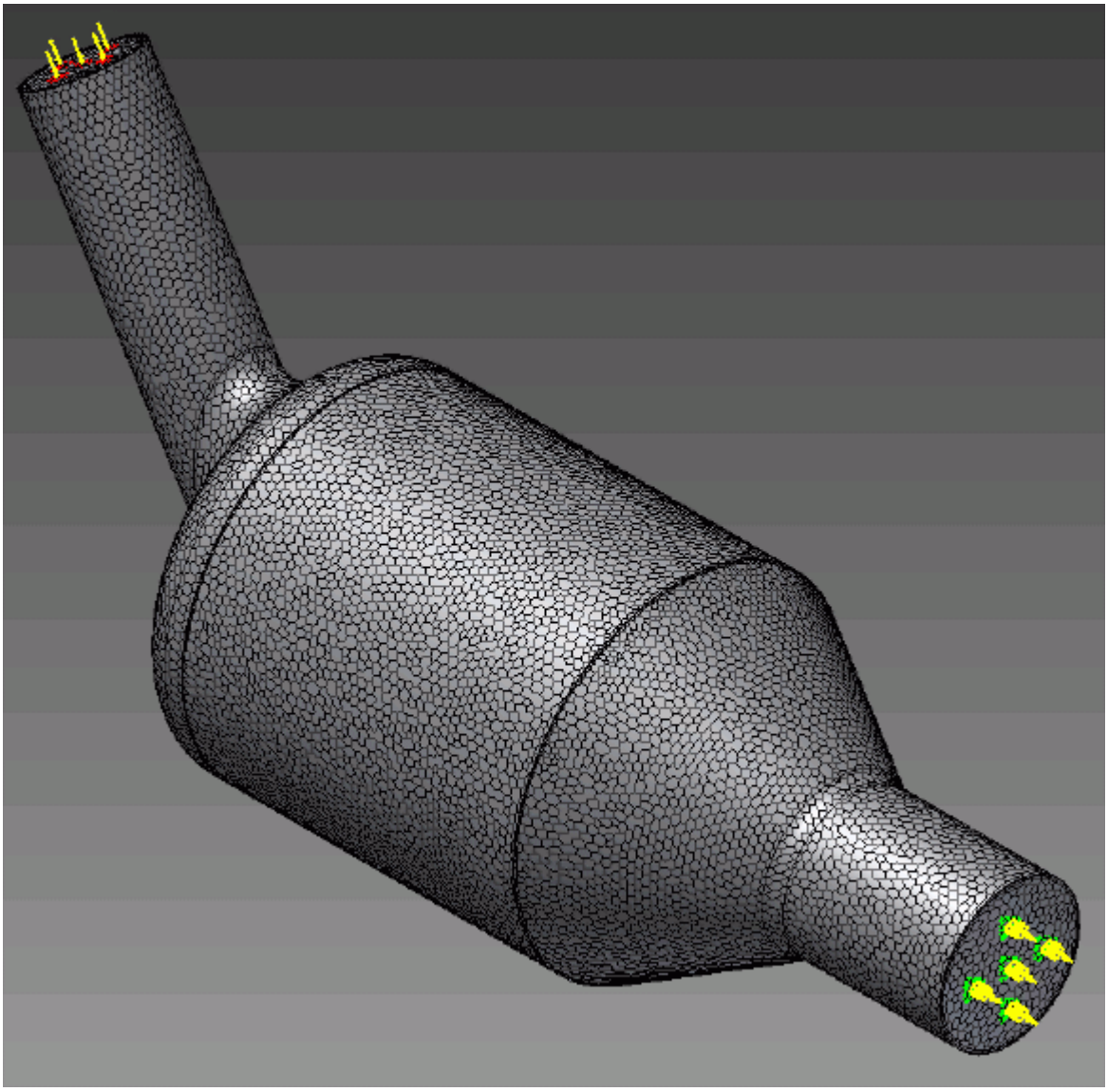
Generating the Volume Mesh

Generate the volume mesh:

1. From the STAR-Inventor ribbon, click **Volume Mesh**.From the STAR-Inventor ribbon, click ▾ next to the **Surface Mesh** icon and select **Volume Mesh**.



The volume mesh is shown below.

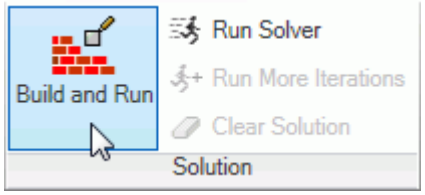


Building and Running

The case is now ready to run.

To run the simulation:

1. From the ribbon, click **Build and Run**.



The *Residuals* plot appears in the STAR-Inventor *Output* window. You can use this plot to monitor the convergence of the solution.

When the solver is done, STAR-CCM+ launches.

Post-Processing Results

You perform all post-processing in STAR-CCM+. By default, a scalar scene, *Scalar Scene 1*, automatically appears in the *Graphics* window when the solver finishes. This scene displays the absolute total pressure.

The results that are contained in the following sections are representative. You may see some variation in the results that you obtain, depending on what platform and package you are using.

1. In STAR-CCM+, click **File** > **Save** to save the simulation.

Contents:

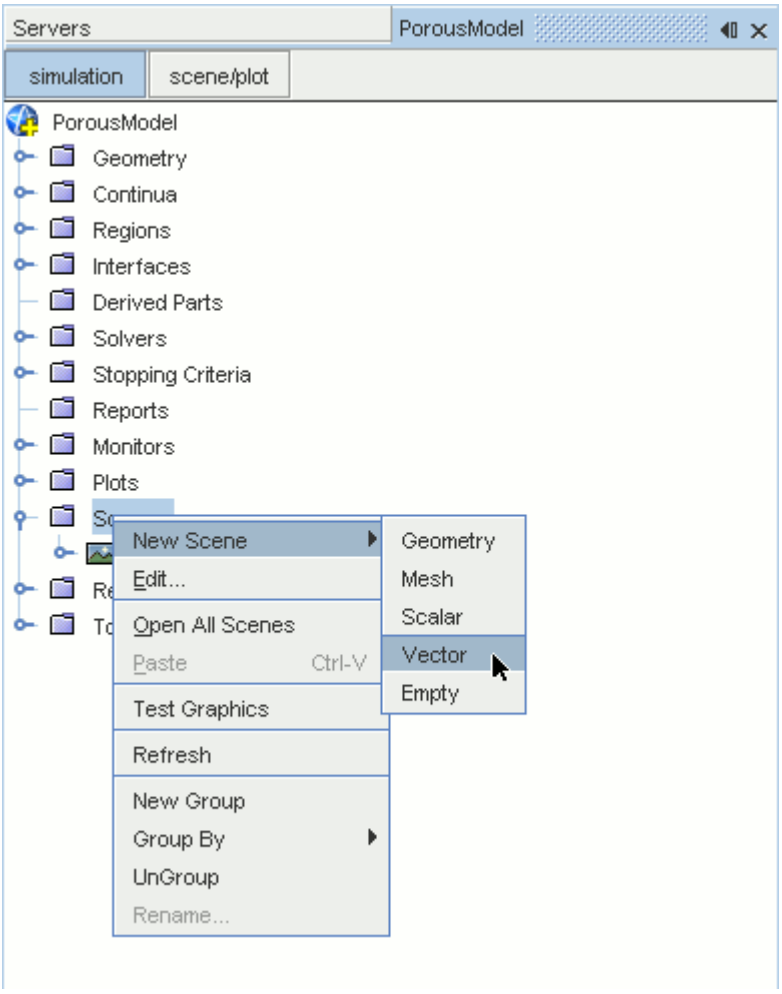
Creating a Scalar Scene
Creating A Streamlines Scene
Creating a Vector Scene
Creating a Streamlines Scene
Creating A Scalar Scene
Visualizing the Wing Skin Temperature
Creating a Streamlines Scene
Creating A Scalar Scene
Creating a Streamlines Scene
Creating a Scalar Scene
Modifying The Scalar Scene
Creating a Streamlines Scene

Creating a Vector Scene

Create a vector scene with a section plane bisecting the geometry. Velocity vectors are plotted on this plane.

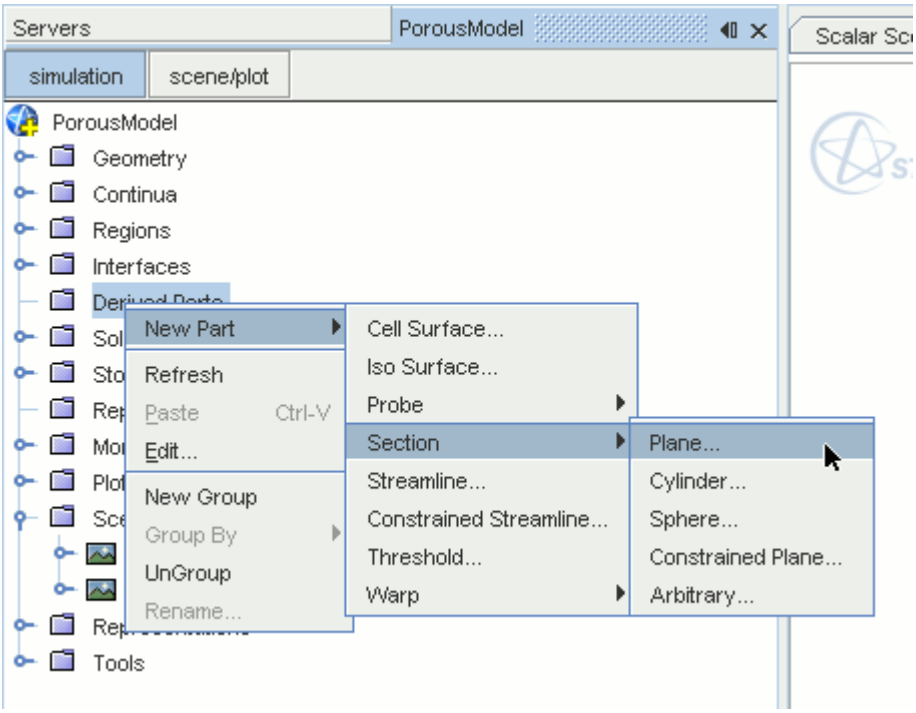
To create the vector scene:

1. Right-click the **Scenes** node in the explorer pane and select **New Scene** > **Vector**.



A new vector scene, *Vector Scene 1*, is created and selected in the *Graphics* window.

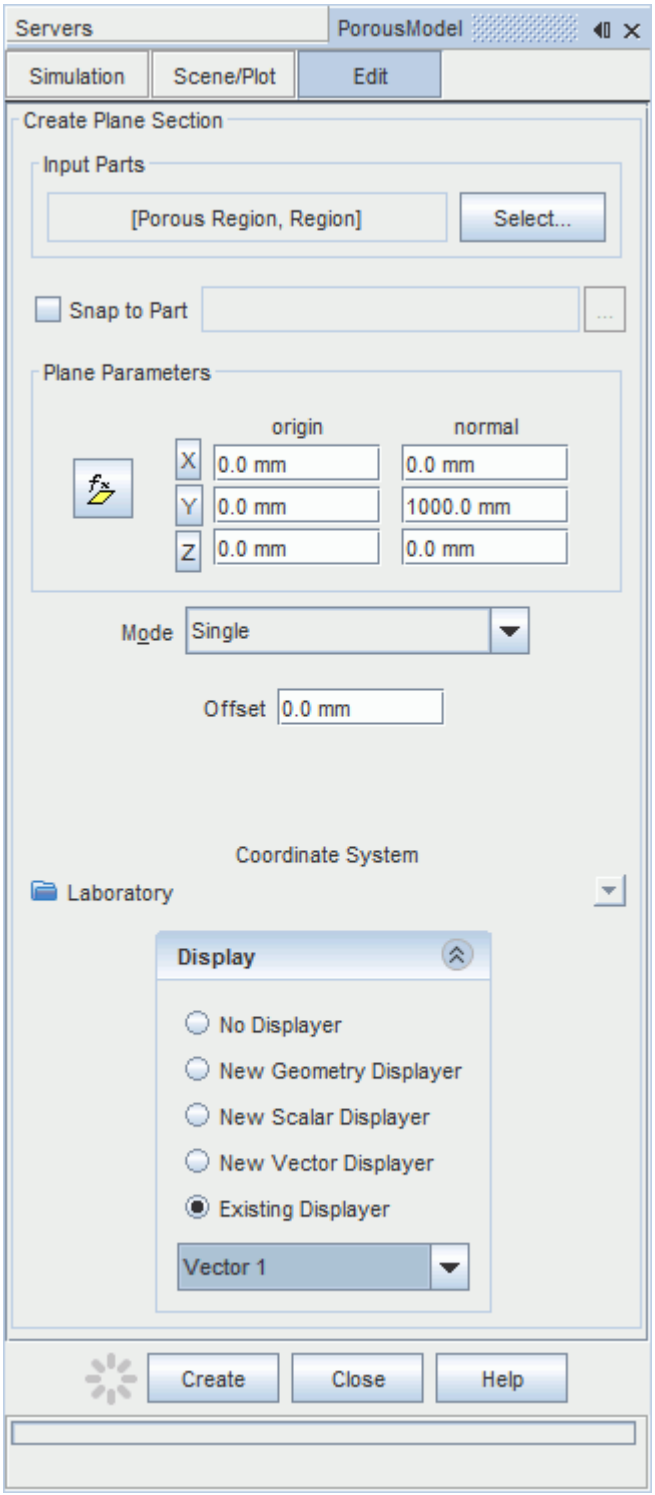
2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane...**.



The *Create Plane Section* dialog appears.

3. In the *Plane Parameters* box, set the *origin* vector to `[0 mm, 0 mm, 0 mm]`.
4. Set the *normal* vector to `[0 mm, 1000 mm, 0 mm]`.
5. In the *Display* group box, select **Existing Displayer** and in the drop-down menu select **Vector 1**.

The completed dialog is shown below.



6. Click **Create** followed by **Close**.

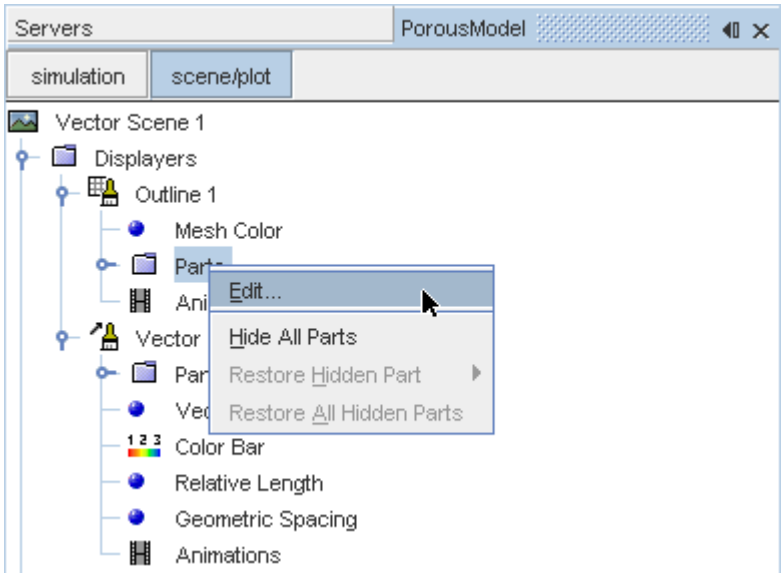
   Velocity is automatically set as the active function when creating a vector displayer.

Change the view of the scene to something more suitable:

7. On the toolbar, click (**Save-Restore-Select views**) and select **Views** > **+X +Y +Z** > **Up +Y**.

In order to remove the unwanted outlines from the scene do the following:

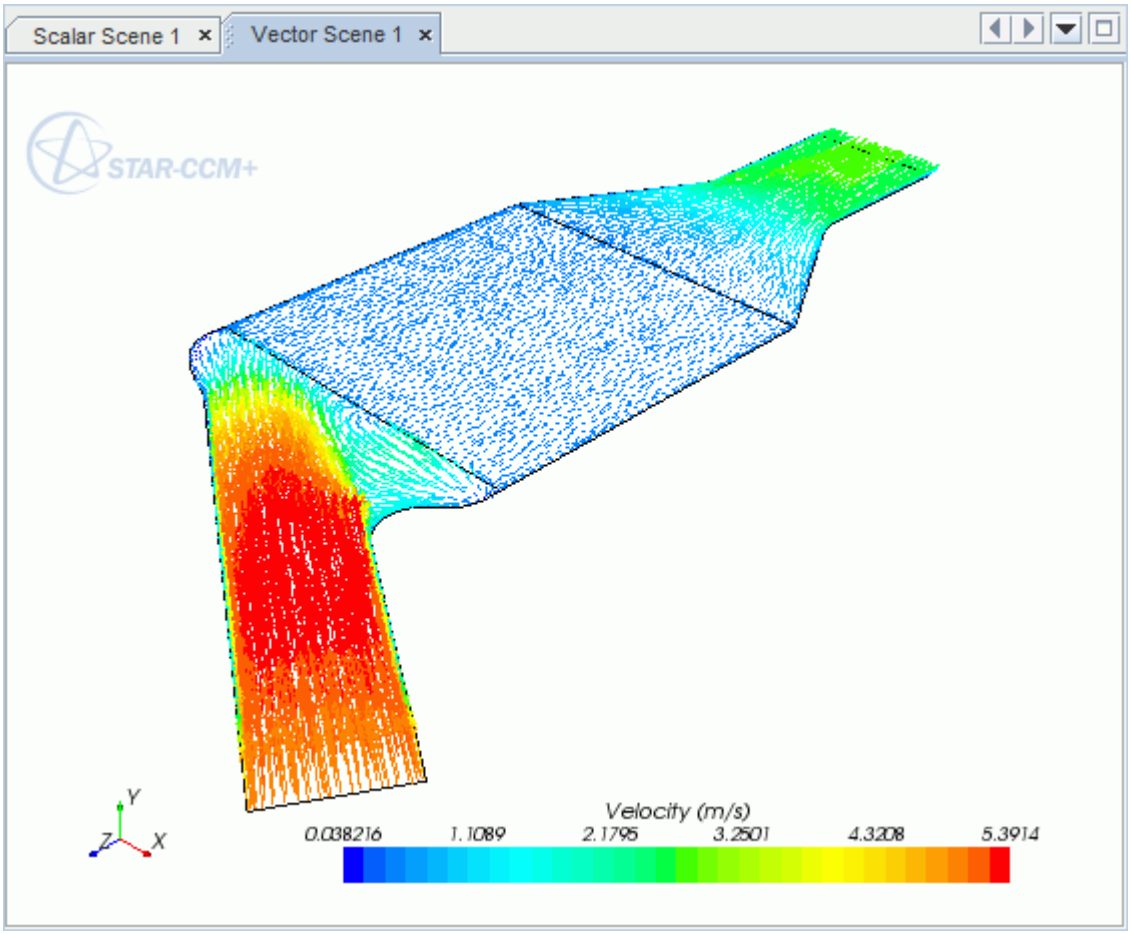
8. Click the **scene/plot** button in the explorer pane.
9. Navigate to **Vector Scene 1** > **Displayers** > **Outline 1** > **Parts**.
10. Right-click the **Parts** node and select **Edit...**.



The *Parts* dialog appears.

11. Right-click on a blank area and choose **Deselect All**.
12. Select the **Derived Parts** > **Plane Section** node.
13. Click **OK**.
14. Click the **simulation** button to return to the simulation object tree.
15. Click (**Save**) to save the simulation .

The finished scene is shown below.



Creating a Streamlines Scene

Create streamlines to get a better understanding of the flow inside the catalyst.

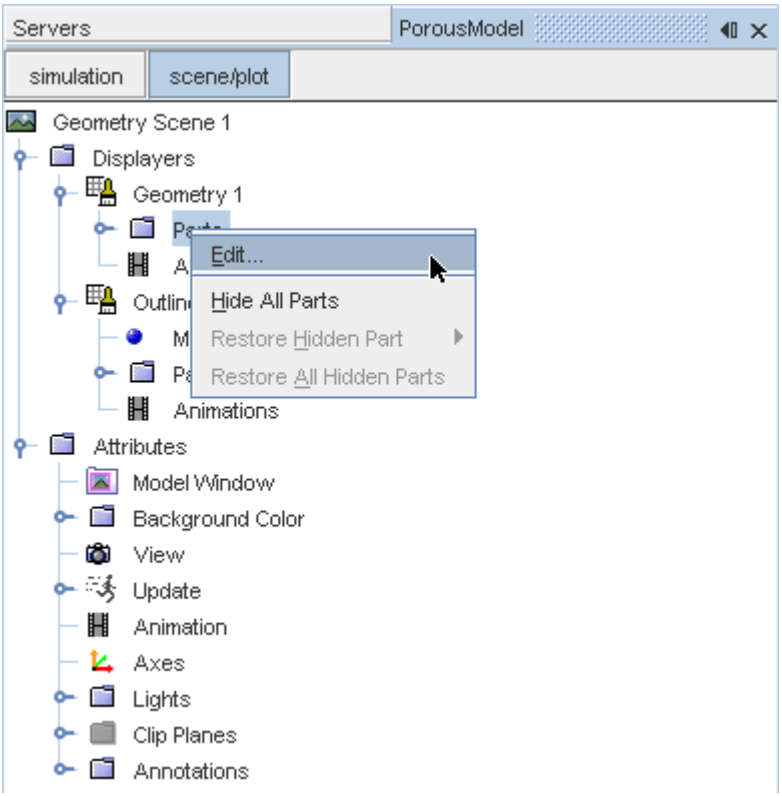
Begin by creating a geometry scene:

1. Right-click the **Scenes** node in the object tree and select **New Scene** > **Geometry**.

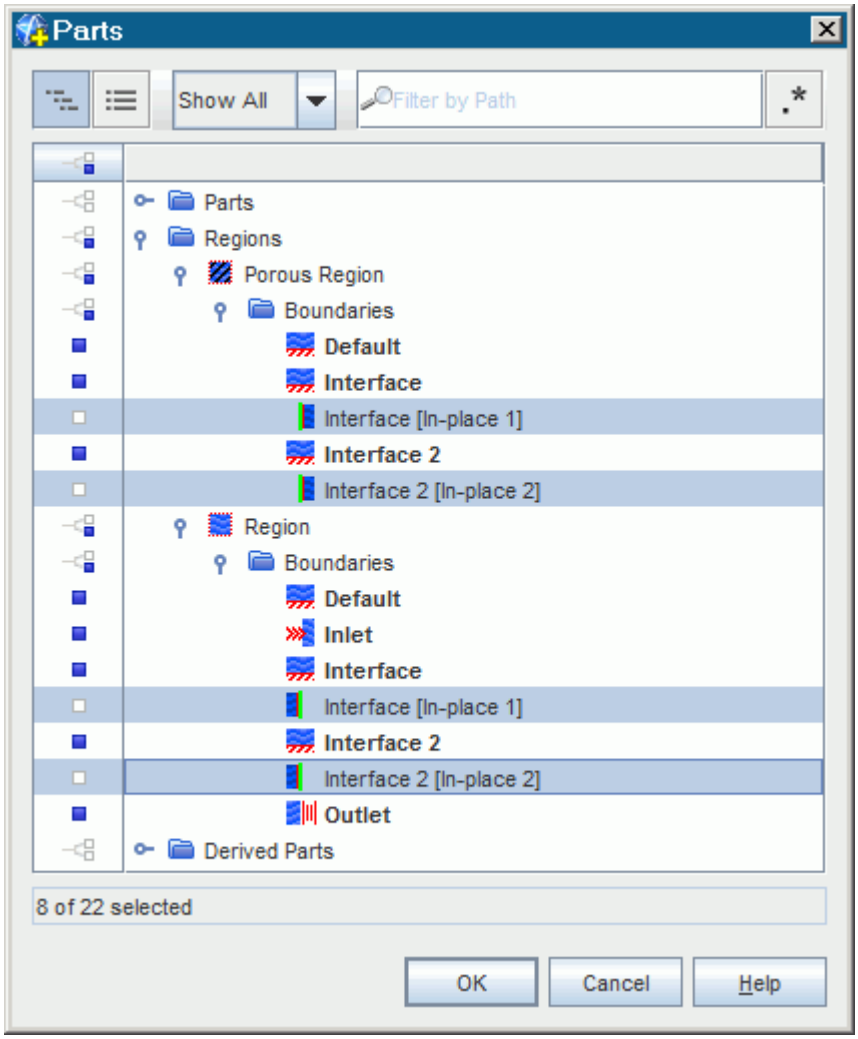
   A new geometry scene is added to the *Graphics* window.
2. Click **scene/plot**.
3. Navigate to **Geometry Scene 1** > **Displayers** > **Geometry 1**.
4. Set *Opacity* to `0.3`.

To remove the yellow interface boundaries from the scene:

5. Right-click **Geometry Scene 1** > **Displayers** > **Geometry 1** > **Parts** and select **Edit...**.



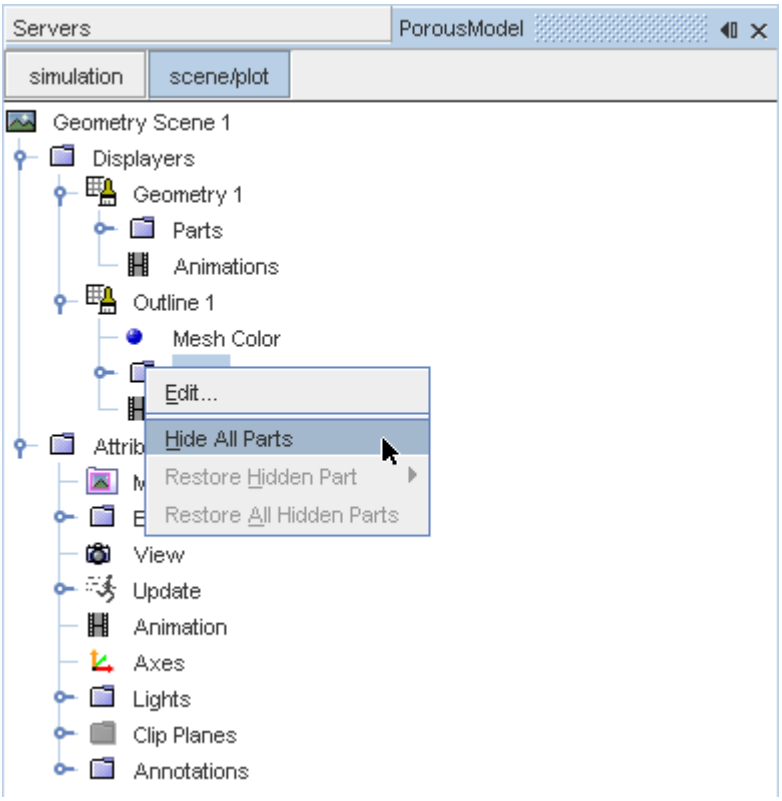
6. In the *Parts* dialog that appears, expand the **Regions** > **Porous Region** > **Boundaries** and **Regions** > **Region** > **Boundaries** nodes.
7. Deactivate the following items in each region:
   - **Interface [In-Place 1]**
   - **Interface 2 [In-Place 2]**



8. Click **OK**.

Hide the outlines on the geometry model:

9. Right-click the **Geometry Scene 1** > **Displayers** > **Outline 1** > **Parts** node in the object tree and select **Hide All Parts**.



10. Click **simulation**.

Change the view of the scene to something more suitable:

11. On the toolbar, click (**Save-Restore-Select views**) and select **Views** > **+X +Y +Z** > **Up +Y**.

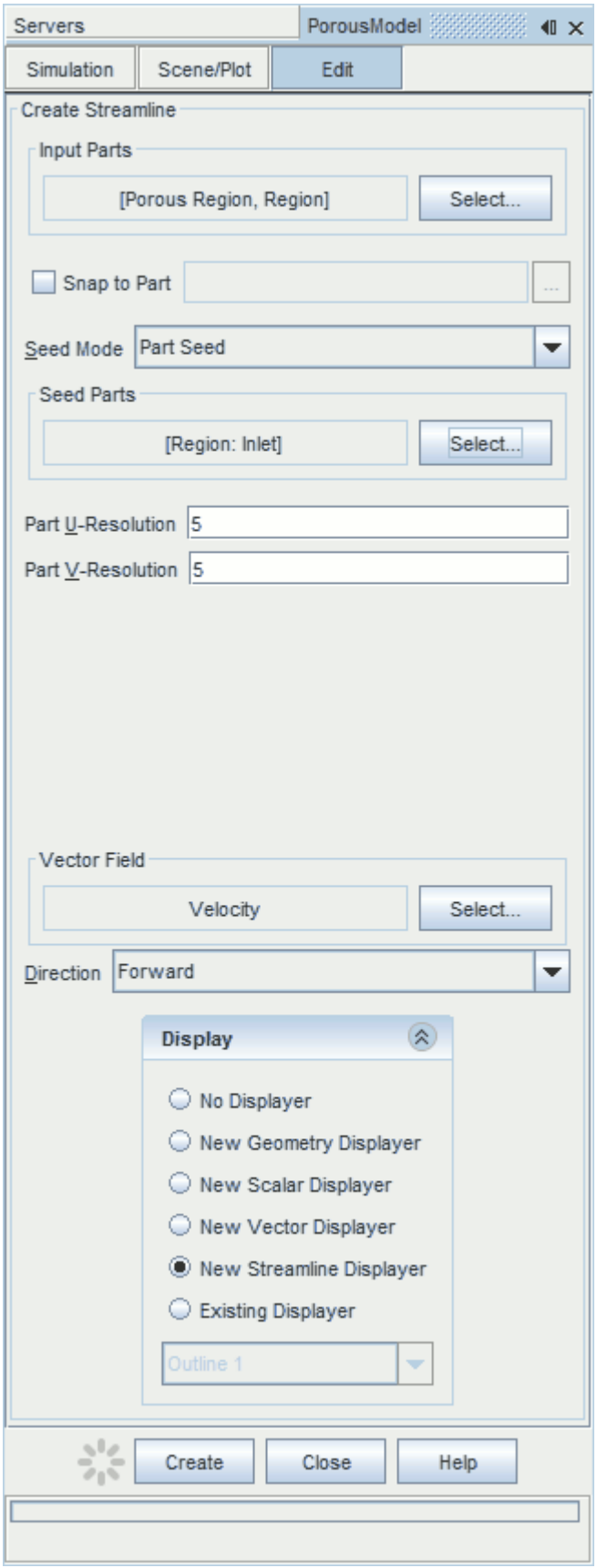
Create the streamlines:

12. Right-click on the **Derived Parts** node and select **New Part** > **Streamline...**.

The *Create Streamline* dialog appears.

13. Make sure that **Part Seed** is selected in the *Seed Mode* section.
14. Set *Seed Parts* to **Regions** > **Region** > **Boundaries** > **Inlet**.
15. Click **OK**.
16. Make sure that **New Streamline Displayer** is selected in the *Display* group box.

The completed dialog is shown below.


Servers
PorousModel
Simulation
Scene/Plot
Edit
Create Streamline
Input Parts
[Porous Region, Region]
Select...
Snap to Part
Seed Mode
Part Seed
Seed Parts
[Region: Inlet]
Select...
Part U-Resolution
5
Part V-Resolution
5
Vector Field
Velocity
Select...
Direction
Forward
Display
No Displayer
New Geometry Displayer
New Scalar Displayer
New Vector Displayer
New Streamline Displayer
Existing Displayer
Outline 1
Create
Close
Help

17. Click **Create** then **Close**.

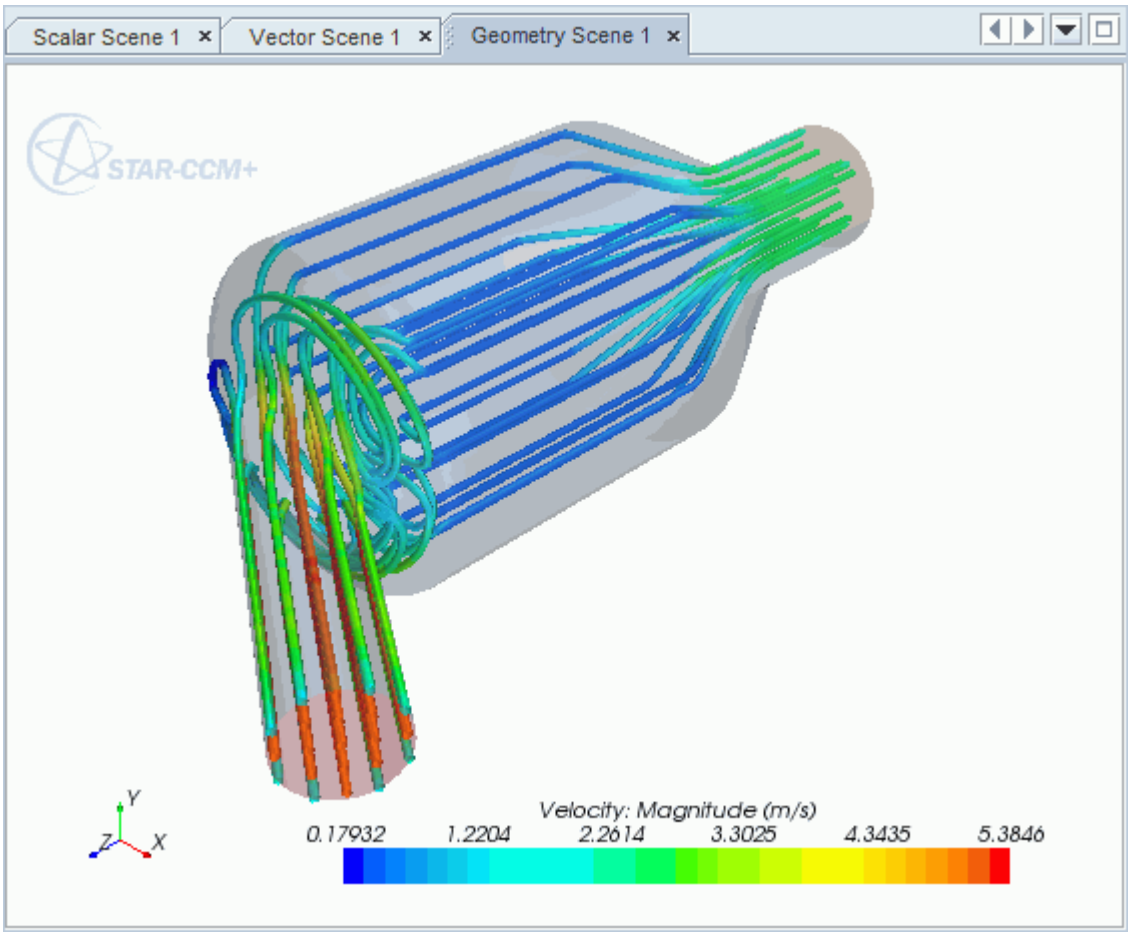
Change the display mode of the streamlines to use tubes instead of the default lines:

18. Click **scene/plot**.
19. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1**.
20. Set *Width* to `0.0012` and *Mode* to **Tubes**.

Change the color of the streamlines to represent velocity magnitude:

21. Navigate to the **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** > **Scalar Field**.
22. Set *Function* to **Velocity** > **Lab Reference Frame** > **Magnitude**.
23. Save the simulation.

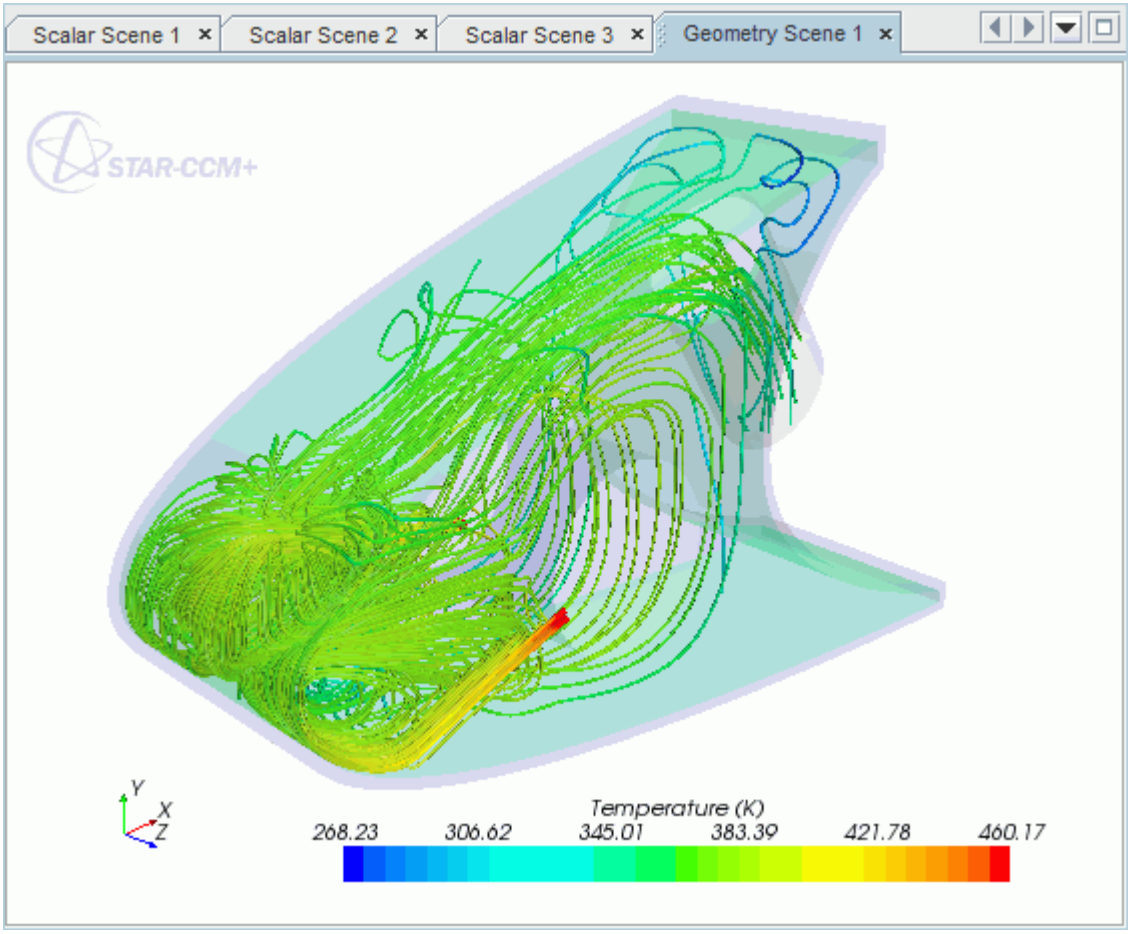
The finished scene is shown below.



Aircraft Wing Deicing System

This tutorial takes you through the steps that are required to run a CFD calculation in the STAR-Inventor environment.

For this tutorial, the model is a modern deicing system that is used on aircraft wings. The hot air from the aircraft's jet engine is used to heat the leading edge of the wing.



Contents:

Opening the CAD Model
Selecting Fluid Physics Models
Adding Solid Physics Models
Setting the Solid Region
Defining Boundary Conditions for the Fluid Region
Defining Boundary Conditions for the Solid Region
Setting the Mesh Pipeline
Setting the Coincident Faces Tolerance

Creating the Volumetric Mesh Controls
Creating a Part Mesh Control
Creating a Face Mesh Control
Modifying Solution Controls
Reviewing the Simulation Set Up
Generating the Surface Mesh
Generating the Volume Mesh
Building and Running
Post-Processing Results

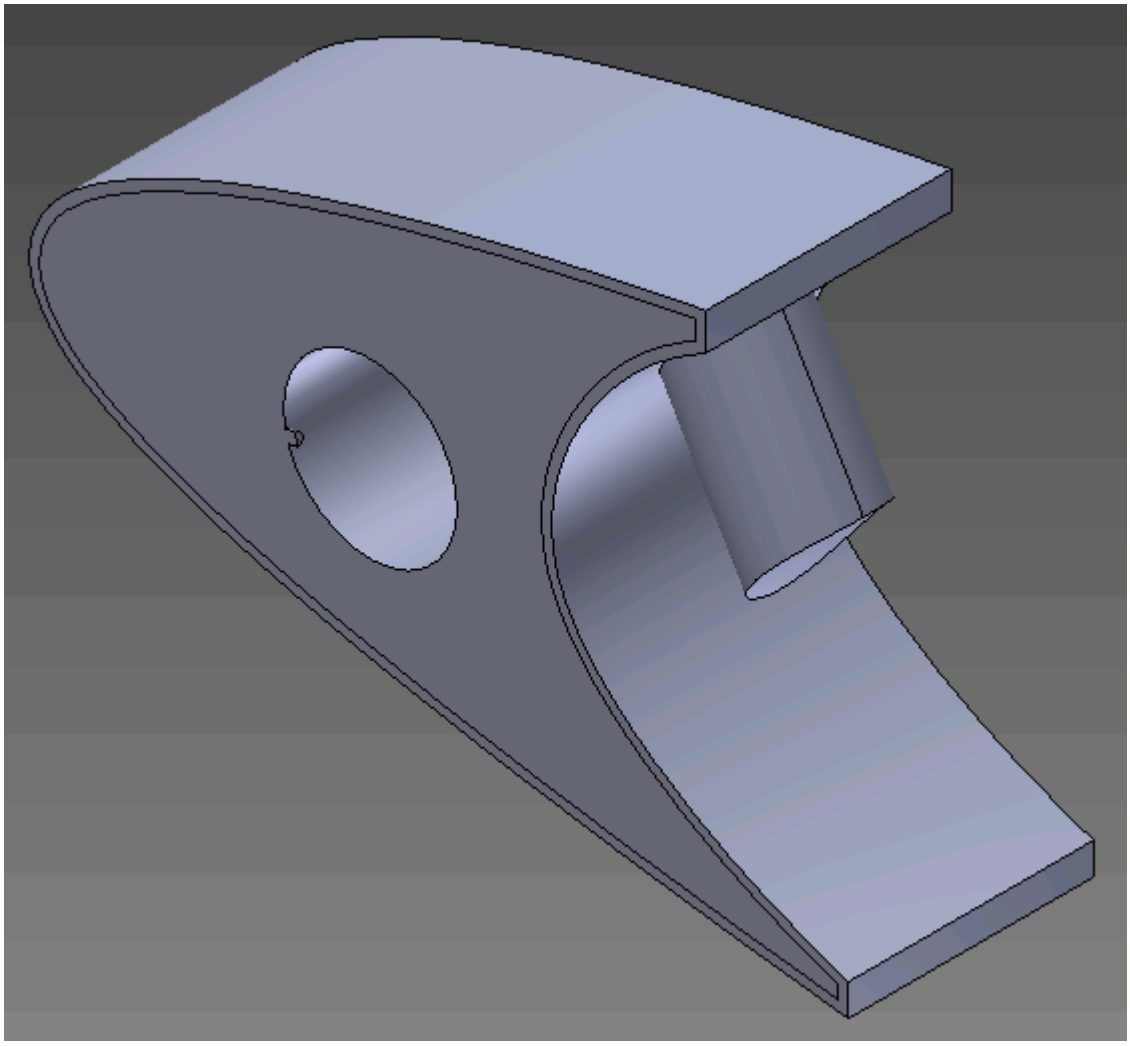
Opening the CAD Model

This directory is located in `[STAR-CCM+_INSTALL_DIR]\doc\startutorialsdata\starinventor\data`.

1. Copy directory `data` to your own disk space and open the part file `piccoloAssem.iam` in STAR-Inventor.

   The assembly is made up of two parts: the solid leading edge wing skin, and the internal air region.

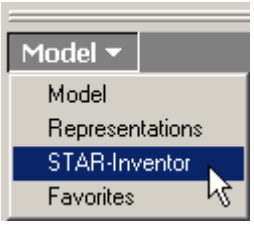
   The imported geometry is shown below.



The STAR-Inventor ribbon appears at the top of the screen.The STAR-Inventor object tree is located on the left of the screen.

To switch to this object tree:

2. Click **Model** and select **STAR-Inventor**.



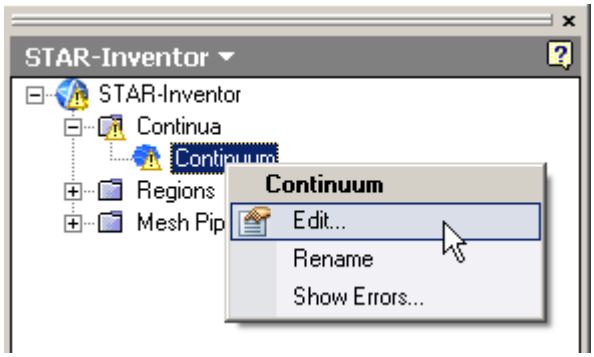
If these items are not visible, refer to Launching STAR-Inventor Manually.

Selecting Fluid Physics Models

The first step in setting up this tutorial is to modify the physics continuum that is used to model air. For this case, the flow is turbulent and thermal. Temperature calculations are on by default in STAR-Inventor.

Modify the initial conditions:

1. Right-click the **Continua** > **Continuum** node in the STAR-Inventor tree and select **Edit**.



   The *Continuum Settings* dialog appears.
2. Set *Name* to `Air`.
3. Set the *Equation of State* to **Ideal Gas**.
4. Click the *Initialization* tab.
5. Set (*Temperature*) to `400` K.
6. Click **OK** to close the dialog.

Adding Solid Physics Models

The next step in setting up this tutorial is to define the physics continuum of the solid to be used.

1. Right-click the **Continua** node in the STAR-Inventor object tree and select **New**.

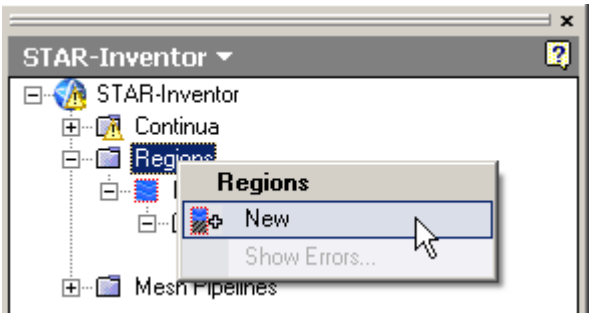
   The *Continuum Settings* dialog appears.
2. Set *Name* to `Al`.
3. Set *Material* to **Solid** and select **Aluminum Alloy** from the list.
4. Click the *Initialization* tab.

5. Set (*Temperature*) to `400 K`.
6. Click **OK**.

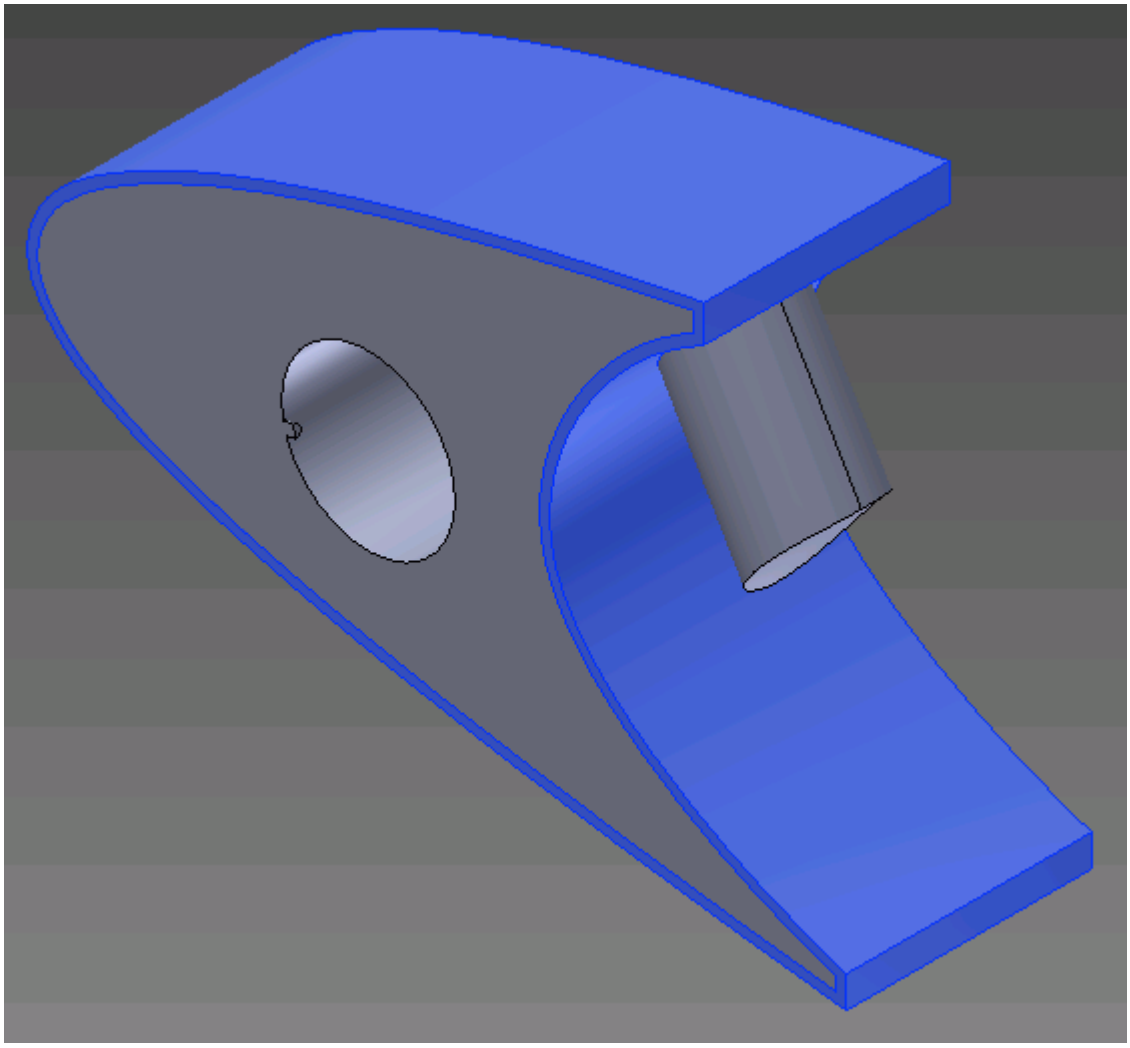
Setting the Solid Region

Create a new solid region for the wing skin:

1. Right-click the **Regions** node in the STAR-Inventor object tree and select **New**.

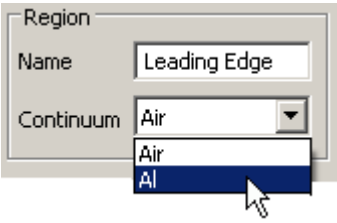


   The *Region Settings* dialog appears.
2. Select the wing skin as shown below.



Once selected, the part becomes highlighted.

3. Set *Name* to `Leading Edge`.
4. Set *Continuum* to **Al**.

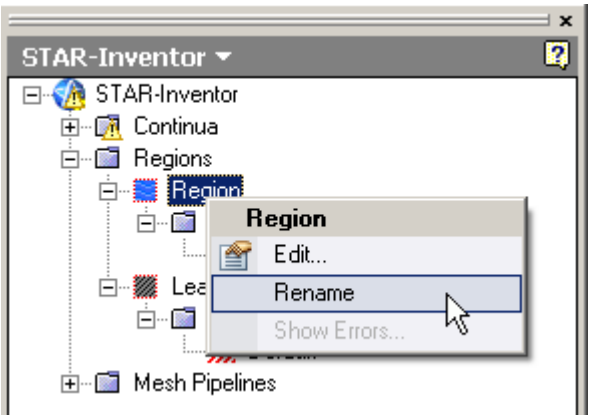


This action assigns the aluminum alloy that you created earlier as the material for this region.

5. Click **OK**.

Now that the solid region of the wing is defined, rename the default region to something more appropriate:

6. Right-click the **Regions** > **Region** node in the STAR-Inventor tree and select **Rename**.



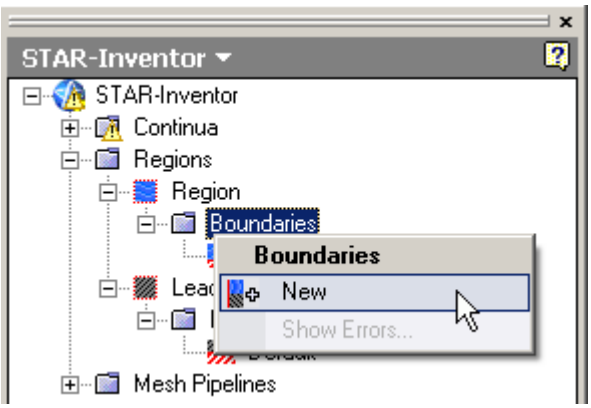
7. Enter `Internal Air` as the new name.
8. Click a blank area to accept the name change.

Defining Boundary Conditions for the Fluid Region

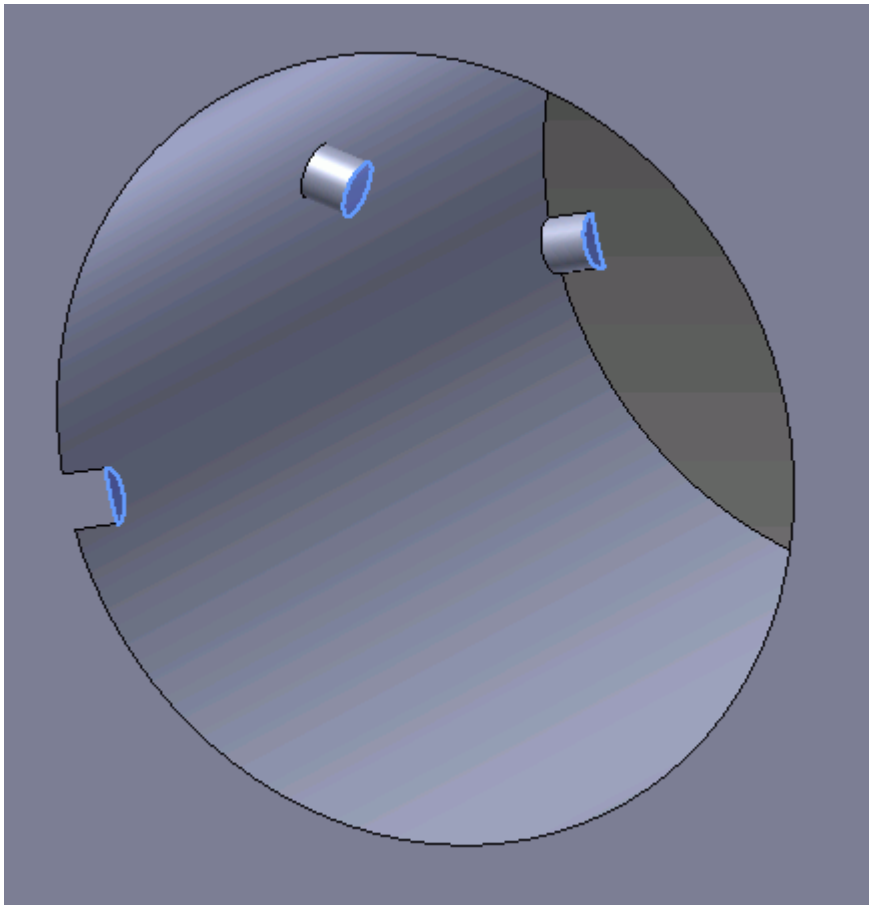
Create the inlet and outlet boundary conditions. Use symmetry boundaries on the side walls to extend the simulation domain. Use a wall boundary with a fixed temperature to model the piccolo walls.

Create the inlet boundary first:

1. Right-click the **Regions** > **Internal Air** > **Boundaries** node and select **New**.



   The *Boundary Settings* dialog appears.
2. On the CAD model, select the three faces that are highlighted below.



These selections are reported in the *Face Selection* box in the *Boundary Settings* dialog.

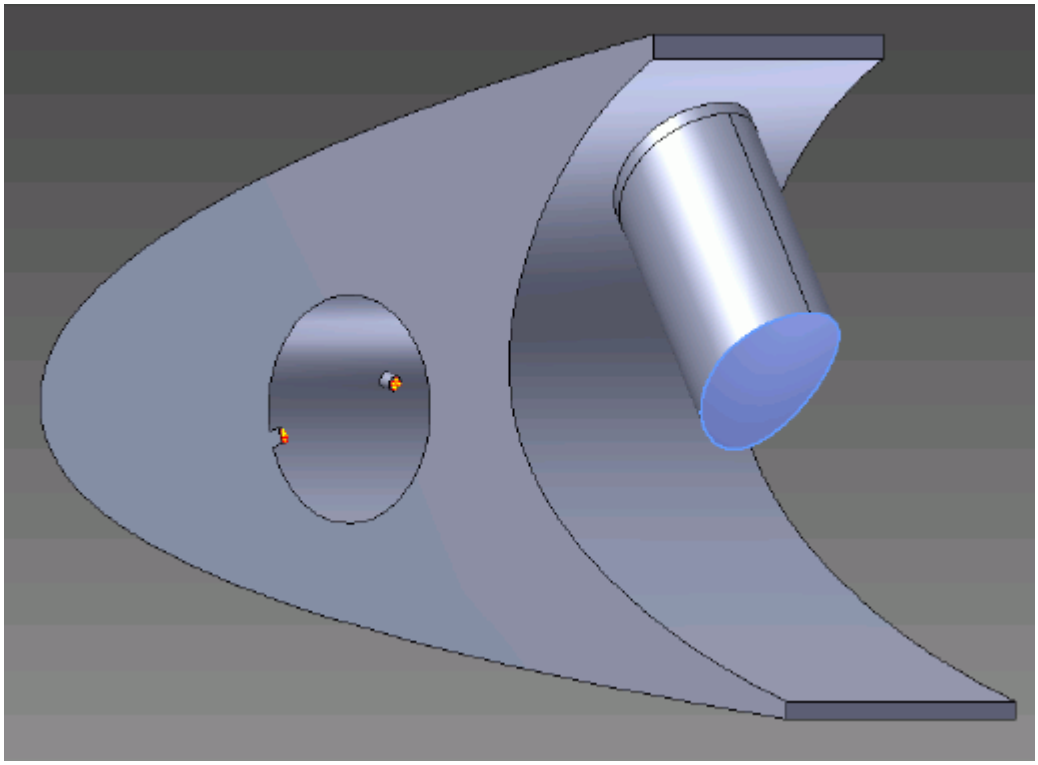
3. Set the following properties:

Parameter	Value
Type	**Velocity Inlet**
Boundary NameName	`Inlet`
(*Normal Velocity*)	`90 m/s`
(*Temperature*)	`460 K`
(*Turbulence Intensity*)	`0.135`
(*Turbulent Length Scale*)	`2.2e-005 m`

4. Click **OK**.

Create the outlet boundary:

5. Using the same technique, create another boundary.
6. Select the face that is highlighted below.



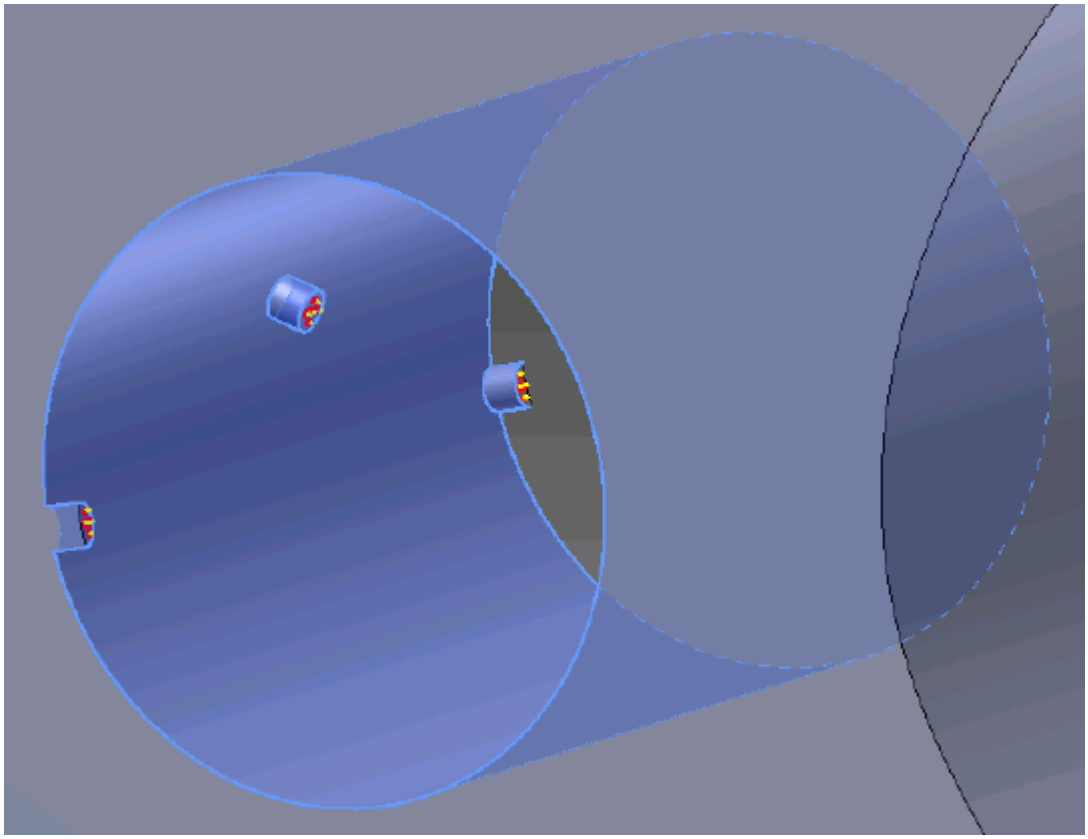
7. Set the following parameters:

Parameter	Value
Type	**Pressure Outlet**
Name	`Outlet`

8. Click **OK**.

Create the wall boundary that models the heated piccolo walls.

9. Create another boundary.
10. Select the four faces highlighted below, make sure to select the faces that are perpendicular to the inlet boundaries.



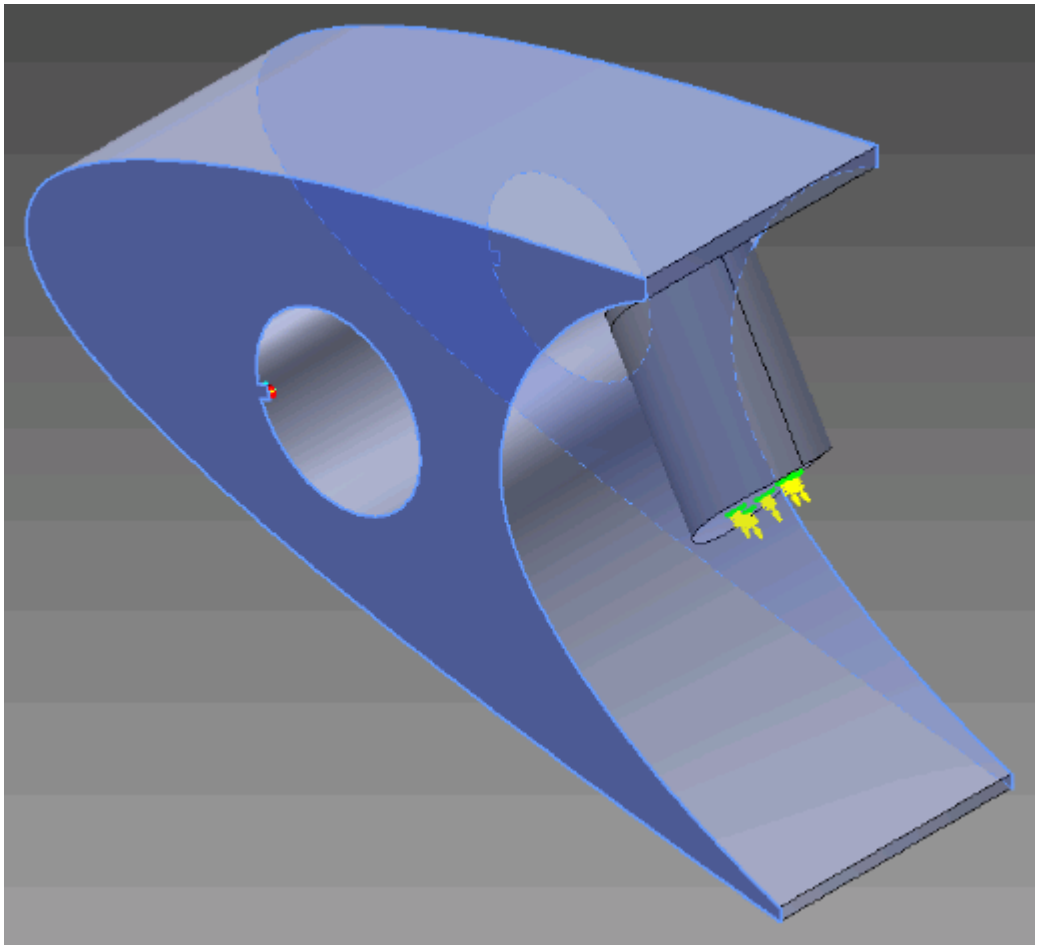
11. Set the following parameters:

Parameter	Value
Type	**Wall**
Name	`Piccolo Wall`
Thermal Specification	**Temperature**
(*Temperature*)	`460 K`

12. Click **OK**.

Create the symmetry boundaries:

13. Create a new boundary.
14. Select the two faces that are highlighted below, one on either side of the internal air region.



15. Set the following parameters:

Parameter	Value
Type	**Symmetry Plane**
Name	`Air Symmetry`

16. Click **OK**.

Defining Boundary Conditions for the Solid Region

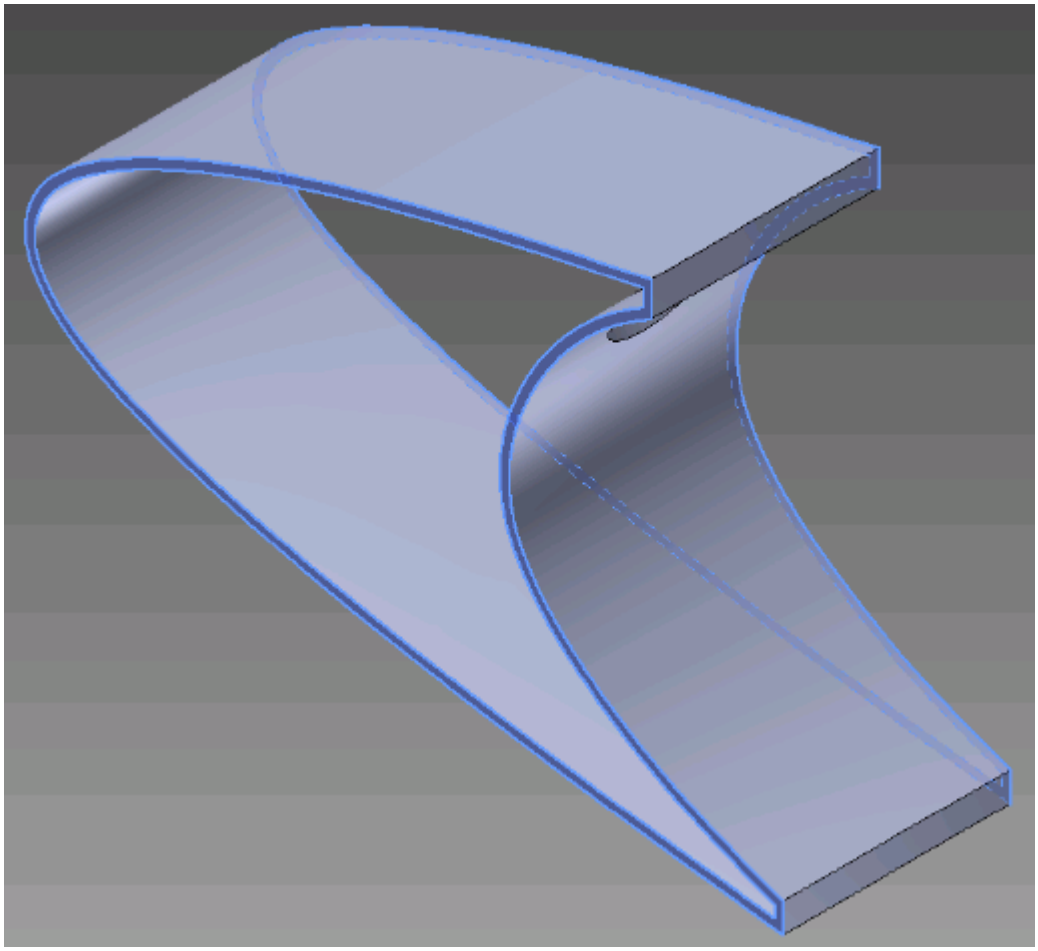
Use the convection boundary condition to model the heat propagation from the solid wing.

The convective boundary condition models ambient conditions on the outside of the domain. Use convection to model the heat propagation in the solid wing skin.

Create the symmetry boundary first:

1. Right-click the **Regions** > **Leading Edge** > **Boundaries** manager and select **New**.

   The *Boundary Settings* dialog appears.
2. Select the two faces highlighted below, one on either side of the geometry.



3. Set the following parameters:

Parameter	Value
Type	**Symmetry Plane**
Name	`Solid Symmetry`

4. Click **OK**.

Modify the default boundary so that it uses convection:

5. Right-click the **Regions** > **Leading Edge** > **Boundaries** > **Default** node and select **Edit**.

   The *Boundary Settings* dialog appears.

6. Set the following properties:

Parameter	Value
Thermal Specification	**Convection**
HTC (*Heat Transfer Coefficient*)	`100 W/m^2.K`
(*Temperature*)	`228 K`

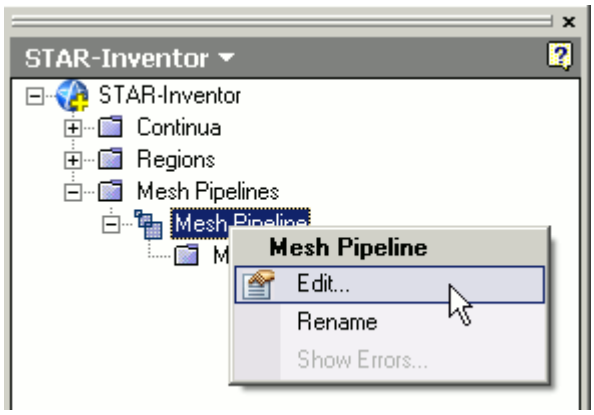
7. Click **OK**.

8. Save your progress.

Setting the Mesh Pipeline

To obtain more accurate results, modify the mesh pipeline:

1. Right-click the **Mesh Pipelines** > **Mesh Pipeline** node in the STAR-Inventor tree and select **Edit**.



The *Mesh Pipeline Settings* dialog appears.

2. Set the following parameters:

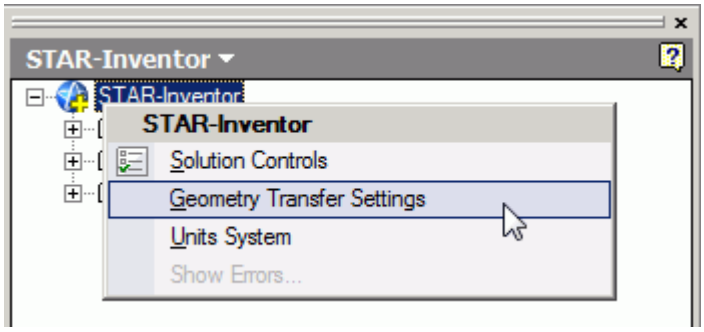
Parameter	Value
Base Size	
Relative to bodies	**Deactivated**
(*Base Size*)	`0.007 m`
Prism Layer	
(*Prism Layer Thickness*)	`10%`

3. Click **OK**.

Setting the Coincident Faces Tolerance

The geometry that is used in this tutorial has large changes in feature size and requires you to reduce the Coincident Faces Detection Tolerance.

1. Right-click STAR-Inventor and select **Geometry Transfer Settings**.



The *Geometry Transfer Settings* dialog appears.

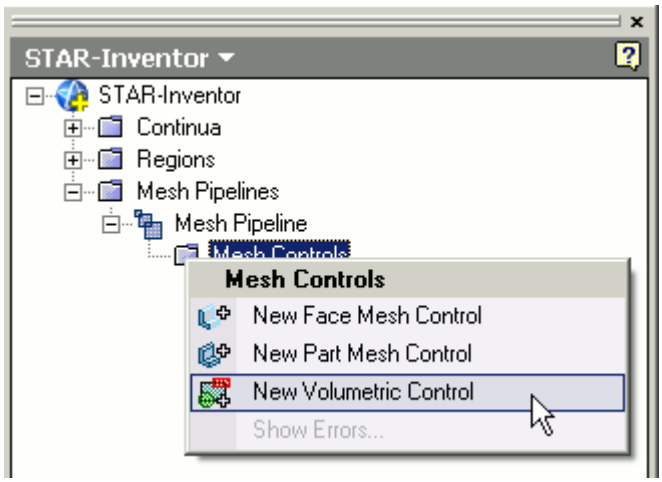
2. Set (*Coincident Faces Tolerance*) to `0.0001`.
3. Click **OK**.

Creating the Volumetric Mesh Controls

Create volumetric mesh control regions to capture the finer details of the flow near the air jets that heat the leading edge of the wing.

Create the volumetric mesh control for the inlet:

1. Right-click the **Mesh Pipelines** > **Mesh Pipeline** > **Mesh Controls** node in the STAR-Inventor tree and select **New Volumetric Control**.



The *Volumetric Control* dialog appears.

2. Set *Name* to `Inlet Box`.
3. Activate the *Use Absolute Value* option.
4. Set *Custom Surface Mesh Size* to `0.0005 m`.
5. Enter the following coordinates for Point 1:

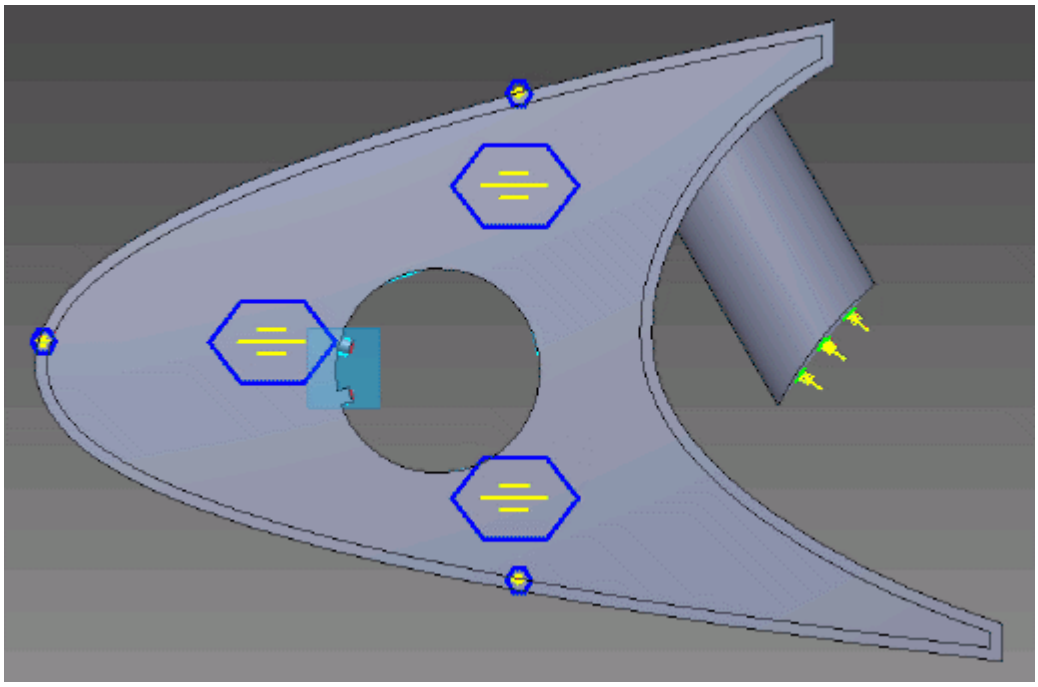
Coordinate	Value
X1	`-100.00`
Y1	`21.00`
Z1	`-20.00`

6. Enter the following coordinates for Point 2:

Coordinate	Value
X2	`-90.00`
Y2	`10.00`
Z2	`60.00`

7. Click **Create**.

   The volumetric control region is shown below.



8. Click **OK**.

Create the volumetric mesh control for the region near the piccolo tube and use the following settings:

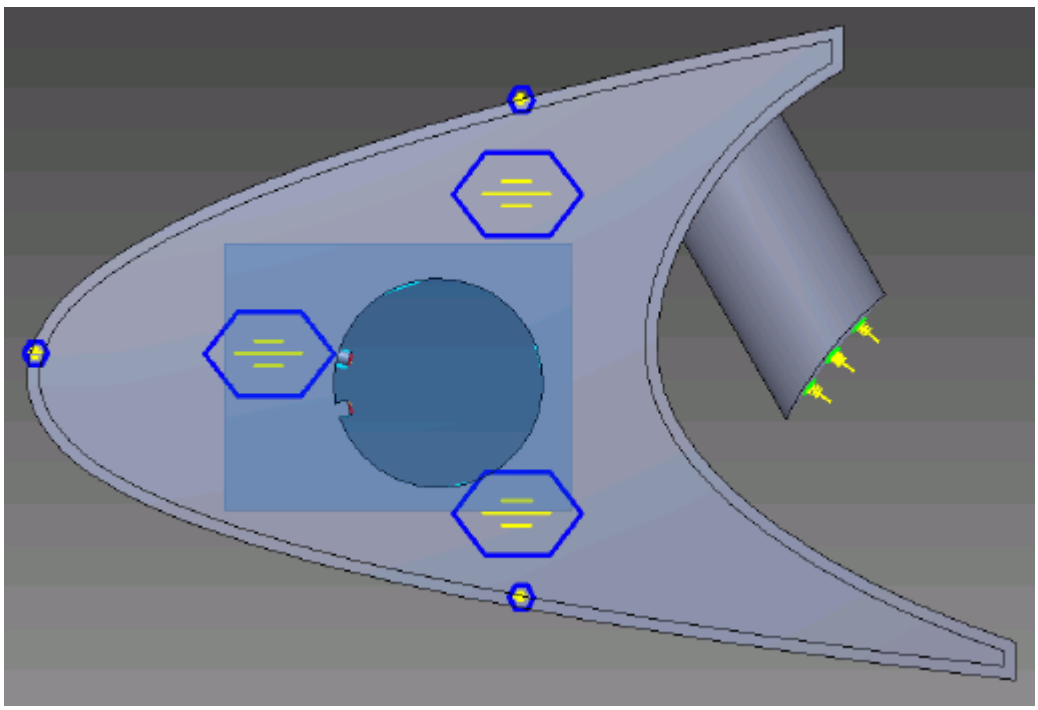
9. Using the same technique, create another volumetric mesh control with the following settings:

Parameter	Value
Name	`Mid Box`
Custom Surface Mesh Size	
Use Absolute Value	**Activated**

Parameter	Value
Custom Surface Mesh Size	0.0015 m
Shape	
X1	-111.00
Y1	34.00
Z1	-20.00
X2	-64.00
Y2	-2.00
Z2	60.00

10. Click **Create**.

    The volumetric control region is shown below.

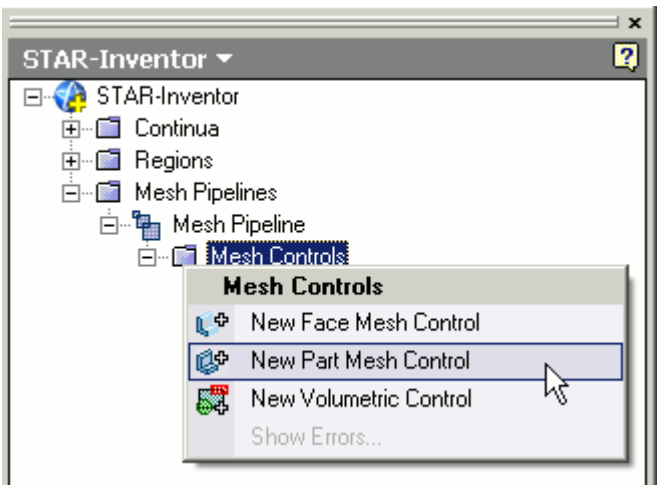


11. Click **OK**.
12. Save your progress.

Creating a Part Mesh Control

Create a part mesh control to disable the prism layer generation in the solid region. Solid regions do not require prism layers.

1. Right-click **Mesh Pipelines** > **Mesh Pipeline** > **Mesh Controls** in the STAR-Inventor tree and select **New Part Mesh Control**.



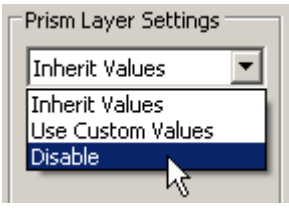
The *Part Mesh Control* dialog appears.

2. Select the solid wing region on the CAD model.

   The selection is added to the *Part Selection* box.

3. Set *Name* to `Solid Region`.

4. In the *Prism Layer* box, set (*Prism layer option*) to **Disable** using the drop-down menu.

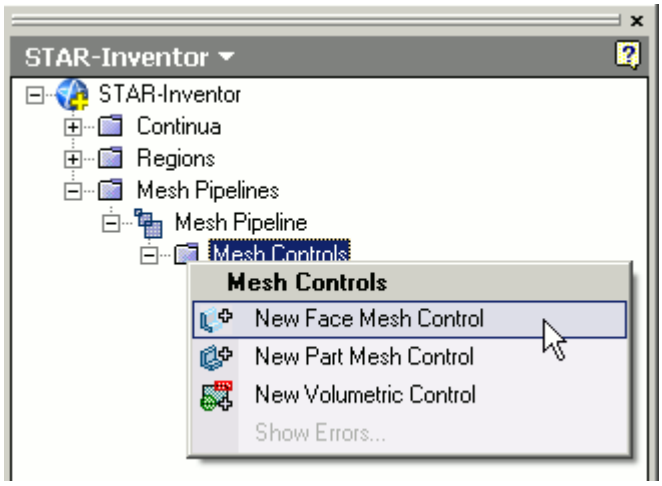


5. Click **OK**.

Creating a Face Mesh Control

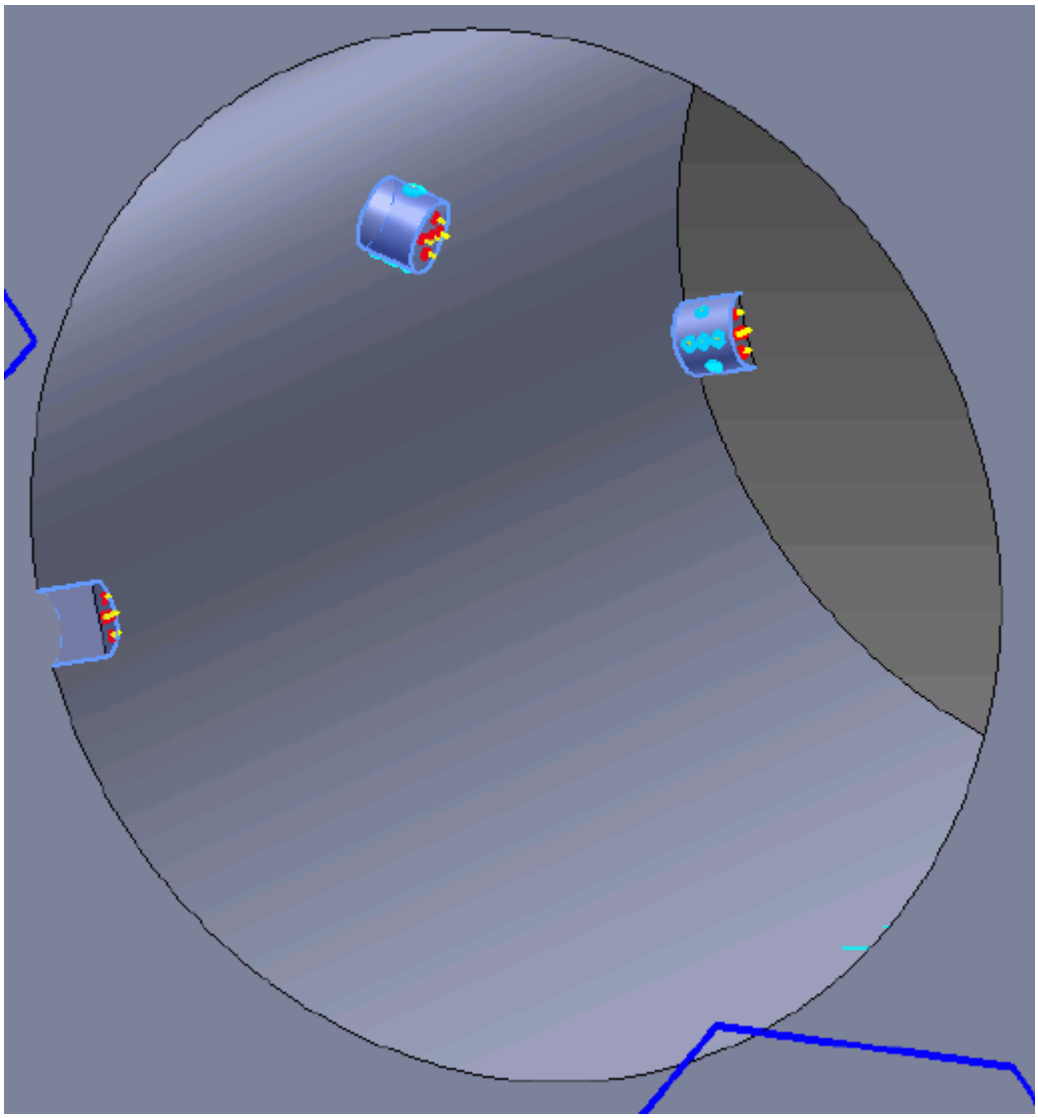
Create a face mesh control to reduce the size of the prism layer thickness at the inlet boundary.

1. Right-click **Mesh Pipelines** > **Mesh Pipeline** > **Mesh Controls** in the STAR-Inventor tree and select **New Face Mesh Control**.



The *Face Mesh Control* dialog appears.

2. Select the cylindrical faces that are perpendicular to the inlet boundary, as shown below.

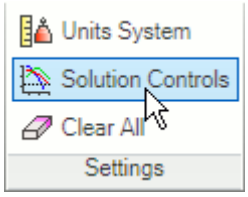


3. Set *Name* to `Inlet Refinement`.
4. Activate the *Custom Surface Mesh Size* option.
5. Activate the *Use Absolute Value* option.
6. Set (*Custom Mesh Size*) to `0.00025`.
7. In the *Prism Layer* box, select **Use Custom Values** from the drop-down list.
8. Activate the *Use Absolute Value* option.
9. Set (*Prism Layer Thickness*) to `0.0002` m.
10. Click **OK**.

Modifying Solution Controls

Modify the maximum number of iterations and the solution accuracy. To help accelerate convergence, edit the solid under-relaxation factor:

1. Click **Solution Controls** on the STAR-Inventor ribbon.



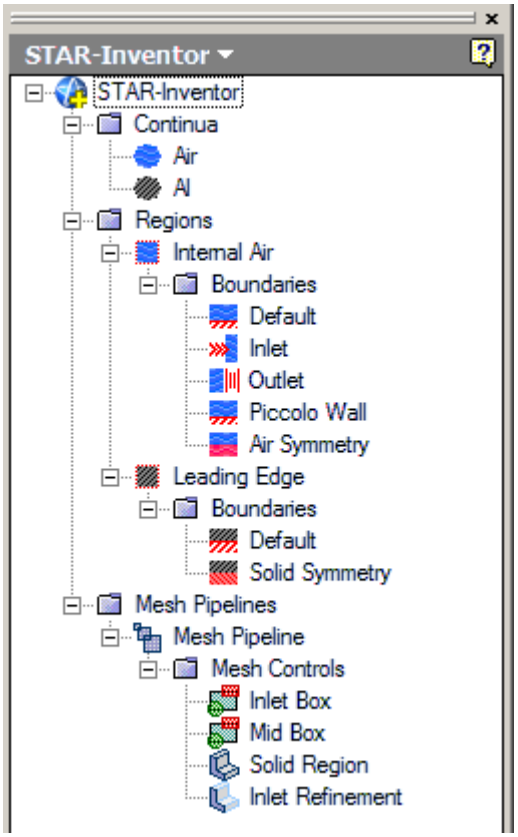
   The *Solution Controls* dialog appears.
2. Set the following parameters:

Parameter	Value
(*Number of Iterations*)	`500`
(*Convergence Level*)	`0.01`
(*Solid Temperature Under-Relaxation Factor*)	`0.9999`

3. Click **OK**.
4. Save your progress.

Reviewing the Simulation Set Up

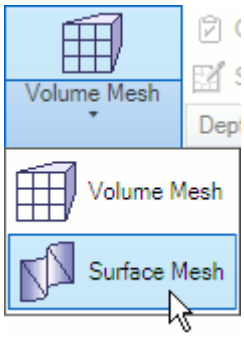
The final STAR-Inventor object tree is shown below.



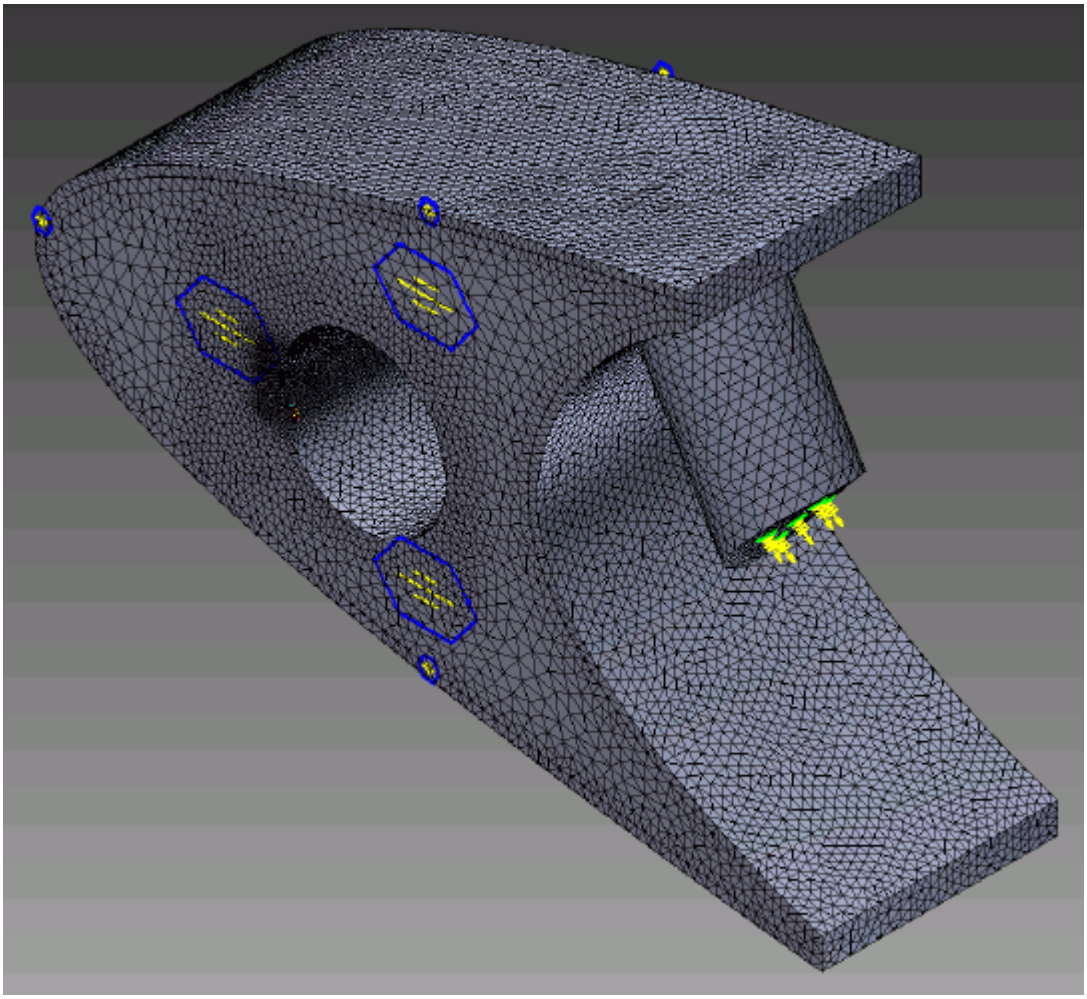
Generating the Surface Mesh

Generate the surface mesh:

1. From the STAR-Inventor ribbon, click **Surface Mesh**.From the STAR-Inventor ribbon, click ▾ next to the **Volume Mesh** icon and select **Surface Mesh**.



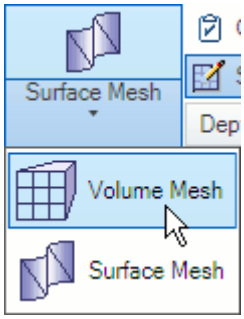
The surface mesh is shown below.



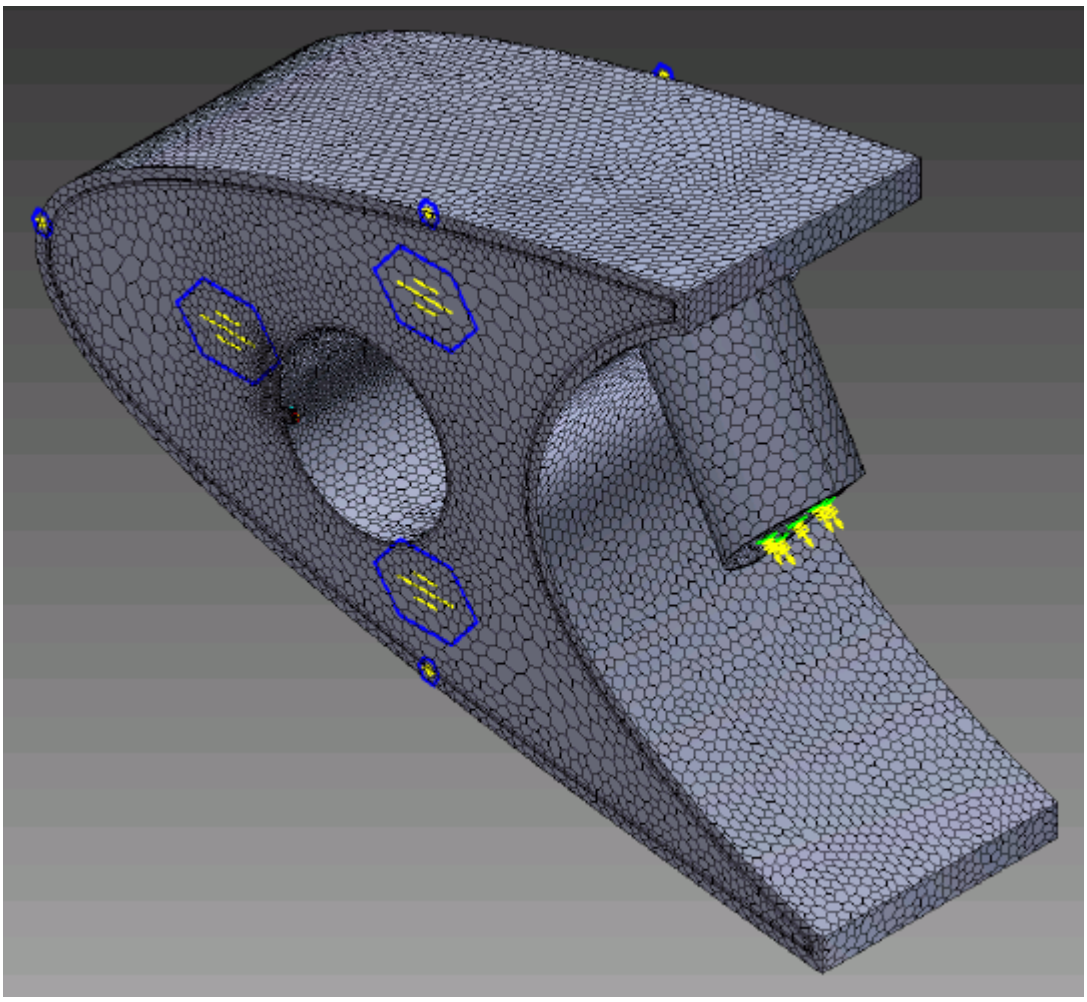
Generating the Volume Mesh

Generate the volume mesh:

1. From the STAR-Inventor ribbon, click **Volume Mesh**.From the STAR-Inventor ribbon, click ▾ next to the **Surface Mesh** icon and select **Volume Mesh**.



The volume mesh is shown below.

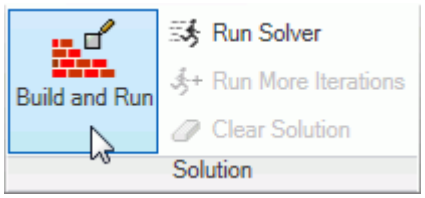


Building and Running

The case is now ready to run.

To run the simulation:

1. From the ribbon, click **Build and Run**.



The *Residuals* plot appears in the STAR-Inventor *Output* window. You can use this plot to monitor the convergence of the solution.

When the solver is done, STAR-CCM+ launches.

Post-Processing Results

You perform all post-processing in STAR-CCM+. By default, a scalar scene, *Scalar Scene 1*, automatically appears in the *Graphics* window when the solver finishes. This scene displays the absolute total pressure.

The results that are contained in the following sections are representative. You may see some variation in the results that you obtain, depending on what platform and package you are using.

1. In STAR-CCM+, click **File** > **Save** to save the simulation.

Contents:

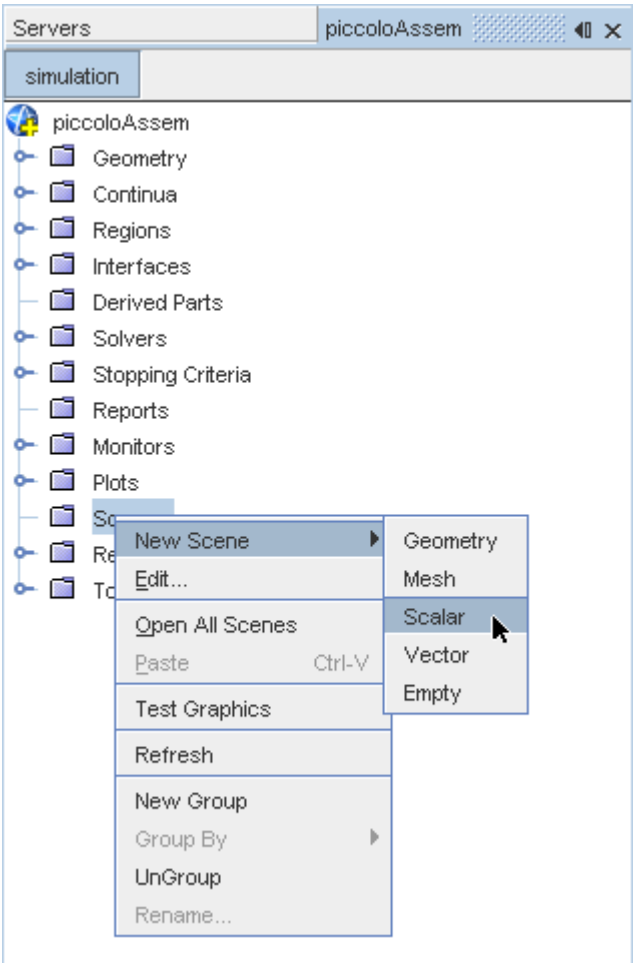
Creating a Scalar Scene
Creating A Streamlines Scene
Creating a Vector Scene
Creating a Streamlines Scene
Creating A Scalar Scene
Visualizing the Wing Skin Temperature
Creating a Streamlines Scene
Creating A Scalar Scene
Creating a Streamlines Scene
Creating a Scalar Scene
Modifying The Scalar Scene
Creating a Streamlines Scene

Creating A Scalar Scene

Create a scalar scene containing a cross-section of the model. Plot temperature on this section to demonstrate how the hot gas coming from the piccolo tube affects the wing skin and the air inside the wing.

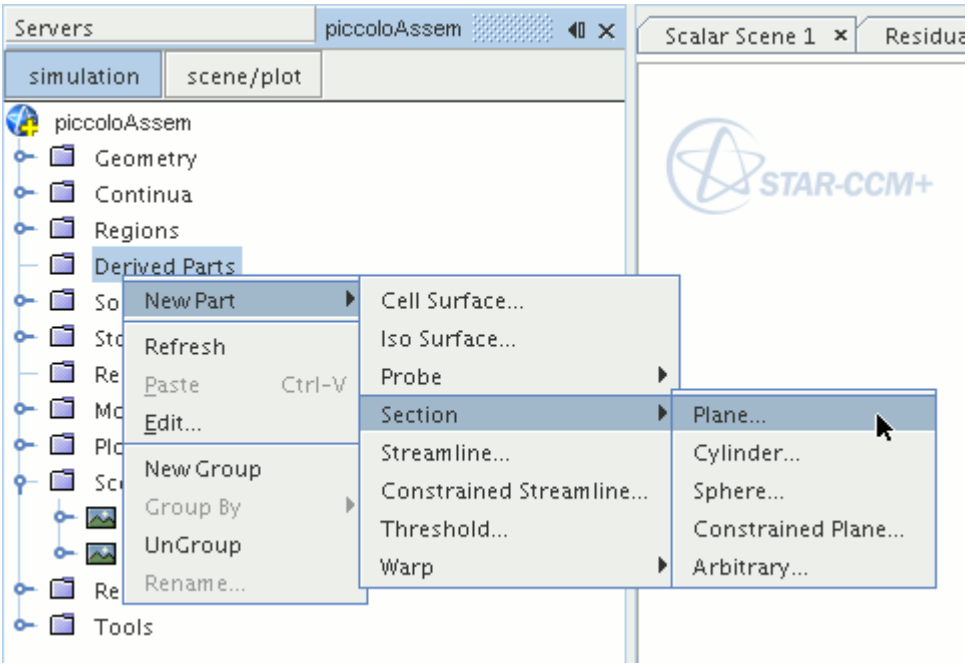
To create the scene:

1. Right-click the **Scenes** node in the explorer pane and select **New Scene** > **Scalar**.



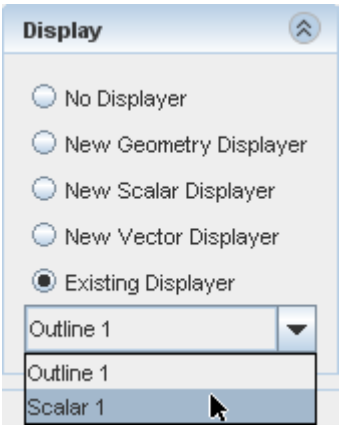
A new scalar scene, *Scalar Scene 2*, is created and selected in the *Graphics* window.

2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane...**.



The *Create Plane Section* dialog appears.

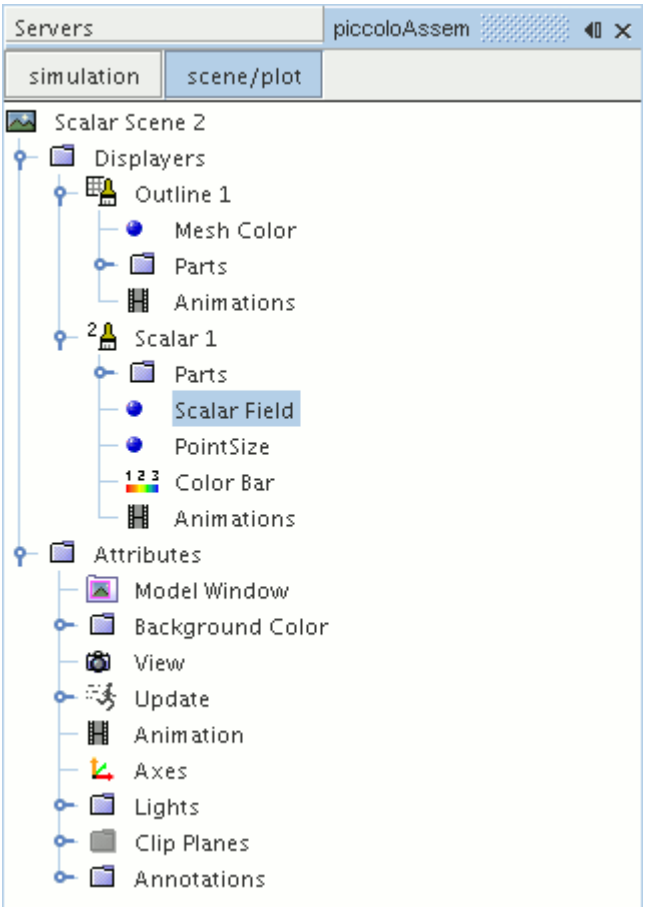
3. Set the origin coordinates to `[0.0 mm, 0.0 mm, 18.4 mm]`.
4. Set the *normal* vector to `[0.0 mm, 0.0 mm, 1.0 mm]`.
5. In the *Display* group box, select **Existing Displayer**, and select **Scalar 1** from the drop-down menu.



6. Click **Create** followed by **Close**.

Change the scalar field to display temperature contours:

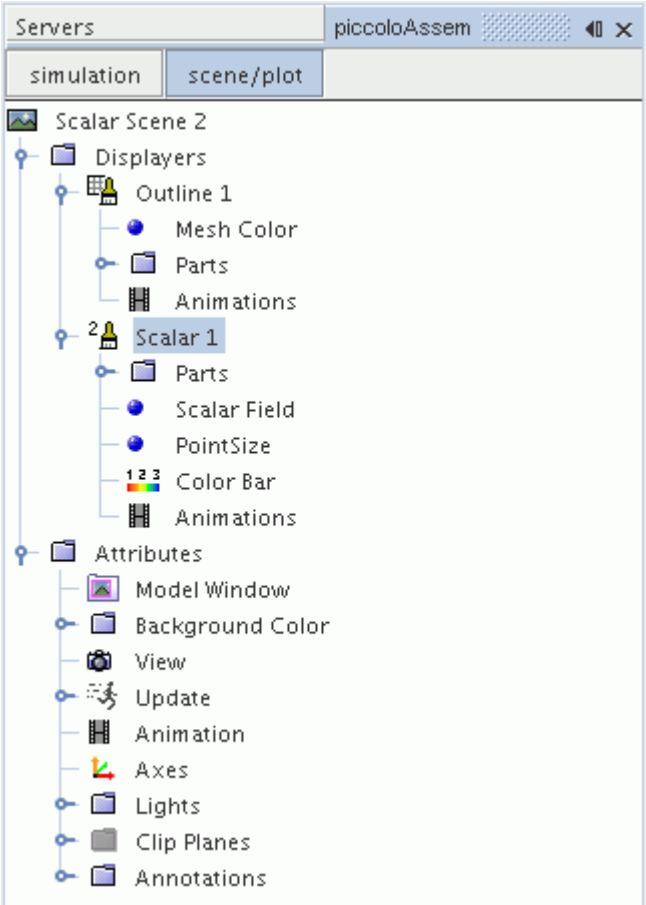
7. Click **scene/plot**.
8. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Scalar Field**.



9. Set *Function* to **Temperature**.

Modify the contour style to have a smooth transition between the different temperature values in the scene:

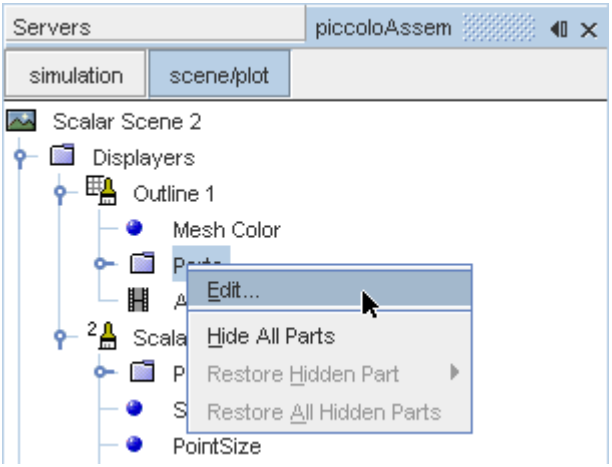
10. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1**.



11. Set *Contour Style* to **Smooth Filled**.

Adjust the outlines in the scene to show only the plane section:

12. Right-click **Scalar Scene 2** > **Displayers** > **Outline 1** > **Parts** and select **Edit...**.

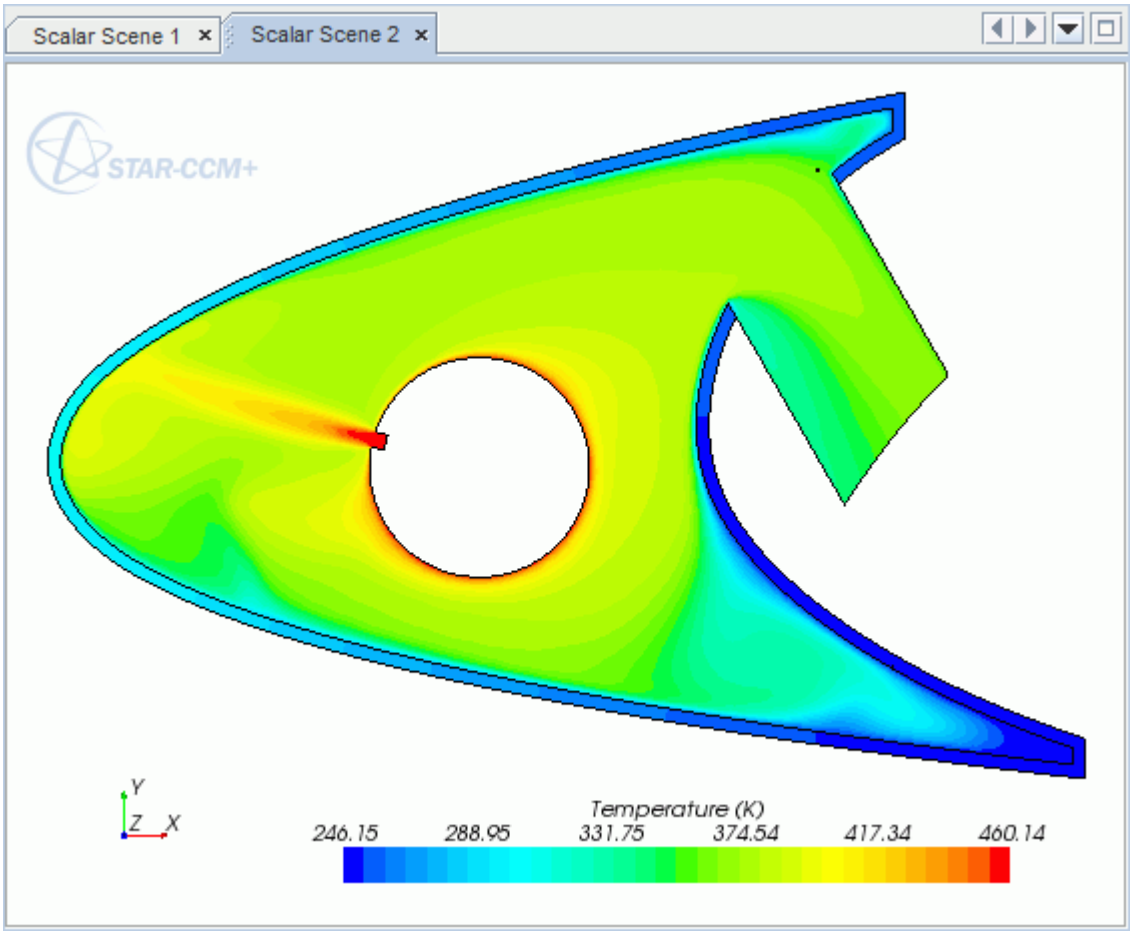


The *Parts* dialog appears.

13. Right-click on a blank area and choose **Deselect All**.

14. Select the **Derived Parts** > **plane section** node.
15. Click **OK**.
16. Position the scene to show the side view of the geometry.
17. Click **simulation**.
18. Click (**Save**) to save the simulation.

The finished scene is shown below.



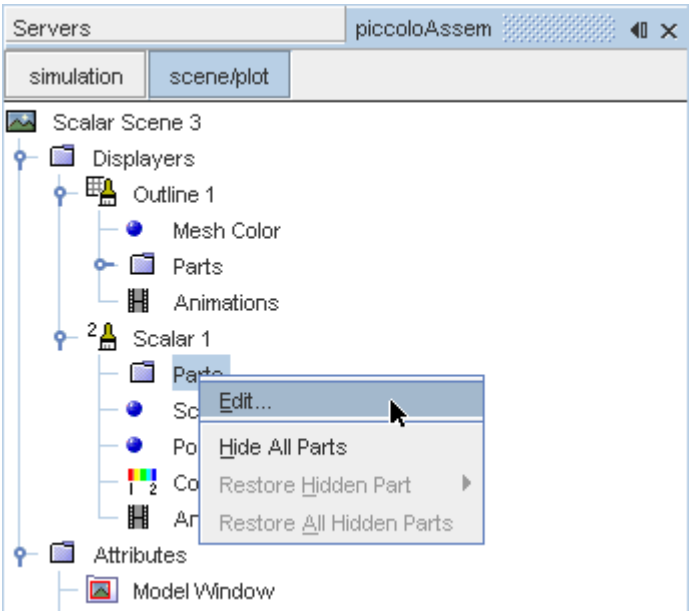
Visualizing the Wing Skin Temperature

Create another scalar scene to display the temperature on the wing skin geometry.

To create another scalar scene:

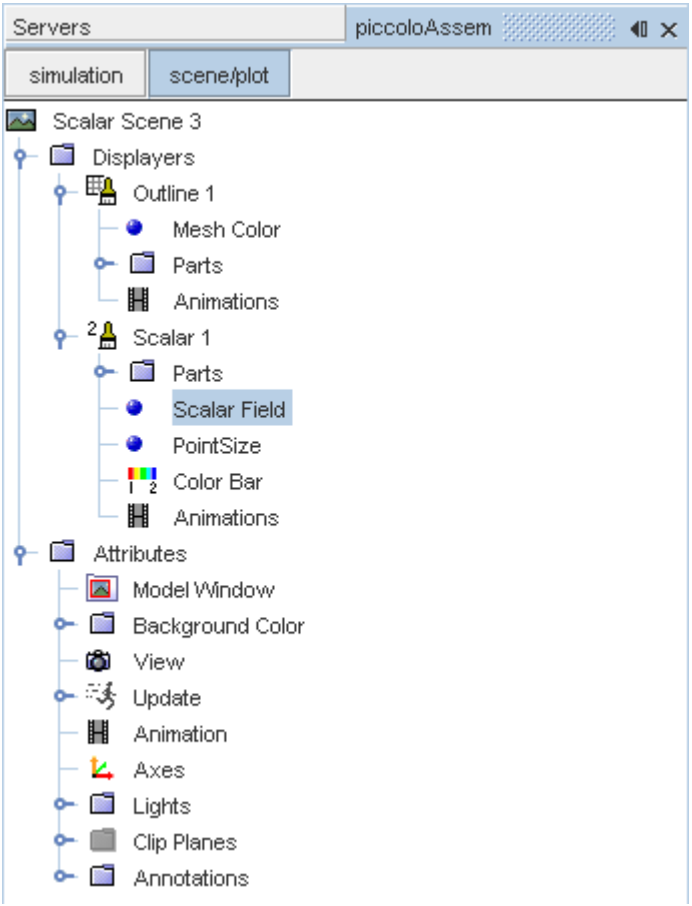
1. Right-click the **Scenes** node in the explorer pane and select **New Scene** > **Scalar**.

   A new scalar scene, *Scalar Scene 3*, is created and selected in the *Graphics* window.
2. Click **scene/plot**.
3. Right-click **Scalar Scene 3** > **Displayers** > **Scalar 1** > **Parts** and select **Edit...**.



The *Parts* dialog appears.

4. Expand the **Regions** node and select the **Leading Edge** node.
5. Click **OK**.
6. Navigate to **Scalar Scene 3** > **Displayers** > **Scalar 1** > **Scalar Field**.



7. Set *Function* to **Temperature.**

To have a smooth transition between the different temperature values, edit the contour style:

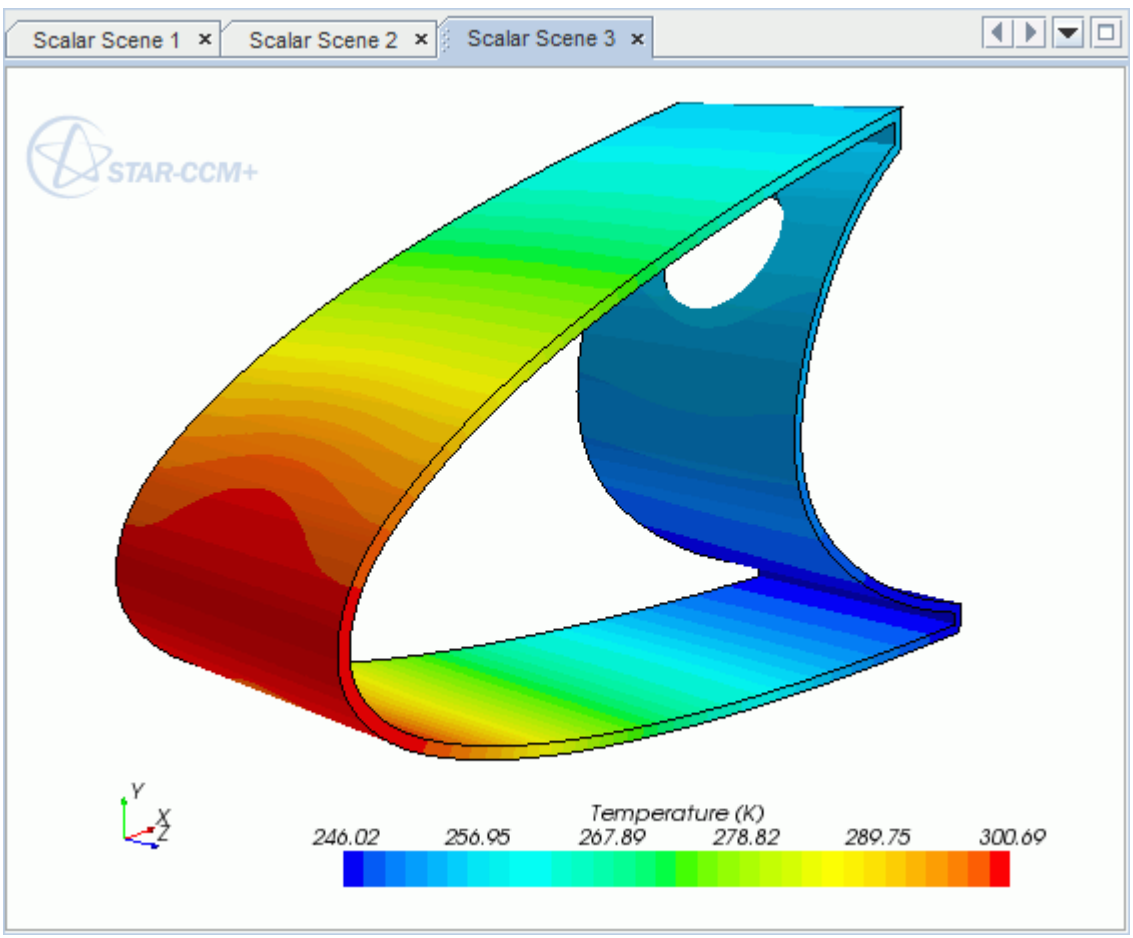
8. Navigate to **Scalar Scene 3** > **Displayers** > **Scalar 1**.
9. Set *Contour Style* to **Smooth Filled**.

Adjust the outlines to appear on the solid region only:

10. Right-click **Scalar Scene 3** > **Displayers** > **Outline 1** > **Parts** and select **Edit...**.

    The *Parts* dialog appears.
11. Expand the **Regions** node and deselect the **Internal Air** node.
12. Click **OK**.
13. Reorient the geometry to get a better view of the leading edge.
14. Click **simulation**.
15. Save the simulation.

The finished scene is shown below.

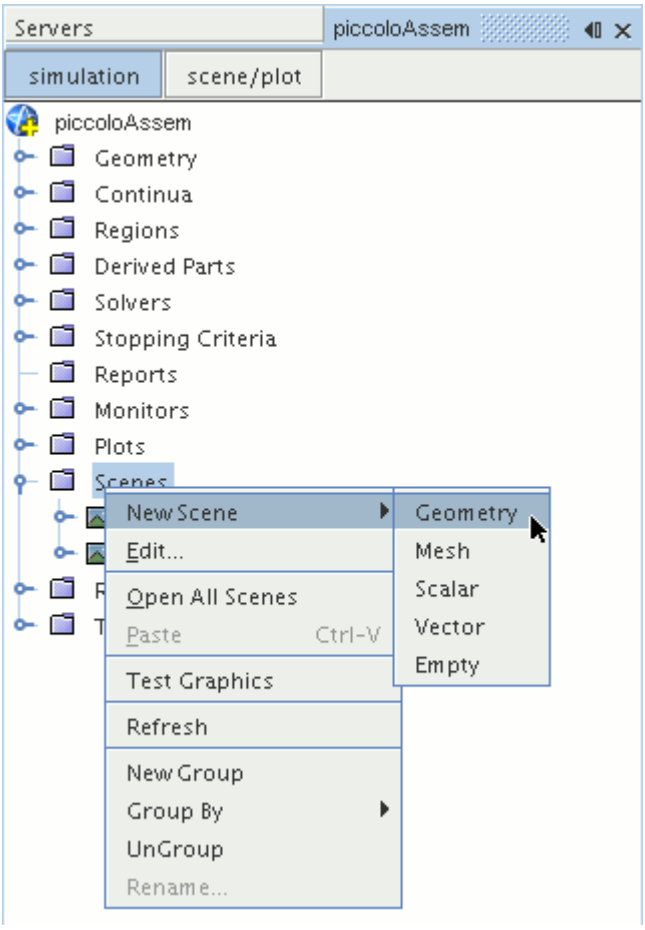


Creating a Streamlines Scene

Create streamlines to get a better understanding of how the hot air flows inside the leading edge of the wing.

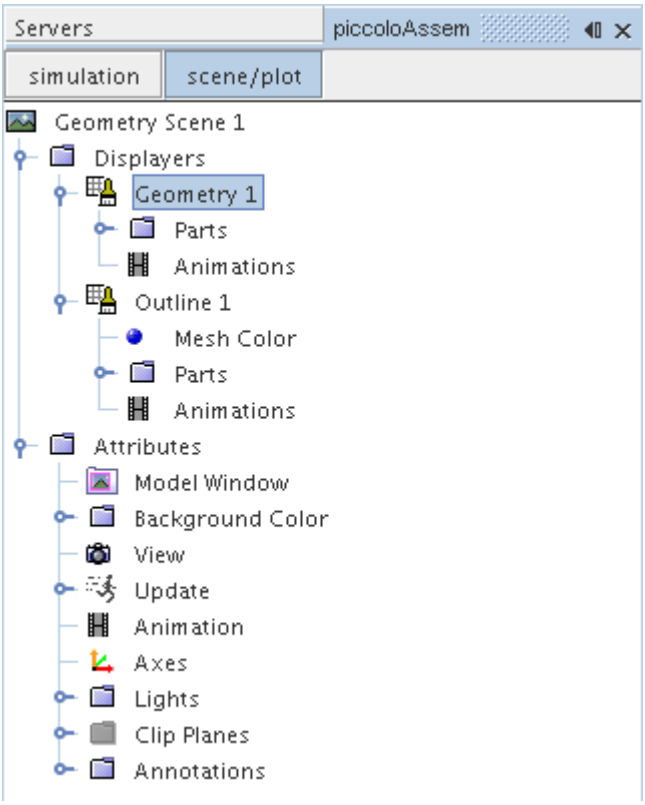
First, create a geometry scene:

1. Right-click the **Scenes** node in the object tree and select **New Scene** > **Geometry**.



   A new geometry scene, *Geometry Scene 1*, is created and selected in the *Graphics* window.

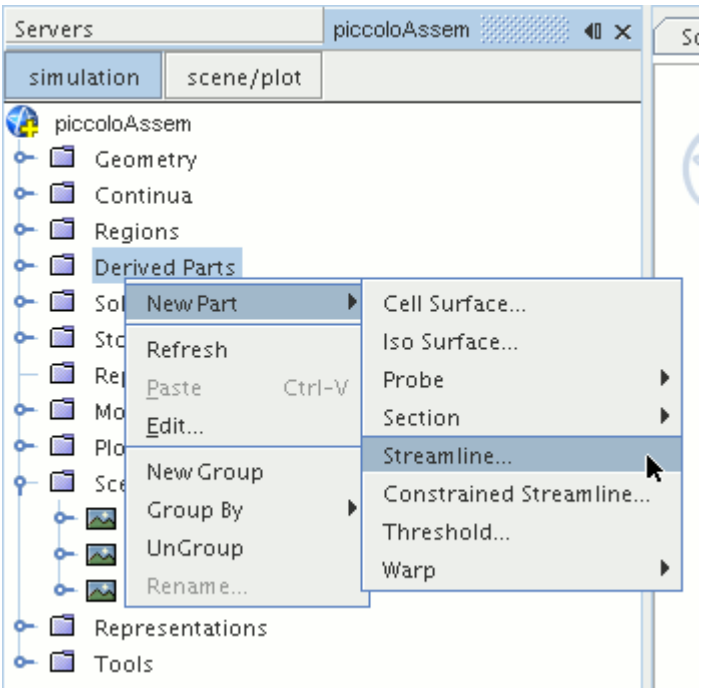
2. Click **scene/plot**.
3. Navigate to the **Geometry Scene 1** > **Displayers** > **Geometry 1** node.



4. Set *Opacity* to `0.1`.

Next, create the streamlines:

5. Click **simulation**.
6. Right-click the **Derived Parts** node and select **New Part** > **Streamline...**.

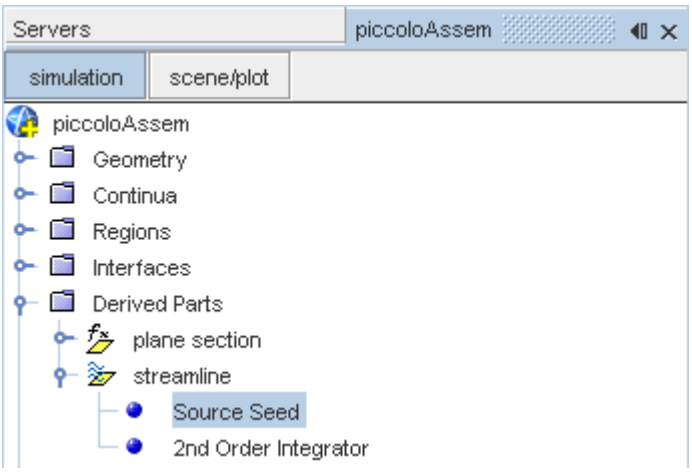


The *Create Streamline* dialog appears.

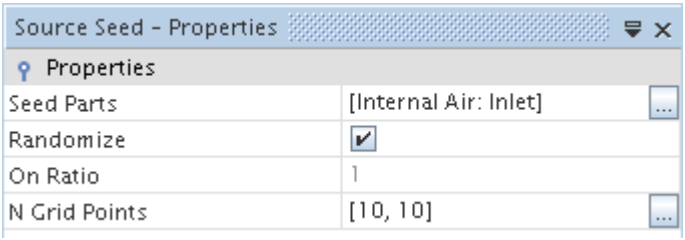
7. In the *Input Parts* box, click **Select...**.
8. In the *Select Objects* dialog, expand the **Regions** node and deselect **Leading Edge**.
9. Click **OK** to close the *Select Objects* dialog.
10. Make sure that **Part Seed** is selected in the *Seed Mode* section.
11. Set *Seed Parts* to **Regions** > **Internal Air** > **Boundaries** > **Inlet**.
12. Set *Part U-Resolution* to `10`.
13. Set *Part V-Resolution* to `10`.
14. Make sure that **New Streamline Displayer** is selected in the *Display* section.
15. Click **Create** followed by **Close**.

The streamlines do not yet appear in the geometry scene. Follow the steps that are outlined below to modify the streamlines:

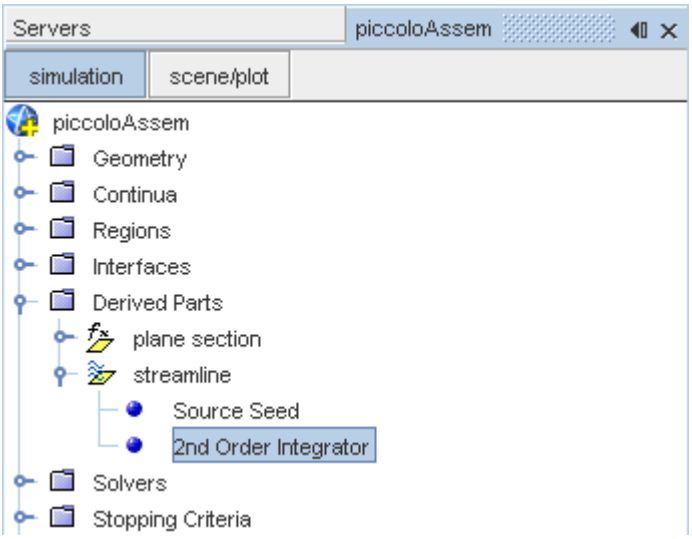
16. Navigate to **Derived Parts** > **Streamline** > **Source Seed**.



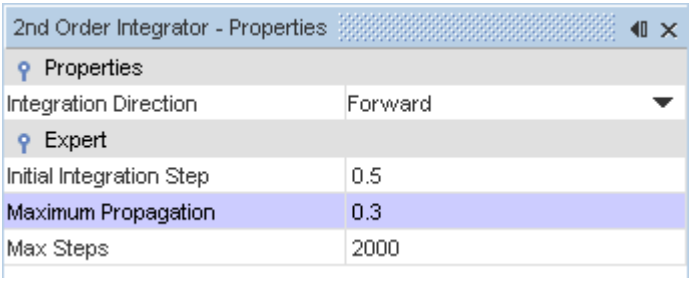
17. Activate the *Randomize* option.



18. Navigate to **Derived Parts** > **Streamline** > **2nd Order Integrator**.



19. Set *Maximum Propagation* to `0.3`.



Adjust the appearance of the streamlines to display tubes instead of the default lines:

20. Click **scene/plot**.
21. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1**.
22. Set *Mode* to **Tubes** and *Width* to `0.00025`.

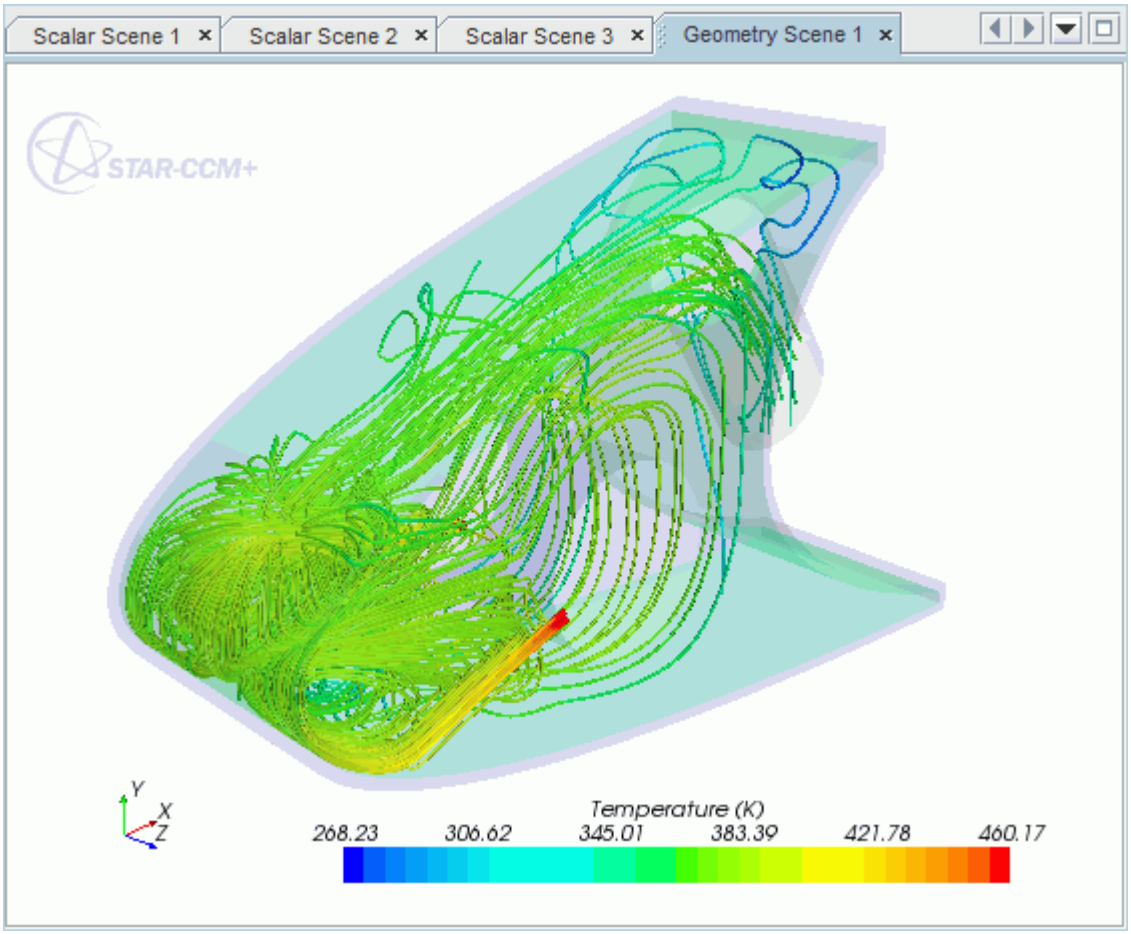
Adjust the scalar field of the streamlines to display temperature:

23. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** > **Scalar Field**.
24. Set *Function* to **Temperature**.

Turn off the outlines in the geometry scene to have a better view of the streamlines.

25. Right-click the **Geometry Scene 1** > **Displayers** > **Outline 1** and select **Toggle Visibility**.
26. Reorient the geometry to get a better view of the leading edge.
27. Save the simulation.

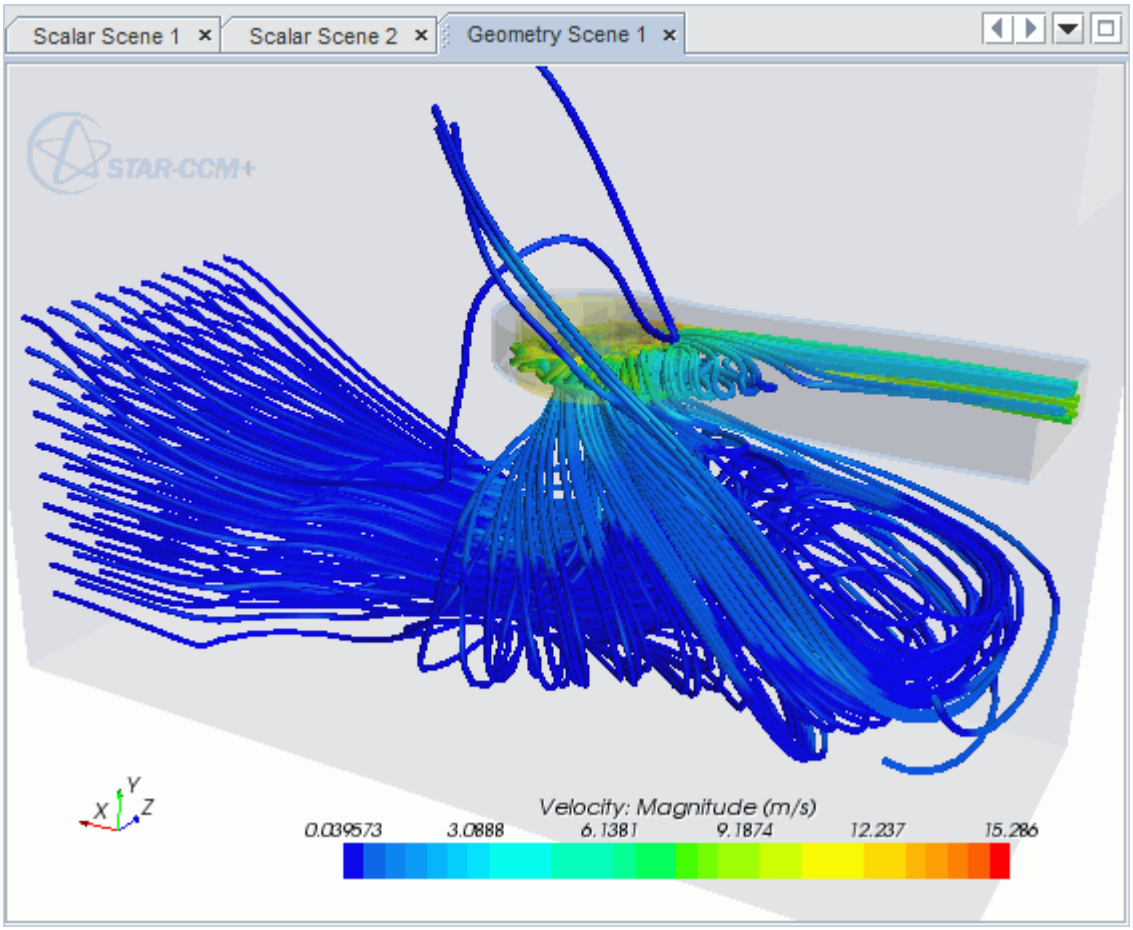
The finished scene is shown below.


Scalar Scene 1
Scalar Scene 2
Scalar Scene 3
Geometry Scene 1
STAR-CCM+
Y
X
Z
Temperature (K)
268.23
306.62
345.01
383.39
421.78
460.17

Rotating Flow Inside a Graphics Card

This tutorial takes you through the steps that are required to run a CFD calculation in the STAR-Inventor environment.

For this tutorial, the model is a fully sealed graphics card that is located inside an empty, idealized PC tower case. The geometry that is used in this tutorial has been simplified on order to decrease the computational cost. As a consequence the mesh is relatively coarse at approximately 200,000 cells compared to the 3-4 million cells used for an industrial analysis.



Contents:

Opening the CAD Model
Selecting Fluid Physics Models
Setting the Rotating Region
Defining Boundary Conditions
Setting the Mesh Pipeline
Creating the Volumetric Mesh Controls
Modifying Solution Controls

Reviewing the Simulation Set Up
Generating The Surface Mesh
Generating the Volume Mesh
Building and Running
Post-Processing Results

Opening the CAD Model

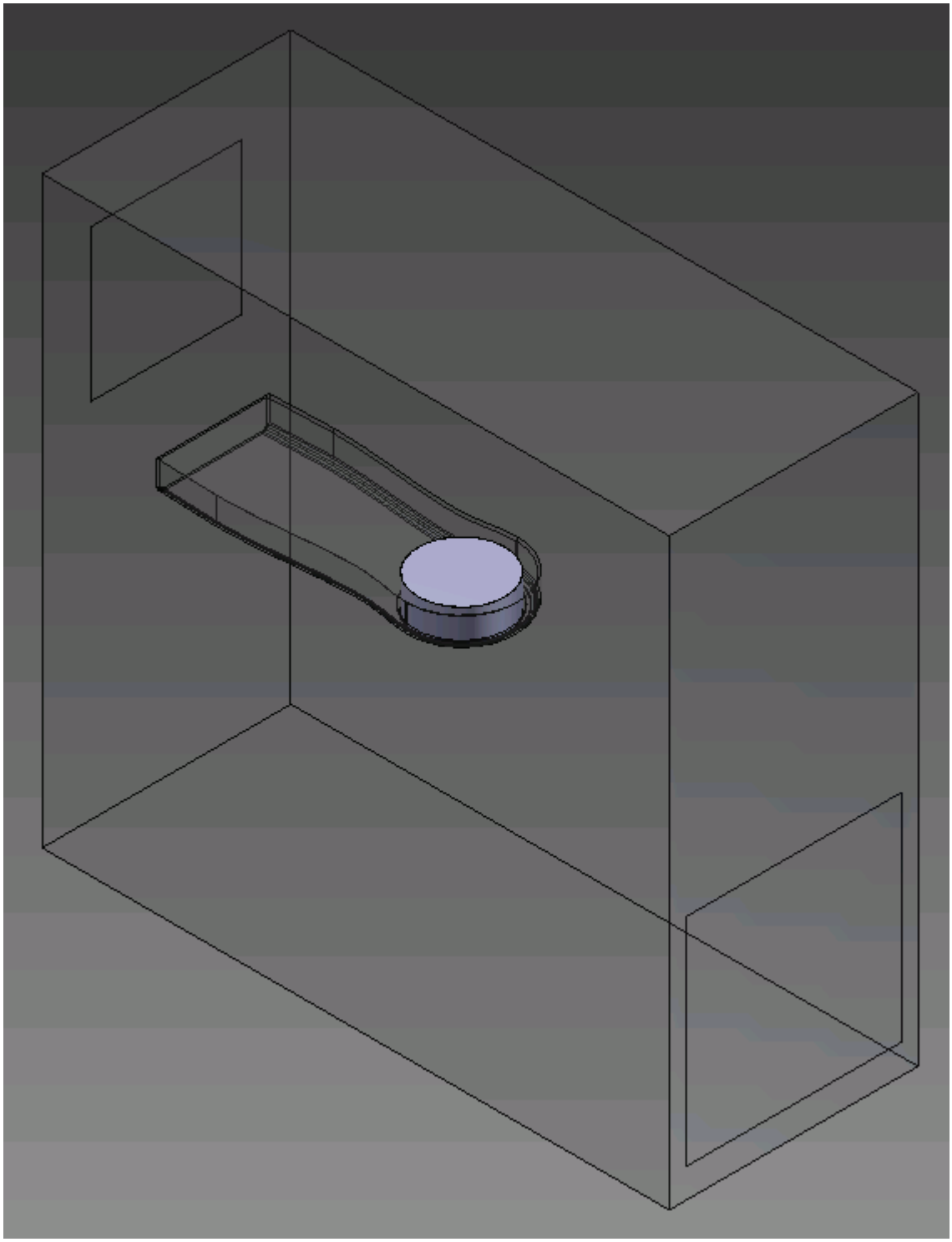
A directory containing the CAD geometry is provided with the STAR-CCM+ distribution.

This directory is located in `[STAR-CCM+_INSTALL_DIR]\doc\startutorialsdata\starinventor\data`.

1. Copy directory `data` to your own disk space and open the part file `gpuAssem.iam` in STAR-Inventor.

   The assembly is made up of two parts: the PC tower case (including the GPU housing), and the rotating region that encloses the blower fan of the GPU.

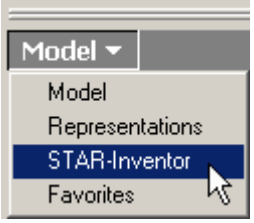
   The imported geometry is shown below.



The STAR-Inventor ribbon appears at the top of the screen.The STAR-Inventor object tree is located on the left of the screen.

To switch to this object tree:

2. Click **Model** and select **STAR-Inventor**.



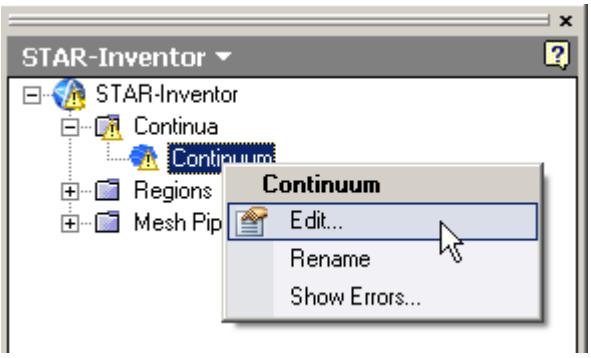
If these items are not visible, refer to Launching STAR-Inventor Manually.

Selecting Fluid Physics Models

The first step in setting up this tutorial is to define the physics of the problem. For this case, the flow is turbulent and isothermal. Temperature calculations are on by default in STAR-Inventor.

To turn temperature calculations off:

1. Right-click the **Continua** > **Continuum** node in the STAR-Inventor tree and select **Edit**.

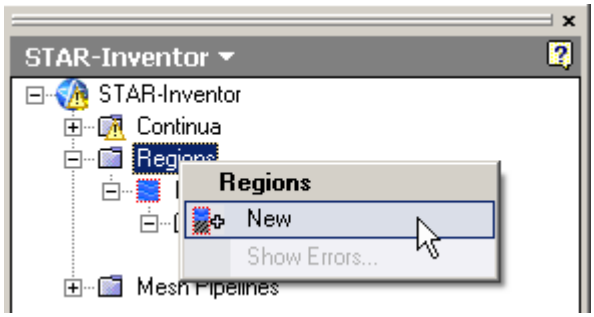


   The *Continuum Settings* dialog appears.
2. Deactivate the **Temperature** option in the *Optional Physics Models* box.
3. Click **OK**.

Setting the Rotating Region

Create the rotating region that contains the GPU fan:

1. Right-click the **Regions** node in the STAR-Inventor object tree and select **New**.

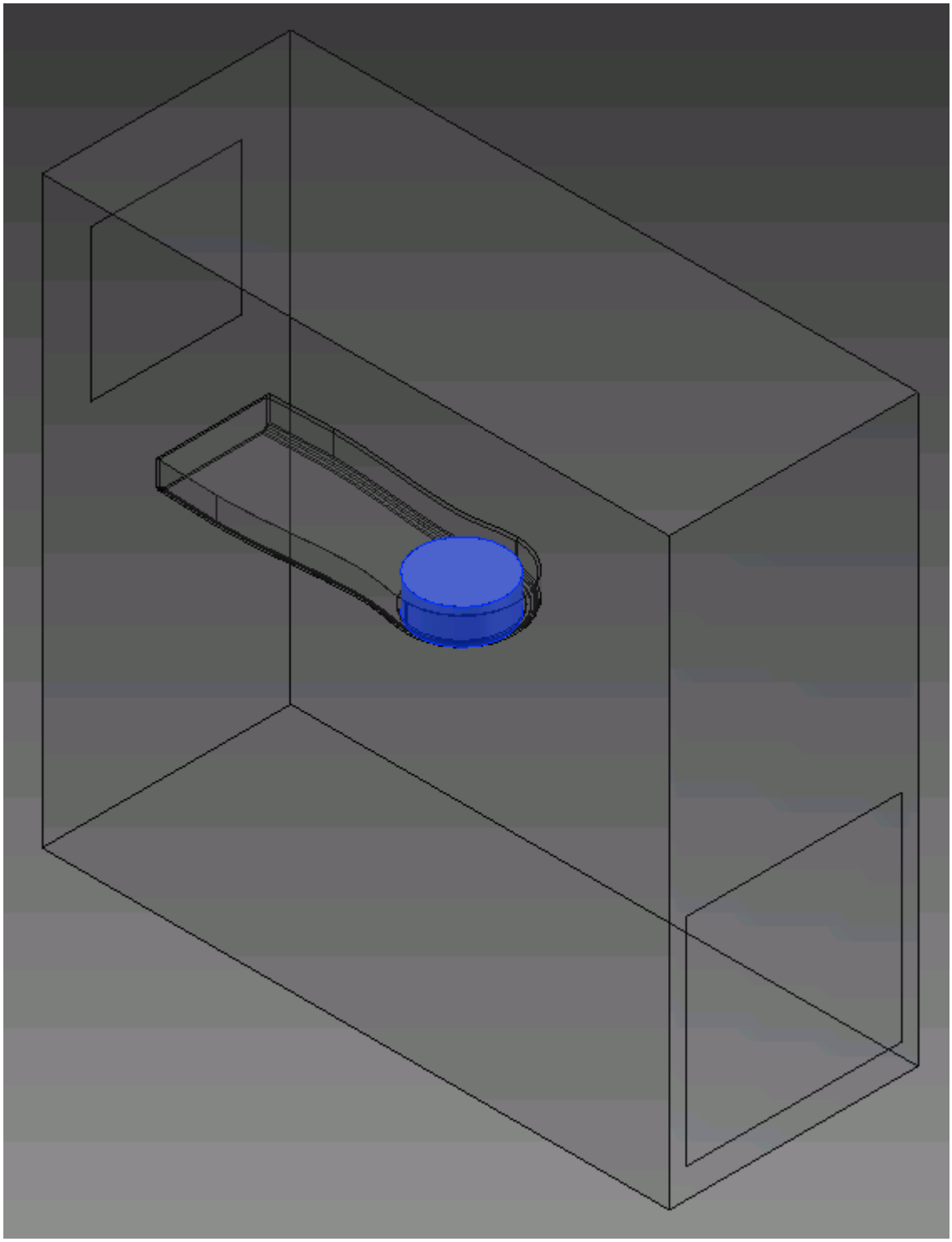


The *Region Settings* dialog appears.

2. Select the disc that is located inside the graphics card housing.

   You can select the part from within the PC tower by right-clicking on the disc and choosing **Select Other...**. The disc is highlighted in white, indicating a preliminary selection. Left-click to confirm the selection.

   Once selected, the part becomes highlighted as shown below.



3. Set *Name* to `Rotating Region`.
4. Activate the **Moving Reference Frame** option.
5. Set *Rotational Velocity* to `-2800 rpm`.

Note: Make sure that you set the units to `rpm` using the drop-down menu before entering the value.

6. Set the origin of the axis of rotation to:

Property	Value
X	-122.1761
Y	-1.9434
Z	-19.0609

7. Set the components of the axis of rotation to:

Property	Value
(*i component*)	0
(*j component*)	1
(*k component*)	0

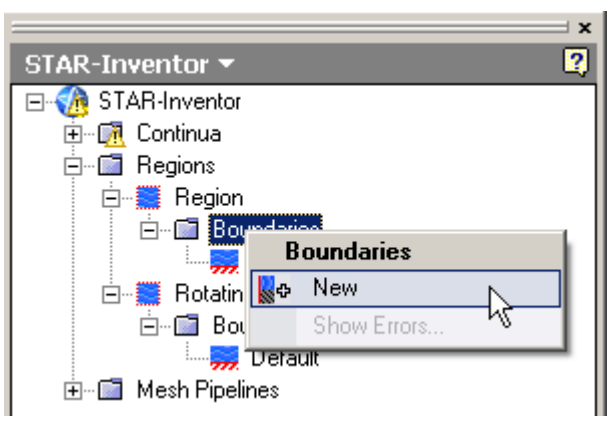
8. Make sure that the *Use Assembly Coordinate System* is activated.
9. Click **OK**.

Defining Boundary Conditions

Create the inlet and outlet boundary conditions for this simulation.

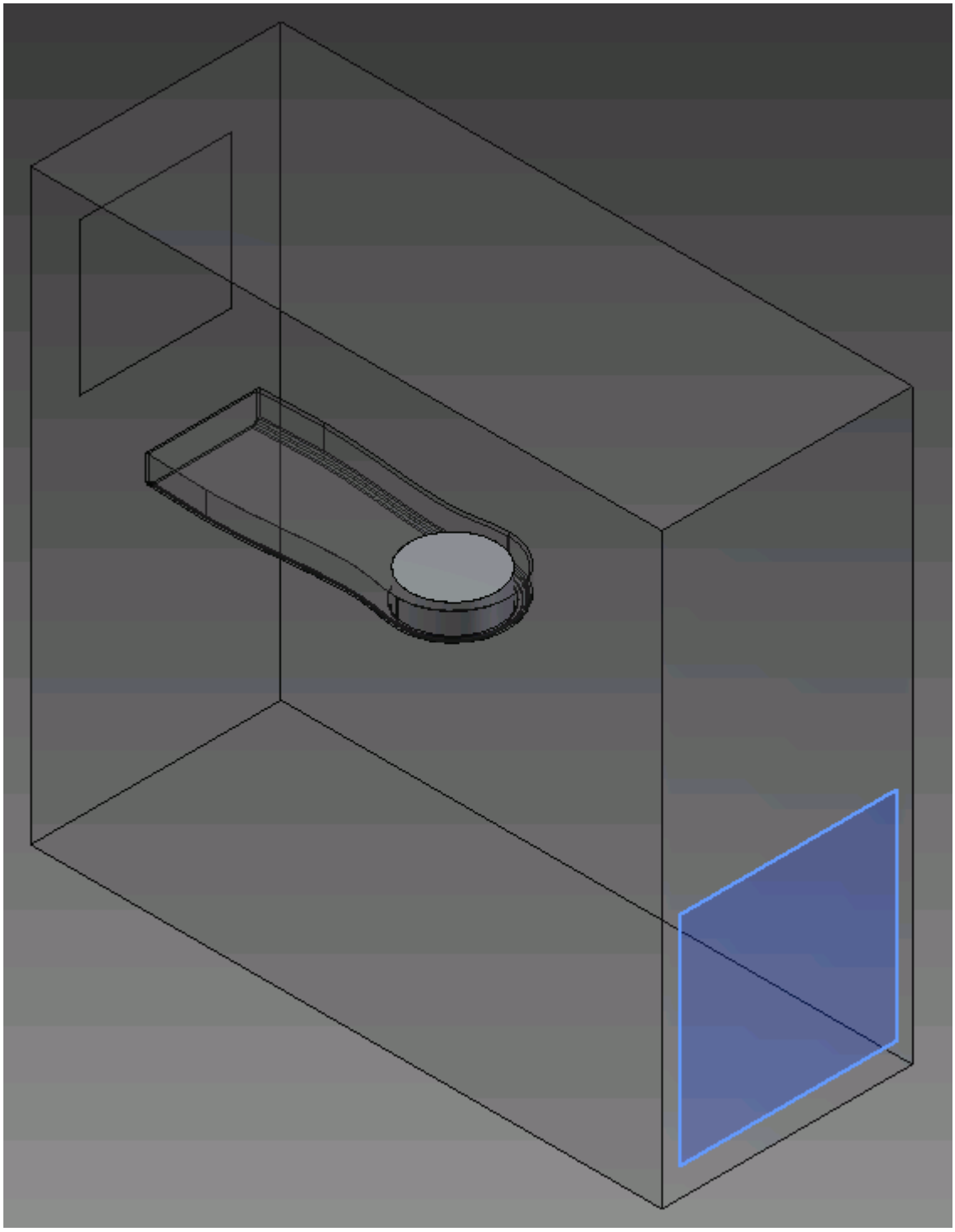
Create the inlet boundary first:

1. Right-click the **Regions** > **Region** > **Boundaries** node and select **New**.



The *Boundary Settings* dialog appears.

2. Select the face that is highlighted below.



These selections are reported in the *Face Selection* box in the *Boundary Settings* dialog.

3. Set the following settings:

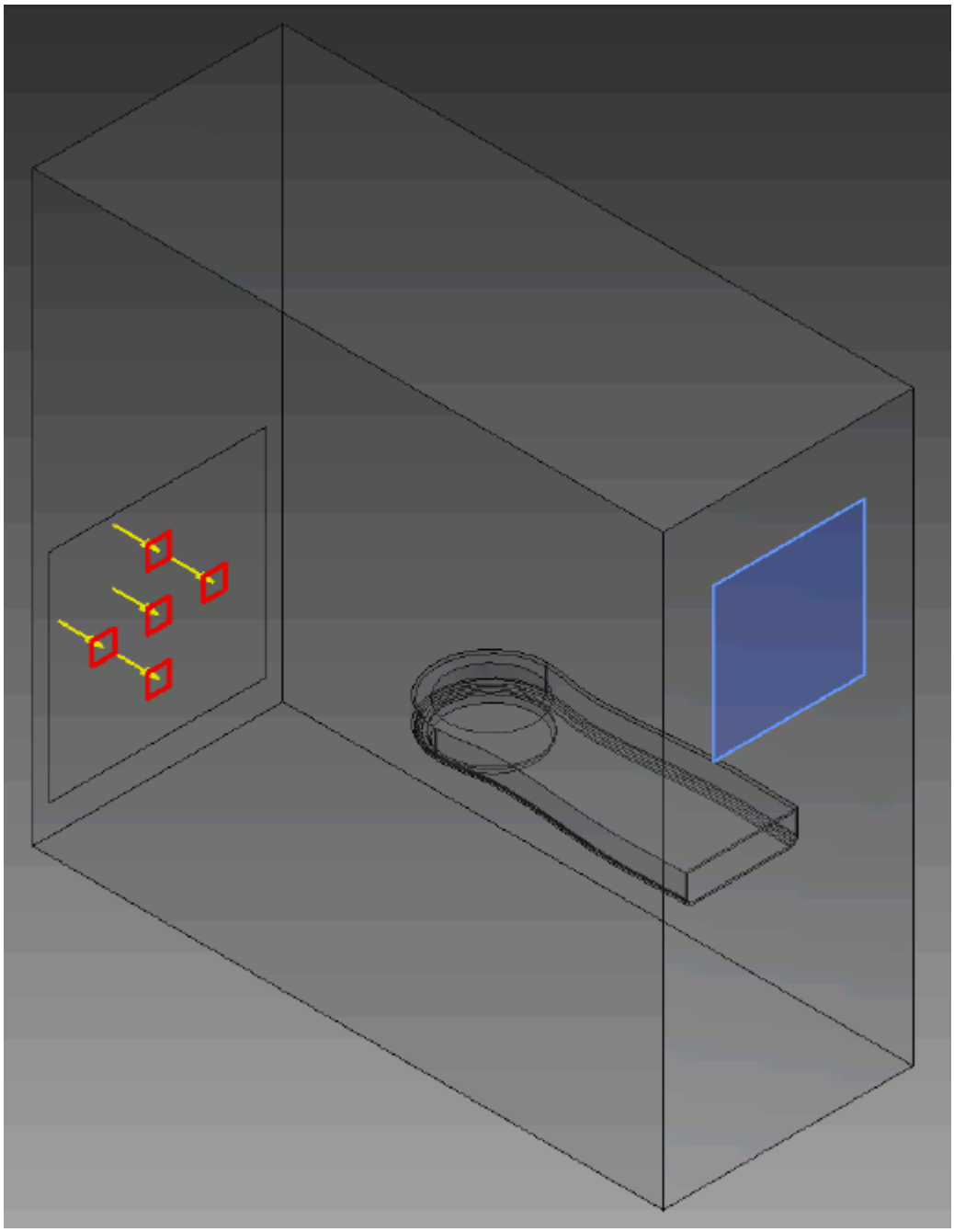
Parameter	Value
Type	**Velocity Inlet**
Name	`Case Inlet`

Parameter	Value
(*Normal Velocity*)	`0.1 m/s`

4. Click **OK**.

Create the case outlet boundary:

5. Using the same technique, create another boundary.
6. Select the face that is highlighted below.



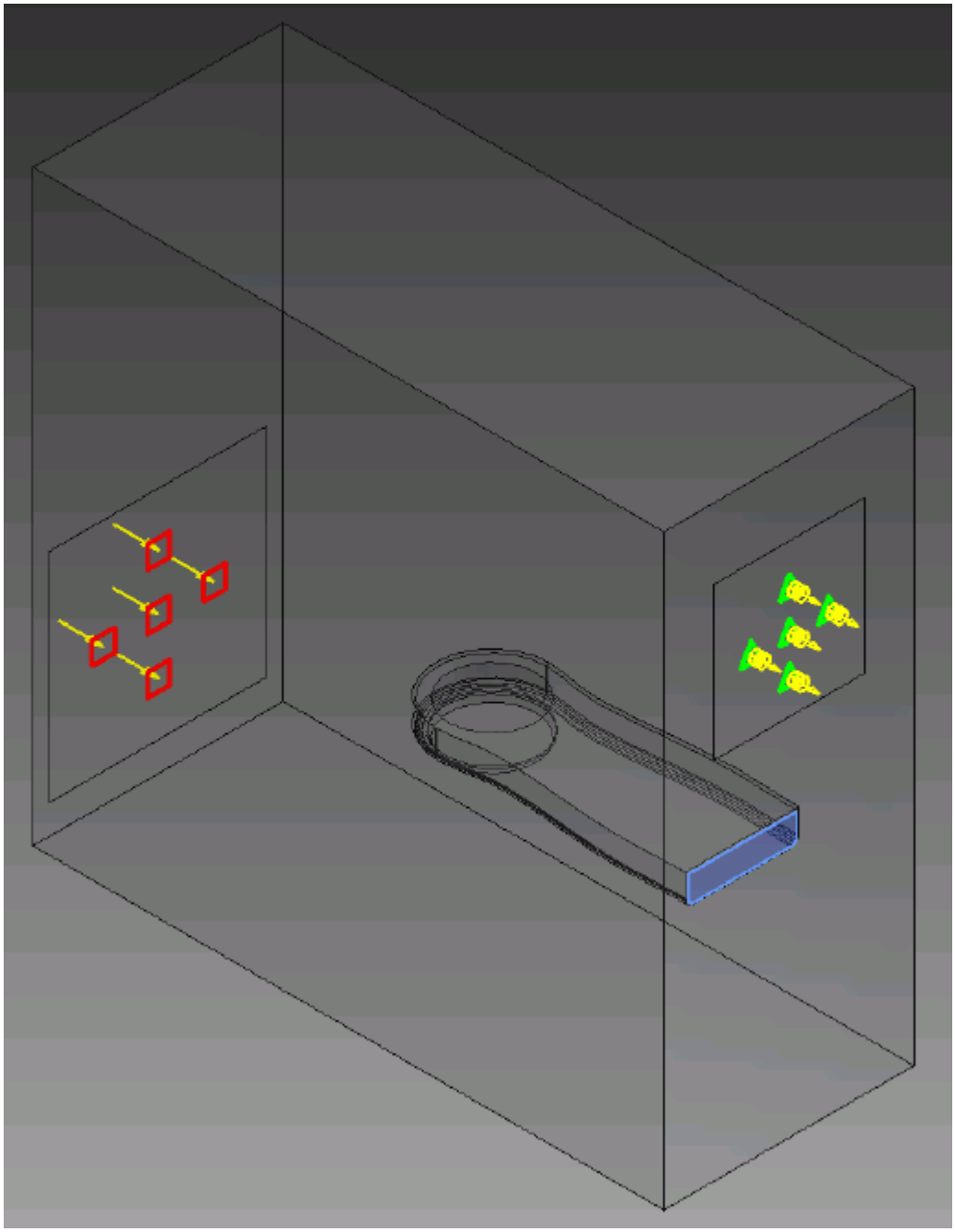
7. Set the following parameters:

Parameter	Value
Type	**Pressure Outlet**
Name	`Case Outlet`

8. Click **OK**.

Create the GPU outlet boundary:

9. Create another boundary and select the following face.



10. Set the following parameters:

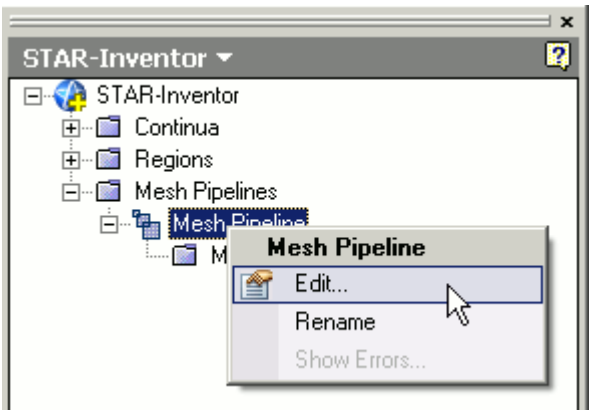
Parameter	Value
Type	**Pressure Outlet**
Name	`GPU Outlet`

11. Click **OK**.
12. Save your progress.

Setting the Mesh Pipeline

To obtain more accurate results, modify the mesh pipeline:

1. Right-click the **Mesh Pipelines** > **Mesh Pipeline** node in the STAR-Inventor tree and select **Edit**.



The *Mesh Pipeline Settings* dialog appears.

2. Set the following parameters:

Parameter	Value
Base Size	
Relative to bodies	**Deactivated**
(*Base Size*)	`0.016 m`
Prism Layer	
(*Prism Layer Thickness*)	`20 %`

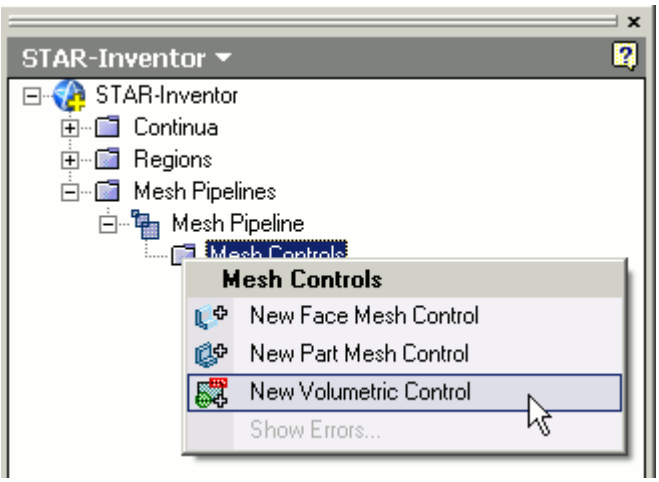
3. Click **OK**.

Creating the Volumetric Mesh Controls

Use volumetric mesh control regions to capture the finer details of the geometry model. Create three regions in total: a small region that closely surrounds the rotating region containing the fan, a region that encloses the GPU case, and a region that acts as an intermediary between the base size and the other volumetric control regions.

Create the volumetric mesh control for the rotating region:

1. Right-click the **Mesh Pipelines** > **Mesh Pipeline** > **Mesh Controls** node in the STAR-Inventor tree and select **New Volumetric Control**.



   The *Volumetric Control* dialog appears.

2. Set *Name* to `Fan Box`.
3. Activate the *Use Absolute Value* option.
4. Set *Custom Surface Mesh Size* to `0.002 m`.
5. Enter the following coordinates for Point 1:

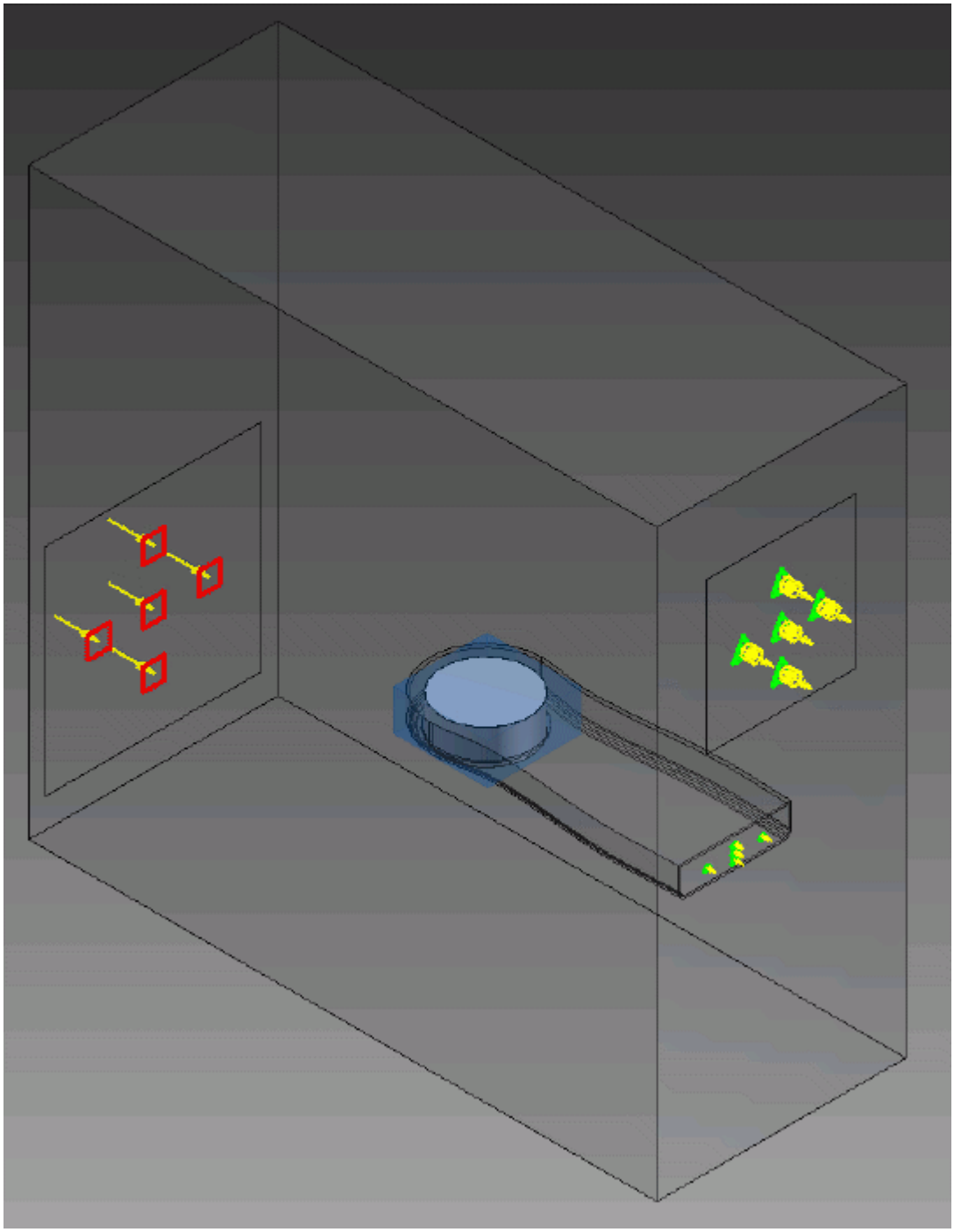
Coordinate	Value
X1	`-166.00`
Y1	`4.00`
Z1	`-61.00`

6. Enter the following coordinates for Point 2:

Coordinate	Value
X2	`-79.00`
Y2	`-34.00`
Z2	`25.00`

7. Click **Create**.

   The volumetric control region is shown below.



8. Click **OK**.

Create the volumetric mesh control for the GPU region:

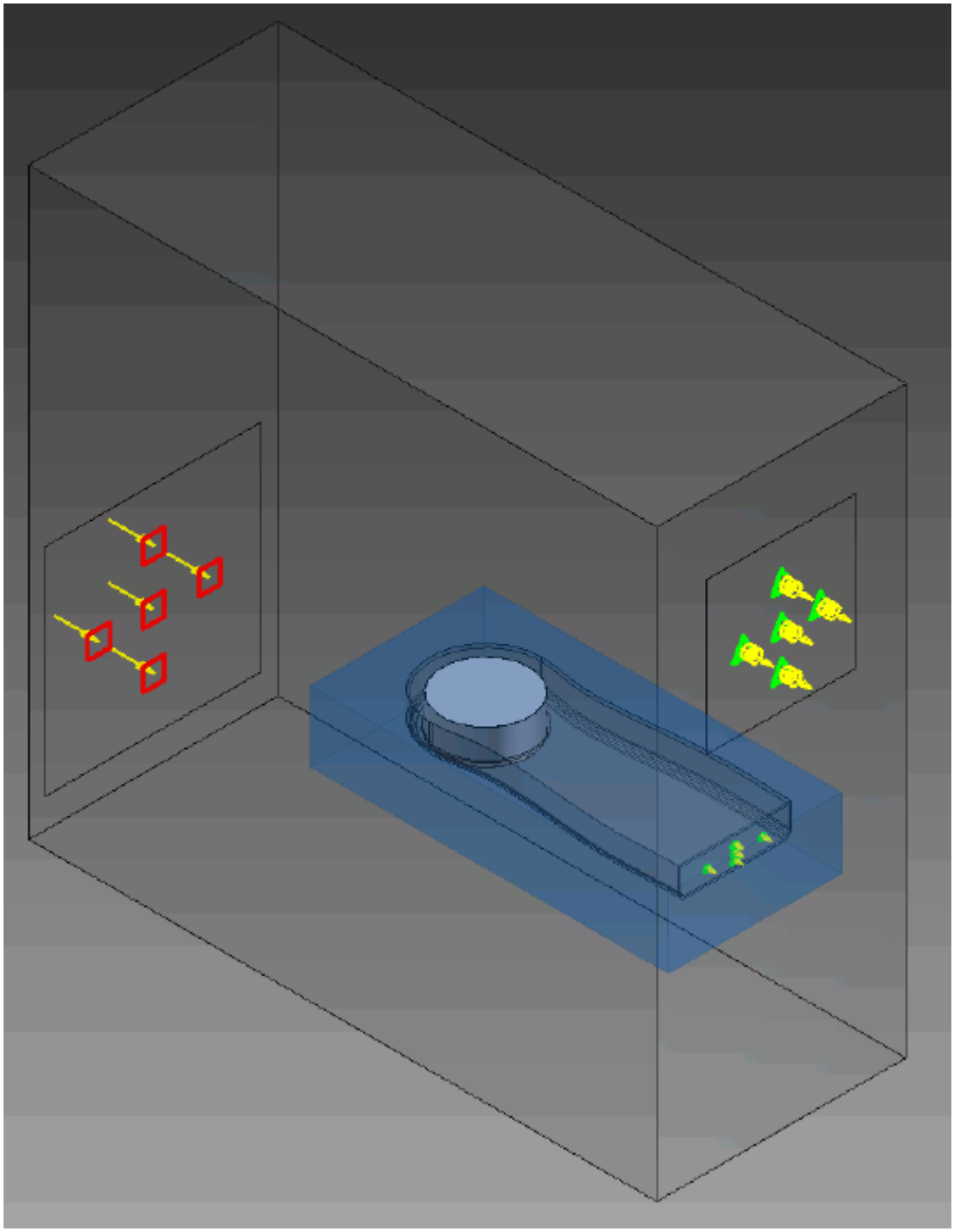
9. Using the same technique, create another volumetric mesh control with the following settings:

Parameter	Value
Name	`GPU Box`
Custom Surface Mesh Size	

Parameter	Value
Use Absolute Value	**Activated**
Custom Surface Mesh Size	`0.005 m`
Shape	
X1	`-383.00`
Y1	`16.00`
Z1	`-111.00`
X2	`-52.00`
Y2	`-49.00`
Z2	`49.00`

10. Click **Create**.

    The volumetric control region is shown below.



11. Click **OK**.

Create the volumetric mesh control for the PC tower case:

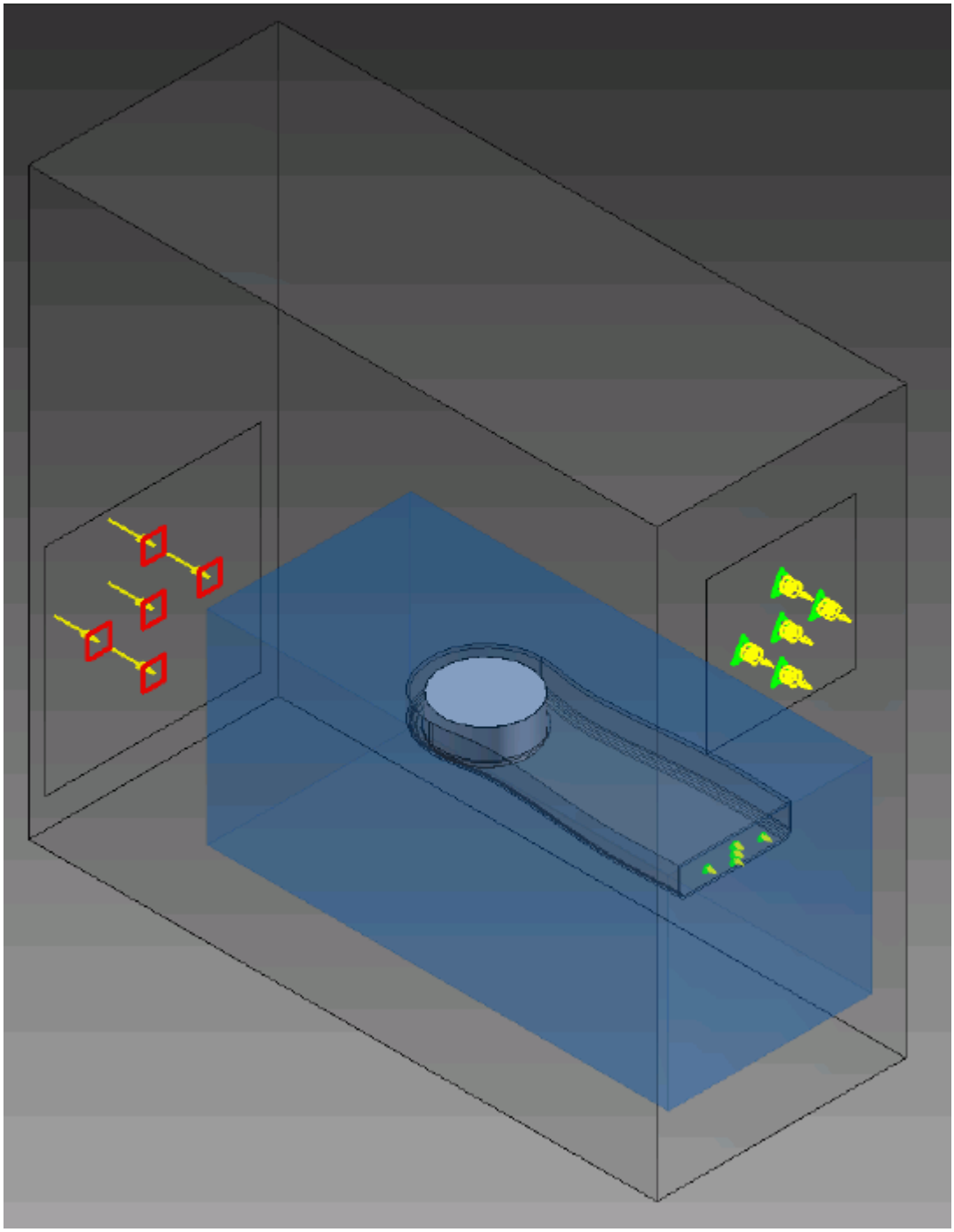
12. Using the same technique, create another volumetric mesh control with the following settings:

Parameter	Value
Name	`Case Box`
Custom Surface Mesh Size	

Parameter	Value
Use Absolute Value	**Activated**
Custom Surface Mesh Size	`0.009 m`
Shape	
X1	`-383.00`
Y1	`30.00`
Z1	`-111.00`
X2	`43.00`
Y2	`-159.00`
Z2	`77.00`

13. Click **Create**.

    The volumetric control region is shown below.

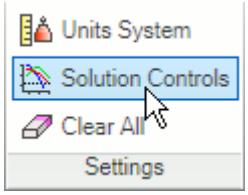


14. Click **OK**.

Modifying Solution Controls

Modify the maximum number of iterations and the solution accuracy:

1. Click **Solution Controls** on the STAR-Inventor ribbon.



The *Solution Controls* dialog appears.

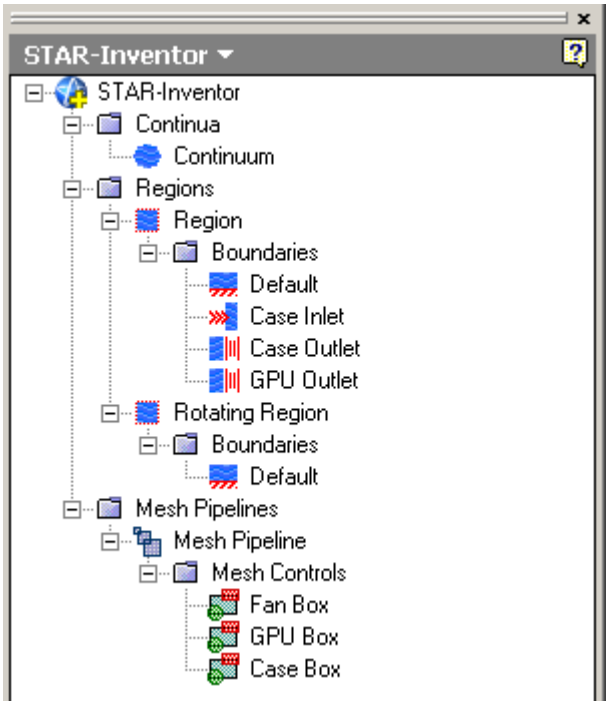
2. Set the following parameters:

Parameter	Value
(*Number of Iterations*)	300
(*Convergence Level*)	0.01
(*Momentum Under-Relaxation Factor*)	0.85

3. Click **OK**.
4. Save your progress.

Reviewing the Simulation Set Up

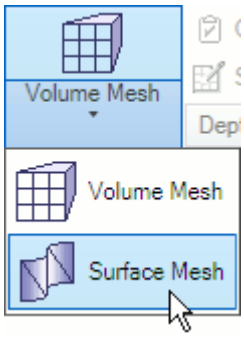
The final STAR-Inventor object tree is shown below.



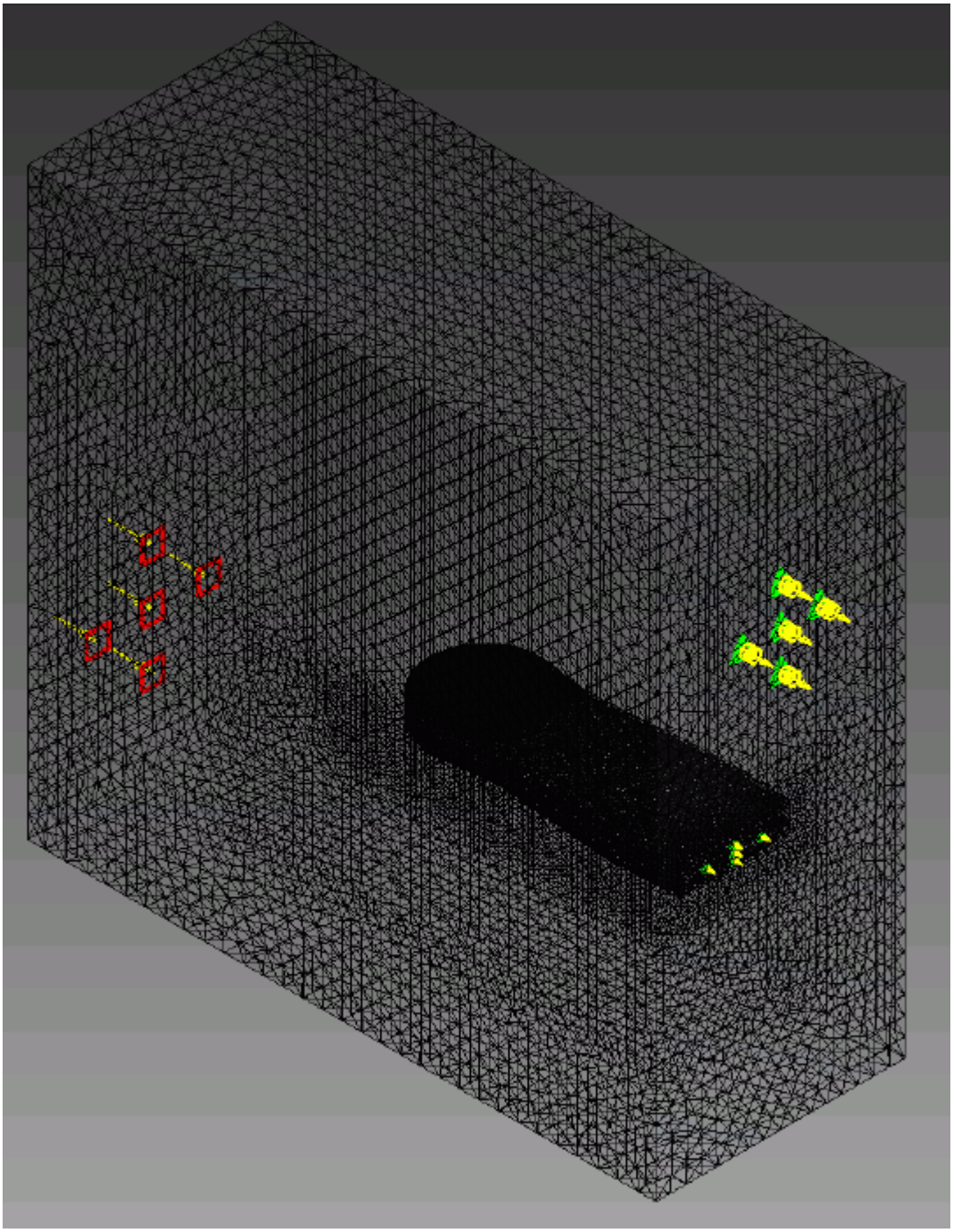
Generating The Surface Mesh

Generate the surface mesh:

1. From the STAR-Inventor ribbon, click ▾ next to the **Volume Mesh** icon and select **Surface Mesh**.



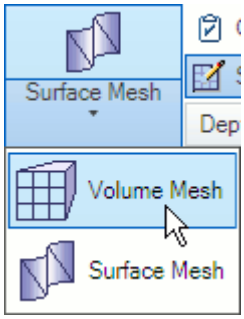
The surface mesh is shown below.



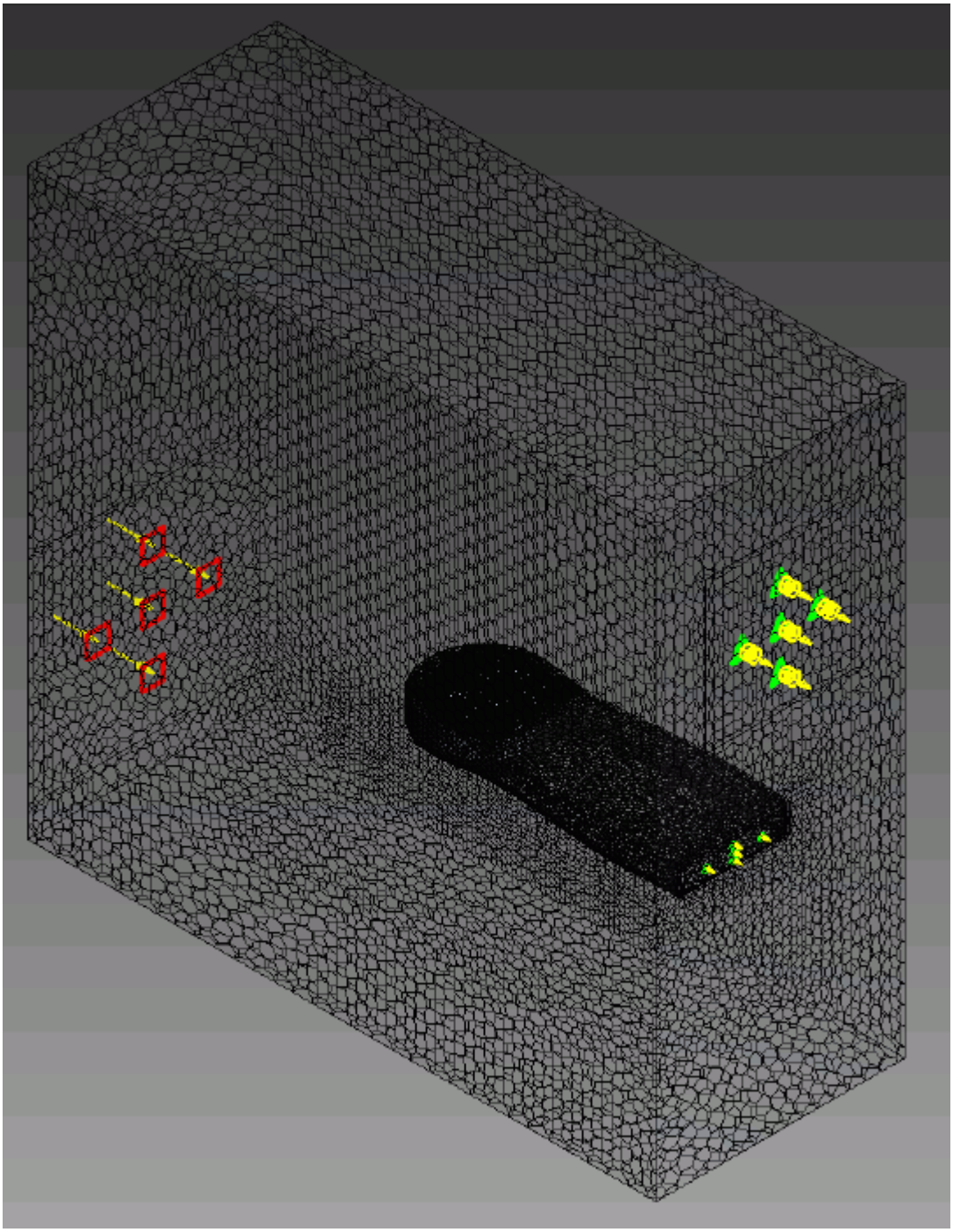
Generating the Volume Mesh

Generate the volume mesh:

1. From the STAR-Inventor ribbon, click ▾ next to the **Surface Mesh** icon and select **Volume Mesh**.



The volume mesh is shown below.

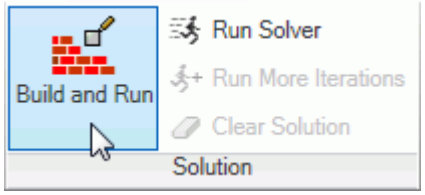


Building and Running

The case is now ready to run.

To run the simulation:

1. From the ribbon, click **Build and Run**.



The *Residuals* plot appears in the STAR-Inventor *Output* window. You can use this plot to monitor the convergence of the solution.

When the solver is done, STAR-CCM+ launches.

Post-Processing Results

You perform all post-processing in STAR-CCM+. By default, a scalar scene, *Scalar Scene 1*, automatically appears in the *Graphics* window when the solver finishes. This scene displays the absolute total pressure.

The results that are contained in the following sections are representative. You may see some variation in the results that you obtain, depending on what platform and package you are using.

1. In STAR-CCM+, click **File** > **Save** to save the simulation.

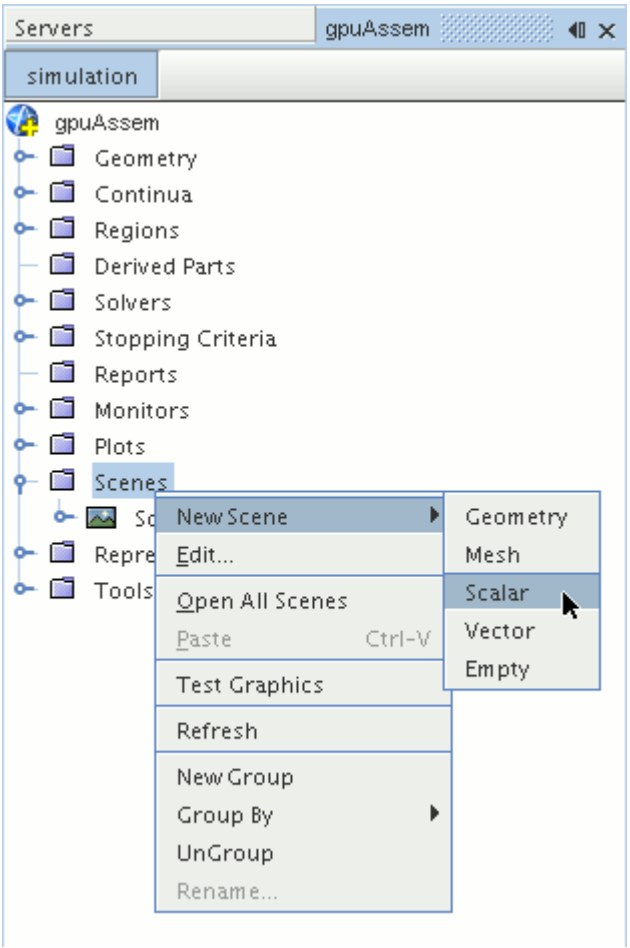
Contents:

Creating a Scalar Scene
Creating A Streamlines Scene
Creating a Vector Scene
Creating a Streamlines Scene
Creating A Scalar Scene
Visualizing the Wing Skin Temperature
Creating a Streamlines Scene
Creating A Scalar Scene
Creating a Streamlines Scene
Creating a Scalar Scene
Modifying The Scalar Scene
Creating a Streamlines Scene

Creating A Scalar Scene

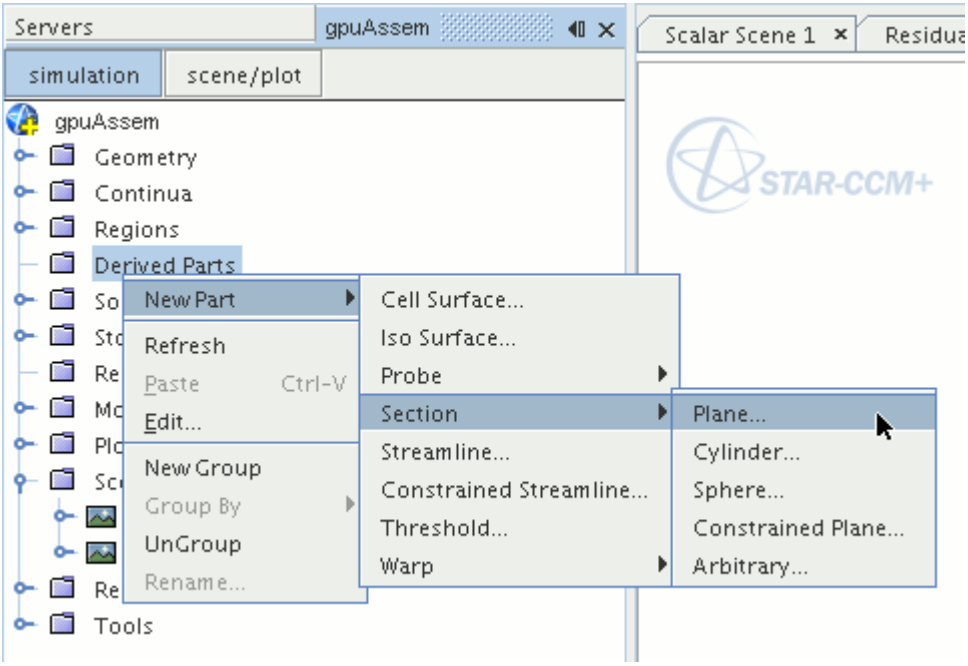
Create a scalar scene containing a cross-section of the GPU and plot velocity magnitude on this plane.

1. Right-click the **Scenes** node in the explorer pane and select **New Scene** > **Scalar**.



A new scalar scene, *Scalar Scene 2*, is created and selected in the *Graphics* window.

2. Right-click the **Derived Parts** node and select **New Part** > **Section** > **Plane...**.

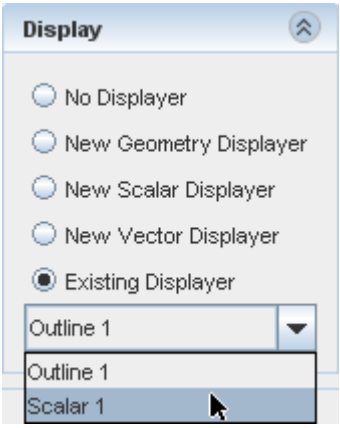


The *Create Plane Section* dialog appears.

3. In the *Plane Parameters* box, set the *origin* vector to `[0 m, -0.012 m, 0 m]` and the *normal* vector to `[0 m, 1 m, 0 m]`.

   Note that the units are in meters.

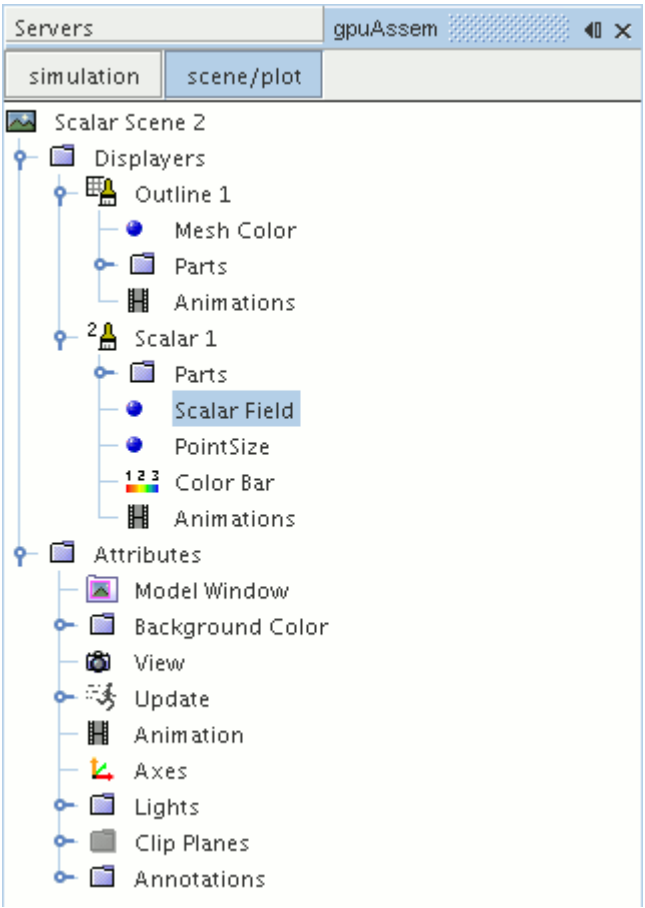
4. In the *Display* group box, select **Existing Displayer** and in the drop-down menu select **Scalar 1**.



5. Click **Create** and then **Close.**

Display velocity magnitude on the section plane:

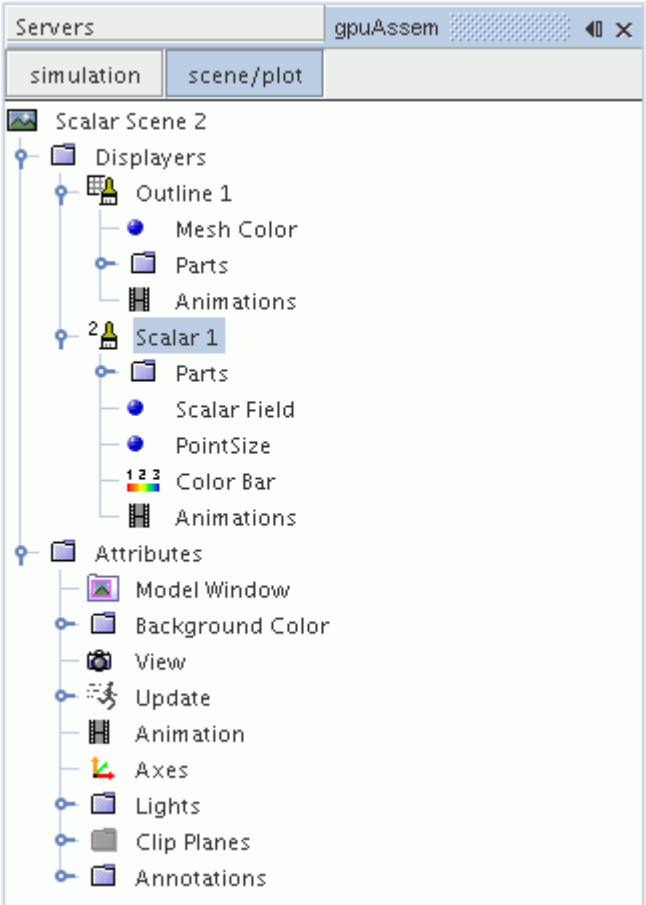
6. Click **scene/plot**.
7. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Scalar Field**.



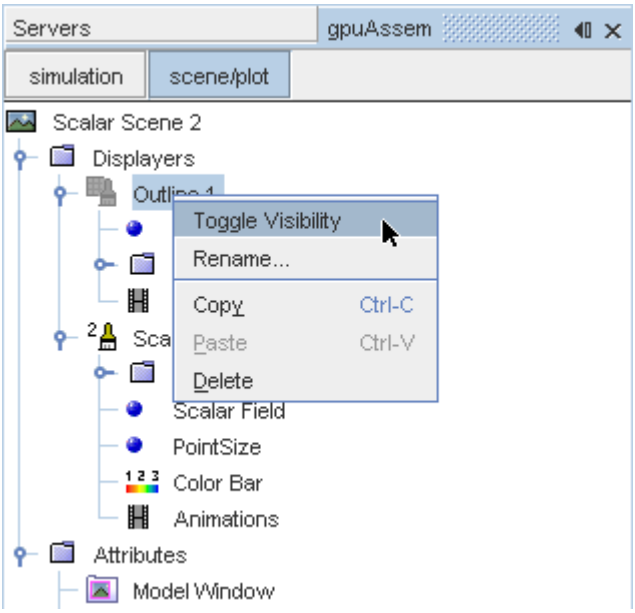
8. Set *Function* to **Velocity** > **Lab Reference Frame** > **Magnitude**.

To have a smooth transition between the different velocity values, set the contour style to smooth:

9. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1**.

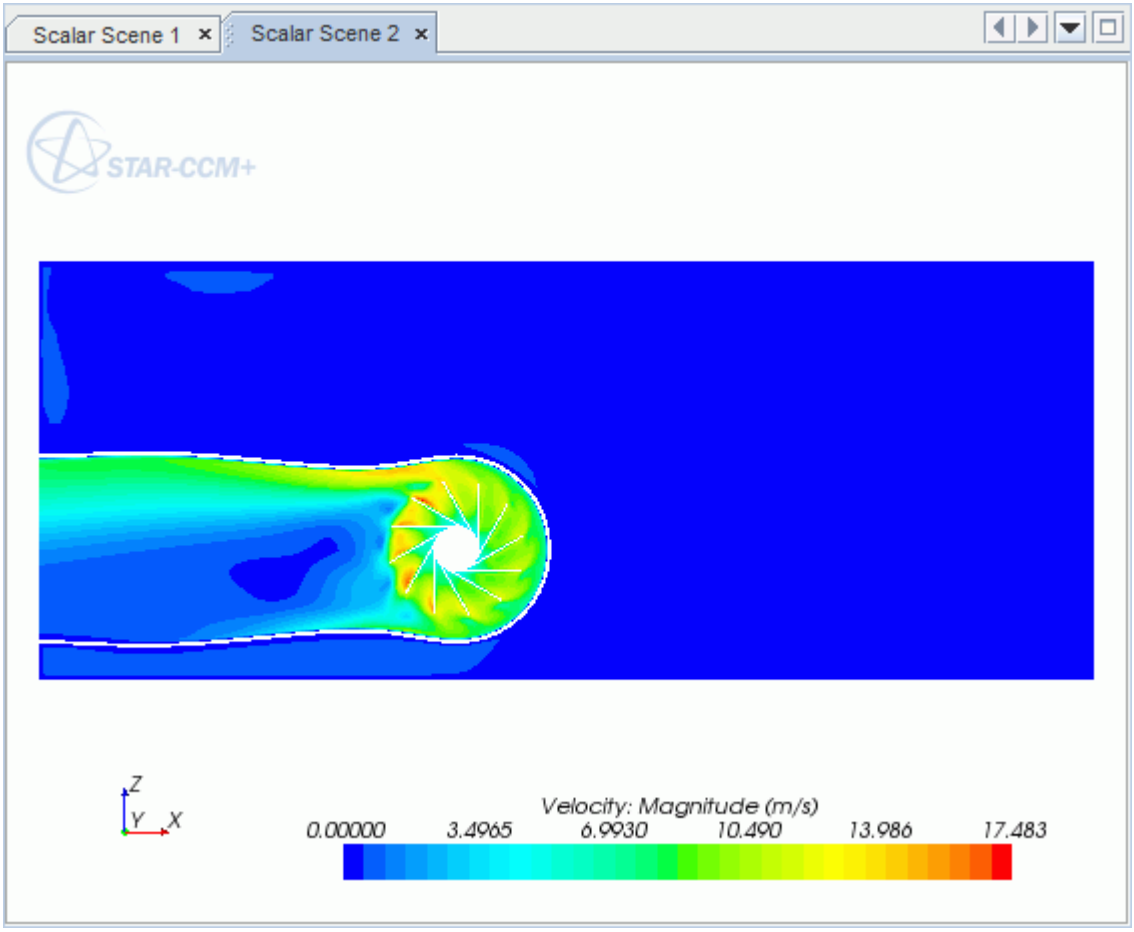


10. Set *Contour Style* to **Smooth Filled**.
11. To hide all outlines, right-click the **Scalar Scene 2** > **Displayers** > **Outline 1** node and select **Toggle Visibility**.



12. Click the *Scalar Scene 2* tab to activate the scene, and press the <S> key to change to the top view of this geometry.
13. Click **simulation**.
14. Click (**Save**).

The finished scene is shown below.

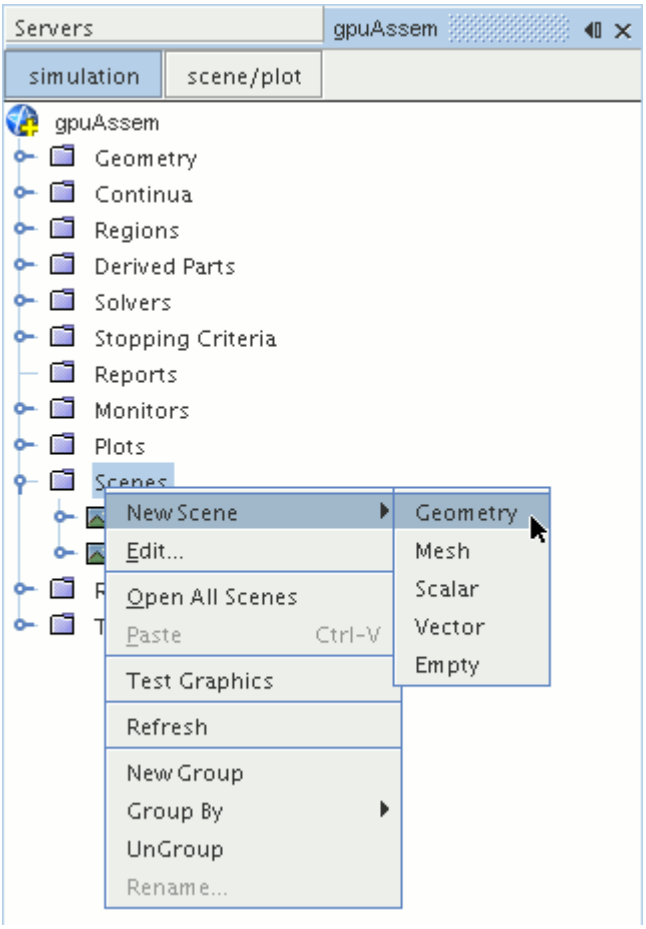


Creating a Streamlines Scene

Create streamlines to get a better understanding of the air flow inside the PC tower case and graphics card.

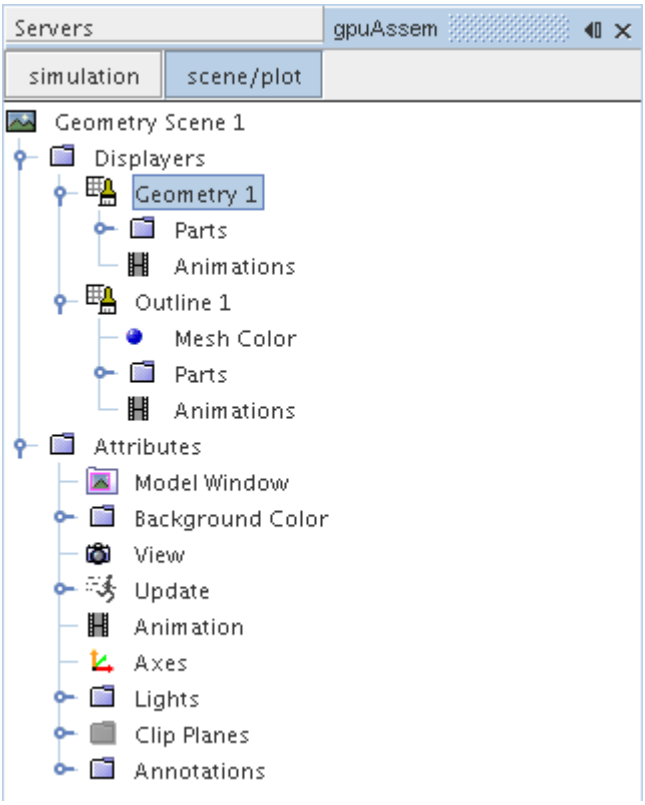
First, create a geometry scene:

1. Right-click the **Scenes** node in the object tree and select **New Scene** > **Geometry**.



A new geometry scene is created and selected in the *Graphics* window.

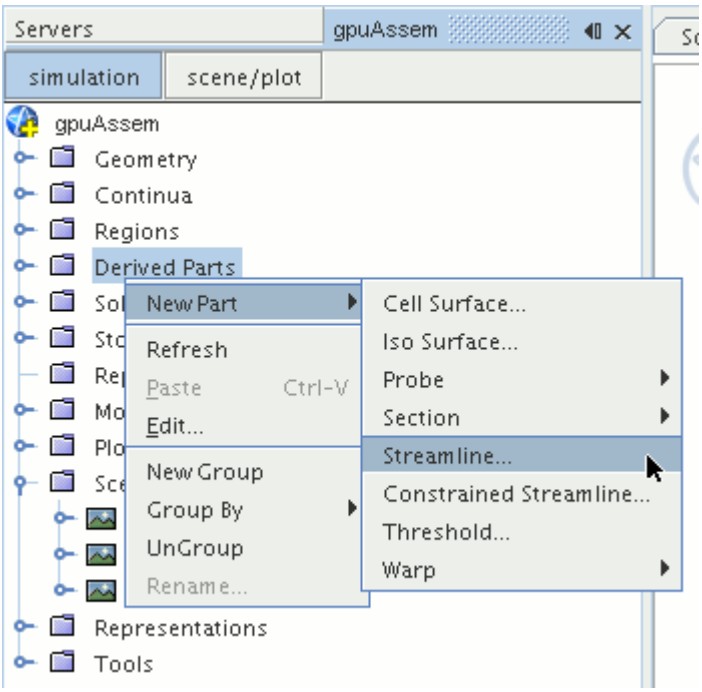
2. Click **scene/plot**.
3. Navigate to **Geometry Scene 1** > **Displayers** > **Geometry 1** node.



4. Set *Opacity* to `0.1`

Create the streamlines:

5. Click **simulation**.
6. Right-click the **Derived Parts** node and select **New Part** > **Streamline...**.



The *Create Streamline* dialog appears.

7. Make sure that **Part Seed** is selected in the *Seed Mode* section.
8. Set *Seed Parts* to **Regions** > **Region** > **Boundaries** > **Case Inlet**.
9. Set *Part U-Resolution* to `10`.
10. Set *Part V-Resolution* to `10`.
11. Make sure that **New Streamline Displayer** is selected in the *Display* section.
12. Click **Create** followed by **Close**.

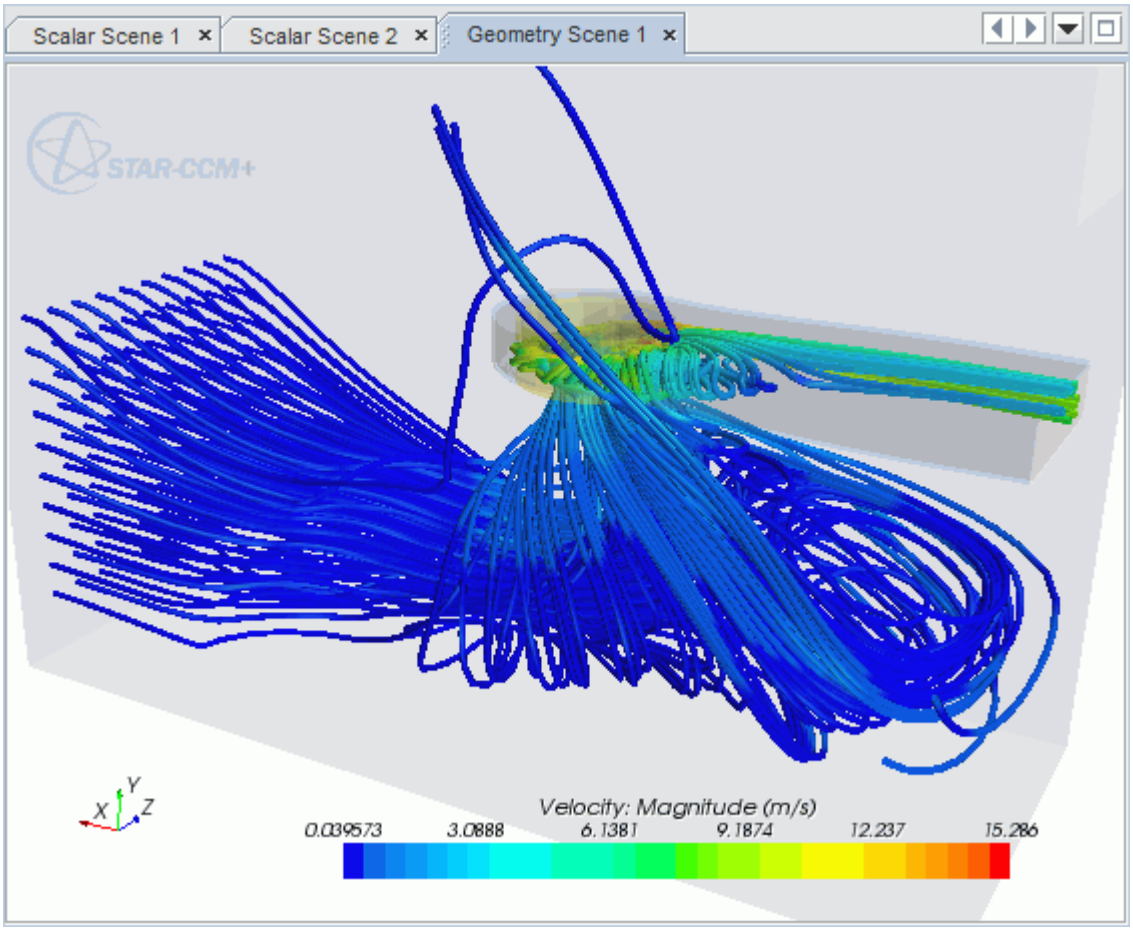
Adjust the appearance of streamlines to display tubes instead of the default lines:

13. Click **scene/plot**.
14. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** and set *Mode* to **Tubes** and *Width* to `0.0020`.
15. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** > **Scalar Field** and set *Function* to **Velocity** > **Lab Reference Frame** > **Magnitude**.

Hide all the outlines in the scene:

16. Right-click the **Geometry Scene 1** > **Displayers** > **Outline 1** node and select **Toggle Visibility**.
17. Reposition the geometry to get a better view of the flow inside the case.
18. Save the simulation.

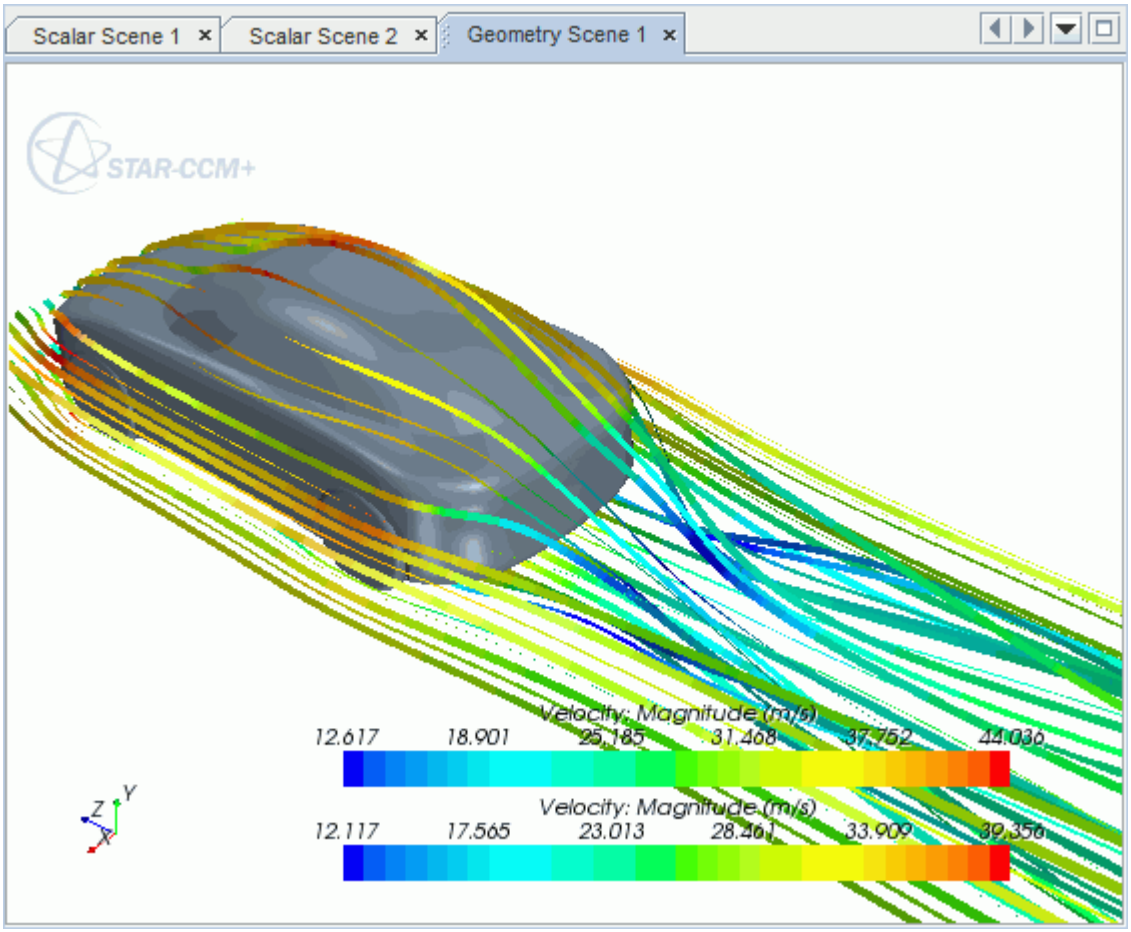
The finished scene is shown below.


Scalar Scene 1
Scalar Scene 2
Geometry Scene 1
STAR-CCM+
Velocity: Magnitude (m/s)
0.039573
3.0888
6.1381
9.1874
12.237
15.286
X
Y
Z

External Flow Over a Car

This tutorial takes you through the steps that are required to run a CFD calculation in the STAR-Inventor environment.

For this tutorial, the model is a car in a fluid domain.



Contents:

Opening the CAD Model
Selecting Fluid Physics Models
Defining Boundary Conditions
Setting the Mesh Pipeline
Creating the Volumetric Mesh Controls
Creating the Face Mesh Control
Modifying Solution Controls
Reviewing the Simulation Set Up
Generating the Surface Mesh

Generating the Volume Mesh
Building and Running
Post-Processing Results

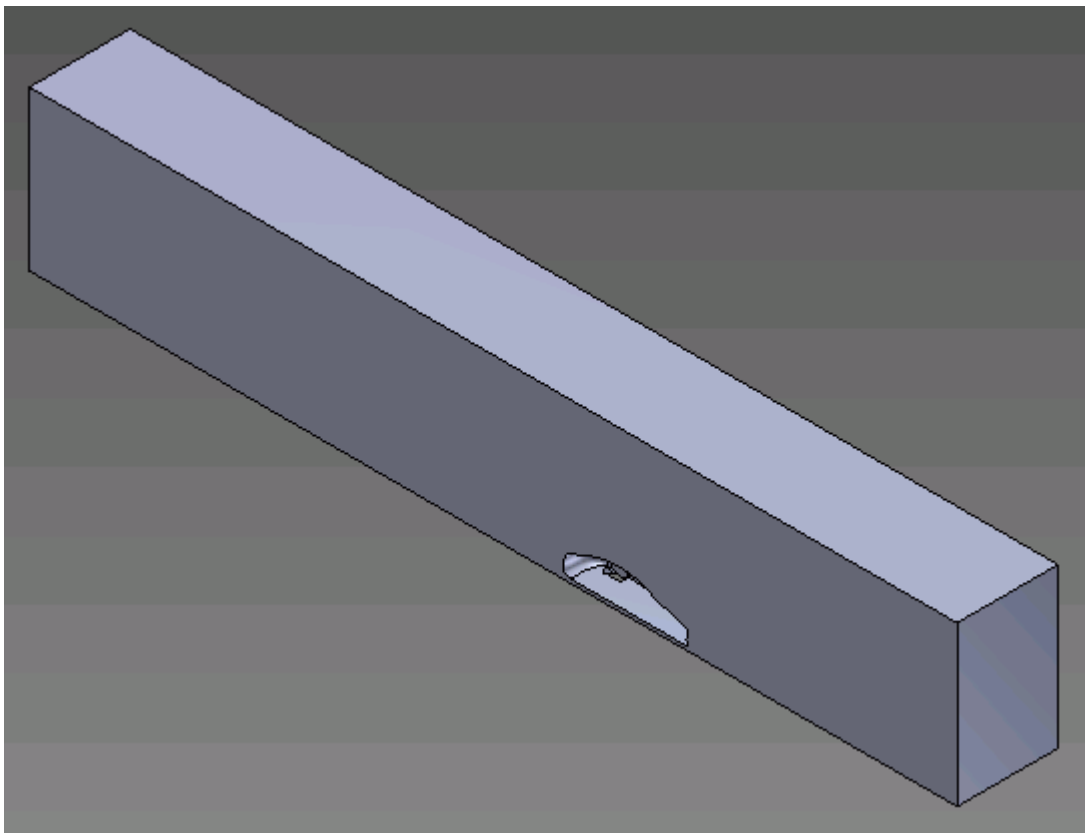
Opening the CAD Model

A directory containing the CAD geometry is provided with the STAR-CCM+ distribution.

This directory is located in `[STAR-CCM+_INSTALL_DIR]\doc\startutorialsdata\starinventor\data`.

1. Copy directory `data` to your own disk space and open the part file `car.ipt` in STAR-Inventor.

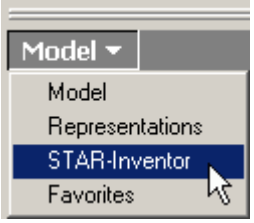
   The imported geometry is shown below.



The STAR-Inventor ribbon appears at the top of the screen.The STAR-Inventor object tree is located on the left of the screen.

To switch to this object tree:

2. Click **Model** and select **STAR-Inventor**.

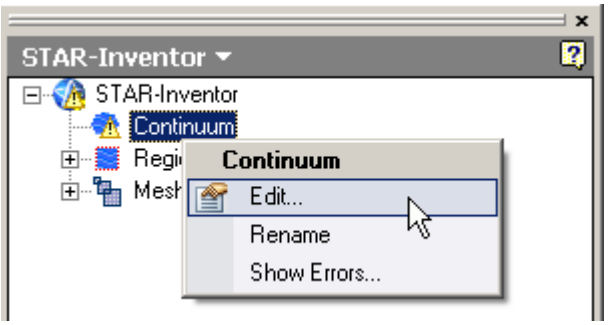


If these items are not visible, refer to Launching STAR-Inventor Manually.

Selecting Fluid Physics Models

The first step in setting up this tutorial is to define the physics of the problem. For this case, the flow is turbulent and isothermal.

1. Right-click the **Continuum** node in the STAR-Inventor tree and select **Edit**.



   The *Continuum Settings* dialog appears.

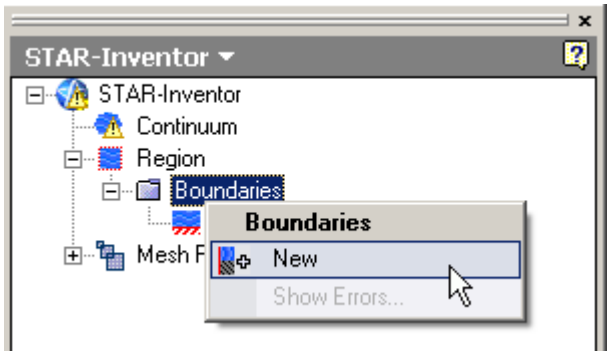
2. Deactivate the **Temperature** option in the *Optional Physics Models*.
3. Click **OK**.

Defining Boundary Conditions

Create the inlet and outlet boundary condition. Use a moving wall to model the relative movement of the road.

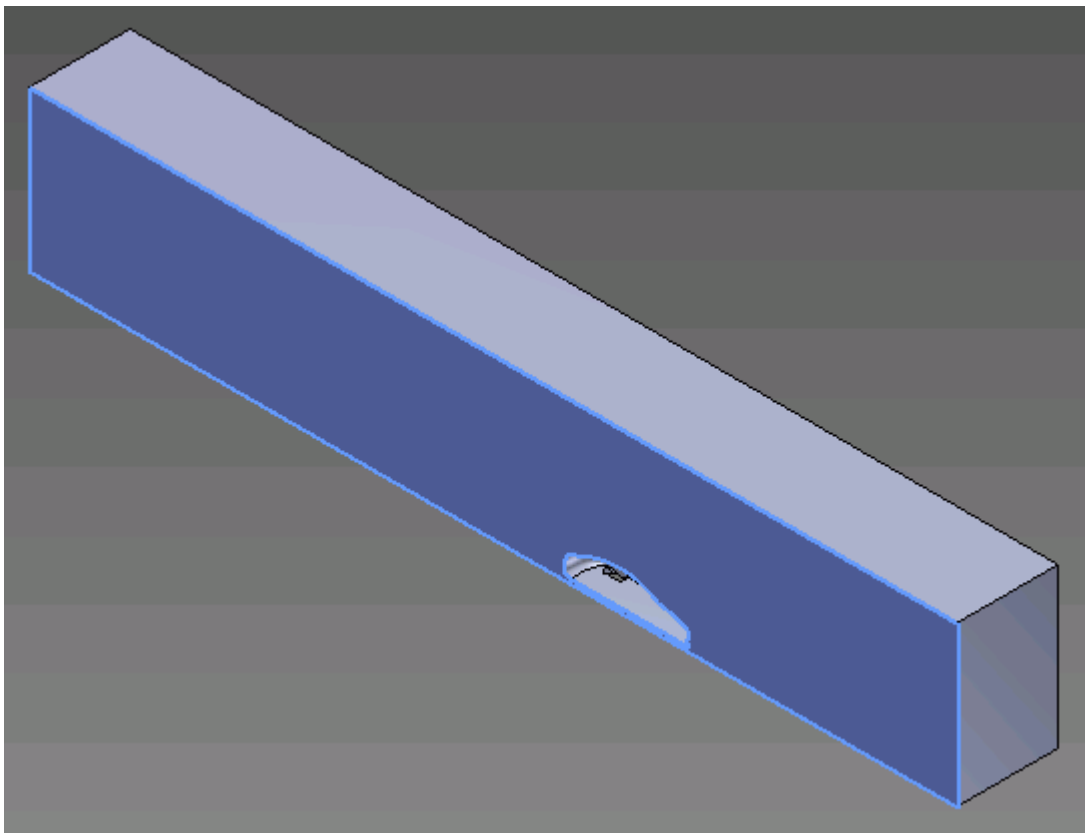
Create the symmetry plane first:

1. Right-click the **Region** > **Boundaries** node and select **New**.



The *Boundary Settings* dialog appears.

2. On the CAD model, select the face that is highlighted below.



These selections are reported in the *Face Selection* box in the *Boundary Settings* dialog.

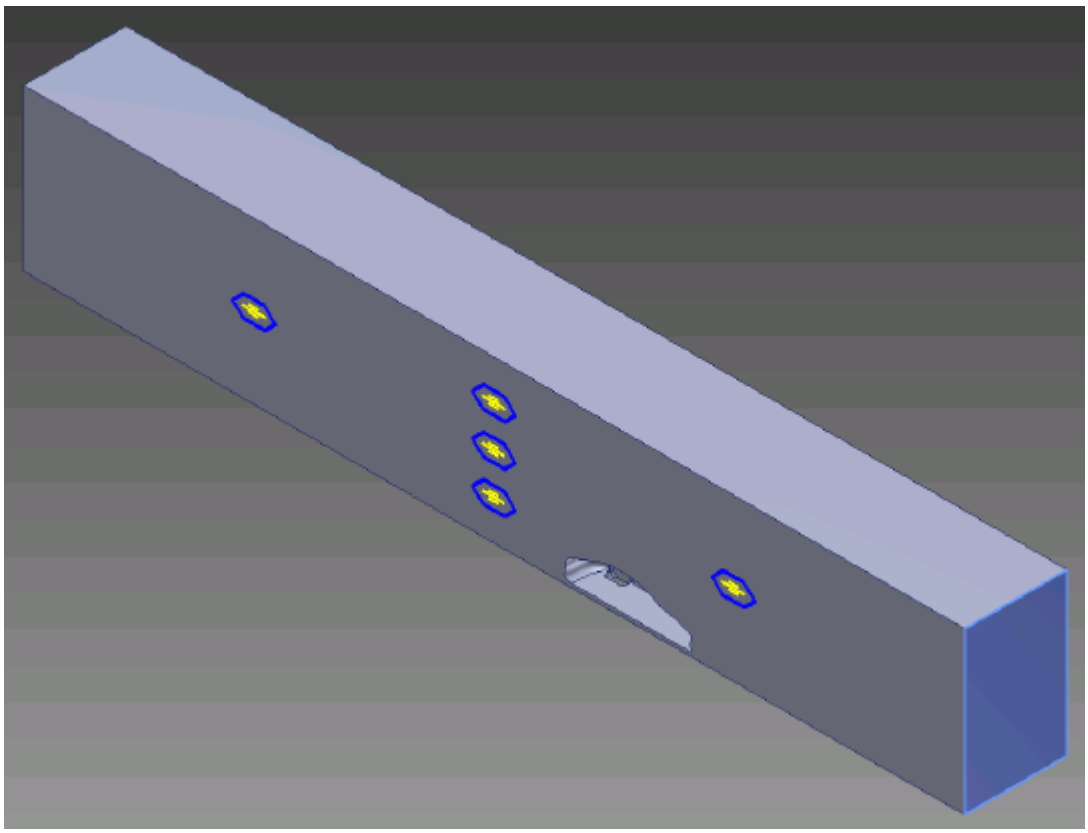
3. Set the following parameters:

Parameter	Value
Type	**Symmetry Plane**
Name	`Symmetry Plane`

4. Click **OK**.

Create the inlet boundary:

5. Create another boundary and select the following face.



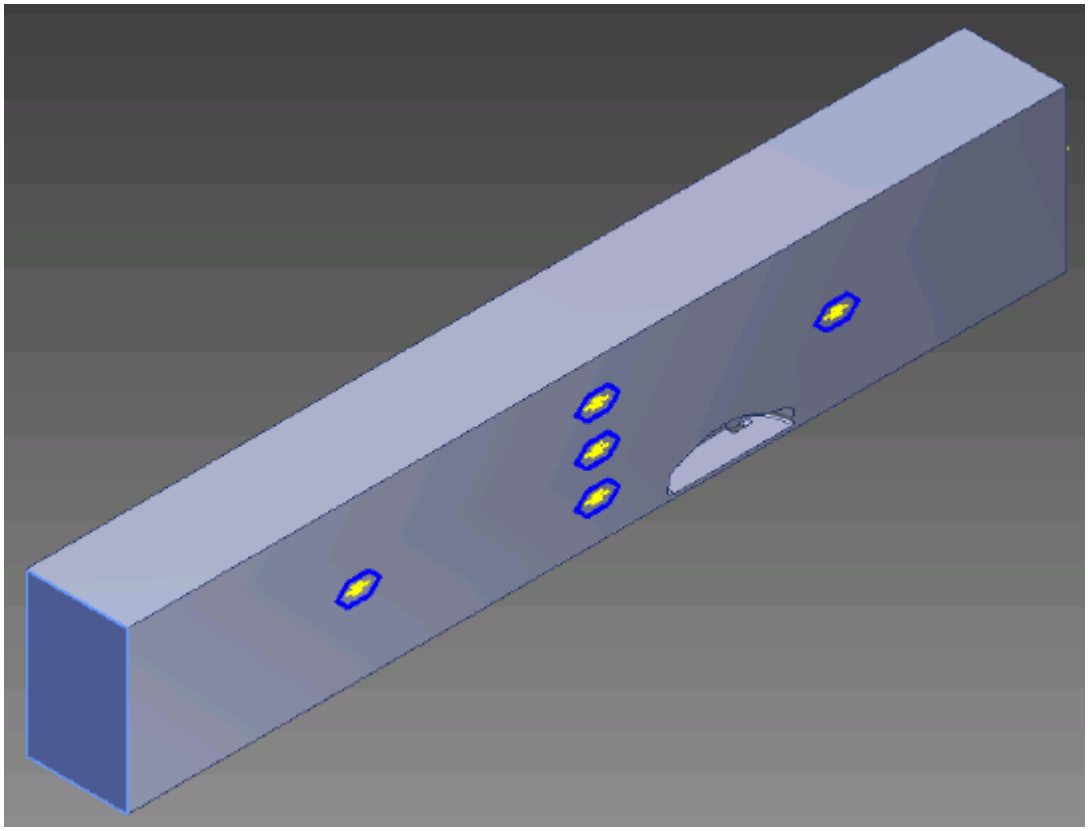
6. Set the following parameters:

Parameter	Value
Type	**Velocity Inlet**
Name	`Inlet`
(*Normal Velocity*)	`70 mph` **Note:** The units are in **mph**.

7. Click **OK**.

Create the outlet boundary:

8. Create another boundary.
9. Rotate the view to display the back face and select the face that is highlighted below.



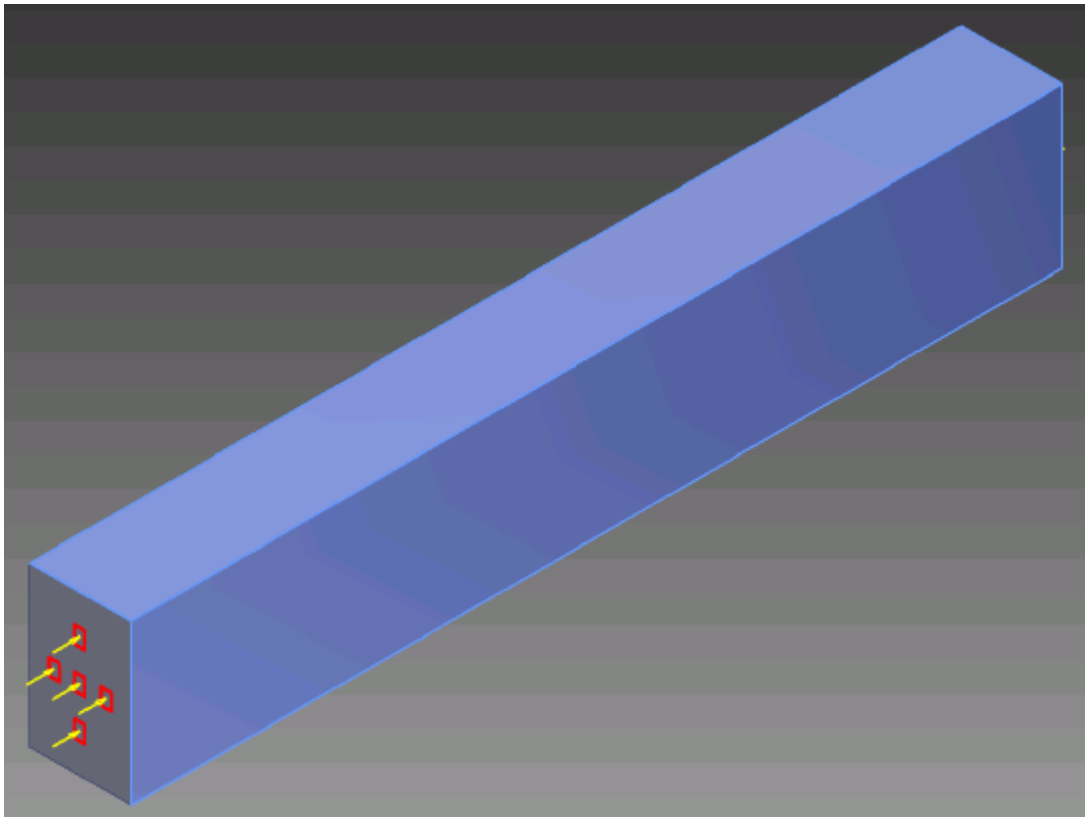
10. Set the following parameters:

Parameter	Value
Type	**Pressure Outlet**
Name	`Outlet`

11. Click **OK**.

Create the fluid domain symmetry boundaries:

12. Create another boundary.
13. Rotate the view to display the top and side walls, and select the faces that are highlighted below.



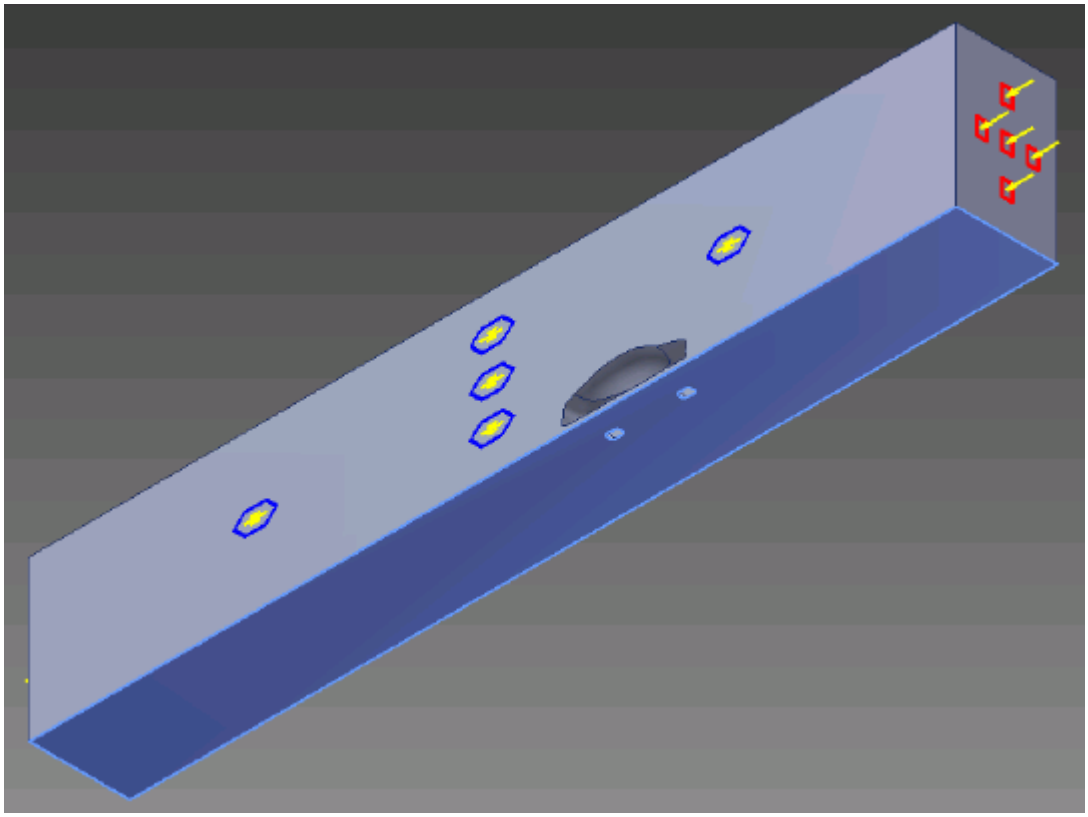
14. Set the following parameters:

Parameter	Value
Type	**Symmetry Plane**
Name	`Boundary Symmetry`

15. Click **OK**.

Create the boundary that will model the moving ground:

16. Create another boundary.
17. Rotate the view to display the bottom face, and select the face that is highlighted below.



18. Set the following parameters:

Parameter	Value
Type	**Wall**
Name	`Ground`
Wall Type	**Translating**
(*W Component of Wall Velocity Vector*)	`-70 mph` **Note:** The units are in **mph**.

19. Click **OK**.

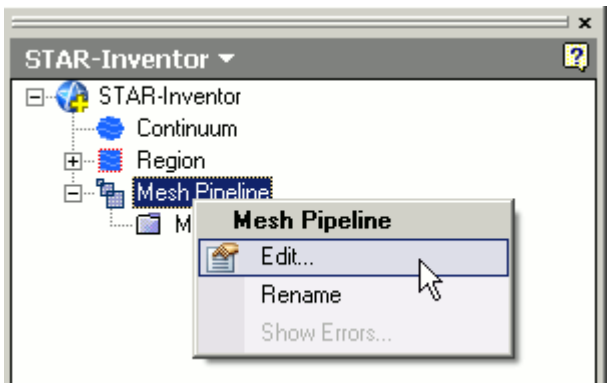
Rename the default boundary:

20. Right-click **Region** > **Boundaries** > **Default** and select **Rename**.
21. Enter `Car` as the new name.
22. You may need to click on a blank area of the workspace to accept the name change.
23. Save your progress.

Setting the Mesh Pipeline

To obtain more accurate results, modify the mesh pipeline:

1. Right-click the **Mesh Pipeline** node in the STAR-Inventor tree and select **Edit**.



The *Mesh Pipeline Settings* dialog appears.

2. Set the following parameters:

Parameter	Value
Base Size	
Relative to bodies	**Deactivated**
(*Base Size*)	`0.03 m`
Prism Layer	
User Absolute Value	**Activated**
(*Prism Layer Thickness*)	`0.03 m`

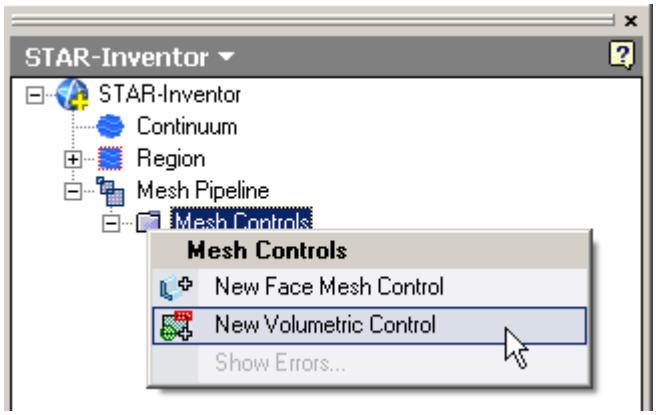
3. Click **OK**.

Creating the Volumetric Mesh Controls

Create two volumetric mesh control regions to capture the finer details of the flow: one around the car and one that extends into the wake of the car.

Create the volumetric mesh control for the car region:

1. Right-click the **Mesh Pipeline** > **Mesh Controls** node in the STAR-Inventor tree and select **New Volumetric Control**.



The *Volumetric Control* dialog appears.

2. Set *Name* to `Car Box`.
3. Activate the *Use Absolute Value* option.
4. Set *Custom Surface Mesh Size* to `0.09 m`.
5. Enter the following coordinates for Point 1:

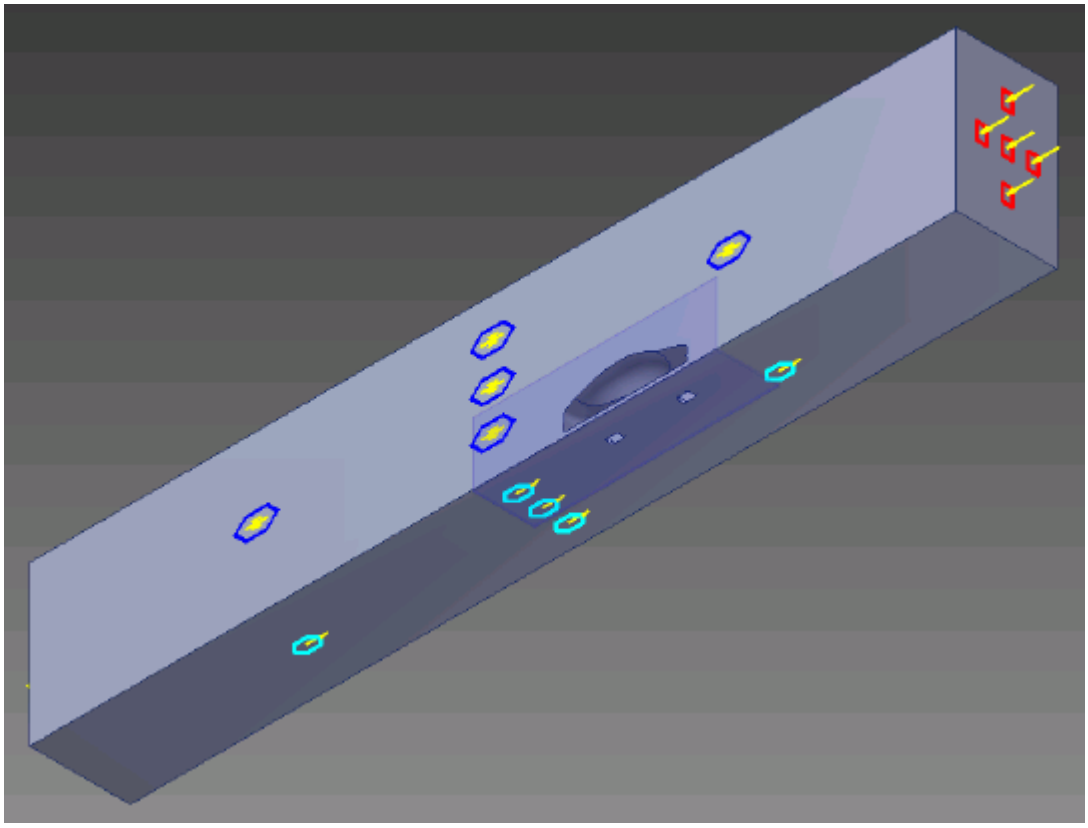
Coordinate	Value
X1	`-50.00`
Y1	`-650.00`
Z1	`3432.00`

6. Enter the following coordinates for Point 2:

Coordinate	Value
X2	`2000.00`
Y2	`1440.00`
Z2	`-4567.00`

7. Click **Create**.

   The volumetric control region is shown below.



8. Click **OK**.

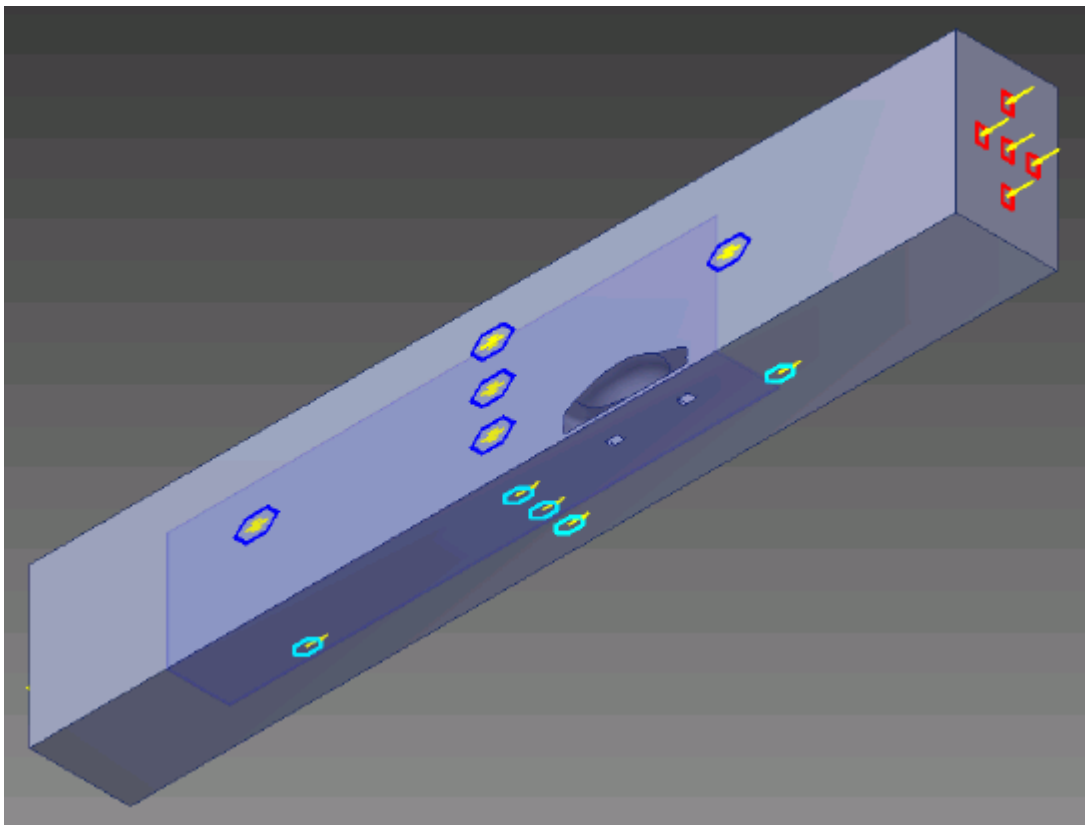
Create the volumetric mesh control for the wake region:

9. Using the same technique, create another volumetric mesh control with the following settings:

Parameter	Value
Name	`Wake Box`
Custom Surface Mesh Size	
Use Absolute Value	**Activated**
Custom Surface Mesh Size	`0.177 m`
Shape	
X1	`-50.00`
Y1	`-650.00`
Z1	`3432.00`
X2	`2000.00`
Y2	`3240.00`
Z2	`-14567.00`

10. Click **Create**.

    The volumetric control region is shown below.



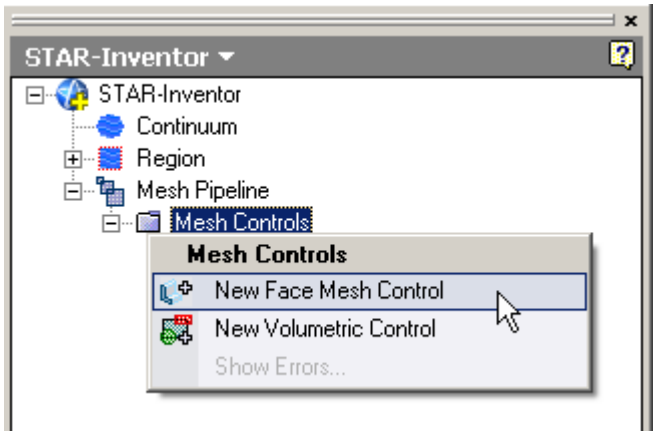
11. Click **OK**.

Creating the Face Mesh Control

Apply a face mesh control to the walls of the fluid domain to coarsen the mesh on these faces.

Create the face mesh control:

1. Right-click the **Mesh Pipeline** > **Mesh Controls** node in the STAR-Inventor tree and select **New Face Mesh Control**.



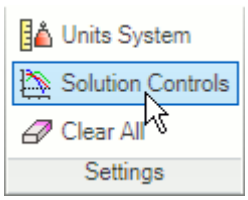
The *Face Mesh Control* dialog appears.

2. Select all six boundary faces that form the fluid domain: `Inlet`, `Outlet`, `Boundary Symmetry`, `Symmetry Plane`, and `Ground`.
3. Set *Name* to `Fluid Domain Face Control`.
4. Activate the **Custom Surface Mesh Size** option.
5. Activate the *Use Absolute Value* option.
6. Set (*Custom Mesh Size*) to `0.35 m`.
7. Click **OK**.

Modifying Solution Controls

Modify the solution controls:

1. Click **Solution Controls** on the STAR-Inventor ribbon.



The *Solution Controls* dialog appears.

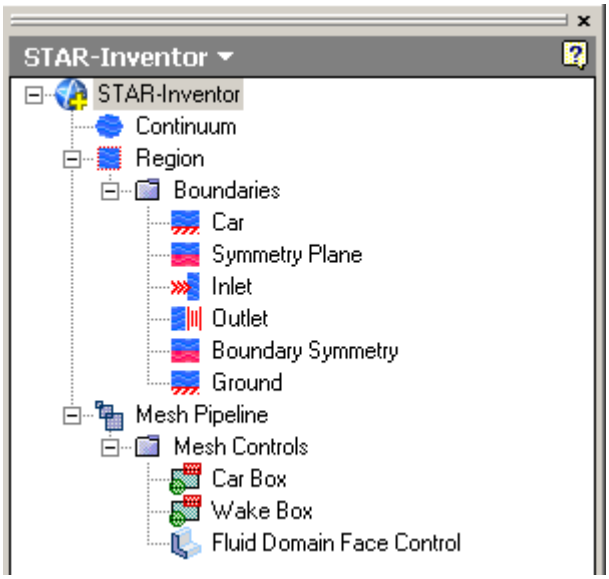
2. Set the following parameters:

Parameter	Value
(*Number of Iterations*)	`400`
(*Convergence Level*)	`0.0001`
(*Momentum Under-Relaxation Factor*)	`0.85`

3. Click **OK**.
4. Save your progress.

Reviewing the Simulation Set Up

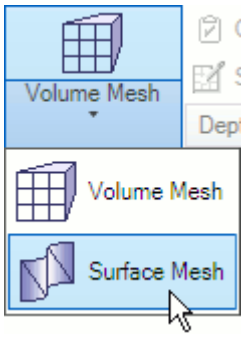
The final STAR-Works object tree is shown below.



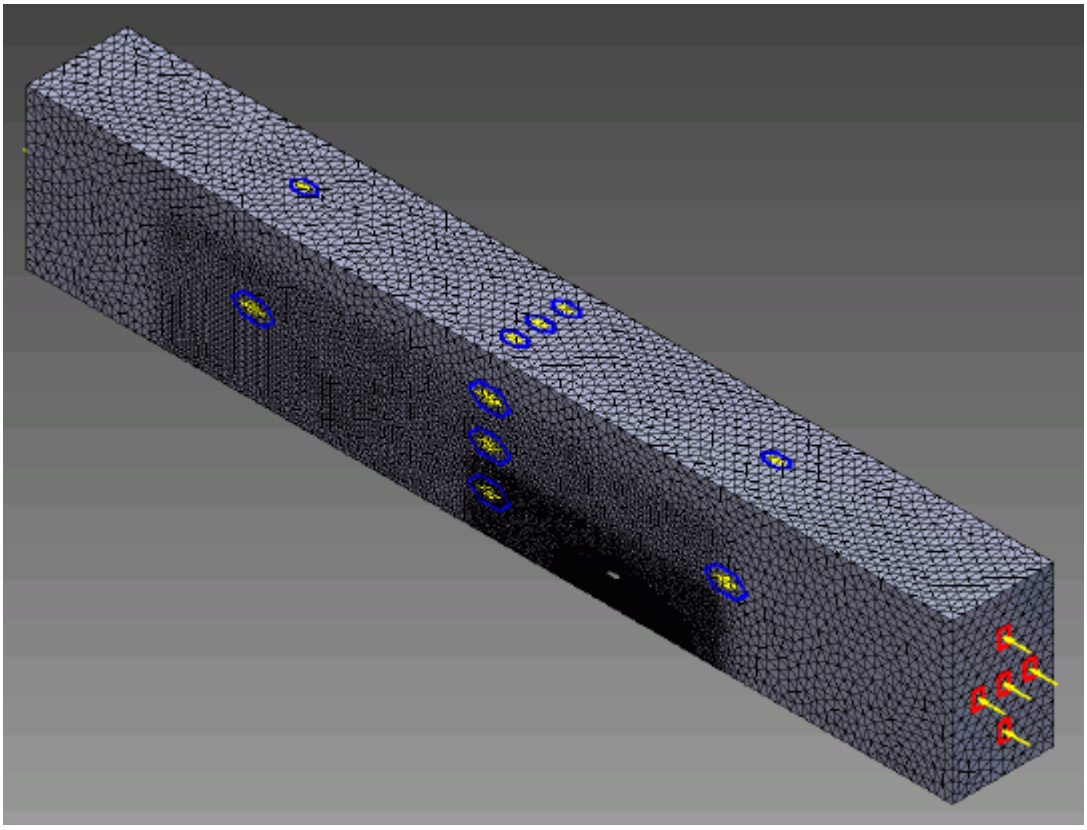
Generating the Surface Mesh

Generate the surface mesh:

1. From the STAR-Inventor ribbon, click ▾ next to the **Volume Mesh** icon and select **Surface Mesh**.



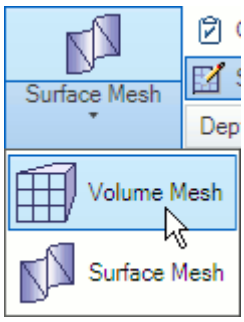
The surface mesh is shown below.



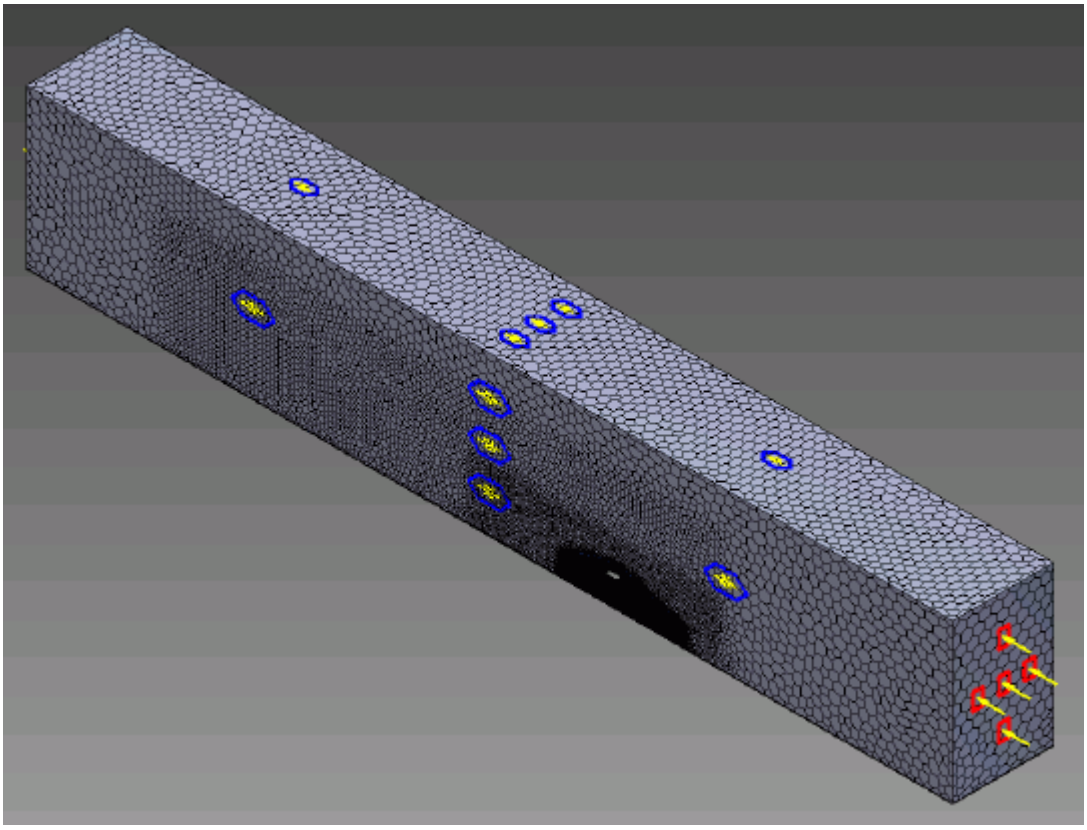
Generating the Volume Mesh

Generate the volume mesh:

1. From the STAR-Inventor ribbon, click ▾ next to the **Surface Mesh** icon and select **Volume Mesh**.



The volume mesh is shown below.

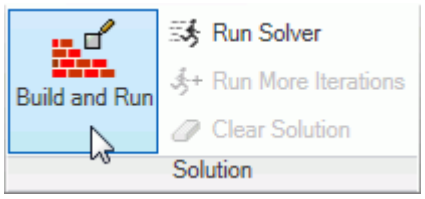


Building and Running

The case is now ready to run.

To run the simulation:

1. From the ribbon, click **Build and Run**.



The *Residuals* plot appears in the STAR-Inventor *Output* window. You can use this plot to monitor the convergence of the solution.

When the solver is done, STAR-CCM+ launches.

Post-Processing Results

You perform all post-processing in STAR-CCM+. By default, a scalar scene, *Scalar Scene 1*, automatically appears in the *Graphics* window when the solver finishes. This scene displays the absolute total pressure.

The results that are contained in the following sections are representative. You may see some variation in the results that you obtain, depending on what platform and package you are using.

1. In STAR-CCM+, click **File** > **Save** to save the simulation.

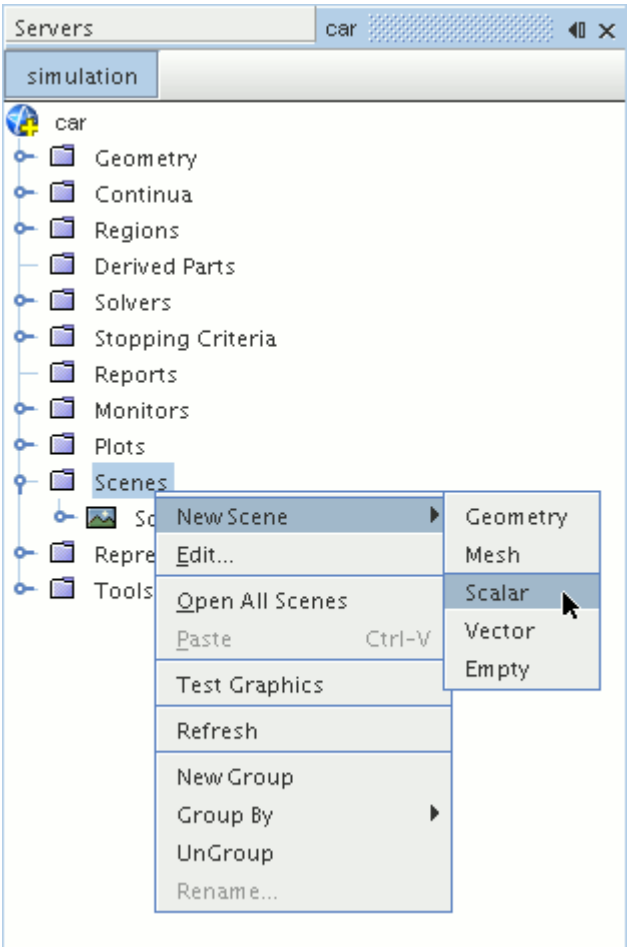
Contents:

Creating a Scalar Scene
Creating A Streamlines Scene
Creating a Vector Scene
Creating a Streamlines Scene
Creating A Scalar Scene
Visualizing the Wing Skin Temperature
Creating a Streamlines Scene
Creating A Scalar Scene
Creating a Streamlines Scene
Creating a Scalar Scene
Modifying The Scalar Scene
Creating a Streamlines Scene

Creating a Scalar Scene

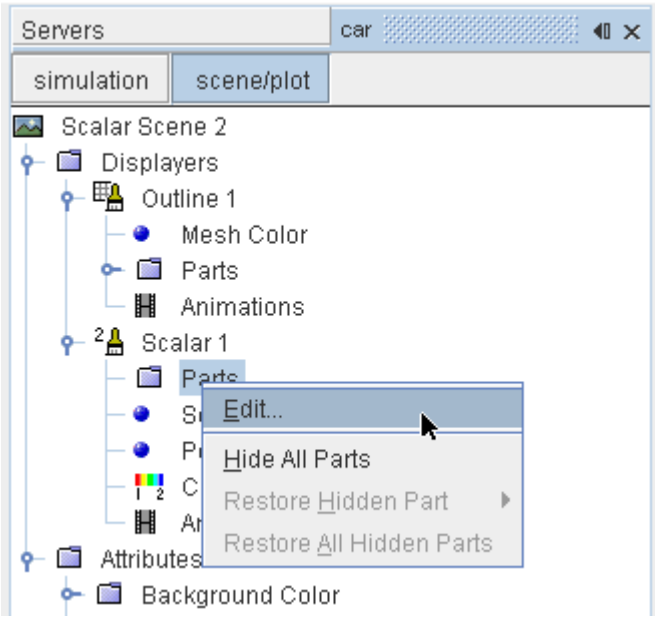
Create a scalar scene to display static pressure.

1. Right-click the **Scenes** node in the explorer pane and select **New Scene** > **Scalar**.



A new scalar scene, *Scalar Scene 2*, is created and selected in the *Graphics* window.

2. Click **scene/plot**.
3. Right-click the **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Parts** node and select **Edit...**.

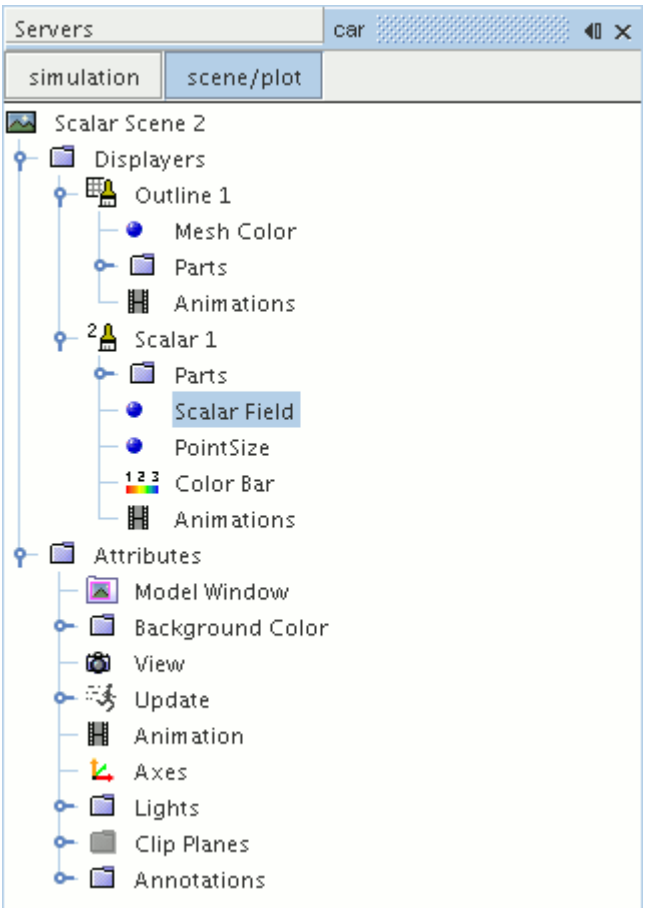


The *Parts* dialog appears.

4. Expand the **Regions** > **Region** > **Boundaries** node and select the following items:
   - **Car**
   - **Ground**
   - **Symmetry Plane**
5. Click **OK**.

Display the pressure contours in the scene:

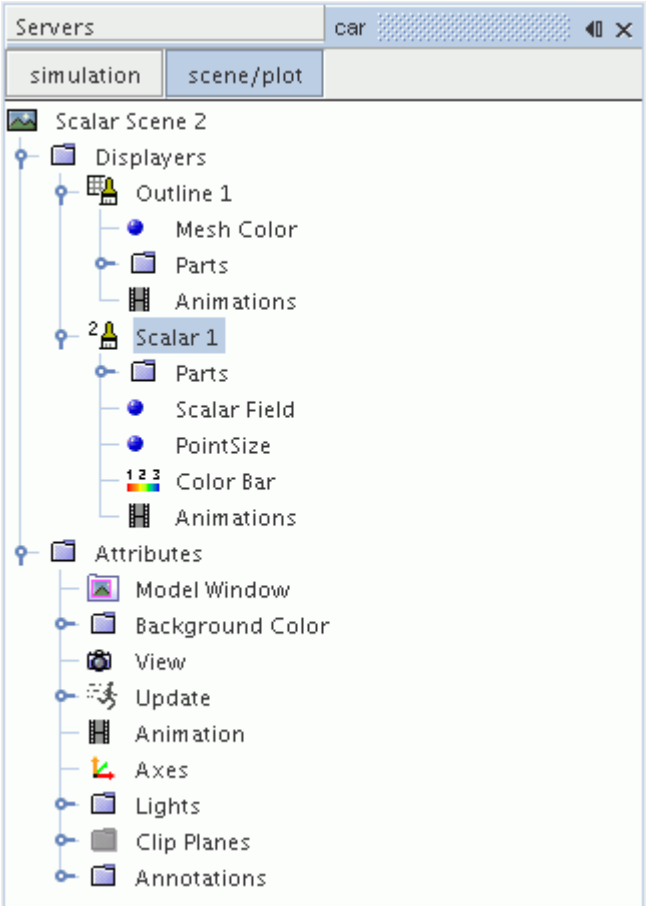
6. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Scalar Field**.



7. Set *Function* to **Static Pressure**.

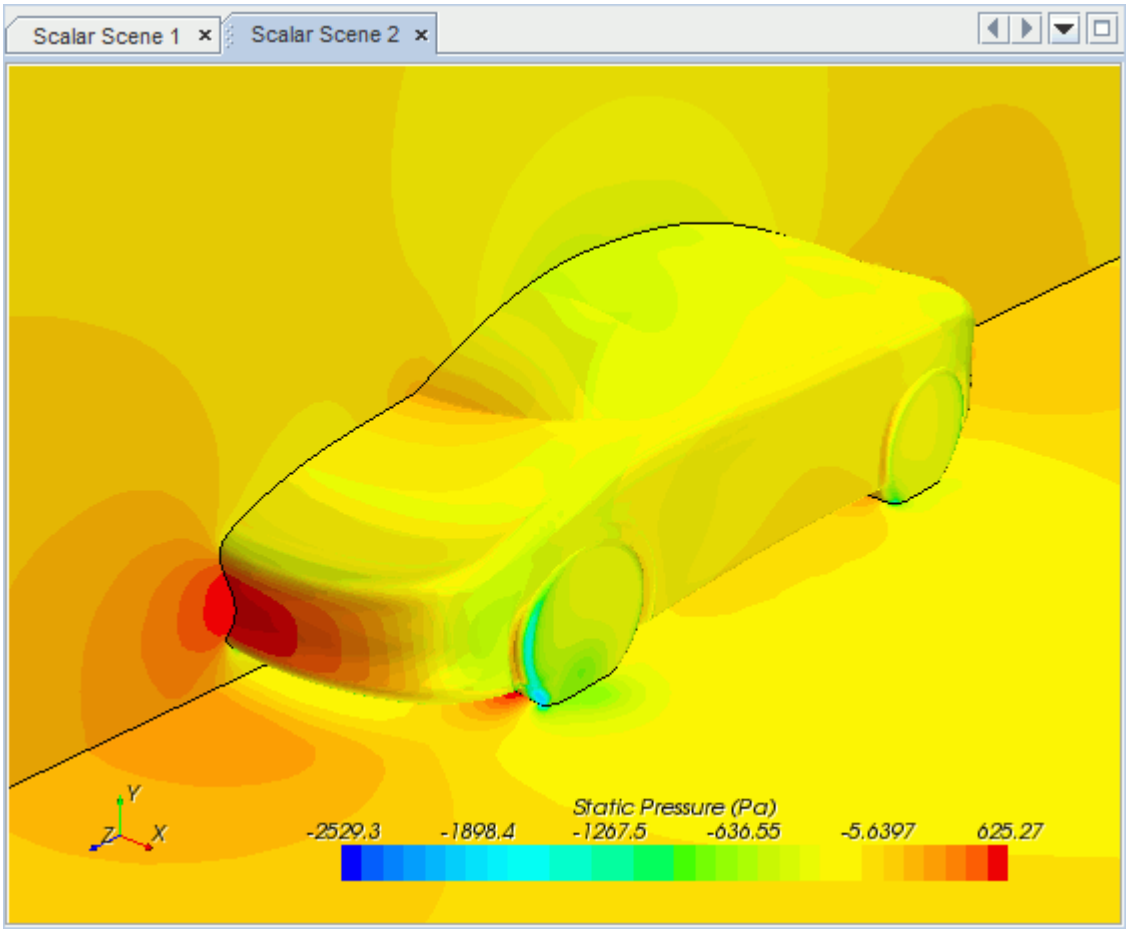
To have a smooth transition between the different pressure values:

8. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1**.



9. Set *Contour Style* to **Smooth Filled**.
10. Rotate the scene to get a better view of the front of the car.
11. Click (**Save**) to save the simulation.

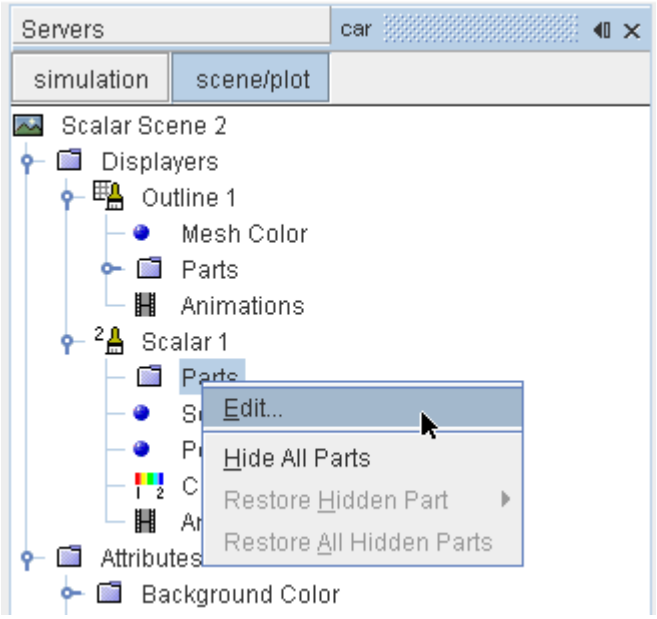
The finished scene is shown below.



Modifying The Scalar Scene

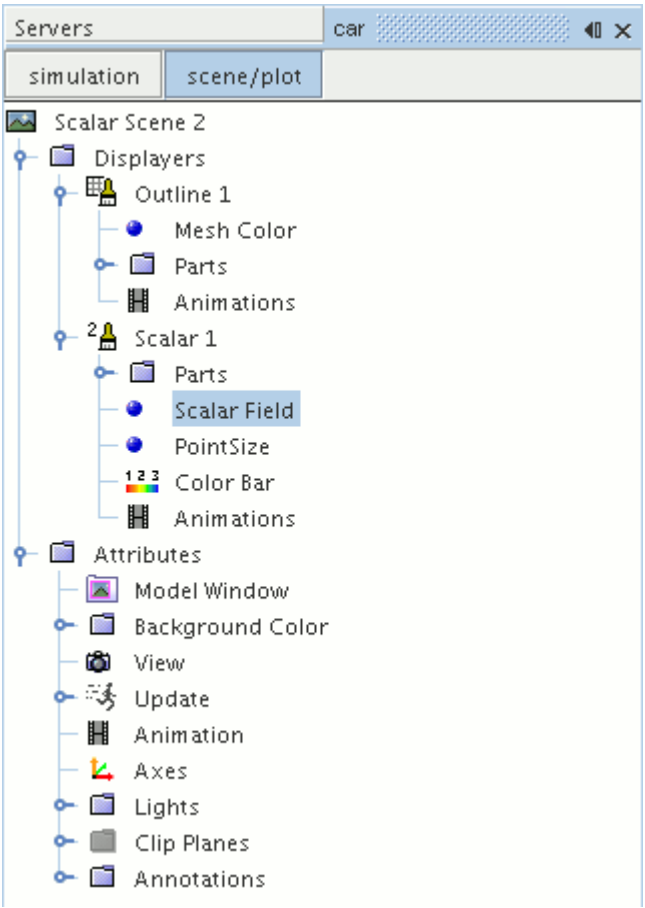
Modify Scalar Scene 2 to display Turbulent Kinetic Energy and remove the symmetry plane from the scene. You also mirror the car body:

1. Right-click the **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Parts** node and select **Edit...**.



The *Parts* dialog appears.

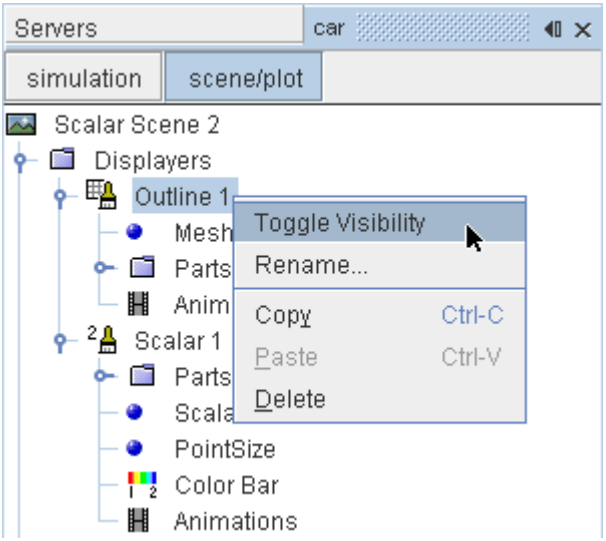
2. Expand the **Regions** > **Region** > **Boundaries** folder and deactivate **Symmetry Plane.**
3. Click **OK**.
4. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1** > **Scalar Field**.



5. Set *Function* to **Turbulent Kinetic Energy**.

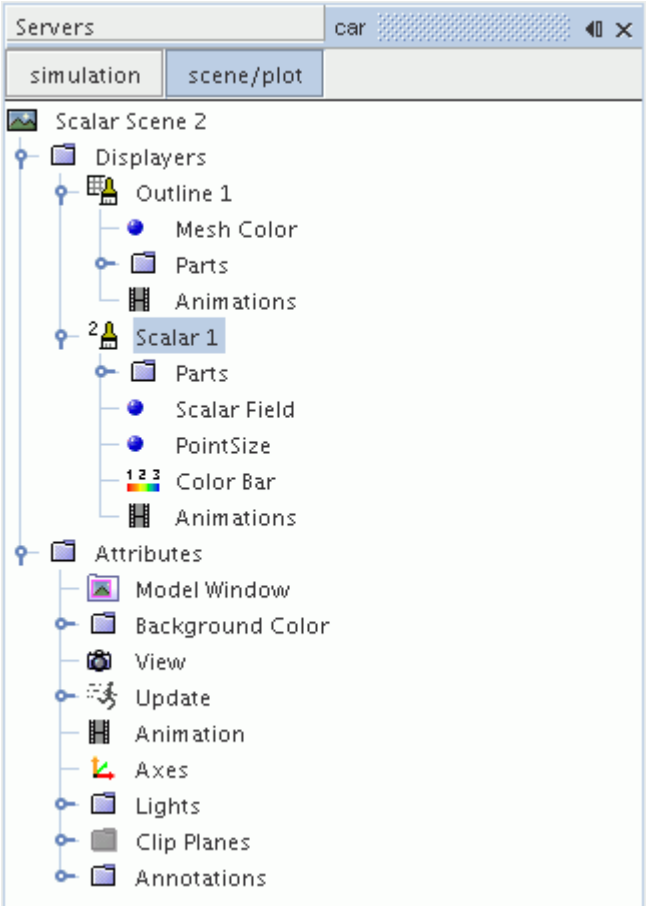
Hide the outlines in the scene:

6. Right-click the **Scalar Scene 2** > **Displayers** > **Outline 1** node and select **Toggle Visibility**.

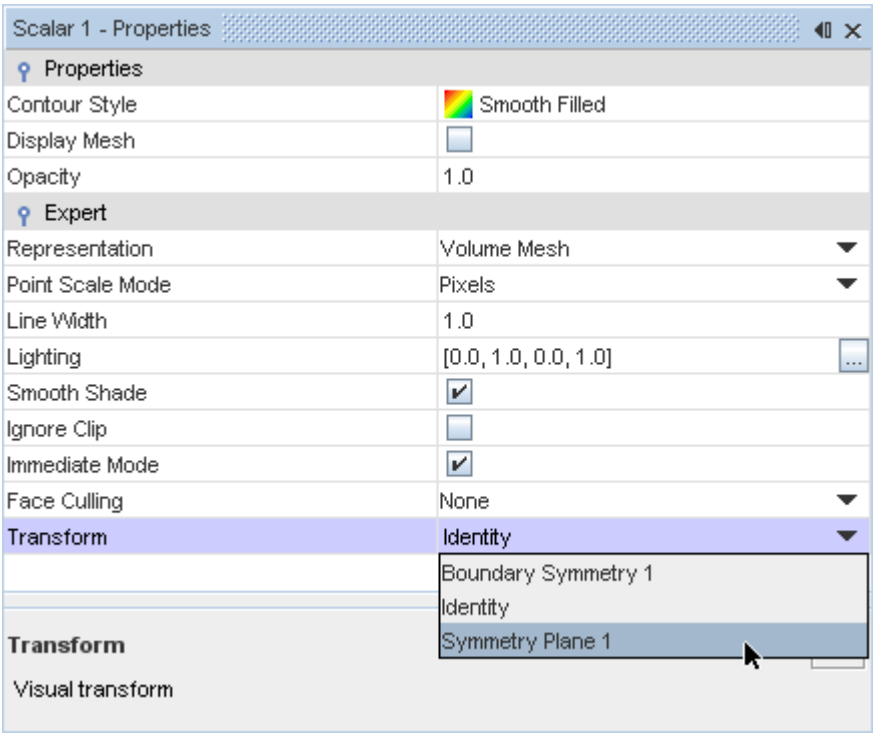


Mirror the car body:

7. Navigate to **Scalar Scene 2** > **Displayers** > **Scalar 1**.

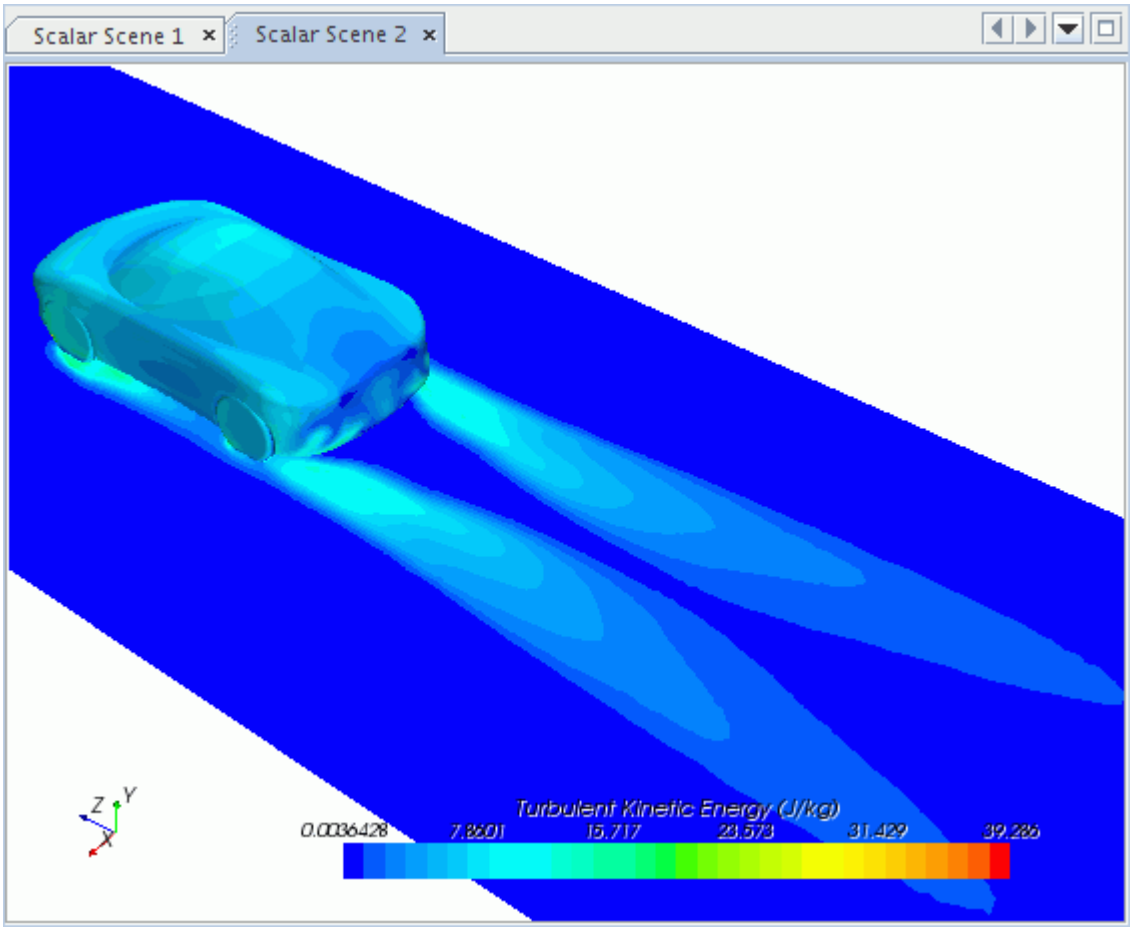


8. Set *Transform* to **Symmetry Plane 1**.



9. Rotate the scene to get a better view of the wake of the car.
10. Click **simulation**.
11. Save the simulation.

The finished scene is shown below.

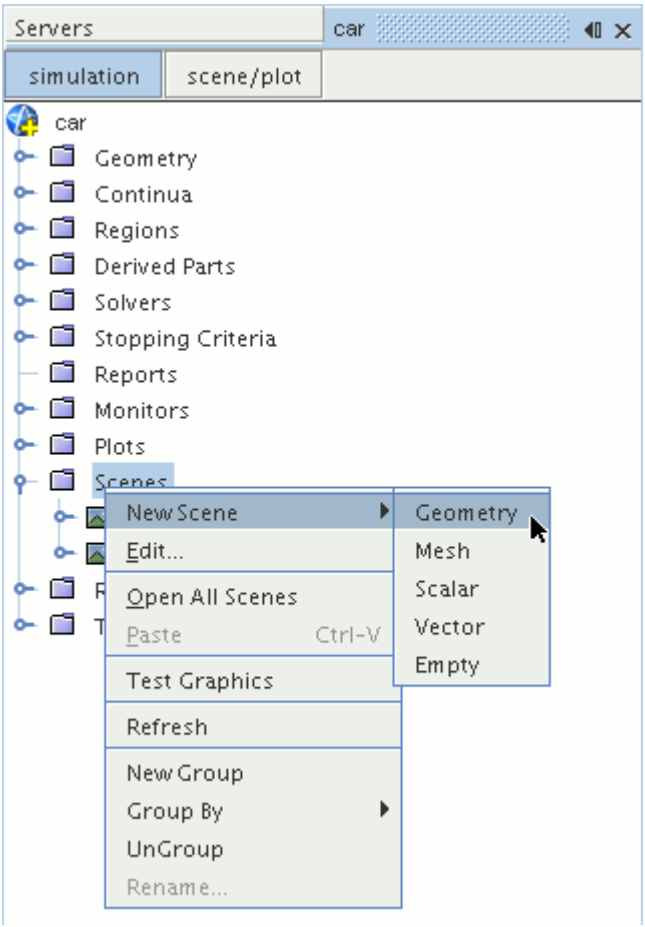


Creating a Streamlines Scene

Create streamlines to get a better understanding of the flow around the car and in the wake area.

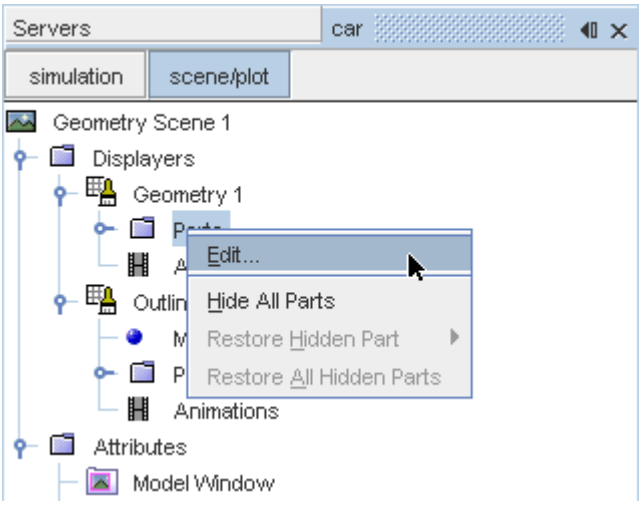
First, create a geometry scene:

1. Right-click the **Scenes** node in the object tree and select **New Scene** > **Geometry**.



A new geometry scene is created and selected in the *Graphics* window.

2. Click **scene/plot**.
3. Right-click the **Geometry Scene 1** > **Displayers** > **Geometry 1** > **Parts** node and select **Edit...**.



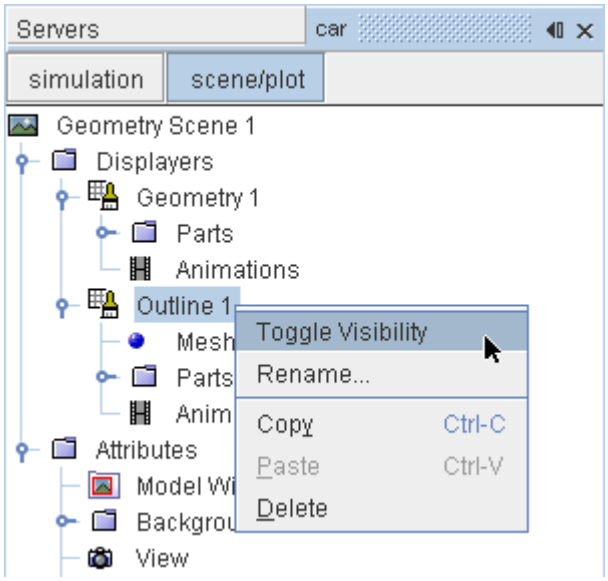
The *Parts* dialog appears.

4. Right-click on a blank area and choose **Deselect All**.

5. Expand the **Regions** > **Region** > **Boundaries** node and select **Car**.
6. Click **OK**.

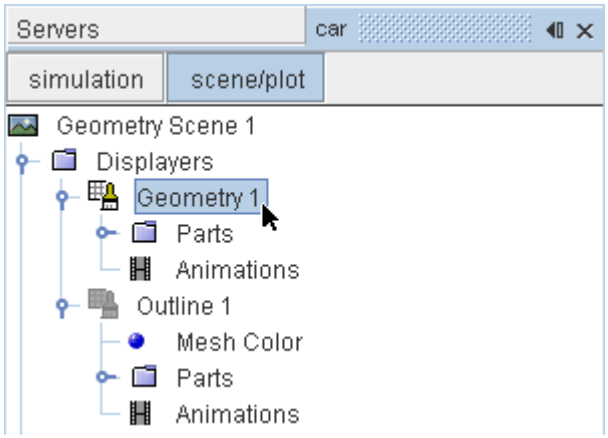
Hide the outlines in the scene:

7. Right-click the **Geometry Scene 1** > **Displayers** > **Outline 1** node and select **Toggle Visibility**.

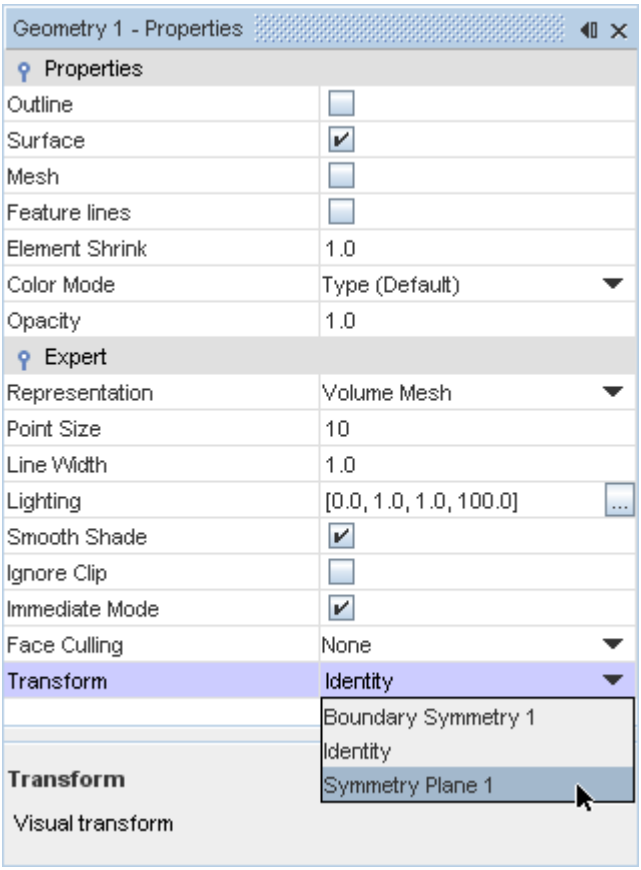


Mirror the car body:

8. Navigate to **Geometry Scene 1** > **Displayers** > **Geometry 1**.



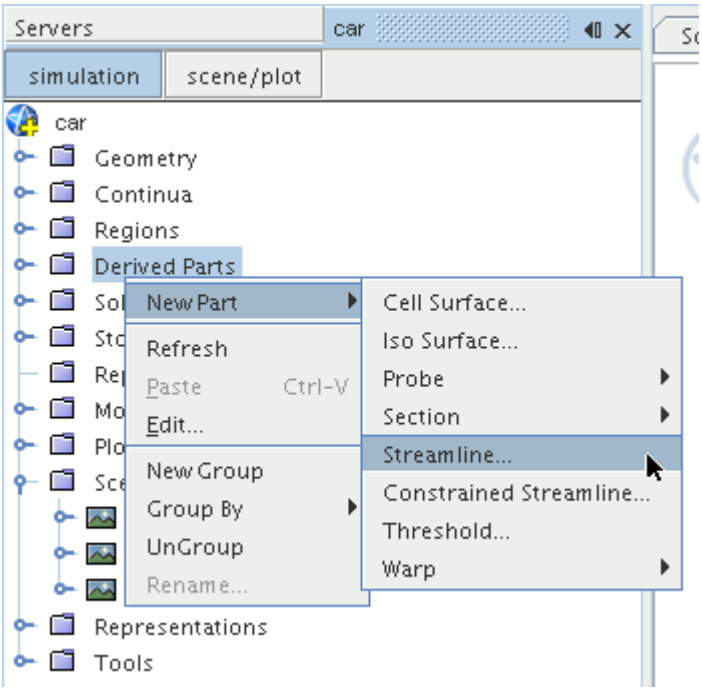
9. Set *Transform* to **Symmetry Plane 1**.



10. Click **simulation**.

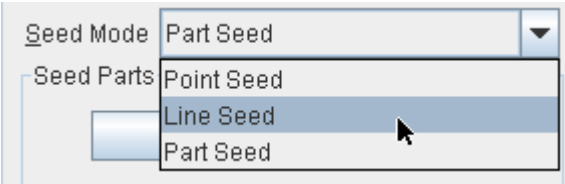
Create the streamlines:

11. Right-click the **Derived Parts** node and select **New Part** > **Streamline...**.



The *Create Streamline* dialog appears.

12. Set *Seed Mode* to **Line Seed**.



13. Enter these coordinates for the line:
    - *Point 1*:
        - X: `0 mm`
        - Y: `-123 mm`
        - Z: `2600 mm`
    - *Point 2*:
        - X: `1200 mm`
        - Y: `-123 mm`
        - Z: `2600 mm`
14. Set *Resolution* to `10`.
15. Make sure that **New Streamline Displayer** is selected in the *Display* section.
16. Click **Create**.

Create another set of streamlines using the existing dialog:

17. Enter these coordinates for the line:
    - *Point 1*:
        - X: `0 mm`
        - Y: `-473 mm`
        - Z: `2600 mm`
    - *Point 2*:
        - X: `1200 mm`
        - Y: `-473 mm`
        - Z: `2600 mm`
18. Click **Create** followed by **Close**.

Modify the appearance of the streamlines to display ribbons instead of the default lines.

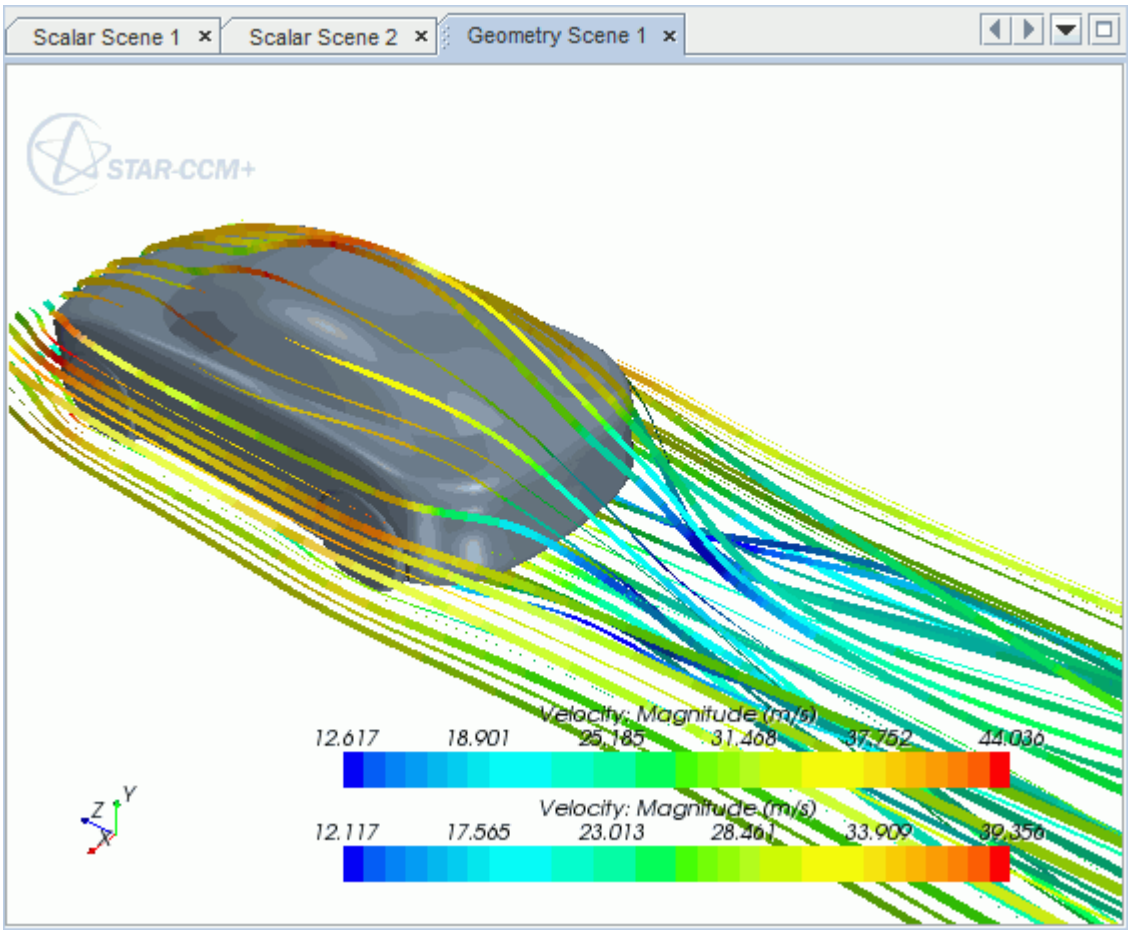
19. Click **scene/plot**.
20. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1**.
21. In the *Properties* window:
    - Set *Width* to `0.025`.
    - Set *Mode* to **Ribbons**.

- Set *Transform* to **Symmetry Plane 1**.

Modify the scalar field of the streamlines to display velocity magnitude:

22. Navigate to **Geometry Scene 1** > **Displayers** > **Streamline Stream 1** > **Scalar Field** and set *Function* to **Velocity** > **Lab Reference Frame** > **Magnitude**.
23. Repeat the above steps for **Streamline Stream 2**.
24. Rotate the scene to get a better view of the wake of the car.
25. Save the simulation.

The finished scene is shown below.



Geometry Transfer: CAD to Simulation

STAR-Inventor allows you to transfer geometry from Inventor to STAR-CCM+ through a direct link between the two packages.

This feature provides you with a workflow that is tailored to design optimization. You can apply changes to the geometry in either STAR-Inventor or STAR-CCM+ and the changes are passed back to the other program. For example, you can make changes to the design after running an analysis in the STAR-CCM+ environment and these changes are passed back to the original Inventor file. This feature requires a CAD client license.

This tutorial demonstrates how to select the design parameters in your original geometry and transfer the geometry from Inventor to STAR-CCM+. In STAR-CCM+, you proceed to create a repeatable workflow (or pipeline) using mesh operations. After setting up this pipeline, you can make further changes to the geometry design and have STAR-CCM+ regenerate the corresponding volume mesh.

Contents:

Opening the CAD Model
Transferring the Geometry to STAR-CCM+
Creating the Shape Parts and Operations
Setting Up the Mesh Pipeline
Modifying the Design

Opening the CAD Model

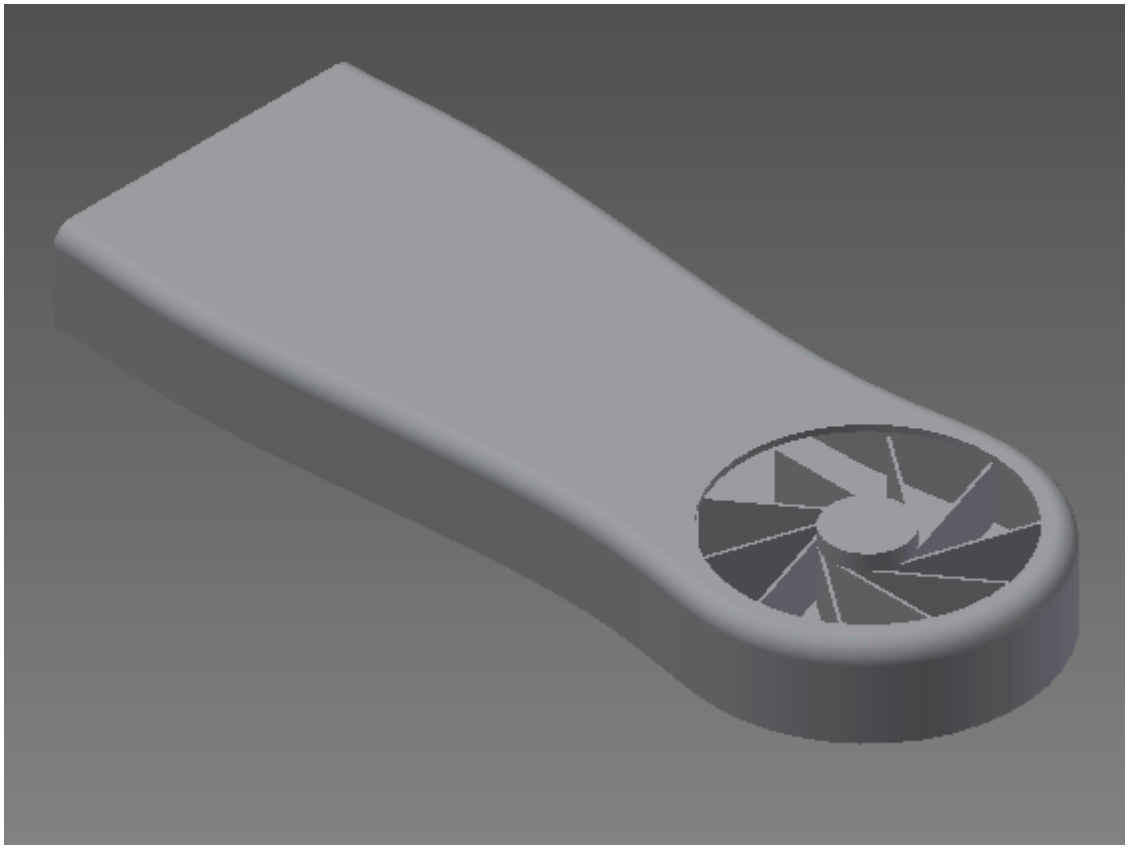
The STAR-CCM+ installation includes a directory containing the CAD geometry for this tutorial.

This directory is located in `[STAR-CCM+_INSTALL_DIR]\doc\startutorialsdata\starinventor\data`.

1. Copy directory `data` to your own disk space.
2. Launch STAR-Inventor and open the assembly file `cardAssem.iam`.

   The assembly is made up of two parts: the card housing and the blower fan.

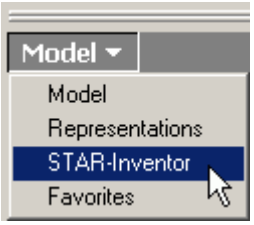
   The imported geometry is shown below.



The STAR-Inventor ribbon appears at the top of the screen. The STAR-Inventor object tree is located on the left of the screen.

To switch to the object tree:

3. Click **Model** and select **STAR-Inventor**.



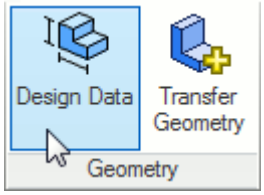
If these items are not visible, refer to Launching STAR-Inventor Manually.

Transferring the Geometry to STAR-CCM+

In STAR-Inventor, choose the design parameters to expose to STAR-CCM+ and then transfer the geometry.

Specify the design parameters to transfer to STAR-CCM+:

1. From the STAR-Inventor ribbon, click **Design Data**.



The *Design Data* dialog appears.

2. In the *Design Parameters* tab, select **blowerFan.ipt\d15**.

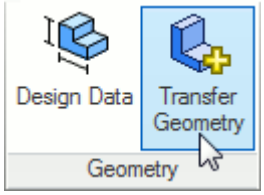
   STAR-Inventor only recognizes native CAD features as design parameters.

   The value of this parameter is displayed in the *Parameter Value* property.

3. Activate the *Transfer to STAR-CCM+* option.
4. Click **OK**.

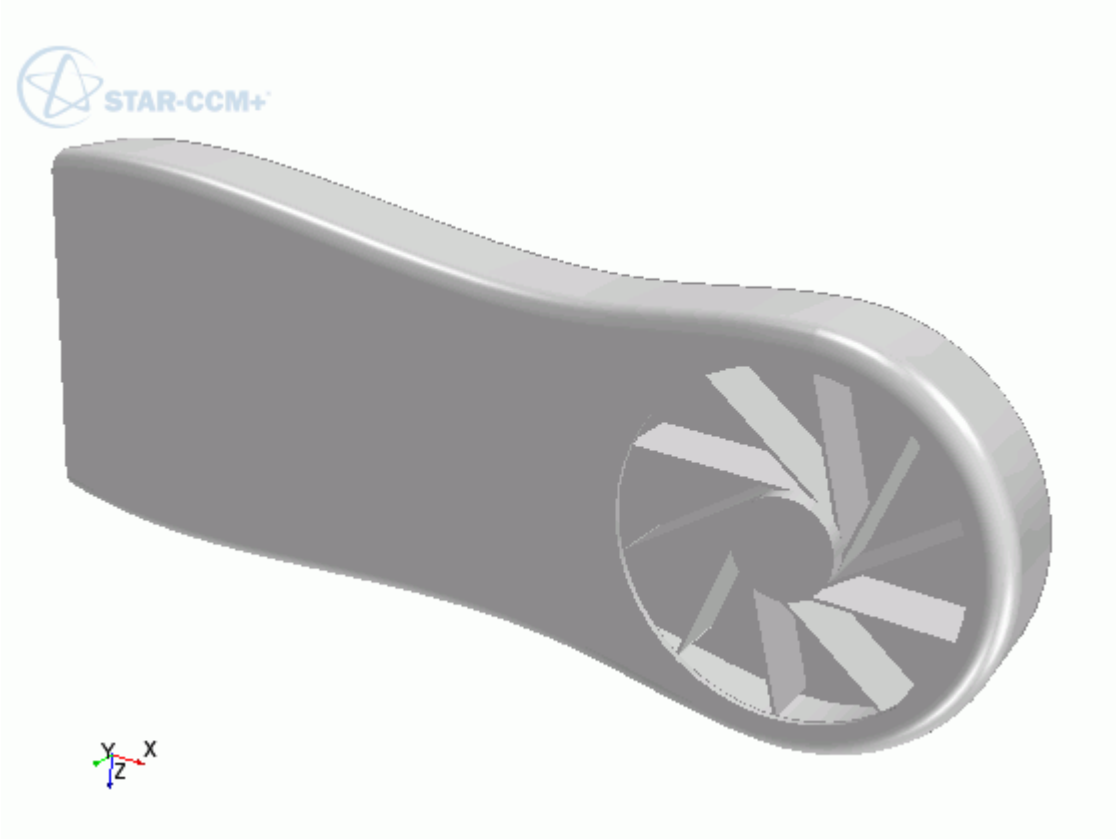
Transfer the geometry to STAR-CCM+:

5. From the STAR-Inventor ribbon, click **Transfer Geometry**.



STAR-CCM+ launches and the CAD geometry is transferred over.

6. In STAR-CCM+, create a geometry scene and position the geometry as shown below.



7. In STAR-CCM+, expand the **Geometry** node.

   The transferred geometry appears under the **Parts** node. The properties that control the connection back to Inventor, including the design parameter value, appear under the **CAD Client Models** node.

When transferring geometry from Inventor to STAR-CCM+, a link exists between the two software packages. This link allows data to be transferred between the two packages. To maintain this link, you must first save the geometry file in Inventor:

8. Switch to Inventor and save the geometry file.

   In response, STAR-CCM+ saves the simulation file with the name of the Inventor geometry file. Using this technique, you can rename or relocate both the geometry and simulation files without breaking the link between the two packages. You can also save the simulation file in STAR-CCM+ so long as you do not rename it or relocate it. Doing so will break the link between the packages.

Creating the Shape Parts and Operations

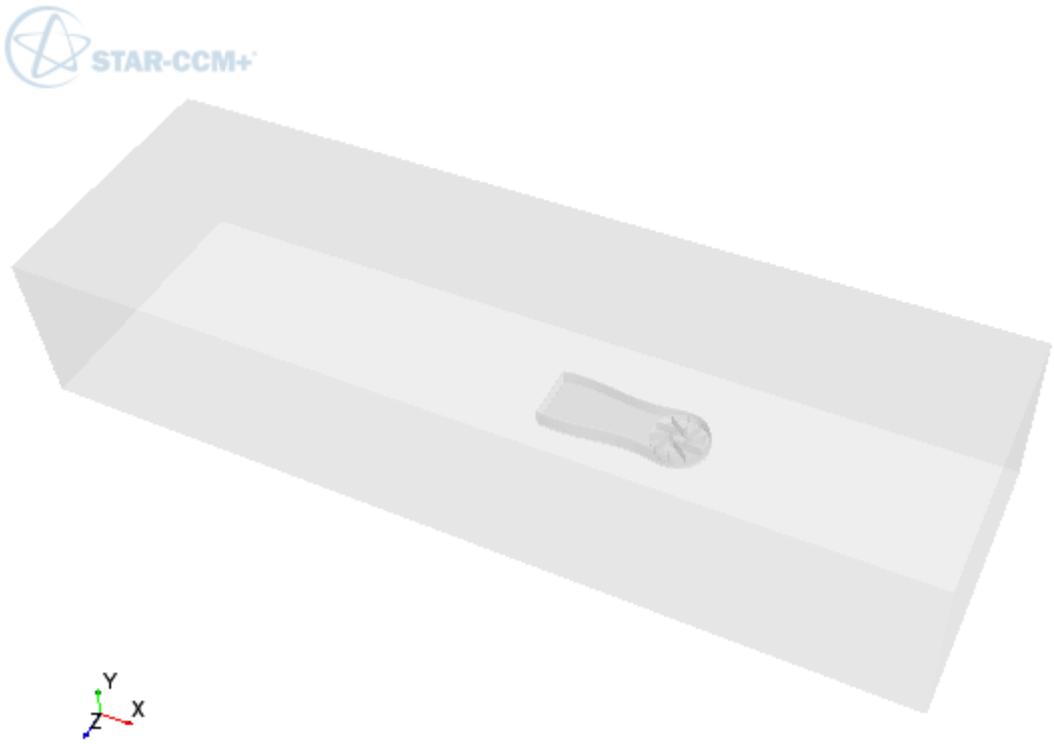
The transferred geometry consists of the card housing and the blower fan. To simulate this case, you require two regions: one static region that provides an enclosure for the simulation, and another for the rotating region around the fan.

The transferred geometry contains the solid parts. You require the fluid volume around these parts. The first stage in obtaining the fluid volume is to create part shapes that correspond to the two regions. To save time, you are provided with these part shapes, one for the enclosure, and another for the rotating region.

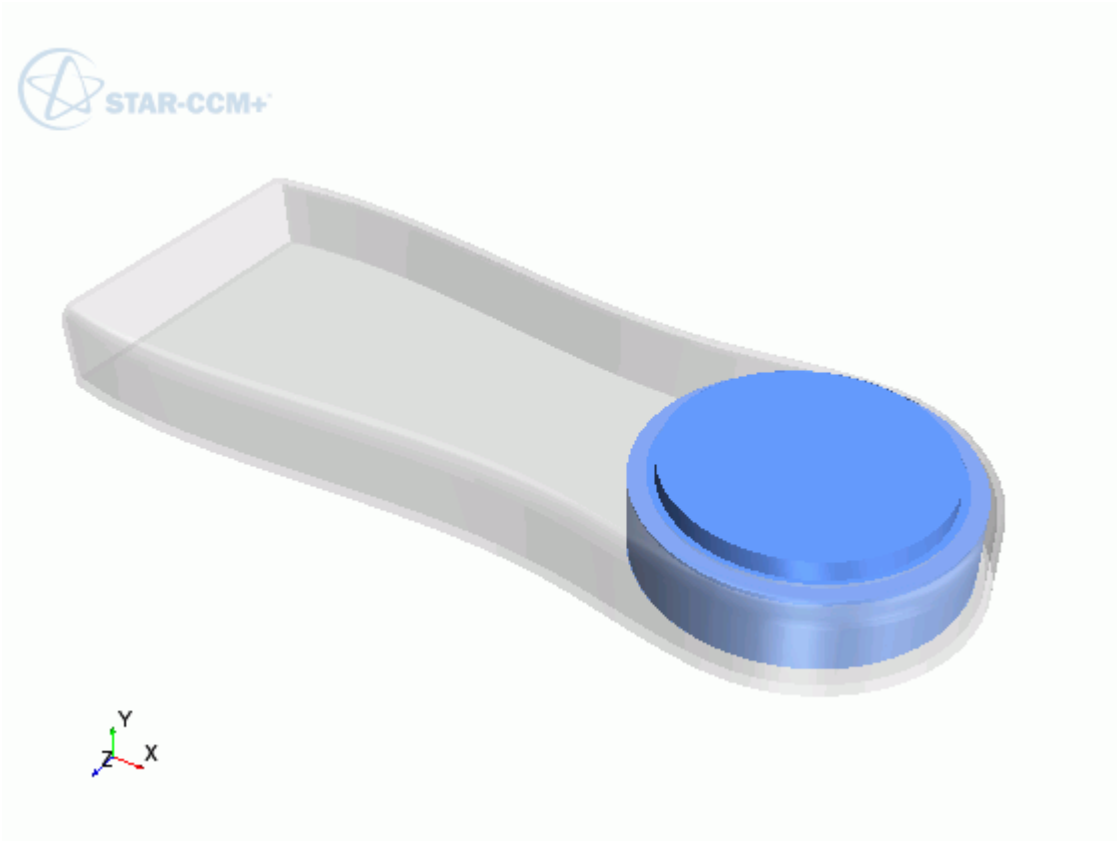
Import the shape parts:

1. Select **File** > **Import** > **Import Surface Mesh**.
2. Import the part shapes:
   a) In the *Open* dialog, navigate to the `data` directory and select **box.x_t** and **disc.x_t**.
   b) Click **Open**.
   c) In the *Import Surface Options* dialog, deactivate the **Open Geometry Scene After Import** option.
   d) Click **OK**.

   The static region, **box**, is shown below in relation to the graphics card:



   The rotating region, **disc**, is show below in relation to the graphics card:



Now subtract the solid CAD geometry from these shape parts to obtain the final fluid volume. Begin by subtracting the blower fan geometry from the **disc** part:

3. Right-click the **Operations** node and select **New** > **Boolean** > **Subtract**.
4. In the *Create Subtract Operation* dialog, set the following properties:

Property	Setting
Input Parts	**disc**
	cardAssem.iam > **blowerFan:1**
Target Parts	**disc**

5. Click **OK**.
6. Rename the **Operations** > **Subtract** node to `Fan Subtraction`.
7. Rename **Parts** > **Subtract** node to `Rotating Region`.

Subtract the card housing geometry and the disc part from the static region:

8. Create another subtract operation with the following settings:

Property	Setting
Input Parts	**box**
	disc
	cardAssem.iam > **cardCase:1**

Property	Setting
Target Parts	**box**

9. Rename **Operations** > **Subtract** node to `Card Subtraction`.
10. Rename **Parts** > **Subtract** node to `Static Region`.

To get a conformal mesh between the two regions, imprint both fluid volumes together:

11. Right-click the **Operations** node and select **New** > **Boolean** > **Imprint**.
12. In the *Create Imprint Operation* dialog, select **Rotating Region** and **Static Region** and click **OK**.

The transferred geometry contains CAD data. To ensure best results, instruct all operations to use the CAD data where available:

13. Under the **Operations** node, multi-select the **Fan Subtraction** and **Card Subtraction** nodes and activate the *Perform CAD Boolean* option.
14. Select the **Operations** > **Imprint** node and set *Merge/Imprint Method* to **CAD Imprint**.

    The contacts between the two parts are created. These contacts are used while creating the regions to define the interfaces between the regions.

Execute all operations:

15. Right-click on the **Operations** node and select **Execute All**.

Setting Up the Mesh Pipeline

Assign the parts to regions and create the automated mesh operation.

Assign the fluid parts to regions:

1. Expand the **Geometry** > **Parts** node and multi-select the **Rotating Region** and **Static Region** nodes.
2. Right-click on one of the selected parts and choose **Assign Parts to Regions...**.
3. In the *Assign Parts to Regions* dialog:
   a) Select **Create a Region for Each Part**.
   b) Select **Create Boundary for Each Part Surface**.
   c) Click **Apply** and **Close**.

Create the automated mesh operation:

4. Right-click the **Operations** node and select **New** > **Mesh** > **Automated Mesh**.
5. In the *Create Automated Mesh Operation* dialog:
   a) From the *Parts* list, select **Rotating Region** and **Static Region**.
   b) Select the following meshers, in order:

Group	Mesher
Surface Meshers	**Surface Remesher**
Volume Meshers	**Polyhedral Mesher**
Optional Boundary Layer Meshers	**Prism Layer Mesher**

c) Click **OK**.

An automated mesh operation is added to the **Operations** node. Refine the mesh settings for this case:

6. Expand the **Automated Mesh** > **Default Controls** node.
7. Select the **Base Size** node and set *Value* to `16 mm`.
8. Select the **Prism Layer Total Thickness** > **Relative Size** node and set *Percentage of Base* to `20`.

Generate the volume mesh:

9. Click (**Generate Volume Mesh**).
10. When the volume mesh is generated, create a mesh scene that displays the following parts:
    - **Rotating Region** > **Boundaries** > **cardAssem.iam.blowerFan:1.<all_faces>**
    - **Static Region** > **Boundaries** > **cardAssem.iam.cardCase:1.<all_faces>**

You can exit STAR-CCM+ and Inventor and maintain the link between the two packages. To demonstrate, in the following steps you save the simulation and exit both applications. In the next section, you will resume the simulation and continue where you left off.

11. Save the simulation.
12. Exit STAR-CCM+.
13. Exit Inventor

Modifying the Design

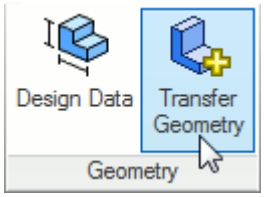
Modify the number of blades on the blower fan using the design parameter.

In this tutorial, you modify the number of blades on the graphics card blower fan from 10 to 15 blades.

To re-establish the link between Inventor and STAR-CCM+, you must load the geometry in STAR-Inventor first and transfer the geometry to STAR-CCM+.

Resume the analysis and re-establish the link between STAR-Inventor and STAR-CCM+:

1. Launch Inventor with STAR-Inventor and load `cardAssem.iam`.
2. From the STAR-Inventor ribbon, click **Transfer Geometry**.



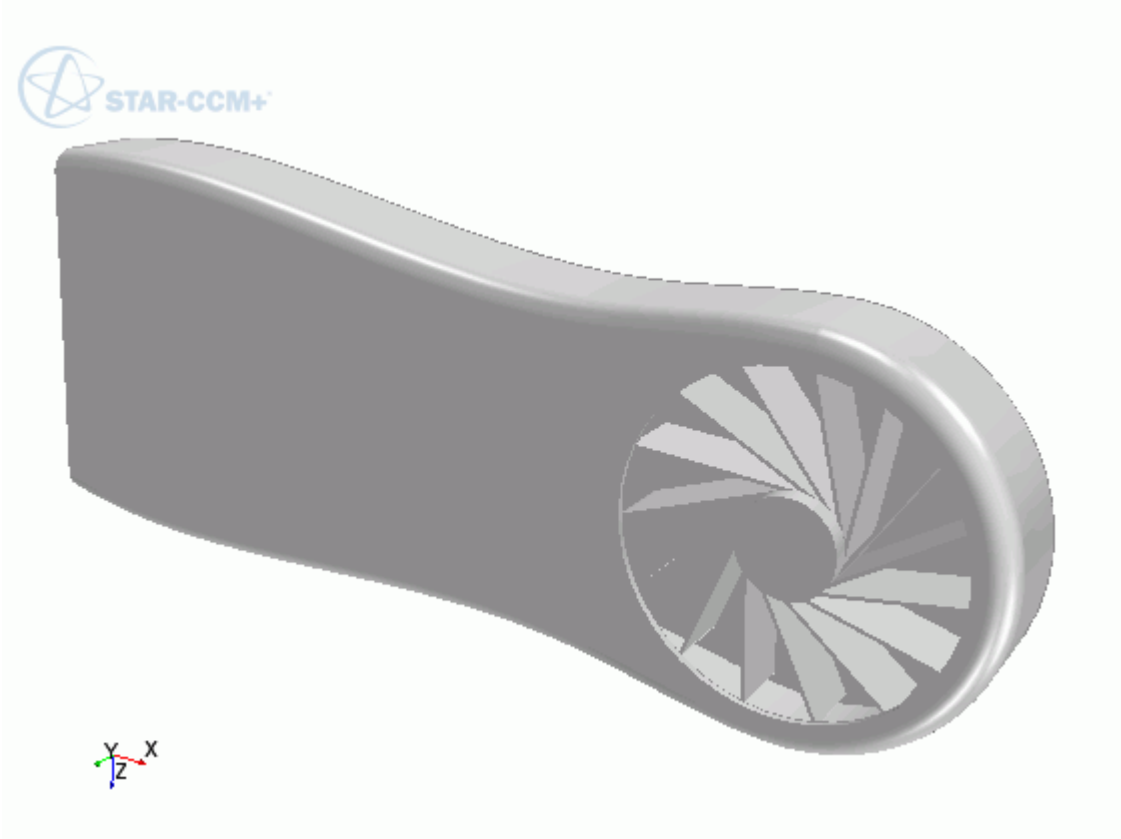
   STAR-CCM+ launches and the CAD geometry is transferred over.
3. Switch to STAR-CCM+.
4. Open the geometry scene to visualize the initial geometry representation.
5. Expand the **Geometry** > **CAD Client Models** > **cardAssem.iam** > **Design Parameters** node.
6. Select the **blowerFan.ipt\d15** node and set *Parameter Value* to `15`.

   A yellow warning icon appears on the design parameter node as well as the assembly node. This icon indicates that the CAD Client Model is out of date.

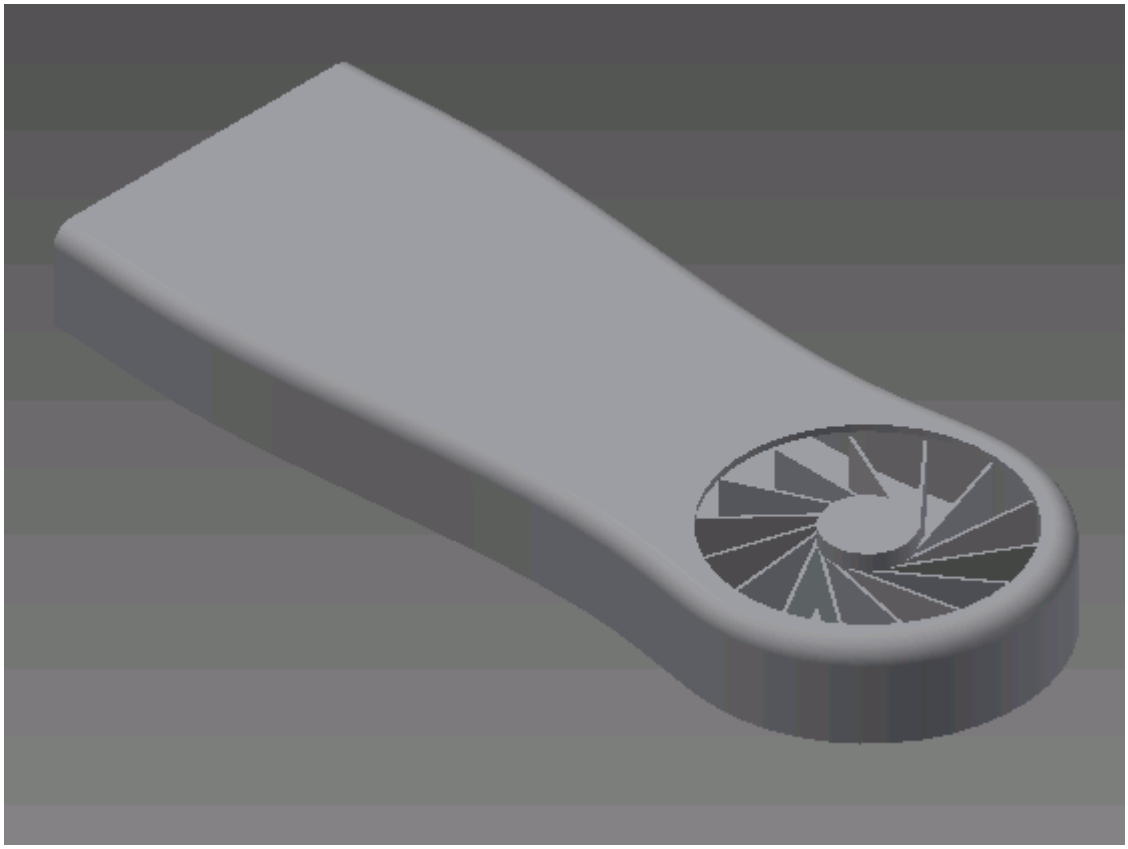
To update the model:

7. Right-click the **cardAssem.iam** node and select **Update Model**.

   The geometry updates in the scene. STAR-CCM+ passes the change in the design back to Inventor.



8. Switch back to Inventor and view the updated geometry.



The pipeline you set up is fully repeatable. To update the volume mesh:

9. In STAR-CCM+, open the mesh scene.

10. Click (**Generate Volume Mesh**).
    The volume mesh updates to reflect the changes in the design.

All boundaries and boundary conditions that you set are maintained after this change. This feature reduces the amount of work that is required when analyzing multiple designs.

Siemens PLM Software

Headquarters
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asia-Pacific
Suites 4301-4302, 43/F
AIA Kowloon Tower,
Landmark East
100 How Ming Street
Kwun Tong, Kowloon
Hong Kong
+852 2230 3308

About Siemens PLM Software

Siemens PLM Software, a business unit of the Siemens Digital Factory Division, is a leading global provider of product lifecycle management (PLM) and manufacturing operations management (MOM) software, systems and services with over 15 million licensed seats and more than 140,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software works collaboratively with its customers to provide industry software solutions that help companies everywhere achieve a sustainable competitive advantage by making real the innovations that matter. For more information on Siemens PLM Software products and services, visit www.siemens.com/plm.

© 2017 Siemens Product Lifecycle Management Software Inc. Siemens and the Siemens logo are registered trademarks of Siemens AG. D-Cubed, Femap, Fibersim, Geolus, GO PLM, I-deas, JT, NX, Parasolid, Polarion, Solid Edge, Syncrofit, Teamcenter and Tecnomatix are trademarks or reg istered trademarks of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries. All other logos, trademarks, registered trademarks or service marks belong to their respective holders.

Index

(m',theta) derived part 5815

1D momentum method 3713
2D meshes 2653
3D-CAD
    3D sketch 1174
    bodies 1309
    Boolean operations 1336
    chamfer 1401
    See also sketch chamfer
    constraints 1237
    defeaturing 1426
    design parameter 1406
    dimensions 1255
    edges 1356
    exporting geometry 1450
    external volume extraction 1421
    extrusion 1272
    faces 1356
    fill holes 1370
    fillet 1398
    See also sketch fillet
    fluid extract 1415
    importing geometry 1439
    internal volume extraction 1417
    mirror 1331
    patch surface 1370
    pattern 1385
    reflect, *See* mirror
    rotate 1331
    scale 1330
    sewing bodies 1320
    sheet body 1310
    shell 1321
    sketch 1163
    sketch chamfer 1226
    sketch fillet 1217
    sketch plane 1182
    solid body 1309
    spline 1196
    split 1205
3D-CAD *(continued)*
    surface modeling 1163
    sweep 1300
    toolbar 1155
    translate 1329
    trim 1205
    twist 1304
3D-CAD models 289

A

Abaqus
    co-simulation 6771
    file-based coupling 6676
accumulated force tables 268
acoustic suppression zone 4020
actuator disk 3668
adjoint method
    control point creation 3603
    error estimates 3599
aeroacoustics
    automobile aerodynamics 5700
    computational approach 4019
    Ffowcs Williams-Hawkings modeling 4024
    Fourier transforms 5713
    guidelines
        axial fans 5709
        freestream boundary condition 5712
        hydroacoustics 5712
        liquid media 5712
        mesh issues 5712
        post-processing 5713
        radial compressors 5706
        sunroof buffeting 5704
        turbulence model selection 5711
        wave6 export 5718
    hydroacoustics 4064
    in incompressible fluids 3996
    wavenumber decomposition 5713
aerodynamics
    guidelines
        incompressible external 5726

transonic/supersonic *(continued)*
guidelines *(continued)*
transonic/supersonic 5728
vehicle simulations 5731
transonic/supersonic
guidelines 5728
air flow in electronic systems
guidelines
boundary conditions 5738
forced or natural convection 5738
geometry preparation 5735
mesh preparation 5737
monitoring 5740
physics set-up 5738
solution set-up 5740
AKN k-epsilon low-Reynolds model 3759
ALE (Arbitrary Lagrangian-Eulerian) method 3400
algebraic multigrid parameters 3289
alternate values,
data mapping 6917
Amesim, co-simulation 6758
ANSYS, file-based coupling 6679
approximate imprinting method 6933
arbitrary probe 5830
arbitrary section 5803
assembly, *See* composite
atomization
pressure swirl injector 4351
axisymmetric meshes 2653
axisymmetric swirl 3354

B

background component 3546
bad cells
identifying 2700
baffles
representing in mesh 2197
batteries
battery cell
configuring 5565
creating 5561
data overrides 5574
definition 5561
electrical mesh resolution 5567
properties 5568
electrical load *(continued)*
battery cell *(continued)*
summary report 5573
Battery Design Studio 5541
batterysim license 5528
ecell 5533
electrical load
defining 5630
electrical solver 5536
electrochemistry data, exporting 5657
electrode structure tortuosity 5659
external casing 5622
field functions 5670
meshing 5614
modeling overview 5530
modeling process 5532
modeling, workflow 5537
selecting 5539
solver coupling 5537
thermal solver 5537
battery cell
configuration within module 5589
electrode masks 5597
jelly roll properties 5595
unrolled view of spiral cell 5593, 5596
Battery Design Studio
overview 5541
program files 5558
spiral cell configuration 5550
stacked cell configuration 5543
battery model
concentration 5664
DistNP 5666
DistSegregated 5667
NTG (Newman, Tiedeman, Gu) 5660
physics-based 5662
potential 5662
RCR 5661
solid electrolyte interface current density 5665
battery module
cell configuration 5583, 5589
cell orientation 5585
configuration 5582
connector ohmic heating 5617, 5620
connector resistance 5605

external casing *(continued)*
energy source definition 5610
external casing
bounding box 5623
casing methods 5622
casing part creation 5627
casing part selection 5626
definition process 5622
mesh continuum set up 5614
mesh creation 5614
module layout 5587
number of cells 5580
overview 5577
parts generation 5608
region generation 5609
setup 5577
tab connections 5601
terminal connections 5598
battery solver 5540
beam data
mapping 6912, 6921
bibliography
batteries 5668
DEM 4610
derived parts 5856
electromagnetism 8010
multiphase segregated flow 7706
solid stress 5267
VOF multiphase 7743
body force, user-defined Lagrangian phase 4299
boiling, bulk
boiling mass transfer rate 4795
interphase mass transfer 4791
model selection 4789
multi-component boiling mass transfer rate 4796
overview 4787
boiling, fluid film
formulation 4504
model selection 4475
overview 4473
workflow 4474
boiling, VOF
boiling temperature 4936
boundary conditions 4942
film boiling vapor volume fraction 4944
boiling, VOF *(continued)*
formulation 4989
heat transfer cross-section area 4941
mixture components 4935
model selection 4937
overview 4932
Rohsenow model 4938
transition model 4940
workflow 4934
Boolean operations
3D-CAD 1336
boundaries 301
boundary conditions
DEM
boundary values 4546
Lagrangian phase 4540
material properties at wall 4546
particle discharge at wall 4543
dispersed multiphase 4879
electromagnetism
electric currents 5316
Eulerian
impermeable outlet 4864
permeability 4863
permeable wall 4866
user wall heat flux coefficient specification 4785
Lagrangian
default conditions 4321
interaction modes 4327
overview 4320
phase impermeable 4325
specific boundary 4322
solid stress (FE), *See* segments
solid stress (FV) 5268
VOF multiphase
boiling 4942
cavitation 4951
overview 4899
boundary values
Lagrangian 4263, 4348
bounded differencing 3252, 3288
bounded-central 3288
Boussinesq Model 3353
bulk ion chemical reactions
overview 5473

C

CAD repair 1668
calcination 4205
casting
  artificial compressibility 7315
  automated time-step control 7310, 7311, 7314
  casting process selection 7322
  criteria functions 7344
  defect prediction using criteria functions 7344
  macro porosity
    available models 7290
    fully coupled 7288
    pure thermal 7288
    setup 7288
  material database, *See* material database, casting
  optional physics models 7336
  pure thermal simulation 7235
  STAR-Cast license 7228
  variable density compensation 7236
  workflow 7229
casting user interface 7317
cavitation
  boundary conditions 4951
  bubble growth rate 4993
  enthalpy change 4999
  formulation 4993
  Homogeneous Relaxation model 5001
  mass transfer 4993
  model properties 4955
  models 4951
  multi-component 4993
  overview 4951
  parameters 4955
  phase composition 4952
  phase interaction 4953
  Rayleigh–Plesset equation 4993
  saturation pressure 4951
  Schnerr-Sauer model 4993
  seed density 4951
  seed diameter 4951
  workflow 4951
ccm mesh format 2161
cell
  mesh definition 2144
CFM combustion
CFM combustion *(continued)*
  order of components 3546
client workspace 94
co-RELAP, co-simulation 6857
co-simulation
  Abaqus 6771
  Amesim 6758
  GT-SUITE 6810
  overview 6597
  reacting channel 4233, 6875
  RELAP 6857
  STAR-CCM+ 6722
  WAVE 6842
co-simulation API 6892
combustion
  burners 4141
  CFL scaling factor 4109
  coal 4141
  flame positioning models
    coherent flame model 4141
    turbulent flame-speed closure 4141
  flame propagation 4140
  furnaces 4141
  gas turbine 4141
  ignitors
    overview 4101
  importing chemical reactions 4247
  importing mixture components 4247
  in multiphase flow 4785
  large residence time 4141
  model overview
    chemical equilibrium 4139
    complex chemistry 4099
    eddy contact micromixing 4100
    eddy-break-up 4100
    flamelet generated manifold 4139
    polymerization 4100
    steady laminar flamelet 4139
    thickened flame 4100
  of coal 4205
  of particles 4205
  overview 4065
  selecting models 4065
  surface chemistry 4221
  workflow 4075

complex chemistry
clustering
properties 4113
importing Chemkin files 4247
in situ adaptive tabulation
properties 4113
concurrent meshing 2153
condensation, fluid film 4927
condensation, VOF 4927
conformal mesh 2159
conjugate heat transfer
conformal mesh 2159
constrained plane section 5799
constraint 6486
constraint checks, data mapping 6917
constraint on solid 5255
constraints on solids 5227
continua 292
convection 3430
convergence control
for the coupled approach 3263
cooling
Newton's law 3437
coordinate system types 2977
corrosion
cathodic protection 5405
overview 5403
coupled energy model 3253
coupling with CAE products 6597
criteria functions, casting
activating in phase interaction 7297
cooling rate 7303
critical temperatures 7309
dendrite arm spacing, primary 7307
dendrite arm spacing, secondary (CR) 7309
dendrite arm spacing, secondary (LST) 7308
dimensionless Niyama criterion 7305
field functions 7348
G/v ratio 7305
isotherm speed 7304
liquid residence time 7300
local solidification time 7303
mean cooling rate 7304
Niyama criterion 7305
solidification time 7301
criteria functions, casting *(continued)*
solidification velocity 7304
temperature gradient 7304
using for defect prediction 7297
crystallization
melt-driven 4744
nucleation 4747
solute-driven 4744
workflow 4742
Curle noise source model 3986
custom controls
mesh size 2263
cylinder section, derived part 5801

D

data focus 6350
data mappers
actions 6920
examples 6937
executing 6918
field functions 6929
heat transfer options 6930
interpolation methods 6931
manual setup
surface, volume, beam sources 6912
table source 6915
targets 6916
validity checks 6917, 6926
defogging 3460
deicing 3460, 4872
DEM phase interaction
bonded particles 4571
breaking bonds 4574
conduction heat transfer 4563
elastic-plastic contact 4559
Hertz Mindlin model 4558
impact heating 4576
linear cohesion 4567
linear spring 4560
parallel bonds 4568
particle collisions, visualizing 4578
rolling resistance 4565
simple failure model 4572
derived parts
cell surface 5832

derived parts *(continued)*
exporting to file 5792
flattened surface of revolution 5815
general workflow 5778
implicit sections 5796
isosurface 5805
overview 5775
probes 5825
properties 5844
reading external file 5790
resampled volume 5833
scalar warp 5809
separation/attachment line 5819
streamline 5812
threshold 5807
vector warp 5810
visual tools 5782
vortex core 5822
DES
theory 7495
description
latest 2259
root 2259
Design Manager
licensing 6500
overview 6481
workflow 6505
design of experiments 6489
design optimization 6489
detached eddy simulation
theory 7495
detached eddy simulation (DES)
elliptic blending k-epsilon 3849
initialization and inflow
synthetic eddy method 3857
overview 3842
Spalart-Allmaras 3845
SST k-omega 3852
devolatilization
biomass 4206
coal 4206
DFBI simulation 5039
dimensions 446
dipole noise sources 3986
Discrete Element Method (DEM)
Discrete Element Method (DEM) *(continued)*
artificial viscosity 4579, 4608
boundary conditions 4540
cohesion between particles 4601
conduction heat transfer model 4595
constant rate damage model 4598
contact force models 4590
drag force 4548
drag torque 4550
EDEM, converting from 4616
erosion 4555
field functions 4612
impact heat formulation 4609
injectors 4581
lift force 4551
lift force formulation 4606
literature references 4610
model selection 4513
momentum balance for particle 4589
overview 4511
particle bonding model 4597
particle times scale 4589
particles, *See* particle, DEM
phase interactions 4556
post-processing data 4616
rolling resistance 4602
simple failure model 4598
simulation guidelines 4587
solver settings 4585
Discrete Ordinate Method (DOM), *See* radiation, participating media
dispersed multiphase modeling
applications 4872
boundary conditions 4879
drag force 4885
energy solver 4882
flow solver 4882
heat transfer 4886
impingement, film 4887
impingement, VOF phase 4887
overview 4872
phase interactions 4876
phases 4874
pressure gradient 4885
solver settings 4882

dispersed multiphase modeling *(continued)*
transport equations 4884
two-way coupling 4874
velocity solver 4882
volume fraction solver 4882
workflow 4873
dispersed phases
dispersed multiphase model 4874
displacement
field function 5235
visualization 5264
displacement, solid
constraint 5255
displayers
geometry 5912
scalar 5923
streamline 5974
vector 5945
DOE study 6489
drag coefficient, DEM
formulation 4603
drag force
DEM 4548
dispersed multiphase 4885
drag coefficient 4690
drag correction 4698
drag torque 4550
emulsion 4732
Eulerian 4688
Lagrangian 4296
multiphase segregated flow 4688
drag torque
formulation 4605
rotational drag coefficient 4550
drawing, *See* sketch
droplet breakup
bag breakup 4358
breakup regimes 4355
distortion model 4359
instabilities 4356
stochastic model 4360
stripping breakup 4358
droplet collision
Lagrangian phase 4303
droplet distortion
Lagrangian phase 4313
droplet evaporation, multi-component
Lagrangian phase 4306
mapping phase components 4308
droplet evaporation, single component
Lagrangian phase 4303, 4304
droplet impingement
dispersed multiphase 4887
Lagrangian
Bai-Gosman model 4334
Bai-ONERA model 4338
Satoh model 4341
VOF phase 4430
dynamic pressure
for fans 3629
dynamic Smagorinsky subgrid scale model
overview 3830
theory 7492

E

eddy viscosity models
theory 7440
eddy-current losses 5331, 5391
electric circuit
adding circuit elements 5649
circuit element reports 5654
circuit element types 5645
circuit solver 5639
connecting circuit elements 5650
connecting terminal nodes 5653
creating circuit connections 5653
defining for battery module 5636
modeling 5397
stopping criteria 5642
unit cell time step 5641
electric currents
boundary conditions 5316
circuit model 5397
current source, user-specified 5314
electric field source, user-specified 5315
electrical conductivity options 5313
electro-deposition coating, *See* electro-deposition coating
polarization curve 5316
potential-voltage characteristic 5316
electric machine
2D simulation 969

electric machine *(continued)*
cooling simulation windings 980
diamond endwindings 979
eccentricity modeling 986
electromagnetic simulation 969
end-ring modeling 990
exterior domain setup 970
importing data from SPEED 962, 965
magnet coordinate systems 997
rotor windings 971
simplified windings 971
skew modeling 987
SPEED data, using in STAR-CCM+ 999
stator windings 971
symmetry-reduced models 994
electric potential solver 5341
electrical load on battery
defining 5630
program file 5630
scalar function 5634
table 5633
electrical solver
overview 5536
electro-deposition coating
boundary specification 5481
field functions 5483
model selection 5479
overview 5479
solver settings 5483
electrochemical surface reactions
heat contributions 5468
overview 5443
troubleshooting 5452
electrochemistry
formulation 5485
local ionic charge densities 5438
modeling surface reactions 5446
nomenclature 5503, 7979
overview 5401
electrodynamics
field functions 5342, 5353
electromagnetism
constants 5399
electric currents, modeling, *See* electric currents
electro-deposition coating formulation 5501
electromagnetism *(continued)*
literature references 8010
magnetic fields, modeling, *See* magnetic fields
overview 5305
troubleshooting simulation 5400
electrostatics
field functions 5338
modeling 5309
elliptic blending DES
theory 7499
elliptic blending eddy viscosity model
theory 7465
elliptic blending k-epsilon model
formulation 3874
elliptic blending RST model
theory 7482
emissions
formulation
fuel NOx 7920
empty entities
part curves 1627
part surfaces 1627
emulsion, modeling
anisotropy tensor 4732
drag force 4732
guidelines 4726
material properties 4732
normal relative viscosity 4732
overview 4725
property settings 4732
relative viscosity 4732
workflow 4726
energy
in multiphase segregated flow 4783
Lagrangian model 4302
phase-coupled fluid energy 4784
transfer between Eulerian phases 4752
energy in solids
coupled approach 3493
formulation 3542
segregated approach 3494
solid shells 3495
energy source, user-defined
Lagrangian phase 4303
equation of state
constant density 4630

equation of state *(continued)*
Eulerian phase 4629
ideal gas 4631
Lagrangian phase 4296
polynomial density 4632
user-defined 3231, 4633
erosion modeling
DEM 4555
DEM model formulation 4600
field functions 4833
model overview 4832
Eulerian multiphase mixture modeling, *See* multiphase mixture modeling
Eulerian multiphase modeling
activating model 4619
defining phases 4620
overview 4618
phase materials 4628
setting phase conditions 4627
See also multiphase segregated flow modeling
Eulerian phase
equation of state 4629
user-defined potential force 4636
viscous regime 4635
evaporation, fluid film
model formulation 4499
model selection 4471
solution procedure 4502
workflow 4471
evaporation, VOF
field functions 4932
formulation 4987
model 4927
workflow 4927
exact imprinting method 6933
excitation coil 5326
exothermic solids 3497
expert driver
plots 3266
explicit algebraic Reynolds stress models (EARSM)
overview 3783
exporting from STAR-CCM+
mesh 2800, 6623
mesh and solution data 2800, 6623
extract volume
operation 2213
extruder in parts
operation 2254
extrusion 3400

F

face
mesh definition 2145
quality metric 2672
fans
and rotating reference frames 3626
and sliding mesh 3626
blade angle 3664
centrifugal 3652
ducted 3630
dynamic pressure 3629
exhausting 3630
forcing 3630
impeller
swirl 3656
intersection tolerance 3651
momentum source
activating 3660
performance curve
polynomial 3642
table 3639
temperature scaling 3646
performance curves
scaling 3666
performance data 3659
rotation rate 3664
static pressure 3629
swirl 3647
total pressure 3629
verbosity 3650
FE Stress solver 5241
feature curves 2324
See also part curves
Ffowcs Williams-Hawkings aeroacoustics
guidelines 4054
models 4025
receivers 4035
surfaces 4040
Fick's Law 3558
field functions
array 401

user field functions *(continued)*
batteries 5670
boiling, fluid film 4476
casting 7348
condensation, fluid film 4473
condensation, VOF 4932
contact time 4915
criteria functions, casting 7348
data mappers 6929
DEM 4612
eddy-current losses 5331, 5391
electro-deposition coating 5483
electrodynamics 5342, 5353
electrostatics 5338
Element Type 5260
erosion 4833
evaporation, fluid film 4473
evaporation, VOF 4932
evaporation/condensation model 4655
film wave stripping 4468
film-VOF mass transfer rate 4485
flow stop, VOF melting and solidification 4965
fluid film thickness 4449
general properties 440
hysteresis losses 5331, 5391
introduction 399
macro porosity 7292
magnetohydrodynamics 5394, 5395
mass error rate 4918
mushy zone permeability 4970
n-phase mixture model 4648
ohmic heating 5393
position 401
S-Gamma model 4853
scalar 401
segregated time step control 7350
slip velocity model 4652
slurry viscosity 4969
solid stress (FE) 5235, 5252
solid stress (FV) 5286
solid volume fraction, fluid film 4480
surface tension 4483
system field functions 399
tensor 402
two-phase thermodynamic equilibrium model 4658

user field functions *(continued)*
types 401, 402
user field functions
examples 437
syntax 417, 421, 427, 432, 436
values interpolation 415
vector 401
VOF waves 5013
wall boiling 4812
wall porosity 4926
file tables 268
file-based coupling
Abaqus 6676
ANSYS 6679
Ensight 6709
es-ice and pro-STAR 6698
mapping data 6636, 6661
NASTRAN 6685
RCAS 6717
TAITherm 6693
film stripping
dispersed multiphase 4466
edge stripping 4469, 4495
Lagrangian multiphase 4465
overview 4464
wave stripping 4467, 4492
Finite Element Analysis (FEA)*, See* solid stress modeling
finite element method
mesh 5220
solvers 5241, 5243
flamelet
overview 4138
flamelet table
adaptive gridding 4164
chemical equilibrium 4147
dimensions 4164
flamelet generated manifold 4159
steady laminar flamelet 4152
flamelet tables
overview 4139
flash boiling
formulation 5001
Homogeneous Relaxation model 5001
model properties 4958
flow

flow *(continued)*
boundary conditions 3321, 3373
reference values 3376
flow types overview 3213
fluid film modeling
boiling, *See* boiling, fluid film
condensation fluid film 4470
conservation equations 4488
evaporation, *See* evaporation, fluid film
flow model 4451
formulation overview 4486
impingement, *See* impingement modeling
literature reference 7764
melting, *See* melting, fluid film
model selection 4448
nomenclature 4508
overview 4442
phase interactions 4453
phase models 4450
resolving as VOF phase 4484, 4490
shell region 4444
solidification, *See* solidification, fluid film
solver settings 4449
stripping, *See* film stripping
surface tension 4480, 4483
surface tension formulation 4491
volume fraction 4490
workflow 4444
fluid flow and energy
guidelines
boundary conditions 5693
conjugate heat transfer (CHT) 5694
cyclone separators 5698
freezing solvers 5698
meshing 5692
thermophysical properties 5693
fluid stream
reference 4167
Fluid-Structure Interaction (FSI)
using FE solid stress 5231, 5249
free molecular flow 3222
free surface
definition 4258
free surface flow 3400
frequency response test 5705
fully developed flow 3357

G

gamma re theta transition model
correlations definition using field functions 3894
field functions 3895
formulation 3906
free-stream edge 3893
guidelines 3904
overview 3893
solver 3895
gamma transition model
field functions 3900
formulation 3914
guidelines 3904
overview 3900
solver 3900
gas dissolution, VOF
bubble growth rate 4993
enthalpy change 4999
formulation 4993
mass transfer 4993
mass transfer model formulation 4993
model parameters 4948
multi-component 4993
overview 4945
Rayleigh–Plesset equation 4993
simulation workflow 4945
gasification
coal 4206
particle 4206
general job submission mode 6495
geometry
export from 3D-CAD 1450
exporting from parts 1637
requirements for solid stress (FE) 5212
solid modelling, *See* 3D-CAD
geometry format
IGES 1492
Parasolid 1493
STEP 1492
VDA 1493
geometry formats
CATIA 1494
Creo 1494
NX 1495
SolidWorks 1495

geometry guide
imprinting 1597
intersecting 1597
subtracting 1597
uniting 1597
geometry parts
combining 1594
converting from 3D-CAD 1536
creating 1540
duplicating 1593
importing 1539
overview 1517
reflecting 1592
rotating 1591
scaling 1588
transforming 1585
translating 1590
geometry repair, *See* CAD repair
Goldstein axisymmetric noise source model 3987
granular drag force
multiphase segregated flow 4716
gravity 3352
Gravity 3351
gray thermal radiation 3929, 3958
gridgen mesh format 2161
GT-POWER, *See* GT-SUITE
GT-SUITE, co-simulation 6810
guidelines
aeroacoustics
automobile aerodynamics 5700
axial fans 5709
freestream boundary condition 5712
liquid media 5712
mesh issues 5712
post-processing 5713
radial compressors 5706
sunroof buffeting 5704
turbulence 5711
wave6 export 5718
aerodynamics
incompressible external 5726
vehicle simulations 5731
air flow in electronic systems
boundary conditions 5738
forced or natural convection 5738
vehicle simulations *(continued)*
air flow in electronic systems *(continued)*
geometry preparation 5735
mesh preparation 5737
monitoring 5740
physics set-up 5738
solution set-up 5740
automobile aerodynamics 5700
cell aspect ratio 5692
fluid flow and energy
boundary conditions 5693
conjugate heat transfer (CHT) 5694
cyclone separators 5698
freezing solvers 5698
meshing 5692
thermophysical properties 5693
incompressible external aerodynamics
domain settings 5726
mesh settings 5727
physics setup 5726
large eddy simulation (LES) 3835
mixing applications 5692
prism layers 5692
transonic/supersonic
domain settings 5728
mesh settings 5729
physics setup 5729
solution procedure 5731
transonic/supersonic aerodynamics
steady state RANS 5728
vehicle simulations
inner iterations 5733
mesh settings 5732
model selection 5732
monitors and reports 5734
time-step 5733

H

HAWT (Horizontal Axis Wind Turbine) 3713
heat conduction 3429
heat exchangers
dual-stream
formulation 3532
single-stream
formulation 3531

heat flux averaging
circumferential 3538
heat transfer
boiling
formulation 3526
boundary conditions 3444
conduction 3429
convection 3430
dispersed multiphase 4886
droplet impingement 4317
environmental conditions 3444, 3445
exothermic solids 3497
in solid shells 3495
moisture boiling 3456
optimization 3435
radiation 3432
validation 3434
verification 3434
heat transfer coefficients 3437
helicopters 3687
histogram tables 268
hybrid-BCD 3288
hydroacoustics 3979, 4064, 5712
hyperbolic stretching
definition 2490
hysteresis losses 5331, 5391

I

ice accretion
Lagrangian 4345
ice layer 3461
ideal gas
entropy relation 3243
ignitors
EBU 4088
fixed temperature 4088
flame area density 4088
progress variable 4088
subgrid spark 4088
impingement modeling
dispersed multiphase 4461, 4881
Eulerian multiphase 4462
impingement model 4461
inertial impingement model 4463, 4498
Lagrangian multiphase 4457
impingement modeling *(continued)*
overview 4456
VOF multiphase 4881
implicit sections 5796
importing geometry 1486
importing into STAR-CCM+
mesh and solution data 6600, 6602
importing volume mesh 2161
imprint
bodies in 3D-CAD 1341
operation on parts 2222
surface repair mode 1858
incompressible external aerodynamics
domain settings 5726
mesh setting 5727
physics setup 5726
initial conditions
overview 2848
setting for continuum 2849
setting for region 2852
injector
conditions 4382
cone angle table 4381
initial velocity of particles 4395
lattice injector 4581
overview 4362
parcel streams 4399
particle diameter 4391
particle flow rate 4387
particle mass 4393
particle size 4391
particle size distributions 4373
random injector 4583
table injectors 4377
types 4363
using in DEM 4581
using in Lagrangian multiphase 4369
values overview 4396
interface turbulence damping
multiphase segregated flow 4711
interfaces
solid stress (FE) 5212
inviscid fluid
AUSM+ flux evaluation scheme 3253
ionic species flux

ionic species flux *(continued)*
modeling 5429
overview 5428
ionic wind
overview 5407
stability 5438
isosurface, derived part 5805

J

jacobian
analytical 7883
numerical 7883

K

k-epsilon mode
AKN low-Reynolds 3759
k-epsilon model
elliptic blending 3760
field functions 3780
Lag elliptic blending 3760
low-Reynolds number approach 3757
overview 3757
realizable 3759
realizable two-layer 3760
selecting models 3761
solvers 3778
standard 3759
standard low-Reynolds 3759
standard two-layer 3759
theory 7444
turbulent viscosity scaling 3823
two-layer approach 3758
V2F 3761
wall treatment
formulation 3872, 3874
k-omega model
field functions 3799
overview 3781
selecting models 3784
solvers 3797
SST 3783
standard 3782
theory 7454
turbulent viscosity scaling 3823
wall treatment
wall treatment *(continued)*
wall treatment *(continued)*
formulation 3875
Knudsen number 3222
Kolmogorov method 7892

L

Lag elliptic blending k-epsilon model 3760
Lagrangian multiphase modeling
applications 4263
boundary conditions 4320
dynamic load balancing 4429
field functions 4434
injectors, *See* injector
overview 4264
particle tracking, *See* particle tracking
passive scalars 4314
phase models 4288
phase parcels 4265
post-processing data 4439
solution methodology 7853
solver settings, *See* Lagrangian Multiphase solver
time models 4283
workflow 4267, 4280
Lagrangian Multiphase solver
load-balancing settings 4285
overview 4284
PISO unsteady solution settings 4285
steady solution settings 4284
two-way coupled solution settings 4285
unsteady solution settings 4284, 4285
Lagrangian phase
coupling with continuous phase 4310
overview 4265
parcel depletion 4318
Lagrangian phase parcel 4265
large deformations, *See* nonlinear geometry
large eddy simulation
theory 7489
large eddy simulation (LES)
field functions 3830
guidelines 3835
initialization and inflow
synthetic eddy method 3857
model selection 3826

subgrid scale models *(continued)*
overview 3824
subgrid scale models
dynamic Smagorinsky model 3830
Smagorinsky model 3830
WALE model 3830
launching STAR-CCM+ 1033
least squares method 6931
LES
theory 7489
level surface for volume rendering 5834
Li-ion battery cells
Li-ion cell report 5487
modeling 5411
overview 5409
troubleshooting 5416
licensing 24
lift force
DEM 4551
formulation, DEM 4606
multiphase segregated flow 4701
particle shear formulation 4606
particle spin formulation 4606
shear lift coefficient 4298, 4553
spin lift coefficient 4551
Lighthill stress tensor 4056
Lilley noise source model 3988
line probe, derived part 5826
linear pressure-strain model
theory 7479
linear pressure-strain two-layer model
theory 7481
Linearized Euler Equation (LEE) noise source model 3989
liquid film, *See* fluid film
load on solid 5255
loads
mechanical loads
application 5223
body loads 5235
nonlinear 5254
thermal loads
application 5225

M

Mach number 3222
magnetic fields
boundary conditions 5323
eddy-current losses 5331, 5391
electric current density sources 5323
excitation coil 5326, 5387
Finite Element 5357
Finite Volume 5366
hysteresis losses 5331, 5391
magnetic flux linkage 5326, 5387
magnetic vector potential 5357, 5366
permanent magnets 5324, 5385
remanent magnetic flux density 5385
transverse magnetic potential 5375
magnetic permeability 5320
magnetism, *See* electromagnetism
magnetohydrodynamics
field functions 5394, 5395
modeling 5336
manual study 6489
mass conservation, VOF
mass error rate model 4918
mass error rate settings 4899
mass flowrate
reporting 3336
mass fraction 3548
mass transfer, Eulerian
between phases 4738
dissolution mass transfer rate 4739
droplet evaporation 4740
equilibrium coefficient 4750
interphase reaction 4755, 4756
Nusselt number, continuous phase 4753
Nusselt number, dispersed phase 4753
reaction components 4757
Sherwood number, continuous phase 4749
Sherwood number, dispersed phase 4749
user-defined interphase mass transfer flux 4751
material data 3195
material database, casting
comparing materials 7256
creating datasets 7269
creating materials 7266
creating owner groups 7275
database editor 7239, 7240
dataset

material selection *(continued)*
dataset *(continued)*
exporting 7255
overview 7250
viewing curves 7254
viewing details 7251
editing datasets 7269
editing materials 7266
editing owner groups 7275
exporting materials
overview 7258
specifying datasets 7258
to .scm file 7260
to STAR-CCM+ 7260
to XML file 7260
importing materials 7283
material database builder
reference 7277
material selection
dataset number 7245
from material property 7249
icon list 7244
menu 7243
methods 7242
search history 7246
string search 7244
user-defined data 7242
viewing material details 7246
online help 7241
overview 7237
private data 7262
user-defined data 7263
viewing material properties 7248
material model
Lagrangian phase 4294
material, multi-component
Lagrangian species 4309
materials
linear isotropic elastic 5244
melting, fluid film
model formulation 4506
model selection 4478
workflow 4477
melting, VOF
flow stop mass compensation 4972, 5002
melting, VOF *(continued)*
flow stop model 4972
formulation 5002
model parameters 4965
mushy zone for flow resistance 5002
mushy zone permeability 4970
mushy zone relative velocity 4974
overview 4961
simulation workflow 4963
slurry viscosity model 4969
slurry viscosity model formulation 5002
solid volume fraction variation 4965
switching between slurry viscosity models 5002
mesh
bad cells 2700
baffles 2197
concurrent meshing 2153
conformal mesh 2159
descriptions 2259
directed meshing 2593
exporting 2800, 6623
face quality 2672
inherited parts 2248
invalid cells 2717
local meshing 2196
mesh diagnostics 2684
mesh metrics 2677
meshing in parallel 2150
operations
fill holes 2216
overview 2141
parts-based meshing 2164
priority indexing 2242
region-based meshing 2302
remeshing intersection curve 2241
representations 2727
scaling the mesh 2787
skewness angle 2697
surface mesh quality 2672
surface wrapper
contact prevention 2182
gap closure 2392
partial wrapping 2177
visualizing the mesh 2665
volume mesh quality 2676

surface wrapper *(continued)*
wake refinement 2297
workflow 2157
mesh definitions
cell 2144
face 2144
vertex 2144
mesh refinement
based on solution 2579
custom controls 2263
downstream wake 2570
part curves 2266
part surfaces 2270
parts 2281
within a volume 2284
mesh settings
automobile aerodynamics 5702
sunroof buffeting 5705
mesh size
customizing 2263
mesh,2D 2653
mesh,axisymmetric 2653
mid-side nodes 5220, 5235, 5260
modeling solid stress
mesh 5260
models 295
moisture boiling 3456
mole fraction 3551
monitors 309, 6257
motion 3025
multi-component material 3079
multi-part assemblies, *See* multi-part solids
multi-part solids
in solid stress (FE) 5214
multiband thermal radiation 3960
multiphase flow
available models 4253
modeling overview 4251
multiphase mixture
definition 4254
multiphase mixture modeling
algebraic slip model 4646, 4658, 4670
available models 4645
condensation 4655
drift flux model 4646, 4658, 4671
multiphase mixture modeling *(continued)*
evaporation 4655
evaporation/condensation model 4665
literature references 4674
n-phase mixture model 4648, 4662
overview 4645
phase interactions 4646
slip velocity model 4652
solver settings 4900
two-phase thermodynamic equilibrium model 4658, 4668
multiphase segregated flow modeling
boiling, *See* boiling, bulk
boundary permeability 4863
combustion 4785
emulsions, *See* emulsion, modeling
energy 4783
granular drag force 4716
interphase mass transfer 4738
interphase turbulence transfer 4781
LES particle-induced turbulence 4777
literature references 7706
mass source terms 4868, 4870
model description 4676
overview 4675
particle-induced mixing 4779
particle-induced turbulence 4775
particle-phase flow 4813
phase interactions 4681
See also phase interactions
selecting model 4679
solver settings 4758, 4765
See also multiphase solvers
suspensions, *See* suspension, modeling
turbulence 4767
turbulence response model 4772
wall boiling, *See* wall boiling
multiphase solvers
multiphase segregated flow 4759
phase coupled velocity 4760
pressure 4762
segregated Volume Fraction 4900
setup guidelines 4765
volume fraction 4763
multiple flow regime
boiling mass transfer rate 4792

multiple flow regime *(continued)*
condensation mass transfer rate 4792
drag force parameters 4720
phase interaction model 4718
surface tension 4723
multiple objective tradeoff study (Pareto front) 6489
MUSCL 3rd-order/CD 3288

N

NASTRAN, file-based coupling 6685
native CAD formats 1493
nearest neighbor method 6931
NetBeans 794
NetBeans modules
deactivating 776
installing 776
managing 776
neutral formats 1492
Newton's law of cooling 3437
ngeom mesh format 2161
noise prediction
far-field 4024
noise sources
Curle model 3986
dipole 3985, 3986
Goldstein axisymmetric model 3987
Lilley model 3988
Linearized Euler Equation (LEE) model 3989
Proudman model 3987
quadrupole 3985, 3986
trailing edge 3980
turbulence-structure interaction 3980
turbulent shear flow 3985
vortex shedding 3980
non-local thermal equilibrium plasma 5511
nonfollower forces, *See* loads nonlinear
nonlinear geometry
controls 5254
workflow and guidelines 5229
normal direction constraint check, data mapping 6917, 6926
nucleation
crystallization 4747
particle size distribution 4747
user-defined 4747
numerical ventilation, eliminating 4906

O

object tree focus 787
objective 6486
ohmic heating
modeling 5334
reference 5393
operation
badge for 2D meshing 2654
extract volume 2213
extruder 2254
surface wrapper 2175, 2177
operator splitting algorithm 7880
optimization study 6489
output table 6497
overset 2904
oxidation
biomass 4206
char 4206

P

parallel 1063
parallel meshing, *See* concurrent meshing
Pareto optimization 6489
part contacts 1655
part curve 1530
participating media radiation 3952
particle tracking
animating tracks 4424
boundary sampling 4413
creating track file 4411
example 4418
mapping data to parcels 4416
overview 4405
reading track file 4411
track file model 4407
using in analysis 4416
particle-phase flow
adaptive multiple size-group model 4835, 4840
erosion 4832
granular energy transfer 4821
granular pressure models 4814
granular temperature models 4817
inertial breakup settings 4858
inertial coalescence settings 4858

particle-phase flow *(continued)*
modeling particle size distribution 4838
overview 4813
particle breakup 4846, 4858
particle coalescence 4846, 4858
particle size distribution models 4834
S-Gamma model 4837
S-Gamma size distribution 4853
S-Gamma solver settings 4853
setting granular pressure at boundary 4827
setting granular pressure within phase 4822
solid pressure force 4830
viscous breakup settings 4858
viscous coalescence settings 4858
particle, DEM
boundary values 4546
capsule 4536
clumps model 4524
coarse grain model 4521
composite model 4522
cylindrical 4534
modeling 4516
momentum balance 4589
phase boundary conditions 4540
polyhedral 4538
rolling resistance 4602
specifying shape 4527
sphere-filled part 4533
spherical model 4519
times scale 4589
wall boundary conditions 4543
particle, granular
drag force 4716
flow, *See* particle-phase flow
multiphase segregated flow 4715
phase interaction 4715
particle, Lagrangian
charged 4299
massless 4291
material 4292
modeling with DEM, *See* particle, DEM
porosity 4319
porosity formulation 7828
residence time 4312
parts 290
passive scalars
residence time 3348
transport equations 3340
passive scalars,contact transfer 4577
performance assessment 6489
performance curve
polynomial 3642
table 3639
temperature scaling 3646
performance curves
scaling 3666
performance value 6486
permeability 3552
phase
definition 4256
Lagrangian, *See* Lagrangian phase
mass transfer 4738
phase condition
definition 4256
phase interaction
component mapping 4456
Continuous-Dispersed Phase 4682
creating 4259
definition 4258
Discrete Element Method 4556
See also DEM phase interaction
dispersed multiphase 4876
Dispersed-Physics Continuum 4876
Dispersed-VOF 4876
dissolution mass transfer rate 4739
drag coefficient 4690
drag correction 4698
drag force 4688
drag force, multiple flow regime 4720
droplet evaporation 4740
equilibrium coefficient 4750
Film-Dispersed Phase 4454
Film-Eulerian Phase 4454
Film-Lagrangian Phase 4454
Film-Physics Continuum 4454
Film-VOF Phase 4454
granular particle phases 4715
inertial breakup settings 4858
inertial coalescence settings 4858
interaction area density 4687

phase interaction *(continued)*
interaction length scale 4684
interface turbulence damping 4711
interphase energy transfer 4752
interphase reaction 4755, 4756
Lagrangian-Eulerian 4431
lift force 4701
mass transfer 4738
material properties 4715
model selection 4260
multiple flow regime 4718, 4720
Nusselt number, continuous phase 4753
Nusselt number, dispersed phase 4753
particle breakup 4858
particle coalescence 4858
passive scalar 4431
reaction components 4757
Sherwood number, continuous phase 4749
Sherwood number, dispersed phase 4749
surface tension, multiple flow regime 4723
turbulent dispersion force 4708
user-defined interphase mass transfer flux 4751
virtual mass coefficient 4712
virtual mass force 4712
virtual mass stress 4714
viscous breakup settings 4858
viscous coalescence settings 4858
VOF boiling models 4937
VOF phase replacement 4909
VOF-VOF Phase 4894
wall lubrication force 4705
physics setup 2822
plane section, derived part 5797
plasma
formulation 5518
model reference topic 5513
modeling
arc welding 5521
circuit breakers 5521
cold, non thermal plasmas 5511
non-LTE plasma 5511
plasma torches 5521
nomenclature 5525
overview 5508
selecting the Coupled Plasma Electron model 5511
modeling *(continued)*
transport equations 5518
types of 5508
plasma actuators
stability 5438
plot3d mesh format 2161
plots 309, 6273
point probe, derived part 5825
polymerization
overview 4132
polynomial density gas 3231
poor quality cells 2700
porosity
Lagrangian particle 4319, 7828
porosity, wall, *See* wall porosity
porous media
and convection
in tube flow 3585
and heat 3567
and turbulence scale 3567
and viscous flux 3567
directed meshing 3563
isotropic resistance 3576
porosity 3567
porous resistance
single-phase fluids 3567
temperature 3584
viscous resistance 3575
porous resistance
single-phase fluids 3567
potential forces
Eulerian phase 4636
flow in porous medium example 4641
foam drainage example 4638
pre-allocation mode 6495
precision 1031
presentation grid, derived part 5828
pressure
initializing flow 3298
pressure coefficient 443, 444, 3380
pressure gradient
dispersed multiphase 4885
Lagrangian 4298
prism layers
guidelines 5692
product configurations 24

project plot 6497
proous media
and viscous flux
in tube flow 3585
Proudman noise souce model 3987
proximity constraint check, data mapping 6917, 6926

Q

quadratic pressure-strain model
theory 7482
quadrupole noise sources 3986
quality
volume mesh 2677
quasi-unsteady losses 5389

R

radial compressors
guidelines
solver settings 5709
steady state performance curve 5706
time-step 5708
radial tables 268
radiation
gray thermal 3929, 3958
in solid shells 3495
modeling in VOF multiphase 4944
multiband thermal 3960
participating media 3952
shell regions 3929
solar loads 3969
surface-to-surface (S2S)
post-processing 3947
view factors 3940
radiation (thermal) 3432
RANS
theory 7439
RANS turbulence
and aeroacoustics 3982
Rayleigh damping
modeling 5249
RCAS, file-based coupling 6717
reacting
model reference 4081
reacting channel, co-simulation 4233, 6875
reacting channels
reacting channels *(continued)*
modeling 4234, 6877
reacting flow
applications
burners 4072
catalytic convertors 4070
coal combustion 4069
fluid catalytic cracking 4069
fluid film reactions 4073
furnaces 4072
gas turbines 4068
glass furnaces 4071
methane reformation 4073
polymerization 4070
reacting channels 4073
reacting surfaces 4071
selective catalytic reduction 4073
trickle bed reactors 4073
calculation speed-up
clustering 4102
in situ adaptive tabulation 4102
flame propagation 4140
overview 4065
selecting models 4065
workflow 4075
reacting species transport
overview 4099
reactions
fluidized bed 4208
interphase 4205
real gas
entropy relation 3243
realizable k-epsilon model 3759
realizable two-layer k-epsilon model 3760
recirculation
in segregated approach troubleshooting 3280
reducing complexity, *See* defeaturing
reduction
ferric oxide 4206
reference frames
formulation 3018–3021
operations 3015
types 3011
reference pressure 3224
refining mesh, *See* mesh refinement
reflections 4020

Refraction 3956
regions 298
reports 309, 6179
representations 2727
resampled volume, derived part 5833
residence time
    passive scalars 3348
residuals
    monitoring 2884
    overview 2883
    plotting 2885
Resource Manager 6532
response 6486
Reynolds Averaged Navier-Stokes modeling
    and aeroacoustics 3982
Reynolds stress transport model
    field functions 3814
    overview 3801
    solvers 3812
    wall treatment
        formulation 3877
Reynolds stress transport models
    theory 7476
Reynolds-averaged Navier-Stokes (RANS)
    averaging process 3746
    eddy viscosity models
        k-epsilon model 3757
        k-omega model 3781
        Spalart-Allmaras model 3748
    initialization 3815
    Reynolds stress transport model 3747, 3801
    selecting models 3747
    theory 7439
    troubleshooting 3816

S

scale-resolving simulation
    theory 7489
scenes
    animations 5996
    annotations 6086
    colormaps 6111
    DFBI 5988
    displaying geometry 5912
    displaying scalars 5923
scenes *(continued)*
    displaying streamlines 5974
    displaying vectors 5945
    views 6135
scripting
    adding macro to toolbar 488
    batch mode for macros 477
    debugging macro 479
    default macro folder 495
    default macro settings 493
    double-precision real number entry 465
    editing macro 463
    integer entry 463
    interactive dialog 469
    Java API 490
    overview 458
    playing macro 474, 477
    recording macro 460
    running multiple simulations 485
    subtask macros 487
    text entry 468
segments 5255
segregated energy
    solver convergence 5620
separation/attachment line, derived part 5819
shape functions
    data mapping 6931
SHERPA search algorithm 6493
Sherwood number
    Lagrangian phase 4307
simulation
    automating tasks, *See* scripting
    changing while running 2889
    clearing solution 2877
    convergence 2890
    customizing with user code, *See* user code
    initial conditions 2848
    initializing solution 2875
    monitoring progress 2888
    monitoring residuals 2882
    overview 288
    running 2880
    running, overview 2847
    saving automatically 2893, 2895
    saving checkpoint file 2896

simulation *(continued)*
saving intermediate solutions 2892
solution process overview 2875
solver settings 2855
steering solution 2888
stepping 2881
stopping 2886
stopping criteria 2856
simulation assistant
object tree focus 787
simulation assistants
adding Java classes 798
adding Java packages 796
adding libraries 802
adding XHTML files 799
annotations 811
classes 810
Client Server Objects (CSOs) 811
collapsible information 787
conditions 789, 813
JAR files 809
Java files 809
loading 792
lookups 811
NetBeans 794
predicates 813
star actions 786, 810
supplied Java classes 814
tasks 785, 810, 812
triggers 813
wizard 831
workflows 781
XHTML code 810
XHTML files 809, 814
XHTML strings 814
simulation objects 311
single-component material 3079
single-phase boiling
formulation 3526
moisture boiling 3456
overview 4787
See also boiling, bulk
slip flow 3222
Smagorinsky subgrid scale model
overview 3830
Smagorinsky subgrid scale model *(continued)*
theory 7491
snapshot 6497
solar loads 3969
solid electrolyte interface
overview 5417
solid energy
coupled approach 3493
formulation 3542
in solid shells 3495
segregated approach 3494
solid mechanics
definitions 5301
modeling, *See* solid stress modeling
solid stress modeling
bibliography 5267
reference 5235
simulation layout 5212
using the FV method 5268
workflow 5209
solid stress modeling (Finite Volume)
boundary conditions 5286
field functions 5286
solidification, fluid film
model formulation 4506
model selection 4478
solidified mass removal 4477
workflow 4477
solidification, VOF
flow resistance 4961
flow stop mass compensation 4972, 5002
flow stop model 4972
formulation 5002
model parameters 4965
mushy zone for flow resistance 5002
mushy zone permeability 4970
mushy zone relative velocity 4974
overview 4961
simulation workflow 4963
slurry viscosity model 4969
slurry viscosity model formulation 5002
solid volume fraction variation 4965
switching between slurry viscosity models 5002
solution analysis
FE solid stress 5264

solution divergence
activating models 2899, 2900
boundary conditions, changing 2899
Courant number 2901
debugging 2901
initial solution 2898
maximum reports 2902
normalization factor 2901
secondary gradients, omitting 2903
solution accuracy 2900
temporary storage, using 2902
threshold parts 2902
troubleshooting 2898
under-relaxation factors 2901
zeroing reconstruction 2903
solver settings
automobile aerodynamics 5704
sunroof buffeting 5706
solvers 307
soot
two-equation
model reference 4199
space 2965
Spalart-Allmaras DES
theory 7497
Spalart-Allmaras model
field functions 3757
selecting models 3749
solvers 3755
theory 7441
Sparse Direct Solver 5243
species
adding 3555
and coupled flow 3543
and segregated flow 3543, 3554
background component 3546
Lagrangian phase 4309
mass fraction 3548
model properties 3546
mole fraction 3551
multi-component fluids 3545
order of components 3546
permeability 3552
removing 3555
source term 3555
species *(continued)*
temperature 3555
wall boundaries 3552
SPEED
.xGDF file 965
data import wizard 963, 965
data, importing into STAR-CCM+ 962
See also electric machine
speed of sound
formulation 8038
sphere section, derived part 5802
sponge zone 4020
spurious reflections 4020
SST k-omega DES
theory 7500
SST k-omega model 3783
standard k-epsilon low-Reynolds model
temperature flux model
formulation 3821
standard k-epsilon model 3759
standard k-omega model 3782
standard two-layer k-epsilon model 3759
STAR-Cast, *See* casting
static pressure
for fans 3629
stiffness matrix
update techniques 5254
stochastic study 6489
stopping criteria
automatically generated 2857
creating from monitor 2869
disabling 2864
enabling 2864
maximum inner iterations 2863
maximum physical time 2861
maximum steps 2858
minimum inner iterations 2865
monitor-based 2867
overview 2856
stop file 2859
using multiple criteria 2873
strain
field functions 5235, 5252
visualization 5264
streamline, derived part 5812
stress

stress *(continued)*
field functions 5235
visualization 5264
stress analysis, *See* solid stress modeling
study analysis 6497
study inputs 6484
study outputs 6486
study runs 6495
study types 6489
subgrid scale models
dynamic Smagorinsky model 3830
model selection 3826
overview 3825
Smagorinsky model 3830
WALE model 3830
support 50
suppression zone 4020
surface data
mapping 6912, 6921
surface repair 1674
surface tension
contact angle for large-scale interface 4913
contact angle for VOF multiphase 4913
fluid film 4480
hybrid film-VOF 4481
multiple flow regime 4723
VOF multiphase 4911
surface wrapper
operation workflow 2175, 2177
overview 2390
surface-to-surface (S2S) radiation
post-processing 3947
view factors 3940
suspension, modeling
anisotropy tensor 4728
guidelines 4725
material properties 4728
normal relative viscosity 4728
overview 4725
property settings 4728
relative viscosity 4728
workflow 4725
sweep study 6489
swirl 3647
synthetic eddy method
synthetic eddy method *(continued)*
overview 3857
setup 3858

T

tables
common operations 276
copying data between simulations 271
creating 270
exporting 271, 275
file format 281
importing from file 269
plotting data 274
properties and actions 277
types 268
tabular data
mapping 6915
TAITherm, file-based coupling 6693
temperature flux model
formulation 3821
thermal non-equilibrium 3231
thermal radiation
absorption by particles 4313
scattering by particles 4313
thermal solver
overview 5537
thermal strain
definition 5301
thin film 3460
threshold, derived part 5807
time 3001
time-step
automobile aerodynamics 5703
sunroof buffeting 5705
trailing edge noise 3980
transient analysis
explicit unsteady 3256
implicit unsteady 3255
transition
gamma model 3900
gamma re theta model 3893
turbulence suppression model 3885
transitional flow 3222
transonic/supersonic
guidelines

guidelines *(continued)*
guidelines *(continued)*
domain settings 5728
mesh settings 5729
physics setup 5729
solution procedure 5731
transonic/supersonic aerodynamics
guidelines
turbulence model 5728
transport equations
for species 3558
passive scalars 3340
tree
expanding/collapsing 327
turbo slicing tool 1463
turbulence
detached eddy simulation (DES) 3842
large eddy simulation (LES) 3824
Reynolds-averaged Navier-Stokes (RANS) 3746
selecting models 3741
theory 7435
transition 3884
turbulence scale
in porous media 3567
turbulence suppression model
field functions 3892
inflow turbulence 3890
mesh restrictions 3890
model selection 3887
selecting turbulence models 3890
solver 3892
transition boundary 3885
turbulence suppression region 3885
turbulence-structure interaction noise 3980
turbulent dispersion
Lagrangian phase 4311
turbulent dispersion force
multiphase segregated flow 4708
turbulent flow
in multiphase segregated flow 4767
interphase turbulence transfer in multiphase segregated flow 4781
LES particle-induced turbulence in multiphase segregated flow 4777
particle-induced mixing in multiphase segregated flow 4779
turbulent flow *(continued)*
particle-induced turbulence in multiphase segregated flow 4775
turbulence response in multiphase segregated flow 4772
turbulent Schmidt number
for species transport 3558

U

under-relaxation factor
in segregated approach troubleshooting 3280
units
operations 448, 452
user code
applying to particles and parcels 764
examples 758
function interface, C 749
function interface, Fortran 753
function variables 757
functions 740
library, accessing 745
library, creating 743
overview 736
shared-object libraries 738
user-defined equation of state 3231

V

V2F k-epsilon model 3761
V2F model
theory 7471
vehicle simulation
guidelines
inner iterations 5733
mesh settings 5732
model selection 5732
monitors and reports 5734
time-step 5733
vehicle simulations
guidelines 5731
velocity
initializing flow 3298
vertex
mesh definition 2145
virtual disk
blade elements 3687
body force propeller 3671

virtual disk *(continued)*
meshing 3722
performance curve 3713
reports 3729
user-defined method 3722
virtual mass
Lagrangian 4299
virtual mass coefficient
multiphase segregated flow 4712
virtual mass force
multiphase segregated flow 4712
virtual mass stress
multiphase segregated flow 4714
viscoelastic flow 3394
viscosity
artificial, in DEM 4579
viscous energy 3214
viscous flow 3214, 3394
viscous regime
Eulerian phase 4635
viscous resistance 3575
VOF multiphase modeling
boiling 4932
boundary conditions 4899
condensation 4927
contact time 4915
discretization, high-resolution interface capturing 4982
discretization, normalized variable 4981
evaporation 4927
gas dissolution, *See* gas dissolution, VOF
governing equations 4979
guidelines 4977
initial conditions 4897
interface momentum dissipation 4917
latent heat of vaporization 4986
literature references 7743
melting, *See* melting, VOF
mixture composition 4892
model reference 4900
overview 4889
phase definition 4892
phase interactions 4894
radiation 4944
region conditions 4897
replacing phases 4906
VOF multiphase modeling *(continued)*
setting mass error rate 4918
simulation setup 4890, 4891
solidification, *See* solidification, VOF
solver settings 4900
surface tension 4911
surface tension formulation 4984
thermal effects 4762, 4903
wall porosity 4919, 4987
VOF wave modeling
boundary conditions 5029
damping 5026
field functions 5013
fifth order wave 5013
flat wave 5013
forcing 5026
guidelines 5027
irregular wave 5013
literature references 5038
meshing guidelines 5027
model description 5009
overview 5007
period 5008
sinusoidal wave 5013
superposition wave 5013
wake waves 5031
wave formulation 5032
wavelength 5008
volume data
mapping 6912, 6921
volume mesh
quality 2677
vortex core, derived part 5822
vortex shedding noise 3980
vortices
confinement 3364
tracking 3364

W

wake
mesh refinement 2570
WALE subgrid scale model
overview 3830
solver 3830
theory 7494

wall
distance 3923, 3924
functions 3862
roughness 3864
treatment 3862
y+ value 3866
wall boiling
boundary settings 4790
bubble departure diameter 4808
bubble departure frequency 4810
bubble influence wall area fraction 4811
field functions 4812
model selection 4790
nucleation site number density 4806
overview 4802
quenching heat transfer 4811
simulation convergence 4802
simulation setup 4800
wall dryout area fraction 4805
wall laws
formulation 3878
wall lubrication force
multiphase segregated flow 4705
wall porosity
ambient pressure 4923
mass extraction factor 4921
model parameters 4921
model selection 4920
overview 4919
setting at contact interface 4923
setting at porous baffle 4924
setting at wall 4923
warp surface, derived part
scalar 5809
vector 5810
WAVE, co-simulation 6842
wave, gravity, *See* VOF wave modeling
weighted sum of all objectives optimization 6489
workflows
simulation assistants 781

X

xyz tables 268

Y

y+ value 3866, 3878